HARDWARE REFERENCE MANUAL

Accessory 72EX



UMAC Fieldbus Interface

300-603958-0U00

July 9, 2023

Document # MN-000251

C E 🕼 🖁

Copyright Information

© 2023 Delta Tau Data Systems, Inc. All rights reserved.

This document is furnished for the customers of Delta Tau Data Systems, Inc. Other uses are unauthorized without written permission of Delta Tau Data Systems, Inc. Information contained in this manual may be updated from time-to-time due to product improvements, etc., and may not conform in every respect to former issues.

To report errors or inconsistencies, call or email your local Omron representative

Operating Conditions

All Delta Tau Data Systems, Inc. motion controller products, accessories, and amplifiers contain static sensitive components that can be damaged by incorrect handling. When installing or handling Delta Tau Data Systems, Inc. products, avoid contact with highly insulated materials. Only qualified personnel should be allowed to handle this equipment. Before powering, please ensure there is no visible damage to the product.

In the case of industrial applications, we expect our products to be protected from hazardous or conductive materials and/or environments that could cause harm to the controller by damaging components or causing electrical shorts. Our products should not be placed in locations that can accrue a lot of dust, salt, or conductive iron-like powder. When our products are used in an industrial environment, install them into an industrial electrical cabinet or industrial PC to protect them from excessive or corrosive moisture, abnormal ambient temperatures, and conductive materials. If Delta Tau Data Systems, Inc. products are directly exposed to hazardous or conductive materials and/or environments, we cannot guarantee their operation. For your own safety, please keep the product's environmental conditions within the range outlined by the Environment Specifications section that can be located from the table of contents in this manual.

Trademarks

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.



EN Dispose in accordance with applicable regulations.

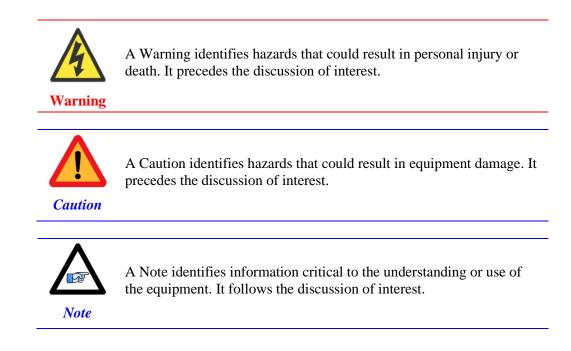
Safety Instructions

Qualified personnel must transport, assemble, install, and maintain this equipment. Properly qualified personnel are persons who are familiar with the transport, assembly, installation, and operation of equipment. The qualified personnel must know and observe the following standards and regulations: IEC364resp.CENELEC HD 384 or DIN VDE 0100 IEC report 664 or DIN VDE 0110 National regulations for safety and accident prevention or VBG 4

Incorrect handling of products can result in injury and damage to persons and machinery. Strictly adhere to the installation instructions. Electrical safety is provided through a low-resistance earth connection. It is vital to ensure that all system components are connected to earth ground.

This product contains components that are sensitive to static electricity and can be damaged by incorrect handling. Avoid contact with high insulating materials (artificial fabrics, plastic film, etc.). Place the product on a conductive surface. Discharge any possible static electricity build-up by touching an unpainted, metal, grounded surface before touching the equipment.

Keep all covers and cabinet doors shut during operation. Be aware that during operation, the product has electrically charged components and hot surfaces. Control and power cables can carry a high voltage, even when the motor is not rotating. Never disconnect or connect the product while the power source is energized to avoid electric arcing.



	REVISION HISTORY							
REV	DESCRIPTION	DATE	CHG	APPVD				
1	Preliminary Manual	SS	SS					
2	Added Power PMAC support and address settings based upon 603958-102	09/24/13	SS	SS				
3	Corrected ACC72EX.Data8[i] references	10/21/13	SS	SS				
4	Added C code and setup examples; corrected typos	07/29/15	DCDP	SS				
5	Fixed Jumper E2 description	03/17/16	SGM	SGM				
6	Added KC Conformity	10/17/18	SM	RN				
7	Added environmental specifications table	09/14/20	SM	RN				
8	Added Mounting and Installation section	12/16/20	SM	RN				
9	Added warning statement and updated drawing illustration for noise chatter in Mounting and Installation section	01/26/21	SM	RN				
А	Added UKCA Marking to front cover and added description in Agency of Approval section	08/11/21	AE	SM				
В	Updated UKCA standard	01/31/22	AE	SF				
С	Updates for Product Lifecycle Management	6/23/23	AA	AA				

Table of Contents

INTRODUCTION	
SPECIFICATIONS	9
Environmental Specifications	9
Agency Approval and Safety	
MOUNTING AND INSTALLATION	
THEORY OF OPERATION	
UBUS Interface	
How ACC-72EX Works	
Turbo PMAC Memory	
Power PMAC Memory	
Hilscher ComX Module Addressing to Turbo PMAC Addressing Conversion	14
Hilscher ComX Module Addressing to Power PMAC Addressing Conversion	
HARDWARE	
E3: UBUS Address	
CS16- Identification	17
Identification Information	
Jumper Settings	
Option Identification Jumpers	
E-Point Jumper Settings	
Communication Option-Dependent E-Point Jumper Settings	
Connector Pinouts	
Fieldbus Port (J4)	
Real-time Ethernet Ports (Ethernet 0 & Ethernet 1)	
Diagnostics Port (Micro A USB)	
DPRAM MEMORY MAP	
DPRAM Blocks	
System Channel	
Handshake Channel	
Communication Channel	
Application Channel	
Auto-Generated Dual-Ported Memory Map	
Address Converter	
Memory Map Generator	
DPRAM DATA PROCESSING	
Non-Cyclic Data Exchange	55
Message or Packets	55

About System and Channel Mailbox	58
Command and Acknowledge	59
Using ulSrc and ulSrcId	62
How to Route rcX Packets	62
Client/Server Mechanism	63
Input/Output Data Image	64
Process Data Handshake Modes	64
Start / Stop Communication	67
Controlled or Automatic Start	
Start / Stop Communication through Dual-Port Memory	67
Reset Command	68
System Reset vs. Channel Initialization	68
Resetting netX through Dual-Port Memory	68
System Reset through Packets	71
SOFTWARE SETUP	72
Required Software Packages	72
SyCon.NET Software Setup	72
ACC-72EX Setup Assistant	
Turbo PMAC Setup for Using ACC-72EX	84
Initialization PLC	
Watchdog Function	85
Enabling the Communication Bus	85
Locating the Input/Output Data Image in PMAC	86
Reading/Writing from/to Input/Output Data Images	87
Power PMAC Setup for Using ACC-72EX	89
ACC72EX[i]. Non-Saved Data Structures	89
C Programming Access to ACC-72EX Structures	92
Global Header for Power PMAC Projects	94
Initialization PLC	
Startup	
Watchdog Function	
Enabling the Communication Bus	
Locating the Input/Output Data Image in PMAC	106
DIAGNOSTICS	108
LEDs	108
PROFIBUS-DP – Master – OPT10	108
PROFIBUS-DP – Slave – OPT11	108
DeviceNet – Master – OPT20	
DeviceNet – Slave – OPT21	109
CANopen – Master – OPT30	
CANopen – Slave – OPT31	110

CC-Link – Slave – OPT51	
EtherCAT – Master – OPT60	
EtherCAT – Slave – OPT61	
EtherNet/IP – Scanner/Master – OPT70	
EtherNet/IP – Adaptor/Slave – OPT71	
Open Modbus/TCP – OPT80	
PROFINET IO – Controller – OPT90	
PROFINET IO – Device – OPT91	
APPENDIX A – SETUP EXAMPLES	
SYCON.net Setup	
RSLogix 5000 Setup	
COMX Test PLC	
APPENDIX B – TURBO PMAC MEMORY MAPS	
APPENDIX C – POWER PMAC MEMORY MAPS	

INTRODUCTION

This manual provides the information needed to configure ACC-72EX, a fieldbus/real-time Ethernet interface for the Turbo or Power UMAC. The ACC-72EX is equipped with a "gateway" daughter card that allows the UMAC (also referred to as host application) to send and receive data through the supported fieldbus/real-time Ethernet protocols. The gateway used is the COMX CN series manufactured by the Hilscher Corporation. Relevant hyperlinks are provided in Appendix D for in-depth information regarding these modules.

There are three connectors located on the front of the ACC-72EX:

First, a Micro B USB connector, which is specified as "Diagnostic Port," and provides USB connectivity to Hilscher's "SyCon.NET" software.

The second connector, which is referred to as the "Fieldbus Port", is a 9-Pin Male D-Sub connector which is used for connecting the fieldbus link to ACC-72EX. The fieldbus protocols supported through this port are:

- PROFIBUS-DP Master OPT10
- PROFIBUS-DP Slave OPT11
- DeviceNet Master OPT20
- DeviceNet Slave OPT21
- CANopen Master OPT30
- CANopen Slave OPT31 (No Longer Available)
- CC-Link Slave OPT51 (No Longer Available)

The third connector is composed of two RJ-45 ports which provide connection to real-time Ethernet networks. The following real-time Ethernet protocols are supported through these ports:

- EtherCAT Master OPT60
- EtherCAT Slave OPT61
- EtherNet/IP Scanner/Master OPT70
- EtherNet/IP Adaptor/Slave OPT71
- Open Modbus/TCP OPT80
- PROFINET IO Controller OPT90
- PROFINET IO Device OPT91

The protocol is dependent upon the equipped COMX gateway. The hardware cannot be programmed for an alternate protocol or change from slave to master or vice versa. However, should the COMX gateway be replaced with one supporting another protocol, the baseboard would function properly as a communications link to UMAC. In this case, proper jumper settings should be set up to ensure proper functionality on communication lines and option detection.

Most gateway cards get their power from the UBUS back plane; however, the DeviceNet option (Options 3 & 4) requires an external 24 VDC power supply through the "Fieldbus Port."

SPECIFICATIONS

Environmental Specifications

Description	Specification	Notes
Operating Temperature	0°C to 55°C	
Storage Temperature	-25°C to 70°C	
Humidity	10% to 95 %	Non-Condensing

Agency Approval and Safety

Item	Description
CE Mark	EN61326-1
EMC	EN55011 Class A Group 1
	EN61000-4-2
	EN61000-4-3
	EN61000-4-4
	EN61000-4-5
	EN61000-4-6
Flammability Class	UL 94V-0
КС	EMI: KN 11
	EMS: KN 61000-6-2
UKCA	2016 No. 1091

사 용 자 안 내 문

이 기기는 업무용 환경에서 사용할 목적으로 적합성평가를 받은 기기로서 가정 용 환경에서 사용하는 경우 전파간섭의 우려가 있습니다.

한국 EMC적용제품 준수사항

본 제품은 전파법(KC 규정)을 준수합니다. 제품을 사용하려면 다음 사항에 유 의하십시오. 이 기기는 업무용 환경에서 사용할 목적으로 적합성평가를 받은 기기로서 가정용 환경에서 사용하는 경우 전파간섭의 우려가 있습니다. 입 력에 EMC 필터, 서지 보호기, 페라이트 코어 또는 1차측의 케이블에 노이즈 필터를 입력으로 사용하십시오.

MOUNTING AND INSTALLATION

To connect a UMAC accessory, simply slide the board into any open slot of the UMAC rack. Customarily, accessories are installed from left to right as follows:

←		/				
0	CPU	Fieldbus	Axes Cards	I/O Cards	Power Supply	0
		ACC-72EX				

10, 15, or 21-slot rack

Prior to installation, make sure that you have set the jumpers and address settings to your desired requirements. Use the guide tracks that have been installed in the empty slots of your UMAC system when installing a board.

As you slide the board into the rack, use caution to ensure none of the components on the board make contact with the front plates of the boards on either side. Getting the front plate flush with the front of the rack and turning the front screws firmly will ensure a good connection with the backplane.

When removing a board from the system, the user must first pull out any wired connections from the top, bottom, and front panels then loosen the pem-nuts on the front of the rack. Next, the user can gently pull the board from the rack and use caution to ensure that none of the components on the board make contact with the boards on either side.

System malfunction can occur due to noise/chatter if the ACC-72EX is placed outside of the recommended order as seen in the illustration above. Note that the ACC-72EX is a Fieldbus device and should be placed adjacent to the CPU, and as close as possible to ensure smooth communication.

Warning

THEORY OF OPERATION

The ACC-72EX board is organized as a motherboard/daughter board system. The motherboard contains the UBUS interface, diagnostics, and the fieldbus connections. The daughter board contains the intelligence (firmware which will be referred as netX) and the interface electronics required for each fieldbus. There is a different daughter board for each fieldbus.

The netX firmware on the daughter board implements each fieldbus communications protocol. Fieldbus data is transferred to/from the fieldbus and placed in a Dual-Ported RAM (memory) on the daughter board. The structure of this DPRAM is given later in this manual and is common for all the field buses. ACC-72EX supports up to 64K DPRAM on each device (one full chip-select width).

The PMAC side of the DPRAM is interfaced to the UBUS. PMAC programs access the fieldbus data by reading or writing data to memory addresses corresponding to the location of the PMAC Gateway 3U board's DPRAM.

UBUS Interface

The UBUS is Delta Tau's bus interface for the UMAC controller. The ACC-72EX maps to the UBUS as a DPRAM style board. It occupies contiguous memory locations (both X and Y memory for Turbo PMAC) of the lower two bytes of the 24-bit (middle 16 bits of each 32 bit word for Power PMAC), DPRAM addresses. Because the DPRAM size supported on ACC-72EX can be as large as 64K, each card will occupy one full Chip Select addressing space. There can be a maximum of two ACC-72EX cards per Turbo/Power UMAC (cannot be in a MACRO Station).

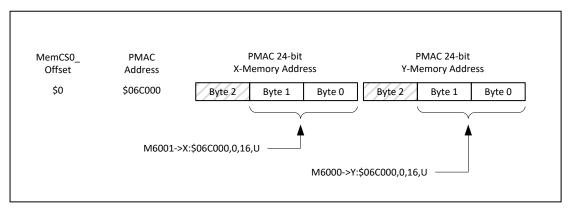
M-Variables can be mapped to these locations to move data to and from the fieldbus and PMAC. In addition to fieldbus data, there is a block of memory that indicates the ACC-72EX's status.

How ACC-72EX Works

- 1. The ACC-72EX organizes fieldbus bytes in dual-port memory on the COMX module. These fieldbus bytes are mapped into PMAC's memory space via the UBUS interface.
- 2. PMAC M-Variables are used to move data to and from the fieldbus or to control the COMX board.
- 3. An E-point jumper on the ACC-72EX sets the address of the board in PMAC memory space.
- 4. The COMX board is configurable via a USB port. SYCON.NET is provided with the COMX board for this purpose.
- 5. Diagnostic LEDs are provided for a visual indication of board status.

Turbo PMAC Memory

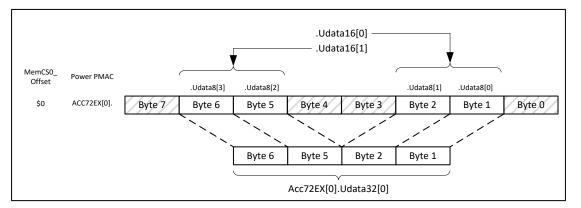
Turbo PMAC uses a DSP (Digital Signal Processor) with a 24-Bit architecture that uses two memory areas: Y and X Memory. Memory is accessed in PMAC programs using M-Variables. The definition of an M-Variable includes its number, address, offset, width, and type. Refer to the Turbo PMAC Software Reference Manual or Turbo PMAC User Manual for additional explanation of M-Variables and their specification, such as in the "M-Variables" section in the User manual.



Turbo PMAC Memory Organization

Power PMAC Memory

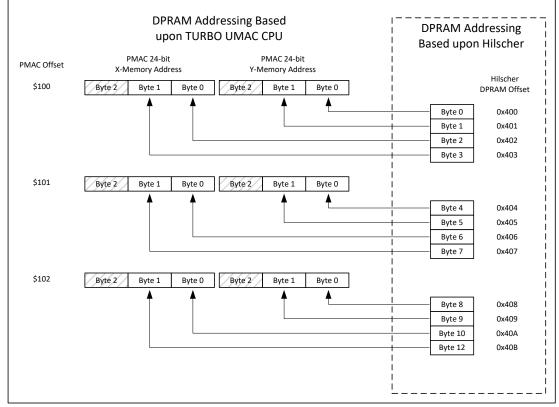
Power PMAC uses 32-bit data bus architecture. ACC-72EX Memory is accessed in Power PMAC data structures or their equivalent **#define** statements. The **#define** statements are included later in this manual.



Power PMAC Memory Organization

Hilscher ComX Module Addressing to Turbo PMAC Addressing Conversion

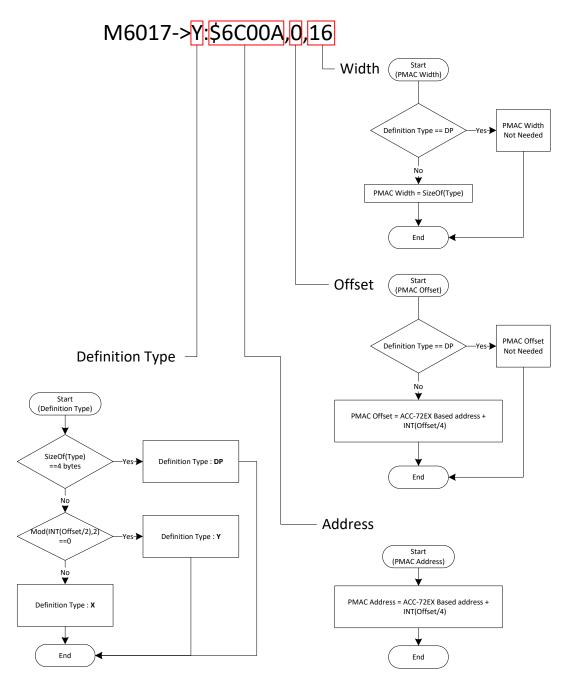
As explained in previous sections, Turbo PMAC places 4 bytes of Hilscher ComX memory data in each PMAC memory word. This means that for each address offset increment on the PMAC side, there will be 4 increments of offset addresses on the Hilscher DPRAM side. The following example shows PMAC addressing for equivalent offset addresses of 0x400 in Hilscher documentation.



Consumed Data Flow

In general, the following flowcharts can be used to convert any Hilscher DPRAM addressing to PMAC's addressing format:

System Information Block							
Offset	Туре	Name	Description				
0x0028	UINT16	usDeviceClass	Device Class netX Device Class (see page 34)				



An address conversion tool is provided in the ACC-72EX Setup Assistant software.

Hilscher ComX Module Addressing to Power PMAC Addressing Conversion

In Power PMAC, specific **Acc72EX**[*i*] data structures have been implemented which allow bit-wide, byte-wide, 2-byte and 4-byte access to Hilscher ComX Dual Ported RAM.

Acc72EX[*i*].Udata16[*j*] structures can be used for individual bit access, both for read and for write purpose.

DPRAM Addressing Based upon Power UMAC CPU	DPRAM Addressing Based upon Hilscher
Power PMAC Structures	Hilscher DPRAM Offset
Acc72EX[0].Udata32[256] Acc72EX[0].Udata32[256] Acc72EX[0].Udata16[512] Acc72EX[0].Udata8[1025] Acc72EX[0].Udata8[1026] Acc72EX[0].Udata8[1027] Byte 1 Byte 1 Byte 2 Byte 3 Byte 3	Byte 0 0x400 (1024) Byte 1 0x401 (1025) Byte 2 0x402 (1026) Byte 3 0x403 (1027)
Acc72EX[0].Udata32[257] Acc72EX[0].Udata16[514] Acc72EX[0].Udata8[1028] Byte 4 I Acc72EX[0].Udata32[257] Acc72EX[0].Udata16[515] Acc72EX[0].Udata8[1030] Byte 6 I Acc72EX[0].Udata16[515] Acc72EX[0].Udata8[1031] Byte 7 I	Byte 4 0x404 (1028) Byte 5 0x405 (1029) Byte 6 0x406 (1030) Byte 7 0x407 (1031)
Acc72EX[0].Udata32[258] Acc72EX[0].Udata32[258] Acc72EX[0].Udata16[516] Acc72EX[0].Udata8[1032] Acc72EX[0].Udata8[1033] Byte 9 Acc72EX[0].Udata8[1034] Acc72EX[0].Udata8[1035] Byte 10 Byte 12 Byte	Byte 8 0x408 (1032) Byte 9 0x409 (1033) Byte 10 0x40A (1034) Byte 12 0x40B (1035)

HARDWARE

E3: UBUS Address

E-point jumper E3 on the ACC-72EX controls the base address and range on the UBUS. Since each ACC-72EX uses full-13 bit addressing, it consumes all the memory addressable through each chip select. As a result, two is the maximum number of ACC-72EX boards that can be used in a Turbo UMAC rack.

E3	Turbo PMAC	Power PMAC
1-2	Y/X:\$6C000 - \$6FFFF	ACC-72EX[0] (\$E00000)
2-3	Y/X:\$74000 - \$7FFFF	ACC-72EX[1] (\$F00000)

The default location on Turbo PMAC is Y/X:\$6C000 - \$6FFFF (\$E00000 on Power PMAC).

Note:

Do not set the ACC-72EX to the DPR address range \$6C000-\$6FFFF if the UMAC is equipped with an Acc-54E. Acc-54E is set to this range as default.

CS16- Identification

One of the features of the UBUS is that memory locations, selected by CS16 (Chip Select 16/Active Low), were reserved for board identification information.

- Vendor ID (8 bits)
- Options Present (10 bits)
- Revision Number (4 bits)
- Product ID (14 bits)

This information (36 bits) is accessible directly with I-Variables added in Turbo PMAC Firmware 1.936 or later. A summary of the PMAC Gateway ID information is in the table below.

I39 controls the values reported.

I39=	I4942I4952 reports the following				
0	36 bits (Vendor ID, Options present, Rev Number, Product ID)				
1	8 bits (Vendor ID)				
2	10 bits (Options Present) Reported by PMAC in HEX (\$)				
3	4 bits (Revision Number)				
4	14 bits (Product ID)				
5	19 bits (Card Base Address)				

Identification Information

The vendor ID, part number, and revision numbers are programmed into the ACC-72EX base board. The Option Number is set by jumpers on the board. The settings below are given for reference only. There is no need to change these from the factory settings.

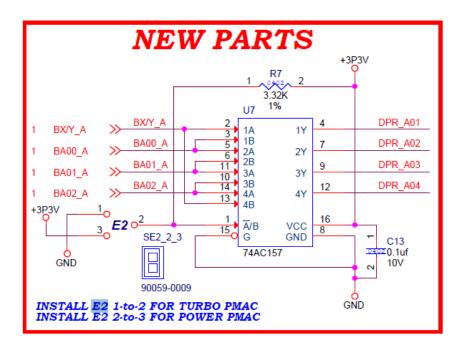
Jumper Settings Option Identification Jumpers

т.		D (N 1	JP1	JP2	JP3	JP4	JP5	JP6	JP7	JP8	JP9
Item	Comm. Protocol Option	Part Number	Bit 9	Bit 10	Bit 11	Bit 12	Bit 13	Bit 14	Bit 15	Bit 16	Bit 17
1	PROFIBUS-DP – Master	310-603958-OPT	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF
2	PROFIBUS-DP – Slave	311-603958-OPT	ON	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF
3	DeviceNet – Master	320-603958-OPT	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF
4	DeviceNet – Slave	321-603958-OPT	ON	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF
5	CANopen – Master	330-603958-OPT	OFF	ON	ON	ON	ON	OFF	OFF	OFF	OFF
6	CANopen – Slave*	331-603958-OPT	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF
7	CC-Link – Slave*	351-603958-OPT	ON	ON	OFF	OFF	ON	ON	OFF	OFF	OFF
8	EtherCAT – Master	360-603958-OPT	OFF	OFF	ON	ON	ON	ON	OFF	OFF	OFF
9	EtherCAT – Slave	361-603958-OPT	ON	OFF	ON	ON	ON	ON	OFF	OFF	OFF
10	EtherNet/IP – Scanner/Master	370-603958-OPT	OFF	ON	ON	OFF	OFF	OFF	ON	OFF	OFF
11	EtherNet/IP – Adaptor/Slave	371-603958-OPT	ON	ON	ON	OFF	OFF	OFF	ON	OFF	OFF
12	Open Modbus/TCP	380-603958-OPT	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF
13	PROFINET IO – Controller	390-603958-OPT	OFF	ON	OFF	ON	ON	OFF	ON	OFF	OFF
14	PROFINET IO – Device	391-603958-OPT	ON	ON	OFF	ON	ON	OFF	ON	OFF	OFF

*No longer available

E-Point Jumper Settings

Point	Default	Description
E1	1-2	Selection of Reset Polarity Signal for Hilscher Module:
		1-2 Selects Low True Reset
		2-3 Selects High True Reset
E2	2-3	Selection of UMAC CPU architecture. This selection affects the data bus and provides
		contiguous data addressing for the DPRAM:
		1-2 Turbo PMAC
		2-3 Power PMAC
E3	1-2	Selection of base address for ACC-72EX:
		1-2 Selects Y/X:\$6C000 - \$6FFFF (Acc72EX[0])
		2-3 Selects Y/X:\$74000 - \$7FFFF (Acc72EX[1])
E5	OFF	Connects DPRAM interrupt to UBUS IRQ-1
E6	OFF	Connects DPRAM interrupt to UBUS IRQ-2
E7	OFF	Connects DPRAM interrupt to UBUS IRQ-3



Communication Option-Dependent E-Point Jumper Settings

Comm. Protocol Option	Option Part Number	E8	E10	E11	E12
PROFIBUS-DP – Master	310-603958-OPT	1-2	OFF	OFF	OFF
PROFIBUS-DP – Slave	311-603958-OPT	1-2	OFF	OFF	OFF
DeviceNet – Master	320-603958-OPT	2-3	ON	OFF	OFF
DeviceNet - Slave	321-603958-OPT	2-3	ON	OFF	OFF
CANopen – Master	330-603958-OPT	OFF	OFF	ON	OFF
CANopen – Slave*	331-603958-OPT	OFF	OFF	ON	OFF
CC-Link – Slave*	351-603958-OPT	2-3	OFF	OFF	ON
EtherCAT – Master	360-603958-OPT	OFF	OFF	OFF	OFF
EtherCAT – Slave	361-603958-OPT	OFF	OFF	OFF	OFF
EtherNet/IP - Scanner/Master	370-603958-OPT	OFF	OFF	OFF	OFF
EtherNet/IP - Adaptor/Slave	371-603958-OPT	OFF	OFF	OFF	OFF
Open Modbus/TCP	380-603958-OPT	OFF	OFF	OFF	OFF
PROFINET IO – Controller	390-603958-OPT	OFF	OFF	OFF	OFF
PROFINET IO – Device	391-603958-OPT	OFF	OFF	OFF	OFF

*No longer available

NOTES:

E8: Determines the signal on pin 5 of the Fieldbus 9-pin D-Sub Connector. The position of the jumper depends on the COMX module installed/option ordered.

E10: Adds 120 $\boldsymbol{\Omega}$ termination resistor for DeviceNet communication lines

E11: Adds 120 Ω termination resistor for CANopen communication lines

E12: Adds 110 Ω termination resistor for CC-Link communication lines

Connector Pinouts

Fieldbus Port (J4)

Protocol Pin No.	PROFIBUS	DeviceNet	CANopen	CC-Link
1		+24 V Power Supply		CC-Link, Shield
2	Positive power supply	CAN High-Signal	CAN_L Bus Line	CC-Link, Function Ground
3	Receive / Send Data-P	Reference potential	CAN Ground	CC-Link, Data A
4	Control			
5	Reference potential	Shield		CC-Link, Data Ground
6	Positive power supply	CAN High-Signal	CAN_L Bus Line	CC-Link, Function Ground
7			CAN_H Bus Line	
8	Receive / Send Data-N			
9		CAN Low-Signal		CC-Link, Data B
NOTES	E8, Jumpered 1-2	E8, 2-3 Jumpered E10 Jumpered	E11 Jumpered	E8, Jumpered 2-3 E12 Jumpred

Real-time Ethernet Ports (Ethernet 0 & Ethernet 1)

Pin No.	Symbol	Description
1	RX+	Receive+
2	RX-	Receive-
3	TX+	Transmit+
4		
5		
6	TX-	Transmit–
7		
8		

Diagnostics Port (Micro A USB)

Pin No.	Symbol	Description
1	VBUS	+5 VDC (Not connected to ACC-72EX +5 VDC)
2	D-	Data -
3	D+	Data +
4	GND	Ground Reference (Connected to ACC-72EX and UMAC's Digital Ground)

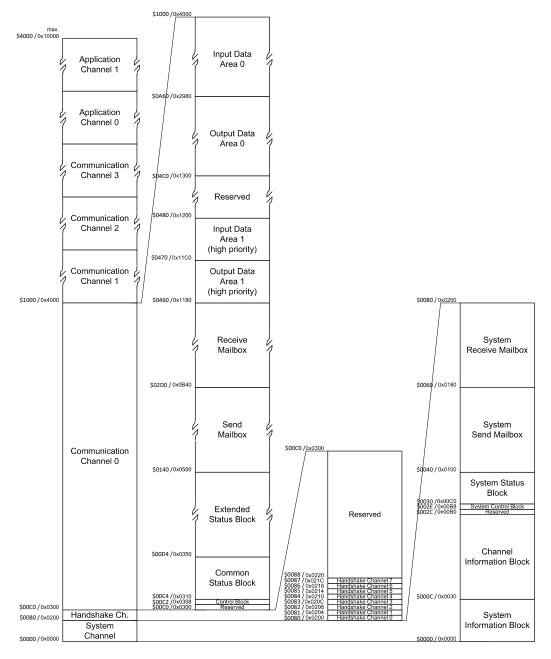
Note:

The USB connection is not a galvanically isolated connection. The ground of the PC will be connected to the ground of the UMAC system through the USB connection, which can damage components on the PC and/or ACC-72EX. Make sure that there is no potential difference between the grounds on both ends.

DPRAM MEMORY MAP

Below is the standard memory map of address offsets found in the DPRAM of the ACC-72EX module. Start and end addresses for each of the memory blocks are specified both in Hilscher offset (0x notation for hexadecimal) and Turbo PMAC offset (\$ notation for hexadecimal) notation. The memory map shown here is the standard memory map. Different COMX modules may have different memory maps. Please refer to the Hilscher manual for each COMX module for detailed information.

These registers should be read from and written to using M-Variables which point to the lower 16 bits of the X/Y-memory with an offset from the base address that is configured with E3. Handshake and System registers are common between all protocols, and others can be auto-generated using the "ACC-72EX Setup Assistant" software.



DPRAM Blocks

In the Hilscher COMX module DPRAM, the system channel and the handshake channel are always present. These channels are used for communicating with the firmware on the COMX module, from this point on referred to as "netX."

The system channel provides information about the state of the COMX module operating system, netX, and the structure of the dual-port memory. It allows basic communication via system mailboxes.

The handshake channel provides a bit toggle mechanism that insures synchronizing data transfer between the UMAC and COMX module. All handshake cells from system, communication, and application channels are brought together in this one location.

Next are the communication and application channels. A communication channel provides network access and occupies an area of the COMX dual-port memory with process, non-cyclic, and diagnostic data. An application channel can be used for any functionality that may be executed in the context of the netX operating system. The application channels are not supported by COMX modules at the time of writing this manual.

DPRAM Suggested Macro Names

ACC-72EX Setup Assistant software, available through the Tools menu in PEWIN32PRO2 software, provides a complete overview of blocks and sub-blocks available to each ACC-72EX. Under each section is a list of macro names provided for M-Variable definition.

For more information on structures and data registers in COMX modules, please refer to references introduced in appendix A of this manual.

System Channel

The System Channel is the first of the channels in the dual-port memory and starts at address offset \$0000. It holds information about the system itself (netX, netX operating system) and provides a mailbox transfer mechanism for system-related messages or packets.

ACC-72EX Setup Assistant software uses the data available in this channel to generate the information in the memory map output file.

System Information Block

The first block of information allows identification of the netX dual memory; it is used for testing proper communication. The first 4 registers hold character values for "netX" (110, 101, 116, 88). If these values are reading properly, the DPRAM communication is in working condition.

	Hilscher Documentation	ACC-72EX Setup Assistant
	abCookie[4]	SI_abCookie_0 SI_abCookie_3_
	ulDpmTotalSize	SI_ulDpmTotalSize
	ulDeviceNumber	SI_ulDeviceNumber
×	ulSerialNumber	SI_ulSerialNumber
Block	ausHwOptions[4]	SI_ausHwOptions_0 SI_ausHwOptions_3_
on E	usManufacturer	SI_usManufacturer
System Information	usProductionDate	SI_usProductionDate
r m	ulLicenseFlags1	SI_ulLicenseFlags1
nfc	ulLicenseFlags2	SI_ulLicenseFlags2
Ē	usNetxLicenseID	SI_usNetxLicenseID
/ste	usNetxLicenseFlags	SI_usNetxLicenseFlags
S	usDeviceClass	SI_usDeviceClass
	bHwRevision	SI_bHwRevision
	bHwCompatibility	SI_bHwCompatibility
	bDevIdNumber	SI_bDevIdNumber

Channel Information Block

The system block includes information about all the other channels and their availability on the COMX module. This information is used to locate and identify different channels in the system.

	Hilscher Documentation	ACC-72EX Setup Assistant
stem Channel Information	bChannelType	SCI_bChannelType
	bSizePositionOfHandshake	SCI_bSizePositionOfHandshake
	bNumberOfBlocks	SCI_bNumberOfBlocks
	ulSizeOfChannel	SCI_ulSizeOfChannel
	usSizeOfMailbox	SCI_usSizeOfMailbox
S	usMailboxStartOffset	SCI_usMailboxStartOffset

	Hilscher Documentation	ACC-72EX Setup Assistant
n i	bChannelType	HCI_bChannelType
Handshake Channel Info	ulSizeOfChannel	HCI_ulSizeOfChannel

	Hilscher Documentation	ACC-72EX Setup Assistant	
lər	bChannelType	CCxI_bChannelType	
ine	bChannelld	CCxI_bChannelId	
tion Ch	bSizePositionOfHandshake	CCxI_bSizePositionOfHandshake	
cation	bNumberOfBlocks	CCxI_bNumberOfBlocks	
unicat nform	ulSizeOfChannel	CCxI_ulSizeOfChannel	
Inf	usCommunicationClass	CCxI_usCommunicationClass	
μμ	usProtocolClass	CCxI_usProtocolClass	
Col	usConformanceClass	CCxI_usConformanceClass	
Note: x in	Note: x in MACRO name is replaced by Application Channel number 0 3		

	Hilscher Documentation	ACC-72EX Setup Assistant
c i	bChannelType	ACxI_bChannelType
ation el Info	bChannelld	ACxI_bChannelId
00	bSizePositionOfHandshake	ACxI_bSizePositionOfHandshake
Appli Chann	bNumberOfBlocks	ACxI_bNumberOfBlocks
C ≯	ulSizeOfChannel	ACxI_ulSizeOfChannel
Note: x in MACRO name is replaced by Application Channel number 0 1		

System Control Block

The system control block is used by UMAC to force netX to execute certain commands in the future. Currently, there are no such commands defined.

	Hilscher Documentation	ACC-72EX Setup Assistant
System Control Block	ulSystemCommandCOS	SCtrl_ulSystemCommandCOS

System Status Block

The system status block provides information about the staus of the netX firmware.

	Hilscher Documentation	ACC-72EX Setup Assistant
×	ulSystemCOS	SStat_ulSystemCOS
Block	ulSystemStatus	SStat_ulSystemStatus
sn	ulSystemError	SStat_ulSystemError
Stat	ulBootError	SStat_ulBootError
E	ulTimeSinceStart	SStat_ulTimeSinceStart
ste	usCpuLoad	SStat_usCpuLoad
Sy	ulHWFeatures	SStat_ulHWFeatures

System Mailbox

The system mailbox is the "window" to the operating system. It is always present even if no firmware is loaded. For more information about using system send/receive mailboxes, please see the examples shown in the following chapters. A complete list of functions, which can be accessed using the mailboxes, can be found in the netX Dual-Ported Memory Interface document available from Hilscher.

	Hilscher Documentation	ACC-72EX Setup Assistant						
	usPackagesAccepted	SSMB_usPackagesAccepted						
×	ulDest	SSMB_ulDest						
Send Mailbox	ulSrc	SSMB_ulSrc						
٨ai	ulDestId	SSMB_ulDestId						
2 p	ulSrcId	SSMB_ulSrcId						
Ser	ulLen	SSMB_ulLen						
Block	ulld	SSMB_ulld						
Blo	ulState	SSMB_ulState						
System	ulCmd	SSMB_ulCmd						
yst	ulExt	SSMB_ulExt						
S	ulRout	SSMB_ulRout						
		SSMB_ultData0 SSMB_ultData20						

	Hilscher Documentation	ACC-72EX Setup Assistant					
	usWaitingPackages	SRMB_usWaitingPackages					
ŏ	ulDest	SRMB_ulDest					
Receive Mailbox	ulSrc	SRMB_ulSrc					
Σ	ulDestId	SRMB_ulDestId					
eive	ulSrcId	SRMB_ulSrcId					
le Cé	ulLen	SRMB_ulLen					
	ulld	SRMB_ulld					
System Block	ulState	SRMB_ulState					
л В В	ulCmd	SRMB_ulCmd					
ster	ulExt	SRMB_ulExt					
Sys	ulRout	SRMB_ulRout					
		SRMB_ultData0 SRMB_ultData20					

Handshake Channel

The handshake channel provides a mechanism that allows the synchronizing of data transfer between the UMAC CPU and ACC-72EX dual-port memory. The handshake channel brings all handshake registers from other channel blocks together in one location. The handshake register could be moved from the handshake block to the beginning of each of the communication channels.

There are three types of handshake cells, described below.

System Handshake Cells

System handshake flags are used to synchronize data transfer between the ACC-72EX Hilscher Module and UMAC via the system mailbox and to handle certain changes of state function. They also hold information about the status of the ACC-72EX Hilscher module and can be used to execute certain commands in the module (for a module-wide reset, for example).

There are two sets of system flags. One set is dedicated for netX writes and is read by UMAC, and the other one is designated for UMAC writes. netX is continuously reading the second set.

netX System Flags

The ACC-72EX Hilscher module firmware writes to the netX system register; UMAC reads this register. The netX system register is located at address offset \$80 in the dual-port memory.

bNe	txFlag	s – ne	etX wr	ites, I	JMAC	read	S									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function					Rese	rved					NSF_RECV_MBX_CMD	NSF_SEND_MBX_ACK	NSF_NETX_COS_CMD	NSF_HOST_COS_ACK	NSF_ERROR	NSF_READY

netX System Flags **bNetxFlags** (ACC-72EX ⇒ UMAC)

Bit No.	Definition / Description
0	Ready (NSF_READY)
	The Ready flag is set as soon as the COMX has initialized itself properly and passed its
	self-test. When the flag is set, the netX is ready to accept packets via the system mailbox.
	If cleared, the netX does not accept any packages.
	Error (NSF_ERROR)
	The Error flag is set when the netX has detected an internal error condition. This is
1	considered to be a fatal error. The Ready flag is cleared and the netX operating system is
	stopped. An error code helping to identify the issue is placed in the ulSystemError
	variable in the system status block.
	Host Change Of State Acknowledge (NSF_HOST_COS_ACK)
2	The Host Change of State Acknowledge flag is set when the netX acknowledges a
2	command from the host system. This flag is used together with the Host Change of State
	Command flag in the host system flags.
	netX Change Of State Command (NSF_NETX_COS_CMD)
3	The netX Change of State Command flag is set if the netX signals a change of its state to
5	the host system. Details of what has changed can be found in the ulSystemCOS variable
	in the system control block.
	Send Mailbox Acknowledge (NSF_SEND_MBX_ACK)
4	Both the Send Mailbox Acknowledge flag and the Send Mailbox Command flag are used
	together to transfer non-cyclic packages between the UMAC and the netX.
	Receive Mailbox Command (NSF_RECV_MBX_CMD)
5	Both the Receive Mailbox Command flag and the Receive Mailbox Acknowledge flag are
	used together to transfer non-cyclic packages between the netX and UMAC.
6, 7 15	6, 7 15 Reserved, set to zero

	Hilscher Documentation	ACC-72EX Setup Assistant
s G	bNetxFlags	HCSC_bNetxFlags
shake Flags	NSF_READY	HCSC_NSF_READY
nds m F	NSF_ERROR	HCSC_NSF_ERROR
n Hand: egister vystem	NSF_HOST_COS_ACK	HCSC_NSF_HOST_COS_ACK
	NSF_NETX_COS_CMD	HCSC_NSF_NETX_COS_CMD
Syster I netX	NSF_SEND_MBX_ACK	HCSC_NSF_SEND_MBX_ACK
N N	NSF_RECV_MBX_CMD	HCSC_NSF_RECV_MBX_CMD

Host System Flags

The host system flags are written by UMAC; the netX reads these flags. The host system register is located at address offset \$81 in the dual-port memory.

bHc	stFlag	stFlags – UMAC writes, netX reads														
Bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1										0					
Function					Rese	rved					HSF_RECV_MBX_ACK	HSF_SEND_MBX_CMD	HSF_NETX_COS_ACK	HSF_HOST_COS_CMD	HSF_BOOTSTART	HSF_RESET

Host System Flags **bHostFlags** (UMAC ⇒ ACC-72EX)

Bit No.	Definition / Description
	Reset (HSF_RESET)
0	The Reset flag is set by the UMAC to execute a system wide reset. This forces the system
	to restart. All network connections are interrupted immediately regardless of their current
	state.
	Bootstart (HSF_BOOTSTART)
1	If set during reset, the Boot-Start flag forces the netX to stay in boot loader mode; a
1	firmware that may reside in the context of the operating system rcX is not started. If
	cleared during reset, the operating system will start the firmware if available.
	Host Change Of State Command (HSF_HOST_COS_CMD)
2	The Host Change of State Command flag is set by the UMAC to signal a change of its state
2	to the netX. Details of what has changed can be found in the ulSystemCommandCOS
	variable in the system control block.
	netX Change Of State Acknowledge (HSF_NETX_COS_ACK)
3	The netX Change of State Acknowledge flag is set by the UMAC to acknowledge the new
5	state of the netX. This flag is used together with the netX Change of State Command flag
	in the netX system flags.
	Send Mailbox Command (HSF_SEND_MBX_CMD)
4	Both the Send Mailbox Command flag and the Send Mailbox Acknowledge flag are used
	together to transfer non-cyclic packages between the UMAC and the netX.
	Receive Mailbox Acknowledge (HSF_RECV_MBX_ACK)
5	Both the Receive Mailbox Acknowledge flag and the Receive Mailbox Command flag are
	used together to transfer non-cyclic packages between the netX and the UMAC.
6, 7	6, 7 15 Reserved; set to zero
15	

	Hilscher Documentation	ACC-72EX Setup Assistant
em e	bHostFlags	HCSC_bHostFlags
hake /sten	HSF_RESET	HCSC_HSF_RESET
Handshá Host Sys ⁱ lags	HSF_BOOTSTART	HCSC_HSF_BOOTSTART
Han Host Iags	HSF_HOST_COS_CMD	HCSC_HSF_HOST_COS_CMD
	HSF_NETX_COS_ACK	HCSC_HSF_NETX_COS_ACK
System Register	HSF_SEND_MBX_CMD	HCSC_HSF_SEND_MBX_CMD
Sy Reg	HSF_RECV_MBX_ACK	HCSC_HSF_RECV_MBX_ACK

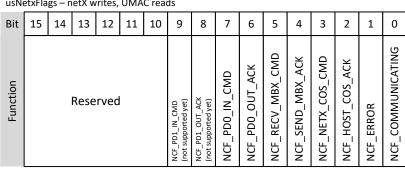
Communication Channel Handshake Cells

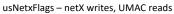
The channel handshake register is used to indicate the status of the protocol stack as well as execute certain commands in the protocol stack (e.g. reset a channel or synchronization of process data). The mailbox flags are used to send and receive non-cyclic messages via the channel mailboxes.

There are two sets of Communication Channel flags. One set is dedicated for netX writes; UMAC continually reads this. The other set is designated for UMAC writes; netX continuously reads this.

netX Communication Flags

This flag register is organized as a bit field. The netX protocol stack writes to the register to control data synchronization via the mailbox system and the process data image. It also informs the UMAC about its current network state. The UMAC reads this register.





Communication Channel Flags usNetXFlags (ACC-72EX ⇒ UMAC)

Bit	Definition / Description
No.	
	Communicating (NCF_COMMUNICATING)
	The NCF_COMMUNICATING flag is set if the protocol stack has successfully opened a
0	connection to at least one of the configured network slaves (for master protocol stacks),
0	respectively has an open connection to the network master (for slave protocol stacks). If
	cleared, the input data should not be evaluated, because it may be invalid, old or both. At
	initialization time, this flag is cleared.
	Error (NCF_ERROR)
	The NCF_ERROR flag signals an error condition that is reported by the protocol stack. It
1	could indicate a network communication issue or something to that effect. The corresponding
	error code is placed in the ulCommunicationError variable in the common status block. At
	initialization time, this flag is cleared.
_	Host Change Of State Acknowledge (NCF_HOST_COS_ACK)
2	The NCF_HOST_COS_ACK flag is used by the protocol stack indicating that the new state
	of the UMAC has been read. At initialization time, this flag is cleared.
	netX Change Of State Command (NCF_NETX_COS_CMD)
	The NCF_NETX_COS_CMD flag signals a change in the state of the protocol stack. The new
3	state can be found in the ulCommunicationCOS register in the common status block. In return
_	the UMAC program then toggles the HCF_NETX_COS_ACK flag in the host
	communication flags acknowledging that the new protocol state has been read. At
	initialization time, this flag is cleared.
	Send Mailbox Acknowledge (NCF_SEND_MBX_ACK)
4	Both the NCF_SEND_MBX_ACK flag and the HCF_SEND_MBX_CMD flag are used
	together to transfer non-cyclic packages between the protocol stack and the UMAC programs.
	At initialization time, this flag is cleared. Receive Mailbox Command (NCF_RECV_MBX_CMD)
	Both the NCF_RECV_MBX_CMD flag and the HCF_RECV_MBX_ACK flag are used
5	together to transfer non-cyclic packages between the UMAC programs and the protocol stack.
	At initialization time, this flag is cleared.
	Process Data 0 Out Acknowledge (NCF_PD0_OUT_ACK)
	Both the NCF_PD0_OUT_ACK flag and the HCF_PD0_OUT_CMD flag are used together to
6	transfer cyclic output data from the UMAC to the protocol stack. At initialization time, this
	flag may be set, depending on the data exchanged mode.
	Process Data 0 In Command (NCF_PD0_IN_CMD)
_	Both the NCF_PD0_IN_CMD flag and the HCF_PD0_IN_ACK flag are used together to
7	transfer cyclic input data from the protocol stack to the UMAC. At initialization time, this
	flag may be set, depending on the data exchanged mode.
	Process Data 1 Out Acknowledge (NCF_PD1_OUT_ACK, not supported yet)
0	Both the NCF_PD1_OUT_ACK flag and the HCF_PD1_OUT_CMD flag are used together to
8	transfer output cyclic data from the UMAC to the protocol stack. At initialization time, this
	flag may be set, depending on the data exchanged mode.
	Process Data 1 In Command (NCF_PD1_IN_CMD, not supported yet)
9	Both the NCF_PD1_IN_CMD flag and the HCF_PD1_IN_ACK flag are used together to
7	transfer cyclic input data from the protocol stack to the UMAC. At initialization time, this
	flag may be set, depending on the data exchange mode.
1015	Reserved, set to 0

	Hilscher Documentation	ACC-72EX Setup Assistant						
	usNetxFlags	HCCCx_usNetxFlags						
gs	NCF_COMMUNICATING	HCCCx_NCF_COMMUNICATING						
Flags er	NCF_ERROR	HCCCx_NCF_ERROR						
cation Fla Register	NCF_HOST_COS_ACK	HCCCx_NCF_HOST_COS_ACK						
unication ke Regist	NCF_NETX_COS_CMD	HCCCx_NCF_NETX_COS_CMD						
uni ake	NCF_SEND_MBX_ACK	HCCCx_NCF_SEND_MBX_ACK						
X Communi Handshake	NCF_RECV_MBX_CMD	HCCCx_NCF_RECV_MBX_CMD						
and	NCF_PD0_OUT_ACK	HCCCx_NCF_PDx_OUT_ACK						
Ha	NCF_PD0_IN_CMD	HCCCx_NCF_PDx_IN_CMD						
ne	NCF_PD1_OUT_ACK	HCCCx_NCF_PD1_OUT_ACK						
	NCF_PD1_IN_CMD	HCCCx_NCF_PD1_IN_CMD						
Note: x i	in MACRO name is replaced by Communication C	hannel number 0 3						

Host Communication Flags

This flag register is organized as a bit field. UMAC writes to this register to control data synchronization via the mailbox system and the process data image. The netX protocol stack reads this register.

usno	shostriags – UNAC writes, netz reads															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function			Rese	rved	I		HCF_PD1_IN_ACK (not supported yet)	HCF_PD1_OUT_CMD (not supported yet)	HCF_PD0_IN_ACK	HCF_PD0_OUT_CMD	HCF_RECV_MBX_ACK	HCF_SEND_MBX_CMD	HCF_NETX_COS_ACK	HCF_HOST_COS_CMD		

usHostFlags – UMAC writes, netX reads

Communication Channel Flags usHostFlags (UMAC ⇒ ACC-72EX)

Bit No.	Definition / Description
0, 1	Reserved, set to 0
2	Host Change Of State Command (HCF_HOST_COS_CMD) The HCF_HOST_COS_CMD flag signals a change in the state of the UMAC. A new state is set in the ulApplicationCOS variable in the communication control block. The protocol stack on the netX then toggles the NCF_HOST_COS_ACK flag in the netX communication flags back acknowledging that the new state has been read. At initialization time, this flag is cleared.
3	Host Change Of State Acknowledge (HCF_NETX_COS_ACK) The HCF_NETX_COS_ACK flag is used by UMAC to indicate that the new state of the protocol stack has been read. At initialization time, this flag is cleared.
4	Send Mailbox Command (HCF_SEND_MBX_CMD) Both the HCF_SEND_MBX_CMD flag and the NCF_SEND_MBX_ACK flag are used together to transfer non-cyclic packages between the UMAC and the protocol stack. At initialization time, this flag is cleared.
5	Receive Mailbox Acknowledge (HCF_RECV_MBX_ACK) Both the HCF_RECV_MBX_ACK flag and the NCF_RECV_MBX_CMD flag are used together to transfer non-cyclic packages between the protocol stack and the UMAC. At initialization time, this flag is cleared.
6	Process Data 0 Out Command (HCF_PD0_OUT_CMD) Both the HCF_PD0_OUT_CMD flag and the NCF_PD0_OUT_ACK flag are used together to transfer cyclic output data from the UMAC to the protocol stack. At initialization time, this flag may be set, depending on the data exchanged mode.
7	Process Data 0 In Acknowledge (HCF_PD0_IN_ACK) Both the HCF_PD0_IN_ACK flag and the NCF_PD0_IN_CMD flag are used together to transfer cyclic input data from the protocol stack to the UMAC. At initialization time, this flag may be set, depending on the data exchanged mode.
8	Process Data 1 Out Command (HCF_PD1_OUT_CMD, not supported yet) Both the HCF_PD1_OUT_CMD flag and the NCF_PD1_OUT_ACK flag are used together to transfer cyclic output data from the UMAC to the protocol stack. At initialization time, this flag may be set, depending on the data exchanged mode.
9	Process Data 1 In Acknowledge (HCF_PD1_IN_ACK, not supported yet) Both the HCF_PD1_IN_ACK flag and the NCF_PD1_IN_CMD flag are used together to transfer cyclic input data from the protocol stack to the UMAC. At initialization time, this flag may be set, depending on the data exchanged mode.
10 15	Reserved, set to 0

	Hilscher Documentation	ACC-72EX Setup Assistant				
gs	usHostFlags	HCCCx_usHostFlags				
Flags er	HCF_HOST_COS_CMD	HCCCx_HCF_HOST_COS_CMD				
ion gist	HCF_NETX_COS_ACK	HCCCx_HCF_NETX_COS_ACK				
unication Fl	HCF_SEND_MBX_CMD	HCCCx_HCF_SEND_MBX_CMD				
uni Ike	HCF_RECV_MBX_ACK	HCCCx_HCF_RECV_MBX_ACK				
Host Commun Handshake	HCF_PD0_OUT_CMD	HCCCx_HCF_PDx_OUT_CMD				
Con	HCF_PD0_IN_ACK	HCCCx_HCF_PDx_IN_ACK				
H	HCF_PD1_OUT_CMD	HCCCx_HCF_PD1_OUT_CMD				
Я	HCF_PD1_IN_ACK	HCCCx_HCF_PD1_IN_ACK				
Note: x i	n MACRO name is replaced by Communication C	hannel number 0 3				

Application Handshake Cells

Although these cells are not supported yet, the following structure groups have been defined for backward compatibility as a placeholder:

netX Communication Flags

Host Communication Flags

Communication Channel

The communication channel structure is mainly dependent on the protocol firmware and COMX module. However, there are common sub-block structures which are common to all protocols.

Control Block

The control block of a dual-port memory features a Watchdog function to allow the operating system running on the netX to supervise the host application and vice versa. The control area is always present in dual-port memory. This block can also be read using the mailbox interface.

Application Change of State Register

The Application Change of State Register is a bit field. The UMAC uses this field to send commands to the communication channel. Changing flags in this register requires the UMAC to toggle the Host Change of State Command flag in the Host Communication Flags register, and then the netX protocol stack will recognize the change.

ulAp	pplicationCOS – UMAC writes, netX reads															
Bit	31 30 29 11 10 9					9	8	7	6	5	4	3	2	1	0	
Function			Re	serv	ed			RCX_APP_COS_DMA_ENABLE	RCX_APP_COS_DMA	RCX_APP_COS_LOCK_CONFIG_ENABLE	RCX_APP_COS_LOCK_CONFIG	RCX_APP_COS_INIT_ENABLE	RCX_APP_COS_INIT	RCX_APP_COS_BUS_ON_ENABLE	RCX_APP_COS_BUS_ON	RCX_APP_COS_APP_READY

Bit	Definition / Description
No.	
0	Application Ready (RCX_APP_COS_APP_READY, not supported yet)
0	If set, the UMAC indicates to the protocol stack that its state is Ready.
	Bus On (RCX_APP_COS_BUS_ON)
	Using the Bus On flag, the UMAC allows or disallows the firmware to open network
1	connections. This flag is used with Bus On Enable flag below. If set, the netX firmware tries
	to open network connections; if cleared, no connections are allowed, and open connections are
	closed.
	Bus On Enable (RCX_APP_COS_BUS_ON_ENABLE)
2	The Bus On Enable flag is used together with the Bus On flag above. If set, this flag enables
	the execution of the Bus On command in the netX firmware.
	Initialization (RCX_APP_COS_INIT)
	Setting the Initialization flag the UMAC forces the protocol stack to restart and evaluate the
3	configuration parameter again. All network connections are interrupted immediately
	regardless of their current state. If the database is locked, re-initializing the channel is not
	allowed.
	Initialization Enable (RCX_APP_COS_INIT_ENABLE)
4	The Initialization Enable flag is used together with the Initialization flag above. If set, this flag
	enables the execution of the Initialization command in the netX firmware.
	Lock Configuration (RCX_APP_COS_LOCK_CONFIG)
5	If set, UMAC does not allow the firmware to reconfigure the communication channel. The
5	database will be locked. The Configuration Locked flag in the channel status block shows if
	the current database has been locked.

Bit No.	Definition / Description
	Lock Configuration Enable (RCX_APP_COS_LOCK_CONFIG_ENABLE)
6	The Lock Configuration Enable flag is used together with the Lock Configuration flag
0	above. If set, this flag enables the execution of the Lock Configuration command in the netX
	firmware.
	Turn on DMA Mode (RCX_APP_COS_DMA)
7	The UMAC sets this flag in order to turn on the DMA mode for the cyclic process data input
	/ output image 0 (abPd0Output and abPd0Input).
	Turn on DMA Mode Enable (RCX_APP_COS_DMA_ENABLE)
8	The DMA Enable flag is used together with the DMA flag above. If set, this flag enables the
	execution of the DMA command in the netX firmware.
9 31	Reserved, set to 0

Device Watchdog Register

The protocol stack supervises the UMAC using a Watchdog function. If the UMAC fails to copy the value from the host Watchdog location to the device Watchdog location, the protocol stack assumes that the UMAC system has a problem and interrupts all network connections immediately, regardless of their current state.

	Hilscher Documentation	ACC-72EX Setup Assistant					
ck	RCX_APP_COS_APP_READY	CCx_RCX_APP_COS_APP_READY					
Block	RCX_APP_COS_BUS_ON	CCx_RCX_APP_COS_BUS_ON					
rol	RCX_APP_COS_BUS_ON_ENABLE	CCx_RCX_APP_COS_BUS_ON_ENABLE					
onti	RCX_APP_COS_INIT	CCx_RCX_APP_COS_INIT					
Communication Control	RCX_APP_COS_INIT_ENABLE	CCx_RCX_APP_COS_INIT_ENABLE					
tio	RCX_APP_COS_LOCK_CFG	CCx_RCX_APP_COS_LOCK_CFG					
nica	RCX_APP_COS_LOCK_CFG_ENA	CCx_RCX_APP_COS_LOCK_CFG_ENA					
nur	RCX_APP_COS_DMA	CCx_RCX_APP_COS_DMA					
um	RCX_APP_COS_DMA_ENABLE	CCx_RCX_APP_COS_DMA_ENABLE					
Ö ulDeviceWatchdog CCx_ulDeviceWatchdog							
Note: x i	Note: x in MACRO name is replaced by Application Channel number 0 3						

Common Status Block

The common status block contains information fields that are common to all protocol stacks. The status block is always present in dual-port memory. This block can also be read using the mailbox interface.

Communication Change of State Register

The Communication Change of State register is a bit field. It contains information about the current operating status of the communication channel and its firmware. Every time the status changes, the netX protocol stack toggles the netX Change of State Command flag in the netX communication flags register. The UMAC then has to toggle the netX Change of State Acknowledge flag back, acknowledging the new state.

ulCo	CommunicationCOS - netX writes, UMAC reads																
Bit	31 30 29 11 10 9 8						8	7	6	5	4	3	2	1	0		
Function				Rese	rved				RCX_COMM_COS_DMA	R CX_COMM_COS_RESTART_REQUIRED_ENABLE	RCX_COMM_COS_RESTART_REQUIRED	RCX_COMM_COS_CONFIG_NEW	RCX_COMM_COS_CONFIG_LOCKED	RCX_COMM_COS_BUS_ON	RCX_COMM_COS_RUN	RCX_COMM_COS_READY	

Bit No.	Definition / Description
	Ready (RCX_COMM_COS_READY)
0	The Ready flag is set as soon as the protocol stack is started properly. Then, the protocol
0	stack awaits a configuration. As soon as the protocol stack is configured properly, the
	Running flag is set.
	Running (RCX_COMM_COS_RUN)
1	The Running flag is set when the protocol stack has been configured properly. Then the
1	protocol stack awaits a network connection. Now, both the Ready flag and the Running flag
	are set.
	Bus On (RCX_COMM_COS_BUS_ON)
	The Bus On flag is set to indicate to the UMAC whether or not the protocol stack has the
2	permission to open network connections. If set, the protocol stack has the permission to
	communicate on the network; if cleared, the permission was denied and the protocol stack
	will not open network connections.
	Configuration Locked (RCX_COMM_COS_CONFIG_LOCKED)
_	The Configuration Locked flag is set if the communication channel firmware has locked the
3	configuration database against being overwritten. Reinitializing the channel is not allowed
	in this state. To unlock the database, the application has to clear the Lock Configuration flag
	in the control block.
	Configuration New (RCX_COMM_COS_CONFIG_NEW)
4	The Configuration New flag is set by the protocol stack to indicate that a new configuration
	became available, but has not yet been activated. This flag may be set together with the
	Restart Required flag. Restart Required (RCX_COMM_COS_RESTART_REQUIRED)
	The Restart Required flag is set when the channel firmware requests to be restarted. This
5	flag is used together with the Restart Required Enable flag below. Restarting the channel
5	firmware may become necessary if a new configuration was downloaded from the UMAC
	or if a configuration upload via the network took place.
	Restart Required Enable (RCX_COMM_COS_RESTART_REQUIRED_ENABLE)
6	The Restart Required Enable flag is used together with the Restart Required flag above. If
0	set, this flag enables the execution of the Restart Required command in the netX firmware.
	DMA Mode On (RCX_COMM_COS_DMA)
7	The protocol stack sets this flag in order to signal to the UMAC that the DMA mode is
-	turned on.
8 31	Reserved, set to 0

Communication State

The communication state field contains current device network communication status information. Depending on the implementation, all or a subset of the definitions below is supported:

Value	Definition / Description
\$0	UNKNOWN
\$1	OFFLINE
\$2	STOP
\$3	IDLE
\$4	OPERATE

Communication Channel Error

This field holds the current error code of the communication channel. If the cause of error is resolved, the communication error field is set to zero (= RCX_S_OK) again. Not all of the error codes are supported in every implementation.

Watchdog Timeout

This field holds the configured Watchdog timeout value in milliseconds. The UMAC may set its Watchdog trigger interval accordingly. If the UMAC fails to copy the value from the host Watchdog location to the device Watchdog location, the protocol stack will interrupt all network connections immediately, regardless of their current state.

Handshake Mode

The protocol stack supports different handshake mechanisms to synchronize process data exchange with the UMAC. Depending on the configured mode, this mechanism insures data consistency over the entire data image and helps synchronize the UMAC with the network. This register holds the configured handshake mode.

Value	Definition / Description
\$0	For compatibility reasons, this value is identical to 0x04 - Buffered Host Controlled IO
\$U	Data Transfer
\$2	Buffered Device-Controlled I/O Data Transfer
\$3	Uncontrolled Mode
\$4	Buffered Host-Controlled IO Data Transfer

Host Watchdog

The protocol stack supervises the UMAC via the Watchdog function. If the UMAC fails to copy the value from the device Watchdog location to the host Watchdog location, the protocol stack assumes that the UMAC has a problem and shuts down all network connections.

Error Count (All Implementations)

This field holds the total number of errors detected since power-up or after a reset. The protocol stack counts all sorts of errors in this field regardless if they were network-related or caused internally. The counter is cleared after a power cycle, reset, or channel initialization.

Error Log Indicator (All Implementations)

Not supported yet; the error log indicator field holds the number of entries in the internal error log. The field is set to zero if all entries are read from the log.

Number of Input Process Data Handshake Errors TBD

Number of Output Process Data Handshake Errors TBD

Number of Synchronization Handshake Errors

This counter will be incremented if the device detects a "not handled synchronization indication." This field is not supported yet.

Synchronization Status

This field is reserved for future use.

Slave State

The Slave State field indicates whether or not the master is in cyclic data exchange to all configured slaves. If there is at least one slave missing or if the slave has a diagnostic request pending, the status

Value	Definition / Description
\$0	UNDEFINED
\$1	OK. No Fault.
\$2	FAILED. At least one slave failed
Other values are reserved	

changes to FAILED. For protocols that support non-cyclic communication only, the slave state is set to OK as soon as a valid configuration is found.

Slave Error Log Indicator

Not supported yet: the error log indicator field holds the number of entries in the internal error log. The field is set to zero if all entries are read from the log.

Number of Configured Slaves

The firmware maintains a list of slaves with which the master has to open a connection. This list is derived from the configuration database created by SYCON.net. This field holds the number of configured slaves.

Number of Active Slaves

The firmware maintains a list of slaves to which the master exchanges process data. This field holds the number of active slaves. Ideally, the number of active slaves is equal to the number of configured slaves. For certain fieldbus systems, it could be possible that a slave is shown as activated, but still has a problem (i.e. a diagnostic issue).

Number of Faulted Slaves

The firmware maintains a list of slaves that are missing on the network, although they may be configured, or are reporting a diagnostic issue. As long as those indications are pending and not serviced, the field holds a nonzero value. If no more diagnostic information is pending, the field is set to zero again.

	Hilscher Documentation	ACC-72EX Setup Assistant	
	RCX_COMM_COS_READY	CCx_RCX_COMM_COS_READY	
	RCX_COMM_COS_RUN	CCx_RCX_COMM_COS_RUN	
	RCX_COMM_COS_BUS_ON	CCx_RCX_COMM_COS_BUS_ON	
	RCX_COMM_COS_CONFIG_LOCKED	CCx_RCX_COMM_COS_CONFIG_LOCKED	
	RCX_COMM_COS_CONFIG_NEW	CCx_RCX_COMM_COS_CONFIG_NEW	
	RCX_COMM_COS_RESTART_REQ	CCx_RCX_COMM_COS_RESTART_REQ	
	RCX_COMM_COS_RESTART_REQ_ENA	CCx_RCX_COMM_COS_RESTART_REQ_ENA	
	RCX_COMM_COS_DMA	CCx_RCX_COMM_COS_DMA	
	ulCommunicationState	CCx_ulCommunicationState	
	ulCommunicationError	CCx_ulCommunicationError	
	usVersion	CCx_usVersion	
Common Status Block	usWatchdogTime	CCx_usWatchdogTime	
B	bPDInHskMode	CCx_bPDInHskMode	
atus	bPDInSource	CCx_bPDInSource	
Sta	bPDOutHskMode	CCx_bPDOutHskMode	
uo	bPDOutSource	CCx_bPDOutSource	
шш	ulHostWatchdog	CCx_ulHostWatchdog	
Ō	ulErrorCount	CCx_ulErrorCount	
	bErrorLogInd	CCx_bErrorLogInd	
	bErrorPDInCnt	CCx_bErrorPDInCnt	
	bErrorPDOutCnt	CCx_bErrorPDOutCnt	
	bErrorSyncCnt	CCx_bErrorSyncCnt	
	bSyncHskMode	CCx_bSyncHskMode	
	bSyncSource	CCx_bSyncSource	
	ulSlaveState	CCx_ulSlaveState	
	ulSlaveErrLogInd	CCx_ulSlaveErrLogInd	
	ulNumOfConfigSlaves	CCx_ulNumOfConfigSlaves	
	ulNumOfActiveSlaves	CCx_ulNumOfActiveSlaves	
	ulNumOfDiagSlaves	CCx_ulNumOfDiagSlaves	
Note: x	Note: x in MACRO name is replaced by Application Channel number 0 3		

Application Channel

The application channel is reserved for user specific implementations. An application channel is not yet supported.

Auto-Generated Dual-Ported Memory Map

ACC-72EX Setup Assistant Software, designed for use with Turbo PMAC, provides some level of automation in the identification of Hilscher COMX modules by generating a memory map file, suggested M-Variable definitions for important registers, and appropriate macro names.

emory Map Generator		Address Converter	
Connect to PM/	NC	ACC-72EX Base Address	\$6C000 -
		Hilscher Address Offset	0x0
Address Selection	\$6C000 -	Hilscher Data Width	32 -
Starting M-Variable Number	6000 🚔	Hilscher Data Start Bit	0 -
			Convert
Generate M-Variable D	efinitions	Equivalent PMAC Address	D:\$6C000
PMAC is selected for communic nnection to PMAC was success ACC-72EX cards detected.		2	

Address Converter

The Address Converter section of the software allows conversion of offset, bit, and width parameters to PMAC memory addresses based on Hilscher documentation.

Memory Map Generator

The Memory Map Generator section of the software identifies the ACC-72EX cards in a UMAC system and generates both a memory map as a text file and M-Variable definition file with proper addressing, both of which indicate the ACC-72EX-based address selection.

Reading the Memory Map Text File

The output file from the software is a text file which can be read with any text editor software. This file includes generic information about the card.

Below is an example output file. Please see the notes in the right column for more information on specific items.

HilscherMemoryMap_\$6C000.txt File C		Notes
Delta Tau Data Systems, Inc		
ACC-72EX Setup Assistant Au	to-generated Memory Map	
ACC-72EX Address: \$6C000		Base address of the ACC- 72EX selected in the Memory Map Generator section
netX Identification:	netX	The identification cookie provided by the netX firmware
Dual-Port Memory Size:	65536 bytes	
Device Number:	1532100	
Serial Number:	21456	
Hardware Assembly Options:		
Port 0:	ETHERNET (internal Phy)	
Port 1:	ETHERNET (internal Phy) ETHERNET (internal Phy) NOT CONNECTED	
Port 2:	NOT CONNECTED	
Port 3:		
Master) (DeviceNet Master)	Date: Week 18 of 2012 ormation: (PROFIBUS Master) (CANopen (AS-Interface Master) (PROFINET IO RT r) (EtherNet/IP Scanner) (SERCOS III	
Fool License Information:	(SYCON.net)	
Device Class:	COMX 100	
+ Block 0:		Block information
Channel Type:	System	For all blocks
Size of Channel:	512 bytes	
Channel Start Address:	\$6C000	
	Cells: IN HANDSHAKE CHANNEL	
netX System Flags Adre	ss: X:\$6C080,0,8	Calculates where the
Host System Flags Adre	ss: X:\$6C080,8,8	handshake registers are
Size of Handshake Cell		Located
<pre> Size of Mailbox: Mailbox Start address:</pre>	256 bytes \$6C040	
Number of Subblocks:		
Subblock 0: COMMON ST. Size:		Lists all channels' Sub- Blocks
Size. Start Offset:	176 bytes	BIOCKS
	IN - OUT (Bi-Directional)	
Transfer Type:	DPM (Dual-Port Memory)	
Handshake Mode:	UNCONTROLLED	
Handshake Bit:		
 Subblock 1: CONTROL		
Size:	8 bytes	
Size: Start Offset:	8 bytes \$6C02E	
Start Offset:	\$6C02E	
Start Offset:		
Start Offset: Transfer Direction:	\$6C02E OUT (Host System to netX)	
Start Offset: Transfer Direction: Transfer Type:	\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory)	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST</pre>	\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS	
<pre>Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size:</pre>	\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset:</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction:</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System)</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction: Transfer Type:</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System) DPM (Dual-Port Memory)</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction:</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System)</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: </pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 3: MAILBOX</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit:</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 3: MAILBOX Size: Start Offset:</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 128 bytes \$6C040</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 3: MAILBOX Size: Start Offset: Transfer Direction:</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 128 bytes</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 3: MAILBOX Size: Start Offset: Transfer Direction: Transfer Type:</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 128 bytes \$6C040 OUT (Host System to netX)</pre>	
<pre> Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 2: COMMON ST Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit: Subblock 3: MAILBOX Size: Start Offset: Transfer Direction: Transfer Type:</pre>	<pre>\$6C02E OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 ATUS 64 bytes \$6C030 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 128 bytes \$6C040 OUT (Host System to netX) DPM (Dual-Port Memory)</pre>	

|--- Subblock 4: MAILBOX Size: 128 bytes Start Offset: \$6C060 Transfer Direction: IN (netX to Host System) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: UNKNOWN Handshake Bit: 5 + Block 1: | Channel Type: Size of Channel: Handshake 256 bytes Channel Start Address: \$6C080 + Block 2: Channel Type: Communication Size of Channel: 15616 bytes Channel Start Address: \$6C0C0 Position of Handshake Cells: IN HANDSHAKE CHANNEL Size of Handshake Cells: 16 BITS NetX Handshake Register: Y:\$6C082,0,16 Host Handshake Register: X:\$6C082,0,16 Communication Class: SCANNER Communication Class: Protocol Class: IO-DEVICE Conformance Class: 0 Number of Subblocks: 9 |--- Subblock 0: CONTROL Size: 8 bytes Start Offset: \$6C0C2 Transfer Direction: OUT (Host System to netX) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: UNCONTROLLED Handshake Bit: 0 |--- Subblock 1: COMMON STATUS Start Offset: Transfer Direction: IN (netX to Host System) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: UNCONTROLLED Handshake Bit: 0 --- Subblock 2: EXTENDED STATUS 432 bytes t: \$6C0D4 Size: Start Offset: Transfer Direction: IN (netX to Host System) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: UNCONTROLLED Handshake Bit: 0 |--- Subblock 3: MAILBOX 1600 bytes : \$6C140 Size: Start Offset: Transfer Direction: OUT (Host System to netX) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: BUFFERED, HOST CONTROLLED Handshake Bit: 4 |--- Subblock 4: MAILBOX 1600 bytes \$6C2D0 Size: Start Offset: Transfer Direction: IN (netX to Host System) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: UNKNOWN Handshake Bit: 5 --- Subblock 5: PROCESS DATA IMAGE Size: 5760 bytes Start Offset: \$6C4C0

Transfer Direction: OUT (Host System to netX) Transfer Type: DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED Handshake Mode: Handshake Bit: 6 |--- Subblock 6: PROCESS DATA IMAGE Size: 5760 bytes \$6CA60 Start Offset: Transfer Direction: IN (netX to Host System) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: BUFFERED, HOST CONTROLLED Handshake Mode: Handshake Bit: 7 |--- Subblock 7: HIGH PRIORITY DATA IMAGE Size: 64 bytes Start Offset: \$6C460 Transfer Direction: OUT (Host System to netX) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: BUFFERED, HOST CONTROLLED Handshake Bit: 8 |--- Subblock 8: HIGH PRIORITY DATA IMAGE 64 bytes Size: Start Offset: \$6C470 Transfer Direction: IN (netX to Host System) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: BUFFERED, HOST CONTROLLED Handshake Bit: 9 + Block 3: Channel Type: Communication 15616 bytes \$6D000 Size of Channel: Channel Start Address: Position of Handshake Cells: IN HANDSHAKE CHANNEL Size of Handshake Cells: 16 BITS NetX Handshake Register: Y:\$6C083,0,16 Host Handshake Register: X:\$6C083,0,16 Communication Class: MESSAGING Communication Class: Protocol Class: UNDEFINED Conformance Class: 0 Number of Subblocks: 9 |--- Subblock 0: CONTROL Size: ° 5, \$6D002 8 bytes Start Offset: Transfer Direction: OUT (Host System to netX) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: UNCONTROLLED Handshake Bit: 0 --- Subblock 1: COMMON STATUS 64 bytes Size: Start Offset: \$6D004 Transfer Direction: IN (netX to Host System) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: UNCONTROLLED Handshake Bit: 0 |--- Subblock 2: EXTENDED STATUS 432 bytes \$6D014 Size: Start Offset: Transfer Direction: IN (netX to Host System) Transfer Type: DPM (Dual-Port Memory) Handshake Mode: UNCONTROLLED Handshake Bit: 0 |--- Subblock 3: MAILBOX 1600 \$6D080 1600 bytes Size: Start Offset: Transfer Direction: OUT (Host System to netX)

Transfer Type: Handshake Mode:	DPM (Dual-Port Memory)	
Handshake Mode: Handshake Bit:	BUFFERED, HOST CONTROLLED 4	
 Subblock 4: MAILBOX		
Size:	1600 bytes \$6D210	
Start Offset: Transfer Direction:	S6D210 IN (netX to Host System)	
Transfer Type:	DPM (Dual-Port Memory) UNKNOWN	
Handshake Mode: Handshake Bit:	UNKNOWN 5	
 Subblock 5: PROCESS D	ATA IMAGE	
Size: Start Offset:	5760 bytes \$6D400	
	OUT (Host System to netX)	
	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	
Handshake Mode: Handshake Bit:	BUFFERED, HOST CONTROLLED 6	
Subblock 6: PROCESS D	ATA IMAGE	
Size:	5760 bytes	
Start Offset: Transfer Direction:	\$6D9A0 IN (netX to Host System)	
Transfer Type:	DPM (Dual-Port Memory)	
Handshake Mode:	BUFFERED, HOST CONTROLLED	
Handshake Bit:	7	
Subblock 7: HIGH PRIO		
Size: Start Offset:	64 bytes \$6D3A0	
	OUT (Host System to netX)	
Transfer Type:	DPM (Dual-Port Memory)	
Handshake Mode: Handshake Bit:	BUFFERED, HOST CONTROLLED 8	
 Subblock 8: HIGH PRIO	RITY DATA IMAGE	
Size:	64 bytes	
Start Offset:	\$6D3B0 IN (netX to Host System)	
	DPM (Dual-Port Memory)	
Handshake Mode:	BUFFERED, HOST CONTROLLED	
Handshake Bit:	9	
- Block 4:		
Channel Type: Size of Channel:	Undefined 0 bytes	
Channel Start Address:	-	
	Cells: BEGINNING OF CHANNEL	
Size of Handshake Cell NetX Handshake Registe		
Host Handshake Registe		
Communication Class:	UNDEFINED	
Protocol Class: Conformance Class:	UNDEFINED 0	
Number of Subblocks:	0	
+ Block 5: Channel Type:	Undefined	
Size of Channel:	0 bytes	
Channel Start Address:	\$6DF40	
Position of Handshake Size of Handshake Cell	Cells: BEGINNING OF CHANNEL s: NOT AVAILABLE	
NetX Handshake Registe		
Host Handshake Registe	r: X:\$6DF40,8,0	
Communication Class: Protocol Class:	UNDEFINED UNDEFINED	
Conformance Class:	0	
Number of Subblocks:	0	

Suggested M-Variable Definition File Here is a sample macro name and suggested M-Variable definition file.

#define Sl abcokie 0 #6000 #define Sl abcokie 2 #6001 #define Sl abcokie 2 #6003 #define Sl ulbpwirotalSize #6004 #define Sl ulbpwirotalSize #6005 #define Sl uswhoptions 2 #6006 #define Sl uswhoptions 2 #6010 #define Sl uswhoptions 2 #6011 #define Sl uswhoptions 2 #6014 #define Sl De	MacroNa	ameDefinition_\$6C000.h File Content	
idefine SI_abcokis_2_ M6002 idefine SI_ulpericNumber M6003 idefine SI_ulpericNumber M6004 idefine SI_ulpericNumber M6006 idefine SI_usNumber M6006 idefine SI_usNumber M6008 idefine SI_usNumber M6009 idefine SI_usNumber(structurer M6010 idefine SI_usNumber(structurer M6011 idefine SI_usNumber(structurer M6012 idefine SI_usNumber(structurer) M6013 idefine SI_usNumber(structurer) M6014 idefine SI_usNumber(structurer) M6018 idefine SI_usNumber(structurer) M6019 idefine SI_usNumber(structurer) M6020 idefine SI_usNumber(structurer) M6021 idefine SI_usNumber(structurer) M6024	#define	SI abCookie 0	M6000
idefine s[_abcokte_3] M6003 idefine s[_ulbevicAlSize M6004 idefine s[_ulbevicAlSize M6005 idefine s[_usRvDplions_0			M6001
idefine SI ulpericENumber M6004 idefine SI ulbericENumber M6005 idefine SI usskuptions 0 M6006 idefine SI usskuptions 1 M6007 idefine SI usskuptions 2 M6009 idefine SI usskuptions 2 M6010 idefine SI usskuptions 2 M6011 idefine SI usskuptions 2 M6011 idefine SI usskuptions 2 M6013 idefine SI usskuptions 2 M6014 idefine SI byskuptions 2 M6018 idefine SI byskuptions 2 M6018 idefine SI byskuption 2 M6018 idefine SI byskuption 2 M6021 idefine SI byskuption 2 M6021 idefine SI byskuption 2 M6021 idefi	#define	SI_abCookie_2_	M6002
Hadrine ST_ulserialNumber M6005 Hddfine ST_ausHwOptions_0M6007 Hddfine ST_ausHwOptions_1M6008 Hddfine ST_ausHwOptions_1M6009 Hddfine ST_ausHwOptions_1M6009 Hddfine ST_ausHwOptions_1M6009 Hddfine ST_ausHwOptions_1M6009 Hddfine ST_uusHwoptions_1M6009 Hddfine ST_uusHwoptions_1M6019 Hddfine ST_beardAusher M6020 Hddfine ST_uusHwoptions_1M6020 Hddfine ST_uu	#define	SI abCookie 3	M6003
<pre>idefine SI_uslevQtions_0M6006 idefine SI_auslevQtions_0M6008 idefine SI_auslevQtions_0M6008 idefine SI_auslevQtions_0M6008 idefine SI_uslevQtionDate idefine SI_uslevQtionSI idefine SI_uslevQtio</pre>	#define	SI ulDpmTotalSize	M6004
idefine SI_ausHxOptions_1	#define	SI ulDeviceNumber	M6005
<pre>idefine ST_usHaOptions_2M6008 idefine ST_usHaOptions_2M6010 idefine ST_usHaOptions_3M6010 idefine ST_usHaOptions_3M6010 idefine ST_usHaOptions_3M6010 idefine ST_usHationseTLagsM6011 idefine ST_usHationseTLagsM6014 idefine ST_usHationseTLagsM6016 idefine ST_usHationseTLagsM6016 idefine ST_usHationseTLagsM6016 idefine ST_usHationseTLagsM6017 idefine ST_bHatCompatibilityM6019 idefine ST_bHatCompatibilityM6020 idefine ST_bHatCompatibilityM6020 idefine ST_bHatCompatibilityM6020 idefine ST_bHatCompatibilityM6020 idefine ST_bHatCompatibilityM6020 idefine ST_bHatCompatibilityM6020 idefine ST_usHatCompatibilityM6020 idefine CT_usHatCompatibilityM6020 idefine CT_usHatCompatibilityM6020 idefine CT_usHatCompatibilityM6020 idefine CTU_UsHatCompatibilityM6020 idefine CTU_UsHatCommandLationClass M6031 idefine CTU_UsHatCommandLationClass M6034 idefine CTU_UsHatCommandLass M6044 idefine CTU_U</pre>			M6006
idefine SI_ausHoptions_2_ M6009 idefine SI_usRoptions_1 M6010 idefine SI_usProductionDate M6012 idefine SI_ullicenselags1 M6013 idefine SI_usNetXicensel1ags1 M6014 idefine SI_usNetXicensel1ags1 M6015 idefine SI_usNetXicensel1ags1 M6016 idefine SI_usNetXicensel1ags1 M6016 idefine SI_usNetXicensel1ags1 M6017 idefine SI_usNetXicensel1ags1 M6018 idefine SI_bHecompatibility M6019 idefine SI_bhecompatibility M6020 idefine SCI_btanelType M6022 idefine SCI_usNatartOffset M6023 idefine SCI_usNatartOffset M6026 idefine CCI_btannelType M6023 idefine CCI_btannelType M6023 idefine CCI_btannelType M6023 idefine CCI_btannelType M6033 idefine CCI_btannelType M6033 idefine CCI_btannelType M6033 idefine CCI_btannelType M633 idefine CCI_btannelType M633 idefine CCI_btannelType M633 <td>#define</td> <td>SI_ausHwOptions_0_</td> <td>M6007</td>	#define	SI_ausHwOptions_0_	M6007
Hedfine SI_usManufacturer M6010 Hedfine SI_usManufacturer M6011 Hedfine SI_usTeroductionDate M6012 Hedfine SI_ullicenseFlags1 M6013 Hedfine SI_usNetXicenseID M6015 Hedfine SI_usNetXicenseID M6016 Hedfine SI_usNetXicenseID M6017 Hedfine SI_usNetXicenseID M6018 Hedfine SI_DNetQuellant M6020 Hedfine SI_DNetQuellant M6020 Hedfine SI_DNetQuellant M6022 Hedfine SI_DNetQuellant M6024 Hedfine SCI_USIzeOfChanel M6024 Hedfine SCI_USIzeOfChanel M6026 Hedfine COI_DNetStartOffset M6026 Hedfine COI_DNetStartOffset M6027 Hedfine COI_DNetStartOffset M6028 Hedfine COI_DNetStartOffset M6028 Hedfine COI_DNetStartOffset M6030 Hedfine COI_DNetStartOffset M6032 Hedfine COI_DNetStartOffset M6032 Hedfine COI_DNetStartOffset M6034 Hedfine COI_DNetStartOffset M6034 Hedfine COI_DNetStartOffset M6034 Hedfine COI_DNetStartOffset M6036 Hedfine COI_DNetStartOffset M6036 Hedfine COI_DNetStartOffset M6036 Hedfine COI_DNetStartOffset	#define	SI_ausHwOptions_1_	M6008
Hedfine SI_usManufacturer M6010 Hedfine SI_usManufacturer M6011 Hedfine SI_usTeroductionDate M6012 Hedfine SI_ullicenseFlags1 M6013 Hedfine SI_usNetXicenseID M6015 Hedfine SI_usNetXicenseID M6016 Hedfine SI_usNetXicenseID M6017 Hedfine SI_usNetXicenseID M6018 Hedfine SI_DNetQuellant M6020 Hedfine SI_DNetQuellant M6020 Hedfine SI_DNetQuellant M6022 Hedfine SI_DNetQuellant M6024 Hedfine SCI_USIzeOfChanel M6024 Hedfine SCI_USIzeOfChanel M6026 Hedfine COI_DNetStartOffset M6026 Hedfine COI_DNetStartOffset M6027 Hedfine COI_DNetStartOffset M6028 Hedfine COI_DNetStartOffset M6028 Hedfine COI_DNetStartOffset M6030 Hedfine COI_DNetStartOffset M6032 Hedfine COI_DNetStartOffset M6032 Hedfine COI_DNetStartOffset M6034 Hedfine COI_DNetStartOffset M6034 Hedfine COI_DNetStartOffset M6034 Hedfine COI_DNetStartOffset M6036 Hedfine COI_DNetStartOffset M6036 Hedfine COI_DNetStartOffset M6036 Hedfine COI_DNetStartOffset	#define	SI_ausHwOptions_2_	M6009
idefine ST_ustroductionDate M6012 idefine ST_ullicenseFlags1 M6013 idefine ST_ustricenseFlags2 M6014 idefine ST_ustricenseFlags M6016 idefine ST_ustricenseFlags M6017 idefine ST_ustricenseFlags M6017 idefine ST_ustricenseFlags M6018 idefine ST_berostion M6019 idefine ST_berostionOfHandshake M6022 idefine SCT_bhumberDBlocks M6023 idefine SCT_ustreDBlocks M6024 idefine SCT_ustreDBlocks M6024 idefine SCT_ustreDFlocks M6026 idefine SCT_ustreDFlocks M6027 idefine CCT_UstreDFlocks M6028 idefine CCT_UstreDFlocks M6028 idefine CCT_DStreDFlocks M6031 idefine CCT_DStreDFlocks M6032 idefine CCT_UstreDFlocks M6033 idefine CCT_UstreDFlocks M6034 idefine CCT_UstreDFlocks M6035 idefine CCT_UstreDFlocks M6035 idefine CCT_UstreDFlocks M6036 idefine CCT_UstreDFlocks M6036 <tr< td=""><td>#define</td><td>SI_ausHwOptions_3_</td><td></td></tr<>	#define	SI_ausHwOptions_3_	
define SI_ullicensePlags1M6013define SI_usNetLicensePlags2M6014define SI_usNetLicensePlagsM6015define SI_usNetLicensePlagsM6016define SI_usNetLicensePlagsM6017define SI_bhweevisionM6020define SI_bhweisionM6020define SI_bhanevisionM6020define SI_blanevisionM6021define SI_blanevisionM6020define SI_blanevisionM6021define SI_blanevisionM6022define SI_blanevisionM6023define SI_usIzeOrShocksM6024define SI_usIzeOrShocksM6025define SI_usIzeOrShocksM6026define CI_usIzeOrShicksM6027define CI_bchannelTypeM6029define CI_blanevisionOrBandshakeM6031define CI_usIzeOrChannelM6031define CI_usIzeOrChannelM6033define CI_usIzeOrChannelM6034define CI_usIzeOrChannelM6035define CI_usIzeOrChannelM6036define CI_usIzeOrChannelM6036define CI_usIzeOrChannelM6036define CI_usIzeOrChannelAM6037define CI_usIzeOrChannelAM6038define CI_usIzeOrChannelAM6039define CI_usIzeOrChannelM6040define CI_usIzeOrChannelM6041define CI_usIzeOrChannelM6045define CI_usIzeOrChannelM6045define CI_usIzeOrChannelM6045define CI_usIzeOrChannelM6045define CI_usIzeOrChannelM6045define CI_usIzeOrChan			M6011
Hedfine ST_ullicenseFlags2M6014Hedfine ST_usNetXicenseFlagsM6015Hedfine ST_usNetxicenseFlagsM6017Hedfine ST_usNetxiceClassM6019Hedfine ST_bMxCompatibilityM6019Hedfine ST_bMxCompatibilityM6012Hedfine ST_bNxCompatibilityM6012Hedfine ST_bNxCompatibilityM6020Hedfine ST_bNxCompatibilityM6021Hedfine ST_bNxCompatibilityM6021Hedfine ST_bNxCompatibilityM6023Hedfine ST_usNet/CompatibilityM6024Hedfine ST_usNet/CompatibilityM6026Hedfine ST_usNet/CompatibilityM6026Hedfine COT_bNameITypeM6028Hedfine COT_bChannelM6028Hedfine COT_bChannelItypeM6030Hedfine COT_bNumberOfFlocksM6030Hedfine COT_UsNetDrofFlocksM6031Hedfine COT_UsNetDrofFlocksM6036Hedfine COT_UsNetDrofFlocksM6036Hedfine COT_usPortocolClassM6036Hedfine COT_UsNetDrofFlocksM6036Hedfine COT_UsContensionM6038Hedfine COT_UsContensionM6040Hedfine COT_UsContensionM6041Hedfine COT_UsContensionM6043Hedfine COT_UsContensionM6044Hedfine COT_UsContensionM6043Hedfine COT_UsContensionM6044Hedfine COT_UsContensionM6044Hedfine COT_UsContensionM6044Hedfine COT_UsContensionM6043Hedfine COT_UsContensionM6044Hedfine COT_UsContensionM6044Hedfine			
define sT_usNetLicenseElDM6015define sT_usNetLicenseElagsM6017define sT_beviceClassM6017define sT_beviceClassM6019define sT_beviceClassM6020define sT_beviceClassM6021define sT_beviceClassM6022define sT_beviceClassM6022define sT_beviceClassM6023define sCT_bSizePositionOfHandshakeM6023define sCT_usSizeOfChanelM6026define sCT_usSizeOfChanelM6026define sCT_usSizeOfChanelM6027define cCOT_bChannelTypeM6027define cCOT_bChannelTypeM6029define cCOT_bNumberOfBlocksM6031define cCOT_bNumberOfBlocksM6032define cCOT_usSizeOfChannelM6034define cCOT_usCordmanctalssM6034define cCOT_usCordmanctalssM6036define cCOT_usCordmanctalssM6036define cCIT_bChannelTypeM6037define cCIT_bSizePositionOfHandshakeM6038define cCIT_bSizePositionOfHandshakeM6038define cCIT_bSizePositionOfHandshakeM6036define cCIT_bSizePositionOfHandshakeM6036define cCIT_bSizePositionOfHandshakeM6038define cCIT_bSizePositionOfHandshakeM6036define cCIT_bSizePositionOfHandshakeM6038define cCIT_bSizePositionOfHandshakeM6038define cCIT_bSizePositionOfHandshakeM6038define cCIT_bSizePositionOfHandshakeM6040define cCIT_bSizePositionOfHandshakeM6041define cCIT_bSizePositionOfHandshak			M6013
define ST_ushevicelassM6016define ST_bhReenithilityM6017define ST_bhReenithilityM6019define ST_boxDistrict Statedefine ST_boxM6021define ST_bracenthilityM6021define ST_bracenthilityM6021define ST_bracenthilityM6021define ST_bracenthilityM6023define ST_bracenthilityM6024define ST_ushalbackM6024define ST_ushalbackM6026define ST_ushalbackM6026define RT_ushalbackM6027define RT_ushalbackM6028define CT_bchannelTypeM6029define CT_bchannelTypeM6030define CT_bchannelTapeM6032define CT_bchannelTapeM6032define CT_UshubbackTablackM6031define CT_bchannelTapeM6032define CT_UshubbackTablackM6031define CT_UshubbackTablackM6034define CT_UshubbackTablackM6036define CT_UsbackTablackM6036define CT_UsbackTablackM6038define CT_UsbackTablackM6038define CT_UsbackTablackM6041define CT_UsbackTablackM6042define CT_UsbackTablackM6043define CT_UsbackTablackM6043define CT_UsbackTablackM6044define CT_UsbackTablackM6045define CT_UsbackTablackM6044define CT_UsbackTablackM6045define CT_UsbackTablackM6046define CT_UsbackTablackM6046define CT_UsbackTa			
define ST_uspeviceClassM6017define SI_bbwcOmpatibilityM6018define SI_bbwcOmpatibilityM6020define SCI_bChannelTypeM6021define SCI_bChannelTypeM6023define SCI_usDizeOcEnnelM6024define SCI_usDizeOCEnnelM6024define SCI_usDizeOCEnnelM6024define SCI_usDizeOCEnnelM6025define SCI_usDizeOCEnnelM6026define CI_usDizeOCEnnelM6027define CCI_bChannelTypeM6027define CCI_bChannelTypeM6029define CCI_bChannelTypeM6031define CCI_bChannelTypeM6031define CCI_usDizeOCEnnelM6033define CCI_usDizeOcEnnelM6034define CCI_usDizeOcEnnelM6034define CCI_usCommunicationClassM6034define CCI_usCommunicationClassM6036define CCI_usCommunicationClassM6037define CCI_bCannelIdM6038define CCI_UsDizeOfEncksM6038define CCI_UsContranceClassM6036define CCI_usContranceClassM6040define CCI_UsDizeOfEncksM6041define CCI_UsDizeOfEncksM6042define CCI_UsDizeOfEncksM6041define CCI_UsDizeOfEncksM6042define CCI_UsDizeOfEncksM6042define CCI_UsDizeOfEncksM6042define CCI_UsDizeOfEncksM6043define CCI_UsDizeOfEncksM6045define CCI_UsDizeOfEncksM6045define CCI_UsDizeOfEncksM6045define CCI_UsDizeOfEncksM6045 <tr< td=""><td></td><td></td><td></td></tr<>			
idefine SI_biskCompatibilityM6019idefine SI_bbevIdNumberM6020idefine SCI_bSIzePositionOfHandshakeM6021idefine SCI_bSIzePositionOfHandshakeM6022idefine SCI_uSIzePositionOfHandshakeM6022idefine SCI_uSizeOChannelM6024idefine SCI_uSizeOChannelM6026idefine SCI_uSAizOCChannelM6027idefine CCI_USAizOCChannelM6027idefine CCI_blareOChannelM6028idefine CCI_blareOChannelM6028idefine CCI_blareOChannelM6030idefine CCI_blareOChannelM6030idefine CCOI_blareOChannelM6030idefine CCOI_uSizeOstionOfHandshakeM6031idefine CCOI_uSizeOstionOfHandshakeM6033idefine CCOI_uSizeOstionOfHandshakeM6033idefine CCOI_uSizeOstionOfHandshakeM6036idefine CCII_uSizeOstionOfHandshakeM6037idefine CCII_uScontentareOlassM6036idefine CCII_blareOfTanceOlassM6037idefine CCII_blareOfTanceOlassM6040idefine CCII_uScontentareOlassM6040idefine CCII_uScontentareOlassM6040idefine CCII_uScontentareOlassM6041idefine CCII_uScontentareOlassM6041idefine CCII_uScontentareOlassM6041idefine CCII_uScontentareOlassM6041idefine CCII_uScontentareOlassM6041idefine CCII_uScontentareOlassM6041idefine CCII_uScontentareOlassM6051idefine CCII_uScontentareOlassM6051idefine CCII_uScontentareOlassM6051			
define ST_bRxCompatibilityM6019define ST_bChannelTypeM6021define SCT_bStzePositionOfHandshakeM6023define SCT_usSizeOCManlbAM6024define SCT_usSizeOCManlbAM6024define SCT_usSizeOCManlbAM6025define SCT_usSizeOCManlbAM6026define SCT_usSizeOCManlbAM6027define SCT_usSizeOCManlbAM6027define CCT_bChannelTypeM6027define CCT_bChannelTypeM6030define CCT_bSizePositionOfHandshakeM6031define CCT_UsSizeOCTAnnelM6033define CCT_UsSizePOsitionOfHandshakeM6032define CCT_UsSizePOsitionOfHandshakeM6032define CCT_UsSizePOsitionOfHandshakeM6033define CCT_UsCommunicationClassM6036define CCT_UsCommunicationClassM6036define CCT_UsCommunicationClassM6037define CCT_UsCommunicationClassM6041define CCT_UsCommunicationClassM6043define CCT_UsCommunicationClassM6041define CCT_UsCommunicationClassM6041define CCT_UsCommunicationClassM6043define CCT_UsSizePOsitionOfHandshakeM6043define CCT_UsCommunicationClassM6043define CCT_UsSizePOsitionOfHandshakeM6041define CCT_UsSizePOsitionOfHandshakeM6043define CCT_UsSizePOsitionOfHandshakeM6043define CCT_UsSizePOsitionOfHandshakeM6044define CCT_UsSizePOsitionOfHandshakeM6043define CCT_UsSizePOsitionOfHandshakeM6043define CCT_UsSizePOsitionOfH		=	
Hearine ST DDevTaNumberM6020Hearine SCT bolizePositionOfHandshakeM6021Hedrine SCT DolizePositionOfHandshakeM6023Hedrine SCT ublizeOfChannelM6024Hedrine SCT ublizeOfChannelM6024Hedrine SCT ublizeOfChannelM6026Hedrine KCT ublizeOfChannelM6026Hedrine KCT ublizeOfChannelM6027Hedrine KCT ublizeOfChannelM6028Hedrine CCOT bolizeOfChannelM6020Hedrine CCOT bolizeOfChannelM6030Hedrine CCOT ublizeOfChannelM6032Hedrine CCOT ublizeOfChannelM6033Hedrine CCOT ublizeOfChannelM6034Hedrine CCOT usCommunicationClassM6035Hedrine CCIT ublizeOfChannelM6037Hedrine CCIT ublizeOfIchannelM6037Hedrine CCIT ublizeOfIchannelM6038Hedrine CCIT ublizeOfIchannelM6037Hedrine CCIT ublizeOfIchannelM6041Hedrine CCIT ublizeOfIchannelM6041Hedrine CCIT ublizeOfIchannelM6042Hedrine CCIT ublizeOfIchannelM6043Hedrine CCIT ublizeOfIchannelM6044Hedrine CCIT ublizeOfIchannelM6046Hedrine CCIT ublizeOfIchannelM6046Hedrine CCIT ublizeOfIchannelM6051Hedrine CCIT ublizeOf			
HefineSCT bchannelTypeM6021#defineSCT blizePositionOfHandshakeM6023#defineSCT blizePofHanloXM6024#defineSCT usSizeOfHanloXM6025#defineSCT usSizeOfHanloXM6026#defineCCT usDizeOfHanloXM6027#defineCCT bchannelTypeM6027#defineCCT bchannelTypeM6029#defineCCT bchannelTypeM6030#defineCCT bchannelTypeM6031#defineCCT usDizeOfEnanelM6032#defineCCT usDizePositionOfHandshakeM6032#defineCCT usDizePositionOfHandshakeM6033#defineCCT usDizePositionOfHandshakeM6034#defineCCT usDizePositionOfHandshakeM6034#defineCCT usDizePositionOfHandshakeM6034#defineCCT usDizePositionOfHandshakeM6034#defineCCT usDizePositionOfHandshakeM6038#defineCCT usDizePositionOfHandshakeM6039#defineCCT usDizePositionOfHandshakeM6040#defineCCT usDizePositionOfHandshakeM6041#defineCCT usDizePositionOfHandshakeM6043#defineCCT usDizePositionOfHandshakeM6043#defineCCT usDizePositionOfHandshakeM6044#defineCCT usDizePositionOfHandshakeM6044#defineCCT usDizePositionOfHandshakeM6046#defineCCT usDizePositionOfHandshakeM6045#defineCCT usDizePositionOfHandshakeM6055			
Hefine SCI_blizePositionOfHandshakeM6022#define SCI_blizePositionOfHandshakeM6023#define SCI_ubsizeOfAnnelM6024#define SCI_ussizeOfManloxM6026#define CCI_ussizeOfChannelM6026#define CCI_blasStartOffsetM6027#define CCI_blasStartOffsetM6028#define CCI_bChannelTypeM6030#define CCOI_bChannelTypeM6031#define CCOI_bSizePositionOfHandshakeM6031#define CCOI_usCommunicationClassM6034#define CCOI_usCommunicationClassM6036#define CCII_bSizePositionOfHandshakeM6036#define CCII_bSizePositionOfHandshakeM6036#define CCII_bSizePositionOfHandshakeM6036#define CCII_bSizePositionOfHandshakeM6036#define CCII_bSizePositionOfHandshakeM6036#define CCII_bSizePositionOfHandshakeM6038#define CCII_usCommunicationClassM6041#define CCII_usCommunicationClassM6041#define CCII_usCommunicationClassM6043#define CCII_usCommunicationClassM6043#define CCII_usConformanceClassM6043#define CCII_usConformanceClassM6043#define CCII_usConformanceClassM6046#define CCII_usConformanceClassM6046#define CCII_usConformanceClassM6051#define CCII_usConformanceClassM6051#define CCII_usConformanceClassM6051#define CCII_usConformanceClassM6051#define CCII_usConformanceClassM6052#define CCII_usConformanceClassM6051			
HefineSCT_DNumberOfBlocksM6023#defineSCT_usSizeOfMailboxM6025#defineSCT_usMailboxtartOffsetM6026#defineKCT_UsSizeOfChannelTypeM6027#defineKCT_UsSizeOfChannelTypeM6028#defineCCOT_bChannelTypeM6020#defineCCOT_bChannelTypeM6020#defineCCOT_bNumberOfBlocksM6031#defineCCOT_usSizePositionOfHandshakeM6032#defineCCOT_usSizePositionClassM6034#defineCCOT_usComunicationClassM6034#defineCCOT_usComunicationClassM6036#defineCCIT_bNumberOfBlocksM6037#defineCCIT_bSizePositionOfHandshakeM6039#defineCCIT_bSizePositionOfHandshakeM6039#defineCCIT_usCommunicationClassM6040#defineCCIT_usCommunicationClassM6041#defineCCIT_usCommunicationClassM6043#defineCCIT_usConformanceClassM6043#defineCCIT_usConformanceClassM6043#defineCCIT_usConformanceClassM6043#defineCCIT_usConformanceClassM6045#defineCCIT_usConformanceClassM6046#defineCCIT_usConformanceClassM6046#defineCCIT_usConformanceClassM6050#defineCCIT_usConformanceClassM6051#defineCCIT_usConformanceClassM6051#defineCCIT_usConformanceClassM6053#defineCCIT_usConformanceClass </td <td></td> <td></td> <td></td>			
Hefine SCT_ulsizeOfChannelM6024#define SCT_usMailboxStartOffsetM6025#define SCT_usMailboxStartOffsetM6026#define RCT_UsInsiteOfChannelM6027#define CCOT_bchannelTypeM6028#define CCOT_bchannelTypeM6020#define CCOT_bsizePositionOfHandshakeM6031#define CCOT_usIceOfChannelM6033#define CCOT_usCommunicationClassM6033#define CCIT_usCommunicationClassM6035#define CCIT_bsizePositionOfHandshakeM6037#define CCIT_usCommunicationClassM6036#define CCIT_bsizePositionOfHandshakeM6037#define CCIT_bchannelTypeM6037#define CCIT_bsizePositionOfHandshakeM6038#define CCIT_bchannelTypeM6037#define CCIT_bsizePositionOfHandshakeM6038#define CCIT_usComformanceClassM6040#define CCIT_usComformanceClassM6040#define CCIT_usComformanceClassM6043#define CCIT_usComformanceClassM6043#define CCIT_usConformanceClassM6043#define CCIT_usConformanceClassM6045#define CCIT_usConformanceClassM6046#define CCIT_bumberOfBlocksM6046#define CCIT_usStepositionOfHandshakeM6050#define CCIT_usStepositionOfHandshakeM6051#define CCIT_usStepositionOfHandshakeM6051#define CCIT_usStepositionOfHandshakeM6055#define CCIT_usStepositionOfHandshakeM6055#define CCIT_usStepositionOfHandshakeM6055#define CCIT_usStepositionOfHandshake			
HefineSCT_usSizeOfMailboxM6025#defineSCT_usMailboxStartOffsetM6026#defineSCT_usMailboxStartOffsetM6027#defineRCT_ulSizeOfChannelTypeM6028#defineCCOT_bChannelTypeM6030#defineCCOT_bSizePositionOfHandshakeM6031#defineCCOT_bSizePositionOfHandshakeM6032#defineCCOT_usCommunicationClassM6033#defineCCOT_usConformanceClassM6034#defineCCIT_bChannelTypeM6037#defineCCIT_bChannelTypeM6037#defineCCIT_bChannelTypeM6037#defineCCIT_bChannelTypeM6037#defineCCIT_bChannelTypeM6043#defineCCIT_bChannelTypeM6040#defineCCIT_usConformanceClassM6040#defineCCIT_usConformanceClassM6040#defineCCIT_usCommunicationClassM6042#defineCCIT_usSizeOfChannelM6043#defineCCIT_usSizeOfChannelM6044#defineCCIT_usSizeOfChannelM6046#defineCCIT_usSizeOfChannelM6046#defineCCIT_usSizeOfChannelM6046#defineCCIT_usSizeOfChannelM6046#defineCCIT_usSizeOfChannelM6046#defineCCIT_usSizeOfChannelM6046#defineCCIT_usSizeOfChannelM6046#defineCCIT_usSizeOfChannelM6046#defineCCIT_usSizeOfChannelM6046#defineCCIT_usSizeOfChan			
#defineSCT_usMailboxStartOffsetM6026#defineMCT_uSizeOfChannelM6027#defineCCOT_bChannelTypeM6028#defineCCOT_bChannelTypeM6030#defineCCOT_bSizePositionOfHandshakeM6031#defineCCOT_bSizePositionOfHandshakeM6032#defineCCOT_usCommunicationClassM6033#defineCCOT_usCommunicationClassM6035#defineCCOT_usCommunicationClassM6035#defineCCOT_usConmentClassM6036#defineCCIT_bChannelTypeM6037#defineCCIT_bChannelTypeM6037#defineCCIT_bSizePositionOfHandshakeM6039#defineCCIT_bSizePositionOfHandshakeM6040#defineCCIT_usCommunicationClassM6041#defineCCIT_usCommunicationClassM6041#defineCCIT_usCommunicationClassM6042#defineCCIT_usConmentClassM6043#defineCCIT_usConmentClassM6044#defineCCIT_usConmunicationClassM6046#defineCCIT_usConmunicationClassM6047#defineCCIT_usConmunicationClassM6050#defineCCIT_usConmunicationClassM6050#defineCCIT_usConmunicationClassM6050#defineCCIT_usConmunicationClassM6050#defineCCIT_usConformanceClassM6050#defineCCIT_usConformanceClassM6050#defineCCIT_usConformanceClassM6050#defineCCIT_usizePosition			
Hetine HCI_DChannelTypeM6027#define HCI_ulSizeOfChannelM6028#define CCOI_bChannelTypeM6029#define CCOI_bChannelTypeM6030#define CCOI_bSPersitionOfHandshakeM6031#define CCOI_ulSizeOfChannelM6033#define CCOI_ulSizeOfChannelM6034#define CCOI_usCommunicationClassM6035#define CCOI_usProtocolClassM6036#define CCII_bChannelTypeM6037#define CCII_bChannelTypeM6037#define CCII_bChannelTypeM6038#define CCII_bChannelTypeM6041#define CCII_bChannelTypeM6041#define CCII_bChannelAM6040#define CCII_usCommunicationClassM6041#define CCII_usCommunicationClassM6043#define CCII_usCommunicationClassM6044#define CCII_bChannelTypeM6045#define CCII_bChannelTypeM6046#define CCII_bChannelTypeM6046#define CCII_bChannelTypeM6046#define CCII_bChannelTypeM6046#define CCII_bChannelTypeM6046#define CCII_bChannelTypeM6046#define CCII_bChannelTypeM6051#define CCII_bChannelTypeM6051#define CCII_usCommunicationClassM6051#define CCII_usProtocolClassM6051#define CCII_usProtocolClassM6052#define CCII_usProtocolClassM6051#define CCII_usProtocolClassM6052#define CCII_usProtocolClassM6056#define CCII_usProtocolClassM6056#define CCII_usPro			
Hetrine HcT_ulSizeOfChannelM6028#define CC0T_bChannelTypeM6029#define CC0T_bSizePositionOfHandshakeM6031#define CC0T_ulSizeOfChannelM6033#define CC0T_ulSizeOfChannelM6033#define CC0T_usCommunicationClassM6034#define CC0T_usCommunicationClassM6035#define CC0T_usConformanceClassM6036#define CC1T_bChannelTypeM6037#define CC1T_bChannelTypeM6038#define CC1T_bChannelTypeM6034#define CC1T_bSizePositionOfHandshakeM6039#define CC1T_usCommunicationClassM6040#define CC1T_usCommunicationClassM6042#define CC1T_usCommunicationClassM6043#define CC1T_usCommunicationClassM6044#define CC2T_bChannelTypeM6045#define CC2T_bChannelTypeM6045#define CC2T_bSizePositionOfHandshakeM6047#define CC2T_bSizePositionOfHandshakeM6046#define CC2T_bSizePositionOfHandshakeM6047#define CC2T_usCommunicationClassM6050#define CC2T_usCommunicationClassM6050#define CC2T_usCommunicationClassM6051#define CC3T_bSizePositionOfHandshakeM6052#define CC3T_bSizePositionOfHandshakeM6055#define CC3T_bSizePositionOfHandshakeM6055#define CC3T_bSizePositionOfHandshakeM6055#define CC3T_bSizePositionOfHandshakeM6055#define CC3T_bSizePositionOfHandshakeM6055#define CC3T_bSizePositionOfHandshakeM6055#define CC3T_bSizePositionOfHands			
#defineCCOT_bChannelTypeM6029#defineCCOT_bChanneltdM6030#defineCCOT_bChanneltdM6031#defineCCOT_uStePositionOfHandshakeM6032#defineCCOT_usCommunicationClassM6033#defineCCOT_usCommunicationClassM6035#defineCCOT_usCommunicationClassM6036#defineCCOT_usConformanceClassM6036#defineCCIT_bChannelTypeM6037#defineCCIT_bChannelTypeM6038#defineCCIT_bChannelTypeM6040#defineCCIT_usCommunicationClassM6040#defineCCIT_usCommunicationClassM6041#defineCCIT_usCommunicationClassM6043#defineCCIT_usConformanceClassM6043#defineCCIT_bChannelTidM6046#defineCCIT_bChannelTidM6046#defineCCIT_bChannelTidM6046#defineCCIT_bChannelTidM6046#defineCCIT_bSizePositionOfHandshakeM6047#defineCCIT_usConformanceClassM6050#defineCCIT_usConformanceClassM6051#defineCCIT_usConformanceClassM6052#defineCCIT_usConformanceClassM6052#defineCCIT_usConformanceClassM6052#defineCCIT_usConformanceClassM6055#defineCCIT_usConformanceClassM6056#defineCCIT_usProtocolClassM6056#defineCCIT_usProtocolClassM6056#defineCCIT_u			
#defineCCOI_bChannelIdM6030#defineCCOI_bSizePositionOfHandshakeM6031#defineCCOI_uSizeOfChannelM6033#defineCCOI_uSCommunicationClassM6034#defineCCOI_usConformanceClassM6035#defineCCII_bChannelTypeM6037#defineCCII_bChannelTypeM6037#defineCCII_bChannelTypeM6037#defineCCII_bChannelTypeM6037#defineCCII_bSizePositionOfHandshakeM6038#defineCCII_uSCommunicationClassM6040#defineCCII_uSCommunicationClassM6041#defineCCII_usConformanceClassM6042#defineCCII_usConformanceClassM6044#defineCCII_usConformanceClassM6044#defineCCII_bSizePositionOfHandshakeM6045#defineCCII_bSizePositionOfHandshakeM6047#defineCCII_bSizePositionOfHandshakeM6047#defineCCII_uSizeOfChannelM6048#defineCCII_uSizeOfChannelM6049#defineCCII_uSizeOfChannelM6050#defineCCII_usConformanceClassM6051#defineCCII_usConformanceClassM6052#defineCCII_usConformanceClassM6053#defineCCII_usConformanceClassM6054#defineCCII_usConformanceClassM6054#defineCCII_usConformanceClassM6056#defineCCII_usConformanceClassM6056#defineCCII_usConformanceClassM6058<			
#defineCC01_bSizePositionOfHandshakeM6031#defineCC01_bNumberOfBlocksM6032#defineCC01_uSCommunicationClassM6033#defineCC01_usConformanceClassM6036#defineCC11_bChannelTypeM6037#defineCC11_bChannelTypeM6037#defineCC11_bChannelIqM6038#defineCC11_bChannelIdM6038#defineCC11_bChannelIdM6040#defineCC11_usCommunicationClassM6040#defineCC11_usCommunicationClassM6041#defineCC11_usCommunicationClassM6042#defineCC11_usConformanceClassM6043#defineCC21_bChannelTypeM6045#defineCC21_bChannelTypeM6046#defineCC21_uSizePositionOfHandshakeM6049#defineCC21_uSizePositionOfHandshakeM6049#defineCC21_uSizePositionOfHandshakeM6049#defineCC21_uSizePositionOfHandshakeM6050#defineCC21_uSizeOfChannelM6049#defineCC21_uSizeOfChannelM6051#defineCC31_uSizeOfChannelAsM6052#defineCC31_uSizeOfChannelAsM6055#defineCC31_bChannelIdM6056#defineCC31_bSizePositionOfHandshakeM6056#defineCC31_bSizeOfChannelM6057#defineCC31_bSizeOfChannelM6057#defineCC31_bSizeOfChannelM6057#defineCC31_bSizeOfChannelM6057#define <t< td=""><td></td><td></td><td></td></t<>			
#define CC0T_bNumberOfBlocksM6032#define CC0T_ulSizeOfChannelM6033#define CC0T_usPortocolClassM6034#define CC0T_usPortocolClassM6035#define CC1T_bChannelTypeM6037#define CC1T_bChannelIdM6038#define CC1T_bSizePositionOfHandshakeM6039#define CC1T_ulSizeOfChannelM6040#define CC1T_usConformanceClassM6040#define CC1T_ulSizeOfChannelM6041#define CC1T_usConformanceClassM6043#define CC1T_usConformanceClassM6043#define CC1T_usConformanceClassM6044#define CC1T_usConformanceClassM6044#define CC2T_bChannelIdM6045#define CC2T_bSizePositionOfHandshakeM6047#define CC2T_bSizePositionOfHandshakeM6047#define CC2T_usConformanceClassM6048#define CC2T_usConformanceClassM6050#define CC2T_usConformanceClassM6050#define CC2T_usConformanceClassM6050#define CC2T_usConformanceClassM6051#define CC3T_usConformanceClassM6052#define CC3T_usConformanceClassM6051#define CC3T_bChannelTypeM6053#define CC3T_bChannelTaM6054#define CC3T_bSizePositionOfHandshakeM6055#define CC3T_bSizePositionOfHandshakeM6055#define CC3T_bSizePositionOfHandshakeM6055#define CC3T_bSizePositionOfHandshakeM6056#define CC3T_bSizePositionOfHandshakeM6056#define CC3T_bSizePositionOfHandshakeM6056#define CC3T_b			
#define CC01_ulSizeOfChannelM6033#define CC01_usCommunicationClassM6035#define CC01_usConformanceClassM6035#define CC01_usConformanceClassM6036#define CC11_bChannelTypeM6037#define CC11_bChannelTypeM6038#define CC11_bSizePositionOfHandshakeM6038#define CC11_ubSizeOfChannelM6040#define CC11_usConformanceClassM6041#define CC11_usConformanceClassM6042#define CC11_usConformanceClassM6044#define CC21_bChannelTypeM6045#define CC21_bChannelIdM6046#define CC21_bChannelIdM6047#define CC21_bChannelIdM6048#define CC21_bChannelIdM6048#define CC21_usConformanceClassM6048#define CC21_bChannelIdM6048#define CC21_usConformanceClassM6050#define CC21_usConformanceClassM6050#define CC21_usConformanceClassM6051#define CC21_usConformanceClassM6051#define CC31_usConformanceClassM6051#define CC31_usConformanceClassM6053#define CC31_bChannelTypeM6053#define CC31_bChannelIdM6054#define CC31_usConformanceClassM6056#define CC31_usConformanceClassM6056#define CC31_usConformanceClassM6056#define CC31_usConformanceClassM6056#define CC31_usConformanceClassM6056#define CC31_usConformanceClassM6056#define CC31_usConformanceClassM6056#define CC31_usConfor			
#defineCC0T_usCommunicationClassM6034#defineCC0T_usConformanceClassM6035#defineCC0T_usConformanceClassM6037#defineCC1T_bChannelTypeM6037#defineCC1T_bChannelIdM6038#defineCC1T_bSIzePositionOfHandshakeM6039#defineCC1T_uSizePositionOfHandshakeM6040#defineCC1T_uSizePositionOfHandshakeM6041#defineCC1T_uSizePositionClassM6042#defineCC1T_usConformanceClassM6043#defineCC2T_bChannelIdM6045#defineCC2T_bChannelIdM6046#defineCC2T_bChannelIdM6046#defineCC2T_bSizePositionOfHandshakeM6047#defineCC2T_usCommunicationClassM6048#defineCC2T_usCommunicationClassM6051#defineCC2T_usConformanceClassM6051#defineCC2T_usConformanceClassM6052#defineCC3T_bChannelTypeM6053#defineCC3T_bChannelIdM6054#defineCC3T_bChannelIdM6055#defineCC3T_uSizePositionOfHandshakeM6056#defineCC3T_uSizePositionOfHandshakeM6055#defineCC3T_uSizePositionOfHandshakeM6056#defineCC3T_uSizePositionOfHandshakeM6056#defineCC3T_uSizePositionOfHandshakeM6056#defineCC3T_uSizePositionOfHandshakeM6056#defineCC3T_uSizePositionOfHandshakeM6056#defineCC3T_uSizePosi		=	
#defineCC0T_usProtocolClassM6035#defineCC0T_usConformanceClassM6036#defineCC1T_bChannelTypeM6037#defineCC1T_bChannelTypeM6038#defineCC1T_bSizePositionOfHandshakeM6039#defineCC1T_ubSizePositionOfHandshakeM6040#defineCC1T_usCommunicationClassM6041#defineCC1T_usConformanceClassM6043#defineCC2T_bChannelTypeM6045#defineCC2T_bChannelTypeM6046#defineCC2T_bChannelTypeM6046#defineCC2T_bSizePositionOfHandshakeM6047#defineCC2T_usConformanceClassM6048#defineCC2T_usCommunicationClassM6048#defineCC2T_usCommunicationClassM6050#defineCC2T_usCommunicationClassM6051#defineCC2T_usConformanceClassM6051#defineCC3T_bChannelTypeM6053#defineCC3T_bChannelTypeM6055#defineCC3T_bSizePositionOfHandshakeM6055#defineCC3T_bSizePositionOfHandshakeM6055#defineCC3T_bSizePositionOfHandshakeM6055#defineCC3T_bSizePositionOfHandshakeM6055#defineCC3T_uSCommunicationClassM6056#defineCC3T_uSCommunicationClassM6057#defineCC3T_uSCommunicationClassM6058#defineCC3T_uSCommunicationClassM6058#defineCC3T_uSCommunicationClassM6059#defineCC3T_uSCom		=	
#defineCC0I_usConformanceClassM6036#defineCC1I_bChannelTypeM6037#defineCC1I_bSizePositionOfHandshakeM6038#defineCC1I_bSizePositionOfHandshakeM6049#defineCC1I_ulSizeOfChannelM6041#defineCC1I_usCommunicationClassM6042#defineCC1I_usConformanceClassM6043#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelTypeM6046#defineCC2I_usConformanceClassM6048#defineCC2I_usConformanceClassM6051#defineCC2I_usConformanceClassM6050#defineCC2I_usConformanceClassM6052#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bSizePositionOfHandshakeM6056#defineCC3I_bSizePositionOfHandshakeM6056#defineCC3I_bSizePositionOfHandshakeM6056 <tr< td=""><td>#define</td><td>CC01 usProtocolClass</td><td></td></tr<>	#define	CC01 usProtocolClass	
#defineCClI_bChannelTypeM6037#defineCClI_bSizePositionOfHandshakeM6038#defineCClI_bNumberOfBlocksM6040#defineCClI_ulSizeOfChannelM6041#defineCClI_usCommunicationClassM6042#defineCClI_usProtocolClassM6043#defineCClI_usConformanceClassM6044#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelTypeM6046#defineCC2I_bSizePositionOfHandshakeM6047#defineCC2I_bSizePositionOfHandshakeM6047#defineCC2I_usProtocolClassM6048#defineCC2I_usizePositionOfHandshakeM6050#defineCC2I_usProtocolClassM6051#defineCC2I_usProtocolClassM6051#defineCC2I_usProtocolClassM6051#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelTypeM6054#defineCC3I_usConformanceClassM6055#defineCC3I_usConformanceClassM6055#defineCC3I_usConformanceClassM6056#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6059#defineCC3I_usConformanceClassM6050#defineCC3I_usConformanceClassM6050#defineCC3I_usConformanceClassM6050#defineCC3I_usConformanceClassM6050#d	#define	CC0I_usConformanceClass	M6036
#define CC11_bSizePositionOfHandshakeM6039#define CC11_bNumberOfBlocksM6040#define CC11_usCommunicationClassM6042#define CC11_usCommunicationClassM6042#define CC11_usConformanceClassM6043#define CC21_bChannelTypeM6045#define CC21_bChannelIdM6046#define CC21_bSizePositionOfHandshakeM6047#define CC21_bNumberOfBlocksM6048#define CC21_usCommunicationClassM6050#define CC21_usCommunicationClassM6050#define CC21_usComformanceClassM6051#define CC21_usCommunicationClassM6052#define CC31_bChannelM6052#define CC31_bChannelIdM6052#define CC31_bChannelIdM6054#define CC31_bChannelIdM6054#define CC31_bChannelIdM6054#define CC31_bSizePositionOfHandshakeM6055#define CC31_bChannelIdM6054#define CC31_bSizePositionOfHandshakeM6055#define CC31_usCommunicationClassM6056#define CC31_usComformanceClassM6056#define CC31_usCommunicationClassM6058#define CC31_usCommunicationClassM6058#define CC31_usCommunicationClassM6059#define CC31_usConformanceClassM6059#define CC31_usConformanceClassM6060#define CC31_usConformanceClassM6059#define CC31_usConformanceClassM6059#define CC31_usConformanceClassM6060#define CC31_usConformanceClassM6060#define CC31_usConformanceClassM6	#define	CC11 bChannelType	M6037
#defineCCl1_bNumberOfBlocksM6040#defineCCl1_ulSizeOfChannelM6041#defineCCl1_usCommunicationClassM6042#defineCCl1_usConformanceClassM6043#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelIdM6046#defineCC2I_bSizePositionOfHandshakeM6047#defineCC2I_ulSizeOfChannelM6049#defineCC2I_ulSizeOfChannelM6050#defineCC2I_ulSizeOfChannelM6050#defineCC2I_usProtocolClassM6051#defineCC2I_usCommunicationClassM6052#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelIdM6055#defineCC3I_bChannelIdM6056#defineCC3I_usDitionOfHandshakeM6056#defineCC3I_usCommunicationClassM6056#defineCC3I_usCommunicationClassM6056#defineCC3I_usCommunicationClassM6056#defineCC3I_usCommunicationClassM6056#defineCC3I_usCommunicationClassM6058#defineCC3I_usCommunicationClassM6058#defineCC3I_usConformanceClassM6059#defineCC3I_usConformanceClassM6059#defineCC3I_usConformanceClassM6060#defineAC0I_bChannelTypeM6061#defineAC0I_bChannelTypeM6061	#define	CC11 bChannelId	M6038
#defineCC1I_ulSizeOfChannelM6041#defineCC1I_usCommunicationClassM6042#defineCC1I_usConformanceClassM6043#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelTypeM6046#defineCC2I_bChannelTofHandshakeM6047#defineCC2I_ulSizePositionOfHandshakeM6048#defineCC2I_ulSizeOfChannelM6049#defineCC2I_ulSizeOfChannelM6050#defineCC2I_usCommunicationClassM6050#defineCC2I_usConformanceClassM6051#defineCC2I_usConformanceClassM6052#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelTypeM6055#defineCC3I_bSizePositionOfHandshakeM6056#defineCC3I_usConformanceClassM6056#defineCC3I_usConformanceClassM6056#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6059#defineCC3I_usConformanceClassM6060#defineCC3I_usConformanceClassM6060#defineCC3I_usConformanceClassM6060#defineCC3I_usConformanceClassM6060#defineCC3I_usConformanceClassM6060#defineCC3I_usConformanceClassM6060#defineCC3I_usConformanceClassM6060 <td>#define</td> <td>CC1I bSizePositionOfHandshake</td> <td>M6039</td>	#define	CC1I bSizePositionOfHandshake	M6039
#defineCClT_usCommunicationClassM6042#defineCClI_usProtocolClassM6043#defineCClI_usConformanceClassM6044#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelIdM6046#defineCC2I_bSizePositionOfHandshakeM6047#defineCC2I_usDumberOfBlocksM6048#defineCC2I_usConformanceClassM6050#defineCC2I_usConformanceClassM6051#defineCC2I_usConformanceClassM6052#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelIdM6054#defineCC3I_bChannelIdM6055#defineCC3I_bChannelIdM6056#defineCC3I_usProstionOfHandshakeM6055#defineCC3I_usProstionOfHandshakeM6056#defineCC3I_usConformanceClassM6058#defineCC3I_usProsticionOfHandshakeM6056#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6058#defineCC3I_usConformanceClassM6061#defineCC3I_usConformanceClassM6061#defineCC3I_usConformanceClassM6061#defineCC3I_usConformanceClassM6061<	#define	CC1I bNumberOfBlocks	M6040
#defineCC1I_usProtocolClassM6043#defineCC1I_usConformanceClassM6044#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelIdM6046#defineCC2I_bSizePositionOfHandshakeM6047#defineCC2I_usDnuberOfBlocksM6048#defineCC2I_usCommunicationClassM6050#defineCC2I_usConformanceClassM6051#defineCC2I_usConformanceClassM6052#defineCC3I_bSizePositionOfHandshakeM6053#defineCC3I_bChannelIdM6054#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_usCommunicationClassM6056#defineCC3I_usIzeOfChannelM6057#defineCC3I_usIzeOfChannelM6058#defineCC3I_usIzeOfChannelM6058#defineCC3I_usCommunicationClassM6058#defineCC3I_usConformanceClassM6060#defineCC3I_usConformanceClassM6060#defineCC3I_usConformanceClassM6061#defineAC0I_bChannelIdM6062	#define	CC1I ulSizeOfChannel	M6041
#defineCC11_usConformanceClassM6044#defineCC21_bChannelTypeM6045#defineCC21_bChannelIdM6046#defineCC21_bSizePositionOfHandshakeM6047#defineCC21_bNumberOfBlocksM6048#defineCC21_ulSizeOfChannelM6049#defineCC21_usCommunicationClassM6050#defineCC21_usConformanceClassM6051#defineCC31_bChannelTypeM6053#defineCC31_bChannelIdM6055#defineCC31_bSizeOfChannelM6055#defineCC31_bSizeOfChannelM6056#defineCC31_usIzeOfChannelM6057#defineCC31_usIzeOfChannelM6058#defineCC31_usIzeOfChannelM6058#defineCC31_usIzeOfChannelM6058#defineCC31_usIzeOfChannelM6058#defineCC31_usIzeOfChannelM6058#defineCC31_usCommunicationClassM6058#defineCC31_usConformanceClassM6059#defineCC31_usConformanceClassM6061#defineCC31_usConformanceClassM6061#defineAC01_bChannelTypeM6061			M6042
#defineCC2I_bChannelTypeM6045#defineCC2I_bChannelIdM6046#defineCC2I_bSizePositionOfHandshakeM6047#defineCC2I_bNumberOfBlocksM6048#defineCC2I_ulSizeOfChannelM6049#defineCC2I_usCommunicationClassM6050#defineCC2I_usConformanceClassM6051#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelIdM6054#defineCC3I_bChannelIdM6055#defineCC3I_bSizePositionOfHandshakeM6056#defineCC3I_usCorformanceClassM6056#defineCC3I_usCorformanceTlassM6056#defineCC3I_usCorformanceTlassM6056#defineCC3I_usCorformanceTlassM6057#defineCC3I_usCorformanceClassM6058#defineCC3I_usCorformanceClassM6059#defineCC3I_usConformanceClassM6060#defineAC0I_bChannelTypeM6061#defineAC0I_bChannelTypeM6061			
#defineCC2I_bChannelIdM6046#defineCC2I_bSizePositionOfHandshakeM6047#defineCC2I_bNumberOfBlocksM6048#defineCC2I_ulSizeOfChannelM6049#defineCC2I_usCommunicationClassM6050#defineCC2I_usProtocolClassM6051#defineCC2I_usConformanceClassM6052#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelIdM6054#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_usizeOfChannelM6056#defineCC3I_usizeOfChannelM6057#defineCC3I_usizeOfChannelM6057#defineCC3I_usizeOfChannelM6058#defineCC3I_usizeOfChannelM6059#defineCC3I_usConformanceClassM6060#defineCC3I_usConformanceClassM6060#defineACOI_bChannelTypeM6061#defineACOI_bChannelTypeM6061			
#defineCC21_bSizePositionOfHandshakeM6047#defineCC21_bNumberOfBlocksM6048#defineCC21_ulSizeOfChannelM6049#defineCC21_usCommunicationClassM6050#defineCC21_usProtocolClassM6051#defineCC21_usConformanceClassM6052#defineCC31_bChannelTypeM6053#defineCC31_bChannelTypeM6054#defineCC31_bSizePositionOfHandshakeM6055#defineCC31_uSizeOfChannelM6057#defineCC31_usCommunicationClassM6058#defineCC31_usCommunicationClassM6059#defineCC31_usConformanceClassM6060#defineCC31_usConformanceClassM6060#defineCC31_usConformanceClassM6060#defineCC31_usConformanceClassM6060#defineAC01_bChannelTypeM6061#defineAC01_bChannelIdM6062			
#defineCC21_bNumberOfBlocksM6048#defineCC21_ulSizeOfChannelM6049#defineCC21_usCommunicationClassM6050#defineCC21_usProtocolClassM6051#defineCC21_usConformanceClassM6052#defineCC31_bChannelTypeM6053#defineCC31_bChannelIdM6054#defineCC31_bSizePositionOfHandshakeM6055#defineCC31_ulSizeOfChannelM6056#defineCC31_ulSizeOfChannelM6057#defineCC31_usCommunicationClassM6058#defineCC31_usProtocolClassM6059#defineCC31_usConformanceClassM6060#defineAC01_bChannelTypeM6061#defineAC01_bChannelIdM6062			
#defineCC2I_ulSizeOfChannelM6049#defineCC2I_usCommunicationClassM6050#defineCC2I_usProtocolClassM6051#defineCC2I_usConformanceClassM6052#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelIdM6054#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_ulSizeOfChannelM6056#defineCC3I_ulSizeOfChannelM6057#defineCC3I_usCommunicationClassM6058#defineCC3I_usProtocolClassM6059#defineCC3I_usConformanceClassM6060#defineAC0I_bChannelTypeM6061#defineAC0I_bChannelIdM6062			
#defineCC2I_usCommunicationClassM6050#defineCC2I_usProtocolClassM6051#defineCC2I_usConformanceClassM6052#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelIdM6054#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_usCommunicationClassM6056#defineCC3I_usCommunicationClassM6057#defineCC3I_usCommunicationClassM6058#defineCC3I_usProtocolClassM6059#defineCC3I_usConformanceClassM6060#defineAC0I_bChannelTypeM6061#defineAC0I_bChannelIdM6062			
#defineCC2I_usProtocolClassM6051#defineCC2I_usConformanceClassM6052#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelIdM6054#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_usIzeOfEllocksM6056#defineCC3I_usIzeOfChannelM6057#defineCC3I_usCommunicationClassM6058#defineCC3I_usProtocolClassM6059#defineCC3I_usConformanceClassM6060#defineAC0I_bChannelTypeM6061#defineAC0I_bChannelIdM6062			
#defineCC2I_usConformanceClassM6052#defineCC3I_bChannelTypeM6053#defineCC3I_bChannelIdM6054#defineCC3I_bSizePositionOfHandshakeM6055#defineCC3I_bNumberOfBlocksM6056#defineCC3I_ulSizeOfChannelM6057#defineCC3I_usCommunicationClassM6058#defineCC3I_usProtocolClassM6059#defineCC3I_usConformanceClassM6060#defineAC0I_bChannelTypeM6061#defineAC0I_bChannelIdM6062			
#defineCC31_bChannelTypeM6053#defineCC31_bChannelIdM6054#defineCC31_bSizePositionOfHandshakeM6055#defineCC31_bNumberOfBlocksM6056#defineCC31_uSizeOfChannelM6057#defineCC31_usCommunicationClassM6058#defineCC31_usProtocolClassM6059#defineCC31_usConformanceClassM6060#defineAC01_bChannelTypeM6061#defineAC01_bChannelIdM6062		_	
#defineCC31_bChannelIdM6054#defineCC31_bSizePositionOfHandshakeM6055#defineCC31_bNumberOfBlocksM6056#defineCC31_uSizeOfChannelM6057#defineCC31_usCommunicationClassM6058#defineCC31_usProtocolClassM6059#defineCC31_usConformanceClassM6060#defineAC01_bChannelTypeM6061#defineAC01_bChannelIdM6062		_	
#defineCC31_bSizePositionOfHandshakeM6055#defineCC31_bNumberOfBlocksM6056#defineCC31_ulSizeOfChannelM6057#defineCC31_usCommunicationClassM6058#defineCC31_usProtocolClassM6059#defineCC31_usConformanceClassM6060#defineAC01_bChannelTypeM6061#defineAC01_bChannelIdM6062			
#defineCC31_bNumberOfBlocksM6056#defineCC31_ulSizeOfChannelM6057#defineCC31_usCommunicationClassM6058#defineCC31_usProtocolClassM6059#defineCC31_usConformanceClassM6060#defineAC01_bChannelTypeM6061#defineAC01_bChannelIdM6062			
#defineCC3I_ulSizeOfChannelM6057#defineCC3I_usCommunicationClassM6058#defineCC3I_usProtocolClassM6059#defineCC3I_usConformanceClassM6060#defineAC0I_bChannelTypeM6061#defineAC0I_bChannelIdM6062		=	
#defineCC31_usCommunicationClassM6058#defineCC31_usProtocolClassM6059#defineCC31_usConformanceClassM6060#defineAC01_bChannelTypeM6061#defineAC01_bChannelIdM6062		=	
#define CC31_usProtocolClassM6059#define CC31_usConformanceClassM6060#define AC01_bChannelTypeM6061#define AC01_bChannelIdM6062		=	
#define CC31_usConformanceClassM6060#define AC01_bChannelTypeM6061#define AC01_bChannelIdM6062			
#define AC01_bChannelType M6061 #define AC01_bChannelId M6062		_	
#define AC0I_bChannelId M6062		=	
#deline ACUL psizePositionOtHandshake Mb063		ACOI bSizePositionOfHandshake	M6063

#define ACOI_bNumberOfBlocks	M6064
#define AC01 ulSizeOfChannel	M6065
#define AC1I bChannelType	M6066
#define AC1I bChannelId	M6067
<pre>#define AC1I_bSizePositionOfHandshake</pre>	M6068
<pre>#define AC11_bNumberOfBlocks #define AC11_ulSizeOfChannel #define SCtrl_ulSystemCommandCOS #define SStat_ulSystemCOS</pre>	M6069
#define AC1I ulSizeOfChannel	M6070
#define SCtrl ulSystemCommandCOS	M6071
<pre>#define Sott1_ulSystemCOS #define SStat_ulSystemCOS #define SStat_ulSystemError #define SStat_ulBootError #define SStat_ulBootError #define SStat_ulTimeSinceStart</pre>	M6072
#define sstat_disystemeos	
#define SStat_ulSystemStatus	M6073
#define SStat ulSystemError	M6074
#define SStat ulBootError	M6075
#define SStat_ulTimeSinceStart	M6076
#define SStat usCpuLoad	M6077
<pre>#define SStat_usCpuLoad #define SStat_ulHWFeatures</pre>	
#define SStat_ulHWFeatures	M6078
#define SSMB_usPackagesAccepted	M6079
#define SSMB_ulDest	M6080
#define SSMB ulSrc	M6081
#define SSMB_ulDestId	M6082
#define SSMB_ulSrcId	M6083
#define SSMB ulLen	M6084
#define SSMB_ulId	M6085
#define SSMB_ulState	M6086
#define SSMB_ulCmd	M6087
#define SSMB_ulExt	M6088
#define SSMB_ulRout	M6089
#define SSMB_ultData0	M6090
#define SSMB ultData1	M6091
#define SSMB_ultData2	M6091 M6092
#define SSMB_ultData3	M6093
#define SSMB ultData4	M6094
#define SSMB_ultData5	M6095
#define SSMB ultData6	M6096
-	
#define SSMB_ultData7	M6097
#define SSMB_ultData8	M6098
#define SSMB ultData9	M6099
#define SSMB_ultData10	M6100
#define SSMB ultData11	M6101
=	
#define SSMB_ultData12	M6102
#define SSMB_ultData13	M6103
#define SSMB_ultData14	M6104
#define SSMB ultData15	M6105
#define SSMB_ultData16	M6105 M6106
#define SSMB_ultData17	M6107
#define SSMB ultData18	M6108
#define SSMB_ultData19	M6109
#define SSMB_ultData20	M6110
#define SRMB_usWaitingPackages	
	M6111
#define SRMB_ulDest	M6112
#define SRMB_ulSrc	M6113
#define SRMB_ulDestId	M6114
#define SRMB ulSrcId	M6115
#define SRMB_ullen	M6116
#define SRMB_ulId	M6117
#define SRMB ulState	M6118
#define SRMB_ulCmd	M6119
#define SRMB ulExt	M6120
#define SRMB_ulRout	M6121
#define SRMB_ultData0	M6122
#define SRMB ultData1	M6123
#define SRMB_ultData2	M6124
#define SRMB ultData3	M6125
#define SRMB_ultData4	M6126
#define SRMB_ultData5	M6127
#define SRMB_ultData6	M6128
#define SRMB_ultData7	M6129
#define SRMB_ultData8	
-	M6130
#define SRMB_ultData9	M6131
#define SRMB ultData10	M6132
#define SRMB_ultData11	M6133
#define SRMB ultData12	M6134
"dettile pittip_dtcbdcd12	110101

detime SNB: ulbetal3 M6135 Addime SNB: ulbetal5 M6137 Addime SNB: ulbetal5 M6137 Addime SNB: ulbetal5 M6137 Addime SNB: ulbetal5 M6137 Addime SNB: ulbetal5 M6147 Addime SNB: ulbetal3 M6142 Addime SNB: ulbetal3 M6143 Addime SNB: SNB: SNB: SNB: SNB: SNB: SNB: SNB:			
Hedrin SHKB_lithatal3 M6137 Hedrin SHKB_lithatal7 M6138 Hedrin SHKB_lithatal7 M6139 Hedrin SHKB_lithatal7 M6139 Hedrin SHKB_lithatal7 M6140 Hedrin SHKB_lithatal8 M6140 Hedrin SKC_NSF PRADY M6141 Hedrin MCC_NSF PRADY M6143 Hedrin MCC_NSF PRADY M6144 Hedrin MCC_NSF PRADY M6144 Hedrin MCC_NSF PRADY M6145 Hedrin MCC_NSF PRADY M6144 Hedrin MCC_NSF PRADY M6147 Hedrin MCC_NSF PRADY M61347 Hedrin MCC_NSF_NDF_MAX_CK M6134 Hedrin MCC_NSF_NDF_MAX_CK M6134 Hedrin MCC_NSF_NDF_MAX_CK M6135 Hedrin MCCO_NSF_NDF_MAX_CK M6135 Hedrin MCCO_NSF PRADY M6137 Hedrin MCCO_NSF PRADY M6136 Hedrin MCCO_NSF PRADY M6137		=	M6135
define SNME_lithcal6 M6139 idefine SNME_lithcal8 M6140 idefine SNME_lithcal9 M6141 idefine SNME_lithcal9 M6141 idefine SNME_lithcal9 M6141 idefine SNME_lithcal9 M6141 idefine SNME_lithcal9 M6143 idefine MCSC_NSF_NSTR_COS_ACK M6145 idefine MCSC_NSF_NSTR_COS_ACK M6146 idefine MCSC_NSF_SND_NSK_COS M6147 idefine MCSC_NSF_SND_NSK_COS M6140 idefine MCSC_NSF_SND_NSK_COS M6130 idefine MCSC_NSF_SND_NSK_COS M6130 idefine MCSC_NSF_SND_NSK_COS M6130 idefine MCSC_NSF_SND_NSK_COS M6131 idefine MCSC_NSF_SND_NSK_COS M	#define	SRMB ultData14	M6136
define SNAP 1111111 M6139 idefine SNAP 11111111 M6140 idefine SNAP 11111111 M6141 idefine SNAP 111111111 M6141 idefine SNAP M6143 M6143 idefine HCRC_NSY_PARON M6144 idefine HCRC_NSY_PARON M6144 idefine HCRC_NSY_PARON M6146 idefine HCRC_NSY_PARON M6146 idefine HCRC_NSY_PARON M6146 idefine HCRC_NSY_PARON M6147 idefine HCRC_NSY_PARON M6148 idefine HCRC_NSY_PARON M6151 idefine HCRC_NSY_PARON M6152 idefine HCRC_NSY_PARON M6153 idefine HCRCC_NSY_PARON M6153	#define	SRMB ultData15	M6137
#define SNML_libials M6140 #define SNML_libials M6141 #define SNML_libials M6143 #define HCSC_NAF_ERROR M6143 #define HCSC_NAF_ERROR M6144 #define HCSC_NAF_ERROR M6144 #define HCSC_NAF_ERROR M6145 #define HCSC_NAF_ERROR M6146 #define HCSC_NAF_ERROR M6146 #define HCSC_NAF_ERROR M6146 #define HCSC_NAF_ERROR M6150 #define HCSC_NAF_ERROR M6151 #define HCSC_NAF_ERROR M6151 #define HCSC_NAF_ERROR M6151 #define HCSC_NAF_ERROR M6151 #define HCSC_NAF_ERROR M6153 #define HCSC_NAF_ERROR M6154 #define HCSCC_NAF_ERROR M6154	#define	SRMB ultData16	M6138
Hedrine SXMB_litData19 M6142 Hofline SXMS_litData10 M6142 Hedrine HCSC_NNS_HKADY M6143 Hedrine HCSC_NNS_HKADY M6144 Hedrine HCSC_NNS_HKADY M6151 Hedrine HCSC_NNS_HKADY M6153 Hedrine HCSC_NNS_HKADY M6154 Hedrine HCSC NON_DENDARC M6154 Hedrine HCSC NON_DENDARC M6154 Hedrine HCSC NON_PENDON_DENARC M6154 Hedrine HCSC NON_PENDON_DENARC M6163 Hedrine HCSC NON_PENDON_DENARC M6164 Hedrine HCSC NON_PENDON_DENARC M6164 Hedrine HCSC NON_PE	#define	SRMB_ultData17	M6139
Hedrine SXMB_litData19 M6142 Hofline SXMS_litData10 M6142 Hedrine HCSC_NNS_HKADY M6143 Hedrine HCSC_NNS_HKADY M6144 Hedrine HCSC_NNS_HKADY M6151 Hedrine HCSC_NNS_HKADY M6153 Hedrine HCSC_NNS_HKADY M6154 Hedrine HCSC NON_DENDARC M6154 Hedrine HCSC NON_DENDARC M6154 Hedrine HCSC NON_PENDON_DENARC M6154 Hedrine HCSC NON_PENDON_DENARC M6163 Hedrine HCSC NON_PENDON_DENARC M6164 Hedrine HCSC NON_PENDON_DENARC M6164 Hedrine HCSC NON_PE		_	
Hedrine Stad Stad Vedrine HCSC Net Karlags M6143 Vedrine HCSC NST ERSOR M6144 Vedrine HCSC NST ERSOR M6145 Vedrine HCSC NST ERSOR M6146 Vedrine HCSC NST ERSOR M6146 Vedrine HCSC NST ERSOR M6146 Vedrine HCSC NST NST MNX ACK M6146 Vedrine HCSC NST NST MNX ACK M6146 Vedrine HCSC NST NST MNX ACK M6151 Vedrine HCSC NST NOTSTATT M6152 Vedrine HCSC NST NOTSTATT M6153 Vedrine HCSC NST NOTSTATT M6154 Vedrine HCSC NST NOTSTATT M6155 Vedrine HCSC NST NOTSTATT M6156 Vedrine HCSC NST NOTSTATT M6157 Vedrine HCSC NST NOTSTATT M6158 Vedrine HCSC NST NOTSTATT M6156 Vedr			
#define HCSC_NST_REATY N6143 #define HCSC_NST_REATY N6144 #define HCSC_NST_REATY N6145 #define HCSC_NST_REATY N6146 #define HCSC_NST_REATY N6146 #define HCSC_NST_REATY N6149 #define HCSC_NST_REATY N6150 #define HCSC_NST_REATY N6151 #define HCSC_NST_REATY N6153 #define HCSC_NST_REATY N6153 #define HCSC_NST_REATY N6153 #define HCSC_NST_REATY N6154 #define HCSC_NST_REATY N6153 #define HCSC_NST_REATY N6154 #define HCSC_NST_REATY N6155 #define HCSCO_NTATY N6154 #define HCSCO_NTATY N6155 #define HCSCO_NTATY N6156 #define HCSCO_NTATY N6156 #define HCSCO_NTATY N6156 #define HCSCO_NTATY N61610 #define HCSCO_NTATY N61610 #define HCSCO_NTATY N61610 #define HCSCO_NTATY N61610 #define HCSCO_NTATY N61610 <t< td=""><td></td><td></td><td></td></t<>			
Hedine HCG_NST_ERROY M6145 Hedine HCGS_NST_ERROY M6145 Hedine HCGS_NST_ERROY M6146 Hedine HCGS_NST_RENCY M6146 Hedine HCGS_NST_RENCY M6146 Hedine HCGS_NST_RENCY M6146 Hedine HCGS_NST_RENCY M6150 Hedine HCGS_NST_RENCY M6151 Hedine HCGS_NST_RENCY M6152 Hedine HCGS_NST_RENCY M6153 Hedine HCGS_NST_RENCY M6153 Hedine HCGS_NST_RENCY M6154 Hedine HCGS_NST_RENCY M6153 Hedine HCGS_NST_RENCY M6154 Hedine HCGS_NST_RENCY M6153 Hedine HCGS_NST_RENCY M6154 Hedine HCGS_NST_RENCY M6157 Hedine HCGS_NST_RENCY M6158 Hedine HCGS_NST_RENCY M6161 Hedine HCGS_NST_RENCY M6161 Hedine HCGS_NST_RENCY M6163 Hedine HCGS_NST_RENCY M6166 Hedine HCGS_NST_RENCY M6166 Hedine HCGS_NST_RENCY M6167 Hedine HCGS_NST_RENCY M6167 <td< td=""><td></td><td></td><td></td></td<>			
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#deline	HCSC_DNetXFlags	
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#derine	HCSC_NSF_READY	M6144
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCSC_NSF_ERROR	M6145
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCSC_NSF_HOST_COS_ACK	M6146
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCSC_NSF_NETX_COS_CMD	M6147
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCSC_NSF_SEND_MBX_ACK	M6148
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCSC NSF RECV MBX CMD	M6149
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCSC bHostFlags	M6150
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCSC HSF RESET	M6151
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCSC HSF BOOTSTART	M6152
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCSC HSF HOST COS CMD	M6153
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#dofino	HCSC_HSE_NETY_COS_ACK	M6153
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#deline	HCSC_HSF_NEIX_COS_ACK	MOIJ4
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#uerine	NCOC HOE DECK NOV JON	MC1EC
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#aetine	HUSC_HSF_RECV_MBX_ACK	M0120
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCCCU_usNetxFlags	M0127
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCCCU_NCF_COMMUNICATING	M6158
#define HCCC0_NCT_HOST_COS_ACK M6160 #define HCCC0_NCT_SEND_MEX_ACK M6162 #define HCCC0_NCT_FRO_OUT_ACK M6163 #define HCCC0_NCT_PD0_OUT_ACK M6164 #define HCCC0_NCT_PD1_IN_CMD M6165 #define HCCC0_NCT_PD1_IN_CMD M6166 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6167 #define HCCC0_NCT_PD1_IN_CMD M6169 #define HCCC0_NCT_PD1_IN_CMD M6170 #define HCCC0_NCT_PD1_IN_CMD M6171 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC0_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_PD1_IN_ACK M6174 #define HCCC1_NCT_COMMUNCATING M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCCC1_NCT_NCT_NCN M6174 #define HCC1_NC	#define	HCCC0_NCF_ERROR	M6159
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCC0_NCF_HOST_COS_ACK	M6160
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCC0_NCF_NETX_COS_CMD	M6161
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCC0 NCF SEND MBX ACK	M6162
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCC0 NCF RECV MBX CMD	M6163
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCCO NCE PDO OUT ACK	M6164
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCCO NCE PDO IN CMD	M6165
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#dofino	HCCCO NCE DD1 OUT ACK	M6166
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#deline	HCCCO_NCF_PDI_OUI_ACK	MOLOO
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#deline	HCCCO_NCF_PDI_IN_CMD	MOLO/
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCC0_usHostFlags	M6168
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCCU_HCF_HOST_COS_CMD	M6169
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCC0_HCF_NETX_COS_ACK	M6170
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCC0_HCF_SEND_MBX_CMD	M6171
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCCO HCF RECV MBX ACK	M6172
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCCO HCF PDO OUT CMD	M6173
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCCO HCF PDO IN ACK	M6174
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCCO HCF PD1 OUT CMD	M6175
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCCO HCF PD1 IN ACK	M6176
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCC1 usNetyFlags	M6177
#defineHCCC1_NCF_ERRORM6179#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_FD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_OUT_ACKM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_UN_CMDM6187#defineHCCC1_UNF_PD1_IN_CMDM6187#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_PD0_OUT_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC1_HCF_PD1_N_ACKM6196#defineHCCC2_UNF_FOMMUNICATINGM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_END_MBX_ACKM6203#defineHCCC2_NCF_ERON_MBX_ACKM6203#defineHCCC2_NCF_END_MBX_ACKM6204	#define	HCCC1 NCE COMMUNICATING	M6179
#defineHCCC1_NCF_HOST_COS_ACKM6180#defineHCCC1_NCF_NETX_COS_CMDM6181#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_RECV_MBX_CMDM6183#defineHCCC1_NCF_PD0_OUT_ACKM6184#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_OUT_ACKM6187#defineHCCC1_USF_PD1_IN_CMDM6187#defineHCCC1_USF_PD1_OUT_ACKM6188#defineHCCC1_USF_PD1_OUT_ACKM6189#defineHCCC1_USF_PD1_COS_ACKM6190#defineHCCC1_HCF_NETX_COS_ACKM6191#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_USFEXFLagsM6197#defineHCCC2_USFEXFLAGSM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_RERORM6199#defineHCCC2_NCF_RERORM6199#defineHCCC2_NCF_REROS_CMDM6201#defineHCCC2_NCF_NETX_COS_CMDM6203#defineHCCC2_NCF_REN_MBX_ACKM6204	#deltue	HCCCI_NCF_COMMUNICATING	MO170
#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_RECV_MBX_CMDM6183#defineHCCC1_NCF_PD0_IN_CMDM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_IN_CMDM6187#defineHCCC1_HCF_BDT_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_PD0_IN_ACKM6192#defineHCCC1_HCF_PD0_IN_ACKM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_NCF_COMMUNICATINGM6199#defineHCCC2_NCF_ERCRM6199#defineHCCC2_NCF_ERCRM6199#defineHCCC2_NCF_ERCRM6199#defineHCCC2_NCF_ERCRM6190#defineHCCC2_NCF_ERCRM6200#defineHCCC2_NCF_ERCR_MBX_CMDM6201#defineHCCC2_NCF_ERCY_MBX_CMDM6203#defineHCC2_NCF_ERCY_MBX_CMDM6204			
#defineHCCC1_NCF_SEND_MBX_ACKM6182#defineHCCC1_NCF_RECV_MBX_CMDM6183#defineHCCC1_NCF_PD0_IN_CMDM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_IN_CMDM6187#defineHCCC1_HCF_BDT_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_PD0_IN_ACKM6192#defineHCCC1_HCF_PD0_IN_ACKM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_NCF_COMMUNICATINGM6199#defineHCCC2_NCF_ERCRM6199#defineHCCC2_NCF_ERCRM6199#defineHCCC2_NCF_ERCRM6199#defineHCCC2_NCF_ERCRM6190#defineHCCC2_NCF_ERCRM6200#defineHCCC2_NCF_ERCR_MBX_CMDM6201#defineHCCC2_NCF_ERCY_MBX_CMDM6203#defineHCC2_NCF_ERCY_MBX_CMDM6204	#define	HCCC1_NCF_HOST_COS_ACK	
WatchineHCCC1_NCF_PD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_IN_CMDM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_IN_CMDM6187#defineHCCC1_USHOSTFlagsM6188#defineHCCC1_HCF_NETX_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_RECV_MBX_CMDM6191#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_IN_ACKM6193#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_USNETXFlagsM6197#defineHCCC2_NCF_CERRORM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6190#defineHCCC2_NCF_SEND_MBX_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_SEND_MBX_ACKM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204	#define	HCCC1_NCF_NETX_COS_CMD	M6181
WatchineHCCC1_NCF_PD0_OUT_ACKM6183#defineHCCC1_NCF_PD0_IN_CMDM6185#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_IN_CMDM6187#defineHCCC1_USHOSTFlagsM6188#defineHCCC1_HCF_NETX_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_RECV_MBX_CMDM6191#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_IN_ACKM6193#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_USNETXFlagsM6197#defineHCCC2_NCF_CERRORM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6190#defineHCCC2_NCF_SEND_MBX_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_SEND_MBX_ACKM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204	#define	HCCC1_NCF_SEND_MBX_ACK	M6182
#defineHCCC1_NCF_PD0_OUT_ACKM6184#defineHCCC1_NCF_PD1_OUT_ACKM6185#defineHCCC1_NCF_PD1_IN_CMDM6187#defineHCCC1_USHOSTFlagsM6188#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6192#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6196#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_USNetxFlagsM6197#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_ERRORM6200#defineHCCC2_NCF_ERRORM6201#defineHCCC2_NCF_SEND_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204	" actine	necet_net_imev_imm_onb	M6183
#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_IN_CMDM6187#defineHCCC1_USHOSTFLagsM6188#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD1_OUT_CMDM6193#defineHCCC1_HCF_PD1_OUT_CMDM6196#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_USNETXFLagsM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_MOST_COS_ACKM6200#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_RECV_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCC2_NCF_PD0_OUT_ACKM6204			M6184
#defineHCCC1_NCF_PD1_OUT_ACKM6186#defineHCCC1_NCF_PD1_IN_CMDM6187#defineHCCC1_USHOSTFLagsM6188#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD1_OUT_CMDM6193#defineHCCC1_HCF_PD1_OUT_CMDM6196#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_USNETXFLagsM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_MOST_COS_ACKM6200#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_RECV_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCC2_NCF_PD0_OUT_ACKM6204	#define	HCCC1 NCF PD0 IN CMD	M6185
#defineHCCC1_NCF_PD1_IN_CMDM6187#defineHCCC1_usHostFlagsM6188#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCC2_usNetxFlagsM6197#defineHCC2_NCF_COMMUNICATINGM6198#defineHCC2_NCF_CS_ACKM6200#defineHCC2_NCF_NETX_COS_ACKM6201#defineHCC2_NCF_NETX_COS_CMDM6201#defineHCC2_NCF_NETX_COS_CMDM6203#defineHCC2_NCF_RECV_MBX_CMDM6203#defineHCC2_NCF_PD_OUT_ACKM6204			
#defineHCCC1_usHostFlagsM6188#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6194#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_usNetxFlagsM6197#defineHCC2_NCF_COMMUNICATINGM6198#defineHCC2_NCF_ERRORM6199#defineHCC2_NCF_ERSD_MBX_ACKM6200#defineHCC2_NCF_SEND_MBX_ACKM6202#defineHCC2_NCF_SEND_MBX_ACKM6203#defineHCC2_NCF_RECV_MBX_CMDM6203#defineHCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC1_HCF_HOST_COS_CMDM6189#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6194#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_USNetxFlagsM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCC2_NCF_ERRORM6199#defineHCC2_NCF_INETX_COS_CMDM6200#defineHCC2_NCF_SEND_MBX_ACKM6202#defineHCC2_NCF_SEND_MBX_ACKM6202#defineHCC2_NCF_RECV_MBX_CMDM6203#defineHCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC1_HCF_NETX_COS_ACKM6190#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6194#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCC2_UNVEXFLagsM6197#defineHCC2_NCF_COMMUNICATINGM6198#defineHCC2_NCF_ERRORM6199#defineHCC2_NCF_HOST_COS_ACKM6200#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCC2_NCF_SEND_MBX_ACKM6203#defineHCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC1_HCF_SEND_MBX_CMDM6191#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6194#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCC2_usNetxFlagsM6197#defineHCC2_NCF_COMMUNICATINGM6198#defineHCC2_NCF_ERRORM6199#defineHCC2_NCF_HOST_COS_ACKM6200#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC1_HCF_RECV_MBX_ACKM6192#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6194#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCC2_uNEtxFlagsM6197#defineHCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCC2_NCF_HOST_COS_ACKM6200#defineHCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC1_HCF_PD0_OUT_CMDM6193#defineHCCC1_HCF_PD0_IN_ACKM6194#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_usNetxFlagsM6197#defineHCC2_NCF_COMMUNICATINGM6198#defineHCC2_NCF_ERRORM6199#defineHCC2_NCF_MOST_COS_ACKM6200#defineHCC2_NCF_NETX_COS_CMDM6201#defineHCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC1_HCF_PD0_IN_ACKM6194#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_usNetxFlagsM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCC2_NCF_ERRORM6199#defineHCCC2_NCF_HOST_COS_ACKM6200#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC1_HCF_PD1_OUT_CMDM6195#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_usNetxFlagsM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_HOST_COS_ACKM6200#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC1_HCF_PD1_IN_ACKM6196#defineHCCC2_usNetxFlagsM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCC2_NCF_ERRORM6199#defineHCC2_NCF_HOST_COS_ACKM6200#defineHCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC2_usNetxFlagsM6197#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_HOST_COS_ACKM6200#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_HOST_COS_ACKM6200#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204	#define	HCCC1_HCF_PD1_IN ACK	M6196
#defineHCCC2_NCF_COMMUNICATINGM6198#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_HOST_COS_ACKM6200#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204	#define	HCCC2 usNetxFlags	M6197
#defineHCCC2_NCF_ERRORM6199#defineHCCC2_NCF_HOST_COS_ACKM6200#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC2_NCF_HOST_COS_ACKM6200#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC2_NCF_NETX_COS_CMDM6201#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204			
#defineHCCC2_NCF_SEND_MBX_ACKM6202#defineHCCC2_NCF_RECV_MBX_CMDM6203#defineHCCC2_NCF_PD0_OUT_ACKM6204			
<pre>#define HCCC2_NCF_RECV_MBX_CMD M6203 #define HCCC2_NCF_PD0_OUT_ACK M6204</pre>			
#define HCCC2_NCF_PD0_OUT_ACK M6204			
#define HCCC2_NCF_PD0_IN_CMD M6205			
	#define	HCCC2_NCF_PD0_IN_CMD	M6205

#define	HCCC2 NCF PD1 OUT ACK	M6206
#define	HCCC2 NCF PD1 IN CMD	M6207
#define	HCCC2 usHostFlags	M6208
#derine	HCCC2_USHOSEFIAGS	
#dellne	HCCC2_HCF_HOST_COS_CMD	M6209
#define	HCCC2_HCF_NETX_COS_ACK	M6210
#define	HCCC2 HCF SEND MBX CMD	M6211
#define	HCCC2 HCF RECV MBX ACK	M6212
#define	HCCC2 HCF PDO OUT CMD	M6213
#define		
#dellne	HCCC2_HCF_PDU_IN_ACK	M6214
#define	HCCC2_HCF_PD1_OUT_CMD	M6215
#define	HCCC2 HCF PD1 IN ACK	M6216
#define	HCCC3_usNetxFlags	M6217
#define	HCCC3 NCE COMMUNICATING	M6218
#derine	HCCCS_NCF_COPHONICATING	
#dellne	HCCC3_NCF_ERROR	M6219
#define	HCCC3_NCF_HOST_COS_ACK	M6220
#define	HCCC3 NCF NETX COS CMD	M6221
#define	HCCC3 NCF SEND MBX ACK	M6222
#define	HCCC3 NCE BECV MBX CMD	M6223
# define		MO223
#ueline	HCCCS_NCF_PDU_OUT_ACK	M0224
#define	HCCC3_NCF_PD0_IN_CMD	M6225
#define	HCCC3_NCF PD1 OUT ACK	M6226
#define	HCCC3 NCF PD1 IN CMD	M6227
#define	HCCC3 usHostFlags	M6228
#dofine	TOOCS THE TOSE COS CMP	M6220
#deline		MOZZS
#define	HCCC3_HCF_NETX_COS_ACK	M6230
#define	HCCC3_HCF_SEND_MBX CMD	M6231
#define	HCCC3 HCF RECV MBX ACK	M6232
#define	HCCC3 HCF PD0 OUT CMD	M6233
#dof:~~	HCCC3 HCF DDO IN ACK	M6234
#dertie	HOOD HOP DD1 ONT OND	MUZJA MCODE
#detine	HCCC3_HCF_PD1_OUT_CMD	M6235
#define	HCCC3_HCF_PD1_IN_ACK	M6236
#define	HCAC0 usNetxFlags	M6237
#define	HCACO NCE COMMUNICATING	M6238
#dofino	UCACO NCE EDBOD	M6230
#dertile	HCCC2_NCF_PD1_OUT_ACK HCCC2_NCF_PD1_IN_CMD HCCC2_usHostFlags HCCC2_HCF_HOST_COS_CMD HCCC2_HCF_NETX_COS_ACK HCCC2_HCF_RECV_MBX_ACK HCCC2_HCF_PD0_OUT_CMD HCCC2_HCF_PD1_OUT_CMD HCCC2_HCF_PD1_OUT_CMD HCCC2_HCF_PD1_IN_ACK HCCC3_USNetxFlags HCCC3_NCF_ERROR HCCC3_NCF_ERROR HCCC3_NCF_ERROR HCCC3_NCF_SEND_MBX_ACK HCCC3_NCF_RECV_MBX_ACK HCCC3_NCF_RECV_MBX_ACK HCCC3_NCF_PD0_OUT_ACK HCCC3_NCF_PD0_IN_CMD HCCC3_NCF_PD1_IN_CMD HCCC3_NCF_PD1_IN_CMD HCCC3_NCF_PD1_IN_CMD HCCC3_NCF_PD1_IN_CMD HCCC3_HCF_PD1_IN_CMD HCCC3_HCF_NETX_COS_ACK HCCC3_HCF_PD1_IN_CMD HCCC3_HCF_NETX_COS_ACK HCCC3_HCF_PD1_IN_CMD HCCC3_HCF_NETX_COS_ACK HCCC3_HCF_PD1_IN_CMD HCCC3_HCF_PD0_IN_ACK HCCC3_HCF_PD0_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC3_HCF_PD1_IN_ACK HCCC0_NCF_ERROR HCAC0_NCF_ERROR HCAC0_NCF_FON_MBX_ACK HCAC0_NCF_PD0_IN_ACK HCAC0_NCF_PD0_IN_ACK HCAC0_NCF_PD1_IN_CMD HCAC0_HCF_PD0_IN_CMD HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK HCAC0_HCF_PD0_IN_ACK	P102.3.3
#deline	nCACU_NCF_HOST_COS_ACK	
#define	HCACU_NCF_NETX_COS_CMD	M6241
#define	HCACO_NCF_SEND MBX ACK	M6242
#define	HCACO NCF RECV MBX CMD	M6243
#define	HCACO NCF PDO OUT ACK	M6244
#dofine	HCACO NCE DDO IN OMD	M6245
#dellie	HONGO NOR DRI OWE - FT	MOZ4J
#detine	HCACU_NCF_PD1_OUT_ACK	M6246
#define	HCAC0_NCF_PD1_IN_CMD	M6247
#define	HCAC0 usHostFlags	M6248
#define	HCACO HCF HOST COS CMD	M6249
#dofine	HCACO HCE NETY COS ACK	 M6250
#dertile	HONOO HOE OEND YOY OND	MC0E1
#define	HCACU_HCF_SEND_MBX_CMD	
#define	HCAC0_HCF_RECV_MBX_ACK	M6252
#define	HCACO HCF PDO OUT CMD	M6253
#define	HCACO_HCF_PD0_OUT_CMD HCACO_HCF_PD0_IN_ACK	M6254
#define	HCACO HCF PD1 OUT CMD	M6255
	HCACO HCF PD1 IN ACK	
		M6256
	HCAC1_usNetxFlags	M6257
	HCAC1_NCF_COMMUNICATING	M6258
#define	HCAC1 NCF ERROR	M6259
#define	HCAC1 NCF HOST COS ACK	M6260
	HCAC1 NCF NETX COS CMD	M6261
	HCAC1_NCF_SEND_MBX_ACK	M6262
	HCAC1_NCF_RECV_MBX_CMD	M6263
#define	HCAC1_NCF_PD0_OUT_ACK	M6264
#define	HCAC1 NCF PD0 IN CMD	M6265
	HCAC1 NCF PD1 OUT ACK	M6266
	HCAC1 NCF PD1 IN CMD	M6267
	HCAC1_usHostFlags	M6268
	HCAC1_HCF_HOST_COS_CMD	M6269
#define	HCAC1 HCF NETX COS ACK	M6270
	HCAC1 HCF SEND MBX CMD	M6271
	HCAC1 HCF RECV MBX ACK	M6272
	HCAC1_HCF_PD0_OUT_CMD	M6273
	HCAC1_HCF_PD0_IN_ACK	M6274
#define	HCAC1 HCF PD1 OUT CMD	M6275
	HCAC1 HCF PD1 IN ACK	M6276

AddIne CCD_NCX_APP_COS_APP_PRAY M6273 AddIne CCD_NCX_APP_COS_DOS_COS_MARK M6273 AddIne CCD_NCX_APP_COS_DOS_COS_MARK M6273 AddIne CCD_NCX_APP_COS_DOS_COS_COS_COS M6273 Machine CCD_NCX_APP_COS_DOX_COS_COS M6274 Adefine CCD_NCX_APP_COS_DOX_COS_COS M6283 Adefine CCD_NCX_APP_COS_DOX_COS_COS M6284 Adefine CCD_NCX_APP_COS_DOX_COS_COS M6284 Adefine CCD_NCX_APP_COS_DOX_COS_COS M6284 Adefine CCD_NCX_COS_COS M6284 Adefine CCD_NCX_COS_COS M6284 Adefine CCD_NCX_COS M6284 Adefine CCD_NCX_COS M6284 Adefine CCD_NCX_COS M6281 Adefine CCD_NCX_COS M6281 Adefine CCD_NCX_COS M6281 Adefine CCD_NCX_COS M6281 Adefine CCD_NCX_COS M6293 Adefine CCD_NCX_COS M6293 Adefine CCD_NCX_COS M6301 Adefine CCD_NCX_COS M6301 Adefine CCD_NCX_COS M6301 Adefine CCD_NCX_COS M6301 Adefine CCD_NCX_COS M6301 <th></th> <th></th> <th></th>			
define CCD_RCX_APT_COS_BUS_ON_ENABLE M6279 define CCD_RCX_APT_COS_INIT M6280 define CCD_RCX_APT_COS_INIT_RABLE M6281 define CCD_RCX_APT_COS_INIT_RABLE M6281 define CCD_RCX_APT_COS_INIT_RABLE M6282 define CCD_RCX_APT_COS_INIT_RABLE M6283 define CCD_RCX_APT_COS_INIT_RABLE M6283 define CCD_RCX_COM_COS_RUNNAR M6283 define CCD_RCX_COM_COS_RUNN M6283 define CCD_RCX_COM_COS_RUN M6283 define CCD_RCX_COM_COS_RUN M6283 define CCD_RCX_COM_COS_RUNNA M6283 define CCD_RCX_COM_COS_RUNNA M6283 define CCD_RCX_COM_COS_RUNNA M6284 define CCD_URX_COM_COS_RUNNA M6284 define CCD_URX_COM_COS_RUNNA M6284 define CCD_URX_COM_COS_RUNNA M6294 define CCD_URX_COM_COS_RUNNA M6294 define CCD_URX_COM_COS_RUNNA M6294 define CCD_URX_COM_COS_RUNNA M6294 define CCD_URANCAMA M6294 define CCD_URANCAMA M6294 define CCD_URANCAMA M6294 <t< td=""><td>#define</td><td>CC0_RCX_APP_COS_APP_READY</td><td>M6277</td></t<>	#define	CC0_RCX_APP_COS_APP_READY	M6277
Hedrine CCD RCX APP COS TNT F M6280 Addrine CCD RCX APP COS TNT EXABLE M6281 Addrine CCD RCX APP COS TAMA M6283 Addrine CCD RCX APP COS TAMA M6284 Addrine CCD RCX APP COS TAMA M6284 Addrine CCD RCX APP COS TAMA M6284 Addrine CCD RCX CAPP COS TAMA M6284 Addrine CCD RCX CAPP COS TAMA M6284 Addrine CCD RCX COMM COS FREDY M6284 Addrine CCD RCX COMM COS CONTE TO COKTD M6290 Addrine CCD RCX COMM COS CONTE TO COKTD M6291 Addrine CCD RCX COMM COS RESTAT FRD M6292 Addrine CCD RCX COMM COS RESTAT FRD M6291 Addrine CCD UNC COMM COS RESTAT FRD M6292 Addrine CCD UNC COMM COS RESTAT FRD M6292 Addrine CCD UNC COMM COS RESTAT FRD M6291 Addrine CCD UNC COMM COS RESTAT FRD M6292 Addrine CCD UNC COMM COS RESTAT FRD M6292 Addrine CCD UNC COMM COS RESTAT FRD M6291 Addrine CCD UNC COMM COS RESTAT FRD M6292 Addrine CCD UNC COMM COS RESTAT FRD M6293 Addrine CCD DEVENDINANA M6294	#define	CCO RCX APP COS BUS ON	M6278
Heffine CCD RCX APE COS INTE FXADLE M6281 Heffine CCD RCX APE COS DMM M6283 Heffine CCD RCX APE COS DMM M6284 Heffine CCD RCX APE COS DMM M6284 Heffine CCD DEVICENCE. M6284 Heffine CCD DEVICENCE. M6284 Heffine CCD DEVICENCE. M6284 Heffine CCD RCX COMP COS DND M6284 Heffine CCD RCX COMP COS DND APP M6291 M6291 Heffine CCD RCX COMP COS DND APP M6291 M6293 Heffine CCD ULCOMMUNICATIONE M6294 M6294 Heffine CCD ULCOMUNICATIONE M630	#define	CCO RCX APP COS BUS ON ENABLE	M6279
Heffine CCD RCX APE COS INTE FXADLE M6281 Heffine CCD RCX APE COS DMM M6283 Heffine CCD RCX APE COS DMM M6284 Heffine CCD RCX APE COS DMM M6284 Heffine CCD DEVICENCE. M6284 Heffine CCD DEVICENCE. M6284 Heffine CCD DEVICENCE. M6284 Heffine CCD RCX COMP COS DND M6284 Heffine CCD RCX COMP COS DND APP M6291 M6291 Heffine CCD RCX COMP COS DND APP M6291 M6293 Heffine CCD ULCOMMUNICATIONE M6294 M6294 Heffine CCD ULCOMUNICATIONE M630	#define	CC0 RCX APP COS INIT	M6280
The Time Cool RGX APP Cool DOX CFC JINA Mc282 Testine CCO RGX APP COOL DOX CFC JINA Mc282 Testine CCO RGX APP COOL DOX CFC JINA Mc283 Testine CCO RGX CONF COOL DOX NUMBER Mc284 Testine CCO RGX CONF COOL CONFIL COXED Mc280 Testine CCO RGX COMP COS RENTART RAD Mc292 Testine CCO RGX COMP COS RENTART RAD Mc292 Testine CCO LUCommunicationError Mc236 Testine CCO LUCommunicationError Mc236 Testine CCO DEDUCTSHMARE Mc238 Testine CCO DEDUCTSHMARE Mc238 Testine CCO LUCommunicationError Mc239 Testine CCO DEDUCTSHMARE Mc339 Testine CCO DEDUCTSHMARE Mc330 Testine CCO DEDUCTSHMARE Mc330 Testine CCO DEDUCTSHMARE	#define	CCO RCX APP COS INIT ENABLE	M6281
Hedrine CCD_RCX APP_COS_DMA M6283 Hedrine CCD_RCX APP_COS_DMA_INALE M6284 Hedrine CCD_RCX APP_COS_DMA_INALE M6285 Hedrine CCD_RCX_APP_COS_DMA_INALE M6286 Hedrine CCD_RCX_COMP_COS_CONTEG_INAL M6287 Hedrine CCD_RCX_COMP_COS_CONTEG_INAL M6281 Hedrine CCD_RCX_COMP_COS_CONTEG_INAL M6291 Hedrine CCD_RCX_COMP_COS_CONTEG_INAL M6293 Hedrine CCD_RCX_COMP_COS_CONTEG M6301 Hedrine CCD_RCX_CONTEG M6301 Hedrine CCD_REVERTION M6303 Hedrine CCD_REVERTION M6303 Hedrine CCD_REVERTION M6304 Hedrine CCD_REVERTION M6303 Hedrine CCD_REVERTION M6304 Hedrine CCD_REVERTION M6304 Hedrine CCD_REVERTION <td< td=""><td>#define</td><td>CCO BCX APP COS LOCK CEG</td><td>M6282</td></td<>	#define	CCO BCX APP COS LOCK CEG	M6282
Medine CCO, RCX, APP, COS, DAA, M6284 Medine CCO, UDENC CONVERSIONAL MC285 M6286 Medine CCO, DAX, CONVERSIONAL MC287 M6286 Medine CCO, RCX, CONVERSIONAL MC287 M6287 Medine CCO, RCX, CONVERSIONAL RESPONSE M6287 Medine CCO, RCX, CONVERSIONAL RESPONSE M6287 Medine CCO, RCX, CONVERSIONAL RESPONSE M6287 Medine CCO, UNICONMERSIONAL MC287 M6288 Medine CCO, UNICONMERSIONAL MC287 M6288 Medine CCO, UNICONMERSIONAL MC288 M6298 Medine CCO, UNICONMERSIONAL MC288 M6298 Medine CCO, DEVISIONAL MC288 M6308 Medine CCO, DEVISIONAL MC388 M6308 Medine CCO, DEVISIONAL MC388 M6308 Medine CCO, DEVISIONAL MC388 M6308	#define	CCO DCY ADD COS LOCK CEC ENA	M6283
Hedine CCD_RCX_APP_COS_DNA_DINALE M6285 Hedine CCD_RCX_COMM_COS_MEANY M6286 Hedine CCD_RCX_COMM_COS_MEANY M6288 Hedine CCD_RCX_COMM_COS_DUS_ON M6289 Hedine CCD_RCX_COMM_COS_DUS_ON M6289 Hedine CCD_RCX_COMM_COS_COMPTIG_LCXED M6299 Hedine CCD_RCX_COMM_COS_COMPTIG_LCXED M6299 Hedine CCD_RCX_COMM_COS_COMPTIG_LCXED M6293 Hedine CCD_RCX_COMM_COS_COMNONNONNON M6293 Hedine CCD_RCX_COMM_COS_COMNONNONNON M6294 Hedine CCD_RCX_COMM_COS_COMNONNONNONNONNON M6294 Hedine CCD_RCX_DUPTOR M6300 Hedine CCD_PEDINISANCe M6301 Hedine CCD_PEDINISANCe M6301 Hedine CCD_REXTRATEACO M6304 Hedine CCD_REXTRATEACO M6304 <td>#dofino</td> <td>CCO BCY ARD COS DMA</td> <td>M6203</td>	#dofino	CCO BCY ARD COS DMA	M6203
<pre>#define CC0 RCX CONM COS EUS ON M6289 #define CC0 RCX CONM COS CONFIG LOCKED M6290 #define CC0 RCX CONM COS CNSTRATERQ M6291 #define CC0 RCX CONM COS RESTART RRQ M6293 #define CC0 ICONMUNICATIONSLUE M6295 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6298 #define CC0 DEPDINSCUECE M6300 #define CC0 DEPDINSCUECE M6301 #define CC0 DEPDINSUECE M6301 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6305 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 UNINACEAEEE M6311 #define CC0 DESTORIGUEAE M6304 #define CC0 UNINACEAEEEE M6310 #define CC0 UNINACEAEEEE M6311 #define CC0 UNINACEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE</pre>	#define	CCU_RCA_AFF_COS_DMA	M0204
<pre>#define CC0 RCX CONM COS EUS ON M6289 #define CC0 RCX CONM COS CONFIG LOCKED M6290 #define CC0 RCX CONM COS CNSTRATERQ M6291 #define CC0 RCX CONM COS RESTART RRQ M6293 #define CC0 ICONMUNICATIONSLUE M6295 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6298 #define CC0 DEPDINSCUECE M6300 #define CC0 DEPDINSCUECE M6301 #define CC0 DEPDINSUECE M6301 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6305 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 UNINACEAEEE M6311 #define CC0 DESTORIGUEAE M6304 #define CC0 UNINACEAEEEE M6310 #define CC0 UNINACEAEEEE M6311 #define CC0 UNINACEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE</pre>	#define	CCU_RCX_APP_COS_DMA_ENABLE	M6285
<pre>#define CC0 RCX CONM COS EUS ON M6289 #define CC0 RCX CONM COS CONFIG LOCKED M6290 #define CC0 RCX CONM COS CNSTRATERQ M6291 #define CC0 RCX CONM COS RESTART RRQ M6293 #define CC0 ICONMUNICATIONSLUE M6295 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6298 #define CC0 DEPDINSCUECE M6300 #define CC0 DEPDINSCUECE M6301 #define CC0 DEPDINSUECE M6301 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6305 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 UNINACEAEEE M6311 #define CC0 DESTORIGUEAE M6304 #define CC0 UNINACEAEEEE M6310 #define CC0 UNINACEAEEEE M6311 #define CC0 UNINACEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE</pre>	#define	CCU_ulDeviceWatchdog	M6286
<pre>#define CC0 RCX CONM COS EUS ON M6289 #define CC0 RCX CONM COS CONFIG LOCKED M6290 #define CC0 RCX CONM COS CNSTRATERQ M6291 #define CC0 RCX CONM COS RESTART RRQ M6293 #define CC0 ICONMUNICATIONSLUE M6295 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6296 #define CC0 UNOMUNICATIONSLUE M6298 #define CC0 DEPDINSCUECE M6300 #define CC0 DEPDINSCUECE M6301 #define CC0 DEPDINSUECE M6301 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6305 #define CC0 DEPDINSUECE M6304 #define CC0 DEPDINSUECE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 DESTORIGUEAE M6304 #define CC0 UNINACEAEEE M6311 #define CC0 DESTORIGUEAE M6304 #define CC0 UNINACEAEEEE M6310 #define CC0 UNINACEAEEEE M6311 #define CC0 UNINACEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE</pre>	#define	CC0_RCX_COMM_COS_READY	M6287
#define CCD_EXC_COMM_COS_CONFIG_LOCKED M6291 #define CCD_EXC_COMM_COS_CONFIG_NEW M6291 #define CCD_EXC_COMM_COS_EXSTATE REQ_ENA M6293 #define CCD_UCOMM_COS_EXSTATE REQ_ENA M6293 #define CCD_UCOMM_COS_EXSTATE REQ_ENA M6293 #define CCD_UCOMMUNICATIONETOR M6294 #define CCD_UCONMUNICATIONETOR M6294 #define CCD_USVersion M6297 #define CCD_USTORISOURCE M6301 #define CCD_DEDINIBSMOde M6301 #define CCD_DEDINISOURCE M6303 #define CCD_DESCOTODINCH M6304 #define CCD_DESCOTODINCH M6307 #define CCD_DESCOTODINCH M6307 #define CCD_DESCOTODINCH M6308 #define CCD_DESCOTODINCH M6307 #define CCD_DESCOTODINCH M6308 #define CCD_DESCOTODINCH M6308 #define CCD_DESCOTODINCH M6308 #define CCD_DESCOTODINCH M6307 #define CCD_DESCOTODINCH M6308 #define CCD_DESCOTODINCH M6308 #define CCD_DESCOTODINCH M6307 #define CCD_DES	#define		M6288
#define CCD_RCX_COMM_COS_RESTART_REQ M6291 #define CCD_RCX_COMM_COS_RESTART_REQ_ENA M6293 #define CCD_UCOMMUNICatIONState M6294 #define CCD_UNOMMUNICATIONSTATE M6294 #define CCD_UNOMMUNICATIONSTATE M6294 #define CCD_UNOMMUNICATIONSTATE M6296 #define CCD_UNOMMUNICATIONSTATE M6296 #define CCD_UNOMMUNICATIONSTATE M6297 #define CCD_PEDINSMUNCE M6301 #define CCD_PEDINSTATE M6301 #define CCD_DEFONUTINENMODE M6303 #define CCD_DEFONUTINE M6304 #define CCD_UNTONCONTIGINSTANS M6314 #define CCD_UNTONCONTIGINSTANS M6314 #define CCD_UNTONCONTIGINSTANS M6314 #define CCD_UNTONCONTIGINANS M6314 #define CCD_UNTONCONTIGINANS M6314 #define CCD_UNTONCONTIGINANS M6314 #de	#define	CC0_RCX_COMM_COS_BUS_ON	M6289
<pre>#define CCD_RCX_COMM_COS_RESTART_REQ_ENA M6293 #define CCD_RCX_COMM_COS_RESTART_REQ_ENA M6294 #define CCD_ulCommunicationError M6296 #define CCD_ulCommunicationError M6296 #define CCD_ulCommunicationError M6296 #define CCD_ulSernationError M6296 #define CCD_USPOILISENdode M6301 #define CCD_ULSernationError M6303 #define CCD_ULSernationError M6303 #define CCD_ULSernationError M6303 #define CCD_ULSernationError M6303 #define CCD_ULSernationError M6304 #define CCD_ULSernationError M6305 #define CCD_ULSernationError M6306 #define CCD_ULSernationError M6307 #define CCD_ULSernationError M6308 #define CCD_ULSernationError M6308 #define CCD_ULSernationError M6308 #define CCD_USernationError M6308 #define CCD_USernationError M6308 #define CCD_USernationError M6308 #define CCD_USernationError M6308 #define CCD_ULSernationError M6310 #define CCD_ULSernationError M6311 #define CCD_ULSernationError M6312 #define CCD_ULSernationError M6314 #define CCD_ULSernationError M6315 #define CCD_ULSERNATIONER M6318 #define CCD_ULSERNATIONER M6318 #define CCD_ULSERNATIONER M6318 #define CCL_ULSERNATIONER M6318 #define CCL_ULSERNATIONER M6318 #define CCL_ULSERNATIONER M6328 #define CCL_ULSERNATIONER M6338 #define CCL_</pre>		CC0 RCX COMM COS CONFIG LOCKED	M6290
<pre>#define CCD_RCX_COMM_COS_RESTART_REQ_ENA M6293 #define CCD_RCX_COMM_COS_RESTART_REQ_ENA M6294 #define CCD_ulCommunicationError M6296 #define CCD_ulCommunicationError M6296 #define CCD_ulCommunicationError M6296 #define CCD_ulSernationError M6296 #define CCD_USPOILISENdode M6301 #define CCD_ULSernationError M6303 #define CCD_ULSernationError M6303 #define CCD_ULSernationError M6303 #define CCD_ULSernationError M6303 #define CCD_ULSernationError M6304 #define CCD_ULSernationError M6305 #define CCD_ULSernationError M6306 #define CCD_ULSernationError M6307 #define CCD_ULSernationError M6308 #define CCD_ULSernationError M6308 #define CCD_ULSernationError M6308 #define CCD_USernationError M6308 #define CCD_USernationError M6308 #define CCD_USernationError M6308 #define CCD_USernationError M6308 #define CCD_ULSernationError M6310 #define CCD_ULSernationError M6311 #define CCD_ULSernationError M6312 #define CCD_ULSernationError M6314 #define CCD_ULSernationError M6315 #define CCD_ULSERNATIONER M6318 #define CCD_ULSERNATIONER M6318 #define CCD_ULSERNATIONER M6318 #define CCL_ULSERNATIONER M6318 #define CCL_ULSERNATIONER M6318 #define CCL_ULSERNATIONER M6328 #define CCL_ULSERNATIONER M6338 #define CCL_</pre>	#define	CC0 RCX COMM COS CONFIG NEW	M6291
Hedefine CC0 RCX COMPL N6293 Hedefine CC0 N1CommunicationState M6294 Hedefine CC0 N1CommunicationState M6294 Hedefine CC0 N1CommunicationState M6294 Hedefine CC0 UswatchdogTime M6293 Hedefine CC0 DPDInHskMode M6294 Hedefine CC0 DPDINtSkMode M6304 Hedefine CC0 DPDINtSkMode M6304 Hedefine CC0 DEFORDSkMode M6310 Hedefine CC0 USANGESkANG M6311 Hedefine CC0 USANGESKANG M6313 Hedefine CC0 USANGESKANG M6314 Hedefine CC1 USANGESKANG M6314<	#define	CCO BCX COMM COS RESTART REO	M6292
#define CCD_RCX_COMM_CCS_IMA M6294 #define CCD_ulCommunicationError M6295 #define CCD_usVersion M6297 #define CCD_usVersion M6299 #define CCD_bFDInBsWode M6299 #define CCD_bFDInBsWode M6299 #define CCD_bFDINSource M6300 #define CCD_bFDOUTSAwtchdog M6301 #define CCD_bFDOUTSAwtchdog M6303 #define CCD_bErorDISIActhdog M6304 #define CCD_bErorDISIActhdog M6306 #define CCD_bErorDISIActhdog M6306 #define CCD_bErorDISIActhdog M6308 #define CCD_bErorDISIActhdog M6308 #define CCD_BSynchstMode M6308 #define CCD_USAysCarce M6310 #define CCD_USAysCarce M6310 #define CCD_USAysCarce M6311 #define CCD_USAysCarce M6313 #define CCD_USAysCarce M6314 #define CCD_USAysCarce M6314 #define CCD_USAysCarces M6314 #define CCL_USAysCarces BUS_ON M6315 #define CCL_USAysCarces BUS_ON M6317	11 -1 - C		M6293
#defineCCO_ulRostWatchdogM6303#defineCCO_ulErorCountM6304#defineCCO_bErrorDigindM6305#defineCCO_bErrorDuctntM6307#defineCCO_bSyncBourceM6309#defineCCO_bSyncBourceM6311#defineCCO_ulSlaveStateM6312#defineCCO_ulNonConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6316#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INITM6324#defineCCI_RCX_COMP_COS_RENDNM6325#defineCCI_RCX_COMP_COS_RENNM6326#defineCCI_RCX_COMP_COS_RENNM6330#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COM	#define	CCO BCX COMM COS DMA	
#defineCCO_ulRostWatchdogM6303#defineCCO_ulErorCountM6304#defineCCO_bErrorDigindM6305#defineCCO_bErrorDuctntM6307#defineCCO_bSyncBourceM6309#defineCCO_bSyncBourceM6311#defineCCO_ulSlaveStateM6312#defineCCO_ulNonConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6316#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INITM6324#defineCCI_RCX_COMP_COS_RENDNM6325#defineCCI_RCX_COMP_COS_RENNM6326#defineCCI_RCX_COMP_COS_RENNM6330#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COM	#define	CCO_ulCommunicationState	
#defineCCO_ulRostWatchdogM6303#defineCCO_ulErorCountM6304#defineCCO_bErrorDigindM6305#defineCCO_bErrorDuctntM6307#defineCCO_bSyncBourceM6309#defineCCO_bSyncBourceM6311#defineCCO_ulSlaveStateM6312#defineCCO_ulNonConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6316#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INITM6324#defineCCI_RCX_COMP_COS_RENDNM6325#defineCCI_RCX_COMP_COS_RENNM6326#defineCCI_RCX_COMP_COS_RENNM6330#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COM	#dof!		
#defineCCO_ulRostWatchdogM6303#defineCCO_ulErorCountM6304#defineCCO_bErrorDigindM6305#defineCCO_bErrorDuctntM6307#defineCCO_bSyncBourceM6309#defineCCO_bSyncBourceM6311#defineCCO_ulSlaveStateM6312#defineCCO_ulNonConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6316#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INITM6324#defineCCI_RCX_COMP_COS_RENDNM6325#defineCCI_RCX_COMP_COS_RENNM6326#defineCCI_RCX_COMP_COS_RENNM6330#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COM	#deline	CCO_utcommunitCationError	
#defineCCO_ulRostWatchdogM6303#defineCCO_ulErorCountM6304#defineCCO_bErrorDigindM6305#defineCCO_bErrorDuctntM6307#defineCCO_bSyncBourceM6309#defineCCO_bSyncBourceM6311#defineCCO_ulSlaveStateM6312#defineCCO_ulNonConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6316#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INITM6324#defineCCI_RCX_COMP_COS_RENDNM6325#defineCCI_RCX_COMP_COS_RENNM6326#defineCCI_RCX_COMP_COS_RENNM6330#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COM	#aeiine	ccu_usversion	
#defineCCO_ulRostWatchdogM6303#defineCCO_ulErorCountM6304#defineCCO_bErrorDigindM6305#defineCCO_bErrorDuctntM6307#defineCCO_bSyncBourceM6309#defineCCO_bSyncBourceM6311#defineCCO_ulSlaveStateM6312#defineCCO_ulNonConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6316#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INITM6324#defineCCI_RCX_COMP_COS_RENDNM6325#defineCCI_RCX_COMP_COS_RENNM6326#defineCCI_RCX_COMP_COS_RENNM6330#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COM	#define	CCU_usWatchdogTime	
#defineCCO_ulRostWatchdogM6303#defineCCO_ulErorCountM6304#defineCCO_bErrorDigindM6305#defineCCO_bErrorDuctntM6307#defineCCO_bSyncBourceM6309#defineCCO_bSyncBourceM6311#defineCCO_ulSlaveStateM6312#defineCCO_ulNonConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6316#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INITM6324#defineCCI_RCX_COMP_COS_RENDNM6325#defineCCI_RCX_COMP_COS_RENNM6326#defineCCI_RCX_COMP_COS_RENNM6330#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COM	#define	CC0_bPDInHskMode	
#defineCCO_ulRostWatchdogM6303#defineCCO_ulErorCountM6304#defineCCO_bErrorDigindM6305#defineCCO_bErrorDuctntM6307#defineCCO_bSyncBourceM6309#defineCCO_bSyncBourceM6311#defineCCO_ulSlaveStateM6312#defineCCO_ulNonConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6316#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INITM6324#defineCCI_RCX_COMP_COS_RENDNM6325#defineCCI_RCX_COMP_COS_RENNM6326#defineCCI_RCX_COMP_COS_RENNM6330#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COM	#define	CC0_bPDInSource	M6300
#defineCCO_ulRostWatchdogM6303#defineCCO_ulErorCountM6304#defineCCO_bErrorDigindM6305#defineCCO_bErrorDuctntM6307#defineCCO_bSyncBourceM6309#defineCCO_bSyncBourceM6311#defineCCO_ulSlaveStateM6312#defineCCO_ulNonConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6314#defineCCO_ulNumOfConfigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6316#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INIT_ENABLEM6323#defineCCI_RCX_APP_COS_INITM6324#defineCCI_RCX_COMP_COS_RENDNM6325#defineCCI_RCX_COMP_COS_RENNM6326#defineCCI_RCX_COMP_COS_RENNM6330#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COMP_COS_RENNM6332#defineCCI_RCX_COMP_COS_RENNM6333#defineCCI_RCX_COM	#define	CC0_bPDOutHskMode	M6301
defineCCO_DETroCLOGINM6005#defineCCO_DETroCPUINCNM6006#defineCCO_DETroCPUINCNM6007#defineCCO_DESTORSUNCHM6009#defineCCO_DSYNCSUNCEM6010#defineCCO_USINUESTATEM6011#defineCCO_USINUESTATEM6014#defineCCO_USINUESTATEM6014#defineCCO_UNUMOFCnfigGlavesM6014#defineCCO_UNUMOFCnfigGlavesM6016#defineCCI_RCX_APP_COS_BUS_ONM6016#defineCCI_RCX_APP_COS_BUS_ONM6017#defineCCI_RCX_APP_COS_INITM6018#defineCCI_RCX_APP_COS_INIT_ENABLEM6020#defineCCI_RCX_APP_COS_INIT_ENABLEM6022#defineCCI_RCX_APP_COS_LOCK_CRGM6022#defineCCI_RCX_APP_COS_DOM_ENABLEM6023#defineCCI_RCX_APP_COS_DOM_ENABLEM6023#defineCCI_RCX_APP_COS_DOM_ENABLEM6024#defineCCI_RCX_COMM_COS_READYM6026#defineCCI_RCX_COMM_COS_READYM6026#defineCCI_RCX_COMM_COS_REATART_REQM60330#defineCCI_RCX_COMM_COS_REATART_REQM60330#defineCCI_RCX_COMM_COS_REATART_REQM60330#defineCCI_RCX_COMM_COS_REATART_REQM60330#defineCCI_RCX_COMM_COS_REATART_REQM60330#defineCCI_RCX_COMM_COS_REATART_REQM60330#defineCCI_RCX_COMM_COS_REATART_REQM6331#defineCCI_RCX_COMM_COS_REATART_REQM6333	#define	CC0 bPDOutSource	M6302
defineCCO_DETroCLOGINM6005#defineCCO_DETroCPUINCNM6006#defineCCO_DETroCPUINCNM6007#defineCCO_DESTORSUNCHM6009#defineCCO_DSYNCSUNCEM6010#defineCCO_USINUMOFCONFIGSLAVESM6011#defineCCO_USINUMOFCONFIGSLAVESM6013#defineCCO_UNINUMOFCONFIGSLAVESM6014#defineCCO_UNINUMOFCONFIGSLAVESM6016#defineCCI_RX_APP_COS_BUS_ONM6016#defineCCI_RX_APP_COS_BUS_ONM6017#defineCCI_RX_APP_COS_INITM6018#defineCCI_RX_APP_COS_INIT_ENABLEM6020#defineCCI_RX_APP_COS_INIT_ENABLEM6022#defineCCI_RX_APP_COS_LOCK_CRGM6221#defineCCI_RX_APP_COS_DOM_ENABLEM6022#defineCCI_RX_APP_COS_DOM_ENABLEM6022#defineCCI_RX_APP_COS_DOM_ENABLEM6023#defineCCI_RX_APP_COS_DOM_ENABLEM6026#defineCCI_RX_COMM_COS_READVM6026#defineCCI_RX_COMM_COS_READNM6028#defineCCI_RX_COMM_COS_REATART_REQM60330#defineCCI_RX_COMM_COS_REATART_REQM60330#defineCCI_RX_COMM_COS_REATART_REQM60330#defineCCI_RX_COMM_COS_REATART_REQM60330#defineCCI_RX_COMM_COS_REATART_REQM60330#defineCCI_RX_COMM_COS_REATART_REQM60330#defineCCI_RX_COMM_COS_REATART_REQM6331#defineCCI_RX_COMM_COS_REATART_RE	#define	CC0 ulHostWatchdog	M6303
#define CC0_DErrorLogInd M6305 #define CC0_DErrorSPLOCH M6306 #define CC0_DErrorSyncCh M6309 #define CC0_DSyncSurce M6310 #define CC0_USlaveState M6311 #define CC0_USlaveState M6313 #define CC0_UNBORCONTORING M6314 #define CC0_UNMOFActiveSlaves M6314 #define CC0_UNMOFActiveSlaves M6316 #define CC1_RCX_APF_COS_RP_READY M6316 #define CC1_RCX_APF_COS_DUS_ON M6317 #define CC1_RCX_APF_COS_DUS_ON M6317 #define CC1_RCX_APF_COS_DUS_ON M6318 #define CC1_RCX_APF_COS_DUCK_CFG M6320 #define CC1_RCX_APF_COS_DUCK_CFG M6321 #define CC1_RCX_APF_COS_DUCK_CFG M6323 #define CC1_RCX_APF_COS_DUX_ENALE M6323 #define CC1_RCX_APF_COS_DUX_ENALE M6324 #define CC1_RCX_CAPF_COS_DUM_ENALE M6327 #define CC1_RCX_COM_COS_CONFIG_LOCKED M6327 #define CC1_RCX_COM_COS_CONFIG_LOCKED M6332 #define CC1_RCX_COM_COS_CONFIG_LOCKED M6332 #define CC1_RCX_COM_COS_CONFIG_LOCKED	#define	CC0_ulErrorCount	
#defineCCODEFrorPDOUTChtM6307#defineCCODEFrorPSyncChtM6308#defineCCODESyncBakModeM6309#defineCCODESyncBakModeM6310#defineCCOULSlaveStateM6311#defineCCOULSlaveStateM6313#defineCCOULNUMOFConFigSlavesM6313#defineCCOULNUMOFConFigSlavesM6316#defineCCOULNUMOFActiveSlavesM6316#defineCCIRCX APF_COSDESON ENABLE#defineCCIRCX APF_COSDESON ENABLE#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX COMM_COS READYM6323#defineCCIRCX COMM_COS READYM6326#defineCCIRCX COMM_COS READYM6326#defineCCIRCX COMM_COS READYM6320#defineCCIRCX COMM_COS REATART_REQM6330#defineCCIRCX COMM_COS REATART_REQM6330#defineCCIRCX COMM_COS REATART_REQM6336#defineCCIRCX COMM_COS REATART_REQM6336#defineCCIRCX COMM_COS REATART_REQ	#define	CC0 bErrorLogInd	
#defineCCODEFrorPDOUTChtM6307#defineCCODEFrorPSyncChtM6308#defineCCODESyncBakModeM6309#defineCCODESyncBakModeM6310#defineCCOULSlaveStateM6311#defineCCOULSlaveStateM6313#defineCCOULNUMOFConFigSlavesM6313#defineCCOULNUMOFConFigSlavesM6316#defineCCOULNUMOFActiveSlavesM6316#defineCCIRCX APF_COSDESON ENABLE#defineCCIRCX APF_COSDESON ENABLE#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX APF_COSJUNT#defineCCIRCX COMM_COS READYM6323#defineCCIRCX COMM_COS READYM6326#defineCCIRCX COMM_COS READYM6326#defineCCIRCX COMM_COS READYM6320#defineCCIRCX COMM_COS REATART_REQM6330#defineCCIRCX COMM_COS REATART_REQM6330#defineCCIRCX COMM_COS REATART_REQM6336#defineCCIRCX COMM_COS REATART_REQM6336#defineCCIRCX COMM_COS REATART_REQ	#define	CCO hErrorPDInCnt	
#defineCCCDEXrorSynChtM6308#defineCCCDSynCHsKModeM6310#defineCCCDSynCBsKModeM6310#defineCCCULSlaveStateM6311#defineCCCULSlaveStateM6313#defineCCCULNumOConfigSlavesM6313#defineCCOULNumOftactiveSlavesM6314#defineCCOULNumOftactiveSlavesM6316#defineCCIRCX APP_COSBUS_ON#defineCCIRCX APP_COSBUS_ON#defineCCIRCX APP_COSBUS_ON#defineCCIRCX APP_COSINTT#defineCCIRCX APP_COSINTT#defineCCIRCX APP_COS_LOCK_CFGM6321#defineCCIRCX APP_COS_LOCK_CFGM6322#defineCCIRCX APP_COS_DMAM6323#defineCCIRCX APP_COS_DMAM6324#defineCCIRCX COM_COS_READYM6326#defineCCIRCX COM_COS_READYM6326#defineCCIRCX COM_COS_READYM6326#defineCCIRCX COM_COS_RESTART_REQM6331#defineCCIRCX COM_COS_RESTART_REQM6332#defineCCIRCX COM_COS_RESTART_REQM6333#defineCCINCX COM_COS_RESTART_REQM6336#defineCCINCX COM_COS_RESTART_REQM6336#defineCCINCX COM_COS_RESTART_REQM6336#defineCCINCX COM_C			
#defineCCODSynchskWodeM6309#defineCCOLSIAveStateM6311#defineCCOLSIAveStateM6311#defineCCOLUNUMOCOnfigSIavesM6313#defineCCOUNUMOFActiveSIavesM6314#defineCCOUNUMOFActiveSIavesM6316#defineCCOUNUMOFActiveSIavesM6316#defineCCIRCX APP_COS_BUS_ONM6317#defineCCIRCX APP_COS_INTM6318#defineCCIRCX APP_COS_INTM6319#defineCCIRCX APP_COS_INTM6320#defineCCIRCX APP_COS_INTM6321#defineCCIRCX APP_COS_INTM6322#defineCCIRCX APP_COS_DMA_ENABLEM6323#defineCCIRCX APP_COS_RINM6323#defineCCIRCX OMM_COS_RADVM6326#defineCCIRCX CMM_COS_RADVM6326#defineCCIRCX CMM_COS_RADYM6330#defineCCIRCX CMM_COS_RADAM6331#defineCCIRCX CMM_COS_RADAM6331#defineCCIRCX CMM_COS_RADAM6333#defineCCIRCX CMM_COS_RADAM6333#defineCCIRCX CMM_COS_RADAM6333#defineCCIRCX CMM_COS_RADAM6333#defineCCIRCX CMM_COS_RADAM6333#defineCCIRCX CMM_COS_RADAM6333#defineCCIRCX CMM_COS_RADAM6334			
#defineCCO_bSyncSourceM6310#defineCCO_ulSlaveStateM6311#defineCCO_ulSlaveStateM6313#defineCCO_ulNumOfDrigSlavesM6314#defineCCO_ulNumOfDrigSlavesM6315#defineCCO_ulNumOfDrigSlavesM6316#defineCCI_RCX_APP_COS_BUS_ONM6317#defineCCI_RCX_APP_COS_INTM6319#defineCCI_RCX_APP_COS_INTM6320#defineCCI_RCX_APP_COS_INTM6321#defineCCI_RCX_APP_COS_INTM6323#defineCCI_RCX_APP_COS_DMAM6323#defineCCI_RCX_APP_COS_DMAM6323#defineCCI_RCX_APP_COS_DMAM6324#defineCCI_RCX_APP_COS_DMAM6326#defineCCI_RCX_COM_COS_READYM6326#defineCCI_RCX_COM_COS_READYM6320#defineCCI_RCX_COM_COS_CONFIG_LOCKEDM6329#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6320#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6320#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6320#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6332#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6332#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6332#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6331#defineCCI_RCX_COMM_COS_CONFIG_LOCKEDM6332#defineCCI_RCX_COMM_CO	#deline		
#defineCCO_ulSlaveStateM6311#defineCCO_ulNumofConfigSlavesM6312#defineCCO_ulNumofActiveSlavesM6313#defineCCO_ulNumofActiveSlavesM6314#defineCCI_RCX_APP_COS_BUS_ONM6316#defineCCI_RCX_APP_COS_BUS_ON_ENABLEM6317#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_INITM6312#defineCCI_RCX_APP_COS_INITM6319#defineCCI_RCX_APP_COS_LOCK_CFGM6321#defineCCI_RCX_APP_COS_LOCK_CFGM6321#defineCCI_RCX_APP_COS_DAAM6323#defineCCI_RCX_APP_COS_DAAM6323#defineCCI_RCX_COM_COS_RAAPYM6326#defineCCI_RCX_COM_COS_RAAPYM6326#defineCCI_RCX_COM_COS_RUS_ONM6327#defineCCI_RCX_COM_COS_RUS_ONM6327#defineCCI_RCX_COM_COS_RUS_NM6326#defineCCI_RCX_COM_COS_RUS_NM6326#defineCCI_RCX_COM_COS_RUS_NM6327#defineCCI_RCX_COM_COS_RUS_NM6332#defineCCI_RCX_COM_COS_RUS_NM6332#defineCCI_RCX_COM_COS_RUS_NM6332#defineCCI_RCX_COM_COS_RUS_NM6332#defineCCI_RCX_COM_COS_RUS_NM6332#defineCCI_RCX_COM_COS_RUS_NM6333#defineCCI_RCX_COM_COS_RUS_NM6333#defineCCI_RCX_COM_COS_RUS_NM6334#defineCCI_RCX_COM_COS_RUS_NM6337#defineCC			
#defineCC1_RCX_APP_COS_BUS_ONM6317#defineCC1_RCX_APP_COS_BUS_ON_ENABLEM6318#defineCC1_RCX_APP_COS_INITM6319#defineCC1_RCX_APP_COS_INIT_ENABLEM6320#defineCC1_RCX_APP_COS_LOCK_CFGM6321#defineCC1_RCX_APP_COS_LOCK_CFGM6322#defineCC1_RCX_APP_COS_DMAM6323#defineCC1_RCX_APP_COS_DMA_ENABLEM6323#defineCC1_RCX_APP_COS_MA_ENABLEM6324#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_READYM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_UCommunicationStateM6333#defineCC1_uUCommunicationErrorM6336#defineCC1_UUCommunicationErrorM6336#defineCC1_bPDINBskModeM6337#defineCC1_bPDINSurceM6343#defineCC1_bPDINSurceM6341#defineCC1_uHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6344#defineCC1_bErrorDPINCntM6346	#define	CC0_bSyncSource	
#defineCC1_RCX_APP_COS_BUS_ONM6317#defineCC1_RCX_APP_COS_BUS_ON_ENABLEM6318#defineCC1_RCX_APP_COS_INITM6319#defineCC1_RCX_APP_COS_INIT_ENABLEM6320#defineCC1_RCX_APP_COS_LOCK_CFGM6321#defineCC1_RCX_APP_COS_LOCK_CFGM6322#defineCC1_RCX_APP_COS_DMAM6323#defineCC1_RCX_APP_COS_DMA_ENABLEM6323#defineCC1_RCX_APP_COS_MA_ENABLEM6324#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_READYM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_UCommunicationStateM6333#defineCC1_uUCommunicationErrorM6336#defineCC1_UUCommunicationErrorM6336#defineCC1_bPDINBskModeM6337#defineCC1_bPDINSurceM6343#defineCC1_bPDINSurceM6341#defineCC1_uHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6344#defineCC1_bErrorDPINCntM6346	#define	CC0_ulSlaveState	
#defineCC1_RCX_APP_COS_BUS_ONM6317#defineCC1_RCX_APP_COS_BUS_ON_ENABLEM6318#defineCC1_RCX_APP_COS_INITM6319#defineCC1_RCX_APP_COS_INIT_ENABLEM6320#defineCC1_RCX_APP_COS_LOCK_CFGM6321#defineCC1_RCX_APP_COS_LOCK_CFGM6322#defineCC1_RCX_APP_COS_DMAM6323#defineCC1_RCX_APP_COS_DMA_ENABLEM6323#defineCC1_RCX_APP_COS_MA_ENABLEM6324#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_READYM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_UCommunicationStateM6333#defineCC1_uUCommunicationErrorM6336#defineCC1_UUCommunicationErrorM6336#defineCC1_bPDINBskModeM6337#defineCC1_bPDINSurceM6343#defineCC1_bPDINSurceM6341#defineCC1_uHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6344#defineCC1_bErrorDPINCntM6346	#define	CC0_ulSlaveErrLogInd	M6312
#defineCC1_RCX_APP_COS_BUS_ONM6317#defineCC1_RCX_APP_COS_BUS_ON_ENABLEM6318#defineCC1_RCX_APP_COS_INITM6319#defineCC1_RCX_APP_COS_INIT_ENABLEM6320#defineCC1_RCX_APP_COS_LOCK_CFGM6321#defineCC1_RCX_APP_COS_LOCK_CFGM6322#defineCC1_RCX_APP_COS_DMAM6323#defineCC1_RCX_APP_COS_DMA_ENABLEM6323#defineCC1_RCX_APP_COS_MA_ENABLEM6324#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_READYM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_UCommunicationStateM6333#defineCC1_uUCommunicationErrorM6336#defineCC1_UUCommunicationErrorM6336#defineCC1_bPDINBskModeM6337#defineCC1_bPDINSurceM6343#defineCC1_bPDINSurceM6341#defineCC1_uHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6344#defineCC1_bErrorDPINCntM6346	#define	CC0 ulNumOfConfigSlaves	M6313
#defineCC1_RCX_APP_COS_BUS_ONM6317#defineCC1_RCX_APP_COS_BUS_ON_ENABLEM6318#defineCC1_RCX_APP_COS_INITM6319#defineCC1_RCX_APP_COS_INIT_ENABLEM6320#defineCC1_RCX_APP_COS_LOCK_CFGM6321#defineCC1_RCX_APP_COS_LOCK_CFGM6322#defineCC1_RCX_APP_COS_DMAM6323#defineCC1_RCX_APP_COS_DMA_ENABLEM6323#defineCC1_RCX_APP_COS_MA_ENABLEM6324#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_READYM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_UCommunicationStateM6333#defineCC1_uUCommunicationErrorM6336#defineCC1_UUCommunicationErrorM6336#defineCC1_bPDINBskModeM6337#defineCC1_bPDINSurceM6343#defineCC1_bPDINSurceM6341#defineCC1_uHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6344#defineCC1_bErrorDPINCntM6346	#define	CC0 ulNumOfActiveSlaves	M6314
#defineCC1_RCX_APP_COS_BUS_ONM6317#defineCC1_RCX_APP_COS_BUS_ON_ENABLEM6318#defineCC1_RCX_APP_COS_INITM6319#defineCC1_RCX_APP_COS_INIT_ENABLEM6320#defineCC1_RCX_APP_COS_LOCK_CFGM6321#defineCC1_RCX_APP_COS_LOCK_CFGM6322#defineCC1_RCX_APP_COS_DMAM6323#defineCC1_RCX_APP_COS_DMA_ENABLEM6323#defineCC1_RCX_APP_COS_MA_ENABLEM6324#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_READYM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_UCommunicationStateM6333#defineCC1_uUCommunicationErrorM6336#defineCC1_UUCommunicationErrorM6336#defineCC1_bPDINBskModeM6337#defineCC1_bPDINSurceM6343#defineCC1_bPDINSurceM6341#defineCC1_uHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6342#defineCC1_UHostWatchdogM6344#defineCC1_bErrorDPINCntM6346	#define	CC0 ulNumOfDiagSlaves	
#defineCC1_RCX_APP_COS_BUS_ON_ENABLEM6318#defineCC1_RCX_APP_COS_INIT_ENABLEM6319#defineCC1_RCX_APP_COS_INIT_ENABLEM6320#defineCC1_RCX_APP_COS_LOCK_CFGM6321#defineCC1_RCX_APP_COS_DACK_CFGM6322#defineCC1_RCX_APP_COS_DMAM6323#defineCC1_RCX_CAPP_COS_DMA_ENABLEM6323#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_BUS_ONM6327#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_RCX_COMM_COS_DNAM6333#defineCC1_UCommunicationStateM6336#defineCC1_ulCommunicationStateM6336#defineCC1_ulCommunicationStateM6337#defineCC1_bPDOutHskModeM6338#defineCC1_bPDOutHskModeM6341#defineCC1_ulCommunicationGM6341#defineCC1_ullForoCountM6343#defineCC1_ullForoCountM6346	#define	CC1 RCX APP COS APP READY	
#defineCC1_RCX_APP_COS_BUS_ON_ENABLEM6318#defineCC1_RCX_APP_COS_INIT_ENABLEM6319#defineCC1_RCX_APP_COS_INIT_ENABLEM6320#defineCC1_RCX_APP_COS_LOCK_CFGM6321#defineCC1_RCX_APP_COS_DACK_CFGM6322#defineCC1_RCX_APP_COS_DMAM6323#defineCC1_RCX_CAPP_COS_DMA_ENABLEM6323#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_BUS_ONM6327#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_RCX_COMM_COS_DNAM6333#defineCC1_UCommunicationStateM6336#defineCC1_ulCommunicationStateM6336#defineCC1_ulCommunicationStateM6337#defineCC1_bPDOutHskModeM6338#defineCC1_bPDOutHskModeM6341#defineCC1_ulCommunicationGM6341#defineCC1_ullForoCountM6343#defineCC1_ullForoCountM6346	#define	CC1 RCX APP COS BUS ON	M6317
#defineCCI_RCX_APP_COS_INIT_ENABLEM6319#defineCCI_RCX_APP_COS_INIT_ENABLEM6320#defineCCI_RCX_APP_COS_LOCK_CFGM6321#defineCCI_RCX_APP_COS_DMAM6323#defineCCI_RCX_APP_COS_DMA_ENABLEM6324#defineCCI_RCX_COM_COS_READYM6325#defineCCI_RCX_COM_COS_READYM6326#defineCCI_RCX_COM_COS_RUNM6327#defineCCI_RCX_COMM_COS_RUNM6328#defineCCI_RCX_COMM_COS_RUNM6328#defineCCI_RCX_COMM_COS_RUNM6330#defineCCI_RCX_COMM_COS_RUNM6331#defineCCI_RCX_COMM_COS_RUNM6332#defineCCI_RCX_COMM_COS_RUNM6332#defineCCI_RCX_COMM_COS_RUNM6332#defineCCI_RCX_COMM_COS_RUNM6332#defineCCI_RCX_COMM_COS_RUNAM6333#defineCCI_RCX_COMM_COS_RUNAM6332#defineCCI_UCOmmunicationErrorM6335#defineCCI_UUCommunicationErrorM6336#defineCCI_UUCommunicationErrorM6337#defineCCI_DUNAtchdogTimeM6337#defineCCI_DDDUNTskModeM6340#defineCCI_DDDUNTskModeM6341#defineCCI_UUErrorCountM6342#defineCCI_UUErrorCountM6343#defineCCI_UUErrorDUNCNM6346	#define	CC1 DCA YDD COG BIIG ON ENIYDIE	M6318
#defineCC1_RCX_APP_COS_INIT_ENABLEM6320#defineCC1_RCX_APP_COS_LOCK_CFGM6321#defineCC1_RCX_APP_COS_DMAM6323#defineCC1_RCX_APP_COS_DMA_ENABLEM6324#defineCC1_RCX_COMM_COS_DMA_ENABLEM6326#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_RUNM6327#defineCC1_RCX_COMM_COS_RUNM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6329#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_REATART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6332#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_ulCommunicationStateM6336#defineCC1_ulCommunicationErrorM6336#defineCC1_ulCommunicationErrorM6336#defineCC1_bPDINSourceM6340#defineCC1_bPDINSourceM6341#defineCC1_bPDINSourceM6342#defineCC1_bPDOutBskModeM6342#defineCC1_ulErrorCountM6343#defineCC1_ulErrorDInCntM6344#defineCC1_bErrorPDInCntM6346			MC210
#defineCC1_RCX_APP_COS_LOCK_CFG_ENAM6322#defineCC1_RCX_APP_COS_DMAM6324#defineCC1_RCX_COM_COS_DMA_ENABLEM6325#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6329#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_RCX_COMM_COS_DMAM6333#defineCC1_ulCommunicationStateM6335#defineCC1_usVersionM6336#defineCC1_bPDInHskModeM6337#defineCC1_bPDInHskModeM6334#defineCC1_bPDInHskModeM6341#defineCC1_bPDOuthSourceM6341#defineCC1_ulHostWatchdogM6342#defineCC1_ulHostWatchdogM6343#defineCC1_bPDOuthSourceM6341#defineCC1_bErrorPDInCntM6345			
#defineCC1_RCX_APP_COS_LOCK_CFG_ENAM6322#defineCC1_RCX_APP_COS_DMAM6324#defineCC1_RCX_COM_COS_DMA_ENABLEM6325#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6329#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQM6333#defineCC1_RCX_COMM_COS_DMAM6333#defineCC1_ulCommunicationStateM6335#defineCC1_usVersionM6336#defineCC1_bPDInHskModeM6337#defineCC1_bPDInHskModeM6334#defineCC1_bPDInHskModeM6341#defineCC1_bPDOuthSourceM6341#defineCC1_ulHostWatchdogM6342#defineCC1_ulHostWatchdogM6343#defineCC1_bPDOuthSourceM6341#defineCC1_bErrorPDInCntM6345	#define	CCI_KCX_APP_COS_INIT_ENABLE	11032U
#defineCC1_RCX_APP_COS_DMA_ENABLEM6323#defineCC1_RCX_COMM_COS_DMA_ENABLEM6325#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_BUS_ONM6327#defineCC1_RCX_COMM_COS_BUS_ONM6329#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_NEWM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6333#defineCC1_ulcommunicationStateM6335#defineCC1_ulcommunicationErrorM6335#defineCC1_ulcommunicationErrorM6336#defineCC1_bPDInHskModeM6337#defineCC1_bPDInthskModeM6341#defineCC1_bPDOutHskModeM6341#defineCC1_ulErrorCountM6343#defineCC1_ulErrorCountM6343#defineCC1_bErrorDDInCntM6346	#define	CC1_RCX_APP_COS_LOCK_CFG	M6321
#defineCC1_RCX_APP_COS_DMA_ENABLEM6323#defineCC1_RCX_COMM_COS_DMA_ENABLEM6325#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_READYM6327#defineCC1_RCX_COMM_COS_BUS_ONM6327#defineCC1_RCX_COMM_COS_BUS_ONM6329#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6330#defineCC1_RCX_COMM_COS_CONFIG_NEWM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6333#defineCC1_ulcommunicationStateM6335#defineCC1_ulcommunicationErrorM6335#defineCC1_ulcommunicationErrorM6336#defineCC1_bPDInHskModeM6337#defineCC1_bPDInthskModeM6341#defineCC1_bPDOutHskModeM6341#defineCC1_ulErrorCountM6343#defineCC1_ulErrorCountM6343#defineCC1_bErrorDDInCntM6346	#define	CC1_RCX_APP_COS_LOCK_CFG_ENA	M6322
#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_RUNM6327#defineCC1_RCX_COMM_COS_BUS_ONM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6329#defineCC1_RCX_COMM_COS_CONFIG_NEWM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6332#defineCC1_ulCommunicationStateM6333#defineCC1_ulCommunicationStateM6335#defineCC1_usVersionM6336#defineCC1_bPDIntBskModeM6337#defineCC1_bPDIntSurceM6339#defineCC1_bPDOutHskModeM6340#defineCC1_ullComrunicationGM6341#defineCC1_ullErorCountM6343#defineCC1_bPDOutHskModeM6343#defineCC1_ullErorDouttM6344#defineCC1_bPDOutSourceM6343#defineCC1_bErrorDouttM6346	#define	CC1 RCX APP COS DMA	M6323
#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_RUNM6327#defineCC1_RCX_COMM_COS_BUS_ONM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6329#defineCC1_RCX_COMM_COS_CONFIG_NEWM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6332#defineCC1_ulCommunicationStateM6333#defineCC1_ulCommunicationStateM6335#defineCC1_usVersionM6336#defineCC1_bPDIntBskModeM6337#defineCC1_bPDIntSurceM6339#defineCC1_bPDOutHskModeM6340#defineCC1_ullComrunicationGM6341#defineCC1_ullErorCountM6343#defineCC1_bPDOutHskModeM6343#defineCC1_ullErorDouttM6344#defineCC1_bPDOutSourceM6343#defineCC1_bErrorDouttM6346	#define	CC1_RCX_APP_COS_DMA ENABLE	M6324
#defineCC1_RCX_COMM_COS_READYM6326#defineCC1_RCX_COMM_COS_RUNM6327#defineCC1_RCX_COMM_COS_BUS_ONM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6329#defineCC1_RCX_COMM_COS_RESTART_REQM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6332#defineCC1_ulCommunicationStateM6334#defineCC1_ulCommunicationErrorM6335#defineCC1_usVersionM6336#defineCC1_bPDIntBskModeM6338#defineCC1_bPDIntSurceM6336#defineCC1_bPDoutHskModeM6340#defineCC1_ulRorroroutM6343#defineCC1_ulBrorrocoutM6343#defineCC1_bPDOutSourceM6343#defineCC1_ulBrorrocoutM6343#defineCC1_bPDOutSourceM6343#defineCC1_bPTorrologIndM6344#defineCC1_bErrorPDInCntM6346	#define	CC1 ulDeviceWatchdog	M6325
#defineCC1_RCX_COMM_COS_RUNM6327#defineCC1_RCX_COMM_COS_BUS_ONM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6329#defineCC1_RCX_COMM_COS_CONFIG_NEWM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6332#defineCC1_ulcommunicationStateM6334#defineCC1_ulcommunicationErrorM6335#defineCC1_usVersionM6336#defineCC1_bPDInHskModeM6338#defineCC1_bPDInHskModeM6340#defineCC1_bPDOutHskModeM6341#defineCC1_ullerorCountM6342#defineCC1_bPDOutSourceM6341#defineCC1_bPDOutSourceM6342#defineCC1_bPDOutSourceM6343#defineCC1_bPTorDInCntM6345#defineCC1_bErrorPDInCntM6345	#define	CC1 RCX COMM COS READY	
#defineCC1_RCX_COMM_COS_BUS_ONM6328#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6329#defineCC1_RCX_COMM_COS_CONFIG_NEWM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6332#defineCC1_RCX_COMM_COS_DMAM6333#defineCC1_ulCommunicationStateM6334#defineCC1_ulCommunicationErrorM6336#defineCC1_usWatchdogTimeM6337#defineCC1_bPDInHskModeM6338#defineCC1_bPDIntskmodeM6340#defineCC1_bPDoutsourceM6341#defineCC1_ulErrorCountM6343#defineCC1_bPDOutsourceM6343#defineCC1_bPDOutsourceM6343#defineCC1_bPDOutsourceM6343#defineCC1_bPTorLoTM6345#defineCC1_bErrorDDICNtM6346			
#defineCC1_RCX_COMM_COS_CONFIG_LOCKEDM6329#defineCC1_RCX_COMM_COS_CONFIG_NEWM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6332#defineCC1_ulCommunicationStateM6333#defineCC1_ulCommunicationErrorM6335#defineCC1_usVersionM6336#defineCC1_usWatchdogTimeM6337#defineCC1_bPDInHskModeM6339#defineCC1_bPDInSourceM6340#defineCC1_ulHostWatchdogM6341#defineCC1_ulErrorCountM6343#defineCC1_ulErrorLogIndM6343#defineCC1_bErrorPDInCntM6345			
#defineCC1_RCX_COMM_COS_CONFIG_NEWM6330#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6332#defineCC1_ulCommunicationStateM6333#defineCC1_ulCommunicationErrorM6335#defineCC1_usVersionM6336#defineCC1_usWatchdogTimeM6337#defineCC1_bPDInHskModeM6338#defineCC1_bPDInSourceM6339#defineCC1_bPDOutHskModeM6340#defineCC1_ulHostWatchdogTM6341#defineCC1_ulErrorCountM6343#defineCC1_bErrorLogIndM6344#defineCC1_bErrorPDInCntM6346			
#defineCC1_RCX_COMM_COS_RESTART_REQM6331#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6332#defineCC1_RCX_COMM_COS_DMAM6333#defineCC1_ulCommunicationStateM6334#defineCC1_ulCommunicationErrorM6335#defineCC1_usWatchdogTimeM6337#defineCC1_bPDInHskModeM6338#defineCC1_bPDInHskModeM6339#defineCC1_bPDoutHskModeM6340#defineCC1_ulHostWatchdogTimeM6341#defineCC1_ulHostWatchdogM6342#defineCC1_ulErrorCountM6343#defineCC1_bErrorLogIndM6345#defineCC1_bErrorPDOutCntM6346			
#defineCC1_RCX_COMM_COS_RESTART_REQ_ENAM6332#defineCC1_RCX_COMM_COS_DMAM6333#defineCC1_ulCommunicationStateM6334#defineCC1_ulCommunicationErrorM6335#defineCC1_usVersionM6336#defineCC1_usWatchdogTimeM6337#defineCC1_bPDInHskModeM6338#defineCC1_bPDInHskModeM6339#defineCC1_bPDOutHskModeM6340#defineCC1_bPDOutHskModeM6341#defineCC1_ulHostWatchdogM6342#defineCC1_ulErrorCountM6343#defineCC1_bErrorLogIndM6344#defineCC1_bErrorPDInCntM6345#defineCC1_bErrorPDOutCntM6346			
#defineCC1_RCX_COMM_COS_DMAM6333#defineCC1_ulCommunicationStateM6334#defineCC1_ulCommunicationErrorM6335#defineCC1_usVersionM6336#defineCC1_usWatchdogTimeM6337#defineCC1_bPDInHskModeM6338#defineCC1_bPDInBsOurceM6339#defineCC1_bPDOutHskModeM6340#defineCC1_bPDOutSourceM6341#defineCC1_ulHostWatchdogM6342#defineCC1_ulErrorCountM6343#defineCC1_bErrorLogIndM6344#defineCC1_bErrorPDInCntM6345#defineCC1_bErrorPDOutCntM6346			
#define CC1_ulCommunicationStateM6334#define CC1_ulCommunicationErrorM6335#define CC1_usVersionM6336#define CC1_usWatchdogTimeM6337#define CC1_bPDInHskModeM6338#define CC1_bPDInSourceM6339#define CC1_bPDOutHskModeM6340#define CC1_bPDOutSourceM6341#define CC1_ulHostWatchdogM6342#define CC1_ulErrorCountM6343#define CC1_bErrorLogIndM6344#define CC1_bErrorPDInCntM6345#define CC1_bErrorPDOutCntM6346			
#define CC1_ulCommunicationErrorM6335#define CC1_usVersionM6336#define CC1_usWatchdogTimeM6337#define CC1_bPDInHskModeM6338#define CC1_bPDInSourceM6339#define CC1_bPDOutHskModeM6340#define CC1_bPDOutSourceM6341#define CC1_ulHostWatchdogM6342#define CC1_ulErrorCountM6343#define CC1_bErrorLogIndM6344#define CC1_bErrorPDInCntM6345#define CC1_bErrorPDOutCntM6346			
#define CC1_usVersionM6336#define CC1_usWatchdogTimeM6337#define CC1_bPDInHskModeM6338#define CC1_bPDInSourceM6339#define CC1_bPDOutHskModeM6340#define CC1_bPDOutSourceM6341#define CC1_ulHostWatchdogM6342#define CC1_ulErrorCountM6343#define CC1_bErrorLogIndM6345#define CC1_bErrorPDInCntM6346			
#defineCC1_usWatchdogTimeM6337#defineCC1_bPDInHskModeM6338#defineCC1_bPDInSourceM6339#defineCC1_bPDOutHskModeM6340#defineCC1_bPDOutSourceM6341#defineCC1_ulHostWatchdogM6342#defineCC1_ulErrorCountM6343#defineCC1_bErrorLogIndM6345#defineCC1_bErrorPDOutCntM6346			
#define CC1_bPDInHskModeM6338#define CC1_bPDInSourceM6339#define CC1_bPDOutHskModeM6340#define CC1_bPDOutSourceM6341#define CC1_uHostWatchdogM6342#define CC1_ulErrorCountM6343#define CC1_bErrorLogIndM6344#define CC1_bErrorPDInCntM6345#define CC1_bErrorPDOutCntM6346		_	M6336
#define CC1_bPDInSourceM6339#define CC1_bPDOutHskModeM6340#define CC1_bPDOutSourceM6341#define CC1_ulHostWatchdogM6342#define CC1_ulErrorCountM6343#define CC1_bErrorLogIndM6344#define CC1_bErrorPDInCntM6345#define CC1_bErrorPDOutCntM6346	#define	CC1_usWatchdogTime	M6337
#define CC1_bPDInSourceM6339#define CC1_bPDOutHskModeM6340#define CC1_bPDOutSourceM6341#define CC1_ulHostWatchdogM6342#define CC1_ulErrorCountM6343#define CC1_bErrorLogIndM6344#define CC1_bErrorPDInCntM6345#define CC1_bErrorPDOutCntM6346	#define	CC1 bPDInHskMode	M6338
#define CC1_bPDOutHskModeM6340#define CC1_bPDOutSourceM6341#define CC1_ulHostWatchdogM6342#define CC1_ulErrorCountM6343#define CC1_bErrorLogIndM6344#define CC1_bErrorPDInCntM6345#define CC1_bErrorPDOutCntM6346			
#define CC1_bPDOutSourceM6341#define CC1_ulHostWatchdogM6342#define CC1_ulErrorCountM6343#define CC1_bErrorLogIndM6344#define CC1_bErrorPDInCntM6345#define CC1_bErrorPDOutCntM6346			
#define CC1_ulHostWatchdogM6342#define CC1_ulErrorCountM6343#define CC1_bErrorLogIndM6344#define CC1_bErrorPDInCntM6345#define CC1_bErrorPDOutCntM6346			
#define CC1_ulerrorCountM6343#define CC1_berrorLogIndM6344#define CC1_berrorPDInCntM6345#define CC1_berrorPDOutCntM6346		-	
#define CC1_bErrorLogIndM6344#define CC1_bErrorPDInCntM6345#define CC1_bErrorPDOutCntM6346			
<pre>#define CC1_bErrorPDInCnt M6345 #define CC1_bErrorPDOutCnt M6346</pre>			
#define CC1_bErrorPDOutCnt M6346			
#define CC1_bErrorSyncCnt M6347			
	#define	CC1_bErrorSyncCnt	M6347

#define CC1 bSyncHskMode	M6348	
#define CC1 bSyncSource	M6349	
#define CC1 ulSlaveState	M6350	
#define CC1 ulSlaveErrLogInd	M6351	
#define CC1_ulNumOfConfigSlaves	M6352	
#define CC1_ulNumOfActiveSlaves	M6353	
#define CC1_ulNumOfDiagSlaves	M6354	

MacroNameDefinition_\$6C000.pmc File Content
CLOSE
END GAT DEL GAT
#Include "MacroNameDefinition_\$6C000.h"
SI_abCookie_0>Y:\$6C000,0,8
SI_abCookie_1>Y:\$6C000,8,8
SI_abCookie_2>X:\$6C000,0,8
SI_abCookie_3>X:\$6C000,8,8
SI_ulDpmTotalSize->DP:\$6C001 SI_ulDeviceNumber->DP:\$6C002
SI ulSerialNumber->DP:\$6C003
SI_ausHwOptions_0>Y:\$6C004,0,16
SI ausHwOptions 1 ->X:\$6C004,0,16
SI_ausHwOptions_2>Y:\$6C005,0,16
SI_ausHwOptions_3>X:\$6C005,0,16
SI_usManufacturer->Y:\$6C006,0,16
SI_usProductionDate->X:\$6C006,0,16
SI_ulLicenseFlags1->DP:\$6C007 SI_ulLicenseFlags2->DP:\$6C008
SI usNetxLicenseID->Y:\$6C009,0,16
SI usNetxLicenseFlags->X:\$6C009,0,16
SI_usDeviceClass->Y:\$6C00A,0,16
SI_bHwRevision->X:\$6C00A,0,8
SI_bHwCompatibility->X:\$6C00A,8,8
SI_bDevIdNumber->Y:\$6C00B,0,8
<pre>SCI_bChannelType->Y:\$6C00C,0,8 SCI_bSizePositionOfHandshake->X:\$6C00C,0,8</pre>
SCI bNumberOfBlocks->X:\$6C00C,8,8
SCI ulSizeOfChannel->DP:\$6C00D
SCI_usSizeOfMailbox->Y:\$6C00E,0,16
SCI_usMailboxStartOffset->X:\$6C00E,0,16
HCI_bChannelType->Y:\$6C010,0,8
HCI_ulSizeOfChannel->DP:\$6C011
CCOI_bChannelType->Y:\$6C014,0,8 CCOI_bChannelId->Y:\$6C014,8,8
CCOI bSizePositionOfHandshake->X:\$6C014,0,8
CCOI bNumberOfBlocks->X:\$6C014,8,8
CC01 ulSizeOfChannel->DP:\$6C015
CCOI_usCommunicationClass->Y:\$6C016,0,16
CCOI_usProtocolClass->X:\$6C016,0,16
CCOI_usConformanceClass->Y:\$6C017,0,16
CC1I_bChannelType->Y:\$6C018,0,8 CC1I_bChannelId->Y:\$6C018,8,8
CC11 bSizePositionOfHandshake->X:\$6C018,0,8
CC1I bNumberOfBlocks->X:\$6C018,8,8
CC1I_ulSizeOfChannel->DP:\$6C019
CC1I_usCommunicationClass->Y:\$6C01A,0,16
CC1I_usProtocolClass->X:\$6C01A,0,16
CC11_usConformanceClass->Y:\$6C01B,0,16
CC2I_bChannelType->Y:\$6C01C,0,8 CC2I_bChannelId->Y:\$6C01C,8,8
CC2I bSizePositionOfHandshake->X:\$6C01C,0,8
CC2I bNumberOfBlocks->X:\$6C01C,8,8
CC2I_ulSizeOfChannel->DP:\$6C01D
CC2I_usCommunicationClass->Y:\$6C01E,0,16
CC2I_usProtocolClass->X:\$6C01E,0,16
CC2I_usConformanceClass->Y:\$6C01F,0,16
CC3I_bChannelType->Y:\$6C020,0,8 CC3I_bChannelId->Y:\$6C020,8,8
Denamiettu=/1.900020,0,0

CC3I bSizePositionOfHandshake->X:\$6C020,0,8 CC3I bNumberOfBlocks->X:\$6C020,8,8 CC3I ulSizeOfChannel->DP:\$6C021 CC3I usCommunicationClass->Y:\$6C022,0,16 CC3I usProtocolClass->X:\$6C022,0,16 CC3I usConformanceClass->Y:\$6C023,0,16 AC01 bChannelType->Y:\$6C024,0,8 ACOI bChannelId->Y:\$6C024,8,8 ACOI bSizePositionOfHandshake->X:\$6C024,0,8 AC01 bNumberOfBlocks->X:\$6C024,8,8 AC01 ulSizeOfChannel->DP:\$6C025 AC11 bChannelType->Y:\$6C028,0,8 AC11_bChannelId->Y:\$6C028,8,8 AC11 bSizePositionOfHandshake->X:\$6C028,0,8 AC11 bNumberOfBlocks->X:\$6C028,8,8 AC11 ulSizeOfChannel->DP:\$6C029 SCtrl ulSystemCommandCOS->DP:\$6C02E SStat_ulSystemCOS->DP:\$6C030 SStat ulSystemStatus->DP:\$6C031 SStat_ulSystemError->DP:\$6C032 SStat ulBootError->DP:\$6C033 SStat ulTimeSinceStart->DP:\$6C034 SStat usCpuLoad->Y:\$6C035,0,16 SStat ulHWFeatures->DP:\$6C036 SSMB usPackagesAccepted->Y:\$6C040,0,16 SSMB_ulDest->DP:\$6C041 SSMB_ulSrc->DP:\$6C042 SSMB ulDestId->DP:\$6C043 SSMB_ulSrcId->DP:\$6C044 SSMB_ullen->DP:\$6C045 SSMB_ulId->DP:\$6C046 SSMB_ulState->DP:\$6C047 SSMB ulCmd->DP:\$6C048 SSMB_ulExt->DP:\$6C049 SSMB ulRout->DP:\$6C04A SSMB ultData0->DP:\$6C04B SSMB ultData1->DP:\$6C04C SSMB_ultData2->DP:\$6C04D SSMB ultData3->DP:\$6C04E SSMB ultData4->DP:\$6C04F SSMB_ultData5->DP:\$6C050 SSMB ultData6->DP:\$6C051 SSMB_ultData7->DP:\$6C052 SSMB_ultData8->DP:\$6C053 SSMB ultData9->DP:\$6C054 SSMB ultData10->DP:\$6C055 SSMB ultData11->DP:\$6C056 SSMB_ultData12->DP:\$6C057 SSMB ultData13->DP:\$6C058 SSMB ultData14->DP:\$6C059 SSMB ultData15->DP:\$6C05A SSMB_ultData16->DP:\$6C05B SSMB_ultData17->DP:\$6C05C SSMB ultData18->DP:\$6C05D SSMB_ultData19->DP:\$6C05E SSMB ultData20->DP:\$6C05F SRMB usWaitingPackages->Y:\$6C060,0,16 SRMB ulDest->DP:\$6C061 SRMB ulSrc->DP:\$6C062 SRMB ulDestId->DP:\$6C063 SRMB_ulSrcId->DP:\$6C064 SRMB_ullen->DP:\$6C065 SRMB ulid->DP:\$6C066 SRMB ulState->DP:\$6C067 SRMB ulCmd->DP:\$6C068 SRMB_ulExt->DP:\$6C069 SRMB_ulRout->DP:\$6C06A SRMB ultData0->DP:\$6C06B SRMB ultData1->DP:\$6C06C SRMB ultData2->DP:\$6C06D SRMB_ultData3->DP:\$6C06E

SRMB ultData4->DP:\$6C06F
SRMB_ultData5->DP:\$6C070
SRMB_ultData6->DP:\$6C071
SRMB ultData7->DP:\$6C072
SRMB_ultData8->DP:\$6C073
-
SRMB_ultData9->DP:\$6C074
SRMB_ultData10->DP:\$6C075
SRMB_ultData11->DP:\$6C076
SRMB_ultData12->DP:\$6C077
SRMB_ultData13->DP:\$6C078
_
SRMB_ultData14->DP:\$6C079
SRMB ultData15->DP:\$6C07A
SRMB_ultData16->DP:\$6C07B
-
SRMB_ultData17->DP:\$6C07C
SRMB ultData18->DP:\$6C07D
SRMB_ultData19->DP:\$6C07E
SRMB_ultData20->DP:\$6C07F
HCSC_bNetxFlags->X:\$6C080,0,8
HCSC_NSF_READY->X:\$6C080,0,1
HCSC_NSF_READI=>X:\$0C000,0,1
HCSC NSF ERROR->X:\$6C080,1,1
<pre>HCSC_NSF_ERROR->X:\$6C080,1,1 HCSC_NSF_HOST_COS_ACK->X:\$6C080,2,1</pre>
HUSU_NSF_NETX_CUS_CMD->X:\$6CU80,3,1
HCSC_NSF_NETX_COS_CMD->X:\$6C080,3,1 HCSC_NSF_SEND_MBX_ACK->X:\$6C080,4,1
HCSC NSF RECV MBX CMD->X S6C080 5 1
HCSC_NSF_RECV_MBX_CMD->X:\$6C080,5,1 HCSC_bHostFlags->X:\$6C080,8,8
HUSU_DHOSTFIAGS->X:\$6C080,8,8
HCSC HSF RESET->X:\$6C080,8,1
HCSC_HSF_RESET->X:\$6C080,8,1 HCSC_HSF_BOOTSTART->X:\$6C080,9,1
HCSC_HSF_BOOTSTART >X.900000, 5,1
HCSC_HSF_HOST_COS_CMD->X:\$6C080,10,1
HCSC HSF NETX COS ACK->X:\$6C080,11,1
HCSC HSF SEND MBX CMD->X:\$6C080,12,1
HCSC_HSF_RECV_MBX_ACK->X:\$6C080,13,1
HCCC0 usNetxFlags->Y:\$6C082,0,16
HCCC0 NCF COMMUNICATING->Y:\$6C082,0,1
HCCC0 NCF ERROR->Y:\$6C082,1,1
HCCC0 NCF HOST COS ACK->Y:\$6C082,2,1
HCCC0_NCF_NETX_COS_CMD->Y:\$6C082,3,1
HCCCO NCF SEND MBX ACK->Y:\$6C082,4,1
HCCC0 NCF RECV MBX CMD->Y:\$6C082,5,1
HCCC0_NCF_PD0_OUT_ACK->Y:\$6C082,6,1
HCCCO NCF PDO IN CMD->Y:\$6C082,7,1
HCCC0 NCF PD1 OUT ACK->Y:\$6C082,8,1
HCCC0_NCF_PD1_IN_CMD->Y:\$6C082,9,1
HCCC0 usHostFlags->X:\$6C082,0,16
HCCC0 HCF HOST COS CMD->X:\$6C082,2,1
HCCC0_HCF_NETX_COS_ACK->X:\$6C082,3,1
HCCC0 HCF SEND MBX CMD->X:\$6C082,4,1
HCCC0 HCF RECV MBX ACK->X:\$6C082,5,1
HCCC0_HCF_PD0_OUT_CMD->X:\$6C082,6,1
HCCCO HCF PDO IN ACK->X:\$6C082,7,1
HCCC0_HCF_PD1_OUT_CMD->X:\$6C082,8,1
HCCC0_HCF_PD1_IN_ACK->X:\$6C082,9,1
HCCC1 usNetxFlags->Y:\$6C083,0,16
HCCC1 NCF COMMUNICATING->Y:\$6C083,0,1
HCCC1_NCF_ERROR->Y:\$6C083,1,1
HCCC1 NCF HOST COS ACK->Y:\$6C083,2,1
HCCC1 NCF NETX COS CMD->Y:\$6C083,3,1
HCCC1_NCF_SEND_MBX_ACK->Y:\$6C083,4,1
HCCC1 NCF RECV MBX CMD->Y:\$6C083,5,1
HCCC1_NCF_PD0_IN_CMD->Y:\$6C083,7,1
HCCC1_NCF_PD0_IN_CMD->Y:\$6C083,7,1 HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8.1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1 HCCC1_HCF_NETX_COS_ACK->X:\$6C083,3,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1 HCCC1_HCF_NETX_COS_ACK->X:\$6C083,3,1 HCCC1_HCF_SEND_MBX_CMD->X:\$6C083,4,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1 HCCC1_HCF_NETX_COS_ACK->X:\$6C083,3,1 HCCC1_HCF_SEND_MBX_CMD->X:\$6C083,4,1 HCCC1_HCF_RECV_MBX_ACK->X:\$6C083,5,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1 HCCC1_HCF_NETX_COS_ACK->X:\$6C083,3,1 HCCC1_HCF_SEND_MBX_CMD->X:\$6C083,4,1 HCCC1_HCF_RECV_MBX_ACK->X:\$6C083,5,1 HCCC1_HCF_PD0_OUT_CMD->X:\$6C083,6,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1 HCCC1_HCF_NETX_COS_ACK->X:\$6C083,3,1 HCCC1_HCF_SEND_MBX_CMD->X:\$6C083,4,1 HCCC1_HCF_RECV_MBX_ACK->X:\$6C083,5,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1 HCCC1_HCF_NETX_COS_ACK->X:\$6C083,3,1 HCCC1_HCF_SEND_MBX_CMD->X:\$6C083,4,1 HCCC1_HCF_RECV_MBX_ACK->X:\$6C083,5,1 HCCC1_HCF_PD0_OUT_CMD->X:\$6C083,6,1 HCCC1_HCF_PD0_IN_ACK->X:\$6C083,7,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1 HCCC1_HCF_NETX_COS_ACK->X:\$6C083,3,1 HCCC1_HCF_SEND_MBX_CMD->X:\$6C083,4,1 HCCC1_HCF_RECV_MBX_ACK->X:\$6C083,5,1 HCCC1_HCF_PD0_OUT_CMD->X:\$6C083,7,1 HCCC1_HCF_PD0_IN_ACK->X:\$6C083,7,1 HCCC1_HCF_PD1_OUT_CMD->X:\$6C083,8,1
HCCC1_NCF_PD1_OUT_ACK->Y:\$6C083,8,1 HCCC1_NCF_PD1_IN_CMD->Y:\$6C083,9,1 HCCC1_usHostFlags->X:\$6C083,0,16 HCCC1_HCF_HOST_COS_CMD->X:\$6C083,2,1 HCCC1_HCF_NETX_COS_ACK->X:\$6C083,3,1 HCCC1_HCF_SEND_MBX_CMD->X:\$6C083,4,1 HCCC1_HCF_RECV_MBX_ACK->X:\$6C083,5,1 HCCC1_HCF_PD0_OUT_CMD->X:\$6C083,6,1 HCCC1_HCF_PD0_IN_ACK->X:\$6C083,7,1

HCCC2 usNetxFlags->Y:\$6C084,0,16 HCCC2 NCF COMMUNICATING->Y:\$6C084,0,1 HCCC2 NCF ERROR->Y:\$6C084,1,1 HCCC2 NCF HOST COS_ACK->Y:\$6C084,2,1 HCCC2 NCF NETX COS CMD->Y:\$6C084,3,1 HCCC2_NCF_SEND_MBX_ACK->Y:\$6C084,4,1 HCCC2_NCF_RECV_MBX_CMD->Y:\$6C084,5,1 HCCC2 NCF PD0 OUT ACK->Y:\$6C084,6,1 HCCC2_NCF_PD0_IN_CMD->Y:\$6C084,7,1 HCCC2_NCF_PD1_OUT_ACK->Y:\$6C084,8,1 HCCC2_NCF_PD1_IN_CMD->Y:\$6C084,9,1 HCCC2_usHostFlags->X:\$6C084,0,16 HCCC2_HCF_HOST_COS_CMD->X:\$6C084,2,1 HCCC2_HCF_NETX_COS_ACK->X:\$6C084,3,1 HCCC2 HCF SEND MBX CMD->X:\$6C084,4,1 HCCC2_HCF_RECV_MBX_ACK->X:\$6C084,5,1 HCCC2_HCF_PD0_OUT_CMD->X:\$6C084,6,1 HCCC2_HCF_PD0_IN_ACK->X:\$6C084,7,1 HCCC2 HCF PD1 OUT CMD->X:\$6C084,8,1 HCCC2_HCF_PD1_IN_ACK->X:\$6C084,9,1 HCCC3 usNetxFlags->Y:\$6C085,0,16 HCCC3 NCF COMMUNICATING->Y:\$6C085,0,1 HCCC3 NCF ERROR->Y:\$6C085,1,1 HCCC3_NCF_HOST_COS_ACK->Y:\$6C085,2,1 HCCC3_NCF_NETX_COS_CMD->Y:\$6C085,3,1 HCCC3 NCF SEND MBX ACK->Y:\$6C085,4,1 HCCC3_NCF_RECV_MBX_CMD->Y:\$6C085,5,1 HCCC3 NCF PD0 OUT ACK->Y:\$6C085,6,1 HCCC3 NCF PD0 IN CMD->Y:\$6C085,7,1 HCCC3_NCF_PD1_OUT_ACK->Y:\$6C085,8,1 HCCC3_NCF_PD1_IN_CMD->Y:\$6C085,9,1 HCCC3 usHostFlags->X:\$6C085,0,16 HCCC3 HCF HOST COS CMD->X:\$6C085,2,1 HCCC3 HCF NETX COS ACK->X:\$6C085,3,1 HCCC3 HCF SEND MBX CMD->X:\$6C085,4,1 HCCC3 HCF RECV MBX ACK->X:\$6C085,5,1 HCCC3 HCF PD0 OUT CMD->X:\$6C085,6,1 HCCC3_HCF_PD0_IN_ACK->X:\$6C085,7,1 HCCC3_HCF_PD1_OUT_CMD->X:\$6C085,8,1 HCCC3 HCF PD1 IN ACK->X:\$6C085,9,1 HCAC0 usNetxFlags->Y:\$6C086,0,16 HCAC0 NCF COMMUNICATING->Y:\$6C086,0,1 HCAC0 NCF ERROR->Y:\$6C086,1,1 HCAC0_NCF_HOST_COS_ACK->Y:\$6C086,2,1 HCAC0_NCF_NETX_COS_CMD->Y:\$6C086,3,1 HCAC0_NCF_SEND_MBX_ACK->Y:\$6C086,4,1 HCACO NCF RECV MBX CMD->Y:\$6C086,5,1 HCACO_NCF_PDO_OUT_ACK->Y:\$6C086,6,1 HCACO_NCF_PDO_IN_CMD->Y:\$6C086,7,1 HCACO_NCF_PD1_OUT_ACK->Y:\$6C086,8,1 HCAC0 NCF PD1 IN CMD->Y:\$6C086,9,1 HCAC0_usHostFlags->X:\$6C086,0,16 HCACO HCF HOST COS CMD->X:\$6C086,2,1 HCACO HCF NETX COS ACK->X:\$6C086,3,1 HCAC0_HCF_SEND_MBX_CMD->X:\$6C086,4,1 HCAC0 HCF RECV MBX ACK->X:\$6C086,5,1 HCACO HCF PDO OUT CMD->X:\$6C086,6,1 HCACO HCF PDO IN ACK->X:\$6C086,7,1 HCAC0_HCF_PD1_OUT_CMD->X:\$6C086,8,1 HCAC0_HCF_PD1_IN_ACK->X:\$6C086,9,1 HCAC1 usNetxFlags->Y:\$6C087,0,16 HCAC1_NCF_COMMUNICATING->Y:\$6C087,0,1 HCAC1 NCF ERROR->Y:\$6C087,1,1 HCAC1 NCF HOST COS ACK->Y:\$6C087,2,1 HCAC1 NCF NETX COS CMD->Y:\$6C087,3,1 HCAC1_NCF_SEND_MBX_ACK->Y:\$6C087,4,1 HCAC1_NCF_RECV_MBX_CMD->Y:\$6C087,5,1 HCAC1 NCF PD0 OUT ACK->Y:\$6C087,6,1 HCAC1_NCF_PD0_IN_CMD->Y:\$6C087,7,1 HCAC1_NCF_PD1_OUT_ACK->Y:\$6C087,8,1 HCAC1_NCF_PD1_IN_CMD->Y:\$6C087,9,1

HCAC1 usHostFlags->X:\$6C087,0,16 HCAC1_HCF_HOST_COS_CMD->X:\$6C087,2,1 HCAC1 HCF NETX COS ACK->X:\$6C087,3,1 HCAC1 HCF SEND MBX CMD->X:\$6C087,4,1 HCAC1 HCF RECV MBX ACK->X:\$6C087,5,1 HCAC1_HCF_PD0_OUT_CMD->X:\$6C087,6,1 HCAC1_HCF_PD0_IN_ACK->X:\$6C087,7,1 HCAC1 HCF PD1 OUT CMD->X:\$6C087,8,1 HCAC1 HCF PD1 IN ACK->X:\$6C087,9,1 CCO RCX APP COS APP READY->Y:\$6C0C2,0,1 CCO RCX APP COS BUS ON->Y:\$6C0C2,1,1 CC0 RCX APP COS BUS ON ENABLE->Y:\$6C0C2,2,1 CC0 RCX APP COS INIT->Y:\$6C0C2,3,1 CC0 RCX APP COS INIT ENABLE->Y:\$6C0C2,4,1 CC0 RCX APP COS LOCK CFG->Y:\$6C0C2,5,1 CC0_RCX_APP_COS_LOCK_CFG_ENA->Y:\$6C0C2,6,1 CC0 RCX APP COS DMA->Y:\$6C0C2,7,1 CCO RCX APP COS DMA ENABLE->Y:\$6C0C2,8,1 CC0 ulDeviceWatchdog->DP:\$6C0C3 CC0 RCX COMM COS READY->Y:\$6C0C4,0,1 CC0 RCX COMM COS RUN->Y:\$6C0C4,1,1 CC0 RCX COMM COS BUS ON->Y:\$6C0C4,2,1 CC0_RCX_COMM_COS_CONFIG_LOCKED->Y:\$6C0C4,3,1 CC0_RCX_COMM_COS_CONFIG_NEW->Y:\$6C0C4,4,1 CC0_RCX_COMM_COS_RESTART_REQ->Y:\$6C0C4,5,1 CC0 RCX COMM COS RESTART REQ ENA->Y:\$6C0C4,6,1 CC0_RCX_COMM_COS_DMA->Y:\$6C0C4,7,1 CC0 ulCommunicationState->DP:\$6C0C5 CC0 ulCommunicationError->DP:\$6C0C6 CCO usVersion->Y:\$6C0C7,0,16 CC0_usWatchdogTime->X:\$6C0C7,0,16 CC0 bPDInHskMode->Y:\$6C0C8,0,8 CC0 bPDInSource->Y:\$6C0C8,8,8 CC0 bPDOutHskMode->X:\$6C0C8,0,8 CC0 bPDOutSource->X:\$6C0C8,8,8 CC0 ulHostWatchdog->DP:\$6C0C9 CC0 ulErrorCount->DP:\$6C0CA CC0 bErrorLogInd->Y:\$6C0CB,0,8 CC0 bErrorPDInCnt->Y:\$6C0CB,8,8 CC0 bErrorPDOutCnt->X:\$6C0CB,0,8 CC0 bErrorSyncCnt->X:\$6C0CB,8,8 CC0 bSyncHskMode->Y:\$6C0CC,0,8 CC0 bSyncSource->Y:\$6C0CC,8,8 CC0 ulSlaveState->DP:\$6C0CE CC0 ulSlaveErrLogInd->DP:\$6C0CF CC0_ulNumOfConfigSlaves->DP:\$6C0D0 CC0_ulNumOfActiveSlaves->DP:\$6C0D1 CC0 ulNumOfDiagSlaves->DP:\$6C0D2 CC1 RCX APP COS APP READY->Y:\$6D002,0,1 CC1 RCX APP COS BUS ON->Y:\$6D002,1,1 CC1_RCX_APP_COS_BUS_ON_ENABLE->Y:\$6D002,2,1 CC1_RCX_APP_COS_INIT=>Y:\$6D002,3,1 CC1_RCX_APP_COS_INIT_ENABLE=>Y:\$6D002,4,1 CC1 RCX APP COS LOCK CFG->Y:\$6D002,5,1 CC1_RCX_APP_COS_LOCK_CFG_ENA->Y:\$6D002,6,1 CC1_RCX_APP_COS_DMA->Y:\$6D002,7,1 CC1 RCX APP COS DMA ENABLE->Y:\$6D002,8,1 CC1_ulDeviceWatchdog->DP:\$6D003 CC1 RCX COMM COS READY->Y:\$6D004,0,1 CC1 RCX COMM COS RUN->Y:\$6D004,1,1 CC1 RCX COMM COS BUS ON->Y:\$6D004,2,1 CC1_RCX_COMM_COS_CONFIG_LOCKED->Y:\$6D004,3,1 CC1 RCX COMM COS CONFIG NEW->Y:\$6D004,4,1 CC1 RCX COMM COS RESTART REQ->Y:\$6D004,5,1 CC1 RCX COMM COS RESTART REQ ENA->Y:\$6D004,6,1 CC1_RCX_COMM_COS_DMA->Y:\$6D004,7,1 CC1_ulCommunicationState->DP:\$6D005 CC1 ulCommunicationError->DP:\$6D006 CC1_usVersion->Y:\$6D007,0,16 CC1 usWatchdogTime->X:\$6D007,0,16 CC1 bPDInHskMode->Y:\$6D008,0,8

CC1 bPDInSource->Y:\$6D008,8,8	
CC1 bPDOutHskMode->X:\$6D008,0,8	
CC1 bPDOutSource->X:\$6D008,8,8	
CC1_ulHostWatchdog->DP:\$6D009	
CC1 ulerrorCount->DP:\$6D00A	
CC1 bErrorLogInd->Y:\$6D00B,0,8	
CC1 bErrorPDInCnt->Y:\$6D00B,8,8	
CC1 bErrorPDOutCnt->X:\$6D00B,0,8	
CC1 bErrorSyncCnt->X:\$6D00B,8,8	
CC1 bSyncHskMode->Y:\$6D00C,0,8	
CC1 bSyncSource->Y:\$6D00C,8,8	
CC1 ulSlaveState->DP:\$6D00E	
CC1_ulSlaveErrLogInd->DP:\$6D00F	
CC1_ulNumOfConfigSlaves->DP:\$6D010	
CC1_ulNumOfActiveSlaves->DP:\$6D011	
CC1 ulNumOfDiagSlaves->DP:\$6D012	

DPRAM DATA PROCESSING

Since there are two processors (i.e. UMAC and netX) attempting to access data registers in Dual-Ported Memory simultaneously, several handshaking modes can be used to guarantee data consistency. Each sub-block defines the type of handshaking, if any, it requires. The ACC-72EX Setup Assistant software output file lists the type of handshaking required for each of the sub-blocks available on the COMX module.

Should handshaking not be used, collision circuitry on the gateway will, in the very least, guarantee consistency within single byte boundaries.

Non-Cyclic Data Exchange

The mailbox of a communication channel or system channel has two areas that are used for non-cyclic message transfer to and from the netX.

Send Mailbox (System / Communication Channel)

Packet transfer from UMAC to netX firmware

Receive Mailbox (System / Communication Channel)

Packet transfer from netX firmware to UMAC

For a communication channel, send and receive mailbox areas are used by fieldbus protocols, providing a non-cyclic data exchange mechanism. Another use of the mailbox system is to allow access to the firmware running on the netX chip for diagnostic and identification purposes. The **send mailbox** is used to transfer cyclic data **to** the network or **to** the netX. The **receive mailbox** is used to transfer cyclic data **from** the netX. Modbus Plus or Ethernet TCP/IP is an example of a fieldbus protocol which utilizes a non-cyclic data exchange.

Whether or not a mailbox is used depends on the function of the firmware.



Each mailbox can hold one packet at a time. netX stores packets in an internal packet queue; these packets are not retrieved by UMAC. This queue has limited space and may fill up, so new packets may be lost. To avoid these deadlock situations, it is strongly recommended to empty the mailbox frequently, even if packets are not expected by the UMAC program. Unexpected command packets should be returned to the sender with an Unknown Command in the status field; unexpected reply messages can be discarded.

Message or Packets

The non-cyclic packets obtained through the netX mailbox have the following structure:

	Hilscher Documentation	ACC-72EX Setup Assistant
box	usPackagesAccepted	SSMB_usPackagesAccepted
	ulDest	SSMB_ulDest
	ulSrc	SSMB_ulSrc
Лаі	ulDestId	SSMB_ulDestId
Send Mailbox	ulSrcId	SSMB_ulSrcId
	ulLen	SSMB_ulLen
Block	ulld	SSMB_ulld
Blo	ulState	SSMB_ulState
System	ulCmd	SSMB_ulCmd
	ulExt	SSMB_ulExt
	ulRout	SSMB_ulRout
		SSMB_ultData0 SSMB_ultData20

	Hilscher Documentation	ACC-72EX Setup Assistant
хо	usWaitingPackages	SRMB_usWaitingPackages
	ulDest	SRMB_ulDest
Mailbox	ulSrc	SRMB_ulSrc
	ulDestId	SRMB_ulDestId
Receive	ulSrcId	SRMB_ulSrcId
ece	ulLen	SRMB_ulLen
	ulld	SRMB_ulid
loc	ulState	SRMB_ulState
System Block	ulCmd	SRMB_ulCmd
ster	ulExt	SRMB_ulExt
Sys	ulRout	SRMB_ulRout
		SRMB_ultData0 SRMB_ultData20

The size of a packet is always at least 40 bytes. Depending on the command, a packet may or may not have a payload in the data field (tData). If present, the contents of the data field are specific to the command or reply.

Destination Queue Handler

The ulDest field identifies a task queue in the context of the netX firmware. The task queue represents the final receiver of the packet and is assigned to a protocol stack. The ulDest field has to be filled out in any case; otherwise, the netX operating system cannot route the packet.

Source Queue Handler

The ulSrc field identifies the sender of the packet. In the context of the netX firmware (inter-task communication), this field holds the identifier of the sending task. Usually, a UMAC program uses this field for its own handle, but it can hold any handle of the sending process. The receiving task does not evaluate this field and will pass it back unchanged to the originator of the packet.

Destination Identifier

The ulDestId field identifies the destination of an unsolicited packet from the netX firmware to the UMAC. It can hold any handle that helps identify the receiver. Its use is mandatory for unsolicited packets. The receiver of unsolicited packets has to register for this service (details are yet to be determined).

Source Identifier

The ulSrcId field identifies the originator of a packet. This field is used by a UMAC program which passes a packet from an external process to an internal netX task. The ulSrcId field holds the handle of the external process. When the netX operating system returns the packet, the UMAC program can identify the packet, and returns it to the originating process. The receiving task on the netX does not evaluate this field, and passes it back unchanged. For inter-task communication, this field is not used.

• Length of Data Field

The ulLen field holds the size of the data field tData in bytes. It defines the total size of the packet's payload that follows the packet's header. Note that the size of the header is not included in ulLen. Depending on the command or reply, a data field may or may not be present in a packet. If no data field is used, the length field is set to zero.

Identifier

The ulld field is used to identify a specific packet among others of the same kind. That way the application or driver can match a specific reply or confirmation packet to a previous request packet. The receiving task does not change this field and passes it back to the originator of the packet. Its use is optional in most of cases, but it is mandatory for fragmented packets. Example: downloading large amounts of data that do not fit into a single packet. For fragmented packets, the identifier field is incremented by one for every new packet.

• Status / Error Code

The ulSta field is used in response or confirmation packets. It informs the originator of the packet about success or failure of the execution of the command. The field may be also used to hold status information in a request packet. Status and error codes that may be returned in ulSta are outlined in Status and Error Code section.

Command / Response

The ulCmd field holds the command code or the response code. The command/response is specific to the receiving task. If a task is not able to execute certain commands, it will return the packet with an error indication. A command is always even (the least significant bit is zero). In the response packet, the command code is incremented by one indicating a confirmation to the request packet.

Extension

The extension field ulExt is used for controlling packets that are sent in a sequenced or fragmented manner. The extension field indicates the first, last, or a packet of a sequence. If fragmentation of packets is not required, the extension field is set to zero.

Routing Information

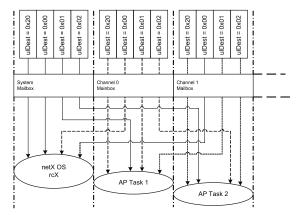
The ulRout field is used internally by the netX firmware only. It has no meaning to a driver type application and therefore is set to zero.

User Data Field

The tData field contains the payload of the packet. Depending on the command or reply, a packet may or may not have a data field. The length of the data field is given in the ulLen field.

About System and Channel Mailbox

The preferred way to address the netX operating system, rcX, is through the system mailbox. The preferred way to address a protocol stack is through its channel mailbox. All mailboxes, however, have a mechanism to route packets to any communication channel or the system channel. Therefore, the destination identifier ulDest in a packet header has to be filled in according to the targeted receiver (see the following image).



The above figure and table below illustrate the use of the destination identifier ulDest.

Value	Definition / Description
\$0	Packet is passed to the netX operating system rcX
\$1	Packet is passed to communication channel 0
\$2	Packet is passed to communication channel 1
\$3	Packet is passed to communication channel 2
\$4	Packet is passed to communication channel 3
\$20	Packet is passed to 'local' communication or system channel
Else	Reserved, Do Not Use

In regards to the channel identifier 0x00000020 (= Channel Token), the Channel Token is valid for any mailbox. That way, the UMAC program uses the same identifier for all packets without actually knowing which mailbox or communication channel is applied. The packet stays "local." The system mailbox is a little bit different because it is used to communicate to the netX operating system, rcX. The rcX has its own range of valid command codes and differs from the communication channels.

If there is a reply packet, the netX operating system returns it to the same mailbox that the request packet went through. Consequently, the UMAC program has to return its reply packet to the mailbox from which the request was received.

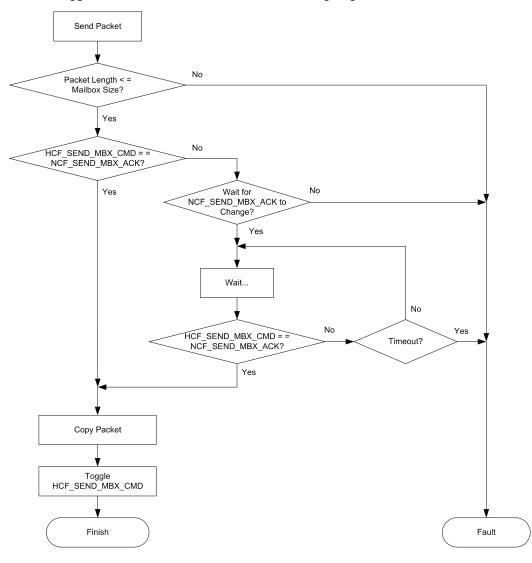
Command and Acknowledge

To ensure data consistency over the content of a mailbox, the firmware uses a pair of flags, each for one direction. Engaging these flags gives access rights alternating to either the user application or the netX firmware. If both UMAC and netX firmware were to access the mailbox at the same time, it may cause loss of data or inconsistency.

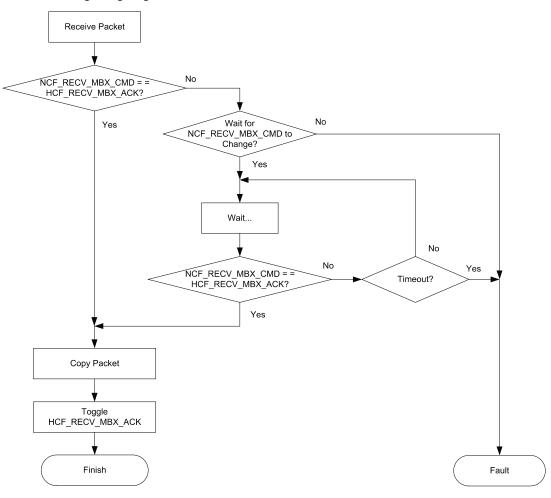
As a general rule, if both flags have the same value (both are set or both are cleared), the process which intends to write has access rights. If they have a different value, the process which intends to read has access rights. The following table illustrates this mechanism.

Send Mailbox	CMD Flag	ACK Flag	
UMAC Has Write Access	0	0	netX Has NO Read Access
UMAC Has NO Write Access	0	1	netX Has Read Access
UMAC Has NO Write Access	1	0	netX Has Read Access
UMAC Has Write Access	1	1	netX Has NO Read Access
Receive Mailbox	CMD Flag	ACK Flag	
UMAC Has NO Read Access	0	0	netX Has Write Access
UMAC Has Read Access	0	1	netX Has NO Write Access
UMAC Has Read Access	1	0	netX Has NO Write Access
UMAC Has NO Read Access	1	1	netX Has Write Access

The following flowcharts illustrate how the transfer mechanism (send and receive packets) works. In order to send a packet, first the function checks if the size of the packet to be sent exceeds the mailbox size. If both the Host Send Mailbox Command flag and the netX Send Mailbox Acknowledge flag are either set or cleared, the host application is allowed to send the packet. When copying data to the mailbox is done, the host toggles the Host Send Mailbox Command flag to give control to the netX firmware.

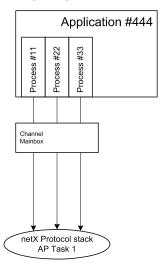


In order to receive a packet, the function checks if the netX Receive Mailbox Command flag and the Host Receive Mailbox Acknowledge flag have different values. If so, the host application is allowed to access the mailbox. When the host is done copying data from the mailbox, the host toggles the Host Receive Mailbox Acknowledge flag to give control to the netX firmware.



Using ulSrc and ulSrcId

Generally, a netX protocol stack is addressed through its communication channel mailbox. The example below shows how a host application addresses a protocol stack running in the context of the netX chip. The application is identified by a number (#444 in this example). The application consists of three processes numbered #11, #22 and #33. These processes communicate through the channel mailbox to the AP task of a protocol stack. See the following image:



Example:

This example applies to command messages imitated by a process in the context of the host application identified by number #444. If the process #22 sends a packet through the channel mailbox to the AP task, the packet header has to be filled in as follows:

Destination Queue Handler	ulDest =	32; /*	0x20: local channel mailbox */
Source Queue Handler	ulSrc =	444; /*	host application */
Destination Identifier	ulDestId=	0; /*	not used */
Source Identifier	ulSrcId =	22; /*	process number */

For packets through the channel mailbox, the application uses 32 (= 0x20, Channel Token) for the destination queue handler ulDest. The source queue handler ulSrc and the source identifier ulSrcId are used to identify the originator of a packet. The destination identifier ulDestId can be used to address certain resources in the protocol stack. It is not used in this example. The source queue handler ulSrc must have an entry, and therefore its use is mandatory; the use of ulSrcId is optional.

The netX operating system passes the request packet to the protocol stack's AP task. The protocol stack then builds a reply to the packet and returns it to the mailbox. The application has to make sure that the packet finds its way back to the originator (process #22 in the example).

How to Route rcX Packets

To route an rcX packet, the source identifier ulSrcId and the source queues handler ulSrc in the packet header hold the identification of the originating process. The router saves the original handle from ulSrcId and ulSrc. It uses handles of its own choice for ulSrcId and ulSrc before it sends the packet to the receiving process. That way, the router can identify the corresponding reply packet and match the handle from that packet with the one stored earlier. Lastly, the router replaces its handles with the original handles and returns the packet to the originating process.

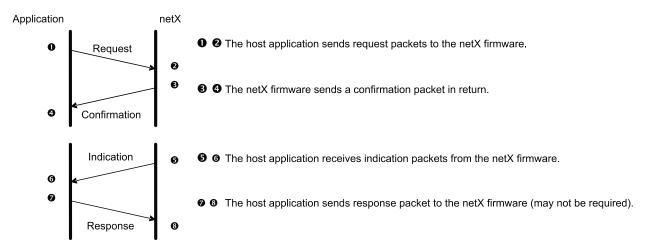
Client/Server Mechanism

Depending on the message destination or packet protocol, the UMAC program or application can act as a client or a server. This section explains both methods, but selection of the appropriate method depends on the destination or protocol option.

Application as Client

The host application may send request packets to the netX firmware at any time (transition $1 \Rightarrow 2$). Depending on the protocol stack running on the netX, parallel packets are not permitted (see protocol specific manual for details). The netX firmware sends a confirmation packet in return, signaling success or failure (transition $3 \Rightarrow 4$) while processing the request.

The host application has to register with the netX firmware in order to receive indication packets (transition $5 \Rightarrow 6$). Depending on the protocol stack, this is done either implicitly (if application opens a TCP/UDP socket) or explicitly (if application wants to receive unsolicited DPV1 packets). Details on when and how to register for certain events is described in the protocol specific manual. Depending on the command code of the indication packet, a response packet to the netX firmware may or may not be required (transition $7 \Rightarrow 8$).



Application as Server

The host application has to register with the netX firmware in order to receive indication packets (unsolicited telegrams). Depending on the protocol stack, this is done either implicitly (if the application opens a TCP/UDP socket) or explicitly (if the application wants to receive unsolicited DPV1 packets). Details on when and how to register for certain events is described in the protocol-specific manual.

When an appropriate event occurs and the host application is registered to receive such a notification, the netX firmware passes an indication packet through the mailbox (transition $1 \Rightarrow 2$). The host application is expected to send a response packet back to the netX firmware (transition $3 \Rightarrow 4$).



Input/Output Data Image

Hilscher products support two methods for accessing the Input/Output Data Image:

- DPM (Dual-Ported Memory) Mode
- DMA (Direct Memory Access) Mode

However, the modules used in ACC-72EX only support the DPM mode, and only Hilscher PCI cards support DMA mode.

In DPM Mode, handshaking between the UMAC (host) program and netX is required for any data transfer.

Process Data Handshake Modes

The netX firmware allows controlling the transfer of data independently for inputs and outputs. Therefore, the process data handshake is carried out individually for input and output image. The handshake cells are located in the handshake channel.

Mode	Controlled by	Consistency	Supported by
Buffered	Host (Application/Driver)	Yes	Master & Slave Firmware

Buffered, Host Controlled Mode

The Buffered data transfer mode can be used for both master- and slave- type devices. In "buffered" mode, the protocol stack handles the exchange of data between internal buffers and the process data images in the dual-port memory with the application via a handshake mechanism. Once copied from/into the input/output area, the host application gives control over the dual-port memory to the protocol stack. Control is given back to the host application when the protocol stack has finished copying, and so on.

The network cycle and the task cycle of the host application are not synchronized, but are consistent.



Note

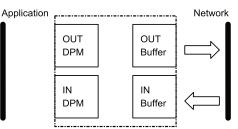
If the host application is faster than the network cycle, it might be possible that data in the output buffers is overwritten without ever being sent to the network. As for the other direction, the host application may read the same input values over several read cycles.

If the host application is slower than the network cycle, the protocol stack overwrites the input buffer with new data received from the network, which were never received by the host application. The output data on the network will be the same over several network cycles.

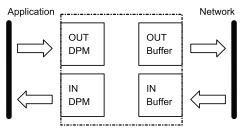
For each valid bus cycle, the protocol stack updates the process data in the internal input buffer. When the application toggles the appropriate input handshake bit, the protocol stack copies the data from the internal IN buffer into the input data image of the dual-port memory. Now the application can copy data from the dual-port memory and then give control back to the protocol stack by toggling the appropriate input handshake bit. When the application/driver toggles the output handshake bit, the protocol stack copies the data from the output data image of the dual-port memory into the internal buffer. From there, the data is transferred to the network. The protocol stack toggles the appropriate handshake bits back, indicating to the application that the transfer is finished and a new data exchange cycle may start. This mode guarantees data consistency over both the input and output areas.

Step-by-Step Procedure

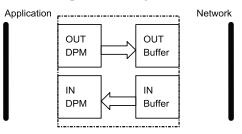
Step 1 The protocol stack sends data from the internal OUT buffer to the network and receives data from the network in the internal IN buffer.



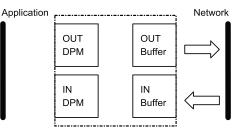
Step 2 The application has control over the dual-port memory and exchanges data with the input and output data images in the dual-port memory. The application then toggles the handshake bits, giving control over the dual-port memory to the protocol stack



Step 3 The protocol stack copies the content of the output data image into the internal OUT buffer, and from the IN buffer to the input data image.



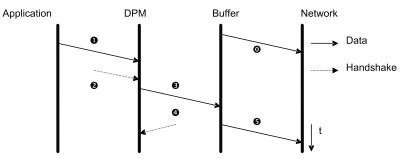
Step 4 The protocol stack toggles the handshake bits, giving control back to the application. Now, the protocol stack uses the new output data image from the OUT buffer to send it to the network, and receives data into the internal IN buffer, and then the cycle repeats.



Time-Related View

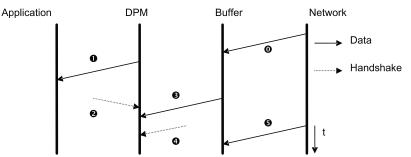
The following figure shows the procedure in a time-related view.

Output Data Exchange



- 1. The protocol stack constantly transmits data from the buffer to the network.
- 2. The application has control over the dual-port memory and can copy data to the output data image.
- 3. The application then toggles the handshake bits, giving control over the dual-port memory to the protocol stack.
- 4. The protocol stack copies the content of the output data image into the internal OUT buffer.
- 5. The protocol stack toggles the handshake bits, giving control back to the application.
- 6. Once updated, the protocol stack uses the new data from the internal buffer and sends it to the network. The cycle repeats with step 1.

Input Data Exchange



- 1. The protocol stack constantly receives data from the network into the buffer.
- 2. The application has control over the dual-port memory input data image and exchanges data with the input data image in the dual-port memory.
- 3. The application then toggles the handshake bits, giving control over the dual-port memory to the netX protocol stack.
- 4. The protocol stack copies the latest content of the internal IN buffer to the input data image of the dual-port memory.
- 5. The protocol stack then toggles the handshake bits, giving control back to the application.
- 1. The protocol stack receives data from the network into the buffer (i.e. the cycle starts over with the first step).



In case of a network fault (e.g. disconnected network cable), a slave firmware keeps the last state of the input data image. As soon as the firmware detects the network fault, it clears the Communicating flag in the netX communication flags. The input data should then no longer be evaluated.

Start / Stop Communication

Controlled or Automatic Start

The firmware has the option to start network communication after power up automatically. Whether or not the network communication will be started automatically is configurable. However, the preferred option is called "Controlled Start of Communication." This option forces the channel firmware to wait for the host application to allow network connection being opened by setting the Bus On flag in the Application Change of State register in the channel's control block. Consequently, the protocol stack will not allow the opening of network connections and does not exchange any cyclic process data until the Bus On flag is set.

The second option enables the channel firmware to open network connections automatically without interacting with the host application. It is called "Automatic Start of Communication." This method is not recommended because the host application has no control over the network connection status. In this case, the Bus On flag is not evaluated.



The Controlled Start of communication is the default method used for the default dual-port memory layout.

Start / Stop Communication through Dual-Port Memory

(Re-)Start Communication

To allow the protocol stack to open connections or to allow connections to be opened, the application sets the Bus On flag in the Application Change of State register in the channel's control block. When firmware has established a cyclic connection to at least one network mode, the channel firmware sets the Communicating flag in the netX Communication Flags register.

Stop Communication

To force the channel firmware to disable all network connections, the host application clears the Bus On flag in the "Application Change of State" register in the channel's control block. The firmware then closes all open network connections. A slave protocol stack would reject attempts to reopen a connection until the application allows opening network connections again (Bus On flag is set). When all connections are closed, the channel firmware clears the Communicating flag in the netX Communication Flags register.

Reset Command

System Reset vs. Channel Initialization

There are several methods to restart the netX firmware. The first is called "System Reset." The System Reset affects the netX operating system, rcX, and the protocol stacks. It forces the chip to immediately stop all running protocol stacks and the rcX itself. During the system reset, the netX is performing an internal memory check and other functions to insure the integrity of the netX chip itself.

The Channel Initialization, as the second method, affects a communication channel only. The channel firmware then reads and evaluates the configuration settings (or SYCON.net database, if available) again. The operating system is not affected. There are no particular tests performed during a channel initialization.

A third method to reset the netX chip is called Boot Start. No firmware is started when a System Reset is executed with the boot start flag set. The netX remains in boot loader mode.

A System Reset, Channel Initialization, and boot start may cause all network connection to be interrupted immediately, regardless of their current state.



During a HW-Reset and the time when the 2nd stage loader starts the Firmware, the content of the dual port memory can be 0xFFFF or 0x0BAD for a short period of time.

When used with Turbo PMAC2 CPU, it is necessary to reset the COMX module for proper functionality after initial power up, cycle power, or a \$\$\$ or \$\$\$*** command.

Resetting netX through Dual-Port Memory

To reset the entire netX firmware, the host application has to set the HSF_RESET bit in the bHostSysFlags register to perform a system-wide reset and respectively the APP_COS_INIT flag for a channel initialization in the ulApplicationCOS variable in the control block of the channel. The system reset and the channel initialization are handled differently by the firmware (see above).

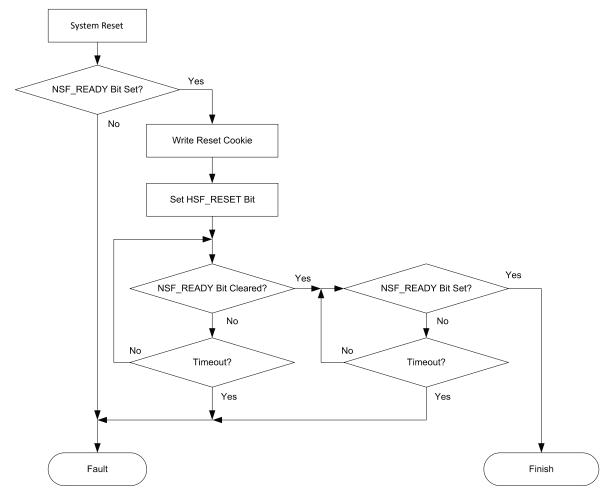
System Reset

To reset the netX operating system rcX and all communication channels, the host application has to write \$55AA55AA (System Reset Cookie) to the ulSystemCommandCOS variable in the system control block. Then, the HSF_RESET flag in bHostSysFlags has to be set. If the operating system does not find \$55AA55AA in the ulSystemCommandCOS variable, the reset command will be ignored.

The operating system clears the NSF_READY flag in bNetxFlags in the system handshake register, indicating that the system-wide reset is in progress. During the reset, all communication channel tasks are stopped, regardless of their current state. The rcX operating system flushes the entire dual-port memory and writes all memory locations to zero. After the reset, if rcX is finished without complications, and all protocol stacks are started properly, the NSF_READY flag is set again. Otherwise, the NSF_ERROR flag in bNetxFlags in the system handshake register is set, and an error code is written in ulSystemError in the system status block (see page 46), which helps identify possible problems.

Value	Definition/Description
\$55AA55AA	System reset cookie

The image below illustrates the steps the host application has to perform in order to execute a systemwide reset on the netX chip through the dual-port memory.



Timing

The duration of the reset outlined above depends on the firmware. Typically, the NSF_READY flag is cleared within around 100 - 500 ms after the HSF_RESET Flag was set. When cleared, the NSF_READY bit will be set again after around 0.5 - 5 s. Generally, the reset should not take more than 6 seconds.

Channel Initialization

In order to force the protocol stack to restart and evaluate the configuration parameter again, the application can set the APP_COS_INIT flag in the ulApplicationCOS register in the control block or send a reset packet to the communication channel. All open network connections are interrupted immediately, regardless of their current state. Reinitializing the channel is not allowed if the database is locked.

Changing flags in the ulApplicationCOS register requires the application also to toggle the host change of state command flag in the host communication flags register. Only then, the netX protocol stack recognizes the reset command.

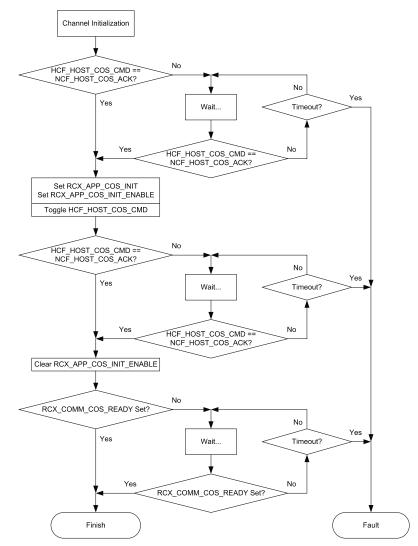
Below is the sequence:

CC0_RCX_APP_COS_INIT=1 CC0_RCX_APP_COS_INIT_ENABLE=1 HCCC0_HCF_HOST_COS_CMD=1

During channel initialization, the RCX_COMM_COS_READY flag and the RCX_COMM_COS_RUN flag are cleared together. The RCX_COMM_COS_READY flag stays cleared for at least 20 ms before it is set again, indicating that the initialization has finished. The RCX_COMM_COS_RUN flag is set if a valid configuration was found. Otherwise, it stays cleared.

After the initialization process has finished, the protocol stack checks ulApplicationCOS register. If the RCX_APP_COS_BUS_ON flag and the RCX_APP_COS_BUS_ON_ENABLE flags are set, network communication will be restored automatically. The same is true for the Lock Configuration feature (RCX_APP_COS_LOCK_CONFIG / RCX_APP_COS_LOCK_CONFIG_ENABLE) and the DMA data transfer mechanism (RCX_APP_COS_DMA / RCX_APP_COS_DMA_ENABLE).

The image below illustrates the steps the host application has to perform in order to execute a channel initialization on the protocol stack through the dual-port memory.



System Reset through Packets

The netX chip can be reset using a packet instead of the dual-port memory. The request packet is passed through the system mailbox. All open network connections are interrupted immediately, regardless of their current state. Reinitializing the channel is not allowed if the database is locked.

For detailed information about reset message settings, please see Hilscher documentation.

SOFTWARE SETUP

ACC-72EX supports multiple protocols, and setting up each protocol can be a bit different, as described in the protocol specific documentation provided by Hilscher. In this section, most of the generic steps are covered with the help of examples and screenshots.

Required Software Packages

Two software packages are required for setting up ACC-72EX:

- 1. SYCON.NET (V1.0310.x.x or newer), available through Hilscher's website.
 - a. If using newer ACC-72Ex modules, V1.0500.230227.42617 is required.
- 2. ACC-72EX Setup Assistant Software.

Both software packages have to be installed on the PC used for initial setup of the system and commissioning of the machine. Notice that neither of these software packages is required after the initial setup and the unit can work as a standalone setup.

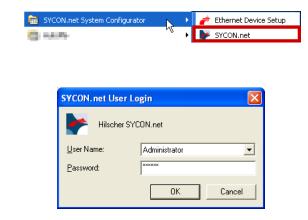
SyCon.NET Software Setup

SYCON.net is a tool for the configuration of Fieldbus and Real-Time Ethernet systems. It is based on the standardized FDT / DTM technology. Online diagnostic indicators and auto-scan function for the reading of network participants assist in the commissioning of the network. SYCON.NET is provided with the gateway module under license from Hilscher Corporation.

As of May 2023, the Profibus Slave and DeviceNet Slave options have changed slightly due to a change in the underlying module, from the COMX10 module to the COMX52 module. Setup and configuration are nearly identical, though users may require a newer version of the Sycon.Net software to support this. Version V1.0500.230227.42617 of the Sycon.Net software is available from the Knowledge Base on Hilscher's website and has been tested as compatible.

Where setup varies, new screenshots and descriptions have been provided below.

With the power off, plug the ACC-72EX into the UBUS backplane and turn on the power to the UMAC rack. Connect the diagnostic port to a USB port on the PC using a micro-USB type cable. Launch the SYCON.NET software on the PC.

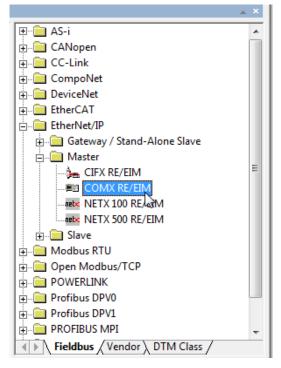


Enter the password:

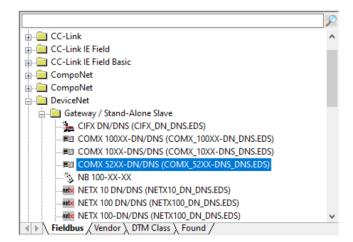
SYCON. net - [Untitled.spj]		
Eile ⊻iew Device Network E⊻tras Help	6	
] D 🗳 🔲 🕄 🛛 🖆 🖆 🔕 🕕 🐀 📟 📑 📆 📆 📆		
		×
Project: Untitled	AS-i CANopen C-Link	E
	SERCOS III Fieldbus / Vendor \ DTM Class / A5-i	•
The sycon.net / netDevice /	<u></u>	Þ
Ready	Administrator	

Start a new project or load an existing project from the File menu:

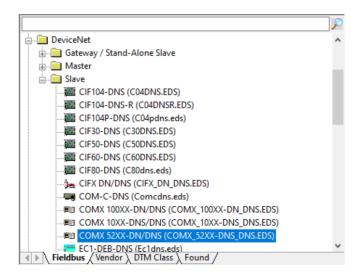
Select the COMX module to which the USB is connected from the Fieldbus protocol list. In this example, an EtherNet/IP module has been selected:



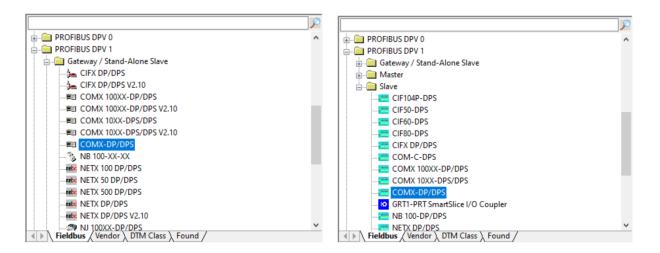
If using the COMX52 module as a Stand-Alone DeviceNet Slave, select that item from the "Gateway / Stand-Alone Slave" folder.



If using the COMX52 module as a slave device (and another Hilscher module as a master device), select that item from the "Slave" folder.



For Profibus, there is no option for a "COMX-52" module in either the "Gateway / Stand-Alone Slave" or "Slave" folders, so instead select "COMX-DP/DPS" from the appropriate location if using this module.



Drag and drop the module onto the BusLine in the netDevice window (notice that the module can only be inserted on the BusLine).

netDevice	
netDevice	
COMX_RE_EIM[COMX RE/EIM]<192.168.10.1>(#1)	

Establish USB communication to the COMX gateway by right-clicking on the device icon and selecting "Configuration...":

сом	X_RE_EIM[COMX RE/EIM]<192.168.10.1>(#1)
	Connect Disconnect	
	Download Upload	
	Cut Copy	
	Paste	
	Configuration	
	Measured Value • • Simulation Diagnostic	
	Additional Functions 🕨	
	Delete	
	Symbolic Name	

In the netDevice Configuration window, select the Driver folder under Settings folder in the NavigationArea, check the checkmark box for netX Driver on the driver list, and click Apply:

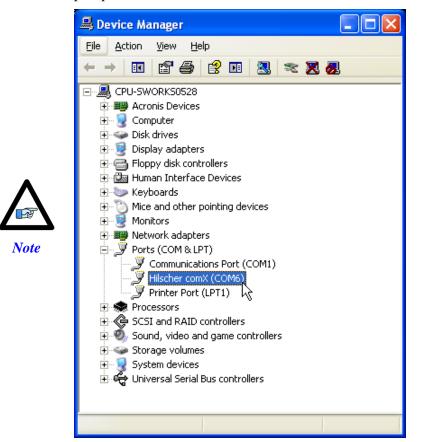
RetDevice - Configuration COMX_RE_EIM[COMX RE/EIM]<192.168.10.1>(#1)							
IO Device: COM Vendor: Hilscl	Device ID: 0×0104 Vendor ID: 0×0118			Fot			
Navigation Area 📃							
🔄 Settings			[
🔄 Driver		Driver	Version	ID			
netX Driver		35Gateway Driver for netX (V3.x)	0.9.1.2	{787CD3A9-4	4CF6-4259-8E4D	109B6A6BEA91}	
Device Assignment		netX Driver	1.101.1.5347	{B54C8CC7-F	F333-4135-8405-	6E12FC88EE62}	
Firmware Download							
Configuration							
Network Settings							
Scanlist							
Process Data							
Address Table							
Scanner Settings							

If a "netX SPM USB Driver" is also available, select it, too.

▶ netDevice - Configuration (OMX 52	XX-DN/DNS (COMX_52XX-DNS_DN	S.EDS)[COMX 52XX-DI	N/DNS (COMX_5	2XX-DNS_D	-		×
	(52XX-DI Ier GmbH	N/DNS (COMX_52XX-DNS_DNS.EDS)		Device ID: Vendor ID:	61 283			FÓT
Navigation Area			Driver	r				
Settings		Driver	Version	ID				
netX Driver		netX SPM USB Driver	1.0.5.8878	{9634996A-AE	A6-42FA-BF8D-5	758AED2	2D303}	
Device Assignment		netX Driver	1.200.7.3154	{B54C8CC7-F3	33-4135-8405-68	E12FC88	EE62}	
Firmware Download								
Configuration								
General								
Connection								

Select the netXDriver node under the Driver folder in the Navigation Area and select the port resembling the USB connection to the COMX module. Click Save and Apply (just click OK if Apply is grayed out).

א netDevice - Configurati	on COMX_RE_EIM[COMX RE/EIM]<192.168.1	10.1>(#1)	
	MX RE/EIM cher GmbH	Device ID: Vendor ID:	0x0104 0x011B
Navigation Area Settings Driver → netX Driver Device Assignment Firmware Download Configuration Network Settings Scanlist Process Data Address Table Scanner Settings	USB/RS232 Connection TCP Connection Enable USB/RS232 Connector (Restart of ODM Select Port: COM6 Port Configuratio COM1 Disable Port Baud Rate: 115.2 kBit/s Stop Bits: 1 Stopbit Send Timeout: 1000 * ms Reset Timeout: 1000 * ms	required) Byte Size: Parity: Keep Alive Timeout: Restore	8 Byte No Parity 2000 ★ ms
\$)⊳ Disconnected 🚺 Data Set		ОК	Cancel Apply Help



Check Windows Device Manager in order to identify which COM port provides the connection to the Hilscher COMX module.

Click Device Assignment under the Driver folder in the Navigation Area. Assign the netX Driver to the detected COMX module by checking the checkmark box next to the detected device, and click Apply.

😽 netDevice - Configurat	ion COMX_RE_E	IM[COMX RE/EI	M]<192	2.168.10	.1>(#1)		
	MX RE/EIM				Device ID:	0×0104	
Yendor: Hils	cher GmbH				Vendor ID:	0×011B	FDT
Navigation Area 📃							
Settings	Scan progress: 2/	/2 Devices (Current	device: -)			
netX Driver							Scan
Device Assignment Firmware Download	Device selection:	suitable only	/ -				. <u></u>
	Device	Hardware Por	Slot	Serial	Driver	Channel Protocol	Access path
Network Settings	Сомх	Ethernet/Ethe	n/a	21456	netX Driver	EtherNet/IP Scanner	\COM6
Scanlist							
Process Data							
Address Table							
Scanner Settings							
	Access path:	{B54C8CC	7-F333-4	135-8405-6	5E12FC88EE62}\	COM6_cifX0_Ch0	
				Γ	ОК	Cancel Appl	y N Help
🕸 Disconnected 🛛 🚺 Data Se	t 🥖 🗌						

When used with Turbo PMAC, the reset line is released too fast for some Hilscher COMX modules, which puts them in a boot mode. This can prevent the device from being detected by Sycon.NET software. Make sure the device receives a system-wide reset using the PMAC suggested M-Variables ulSystemCommandCOS and HSF_RESET registers as shown here.

Note

SCtrl_ulSystemCommandCOS=\$55AA55AA

HCSC_HSF_RESET=1

Note that ACC-72EX Setup Assistant software automatically resets the cards if it cannot detect the identification cookie.

The rest of the steps are protocol/module dependent, and it is strongly recommended to follow the directions for these modules in Hilscher documentation available through their website. The current example will be continued with specifics to EtherNet/IP Scanner/Adapter setup.

Now that the COMX driver for communication between the PC and COMX module using the diagnostic port has been set up, go through protocol specific setup parameters under the Configuration folder in the Navigation Area.

RetDevice - Configurat	ion COMX_RE_EIM[COM)	RE/EIM]<192.168.10	.1>(#1)		
	MX RE/EIM scher GmbH		Device ID: Vendor ID:	0x0104 0x011B	FDT
Navigation Area 📃					
Settings Triver netX Driver	Description: COM	(_RE_EIM			
Device Assignment Firmware Download	IP Settings				
	🔲 <u>р</u> нср				
Network Settings Scanlist	BootP				
Process Data Address Table	Fixed Addresses				
Scanner Settings	IP Address:	192 . 168 . 10), 1		
	<u>N</u> etwork Mask:	255 . 255 . 25	5.0		
	Gateway Address:	0.0.0	. 0		
	Note: The priority sequer	nce is DHCP, BootP, Fixed.			
	Operation mode: All cap	oable, Auto Negotiation enab	oled		•
			ок	Cancel Apply	Help
↓Disconnected Data Set					

After finishing modifying the settings for the device, press the OK button.

Back in the netDevice tree, right click on the device icon, and select Connect (as shown below).

netDevice		
COMX_RE_EIM[COMX RE/EIM]<192.168.10.1>(#1)	
COMX	RE_EIS[COMX RE/EIS]<192.168.10.2> Connect Disconnect Disconnect Upload Cut Copy Paste	
	Configuration Measured Value	
	Simulation	
	Diagnostic	
	Additional Functions	
	Delete	
	Symbolic Name	
<		>

Once connected, right click on the device icon one more time and select Download (as shown below). This will download all the configurations from PC to COMX module.

COMX_RE_EIM	1[COMX RE/EIM]<192.168.10.1>(#1)
	1X_RE_EIS[COMX RE/EIS]<192.168.10.2>
1	5 C
	Connect
	Disconnect
	Download
	Upload ^K
	Cut
	Copy Paste
	Configuration
	Measured Value
	Simulation
	Diagnostic
	Additional Functions 🔸
	Delete
	Symbolic Name

Once the configuration is downloaded to the COMX module, make sure to save the SYCON.net project for later use.

The Hilscher slave module (COMX_RE_IES) above was dragged and dropped from the fieldbus protocol list. Third party slave modules can be added to that list by going to "Import Device Descriptions..." in the Network tab:

File View Device	Net	Network Extras Help						
🗅 🚅 🔚 😰 🕯	랔	Add	Busline					
netProject	2-	Delete Last Busline						
🖃 💼 Project: Untitled	3a	强 Start Project Debug Mode						
🚊 🕮 Comx_re_ei	STOP	Stop	Project [Debug Mode				
COMX_R	ECOMX_R 🔄 Device Catalog							
	Import Device Descriptions							
	Print Project Data							

See Appendix C for an example setup using an ACC-72EX Ethernet IP slave with a third party Ethernet IP master PLC controller.

ACC-72EX Setup Assistant

The next step is to generate the memory map and suggested M-Variables for the Hilscher module. Run the ACC-72EX Setup Assistant. In the Memory Map Generator groupbox, click the Connect to PMAC button, and select the UMAC where the ACC-72EX is installed. Once connected, the software will detect any available ACC-72EX(s) in the rack and list it based upon the base address(es).

ACC-72EX Setup Assistant		
Memory Map Generator	Address Converter	
Connect to PMAC	ACC-72EX Base Address \$6C000	~
	Hilscher Address Offset 0x0	
Address Selection \$6C000 👻	Hilscher Data Width 32	~
Starting M-Variable Number 6000 😂	Hilscher Data Start Bit 0	~
	Convert	
Generate M-Variable Definitions	Equivalent PMAC Address D:\$6C000	
No PMAC is selected for communication. Connection to PMAC was successful. 2x ACC-72EX cards detected.		
Connected		

Select the starting number for M-Variable assignment, and click the "Generate M-Variable Definitions" button. The program asks for a folder location to save the M-Variable definition and memory map files.



This will generate three files which are named based upon the ACC-72EX base address.

😂 ACC-72EX Output				
<u>File E</u> dit <u>V</u> iew F <u>a</u> vori	tes <u>T</u> ools	Help		1
🌀 Back 🝷 🕥 🕤 🗗	🏂 🔎 si	earch	6 Folders	•
Address			1	🖌 🔁 Go
HilscherMemoryMap_\$6C0	00.txt			
MacroNameDefinition_\$6C	000.h			
M-VariableDefinition_\$6C0	00.pmc			
3 objects	26.7 KB	5	My Computer	

The M-variable definition and its header file can be used in writing PLCs and motion programs in PMAC. The memory map file is useful for identifying the process data image locations.

Turbo PMAC Setup for Using ACC-72EX

All interactions between PMAC and ACC-72EX occur through M-variables. Most of the important registers which are required are mapped in the suggested M-Variable definition files generated by ACC-72EX Setup Assistant software. The generated files can be included in the header section of the project files in PEWIN32PRO2.



If multiple ACC-72EX cards are in the same UMAC rack, the Mvariable macro names will be identical for both files, despite the file name difference based upon the ACC-72EX base address. Make sure to add a prefix or suffix to the macro names both in the header file and definition file in order to distinguish the proper macro names for different ACC-72EXs. Note that no macro name should be longer than 32 characters.

There are multiple steps in getting the COMX module working with the network/fieldbus. Some of these steps are protocol-specific, and it is recommended to follow the requirements based upon each protocol manual provided by Hilscher.

Initialization PLC

Recall that ACC-72EX requires a reset after each power up, power cycle, \$\$\$ (reset), or \$\$\$*** (factory default reset). This can be achieved with a startup (or initialization) PLC. Example:

```
CLOSE
END GAT
DEL GAT
#include "M-VariableDefinition $6C000.pmc"
#include "M-VariableDefinition $74000.pmc"
#define CommErrorFlag P1
OPEN PLC 1 CLEAR
DISABLE PLC 2..31
                                              // Disable all other tasks
SCtrl ulSystemCommandCOS=$55AA55AA
                                              // Reset token for MASTER Unit
HCSC HSF RESET=1
                                              // Reset bit, token required for reset to complete
S SCtrl ulSystemCommandCOS=$55AA55AA
                                              // Reset token for SLAVE Unit
S HCSC HSF RESET=1
                                              // Reset bit, token required for reset to complete
CommErrorFlag=0
timer = 1000 msec
                                              // Reset Time-out Timer
WHILE (CommErrorFlag=0 AND HCSC NSF READY=0) // Wait for reset to complete
       IF (timer<0)
                                              // Check for reset timeout
               CommErrorFlag = 1
       ENDIF
ENDWHILE
                                                             11
IF (CommErrorFlag=0)
       WHILE (CC0 RCX COMM COS RUN=0 OR S CC0 RCX COMM COS RUN=0)
                                                                     // wait for comm tasks to
                                                                     //\ {\rm start} on COMX modules
                       HCCC0 HCF NETX COS ACK = HCCC0 HCF NETX COS ACK ^ 1
                       // Toggle Communication Channel 0's Change of State Acknowledge bit in
                       // order to read the CCO RCX COMM COS RUN which is a part of Communication
                       // Channel 0 State Register
                       S HCCCO HCF NETX COS ACK = S HCCCO HCF NETX COS ACK ^ 1
       ENDWHILE
       ENABLE PLC 28
       ENABLE PLC 10
       ENABLE PLC 11
ENDIF
DISABLE PLC 1
CLOSE
```

Watchdog Function

The host Watchdog and the device Watchdog cells in the control block of each of the communication channels allow the operating system running on the netX to supervise the host or UMAC application and vice versa. There is no Watchdog function for the system block or for the handshake channel. The Watchdog for the channels is located in the control block of the status block of each communication channel.

The netX firmware reads the contents of the device Watchdog cell, increments the value by one, and copies it back into the host Watchdog location. Then, the application has to copy the new value from the host Watchdog location into the device Watchdog location. Copying the host Watchdog cell to the device Watchdog cell has to happen in the configured Watchdog time. When the overflow occurs, the firmware starts over and "1" appears in the host Watchdog cell. A zero turns off the Watchdog and therefore never appears in the host Watchdog cell in the regular process.

The minimum Watchdog time is 20 ms. The application can start the Watchdog function by copying any value unequal to zero into device Watchdog cell. A zero in the device Watchdog location stops the Watchdog function. The Watchdog timeout is configurable in SYCON.net and can be downloaded to the netX firmware.

If the application fails to copy the value from the host Watchdog location to the device Watchdog location within the configured Watchdog time, the protocol stack will interrupt all network connections immediately, regardless of their current state. If the Watchdog tripped, then power cycling, channel reset, or channel initialization will allow the communication channel to open network connections again.

Here is sample code for copying the host Watchdog location to the device Watchdog location:

```
CLOSE
END GAT
DEL GAT
#include "M-VariableDefinition_$6C000.pmc"
#include "M-VariableDefinition_$74000.pmc"
OPEN PLC 28 CLEAR
CC0_ulDeviceWatchdog = CC0_ulHostWatchdog // copies the host Watchdog content
// to device Watchdog cell
// for the 1st ACC-72EX
S_CC0_ulDeviceWatchdog = S_CC0_ulHostWatchdog // copies the host Watchdog content
// to device Watchdog content
// to device Watchdog cell
// for the 2nd ACC-72EX
CLOSE
```

Enabling the Communication Bus

Using the Bus On flag (CCx_RCX_APP_COS_BUS_ON, where x is the communication channel number), the host or UMAC application allows or disallows the netX firmware to open network connections. This flag is used together with the Bus On Enable flag

(CCx_RCX_APP_COS_BUS_ON_ENABLE, where x is the communication channel number). If set, the netX firmware tries to open network connections; if cleared, no connections are allowed, and open connections are closed. If the Bus On Enable flag is set, it enables the execution of the Bus On command in the netX firmware:

CC0_RCX_APP_COS_BUS_ON=1 CC0_RCX_APP_COS_BUS_ON_ENABLE=1	<pre>// Setting the Bus On flag for 1st ACC-72EX // Enabling the execution of Bus On Flag for 1st ACC-72EX</pre>
S_CC0_RCX_APP_COS_BUS_ON=1	// Setting the Bus On flag for 2nd ACC-72EX
S_CC0_RCX_APP_COS_BUS_ON_ENABLE=1	// Enabling the execution of Bus On Flag for 2nd ACC-72EX

Locating the Input/Output Data Image in PMAC

Although the ACC-72EX Setup Assistant software defines M-Variables for accessing setup registers and flags in COMX modules, it does not assign any M-Variables for input/output data images. However, starting address and size of each input/output processed data image in's PMAC memory addressing format are calculated and included as a part of the memory map file that is generated. The following is an example from an EtherNet/IP option. The highlighted sections show the addressing for the processed data images:

Block 2: Channel Type: Size of Channel: Channel Start Address: Position of Handshake (Size of Handshake Cells NetX Handshake Register Host Handshake Register Communication Class: Protocol Class: Conformance Class: Number of Subblocks:	r: Y:\$6C082,0,16
	8 bytes \$6C0C2 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0
Transfer Type:	ATUS 64 bytes \$6C0C4 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0
	432 bytes \$6C0D4 IN (netX to Host System) DPM (Dual-Port Memory)
Subblock 3: MAILBOX Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit:	1600 bytes \$6C140 OUT (Host System to netX) DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED 4
Subblock 4: MAILBOX Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit:	1600 bytes \$6C2D0 IN (netX to Host System) DPM (Dual-Port Memory) UNKNOWN 5
Subblock 5: PROCESS D2 Size: Start Offset: Transfer Direction: Transfer Type: Handshake Mode: Handshake Bit:	ATA IMAGE 5760 bytes <mark>\$6C4C0</mark> OUT (Host System to netX) DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED 6

```
--- Subblock 6: PROCESS DATA IMAGE
       Size:
                              5760 bytes
       Size: 5760 by
Start Offset: $6CA60
       Transfer Direction: IN (netX to Host System)
       Transfer Type: DPM (Dual-Port Memory)
Handshake Mode: BUFFERED, HOST CONTROLLED
                            7
       Handshake Bit:
|--- Subblock 7: HIGH PRIORITY DATA IMAGE
       Size: 64 bytes
Start Offset: $6C460
       Size:
       Transfer Direction: OUT (Host System to netX)
       Transfer Type: DPM (Dual-Port Memory)
Handshake Mode: BUFFERED, HOST CONTROLLED
       Handshake Bit:
                           8
|--- Subblock 8: HIGH PRIORITY DATA IMAGE
       Size: 64 bytes
Start Offset: $6C470
       Transfer Direction: IN (netX to Host System)
       Transfer Type: DPM (Dual-Port Memory)
       Handshake Mode: BUFFERED, HOST CONTROLLED
       Handshake Bit:
                             9
```

Depending on the protocol, users might be interested in:

- Processed Data Images
- High Priority Data Images
- Mailboxes

Also listed in the memory map are starting address, size of each of these memory blocks, handshake method, and flag.

Reading/Writing from/to Input/Output Data Images

There are two methods for accessing processed data images:

1. Direct M-Variable definition to each register

This method is useful if the number of I/O data variables is small enough

2. Indirect M-Variable access

This method is mostly used if the number of I/O data count is greater than a comfortable level which can be handled by the direct M-Variable definition method. Refer to the Turbo PMAC Users Manual for detailed information on how to utilize the indirect addressing method.

This example demonstrates a 16-bit integer register transfer. Notice that only the first 16-bit portion of the integer in P200 will be transferred.

CLOSE END GAT DEL GAT #include "M-VariableDefinition \$6C000.pmc" #include "M-VariableDefinition_\$74000.pmc" #define Master OutputData1 M2000 M2001 #define Master InputData1 #define Slave_OutputData1
#define Slave_InputData1 M2002 M2003 // Pointer to byte 0 and 1 of Output Data Image Master OutputData1->Y:\$6C4C0,0,16,S // of Communication Channel 0 on Master COMX module Master InputData1->Y:\$6CA60,0,16,S // Pointer to byte 0 and 1 of Input Data Image // of Communication Channel 0 on Master COMX module Slave OutputData1->Y:\$744C0,0,16,S // Pointer to byte 0 and 1 of Output Data Image // of Communication Channel 0 on Slave COMX module // Pointer to byte 0 and 1 of Input Data Image Slave InputData1->Y:\$74A60,0,16,S // of Communication Channel 0 on Slave COMX module P200=0 OPEN PLC 10 CLEAR IF (HCCC0 HCF PD0 OUT CMD = HCCC0 NCF PD0 OUT ACK) // Making sure the ACK flag matches the CMD // flag before writing the value to the // output data image register P200=P200+1 Master OutputData1 = P200 // Copy the value to register HCCC0 HCF PD0 OUT CMD = HCCC0 HCF PD0 OUT CMD^1 // Toggle the CMD flag (^: XOR) ENDIF CLOSE

In a similar approach the data can be read from an input data image:

```
OPEN PLC 11 CLEAR

IF (HCCC0_NCF_PD0_IN_CMD = HCCC0_HCF_PD0_IN_ACK) // If CMD flag and ACK flags are

// equal, then the input data image

// register can be read

// read the input data image register

// toggle the acknowledge bit

// toggle the acknowledge bit

// indicating read completion
```

Notice that depending on M-Variable definition, different types of data formats can be transferred over the DPR and network:

Power PMAC Setup for Using ACC-72EX

Power PMAC has full support for ACC-72EX and all its fieldbus communication variations. Due to builtin data structures for accessing ACC-72EX dual ported RAM from Power PMAC, no additional software is required for memory mapping and/or identification in comparison to Turbo PMAC.

This section of the manual covers Power PMAC's built in data structures for ACC-72EX in addition to providing examples for header files, start-up and handshaking PLCs.

ACC72EX[i]. Non-Saved Data Structures

All of the interactions with ACC-72EX can be achieved through data structures defined specifically for ACC-72EX in Power PMAC firmware. The following structures allow access to the DPRAM in bit, byte, 2-byte and 4-byte wide access modes. The bit-wise read and write is only supported through Acc72EX[*i*].Udata16[*j*] data structure.

Acc72EX[*i*].Data8[*j*]

Description: Dual Ported RAM "unsigned 8-bit integer" data array element

Range: $0 .. 2^{8} - 1$

Units: address dependent

Power-on default: address dependent

Acc72Ex[*i*].Data8[*j*] is the "*j*th" unsigned 8-bit integer data array element in the Acc72EX[*i*] dual-ported RAM. Each of these elements occupies one byte in the DPRAM, and is located starting at *j* addresses past the beginning of the buffer (which is located at the address in Acc72EX[*i*].a). This array is defined based upon the Hilscher ComX memory map.

Index values j in the square brackets can be integer constants in the range 0 to 524,287, or local L-variables. No expressions or non-integer constants are permitted. The size of the DPRAM is dependent on the ACC-72EX communication option and installed Hilscher ComX module.

Acc72Ex[*i*].Data8[*j*] is located in the same registers as Acc72Ex[*i*].Idata16[*j*/2], Acc72Ex[*i*].Udata16[*j*/2], Acc72Ex[*i*].Idata32[*j*/4], Acc72Ex[*i*].Idata32[*j*/4] and Acc72Ex[*i*].Udata32[*j*/4]. It is the user's responsibility to prevent possible multiple uses of the same

register.

In C, this element should be accessed through the C functions ACC72EX_GetData8 and ACC72EX_SetData8 described later in this manual.

Acc72EX[i].Idata16[j]

Description: Dual Ported RAM "signed 16-bit integer" data array element

Range: $-2^{15} .. 2^{15} -1$

Units: address dependent

Power-on default: address dependent

Acc72Ex[*i*].Idata16[*j*] is the "*j*th" signed 16-bit integer data array element in the Acc72EX[*i*] dual-ported RAM. Each of these elements occupies two bytes in the DPRAM, and is located starting at 2**j* addresses past the beginning of the buffer (which is located at the address in Acc72EX[*i*].a). This array is defined based upon the Hilscher ComX memory map.

Index values j in the square brackets can be integer constants in the range 0 to 262,143, or local L-variables. No expressions or non-integer constants are permitted. The size of the DPRAM is dependent on the ACC-72EX communication option and installed Hilscher ComX module.

Acc72Ex[*i*].Idata16[*j*] is located in the same registers as Acc72Ex[*i*].Data8[2**j*] to Acc72Ex[*i*].Data8[2**j*+1], Acc72Ex[*i*].Udata16[*j*], Acc72Ex[*i*].Idata32[*j*/2] and Acc72Ex[*i*].Udata32[*j*/2]. It is the user's responsibility to prevent possible multiple uses of the same register.

In C, this element should be accessed through the C functions ACC72EX_GetIdata16 and ACC72EX_SetIdata16 described later in this manual.

Acc72EX[*i*].Udata16[*j*]

Description: Dual Ported RAM "unsigned 16-bit integer" data array element

Range: $0 .. 2^{16} - 1$

Units: address dependent

Power-on default: address dependent

Acc72Ex[*i*].Udata16[*j*] is the "*j*th" unsigned 16-bit integer data array element in the Acc72EX[*i*] dualported RAM. Each of these elements occupies two bytes in the DPRAM, and is located starting at 2**j* addresses past the beginning of the buffer (which is located at the address in Acc72EX[*i*].a). This array is defined based upon the Hilscher ComX memory map.

Index values j in the square brackets can be integer constants in the range 0 to 262,143, or local L-variables. No expressions or non-integer constants are permitted. The size of the DPRAM is dependent on the ACC-72EX communication option and installed Hilscher ComX module.

Acc72Ex[*i*].Udata16[*j*] is located in the same registers as Acc72Ex[*i*].Data8[2**j*] to Acc72Ex[*i*].Data8[2**j*+1], Acc72Ex[*i*].Idata16[*j*], Acc72Ex[*i*].Idata32[*j*/2] and Acc72Ex[*i*].Udata32[*j*/2]. It is the user's responsibility to prevent possible multiple uses of the same register.

In C, this element should be accessed through the C functions ACC72EX_GetUdata16 and ACC72EX_SetUdata16 described later in this manual.

Acc72EX[/].Idata32[/]

Description: Dual Ported RAM "signed 32-bit integer" data array element

Range: $-2^{31} .. 2^{31} - 1$

Units: address dependent

Power-on default: address dependent

Acc72Ex[*i*].Idata32[*j*] is the "*j*th" signed 32-bit integer data array element in the Acc72EX[*i*] dual-ported RAM. Each of these elements occupies four bytes in the DPRAM, and is located starting at 4**j* addresses past the beginning of the buffer (which is located at the address in Acc72EX[*i*].a). This array is defined based upon the Hilscher ComX memory map.

Index values j in the square brackets can be integer constants in the range 0 to 131,072, or local L-variables. No expressions or non-integer constants are permitted. The size of the DPRAM is dependent on the ACC-72EX communication option and installed Hilscher ComX module.

Acc72Ex[*i*].Idata32[*j*] is located in the same registers as Acc72Ex[*i*].Data8[4**j*] to Acc72Ex[*i*].Data8[4**j*+5], Acc72Ex[*i*].Idata16[2**j*] to Acc72Ex[*i*].Idata16[2**j*+1], Acc72Ex[*i*].Udata16[2**j*] to Acc72Ex[*i*].Udata16[2**j*+1] and Acc72Ex[*i*].Udata32[*j*]. It is the user's responsibility to prevent possible multiple uses of the same register.

In C, this element should be accessed through the C functions ACC72EX_GetIdata32 and ACC72EX_SetIdata32 described later in this manual.

Acc72EX[*i*].Udata32[*j*]

Description: Dual Ported RAM "unsigned 16-bit integer" data array element

Range: $0 .. 2^{32} - 1$

Units: address dependent

Power-on default: address dependent

Acc72Ex[*i*].Udata32[*j*] is the "*j*th" unsigned 32-bit integer data array element in the Acc72EX[*i*] dual ported RAM. Each of these elements occupies four bytes in the DPRAM, and is located starting at 4**j* addresses past the beginning of the buffer (which is located at the address in Acc72EX[*i*].a). This array is defined based upon the Hilscher ComX memory map.

Index values j in the square brackets can be integer constants in the range 0 to 262,143, or local L-variables. No expressions or non-integer constants are permitted. The size of the DPRAM is dependent on the ACC-72EX communication option and installed Hilscher ComX module.

Acc72Ex[*i*].Udata32[*j*] is located in the same registers as Acc72Ex[*i*].Data8[4**j*] to Acc72Ex[*i*].Data8[4**j*+5], Acc72Ex[*i*].Idata16[2**j*] to Acc72Ex[*i*].Idata16[2**j*+1], Acc72Ex[*i*].Udata16[2**j*] to Acc72Ex[*i*].Udata16[2**j*+1] and Acc72Ex[*i*].Idata32[*j*]. It is the user's responsibility to prevent possible multiple uses of the same register.

In C, this element should be accessed through the C functions ACC72EX_GetUdata32 and ACC72EX_SetUdata32 described later in this manual.

C Programming Access to ACC-72EX Structures

One can use the following header file full of functions to read from and write to the aforementioned Acc72EX[i] structures from a C program. The input argument CardIndex is *i* and ArrayIndex is *j* as above. Use the "Get" functions to retrieve the structure values; use the "Set" functions to write to the structures. In the "Set" functions, the Input argument is the value to which to set the structure.

```
int Acc72EX GetIdata32 (unsigned int CardIndex, unsigned int ArrayIndex);
unsigned int Acc72EX GetUdata32(unsigned int CardIndex, unsigned int ArrayIndex);
short Acc72EX GetIdata16(unsigned int CardIndex, unsigned int ArrayIndex);
char Acc72EX GetData8 (unsigned int CardIndex, unsigned int ArrayIndex);
unsigned short Acc72EX GetUdata16 (unsigned int CardIndex, unsigned int ArrayIndex);
void Acc72EX SetIdata16(unsigned int CardIndex, unsigned int ArrayIndex, short Input);
void Acc72EX SetUdata16 (unsigned int CardIndex, unsigned int ArrayIndex, unsigned short Input);
void Acc72EX_SetIdata32(unsigned int CardIndex, unsigned int ArrayIndex, int Input);
void Acc72EX SetUdata32 (unsigned int CardIndex, unsigned int ArrayIndex, unsigned int Input);
void Acc72EX SetData8(unsigned int CardIndex, unsigned int ArrayIndex, char Input);
short Acc72EX GetIdata16(unsigned int CardIndex, unsigned int ArrayIndex)
{
       unsigned int *myptr = (unsigned int *)piom + (DPRCSBase + CardIndex * 0x100000) / 4;
       return (short)((myptr[ArrayIndex] << 8) >> 16);
}
unsigned short Acc72EX GetUdata16(unsigned int CardIndex, unsigned int ArrayIndex)
{
       unsigned int *myptr = (unsigned int *)piom + (DPRCSBase + CardIndex * 0x100000) / 4;
       return (unsigned short)((myptr[ArrayIndex] << 8) >> 16);
char Acc72EX GetData8 (unsigned int CardIndex, unsigned int ArrayIndex)
{
       unsigned int *myptr = (unsigned int *)piom + (DPRCSBase + CardIndex * 0x100000) / 4;
return (myptr[ArrayIndex / 2] << (16 / (1 + (ArrayIndex % 4) % 2))) >> 24;
unsigned int Acc72EX GetUdata32(unsigned int CardIndex, unsigned int ArrayIndex)
{
       unsigned int i = ArrayIndex * 4, j, k = 0;
       unsigned int out = 0;
       for(j = i; j <= i + 3; j++)</pre>
       {
               out |= (unsigned int)((unsigned int)Acc72EX GetData8(CardIndex, j) << (8 * k));</pre>
               k++;
       }
       return out:
int Acc72EX GetIdata32 (unsigned int CardIndex, unsigned int ArrayIndex)
{
       return (int)Acc72EX GetUdata32(CardIndex, ArrayIndex);
}
void Acc72EX SetIdata16 (unsigned int CardIndex, unsigned int ArrayIndex, short Input)
{
       unsigned int *myptr = (unsigned int *)piom + (DPRCSBase + CardIndex * 0x100000) / 4;
       myptr[ArrayIndex] = (Input << 8) & 0x00FFFF00;</pre>
void Acc72EX SetUdata16 (unsigned int CardIndex, unsigned int ArrayIndex, unsigned short Input)
{
       unsigned int *myptr = (unsigned int *)piom + (DPRCSBase + CardIndex * 0x100000) / 4;
       myptr[ArrayIndex] = (Input << 8) & 0x00FFFF00;</pre>
```

```
void Acc72EX SetIdata32 (unsigned int CardIndex, unsigned int ArrayIndex, int Input)
{
       unsigned int *myptr = (unsigned int *)piom + (DPRCSBase + CardIndex * 0x100000) / 4;
       myptr[ArrayIndex] = ((Input << 16) >> 8);
       myptr[ArrayIndex + 1] = ((Input >> 16) << 8);</pre>
}
void Acc72EX SetUdata32(unsigned int CardIndex, unsigned int ArrayIndex, unsigned int Input) {
       unsigned int *myptr = (unsigned int *)piom + (DPRCSBase + CardIndex * 0x100000) / 4;
       myptr[ArrayIndex] = (Input << 16) >> 8;
       myptr[ArrayIndex + 1] = ((Input >> 16) << 8);</pre>
}
void Acc72EX SetData8(unsigned int CardIndex, unsigned int ArrayIndex, char Input) {
       unsigned int *myptr = (unsigned int *)piom + (DPRCSBase + CardIndex * 0x100000) / 4;
       unsigned int shift = (8 * (1 + ArrayIndex % 2));
       unsigned int ind = ArrayIndex / 2;
       myptr[ind] &= ~(0x000000FF << shift);</pre>
       myptr[ind] |= (Input << shift);</pre>
```

Global Header for Power PMAC Projects

This section provides example for header files which allow use of native netX variable names rather than using Power PMAC structures. The following header file is written as generically as possible allowing access to most used registers in System, Handshake and Communication Channels.

```
/* ACC-72EX Power PMAC Project Header
/* This header file provides macro definitions for most common registers in Hilsche COMX modules
/* used in ACC-72EX.
/*
/* Instructions:
/* Uncomment the related #define depending on ACC-72EX option
/*
//#define __DeviceNet_Slave__
//#define __CANopen_Master__
//#define __CANopen_Slave__
//#define __CC_Link_Slave
//#define __EtherCAT_Master____
//#define __EtherCAT_Slave____
//#define __EtherNetIP_Scanner_Master____
//#define ___EtherNetIP_Adapter_Slave
// System Information Block Structure
       #define SI_abCookie_0_
                                                           Acc72Ex[0].Data8[0]
       #define SI_abCookie_1_
                                                           Acc72Ex[0].Data8[1]
       #define SI abCookie 2
                                                           Acc72Ex[0].Data8[2]
       #define SI abCookie 3
                                                           Acc72Ex[0].Data8[3]
       #define SI ulDpmTotalSize
                                                           Acc72Ex[0].Udata32[1]
       #define SI_ulDeviceNumber
#define SI_ulSerialNumber
                                                           Acc72Ex[0].Udata32[2]
                                                           Acc72Ex[0].Udata32[3]
       #define SI ausHwOptions 0
                                                           Acc72Ex[0].Udata16[8]
       #define SI_ausHwOptions_1_
                                                           Acc72Ex[0].Udata16[9]
       #define SI_ausHwOptions_2_
#define SI_ausHwOptions_3_
                                                           Acc72Ex[0].Udata16[10]
                                                           Acc72Ex[0].Udata16[11]
       #define SI usManufacturer
                                                           Acc72Ex[0].Udata16[12]
       #define SI_usProductionDate
#define SI_ulLicenseFlags1
                                                           Acc72Ex[0].Udata16[13]
                                                           Acc72Ex[0].Udata32[7]
       #define SI ulLicenseFlags2
                                                           Acc72Ex[0].Udata32[8]
       #define SI_usNetxLicenseID
                                                           Acc72Ex[0].Udata16[18]
       #define SI_usNetxLicenseFlags
                                                           Acc72Ex[0].Udata16[19]
       #define SI usDeviceClass
                                                           Acc72Ex[0].Udata16[20]
       #define SI bHwRevision
                                                           Acc72Ex[0].Data8[42]
       #define SI_bHwCompatibility
                                                           Acc72Ex[0].Data8[43]
       #define SI bDevIdNumber
                                                           Acc72Ex[0].Data8[44]
// System Channel Information Structure
       #define SCI bChannelType
                                                           Acc72Ex[0].Data8[48]
       #define SCI bSizePositionOfHandshake
                                                           Acc72Ex[0].Data8[50]
       #define SCI bNumberOfBlocks
                                                           Acc72Ex[0].Data8[51]
                                                           Acc72Ex[0].Udata32[13]
       #define SCI_ulSizeOfChannel
       #define SCI usSizeOfMailbox
                                                           Acc72Ex[0].Udata16[28]
       #define SCI usMailboxStartOffset
                                                           Acc72Ex[0].Udata16[29]
// Handshake Channel Information Structure
       #define HCI bChannelType
                                                           Acc72Ex[0].Data8[64]
       #define HCI ulSizeOfChannel
                                                           Acc72Ex[0].Udata32[17]
// Communication Channel 0 Information Structure
       #define CC0I bChannelType
                                                           Acc72Ex[0].Data8[80]
       #define CC0I_bChannelId
                                                           Acc72Ex[0].Data8[81]
       #define CC0I bSizePositionOfHandshake
                                                           Acc72Ex[0].Data8[82]
```

```
#define CCOI bNumberOfBlocks
                                                                Acc72Ex[0].Data8[83]
        #define CC0I_ulSizeOfChannel
#define CC0I_usCommunicationClass
                                                                Acc72Ex[0].Udata32[21]
                                                                Acc72Ex[0].Udata16[44]
        #define CC01 usProtocolClass
                                                                Acc72Ex[0].Udata16[45]
        #define CC0I usConformanceClass
                                                                Acc72Ex[0].Udata16[46]
// Communication Channel 1 Information Structure
        #define CC1I bChannelType
                                                                Acc72Ex[0].Data8[96]
        #define CC11_bChannelId
                                                                Acc72Ex[0].Data8[97]
        #define CC11 bSizePositionOfHandshake
                                                                Acc72Ex[0].Data8[98]
        #define CC1I bNumberOfBlocks
                                                               Acc72Ex[0].Data8[99]
        #define CC1I_ulSizeOfChannel
                                                               Acc72Ex[0].Udata32[25]
        #define CC1I_usCommunicationClass
#define CC1I_usProtocolClass
                                                                Acc72Ex[0].Udata16[52]
                                                                Acc72Ex[0].Udata16[53]
        #define CC11 usConformanceClass
                                                                Acc72Ex[0].Udata16[54]
// Communication Channel 0 Information Structure
        #define CC2I bChannelType
                                                                Acc72Ex[0].Data8[112]
        #define CC2I bChannelId
                                                                Acc72Ex[0].Data8[113]
        #define CC21_bSizePositionOfHandshake
#define CC21_bNumberOfBlocks
                                                                Acc72Ex[0].Data8[114]
                                                                Acc72Ex[0].Data8[115]
        #define CC2I ulSizeOfChannel
                                                               Acc72Ex[0].Udata32[29]
        #define CC2I_usCommunicationClass
#define CC2I_usProtocolClass
                                                                Acc72Ex[0].Udata16[60]
                                                                Acc72Ex[0].Udata16[61]
        #define CC2I usConformanceClass
                                                               Acc72Ex[0].Udata16[62]
// Communication Channel 1 Information Structure
        #define CC3I bChannelType
                                                               Acc72Ex[0].Data8[128]
        #define CC3I bChannelId
                                                                Acc72Ex[0].Data8[129]
                                                                Acc72Ex[0].Data8[130]
        #define CC3I_bSizePositionOfHandshake
        #define CC3I bNumberOfBlocks
                                                                Acc72Ex[0].Data8[131]
        #define CC3I_ulSizeOfChannel
                                                               Acc72Ex[0].Udata32[33]
        #define CC3I_usCommunicationClass
                                                               Acc72Ex[0].Udata16[68]
        #define CC3I_usProtocolClass
#define CC3I_usConformanceClass
                                                                Acc72Ex[0].Udata16[69]
                                                                Acc72Ex[0].Udata16[70]
// Application Channel 0 Information Structure
        #define AC0I bChannelType
                                                                Acc72Ex[0].Data8[144]
        #define AC01 bChannelId
                                                                Acc72Ex[0].Data8[145]
        #define AC01 bSizePositionOfHandshake
                                                                Acc72Ex[0].Data8[146]
        #define AC01_bNumberOfBlocks
#define AC01_ulSizeOfChannel
                                                                Acc72Ex[0].Data8[147]
                                                                Acc72Ex[0].Udata32[37]
// Application Channel 1 Information Structure
        #define AC1I bChannelType
                                                                Acc72Ex[0].Data8[160]
        #define AC11_bChannelId
                                                                Acc72Ex[0].Data8[161]
        #define AC11_bSizePositionOfHandshake
                                                                Acc72Ex[0].Data8[162]
        #define AC1I_bNumberOfBlocks
#define AC1I_ulSizeOfChannel
                                                                Acc72Ex[0].Data8[163]
                                                                Acc72Ex[0].Udata32[41]
// System Control Block Structure
        #define SCtrl ulSystemCommandCOS
                                                                Acc72Ex[0].Udata32[46]
// System Status Block Structure
        #define SStat_ulSystemCOS
                                                                Acc72Ex[0].Udata32[48]
        #define SStat ulSystemStatus
                                                                Acc72Ex[0].Udata32[49]
        #define SStat ulSystemError
                                                                Acc72Ex[0].Udata32[50]
        #define SStat_ulBootError
                                                                Acc72Ex[0].Udata32[51]
        #define SStat ulTimeSinceStart
                                                                Acc72Ex[0].Udata32[52]
        #define SStat usCpuLoad
                                                                Acc72Ex[0].Udata16[106]
                                                                Acc72Ex[0].Udata16[108]
        #define SStat ulHWFeatures
// NETX SYSTEM SEND MAILBOX
        #define SSMB usPackagesAccepted
                                                                Acc72Ex[0].Udata16[128]
        #define SSMB ulDest
                                                                Acc72Ex[0].Udata32[65]
        #define SSMB_ulSrc
                                                                Acc72Ex[0].Udata32[66]
        #define SSMB ulDestId
                                                                Acc72Ex[0].Udata32[67]
        #define SSMB_ulSrcId
                                                                Acc72Ex[0].Udata32[68]
        #define SSMB ulLen
                                                                Acc72Ex[0].Udata32[69]
        #define SSMB ulId
                                                                Acc72Ex[0].Udata32[70]
```

#define SSMB ulState Acc72Ex[0].Udata32[71] #define SSMB_ulCmd Acc72Ex[0].Udata32[72] #define SSMB_ulExt Acc72Ex[0].Udata32[73] #define SSMB ulRout Acc72Ex[0].Udata32[74] ptr SSMB_Data8(84)->*;
ptr SSMB_Data16(42)->*; ptr SSMB Data32(21)->*; // NETX SYSTEM RECEIVE MAILBOX #define SRMB usWaitingPackages Acc72Ex[0].Udata16[192] #define SRMB ulDest Acc72Ex[0].Udata32[97] #define SRMB_ulSrc Acc72Ex[0].Udata32[98] #define SRMB ulDestId Acc72Ex[0].Udata32[99] #define SRMB ulSrcId Acc72Ex[0].Udata32[100] #define SRMB ulLen Acc72Ex[0].Udata32[101] #define SRMB_ulId Acc72Ex[0].Udata32[102] #define SRMB ulState Acc72Ex[0].Udata32[103] #define SRMB ulCmd Acc72Ex[0].Udata32[104] #define SRMB ulExt Acc72Ex[0].Udata32[105] #define SRMB ulRout Acc72Ex[0].Udata32[106] ptr SRMB Data8(84)->*; ptr SRMB Data16(42)->*; ptr SRMB Data32(21)->*; // SC bNetxFlags #define HCSC NSF READY Acc72Ex[0].Udata16[257].0 #define HCSC_NSF_ERROR Acc72Ex[0].Udata16[257].1 #define HCSC_NSF_HOST_COS_ACK
#define HCSC_NSF_NETX_COS_CMD Acc72Ex[0].Udata16[257].2 Acc72Ex[0].Udata16[257].3 #define HCSC_NSF_SEND_MBX_ACK Acc72Ex[0].Udata16[257].4 #define HCSC NSF RECV MBX CMD Acc72Ex[0].Udata16[257].5 // SC bHostFlags #define HCSC HSF RESET Acc72Ex[0].Udata16[257].8 #define HCSC_HSF_BOOTSTART
#define HCSC_HSF_HOST_COS_CMD Acc72Ex[0].Udata16[257].9 Acc72Ex[0].Udata16[257].10 #define HCSC HSF NETX COS ACK Acc72Ex[0].Udata16[257].11 #define HCSC HSF SEND MBX CMD Acc72Ex[0].Udata16[257].12 #define HCSC HSF RECV MBX ACK Acc72Ex[0].Udata16[257].13 // CC0 usNetxFlags #define HCCC0 usNetxFlags Acc72Ex[0].Udata16[260] Acc72Ex[0].Udata16[260].0 #define HCCC0_NCF_COMMUNICATING #define HCCC0 NCF ERROR Acc72Ex[0].Udata16[260].1 #define HCCC0 NCF HOST COS ACK Acc72Ex[0].Udata16[260].2 #define HCCC0 NCF NETX COS CMD Acc72Ex[0].Udata16[260].3 #define HCCC0_NCF_SEND_MBX_ACK
#define HCCC0_NCF_RECV_MBX_CMD Acc72Ex[0].Udata16[260].4 Acc72Ex[0].Udata16[260].5 #define HCCC0 NCF PD0 OUT ACK Acc72Ex[0].Udata16[260].6 #define HCCC0_NCF_PD0_IN_CMD Acc72Ex[0].Udata16[260].7 #define HCCC0_NCF_PD1_OUT_ACK
#define HCCC0_NCF_PD1_IN_CMD Acc72Ex[0].Udata16[260].8 Acc72Ex[0].Udata16[260].9 // CC0 usHostFlags #define HCCC0 usHostFlags Acc72Ex[0].Udata16[261] #define HCCC0 HCF HOST COS CMD Acc72Ex[0].Udata16[261].2 #define HCCC0 HCF NETX COS ACK Acc72Ex[0].Udata16[261].3 #define HCCC0_HCF_SEND_MBX_CMD Acc72Ex[0].Udata16[261].4 #define HCCC0 HCF RECV MBX ACK Acc72Ex[0].Udata16[261].5 #define HCCC0_HCF_PD0_OUT_CMD Acc72Ex[0].Udata16[261].6 #define HCCC0 HCF PD0 IN ACK Acc72Ex[0].Udata16[261].7 #define HCCC0_HCF_PD1_OUT_CMD #define HCCC0_HCF_PD1_IN_ACK Acc72Ex[0].Udata16[261].8 Acc72Ex[0].Udata16[261].9 // CC1 usNetxFlags #define HCCC1_usNetxFlags Acc72Ex[0].Udata16[262] #define HCCC1_NCF_COMMUNICATING
#define HCCC1_NCF_ERROR Acc72Ex[0].Udata16[262].0 Acc72Ex[0].Udata16[262].1 #define HCCC1 NCF HOST COS ACK Acc72Ex[0].Udata16[262].2 #define HCCC1_NCF_NETX_COS_CMD
#define HCCC1_NCF_SEND_MBX_ACK Acc72Ex[0].Udata16[262].3 Acc72Ex[0].Udata16[262].4 #define HCCC1 NCF RECV MBX CMD Acc72Ex[0].Udata16[262].5

```
#define HCCC1 NCF PD0 OUT ACK
          #define HCCC1_NCF_PD0_IN_CMD
#define HCCC1_NCF_PD1_OUT_ACK
          #define HCCC1 NCF PD1 IN CMD
// CC1 usHostFlags
          #define HCCC1_usHostFlags
#define HCCC1_HCF_HOST_COS_CMD
          #define HCCC1 HCF NETX COS ACK
          #define HCCC1_HCF_SEND_MBX_CMD
#define HCCC1_HCF_RECV_MBX_ACK
          #define HCCC1 HCF PD0 OUT CMD
          #define HCCC1_HCF_PD0_IN_ACK
          #define HCCC1_HCF_PD1_OUT_CMD
#define HCCC1_HCF_PD1_IN_ACK
// CC2 usNetxFlags
          #define HCCC2_usNetxFlags
#define HCCC2_NCF_COMMUNICATING
          #define HCCC2 NCF ERROR
          #define HCCC2 NCF HOST COS ACK
          #define HCCC2_NCF_NETX_COS_CMD
#define HCCC2_NCF_SEND_MBX_ACK
          #define HCCC2 NCF RECV MBX CMD
          #define HCCC2_NCF_PD0_OUT_ACK
#define HCCC2_NCF_PD0_IN_CMD
          #define HCCC2 NCF PD1 OUT ACK
          #define HCCC2_NCF_PD1_IN_CMD
// CC2 usHostFlags
          #define HCCC2 usHostFlags
          #define HCCC2 HCF HOST COS CMD
          #define HCCC2_HCF_NETX_COS_ACK
#define HCCC2_HCF_SEND_MBX_CMD
#define HCCC2_HCF_RECV_MBX_ACK
          #define HCCC2_HCF_PD0_OUT_CMD
          #define HCCC2_HCF_PD0_IN_ACK
#define HCCC2_HCF_PD1_OUT_CMD
          #define HCCC2 HCF PD1 IN ACK
// CC3 usNetxFlags
          #define HCCC3 usNetxFlags
          #define HCCC3 NCF COMMUNICATING
          #define HCCC3 NCF ERROR
          #define HCCC3_NCF_HOST_COS_ACK
#define HCCC3_NCF_NETX_COS_CMD
          #define HCCC3 NCF SEND MBX ACK
          #define HCCC3_NCF_DEND_HDA_ACK
#define HCCC3_NCF_PD0_OUT_ACK
#define HCCC3_NCF_PD0_IN_CMD
          #define HCCC3 NCF PD1 OUT ACK
          #define HCCC3 NCF PD1 IN CMD
// CC3 usHostFlags
          #define HCCC3 usHostFlags
          #define HCCC3_HCF_HOST_COS_CMD
          #define HCCC3 HCF NETX COS ACK
          #define HCCC3 HCF SEND MBX CMD
          #define HCCC3 HCF RECV MBX ACK
          #define HCCC3_HCF_PD0_OUT_CMD
#define HCCC3_HCF_PD0_IN_ACK
#define HCCC3_HCF_PD1_OUT_CMD
          #define HCCC3 HCF PD1 IN ACK
// CCO Control Block
          #define CC0 RCX APP COS APP READY
          #define CC0_RCX_APP_COS_BUS_ON
#define CC0_RCX_APP_COS_BUS_ON_ENABLE
          #define CC0 RCX APP COS INIT
          #define CC0 RCX APP COS INIT ENABLE
          #define CC0_RCX_APP_COS_LOCK_CFG
#define CC0_RCX_APP_COS_LOCK_CFG_ENA
          #define CC0 RCX APP COS DMA
          #define CC0 RCX APP COS DMA ENABLE
```

Acc72Ex[0].Udata16[262].6 Acc72Ex[0].Udata16[262].7 Acc72Ex[0].Udata16[262].8 Acc72Ex[0].Udata16[262].9 Acc72Ex[0].Udata16[263] Acc72Ex[0].Udata16[263].2 Acc72Ex[0].Udata16[263].3 Acc72Ex[0].Udata16[263].4 Acc72Ex[0].Udata16[263].5 Acc72Ex[0].Udata16[263].6 Acc72Ex[0].Udata16[263].7 Acc72Ex[0].Udata16[263].8 Acc72Ex[0].Udata16[263].9 Acc72Ex[0].Udata16[264] Acc72Ex[0].Udata16[264].0 Acc72Ex[0].Udata16[264].1 Acc72Ex[0].Udata16[264].2 Acc72Ex[0].Udata16[264].3 Acc72Ex[0].Udata16[264].4 Acc72Ex[0].Udata16[264].5 Acc72Ex[0].Udata16[264].6 Acc72Ex[0].Udata16[264].7 Acc72Ex[0].Udata16[264].8 Acc72Ex[0].Udata16[264].9 Acc72Ex[0].Udata16[265] Acc72Ex[0].Udata16[265].2 Acc72Ex[0].Udata16[265].3 Acc72Ex[0].Udata16[265].4 Acc72Ex[0].Udata16[265].5 Acc72Ex[0].Udata16[265].6 Acc72Ex[0].Udata16[265].7 Acc72Ex[0].Udata16[265].8 Acc72Ex[0].Udata16[265].9 Acc72Ex[0].Udata16[266] Acc72Ex[0].Udata16[266].0 Acc72Ex[0].Udata16[266].1 Acc72Ex[0].Udata16[266].2 Acc72Ex[0].Udata16[266].3 Acc72Ex[0].Udata16[266].4 Acc72Ex[0].Udata16[266].5 Acc72Ex[0].Udata16[266].6 Acc72Ex[0].Udata16[266].7 Acc72Ex[0].Udata16[266].8 Acc72Ex[0].Udata16[266].9 Acc72Ex[0].Udata16[267] Acc72Ex[0].Udata16[267].2 Acc72Ex[0].Udata16[267].3 Acc72Ex[0].Udata16[267].4 Acc72Ex[0].Udata16[267].5 Acc72Ex[0].Udata16[267].6 Acc72Ex[0].Udata16[267].7 Acc72Ex[0].Udata16[267].8 Acc72Ex[0].Udata16[267].9 Acc72Ex[0].Udata16[388].0 Acc72Ex[0].Udata16[388].1 Acc72Ex[0].Udata16[388].2 Acc72Ex[0].Udata16[388].3 Acc72Ex[0].Udata16[388].4 Acc72Ex[0].Udata16[388].5 Acc72Ex[0].Udata16[388].6 Acc72Ex[0].Udata16[388].7 Acc72Ex[0].Udata16[388].8

#define CC0_ulDeviceWatchdog

Acc72Ex[0].Udata32[195]

```
// CC0 CommunicationCOS
                                                                     Acc72Ex[0].Udata16[392].0
Acc72Ex[0].Udata16[392].1
        #define CC0 RCX COMM COS READY
         #define CC0 RCX COMM COS RUN
         #define CC0 RCX COMM COS BUS ON
                                                                     Acc72Ex[0].Udata16[392].2
                                                                 Acc72Ex[U].Udata16[392].3
Acc72Ex[0].Udata16[392].4
Acc72Ex[0].Udata16[392].4
Acc72Ex[0].Udata16[392].5
        #define CC0 RCX COMM COS CONFIG LOCKED
        #define CC0_RCX_COMM_COS_CONFIG_NEW
#define CC0_RCX_COMM_COS_RESTART_REQ
                                                                Acc72Ex[0].Udata16[392].6
         #define CC0 RCX COMM COS RESTART REQ ENA
        #define CC0_RCX_COMM_COS_DMA
                                                                      Acc72Ex[0].Udata16[392].7
// CC0 Status Block
        #define CC0_ulCommunicationState
                                                                      Acc72Ex[0].Udata32[197]
         #define CC0_ulCommunicationError
                                                                       Acc72Ex[0].Udata32[198]
         #define CC0 usVersion
                                                                      Acc72Ex[0].Udata16[398]
         #define CC0 usWatchdogTime
                                                                       Acc72Ex[0].Udata16[399]
        #define CC0_bPDInHskMode
                                                                       Acc72Ex[0].Data8[800]
        #define CC0 bPDInSource
                                                                       Acc72Ex[0].Data8[801]
        #define CC0 bPDOutHskMode
                                                                       Acc72Ex[0].Data8[802]
        #define CC0 bPDOutSource
                                                                      Acc72Ex[0].Data8[803]
        #define CC0_ulHostWatchdog
#define CC0_ulErrorCount
                                                                       Acc72Ex[0].Udata32[201]
                                                                       Acc72Ex[0].Udata32[202]
        #define CC0 bErrorLogInd
                                                                      Acc72Ex[0].Data8[812]
        #define CC0_bErrorPDInCnt
#define CC0_bErrorPDOutCnt
                                                                      Acc72Ex[0].Data8[813]
                                                                       Acc72Ex[0].Data8[814]
         #define CC0 bErrorSyncCnt
                                                                      Acc72Ex[0].Data8[815]
         #define CC0_bSyncHskMode
                                                                       Acc72Ex[0].Data8[816]
        #define CC0 bSyncSource
                                                                      Acc72Ex[0].Data8[817]
// CC1_Control Block
        #define CC1_RCX_APP_COS_APP_READY
#define CC1_RCX_APP_COS_BUS_ON
                                                                      Acc72Ex[0].Udata16[8196].0
                                                                      Acc72Ex[0].Udata16[8196].1
         #define CC1 RCX APP COS BUS ON ENABLE
                                                                      Acc72Ex[0].Udata16[8196].2
        #define CC1 RCX APP COS INIT
                                                                     Acc72Ex[0].Udata16[8196].3
        #define CC1_RCX_APP_COS_INIT_ENABLE
#define CC1_RCX_APP_COS_LOCK_CFG
                                                                     Acc72Ex[0].Udata16[8196].4
Acc72Ex[0].Udata16[8196].5
         #define CC1 RCX APP COS LOCK CFG ENA
                                                                     Acc72Ex[0].Udata16[8196].6
        #define CC1_RCX_APP_COS_DMA
#define CC1_RCX_APP_COS_DMA_ENABLE
                                                                      Acc72Ex[0].Udata16[8196].7
                                                                      Acc72Ex[0].Udata16[8196].8
        #define CC1 ulDeviceWatchdog
                                                                      Acc72Ex[0].Udata32[4099]
// CC1 CommunicationCOS
         #define CC1 RCX COMM COS READY
                                                                      Acc72Ex[0].Udata16[8200].0
         #define CC1 RCX COMM COS RUN
                                                                      Acc72Ex[0].Udata16[8200].1
                                                                     Acc72Ex[0].Udata16[8200].2
         #define CC1_RCX_COMM_COS_BUS_ON
                                                                Acc72Ex[0].Udata16[8200].2
Acc72Ex[0].Udata16[8200].3
Acc72Ex[0].Udata16[8200].4
Acc72Ex[0].Udata16[8200].5
Acc72Ex[0].Udata16[8200].6
        #define CC1_RCX_COMM_COS_CONFIG_LOCKED
#define CC1_RCX_COMM_COS_CONFIG_NEW
         #define CC1 RCX COMM COS RESTART REQ
        #define CC1_RCX_COMM_COS_RESTART_REQ_ENA
#define CC1_RCX_COMM_COS_DMA
                                                                      Acc72Ex[0].Udata16[8200].7
// CC1 Status Block
        #define CC1 ulCommunicationState
                                                                      Acc72Ex[0].Udata32[4101]
         #define CC1_ulCommunicationError
                                                                      Acc72Ex[0].Udata32[4102]
         #define CC1 usVersion
                                                                      Acc72Ex[0].Udata16[8206]
        #define CC1_usWatchdogTime
#define CC1 bPDInHskMode
                                                                       Acc72Ex[0].Udata16[8207]
                                                                       Acc72Ex[0].Data8[16416]
        #define CC1 bPDInSource
                                                                       Acc72Ex[0].Data8[16417]
        #define CC1_bPDOutHskMode
                                                                       Acc72Ex[0].Data8[16418]
         #define CC1 bPDOutSource
                                                                       Acc72Ex[0].Data8[16419]
        #define CC1_ulHostWatchdog
                                                                      Acc72Ex[0].Udata32[4105]
        #define CC1_ulErrorCount
                                                                      Acc72Ex[0].Udata32[4106]
        #define CC1_bErrorLogInd
#define CC1_bErrorPDInCnt
                                                                       Acc72Ex[0].Data8[16428]
                                                                       Acc72Ex[0].Data8[16429]
         #define CC1 bErrorPDOutCnt
                                                                      Acc72Ex[0].Data8[16430]
         #define CC1 bErrorSyncCnt
                                                                      Acc72Ex[0].Data8[16431]
        #define CC1_bSyncHskMode
#define CC1_bSyncSource
                                                                       Acc72Ex[0].Data8[16432]
                                                                      Acc72Ex[0].Data8[16433]
// CC2 Control Block
         #define CC2 RCX APP COS APP READY
                                                                      Acc72Ex[0].Udata16[16004].0
         #define CC2 RCX APP COS BUS ON
                                                                      Acc72Ex[0].Udata16[16004].1
```

```
#define CC2 RCX APP COS BUS ON ENABLE
         #define CC2_RCX_APP_COS_INIT
#define CC2_RCX_APP_COS_INIT_ENABLE
         #define CC2 RCX APP COS LOCK CFG
         #define CC2 RCX APP COS LOCK CFG ENA
         #define CC2_RCX_APP_COS_DMA
#define CC2_RCX_APP_COS_DMA_ENABLE
        #define CC2_ulDeviceWatchdog
// CC2 CommunicationCOS
         #define CC2 RCX COMM COS READY
         #define CC2_RCX_COMM COS RUN
         #define CC2 RCX COMM COS BUS ON
         #define CC2 RCX COMM COS CONFIG LOCKED
         #define CC2 RCX COMM COS CONFIG NEW
         #define CC2_RCX_COMM_COS_RESTART_REQ
         #define CC2 RCX COMM COS RESTART REQ ENA
         #define CC2 RCX COMM COS DMA
// CC2 Status Block
         #define CC2 ulCommunicationState
         #define CC2 ulCommunicationError
        #define CC2_usVersion
#define CC2_usWatchdogTime
#define CC2_bPDInHskMode
         #define CC2_bPDInSource
         #define CC2 bPDOutHskMode
         #define CC2 bPDOutSource
         #define CC2_ulHostWatchdog
         #define CC2_ulErrorCount
         #define CC2_bErrorLogInd
#define CC2_bErrorPDInCnt
         #define CC2 bErrorPDOutCnt
         #define CC2_bErrorSyncCnt
#define CC2_bSyncHskMode
         #define CC2 bSyncSource
// CC3 Control Block
         #define CC3 RCX APP COS APP READY
         #define CC3 RCX APP COS BUS ON
         #define CC3_RCX_APP_COS_BUS_ON_ENABLE
#define CC3_RCX_APP_COS_INIT
         #define CC3 RCX APP COS INIT ENABLE
         #define CC3_RCX_APP_COS_LOCK_CFG
         #define CC3_RCX_APP_COS_LOCK_CFG_ENA
#define CC3_RCX_APP_COS_DMA
         #define CC3 RCX APP COS DMA ENABLE
         #define CC3 ulDeviceWatchdog
// CC3 CommunicationCOS
         #define CC3_RCX_COMM_COS_READY
         #define CC3 RCX COMM COS RUN
         #define CC3 RCX COMM COS BUS ON
         #define CC3 RCX COMM COS CONFIG LOCKED
         #define CC3_RCX_COMM_COS_CONFIG_NEW
#define CC3_RCX_COMM_COS_RESTART_REQ
         #define CC3 RCX COMM COS RESTART REQ ENA
         #define CC3 RCX COMM COS DMA
// CC3 Status Block
         #define CC3 ulCommunicationState
         #define CC3_ulCommunicationError
#define CC3_usVersion
         #define CC3 usWatchdogTime
         #define CC3 bPDInHskMode
         #define CC3_bPDInSource
#define CC3_bPDOutHskMode
         #define CC3 bPDOutSource
         #define CC3_ulHostWatchdog
#define CC3_ulErrorCount
         #define CC3 bErrorLogInd
```

Acc72Ex[0].Udata16[16004].2 Acc72Ex[0].Udata16[16004].3 Acc72Ex[0].Udata16[16004].4 Acc72Ex[0].Udata16[16004].5 Acc72Ex[0].Udata16[16004].6 Acc72Ex[0].Udata16[16004].7 Acc72Ex[0].Udata16[16004].8 Acc72Ex[0].Udata32[8003] Acc72Ex[0].Udata16[16008].0 Acc72Ex[0].Udata16[16008].1 Acc72Ex[0].Udata16[16008].2 Acc72Ex[0].Udata16[16008].3 Acc72Ex[0].Udata16[16008].4 Acc72Ex[0].Udata16[16008].4 Acc72Ex[0].Udata16[16008].5 Acc72Ex[0].Udata16[16008].5 Acc72Ex[0].Udata16[16008].7 Acc72Ex[0].Udata32[8005] Acc72Ex[0].Udata32[8006] Acc72Ex[0].Udata16[16014] Acc72Ex[0].Udata16[16015] Acc72Ex[0].Data8[32032] Acc72Ex[0].Data8[32033] Acc72Ex[0].Data8[32034] Acc72Ex[0].Data8[32035] Acc72Ex[0].Udata32[8009] Acc72Ex[0].Udata32[8010] Acc72Ex[0].Data8[32044] Acc72Ex[0].Data8[32045] Acc72Ex[0].Data8[32046] Acc72Ex[0].Data8[32047] Acc72Ex[0].Data8[32048] Acc72Ex[0].Data8[32049] Acc72Ex[0].Udata16[23812].0 Acc72Ex[0].Udata16[23812].1 Acc72Ex[0].Udata16[23812].2 Acc72Ex[0].Udata16[23812].3 Acc72Ex[0].Udata16[23812].4 Acc72Ex[0].Udata16[23812].5 Acc72Ex[0].Udata16[23812].6 Acc72Ex[0].Udata16[23812].7 Acc72Ex[0].Udata16[23812].8 Acc72Ex[0].Udata32[11907] Acc72Ex[0].Udata16[23816].0 Acc72Ex[0].Udata16[23816].1 Acc72Ex[0].Udata16[23816].2 Acc72Ex[0].Udata16[23816].3 Acc72Ex[0].Udata16[23816].4 Acc72Ex[0].Udata16[23816].5 Acc72Ex[0].Udata16[23816].6 Acc72Ex[0].Udata16[23816].7 Acc72Ex[0].Udata32[11909] Acc72Ex[0].Udata32[11910] Acc72Ex[0].Udata16[23822] Acc72Ex[0].Udata16[23823] Acc72Ex[0].Data8[47648] Acc72Ex[0].Data8[47649] Acc72Ex[0].Data8[47650] Acc72Ex[0].Data8[47651] Acc72Ex[0].Udata32[11913] Acc72Ex[0].Udata32[11914]

Acc72Ex[0].Data8[47660]

<pre>#define CC3_bErrorPDInCnt #define CC3_bErrorPDOutCnt #define CC3_bErrorSyncCnt #define CC3_bSyncHskMode #define CC3_bSyncSource</pre>		A A A A A
<pre>#ifdefPROFIBUS_DP_Master_ #define CC0_PD0_OUT_OFFSET_2BYTE #define CC0_PD0_OUT_SIZE_2BYTE #define CC0_PD0_IN_OFFSET_2BYTE #define CC0_PD1_OUT_OFFSET_2BYTE #define CC0_PD1_OUT_OFFSET_2BYTE #define CC0_PD1_OUT_SIZE_2BYTE #define CC0_PD1_IN_OFFSET_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #endif</pre>	\$980 2880 \$14C0 2880 \$8C0 32 \$8E0 32	
<pre>#ifdefPROFIBUS_DP_Slave #define CC0_PD0_OUT_OFFSET_2BYTE #define CC0_PD0_OUT_SIZE_2BYTE #define CC0_PD0_IN_OFFSET_2BYTE #define CC0_PD0_IN_SIZE_2BYTE #define CC0_PD1_OUT_OFFSET_2BYTE #define CC0_PD1_OUT_SIZE_2BYTE #define CC0_PD1_IN_OFFSET_2BYTE #define CC0_PD1_IN_OFFSET_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #endif</pre>	768 \$C80 768 \$8C0	
<pre>#ifdefDeviceNet_Master #define CC0_PD0_OUT_OFFSET_2BYTE #define CC0_PD0_OUT_SIZE_2BYTE #define CC0_PD0_IN_OFFSET_2BYTE #define CC0_PD1_OUT_OFFSET_2BYTE #define CC0_PD1_OUT_OFFSET_2BYTE #define CC0_PD1_OUT_SIZE_2BYTE #define CC0_PD1_IN_OFFSET_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #endif</pre>	\$980 2880 \$14C0 2880 \$8C0 32 \$8E0 32	
<pre>#ifdefDeviceNet_Slave_ #define CC0_PD0_OUT_OFFSET_2BYTE #define CC0_PD0_OUT_SIZE_2BYTE #define CC0_PD0_IN_OFFSET_2BYTE #define CC0_PD1_OUT_OFFSET_2BYTE #define CC0_PD1_OUT_OFFSET_2BYTE #define CC0_PD1_OUT_SIZE_2BYTE #define CC0_PD1_IN_OFFSET_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #endif</pre>	\$980 768 \$C80 768 \$8C0 32 \$8E0 32	
<pre>#ifdefCANopen_Master #define CC0_PD0_OUT_OFFSET_2BYTE #define CC0_PD0_OUT_SIZE_2BYTE #define CC0_PD0_IN_OFFSET_2BYTE #define CC0_PD0_IN_SIZE_2BYTE #define CC0_PD1_OUT_OFFSET_2BYTE #define CC0_PD1_OUT_SIZE_2BYTE #define CC0_PD1_IN_OFFSET_2BYTE #define CC0_PD1_IN_OFFSET_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #define CC0_PD1_IN_SIZE_2BYTE #endif</pre>	\$980 2880 \$14C0 2880 \$8C0 32 \$8E0 32	
<pre>#ifdefCANopen_Slave #define CC0_PD0_OUT_OFFSET_2BYTE #define CC0_PD0_OUT_SIZE_2BYTE #define CC0_PD0_IN_OFFSET_2BYTE #define CC0_PD0_IN_SIZE_2BYTE #define CC0_PD1_OUT_OFFSET_2BYTE #define CC0_PD1_OUT_SIZE_2BYTE #define CC0_PD1_IN_OFFSET_2BYTE #define CC0_PD1_IN_SIZE_2BYTE</pre>	\$980 768 \$C80 768 \$8C0 32 \$8E0 32	

#endif

#ifdef	CC Link Slave	
#define	CC0 PD0 OUT OFFSET 2BYTE	\$980
#define	CC0_PD0_OUT_SIZE_2BYTE	768
#define	CC0 PD0 IN OFFSET 2BYTE	\$C80
	CC0_PD0_IN_SIZE_2BYTE	768
#define	CC0 PD1 OUT OFFSET 2BYTE	\$8C0
#define	CC0_PD1_OUT_SIZE_2BYTE	32
#define	CC0 PD1 IN OFFSET 2BYTE	\$8E0
	CC0 PD1 IN SIZE 2BYTE	32
#endif	eeo_rbi_in_bibb_zbiib	52
#enarr		
#ifdef	EtherCAT Master	
	CC0 PD0 OUT OFFSET 2BYTE	\$980
		2880
		\$14C0
#define	CC0 PD0 IN SIZE 2BYTE	2880
		\$8C0
	CC0 PD1 OUT SIZE 2BYTE	32
#deline	CCU_PDI_OUI_SIZE_ZBIIE	
#deline		\$8E0
	CC0_PD1_IN_SIZE_2BYTE	32
#endif		
щаельс	There and the second	
	EtherCAT_Slave	0000
	CC0_PD0_OUT_OFFSET_2BYTE	\$980
#define		2880
#define	CC0_PD0_IN_OFFSET_2BYTE	\$14C0
#define	CC0_PD0_IN_SIZE_2BYTE	2880
#define		\$8C0
#define	CC0_PD1_OUT_SIZE_2BYTE	32
#define	CC0_PD1_IN_OFFSET_2BYTE	\$8E0
#define	CC0_PD1_IN_SIZE_2BYTE	32
#endif		
#ifdef	EtherNetIP Scanner Master	
#define	CC0 PD0 OUT OFFSET 2BYTE	\$980
#define #define	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE	\$980 2880
#define #define #define	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE	
<pre>#define #define #define #define</pre>	CC0 PD0 OUT OFFSET 2BYTE CC0 PD0 OUT SIZE 2BYTE CC0 PD0 IN OFFSET 2BYTE CC0 PD0 IN SIZE 2BYTE	2880
<pre>#define #define #define #define</pre>	CC0 PD0 OUT OFFSET 2BYTE CC0 PD0 OUT SIZE 2BYTE CC0 PD0 IN OFFSET 2BYTE CC0 PD0 IN SIZE 2BYTE	2880 \$14C0
<pre>#define #define #define #define #define #define #define</pre>	CC0 PD0 OUT OFFSET 2BYTE CC0 PD0 OUT SIZE 2BYTE CC0 PD0 IN OFFSET 2BYTE CC0 PD0 IN SIZE 2BYTE CC0 PD1 OUT OFFSET 2BYTE CC0 PD1 OUT SIZE 2BYTE	2880 \$14C0 2880
<pre>#define #define #define #define #define #define #define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0
<pre>#define #define #define #define #define #define #define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32
<pre>#define #define #define #define #define #define #define #define #define</pre>	CC0 PD0 OUT OFFSET 2BYTE CC0 PD0 OUT SIZE 2BYTE CC0 PD0 IN OFFSET 2BYTE CC0 PD0 IN SIZE 2BYTE CC0 PD1 OUT OFFSET 2BYTE CC0 PD1 OUT SIZE 2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0
<pre>#define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0
<pre>#define #define #ifdefine</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE EtherNetIP Adapter Slave	2880 \$14C0 2880 \$8C0 32 \$8E0 32
<pre>#define #define #ifdef #ifdef</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE EtherNetIP_Adapter_Slave CC0_PD0_OUT_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980
<pre>#define #define #define #define #define #define #define #define #define #define #ifdef #ifdef #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE 	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880
<pre>#define #define #define #define #define #define #define #define #define #ifdef #ifdef #define #define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE EtherNetIP_Adapter_Slave_ CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0
<pre>#define #define #define #define #define #define #define #define #define #define #ifdef #ifdef #define #define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880
<pre>#define #define #define #define #define #define #define #define #endif #ifdef #define #define #define #define #define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$8C0
<pre>#define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880
<pre>#define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$8C0
<pre>#define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$8C0 32
<pre>#define #define #define</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$8E0 \$880 2880 \$14C0 2880 2880 32 \$8E0
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$8E0 \$880 2880 \$14C0 2880 2880 32 \$8E0
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$8E0 \$880 2880 \$14C0 2880 2880 32 \$8E0
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE C00_PD1_IN_SIZE_2BYTE C00_PD1_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$8E0 \$880 2880 \$14C0 2880 2880 32 \$8E0
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE Open_Modbus_TCP CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$14C0 2880 \$8E0 32 \$8E0 32
<pre>#define #define #ifdef #ifdef #ifdef #ifdef #define #defi</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$14C0 2880 \$4C0 32 \$8E0 32 \$8E0 32
<pre>#define #define #ifdef #ifdef #ifdef #ifdef #define #defi</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE Open_Modbus_TCP CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$8E0 32
<pre>#define #define #ifdef #ifdef #ifdef #ifdef #ifdef #define #defin</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_OUT_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$880 \$14C0 2880 \$8C0 32 \$8E0 32 \$8E0 32 \$8E0 32 \$8E0 32 \$8E0 32
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$280 32 \$980 2880 \$2880 \$14C0 2880
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$12C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$12C0 \$12C0 \$12C0 \$12C0 \$14C0 2880 \$12C0 \$12
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$420 2880 \$14C0 2880 \$820 32
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD0_IN_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$12C0 2880 \$12C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$12C0 2880 \$12C0 2880 \$14C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$12C0 2880 \$14C0 2880 \$14C0 2880 \$12C0 2880 \$12C0 2880 \$12C0 2880 \$14C0 2880 \$12C0 \$12
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$8E0 2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 \$14C0 2880 32
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$14C0 2880 \$440 32 \$8E0 32
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$14C0 2880 \$8C0 32 \$980 2880 \$14C0 32 \$14C0 2880 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0 \$14C0\$14C0 \$14
<pre>#define #define #</pre>	CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_OUT_OFFSET_2BYTE CC0_PD0_OUT_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD0_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_OUT_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_SIZE_2BYTE CC0_PD1_IN_OFFSET_2BYTE CC0_PD1_IN_SIZE_2BYTE	2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$8C0 32 \$8E0 32 \$980 2880 \$14C0 2880 \$14C0 2880 \$440 32 \$8E0 32

<pre>#define CC0_PD0_IN_OFFSET_2B' #define CC0_PD0_IN_SIZE_2BYTI #define CC0_PD1_OUT_OFFSET_21 #define CC0_PD1_OUT_SIZE_2BYTI #define CC0_PD1_IN_OFFSET_2B' #define CC0_PD1_IN_SIZE_2BYTI #endif</pre>	E 2880 BYTE \$8C0 TE 32 YTE \$8E0
<pre>#ifdefPROFINET_IO_Device_3 #define_CCO_PDO_OUT_OFFSET_21 #define_CCO_PDO_OUT_SIZE_2BYT #define_CCO_PDO_IN_OFFSET_2BT #define_CCO_PDO_IN_SIZE_2BYTI #define_CCO_PD1_OUT_OFFSET_2BT #define_CCO_PD1_OUT_SIZE_2BYTI #define_CCO_PD1_IN_OFFSET_2BT #define_CCO_PD1_IN_SIZE_2BYTI #endif</pre>	BYTE \$980 FE 2880 YTE \$14C0 E 2880 BYTE \$8C0 FE 32 YTE \$8E0
ptr CC0_PD0_OUT16(CC0_PD0_OUT ptr CC0_PD0_IN16(CC0_PD0_IN_3 ptr CC0_PD1_OUT16(CC0_PD1_OUT ptr CC0_PD1_IN16(CC0_PD1_IN_3	SIZE_2BYTE)->*; T_SIZE_2BYTE)->*;

Initialization PLC

Recall that ACC-72EX requires a reset after each power up, power cycle, \$\$\$ (reset), or \$\$\$*** (factory default reset). This can be achieved with a startup (or initialization) PLC. Example:

```
// ACC-72EX initialization PLC
open plc Acc72EX StartupPLC
local endtime;
disable plc 2..31
                                                 // Disable all other tasks
// Defining pointers for system channel mailboxes
L_{0}=0
while(L0<84)</pre>
{
      CMD"SSMB Data8(%d) ->Acc72Ex[0].Data8[%d]", L0, L0+300
      sendallcmds
      L0++
}
L0=0
while(L0<42)</pre>
{
      CMD"SSMB Data16(%d) ->Acc72Ex[0].uData16[%d]", L0, L0+150
      sendallcmds
      L0++
}
L_{0}=0
while (LO<21)
{
     CMD"SSMB Data32(%d)->Acc72Ex[0].uData32[%d]",L0,L0+75
      sendallcmds
      L0++
L_{0}=0
while(LO<84)
{
      CMD"SRMB Data8(%d)->Acc72Ex[0].Data8[%d]",L0,L0+428
      sendallcmds
      L0++
}
L0=0
while(L0<42)</pre>
{
      CMD"SRMB Data16(%d)->Acc72Ex[0].uData16[%d]",L0,L0+214
      sendallcmds
```

```
L0++
}
L_{0}=0
while (LO<21)
{
      CMD"SRMB Data32(%d)->Acc72Ex[0].uData32[%d]",L0,L0+107
      sendallcmds
      T_{1}0++
}
// Defining pointers to Out/In PDOs
L0=0
while (LO<CCO_PD0_OUT_SIZE_2BYTE)
{
      CMD"CC0 PD0 OUT16(%d)->Acc72Ex[0].uData16[%d]",L0,L0+CC0 PD0 OUT OFFSET
_2BYTE;
      sendallcmds
      L0++
}
L0=0
while (LO< CCO PDO IN SIZE 2BYTE)
{
      CMD"CC0 PD0 IN16(%d)->Acc72Ex[0].uData16[%d]",L0,L0+CC0 PD0 IN OFFSET 2
BYTE
      sendallcmds
      L0++
}
L0=0
while (LO<CCO PD1 OUT SIZE 2BYTE)
{
      CMD"CC0 PD1 OUT16(%d)->Acc72Ex[0].uData16[%d]",L0,L0+CC0 PD1 OUT OFFSET
2byte
      sendallcmds
      L0++
}
L0=0
while (LO< CC0 PD1 IN SIZE 2BYTE)
{
      CMD"CC0 PD1 IN16(%d) ->Acc72Ex[0].uData16[%d]", L0, L0+
CC0 PD1 IN OFFSET 2BYTE
      sendallcmds
      L0++
SCtrl_ulSystemCommandCOS=$55AA55AA
                                           // Reset token for MASTER Unit
HCSC HSF RESET=1
                                    // Reset bit, token required for reset to
complete
CommErrorFlag=0;
                                           // Reset Time-out Timer
endtime = Sys.Time + 2;
while (CommErrorFlag==0 && HCSC NSF READY==0) // Wait for reset to complete
{
      if (endtime<Sys.Time)</pre>
                                                 // Check for reset timeout
      {
            CommErrorFlag = 1;
      }
call Timer(0.100); // 100 msec
```

```
// Toggle Communication Channel 0's Change of State Acknowledge bit in
// order to read the CCO RCX COMM COS RUN which is a part of Communication
// Channel 0 State Register
HCCC0 HCF NETX COS ACK = HCCC0 HCF_NETX_COS_ACK ^ 1
if (CommErrorFlag==0)
{
       while ((CC0 RCX COMM COS RUN) == 0) // Wait for comm tasks to
       {
              // start on COMX modules
              // Repeating the toggle action for next while loop read
              HCCCO HCF NETX COS ACK = HCCCO HCF NETX COS ACK ^ 1
              call Timer(0.010); // 10 msec
      enable plc Acc72EX_WatchdogPLC // Enable the Watchdog plc
enable plc Acc72EX_PDO_WritePLC // Enable the write plc
enable plc Acc72EX_PDO_ReadPLC // Enable the write plc
       }
disable plc Acc72EX StartupPLC
close
```

The above PLC uses a Timer subprogram call that must be added to the PMAC Script Language \rightarrow Libraries folder of the IDE project:

```
open subprog Timer(wait_duration) // wait_duration in seconds
local EndTime = Sys.Time + wait_duration;
while(Sys.Time < EndTime){}
close</pre>
```

Startup

To enable this startup PLC at power-up or reset, add the following line to pp_startup.txt in the Configuration folder:

enable plc Acc72EX_StartupPLC

Watchdog Function

The host Watchdog and the device Watchdog cells in the control block of each of the communication channels allow the operating system running on the netX to supervise the host or UMAC application and vice versa. There is no Watchdog function for the system block or for the handshake channel. The Watchdog for the channels is located in the control block of the status block of each communication channel.

The netX firmware reads the content of the device Watchdog cell, increments the value by one and copies it back into the host Watchdog location. Then, the application has to copy the new value from the host Watchdog location into the device Watchdog location. Copying the host Watchdog cell to the device Watchdog cell has to happen in the configured Watchdog time. When the overflow occurs, the firmware starts over and a one appears in the host Watchdog cell. A zero turns off the Watchdog and therefore never appears in the host Watchdog cell in the regular process.

The minimum Watchdog time is 20 ms. The application can start the Watchdog function by copying any value unequal to zero into device Watchdog cell. A zero in the device Watchdog location stops the Watchdog function. The Watchdog timeout is configurable in SYCON.net and downloaded to the netX firmware.

If the application fails to copy the value from the host Watchdog location to the device Watchdog location within the configured Watchdog time, the protocol stack will interrupt all network connections

immediately, regardless of their current state. If the Watchdog tripped, then power cycling, channel reset, or channel initialization will allow the communication channel to open network connections again.

Here is sample code for copying the host Watchdog location to the device Watchdog location:

Enabling the Communication Bus

Using the Bus On flag (CCx_RCX_APP_COS_BUS_ON, where x is the communication channel number), the host or UMAC application allows or disallows the netX firmware to open network connections. This flag is used together with the Bus On Enable flag

(CCx_RCX_APP_COS_BUS_ON_ENABLE, where x is the communication channel number). If set, the netX firmware tries to open network connections; if cleared, no connections are allowed, and open connections are closed. If the Bus On Enable flag is set, it enables the execution of the Bus On command in the netX firmware.

```
CC0_RCX_APP_COS_BUS_ON=1// Setting the Bus On flag for 1st ACC-72EXCC0_RCX_APP_COS_BUS_ON_ENABLE=1// Enabling the execution of Bus On Flag for 1st ACC-72EXS_CC0_RCX_APP_COS_BUS_ON=1// Setting the Bus On flag for 2nd ACC-72EXS_CC0_RCX_APP_COS_BUS_ON_ENABLE=1// Enabling the execution of Bus On Flag for 2nd ACC-72EX
```

Locating the Input/Output Data Image in PMAC

The header file provided for use with ACC-72EX provides proper addressing and offsets for each of the PDOs available for each communication module. There are also pointers declared in the header file and are defined as a part of the initialization PLC shown above. These pointers will be used to access different PDOs defined by SYCON.net software.

The following example PLCs are for reference only in order to demonstrate the proper handshaking necessary for reading and writing data to ACC-72EX from Power PMAC.

```
// ACC-72EX Writing to PDO Sample PLC
open plc Acc72EX PDO WritePLC
if (HCCC0 HCF PD0 OUT CMD == HCCC0 NCF PD0 OUT ACK)
                                        {
// Making sure the ACK flag matches the CMD
// flag before writing the value to the
// output data image register
    P200=P200+1
    CC0 PD0 OUT16(0) = P200; // write the output data image register
    // Toggle the CMD flag (^: XOR)
    HCCCO HCF PDO OUT CMD = HCCCO HCF PDO OUT CMD^1
    // indicating write completion
}
close
// ACC-72EX Reading from PDO Sample PLC
```

```
open plc Acc72EX_PDO_ReadPLC
if(HCCC0_NCF_PDO_IN_CMD == HCCC0_HCF_PD0_IN_ACK)
// If CMD flag and ACK flags are
// equal, then the input data image
// register can be read
{
        P201=CC0_PD0_IN16(0); // read the input data image register
        HCCC0_HCF_PD0_IN_ACK = HCCC0_HCF_PD0_IN_ACK ^ 1
        // toggle the acknowledge bit
        // indicating read completion
}
close
```

DIAGNOSTICS

LEDs

There is one system LED (SYS LED) per ACC-72EX. SYS LED is always present as described below. There are up to 4 LEDs per communication and application channel. These LEDs, like the communication channel LED (COM LED), are network-specific and are described separately.

SYS LED

The system status LED (SYS LED) is always available. It indicates the state of the system and its protocol stacks. The following blink patterns are defined:

Color	State	Meaning
Yellow	Flashing Cyclically at 1 Hz	netX is in Boot Loader Mode and is Waiting for Firmware
		Download
	Solid	netX is in Boot Loader Mode, but an Error Occurred
Green	Solid	netX Operating System is Running and a Firmware is Started
Vellow /	Flashing Alternating	2nd Stage Bootloader is active
Green		
Off	N/A	netX has no Power Supply or Hardware Defect Detected

PROFIBUS-DP – Master – OPT10

The following table describes the meaning of the LEDs for the comX PROFIBUS-DP Master communication modules (COMX 100CA-DP/ COMX100CN-DP) when the firmware of the PROFIBUS DP Master protocol is loaded to the comX communication module:

COM LED (COM0)

Color	State	Meaning
Green	Flashing acyclic	No configuration or stack error
Green	Flashing cyclic	Profibus is configured, but bus communication is not yet released from the application
Green	On	Communication to all Slaves is established
Red	Flashing cyclic	Communication to at least one Slave is disconnected
Red	On	Communication to one/all Slaves is disconnected

PROFIBUS-DP – Slave – OPT11

The subsequent table describes the meaning of the LEDs for the comX PROFIBUS-DP Slave communication modules (COMX CA-DP/ COMX CNDP) when the firmware of the PROFIBUS DP Slave protocol is loaded to the comX communication module.

COM LED (COM0)

Color	State	Meaning
Green	On	RUN, cyclic communication
Red	Flashing cyclic	STOP, no communication, connection error
Red	Flashing acyclic	not configured

DeviceNet – Master – OPT20

The following table describes the meaning of the LEDs for the comX communication modules when the firmware of the DeviceNet Master protocol is loaded to the comX communication module:

MNS LED (COM0)

Color	State	Meaning
Green	On	Device is online and has established one or more
		connections
Green	Flashing	Device is online and has established no connection
Green/Red	Green/Red/Off	Self-test after power on:
		Green on for 0,25 s, then red on for 0,25 s, then off
Red	Flashing	Connection timeout
Red	On	Critical connection failure; device has detected a network error: duplicate MAC-
		ID or severe error in CAN network (CAN-bus off)
Red	Off	After start of the device and during duplicate MAC-ID check

DeviceNet – Slave – OPT21

The following table describes the meaning of the LEDs for the comX communication modules when the firmware of the DeviceNet Slave protocol is loaded to the comX communication module:

MNS LED (COM0)

Color	State	Meaning
Green	On	Device is online and has established one or more
		connections
Green	Flashing	Device is online and has established no connection
Green/Red	Green/Red/Off	Self-test after power on:
		Green on for 0,25 s, then red on for 0,25 s, then off
Red	Flashing	Connection timeout
Red	On	Critical connection failure; device has detected a network error: duplicate MAC-
		ID or severe error in CAN network (CAN-bus off)
Red	Off	After start of the device and during duplicate MAC-ID check

CANopen – Master – OPT30

The following table describes the meaning of the LEDs for the comX CANopen Master communication modules (COMX-CA-CO/ COMX-CNCOM) when the firmware of the CANopen Master protocol is loaded to the comX communication module:

CAN LED (COM0)

Color	State	Meaning
Green	Off	The device is executing a reset
Green	Single Flash	STOPPED: The Device is in STOPPED state
Green	Blinking	PREOPERATIONAL: The Device is in the PREOPERATIONAL state
		The indicator turns on and off with a frequency of 2,5 Hz: on for 200 ms,
		followed by off for 200 ms.
Green	On	OPERATIONAL: The Device is in the OPERATIONAL state
Red	Single flash	Warning Limit reached: At least one of the error counters of the CAN controlle
		has reached or exceeded the warning level (too many error frames).
		The indicator shows one short flash (200 ms) followed by a long off phase
		(1,000 ms).
Red	Double flash	Error Control Event: A guard event (NMT Slave or NMTmaster) or a heartbeat
		event (Heartbeat consumer) has occurred.
		The indicator shows a sequence of two short flashes (each 200 ms), separated
		by a short off phase (200 ms). The sequence is finished by a long off phase
		(1,000 ms).
Red	On	Bus Off: The CAN controller is bus off

CANopen – Slave – OPT31

The following table describes the meaning of the LEDs for the comX CANopen Slave communication modules (COMX-CA-CO/ COMX-CNCOS) when the firmware of the CANopen Slave protocol is loaded to the comX communication module:

CAN LED (COM0)

Color	State	Meaning
Green	Off	The device is executing a reset
Green	Single Flash	STOPPED: The Device is in STOPPED state
Green	Blinking	PREOPERATIONAL: The Device is in the PREOPERATIONAL state
		The indicator turns on and off with a frequency of 2,5 Hz: on for 200 ms, followed by off for 200 ms.
Green	On	OPERATIONAL: The Device is in the OPERATIONAL state
Red	Off	No Error: The Device is in working condition
Red	Single flash	Warning Limit reached: At least one of the error counters of the CAN controller has reached or exceeded the warning level (too many error frames). The indicator shows one short flash (200 ms) followed by a long off phase (1,000 ms).
Red	Double flash	Error Control Event: A guard event (NMT Slave or NMTmaster) or a heartbeat event (Heartbeat consumer) has occurred. The indicator shows a sequence of two short flashes (each 200 ms), separated by a short off phase (200 ms). The sequence is finished by a long off phase (1,000 ms).
Red	On	Bus Off: The CAN controller is bus off

CC-Link – Slave – OPT51

The following table describes the meaning of the LEDs for the comX CCLink Slave communication modules (COMX 100CA-CCS/ COMX 100CNCCS) when the firmware of the CC-Link Slave protocol is loaded to the comX communication module:

RUN/ERR LED (COM0)

Color	State	Meaning
Green	Off	1. Before participating in the network
		2. Unable to detect carrier
		3. Timeout
		4. Resetting hardware
Green	On	Receive both refresh and polling signals or just the refresh signal normally, after
		participating in the network.
Red	Off	1. Normal communication
		2. Resetting hardware
Red	Blinking	The switch setting has been changed from the setting at the reset cancellation
		(blinks for 0.4 sec.).
Red	On	1. CRC error
		2. Address parameter error (0, 65 or greater is set including the number of
		occupied stations)
		3. Baud rate switch setting error during cancellation of reset (5 or greater)

EtherCAT – Master – OPT60

The following table describes the meaning of the LEDs for the comX Real-Time Ethernet communication modules (COMX 100CA-RE/ COMX 100CN-RE) when the firmware of the EtherCAT Master protocol is loaded to the comX communication module:

RUN LED (COM0)

Color	/ State	Magning
Color	State	Meaning
Green	Off	INIT: The device is in state INIT
Green	Blinking	PRE-OPERATIONAL: The device is in PREOPERATIONAL state
Green	Flickering	BOOT: Device is in BOOT state
Green	Single Flash	SAFE-OPERATIONAL: The device is in SAFE-OPERATIONAL state
Green	On	OPERATIONAL: The device is in OPERATIONAL state

ERR LED (COM1)

Color	State	Meaning
Red	Off	Master has no errors
Red	On	Master has detected a communication error. The error is indicated in the DPM

LINK LED

Green LED on ETH0 connector

Color	State	Meaning
Green	On	A link is established
Green	Off	No link established

ACT LED

Yellow LED on ETH0 connector

Color	State	Meaning
Yellow	Flashing	The device sends/receives Ethernet frames

LED State Definition for EtherCAT Master for the RUN and ERR LEDs

Color	Meaning	
On	The indicator is constantly on.	
Off	The indicator is constantly off.	
Blinking	The indicator turns on and off with a frequency of 2,5 Hz: on for 200 ms, followed by off for 200 ms.	
Flickering	The indicator turns on and off with a frequency of approximately 10 Hz: on for approximately 50 ms, followed	
	by off for 50 ms.	
Single Flash	The indicator shows one short flash (200 ms) followed by a long off phase (1,000 ms).	
Double Flash	The indicator shows a sequence of two short flashes (each 200 ms), separated by a short off phase (200 ms).	
	The sequence is finished by a long off phase (1,000 ms).	

EtherCAT – Slave – OPT61

The following table describes the meaning of the LEDs for the comX Real-Time Ethernet communication modules (COMX 100CA-RE/ COMX 100CN-RE) when the firmware of the EtherCAT Slave protocol is loaded to the comX communication module:

RUN LED (COM0)

Color	State	Meaning
Green	Off	INIT: The device is in state INIT
Green	Blinking	PRE-OPERATIONAL: The device is in PREOPERATIONAL state
Green	Flickering	BOOT: Device is in BOOT state
Green	Single Flash	SAFE-OPERATIONAL: The device is in SAFE-OPERATIONAL state
Green	On	OPERATIONAL: The device is in OPERATIONAL state

ERR LED (COM1)

Color	State	Meaning
Red	Off	No error: The EtherCAT communication of the device is in working condition
Red	Blinking	Invalid Configuration: General Configuration Error Possible reason: State change commanded by master is impossible due to register or object settings.
Red	Single Flash	Local Error: Slave device application has changed the EtherCAT state autonomously. Possible reason 1: A host Watchdog timeout has occurred. Possible reason 2: Synchronization Error, device enters Safe-Operational automatically.
Red	Double Flash	Application Watchdog Timeout: An application Watchdog timeout has occurred. Possible reason: Sync Manager Watchdog timeout.

LINK/ACT LED

Green LED on ETH0(IN) / ETH1(OUT) connectors:

Color	State	Meaning
Green	On	A link is established
Green	Flashing	The device sends/receives Ethernet frames
Green	Off	No link established

Yellow LED on ETH0 / ETH1 connectors:

Color	State	Meaning
Yellow	-	-

LED State Definition for EtherCAT Slave for the RUN and ERR LEDs

Color	Meaning
On	The indicator is constantly on.
Off	The indicator is constantly off.
Blinking	The indicator turns on and off with a frequency of 2,5 Hz: on for 200 ms, followed by off for 200 ms.
Single Flash	The indicator shows one short flash (200 ms) followed by a long off phase (1,000 ms).
Double Flash	The indicator shows a sequence of two short flashes (each 200 ms), separated by a short off phase (200 ms).
	The sequence is finished by a long off phase (1,000 ms).

EtherNet/IP – Scanner/Master – OPT70

The following table describes the meaning of the LEDs for the comX Real-Time Ethernet communication modules (COMX-CA-RE/ COMX-CNRE) when the firmware of the EtherNet/IP Scanner (Master) protocol is loaded to the comX communication module:

MS LED (COM0)

Color	State	Meaning
Green	On	Device operational: If the device is operating correctly, the module status
		indicator shall be steady green.
Green	Flashing	Standby: If the device has not been configured, the module status indicator
		shall be flashing green.
Red	On	Major fault: If the device has detected a non-recoverable major fault, the
		module status indicator shall be steady red.
Green	Flashing	Minor fault: If the device has detected a recoverable minor fault, the module
		status indicator shall be flashing red. NOTE: An incorrect or inconsistent
		configuration would be considered a minor fault.
Red/Green	Flashing	Self-test: While the device is performing its power up testing, the module status
		indicator shall be flashing green/red.
-	Off	No power: If no power is supplied to the device, the module status indicator
		shall be steady off.

NS LED (COM1)

Color	State	Meaning
Green	On	Connected: If the device has at least one established connection (even to the
		Message Router), the network status indicator shall be steady green.
Green	Flashing	No connections: If the device has no established connections, but has obtained
		an IP address, the network status indicator shall be flashing green.
Red	On	Duplicate IP: If the device has detected that its IP address is already in use, the
		network status indicator shall be steady red.
Red	Flashing	Connection timeout: If one or more of the connections in which this device is
		the target has timed out, the network status indicator shall be flashing red. This
		shall be left only if all timed out connections are reestablished or if the device is
		reset.
Red/Green	Flashing	Self-test: While the device is performing its power up testing, the network
		status indicator shall be flashing green/red.
-	Off	Not powered, no IP address: If the device does not have an IP address (or is
		powered off), the network status indicator shall be steady off.

LINK LED

Green LED on ETH0 / ETH1 connectors:

Color	State	Meaning
Green	On	A connection to the Ethernet exists
Green	Off	The device has no connection to the Ethernet

ACT LED

Color	State	Meaning
Yellow	Flashing	The device sends/receives Ethernet frames

EtherNet/IP – Adaptor/Slave – OPT71

The following table describes the meaning of the LEDs for the comX Real-Time Ethernet communication modules (COMX-CA-RE/ COMX-CNRE) when the firmware of the EtherNet/IP Adapter (Slave) protocol is loaded to the comX communication module:

MS LED (COM0)

Color	State	Meaning
Green	On	Device operational: If the device is operating correctly, the module status
		indicator shall be steady green.
Green	Flashing	Standby: If the device has not been configured, the module status indicator
		shall be flashing green.
Red	On	Major fault: If the device has detected a non-recoverable major fault, the
		module status indicator shall be steady red.
Green	Flashing	Minor fault: If the device has detected a recoverable minor fault, the module
		status indicator shall be flashing red. NOTE: An incorrect or inconsistent
		configuration would be considered a minor fault.
Red/Green	Flashing	Self-test: While the device is performing its power up testing, the module status
		indicator shall be flashing green/red.
-	Off	No power: If no power is supplied to the device, the module status indicator
		shall be steady off.

NS LED (COM1)

Color	State	Meaning
Green	On	Connected: If the device has at least one established connection (even to the
		Message Router), the network status indicator shall be steady green.
Green	Flashing	No connections: If the device has no established connections, but has obtained
		an IP address, the network status indicator shall be flashing green.
Red	On	Duplicate IP: If the device has detected that its IP address is already in use, the
		network status indicator shall be steady red.
Red	Flashing	Connection timeout: If one or more of the connections in which this device is
		the target has timed out, the network status indicator shall be flashing red. This
		shall be left only if all timed out connections are reestablished or if the device is
		reset.
Red/Green	Flashing	Self-test: While the device is performing its power up testing, the network
		status indicator shall be flashing green/red.
-	Off	Not powered, no IP address: If the device does not have an IP address (or is
		powered off), the network status indicator shall be steady off.

LINK LED

Green LED on ETH0 / ETH1 connectors:

Color	State	Meaning
Green	On	A connection to the Ethernet exists
Green	Off	The device has no connection to the Ethernet

ACT LED

Color	State	Meaning
Yellow	Flashing	The device sends/receives Ethernet frames

Open Modbus/TCP – OPT80

The following table describes the meaning of the LEDs for the comX Real-Time Ethernet communication modules (COMX 100CA-RE/ COMX 100CN-RE) when the firmware of the Open Modbus/TCP protocol is loaded to the comX communication module:

RUN LED (COM0)

Color	State	Meaning
Green	Off	Not Ready
		OMB task is not ready
Green	Flashing cyclic with 1Hz	Ready, not configured yet
		OMB task is ready and not configured yet
Green	Flashing cyclic with 5Hz	Waiting for Communication:
		OMB task is configured
Green	On	Connected:
		OMB task has communication – at least one TCP connection is established

ERR LED (COM1)

Color	State	Meaning
Red	Off	No communication error
Red	Flashing cyclic with 2Hz (On/Off ratio 25%)	System error
Red/Green	On	Communication error active

LINK LED

Green LED on ETH0 / ETH1 connectors:

Color	State	Meaning
Green	On	A connection to the Ethernet exists
Green	Off	The device has no connection to the Ethernet

ACT LED

Color	State	Meaning
Yellow	Flashing	The device sends/receives Ethernet frames

PROFINET IO – Controller – OPT90

The following table describes the meaning of the LEDs for the comX Real-Time Ethernet communication modules (COMX 100CA-RE/ COMX 100CN-RE) when the firmware of the PROFINET IO-RT Controller protocol is loaded to the comX communication module:

SF LED (COM0)

Color	State	Meaning	
Red	On	(together with BF "red ON") No valid Master license	
Red	Flashing cyclic with 2Hz	System error: Invalid configuration, Watchdog error or internal error	
Red	Off	No error	

BF LED (COM1)

Color	State	Meaning	
Red	On	No Connection: No Link or (together with SF "red ON")	
		No valid Master license	
Red	Flashing cyclic with 2Hz	Configuration fault: not all configured IO-Devices are connected.	
Red	Off	No error	

LINK LED

Green LED on ETH0 / ETH1 connectors:

Color	State	Meaning
Green	On	A connection to the Ethernet exists
Green	Off	The device has no connection to the Ethernet

ACT LED

Color	State	Meaning
Yellow	Flashing	The device sends/receives Ethernet frames

PROFINET IO – Device – OPT91

The following table describes the meaning of the LEDs for the comX Real-Time Ethernet communication modules (COMX-CA-RE/ COMX-CNRE) when the firmware of the PROFINET IO-RT-Device protocol is loaded to the comX communication module:

SF LED (COM0)

Color	State	Meaning	
Red	On	Watchdog timeout; channel, generic or extended diagnosis present; system	
		error	
Red	Flashing cyclic with 2Hz	DCP signal service is initiated via the bus	
	(for 3 seconds)		
Red	Off	No error	

BF LED (COM1)

Color	State	Meaning	
Red	On	No configuration; or low speed physical link; or no physical link	
Red	Flashing cyclic with 2Hz	No data exchange	
Red	Off	No error	

LINK LED

Green LED on ETH0 / ETH1 connectors

Color	State	Meaning
Green	On	A connection to the Ethernet exists
Green	Off	The device has no connection to the Ethernet

ACT LED

Color	State	Meaning
Yellow	Flashing	The device sends/receives Ethernet frames

APPENDIX A – SETUP EXAMPLES

SYCON.net Setup

The following is a sample setup using an ACC-72EX Ethernet IP slave with an Allen-Bradley CompactLogix controller (1769-L18ERM-BB1B) as a master. SYCON.net for netX 1.310 was used in this example.

With the power off, plug the ACC-72EX into the UBUS backplane, and then power the UMAC rack. Connect the diagnostic port to a USB port on the PC using a micro-USB type cable.

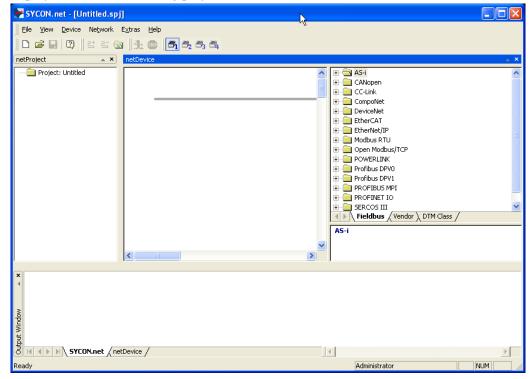
Launch the SYCON.NET software on the PC.



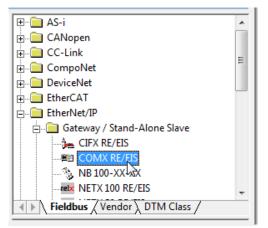
Enter the password:

SYCON.net User Login		
Hilscher SY0	CON.net	
<u>U</u> ser Name:	Administrator	-
Password:	*****	
	ОК	Cancel

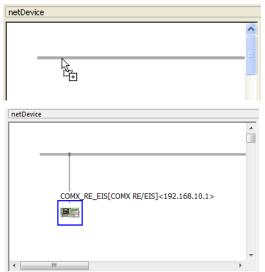
Start a new project or load an existing project from the File menu:



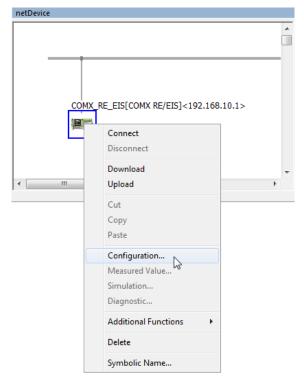
Select the COMX module, to which the USB is connected, from the Fieldbus protocol list:



Drag and drop the module onto the BusLine in the netDevice window (notice that the module can only be inserted on the BusLine):



Establish USB communications to the COMX gateway by right clicking on the device icon and selecting "Configuration...":



In the netDevice Configuration window, select the Driver folder under Settings folder in the Navigation Area, check the checkmark box for netX Driver on the driver list, and click Apply:

retDevice - Configuration CO	MX_RE_EIS[COMX RE/EIS]<192.168.10.1>		
IO Device: COMX I Vendor: Hilsche		Device Vendor	
Navigation Area		Driver	
Settings	Driver	Version	ID
netX Driver	3SGateway Driver for netX (V3.x)	0.9.1.2	{787CD3A9-4CF6-4259-8E4D-109B6A6BEA91}
Device Assignment	netX Driver	1.101.1.5347	{B54C8CC7-F333-4135-8405-6E12FC88EE62}
Firmware Download			
Configuration			
General			
Electronic Keying			
Connection			
Assembly			
Device Settings			
Contraction Contraction			
Device Info			

Select the netX Driver node in Driver folder in the Navigation Area, and select the port for the USB connection to the COMX module. Click Save and Apply (just click OK if Apply is grayed out):

IO Device: COMX 1 Vendor: Hilscher	00XX-RE/EIS GmbH	Device ID: Vendor ID:	0x0103 0x011B	•
Navigation Area Settings Triver Priver Device Assignment Firmware Download Configuration General Electronic Keying Connection Assembly Device Settings Description Device Info	Stop Bits: 1 Stopbit Send Timeout: 1000	e v arity v	Save	Save All
		 ОК	Cancel Apply	Help

Note: You can Check Windows Device Manager in order to identify which COM port provides the connection to the Hilscher COMX module:

🚔 Device Manager	
File Action View Help	
🗢 🔿 🖬 📓 🛛 🖬 🔍 😭 🚱	
🖌 🛁 LT-TechSupW703 🔹	
A-B Virtual Backplane	
🛛 🔊 Batteries	
> 📲 Computer	
Disk drives	
🔈 📲 Display adapters	
DVD/CD-ROM drives	
🕨 🕼 Human Interface Devices	
De ATA/ATAPI controllers	
🔈 📲 IEEE 1394 Bus host controllers	
🔉 🔚 Imaging devices	
⊳ Keyboards E	2
Mice and other pointing devices	
Monitors	
A Providence And Andrews	ł
Atheros AR9287 Wireless Network Adapter	
Marvell Yukon 88E8057 PCI-E Gigabit Ethernet Controller	
Microsoft Virtual WiFi Miniport Adapter	
Portable Devices	
Ports (COM & LPT)	
Hilscher comX (COM8)	
Processors	
Sound, video and game controllers	
⊳ ₁III System devices	
Universal Serial Bus controllers	"

Click the Device Assignment under Driver folder in the Navigation Area. Assign the netX Driver to the detected COMX module by checking the checkmark box next to the detected device, and click Apply:

	1X 100XX-RE/EIS her GmbH					Device ID: Vendor ID:	0x0103 0x011B	
lavigation Area 📃				Device Ass	signment			
Settings	Scan progress: 3/3 De	evices (Current device: -)						
netX Driver	Device selection:	suitable only						Scan
Firmware Download	Device	Hardware Port 0/1/2/3	Slot nu	Serial num	Driver	Channel Protocol	Access path	
Configuration General	COMX 100CN-	RE Ethernet/Ethernet/-/-	n/a	20311	netX Driver	EtherNet/IP Adapter	\COM8_cifX0_Ch0	
Electronic Keying								
Connection								
Assembly								
Device Settings								
Description								
Device Info								
	Access path:	{B54C8CC7-F333-4135-8-	405-6E12FC8	88EE62}\COM8	_cifX0_Ch0			

When used with Turbo PMAC, the reset line is released too fast for some Hilscher COMX modules, which puts them in a boot mode. This can prevent the device from being detected by Sycon.NET software. Make sure the device receives a system wide reset using the PMAC suggested M-variables ulSystemCommandCOS and HSF_RESET registers as shown here.



SCtrl_ulSystemCommandCOS=\$55AA55AA

HCSC_HSF_RESET=1

Note that ACC-72EX Setup Assistant software automatically resets the cards if it cannot detect the identification cookie.

retDevice - Configuration C	OMX_RE_EIS[COMX RE/E	IS]<192.168.1.210>				- • ×
	(RE/EIS er GmbH			Device ID: Vendor ID:	0x0103 0x011B	FDT
Navigation Area			General			
Settings	Description:	DMX_RE_EIS				
netX Driver Device Assignment Firmware Download Configuration Figuration Electronic Keying Connection Assembly Device Settings Description Device Info	IP Settings DHCP BootP Fixed Addresses IP Address: Network mask: Gateway: Note: The priority set	192 . 168 . 1 . 210 255 . 255 . 255 . 0 0 . 0 . 0 . 0 quence is DHCP, BootP, Fixed.				
	Operation mode:	All capable, Auto Negotiation	enabled			•
				ОК	Cancel Apply	Help
Disconnected 🚺 Data Set						//

Set the IP address for the COMX module in the General Configuration window:

Set Connections:

	DMX_RE_EIS[COMX RE/EI (RE/EIS er GmbH	5]<192.168.1.210>	Device ID: Vendor ID:	0x0103 0x011B	
Navigation Area 📄 🔄 Settings	Connection name:	Connect1	nection		
Driver netX Driver Device Assignment Firmware Download	Originator to Target	·			
Configuration General Electronic Keying	RT transfer format: Target to Originator	32-bit run/idle header	_		
Connection Assembly Device Settings	RT transfer format:	32-bit run/idle header O data length depends on existance of run/i	vidle beader (02T, T20)		
Description Device Info					
			ок с	ancel App	y Help
0⊳ Disconnected 🚺 Data Set		2			

Set Instance IDs and Data lengths in the Assembly window. 240 is the maximum length for the CompactLogix 1769-L18ERM-BB1B controller.

🕨 netDevice - Configuratio	on COMX_100XX_	RE_EIS[COMX 1	00XX-RE/EIS]<192.168.1.210>			_	- • X
	COMX 100XX-RE/EI Hilscher GmbH	5			Device ID: Vendor ID:	0x0103 0x011B	FD
Navigation Area				Assembly			
 Settings Driver netX Driver Device Assignmen Firmware Downloa Configuration General Electronic Keying Connection Assembly Device Settings Description Device Info 		IT Connect1	Connection name	Instance ID 101 101 100 2	Data length 240		Max. length 504
					ОК	Cancel A	oply Help
Disconnected 🚺 Data	Set						

RSLogix 5000 Setup

RSLogix 5000 version 20 is used in this example.

Launch RSLogix, and click on \underline{W} ho Active in the Communications pull down menu to find the CompactLogix controller:

😰 RSLogix 5000									
File	Edit	View	Search	Logic	Co	mmunications Tools			
						<u>W</u> ho Active			
						Select Recent Path			
No C	No Controller 🛛 🗸 🔲 RUN		IUN		<u>G</u> o Online				
No Forces					<u>U</u> pload				

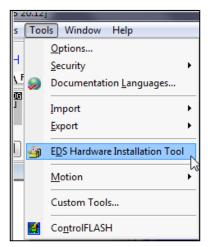
V Autobrowse Refresh	
□ ■ Workstation, LT-TECHSUPW703 □	Go Online Upload Download Update Firmward Close Help
ath: AB_ETHIP-1\192.168.1.200 ath in Project: <none></none>	Set Project Pat

Select the controller, and click the Go Online button to test communication:

The Controller OK indicator box should change to green like below:

8	RSLog	ix 5000) - Ethe	rNetIPTe	st [1769-	L18E
Ĭ	File	Edit	View	Search	Logic	Con
						Ī
Pro	gram		04	Progra	m Mode	
No	Forces		- ▶_	Contro		
No	Edits		2	E Battery	y UK at Doopou	dina
				= 1/0 NG	unespor	naing

Next, install the EDS file for the Hilscher COMX slave of the ACC-72EX. Go to the Tools pull down menu, and select EDS Hardware Installation Tool:



Click Next:



Select the Register an EDS file(s) radial button:

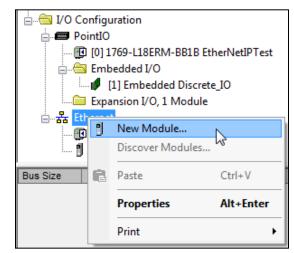


Browse to and select the Hilsher EDS file. EDS files can be downloaded at hilscher.com at http://www.hilscher.com/hcuk/support_software.html. Click Next.

Registration Electronic Data Sheet file(s) will t Automation applications.	be added to your system for use in Rockwell
 Register a single file 	
C Register a directory of EDS files	Look in subfolders
Named:	
C:\Users\techsupport\HILSCHER C	OMX-RE EIS V1.1.EDS Browse
• If there is an icon file (.ico) then this image will be assoc	with the same name as the file(s) you are registering ciated with the device.

Follow the directions in the remaining windows for finishing the EDS installation.

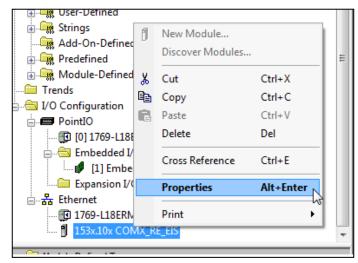
The next step will apply the EDS installation, but first the controller needs to be offline. Click the <u>Go</u> Offline selection under the Communications tab (<u>Go</u> Offline is displayed when the controller is online, and <u>Go</u> Online is displayed when offline). Under I/O Configuration in the Controller Organizer, right-click on Ethernet, and select New Module...:



Scroll down to and select the COMX slave module, and press Create:

Se	ect Module Type	ites			
	Enter Search Text for Module 1	Clear Filters			Show Filters 🛛
	Catalog Number	Description	Vendor	Category	*
	1336T-FORCEDrivePLC 1336T-FORCEDriveStd-E 1397DigitalDCDrive-EN1 150 SMC Flex-E 150-SMCDialogPlus-EN1	AC Drive, PLC Comm Adapter via 1203-EN1 AC Drive, Standard Adapter via 1203-EN1 DC Drive via 1203-EN1 Smart Motor Controller via 20-COMM-E Smart Motor Controller via 1203-EN1	Allen-Bradley Allen-Bradley Allen-Bradley Allen-Bradley Allen-Bradley	Drive Drive Drive Drive Drive	
	153x:10x 153x:10x 1557 1715-AENTR 1732E-12X4M12QCDR 1732E-16CFGM12 1732E-16CFGM12QCR 1732E-16CFGM12R 1732E-16CFGM12R 1732E-16CFGM12W	COMX RE/EIS COMX RE/EIM 1557 Medium Voltage AC Drive 1715 Ethemet Adapter, Twisted Pair Media 12 Point Input/4 Point Output 24V DC Quick Connect EtherNet/IP 16 Point Self-configuring 24VDC 16 Point 24V DC Self-Configuring Quick Connect Weld 16 Point 24V DC Self-Configuring Quick Connect Weld 16 Point 24V DC Self-Configuring 2-Port EtherNet/IP WeldBlock 16 Point Self-configuring 24/DC	Hilscher GmbH Hilscher GmbH Allen-Bradley Allen-Bradley Allen-Bradley Allen-Bradley Allen-Bradley Allen-Bradley Allen-Bradley Allen-Bradley	Communications Adapter Communications Adapter SCANport Drives on EtherNet/IP Communication Digital Digital Digital Digital Digital Digital	•
	231 of 231 Module Types Four	ld			Add to Favorites
	Close on Create			Create	Close Help

The created entry should appear under Ethernet in the Control Organizer. Right-click on it, and select Properties:



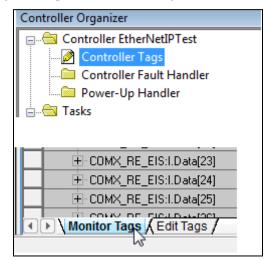
Under the General tab, set the IP address of the ACC-72EX:

💵 Module Proper	rties: Local (153x.10x 1.1)
General Conne	ection Module Info Internet Protocol Port Configuration Network
Туре:	153x.10x COMX RE/EIS
Vendor:	Hilscher GmbH
Parent:	Local
Name:	COMX_RE_EIS Ethernet Address
Description:	Private Network: 192.168.1. 210
	O IP Address:
	Host Name:
	O Host Name.
	-
~ Module Defin	nition
Revision:	1.1
Electronic Ke	eying: Exact Match
Connections:	Exclusive Owner , Input_CP=101, Output
	Change
CL	
Status: Offline	OK Cancel Apply Help

Check the settings under the Connection tab:

General	Connection Module Info Internet Protocol	Port Configuration Network		
	Name	Requested Packet Interv (RPI) (ms)	al Input Type	Input Trigger
Exclus	sive Owner , Input CP = 101, Output CP = 100	20.0 ≑ 1.0 - 3200.0	Unicast 🖉	Cyclic 💌
📰 Maji	oit Module or Fault On Controller If Connection Fails While in le Fault	Run Mode		

Double-click on Controller Tags, and open the Monitor Tags tab:



Click on "+" to expand the input data entries. Now input values from the ACC-72EX can be seen in the Value column (controller must be online).

	Favorites Add-On A	Alarms 🔏 Bit 🔏 Timer/Counter 🔏 Inp	ut/Output 🔏 Compar	re 🔏 Compute/Mat	n 🖌 Move/Logical 🚶
Controller Organizer 🚽 🕇 🗙	Scope: 🔞EtherNetIPTest 👻 Show: Al	ll Tags			
Controller EtherNetIPTest	Name == △	Value 🗲	Force Mask 🛛 🗲	Style	Data Type
Controller Fags	-COMX_RE_EIS:I	{}	{}		_011B:153x10x
Power-Up Handler	COMX_RE_EIS:I.ConnectionFault	0		Decimal	BOOL
	-COMX_RE_EIS:I.RunMode	1		Decimal	BOOL
🔒 🛱 MainTask	- COMX_RE_EIS:I.Data	{}	{}	Decimal	INT[240]
🚽 🖓 MainProgram	COMX_RE_EIS:I.Data[0]	24		Decimal	INT
Program Tags	+ COMX_RE_EIS:I.Data[1]	24		Decimal	INT
MainRoutine	COMX_RE_EIS:I.Data[2]	24		Decimal	INT
Unscheduled Programs	COMX_RE_EIS:I.Data[3]	24		Decimal	INT
🗄 📇 Motion Groups	+ COMX_RE_EIS:I.Data[4]	24		Decimal	INT
Ungrouped Axes	E - COMX_RE_EIS:I.Data[5]	24		Decimal	INT
Add-On Instructions	COMX_RE_EIS:I.Data[6]	24		Decimal	INT
🚊 🖂 Data Types		24		Decimal	INT
🗄 🛄 User-Defined		24		Decimal	INT
🖶 🛺 Strings	COMX_RE_EIS:I.Data[9]	24		Decimal	INT
Add-On-Defined	COMX_RE_EIS:I.Data[10]	24		Decimal	INT
		24		Decimal	INT
🗄 🖣 Module-Defined	COMX_RE_EIS:I.Data[12]	24		Decimal	INT
Irends	ECOMX_RE_EIS:I.Data[13]	24		Decimal	INT
	+ COMX_RE_EIS:I.Data[14]	24		Decimal	INT
011769-L18ERM-BB1B	+ COMX_RE_EIS:I.Data[15]	24		Decimal	INT
Embedded I/O	ECOMX_RE_EIS:I.Data[16]	24		Decimal	INT
	+ COMX_RE_EIS:I.Data[17]	24		Decimal	INT
F	COMX_RE_EIS:I.Data[18]	24		Decimal	INT

Click on "+" to expand the output data entries. The values seen in the Value column should now be seen as inputs in the ACC-72EX. Values can be changed here manually, or in program logic such as in the ladder logic example that follows.

Name		Value 🗲	Force Mask 💦 🔶	Style	Data Type
🗄 - COMX_RE_EIS:I.Dat	a[238]	23		Decimal	INT
🗄 - COMX_RE_EIS:I.Dat	a[239]	23		Decimal	INT
- COMX_RE_EIS:0		{}	{}		_011B:153x10x
⊡-COMX_RE_EIS:0.D ata		{}	{}	Decimal	INT[240]
🛨 - COMX_RE_EIS:0.Da	ta[0]	24		Decimal	INT
∃ COMX_RE_EIS:0.Da	ita[1]	24		Decimal	INT
🗄 - COMX_RE_EIS:0.Da	ita[2]	24		Decimal	INT
🗄 - COMX_RE_EIS:0.Da	ta[3]	24		Decimal	INT
🗄 - COMX_RE_EIS:O.Da	ta[4]	24		Decimal	INT
COMX_RE_EIS:0.Da	ita[5]	24		Decimal	INT
	ta[6]	24		Decimal	INT
🗄 - COMX_RE_EIS:0.Da	ita[7]	24		Decimal	INT
🗄 - COMX_RE_EIS:0.Da	ta[8]	24		Decimal	INT
🗄 - COMX_RE_EIS:0.Da	ita[9]	24		Decimal	INT
🛨 COMX_RE_EIS:0.Da	ta[10]	24		Decimal	INT
E - COMX_RE_EIS:0.Da	ita[11]	24		Decimal	INT
COMX_RE_EIS:0.Da	ita[12]	24		Decimal	INT
🗄 - COMX_RE_EIS:0.Da	ta[13]	24		Decimal	INT
COMX_RE_EIS:0.Da	ita[14]	24		Decimal	INT
	ta[15]	24		Decimal	INT
COMX_RE_EIS:0.Da	ita[16]	24		Decimal	INT
🕀 COMX_RE_EIS:0.Da	ita[17]	24		Decimal	INT
🗄 - COMX_RE_EIS:0.Da	ta[18]	24		Decimal	INT

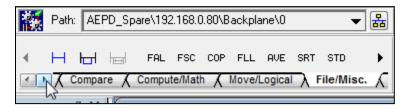
The following is an example which uses a "Copy" function to transfer all of the input values into corresponding output values. Double-click on MainRoutine:

Offline I RUN No Forces C BAT No Edits I I/O	Path: AEPD_Spare\192.168.0.80\Backplane\0 Image: Spare\192.168.0.80\Backplane\0 Image: Spare\192.168.0.80\Backplane\0 Image: Spare\192.168.0.80\Backplane\0 Image: Spare\192.168.0.80\Backplane\0 Image: Spare\192.168.0.80\Backplane\19	
Controller Organizer Controller AEPD_Spare Controller Tags Controller Fault Handler Power-Up Handler Tasks MainTask MainTask MainProgram MainRoutine Unscheduled Programs		

Click to select the top rung:

📕 MainPro	gram - MainR	outine*		×
电鹭	3 e e (abcd ab ab.	💌 <ab></ab>	
e e e (End)				•

Click on the right arrow as needed to bring into view the File/Misc. ladder entries tab:



Click on the File/Misc. tab, and then drag and drop COP (copy) onto a rung. Look for a green dot to appear on the left side of the rung when the cursor is hovered there, and then drop the COP function. To copy all the ACC-72EX inputs into corresponding CopactLogix outputs, set Source to COMX_RE_EIS:I.Data[0], Dest to COMX_RE_EIS:O.Data[0], and Length to 240:

Controller Organizer - 4 X	₩ 58 66 66 60 \$ 6. × ••	
Controller EtherNetiP lest		
Controller Tags		COP
Controller Fault Handler		Copy File
Power-Up Handler		Source COMX_RE_EIS:LData[0] Dest COMX_RE_EIS:O.Data[0]
Controller Fault Handler Power-Up Handler Sasks		ength 240
🛓 🤯 MainTask		
🚊 🚭 MainProgram		
Program Tags		
MainRoutine	ind)	
Unscheduled Programs		

COMX Test PLC

The following Turbo PMAC PLCs can be used to test the communication of the COMX module. Run PLC 1 and check the variable M_CommErrorFlag. If it is =0 after PLC 1 finishes, the COMX module is communicating properly.

```
CLOSE
END GAT
DEL GAT
#include "M-VariableDefinition $6C000.pmc"
#define M CommErrorFlag P1
#define timer i6612
#define msec *8388608/i10while(i6612>0)endwhile
OPEN PLC 1 CLEAR
DISABLE PLC 2..31 // Disable all other tasks
M SCtrl ulSystemCommandCOS=$55AA55AA // Reset token for MASTER Unit
M HCSC HSF RESET=1 // Reset bit, token required for reset to complete
M CommErrorFlag=0
timer = 4000 msec // Reset Time-out Timer
WHILE (M CommErrorFlag=0 AND M HCSC NSF READY=0) // Wait for reset to complete
IF (timer<0) // Check for reset timeout
M CommErrorFlag = 1
ENDIF
ENDWHILE
IF (M CommErrorFlag=0) //
WHILE (M CCO RCX COMM COS RUN=0) // wait for comm tasks to
// start on COMX modules
M HCCCO HCF NETX COS ACK = M HCCCO HCF NETX COS ACK ^ 1
// Toggle Communication Channel 0's Change of State Acknowledge bit in
\prime\prime order to read the CCO RCX COMM COS RUN which is a part of Communication
// Channel 0 State Register
enable plc 2
ENDWHILE
ENDIF
DISABLE PLC 1
CLOSE
open plc2 clr
timer =4000 msec // Reset Time-out Timer
//M CCO RCX APP COS APP READY=1
       CSC_HSF_HOST_COS_CMD = M_HCSC_NSF_HOST_COS_ACK)
M_CC0_RCX_APP_COS_BUS_ON=1 // Setting the Bus On flag for 1st ACC-72EX
IF (M HCSC HSF HOST COS CMD
       M CCO RCX APP COS BUS ON ENABLE=1
 M HCSC HSF HOST COS CMD
                               = M HCSC HSF HOST COS CMD^1
ENDIF
timer = 1000 msec // Reset Time-out Timer
ENABLE PLC 28
ENABLE PLC 10
ENABLE PLC 11
ENABLE PLC 26
disable plc2
close
OPEN PLC 28 CLEAR
M CCO ulDeviceWatchdog = M CCO ulHostWatchdog // copies the host watchdog content
CLOSE
OPEN PLC 10 CLEAR
IF (M HCCC0 HCF PD0 OUT CMD = M HCCC0 NCF PD0 OUT ACK) // Making sure the ACK flag matches the
CMD
```

M_HCCC0_HCF_PD0_OUT_CMD = M_ ENDIF	HCCC0_HCF_PD0_OUT_CMD^1 // Toggling the CMD flag (^: XOR)
CLOSE	
	1_HCCC0_HCF_PD0_IN_ACK) // If CMD flag and ACK flags are = M_HCCC0_HCF_PD0_IN_ACK ^ 1 // toggle the acknowledge bit
CLOSE	
i5=2 Ena plc1	
m91->x:\$6C4C0,0,16 M92->y:\$6CA60,0,16 m93->x:\$6CA60,0,16 M94->Y:\$00405A,0,20 m95->Y:\$00405B,0,20	;Read Address ;address of M90 ;address of m91
	;address of M92 ;address of m93

The above PLCs require the following header files:

ſ	// MacroNameDefinition \$6C000.h				
	#define M SI abCookie 0	M5000			
	#define M_SI_abCookie_1_	M5001			
	#define M_SI_abCookie_2_	M5002			
	#define M_SI_abCookie_3_	M5003			
	#define M_SI_ulDpmTotalSize	M5004			
	#define M_SI_ulDeviceNumber	M5005			
	#define M SI ulSerialNumber	M5006			
	#define M_SI_ausHwOptions 0	M5007			
	#define M SI ausHwOptions 1	M5008			
	#define M SI ausHwOptions 2	M5009			
	#define M SI ausHwOptions 3	M5010			
	#define M SI usManufacturer	M5011			
	#define M_SI_usProductionDate	M5012			
	#define M_SI_ulLicenseFlags1	M5013			
	#define M SI ulLicenseFlags2	M5014			
	#define M SI usNetxLicenseID	M5015			
	#define M_SI_usNetxLicenseFlags		M5016		
	#define M_SI_usDeviceClass	M5017			
	#define M SI bHwRevision	M5018			
	#define M_SI_bHwCompatibility	M5019			
	#define M_SI_bDevIdNumber	M5020			
	#define M_SCI_bChannelType	M5021			
	#define M_SCI_bSizePositionOfHandsha	ke		M5022	
		M5023			
		M5024			
	#define M_SCI_usSizeOfMailbox	M5025			
	#define M_SCI_usMailboxStartOffset		M5026		
		M5027			
	#define M_HCI_ulSizeOfChannel	M5028			
	#define M_CC0I_bChannelType	M5029			
	#define M_CC01_bChannelId	M5030			
	#define M_CC01_bSizePositionOfHandsh	ake		M5031	
	#define M_CC01_bNumberOfBlocks		M5032		
	#define M_CC0I_ulSizeOfChannel		M5033		
	#define M CC0I usCommunicationClass		M5034		
	#define M_CC0I_usProtocolClass		M5035		
	#define M_CC0I_usConformanceClass		M5036		
	#define M_CC1I_bChannelType	M5037			
	#define M CC1I bChannelId	M5038			
	#define M CC1I bSizePositionOfHandsh	ake		M5039	
	#define M CC1I bNumberOfBlocks		M5040		

		-		145 0 4 1	
	M_CC1I_ulSizeOfChanne			M5041	
	M_CC1I_usCommunication			M5042	
	M_CC1I_usProtocolClas			M5043	
#define	M_CC1I_usConformanceC	lass		M5044	
#define	M CC2I bChannelType		M5045		
#define	M CC2I bChannelId		M5046		
	M CC2I bSizePositionO	fHandsha	ake		M5047
	M CC2I bNumberOfBlock			M5048	
	M CC2I ulSizeOfChanne			M5049	
	M CC2I usCommunication			M5050	
	M CC2I usProtocolClass			M5050 M5051	
	M_CC2I_usConformanceC			M5052	
	M_CC3I_bChannelType		M5053		
	M_CC3I_bChannelId		M5054		
	M_CC3I_bSizePositionO		ake		M5055
#define	M_CC3I_bNumberOfBlock	s		M5056	
#define	M CC3I ulSizeOfChanne	1		M5057	
#define	M CC3I usCommunication	nClass		M5058	
	M CC3I usProtocolClass			M5059	
	M CC3I usConformanceC			M5060	
	M ACOI bChannelType		M5061	110000	
	M_ACOI_bChannelId		M5061 M5062		
					M5063
	M_ACOI_bSizePositionO		аке		M2063
	M_AC0I_bNumberOfBlock			M5064	
	M_ACOI_ulSizeOfChanne	T		M5065	
	M_AC1I_bChannelType		M5066		
#define	M_AC1I_bChannelId		M5067		
#define	M AC1I bSizePositionO	fHandsha	ake		M5068
#define	M AC1I bNumberOfBlock	s		M5069	
#define	M AC1I ulSizeOfChanne	1		M5070	
	M SCtrl ulSystemComman			M5071	
	M SStat ulSystemCOS		M5072		
	M SStat ulSystemStatu:	q	110072	M5073	
			M5074	115075	
	M_SStat_ulSystemError				
	M_SStat_ulBootError		M5075	115076	
	M_SStat_ulTimeSinceSt	art		M5076	
	M_SStat_usCpuLoad		M5077		
	M_SStat_ulHWFeatures		M5078		
#define	M_SSMB_usPackagesAcce	pted		M5079	
#define	M SSMB ulDest	M5080			
#define	M SSMB ulSrc	M5081			
#define	M SSMB ulDestId		M5082		
	M SSMB ulSrcId	M5083			
	M SSMB ullen	M5084			
	M SSMB ulld	M5085			
		M5086			
	M_SSMB_uiState M_SSMB_uiCmd	M5087			
	M_SSMB_ulExt	M5088			
	M_SSMB_ulRout	M5089			
	M_SSMB_ultData0		M5090		
	M_SSMB_ultData1		M5091		
	M_SSMB_ultData2		M5092		
#define	M SSMB ultData3		M5093		
#define	M SSMB ultData4		M5094		
	M SSMB ultData5		M5095		
	M SSMB ultData6		M5096		
	M SSMB ultData7		M5097		
	M SSMB ultData8		M5098		
	M_SSMB_ultData9		M5099		
	M_SSMB_ultData10		M5100		
	M_SSMB_ultData11		M5101		
	M_SSMB_ultData12		M5102		
	M_SSMB_ultData13		M5103		
	M_SSMB_ultData14		M5104		
#define	M_SSMB_ultData15		M5105		
#define	M_SSMB_ultData16		M5106		
	M SSMB ultData17		M5107		
	M SSMB ultData18		M5108		
	M SSMB ultData19		M5109		
	M SSMB ultData20		M5110		
	M_SSMB_uitbata20 M_SRMB_usWaitingPacka	nes	110 1 1 0	M5111	
" actile	ii_biuib_ubwai ciliyi acka	900		1.10 T T T	

#define	M_SRMB_ulDest M5112		
#define	M_SRMB_ulSrc M5113		
#define	M_SRMB_ulDestId	M5114	
#define	M SRMB ulSrcId M5115		
	M SRMB ullen M5116		
	M SRMB ulid M5117		
	M SRMB ulState M5118		
#define	M SRMB ulCmd M5119		
#define	M_SRMB_ulExt M5120		
#define	M SRMB ulRout M5121		
#define	M_SRMB_ulRout M5121 M_SRMB_ultData0 M_SRMB_ultData1	M5122	
#define	M SRMB ultData1	M5123	
#define	M SRMB ultData2	M5124	
	M SRMB ultData3	M5125	
	M SRMB ultData5	M5126 M5127	
	M SRMB ultData6	M5128	
	M SRMB_ultData7	M5128 M5129	
	M SRMB_ultData8	M5130	
	M SRMB_ultData9	M5130 M5131	
		M5132 M5133	
	M SRMB ultData12		
	M SRMB_ultData13	M5134 M5135	
	M_SRMB_ultData15 M_SRMB_ultData14	M5135 M5136	
		M5130 M5137	
		M5137 M5138	
	M SRMB_ultData10	M5130 M5139	
	M SRMB_ultData18	M5140	
#define		M5140 M5141	
#define	M SRMB_ultData19	M5141 M5142	
#define	M_HCSC_bNetyFlags	M5142 M5143	
#define		M5143 M5144	
#define	M HCSC NSF ERROR	M5145	
#define	M_HCSC_NSF_HOST_COS_ACK	MJ14J	M5146
#define	M HCSC NSF NETTY COS CMD		M5140 M5147
#define	M_HCSC_NSF_NETX_COS_CMD M_HCSC_NSF_SEND_MBX_ACK		M5147 M5148
#define	M_HCSC_NSF_RECV_MBX_CMD		M5140 M5149
#define	M_HCSC_bHostFlags	M5150	MJIHJ
#define	M_HCSC_bHostFlags M_HCSC_HSF_RESET	M5150 M5151	
#define	M_HCSC_HSF_KESEI M_HCSC_HSF_BOOTSTART	M5151 M5152	
#define	M HCSC HSF HOST COS CMD	MJ1J2	M5153
#define	M HCSC HSE NETX COS ACK		M5154
#define	M HCSC HSE SEND MBX CMD		
#define	H HODO HDI DEND HEA CHD		
	M HOSO HSE RECV MBY ACK		M5155
#detine	M_HCSC_HSF_HOST_COS_CMD M_HCSC_HSF_NETX_COS_ACK M_HCSC_HSF_SEND_MBX_CMD M_HCSC_HSF_RECV_MBX_ACK M_HCCCQ_USNetxFlags	M5157	
#deline	M ACCCU USNetXFIAGS	M5157	M5155 M5156
#define	M_HCCC0_USNELXFIAGS M_HCCC0_NCF_COMMUNICATING		M5155
#define #define	M_HCCC0_USNetXFlags M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR	M5157 M5159	M5155 M5156 M5158
#define #define #define #define	M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK		M5155 M5156 M5158 M5160
#define #define #define #define #define	M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD		M5155 M5156 M5158 M5160 M5161
<pre>#define #define #define #define #define #define #define</pre>	M_HCCC0_USNELXFIAGS M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK		M5155 M5156 M5158 M5160 M5161 M5162
<pre>#define #define #define #define #define #define #define #define</pre>	M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_RECV_MBX_CMD		M5155 M5156 M5158 M5160 M5161 M5162 M5163
<pre>#define #define #define #define #define #define #define #define #define</pre>	M_HCCC0_USNELXFlags M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_RECV_MBX_CMD M_HCCC0_NCF_PD0_OUT_ACK		M5155 M5156 M5158 M5160 M5161 M5162 M5163 M5164
<pre>#define #define #define #define #define #define #define #define #define #define</pre>	M_HCCC0_USNELXFlags M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_RECV_MBX_CMD M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD0_IN_CMD		M5155 M5156 M5158 M5160 M5161 M5162 M5163 M5164 M5165
<pre>#define #define #define #define #define #define #define #define #define #define #define</pre>	M_HCCC0_USNETRIAGS M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_RECV_MBX_CMD M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD0_IN_CMD M_HCCC0_NCF_PD1_OUT_ACK		M5155 M5156 M5160 M5161 M5162 M5163 M5164 M5165 M5166
<pre>#define #define #define</pre>	M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_MBX_CMD M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD0_IN_CMD M_HCCC0_NCF_PD1_OUT_ACK M_HCCC0_NCF_PD1_IN_CMD	M5159	M5155 M5156 M5158 M5160 M5161 M5162 M5163 M5164 M5165
<pre>#define #define #define</pre>	M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGN M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGS M_HCCC0_USNETRIAGS		M5155 M5156 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167
<pre>#define #define #define</pre>	M_HCCC0_USNECKF14gs M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PEND_MBX_ACK M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD1_OUT_ACK M_HCCC0_NCF_PD1_OUT_ACK M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOstFlags M_HCCC0_HCF_HOST_COS_CMD	M5159	M5155 M5156 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169
<pre>#define #define #define</pre>	M_HCCC0_USNECKF14GS M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_UT_ACK M_HCCC0_NCF_PD0_UT_ACK M_HCCC0_NCF_PD1_UT_ACK M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOstFlags M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_NETX_COS_ACK	M5159	M5155 M5156 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169 M5170
<pre>#define #define #define</pre>	M_HCCC0_USNELXFlags M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_ERROR M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD0_UT_ACK M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_NCF_PD1_UT_ACK M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOSTFlags M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_SEND_MBX_CMD	M5159	M5155 M5156 M5158 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169 M5170 M5171
<pre>#define #define #define</pre>	M_HCCC0_USNELXFlags M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD0_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOSTFlags M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_CMD M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_RECV_MBX_ACK	M5159	M5155 M5156 M5158 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169 M5170 M5171 M5172
<pre>#define #define #define</pre>	M_HCCC0_USNELXFlags M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD0_IN_CMD M_HCCC0_NCF_PD1_OUT_ACK M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOSTFlags M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_RECV_MBX_ACK M_HCCC0_HCF_RECV_MBX_ACK M_HCCC0_HCF_PD0_OUT_CMD	M5159	M5155 M5156 M5158 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169 M5170 M5171 M5172 M5173
<pre>#define #define #define</pre>	M_HCCC0_USNETRIAGS M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD0_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOSTFlags M_HCCC0_HCF_HOST_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_RECV_MBX_ACK M_HCCC0_HCF_PD0_OUT_CMD M_HCCC0_HCF_PD0_IN_ACK	M5159	M5155 M5156 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169 M5170 M5171 M5172 M5173 M5174
<pre>#define #define #define</pre>	M_HCCC0_USNETRIAGS M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD0_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOSTFLAGS M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_HOST_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_PD0_OUT_CMD M_HCCC0_HCF_PD0_IN_ACK M_HCCC0_HCF_PD0_IN_ACK M_HCCC0_HCF_PD1_OUT_CMD	M5159	M5155 M5156 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169 M5170 M5170 M5172 M5172 M5173 M5174 M5175
<pre>#define #define #define #define #define #define #define #define #define #define #define #define #define #define #define #define #define #define #define #define</pre>	M_HCCC0_USNECKF1495 M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_HOST_COS_ACK M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_UT_ACK M_HCCC0_NCF_PD0_UT_ACK M_HCCC0_NCF_PD1_UT_ACK M_HCCC0_NCF_PD1_UT_ACK M_HCCC0_USHOSTF1ags M_HCCC0_HCF_PD1_UT_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_RECV_MBX_ACK M_HCCC0_HCF_RECV_MBX_ACK M_HCCC0_HCF_PD0_UT_CMD M_HCCC0_HCF_PD1_OUT_CMD M_HCCC0_HCF_PD1_UN_ACK	M5159 M5168	M5155 M5156 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169 M5170 M5171 M5172 M5173 M5174
<pre>#define #define #</pre>	M_HCCC0_USNECRIARS M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_ERROR M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_UT_ACK M_HCCC0_NCF_PD0_UT_ACK M_HCCC0_NCF_PD1_UT_ACK M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_RECV_MBX_ACK M_HCCC0_HCF_RECV_MBX_ACK M_HCCC0_HCF_PD0_UT_CMD M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK	M5159	M5155 M5156 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169 M5170 M5171 M5172 M5173 M5174 M5175 M5176
<pre>#define #define #</pre>	M_HCCC0_USNECKTIAGS M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_ERROR M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_UT_ACK M_HCCC0_NCF_PD0_UT_ACK M_HCCC0_NCF_PD1_UT_ACK M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_HCF_PD1_IN_CMD M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_RECV_MBX_ACK M_HCCC0_HCF_PD0_UT_CMD M_HCCC0_HCF_PD1_UT_CMD M_HCCC0_HCF_PD1_OUT_CMD M_HCCC0_HCF_PD1_OUT_CMD M_HCCC0_HCF_PD1_UT_CMD M_HCCC0_HCF_PD1_UT_CMD M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC1_USNELXFLAGS M_HCCC1_NCF_COMMUNICATING	M5159 M5168 M5177	M5155 M5156 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5169 M5170 M5170 M5172 M5172 M5173 M5174 M5175
<pre>#define #define #</pre>	M_HCCC0_USNECRIARS M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_ERROR M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_UNT_ACK M_HCCC0_NCF_PD0_UNT_ACK M_HCCC0_NCF_PD1_OUT_ACK M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOSTFLags M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_PD0_UT_CMD M_HCCC0_HCF_PD0_UT_CMD M_HCCC0_HCF_PD1_UN_ACK M_HCCC0_HCF_PD1_UN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC1_USNETXFLags M_HCCC1_NCF_COMMUNICATING M_HCCC1_NCF_ERROR	M5159 M5168	M5155 M5156 M5158 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5170 M5171 M5172 M5173 M5175 M5176 M5178
<pre>#define #define #</pre>	M_HCCC0_USNECKFlags M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_ERROR M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_OUT_ACK M_HCCC0_NCF_PD0_UN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOSTFlags M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_PD0_UT_CMD M_HCCC0_HCF_PD0_UT_CMD M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_UT_CMD M_HCCC0_HCF_PD1_UT_CMD M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC1_USNETXFLAGS M_HCCC1_NCF_COMMUNICATING M_HCCC1_NCF_ERROR M_HCCC1_NCF_HOST_COS_ACK	M5159 M5168 M5177	M5155 M5156 M5158 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5170 M5171 M5172 M5173 M5174 M5175 M5178 M5178
<pre>#define #define</pre>	M_HCCC0_USNECRIARS M_HCCC0_NCF_COMMUNICATING M_HCCC0_NCF_ERROR M_HCCC0_NCF_ERROR M_HCCC0_NCF_NETX_COS_CMD M_HCCC0_NCF_SEND_MBX_ACK M_HCCC0_NCF_PD0_UNT_ACK M_HCCC0_NCF_PD0_UNT_ACK M_HCCC0_NCF_PD1_OUT_ACK M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_NCF_PD1_IN_CMD M_HCCC0_USHOSTFLags M_HCCC0_HCF_HOST_COS_CMD M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_NETX_COS_ACK M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_SEND_MBX_CMD M_HCCC0_HCF_PD0_UT_CMD M_HCCC0_HCF_PD0_UT_CMD M_HCCC0_HCF_PD1_UN_ACK M_HCCC0_HCF_PD1_UN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC0_HCF_PD1_IN_ACK M_HCCC1_USNETXFLags M_HCCC1_NCF_COMMUNICATING M_HCCC1_NCF_ERROR	M5159 M5168 M5177	M5155 M5156 M5158 M5160 M5161 M5162 M5163 M5164 M5165 M5166 M5167 M5170 M5171 M5172 M5173 M5175 M5176 M5178

#define	M HCCC1 NCF RECV MBX CMD		M5183
	M HCCC1 NCF PD0 OUT ACK		M5184
	M_HCCC1_NCF_PD0_IN_CMD		M5185
#deiine	M_HCCC1_NCF_PD1_OUT_ACK		M5186
#define	M HCCC1 NCF PD1 IN CMD		M5187
#define	M HCCC1 usHostFlags	M5188	
	M HCCC1 HCF HOST COS CMD		M5189
	M_HCCC1_HCF_NETX_COS_ACK		M5190
	M_HCCC1_HCF_SEND_MBX_CMD		M5191
#define	M_HCCC1_HCF_RECV_MBX_ACK		M5192
	M HCCC1 HCF PD0 OUT CMD		M5193
	M HCCC1 HCF PD0 IN ACK		M5194
	M HCCC1 HCF PD1 OUT CMD		M5195
	M_HCCC1_HCF_PD1_IN_ACK		M5196
#define	M_HCCC2_usNetxFlags	M5197	
#define	M HCCC2 NCF COMMUNICATING		M5198
#define	M HCCC2 NCF ERROR	M5199	
	M HCCC2 NCF HOST COS ACK	110 1 9 9	M5200
	M_HCCC2_NCF_NETX_COS_CMD		M5201
#define	M_HCCC2_NCF_SEND_MBX_ACK		M5202
#define	M HCCC2 NCF RECV MBX CMD		M5203
#define	M HCCC2 NCF PD0 OUT ACK		M5204
	M HCCC2 NCF PD0 IN CMD		M5205
	M_HCCC2_NCF_PD1_OUT_ACK		M5206
#define	M_HCCC2_NCF_PD1_IN_CMD		M5207
#define	M HCCC2 usHostFlags	M5208	
#define	M HCCC2 HCF HOST COS CMD		M5209
	M HCCC2 HCF NETX COS ACK		M5210
	M_HCCC2_HCF_SEND_MBX_CMD		M5211
#define	M_HCCC2_HCF_RECV_MBX_ACK		M5212
#define	M HCCC2 HCF PD0 OUT CMD		M5213
#define	M HCCC2 HCF PD0 IN ACK		M5214
	M HCCC2 HCF PD1 OUT CMD		M5215
	M HCCC2 HCF PD1 IN ACK		M5216
			MJZIO
	M_HCCC3_usNetxFlags	M5217	
#dofino	M HCCC3 NCF COMMUNICATING		
#derine	<u></u>		M5218
	M HCCC3 NCF ERROR	M5219	M5218
#define	M_HCCC3_NCF_ERROR	M5219	M5218 M5220
#define #define	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK	M5219	M5220
#define #define #define	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD	M5219	M5220 M5221
#define #define #define #define	M HCCC3_NCF_ERROR M HCCC3_NCF_HOST_COS_ACK M HCCC3_NCF_NETX_COS_CMD M HCCC3_NCF_SEND_MBX_ACK	M5219	M5220 M5221 M5222
#define #define #define #define #define	M HCCC3_NCF_ERROR M HCCC3_NCF_HOST_COS_ACK M HCCC3_NCF_NETX_COS_CMD M HCCC3_NCF_SEND_MBX_ACK M HCCC3_NCF_RECV_MBX_CMD	M5219	M5220 M5221 M5222 M5223
#define #define #define #define #define	M HCCC3_NCF_ERROR M HCCC3_NCF_HOST_COS_ACK M HCCC3_NCF_NETX_COS_CMD M HCCC3_NCF_SEND_MBX_ACK	M5219	M5220 M5221 M5222
<pre>#define #define #define #define #define #define #define</pre>	M HCCC3_NCF_ERROR M HCCC3_NCF_HOST_COS_ACK M HCCC3_NCF_NETX_COS_CMD M HCCC3_NCF_SEND_MBX_ACK M HCCC3_NCF_RECV_MBX_CMD	M5219	M5220 M5221 M5222 M5223
<pre>#define #define #define #define #define #define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD0_IN_CMD	M5219	M5220 M5221 M5222 M5223 M5224 M5225
<pre>#define #define #define #define #define #define #define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD0_IN_CMD M_HCCC3_NCF_PD1_OUT_ACK	M5219	M5220 M5221 M5222 M5223 M5224 M5225 M5226
<pre>#define #define #define #define #define #define #define #define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD0_IN_CMD M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD		M5220 M5221 M5222 M5223 M5224 M5225
<pre>#define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOSTFlags	M5219 M5228	M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227
<pre>#define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOstFlags M_HCCC3_HCF_HOST_COS_CMD		M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227 M5229
<pre>#define #define #define</pre>	M HCCC3_NCF_ERROR M HCCC3_NCF_HOST_COS_ACK M HCCC3_NCF_NETX_COS_CMD M HCCC3_NCF_SEND_MBX_ACK M HCCC3_NCF_RECV_MBX_CMD M HCCC3_NCF_PD0_OUT_ACK M HCCC3_NCF_PD0_IN_CMD M HCCC3_NCF_PD1_OUT_ACK M HCCC3_NCF_PD1_IN_CMD M HCCC3_USF_PD1_IN_CMD M HCCC3_USF_SENT_ACK M HCCC3_HCF_HOST_COS_CMD M HCCC3_HCF_NETX_COS_ACK		M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227 M5229 M5229
<pre>#define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOstFlags M_HCCC3_HCF_HOST_COS_CMD		M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227 M5229
<pre>#define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD0_IN_CMD M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOstFlags M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_SEND_MBX_CMD		M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227 M5229 M5230 M5231
<pre>#define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD0_IN_CMD M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOSTFLags M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_SEND_MBX_CMD M_HCCC3_HCF_RECV_MBX_ACK		M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227 M5229 M5230 M5231 M5232
<pre>#define #define #define</pre>	M HCCC3 NCF ERROR M HCCC3 NCF HOST COS ACK M HCCC3 NCF HOST COS ACK M HCCC3 NCF SEND MBX ACK M HCCC3 NCF SEND MBX ACK M HCCC3 NCF PD0 OUT ACK M HCCC3 NCF PD0 IN CMD M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 IN CMD M HCCC3 LF HOST COS CMD M HCCC3 HCF NETX COS ACK M HCCC3 HCF SEND MBX CMD M HCCC3 HCF SEND MBX CMD M HCCC3 HCF RECV MBX ACK M HCCC3 HCF PD0 OUT CMD		M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227 M5229 M5230 M5231 M5232 M5233
<pre>#define #define #define</pre>	M HCCC3 NCF ERROR M HCCC3 NCF HOST COS ACK M HCCC3 NCF HOST COS ACK M HCCC3 NCF SEND MBX ACK M HCCC3 NCF ZEV MBX CMD M HCCC3 NCF PD0 OUT ACK M HCCC3 NCF PD0 IN CMD M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 IN CMD M HCCC3 HCF HOST COS CMD M HCCC3 HCF NETX COS ACK M HCCC3 HCF SEND MBX CMD M HCCC3 HCF RECV MBX ACK M HCCC3 HCF PD0 OUT CMD M HCCC3 HCF PD0 OUT CMD M HCCC3 HCF PD0 IN ACK		M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227 M5229 M5230 M5231 M5233 M5233 M5233
<pre>#define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_SEND_MBX_ACK M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD1_OUT_CMD		M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227 M5229 M5230 M5231 M5232 M5233
<pre>#define #define #define</pre>	M HCCC3 NCF ERROR M HCCC3 NCF HOST COS ACK M HCCC3 NCF HOST COS ACK M HCCC3 NCF SEND MBX ACK M HCCC3 NCF ZEV MBX CMD M HCCC3 NCF PD0 OUT ACK M HCCC3 NCF PD0 IN CMD M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 IN CMD M HCCC3 HCF HOST COS CMD M HCCC3 HCF NETX COS ACK M HCCC3 HCF SEND MBX CMD M HCCC3 HCF RECV MBX ACK M HCCC3 HCF PD0 OUT CMD M HCCC3 HCF PD0 OUT CMD M HCCC3 HCF PD0 IN ACK		M5220 M5221 M5222 M5223 M5224 M5225 M5226 M5227 M5229 M5230 M5231 M5233 M5233 M5233
<pre>#define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOSTFlags M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_SEND_MBX_ACK M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK		M5220 M5221 M5222 M5223 M5226 M5226 M5227 M5229 M5230 M5231 M5231 M5233 M5233 M5233 M5233
<pre>#define #define #define</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK	M5228	M5220 M5221 M5222 M5223 M5225 M5227 M5227 M5230 M5231 M5232 M5233 M5233 M5233 M5235
<pre>#define #define #define</pre>	M HCCC3 NCF ERROR M HCCC3 NCF HOST COS ACK M HCCC3 NCF NETX COS CMD M HCCC3 NCF SEND MBX ACK M HCCC3 NCF RECV MBX CMD M HCCC3 NCF PD0 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 IN CMD M HCCC3 NCF PD1 IN CMD M HCCC3 HCF HOST COS CMD M HCCC3 HCF NETX COS ACK M HCCC3 HCF NETX COS ACK M HCCC3 HCF SEND MBX CMD M HCCC3 HCF RECV MBX ACK M HCCC3 HCF PD0 OUT CMD M HCCC3 HCF PD0 IN ACK M HCCC3 HCF PD1 OUT CMD M HCCC3 HCF PD1 OUT CMD M HCCC3 HCF PD1 OUT CMD M HCCC3 HCF PD1 IN ACK M HCAC0 USNETXFLAGS M HCAC0 NCF COMMUNICATING	M5228 M5237	M5220 M5221 M5222 M5223 M5226 M5226 M5227 M5229 M5230 M5231 M5231 M5233 M5233 M5233 M5233
<pre>#define #define #</pre>	M HCCC3 NCF ERROR M HCCC3 NCF HOST COS ACK M HCCC3 NCF NETX COS CMD M HCCC3 NCF SEND MBX ACK M HCCC3 NCF RECV MBX CMD M HCCC3 NCF PD0 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 IN CMD M HCCC3 NCF PD1 IN CMD M HCCC3 HCF NETX COS CMD M HCCC3 HCF NETX COS ACK M HCCC3 HCF SEND MBX CMD M HCCC3 HCF RECV MBX ACK M HCCC3 HCF PD0 OUT CMD M HCCC3 HCF PD0 IN ACK M HCCC3 HCF PD1 IN ACK M HCCC3 HCF PD1 IN ACK M HCCC3 HCF PD1 IN ACK M HCCC0 USNETXFLAGS M HCAC0 NCF COMMUNICATING M HCAC0 NCF ERROR	M5228	M5220 M5221 M5222 M5223 M5225 M5226 M5227 M5229 M5230 M5231 M5232 M5233 M5233 M5233 M5236 M5238
<pre>#define #define #</pre>	M HCCC3 NCF ERROR M HCCC3 NCF HOST COS ACK M HCCC3 NCF NETX COS CMD M HCCC3 NCF SEND MBX ACK M HCCC3 NCF ZEND MBX ACK M HCCC3 NCF PD0 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 IN CMD M HCCC3 NCF PD1 IN CMD M HCCC3 HCF PD1 OUT ACK M HCCC3 HCF PD1 OUT CMD M HCCC3 HCF PD0 OUT CMD M HCCC3 HCF PD0 IN ACK M HCCC3 HCF PD1 IN ACK M HCCC0 NCF COMMUNICATING M HCAC0 NCF ERROR M HCAC0 NCF HOST COS ACK	M5228 M5237	M5220 M5221 M5222 M5223 M5225 M5226 M5227 M5229 M5230 M5231 M5232 M5233 M5233 M5234 M5235 M5236 M5238
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOSTFlags M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_CMD M_HCCC3_HCF_SEND_MBX_CMD M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_USNETXFLAgS M_HCAC0_NCF_COMMUNICATING M_HCAC0_NCF_ERROR M_HCAC0_NCF_HOST_COS_ACK M_HCAC0_NCF_HOST_COS_ACK M_HCAC0_NCF_NETX_COS_CMD	M5228 M5237	M5220 M5221 M5222 M5223 M5225 M5226 M5227 M5229 M5230 M5231 M5232 M5233 M5233 M5234 M5235 M5236 M5238 M5240 M5241
<pre>#define #define #</pre>	M HCCC3 NCF ERROR M HCCC3 NCF HOST COS ACK M HCCC3 NCF NETX COS CMD M HCCC3 NCF SEND MBX ACK M HCCC3 NCF ZEND MBX ACK M HCCC3 NCF PD0 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 OUT ACK M HCCC3 NCF PD1 IN CMD M HCCC3 NCF PD1 IN CMD M HCCC3 HCF PD1 OUT ACK M HCCC3 HCF PD1 OUT CMD M HCCC3 HCF PD0 OUT CMD M HCCC3 HCF PD0 IN ACK M HCCC3 HCF PD1 IN ACK M HCCC0 NCF COMMUNICATING M HCAC0 NCF ERROR M HCAC0 NCF HOST COS ACK	M5228 M5237	M5220 M5221 M5222 M5223 M5225 M5226 M5227 M5229 M5230 M5231 M5232 M5233 M5233 M5234 M5235 M5236 M5238
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PDO_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOSTFlags M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_CMD M_HCCC3_HCF_SEND_MBX_CMD M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_COMMUNICATING M_HCAC0_NCF_ERROR M_HCAC0_NCF_HOST_COS_ACK M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_SEND_MBX_ACK	M5228 M5237	M5220 M5221 M5222 M5223 M5226 M5227 M5229 M5230 M5231 M5231 M5233 M5233 M5233 M5233 M5233 M5238 M5238
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_COMMUNICATING M_HCAC0_NCF_ERROR M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_RECV_MBX_ACK M_HCAC0_NCF_RECV_MBX_ACK M_HCAC0_NCF_RECV_MBX_ACK	M5228 M5237	M5220 M5221 M5223 M5224 M5226 M5226 M5227 M5230 M5231 M5231 M5233 M5234 M5233 M5234 M5238 M5238 M5238 M5240 M5241 M5242 M5243
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_SEND_MBX_CMD M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_CMMUNICATING M_HCAC0_NCF_ERROR M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_RECV_MBX_CMD M_HCAC0_NCF_RECV_MBX_CMD M_HCAC0_NCF_RECV_MBX_CMD M_HCAC0_NCF_RECV_MBX_CMD M_HCAC0_NCF_RECV_MBX_CMD M_HCAC0_NCF_PD0_OUT_ACK	M5228 M5237	M5220 M5221 M5223 M5224 M5225 M5226 M5227 M5230 M5230 M5231 M5232 M5233 M5234 M5233 M5234 M5238 M5240 M5241 M5242 M5243 M5244
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_ERROR M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_RECV_MBX_CMD M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK	M5228 M5237	M5220 M5221 M5222 M5223 M5225 M5226 M5227 M5230 M5230 M5230 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5240 M5241 M5241 M5242 M5243 M5244 M5243
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_PEND_MBX_ACK M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_PD0_UT_CMD M_HCCC3_HCF_PD0_UT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_ERROR M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_RECV_MBX_ACK M_HCAC0_NCF_RECV_MBX_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD0_IN_CMD	M5228 M5237	M5220 M5221 M5222 M5223 M5225 M5226 M5227 M5230 M5230 M5231 M5232 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5240 M5241 M5241 M5242 M5243 M5244 M5243 M5244 M5245 M5246
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_SEND_MBX_CMD M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_IN_CMD	M5228 M5237	M5220 M5221 M5222 M5223 M5225 M5226 M5227 M5230 M5230 M5230 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5240 M5241 M5241 M5242 M5243 M5244 M5243
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_PEND_MBX_ACK M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_PD0_UT_CMD M_HCCC3_HCF_PD0_UT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_ERROR M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_RECV_MBX_ACK M_HCAC0_NCF_RECV_MBX_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD0_IN_CMD	M5228 M5237	M5220 M5221 M5222 M5223 M5225 M5226 M5227 M5230 M5230 M5231 M5232 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5240 M5241 M5241 M5242 M5243 M5244 M5243 M5244 M5245 M5246
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_PEOV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_SEND_MBX_CMD M_HCCC3_HCF_SEND_MBX_CMD M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_SEND_MBX_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_IN_ACK M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD	M5228 M5237 M5239	M5220 M5221 M5222 M5223 M5225 M5225 M5227 M5230 M5231 M5232 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5233 M5240 M5241 M5242 M5243 M5243 M5243 M5244 M5245 M5246 M5247
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PDO_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOSTFlags M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_CMD M_HCCC3_HCF_SEND_MBX_CMD M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_COMMUNICATING M_HCAC0_NCF_ERROR M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_SEND_MBX_ACK M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD	M5228 M5237 M5239	M5220 M5221 M5222 M5223 M5225 M5226 M5227 M5230 M5231 M5232 M5233 M5234 M5235 M5236 M5236 M5240 M5241 M5242 M5243 M5244 M5243 M5244 M5247 M5249
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_RECV_MBX_ACK M_HCCC3_NCF_PDO_OUT_ACK M_HCCC3_NCF_PDO_IN_CMD M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_USHOSTFlags M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_SEND_MBX_ACK M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_PD0_IN_ACK M_HCCC3_HCF_PD1_IN_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_USNEtxFlags M_HCAC0_NCF_COMMUNICATING M_HCAC0_NCF_ERROR M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD	M5228 M5237 M5239	M5220 M5221 M5222 M5223 M5226 M5227 M5229 M5230 M5231 M5231 M5233 M5234 M5233 M5234 M5238 M5238 M5240 M5241 M5242 M5241 M5242 M5243 M5244 M5243 M5244 M5245 M5247 M5249 M5250
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_ERROR M_HCAC0_NCF_ERROR M_HCAC0_NCF_NETX_COS_ACK M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD1_IN_CMD	M5228 M5237 M5239	M5220 M5221 M5222 M5223 M5226 M5227 M5229 M5230 M5231 M5232 M5233 M5234 M5233 M5234 M5233 M5234 M5238 M5240 M5241 M5242 M5241 M5242 M5243 M5244 M5243 M5244 M5245 M5246 M5247 M5249 M5250 M5251
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_NETX_COS_CMD M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD0_OUT_CMD M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_ERROR M_HCAC0_NCF_ERROR M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_HCF_NETX_COS_CMD M_HCAC0_HCF_NETX_COS_ACK M_HCAC0_HCF_NETX_COS_ACK M_HCAC0_HCF_NETX_COS_ACK	M5228 M5237 M5239	M5220 M5221 M5222 M5223 M5226 M5227 M5229 M5230 M5231 M5232 M5233 M5234 M5233 M5234 M5233 M5234 M5238 M5240 M5241 M5242 M5241 M5242 M5244 M5242 M5243 M5244 M5243 M5244 M5245 M5246 M5247 M5249 M5250 M5251 M5252
<pre>#define #define #</pre>	M_HCCC3_NCF_ERROR M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_HOST_COS_ACK M_HCCC3_NCF_SEND_MBX_ACK M_HCCC3_NCF_RECV_MBX_CMD M_HCCC3_NCF_PD0_OUT_ACK M_HCCC3_NCF_PD1_OUT_ACK M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_NCF_PD1_IN_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_HOST_COS_CMD M_HCCC3_HCF_NETX_COS_ACK M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_RECV_MBX_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_OUT_CMD M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCCC3_HCF_PD1_IN_ACK M_HCAC0_NCF_COMMUNICATING M_HCAC0_NCF_ERROR M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_NETX_COS_CMD M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_OUT_ACK M_HCAC0_NCF_PD0_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_OUT_ACK M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_NCF_PD1_IN_CMD M_HCAC0_HCF_NETX_COS_ACK M_HCAC0_HCF_NETX_COS_ACK M_HCAC0_HCF_NETX_COS_ACK M_HCAC0_HCF_NETX_COS_ACK	M5228 M5237 M5239	M5220 M5221 M5222 M5223 M5226 M5227 M5229 M5230 M5231 M5232 M5233 M5234 M5233 M5234 M5233 M5234 M5238 M5240 M5241 M5242 M5241 M5242 M5243 M5244 M5243 M5244 M5245 M5246 M5247 M5249 M5250 M5251

fine M HCACO HCF PDI IN ACK M5254 fine M HCACO HCF PDI IN ACK M5255 fine M HCACO INCF PDI IN ACK M5257 fine M HCACI NCF COMMUNICATING M5258 fine M HCACI NCF CONTON CONTON M5259 fine M HCACI NCF ERROR M5260 fine M HCACI NCF PERVOS CMD M5261 fine M HCACI NCF PERVORX M5262 fine M HCACI NCF PERVORX M5263 fine M HCACI NCF PDO IT ACK M5264 fine M HCACI NCF PDO IT ACK M5266 fine M HCACI NCF PDO IT ACK M5269 fine M HCACI HCF PDO IT ACK M5269 fine M HCACI HCF PDO IT ACK M5269 fine M HCACI HCF PDO IT ACK M5271 fine M HCACI HCF PDO IT ACK M5271 fine M HCACI HCF PDO IT ACK M5273 fine M HCACI HCF PDO IT ACK M5276 fine M HCACI HCF PDO IT ACK M5276 fine M HCACI HCF PDO IT ACK					
Hine M MCACO MCS26 Mine M MCACI UNEXPIRE MS257 fine M MCACI NCC COMMUNICATING MS258 fine M MCACI NCC COMMUNICATING MS258 fine M MCACI NCP NCS MS260 fine M HCACI NCP <nexx< td=""> COS MS261 fine M HCACI NCP<nexx< td=""> MS263 fine M HCACI NCP<nexx< td=""> MS264 fine M HCACI NCP<pd0< td=""> NT MS266 fine M HCACI NCP<pd1< td=""> NCMD MS267 fine M HCACI NCP<pd1< td=""> NCMD MS267 fine M HCACI NCP<pd1< td=""> NCMD MS270 fine M HCACI HCP<pd1< td=""> NCMD MS271 fine M HCACI HCP<pd0< td=""> NCMD MS273 fine M HCACI HCP<pd0< td=""> NCMD MS274 fine M HCACI HCP<pd0< td=""> NCMD</pd0<></pd0<></pd0<></pd1<></pd1<></pd1<></pd1<></pd0<></nexx<></nexx<></nexx<>	#define	M_HCAC0_HCF_PD0_IN_ACK		M5254	
Hine M MCACO MCS26 Mine M MCACI UNEXPIRE MS257 fine M MCACI NCC COMMUNICATING MS258 fine M MCACI NCC COMMUNICATING MS258 fine M MCACI NCP NCS MS260 fine M HCACI NCP <nexx< td=""> COS MS261 fine M HCACI NCP<nexx< td=""> MS263 fine M HCACI NCP<nexx< td=""> MS264 fine M HCACI NCP<pd0< td=""> NT MS266 fine M HCACI NCP<pd1< td=""> NCMD MS267 fine M HCACI NCP<pd1< td=""> NCMD MS267 fine M HCACI NCP<pd1< td=""> NCMD MS270 fine M HCACI HCP<pd1< td=""> NCMD MS271 fine M HCACI HCP<pd0< td=""> NCMD MS273 fine M HCACI HCP<pd0< td=""> NCMD MS274 fine M HCACI HCP<pd0< td=""> NCMD</pd0<></pd0<></pd0<></pd1<></pd1<></pd1<></pd1<></pd0<></nexx<></nexx<></nexx<>	define	M_HCAC0_HCF_PD1_OUT_CMD		M5255	
fine M_HCAC1_NCF_COMMUNICATING M5259 fine M_HCAC1_NCF_HOST_COS_ACK M5260 fine M_HCAC1_NCF_NETX_COS_CMD M5261 fine M_HCAC1_NCF_RECV_MX_CMD M5263 fine M_HCAC1_NCF_RECV_MX_CMD M5264 fine M_HCAC1_NCF_PDD_UT_ACK M5266 fine M_HCAC1_NCF_PDD_UT_ACK M5266 fine M_HCAC1_NCF_PDT_UN_CMD M5267 fine M_HCAC1_HCF_NETX_COS_CMD M5269 fine M_HCAC1_HCF_NETX_COS_ACK M5270 fine M_HCAC1_HCF_NETX_COS_ACK M5271 fine M_HCAC1_HCF_PDT_OUT_CMD M5273 fine M_HCAC1_HCF_PDD_UT_CMD M5275 fine M_HCAC1_HCF_PDD_OUT_CMD M5276 fine M_HCAC1_HCF_PDD_OUT_CMD M5276 fine M_HCAC1_HCF_PCOS_IDX_N M5276 fine M_CCC1_CX_APP_COS_INIT M5280 fine M_CCC0_RCX_APP_COS_INIT M5281 fine M_CCO_RCX_APP_COS_INIT M5280 fine M_CCO_RCX_APP_COS_DMA_ENABLE M5283 fine M_CCO_RCX_APP_COS_DMA_ENABLE M5281 fine M_CCO_RCX_APP_COS_DMA_ENABLE M5281 fine M_CCO_RCX_APP_COS_DMA_ENABLE M5281	#define	M_HCAC0_HCF_PD1_IN_ACK		M5256	
fine M MCAC1 NCF_HOST_COS_ACK M5260 fine M MCAC1 NCF SEND_MEX_ACK M5261 fine M HCAC1 NCF_RECV_MEX_CMD M5263 fine M HCAC1 NCF_PDO_INT_ACK M5264 fine M HCAC1 NCF_PDO_INT_ACK M5266 fine M HCAC1 NCF_PDI_IN_CMD M5267 fine M HCAC1_USFDI_IN_CMD M5267 fine M HCAC1_HCF_PDOT_CCK M5266 fine M HCAC1_HCF_PDOT_CCK M5267 fine M HCAC1_HCF_PDOT_CCMD M5271 fine M HCAC1_HCF_PDOT_CMD M5273 fine M HCAC1_HCF_PDO_TIN_ACK M5274 fine M HCAC1_HCF_PDO_TIN_ACK M5276 fine M HCAC1_HCF_PDO_TIN_ACK M5277 fine M HCAC1_HCF_PDO_TIN_ACK M5277 fine M HCAC1_HCF_PDO_SON_ENABLE M5281 fine M_CC0_RCX_APP_COS_BUS_ON M5283 M5279 fine M_CC0_RCX_APP_COS_LOCK_CFG M5283 M5283 fine M_CCO_RCX_APP_COS_NUM_ENABLE M5284 M5283 fine M_CCO	#define	M_HCAC1_usNetxFlags	M5257		
fine M HCAC1_NCF_HOST_COS_ACK M5260 fine M HCAC1_NCF_SEND_MEX_ACK M5261 fine M HCAC1_NCF_SEND_MEX_ACK M5263 fine M HCAC1_NCF_PD0_UT_ACK M5264 fine M HCAC1_NCF_PD1_UT_ACK M5266 fine M HCAC1_NCF_PD1_UT_ACK M5266 fine M HCAC1_NCF_PD1_UT_ACK M5266 fine M HCAC1_HCF_PDST_COS_CMD M5267 fine M HCAC1_HCF_PDST_COS_CMD M5267 fine M HCAC1_HCF_PDN_TACK M5270 fine M HCAC1_HCF_PD0_TN_ACK M5271 fine M HCAC1_HCF_PD0_TN_ACK M5274 fine M HCAC1_HCF_PD0_TN_ACK M5276 fine M HCAC1_HCF_PD1_TN_ACK M5276 fine M HCAC1_HCF_PD1_COS_APP_READY M5277 fine M HCAC1_HCF_PCOS_BUS_ON M5280 fine M_CC0_RCX_APP_COS_INIT M5280 M5279 fine M_CC0_RCX_APF_COS_LOCK_CFG M5281 M5283 fine M_CC0_RCX_APF_COS_LOCK_CFG M5281 M5283 fine	#define	M_HCAC1_NCF_COMMUNICATING		M5258	
fine M HCAC1_NCF_HOST_COS_ACK M5260 fine M HCAC1_NCF_SEND_MEX_ACK M5261 fine M HCAC1_NCF_SEND_MEX_ACK M5263 fine M HCAC1_NCF_PD0_UT_ACK M5264 fine M HCAC1_NCF_PD1_UT_ACK M5266 fine M HCAC1_NCF_PD1_UT_ACK M5266 fine M HCAC1_NCF_PD1_UT_ACK M5266 fine M HCAC1_HCF_PDST_COS_CMD M5267 fine M HCAC1_HCF_PDST_COS_CMD M5267 fine M HCAC1_HCF_PDN_TACK M5270 fine M HCAC1_HCF_PD0_TN_ACK M5271 fine M HCAC1_HCF_PD0_TN_ACK M5274 fine M HCAC1_HCF_PD0_TN_ACK M5276 fine M HCAC1_HCF_PD1_TN_ACK M5276 fine M HCAC1_HCF_PD1_COS_APP_READY M5277 fine M HCAC1_HCF_PCOS_BUS_ON M5280 fine M_CC0_RCX_APP_COS_INIT M5280 M5279 fine M_CC0_RCX_APF_COS_LOCK_CFG M5281 M5283 fine M_CC0_RCX_APF_COS_LOCK_CFG M5281 M5283 fine	#define	M_HCAC1_NCF_ERROR	M5259		
HilleMCC0_RCX_APP_COS_INITM5280effineM_CC0_RCX_APP_COS_INIT_ENABLEM5281effineM_CC0_RCX_APP_COS_LOCK_CFGM5282effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_COM_COS_READYM5286effineM_CC0_RCX_COM_COS_READYM5287effineM_CC0_RCX_COM_COS_RUNM5289effineM_CC0_RCX_COM_COS_CONFIG_LOCKEDM5290effineM_CC0_RCX_COM_COS_CONFIG_NEWM5291effineM_CC0_RCX_COM_COS_RESTART_REQM5292effineM_CC0_RCX_COM_COS_RESTART_REQM5293effineM_CC0_RCX_COM_COS_RESTART_REQM5293effineM_CC0_UlCommunicationStateM5296effineM_CC0_ulCommunicationStateM5296effineM_CC0_UlCommunicationStateM5296effineM_CC0_bPDINSourceM5301effineM_CC0_bPDINSourceM5302effineM_CC0_bPDINSourceM5302effineM_CC0_bPDINSourceM5303effineM_CC0_bPDINCNM5304effineM_CC0_bErrorDIORNM5305effineM_CC0_bErrorDIORNM5306effineM_CC0_bErrorPDICNTM5308effineM_CC0_bErrorSyncCntM5308effineM_CC0_bErrorSyncCntM5301effineM_CC0_ulSlaveStateM5311effineM_CC0_ulSlaveErrLogIndM5312	#define	M_HCAC1_NCF_HOST_COS_ACK		M5260	
HilleMCC0_RCX_APP_COS_INITM5280effineM_CC0_RCX_APP_COS_INIT_ENABLEM5281effineM_CC0_RCX_APP_COS_LOCK_CFGM5282effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_COM_COS_READYM5286effineM_CC0_RCX_COM_COS_READYM5287effineM_CC0_RCX_COM_COS_RUNM5289effineM_CC0_RCX_COM_COS_CONFIG_LOCKEDM5290effineM_CC0_RCX_COM_COS_CONFIG_NEWM5291effineM_CC0_RCX_COM_COS_RESTART_REQM5292effineM_CC0_RCX_COMM_COS_RESTART_REQM5293effineM_CC0_RCX_COMM_COS_RESTART_REQM5293effineM_CC0_RCX_COMM_COS_RESTART_REQM5296effineM_CC0_UlCommunicationStateM5296effineM_CC0_UlCommunicationStateM5296effineM_CC0_UlCommunicationStateM5296effineM_CC0_DEDDINSourceM5300effineM_CC0_DEDDINSourceM5301effineM_CC0_DEDDINSourceM5302effineM_CC0_DEDINSourceM5304effineM_CC0_DETrorDOUTCNM5306effineM_CC0_DETrorDOUTCNM5307effineM_CC0_DETrorDOUTCNM5308effineM_CC0_DETrorSyncCntM5308effineM_CC0_DESyncSourceM5311effineM_CC0_UISlaveStateM5311effineM_CC0_UISlaveErrLogINdM5312	#define	M_HCAC1_NCF_NETX_COS_CMD		M5261	
HilleM_CCO_RCX_APP_COS_INITM5280effineM_CCO_RCX_APP_COS_INIT_ENABLEM5281effineM_CCO_RCX_APP_COS_LOCK_CFGM5282effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_OMM_COS_READYM5286effineM_CCO_RCX_COMM_COS_READYM5280effineM_CCO_RCX_COMM_COS_RUNM5289effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290effineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_RESTART_REQM5294effineM_CCO_ULCommunicationStateM5295effineM_CCO_ULCommunicationStateM5296effineM_CCO_USWatchdogTimeM5298effineM_CCO_DEDDINSourceM5300effineM_CCO_DEDDINSourceM5301effineM_CCO_DEDDINSourceM5302effineM_CCO_DEDINSOurceM5303effineM_CCO_DEDINSOurceM5304effineM_CCO_DEDINCOUNTM5306effineM_CCO_DEDINCOUNTM5307effineM_CCO_DEDINCOUNTM5308effineM_CCO_DEDINCOUNTM5308effineM_CCO_DESyncSourceM5310effineM_CCO_DESyncSourceM5311effineM_CCO_UISlaveEstateM5311effineM_CCO_UISlaveErrLogIndM5312	#define	M_HCAC1_NCF_SEND_MBX_ACK		M5262	
HilleM_CCO_RCX_APP_COS_INITM5280effineM_CCO_RCX_APP_COS_INIT_ENABLEM5281effineM_CCO_RCX_APP_COS_LOCK_CFGM5282effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_OMM_COS_READYM5286effineM_CCO_RCX_COMM_COS_READYM5280effineM_CCO_RCX_COMM_COS_RUNM5289effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290effineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_RESTART_REQM5294effineM_CCO_ULCommunicationStateM5295effineM_CCO_ULCommunicationStateM5296effineM_CCO_USWatchdogTimeM5298effineM_CCO_DEDDINSourceM5300effineM_CCO_DEDDINSourceM5301effineM_CCO_DEDDINSourceM5302effineM_CCO_DEDINSOurceM5303effineM_CCO_DEDINSOurceM5304effineM_CCO_DEDINCOUNTM5306effineM_CCO_DEDINCOUNTM5307effineM_CCO_DEDINCOUNTM5308effineM_CCO_DEDINCOUNTM5308effineM_CCO_DESyncSourceM5310effineM_CCO_DESyncSourceM5311effineM_CCO_UISlaveEstateM5311effineM_CCO_UISlaveErrLogIndM5312	#define	M_HCAC1_NCF_RECV_MBX_CMD		M5263	
HilleM_CCO_RCL_APP_COS_INITM5280ffineM_CCO_RCX_APP_COS_INITM5281ffineM_CCO_RCX_APP_COS_LOCK_CFGM5282ffineM_CCO_RCX_APP_COS_DMAM5284ffineM_CCO_RCX_APP_COS_DMA_ENABLEM5285ffineM_CCO_RCX_APP_COS_DMA_ENABLEM5285ffineM_CCO_RCX_COMP_COS_READYM5286ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290ffineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291ffineM_CCO_RCX_COMM_COS_RESTART_REQM5293ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_ULCommunicationStateM5295ffineM_CCO_USVersionM5297ffineM_CCO_USVersionM5297ffineM_CCO_DPDINSourceM5300ffineM_CCO_DPDOUTSourceM5302ffineM_CCO_ULErrorCountM5304ffineM_CCO_DErrorPDINCNTM5306ffineM_CCO_DErrorPDINCNTM5306ffineM_CCO_DErrorSyncCntM5308ffineM_CCO_DSyncSourceM5310ffineM_CCO_ULSlaveStateM5311ffineM_CCO_ULSlaveStateM5311	#define	M HCAC1 NCF PD0 OUT ACK		M5264	
HilleM_CCO_RCL_APP_COS_INITM5280ffineM_CCO_RCX_APP_COS_INITM5281ffineM_CCO_RCX_APP_COS_LOCK_CFGM5282ffineM_CCO_RCX_APP_COS_DMAM5284ffineM_CCO_RCX_APP_COS_DMAM5285ffineM_CCO_RCX_APP_COS_DMAM5286ffineM_CCO_RCX_APP_COS_DMAM5286ffineM_CCO_RCX_COMM_COS_READYM5287ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290ffineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291ffineM_CCO_RCX_COMM_COS_RESTART_REQM5293ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_ULCommunicationStateM5295ffineM_CCO_USVersionM5297ffineM_CCO_USVersionM5296ffineM_CCO_DEPDINSourceM5300ffineM_CCO_DEPDINSourceM5302ffineM_CCO_DEPTORDUTONM5304ffineM_CCO_DEFrorPDINCNTM5306ffineM_CCO_DEFrorPDINCNTM5306ffineM_CCO_DEFrorPDINCNTM5308ffineM_CCO_DEFrorSyncCntM5308ffineM_CCO_DSyncSourceM5310ffineM_CCO_UISlaveStateM5311ffineM_CCO_UISlaveStateM5311	#define	M HCAC1 NCF PD0 IN CMD		M5265	
HilleMCC0_RCL_APP_COS_INITM5280ffineM_CC0_RCX_APP_COS_INITM5281ffineM_CC0_RCX_APP_COS_LOCK_CFGM5282ffineM_CC0_RCX_APP_COS_DMAM5284ffineM_CC0_RCX_APP_COS_DMAM5284ffineM_CC0_RCX_APP_COS_DMA_ENABLEM5285ffineM_CC0_RCX_COMP_COS_READYM5286ffineM_CC0_RCX_COMM_COS_READYM5288ffineM_CC0_RCX_COMM_COS_READYM5289ffineM_CC0_RCX_COMM_COS_CONFIG_LOCKEDM5290ffineM_CC0_RCX_COMM_COS_CONFIG_NEWM5291ffineM_CC0_RCX_COMM_COS_RESTART_REQM5292ffineM_CC0_RCX_COMM_COS_RESTART_REQM5293ffineM_CC0_RCX_COMM_COS_DMAM5294ffineM_CC0_RCX_COMM_COS_DMAM5294ffineM_CC0_ulcommunicationStateM5295ffineM_CC0_ulcommunicationErrorM5296ffineM_CC0_bPDINBskModeM5299ffineM_CC0_bPDUTBSURCEM5300ffineM_CC0_bPDOutHskModeM5301ffineM_CC0_bPDOutSourceM5303ffineM_CC0_bErrorPDInCntM5306ffineM_CC0_bErrorPDInCntM5306ffineM_CC0_bErrorPDInCntM5307ffineM_CC0_bErrorSyncCntM5308ffineM_CC0_bSyncSourceM5310ffineM_CC0_ulSlaveStateM5311ffineM_CC0_ulSlaveErtLogIndM5312	#define	M HCAC1 NCF PD1 OUT ACK		M5266	
HilleM_CCO_RCL_APP_COS_INITM5280ffineM_CCO_RCX_APP_COS_INITM5281ffineM_CCO_RCX_APP_COS_LOCK_CFGM5282ffineM_CCO_RCX_APP_COS_DMAM5284ffineM_CCO_RCX_APP_COS_DMAM5285ffineM_CCO_RCX_APP_COS_DMAM5286ffineM_CCO_RCX_APP_COS_DMAM5286ffineM_CCO_RCX_COMM_COS_READYM5287ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290ffineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291ffineM_CCO_RCX_COMM_COS_RESTART_REQM5293ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_ULCommunicationStateM5295ffineM_CCO_USVersionM5297ffineM_CCO_USVersionM5296ffineM_CCO_DEPDINSourceM5300ffineM_CCO_DEPDINSourceM5302ffineM_CCO_DEPTORDUTONM5304ffineM_CCO_DEFrorPDINCNTM5306ffineM_CCO_DEFrorPDINCNTM5306ffineM_CCO_DEFrorPDINCNTM5308ffineM_CCO_DEFrorSyncCntM5308ffineM_CCO_DSyncSourceM5310ffineM_CCO_UISlaveStateM5311ffineM_CCO_UISlaveStateM5311	#define	M_HCAC1_NCF_PD1_IN_CMD		M5267	
HilleM_CCO_RCL_APP_COS_INITM5280ffineM_CCO_RCX_APP_COS_INIT_ENABLEM5281ffineM_CCO_RCX_APP_COS_LOCK_CFGM5282ffineM_CCO_RCX_APP_COS_DMAM5284ffineM_CCO_RCX_APP_COS_DMA_ENABLEM5285ffineM_CCO_RCX_APP_COS_DMA_ENABLEM5285ffineM_CCO_RCX_COMP_COS_READYM5286ffineM_CCO_RCX_COMM_COS_READYM5287ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290ffineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290ffineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291ffineM_CCO_RCX_COMM_COS_RESTART_REQM5293ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_ULCommunicationStateM5295ffineM_CCO_USVersionM5297ffineM_CCO_USVersionM5297ffineM_CCO_DPDINSourceM5300ffineM_CCO_DPDUTSourceM5302ffineM_CCO_DPDUTSourceM5302ffineM_CCO_ULErrorCountM5304ffineM_CCO_DErrorPDINCNTM5306ffineM_CCO_DErrorPDINCNTM5307ffineM_CCO_DErrorSyncCntM5308ffineM_CCO_DSyncSourceM5310ffineM_CCO_ULSlaveStateM5311ffineM_CCO_ULSlaveStateM5311	#define	M_HCAC1_usHostFlags	M5268		
HilleM_CCO_RCX_APP_COS_INITM5280effineM_CCO_RCX_APP_COS_INIT_ENABLEM5281effineM_CCO_RCX_APP_COS_LOCK_CFGM5282effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_OMM_COS_READYM5286effineM_CCO_RCX_COMM_COS_READYM5287effineM_CCO_RCX_COMM_COS_READYM5289effineM_CCO_RCX_COMM_COS_RONM5289effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290effineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_RESTART_REQM5294effineM_CCO_ULCommunicationStateM5296effineM_CCO_ULCommunicationStateM5296effineM_CCO_ULCommunicationStateM5296effineM_CCO_DEDDINSourceM5300effineM_CCO_DEDDINSourceM5301effineM_CCO_DEDDINSourceM5302effineM_CCO_DEDINSourceM5304effineM_CCO_DEDINCOUNTM5304effineM_CCO_DEDINCOUNTM5306effineM_CCO_DEDINCOUNTM5307effineM_CCO_DEDINCOUNTM5307effineM_CCO_DESYNCCNTM5308effineM_CCO_DESYNCCNTM5308effineM_CCO_DESyncSourceM5311effineM_CCO_UISlaveEstateM5311effine	#define	M_HCAC1_HCF_HOST_COS_CMD		M5269	
HilleMCC0_RCX_APP_COS_INITM5280effineM_CC0_RCX_APP_COS_INIT_ENABLEM5281effineM_CC0_RCX_APP_COS_LOCK_CFGM5282effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_COM_COS_READYM5286effineM_CC0_RCX_COM_COS_READYM5287effineM_CC0_RCX_COM_COS_RUNM5289effineM_CC0_RCX_COM_COS_CONFIG_LOCKEDM5290effineM_CC0_RCX_COM_COS_CONFIG_NEWM5291effineM_CC0_RCX_COM_COS_RESTART_REQM5292effineM_CC0_RCX_COM_COS_RESTART_REQM5293effineM_CC0_RCX_COM_COS_RESTART_REQM5293effineM_CC0_UlCommunicationStateM5296effineM_CC0_ulCommunicationStateM5296effineM_CC0_UlCommunicationStateM5296effineM_CC0_bPDINSourceM5301effineM_CC0_bPDINSourceM5302effineM_CC0_bPDINSourceM5302effineM_CC0_bPDINSourceM5303effineM_CC0_bPDINCNM5304effineM_CC0_bErrorDIORNM5305effineM_CC0_bErrorDIORNM5306effineM_CC0_bErrorPDICNTM5308effineM_CC0_bErrorSyncCntM5308effineM_CC0_bErrorSyncCntM5301effineM_CC0_ulSlaveStateM5311effineM_CC0_ulSlaveErrLogIndM5312	#define	M HCAC1 HCF NETX COS ACK		M5270	
HilleMCCCU_RCX_APP_COS_INITM5280effineM_CCO_RCX_APP_COS_INIT_ENABLEM5281effineM_CCO_RCX_APP_COS_LOCK_CFGM5282effineM_CCO_RCX_APP_COS_DMA_ENABLEM5283effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_OMM_COS_READYM5286effineM_CCO_RCX_COMM_COS_READYM5287effineM_CCO_RCX_COMM_COS_READYM5289effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5291effineM_CCO_RCX_COMM_COS_RESTART_REQM5292effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_ULCommunicationStateM5296effineM_CCO_ULCommunicationStateM5296effineM_CCO_ULCommunicationStateM5296effineM_CCO_DEDDINSourceM5300effineM_CCO_DEDDINSourceM5301effineM_CCO_ULBERTORCOUNTM5304effineM_CCO_DEDFUNCTM5304effineM_CCO_DETROPDUTCNTM5306effineM_CCO_DETROPDUTCNTM5306effineM_CCO_DETROPDUTCNTM5308effineM_CCO_DETROPSyncCntM5308effineM_CCO_DESyncSourceM5311effineM_CCO_ULSlaveEstateM5311effineM_CCO_ULSlaveErrLogINdM5312	#define	M HCAC1 HCF SEND MBX CMD		M5271	
HilleMCCU_RCX_APP_COS_INITMS280effineM_CCO_RCX_APP_COS_INIT_ENABLEM5281effineM_CCO_RCX_APP_COS_LOCK_CFGM5282effineM_CCO_RCX_APP_COS_DMAM5284effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_APP_COS_DMA_ENABLEM5286effineM_CCO_RCX_COMM_COS_READYM5287effineM_CCO_RCX_COMM_COS_RUNM5288effineM_CCO_RCX_COMM_COS_RUNM5289effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5291effineM_CCO_RCX_COMM_COS_RESTART_REQM5292effineM_CCO_RCX_COMM_COS_REQ_ENAM5293effineM_CCO_RCX_COMM_COS_DMAM5294effineM_CCO_ulCommunicationStateM5296effineM_CCO_ulCommunicationErrorM5296effineM_CCO_ulCommunicationErrorM5296effineM_CCO_bPDINSourceM5300effineM_CCO_bPDINSourceM5301effineM_CCO_bPDINSourceM5302effineM_CCO_bPDINCAM5304effineM_CCO_bErrorDIGINdM5305effineM_CCO_bErrorDICATM5306effineM_CCO_bErrorSyncCntM5308effineM_CCO_bSyncSourceM5301effineM_CCO_ulSlaveErrLogINdM5311	#define	M_HCAC1_HCF_RECV_MBX_ACK		M5272	
HilleMCCCU_RCX_APP_COS_INITM5280effineM_CCO_RCX_APP_COS_INIT_ENABLEM5281effineM_CCO_RCX_APP_COS_LOCK_CFGM5282effineM_CCO_RCX_APP_COS_DMA_ENABLEM5283effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_OMM_COS_READYM5286effineM_CCO_RCX_COMM_COS_READYM5287effineM_CCO_RCX_COMM_COS_RUNM5288effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5291effineM_CCO_RCX_COMM_COS_RESTART_REQM5292effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_REQ_ENAM5293effineM_CCO_ULCommunicationStateM5296effineM_CCO_ULCommunicationStateM5296effineM_CCO_USWatchdogTimeM5298effineM_CCO_DEDDINSourceM5300effineM_CCO_DEDDINSourceM5301effineM_CCO_ULFrortogIndM5304effineM_CCO_DEFrortogIndM5305effineM_CCO_DEFrortogIndM5306effineM_CCO_DEFrortogIndM5307effineM_CCO_DEFrortogIndM5308effineM_CCO_DEFrortogIntM5308effineM_CCO_DEFrortogIntM5308effineM_CCO_DEFrortogIntM5308effineM_CCO_DEFrortogIntM5309effineM_CCO_DEFrortogIntM5308effineM_CCO_DESyncSourceM5311effine <td>#define</td> <td>M_HCAC1_HCF_PD0_OUT_CMD</td> <td></td> <td>M5273</td> <td></td>	#define	M_HCAC1_HCF_PD0_OUT_CMD		M5273	
HilleMCCCU_RCX_APP_COS_INITM5280effineM_CCO_RCX_APP_COS_INIT_ENABLEM5281effineM_CCO_RCX_APP_COS_LOCK_CFGM5282effineM_CCO_RCX_APP_COS_DMA_ENABLEM5283effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_OMM_COS_READYM5286effineM_CCO_RCX_COMM_COS_READYM5287effineM_CCO_RCX_COMM_COS_READYM5289effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5291effineM_CCO_RCX_COMM_COS_RESTART_REQM5292effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_ULCommunicationStateM5296effineM_CCO_ULCommunicationStateM5296effineM_CCO_ULCommunicationStateM5296effineM_CCO_DEDDINSourceM5300effineM_CCO_DEDDINSourceM5301effineM_CCO_ULBERTORCOUNTM5304effineM_CCO_DEDFUNCTM5304effineM_CCO_DETROPDUTCNTM5306effineM_CCO_DETROPDUTCNTM5306effineM_CCO_DETROPDUTCNTM5308effineM_CCO_DETROPSyncCntM5308effineM_CCO_DESyncSourceM5311effineM_CCO_ULSlaveEstateM5311effineM_CCO_ULSlaveErrLogINdM5312	#define	M_HCAC1_HCF_PD0_IN_ACK		M5274	
HilleMCCCU_RCX_APP_COS_INITM5280effineM_CCO_RCX_APP_COS_INIT_ENABLEM5281effineM_CCO_RCX_APP_COS_LOCK_CFGM5282effineM_CCO_RCX_APP_COS_DMA_ENABLEM5283effineM_CCO_RCX_APP_COS_DMA_ENABLEM5285effineM_CCO_RCX_OMM_COS_READYM5286effineM_CCO_RCX_COMM_COS_READYM5287effineM_CCO_RCX_COMM_COS_RUNM5288effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290effineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5291effineM_CCO_RCX_COMM_COS_RESTART_REQM5292effineM_CCO_RCX_COMM_COS_RESTART_REQM5293effineM_CCO_RCX_COMM_COS_REQ_ENAM5293effineM_CCO_ULCommunicationStateM5296effineM_CCO_ULCommunicationStateM5296effineM_CCO_USWatchdogTimeM5298effineM_CCO_DEDDINSourceM5300effineM_CCO_DEDDINSourceM5301effineM_CCO_ULFrortogIndM5304effineM_CCO_DEFrortogIndM5305effineM_CCO_DEFrortogIndM5306effineM_CCO_DEFrortogIndM5307effineM_CCO_DEFrortogIndM5308effineM_CCO_DEFrortogIntM5308effineM_CCO_DEFrortogIntM5308effineM_CCO_DEFrortogIntM5308effineM_CCO_DEFrortogIntM5309effineM_CCO_DEFrortogIntM5308effineM_CCO_DESyncSourceM5311effine <td>#define</td> <td>M_HCAC1_HCF_PD1_OUT_CMD</td> <td></td> <td>M5275</td> <td></td>	#define	M_HCAC1_HCF_PD1_OUT_CMD		M5275	
HilleMCC0_RCX_APP_COS_INITM5280effineM_CC0_RCX_APP_COS_INIT_ENABLEM5281effineM_CC0_RCX_APP_COS_LOCK_CFGM5282effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_COM_COS_READYM5286effineM_CC0_RCX_COM_COS_READYM5287effineM_CC0_RCX_COM_COS_RUNM5289effineM_CC0_RCX_COM_COS_CONFIG_LOCKEDM5290effineM_CC0_RCX_COM_COS_CONFIG_NEWM5291effineM_CC0_RCX_COM_COS_RESTART_REQM5292effineM_CC0_RCX_COMM_COS_RESTART_REQM5293effineM_CC0_RCX_COMM_COS_RESTART_REQM5293effineM_CC0_RCX_COMM_COS_RESTART_REQM5296effineM_CC0_UlCommunicationStateM5296effineM_CC0_UlCommunicationStateM5296effineM_CC0_UlCommunicationStateM5296effineM_CC0_DEDDINSourceM5300effineM_CC0_DEDDINSourceM5301effineM_CC0_DEDDINSourceM5302effineM_CC0_DEDINSourceM5304effineM_CC0_DETrorDOUTCNM5306effineM_CC0_DETrorDOUTCNM5307effineM_CC0_DETrorDOUTCNM5308effineM_CC0_DETrorSyncCntM5308effineM_CC0_DESyncSourceM5311effineM_CC0_UISlaveStateM5311effineM_CC0_UISlaveErrLogINdM5312	#define	M HCAC1 HCF PD1 IN ACK		M5276	
HilleMCC0_RCX_APP_COS_INITM5280effineM_CC0_RCX_APP_COS_INIT_ENABLEM5281effineM_CC0_RCX_APP_COS_LOCK_CFGM5282effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_APP_COS_DMA_ENABLEM5285effineM_CC0_RCX_COM_COS_READYM5286effineM_CC0_RCX_COM_COS_READYM5287effineM_CC0_RCX_COM_COS_RUNM5289effineM_CC0_RCX_COM_COS_CONFIG_LOCKEDM5290effineM_CC0_RCX_COM_COS_CONFIG_NEWM5291effineM_CC0_RCX_COM_COS_RESTART_REQM5292effineM_CC0_RCX_COM_COS_RESTART_REQM5293effineM_CC0_RCX_COM_COS_RESTART_REQM5293effineM_CC0_UlCommunicationStateM5296effineM_CC0_ulCommunicationStateM5296effineM_CC0_UlCommunicationStateM5296effineM_CC0_bPDINSourceM5300effineM_CC0_bPDINSourceM5301effineM_CC0_bPDINSourceM5302effineM_CC0_bPDINSourceM5303effineM_CC0_bPDINTSWAtchdogM5303effineM_CC0_bErrorDountM5304effineM_CC0_bErrorDINCNM5306effineM_CC0_bErrorPDINCNM5307effineM_CC0_bErrorSyncCntM5308effineM_CC0_bErrorSyncCntM5308effineM_CC0_bSyncSourceM5311effineM_CC0_ulSlaveErrLogIndM5312	#define	M CCO RCX APP COS APP READY		M5277	
HilleM_CCO_RCL_APP_COS_INITM5280ffineM_CCO_RCX_APP_COS_INIT_ENABLEM5281ffineM_CCO_RCX_APP_COS_LOCK_CFGM5282ffineM_CCO_RCX_APP_COS_DMAM5284ffineM_CCO_RCX_APP_COS_DMA_ENABLEM5285ffineM_CCO_RCX_APP_COS_DMA_ENABLEM5285ffineM_CCO_RCX_COMP_COS_READYM5286ffineM_CCO_RCX_COMM_COS_READYM5287ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290ffineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290ffineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291ffineM_CCO_RCX_COMM_COS_RESTART_REQM5293ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_ULCommunicationStateM5295ffineM_CCO_USVersionM5297ffineM_CCO_USVersionM5297ffineM_CCO_DPDINSourceM5300ffineM_CCO_DPDUTSourceM5302ffineM_CCO_DPDUTSourceM5302ffineM_CCO_ULErrorCountM5304ffineM_CCO_DErrorPDINCNTM5306ffineM_CCO_DErrorPDINCNTM5307ffineM_CCO_DErrorSyncCntM5308ffineM_CCO_DSyncSourceM5310ffineM_CCO_ULSlaveStateM5311ffineM_CCO_ULSlaveStateM5311	#define	M CCO RCX APP COS BUS ON		M5278	
HilleM_CCO_RCL_APP_COS_INITM5280ffineM_CCO_RCX_APP_COS_INITM5281ffineM_CCO_RCX_APP_COS_LOCK_CFGM5282ffineM_CCO_RCX_APP_COS_DMAM5284ffineM_CCO_RCX_APP_COS_DMAM5285ffineM_CCO_RCX_APP_COS_DMAM5286ffineM_CCO_RCX_APP_COS_DMAM5286ffineM_CCO_RCX_COMM_COS_READYM5287ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_READYM5289ffineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290ffineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291ffineM_CCO_RCX_COMM_COS_RESTART_REQM5293ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_RCX_COMM_COS_DMAM5294ffineM_CCO_ULCommunicationStateM5295ffineM_CCO_USVersionM5297ffineM_CCO_USVersionM5296ffineM_CCO_DEPDINSourceM5300ffineM_CCO_DEPDINSourceM5302ffineM_CCO_DEPTORDUTONM5304ffineM_CCO_DEFrorPDINCNTM5306ffineM_CCO_DEFrorPDINCNTM5306ffineM_CCO_DEFrorPDINCNTM5308ffineM_CCO_DEFrorSyncCntM5308ffineM_CCO_DSyncSourceM5310ffineM_CCO_UISlaveStateM5311ffineM_CCO_UISlaveStateM5311	#define	M_CC0_RCX_APP_COS_BUS_ON_ENAL	BLE		M5279
fine M_CCO_RCX_APP_COS_INIT_ENABLE M5281 fine M_CCO_RCX_APP_COS_LOCK_CFG M5282 fine M_CCO_RCX_APP_COS_DMA M5284 fine M_CCO_RCX_APP_COS_DMA M5284 fine M_CCO_RCX_CAPP_COS_DMA_ENABLE M5285 fine M_CCO_RCX_COM_COS_READY M5287 fine M_CCO_RCX_COM_COS_READY M5288 fine M_CCO_RCX_COM_COS_READY M5289 fine M_CCO_RCX_COM_COS_CONFIG_LOCKED M5290 fine M_CCO_RCX_COM_COS_CONFIG_NEW M5291 fine M_CCO_RCX_COM_COS_CONFIG_NEW M5291 fine M_CCO_RCX_COM_COS_RESTART_REQ M5293 fine M_CCO_RCX_COM_COS_DMA M5293 fine M_CCO_RCX_COM_COS_DMA M5293 fine M_CCO_RCX_COM_COS_DMA M5293 fine M_CCO_ULCommunicationState M5295 fine M_CCO_ULCommunicationError M5296 fine M_CCO_ULCommunicationError M5296 fine M_CCO_DPDINHSKMOde M5299 fine M_CCO_bPDINHSKMOde M5209 fine M_CCO_bPDINHSKMOde M5301 fine M_CCO_ULBERORCE M5302 fine M_CCO_ULBERORCE M5302 fine M_CCO_DPDOUTHSKMODE M5303 fine M_CCO_DEFTORLOGIND M5304 fine M_CCO_DEFTORLOGIND M5305 fine M_CCO_DETTORLOGIND M5307 fine M_CCO_DETTORLOGIND M5307 fine M_CCO_DETTORLOGIND M5307 fine M_CCO_DETTORLOGIND M5307 fine M_CCO_DETTORLOGIND M5307 fine M_CCO_DETTORLOGIND M5307 fine M_CCO_DETTORLOGIND M5309 fine M_CCO_DETTORLOGIND M5309 fine M_CCO_DETTORLOGIND M5301 fine M_CCO_DETTORSYNCCT M5308 fine M_CCO_DETTORSYNCCT M5308 fine M_CCO_DETTORSYNCCT M5308 fine M_CCO_DETTORSYNCCT M5308 fine M_CCO_DETTORSYNCCT M5308 fine M_CCO_DETTORSYNCCT M5308 fine M_CCO_DESTORSYNCCT M5308 fine M_CCO_USANCTOSYNCCT M5308 fine M_CCO_USANCTOSYNCC	#deline	M_CCU_RCA_APP_COS_INIT		M3280	
Herine M_CCO_RCX_APP_COS_LOCK_CFG_ENAM5283Affine M_CCO_RCX_APP_COS_DMAM5284Affine M_CCO_RCX_APP_COS_DMA_ENABLEM5285Affine M_CCO_UlDeviceWatchdogM5286Affine M_CCO_RCX_COMM_COS_READYM5287Affine M_CCO_RCX_COMM_COS_RUNM5288Affine M_CCO_RCX_COMM_COS_BUS_ONM5289Affine M_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290Affine M_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5291Affine M_CCO_RCX_COMM_COS_RESTART_REQM5293Affine M_CCO_RCX_COMM_COS_DMAM5294Affine M_CCO_ULCommunicationStateM5295Affine M_CCO_USVersionM5297Affine M_CCO_bPDINB&URCEM5300Affine M_CCO_bPDINB&URCEM5301Affine M_CCO_UPDINB&URCEM5302Affine M_CCO_ULBERTARTM5304Affine M_CCO_DEDUTHSKMOdeM5303Affine M_CCO_DEDUTSOURCEM5303Affine M_CCO_DEDUTSOURCEM5304Affine M_CCO_DETORDUTONM5304Affine M_CCO_DETORDUTONM5306Affine M_CCO_bErrorDINCATM5307Affine M_CCO_bErrorSyncCntM5308Affine M_CCO_bSyncBsURGEM5310Affine M_CCO_USAURCEM5310Affine M_CCO_USAURCEM5311	#define	M CCO RCX APP COS INIT ENABL	F.	M5281	
Herine M_CCO_RCX_APP_COS_LOCK_CFG_ENAM5283Affine M_CCO_RCX_APP_COS_DMAM5284Affine M_CCO_RCX_COPPCOS_DMA_ENABLEM5285Affine M_CCO_UlDeviceWatchdogM5286Affine M_CCO_RCX_COMM_COS_READYM5287Affine M_CCO_RCX_COMM_COS_RUNM5288Affine M_CCO_RCX_COMM_COS_BUS_ONM5289Affine M_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290Affine M_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5292Affine M_CCO_RCX_COMM_COS_RESTART_REQM5293Affine M_CCO_RCX_COMM_COS_DMAM5294Affine M_CCO_RCX_COMM_COS_DMAM5294Affine M_CCO_UlCommunicationStateM5295Affine M_CCO_usWersionM5297Affine M_CCO_bPDIntskModeM5299Affine M_CCO_bPDIntskModeM5301Affine M_CCO_UlHostWatchdogTimeM5302Affine M_CCO_ULErrorCountM5304Affine M_CCO_bErrorDInCntM5306Affine M_CCO_bErrorDInCntM5307Affine M_CCO_bSyncBikModeM5309Affine M_CCO_bSyncSourceM5309Affine M_CCO_bSyncSourceM5310Affine M_CCO_ulSlaveStateM5312	#define	M_CC0_RCX_APP_COS_LOCK_CFG		M5282	
fineM_CC0_RCX_APP_COS_DMA_ENABLEM5285fineM_CC0_ulDeviceWatchdogM5286fineM_CC0_RCX_COMM_COS_READYM5287fineM_CC0_RCX_COMM_COS_BUS_ONM5289fineM_CC0_RCX_COMM_COS_CONFIG_LOCKEDM5290fineM_CC0_RCX_COMM_COS_CONFIG_NEWM5291fineM_CC0_RCX_COMM_COS_RESTART_REQM5293fineM_CC0_RCX_COMM_COS_REQ_ENAM5294fineM_CC0_ulCommunicationStateM5296fineM_CC0_ulCommunicationErrorM5296fineM_CC0_usWatchdogTimeM5297fineM_CC0_bPDInHskModeM5293fineM_CC0_bPDoutHskModeM5301fineM_CC0_ullerrorCountM5303fineM_CC0_bPDoutHskModeM5305fineM_CC0_berrorPDoutCntM5306fineM_CC0_berrorSyncCntM5308fineM_CC0_berrorSyncCntM5308fineM_CC0_bsyncHskModeM5311fineM_CC0_ulSlaveErrLogIndM5312	#deiine	M CCU RCX APP COS LOCK CFG EI	NA		M5283
fineM_CCO_RCX_COMM_COS_READYM5287fineM_CCO_RCX_COMM_COS_RUNM5288efineM_CCO_RCX_COMM_COS_BUS_ONM5289efineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290efineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291efineM_CCO_RCX_COMM_COS_RESTART_REQM5293efineM_CCO_RCX_COMM_COS_BDAAM5293efineM_CCO_ULCOMMUNICATIONSTATEM5295efineM_CCO_ULCOMMUNICATIONETROTM5296efineM_CCO_ULCOMMUNICATIONETROTM5296efineM_CCO_USVersionM5297efineM_CCO_DEPDINHSKMOdeM5299efineM_CCO_DEPDINSOURCEM5300efineM_CCO_DEPDINSOURCEM5301efineM_CCO_ULBERTORCOUNTM5304efineM_CCO_ULBERTORCOUNTM5306efineM_CCO_DEFTORDUTCNTM5307efineM_CCO_DEFTORDUTCNTM5308efineM_CCO_DEFTORSYNCCNTM5308efineM_CCO_DSYNCSOURCEM5310efineM_CCO_ULSLaveErrLogINDM5312	#define	M_CC0_RCX_APP_COS_DMA	M5284		
fineM_CCO_RCX_COMM_COS_READYM5287fineM_CCO_RCX_COMM_COS_RUNM5288efineM_CCO_RCX_COMM_COS_BUS_ONM5289efineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290efineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291efineM_CCO_RCX_COMM_COS_RESTART_REQM5293efineM_CCO_RCX_COMM_COS_BDAAM5293efineM_CCO_ULCOMMUNICATIONSTATEM5295efineM_CCO_ULCOMMUNICATIONETROTM5296efineM_CCO_ULCOMMUNICATIONETROTM5296efineM_CCO_USVersionM5297efineM_CCO_DEPDINHSKMOdeM5299efineM_CCO_DEPDINSOURCEM5300efineM_CCO_DEPDINSOURCEM5301efineM_CCO_ULBERTORCOUNTM5304efineM_CCO_ULBERTORCOUNTM5306efineM_CCO_DEFTORDUTCNTM5307efineM_CCO_DEFTORDUTCNTM5308efineM_CCO_DEFTORSYNCCNTM5308efineM_CCO_DSYNCSOURCEM5310efineM_CCO_ULSLaveErrLogINDM5312	#define	M_CC0_RCX_APP_COS_DMA_ENABLE		M5285	
fineM_CCO_RCX_COMM_COS_READYM5287fineM_CCO_RCX_COMM_COS_RUNM5288efineM_CCO_RCX_COMM_COS_BUS_ONM5289efineM_CCO_RCX_COMM_COS_CONFIG_LOCKEDM5290efineM_CCO_RCX_COMM_COS_CONFIG_NEWM5291efineM_CCO_RCX_COMM_COS_RESTART_REQM5293efineM_CCO_RCX_COMM_COS_BDAAM5293efineM_CCO_ULCOMMUNICATIONSTATEM5295efineM_CCO_ULCOMMUNICATIONETROTM5296efineM_CCO_ULCOMMUNICATIONETROTM5296efineM_CCO_USVersionM5297efineM_CCO_DEPDINHSKMOdeM5299efineM_CCO_DEPDINSOURCEM5300efineM_CCO_DEPDINSOURCEM5301efineM_CCO_ULBERTORCOUNTM5304efineM_CCO_ULBERTORCOUNTM5306efineM_CCO_DEFTORDUTCNTM5307efineM_CCO_DEFTORDUTCNTM5308efineM_CCO_DEFTORSYNCCNTM5308efineM_CCO_DSYNCSOURCEM5310efineM_CCO_ULSLaveErrLogINDM5312	#define	M_CC0_ulDeviceWatchdog		M5286	
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_RCX_COMM_COS_READY		M5287	
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_RCX_COMM_COS_RUN		M5288	
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_RCX_COMM_COS_BUS_ON		M5289	
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_RCX_COMM_COS_CONFIG_LO	CKED		M5290
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_RCX_COMM_COS_CONFIG_NET	N	M5291	
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_RCX_COMM_COS_RESTART_RI	EQ		M5292
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_RCX_COMM_CO_REQ_ENA		M5293	
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_RCX_COMM_COS_DMA		M5294	
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_ulCommunicationState		M5295	
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_ulCommunicationError		M5296	
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CC0_usVersion	M5297		
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorDgIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5307efineM_CC0_bErrorSyncCntM5308efineM_CC0_bSyncHskModeM5309efineM_CC0_ulSlaveStateM5311efineM_CC0_ulSlaveErrLogIndM5312	#define	M_CCU_usWatchdogTime	M5298		
fineM_CC0_ulErrorCountM5304fineM_CC0_bErrorLogIndM5305fineM_CC0_bErrorPDInCntM5306fineM_CC0_bErrorSyncCntM5308fineM_CC0_bSyncHskModeM5309fineM_CC0_ulSlaveStateM5311fineM_CC0_ulSlaveErrLogIndM5312	#define	M_CCU_bPDInHskMode	M5299		
fineM_CCO_ulErrorCountM5304fineM_CCO_bErrorLogIndM5305fineM_CCO_bErrorPDInCntM5306fineM_CCO_bErrorSyncCntM5307efineM_CCO_bErrorSyncCntM5308efineM_CCO_bSyncHskModeM5309efineM_CCO_ulSlaveStateM5311efineM_CCO_ulSlaveErrLogIndM5312	#define	M_CCU_bPDInSource	M5300		
fineM_CCO_ulErrorCountM5304fineM_CCO_bErrorLogIndM5305fineM_CCO_bErrorPDInCntM5306fineM_CCO_bErrorSyncCntM5307efineM_CCO_bErrorSyncCntM5308efineM_CCO_bSyncHskModeM5309efineM_CCO_ulSlaveStateM5311efineM_CCO_ulSlaveErrLogIndM5312	#define	M_CC0_bPDOutHskMode	M5301		
fineM_CCO_ulErrorCountM5304fineM_CCO_bErrorLogIndM5305fineM_CCO_bErrorPDInCntM5306fineM_CCO_bErrorSyncCntM5307efineM_CCO_bErrorSyncCntM5308efineM_CCO_bSyncHskModeM5309efineM_CCO_ulSlaveStateM5311efineM_CCO_ulSlaveErrLogIndM5312	#define	M_CCU_bPDOutSource	M5302		
fineM_CCO_bErrorLogIndM5305efineM_CCO_bErrorPDInCntM5306efineM_CCO_bErrorPDOutCntM5307efineM_CCO_bErrorSyncCntM5308efineM_CCO_bSyncHskModeM5309efineM_CCO_bSyncSourceM5310efineM_CCO_ulSlaveStateM5311efineM_CCO_ulSlaveErrLogIndM5312					
fineM_CCO_bErrorPDInCntM5306efineM_CCO_bErrorPDOutCntM5307efineM_CCO_bErrorSyncCntM5308efineM_CCO_bSyncHskModeM5309efineM_CCO_bSyncSourceM5310efineM_CCO_ulSlaveStateM5311efineM_CCO_ulSlaveErrLogIndM5312					
fineM_CCO_bErrorPDOutCntM5307efineM_CCO_bErrorSyncCntM5308efineM_CCO_bSyncHskModeM5309efineM_CCO_bSyncSourceM5310efineM_CCO_ulSlaveStateM5311efineM_CCO_ulSlaveErrLogIndM5312					
fineM_CCO_bErrorSyncCntM5308fineM_CCO_bSyncHskModeM5309fineM_CCO_bSyncSourceM5310fineM_CCO_ulSlaveStateM5311fineM_CCO_ulSlaveErrLogIndM5312					
efine M_CCO_bSyncHskMode M5309 efine M_CCO_bSyncSource M5310 efine M_CCO_ulSlaveState M5311 efine M_CCO_ulSlaveErrLogInd M5312					
efine M_CCO_bSyncSource M5310 Efine M_CCO_ulSlaveState M5311 Efine M_CCO_ulSlaveErrLogInd M5312					
fine M_CCO_ulSlaveState M5311 fine M_CCO_ulSlaveErrLogInd M5312					
fine M_CC0_ulSlaveErrLogInd M5312					
			M5311		
		M_CCO_ulNumOfConfigSlaves		M5313	
fine M_CCO_ulNumOfActiveSlaves M5314					
fine M_CC0_ulNumOfDiagSlaves M5315	#define	M_CCU_ulNumOfDiagSlaves		M5315	

Next file:

// M-VariableDefinition_\$6C000.pmc CLOSE END GAT DEL GAT #Include "MacroNameDefinition \$6C000.h" M SI abCookie 0 ->Y:\$6C000,0,8 M_SI_abCookie_1_->Y:\$6C000,8,8
M_SI_abCookie_2_->X:\$6C000,0,8
M_SI_abCookie_3_->X:\$6C000,8,8 M SI ulDpmTotalSize->DP:\$6C001 M_SI_ulDeviceNumber->DP:\$6C002 M SI ulSerialNumber->DP:\$6C003 M_SI_ausHwOptions_0_->Y:\$6C004,0,16 M_SI_ausHwOptions 1 ->X:\$6C004,0,16
M_SI_ausHwOptions 2 ->Y:\$6C005,0,16
M_SI_ausHwOptions 3 ->X:\$6C005,0,16 M SI usManufacturer->Y:\$6C006,0,16 M_SI_usProductionDate->X:\$6C006,0,16 M SI ulLicenseFlags1->DP:\$6C007 M SI ulLicenseFlags2->DP:\$6C008 M SI usNetxLicenseID->Y:\$6C009,0,16 M SI usNetxLicenseFlags->X:\$6C009,0,16 M SI usDeviceClass->Y:\$6C00A,0,16 M SI bHwRevision->X:\$6C00A,0,8 M SI bHwCompatibility->X:\$6C00A,8,8 M SI bDevIdNumber->Y:\$6C00B,0,8 M SCI bChannelType->Y:\$6C00C,0,8 M SCI bSizePositionOfHandshake->X:\$6C00C,0,8 M SCI bNumberOfBlocks->X:\$6C00C,8,8 M SCI ulSizeOfChannel->DP:\$6C00D M SCI usSizeOfMailbox->Y:\$6C00E,0,16 M_SCI_usMailboxStartOffset->X:\$6C00E,0,16 M HCI bChannelType->Y:\$6C010,0,8 M HCI ulSizeOfChannel->DP:\$6C011 M CCOI bChannelType->Y:\$6C014,0,8 M CCOI bChannelId->Y:\$6C014,8,8 M CCOI bSizePositionOfHandshake->X:\$6C014,0,8 M CCOI bNumberOfBlocks->X:\$6C014,8,8 M CC0I ulSizeOfChannel->DP:\$6C015 M CCOI usCommunicationClass->Y:\$6C016,0,16 M CCOI usProtocolClass->X:\$6C016,0,16 M CCOI usConformanceClass->Y:\$6C017,0,16 M_CC1I_bChannelType->Y:\$6C018,0,8
M_CC1I_bChannelId->Y:\$6C018,8,8 M CC11 bSizePositionOfHandshake->X:\$6C018,0,8 M CC1I bNumberOfBlocks->X:\$6C018,8,8 M CC1I ulSizeOfChannel->DP:\$6C019 M CC1I usCommunicationClass->Y:\$6C01A,0,16 M CC1I usProtocolClass->X:\$6C01A,0,16 M CC1I usConformanceClass->Y:\$6C01B,0,16 M_CC2I_bChannelType->Y:\$6C01C,0,8 M CC2I bChannelId->Y:\$6C01C,8,8 M CC2I bSizePositionOfHandshake->X:\$6C01C,0,8 M CC2I bNumberOfBlocks->X:\$6C01C,8,8 M CC2I ulSizeOfChannel->DP:\$6C01D M CC2I usCommunicationClass->Y:\$6C01E,0,16 M CC2I usProtocolClass->X:\$6C01E,0,16 M CC2I usConformanceClass->Y:\$6C01F,0,16 M CC3I bChannelType->Y:\$6C020,0,8 M CC3I bChannelId->Y:\$6C020,8,8 M CC3I bSizePositionOfHandshake->X:\$6C020,0,8 M CC3I bNumberOfBlocks->X:\$6C020,8,8 M CC3I ulSizeOfChannel->DP:\$6C021 M_CC3I_usCommunicationClass->Y:\$6C022,0,16 M CC3I usProtocolClass->X:\$6C022,0,16 M CC3I usConformanceClass->Y:\$6C023,0,16 M ACOI bChannelType->Y:\$6C024,0,8 M ACOI bChannelId->Y:\$6C024,8,8

M ACOI bSizePositionOfHandshake->X:\$6C024,0,8 M ACOI bNumberOfBlocks->X:\$6C024,8,8 M ACOI ulSizeOfChannel->DP:\$6C025 M AC11 bChannelType->Y:\$6C028,0,8 M AC1I bChannelId->Y:\$6C028,8,8 M AC11 bSizePositionOfHandshake->X:\$6C028,0,8 M_AC1I_bNumberOfBlocks->X:\$6C028,8,8 M AC1I ulSizeOfChannel->DP:\$6C029 M_SCtrl_ulSystemCommandCOS->DP:\$6C02E M SStat ulSystemCOS->DP:\$6C030 M SStat ulSystemStatus->DP:\$6C031 M_SStat_ulSystemError->DP:\$6C032 M SStat ulBootError->DP:\$6C033 M SStat ulTimeSinceStart->DP:\$6C034 M SStat usCpuLoad->Y:\$6C035,0,16 M_SStat_ulHWFeatures->DP:\$6C036 M SSMB usPackagesAccepted->Y:\$6C040,0,16 M SSMB ulDest->DP:\$6C041 M SSMB ulSrc->DP:\$6C042 M SSMB ulDestId->DP:\$6C043 M SSMB ulSrcId->DP:\$6C044 M SSMB ullen->DP:\$6C045 M_SSMB_ulId->DP:\$6C046 M SSMB ulState->DP:\$6C047 M SSMB ulCmd->DP:\$6C048 M SSMB ulExt->DP:\$6C049 M SSMB ulRout->DP:\$6C04A M SSMB ultData0->DP:\$6C04B M SSMB ultData1->DP:\$6C04C M SSMB ultData2->DP:\$6C04D M SSMB ultData3->DP:\$6C04E M SSMB ultData4->DP:\$6C04F M SSMB ultData5->DP:\$6C050 M SSMB ultData6->DP:\$6C051 M SSMB ultData7->DP:\$6C052 M SSMB ultData8->DP:\$6C053 M SSMB ultData9->DP:\$6C054 M SSMB ultData10->DP:\$6C055 M SSMB ultData11->DP:\$6C056 M SSMB ultData12->DP:\$6C057 M SSMB ultData13->DP:\$6C058 M SSMB ultData14->DP:\$6C059 M SSMB ultData15->DP:\$6C05A M SSMB ultData16->DP:\$6C05B M SSMB ultData17->DP:\$6C05C M SSMB ultData18->DP:\$6C05D M SSMB ultData19->DP:\$6C05E M SSMB ultData20->DP:\$6C05F M SRMB usWaitingPackages->Y:\$6C060,0,16 M SRMB ulDest->DP:\$6C061 M SRMB ulSrc->DP:\$6C062 M SRMB ulDestId->DP:\$6C063 M SRMB ulSrcId->DP:\$6C064 M SRMB ullen->DP:\$6C065 M_SRMB_ulid->DP:\$6C066 M SRMB ulState->DP:\$6C067 M SRMB ulCmd->DP:\$6C068 M_SRMB_ulExt->DP:\$6C069 M SRMB ulRout->DP:\$6C06A M SRMB ultData0->DP:\$6C06B M SRMB ultData1->DP:\$6C06C M SRMB ultData2->DP:\$6C06D M SRMB ultData3->DP:\$6C06E M SRMB ultData4->DP:\$6C06F M SRMB ultData5->DP:\$6C070 M SRMB ultData6->DP:\$6C071 M_SRMB_ultData7->DP:\$6C072 M SRMB ultData8->DP:\$6C073 M_SRMB_ultData9->DP:\$6C074 M SRMB ultData10->DP:\$6C075 M SRMB ultData11->DP:\$6C076

M SRMB ultData12->DP:\$6C077 M SRMB ultData13->DP:\$6C078 M SRMB ultData14->DP:\$6C079 M SRMB ultData15->DP:\$6C07A M SRMB ultData16->DP:\$6C07B M SRMB ultData17->DP:\$6C07C M SRMB ultData18->DP:\$6C07D M SRMB ultData19->DP:\$6C07E M_SRMB_ultData20->DP:\$6C07F M HCSC bNetxFlags->X:\$6C080,0,8 M HCSC NSF READY->X:\$6C080,0,1 M HCSC NSF ERROR->X:\$6C080,1,1 M HCSC NSF HOST COS ACK->X:\$6C080,2,1 M HCSC NSF NETX COS CMD->X:\$6C080,3,1 M HCSC NSF SEND MBX ACK->X:\$6C080,4,1 M_HCSC_NSF_RECV_MBX_CMD->X:\$6C080,5,1 M HCSC bHostFlags->X:\$6C080,8,8 M HCSC HSF RESET->X:\$6C080,8,1 M HCSC HSF BOOTSTART->X:\$6C080,9,1 M HCSC HSF HOST COS CMD->X:\$6C080,10,1 M HCSC HSF NETX COS ACK->X:\$6C080,11,1 M HCSC HSF SEND MBX CMD->X:\$6C080,12,1 M_HCSC_HSF_RECV_MBX_ACK->X:\$6C080,13,1 M HCCCO usNetxFlags->Y:\$6C082,0,16 M HCCC0 NCF COMMUNICATING->Y:\$6C082,0,1 M_HCCC0_NCF_ERROR->Y:\$6C082,1,1 M HCCC0 NCF HOST COS ACK->Y:\$6C082,2,1 M HCCC0 NCF NETX COS CMD->Y:\$6C082,3,1 M HCCCO NCF SEND MBX ACK->Y:\$6C082,4,1 M_HCCC0_NCF_RECV_MBX_CMD->Y:\$6C082,5,1 M HCCCO_NCF_PD0_OUT_ACK->Y:\$6C082,6,1
M_HCCCO_NCF_PD0_IN_CMD->Y:\$6C082,7,1 M HCCCO NCF PD1 OUT ACK->Y:\$6C082,8,1 M_HCCC0_NCF_PD1_IN_CMD->Y:\$6C082,9,1 M HCCC0 usHostFlags->X:\$6C082,0,16 M HCCCO HCF HOST COS CMD->X:\$6C082,2,1 M_HCCC0_HCF_NETX_COS_ACK->X:\$6C082,3,1 M HCCC0 HCF SEND MBX CMD->X:\$6C082,4,1 M HCCC0 HCF RECV MBX ACK->X:\$6C082,5,1 M HCCCO HCF PDO OUT CMD->X:\$6C082,6,1 M_HCCC0_HCF_PD0_IN_ACK->X:\$6C082,7,1 M_HCCC0_HCF_PD1_OUT_CMD->X:\$6C082,8,1 M_HCCC0_HCF_PD1_IN_ACK->X:\$6C082,9,1 M_HCCC1_usNetxFlags->Y:\$6C083,0,16 M HCCC1 NCF COMMUNICATING->Y:\$6C083,0,1 M HCCC1 NCF ERROR->Y:\$6C083,1,1 M HCCC1 NCF HOST COS ACK->Y:\$6C083,2,1 M HCCC1 NCF NETX COS CMD->Y:\$6C083,3,1 M HCCC1 NCF SEND MBX ACK->Y:\$6C083,4,1 M HCCC1 NCF RECV MBX CMD->Y:\$6C083,5,1 M_HCCC1_NCF_PD0_OUT_ACK->Y:\$6C083,6,1 M HCCC1 NCF PD0 IN CMD->Y:\$6C083,7,1 M HCCC1 NCF PD1 OUT ACK->Y:\$6C083,8,1 M HCCC1 NCF PD1 IN CMD->Y:\$6C083,9,1 M_HCCC1_usHostFlags->X:\$6C083,0,16 M HCCC1 HCF HOST COS CMD->X:\$6C083,2,1 M HCCC1 HCF NETX COS ACK->X:\$6C083,3,1 M_HCCC1_HCF_SEND_MBX_CMD->X:\$6C083,4,1 M HCCC1 HCF RECV MBX ACK->X:\$6C083,5,1
M HCCC1 HCF PD0 OUT CMD->X:\$6C083,6,1 M HCCC1 HCF PD0 IN ACK->X:\$6C083,7,1 M HCCC1 HCF PD1 OUT CMD->X:\$6C083,8,1 M HCCC1 HCF PD1 IN ACK->X:\$6C083,9,1 M HCCC2 usNetxFlags->Y:\$6C084,0,16 M_HCCC2_NCF_COMMUNICATING->Y:\$6C084,0,1 M_HCCC2_NCF_ERROR->Y:\$6C084,1,1 M_HCCC2_NCF_HOST_COS_ACK->Y:\$6C084,2,1 M HCCC2 NCF NETX COS CMD->Y:\$6C084,3,1 M_HCCC2_NCF_SEND_MBX_ACK->Y:\$6C084,4,1 M HCCC2 NCF RECV MBX CMD->Y:\$6C084,5,1 M HCCC2 NCF PD0 OUT ACK->Y:\$6C084,6,1

M HCCC2 NCF PD0 IN CMD->Y:\$6C084,7,1 M HCCC2_NCF_PD1_OUT_ACK->Y:\$6C084,8,1 M HCCC2_NCF_PD1_IN_CMD->Y:\$6C084,9,1 M HCCC2 usHostFlags->X:\$6C084,0,16 M HCCC2 HCF HOST COS CMD->X:\$6C084,2,1 M_HCCC2_HCF_NETX_COS_ACK->X:\$6C084,3,1 M_HCCC2_HCF_SEND_MBX_CMD->X:\$6C084,4,1 M HCCC2 HCF RECV MBX ACK->X:\$6C084,5,1 M_HCCC2_HCF_PD0_OUT_CMD->X:\$6C084,6,1 M_HCCC2_HCF_PD0_IN_ACK->X:\$6C084,7,1 M HCCC2 HCF PD1 OUT CMD->X:\$6C084,8,1 M_HCCC2_HCF_PD1_IN_ACK->X:\$6C084,9,1 M HCCC3 usNetxFlags->Y:\$6C085,0,16 M HCCC3 NCF COMMUNICATING->Y:\$6C085,0,1 M HCCC3 NCF ERROR->Y:\$6C085,1,1 M_HCCC3_NCF_HOST_COS_ACK->Y:\$6C085,2,1 M HCCC3 NCF NETX COS CMD->Y:\$6C085,3,1 M HCCC3 NCF SEND MBX ACK->Y:\$6C085,4,1 M HCCC3 NCF RECV MBX CMD->Y:\$6C085,5,1 M_HCCC3_NCF_PD0_OUT_ACK->Y:\$6C085,6,1 M_HCCC3_NCF_PD0_IN_CMD->Y:\$6C085,7,1 M HCCC3 NCF PD1 OUT ACK->Y:\$6C085,8,1 M_HCCC3_NCF_PD1_IN_CMD->Y:\$6C085,9,1 M HCCC3 usHostFlags->X:\$6C085,0,16 M HCCC3 HCF HOST COS CMD->X:\$6C085,2,1 M_HCCC3_HCF_NETX_COS_ACK->X:\$6C085,3,1 M_HCCC3_HCF_SEND_MBX_CMD->X:\$6C085,4,1 M_HCCC3_HCF_RECV_MBX_ACK->X:\$6C085,5,1 M HCCC3 HCF PD0 OUT CMD->X:\$6C085,6,1 M_HCCC3_HCF_PD0_IN_ACK->X:\$6C085,7,1 M HCCC3 HCF PD1 OUT CMD->X:\$6C085,8,1 M HCCC3 HCF PD1 IN ACK->X:\$6C085,9,1 M HCAC0 usNetxFlags->Y:\$6C086,0,16 M HCACO NCF COMMUNICATING->Y:\$6C086,0,1
M HCACO NCF ERROR->Y:\$6C086,1,1 M HCACO NCF HOST COS ACK->Y:\$6C086,2,1 M_HCAC0_NCF_NETX_COS_CMD->Y:\$6C086,3,1 M HCACO NCF SEND MBX ACK->Y:\$6C086,4,1 M HCACO NCF RECV MBX CMD->Y:\$6C086,5,1 M HCACO NCF PDO OUT ACK->Y:\$6C086,6,1 M_HCACO_NCF_PD0_IN_CMD->Y:\$6C086,7,1 M_HCACO_NCF_PD1_OUT_ACK->Y:\$6C086,8,1 M_HCACO_NCF_PD1_IN_CMD->Y:\$6C086,9,1 M HCACO usHostFlags->X:\$6C086,0,16 M HCACO HCF HOST COS CMD->X:\$6C086,2,1 M HCACO HCF NETX COS ACK->X:\$6C086,3,1 M HCACO HCF SEND MBX CMD->X:\$6C086,4,1 M HCAC0 HCF RECV MBX ACK->X:\$6C086,5,1
M HCAC0 HCF PD0 OUT CMD->X:\$6C086,6,1 M HCACO HCF PDO IN ACK->X:\$6C086,7,1 M_HCAC0_HCF_PD1_OUT_CMD->X:\$6C086,8,1 M HCACO HCF PD1 IN ACK->X:\$6C086,9,1 M HCAC1 usNetxFlags->Y:\$6C087,0,16 M HCAC1 NCF COMMUNICATING->Y:\$6C087,0,1 M HCAC1_NCF_ERROR->Y:\$6C087,1,1
M HCAC1_NCF_HOST_COS_ACK->Y:\$6C087,2,1 M HCAC1 NCF NETX COS CMD->Y:\$6C087,3,1 M_HCAC1_NCF_SEND_MBX_ACK->Y:\$6C087,4,1 M_HCAC1_NCF_RECV_MBX_CMD->Y:\$6C087,5,1 M HCAC1 NCF PD0 OUT ACK->Y:\$6C087,6,1 M HCAC1 NCF PD0 IN CMD->Y:\$6C087,7,1 M HCAC1_NCF_PD1_OUT_ACK->Y:\$6C087,8,1 M HCAC1_NCF_PD1_IN_CMD->Y:\$6C087,9,1 M HCAC1 usHostFlags->X:\$6C087,0,16 M_HCAC1_HCF_HOST_COS_CMD->X:\$6C087,2,1 M HCAC1 HCF NETX COS ACK->X:\$6C087,3,1 M HCAC1 HCF SEND MBX CMD->X:\$6C087,4,1 M HCAC1 HCF RECV MBX ACK->X:\$6C087,5,1 M HCAC1 HCF PD0 OUT CMD->X:\$6C087,6,1 M HCAC1 HCF PD0 IN ACK->X:\$6C087,7,1 M HCAC1 HCF PD1 OUT CMD->X:\$6C087,8,1

```
M HCAC1 HCF PD1 IN ACK->X:$6C087,9,1
M_CC0_RCX_APP_COS_APP_READY->Y:$6C0C2,0,1
M_CC0_RCX_APP_COS_BUS_ON->Y:$6C0C2,1,1
M CCO RCX APP COS BUS ON ENABLE->Y:$6C0C2,2,1
M CCO RCX APP COS INIT->Y:$6C0C2,3,1
M_CC0_RCX_APP_COS_INIT_ENABLE->Y:$6C0C2,4,1
M_CC0_RCX_APP_COS_LOCK_CFG->Y:$6C0C2,5,1
M CCO RCX APP COS LOCK CFG ENA->Y:$6C0C2,6,1
M_CC0_RCX_APP_COS_DMA->Y:$6C0C2,7,1
M_CC0_RCX_APP_COS_DMA_ENABLE->Y:$6C0C2,8,1
M CCO ulDeviceWatchdog->DP:$6C0C3
M_CC0_RCX_COMM_COS_READY->Y:$6C0C4,0,1
M CC0 RCX COMM COS RUN->Y:$6C0C4,1,1
M CC0 RCX COMM COS BUS_ON->Y:$6C0C4,2,1
M CCO RCX COMM COS CONFIG LOCKED->Y:$6C0C4,3,1
M_CC0_RCX_COMM_COS_CONFIG_NEW->Y:$6C0C4,4,1
M_CCO_RCX_COMM_COS_RESTART_REQ->Y:$6C0C4,5,1
M_CCO_RCX_COMM_CO_REQ_ENA->Y:$6C0C4,6,1
M CC0 RCX COMM COS DMA->Y:$6C0C4,7,1
M CC0 ulCommunicationState->DP:$6C0C5
M CC0 ulCommunicationError->DP:$6C0C6
M CCO usVersion->Y:$6C0C7,0,16
M_CC0_usWatchdogTime->X:$6C0C7,0,16
M CC0 bPDInHskMode->Y:$6C0C8,0,8
M CC0 bPDInSource->Y:$6C0C8,8,8
M CC0 bPDOutHskMode->X:$6C0C8,0,8
M CC0 bPDOutSource->X:$6C0C8,8,8
M CC0 ulHostWatchdog->DP:$6C0C9
M CCO ulErrorCount->DP:$6C0CA
M CCO bErrorLogInd->Y:$6C0CB,0,8
M CCO bErrorPDInCnt->Y:$6C0CB,8,8
M CC0 bErrorPDOutCnt->X:$6C0CB,0,8
M CC0 bErrorSyncCnt->X:$6C0CB,8,8
M CC0 bSyncHskMode->Y:$6C0CC,0,8
M CCO bSyncSource->Y:$6C0CC,8,8
M CCO ulSlaveState->DP:$6C0CE
M CCO ulSlaveErrLogInd->DP:$6C0CF
M CC0 ulNumOfConfigSlaves->DP:$6C0D0
M CC0 ulNumOfActiveSlaves->DP:$6C0D1
M CC0 ulNumOfDiagSlaves->DP:$6C0D2
```

APPENDIX B – TURBO PMAC MEMORY MAPS

	PROFIBUS-DP Master	PROFIBUS-DP Slave	DeviceNet Master	DeviceNet Slave	CANopen Master	CANopen Slave	CC-Link Slave	EtherCAT Master	EtherCAT Slave	EtherNet/IP Scanner/Master	EtherNet/IP Adapter/Slave	Open Modbus/TCP	PROFINET IO Controller/Master	PROFINET IO Device/Slave
CC-72EX Address	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000	\$6C000
etX Identification ual-Port Memory Size	netX 16384 bytes	netX 8192 bytes	netX 16384 bytes	netX 8192 bytes	netX 65536 bytes	netX 8192 bytes	netX 8192 bytes	netX 16384 bytes	netX 16384 bytes	netX 65536 bytes	netX 16384 bytes	netX 16384 bytes	netX 32768 bytes	netX 32768 bytes
evice Number	1532410	1562420	1532510	1562520	1532500	1562540	1562740	1532100	1532100	1532100	1532100	1532100	1532100	1532100
ardware Assembly Options Port 0	NOT CONNECTED	PROFIBUS	NOT CONNECTED	DEVICENET	NOT CONNECTED	CAN	CC-LINK	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)
Port 1	NOT CONNECTED	NOT AVAILABLE	NOT CONNECTED	NOT AVAILABLE	NOT CONNECTED	NOT AVAILABLE	NOT AVAILABLE	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)	ETHERNET (internal Phy)
Port 2 Port 3	PROFIBUS NOT CONNECTED	NOT AVAILABLE NOT AVAILABLE	DEVICENET NOT CONNECTED	NOT AVAILABLE NOT AVAILABLE	CAN NOT CONNECTED	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	NOT CONNECTED NOT CONNECTED	NOT CONNECTED NOT CONNECTED	NOT CONNECTED NOT CONNECTED	NOT CONNECTED NOT CONNECTED	NOT CONNECTED NOT CONNECTED	NOT CONNECTED NOT CONNECTED	NOT CONNECTED NOT CONNECTED
lilscher Module License Information	(PROFIBUS Master) (CANopen	Unlimited number of master	(PROFIBUS Master) (CANopen	Unlimited number of master	(PROFIBUS Master) (CANopen	Unlimited number of master	Unlimited number of master	(PROFIBUS Master) (CANopen	Unlimited number of master	(PROFIBUS Master) (CANopen	Unlimited number of master	Unlimited number of master	(PROFIBUS Master) (CANopen	Unlimited number of master
	Master) (DeviceNet Master) (AS- Interface Master) (PROFINET IO RT	licenses	Master) (DeviceNet Master) (AS- Interface Master) (PROFINET IO RT	licenses	Master) (DeviceNet Master) (AS- Interface Master) (PROFINET IO RT	licenses	licenses	Master) (DeviceNet Master) (AS- Interface Master) (PROFINET IO RT	licenses	Master) (DeviceNet Master) (AS- Interface Master) (PROFINET IO RT	licenses	licenses	Master) (DeviceNet Master) (AS- Interface Master) (PROFINET IO RT	licenses
	Controller) (EtherCAT Master) (EtherNet/IP Scanner) (SERCOS III		Controller) (EtherCAT Master)		Controller) (EtherCAT Master)			Controller) (EtherCAT Master) (EtherNet/IP Scanner) (SERCOS III		Controller) (EtherCAT Master)			Controller) (EtherCAT Master)	
	(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License		(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License		(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License			(EtherNet/IP Scanner) (SERCOS III Master) Unlimited number of		(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License			(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License	
								master licenses						
ool License Information Device Class	(SYCON.net) COMX 100	COMX 10	(SYCON.net) COMX 100	COMX 10	(SYCON.net) COMX 100	COMX 10	COMX 10	(SYCON.net) COMX 100	COMX 100	(SYCON.net) COMX 100	COMX 100	COMX 100	(SYCON.net) COMX 100	COMX 100
Block 0 Channel Type	System	System	System	System	System	System	System	System	System	System	System	System	System	System
Size of Channel	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes	512 bytes
Channel Start Address Position of Handshake Cells	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL	\$6C000 IN HANDSHAKE CHANNEL
netX System Flags Adress	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8	X:\$6C080,0,8
Host System Flags Adress	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8	X:\$6C080,8,8
Size of Handshake Cells Size of Mailbox	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes
Mailbox Start address	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040	\$6C040
Number of Subblocks	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Subblock 0	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS
Size	176 bytes \$6000	176 bytes \$6000	176 bytes \$6000	176 bytes \$60000	176 bytes \$6000	176 bytes \$6000	176 bytes \$6C000	176 bytes \$6C000	176 bytes \$6C000	176 bytes \$6000	176 bytes \$6000	176 bytes \$6000	176 bytes \$6000	176 bytes \$6000
Start Offset Transfer Direction	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)	\$6C000 IN - OUT (Bi-Directional)
Transfer Type	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)
Handshake Mode Handshake Bit	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0
		5	5	5	0		5		5	,		5	5	
Subblock 1 Size	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes
Start Offset	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E	\$6C02E
Transfer Direction	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)
Transfer Type Handshake Mode	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED
Handshake Bit	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Subblock 2	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS
Size	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes
Start Offset	\$6C030	\$6C030	\$6C030 IN (netX to Host System)	\$6C030	\$6C030	\$6C030	\$6C030	\$6C030	\$6C030	\$6C030 IN (netX to Host System)	\$6C030	\$6C030	\$6C030	\$6C030
Transfer Direction Transfer Type	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)
Handshake Mode	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED
Handshake Bit	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Subblock 3	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX
Size Start Offset	128 bytes \$6C040	128 bytes \$6C040	128 bytes \$60040	128 bytes \$6C040	128 bytes \$6C040	128 bytes \$6C040	128 bytes \$6C040	128 bytes \$6C040	128 bytes \$6C040	128 bytes \$60040	128 bytes \$6C040	128 bytes \$6C040	128 bytes \$60040	128 bytes \$6C040
Transfer Direction	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)
Transfer Type	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)
Handshake Mode Handshake Bit	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED 4
1														
Subblock 4 Size	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes
Start Offset	\$6C060	\$60060	\$6C060	\$60060	\$6C060	\$60060	\$6C060	\$6C060	\$60060	\$60060	\$6C060	\$6C060	\$60060	\$6C060
Transfer Direction	IN (netX to Host System)	1117 - 117 - 11 - 1 C - 1	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)
Transfer Type		IN (netX to Host System)				DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory) UNKNOWN	DPM (Dual-Port Memory) UNKNOWN	DPM (Dual-Port Memory)	DPM (Dual-Port Memory) UNKNOWN
Handshake Mode	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)			UNKNOWN	UNKNOWN					
Handshake Mode Handshake Bit			DPM (Dual-Port Memory) UNKNOWN 5	DPM (Dual-Port Memory) UNKNOWN 5	DPM (Dual-Port Memory) UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	5	5	UNKNOWN 5	5
Handshake Bit	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)						UNKNOWN 5	UNKNOWN 5	5	5	5	5	5
Handshake Bit • Block 1	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)						UNKNOWN 5 Handshake	UNKNOWN 5 Handshake	5 Handshake	5 Handshake	5 Handshake	UNKNUWN 5 Handshake	5 Handshake
Handshake Bit Block 1 Channel Type Size of Channel	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes	UNKNOWN 5 Handshake 256 bytes	UNKNOWN 5 Handshake 256 bytes	UNKNOWN 5 Handshake 256 bytes	UNKNOWN 5 Handshake 256 bytes	UNKNOWN 5 Handshake 256 bytes	5 Handshake 256 bytes	5 Handshake 256 bytes	5 Handshake 256 bytes	5 Handshake 256 bytes	5 Handshake 256 bytes	5 Handshake 256 bytes	256 bytes
Handshake Bit Block 1 Channel Type Size of Channel	DPM (Dual-Port Memory) UNKNOWN 5 Handshake	DPM (Dual-Port Memory) UNKNOWN 5 Handshake	UNKNOWN 5 Handshake	UNKNOWN 5 Handshake	UNKNOWN 5 Handshake	UNKNOWN 5 Handshake	UNKNOWN 5 Handshake	5 Handshake	5 Handshake	5 Handshake	5 Handshake	5 Handshake	5 Handshake	
Handshake Bit Block 1 Channel Type Size of Channel Channel Start Address Block 2	DPM (Dual-Port Memory) UNINOWN 5 Handshake 256 bytes 56C080	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes \$66080	UNKNOWN S Handshake 256 bytes S6C080	UNKNOWN 5 Handshake 256 bytes 566080	UNKNOWN 5 Handshake 256 bytes \$6C080	UNKNOWN 5 Handshake 256 bytes 56C080	UNKNOWN 5 Handshake 256 Dytes SeC080	5 Handshake 256 bytes \$6C080	S Handshake 256 bytes \$6C080	5 Handshake 256 bytes \$6C080	S Handshake 256 bytes \$6C080	5 Handshake 256 bytes \$6C080	5 Handshake 256 bytes \$6C080	256 bytes \$6C080
Handshake Bit Block 1 Channel Type Size of Channel Channel Start Address Block 2 Channel Type	DPM (bui-Port Memory) UNINOWN 5 Handshake 256 bytes 56C080 Communication	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56C080 Communication	UNINOWN 5 Handshake 256 bytes 56C080 Communication	UNKNOWN 5 Handshake 256 bytes \$6C080 Communication	UNKNOWN 5 Handshake 256 bytes 56C080 Communication	UNKNOWN 5 Handshake 256 bytes \$6C080 Communication	UNKNOWN 5 Handshake 256 bytes \$6C080 Communication	S Handshake 256 bytes 56C080 Communication	S Handshake 256 bytes 56C080 Communication	S Handshake 256 bytes 56C080 Communication	S Handshake 256 bytes 56C080 Communication	5 Handshake 256 bytes S6C080 Communication	5 Handshake 256 bytes S6C080 Communication	256 bytes \$6C080 Communication
Handshake Bit Block 1 Channel Type Size of Channel Channel Start Address Block 2 Channel Type Size of Channel Channel Start Address	DPM (bui-Port Memory) UNIXNOWN S Handshake 256 bytes \$6C080 Communication 15516 bytes \$6C0C0	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C00	UNKKOWN 5 Handshake 256 bytes 55C080 Communication 155616 bytes 55CC0	UNKNOWN F Handshake 256 bytes \$6C080 Communication 7424 bytes \$6C00	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 155616 bytes 56C00	UNKNOWN F Handshake 256 bytes 56C080 Communication 7424 bytes 56C00	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C00	S Handshake 256 bytes 56C080 Communication 15516 bytes 56C0C0	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C00	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0	S Handshake 256 bytes 56080 Communication 15616 bytes 56000	5 Handshake 256 bytes 56C080 Communication 15636 bytes 56C0C0	256 bytes \$6C080 Communication 15616 bytes \$6C0C0
Handshake Bit Block 1 Channel Type Channel Start Address Block 2 Channel Type Size of Channel Channel Start Address Position of Handshake Cells	DPM (Dual-Port Memory) UNINOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHARE (HANNEL	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56 componication 7424 bytes 56 ccco IN HANDSHAKE CHANNEL	UNKKNOWN 5 Handshake 256 bytes 56:080 Communication 15616 bytes 56:000 IN HANDSHAKE CHANNEL	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANDSHARE CHANNEL	UNKNOWN 5 Handbale 256 bytes 56C080 Communication 15616 bytes 56C00 IN HANDSHARE CHANNEL	UNKNOWN 5 Handshale 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANDSHARE CHANNEL	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANDSHAKE CHANNEL	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56CCC0 IN HANDSHAKE CHANNEL	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL	256 bytes S6C080 Communication 15616 bytes S6C0C0 IN HANDSHAKE CHANNEL
Handshake Bit Block 1 Channel Type Size of Channel Channel Start Address Block 2 Channel Type Size of Channel Channel Start Address Position of Handshake Cells Size of Handshake Cells	DPM (bui-Port Memory) UNIXNOWN S Handshake 256 bytes \$6C080 Communication 15516 bytes \$6C0C0	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56 cos0 Communication 7424 bytes 56 coc0 IN HANDSHAKE (HANNEL 16 BITS Y-\$5002.0.16	UNKKIOWN 5 Handshake 256 bytes 56C080 Communication 156 16 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS V*\$5C028,0,16	UNKNOWN 5 Handbate 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANDSHARE CHANNEL 16 BITS Y*\$5C082,0,16	UNKKNOWN 5 Handhale 256 bytes 56C00 Communication 15616 bytes 56C00 IN HANDSHARE CHANNEL 16 BITS Y*SEC020,0.6	UNKNOWN 5 Handhale 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN FANDSHARE CHANNEL 16 BITS Y-\$5C082,0,16	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y*\$5C082,0,16	5 Handshake 256 bytes 56C080 Communication 15516 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y*\$5C022,0,16	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C00	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS v*55CR20,216	5 Handshake 256 bytes 56 como 15616 bytes 56 coco IN HANDSHAKE (HANNEL 16 BITS Y-\$50 20.0.16	5 Handshake 256 bytes 56000 Communication 15616 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS Y:\$6008.0,16	5 Handshake 256 bytes 56C080 Communication 15636 bytes 56C0C0	256 bytes S6C080 Communication 15616 bytes 56C0CO IN HANDSHAKE CHANNEL 16 BITS V*SEC028,0,16
Handshake Bit Block 1 Channel Type Size of Channel Channel Start Address Block 2 Channel Type Size of Channel Channel Start Address Position of Handshake Cells Size of Handshake Cells Netk Handshake Register Host Handshake Register	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 2:56 bytes 56:000 Communication 1:56:16 bytes 5:60:00 IIII HANDSHAKE (HANNEL 1:6 6175 Y:56:0082,0,16 X:56:0082,0,16	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C000 IN HANDSHAKE CHANNEL 16 BITS Y-\$5C082,0,16 X-\$5C082,0,16	UNKKIOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y:\$5C028,0,16	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C00 N HANDSHAKE CHANNEL 16 BTS Y-\$5C082,0,16 X-\$5C082,0,16	UNKNOWN 5 Handbake 256 bytes 55C080 Communication 15616 bytes 56C000 IN HANDSHAKE CHANNEL 16 BITS Y*55C020, J.16 X*55C082, J.16	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y-\$5C082,0,16 X-\$5C082,0,16	UNKKNOWN 5 Handshake 256 bytes 55C080 Communication 7424 bytes 56C00 N NANDSHAKE CHANNEL 16 BITS Y*\$5C023,0.16	S Handshake 255 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS V*\$5C028,0,16	S Handshake 255 bytes 56C080 Communication 155616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y*\$5C022,0,16	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS YS5C022,0,16	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Yr:55C082,0,16 Xr:55C082,0,16	S Handshake 256 bytes 56080 Communication 15616 bytes 560200 IN HANDSHAKE CHANNEL 16 BITS YS65082,0,16 X560282,0,16	5 Handshake 256 bytes 560080 Communication 15616 bytes 560000 IN HANDSHAKE CHANNEL 16 BITS Y-\$60082,0,16 X-\$60082,0,16	256 bytes Secose Communication 15616 bytes Secoco INI HANDSHAKE CHANNEL 16 BITS Y:SECO29.0,16 X:SECO29.0,16
Handshake Bit Block 1 Channel Type Size of Channel Channel Start Address Block 2 Channel Type Size of Channel Size of Channel Size of Channel Size of Channel Size of Handshake Cells Size of Handshake Cells NetX Handshake Register Hoot Handshake Register Communication Class Protocol Class	DPM (Dual-Port Memory) UNINOWN 5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHARE CHANNEL 16 BITS Y:56C020,0,16	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56 cos0 Communication 7424 bytes 56 coc0 IN HANDSHAKE CHANNEL 16 BITS Y-\$5002.0.16	UNKKIOWN 5 Handshake 256 bytes 56C080 Communication 156 16 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS V*\$5C028,0,16	UNKNOWN 5 Handbate 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANDSHARE CHANNEL 16 BITS Y*\$5C082,0,16	UNKKNOWN 5 Handhale 256 bytes 56C00 Communication 15616 bytes 56C00 IN HANDSHARE CHANNEL 16 BITS Y*SEC020,0.6	UNKNOWN 5 Handhale 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN FANDSHARE CHANNEL 16 BITS Y-\$5C082,0,16	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y*\$5C082,0,16	5 Handshake 256 bytes 56C080 Communication 15516 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y*\$5C022,0,16	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y*\$5C082,0,16	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS v*55CR20,216	5 Handshake 256 bytes 56 como 56 comounication 156 16 bytes 56 cocco IN HANDSHAKE CHANNEL 16 BITS Y-\$50 20.0.16	5 Handshake 256 bytes 56000 Communication 15616 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS Y:\$6008.0,16	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y-56C082,0,16	256 bytes S6C080 Communication 15616 bytes S6C0C0 IN HANDSHAKE CHANNEL 16 BITS Y:S6C020,0.16 X:S6C020,0.16 IO-DEVICE Programmable Logic Controller (PI
Handshake Bit Block 1 Channel Type Size of Channel Channel Type Channel Type Size of Channel Channel Start Address Block 2 Dosition of Handshake Cells Size of Handshake Register Host Handshake Register Host Handshake Register Communication Class Protocol Class Conformance Class	DPM (Dual-Port Memory) UNINOWN 5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHARE CHANNEL 16 BITS Y:\$6C082.0,16 X:\$6C082.0,16 X:\$6C082.0,16 MASTER	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 566	UNKKIOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y:\$5C028,0,16	UNKNOWN 5 Handbate 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANOSHAKE CHANNEL 16 BITS 7+56C082,0,16 X-56C082,0,16 SLAVE Server 0	UNKKNOWN 5 Handbake 256 bytes 56C080 Communication 15616 bytes 56C0C0 INI FANGDSHAKE CHANNEL 16 BITS Y-SEC082,0,16 X-SEC082,0,16 MASTER Scanner 0	UNKNOWN 5 Handhake 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS 7-55C022,0,16 X-55C022,0,16 SLAVE	UNKKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C0200 IN HANDSHAKE CHANNEL 16 BITS V:56C0220,16 X:56C022,0,16 SLAVE	5 Handshake 256 bytes 56 bytes 56 bytes 56 bytes 56 coco IN HANDSHAKE CHANNEL 16 BITS Y \$5 c028,0,16 X \$5 c028,0,16 MASTER	5 Handshake 256 bytes 56080 Communication 15616 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS Y*\$60020,016 X-\$50020,016 SLAVE	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y*\$5C082,0,16 XCANNER Io-Device 0	5 Handshake 256 bytes 56 commonication 15616 bytes 56 cocco IN HANDSHAKE CHANNEL 16 BITS 7, 56 core 2, 1, 16 X, 56 core 2, 1, 16 ADAPTER	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y:56C082,0,16 X:56C082,0,16 MESSAGING Combination Firmware 0	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHARE CHANNEL 16 BITS 756C082,0,16 X56C082,0,16 X56C082,0,16	256 bytes Secolo 15616 bytes Secoco IN HANDSHARE CHANNEL 16 BITS Y:SecOlo 2,016 X:SecOlo 2,016 DO DEVICE
Handshake Bit Block 1 Channel Type Size of Channel Channel Start Address Block 2 Channel Start Address Block 2 Disco of Channel Size of Channel Size of Channel Size of Channel Size of Handshake Cells Disco of Handshake Register Hext Handshake Register Communication Class Protocol Class	DPM (Dual-Port Memory) UNINOWN 5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHARE CHANNEL 16 BITS Y:\$6C082.0,16 X:\$6C082.0,16 X:\$6C082.0,16 MASTER	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 566	UNKKIOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y:\$5C028,0,16	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C00 N NANDSHAKE CHANNEL 16 BITS Y-\$5C082,0,16 X-\$5C082,0,16 SLAVE Server	UNKKNOWN 5 Handshate 256 bytes 56C080 Communication 15515 bytes 56C0C0 15515 bytes 56C0C0 15 bits 56C0C0 15 bits 16 bits 15 bits 16 bi	UNKNOWN 5 Handhake 256 bytes 56C080 Communication 7424 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS 7-55C022,0,16 X-55C022,0,16 SLAVE	UNKNOWN 5 Handbake 256 bytes 55C080 Communication 7424 bytes 56C00 IN HANDSHAKE CHANNEL 16 HITS V*S5C023,0.16 SLAVE Adapter	5 Handshake 256 bytes 56 bytes 56 bytes 56 bytes 56 coco IN HANDSHAKE CHANNEL 16 BITS Y \$5 c028,0,16 X \$5 c028,0,16 MASTER	5 Handshake 256 bytes 56080 Communication 15616 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS Y*\$60020,016 X-\$50020,016 SLAVE	5 Handshake 256 bytes 56000 Communication 15616 bytes 56020 IN HANDSHAKE CHANNEL 15 BITS YS50022,0,16 XS50022,0,16 SCANNER Io-Device	5 Handshake 256 bytes 560080 Communication 15616 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS Y:560082,0,16 X:560082,0,16 X:560082,0,16 X:560082,0,16 DeVice	5 Handshake 256 bytes 50000 Communication 15616 bytes 50000 IN HANDSHAKE CHANNEL 16 BITS Y-56008,0,16 X-55008,0,16 MESSARING	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHARE CHANNEL 16 BITS 756C082,0,16 X56C082,0,16 X56C082,0,16	256 bytes S6C080 Communication 15616 bytes S6C0C0 IN HANDSHAKE CHANNEL 16 BITS Y:S6C020,0.16 X:S6C020,0.16 IO-DEVICE Programmable Logic Controller (PI
Handshake Bit Block 1 Channel Yape Size of Channel Channel Yape Size of Channel Channel Start Address Block 2 Channel Start Address Position of Handshake Cells Size of Handshake Cells Size of Handshake Cells Conformatice Class Conformance Class C	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 255 bytes 56C080 Communication 15516 bytes 55C080 M HANDHAKCHANNEL 16 PUT 16 P	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 56C080 56C082 56C0	UNKKIOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 NIN HANDSHAKE CHANNEL 168175 VY56C022,0,16 MASTER 58rver 0 59 CONTROL	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 HITS Y-56C082,0,16 SLAVE SLAVE 54VE 9 CONTROL	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 15616 bytes 55C020 N NANDSHAKE CHANNEL 16 8175 55C020 N SCOR20, J.16 MASTER Scanner 0 9 CONTROL	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 ANT 56C020, 16 X-S5C082, 0, 16 SLAVE 5C082, 0, 16 SLA	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 7424 bytes 55C000 IN HANDSHAKE CHANNEL 16 BITS 55C0C0 IN HANDSHAKE CHANNEL 16 BITS V:\$5C023,0.16 SLAVE Adapter 0 9 CONTROL	5 Handshake 255 bytes 56C080 Communication 15516 bytes 56C020 IN BANG SHAKE CHANNEL 1550082.0.16 X550082.0.16 X550082.0.16 Mo-Controller 0 9 CONTROL	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN DANS SHAKE CHANNEL 1556C082 0.16 X56C082 0.16 X56C082 0.16 X56C082 0.16 Stave Io-Controller 0 9	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 WI JANSHAKE CHANNEL 1956078 0.16 X56C072,0.16 X56C072,0.16 SCANNER Io-Device 0 9	5 Handshake 256 bytes 56 bytes 56 bytes 56 correst 56 bytes 56 correst 19 M NAME CHANNEL 19 M NAME CHA	5 Handshake 256 bytes 56 communication 156 16 bytes 56 COCO IN HANCSHARE CHANNEL 1556 COR2,0 16 X55 COR2,0 16 MESSAGING CONTROL	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 1 545C082.0.16 K-56C082.0.15 K-56C082.0.15 Programmable Logic Controller (PIc) 0 9 5 CONTROL	256 bytes Secolo Communication 15616 bytes Secoco INI HANDSHAKE CHANNEL 16 BITS 14 SECOSO 116 14 SECOSO 116 14 Secolo 116 14 Secolo 116 14 Secolo 116 14 Secolo 116 14 Secolo 116 15 Secolo 116 15 Secolo 116 15 Secolo 116 15 Secolo 116 16 Secolo 116 17 Secolo 116 18 Sec
Handshake Bit Block 1 Channel Type Size of Channel Channel Start Address Block 2 Channel Start Address Dock 2 Channel Start Address Size of Handshake Cells Size of Handshake Register Host Handshake Register Host Handshake Register Communication Class Protocol Class Conformance Class Number of Subblocks Subblock 0 Size	DPM (buil-Port Memory) UNINOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHARE (HANNEL 16 BITS Y:SEC082,0,16 X:SEC082,0,16 X:SEC082,0,16 Managing Node 0 9 CONTROL 8 bytes	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y-\$5C082,0,16 X-\$5C082,0,16 SLAVE Managing Node 0 9 CONTROL 8 bytes	UNKKNOWN 5 5 Handshake 255 bytes 55C080 Communication 155616 bytes 56C020 56C020 56C020 56C020 56C020 56C020 56C020 55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C0220,15 X55C020,1	UNKNOWN 5 Handshke 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 56C080 9 56C080,0.16 54AVE 56C080,0.16 54AVE 56C080,0.16 54AVE 9 9 CONTROL 8 bytes	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 155616 bytes 56C080 Communication 155616 bytes 56C080 156C0	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 9 56C080 9 9 CONTROL 8 bytes	UNIKNOWN 5 Handshake 256 bytes 56C080 Communication 7824 bytes 56C080 Compared Channel His Juris VSEC082,0,16 SLAVE SLAVE 0 9 CONTROL 8 bytes	5 Handshake 256 bytes 56C080 Communication 1556 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y*56C022,0.16 X*55C022,0.16 MASTER Io-Controller 0 9 CONTROL 8 bytes	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y*\$6C082,0.16 SLAVE Io-Controller 0 9 CONTROL 8 bytes	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS V*\$5C082,0.16 SCANNER Ic-Device 0 9 CONTROL 8 bytes	5 Handshake 256 bytes 56 comounication 15616 bytes 56 coco IN HANDSHAKE (HANNEL 16 BITS Y-\$50082,0,16 X-\$50082,0,16 ADAPTEN Io-Device 0 9 CONTROL 8 bytes	5 Handshake 256 bytes 55C080 Communication 15616 bytes 55C020 IN HANDSHAKE CHANNEL 16 BITS Y:55C082,0,16 X:55C082,0,16 X:55C082,0,16 MESSAGING Combination Firmware 0 9 CONTROL 8 bytes	5 Handshake 256 bytes 56C080 Communication 15516 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS Y-\$6C082,0,16 X-\$6C082,0,16\\X-\$6C082,0,16\\X-\$6C082,0,16\\X-\$6C082,	256 bytes S6C080 Communication 15616 bytes S6C0C0 IN HANDSHAKE CHANNEL 16 BITS Y-S6C082,0,16 X-S6C082,0,16 I/O-DEVICE Programmable Logic Controller (PI 67 9 CONTROL 8 bytes
Handshake Bit Block 1 Channel Yope Size of Channel Channel Start Address Block 2 Channel Start Address Position of Handshake Cells Size of Channel Channel Start Address Fosition of Handshake Cells Size of Handshake Register Communication Class Conformance Class Co	DPM (buil-Port Memory) UNINOWN 5 	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 56C080 56C082 56C0	UNKKNOWN 5 Handbake 255 bytes 56C080 Communication 15616 bytes 56C00 Communication 15616 bytes 56C00 NIAND5HAKE CHANNEL 156175 MAXD5HAKE CHANNEL 156175 MAXD5HAKE CHANNEL 156175 MAXD5HAKE CHANNEL 156175 9 CONTROL 8 bytes 56C0C2 OUT (Hote System to netX)	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 HITS Y-56C082,0,16 SLAVE SLAVE 54VE 9 CONTROL	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 N NI ANDSHAKE CHANNEL 16 BITS 56C020 JL6 X 56C020 JL6 X 56C020 JL6 X 56C020 JL6 X 56C020 JL6 X 56C020 JL6 X 56C020 JL6 S 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 ANT 56C020, 16 X-S5C082, 0, 16 SLAVE 5C082, 0, 16 SLA	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 7424 bytes 55C000 IN HANDSHAKE CHANNEL 16 BITS 55C0C0 IN HANDSHAKE CHANNEL 16 BITS V:\$5C023,0.16 SLAVE Adapter 0 9 CONTROL	5 Handshake 255 bytes 56C080 Communication 15516 bytes 56C020 NI NAMSHAKE CHANNEL 1550082.0.16 X550082.0.16 X550082.0.16 NSS0082.0.16 N	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN DANS SHAKE CHANNEL 1556C082 0.16 X56C082 0.16 X56C082 0.16 X56C082 0.16 Stave Io-Controller 0 9	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS V*56C022,0.16 SCANNER Ib-Device 0 9 CONTROL 8 bytes 56C02 OUT (flota System to netX)	5 Handshake 256 bytes 56 bytes	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y:56C082,0.16 X:56C082,0.16 MESSAGING Combination Firmware 0 9 CONTROL 8 bytes 56C02 OUT (Hot System to netX)	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 1 545C082.0.16 K-56C082.0.15 K-56C082.0.15 Programmable Logic Controller (PIc) 0 9 5 CONTROL	256 bytes Secolo Communication 15616 bytes Secoco INI HANDSHAKE CHANNEL 16 BITS 14 SECOSO 116 14 SECOSO 116 14 Secolo 116 14 Secolo 116 14 Secolo 116 14 Secolo 116 14 Secolo 116 15 Secolo 116 15 Secolo 116 15 Secolo 116 15 Secolo 116 16 Secolo 116 17 Secolo 116 18 Sec
Handshake Bit Block 1 Channel Yape Size of Channel Channel Start Address Block 2 Channel Yape Size of Channel Channel Start Address Position of Handshake Cells Size of Handshake Register Host Handshake Register Size of Subblocks Subblock 0 Size Start Offset Transfer Direction Transfer Direction	DPM (Dual-Pott Memory) UNINGWN 5 Handshake 255 bytes 56C0C0 Communication 15515 bytes 56C0C0 IN HANDSHAKE CHANNEL 35875 56C0C0 IN HANDSHAKE CHANNEL 35875 56C0C0 IN HANDSHAKE CHANNEL 35875 56C0C0 0 9 9 CONTROL 8 bytes 56C0C0 2017 (Host System to netX) DPM (Dual-Pott Memory)	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56C080 Communication 742k bytes 56C080 Communication 742k bytes 56C080 Managing Node 0 9 CONTROL 8 bytes 56C020 OUTROL 8 bytes 56C020 OUTROL	UNKKOWN 5 Handshake 256 bytes 56C080 Communication 156156 bytes 56C020 Ni NANOSHAKE CHANNEL 158175 V:56C022,0,16 MASTER 58rver 0 5 9 CONTROL 8 bytes 56C022 0 CONTROL 8 bytes 56C022 0 UT (Host System to metX) DPM (Dual-Fort Memory)	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 HATS Y-56C082,0.16 X-56C082,0.16 SLAVE 54VE 0 9 CONTROL 8 bytes 56C02 OUT(Rol. System to netX) DPM (Dual-Port Memory)	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 15616 bytes 55C020 N HANDSHAKE CHANNEL 16 8175 456022.0.16 MASTER 55C022.0.16 MASTER 5Canner 0 9 CONTROL 8 bytes 55C022 00 (/ (Host System to netX) DPM (Duel-Pot Memory)	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 ANDSHAKE CHANNEL 16 ANDSHAKE CHANNEL 55C020 IN 6 ANDSHAKE CHANNEL 55C020 CONTROL 8 bytes 56C02 OUT (Host System to netX) DPM (Dual-Port Memory)	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 7424 bytes 55C080 Communication 7424 bytes 55C020 IN HANDSHAKE CHANNEL 16 BITS Y*55C023,0.16 SLAVE Adapter 0 9 CONTROL 8 bytes 55C02 OUT(RoL 55C02 OUT(RoL 55tem to netX) DPM(Duah Fort Memory)	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS 1756C020,016 X 56C02,016 X 56C02,016 CONTROL 8 bytes 56C0C2 OUT (Hoat System to netX) DPM (Dual-Port Memory)	5 Handshake 256 bytes 56 coso Communication 156 l5 bytes 56 coco IN HANDSHAKE CHANNEL 16 BITS 14 SEC082.0.16 K SEC082.0.16 K SEC082.0.16 S Controller 0 9 5 CONTROL 8 bytes 56 CC2 OUT (Host System to netX) DPM (Duel-Fort Memory)	5 Handshake 256 bytes 56C080 Commication 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 15 BITS 1*56C082,0,16 KSCN028,0,16 KSCN028,0,16 KSCN028,0,16 Io-Device 0 9 9 CONTROL 8 bytes 56C0C2 OUT (Host System to netX) DPM (Dual-Part Memory)	5 Handshake 256 bytes 256 bytes 56 Commiscation 15616 bytes 56C0C0 IN HANDSHAKE CHANNEL 16 BITS V-SEC082 0.16 X-SPATED, 16 b-Device 0 9 CONTROL 8 bytes 56C0C2 OUT (Host System to netX) DPM (Dual-Port Memory)	5 Handshake 256 bytes 256 bytes 256 bytes 56C00 Communication 15616 bytes 56C020 NN HANDSHAKE CHANNEL 16 8175 Y 55C020,016 X Y 55C020,016 Combination Firmware 0 9 CONTROL 8 bytes 56C02 OUT (Host System to netX) DPM (Usal-Port Memory)	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS 74 56C020,015 K05020,015 K05020,015 K05000 TROLLER Programmable Logic Controller (Pic) 0 9 5 CONTROL 8 bytes 56C0C2 OUT (Host System to netX) DPM (Dual-Port Memory)	256 bytes SEC080 Communication 15616 bytes SeC020 IN HANDSHAKE CHANNEL 16 BITS Y-SEC082,0,16 K-SEC082,0
Handshake Bit Block 1 Channel Ype Size of Channel Channel Start Address Block 2 Channel Start Address Dock 2 Channel Start Address Size of Channel Channel Start Address Position of Handshake Cells Size of Handshake Register Host Handshake Register Host Handshake Register Communication Class Protocol Class Conformance Class Number of Subblocks	DPM (buil-Port Memory) UNINOWN 5 	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56 commication 7424 bytes 56 coso Commication 7424 bytes 56 coco IN HANDSHAKE (HANNEL 16 BITS V-\$50082,0,16 X-\$50082,0,16 SLAVE Managing Node 0 9 CONTROL 8 bytes 56 coc2 OUT (Hox System to netX)	UNKKNOWN 5 Handbake 255 bytes 56C080 Communication 15616 bytes 56C00 Communication 15616 bytes 56C00 NIAND5HAKE CHANNEL 156175 MAXD5HAKE CHANNEL 156175 MAXD5HAKE CHANNEL 156175 MAXD5HAKE CHANNEL 156175 9 CONTROL 8 bytes 56C0C2 OUT (Hote System to netX)	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C02 NN HANDSHAKE CHANNEL 16 BITS 56V080,0,16 56AVE 56V08 0 9 CONTROL 8 bytes 56C02 OUT[flots System to netX]	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 N NI ANDSHAKE CHANNEL 16 BITS 56C020 JL6 X 56C020 JL6 X 56C020 JL6 X 56C020 JL6 X 56C020 JL6 X 56C020 JL6 X 56C020 JL6 S 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS 56C020 0 HANDSHAKE CHANNEL 16 BITS 56C020 0 Stave 5 Save 5	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C00 10 HANDSHAKE CHANNEL 11 H	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS V*56C020,0.16 MASTER Ic-Controller 0 9 CONTROL 8 bytes 56C02 OUT (Hot System to netX)	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS V*56C020,0.16 SLAVE Ic-Controller 0 9 CONTROL 8 bytes 56C022 0 COUT(flots System to netX)	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS V*56C022,0.16 SCANNER Ib-Device 0 9 CONTROL 8 bytes 56C02 OUT (flota System to netX)	5 Handshake 256 bytes 56 C080 Communication 15616 bytes 56 COC0 IN HANDSHAKE (CHANNEL 16 BITS Y-\$5082,0,16 ADAPTE Is-Device 0 9 CONTROL 8 bytes 56 COC2 OUT (Hox System to netX)	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y:56C082,0.16 X:56C082,0.16 MESSAGING Combination Firmware 0 9 CONTROL 8 bytes 56C02 OUT (Hot System to netX)	5 Handshake 256 bytes 56C080 Communication 15516 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y-\$6C082,0,16 X-\$6C082,0,16 IO-CONTROLLER Programmable Logic Controller (Pic) 0 9 CONTROL 8 bytes 56C0C2 OUT (Host System to netX)	256 bytes S6C080 Communication 15616 bytes S6C020 IN HANDSHAKE CHANNEL 16 BITS Y:S6C082,0,16 IV-S6C082,0,16 IV-DEVICE Programmable Logic Controller (PI 67 9 CONTROL 8 bytes S6C0C2 OUT (Host System to netX)
Handshake Bit Block 1 Channel Ype Size of Channel Channel Ype Size of Channel Channel Start Address Block 2 Channel Start Address Position of Handshake Cells Size of Handshake Register Host Handshake Register Host Handshake Register Communication Class Protocol Class Conformance Class Number of Subblocks	DPM (Dual-Port Memory) UNKNOWN 5 256 bytes 55000 Communication 15616 bytes 56000 Communication 15616 bytes 56000 16 HANDSHAKE (HANNEL 16 BITS 7: \$60082,0,16 X-560082,0,16	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C000 IN HANDSHAKE CHANNEL 16 BITS 56C000 IN HANDSHAKE CHANNEL 16 BITS 56C001 Managing Node 0 9 CONTROL 8 bytes 55C002 OUT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 15616 bytes 56C00 NI NANDSHARE CHANNEL 15 BITS 56CC0 NI NANDSHARE CHANNEL 15 BITS 56CC0 9 CONTROL 8 bytes 56CC2 0 UNCINTROL 8 bytes 56CC2 0 UNCINTROL 8 bytes 56CC2 0 UNCINTROL 8 bytes 56CC2 0 UNCINTROL 8 bytes 56CC2 0 UNCINTROL 8 bytes 56CC2 0 UNCINTROLED 0	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 N NAADSHAKE CHANNEL 16 BITS 56C020 N NAADSHAKE CHANNEL 16 BITS 56C020 9 CONTROL 8 bytes 56C020 9 CONTROL 8 bytes 56C020 0 UNCONTROLLED 00	UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 155616 bytes 56C020 NIX HANDSHAKE CHAINEL 15 BITS 56C020 NIX HANDSHAKE CHAINEL 15 BITS 56C020 NIX HANDSHAKE CHAINEL 16 BITS 56C020 0 CONTROL 8 bytes 56C021 OUT (Hots System to netX) DPM (Dual-Port Memory) UNIXONTROLLED 0	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 N HANDSHAKE CHANNEL 16 BITS 56C020 N HANDSHAKE CHANNEL 16 BITS 56C020 Scanner 0 9 CONTROL 8 bytes 56C020 9 CONTROL 8 bytes 56C020 0UT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C020 NIN FANDBAKE CHANNEL 15 BITS 56C020 NIN FANDBAKE CHANNEL 15 BITS SECO22,016 SEAVE Adapter 0 9 CONTROL 8 bytes 56C022 OUT (Hots System to netX) DPM (Dual-Port Memory) UNKCONTROLLED 0	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS V*\$6C020,0.16 X*\$5C020,0.16 MASTER Io-Controller 0 9 CONTROL 8 bytes 56C02 OUT (Houd System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 0	5 Handshake 256 bytes 55C080 Communication 15616 bytes 55C020 IN HANDSHAKE CHANNEL 16 BITS Y*55C022,0.16 SLAVE Io-Controller 0 9 CONTROL 8 bytes 55C022 OUT (Idua 5ystem to netX) DPM (Dual-Port Memory) UNCONTROLLED 0	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y*56C022,0,16 XCANNER Io-Device 0 9 CONTROL 8 bytes 56C022 0 9 CONTROL 8 bytes 56C022 0 0 10 CONTROL 8 bytes 56C022 0 0 CONTROL 8 bytes 56C022 0 UIT (Ious System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 0	5 Handshake 256 bytes 560000 Communication 15616 bytes 560000 IN HANDSHAKE CHANNEL 16 BITS Y*\$50082,0.16 X*\$50082,0.16 ADAPTEN Io-Device 0 9 CONTROL 8 bytes 56002 00UT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0	S Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y:56C082,0.16 X:56C082,0.16 X:56C082,0.16 MESSAGING Combination Firmware 0 9 CONTROL 8 bytes 56C02 OUT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROLLED	5 Handshake 256 bytes Secces Communication 15616 bytes Secces INH ANDSHARE CHANNEL 16 BITS Y-Secc082.0.16 X-Sec082.0.16 X-Sec082.0.16 DC-ONTROLLER Programmable Logic Controller (Pic) 0 9 CONTROL 8 bytes SecCO2 OUT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROLLER 0 0	256 bytes S6C080 Communication 15616 bytes S6C020 IN HANDSHAKE CHANNEL 16 BITS Y-S6C082,0,16 K-S6C082,0,16 K-S6C082,0,16 IO-DEVICE Programmable Logic Controller (PI 67 9 CONTROL 8 bytes S6C02 OUT (Hoat System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0
Handshake Bit Block 1 Channel Ype Size of Channel Channel Start Address Block 2 Channel Start Address Block 2 Channel Start Address Position of Handshake Cells Size of Handshake Register Host Handshake Register Host Handshake Register Communication Class Protocol Class Conformance Class Number of Subblocks Number of Subblocks Size Start Offset Transfer Direction Transfer Direction Transfer Direction Transfer Direction Size Sust Mode Handshake Mode Handshake Mode	DPM (Dual-Port Memory) UNINNOWN 5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS 55C020 J.0.16 X.56C082,0.16 X.56C082,0.16 Managing Node 0 9 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 5 CONTROL 8 bytes 5 CONTROL 8 C	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56 c080 Communication 7424 bytes 56 c000 Minimum Sector 10 N FANDSHAKE CHANNEL 16 BITS 7426 bytes 56 c002 11 N FANDSHAKE CHANNEL 16 BITS 74 SCC022,016 54 AVE Managing Node 0 9 9 CONTROL 8 bytes 56 c02 CONTROL 8 bytes 56 c02 CONTROL 76 c02 CONTROL	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 NIN HANDSHAKE CHANNEL 158175 1	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 7424 bytes 55C080 Communication 7424 bytes 55C020 IN FANDSHAKE CHANNEL 16 FANDSHAKE CHANNEL 17 SG082.0.16 SAVE 55C02 0 CONTROL 8 bytes 55C02 0 CONTROL 8 bytes 55C02 0 CONTROL 9 CONTROL 0 COMMON STATUS	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 15616 bytes 55C020 N HANDSHAKE CHANNEL 16 AITS 55C020 N HANDSHAKE CHANNEL 16 AITS 55C020 0 AITS 55C020 9 CONTROL 8 bytes 55C020 9 CONTROL 8 bytes 55C020 0 UT (Host System to netX) DPM (Dul-Pot Memory) UNKONTROLED 0 COMMON STATUS	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 ANDSHAKE CHANNEL 16 ANDSHAKE CHANNEL 17 SG0020.016 SLAVE Scanner 0 9 CONTROL 8 bytes 56C0C2 OUT (Hoat-System to netX) DPM (DueTo Memory) UNCONTROLED 0 COMMON STATUS	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 7424 bytes 55C020 IN HANDSHAKE CHANNEL 16 HITS 55C020 IN HANDSHAKE CHANNEL 16 HITS 55C020 0,016 SLAVE Adapter 0 9 CONTROL 8 bytes 55C02 0UT (Host System to netX) DPM (Dual-Fort Memory) UNKCONTROLED 0 COMMON STATUS	5 Handshake 255 bytes 55C080 Communication 15516 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y 55C022,0.16 X 55C022,0.16 MASTRE Io-Controller 0 9 9 CONTROL 8 bytes 55C022 Off (Host System to netX) DPM (Dual-For Kemory) UNCONTROLED 0 COMMON STATUS	5 Handshake 256 bytes 256 bytes 56 CORO 156 Ib bytes 56 CORO 18 HARDSHAKE CHANNEL 16 BITS V:SEC022,0.16 XLAVE 54 CONTROL 54 AVE 9 9 CONTROL 8 bytes 56 COC2 OUT (Host System to metX) DPM (Dual-For Memory) UNCONTROLED 0 COMMON STATUS	5 Handshake 256 bytes 356 bytes 356 bytes 560 CO0 IN HANDSHAKE CHANNEL 15 BITS V*550 20,16 X*500 20,16 ScANNER b Bewice 0 9 9 CONTROL B bytes 560 CO2 OUT (Hock System to netX) DPM (Dual-For Memory) UNCONTROLED 0 COMMON STATUS	5 Handshake 256 bytes 56 bytes 56 bytes 56 correlation 15616 bytes 56 correlation 15616 bytes 56 correlation 16 bits 16 bits 15 bits 16 bits 1	5 Handshake 256 bytes 256 bytes 256 bytes 56CC00 Communication 156 16 bytes 56CC02 NI HANDSHAKE CHANNEL 16 BITS Y35C020,016 X556C02,016 MIESSA600 OUTIROL 8 bytes 56C02 OUTIROL 8 bytes 56C02 OUTIROL 8 bytes 56C02 OUTIROL 9 COMMON STATUS	S Handshake 256 bytes 256 bytes 55CC00 Communication 15616 bytes 55CC00 IN HANDSHAKE CHANNEL 16 BITS Y-55C082.0,16 X-55C082.0,16 C-ONTROLER Programmable Logic Controller (Pic) 0 9 0 CONTROL 8 bytes 55CC02 OUT (Host System to netX) DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS	256 bytes SEC080 Communication 15616 bytes SEC020 IN HANDSHARE CHANNEL 16 BITS Y:SEC020,0.16 K:SEC020,0.16 K:SEC020,0.16 K:SEC020,0.16 CONTROL 8 bytes SEC02 OUT (Host System to netX) OPM (Unai-Port Memory) UNCONTROLLED 0 COMMON STATUS
Handshake Bit Biok 1 Channel Type Sike of Channel Channel Start Address Biok 2 Channel Type Sike of Channel Channel Start Address Destino of Handshake Cells Sike of Handshake Cells Destino nor Handshake Cells Channel Start Address Protocol Class Communication Class Protocol Class Comformance Class Number of Sublocks Number of Sublocks Sike Sike Sike Sike Mandshake Mode Handshake Bit Sike Sike Sike Sike Sike Sike Sike Sike	DPM (Dual-Port Memory) UNINNOWN 5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C080 NH HANDSHAKE CHANNEL 16 BITS 7:56C082,0,16 X:56C082,0,16 X:56C082,0,16 Managing Node 0 9 CONTROL 8 bytes 56CCC 0CONTROL 8 bytes 56CCC 0 CONTROL 8 bytes 56CCC 0 CONTROL 8 bytes 56CCC 0 CONTROL 8 bytes 56CCC	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 COMMUNICATION 16 BitS 17-56C082,0,16 51AVE Managing Node 0 9 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 COMMON STATUS 64 bytes 56C02	Unkrklown 5 Handshake 256 bytes 56000 Communication 156 li6 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS V156008,0.16 X155008,0.16 X55008,0.16X55008,0.16 X55008,0.16X55008,0.16 X55008,0.16X55008,0.16 X55008,0.16X55008,0.16	UNIXNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 Score 0 9 Control Case 5425 Server 0 9 Control 8 bytes 56C02	UNIKNOWN 5 Handhake 256 bytes 55C080 5 Commiscalion 155/16 bytes 55C020 NI HANDSHAKE CHANNEL 16 BITS V*55C082.0.16 X*55C082.0.16 X*55C082.0.16 MASTER 0 9 CONTROL 8 bytes 5 SC0C0 Bit bytes 5 SC0C0 COMMON STATUS 64 bytes 5 SC0C4	UNKNOWN 5 Handbale 256 bytes 56C080 Convincientian 6 Birls Y-SEC082.0.16 SLAVE Scanner 0 9 CONTROL E bytes 56C02 0 9 CONTROL E bytes 56C02 COMMON STATUS 64 bytes 56C02	UNIKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 16 BITS V*56C082.0.16 X*56C082.0.16 SLAVE 0 9 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROL 9	5 Handshake 255 bytes 55C080 Communication 15516 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS V*55C022,0.16 MASTRE IC-Controller 0 9 CONTROL 8 bytes 55C0C2 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes 55C0C4	5 Handshake 256 bytes 36000 Communication 15616 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS V:SEC022,0.16 SLAVE SCE022,0.16 SLAVE CONTROL Bytes 56000 0 CONTROL Bytes 56000 CONTROL CONTROLED 0 CONTROLED	5 Handshake 256 bytes 36C080 Communication 15616 bytes 36C0C0 IN HANDSHAKE CHANNEL 16 BITS V*SC022,0,16 SCNNER IS IS CONTROL BYtes 36C002 CONTROL Bytes 36C02 CONTROL BYtes 36C02 CONTROLED 0 CONTROLED	5 Handshake 256 bytes 56 c080 Communication 15616 bytes 56 c020 NI HANDSHAKE CHANNEL 16 BITS Y-56 c082 0,16 ADAPTER Ico Device 0 9 CONTROL 8 bytes 56 c02 CONTROL 8 bytes 56 c02 CONTROLED 0 COMMON STATUS 64 bytes 56 c02	5 Handshake 256 bytes 256 bytes 256 bytes 56000 Communication 156 16 bytes 56000 NI HANDSHAKE CHANNEL 16 BITS V356020,016 X556020,016 X556020,016 MIESSA600 0 OUTROL 8 bytes 56000 0 COMMON STATUS 64 bytes 56000 0 COMMON STATUS 64 bytes 56000	5 Handshake 256 bytes 256 bytes 256 bytes 556C02 Communication 15616 bytes 556C02 IN HANDSHAKE CHANNEL 16 BITS Y-56C082,0,16 X-56C082,0,16 IO-CONTROLIER Programmable Logic Controller (Pic) 0 9 CONTROL 8 bytes 556C02 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 556C02	256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHARE CHANNEL 16 BITS Y:56C082,0,16 K:56C082,0,16 K:56C082,0,16 K:56C082,0,16 K:56C082,0,16 K:56C082,0,16 CONTROL 8 bytes 56C0C2 OUT (Host System to netX) OPM (Unai-Nert Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 56C0C4
Handshake Bit Biock 1 Channel Yupe Size of Channel Channel Yath Address Biock 2 Channel Yupe Size of Channel Channel Yupe Size of Channel Channel Yupe Size of Channel Channel Yupe Size of Handshake Cells Net X Handshake Register Oommunication Class Conformance Class Conformance Class Conformance Class Conformance Class Conformance Class Conformance Class Size Of Subblocks	DPM (Dual-Port Memory) UNIXOWN 5 	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 2424 bytes 55C000 IN HANDSHAKE CHANNEL 16 BATS 55C000 IN HANDSHAKE CHANNEL 16 BATS 55C000 IN HANDSHAKE CHANNEL 16 BATS 55C000 9 CONTROL 8 bytes 55C000 9 CONTROL 8 bytes 55C000 0 CONTROL 8 bytes 55C000 0 CONTROL 55C000 0 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTRO	UNKKOWN 5 Nandshake 256 bytes 56C080 Communication 15616 bytes 56C020 N NANDSHAKE CHANNEL 15 8175 56C020 0 9 CONTROL 8 bytes 56C022 0 9 CONTROL 8 bytes 56C022 0 0 CONTROL 8 bytes 56C023 0 0 CONTROL 8 bytes 56C024 0 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 76	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 BIT 56 BIT 56 BIT 56 Server 0 9 COMTROL 8 bytes 56C02 0 UNCONTROL 8 bytes 56C02 0 COMTROL 8 bytes 56C02 0 COMTROL 50	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 155616 bytes 56C020 N NAMDSHAKE CHANNEL 16 BITS 56C020 N NAMDSHAKE CHANNEL 16 BITS 56C020 Scanner 0 9 CONTROL 8 bytes 56C021 OUT (Hots System to netX) DPM (Daul-Port Memory) UNKOWTROLED 0 COMMON STATUS 64 bytes 56C024 N NetX Lb host System)	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS 56C020 IN HANDSHAKE CHANNEL 16 BITS 56C020 Scanner 0 9 CONTROL 8 bytes 56C021 OUT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes 56C024 IN (netX to host System)	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C082,0,16 X59C082,0,16 X59	5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C080 Communication 15616 bytes 56C020 Communication 15617 Sec0220 Communication 15617 Sec022 Communication Common Common Sec022 Common	5 Handshake 256 bytes 56C080 Gommunication 56C080 Gommunication 16 Handshake CHANNEL 16 HANDSHAKE CHANNEL 16 HANDSHAKE CHANNEL 16 SCC02,0,16 SLAVE 16-Controller 0 9 CONTROL 8 bytes 56C02 0 CONTROL 9 CONTROL	5 Handshake 256 bytes 56C080 Gommunication 56200 Gommunication 56200 NHANDSHAKE CHANNEL 15 BITS Y56C082.0.16 X55C082.0.16 SCANNER 15-Device 0 5 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL	5 Handshake 256 bytes 56C080 Communication 55C080 Communication 1540 55C00 N HANDSHAKE CHANNEL 155C082.0.16 ADAPTER 10-Device 0 5 CONTROL 8 bytes 56C0C2 0 CONTROL 8 bytes 56C0C2 0 COMMON STATUS 64 bytes 55C0C4 N (next to heat System)	5 Handshake 256 bytes 56C080 Communication 15000 Communication 15000 15000 Socologic S	S Handshake 255 bytes S5C080 Communication S56080 Communication S56080 HANDSHAKE CHANNEL IB BITS Y-S5C082.0.16 IO-CONTROLLER Programmabile Logic Controller (Pic) 0 SCONTROL Bbytes SCOC2 CONTROL Bbytes SCOC4 COMMON STATUS 64 bytes SSCOC4 IN (net Xto Host System)	256 bytes S6C080 Communication 15616 bytes S6C020 IN HANDSHAKE CHANNEL 16 BITS Y-S6C082,0,16 K-S5C082,0
Handshake Bit Biock 1 Channel Ype Size of Channel Channel Start Address Biock 2 Channel Start Address Biock 2 Channel Start Address Size of Channel Channel Start Address Protocol Class Communication Class Protocol Class Conformance Class Number of Subblocks Size Size Size Size Size Size Size Size	DPM (Dual-Pott Memory) UNINGWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 NH HANDSHAKE CHANNEL 16 BITS Y-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 Maraging Node 0 9 CONTROL 8 bytes 56C0C2 0 CONTROL 8 bytes 56C0C2 0 CONMON STATUS 6 d bytes 56C0C4 N (netX to Host System) DPM (Dual-Port Memory)	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56 core Communication 7424 bytes 56 core 10 HANDSHAKE CHANNEL 16 BITS 7426 bytes 56 core 11 HANDSHAKE CHANNEL 16 BITS 745 core 20,16 X-56 core 20,16 StAVE 0 9 CONTROL 8 bytes 56 core 0 9 CONTROL 8 bytes 56 core 0 CONTROL 8 bytes 56 core 0 CONTROL 8 bytes 56 core 0 CONTROL 8 bytes 56 core 0 CONTROL 8 bytes 56 core 0 CONTROL 8 bytes 56 core 0 CONTROL STATUS 64 bytes 56 core 1 N (Host System) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 54 bytes 56 core 1 N (Host System) DPM (Dual-Port Memory) DPM (Dual-Port Memory) DPM (Dual-Port Memory) DPM (Dual-Port Memory)	UNKKOWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C080 Wit HANOSHAKE CHANNEL 16 BITS 19 SEC082,0,16 MASTER 5 Server 0 9 CONTROL 8 bytes 5 SEC02 0 CONTROL 8 bytes 5 SEC02 0 CONTROL 8 bytes 5 SEC02 0 COMMON STATUS 64 bytes 5 SEC02 0 0 COMMON STATUS 64 bytes 5 SEC02 0 COMMON STATUS 7 SEC02 0 COMMON STATUS 7 SEC02 0 COMMON STATUS 7 SEC02 0 SEC02 0 SEC02 0 SEC02 0 SEC02 0 SEC02 0 SEC	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7/284 bytes 56C080 Communication 7/284 bytes 56C080 Server 0 9 CONTROL 8 bytes 56C02 0 9 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 CONTROL 9 CONTROL 8 bytes 56C02 CONTROL 9	UNIKNOWN 5 Handshake 256 bytes 56C080 Communication 15615 bytes 56C080 NH HANDSRAKE CHANNEL 16 1815 YS5C082.0.16 XS5C	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C082.0.16 X/56C082.0.16 SLAVE 56A00 9 CONTROL 8 bytes 56C02 0 0 9 CONTROL 8 bytes 56C02 COMMON STATUS 64 bytes 56C02 IN (neX1X biot System) DPM (Dual-Port Memory) EXECUTION	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Second N HANDSHAKE CHANNEL 16 18 175 YS5C082,0,16 SLAVE 0 9 CONTROL 8 bytes 56C02 001/Flocts System to metX) DPM (Dual-Fort Memory) 0 CONTROL 6 bytes 56C02 0 COMMON STATUS 6 bytes 56C02 N (netX to host System) DPM (Dual-Fort Memory)	5 Handshake 256 bytes 56C080 Communication 15516 bytes 56C00 IN HANDSHAKE CHANNEL 16 BITS Y*56C022,0.16 MASTER Io-Controller 0 9 CONTROL bytes 56C0C2 OUT (Host System to netX) DeM (Dual-For Memory) UKCONTROLLED 0 COMMON STATUS 64 bytes 55C0C4 IN (netX to Host System) DeM (Dual-For Memory)	5 Handshake 256 bytes 56C00 Communication 15616 bytes 56C00 IN HANDSHAKE CHANNEL 16 BITS Y*56C022,0.16 SLAVE Io-Controller 0 9 CONTROL bbytes 56C02 OUT (Host System to netX) DPM (Dual-Fort Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 55C02 IN InetX to host System) DPM (Dual-Fort Memory) IN HANDSHAKE IN INEX to host System) DPM (Dual-Fort Memory) IN INEX to host System) IN I	5 Handshake 256 bytes 36000 Communication 15616 bytes 36000 IN HANDSHARE CHANNEL 16 BITS Y*50022,0.16 SCANNER bo-Device 0 9 CONTROL bo-Device 0 9 CONTROL bo-Device 0 CONTROL bo-Device CONT	5 Handshake 256 bytes 56 COB0 Communication 156 16 bytes 56 COC0 IN HANDSHAKE CHANNEL 16 BITS 7, 56 COB2,0,16 ADAPTER 16 BTS 7, 56 COB2,0,16 ADAPTER 16 Device 0 9 CONTROL 8 bytes 56 COC2 OUT (Host System to netX) OPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 56 COC4 IN (netX to host System) DPM (Dual-Port Memory) EXECUTED 16 COMMON STATUS 16 bytes 17 COMMON STATUS 15 COC4 18 (netX to host System) 17 CPM (Dual-Port Memory) 18 COC4 18 (netX to host System) 17 CPM (Dual-Port Memory) 17 CPM (Dual-Port Memory) 18 COC4 18 (netX to host System) 17 CPM (Dual-Port Memory) 18 CPM (Dual-Port Memory) 18 CPM (Dual-Port Memory) 18 CPM (Dual-Port Memory) 19 CPM (Dual-Port Memory) 10 CP	S Handshake 256 bytes 256 bytes 56000 Communication 15616 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS Y-56008,0.16 MISSARION MISSARION Ombination Firmware 0 9 COMTROL 8 bytes 56000 0 COMMON STATUS 64 bytes 56000 0 COM 64 bytes 56000 COM 64 byte 56000 0 COM 64 bytes 56000 COM 64 bytes 5600	5 Handshake 256 bytes 256 bytes 55CC00 Communication 15616 bytes 55CC00 IN HANDSHAKE CHANNEL 16 BITS Y-55C082,0,16 ID-CONTROLER Programmable Logic Controller (PIc) 0 9 CONTROL 8 bytes 55CC02 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 55CC2 IN (netX to Host System) DPM (Dual-Port Memory) IN (netX to Host System) DPM (Dual-Port Memory) IN (netX to Host System) IN (netX to Ho	256 bytes SEC080 Communication 15616 bytes SEC020 IN HANDSHAKE CHANNEL 16 BITS Y-SEC020,0.16 IO-DEVICE FORGER O-DEVICE CONTROL Bytes SEC02 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes SEC02 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 0 COMMON STATUS 64 bytes SEC02 IN (netX to Host System) DPM (Dual-Port Memory) DPM (Dual-Port Memory) DPM (Dual-Port Memory) DPM (Dual-Port Memory)
Handshake Bit Block 1 Channel Ype Size of Channel Channel Start Address Block 2 Channel Ype Size of Channel Channel Ype Size of Channel Channel Start Address Block 2 Channel Ype Size of Mandshake Register Communication Class Conformance Class Size Of Stubblocks	DPM (Dual-Port Memory) UNIXOWN 5 	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 2424 bytes 55C000 IN ANDSHAKE CHANNEL 16 ANDS 55C000 IN HANDSHAKE CHANNEL 16 ANDS 55C000 9 CONTROL 8 bytes 55C000 9 CONTROL 8 bytes 55C000 0 CONTROL 8 bytes 55C000 0 CONTROL 55C000 0 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL 7 CONTROL	UNKKOWN 5 Nandshake 256 bytes 56C080 Communication 15616 bytes 56C020 N NANDSHAKE CHANNEL 15 8175 56C020 0 9 CONTROL 8 bytes 56C022 0 9 CONTROL 8 bytes 56C022 0 0 CONTROL 8 bytes 56C023 0 0 CONTROL 8 bytes 56C024 0 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 8 bytes 56C024 0 CONTROL 76	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 BIT 56 BIT 56 BIT 56 Server 0 9 COMTROL 8 bytes 56C02 0 UNCONTROL 8 bytes 56C02 0 COMTROL 8 bytes 56C02 0 COMTROL 50	UNKNOWN 5 Handshake 256 bytes 55C080 Communication 155616 bytes 56C020 N NAMDSHAKE CHANNEL 16 BITS 56C020 N NAMDSHAKE CHANNEL 16 BITS 56C020 Scanner 0 9 CONTROL 8 bytes 56C021 OUT (Hots System to netX) DPM (Daul-Port Memory) UNKOWTROLED 0 COMMON STATUS 64 bytes 56C024 N NetX Lb host System)	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS 56C020 IN HANDSHAKE CHANNEL 16 BITS 56C020 Scanner 0 9 CONTROL 8 bytes 56C021 OUT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes 56C024 IN (netX to host System)	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 IN HANDSHAKE CHANNEL 16 BI7 6 S6C02 IN HANDSHAKE CHANNEL 16 BI7 6 S6C02 IN HANDSHAKE CHANNEL 1 B bytes 56C02 OUT (Hot System to netX) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes 56C02 IN (NetX to hot System)	5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C080 Communication 15616 bytes 56C020 Communication 15617 Sec0220 Communication 15617 Sec022 Communication Common Common Sec022 Common	5 Handshake 256 bytes 56C080 Gommunication 56C080 Gommunication 16 HaNDSHAKE CHANNEL 16 HANDSHAKE CHANNEL 16 HANDSHAKE CHANNEL 16 SCO02,0,16 SLAVE 16-Controller 0 9 CONTROL 8 bytes 56C02 0 CONTROL B bytes 56C02 0 COMMON STATUS 64 bytes 55C024 N (next to host System)	5 Handshake 256 bytes 56C080 Gommunication 56200 Gommunication 56200 NHANDSHAKE CHANNEL 15 BITS Y56C082.0.16 X55C082.0.16 SCANNER 15-Device 0 5 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL	5 Handshake 256 bytes 56C080 Communication 55C080 Communication 1540 55C00 N HANDSHAKE CHANNEL 155C082.0.16 ADAPTER 10-Device 0 5 CONTROL 8 bytes 56C0C2 0 CONTROL 8 bytes 56C0C2 0 COMMON STATUS 64 bytes 55C0C4 N (next to heat System)	5 Handshake 256 bytes 56C080 Communication 15000 Communication 15000 15000 Socologic S	S Handshake 255 bytes S5C080 Communication S56080 Communication S56080 HANDSHAKE CHANNEL IB BITS Y-S5C082.0.16 IO-CONTROLLER Programmabile Logic Controller (Pic) 0 SCONTROL Bbytes SCOC2 CONTROL Bbytes SCOC4 COMMON STATUS 64 bytes SSCOC4 IN (net Xto Host System)	256 bytes S6C080 Communication 15616 bytes S6C020 IN HANDSHAKE CHANNEL 16 BITS Y-S6C082,0,16 K-S6C082,0
Handshake Bit Biok 1 Channel Type Size of Channel Channel Start Address Biock 2 Channel Start Address Biock 2 Channel Start Address Position of Handshaelie HeiX Handshake Register HeiX Handshake Register HeiX Handshake Register Start Offset Start Offset Transfer Direction Transfer Direction Size Size Size Size Size Size Size Size	DPM (Dual-Port Memory) UNINNOWN 5 Handshake 256 bytes 56C008 Communication 155116 bytes 56C00 16 HANDSHAKE (CHANNEL 16 BTS 755C082,0,16 X-55C0	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 55C00 N HANDSHAKE CHANNEL 155 SEC082 N HANDSHAKE CHANNEL 155 SEC082 N HANDSHAKE CHANNEL 155 SEC082 N HANDSHAKE CHANNEL 155 SEC082 0 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 8 CONTROL 9 CONTROL 8 CONTROL 8 CONTROL 9 CONTROL 8 CONTROL 8 CONTROL 8 CONTROL 8 CONTROL 8	UNKNOWN 5 Randshake 256 bytes 56C080 Communication 156156 bytes 56C020 NIN HANDSHAKE CHANNEL 156175 56C020 0 NIN HANDSHAKE CHANNEL 156175 Server 0 9 CONTROL 8 bytes 56C02 00/// (Hot System to netX) DMM (Dual-Port Memory) UNKONTROLLED 0 COMMON STATUS 56 bytes 56C02 0 COMMON STATUS 56 bytes 56 bytes	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 Server 0 9 COMTROL 8 bytes 56C082.0.16 SkAVE 9 CONTROL 9 COMTROL 9 COMMON STATUS 64 bytes 55C024 0 COMMON STATUS 55C02 55C02 55C02 55C02 55C02 55C02 55C0 55C0	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 156165 bytes 56C020 N NANDSHARE CHANNEL 16 BITS 56C020 N NANDSHARE CHANNEL 16 BITS 56C020 9 CONTROL 8 bytes 56C02 00F (Hot System to netX) DMM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 56C024 0 COMMON STATUS 56C024 0 COMMON STATUS 57C024 0 COMMON STATUS 57C024 0 COMM	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 SGC00 IN HANDSHAKE CHANNEL 16 BITS 56C020 IN HANDSHAKE CHANNEL 16 BITS SGC020 SCANNEL 5 Scanner 0 9 COMMON STATUS 6 Bit Status 5 SCO2 OUTFIOLED 0 COMMON STATUS 6 Bit Status 5 SCO2 0 COMMON STATUS 6 Bit Status 6 Bit Status 6 Bit Status 7 COMMON STATUS 6 Bit Status 8 SCO2 0 COMMON STATUS 6 Bit Status 9 COMMON STATUS 6 Bit Status 8 SCO2 0 COMMON STATUS 6 Bit Status 8 SCO2 0 COMMON STATUS 6 Bit Status 8 SCO2 0 COMMON STATUS 6 Bit Status 8 SCO2 0 COMMON STATUS 6 Bit Status 9 COMMON STATUS 6 Bit Status 8 SCO2 0 COMMON STATUS 8 SCO2 0 COMMON STATUS 8 SCO2 0 COMMON STATUS 8 SCO2 0 COMMON STATUS 8 SCO2 0 COMMON STATUS 8 SCO2 0 COMMON STATUS 8 SCO2 0 COMMON STATUS 8 SCO2 SCO2 SCO3 SCO2 SCO3 SCO2 SCO3 SCO3 SCO3 SCO3 SCO3 SCO3 SCO3 SCO3	UNIKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 INIANDSHAKE CHANNEL 16 BITS 56C082,0,16 5 SLAVE Adapter 0 9 CONTROL 8 bytes 56C02 00/I (HotStystem to netX) DMI (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C024 0 COMMON STATUS 56 bytes 56C024 0 COMMON STATUS 56 bytes 56 byt	5 Handshake 25 bytes 56C080 Communication 15516 bytes 56C020 Handson Second 56C020 Handson Second Handson Hand	5 Handshake 256 bytes 56C080 Communication 15516 bytes 56C020 S5C020 His BITS 95C022 0,16 X 55C022 0,16 X 55C022 0,16 X 55C022 0,16 S X 55C02 0,17 S X 55C02	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 History 15617 Scores 56C020 History 15817 NSG0282,0,16 Scores 16-Device 0 9 CONTROL 8 bytes 56C02 0UT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 56 bytes 56C04 NI (netX to host System) DPM (Dual-Port Memory) UNCONTROLLED 0	5 Handshake 25 bytes 56 C080 Communication 15616 bytes 56 C080 Communication 15616 56 C080 156175 Y-56 C082.0.16 ADAPTER 4:- Device 0 9 CONTROL 8 bytes 56 C022 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 55 C024 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 0 COMMON STATUS 55 C024 0 COMMON STATUS 55 C024 0 COMMO	S Handshake 256000 S6000 NI MADA S6000 NI MA	S Handshake 256 bytes SGC000 His HTS SGC000 HHAKE CHANNEL 15516 bytes SGC000 His HTS YSC082.0.16 HO-CONTROLLER PO-CONTROLLER PO-CONTROLLER PO-CONTROLLER PO-CONTROLLER Bytes SGC02 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS SSC02 Bytes SSC02 DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS SSC02 Bytes Status SSC02 DPM (Dual-Port Memory) UNCONTROLLED DPM (Dual-Port Memory) UNCONTROLLED D	256 bytes SECOBO Communication 15616 bytes SeCOCO INI HANDSHAKE CHANNEL 16 BITS +*SECOSO_16 Xx050020C_150 CONTROL 8 bytes SECOC2 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 66 bytes SECOC2 0 COMMON STATUS 66 bytes SECOC2 0 COMMON STATUS 66 bytes SECOC2 0 CONTROLLED 0 CONTROLLED DPM (Dual-Port Memory) UNCONTROLLED 0 0
Handshake Bit Block 1 Channel Ype Size of Channel Channel Start Address Block 2 Channel Start Address Block 2 Size of Channel Start Address Comformance Class Conformance Class Start Offset Transfer Direction Transfer Mode	DPM (Dual-Port Memory) UNIXNOWN 5 256 bytes 56C08 Communication 15616 bytes 56C00 RCOC0 RC	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56 core Communication 7424 bytes 56 core 10 HANDSHAKE CHANNEL 16 BITS 7426 bytes 56 core 11 HANDSHAKE CHANNEL 16 BITS 745 core 20,16 X-56 core 20,16 Stave 0 9 CONTROL 8 bytes 56 core 0 9 CONTROL 8 bytes 56 core 0 CONTROL 8 bytes 56 core 0 CONTROL STATUS 56 bytes 56 core 0 CONTROLED 0 CONMON STATUS 56 bytes 56 core 0 CONTROLED 0 CONMON STATUS 56 bytes 56 core 0 CONTROLED CONTROLED 0 CONTROLED CONTROLED CO	UNKKOWN 5 Handbake 256 bytes 56C080 Communication 15616 bytes 56C000 IN IAADSHAKE CHANNEL 15 8175 56CC00 NI NAADSHAKE CHANNEL 15 8175 56CC02 0 9 CONTROL 8 bytes 56CC02 0 0 CONTROL 8 bytes 56CC04 0 0 CONTROL 8 bytes 56CC04 0 0 CONTROL 8 bytes 56CC04 0 0 CONTROL 8 bytes 56CC04 0 0 CONTROL 8 bytes 56CC04 0 0 CONTROL 8 bytes 56CC04 0 0 CONTROL 8 bytes 56CC04 0 0 CONTROL 8 bytes 56CC04 0 CONTROL 8 bytes 56CC04 0 CONTROL 8 CO	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C020 N HANDSHAKE CHANNEL 16 BITS 56C02 Server 0 9 CONTROL 8 bytes 56C02 0 CONTROL 9 CONT	UNIKNOWN 5 Handshake 256 bytes 56C080 Communication 155616 bytes 56C080 Communication 156815 56C020 N HANDSHAKE CHANNEL 16 BITS 56C020 N HANDSHAKE CHANNEL 16 BITS Sconner 0 9 CONTROL 8 bytes 56C021 OUT (Hock System to netX) DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C024 UNICONTROLLED	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS 56C020 Scanner 0 9 CONTROL 8 bytes 56C021 OUT (Hock System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 56C024 UNCONTROLLED	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 Kistore Communication 15 Birs Scoce Scoce Scoce Scoce Scoce Scoce Scoce Scoce Scoce Common Status Scoce	5 Handshake 255 bytes 56C080 Communication 56E080 Communication 15 Bits 56C080 NHANDSHAKE CHANNEL 15 Bits 56C082,0,16 MASTER 16-Controller 0 9 CONTROL 8 bytes 56C022 0 COMMON STATUS 64 bytes 55C024 NH Intext Kb Into System) DPM [Dual-Port Memory] UNCONTROLLED	5 Handshake 256 bytes 56C080 Gommunication 56C00 Gommunication 56C00 IN HANDSHAKE CHANNEL 15 BITS 56C02 S5C02 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROLED 0 COMMON STATUS 64 bytes 55CC04 IN Intext to host System) DPM (Dual-Port Memory) DPM (Dual-Port Memory) DPM (Dual-Port Memory) DM (Dual-Port Mem	5 Handshake 256 bytes 56C080 Gommunication 56D0 Gommunication 56D0 Kontex 56C080 Kontex 56D0 Kontex 56	5 Handshake 256 bytes 56C080 Communication 56C080 Communication 15 arts 55C00 N NANDSHAKE CHANNEL 15 arts 55C02 N NANDSHAKE CHANNEL 15 arts 55C02 0 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROLEB 0 CO	5 Handshake 256 bytes 56C080 Communication 56C000 III HANDSHAKE CHANNEL 158 ITS 55C020 III HANDSHAKE CHANNEL 158 ITS 755C020,016 MS56C020,016 MS56C020,016 MS56C020,016 COMMON STATUS 64 bytes 55C0C4 III (Inext System) DPM (Usal-Port Memory) UNCONTROLLED	S Handshake 255 bytes S5C020 Communication S560C0 IN HANDSHAKE CANNEL 16 BITS Y-SEC082.0.16 IC-CONTROLLER Programmable Logic Controller (PIc) 0 SCONTROL 8 bytes SC0C2 CONTROL 8 bytes SC0C2 COMMON STATUS 64 bytes SSC0C4 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED	256 bytes S6C080 Communication 15616 bytes S6C020 IN HANDSHAKE CHANNEL 16 BITS Y:S6C082,0,16 IV:S6C082
Handshake Bit Biock 1 Channel Ype Size of Channel Channel Start Address Biock 2 Channel Start Address Biock 2 Channel Start Address Biock 2 Channel Start Address Channel Start Address Channel Start Address Channel Start Address Communication Class Protocol Class Communication Class Protocol Class Conformance Class Number of Subblocks	DPM (Dual-Port Memory) UNINGWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C080 NH HANDSHARE CHANNEL 16 BITS Y -56C082,0,16 X -56C082,0,16 X -56C082,0,16 Managing Node 0 9 CONTROL 8 bytes 56C02 OUT (Host System to netX) DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C02 DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C02 DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C02 DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C02 DPM (Dual-Port Memory) UNICONTROLLED 0 EXTENDED STATUS 432 bytes 56C02	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56 commication 7424 bytes 56 core 10 N HANDSHARE CHANNEL 16 BITS 7424 bytes 56 core 11 N HANDSHARE CHANNEL 16 BITS 7426 core 7426 cor	UNKKNOWN 5 Handbake 256 bytes 55C080 Communication 1568 bytes 56C080 Communication 1568 bytes 56C080 N NATS PARC CHANNEL 1567 COMMON STATUS 8 bytes 56C02 OUT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROL 8 bytes 56C02 OUT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 65C02 N (NetX to Hots System) DPM (Dual-Port Memory) UNCONTROLED 0 EXTENDED STATUS 422 bytes 55C02	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 0 9 CONTROL 8 bytes 56C02 0 9 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C04 EXTENDED 51ATUS 432 bytes 56C04	UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 155616 bytes 56C080 Communication 155616 bytes 56C020 1556020,0.16 WASTER Science 0 9 CONTROL 8 bytes 56C020 0 CONTROL 8 bytes 56C020 CONTROL 8 bytes 56C020 CONT	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C08 Communication 7424 25 Communication Communica	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 72424 bytes 56C080 Communication 74244 bytes 56C080 SGC00	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C0C0 IN HANDSHARC CHANNEL 16 BITS Y*56C022,0,16 MASTER Io-Controller 0 9 CONTROL Bytes 56C02 OUT (Joint System to netX) DPM (Dual-For Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 56C0C4 IN (netX to Host System) DPM (Dual-For Memory) UNCONTROLLED 0 EXTENDED STATUS 432 bytes 56C04	5 Handshake 256 bytes 256 bytes 256 bytes 56C00 Communication 15616 bytes 56C00 IN HANDSHAKE CHANNEL 16 BITS Y*56C020,0.16 SLAVE I6 Controller 0 9 CONTROL 66 bytes 56C02 OUT (Host System to netX) DPM (Dual-Fort Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 55CC04 IN InetX to Host System) DPM (Dual-Fort Memory) UNCONTROLLED 0 EXTENDED STATUS 432 bytes 55CC04	5 Handshake 256 bytes 256 bytes 256 bytes 56000 Communication 15616 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS Y*56082,0.16 SCANNEK IG-Device 0 9 CONTROL BATCS SCANNEK SCANNEK G-Device 0 CONTROL G-Bytes 5602 CONTROL BATCS SCANNE COMMON STATUS G-Bytes 55000 CONTROLED 0 COMMON STATUS G-Bytes 55000 EXTENDED STATUS G-Bytes 56000 COMMON STATUS G-Bytes 55000 COM	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE (HANNEL 16 BITS Y+56C082,0,16 X-56C082,0,16 ADAPTEN Ic-Device 0 9 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 6 bytes 56C02 IN (next to host System) DPM (Dual-For Memory) UNCONTROLLED 0 EXTENDED STATUS 64 bytes 56C02	S Handshake 256 bytes 256 bytes 256 communication 15616 bytes 56CC0 IN HANDSHAKE CHANNEL 16 BITS Y56C080,016 X556C08,0,16 MESARION MESARION MESARION 0 9 CONTROL 8507 COmbination Firmware 0 9 CONTROL 8507 CONTROL 8507 CONTROL 9	S Handshake 256 bytes ScCO80 Communication 15616 bytes ScCC0 IN HANDSHAKE CHANNEL 16 BITS Y-ScC082,0,16 X-ScC082,0,16 IN-CONTROLER Programmable Logic Controller (PIc) 0 9 CONTROL 8 bytes ScC07C16 Stytem to netX) DPM (Dul-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes ScC074 IN (netX to Host System) DPM (Dul-Port Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes ScC074	256 bytes 56C080 Communication 15516 bytes 56C020 IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL Frogrammable Logic Controller (P 67 9 CONTROL 8 bytes 56C02 O COMMON STATUS 64 bytes 56C02 0 EXTENDED STATUS 43 bytes 56C02
Handshake Bit Biock 1 Channel Yope Size of Channel Channel Start Address Biock 2 Channel Start Address Biock 2 Channel Start Address Biock 2 Channel Start Address Size of Start Offset Transfer Direction Transfer Direction Transfer Type Handshake Bit Size Offset Handshake Mode Handshake	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 255 bytes 56C009 Communication 15515 bytes 56C009 Control 155 X-58C082.0.16 X-58C	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 56C080 N HANDSHAKE CHANNEL 155C082.0.16 X55C082.0.16 X55C082.0.16 X55C082.0.16 X55C082.0.16 X55C082.0.16 X55C082.0.16 SLAVE Managing Node 0 9 CONTROL 8 bytes 56C0C2 OUT (Host System to netX) DPM (Dual-Port Memory) UNICONTROLED 0 COMMON STATUS 66 bytes 56C0C4 N (netX host System) DPM (Dual-Port Memory) UNICONTROLED 0 EXTENDED STATUS 432 bytes 56C0C4 N (netX host System) DPM (Dual-Port Memory) UNICONTROLED 0 EXTENDED STATUS 432 bytes 56C0C4 N (netX host System) DPM (Dual-Port Memory) UNICONTROLED 0 EXTENDED STATUS 432 bytes 55C0C4	UNKNOWN 5 Handshake 255 bytes 55C080 Communication 155616 bytes 55C080 Communication 155616 bytes 55C020 IN HANDSHAKE CHANNEL 15 8175 SFCVET 0 5 SFCVET 0 5 CONTROL 8 bytes 5 CONTROL 8 bytes 5 CONTROL 9 CONTROL 9 CONTROL 8 bytes 5 CONTROL 9 CONTR	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 S6C020 IN HANDSHAKE CHANNEL 16 BITS S6C020 IN HANDSHAKE CHANNEL 16 BITS Server 0 9 COMTROL Server 0 9 COMTROL Server 0 0 COMMON STATUS 64 bytes 55C0C4 IN (next Ku host System) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes 55C0C4 IN (next Ku host System) DPM (Dual-Port Memory) UNCONTROLED 0 EXTENDED STATUS 422 bytes	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 15616 bytes 56C080 Communication 15616 bytes 56C00 IN HANDSHAKE CHANNEL 16 BITS 56C00 IN HANDSHAKE CHANNEL 16 BITS Scanner 0 9 CONTROL 8 bytes 50T (Hot System to netX) 00T (Hot System) 0 EXTENDED STATUS 42 bytes 55C004 IN (NetX to hot System) 0	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 COMMUNICATION 56C00 16 BTS 56C00 16 BTS 56C00	UNIKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 IN FANDSHAKE CHANNEL 16 BITS SCORE 16 BITS SCORE 16 BITS SLAVE Adapter 0 9 CONTROL 8 bytes 50T (Hot System to netX) 00T (Hot System) 0 CONTROLLED CONTROLLED 0 CONTROLLED CONTROLLED 0 CONTROLLED CONTROL	5 Handshake 255 bytes 56C080 Communication 15516 bytes 56C020 Communication 15516 bytes 56C020 K \$56082,0,16 K \$56	5 Handshake 25 bytes 56C080 Communication 15516 bytes 56C020 Communication 15516 bytes 56C020 Second 15816 Second 15817 Y55C082,0,16 Second 15817 Y55C082,0,16 Second 15817 Second 15817 Second 15817 Second 15817 Second 158 Second 15	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C080 W JANSHAKE CHANNEL 15516 bytes 56C020 W JANSHAKE CHANNEL 15516 555020,016 555020,016 555020,016 0 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 9 CONTROL	5 Handshake 25 bytes 56 CD80 Communication 15616 bytes 56 CD80 Communication 15616 bytes 56 CO20 IN IMM SIAKE CHANNEL 151 55 SG CO20 ADAPTER IN-Device 0 CONTROL 8 bytes 56 CO2 OUT (Not System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 66 bytes 56 CO2 0 COMMON STATUS 67 bytes 56 CO2 0 COMMON STATUS 56 CO2 0 COM 56 bytes 56 CO2	S Handshake 25 bytes 56 cos Communication 156 i6 bytes 56 coco IN HARGSHARE CHANNEL 155 for 156 for 15	S Handshake 255 bytes 55C080 Communication 155ii5 bytes 55C020 IN HAN HARKE CHANNEL 155ii5 bytes 55C020 SCOLO IN HAN HARKE CHANNEL 155ii5 bytes 7528028.0.15 IN CONTROLLER Programmable Logic Controller (PIc) 0 9 CONTROL 8 bytes 55C02 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 55C04 IN (netX to host System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 432 bytes 55C04 IN (netX to host System)	255 bytes 56009 Communication 15516 bytes 56000 IN HANDSHAE CHANNEL III HANDSHAE CHANNEL IIII HANDSHAE CHANNEL IIII HANDSHAE CHANNEL IIII HANDSHAE CHANNEL IIIII HANDSHAE CHANNEL IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
Handshake Bit Biock 1 Channel Yope Channel Yope Size of Channel Channel Yope Size of Channel Channel Start Address Biock 2 Channel Start Address Biock 2 Channel Start Address Channel Start Address Protocol Class Communication Class Protocol Class Conformance Class Number of Subblocks	DPM (Dual-Port Memory) UNINGWN 5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C080 NH HANDSHARE CHANNEL 16 BITS Y -56C082,0,16 X -56C082,0,16 X -56C082,0,16 Managing Node 0 9 CONTROL 8 bytes 56C02 OUT (Host System to netX) DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C02 DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C02 DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C02 DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 56C02 DPM (Dual-Port Memory) UNICONTROLLED 0 EXTENDED STATUS 432 bytes 56C02	DPM (Dual-Port Memory) UNKNOWN 5 Handshake 256 bytes 56 commication 7424 bytes 56 core 10 N HANDSHARE CHANNEL 16 BITS 7424 bytes 56 core 11 N HANDSHARE CHANNEL 16 BITS 7426 core 7426 cor	UNKKNOWN 5 Handbake 256 bytes 55C080 Communication 1568 bytes 56C080 Communication 1568 bytes 56C080 N NATS PARC CHANNEL 1567 COMMON STATUS 8 bytes 56C02 OUT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROL 8 bytes 56C02 OUT (Hots System to netX) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 65C02 N (NetX to Hots System) DPM (Dual-Port Memory) UNCONTROLED 0 EXTENDED STATUS 422 bytes 55C02	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C020 Communication 7424 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 9 C	UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 155616 bytes 56C080 Communication 155616 bytes 56C020 1556020,0.16 WASTER Science 0 9 CONTROL 8 bytes 56C020 0 CONTROL 8 bytes 56C020 CONTROL 8 bytes 56C020 CONT	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 COMMUNICATION 5 COMMONICATION 9 CONTROL 8 bytes 56C02 0 COMMONICATION 0 CO	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 72424 bytes 56C080 Communication 74244 bytes 56C080 SGC00	5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C020 NI NANOSHAKE CHANNEL 15560780.016 X550780.016 X5507	5 Handshake 25 bytes 56C080 Communication 15616 bytes 56C020 IN PARDSHARE CHANNEL 15616 bytes 56C020 S56C020 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROL 9 CONTROL 8 bytes 56C02 CONTROL 9 CONTROLED 0 CON	5 Handshake 256 bytes 256 bytes 256 bytes 56000 Communication 15616 bytes 56000 IN HANDSHAKE CHANNEL 16 BITS Y*56082,0.16 SCANNEK IG-Device 0 9 CONTROL BATCS SCANNEK SCANNEK G-Device 0 CONTROL G-Bytes 5602 CONTROL BATCS SCANNE COMMON STATUS G-Bytes 55000 CONTROLED 0 COMMON STATUS G-Bytes 55000 EXTENDED STATUS G-Bytes 56000 COMMON STATUS G-Bytes 55000 COM	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE (HANNEL 16 BITS Y+56C082,0,16 X-56C082,0,16 ADAPTEN Ic-Device 0 9 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 6 bytes 56C02 IN (next to host System) DPM (Dual-For Memory) UNCONTROLLED 0 EXTENDED STATUS 64 bytes 56C02	S Handshake 256 bytes 256 bytes 256 communication 15616 bytes 56CC0 IN HANDSHAKE CHANNEL 16 BITS Y56C080,016 X556C08,0,16 MESARION MESARION MESARION 0 9 CONTROL 8507 COmbination Firmware 0 9 CONTROL 8507 CONTROL 8507 CONTROL 9	5 Handshake 256 bytes ScCO80 Communication 15616 bytes ScCC0 IN HANDSHAKE CHANNEL 16 BITS Y-ScC082,0,16 X-ScC082,0,16 IN-CONTROLER Programmable Logic Controller (PIc) 0 9 CONTROL 8 bytes ScC07C16 Stytem to netX) DPM (Dui-Nort Memory) UNCONTROLED 0 COMMON STATUS 64 bytes ScC074 IN (netX to Host System) DPM (Dui-Nort Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes ScC074	256 bytes 56C080 Communication 15516 bytes 56C020 IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL IN HANDSHARE CHANNEL Frogrammable Logic Controller (P 67 9 CONTROL 8 bytes 56C02 O COMMON STATUS 64 bytes 56C02 0 EXTENDED STATUS 43 bytes 56C02
Handshake Bit Biok 1 Channel Type Size of Channel Channel Type Size of Channel Channel Start Address Biok 2 Channel Start Address Destituto of Handshake Cells Size of Handshake Cells New Handshake Register Host Handshake Register Communication Class Protocol Class Communication Class Protocol Class Communication Class Protocol Class Conformance Class Number of Subblocks Size Size Of Subblocks Size Size Size Of Subblocks Size Size Size Of Subblocks Size Size Size Size Of Subblocks Size Size Size Size Size Size Size Size	DPM (Dual-Port Memory) UNIXOWN 5 Communication 15616 bytes 55C000 Communication 15616 bytes 55C0C0 III HANDSHAKE CHAINEL 16 8175 Y-\$C082,0,16 X-\$C082,0,16 X-\$C082,0,16 X-\$C082,0,16 X-\$C082,0,16 MASTER Managin Node 0 9 COMTROL E bytes 54C0C2 OUT (Host System to netX) DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 54C0C4 III (Host System) DPM (Dual-Port Memory) UNICONTROLLED 0 E EXTENDED STATUS 4432 bytes 54C0C4 III (Inclassion Common Common DPM (Dual-Port Memory) UNICONTROLLED 0 EXTENDED STATUS 4432 bytes 54C0C4 III (Inclassion Common DPM (Dual-Port Memory) UNICONTROLLED 0 EXTENDED STATUS 4432 bytes 55C0C4 III (Inclassion Common DPM (Dual-Port Memory) UNICONTROLLED 0 EXTENDED STATUS	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 56C060 IN HANDSHAKE CHANNEL 16 BITS 55C062 0 55C062 0 CONTROL 8 bytes 55C062 0 CONTROL 8 bytes 55C064 0 CONTROL 8 bytes 55C064 0	UNKNOWN 5 Handbake 255 bytes 56C080 Communication 1566 bytes 56C080 Communication 1566 bytes 56C080 Communication 1566 bytes 56C080 0 9 CONTROL 8 bytes 56C02 0 0 0 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 56 bytes 56C02 0 0 CONTROL 56 bytes 56C02 0 0 0 CONTROL 57 bytes 56C02 0 0 0 0 0 0 0 0 0 0 0 0 0	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 SGC0 0 9 CONTROL 8 bytes 56C080,016 SLAVE 9 CONTROL 8 bytes 56C02 0 9 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROLED 0 CONT	UNIXNOWN 5 Handbake 256 bytes 56C080 Communication 15616 bytes 56C080 Communication 15616 bytes 56C020 N NANDSHAKE CHANNEL 156175 56C020 0 S CONTROL 8 bytes 56C021 0 S CONTROL 8 bytes 56C020 0 UIT (Idox System to netX) DPM (Dual-Port Memory) UNIXONTROLLED 0 COMMON STATUS 56 bytes 56C024 N IN IREX to Host System) DPM (Dual-Port Memory) UNIXONTROLLED 0 COMMON STATUS 56 bytes 56C024 N IN IREX to Host System) DPM (Dual-Port Memory) UNIXONTROLLED 0 CYTENDED STATUS 42 bytes 56C024 N IN IREX to Host System) DPM (Dual-Port Memory) 0 CYTENDED STATUS 42 bytes 56C04 IN IREX to Host System) DPM (Dual-Port Memory) 0 CYTENDED STATUS 42 bytes 56C04 IN IREX to Host System) DPM (Dual-Port Memory)	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 Status 56C08 Communication 7424 Communication 742 Communication 74 Communication 7	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 72424 bytes 56C080 Communication 7424 bytes 56C080 S6C08 CONTROL 8 bytes 56C02 CONTROL 9 CONTROL	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN HANDSHAKE CHANNEL 16 BITS Y*56C020,016 X×56C02,0,16 MASTER Ic-Controller 0 9 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 0 CONTROL 9	5 Handshake 256 bytes 356 coso Communication 15616 bytes 366 coco IN HANDSHAKE CHANNEL 16 BITS Y*SEG020,16 SLAVE I6 Controller 0 9 CONTROL 8 bytes 50 CONTROL 8 bytes 50 CONTROL 8 bytes 50 CONTROL 9 CONTROLLED 0 0 COMMON STATUS 64 bytes 55CC04 IN IneX to host System) DPM (Dual-For Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes 55C04 IN IneX to host System) DPM (Dual-For Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes 55C04 IN IneX to host System) DPM (Dual-For Memory) UNCONTROLED 0	5 Handshake 256 bytes 356 cos0 Communication 156 l6 bytes 366 coc0 IN HANDSHAKE CHANNEL 16 BITS Y*SEC082.0.16 SCANNER IN HANDSHAKE CHANNEL 16 BITS Y*SEC082.0.16 SCANNER IN HANDSHAKE CHANNEL 16 BUTS SCANNER IN HANDSHAKE CHANNEL 16 BUTS COMMON STATUS 64 bytes 55C0 0 0 COMMON STATUS 64 bytes 55C0 CCMMON STATUS 64 bytes 55C0 IN IN INEX IS HOST System 0 D COMMON STATUS 64 bytes 55C0 IN IN INEX IS HOST System 10 D COMMON STATUS 64 bytes 55C0 IN IN INEX IS HOST System 10 D IN IN IN	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C020 IN NANDSHAKE (HANNEL 16 BITS Y+56C082,0,16 X-56C082,0,16 ADAPTEN Ic-Device 0 9 CONTROL 8 bres 56C02 0 CONTROL 8 bres 56C02 0 CONTROL 16 bytes 56C02 0 EXTENDED STATUS 432 bytes 56C02 IN (next to host System) DPM (Duai-Port Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes 56C02 IN (next to host System) DPM (Duai-Port Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes 56C02 IN (next to host System) DPM (Duai-Port Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes 56C02 IN (next to host System) DPM (Duai-Port Memory) UNCONTROLED 0	S Handshake 256 bytes 256 bytes 256 communication 15616 bytes 56CC00 IN HANDSHAKE CHANNEL 16 BITS Y56C080,016 X556C08,0,16 MESARING Combination Firmware 0 9 CONTROL 8 bytes 9 CONTROL 8 bytes 9 CONTROL 8 bytes 9 CONTROL 9 COMMON STATUS 64 bytes 55CC0 IN (HeX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 432 bytes 55CC04 IN (HeX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 432 bytes 55CC04 IN (HeX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 432 bytes 55CC04 IN (HeX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0	S Handshake 256 bytes ScC080 Communication 15616 bytes ScC020 IN HANDSHAKE CHANNEL 16 BITS Y-ScC082,0,16 X-ScC082,0,16 IN-CONTROLER Programmable Logic Controller (PIc) 0 9 CONTROL 8 bytes ScC022 OUT (Hoad System to netX) DFM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes ScC024 IN (netX to host System) 0 EXTENDED STATUS 432 bytes ScC04 IN (netX to host System) DFM (Dual-Port Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes ScC04 IN (netX to host System) DFM (Dual-Port Memory) UNCONTROLED 0	256 bytes 56C080 Communication 15516 bytes 56C020 IN HANDSHARE (HANNEL 16 BITS 7-5C082,0,16 X-5C082,0,16 X-5C082,0,16 IN O-BEVICE Programmable Logic Controller (I 67 9 CONTROL 8 bytes 56C02 OUT (Host System to netX) DMI (Dusi-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 56C02 0 EXTENDED STATUS 43 2 bytes 56C04 IN (netX to Host System) DPM (Dusi-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 43 2 bytes 56C02 N(netX to Host System) DPM (Dusi-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 43 2 bytes 56C02 IN (netX to Host System) DPM (Dusi-Port Memory) UNCONTROLLED 0 IN (netX to Host System) DPM (Dusi-Port Memory) UNCONTROLLED 0 IN (netX to Host System) DPM (Dusi-Port Memory) IN (PATOR DE TATUS 43 2 bytes 56C004 IN (netX to Host System) DPM (Dusi-Port Memory) IN (Dusi-Port Memory
Handshake Bit Biok 1 Channel Type Size of Channel Size dannel Channel Start Address Biok 2 Channel Type Size of Channel Channel Start Address Biok 2 Size of Channel Channel Start Address Channel Start Address Channel Start Address Size of Channel Channel Start Address Channel Start Address Size of Channel Channel Start Address Channel Start Address Size of Channel Size of Chanel Size Of Size Of Size Size Of Size Of Size Size Of Size Of Size Of Size Size Of Size Of Size Of Size Size Of Size Of Size Of Size Of Size Size Of Size O	DPM (Dual-Port Memory) UNIXOWN 5 Communication 15616 bytes 55C000 Communication 15616 bytes 55C0C0 III HANDSHAKE CHAINEL 16 8175 Y-\$C082,0,16 X-\$C082,0,16 X-\$C082,0,16 X-\$C082,0,16 X-\$C082,0,16 MASTER Managin Node 0 9 COMTROL E bytes 54C0C2 OUT (Host System to netX) DPM (Dual-Port Memory) UNICONTROLLED 0 COMMON STATUS 64 bytes 54C0C4 III (Host System) DPM (Dual-Port Memory) UNICONTROLLED 0 E EXTENDED STATUS 4432 bytes 54C0C4 III (Inclassion Common Common DPM (Dual-Port Memory) UNICONTROLLED 0 EXTENDED STATUS 4432 bytes 54C0C4 III (Inclassion Common DPM (Dual-Port Memory) UNICONTROLLED 0 EXTENDED STATUS 4432 bytes 55C0C4 III (Inclassion Common DPM (Dual-Port Memory) UNICONTROLLED 0 EXTENDED STATUS	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 56C060 IN HANDSHAKE CHANNEL 16 BITS 55C062 0 55C062 0 CONTROL 8 bytes 55C062 0 CONTROL 8 bytes 55C064 0 CONTROL 8 bytes 55C064 0	UNKKOWN S Handshake 256 bytes 56C080 Communication 15616 bytes 56C080 IN Instance Characteristics 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONMON STATUS 56 bytes 56C04 IN Intek to host System DPM (Dual-Port Memory) DM (DM (Dual-Port Memory) DM (DM (DM (DM (DM (DM (DM (DM (DM (DM (UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 Server 0 9 COMTROL 8 bytes 56C02 0 COMTROL 9 CONTROL 9 C	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 156165 bytes 56C080 Communication 156165 bytes 56C00 IN HANDSHAKE CHANNEL 16 BITS 56C00 N HANDSHAKE CHANNEL 16 BITS Scanner 0 9 CONTROL 8 bytes 56C00 00 COMMON STATUS 64 bytes 56C004 IN Inext to host System DPM (Dual-Port Memory) DM (Dual-Port	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 Status 56C08 Communication 7424 Communication 742 Communication 74 Communication 7	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 72424 bytes 56C080 Communication 7424 bytes 56C080 S6C08 CONTROL 8 bytes 56C02 CONTROL 9 CONTROL	5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C020 NEANSTARE CHANNEL 15516 bytes 56C020 NEANSTARE CHANNEL 15516 7550020,016 X550020,016 X550020,016 X550020,016 0 CONTROL 8 bytes 56C02 0 CONTROL 9 CONTROL 0 CONTROL 0 CONTROLED 0 CONTROL	5 Handshake 256 bytes 356 coso Communication 15616 bytes 366 coco IN HANDSHAKE CHANNEL 16 BITS Y*SEG020,16 SLAVE I6 Controller 0 9 CONTROL 8 bytes 50 CONTROL 8 bytes 50 CONTROL 8 bytes 50 CONTROL 9 CONTROLLED 0 0 COMMON STATUS 64 bytes 55CC04 IN IneX to host System) DPM (Dual-For Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes 55C04 IN IneX to host System) DPM (Dual-For Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes 55C04 IN IneX to host System) DPM (Dual-For Memory) UNCONTROLED 0	5 Handshake 256 bytes 56C080 Communication 15616 bytes 56C080 W BANSHAKE CHANNEL 15516 bytes 56C020 W BANSHAKE CHANNEL 15516 55C020,016 55C020 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 9 CONTR	5 Handshake 25 bytes 56 CD80 Communication 15616 bytes 56 CD80 Communication 15616 bytes 56 CO20 N H D00 Second Se	S Handshake 25 bytes 56C08 Communication 15616 bytes 56C00 IN HARSHARE CHANNEL 1567 S6C02,016 X55C02,016 X55C02,016 K55C02,016 CONTROL 8 bytes 56C02 OUT (Host System to netX) DPM (Joual-Port Memory) UKCONTROLLED 0 EXTENDED STATUS 56 bytes 56C04 S0 EXTENDED STATUS 56 bytes 56C04 IN (netX to Host System) DPM (Joual-Port Memory) UKCONTROLLED 0 EXTENDED STATUS 56 bytes 56C04 N0 (netX to Host System) DPM (Joual-Port Memory) UKCONTROLLED 0 EXTENDED STATUS 56 bytes 56C04 N0 (netX to Host System) DPM (Joual-Port Memory) UKCONTROLLED 0 EXTENDED STATUS 56 bytes 56C04 N0 (netX to Host System) DPM (Joual-Port Memory) UKCONTROLLED 0 EXTENDED STATUS 56 bytes 56C04 N0 (netX to Host System) DPM (Joual-Port Memory) UKCONTROLLED 0	S Handshake 256 bytes ScC080 Communication 15616 bytes ScC020 IN HANDSHAKE CHANNEL 16 BITS Y-ScC082,0,16 X-ScC082,0,16 IN-CONTROLER Programmable Logic Controller (PIc) 0 9 CONTROL 8 bytes ScC022 OUT (Hoad System to netX) DFM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes ScC024 IN (netX to host System) 0 EXTENDED STATUS 432 bytes ScC04 IN (netX to host System) DFM (Dual-Port Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes ScC04 IN (netX to host System) DFM (Dual-Port Memory) UNCONTROLED 0	255 bytes 56080 Communication 15516 bytes 56000 IN HANDSHARE (HANNEL III HIS 15617 Y-50082,016 X-5002,016
Handshake Bit Block 1 Channel Yope Size of Channel Channel Start Address Block 2 Channel Start Address Channel Start Address Size of Handshake Cells Size of Handshake Register Host Handshake Register Host Handshake Register Host Handshake Register Size of Subblocks Size Subblock 1 Size Subblock 1 Size Subblock 1 Size Start Offset Transfer Direction Transfer Direction Transfer Direction Transfer Direction Transfer Poirection Transfer Direction Transfer	DPM (Dual-Port Memory) UNIXNOWN 5 3 4 4andshake 255 bytes 56C080 7 55156 bytes 55000 7 7 55156 bytes 55000 7 7 7 7 7 7 8 7 8 7 8 7 7 8 7 8 7 8	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 742k bytes 56C080 742k bytes 56C080 742k bytes 56C082.0.16 X-55C082.0.16 X-55C082.0.16 X-55C082.0.16 SIAVE MARCHARCHARCHARCHARCHARCHARCHARCHARCHARCH	UNKNOWN 5 Handshate 256 bytes 55C080 Communication 156 life bytes 56C080 IN HANOSHAKE CHANNEL 16 BITS V156C082.0.16 X55C08	UNKNOWN 5 Handshake 256 bytes 5C0800 Communication 7424 bytes 5C0800 Communication 7424 bytes 5C0800 N HANDSHAKE CHANNEL 16 BITS V*SC0820.016 X-SC0820.016 X-SC0820.016 SLAVE 0 9 CONTROL 8 bytes 5C0CC 0 9 CONTROL 8 bytes 5C0CC 0 0 COMMON STATUS 64 bytes 5CCC4 IN (netX to host System) DPM (Dual-Port Memory) UNCONTROLED 0 0 CONTROL System 10 pm (Dual-Port Memory) UNCONTROLED 0 CONTROL System 10 pm (Dual-Port Memory) UNCONTROLED 0 CONTROLED CONTROLED 0 CONTROLED CONTR	UNKNOWN S Handshake 256 bytes 56000 56000 56000 56000 15616 bytes 56000 1611 16	UNKNOWN S Handbake 256 bytes 56C080 Communication 742 bytes 56C080 Communication 742 bytes 56C080 Communication 742 bytes 56C082.0.16 55LVE 56C08 COMMON STATUS 66 bytes 56C08 COMMON STATUS 56C08 COMMON S	UNKNOWN S UNKNOWN S Handshake 256 bytes 56C000 Communication 7424 bytes 56C020 NH HANDSHAKE CHANNEL 16 BITS V*56C082.0.16 X*56C082.0.16 SLAVE 0 9 CONTROL B 16 Bit/se 56C021 SLAVE 0 CONTROL B 16 Bit/se 56C022 CONTROL B 16 Bit/se 56C024 CONTROLED 0 COMMON STATUS 56 bytes 56C02 CONTROLED 0 COMMON STATUS 56 bytes 56C02 CONTROLED 0 C	5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C020 IN TANDOSHARE CHANNEL 158175 V56C020 Jol V56C020 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROLED 0 CONTROLED	5 Handshake 25 bytes 56C00 Communication 15615 bytes 56C00 IN FANDSHAKE CHANNEL 15817 IN FANDSHAKE CHANNEL 158175 X56C02 IN FANDSHAKE CHANNEL 158175 X56C02 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 COMMON STATUS 64 bytes 56C04 IN IneX K4 hots System 0 COMMON STATUS 64 bytes 55C04 IN INEX K4 hots System 0 COMMON STATUS 64 bytes 55C04 IN INEX K4 hots System 0 DMM [Dual-Port Memory] UNCONTROLED 0 MILEOX 1600 bytes 1600 bytes	5 Handshake 256 bytes 36C080 Communication 15616 bytes 36C020 IN TANDSHAKE CHANNEL 15 BITS VSGC020 JSG VSGC02 JSGC02 JSG VSGC02 JSG VSGC02 JSG VSGC02 JSGC02 JSGC V	5 Handshake 256 bytes 56 c080 Communication 156 16 bytes 56 c020 IN HANOSHAKE CHANNEL 15 B175 V*55 c020 0.16 V*55 c020 0.16 Avb7ER 0 CONTROL 8 bytes 56 c02 CONTROL 8 bytes 56 c02 CONTROL 8 bytes 56 c02 COMMON STATUS 64 bytes 56 c02 0 COMMON STATUS 64 bytes 6 COMMON STATUS 6 6 C0 6 COMMON STATUS 6 6 6 C0 6 COMMON STATUS 6 6 7 6 C 6 C 6 C 6 C 6 C 6 C 6 C 7 6 C 7 6 C 7 6 C 7 6 C 7 7 7 7	S Handshake 25 bytes 25 cool Communication 156 ib bytes 56 coco N HANDSHAKE CHANNEL 16 075 X55 CO2 N HANDSHAKE CHANNEL 16 075 X55 CO2 CONTROL 8 bytes 56 coc 0 CONTROL 8 bytes 56 coc 0 COMMON STATUS 64 bytes 56 coc 0 COMMON STATUS 65 bytes 56 coc 0 COMMON STATUS 56 coc 160 com 0 COMMON STATUS 56 coc 160 com 0 COM 160 com 0 COM 160 com 0 COM 160 com 16	S Handshake 255 bytes 55C020 Communication 15616 bytes 55C020 IN HANDSHAKE CHANNEL 16 BITS V-55C020 0.15 V-55C020 0.15 V-55C020 0.15 V-55C020 0.15 V-55C02 0 CONTROL 8 bytes 55C02 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes 55C02 UN CONTROLED 0 EXTENDED STATUS 432 bytes 55C024 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLED 0 MALBOX 1600 bytes	256 bytes S6C080 Communication 15616 bytes S6C020 IN HANDSHAKE CHANNEL 16 BITS Y*S6C082,0,16 X*S6C082,0,16 K*S6C082,0,16 HO-DEVICE Programmable Logic Controller (P 67 9 CONTROL 8 bytes S6C02 OUT (Hoat System to netk) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes S6C024 IN (netX to Hoat System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 42 bytes S6C04 IN (netX to Hoat System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 42 bytes S6C04 IN (netX to Hoat System) DPM (Dual-Port Memory) UNCONTROLLED 0 MALBOX
Handshake Bit Block 1 Channel Star Address Channel Star Address Block 2 Channel Star Address Block 2 Channel Star Address Control Type Size of Channel Address Control Type Size of Channel Address Conformate Calss Size of Handshake Register Communication Class Protocol Class Conformance Class Number of Subblocks	DPM (Dual-Port Memory) UNIXNOWN 5 256 bytes 56C000 Communication 15616 bytes 56C000 N FORCO 156 bytes 56C000 N FANDSHARE CHANNEL 156 bit 7 SEC082,0,15 X-56C082,0,15 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C082,0,16 X-56C08 X-1000 Status 0 O COMMON STATUS 64 bytes 56C004 N N (net K biots System) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes 56C04 N N (net K biots System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 432 bytes 56C04 NHEOX hiots System) DPM (Dual-Port Memory) UNCONTROLLED 0 MALBOX 1500 bytes 56C100	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 2424 bytes 55CC00 IN HANDSHAKE CHANNEL 16 BITS 55CC00 IN HANDSHAKE CHANNEL 16 BITS 55CC00 IN HANDSHAKE CHANNEL 16 BITS 55CC00 IN HANDSHAKE CHANNEL 16 BITS 55CC00 9 CONTROL 8 bytes 55CC00 9 CONTROL 8 bytes 55CC00 9 CONTROL 8 bytes 55CC04 IN (next Kots System) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes 55CC04 IN (next Kots System) DPM (Dual-Port Memory) UNCONTROLED 0 EXTENDED STATUS 432 bytes 55CC04 IN (next Kots System) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes 55CC04 IN (next Kots System) DPM (Dual-Port Memory) UNCONTROLED 0 MALEOX 1000 bytes 55CC04	UNKNOWN 5 Handbake 255 bytes 56C080 Communication 15616 bytes 56C080 Communication 15615 bytes 56C00 IN INANDSHARE CHANNEL 156175 KSG020,016 KSG020,017 KSG020,	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C08 0 Common Status 8 bytes 56C08 Common Status 8 bytes 56C08 Common Status 6 bytes 56C08 C	UNIKNOWN S Handbake 256 bytes 56C080 Communication 15616 bytes 56C080 Communication 15616 bytes 56C020 NI NAKOSHAKE CHANNEL 16 BITS 6 Motion 20,16 Motion 20,17 Motion 20	UNKNOWN 5 Handbake 256 bytes 56C080 Communication 7424 bytes 56C08 0 Common 5	UNKNOWN 5 Handshake 256 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C080 Communication 7424 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROL 8 bytes 56C02 0 CONTROLED 0 COMMON STATUS 64 bytes 56C02 0 COMMON STATUS 64 bytes 56C04 MI (IneX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 56C10 0 COM	5 Handshake 255 bytes 56C080 Communication 55C080 Communication 55C080 N HANDSHAKE CHANNEL 55C082,0,16 MASTER 16-Controller 0 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 0 COMMON STATUS 64 bytes 56C02 0 COMMON STATUS 64 bytes 55C02 0 COMMON STATUS 64 bytes 55C04 N Intext Subst System DPM [Dual-Port Memory] DPM [Dual-Port Memory] DPM [Dual-Port Memory] 0 COMMON STATUS 64 bytes 55C04 N Intext Subst System DPM [Dual-Port Memory] 0 COMMON STATUS 64 bytes 55C04 N Intext Subst System DPM [Dual-Port Memory] 0 COMMON STATUS 64 bytes 55C04 N Intext Subst System DPM [Dual-Port Memory] 0 COMMON STATUS 64 bytes 55C10 DPM [Dual-Port Memory] 0 COMON STATUS 64 bytes 55C10 COM DPM [Dual-Port Memory] 0 COMON STATUS 64 bytes 55C10 COM DPM [Dual-Port Memory] 0 COMON STATUS 64 bytes 55C10 COM DPM [Dual-Port Memory] 0 COMON STATUS 64 bytes 55C10 COM DPM [Dual-Port Memory] 0 COMON STATUS 64 bytes 55C10 COM DPM [Dual-Port Memory] 0 COMON STATUS 64 bytes 55C10 COM DPM [Dual-Port Memory] 0 COMON STATUS 64 bytes 55C10 COM DPM [Dual-Port Memory] 0 COMON STATUS 64 bytes 55C10 COM DPM [Dual-Port Memory] 0 COMON STATUS 64 bytes 55C10 COM DPM [Dual-Port Memory] 0 COM DPM [DUAL-PORT	5 Handshake 256 bytes 56C080 Gommunication S6C080 Gommunication S6C080 Koster S6C082,0.16 S4S082,0.16 S4S082,0.16 S4S082,0.16 S4S082,0.16 S5C022 CONTROL Bytes S6C02 CONTROL Bytes S6C02 CONTROLEB CONTROLEB CONTROLED C	5 Handshake 256 bytes 56C080 Gommunication 56C080 Gommunication 56C080 N HANDSHAKE CHANNEL 55C082.0.16 S5CNAE b-Device 0 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROLLED 0 COMMON STATUS 64 bytes 55C02 0 COM 55C02 0 COM 55C02 0 COM 55C02 0 COM 55C02 55C0 55C0 55C0 55C0 55C0 55C0 55C	5 Handshake 256 bytes 56C080 Communication 55C080 Communication 55C080 S5C080 N HANDSHAKE CHANNEL 555C082.0.16 ADAPTER b-Device 0 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROLLED 0 COMMON STATUS 64 bytes 55C024 N (next Substram) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 432 bytes 55C04 N (next Substram) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 432 bytes 55C04 N (next Not Stram) DPM (Dual-Port Memory) UNCONTROLLED 0 MMLEDX 1600 bytes 55C14	S Handshake 256 bytes 56C00 Communication 45C00 Communication 45C00 Kicks 45C0	S Handshake 255 bytes S5C026 Communication S56020 Communication S56020 Contract HANDSHAKE CANNEL IF BITS Y-SEC082.0.16 CONTROLER Programmabile Logic Controller (Pic) 0 SCONTROL Bbytes SC022 CONTROL Bbytes SC02 COMMON STATUS G4 bytes S5C024 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS S5C024 S5C024 S5C024 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS S5C024 S5C024 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS S5C024 S5C024 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS S5C024 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS S5C10 MALBOX 1500 bytes S5C14 IS00 IS00 IS00 IS00 IS00 IS00 IS00 IS0	2256 bytes SECORD Communication 15616 bytes SECORD IN HANDSHAKE CHANNEL 16 BITS Y-SECOR2,0,16 X-SECOR2,0,16 IO-DEVICE Programmable Logic Controller (P 67 9 CONTROL 8 bytes SECOR2 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS SECOR2 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS SECOR2 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 0 MALBOX 1600 bytes SECOR4
Handshake Bit Biok 1 Channel Type Size of Channel Channel Start Address Biok 2 Channel Type Size of Channel Channel Start Address Biok 2 Eiske of Channel Eiske of	DPM (Dual-Port Memory) UNIXNOWN 5 3 4 4andshake 255 bytes 56C080 7 55156 bytes 55000 7 7 55156 bytes 55000 7 7 7 7 7 7 8 7 8 7 8 7 7 8 7 8 7 8	DPM (Dual-Port Memory) UNIXNOWN 5 Handshake 256 bytes 56C080 Communication 742k bytes 56C080 742k bytes 56C080 742k bytes 56C082.0.16 X-55C082.0.16 X-55C082.0.16 X-55C082.0.16 SIAVE MARCHARCHARCHARCHARCHARCHARCHARCHARCHARCH	UNKNOWN 5 Handshate 256 bytes 55C080 Communication 156 life bytes 56C080 IN HANOSHAKE CHANNEL 16 BITS V156C082.0.16 X55C08	UNKNOWN 5 Handshake 256 bytes 5C0800 Communication 7424 bytes 5C0800 Communication 7424 bytes 5C0800 N HANDSHAKE CHANNEL 16 BITS V*SC0820.016 X-SC0820.016 X-SC0820.016 SLAVE 0 9 CONTROL 8 bytes 5C0CC 0 9 CONTROL 8 bytes 5C0CC 0 0 COMMON STATUS 64 bytes 5CCC4 IN (netX to host System) DPM (Dual-Port Memory) UNCONTROLED 0 0 CONTROL System 10 pm (Dual-Port Memory) UNCONTROLED 0 CONTROL System 10 pm (Dual-Port Memory) UNCONTROLED 0 CONTROLED CONTROLED 0 CONTROLED CONTR	UNKNOWN S Handshake 256 bytes 56000 56000 56000 56000 15616 bytes 56000 1611 16	UNKNOWN S Handbake 256 bytes 56C080 Communication 742 bytes 56C080 Communication 742 bytes 56C080 Communication 742 bytes 56C082.0.16 55LVE 56C08 COMMON STATUS 66 bytes 56C08 COMMON STATUS 56C08 COMMON S	UNKNOWN S UNKNOWN S Handshake 256 bytes 56C000 Communication 7424 bytes 56C020 NH HANDSHAKE CHANNEL 16 BITS V*56C082.0.16 X*56C082.0.16 SLAVE 0 9 CONTROL B 16 Bit/se 56C021 SLAVE 0 CONTROL B 16 Bit/se 56C022 CONTROL B 16 Bit/se 56C024 CONTROLED 0 COMMON STATUS 56 bytes 56C02 CONTROLED 0 COMMON STATUS 56 bytes 56C02 UNCONTROLED 0 COMMON STATUS 56 bytes 56C02 SCO2 SCO2 SCO2 SCO2 SCO2 SCO2 SCO2 SCO	5 Handshake 255 bytes 56C080 Communication 15616 bytes 56C020 IN TANDOSHARE CHANNEL 158175 V56C020 Jol V56C020 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 CONTROLED 0 CONTROLED	5 Handshake 25 bytes 56C00 Communication 15615 bytes 56C00 IN FANDSHAKE CHANNEL 15817 IN FANDSHAKE CHANNEL 158175 X56C02 IN FANDSHAKE CHANNEL 158175 X56C02 CONTROL 8 bytes 56C02 CONTROL 8 bytes 56C02 COMMON STATUS 64 bytes 56C04 IN IneX K4 hots System 0 COMMON STATUS 64 bytes 55C04 IN INEX K4 hots System 0 COMMON STATUS 64 bytes 55C04 IN INEX K4 hots System 0 DMM [Dual-Port Memory] UNCONTROLED 0 MILEOX 1600 bytes 1600 bytes	5 Handshake 256 bytes 36C080 Communication 15616 bytes 36C020 IN TANDSHAKE CHANNEL 15 BITS VSGC020 JSG VSGC02 JSGC02 JSG VSGC02 JSG VSGC02 JSG VSGC02 JSGC02 JSGC V	5 Handshake 256 bytes 56 c080 Communication 156 16 bytes 56 c020 IN HANOSHAKE CHANNEL 15 B175 V*55 c020 0.16 V*55 c020 0.16 Avb7ER 0 CONTROL 8 bytes 56 c02 CONTROL 8 bytes 56 c02 CONTROL 8 bytes 56 c02 COMMON STATUS 64 bytes 56 c02 0 COMMON STATUS 64 bytes 55 c004 IN Inext to host System DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 55 c004 IN Inext to host System DPM (Dual-Port Memory) UNCONTROLLED 0 MALBOX 1600 bytes 1600 bytes	S Handshake 25 bytes 25 cool Communication 156 ib bytes 56 coco N HANDSHAKE CHANNEL 16 075 X55 CO2 N HANDSHAKE CHANNEL 16 075 X55 CO2 CONTROL 8 bytes 56 coc 0 CONTROL 8 bytes 56 coc 0 COMMON STATUS 64 bytes 56 coc 0 COMMON STATUS 65 bytes 56 coc 0 COMMON STATUS 56 coc 160 com 0 COMMON STATUS 56 coc 160 com 0 COM 160 com 0 COM 160 com 0 COM 160 com 16	S Handshake 255 bytes 55C020 Communication 15616 bytes 55C020 IN HANDSHAKE CHANNEL 16 BITS V-55C020 0.15 V-55C020 0.15 V-55C020 0.15 V-55C020 0.15 V-55C02 0 CONTROL 8 bytes 55C02 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLED 0 COMMON STATUS 64 bytes 55C02 UN CONTROLED 0 EXTENDED STATUS 432 bytes 55C024 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLED 0 MALBOX 1600 bytes	256 bytes S6C080 Communication 15616 bytes S6C020 IN HANDSHAKE CHANNEL 16 BITS Y*S6C082,0,16 X*S6C082,0,16 K*S6C082,0,16 IN OEVICE CONTROL 8 bytes S6C02 OUT (Host System to netX) DPM (Dual-Port Memory) UNCONTROLLED 0 COMMON STATUS 64 bytes S6C024 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 42 bytes S6C004 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 42 bytes S6C004 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 EXTENDED STATUS 42 bytes S6C004 IN (netX to Host System) DPM (Dual-Port Memory) UNCONTROLLED 0 MALBOX

	PROFIBUS-DP Master	PROFIBUS-DP Slave	DeviceNet Master	DeviceNet Slave	CANopen Master	CANopen Slave	CC-Link Slave	EtherCAT Master	EtherCAT Slave	EtherNet/IP Scanner/Master	EtherNet/IP Adapter/Slave	Open Modbus/TCP	PROFINET IO Controller/Master	PROFINET IO Device/Slave
Subblock 4 Size	MAILBOX 1600 bytes													
Start Offset	\$6C2D0	\$6C2D0 IN (netX to Host System)	\$6C2D0	\$6C2D0	\$6C2D0	\$6C2D0	\$6C2D0	\$6C2D0						
Transfer Direction Transfer Type	IN (netX to Host System) DPM (Dual-Port Memory)													
Handshake Mode Handshake Bit	UNKNOWN 5													
Subblock 5	PROCESS DATA IMAGE													
Size	5760 bytes	1536 bytes	5760 bytes	1536 bytes	5760 bytes	1536 bytes	1536 bytes	5760 bytes						
Start Offset Transfer Direction	\$6C4C0 OUT (Host System to netX)													
Transfer Type Handshake Mode	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory)	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory)	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROL							
Handshake Mode Handshake Bit	6	BUFFERED, HOST CONTROLLED 6	6	BUFFERED, HOST CONTROLLED 6	6	6	6							
- Subblock 6	PROCESS DATA IMAGE													
Size	5760 bytes	1536 bytes	5760 bytes	1536 bytes \$6C640	5760 bytes	1536 bytes	1536 bytes	5760 bytes	5760 bytes \$6CA60	5760 bytes \$6CA60	5760 bytes \$6CA60	5760 bytes	5760 bytes	5760 bytes \$6CA60
Start Offset Transfer Direction	\$6CA60 IN (netX to Host System)	\$6C640 IN (netX to Host System)	\$6CA60 IN (netX to Host System)	IN (netX to Host System)	\$6CA60 IN (netX to Host System)	\$6C640 IN (netX to Host System)	\$6C640 IN (netX to Host System)	\$6CA60 IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	\$6CA60 IN (netX to Host System)	\$6CA60 IN (netX to Host System)	IN (netX to Host System)
Transfer Type Handshake Mode	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROL												
Handshake Bit	7	7	7	7	7	7	7	7	7	7	7	7	7	7
- Subblock 7	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAG												
Size Start Offset	64 bytes \$6C460													
Transfer Direction	OUT (Host System to netX)													
Transfer Type Handshake Mode	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROL												
Handshake Bit	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Subblock 8	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAG												
Size Start Offset	64 bytes \$6C470													
Transfer Direction	IN (netX to Host System)													
Transfer Type Handshake Mode	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROL												
Handshake Bit	9	9	9	9	9	9	9	9	9	9	9	9	9	9
llock 3														
Channel Type Size of Channel	Undefined 0 bytes	Communication 15616 bytes	Undefined 0 bytes	Undefined 0 bytes	Undefined 0 bytes	Communication 15616 bytes								
Channel Start Address	\$6D000	\$6C800	\$6D000	\$6C800	\$6D000	\$6C800	\$6C800	\$6D000						
Position of Handshake Cells Size of Handshake Cells	BEGINNING OF CHANNEL NOT AVAILABLE	IN HANDSHAKE CHANNEL 16 BITS	BEGINNING OF CHANNEL NOT AVAILABLE	BEGINNING OF CHANNEL NOT AVAILABLE	BEGINNING OF CHANNEL NOT AVAILABLE	IN HANDSHAKE CHANNEL 16 BITS								
NetX Handshake Register Host Handshake Register	X:\$6D000 X:\$6D000,8,0	X:\$6C800 X:\$6C800,8,0	X:\$6D000 X:\$6D000,8,0	X:\$6C800 X:\$6C800.8.0	X:\$6D000 X:\$6D000,8,0	X:\$6C800 X:\$6C800,8,0	X:\$6C800 X:\$6C800,8,0	X:\$6D000 X:\$6D000,8,0	X:\$6D000 X:\$6D000,8,0	Y:\$6C083,0,16 X:\$6C083,0,16	X:\$6D000 X:\$6D000,8,0	X:\$6D000 X:\$6D000,8,0	X:\$6D000 X:\$6D000,8,0	Y:\$6C083,0,16 X:\$6C083,0,16
Communication Class	UNDEFINED	MESSAGING	UNDEFINED	UNDEFINED	UNDEFINED	MESSAGING								
Protocol Class Conformance Class	UNDEFINED 0													
Number of Subblocks	0	0	0	0	0	0	0	0	0	9	0	0	0	9
Subblock 0										CONTROL				CONTROL
Size Start Offset										8 bytes \$6D002				8 bytes \$6D002
Transfer Direction										OUT (Host System to netX)				OUT (Host System to netX)
Transfer Type Handshake Mode										DPM (Dual-Port Memory) UNCONTROLLED				DPM (Dual-Port Memory) UNCONTROLLED
Handshake Bit										0				0
Subblock 1										COMMON STATUS				COMMON STATUS
Size Start Offset										64 bytes \$6D004				64 bytes \$6D004
Transfer Direction										IN (netX to Host System)				IN (netX to Host System)
Transfer Type Handshake Mode										DPM (Dual-Port Memory) UNCONTROLLED				DPM (Dual-Port Memory) UNCONTROLLED
Handshake Bit										0				0
Subblock 2										EXTENDED STATUS				EXTENDED STATUS
Size Start Offset										432 bytes \$6D014				432 bytes \$6D014
Transfer Direction										IN (netX to Host System)				IN (netX to Host System)
Transfer Type Handshake Mode										DPM (Dual-Port Memory) UNCONTROLLED				DPM (Dual-Port Memory) UNCONTROLLED
Handshake Bit										0				0
Subblock 3										MAILBOX				MAILBOX
Size Start Offset										1600 bytes \$6D080				1600 bytes \$6D080
Transfer Direction										OUT (Host System to netX)				OUT (Host System to netX)
Transfer Type Handshake Mode										DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED				DPM (Dual-Port Memory) BUFFERED, HOST CONTROLI
Handshake Bit										4				4
- Subblock 4										MAILBOX				MAILBOX
Size Start Offset										1600 bytes \$6D210				1600 bytes \$6D210
Transfer Direction Transfer Type										IN (netX to Host System) DPM (Dual-Port Memory)				IN (netX to Host System) DPM (Dual-Port Memory)
Handshake Mode										UNKNOWN				UNKNOWN
Handshake Bit										5				5
- Subblock 5										PROCESS DATA IMAGE				PROCESS DATA IMAGE
Size Start Offset										5760 bytes \$6D400				5760 bytes \$6D400
Transfer Direction Transfer Type										OUT (Host System to netX) DPM (Dual-Port Memory)				OUT (Host System to netX) DPM (Dual-Port Memory)
Handshake Mode										BUFFERED, HOST CONTROLLED				BUFFERED, HOST CONTROLL
Handshake Bit										6				6
Subblock 6										PROCESS DATA IMAGE 5760 bytes				PROCESS DATA IMAGE 5760 bytes
Size Start Offset										\$6D9A0				\$6D9A0
Transfer Direction Transfer Type										IN (netX to Host System) DPM (Dual-Port Memory)				IN (netX to Host System) DPM (Dual-Port Memory)
Handshake Mode										BUFFERED, HOST CONTROLLED				BUFFERED, HOST CONTROLL
Handshake Bit										7				7
Subblock 7										HIGH PRIORITY DATA IMAGE				HIGH PRIORITY DATA IMAGE
Size Start Offset										64 bytes \$6D3A0				64 bytes \$6D3A0
Transfer Direction										OUT (Host System to netX)				OUT (Host System to netX)
Transfer Type Handshake Mode										DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED				DPM (Dual-Port Memory) BUFFERED, HOST CONTROLL
Handshake Bit										8				8
- Subblock 8										HIGH PRIORITY DATA IMAGE				HIGH PRIORITY DATA IMAG
Size Start Offset										64 bytes \$6D3B0				64 bytes \$6D3B0
Transfer Direction										IN (netX to Host System)				IN (netX to Host System)
Transfer Type Handshake Mode										DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED				DPM (Dual-Port Memory) BUFFERED, HOST CONTROL
Handshake Bit										9				9
ock 4														
hannel Type	Undefined													

	PROFIBUS-DP Master	PROFIBUS-DP Slave	DeviceNet Master	DeviceNet Slave	CANopen Master	CANopen Slave	CC-Link Slave	EtherCAT Master	EtherCAT Slave	EtherNet/IP	EtherNet/IP	Open Modbus/TCP	PROFINET IO	PROFINET IO
										Scanner/Master	Adapter/Slave		Controller/Master	Device/Slave
Channel Start Address	\$6D000	\$6C800	\$6D000	\$6C800	\$6D000	\$6C800	\$6C800	\$6D000	\$6D000	\$6DF40	\$6D000	\$6D000	\$6D000	\$6DF40
Position of Handshake Cells	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL
Size of Handshake Cells	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE
NetX Handshake Register	X:\$6D000	X:\$6C800	X:\$6D000	X:\$6C800	X:\$6D000	X:\$6C800	X:\$6C800	X:\$6D000	X:\$6D000	X:\$6DF40	X:\$6D000	X:\$6D000	X:\$6D000	X:\$6DF40
Host Handshake Register	X:\$6D000,8,0	X:\$6C800,8,0	X:\$6D000,8,0	X:\$6C800,8,0	X:\$6D000,8,0	X:\$6C800,8,0	X:\$6C800,8,0	X:\$6D000,8,0	X:\$6D000,8,0	X:\$6DF40,8,0	X:\$6D000,8,0	X:\$6D000,8,0	X:\$6D000,8,0	X:\$6DF40,8,0
Communication Class	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED
Protocol Class	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED
Conformance Class	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Number of Subblocks	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Block 5														
	Undefined		Undefined		Undefined		11-1-61	Undefined	Undefined	Undefined		11-1-71	Undefined	
Channel Type		Undefined		Undefined		Undefined	Undefined				Undefined	Undefined		Undefined
Size of Channel	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes	0 bytes
Channel Start Address	\$6D000	\$6C800	\$6D000	\$6C800	\$6D000	\$6C800 BEGINNING OF CHANNEL	\$6C800	\$6D000 BEGINNING OF CHANNEL	\$6D000	\$6DF40	\$6D000	\$6D000	\$6D000 BEGINNING OF CHANNEL	\$6DF40
Position of Handshake Cells	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL NOT AVAILABLE	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL		BEGINNING OF CHANNEL NOT AVAILABLE		BEGINNING OF CHANNEL	BEGINNING OF CHANNEL NOT AVAILABLE	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL NOT AVAILABLE		BEGINNING OF CHANNEL
Size of Handshake Cells	NOT AVAILABLE	NOT AVAILABLE		NOT AVAILABLE	NOT AVAILABLE	NOT AVAILABLE		NOT AVAILABLE	NOT AVAILABLE		NOT AVAILABLE		NOT AVAILABLE	NOT AVAILABLE
NetX Handshake Register	X:\$6D000	X:\$6C800	X:\$6D000	X:\$6C800	X:\$6D000	X:\$6C800	X:\$6C800	X:\$6D000	X:\$6D000	X:\$6DF40	X:\$6D000	X:\$6D000	X:\$6D000	X:\$6DF40
Host Handshake Register	X:\$6D000,8,0	X:\$6C800,8,0	X:\$6D000,8,0	X:\$6C800,8,0	X:\$6D000,8,0	X:\$6C800,8,0	X:\$6C800,8,0	X:\$6D000,8,0	X:\$6D000,8,0	X:\$6DF40,8,0	X:\$6D000,8,0	X:\$6D000,8,0	X:\$6D000,8,0	X:\$6DF40,8,0
Communication Class	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED
Protocol Class	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED
Conformance Class	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Number of Subblocks	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APPENDIX C – POWER PMAC MEMORY MAPS

	PROFIBUS-DP Master	PROFIBUS-DP Slave	DeviceNet Master	DeviceNet Slave	CANopen Master	CANopen Slave	CC-Link Slave	EtherCAT Master	EtherCAT Slave	EtherNet/IP Scanner/Master	EtherNet/IP Adapter/Slave	Open Modbus/TCP	PROFINET IO Controller/Master	PROFINET IO Device/Slave
ACC-72EX Address letX Identification	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX	Acc72EX[i].a netX
Dual-Port Memory Size Device Number	16384 bytes 1532410	8192 bytes 1562420	16384 bytes 1532510	8192 bytes 1562520	65536 bytes 1532500	8192 bytes 1562540	8192 bytes 1562740	16384 bytes 1532100	16384 bytes 1532100	65536 bytes 1532100	16384 bytes 1532100	16384 bytes 1532100	32768 bytes 1532100	32768 bytes 1532100
ardware Assembly Options														
Port 0 Port 1	NOT CONNECTED NOT CONNECTED	PROFIBUS NOT AVAILABLE	NOT CONNECTED NOT CONNECTED	DEVICENET NOT AVAILABLE	NOT CONNECTED NOT CONNECTED	CAN NOT AVAILABLE	CC-LINK NOT AVAILABLE	ETHERNET (internal Phy) ETHERNET (internal Phy)	ETHERNET (internal Phy) ETHERNET (internal Phy)	ETHERNET (internal Phy) ETHERNET (internal Phy)	ETHERNET (internal Phy) ETHERNET (internal Phy)	ETHERNET (internal Phy) ETHERNET (internal Phy)	ETHERNET (internal Phy) ETHERNET (internal Phy)	ETHERNET (internal Phy) ETHERNET (internal Phy)
Port 2	PROFIBUS	NOT AVAILABLE	DEVICENET	NOT AVAILABLE	CAN	NOT AVAILABLE	NOT AVAILABLE	NOT CONNECTED	NOT CONNECTED	NOT CONNECTED	NOT CONNECTED	NOT CONNECTED	NOT CONNECTED	NOT CONNECTED
Port 3 ilscher Module License Information	NOT CONNECTED (PROFIBUS Master) (CANopen	NOT AVAILABLE Unlimited number of master	NOT CONNECTED (PROFIBUS Master) (CANopen	NOT AVAILABLE Unlimited number of master	NOT CONNECTED (PROFIBUS Master) (CANopen	NOT AVAILABLE Unlimited number of master	NOT AVAILABLE Unlimited number of master	NOT CONNECTED (PROFIBUS Master) (CANopen	NOT CONNECTED Unlimited number of master	NOT CONNECTED (PROFIBUS Master) (CANopen	NOT CONNECTED Unlimited number of master	NOT CONNECTED Unlimited number of master	NOT CONNECTED (PROFIBUS Master) (CANopen	NOT CONNECTED Unlimited number of master
	Master) (DeviceNet Master) (AS-	licenses	Master) (DeviceNet Master) (AS-	licenses	Master) (DeviceNet Master) (AS-	licenses	licenses	Master) (DeviceNet Master) (AS-	licenses	Master) (DeviceNet Master) (AS-	licenses	licenses	Master) (DeviceNet Master) (AS-	licenses
	Interface Master) (PROFINET IO RT Controller) (EtherCAT Master)		Interface Master) (PROFINET IO RT Controller) (EtherCAT Master)		Interface Master) (PROFINET IO RT Controller) (EtherCAT Master)			Interface Master) (PROFINET IO RT Controller) (EtherCAT Master)		Interface Master) (PROFINET IO RT Controller) (EtherCAT Master)			Interface Master) (PROFINET IO RT Controller) (EtherCAT Master)	
	(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License		(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License		(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License			(EtherNet/IP Scanner) (SERCOS III Master) Unlimited number of master licenses		(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License			(EtherNet/IP Scanner) (SERCOS III Master) 1 Master License	
Tool License Information Device Class	(SYCON.net) COMX 100	COMX 10	(SYCON.net) COMX 100	COMX 10	(SYCON.net) COMX 100	COMX 10	COMX 10	(SYCON.net) COMX 100	COMX 100	(SYCON.net) COMX 100	COMX 100	COMX 100	(SYCON.net) COMX 100	COMX 100
Block 0														
Channel Type	System	System	System	System	System	System	System	System	System	System	System	System	System	System
Size of Channel Channel Start Address	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a	512 bytes Acc72EX[i].Data8[0].a
Position of Handshake Cells netX System Flags Address	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a	IN HANDSHAKE CHANNEL Acc72EX[i].Data8[514].a
Host System Flags Address	Acc72EX[i].Data8[514].a	Acc72EX[i].Data8[515].a	Acc72EX[i].Data8[515].a	Acc72EX[i].Data8[515].a	Acc72EX[i].Data8[514].a Acc72EX[i].Data8[515].a	Acc72EX[i].Data8[514].a	Acc72EX[i].Data8[514].a	Acc72EX[i].Data6[514].a	Acc72EX[i].Data8[514].a	Acc72EX[i].Data8[514].a	Acc72EX[i].Data8[514].a	Acc72EX[i].Data8[514].a	Acc72EX[i].Data8[514].a	Acc72EX[i].Data8[515].a
Size of Handshake Cells Size of Mailbox	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes	8 BITS 256 bytes
Mailbox Start address	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a	Acc72EX[i].Data8[256].a
Number of Subblocks	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Subblock 0	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS
Size Start Offset	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a	176 bytes Acc72EX[i].Data8[0].a
Transfer Direction Transfer Type	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)	IN - OUT (Bi-Directional) DPM (Dual-Port Memory)
Handshake Mode	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED
Handshake Bit	0	0	0	0	0	0	0	0	0	0	0	0	0	0
- Subblock 1	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL
Size Start Offset	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a	8 bytes Acc72EX[i].Data8[184].a
Transfer Direction	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)	OUT (Host System to netX)
Transfer Type Handshake Mode	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED	DPM (Dual-Port Memory) UNCONTROLLED
Handshake Bit	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Subblock 2	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS	COMMON STATUS
Size	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes
Start Offset Transfer Direction	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)	Acc72EX[i].Data8[192].a IN (netX to Host System)
Transfer Type	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)
Handshake Mode Handshake Bit	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0
Subblock 3	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX
Size	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes
Start Offset Transfer Direction	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)	Acc72EX[i].Data8[256].a OUT (Host System to netX)
Transfer Type	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)
Handshake Mode Handshake Bit	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4	BUFFERED, HOST CONTROLLED 4
Subblock 4 Size	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes	MAILBOX 128 bytes
Size Start Offset	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a	Acc72EX[i].Data8[384].a
Transfer Direction Transfer Type	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)
Handshake Mode	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN
Handshake Bit	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Block 1														
Channel Type Size of Channel	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes	Handshake 256 bytes
Channel Start Address	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a	Acc72EX[i].Data8[512].a
Block 2 Channel Type	Communication	Communication	Communication	Communication	Communication	Communication	Communication	Communication	Communication	Communication	Communication	Communication	Communication	Communication
Size of Channel	15616 bytes	7424 bytes	15616 bytes	7424 bytes	15616 bytes	7424 bytes	7424 bytes	15616 bytes	15616 bytes	15616 bytes	15616 bytes	15616 bytes	15616 bytes	15616 bytes
Channel Start Address Position of Handshake Cells	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL	Acc72EX[i].Data8[768].a IN HANDSHAKE CHANNEL
Size of Handshake Cells NetX Handshake Register	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a	16 BITS Acc72EX[i].Data8[520].a
Host Handshake Register	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a	Acc72EX[i].Data8[522].a
Communication Class Protocol Class	MASTER Managing Node	SLAVE Managing Node	MASTER Server	SLAVE Server	MASTER Scanner	SLAVE Scanner	SLAVE Adapter	MASTER Io-Controller	SLAVE Io-Controller	SCANNER Io-Device	ADAPTER Io-Device	MESSAGING Combination Firmware	IO-CONTROLLER Programmable Logic Controller (Plc)	IO-DEVICE Programmable Logic Controller
Conformance Class Number of Subblocks	0	0	0	0	0	0	0	0	0	0	0	0	0 9	67 9
- Subblock 0 Size	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes	CONTROL 8 bytes
Start Offset	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a	Acc72EX[i].Data8[768].a
Transfer Direction Transfer Type	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)
Handshake Mode Handshake Bit	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0
	-	· · · · · · · · · · · · · · · · · · ·								-	· · · · · · · · · · · · · · · · · · ·	-		
Subblock 1 Size	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes	COMMON STATUS 64 bytes
Start Offset	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a	Acc72EX[i].Data8[784].a
Transfer Direction Transfer Type	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)
Handshake Mode Handshake Bit	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED 0	UNCONTROLLED	UNCONTROLLED 0	UNCONTROLLED	UNCONTROLLED 0	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED 0	UNCONTROLLED 0
			EVERADED CEATING	EVTENDED STAT	EVERADED CEATING			EVTENDED CTATUS	EXTENDED CLATHS				EVTENDED CEATUR	
Subblock 2 Size	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes	EXTENDED STATUS 432 bytes
Start Offset	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a	Acc72EX[i].Data8[848].a
Transfer Direction Transfer Type	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)
Handshake Mode	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED	UNCONTROLLED
Handshake Bit	0	U	U	U	U	U	U	U	U	0	U	U	U	U
Subblock 3	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX	MAILBOX
Size Start Offset	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a	1600 bytes Acc72EX[i].Data8[1280].a
Transfer Direction	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)
Transfer Type Handshake Mode	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED
Handshake Bit	4	4	4	4	4	4	4	4	4	4	4	4	4	4

	PROFIBUS-DP Master	PROFIBUS-DP Slave	DeviceNet Master	DeviceNet Slave	CANopen Master	CANopen Slave	CC-Link Slave	EtherCAT Master	EtherCAT Slave	EtherNet/IP Scanner/Master	EtherNet/IP Adapter/Slave	Open Modbus/TCP	PROFINET IO Controller/Master	PROFINET IO Device/Slave
Subblock 4 Size	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes	MAILBOX 1600 bytes
Start Offset	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a	Acc72EX[i].Data8[2280].a
Transfer Direction Transfer Type	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)
Handshake Mode Handshake Bit	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5	UNKNOWN 5
Subblock 5	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE
Size Start Offset	5760 bytes Acc72EX[i].Data8[4480].a	1536 bytes Acc72EX[i].Data8[4480].a	5760 bytes Acc72EX[i].Data8[4480].a	1536 bytes Acc72EX[i].Data8[4480].a	5760 bytes Acc72EX[i].Data8[4480].a	1536 bytes Acc72EX[i].Data8[4480].a	1536 bytes Acc72EX[i].Data8[4480].a	5760 bytes Acc72EX[i].Data8[4480].a	5760 bytes Acc72EX[i].Data8[4480].a	5760 bytes Acc72EX[i].Data8[4480].a	5760 bytes Acc72EX[i].Data8[4480].a	5760 bytes Acc72EX[i].Data8[4480].a	5760 bytes Acc72EX[i].Data8[4480].a	5760 bytes Acc72EX[i].Data8[4480].a
Transfer Direction Transfer Type	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)	OUT (Host System to netX) DPM (Dual-Port Memory)
Handshake Mode Handshake Bit	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLE
- Subblock 6	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE	PROCESS DATA IMAGE
Size Start Offset	5760 bytes Acc72EX[i].Data8[10240].a	1536 bytes Acc72EX[i].Data8[6016].a	5760 bytes Acc72EX[i].Data8[10240].a	1536 bytes Acc72EX[i] Data8[6016] a	5760 bytes Acc72EX[i].Data8[10240].a	1536 bytes Acc72EX[i].Data8[6016].a	1536 bytes Acc72EX[i].Data8[6016].a	5760 bytes Acc72EX(i).Data8(10240).a	5760 bytes Acc72EX[i].Data8[10240].a					
Transfer Direction	IN (netX to Host System)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System) DPM (Dual-Port Memory)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)
Transfer Type Handshake Mode	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLE
Handshake Bit	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Subblock 7 Size	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes	HIGH PRIORITY DATA IMAGE 64 bytes
Start Offset Transfer Direction	Acc72EX[i].Data8[16000].a OUT (Host System to netX)	Acc72EX[i].Data8[7552].a OUT (Host System to netX)	Acc72EX[i].Data8[16000].a OUT (Host System to netX)	Acc72EX[i].Data8[7552].a OUT (Host System to netX)	Acc72EX[i].Data8[16000].a OUT (Host System to netX)	Acc72EX[i].Data8[7552].a OUT (Host System to netX)	Acc72EX[i].Data8[7552].a OUT (Host System to netX)	Acc72EX[i].Data8[16000].a OUT (Host System to netX)						
Transfer Type	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)	DPM (Dual-Port Memory)
Handshake Mode Handshake Bit	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLED 8	BUFFERED, HOST CONTROLLE 8
Subblock 8	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE	HIGH PRIORITY DATA IMAGE
Size Start Offset	64 bytes Acc72EX[i].Data8[16064].a	64 bytes Acc72EX[i].Data8[7616].a	64 bytes Acc72EX[i].Data8[16064].a	64 bytes Acc72EX[i].Data8[7616].a	64 bytes Acc72EX[i].Data8[16064].a	64 bytes Acc72EX(i).Data8(7616).a	64 bytes Acc72EX[i].Data8[7616].a	64 bytes Acc72EX[i].Data8[16064].a						
Transfer Direction	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)	IN (netX to Host System)
Transfer Type Handshake Mode	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED	DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED
Handshake Bit	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Block 3 Channel Type	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Communication	Undefined	Undefined	Undefined	Communication
Size of Channel Channel Start Address	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	15616 bytes Acc72EX[i].Data8[212992].a	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	0 bytes NOT AVAILABLE	15616 bytes Acc72EX[i].Data8[212992].a
Position of Handshake Cells	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	IN HANDSHAKE CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	BEGINNING OF CHANNEL	IN HANDSHAKE CHANNEL
Size of Handshake Cells NetX Handshake Register	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	16 BITS Acc72EX[i].Data8[524].a	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	NOT AVAILABLE NOT AVAILABLE	16 BITS Acc72EX[i].Data8[524].a
Host Handshake Register Communication Class	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	Acc72EX[i].Data8[526].a MESSAGING	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	NOT AVAILABLE UNDEFINED	Acc72EX[i].Data8[526].a MESSAGING
Protocol Class Conformance Class	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED
Number of Subblocks	0	0	0	0	0	0	0	0	0	9	0	0	0	9
Subblock 0										CONTROL				CONTROL
Size Start Offset										8 bytes Acc72EX[i].Data8[213000].a				8 bytes Acc72EX[i].Data8[213000].a
Transfer Direction Transfer Type										OUT (Host System to netX) DPM (Dual-Port Memory)				OUT (Host System to netX) DPM (Dual-Port Memory)
Handshake Mode										UNCONTROLLED				UNCONTROLLED
Handshake Bit										0				0
Subblock 1 Size										COMMON STATUS 64 bytes				COMMON STATUS 64 bytes
Start Offset Transfer Direction										Acc72EX[i].Data8[213008].a IN (netX to Host System)				Acc72EX[i].Data8[213008].a IN (netX to Host System)
Transfer Type										DPM (Dual-Port Memory)				DPM (Dual-Port Memory)
Handshake Mode Handshake Bit										UNCONTROLLED 0				UNCONTROLLED 0
Subblock 2										EXTENDED STATUS				EXTENDED STATUS
Size Start Offset										432 bytes Acc72EX[i].Data8[213072].a				432 bytes Acc72EX[i].Data8[213072].a
Transfer Direction										IN (netX to Host System)				IN (netX to Host System)
Transfer Type Handshake Mode										DPM (Dual-Port Memory) UNCONTROLLED				DPM (Dual-Port Memory) UNCONTROLLED
Handshake Bit										0				0
Subblock 3 Size										MAILBOX 1600 bytes				MAILBOX 1600 bytes
Start Offset										Acc72EX[i].Data8[213504].a				Acc72EX[i].Data8[213504].a
Transfer Direction Transfer Type										OUT (Host System to netX) DPM (Dual-Port Memory)				OUT (Host System to netX) DPM (Dual-Port Memory)
Handshake Mode Handshake Bit										BUFFERED, HOST CONTROLLED				BUFFERED, HOST CONTROLLED
										MAILBOX				* MAILBOX
Subblock 4 Size										1600 bytes				1600 bytes
Start Offset Transfer Direction										Acc72EX[i].Data8[215104].a IN (netX to Host System)				Acc72EX[i].Data8[215104].a IN (netX to Host System)
Transfer Type Handshake Mode										DPM (Dual-Port Memory) UNKNOWN				DPM (Dual-Port Memory) UNKNOWN
Handshake Bit										5				5
Subblock 5										PROCESS DATA IMAGE				PROCESS DATA IMAGE
Size Start Offset										5760 bytes Acc72EX[i].Data8[217088].a				5760 bytes Acc72EX[i].Data8[217088].a
Transfer Direction Transfer Type										OUT (Host System to netX) DPM (Dual-Port Memory)				OUT (Host System to netX) DPM (Dual-Port Memory)
Handshake Mode Handshake Bit										BUFFERED, HOST CONTROLLED				BUFFERED, HOST CONTROLLER
														0
Subblock 6 Size										PROCESS DATA IMAGE 5760 bytes				PROCESS DATA IMAGE 5760 bytes
Start Offset Transfer Direction										Acc72EX[i].Data8[222848].a IN (netX to Host System)				Acc72EX[i].Data8[222848].a IN (netX to Host System)
Transfer Type										DPM (Dual-Port Memory)				DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLEI
Handshake Mode Handshake Bit										BUFFERED, HOST CONTROLLED 7				BUFFERED, HOST CONTROLLE 7
Subblock 7										HIGH PRIORITY DATA IMAGE				HIGH PRIORITY DATA IMAGE
Size Start Offset										64 bytes Acc72EX[i].Data8[216704].a				64 bytes Acc72EX[i].Data8[216704].a
Transfer Direction										OUT (Host System to netX)				OUT (Host System to netX)
Transfer Type Handshake Mode										DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLED				DPM (Dual-Port Memory) BUFFERED, HOST CONTROLLE
Handshake Bit										8				8
Subblock 8 Size										HIGH PRIORITY DATA IMAGE 64 bytes				HIGH PRIORITY DATA IMAGE 64 bytes
Start Offset										Acc72EX[i].Data8[216768].a				Acc72EX[i].Data8[216768].a
Transfer Direction Transfer Type										IN (netX to Host System) DPM (Dual-Port Memory)				IN (netX to Host System) DPM (Dual-Port Memory)
Handshake Mode Handshake Bit										BUFFERED, HOST CONTROLLED 9				BUFFERED, HOST CONTROLLE 9
Block 4									Undefined					

	PROFIBUS-DP Master	PROFIBUS-DP Slave	DeviceNet Master	DeviceNet Slave	CANopen Master	CANopen Slave	CC-Link Slave	EtherCAT Master	EtherCAT Slave	EtherNet/IP	EtherNet/IP	Open Modbus/TCP	PROFINET IO	PROFINET IO
										Scanner/Master	Adapter/Slave		Controller/Master	Device/Slave
Channel Start Address	NOT AVAILABLE													
Position of Handshake Cells	BEGINNING OF CHANNEL													
Size of Handshake Cells	NOT AVAILABLE													
NetX Handshake Register	NOT AVAILABLE													
Host Handshake Register	NOT AVAILABLE													
Communication Class	UNDEFINED													
Protocol Class	UNDEFINED													
Conformance Class	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Number of Subblocks	0	0	0	0	0	0	0	0	0	0	0	0	0	0
llock 5														
Channel Type	Undefined													
Size of Channel	0 bytes													
Channel Start Address	NOT AVAILABLE													
Position of Handshake Cells	BEGINNING OF CHANNEL													
Size of Handshake Cells	NOT AVAILABLE													
NetX Handshake Register	NOT AVAILABLE													
Host Handshake Register	NOT AVAILABLE													
Communication Class	UNDEFINED													
Protocol Class	UNDEFINED													
onformance Class	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Number of Subblocks	0	0	0	0	0	0	0	0	0	0	0	0	0	0