ABB Low Voltage Drives:
Add-On Instruction(s) for Logix5000™ Controller Platform

Contents

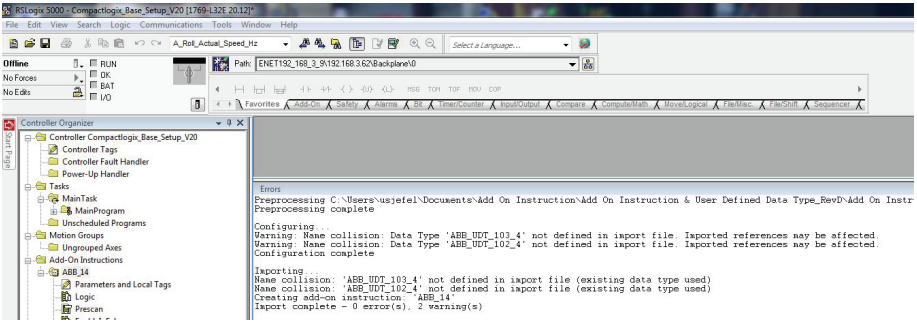
Trademarks

ControlNet™ is a trademark of ControlNet International, LTD.
DeviceNet™ is a trademark of the Open DeviceNet Vendor Association.
EtherNet/IP™ is a trademark of Open DeviceNet Vendor Association.
RSLogix™ 5000 and RSNetWorxTM are trademarks of Rockwell Software Inc.

Preface: Notices and Cautions

The caution below that might be seen in RSLogix5000 is providing the following information. The AOI that you are importing is using some UDT's that you have already defined. Be forewarned that the UDT's are not 'embedded' in the AOI and the imported AOI will 'attempt' to use what you already have installed.

The "Notice" or clarification here is that there are no errors. Rockwell Automation just wants to make sure you are aware that importing UDT's that are not compatible with an AOI might cause issues, so they issue the following warnings:



Section 1: General Information

Overview

This document will provide an overview on how to import and configure the ABB Add-on Instructions using RSLogix 5000™ / Studio 5000™ version 16 or later.

The ABB LVD Add-On Instructions are instructions that define the drives input and output assemblies. These instructions and data types can then be imported into a RSLogix 5000 project. Once defined in a project, they behave similarly to the built-in instructions already in the Logix controllers. An Add-On Instruction lets you encapsulate your most commonly used logic as sets of instructions. The instructions let you easily reuse ABB instructions in your projects. The instructions will promote consistency in all projects.

This document assumes that the reader has basic knowledge of; ControlNet or EtherNet/IP protocols, ABB drives and Rockwell RSLogix5000

Reference Documentation:

ACS355 User's Manual
3AUA0000066143

ACS550-U1 User's Manual
3AUA0000001609

ACS800 Firmware Manual
3AFE64527592

ACS850 Firmware manual
Standard Control Program
3AUA0000045497

ACSM1 Firmware Manual
Speed and Torque Control Program
3AFE68848261

ACS880 Firmware manual
Primary Control Program
3AUA0000085967

ACS380 Firmware Manual
Machinery Control Program
3AXD50000029275

DCS800 Firmware Manual
3ADW000193

FENA-01/11/21 Ethernet Adapter Module
User's Manual
3AUA0000093568

RETA-01 Ethernet/IP Adapter Module
User's Manual
3AFE64539736

FCNA-01 ControlNet Adapter Module
User's Manual
3AUA0000141650

RCNA-01 ControlNet Adapter Module
User's Manual
3AFE64506005

[This page intentionally left blank]

Section 2: Selecting an ABB Add-On Instruction

The library of Add-On Instructions and associated User Defined Data Types are organized by drive product line, fieldbus adapter, and the IO assembly instances for that product. A few of the I/O assembly Instance folders have more than one definition. This is to support a range of I/O configurations. Tables 1 – 6 below can be used to select the proper Add-On Instruction and User Defined Data Type for any application.

EtherNet/IP - Selecting an Add-On Instruction & User Defined Data Type

Table 1 - FENA-01/11/21: ACS350 /ACS355 /ACS850 /ACS880 /ACSM1 /ACS380 /ACH-S-Q580 – ODVA & ABB Profile*

ABB Add-On Instructions & User Defined Data Types - EtherNet/IP								
Input Assy. Instance	Output Assy. Instance	# of Bytes	Instance Size INT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
70	20	4	2	ODVA	ABB_AOI_70_20_2W	ABB_1	ABB_UDT_70	ABB_UDT_20
71	21	4	2	ODVA	ABB_AOI_71_21_2W	ABB_2	ABB_UDT_71	ABB_UDT_21
72	22	6	3	ODVA	ABB_AOI_72_22_3W	ABB_3	ABB_UDT_72	ABB_UDT_22
73	23	6	3	ODVA	ABB_AOI_73_23_3W	ABB_4	ABB_UDT_73	ABB_UDT_23
170	120	24	12	ODVA	ABB_AOI_170_120_12W	ABB_5	ABB_UDT_170	ABB_UDT_120
171	121	24	12	ODVA	ABB_AOI_171_121_12W	ABB_22	ABB_UDT_171_12	ABB_UDT_121_12
172	122	26	13	ODVA	ABB_AOI_172_122_13W	ABB_7	ABB_UDT_172	ABB_UDT_122
173	123	26	13	ODVA	ABB_AOI_173_123_13W	ABB_8	ABB_UDT_173	ABB_UDT_123
51	1	4	2	ABB	ABB_AOI_51_1_2W	ABB_9	ABB_UDT_51	ABB_UDT_1
52	2	6	3	ABB	ABB_AOI_52_2_3W	ABB_11	ABB_UDT_52	ABB_UDT_2
151	101	24	12	ABB	ABB_AOI_151_101_12W	ABB_12	ABB_UDT_151	ABB_UDT_101o
152	102	26	13	ABB	ABB_AOI_152_102_13W	ABB_13	ABB_UDT_152	ABB_UDT_102

Table 2 - FENA-01/21: ACS355 – 32 Bit - Standard Transparent (DCU Profile)

ABB Add-On Instructions & User Defined Data Types – ACS355 - EtherNet/IP								
Input Assy. Instance	Output Assy. Instance	# of Bytes	Instance Size DINT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
71	21	8	2	S. Trans	ABB_ACS355_ AOI_71_21_2W_32B	ABB_57	ABB_ACS355_ UDT_71_32	ABB_ACS355_ UDT_21_32
171	121	28	7	S. Trans	ABB_ACS355_ AOI_171_121_12W_32B	ABB_58	ABB_ACS355_ UDT_171_32	ABB_ACS355_ UDT_121_32
72	22	12	3	S. Trans	ABB_ACS355_ AOI_72_22_3W_32B	ABB_59	ABB_ACS355_ UDT_72_32	ABB_ACS355_ UDT_22_32
172	122	32	8	S. Trans	ABB_ACS355_ AOI_172_122_13W_32B	ABB_60	ABB_ACS355_ UDT_172_32	ABB_ACS355_ UDT_122_32

* ACS880/ACH/S/Q580 only supports FENA-11/21

Table 3 - FENA-11/21: ACS850 / ACSM1 – 32 Bit - Standard Transparent & Position Transparent*
*(Position Transparent for ACSM1 only)

ABB Add-On Instructions & User Defined Data Types – ACS850 / ACSM1 - EtherNet/IP								
Input Assy. Instance	Output Assy. Instance	# of Bytes	Instance Size DINT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
71	21	8	2	S. Trans	ABB_AOI_71_21_2W_32B	ABB_18	ABB_UDT_71_32	ABB_UDT_21_32
171	121	28	7	S. Trans	ABB_AOI_171_121_12W_32B	ABB_19	ABB_UDT_171_32	ABB_UDT_121_32
72	22	12	3	S. Trans	ABB_AOI_72_22_3W_32B	ABB_20	ABB_UDT_72_32	ABB_UDT_22_32
172	122	32	8	S. Trans	ABB_AOI_172_122_13W_32B	ABB_21	ABB_UDT_172_32	ABB_UDT_122_32
71	21	8	2	P. Trans	ABB_AOI_71_21_2W_32B_ Pos*	ABB_45	ABB_UDT_71_32_ Pos	ABB_ UDT_21_32_Pos
171	121	28	7	P. Trans	ABB_AOI_171_121_12W_32B_ Pos*	ABB_46	ABB_UDT_171_32_ Pos	ABB_ UDT_121_32_Pos
72	22	12	3	P. Trans	ABB_AOI_72_22_3W_32B_ Pos*	ABB_47	ABB_UDT_72_32_ Pos	ABB_ UDT_22_32_Pos
172	122	32	8	P. Trans	ABB_AOI_172_122_13W_32B_ Pos*	ABB_48	ABB_UDT_172_32_ Pos	ABB_ UDT_122_32_Pos

Table 4 - FENA-11/21: ACS880 Standard Transparent - 32 Bit

ABB Add-On Instructions & User Defined Data Types – ACS880 - EtherNet/IP								
Input Assy. Instance	Output Assy. Instance	# of Bytes	Instance Size DINT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
71	21	8	2	S. Trans 32 bit	ABB_AOI_71_21_2W_32B	ABB_53	ABB_ACS880_ UDT_71_32	ABB_ACS880_ UDT_21_32
171	121	24	6	S. Trans 32 bit	ABB_AOI_171_121_12W_32B	ABB_54	ABB_ACS880_ UDT_171_32	ABB_ACS880_ UDT_121_32
72	22	12	3	S. Trans 32 bit	ABB_AOI_72_22_3W_32B	ABB_55	ABB_ACS880_ UDT_72_32	ABB_ACS880_ UDT_22_32
172	122	24	6	S. Trans 32 bit	ABB_AOI_172_122_12W_32B	ABB_56	ABB_ACS880_ UDT_172_32	ABB_ACS880_ UDT_122_32

Table 5 - FENA-11/21: ACS880 Standard Transparent - 16 bit

ABB Add-On Instructions & User Defined Data Types – ACS880 - EtherNet/IP								
Input Assy. Instance	Output Assy. Instance	# of Bytes	Instance Size DINT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
162	112	26	13	S. Trans 16 bit	ABB_FENA_ AOI_162_112_13W	ABB_75	ABB_FENA_ UDT_162	ABB_FENA_ UDT_112

Table 6 - RETA-01: ACH550 / ACS550 / ACQ550 / ACS800 / ACQ800 – ODVA & ABB Profile

ABB Add-On Instructions & User Defined Data Types - EtherNet/IP								
Input Assy. Instance	Output Assy. Instance	# of Bytes	Instance Size INT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
70	20	4	2	ODVA	ABB_AOI_70_20_2W	ABB_1	ABB_UDT_70	ABB_UDT_20
71	21	4	2	ODVA	ABB_AOI_71_21_2W	ABB_2	ABB_UDT_71	ABB_UDT_21
171	121	12	6	ODVA	ABB_AOI_171_121_6W	ABB_6	ABB_UDT_171_6	ABB_UDT_121_6
171	121	24	12	ODVA	ABB_AOI_171_121_12W	ABB_22	ABB_ UDT_171_12	ABB_UDT_121_12
171	121	30	15	ODVA	ABB_AOI_171_121_15W only ACx550	ABB_50	ABB_ UDT_171_15	ABB_UDT_121_15
101	100	4	2	ABB	ABB_AOI_101_100_2W	ABB_10	ABB_UDT_101i	ABB_UDT_100
103	102	8	4	ABB	ABB_AOI_103_102_4W	ABB_14	ABB_UDT_103_4	ABB_UDT_102_4
103	102	18	9	ABB	ABB_AOI_103_102_9W	ABB_15	ABB_UDT_103_9	ABB_UDT_102_9
103	102	24	12	ABB	ABB_AOI_103_102_12W	ABB_16	ABB_ UDT_103_12	ABB_UDT_102_12
103	102	30	15	ABB	ABB_AOI_103_102_15W only ACx550	ABB_17	ABB_ UDT_103_15	ABB_UDT_102_15

Table 7 - RETA-01: DCS800 – ODVA & ABB Profile

ABB Add-On Instructions & User Defined Data Types - EtherNet/IP								
Input Assy. Instance	Output Assy. Instance	# of Bytes	Instance Size INT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
101	100	4	2	ABB	ABB_DCS800_ AOI_101_100_2W	ABB_37	ABB_DCS800_ UDT_101i	ABB_DCS800_ UDT_100
103	102	8	4	ABB	ABB_DCS800_ AOI_103_102_4W	ABB_38	ABB_DCS800_ UDT_103_4	ABB_DCS800_ UDT_102_4
103	102	18	9	ABB	ABB_DCS800_ AOI_103_102_9W	ABB_39	ABB_DCS800_ UDT_103_9	ABB_DCS800_ UDT_102_9
103	102	24	12	ABB	ABB_DCS800_ AOI_103_102_12W	ABB_40	ABB_DCS800_ UDT_103_12	ABB_DCS800_ UDT_102_12
70	20	4	2	ODVA	ABB_AOI_70_20_2W	ABB_1	ABB_UDT_70	ABB_UDT_20
71	21	4	2	ODVA	ABB_AOI_71_21_2W	ABB_2	ABB_UDT_71	ABB_UDT_21
171	121	12	6	ODVA	ABB_AOI_171_121_6W	ABB_6	ABB_UDT_171_6	ABB_ UDT_121_6
171	121	24	12	ODVA	ABB_AOI_171_121_12W	ABB_22	ABB_UDT_171_12	ABB_ UDT_121_12

Table 8 - ACS2000 ABB Profile EtherNet/IP

ABB Add-On Instructions & User Defined Data Types - EtherNet/IP								
Input Assy. Instance	Output Assy. Instance	# of Bytes	Instance Size INT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
100	150	24	12	N/A	ABB_AOI_100_150_12W	ABB_80	ABB_UDT_100	ABB_UDT_150o

ControlNet - Selecting an Add-On Instruction & User Defined Data Type

Table 9 - FCNA-01: ACS355/ ACS850 / ACSM1 / ACS880 / ACH-S-Q580 – ODVA & ABB Profile

ABB Add-On Instructions & User Defined Data Types - ControlNet								
Input Assy. Instance	Output Assy. Instance	# of Bytes IN;OUT	Instance Size IN;OUT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
70	20	8;4	4;2	ODVA	ABB_RCNA_AOI_70_20_2W	ABB_23	ABB_RCNA_ UDT_70	ABB_RCNA_ UDT_20
71	21	8;4	4;2	ODVA	ABB_RCNA_AOI_71_21_2W	ABB_24	ABB_RCNA_ UDT_71	ABB_RCNA_ UDT_21
171	121	16;12	8;6	ODVA	ABB_RCNA_AOI_171_121_6W	ABB_25	ABB_RCNA_ UDT_171_6	ABB_RCNA_ UDT_121_6
171	121	22;18	11;9	ODVA	ABB_RCNA_AOI_171_121_9W	ABB_26	ABB_RCNA_ UDT_171_9	ABB_RCNA_ UDT_121_9
101	100	8;4	4;2	ABB	ABB_DCS800_RCNA_ AOI_101_100_2W	ABB_32	ABB_DCS800_ RCNA_UDT_101i	ABB_ DCS800_ RCNA_ UDT_100
103	102	12;8	6,4	ABB	ABB_DCS800_RCNA_ AOI_103_102_4W	ABB_33	ABB_DCS800_ RCNA_ UDT_103_4	ABB_ DCS800_ RCNA_ UDT_102_4
103	102	22;18	11,9	ABB	ABB_DCS800_RCNA_ AOI_103_102_9W	ABB_34	ABB_DCS800_ RCNA_ UDT_103_9	ABB_ DCS800_ RCNA_ UDT_102_9
103	102	28;24	14,12	ABB	ABB_DCS800_RCNA_ AOI_103_102_12W	ABB_35	ABB_DCS800_ RCNA_ UDT_103_12	ABB_ DCS800_ RCNA_ UDT_102_12
51	1	8;4	4;2	ABB	ABB_FCNA_AOI_51_1_2W	ABB_69	ABB_FCNA_ UDT_51	ABB_FCNA_ UDT_1
52	2	10;6	5;3	ABB	ABB_FCNA_AOI_52_2_3W	ABB_71	ABB_FCNA_ UDT_52	ABB_FCNA_ UDT_2
151	101	28;24	14;12	ABB	ABB_FCNA_AOI_151_101_12W	ABB_72	ABB_FCNA_ UDT_151	ABB_FCNA_ UDT_101o
152	102	30;26	15;13	ABB	ABB_FCNA_AOI_152_102_13W	ABB_73	ABB_FCNA_ UDT_152	ABB_FCNA_ UDT_102

Table 10 - FCNA-01: ACS880 Transparent 16 bit

ABB Add-On Instructions & User Defined Data Types - ControlNet								
Input Assy. Instance	Output Assy. Instance	# of Bytes IN;OUT	Instance Size IN;OUT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
162	112	30;26	15;13	S. Trans 16 bit	ABB_FCNA_AOI_162_112_13W	ABB_74	ABB_FCNA_ UDT_162	ABB_FCNA_ UDT_112

Table 11 - RCNA-01: ACH550 / ACS550 / ACQ550 / ACS800 / ACQ800 – ODVA & ABB Profile

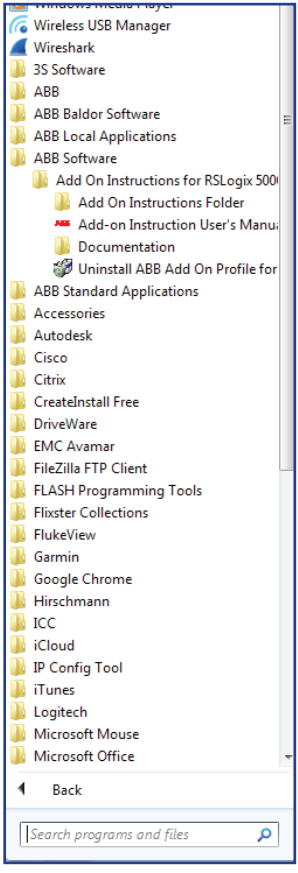
ABB Add-On Instructions & User Defined Data Types - ControlNet								
Input Assy. Instance	Output Assy. Instance	# of Bytes IN;OUT	Instance Size INT IN;OUT	Profile Type	Add-On Instruction		User Defined Data Type	
					File Name	Instr. Name	Input	Output
70	20	8;4	4;2	ODVA	ABB_RCNA_AOI_70_20_2W	ABB_23	ABB_RCNA_ UDT_70	ABB_RCNA_ UDT_20
71	21	8;4	4;2	ODVA	ABB_RCNA_AOI_71_21_2W	ABB_24	ABB_RCNA_ UDT_71	ABB_RCNA_ UDT_21
171	121	16;12	8;6	ODVA	ABB_RCNA_AOI_171_121_6W	ABB_25	ABB_RCNA_ UDT_171_6	ABB_RCNA_ UDT_121_6
171	121	22;18	11;9	ODVA	ABB_RCNA_AOI_171_121_9W	ABB_26	ABB_RCNA_ UDT_171_9	ABB_RCNA_ UDT_121_9
101	100	8;4	4;2	ABB	ABB_RCNA_AOI_101_100_2W	ABB_27	ABB_RCNA_ UDT_101i	ABB_RCNA_ UDT_100
103	102	12;8	6;4	ABB	ABB_RCNA_AOI_103_102_4W	ABB_28	ABB_RCNA_ UDT_103_4	ABB_RCNA_ UDT_102_4
103	102	22;18	11;9	ABB	ABB_RCNA_AOI_103_102_9W	ABB_29	ABB_RCNA_ UDT_103_9	ABB_RCNA_ UDT_102_9
103	102	28;24	14;12	ABB	ABB_RCNA_AOI_103_102_12W	ABB_30	ABB_RCNA_ UDT_103_12	ABB_RCNA_ UDT_102_12
103	102	34;30	17;15	ABB	ABB_RCNA_AOI_103_102_15W only ACx550	ABB_31	ABB_RCNA_ UDT_103_15	ABB_RCNA_ UDT_102_15

Table 12 - RCNA-01: DCS800 – ODVA & ABB Profile

ABB Add-On Instructions & User Defined Data Types - ControlNet								
Input Assy. Instance	Output Assy. Instance	# of Bytes	Instance Size INT IN;OUT	Profile Type	Add-On Instruction		User Defined Data Type	
					File name	Instr. Name	Input	Output
70	20	8;4	4;2	ODVA	ABB_RCNA_AOI_70_20_2W	ABB_23	ABB_RCNA_ UDT_70	ABB_RCNA_ UDT_20
71	21	8;4	4;2	ODVA	ABB_RCNA_AOI_71_21_2W	ABB_24	ABB_RCNA_ UDT_71	ABB_RCNA_ UDT_21
171	121	16;12	8;6	ODVA	ABB_RCNA_ AOI_171_121_6W	ABB_25	ABB_RCNA_ UDT_171_6	ABB_RCNA_ UDT_121_6
171	121	22;18	11;9	ODVA	ABB_RCNA_ AOI_171_121_9W	ABB_26	ABB_RCNA_ UDT_171_9	ABB_RCNA_ UDT_121_9
101	100	8;4	4;2	ABB	ABB_DCS800_RCNA_ AOI_101_100_2W	ABB_32	ABB_DCS800_ RCNA_UDT_101i	ABB_DCS800_ RCNA_UDT_100
103	102	12;8	6,4	ABB	ABB_DCS800_RCNA_ AOI_103_102_4W	ABB_33	ABB_DCS800_ RCNA_UDT_103_4	ABB_DCS800_ RCNA_UDT_102_4
103	102	22;18	11,9	ABB	ABB_DCS800_RCNA_ AOI_103_102_9W	ABB_34	ABB_DCS800_ RCNA_UDT_103_9	ABB_DCS800_ RCNA_UDT_102_9
103	102	28;24	14,12	ABB	ABB_DCS800_RCNA_ AOI_103_102_12W	ABB_35	ABB_DCS800_ RCNA_UDT_103_12	ABB_DCS800_ RCNA_UDT_102_12

Navigating the Add-On Instruction & User Defined Data Type Library

After running the AOI executable from the CD, you should find the Add On Instruction and User defined data types under the Windows start menu in folder: ABB Software/Add On Instruction for RSLogix 5000. As an example, see the graphic below, should you need to manually install the folders and files please follow the example below:

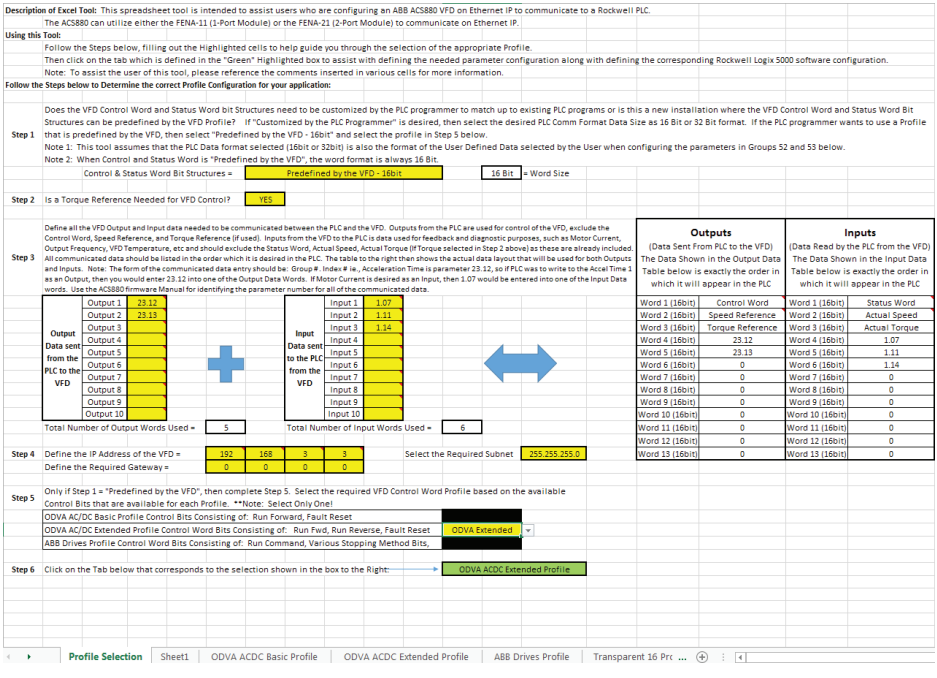


Navigating the Add-On Instruction & User Defined Data Type Library

The graphic below shows the file structure used to organize the library of Add-On Instructions. After selecting the drive type, fieldbus adapter type and the desired I/O Assembly the required Add-On Instructions and User Defined Data Type will be contained in one folder.

ACS880 Configuration Tool

Below is a picture of the ACS880 Configuration tool. It can be downloaded from our connectivity site www.abbnow.com/abbconnectivity. This tool was designed to assist in configuring the ACS880 drive on EtherNet/IP to a Rockwell PLC. There are 6 steps to this tool. Step 1 is to select the control and status bit structure. It can be either predefined 16 bit (ODVA or ABB drives profile), or customized 16 bit or customized 32 bit (transparent mode). Step 2 is to decide if a second reference is needed (ex. Torque or Ext 2 reference). Step 3 is to fill out what data is to be sent to the drive (output data ex. Accel time, Decel time, preset speed, etc.) and what is to be read from the drive (input data ex. Motor current, DC bus voltage, etc.). The control word and speed reference are automatically setup to be sent to the drive as well as the status word and speed feedback, so it is not necessary to put these in the table. Step 4 is to define the IP address, subnet mask, and gateway (if used). Step 5 is to select what profile is to be used by the drive (ODVA Basic, ODVA Extended, or ABB Drives Profile). Step 6, click on the corresponding tab for the profile that you choose.



After the tab is clicked to match what was selected in Step 6, the following page will be shown. This shows all of the parameters that need to be set in the drive to setup the communications based on the information put into the tool. It also creates a snap shot of the Generic Ethernet Module that needs to be created inside the Rockwell software and shows the control and status structure for the profile that was selected from the profile tab of the tool.

EIP ODVA AC/DC Drive Extended Profile

VFD Signals Controlled by PLC over EIP			
Signal	**No/Yes**	**Parameter**	**Parameter Setting**
Run Command	YES	20.01	Fieldbus A
Fault Reset	YES	31.11	"No Setting Required"
Speed Reference	YES	22.11	FB A ref1
Direction Command	YES	20.24	Selected
Torque Reference	NO	26.11	"User Defines Setting"

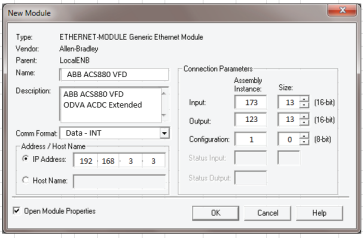
Parameters for Configuring User Defined Data Words			
Outputs from PLC to VFD		**Inputs to PLC from VFD**	
Parameter Setting	**Parameter**	**Parameter**	**Parameter Setting**
23.12	53.01	52.01	1.07
23.13	53.02	52.02	1.11
0.00	53.03	52.03	1.14
0.00	53.04	52.04	0.00
0.00	53.05	52.05	0.00
0.00	53.06	52.06	0.00
0.00	53.07	52.07	0.00
0.00	53.08	52.08	0.00
0.00	53.09	52.09	0.00
0.00	53.10	52.10	0.00

FENA-11/21 Configuration Parameters (Group 50 & 51)		
Parameter Name	**Parameter**	**Parameter Setting**
FBA A enable	50.01	Option Slot 1
FBA A comm loss func	50.02	Fault
FBA A Type	51.01	Ethernet
Portocol / Profile	51.02	100 (EIP AC/DC)
IP configuration	51.04	0 (Static)
IP Address 1	51.05	192
IP Address 2	51.06	168
IP Address 3	51.07	3
IP Address 4	51.08	3
Subnet CIDR	51.09	24
GW Address 1	51.10	0
GW Address 2	51.11	0
GW Address 3	51.12	0
GW Address 4	51.13	0
Control Timeout	51.20	20
Idle Action	51.21	0 (Off-line)
ODVA Stop Function	51.22	Ramp
ODVA Speed Scale	51.23	128
ODVA Torque Scale	51.24	128
FBA Par Refresh	51.27	Refresh

Other VFD Parameters that Accompany Communication Control		
Parameter Name	**Parameter**	**Parameter Setting**
Acceleration time 1	23.12	5.00 s
Deceleration time 1	23.13	5.00 s
Minimum Speed	30.11	0 rpm
Maximum Speed	30.12	1800 rpm
Speed Scaling	46.01	1800 rpm

Rockwell Logix 5000 Software Configuration to add ABB ACS880 as a new Ethernet IP Device

Note: The configuration below is specific to the ODVA AC/DC Extended Profile and the data supplied on the "Profile Selection" Tab of this Spreadsheet Tool. This PLC configuration matches the parameters defined on this page and will deliver successful Communications between the PLC and the ABB ACS880 VFD via Ethernet IP



ODVA Extended Control Word Bit Structure

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
1								

ODVA Extended Status Word Bit Structure

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	Ctrl From Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
1	Drive State See section *State (Control supervisor object)* on page *181*.							

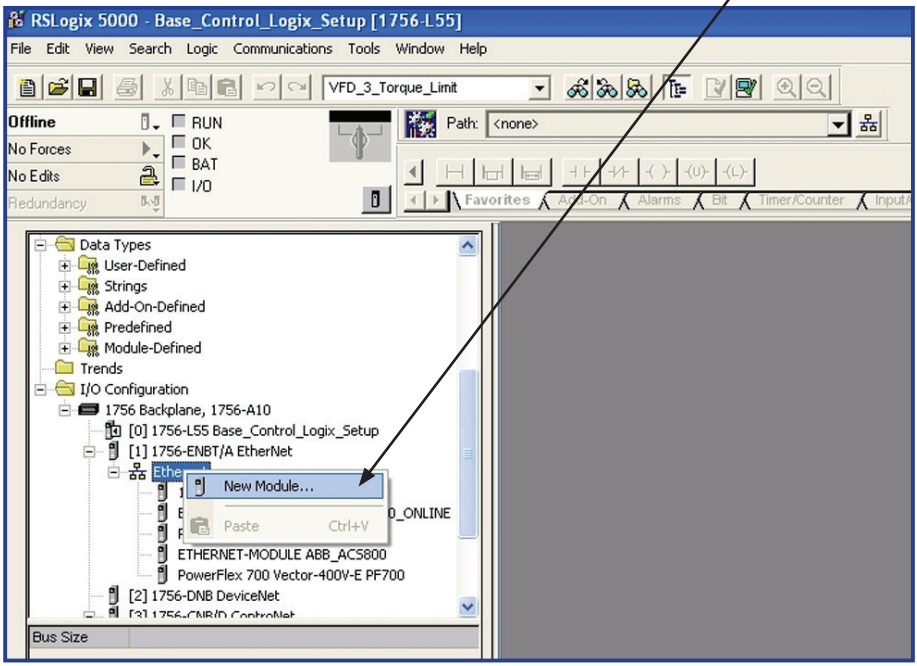
Section 3: Importing User Defined Data Type and Add-On Instruction.

NOTE! The User Defined Data Type must be imported before the Add-On Instruction.

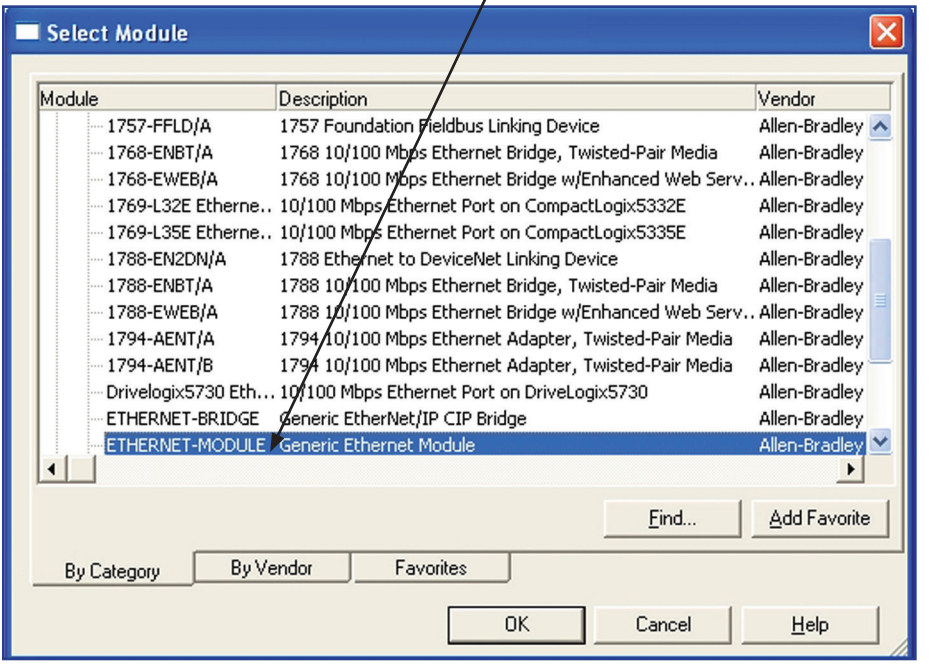
Creating a Generic module and importing User Defined Data Types and Add-On Instruction

For this example a ControlLogix PLC was used. These basic instructions can be used for any A-B PLC that uses RSLogix5000 and supports ControlNet or EtherNet/IP. See the ABB document; Basic Guide to Communication, ODVA for additional examples.

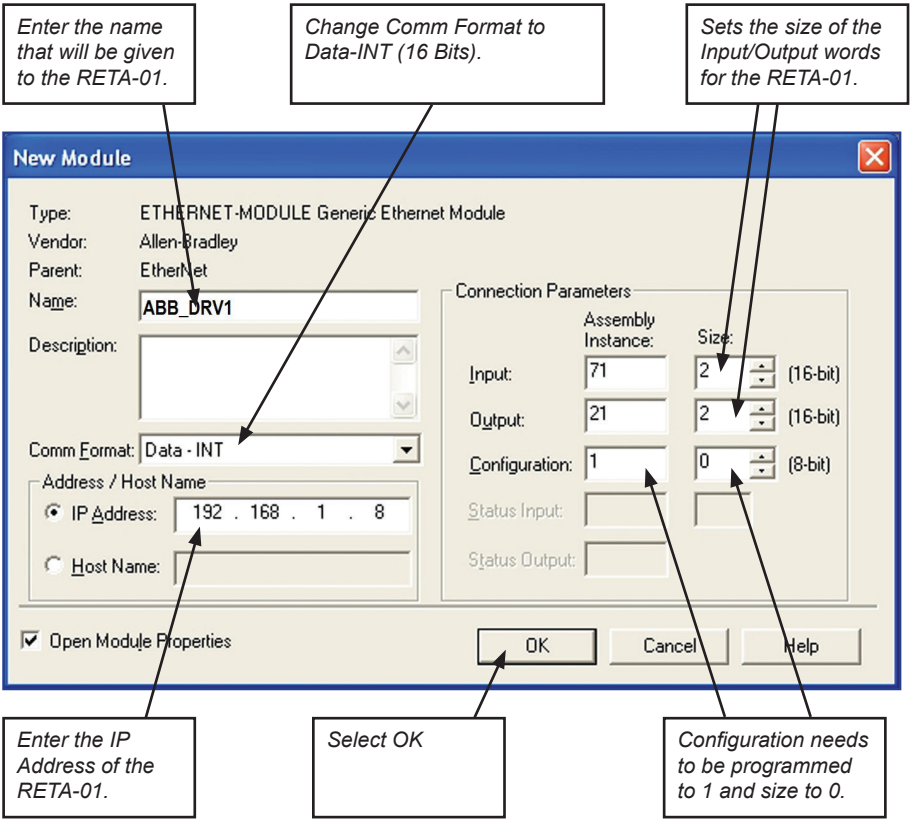
1. Open RSLogix 5000 and open a RSLogix 5000 program. Right Click on the 1769-L32E Ethernet Port LocalENB and Select New Module.



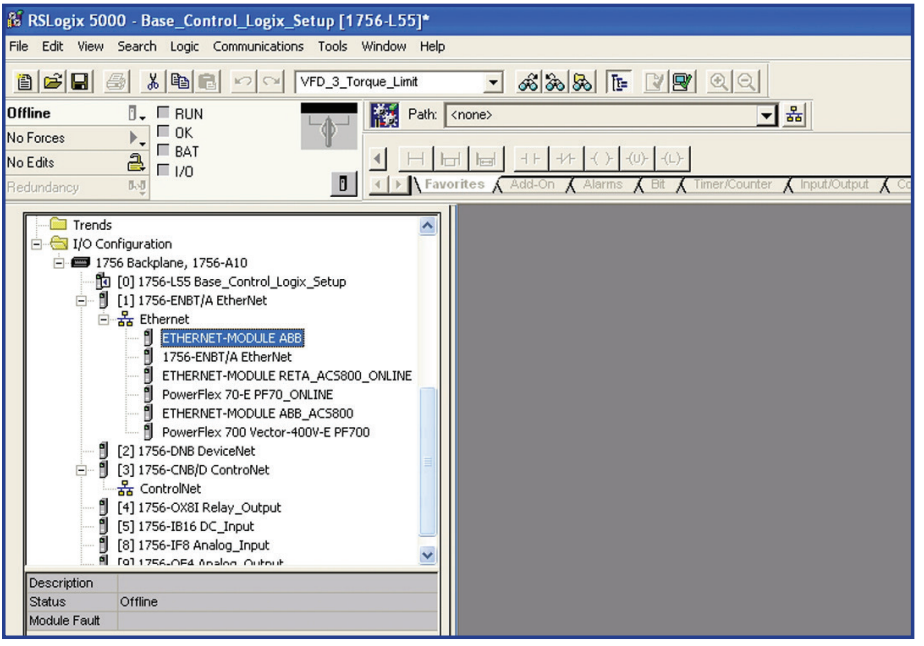
2. Under Communications, select ETHERNET-MODULE.



3. Program the following information below. The example below is using the Generic Drive Profile Input Assembly Instance 71 and Output Assembly Instance 21.

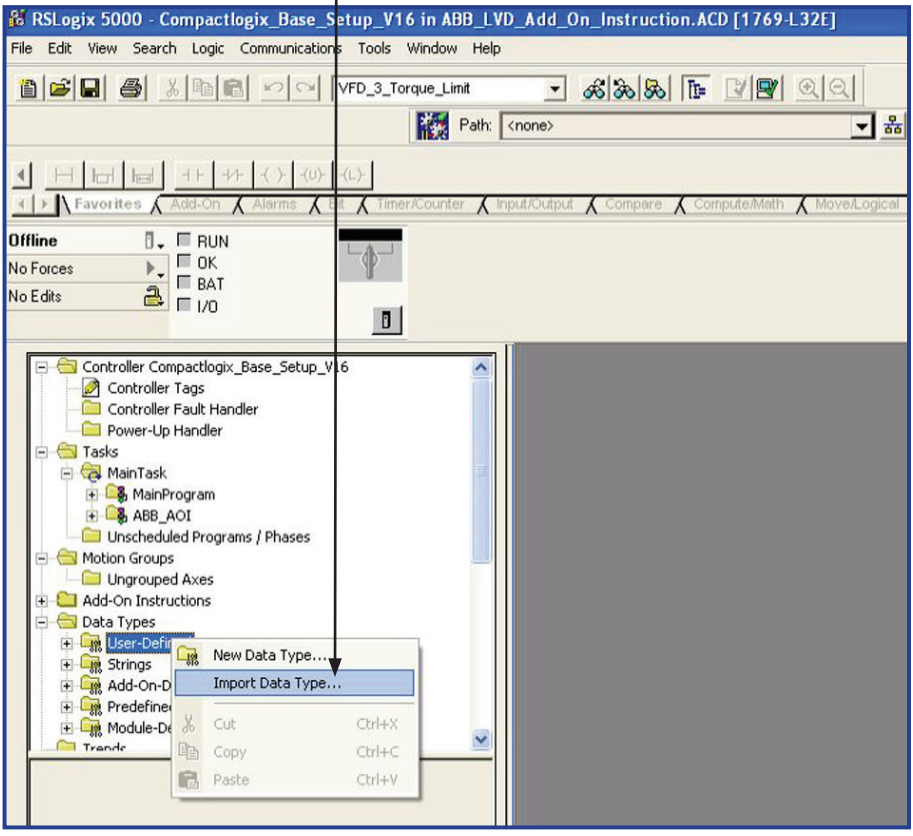


4. The drive has now been added to the PLC I/O.
   Repeat steps 1-5 for additional drives.



Importing ABB User Defined Data Type

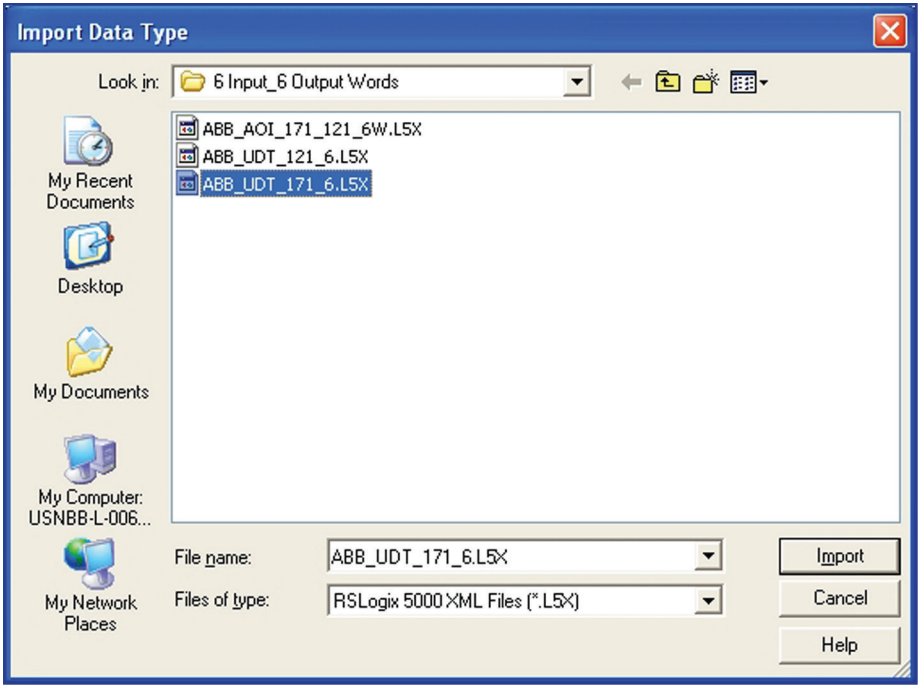
1. Right Click the User Defined folder in the Data Type directory. Then, select Import Data Type.



Importing ABB User Defined Data Type (continued)

2. Browse to the location of the ABB User Defined Data Type library and select the desired input User Defined Data Type from Section 2 and click Import.

NOTE! ***The ABB User Defined Data Type you select must match the module I/O Assembly Instance size and type from section 2.***

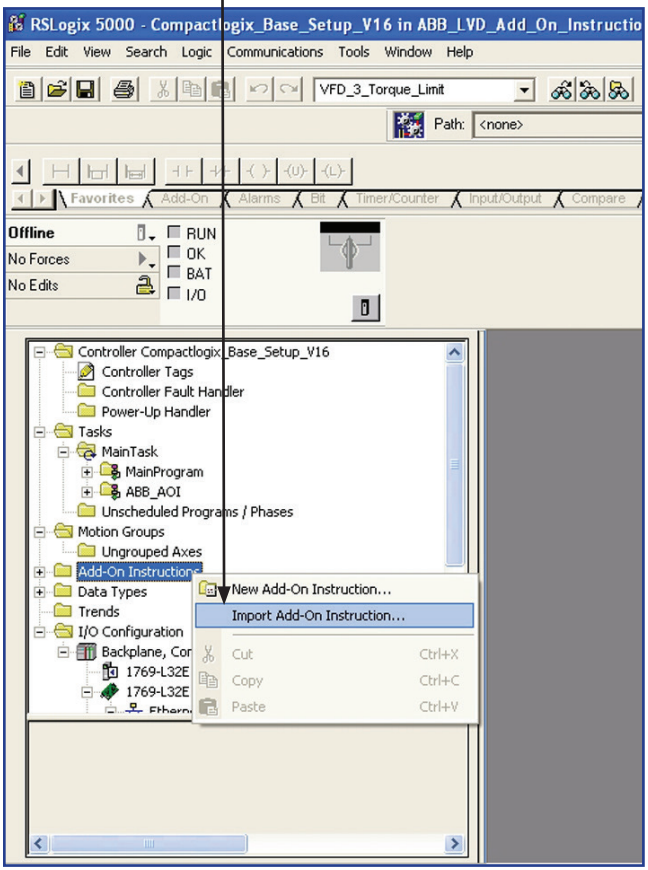


3. Repeat this step for the desired output User Defined Data Type.

Importing ABB Add-On Instruction

NOTE! ***The User Defined Data Type must be imported before the Add-On Instruction.***

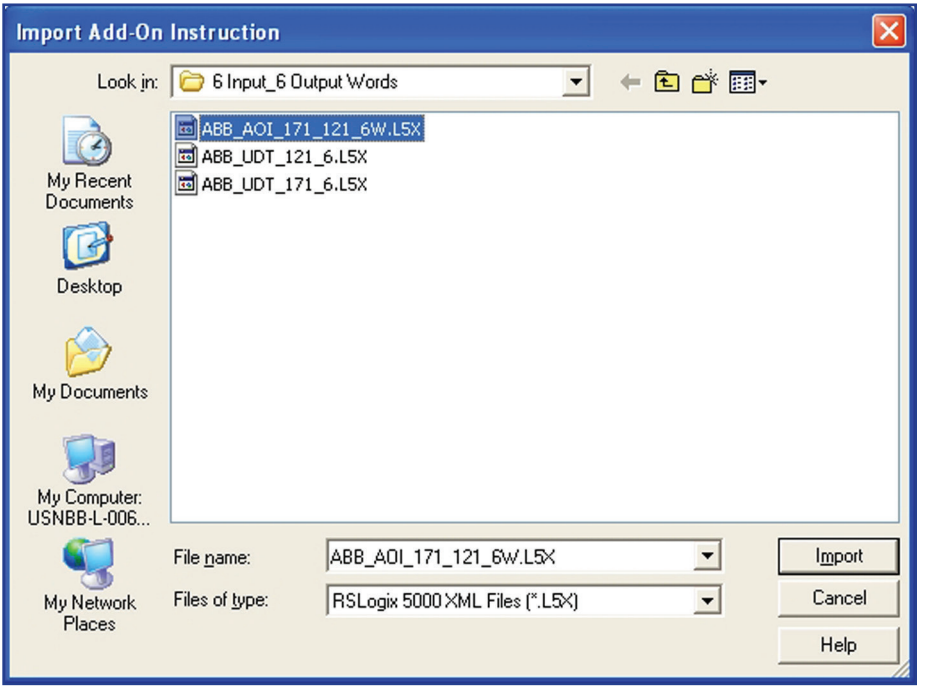
4. Right Click on the Add-On Instruction folder and select Import Add-On Instruction.



Importing ABB Add-On Instruction (continued)

5. Browse to the location of the ABB Add-On Instruction (AOI) library and select the desired AOI from section 2 and click Import.

NOTE! ***The ABB AOI you select must match the module I/O Assembly Instance size and type from section 2.***



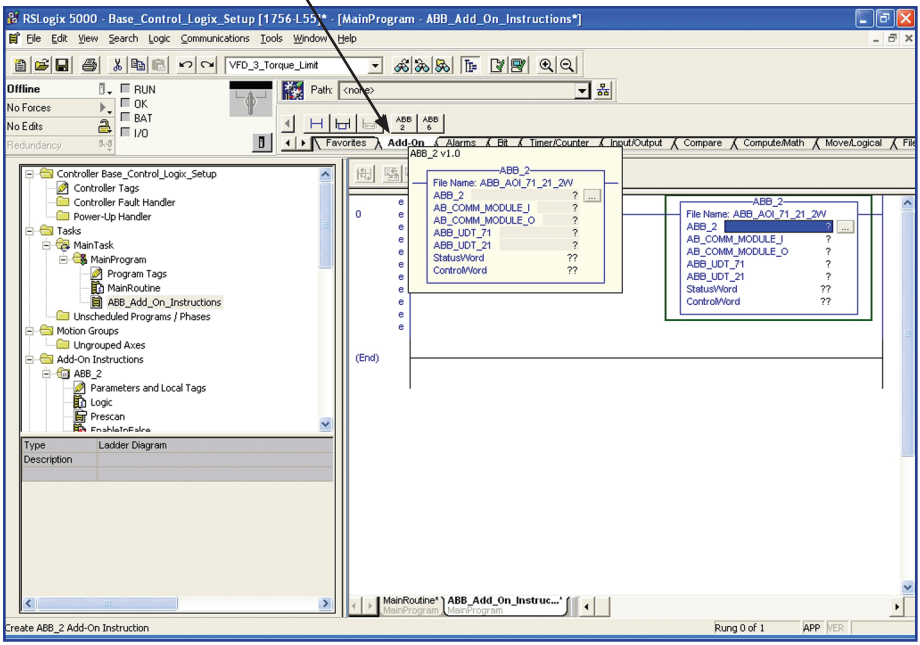
6. Repeat this step for each I/O assembly instance the application requires.

Section 4: Using User Defined Data Type and Add-On Instruction

For this example a ControlLogix PLC was used. These basic instructions can be used for any A-B PLC that uses RSLogix5000 and supports ControlNet or EtherNet/IP. See the ABB document; Basic Guide to Communication, ODVA for additional examples.

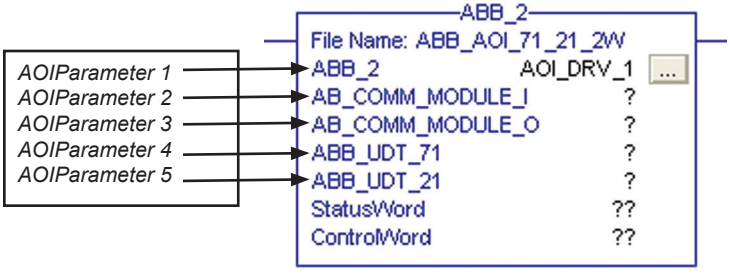
Using the Add-On Instruction in PLC program.

1. Insert one ABB Add-On Instruction into your PLC program for each ABB drive in the system. Do this by selecting the Add-On Instruction from the element group: Add-On.

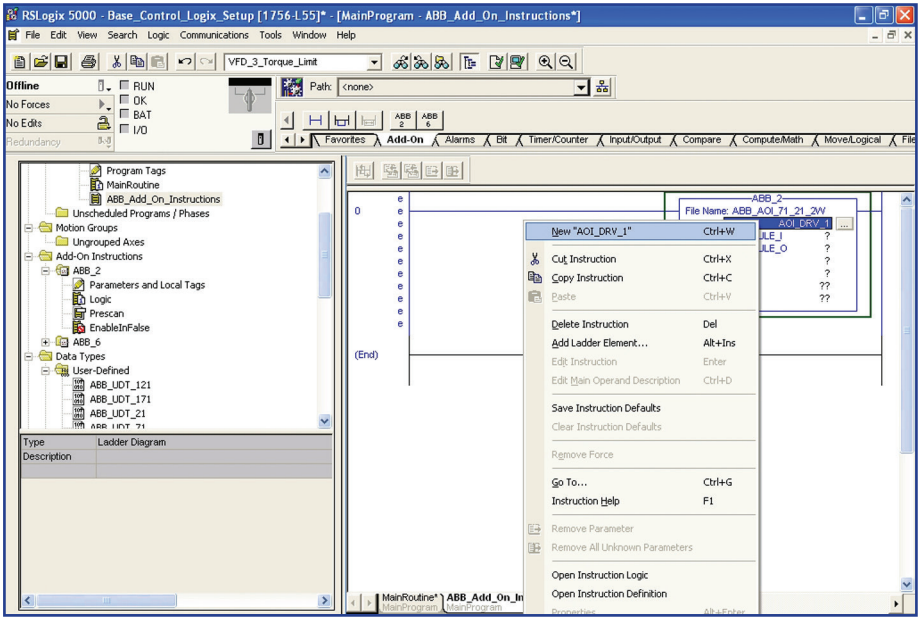


2. To complete the configuration of the Add-On Instructions create new controller tags for each Add-On Instruction parameter.

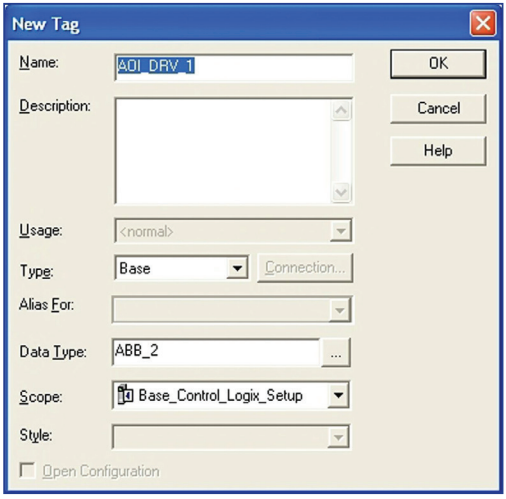
   A. Enter a unique name for AOI Parameter 1.



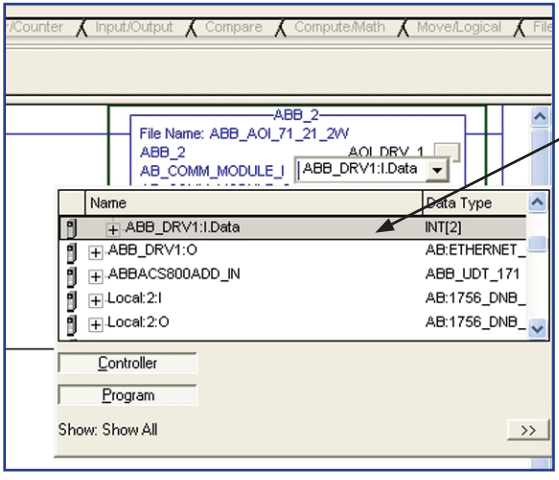
   B. Right Click the parameter name and select New... to open a dialogue box to define the new tag name.



C. Verify that the Type is Base, Data Type is the desired ABB Add-On Instruction and that the scope is selected correctly for the application.

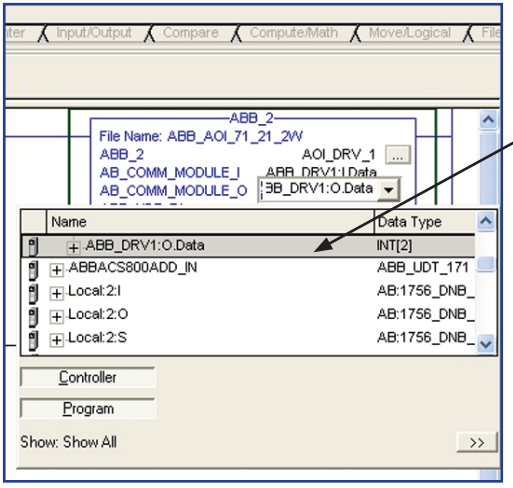


D. Next define AOI parameters 2 and 3. This will link the Generic module to the User Defined Data Type. Double Click AOI parameter AB_COMM_MODULE_I.



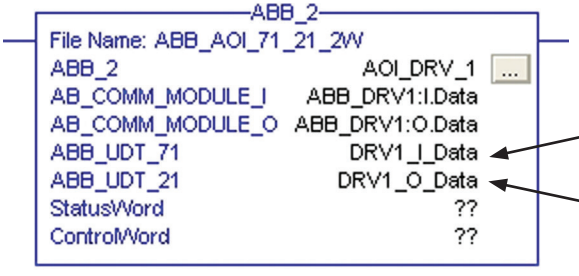
Using the drop down menu browse to the Generic module Input that was created in section 4. Select the input module at the level shown to the right.

E. Repeat the same steps for AOI parameter 3 AB_COMM_MODULE_O



Using the drop down menu browse to the Generic module Output that was created in section 4. Select the input module at the level shown to the right.

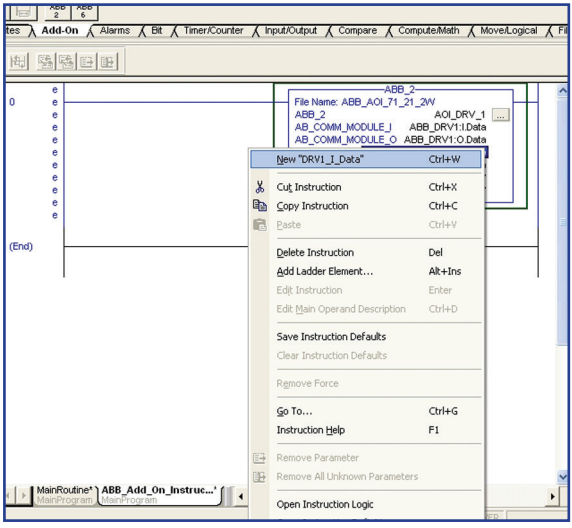
F. Then define the final two AOI parameters 4 and 5. Doing this will link the Generic module I/O data to the ABB User Defined Data Types that were imported in Section 3.



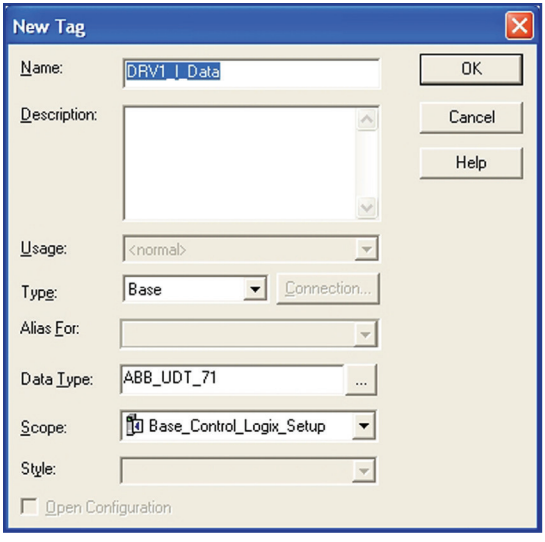
Enter a unique tag name for parameter 4.

Enter a unique tag name for parameter 5.

G. Right Click AOI parameter 4 and select New...
to open a dialogue box to define the new tag name.

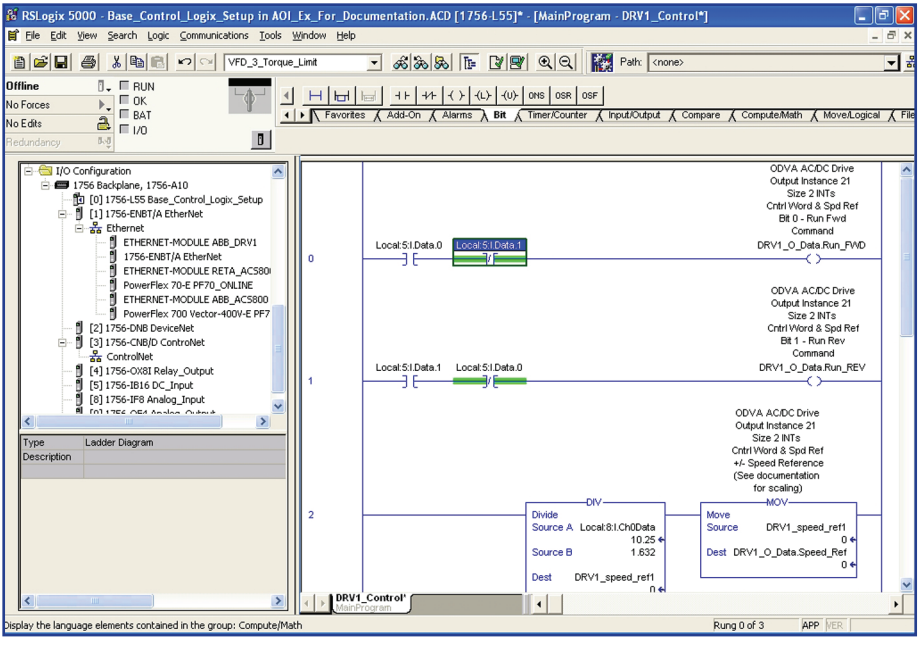


H. Verify that the Type is Base, Data Type is the desired ABB User Defined Data Type and that the scope is selected correctly for the application.



I. Repeat steps G and H for AOI parameter 5.

The ABB Add-On Instruction and User Defined Data Types are ready to be used within the PLC program. Use the User Defined Data Type tag names in the PLC program.



Section 5: Configuring Dynamic User Defined Data Types

NOTE! ***Changing the following user defined data type member names is not allowed:***
- ***Control Word***
- ***Reference 1***
- ***Reference 2***

Doing so will create an incompatiblity with the associated Add-On Instruction.

This section will describe how to customize a user defined data type for a dynamic I/O assembly instance.

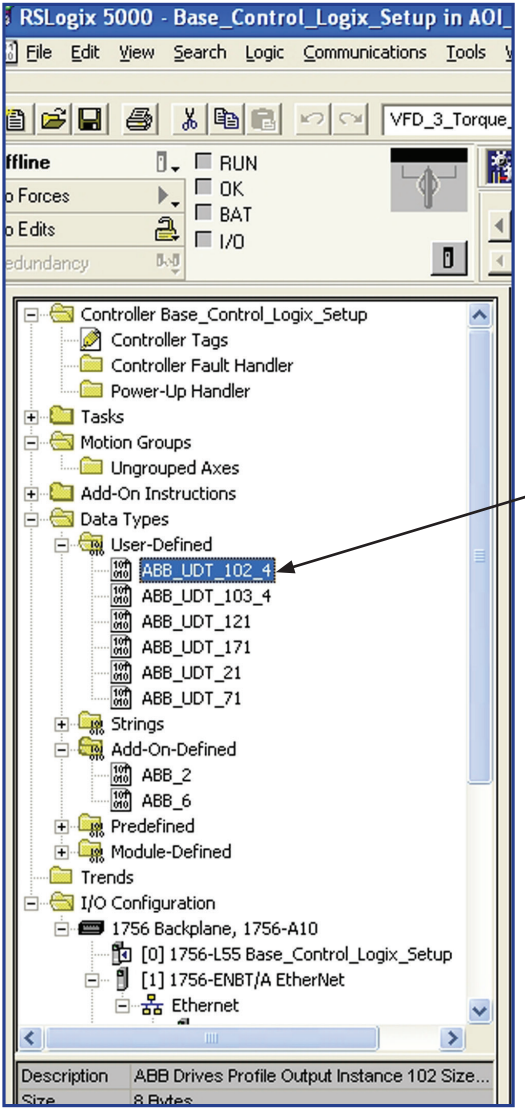
All of the ABB drives support both static and dynamic input and output assembly instances. An I/O assembly instance is static if all of the data input/output words are mapped to predefined drive registers (i.e. Control Word, Reference 1, Status Word and Actual 1). An I/O assembly instance is dynamic if all of the data words are not mapped to predefined drive registers. The user defined data types that support the dynamic I/O assembly instances can be easily edited to give the freely mapped data inputs and data outputs meaning full names (i.e. Custom_Out_1 = Acceleration Time 2).

The following I/O Assemblies are dynamic:
170/120, 171/121, 172/122, 173/123, 151/101, 152/102, 103/102.

The following I/O Assemblies are static:
70/20, 71/21, 72/22, 73/23, 51/1, 101/100, 52/2

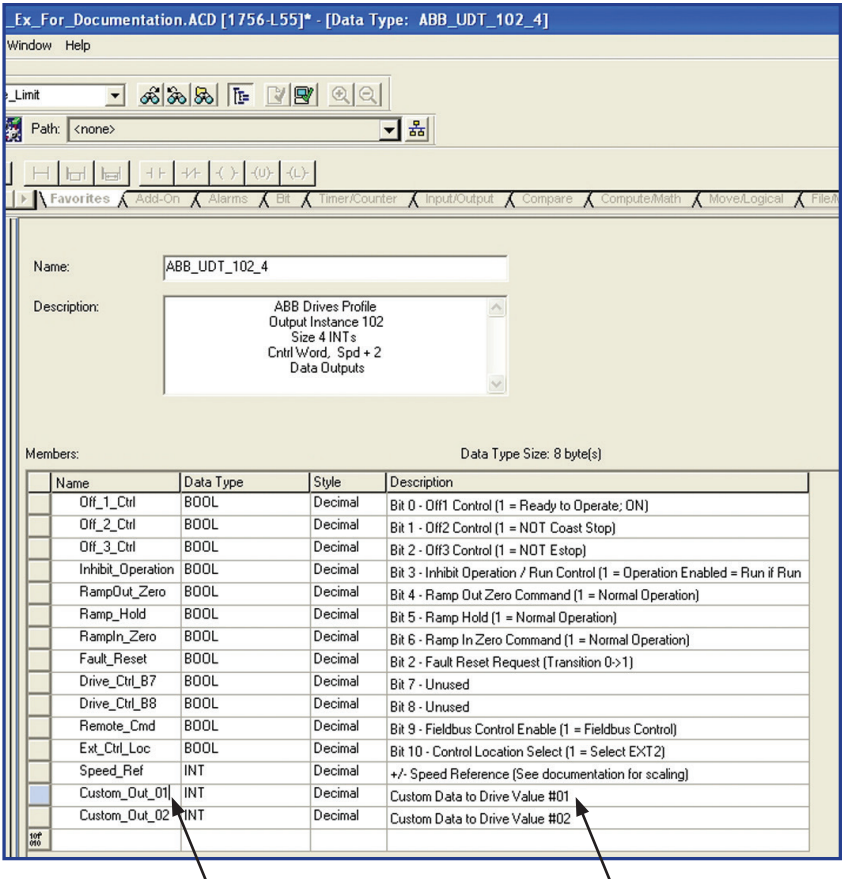
Assembly 102/103 is a User Specific Control Assembly. The content of the I/O can be configured from the drives parameters. This section will demonstrate how to modify the user defined data type ABB_UDT_102_4 and ABB_UDT_103_4 with application specific names. This section will assume that the add-on instruction and user defined data type have been imported and configured per section 5.

1. Open the User Defined Data Type.



Double click the ABB_UDT_102_4 User Defined Data Type from the RSLogix5000 I/O tree.

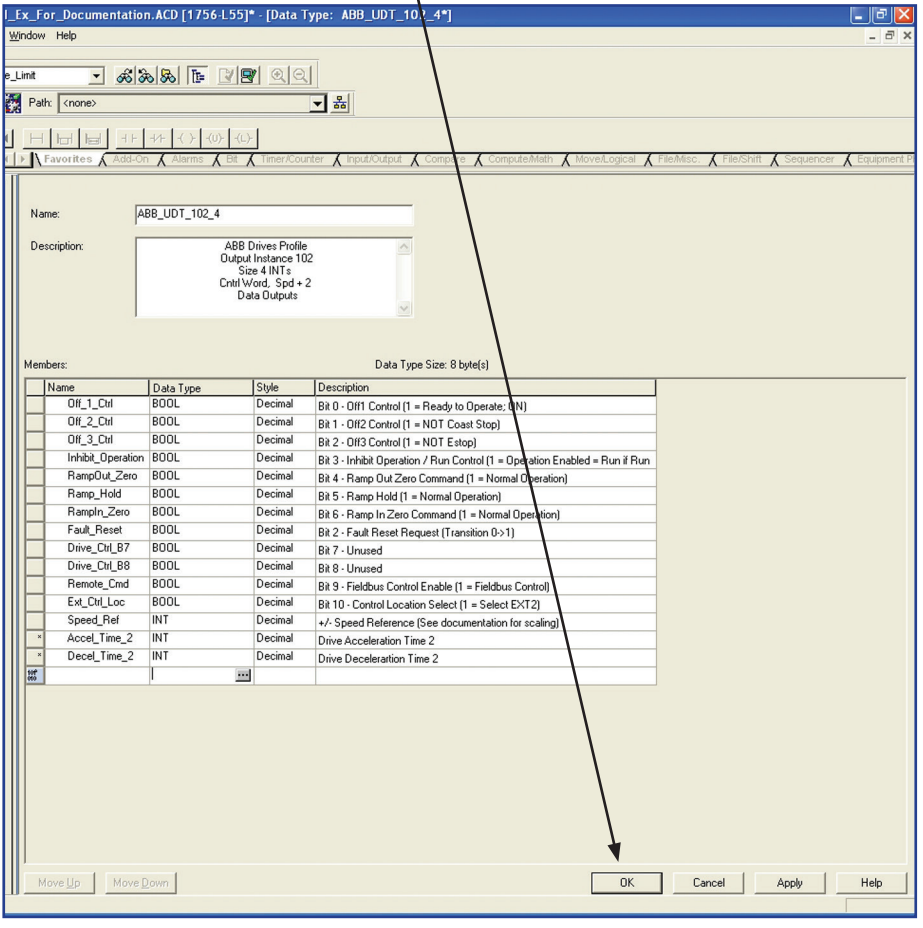
2. User Defined Data Type ABB_UDT_102_4 word 3 and word 4 are custom data outputs. Change the name of custom data out 1 and 2.



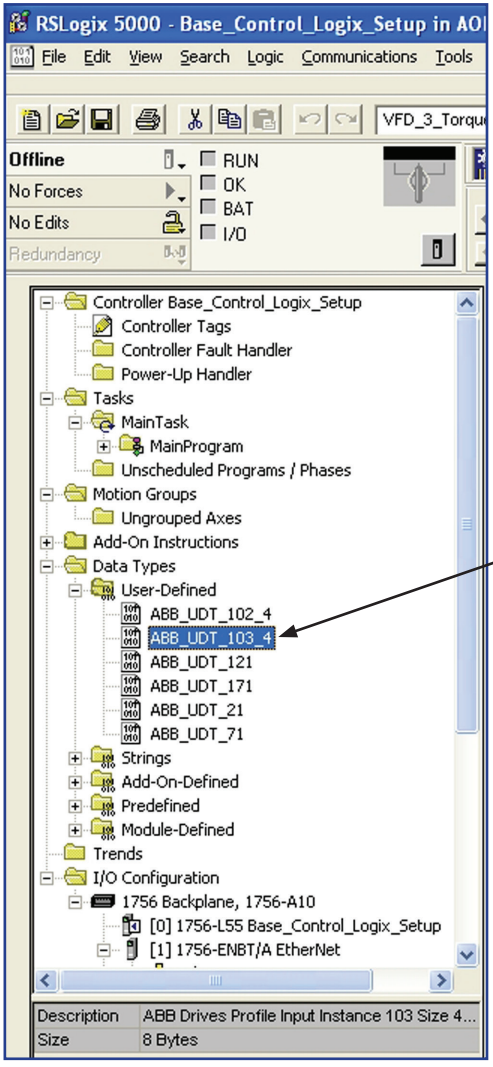
Select Custom_Out_01 cell and type Accel_Time_2. Then select the Custom_Out_02 cell and type Decel_Time_2.

The description column may also be changed to provide a more detailed description of the data output.

3. Save the changes by clicking OK.

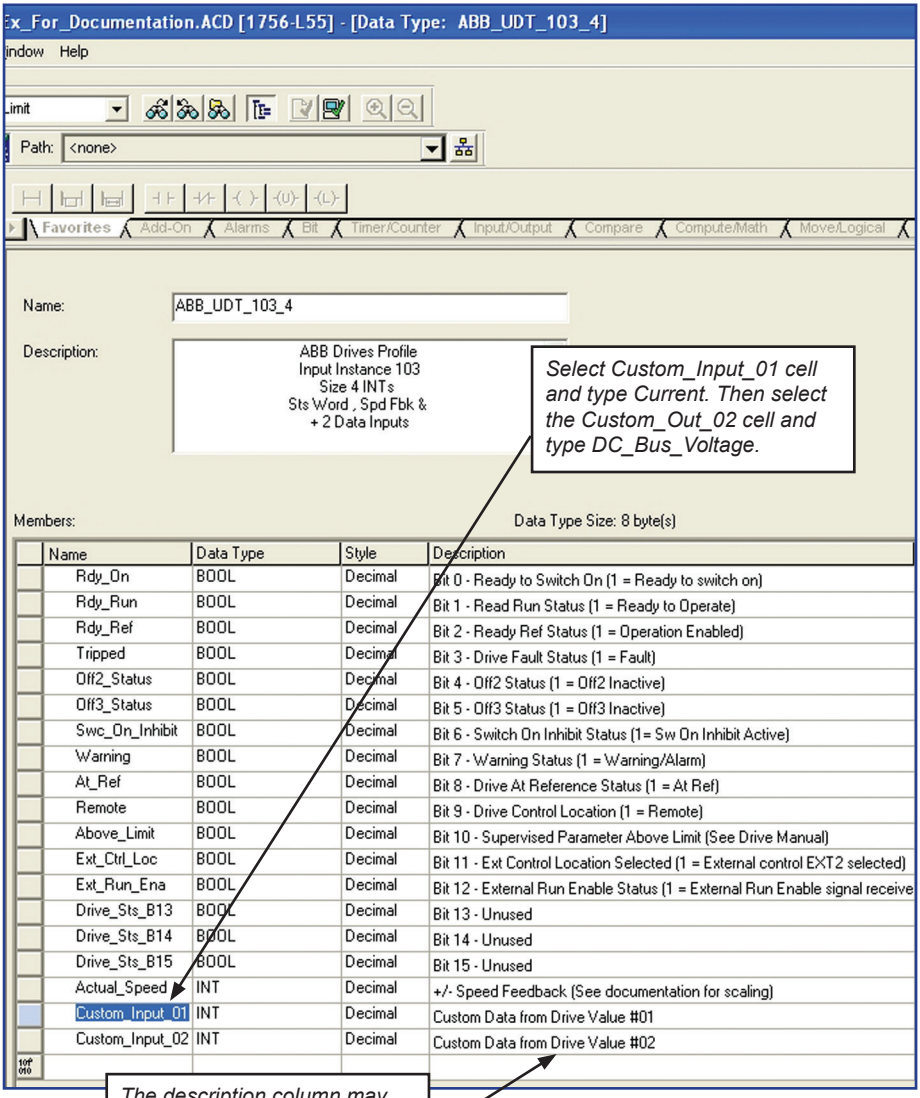


4. The input user defined data type ABB_UDT_103_4 can also be modified. Open the input user defined data type.



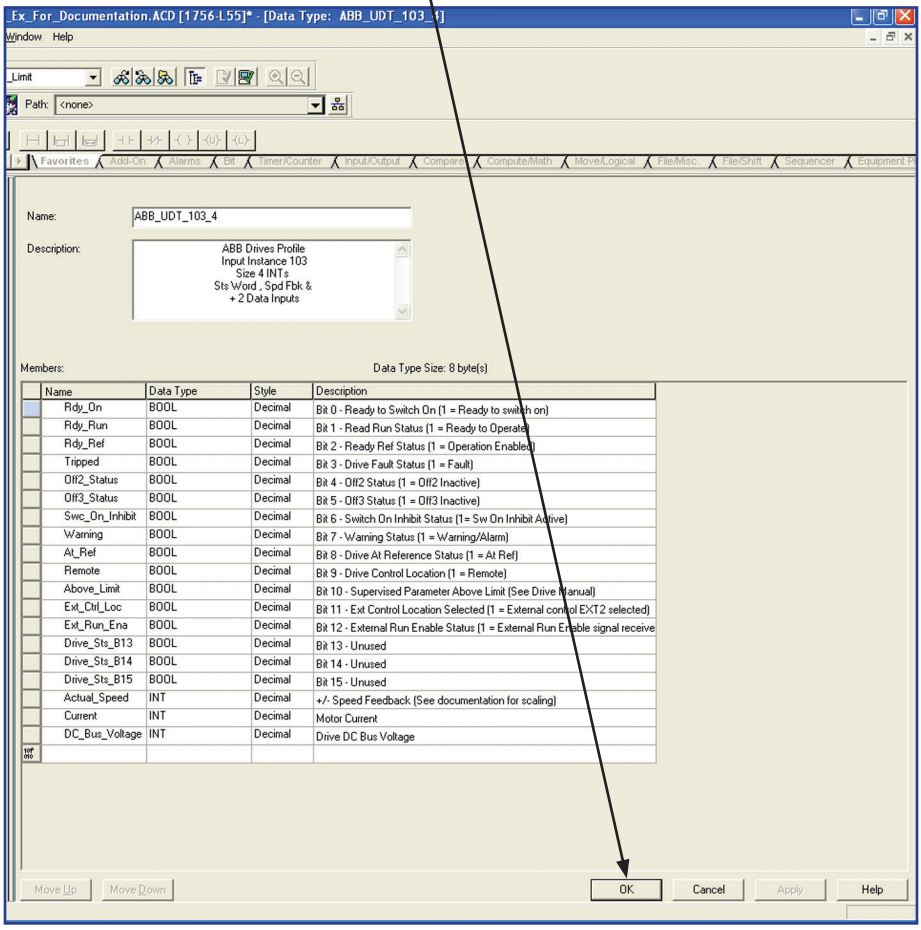
Double click the ABB_UDT_103_4 User Defined Data Type from the RSLogix5000 I/O tree.

5. User Defined Data Type ABB_UDT_102_4 word 3 and word 4 are custom data outputs. Change the name of custom data out 1 and 2.



Name	Data Type	Style	Description
Rdy_On	BOOL	Decimal	Bit 0 - Ready to Switch On (1 = Ready to switch on)
Rdy_Run	BOOL	Decimal	Bit 1 - Read Run Status (1 = Ready to Operate)
Rdy_Ref	BOOL	Decimal	Bit 2 - Ready Ref Status (1 = Operation Enabled)
Tripped	BOOL	Decimal	Bit 3 - Drive Fault Status (1 = Fault)
Off2_Status	BOOL	Decimal	Bit 4 - Off2 Status (1 = Off2 Inactive)
Off3_Status	BOOL	Decimal	Bit 5 - Off3 Status (1 = Off3 Inactive)
Swc_On_Inhibit	BOOL	Decimal	Bit 6 - Switch On Inhibit Status (1= Sw On Inhibit Active)
Warning	BOOL	Decimal	Bit 7 - Warning Status (1 = Warning/Alarm)
At_Ref	BOOL	Decimal	Bit 8 - Drive At Reference Status (1 = At Ref)
Remote	BOOL	Decimal	Bit 9 - Drive Control Location (1 = Remote)
Above_Limit	BOOL	Decimal	Bit 10 - Supervised Parameter Above Limit (See Drive Manual)
Ext_Ctrl_Loc	BOOL	Decimal	Bit 11 - Ext Control Location Selected (1 = External control EXT2 selected)
Ext_Run_Ena	BOOL	Decimal	Bit 12 - External Run Enable Status (1 = External Run Enable signal receive
Drive_Sts_B13	BOOL	Decimal	Bit 13 - Unused
Drive_Sts_B14	BOOL	Decimal	Bit 14 - Unused
Drive_Sts_B15	BOOL	Decimal	Bit 15 - Unused
Actual_Speed	INT	Decimal	+/- Speed Feedback (See documentation for scaling)
Custom_Input_01	INT	Decimal	Custom Data from Drive Value #01
Custom_Input_02	INT	Decimal	Custom Data from Drive Value #02

6. Save the changes by clicking OK.



Notes:

Section 6: Modifying User Defined Data Types Tag Names.

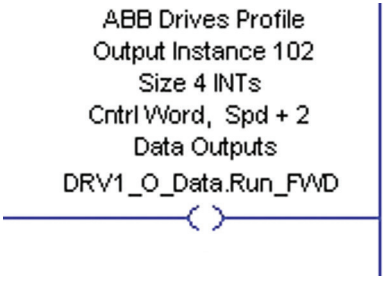
NOTE! ***Changing the following user defined data type member names is not allowed:***
- ***Control Word***
- ***Reference 1***
- ***Reference 2***

Doing so will create an incompatiblity with the associated Add-On Instruction.

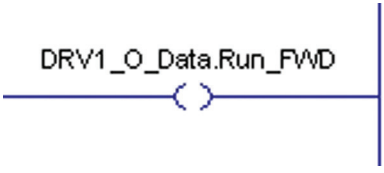
In addition to making changes to the names and descriptions of the custom input and outputs, the member name of the user defined data types can be changed as well. The description of the user defined data type will precede the individual member name.

The descriptions of the user defined data type are intended to provide the programmer with a thorough description of I/O assembly instance that the user defined data type supports. This description will appear in the PLC code. The programmer may decide to delete this description. This will reduce the size and complexity of the tag description.

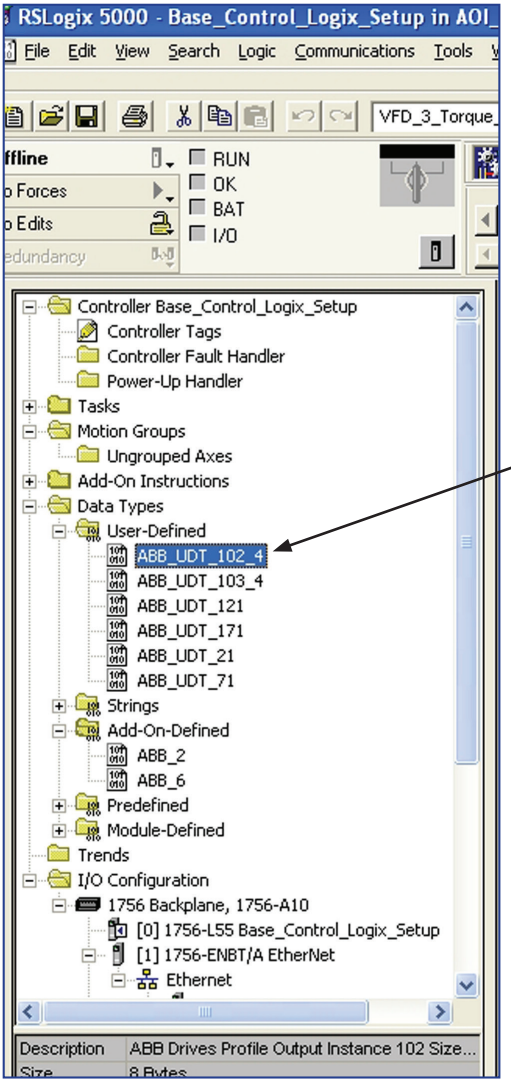
1. Here is an example of what the default tag description for run forward will look like when using output instance 102.



The description above includes the description for the user defined data type and then the user defined data type member name of bit 0 in the control word. If the description in the user defined data type is deleted then the tag name will only include the name of the user defined member.

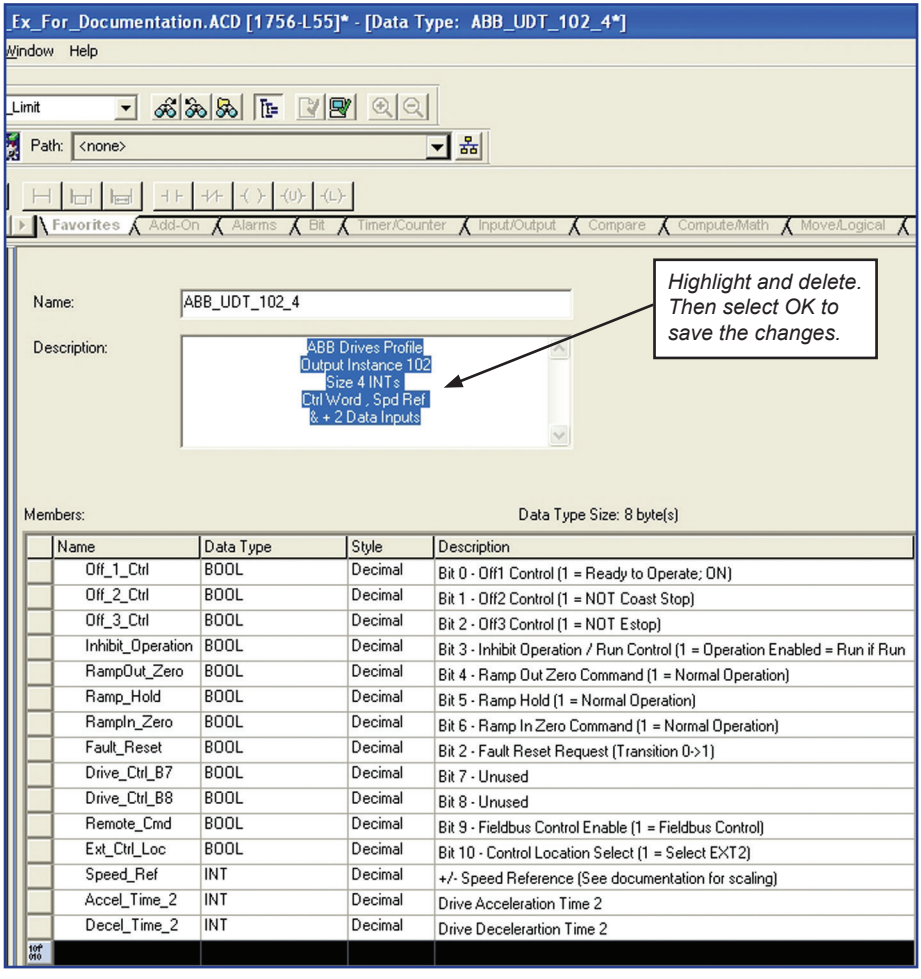


2. To delete the description of the output user defined data type. Open the input user defined data type ABB_UDT_102_4.



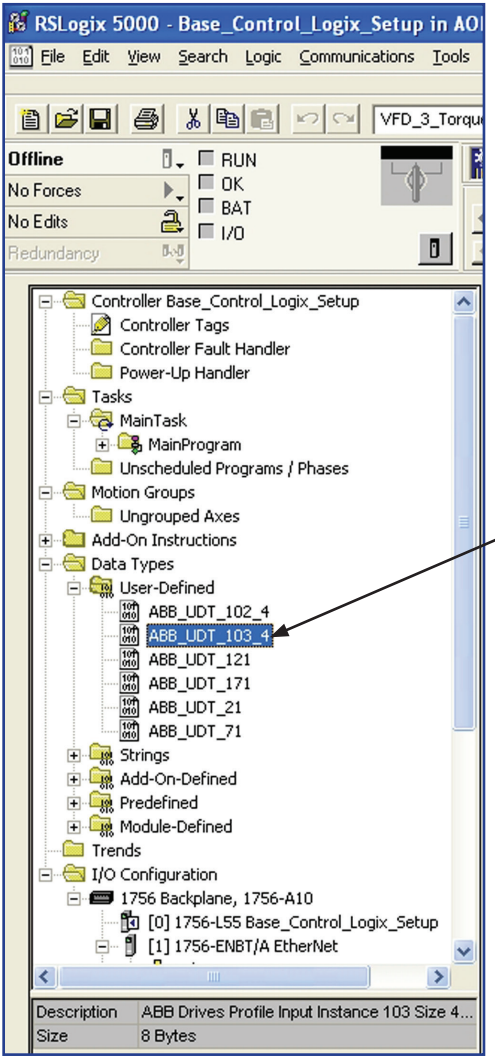
Double click the ABB_UDT_102_4 User Defined Data Type from the RSLogix5000 I/O tree.

3. Highlight the user defined data type.



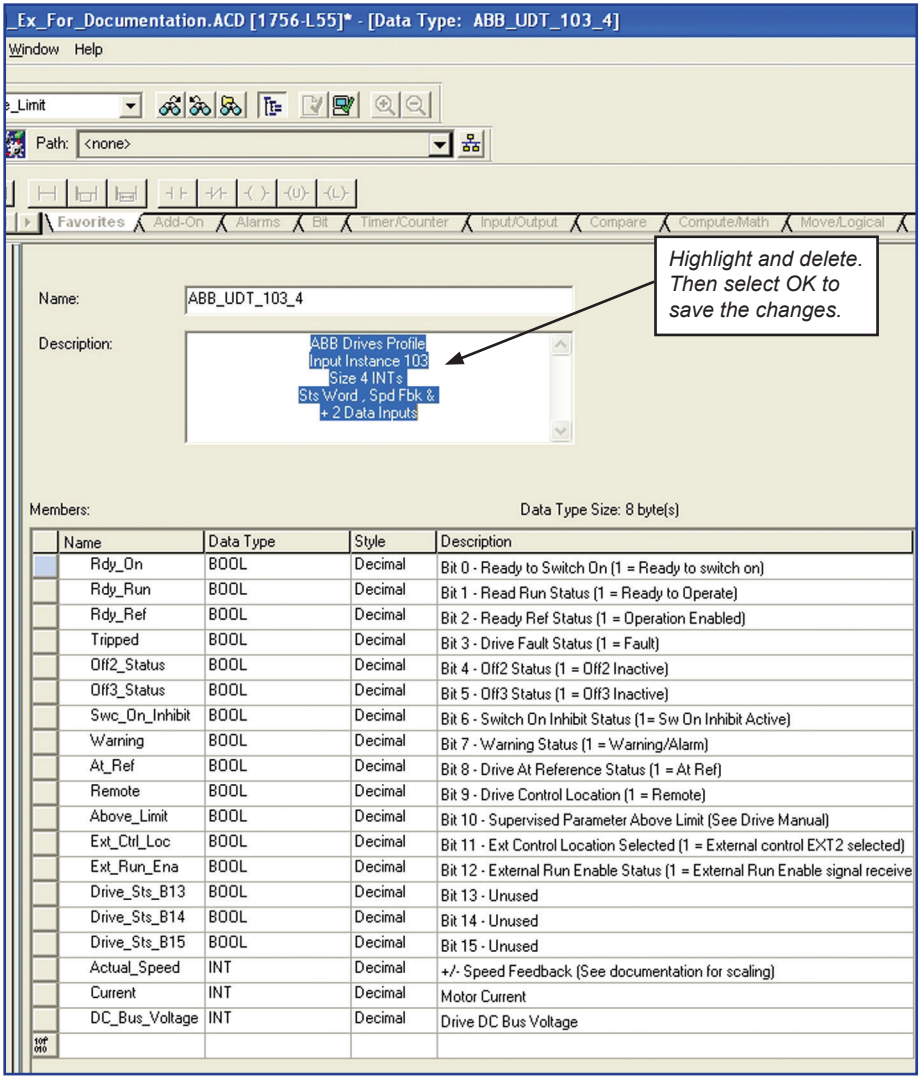
Name	Data Type	Style	Description
Off_1_Ctrl	BOOL	Decimal	Bit 0 - Off1 Control (1 = Ready to Operate; ON)
Off_2_Ctrl	BOOL	Decimal	Bit 1 - Off2 Control (1 = NOT Coast Stop)
Off_3_Ctrl	BOOL	Decimal	Bit 2 - Off3 Control (1 = NOT Estop)
Inhibit_Operation	BOOL	Decimal	Bit 3 - Inhibit Operation / Run Control (1 = Operation Enabled = Run if Run
RampOut_Zero	BOOL	Decimal	Bit 4 - Ramp Out Zero Command (1 = Normal Operation)
Ramp_Hold	BOOL	Decimal	Bit 5 - Ramp Hold (1 = Normal Operation)
RampIn_Zero	BOOL	Decimal	Bit 6 - Ramp In Zero Command (1 = Normal Operation)
Fault_Reset	BOOL	Decimal	Bit 2 - Fault Reset Request (Transition 0->1)
Drive_Ctrl_B7	BOOL	Decimal	Bit 7 - Unused
Drive_Ctrl_B8	BOOL	Decimal	Bit 8 - Unused
Remote_Cmd	BOOL	Decimal	Bit 9 - Fieldbus Control Enable (1 = Fieldbus Control)
Ext_Ctrl_Loc	BOOL	Decimal	Bit 10 - Control Location Select (1 = Select EXT2)
Speed_Ref	INT	Decimal	+/- Speed Reference (See documentation for scaling)
Accel_Time_2	INT	Decimal	Drive Acceleration Time 2
Decel_Time_2	INT	Decimal	Drive Decelerartion Time 2

4. To delete the description of the input user defined data type. Open the input user defined data type ABB_UDT_103_4.



Double click the ABB_UDT_103_4 User Defined Data Type from the RSLogix5000 I/O tree.

5. Highlight the user defined data type description.



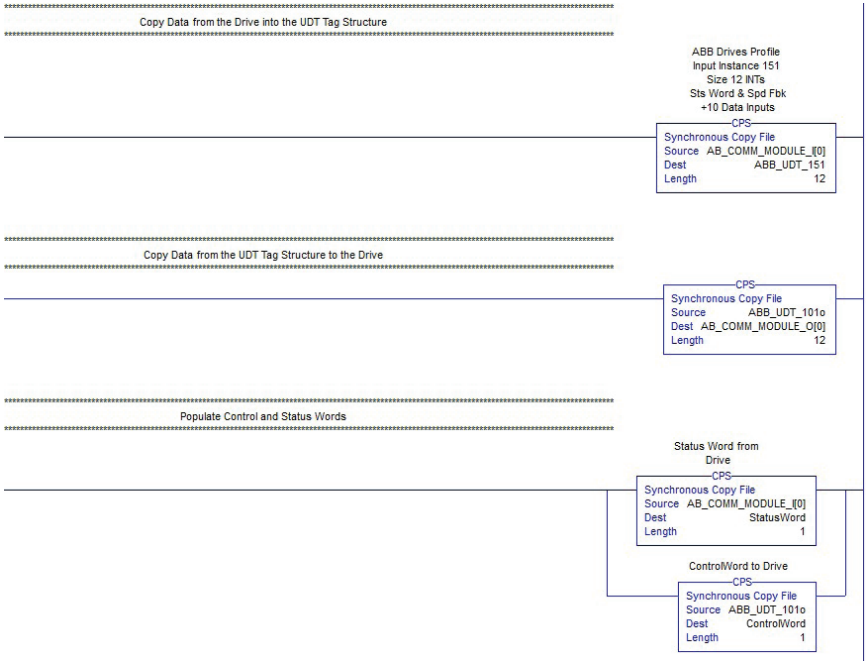
Name	Data Type	Style	Description
Rdy_On	BOOL	Decimal	Bit 0 - Ready to Switch On (1 = Ready to switch on)
Rdy_Run	BOOL	Decimal	Bit 1 - Read Run Status (1 = Ready to Operate)
Rdy_Ref	BOOL	Decimal	Bit 2 - Ready Ref Status (1 = Operation Enabled)
Tripped	BOOL	Decimal	Bit 3 - Drive Fault Status (1 = Fault)
Off2_Status	BOOL	Decimal	Bit 4 - Off2 Status (1 = Off2 Inactive)
Off3_Status	BOOL	Decimal	Bit 5 - Off3 Status (1 = Off3 Inactive)
Swc_On_Inhibit	BOOL	Decimal	Bit 6 - Switch On Inhibit Status (1= Sw On Inhibit Active)
Warning	BOOL	Decimal	Bit 7 - Warning Status (1 = Warning/Alarm)
At_Ref	BOOL	Decimal	Bit 8 - Drive At Reference Status (1 = At Ref)
Remote	BOOL	Decimal	Bit 9 - Drive Control Location (1 = Remote)
Above_Limit	BOOL	Decimal	Bit 10 - Supervised Parameter Above Limit (See Drive Manual)
Ext_Ctrl_Loc	BOOL	Decimal	Bit 11 - Ext Control Location Selected (1 = External control EXT2 selected)
Ext_Run_Ena	BOOL	Decimal	Bit 12 - External Run Enable Status (1 = External Run Enable signal receive
Drive_Sts_B13	BOOL	Decimal	Bit 13 - Unused
Drive_Sts_B14	BOOL	Decimal	Bit 14 - Unused
Drive_Sts_B15	BOOL	Decimal	Bit 15 - Unused
Actual_Speed	INT	Decimal	+/- Speed Feedback (See documentation for scaling)
Current	INT	Decimal	Motor Current
DC_Bus_Voltage	INT	Decimal	Drive DC Bus Voltage

[This page intentionally left blank]

Section 7: How the AOI works

The Add-on Instruction is simply doing a copy function from the User Defined Data Types imported to the Generic Module that is created in the project. The Add-on Instruction is also left unlocked so that PLC code can be added so that the AOI can be used for other functions.

Below is an image of the code inside the AOI. As you can see all that the AOI is doing is doing a copy of the data from the UDT's imported into the project to the Generic Module data created for the drive. If multiple drives are being used, only 1 set of AOI and UDT's need to be imported into the project assuming the same I/O assemblies are being used. The AOI will need to be called separately for each drive used along with a separate Generic Module for each.



[This page intentionally left blank]

[This page intentionally left blank]

LVD-PNTG07U-EN REVE Effective 08/31/2017 Subject to Change Without Notice

ABB Inc.
Low Voltage Drives
16250 W. Glendale Drive
New Berlin, WI 53151
Ph: 800-752-0696

www.abbnow.com/abbconnectivity