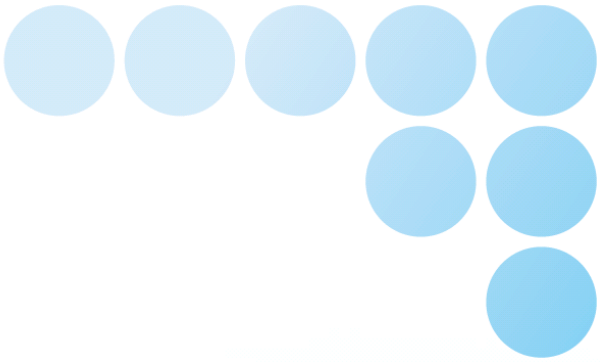
OMRON

Vision Sensor

FH Series

Vision System



Robot Connection Guide

ABB Corporation Edition

Z459-E1-01

NOTE

- All rights reserved.
- No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.
- No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions.
  Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- This software is based in part on the work of the Independent JPEG Group.
- Microsoft, Windows, Windows Vista, Excel, and Visual Basic are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- Intel, Core and Pentium are trademarks of Intel Corporation in the U.S. and/or other countries.
- EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- ODVA, CIP, CompoNet, DeviceNet, and EtherNet/IP are trademarks of ODVA.
- The SD, SDHC, microSD, and microSDHC logos are trademarks of SD-3C, LLC.

- QR Code is a registered trademark of DENSO WAVE INCORPORATED.
- MELSEC is a registered trademarks of Mitsubishi Electric Corporation.

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

Copyrights

Microsoft product screen shots used with permission from Microsoft.

CONTENTS

Introduction

Thank you for purchasing the FH Series.

This manual contains information that is necessary to use the FH Series.

Please read this manual and make sure you understand the functionality and performance of the FH Series before you attempt to use it in a control system.

Keep this manual in a safe place where it will be available for reference during operation.

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

● Exclusive Warranty

Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

● Limitations

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

● Buyer Remedy

Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANYWAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Safety Precautions

For details on Safety Precautions, refer to Safety Precautions in the *Vision System FH Series 3D Robot Vision Application Construction Guide (Cat. No. Z446)*.

Waring

For details on Waring, refer to Waring in the *Vision System FH Series 3D Robot Vision Application Construction Guide (Cat. No. Z446)*.

Precautions for Safe Use

For details on Precautions for Safe Use, refer to Precautions for Safe Use in the *Vision System FH Series 3D Robot Vision Application Construction Guide (Cat. No. Z446)*.

Precautions for Correct Use

For details on Precautions for Correct Use, refer to Precautions for Correct Use in the *Vision System FH Series 3D Robot Vision Application Construction Guide (Cat. No. Z446)*.

Regulations and Standards

For details on Regulations and Standards, refer to Regulations and Standards in the *Vision System FH Series 3D Robot Vision Application Construction Guide (Cat. No. Z446)*.

Related Manuals

<Application Construction Guide>

Name of Manual	Cat. No.	Model	Purpose	Contents
Vision Sensor FH Series 3D Robot Vision Application Construction Guide	Z446	FH-5050 FH-SMDA-GS050B	When User want to know about the FH series 3D robot vision system.	Describes the soft functions, setup, and operations to use FH series 3D robot vision system.

<Robot Manual OMRON>

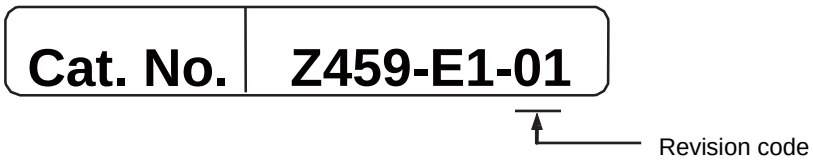
Name of Manual	Cat. No.	Model	Purpose	Contents
Robot Safety Guide	I590	-	When User want to know how to handle an industrial robot safely	Describes precautions for safe handling of the robot.

<Robot Manual ABB>

Name of Manual	Cat. No.	Model	Purpose	Contents
Operating manual Trouble shooting IRC5	3HAC020738 -001	Robot Controller IRC5	When User want to know the solution of robot is interrupted by malfunction.	Describes the reason of the interrupted in robot controller and the solution.
Product manual IRB120	3HAC035728 -001	Vertical multi-joint robot IRB120	When User want to know how to setup the robot.	Describes the installation, maintenance, repair of the robot controller.
Product manual IRC5	3HAC047136 -001	Robot Controller IRC5	When User want to know how to setup the robot.	Describes the installation, maintenance, repair of the robot controller.
Operating manual IRC5 with FlexPendant	3HAC050941 -001	Robot Controller IRC5	when user want to know how operate the IRC5 with FlexPendant	Describes the instructions for operation of IRC5 based robot systems using a FlexPendant

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Rev. Code	Rev. Date	Revision Contents
01	Jun. 2021	Original product

1. Overview

1.1. Overview

This manual describes procedures for connections and settings required for constructing robot vision applications by connecting your robot controller to the Vision Sensor FH (hereafter referred to as Vision Sensor).
Utilizing this manual and Robot Vision Application Construction Guide can reduce man-hours to connect the Vision Sensor to your robot controller, set the Vision Sensor, and create robot programs.

1.2. Instructions for Building a 3D Robot Vision Application

Please follow the flow below for constructing 3D robot vision applications

Procedure	Reference
Creating Data Set for Robot Vision	[3D Robot Vision Application Construction Guide] Chapter 6
↓	
System Settings for Vision Sensors	[3D Robot Vision Application Construction Guide] Chapter 7
↓	
Setting Communications for Robot controller	Refer to Chapter 3.1
↓	
Connecting Vision Sensor to Robot Controller	Refer to Chapter 3.2 Refer to Chapter 3.3
↓	
Robot Vision Settings for Vision Sensors	[3D Robot Vision Application Construction Guide] Chapter 8
↓	
Description of the sample programs	Refer to Chapter 6

1.3. Robot Programs Covered in this Manual

The two types of robot programs covered in this manual are output from the Robot Vision Dataset Output Tool. Each program is used for a different purpose.

Program	Program Name	Detail
Setup Program	fhsetup_main	This program allows the Vision Sensor to give operating instructions to the robot to configure the Vision Sensor for robot vision. This program consists of the following functions - Send the current robot position to the Vision Sensor. - Move to the indicated position on the Vision Sensor.
Sample Program	fhsample_main	This program is a sample of the basic program flow for a pick application. In this program, the robot gives control instructions to the Vision Sensor. The program consists of the following functions - Connecting to the Vision Sensor - Scene switching of the Vision Sensor - Moving to the measurement position - Registering the current robot position to the Vision Sensor - Execute measurement instructions to the Vision Sensor - Receives the position of the workpiece to be recognized - Move to approach position - Move to the target work location (grasping position) Based on this program, a pick-and-place application is built by adding the robot movement to operate the end-effector (hand) and to place the workpiece.

2. System Configuration

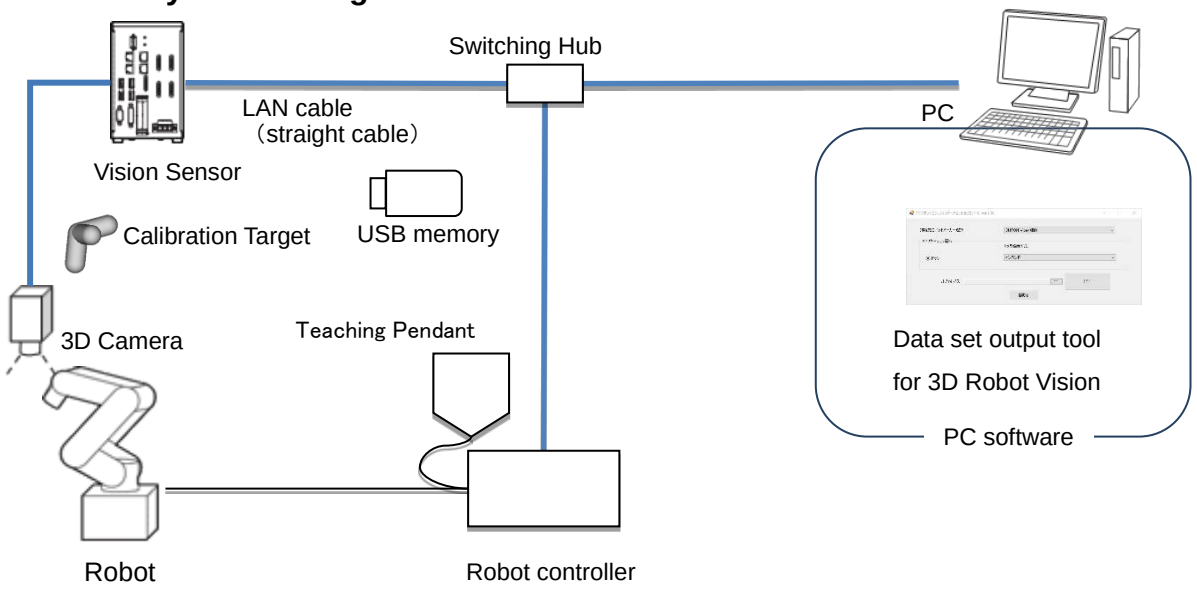
This chapter describes the system configuration and target devices to construct robot vision applications.

2.1. Cautions for Robot Equipment

Enable the PC interface option in Robot controller.

2.2. When using Vision Sensor FH Series 3D Vision Sensor

2.2.1. System Configuration



2.2.2. Target Devices

Device name	Manufacturer	Name	Model	Remarks
Vision Sensor	OMRON	Vision Sensor FH Series	FH-5050	Ver. 6.40 or later Controllers other than FH-5050 are not supported.
3D Camera	OMRON	3D Vision Sensor	FH-SMDA-GS050B	-
Camera Cable	OMRON	Ethernet cable super bending resistance	FHV-VNBX□M FHV-VNLBX□M	-
Camera I/O cable	OMRON	I/O cable super bending resistance	FH-VSDX-BX□M FH-VSDX-LBX□M	-
Calibration target	OMRON	Handeye Calibration Target	FH-XCAL-R	-
	OMRON	Camera Calibration Target	FH-XCAL-S	-

3D Software	OMRON	3D Robot Vision Software Installer	FH-UM3D1	-
Robot controller	ABB	Robot controller	IRC5	Necessary to enable the PC interface option. Please contact the ABB for the detail.
Robot	ABB	Vertical multi-joint robot	IRB120	-
Teaching pendant	ABB	Teaching pendant (FlexPendant)	-	-
PC software	OMRON	Data set output tool for 3D robot vision	-	Ver.1.10 or later Please contact us for how to obtain it.
Switching Hub	OMRON	Switching Hub for industrial	W4S1-□□□	Recommended product
USB memory	OMRON	USB memory	FZ-MEM8G	Recommended product

Precautions for Correct Use

Do not use any device except mentioned above for each device of the system configuration.

Additional Information

This manual does not provide operations, installation, and wiring methods for each device.
For details, refer to manuals noted in Related Manuals.

3. Connecting Vision Sensor to Robot Controller

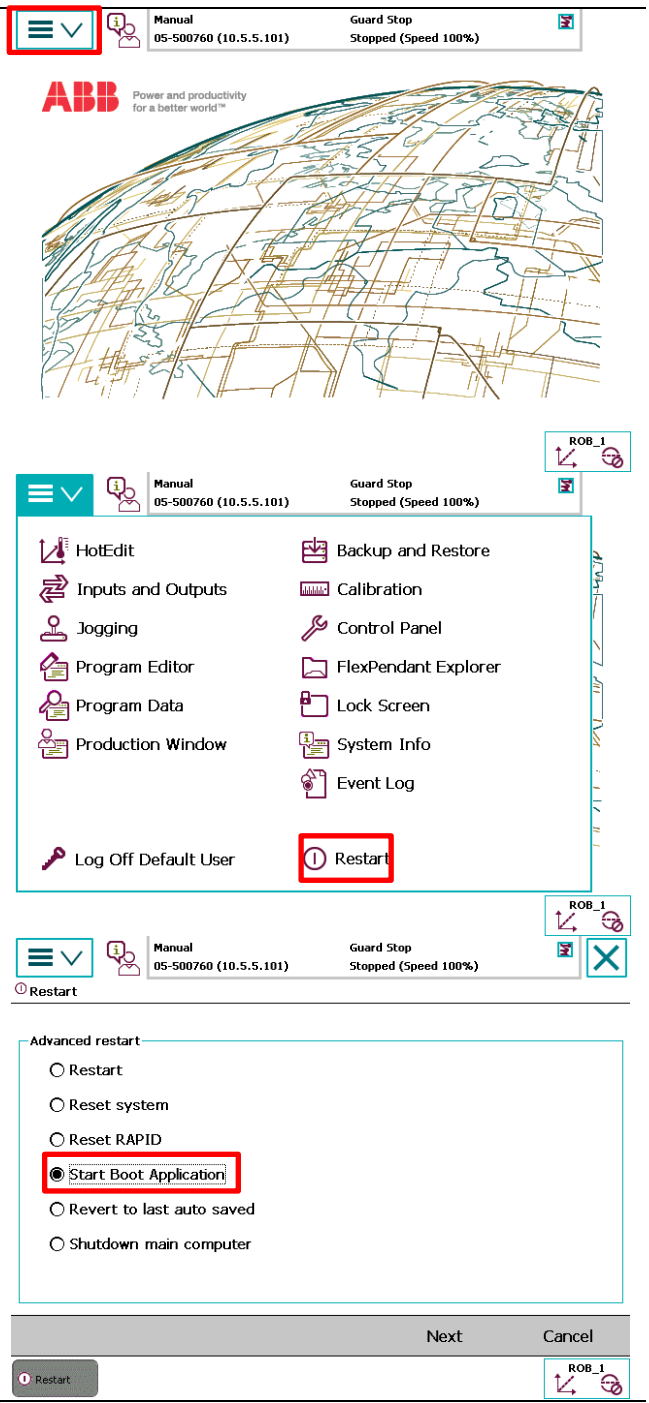
This chapter describes procedures to connect the Vision Sensor to the robot controller.
Please follow the flow below for the settings.
The IP address of each device is described below.

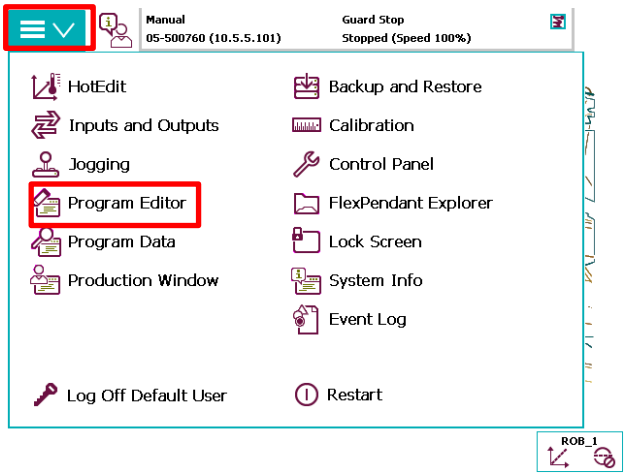
Vision Sensor (Ethernet1) : 10.5.5.100
Robot controller : 10.5.5.101

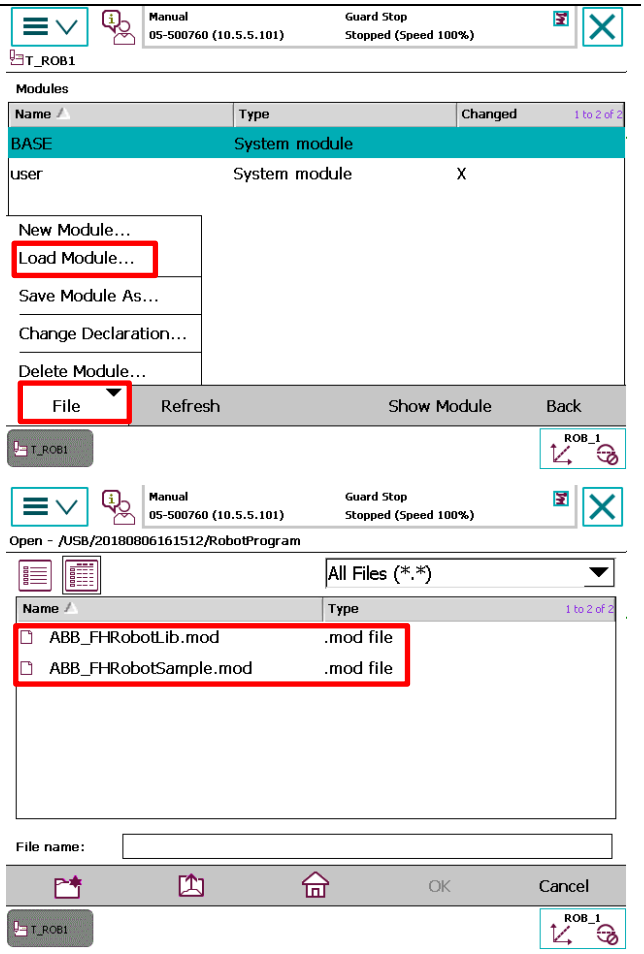
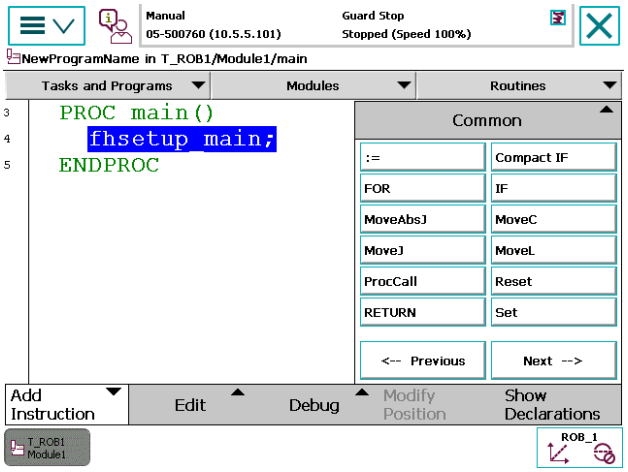
3.1	Setting communications for the robot controller	Modify the default IP address of the robot controller with the teaching pendant to fit the communication settings in the Vision Sensor.
▼		
3.2	Verifying Ethernet Communication	Check the connection status of Ethernet using PING command.
▼		
3.3	Verify Commands Sent/Received	Run a robot program for startup to establish the TCP/IP connection between the Vision Sensor and the robot controller. Check the communication status by sending and receiving commands.

3.1. Setting Communications for Robot Controller

Please follow the procedures below to set the communications for the robot controller.

1	Tap "ABB Menu" in the upper-left corner of the teaching pendant screen. Tap [Restart], then tap the [Advanced...]. Select "Start Boot Application" to restart the robot controller.	
2	After the robot controller restarts, tap [Settings] on the "ABB Robotics Boot Application" screen. On the displayed "Network Connection" screen, select	

	[Use the following IP settings:], and then set the [IP address], [Subnet mask], and [Default gateway]. Set an IP address not to overlap with other devices.	
3	Tap [Select System] on the "ABB Robotics Boot Application" screen. Select the system to start on the "Select System" screen.	
4	Tap [Restart Controller] on the "ABB Robotics Boot Application" screen to restart the robot controller.	
5	Copy a robot program outputted from the data set output tool to the USB memory stick. Insert the USB memory stick storing the copied robot program to a USB port on the teaching pendant or the robot controller. From the "ABB Menu" of the teaching pendant screen, tap [Program Editor].	

	Tap [File] and select [Load Module]. Load the robot program that copied to the USB memory stick.	
6	Add the following program into the main routine. “fhsetup_main;” Initial setting values of communication are as follows. IP address: 10.5.5.100 Port number: 9876 Change the “fhsetup_main” routine based on the actual environment.	

Additional Information

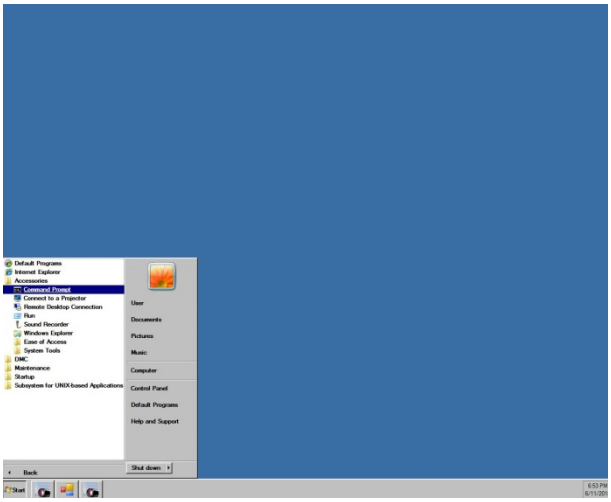
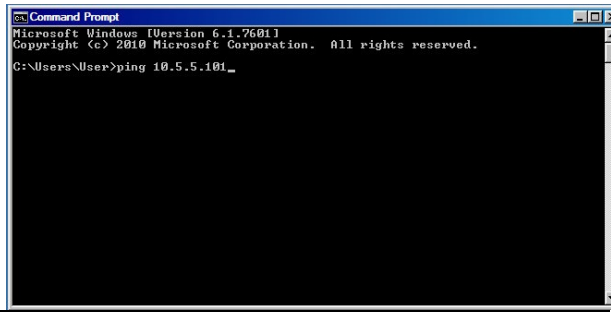
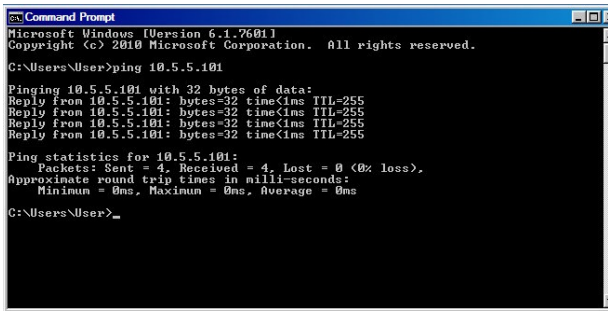
This manual does not provide operation, installation, and wiring methods for each device.

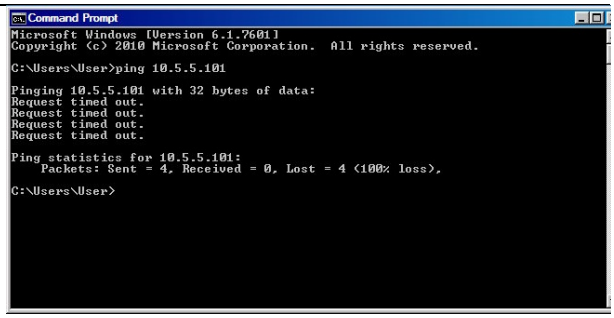
For details, refer to manuals noted in Related Manuals.

3.2. Connecting and Checking Vision Sensor and Robot Controller

Follow the procedures below to connect the Vision Sensor and the robot controller and to check the connection status.

3.2.1. Verifying Ethernet Communication (FH Series Vision Sensor)

1	(Operation of the Vision Sensor) Move the mouse cursor to lower left of the window to display [Start]. Select [Start] - [All Programs] - [Accessories] - [Command Prompt] to launch [Command Prompt].	
2	(Operation of the Vision Sensor) Execute PING command to the IP address of the robot controller.	
3	(Operation of the Vision Sensor) When 32-byte data could be successfully sent/received four times as shown in the figure on the right, that means that the communications have been established and the wiring and settings of Ethernet is correctly done. When 32-byte data cannot be sent/received four times and	

	PING command timed out, check whether or not the robot controller is turned on, the wiring was correctly done, or communication settings are correct.	

3.3. Verify Commands Sent/Received

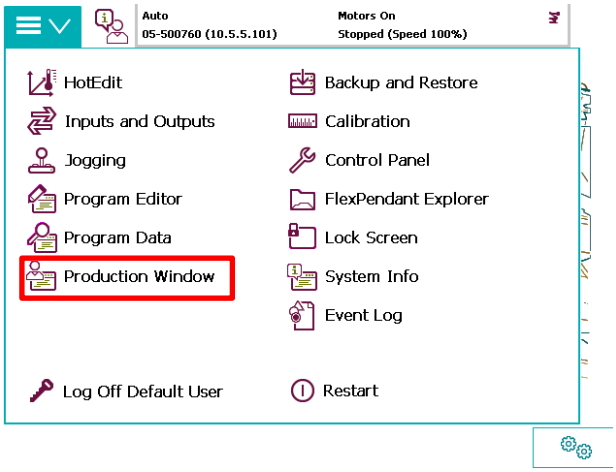
Execute the setup program on the robot controller and follow the steps below to confirm that commands can be sent and received from the Vision Sensor.

1

(Operation of the Teaching pendant)
Set the robot controller's "Mode switch" to "Auto" mode. And press "Motor on" on the robot controller.

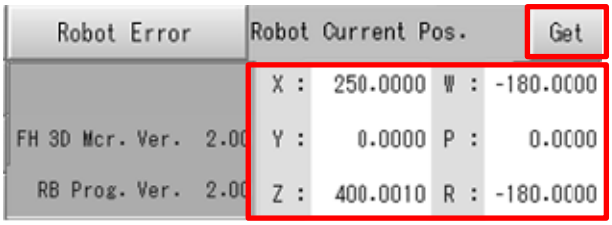
Tap "ABB Menu" in the upper-left corner of the teach pendant screen and tap [Production Window].

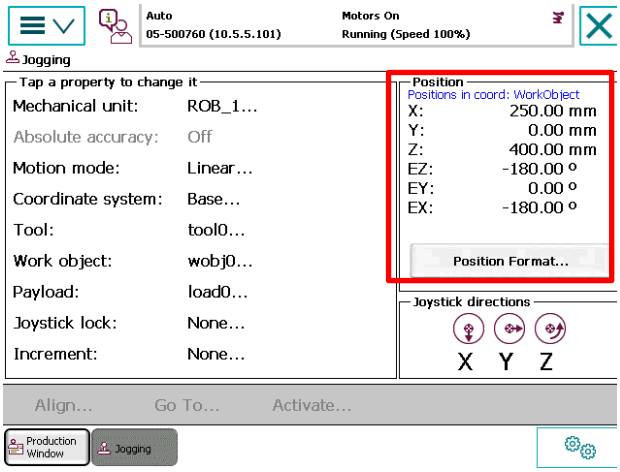
Tap [PP to Main], and press [Start] (▶) on the teaching pendant.



2

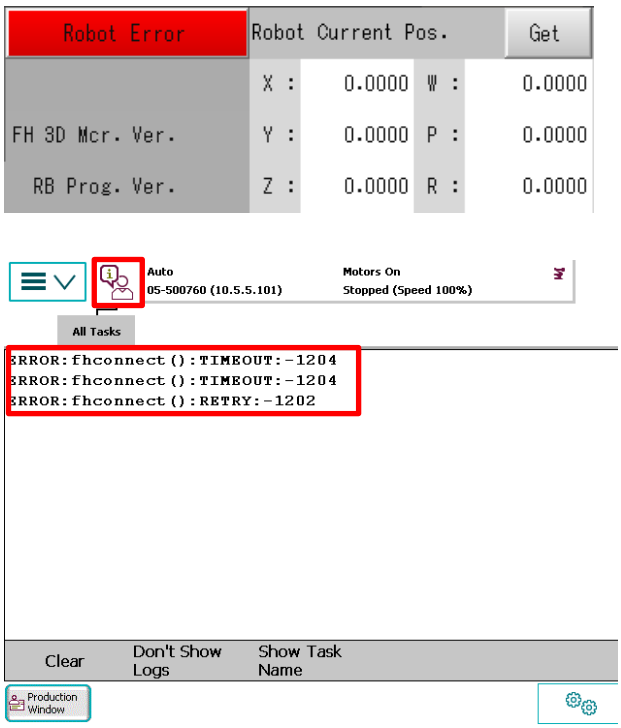
(Operation of the Teaching pendant and the Vision Sensor)
As shown on the right, when [Get] is clicked on the Main Window of the Vision Sensor and the current position of the robot on the teaching pendant is displayed at the same position on the Main Window of the Vision Sensor, commands have been successfully sent and received between the devices.





* The corresponding displays

Vision Sensor	Teaching pendant
W	EX
P	EY
R	EZ

3	(Operation of the Teaching pendant and the Vision Sensor) As shown on the right, when “Robot Error” is show on the Main Window of the Vision Sensor, connection was failed. Check the error message is displayed on the operator window, Check the wiring and others.	
4	(Operation of the Teaching pendant) The connection between the Vision Sensor and the robot controller was completed successfully. Press [Stop] (■) on the teaching pendant.	

Additional Information

This manual does not provide operation, installation, and wiring methods for each device.
For details, refer to manuals noted in Related Manuals.

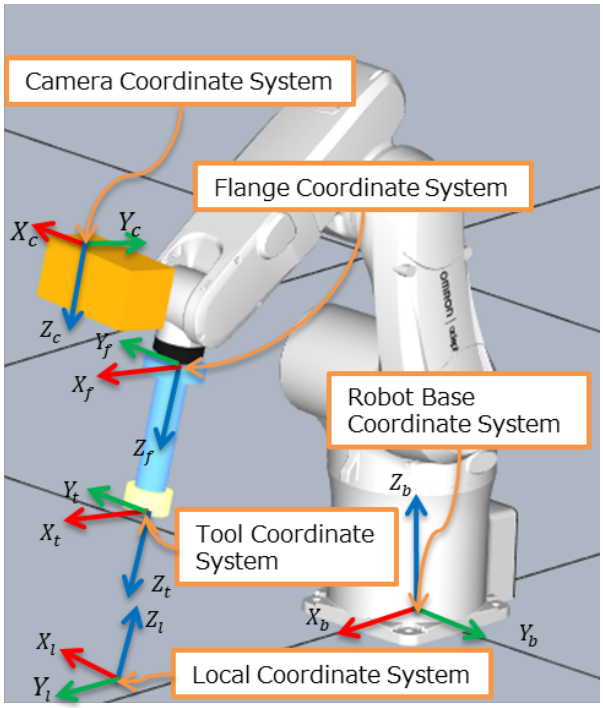
4. Coordinate System

This chapter describes the coordinate system handled by the robot vision application.

4.1. Name of Coordinate System

The robot coordinate system of the Vision Sensor uses the name shown in the table below.

Coordinate System	Meaning
Robot Base Coordinate System	Coordinate system with the robot base as the origin
Local Coordinate System	User-defined coordinate system
Flange Coordinate system	Coordinate system defined on the flange surface of the robot
Tool Coordinate System	The coordinate system is defined in the tool center point by offsetting the origin of the flange coordinates system.
Camera Coordinate System	With the optical center of the camera as the starting point, the X and Y axes are the horizontal and vertical directions of the image, and the Z axis is the optical axis of the camera.



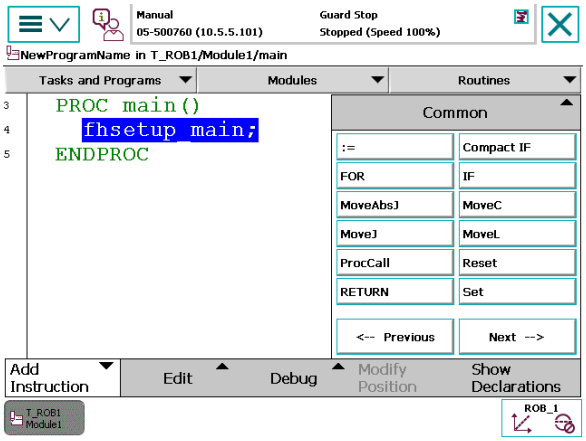
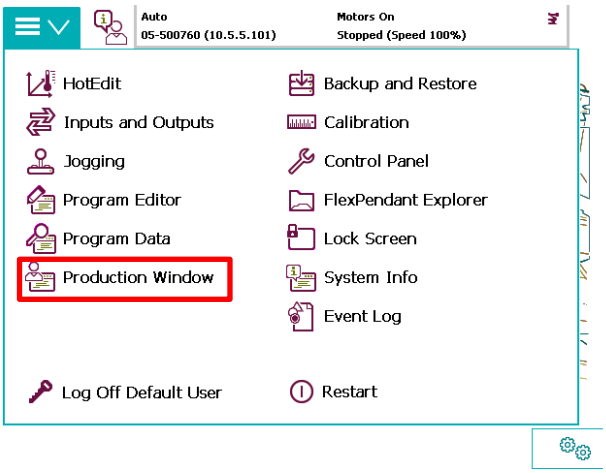
The orientation of the coordinate axes of each coordinate system depends on the robot. Please refer to the instruction manual for each robot.

There are the following differences between the names of the coordinate system in the Vision Sensor and the coordinate system in ABB Ltd.

Vision Sensor	ABB Ltd.
Local Coordinate System	Work Objects Coordinate System
Tool Coordinate System	Tool Coordinate System

5. How to Start the Setup Program

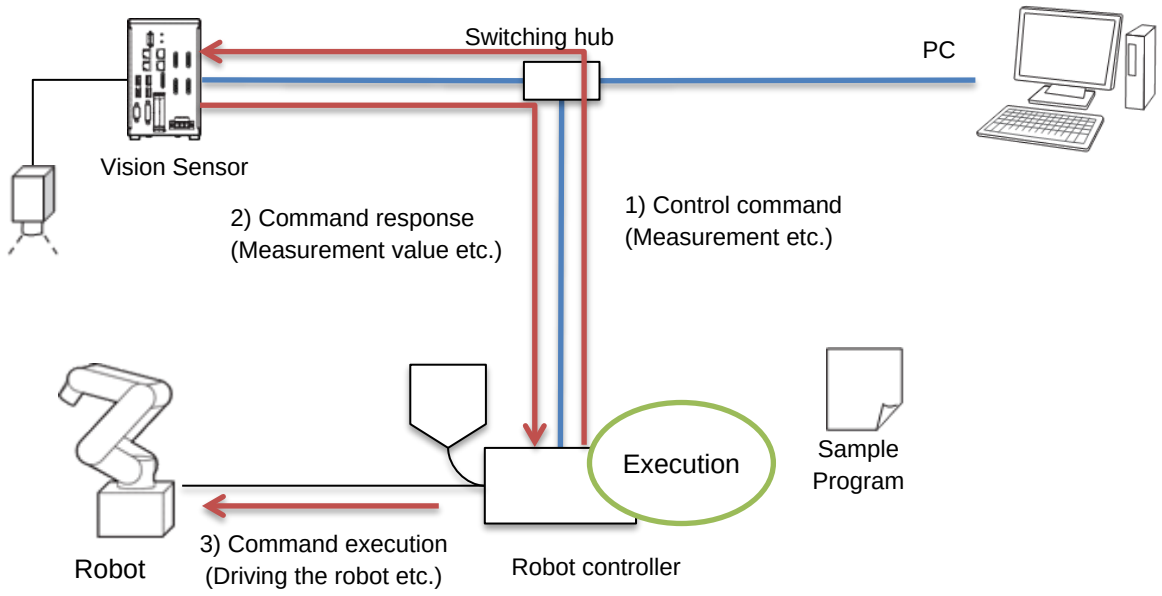
This chapter describes how to start the setup program. To set the robot vision of the Vision Sensor, the setup program must be running on the robot side. Establish the connection between the Vision Sensor and the robot controller by [3. Connecting Vision Sensor to Robot Controller]

1	Add the following program into the main routine. “fhsetup_main;” Initial setting values of communication are as follows. IP address: 10.5.5.100 Port number: 9876 Change the “fhsetup_main” routine based on the actual environment.	
2	(Operation of the Teaching pendant) Set the robot controller's "Mode switch" to "Auto" mode. And press "Motor on" on the robot controller. Tap "ABB Menu" in the upper-left corner of the teach pendant screen and tap [Production Window]. Tap [PP to Main], and press [Start] (▶) on the teaching pendant.	
3	(Operation of the Teaching pendant) Check the state of Program is “Running”.	

6. Description of the Sample Programs

This chapter describes design examples of robot programs to construct applications using a sample program.

You can understand how to implement a robot program to control the Vision Sensor as shown in the following figure.



The sample program is implemented with the following procedures. When building an actual application, design, implement and test the robot program, utilizing the functions described in Chapter 7.

6.1	Connecting the Vision Sensor to the robot controller
▼	
6.2	Switching scenes on the Vision Sensor
▼	
6.3	Moving the Robot to the Image Position
▼	
6.4	Register the Current Robot Position in the Vision Sensor
▼	
6.5	Executing Measurements on the Vision Sensor
▼	
6.6	Getting the Measurement Results
▼	
6.7	Moving the Robot to the Robot Approach Position at Measurement
▼	
6.8	Moving the Robot to the Robot Command position at Measurement
▼	
6.9	Disconnecting the Vision Sensor from the Robot Controller

Precautions for Correct Use

The implementation procedures for robot programs noted in this chapter are a reference. You should design, implement, and test actually operating robot programs based on your specific environment and applications.

In the Main Window or “Layout setup” of the Vision Sensor, check that the “Output” of the current layout is ON. If the setting were OFF, the Vision Sensor will not output measurement values.

6.1. Connecting Vision Sensor to Robot Controller

For connecting the Vision Sensor to the Robot Controller, follow the procedures below.

1

Declare internal variables.

*Omitted (Refer to the source code)

2

Execute the initialization for external variables.

```
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! (2) Initialize global variables
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
err_no := success;
!Set Current Coord No
cur_local_coord_no := 0;
cur_tool_coord_no := 0;
```

3

Set the IP address and the port number of the Vision Sensor to the var iables. (When changing those from the default.)

```
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! 2. Example: Network connection sequence
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! (1)Set the network configuration
!    You have to configure the following communication settings.
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
ip_address := "10.5.5.100";      IP address
port_no := 9876;                 Port number
retries_connect := 2;
timeout_connect := 4;
retries_recv := 2;
timeout_recv := 4;
```

4

Set the variables as arguments for the connection function (fhconnect) to the Vision Sensor and execute it.

```
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! (2)Connect to the FH server
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
WHILE bconnected = FALSE DO
   fhconnect fh_socket,ip_address, port_no, retries_connect, timeout_connect, err_no;

   !Error check                  Connection function with the Vision Sensor (FH Server)
   IF err_no <> success THEN
      GOTO exit_program;
   ENDIF
ENDWHILE
```

6.2. Switching Scenes on the Vision Sensor

For a processing to switch scenes on the Vision Sensor, follow the procedures below.

1 Set a scene number for the switching destination to a variable.

```
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! (1)Change the scene of the FH
!     You have to select a scene No. for your application.
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
scene_no := 0;
```

Scene number

2 Set the variables as arguments to the scene switching command execution sample function (fhsample_chgscn) to the Vision Sensor and execute it.

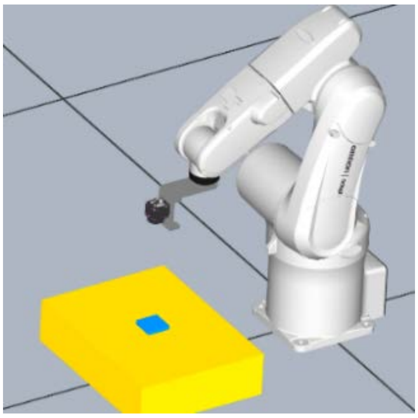
```
fhsample_chgscn fh_socket, retries_recv, timeout_recv, scene_no, err_no;
!Error check
IF err_no <> success THEN
    GOTO exit_program;
ENDIF
```

Scene switching command execution sample function

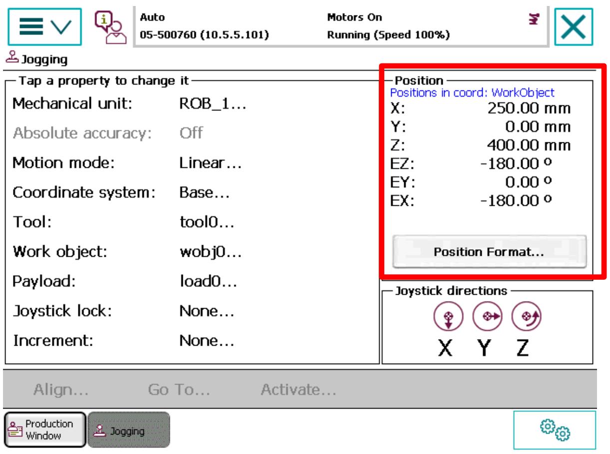
6.3. Moving Robot to Robot Image Position

For a processing to move the robot to the robot image position, follow the procedures below.

1 Move the robot to the imaging position.



Check the current robot position.



2

Set the referenced imaging position to the variables.

```
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! (2)Move the robot to the imaging position
!     You have to configure the following robot position for your application.
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
std_img_pos_x := 250;
std_img_pos_y := 0;
std_img_pos_z := 400;
std_img_pos_w := -180;
std_img_pos_p := 0;
std_img_pos_r := -180;
```

Set the robot imaging position to the variables.

3

Set the variables as arguments for the robot motion sample function (fhsample_move) and execute it.

```
!!!!!!!!!! CAUTION !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The following function drives a robot motion immediately.
! Confirm the settings before execution.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
fhsample_move std_img_pos_x, std_img_pos_y, std_img_pos_z, std_img_pos_w,
std_img_pos_p, std_img_pos_r, err_no;

!Error check
IF err_no <> success THEN
    GOTO exit_program;
ENDIF
```

Robot motion sample function

⚠WARNING

- These operations drive the robot.
- Operate the robot in the state whereby pressing the [Emergency stop] button can stop its motion anytime.

6.4. Register the Current Robot Position in the Vision Sensor

To register the current robot position to the Vision Sensor, use "fhsample_regpos".

1

Before executing measurements, execute the current position registration command execution function (fhsmaple_regpos) to the Vision Sensor.

```
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! (3)Register the current position of the robot to the FH before measurement
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
fhsample_regpos fh_socket, retries_recv, timeout_recv, err_no;

!Error check
IF err_no <> success THEN
    GOTO exit_program;
ENDIF
```

Current position registration command execution sample function

6.5. Executing Measurements on Vision Sensor

Send the measurement command to the Vision Sensor and receives a response to that command.

1

To send the measurement command to the Vision Sensor, set the command name to "MEASURE" and execute "fhrunsendcmd".

```
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! (4)Execute measurement and get the measurement results of the FH
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
!Send Measure Command
cmd_name := "MEASURE";
cmd_arg{1} := "";
cmd_arg{2} := "";
cmd_arg{3} := "";
cmd_arg{4} := "";
cmd_arg{5} := "";
cmd_arg_num := 0;
fhrunsendcmd fh_socket, cmd_name, cmd_arg, cmd_arg_num, err_no;

!Error check
IF err_no <> success THEN
    GOTO exit_program;
ENDIF
```

nonprocedural command transmission function

2

To receive the response to the measurement command from the Vision Sensor, execute "fhrunrecvres".

```
!Send Measure Command
fhrunrecvres fh_socket, retries_recv, timeout_recv, cmd_res, err_no;
IF err_no <> success THEN
    GOTO exit_program;
ENDIF
!Command Response Check
IF cmd_res <> 1 THEN
    GOTO exit_program;
ENDIF
```

command response receiving function

If the response is not OK, exit the program.

6.6. Getting the Measurement Results

The Vision Sensor measurements are received using "fhrunrecvval". In this sample program, it is assumed that the Vision Sensor measurements are sent in the order "TJG X Y Z W P R".

1

In order to receive the measurement results from the Vision Sensor, we execute "fhrunrecvval" and check the received overall judgment value.

```
! Get the Measurement Result
fhrunrecvval fh_socket, retries_recv, timeout_recv, outputdata, err_no;
IF err_no <> success THEN
    GOTO exit_program;
ENDIF
! Total Judge Check
IF outputdata{1} <> 1 THEN
    GOTO exit_program;
ENDIF
```

numerical sequence receiving function

If the overall judgment is not OK, exit the program

2

The measurement results X - R are stored in the variables.

```
!Measurement results
res_cmd_pos_x := outputdata{2};
res_cmd_pos_y := outputdata{3};
res_cmd_pos_z := outputdata{4};
res_cmd_pos_w := outputdata{5};
res_cmd_pos_p := outputdata{6};
res_cmd_pos_r := outputdata{7};
```

The measurement results X-R

6.7. Moving Robot to Robot Approach Position at Measurement

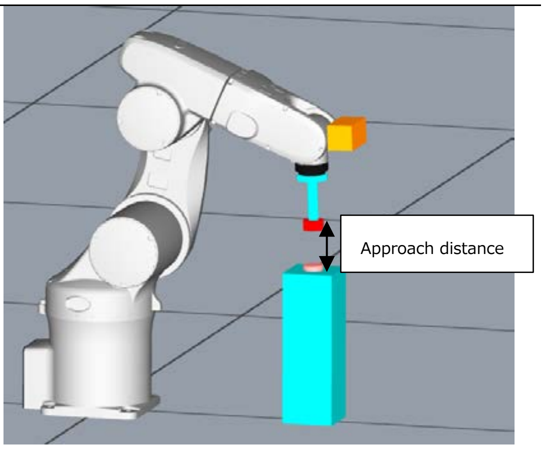
For a processing to move the robot to the robot approach position at measurement, follow the procedures below.

1

By the procedures at the step 2 in Chapter 6.6, check that the measurement results are stored in the variables

2

The depth of the container, the height of the workpiece position and the length of the hand are taken into account to determine the approach distance to the workpiece.



3

Set the referenced robot approach distance to the variable and add the robot approach distance to the measurement results acquired at step 2 in Chapter 6.6.

```
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! (5)Move the robot to the approach position
!     You have to configure the following robot approach
!    distance for your application.
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
std_appro_dist := 30;
```

Set the robot approach distance to the variable.

```
res_appro_pos_x := res_cmd_pos_x;
res_appro_pos_y := res_cmd_pos_y;
res_appro_pos_z := res_cmd_pos_z + std_appro_dist;
res_appro_pos_w := res_cmd_pos_w;
res_appro_pos_p := res_cmd_pos_p;
res_appro_pos_r := res_cmd_pos_r;
```

Add the robot approach distance to the measurement results

4

Set the variable as arguments for the robot motion sample function (fhsample_move) and execute it.

```
!!!!!!!!! CAUTION !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The following function drives a robot motion immediately.
! Confirm the settings before execution.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    fhsample_move res_appro_pos_x, res_appro_pos_y, res_appro_pos_z, res_appro_pos_w,
res_appro_pos_p, res_appro_pos_r,err_no;
```

Robot motion sample function

```
!Error check
IF err_no <> success THEN
    GOTO exit_program;
ENDIF
```

⚠ WARNING

- These operations drive the robot.
- Operate the robot in the state whereby pressing the [Emergency stop] button can stop its motion anytime.

6.8. Moving Robot to Robot Command Position at Measurement

For a processing to move the robot to the robot command position at measurement, follow the procedures below.

1 By the procedures at step 2 in Chapter 6.6, check that the measurement results are stored in variables.

2 Set the variables as arguments for the robot motion sample function (fhsample_move) and execute it.

```
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
! (6)Move the robot to the measured position
!     You have to edit this section for your application.
!     e.g.: Adding an end effector control, depart path motion, etc.
!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

!!!!!!!!!! CAUTION !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The following function drives a robot motion immediately.
! Confirm the settings before execution.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
fhsample_move res_cmd_pos_x, res_cmd_pos_y, res_cmd_pos_z, res_cmd_pos_w,
res_cmd_pos_p, res_cmd_pos_r, err_no;
```

Robot motion sample function

```
! Error check
IF err_no <> success THEN
    GOTO exit_program;
ENDIF
```

⚠ WARNING

- These operations drive the robot.
- Operate the robot in the state whereby pressing the [Emergency stop] button can stop its motion anytime.

6.9. Disconnecting Vision Sensor from Robot Controller

For a processing to disconnect the Vision Sensor from the Robot Controller, follow the procedures below.

1	Execute the disconnection function (fhclose) to the Vision Sensor.
	`!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;`
	`! 4. Example: Finalization sequence`
	`!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;`
	`!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;`
	`! (1)Disconnect from the FH server`
	`!;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;`
	`exit_program:`
	`IF bconnected = TRUE THEN`
	`    fhclose fh_socket, err_no;`
	`ENDIF`
	Disconnection function to the Vision Sensor (FH Server)

7. Function Reference

This chapter describes the functions for building a robot vision application

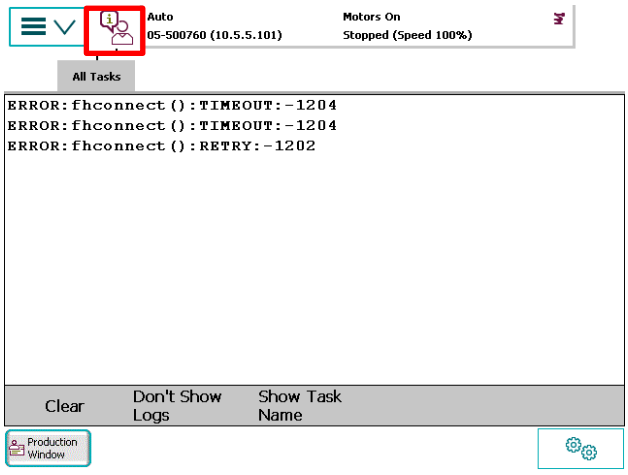
7.1. List of Functions

This is a list of functions that can be used by the actual driving robot program.

Function Name	Description	Reference
fhconnect	Connect to the Vision Sensor	Chapter 7.3.1
fhclose	Disconnects from the Vision Sensor	Chapter 7.3.2
fhsample_chgscn	Switching the scene of the Vision Sensor	Chapter 7.3.3
fhsample_regpos	Register the current robot position to the Vision Sensor	Chapter 7.3.4
fhsample_trig	Sends measurement commands to the Vision Sensor and receives the measurement results from the Vision Sensor	Chapter 7.3.5
fhsample_move	Move the robot	Chapter 7.3.6
fhrunsendcmd	Send a no-procedure command to the Vision Sensor	Chapter 7.3.7
fhrunrecvres	Receive a command response from the Vision Sensor	Chapter 7.3.8
fhrunrecvval	Receive numerical data from the Vision Sensor	Chapter 7.3.9

7.2. Error Message

The error message will be displayed on the operator window of the teaching pendant.



7.3. Function Details

7.3.1. fhconnect

■ Function

Connect to the Vision Sensor

■ Syntax

fhconnect fh_socket, ip_address, port_no, retries_connect, timeout_connect, err_no

■ Parameters

Argument	Input/Output	Data type	Description
fh_socket	Output	socketdev	Get the socket data of connect to Vision Sensor.
ip_address	Input	string	IP address of the Vision Sensor
port_no	Input	num	Port number of the Vision Sensor
retries_connect	Input	num	Number of connections retries (0 to 99)
timeout_connect	Input	num	Connection Timeout Time (0 to 99sec)
err_no	Output	num	Error number Store the result of the execution of this function.

■ Remarks

Return a socket data if connect to the Vision Sensor successful.
Connects to the Vision Sensor specified in the Number of socket parameter.
Request a connection every hour specified in the Connection Timeout Time parameter.
Request a connection as many times as specified in the connection retry count parameter.
Return an error if the connection fails.
Return an error if the parameter is out of the input range

■ Return Value

Err. No.	Error Message	Description
0	-	normal termination
-1200	ERROR:fhconnect():RETRY:-1200	Out of connection retry count input range
	ERROR: fhconnect():TIMEOUT:-1200	Connection Timeout Time Input Range
-1202	ERROR: fhconnect():RETRY:-1202	Connection retry count over
-1204	ERROR: fhconnect():TIMEOUT:-1204	Connection timeout time is over.

■ Precautions

Only one Vision Sensor can be connected to the robot controller.

If you want to connect to another Vision Sensor, disconnect from the connected Vision Sensor.

■ Example

In the following example, we will connect to the Vision Sensor with IP address "10.5.5.100" and port number "9876".

```
ip_address := "10.5.5.100";
port_no := 9876;
retries_connect := 2;
timeout_connect := 4;

fhconnect fh_socket, ip_address, port_no, retries_connect, timeout_connect, err_no;
```

7.3.2. fhclose

■ Function

Disconnects from the Vision Sensor

■ Syntax

fhclose fh_socket, err_no

■ Parameter

Parameter Name	Input/Output	Data type	Description
fh_socket	Input	socketdev	Data of socket already connected to the Vision Sensor
err_no	Output	num	Error number Store the result of the execution of this function.

■ Remarks

Disconnects from the Vision Sensor

■ Return Value

Err. No.	Error Message	Description
0	-	normal termination

■ Precautions

None

■ Example

In the following example, we will disconnect to the Vision Sensor.

```
fhclose fh_socket, err_no;
```

7.3.3. fhsample_chgscn

■ Function

Switching the scene of the Vision Sensor.

■ Syntax

fhsample_chgscn fh_socket, retries_recv, timeout_recv, scene_no, err_no

■ Parameter

Argument	Input/Output	Data type	Description
fh_socket	Input	socketdev	Data of socket already connected to the Vision Sensor
retries_recv	Input	num	Number of receive retries (0 to 99)
timeout_recv	Input	num	Receive timeout time (0 to 99sec)
scene_no	Input	num	Scene number to switch to (0 to 127)
err_no	Output	num	error number Store the result of the execution of this function.

■ Remarks

Sends a command to the Vision Sensor to switch to the scene number specified in the parameter.

It returns an error if this command is not connected to the Vision Sensor.

Return an error is returned if the scene number specified in the parameters is out of the input range.

Returns an error if a response is received from the Vision Sensor indicating that the scene change command failed.

■ Return value

Err. No.	Error Message	Description
0	-	normal termination
-1000	ERROR:fhsample_chgscn():Invalid Scene No.:-1000	Out of scene numbering range
-1300	ERROR:fhrecvstring():RETRY:-1300	Out of the range of receive retry count input
	ERROR:fhrecvstring():TIMEOUT:-1300	Out of the input range for the receive timeout count
-1301	ERROR:fhrecvstring():NO_CONNECTION:-1301	Calling in the unconnected state
-1303	ERROR:fhrecvstring():RETRY:-1303	Receive retry count overrun
-1304	ERROR:fhrecvstring():TIMEOUT:-1304	Receive timeout time is over.

-1601	ERROR:fhsample_chgscn():No Connection:-1601	Calling in the unconnected state
-1800	ERROR:fhsample_chgscn():Scene Change Failed:-1800	Response NG

■ Precautions

None

■ Example

In the following example, we will switch to scene 0.

```
retries_recv := 2;
timeout_recv := 4;
scene_no := 0;

fhsample_chgscn fh_socket, retries_recv, timeout_recv, scene_no, err_no;
```

7.3.4. fhsample_regpos

■ Function

Register the current robot position to the Vision Sensor.

■ Syntax

fhsample_regpos fh_socket, retries_recv, timeout_recv, err_no

■ Parameter

Argument	Input/Output	Data type	Description
fh_socket	Input	socketdev	Data of socket already connected to the Vision Sensor
retries_recv	Input	num	Number of receive retries (0 to 99)
timeout_recv	Input	num	Receive timeout time (0 to 99sec)
err_no	Output	num	error number Store the result of the execution of this function.

■ Remarks

Get the current robot position and register the current robot position to the Vision Sensor.
Return an error if this function is called while the Vision Sensor is not connected.
Returns an error if a response of current robot position registration failure is received from the Vision Sensor.

■ Return value

Err. No.	Error Message	Description
0	-	normal termination
-1300	ERROR:fhrecvstring():RETRY:-1300	Out of the range of receive retry count input
	ERROR:fhrecvstring():TIMEOUT:-1300	Out of the input range for the receive timeout count
-1301	ERROR:fhrecvstring():NO_CONNECTION: -1301	Calling in the unconnected state
-1303	ERROR:fhrecvstring():RETRY:-1303	Receive retry count overrun
-1304	ERROR:fhrecvstring():TIMEOUT:-1304	Receive timeout time is over.
-1505	ERROR:fhreflectcoord() : LOCAL_COORD_NO :-1505	Out of the range of Local Coordinate
	ERROR:fhreflectcoord() : TOOL_COORD_NO : -1505	Out of the range of Tool Coordinate
-1601	ERROR:fhsample_regpos():No Connection: -1601	Calling in the unconnected state

-1800	ERROR:fhsample_regpos():Trigger NG:-1800	Response NG

■ Precautions

None

■ Example

In the following example, the current robot position is registered to the Vision Sensor.

```
retries_recv := 2;
timeout_recv := 4;

fhsample_regpos fh_socket, retries_recv, timeout_recv, err_no;
```

7.3.5. fhsample_trig

■ Function

Sends measurement commands to the Vision Sensor and receives the measurement results from the Vision Sensor

■ Syntax

fhsample_trig fh_socket, retries_recv, timeout_recv, param{*}, err_no

■ Parameter

Argument	Input/Output	Data type	Description
fh_socket	Input	socketdev	Data of socket already connected to the Vision Sensor
retries_recv	Input	num	Number of receive retries (0 to 99)
timeout_recv	Input	num	Receive timeout time (0 to 99sec)
param{*}	Output	num	param{1}:Target X coordinate param{2}:Target Y coordinate param{3}:Target Z coordinate param{4}:Target W(EX) coordinate param{5}:Target P(EY) coordinate param{6}:Target R(EZ) coordinate
err_no	Output	num	error number Store the result of the execution of this function.

■ Remarks

Sends measurement commands to the Vision Sensor.
Receives the measurement results from the Vision Sensor and get the robot position.
Returns an error if called while not connected to the Vision Sensor.
Returns an error if a measurement command failure response is received from the Vision Sensor.
Returns an error if the Vision Sensor's overall judgment is NG.

■ Return Value

Err. No.	Error Message	Description
0	-	normal termination
-1300	ERROR:fhrecvstring():RETRY:-1300	Out of the range of receive retry count input
	ERROR:fhrecvstring():TIMEOUT:-1300	Out of the input range for the receive timeout count

-1303	ERROR:fhrecvstring():RETRY:-1303	Receive retry count overrun
-1304	ERROR:fhrecvstring():TIMEOUT:-1304	Receive timeout time is over.
-1502	ERROR:fhmeasureresult():PARAM_NUM: -1502	Abnormal number of parameters
-1601	ERROR:fhsample_trig():No Connection: -1601	Calling in the unconnected state
-1800	ERROR:fhmeasureresult():TRIG_NG:-1800	Response NG
-1801	ERROR:fhmeasureresult():TJG_NG:-1801	Overall judgment NG

■ Precautions

The measurement result received from the Vision Sensor will be stored in param{*}. To get the measurement result with this function, Result Output (Message) must be placed in the flow, and the settings must be as follows

Result Output (Message) Processing Item		The destination of the received measurement results
Setting Target	Setting details	
Output device	IoModule2: Serial (Ethernet)	-
Termination string	\r (Carriage Return)	-
Delimiter string	\x20 (Space)	-
0	Overall judgment (*1)	(Do not store)
1	Robot command position X (*1)	param {1}
2	Robot command position Y (*1)	param {2}
3	Robot command position Z (*1)	param {3}
4	Robot command position W (*1)	param {4}
5	Robot command position P (*1)	param {5}
6	Robot command position R (*1)	param {6}

*1: The output data format should be set as follows

- Data type: Number
- Digits of integer: 6
- Digits of decimal: 4

■ Example

In the following example, a measurement command is sent to the Vision Sensor, and after receiving the measurement result from the Vision Sensor, the robot moves to the position of the measurement result.

```
retries_recv := 2;
timeout_recv := 4;

fhsample_trig fh_socket, retries_recv, timeout_recv, param, err_no;

res_cmd_pos_x := param{1};
res_cmd_pos_y := param {2};
res_cmd_pos_z := param {3};
res_cmd_pos_w := param {4};
res_cmd_pos_p := param {5};
res_cmd_pos_r := param {6};

fhsample_move res_cmd_pos_x, res_cmd_pos_y, res_cmd_pos_z, res_cmd_pos_w,
res_cmd_pos_p, res_cmd_pos_r, err_no)
```

7.3.6. fhsample_move

■ Function

Move the robot

■ Syntax

fhsample_move pos_x, pos_y, pos_z, pos_w, pos_p, pos_r, err_no

■ Parameter

Argument	Input/Output	Data type	Description
pos_x	Input	num	Moving target robot coordinates X
pos_y	Input	num	Moving target robot coordinates Y
pos_z	Input	num	Moving target robot coordinates Z
pos_w	Input	num	Moving target robot coordinates W
pos_p	Input	num	Moving target robot coordinates P
pos_r	Input	num	Moving target robot coordinates R
err_no	Output	num	error number Store the result of the execution of this function.

■ Remarks

Moves the robot to the position specified by the parameter.

Returns an error if the target robot position is out of the movement range.

■ Return Value

Err. No.	Error Message	Description
0	-	normal termination
-1503	ERROR:fhsample_move():The Robot Position is Out of Range:-1503	out of range error
-1505	ERROR:fhreflectcoord() : LOCAL_COORD_NO :-1505	Out of the range of Local Coordinate
	ERROR:fhreflectcoord() : TOOL_COORD_NO : -1505	Out of the range of Tool Coordinate

■ Precautions

None

■ Example

In the following example, Move the robot to the (X, Y, Z, W, P, R)=(300,0,200,180,0,0)

```
pos_x := 300;
pos_y := 0;
pos_z := 200;
pos_w := 180;
pos_p := 0;
pos_r := 0;

fhsample_move pos_x, pos_y, pos_z, pos_w, pos_p, pos_r, err_no;
```

7.3.7. fhrunsendcmd

■ Function

Send a no-procedure command to the Vision Sensor

■ Syntax

fhrunsendcmd fh_socket, cmd_name, cmd_arg{*}, cmd_arg_num, err_no

■ Parameter

Argument	Input/Output	Data type	Description
fh_socket	Input	socketdev	Data of socket already connected to the Vision Sensor
cmd_name	Input	string	No-procedural commands to be sent to the Vision Sensor
cmd_arg{*}	Input	string	cmd_arg{1}: Argument 1 of the no-procedure command sent to the Vision Sensor(string). cmd_arg{2}: Argument 2 of the no-procedure command sent to the Vision Sensor(string). cmd_arg{3}: Argument 3 of the no-procedure command sent to the Vision Sensor(string). cmd_arg{4}: Argument 4 of the no-procedure command sent to the Vision Sensor(string). cmd_arg{5}: Argument 5 of the no-procedure command sent to the Vision Sensor(string).
cmd_arg_num	Input	num	Number of no-procedural command arguments to be sent to the Vision Sensor (0 to 5)
err_no	Output	num	error number Store the result of the execution of this function.

■ Remarks

Sends a no-procedure command to the Vision Sensor, concatenating the parameters according to the following format.

If the number of no-protocol command arguments is out of the input range, an error is returned.

<Format>

No-protocol command	SP(*1)	Command argument 1	SP	Command argument 2	SP	…	Command argument n(*2)

*1: “SP” is space

*2: The command argument n depends on the number of non-procedural command arguments.

■ Return Value

Err. No.	Error Message	Description
0	-	normal termination
-1506	ERROR:fhrunsendcmd():Invalid Command Argument No.:-1506	The number of no-procedural command arguments is out of the input range.
-1601	ERROR:fhsendstring():NO_CONNECTION: -1601	Calling in the unconnected state
	ERROR:fhsendstring():NO_DATA:-1601	Send string length 0
	ERROR:fhsendstring():CLOSE_SOCKET:-1601	Socket close
-1602	ERROR:fhsendstring():STRING_LEN:-1602	Send failure

■ Precautions

The length of the string of the no-stepping command that can be sent is 127 bytes (not including the delimiter).

Set the parameters of cmd_name and cmd_arg{*} so that the length of the string of the no-procedure command to be sent does not exceed 127 bytes.

■ Example

The following example shows how to send the measurement command "MEASURE" to the Vision Sensor

```
cmd_name := "MEASURE";
cmd_arg{1} := "";
cmd_arg{2} := "";
cmd_arg{3} := "";
cmd_arg{4} := "";
cmd_arg{5} := "";
cmd_arg_num := 0;

fhrunsendcmd fh_socket, cmd_name, cmd_arg, cmd_arg_num, err_no;
```

7.3.8. fhrunrecvres

■ Function

Receive a command response from the Vision Sensor

■ Syntax

fhrunrecvres fh_socket, retries_recv, timeout_recv, cmd_res, err_no

■ Parameter

Argument	Input/Output	Data type	Description
fh_socket	Input	socketdev	Data of socket already connected to the Vision Sensor
retries_recv	Input	num	Number of receive retries (0 to 99)
timeout_recv	Input	num	Receive timeout time (0 to 99sec)
cmd_res	Output	num	Command Response Result Storage Register Number (1: command response “OK” – 1: other than the command response “OK”)
err_no	Output	num	error number Store the result of the execution of this function.

■ Remarks

Receive the response (command response) to the no-procedure command sent to the Vision Sensor.

If the command response is OK, assign 1 to the command response result cmd_res.

If the command response is not OK, assign "-1" to the command response result cmd_res.

■ Return Value

Err. No.	Error Message	Description
0	-	normal termination
-1300	ERROR:fhrecvstring():RETRY:-1300	Out of the range of receive retry count input
	ERROR:fhrecvstring():TIMEOUT:-1300	Out of the input range for the receive timeout count
-1301	ERROR:fhrecvstring():NO_CONNECTION: -1301	Calling in the unconnected state
-1303	ERROR:fhrecvstring():RETRY:-1303	Receive retry count overrun
-1304	ERROR:fhrecvstring():TIMEOUT:-1304	Receive timeout time is over.

■ Precautions

None

■ Example

In the following example, a command response is received from the Vision Sensor, and if the received command response is not OK (0), the program is terminated.

```
retries_recv := 2;
timeout_recv := 4;
fhrunrecvres fh_socket, retries_recv, timeout_recv, cmd_res, err_no;

IF err_no <> success THEN
    GOTO exit_program;
ENDIF

IF cmd_res <> 1 THEN
    GOTO exit_program;
ENDIF
```

7.3.9. fhrunrecvval

■ Function

Receive numerical data from the Vision Sensor

■ Syntax

fhrunrecvval fh_socket, retries_recv, timeout_recv, param{*}, err_no)

■ Parameter

Argument	Input/Output	Data type	Description
fh_socket	Input	socketdev	Data of socket already connected to the Vision Sensor
retries_recv	Input	num	Number of receive retries (0 to 99)
timeout_recv	Input	num	Receive timeout time (0 to 99sec)
param{*}	Output	num	Results of the analysis of the received numerical sequence (Element count: 10)
err_no	Output	num	error number Store the result of the execution of this function.

■ Remarks

This function stores the numerical data sent from the Vision Sensor into the register specified by the parameter.

This function outputs up to 10 values.

If there are more than 11 numbers, this function outputs only the first 10.

If the length of the segmented string is longer than 12 bytes, this function returns an error of abnormal parameter length.

Returns an error if the number of numeric data is zero.

The following is an example of the output when a string is included.

<Conversion example>

The string before conversion	The result of the analysis after conversion
abc	0
123abc	0
abc123	0
1.00E+03	1000

■ Return Value

Err. No.	Error Message	Description
0	-	normal termination
-1300	ERROR:fhrecvstring():RETRY:-1300	Out of the range of receive retry count input
	ERROR:fhrecvstring():TIMEOUT:-1300	Out of the input range for the receive timeout count
-1301	ERROR:fhrecvstring():NO_DATA:-1301	Receive data length 0
	ERROR:fhrecvstring():NO_CONNECTION: -1301	Calling in the unconnected state
-1302	ERROR:fhrecvstring():STRING_LEN:-1302	Receive data length over
-1303	ERROR:fhrecvstring():RETRY:-1303	Receive retry count overrun
-1304	ERROR:fhrecvstring():TIMEOUT:-1304	Receive timeout time is over.
-1502	ERROR:fhparseval():PARAM_TOO_LONG: -1502	Abnormal parameter length
	ERROR:fhparseval():PARAM_NUM:-1502	Abnormal number of parameters
-1504	ERROR:fhparseval():NO_CMD:-1504	The length of the string to be divided is 0.

■ Precautions

The maximum length of the string to be received is 127 bytes (not including the delimiter). if more than 128 bytes are received, an error is returned.

To get the measurement result with this function, Result Output (Message) must be placed in the flow, and the settings must be as follows

Result Output (Message) Processing Item		The destination of the received measurement results
Setting Target	Setting details	
Output device	IoModule2: Serial (Ethernet)	-
Termination string	\r (Carriage Return)	-
Delimiter string	\x20 (Space)	-
Output Data 0 - 9	numerical data(*1)	param{1} – param{10}

*1: The output data format should be set as follows

- Data type: Number
- Digits of integer: 6
- Digits of decimal: 4

■ Example

In the following example, the program receives a sequence of numbers sent by the Vision Sensor and exits the program if the first received data is not 1.

```
retries_recv := 2;
timeout_recv := 4;

fhrunrecvval fh_socket, retries_recv, timeout_recv, outputdata, err_no;

IF err_no <> success THEN
    GOTO exit_program;
ENDIF

IF outputdata{1} <> 1 THEN
    GOTO exit_program;
ENDIF
```

OMRON Corporation **Industrial Automation Company**
Kyoto, JAPAN
Contact: www.ia.omron.com

Regional Headquarters
OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

Cat. No. Z459-E1-01 0621