Manual

FlexiBowl®

Yaskawa YRC1000 Plug-In

ars automation

INDEX

This Plugin was developed with the idea of communicating **quickly and safely with FlexiVision** via the **YASKAWA** robot.

The Plug-in requires the "**MotoPLUS**" package installed on the robot.

The robot controller version must be **YRC1000**.

FlexiBowl® is a partner product of Yaskawa.

FlexiBowl® Plug-In

Plug-In Installation

Step 1.

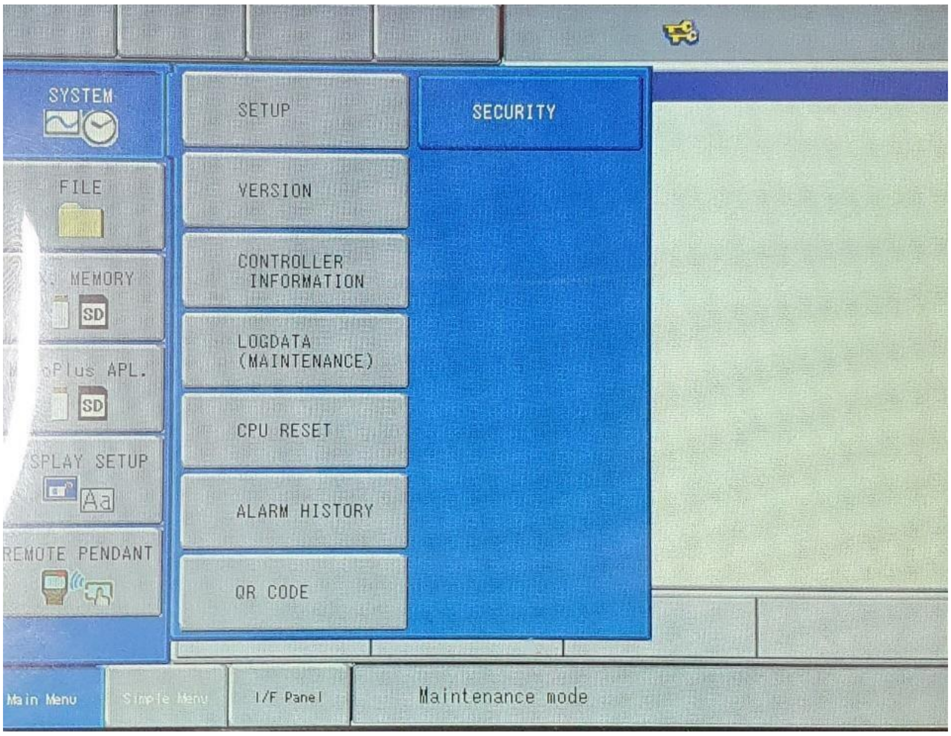
In order to install the Plug-in, it is necessary to insert a USB memory stick with the ***YaskawaFlexiVisionVX.out*** file (located in the *Yaskawa_FlexiVisionV0_Plug-in* folder) provided and start the robot in MAINTENANCE mode:

a) Press and hold down the MAIN MENU button on the pendant and turn the robot on at the same time.

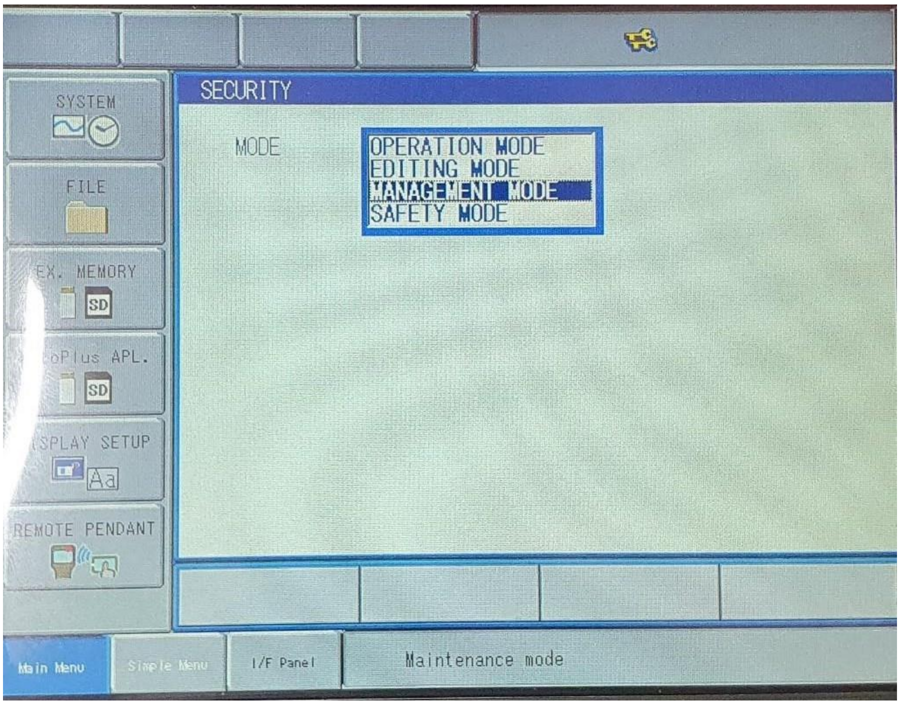
b) Wait until the robot is switched on.

Now, change the robot's security level to MANAGEMENT MODE: go to: MAIN MENU > SYSTEM > SECURITY and change the MODE field to MANAGEMENT MODE.

Default password: 9999999999999999



Plug-In Installation

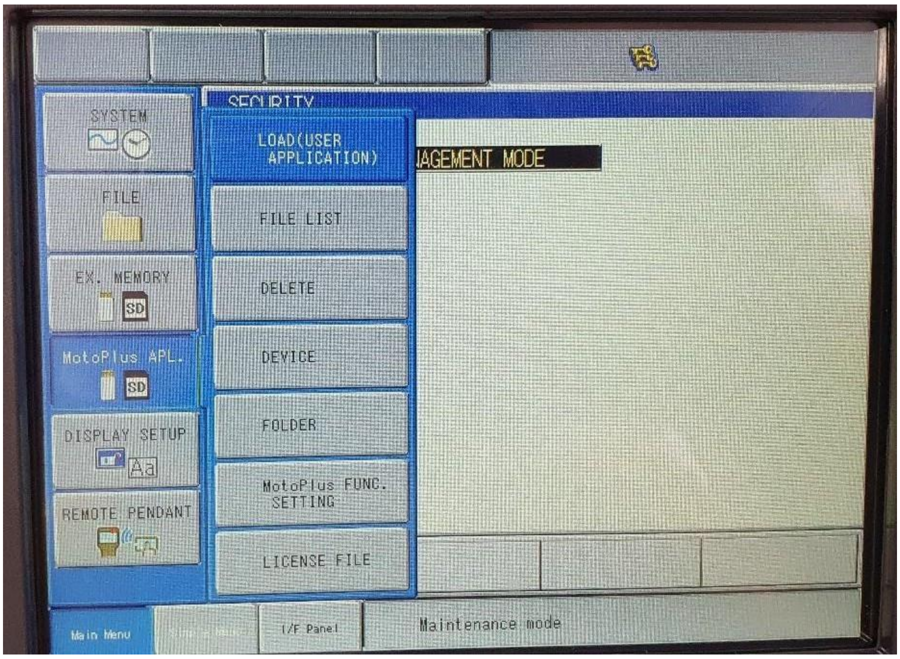

SECURITY

MODE

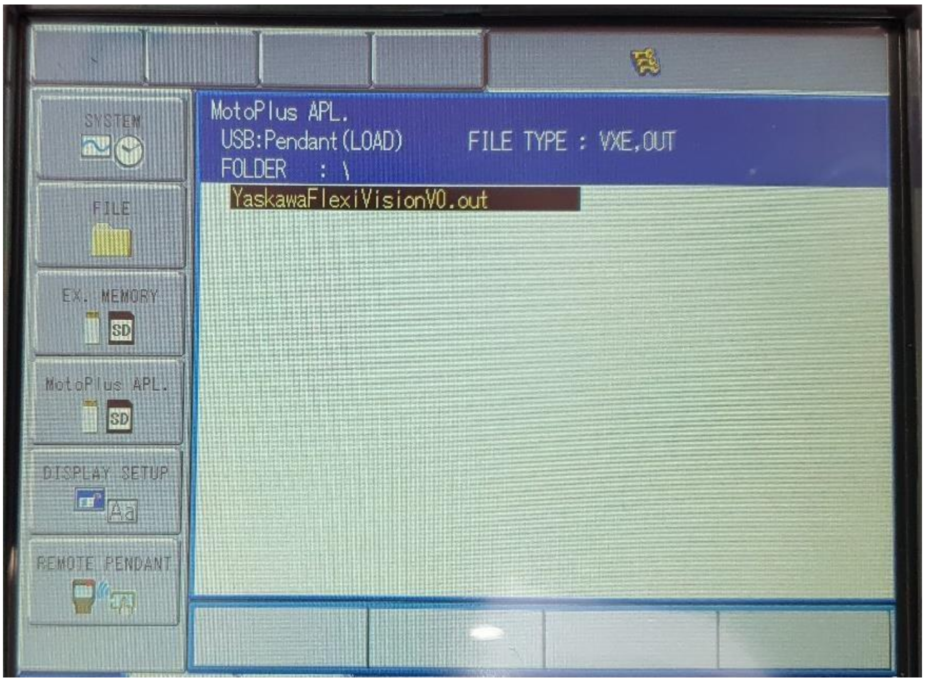
OPERATION MODE
EDITING MODE
MANAGEMENT MODE
SAFETY MODE

Now go to: MAIN MENU > MOTOPLUS APL. > LOAD(USER APPLICATION) and select ***YaskawaFlexiVisionVX.out*** using the SELECT button.

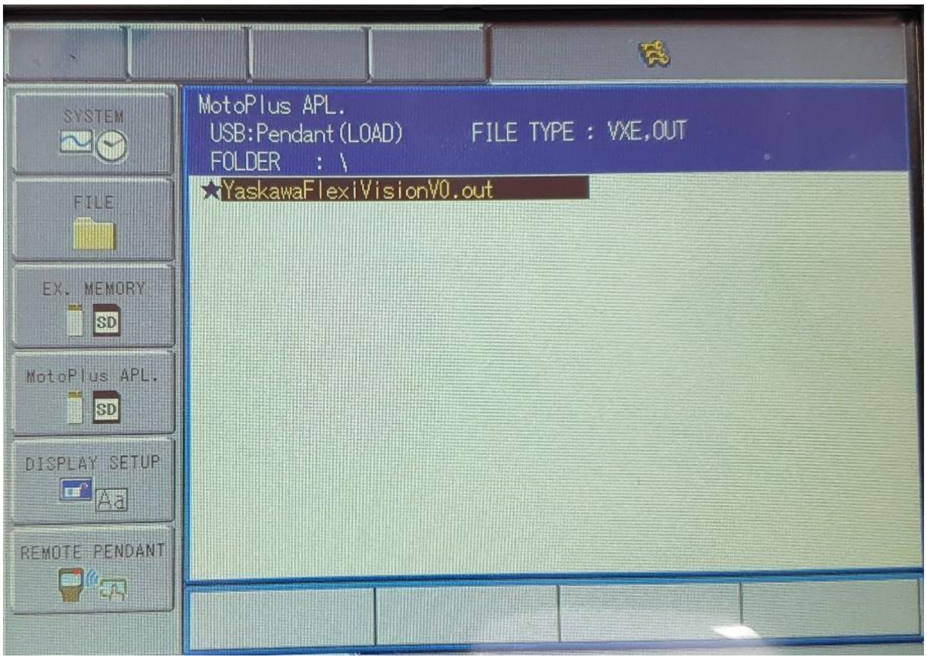
A star appears to the left of the selected file.



Plug-In Installation



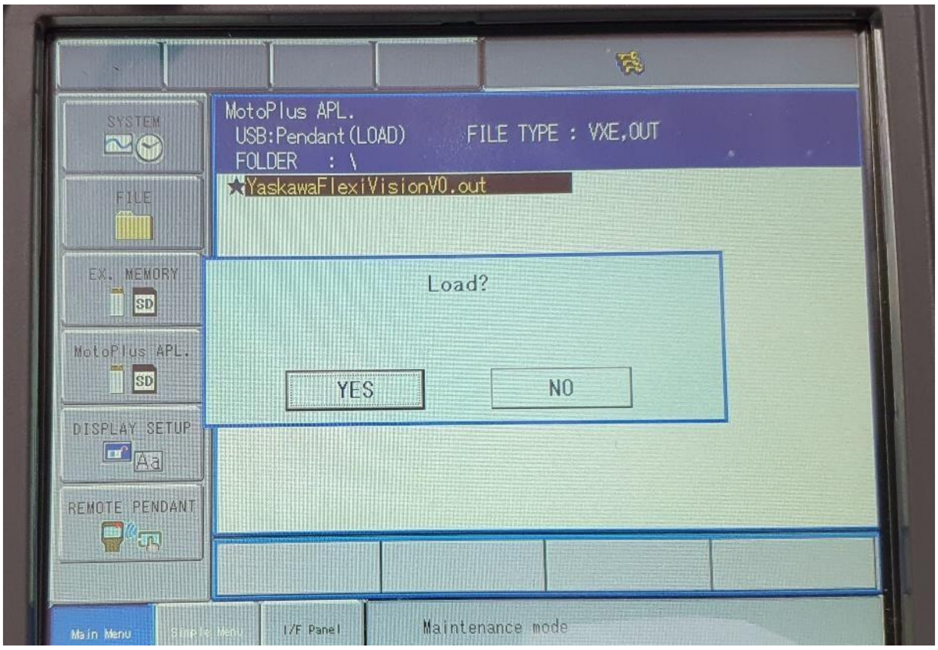
MotoPlus APL.
USB:Pendant(LOAD) FILE TYPE : VXE,OUT
FOLDER : \
YaskawaFlexiVisionV0.out



MotoPlus APL.
USB:Pendant(LOAD) FILE TYPE : VXE,OUT
FOLDER : \
★YaskawaFlexiVisionV0.out

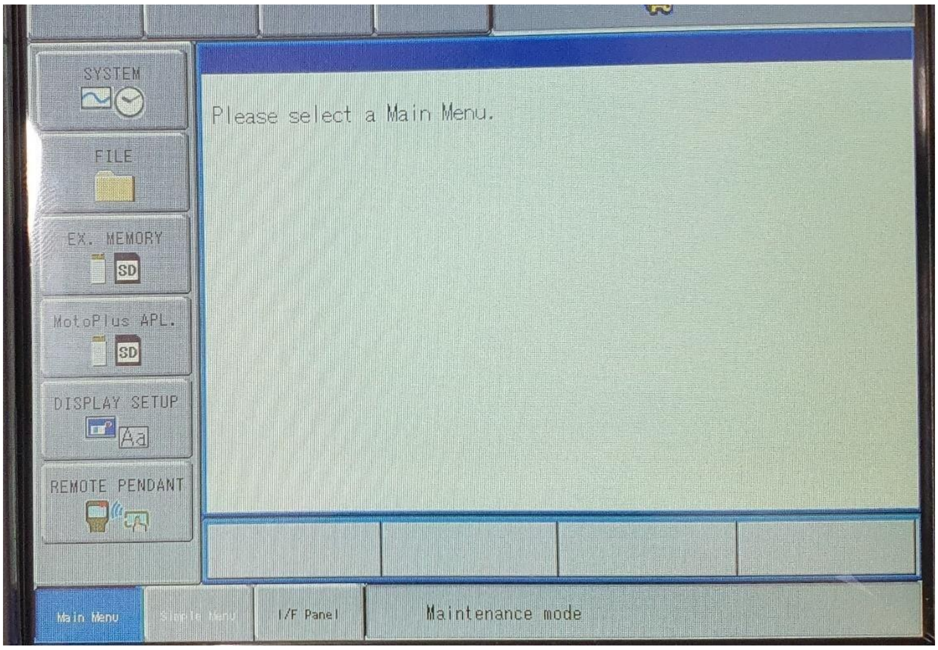
Press **ENTER** and select YES at the LOAD? popup.

Plug-In Installation


MotoPlus APL.
USB:Pendant(LOAD) FILE TYPE : VXE,OUT
FOLDER : \
★YaskawaFlexiVisionV0.out

Load?

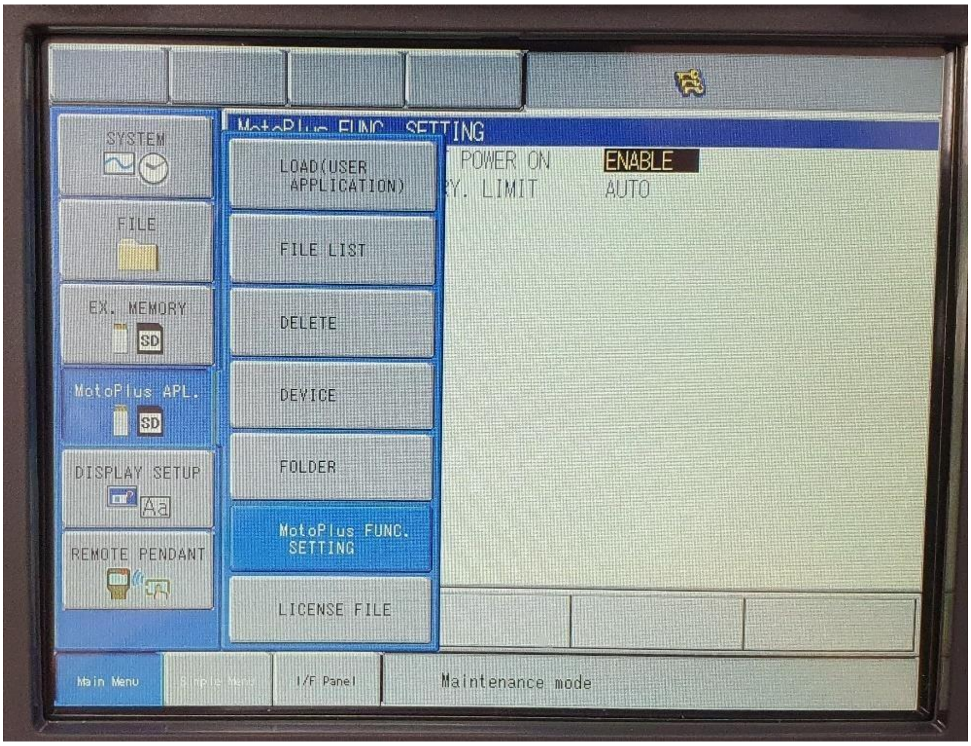
A few soft “beeps” will be played and you will be redirected to the next screen.


Please select a Main Menu.

Plug-In Installation

Make sure ***AUTOSTART*** is enabled for the execution of the application at controller start-up.

Go to: MAIN MENU > MOTOPLUS APL. > MOTOPLUS FUNC.SETTING and set APPLI.AUTOSTART AT POWER ON on the ENABLE value.



MotoPlus FUNC. SETTING

POWER ON ENABLE

Y. LIMIT AUTO

Proceed with restarting the controller.

Plug-In Installation

Step 2.

Restarting the controller, implemented in Step 1, allows the newly installed Plug-in to be started. ***Wait a few seconds*** after switching on to allow it to start correctly. The robot will then open a server service for each network interface available at port 8001.

Now go to MAIN MENU > VARIABLE and open the INTEGER VARIABLE window. Set integer variable no. 93 to value 1 (or any value other than 0).

It will then now be **necessary** to follow the instructions in the FlexiVision user manual to physically connect the devices together (camera, FlexiBowl, PC, robot) and establish ***FlexiVision-FlexiBowl*** (using IP address of FlexiBowl) and ***FlexiVision-Robot*** (using IP address of robot, port 8001) ***communications***.

If the ***FlexiVision-Robot connection*** was successful, the value of the integer variable no. 93 will automatically be set to 0.
At this point, you can use the robot to request the execution of a command to FlexiVision. This request can be made ***via a pendant*** or ***via*** any other ***application*** running on the robot.

If the connection failed (integer no. 93 different from 0), the connections between the devices, their IP addresses and the FlexiBowl® and robot settings should be checked. If necessary, a debugging session can be started to better understand the cause of the error.

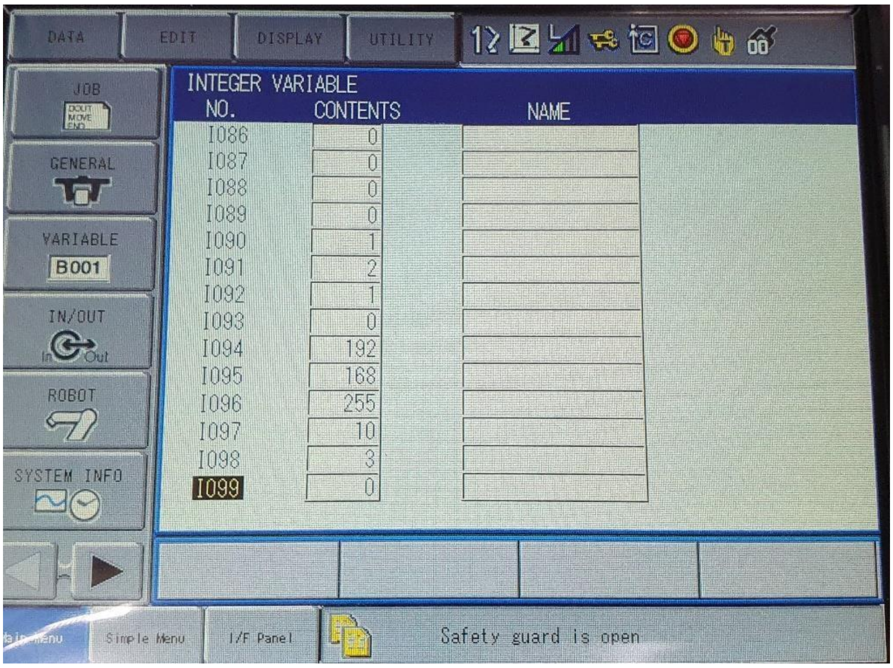
Using FlexiVision

VIA PENDANT

In the integer variables window, you can select the command to be sent to FlexiVision and request the execution of the command via index variables 92 and 93.

Variable 92 allows the selection of a command in Table A by setting its value to one of the values in the range 1-8 corresponding to the command index.

For example, in the following image, the value set for variable 92 is 1, so the selected command will be “start_Locator”. Do not consider the value of the other variables for the time being.



INTEGER VARIABLE

NO.	CONTENTS	NAME
I086	0	
I087	0	
I088	0	
I089	0	
I090	1	
I091	2	
I092	1	
I093	0	
I094	192	
I095	168	
I096	255	
I097	10	
I098	3	
I099	0	

To request execution of the command on FlexiVision, simply set the value of index variable 93 to 1.

The command will now be executed on FlexiVision and the integer no. 93 will remain at 1 as long as the command is running. As soon as the latter is finished, the value of integer no. 93 will return to 0 pending the next command.

Using FlexiVision

VIA ROBOT APPLICATION

As in the previous case, the execution of a command via any robot application is possible via the index integers 92 and 93.

Variable 92 allows the selection of a command in Table A by setting its value to one of the values in the range 1-8 corresponding to the command index.

To request execution of the command on FlexiVision, simply set the value of index variable 93 to 1.

The command will now be executed on FlexiVision and the integer no. 93 will remain at 1 as long as the command is running.

As soon as the latter is finished, the value of integer no. 93 will return to 0 pending the next command.

Execution and output

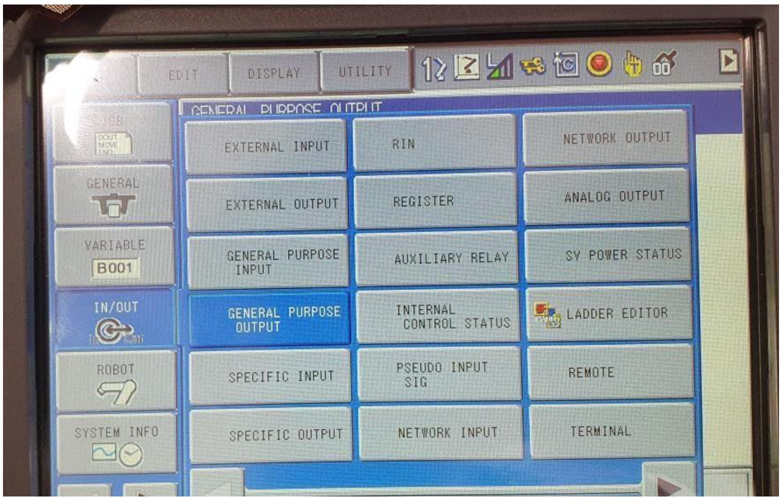
In this chapter, a few peculiarities of the Plug-in relating to execution and the outputs produced by it are given.

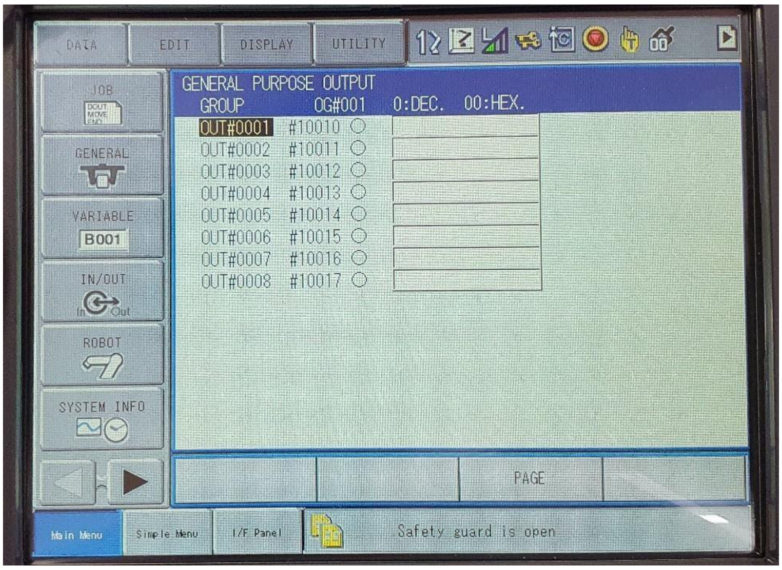
1. It is necessary to ***set the recipe manually*** via FlexiVision as the Plug-in does not allow this possibility.
2. The FlexiVision recipe must return strings in the format "***Pattern_ID;x;y;rotZ***", "***Control_ID;x;y;rotZ***", "***Hopper;output;time***" where:
   - **a. ID:** ID of the identified object (> 0).
   - **b. x:** X co-ordinate of the identified object or **NULL**.
   - **c. y:** Y co-ordinate of the identified object or **NULL**.
   - **d. rotZ:** rotation along the Z axis of the identified object or **NULL**.
   - **e. output:** index of the digital output to which the hopper is connected. ***Value within the range***: ***10010 – 15127***.
   - **f. time:** time duration of hopper activation, expressed in ms.
3. If the string returned by FlexiVision is of type "***Pattern_ID;x;y;rotZ***", the output values will be saved in the following robot variables:
   - **a. Variable double no. 99:** received X co-ordinate multiplied by 1000.
   - **b. Variable double no. 98:** received Y co-ordinate multiplied by 1000.
   - **c. Variable double no. 97:** rotation along Z axis multiplied by 10000.
   - **d. Integer variable no. 90:** ID of the identified object.
   - **e. Integer variable no. 88:** value 1 (valid co-ordinates).
   - **f. Integer variable no. 91:** value 2 (Locator case).
   - **g. Integer variable no. 93:** value 0 (command execution terminated).

Execution and output

4. If instead the returned string is "***Pattern_ID;NULL;NULL;NULL***", the value of integer variable no. 88 will be saved as 0 (invalid co-ordinates) and the value of integer variable no. 93 will be saved as 0 (command execution terminated).
5. If the string returned by FlexiVision is of type "***Control_ID;x;y;rotZ***", the output values will be saved in the following robot variables:
   - a. **Variable double no. 99:** received X co-ordinate multiplied by 1000.
   - b. **Variable double no. 98:** received Y co-ordinate multiplied by 1000.
   - c. **Variable double no. 97:** rotation along Z axis multiplied by 10000.
   - d. **Integer variable no. 90:** ID of the identified object.
   - e. **Integer variable no. 88:** value 1 (valid co-ordinates).
   - f. **Integer variable no. 91:** value 1 (Control case).
   - g. **Integer variable no. 93:** value 0 (command execution terminated).
6. If instead the returned string is "***Control_ID;NULL;NULL;NULL***", the value of integer variable no. 88 will be saved as 0 (invalid co-ordinates) and the value of integer variable no. 93 will be saved as 0 (command execution terminated).
7. If the string returned by FlexiVision is of the type "***Hopper;output;time***", the robot will take care of activating, with a parallel task, the hopper located at the address specified in "output" of the *GENERAL PURPOSE OUTPUT* digital outputs (range: 10010 – 15127) per "time" milliseconds. "output" and "time" are values chosen by the user via the FlexiVision interface.

Execution and output





```
GENERAL PURPOSE OUTPUT
 GROUP        OG#001   0:DEC.  00:HEX.
 OUT#0001  #10010 ○
 OUT#0002  #10011 ○
 OUT#0003  #10012 ○
 OUT#0004  #10013 ○
 OUT#0005  #10014 ○
 OUT#0006  #10015 ○
 OUT#0007  #10016 ○
 OUT#0008  #10017 ○
```

Safety guard is open

8. Any error cases will be notified via the integer variable no. 87:

a. ***Value 0:*** No error.

b. ***Value 1:*** FlexiVision error.

c. ***Value 2:*** Socket communication error.

If the error originates from socket communication, the first step to resolve it is to restart the controller and then start again from Chapter 1 – Step 2. Otherwise, if the error comes from FlexiVision, simply fix the specified problem and then restart from Chapter 2, without restarting.

In any case the error cannot be resolved, a restart of the controller is strongly recommended. For in-depth research into the cause of the error, a debugging session can be started (Chapter 4).

Execution and output

Summary of robot I/O

I/O	ROBOT VARIABLE	VALUE	VALUE DESCRIPTION
INPUT / OUTPUT	Integer no. 93	0	Robot ready to request FlexiVision command / FlexiVision command execution finished
		1	FlexiVision Command execution request start / Execution in progress
INPUT	Integer no. 92	1	start_Locator
		2	stop_Locator
		3	turn_Locator
		4	test_Locator
		5	start_Control
		6	start_Empty
		7	set_Recipe
		8	get_Recipe
OUTPUT	Integer no. 91	0	Invalid case
		1	Locator Case
		2	Control Case
OUTPUT	Integer no. 90	0	Invalid ID
		> 0	ID of the identified object
OUTPUT (RESERVED)	Integer no. 89	0	Hopper waiting to be activated
		1	Hopper activated
OUTPUT	Integer no. 88	0	Invalid co-ordinates
		1	Valid co-ordinates
OUTPUT	Integer no. 87	0	No error
		1	FlexiVision Error
		2	Socket communication error

Execution and output

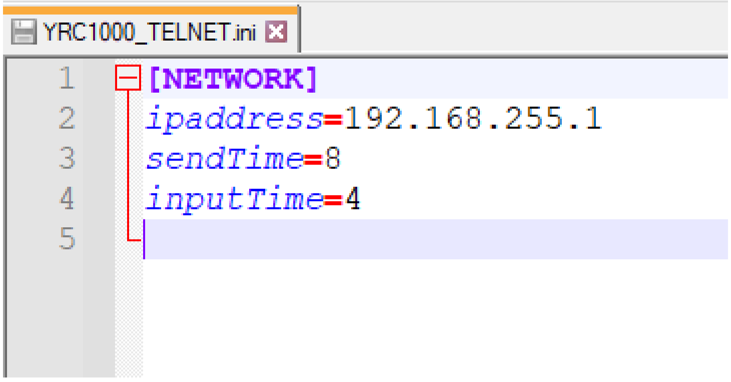
I/O	ROBOT VARIABLE	VALUE	VALUE DESCRIPTION
OUTPUT	Double no. 99	∞	X Co-ordinate received (x1000)
OUTPUT	Double no. 98	∞	Y Co-ordinate received (x1000)
OUTPUT	Double no. 97	∞	Z-axis rotation received (x10000)
INPUT	Integer no. 85	>= 1	Selected recipe, ready for “set_Recipe” command.
OUTPUT	Integer no. 86	>= 1	FlexiVision current active recipe returned by “get_Recipe” command

Debugging

If you need to identify the nature of any problems or follow the progress of the Plug-in on the robot, ***you need to start a Telnet session*** using a PC in order to receive status information from the robot.

First of all, check that the PC connected to the robot has an IP address in the same subnet as the robot.

It is necessary to edit the file "***YRC1000_TELNET.ini***" inside the ***YRC1000_TELNET*** folder by setting the value "ipaddress" to the robot's IP address. For example, in the following image, the robot's IP address is 192.168.255.1



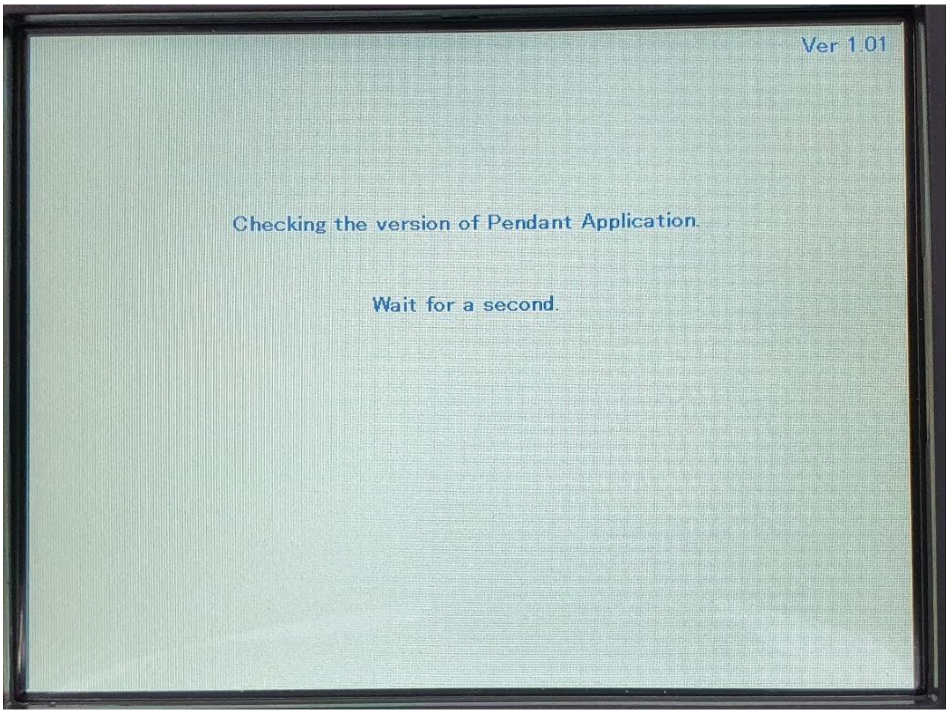
```
[NETWORK]
ipaddress=192.168.255.1
sendTime=8
inputTime=4

```

Now, while powering up the controller, you need to run the ***YRC1000_TELNET.exe*** file (inside the YRC1000_TELNET folder) on your PC, which will initiate a Telnet connection with the robot and show its current status on the screen.

It is advisable to start ***YRC1000_TELNET.exe*** as soon as the following screen is shown in the Yaskawa pendant during power-up, in order for the Telnet to connect correctly.

Debugging



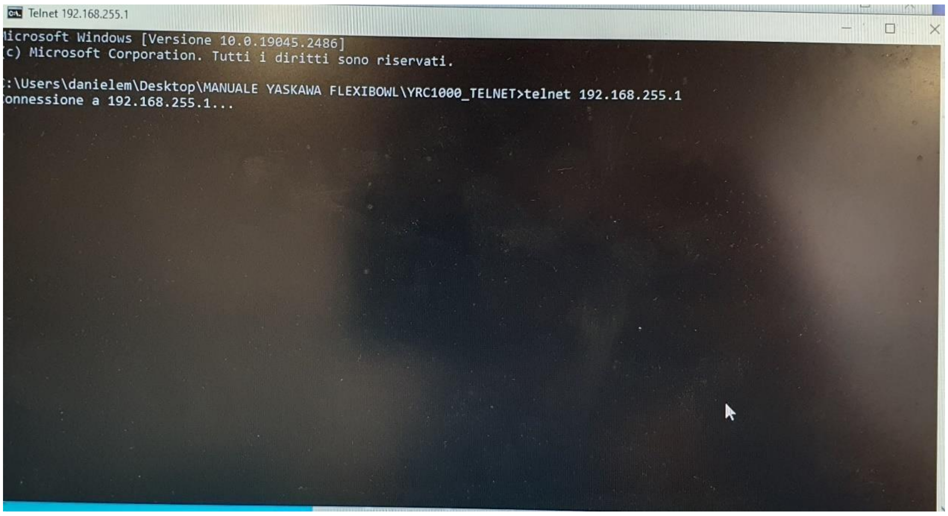
Ver 1.01

Checking the version of Pendant Application.

Wait for a second.

If the Telnet connection is successful, a prompt window will automatically open showing the robot's status messages (including error states).

See the following images.

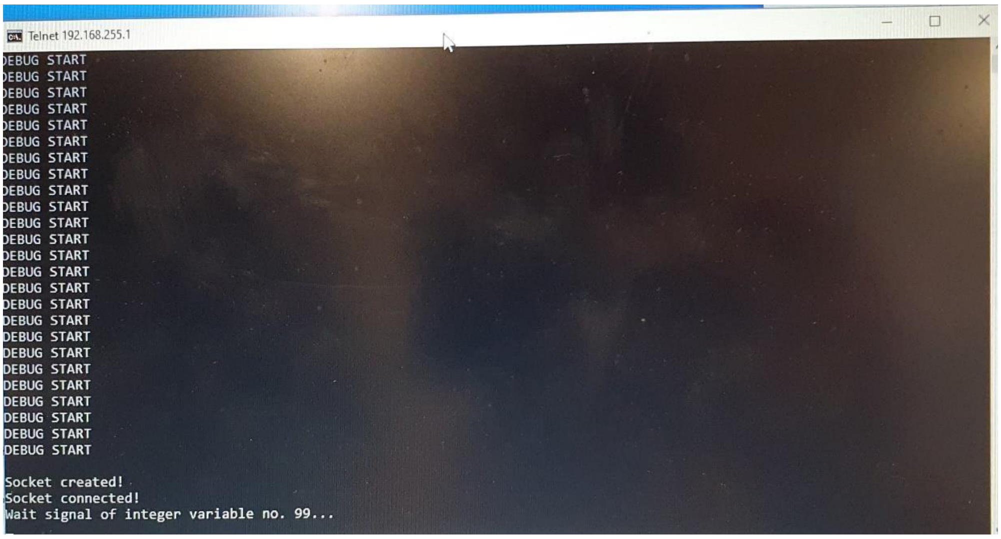


Telnet 192.168.255.1

```
licrosoft Windows [Versione 10.0.19045.2486]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

:\Users\danielem\Desktop\MANUALE YASKAWA FLEXIBOWL\YRC1000_TELNET>telnet 192.168.255.1
onnessione a 192.168.255.1...
```

Debugging



```
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START
DEBUG START

Socket created!
Socket connected!
Wait signal of integer variable no. 99...
```

If you do not get the following messages on the screen, you must restart the controller and retry the Telnet connection.

FlexiVision Command List

To send the command to FlexiVision you must modify the value of the “command” string.

N_Mission	Command	Action
1	“start_Locator"	Starts the parts localization process by recalling the FlexiBowl handling routine in case there are no parts that can be picked up. ***Return:*** “Pattern1;x;y;r”.
2	“stop_Locator"	Stops the process of locating the object with the aid of the FlexiBowl.
3	“turn_Locator"	If no parts are picked up, by this command the operator can make the FlexiBowl rotate and the “start_Locator” routine start. ***Return:*** “Pattern1;x;y;r”.
4	“test_Locator"	Starts the process of locating the object without the aid of the FlexiBowl. ***Return:*** “Pattern1;x;y;r”.
5	“start_Control"	Starts the inspection cycle. ***Return:*** “Control1;x;y;r”.
6	“state_Locator"	Locator status diagnostics is shown: ***Return:*** “Locator is Running” “Locator is in Error” “Locator is not Running”.
7	“start_Empty"	Start the FlexiBowl® Quick-Emptying sequence. ***Return:*** “start_Empty ended”
8	“get_Recipe"	The name of the recipe currently loaded on FlexiVision is shown. ***Return:*** “recipe name”.
9	“set_Recipe=recipe name"	The recipe corresponding to the sent "recipe name" is loaded.