

# Manual



# FlexiVision

KAWASAKI PLUG-IN

**ars**

Feeding  
Industrial  
Robotics

# INDEX

1. Plug-In Installation
2. Plug-In Configuration
3. FlexiVision Command List

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

## FlexiBowl<sup>®</sup> Plug-In

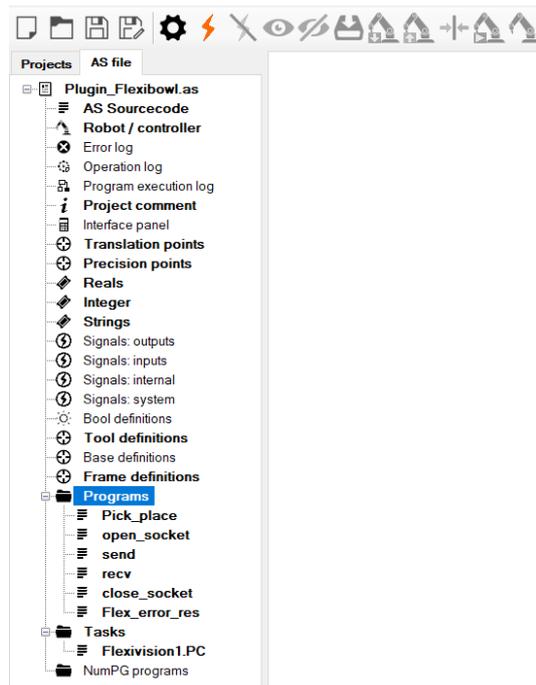


# Plug-In Installation

Using the **KIDE** program, open the file provided by “**ARS srl**” **Flexivision\_Plugin1.AS** via the **File/Open** path.



At the end of the loading procedure, the following items should be displayed in bold.



It is now possible to modify the programs and data within the **.AS file**.

# Plug-In Configuration

## Step 1.

### Description of the Flexivision1.PC communication program

The **Flexivision1.PC** program allows communication between Flexivision and the **Kawasaki** Robot.

Below is a description of the main variables used in the code. These variables must not be changed in the **Flexivision1.PC** program but in the **Pick\_place** program. (See 3. Configuration of the Pick\_place parameters)

- **Port:** Indicates the port used for communication with Flexivision, by **default 8192**.
- **Flexi\_resp:** Contains Flexivision's response to the command sent.
- **Pattern\_id:** The assumed values are shown below:
  - **-1:** The sent command is "test\_Locator" and no Pattern was found. (Response from Flexivision "Null;Null;Null;Null")
  - **0:** The sent command does not return a Pattern location. (e.g.: "get\_Recipe")
  - **Val>0:** Id of the detected Pattern. ("**Pattern\_1;10;11.2;4;**" Val assumes value 1)
- **Error\_Flb:** The assumed values are shown below:
  - **0:** There is no error.
  - **1:** The disconnection between Flexivision and the Kawasaki Robot has occurred.
  - **2:** Flexivision sent an error (e.g. an error on the camera), this error can be seen in *Flexi\_err\_str*.
  - **3:** command not defined for Flexivision.
- **Flexi\_err\_str:** Contains the generated error messages.

# Plug-In Configuration

- **Flexi\_Flag:** Allows execution of the command. By setting "**Flexi\_Flag=0**" the command in the "**command**" string variable is sent. After the communication "**Flexi\_Flag**" is automatically set to 1.
- **Flexi\_stop:** By setting "**Flexi\_stop=1**", the "**stop\_Locator**" command is sent. The "**stop\_Locator**" command can be sent in any condition using the instructions:  
**Flag\_Flexi=0**  
**Flexi\_stop=1**  
After the stop, "**Flexi\_stop**" is automatically reset to 0.
- **Flexi\_stopped:** It is automatically set to 1 when the "**stop\_Locator**" command has been sent. It must be reset to the default value 0, through the **Pick\_Place** program, to be able to send other commands to Flexivision.
- **Command:** Contains the command sent to Flexivision when "**Flexi\_Flag**" is set to 0.
- **Flexi\_point:** Contains the coordinates returned by Flexivision.

The Flexivision1.PC program uses the following programs:

- **Open\_socket**
- **Send**
- **Recv**
- **Close\_Socket**

For correct management of the communication Socket.

## Step 2.

### Configuration of the Pick\_Place program parameters

- **"Fixed" coordinates:** **z\_pos**, **rx\_pos**, **ry\_pos** are not detected by the vision system and must be configured according to your robot configuration. These coordinates refer to the "**Flexi\_frame**" reference system.
- **Flexi\_frame:** It represents the common frame of the robot and Flexivision. Proceed with the calibration of the camera and the "**teach**" function of the same frame with the name "**Flexi\_frame**".
- **Tool configuration**
- **Configuration of the communication**

# Plug-In Configuration

```
;CARICO LA RICETTA E VERIFICO CHE SIA STATA CARICATA
$command="set_recipe=kawasaki_plugin"
Flexi_Flag = 0
WAIT Flexi_Flag==1
```

The command to be sent must be assigned to the "**command**" string to send the command to Flexivision. By setting "**Flexi\_Flag=0**", the command is sent. **Flexi\_Flag** is automatically set to 1 by **Flexivision1.PC** once the response is obtained from "Flexivision".

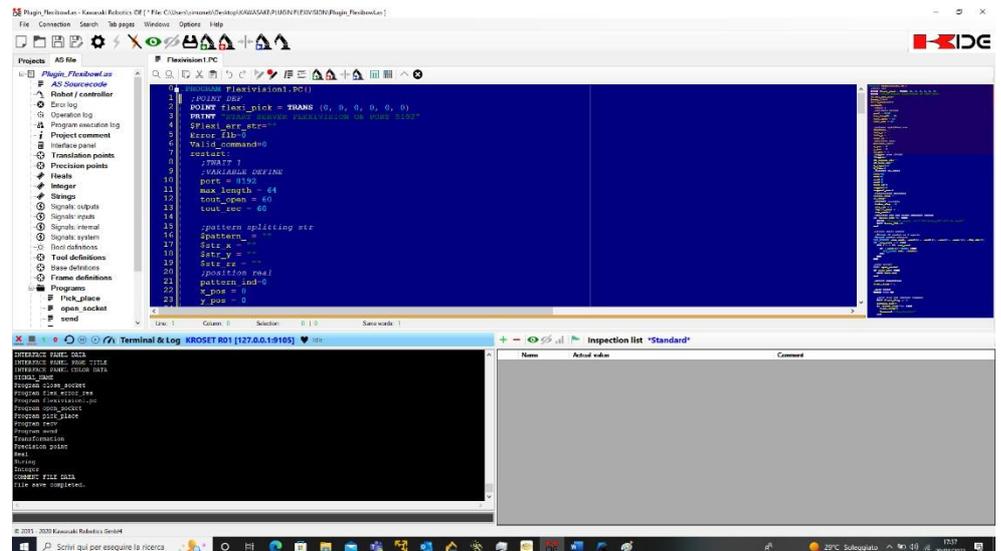
The **Pick\_Place** program calls the **Flex\_Error\_Res** function when an error occurs. This function allows the "**Error\_Flb**" flag to be reset. It will not be possible to send commands to Flexivision until this Flag is reset.

Enter any procedures to be performed in the event of an error in the "**Flex\_Error\_Res**" function.

## Step 3.

### Loading Plug-In on the real controller

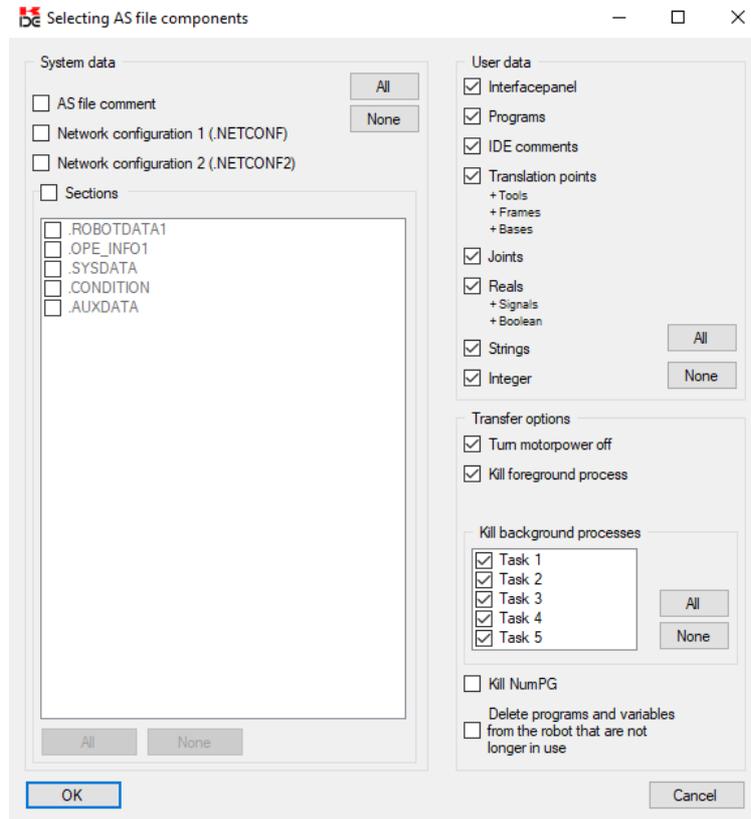
After making the necessary changes, load the Plug-in on the real controller. The connection with the physical controller is established via the ⚡ icon.



# Plug-In Configuration

## Step 4.

Send the **.AS** file to the controller via the icon  by selecting the ticks on all the **"User data"**, as shown below.



# 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. <b>Return:</b> "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. <b>Return:</b> "Pattern1;x;y;r".
4	"test_Locator"	Starts the process of locating the object without the aid of the FlexiBowl. <b>Return:</b> "Pattern1;x;y;r".
5	"start_Control"	Starts the inspection cycle. <b>Return:</b> "Control1;x;y;r".
6	"state_Locator"	Locator status diagnostics is shown: <b>Return:</b> "Locator is Running" "Locator is in Error" "Locator is not Running".
7	"start_Empty"	Start the FlexiBowl® Quick-Emptying sequence. <b>Return:</b> "start_Empty ended"
8	"get_Recipe"	The name of the recipe currently loaded on FlexiVision is shown. <b>Return:</b> "recipe name".
9	"set_Recipe=recipe name"	The recipe corresponding to the sent "recipe name" is loaded.