Manual

FlexiVision

FANUC PLUG-IN

ars | Feeding Industrial Robotics

INDEX

This Plugin was born with the idea of communicating **quickly and safely with Flexivision** and **FANUC** robots**.** The Plugin requires the **"Fanuc User Socket Messaging"** license for correct operation.

FlexiBowl® Plug-In

FANUC

Plug-In Configuration

Step 1.

Overview
In order to use Socket Messaging, you need to configure the following network hardware and software parameters, we need configure a Server Tag.

Setting up a Client Tag

Steps
1. Could start the controller.
   a. **On the teach pendant**, press and hold the **SHIFT** and **RESET** keys. Or, on the operator panel, press and hold **RESET**.
   b. While still pressing **SHIFT** and **RESET** on the **teach pendant** (or **RESET** on the operator panel), **turn on the power** disconnect circuit breaker.
   c. **Release all** of the keys.
2. On the teach pendant, press **MENU**.
3. Select **SETUP**.
4. Press **F1, [TYPE].**
5. Select **Host Comm**.
6. Press **F4, [SHOW].**
7. Select **Servers**.
8. Move the cursor to the tag **'S3'** and press **F3**, **DETAIL**. You will see screen similar to the following.

```
SETUP Tags

  Tag S3:

  Comment:              ****************
  Protocol Name:        ********
  Current State:        UNDEFINED
  Startup State:
  Server IP/Hostname:*****************
  Remote Path/Share: *****************
  Port:                 *****
  Inactivity Timeout:   15 min
  Username:                      anonymous
  Password                      **********
```

9. Move the cursor to the **Protocol Name** item, and press **F4, [CHOICE]**.
10. Select **SM**.
11. Move the cursor to **Startup State**, press **F4, [CHOICE].**
12. Select **START**

Plug-In Configuration

13. Press **F2 [ACTION].**
14. Select **DEFINE**.
15. Press **F2 [ACTION].**
16. Select **START**.

17. Set the system variable:

a. Press **MENU**.
b. Select **NEXT**.
c. Select **SYSTEM**, and press **F1, [TYPE].**
d. Select **Variables**.
e. Move the cursor to **$HOSTC_CFG**, and press **ENTER**.
f . Move the cursor to the structure corresponding to the tag '**S3**', namely move the cursor to structure element **[3]**.

```
SYSTEM Variables
$HOSTS_CFG
   1   [1]           HOST_CFG_T
   2   [2]           HOST_CFG_T
   3   [3]           HOST_CFG_T
   4   [4]           HOST_CFG_T
   5   [5]           HOST_CFG_T
   6   [6]           HOST_CFG_T
   7   [7]           HOST_CFG_T
   8   [8]           HOST_CFG_T
```

g. Press **ENTER**. You will see screen similar to the following.

```
SYSTEM Variables
$HOSTS_CFG[3]
  1 $COMMENT         *uninit*
  2 $PROTOCOL        'SM'
  3 $PORT            *uninit*
  4 $OPER            3
  5 $STATE           3
  6 $MODE            *uninit*
  7 $REMOTE          *uninit*
  8 $REPERRS         FALSE
  9 $TIMEOUT         15
 10 $PATH            *uninit*
 11 $STRT_PATH       *uninit*
 12 $STRT_REMOTE     *uninit*
 13 $USERNAME        *uninit*
 14 $PWRD_TIMOUT     0
 15 $SERVER_PORT     0
```

h. Move the cursor on **$SERVER_PORT**. Insert the port '**4001**'.

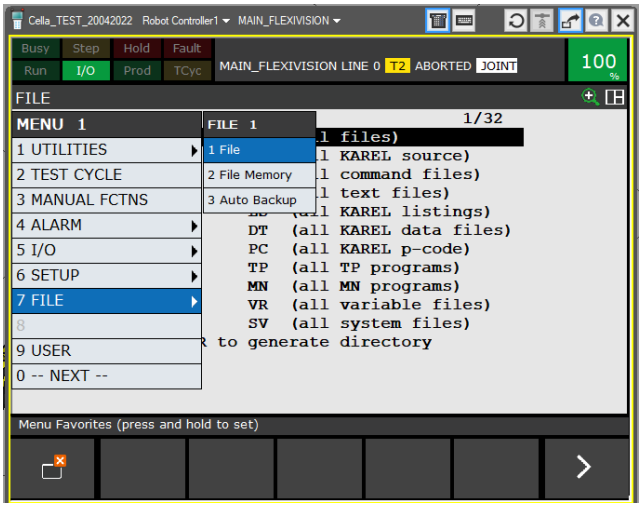
Plug-In Configuration

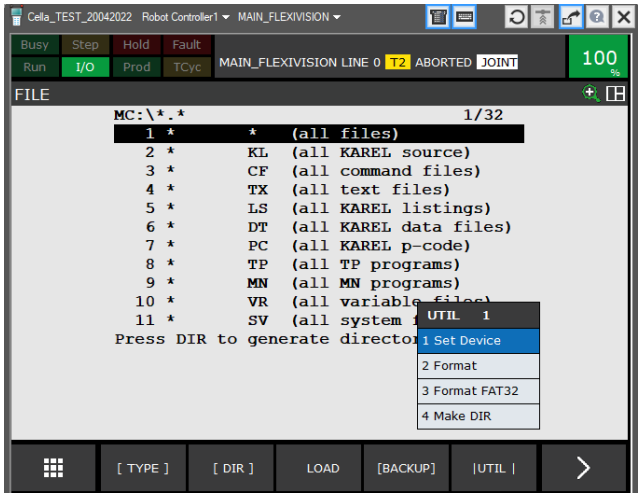
Step 2.

LOAD THE FILE “FLEXIVISION.PC” AND “MAIN_FLEXIVISION.TP”

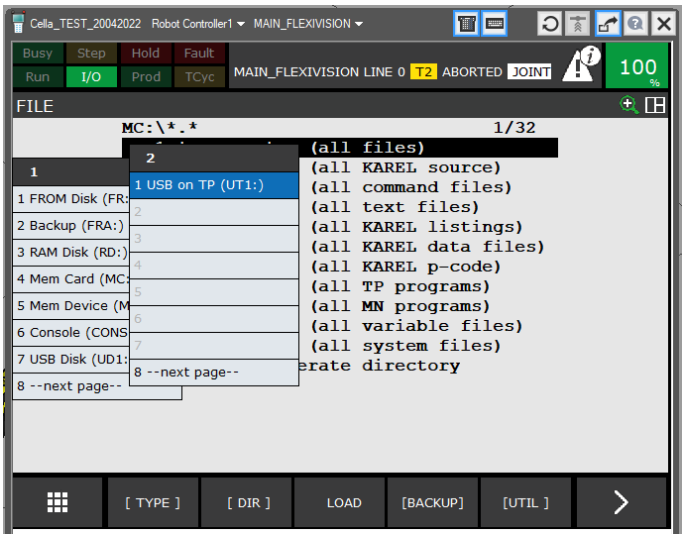
The file "**Flexivision.pc**" is the file that allows communication with Flexivision while the file "**Main_Flexivision.tp**" is an example that illustrates the correct use of the communication file.

Once the server is configured, you must place these files on the controller. To do this, copy the files to a USB stick and insert it into the USB port of the pendant.





Plug-In Configuration



Now press "**Load**" and import the file **Flexivision.pc** and **Main_Flexivision.tp**.

Step 3.

DESCRIPTION OF THE FUNCTIONALITY FOR THE FILE "FLEXIVISION.PC" AND "MAIN_FLEXIVISION.TP"

Elements of operation for the program:

- **Flag 1**
  The flag that uses **Flexivision.pc** program for connection with the main **.tp** program is the number 1.

This flag will be set to **ON** by the **.pc** program once the connection has been made. Subsequently, every time the flag will be returned to **OFF** by the main program there will be the sending of the command.
Received the response from Flexivision and updated the register, the **program.pc** will set the flag to **ON**.

In the **program.tp** the flag is set to **OFF**,then the Flexivision routine starts and we have to wait for the connection to occur.

Plug-In Configuration

File .tp

```
F[5]=(OFF)


!I deny FlexiVision to send
!the commands
F[1]=(OFF)

!I run the Karel program
RUN FLEXIVISION
LBL[5]

IF (F[5]=ON) THEN
WAIT ((F[1]=ON))
F[5]=(OFF)
ENDIF
!wait the flexivision connection
WAIT (F[1]=ON)
```

This part allows you to manage the reconnection

Flag 5 is used only to manage reconnection.
You will move past the **WAIT (F[1]=ON)** only after the connection with Flexivision has taken place.

- **Flexivision_command**
  The Flexivision_command register is the string register number 14 and hosts the command to be sent to Flexivision. From this register the **Flexivision.pc** file will read the command to send it to Flexivision.

Then, the command is written to the desired register and the **Flag** is set to **OFF**, to access the data you wait for the Flag to return to **ON**.

File .tp

```
!I set the command
CALL SET_FLEXIVISION_COMMAND(
'start_Locator')
!I allow FlexiVision to send
!the commands
F[1]=(OFF)
LBL[1:main cycle]
!I wait the Karel program
!to deny FlexiVision
!to send the commands
WAIT (F[1]=ON)
```

Plug-In Configuration

- **Id_pattern**
  The **Id_pattern** register is the numerical register number 70. On this register is written:
  - **-200**: Flexivision not connected
  - **-100**: Flexivision in error
  - **-2**: Command not known
  - **-1**: Command not for the pattern search
  - **0**: Command of search pattern and pattern not found
  - **Number>0**:Command of search pattern and '**Id**' of the pattern

File .tp

```
!I wait the Karel program
!to deny FlexiVision
!to send the commands
WAIT (F[1]=ON OR
R[70:id_pattern]=(-200))
!Reconnection management
IF (R[70:id_pattern]=(-200))
THEN
F[5]=(ON)
JMP LBL[5]
ELSE
```

This part is used to manage the reconnection

Plug-In Configuration

```
!If i receive this value
!I abort all the tasks
!and I return an error
!"FlexiVision error"
IF (R[70:id_pattern]=(-100))
THEN
UALM[1]
ELSE
!If i receive this value
!I abort all the tasks
!and I return an error
!"Undefined command"
IF (R[70:id_pattern]=(-2)) THEN
UALM[2]
ELSE

!If i receive this value
!means that there are
!no found parts so I have
!to send a rotation command to
!the FlexiBowl

IF (R[70:id_pattern]=0) THEN
CALL FLB_PLUGIN('QX3')

!I allow FlexiVision to send
!the commands
F[1]=(OFF)
ELSE

!If i receive a valid ID
!I start the main cycle
IF (R[70:id_pattern]<>(-1)) THEN
CALL PROG_1
F[1]=(OFF)
```

Simple pick and place program

We read the value of the register **id_pattern** to know the actions to be carried out.

The following is the function **SET_FLEXIVISION_COMMAND()** that allows you to write the command to the "**Flexivision_Command**" register.

```
1:  !Write the function argument
2:  !on the string register
3:  SR[14]=AR[1]
```

The movement of the robot in the **.tp** file takes place with **tool 5** and with **frame 7**. **Frame 7** must coincide with the calbration frame used for Flexivision.

Comments about the file

```
!I set the user tool
!and the user frame
UTOOL_NUM=5
UFRAME_NUM=7
```

Flexivision returns the **x,y,r coordinates**. Update the **z** and **w** values according to the frame used for calibration and your needs.

```
!I write on the position register
!the Z height of the part
PR[60,3:Flexivision_Posi]=R[71]


!I set to zero the rotation of
!the W axis
PR[60,4:Flexivision_Posi]=0
```

In the **.tp** file the first two commands are "**set_Recipe**" and "**get_Recipe**". Create a "**Recipe_Name**" recipe and enter the same name "**Recipe_Name**" in the register **number 12** that is used by the program for the control.

If you do not need these "**commands**", it can be removed with the respective ON/OFF flag.

```
CALL SET_FLEXIVISION_COMMAND(
'set_Recipe=Fanuc_flexivision')

!I allow FlexiVision to send
!the commands
F[1]=(OFF)

!I wait the Karel program
!to deny FlexiVision
!to send the commands
WAIT (F[1]=ON)
```

- **Flexivision_Response**
  The **Flexivision_Response** register is the string register number 15. On this register it is 'written' the string returned by Flexivision when the command is not of search pattern.

- **Flexivision_Position**
  The **Flexivision_position** register is the position **register 60**. On this register it is 'written' the position returned by Flexivision, when the pattern is found.
  You will have to access this register only when the "**id_pattern**" register has a **value >0**.

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.