

# Manual



# FlexiVision

FANUC PLUG-IN

**ars**

Feeding  
Industrial  
Robotics

# INDICE

1. **Configurazione Plug-In**
2. **Commenti sui file .pt**
3. **Lista comandi FlexiVision**

Questo Plugin è nato con l'idea di comunicare in maniera **rapida e sicura con Flexivision** tramite i robot **FANUC**.

Il Plugin necessita della licenza "**Fanuc User Socket Messaging**" per il corretto funzionamento.

**FlexiBowl<sup>®</sup> Plug-In**

**FANUC**

# Configurazione Plug-In

## Step 1.

### Panoramica

Al fine di usare Socket Messaging, è necessario configurare i parametri hardware and software del network, per la comunicazione con Flexivision dovrà essere configurato un Server Tag.

### Impostazione di un Server Tag

#### Passaggi

1. Cold start del controller.
  - a. Sulla **teach pendant**, premere a lungo I tasti **SHIFT** e **RESET**. Oppure, sul pannello operatore, premere a lungo **RESET**.
  - b. Mentre state premendo **SHIFT** e **RESET** sulla **teach pendant** (o **RESET** sul pannello operatore), **accendere l'alimentazione**.
  - c. **Rilasciare tutti i tasti**.
2. Sulla **teach pendant**, premere **MENU**.
3. Selezionare **SETUP**.
4. Premere **F1, [TYPE]**.
5. Selezionare **Host Comm**.
6. Premere **F4, [SHOW]**.
7. Scegliere **Servers**.
8. Muovere il cursore sul **tag 'S3'** e premere **F3, DETAIL**. Vedrete una schermata simile alla seguente.

```
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. Muovere il cursore su **Protocol Name** item, e premere **F4, [CHOICE]**.
10. Selezionare **SM**.
11. Muovere il cursore su **Startup State**, premere **F4, [CHOICE]**.
12. Selezionare **START**

# Configurazione Plug-In

13. Premere **F2 [ACTION]**.
14. Selezionare **DEFINE**.
15. Premere **F2 [ACTION]**.
16. Selezionare **START**.

## 17. Impostare le variabili di sistema:

- a. Premere **MENUS**.
- b. Selezionare **NEXT**.
- c. Selezionare **SYSTEM**, e premere **F1, [TYPE]**.
- d. Selezionare **Variables**.
- e. Muovere il cursore su **\$HOSTC\_CFG**, e premere **ENTER**.
- f. Muovere il cursore sulla struttura corrispondente al tag '**S3**', cioè muovere il cursore su cursore sull'elemento **[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. Premere **ENTER**. Vedrete una schermata simile alla seguente.

```
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_TIMEOUT 0
15 $SERVER_PORT  0
```

- h. Muovere il cursore su **\$SERVER\_PORT**. Inserire come porta la '**4001**'.

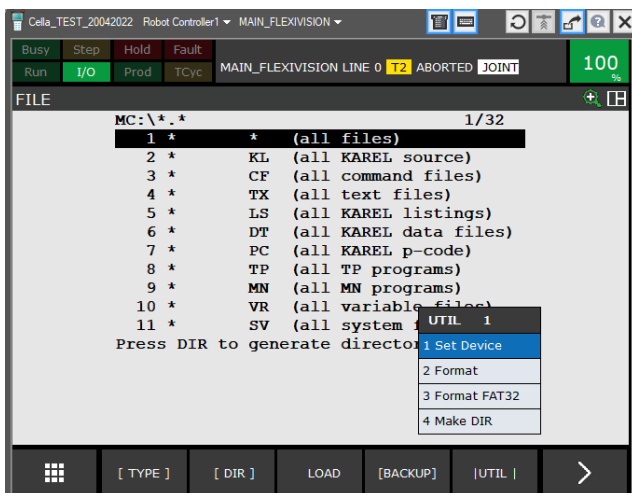
# Configurazione Plug-In

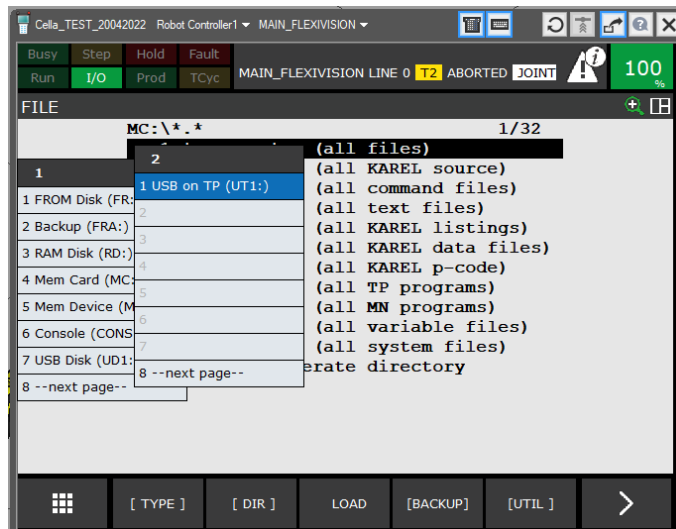
## Step 2.

### CARICAMENTO DEI FILE “FLEXIVISION.PC” E “MAIN\_FLEXIVISION.TP”

Il file “**Flexivision.pc**” è il file che permette la comunicazione con Flexivision mentre il file “**Main\_Flexivision.tp**” è un esempio che illustra il corretto utilizzo del file di comunicazione.

Una volta che il client è configurato, si devono inserire tali file nel controller. Per fare ciò, copiare i file su una chiavetta USB e inserirla nella porta USB della pendant.





Now press “**Load**” and import the file **Flexivision.pc** and **Main\_Flexivision.tp**

### Step 3.

## DESCRIZIONE DEL FUNZIONAMENTO DEL FILE “FLEXIVISION.PC” E “MAIN\_FLEXIVISION.TP”

Elementi di funzionamento del programma:

- **Flag 1**

La flag che utilizza di programma **Flexivision.pc** per il raccordo con il programma principale **.tp** è la numero 1.

Tale flag sarà portata a **ON** dal programma **.pc** una volta avvenuta la connessione. Successivamente ogni volta che la flag sarà riportata a **OFF** dal programma principale ci sarà l’invio del comando. Avvenuta la ricezione e aggiornati i registri (descritti successivamente) la flag sarà riportata a **ON**.

Nel **programma.tp** viene portata la flag a **OFF**, si avvia la routine di Flexivision e attendiamo che la connessione sia avvenuta.

# Configurazione Plug-In

## 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

La flag 5 è utilizzata solo per gestire la riconnessione. Si proseguirà oltre la **WAIT (F[1]=ON)** solo dopo che è avvenuta la connessione con Flexivision.

- **Flexivision\_command**  
Il registro Flexivision\_command è il registro stringa numero 14 ed ospita il comando da inviare a Flexivision. Da tale registro il file **Flexivision.pc** leggerà il comando per inviarlo a Flexivision.

Viene quindi scritto il comando sul registro desiderato e poi viene portata la **Flag** a **OFF**, per accedere ai dati si attende che la Flag ritorni ad **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)
```

# Configurazione Plug-In

- **Id\_pattern**

Il registro **Id\_pattern** è il registro numerico numero 70. Su tale registro viene scritto:

- **-200**: Flexivision non connesso
- **-100**: Flexivision in errore
- **-2**: Comando non conosciuto
- **-1**: Comando non di ricerca pattern
- **0**: Comando di ricerca pattern e pattern non trovato
- **numero>0**: Comando di ricerca pattern e id del pattern trovato

## 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



# Configurazione Plug-In

```

!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]=(DEF)
ELSE

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

```

Simple pick and place program

Leggiamo quindi il valore del registro **id\_pattern** per sapere le azioni da svolgere.

Di seguito viene riportato la funzione **SET\_FLEXIVISION\_COMMAND()** che permette di scrivere il comando all'interno del registro "**Flexivision\_Command**".

```

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

```

Il movimento del robot nel file **.tp** avviene con il **tool 5** e con il **frame 7**.

Il frame 7 dovrà coincidere con il frame di calibrazione utilizzato per Flexivision

# Commenti sui file .tp

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

Flexivision restituisce le **coordinate x,y,r**. Aggiornare i valori di **z** e **w** in base al frame utilizzato per la calibrazione e alle proprie esigenze.

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

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

Nel file **.tp** i primi due comandi sono di **"set\_Recipe"** e di **"get\_Recipe"**.

Creare quindi una ricetta **"Nome\_ricetta"** e inserire lo stesso nome **"Nome\_ricetta"** nel **registro 12** che è utilizzato dal programma per il controllo.

Qualora non si abbia bisogno di tali **"comandi"** alla loro eliminazione con le rispettive flag di raccordo ON/OFF.

```
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**

Il registro **Flexivision\_Response** è il registro stringa numero 15. Su tale registro viene 'scritta' la stringa restituita da Flexivision, quando il comando non è di ricerca pattern.

- **Flexivision\_Position**

Il registro **Flexivision\_position** è il registro di posizione numero 60. Su tale registro viene 'scritta' la posizione restituita da Flexivision, quando viene trovato il pattern.

Si dovrà accedere a tale registro solo nel momento in cui il registro **"id\_pattern"** ha un valore **>0**.

# Lista Comandi FlexiVision

Per inviare il comando a FlexiVision è necessario modificare il valore della stringa "command".

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.