

Manual



FlexiVision

DENSO PLUG-IN

INDICE

1. Installazione Plug-In
2. Lista Comandi FlexiVision
3. Script

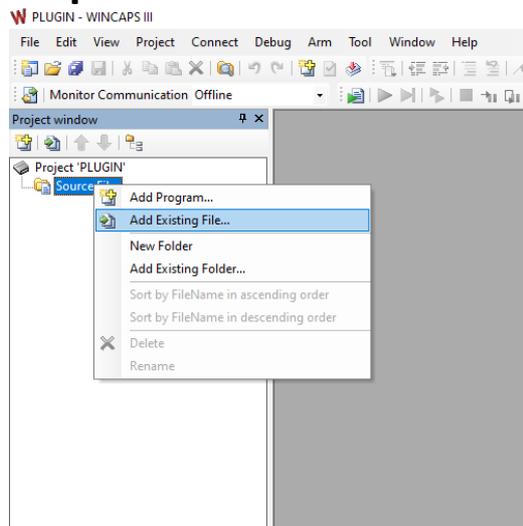
Questo Plugin è nato con l'idea di comunicare in **maniera rapida e sicura** con il software di visione FlexiVision 2.0 tramite i robot **DENSO**, mediante l'utilizzo di istruzioni in linguaggio **PacScript**.
Il Plugin NON necessita di una licenza aggiuntiva per la gestione dei socket.

FlexiBowl[®] Plug-In

DENSO

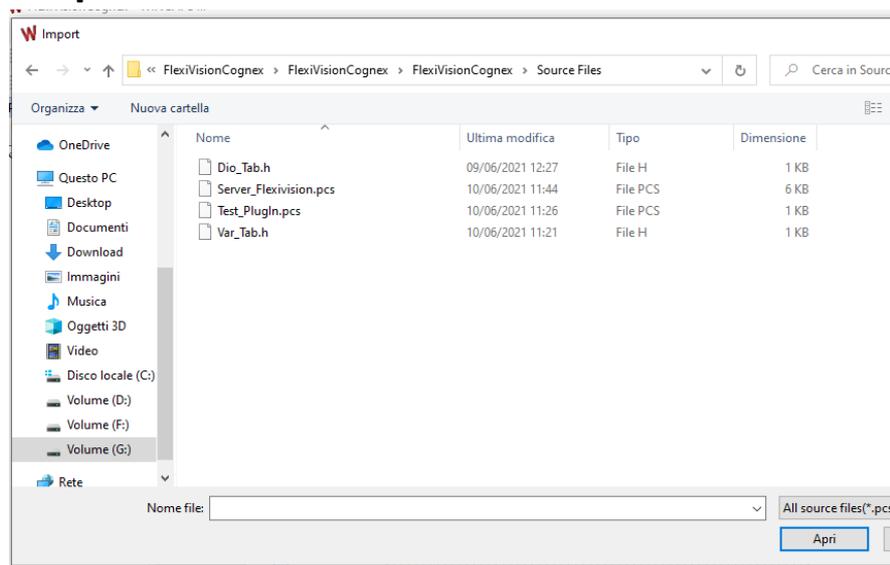
Installazione Plug-In

Step 1.



Aprire il software **DENSO Wincaps 3**, fare click con il tasto destro su **"Source Files"** e successivamente selezionare **"Add Existing File.."**.

Step 2.

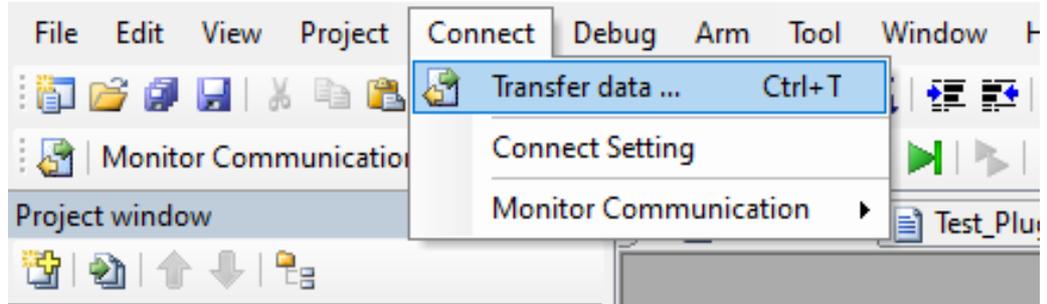


Selezionare i 4 file inviati da ARS, premere il pulsante **"Apri"** così da caricarli all'interno del progetto.

Installazione Plug-In

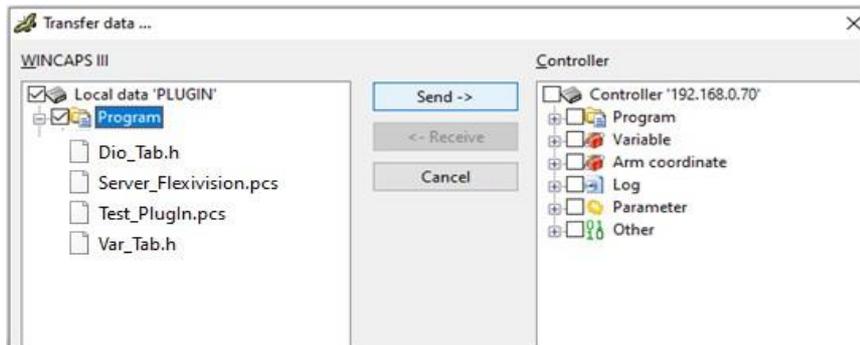
Step 3.

W PLUGIN - WINCAPS III



Dal programma WINCAPS selezionare "**Connect**" e successivamente "**Transfer Data**".

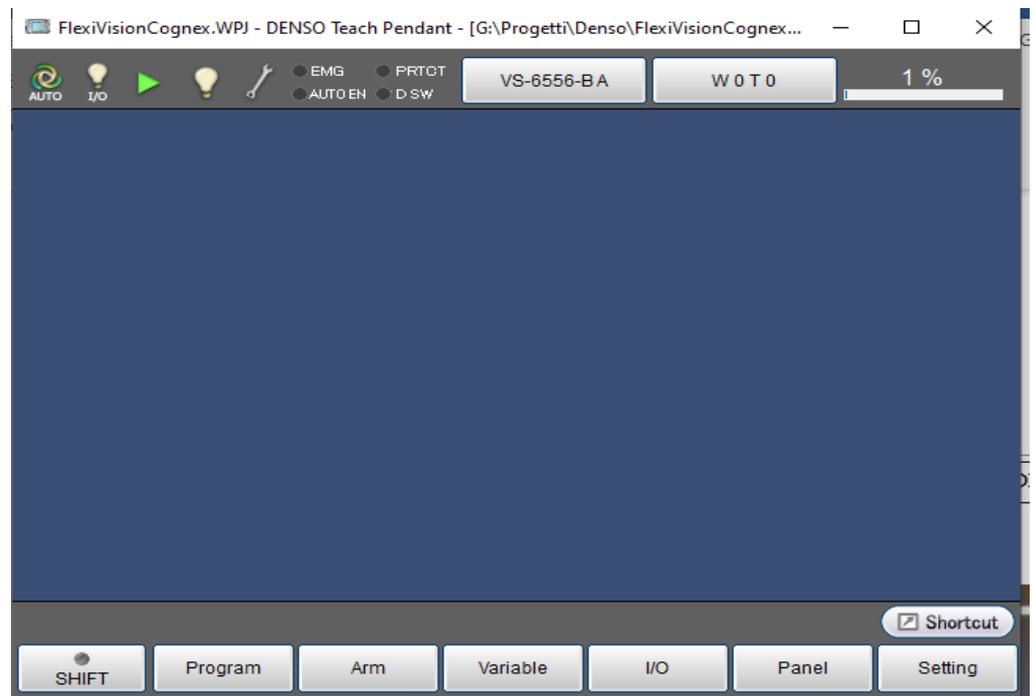
Step 4.



Trasferire quindi i programmi nella memoria del controller **DENSO**.

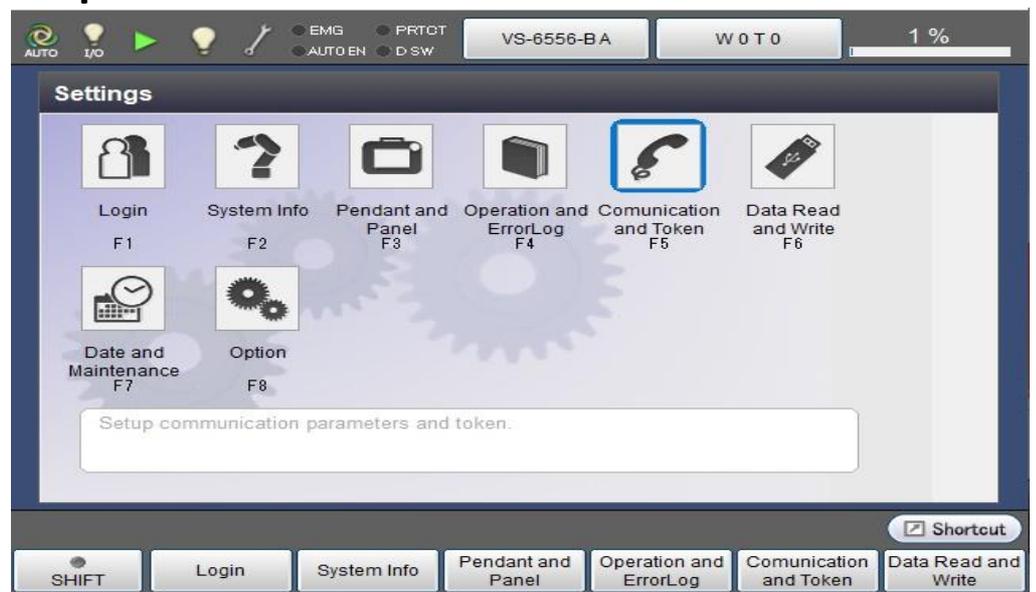
Installazione Plug-In

Step 5.



Utilizzando la **TeachPendant** del robot, dal **menu principale**, premere il pulsante **"Setting"**.

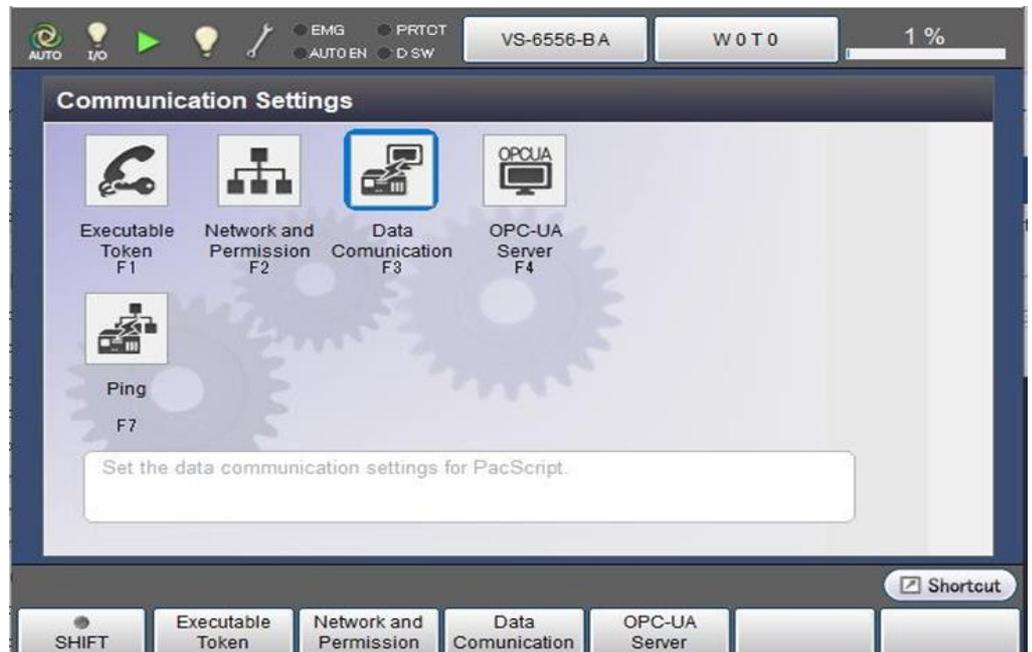
Step 6.



Selezionare **"Communication and Token F5"**.

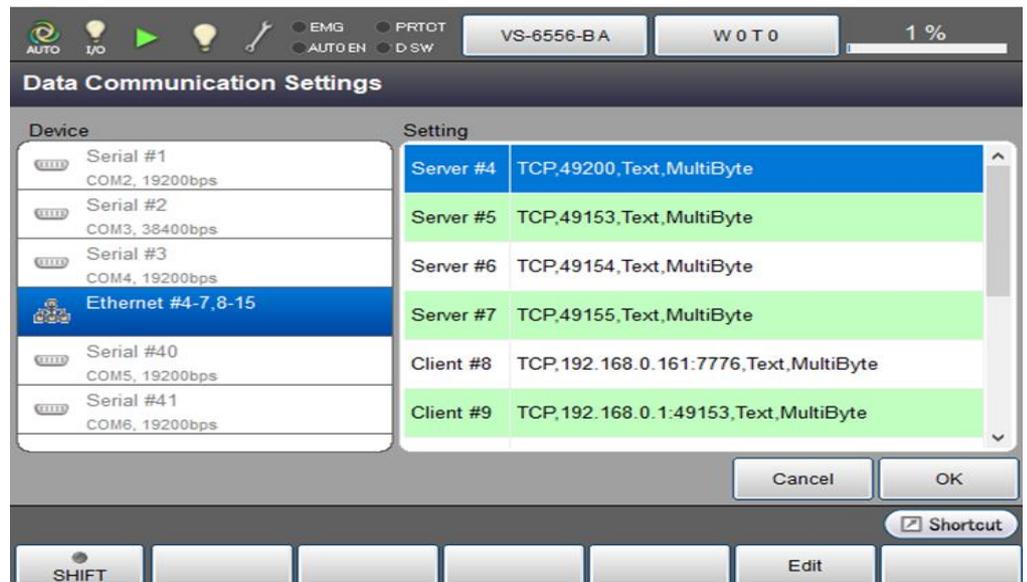
Installazione Plug-In

Step 7.



Selezionare **"Data Communication F3"**.

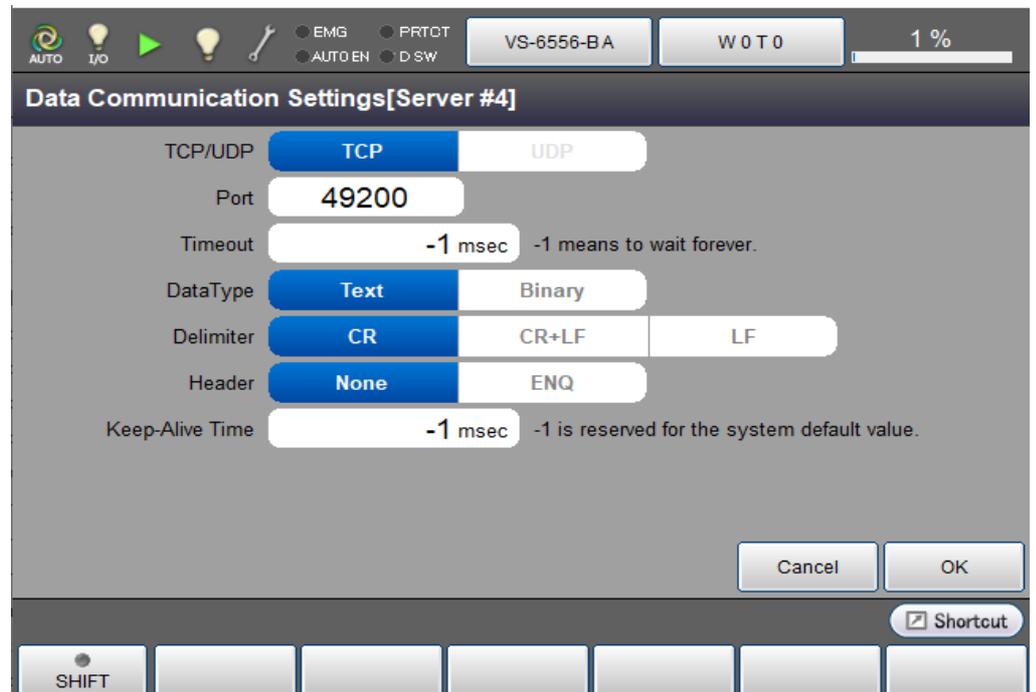
Step 8.



Menù di sinistra selezionare la voce **"Ethernet #4-7,8-15"**, poi selezionare **"Server#4"** e premere il pulsante **"Edit"** in basso a destra.

Installazione Plug-In

Step 9.



Impostare la " **porta di comunicazione**" e copiare i parametri mostrati nell'immagine. Premere "**OK**" per tornare indietro e salvare.

Installazione Plug-In

Step 10.

```
'Lancio il server in un task parallelo  
'Run Server in a parallel task  
run Server_Flexivision
```

Sarà importante far partire il Server di comunicazione **"Server_Flexivision"** in un task parallelo così da poter comunicare con il sistema di visione anche durante l'esecuzione del programma Pick&Place del robot.

Le variabili globali utilizzate dal programma **"Server_Flexivision"** per la gestione della comunicazione sono le seguenti:

- **S[S2_ComandoTx]** → Dovrà contenere la stringa che si desidera inviare al sistema di visione.
- **IO128 (IO[BoolSendCommand])** → Impostando il bit 128 a ON (**set I 0128**) la stringa contenuta nella variabile **S[S2_ComandoTx]** verrà effettivamente inviata. Dopo aver inviato il comando e ricevuto la risposta il bit128 verrà resettato e tornerà allo stato OFF.
- **S[S0_StringaRx]** → Questa stringa conterrà la risposta ricevuta da FlexiVision.
- **P[P0_PickPosition]** → A seguito di un comando di localizzazione (**"start_Locator"/"turn_Locator"** ecc.), le coordinate X – Y – RZ di questa posizione verranno aggiornate con le coordinate del componente individuato.
- **S[S3_ModelName]** →] → A seguito di un comando di localizzazione (**"start_Locator"/"turn_Locator".**) tale stringa conterrà il nome del Pattern.

Il programma **"Test_Plugin.pcs"** è presente nei file forniti solo per capire/verificare il funzionamento del **"Server_Flexivision"** prima di sviluppare l'applicazione definitiva.

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. 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.

Script

Dio_Tab.

```
!TITLE "Denso robot program"  
#define <constance> <Strings>  
  
#define BoolSendCommand 128
```

Var_Tab.

```
!TITLE "Denso robot program"  
#define <constance> <Strings>  
  
'===== Type P Variables =====  
#define P0_PickPosition '===== 0 ' X/Y/R Presa Da Flexibowl  
Type S Variables =====  
#define S0_StringaRx 0 '  
#define S2_ComandoTx 2 '  
#define S3_ModelName 3 '
```

Script

```
'!TITLE "Denso robot program"  
#Include "Server_Flexivision.pcs"  
#Include "Dio_Tab.h"  
#Include "Var_Tab.h"
```

Sub Main

```
'Dichiarazione variabili locali  
'Def local variables  
TakeArm Keep = 0  
dim ReturnStr As String  
dim SendCmd as integer  
dim StringCmd as string  
dim ReceivedStr as string  
dim model as string
```

```
'Lancio il server in un task parallelo  
'Run Server in a parallel task  
run Server_Flexivision
```

*LBL_Start:

```
'Attendo di aver completato la scrittura nella stringa S[S2_ComandoTx]  
'Prima di impostare il valore 1 alla variabile  
'I'm waiting to finish writing to string S[S2_ComandoTx]  
'Before setting the value 1 on the variable  
wait (SendCmd==1)
```

```
'Assegno lo stesso valore alla variabile globale,  
'questo è utile in fase di test poichè si riesce a modificare il valore della stringa "StringCmd"  
anche mentre il programma è in esecuzione  
'I assign the same value to the global variable,  
'this is useful in testing as it is possible to change the value of the string "StringCmd"  
even while the program is running  
S[S2_ComandoTx]=StringCmd
```

```
'Per far inviare il comando al Server imposto il bit 128 ON, questo bit è SW non fisico  
'To send the command to the Server set bit 128 to ON, this bit is non-physical SW  
set IO128
```

```
'Attendo che il server comunichi di aver completato l'operazione richiesta  
impostando a OFF il bit 128  
'I wait for the server to communicate that it has completed the requested operation  
by setting bit 128 to OFF  
Wait IO[128] = OFF
```

```
'Leggo la stringa ricevuta  
'Read the received string  
ReceivedStr=S[S0_StringaRx]  
model=S[S3_ModelName]
```

```
'Resetto la variabile "SendCmd" così da non inviare nuovamente il comando  
'I reset the "SendCmd" variable to avoid sending the same command again  
SendCmd=0
```

```
goto *LBL_Start
```

End Sub

Script

```

!TITLE "Gestione comunicazione Flexibowl"
#include "Dio_Tab.h"
#include "Var_Tab.h"

```

Sub Main

```

'Dichiarazione variabili locali
'Def local variables
Dim tCommOpen As Integer
Dim stringaAppoggio As Variant

```

```

S[S0_StringaRx] = "" 'Stringa ricevuta ; Received String
S[S2_ComandoTx] = "" 'Stringa da inviare; String to send
P[P0_PickPosition] = P(0, 0, 0, 0, 0, 0)
'Turning OFF the internal I/O 128 (trigger to send string)
Reset IO[BoolSendCommand]

```

```

Comm.Close - 1
tCommOpen = Timer
Delay 3000
ClrErr
On Error GOTO Errore

```

*Inizio:

```

Select Case Comm.State(4)
Case 0 'A line is not opened by any task'
    If Timer > (tCommOpen + 2000) Then
        'Al primo giro del programma apro il server
        'Open Server
        Comm.Open 4
        tCommOpen = Timer
        Comm.Clear 4
        S[S0_StringaRx] = ""
        S[S2_ComandoTx] = ""
    End If

Case 1
    'A line is opened by either task.
    'Porta già aperta ma da un'altro task, chiudo la connessione
    'Comm.Close - 1
    Delay 10

Case 2
    'The status that the line is established and communication can be made.
    'Connessione stabilita

    'Attendo che il bit interno n128 sia ON così da poter inviare la stringa
    'Waiting bit128=ON
    Wait IO[BoolSendCommand] = ON

    *****[set_Recipe]*****
    if (InStr(1,S[S2_ComandoTx] ,"set_Recipe")>0) then
        'invio il comando/'send command
        Comm.Output 4, S[S2_ComandoTx], 500
        'nessuna risposta per questo comando da parte
        'di Flexivision
        'No answer for this command from FlexiVision
        S[S0_StringaRx] = "OK"
        Reset IO[BoolSendCommand]
    end if
    *****

```

Script

```

*****[get_Recipe]*****
if (InStr(1,S[S2_ComandoTx],"get_Recipe")>0) then
  'invio il comando/'send command
  Comm.Output 4, S[S2_ComandoTx], 500
  'Attendo la risposta da parte di Flexivision
  'Waiting answer from Flexivision
  S[S0_StringaRx] = Comm.Input(4, -1)
  'Stringa ricevuta
  'String received
  Reset IO[BoolSendCommand]
end if
*****

*****[state_Locator]*****
if (InStr(1,S[S2_ComandoTx],"state_Locator")>0) then
  'invio il comando/'send command
  Comm.Output 4, S[S2_ComandoTx], 500
  'Attendo la risposta da parte di Flexivision
  'Waiting answer from Flexivision
  S[S0_StringaRx] = Comm.Input(4, -1)
  'Stringa ricevuta
  'String received
  Reset IO[BoolSendCommand]
end if
*****

*****[start_Empty]*****
if (InStr(1,S[S2_ComandoTx],"start_Empty")>0) then
  'invio il comando/'send command
  Comm.Output 4, S[S2_ComandoTx], 500
  'Attendo la risposta da parte di Flexivision
  'Waiting answer from Flexivision
  S[S0_StringaRx] = Comm.Input(4, -1)
  'Stringa ricevuta
  'String received
  Reset IO[BoolSendCommand]
end if
*****

*****[stop_Locator]*****
if (InStr(1,S[S2_ComandoTx],"stop_Locator")>0) then
  'invio il comando/'send command
  Comm.Output 4, S[S2_ComandoTx], 500
  'Nessuna risposta per questo comando da parte
  'di Flexivision
  'No answer for this command from FlexiVision
  S[S0_StringaRx] = "OK"
  Reset IO[BoolSendCommand]
end if
*****

*****[test_Locator]*****
if (InStr(1,S[S2_ComandoTx],"test_Locator")>0) then
  'invio il comando/'send command
  Comm.Output 4, S[S2_ComandoTx], 500
  'Attendo la risposta da parte di Flexivision
  'Waiting answer from Flexivision
  S[S0_StringaRx] = Comm.Input(4, -1)
  'Stringa ricevuta
  'String received
  Reset IO[BoolSendCommand]
end if
*****

```

Script

```

*****[start_Control]*****
if (InStr(1,S[S2_ComandoTx], "start_Control")>0) then
    'invio il comando/'send command
    Comm.Output 4, S[S2_ComandoTx], 500
    'Attendo la risposta da parte di Flexivision
    'Waiting answer from Flexivision
    S[S0_StringaRx] = Comm.Input(4, -1)
    'Stringa ricevuta
    'String received
    Reset IO[BoolSendCommand]
end if
*****

*****[start_Control]*****
if ((InStr(1,S[S2_ComandoTx], "start_Locator")>0) OR
(InStr(1,S[S2_ComandoTx], "turn_Locator")>0))then
    'invio il comando/'send command
    Comm.Output 4, S[S2_ComandoTx], 500
    'Attendo la risposta da parte di Flexivision
    'Waiting answer from Flexivision
    S[S0_StringaRx] = Comm.Input(4, -1)

    'Controllo se viene richiesta l'attivazione della tramoggia
    'Check if the activation of the hopper is requested
    if (InStr(1,S[S0_StringaRx], "Hopper")>0) then
        'Attivazione tramoggia
        *****
        'Hopper signal
    else
        'verifico se ci sono stati errori nel Locator

        if (InStr(1,S[S0_StringaRx], "#")>0) then
            'Errore Locator
            'Error occured
            else
                'è stata ricevuta una posizione la divido
                'Position Received
                stringaAppoggio = Split(S[S0_StringaRx], ";")
                S[S3_ModelName]=stringaAppoggio(0)
                LetX P[P0_PickPosition] = Val(stringaAppoggio(1))
                LetY P[P0_PickPosition] = Val(stringaAppoggio(2))
                LetRZ P[P0_PickPosition] = Val(stringaAppoggio(3))
            end if
        end if
        'Stringa ricevuta
        'String Received
        Reset IO[BoolSendCommand]
    end if
*****

```

```

End Select
Delay 10
GOTO Inizio
*Errore:
Resume Inizio
End Sub

```