Manual

FlexiVision

DENSO PLUG-IN

ars | Feeding Industrial Robotics

INDEX

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

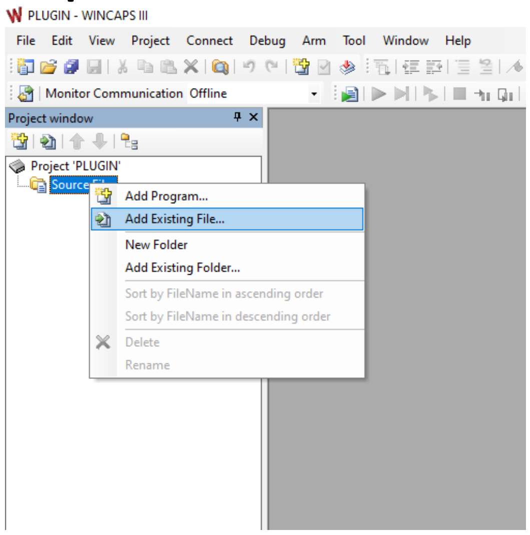
This Plugin was developed with the idea of communicating with the **FlexiVision 2.0** vision software **in a fast and safe way** through the **DENSO** robots**,** through the use of instructions in **PacScript** language**.**
The Plugin does NOT require an additional license to manage the sockets.

FlexiBowl® Plug-In

DENSO

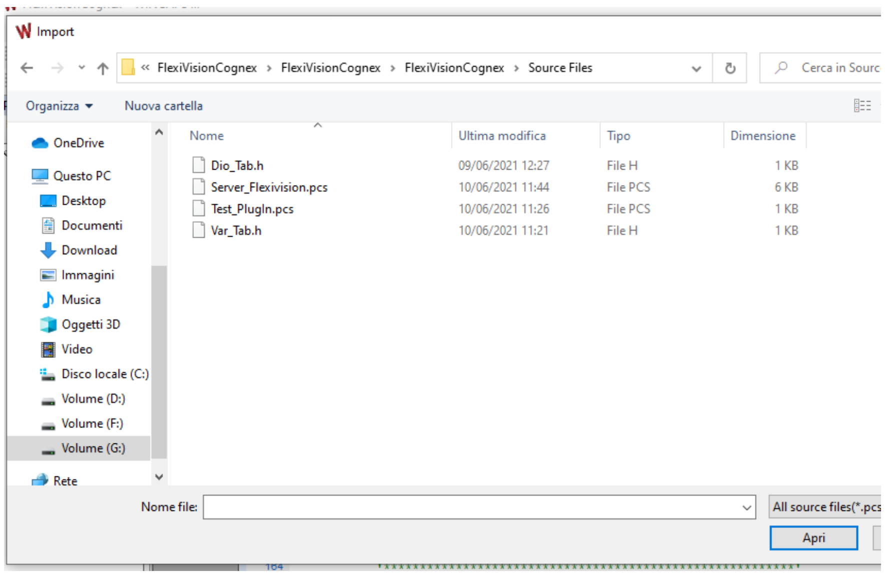
Plug-In Installation

Step 1.


PLUGIN - WINCAPS III
File Edit View Project Connect Debug Arm Tool Window Help
Monitor Communication Offline
Project window
Project 'PLUGIN'
Source Files
Add Program...
Add Existing File...
New Folder
Add Existing Folder...
Sort by FileName in ascending order
Sort by FileName in descending order
Delete
Rename

Open the DENSO Wincaps 3 software, right click on **“Source Files”** and then select **“Add Existing File..”.**

Step 2.

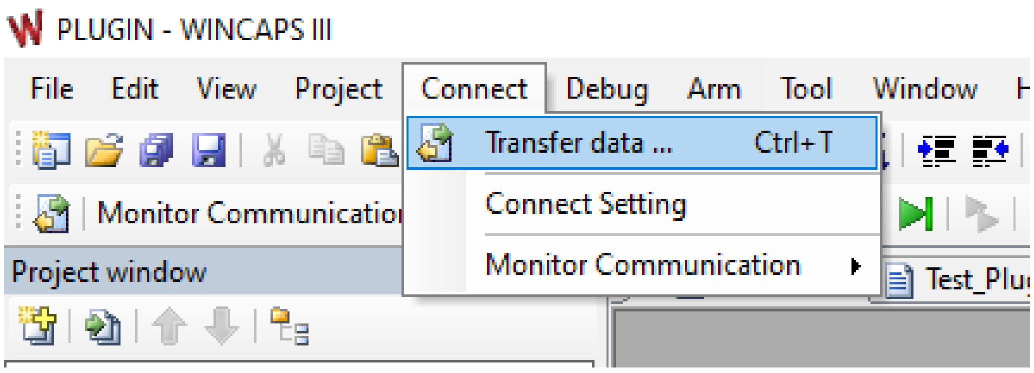


Nome	Ultima modifica	Tipo	Dimensione
Dio_Tab.h	09/06/2021 12:27	File H	1 KB
Server_Flexivision.pcs	10/06/2021 11:44	File PCS	6 KB
Test_PlugIn.pcs	10/06/2021 11:26	File PCS	1 KB
Var_Tab.h	10/06/2021 11:21	File H	1 KB

Select the 4 files sent by ARS, press the “Open” button to upload them into the project.

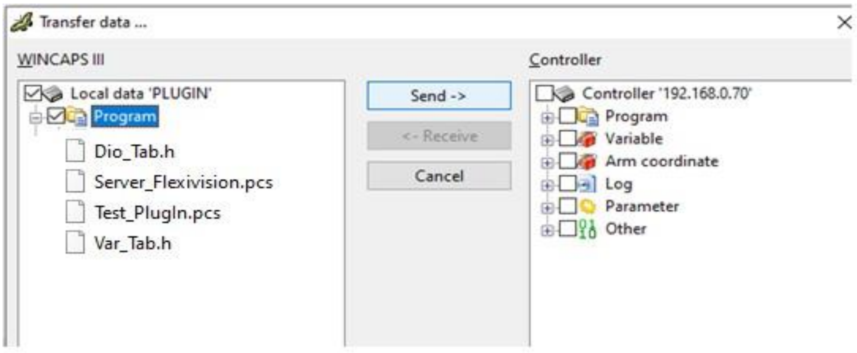
Plug-In Installation

Step 3.



Select “Connect” from the WINCAPS program and then “Transfer Data”.

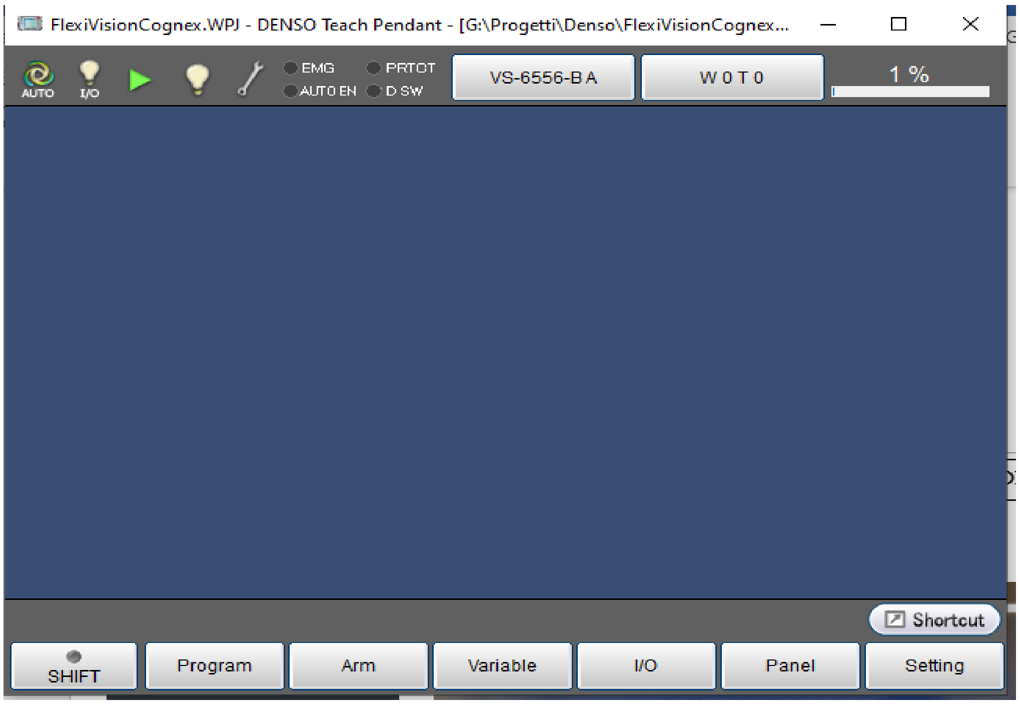
Step 4.



Then transfer the programs to the DENSO controller memory.

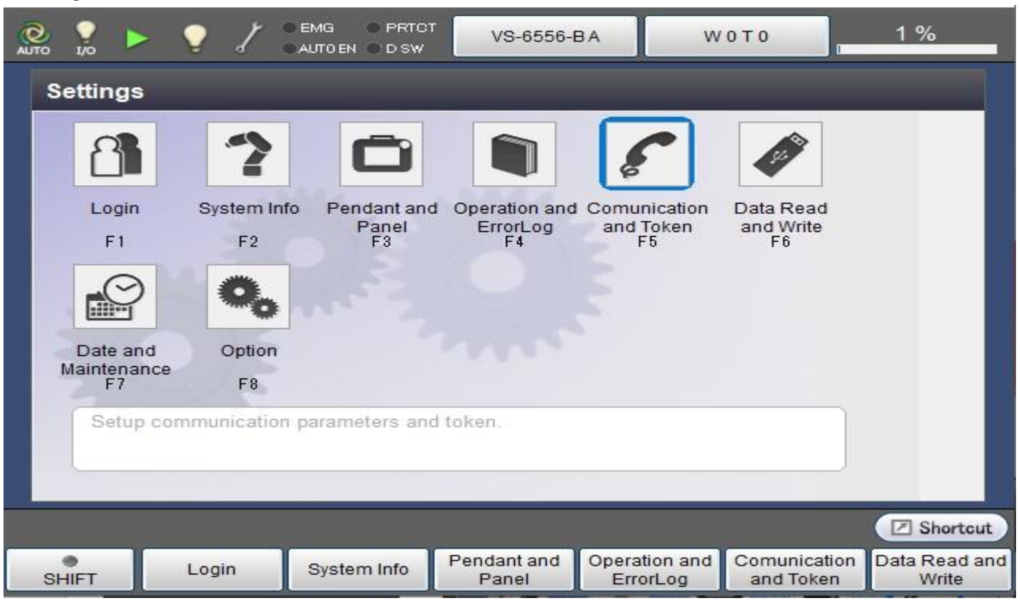
Plug-In Installation

Step 5.



Using the robot's **TeachPendant**, from the **main menu**, press the "**Setting**" button.

Step 6.

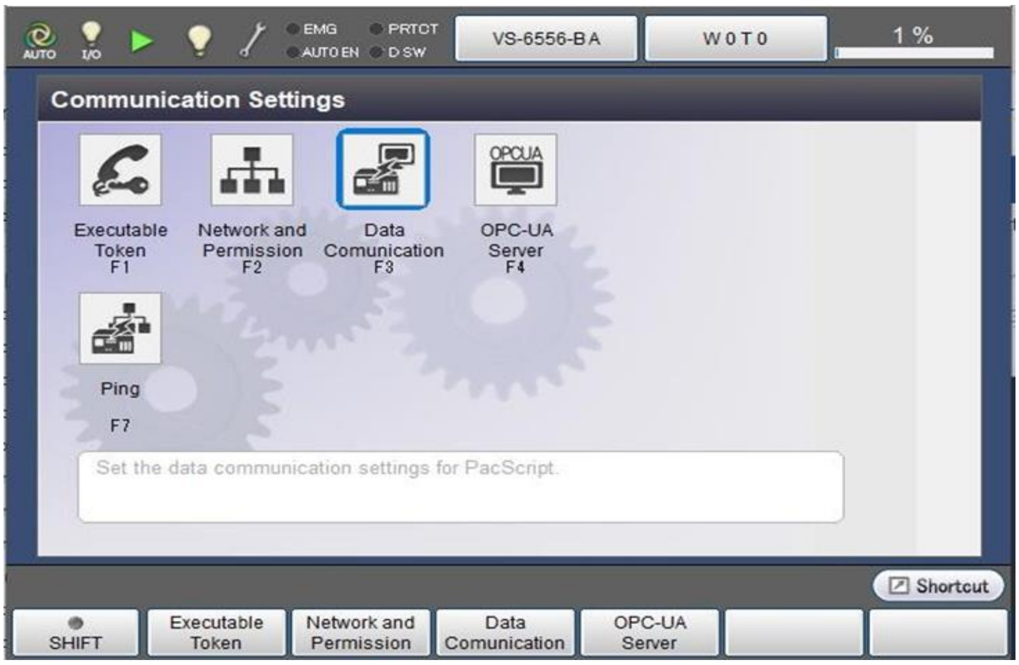


Setup communication parameters and token.

Select "**Communication and Token F5**"

Plug-In Installation

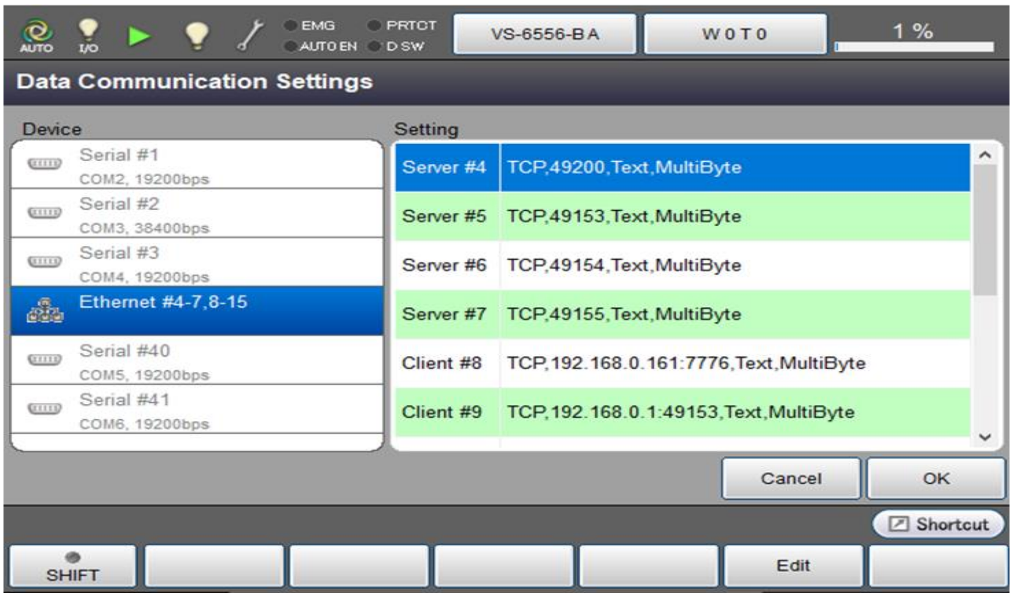
Step 7.



Set the data communication settings for PacScript.

Select "**Data Communication F3**".

Step 8.



Data Communication Settings

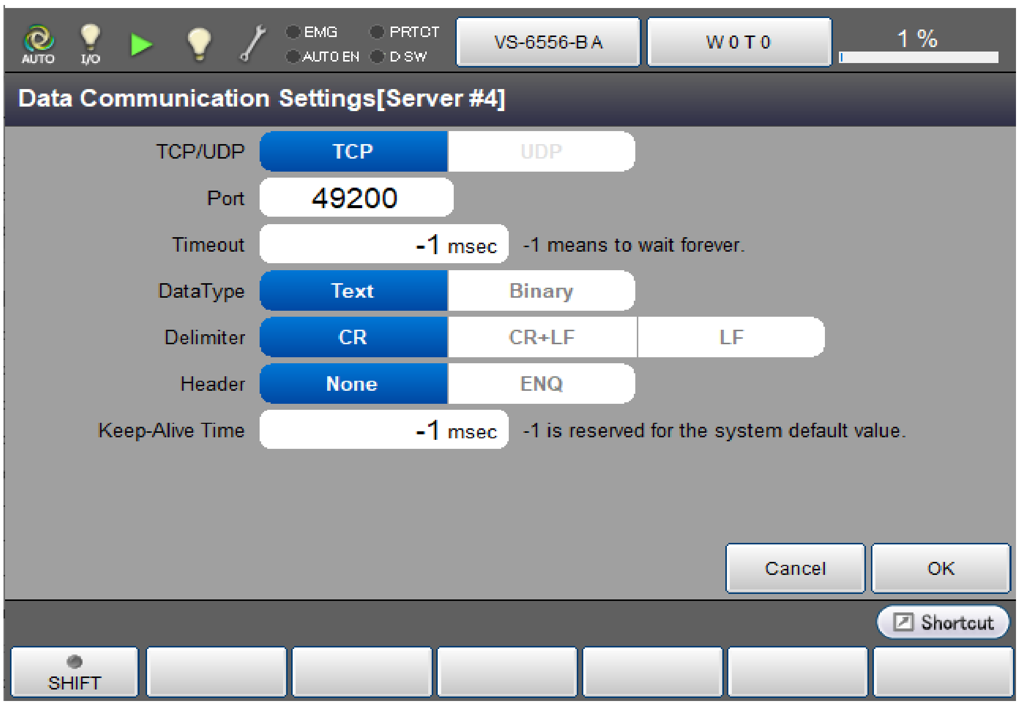
Device
Serial #1 COM2, 19200bps
Serial #2 COM3, 38400bps
Serial #3 COM4, 19200bps
Ethernet #4-7,8-15
Serial #40 COM5, 19200bps
Serial #41 COM6, 19200bps

Setting	
Server #4	TCP,49200,Text,MultiByte
Server #5	TCP,49153,Text,MultiByte
Server #6	TCP,49154,Text,MultiByte
Server #7	TCP,49155,Text,MultiByte
Client #8	TCP,192.168.0.161:7776,Text,MultiByte
Client #9	TCP,192.168.0.1:49153,Text,MultiByte

From the left menu, select the "**Ethernet #4-7,8-15**" item, then select "**Server#4**" and press the "**Edit**" button at the bottom right.

Plug-In Installation

Step 9.



EMG PRTCT AUTO EN D SW VS-6556-B A W 0 T 0 1 %

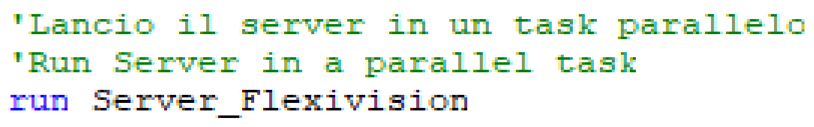
Data Communication Settings[Server #4]

TCP/UDP	[✓ TCP]	UDP	
Port	49200		
Timeout	-1 msec	-1 means to wait forever.	
DataType	[✓ Text]	Binary	
Delimiter	[✓ CR]	CR+LF	LF
Header	[✓ None]	ENQ	
Keep-Alive Time	-1 msec	-1 is reserved for the system default value.	

Set the communication "**port**" and copy the parameters shown in the image.
Press "**OK**" to go back and save.

Plug-In Installation

Step 10.



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

It will be important to start the **"Server_Flexivision"** communication Server in a parallel task so as to be able to communicate with the vision system also when running the robot's Pick&Place program.

The global variables used by the **"Server_Flexivision"** program to manage communication are the following:

- **S[S2_ComandoTx]**→ This must contain the string that you wish to send to the vision system.
- **IO128 (IO[BoolSendCommand])**→ By setting the bit 128 at ON **(set IO128)** the string in the variable **S[S2_ComandoTx]** will actually be sent. After having sent the command and received the response, the bit128 will be reset and return to OFF status.
- **S[S0_StringaRx]**→ This string will contain the response received from FlexiVision.
- **P[P0_PickPosition]**→ Following a localisation command ("start_Locator"/ "turn_Locator" etc.), coordinates X – Y – RZ of this position will be updated with the coordinates of the identified component.
- **S[S3_ModelName]→]** → Following a localisation command ("start_Locator"/ "turn_Locator".) this string will contain the name of the Pattern.

The **"Test_PlugIn.pcs"** program is present in the files supplied only to understand/verify operation of the **"Server_Flexivision"** before developing the final application.

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.

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, 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
```