

DENSO
Wincaps III

ars

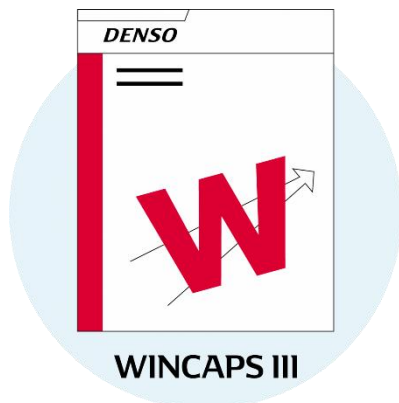
DENSO FLEXIBOWL PLUGIN



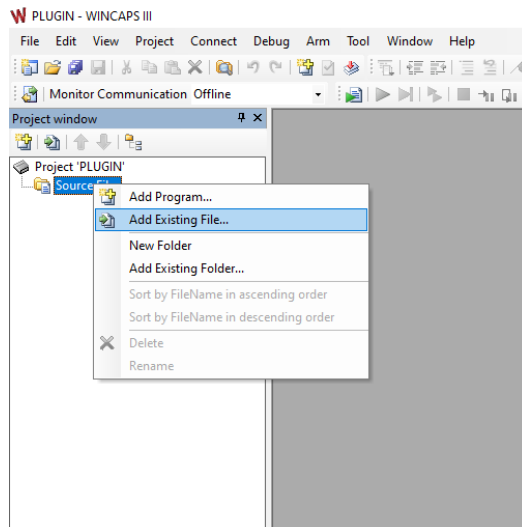
Questo Plugin è nato con l'idea di comunicare in maniera rapida e sicura con il FlexiBowl® tramite i robot DENSO, mediante l'utilizzo di istruzioni in linguaggio PacScript. Il Plugin NON necessita di una licenza aggiuntiva per la gestione dei socket.

FlexiBowl®

DENSO robotics

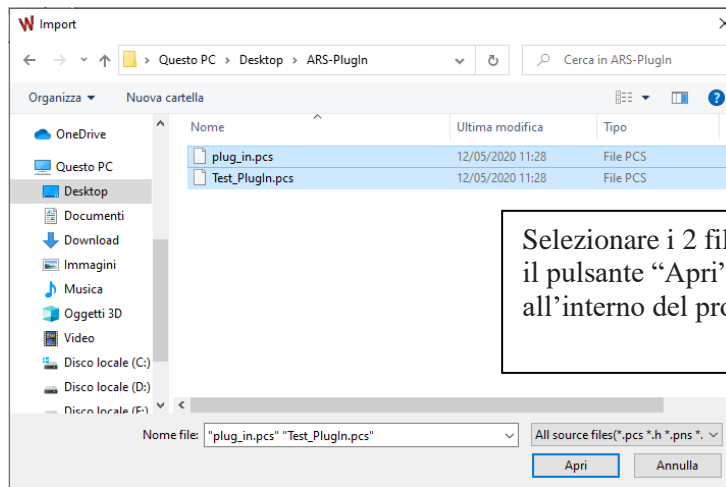


STEP 1:



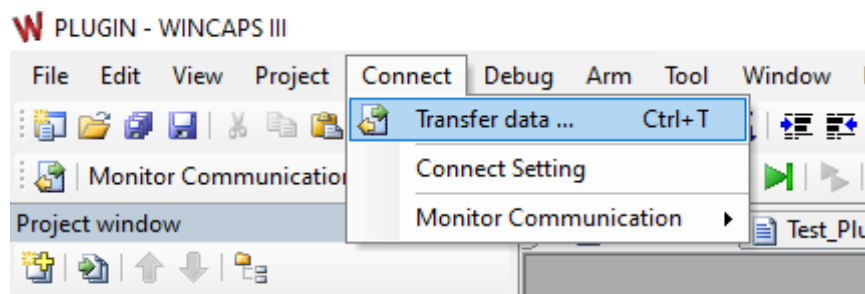
Aprire il software DENSO Wincaps 3 , fare click con il tasto destro su “Source Files” e successivamente selezionare “Add Existing File..” .

STEP 2:



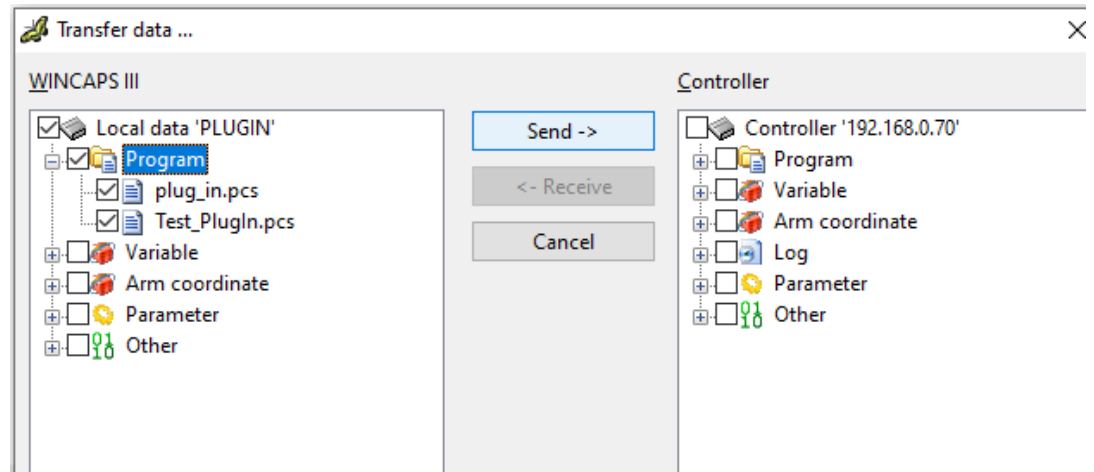
Selezionare i 2 file inviati da ARS, premere il pulsante “Apri” così da caricarli all’interno del progetto.

STEP 3:



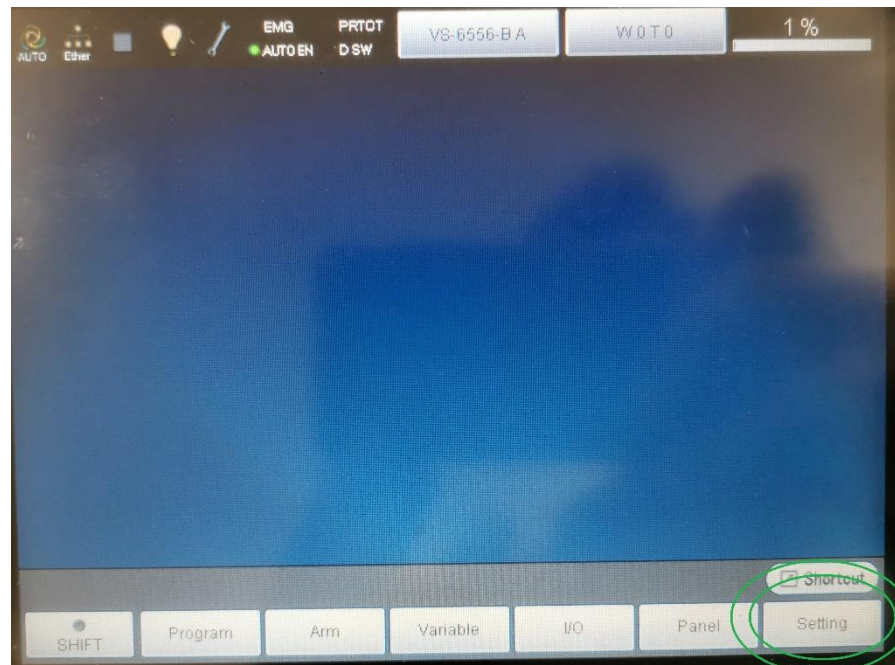
Dal programma WINCAPS selezionare “Connect” e successivamente “Transfer Data”.

STEP 4:



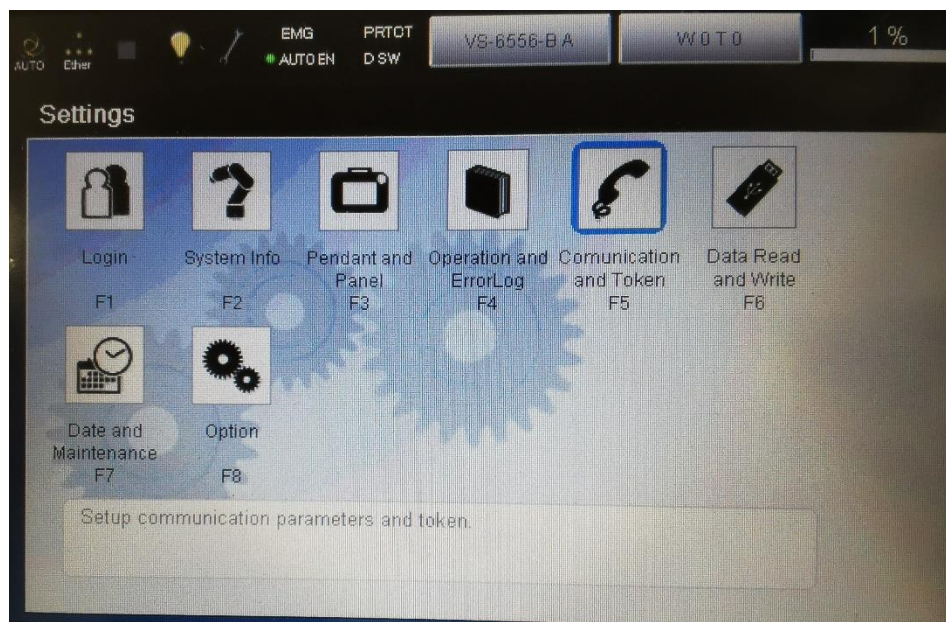
Trasferire quindi i programmi nella memoria del controller DENSO.

STEP 5:



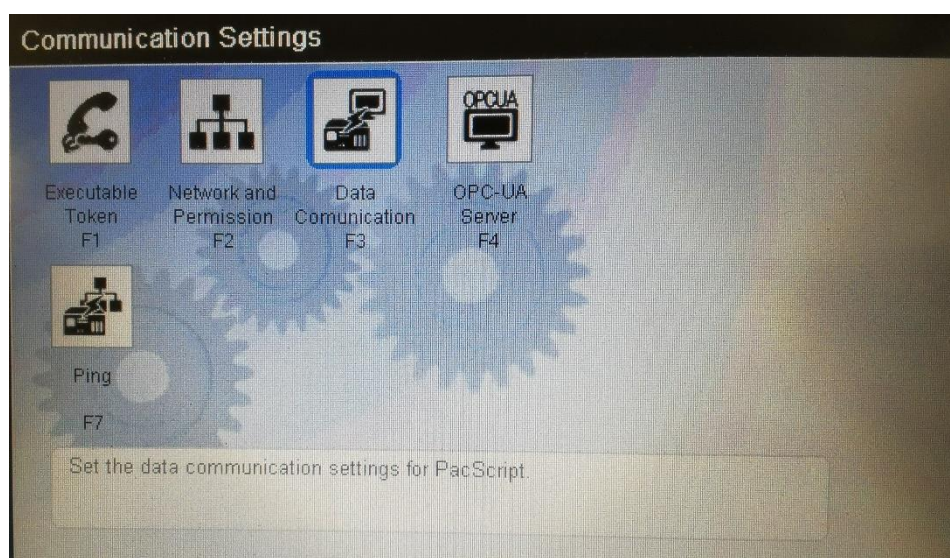
Utilizzando la TeachPendant del robot , dal menu principale, premere il pulsante “Setting”.

STEP 6:



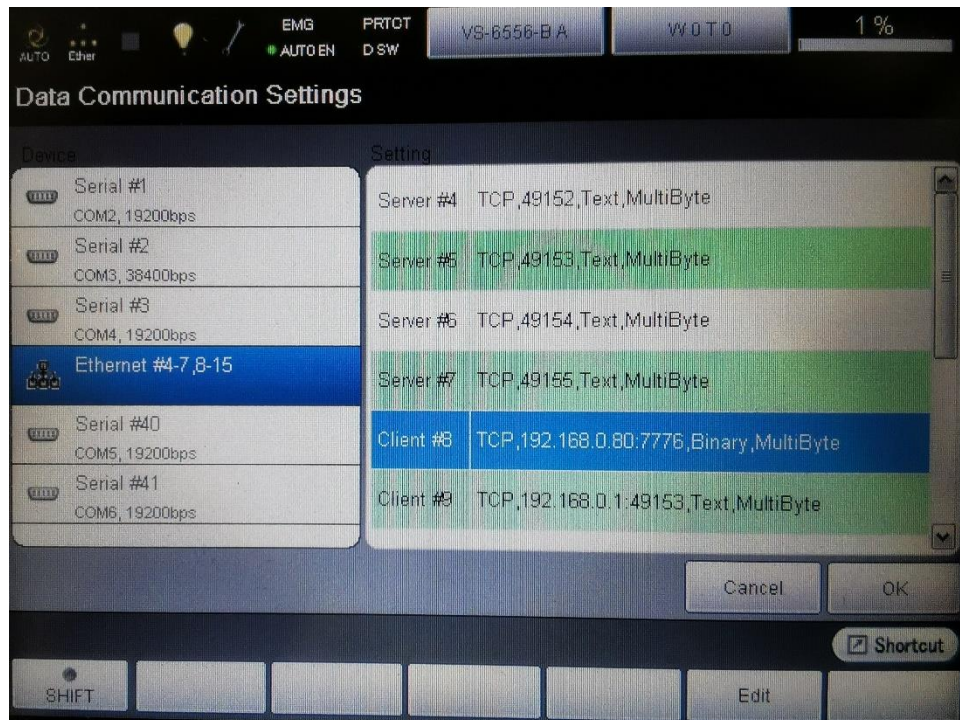
Selezionare "Communication and Token F5".

STEP 7:



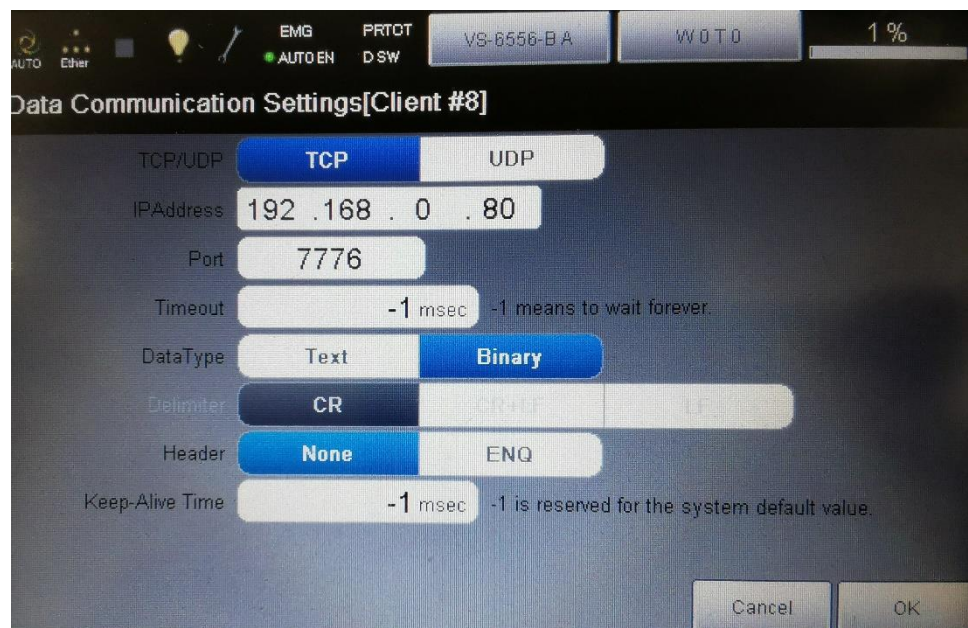
Selezionare "Data Communication F3".

STEP 8:



Denù di sinistra selezionare la voce “Ethernet #4-7,8-15” , poi selezionare “Client #8” e premere il pulsante “Edit” in basso a destra.

STEP 9:



Impostare l’indirizzo IP del Flexibowl modificando la voce “IPAddress” e copiare i parametri mostrati nell’immagine. Premere “OK” per tornare indietro e salvare.

STEP 10

```

Main
01 '!TITLE "Denso robot program"
02 #Include "plug_in.pcs"
03
04 Sub Main
05     TakeArm Keep = 0
06     dim ReturnStr As String
07
08     call plug_in("QX2",ReturnStr)
09
10     PrintMsg (ReturnStr)
11
12 End Sub
    
```

Specificare come primo argomento del programma “plug_in” **il comando** che si desidera inviare.

Il programma “Plug_in” fornirà in output una stringa (**ReturnStr**) contenente :

- “**Done**” se il comando inviato era un comando di movimento.
- “**Disconnect**” se c’è stato un errore.
- “\00” “\07” “risposta dal FlexiBowl” “\0D” se viene inviato un comando di interrogazione al driver .

Es.

Comando inviato= “\00” “\07” “SC” “\0D”

Risposta dal FlexiBowl= -“\00” “\07” “SC=0001” “\0D”

COMMAND
STRING
FORMAT:

Sintassi corretta per ogni pacchetto			
Header		Command	Footer
Chr(0)	Chr(7)	Comando	Chr(13)

COMMAND
LIST:

Comandi	Descrizione
QX2	Move
QX3	Move-Flip
QX4	Move-Flip-Blow
QX5	Move-Blow
QX6	Shake
QX7	Light on
QX8	Light off
QX9	Blow
QX10	Flip
QX11	Quick Emptying Option
QX12	Reset Alarm

SCRIPT:

```
'!TITLE "Denso robot program"
```

```
Sub Main(command as string, ReturnString as string)
```

```
'Dichiaro le variabili
```

```
'Define variables
```

```
Dim nmax As Integer
```

```
Dim n As Integer
```

```
dim TmpInt as integer
```

```
dim InputBytesCount as Integer
```

```
dim TimerVar as integer
```

```
dim CapitalCommand as string
```

```
Dim InputData as String
```

```
Dim BinaryArrayToSend As Variant
```

```
Dim ReceivedBinaryArray As Variant
```

```
Dim FlbReady As Variant
```

```
'Chiudo socket
```

```
'Close socket
```

```
Comm.Close 8
```

```
'Pulisco errori e definisco Error handling
```

```
'Clear errors and define Error handling
```

```
ClrErr
```

```
On Error GOTO ErrHandler
```

```
'Creo Array
```

```
'Create the array
```

```
BinaryArrayToSend = CreateArray( len(command)+3 )
```

```
'Definisco Header (0 7)
```

```
'Define the Header (0 7)
```

```
BinaryArrayToSend( 0 ) = &h00
```

```
BinaryArrayToSend( 1 ) = &h07
```

```
'Definisco il carattere di fine stringa
```

```
'Define "end string" char
```

```
BinaryArrayToSend(len(command)+2) = &h0D
```

```
'Metto in maiuscolo tutti i caratteri
```

```
'Set all characters in uppercase
```

```
CapitalCommand=UCCase(command)
```

```
'Converto in esadecimale il comando
```

```
'Convert to hexadecimal
```

```
For n=0 to (len(CapitalCommand)-1)
```

```
BinaryArrayToSend(2+n)="&H"+Hex(ASC(left(CapitalCommand,1)))
```

```
CapitalCommand=right(CapitalCommand,(len(CapitalCommand)-1))
```

```
Next
```

SCRIPT:

```
'Apro la porta comunicazione
'Open the socket port
Comm.Open 8
'Pulisco i dati
'Clear old datas
Comm.Clear 8
'Aspetto che la connessione venga stabilita
'Wait for the connection
Wait Comm.State(8) = 2

'Invio il comando al Flexibowl
'Send the command to Flexibowl
Comm.Output 8, BinaryArrayToSend

'Attendo la risposta
'Wait the answer
TimerVar=timer
do
InpuntBytesCount=comm.count(8)
    if (timer-TimerVar> 10000) then
        GOTO ErrHandler
    end if
loop until (InpuntBytesCount>0)

'Leggo la risposta
'Read the asnwer
ReceivedBinaryArray = Comm.Input(8,10000,InpuntBytesCount)

'Converto ogni byte (Esadecimale) in una stringa unica
'Convert each byte into a single string
nmax = Ubound(ReceivedBinaryArray)
For n = 0 To nmax
    TmpInt = ReceivedBinaryArray(n)
    InputData= InputData+chr(TmpInt)
Next

'Controllo la risposta
'Check the answer
If ((InStr(1,InputData,"%")>0) AND (InStr(1,UCase(command),"Q")>0)) then
    'Istruzione di movimento
    'Movement instruction
    FlbReady=0

'Definisco il messaggio da inviare al Flexibowl SC (Status Control)
'Define the message to be sent to the Flexibowl
BinaryArrayToSend = CreateArray(5)
BinaryArrayToSend( 0 ) = &h00    '00
BinaryArrayToSend( 1 ) = &h07    '07
BinaryArrayToSend( 2 ) = &h53    'S
BinaryArrayToSend( 3 ) = &h43    'C
BinaryArrayToSend( 4 ) = &h0D    '13
```


SCRIPT:

'Entro in un ciclo while fino a che la risposta non identifica la fine del movimento
'Wait in a loop cycle until the answer identify the end of the movement

do

'Invio il comando al Flexibowl
'Send the command to Flexibowl
Comm.Output 8, BinaryArrayToSend

'Attendo la risposta
'Wait the answer

TimerVar=timer
do

 InpuntBytesCount=comm.count(8)
 if (timer-TimerVar> 10000) then
 GOTO ErrHandler

 end if

loop until (InpuntBytesCount>0)

'Leggo la risposta
'Read the answer

ReceivedBinaryArray = Comm.Input(8,10000,InpuntBytesCount)

'Converto ogni byte (Esadecimale) in una stringa unica
'Convert each byte into a single string

nmax = Ubound(ReceivedBinaryArray)

InputData=""

For n = 0 To nmax

 TmpInt = ReceivedBinaryArray(n)
 InputData= InputData+chr(TmpInt)

Next

'Controllo il valore del byte meno significativo
'Check the value of the less significant byte

FlbReady= Right(InputData, 2)

'Aggiungo piccolo ritardo(10 ms) prima di controllare nuovamente

'Add a small delay before check again

delay 10

loop while (FlbReady<>1)'Resto in attesa fino a che il bit meno segnificativo con è uguale a 1

'Movimento terminato

'movement completed

ReturnString= "Done"

else

'Istruzione non di movimento

'No-Movement istration

ReturnString= InputData

end if

'Chiudo la connessione

'Close the connection

Comm.Close 8

'Ritorno al programma chiamante

'Return to main program

Exit Sub

'Errore durante la chiamata

'Error

*ErrorHandler:

 Comm.Close 8, -1

 PrintDbg "Err = " & Hex(Err.OriginalNumber)

 ReturnString= "Disconnect"

End Sub