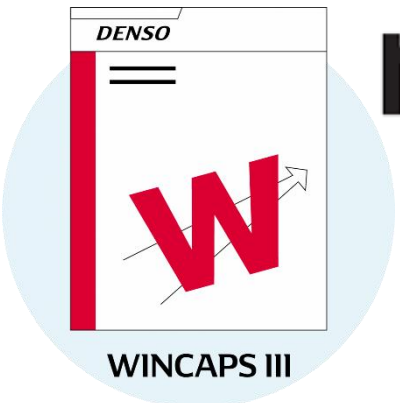
DENSO
Wincaps |||

DENSO FLEXIBOWL PLUGIN

This Plugin was developed with the idea of communicating quickly and safely with FlexiBowl® through DENSO robots by using instructions in PacScript.
The Plugin does NOT require an additional license to manage the sockets.

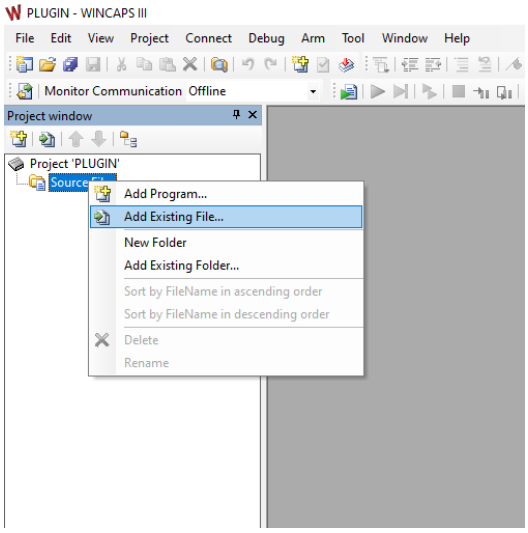
FlexiBowl®

DENSO robotics



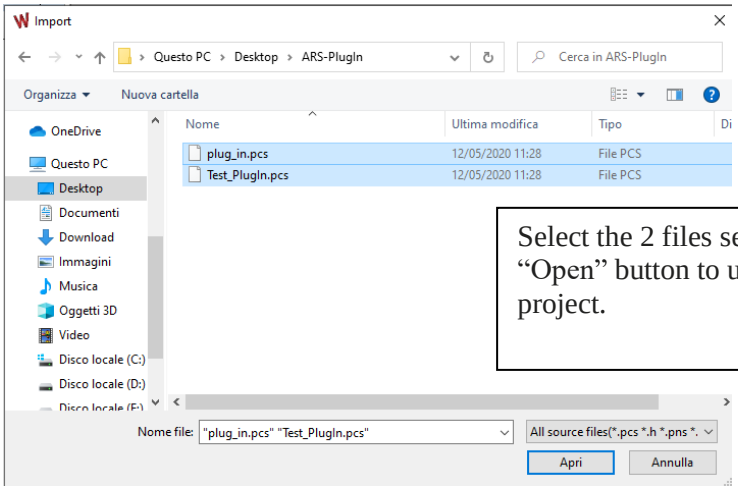
DENSO Wincaps III

STEP 1:



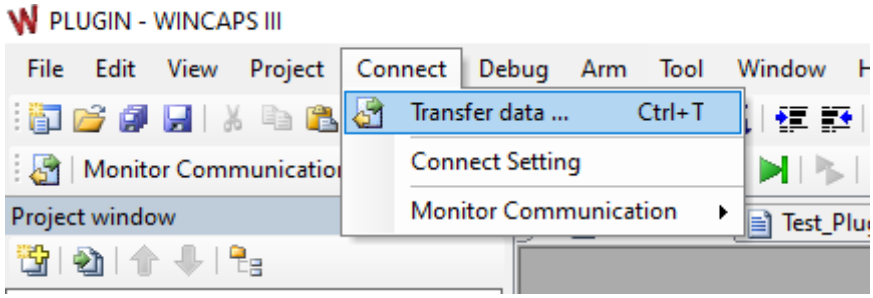
Open the DENSO Wincaps 3 software, right click on “Source Files” and then select “Add Existing File..”.

STEP 2:



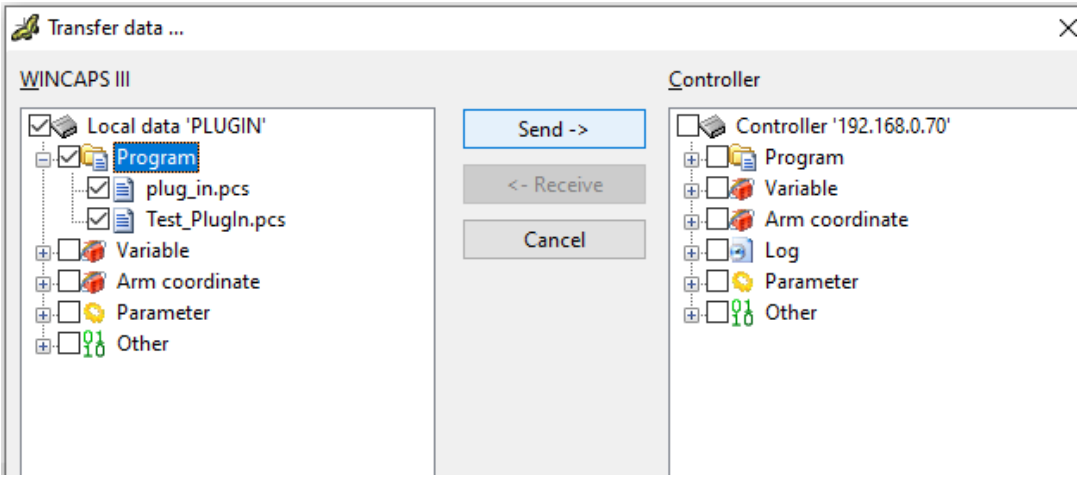
Select the 2 files sent by ARS, press the “Open” button to upload them into the project.

STEP 3:



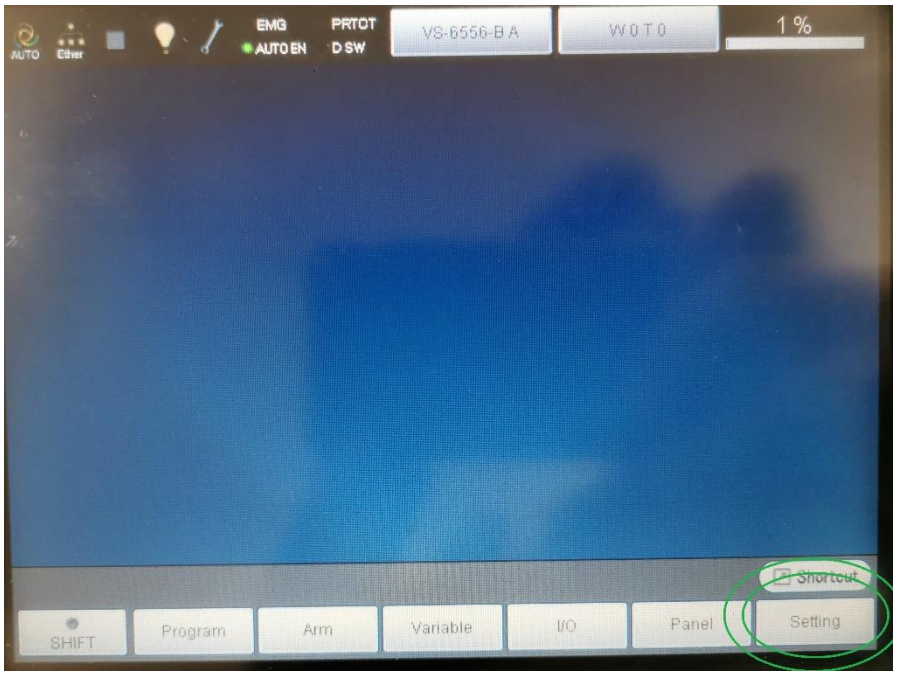
Select “Connect” from the WINCAPS program and then “Transfer Data”.

STEP 4:



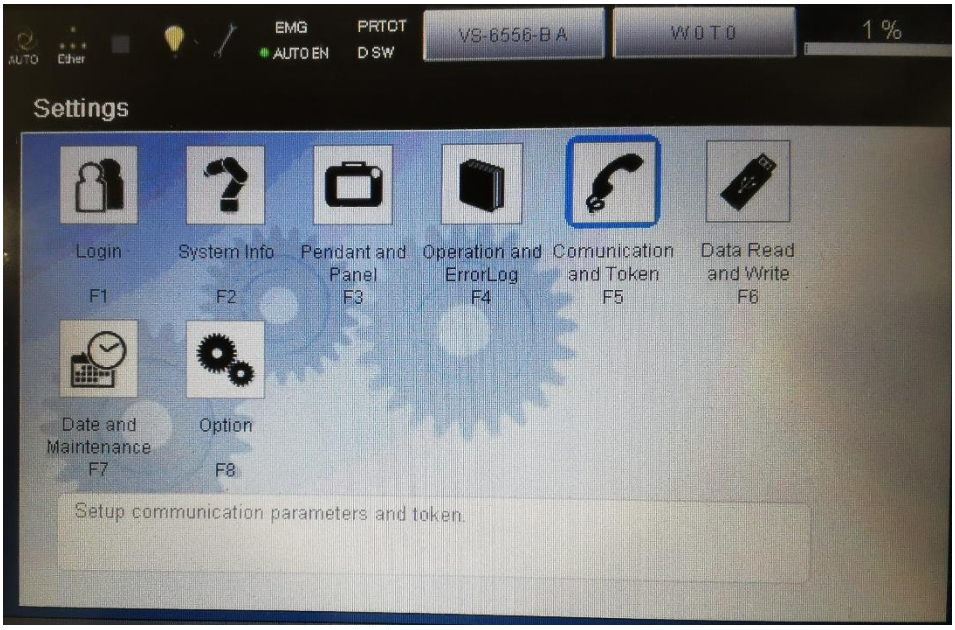
Then transfer the programs to the DENSO controller memory.

STEP 5:



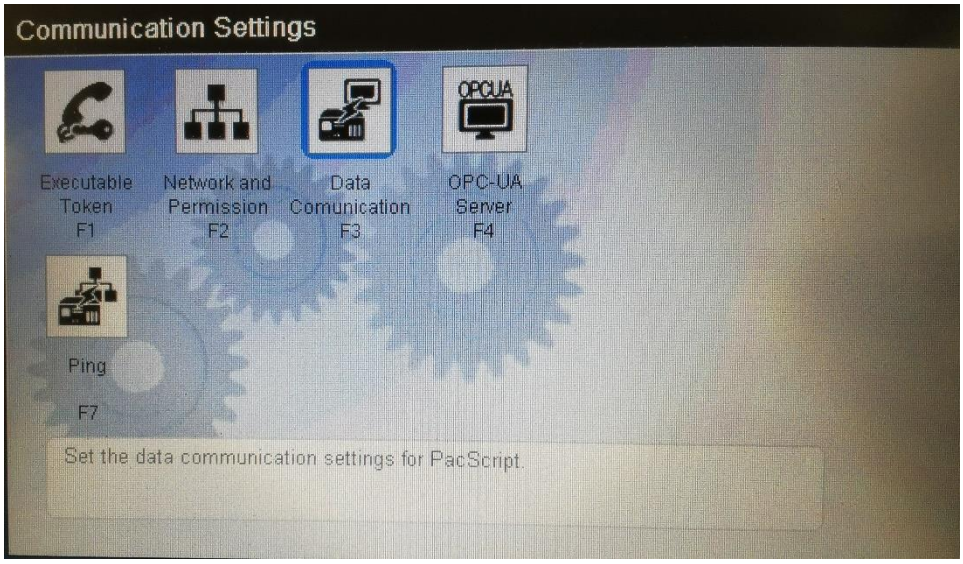
Using the robot's TeachPendant, from the main menu, press the "Setting" button.

STEP 6:



Select “Communication and Token F5”.

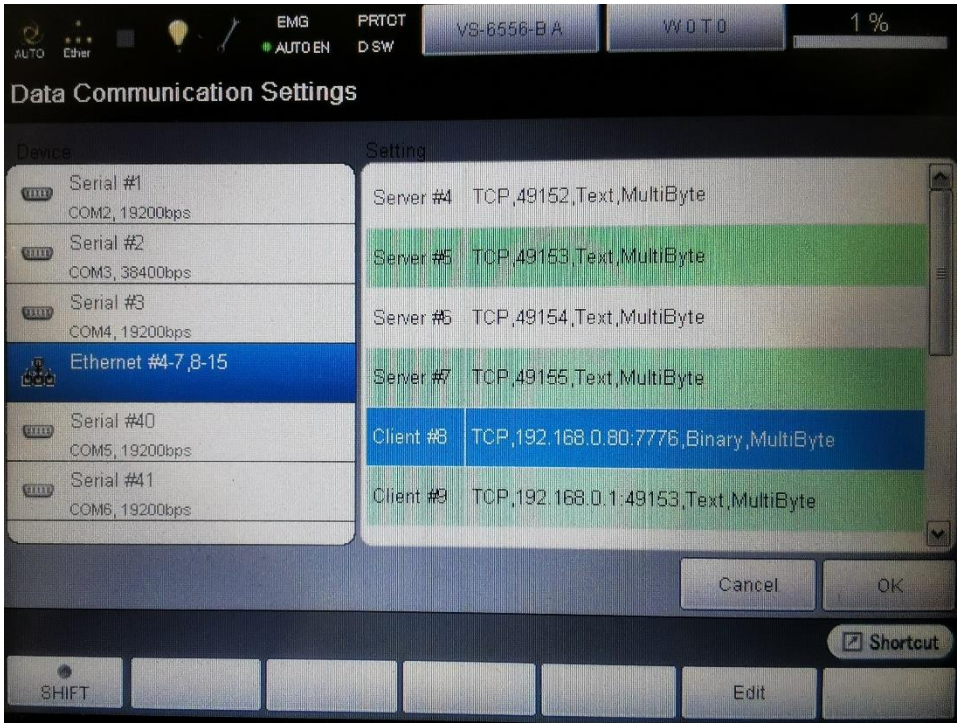
STEP 7:



Select “Data Communication F3”.

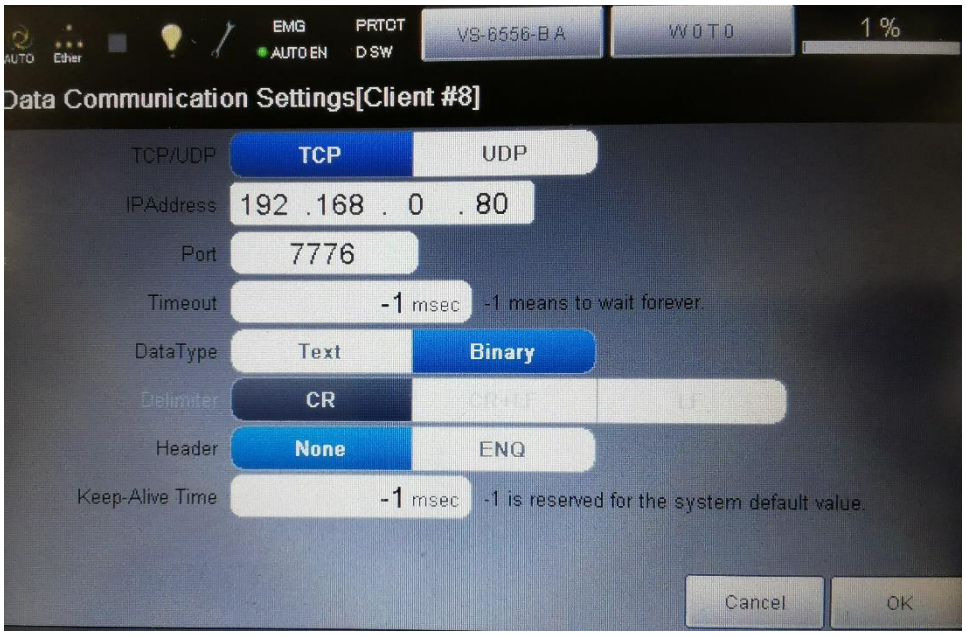
DENSO
Wincaps |||

STEP 8:



Left menu, select the “Ethernet #4-7,8-15” item, then select “Client #8” and press the “Edit” button at the bottom right.

STEP 9:



Set the Flexibowl IP address by changing the “IPAddress” item and copy the parameters shown in the image.
Press “OK” to go back and save.

DENSO
Wincaps |||

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
```

Specify **the command** you want to send as the first argument of the "plug_in" program.

The "Plug_in" program will output a string **(ReturnStr)** that contains:
-**"Done"** if the command sent was a movement command.
-**"Disconnect"** if there was an error.
-***"\00" "\07" "reply from the FlexiBowl" "\0D"*** if a query command is sent to the driver.

E.g.
Command sent= "\00" "\07" "SC" "\0D"
Reply from the 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

DENSO
Wincaps |||

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 InpuntBytesCount 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=UCase(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
InputBytesCount=comm.count(8)
        if (timer-TimerVar> 10000) then
                GOTO ErrHandler
        end if
loop until (InputBytesCount>0)

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

'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 signaficativo
	'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 istruction
ReturnString= InputData
end if
'Chiudo la connesione
'Close the connection
Comm.Close 8

'Ritorno al programma chiamante
'Return to main program
Exit Sub

'Errore durante la chiamata
'Error
 *ErrHandler:
	Comm.Close 8, -1
	PrintDbg "Err = " & Hex(Err.OriginalNumber)
	ReturnString= "Disconnect"

End Sub
```