STAUBLI
Robotics Suite

ars

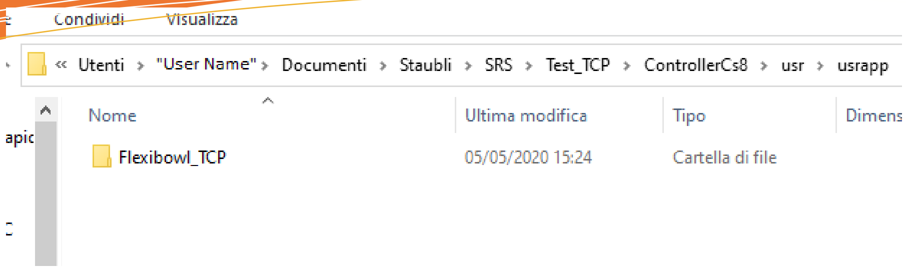
STAUBLI FLEXIBOWL PLUGIN

This Plugin was developed with the idea of communicating quickly and safely with FlexiBowl® through STAUBLI robots by using instructions in VAL3.
The Plugin does NOT require an additional license to manage the sockets.

FlexiBowl®

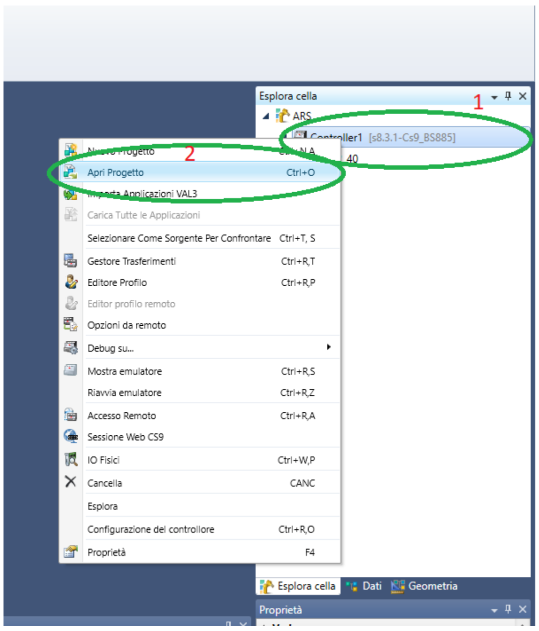
SRS 2016 - STAUBLI Robotics Suite 2016.6.6 - © Copyright - STAUBLI SA - 2003-2018
STÄUBLI
ROBOTICS
MAN AND MACHINE
SRS
Initializing...

STEP 1:



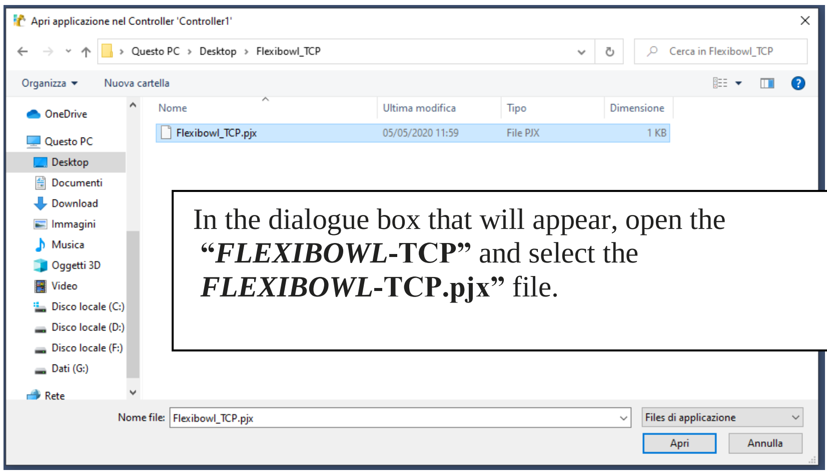
Insert the “**Flexibowl-TCP**” folder in the folder related to the SRS cell.
The “**Flexibowl-TCP**” folder will be provided by ARS.
The default path for SRS cells is:
C:\Users\”Nome Utente”\Documents\Staubli\SRS\”Nome cella”
\ControllerCs8\usr\usrapp.

STEP 2:


Esplora cella
ARS
Controller1 [s8.3.1-Cs9_BS885]
1
2
Nuovo Progetto
Apri Progetto Ctrl+O
Importa Applicazioni VAL3
Carica Tutte le Applicazioni
Selezionare Come Sorgente Per Confrontare Ctrl+T, S
Gestore Trasferimenti Ctrl+R,T
Editore Profilo Ctrl+R,P
Editor profilo remoto
Opzioni da remoto
Debug su...
Mostra emulatore Ctrl+R,S
Riavvia emulatore Ctrl+R,Z
Accesso Remoto Ctrl+R,A
Sessione Web CS9
IO Fisici Ctrl+W,P
Cancella CANC
Esplora
Configurazione del controllore Ctrl+R,O
Proprietà F4
Esplora cella
Dati
Geometria
Proprietà

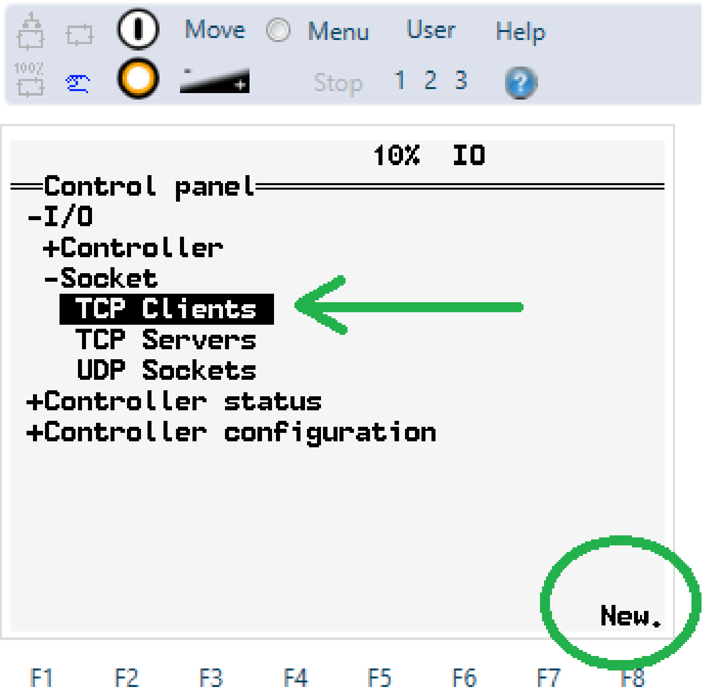
Open your cell with “**STAUBLI Robotics Suite**” .
From the “**Explore Cell**” menu right click on “**Controller**” and select “**Open Project**”.

STEP 3:



In the dialogue box that will appear, open the
“***FLEXIBOWL*-TCP**” and select the
***FLEXIBOWL*-TCP.pjx**” file.

STEP 4:

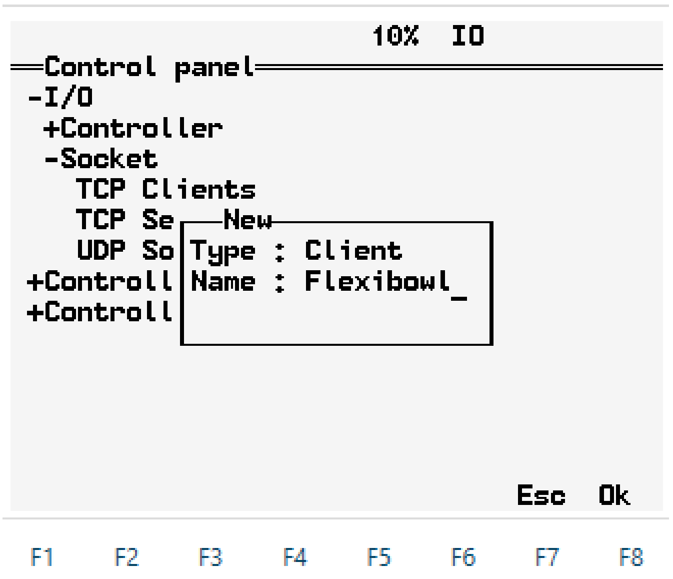


```
                   10%  IO
═Control panel═══════════════════
 -I/O
  +Controller
  -Socket
    TCP Clients
    TCP Servers
    UDP Sockets
 +Controller status
 +Controller configuration

                                New.
```

From the pendant, select the **"Control Panel"→I/O → "Socket"→"TCP Clients"** menu and create a new TCP (F8) client.

STEP 5:

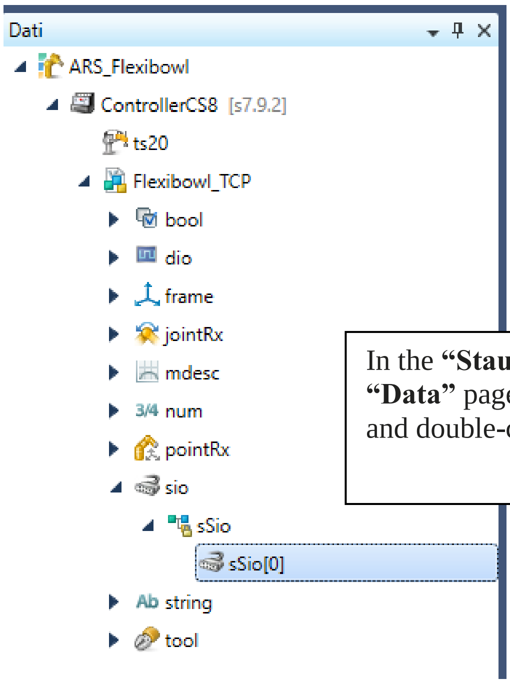


```
                   10%  IO
═Control panel═══════════════════
 -I/O
  +Controller
  -Socket
    TCP Clients
    TCP Se┌──New──────────────┐
    UDP So│Type : Client      │
 +Controll│Name : Flexibowl_  │
 +Controll│                   │
          └───────────────────┘

                          Esc  Ok
```

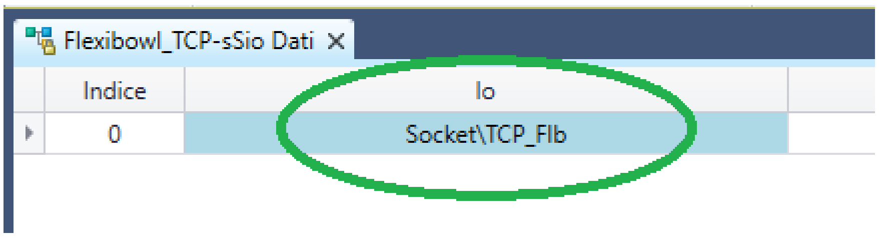
Set a name for the newly created Client, press F8 (Ok) and then press F8 again to end the procedure for inserting a new Client.
It is not important to define the client parameters as they will be set directly with the program provided by ARS.

STEP 6:



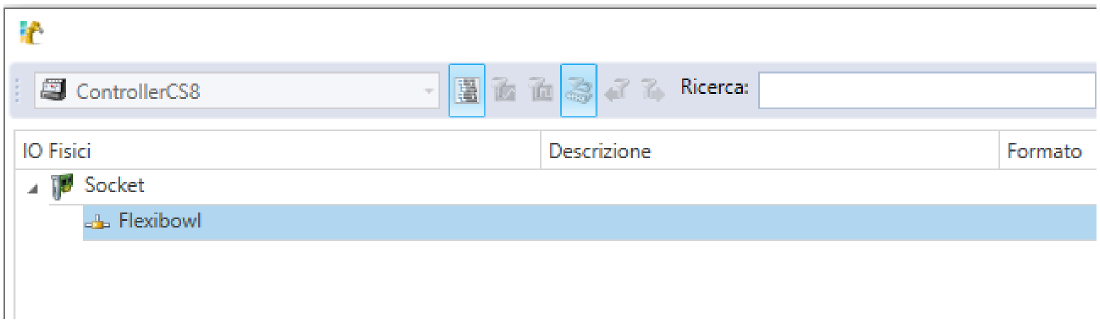
In the **"Staubli Robotics Suite"** program, select the **"Data"** page then open the **"Flexibowl_TCP"** project and double-click the **sSio[0]** variable.

STEP 7:



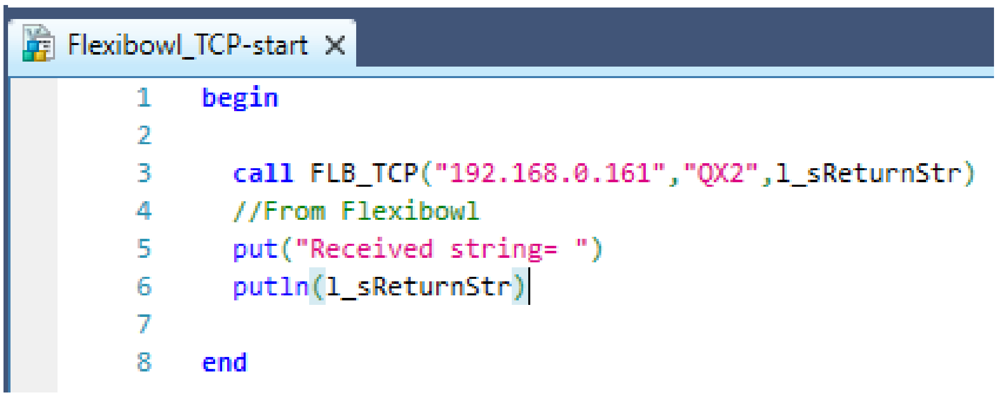
Indice	Io
0	Socket\TCP_Flb

Double-click the **"Socket\ TCP_Flb"** item.



Double click again on the highlighted item to complete the association of the Socket with the **"sSio"** variable

STEP 8:


Flexibowl_TCP-start

```
begin

  call FLB_TCP("192.168.0.161","QX2",l_sReturnStr)
  //From Flexibowl
  put("Received string= ")
  putln(l_sReturnStr)

end
```

Specify the **IP address** of the Flexibowl as the first argument of the FLB_TCP program, and **the command** you want to send as the second argument.

The ***FLB-TCP*** function will provide as output a string **(l_sReturnStr)** containing:
-**"*Done*"** if the command sent was a movement command.
-**"Communication error"** if an error occurred.
-***"\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

SCRIPT:

```
//Imposto la porta di comunicazione TCP
//Set the TCP communication port
l_iPortNum = 7776

//*********************************************
//Definisco i parametri del Socket ( porta, Indirizzo IP Flexibowl, carattere di fine stringa)
//Define Socket parameters (port, iP Address(Flexibowl), end of string character)
sioCtrl( sSio, "port",l_iPortNum)
sioCtrl( sSio, "target",x_sIP)
sioCtrl( sSio, "endOfString",13)
sioCtrl( sSio, "nangle",true)


//*********************************************
//Pulisco il buffer
//Clear socket buffer
clearBuffer( sSio)

//*********************************************
//Invio il messaggio al Flexibowl (chr(0)+chr(7) + Comando + chr(13)
//Send the string to Flexibowl

//invio 2 volte il chr(0)
//Send "chr(0" 2 times
l_nEOS=0
l_nResult=sioSet(sSio,l_nEOS)
l_nResult=sioSet(sSio,l_nEOS)
//Controllo se ci sono stati errori
//Check for errors
if l_nResult!=1
  x_sReturnString="Communication error"
  return
endIf

//invio il resto della stringa [chr(7)+comando+chr(13)]
//Send the rest of the string [chr(7)+comando+chr(13)]
l_sMessage=chr(7)+x_sCommand+chr(13)
for i=0 to len(l_sMessage)-1
  l_nResult=sioSet(sSio,asc(l_sMessage,i))
  //Controllo se ci sono stati errori
  //Check for errors
  if l_nResult!=1
    x_sReturnString="Communication error"
    return
  endIf
endFor
```

SCRIPT:

```
//*********************************************
  //Leggo la risposta del Flexibowl
  //Read the answer of Flexibowl
  l_sInputData=""
  do
    l_nResult=sioGet(sSio,l_nReceiveByte)
    l_sInputData=l_sInputData+chr(l_nReceiveByte)
  until l_nResult!=1 or l_nReceiveByte==13
  //Controllo se ci sono stati errori
  //Check for errors
  if l_nResult!=1
    x_sReturnString="Communication error"
    return
  endIf
  //*********************************************

  //Controllo il contenuto della risposta
  //Check the contents of the strings

  if((find(l_sInputData,"%")>0) and (find(l_sMessage,"Q")>0))
    //Istruzione di movimento
    //Movement instruction

    //Entro in un ciclo "infinito" fino a che il movimento non è completo/si sono verificati
errori
    //Go into a while loop until the movement is complete/ errors have occurred

               l_FlbMoving=1

                         while(l_FlbMoving==1)
      //Invio il messaggio al Flexibowl (chr(0)+chr(7) + Comando + chr(13)
      //Send the string to Flexibowl

      //invio 2 volte il chr(0)
      //Send 2 times chr(0)
      l_nEOS=0
      l_nResult=sioSet(sSio,l_nEOS)
      l_nResult=sioSet(sSio,l_nEOS)
      //Controllo se ci sono stati errori
      //Check for errors
      if l_nResult!=1
        x_sReturnString="Communication error"
        return
      endIf
```

SCRIPT:

```
//invio il comando "Status Control" per controllare se il movimento è terminato
   //Send the command "Sc" (Check status) and check if the movement has been completed
   l_sMessage=chr(7)+"SC"+chr(13)
   for i=0 to len(l_sMessage)-1
    l_nResult=sioSet(sSio,asc(l_sMessage,i))
    //Controllo se ci sono stati errori
    //Check for errors
    if l_nResult!=1
     x_sReturnString="Communication error"
     return
    endIf
   endFor

   // Leggo la risposta
   //Read the answer
   l_sInputData=""
   do
    l_nResult=sioGet(sSio,l_nReceiveByte)
    l_sInputData=l_sInputData+chr(l_nReceiveByte)
   until l_nResult!=1 or l_nReceiveByte==13
   //Controllo se ci sono stati errori
   //Check for errors
   if l_nResult!=1
    x_sReturnString="Communication error"
    return
   endIF

   //Controllo il valore del bit meno signaficativo: 1= Flexibowl pronto per ricevere un
nuovo comando / =9--> il Flexibowl è ancora in movimento
   //Chekc the value of the last numeric char: =1 --> Flexibowl is ready to receive new
command/ =9 --> Flexibowl is still moving
   l_sInputData=right(l_sInputData,2)
   toNum(l_sInputData,l_nSC_Status,l_bError)
   if(l_nSC_Status==1)
    l_FlbMoving=0
   endIf

   //Delay of 10 ms
   delay(0.1)

  endWhile
  //movimento completato
  //movement completed
  x_sReturnString="Done"
 else

  //Se il comando inviato non era relativo alla movimentazione del Flexibowl, verrà
restituita la risposta del Flexibowl.
  //If the command sent was not related to the movement of the Flexibowl, the Flexibowl's
response is returned
  x_sReturnString=l_sInputData
 endIf
```