STAUBLI
Robotics Suite

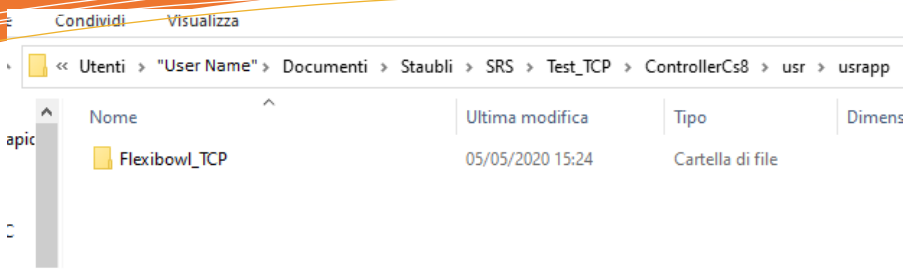
STAUBLI FLEXIBOWL PLUGIN

Questo Plugin è nato con l'idea di comunicare in maniera rapida e sicura con il FlexiBowl® tramite i robot STAUBLI, mediante l' utilizzo di instruzioni in linuaggio VAL3.
Il Plugin NON necessita di una licenza aggiuntiva per la gestione dei socket.

FlexiBowl®

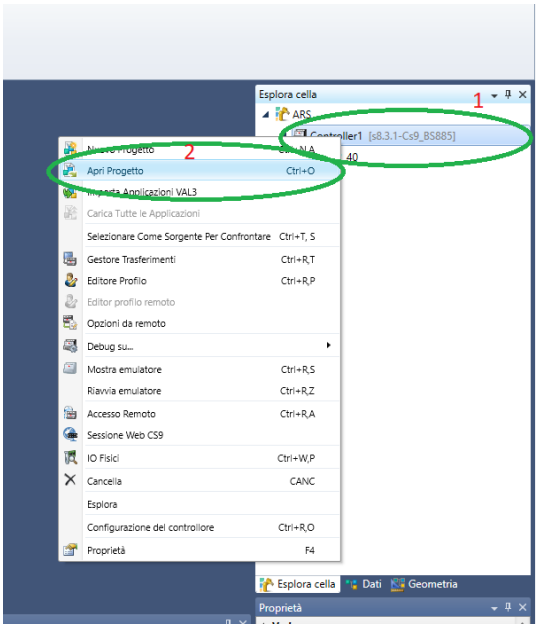
STAUBLI
Robotics Suite

STEP 1:



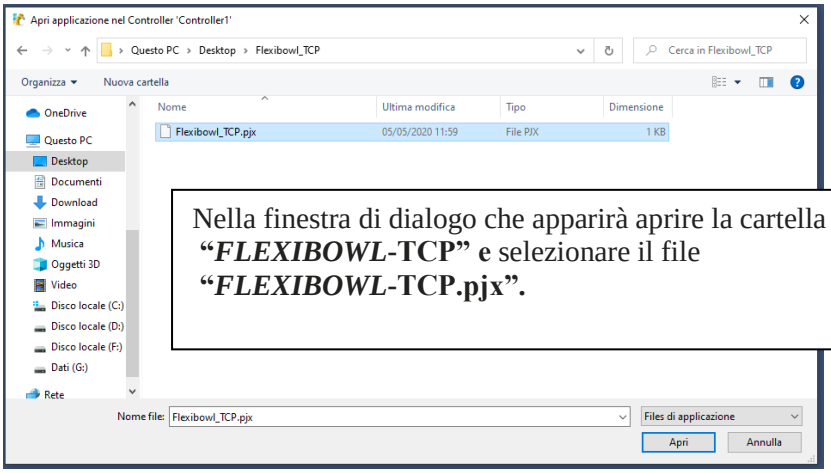
Inserire la cartella "**Flexibowl-TCP**" all'interno della cartella relariva alla cella SRS.
La cartella "**Flexibowl-TCP**" verrà fornita da ARS.
Il percorso di default per le celle SRS è:
C:\Users\"Nome Utente"\Documents\Staubli\SRS\"Nome cella" \ControllerCs8\usr\usrapp.

STEP 2:



Aprire la propria cella con "**STAUBLI Robotics Suite**" .
Dal menu "**Esplora Cella**" fare click con il tasto destro su "**Controller**" e selezionare "**Apri Progetto**".

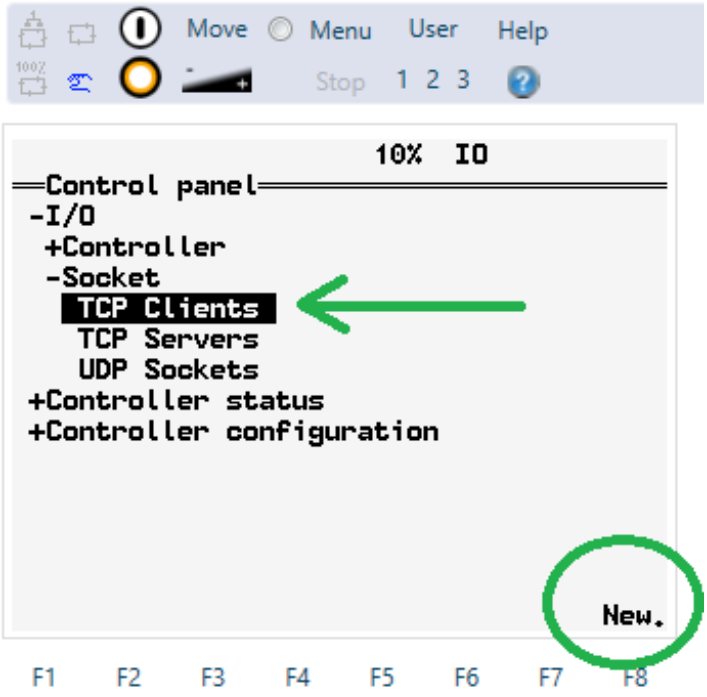
STEP 3:



Nella finestra di dialogo che apparirà aprire la cartella **"*FLEXIBOWL*-TCP" e** selezionare il file **"*FLEXIBOWL*-TCP.pjx".**

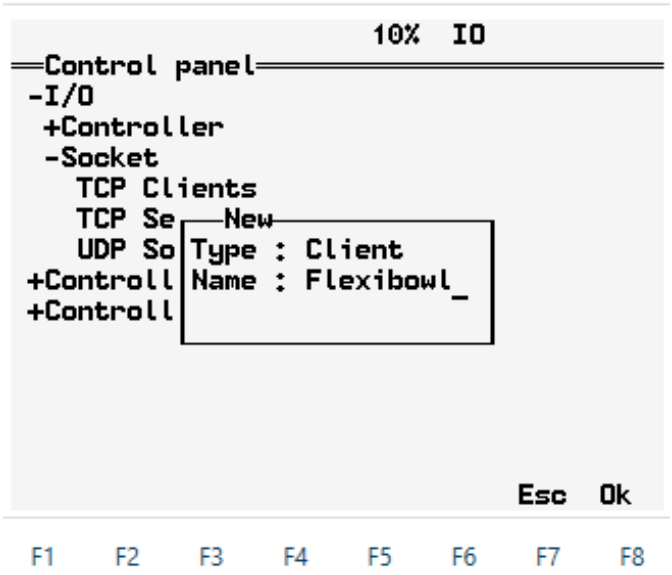
STAUBLI
Robotics Suite

STEP 4:



Dalla pendant selezionare il menu **"Control Panel"→I/O → "Socket"→"TCP Clients"** e creare un nuovo client TCP (F8).

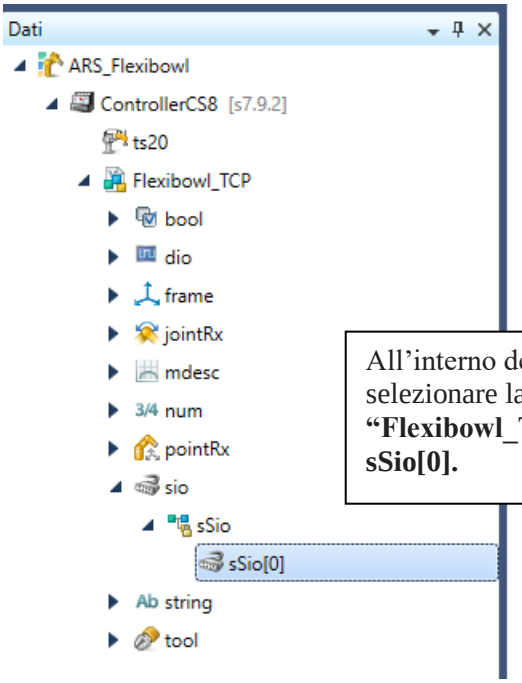
STEP 5:



Impostare un nome al Client appena creato, premere il tasto F8(Ok) e successivamente premere nuovamente il tasto F8 per terminare la procedura di inserimento di un nuovo Client .
Non è importante definire i parametri relativi al client poiché sarannò impostati direttamente con il programma fornito da ARS.

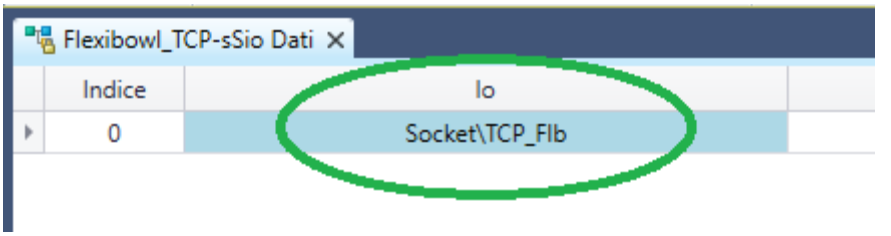
STAUBLI
Robotics Suite

STEP 6:

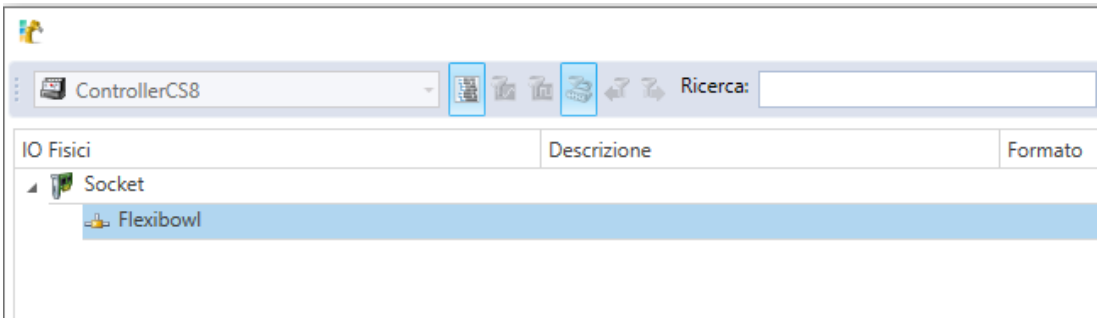


All'interno del programma **"Staubli Robotics Suite"** selezionare la pagina **"Dati"** quindi aprire il progetto **"Flexibowl_TCP"** e fare doppio click nella variabile **sSio[0].**

STEP 7:



Fare doppio click nella voce **"Socket\ TCP_Flb".**



Fare nuovamente doppio click nella voce evidenziata per completare l'associazione del Socket con la variabile **"sSio"**

STAUBLI
Robotics Suite

STEP 8:

Flexibowl_TCP-start

```
1   begin
2
3     call FLB_TCP("192.168.0.161","QX2",l_sReturnStr)
4     //From Flexibowl
5     put("Received string= ")
6     putln(l_sReturnStr)
7
8   end
```

Specificare come primo argomento del programma FLB_TCP **l'indirizzo IP** del Flexibowl ,come secondo argomento **il comando** che si desidera inviare.

La funzione ***FLB-TCP***, fornirà in output una stringa **(l_sReturnStr)** contenente :
- **"*Done*"** se il comando inviato era un comando di movimento.
-**"Communication error"** 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

STAUBLI
Robotics Suite

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
```

STAUBLI
Robotics Suite

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
```