

Manual



FlexiVision

STÄUBLI PLUG-IN

ars

Feeding
Industrial
Robotics

INDEX

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

This Plugin was developed with the idea of communicating with the **FlexiVision 2.0 system in a fast and safe way** through the **STAUBLI** robots, through the use of instructions in **VAL3** language.

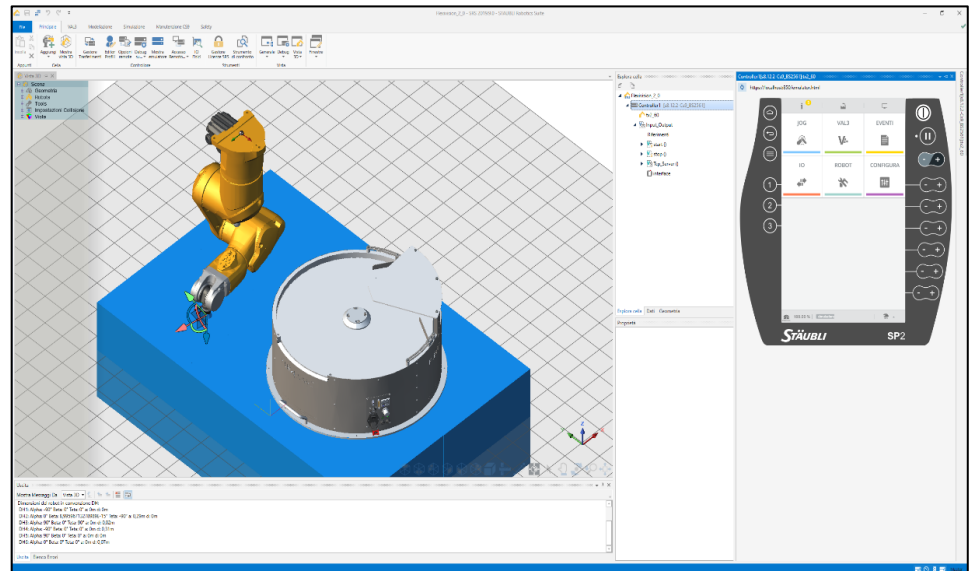
The Plugin does NOT require an additional license to manage the sockets.

FlexiBowl[®] Plug-In

STÄUBLI

Plug-In Installation

Together with this guide, a basic example developed directly with Srs is provided in order to understand the steps to implement the application.



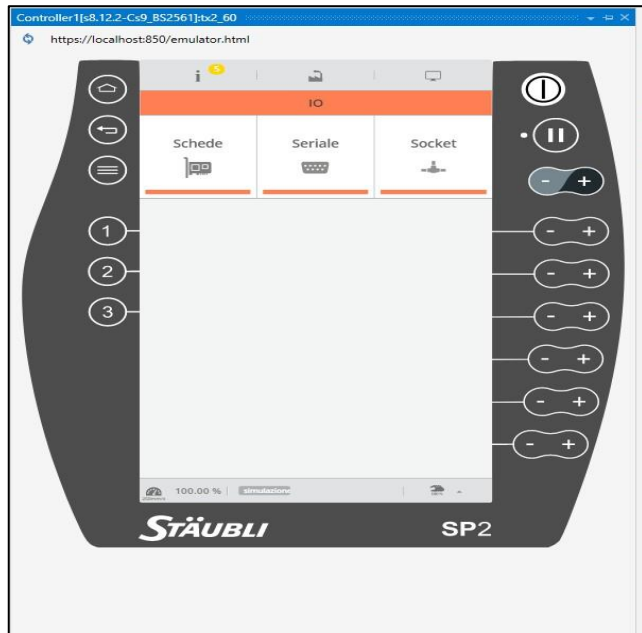
Step 1.

First of all, **create the server** from the SRS application to create communication with FlexiVision.

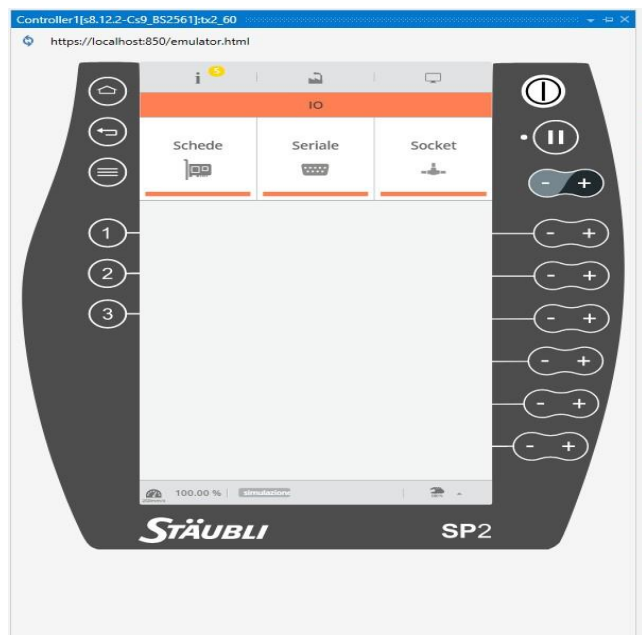


Plug-In Installation

Select IO



Select Socket



Plug-In Installation

Edit the Server



Step 2.

We will now view the script that manages communication with FlexiVision.

The Start program starts the **Tcp_Server** task that is used for communication with FlexiVision.

Once started, it will always be active.

There are two essential variables to call up a function in **TCP_Server**.

B_Mission, is a bool that must be set as true in order to start running the TCP_Server. On completion of the operation, the TCP_Server will set this variable back to False.

N_Mission, is the mission number to be executed, for example, setting **N_Mission=1** and **B_Mission** as *True*, the Tcp_Server will run the **Start_Locator**.

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

Let's analyse the Main script

```

1  begin
2
3  resetMotion()
4  b_ServerON=false
5  b_Mission=false
6  n_Mission=0
7
8
9  //mi muovo in home
10 movej(Home,tTool,mNomSpeed)
11
12 //faccio partire il server
13 //call Tcp_Server()
14 taskCreate "Tcp_Server" , 10 , Tcp_Server()
15
16 //In attesa del server connesso con il client
17 wait(b_ServerON)
18
19 while(true)
20
21 //chiedo la loxcazione a flexivision
22 n_Mission=1
23 b_Mission=true
24 wait(b_Mission==false)
25
26
27 movej(appro(pPickFlexivision,tShift),tTool,mNomSpeed)
28 movel(pPickFlexivision,tTool,mNomSpeed)
29 waitEndMove()
30 movej(appro(pPickFlexivision,tShift),tTool,mNomSpeed)
31 movej(appro(Place,tShift),tTool,mNomSpeed)
32 movel(Place,tTool,mNomSpeed)
33 waitEndMove()
34 movej(appro(Place,tShift),tTool,mNomSpeed)
35 endwhile
36
37 end

```

At the beginning, the variables are reset:

- **b_ServerOn**, to understand if connection with FlexiVision has been executed correctly
- **b_Mission**, the semaphore to execute a request to FlexiVision
- **n_Mission**, to define the mission to be executed

Script

The **TaskCreate** function creates the TCP_Server task and runs it, on standby for the customer (FlexiVision) to connect b_ServerOn. When FlexiVision is connected, set the number of missions to be performed N_Mission=1 ("start_Locator")

Set b_Mission = True in order to run the task

Wait for b_Mission to become negative in order to read the return variable.

It is now possible to move to the pick-up point.

Having processed the pick-up point, I am now ready to reset B_Mission as True to receive the next pick-up location.

Tcp_Server function

```

Input_Output-Tcp_Server  X Vista 3D
1  begin
2
3  b_ServerON=false
4
5  //quota di presa
6  n_Z=200
7
8  //inizializzo il server
9  sioCtrl(sSio, "baudRate", 4001)
10 sioCtrl(sSio, "bits", 8)
11 sioCtrl(sSio, "parity", "none")
12 sioCtrl(sSio, "stopBits", 1)
13 sioCtrl(sSio, "timeout", 15)
14 sioCtrl(sSio, "endOfString", 13)
15
16 //server attivo, in attesa del client
17 popUpMsg("Server Attivo, in attesa del client")
18
19 Tcp_Return=-1
20 while(Tcp_Return!=1)
21     Tcp_Return = sioSet(sSio,0)
22 endwhile
23 popUpMsg("Client Connesso",1)
24 //clearBuffer(sSio)
25
26 b_ServerON=true
27

```


Script

```

switch n_Mission
  //////////////////////////////////////
  case 1

//START_LOCATOR

//Set string I_sMessage="start_Locator"
//Set array resize(ArrayStart,1,len(I_sMessage))
//Preparo array
for i=0 to len(I_sMessage)-1 ArrayStart[i]=asc(I_sMessage,i)
endFor
//Send Array Tcp_Return=sioSet(sSio[0],ArrayStart)

// Controllo se ci sono stati errori
//Check for errors
if Tcp_Return!=len(I_sMessage) popUpMsg("Communication error",3)
return
endif

//Leggo la risposta del Flexibowl
//Read the answer of Flexibowl I_sInputData=""
do
Tcp_Return=sioGet(sSio,I_nReceiveByte)
I_sInputData=I_sInputData+chr(I_nReceiveByte)
until I_nReceiveByte==13
popUpMsg("Received String: "+I_sInputData)

if (find(I_sInputData,"#")!=-1) popUpMsg("FlexiVision Error: "+I_sInputData,3) else
while (find(I_sInputData,"Hopper")!=-1)
//attivo la tramoggia, meglio con pulse
//dio[0]=true,0.5s I_sInputData="" do
  Tcp_Return=sioGet(sSio,I_nReceiveByte) I_sInputData=I_sInputData+chr(I_nReceiveByte)
  until I_nReceiveByte==13
  popUpMsg("Received String: "+I_sInputData)

endWhile

//splitto la risposta
//NAME i=find(I_sInputData,";")
//s_Temp= toString("1",i) s_NameModel=left(I_sInputData,i)
popUpMsg("Model Name: "+s_NameModel) I_sInputData=mid(I_sInputData,len(I_sInputData)-(i+1),i+1)
//popUpMsg(I_sInputData)

//X i=find(I_sInputData,",") s_X=left(I_sInputData,i) nJ=find(s_X,",")
if(nJ=-1) s_X=replace(s_X,",",",",1,nJ) endif
popUpMsg("X: "+s_X) I_sInputData=mid(I_sInputData,len(I_sInputData)-(i+1),i+1)
//popUpMsg(I_sInputData)

//Y i=find(I_sInputData,",") s_Y=left(I_sInputData,i) nJ=find(s_Y,",")
if(nJ=-1) s_Y=replace(s_Y,",",",",1,nJ) endif
popUpMsg("Y: "+s_Y) I_sInputData=mid(I_sInputData,len(I_sInputData)-(i+1),i+1)
//popUpMsg(I_sInputData)

```

Script

```
//RZ i=find(I_sInputData,";") s_RZ=left(I_sInputData,i) nJ=find(s_RZ,";")
if(nJ!= -1) s_RZ=replace(s_RZ,";",1,nJ) endif
popUpMsg("RZ: "+s_RZ)

toNum(s_X,n_X,bOk) toNum(s_Y,n_Y,bOk) toNum(s_RZ,n_RZ,bOk)
pPickFlexiVision={{n_X,n_Y,n_Z,0,0,n_RZ},{lefty, epositive, wpositive}}

b_Mission=false endif
Break

case 2
//STOP_LOCATOR
//Set string I_sMessage="stop_Locator"
//Set array resize(ArrayStart,1,len(I_sMessage))
//Preparo array
for i=0 to len(I_sMessage)-1
ArrayStart[i]=asc(I_sMessage,i) endFor

//Send Array Tcp_Return=sioSet(sSio[0],ArrayStart)

// Controllo se ci sono stati errori
//Check for errors
if Tcp_Return!=len(I_sMessage)
popUpMsg("Communicationerror",3) return
endif

b_Mission=false
Break

////////////////////////////////////
case 3
//TURN_LOCATOR.
//Set string I_sMessage="turn_Locator«
//Set array resize(ArrayStart,1,len(I_sMessage))
//Preparo array
for i=0 to len(I_sMessage)-1
ArrayStart[i]=asc(I_sMessage,i)
endFor

//Send Array Tcp_Return=sioSet(sSio[0],ArrayStart)

// Controllo se ci sono stati errori
//Check for errors
if Tcp_Return!=len(I_sMessage)
popUpMsg("Communication error",3) return
endif
```

Script

```
//Leggo la risposta del Flexibowl
//Read the answer of Flexibowl I_sInputData=""
do
  Tcp_Return=sioGet(sSio,I_nReceiveByte) I_sInputData=I_sInputData+chr(I_nReceiveByte) until
  I_nReceiveByte==13
  popUpMsg("Received String: "+I_sInputData)

if (find(I_sInputData,"#")!=-1) popUpMsg("FlexiVision Error: "+I_sInputData,3) else

while (find(I_sInputData,"Hopper")!=-1)
  //attivo la tramoggia, meglio con pulse
  //dio[0]=true,0.5s I_sInputData="" do
    Tcp_Return=sioGet(sSio,I_nReceiveByte) I_sInputData=I_sInputData+chr(I_nReceiveByte) until
    I_nReceiveByte==13
    popUpMsg("Received String: "+I_sInputData)

  endwhile

//split the answer

//NAME i=find(I_sInputData,",")
//s_Temp= toString("1",i)
s_NameModel=left(I_sInputData,i) popUpMsg("Model Name: "+s_NameModel)
I_sInputData=mid(I_sInputData,len(I_sInputData)-(i+1),i+1)
//popUpMsg(I_sInputData)

//X i=find(I_sInputData,",") s_X=left(I_sInputData,i) nJ=find(s_X,",")
s_X=replace(s_X,",",1,nJ) popUpMsg("X: "+s_X)
I_sInputData=mid(I_sInputData,len(I_sInputData)-(i+1),i+1)
//popUpMsg(I_sInputData)

//Y i=find(I_sInputData,",") s_Y=left(I_sInputData,i) nJ=find(s_Y,",")
s_Y=replace(s_Y,",",1,nJ) popUpMsg("Y: "+s_Y)
I_sInputData=mid(I_sInputData,len(I_sInputData)-(i+1),i+1)
//popUpMsg(I_sInputData)

//RZ i=find(I_sInputData,",") s_RZ=left(I_sInputData,i) nJ=find(s_RZ,",")
s_RZ=replace(s_RZ,",",1,nJ) popUpMsg("RZ: "+s_RZ)

//STRING TO NUM
toNum(s_X,n_X,bOk) toNum(s_Y,n_Y,bOk) toNum(s_RZ,n_RZ,bOk)

//POSITION FLEXIVISION
pPickFlexiVision={{n_X,n_Y,n_Z,0,0,n_RZ},{sfree,efree,wfree}}

b_Mission=false endif
break

////////////////////////////////////
```

Script

```

case 4
//TEST_LOCATOR
//Set string
  I_sMessage="test_Locator"
  //Set array
  resize(ArrayStart,1,len(I_sMessage))
  //Preparo array
  for i=0 to len(I_sMessage)-1
  ArrayStart[i]=asc(I_sMessage,i)
  endFor

  //Send Array
  Tcp_Return=sioSet(sSio[0],ArrayStart)

  // Controllo se ci sono stati errori
  //Check for errors
if Tcp_Return!=len(I_sMessage)
  popUpMsg("Communication error",3)
  return
endif

//Leggo la risposta del Flexibowl
//Read the answer of Flexibowl
I_sInputData=""
do
  Tcp_Return=sioGet(sSio,I_nReceiveByte)
  I_sInputData=I_sInputData+chr(I_nReceiveByte)
  until I_nReceiveByte==13
  popUpMsg("Received String: "+I_sInputData)

if (find(I_sInputData,"#")!=-1)
  popUpMsg("FlexiVision Error: "+I_sInputData,3)
  else

  //splitto la risposta
  //NAME
  i=find(I_sInputData,";")
  //s_Temp= toString("1",i)
  s_NameModel=left(I_sInputData,i)
  popUpMsg("Model Name: "+s_NameModel)
  I_sInputData=mid(I_sInputData,len(I_sInputData)-(i+1),i+1)
  //popUpMsg(I_sInputData)

  //X i=find(I_sInputData,";")
  s_X=left(I_sInputData,i)
  nJ=find(s_X,"")
  s_X=replace(s_X,".",",",1,nJ)
  popUpMsg("X: "+s_X)
  I_sInputData=mid(I_sInputData,len(I_sInputData)-(i+1),i+1)
  //popUpMsg(I_sInputData)

```

Script

```

//Y
i=find(l_sInputData,";")
s_Y=left(l_sInputData,i)
nJ=find(s_Y,";")
s_Y=replace(s_Y,";",1,nJ)
popUpMsg("Y:"+s_Y)
l_sInputData=mid(l_sInputData,len(l_sInputData)-(i+1),i+1)
//popUpMsg(l_sInputData)

//RZ
i=find(l_sInputData,";")
s_RZ=left(l_sInputData,i)
nJ=find(s_RZ,";")
s_RZ=replace(s_RZ,";",1,nJ)
popUpMsg("RZ:"+s_RZ)

toNum(s_X,n_X,bOk)
toNum(s_Y,n_Y,bOk)
toNum(s_RZ,n_RZ,bOk)
pPickFlexiVision={n_X,n_Y,n_Z,0,0,n_RZ},{sfree,efree,wfree}}
b_Mission=false
endif
break
////////////////////////////////////

case 5
//START CONTROL
//Set string l_sMessage="start_Control"
//Set array
resize(ArrayStart,1,len(l_sMessage))
//Preparo array
for i=0 to len(l_sMessage)-1
  ArrayStart[i]=asc(l_sMessage,i)
endFor

//Send Array Tcp_Return=sioSet(sSio[0],ArrayStart)

// Controllo se ci sono stati errori
//Check for errors
if Tcp_Return!=len(l_sMessage)
popUpMsg("Communication error",3)
return
endif

//Leggo la risposta del Flexibowl
//Read the answer of Flexibowl
l_sInputData=""
do
  Tcp_Return=sioGet(sSio,l_nReceiveByte)
  l_sInputData=l_sInputData+chr(l_nReceiveByte)
until l_nReceiveByte==13
popUpMsg("Received String: "+l_sInputData)

if(find(l_sInputData,"#")!=-1)
popUpMsg("FlexiVision Error: "+l_sInputData,3)
Else
//splitto la risposta
//NAME i=find(l_sInputData,";")
//s_Temp= toString("1",i)
s_NameModel=left(l_sInputData,i)
popUpMsg("Model Name: "+s_NameModel)
l_sInputData=mid(l_sInputData,len(l_sInputData)-(i+1),i+1)
//popUpMsg(l_sInputData)

```

Script

```

if(find(l_sInputData,"#")!=-1)
  popUpMsg("FlexiVision Error: "+l_sInputData,3)
Else
  //splitto la risposta
  //NAME i=find(l_sInputData,";")
  //s_Temp= toString("1",i)
  s_NameModel=left(l_sInputData,i)
  popUpMsg("Model Name: "+s_NameModel)
  l_sInputData=mid(l_sInputData,len(l_sInputData)-(i+1),i+1)
  //popUpMsg(l_sInputData)

  //X i=find(l_sInputData,";")
  s_X=left(l_sInputData,i) nJ=find(s_X,"")
  s_X=replace(s_X,".",1,nJ) popUpMsg("X: "+s_X)
  l_sInputData=mid(l_sInputData,len(l_sInputData)-(i+1),i+1)
  //popUpMsg(l_sInputData)

  //Y
  i=find(l_sInputData,";")
  s_Y=left(l_sInputData,i) nJ=find(s_Y,"")
  s_Y=replace(s_Y,".",1,nJ)
  popUpMsg("Y: "+s_Y)
  l_sInputData=mid(l_sInputData,len(l_sInputData)-(i+1),i+1)
  //popUpMsg(l_sInputData)

  //RZ
  i=find(l_sInputData,";")
  s_RZ=left(l_sInputData,i)
  nJ=find(s_RZ,"")
  s_RZ=replace(s_RZ,".",1,nJ)
  popUpMsg("RZ: "+s_RZ)

  toNum(s_X,n_X,bOk)
  toNum(s_Y,n_Y,bOk)
  toNum(s_RZ,n_RZ,bOk)
  pPickFlexiVision={{n_X,n_Y,n_Z,0,0,n_RZ},{sfree,efree,wfree}}

  b_Mission=false
endif
break
////////////////////////////////////

```

Script

```
case 6
//STATE_LOCATOR
//Set string
I_sMessage="state_Locator"
//Set array
resize(ArrayStart,1,len(I_sMessage))
//Preparo array
for i=0 to len(I_sMessage)-1
ArrayStart[i]=asc(I_sMessage,i) endFor

//Send Array
Tcp_Return=sioSet(sSio[0],ArrayStart)

// Controllo se ci sono stati errori
//Check for errors
if Tcp_Return!=len(I_sMessage)
popUpMsg("Communication error",3)
return
endif

//Leggo la risposta del Flexibowl
//Read the answer of Flexibowl
I_sInputData=""
do
Tcp_Return=sioGet(sSio,I_nReceiveByte)
I_sInputData=I_sInputData+chr(I_nReceiveByte)
until I_nReceiveByte==13

if(find(I_sInputData,"#")!=-1)
popUpMsg("FlexiVision Error: "+I_sInputData,3)
else
popUpMsg("Received String: "+I_sInputData)
sReturnString=I_sInputData
endif
b_Mission=false

break
////////////////////////////////////
```

Script

```
case 7
//START_EMPTYING
//Set string
l_sMessage="start_Empty"
//Set array
resize(ArrayStart,1,len(l_sMessage))
//Preparo array
for i=0 to len(l_sMessage)-1
ArrayStart[i]=asc(l_sMessage,i) endFor

//Send Array
Tcp_Return=sioSet(sSio[0],ArrayStart)

// Controllo se ci sono stati errori
//Check for errors
if Tcp_Return!=len(l_sMessage)
popUpMsg("Communication error",3)
return
endif

//Leggo la risposta del Flexibowl
//Read the answer of Flexibowl
l_sInputData=""
do
Tcp_Return=sioGet(sSio,l_nReceiveByte)
l_sInputData=l_sInputData+chr(l_nReceiveByte)
until l_nReceiveByte==13

if(find(l_sInputData,"#")!=-1)
popUpMsg("FlexiVision Error: "+l_sInputData,3)
Else
popUpMsg("Received String: "+l_sInputData)
sReturnString=l_sInputData
endif
b_Mission=false

break
```


Script

```

case 8
//GET_RECIPE
//Set string
I_sMessage="get_Recipe"
//Set array
resize(ArrayStart,1,len(I_sMessage))
//Preparo array
for i=0 to len(I_sMessage)-1
ArrayStart[i]=asc(I_sMessage,i) endFor

//Send Array
Tcp_Return=sioSet(sSio[0],ArrayStart)

// Contollo se ci sono stati errori
//Check for errors
if Tcp_Return!=len(I_sMessage)
popUpMsg("Communication error",3)
return
endif

//Leggo la risposta del Flexibowl
//Read the answer of Flexibowl
I_sInputData=""
do
    Tcp_Return=sioGet(sSio,I_nReceiveByte)
    I_sInputData=I_sInputData+chr(I_nReceiveByte) until
    I_nReceiveByte==13

if(find(I_sInputData,"#")!=-1)
    popUpMsg("FlexiVision Error: "+I_sInputData,3)
Else
    popUpMsg("Received String: "+I_sInputData)
    sReturnString=I_sInputData
endif
b_Mission=false

break
////////////////////////////////////

```

Script

```
case 9
//SET_RECIPE
//Set string
l_sMessage="set_Recipe"
//Set array
resize(ArrayStart,1,len(l_sMessage))
//Preparo array
for i=0 to len(l_sMessage)-1
ArrayStart[i]=asc(l_sMessage,i)
endFor

//Send Array
Tcp_Return=sioSet(sSio[0],ArrayStart)

// Controllo se ci sono stati errori
//Check for errors
if Tcp_Return!=len(l_sMessage)
popUpMsg("Communication error",3)
return
endif
b_Mission=false

break
////////////////////////////////////
endSwitch
endWhile
```