EPSON FLEXIBOWL PLUGIN

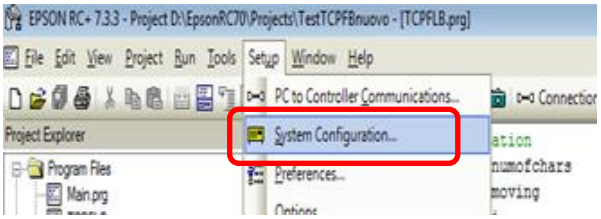
Questo Plugin è nato con l'idea di comunicare in maniera rapida e sicura con il FlexiBowl® tramite i robot Epson, mediante l' utilizzo di instruzioni in linuaggio RC+.
Il Plugin non necessita di nessuna licenza aggiuntiva

FlexiBowl®

EPSON SEIKO EPSON CORPORATION

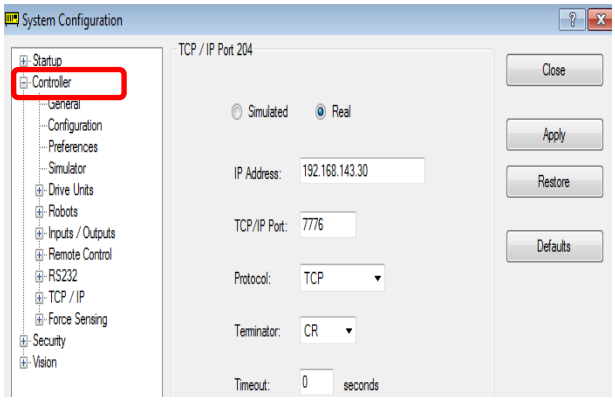
Epson RC+

STEP 1:



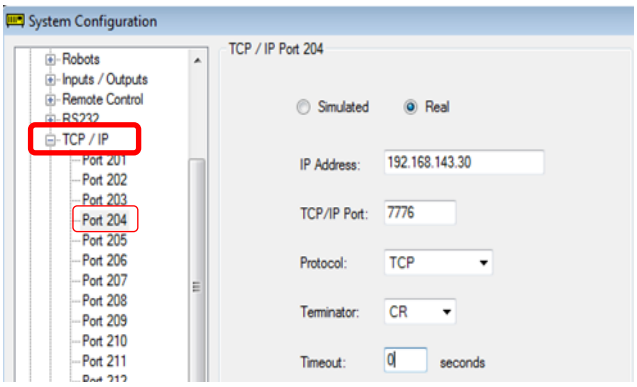
Nel menù Rc+ selezionare:
Setup→ System Configuration

STEP 2:



Nella finestra di dialogo che apparirà , selezionare ***Controller***

STEP 3:



Successivamente selezionare il menu relativo alla comunicazione ***TCP/IP.*** Una volta scelta la porta di comunicazione ci basterà configurarla seguendo il seguente standard

- ***IP:*** ip del flexibowl
- ***Tcp/ip port:*** 7776(standard)
- ***Protocol:*** TCP
- ***Terminator:*** CR

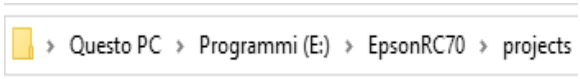
Epson RC+

STEP 4:

```
'opening com
OpenNet #204 As Client
WaitNet #204
```

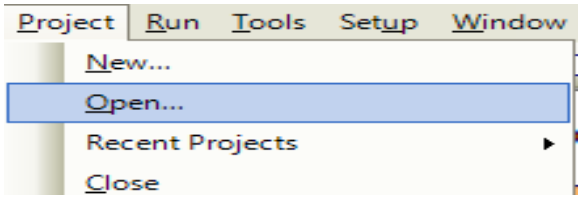
Il Plug-in Epson usa come standard la porta ***TCP/IP*** 204, se volessimo cambiarla ci basterà editare la parte di codice dove viene selezionata la porta da utilizzare .

STEP 5:



Copiare il Plug-in fornito da ARS nella cartella ***projects*** del programma ***RC+***

STEP 6:



Aprire il Progetto del Plug-in precedentemente importato.

STEP 7:

Call TCPFLB(Command$)

Basterà inserire questa linea di codice all' interno del nostro programma principale per poter comandare il **FlexiBowl®** in ogni sua funzione. Dovremo solo inserire il comando da eseguire come argomento.

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

Epson RC+

SCRIPT:

```
'varible declaration
Global Integer numofchars
Global Integer moving
Global Integer i
Global Integer pos
Global Double timeblow
Global String appoggio$
Global String appoggiomoving$
Global String data_tmp_Str$(0)
Global Byte inputbin(7)
Global Byte commandarray(5)
Global Byte movingbit(0)
Global Byte inputbyte
Global String messagearray$(0)

Function TCPFLB(message$ As String)
' Uncomment next line for debug mode
' #define DEBUG 1

'manual test command  ONLY IN DEBUG MODE
#ifdef DEBUG
Print "input command"
Line Input message$ 'Read one line input data into message
#endif

'clear vars
Redim commandarray(5)

'header
commandarray(0) = 0
commandarray(1) = 7

'command BODY
commandarray(2) = Asc("Q")
commandarray(3) = Asc("X")

'close communication char
commandarray(5) = 13

'opening com
OpenNet #204 As Client
WaitNet #204

message$ = LCase$(message$)
 'select the correct type of movement according to the input
    Select message$
```

Epson RC+

SCRIPT:

```
'****************************************MOVE**************************
      Case "move"

      commandarray(4) = Asc("2")

      #ifdef DEBUG
      Print "message =", message$
      Print "case = move"
      For i = 0 To UBound(commandarray)
      Print "pos", i, "=", commandarray(i)
      Next
      i = 0
      #endif

      WriteBin #204, commandarray(), 6
      Call waitmoving

'****************************************MOVE FLIP**********************

      Case "move flip"

      commandarray(4) = Asc("3")

      #ifdef DEBUG
      Print "message =", message$
      Print "case = move flip"
      For i = 0 To UBound(commandarray)
      Print "pos", i, "=", commandarray(i)
      Next
      i = 0
      #endif

      WriteBin #204, commandarray(), 6
      Call waitmoving
'*************************************MOVE FLIP BLOW*****************

      Case "move blow flip"

      commandarray(4) = Asc("4")

      #ifdef DEBUG
      Print "message =", message$
      Print "case = move blow flip"
      For i = 0 To UBound(commandarray)
      Print "pos", i, "=", commandarray(i)
      Next
      i = 0
      #endif

      WriteBin #204, commandarray(), 6
      Call waitmoving

```

Epson RC+

SCRIPT:

```
'****************************************MOVE BLOW************************
      Case "move blow"

          commandarray(4) = Asc("5")

          #ifdef DEBUG
          Print "message =", message$
          Print "case = move blow"
          For i = 0 To UBound(commandarray)
          Print "pos", i, "=", commandarray(i)
          Next
          i = 0
          #endif

          WriteBin #204, commandarray(), 6
          Call waitmoving
   '****************************************SHAKE************************

          Case "shake"

          commandarray(4) = Asc("6")

          #ifdef DEBUG
          Print "message =", message$
          Print "case = shake"
          For i = 0 To UBound(commandarray)
          Print "pos", i, "=", commandarray(i)
          Next
          i = 0
          #endif

          WriteBin #204, commandarray(), 6
          Call waitmoving
   '****************************************LIGHTON**********************
          Case "lighton"
          commandarray(4) = Asc("7")

          #ifdef DEBUG
          Print "message =", message$
          Print "case = shake"
          For i = 0 To UBound(commandarray)
          Print "pos", i, "=", commandarray(i)
          Next
          i = 0
          #endif

          WriteBin #204, commandarray(), 6
```

Epson RC+

SCRIPT:

```
'*****************************************LIGHTOFF*********************
*
        Case "lightoff"

        commandarray(4) = Asc("8")

        #ifdef DEBUG
        Print "message =", message$
        Print "case = shake"
        For i = 0 To UBound(commandarray)
        Print "pos", i, "=", commandarray(i)
        Next
        i = 0
        #endif

        WriteBin #204, commandarray(), 6

'*************************************************************************

        Default
'*****************************************BLOW************************

           If Not InStr(message$, "blow") Then
          'reallocate the command array
           Redim commandarray(7)
          'prepare the blowing request
           commandarray(0) = 0
           commandarray(1) = 7
           commandarray(2) = Asc("S")
           commandarray(3) = Asc("O")
           commandarray(4) = Asc("3")
           commandarray(6) = 13
           'extract the time for the blow
           pos = InStr(message$, "_")
           appoggio$ = Right$(message$, Len(message$) - pos)
           timeblow = Val(appoggio$)


           Print "blowing"
           commandarray(5) = Asc("L") 'SO3L turn on blow
           WriteBin #204, commandarray(), 6

           Wait timeblow

           commandarray(5) = Asc("H") 'SO3H turn off blow
           WriteBin #204, commandarray(), 6

           Print "stop blowing"
```

Epson RC+

SCRIPT:

```
'****************************************GENERIC COMMAND**************

            Else
           ' to upper message
            message$ = UCase$(message$)
' re allocate the dimension of the array according to the input string
            Redim commandarray(Len(message$) + 2)
            Redim messagearray$(Len(message$) - 1)
            commandarray(0) = 0
            commandarray(1) = 7
            appoggio$ = message$
           'split the input message in a chr array
            For i = 0 To (Len(message$) - 1)
               messagearray$(i) = Left$(appoggio$, 1)
               appoggio$ = Right$(appoggio$, Len(appoggio$) - 1)
            Next
           'format the commandarray
            For i = 0 To UBound(messagearray$)

               commandarray(i + 2) = (Asc(messagearray$(i)))

            Next

            commandarray(Len(message$) + 2) = 13


            'send the command array
            WriteBin #204, commandarray(), UBound(commandarray) + 1


            Call waitmoving


            EndIf
'*************************************************************************
```

Epson RC+

SCRIPT:

```
224.    Send
225.
226.    CloseNet #204
227.
228.    Fend
229.    Function waitmoving
230.    Print "Start..."
231.    'clear the commandarray
232.    Redim commandarray(4)
233.
234.    'prepare the moving? request
235.    commandarray(0) = 0
236.    commandarray(1) = 7
237.    commandarray(2) = Asc("S")
238.    commandarray(3) = Asc("C")
239.    commandarray(4) = 13
240.
241.    'wait the start of the movement
242.    moving = 0
243.    i = 0
244.    numofchars = ChkNet(204)
245.    If (numofchars >= 0) Then
246.    ReadBin #204, inputbyte
247.    EndIf
248.    'loop until the moving bit turn at 0 (chr48=0)
249.    Do While (moving <> 48)
250.
251.        WriteBin #204, commandarray(), 5
252.        numofchars = ChkNet(204)
253.        If (numofchars >= 5) Then
254.        Redim inputbin(0)
255.        Redim inputbin(numofchars)
256.        ReadBin #204, inputbyte
257.        ReadBin #204, inputbin(), numofchars
258.        Redim movingbit(UBound(inputbin))
259.        For i = 0 To UBound(inputbin)
260.        If (inputbin(i) = 13 Or inputbin(i) = 37) Then
261.        i = UBound(inputbin)
262.        Else
263.
264.        movingbit(i) = inputbin(i)
265.
266.        EndIf
267.        Next
268.        moving = movingbit(5)
269.        Wait 0.01
270.
271.        EndIf
272.
273.    Loop
274.
275.    Print "Done..."
276.
277.    Fend
```