

ars

MITSUBISHI FLEXIBOWL PLUGIN



Questo Plugin è nato con l'idea di comunicare in maniera rapida e sicura con il flexibowl tramite i robot Mitsubishi, mediante il software RT TOOLBOX ver. 1.81k.
Il Plugin non necessita di licenze aggiuntive Mitsubishi.

FlexiBowl®



Integrated FA Software

Robot Total Engineering Support Software

RT ToolBox3

Version 1.81K

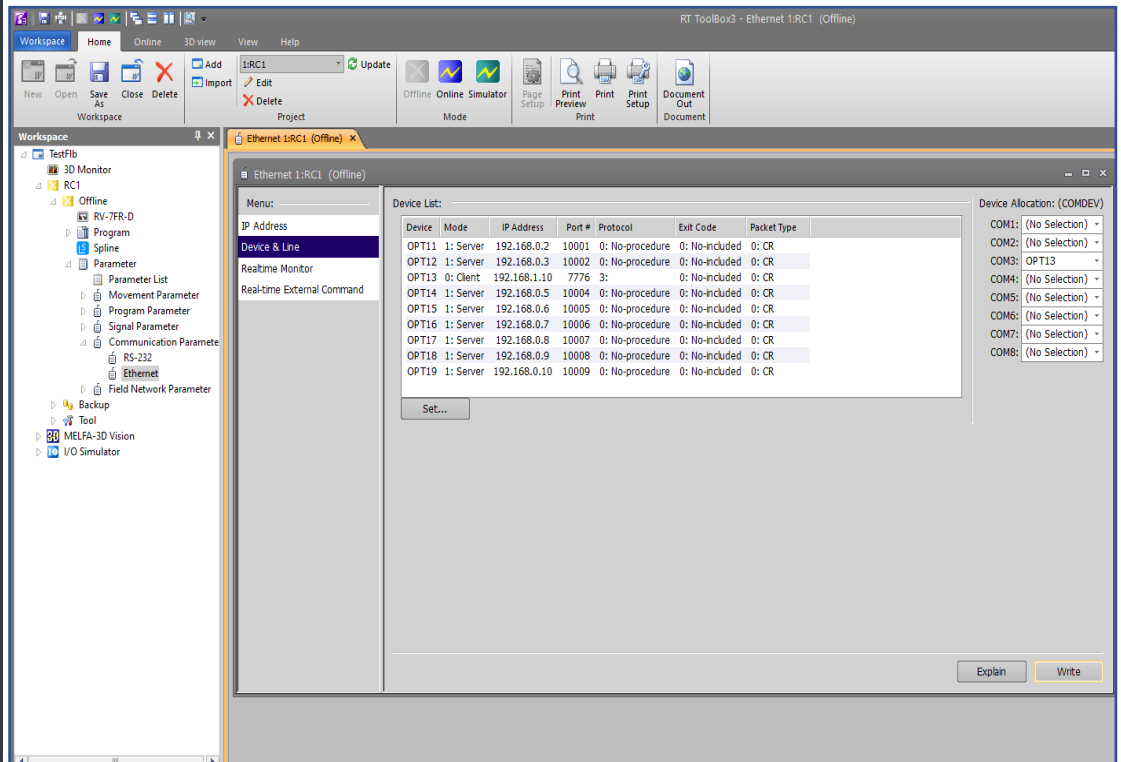
MELSOFT



MELSOFT

©2017 MITSUBISHI ELECTRIC CORPORATION ALL RIGHTS RESERVED

STEP 1:



Considerando che il FlexiBowl® abbia indirizzo Ip 192.168.1.10, settare una porta di comunicazione come nell'immagine.

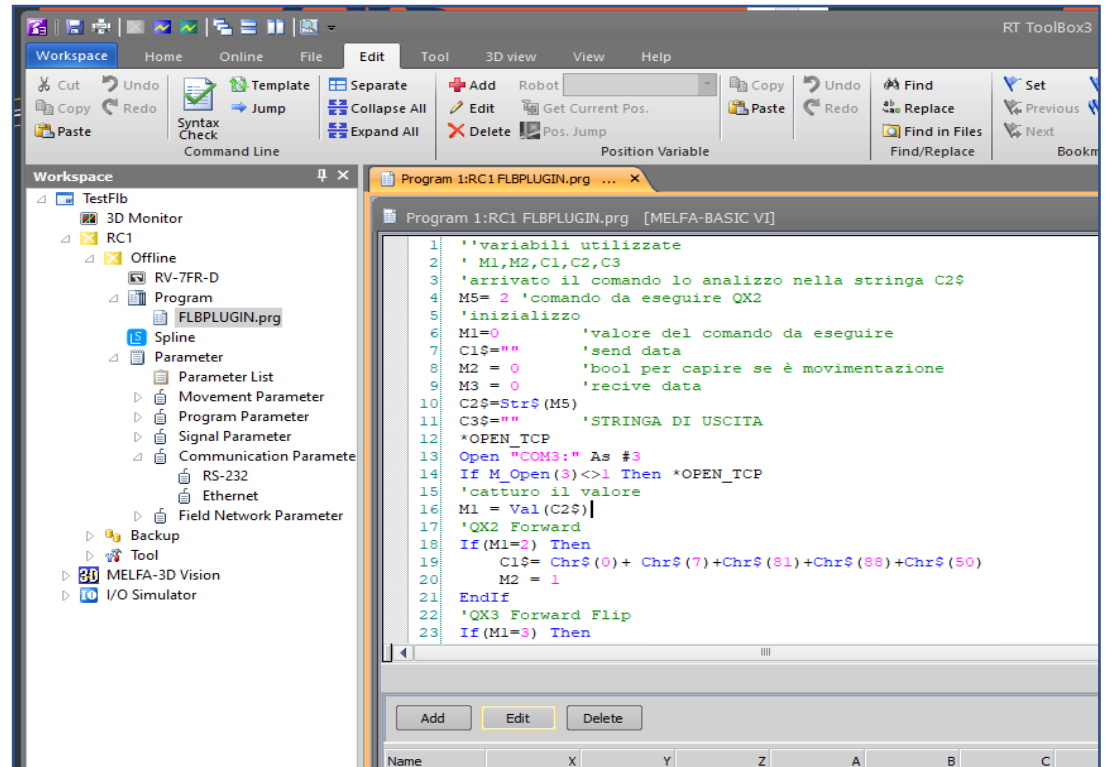
Porta 7775

Nell'immagine viene settata l'Ip, la porta, e successivamente settare la Com3 come OPT13

Una volta settata la "Device & Line", siamo pronti ad effettuare la comunicazione con il FlexiBowl®

STEP 2:

Ora verrà mostrato il codice utilizzato per movimentare il Flexibowl®



```

1  ''variabili utilizzate
2  ' M1,M2,C1,C2,C3
3  'arrivato il comando lo analizzo nella stringa C2$
4  M5= 2 'comando da eseguire QX2
5  'inizializzo
6  M1=0   'valore del comando da eseguire
7  C1$="" 'send data
8  M2 = 0 'bool per capire se è movimentazione
9  M3 = 0 'recive data
10 C2$=Str$(M5)
11 C3$="" 'STRINGA DI USCITA
12 *OPEN_TCP
13 Open "COM3:" As #3
14 If M_Open(3)<>1 Then *OPEN_TCP
15 'catturo il valore
16 M1 = Val(C2$)
17 'QX2 Forward
18 If (M1=2) Then
19   C1$= Chr$(0) + Chr$(7) +Chr$(81) +Chr$(88) +Chr$(50)
20   M2 = 1
21 Endif
22 'QX3 Forward Flip
23 If (M1=3) Then

```

```

''variabili utilizzate
' M1,M2,C1,C2,C3
'arrivato il comando e lo analizzo nella stringa C2$
M5= 2 'comando da eseguire QX2
'inizializzo
M1=0   'valore del comando da eseguire
C1$="" 'send data
M2 = 0 'bool per capire se il FlexiBowl è in movimento
M3 = 0 'recive data
C2$=Str$(M5)
C3$="" 'STRINGA DI USCITA
*OPEN_TCP
Open "COM3:" As #3
If M_Open(3)<>1 Then *OPEN_TCP
'catturo il valore
M1 = Val(C2$)
'QX2 Forward

```

```
If(M1=2) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(50)
  M2 = 1
EndIf
'QX3 Forward Flip
If(M1=3) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(51)
  M2 = 1
EndIf
'QX4 Forward Flip Blow
If(M1=4) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(52)
  M2 = 1
EndIf
'QX5 Forward Blow
If(M1=5) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(53)
  M2 = 1
EndIf
'QX6 Shake
If(M1=6) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(54)
  M2 = 1
EndIf
'QX7 Light On
If(M1=7) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(55)
  M2 = 1
EndIf
'QX8 Light Off
If(M1=8) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(56)
  M2 = 0
EndIf
'QX9 Blow
If(M1=9) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(57)
  M2 = 0
EndIf
'QX10 Flip
If(M1=10) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(58)
  M2 = 1
EndIf
'QX11 Empty_Flb
If(M1=11) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(59)
  M2 = 1
EndIf
'QX12 Reset Allarm
If(M1=12) Then
  C1$= Chr$(0)+ Chr$(7)+Chr$(81)+Chr$(88)+Chr$(60)
  M2 = 0
EndIf
```

```
'invio il comando al flexibowl
Print #3, C1$
'leggo la risposta
Input #3, C20$
C2$=Mid$(C20$,4,1)
M3= Asc(C2$)
If(M3=37)Then
  If(M2=1)Then
    'wait move
    *WaitMove
    C1$= Chr$(0)+ Chr$(7)+Chr$(73)+Chr$(79)
    'invio il comando al flexibowl
    Print #3, C1$
    'leggo la risposta
    Input #3, C20$
    C2$=Mid$(C20$,14,1)
    M3= Asc(C2$)
    If(M3=48) Then 'se è uguale a 0 continua ad aspettare
      GoTo *WaitMove
      C3$="Done"
    EndIf
  else
    'gli passo la risposta
    C3$ = C2$
  EndIf
Else
  'gli passo la risposta
  C3$ = C2$
EndIf
Close #3
```

In pratica viene impostato nella variabile M5 il valore del comando da chiamare, solo numerico senza lettere QX, sotto viene riportata la tabella dei possibili comandi

| Action | Description |
|-----------------------|---|
| MOVE | Moves the feeder the current parameters. |
| MOVE-FLIP | Moves the feeder and activates Flip simultaneously |
| MOVE-BLOW-FLIP | Moves the feeder and activates Flip and blow simultaneously |
| MOVE-BLOW | Moves the feeder and activates Flip simultaneously |
| SHAKE | Shakes the feeder with the current parameters |
| LIGHT ON | Light on |
| LIGHT OFF | Light off |
| FLIP | Flip |
| BLOW | Blow |
| QUICK_EMPTYING | Quick Emptying Option |
| RESET_ALARM | Reset Alarm and enable the motor |

| Command | Description |
|-------------|-----------------------|
| QX2 | Move |
| QX3 | Move - Flip |
| QX4 | Move - Blow - Flip |
| QX5 | Move - Blow |
| QX6 | Shake |
| QX7 | Light on |
| QX8 | Light off |
| QX9 | Flip |
| QX10 | Blow |
| QX11 | Quick Emptying Option |
| QX12 | Reset Alarm |

Una volta settato il numero di comando da inviare, viene tradotto nel messaggio Char da inviare al flexibowl

Sending Commands to a Drive

An eSCL UDP packet consists of three parts, the header (binary 07), the SCL string (a sequence of ASCII encoded characters) and the SCL terminator (ASCII carriage return, 13)

| header | SCL string | <cr> |
|--------|------------|------|
|--------|------------|------|

Example: Sending "RV"

- SCL Header = 07 (two bytes)
- R = ASCII 82
- V = ASCII 86
- <cr> (ASCII carriage return) = 13

| header | | "RV" | | <cr> |
|--------|---|------|----|------|
| 0 | 7 | 82 | 86 | 13 |

Preparata la stringa da inviare viene spedita, se la stringa spedita ha un movimento in automatico il software grazie alla funzione WaitMove attenderà il completamento del movimento (C3\$="Done"), mentre se la stringa spedita è un'interrogazione, tipo "AL", verrà restituita nella stringa C3\$ l'Echo del Flexibowl.