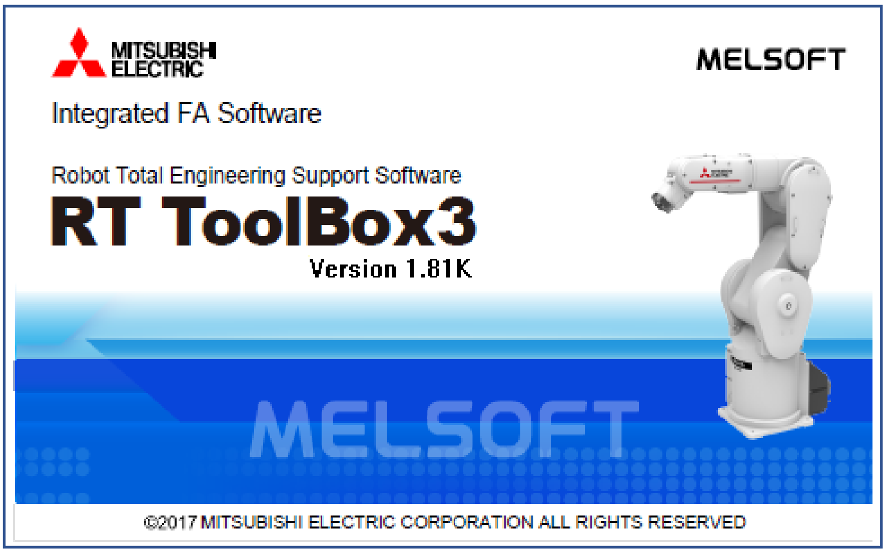
MITSUBISHI FLEXIBOWL PLUGIN

This Plugin was developed with the idea of communicating quickly and safely with FlexiBowl® through Mitsubishi robots, by using the software RT TOOLBOX ver. 1.81k.
The Plugin does NOT require an additional Mistubishi licenses.


MITSUBISHI ELECTRIC

MELSOFT

Integrated FA Software

Robot Total Engineering Support Software

RT ToolBox3

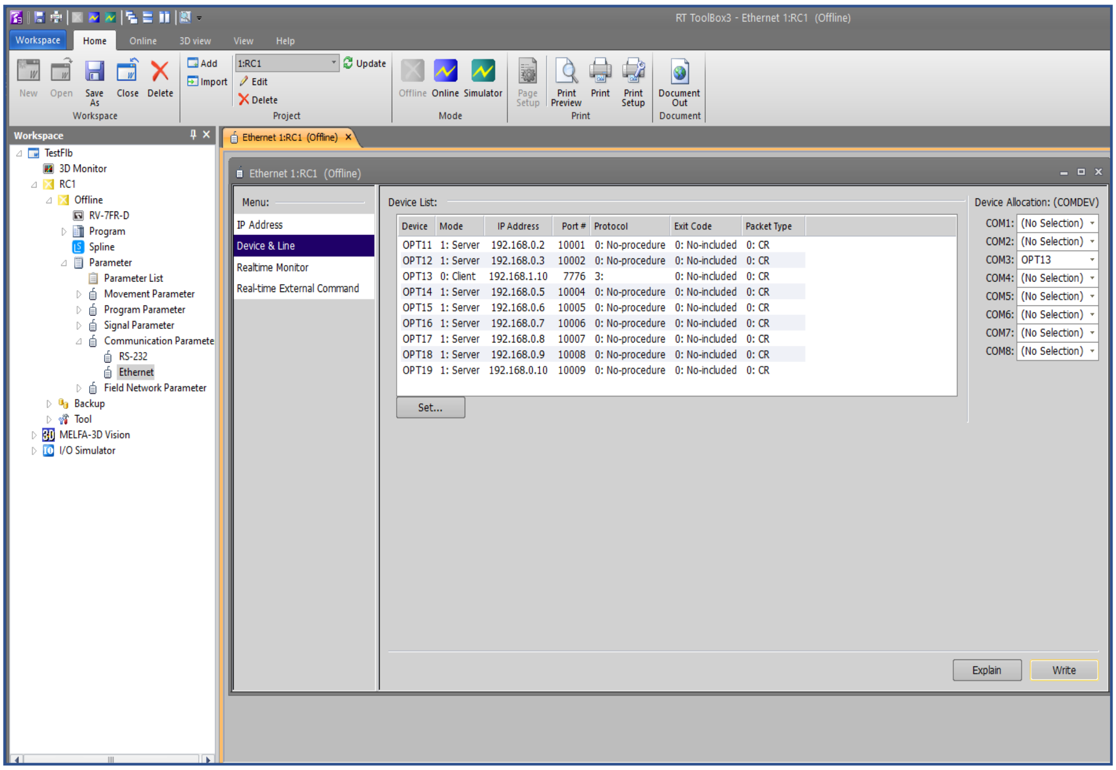
Version 1.81K

MELSOFT

©2017 MITSUBISHI ELECTRIC CORPORATION ALL RIGHTS RESERVED

RT TOOLBOX

STEP 1:



Device List:

Device	Mode	IP Address	Port #	Protocol	Exit Code	Packet Type
OPT11	1: Server	192.168.0.2	10001	0: No-procedure	0: No-included	0: CR
OPT12	1: Server	192.168.0.3	10002	0: No-procedure	0: No-included	0: CR
OPT13	0: Client	192.168.1.10	7776	3:	0: No-included	0: CR
OPT14	1: Server	192.168.0.5	10004	0: No-procedure	0: No-included	0: CR
OPT15	1: Server	192.168.0.6	10005	0: No-procedure	0: No-included	0: CR
OPT16	1: Server	192.168.0.7	10006	0: No-procedure	0: No-included	0: CR
OPT17	1: Server	192.168.0.8	10007	0: No-procedure	0: No-included	0: CR
OPT18	1: Server	192.168.0.9	10008	0: No-procedure	0: No-included	0: CR
OPT19	1: Server	192.168.0.10	10009	0: No-procedure	0: No-included	0: CR

Device Allocation: (COMDEV)

COM1:	(No Selection)
COM2:	(No Selection)
COM3:	OPT13
COM4:	(No Selection)
COM5:	(No Selection)
COM6:	(No Selection)
COM7:	(No Selection)
COM8:	(No Selection)

Consider FlexiBowl having 192.168.1.10 as Ip address, set a communication port as in the image.

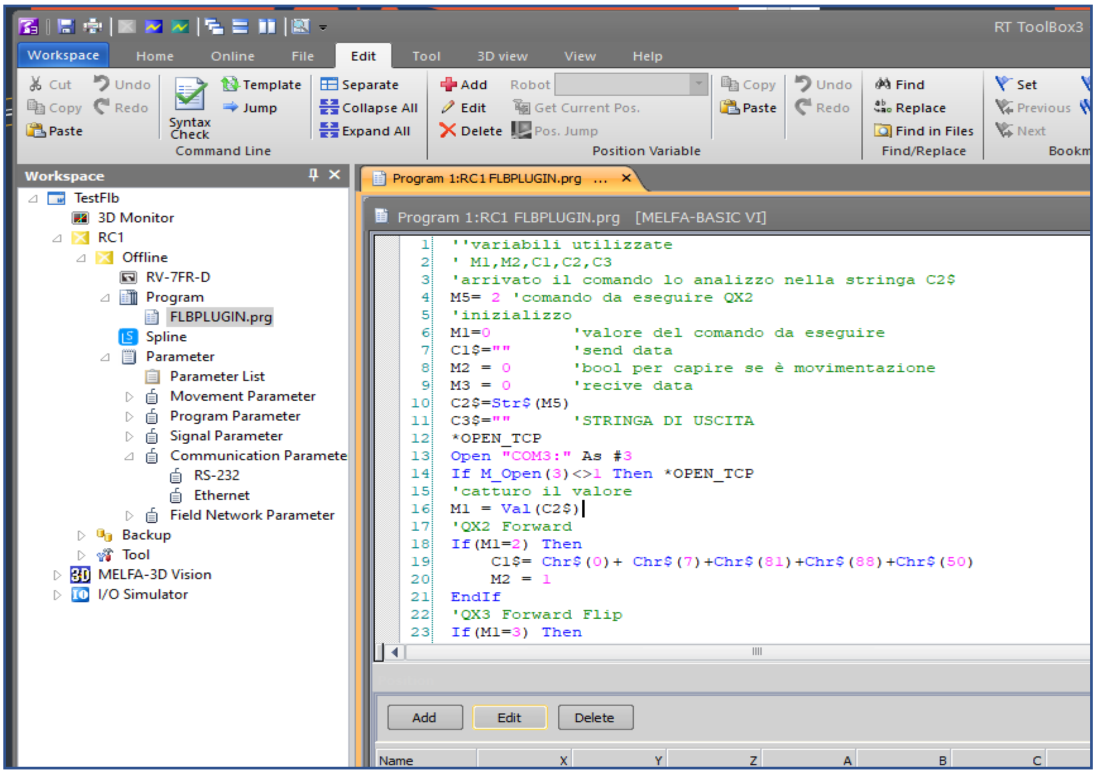
Port 7775

In the image the Ip and the port are set and then set the Com3 as OPT13.

Once the Device & Line is set, we are ready to make the communication with FlexiBowl.

STEP 2:

Here the code used to move FlexiBowl:



```
''variabili utilizzate
' M1,M2,C1,C2,C3
'arrivato il comando lo analizzo nella stringa C2$
M5= 2 'comando da eseguire QX2
'inizializzo
M1=0        'valore del comando da eseguire
C1$=""      'send data
M2 = 0      'bool per capire se è movimentazione
M3 = 0      'recive data
C2$=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
```

```
"used variables
' M1,M2,C1,C2,C3
'once the command arrives, analize it in the string C2$
M5= 2 'command to execute QX2
'initialize
M1=0       'value of the command to execute
C1$=""     'send data
M2 = 0     'bool to understand if the status is “moving”
M3 = 0     'recive data
C2$=Str$(M5)
C3$=""     'OUTPUT STRING
*OPEN_TCP
Open "COM3:" As #3
If M_Open(3)<>1 Then *OPEN_TCP
'catch the value
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
```

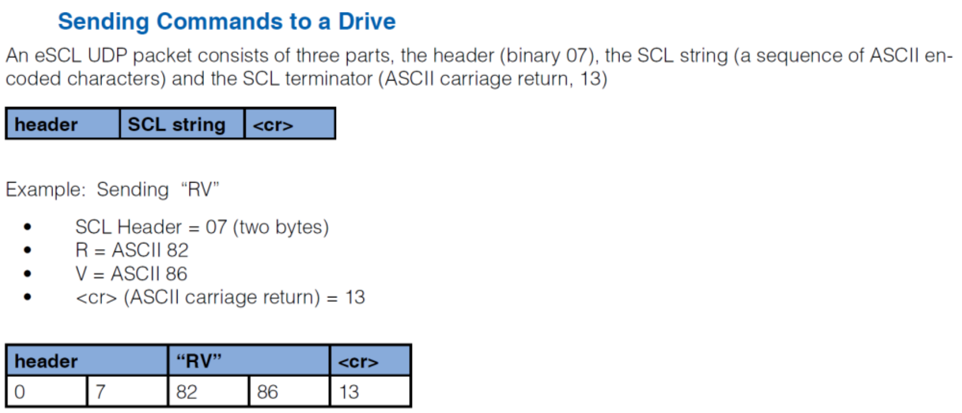
```
'send the command to FlexiBowl
Print #3, C1$
'read the answer
Input #3, C20$
C2$=Mid$(C20$,4,1)
M3= Asc(C2$)
If(M3=37)Then
    If(M2=1)Then
        'wait move
        *WaitMove
        C1$= Chr$(0)+ Chr$(7)+Chr$(73)+Chr$(79)
        'send the command to FlexiBowl
        Print #3, C1$
        'read the answer
        Input #3, C20$
        C2$=Mid$(C20$,14,1)
        M3= Asc(C2$)
        If(M3=48) Then  'if it is 0, keep on waiting
            GoTo *WaitMove
            C3$="Done"
        EndIf
       else
        'pass on the reply
        C3$ = C2$
       EndIf
 Else
        'pass on the reply
        C3$ = C2$
 EndIf
Close #3
```

The command to be executed is set in the M5 variable, numbers only, no QX letters. Below is a chart of the possible commands:

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_EMPTING	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

Once the number of the command to send is set, it is translated in the Char message to send to 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

Once the string is ready, it is sent to FlexiBowl; if the sent string has an automatic movement, thanks to the WaitMove function, the software will wait for the completion of the movement (C3$="Done"). While, if the sent string is an interrogation, such as "AL", the Echo of FlexiBowl will be returned in the C3$ string.