ABB FLEXIBOWL PLUGIN

Questo Plugin è nato con l'idea di comunicare in maniera rapida e sicura con il FlexiBowl® tramite i robot ABB, mediante l' utilizzo di instruzioni in linuaggio RAPID.
Il Plugin necessita di una licenza aggiuntiva per la gestione dei socket:
-Pc interface.
Può essere utilizzata anche una licenza opzionale:
-Multi tasking
Per poter effettuare il controllo e l' attivazione della tramoggia in un task parellolo senza dover bloccare il ciclo principale del robot.

FlexiBowl®

RobotStudio®

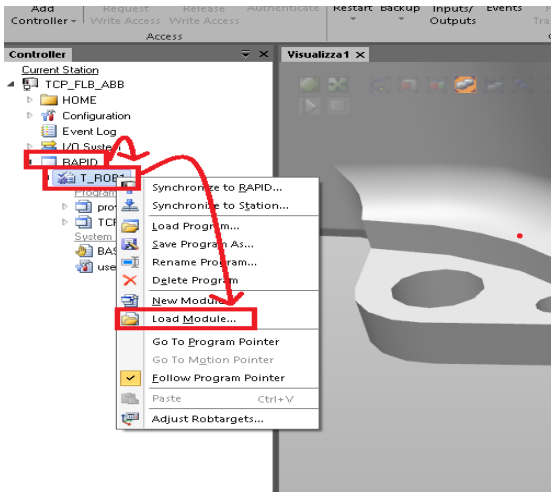
Informazioni su RobotStudio

RobotStudio 2019.2 (A 64 bit)
Versione 7.0.8504.0151

Supports RobotWare 5.06 to 6.09

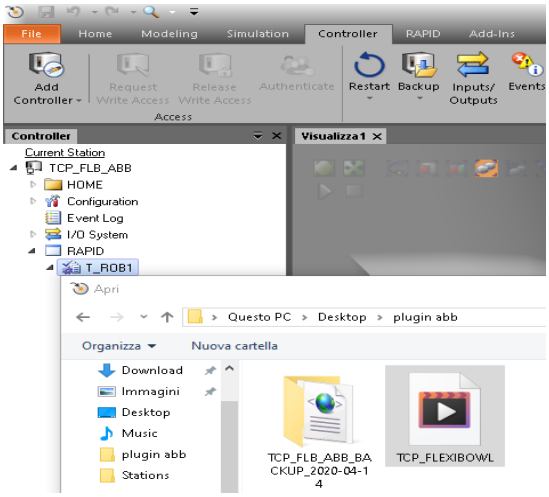
RobotStudio

STEP 1:



Nel menù ***Controller*** selezionare: ***Rapid→T_ROB1***
Premere con il tasto destro su ***T_ROB1*** e selezionare ***Load Module***

STEP 2:



Nella finestra di dialogo che apparirà , selezionare il Plugin ***TCP_FLEXIBOWL*** fornito da ARS.

STEP 3:

```
4
5       var string returnflb;
6       returnflb:= TCP_FLB("192.168.0.161","QX2
7
```

Dopo aver importato il Plug-in del FlexiBowl. Ci basterà aggiungere queste due semplici linee di codice nel nostro programma principale per poterlo controllare in ogni sua funzione.

RobotStudio

STEP 4:

```
5   var string returnflb;
6   returnflb:= TCP_FLB("192.168.0.161","QX2")
```

Dovremo quindi dare alla funzione ***TCP_FLB*** "***IP***" e "***COMANDO***" come argomenti

STEP 5:

```
returnflb:= TCP_FLB
```

La funzione ***TCP_FLB***, fornirà in output una stringa contenente :
- "***done***" se il comando inviato era un comando di movimento.
-***"\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

RobotStudio

SCRIPT:

```
MODULE TCP_FLEXIBOWL
!Dichiarazione variabili globali
!Global variables declaration
    VAR string Return_From_FLB;
    VAR socketdev server_socket;

    func string  TCP_FLB(string ip,string command)

        !Dichiarazione variabili locali
        !Local variables declaration
        VAR num provatest;
        VAR string TCP_IP;
        VAR num TCP_port;

        !chiusura socket aperto precedentemente
        !closing the previus socket instance
        SocketClose server_socket;

        !creazione socket
        !creating new socket
        SocketCreate server_socket;

        !definisco ip e porta(porta standard tcp=7776 / porta standard udp=7775)
        !define ip and port (stardard tcp port= 7776 /standard udp port=7775)
        TCP_IP:=ip;
        TCP_port:=7776;
        Return_From_FLB:="";

       !Trovo i caratteri minuscoli e li trasformo in maiuscoli
       !Convert all Lower Char in Upper Char
       command := StrMap(command,STR_LOWER, STR_UPPER);

        !chiamo la procedura di connessione
        !call connection proc
        Connection_FLB TCP_IP,TCP_port;

        !chiamo la progedura di preparazione ed invio del comando
        !call sending formatted command proc
        Send_FLB command;

        !Se è un comando di movimento(QX...) aspetto fino alla fine del movimento
        !If was a movement command (QX...) will wait untill the end of the movement
        IF ( StrFind(Return_From_FLB,1,"%")<= strlen(Return_From_FLB)) THEN

          !aspetto che il movimento sia finito e inserisco "done"come valore di ritorno
          !wait untill the movements end and store "done" as return from FlexiBowl
          IF ( StrFind(command,1,"Q")<= strlen(Return_From_FLB)) THEN
             Wait_No_Move;
             Return_From_FLB:="done";
          ENDIF

        ENDIF

        !Ritorno "done" se il comando era di movimento mentre ritorno la risposta del FlexiBowl se il comando era un instruzione
        !Return "done" if the command was a movement or return the answare from FlexiBowl if was an instruction
        RETURN Return_From_FLB;

    endfunc
```

RobotStudio

SCRIPT:

```
!*****************************************************************************************************

    !tento la connessione ed intercetto le eccezioni
    !handle the connection and catch rising exceptions
    PROC Connection_FLB(string ip,num port)


        SocketConnect server_socket, ip, port;

    ERROR

        !Se si verifica un errore in fase di connessione chiudo il socket e scrivo sulla pendant
        !If an error occours during the connection close the socket and write on the tp
        SocketClose server_socket;
        TPWrite("Connection Problems");

    ENDPROC

!*****************************************************************************************************

    !preparo la stringa e la mando al flb
    !format and send the command string to flb
    PROC Send_FLB(string command)

        !preparo il messaggio aggiungendo l' header del pacchetto ( chr0 chr7)  e il carattere di fine riga (chr13)
        !format the message adding the header (chr0 chr7) and the end of line character (chr13)
        SocketSend server_socket \Str := "\00\07"+command+"\0D";

        !invio al FlexiBowl
        !send to FlexiBowl
        SocketReceive server_socket \Str := Return_From_FLB;

     ERROR

        !Se si verifica un errore in fase di connessione chiudo il socket e scrivo sulla pendant
        !If an error occours during the connection close the socket and write on the tp
        SocketClose server_socket;
        TPWrite("Connection Problems");

    ENDPROC
```

RobotStudio

SCRIPT:

```
!***********************************************************************************************************
!aspetto la fine del movimento
!wait untill the movement finish
PROC Wait_No_Move()

    VAR bool ok;
    VAR string moving;
    VAR num    int_moving;
    moving :="1";
    WaitTime(0.1);

    WHILE TRUE DO

        !mando il comando SC "statu controll"
        !send the "status controll" command SC
        Send_FLB "SC";

        !la risposta sarà del tipo : (chr0)(chr7)SC=abcd(chr13) andremo quindi ad controllare il bit "c" per sapere se il movimento è terminato
        !the standard answare will be like : (chr0)(chr7)SC=abcd(chr13) we will controll the "c" bit to know if the movement is terminated
        moving:=StrPart(Return_From_FLB,7,1);

        !trasformo la stringa in intero e la controllo
        !cast string into integer and then controll it
        ok := StrToVal(moving,int_moving);

        IF (int_moving=0) THEN

            WaitTime(0.1);
            Return_From_FLB:="done";
            RETURN;

        ENDIF

        WaitTime(0.1);

    ENDWHILE

    Return_From_FLB:="done";


  ENDPROC


!***********************************************************************************************************

ENDMODULE
```