KAWASAKI ROBOTICS IDE

KAWASAKI FLEXIBOWL PLUGIN

This Plugin was developed with the idea of communicating quickly and safely with FlexiBowl® through KAWASAKI robots, by using instructions in AS.

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

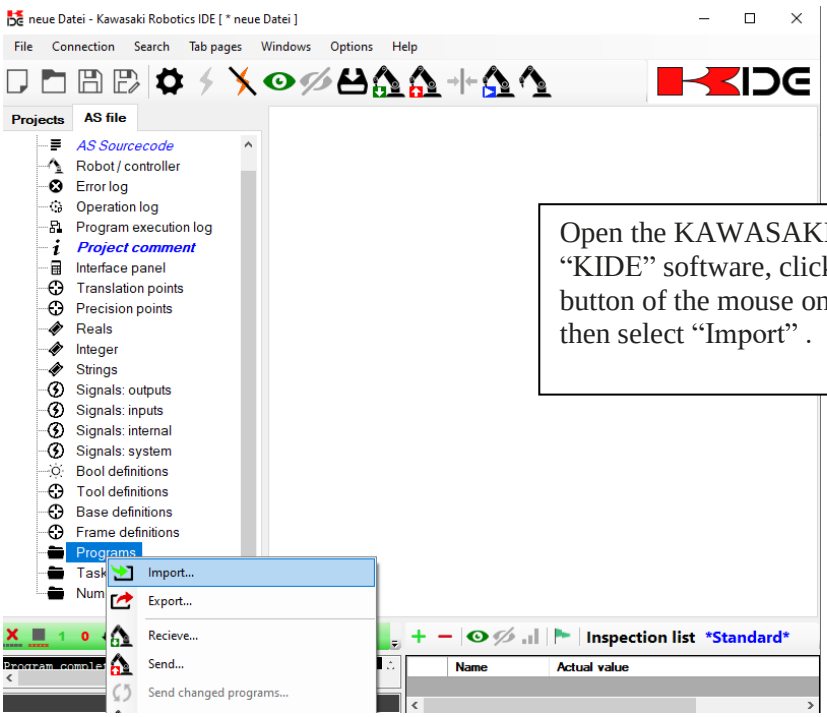
FlexiBowl®

Kawasaki Robotics

IDE Kawasaki Robotics

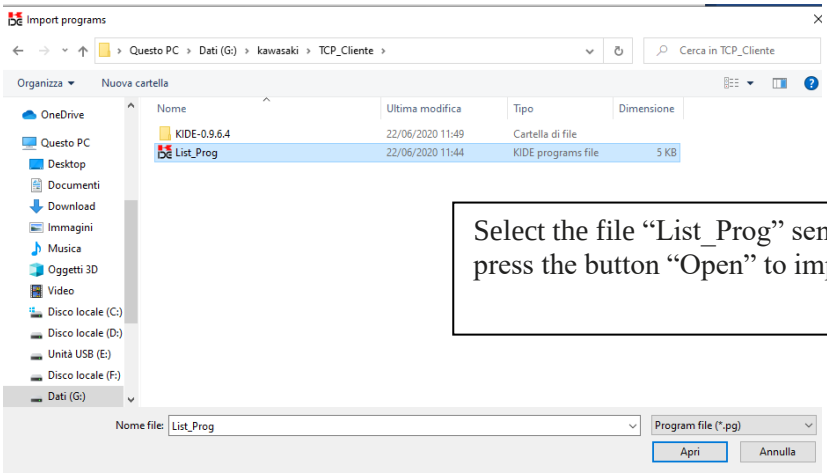
KAWASAKI ROBOTICS IDE

STEP 1:



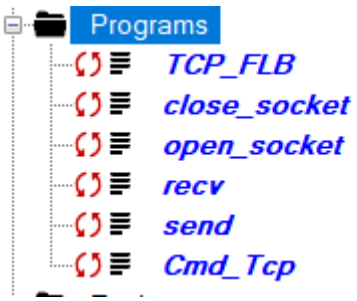
Open the KAWASAKI ROBOTICS IDE "KIDE" software, click with the right button of the mouse on "Programs" and then select "Import" .

STEP 2:



Select the file "List_Prog" sent by ARS and press the button "Open" to import the files.

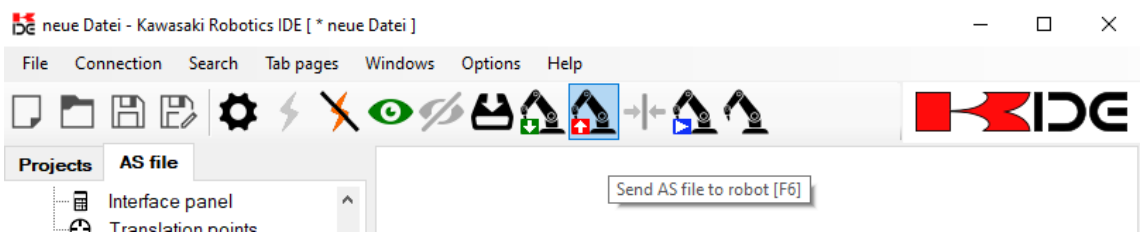
STEP 3:



The programs that are now transferred in the project, must be transferred to the robot contoller.
Then connect the "KIDE" program with the controller.

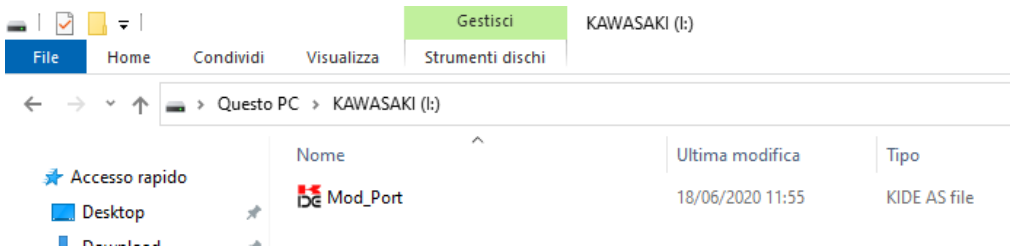
KAWASAKI ROBOTICS IDE

STEP 4:



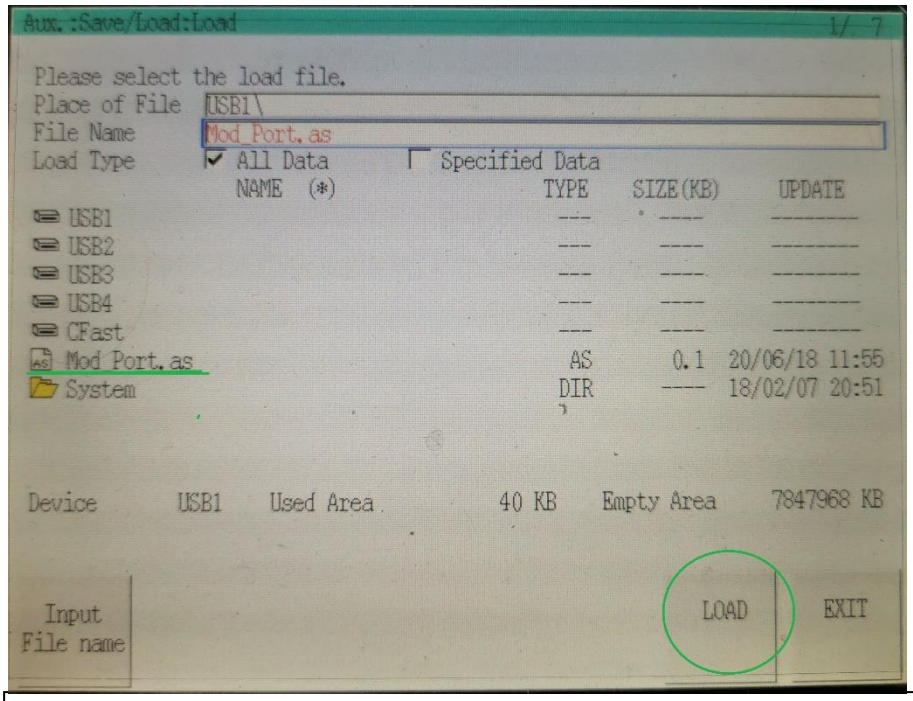
Transfer the programs in the memory of the Kawasaki controller.

STEP 5:



Before being able to use the programs that were transferred, it will be necessary to unlock the communication ports, starting from the 1024.
Select the file “Mod_Port” sent by ARS and copy it on an empty USB memory.

STEP 6:



Insert the USB memory in the USB slot present on the Kawasaki controller and upload the file from the menu AUX→Save/Load→ Load
Confirm the transfer if required by the system.

KAWASAKI ROBOTICS IDE

STEP 7

```
0  .PROGRAM Cmd_Tcp ()
1    ; ************************************************************
2    ;
3    ; Program:      Cmd_Tcp
4    ; Comment:
5    ; Author:       ARS AUTOMATION
6    ;
7    ; Date:         6/16/2020
8    ;
9    ; ************************************************************
10   CALL TCP_FLB ("192.168.1.10", "QX2", .$returnstring)
11   PRINT .$returnstring
12
13
14 .END
```

Specify the **IP address** of the Flexibowl as the first argument of the FLB_TCP program, and **the command** you want to send as the second argument.

The "TCP_FLB" program will provide as output a string "**.$returnstring)**" containing:

- **"*Done*"** if the command sent was a movement command.
-**"Error"** if an error occurred.
- ***"risposta dal FlexiBowl"*** if a query command is sent to the driver.

Es.
Command sent= "\00" "\07" "SC" "\0D"
Reply from the FlexiBowl = *"SC=0001"*

COMMAND STRING FORMAT:

Correct syntax for each datagram			
Header		**Command**	**Footer**
Chr(0)	Chr(7)	Comando	Chr(13)

COMMAND LIST:

Commands	Description
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

KAWASAKI ROBOTICS IDE

SCRIPT: TCP_FLB

```
.PROGRAM TCP_FLB (.$ip, .$command, .$retstr);
  ; ************************************************************************
  ;
  ; Program:      TCP_FLB
  ; Comment:
  ; Author:       ARS AUTOMATION
  ;
  ; Date:         6/15/2020
  ;
  ; ************************************************************************
  ;'Dichiaro le variabili
  ;Define variables
  port = 7776
  max_length = 255
  tout_open = 60
  tout_rec = 60
  eret = 0
  ret = 0
  text_id = 0
  tout = 60
  .$retstr = ""
  $temp = ""
  flbready = -1

  ;Define ip address
  ;Dichiaro l'indirizzo IP
  FOR i = 1 TO 3
    $tmp_str = $DECODE (.$ip, ".", 0)
    ip[i] = VAL ($tmp_str)
    $tmp_str = $DECODE (.$ip, ".", 1)
  END
  ip[4] = VAL (.$ip)

  ;Chiudo il socket se è aperto
  ;Normal socket closure
  TCP_STATUS .num_sock, .port[1], .sock[1], .err[1], .serr[1], .$ip_adr[1]
  IF .num_sock > 0 THEN
    FOR i = 1 TO .num_sock
      IF (.port[i] == 7776) THEN
        TCP_CLOSE ret, .sock[i]
      END
    END
  END

  ;Connecting communication
  ;Apro socket per la comunicazione
  CALL open_socket
  IF sock_id < 0 THEN
    GOTO exit_NoConnection
  END

;Send the command to Flexibowl
  ;Invio il comando al Flexibowl
  CALL send (ret, .$command)
```

SCRIPT:
TCP_FLB

```
  ;Read the answer
  ;Leggo la risposta
  CALL recv (.numbytes)
  IF ret < 0 THEN
    PRINT "Communication error code=", ret
    GOTO exit
  END

  ;Check the answer
  ;Controllo la risposta
  $temp = $LEFT (.$command, 1)
  PRINT "TEMP=", $temp
  IF (($recv_buf[.numbytes - 1] == "%") AND ($temp == "Q")) THEN
    ;Entro in un ciclo while fino a che la risposta non identifica la fine del movimento
    ;Wait in a loop cycle until the answer identify the end of the movement
    WHILE (flbready <> 1) DO
      ;Send the command to Flexibowl
      ;invio comando al Flexibowl
      CALL send (ret, "SC")

      ;Read the answer
      ;Leggo la risposta
      CALL recv (.numbytes)
      IF ret < 0 THEN
        PRINT "Communication error code=", ret
        GOTO exit
      END

      ;Check the value of the less significant byte
      ;Controllo il valore del byte meno signaficativo
      flbready = VAL ($recv_buf[9])

      DELAY 0.05
    END
    ;Movimento terminato
    ;Movement completed
    .$retstr = "Done"

  ELSE
    ;Istruzione non di movimento
    ;No-Movement istruction
    FOR i = 3 TO (.numbytes - 1)
      .$retstr = .$retstr + $recv_buf[i]
    END

  END

  ;Disconnecting communication
  ;Chiudo la connessione
  CALL close_socket

  ;Return to main program
  ;Esco dal programma e torno al programma principale
  RETURN
exit:
  ;Disconnecting communication
  ;Chiudo la connessione
  CALL close_socket

exit_NoConnection:
  .$retstr = "Error"
.END
```

KAWASAKI ROBOTICS IDE

SCRIPT: close_socket

```
.PROGRAM close_socket ();Closing communication
        TCP_CLOSE ret, sock_id;Normal socket closure
        IF ret < 0 THEN
                PRINT "TCP_CLOSE error ERROE=(",ret," )", $ERROR (ret)
                TCP_CLOSE ret1, sock_id;Forced closure of socket (shutdown)
                IF ret1 < 0 THEN
                        PRINT "TCP_CLOSE error id=", sock_id
                END
        ELSE
                PRINT "TCP_CLOSE OK id=", sock_id
        END
.END
```

SCRIPT: open_socket

```
.PROGRAM open_socket ();Starting communication
        .er_count = 0
        tout_open = 2
connect:
        TCP_CONNECT sock_id, port, ip[1], tout_open
        IF sock_id < 0 THEN
                IF .er_count >= 5 THEN
                        sock_id = -1
                        PRINT "Connection with PC failed. Program is stopped."
                ELSE
                        .er_count = .er_count + 1
                        PRINT "TCP_CONNECT error id=", sock_id, " error count=", .er_count
                        GOTO connect
                END
        ELSE
                PRINT "TCP_CONNECT OK id=", sock_id
        END
.END
```

SCRIPT: send

```
.PROGRAM send (.ret, .$data);Communication Sending data
        $send_buf[1] = $CHR (0)
        $send_buf[2] = $CHR (7) + .$data + $CHR (13)
        buf_n = 2
        .ret = 1
send_rt:
        TCP_SEND sret, sock_id, $send_buf[1], buf_n, tout
        IF sret < 0 THEN
                .ret = -1
                PRINT "TCP_SEND error in SEND", sret
        ELSE
                PRINT "TCP_SEND OK in SEND", sret
        END
.END
```

KAWASAKI ROBOTICS IDE

SCRIPT: recv

```
.PROGRAM recv (.numbytes); Communication Receiving data
        .numbytes = 0
        max_length = 1
        tout_rec = 2
        ret = 0

        TCP_RECV ret, sock_id, $recv_buf[1], .numbytes, tout_rec, max_length
        IF ret < 0 THEN
                PRINT "TCP_RECV error in RECV", ret
                $recv_buf[1] = "000"
        ELSE
                IF .numbytes > 0 THEN
                        PRINT "TCP_RECV OK in RECV", ret
                        FOR i = 1 TO .numbytes
                                PRINT "RecBuff[", i, "]= ", $recv_buf[i]
                        END
                ELSE
                        $recv_buf[1] = "000"
                        ret = -1
                END
        END
.END
```

SCRIPT: Cmd_Tcp

```
.PROGRAM Cmd_Tcp ()
  ; *****************************************************************
  ;
  ; Program:      Cmd_Tcp
  ; Comment:
  ; Author:       ARS AUTOMATION
  ;
  ; Date:         6/16/2020
  ;
  ; *****************************************************************
  CALL TCP_FLB ("192.168.1.10", "QX2", .$returnstring)
  PRINT .$returnstring


.END
```