25 mei 2023

Aarts,Luuk L.H.K.

Fontys FHICT

The Igus Robot arm

communication with   
the igus robot

fontys industry project

# Contact base

The Igus robot arm has a contact protocol that it follows while using the ethernet port. In this case the IP address of the robot is 192.168.3.11 and the port that should be used when reaching out is port 3920. When reaching out to the robot arm it will send a return message, this indicates that a connection is established. In this case the connection should be kept alive by use of alive messages. The Igus robot expects at least one alive message every 2 seconds. This can be more, but not less, if this message is not sent, the connection will be closed and can be reestablished after 1 second.

## Message base

### Client message

The base of every message exists out of 5 parts send in the form of a String. Each capital written word won’t change each regular word will change depending on the goal of the message and timing.

The message looks like: “CRISTART counter CMD\_CATEGORY command\_details CRIEND”.

* CRISTART will not change, starts the message.
* The counter is set by middle of a value that goes from 1 to 9999 and then set back to 1. This value will grow by 1 with each message send.
* CMD\_CATEGORY will be changed into the command it is sending, like “ALIVEJOG”.
* Command\_details will form the data that is send with this command. In case of the alive message, this will be the jog values of all 6 joints and 3 gripper joints, these jog values will be written as floats with a range of -100.0 – 100.0 and will look like:

“10.0 20.0 30.0 40.0 50.0 60.0 -10.0 -20.0 -30.0”.

* CRIEND does not change either, ends the message.

### Server return message

The return message of the Robot arm (server) will be sent each time it receives a message from the PLC (client). Just as with the message from the client, the message starts with “CRISTART count” and ends with “CRIEND”. Count is the server count, which is not in contact with the client count. The message will contain all data like, error status and joint positions, sent in one long string. The result is a message that will look somewhat like this:

CRISTART 1234 STATUS MODE joint POSJOINTSETPOINT 1.00 2.00 3.00 .... 15.00 16.00 POSJOINTCURRENT 1.00 2.00 3.00 .... 15.00 16.00 POSCARTROBOT 10.0 20.0 30.0 0.00 90.00 0.00 POSCARTPLATFORM 10.0 20.0 180.00  
OVERRIDE 80.0  
DIN 0 DOUT 0  
ESTOP 3 SUPPLY 23000 CURRENTALL 2600 CURRENTJOINTS 150 200 ... 140 160  
ERROR no\_error 8 8 8 ... 8 8 8  
KINSTATE 3  
CRIEND

But then printed behind each other in a string.

The exact meaning of every value can be found on page 5 and 6 of the CPR\_RobotInterfaceCRI.pdf file that is linked on the Sources page of this document.

## Client commands

As handled in the earlier, will the message send by the client contain 5 parts combined into one String. The CMD\_CATEGORY should be changed into CMD when sending commands and the command\_details will contain the command and details of the command. The available commands are:

* “Connect” establishes the contact between the client and the server.
* “Disconnect” disconnects the client and the server.
* “Reset” Resets the connection between client and server.
* “Enable” enables the joints to move.
* “Disable” disables the joints to move.
* “SetJointsToZero” returns all joints to their zero (rest) position.

When sending commands, the Robot arm (server) will return an acknowledgement or an error message, like: “CRISTART sCnt CMDACK ref\_to\_cCnt CRIEND” in this case the sCnt is server count and cCnt is client count. The ref\_to\_cCnt will be the reference to the cCnt of the command message that is acknowledged. The error message,

like: “CRISTART sCnt CMDERROR ref\_to\_cCnt error\_description CRIEND”.

## Client Variables

To let the Robot arm move, the variables that represent the target position should be changed. This can be done with the CMD\_CATEGORY, “VAR” and the CMD\_DETAILS “Joint” combined with the values of the target location of the joints. Ending up looking like:

‘CRISTART 1 VAR Joint 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 CRIEND’

The second option is to move on the arms tool location, which will make the grabbing of an object much easier. To control these variables in the robot we need to set the ‘VAR’ combined with ‘Linear’ and the target values which would look like:

‘CRISTART 1 VAR Linear 10 10 10 CRIEND’