TECHNICAL NOTE

Omron-TM Sysmac FB Library

Listen node – Expression editor

ATC Europe
C/Josep Pla, 2, Edif. B2
Barcelona, 08019
T: +34 932 14 06 00

Reference	N/A
Doc. Rev.	V2.1
Date	February 2019

ATC Contact	Antonio Núñez
Mail	Antonio.nunez@eu.omron.com

Doc. version	Contents	Author	Reviewer
V1.0	Initial edition	Antonio Núñez	Josep Sánchez
V2.0	Structure Review	Antonio Núñez	Josep Sánchez
V2.1	FB Library Overview	Antonio Núñez	Antonio Núñez
V2.2	Document update	Antonio Núñez	Josep Sánchez
V2.3	Document update	Antonio Núñez	Josep Sánchez

INTRODUCTION

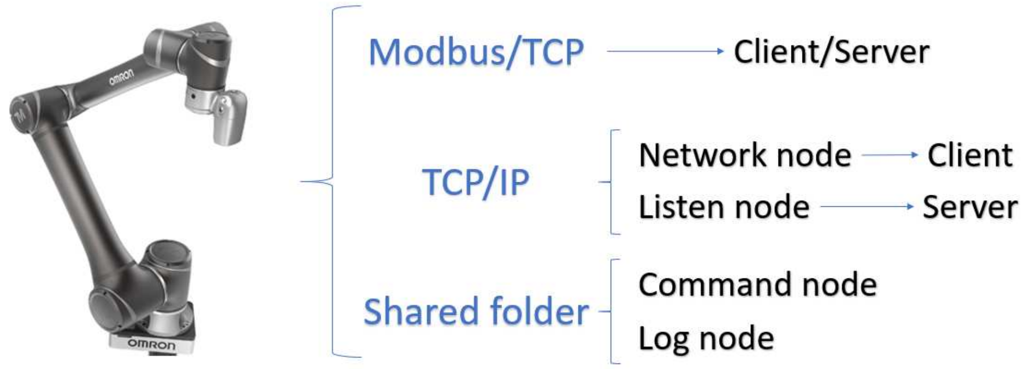
The purpose of the **Omron-TM Sysmac Library** *(TM_Commands_V1-01.slr)* is to provide a set of FBs in order to command the robot from an external NJ device through a TCP-IP connection.

The Omron-TM robot allows to establish a TCP/IP Socket Server connection through the “Listen Node”, which provides an specific protocol that allows to execute some of the functions available in the “Expression Editor” from TM Flow.

The Omron-TM Sysmac Library provides some of these “Expression Editor” functions through the FBs.

Omron-TM COMMUNICATIONS OVERVIEW

There are different ways of Ethernet communications from the Omron-TM robot point of view which are described as follows:



1. Modbus/TCP:

- ➢ Refer to Omron-TM Software manual.

2. TCP/IP:

- ➢ There are two possibilities of communication, as a client and as a server.
  - o In case that the Omron-TM robot is a client (Network Node) the PLC will accept the connection from the robot and it will request Read/Write data from/to the PLC.
  - o In case that the Omron-TM robot is a server (Listen Node) the PLC will establish the connection to the robot. In this way, PLC is able to send Expression Editor commands, such as Point Node movement (PTP, Line,…), Change base, Change TCP…

3. Shared folder:

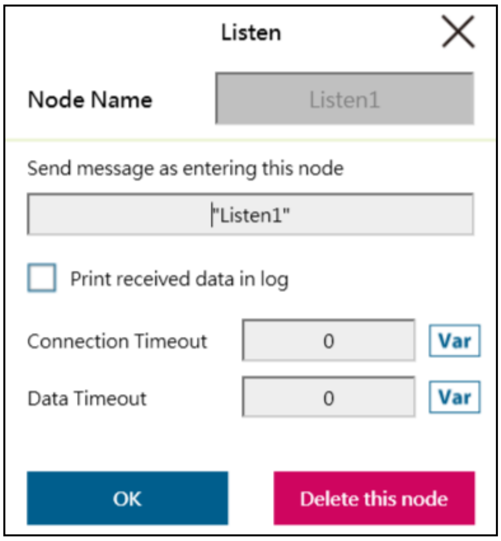
- ➢ Command Node: The File Command provides the user with operations of reading, modifying and deleting files in the shared folder.
- ➢ Log Node: The set variables and strings can be saved to the shared folder which has been created previously with this Node.

LISTEN Node

In the Listen Node, a TCP/IP server (Socket Server) can be established and be connected by an external device. Some of the functions which are available in “Expression Editor” can be executed in Listen Node.

Configuration screen

To configure the Listen Node, consider the following parameters:



Node Name: Listen1

Send message as entering this node: "Listen1"

□ Print received data in log

Connection Timeout: 0

Data Timeout: 0

- **Send message as entering this node:** When the flow processed in the TM Flow reaches this node, this message will be sent by the Server (TM).
- **Print Log:** Enable the Communication Log (shown on the right side).
- **Connection Timeout:** When entering this node, if more than the time set (in milliseconds) is not connected, it will be overtime. Use 0 to disable this Timeout.
- **Data Timeout:** When entering this node and connected, if no data is received during the time set (in milliseconds) it will be overtime. Use 0 to disable this Timeout.

Socket TCP Listener is started up after the project is executed and closed when it is stopped. The socket parameters are the following:

- IP: HMI → System → Network → IP Address.
- Port number: 5890.

The flow will keep at Listen node, until either of the two exit conditions is fulfilled.

- **Pass**: “ScriptExit()” is executed or the project is stopped.
- **Fail**: Connection Timeout, Data Timeout or in case the flow entered this Listen Node before the TCP Listener is started up.

There are two categories of commands:

1. First category is commands which can be accomplished in instance, like assigning variable value.

2. Second category is commands which need to be executed in sequence (motion command and IO value assigning). These commands will be placed in queue and executed in order.

Time parameters configuration

(Connection Timeout = 0) AND (Data Timeout = 0)

With this configuration, TM Flow will remain in Listen Node until an ‘ScriptExit’ command is received and the robot movement is finished.

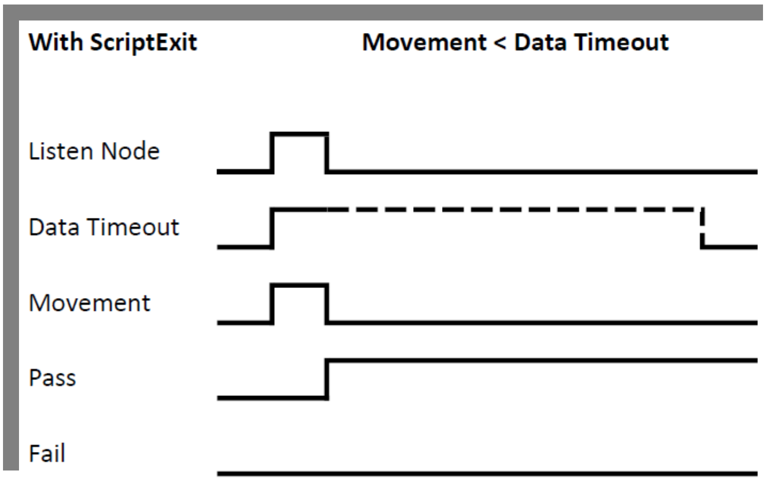
(Connection Timeout > 0) AND (Data Timeout = 0)

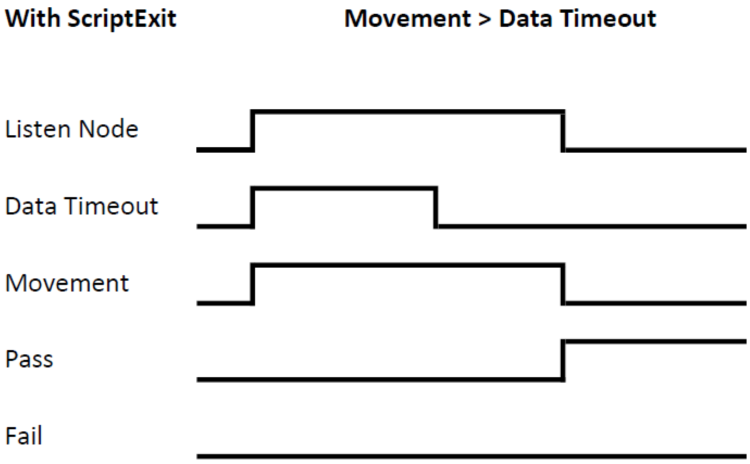
If there is no connection established with the robot before ‘Connection Timeout’ defined, the Listen Node will exit through ‘Fail’ path.

(Connection Timeout = 0) AND (Data Timeout > 0)

➢ With ‘ScriptExit’ command:

When the movement is finished the flow will continue through the ‘Pass’ path. The robot movement will be finished both when Movement is longer than the ‘Data Timeout’ value and when Movement is shorter than the ‘Data Timeout’ value.

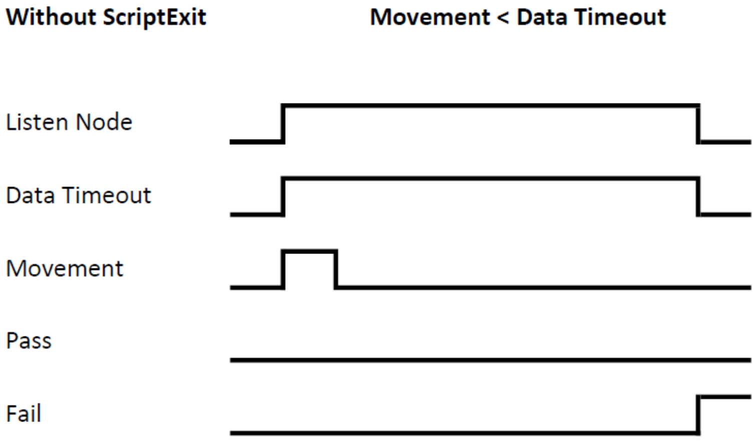


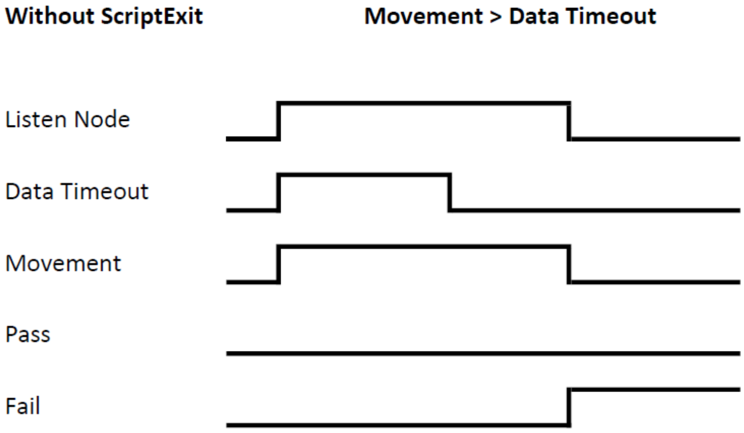


- Without ‘ScriptExit’ command:

When the movement is shorter that the ‘Data Timeout’ value, the flow will continue through ‘Fail’ path. The movement will be executed, afterwards, the robot will wait for the rest of the time (defined as ‘Data Timeout’) and the flow will continue through ‘Fail’ path.

When the movement is longer than the ‘Data Timeout’ value, after the movement is finished, the flow will continue through ‘Fail’ path.


Without ScriptExit **Movement < Data Timeout**

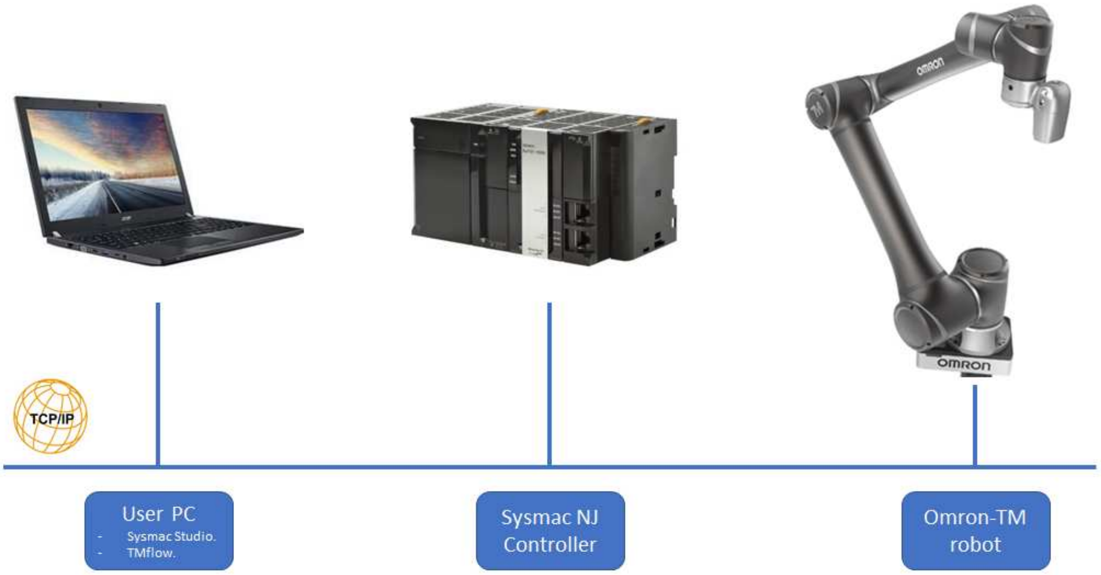

Without ScriptExit **Movement > Data Timeout**

Omron-TM/PLC SYSTEM SETUP

Device Configuration

The following picture shows a typical TM-PLC System Setup example, with the following components connected through TCP/IP:

- User PC (with Sysmac Studio V1.13 or later and TM Flow).
- Sysmac NJ controller.
- TM Robot.



Applicable Devices

Omron-TM Robot	HW 3.00
TM Flow	V1.68.6800
CPU	(Sysmac family)
Sysmac Studio	V1.13 or later

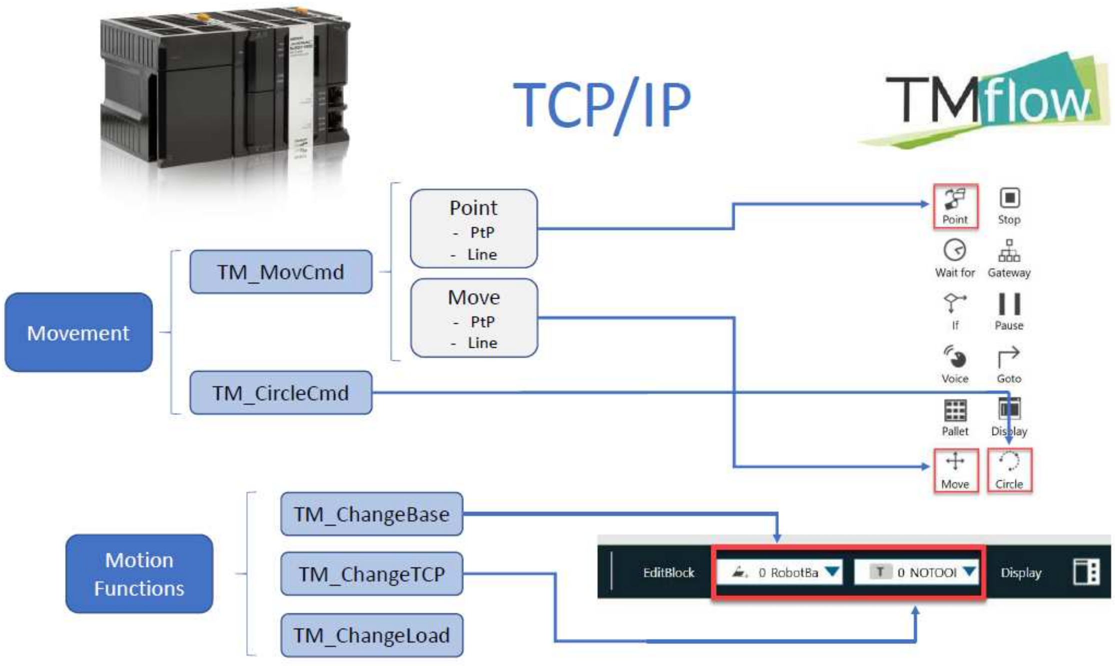
Omron-TM SYSMAC LIBRARY (TM_Commands_V1-01.slr)

Overview

There are two types of FBs depending on its purpose, and all of them have an equivalent function in TM Flow:

- Movement Commands
- Motion Function Commands

The following picture summarizes such equivalence:



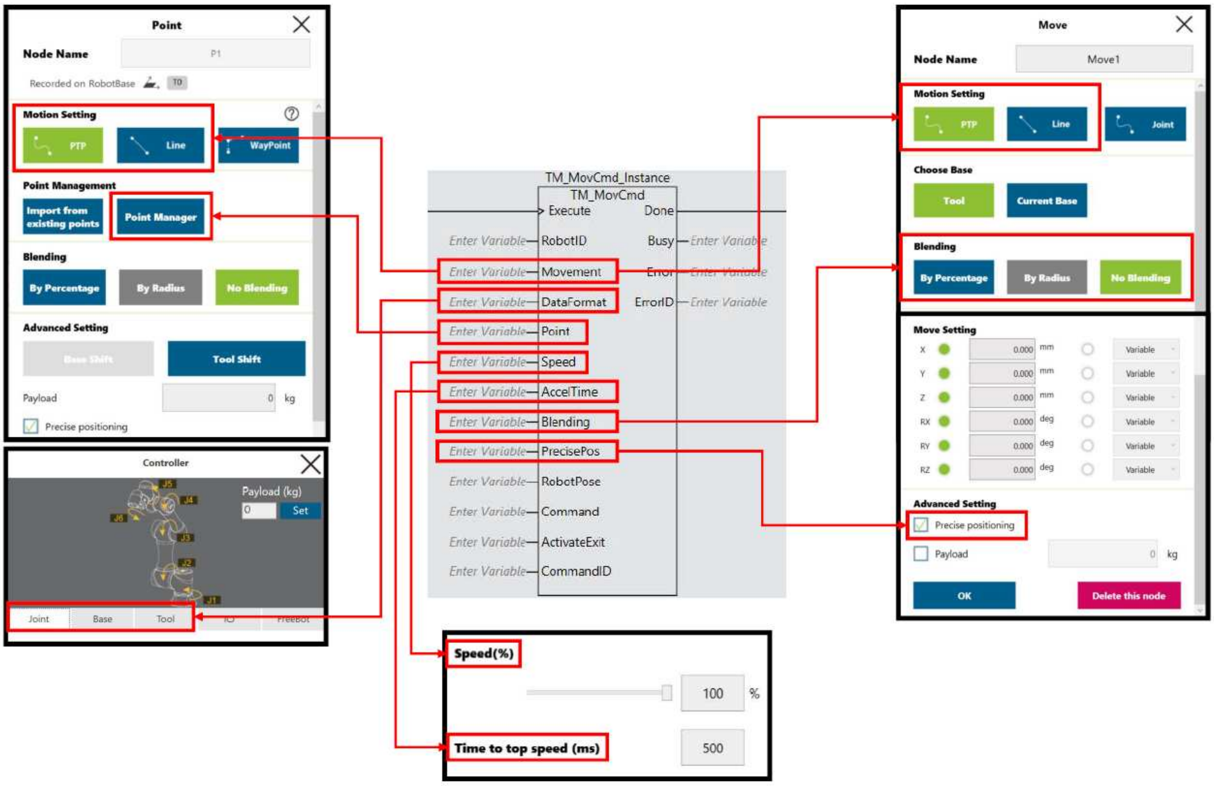
FB Structure

All the FBs provides the following structure regarding its input parameters:

- FB Command data: the required specific data to configure the corresponding FB command.
- Additional Command: allows to include an additional "Expression Editor" commad; this function can be use to write variables in TM Flow. This allows, for example, to control the flow sequence to execute in the robot.
- ActiveExit: This input allows to exit the Listen Node at the execution of the function corresponding to this FB, or to buffer several commands on it before to exit. This functionality allows to control the motion sequence of the robot, and the blending among them.

FB Command data Overview

In the following pictures, there is described the FB parameters equivalence with TMflow.



Point

Node Name: P1

Recorded on RobotBase T0

Payload: 0 kg

☑ Precise positioning

Controller

Payload (kg): 0

Move

Node Name: Move1

Move Setting

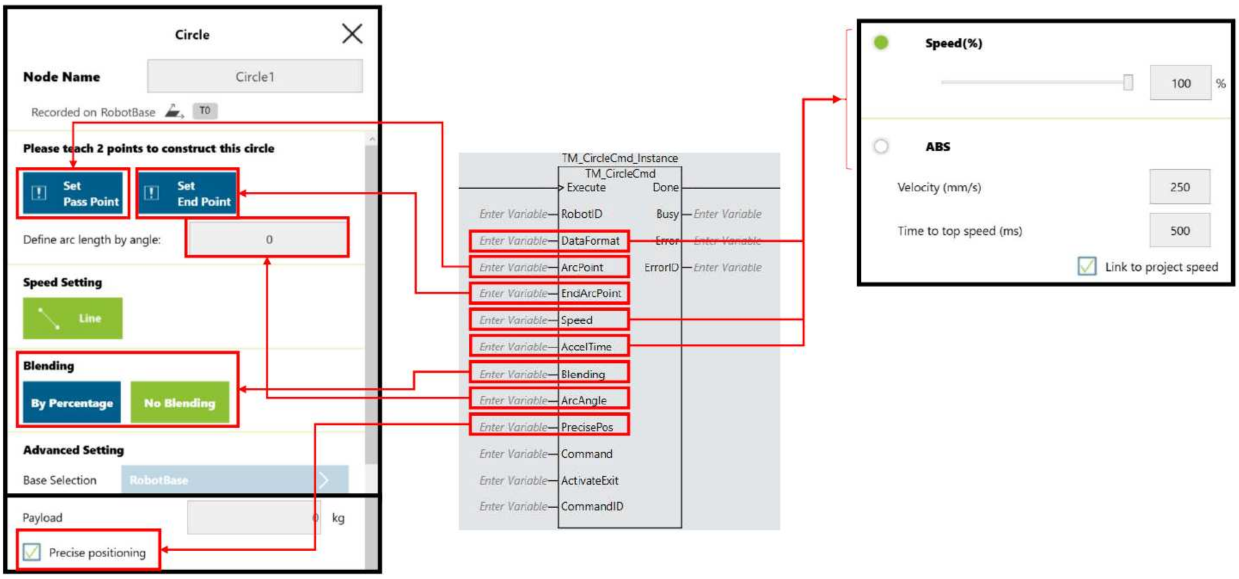
X	●	0.000	mm	○	Variable
Y	●	0.000	mm	○	Variable
Z	●	0.000	mm	○	Variable
RX	●	0.000	deg	○	Variable
RY	●	0.000	deg	○	Variable
RZ	●	0.000	deg	○	Variable

☑ Precise positioning

□ Payload: 0 kg

Speed(%): 100 %

Time to top speed (ms): 500



Circle

Node Name: Circle1

Recorded on RobotBase T0

Please teach 2 points to construct this circle

Define arc length by angle: 0

Base Selection: RobotBase

Payload: 0 kg

☑ Precise positioning

● Speed(%): 100 %

○ ABS

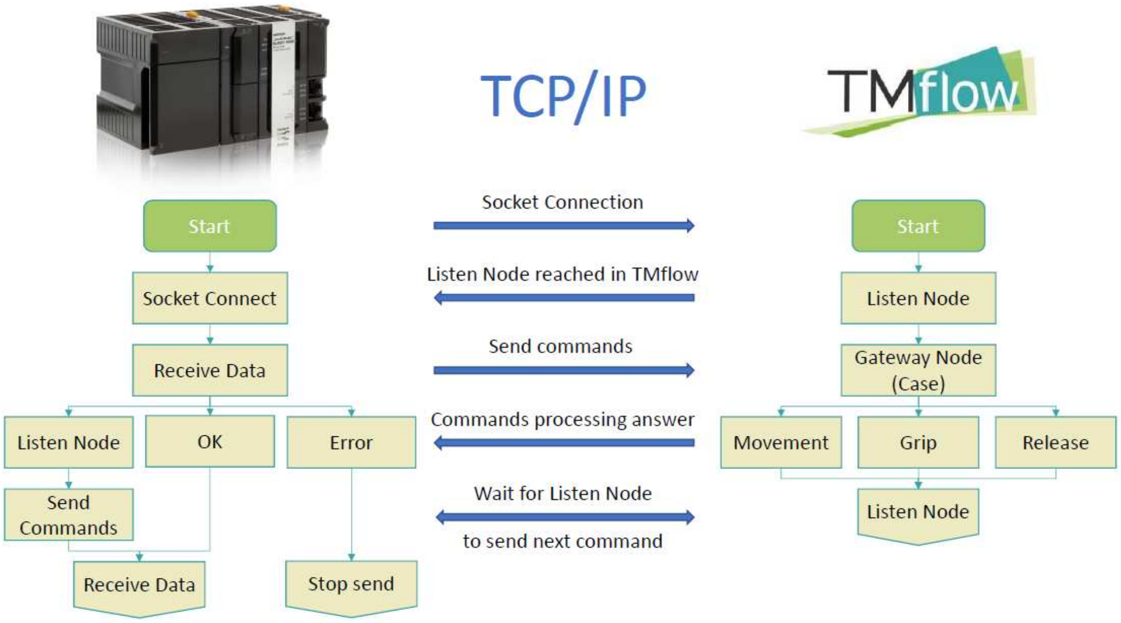
Velocity (mm/s): 250

Time to top speed (ms): 500

☑ Link to project speed

Communications model

The communication flow it is described in the following picture.

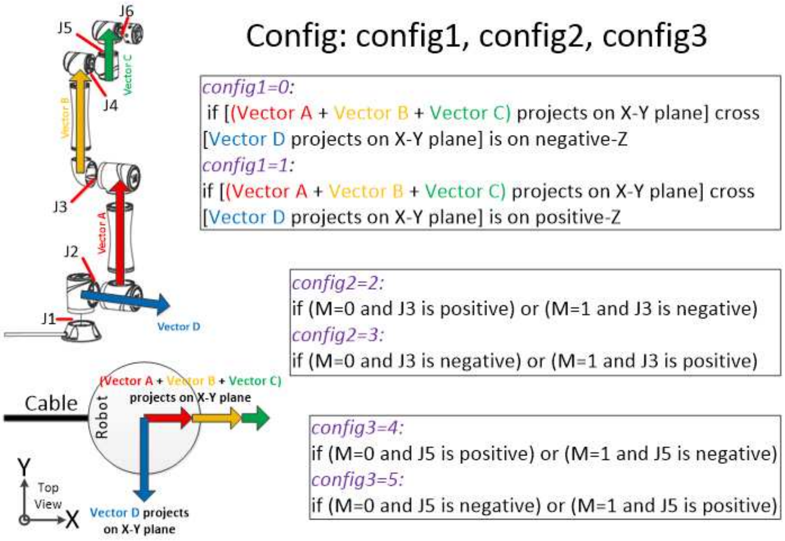


FBs DESCRIPTION

Movement commands

A movement command can be executed after entering Listen Node. There are two different FBs to execute the different robot movements from Omron Controller.

- **TM_MovCmd:** Sends a movement command script from a TCP port for the built-in EtherNet/IP to the Omron-TM robot. This FB has the following parameters.
    - Execute: FB will be executed with rise flank.
    - RobotID: Select the robot to send the command. Data type is structure _sSOCKET.
    - Movement: Define the type of movement. The following types are included:
        1. Point Node (PTP or Line).
        2. Move Node (PTP or Line).
    - DataFormat: Definition of data format which combines three letters.
        1. Motion target format:
            - J → Expressed in joint angles.
            - C → Expressed in cartesian coordinates.
            - T → Expressed in tool coordinates.
        2. Speed format:
            - P → Expressed as a percentage.
            - A → Expressed in velocity (mm/s).
        3. Blending format:
            - P → Expressed as a percentage.
            - R → Expressed in radius.
    - Point: Motion target.
    - Speed: The speed setting, expressed as a percentage (%) or in velocity (mm/s).
    - AccelTime: Time interval to accelerate to top speed (ms).
    - Blending: Blending value, expressed as a percentage (%) or in radius (mm).
    - PrecisePos: Disable precise positioning (when is set to True).
    - RobotPose: The pose of robot [Config1, Config2, Config3]. When nothing is introduced the current robot pose will be used. See the following picture for the robot pose configuration parameters.



Config: config1, config2, config3

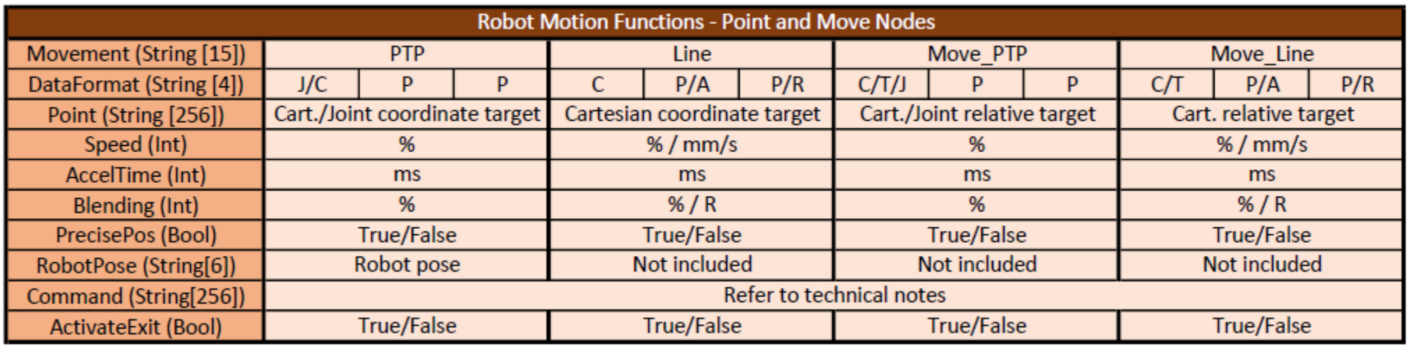
config1=0:
if [(Vector A + Vector B + Vector C) projects on X-Y plane] cross [Vector D projects on X-Y plane] is on negative-Z
config1=1:
if [(Vector A + Vector B + Vector C) projects on X-Y plane] cross [Vector D projects on X-Y plane] is on positive-Z

config2=2:
if (M=0 and J3 is positive) or (M=1 and J3 is negative)
config2=3:
if (M=0 and J3 is negative) or (M=1 and J3 is positive)

config3=4:
if (M=0 and J5 is positive) or (M=1 and J5 is negative)
config3=5:
if (M=0 and J5 is negative) or (M=1 and J5 is positive)

  - Command: Additional script command (e.g. ‘var_i=1000’ change robot variable value).
  - ActivateExit: Executes ScriptExit command to exit the Listen Node when is set to True.
  - CommandID: Script ID used as specifying the target Script of return message.

In the following picture, there is a description of the different parameters and robot movement scripts allowed.

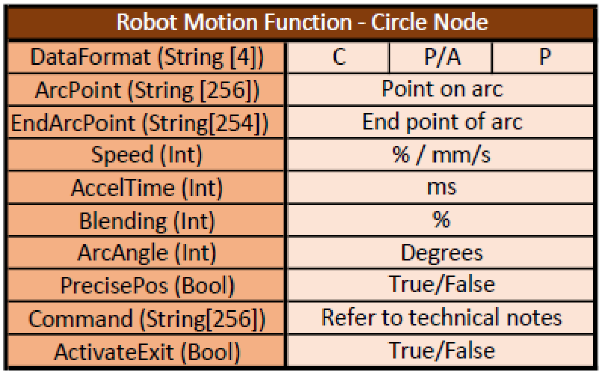


Robot Motion Functions - Point and Move Nodes												
Movement (String [15])	PTP			Line			Move_PTP			Move_Line		
DataFormat (String [4])	J/C	P	P	C	P/A	P/R	C/T/J	P	P	C/T	P/A	P/R
Point (String [256])	Cart./Joint coordinate target			Cartesian coordinate target			Cart./Joint relative target			Cart. relative target		
Speed (Int)	%			% / mm/s			%			% / mm/s		
AccelTime (Int)	ms			ms			ms			ms		
Blending (Int)	%			% / R			%			% / R		
PrecisePos (Bool)	True/False			True/False			True/False			True/False		
RobotPose (String[6])	Robot pose			Not included			Not included			Not included		
Command (String[256])	Refer to technical notes											
ActivateExit (Bool)	True/False			True/False			True/False			True/False		

- **TM_CircleCmd:** Sends a circle movement command script from a TCP port for the built-in EtherNet/IP to the Omron-TM robot. This FB has the following parameters.

  - Execute: FB will be executed with rise flank.
  - RobotID: Select the robot to send the command. Data type is structure _sSOCKET.
  - DataFormat: Definition of data format which combines three letters.
    1. Motion target format:
       - C → Expressed in cartesian coordinates.
    2. Speed format:
       - P → Expressed as a percentage.
       - A → Expressed in velocity (mm/s).
    3. Blending format:
       - P → Expressed as a percentage.
  - ArcPoint: A point on arc.
  - EndArcPoint: The end point of arc.
  - Speed: The speed setting, expressed as a percentage (%) or in velocity (mm/s).
  - AccelTime: Time interval to accelerate to top speed (ms).
  - Blending: Blending value, expressed as a percentage (%).
  - PrecisePos: Disable precise positioning (when is set to True).
  - Command: Additional script command (e.g. ‘var_i=1000’ change robot variable value).
  - ActivateExit: Executes ScriptExit command to exit the Listen Node when is set to True.
  - CommandID: Script ID used as specifying the target Script of return message.

In the following picture, there is a description of the different parameters allowed.



Robot Motion Function - Circle Node			
DataFormat (String [4])	C	P/A	P
ArcPoint (String [256])	Point on arc		
EndArcPoint (String[254])	End point of arc		
Speed (Int)	% / mm/s		
AccelTime (Int)	ms		
Blending (Int)	%		
ArcAngle (Int)	Degrees		
PrecisePos (Bool)	True/False		
Command (String[256])	Refer to technical notes		
ActivateExit (Bool)	True/False		

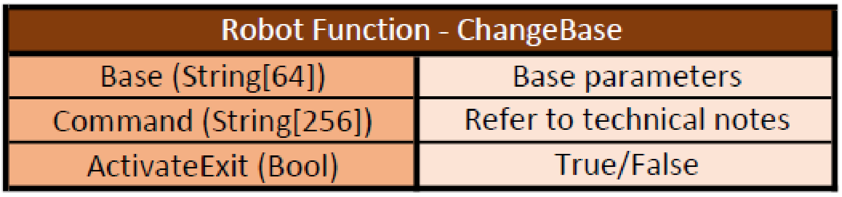
Robot motion function commands

The commands to change the Base, TCP or Payload can be executed for the follow-up motions.

- **TM_ChangeBase:** Sends the command of changing the base of the follow-up motions into buffer for execution from a TCP port for the built-in EtherNet/IP to the Omron-TM robot. This FB has the following parameters.

    - Execute: FB will be executed with rise flank.
    - RobotID: Select the robot to send the command. Data type is structure _sSOCKET.
    - Base: Base listed on the base list in TM Flow or specify base parameters “X, Y, Z, RX, RY, RZ”.
    - Command: Additional script command (e.g. ‘var_i=1000’ change robot variable value).
    - ActivateExit: Executes ScriptExit command to exit the Listen Node when is set to True.
    - CommandID: Script ID used as specifying the target Script of return message.

In the following picture, there is a description of the different parameters allowed.



Robot Function - ChangeBase	
Base (String[64])	Base parameters
Command (String[256])	Refer to technical notes
ActivateExit (Bool)	True/False

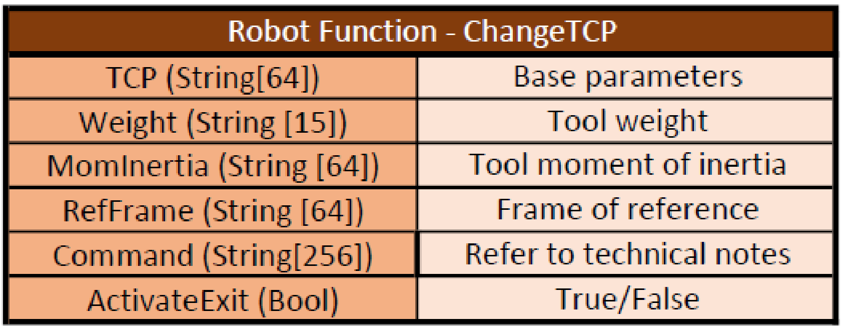
- **TM_ChangeTCP:** Sends the command of changing the TCP of the follow-up motions into buffer for execution from a TCP port for the built-in EtherNet/IP to the Omron-TM robot. This FB has the following parameters.

    - Execute: FB will be executed with rise flank.
    - RobotID: Select the robot to send the command. Data type is structure _sSOCKET.
    - TCP: TCP listed on the list in TM Flow or specify TCP parameters “X, Y, Z, RX, RY, RZ”.
    - Weight: Tool’s weight.
    - MomInertia: Tool’s moment of inertia “Ixx, Iyy, Izz”.

- RefFrame: Moment of inertia frame of reference “X, Y, Z, RX, RY, RZ”.
- Command: Additional script command (e.g. ‘var_i=1000’ change robot variable value).
- ActivateExit: Executes ScriptExit command to exit the Listen Node when is set to True.
- CommandID: Script ID used as specifying the target Script of return message.

➢ **Note**: When TCP is changed by specifying a set of TCP parameters, after the TMflow is stopped it will be necessary to select a TCP defined in the TMflow. It is proposed to change the TCP to a one defined in the TMflow rather than using a created one with TCP parameters. To introduce the Moment of Inertia, is necessary to define also the tool’s weight and the Reference Frame is only available with the Moment of Inertia defined.

In the following picture, there is a description of the different parameters allowed.

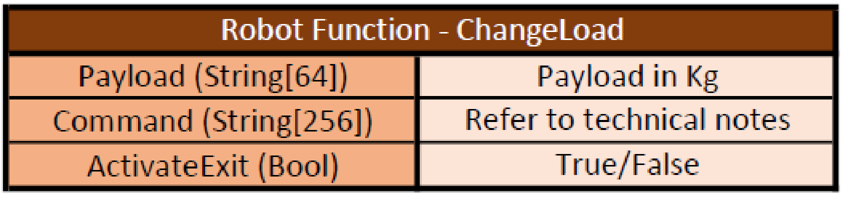


Robot Function - ChangeTCP	
TCP (String[64])	Base parameters
Weight (String [15])	Tool weight
MomInertia (String [64])	Tool moment of inertia
RefFrame (String [64])	Frame of reference
Command (String[256])	Refer to technical notes
ActivateExit (Bool)	True/False

- **TM_ChangeLoad:** Sends the command of changing the TCP of the follow-up motions into buffer for execution from a TCP port for the built-in EtherNet/IP to the Omron-TM robot. This FB has the following parameters.

  - Execute: FB will be executed with rise flank.
  - RobotID: Select the robot to send the command. Data type is structure _sSOCKET.
  - Payload: Payload in Kg.
  - Command: Additional script command (e.g. ‘var_i=1000’ change robot variable value).
  - ActivateExit: Executes ScriptExit command to exit the Listen Node when is set to True.
  - CommandID: Script ID used as specifying the target Script of return message.

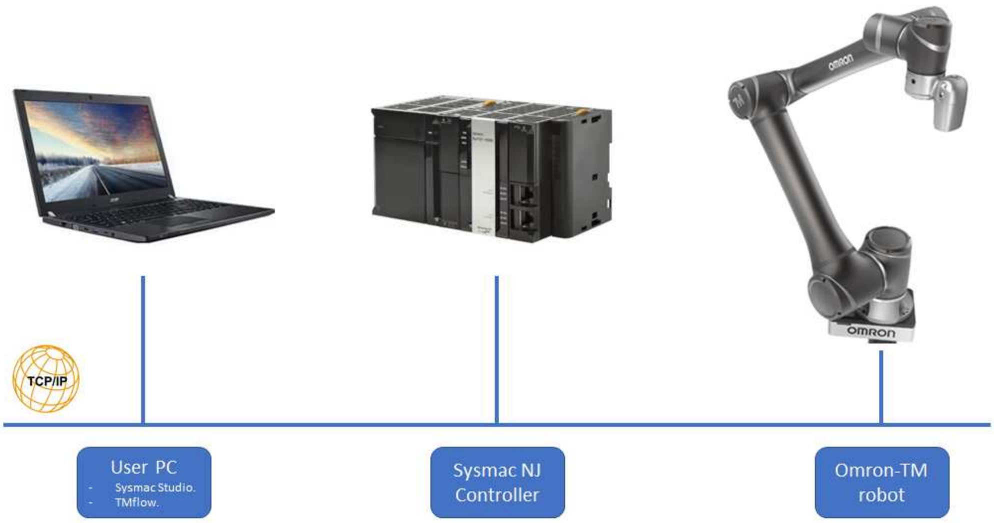
In the following picture, there is a description of the different parameters allowed.



Robot Function - ChangeLoad	
Payload (String[64])	Payload in Kg
Command (String[256])	Refer to technical notes
ActivateExit (Bool)	True/False

SYSTEM CONFIGURATION PROCEDURE (Example)

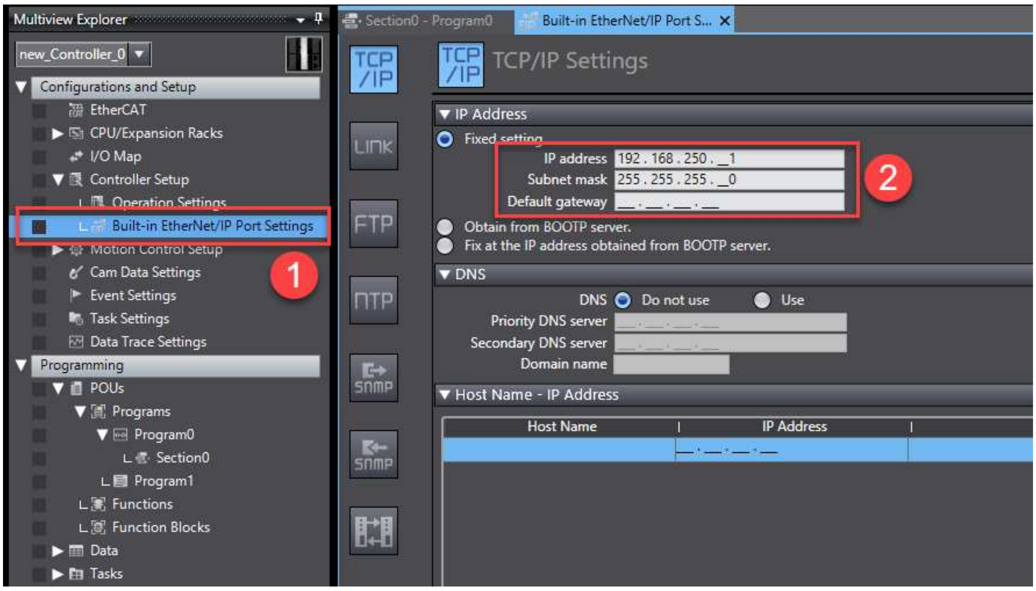
IP Address Settings



	IP Address	Subnet mask
User PC	192.168.250.61	255.255.255.0
Sysmac NJ	192.168.250.1	255.255.255.0
Omron-TM	192.168.250.30	255.255.255.0

NJ TCP/IP Configuration (Sysmac Studio settings)

Define the IP Address of the PLC as described in the following picture.



TCP/IP Settings

IP Address

● Fixed setting

IP address 192 . 168 . 250 . _1

Subnet mask 255 . 255 . 255 . _0

Default gateway __ . __ . __ . __

○ Obtain from BOOTP server.

○ Fix at the IP address obtained from BOOTP server.

DNS

DNS ● Do not use ○ Use

Priority DNS server __ . __ . __ . __

Secondary DNS server __ . __ . __ . __

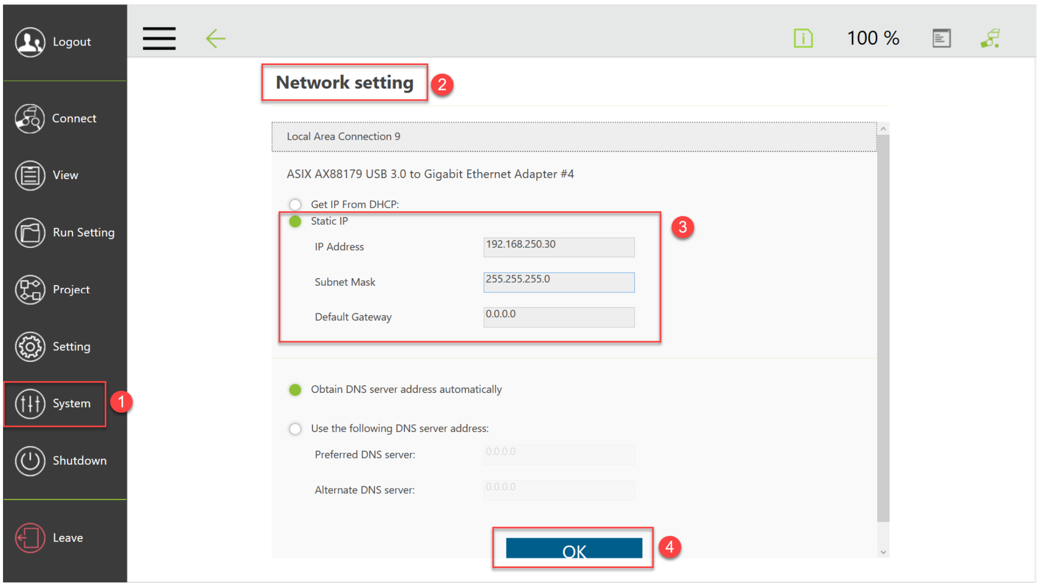
Domain name

Host Name - IP Address

Host Name	IP Address
	__ . __ . __ . __

Omron-TM Robot settings

To set the IP Address of the Omron-TM robot, go to System – Network – Select the local area connection – Set IP address and Subnet mask (see the following picture).



Network setting

Local Area Connection 9

ASIX AX88179 USB 3.0 to Gigabit Ethernet Adapter #4

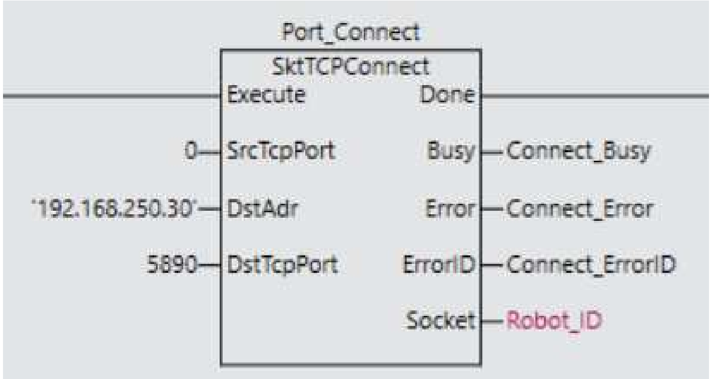
○ Get IP From DHCP:
● Static IP
IP Address 192.168.250.30
Subnet Mask 255.255.255.0
Default Gateway 0.0.0.0

● Obtain DNS server address automatically
○ Use the following DNS server address:
Preferred DNS server: 0.0.0.0
Alternate DNS server: 0.0.0.0

OK

Socket connection

Once the IP Addresses have been set, as described in the previous section, the PLC will be able to establish the connection with the Omron-TM robot. For that, the FB “SktTCPConnect” will be used as follow:



Port_Connect
SktTCPConnect
Execute — Done
0 — SrcTcpPort — Busy — Connect_Busy
'192.168.250.30' — DstAdr — Error — Connect_Error
5890 — DstTcpPort — ErrorID — Connect_ErrorID
Socket — Robot_ID

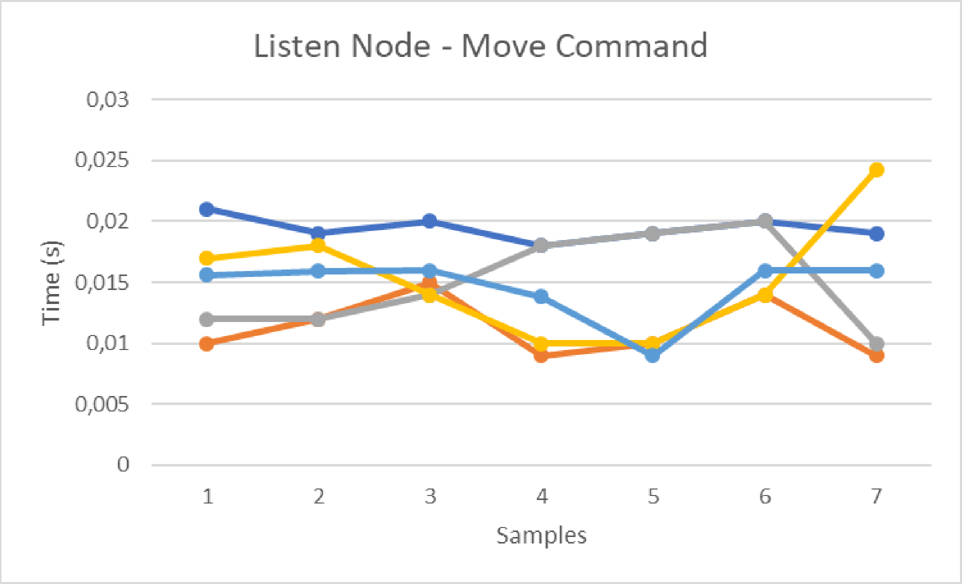
It is necessary to define the DstAdr of the robot (defined in TM Flow under System - Network setting) and the port number ‘**5890**’. For the PLC, the source port number has been defined as ‘**0**’, in this case, the socket will use a different port number each time the connection is requested. Should be considered that Socket information ‘**Robot_ID**’ will be used in the movements/commands FBs.

Timing

In the following charts, there are shown the different timings during the PLC and Omron-TM robot communication.

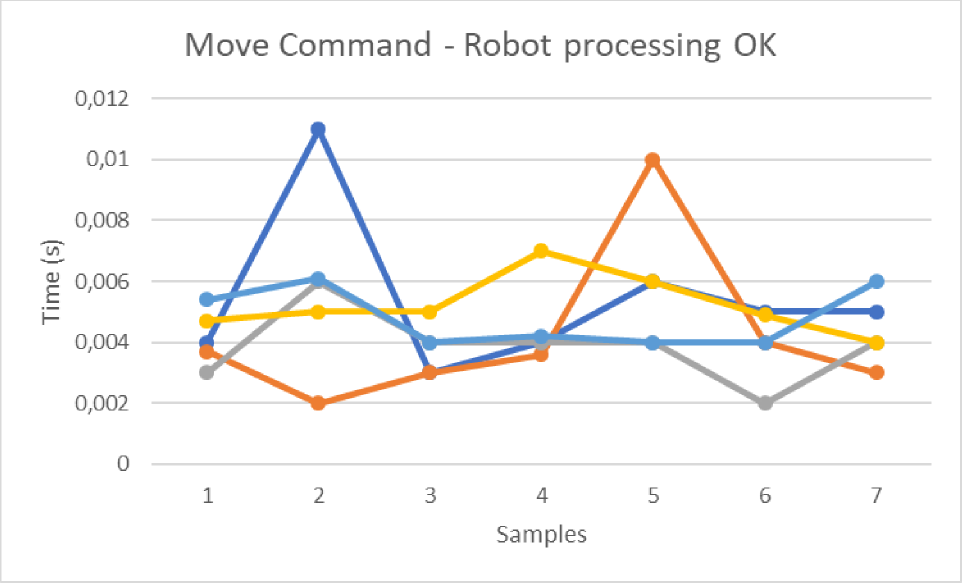
1. Time between the **Listen Node** is reached (when the message is sent from the robot to the PLC) and the **Move Command** is sent from the PLC to the robot for processing.

   The average time is around 15ms.



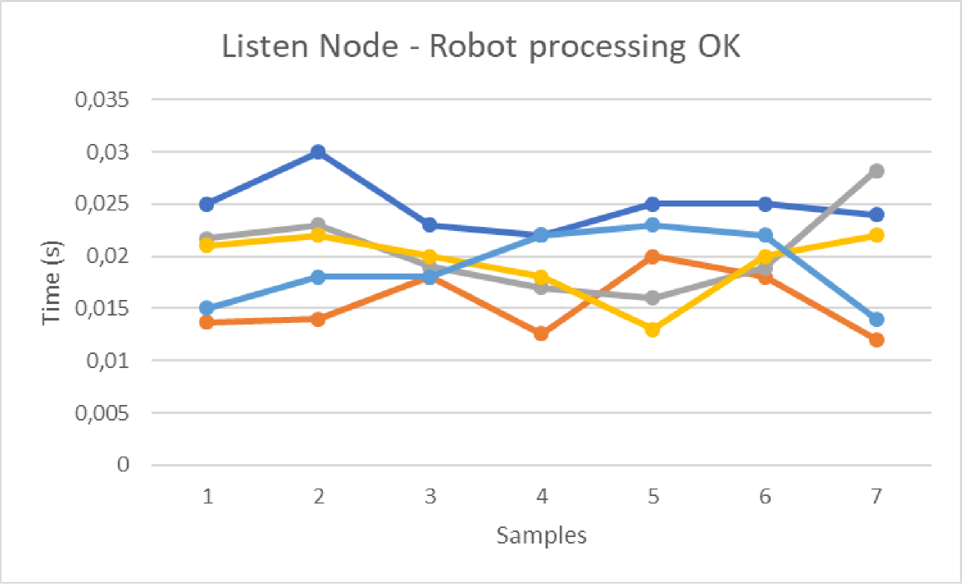
2. Time between the **Move Command** is sent for processing and the robot confirms – **OK** – (or gives an error) to the command after processing.

   The average time is around 5ms.



3. Time between the **Listen Node** is reached (when the message is sent from the robot to the PLC) and the robot confirms – **OK** – (or gives an error) to the command after processing.

   The average time is around 20ms.



Limitations

The current version of the Listen node and Expression Editor functions, doesn't support getting any kind of feedback or status information from the robot, only sending commands and writing variables is possible.

It is suggested to use the "socket disconnection" as a condition to manage error states in application programs.