



# Listen Node

# Description

## ➤ Purpose

### Listen Node

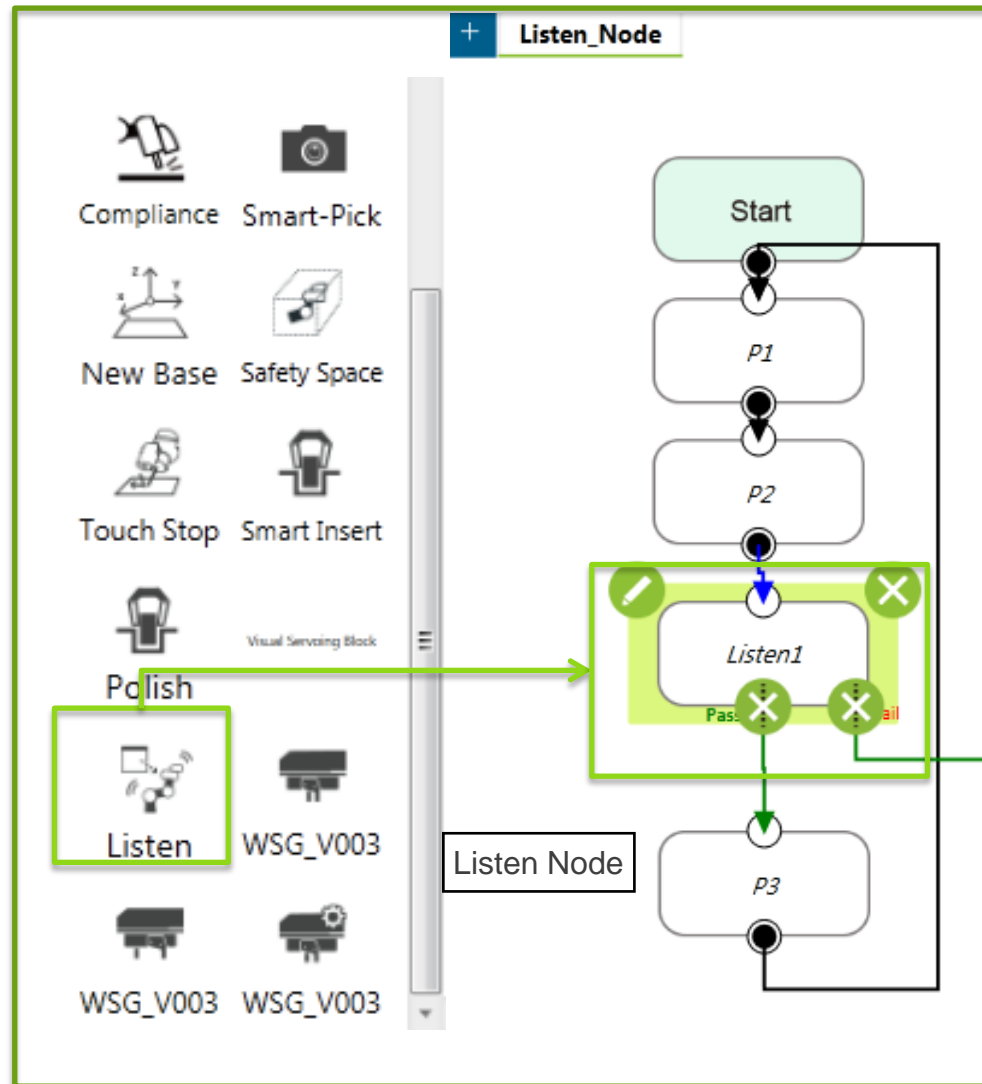
1. Listen Node is a build-in Service, so that external device could send motion command through TCP/IP.
2. Listen node is always be used on the Flow with other function nodes, in order to build more flexible applications

## ➤ Device Require

1. TM5
2. NB

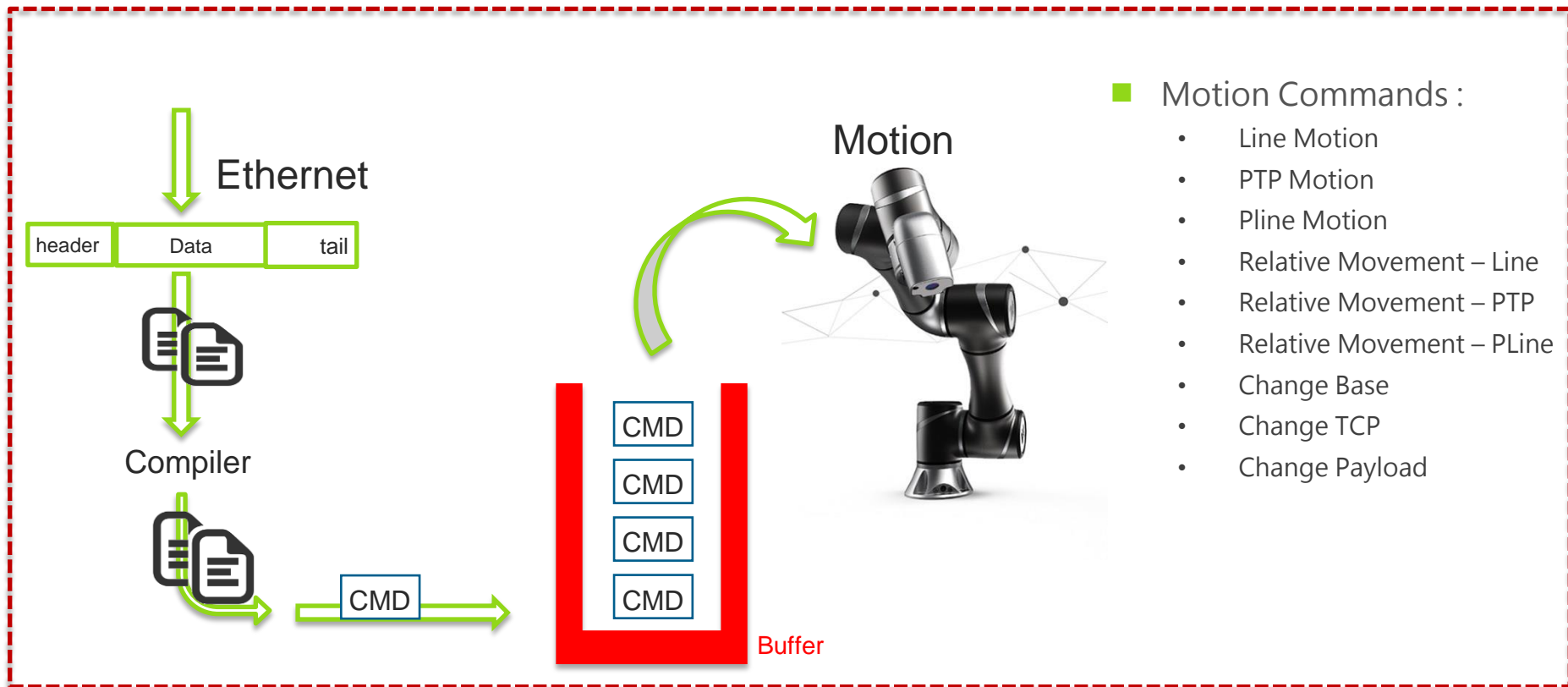
## Application Example

- Function Operation
- Sample Flow



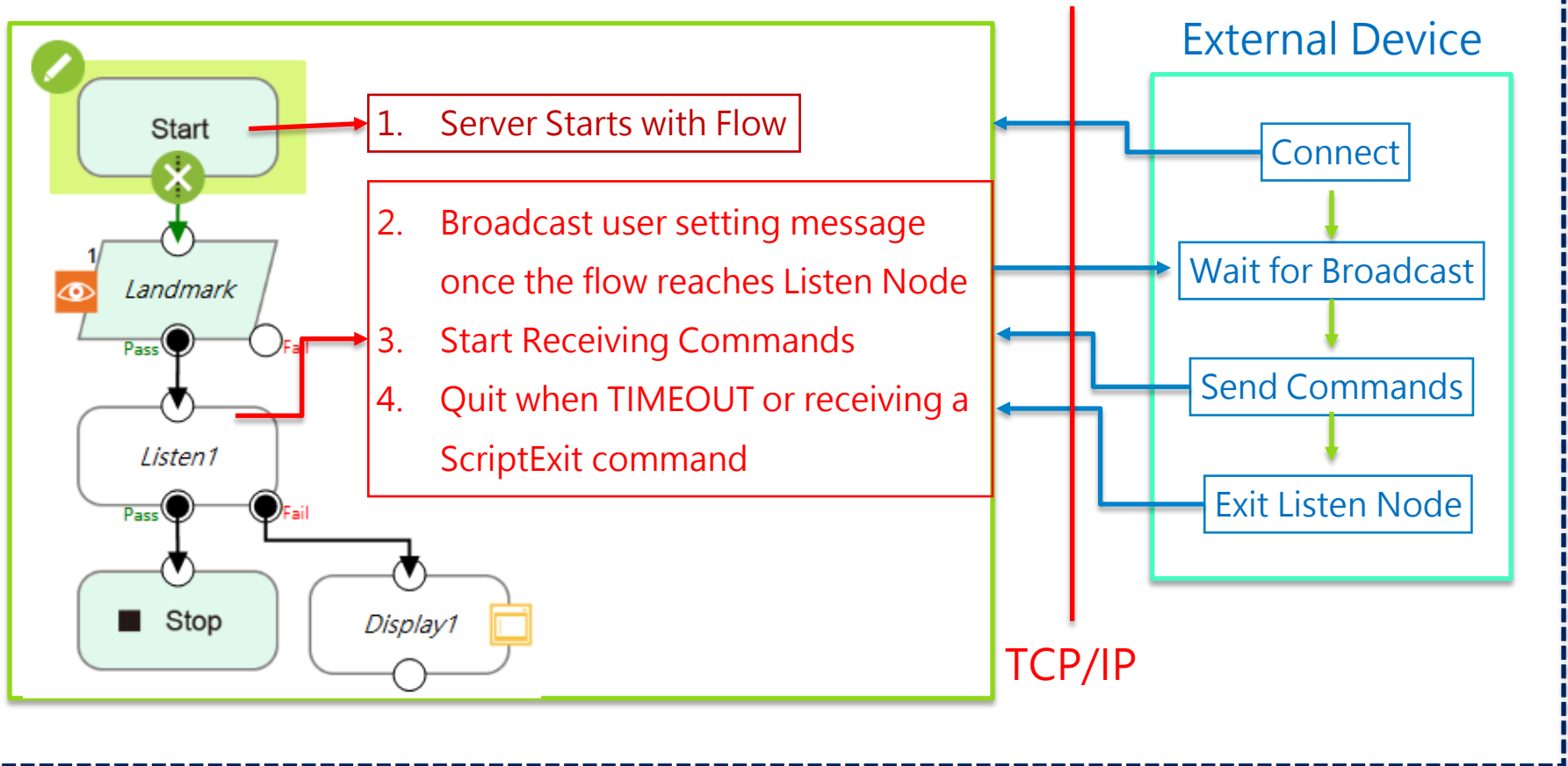
# Listen Node

- Listen Node is a build-in Server on TM robot with specified PORT. Listen Server activates once a TM Project is started, when the project flow reaches Listen node, the Server starts accept commands from any external devices connected until TIMEOUT or a ScriptExit command or the Project STOP.
- After the command received is compiled and checked, Listen Server will response with a message specific to the command or if this command is illegal. One Command receives one Response from the Listen Server



# Listen Node

- ✓ Listen node is a build-in **Server** ※ Fixed PORT 5890
- ✓ External Devices (**Client**) send commands through **TCP/IP**
- ✓ Combine with other function nodes as a Flow



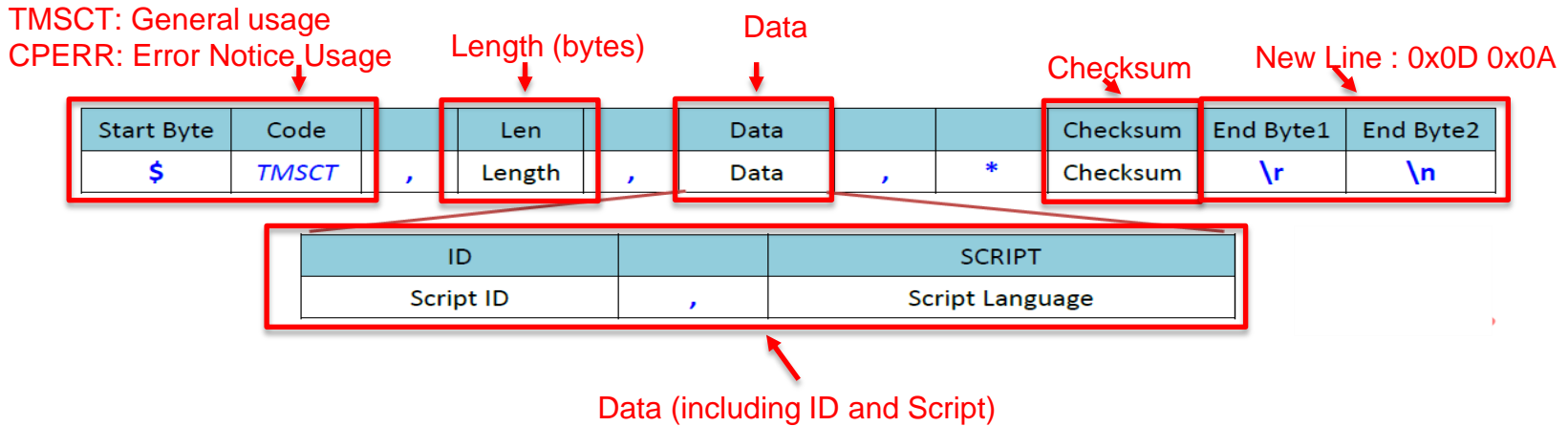
# Listen Node

The screenshot shows a dialog box titled "Listen" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Node Name:** A text input field containing "Listen1".
- Send message as entering this node:** A text input field containing "\"Listen1\"".
- Print received data in log:** A checkbox that is currently unchecked.
- Connection Timeout:** A numeric input field containing "0" and a "Var" button to its right.
- Data Timeout:** A numeric input field containing "0" and a "Var" button to its right.
- Buttons:** "OK" (blue) and "Delete this node" (pink) buttons at the bottom.

- **Node Name :** Node Name
- **Send message as entering this node :**
  - This defined message will be send once Listen node starts broadcasting
  - Default Message is the name of this node
  - No Empty Space inside this message
- **Print received data in log :** Select if the received message being shown at View/Log
- **Connection Timeout :** If there is no connection within this time interval, quit this node
- **Data Timeout :** After being connected, it there is not command being received, quit this node

# Listen Node Command



## ➤ Checksum

- Checksum is to confirm the correctness of the message (for the receiver) which is calculated by using XOR (exclusive OR) on all bytes between \$ and \* (\$ and \* is not included). For example :

\$ **TMSCT,100,Data**,\*CS\r\n

Checksum = Byte[1] ^ Byte[2] .... ^ Byte[N-6]

- The format of Checksum is written in **2bytes (ASCII) form the Hexadecimal result (not necessary to include 0x)**. For example :

\$ **TMSCT,5,10,OK**,\*6D\r\n

CS (XOR) = 0x54 ^ 0x4D ^ 0x53 ^ 0x43 ^ 0x54 ^ 0x2C ^ 0x35 ^ 0x2C ^ 0x31 ^ 0x30 ^ 0x2C ^ 0x4F ^ 0x4B ^ 0x2C = 0x6D

CS (string) = 6D → CS(2bytes) = 0x36 0x44



Checksum\_tool

# Listen Node Command

## Example 1: using PTP() for angular motion

```
$TMSCT,,1,PTP("JPP",-90,0,90,0,90,0,20,200,0,false)
PTP("JPP",0,0,90,0,90,0,20,200,0,false)
PTP("JPP",90,0,90,0,90,0,20,200,0,false),*2C(newline)

$TMSCT,,1,PTP("JPP",-90,0,90,0,90,0,20,200,0,false)
PTP("JPP",0,0,90,0,90,0,20,200,100,false)
PTP("JPP",90,0,90,0,90,0,20,200,0,false),*22 (newline)
```

## Example 2: Line for a target point (Cartesian coordinates)

```
$TMSCT,,1,Line("CAP",500,0,500,180,0,90,10,200,0,false)
Line("CAP",500,100,500,180,0,90,10,200,0,false)
Line("CAP",500,-100,500,180,0,90,10,200,0,false),*72 (newline)
```

## Example 3 : ChangeBase

```
$TMSCT,,1,
float[] basevalue
basevalue = {0,-100,0,0,0,0}
ChangeBase(basevalue)
Line("CAP",500,100,500,180,0,90,10,200,0,false)
ChangeBase("RobotBase")
Line("CAP",500,100,500,180,0,90,10,200,0,false),*0E (newline)
```

## Quit Listen Node

```
$TMSCT,,1,ScriptExit(),*62(newline)
```

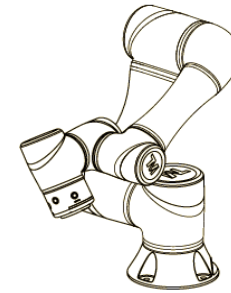


TM\_Robot\_Function\_Motion

# Example 1 : Client send Command to Robot



NB (Client)



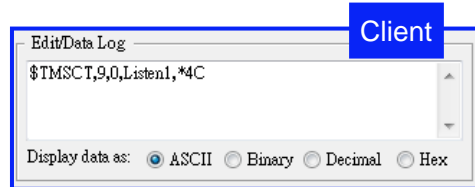
TM Robot (Server)

1. Connection

2. Listen Node Send message

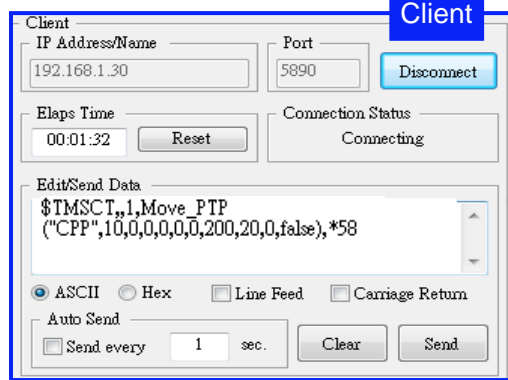
3. Client Send Command

4. Client Send Exit Listen Node



Client

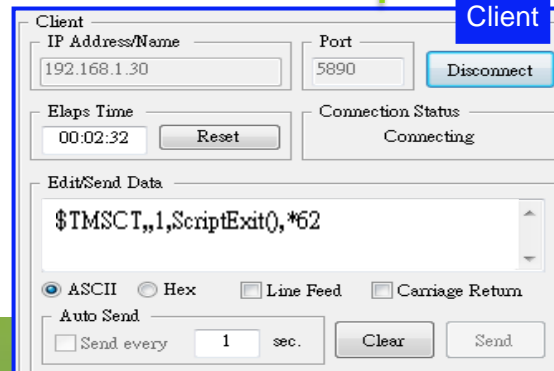
A broadcast message is received once connected to Listen Node



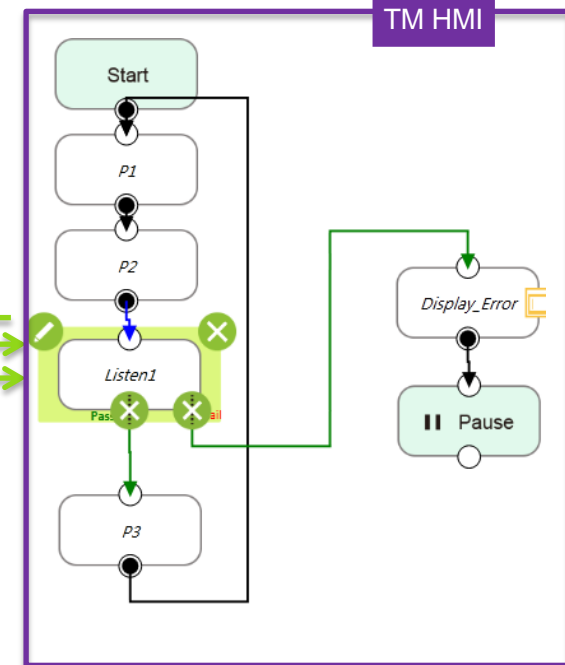
Client

Send "Move\_PTP"

Send "ScriptExit"



Client



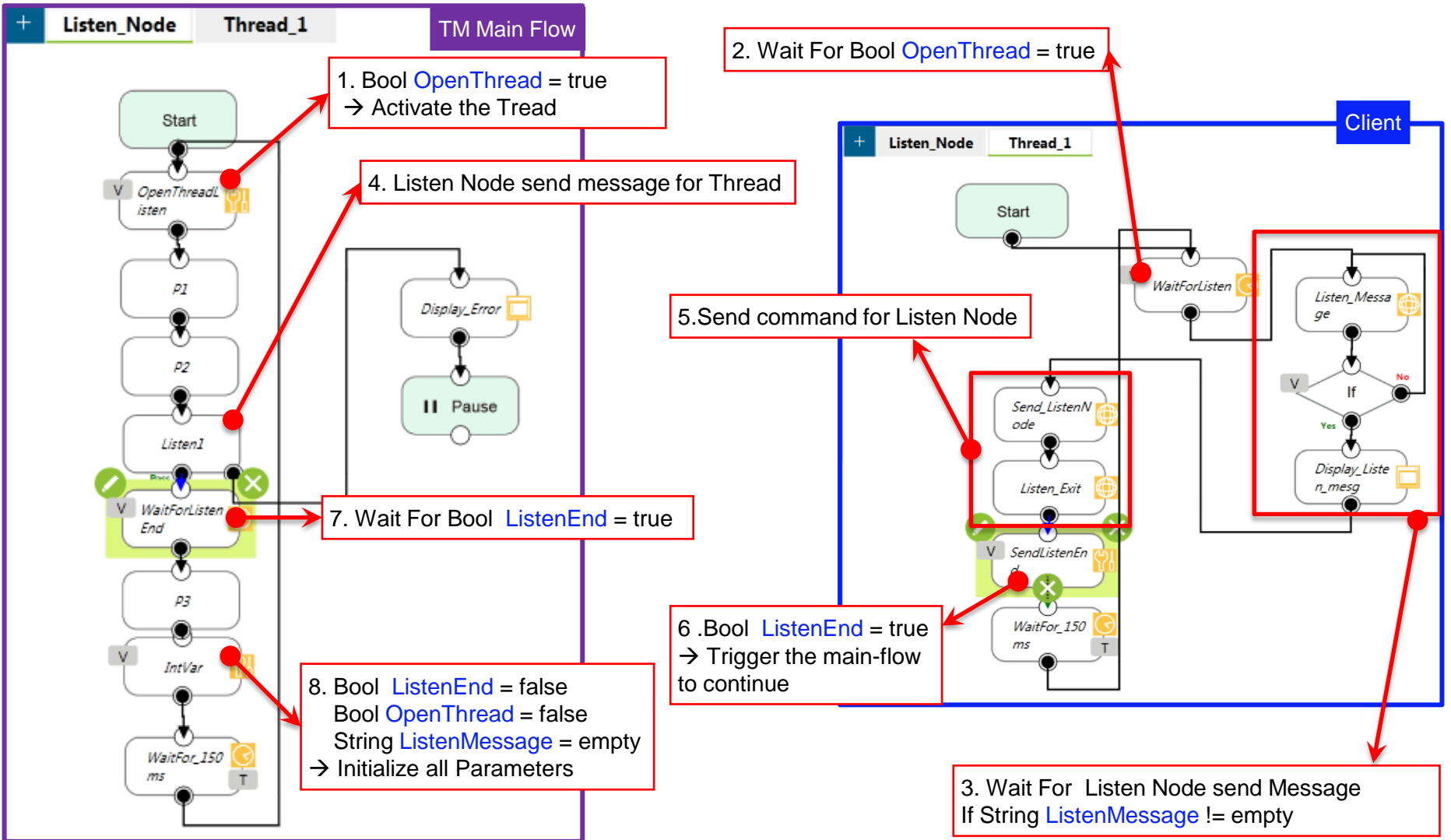
TM HMI

\$TMSCT,4,1,OK,\*5C  
 \$TMSCT,,1,ScriptExit()  
 Transmit succeed, an "OK" message is received

- ◆ Port 5890 (Fixed)
- ◆ Newline is necessary for each Command



# Example 2 : Send command through HMI Thread



※ This method is usually being used if user wants a specific configuration for a pose (without a external comm. device)

# Example 2 : Send command through HMI Thread

Node Name: Send\_ListenNode

Choose Device: Listen\_NetWork

Robot IP and Port 5890

Add Device Edit Device

Receive to Variable Send Send Listen node Motion command

Typing

```
ListenPacket("1,Move_PTP  
("CPP",20,0,0,0,0,200,200,0,false)")
```

Network

Node Name: Listen\_Exit

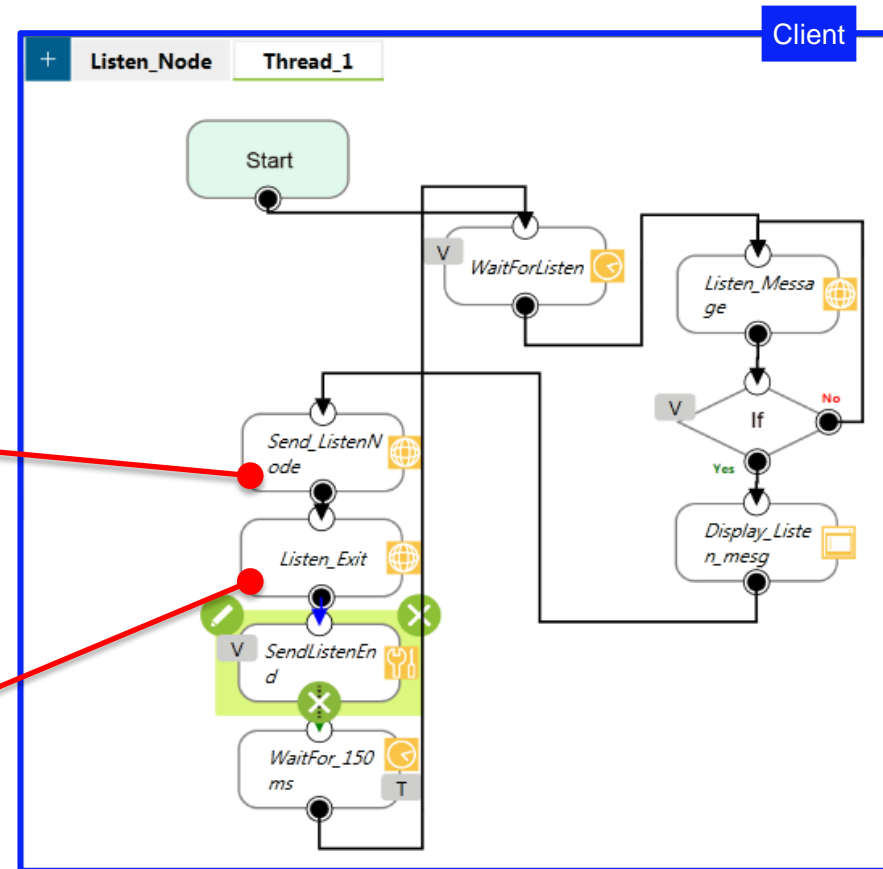
Choose Device

Add Device Edit Device

Receive to Variable Send Send Exit Listen Node command

Typing

```
ListenPacket("1,ScriptExit()")
```



END