

The top section shows a 3D model of a conveyor belt system with various status indicators:

- SENSORS:** FACTORY I/O (Paused), FACTORY I/O (Reset), FACTORY I/O (Running), FACTORY I/O (Time Scale), Sensor.
- ACTUATORS:** Conveyor, FACTORY I/O (Camera Position), FACTORY I/O (Pause), FACTORY I/O (Reset), FACTORY I/O (Run).
- Server:** Mitsubishi.MXOPC.6 (6) - Connected, Receives, Transmits.
- Tags:** TagM0, TagX0, TagY0.

MELSOFT Series GX Works2 (Untitled Project) - [[PRG]Monitor Executing MAIN (Read Only) 4 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation Project Parameter [PRG]Monitor Executing M... X

```
graph LR; X000 --> M0[M0]; M0 --> Y000((Y000)); Y000 --> END[END];
```

MXConfigurator.mdb - MX OPC Configurator

File Edit View Go Tools Help

Active Configuration: C:\MELSEC\MX OPC Server 6.10\MXConfigurator.mdb

Address Space

Name	Enable	Simulate	Address	Access Rig...	Alarms
DYNAMIC TAGS					
TagM0	Yes	No	M0	Read, Write	No
TagX0	Yes	No	X0	Read, Write	No
TagY0	Yes	No	Y0	Read, Write	No

GX Simulator2

Tool Options

Switch: STOP (radio button), RUN (radio button)

LED:

POWER	GREEN
RUN	GREEN
ERR.	GREY

Ready 4 Object(s) NUM



Datalogger and Alarm Log

Historical data collector
 00:Historical data collector0

General Data Item

Name: Historical data collector0

Memory

Sampling total: 10000 (circled)

Memory Required:

Note: The number of memory required = (20+ all data items in the data category) * The total number of samples

Sampling Method

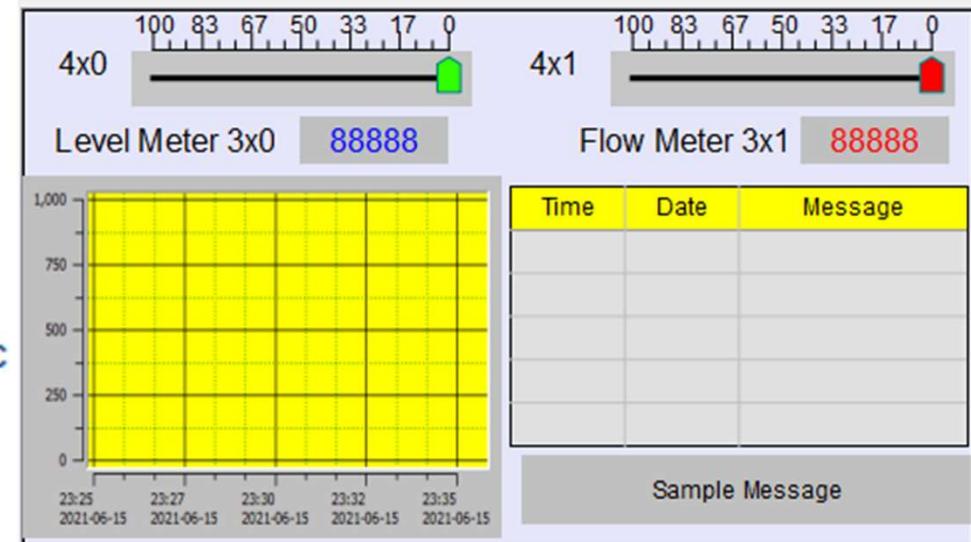
Timing Distance: 5 Second(s) (circled)

Trigger

General Data Item

Language: Chinese Add Alter Delete

Name	Address	Data type	Upload
1 LevelMeter	3x0	16-Bit Unsigned	<input type="checkbox"/>
2 FlowMeter	3x1	16-Bit Unsigned	<input type="checkbox"/>



Analog Alarm Block

Name: Analog Alarm Login1
Type: Numerical comparison
Read Address: 3x0
Size: 1
Scanning Time: 5 Sec

Use	Message
3x0 <input checked="" type="checkbox"/> Low Low	<<200
<input checked="" type="checkbox"/> Low	<300
<input checked="" type="checkbox"/> High	>700
<input checked="" type="checkbox"/> High High	>>800

Alarm setting

Data type: 16-Bit Int
Address: 3x0
Alarm Message: Out of 100 - 900

Alarm value setting

Condition: Alarm with the outside the range
Lower limit: 100
Upper limit: 900

Communication Port Properties

? X >

General

Parameter

Link ID: 1

Link Name: ModbusSlaveTCP

Link Interface: Ethernet

HMI Site: Local

Setting

COM port (master-slave mode) port:1

Connection Services: Modbus

Modbus Slave TCP/IP

General

Parameter

HMI IP

Connected equipment ip

IP Address: 192 ■ 168 ■ 1 ■ 51

Port number: 502

Subnet mask:

Gate

Other

HMI Address: 1

Plc station: 0

Communication time: 5 (ms)

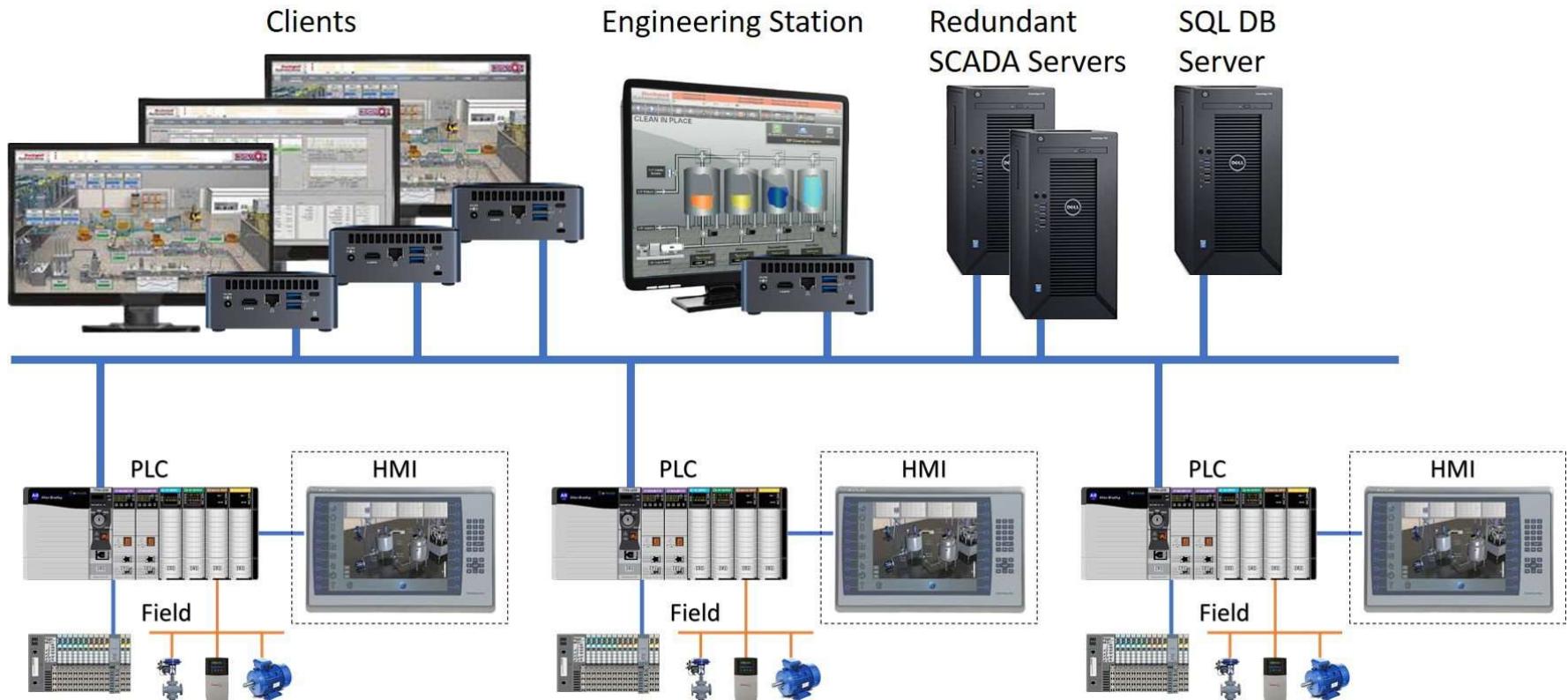
Overtime time 1: 1000 (ms)

Overtime time 2: 5 (ms)

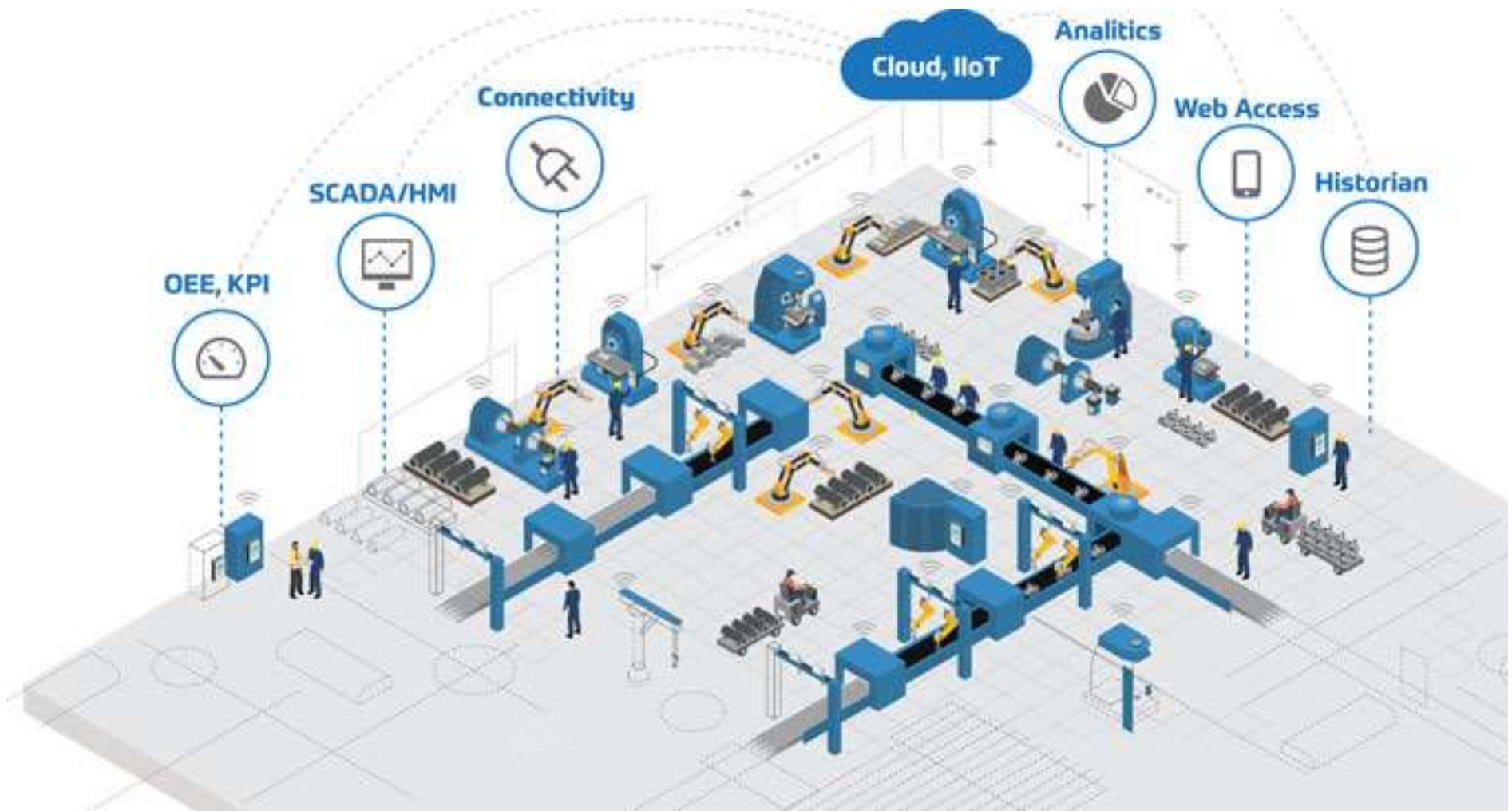
Retries: 3

WHAT IS SCADA

- The utilization of Supervisory Control and Data Acquisition (SCADA)



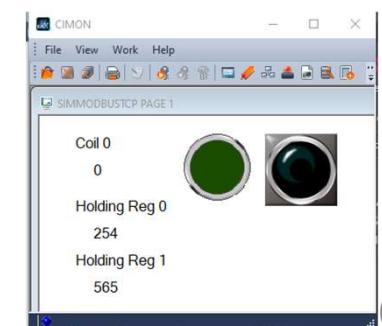
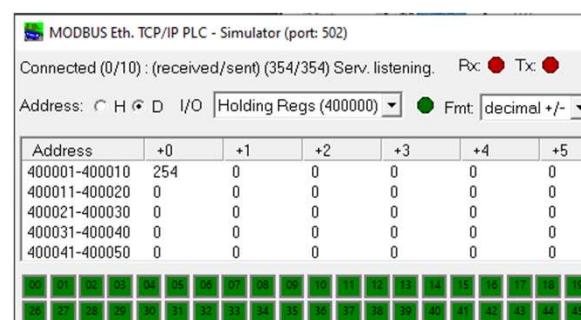
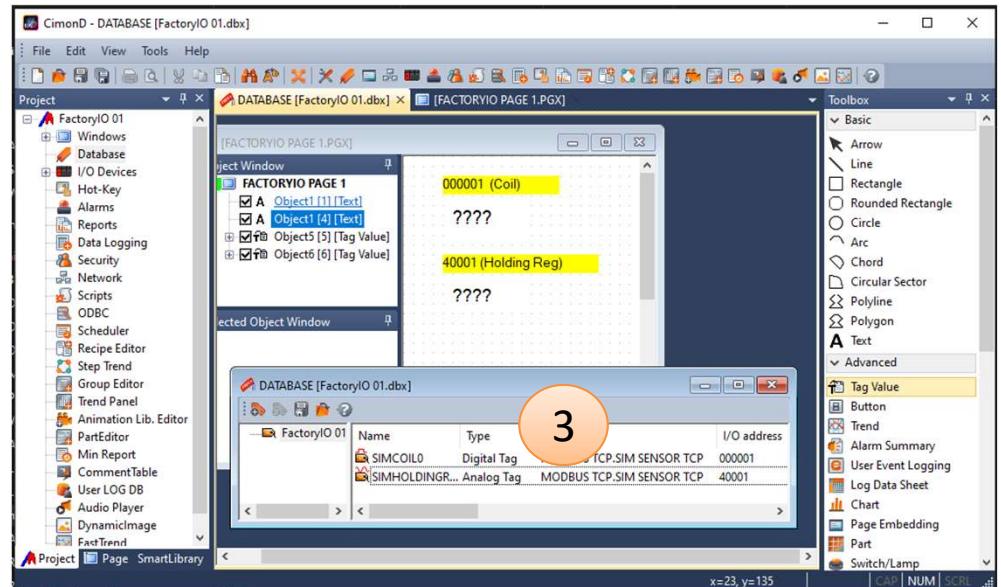
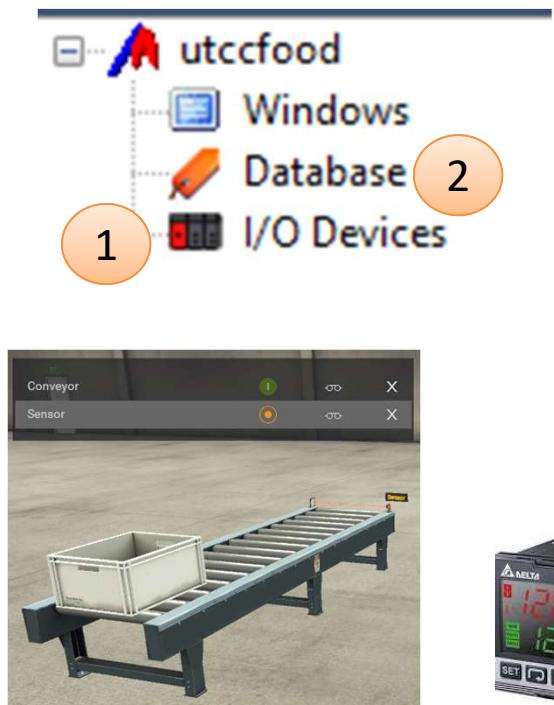
IIoT & Industrial 4.0



<https://www.tdsthailand.com/industrial/detail.php?i=12>

Cimon SCADA

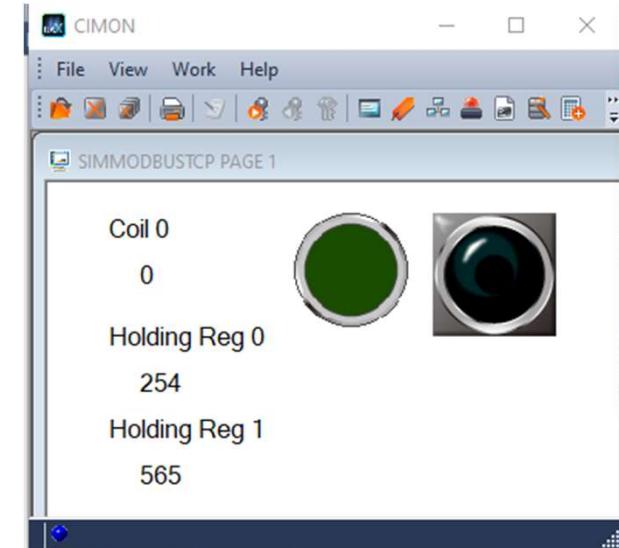
- Cimon - FactoryIO



YouTube SCADA - 01 SimModbus

The image displays a SCADA system interface with five numbered steps:

- 1**: A screenshot of the "MODBUS Eth. TCP/IP PLC - Simulator (port: 502)" window. It shows a table of holding registers (Holding Regs) from address 400001 to 400041. The first row (400001-400010) has values 254, 0, 0, 0, 0, 0. An orange circle labeled **1** highlights the first row.
- 2**: A screenshot of the "MODICON - MODBUS/TCP" configuration window. It shows a "Station" and a "COM Port" section. Under "ModRSsim2", it lists two connection profiles: "MODRSSIM2 [127.0.0.1]" with addresses 0 - 01, 2 Word : 1.0 sec and 1 - 41, 2 Word : 1.0 sec. An orange circle labeled **2** points to the "ModRSsim2" icon.
- 3**: A screenshot of the "CimonD - [SIMMODBUSTCP PAGE 1.PGX]" project structure. It shows a tree view with "utccfood" expanded, containing "Windows", "Database", and "I/O Devices". An orange circle labeled **3** points to the "Database" node.
- 4**: A screenshot of the "DATABASE [SimModbusTCP.dbx]" window. It shows a table of I/O devices with three entries: COIL0 (Digital Tag), HR0 (Analog Tag), and HR1 (Analog Tag). An orange circle labeled **4** points to the "Database" window.
- 5**: A screenshot of the "MODBUSTCP PAGE 1.PGX" visualization window. It contains three objects: "Coil 0" (represented by a green circle), "Holding Reg 0" (represented by a black square with a white border), and "Holding Reg 1" (represented by another black square with a white border). An orange circle labeled **5** points to the "Holding Reg 0" object.



MODBUS Eth. TCP/IP PLC - Simulator (port: 502)

Connected (1/10) : (received/sent) (168/168) Serv. write data Rx: Tx:

Address: H D I/O **Coil Outputs (000000)** Fmt: decimal +/-

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10
000001-000016	0	0	0	0	0	0	0	0	0	0	0
000017-000032	0	0	0	0	0	0	0	0	0	0	0

MODBUS Eth. TCP/IP PLC - Simulator (port: 502)

Connected (1/10) : (received/sent) (248/248) Serv. write data

Address: H D I/O **Holding Regs (400000)** Fmt: decimal

MODICON - MODBUS/TCP

Station COM Port

ModRSsim2

- MODRSSIM2 [127.0.0.1]
- 0 - 01, 2 Word : 1.0 sec
- 1 - 41, 2 Word : 1.0 sec

MODBUS Eth. TCP/IP PLC - Simulator (port: 502)

Connected (1/10) : (received/sent) (248/248) Serv. write data

Address: H D I/O **Holding Regs (400000)** Fmt: decimal

Address	+0	+1	+2	+3
400001-400010	254	565	0	0
400011-400020	0	0	0	0

DATABASE [SimModbusTCP.dbx]

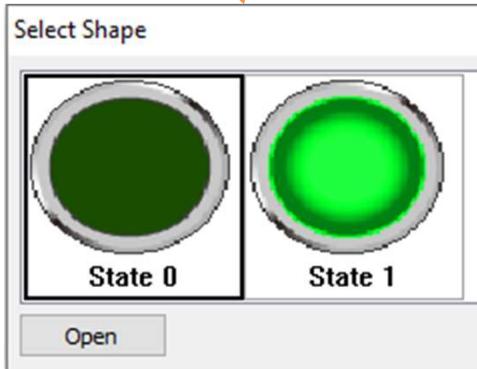
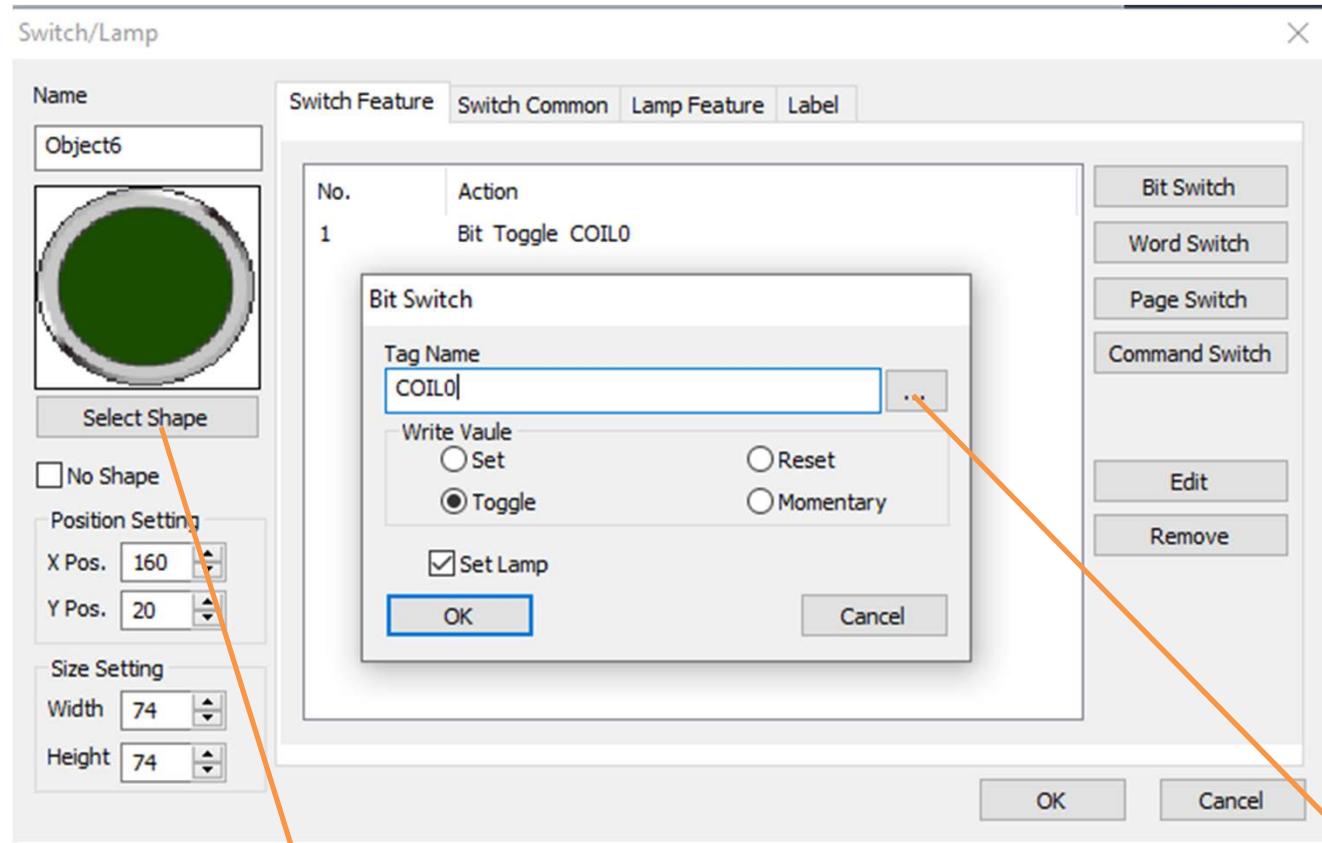
SimModbusTCP

Name	Type	I/O device	I/O address	Initial v...
COIL0	Digital Tag	MODRSSIM2.MODRSSIM2	000001	0
HR0	Analog Tag	MODRSSIM2.MODRSSIM2	40001	0
HR1	Analog Tag	MODRSSIM2.MODRSSIM2	40002	0

The image shows a SCADA configuration interface with several windows open:

- Toolbox:** A sidebar with icons for Polygon, Text, Advanced, and Tag Value.
- Object Config (Main Window):** Shows a "Switch/Lamp" icon and a preview of a green button. It lists objects: Coil 0, Holding Reg 0, Holding Reg 1, and three "?????" entries. Red arrows point from the "Tag Value" icon in the Toolbox to these entries.
- Object Config (Object2):** A configuration dialog for "Object2". It has a "Dynamic Tag" tab. Under "Config", "Style" and "Tag Value" are checked. In the "Tag Value" section, "Tag name" is set to "COILO", "Preview" is "12345.12345", and "Display Format" is "?????".
- Object Config (Object3):** A configuration dialog for "Object3". It has a "Text" type. Under "Config", "Style" and "Text Editing" are checked. The "Text Editor" pane contains the text "Coil 0".
- Select Tag:** A dialog showing a "Select" field with "COILO" and a "Database" field with "SimModbusTCP".
- Table:** A table showing database entries:

Name	Type
COILO	Digital Tag
HR0	Analog Tag
HR1	Analog Tag

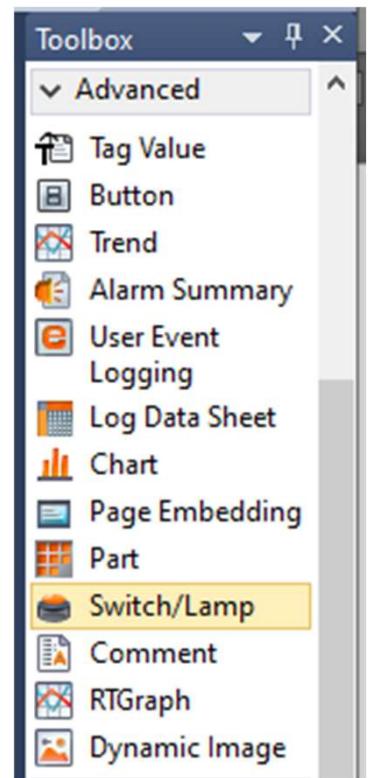


Select Tag

Select: COILO

Database: SimModbusTCP

Name	Type
COILO	Digital Tag
HR0	Analog Tag
HR1	Analog Tag



Switch/Lamp

Name: Object7

Switch Feature: Bit Momentary COIL0

Action: Bit Momentary COIL0

Bit Switch:

- Tag Name:** COIL0
- Write Value:**
 - Set
 - Reset
 - Momentary
- Set Lamp

Position Setting:

- X Pos.: 250
- Y Pos.: 20

Size Setting:

- Width: 80
- Height: 80

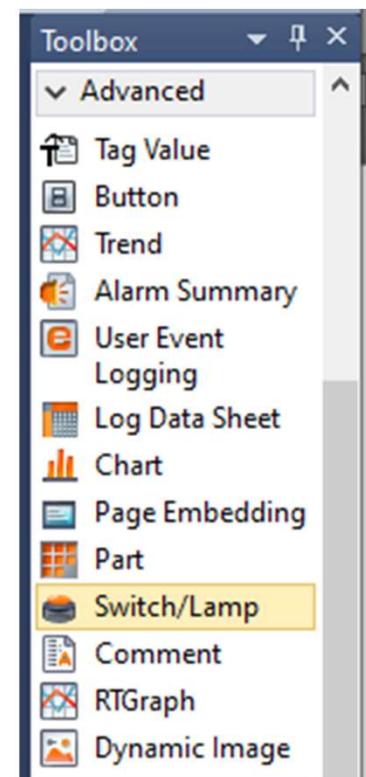
Select Shape:

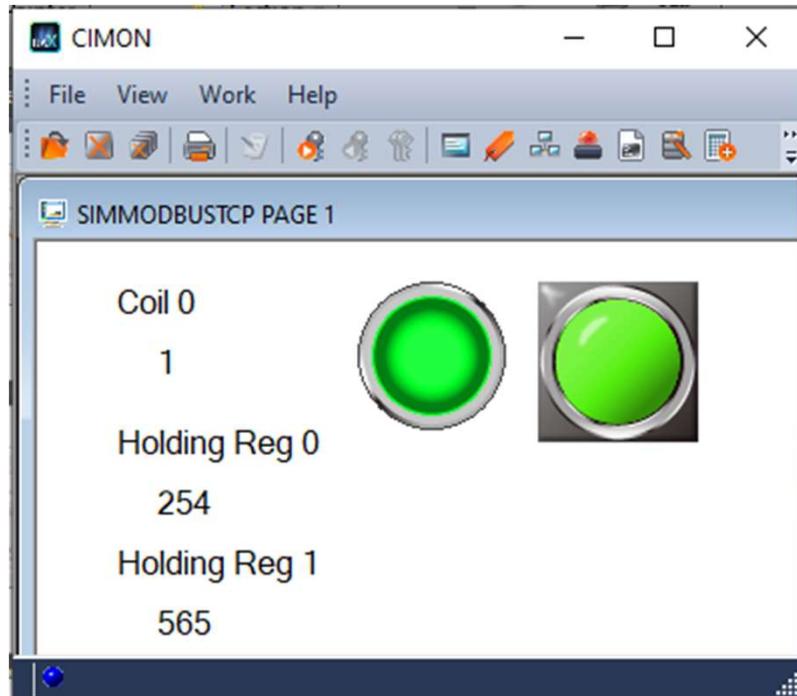
Select Tag:

Select: COIL0

Database: SimModbusTCP

Name	Type
COIL0	Digital Tag
HR0	Analog Tag
HR1	Analog Tag





MODBUS Eth. TCP/IP PLC - Simulator (port: 502)

Connected (1/10) : (received/sent) (3329/3329) Serv. write de Rx: ● Tx: ●

Address: C H D I/O Coil Outputs (000000) Fmt: decimal +/-

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10
000001-000016	1	0	0	0	0	0	0	0	0	0	0
000017-000032	0	0	0	0	0	0	0	0	0	0	0

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45

MODBUS Eth. TCP/IP PLC - Simulator (port: 502)

Connected (1/10) : (received/sent) (3236/3236) Serv. write de Rx: ● Tx: ●

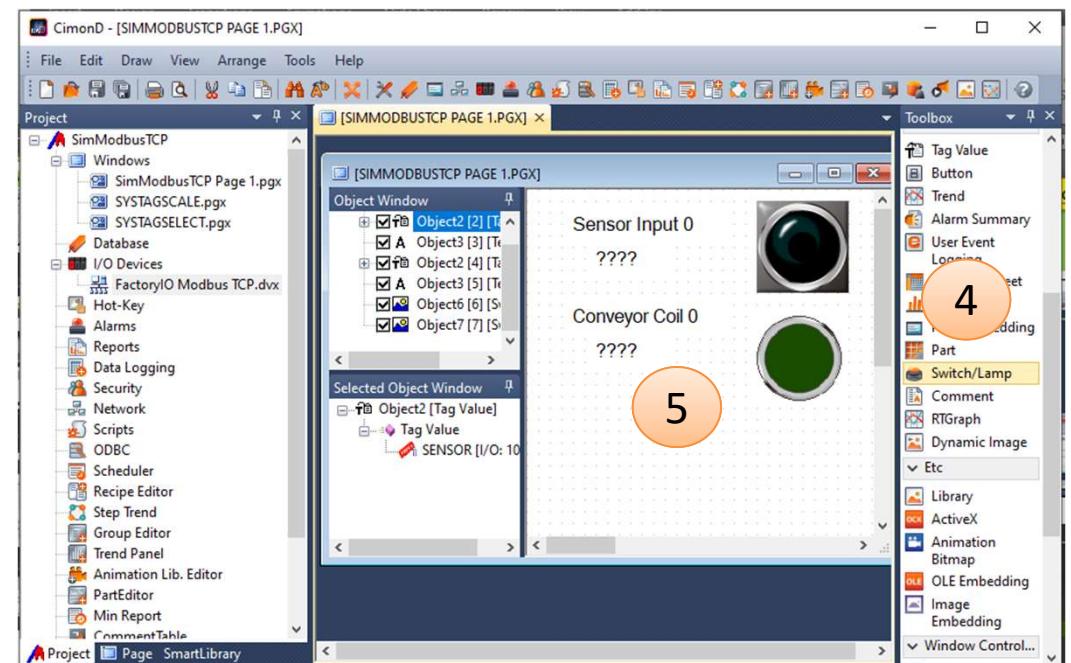
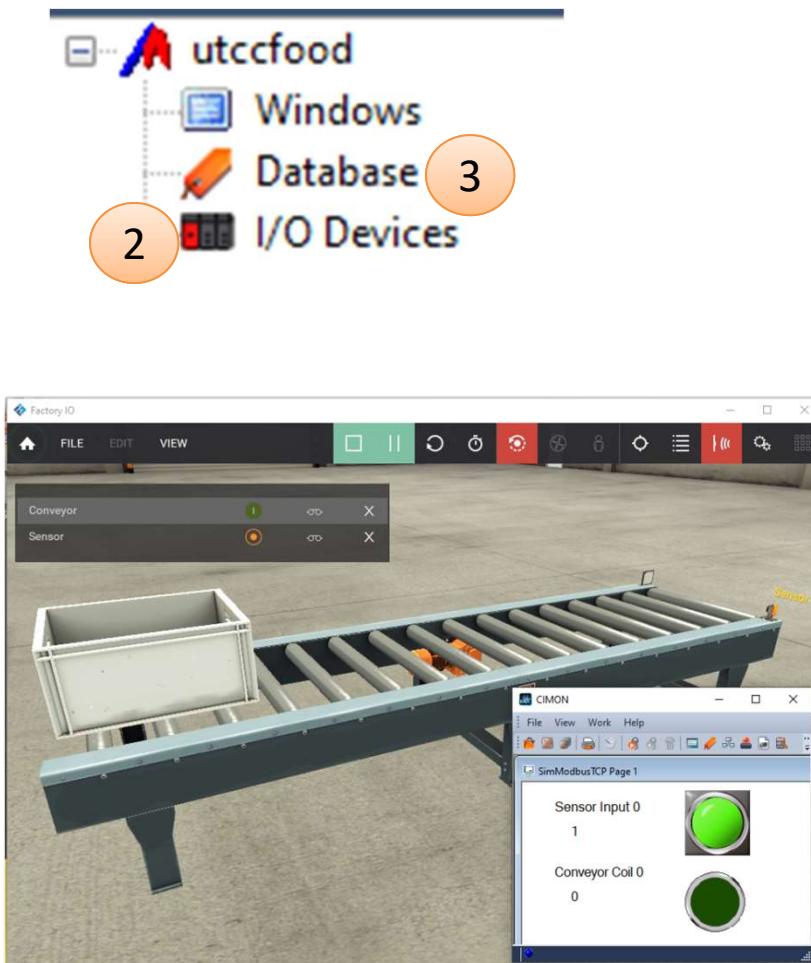
Address: C H D I/O Holding Regs (400000) Fmt: decimal +/-

Address	+0	+1	+2	+3	+4	+5
400001-400010	254	565	0	0	0	0
400011-400020	0	0	0	0	0	0

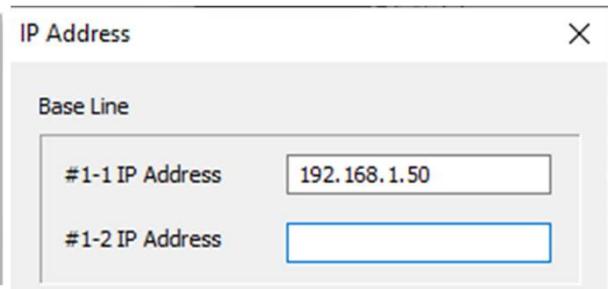
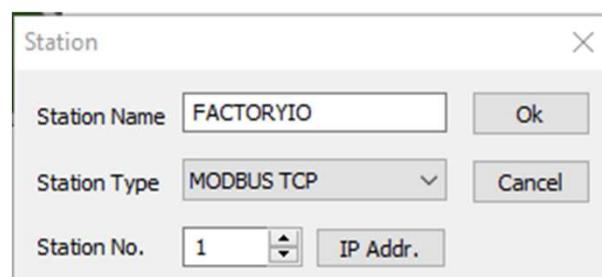
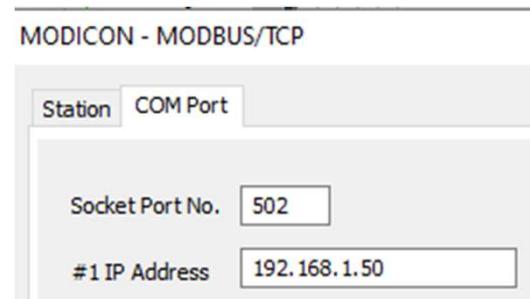
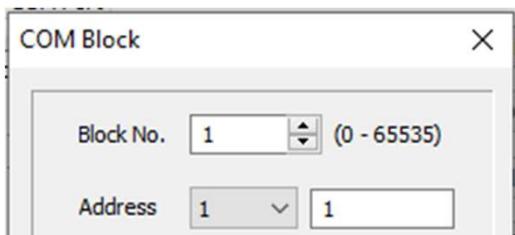
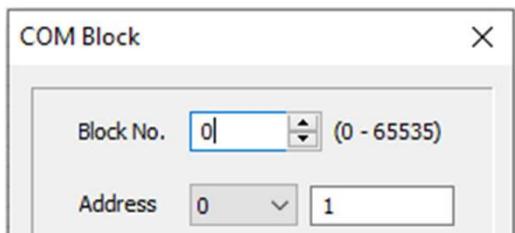
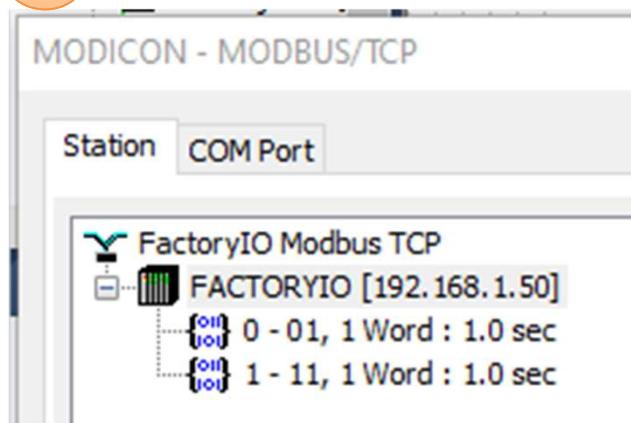
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45

SCADA Demo

- Cimon - FactoryIO



1



2

DATABASE [SimModbusTCP.dbx]

	Name	Type	I/O device	I/O address	Initial v
...> SimModbusTCP	SENSOR	Digital Tag	FACTORYIO MODBUS TCP.FACTORYIO	100001	0
	CONVEYOR	Digital Tag	FACTORYIO MODBUS TCP.FACTORYIO	000001	0

Sensor Input 0

Conveyor Coil 0

Object Config

Name: Object2

Type: Tag Value

Config

- Style
- Visible Blink
- V-Size H-Size
- V-Move H-Move
- V-Fill H-Fill
- V-Slider H-Slider
- Color Rotate
- Touch EntryData
- Tag Value

Dynamic Tag

Tag name: SENSOR

Preview: 12345.12345

Display Format: ????

Toolbox

- Advanced
 - Tag Value
 - Button
 - Trend
 - Alarm Summary
 - User Event Logging
 - Log Data Sheet
 - Chart
 - Page Embedding
 - Part
 - Switch/Lamp
 - Comment
 - RTGraph
 - Dynamic Image

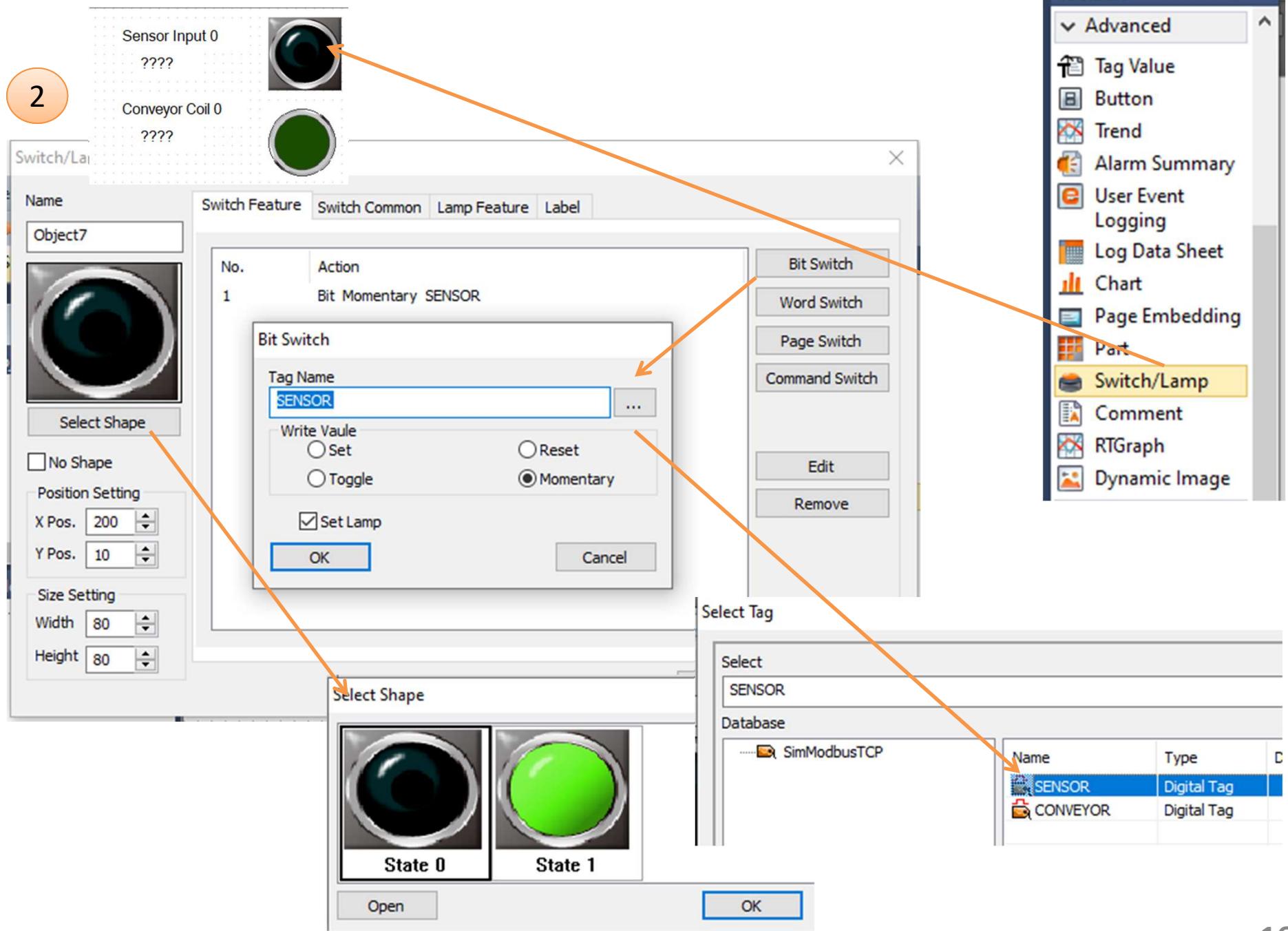
Select Tag

Select: SENSOR

Database

SimModbusTCP

Name	Type
SENSOR	Digital Tag
CONVEYOR	Digital Tag



Sensor Input 0
????



Conveyor Coil 0
????



Switch/

Name: Object6

Switch Feature | Switch Common | Lamp Feature | Label

No. Action

1 Bit Toggle CONVEYOR

Bit Switch

Tag Name: CONVEYOR

Write Value:

- Set
- Toggle
- Reset
- Momentary

Set Lamp

OK Cancel

Bit Switch Word Switch Page Switch Command Switch Edit Remove

Select Shape

No Shape

Position Setting

X Pos.: 200 Y Pos.: 110

Size Setting

Width: 74 Height: 74

Select Tag

Select: CONVEYOR

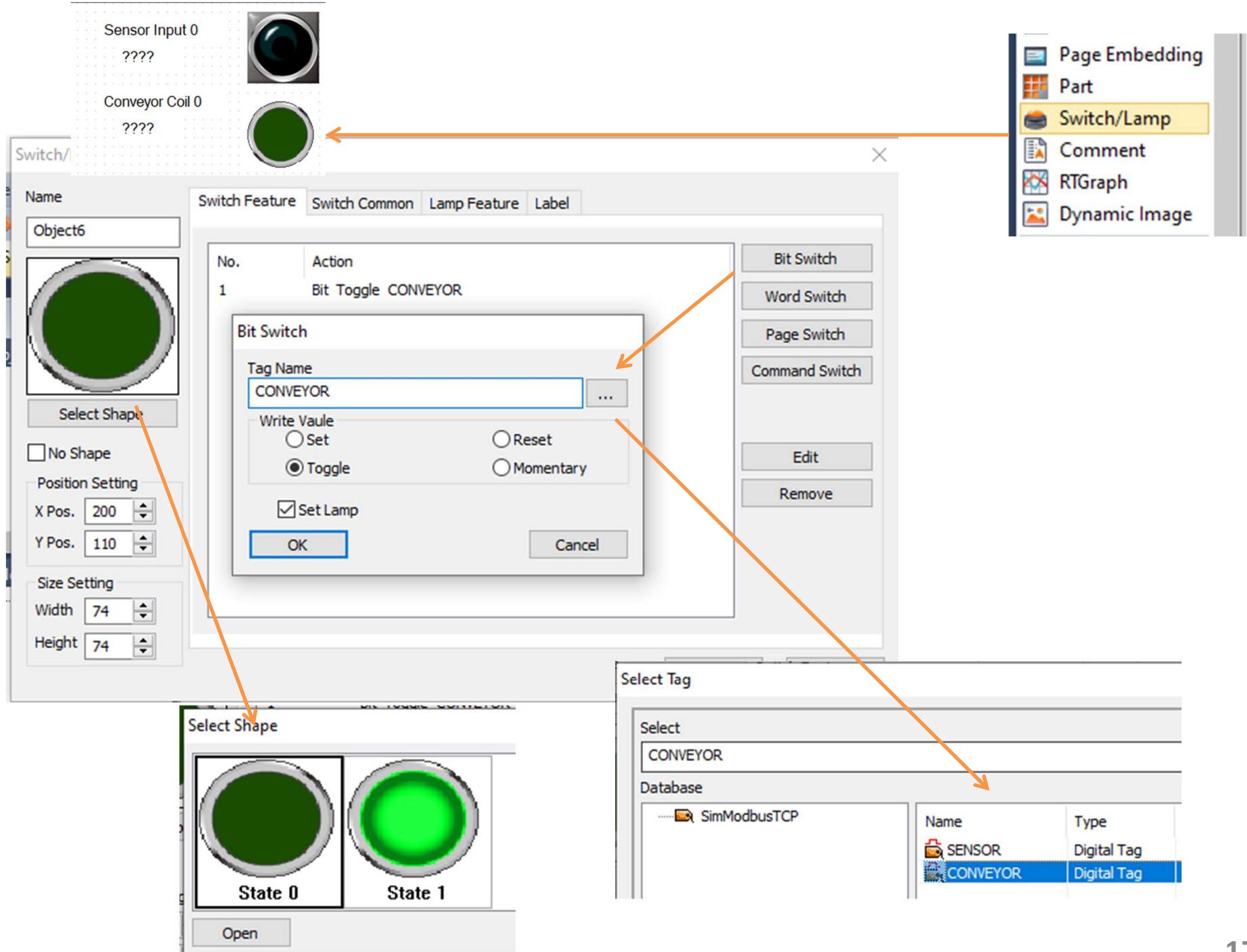
Database: SimModbusTCP

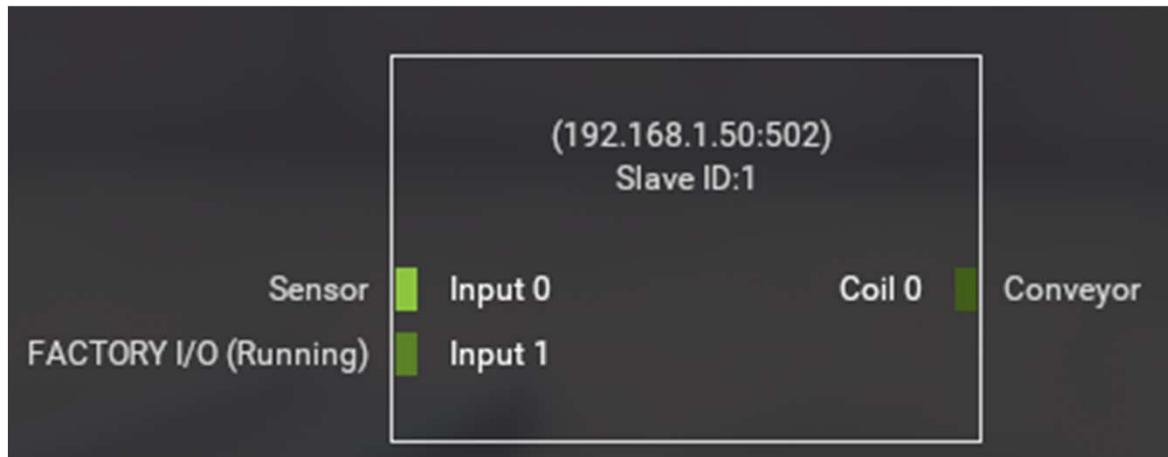
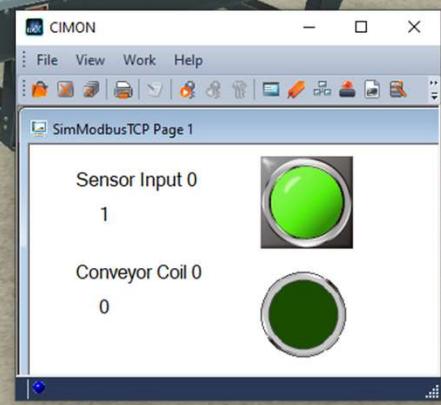
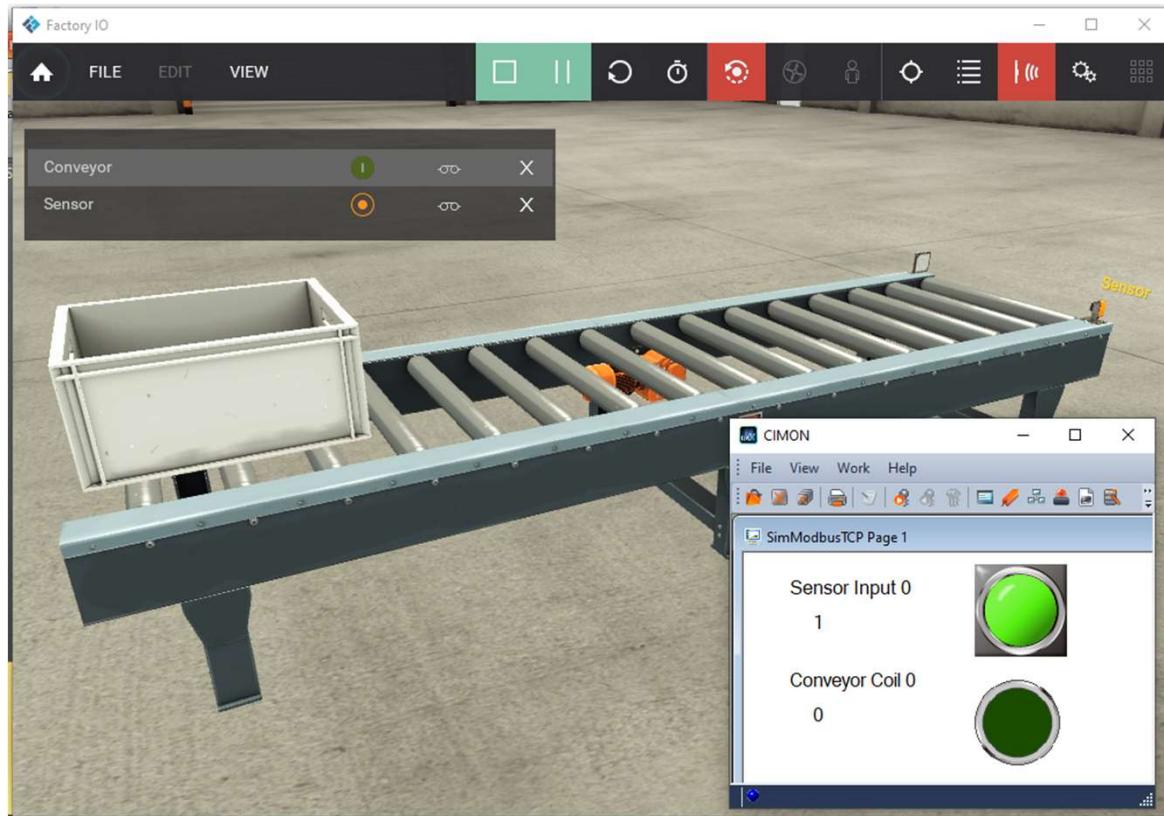
Name	Type
SENSOR	Digital Tag
CONVEYOR	Digital Tag

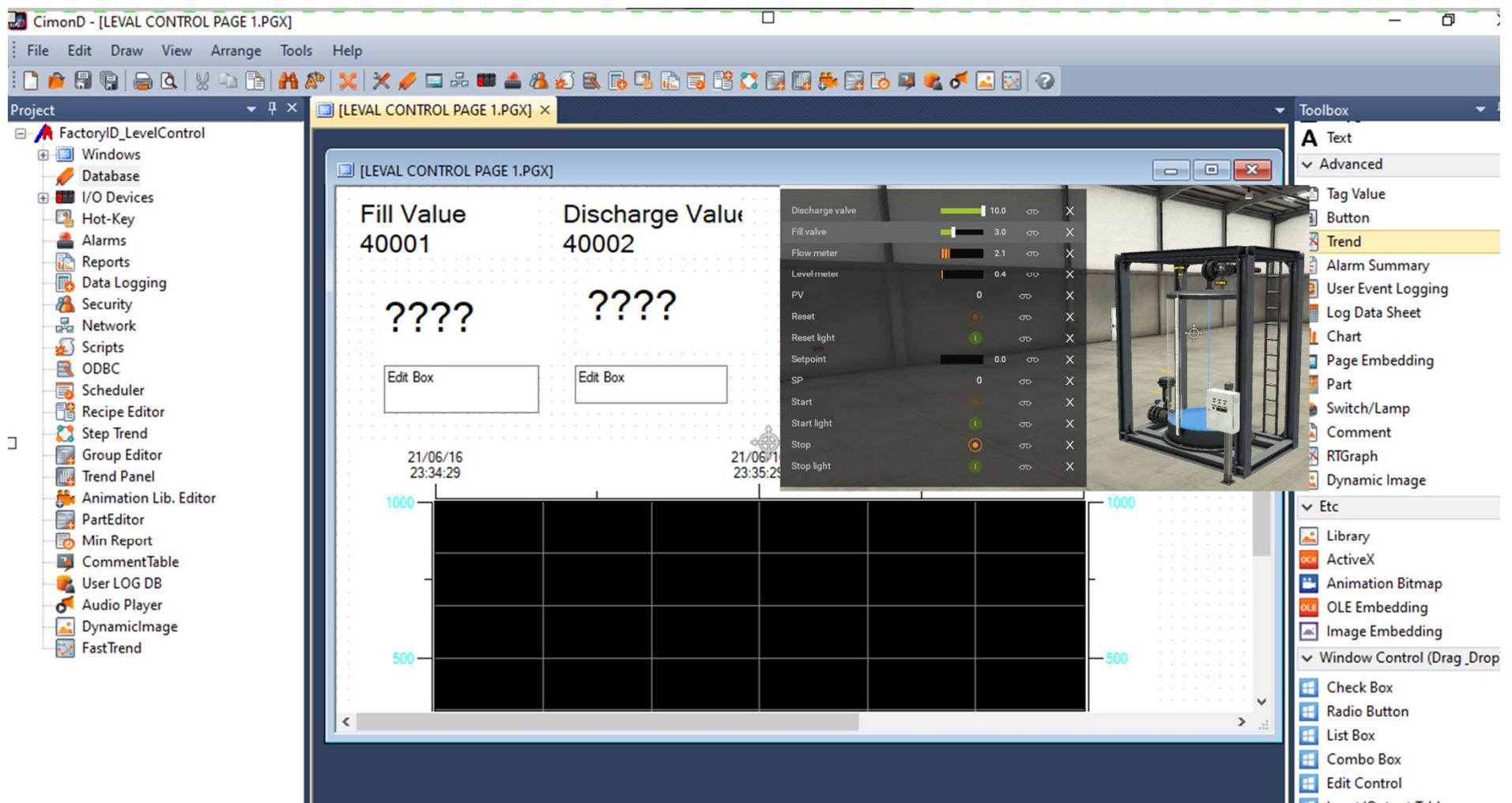
Page Embedding Part Switch/Lamp Comment RTGraph Dynamic Image

State 0 State 1

Open





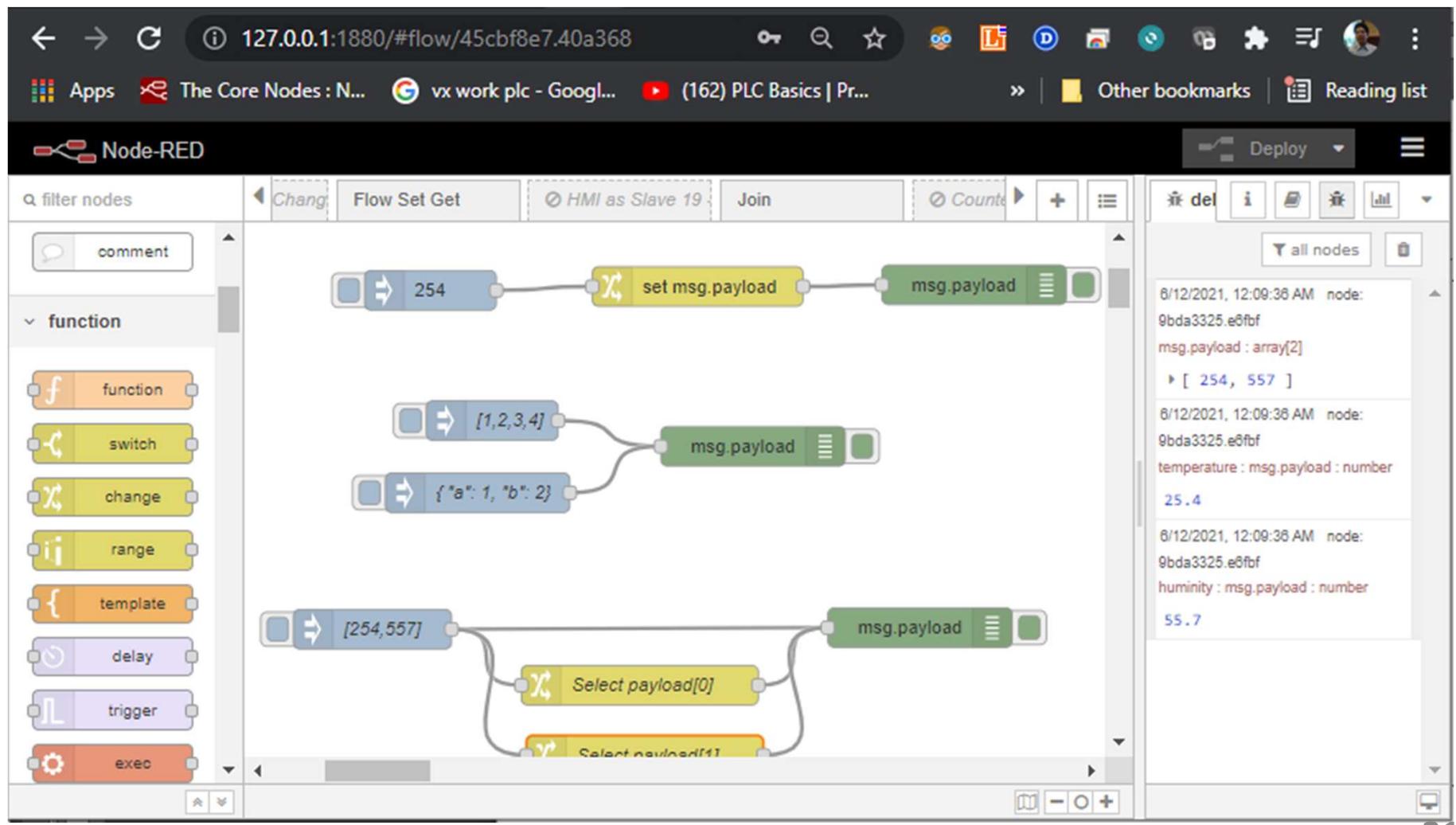


Node-Red

- *Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.*
- Cmd
 - Node-red
- <http://localhost:1880>
- 127.0.0.1:1880

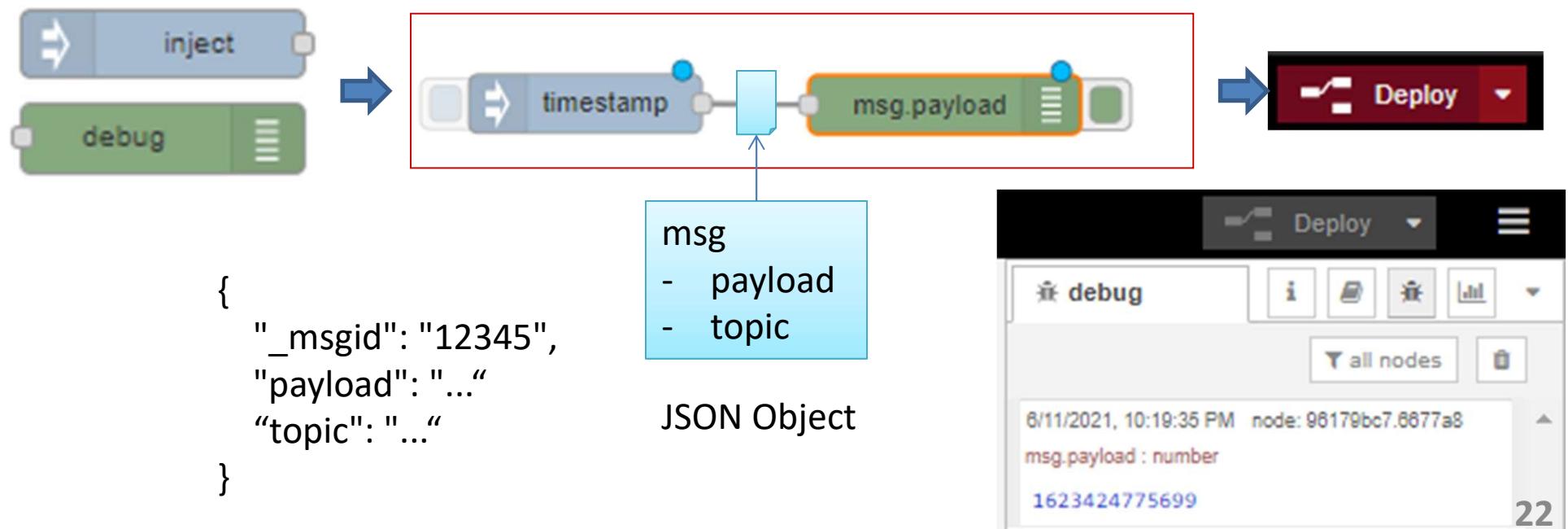


Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.



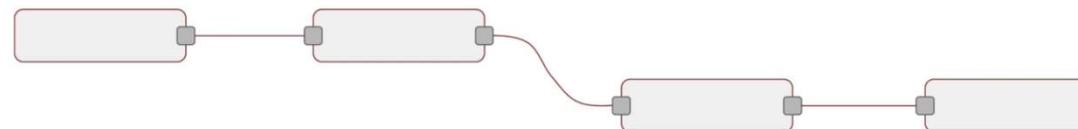
Basic Nodes

- Input Node นำข้อมูลเข้า node-red
- Function Node กำหนดการทำงาน เปลี่ยนแปลงค่า
- Output Node แสดงผลข้อมูลในรูปแบบต่าง



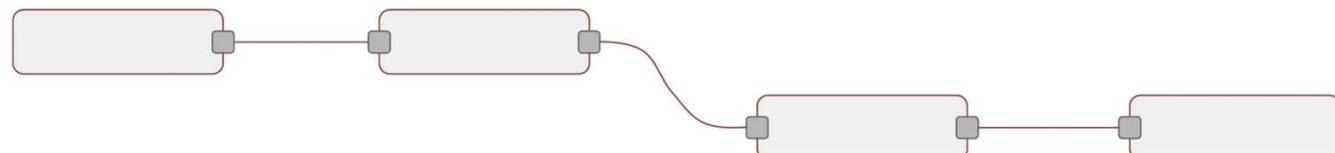
JSON Object (Data Transfer)

```
{ payload: "sunny",  
topic: "weather/uk" }
```



- Boolean - true, false
- Number - eg 0, 123.4
- String - "hello"
- Array - [1,2,3,4]
- Object - { "a": 1, "b": 2}
- Null

```
{ payload: "sunny",  
topic: "weather/uk",  
color: "red",  
temp: 20.2 }
```



JSON Object Examples

ชื่อตัวแปร ค่าตัวแปร

{ "name" : "John" } ข้อความ

{ "age" : 30 } ตัวเลข

{"name" : "John" , "age" : 30 } หลายตัวแปร

Array Object

[{ "name": "John", "age": 30 } , { "name": "jenny", "age": 22 }]

payload and topic properties (default)

YouTube Node-Red - 01 Basic Debug

The screenshot shows the Node-RED interface with a flow diagram and an open configuration dialog for an 'inject' node.

Flow Diagram:

- A 'timestamp' node is connected to an 'inject' node.
- An 'inject' node is connected to a blank blue placeholder node.

msg
- payload
- topic

Edit inject node

Properties:

- Name:** Name
- msg. payload**: timestamp
- msg. topic**: a_z

msg. topic dropdown options (highlighted with a red bracket):

- msg.
- flow.
- global.
- a_z string
- b_g number
- boolean
- { } JSON
- 01 buffer
- 02 timestamp
- J expression
- \$ env variable

Repeat: none

Enabled:

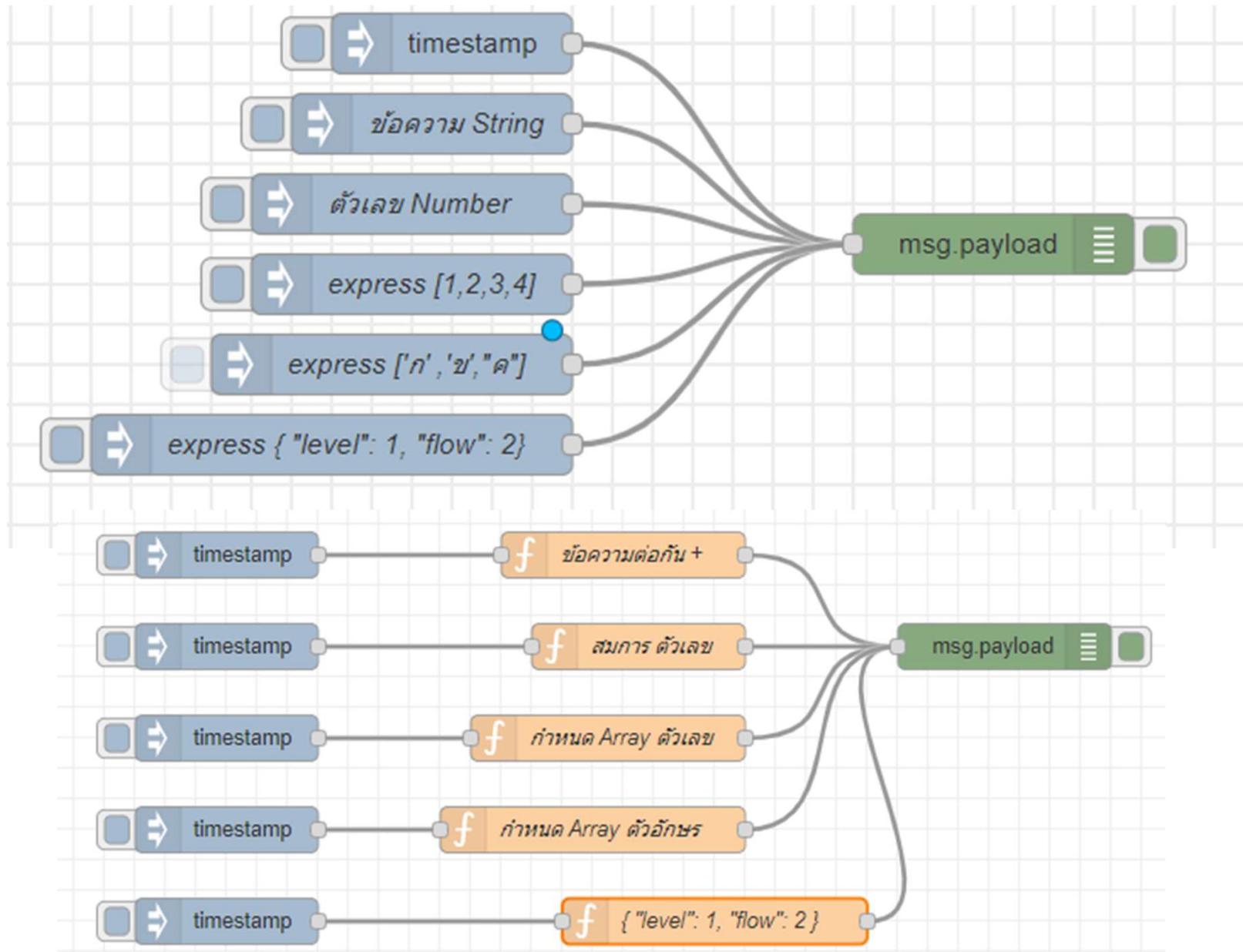
Notes:

- กำหนดชื่อ module
- กำหนดเวลาทำซ้ำ

Repeat dropdown options (highlighted with a red border):

- none
- interval
- interval between times
- at a specific time

25



กำหนดค่า payload และการแสดงค่า Debug

JavaScript Types

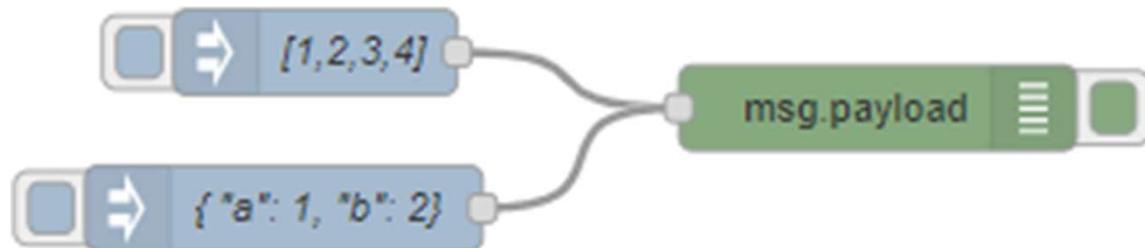
boolean : true / false

number : 123

string : "hello"

array : [1, 2, 3, 4]

object : { "color": "red" }



```
6/11/2021, 11:30:40 PM node: e  
msg.payload : array[4]  
▼ array[4]  
 0: 1  
 1: 2  
 2: 3  
 3: 4  
  
6/11/2021, 11:30:42 PM node: e  
msg.payload : Object  
▼ object  
 a: 1  
 b: 2
```

◆ Name [1,2,3,4] [1,2,3,4]

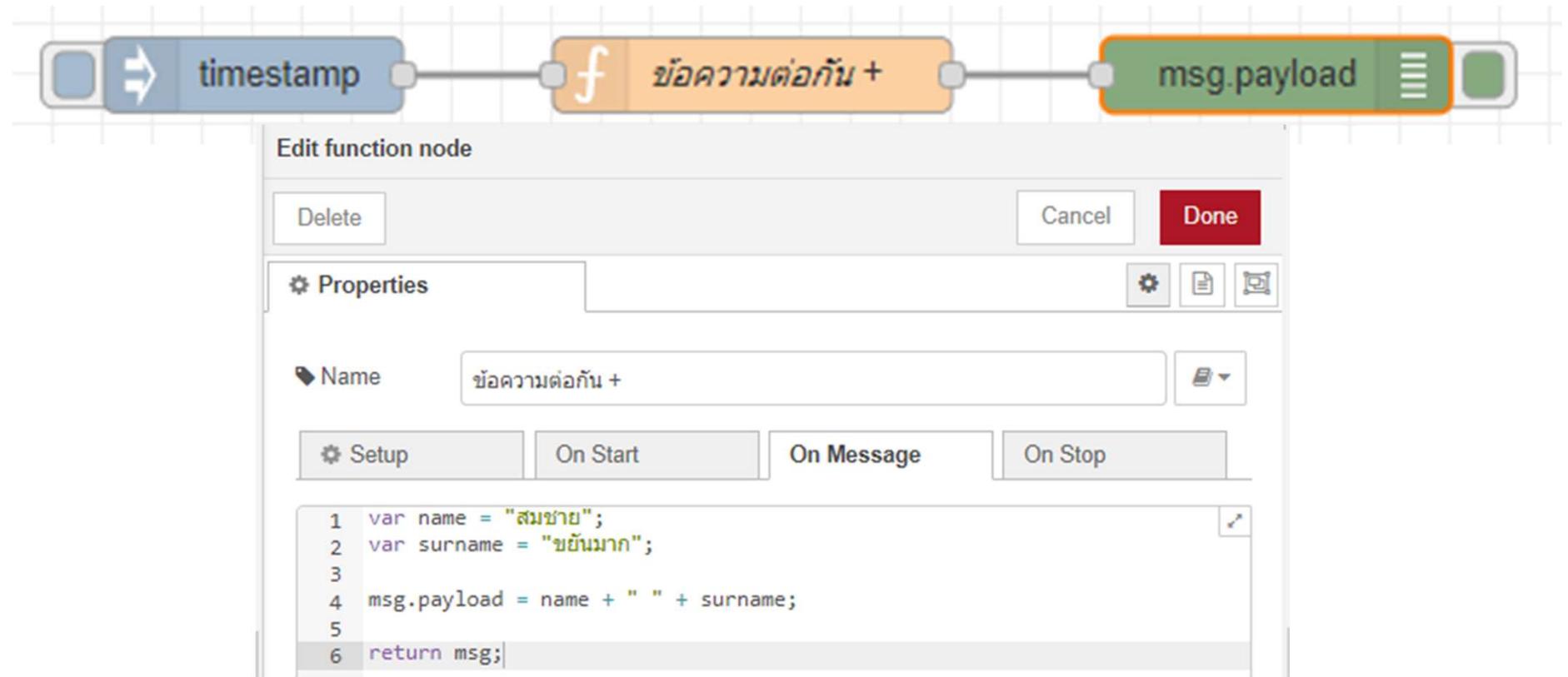
≡ msg. payload = ▼ ↴ [1,2,3,4]

≡ msg. topic = ▼ ↴ a_z

◆ Name { "a": 1, "b": 2} { "a": 1, "b": 2 }

≡ msg. payload = ▼ ↴ { "a": 1, "b": 2}

≡ msg. topic = ▼ ↴ a_z



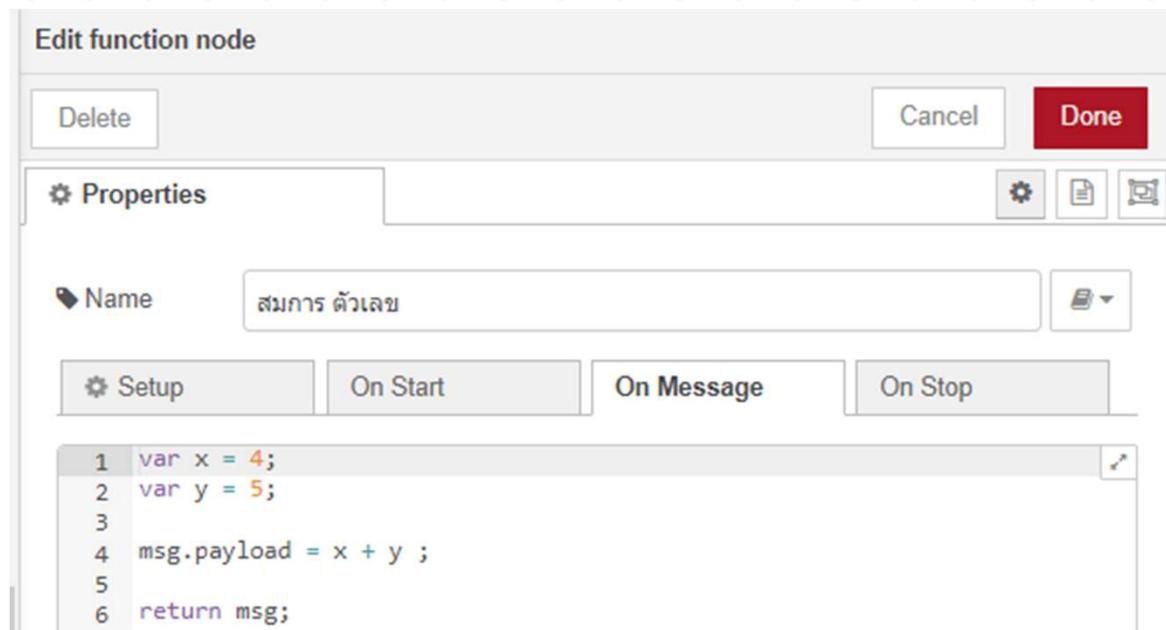
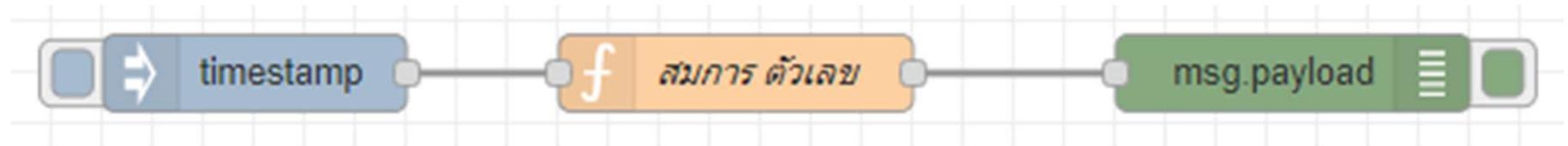
Msg
.topic
.payload

Msg : { topic , payload}

var name = "สมชาย" ;
var surname = 'ขยันมาก' ;

msg.payload = name + " " + surname ;

return msg ;



var x = 4 ;

var y = 5 ;

msg.payload = x + y ;

return msg ;



```

msg.payload = [1,2,3,4];
return msg;
  
```

```

msg.payload = ['ก','ข','ค'];
return msg;
  
```

```

msg.payload = { "level": 1, "flow": 2 };
return msg;
  
```

8/16/2021, 9:18:57 PM nod
msg.payload : array[4]

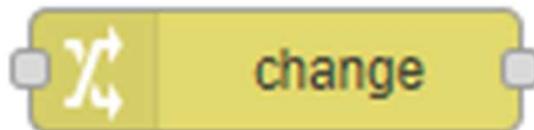
- ▼ array[4]
- 0: 1
- 1: 2
- 2: 3
- 3: 4

8/16/2021, 9:18:58 PM nod
msg.payload : array[3]

- ▼ array[3]
- 0: "ก"
- 1: "ข"
- 2: "ค"

8/16/2021, 9:18:59 PM nod
msg.payload : Object

- ▼ object
- level: 1
- flow: 2



- Set a property to a value,
- Change a String property by performing a search and replace,
- Delete a property,
- Move a property.

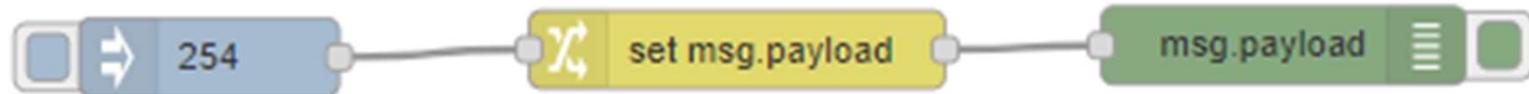
The screenshot shows a node-red flow editor with two 'Set' nodes. The first 'Set' node has 'Set' selected in the dropdown and 'msg.payload' in the target field. In the 'to' field, there is a dropdown menu with 'msg.' selected, which is highlighted with a red border. Other options in the dropdown include 'flow.', 'global.', 'string', 'number', 'boolean', 'JSON', 'buffer', 'timestamp', and 'expression'. The second 'Set' node has 'Set' selected in the dropdown and 'msg.payload.temperature_c' in the target field. In the 'to' field, it contains the expression `J: (payload.temperature-32)*5/9`. A preview window on the right shows the resulting JSON object:

```
{  
  "payload": {  
    "temperature": 90,  
    "temperature_c": 32.22222  
  }  
}
```

- Boolean - `true, false`
- Number - eg `0, 123.4`
- String - `"hello"`
- Array - `[1,2,3,4]`
- Object - `{ "a": 1, "b": 2}`
- Null

Change กำหนดค่า payload / 10

GitHub Node-Red - 02 Basic Change



payload (254) - input
Change (254/10)
Payload 25.4 - output

Edit inject node

Delete

Properties

Name Name Number

msg. payload = 245 245

msg. topic = a_z

Edit change node

Delete

Properties

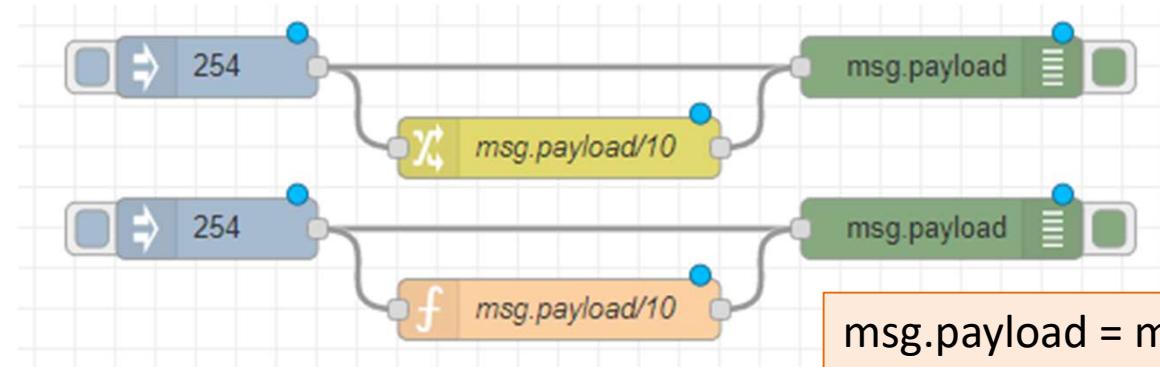
Name Name

Rules

Set msg. payload to payload/10

Payload/10

Express



`msg.payload = msg.payload/10
;return msg;`

Edit change node

Delete Cancel Done

Properties

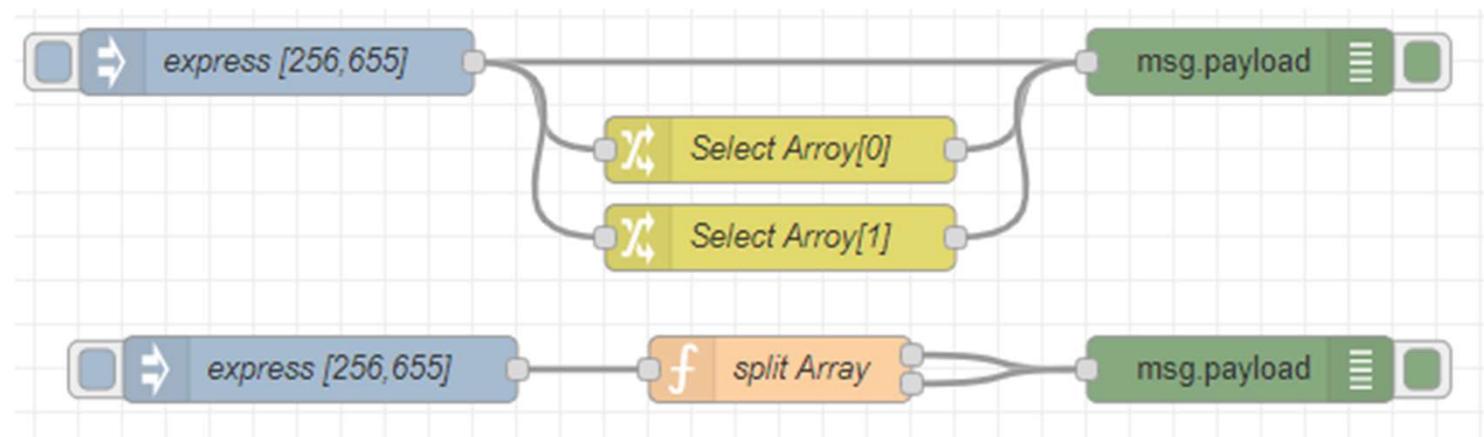
Name: msg.payload/10

Rules

Set msg. payload to J: payload/10

6/16/2021, 9:25:53 PM node:
1d98376f.4e0a19
msg.payload : number
254

6/16/2021, 9:25:53 PM node:
1d98376f.4e0a19
msg.payload : number
25.4



Set msg. payload to J: payload[0]

6/16/2021, 9:26:23 PM node:
89d312c3.58257
msg.payload : array[2]
▶ [256, 655]

Set msg. payload to J: payload[1]

6/16/2021, 9:26:23 PM node:
89d312c3.58257
msg.payload : number
256

Setup On Start Outputs 2

```

var msg1 = {};
var msg2 = {};
msg1.payload = msg.payload[0];
msg2.payload = msg.payload[1];
return [ msg1, msg2 ];

```

6/16/2021, 9:26:23 PM node:
89d312c3.58257
msg.payload : number
655

```

graph TD
    In1["{"level": 10, "flow": 20}"] --> S1[Select payload.level]
    In1 --> S2[Select payload.flow]
    S1 --> Out1["msg.payload"]
    S2 --> Out1
    In2["{"level": 10, "flow": 20}"] --> Split["split value"]
    Split --> Out2["msg.payload"]
    Split --> Out3["msg.payload"]

```

Set Node Configuration:

- Top Set Node: Set msg. payload to J: payload.level
- Bottom Set Node: Set msg. payload to J: payload.flow

Terminal Log Output:

```

6/16/2021, 9:34:48 PM node:
61f802b9.82c3ac
msg.payload : Object
▶ { level: 10, flow: 20 }

6/16/2021, 9:34:48 PM node:
61f802b9.82c3ac
msg.payload : number
10

6/16/2021, 9:34:48 PM node:
61f802b9.82c3ac
msg.payload : number
20

```

Code Block:

```

var msg1 = {};
var msg2 = {};
msg1.payload = msg.payload.level;
msg2.payload = msg.payload.flow ;
return [ msg1, msg2 , null ];

```

Change กำหนดค่า topic

GitHub Node-Red - 02 Basic Change



```
6/11/2021, 11:56:04 PM node: 96179b  
first topic : msg.payload : string[11]  
"Hello World"  
6/11/2021, 11:56:04 PM node: 96179b  
second topic : msg.payload : string[11]  
"Hello World"
```

Edit inject node

Delete

Properties

Name Name

msg. payload = a_z Hello World

msg. topic = a_z first topic

Edit change node

Delete

Properties

Name Name

Rules

Set msg. topic to second topic



Rules

Change **msg. payload**

Search for **a_z @1**

Replace with **ส้มชัย**

Change **msg. payload**

Search for **a_z @2**

Replace with **ขยันมาก**

+ add

6/16/2021, 9:44:06 PM node:

1cc46d5c.078543

msg.payload : string[16]

"My name is @1 @2"

6/16/2021, 9:44:06 PM node:

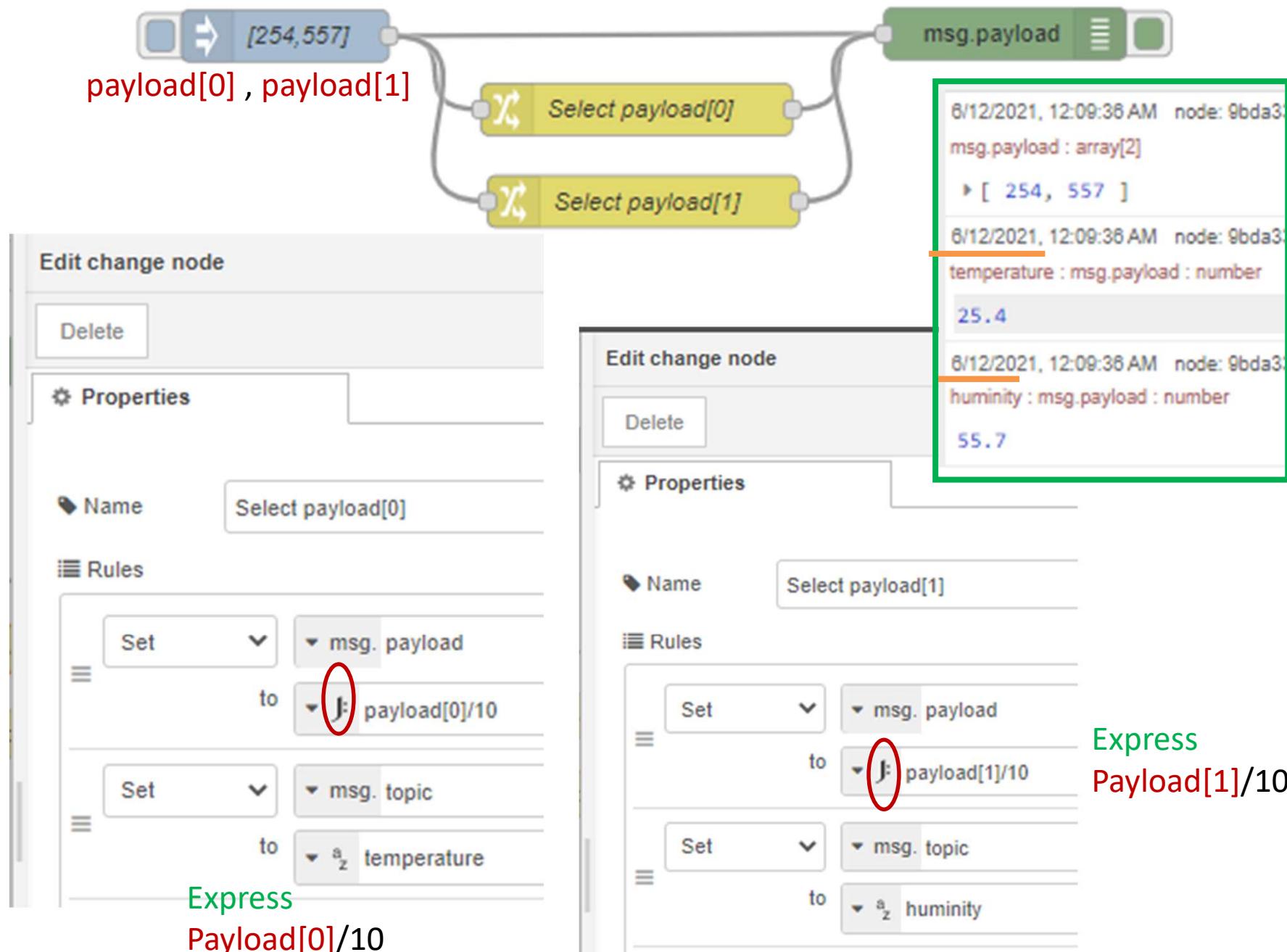
1cc46d5c.078543

msg.payload : string[24]

"My name is ส้มชัย ขยันมาก"

แยกค่าเป็น สองทางและกำหนด topic ใหม่

GitHub Node-Red - 02 Change-Function





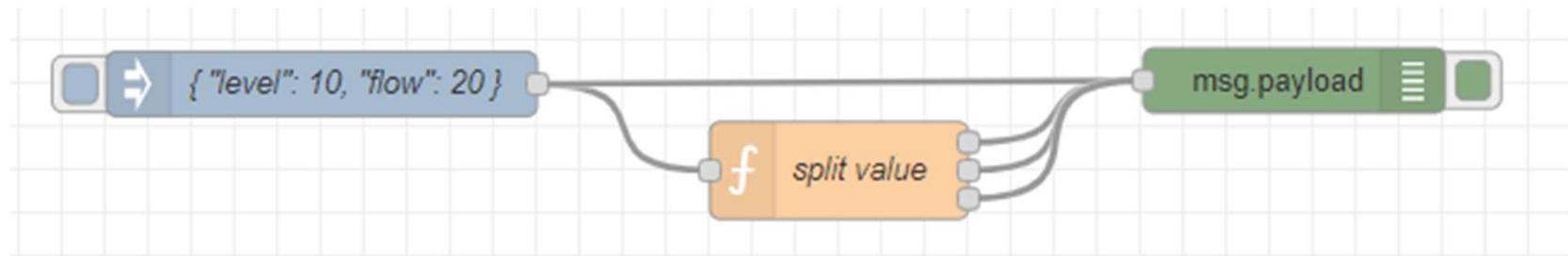
```

if (msg.payload < msg.topic ) return null ;

return msg;

```

6/18/2021, 9:46:59
 PM node:
 17f32694.c92379
 2 : msg.payload : number
 6



Setup
 On Start
 On Message
 Outputs

```

var msg1 = {};
var msg2 = {};
var msg3 = null;

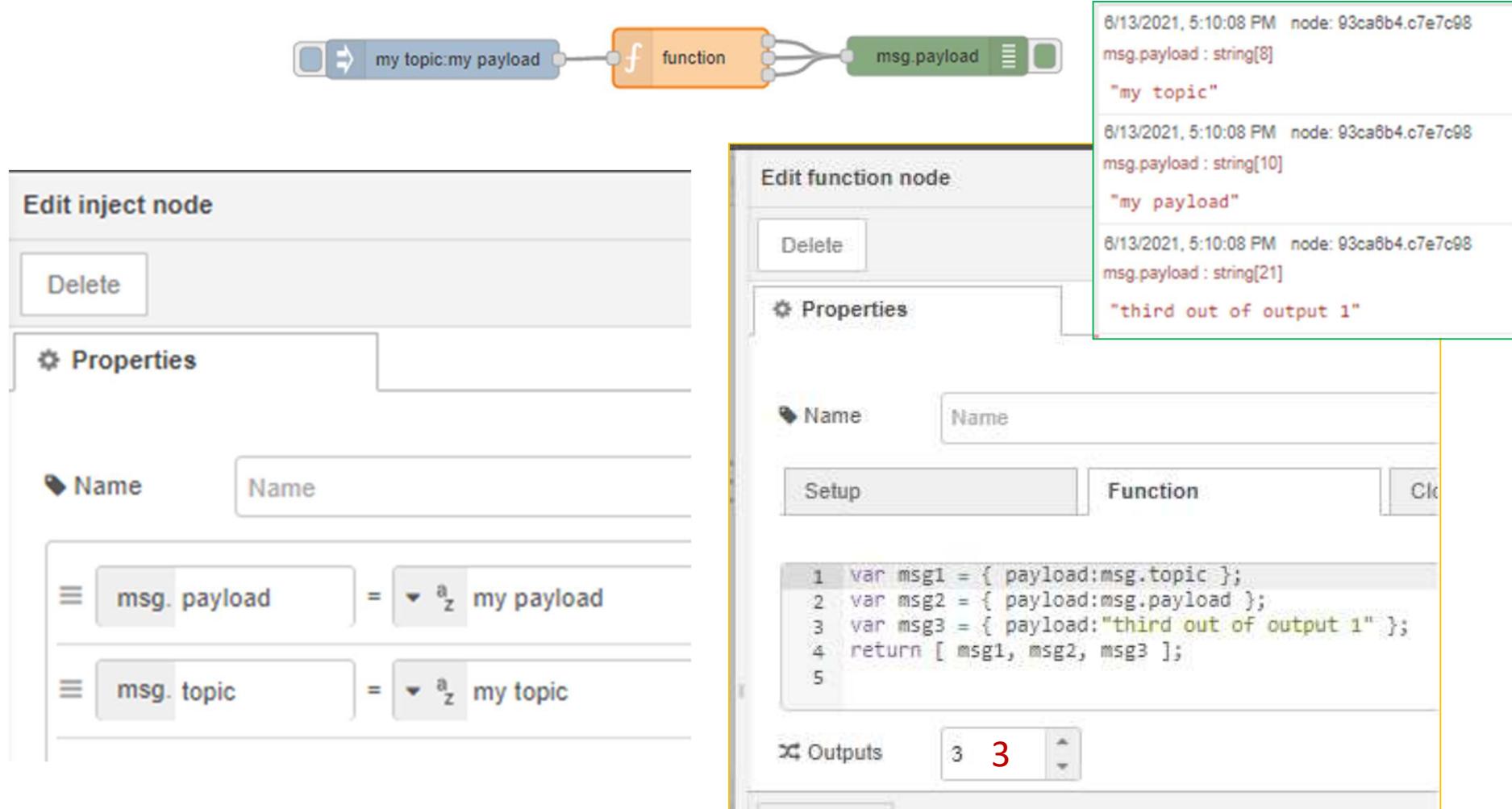
msg1.payload = msg.payload.level;
msg2.payload = msg.payload.flow ;

return [ msg1, msg2 ,msg3 ];
  
```

6/16/2021, 9:49:22 PM node:
 8b3d7152.44a13
 msg.payload : Object
 ▶ { level: 10, flow: 20 }

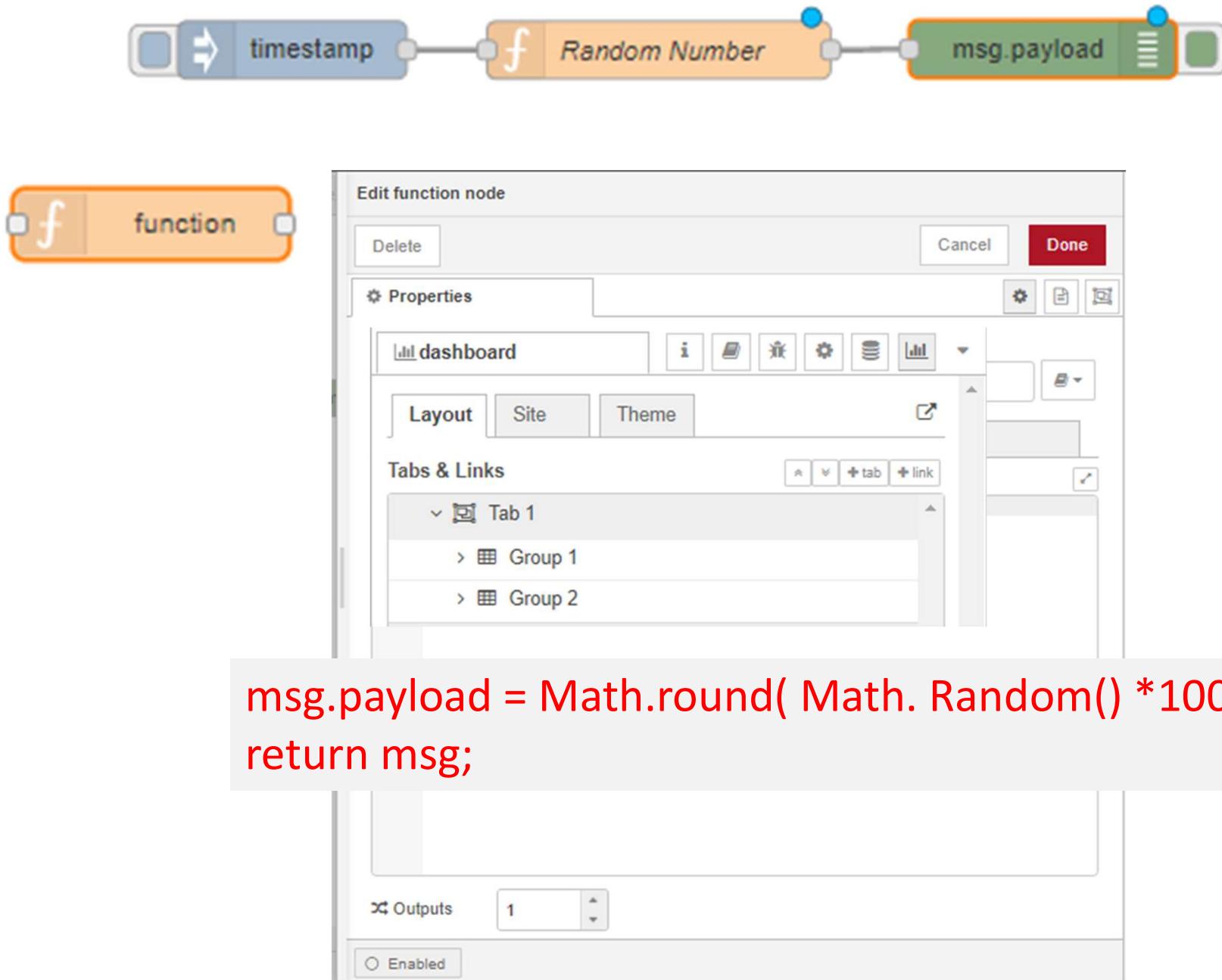
 6/16/2021, 9:49:22 PM node:
 8b3d7152.44a13
 msg.payload : number
 10

 6/16/2021, 9:49:22 PM node:
 8b3d7152.44a13
 msg.payload : number
 20

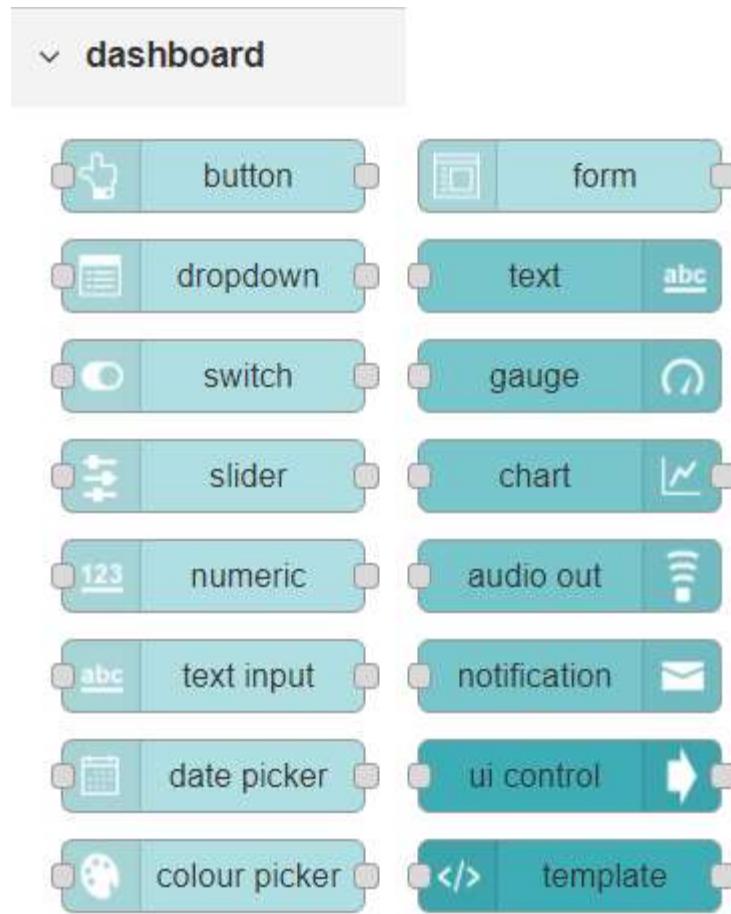


```

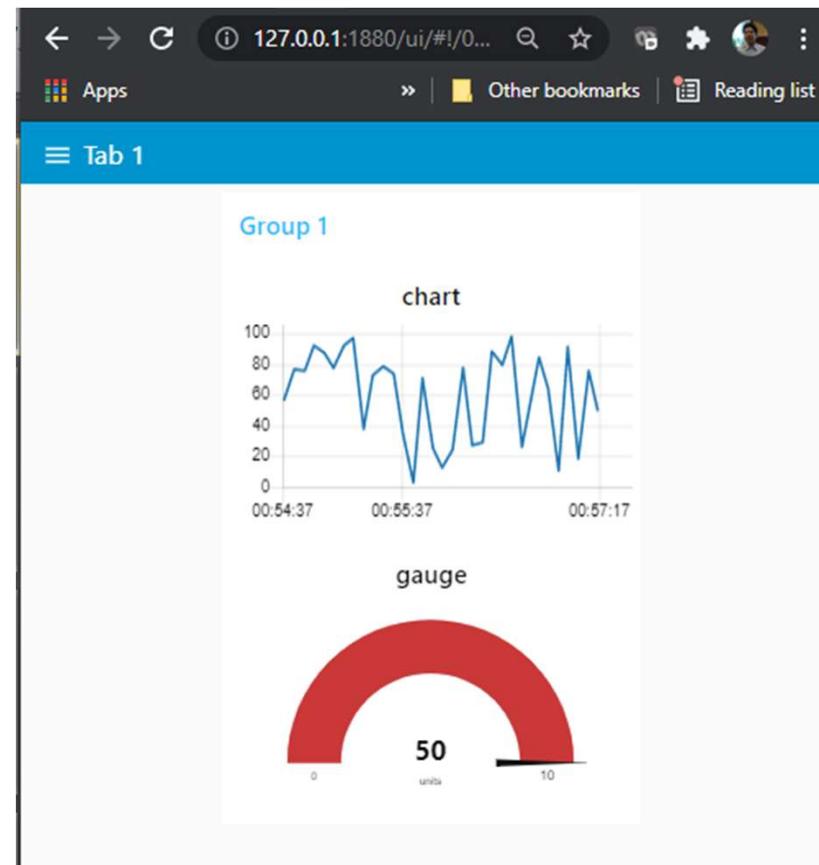
var msg1 = { payload:msg.topic };
var msg2 = { payload:msg.payload };
var msg3 = { payload:"third out of output 1" };
return [ msg1, msg2, msg3 ];
    
```

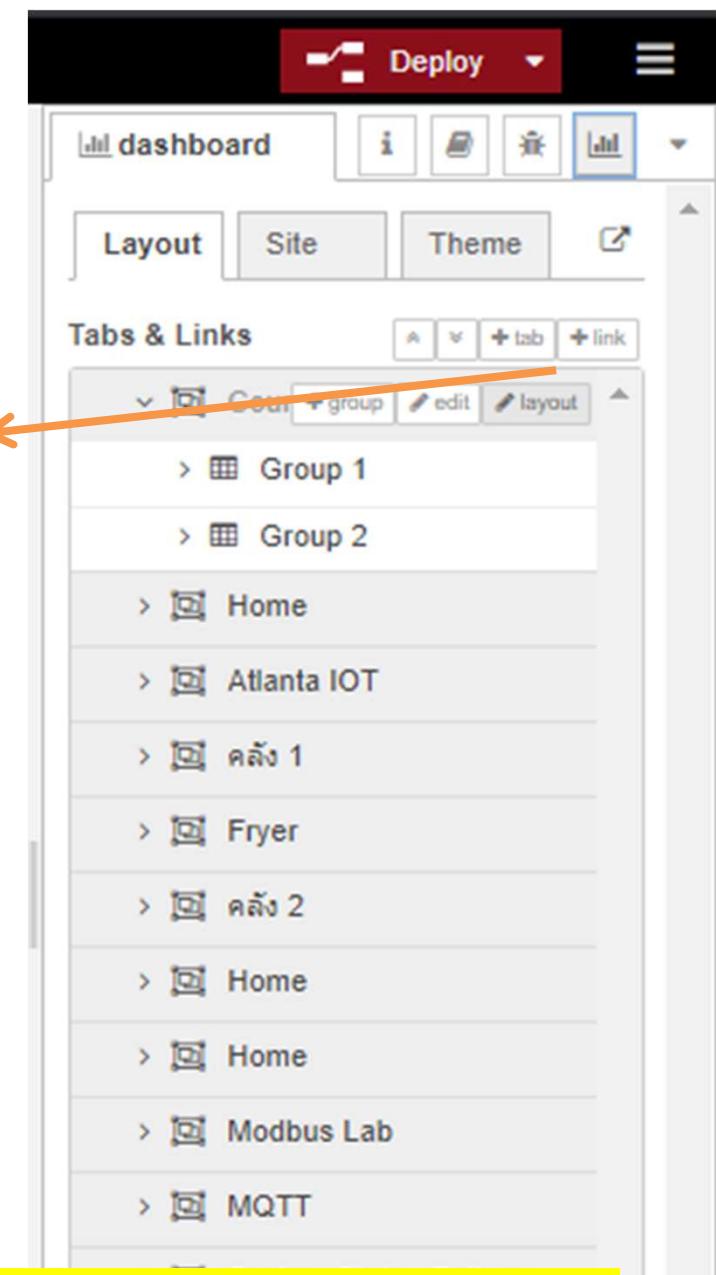
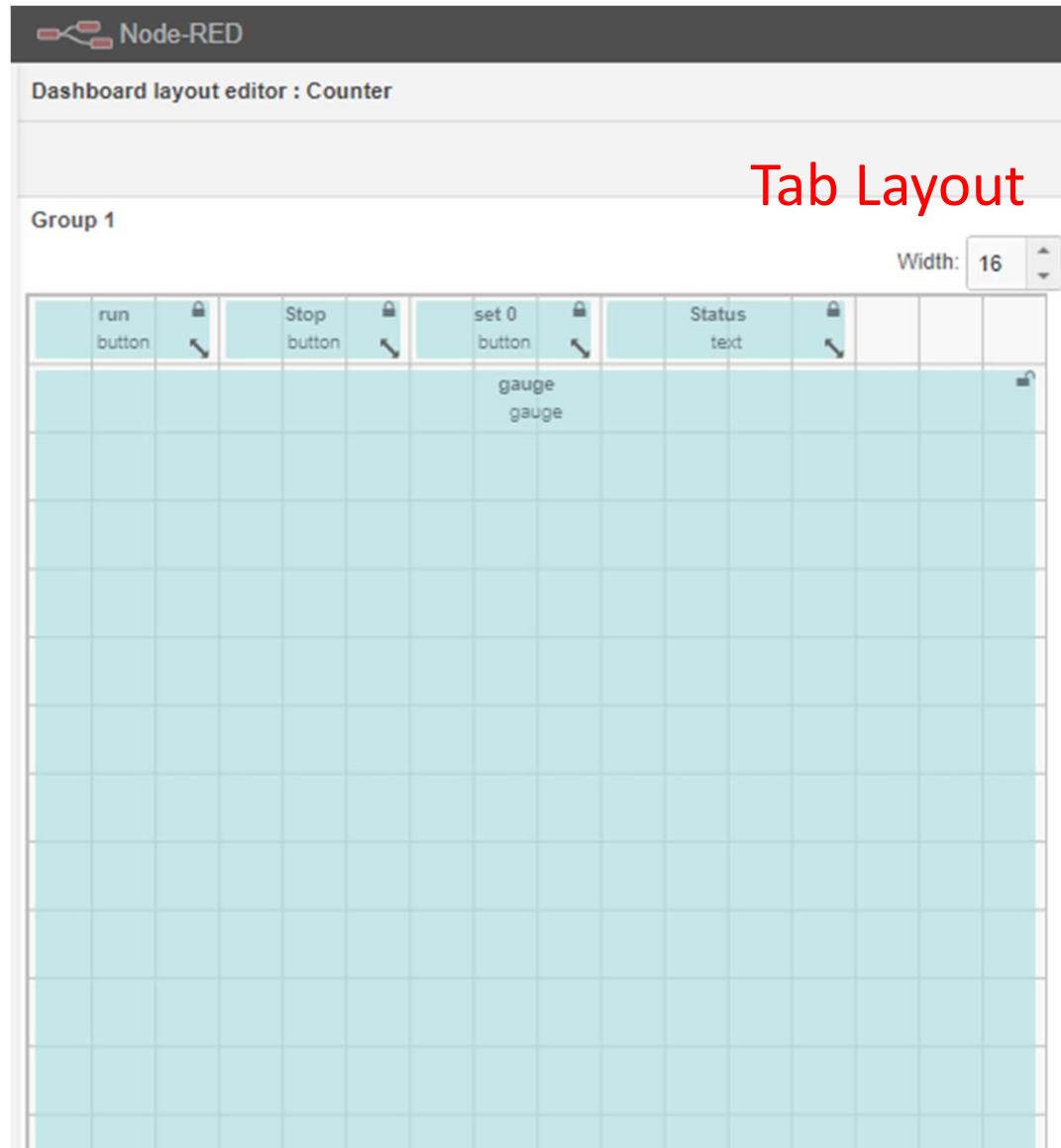


Dashboard Module



<http://127.0.0.1:1880/>

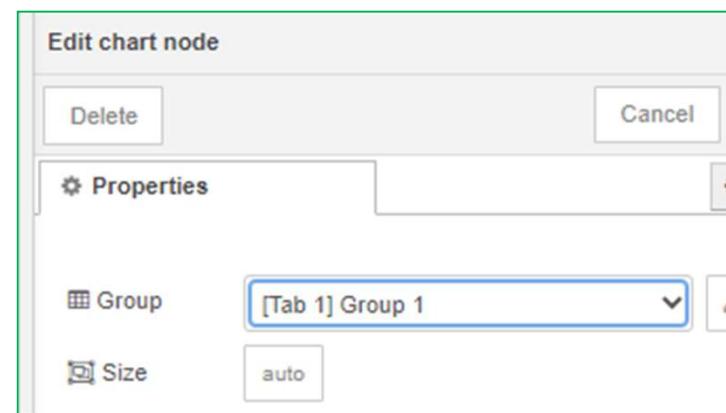
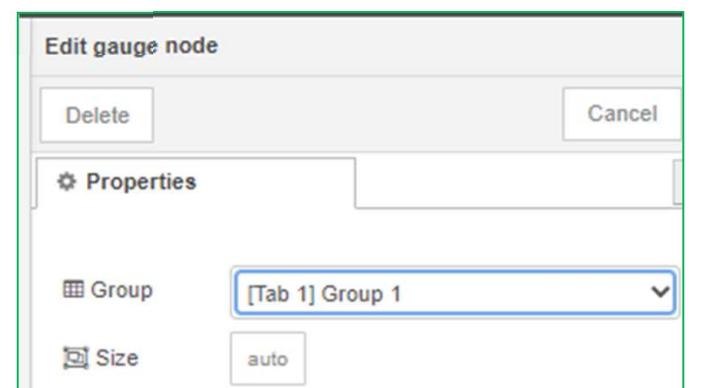
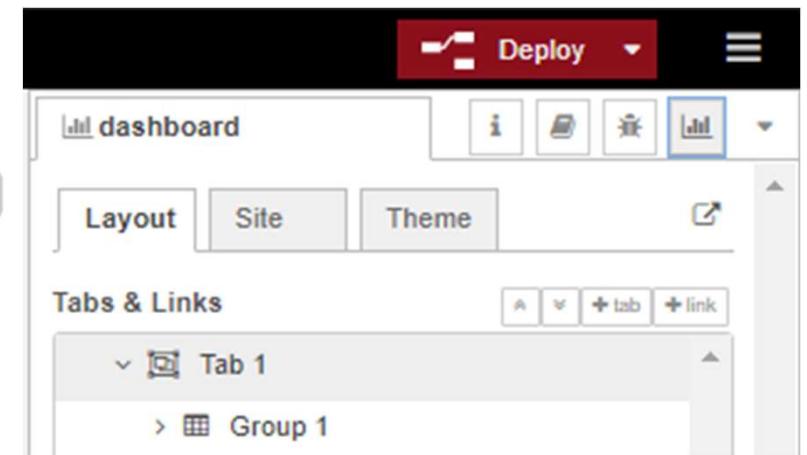
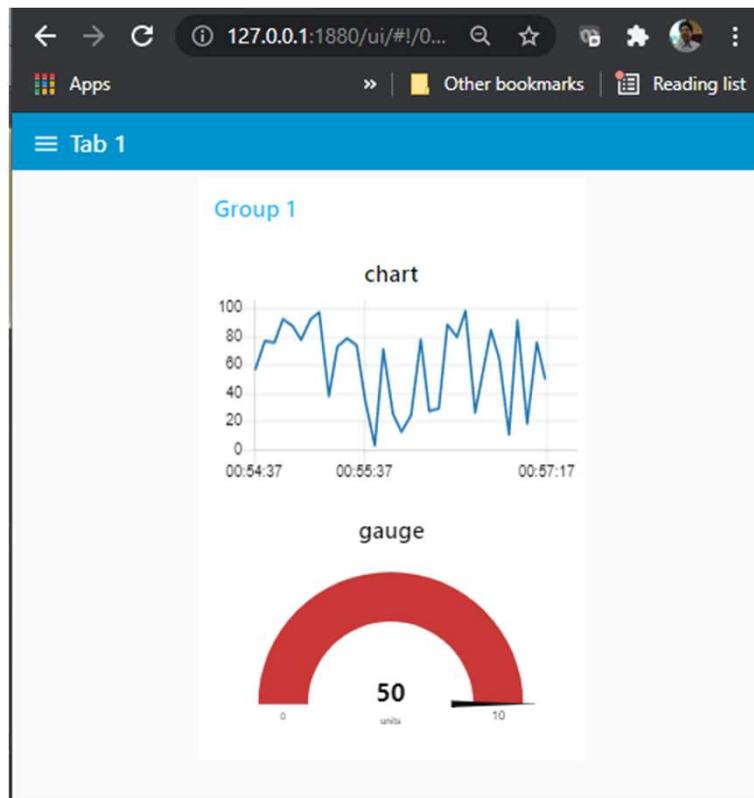


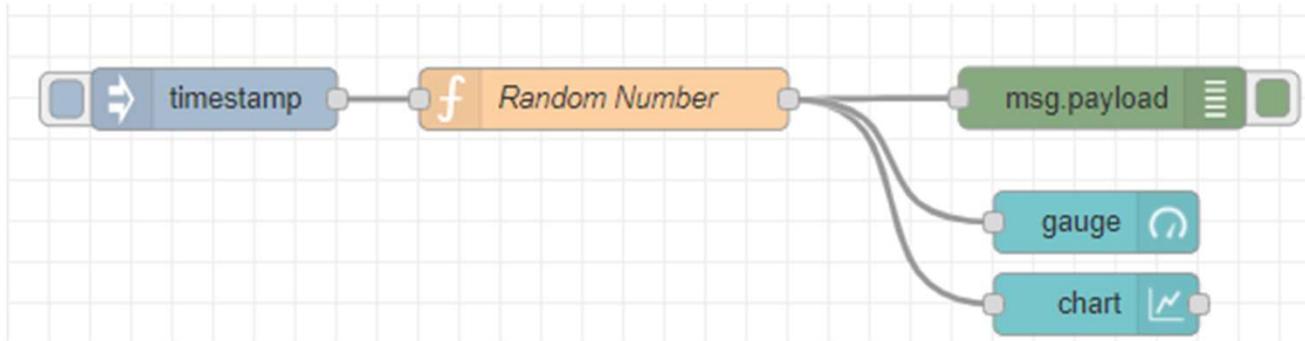


Basic Dashboard Control

GitHub Node-Red 03 Basic Dashboard

```
msg.payload = Math.round( Math. Random() *100)  
return msg;
```





Dashboard URL: 127.0.0.1:1880/ui/#/0?...

UtccFoood

Group 1

chart

22:09:00 22:14:00 22:22:00

gauge

0 10
units

Group 2

text 1

ON (1)

OFF (0)

numeric

5

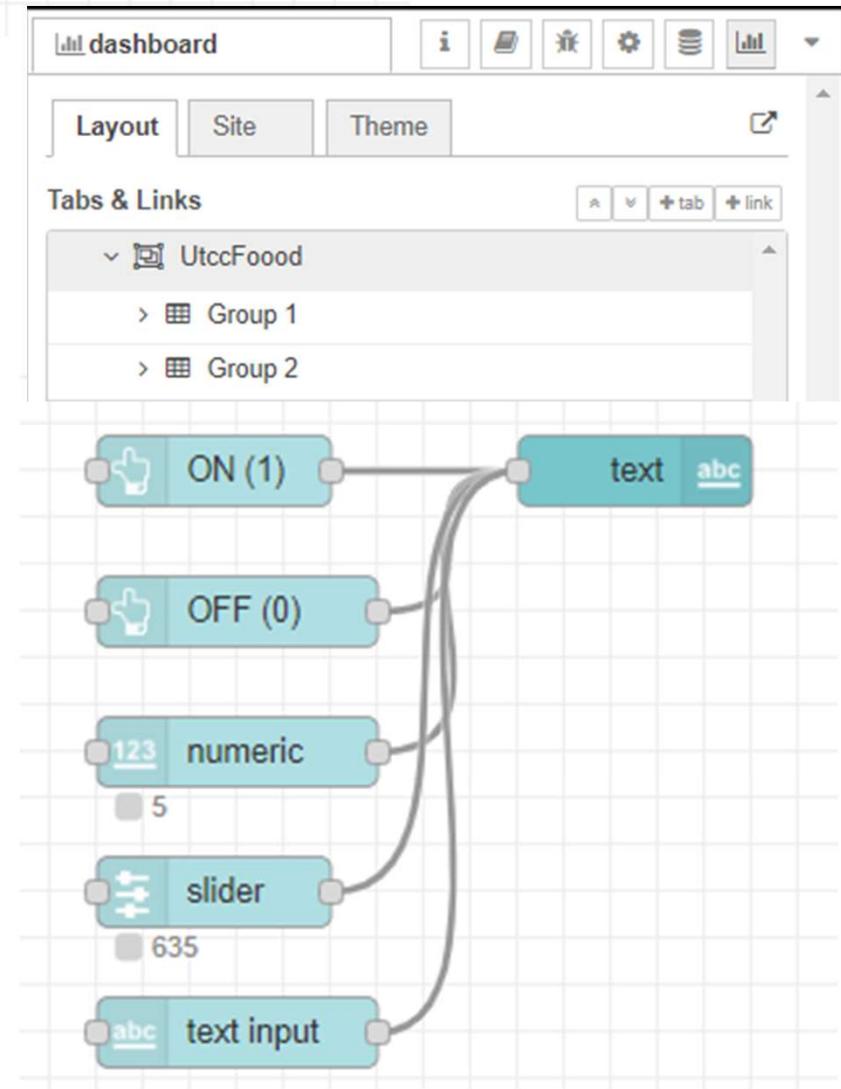
slider

45

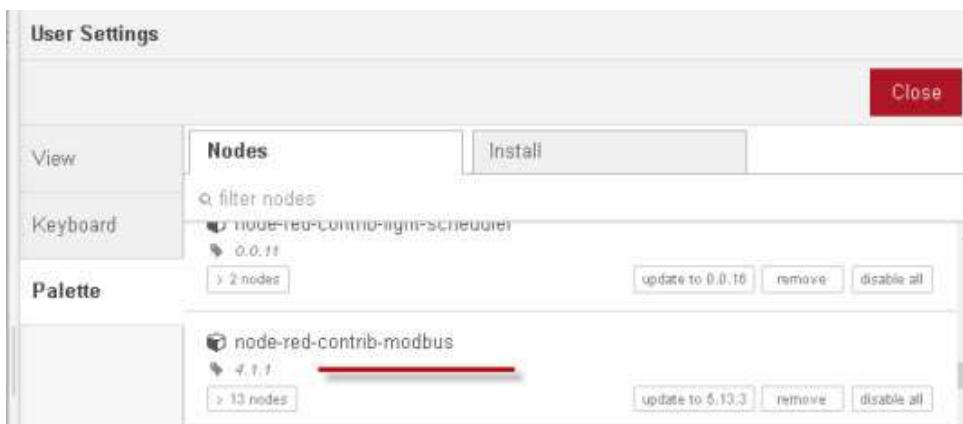
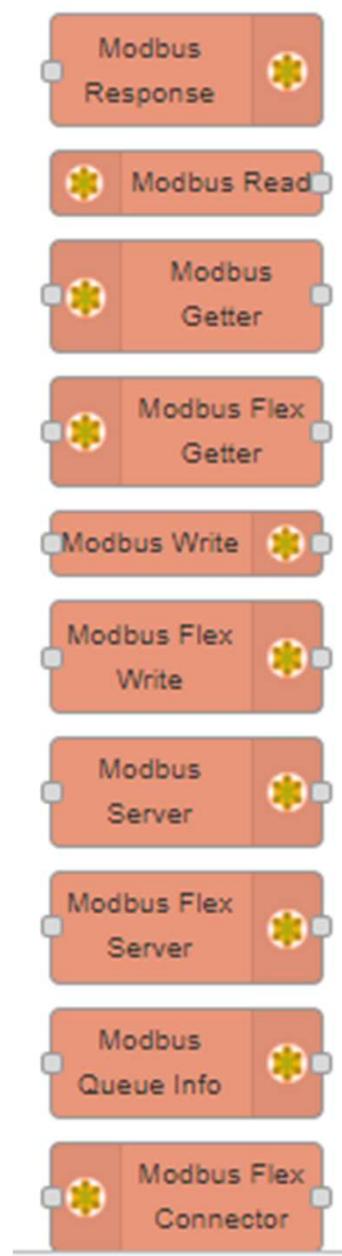
Layout **Site** **Theme**

Tabs & Links

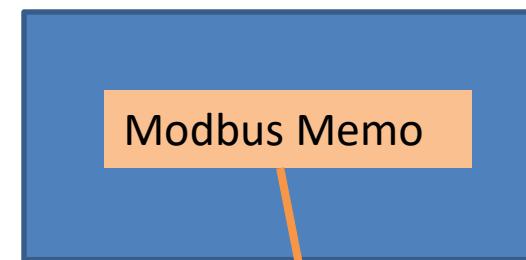
- UtccFoood
 - Group 1
 - Group 2



Modbus



Sensor , Machine, PLC,HMI

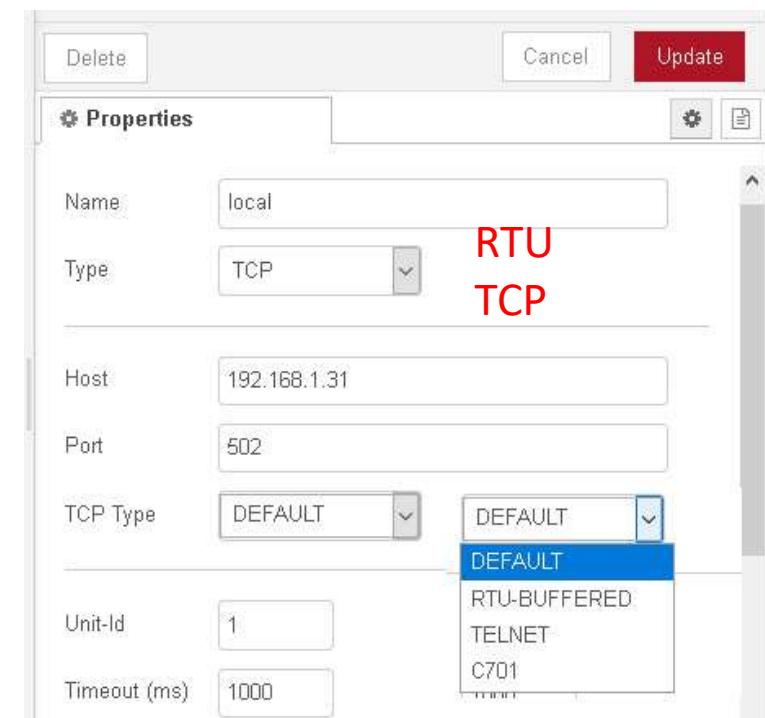


MODBUS Eth. TCP/IP PLC - Simulator (port: 502)

Connected (1/10) : (received/sent) (7681/7681) Serv. write de Rx: ● Tx: ●

Address: H D I/O Coil Outputs (000000) Fmt: decimal +/-

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+1
000001-000016	1	0	0	0	0	0	0	0	0	0	0
000017-000032	0	0	0	0	0	0	0	0	0	0	0
000033-000048	0	0	0	0	0	0	0	0	0	0	0
000049-000064	0	0	0	0	0	0	0	0	0	0	0
000065-000080	0	0	0	0	0	0	0	0	0	0	0
000081-000096	0	0	0	0	0	0	0	0	0	0	0
000097-000112	0	0	0	0	0	0	0	0	0	0	0
000113-000128	0	0	0	0	0	0	0	0	0	0	0
000129-000144	0	0	0	0	0	0	0	0	0	0	0



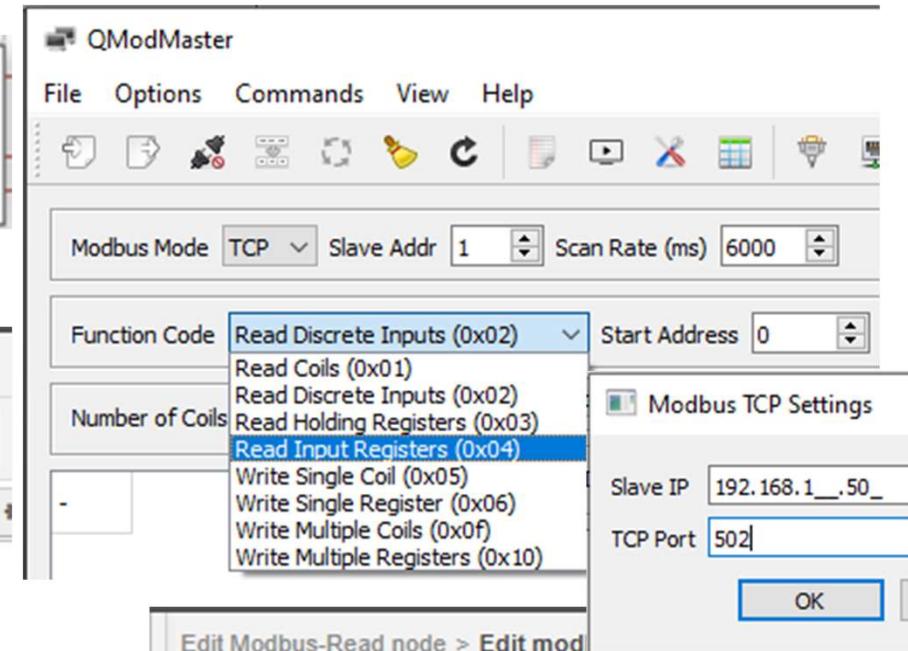


Edit Modbus-Read node

Properties

Settings Optionals

Name	<input type="text" value="Name"/>
Topic	<input type="text" value="Topic"/>
Unit-Id	1 (slave Addr) 1
FC	<input type="button" value="FC 1: Read Coil Status"/>
Address	<input type="text" value="0"/> Start Address ตำแหน่งข้อมูล
Quantity	<input type="text" value="1"/> Number of จำนวนข้อมูล
Poll Rate	<input type="text" value="1-65535"/> <input type="button" value=""/> กำหนดรอบเวลาในการอ่าน
Delay on start	<input type="checkbox"/>
Server	<input type="button" value="FactoryIO 192.168.1.50:502"/> <input type="button" value=""/>

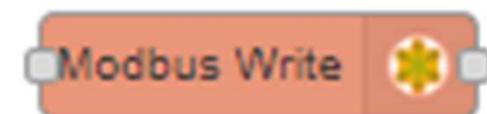


Edit Modbus-Read node > Edit mod

Properties

Name	<input type="text" value="Name"/>
Type	<input type="button" value="TCP"/>
Host	<input type="text" value="127.0.0.1"/>
Port	<input type="text" value="502"/>
TCP Type	<input type="button" value="DEFAULT"/>

กำหนดอุปกรณ์ที่ต้องการติดต่อ (RTU,TCP)



Edit Modbus-Write node

Delete Cancel Done

Properties

Name: Name

Unit-Id: 1

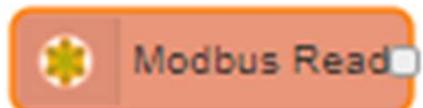
FC:

- FC 5: Force Single Coil
- FC 6: Preset Single Register
- FC 15: Force Multiple Coils
- FC 16: Preset Multiple Registers

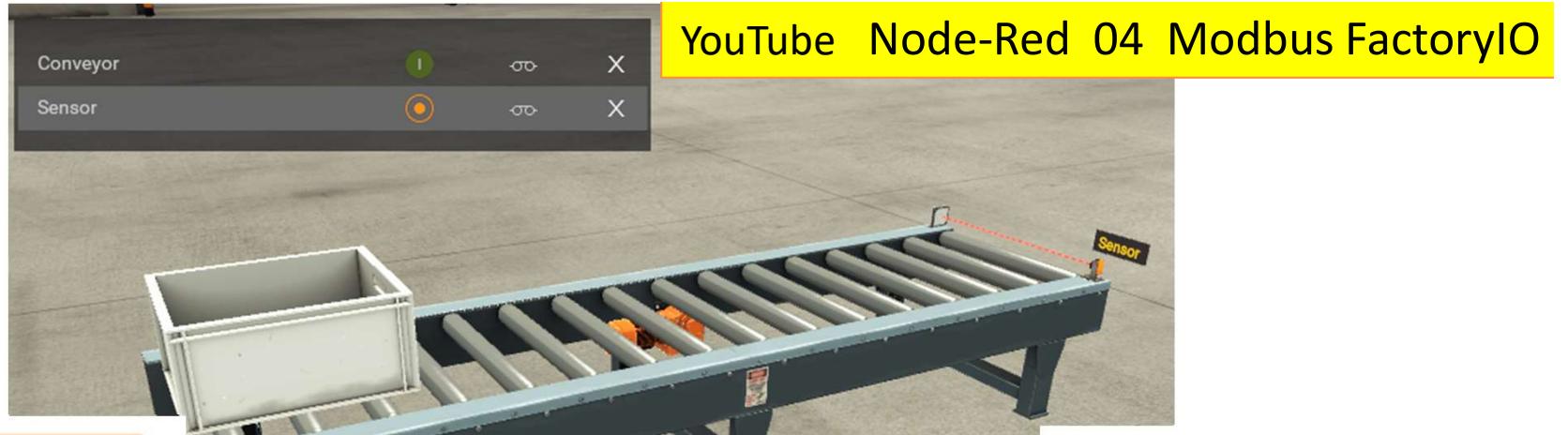
Address:

Quantity:

Server: modbus-tcp@127.0.0.1:502



- FC 1: Read Coil Status
- FC 2: Read Input Status
- FC 3: Read Holding Registers
- FC 4: Read Input Registers



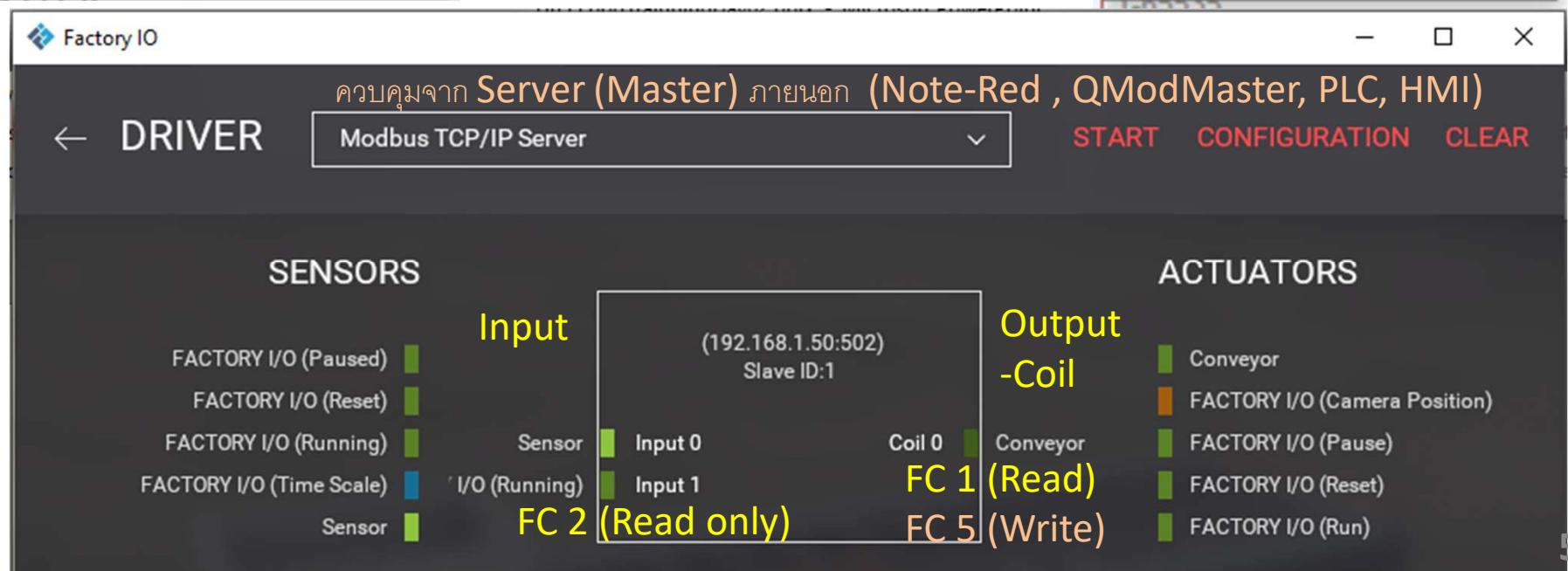
Modbus Read

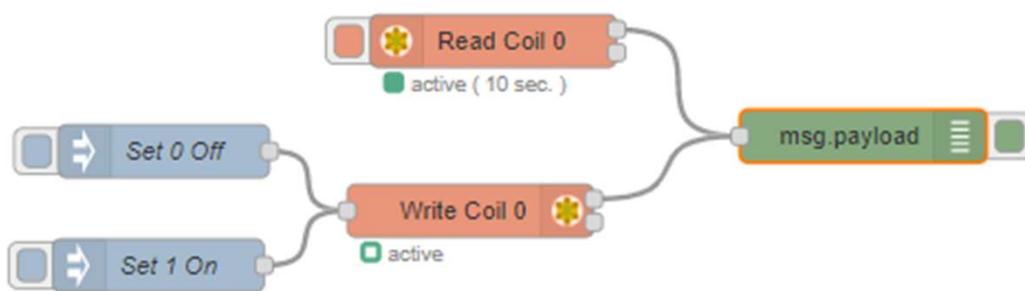
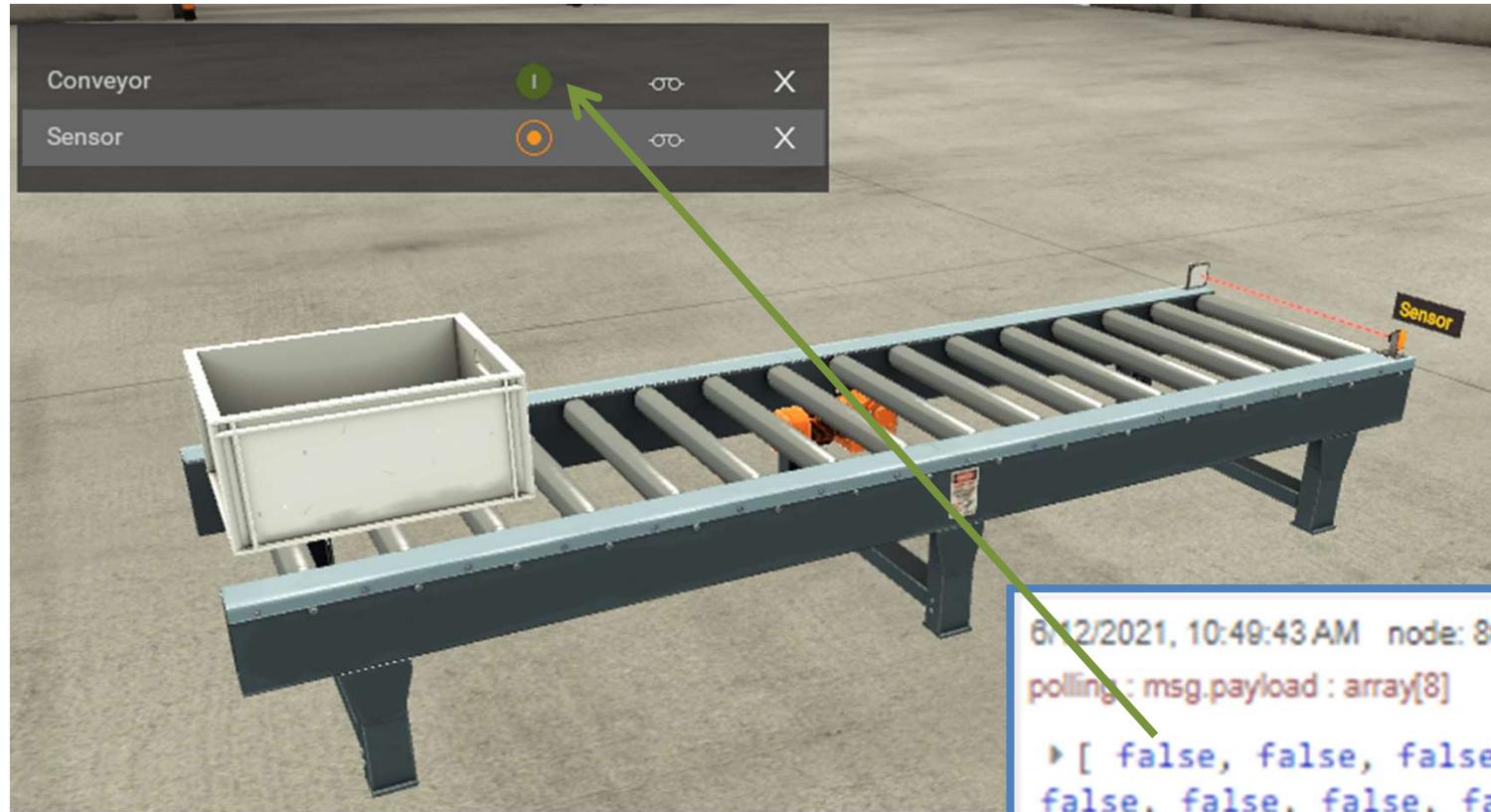
- FC 1: Read Coil Status
- FC 2: Read Input Status
- FC 3: Read Holding Registers
- FC 4: Read Input Registers

คำสั่ง FC 1 (Re)

Modbus Write

- FC 5: Force Single Coil
- FC 6: Preset Single Register
- FC 15: Force Multiple Coils
- FC 16: Preset Multiple Registers





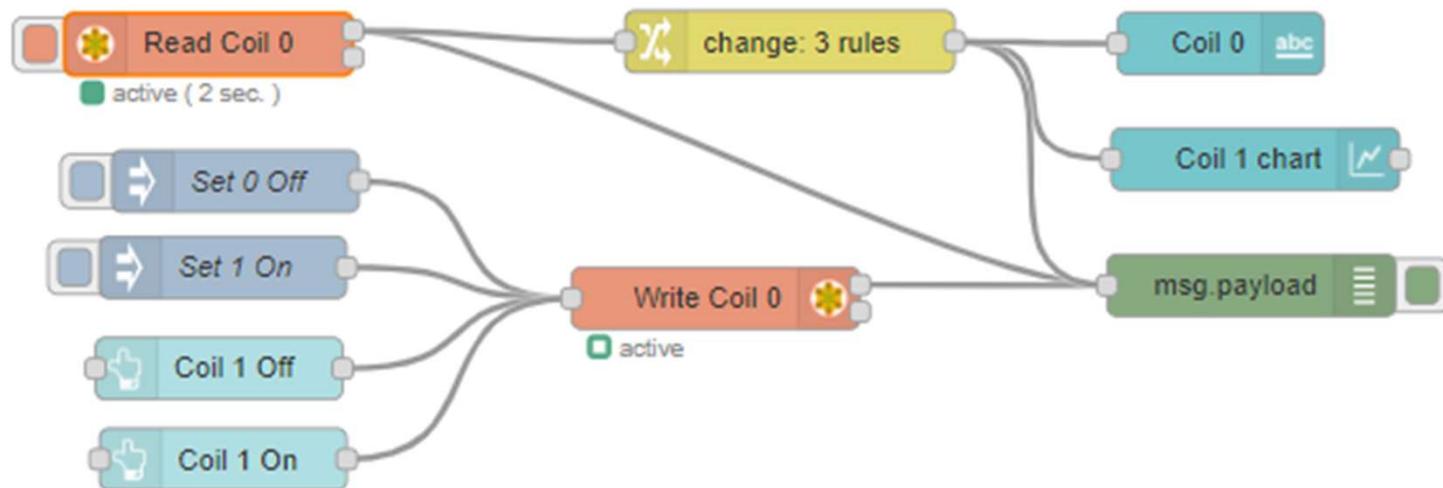
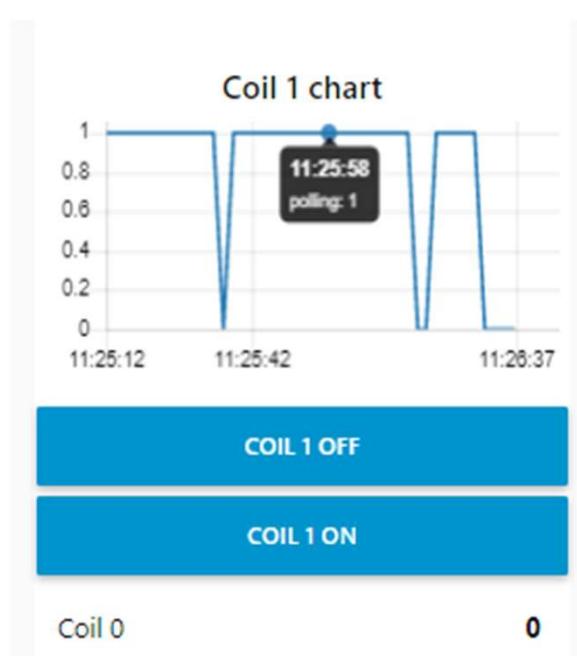
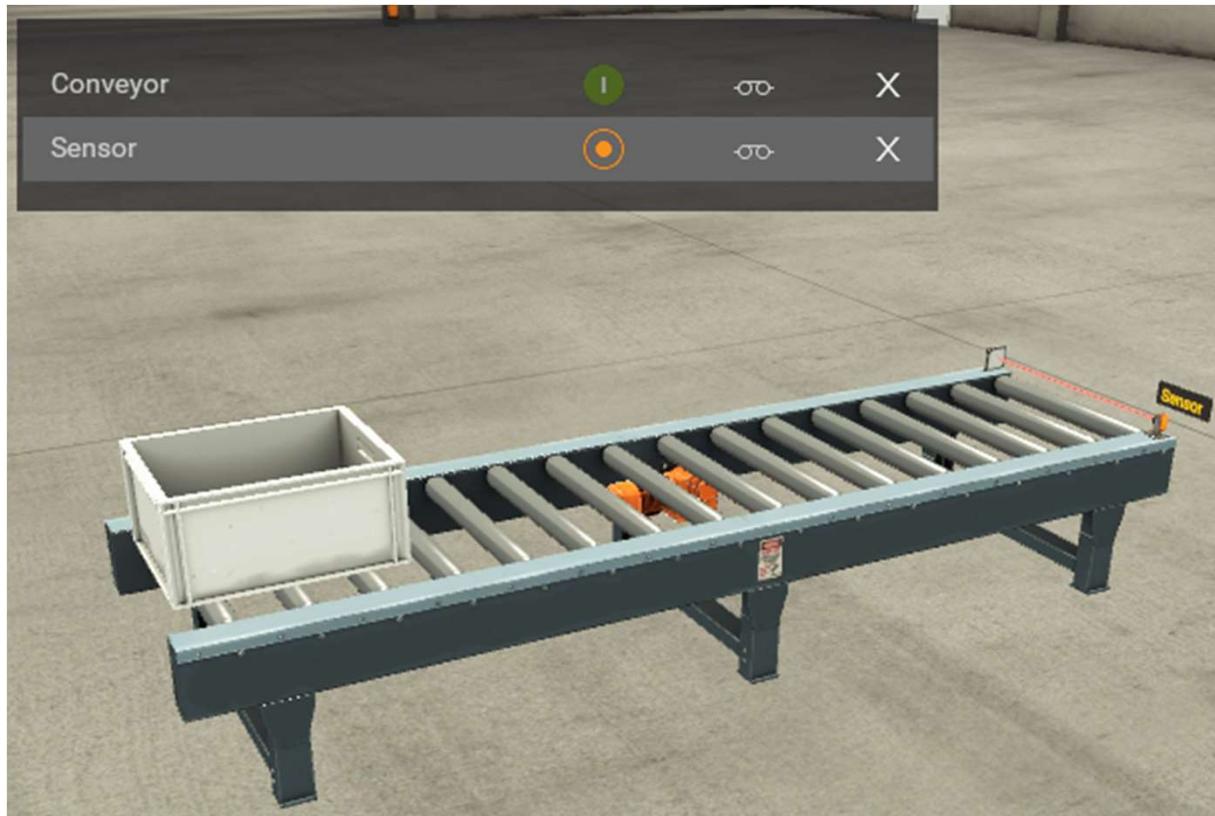
```

6/12/2021, 10:49:43 AM node: 8fad0f3b.89d87
polling : msg.payload : array[8]
▶ [ false, false, false, false,
  false, false, false, false ]

6/12/2021, 10:49:47 AM node: 8fad0f3b.89d87
fa64ffb4.b3eb3 : msg.payload : Object
▶ { value: true, unitid: "1", fc: 5,
  address: 0, quantity: "1" ... }

6/12/2021, 10:49:53 AM node: 8fad0f3b.89d87
polling : msg.payload : array[8]
▶ [ true, false, false, false, false,
  false, false, false ]

```

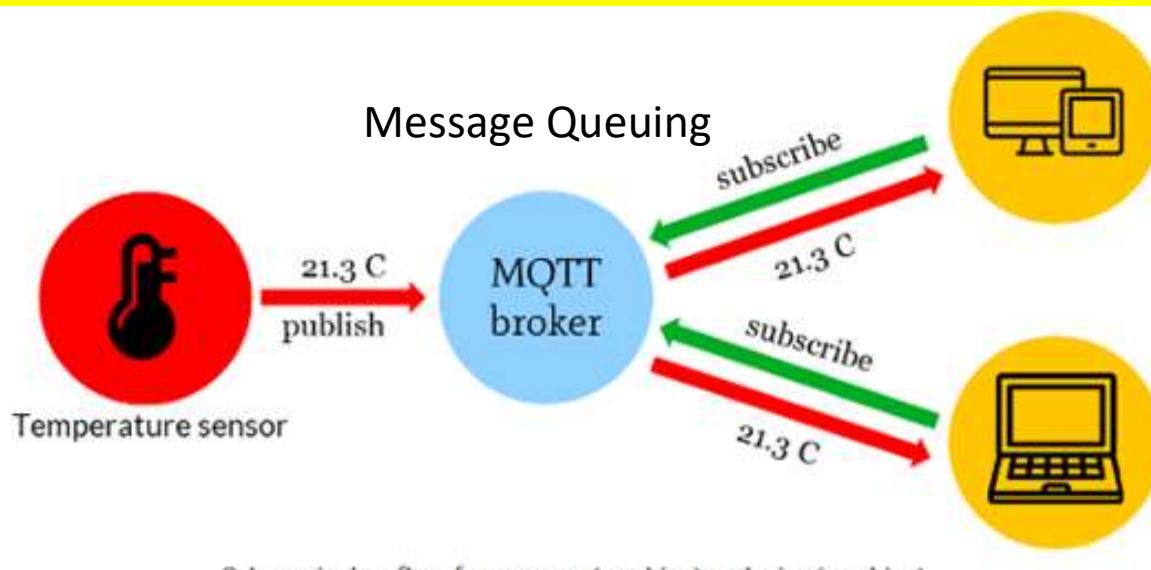


MQTT

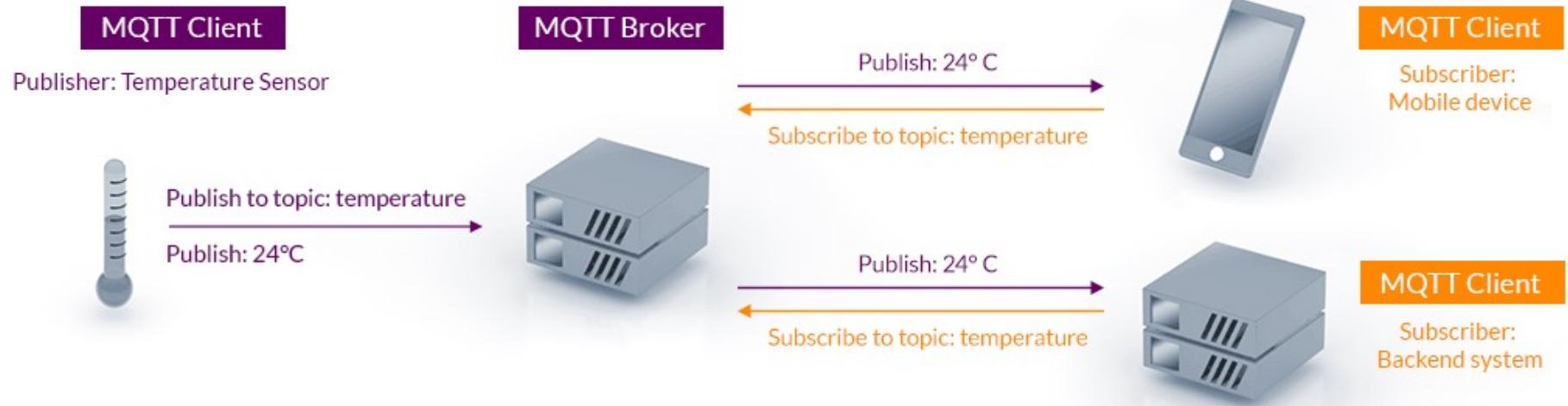
- Message Queuing Telemetry Transport (MQTT)
- เป็น Protocol ที่ออกแบบมาเพื่อการเชื่อมต่อแบบ M2M (machine-to-machine) คือ อุปกรณ์ติดต่อหรือสื่อสารกับ อุปกรณ์ โดยเป็นส่วนหนึ่งของเทคโนโลยี IoT (Internet of Things) ซึ่งเป็นเทคโนโลยีที่อินเทอร์เน็ตเชื่อมต่อกับอุปกรณ์ต่าง ๆ เช่น โทรศัพท์มือถือ รถยนต์ โทรทัศน์ ตู้เย็น เข้ากับอินเทอร์เน็ตทำให้ สามารถเชื่อมโยงสื่อสารกับอุปกรณ์ต่าง ๆ ได้ โดยผ่านเครือข่าย อินเทอร์เน็ต ซึ่งจะทำให้มนุษย์สามารถ ควบคุมอุปกรณ์ต่าง ๆ จากที่อื่น ได้ เช่นการสั่งปิดเปิดไฟในบ้านจากที่อื่น ๆ

MQTT Protocol

- การส่งข้อมูล (publish) ผู้ส่ง (อุปกรณ์ IOT, Software)
- ผู้บริการรับส่งข้อมูล (MQTT broker) ตัวกลาง Server , Cloud
- การขอรับข้อมูล (subscribe) ผู้รับ (อุปกรณ์ IOT, Software)



- สามารถรับ และส่ง ได้ใน อุปกรณ์เดียวกัน
- เป็นการสื่อสารสองทาง



YouTube MQTT 01 Basic

Connections

MQTT Connection mqtt://test.mosquitto.org:1883/

Name test.mosquitto.org

Validate certificate

Encryption (tls)

Protocol mqtt:// Host test.mosquitto.org Port 1883

Username Password

DELETE ADVANCED SAVE CONNECT

ผู้ให้บริการรับส่งต่อข้อมูล MQTT Server (MQTT broker)

Connections

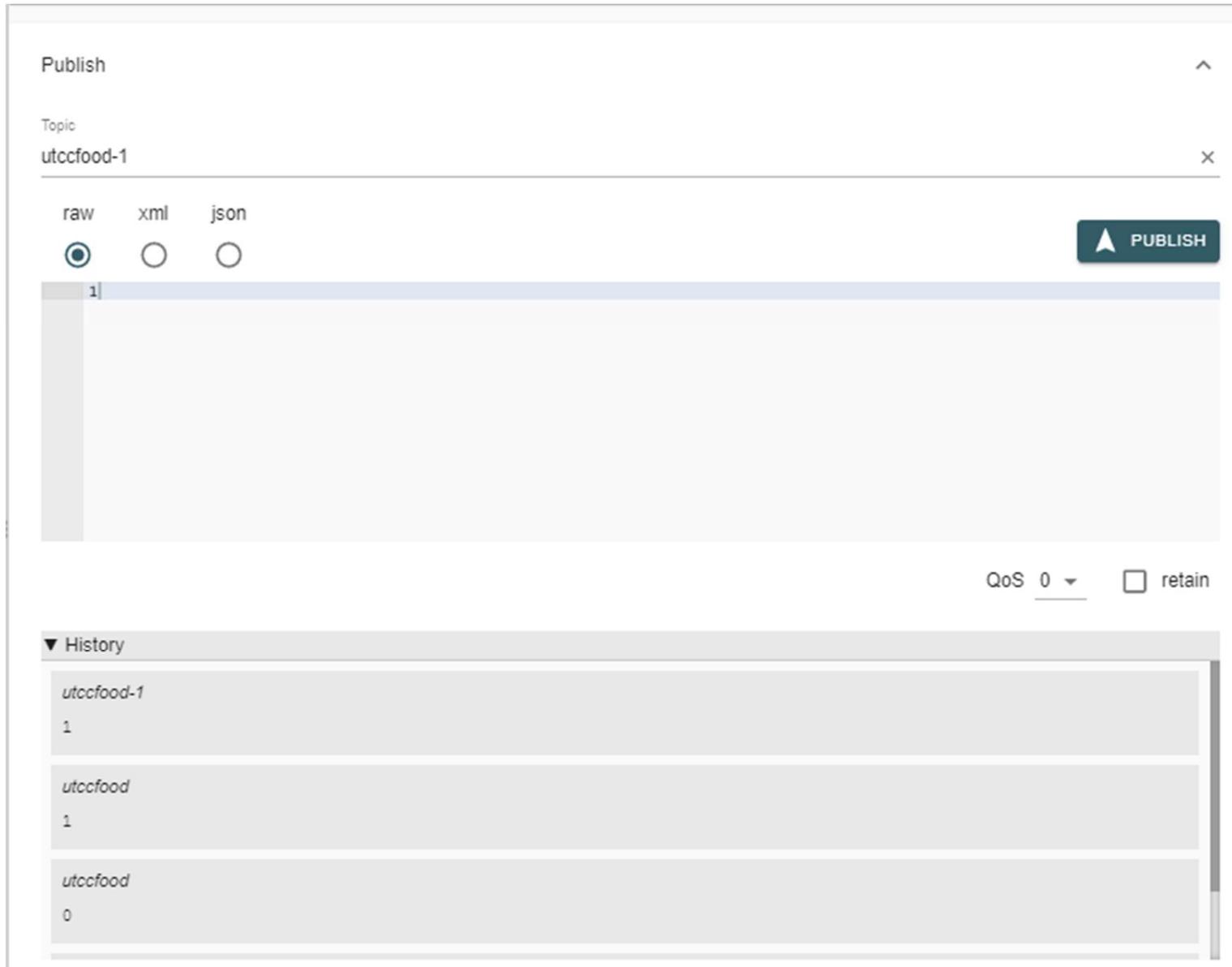
MQTT Connection mqtt://test.mosquitto.org:1883/

Scribe Topic QoS 0 + ADD

Topic	QoS
utccfood	0
utccfood-1	0

MQTT Client ID mqtt-explorer-0873aa4e CERTIFICATES BACK

ติดตามข้อมูลจาก Topic (subscribe)



YouTube MQTT 02 Node-Red

Edit mqtt out node

Delete Cancel Done

Properties

Server: test.mosquitto.org

Topic: utccfood

QoS: 0 Retain: 0

Name: Name

Edit mqtt in node

Delete Cancel Done

Properties

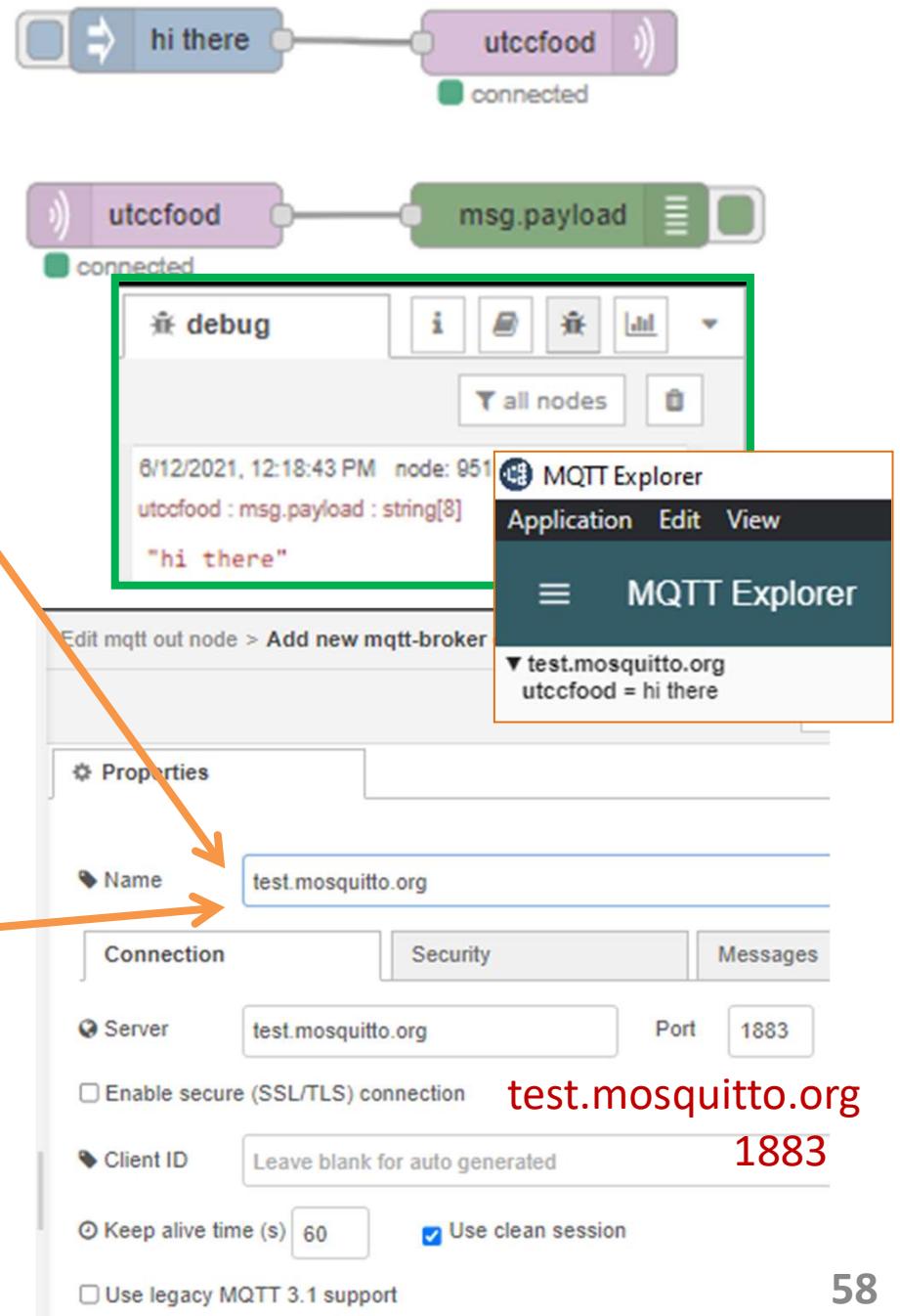
Server: test.mosquitto.org

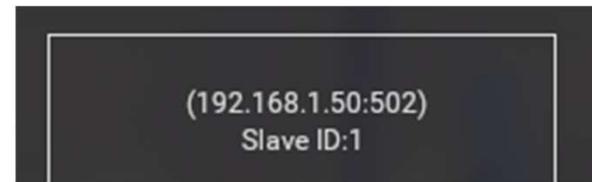
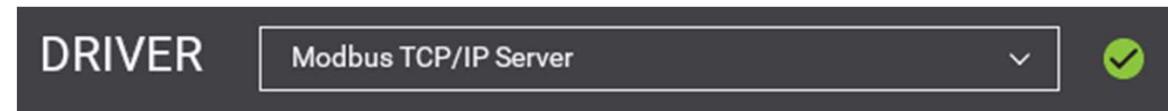
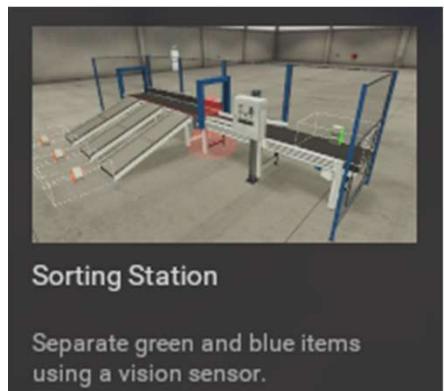
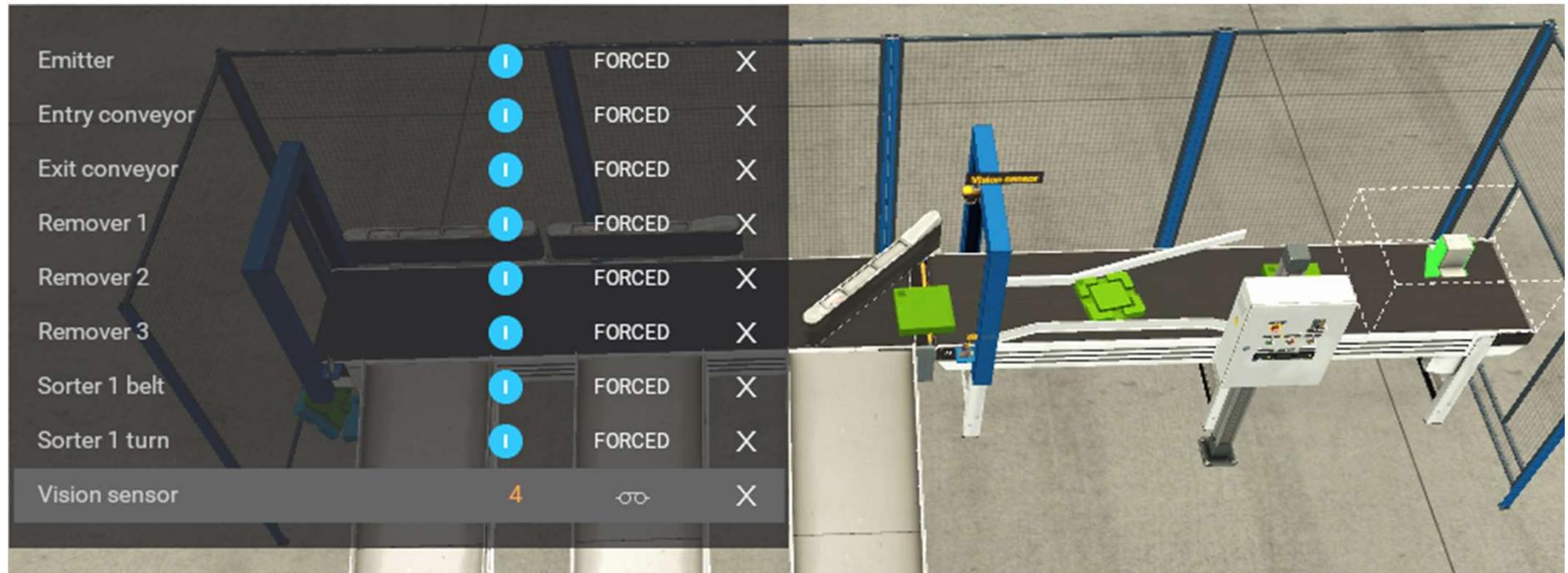
Topic: utccfood

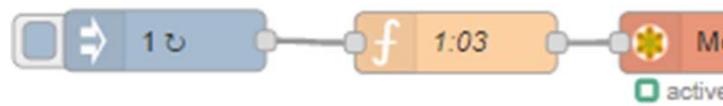
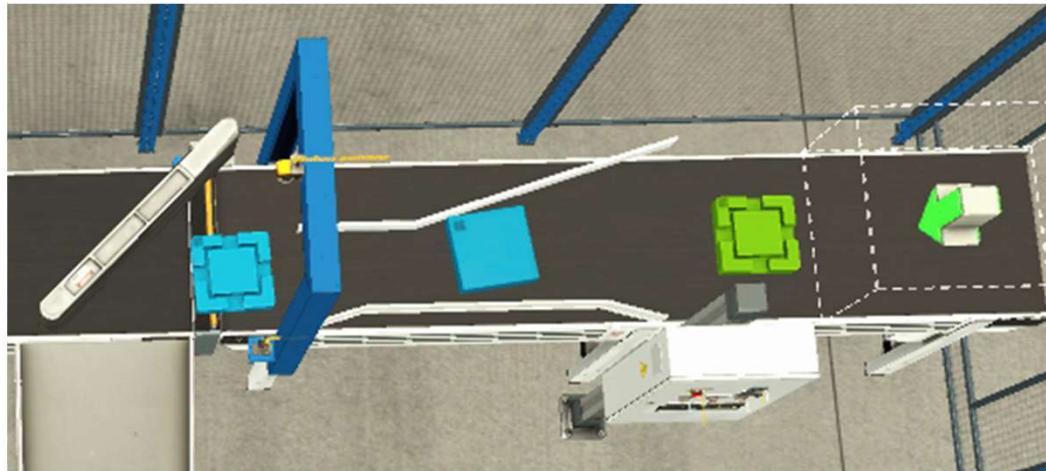
QoS: 2

Output: auto-detect (string or buffer)

Name: Name

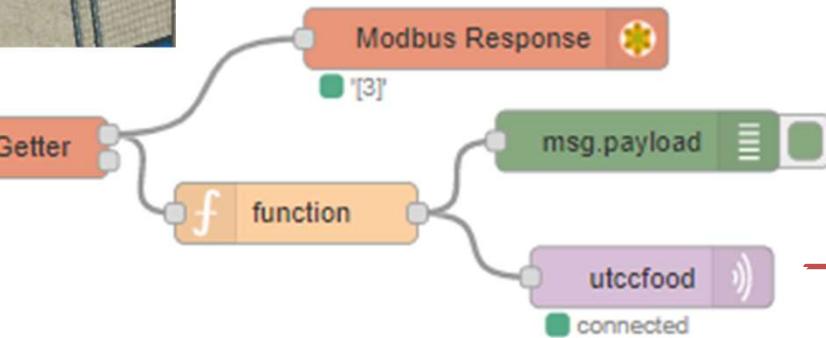
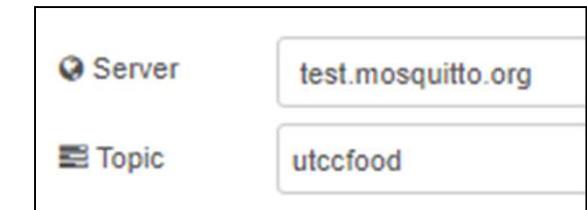






```
msg.payload = {
  'fc': 4,
  'unitid': 1,
  'address': 0 ,
  'quantity': 1 };

return msg;
```



```
msg.payload = msg.payload[0];

if (msg.payload == 1)
  { msg.payload ="Error 1" ; }
else if (msg.payload == 4)
  { msg.payload = "Error 4"; }
else { msg=null; }

return msg ;
```

Subscriptions

Topic: "utccfood" Showing the last 5 messages — +

Time Topic QoS
46 9:11:58 utccfood 0

Message: Error 1

Time Topic QoS
47 9:12:04 utccfood 0

Message: Error 4

Time Topic QoS
48 9:12:08 utccfood 0

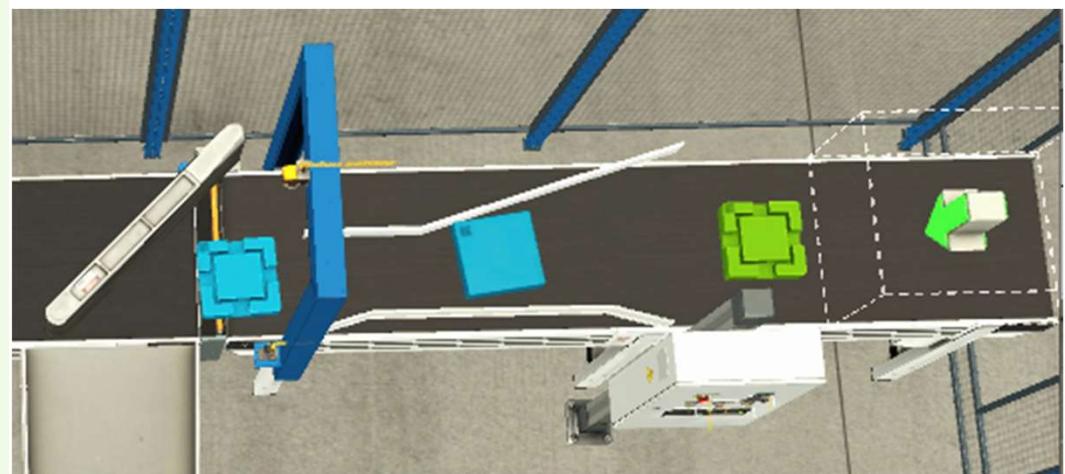
Message: Error 1

Time Topic QoS
49 9:12:14 utccfood 0

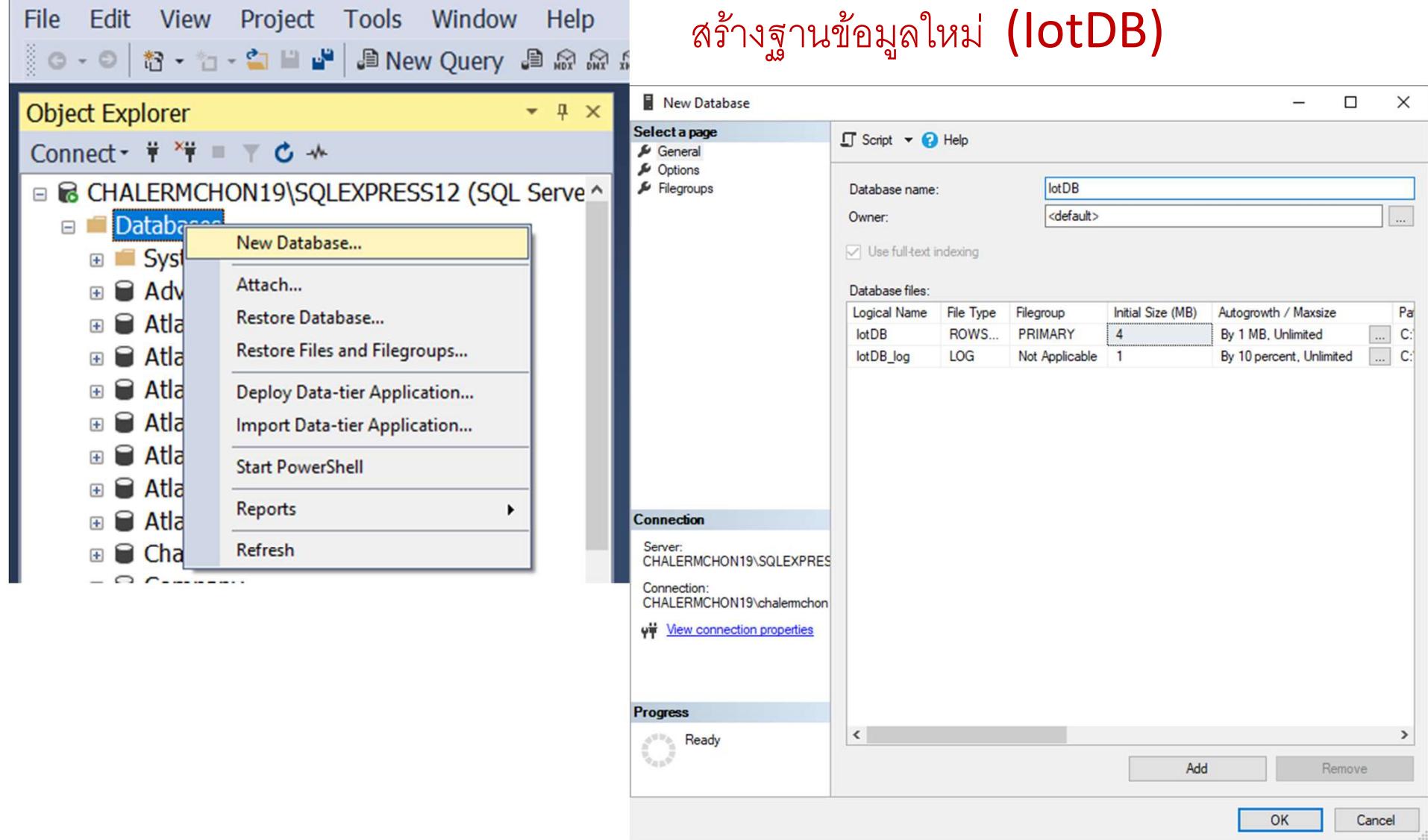
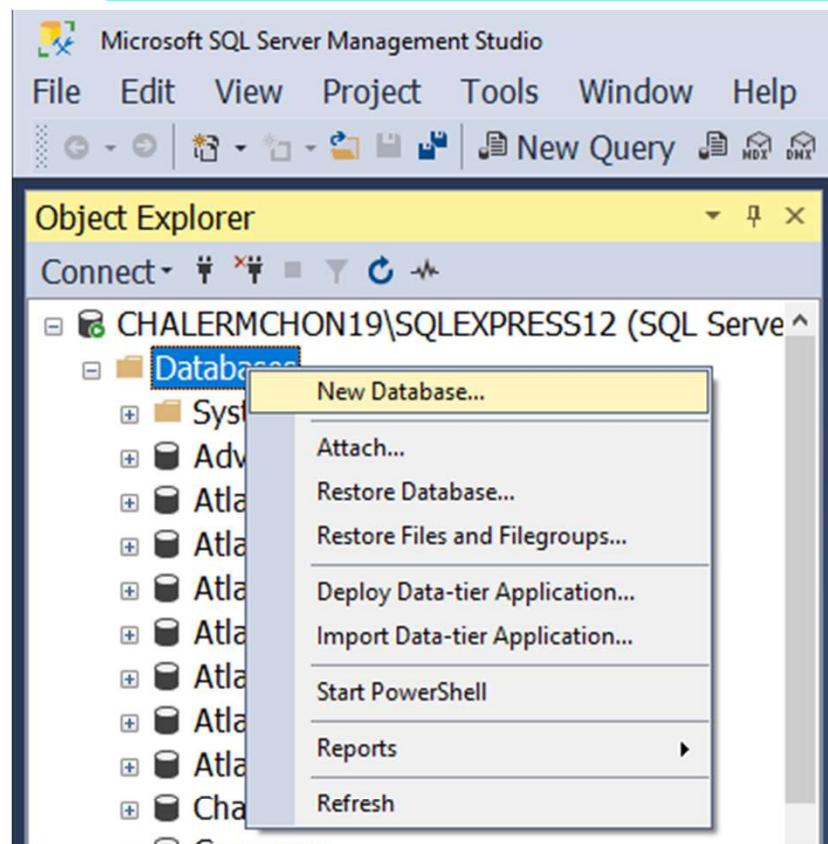
Message: Error 4

Time Topic QoS
50 9:12:16 utccfood 0

Message: Error 1



สร้างระบบฐานข้อมูลใหม่ (Mssql)



สร้างฐานข้อมูลใหม่ (iotDB)

สร้างตารางข้อมูลใหม่ ใน IoTDB

Object Explorer

Connect ▾

- IotDB
 - Database Diagrams
 - Table... (highlighted)
 - View
 - Sync
 - Programs
 - Server Objects
 - Storage
 - Security

New ▾

- Table...
- File Table...

CHALERMCHON...dbo.IotData* ▾

Column Name	Data Type	Allow Nulls	Choose Name
id	int	<input type="checkbox"/>	Choose Name
time	timestamp	<input checked="" type="checkbox"/>	
iotCode	nvarchar(50)	<input checked="" type="checkbox"/>	
SensorCode	nvarchar(50)	<input checked="" type="checkbox"/>	
SensorValue	float	<input checked="" type="checkbox"/>	
LocationCode	nvarchar(50)	<input checked="" type="checkbox"/>	
CreateDate	datetime	<input checked="" type="checkbox"/>	

Enter a name for the table:
IotData

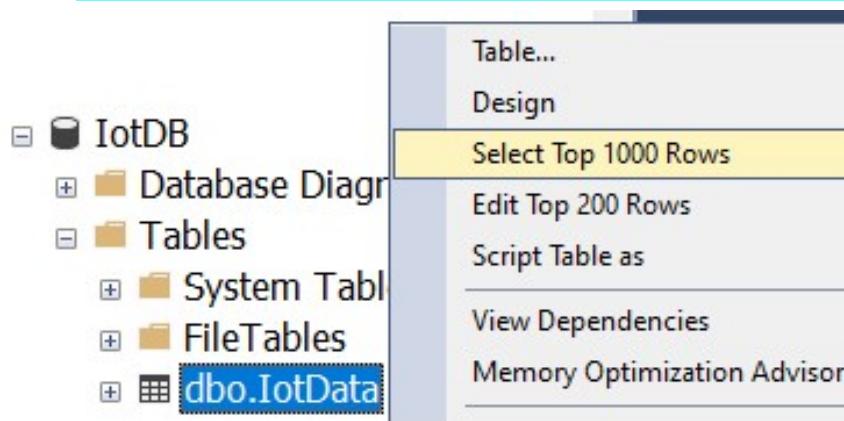
OK

Column Properties

(General)
(Name) CreateDate
Allow Nulls Yes
Data Type datetime
Default Value or Binding getdate()

Table Designer

Select Data (Sql)



The screenshot shows the SQL Query Editor window with the following content:

```
SQLQuery1.sql...rmchon (57)  ✎ ×
/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [id]
    ,[time]
    ,[IotCode]
    ,[SensorCode]
    ,[SensorValue]
    ,[LocationCode]
    ,[CreateDate]
FROM [IotDB].[dbo].[IotData]
```

The results pane shows the schema for the table:

	id	time	IotCode	SensorCode	SensorValue	LocationCode	CreateDate
--	----	------	---------	------------	-------------	--------------	------------

Insert New Data

CHALERMCHON...dbo.IotData

	id	time	iotCode	SensorCode	SensorValue	LocationCode	CreateDate
✓	NULL	NULL	iot01	t01	25.4	wh01	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

IotDB

- Database Diagrams
- Tables
 - System Tables
 - FileTables
 - dbo.IotData

Design

Select Top 1000 Rows

Edit Top 200 Rows

Script Table as

View Dependencies

Memory Optimization Advisor

```
SELECT TOP (1000) [id]
,[time]
,[iotCode]
,[SensorCode]
,[SensorValue]
,[LocationCode]
,[CreateDate]
FROM [IotDB].[dbo].[IotData]
```

Results Messages

	id	time	iotCode	SensorCode	SensorValue	LocationCode	CreateDate
1	NULL	0x0000000000000007D1	iot01	t01	25.4	wh01	2021-06-13 22:33:37.550

Select IoTData table by Node-Red

The screenshot displays a Node-Red flow and its configuration interface. The flow consists of three main components: a timestamp node, an MSSQL-PLUS node, and a msg.payload node. The timestamp node outputs to the MSSQL-PLUS node, which then outputs to the msg.payload node. A green 'done' button is located below the MSSQL-PLUS node.

Node-Red Flow:

```

graph LR
    timestamp[timestamp] --> MSSQLPLUS[MSSQL-PLUS]
    MSSQLPLUS --> msgpayload[msg.payload]
    
```

Edit MSSQL node Configuration:

- Properties:** Delete, Done, Properties (highlighted with a red circle).
- Connection:** MsSQL Server (dropdown), Edit icon (highlighted with a red circle).
- Name:** Name
- Query mode:** Query
- Query:** Editor


```

1 SELECT TOP (30) *
2 FROM [IotDB].[dbo].[IoTData]
3
      
```

msg.payload Data Structure:

```

6/13/2021, 10:42:34 PM node: bbb0e293.34a2c
msg.payload : array[1]
  ↘ array[1]
    ↘ 0: object
      id: null
      time: buffer[8]
      IoTCode: "iot01"
      SensorCode: "t01"
      SensorValue: 25.4
      LocationCode: "wh01"
      CreateDate: "2021-06-13T15:33:37.550Z"
      
```

Server Configuration (Edit MSSQL node > Edit MSSQL):

- Properties:** Delete, Cancel, Update, Properties.
- Name:** MsSQL Server
- Server:** localhost (highlighted in red)
- Port:** 1433
- Username:** iot (highlighted in red)
- Password:** @iot (highlighted in red)

SELECT TOP (30) *

FROM [IotDB].[dbo].[IoTData]

```
INSERT INTO [IotData] (IotCode, SensorCode, SensorValue ,LocationCode)
VALUES ('iot01', 't01',25.9,'wh01');
```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "SQLQuery6.sql - localhost.IotDB (CHALERMCHON19\chalermchon (54))* - Microsoft SQL Server Management Studio". The menu bar includes File, Edit, View, Query, Project, Tools, Window, and Help. The toolbar has various icons for database management. The Object Explorer on the left lists databases: DevelopmentDatabase, EmployeeDB, formula, GrafanaDB, and IotDB. The main window has a tab titled "SQLQuery6.sq...rmchon (54)*" containing the following SQL code:

```
INSERT INTO [IotData] (IotCode, SensorCode, SensorValue ,LocationCode)
VALUES ('iot01', 't01',25.9,'wh01');
```

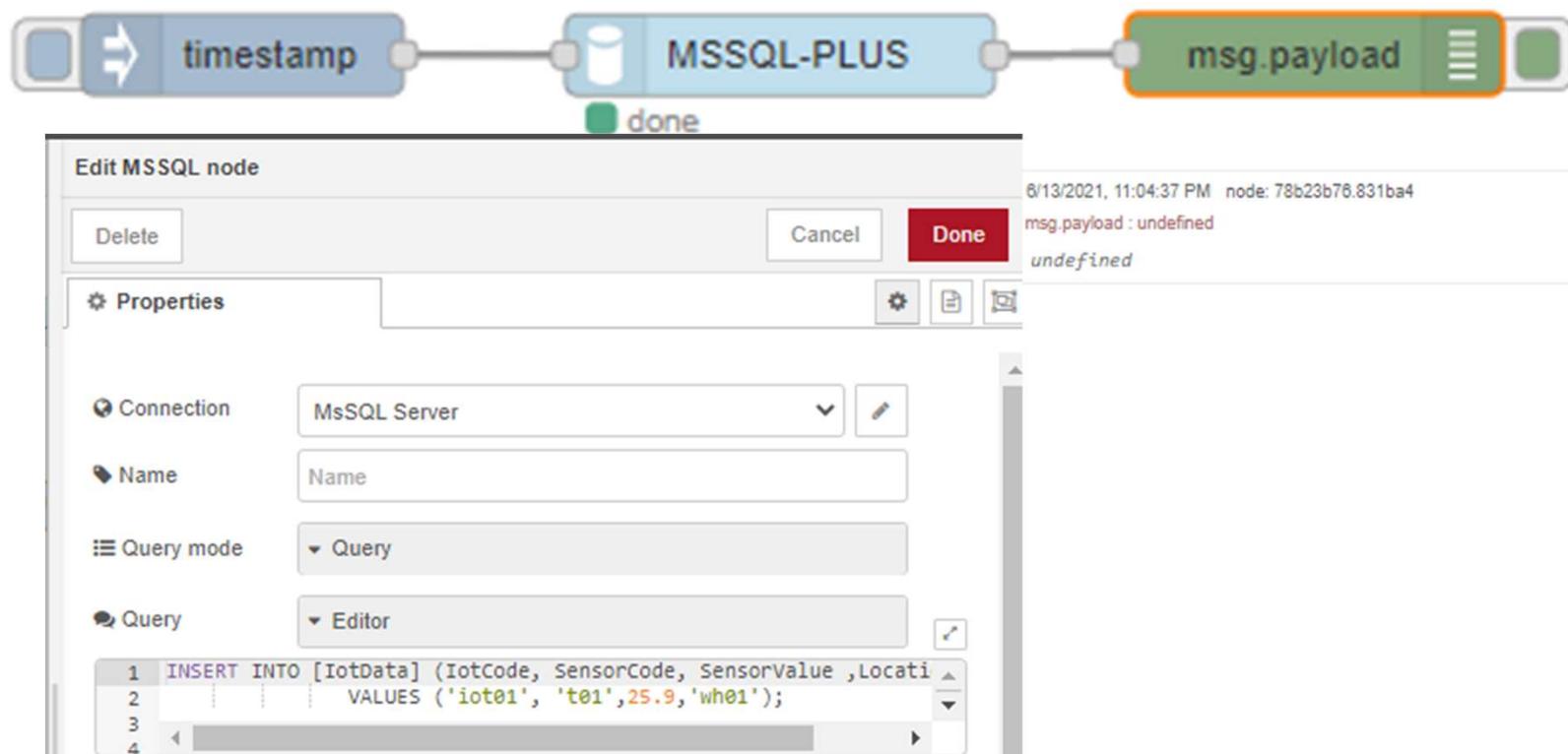
Below this, another tab shows a script for "SelectTopNRows" command from SSMS:

```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) *
FROM [IotDB].[dbo].[IotData]
```

The results pane at the bottom displays two rows of data from the IotData table:

	id	time	IotCode	SensorCode	SensorValue	LocationCode	CreateDate
1	1	0x00000000000000007D3	iot01	t01	25.4	wh01	2021-06-13 22:33:37.550
2	2	0x00000000000000007D5	iot01	t01	25.9	wh01	2021-06-13 22:59:49.507

Select IoTData table by Node-Red



```
INSERT INTO [IoTData] (IoTCode, SensorCode, SensorValue ,LocationCode)
VALUES ('iot01', 't01',25.9,'wh01');
```

Results

	id	time	IoTCode	SensorCode	SensorValue	LocationCode	CreateDate
1	1	0x00000000000000007D3	iot01	t01	25.4	wh01	2021-06-13 22:33:37.550
2	2	0x00000000000000007D5	iot01	t01	25.9	wh01	2021-06-13 22:59:49.507
3	3	0x00000000000000007D6	iot01	t01	25.9	wh01	2021-06-13 23:04:37.707

Insert IoTData by topic and payload

The screenshot shows a Node-RED flow and the configuration dialog for an 'MSSQL-PLUS' node.

Node-RED Flow:

```

graph LR
    t01:24.8 --> MSSQL[MSSQL-PLUS]
    MSSQL --> msg.payload
  
```

MSSQL-PLUS Node Configuration:

- Properties:**
 - Connection: MsSQL Server
 - Name: Name
 - Query mode: Query
 - Query (Editor):

```

1 INSERT INTO [IoTData] (IoTCode, SensorCode, SensorValue ,LocationCode)
2     VALUES ('iot01', '{{topic}}', {{payload}}, 'wh01');
3
        
```

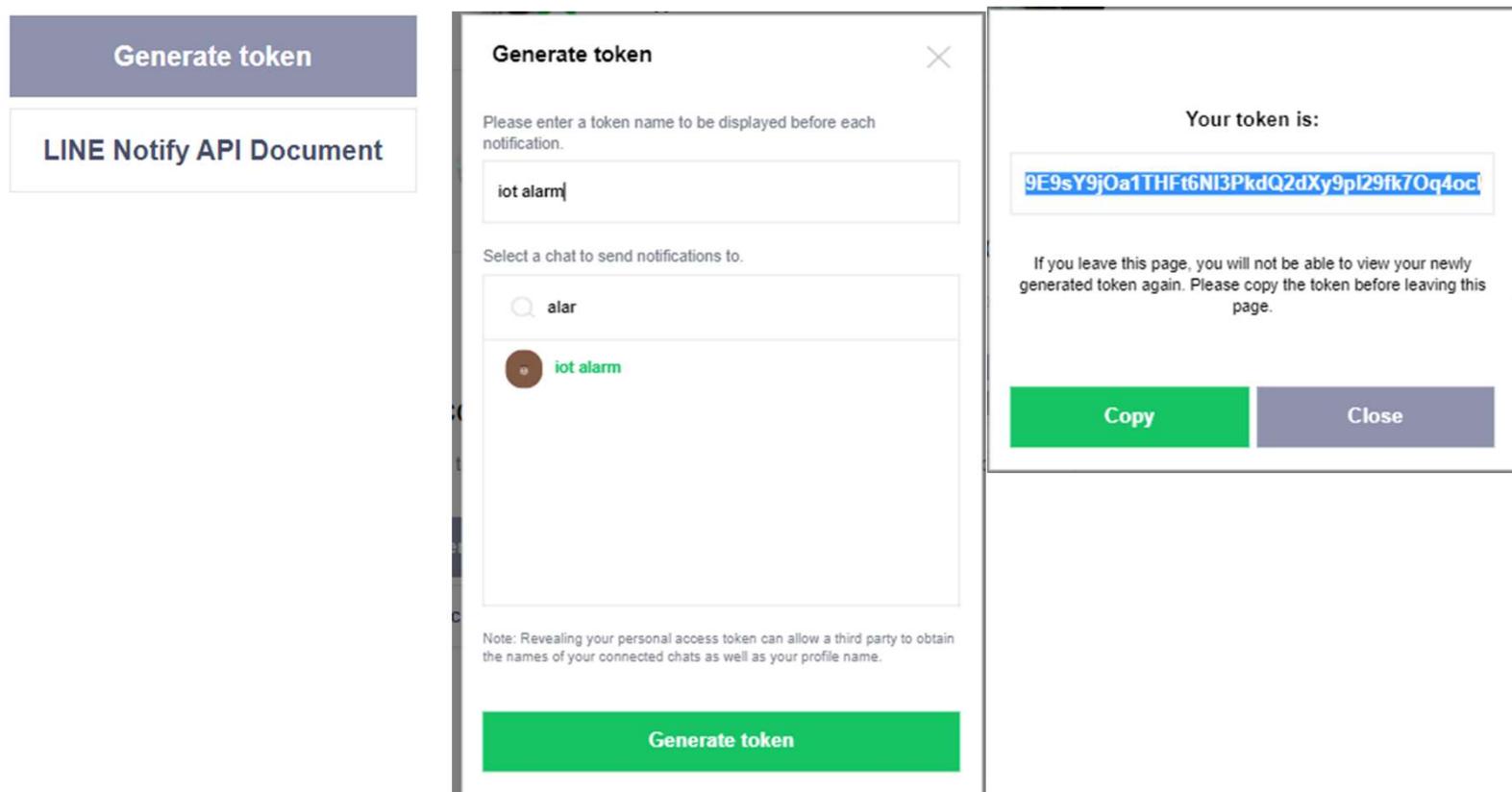
INSERT INTO [IoTData] (IoTCode, SensorCode, SensorValue ,LocationCode)
 VALUES ('iot01', '{{topic}}', {{payload}}, 'wh01');

	id	time	IoTCode	SensorCode	SensorValue	LocationCode	CreateDate
1	1	0x00000000000000007D3	iot01	t01	25.4	wh01	2021-06-13 22:33:37.550
2	2	0x00000000000000007D5	iot01	t01	25.9	wh01	2021-06-13 22:59:49.507
3	3	0x00000000000000007D6	iot01	t01	25.9	wh01	2021-06-13 23:04:37.707
4	4	0x00000000000000007D7	iot01	t01	24.8	wh01	2021-06-13 23:10:22.693

Line Notify

Generate access token (For developers)

By using personal access tokens, you can configure notifications without having to add a web service.



9E9sY9jOa1THFt6NI3PkQ2dXy9pl29fk7Oq4ocNxA



```

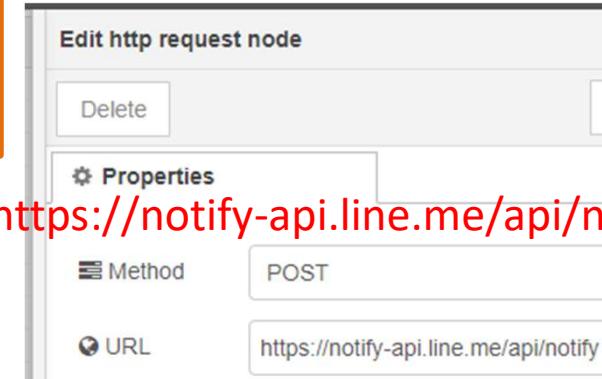
msg.headers = {'content-type':'application/x-www-form-urlencoded'
               , 'Authorization':'Bearer YourLineToken'};
msg.payload = {"message":"alarm " + msg.payload };
return msg;

```

```

msg.headers = {'content-type':'application/x-www-
form-urlencoded','Authorization':'Bearer
9E9sY9jOa1THFt6NI3PkQ2dXy9pl29fk7Oq4ocNxA'};
msg.payload = {"message":"alarm " + msg.payload };
return msg;

```



<https://notify-api.line.me/api/notify>

