

# INDUSTRIAL INTERNET OF THINGS (IIOT) AND SCADA

- จากหัวใจหลักของอุตสาหกรรม 4.0 คือ การสื่อสาร และการรวมข้อมูล ซึ่งในอุตสาหกรรมมีสองแนวทาง คือ
  - การประยุกต์อุปกรณ์ IoT เข้ามาใช้งานร่วมกับเครื่องจักรอุตสาหกรรมที่มีอยู่แล้ว เพื่อเพิ่มความสามารถให้เครื่องจักรสามารถสื่อสารและจัดส่งข้อมูลเข้าระบบฐานข้อมูลได้ เพื่อลดกระบวนการในการจดบันทึก
  - การเปลี่ยนอุปกรณ์และเครื่องจักรใหม่ ที่มีความสามารถในการสื่อสาร
  - การใช้งานโปรแกรมจัดการแบบสำเร็จรูป เช่น โปรแกรม SCADA ซึ่งสะดวก แต่มีค่าใช้จ่ายสูง
  - การใช้งานโปรแกรมแบบ Open Source ซึ่งไม่มีค่าใช้จ่าย แต่ต้องใช้บุคลากรที่ความรู้และเข้าใจ
- การรวมข้อมูลในกระบวนการผลิตในรูปแบบดิจิตอล จะส่งผลให้การวิเคราะห์ข้อมูลได้อย่างรวดเร็ว และความถูกต้อง ซึ่งจะช่วยทึ้งในด้านการควบคุมการผลิต การจัดการคลังสินค้า และด้านการตรวจสอบเครื่องจักร การบำรุงรักษา การลด downtime และต้นทุน และประสิทธิภาพในการทำงาน

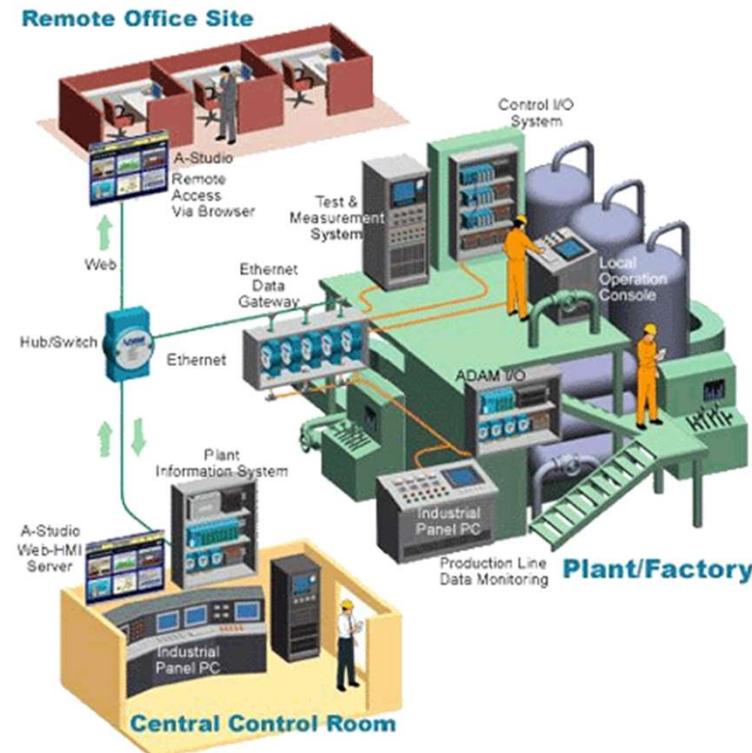
# Industrial Automation Control

**PLC** – Programmable Logic Controller

**HMI** – Human Machine Interface

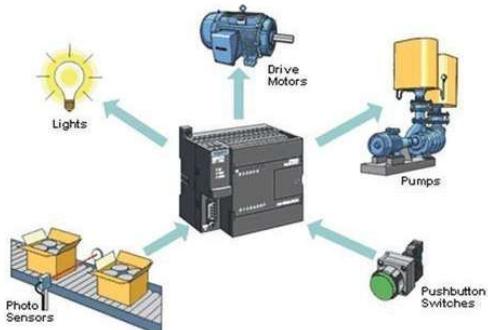
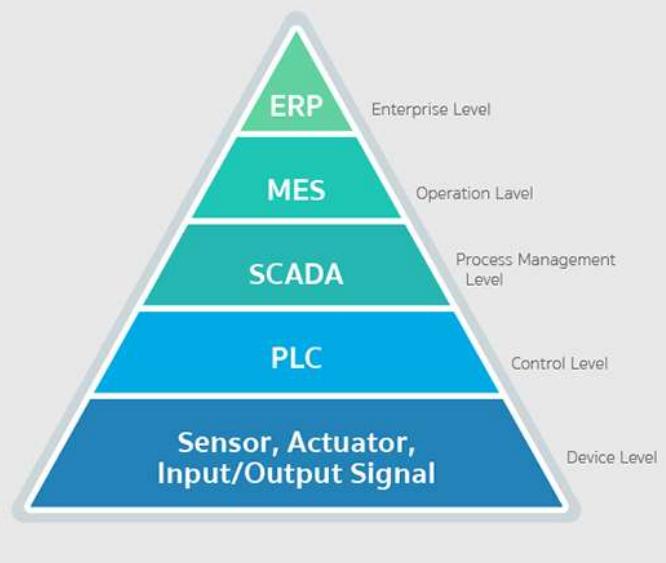
**SCADA** – Supervisory Control and Data Acquisition

**DCS** – Distributed Control System

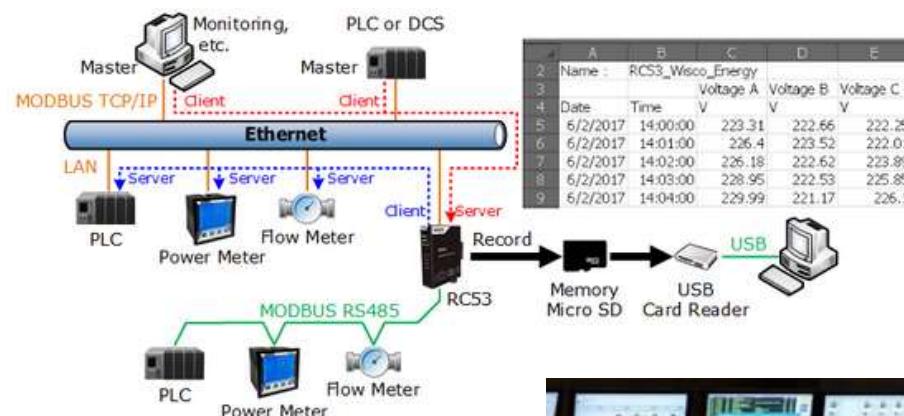


<https://electricaltrends.com/2017/03/21/10-trends-industrial-automation-and-control-market/>

## Automation Pyramid (Automation Hierarchy)

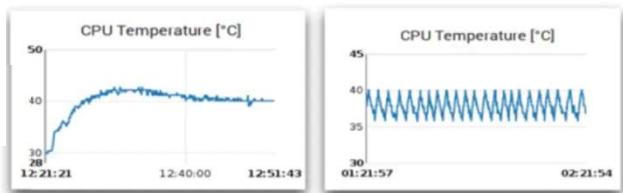


Protocol	Level	Common Applications
ModBus	Device	Manufacturing, Electric Utility
Profibus	Device	Process Industry
DeviceNet	Device	Manufacturing
DNP 3.0	Device	Electric Utility SCADA
BACNet	Control	HVAC Control, Building Automation
ControlNet	Control	Manufacturing
ARCNet	Supervisory	Office Automation, Gaming
Ethernet/IP	Supervisory	Office Automation, Internet





### PID control VS On/Off control



SV

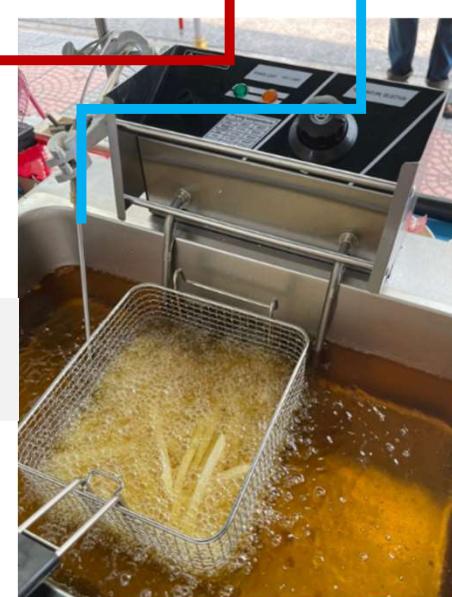
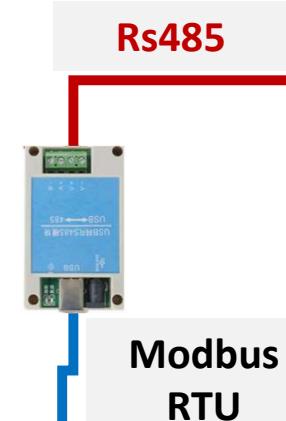
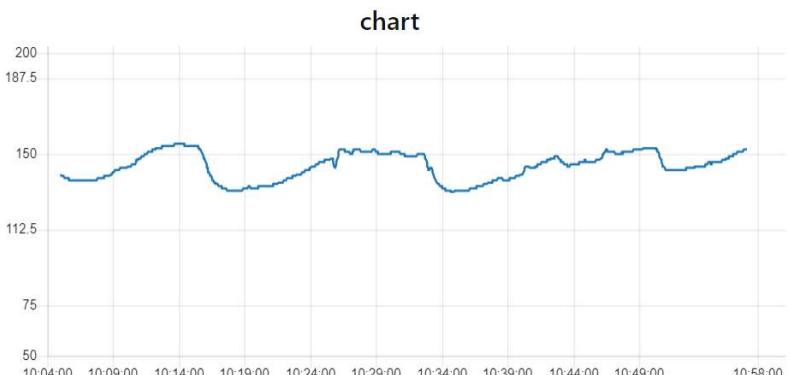
ปิด เปิด  
OnOff

Set SV Control 150

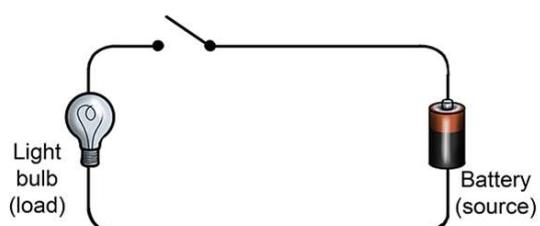
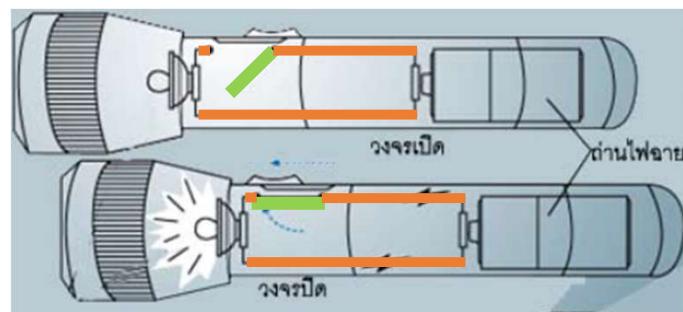
Frying Co... 4

Set Lower 145

Set Upper 150

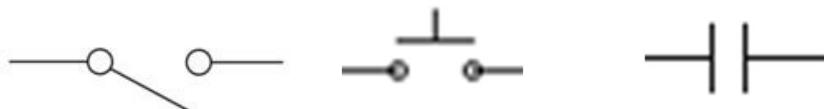


# การควบคุมสวิตซ์ Switch Control



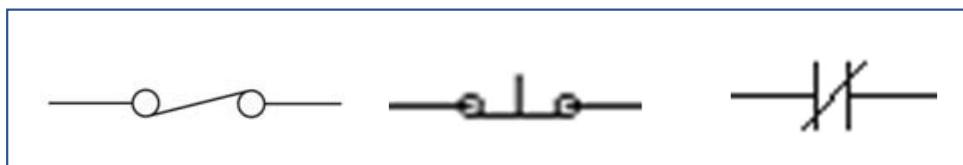
## วงจรเปิด (Open Circuit)

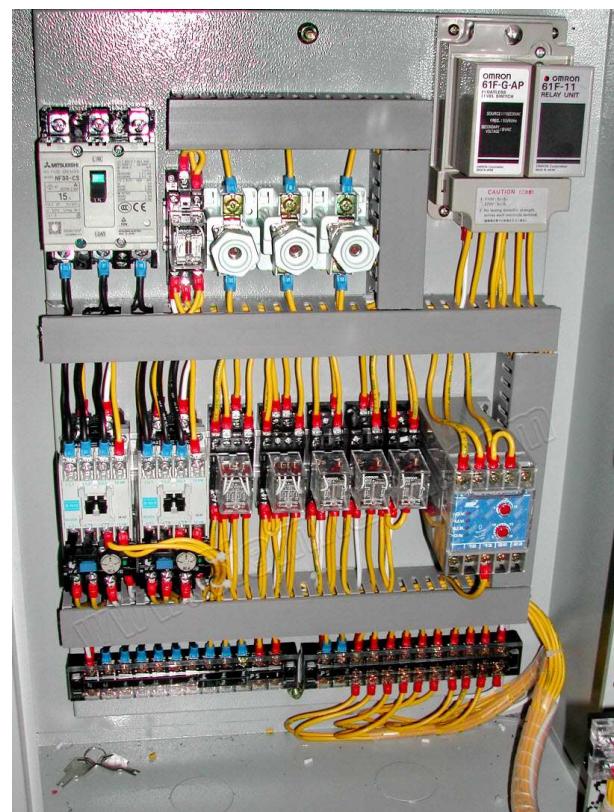
- กระแสไฟฟ้าไม่สามารถไหลผ่านได้ ทำให้อุปกรณ์ไม่ทำงาน
- หน้าตัวต่อไม่เชื่อมต่อกัน (Open Contact)



## วงจรปิด (Close Circuit)

- หน้าตัวต่อเชื่อมต่อกัน (Close Contact)
- กระแสไฟฟ้าไหลในวงจรได้ ทำให้อุปกรณ์ทำงาน





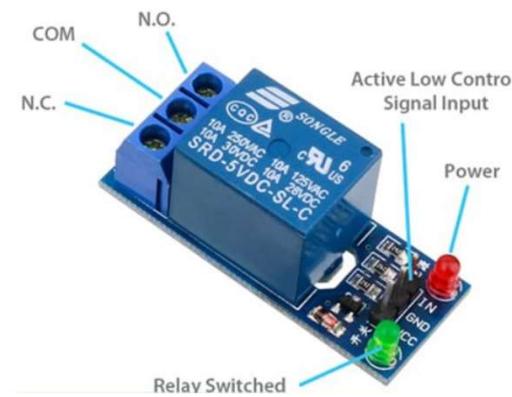
Normally Open Contact



Normally Closed Contact



Changeover Contact

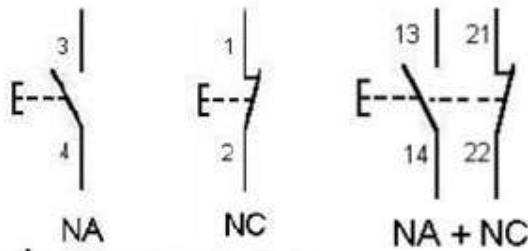




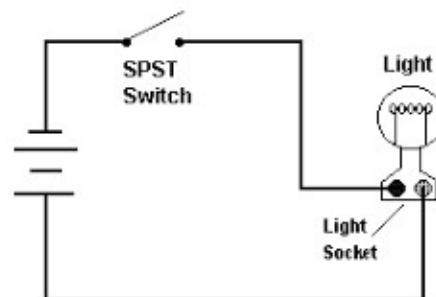
หมุนไปทาง MAN สั่งเปิดปิด  
จากสวิตช์หน้าคู่



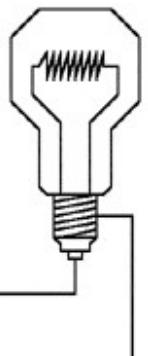
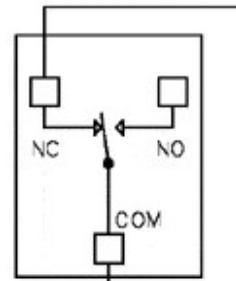
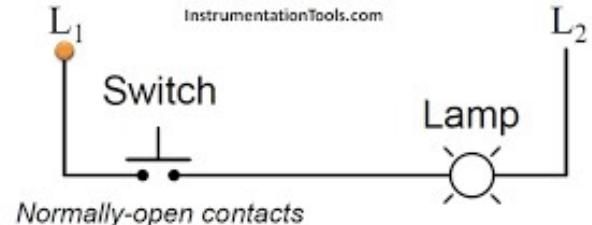
หมุนไปทาง AUTO รอเวลาที่ตั้งไว้  
จาก timer



Light Switch Circuit



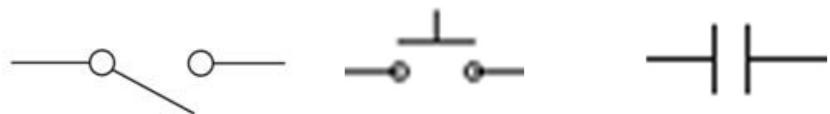
InstrumentationTools.com



Light Relay Turns Off when  
Turns On

## วงจรเปิด (Open Circuit)

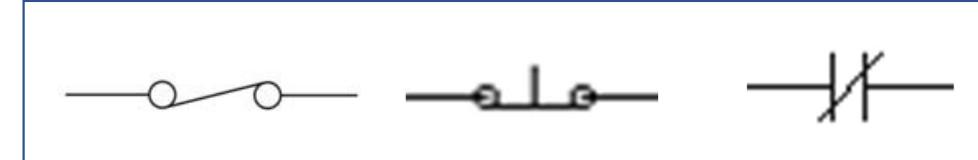
- กระแสไฟฟ้าไม่สามารถไหลผ่านได้ ทำให้อุปกรณ์ไม่ทำงาน
- หน้าสัมผัสไม่เชื่อมต่อกัน (Open Contact)



A Normal Open contact (NO)

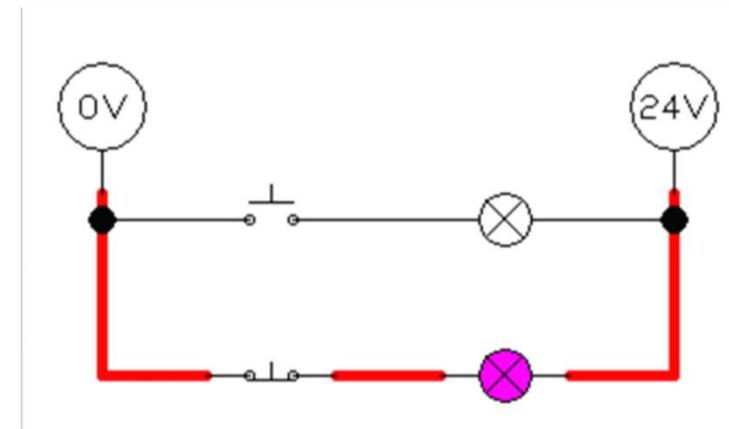
## วงจรปิด (Close Circuit)

- หน้าสัมผัสเชื่อมต่อกัน (Close Contact)
- กระแสไฟฟ้าไหลในวงจรได้ ทำให้อุปกรณ์ทำงาน

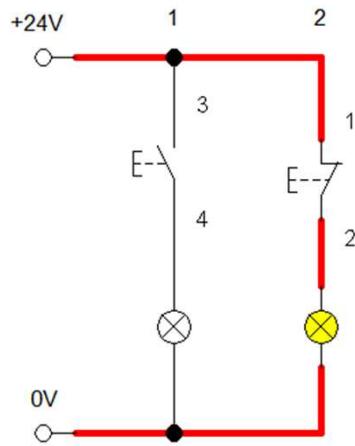
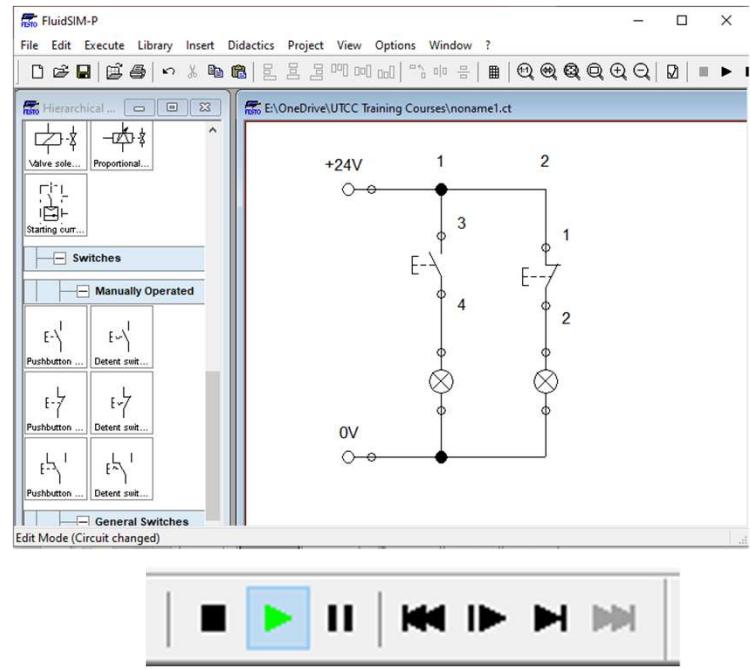
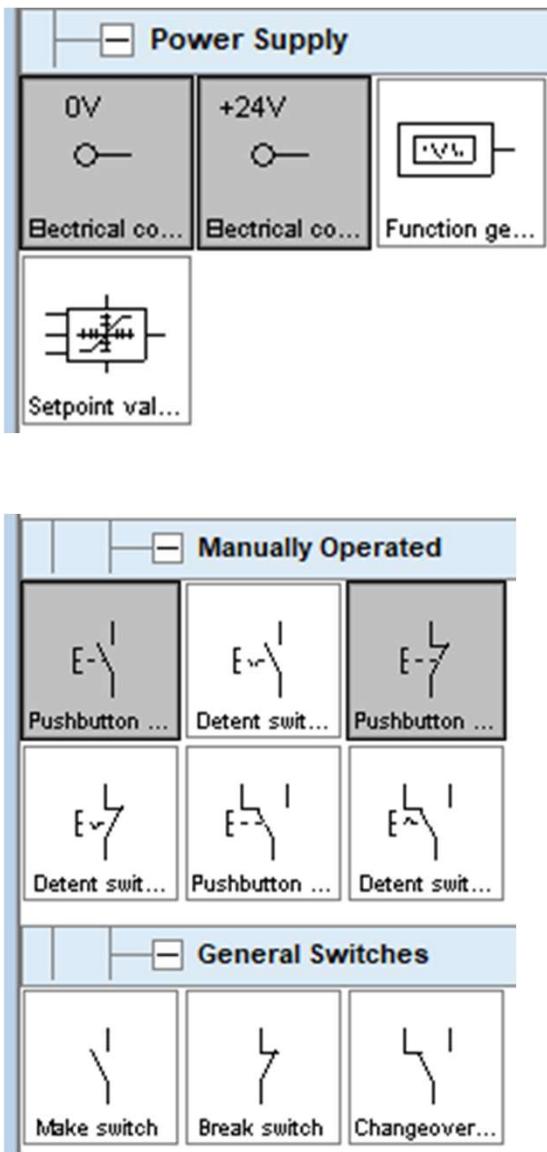
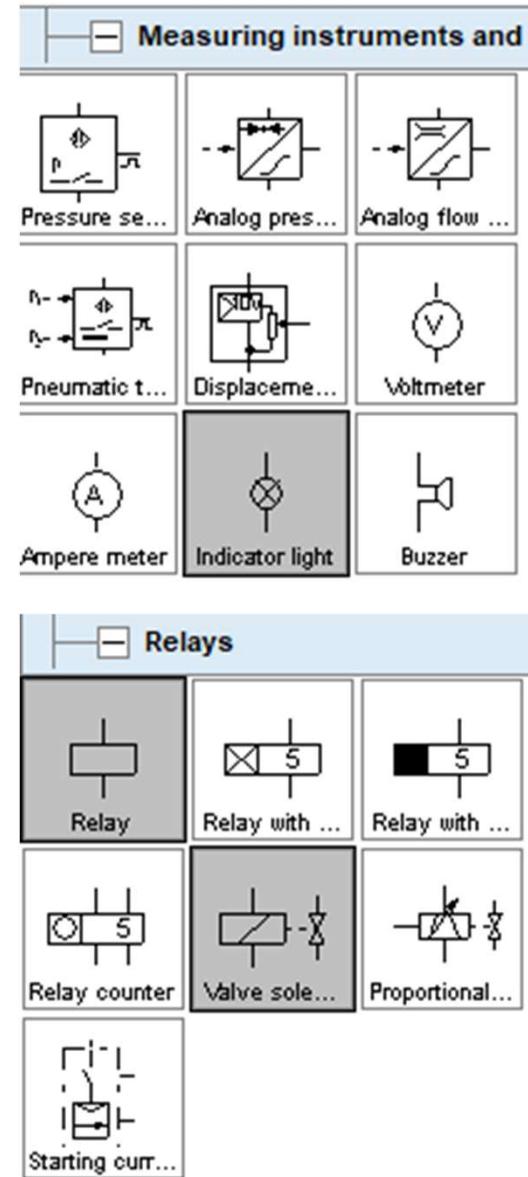


A Normal Closed contact (NC)

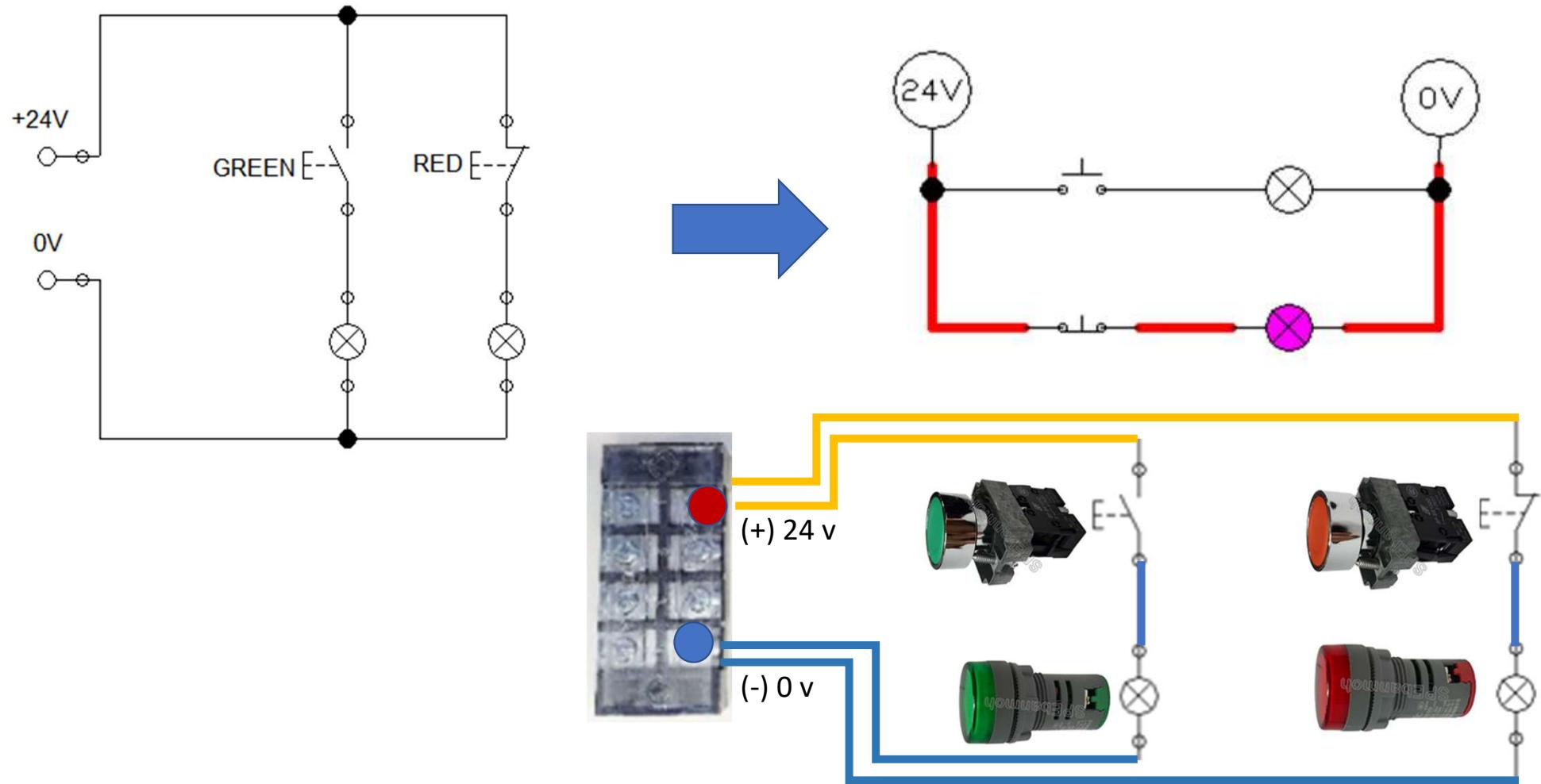
### ปกติหน้าสัมผัสเปิด



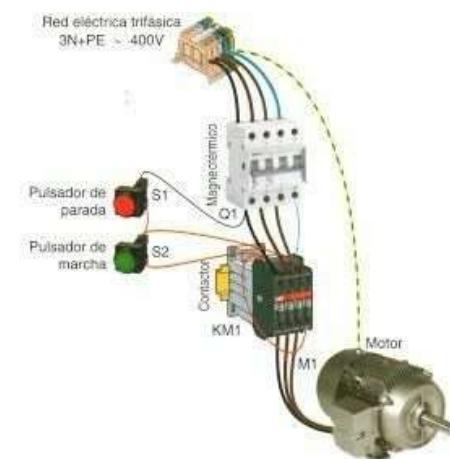
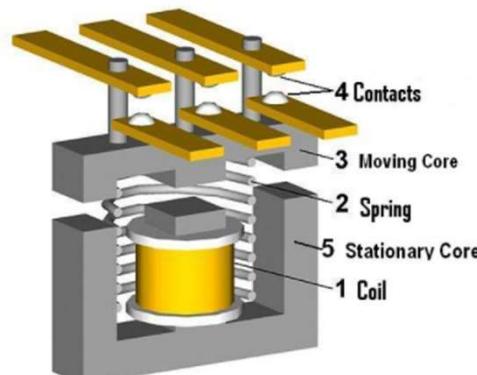
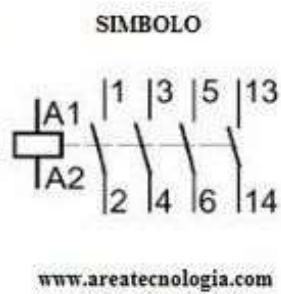
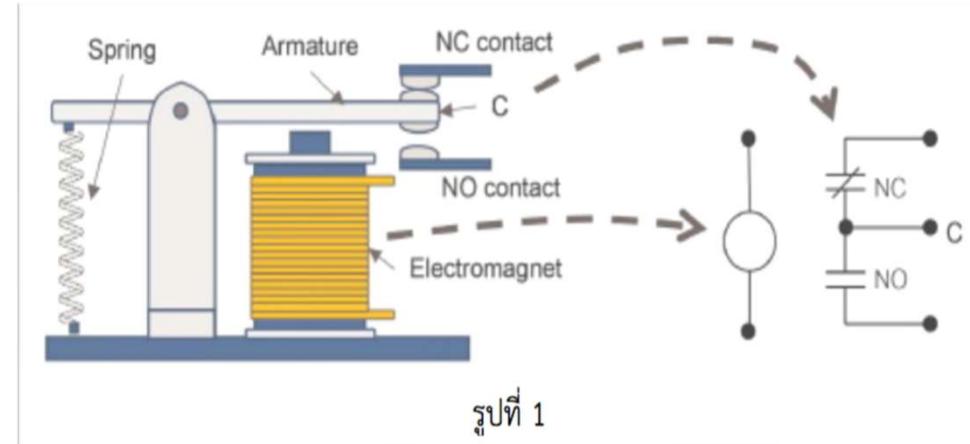
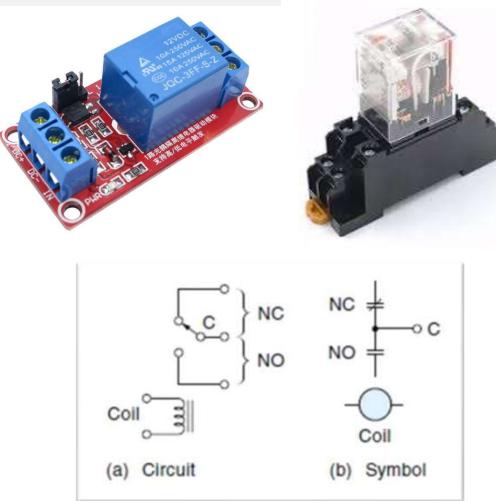
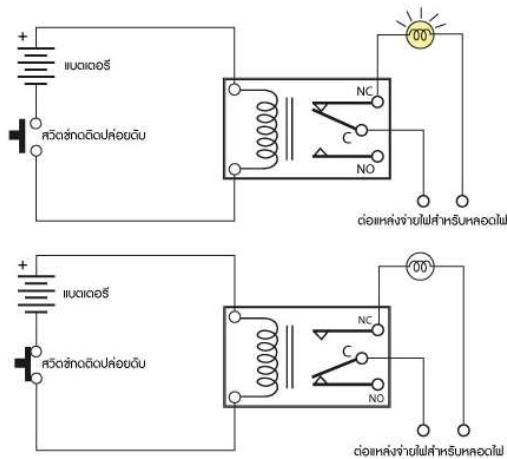
### ปกติหน้าสัมผัสปิด

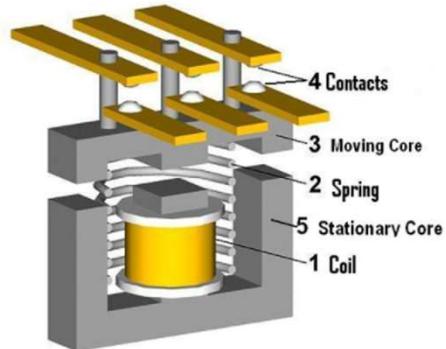
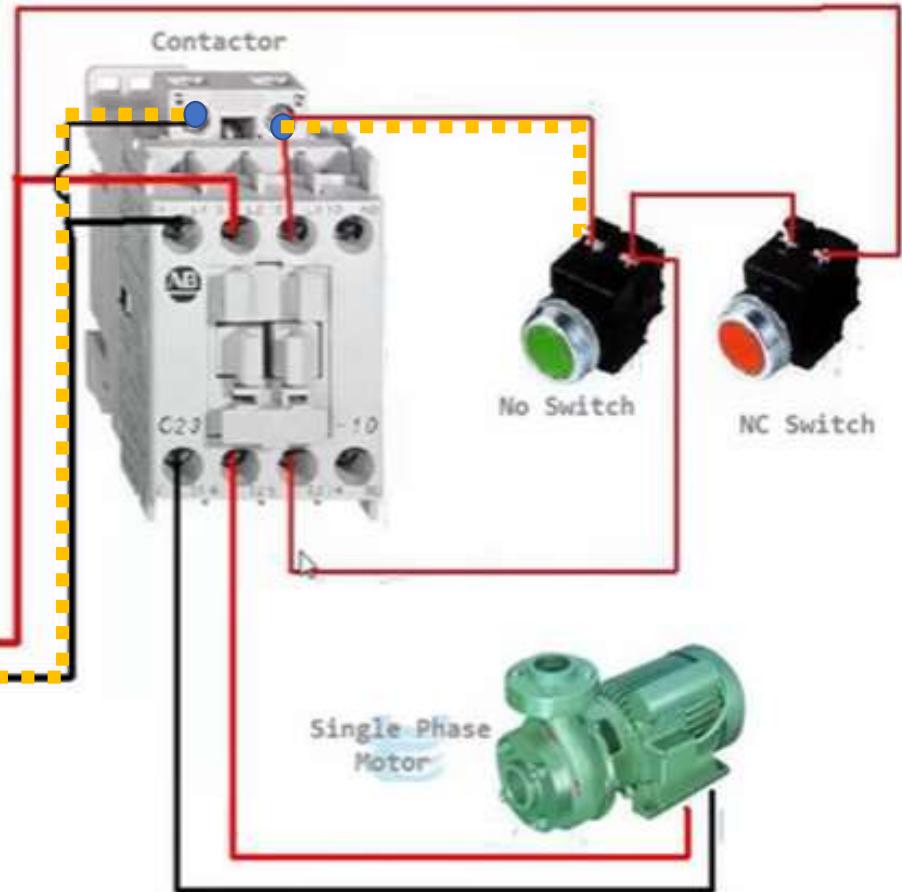
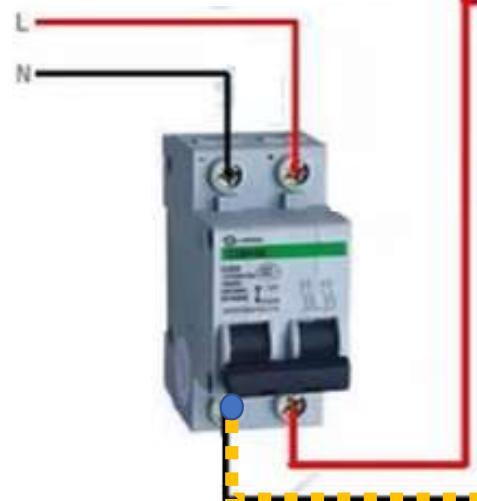
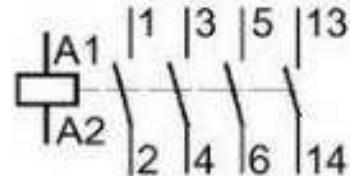
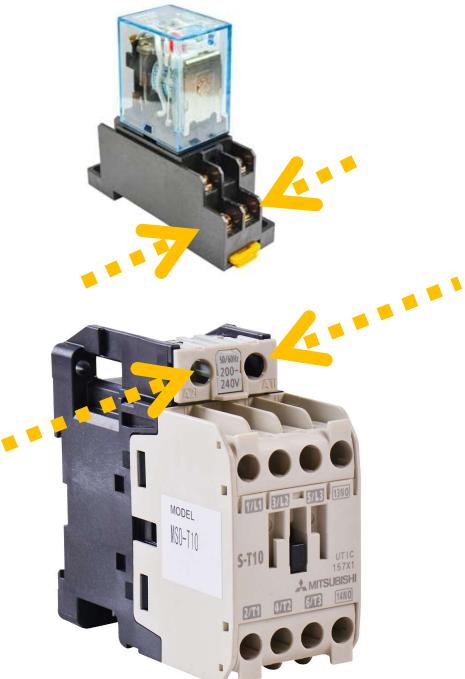


## Basic 01 - Switch Control (NO-NC) youtube

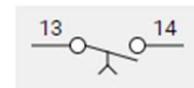
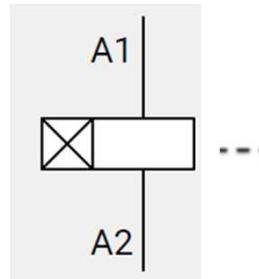


# รีเล耶 Relay ( สวิตช์ที่ใช้กระแสไฟฟ้าในสั่งงาน หน้าสัมผัสเปิดปิด )

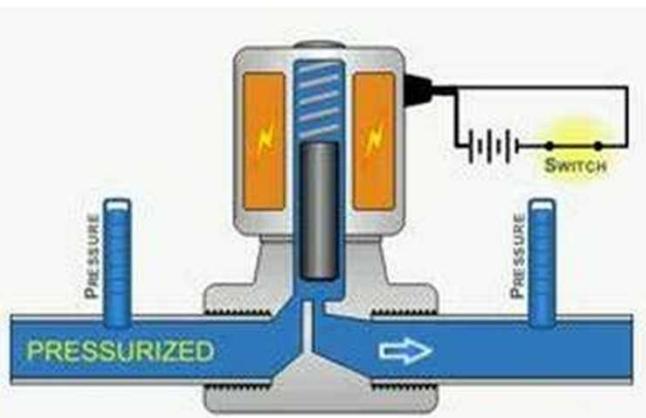
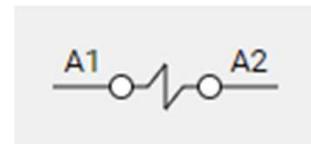
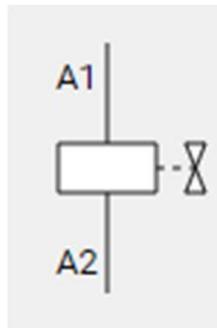




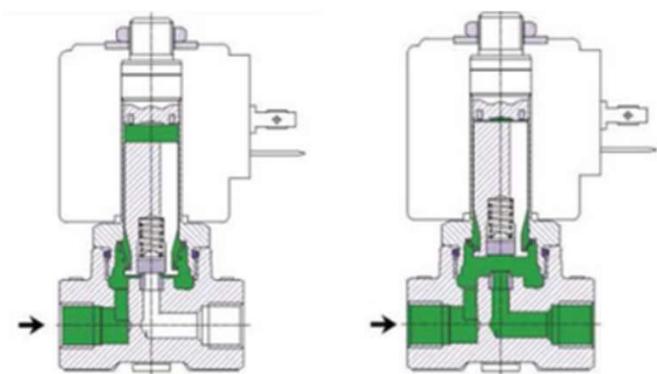
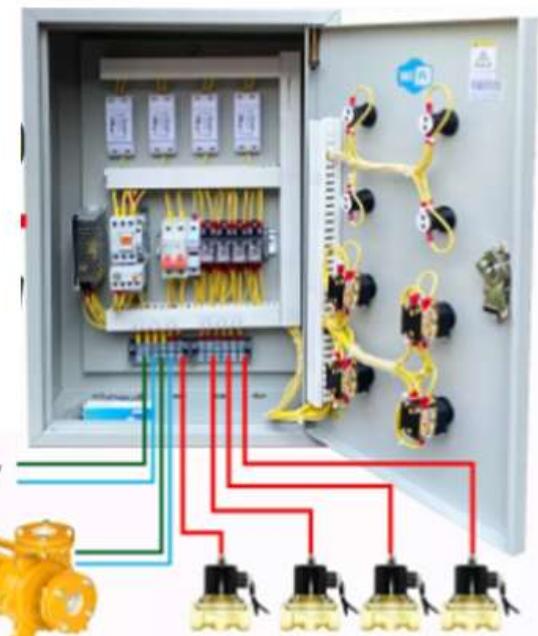
# ตัวตั้งเวลา Timer



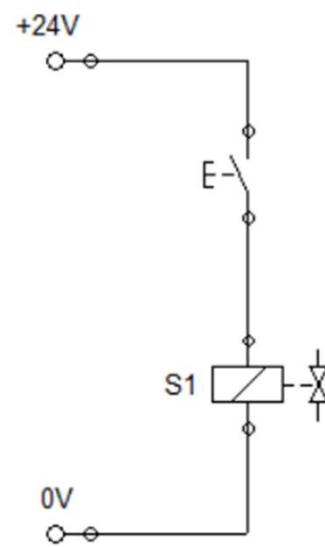
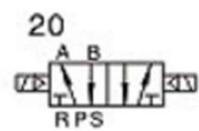
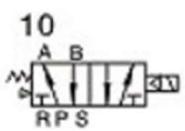
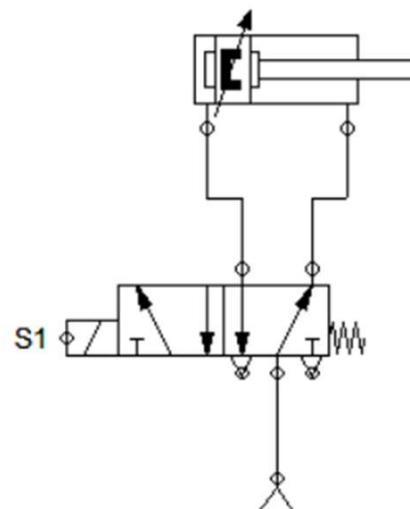
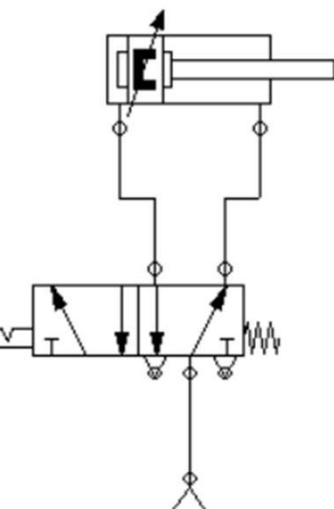
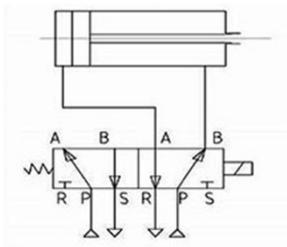
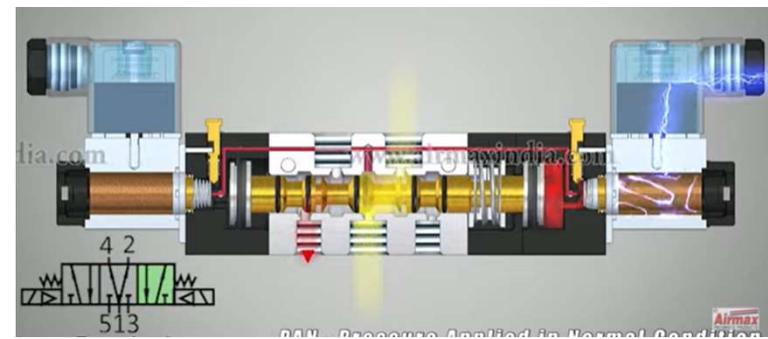
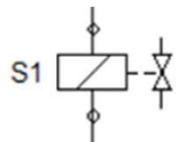
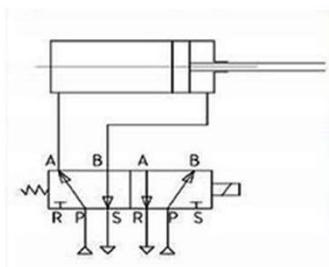
## Solenoid Valves

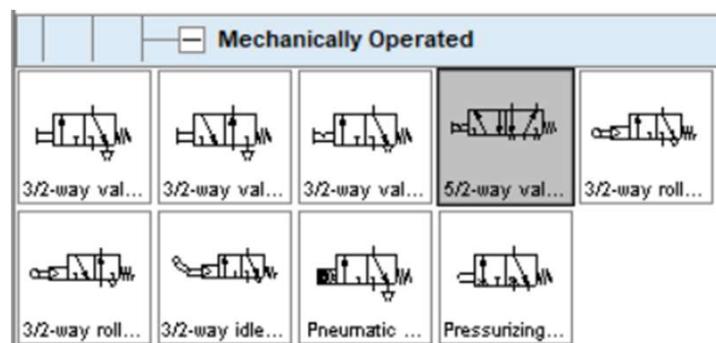
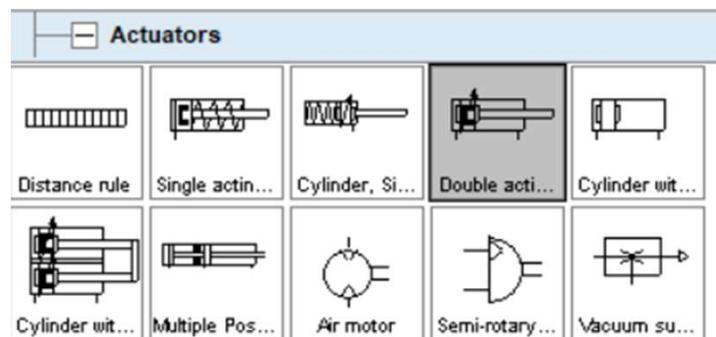


ระบบบาน้ำอัตโนมัติ 4 โซน



# Solenoid Valves





**Configure Way Valve**

**Left Actuation**

Spring-returned  
 Piloted  
 External supply  
 Pneumatic spring  
 External supply

**Description**

5/2-way valve, with selection switch

**Valve Body**

Reversible

Manually   
 Mechanically   
 Pneumatically/  
 Electrically

**Initial Position**

Left  Dominant Signal  Right

**Standard Nominal Flow Rate**  l/min (0.1..5000)

**Mirror**

Horizontal

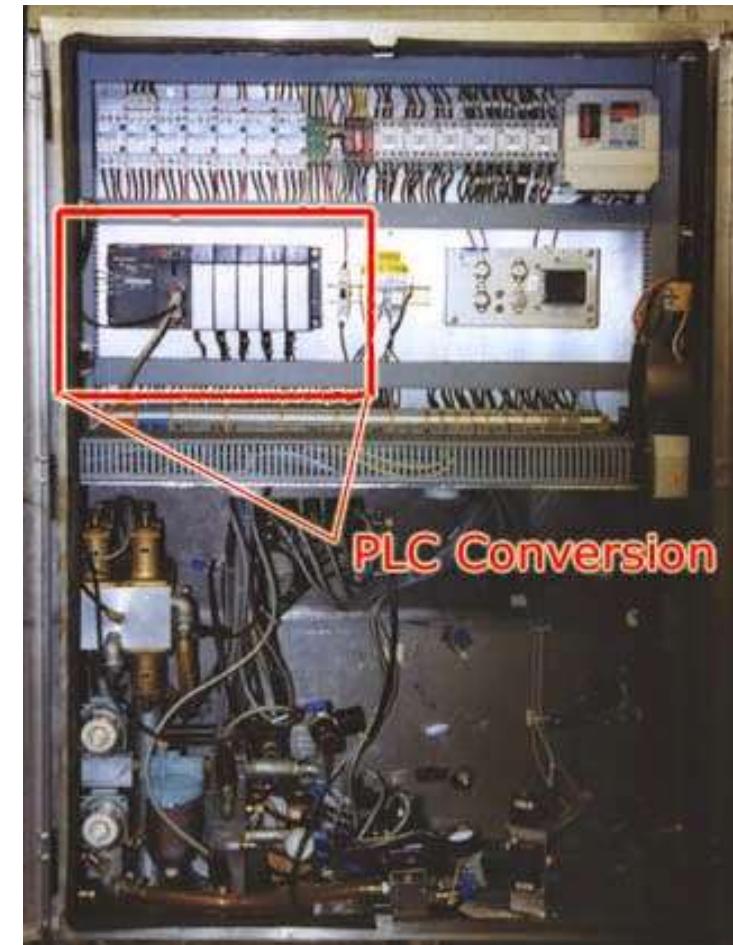
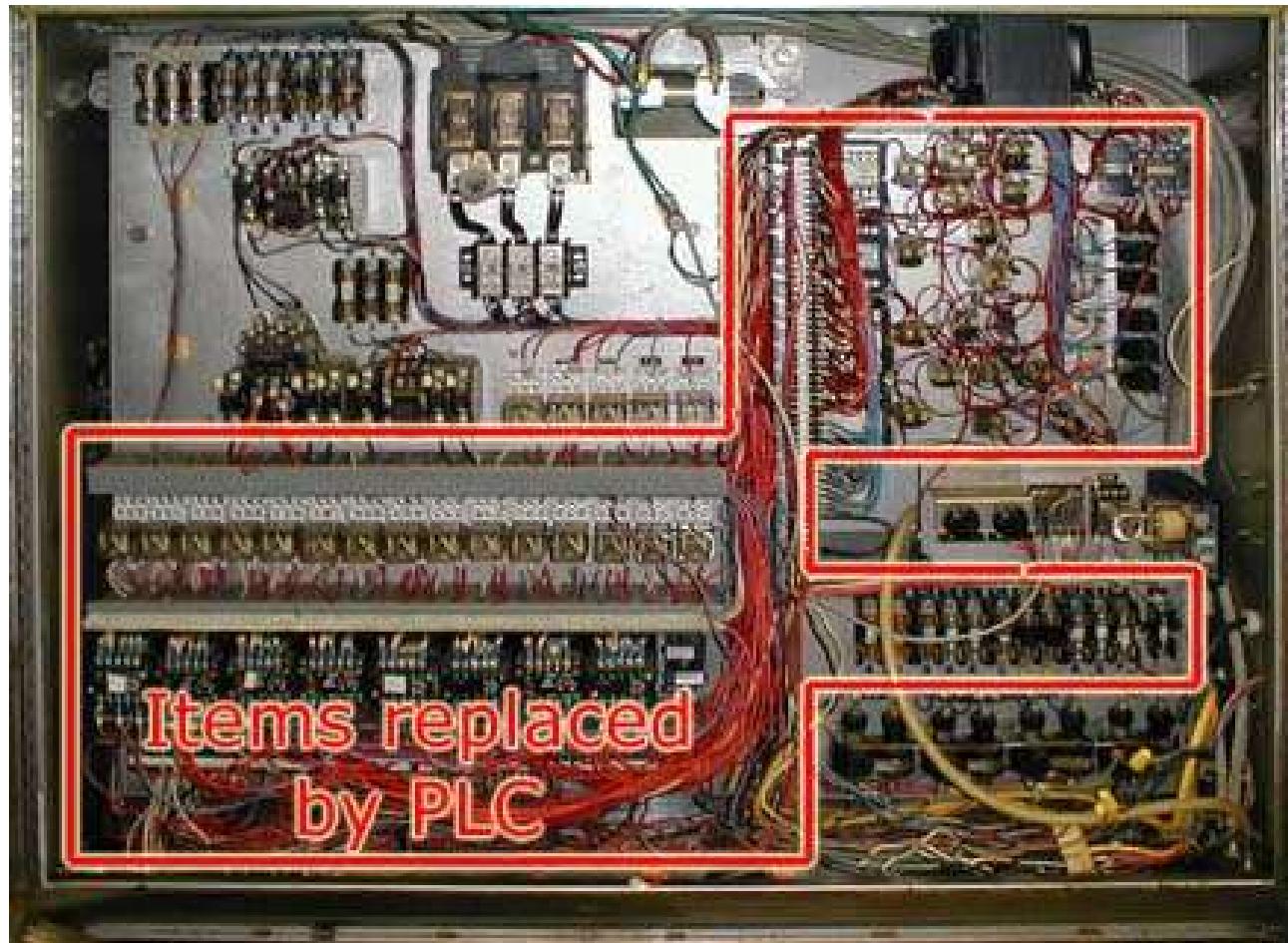
Vertical

**Right Actuation**

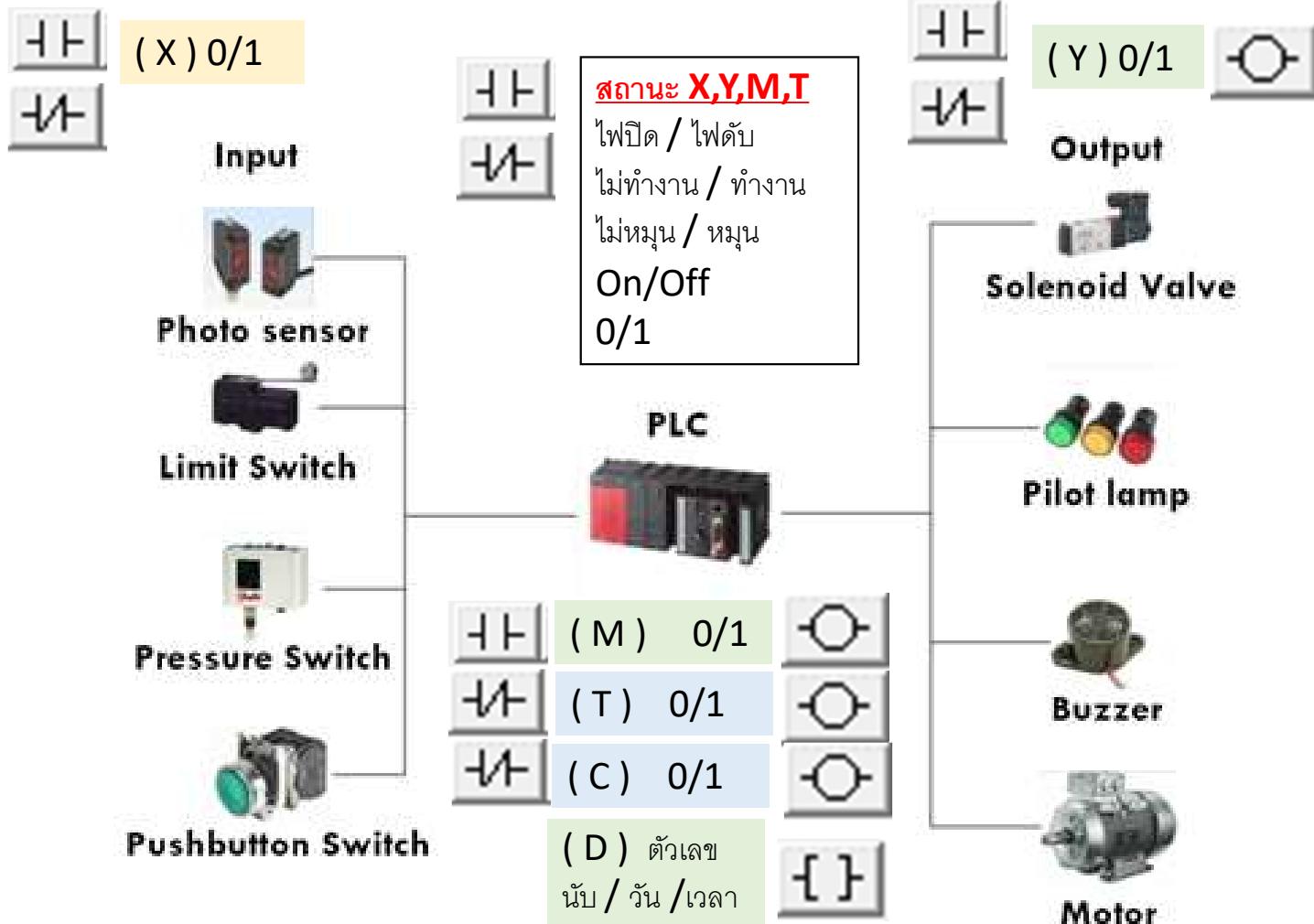
Spring-returned  
 Piloted  
 External supply  
 Pneumatic spring  
 External supply

Manually   
 Mechanically   
 Pneumatically/  
 Electrically

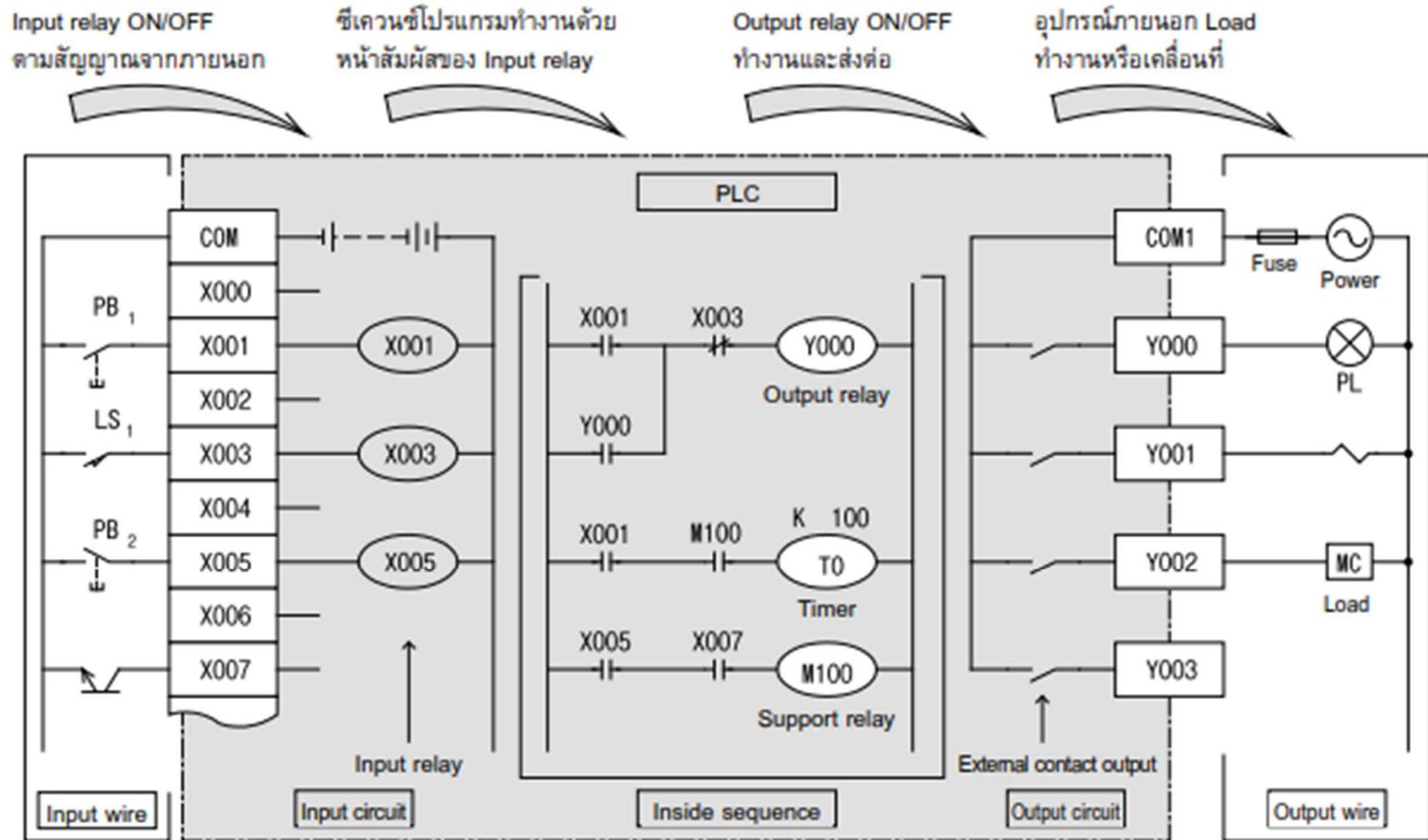
**OK** **Cancel** **Help**

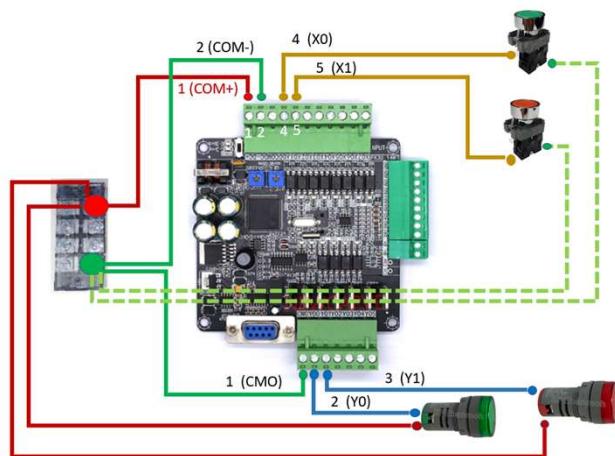


# PLC (Programmable Logic Control)



- ( X ) Input
- ( Y ) Output
- ( M ) Mem Relay
- ( T ) Timer
- ( C ) Counter
- ( D ) Memory



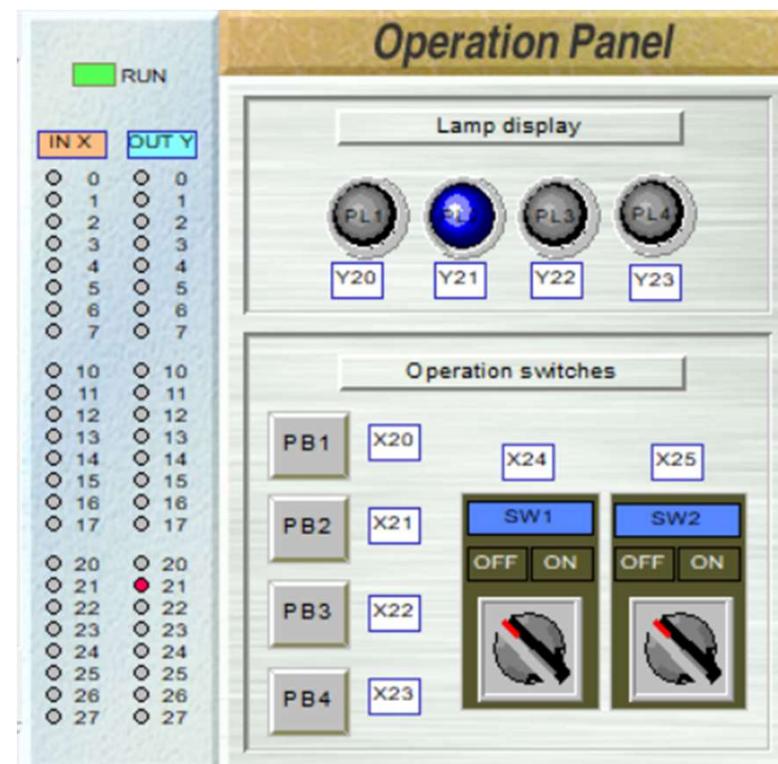
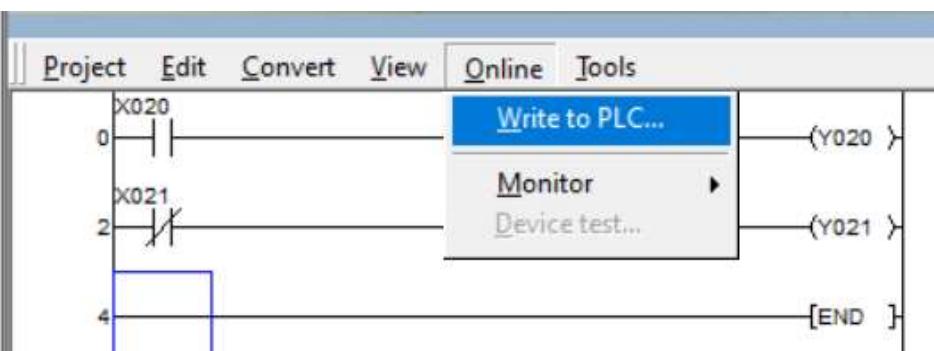
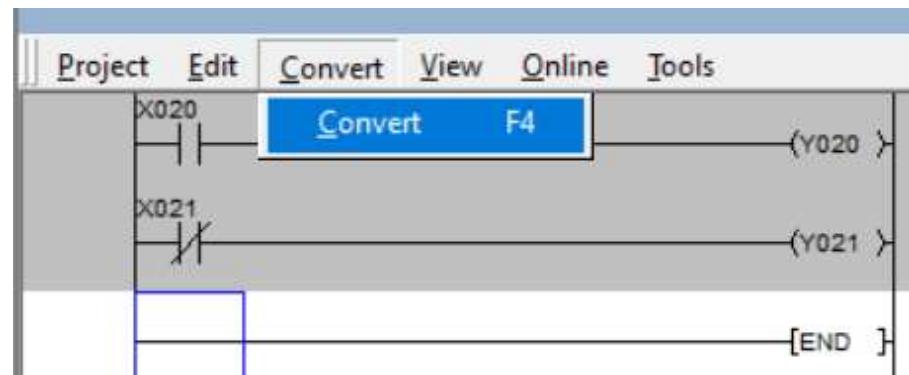
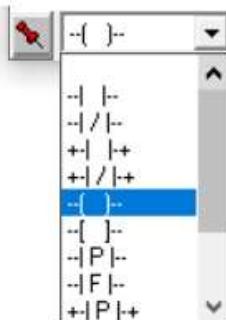


An interactive learning interface for PLC basics. It features a navigation bar at the top with links like 'Let's Learn the FX Series PLC!' and 'Let's Study the Basics'. Below are four main sections: 'B-1. Basic I/O Program' (with a star icon), 'B-2. Standard Program' (with a star icon), 'B-3. Control Precedence Program' (with a star icon), and 'B-4. Reading the Input Status' (with two star icons).

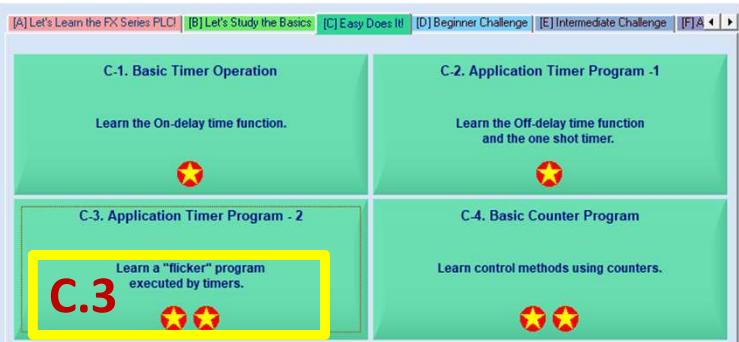
A screenshot of the FX Designer software interface. The top menu includes File, Edit, Simulation, Tools, and Help. The main window displays a ladder logic program titled 'Basic I/O Program'. The program consists of four rungs: Rung 1: X020 → Y020; Rung 2: X021 → Y021; Rung 3: X022 → [END]; Rung 4: X023 → [END]. The right side of the screen shows an 'Operation Panel' with buttons for PLC, Lamp display, Operation switches, and a status table for inputs (IN X) and outputs (DUTY Y). A legend at the bottom defines symbols like Com, F5, F6, etc.



Enter symbol

OK
Cancel


# PLC 04 – Open and Close Door YouTube



This screenshot shows a ladder logic editor interface for a PLC. On the left, there's a toolbar with a cartoon character, buttons for 'Edit Ladder', 'Write to PLC', 'Reset', and three function keys (F, T, S). Below that are buttons for 'Main' and 'RUN'. The main area displays two parallel ladder logic circuits. The top circuit has coil Y000 and input X020 (normally open) in series with input X001 (normally closed). The bottom circuit has coil Y001 and input X021 (normally open) in series with input X000 (normally closed). To the right is a 3D simulation of a door mechanism. Labels indicate X1(Upper limit), Y5(Red), Y6(Green), Y7(Yellow) at the top; Y0(Door up command) and Y1(Door down command) on the door; and X0(Lower limit) at the bottom. A legend on the far right shows PB1/X20 and PB2/X21. The title bar reads 'SW0D5C-FXTRN-BEG-E'.

# PLC 03 – Basic Timer youtube

File Edit Simulation Tools Help

The screenshot shows a PLC ladder logic program. On the left, there's a toolbar with a cartoon character, buttons for 'Edit Ladder', 'Write to PLC', 'Reset', and three function keys (F, T, S). Below that is a 'Main' button and a 'RUN' button. The ladder logic itself has four rungs:

- X020 (normally open contact) connected to the coil of timer T0 with value K30.
- T0 coil connected to output Y000.
- X021 (normally open contact) connected to the coil of timer T1 with value K40.
- T1 coil connected to output Y001.

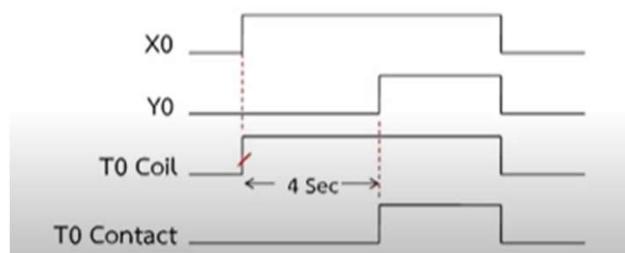
A note in the center says: "3 Press the [F4] key to convert the program you have input."

To the right, there's a 3D model of a door mechanism. It shows a door with a handle, a red limit switch labeled X0 (Lower limit), a green limit switch labeled X1 (Upper limit), and two output contacts labeled Y0 (Door up command) and Y1 (Door down command). To the right of the door, there's a legend: Y5 (Red), Y6 (Green), Y7 (Yellow). At the bottom right, there are two buttons: PB1 (X20) and PB2 (X21).

Basic Timer Operation

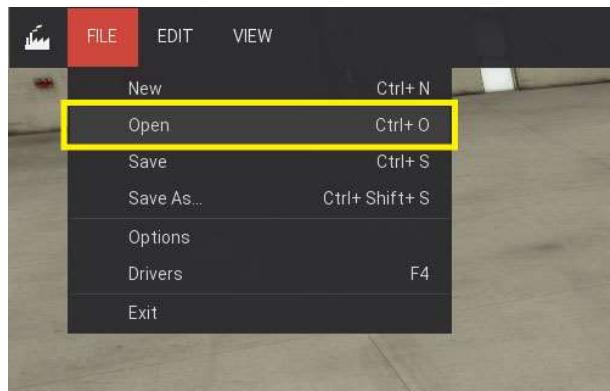
Enter symbol

A dialog box titled "Enter symbol" contains a symbol icon, a dropdown menu, and the text "T0 K30".



[A] Let's Learn the FX Series PLC | [B] Let's Study the Basics | [C] Easy Does It | [D] Beginner Challenge | [E] Intermediate Challenge | [F] G | [H] H

C.1. Basic Timer Operation	C.2. Application Timer Program - 1
Learn the On-delay time function.	Learn the Off-delay time function and the one shot timer.
<b>C.3. Application Timer Program - 2</b>	C.4. Basic Counter Program
<b>C.3</b> Learn a "flicker" program executed by timers.	Learn control methods using counters.



← Open Scene

My Scenes

Scenes

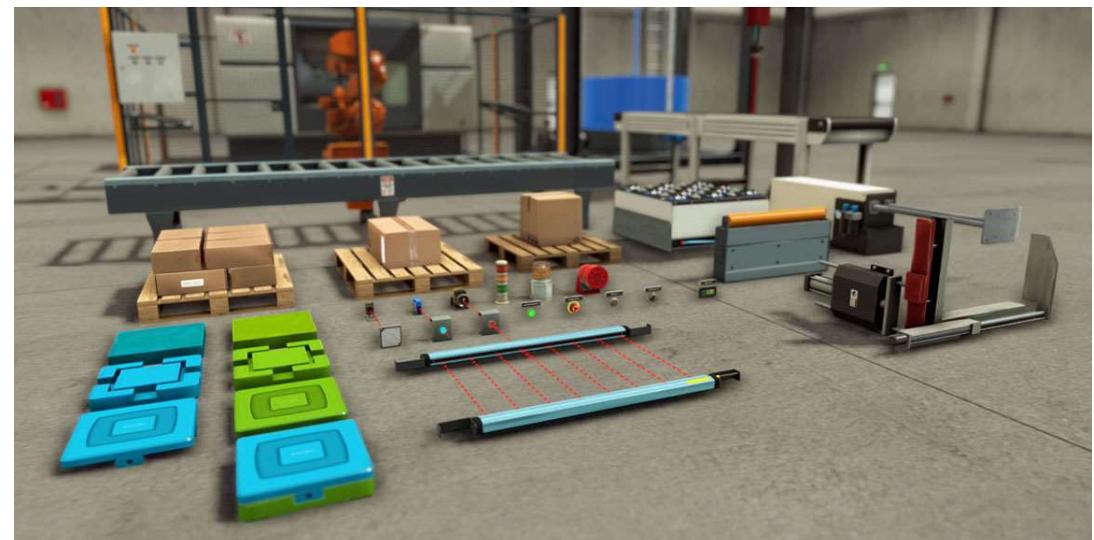
These are scenes inspired by common industrial applications which different challenges ranging from beginner to advanced. They can be edited and used as a base for your own scenes.

 1 - From A to B  
Transport a box until it reaches a sensor.

 2 - From A to B (Set and Reset)  
Transport a box from sensor A to sensor B.

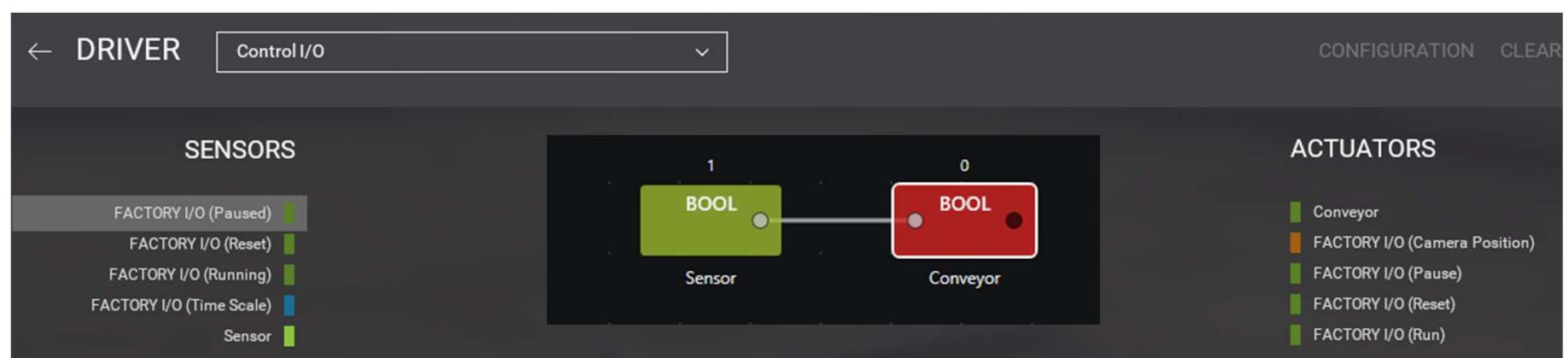
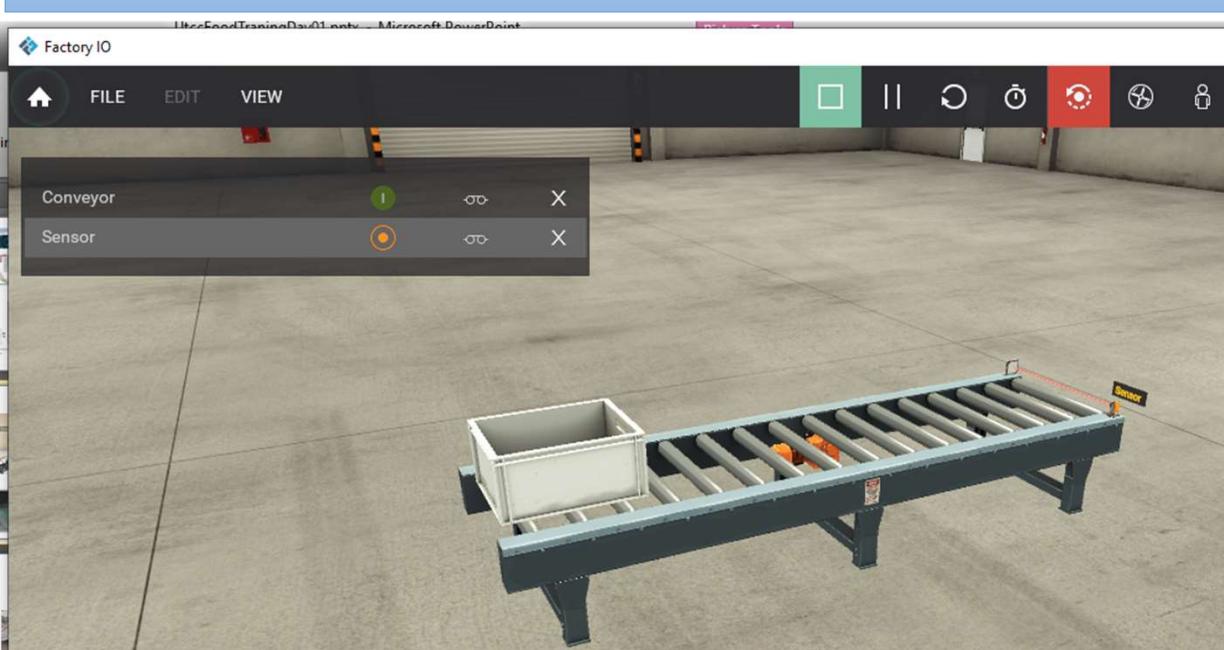
 3 - Filling Tank (Timers)  
Fill and empty a tank using timers.

 4 - Queue of Items (Counters)  
Load and unload three boxes onto a conveyor.



# Basic Control with Machine Simulator (FactoryIO)

YouTube FactoryIO - 03 Basic Control I/O

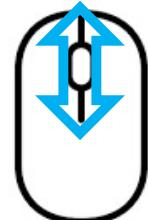


## Dock All Tags

Conveyor	:	0			X	
Sensor	:	1			X	
Start Button 1	:	0			X	
Start Button 1 (Light)	:	1		FORCED	X	
Stop Button 1	:	2			X	
Stop Button 1 (Light)	:	2			X	



Zoom In/Out



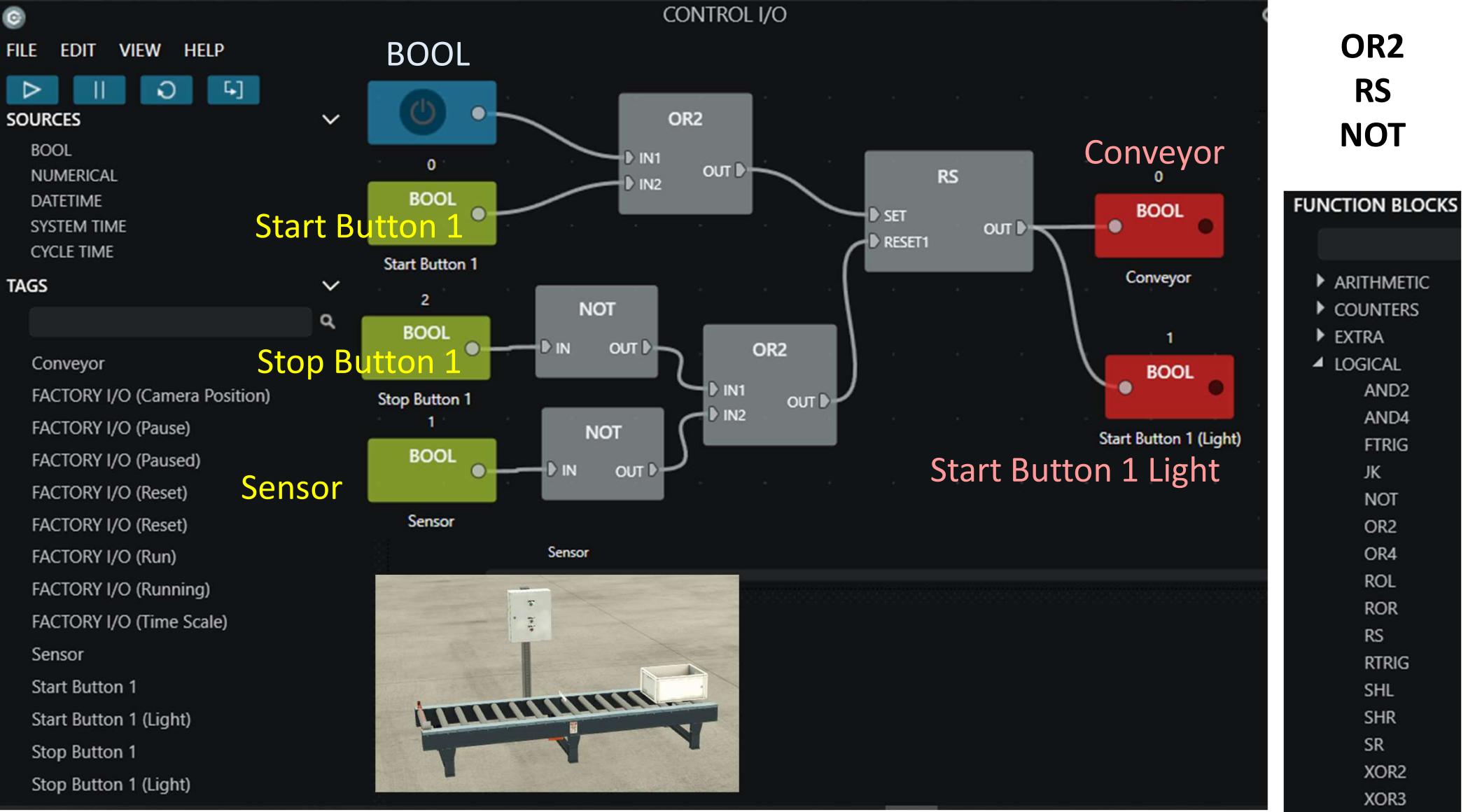
Rotate



## Operators

### Search





## YouTube FactoryIO - 04 Counter



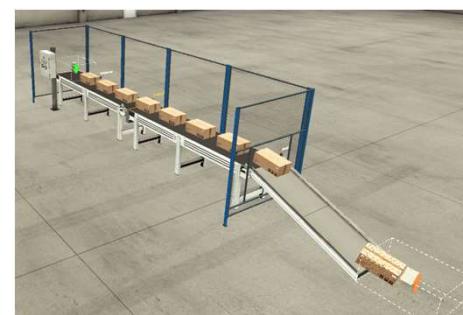
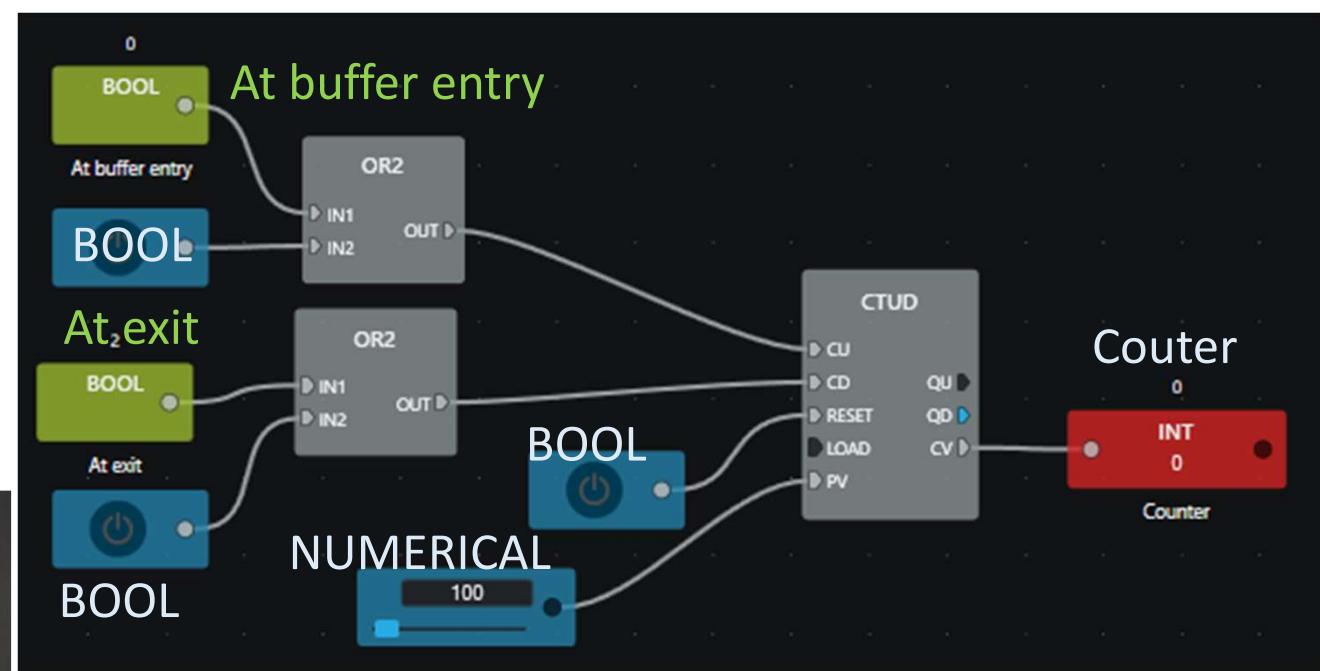
Buffer Station

Buffer and separate up to five boxes.

← DRIVER

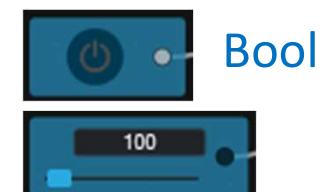
Control I/O

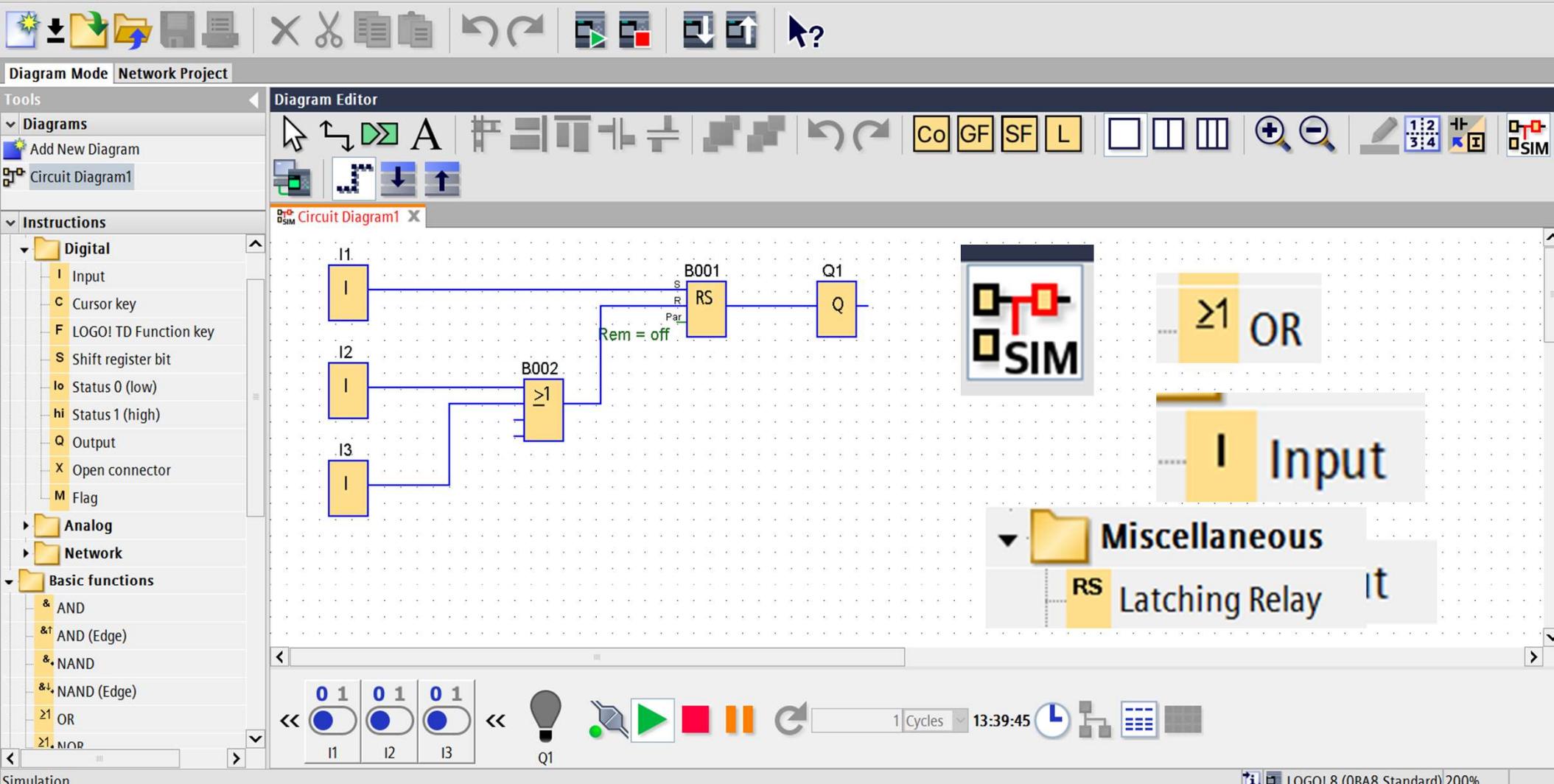
At buffer entry	<input type="radio"/>	<input type="radio"/>
At buffer exit	<input type="radio"/>	<input type="radio"/>
At exit	<input type="radio"/>	<input type="radio"/>
Auto	<input type="radio"/>	<input type="radio"/>
Buffer conveyor	<input type="checkbox"/>	2.8 FORCED
Buffer Vel.	<input type="checkbox"/>	0.0 <input type="radio"/>
Counter	0	<input type="radio"/>
Emergency stop	<input type="radio"/>	<input type="radio"/>
Emitter	<input type="radio"/>	FORCED
Exit conveyor	<input type="checkbox"/>	2.8 FORCED



### SOURCES

- BOOL
- NUMERICAL
- DATETIME
- SYSTEM TIME
- CYCLE TIME





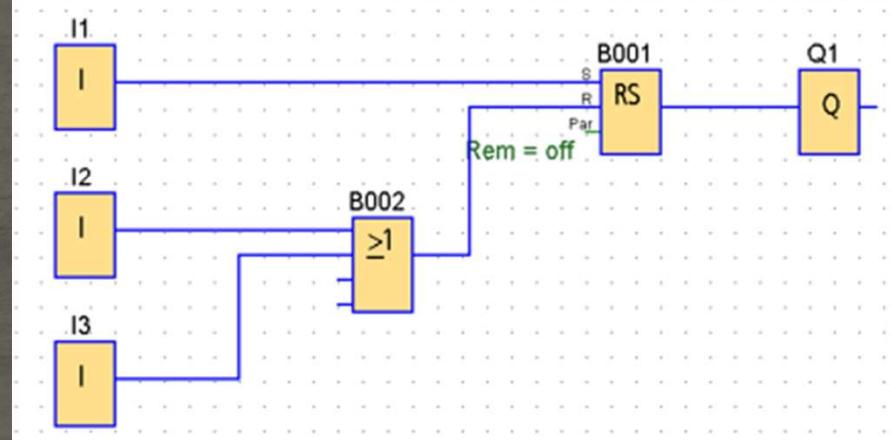
# LOGO! 8 (OBA8.Standard)

DRIVER

Siemens LOGO!

## SENSORS

- FACTORY I/O (Paused)
- FACTORY I/O (Reset)
- FACTORY I/O (Running)
- FACTORY I/O (Time Scale)
- Sensor



Host: 192.168.1.101

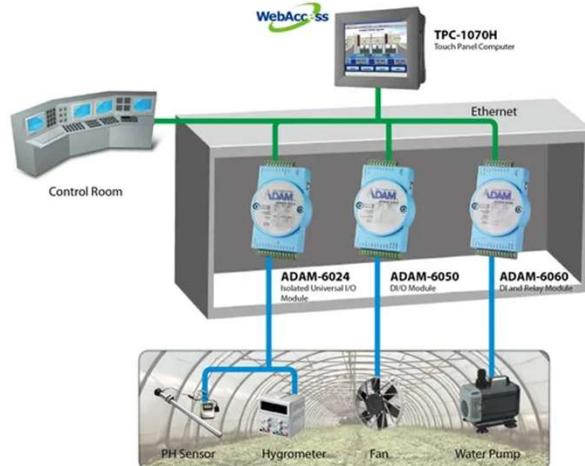
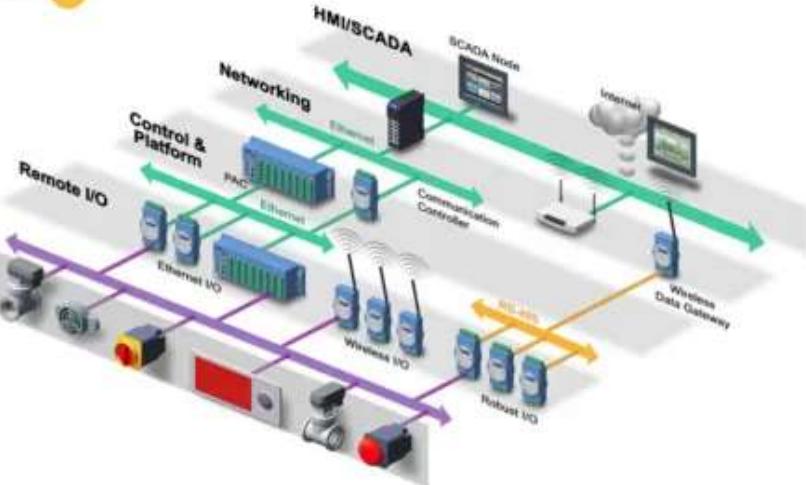
- Sensor
- V0.0
  - V0.1

FACTORY I/O (Running)

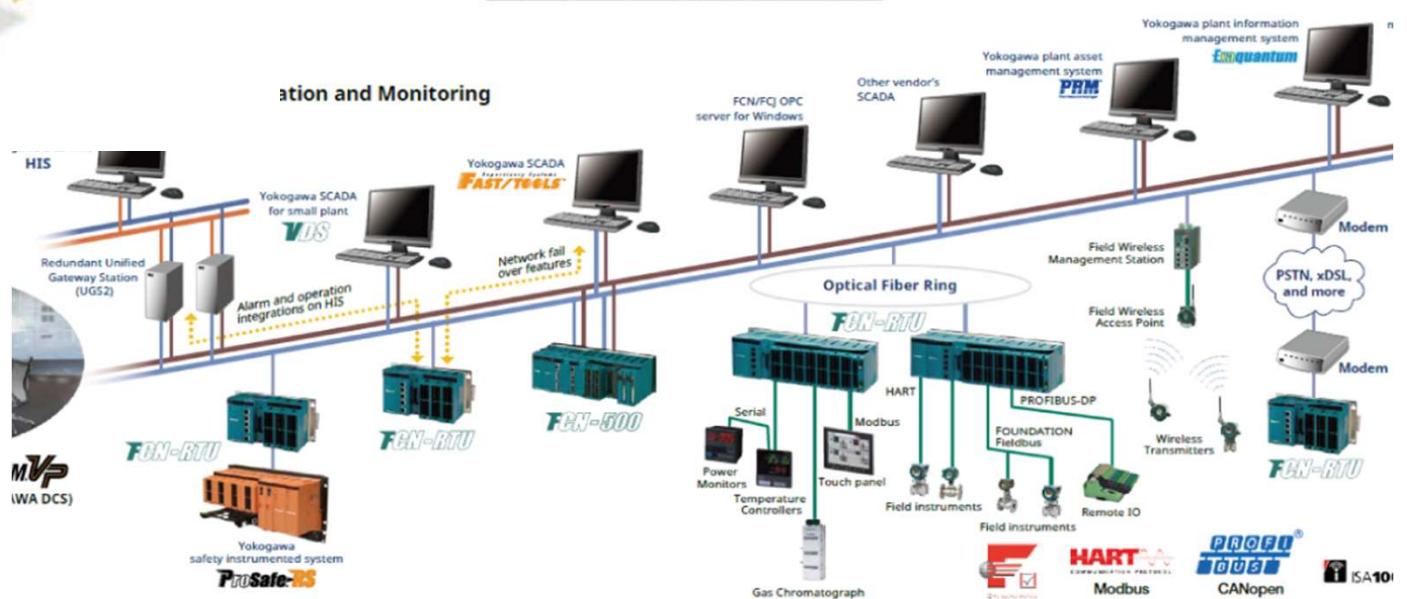
Q1 Conveyor

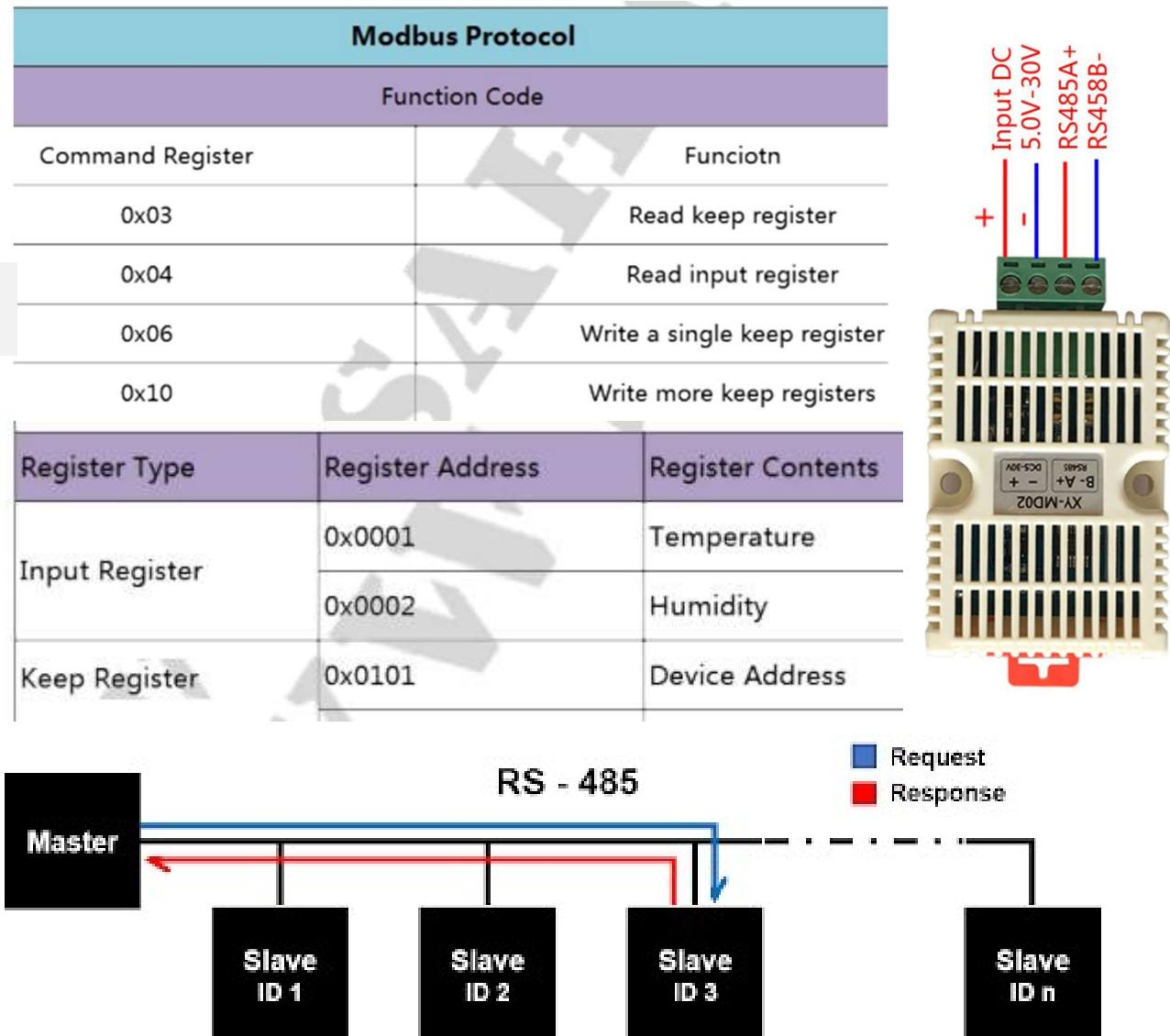
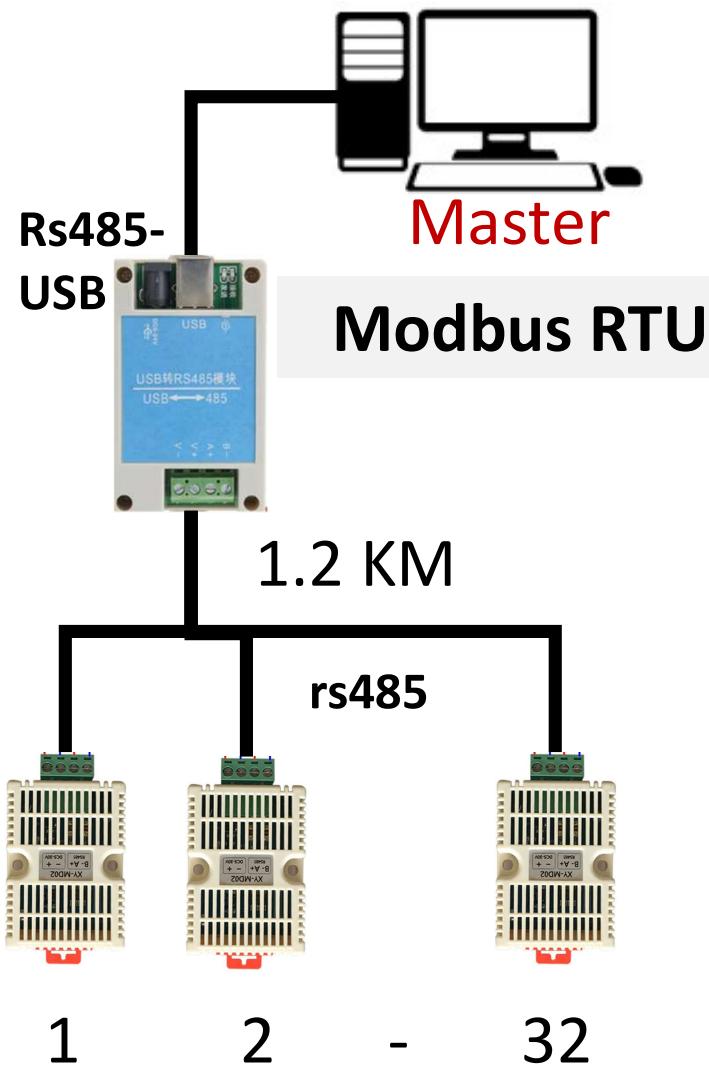
Powered by ComDrvS7  
www.mhj-tools.com



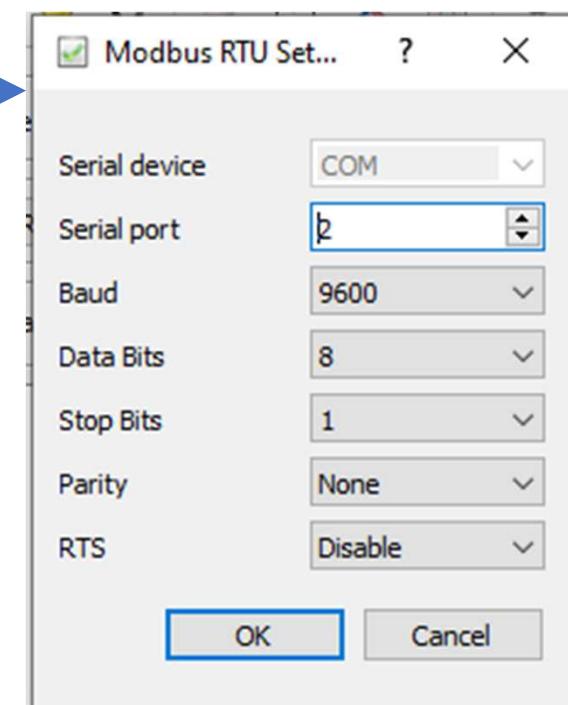
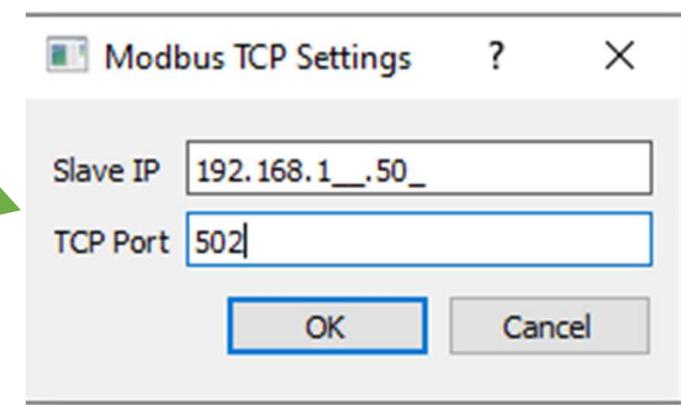
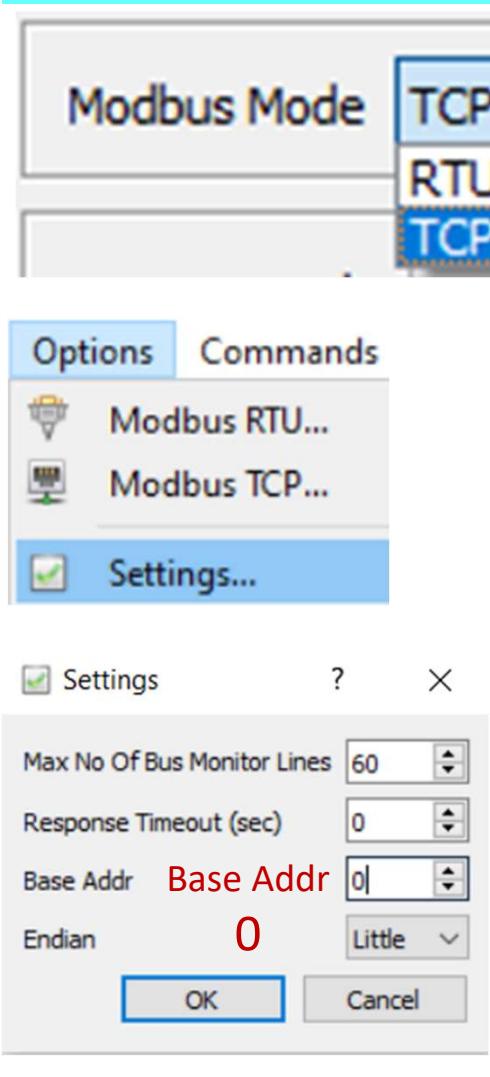


## Integration and Monitoring

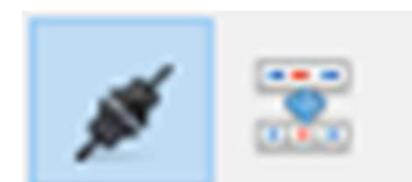




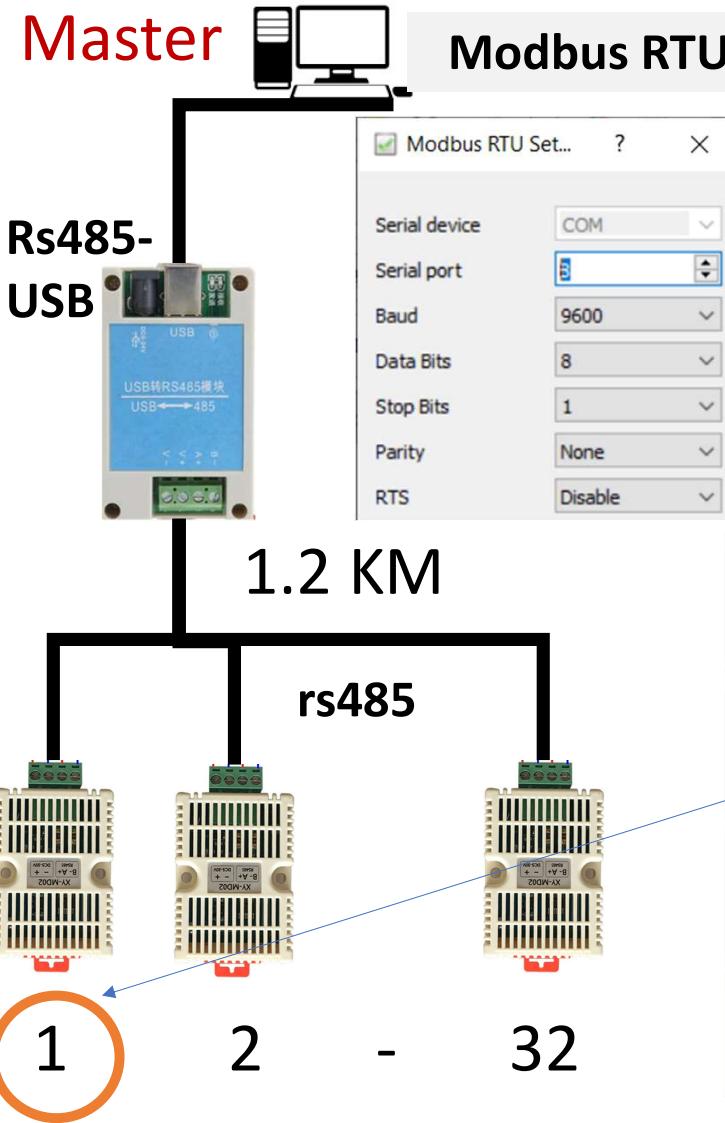
## การใช้งานโปรแกรม QModMaster (Modbus Master)



Connect/Disconnect



Send Command



**Modbus RTU**

Register Type	Register Address	Register Contents
Input Register	0x0001	Temperature
	0x0002	Humidity
Keep Register	0x0101	Device Address

Serial Port ? Baud 9600 Data Bits 8 Stop Bits 1 Parity None

Modbus Mode RTU Slave Addr 1 Scan Rate (ms) 3000

Function Code Read Input Registers (0x04) Start Address 1

Number of Registers 2 Data Format Dec Signed

256	654	x	x	x	x	x	x	x	x
-----	-----	---	---	---	---	---	---	---	---

## Set Device No

**(0x03) Read**

Start Address 257  
(Device No)  
# Single , Multiple

**(0x06) Write**

Start Address 257  
(Device No)  
# 06 Only Write Single

Function Code: Read Holding Registers (0x03)

Start Address: 257

Number of Registers: 1

Data Format: Dec

Signed:

Modbus Mode: RTU

Slave Addr: 1

Scan Rate (ms): 2000

Function Code: Read Holding Registers (0x03)

Start Address: 256

Number of Coils: 1

Data Format: Dec

Signed:

Start Address: 101

Dec

Hex

Function Code: Write Single Register (0x06)

Start Address: 257

Number of Registers: 1

Data Format: Dec

Signed:

Modbus Mode: RTU

Slave Addr: 1

Scan Rate (ms): 2000

Function Code: Write Single Register (0x06)

Start Address: 256

Number of Registers: 1

Data Format: Dec

Signed:

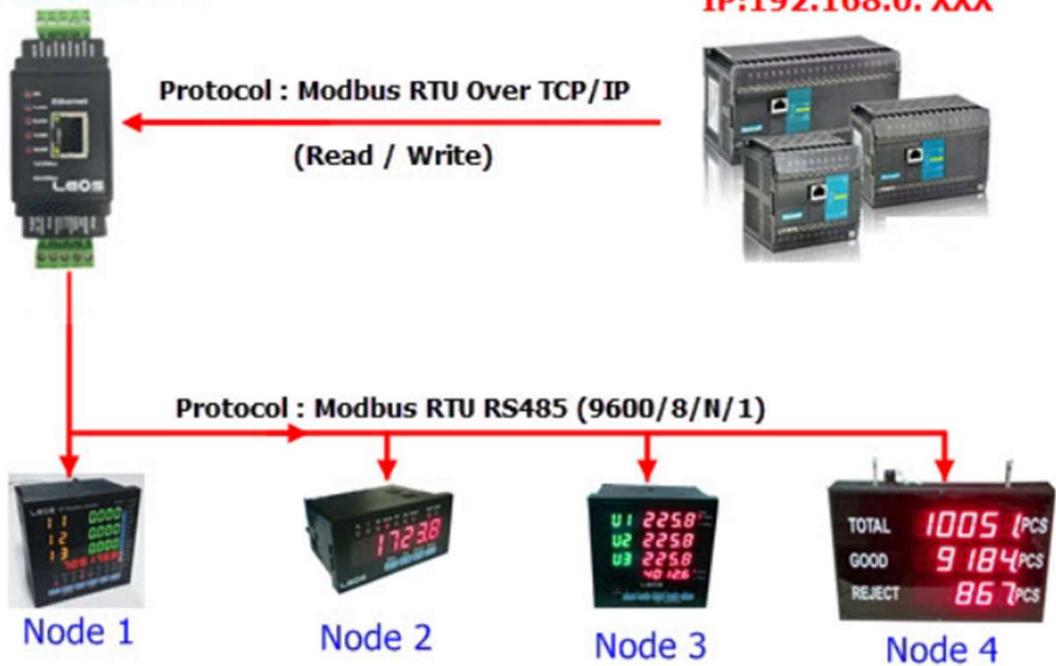
Start Address: 257

Dec

2

Protocol converter  
MPL-J01  
IP:192.168.0.244

PLC : Modbus TCP/IP  
IP:192.168.0. XXX

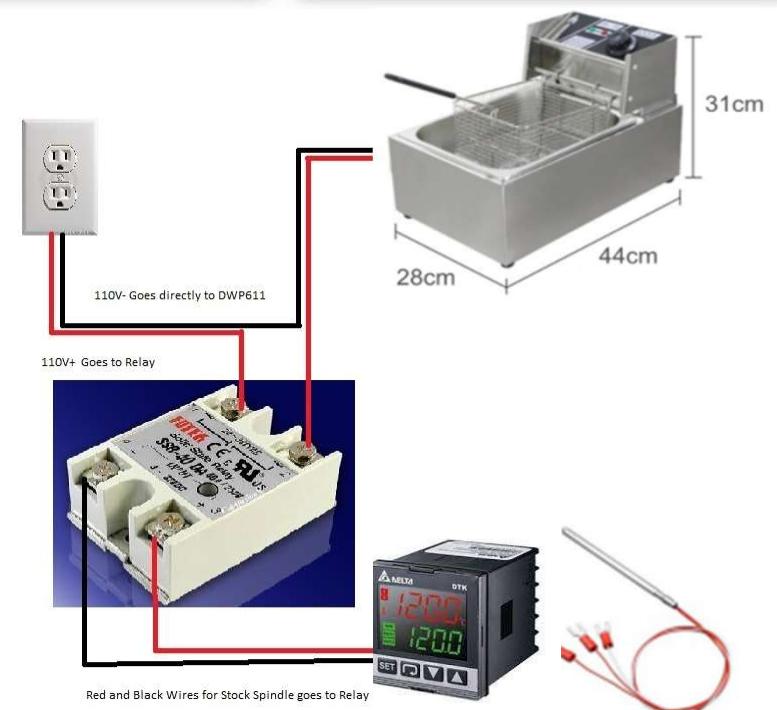
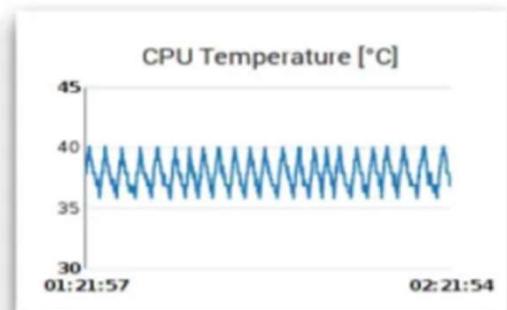
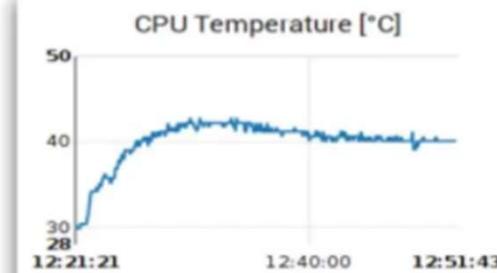


# Temperature PID Control

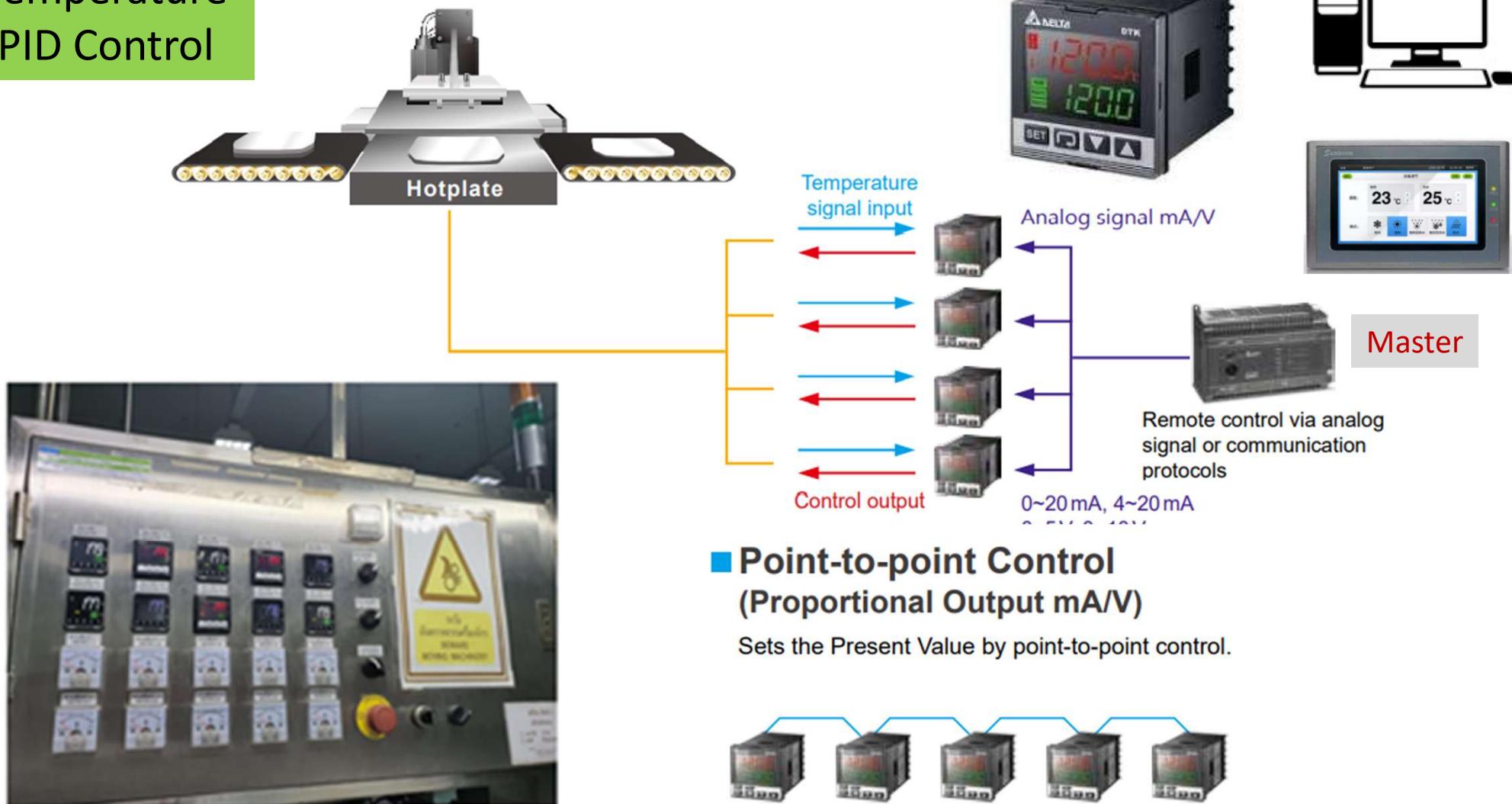
[DTK4848V12](#)



**PID control   VS   On/Off control**



## Temperature PID Control



## Delta Temperature Controller Model

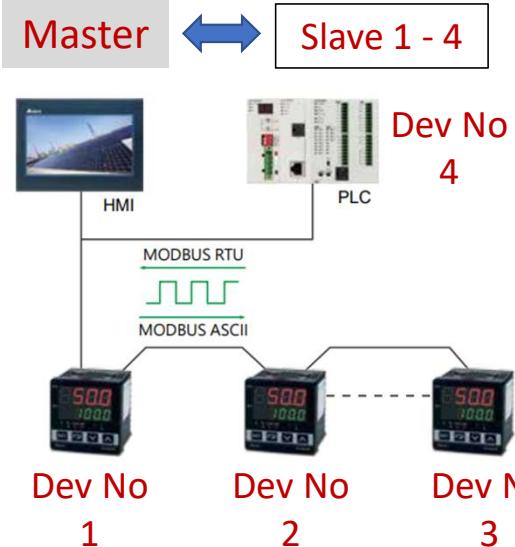
:DTK4848V12

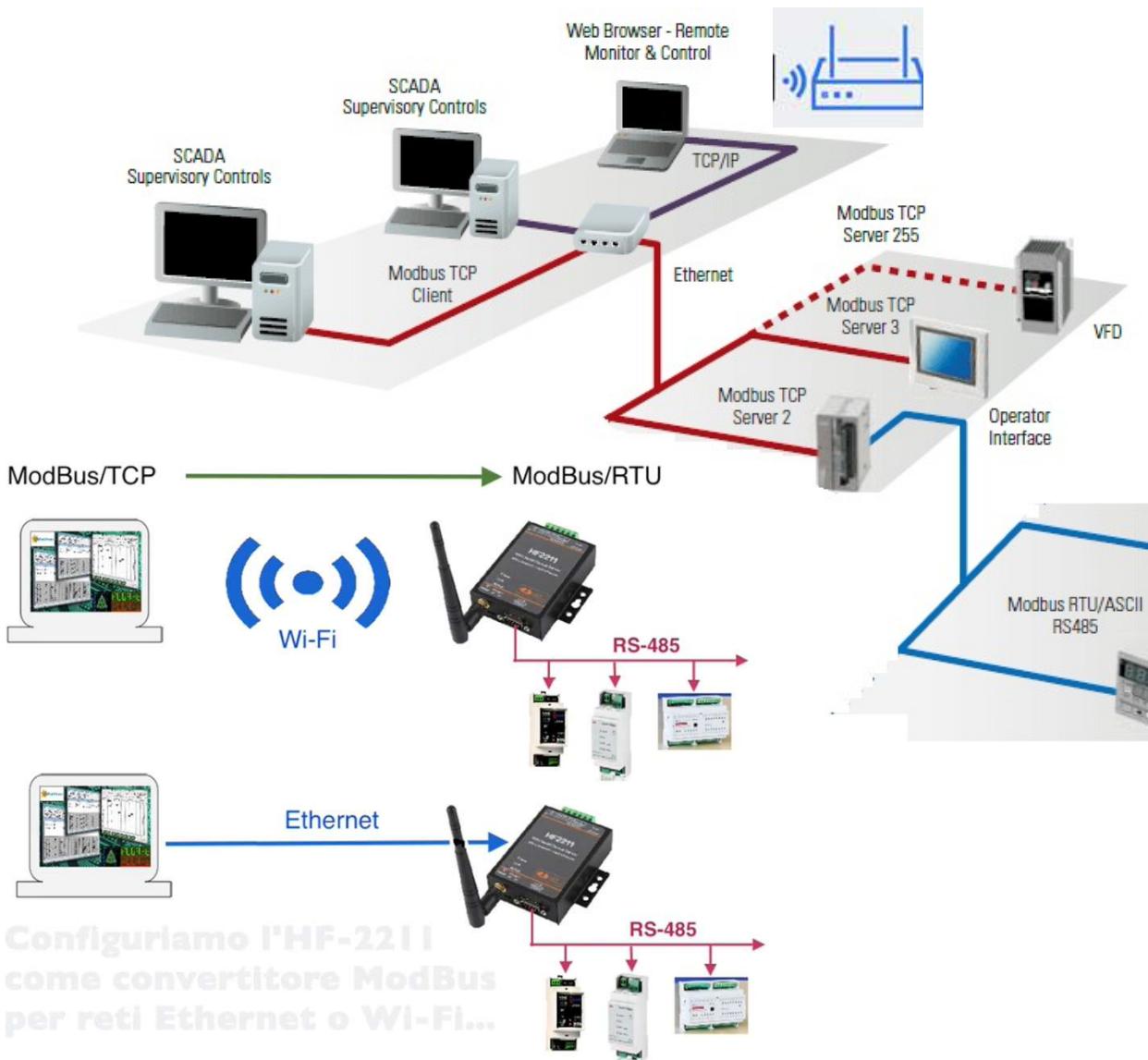
## RS-485 Communication



DT3 supports baudrate 2,400 to 38,400 bps, MODBUS ASCII/RTU protocol, function code 03H and reads maximum 8 words from the register.

Address	Content	Definition
1000H	Present value (PV)	Measuring unit: 0.1 scale. The following values read mean error occurs. 8002H: Temperature not yet acquired 8003H: Not connected to sensor 8004H: Incorrect sensor
1001H	Set value (SV)	Measuring unit: 0.1 scale
1002H	Upper limit of temp. range	Cannot exceed the default value
1003H	Lower limit of temp. range	Cannot fall below the default value
1005H	Control mode	0: PID, 1: ON/OFF, 2: Manual, 3: FUZZY
1006H	Heating/ Cooling control	0: Heating/ Heating, 1: Cooling/ Heating, 2: Heating/ Cooling, 3: Cooling/ Cooling
1007H	1 <sup>st</sup> Heating/ Cooling control cycle	0.1 ~ 99 sec.
1008H	2 <sup>nd</sup> Heating/ Cooling control cycle	0.1 ~ 99 sec.
1009H	Proportional band (PB)	0.1 ~ 999.9
100AH	Ti value	0 ~ 9999
100BH	Td value	0 ~ 9999
1012H	Read/write Output 1 volume	Unit: 0.1%, only valid in manual control mode
1013H	Read/write Output 2 volume	Unit: 0.1%, only valid in manual control mode





## Modbus TCP (Wifi /Ethernet)



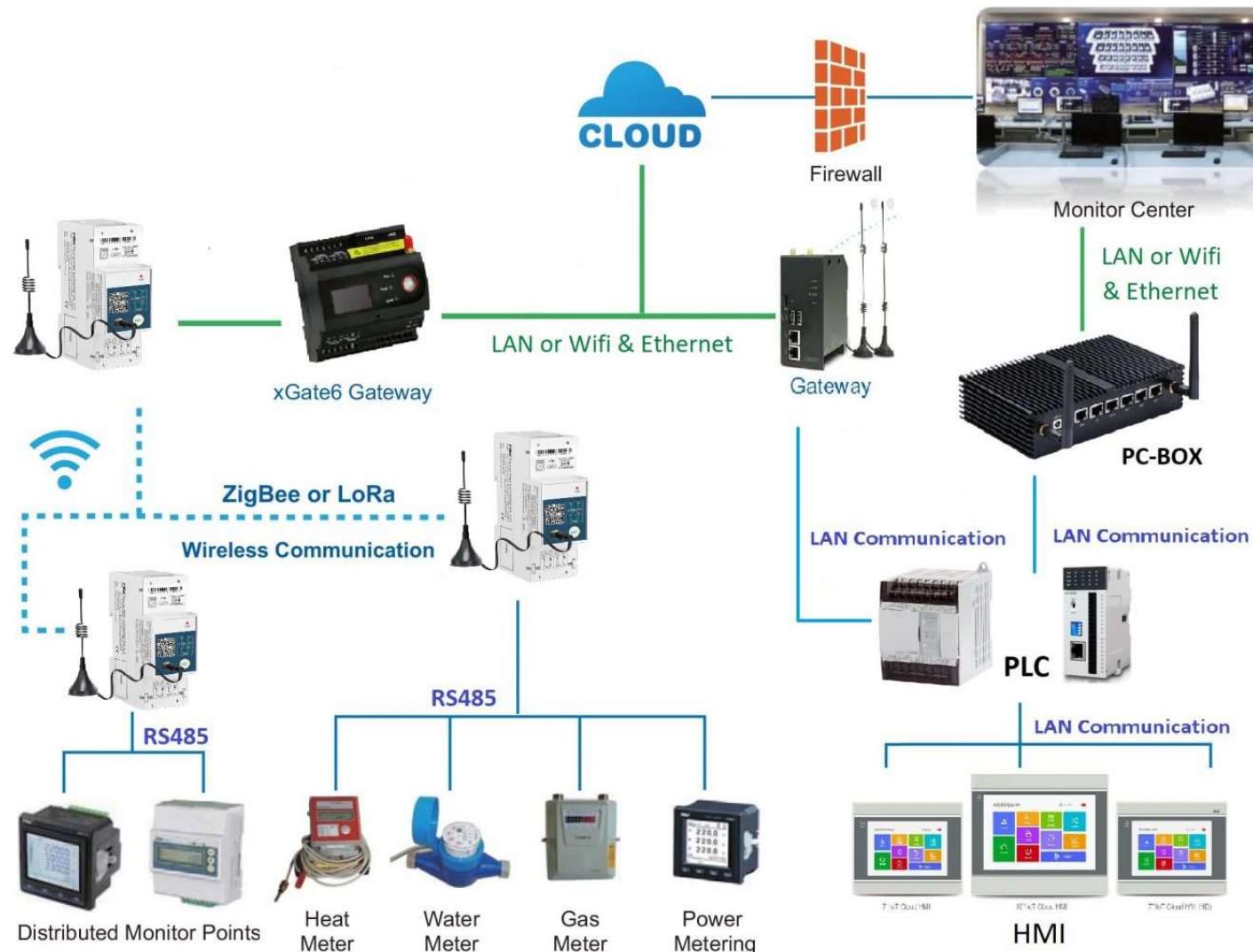
Modbus RTC

Software & Cloud

Data Collection Gateway

Wireless Comm.  
LAN Comm.

Smart Device



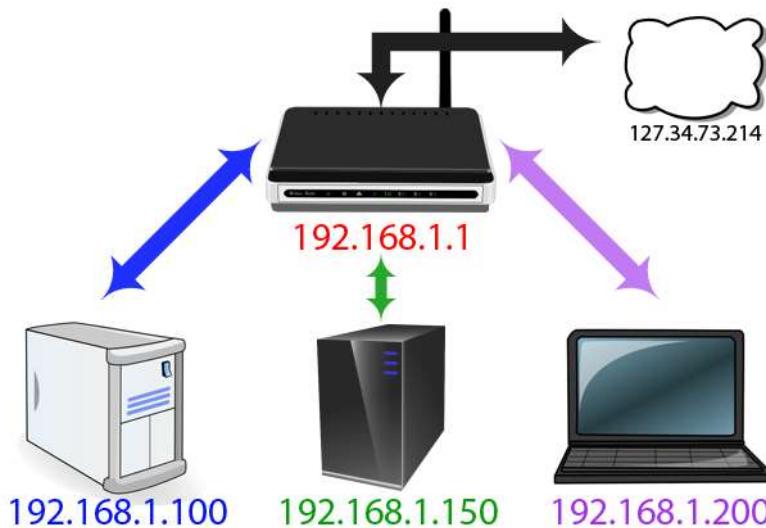
<http://www.hi-flying.com/>

<https://www.youtube.com/watch?v=iSv0HaSmPRO>

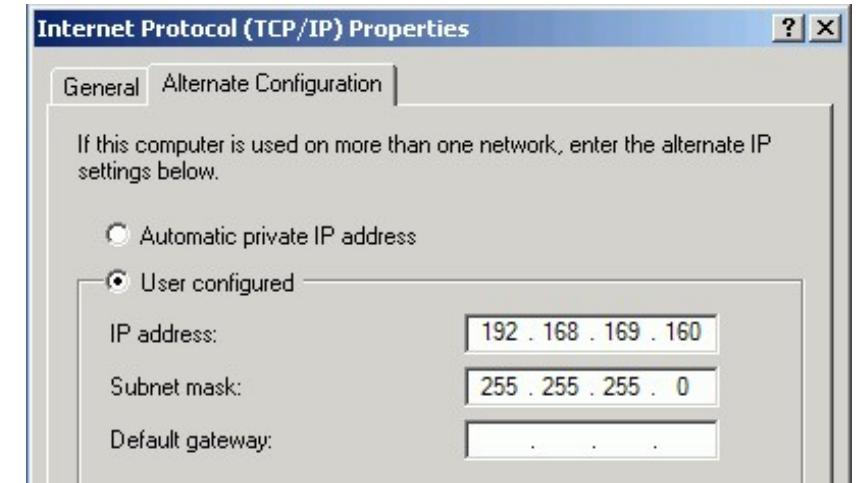
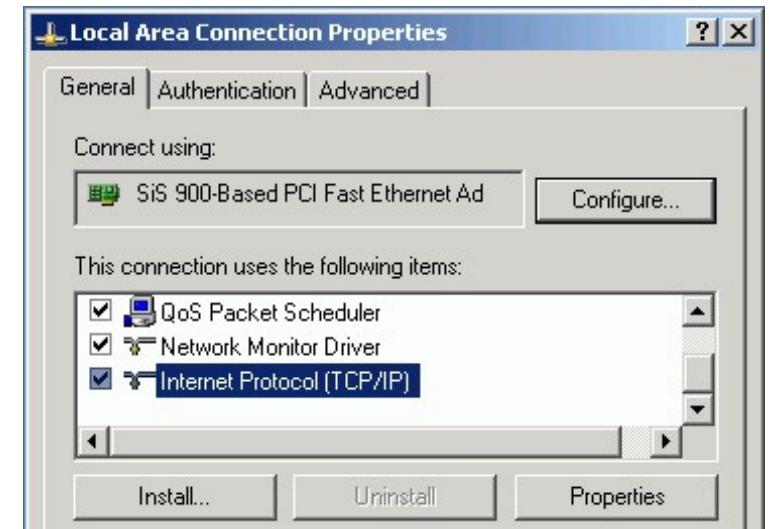
## Elfin Series Assembly Drawing



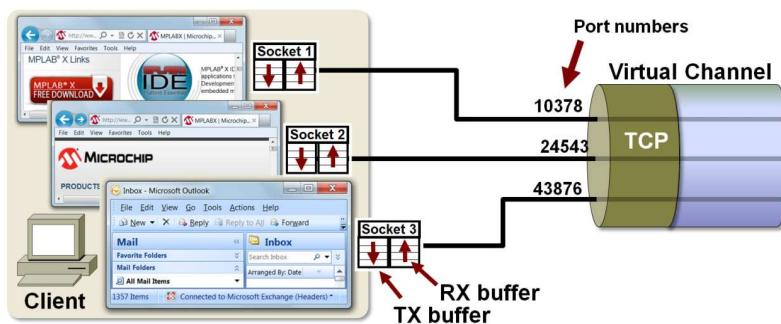
IP Address  
192.168.1.1  
Subnet Mask  
255.255.255.0



192.168.1.xxx  
192.168.1.150  
192.168.1.100  
192.168.1.200



IP : Port



Cmd: ipconfig

## การใช้งานโปรแกรม SimModbus 2

### Modbus Devices Simulator (Simulating Modbus Client/Slave)



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

Connected (0/10) : (received/sent) (0/0) Serv. listening. Rx: ● Tx: ●

Address: C H D I/O Holding Regs (40000) Fmt: decimal +/- Prot: MODBUS TCP/IP

Address +0 +3 +4 +5 +6 +7 +8

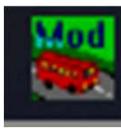
Address	+0	+3	+4	+5	+6	+7	+8
400001-400010	0	0	0	0	0	0	0
400011-400020	0	0	0	0	0	0	0
400021-400030	0	0	0	0	0	0	0
400031-400040	0	0	0	0	0	0	0
400041-400050	0	0	0	0	0	0	0
400051-400060	0	0	0	0	0	0	0
400061-400070	0	0	0	0	0	0	0
400071-400080	0	0	0	0	0	0	0
400081-400090	0	0	0	0	0	0	0
400091-400100	0	0	0	0	0	0	0
400101-400110	0	0	0	0	0	0	0
400111-400120	0	0	0	0	0	0	0
400121-400130	0	0	0	0	0	0	0
400131-400140	0	0	0	0	0	0	0
400141-400150	0	0	0	0	0	0	0

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0G 0H 0I 0J 0K 0L 0M 0N 0O 0P 0Q 0R 0S 0T 0U 0V

OK Cancel IP Your Com (127.0.0.1) Port 502

Load register values at startup.

## การใช้งานโปรแกรม SimModbus 2 จำลอง Modbus Memory Device (TCP)



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

Connected (0/10) : (received/sent) (0/0) Serv. listening. Rx

Address:  H  I/O Holding Regs (400000) ▾

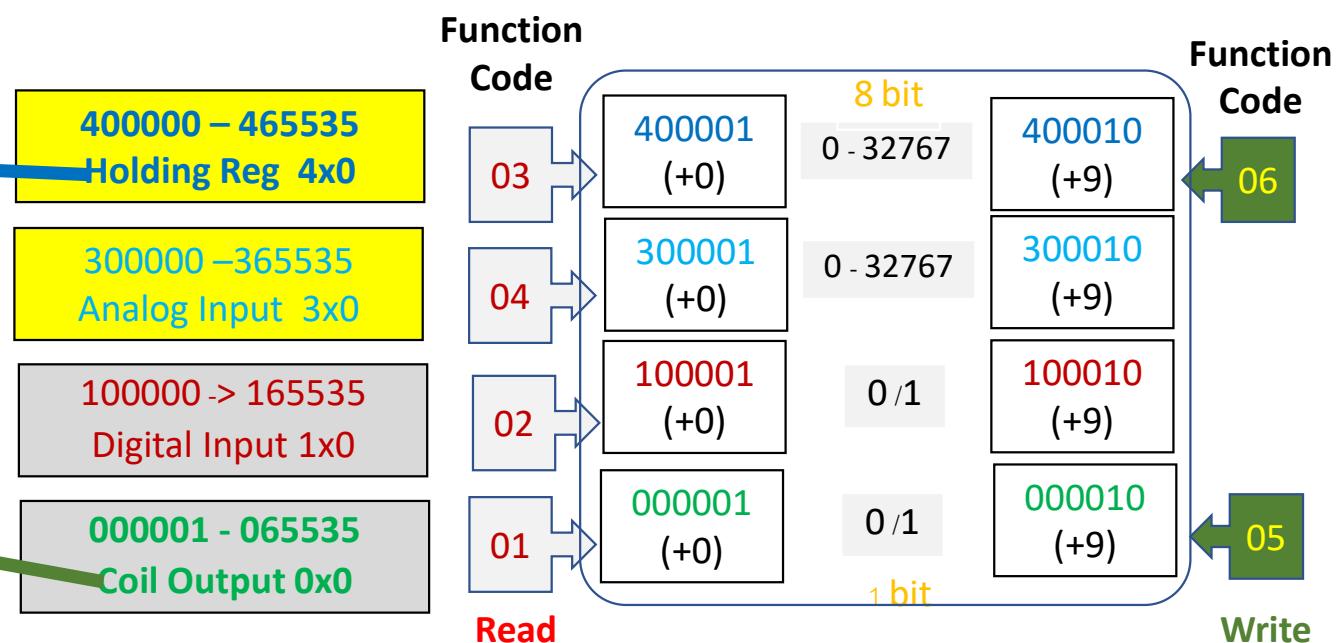
Address	+0	+1	+2	+3	+4
400001-400010	830	32767	0	0	0
400011-400020	0	0	0	0	0

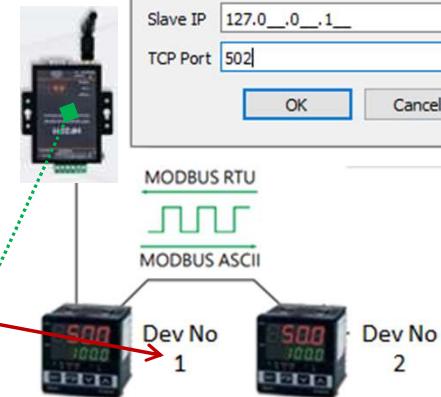
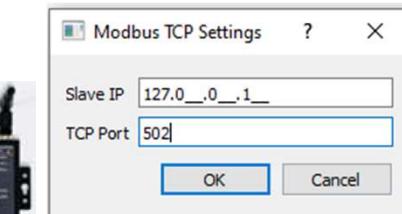
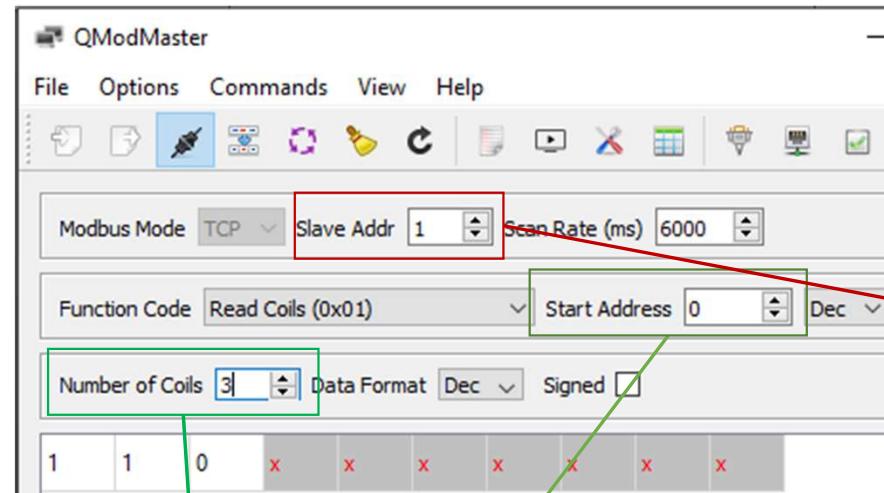
Connected (0/10) : (received/sent) (0/0) Serv. listening

Address:  H  I/O Coil Outputs (000000) ▾

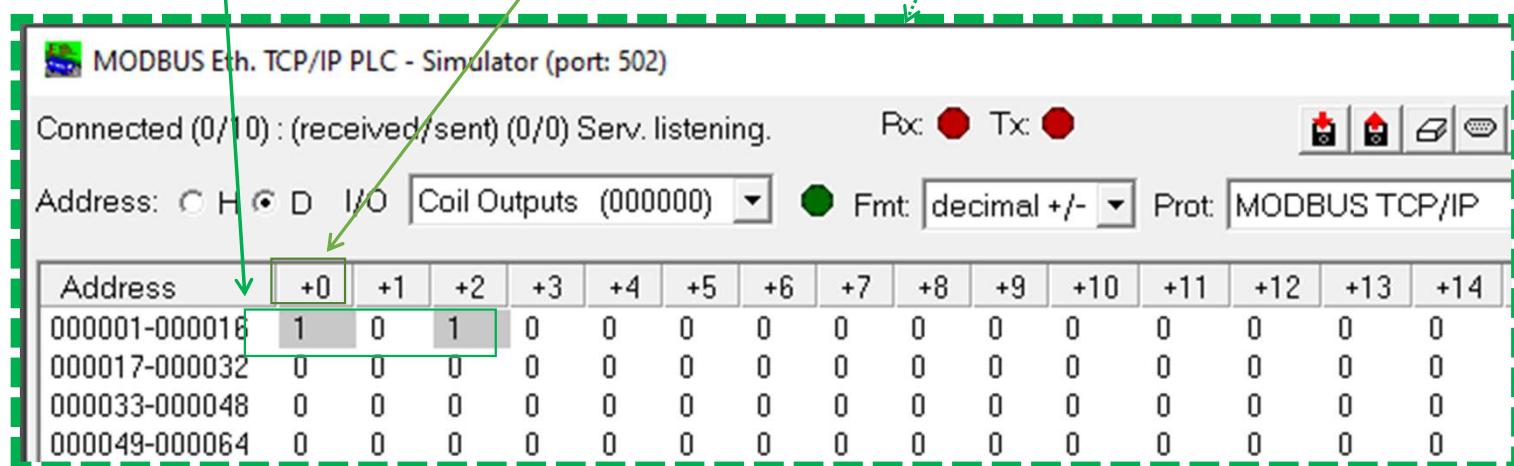
Address	+0	+1	+2	+3	+4	+5
000001-000016	1	0	1	0	0	0
000017-000032	0	0	0	0	0	0
000033-000048	0	0	0	0	0	0
000049-000064	0	0	0	0	0	0



หน่วยความจำเก็บข้อมูลภายในอุปกรณ์



## Device Simulator



Modbus Mode: TCP Slave Addr: 1 Scan Rate (ms): 2000

Function Code: Read Coils (0x01) Start Address: 0 Dec

Number of Coils: 1 Data Format: Dec Signed:

Read Coils (0x01)  
Read Discrete Inputs (0x02)  
Read Holding Registers (0x03)  
Read Input Registers (0x04)  
Write Single Coil (0x05)  
Write Single Register (0x06)  
Write Multiple Coils (0x0f)  
Write Multiple Registers (0x10)

MODBUS Eth. TCP/IP PLC - Simulator (port: 502)  
Connected (0/10) : (received/sent) (0/0) Serv. listening. Rx: Tx:

Address: C H D I/O Coil Outputs (0000000) Fmt: decimal +/- Prot: MODBUS TCP/IP Clone Log

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	Total
000001-000016	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0005
000017-000032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
000033-000048	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

(0x01) Single Coil  
(0) => ? = 1

(0x05) Sing Coil  
(0) <= 0 , 1

(0x0f) Multiple Write HR  
(12) <= [3] 1,1,1

Holding Register (HR) + 0  
= 40001 + 0 = 40001

(0x0f) Multiple Read HR  
(12) => [3] ?,?,? = 0,0,0

The screenshot shows the Factory IO software interface. At the top, there's a toolbar with FILE, EDIT, and VIEW tabs, along with various icons for saving, opening, and navigating. Below the toolbar is a 3D simulation of a conveyor belt system. A white bin is on the left conveyor, and an orange object is on the right conveyor. A sensor is positioned above the right conveyor. In the bottom-left corner of the simulation area, there's a red-bordered box containing a blue exclamation mark icon and the word "FORCED". On the far left, there's a vertical sidebar with icons for file operations like New, Open, Save, and Drivers.

**VIEW**

- Palette P
- Cameras I
- Camera Navigation O
- Sensors Tags
- Actuators Tags
- Show Tags Addresses
- Dock All Tags**
- Clear Docked Tags**
- Show Sensors Range
- Show Stats
- [Open Console](#)
- [Backslash](#)

**DRIVER** Modbus TCP/IP Server ✓ STOP CONFIGURATION CLEAR

**FILE** EDIT VIEW

- New Ctrl+N
- Open Ctrl+O
- Save Ctrl+S
- Save As... Ctrl+Shift+S
- Options
- Drivers F4**
- Exit

**SENSORS**

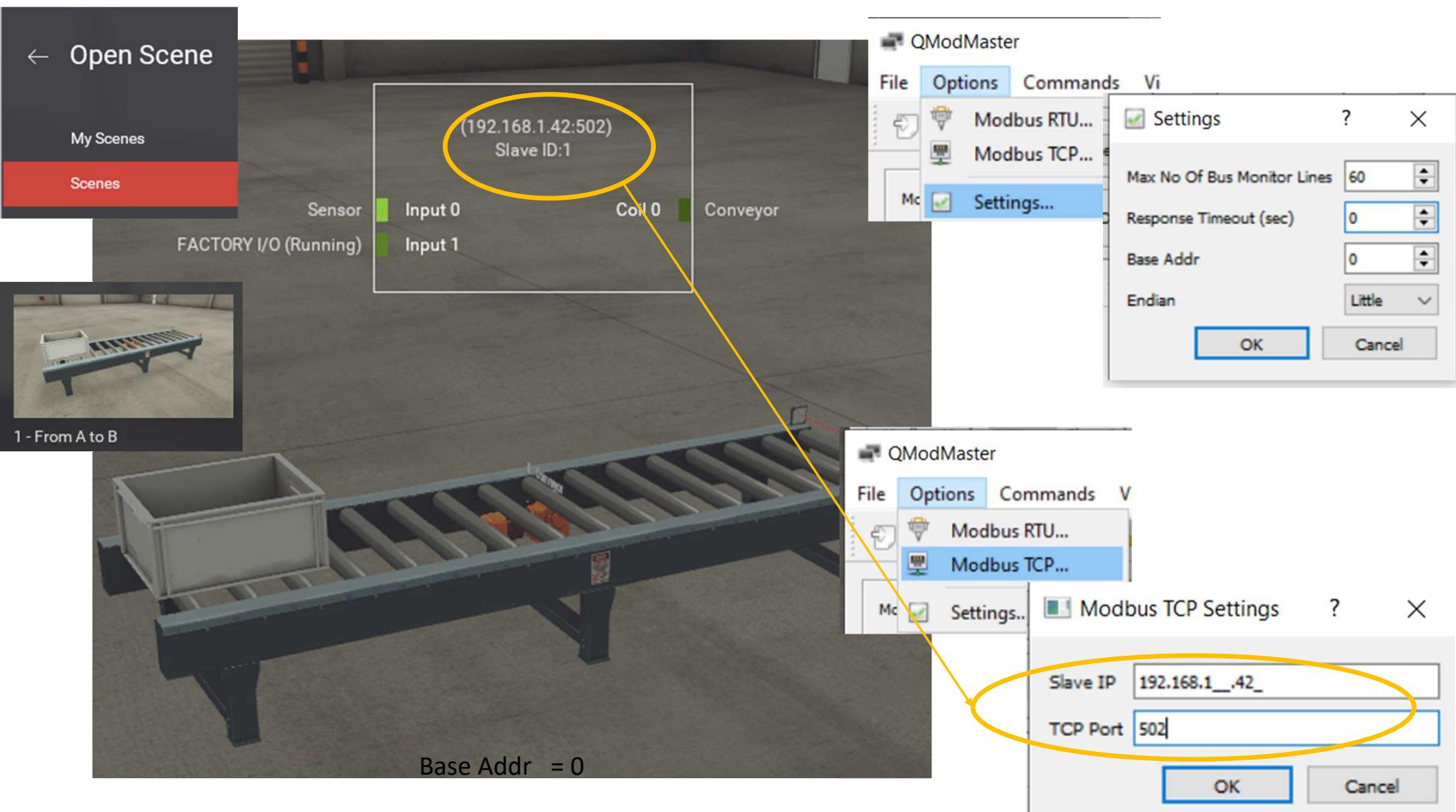
- FACTORY I/O (Paused) Sensor
- FACTORY I/O (Reset) Sensor
- FACTORY I/O (Running) Sensor
- FACTORY I/O (Time Scale) Sensor
- FACTORY I/O (Running) Sensor

(192.168.1.50:502)  
Slave ID:1

Input 0	Coil 0	Conveyor
Input 1		

**ACTUATORS**

- Conveyor
- FACTORY I/O (Camera Position)
- FACTORY I/O (Pause)
- FACTORY I/O (Reset)
- FACTORY I/O (Run)





Modbus Mode TCP Slave Addr 1 Scan Rate (ms) 6000

Function Code Read Discrete Inputs (0x02) Start Address 0

Number of Inputs 1 Data Format Dec Signed

1 **Read Sensor**  
Read Input - FC 01 (1 = on, 0 = off)  
Start Address 0 Number of Input 1

QModMaster

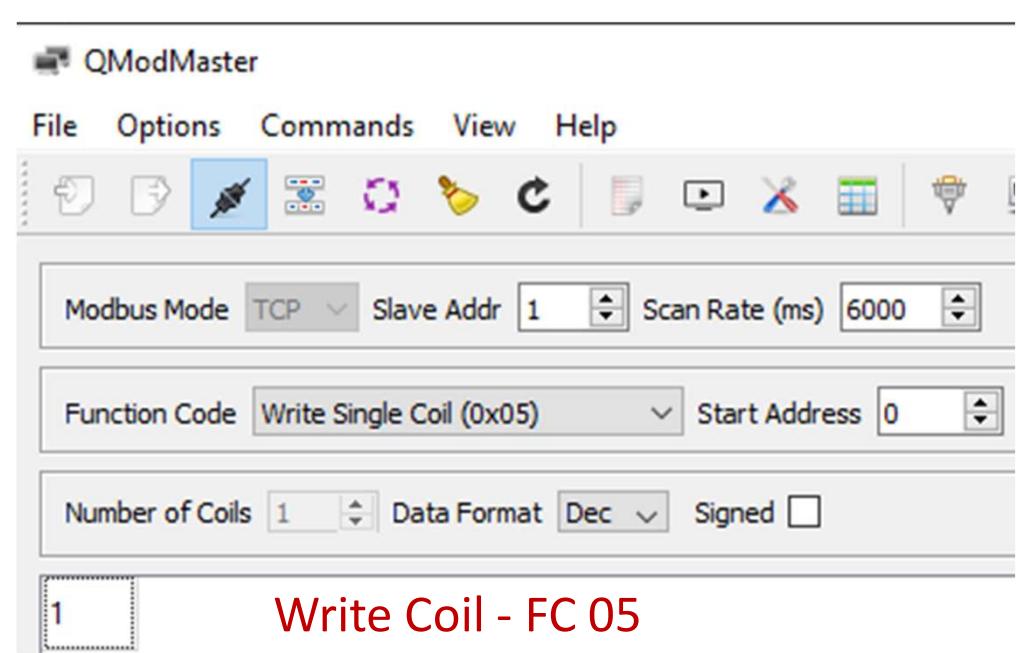
File Options Commands View Help

Modbus Mode TCP Slave Addr 1 Scan Rate (ms) 6000

Function Code Read Coils (0x01) Start Address 0 Dec

Number of Coils 1 Data Format Dec Signed

0 **Read Conveyer**  
Read Coil - FC 01 (1 = on, 0 = off)  
Start Address 0 Number of Input 1



**Write Coil - FC 05  
Start Address 0**

(1 = on, 0 = off)

Write 0 Stop , 1 Start

# Advanced Control with Machine Simulator (FactoryIO)

Factory IO

FILE EDIT VIEW

Discharge valve 6.1 FORCED X

Fill valve 7.8 FORCED X

Flow meter 2.3 X

Level meter 1.4 X

PV 0 X

Reset X

Reset light I X

Setpoint 0.0 X

SP 0 X

Start X

Start light I X

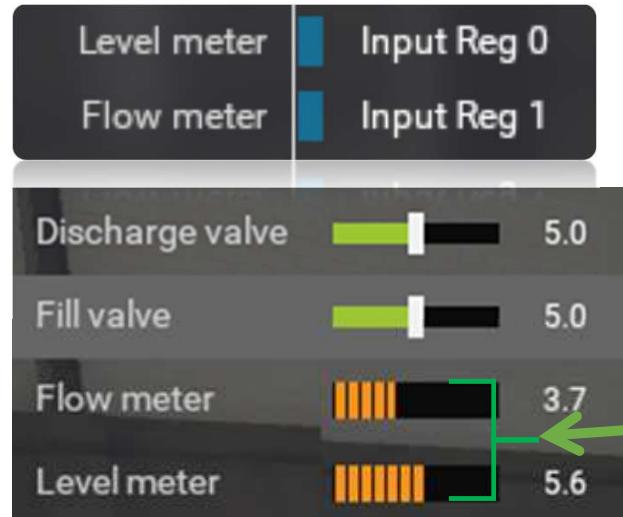
Stop X

Stop light I X

← Open Scene

My Scenes

Scenes



Modbus TCP Settings dialog box:

- Modbus Mode: TCP
- Slave Addr: 1
- Scan Rate (ms): 6000
- Function Code: Read Input Registers (0x04)
- Start Address: 0
- Number of Registers: 2
- Data Format: Dec
- Signed:

Modbus TCP Settings dialog box (Slave IP and Port):

- Slave IP: 192.168.1.50
- TCP Port: 502

**SENSORS**

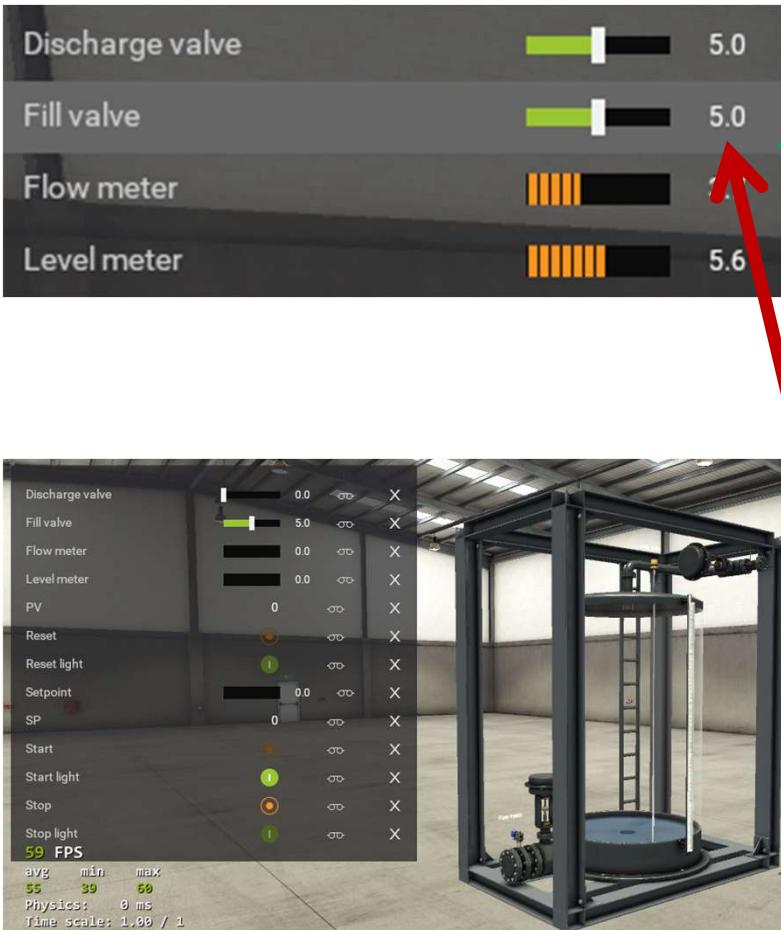
FACTORY I/O (Paused)	FACTORY I/O (Reset)	FACTORY I/O (Running)	FACTORY I/O (Time Scale)	Start	Reset	Stop	FACTORY I/O (Running)	Input 0	Coil 0	Start light
								Input 1	Coil 1	Reset light
								Input 2	Coil 2	Stop light
								Input 3	Holding Reg 0	Fill valve
								Input Reg 0	Holding Reg 1	Discharge valve
								Input Reg 1	Holding Reg 2	SP
								Input Reg 2	Holding Reg 3	PV
										SP
										Start light

Legend:

- FACTORY I/O (Camera Position)
- FACTORY I/O (Pause)
- FACTORY I/O (Reset)
- FACTORY I/O (Run)
- Fill valve
- Discharge valve
- PV
- Reset light
- SP
- Start light

Slave ID: (192.168.1.50, 502)

## FC 03 Read / FC 06 Write Single (HR) - Read Fill Valve Read Holding Reg (0x03)



Modbus Mode TCP Slave Addr 1 Scan Rate (ms) 6000

Function Code Read Holding Registers (0x03) Start Address 0

Number of Registers 1 Data Format Dec Signed

500

Holding Reg 0 Fill valve  
Holding Reg 1 Discharge valve

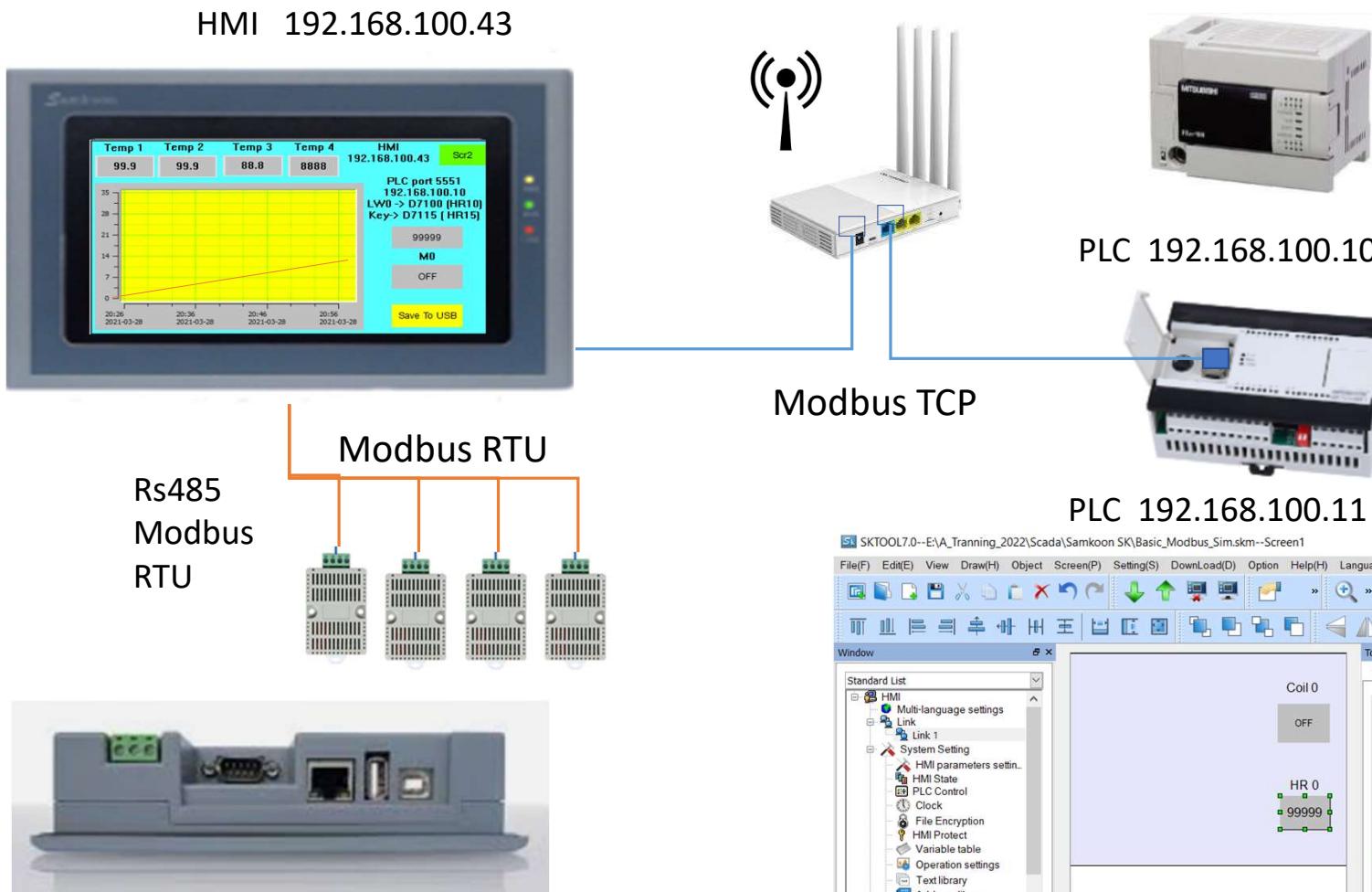
Modbus Mode TCP Slave Addr 1 Scan Rate (ms) 6000

Function Code Write Single Register (0x06) Start Address 0 Dec

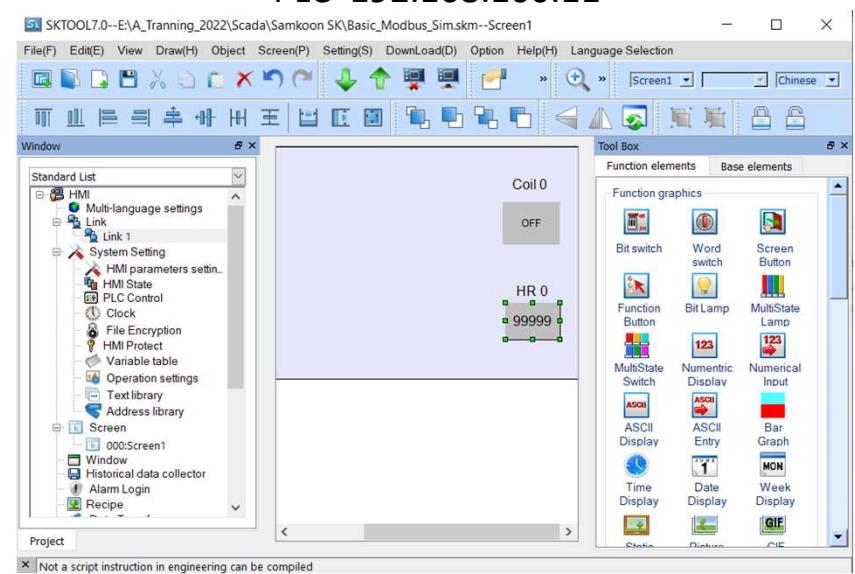
Number of Registers 1 Data Format Dec Signed

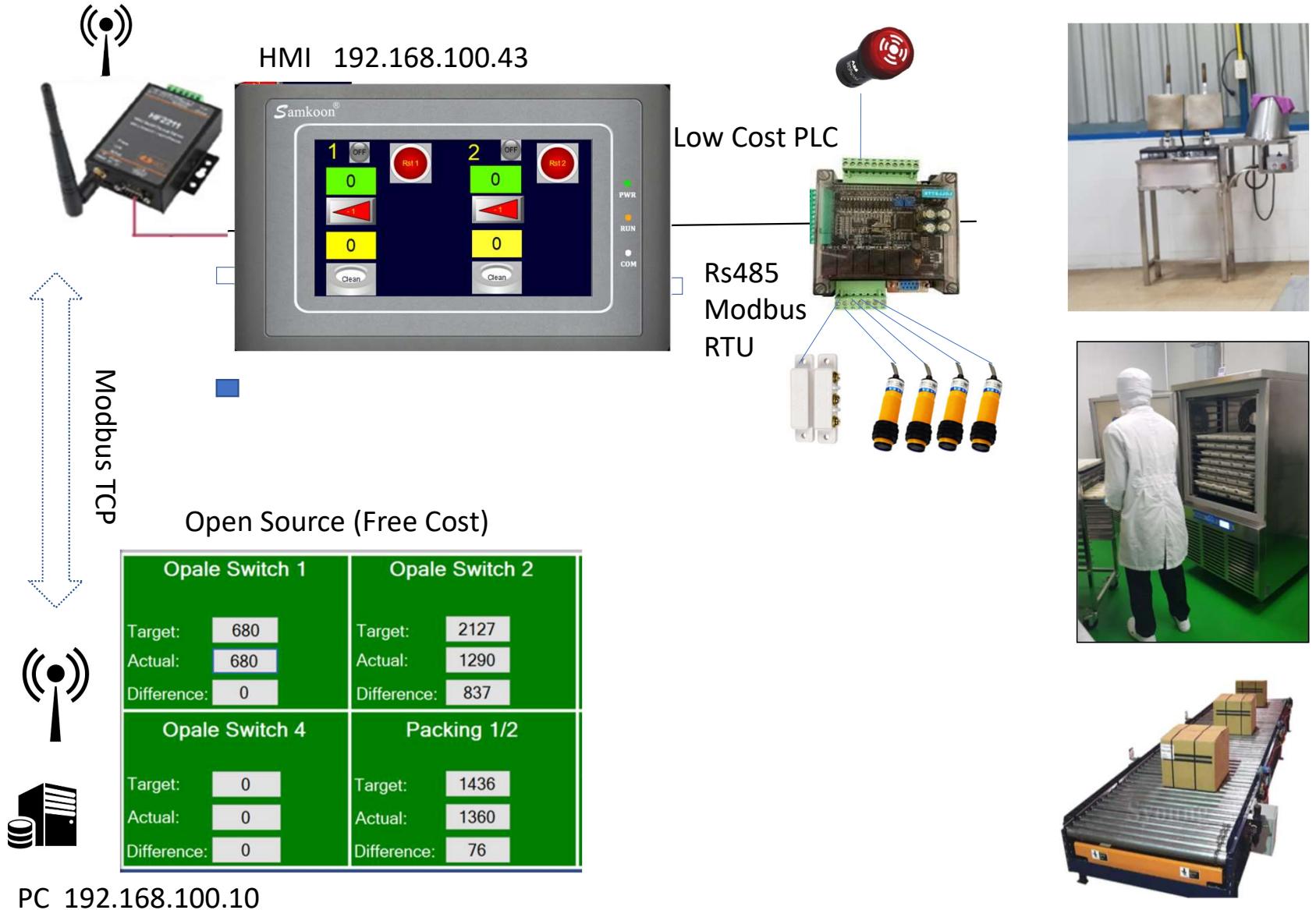
500

Control : Fill Valve Write Single Reg (0x06)



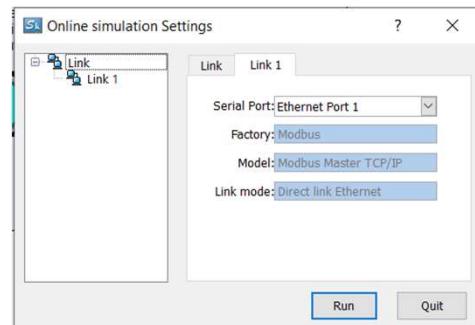
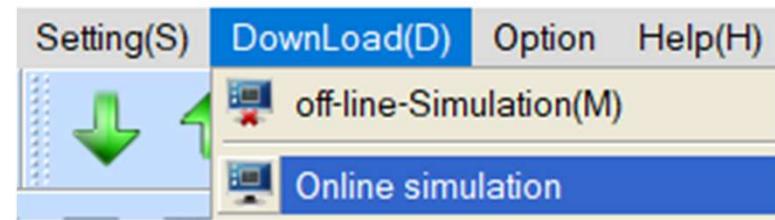
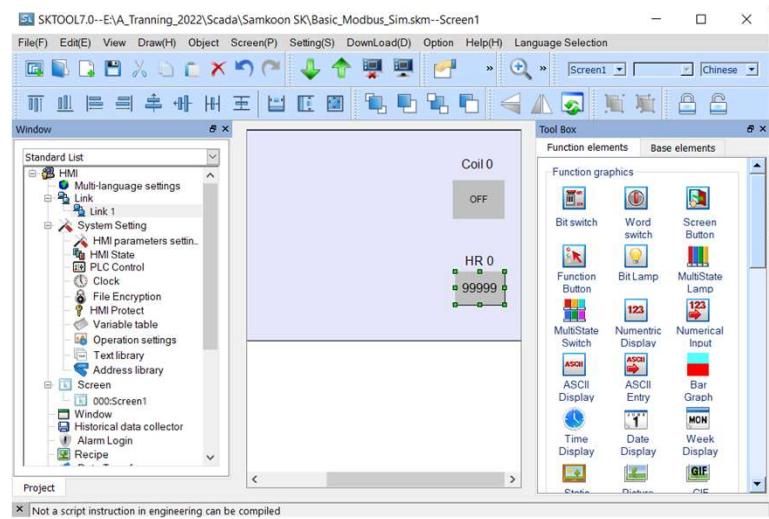
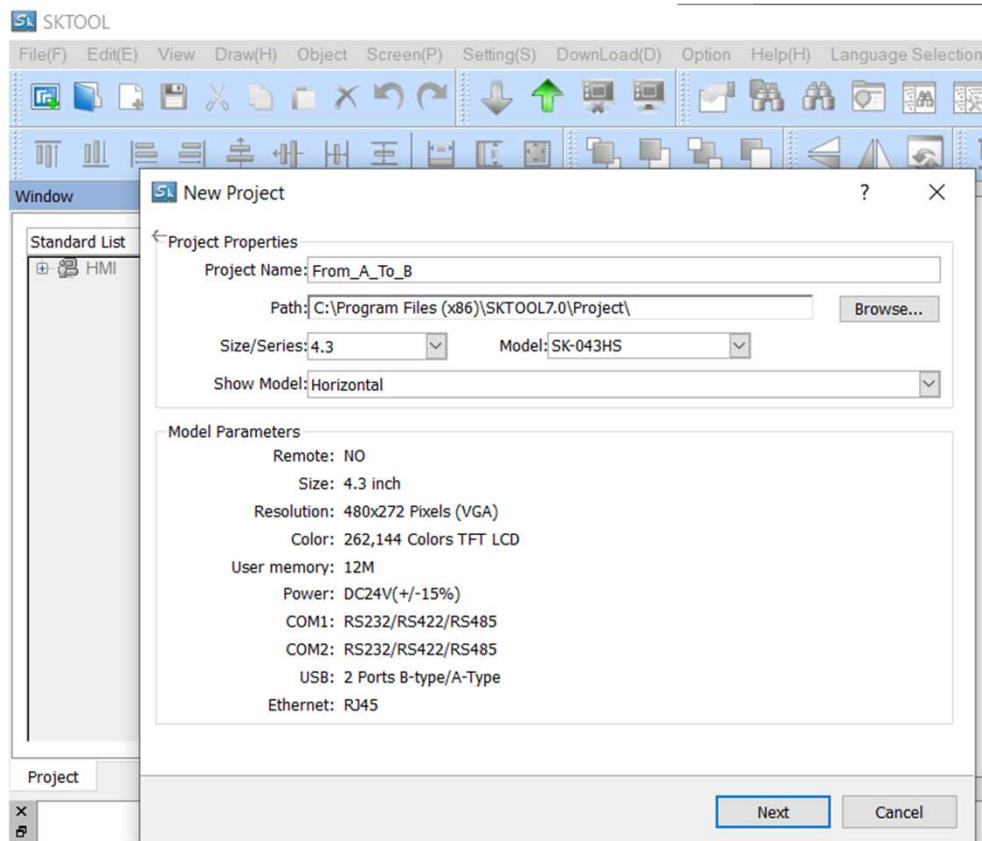
**SAMSOOK SK043HS  
SKTOOL 7**





# SAMSOOK SK043HS

## SKTOOL 7



SKTOOL7.0-EIA\_Training\_2022\Scada\Samkoon SK\Basic\_Modbus\_Sim.skf--Screen1

File(F) Edit(E) View Draw(R) Object Screen(P) Setting(S) Download(D) Option Help(H) Language Selection

Tool Box

Standard List

- HMI
  - Multi-language settings
  - Link 1
  - System Setting
    - HMI State
    - PLC Control
    - Clock
    - File Encryption
    - HMI Protect
    - Variable table
    - Operation settings
    - Text library
    - Address library
  - Screen
    - oddscreen1
    - Window
    - Historical data collector
    - Alarm Login
    - Recipe

Project

Not a script instruction in engineering can be compiled

**HMI**

- Multi-language settings
- Link
  - Link 1

**Communication Port Properties**

General Parameter

Link ID: 1

Link Name: Link 1

Link Interface: Ethernet

HMI Site: Local Setting

Connection Services: Modbus

COM port (master-slave mode) port:1

Modbus Master TCP/IP

Tool Box

Function elements Base elements

Function graphics

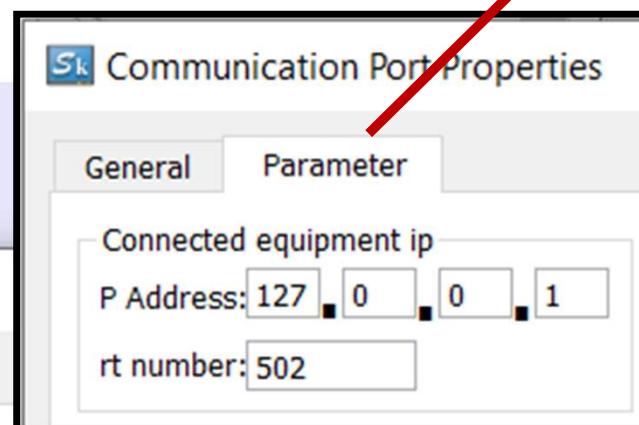
- Bit switch
- Word switch
- Screen Button
- Function Button
- Bit Lamp
- Multistate Lamp
- Multistate Switch
- Numeric Display
- Numerical Input
- ASCI Display
- ASCI Entry
- Bar Graph
- MON
- Time Display
- Date Display
- Week Display
- Display
- Clocks

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

Connected (0/10) : (received/sent) (22781/22781) Serv. lis Rx: Tx:

Address: H I/O Holding Regs (40000) Fmt: decimal +/- Prot: MODBUS TCP/IP

Address	+0	+1	+2	+3	+4
400001-400010	1000	0	0	0	0
400011-400020	0	0	0	0	0



Connection Service  
Modbus -  
Modbus Master TCP/IP

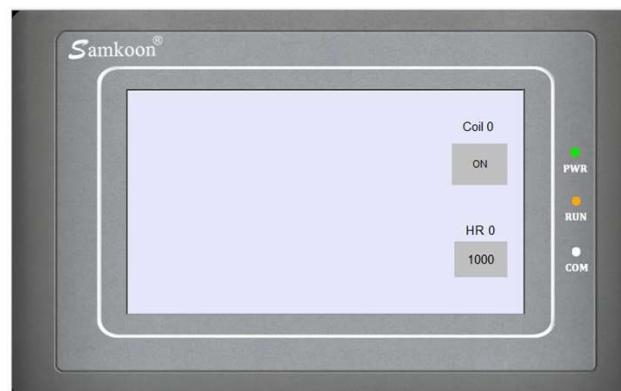
HMI Sim IP : 127.0.0.1



Bit switch

Bit Lamp

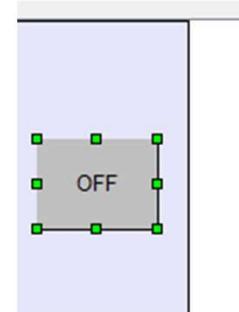
Numerical Input



**Standard**

Link 1

0x 0



**Bit switch**

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

Connected (0/10) : (received/sent) (3/3) Serv. listening

Address:  H  C I/O Coil Outputs (00000000)

Address	+0	+1	+2	+3	+4	+5
000001-000016	0	0	0	0	0	0

Address	+0	+1	+2	+3	+4	+5	+6
000001-000016	1	0	0	0	0	0	0

**Tool Box**

Function elements Base elements

**Function graphics**

- Bit switch
- Word
- Screen

**Sk Bit Button**

element type: Bit Switch

ID: BB0000

View

State: 1 0

Border Color:

FG Color: black

BG Color: light grey

Pattern: Solid

Function

Function: Invert

Mode: Press execute

Prompt

Write Address: 0x1

Monitor  Monitor Address: 0x1

Address Entry

Standard

Link 1

0x 1

Function: Invert

Invert

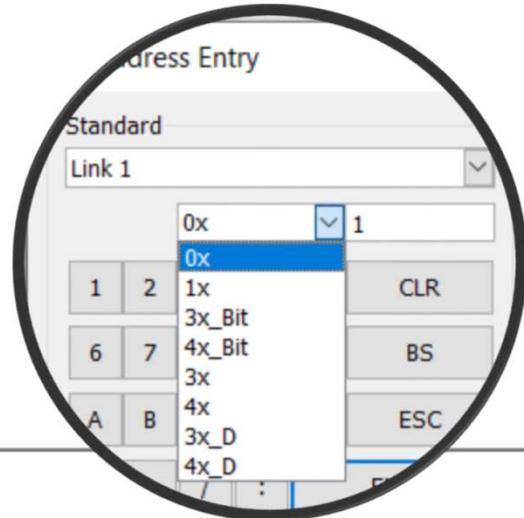
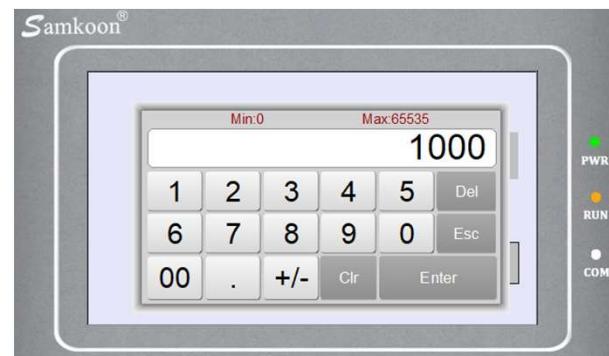
Mode: Inching

Set OFF

Set ON



## 123 Numerical Input



Standard

Link 1

4x 0

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

Connected (1/10) : (received/sent) (17282/17282) Se

Address:  H  D I/O Holding Regs (40000) ▾

Address +0 +1

400001-400010 1000 0

Sk Numeric Entry

Element type: Numeric Input

ID: NE0000

View

99999

General Picture Advanced Visibility

Shape

Border Color:

FG Color:

BG Color:

Text Color:

Pattern:  Solid

Data Type: 16-Bit Unsigned Int   Unit

Display Type: 16-Bit Unsigned Decimal

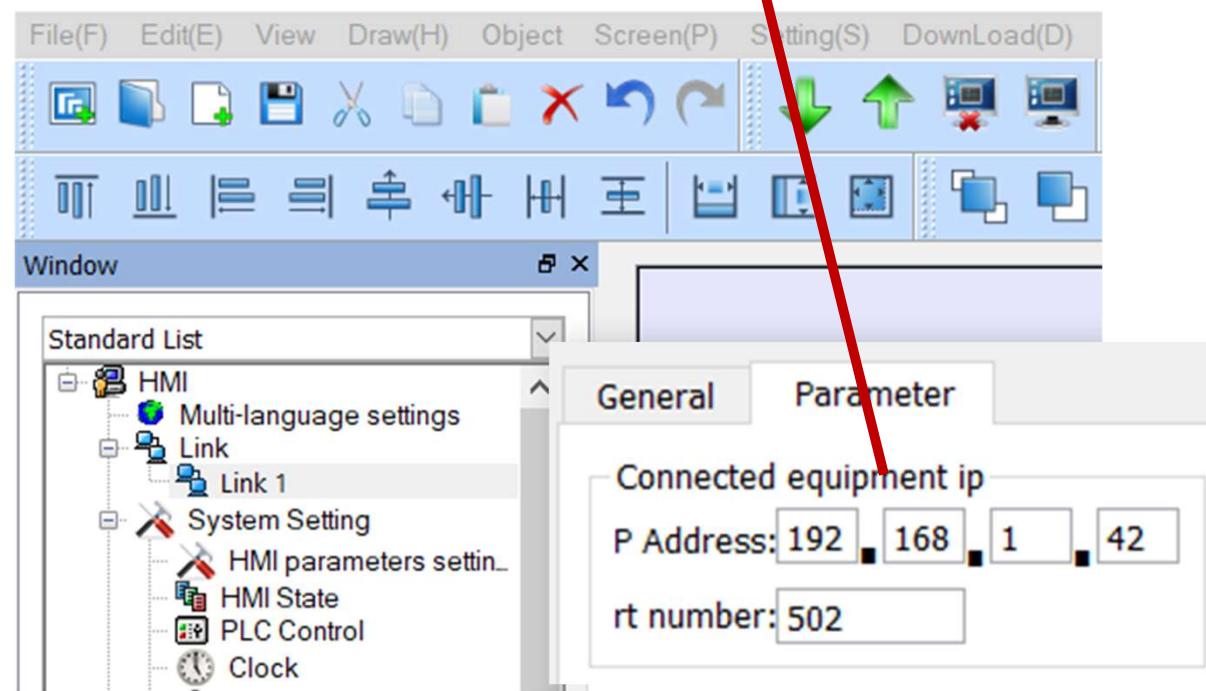
Write Address: 4x0   Offset

## DRIVER

Modbus TCP/IP Server



SK SKTOOL7.0--E:\A\_Tranning\_2022\Scada\Samkoon SK\Basic\_Modbus\_Sim.ski





Conveyer  
Read Coil 0  
Coil Outputs 0

**Bit switch**

**Sk Bit Button**

element type  
**Bit Switch**

ID: **BB0000**

**View**

OFF

**General** **Appearance** **Advanced**

State: **1** **0**

Border Color:

FG Color: **Black**

BG Color:

Pattern:  Solid

**Function**

Function: **Invert**

Mode: Press execute

**Prompt**

Write Address: **0x0**

**Address Entry**

Standard

Link 1

0x **0**

Function: ON / OFF status monitoring



**Bit Lamp**

**Sk Bit Lamp**

Element type  
**Bit Lamp**

ID: **BL0000**

**View**

OFF

**General** **Appearance** **Visibility**

Shape

State: **1** **0**

Border Color:

FG Color: **Black**

BG Color:

Pattern:  Solid

Data Type: **Bit**

Monitor Address: **1x0**  Offset

**Address Entry**

Standard

Link 1

1x **0**

**Prompt**

Function:ON / OFF monitoring, real-time monitoring user settings changing register values, Dynamics reactions

**FACTORY I/O (Running)**

Sensor	Input 0	<b>1x</b>	Input 1	0x	Coil 0	Conveyor
--------	---------	-----------	---------	----	--------	----------

**Bit Button**

element type: Bit Switch  
ID: BB0000

General Appearance Advanced Visibility

Language: Chinese  All Texts use the first language  
 Use text lib Text lib

Status 0 Text	Status 1 Text	Status 0 Picture	Status 1 Picture
---------------	---------------	------------------	------------------

Picture Source:  Systems Library  File

Select Picture 

Fit to Object Size

  
Prompt

**Bit Lamp**

Element type: Bit Lamp  
ID: BL0000

General Appearance Visibility

Language: Chinese  All Texts use the first language  
 Use text lib Text lib

Status 0 Text	Status 1 Text	Status 0 Picture	Status 1 Picture
---------------	---------------	------------------	------------------

Picture Source:  Systems Library  File

Select Picture 

Fit to Object Size

  
Prompt

Function: ON / OFF monitoring, real-time

**Bit switch**



Status 0 Text Status 1 Text Status 0 Picture Status 1 Picture

Picture Source:  Systems Library  File

Select Picture 

Fit to Object Size



**Bit Lamp**

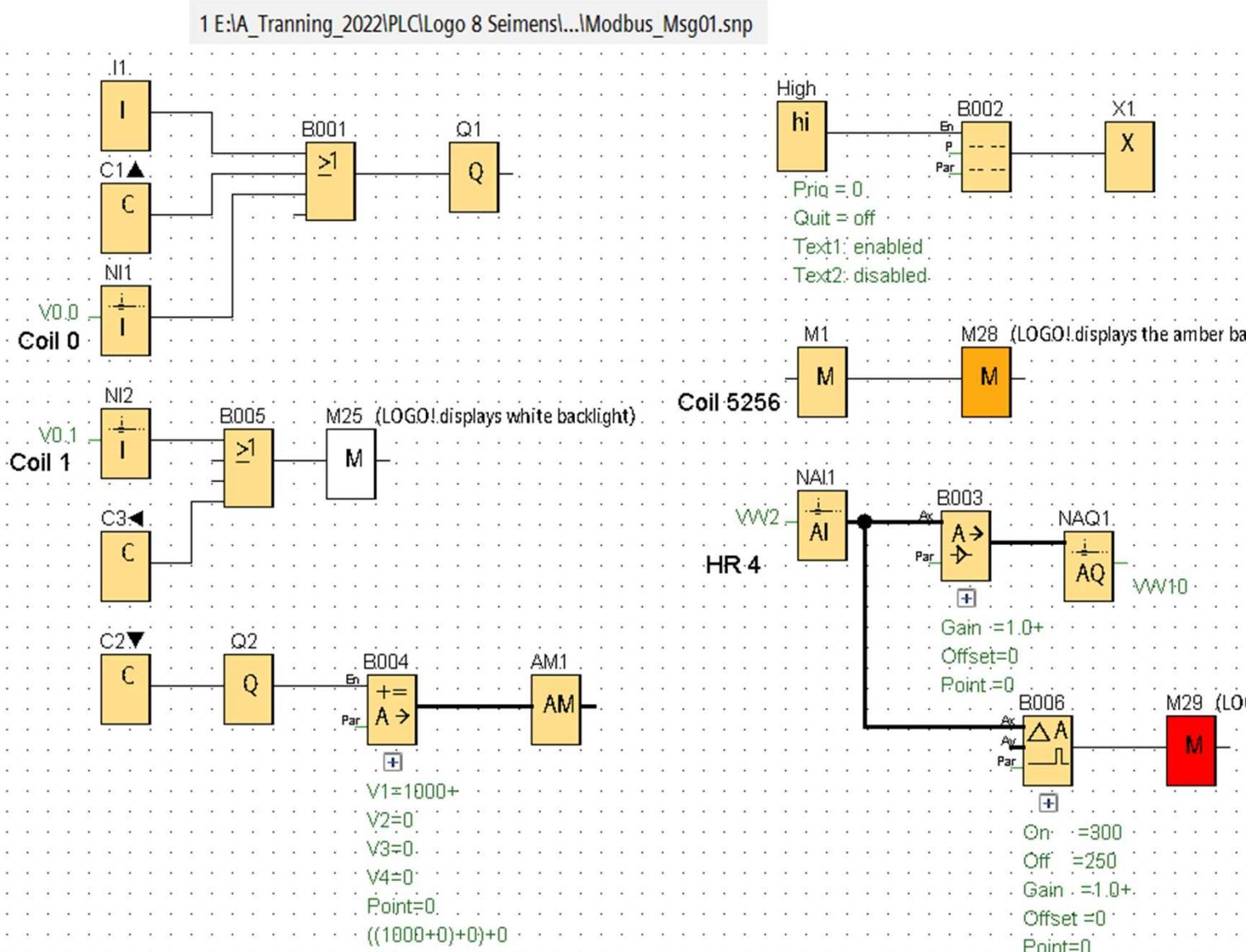


SENSEOR  
Read Discrete Input 1  
Digital Inputs 1

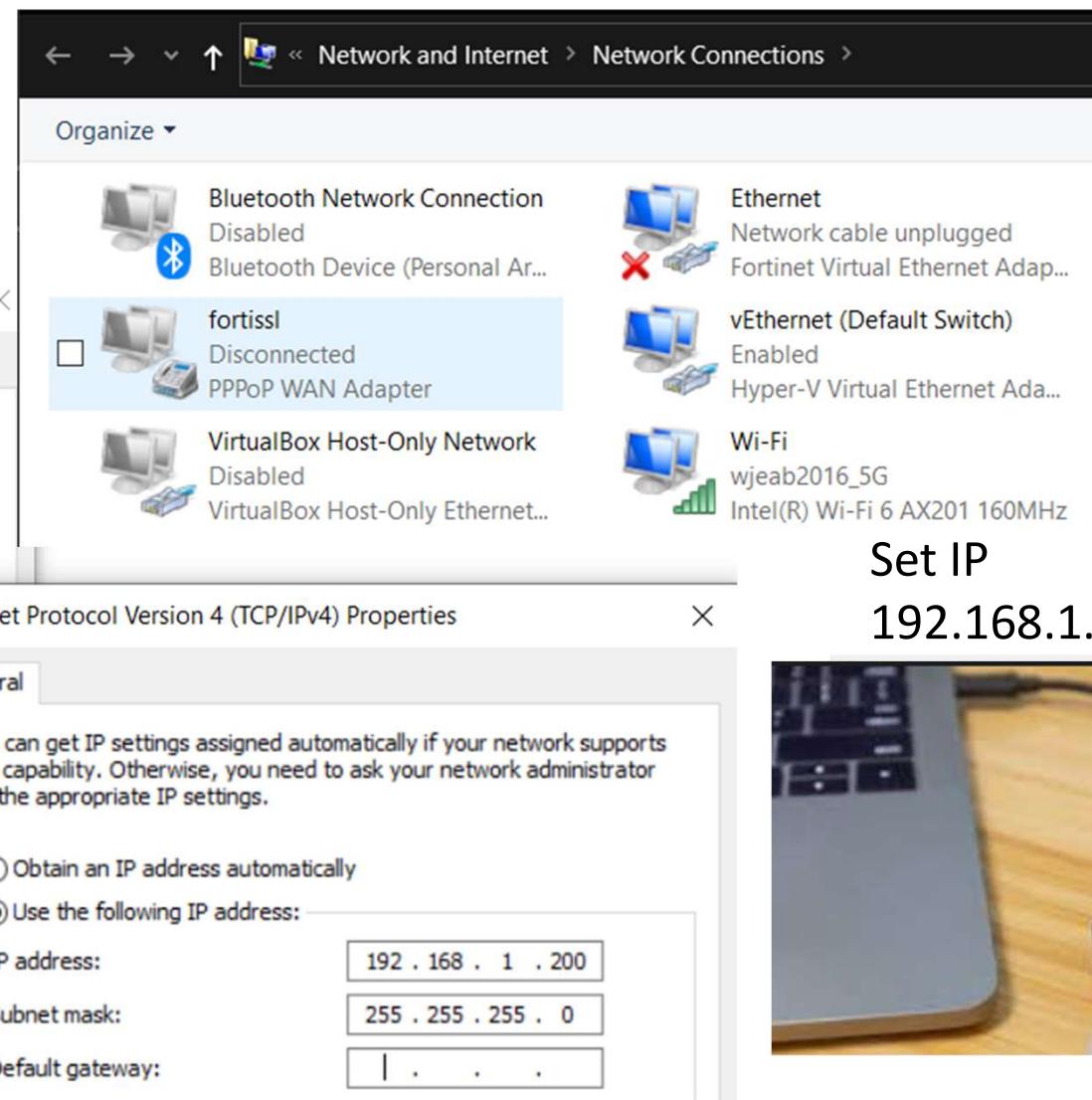
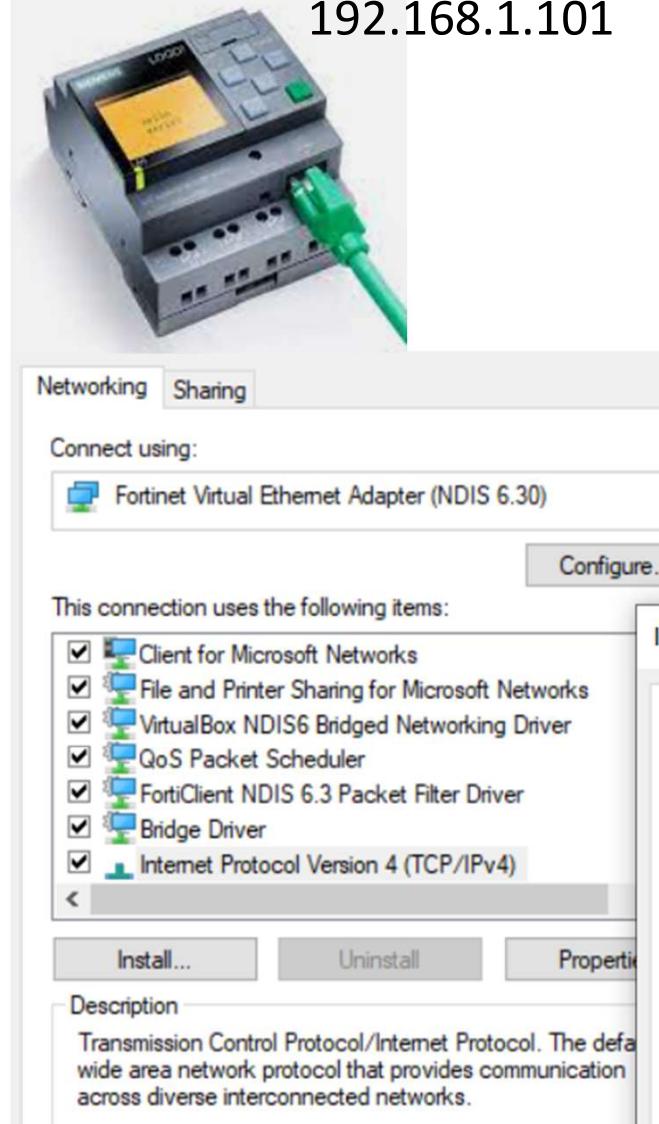
**Conveyer**

**Read Coil 0**

**Coil Outputs 0**



192.168.1.101



Set IP  
192.168.1.200

## Modbus address space

Address Type	Range	Mapped Modbus Address	Direction	Unit	
I	1 - 24	Discrete Input (DI) 1 - 24	R	bit	^
Q	1 - 20	Coil 8193 - 8212	R/W	bit	
M	1 - 64	Coil 8257 - 8320	R/W	bit	
V	0.0 - 850.7	Coil 1 - 6808	R/W	bit	
AI	1 - 8	Input Register (IR) 1 - 8	R	word	
W	0 - 850	Holding Register (HR) 1 - 425	R/W	word	
AQ	1 - 8	Holding Register (HR) 513 - 520	R/W	word	▼

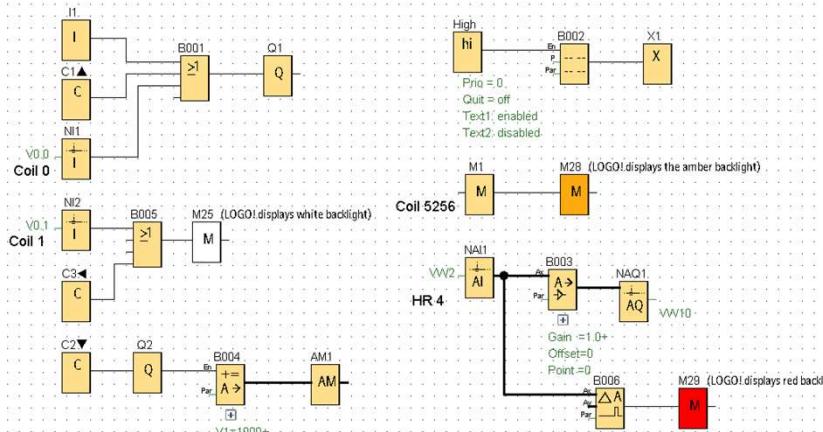
Offset -1

Q1 = Coil (8193-1) = Coil 8192  
M1 = Coil (8257-1) = Coil 8256



Offset /2

VM1 = HR0  
VM2 = HR1  
VM4 = HR2



Modbus TCP Settings

Modbus Mode	TCP	Slave Addr	1	Scan Rate (ms)	3000
Function Code	Write Single Coil (0x05)	Start Address	1		
Slave IP	192.168.1.101				
TCP Port	502				
Number of Coils	1	Data Format	Bin		

Modbus Mode

TCP	Slave Addr	1	Scan Rate (ms)	3000	
Function Code	Read Coils (0x01)	Start Address	0		
Number of Coils	8	Data Format	Bin		
0 1 0 0 0 0 0 0 x x					

Offset -1  
Q1 = Coil 0

Modbus Mode

TCP	Slave Addr	1	Scan Rate (ms)	3000	
Function Code	Write Single Coil (0x05)	Start Address	1		
Number of Coils	1	Data Format	Bin		
1					

Modbus Mode

TCP	Slave Addr	1	Scan Rate (ms)	3000	
Function Code	Read Holding Registers (0x03)	Start Address	0		
Number of Registers	8	Data Format	Dec	Signed	<input type="checkbox"/>
512 333 0 0 0 333 0 0 x x					

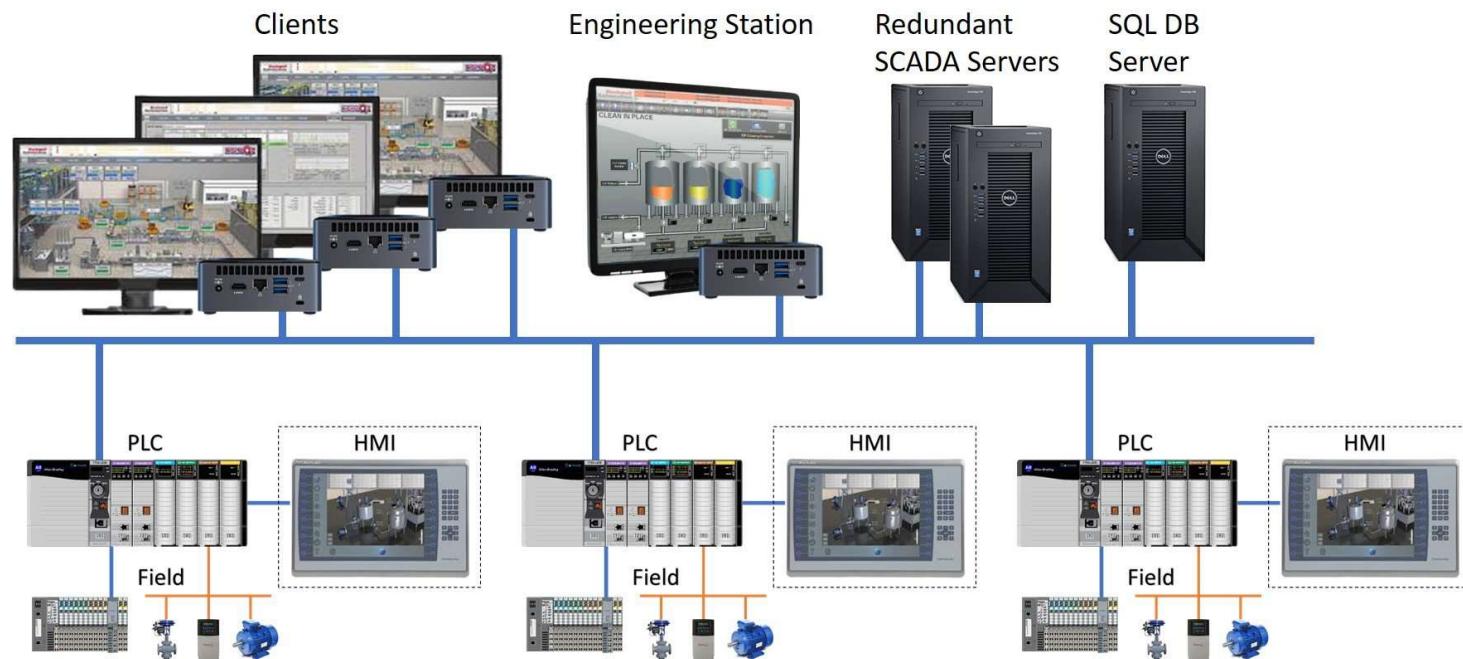
Offset /2  
VM2 =HR1  
VM4 =HR2

Modbus Mode

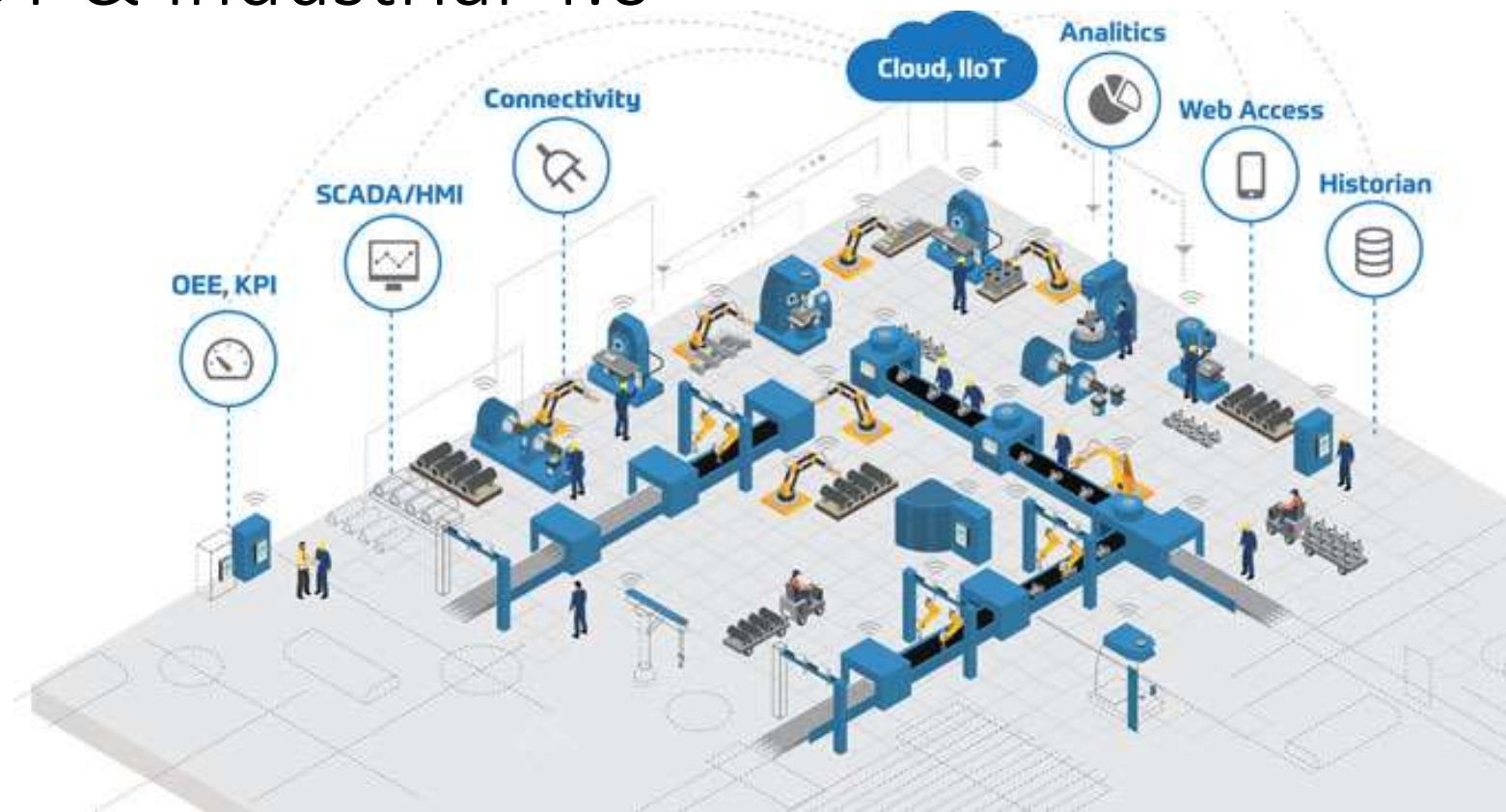
TCP	Slave Addr	1	Scan Rate (ms)	3000	
Function Code	Write Single Register (0x06)	Start Address	1		
Number of Registers	1	Data Format	Dec	Signed	<input type="checkbox"/>
333					

# WHAT IS SCADA

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



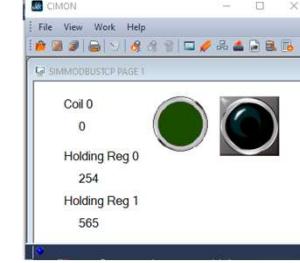
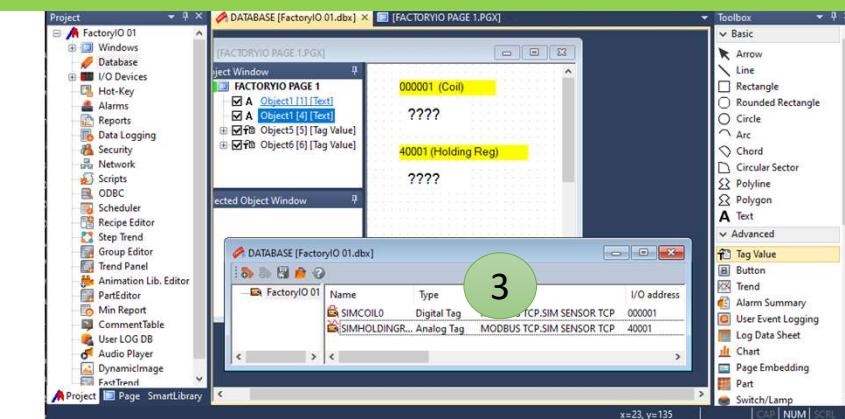
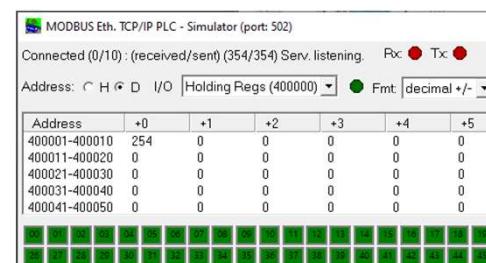
# IIoT & Industrial 4.0

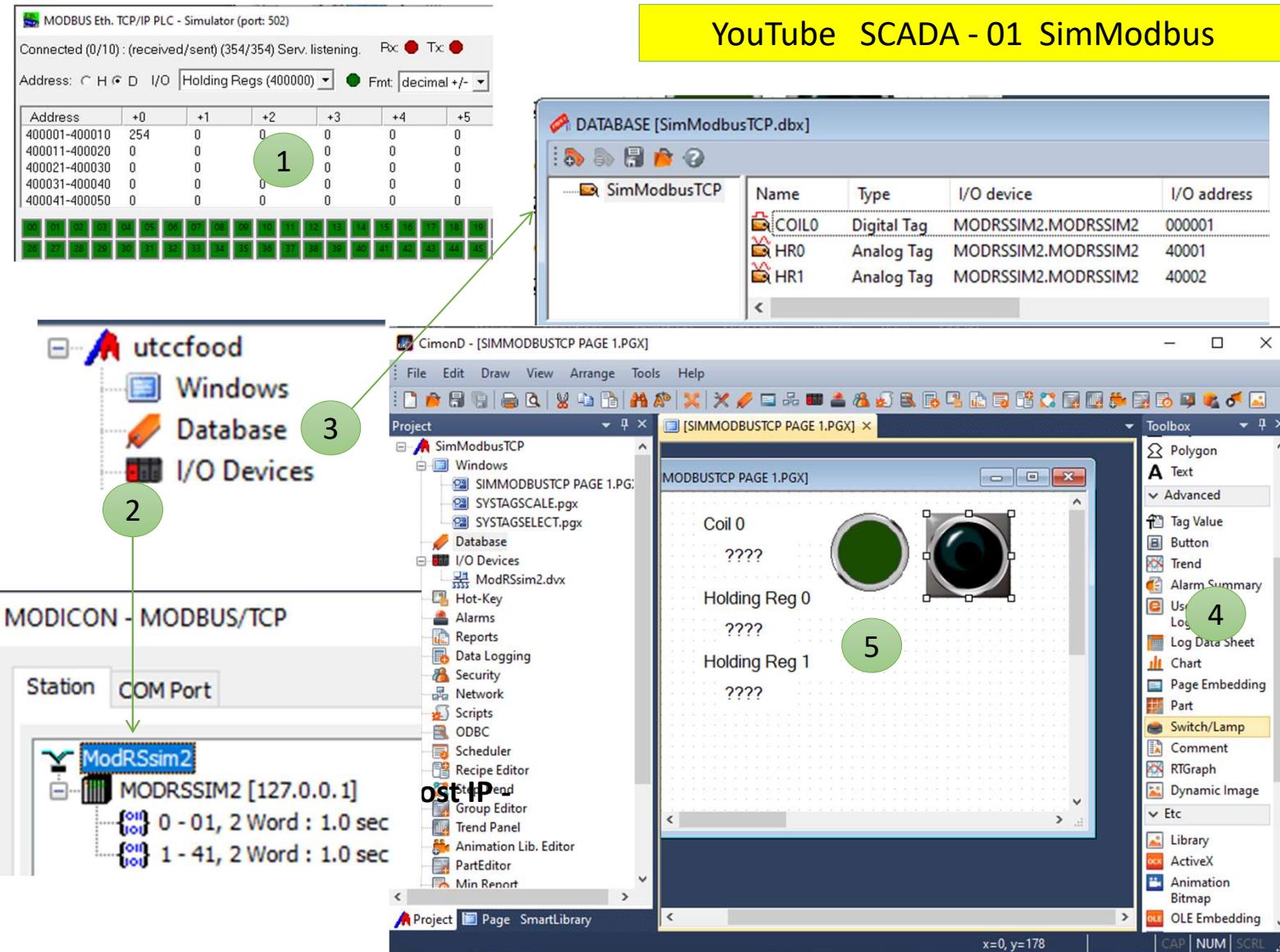


<https://www.tdsthailand.com/industrial/detail.php?i=12>

# Cimon SCADA

- Cimco utccfood





**CIMON**

SIMMODBUSTCP PAGE 1

Coil 0  
0

Holding Reg 0  
254

Holding Reg 1  
565

**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) (168/168) Serv. write data 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	0	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) (248/248) Serv. write data

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

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

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40

**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



← DRIVER

Modbus TCP/IP Server



MODICON

## SENSORS

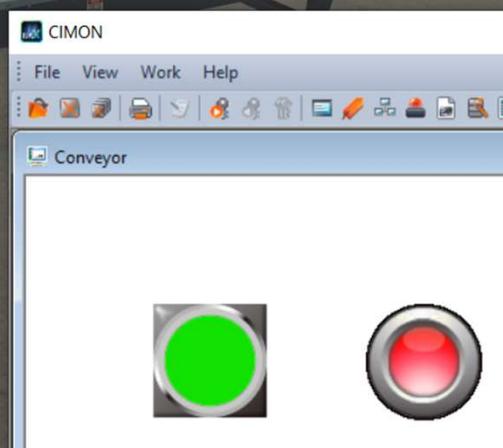
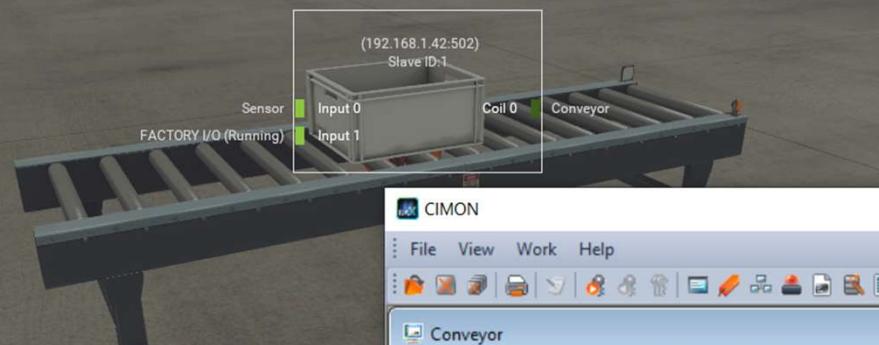
FACTORY I/O (Paused)

FACTORY I/O (Reset)

FACTORY I/O (Running)

FACTORY I/O (Time Scale)

Sensor



DATABASE [t.dbx]						
	Name	Type	I/O device	I/O addr...	Initial...	Add...
...	SENSOR	Digital Tag	FACTORYI...	100001	0	
...	CONVEYOR	Digital Tag	FACTORYI...	000001	0	

MODICON Modbus TCP(Block)

## Modbus Ethernet Setting

Station

Communication Port

FactoryID

CONVEYOR [192.168.1.42] :

- [IO] 00001, 1 DataNo : Scan 1000 msec
- [IO] 10001, 1 DataNo : Scan 1000 msec

## ComBlock Setting

Addr Type Coil Status

Start Addr 1

Data Num 1

Period 1000 msec

## ComBlock Setting

Addr Type Input Status

Start Addr 1

Data Num 1

Period 1000 msec

## I/O Devices

I/O Device Configuration

New Device Edit

MODICON

Factory IO

I/O Device Selection

Device Name	Description
FactoryIO	LSIS XGK/XGB PLC Ethernet
LSIS	LSIS XGI/XGR/XEC PLC Serial
MODICON	MODICON Modbus TCP(Block)
<b>MODICON</b>	<b>MODICON Modbus TCP</b>
MODICON	MODICON Modbus RTU Mode
MODICON	MODICON Modbus TCP Slave
MODICON	Modbus Cdma
OPC	OPC Client DA 3.0

OK

FACTORY I/O (Running)

Sensor Input 0 Input 1 Coil 0 Conveyor

(192.168.1.42:502) Slave ID:1

Modbus Ethernet Setting

Station Communication Port

FactoryID CONVEYOR [192.168.1.42] :

- 00001, 1 DataNo : Scan 1000 msec
- 10001, 1 DataNo : Scan 1000 msec

1. Add COM  
2. Add COM

ComBlock Add

Modbus Ethernet Setting

Station Add

FactoryID

Station Add

Save Exit

MODICON Modbus TCP(Block)

Station Setting

Name	CONVEYOR
IP Address	192.168.1.42
Port	502
Device ID	1
Protocol Type	<input type="radio"/> TCP <input checked="" type="radio"/> UDP
Recv Time Out	3000 msec

ComBlock Setting

1. Add COM (Coil Status)

Addr Type	Coil Status
Start Addr	1
Data Num	1
Period	1000 msec

2. Add COM (Input Status)

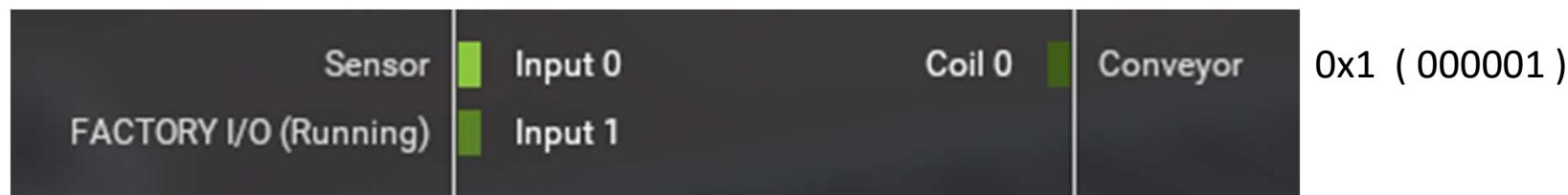
Addr Type	Input Status
Start Addr	1
Data Num	1
Period	1000 msec

## Database

DATABASE [t.dbx]

	Name	Type	I/O device	I/O addr...	Initial...	Add
...	SENSOR	Digital Tag	FACTORYI...	100001	0	
...	CONVEYOR	Digital Tag	FACTORYI...	000001	0	

1x1 ( 100001 )  
1x2 ( 100002 )



Edit Tag

General Advanced

Name: SENSOR

Type: Digital

Real Tag Virtual Tag

I/O Device: FACTORYID.CONVEYOR

I/O Address: 000001

Switch/Lamp

Edit Tag

General Advanced

Name: CONVEYOR

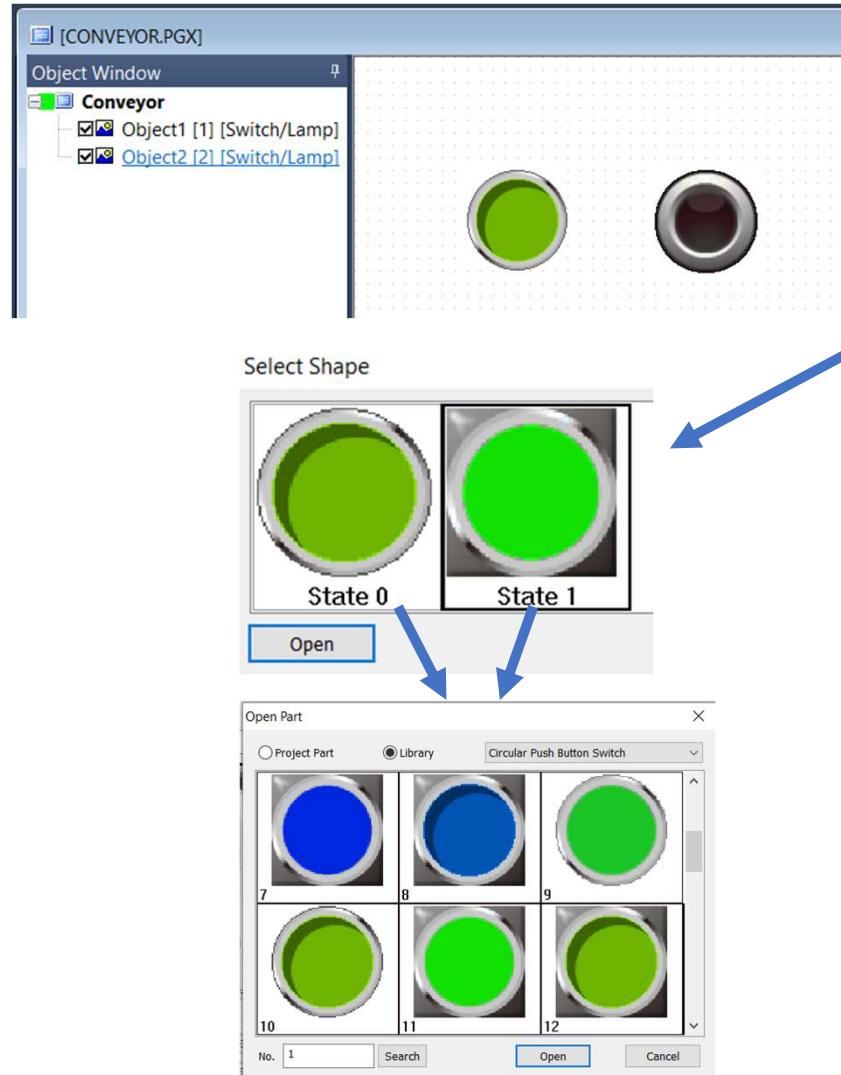
Type: Digital

Real Tag Virtual Tag

I/O Device: FACTORYID.CONVEYOR

I/O Address: 100001

# Add New Page



Switch/Lamp

Name: Object1

Switch Feature

No. 1 Action: Bit Momentary CONVEYOR

Bit Switch

Tag: CONVEYOR

Write Value:
  Set
  Reset
  Toggle
  Momentary

Set Lamp

OK Cancel

Bit Switch

Word Switch

Page Switch

Command Switch

Edit Remove

Switch/Lamp

Name: Object2

Switch Feature

No. 1 Action: Bit Set SENSOR

Bit Switch

Tag: SENSOR

Write Value:
  Set
  Reset
  Toggle

Set Lamp

OK Cancel

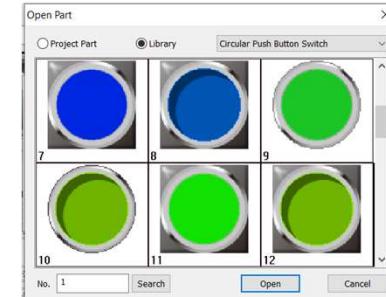
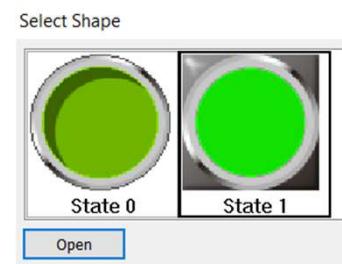
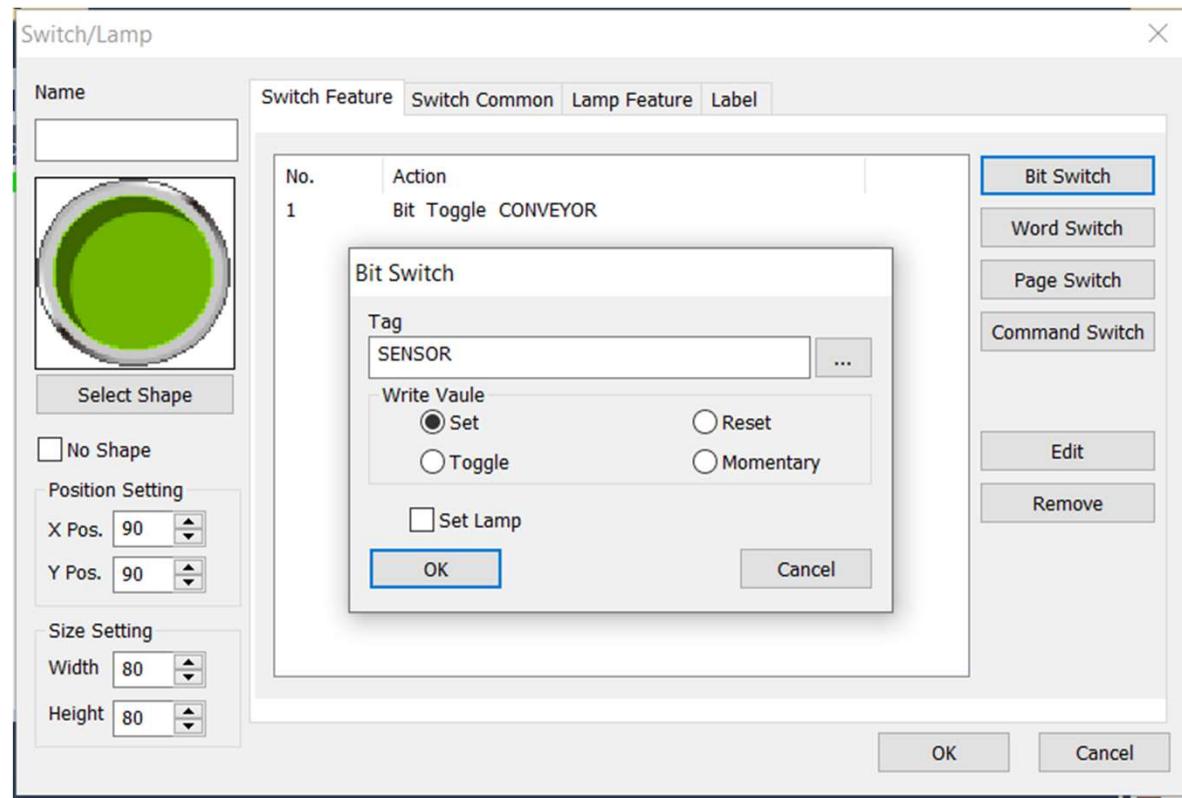
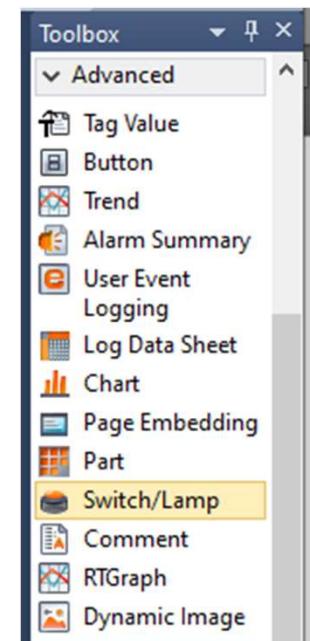
Bit Switch

Word Switch

Page Switch

Command Switch

Edit Remove



Factory IO

FILE EDIT VIEW

	Value	Unit	Control
Discharge valve	1	5.0	X
Fill valve	0	2.0	X
Flow meter	1	4.3	X
Level meter	0	7.5	X
PV	1	0	X
Reset	2		X
Reset light	2	!	X
Setpoint	2	0.0	X
SP	0	0	X
Start	0		X
Start light	0	!	X
Stop	1	!	X
Stop light	1	!	X

CIMON

File View Work Help

LevelControl

Parameter	Value	Unit
Level Meter	766	
Fill Value	200	200
Flow Meter	437	
Discharge Value	500	500

DRIVER

Modbus TCP/IP Server

(192.168.1.42:502)  
Slave ID:1

Action	Input	Output
Start	Input 0	Coil 0
Reset	Input 1	Coil 1
Stop	Input 2	Coil 2
FACTORY I/O (Running)	Input 3	Holding Reg 0
Level meter	Input Reg 0	Holding Reg 1
Flow meter	Input Reg 1	Holding Reg 2
Setpoint	Input Reg 2	Holding Reg 3

Start light  
Reset light  
Stop light  
Fill valve  
Discharge valve  
SP  
PV

Level Control

I/O Device Configuration

New Device Edit

I/O Device Selection

Device Name	FactoryIO
Manufacturer	Description
LSIS	LSIS XGK/XGB PLC Ethernet
LSIS	LSIS XG1/XGR/XEC PLC Serial
MODICON	MODICON Modbus TCP(Block)
MODICON	MODICON Modbus TCP
MODICON	MODICON Modbus RTU Mode
MODICON	MODICON Modbus TCP Slave
MODICON	Modbus Cdma
OPC	OPC Client DA 3.0

Ok

Station Setting

Name	LEVELCONTROL
IP Address	192.168.1.42
Port	502
Device ID	1
Protocol Type	<input checked="" type="radio"/> TCP <input type="radio"/> UDP
Recv Time Out	3000 msec

## DRIVER

Modbus TCP/IP Server

	(192.168.1.42:502) Slave ID:1	
Start	Input 0	Coil 0
Reset	Input 1	Coil 1
Stop	Input 2	Coil 2
FACTORY I/O (Running)	Input 3	Holding Reg 0
Level meter	Input Reg 0	Holding Reg 1
Flow meter	Input Reg 1	Holding Reg 2
Setpoint	Input Reg 2	Holding Reg 3
		Start light Reset light Stop light Fill valve Discharge valve SP PV

## I/O Devices



Level Control

I/O Device Configuration

New Device Edit

I/O Device Selection

Device Name	FactoryIO
Manufacturer	Description
LSIS	LSIS XGK/XGB PLC Ethernet
LSIS	LSIS XG1/XGR/XEC PLC Serial
MODICON	MODICON Modbus TCP(Block)
MODICON	MODICON Modbus TCP
MODICON	MODICON Modbus RTU Mode
MODICON	MODICON Modbus TCP Slave
MODICON	Modbus Cdma
OPC	OPC Client DA 3.0

OK

MODICON

MODICON Modbus TCP(Block)

## Modbus Ethernet Setting

Station Communication Port

- FactoryIO
- LEVELCONTROL [192.168.1.42] :
  - 30001, 3 DataNo : Scan 1000 msec
  - 40001, 4 DataNo : Scan 1000 msec

Modify

Delete

Station Add

ComBlock Add

### Station Setting

Name: LEVELCONTROL  
IP Address: 192.168.1.42  
Port: 502  
Device ID: 1  
Protocol Type:  TCP  UDP  
Recv Time Out: 3000 msec

### ComBlock Setting

Addr Type: Input Register  
Start Addr: 1  
Data Num: 3  
Period: 1000 msec

### ComBlock Setting

Addr Type: Holding Register  
Start Addr: 1  
Data Num: 4  
Period: 1000 msec

FACTORY I/O (Running)	<b>Input 3</b>	Holding Reg 0	<b>Fill valve</b>
Level meter	<b>Input Reg 0</b>	Holding Reg 1	<b>Discharge valve</b>
Flow meter	<b>Input Reg 1</b>	Holding Reg 2	<b>SP</b>
Setpoint	<b>Input Reg 2</b>	Holding Reg 3	<b>PV</b>

3x1 ( 30001 )

3x2 ( 30002 )

4x1 ( 40001 )

4x2 ( 40002 )

DATABASE [LevelControl.dbx]				
	Name	Type	I/O device	I/O address
LevelCon	LEVEL_METER	Analog Tag	FACTORYI...	30001
	FLOW_METER	Analog Tag	FACTORYI...	30002
	FILL_VALUE	Analog Tag	FACTORYI...	40001
	DISCHARGE	Analog Tag	FACTORYI...	40002

Edit Tag

Group

Name **LEVEL\_METER**

Type

- Group
- Digital
- Analog
- String

General Advanced

Real Tag  Virtual Tag

I/O Device FACTORYIO.LEVELCONT

I/O Address 30001

Edit Tag

Group

Name **FILL\_VALUE**

Type

- Group
- Digital
- Analog
- String

General Advanced

Real Tag  Virtual Tag

I/O Device FACTORYIO.LEVELCONT

I/O Address 40001

Edit Tag

Group

Name **FLOW\_METER**

Type

- Group
- Digital
- Analog
- String

General Advanced

Real Tag  Virtual Tag

I/O Device FACTORYIO.LEVELCONT

I/O Address 30002

Edit Tag

Group

Name **DISCHARGE**

Type

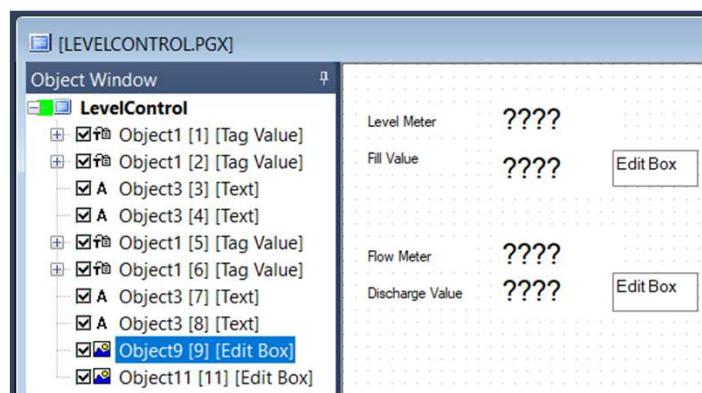
- Group
- Digital
- Analog
- String

General Advanced

Real Tag  Virtual Tag

I/O Device FACTORYIO.LEVELCONT

I/O Address 40002



## Tag Value

????

Tag Value

Tag name

Preview

Display Format

Ok Cancel

A Text

Level Meter

????

Fill Value

????

Edit Box

Flow Meter

????

Discharge Value

????

Edit Box

## Select Tag

Select

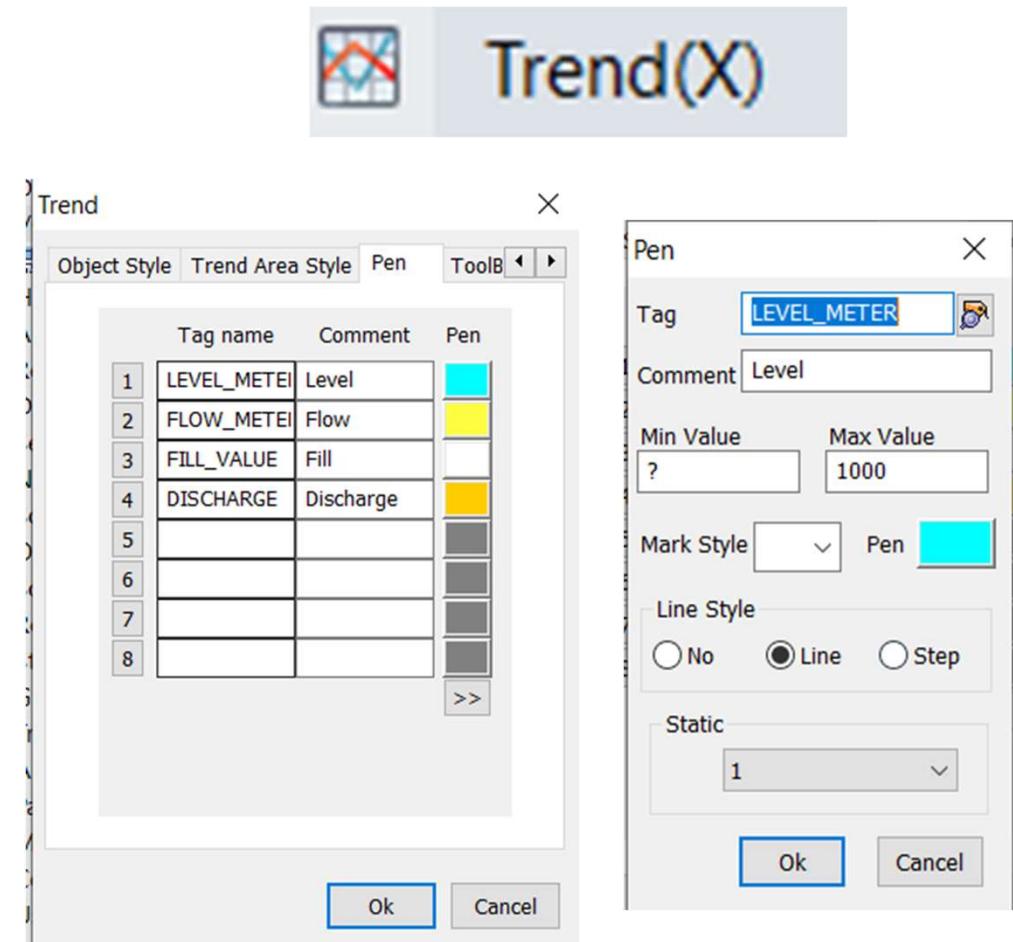
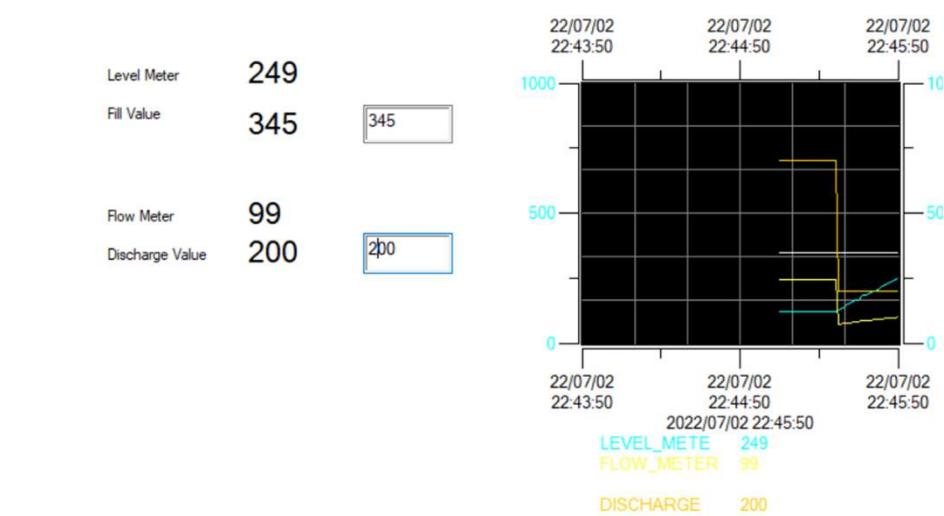
Database

LevelControl

Name	Type
LEVEL_METER	Analog Tag
FLOW_METER	Analog Tag
FILL_VALUE	Analog Tag
DISCHARGE	Analog Tag



Edit Control



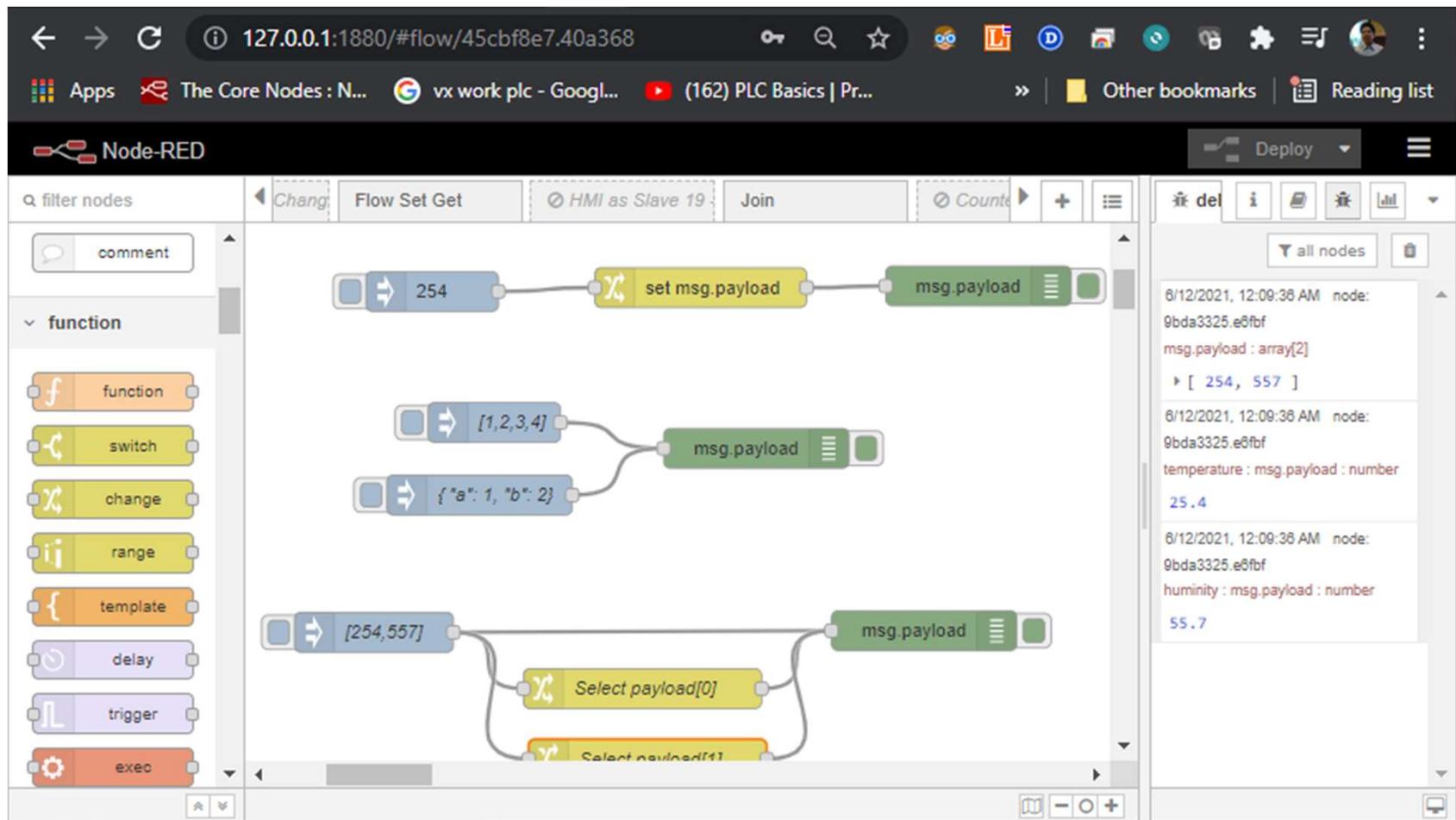
# Node-Red

- *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



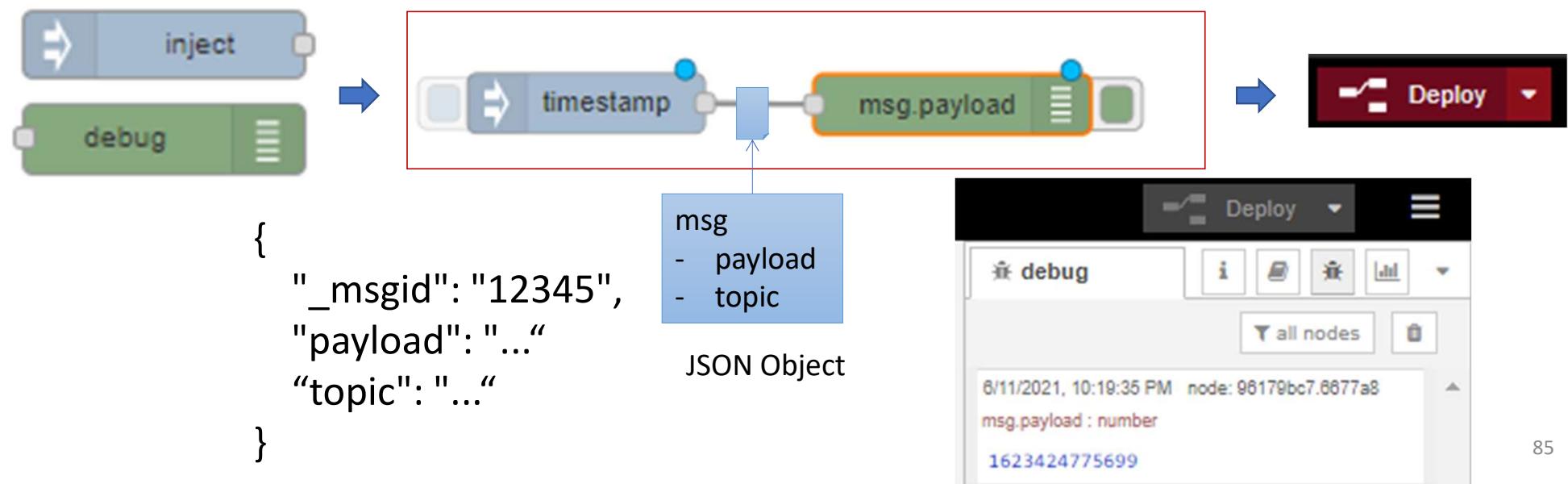
The screenshot shows a Windows Command Prompt window titled "Select Command Prompt". The window title bar includes standard window controls (minimize, maximize, close). The main area of the window displays the following text:  
Microsoft Windows [Version 10.0.19042.1052]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\chalermchon>`node-red`

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



# Basic Nodes

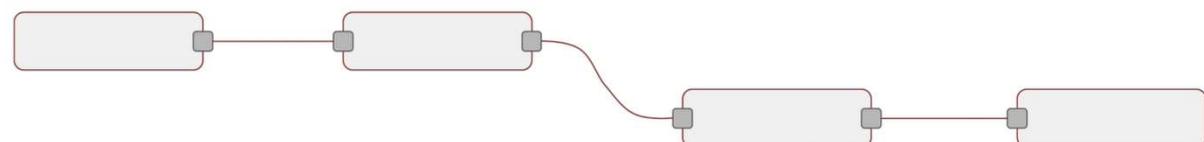
- Input Node นำข้อมูลเข้า node-red
- Function Node กำหนดการทำงานเปลี่ยนแปลงค่า
- Output Node แสดงผลข้อมูลในรูปแบบต่าง



# JSON Object (Data Transfer)

- 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

ข้อมูลตัวแปร      ค่าตัวแปร

{ "name" : "John" } ข้อมูลความ

{ "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 containing an 'inject' node. The 'msg' object is expanded to show 'payload' and 'topic' properties. The 'topic' property is being configured in the 'Edit inject node' dialog.

**Edit inject node**

**Properties**

- Name:** Name
- msg. payload:** timestamp
- msg. topic:**  (highlighted with a red box)

**Repeat:** none

**Enabled:**

**⋮** string  
number  
boolean  
JSON  
buffer  
timestamp  
expression  
env variable

**⋮** Inject once  seconds, then

**⋮** Repeat  none

Annotations in Thai:

- A red box highlights the 'topic' input field with the text: กำหนดค่าชนิดข้อมูล (Define data type).
- A red box highlights the 'Repeat' dropdown with the text: กำหนดเวลาทำซ้ำ (Set repeat time).
- A red box highlights the 'Repeat' dropdown menu with the text: กำหนดเวลาทำซ้ำ (Set repeat time).

## กำหนดค่า payload และการแสดงค่า Debug

### JavaScript Types

```
boolean : true / false  
number : 123  
string : "hello"  
array : [ 1, 2, 3, 4 ]  
object : { "color": "red" }
```



▶ Name

[1,2,3,4]

≡ msg. payload = ▾ J: [ 1,2,3,4 ]

≡ msg. topic = ▾ a\_z

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

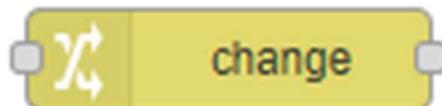
a: 1  
b: 2

▶ Name

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

≡ msg. payload = ▾ J: { "a": 1, "b": 2 }

≡ msg. topic = ▾ a\_z



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

Set

to msg. payload

msg. payload

to msg.

- msg.
- flow.
- global.
- string
- number
- boolean
- JSON
- buffer
- timestamp
- expression

Set

to msg. payload.temperature\_c

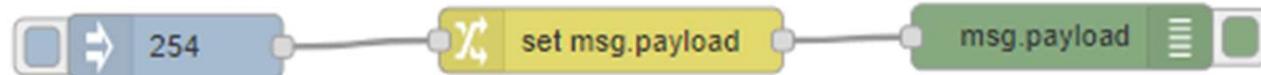
to J: (payload.temperature-32)\*5/9

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

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

Change กำหนดค่า payload / 10

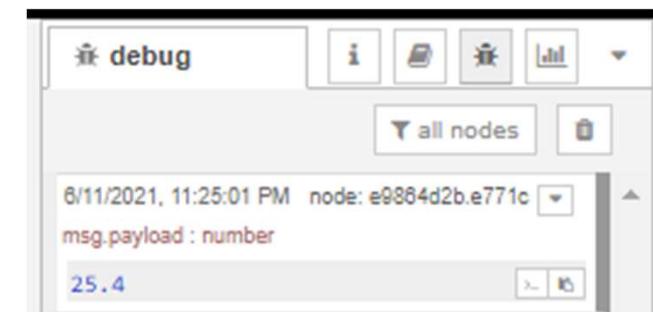
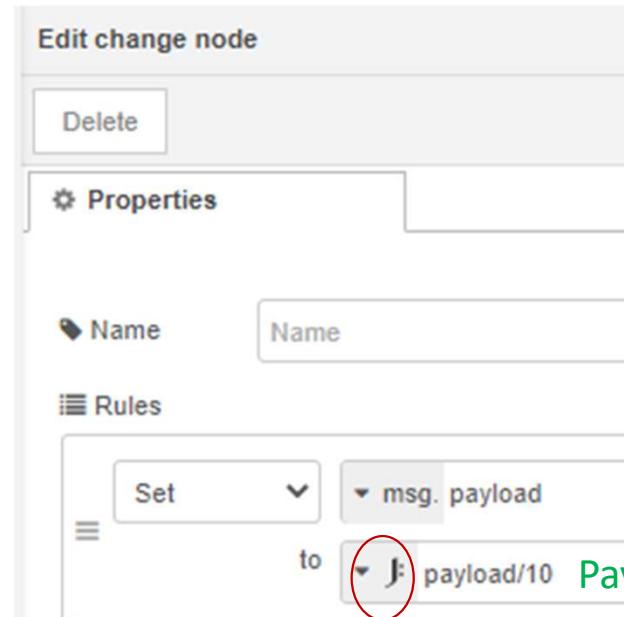
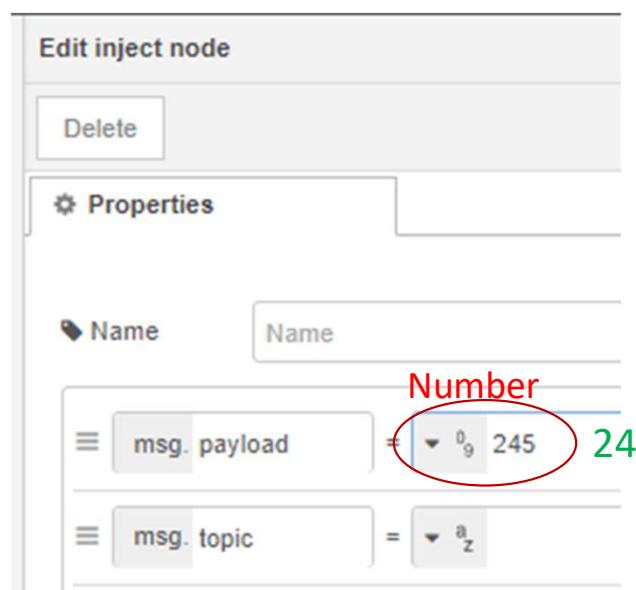
GitHub Node-Red - 02 Basic Change



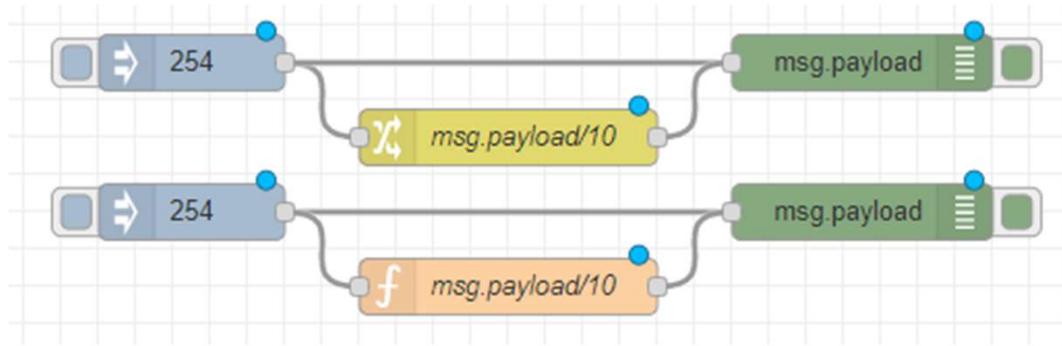
payload (254) - input

Change (254/10)

Payload 25.4 - output



<https://github.com/chalermchonv/UtccFoodlotCodes/tree/main/Node-Red>



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

Edit change node

Delete      Cancel      Done

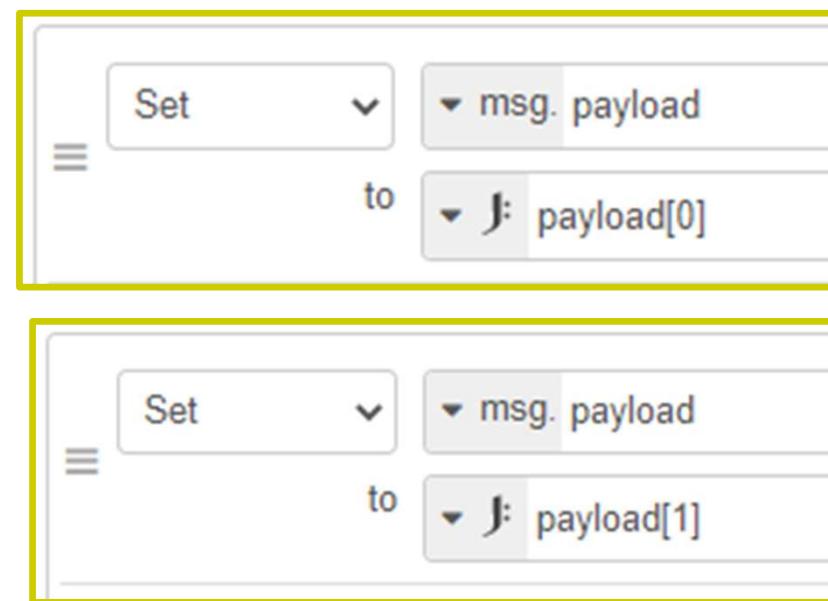
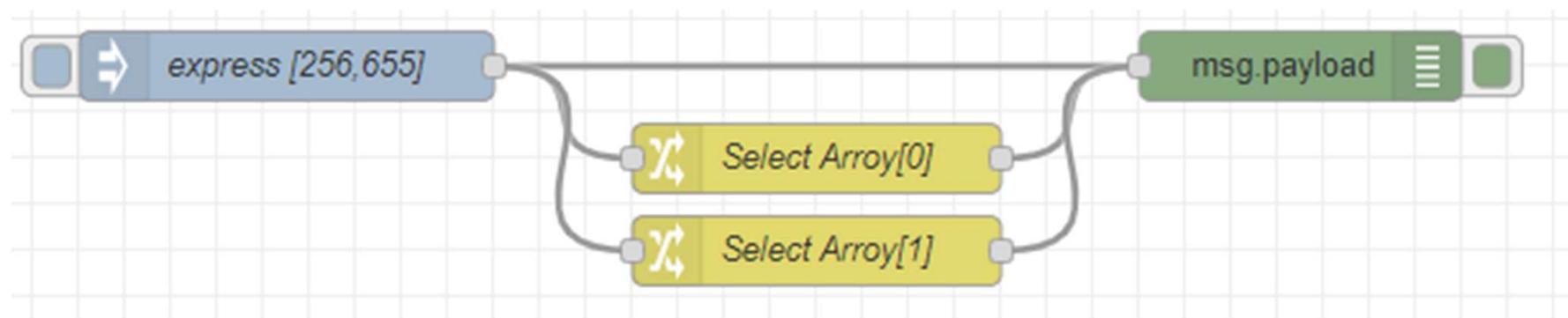
**Properties**

Name: msg.payload/10

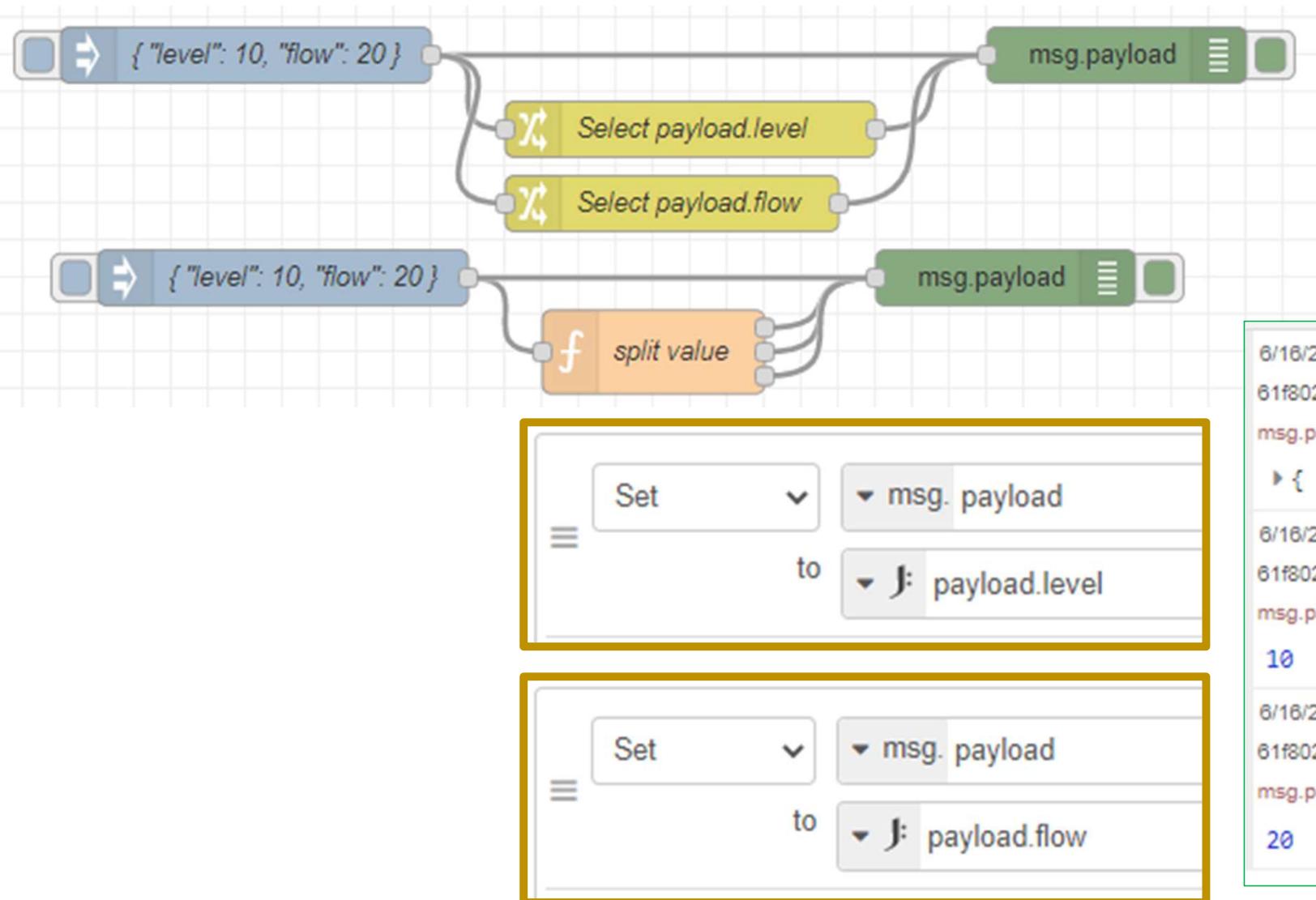
**Rules**

Set msg.payload to J: payload/10

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



Timestamp	Node ID	msg.payload
6/16/2021, 9:26:23 PM	89d312c3.58257	msg.payload : array[2] [ 256, 655 ]
6/16/2021, 9:26:23 PM	89d312c3.58257	msg.payload : number 256
6/16/2021, 9:26:23 PM	89d312c3.58257	msg.payload : number 655



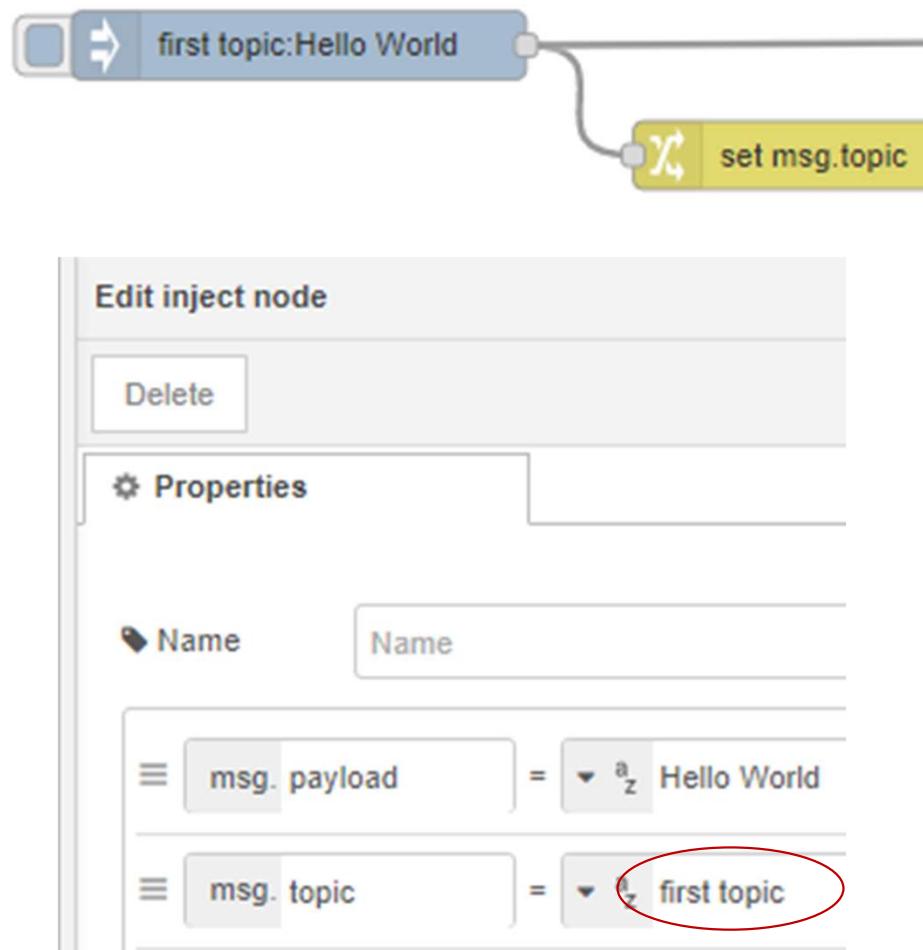
```

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
  
```

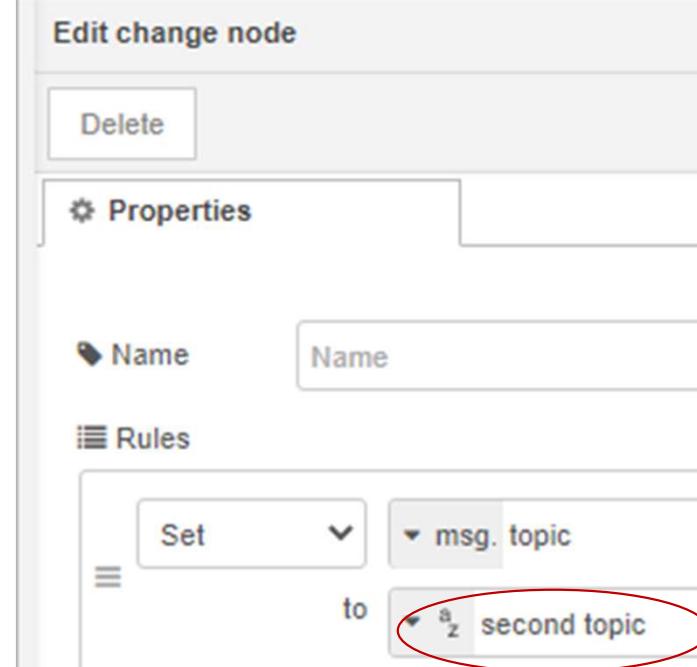
## Change กำหนดค่า topic



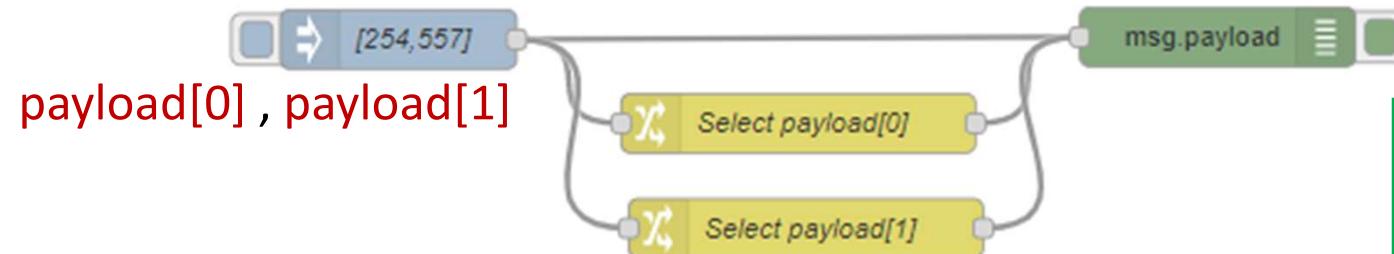
## GitHub Node-Red - 02 Basic Change

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"



## แยกค่าเป็น สອງทางและกำหนด topic ใหม่ GitHub Node-Red - 02 Change-Function



payload[0] , payload[1]

**Edit change node**

Delete

**Properties**

Name: Select payload[0]

**Rules**

Set msg. payload to **payload[0]/10** Express Payload[0]/10

Set msg. topic to **a\_z temperature**

Set msg. payload to **payload[1]/10** Express Payload[1]/10

Set msg. topic to **a\_z humidity**

**Edit change node**

Delete

**Properties**

Name: Select payload[1]

**Rules**

Set msg. payload to **payload[1]/10** Express Payload[1]/10

Set msg. topic to **a\_z humidity**

6/12/2021, 12:09:36 AM node: 9bda3  
msg.payload : array[2]  
[ 254, 557 ]

6/12/2021, 12:09:36 AM node: 9bda3  
temperature : msg.payload : number  
25.4

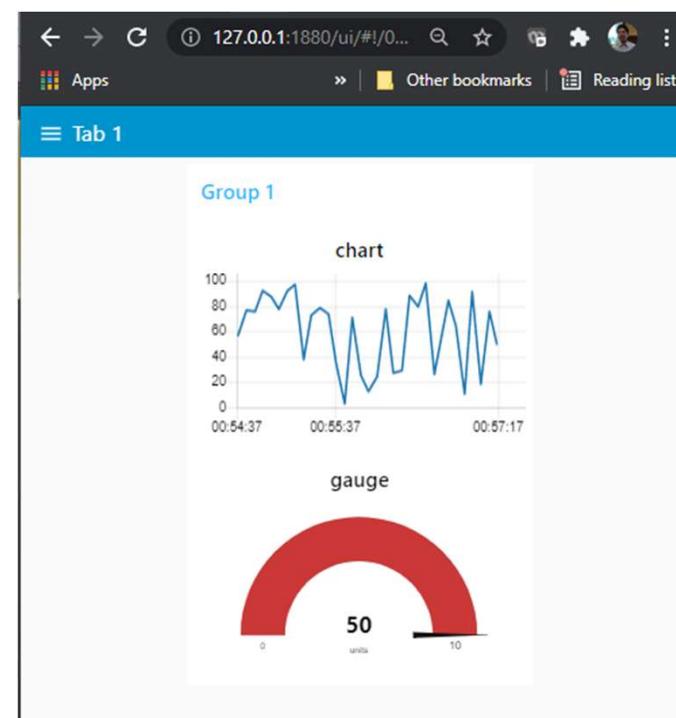
6/12/2021, 12:09:36 AM node: 9bda3  
humidity : msg.payload : number  
55.7

Express  
Payload[1]/10

# Dashboard Module

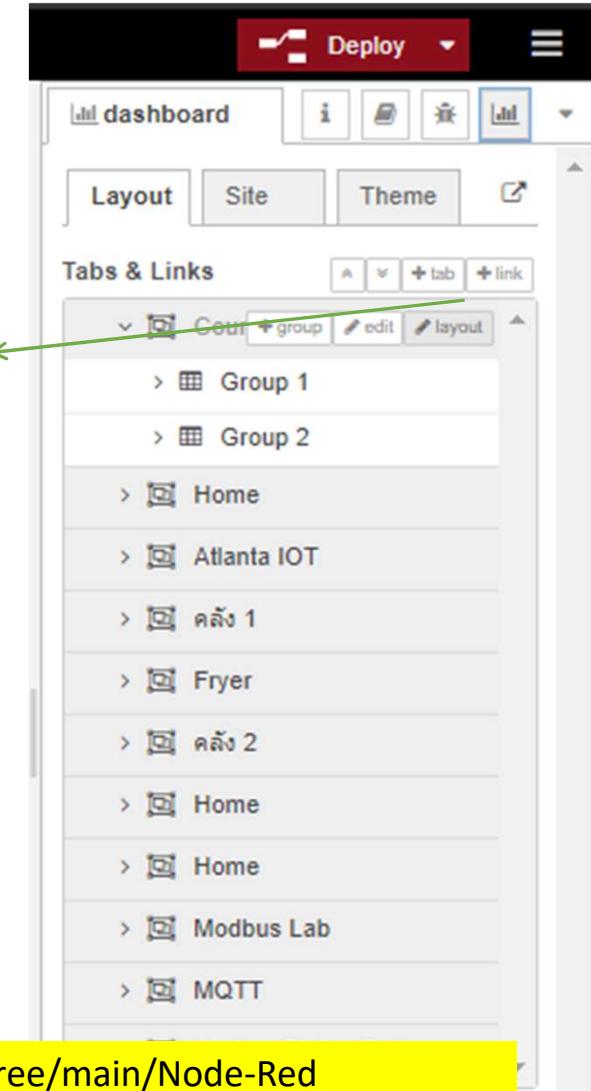
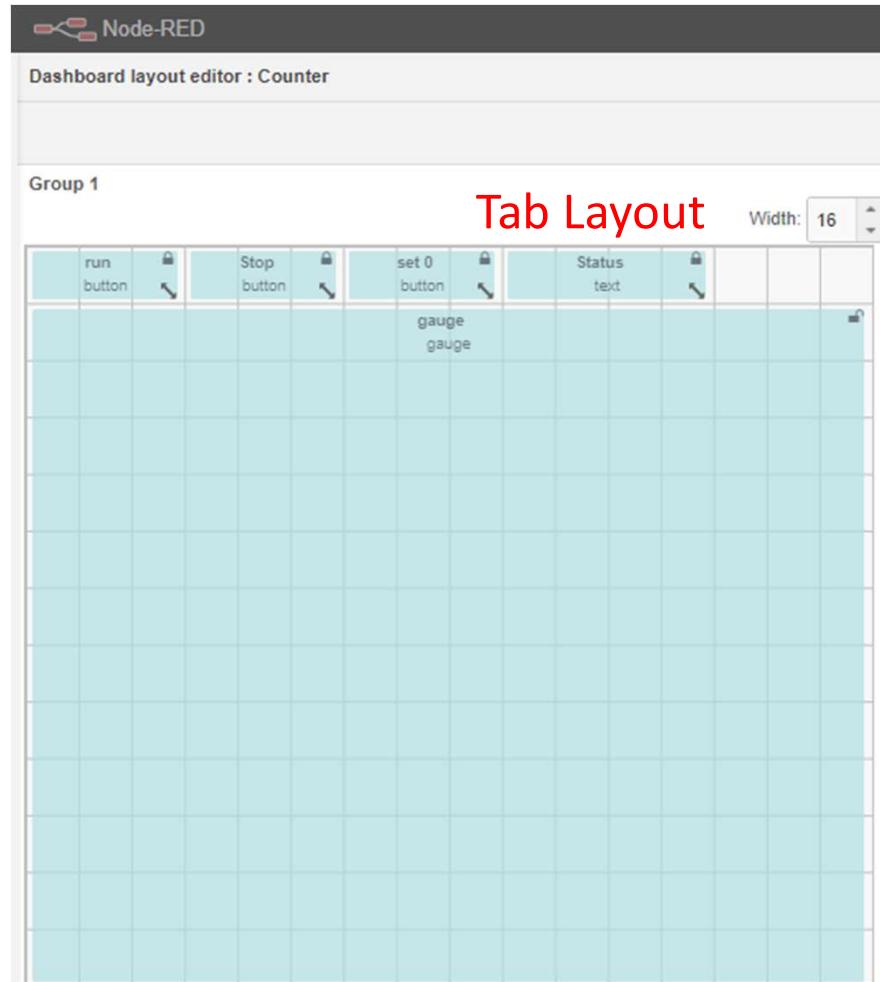


<http://127.0.0.1:1880/>



สร้าง Tab และ Group เพื่อกำหนดพื้นที่แสดง

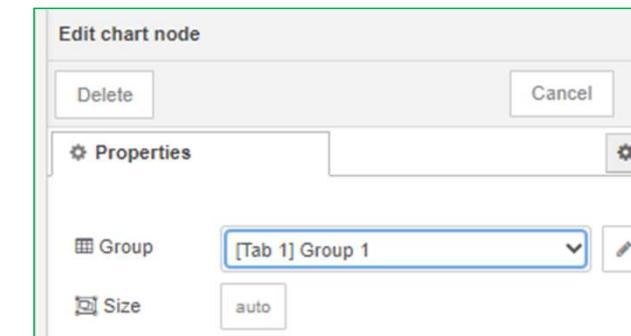
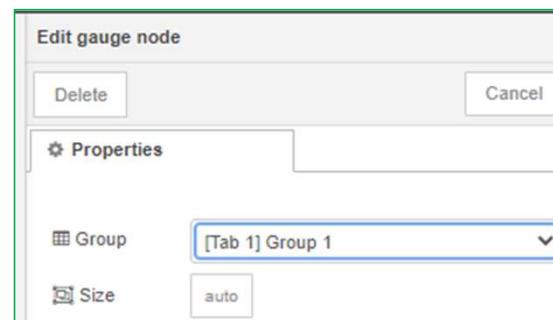
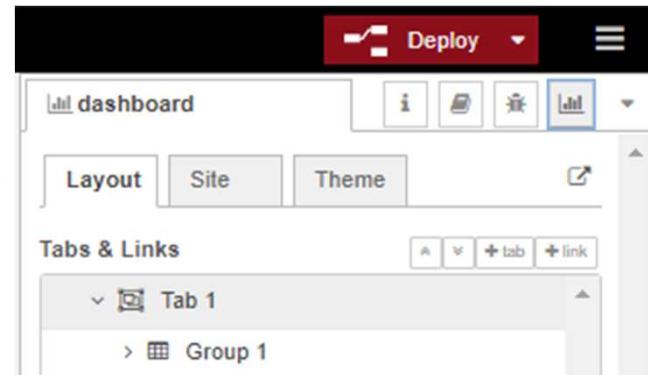
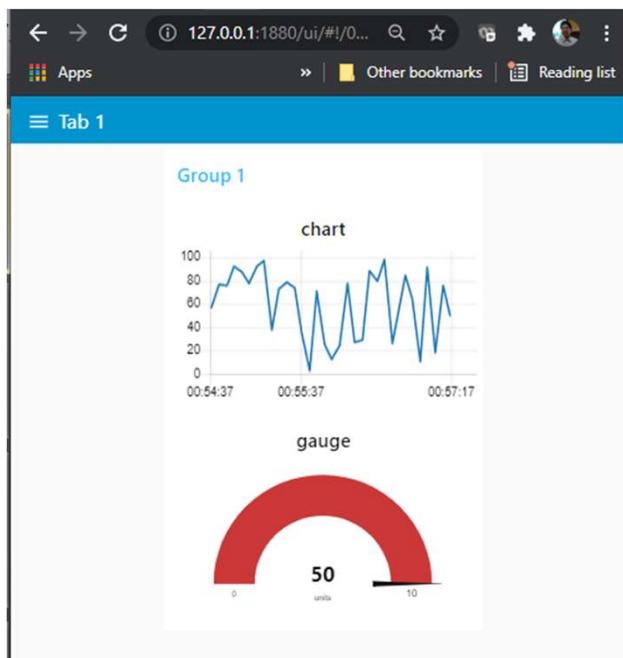
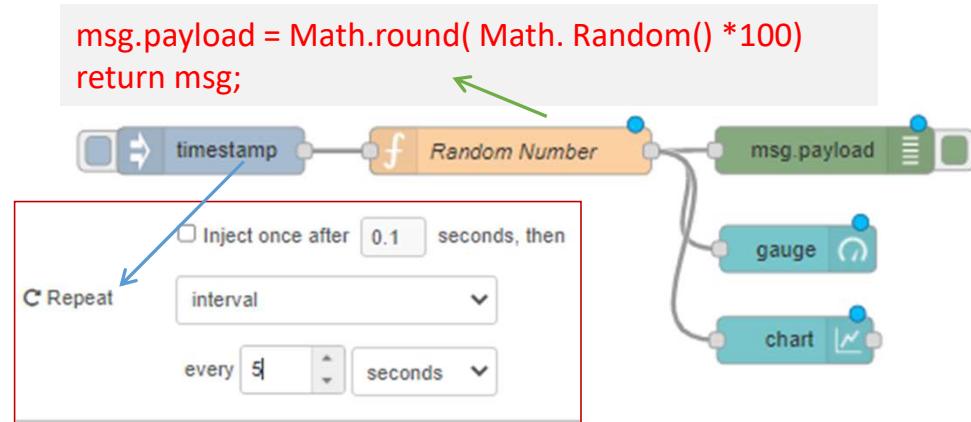
GitHub Node-Red 03 Basic Dashboard

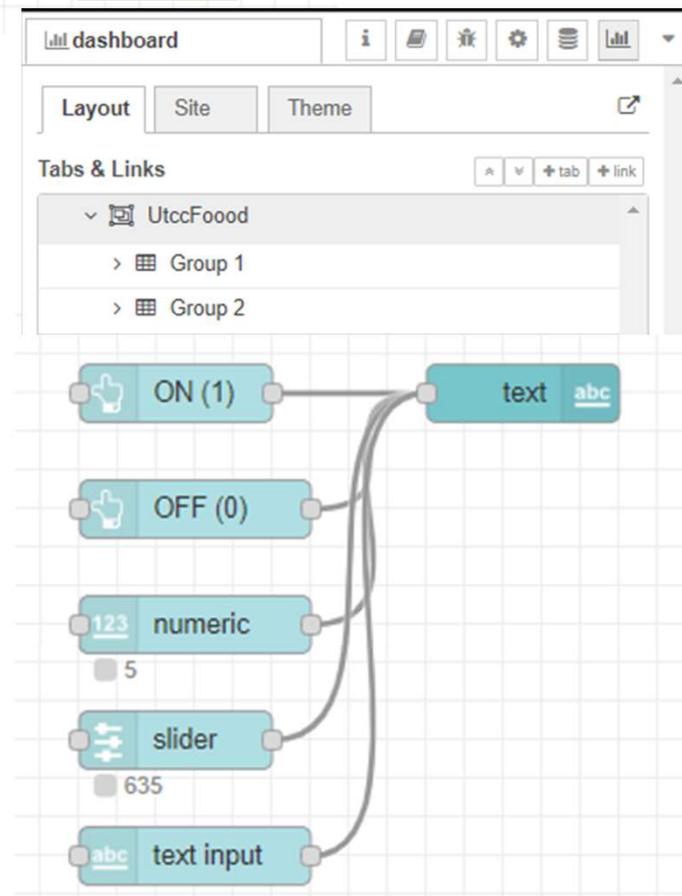
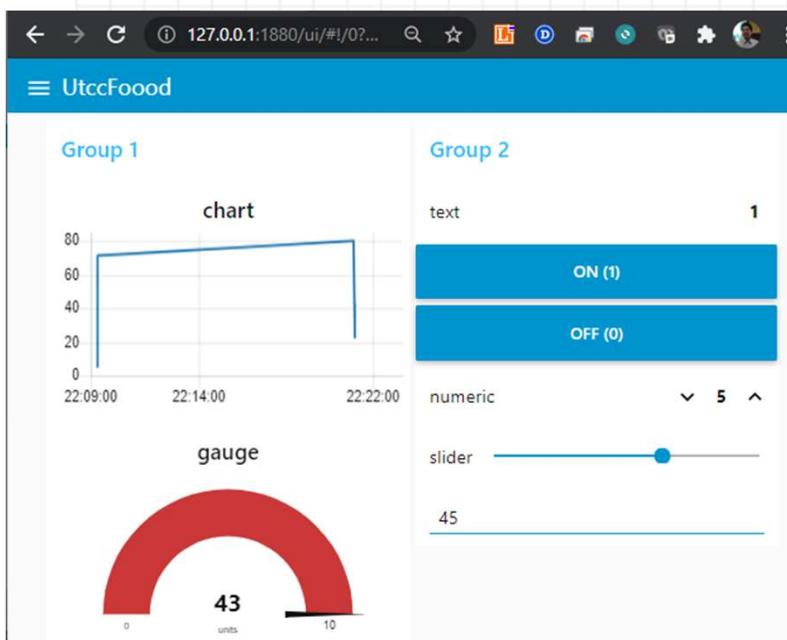
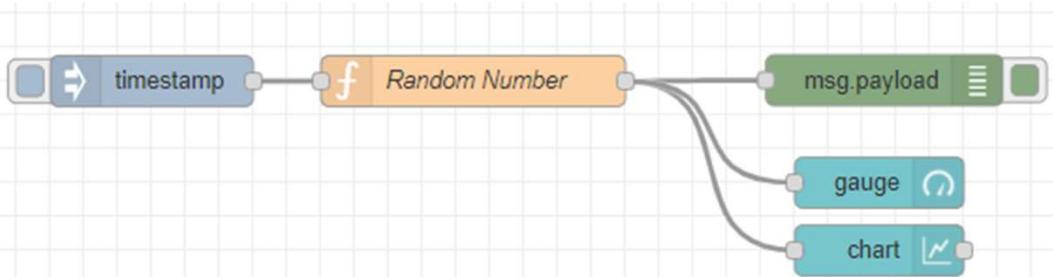


<https://github.com/chalermchonv/UtccFoodlotCodes/tree/main/Node-Red>

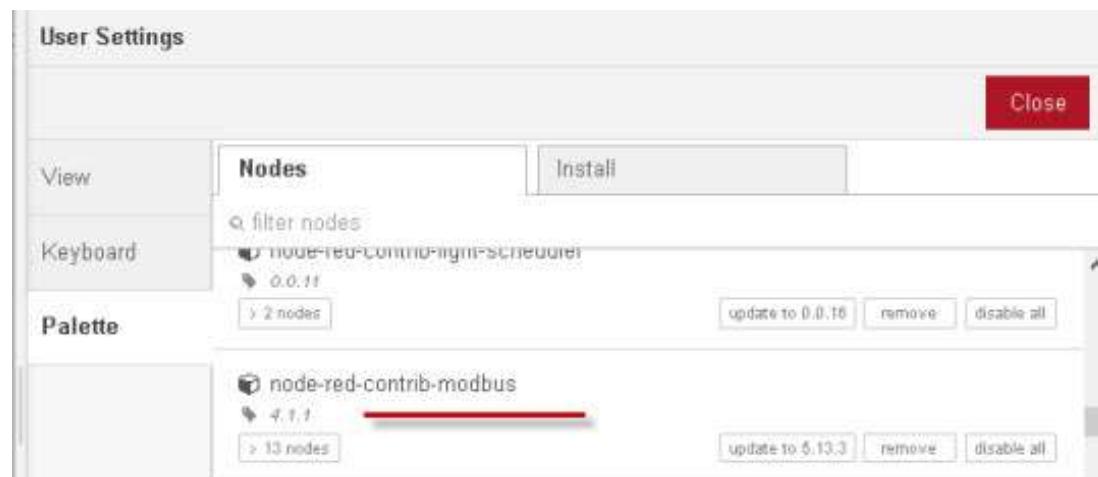
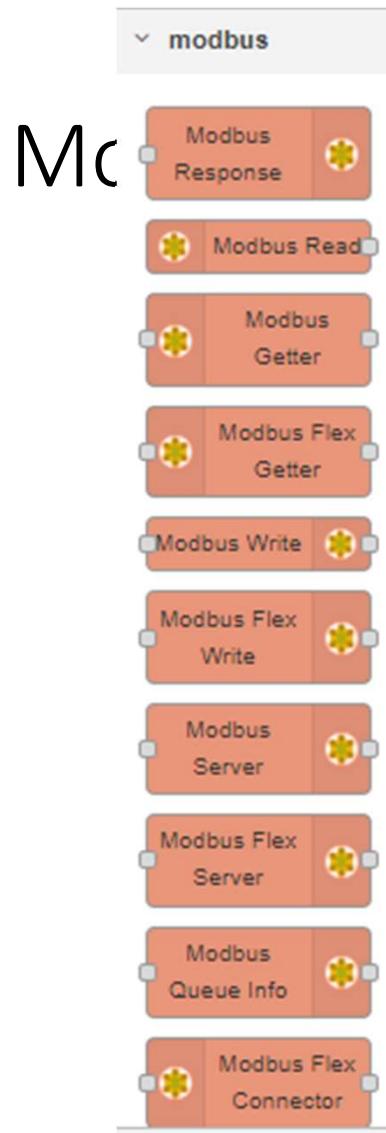
## Basic Dashboard Control

## GitHub Node-Red 03 Basic Dashboard





## YouTube Node-Red 04 Modbus



Sensor , Machine, PLC,HMI

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

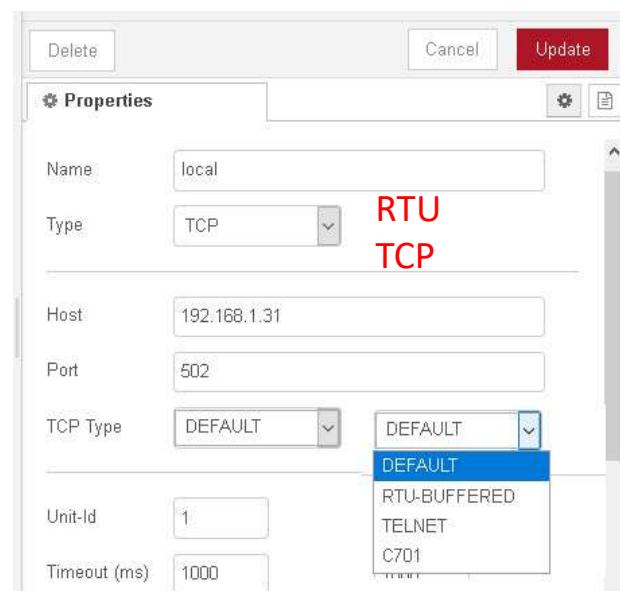
Connected (1/10) : (received/sent) (7681/7681) Serv. write de Rx: Tx:

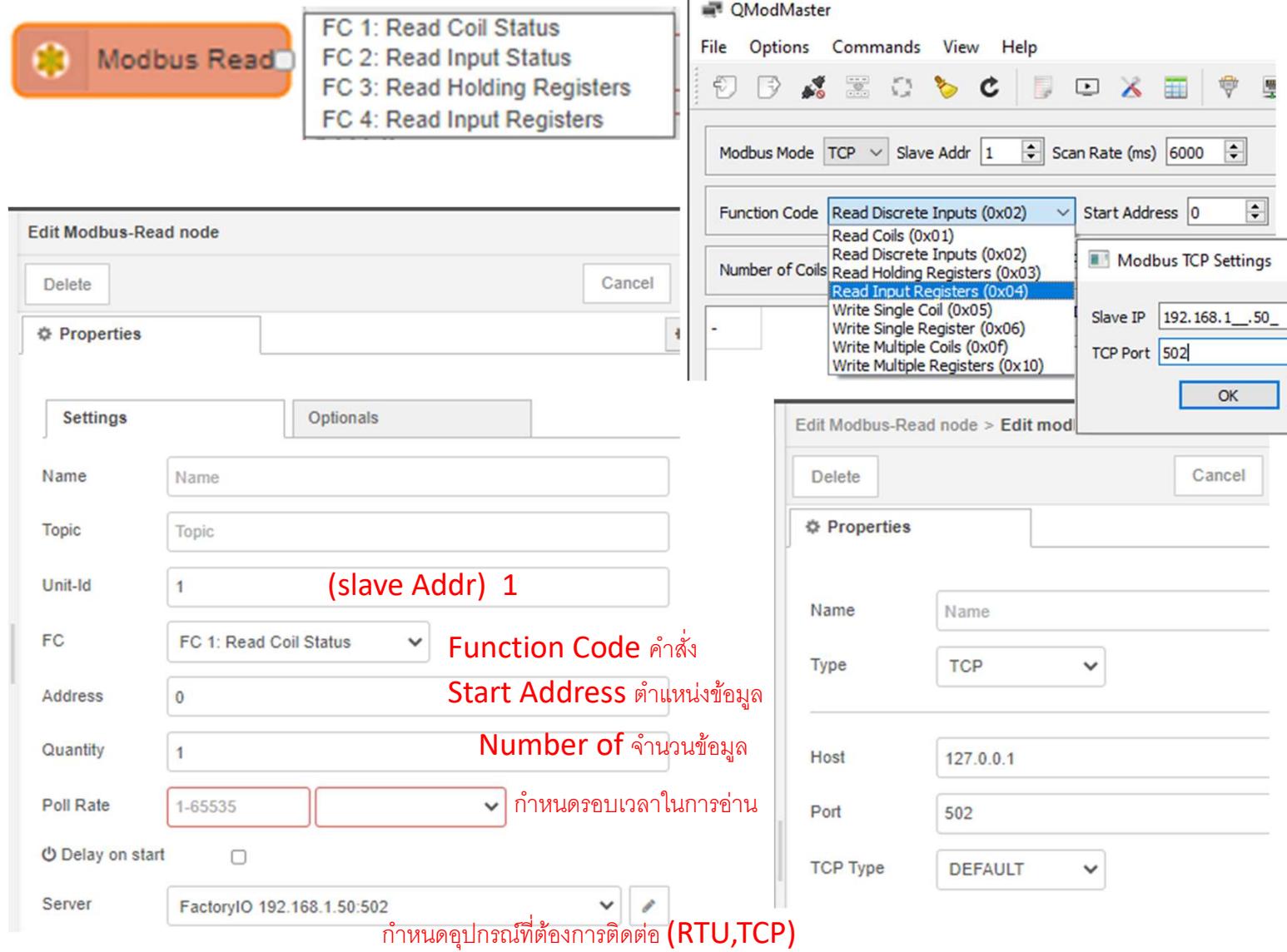
Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+1
000001-000016	1	0	0	0	0	0	0	0	0	0	0
000017-000032	0	0	0	0	0	0	0	0	0	0	0
000033-000048	0	0	0	0	0	0	0	0	0	0	0
000049-000064	0	0	0	0	0	0	0	0	0	0	0

Hex Dump:

```

000001-000016: 41 00 00 00 00 00 00 00 00 00 00 00
000017-000032: 00 00 00 00 00 00 00 00 00 00 00 00
000033-000048: 00 00 00 00 00 00 00 00 00 00 00 00
000049-000064: 00 00 00 00 00 00 00 00 00 00 00 00
    
```





The screenshot shows the configuration dialog for a 'Modbus Write' node in Node-Red. The dialog has a header with 'Edit Modbus-Write node' and buttons for 'Delete', 'Cancel', and 'Done'. Below the header is a 'Properties' tab with three icons: gear, file, and square.

The configuration fields are:

- Name: Name
- Unit-Id: 1
- FC: A dropdown menu showing:
  - FC 5: Force Single Coil
  - FC 6: Preset Single Register
  - FC 15: Force Multiple Coils
  - FC 16: Preset Multiple Registers
- Address: (empty)
- Quantity: (empty)
- Server: modbus-tcp@127.0.0.1:502

To the right of the dialog, there is a 'Modbus Read' node icon and a list of function codes:

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

YouTube Node-Red 04 Modbus FactoryIO

The image shows a 3D model of a conveyor belt system. A sensor is mounted on the side of the conveyor, connected by a red dashed line to a yellow callout box labeled "Sensor". Above the conveyor, there is a small control panel with two buttons: "Conveyor" (green) and "Sensor" (orange).

**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

Factory IO

ค่าปุ่มจาก Server (Master) ภายนอก (Node-Red , QModMaster, PLC, HMI)

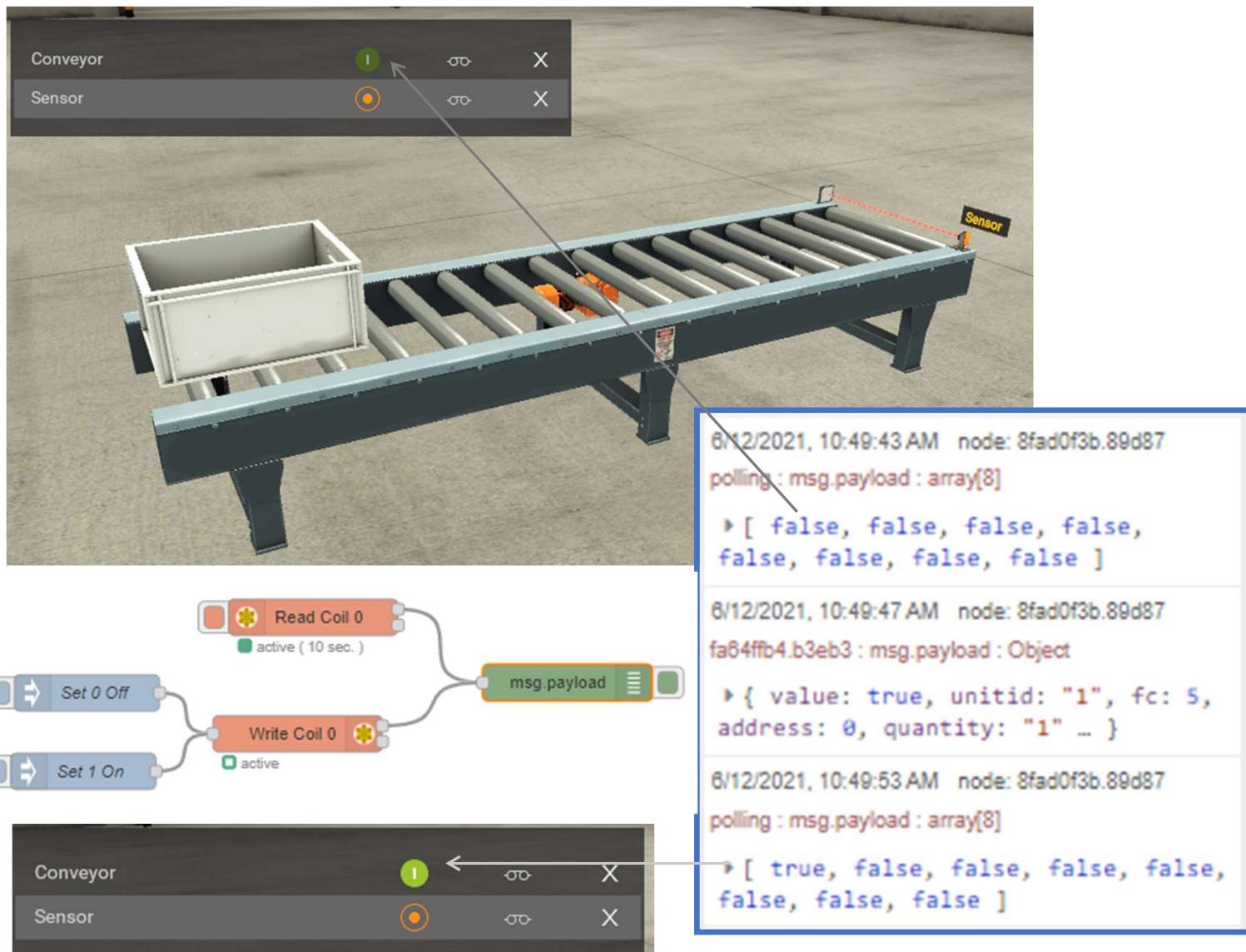
← DRIVER Modbus TCP/IP Server START CONFIGURATION CLEAR

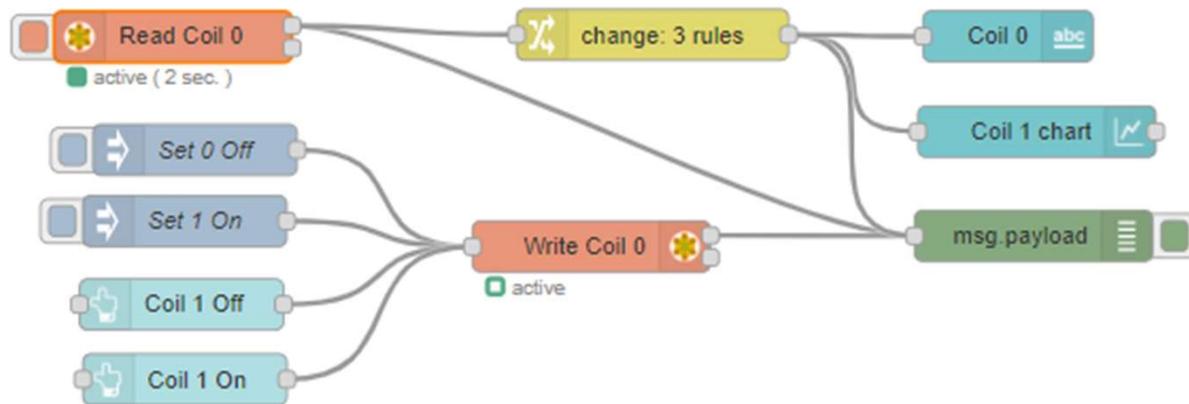
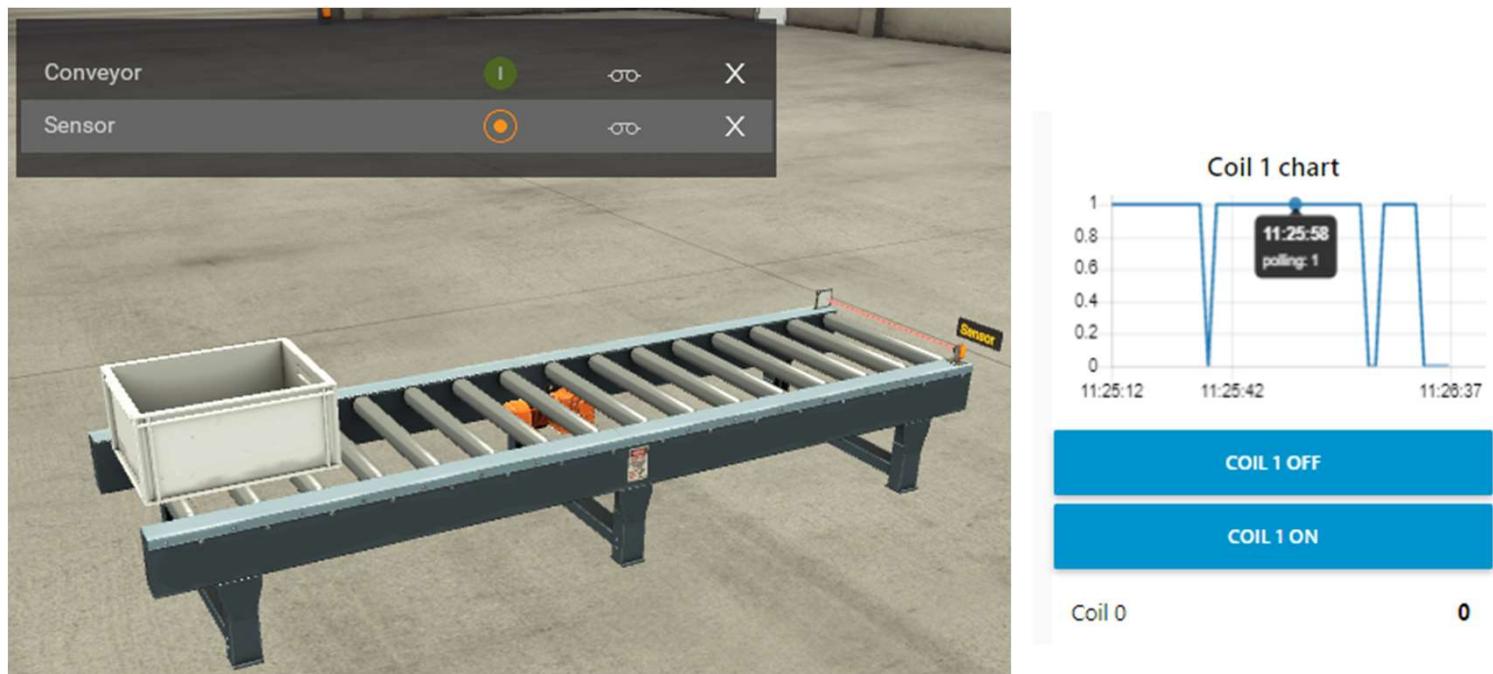
**SENSORS**

	Input	
FACTORY I/O (Paused)	Sensor	(192.168.1.50:502) Slave ID:1
FACTORY I/O (Reset)		Input 0
FACTORY I/O (Running)		Input 1
FACTORY I/O (Time Scale)	I/O (Running)	
	Sensor	

**ACTUATORS**

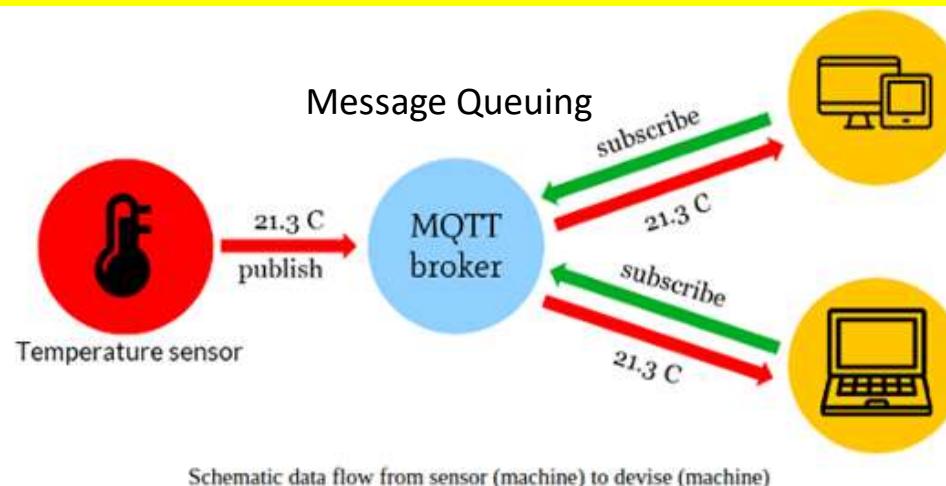
	Output -Coil	
Conveyor	Coil 0	Conveyor
FACTORY I/O (Camera Position)	FC 1 (Read)	
FACTORY I/O (Pause)		FC 5 (Write)
FACTORY I/O (Reset)		
FACTORY I/O (Run)		





# 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

## Protocol

mqtt / tcp

## Host

159.138.241.21



Menu



Connected

Add publisher

Add subscriber



UTCC MQTT - mqtt://159.138.241.21

Topic to publish

msg

QoS

0 - Almost Once

Retain

Payload Type

Strings / JSON / XML / Characters

e.g. {"hello": "world"}

Payload

{temp:32,Himi:80}

Publish

x msg

{temp:32,Himi:80}

qos : 0, retain : false, cmd : publish, dup : false, topic : msg, messageId : , length : 22

{temp:32,Himi:80}

qos : 0, retain : false, cmd : publish, dup : false, topic : msg, messageId : , length : 22

☰ Menu



MQTT CLIENT SETTINGS

MQTT Client Name

UTCC MQTT

MQTT Client Id

dragino-205f4c

Protocol

mqtt / tcp

Host

159.138.241.21

Username

Username

Password

Password

Reconnect Period (milliseconds)

1000

Connect Timeout (milliseconds)

30000

Will - Topic

Will - Topic

Will - QoS

0 - Almost Once

Save

## YouTube MQTT 02 Node-Red

Edit mqtt out node

Properties

Server: test.mosquitto.org

Topic: utccfood

QoS:  Retain:

Name: Name

Edit mqtt in node

Properties

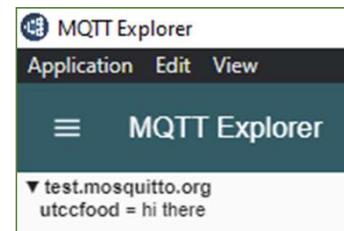
Server: test.mosquitto.org

Topic: utccfood

QoS: 2

Output: auto-detect (string or buffer)

Name: Name



Edit mqtt out node > Add new mqtt-broker config node

Properties

Name: test.mosquitto.org

Connection  Security  Messages

Server: 159.138.241.21 Port: 1883

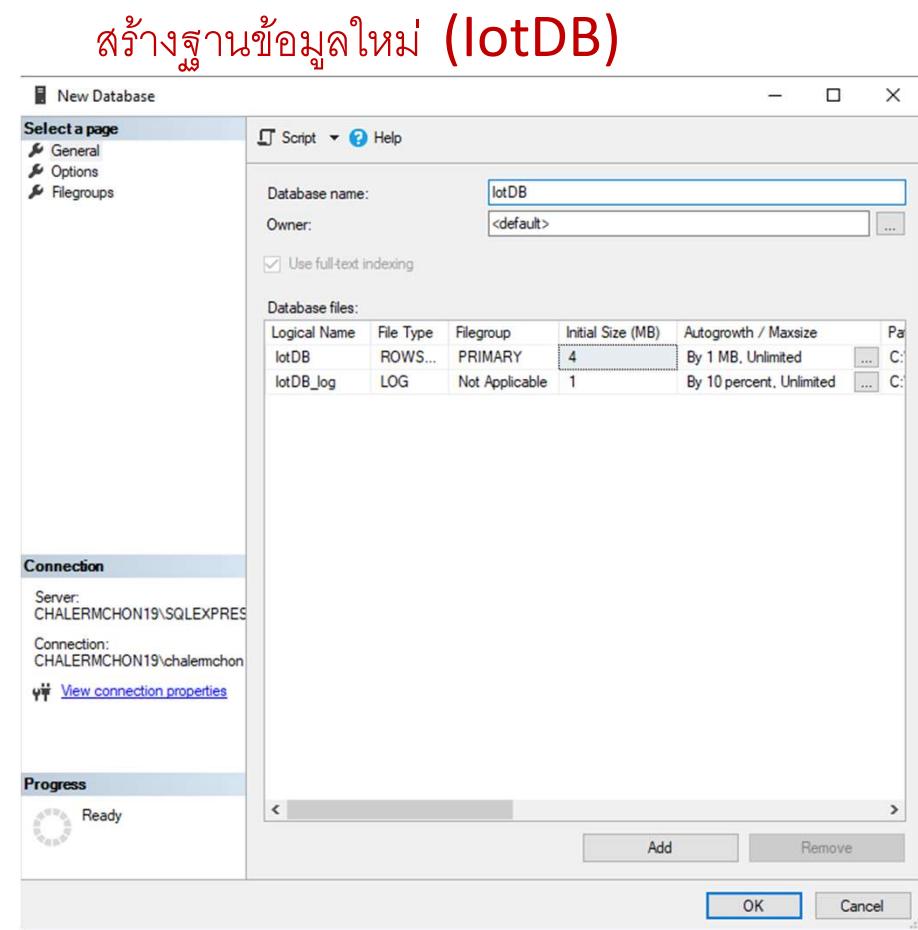
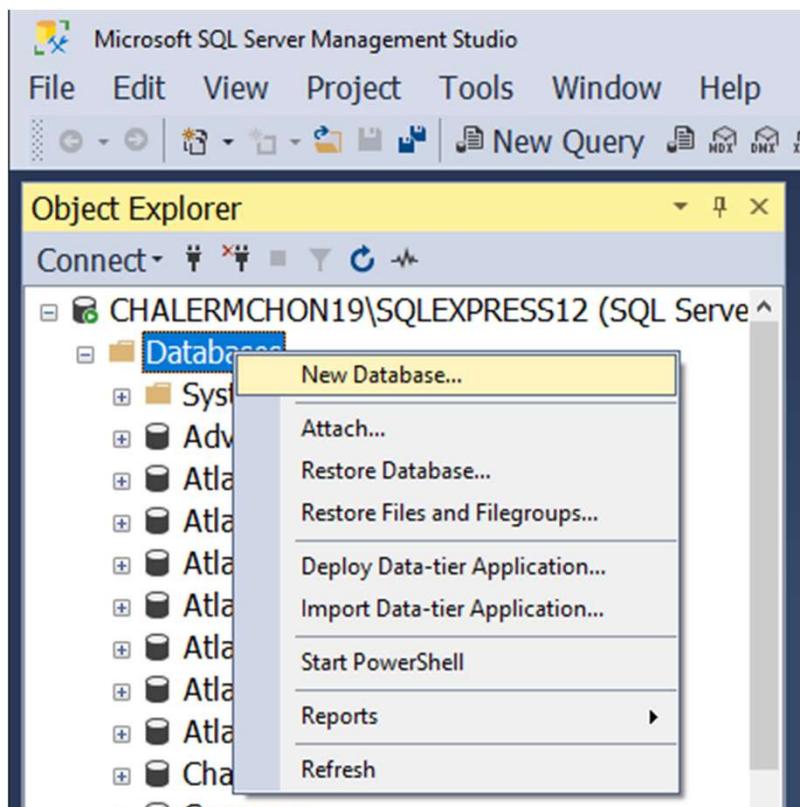
Enable secure (SSL/TLS) connection

Client ID: Leave blank for auto generated

Keep alive time (s): 60  Use clean session

Use legacy MQTT 3.1 support

# สร้างระบบฐานข้อมูลใหม่ (Mssql)



# สร้างตารางข้อมูลใหม่ ใน IoTDB

Object Explorer

Connect ▾

- IotDB
  - Database Diagrams
  - Table** ▾
    - New
    - Table...
    - File Table...
  - View
  - Sync
  - Programs
  - Server
  - Storage
  - Security

CHALERMCHON...dbo.IotData\* ▾

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
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"/>

Choose Name

Enter a name for the table:

**OK**

Column Properties

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

## Insert New Data

YouTube MsSQL 01

### IotDB

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

Table...  
Design  
Select Top 1000 Rows  
**Edit Top 200 Rows**

**CHALERMCHON...dbo.IotData**

	<b>id</b>	<b>time</b>	<b>iotCode</b>	<b>SensorCode</b>	<b>SensorValue</b>	<b>LocationCode</b>	<b>CreateDate</b>
...	NULL	NULL	iot01	t01	25.4	wh01	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Select Data (Sql)

### IotDB

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

Table...  
Design  
**Select Top 1000 Rows**  
Edit Top 200 Rows  
Script Table as  
View Dependencies  
Memory Optimization Advisor

**SQLQuery1.sq...rmchon (57)**

```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [id]
,[time]
,[iotCode]
,[SensorCode]
,[SensorValue]
,[LocationCode]
,[CreateDate]
FROM [IotDB].[dbo].[IotData]
```

95 %

Results Messages

	<b>id</b>	<b>time</b>	<b>iotCode</b>	<b>SensorCode</b>	<b>SensorValue</b>	<b>LocationCode</b>	<b>CreateDate</b>
1	NULL	0x000000000000007D1	iot01	t01	25.4	wh01	2021-06-13 22:33:37.550

112

# Select IoTData table by Node-Red

The screenshot displays a Node-Red flow and its configuration interface. The flow consists of three nodes: a timestamp node, an MSSQL-PLUS node, and a msg.payload node. The timestamp node is followed by the MSSQL-PLUS node, which is then followed by the msg.payload node.

**Configuration Details:**

- MSSQL-PLUS Node Properties:**
  - Connection: MsSQL Server
  - Name: Name
  - Query mode: Query
  - Query Editor:

```
1 SELECT TOP (30) *
2 FROM [IoTDB].[dbo].[IoTData]
3
```
- msg.payload Node:** Outputs the payload from the MSSQL node.
- Message Preview:** A modal window shows the message payload structure and content. The payload is an array of objects, with the first object containing fields like id, time, IoTCode, SensorCode, SensorValue, LocationCode, and CreateDate.
- Server Configuration:** A separate configuration window shows the connection details for the MsSQL Server:
  - Name: MsSQL Server
  - Server: localhost
  - Port: 1433
  - Username: iot
  - Password: @iot

**Text at Bottom Left:**

```
SELECT TOP (30) *
FROM [IoTDB].[dbo].[IoTData]
```

**Page Number:** 113

```
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 current database is 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');
```

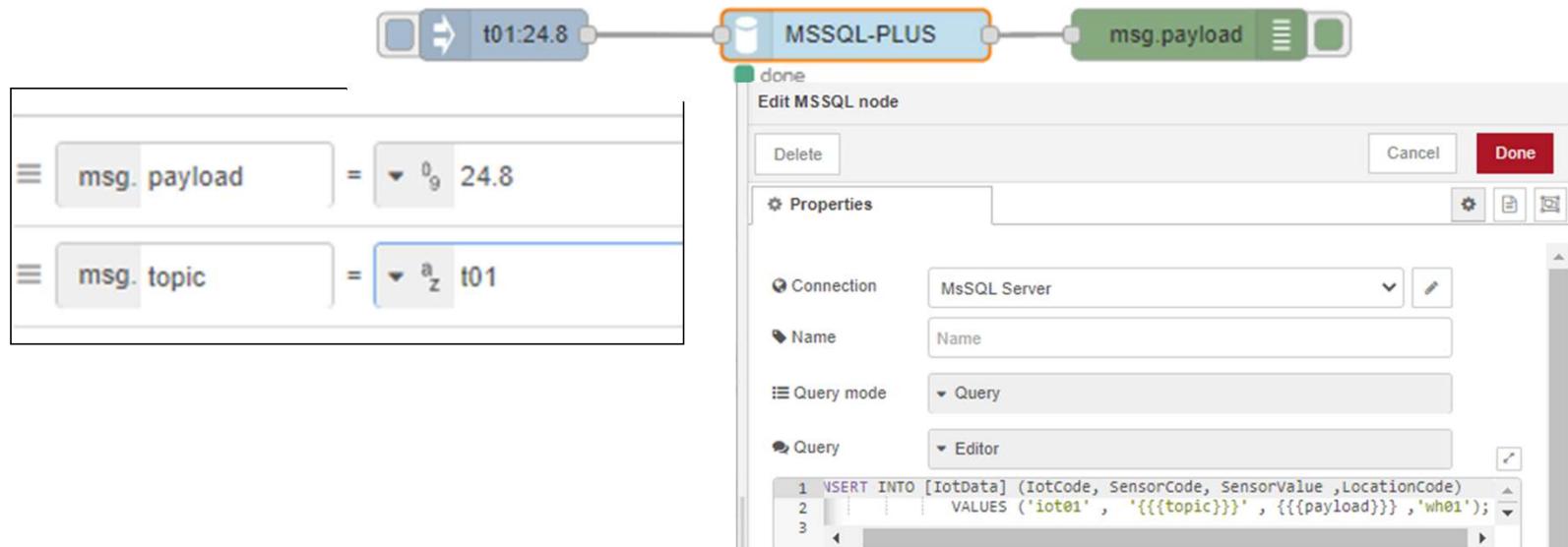
Below the code, there is a script for a "SelectTopNRows" command:

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

The Results pane at the bottom shows a table with two rows of data:

	id	time	IotCode	SensorCode	SensorValue	LocationCode	CreateDate
1	1	0x0000000000000007D3	iot01	t01	25.4	wh01	2021-06-13 22:33:37.550
2	2	0x0000000000000007D5	iot01	t01	25.9	wh01	2021-06-13 22:59:49.507

# Insert lotData by topic and payload



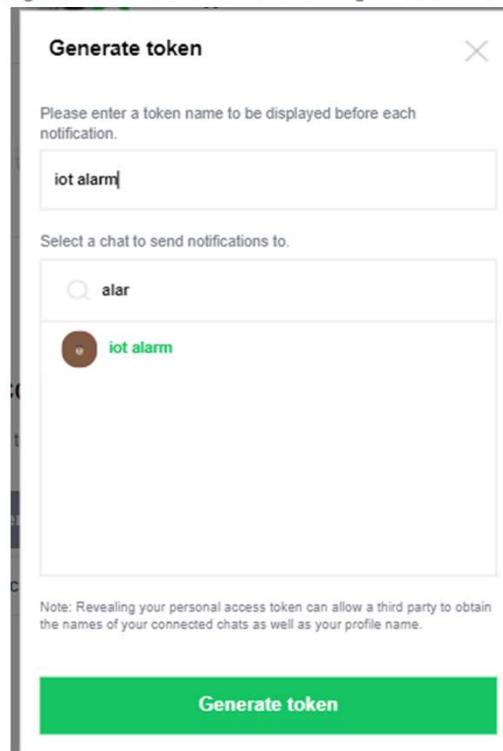
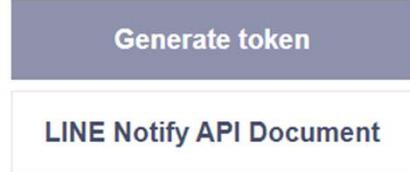
INSERT INTO [IoTData] (IoTCode, SensorCode, SensorValue ,LocationCode)  
VALUES ('iot01', '{{topic}}', {{payload}}, 'wh01');

	id	time	IoTCode	SensorCode	SensorValue	LocationCode	CreateDate
1	1	0x0000000000000007D3	iot01	t01	25.4	wh01	2021-06-13 22:33:37.550
2	2	0x0000000000000007D5	iot01	t01	25.9	wh01	2021-06-13 22:59:49.507
3	3	0x0000000000000007D6	iot01	t01	25.9	wh01	2021-06-13 23:04:37.707
4	4	0x0000000000000007D7	iot01	t01	24.8	wh01	2021-06-13 23:10:22.693

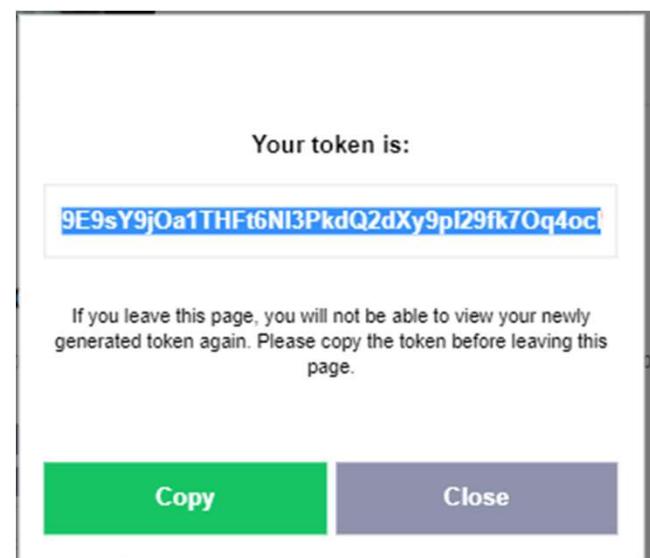
# Line Notify

## Generate access token (For developers)

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



9E9sY9jOa1THFt6NI3PkQ2dXy9pl29fk7Oq4ocNxcA





```

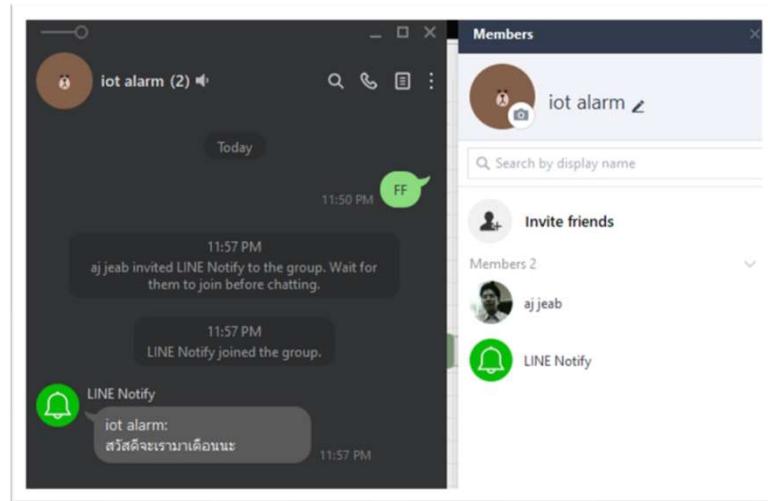
msg.headers = {'content-type': 'application/x-www-form-urlencoded'
    , 'Authorization': 'Bearer YourLineToken'};
msg.payload = {"message": "alarm " + msg.payload };
return msg;

```

```

msg.headers = {'content-type': 'application/x-www-form-
urlencoded', 'Authorization': 'Bearer
9E9sY9jOa1THFt6NI3PkdQ2dXy9pl29fk7Oq4ocNxcA'};
msg.payload = {"message": "alarm " + msg.payload };
return msg;

```



Edit http request node

Delete

**Properties**

Method: POST

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

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

# Create Google Form and Google Sheet

lotInput

lotInput Sheet

Form description

metric

Short answer text

t01

value

25

tag

m01

Get link

lotInput (Responses)

File Edit View Insert Format Data Tools F

G8

	A	B	C	D
1	Timestamp	metric	value	tag
2	6/25/2021 9:11:47	t01	25	m01
3				
4				

[https://docs.google.com/forms/fromyourgoogleform/viewform?usp=pp\\_url&entry.179783227=t01&entry.1383307148=25&entry.981248842=m01](https://docs.google.com/forms/fromyourgoogleform/viewform?usp=pp_url&entry.179783227=t01&entry.1383307148=25&entry.981248842=m01)



[https://docs.google.com/forms/fromyourgoogleform  
/viewform?usp=pp\\_url&entry.179783227=t01  
&entry.1383307148=25  
&entry.981248842=m01](https://docs.google.com/forms/fromyourgoogleform/viewform?usp=pp_url&entry.179783227=t01&entry.1383307148=25&entry.981248842=m01)

```

var max = 30 ;
var min = 20;
var ranv = Math.round(Math.random()*(max-min)) + min;
msg.payload = {"metric":'t01',"value": ranv,"tag": 'm01'};
return msg;
  
```

viewform => formResponse

[https://docs.google.com/forms/d/e/1FAIpQLSfCHBjMEH2KKtr-  
k15SwKM9GZXb1o4KY\\_JpzF\\_OCIW4RYcMhw/formResponse?usp=pp\\_url&entry.  
.179783227={{payload.metric}}&entry.1383307148={{payload.value}}&entry.  
981248842={{payload.tag}}}](https://docs.google.com/forms/d/e/1FAIpQLSfCHBjMEH2KKtr-k15SwKM9GZXb1o4KY_JpzF_OCIW4RYcMhw/formResponse?usp=pp_url&entry.179783227={{payload.metric}}&entry.1383307148={{payload.value}}&entry.981248842={{payload.tag}})

```

6/25/2021, 9:14:00 AM node: 5a8843a7.e408ac
msg.payload : Object
  ▶ { metric: "t01", value: 26, tag: "m01" }

6/25/2021, 9:14:07 AM node: 5a8843a7.e408ac
msg.payload : Object
  ▶ { metric: "t01", value: 22, tag: "m01" }
  
```

lotInput (Responses)

	A	B	C	D
1	Timestamp	metric	value	tag
2	6/25/2021 9:11:47	t01	25	m01
3	6/25/2021 9:14:02	t01	26	m01
4	6/25/2021 9:14:09	t01	22	m01