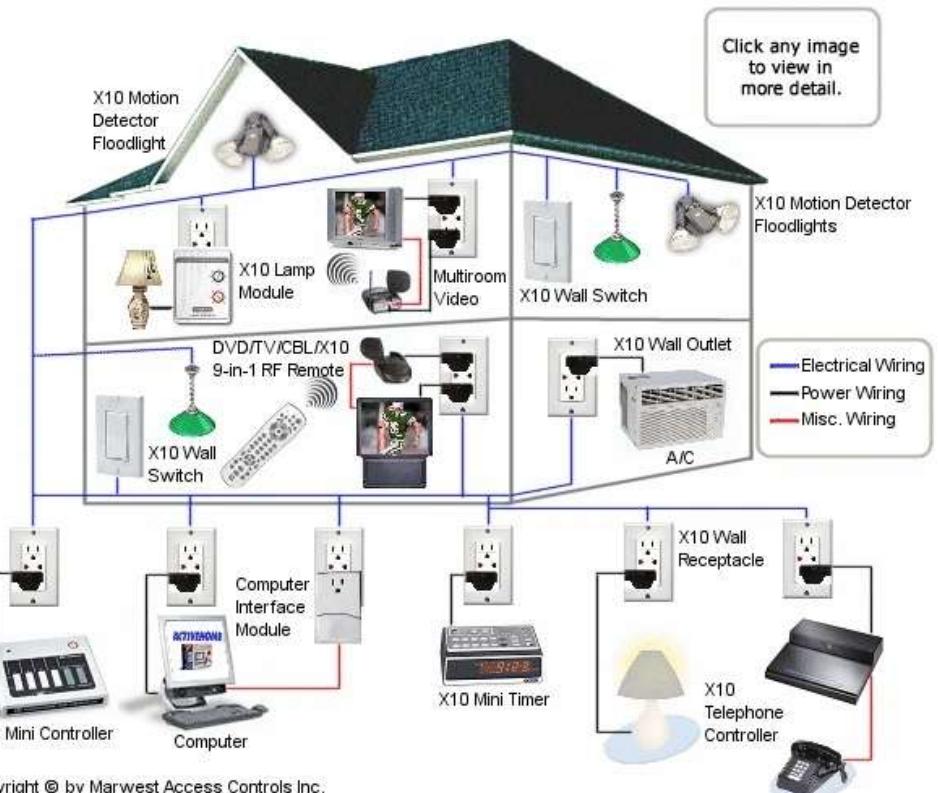
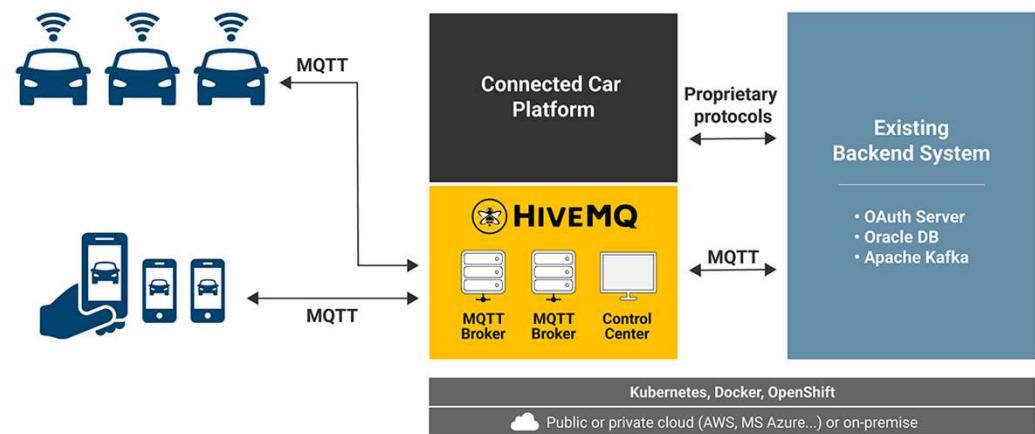


Internet of Things (IoT)

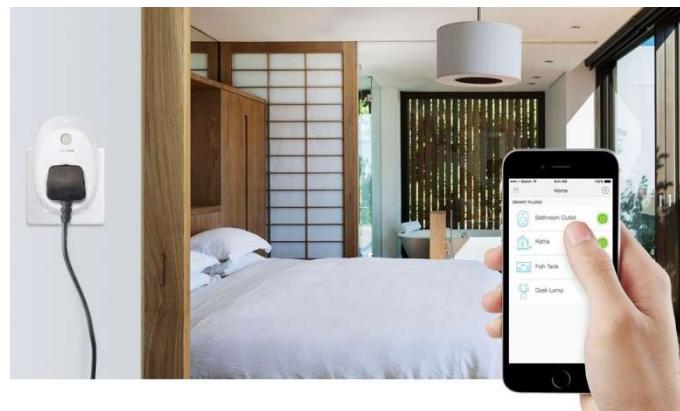
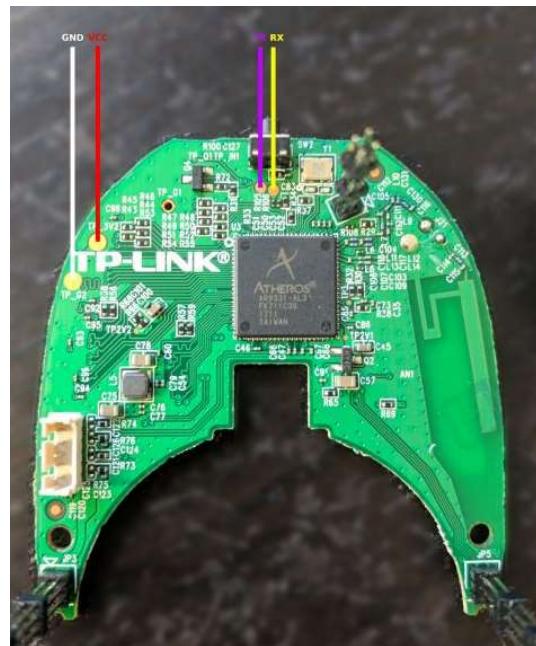
- การที่อุปกรณ์อิเล็กทรอนิกส์ที่สามารถเชื่อมโยงหรือสื่อสารข้อมูลระหว่างอุปกรณ์และระบบคอมพิวเตอร์ ผ่านระบบเครือข่ายภายในและอินเตอร์เน็ต ผ่านทางช่องทางการสื่อสารแบบต่าง ๆ เช่น Wireless ,Lan ,Wifi ,NB-IOT,Lora, 3G ,4G ,5G เช่น
 - Smart Device
 - Smart Grid
 - Smart Home
 - Smart Network
 - Smart Intelligent Transportation

Smart Home and Smart Left (DIY IOT Lab)

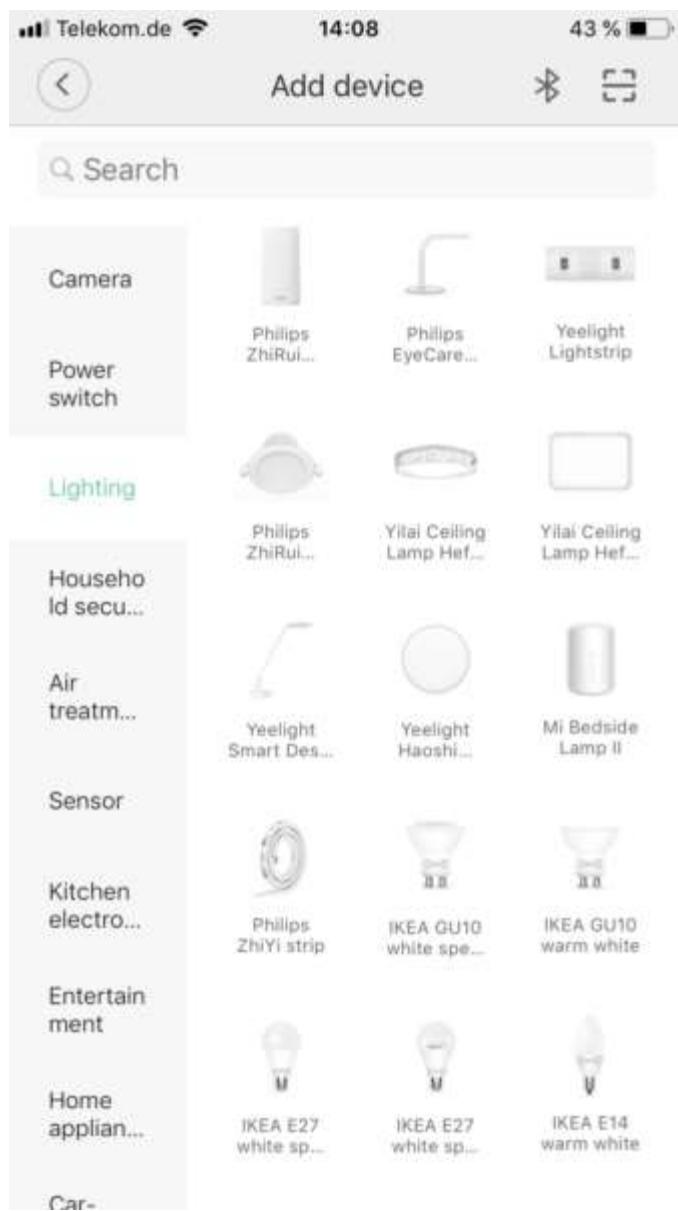


Smart Plug Monitor

โปรแกรมควบคุมปลั๊กไฟ TP-Link HS110



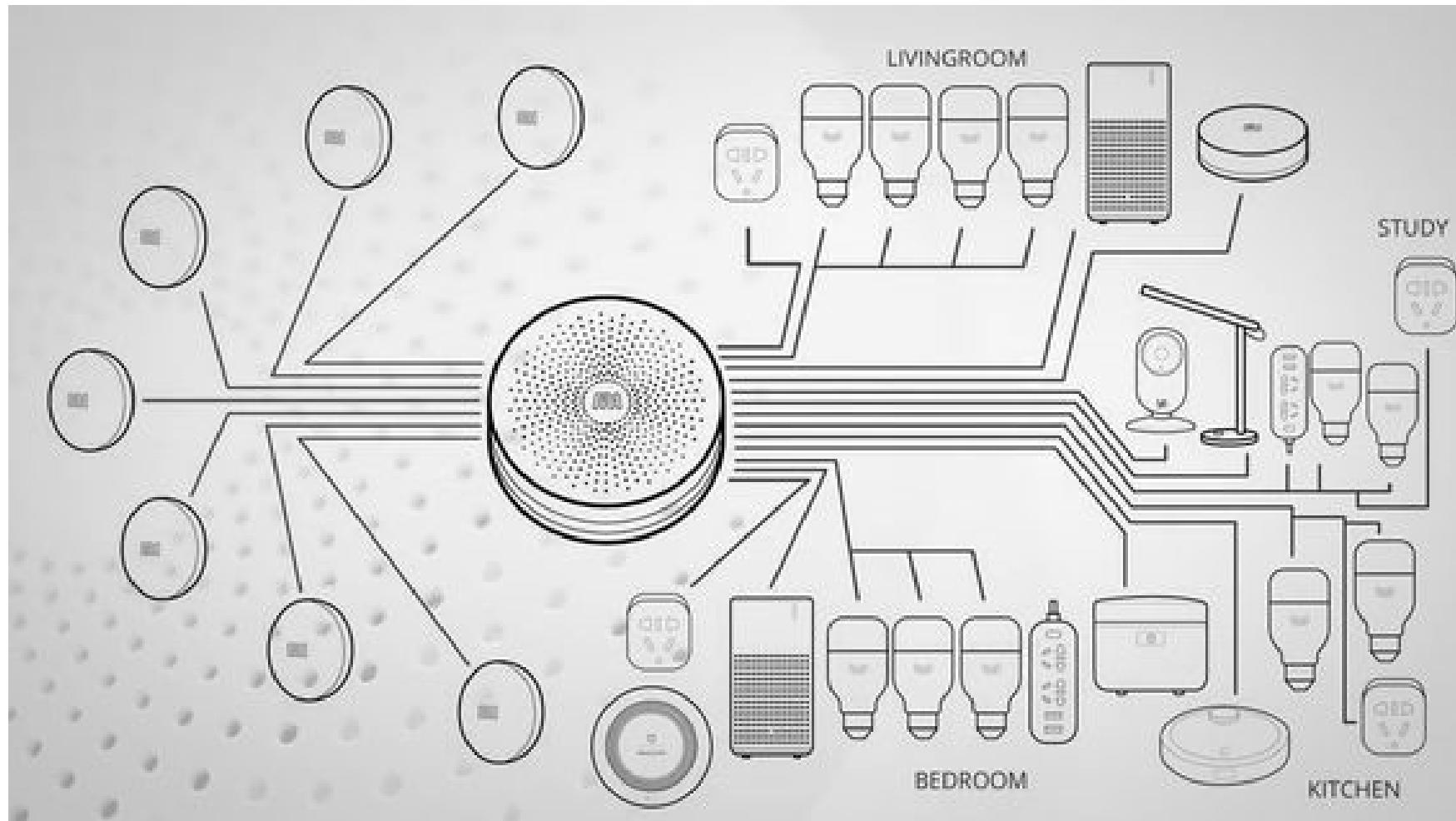
Smart home หรือ บ้านอัจฉริยะ



- โดยทั่วไปเรียกว่า **home automation** ซึ่งสามารถถูกจำแนกความสามารถและความซับซ้อนในการควบคุมออกเป็น
 - ระบบควบคุมไฟฟ้าแสงสว่าง เช่น เปิด/ปิด หรือปรับระดับความสว่าง
 - ระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน เช่น สั่งงานเครื่องปรับอากาศ หรือการเปิดปิดม่าน
 - ระบบรักษาความปลอดภัย เช่น เชื่อมต่อระบบกันขโมย/กล้องกับ บ.รักษาความปลอดภัย



Smart home Features

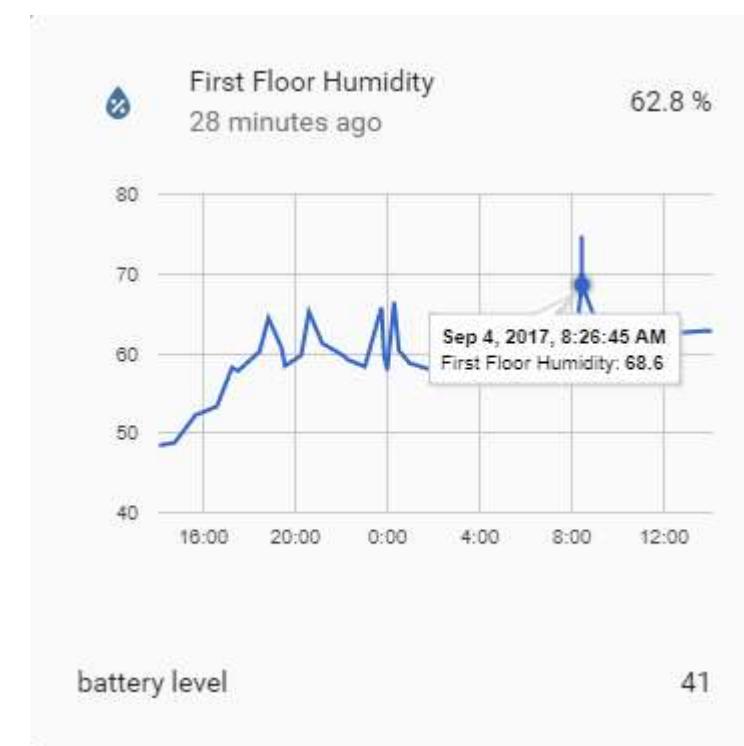
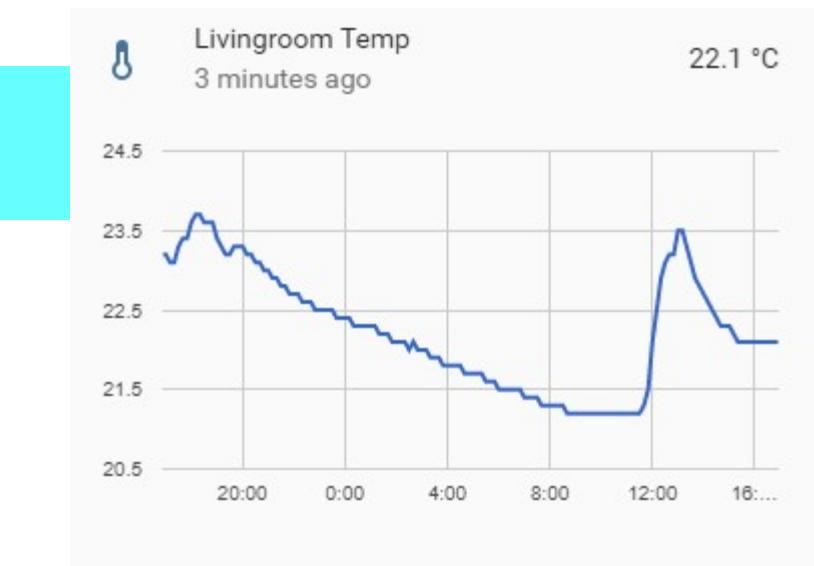


Mi Temperature and Humidity Sensor



Measure things such as temperature, movement, light and sound, while connected with other Xiaomi smart home devices.

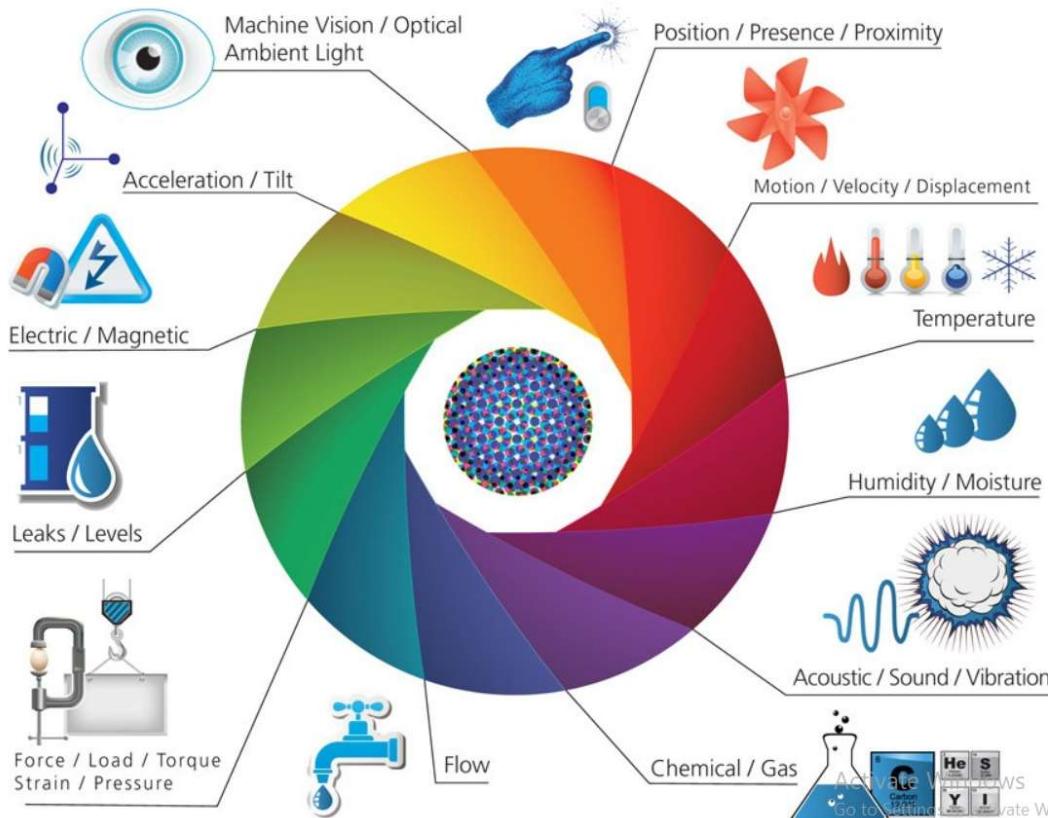




Basic IOT Sensors

Digital Sensors

- On/Off
- High/Low
- True/False
- 0/1



Analog Sensors

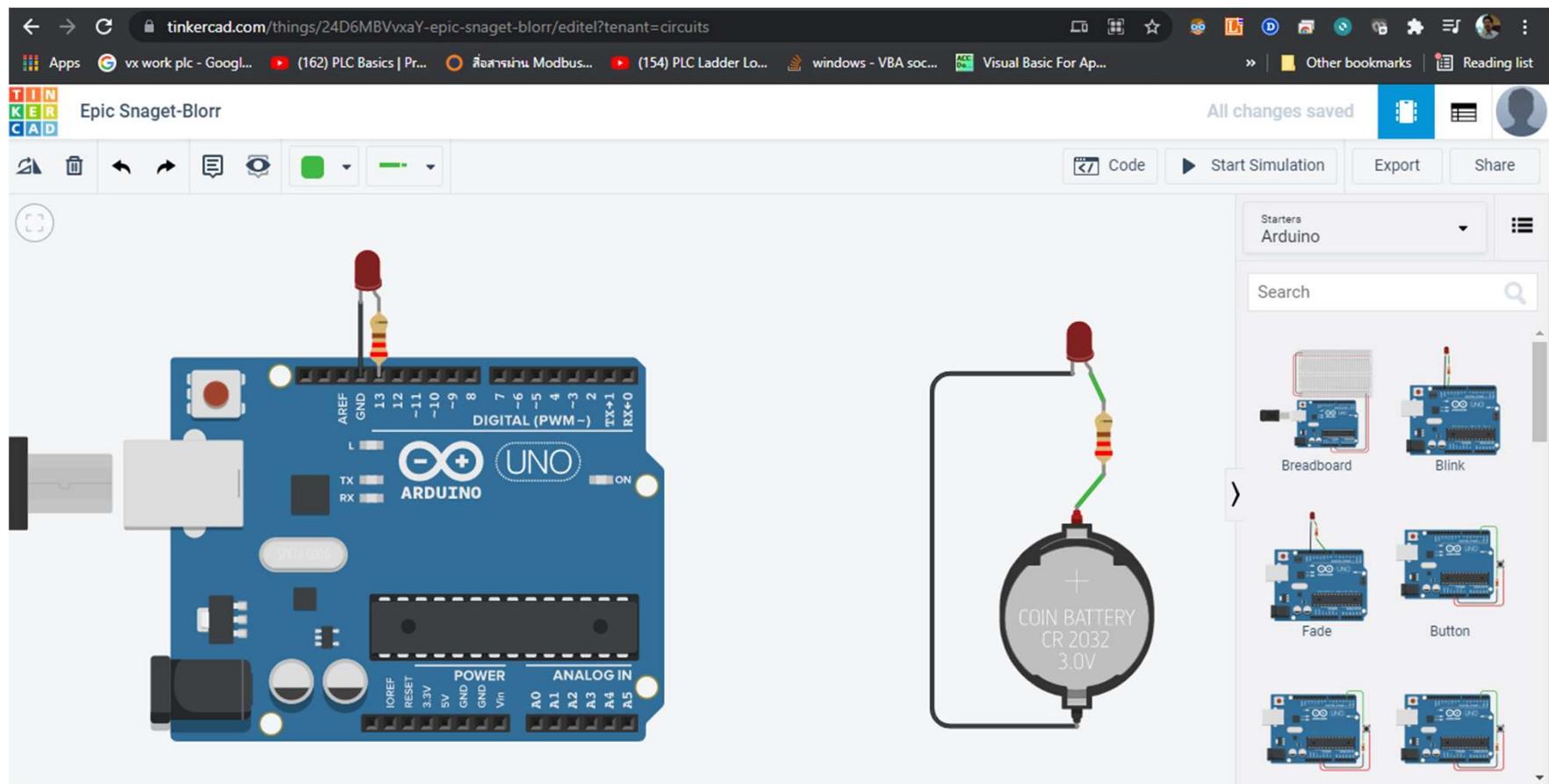
- 0 – 20 mv
- 0 -10 v
- 0 – 65535 (FFFF H)

Different Types of Sensors

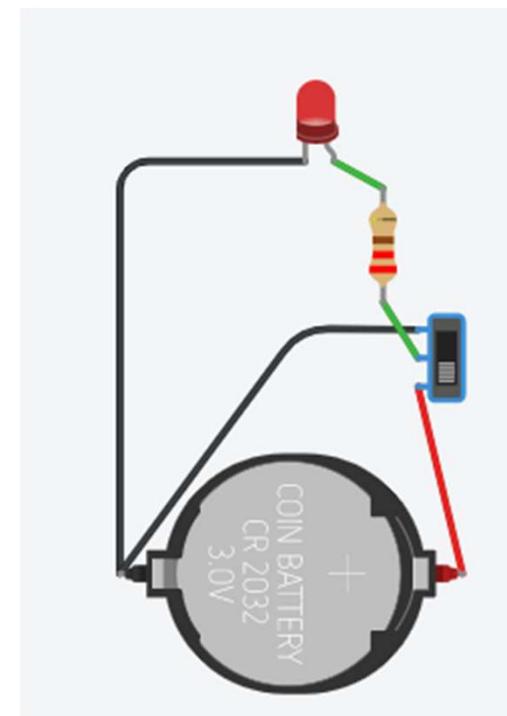
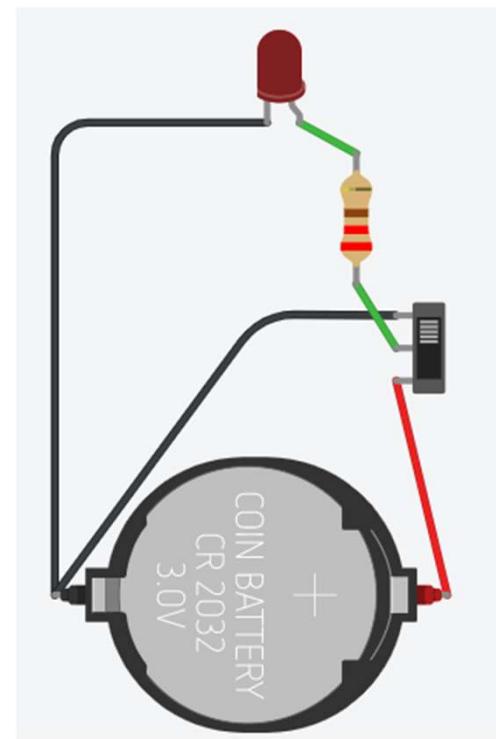
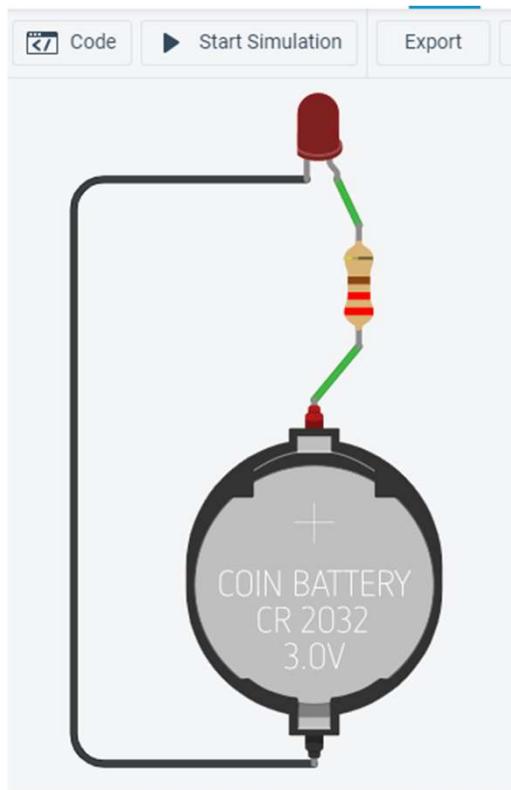


Basic IOT (Arduino Simulator)

- <https://www.tinkercad.com/dashboard>



Basic LED Light UP Circuit Control



IOT Basic Programming (On/Off LED)

The screenshot shows the Tinkercad interface for a breadboard setup and a Scratch-like code editor.

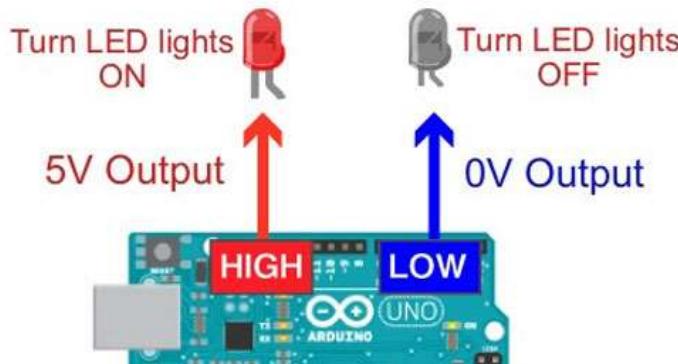
Breadboard Setup: An Arduino Uno R3 board is connected to a breadboard. A red LED is connected in series with a 220 ohm resistor between digital pin 13 and ground. The Arduino's 5V power is connected to the breadboard's positive rail, and the ground connection is also present.

Code Editor: The code editor displays a Scratch script for an Arduino Uno R3. The script consists of the following blocks:

- set built-in LED to HIGH
- set pin 0 to HIGH
- set pin 3 to 0
- rotate servo on pin 0 to 0 degrees
- play speaker on pin 0 with tone 6
- turn off speaker on pin 0
- print to serial monitor "hello world" with
- comment: turn the LED on (HIGH is the voltage level)
- comment: turn the LED off by making the voltage LOW

The code editor includes tabs for Code, Start Simulation, Export, and Share, and a status bar indicating "All changes saved".

Toggle between two different states

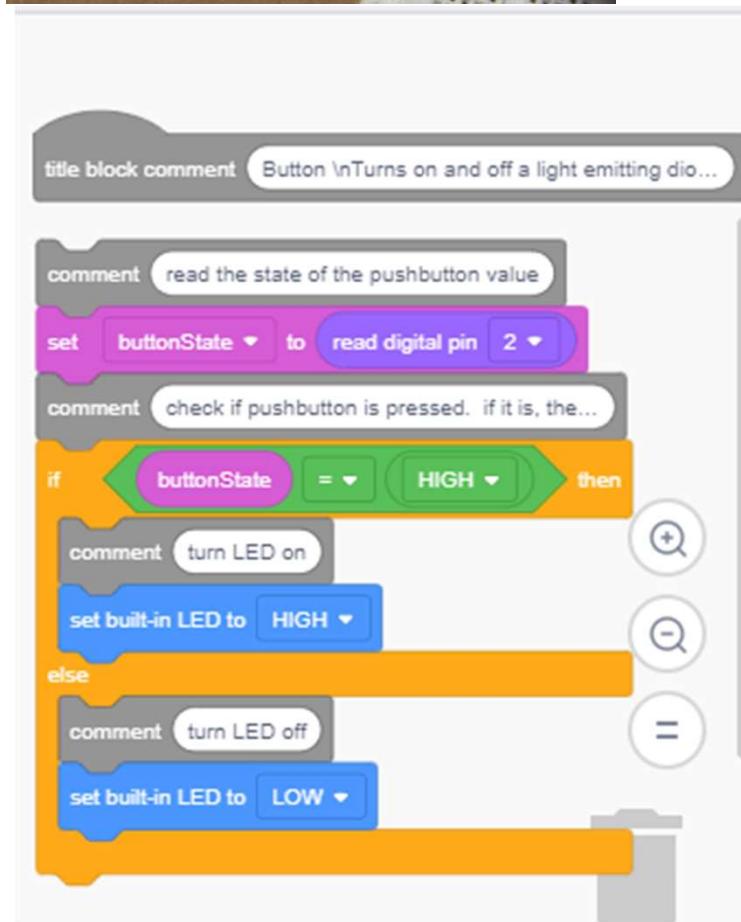
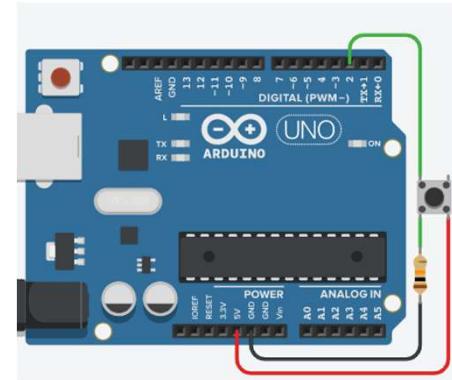
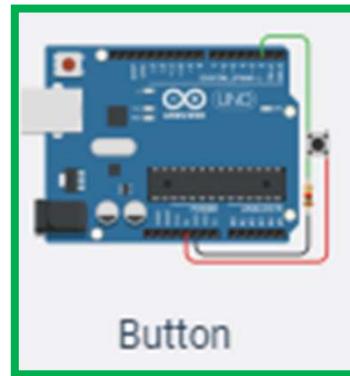
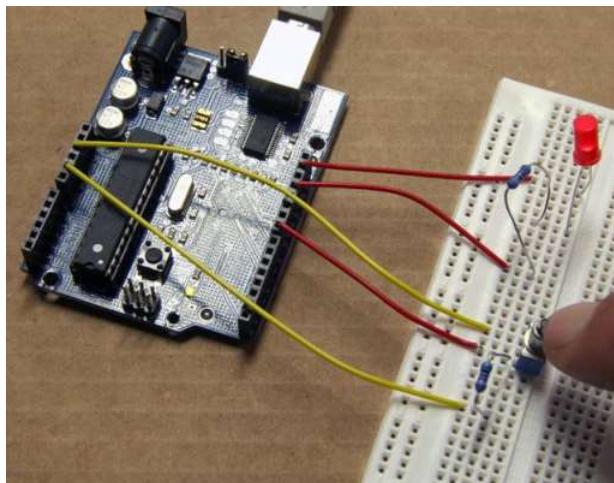


```
void setup() {  
    pinMode( 13 , OUTPUT );  
}  
  
void loop() {  
    digitalWrite( 13 , HIGH );  
}
```



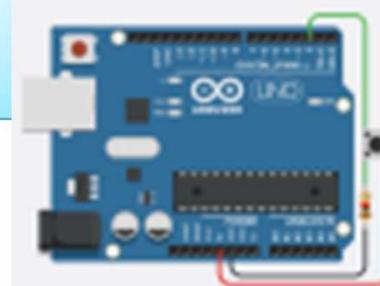
```
1 // C++ code  
2 //  
3 /*  
4     This program blinks pin 13 of the Arduino (the  
5     built-in LED)  
6 */  
7  
8 void setup()  
9 {  
10     pinMode(13, OUTPUT);  
11 }  
12  
13 void loop()  
14 {  
15     // turn the LED on (HIGH is the voltage level)  
16     digitalWrite(13, HIGH);  
17     delay(1000); // Wait for 1000 millisecond(s)  
18     // turn the LED off by making the voltage LOW  
19     digitalWrite(13, LOW);  
20     delay(1000); // Wait for 1000 millisecond(s)  
21 }
```

YouTube IOT- 03 Arduino Switch



```
20 This example code is in the public domain.
21 http://www.arduino.cc/en/Tutorial/Button
22 */
23
24 int buttonState = 0;
25
26 void setup()
27 {
28     pinMode(2, INPUT);
29     pinMode(13, OUTPUT);
30 }
31
32 void loop()
33 {
34     // read the state of the pushbutton value
35     buttonState = digitalRead(2);
36     // check if pushbutton is pressed. if it is, the
37     // buttonState is HIGH
38     if (buttonState == HIGH) {
39         // turn LED on
40         digitalWrite(13, HIGH);
41     } else {
42         // turn LED off
43         digitalWrite(13, LOW);
44     }
45     delay(10); // Delay a little bit to improve simula
```

Basic Digital Switch Sensor



Blocks + Text ▾

Output Control
Input Math
Notation Variables

title block comment DigitalReadSerial \n\nReads a digital input on pin 2 and prints it to the serial monitor.

comment read the input pin

set buttonState to read digital pin 2

comment print out the state of the button

print to serial monitor buttonState with newline

```
int buttonState = 0;

void setup()
{
    pinMode(2, INPUT);
    Serial.begin(9600);
}

void loop()
{
    // read the input pin
    buttonState = digitalRead(2);
    // print out the state of the
    Serial.println(buttonState);
    delay(10); // Delay a little b
}
```

Serial Monitor

0
0
0
0
0
0
0
0

Send Clear

Basic Analog Distance Sensor

TIN KER CAD Swanky Bruticus All changes saved

Code Stop Simulation Export Share 1 (Arduino Uno R3)

Simulator time: 00:01:26.085

Ultrasonic Distance Sensor

Name 1

Blocks

- Output
- Control
- Input
- Math
- Notation
- Variables

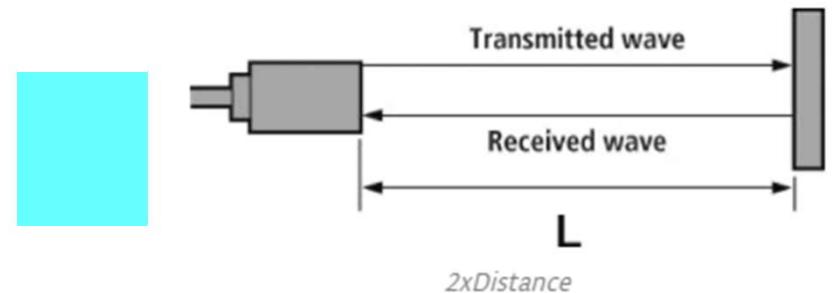
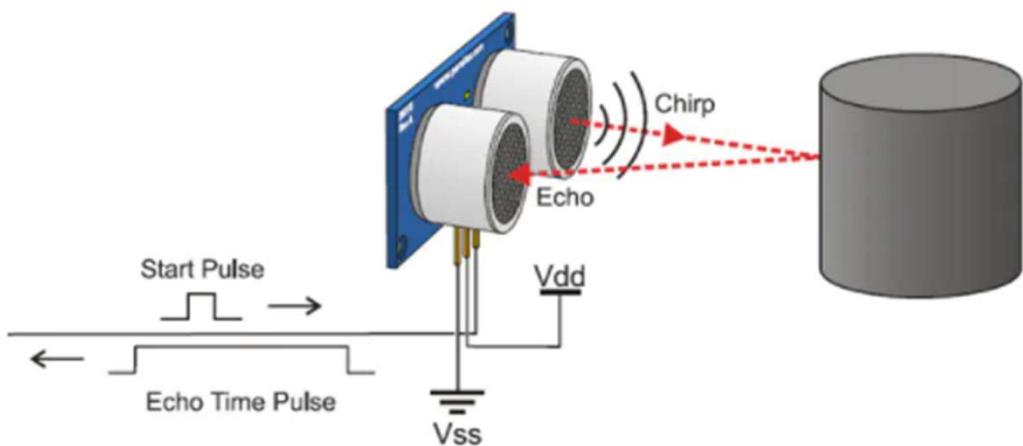
Code (Scratch-like blocks)

```
title block comment (Ping))) Sensor\nInThis sketch reads a PING))) ...  
comment measure the ping time in cm  
set cm to read ultrasonic distance sensor on trigger pin 7 echo pin sam  
comment convert to inches by dividing by 2.54  
set inches to cm / 2.54  
print to serial monitor inches without newline  
print to serial monitor in without newline  
print to serial monitor cm without newline  
print to serial monitor cm with newline  
wait 1 secs
```

Serial Monitor

69in, 176cm
69in, 176cm

Send Clear

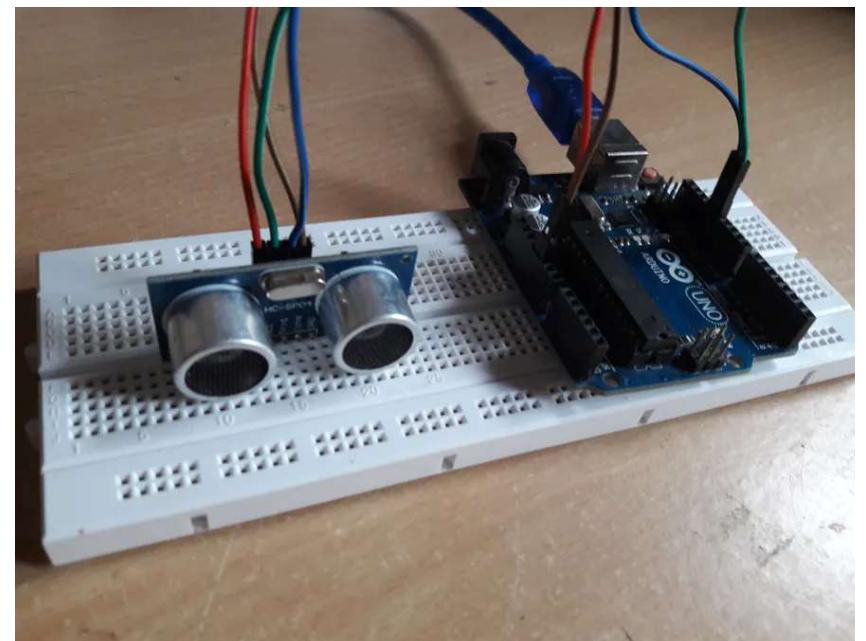


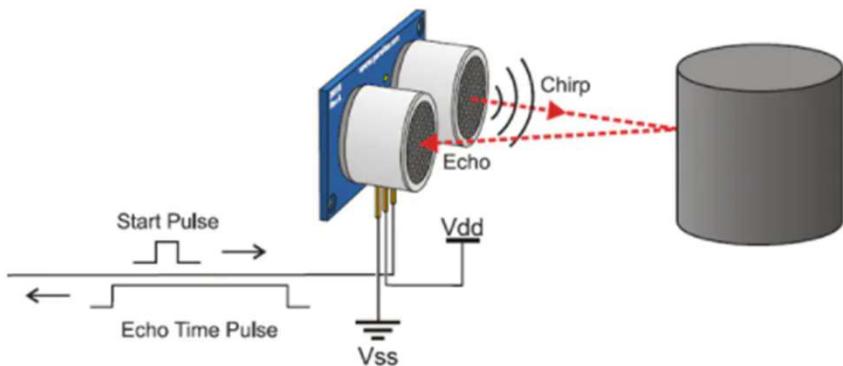
$$\text{Distance} = (\text{Time}/2) \times \text{Speed}$$

$$\text{Distance} = (\text{Time}/2) \times 29.1$$



7. Arduino Radar (Sonar)





SR-04F Working

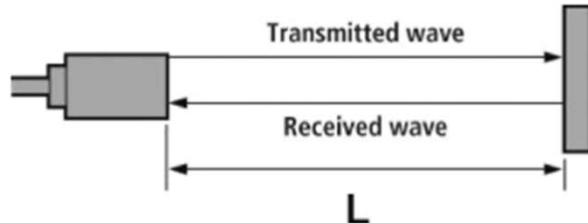
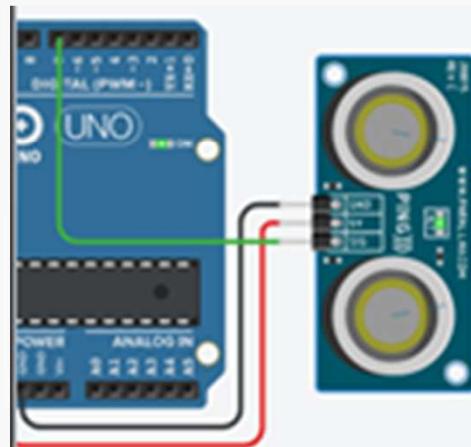
title block comment Ping))) SensorIn\nThis sketch reads a PING))) ...

comment measure the ping time in cm

```

set cm ▾ to read ultrasonic distance sensor on trigger pin 7 ▾ echo pin same as trigger ▾ in units cm ▾
comment convert to inches by dividing by 2.54
set inches ▾ to cm / 2.54
print to serial monitor inches without ▾ newline
print to serial monitor in, without ▾ newline
print to serial monitor cm without ▾ newline
print to serial monitor cm with ▾ newline
wait .1 secs

```



$$\text{Distance} = (\text{Time}/2) \times \text{Speed}$$

$$\text{Distance} = (\text{Time}/2) \times 29.1$$

```

24 int inches = 0;
25
26 int cm = 0;
27
28 long readUltrasonicDistance(int triggerPin, int echo
29 {
30     pinMode(triggerPin, OUTPUT); // Clear the trigger
31     digitalWrite(triggerPin, LOW);
32     delayMicroseconds(2);
33     // Sets the trigger pin to HIGH state for 10 micro
34     digitalWrite(triggerPin, HIGH);
35     delayMicroseconds(10);
36     digitalWrite(triggerPin, LOW);
37     pinMode(echoPin, INPUT);
38     // Reads the echo pin, and returns the sound wave
39     return pulseIn(echoPin, HIGH);
40 }
41
42 void setup()
43 {
44     Serial.begin(9600);
45 }
46
47
48 void loop()
49 {
50     // measure the ping time in cm
51     cm = 0.01723 * readUltrasonicDistance(7, 7);
52     // convert to inches by dividing by 2.54
53     inches = (cm / 2.54);
54     Serial.print(inches);
55     Serial.print("in, ");
56     Serial.print(cm);
57     Serial.println("cm");
58     delay(100); // Wait for 100 millisecond(s)
59 }

```

YouTube IOT- 05 Arduino Distance Sensor

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    // measure the ping time in cm
    cm = 0.01723 * readUltrasonicDistance(7, 7);
    // convert to inches by dividing by 2.54

    inches = (cm / 2.54);
    Serial.print(inches);
    Serial.print("in, ");
    Serial.print(cm);
    Serial.println("cm");
    delay(100); // Wait for 100 millisecond(s)
}
```

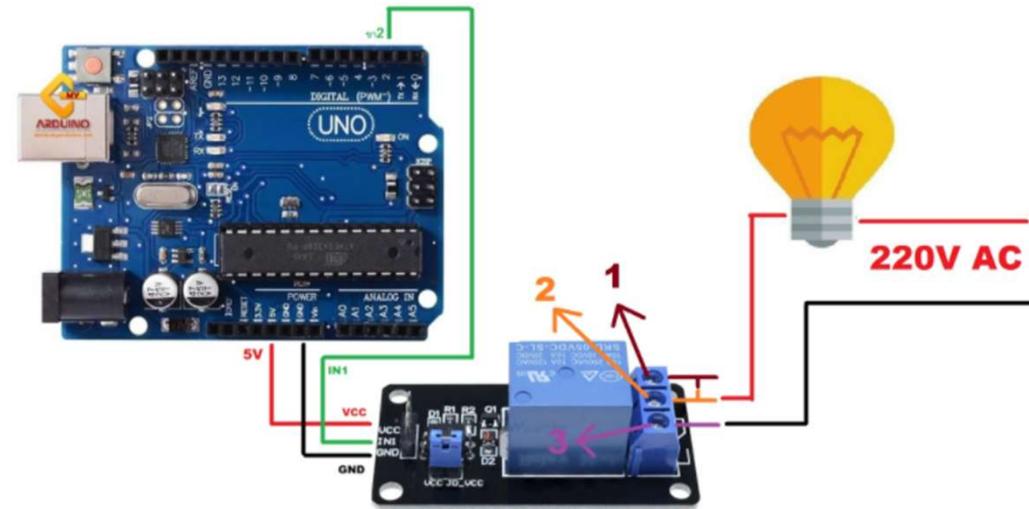
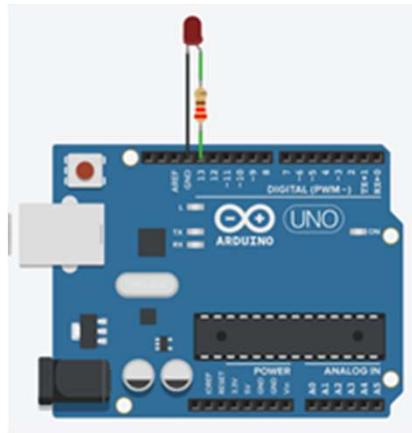
```
// measure the ping time in cm
int sv = readUltrasonicDistance(7, 7);
cm = 0.01723 * sv;
Serial.print("sv = ");
Serial.print(sv);
Serial.print(", ");
// convert to inches by dividing by 2.54
```

The screenshot shows the Arduino IDE interface. The top bar has tabs for 'Text' and other options like download, upload, and verify. The code area displays the provided Arduino sketch. A red box highlights the section of code from line 51 to line 63, which handles the conversion of centimeters to inches. The bottom part of the interface shows the 'Serial Monitor' window, which displays the output of the printed statements. The output shows repeated measurements of 10244, 69in, and 176cm.

```
47
48 void loop()
49 {
50     // measure the ping time in cm
51     int sv = readUltrasonicDistance(7, 7);
52     cm = 0.01723 * sv;
53     Serial.print("sv = ");
54     Serial.print(sv);
55     Serial.print(", ");
56     // convert to inches by dividing by 2.54
57     inches = (cm / 2.54);
58     Serial.print(inches);
59     Serial.print("in, ");
60     Serial.print(cm);
61     Serial.println("cm");
62     delay(100); // Wait for 100 millisecond(s)
63 }
```

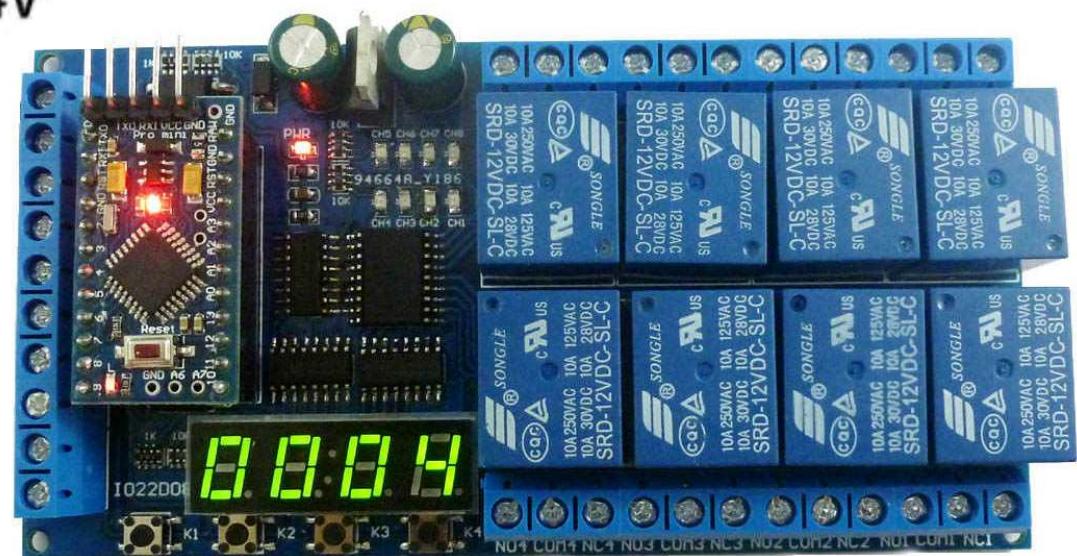
Serial Monitor
sv = 10244 , 69in, 176cm
sv = 10231 , 69in, 176cm
sv = 10233 , 69in, 176cm

- gnd -> GND
- Vcc -> 5V
- In-> ขา2

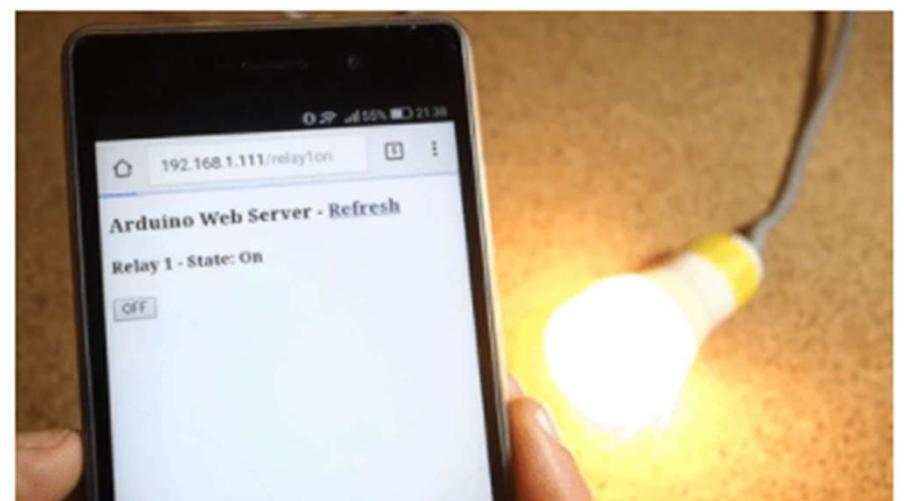
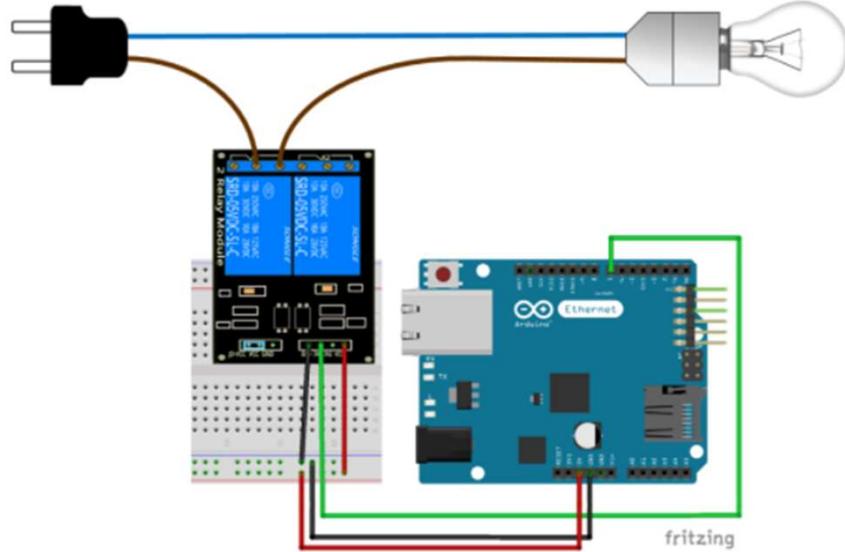


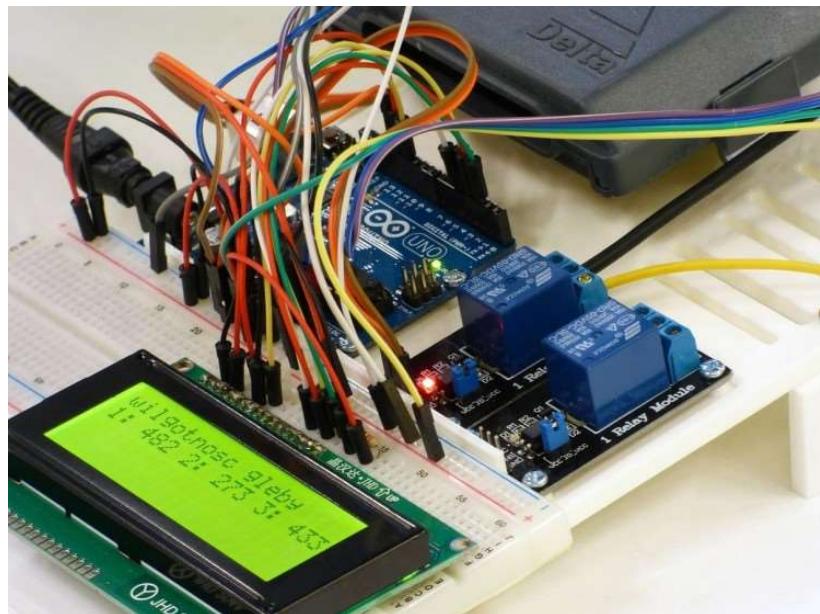


5V 12V 24V

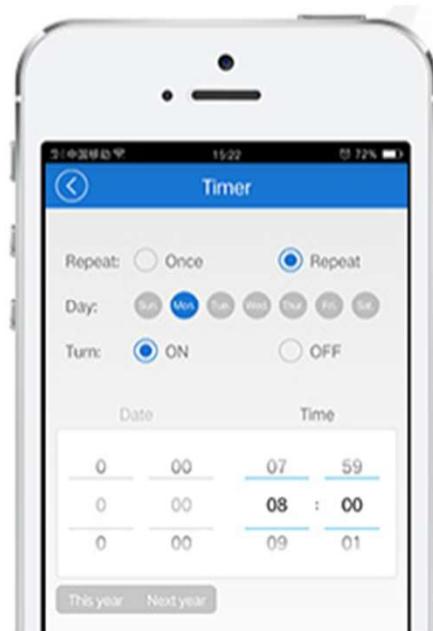
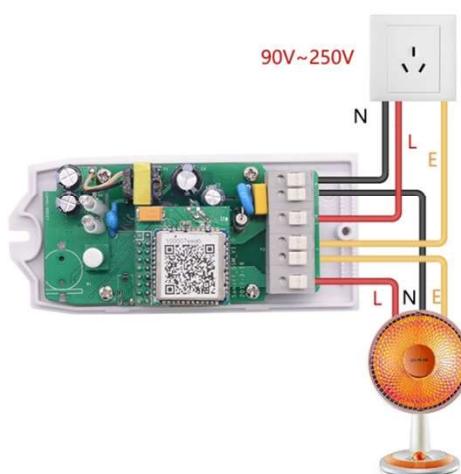
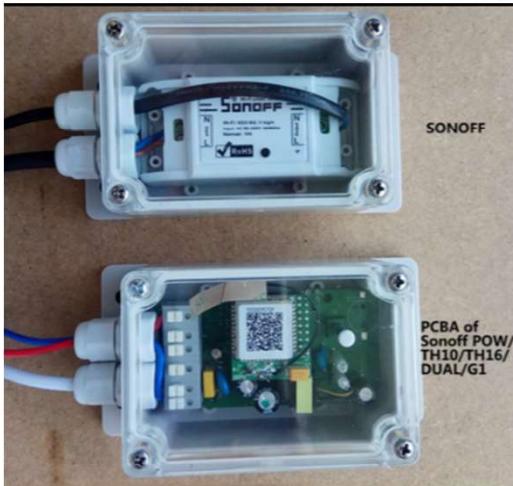


Basic IOT (Smart Home)





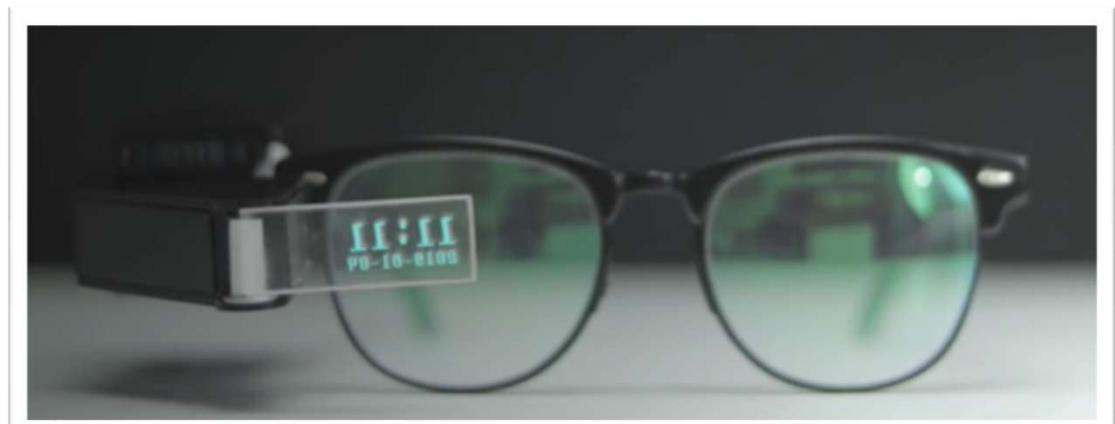
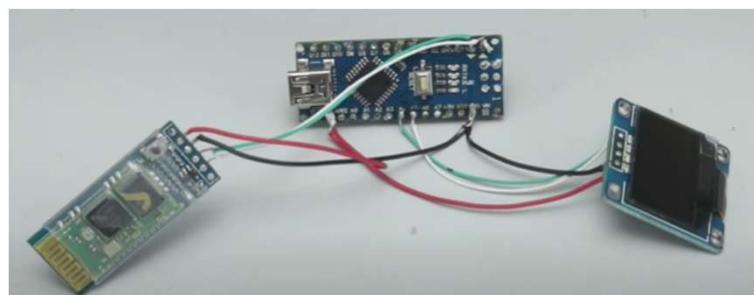
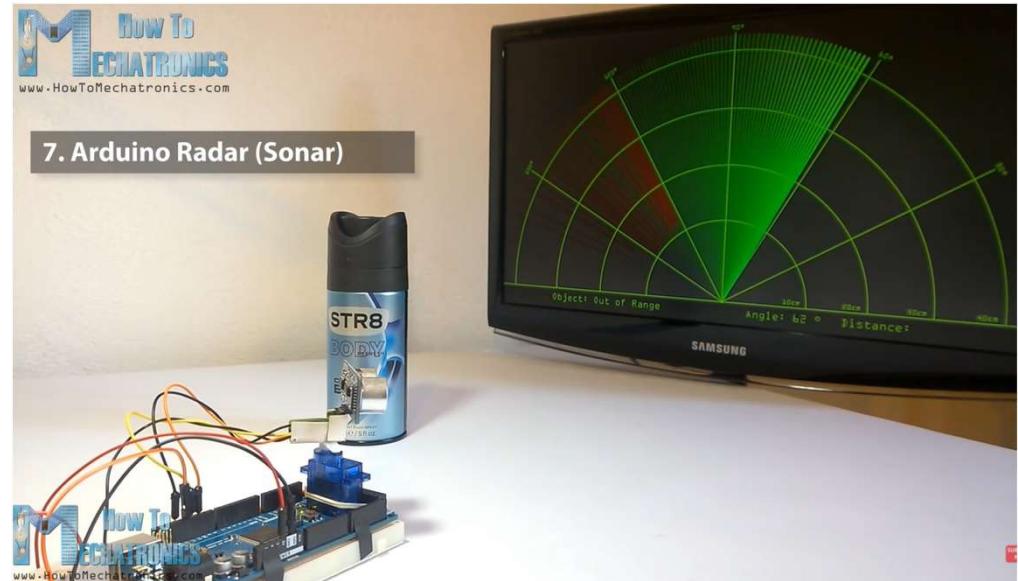
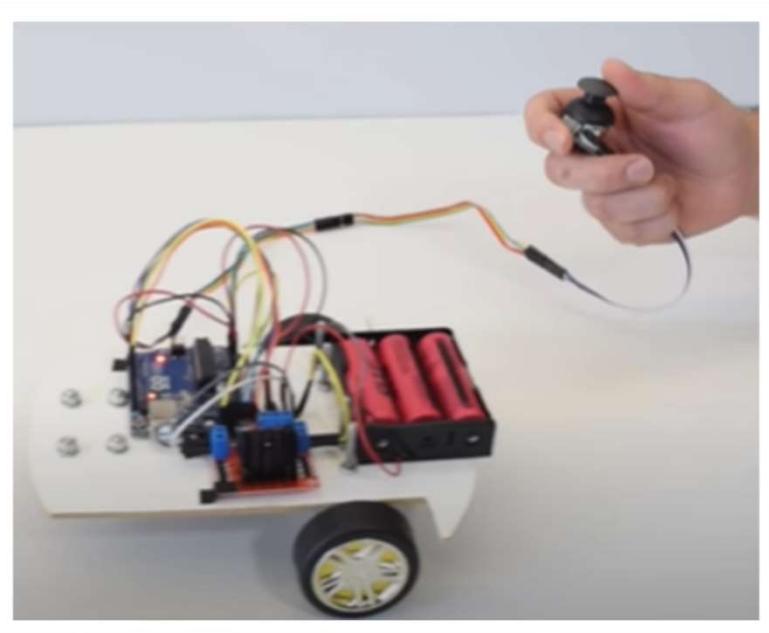
Esp8266

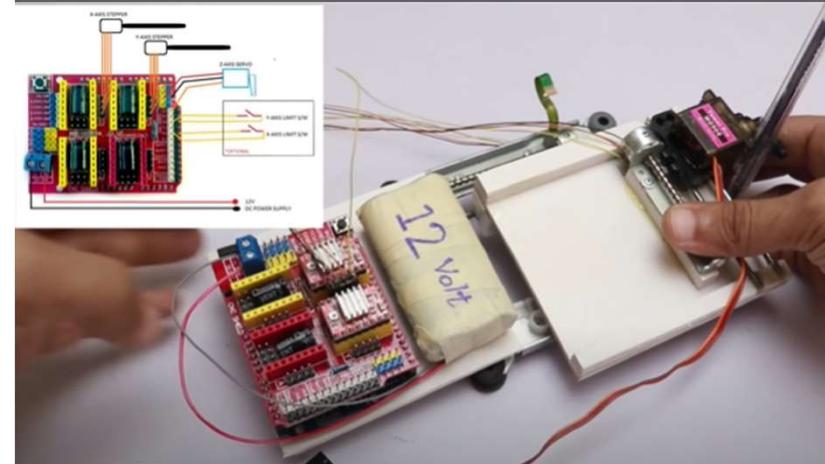
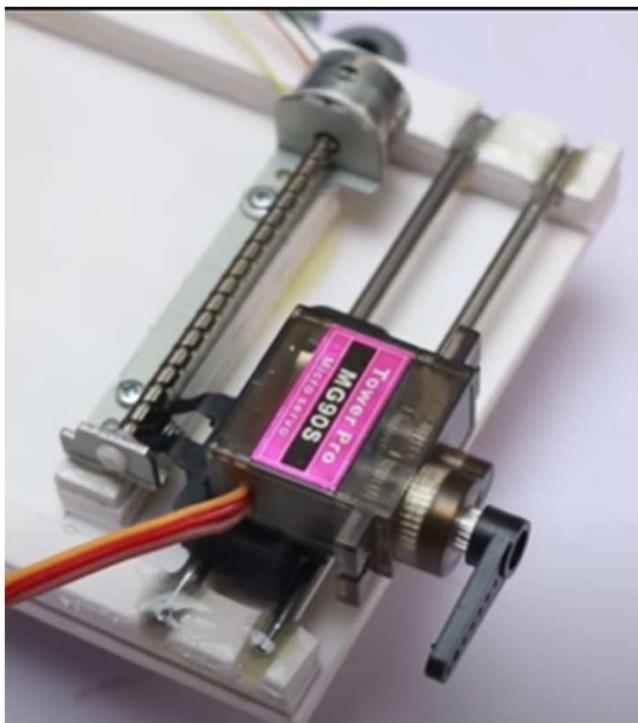
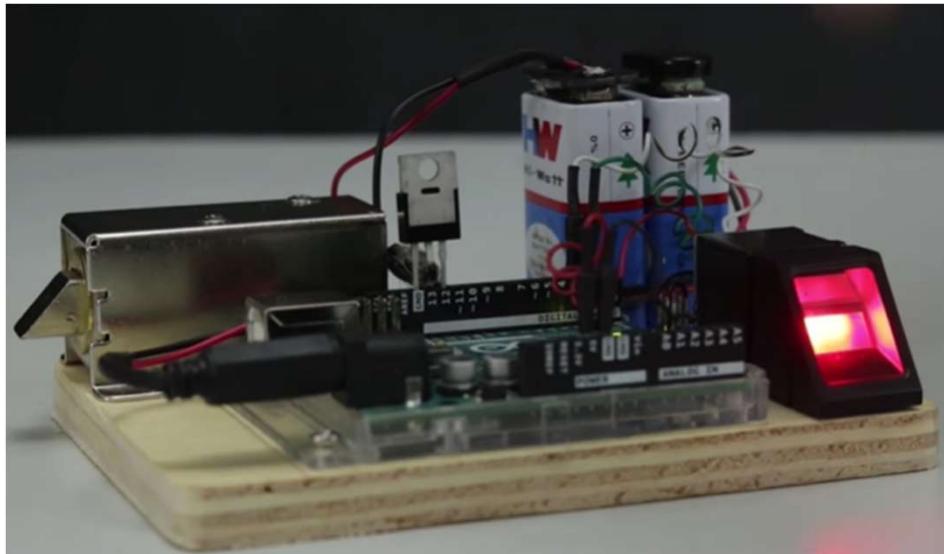


Create
single/repeat/countdown timers
to auto-turn on/off appliances

Lights lit up at 8:00 and auto-off at 23:00 every day

10 Arduino Projects with DIY Step by Step Tutorials









Open Source Controller
(1,500 บาท)



Modix Big-60 Assembly

DIY with same electronic equipments

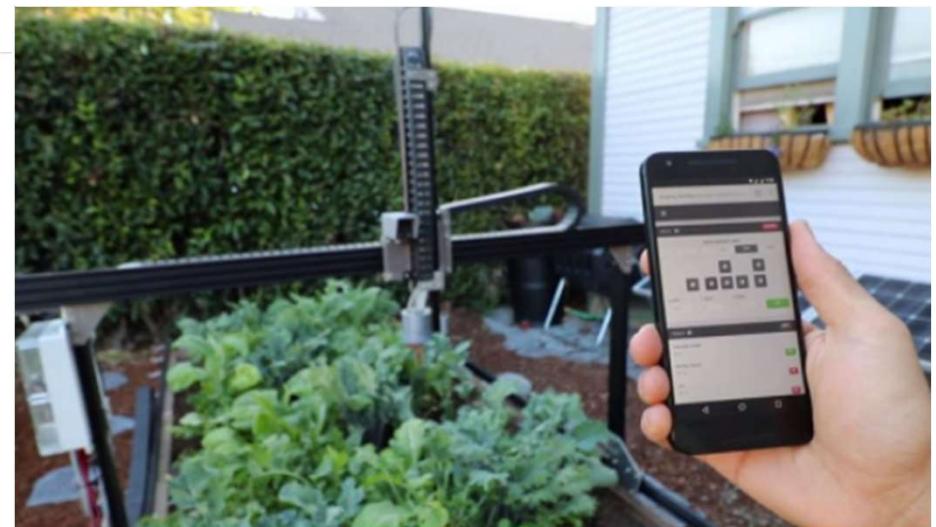
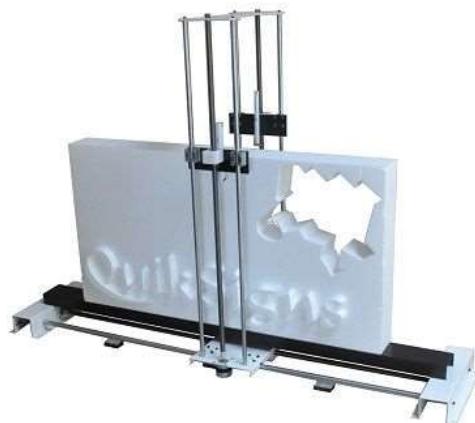
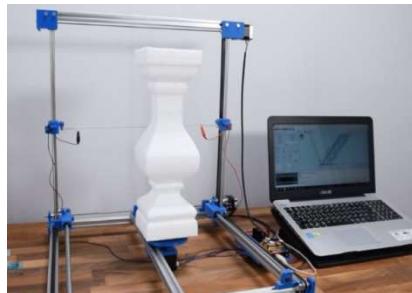


(4,500 บาท)



[ht
DN](#)

นวัตกรรม Open Source ที่สามารถสร้างได้ด้วยตนเอง



VDO Demo

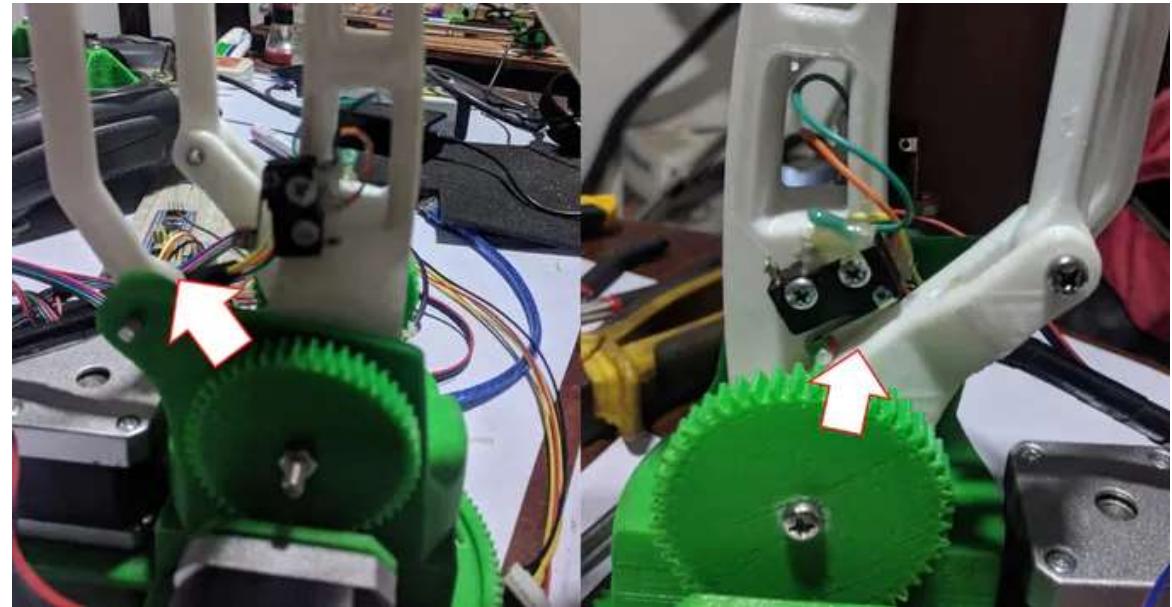
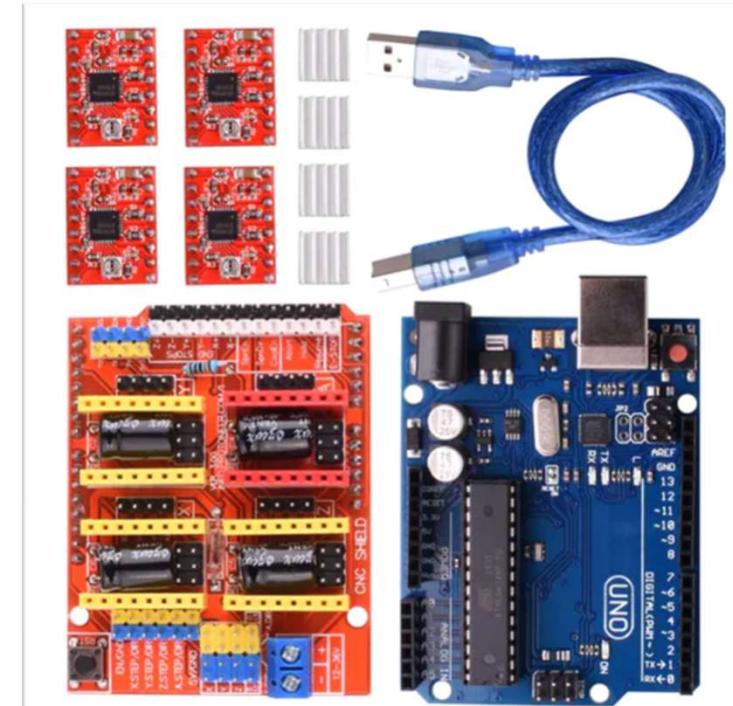
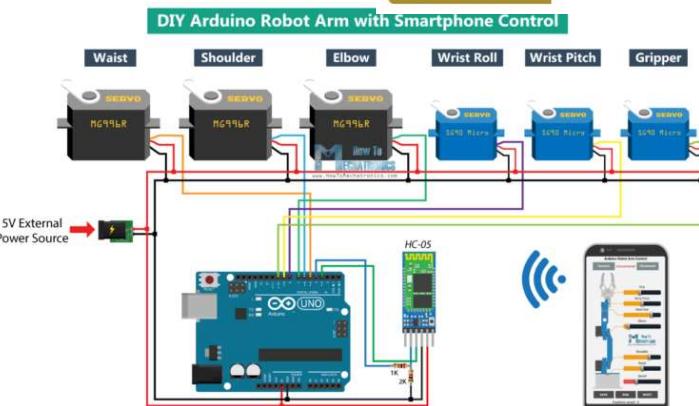
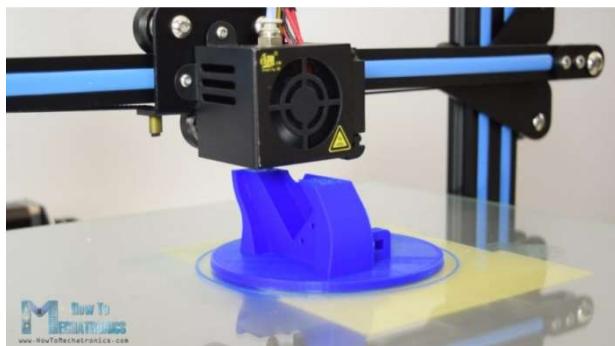
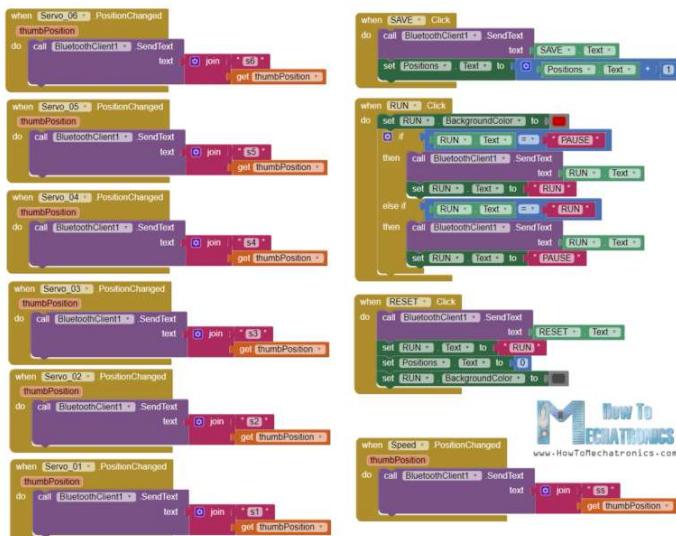
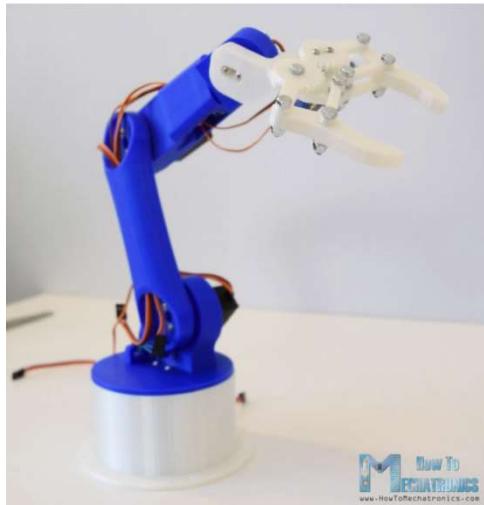


Figure 1 : On the left, the MK2 Plus arm. On the right, the ABB IRB460.



<https://create.arduino.cc/projecthub/yasaspeiris/mk2-plus-robot-arm-controller-458d55>



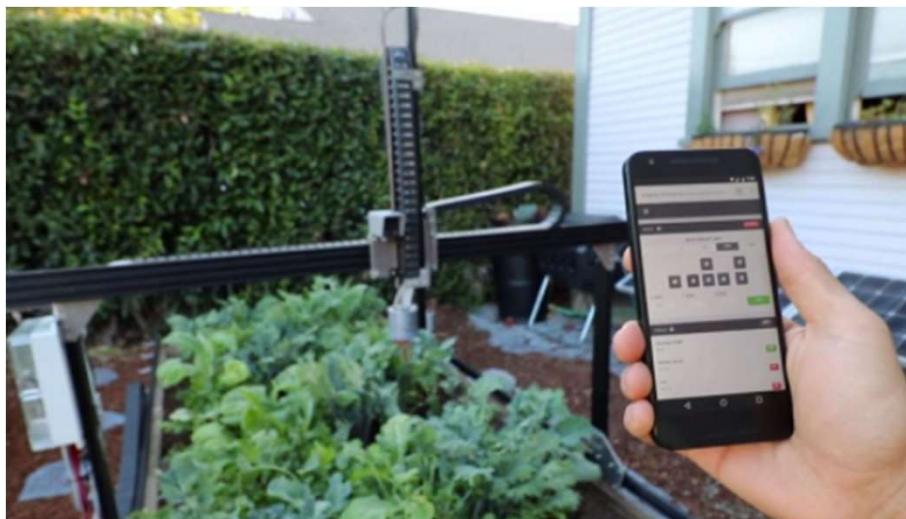
```

SoftwareSerial Bluetooth(3, 4); // Arduino()

int servo1Pos, servo2Pos, servo3Pos, servo4Pos;
int servo1PPos, servo2PPos, servo3PPos, servo4PPos;
int servo01SP[50], servo02SP[50], servo03SP[50];
// for storing positions/steps
  
```



<https://howtomechtronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/>



<https://stemgarden.co/farmbot/>

Open Source Food Technology Lab



เครื่องวาดเพนเคก



เครื่องพิมพ์ สี บนอาหาร



Open Source Food Technology Lab

แสดงตัวอย่างของอุปกรณ์เกิดจากบริษัท Start Up

4สำหรับSelfiesอาหารเครื่องพิมพ์Latteเครื่องพิมพ์อาหารพิมพ์macaron

US \$968.99 ลด 10% ฿76,666 -10%

ส่วนลดชั่วคราว: ลด US \$9.00 ถึง US \$699.00 ~

เพิ่ม US \$2.00 เมื่อซื้อ US \$99.00 การจัดส่งฟรี

สี:

จำนวน:

1 + 200 ชิ้น พิมพ์

การจัดส่ง US \$172.63
ไป Thailand โดยEMS



Padอาหาร 3D เครื่องพิมพ์ข้าวนาโนเย็นน้ำเจล 3D พิมพ์เครื่อง

US \$1,000.00

ส่วนลดชั่วคราว: ลด US \$2.00 ถึง US \$60.00 ~

สี:

red yellow

จำนวน:

1 + 99 ชิ้น พิมพ์ไม่จำกัด

การจัดส่ง US \$99.78
ไป Thailand โดย EMS Standard Shipping

การจัดส่ง โดย EMS ประมาณ 13-20 วัน

ซื้อทันที เพิ่มลงในรถเข็น ♡ 247

การคืนเงินฟรี 90 วัน
การรับประกันศูนย์



พิมพ์สินค้าขนาดเล็กก้อนDeltaเครื่องพิมพ์ อาหาร 3Dเครื่องพิมพ์สำหรับทำกรอบรากฟัน

US \$1,092.50 ลด \$1,150.00 -5%

ส่วนลดชั่วคราว: ลด US \$2.00 ถึง US \$60.00 ~

จำนวน:

1 + 1249 ชิ้น พิมพ์ไม่จำกัด

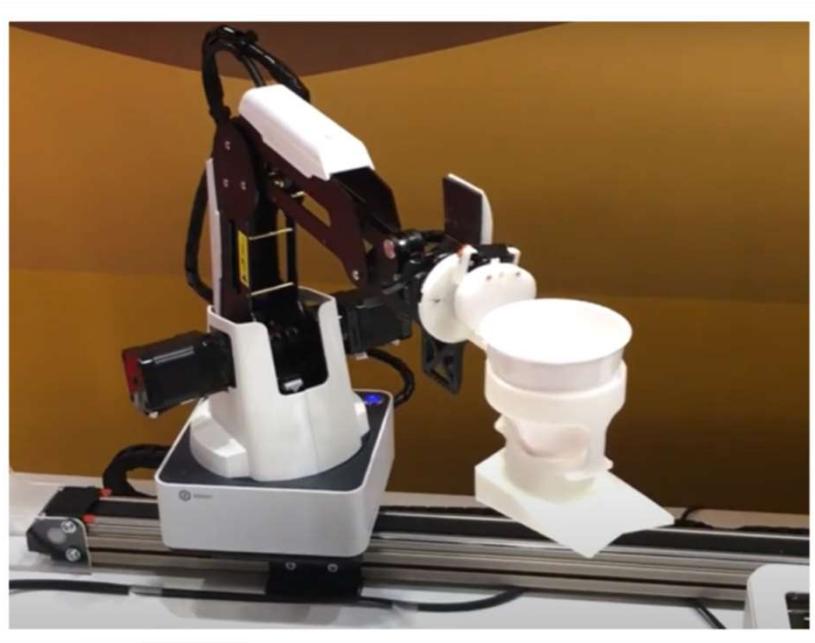
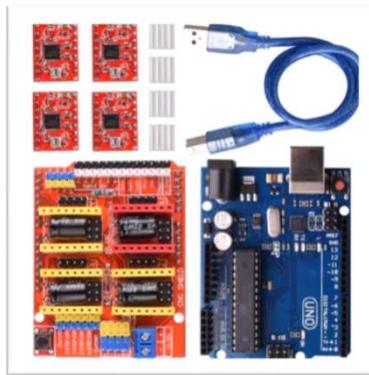
การจัดส่ง โดย EMS DHL

การจัดส่ง โดย EMS ประมาณ 7-15 วัน

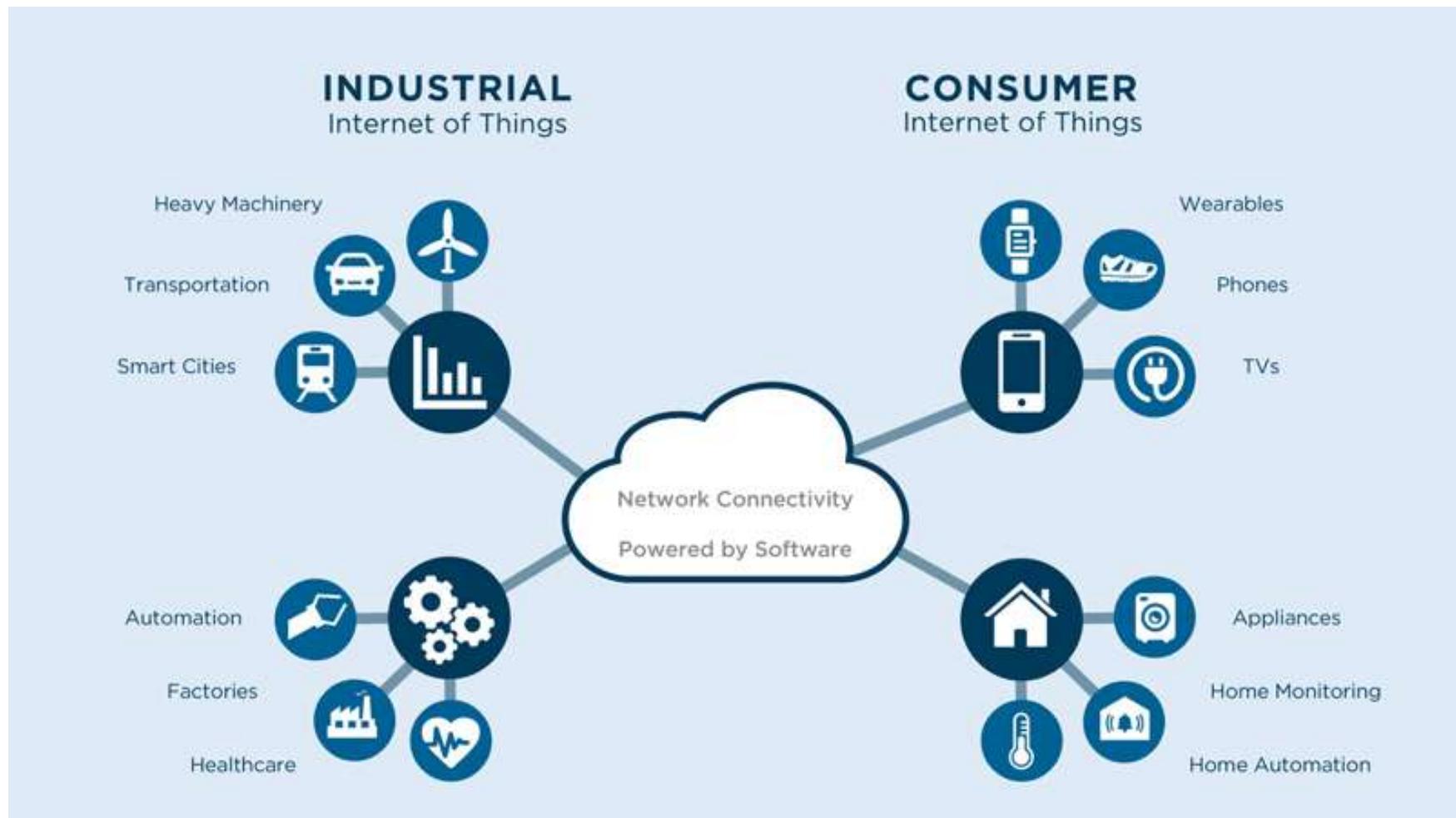
ซื้อทันที เพิ่มลงในรถเข็น ♡ 67

การคืนเงินฟรี 90 วัน
การรับประกันศูนย์





IIOT vs IOT



INDUSTRIAL INTERNET OF THINGS (IIOT)

- 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

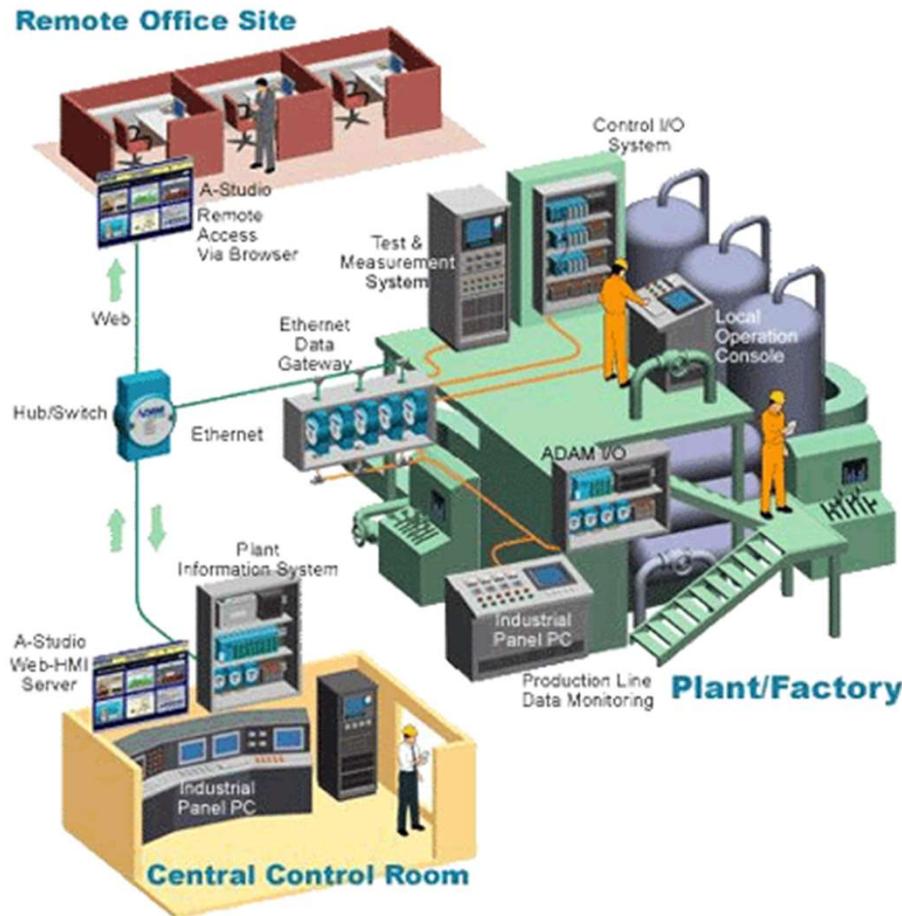
Industrial Automation Control

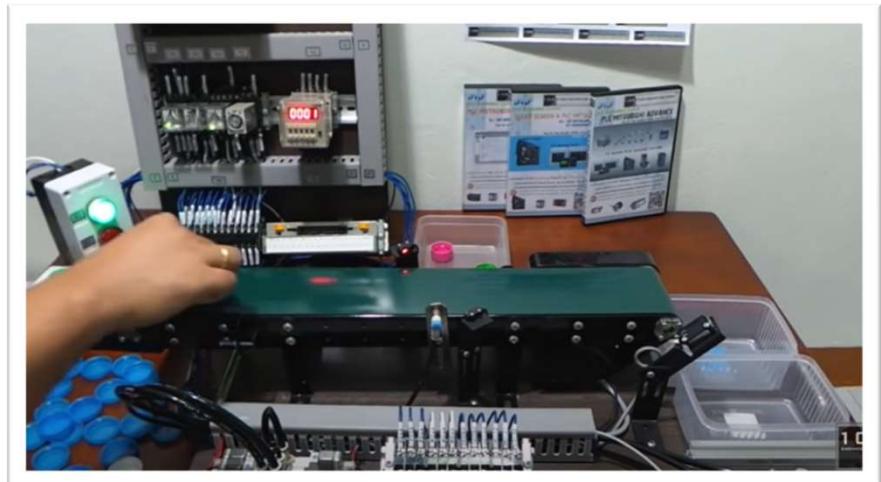
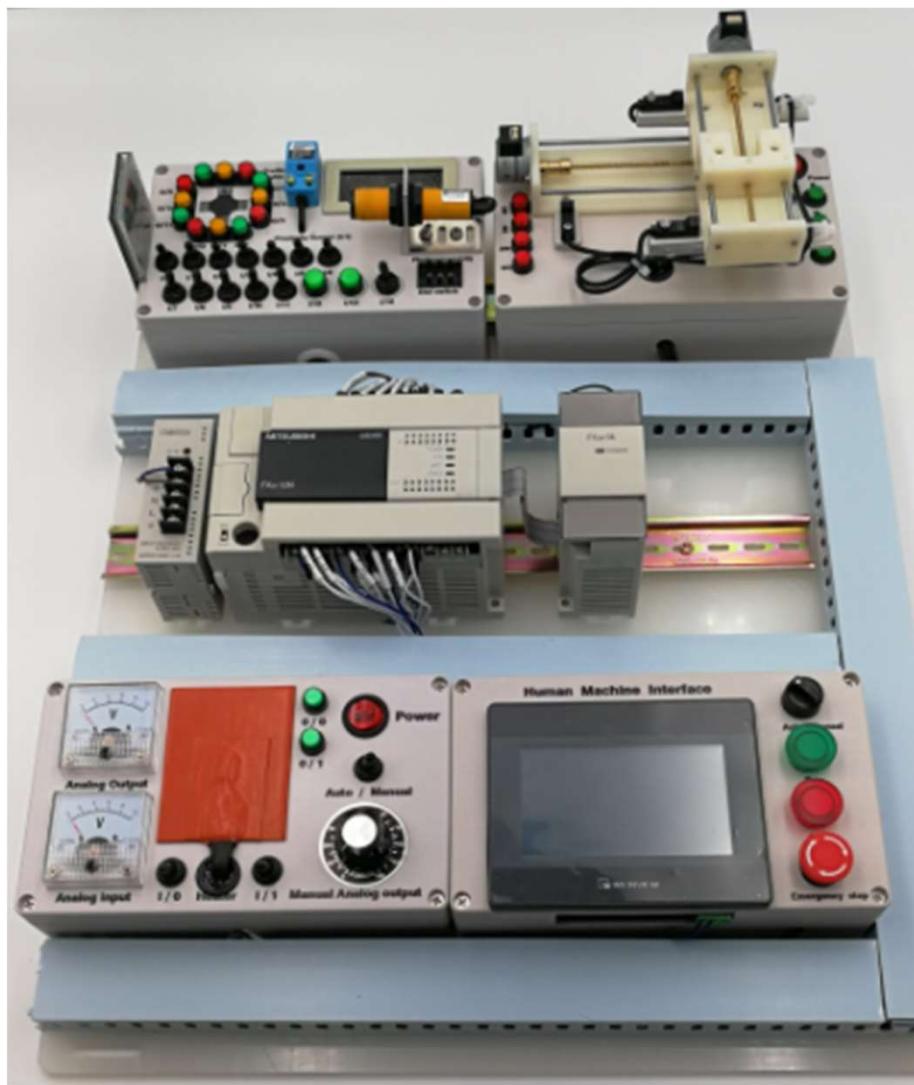
PLC – Programmable Logic Controller

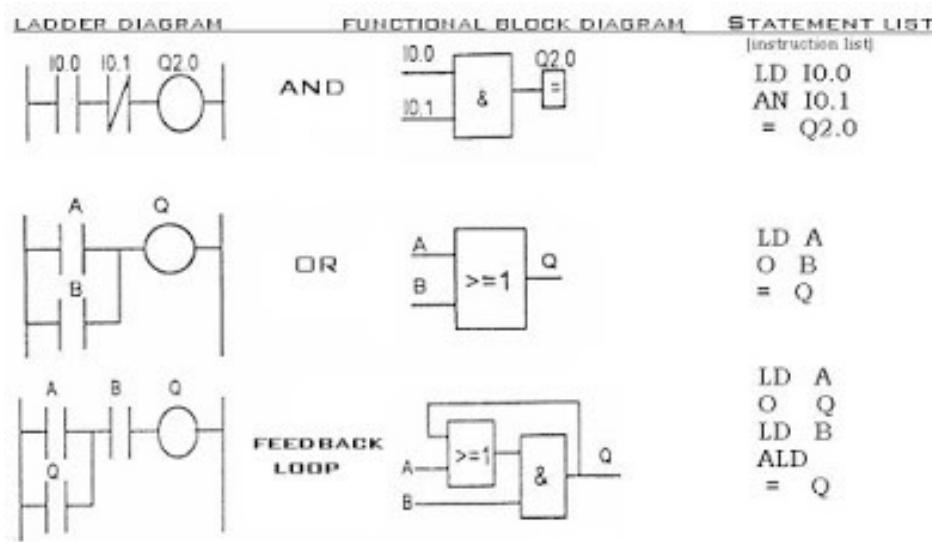
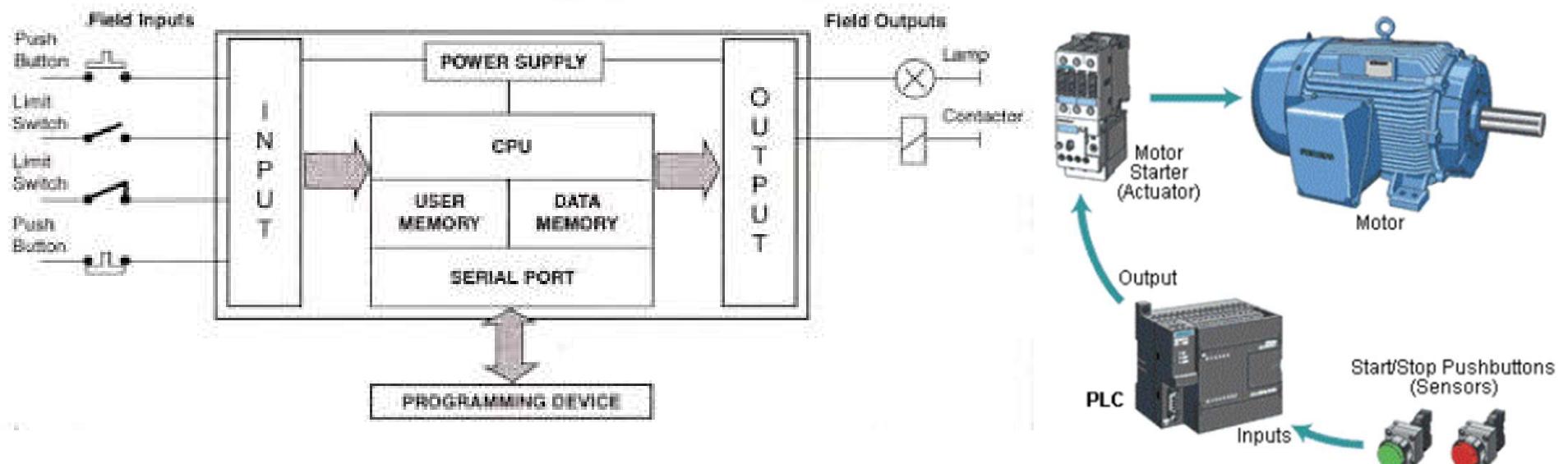
HMI – Human Machine Interface

SCADA – Supervisory Control and Data Acquisition

DCS – Distributed Control System







■ System configuration



■ Main units



- FX3u/FX3UC
- FX3G/FX3GC
- FX3s

■ Special adapters



- Analog I/O
- Communication
- Data collection
- High-speed I/O

■ Expansion units



- I/O extension block
- Analog I/O block
- Temperature control block
- Temperature sensor input block
- Positioning control block
- Communication/network block
- Extension power supply unit

■ Expansion boards



- Communication
- Analog I/O
- 8-point variable analog potentiometer
- Extended I/O
- Special adapter connection

■ Options



- Display module
- Memory cassette
- Battery
- Extension cable
- Conversion adapter

*2: Connectable special adapters, extension units, expansion boards, and other options differ by the models. For details, please refer to the manual of the relevant product.

Digital PLC
Trainer Unit

\$600.00 \$300.00

Analog PLC
Trainer Unit

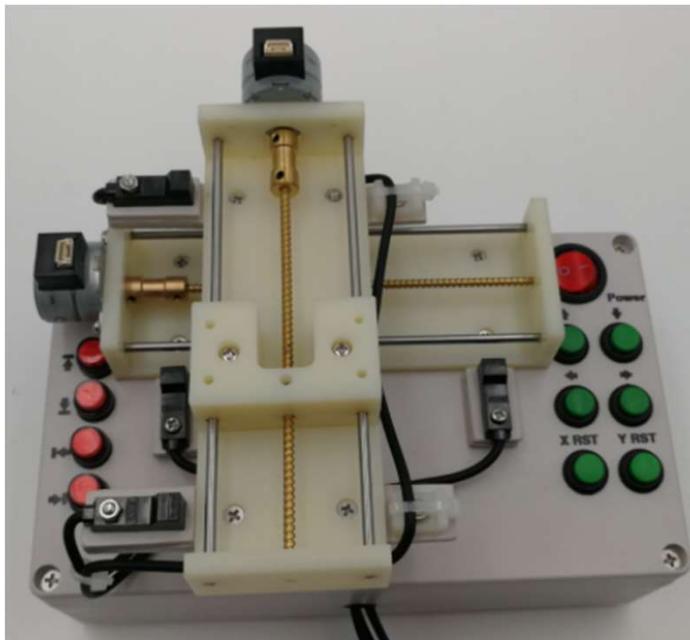
\$600.00 \$300.00

Stepper Position
PLC Trainer Unit

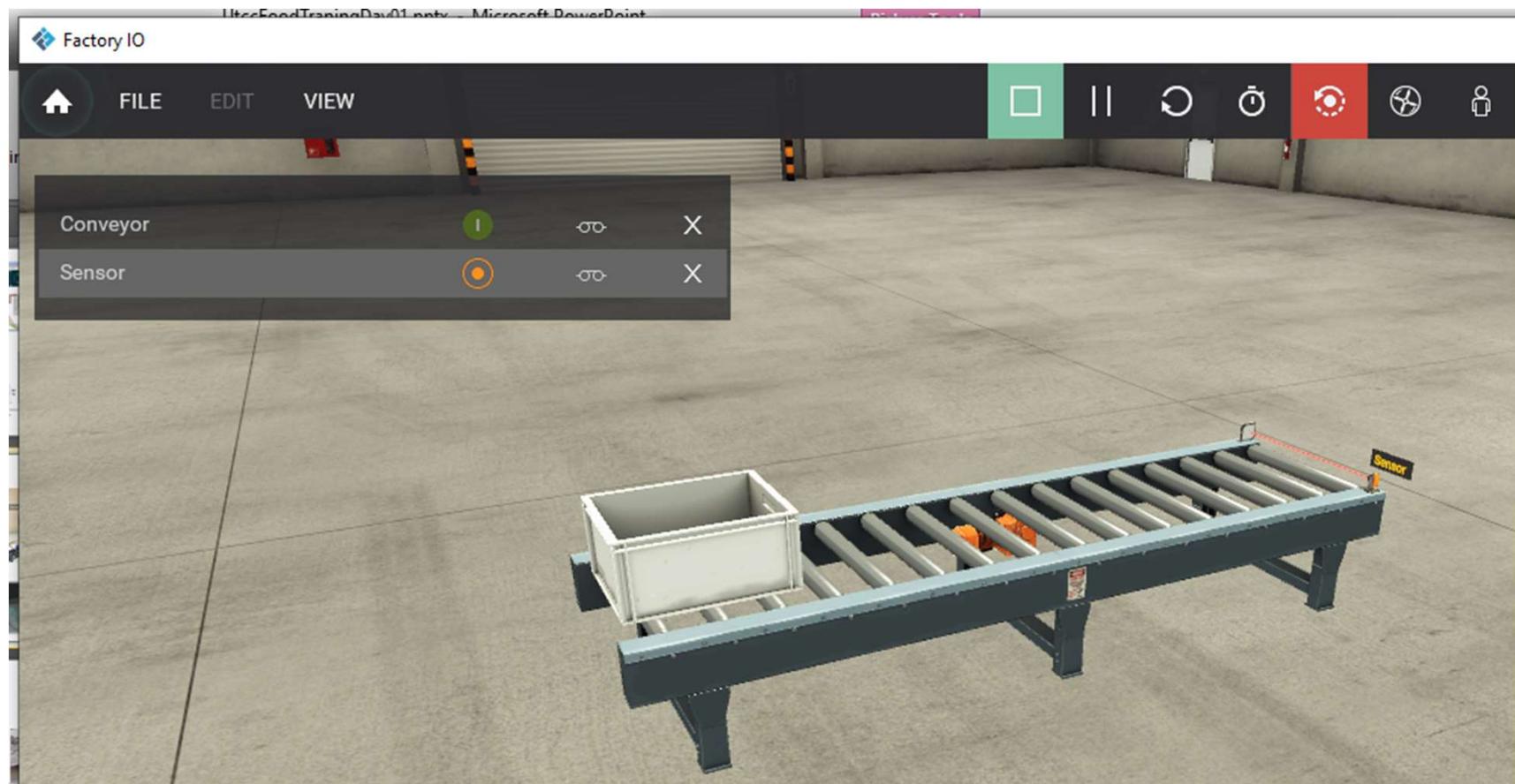
\$600.00 \$300.00

HMI Trainer Unit

\$600.00 \$300.00



Basic Control with Machine Simulator (FactoryIO)



A screenshot of a simulation software interface. At the top is a menu bar with icons for FILE, EDIT, and VIEW. The FILE menu is open, showing options: New (Ctrl+N), Open (Ctrl+O), Save (Ctrl+S), Save As... (Ctrl+Shift+S), Options, Drivers, and Exit. The 'Open' option is highlighted with a yellow box. Below the menu is a dark overlay window titled 'Open Scene'. On the left of this window is a sidebar with 'My Scenes' and 'Scenes' buttons; the 'Scenes' button is highlighted with a red box. The main area of the overlay shows four industrial scenes numbered 1 through 4, each with a thumbnail image and a description. The first scene, '1 - From A to B', has its thumbnail image outlined in orange.

← Open Scene

My Scenes

Scenes

FILE EDIT VIEW

New Ctrl+N

Open Ctrl+O

Save Ctrl+S

Save As... Ctrl+Shift+S

Options

Drivers F4

Exit

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.

UteFoodTrainingDay01.pptv - Microsoft PowerPoint

Factory IO

FILE EDIT VIEW

Conveyor I OO X

Sensor O OO X

FILE EDIT VIEW

New Ctrl+N

Open Ctrl+O

Save Ctrl+S

Save As... Ctrl+Shift+S

Options

Drivers F4

Exit

STOP CONFIGURATION CLEAR

DRIVER Modbus TCP/IP Server ✓

SENSORS

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

Sensor FACTORY I/O (Running) Sensor

(192.168.1.50:502)
Slave ID:1

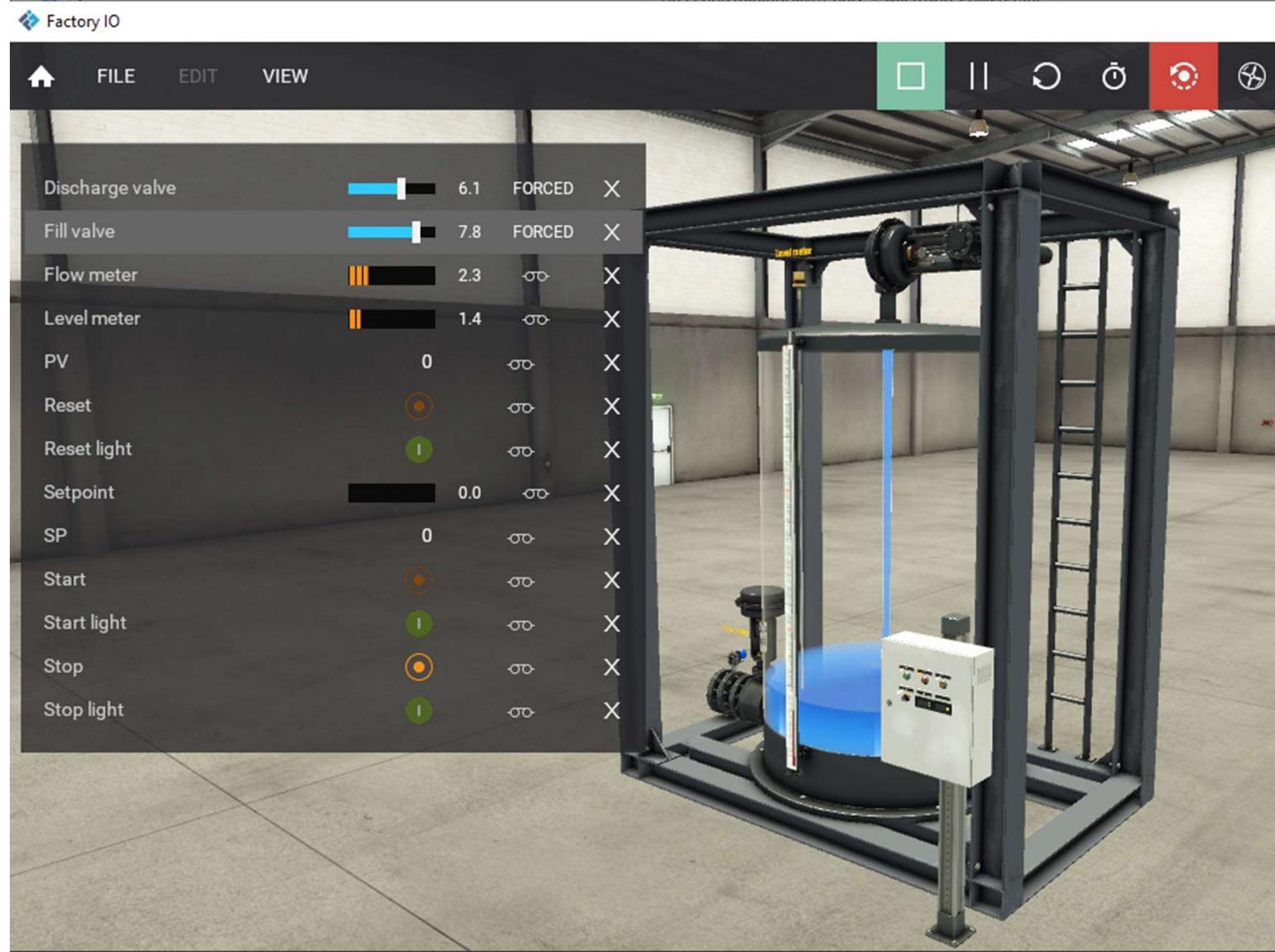
Input 0 Input 1

ACTUATORS

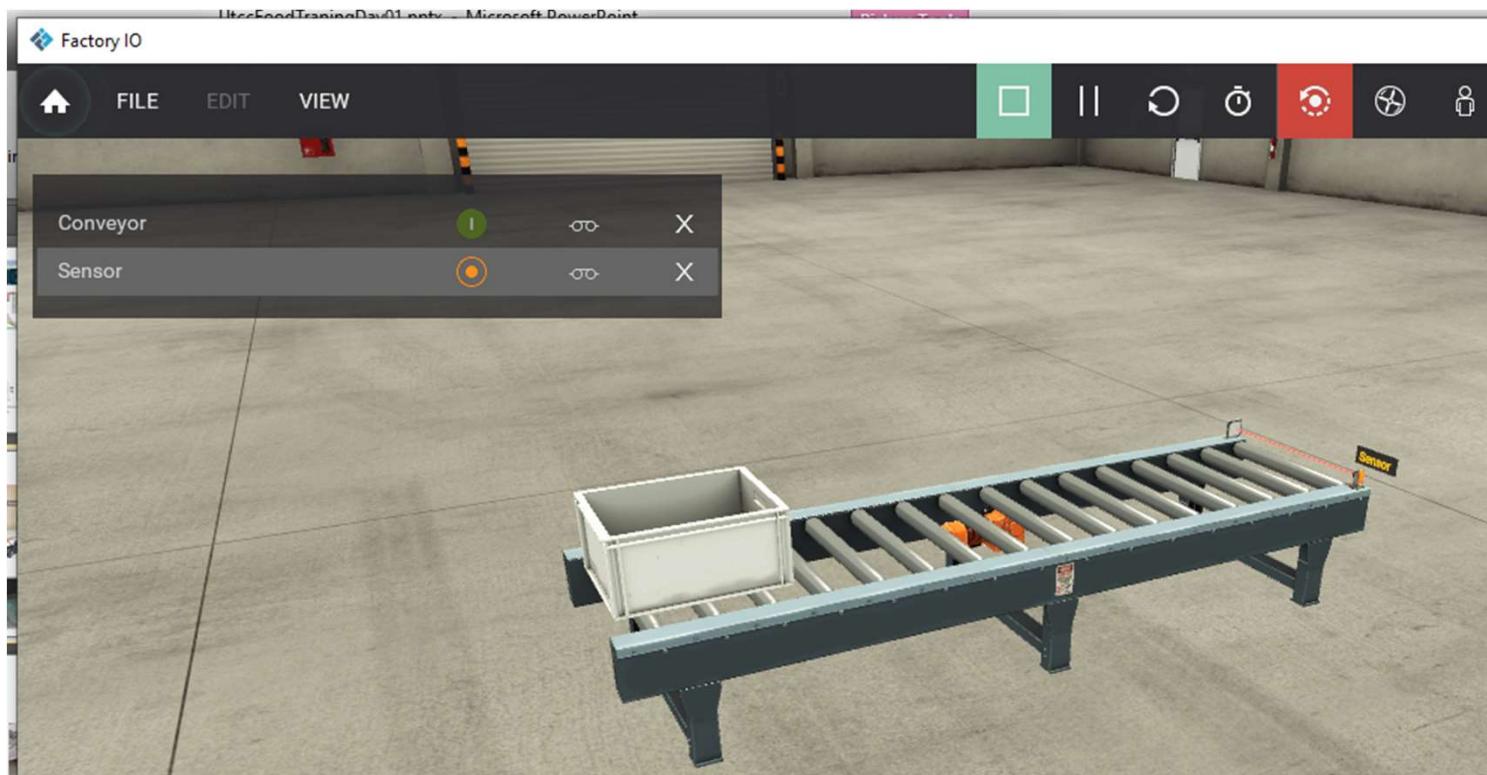
Conveyor FACTORY I/O (Camera Position) FACTORY I/O (Pause) FACTORY I/O (Reset) FACTORY I/O (Run)

Coil 0 Conveyor

Advanced Control with Machine Simulator (FactoryIO)



Basic Control with Machine Simulator (FactoryIO)



← DRIVER Control I/O ▾ CONFIGURATION CLEAR

SENSORS

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

ACTUATORS

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

Diagram showing a connection between a Sensor and a Conveyor actuator:

```
graph LR; Sensor[Sensor] --> Conveyor[Conveyor]
```

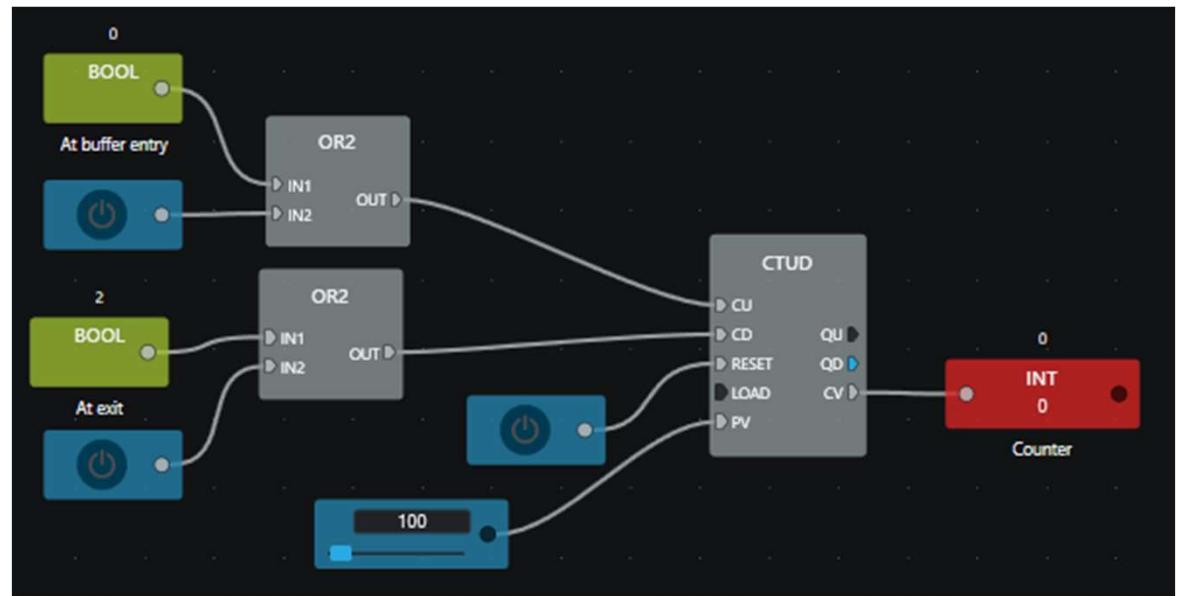
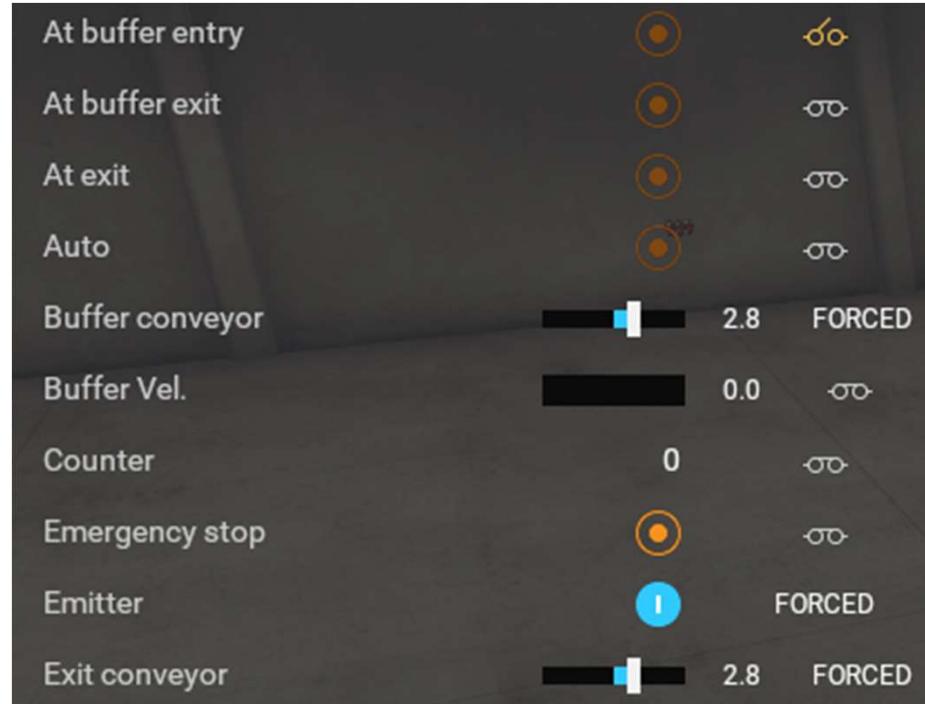
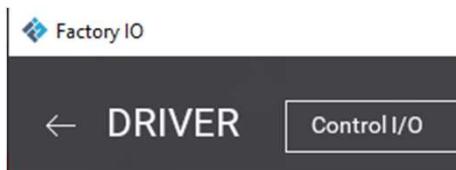
The diagram shows a connection between a Sensor and a Conveyor actuator. The Sensor is represented by a green rectangle labeled "BOOL" with a value of "1". It is connected to a red rectangle labeled "BOOL" with a value of "0", which is labeled "Conveyor".

YouTube FactoryIO - 04 Counter

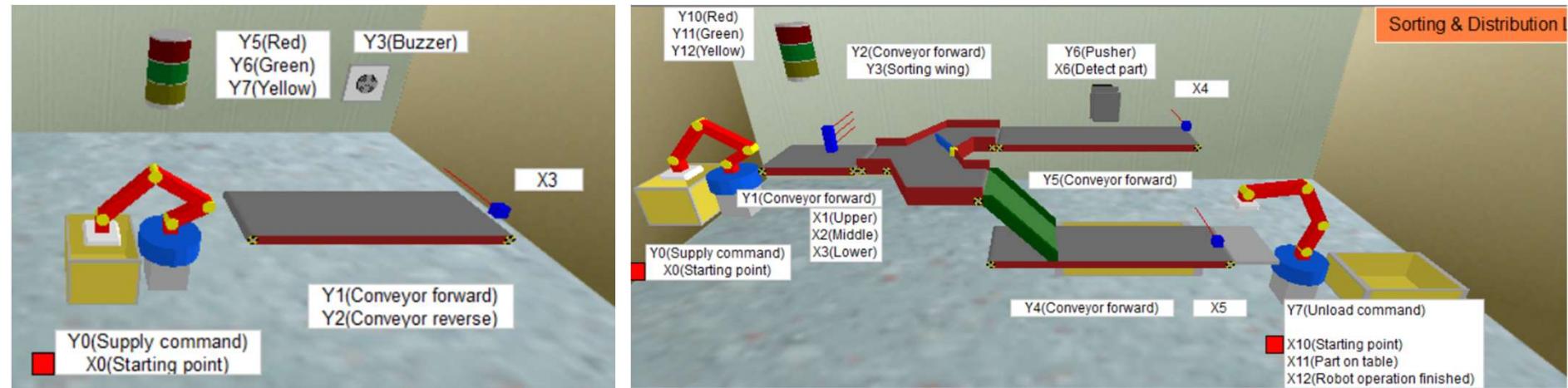
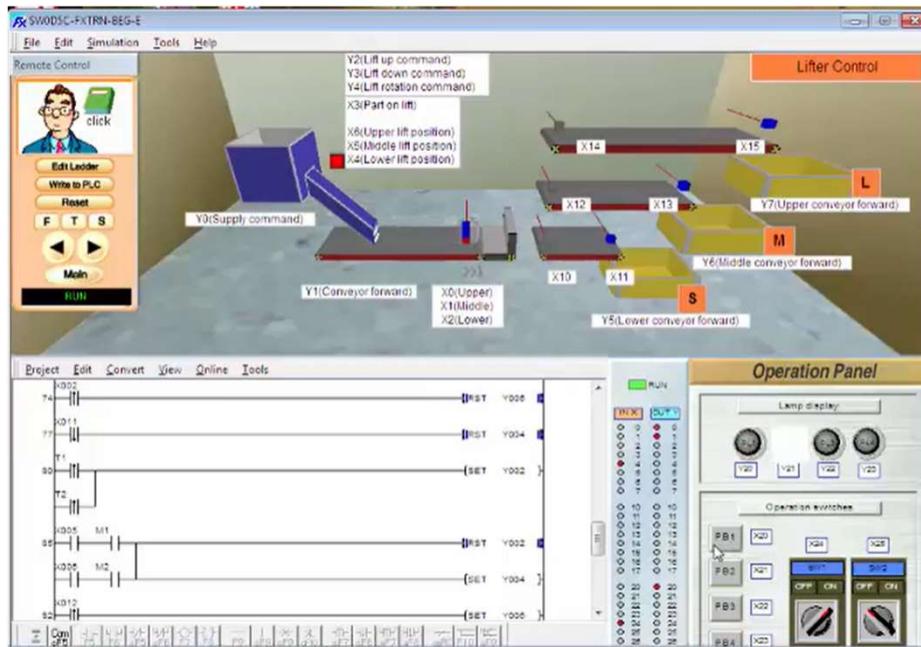


Buffer Station

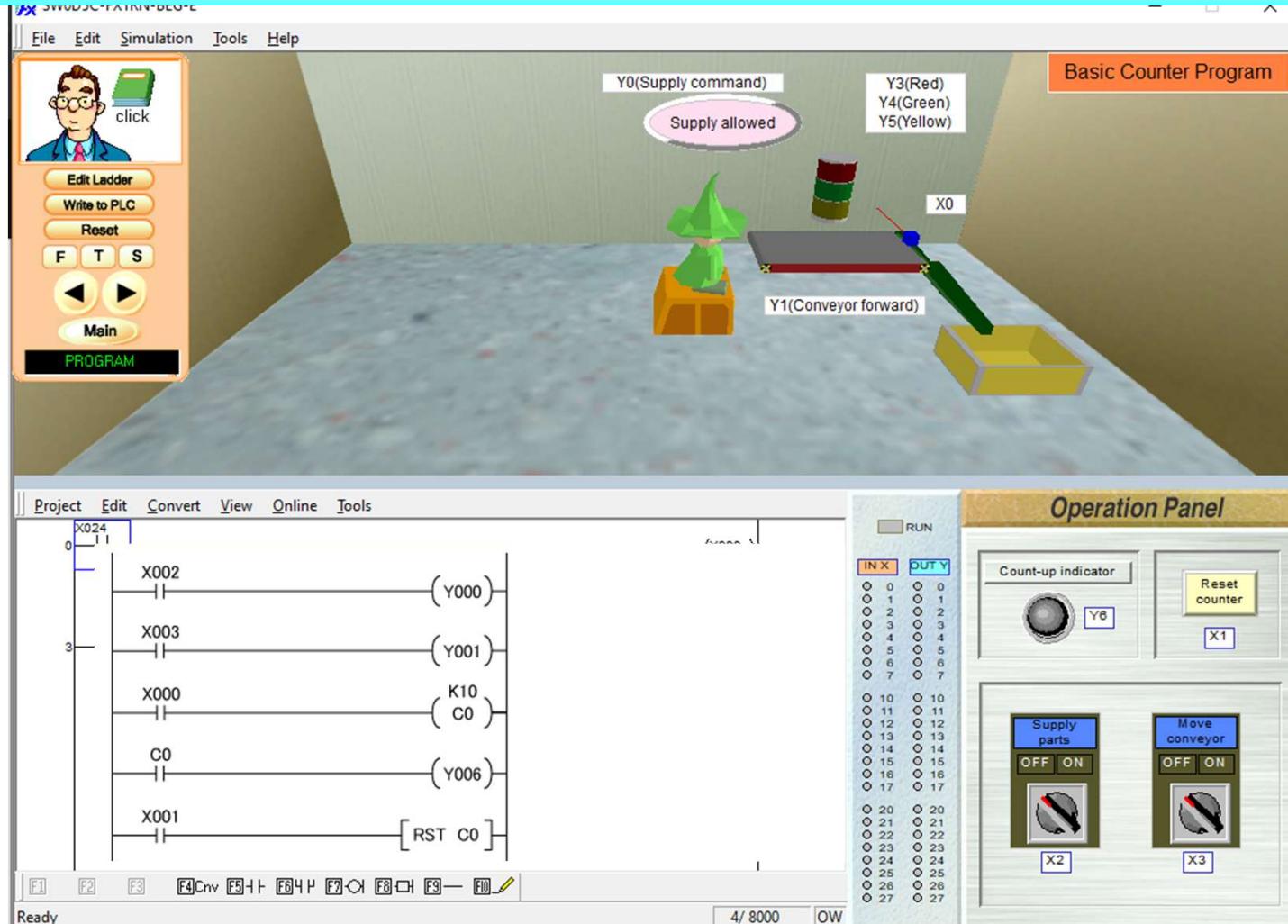
Buffer and separate up to five boxes.



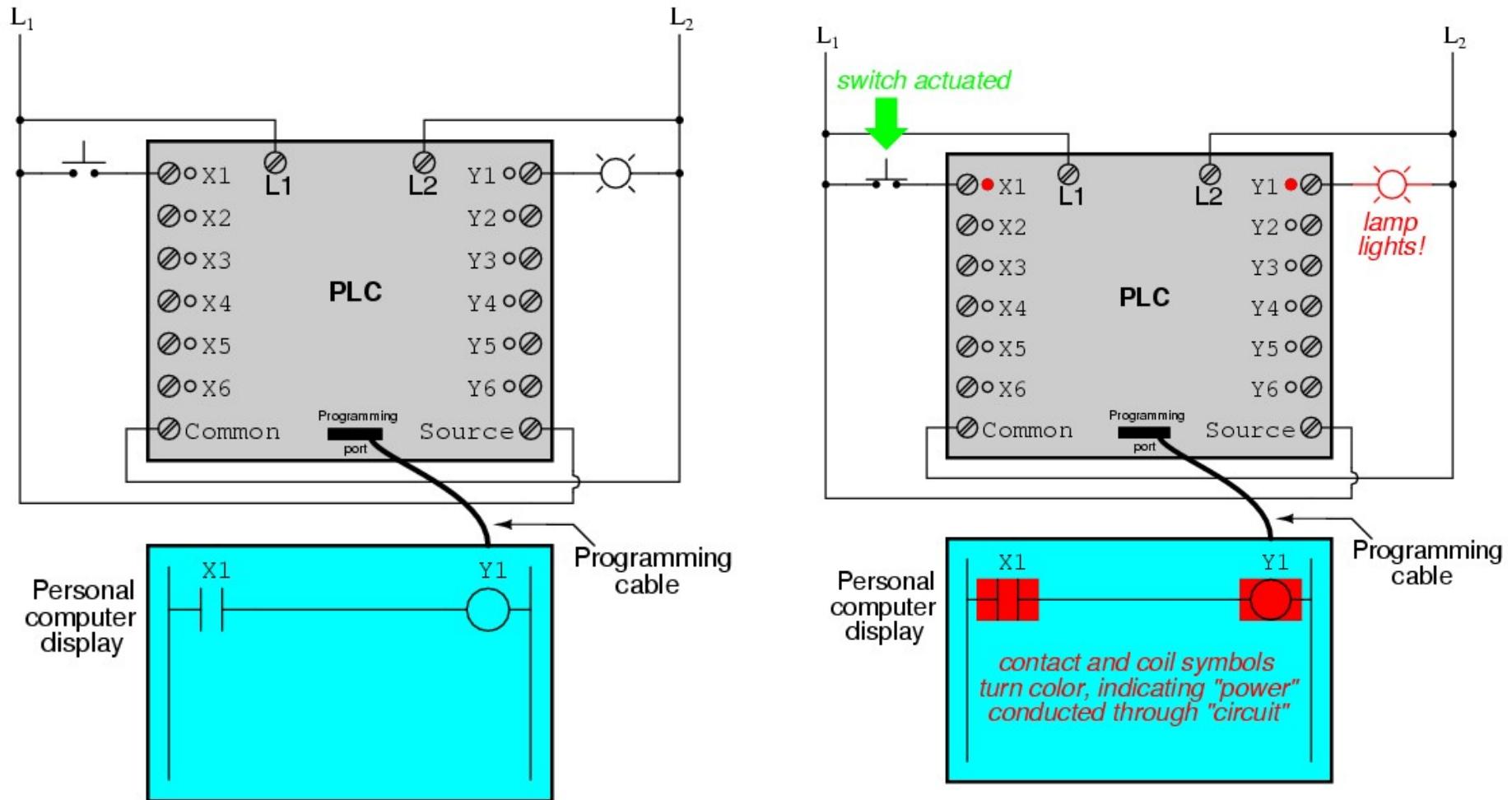
PLC Training (Mitsubishi Simulator/Omron Training Set)

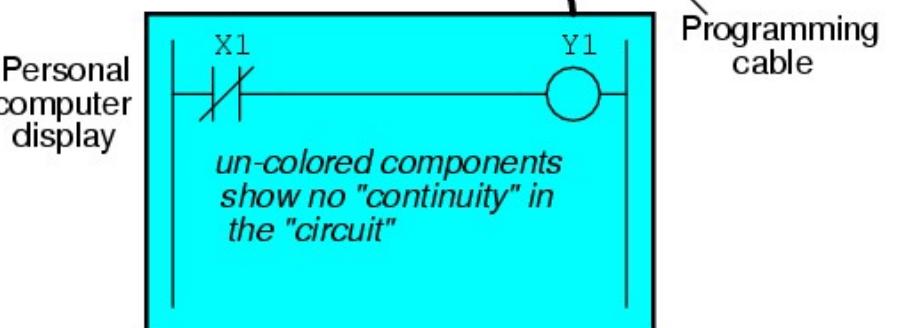
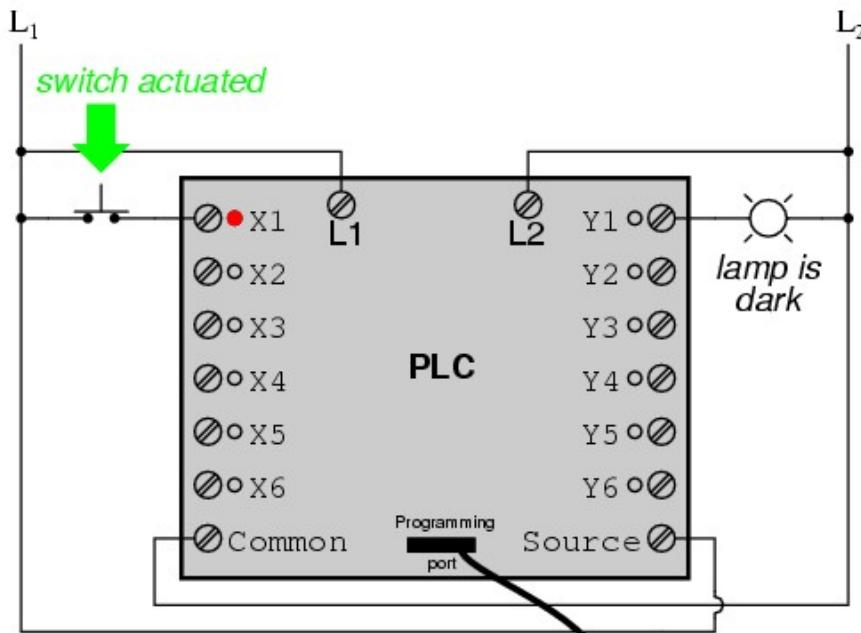
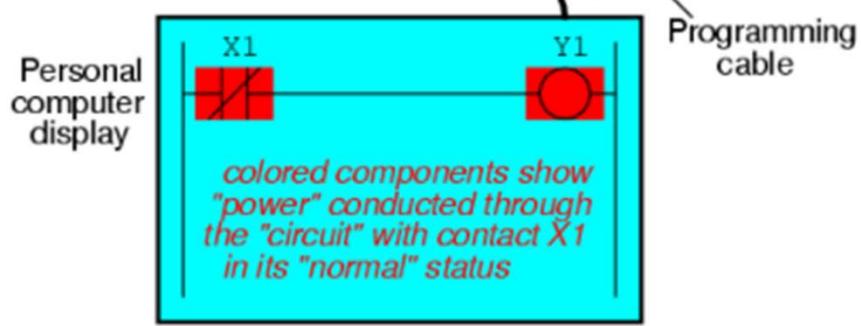
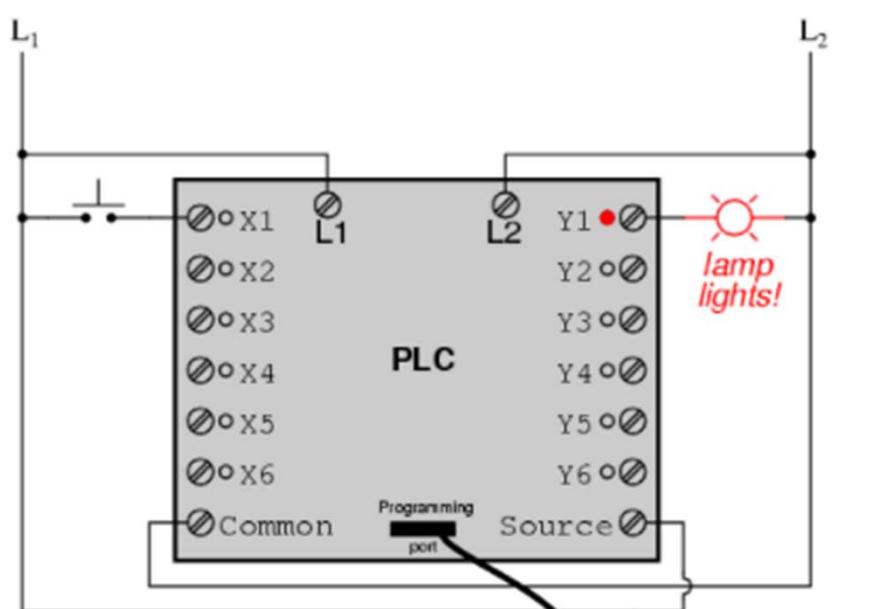


ระบบควบคุมพื้นฐานในอุตสาหกรรมด้วย PLC

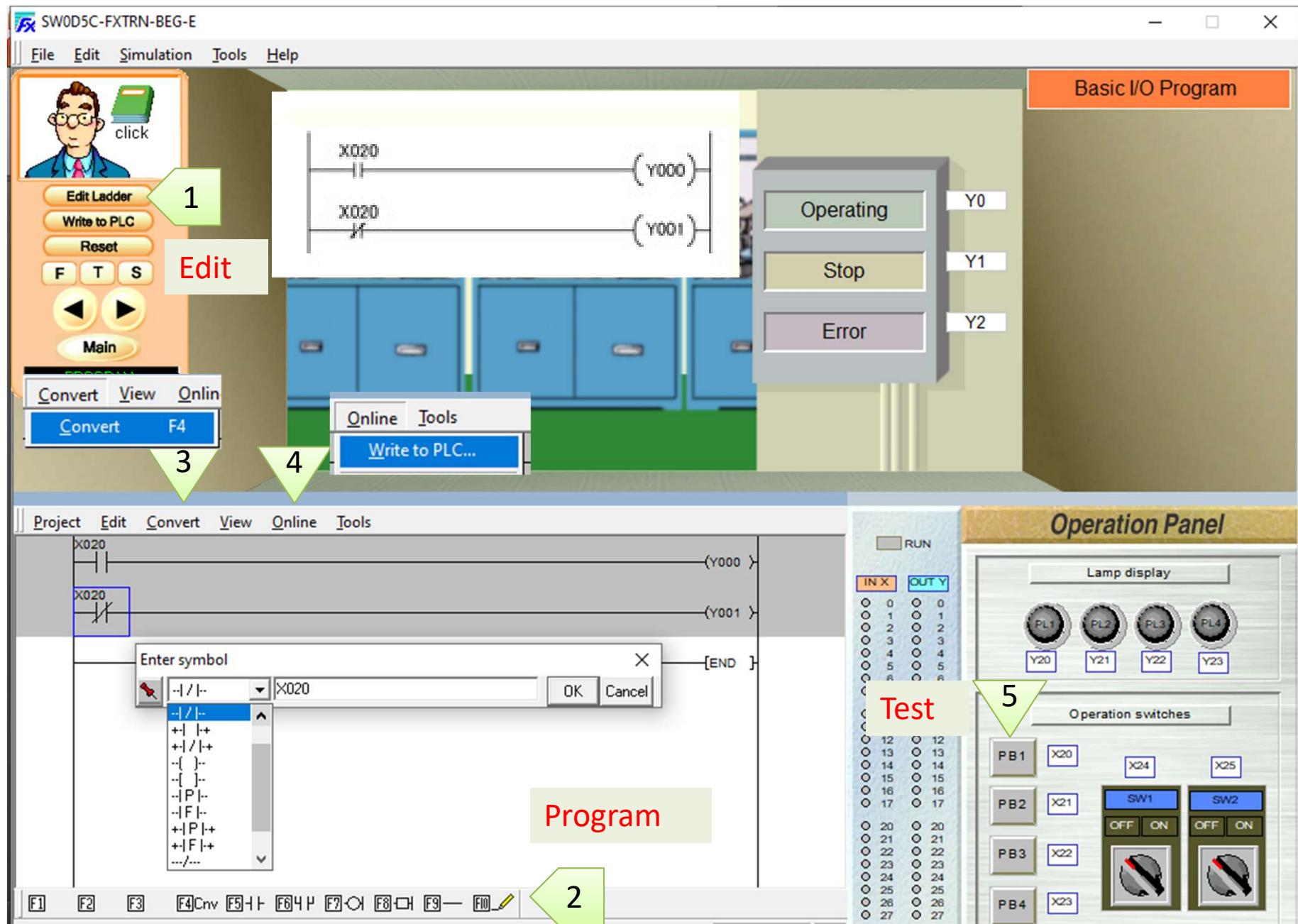


Programmable Logic Controllers (PLC)





YouTube PLC - 01 Basic Program



B-3. Control Precedence Program

Learn an interlock program which controls conflicting operations.

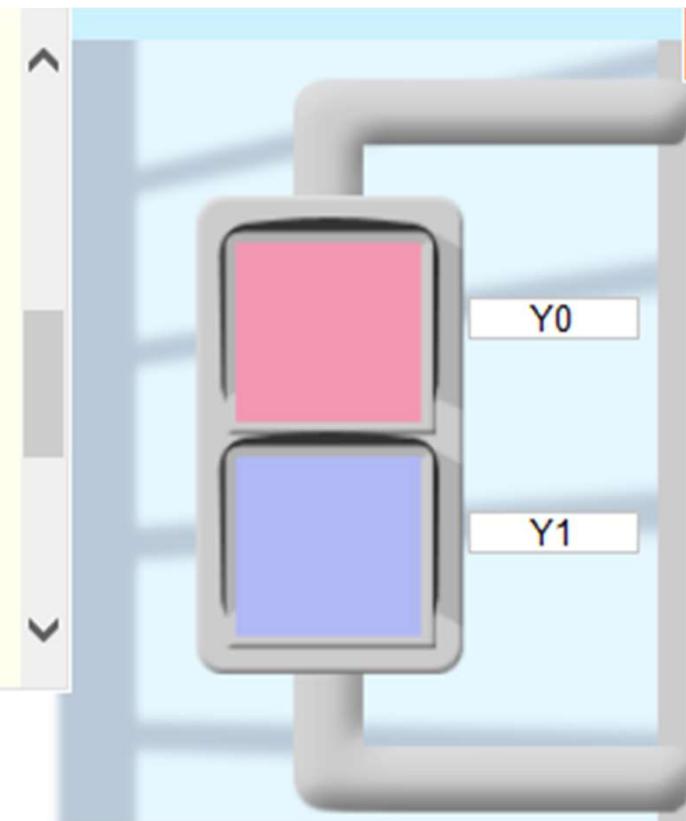
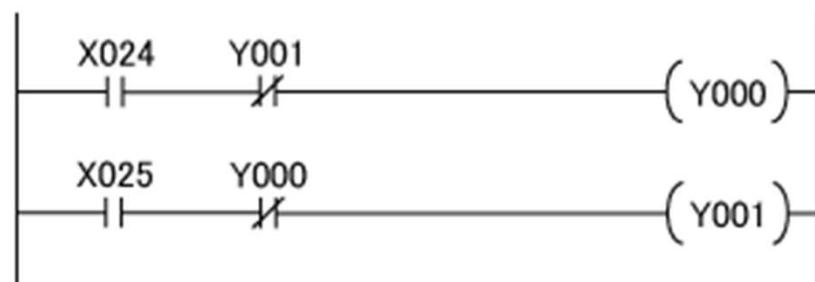


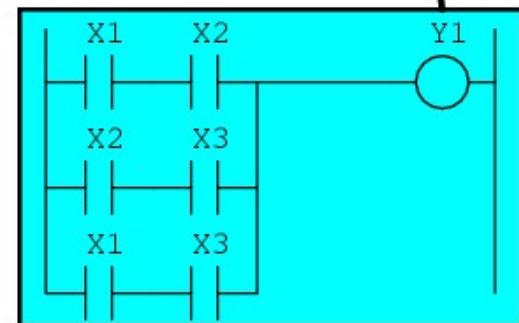
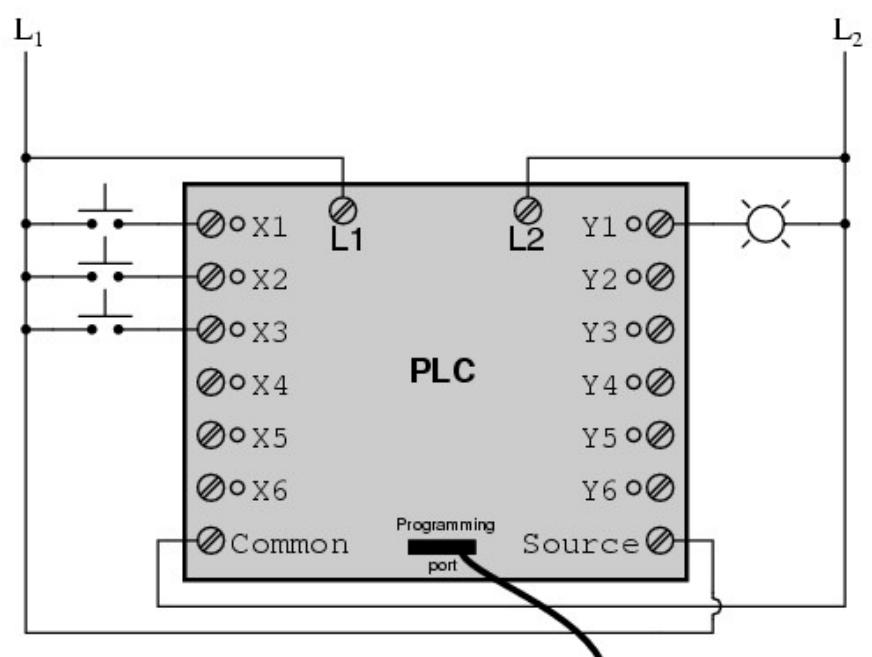
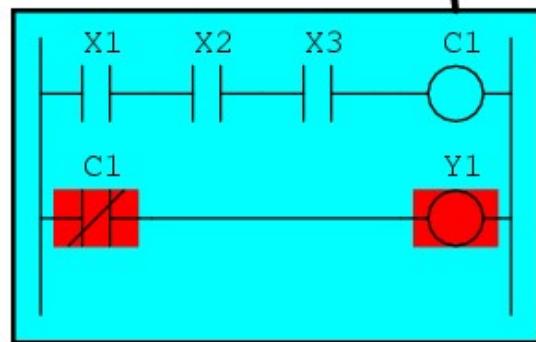
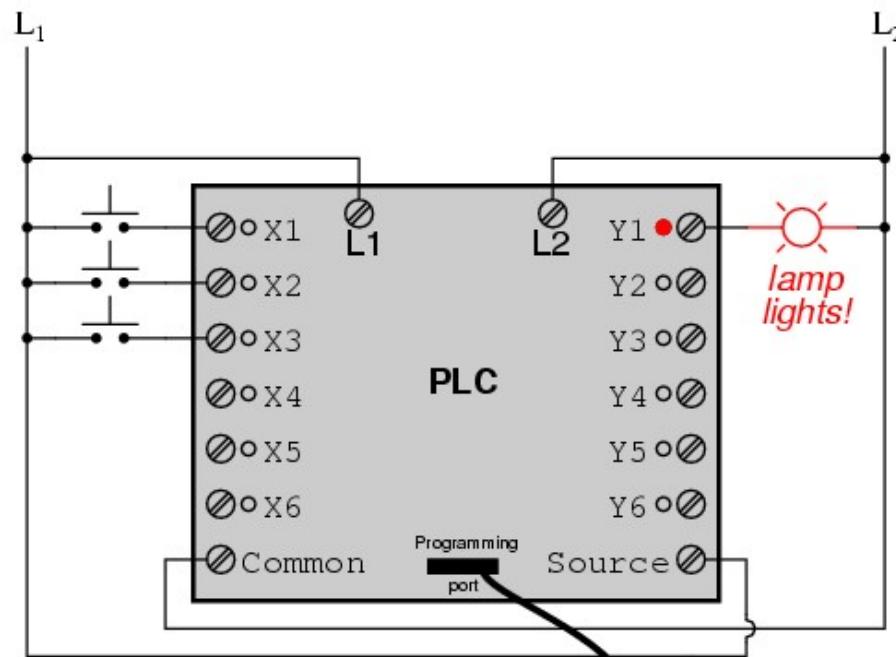
These symbols can be input by pressing the following keys.

—|—:[F5]

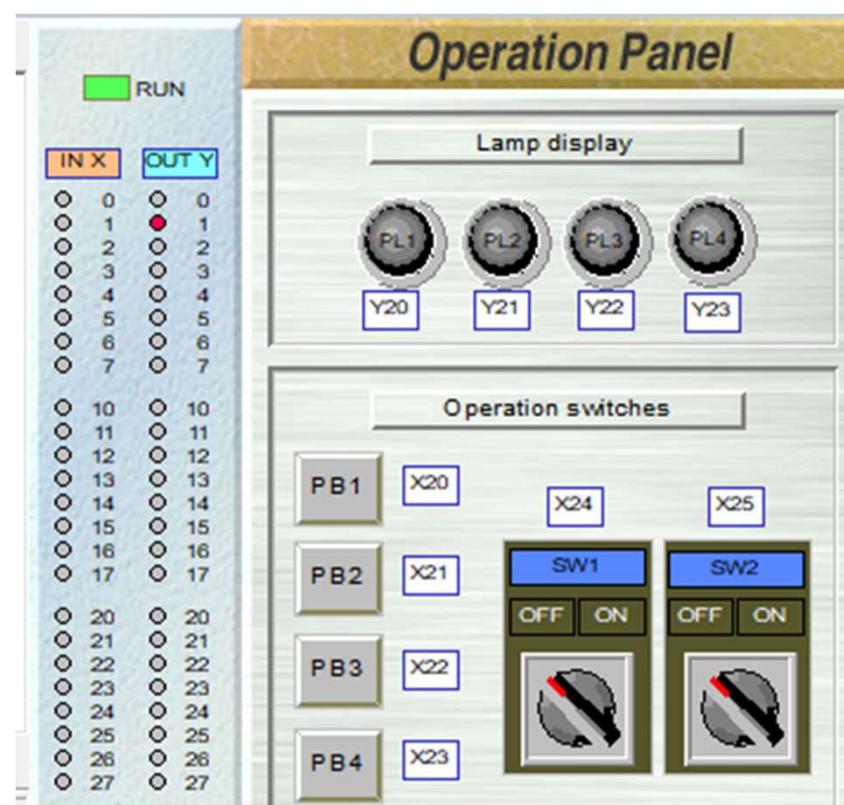
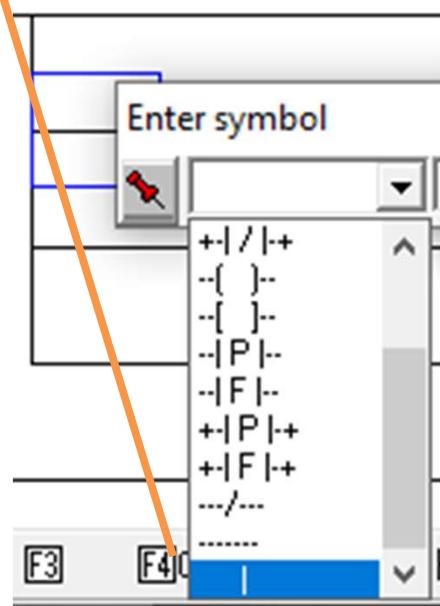
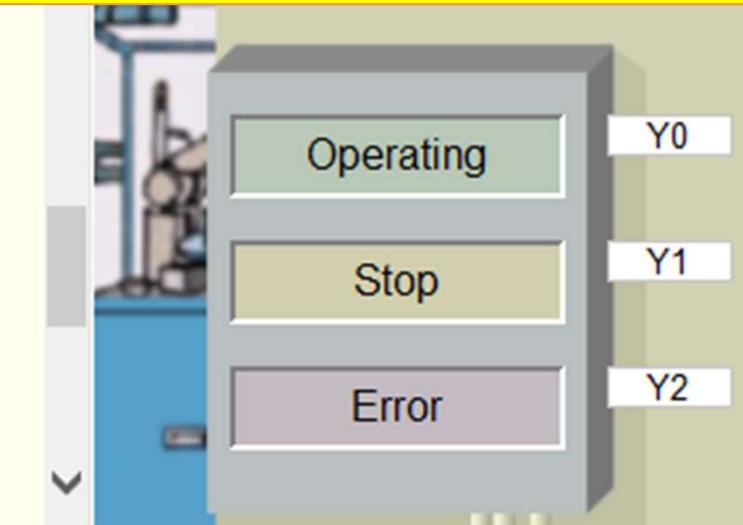
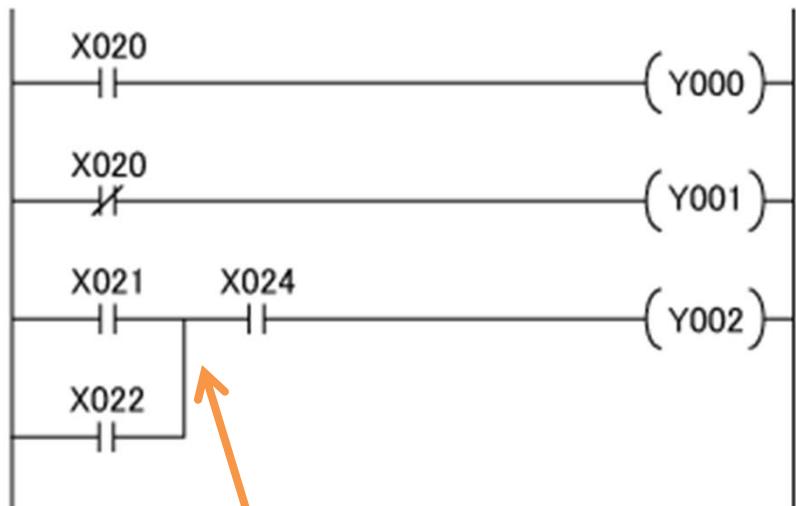
—X—:[Shift]+[F5]

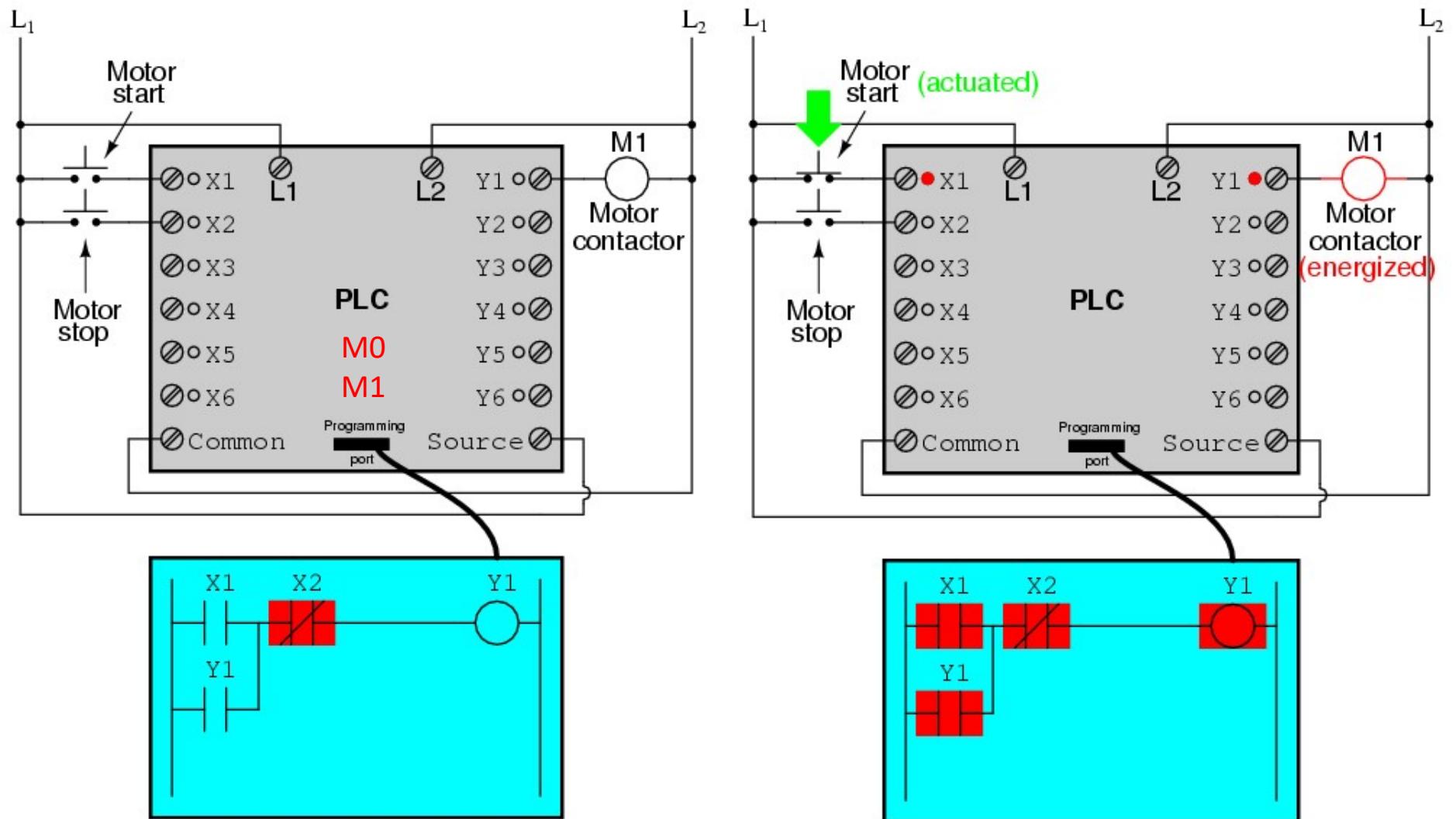
{ }:[F7]





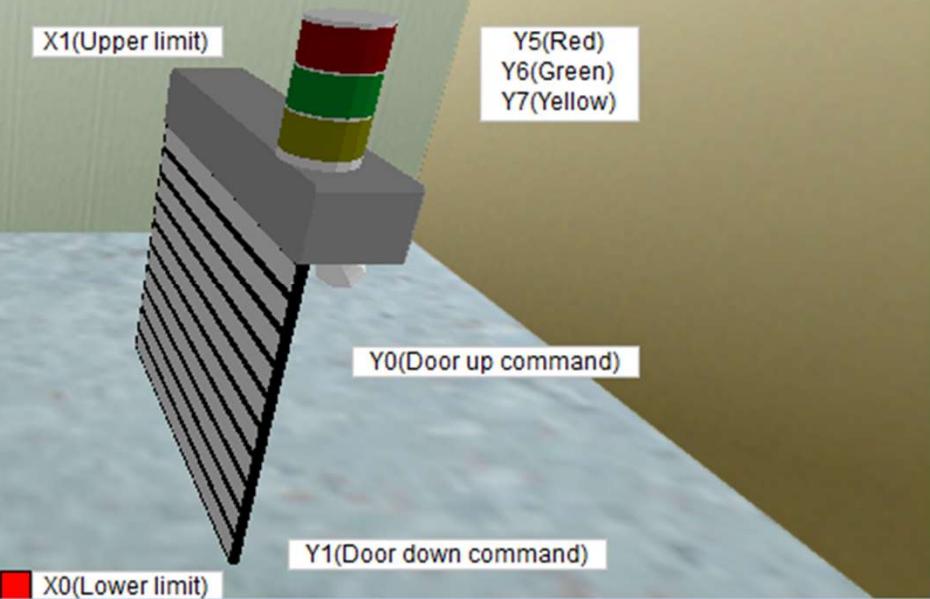
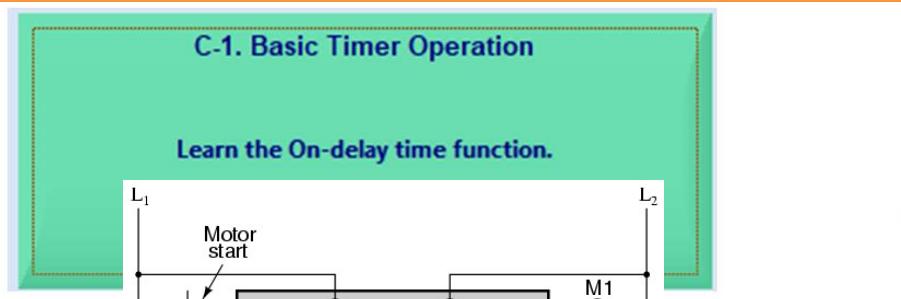
YouTube PLC - 03 Basic Program





YouTube PLC - 04 Self Holding Program

Basic Timer Operation



Project Edit Convert View Online Tools

IN X	OUT Y
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27

Op

PL1 Y20

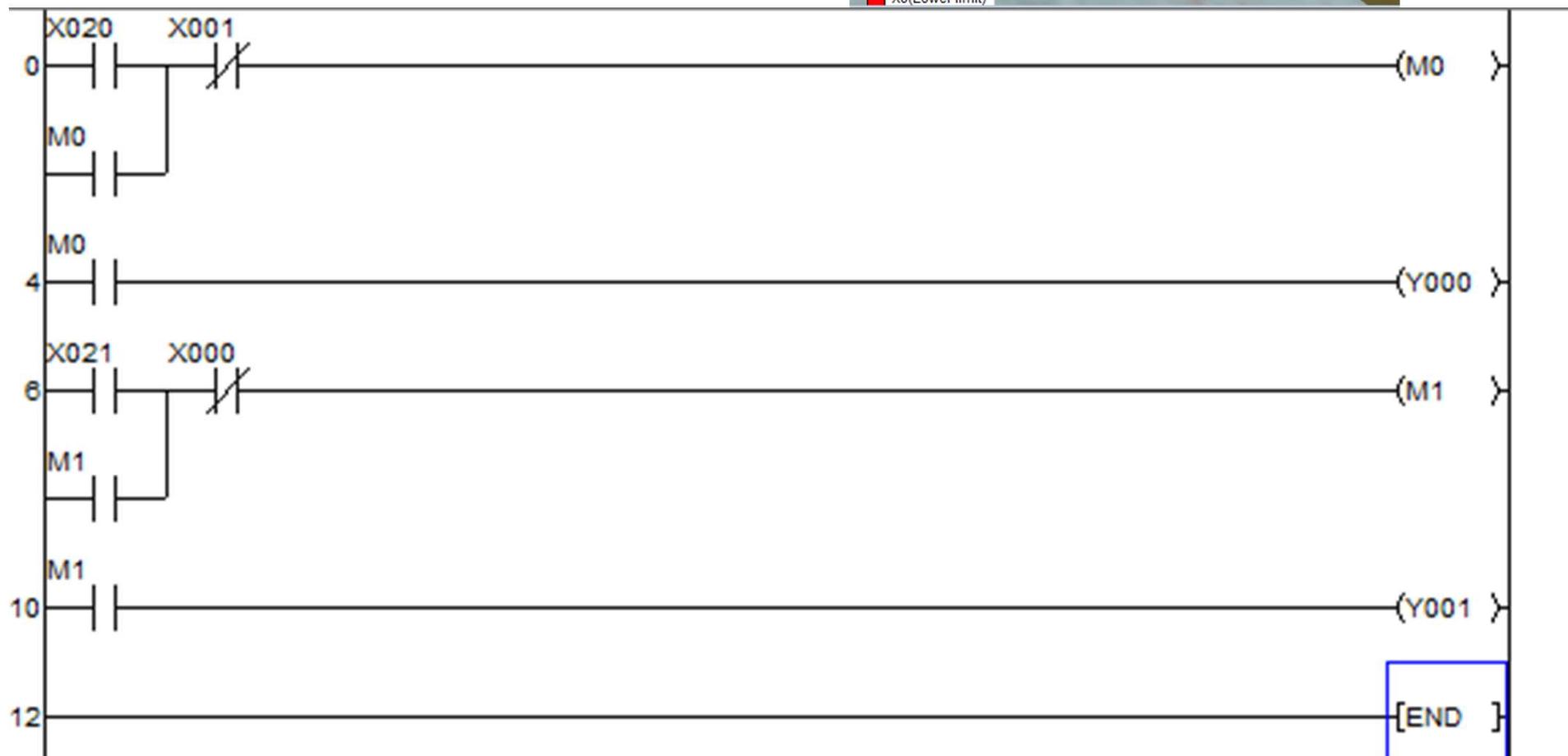
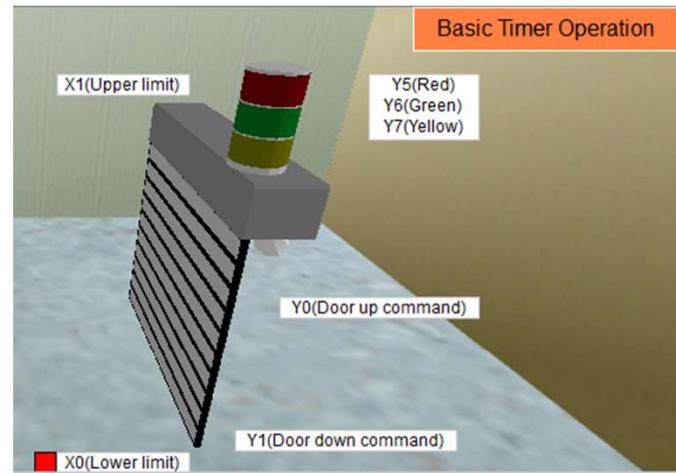
PB1 X20

PB2 X21

PB3 X22

PB4 X23

YouTube PLC - 04 Self Holding Program



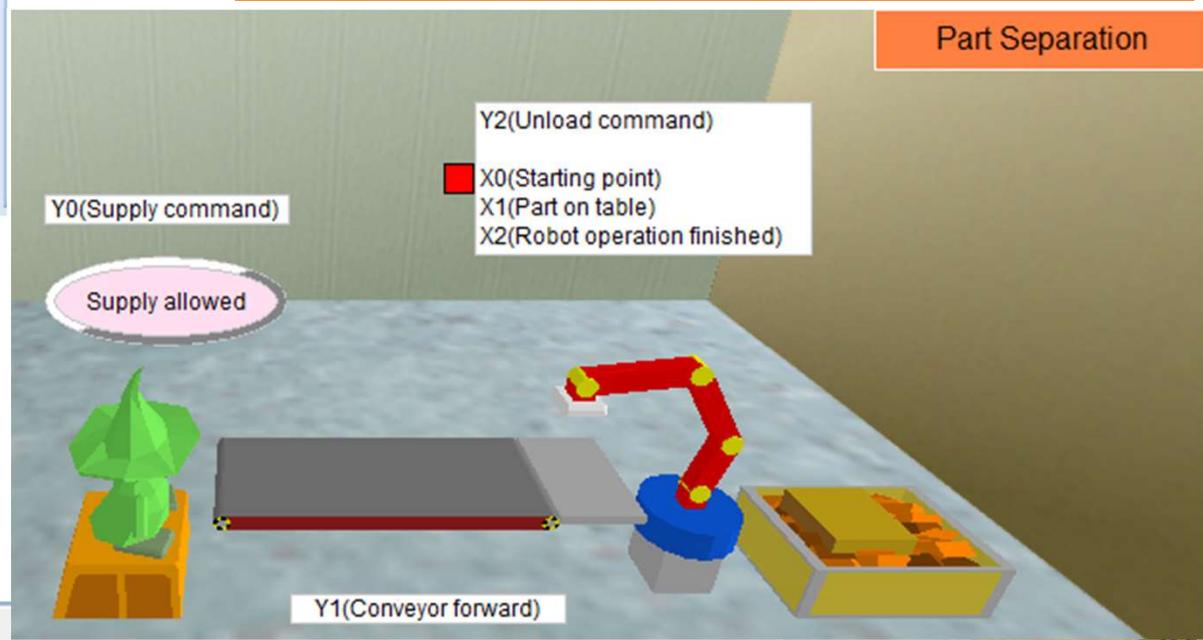
YouTube PLC - 05 Timer Program

E-3. Part Separation

Give the command to a robot to move a part to a new location.

```

    X001 ---|--- X024 ---|--- Y000
    |           |
    |           +-- T0 (Y001)
    |           |
    |           +-- T1 (Y002)
    |           |
    |           +-- Y020
    |           |
    |           +-- T0 (Y010)
    |           |
    |           +-- T1 (Y011)
    |           |
    |           +-- Y019
    |           |
    |           +-- T0 (Y012)
    |           |
    |           +-- T1 (Y013)
    |           |
    |           +-- Y018
    |           |
    |           +-- T0 (Y014)
    |           |
    |           +-- T1 (Y015)
    |           |
    |           +-- Y017
    |           |
    |           +-- T0 (Y016)
    |           |
    |           +-- T1 (Y017)
    |           |
    |           +-- Y016
    |           |
    |           +-- T0 (Y018)
    |           |
    |           +-- T1 (Y019)
    |           |
    |           +-- Y015
    |           |
    |           +-- T0 (Y010)
    |           |
    |           +-- T1 (Y011)
    |           |
    |           +-- Y014
    |           |
    |           +-- T0 (Y012)
    |           |
    |           +-- T1 (Y013)
    |           |
    |           +-- Y013
    |           |
    |           +-- T0 (Y014)
    |           |
    |           +-- T1 (Y015)
    |           |
    |           +-- Y012
    |           |
    |           +-- T0 (Y016)
    |           |
    |           +-- T1 (Y017)
    |           |
    |           +-- Y011
    |           |
    |           +-- T0 (Y018)
    |           |
    |           +-- T1 (Y019)
    |           |
    |           +-- Y010
    |           |
    |           +-- T0 (Y010)
    |           |
    |           +-- T1 (Y011)
    |           |
    |           +-- Y009
    |           |
    |           +-- T0 (Y012)
    |           |
    |           +-- T1 (Y013)
    |           |
    |           +-- Y008
    |           |
    |           +-- T0 (Y014)
    |           |
    |           +-- T1 (Y015)
    |           |
    |           +-- Y007
    |           |
    |           +-- T0 (Y016)
    |           |
    |           +-- T1 (Y017)
    |           |
    |           +-- Y006
    |           |
    |           +-- T0 (Y018)
    |           |
    |           +-- T1 (Y019)
    |           |
    |           +-- Y005
    |           |
    |           +-- T0 (Y010)
    |           |
    |           +-- T1 (Y011)
    |           |
    |           +-- Y004
    |           |
    |           +-- T0 (Y012)
    |           |
    |           +-- T1 (Y013)
    |           |
    |           +-- Y003
    |           |
    |           +-- T0 (Y014)
    |           |
    |           +-- T1 (Y015)
    |           |
    |           +-- Y002
    |           |
    |           +-- T0 (Y016)
    |           |
    |           +-- T1 (Y017)
    |           |
    |           +-- Y001
    |           |
    |           +-- T0 (Y018)
    |           |
    |           +-- T1 (Y019)
    |           |
    |           +-- Y000
  
```



Project Edit Convert View Online Tools

IN X	OUT Y
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17
16	20
17	21
18	22
19	23
20	24
21	25
22	26
23	27
24	28
25	29
26	30
27	31

RUN

PL1 Y20

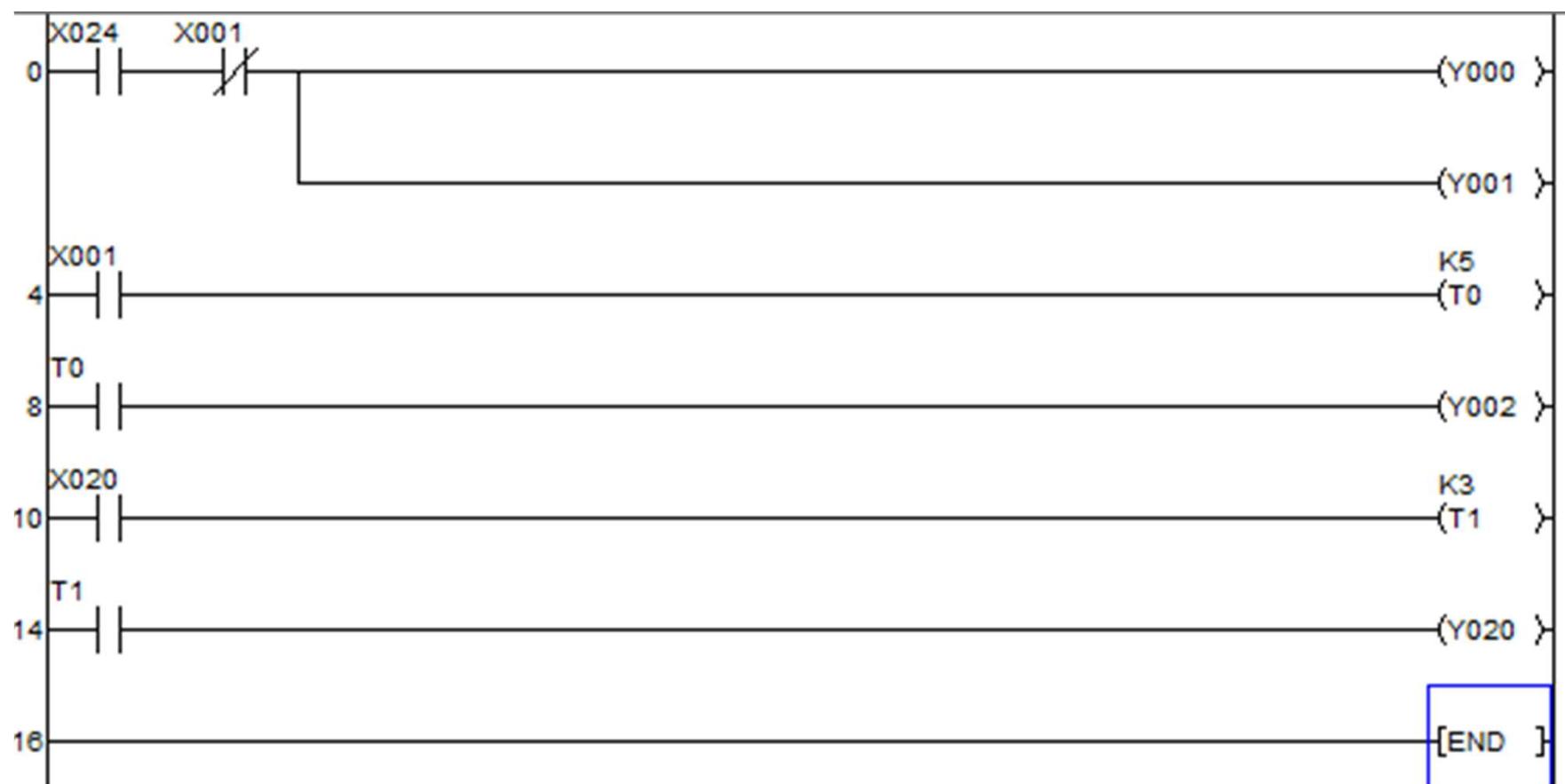
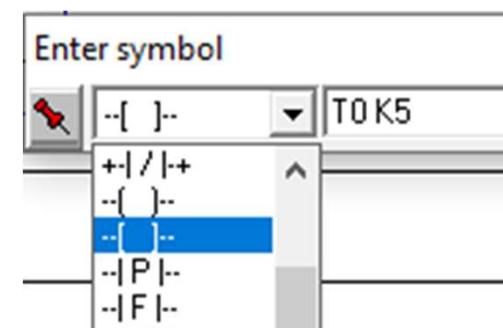
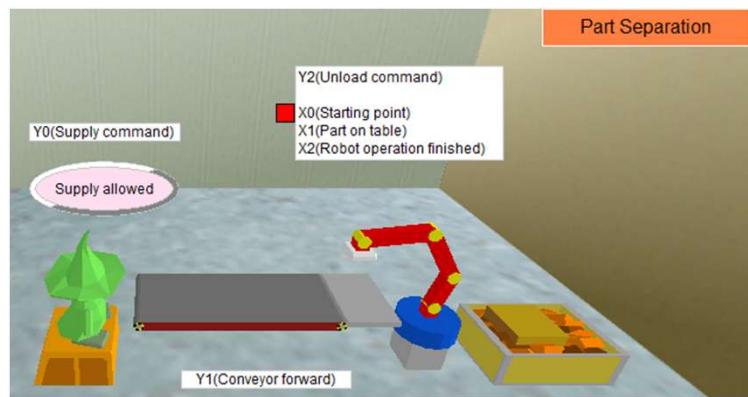
PB1 X20

PB2 X21

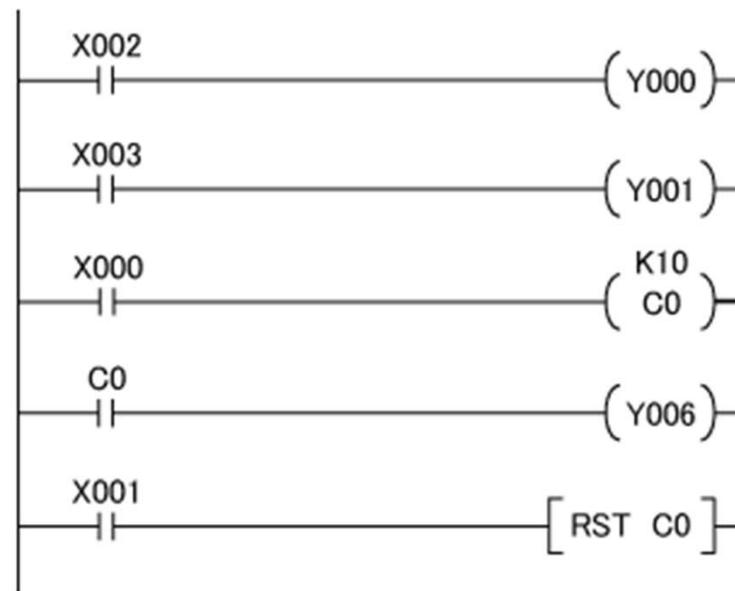
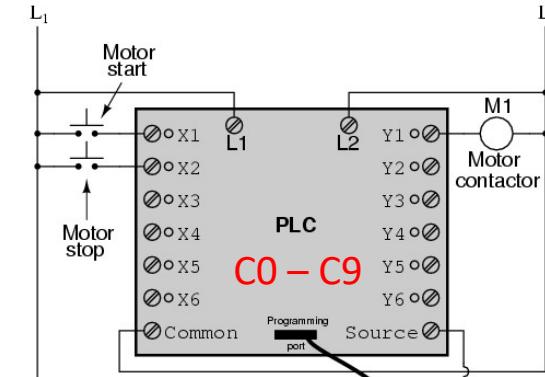
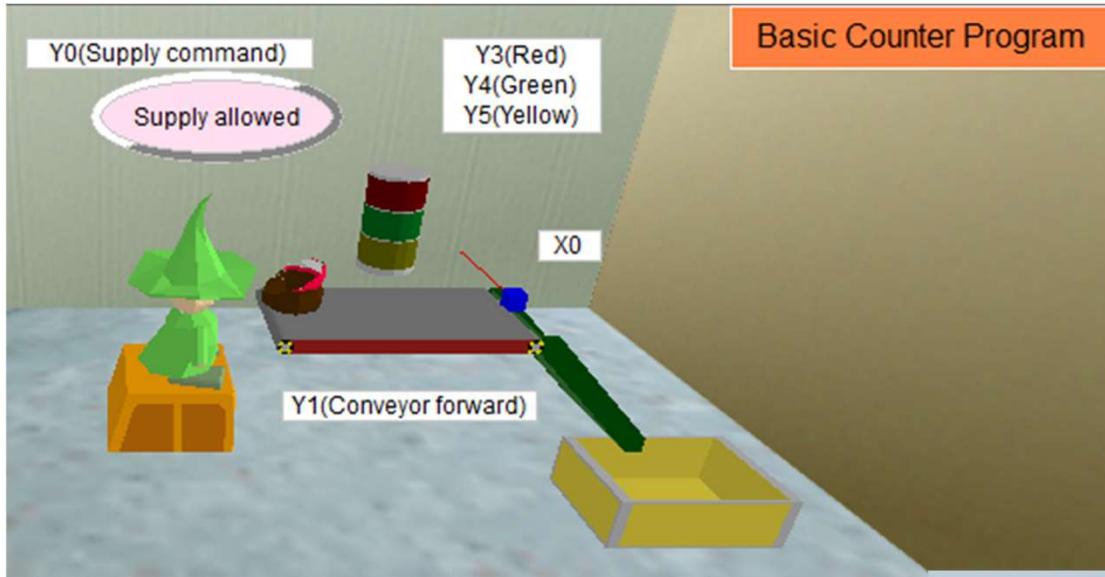
PB3 X22

PB4 X23

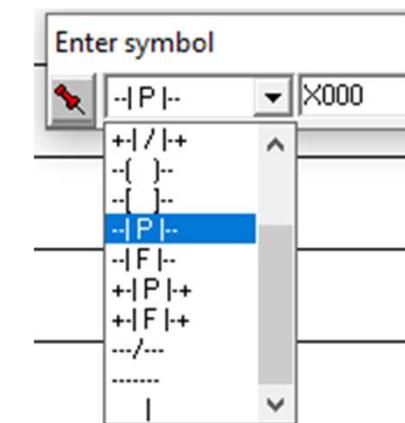
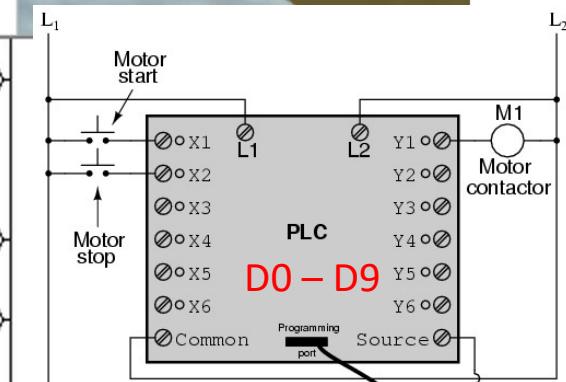
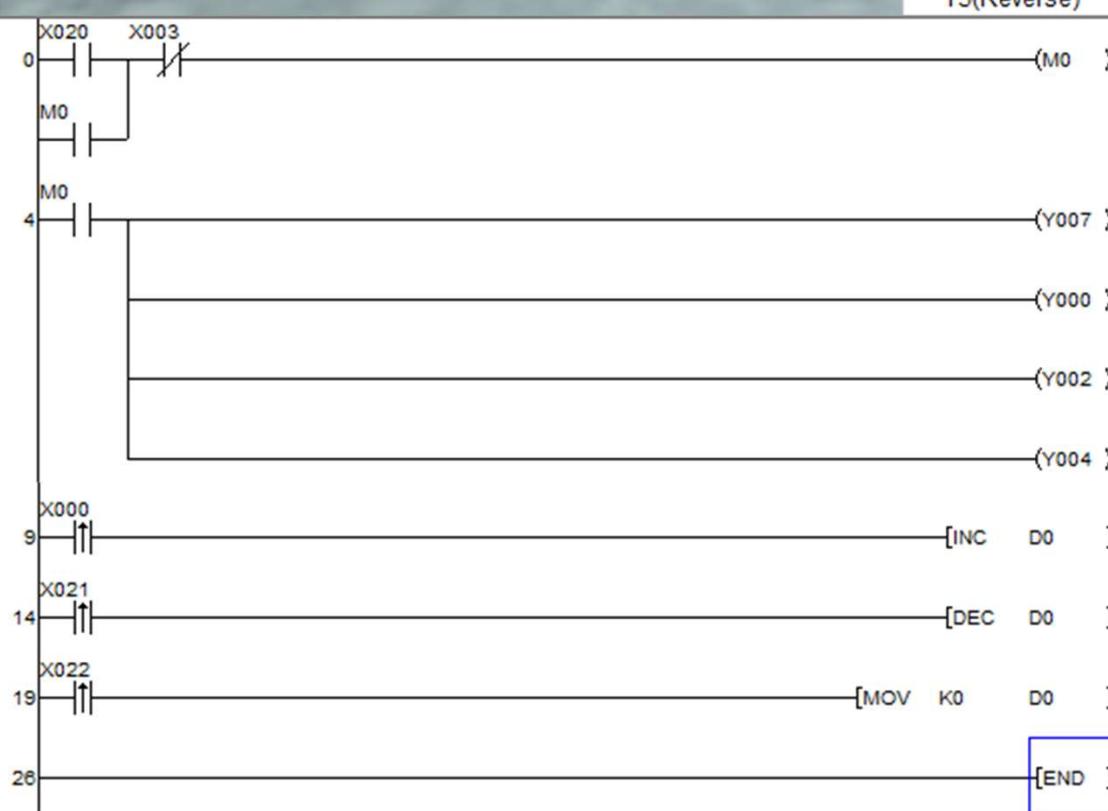
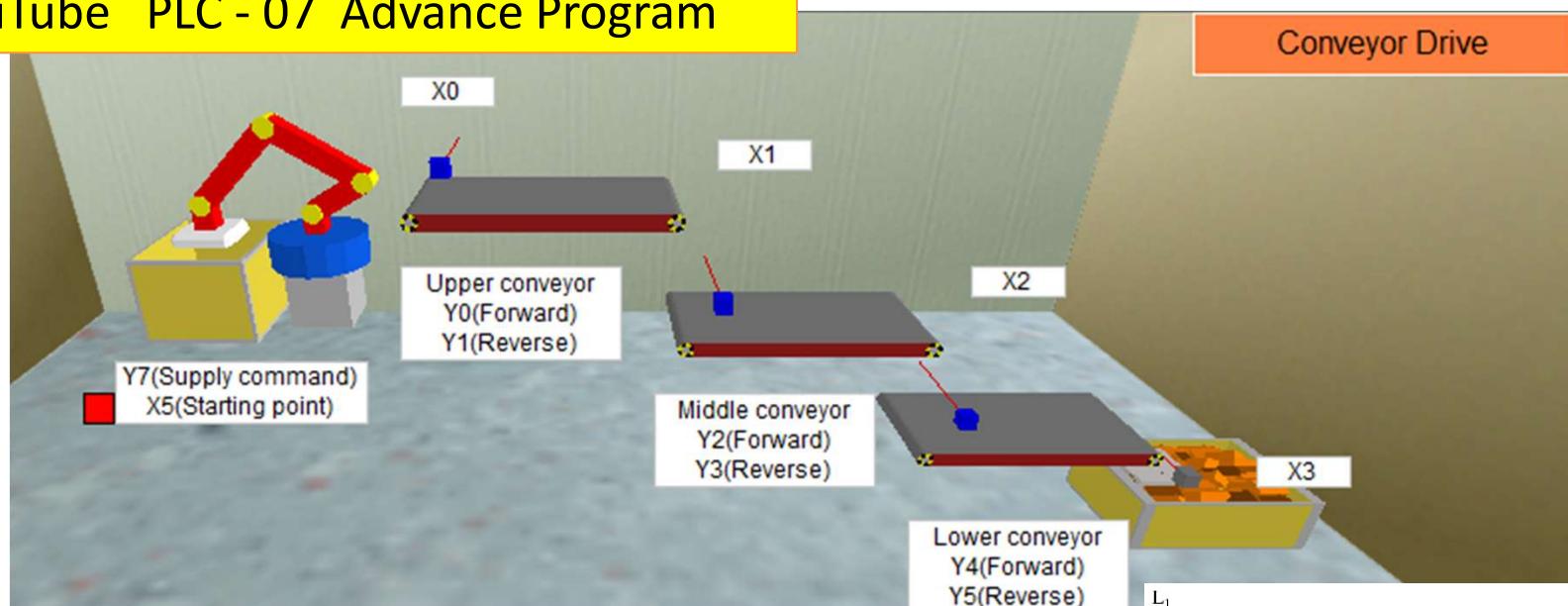
YouTube PLC - 05 Timer Program



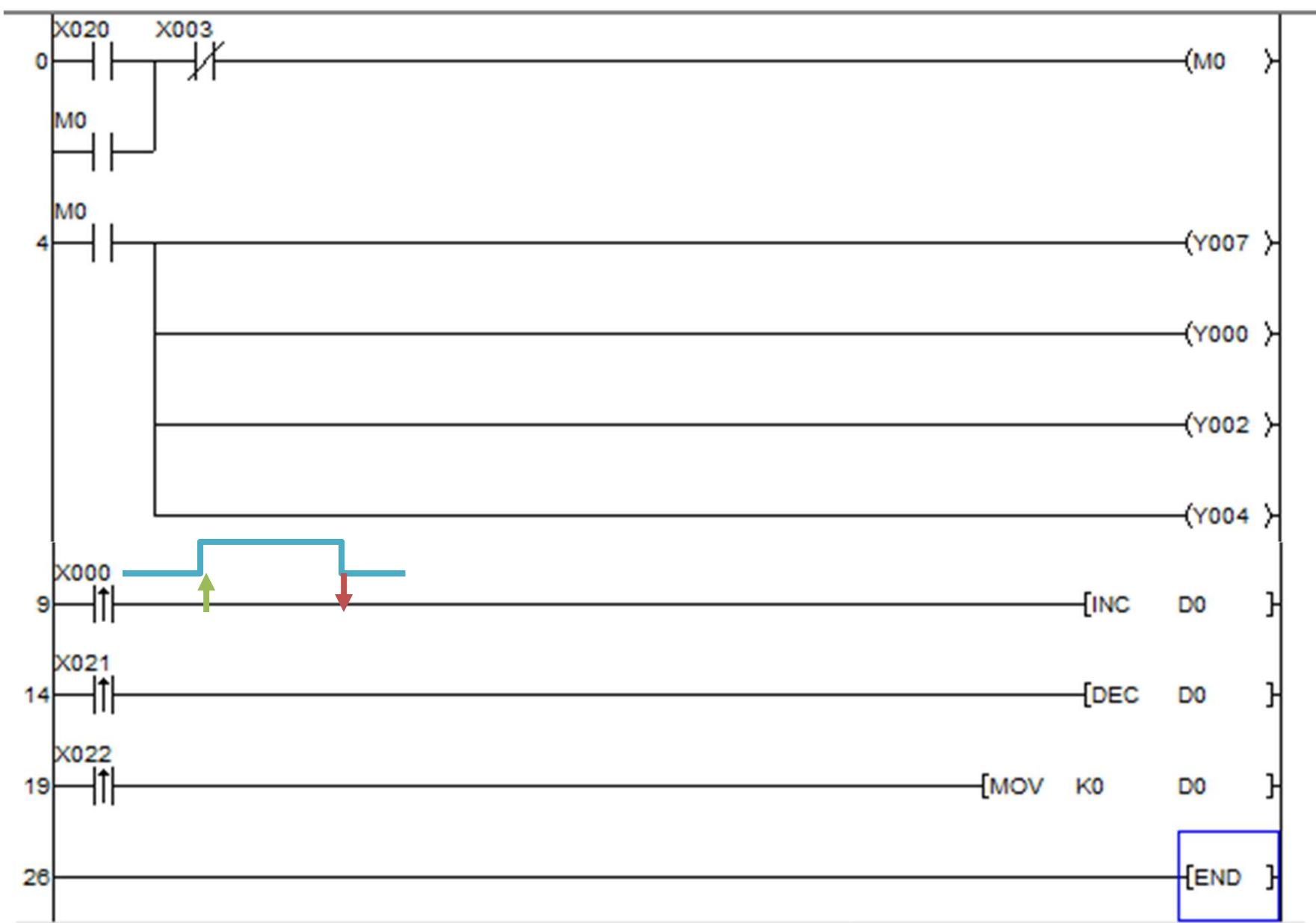
YouTube PLC - 06 Counter Program

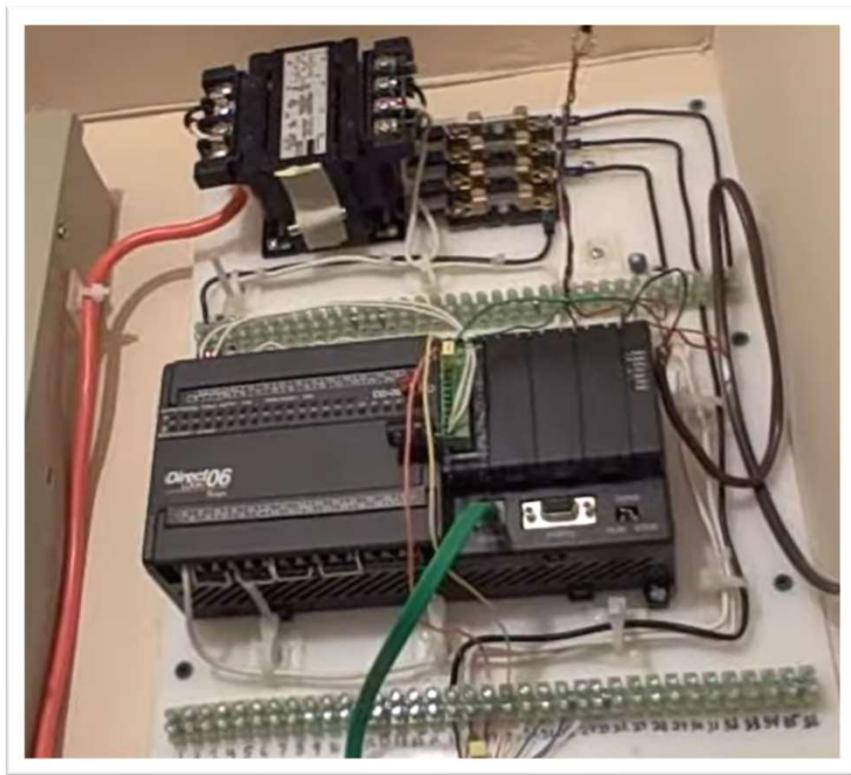


YouTube PLC - 07 Advance Program

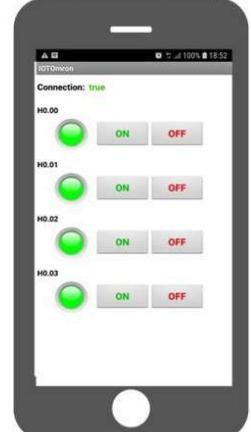


YouTube PLC - 07 Advance Program

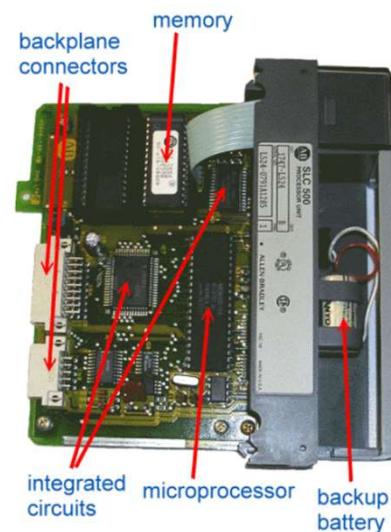
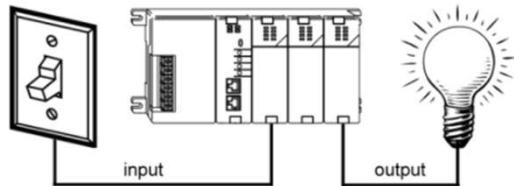
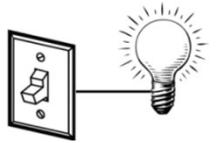




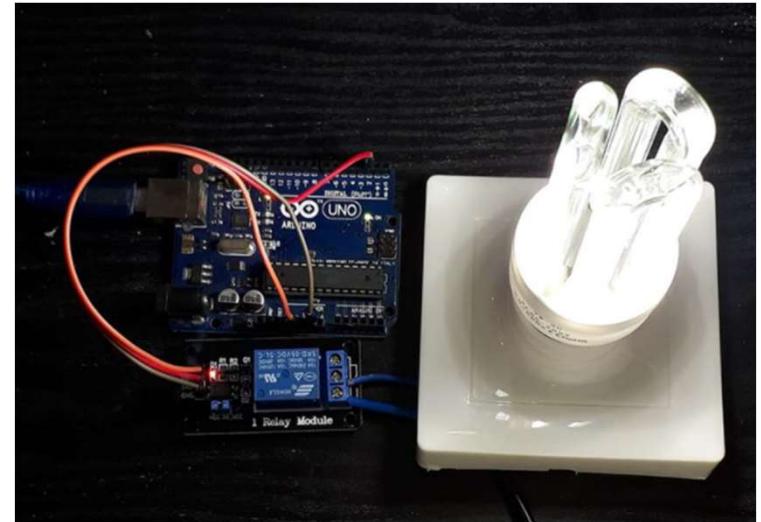
เปิดปิด ผ่านมือถือได้ทุกที่



แอพติดตั้งบนแอนดรอยด์



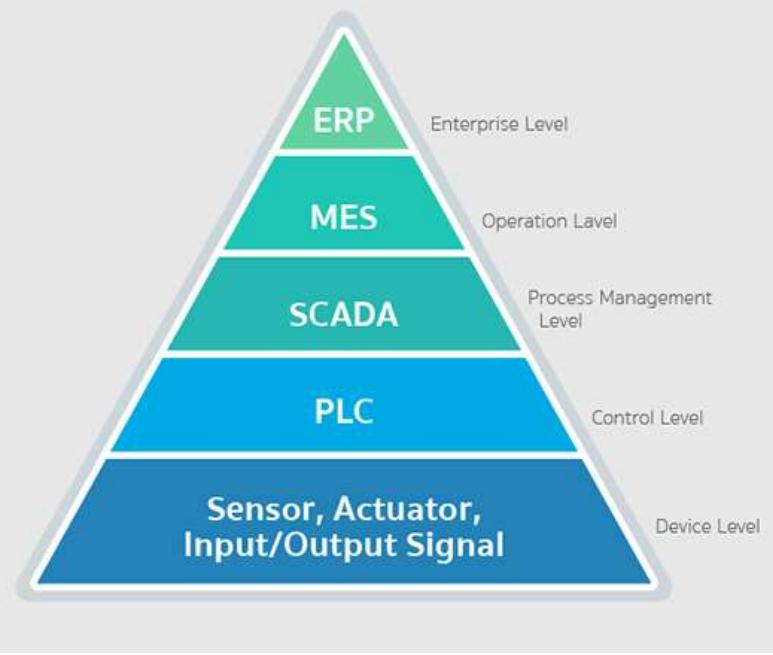
Smart Home (PLC vs IOT)



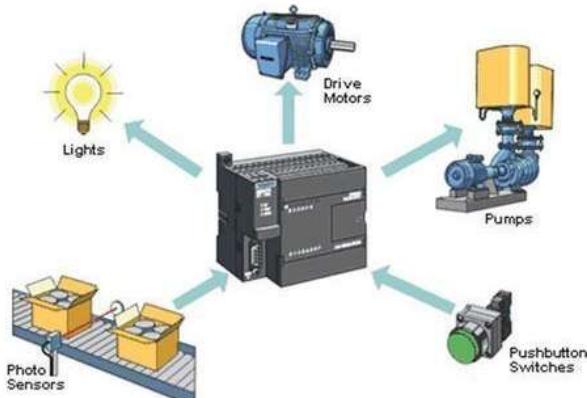
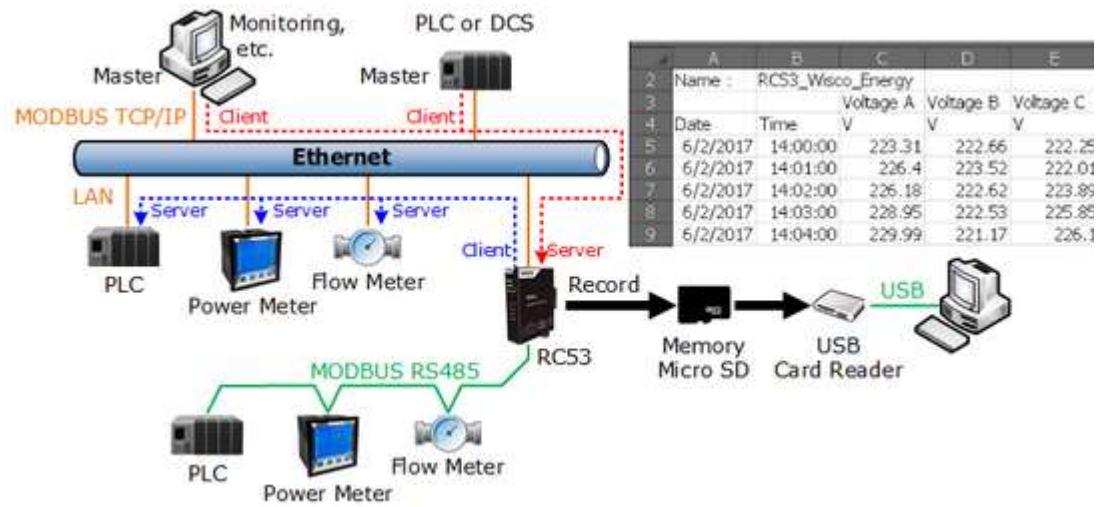
communications and protocols on the control system level

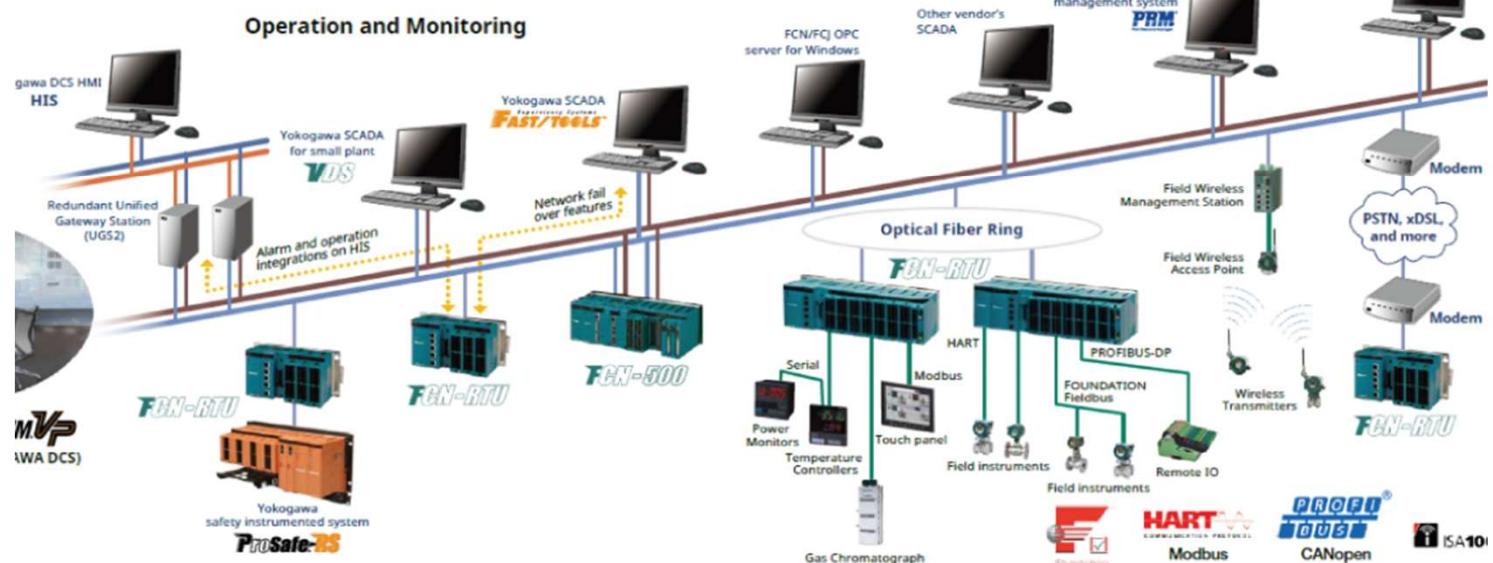
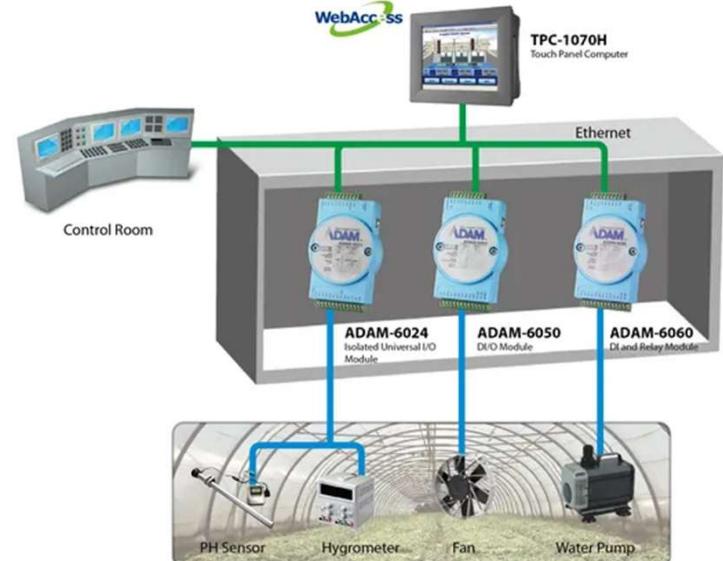
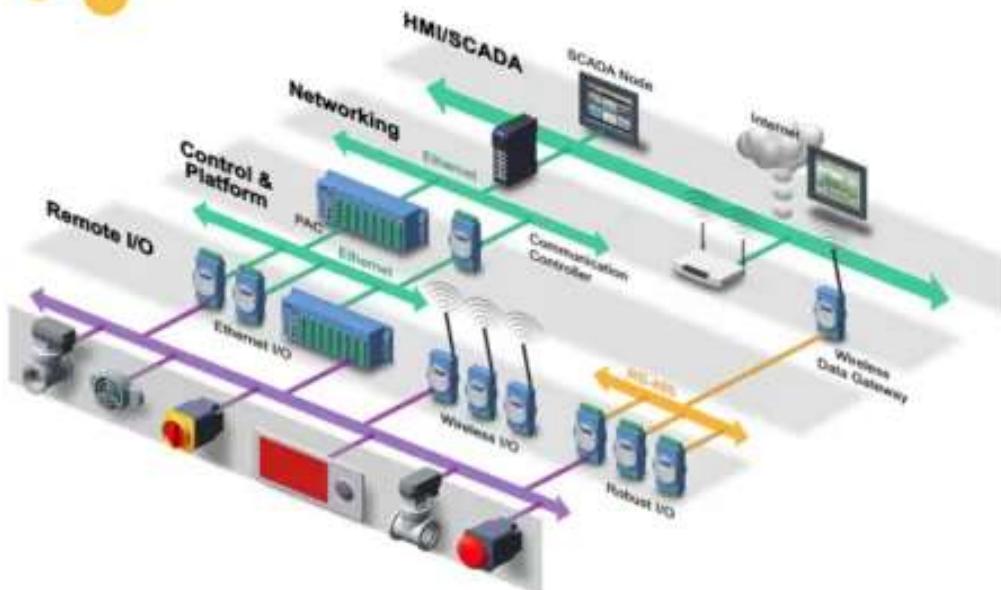
Protocol	Level	Common Applications
ModBus	Device	Manufacturing, Electric Utility
Profibus	Device	Process Industry
DeviceNet	Device	Manufacturing
DNP 3.0	Device	Electric Utility SCADA
BACNet	Control	HVAC Control, Building Automation
ControlNet	Control	Manufacturing
ARCNet	Supervisory	Office Automation,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

Automation Pyramid (Automation Hierarchy)



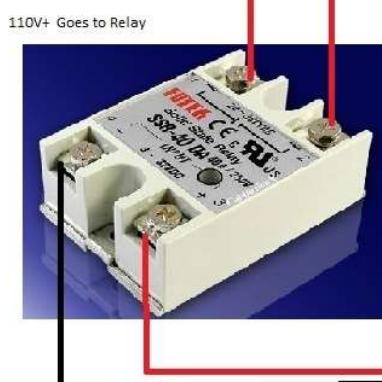
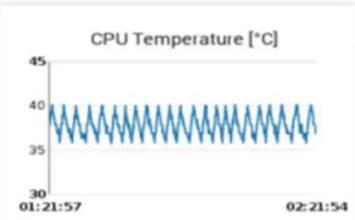
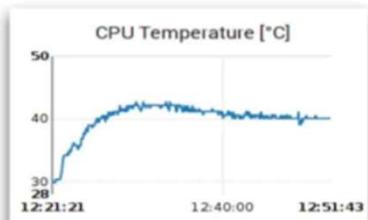
Protocol	Level	Common Applications
ModBus	Device	Manufacturing, Electric Utility
Profibus	Device	Process Industry
DeviceNet	Device	Manufacturing
DNP 3.0	Device	Electric Utility SCADA
BACNet	Control	HVAC Control, Building Automation
ControlNet	Control	Manufacturing
ARCNet	Supervisory	Office Automation,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





Temperature PID Control

PID control VS On/Off control



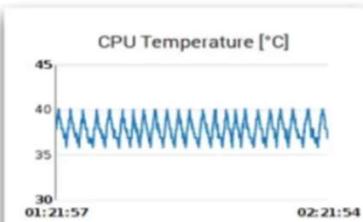
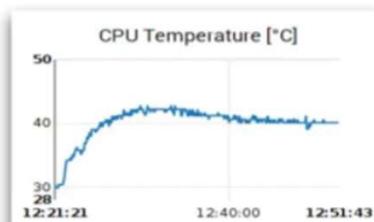
Modbus RTU



[DTK4848V12](#)

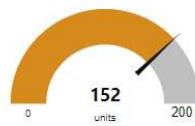
Red and Black Wires for Stock Spindle goes to Relay |

PID control VS On/Off control



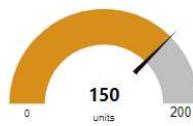
Group 1

PV



152

SV



ปิด เปิด
OnOff

1

Set SV Control
150

Set Upper
150

Frying C
4

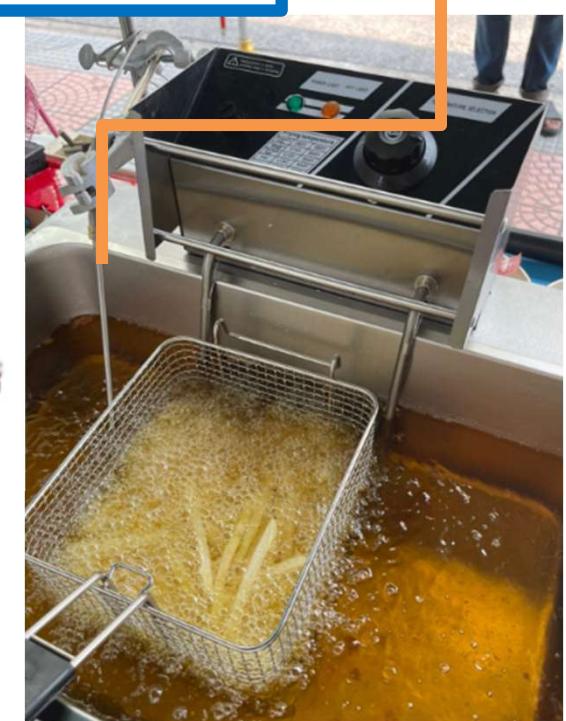
RESE

chart



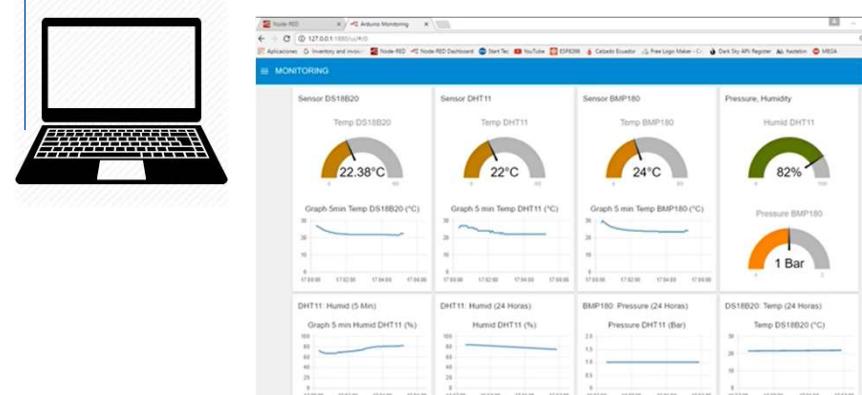
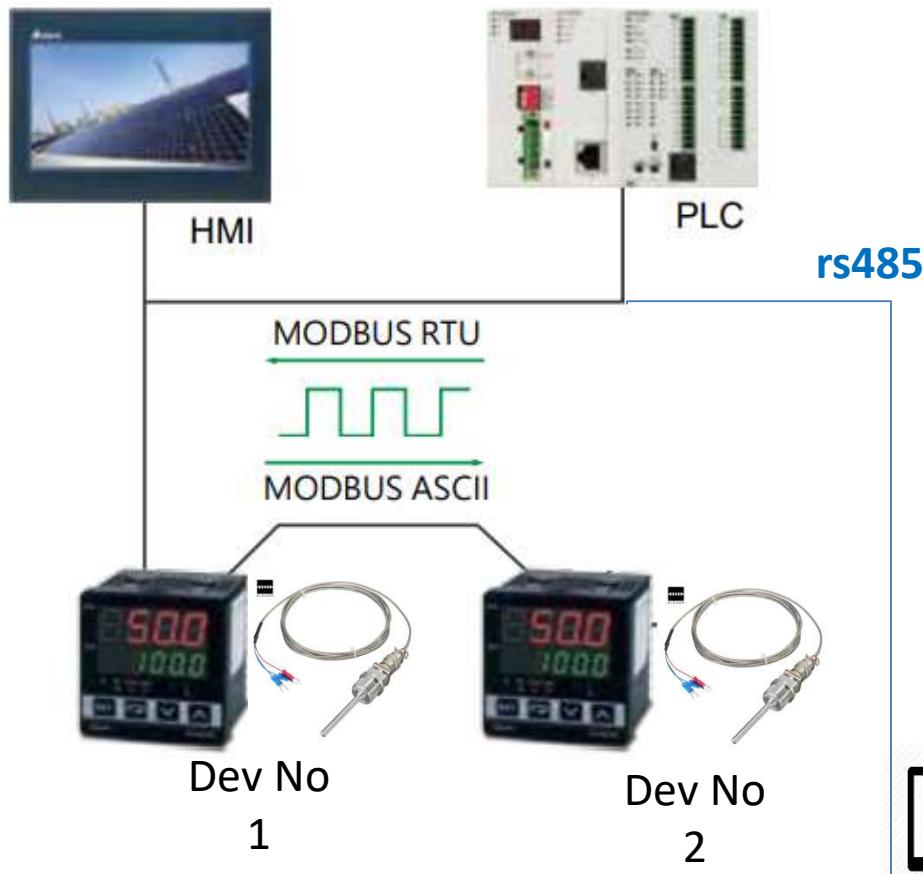
Cur Temp

set Current Temp 152

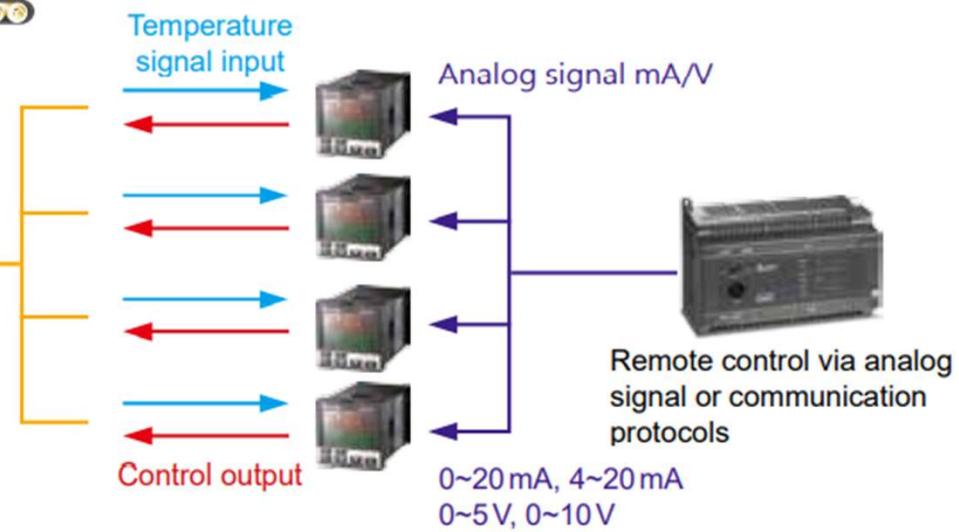
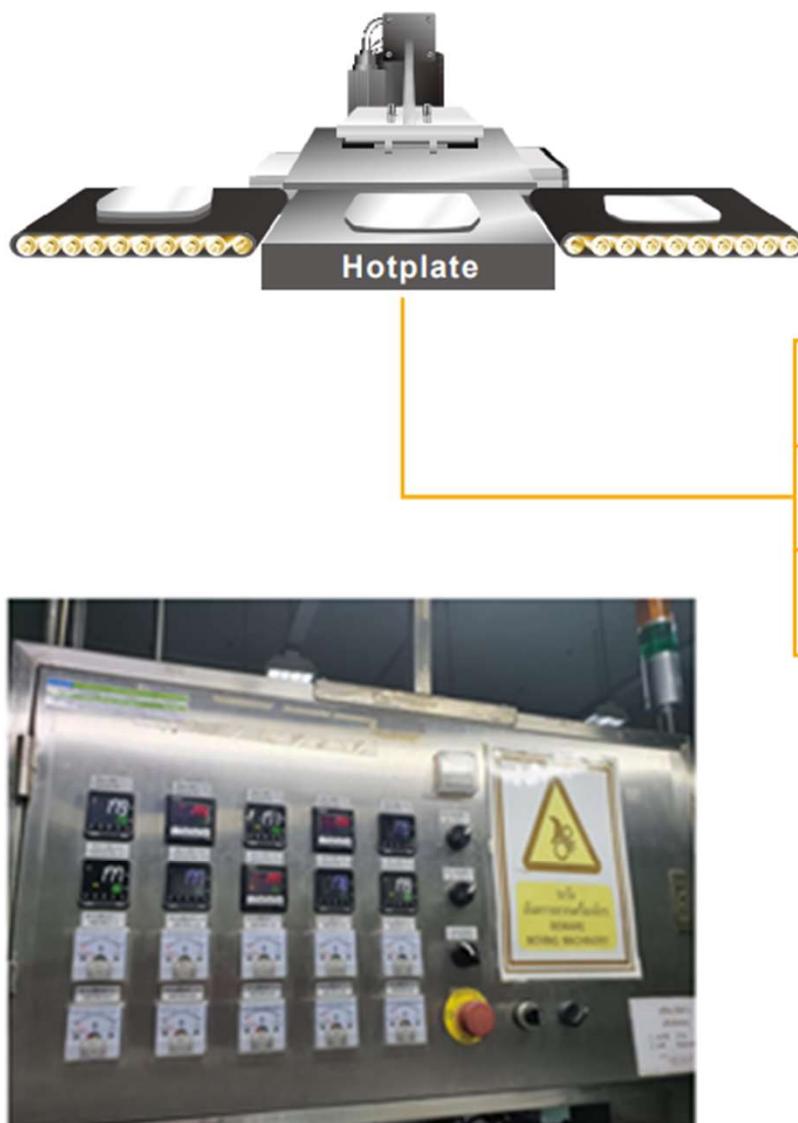


Communication:

- RS-485 communication interface, supporting MODBUS ASCII/RTU communication



Temperature Controller



■ Point-to-point Control (Proportional Output mA/V)

Sets the Present Value by point-to-point control.



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 st Heating/ Cooling control cycle	0.1 ~ 99 sec.
1008H	2 nd 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



HMI

PLC

MODBUS RTU



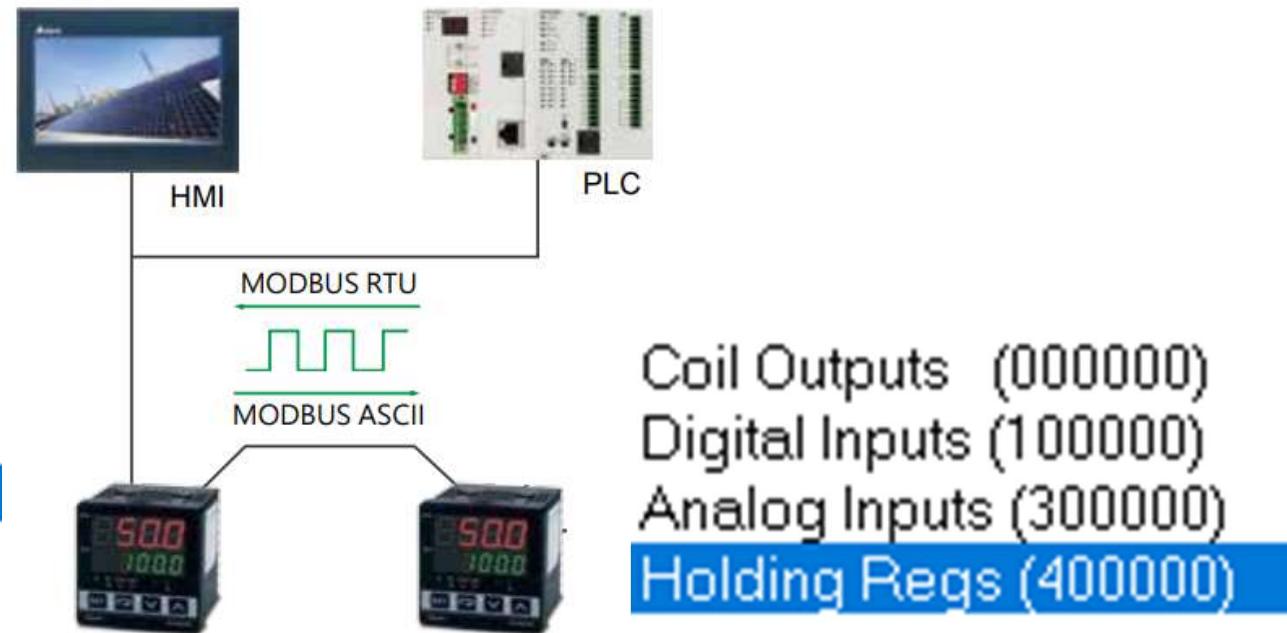
MODBUS ASCII



Modbus RTU

Communication:

- RS-485 communication interface, supporting MODBUS ASCII/RTU communication



Coil Outputs (000000)

Digital Inputs (100000)

Analog Inputs (300000)

Holding Regs (400000)

400000 (FC 03) + 4096 (1000H)

Address	Content
1000H	Present value (PV)
1001H	Set value (SV)

400000 (FC 03) + 4097 (1001H)

Coil Outputs (000000)

Digital Inputs (100000)

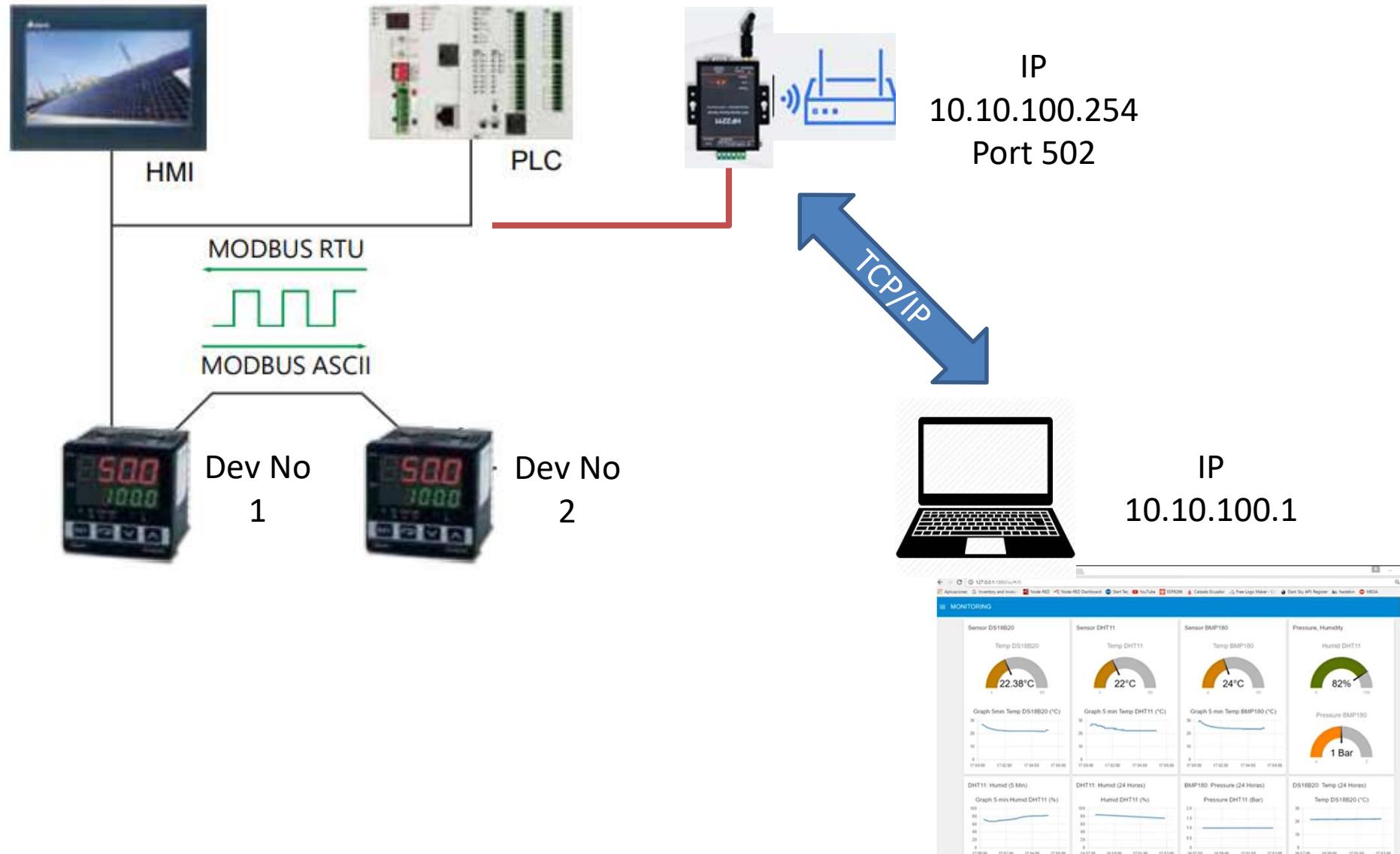
Analog Inputs (300000)

Holding Regs (400000)

Address	Content
1000H	Present value (PV)
1001H	Set value (SV)

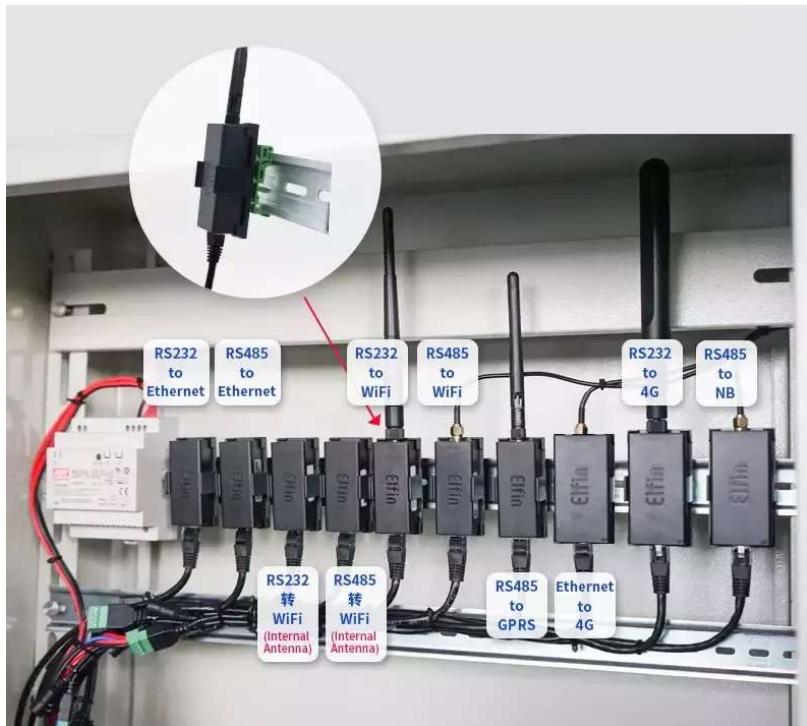
Communication:

- RS-485 communication interface, supporting MODBUS ASCII/RTU communication

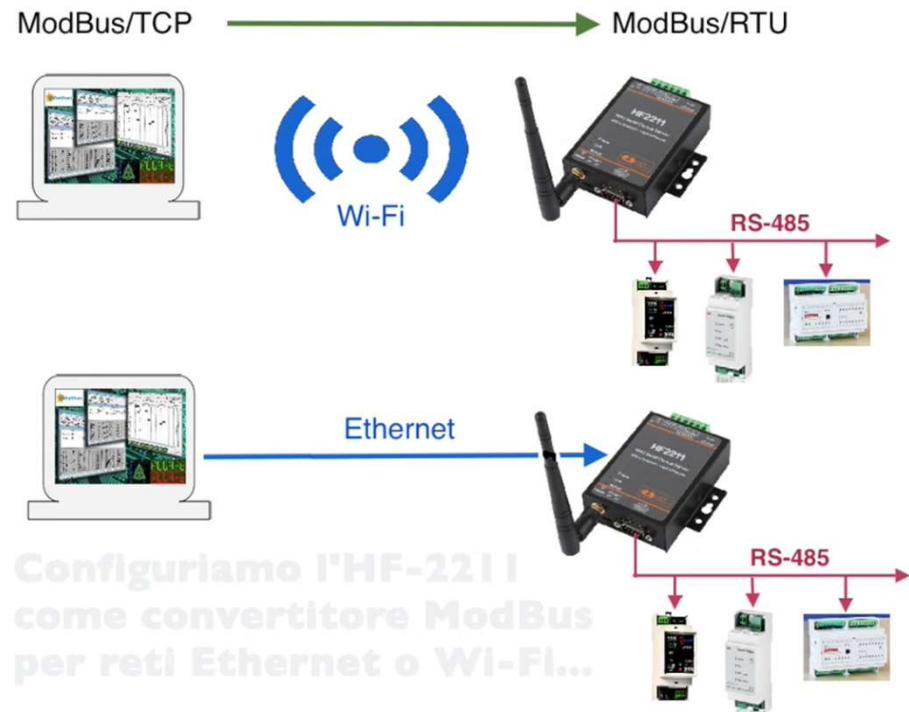


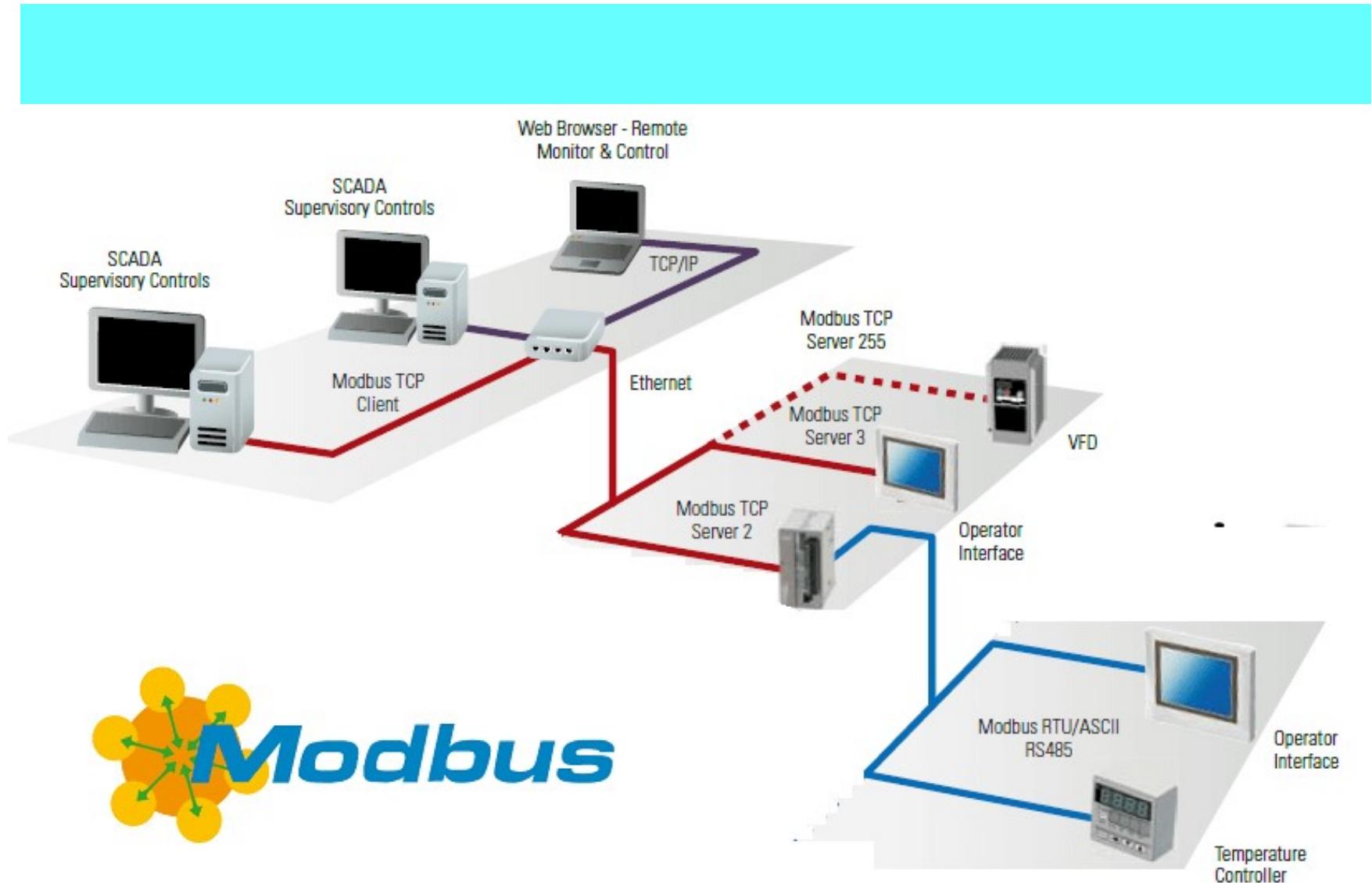
<http://www.hi-flying.com/>

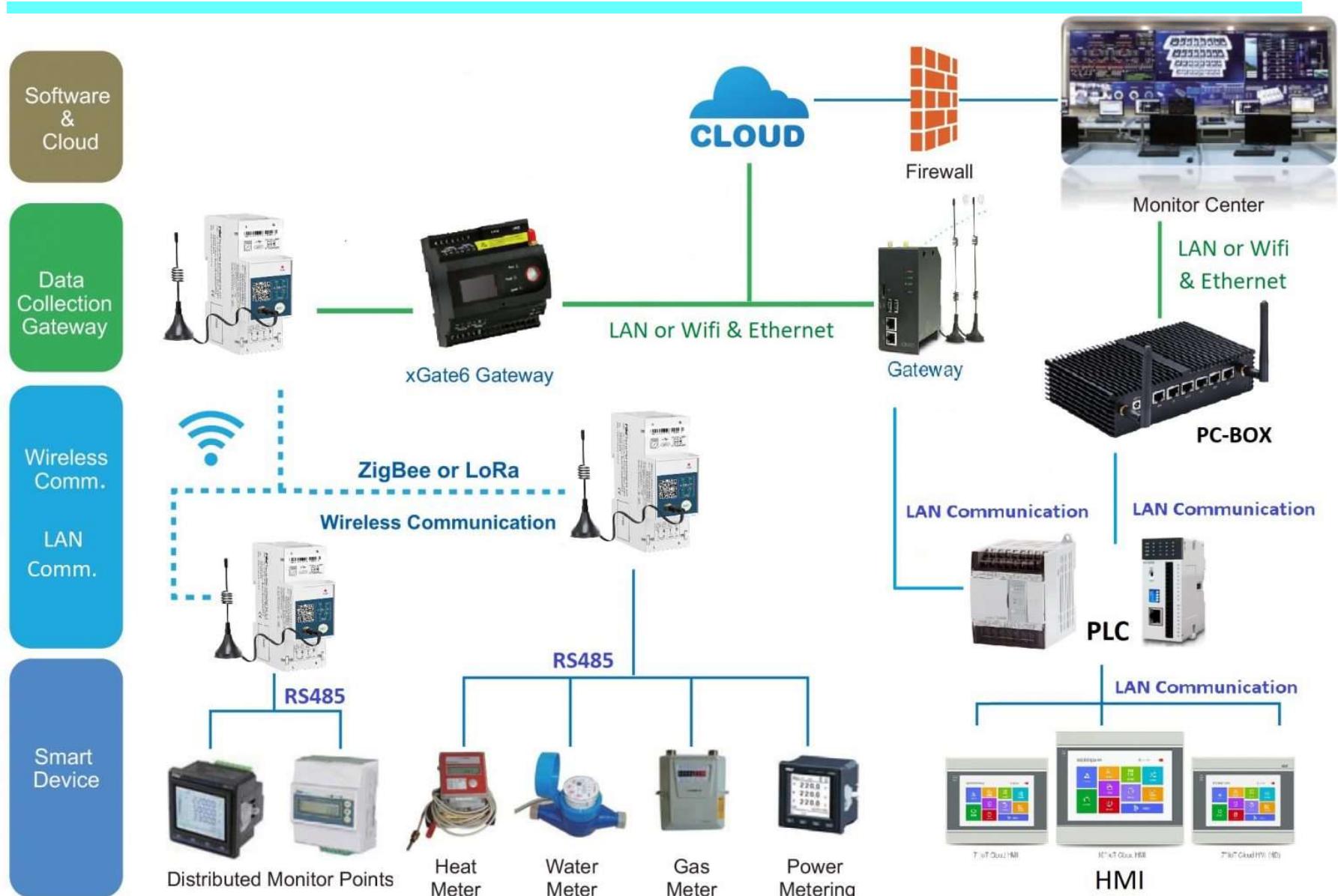
Elfin Series Assembly Drawing

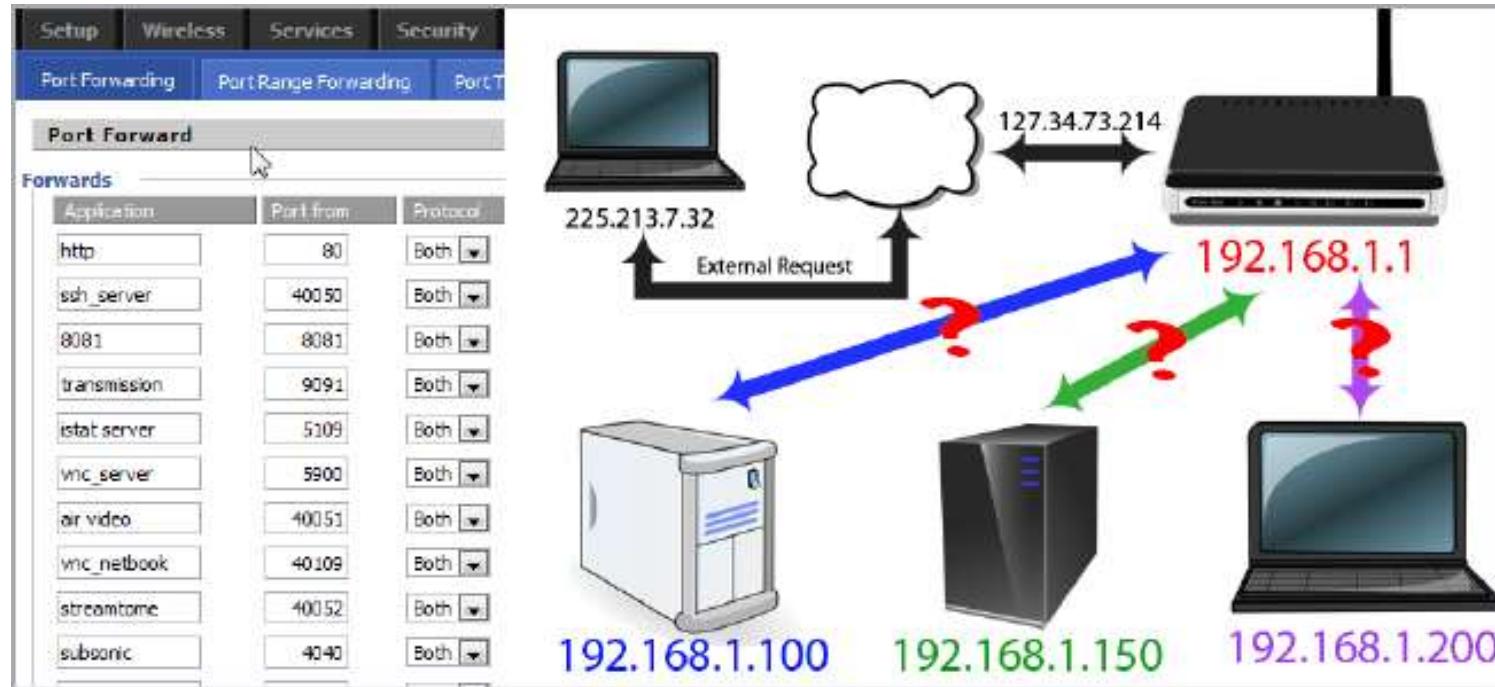


<https://www.youtube.com/watch?v=iSv0HaSmPro>









```
Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

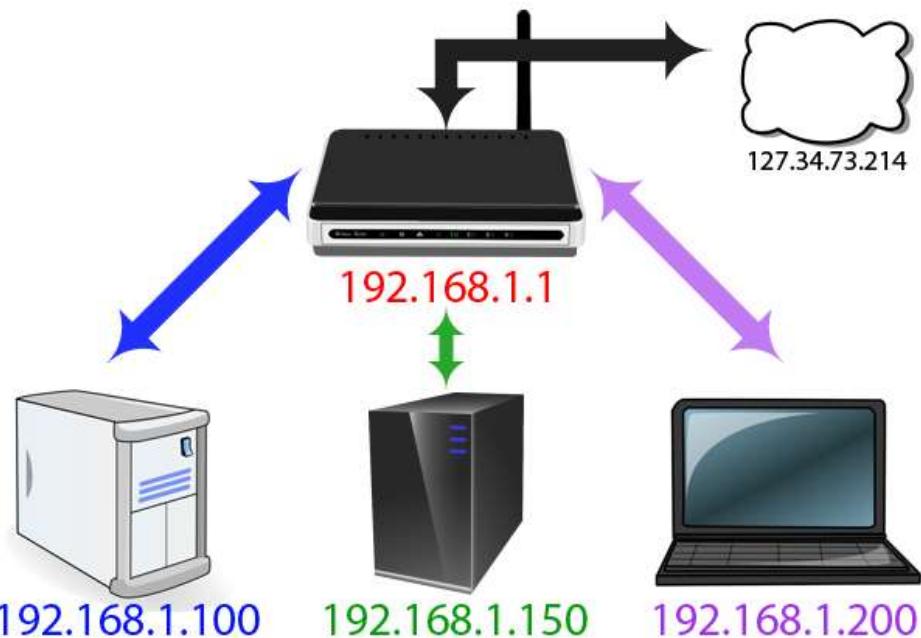
J:\>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

          Connection-specific DNS Suffix  : 10.135.151.39
          IP Address. . . . . : 10.135.151.39
          Subnet Mask . . . . . : 255.255.255.0

J:\>
```



```
C:\>Documents and Settings\demo>ipconfig
```

Windows IP Configuration

Ethernet adapter Local Area Connection:

```
Connection-specific DNS Suffix . . . : racom.cz
IP Address . . . . . : 192.168.169.160
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
```

```
C:\>Documents and Settings\demo>ping 192.168.169.169
```

Pinging 192.168.169.169 with 32 bytes of data:

```
Reply from 192.168.169.169: bytes=32 time<1ms TTL=64
```

Internet Protocol (TCP/IP) Properties

General Alternate Configuration

If this computer is used on more than one network, enter the alternate IP settings below.

Automatic private IP address

User configured

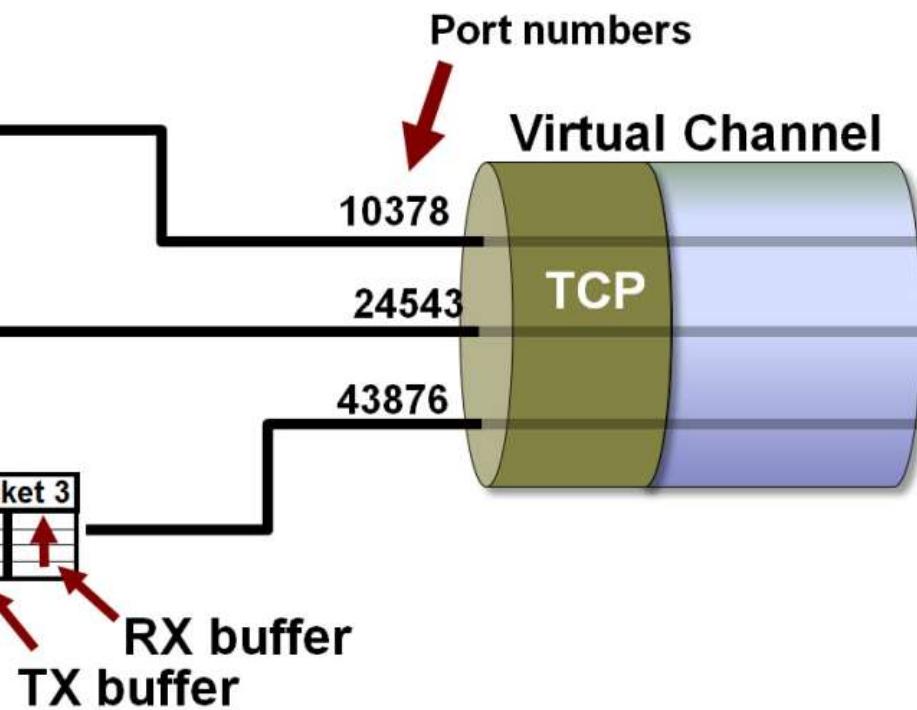
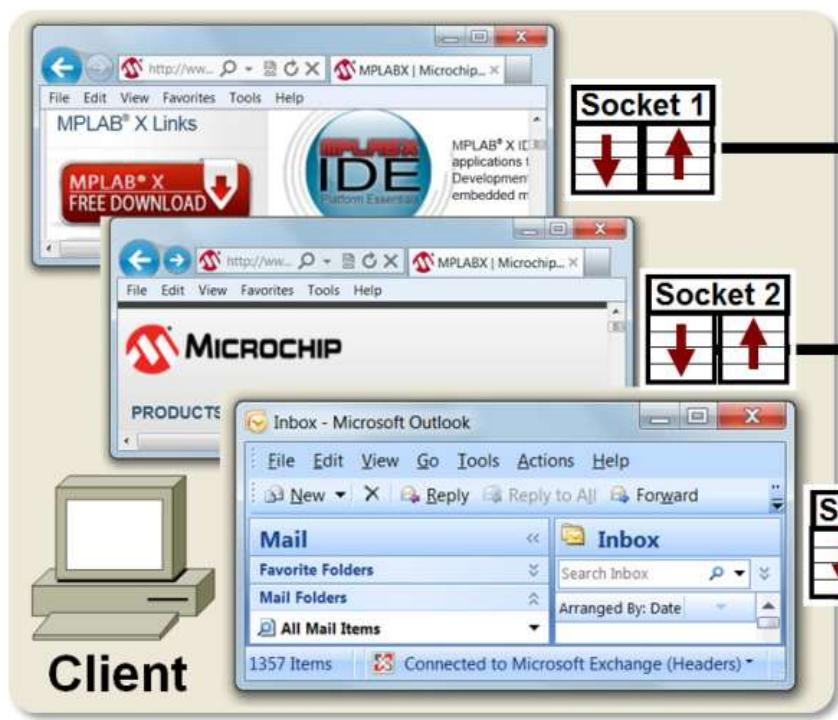
IP address: 192.168.169.160

Subnet mask: 255.255.255.0

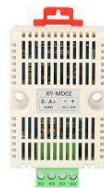
Default gateway:



192.168.1.100



การใช้งานโปรแกรม SimModbus 2 จำลอง Modbus Memory Device (TCP)



ตำแหน่งหน่วยความจำ ภายในอุปกรณ์

000001 - 065535 Coil Output

100000 -> 165535 Digital Input

300000 – 365535 Analog Input

400000 – 465535 Holding Reg

Read / Write

Read Only

เก็บค่าตัวเลข
0 และ 1

เก็บค่าจำนวนเต็ม
(0- FFFFH)
(0- 65535)

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

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

Address: H D I/O Coil Outputs (000000) Fmt: decimal +/- Prot: MODBUS TCP/IP Clone Log

Address	+0	+1	Digital Inputs (100000)	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	Total
000001-000016	1	1	Digital Inputs (100000)	0	0	0	0	0	0	0	0	0	0	0003
000017-000032	0	0	Analog Inputs (300000)	0	0	0	0	0	0	0	0	0	0	0000
000033-000048	0	0	Holding Regs (400000)	0	0	0	0	0	0	0	0	0	0	0000
000049-000064	0	0	Extended Registers	0	0	0	0	0	0	0	0	0	0	0000
000065-000080	0	0		0	0	0	0	0	0	0	0	0	0	0000
000081-000096	0	0		0	0	0	0	0	0	0	0	0	0	0000
000097-000112	0	0		0	0	0	0	0	0	0	0	0	0	0000
000113-000128	0	0		0	0	0	0	0	0	0	0	0	0	0000
000129-000144	0	0		0	0	0	0	0	0	0	0	0	0	0000
000145-000160	0	0		0	0	0	0	0	0	0	0	0	0	0000

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

Ver. 8.21.2.7

การใช้งานโปรแกรม SimModbus 2

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

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

Address: C H D I/O Holding Regs (400000) Rx: Tx: Fmt: decimal +/- Prot: MODBUS TCP/IP Clone Log

Address	+0	+3	+4	+5	+6	+7	+8	+9
400001-400010	0	0	0	0	0	0	0	0
400011-400020	0	0	0	0	0	0	0	0
400021-400030	0	0	0	0	0	0	0	0
400031-400040	0	0	0	0	0	0	0	0
400041-400050	0	0	0	0	0	0	0	0
400051-400060	0	0	0	0	0	0	0	0
400061-400070	0	0	0	0	0	0	0	0
400071-400080	0	0	0	0	0	0	0	0
400081-400090	0	0	0	0	0	0	0	0
400091-400100	0	0	0	0	0	0	0	0
400101-400110	0	0	0	0	0	0	0	0
400111-400120	0	0	0	0	0	0	0	0
400121-400130	0	0	0	0	0	0	0	0
400131-400140	0	0	0	0	0	0	0	0
400141-400150	0	0	0	0	0	0	0	0

Ethernet TCP/IP Settings

Status: Supporting 10 simultaneous connections.

Local IP: Chalermchon19

Remote IP: IP Your Com (127.0.0.1)

OK Cancel

Server settings

Server connections: 10

Port (502): 502

Alternate port: 501

Socket Timeout (sec): 100 (10 to 1000 sec)

Responsiveness (ms): 0 (0 to 10 000 ms)

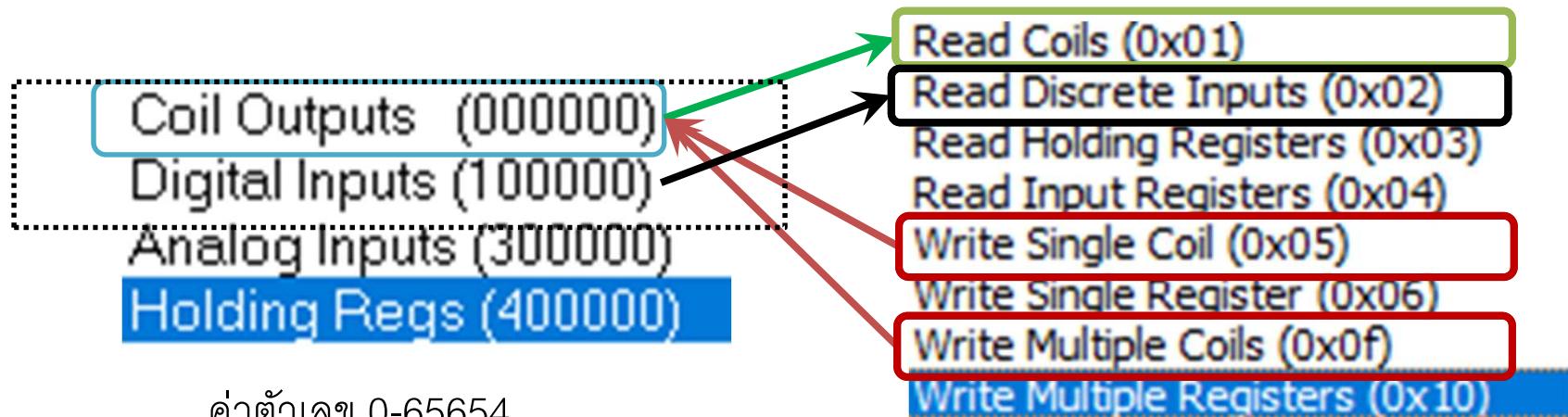
Port 502

Modbus Devices Simulator

Load register values at startup.

Units are all off at startup

Linger on close of socket (SO_LINGER)



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

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

Address: H D I/O Coil Outputs (000000) 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	0005	
000017-000032	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	0000	
000049-000064	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

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

Single HR : 40001 + 0 = **40001**

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

Modbus Mode RTU 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: 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	0005	
000017-000032	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	0000	
000049-000064	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	

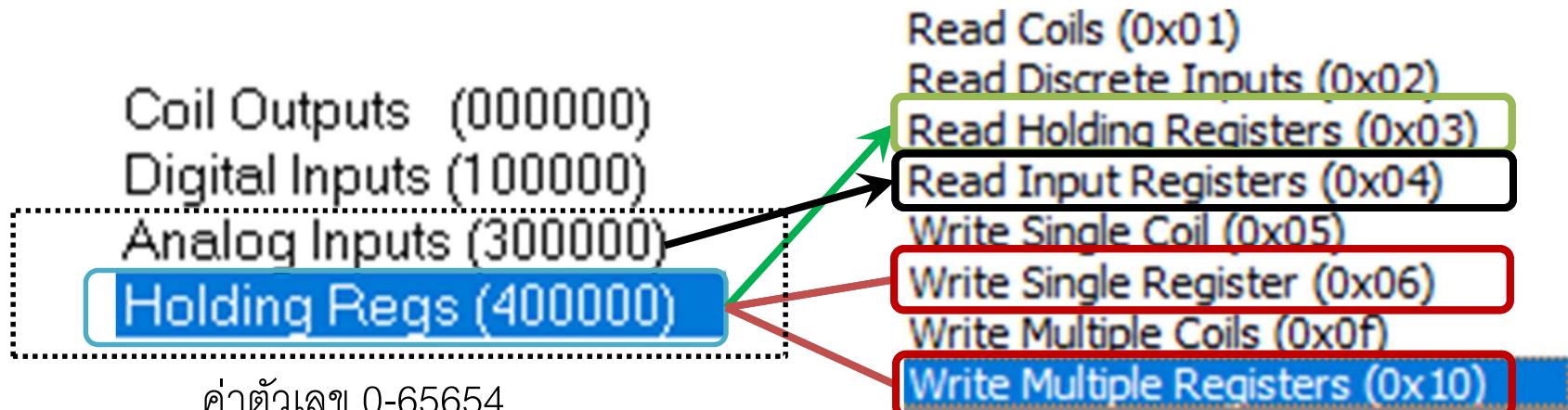
(0x01) Single Coil
(0) => ? = 1

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

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

Single HR : 40001 + 0 = 40001

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



Analog Input (300001-399999) (Read Only) Holding Registers 400001-499999 (Read/Write)

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

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

Address: H D I/O Holding Regs (400000) Fmt: decimal +/- Prot: MODBUS TCP/IP Clone Log

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	▲
400001-400010	256	565	0	0	0	0	0	0	0	0	0
400011-400020	0	0	0	0	0	0	0	0	0	0	0
400021-400030	0	0	0	0	0	0	0	0	0	0	0

(0x03) Single Read
 HR (0) => ? = 256

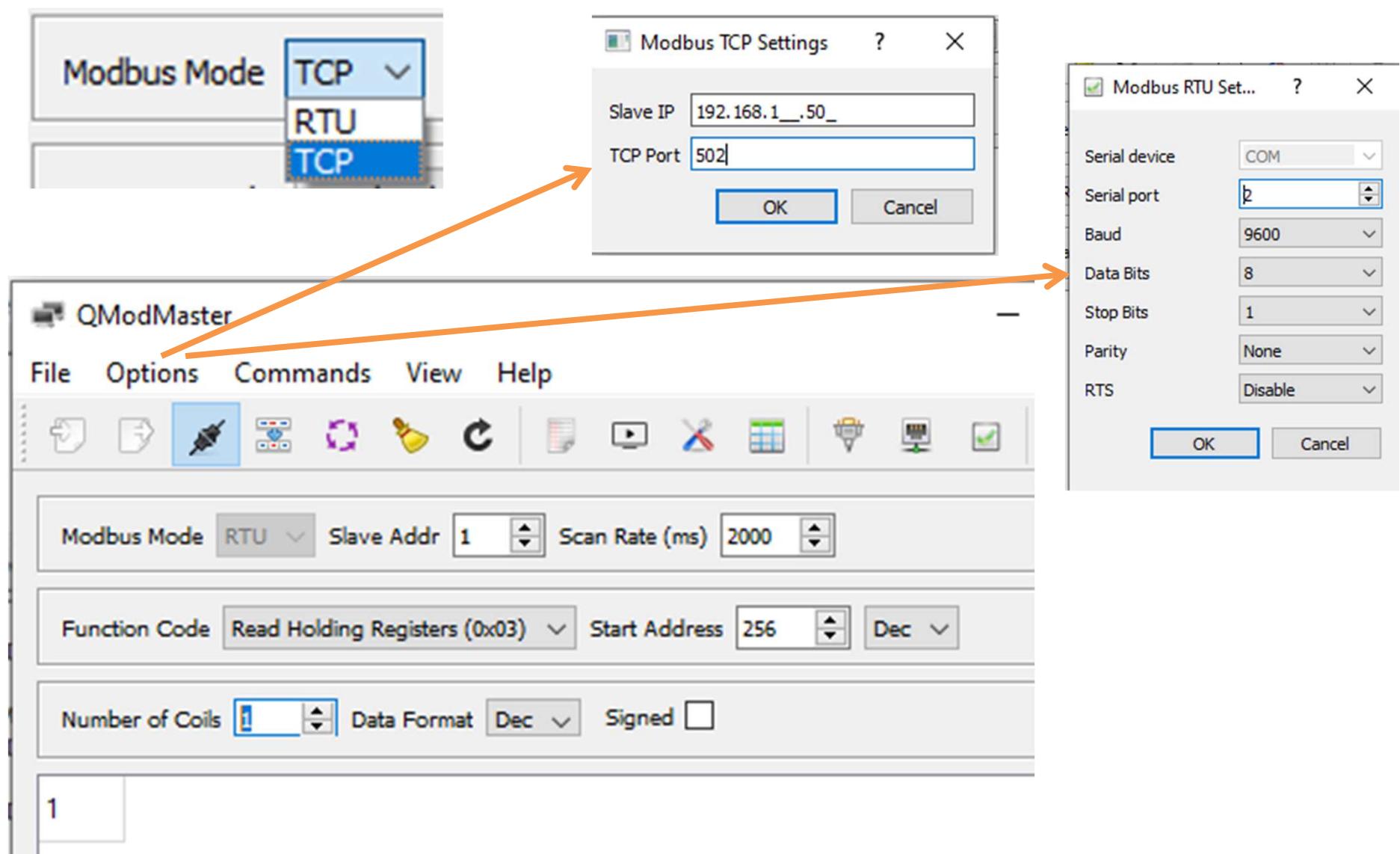
(0x06) Sing Write HR
 (0) <= 302

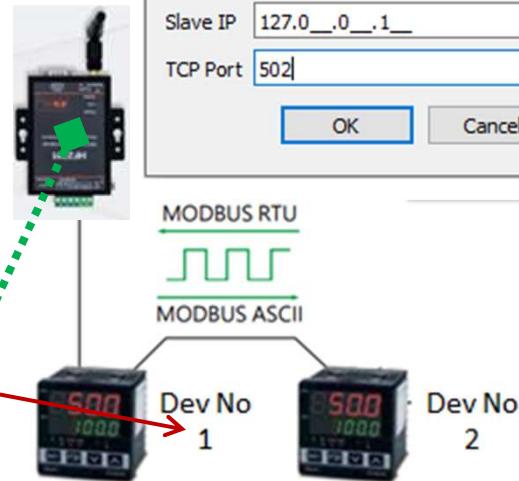
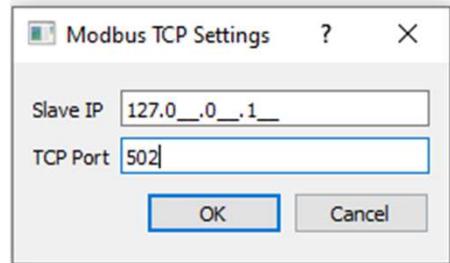
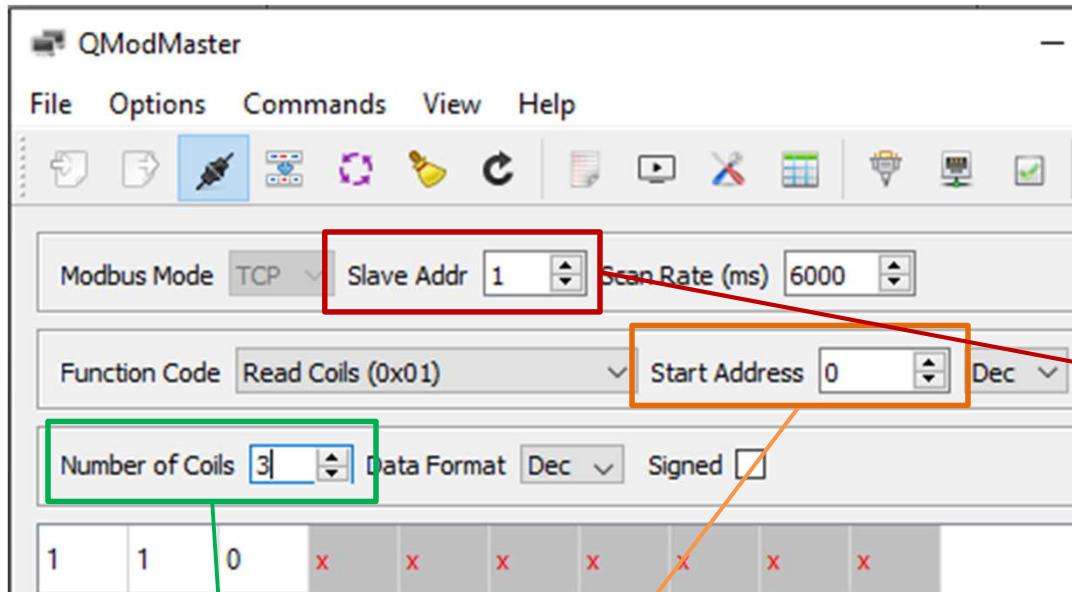
(0x10) Multi Write HR
 (6) <= [3] 302,322,333

Single HR : 40001 + 0 = **40001**

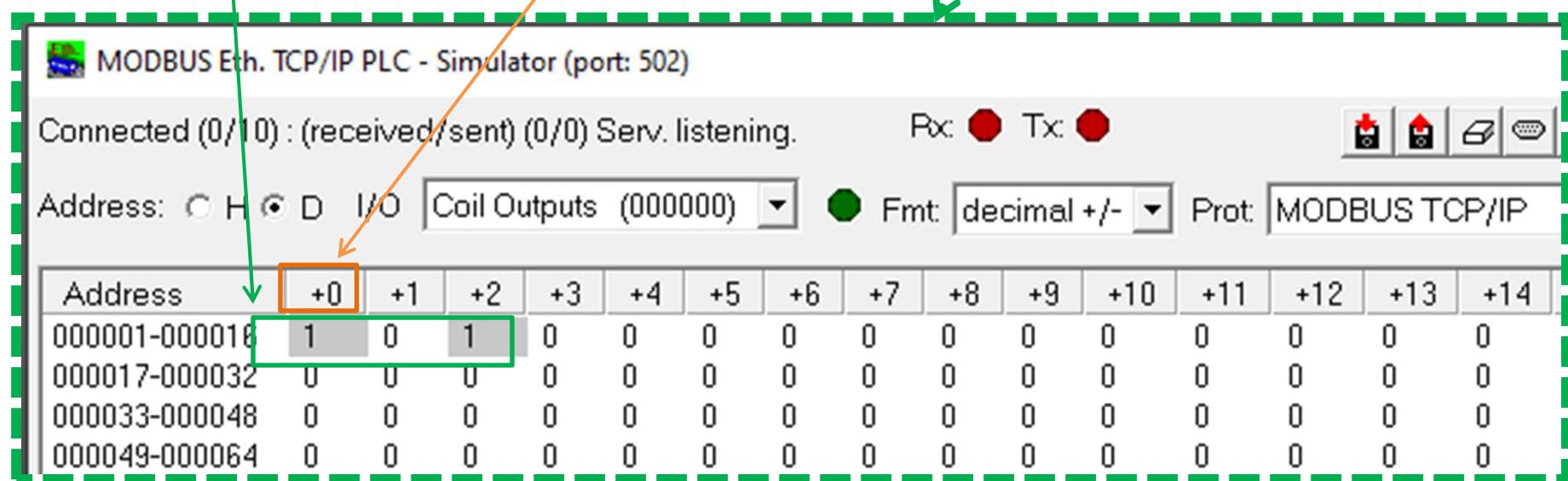
(0x03) Multi Read HR
 (6) => [3] ?,?,? = 0 , 0 , 0

การใช้งานโปรแกรม QModMaster (Modbus Master)

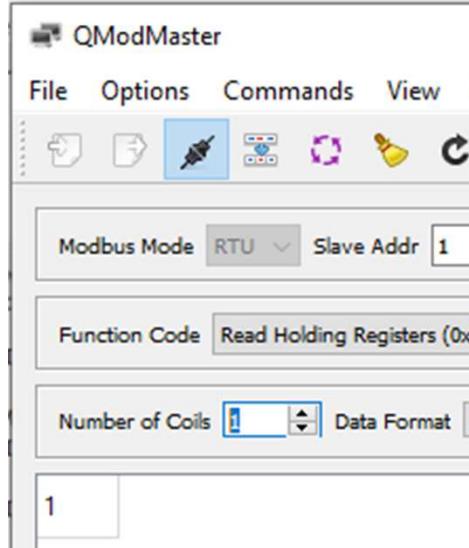




Device Simulator



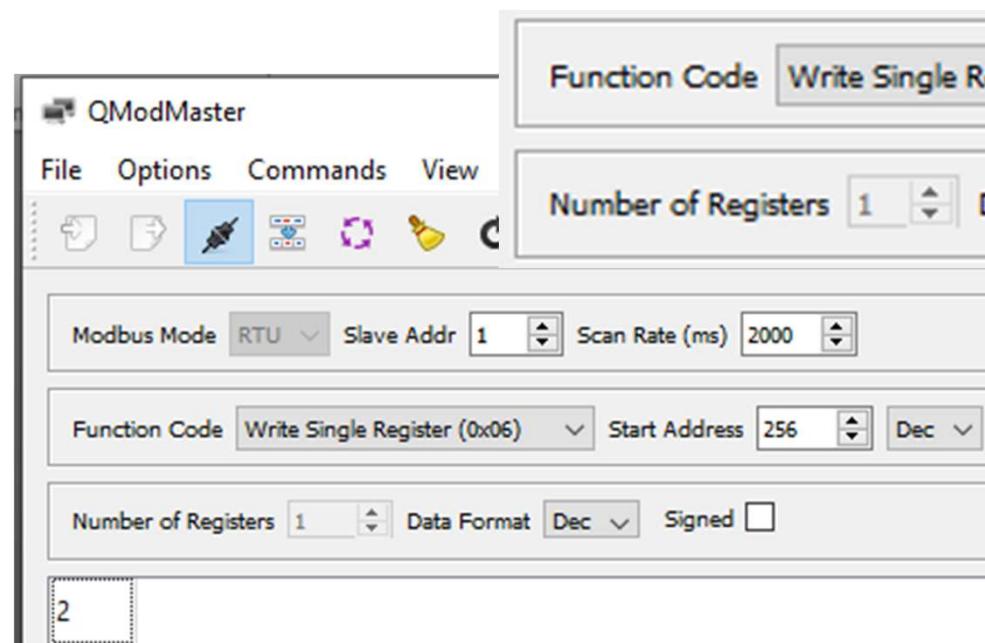
Set Device No



(0x03) Read

Start Adress 256
(Device No)

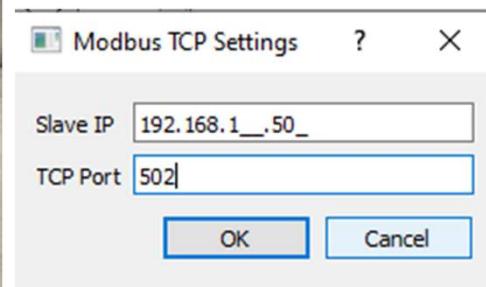
Single , Multiple



(0x06) Write

Start Adress 256
(Device No)# 06 Only Write Single
60 for Multi Write

YouTube Modbus – 03 FactoryIO Conveyor



QModMaster

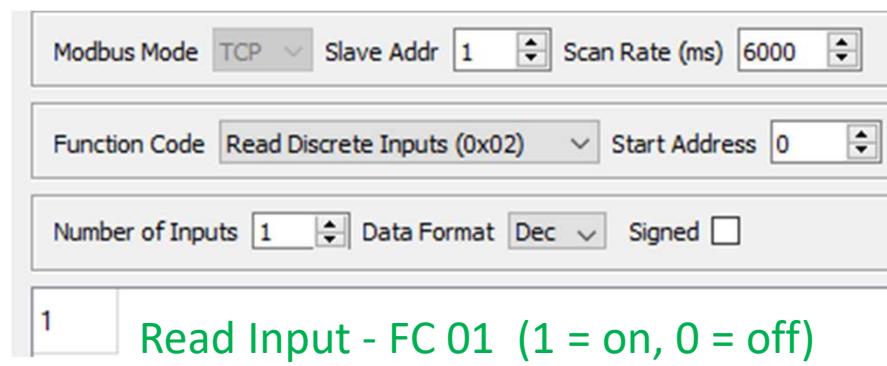
File Options Commands View Help

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 Coil - FC 01 (1 = on, 0 = off)



← DRIVER Modbus TCP/IP Server

SENSORS

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

I/O (Running)

Sensor

(192.168.1.50:502)
Slave ID:1

Input 0

Input 1

ACTUATORS

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

QModMaster

File Options Commands View Help

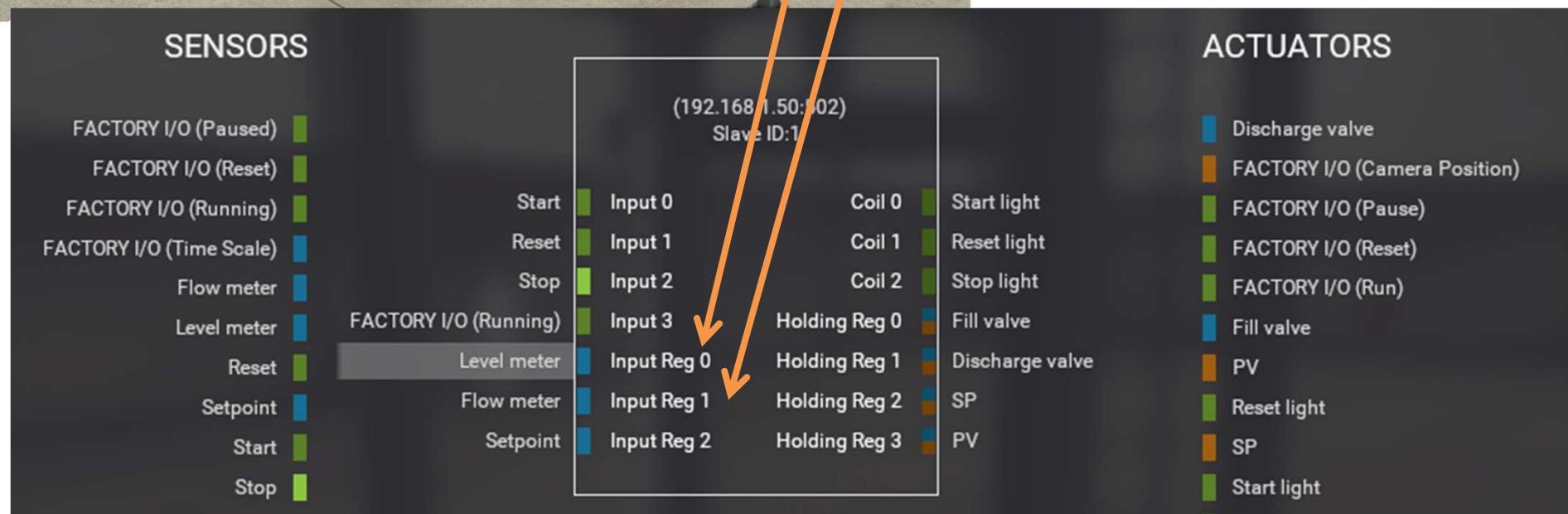
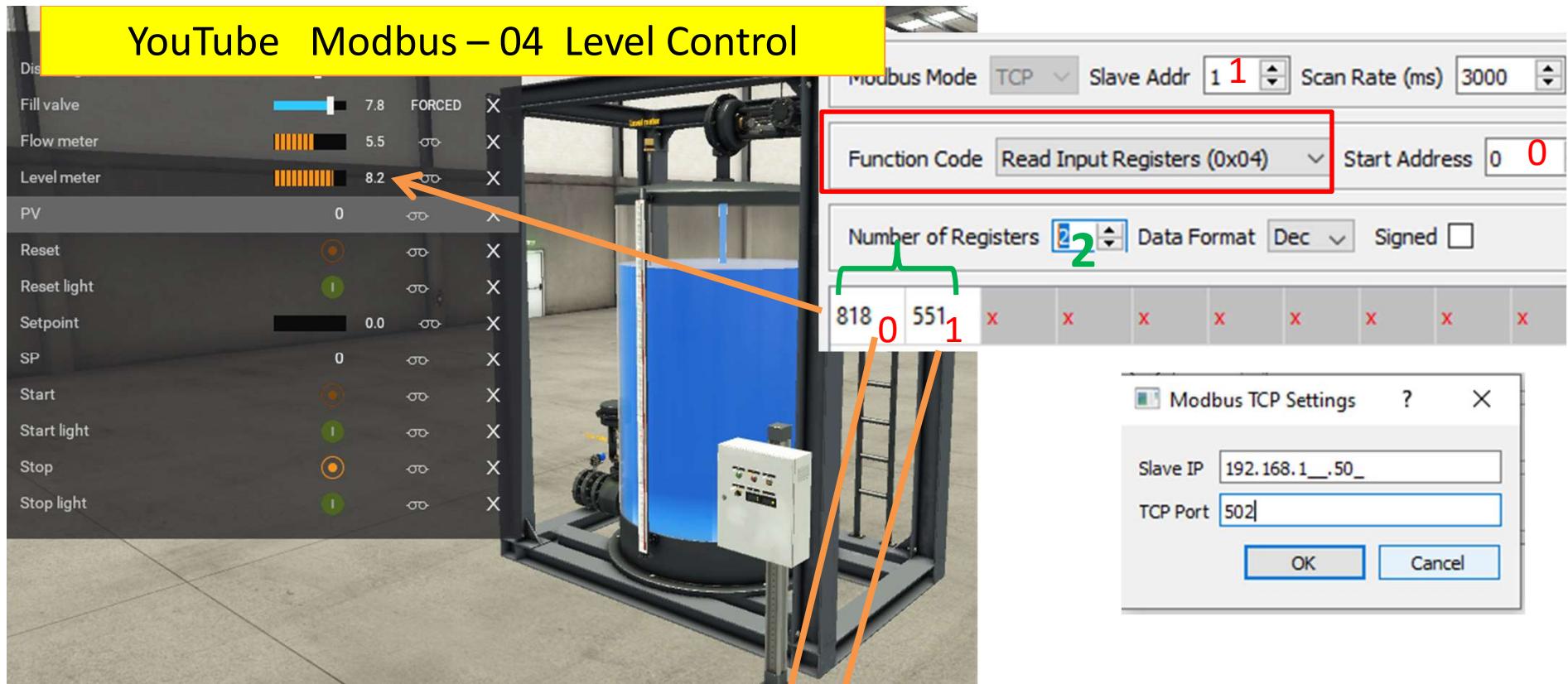
Modbus Mode TCP Slave Addr 1 Scan Rate (ms) 6000

Function Code Write Single Coil (0x05) Start Address 0

Number of Coils 1 Data Format Dec Signed

1 Write Coil - FC 05 (1 = on, 0 = off)

YouTube Modbus – 04 Level Control



การติดต่อ Modbus Machine with QModMaster



Read Input Reg - FC 04 (ตัวเลข)

Modbus Mode	TCP	Slave Addr	1	Scan Rate (ms)	6000
Function Code	Read Input Registers (0x04)	Start Address	0	Data Format	Dec
Number of Registers	2	Signed	<input type="checkbox"/>		
591	384	x	x	x	x



Read Coil - FC 01 (1 = on, 0 = off)

(192.168.1.50:502) Slave ID:1			
Start	Input 0	Coil 0	Start light
Reset	Input 1	Coil 1	Reset light
Stop	Input 2	Coil 2	Stop light
FACTORY I/O (Running)	Input 3	Holding Reg 0	Fill valve
Level meter	Input Reg 0	Holding Reg 1	Discharge valve
Flow meter	Input Reg 1	Holding Reg 2	SP
Setpoint	Input Reg 2	Holding Reg 3	PV

Holding Reg - FC 03 (ตัวเลข)

Modbus TCP Settings	?	X
Slave IP	192.168.1.50	
TCP Port	502	
OK	Cancel	

Modbus Mode	TCP	Slave Addr	1	Scan Rate (ms)	6000
Function Code	Read Holding Registers (0x03)	Start Address	0	Data Format	Dec
Number of Registers	4	Signed	<input type="checkbox"/>		
500	500	0		Write Coil - FC 05	(1 = on, 0 = off)

FC 03 Read / FC 06 Write Single (HR)

The screenshot shows a software interface for Modbus communication. At the top, there is a list of four devices: Discharge valve, Fill valve, Flow meter, and Level meter. Below this is a configuration section with the following fields:

- 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:

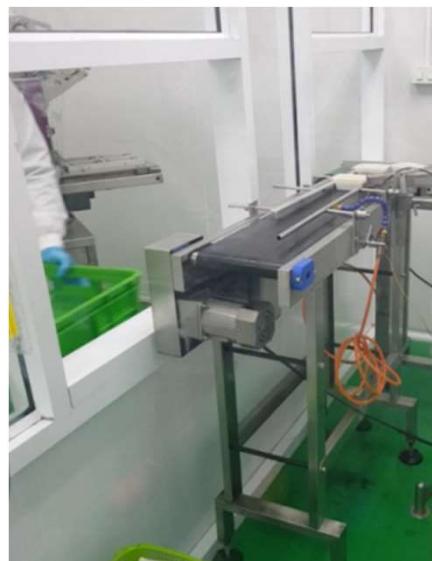
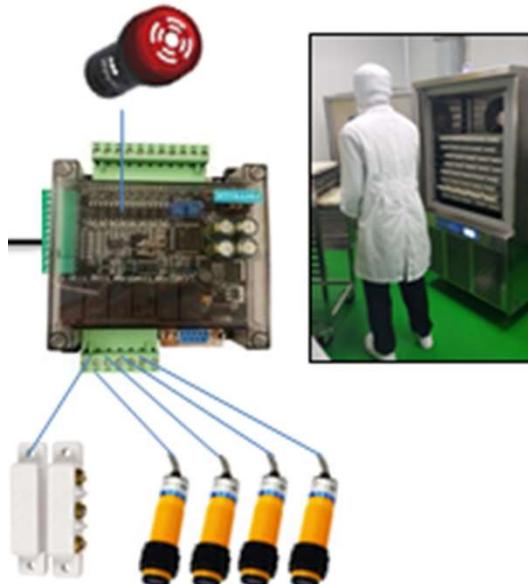
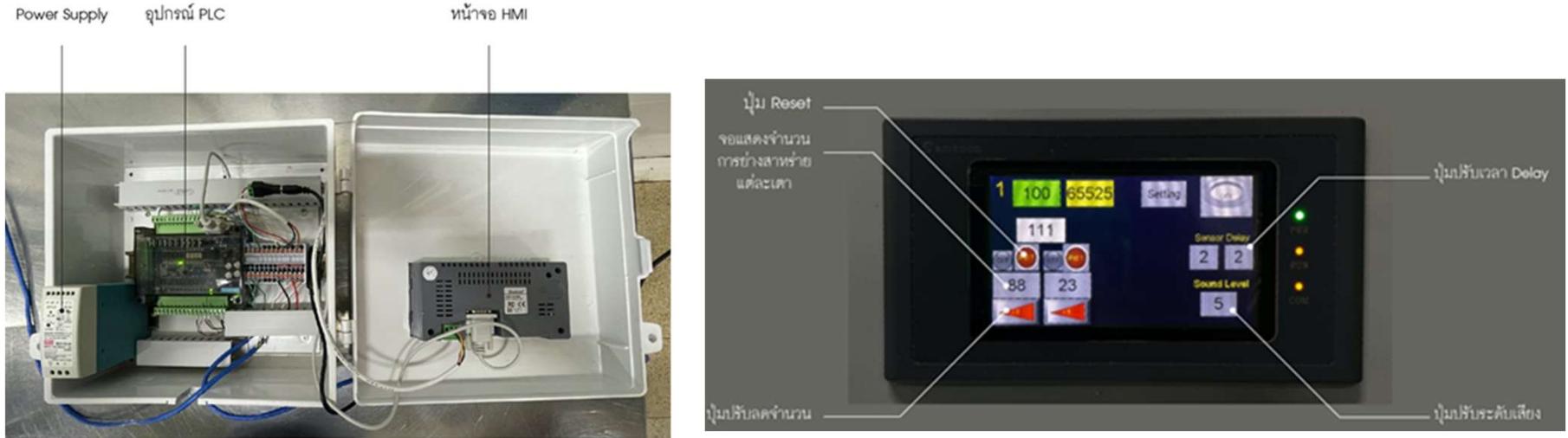
A large input field contains the value "500". Below this is another set of configuration fields for writing:

- 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:

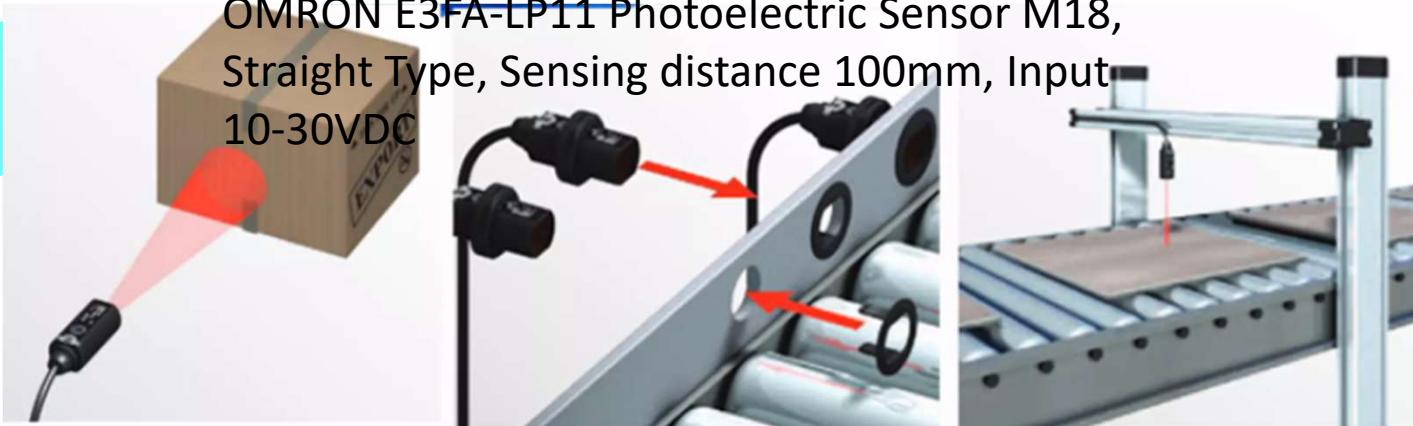
A second large input field also contains the value "500". At the bottom right, there is a legend mapping register numbers to device names:

Holding Reg 0	Fill valve
Holding Reg 1	Discharge valve

A photograph of a large industrial storage tank or reactor. The tank is cylindrical with a vertical ladder on its side. A control panel is visible on the right side of the tank. The tank appears to be made of metal and has some markings or labels on it.



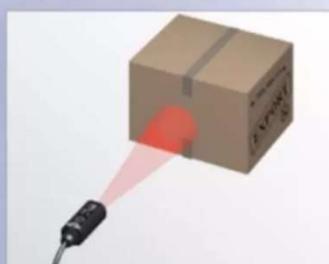
OMRON E3FA-LP11 Photoelectric Sensor M18,
Straight Type, Sensing distance 100mm, Input
10-30VDC



Unrivaled Detection with Simplicity in Setup and Installation



The short body of the E3FA/E3RA fits in tighter mounting spaces.



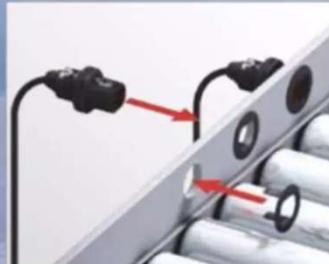
Visible red LED light for easy alignment.



Transparent object detection sensors utilize Omron's unique technology for detecting objects with birefringent (double refraction) properties.



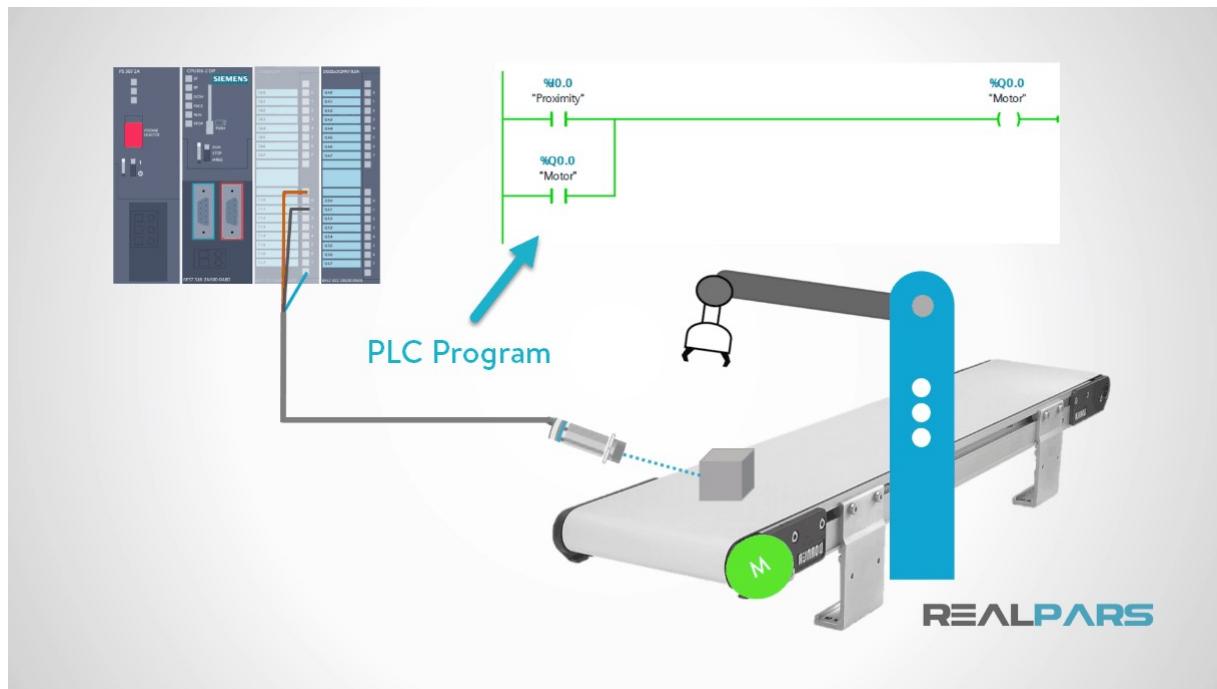
Bright LED Indicators for status visibility and large sensor adjustors for use with a standard size screwdriver.



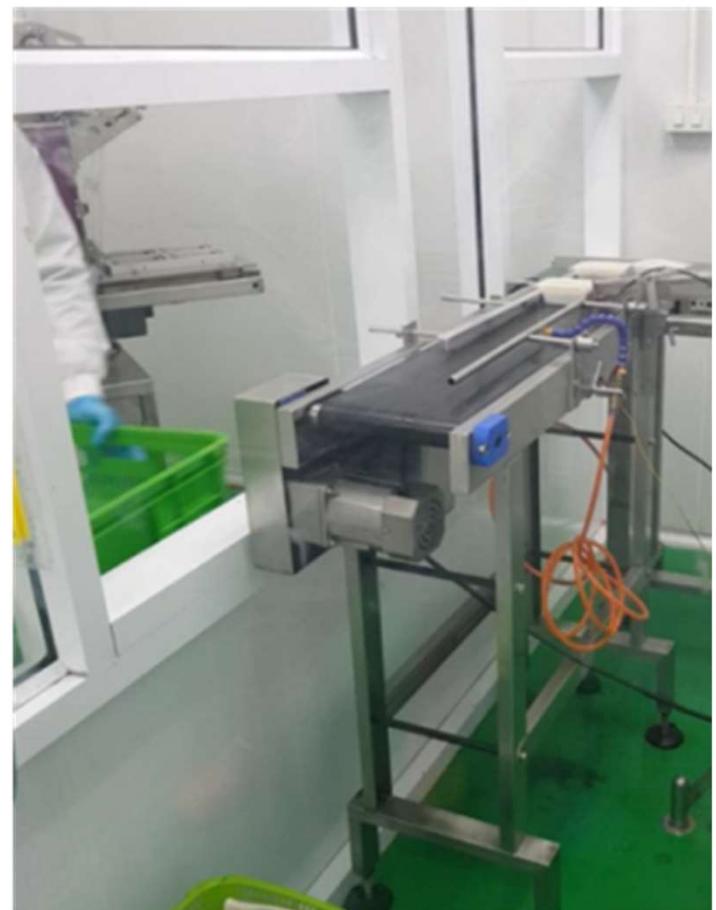
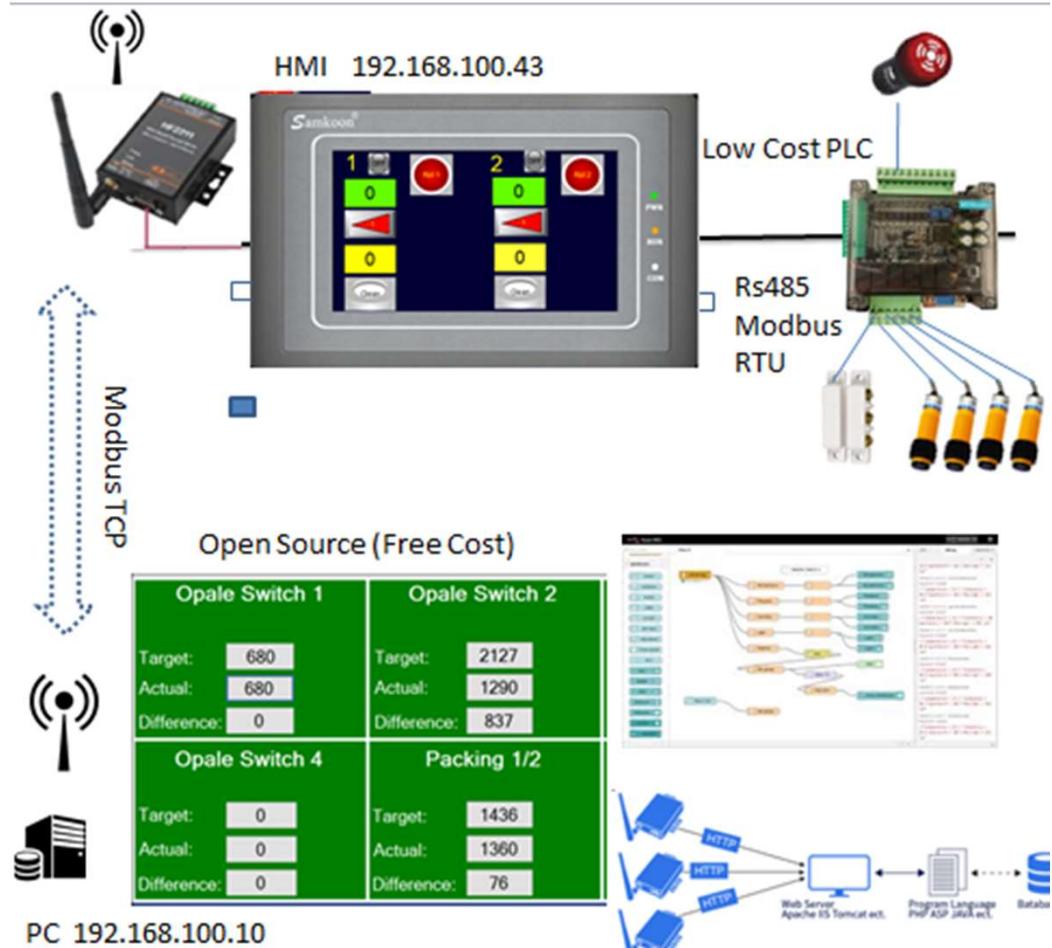
Flush, snap mounting option for quick and easy installation.



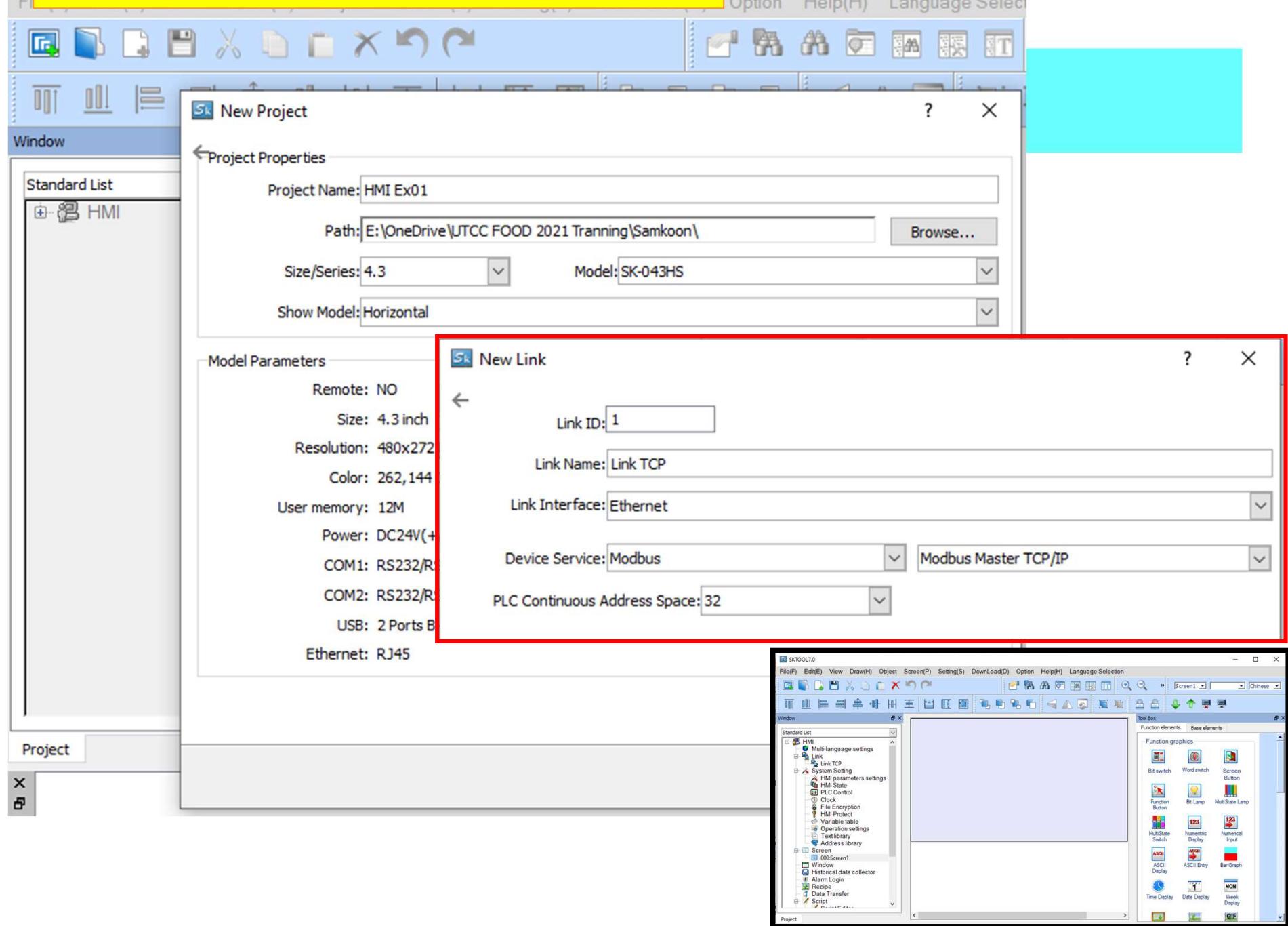
High power LED to compensate for dirt and misalignment.



ตัวอย่างการใช้ HMI (Touch Screen) ร่วมกับ PLC



YouTube Modbus – 05 HMI (Samkoon)





Bit switch

Sk Bit Button

element type: Bit Switch

ID: BB0000

View:

Prompt: Function: ON / OFF status monitoring

General tab (selected):

- State:** 1 (gray) | 0 (blue)
- Border Color:** [Color swatch]
- FG Color:** [Solid black bar]
- BG Color:** [Solid light gray bar]
- Pattern:** Solid

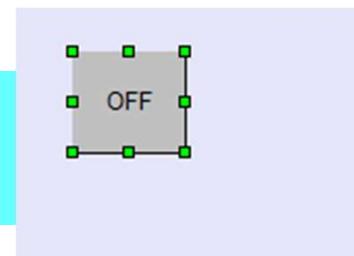
Function tab:

- Function:** Invert
- Mode:** Press execute

Script tab:

- Write Address:** 0x0
- Monitor** (checkbox checked)
- Monitor Address:** 0x0
- Monitor Address Identical to Write Address** (checkbox checked)
- Use Script** (checkbox)

Ok button



Sk Address Entry

Standard tab:

Link TCP dropdown menu:

- 0x
- 0x (highlighted)
- 1x
- 3x_Bit
- 4x_Bit
- 3x
- 4x
- 3x_D
- 4x_D

Buttons:

- CLR
- BS
- ESC
- ENT

Address input field: 0x0

Ok button

Function

Function: Invert

Invert (highlighted)

Mode: Inching

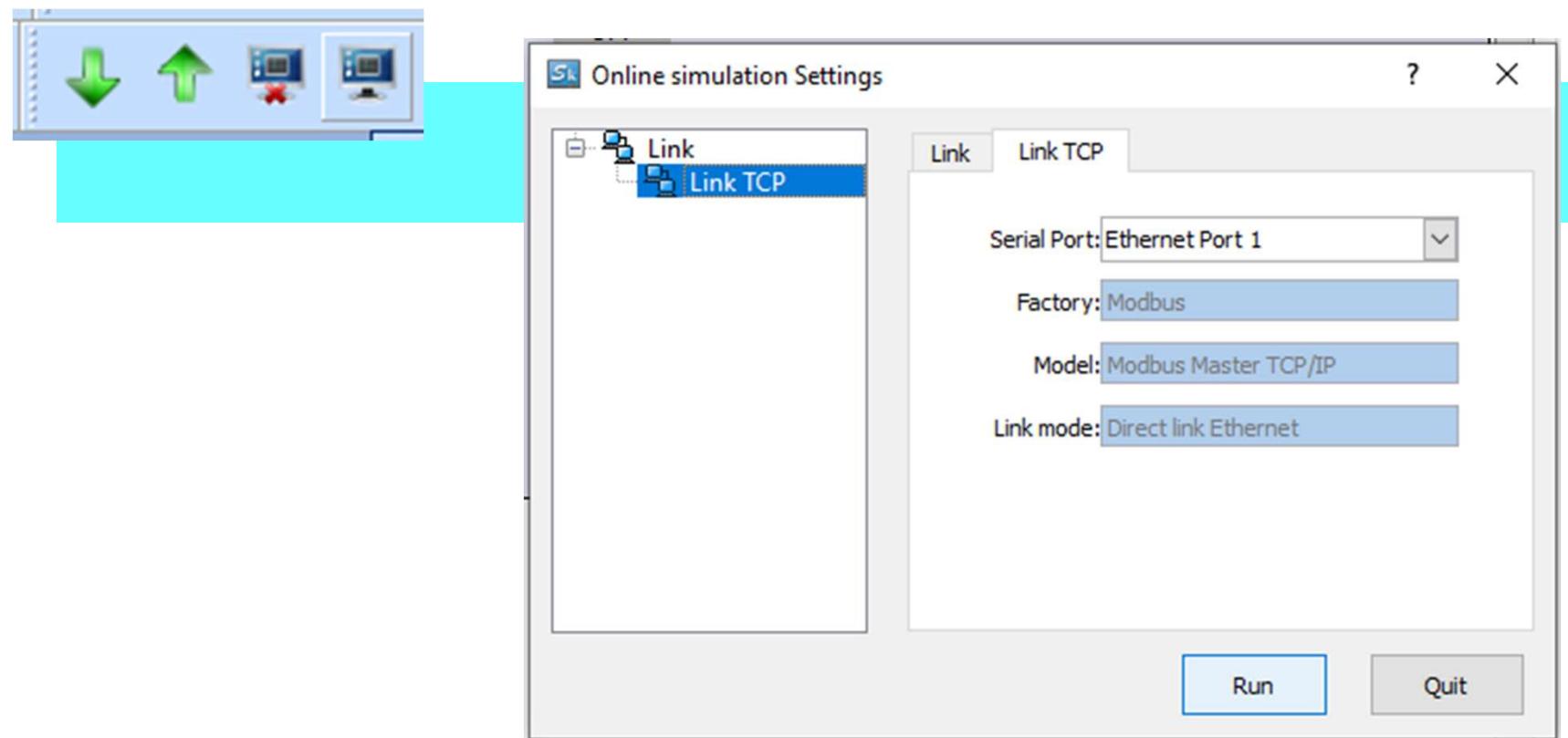
Set OFF

Set ON

Standard

Link TCP dropdown menu:

- Link TCP (highlighted)
- Internal storage area



The screenshot shows the SIMATIC Manager interface. At the top, there is a header with 'DRIVER' (selected), 'Modbus TCP/IP Server' (dropdown menu), a green checkmark icon, and buttons for 'STOP', 'CONFIGURATION', and 'CLEAR'.

The main area is divided into 'SENSORS' on the left and 'ACTUATORS' on the right. In the center is a detailed view of a connection:

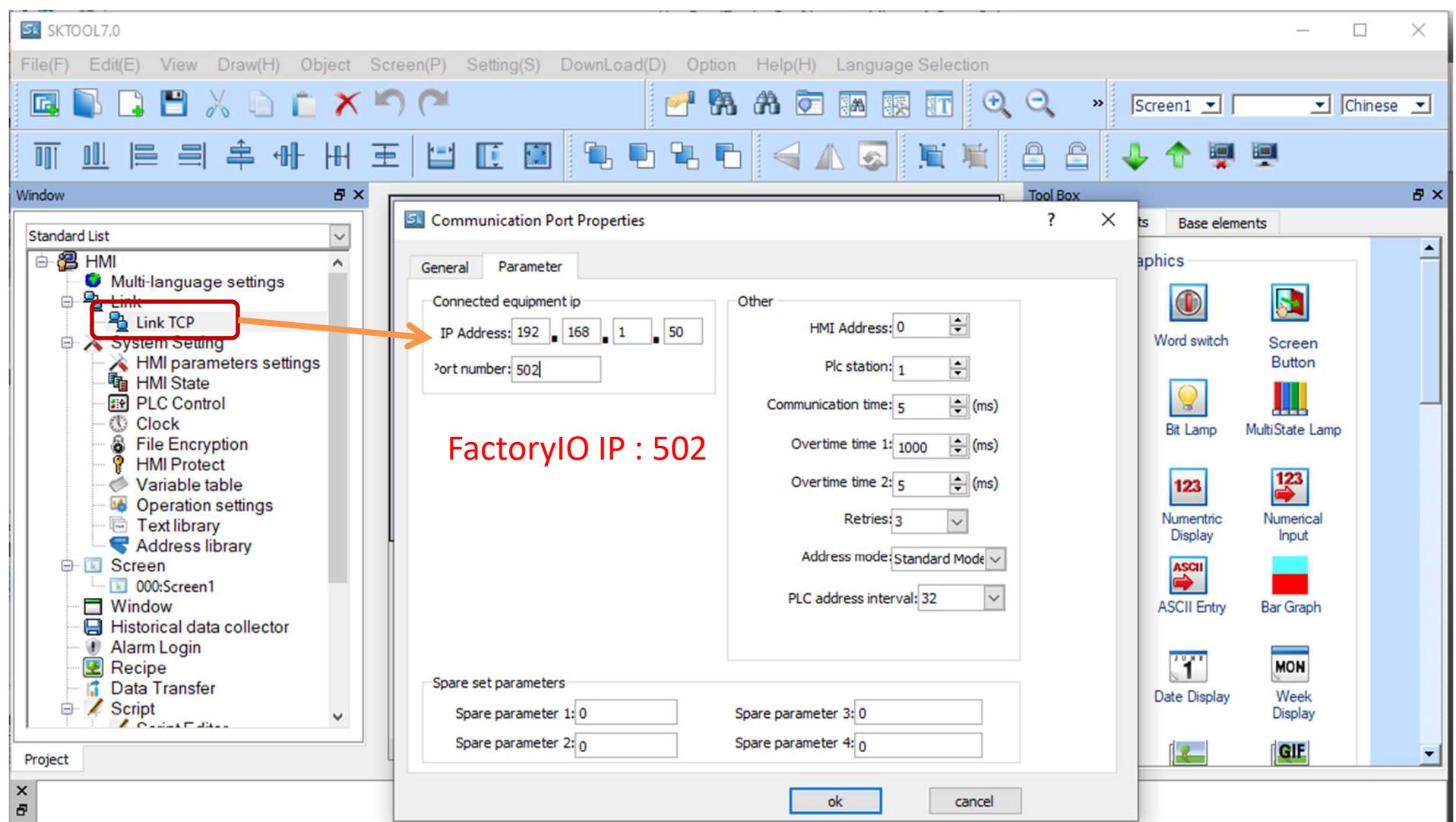
- SENSORS:** A box labeled '(192.168.1.50:502)' and 'Slave ID:1'. It contains two entries: 'Sensor' (green bar) and 'FACTORY I/O (Running)' (blue bar).
- ACTUATORS:** A box labeled 'Coil 0' (green bar). It contains one entry: 'Conveyor' (green bar).

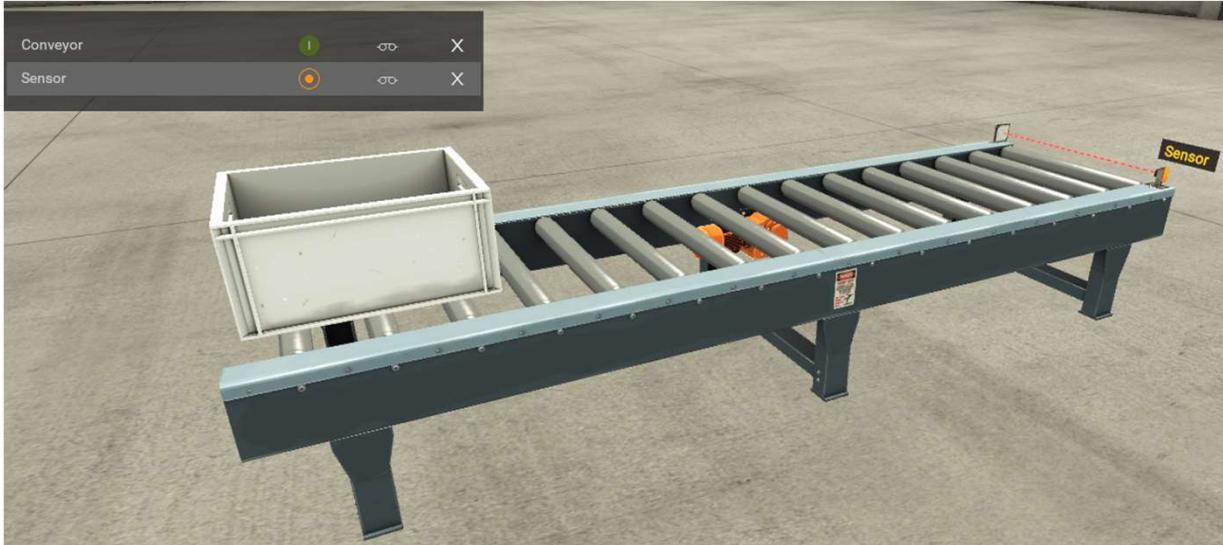
On the far left, there is a legend for 'FACTORY I/O' status:

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

On the far right, there is a legend for 'Conveyor' components:

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





DRIVER Modbus TCP/IP Server ▾ START CONFIGURATION CLEAR

SENSORS

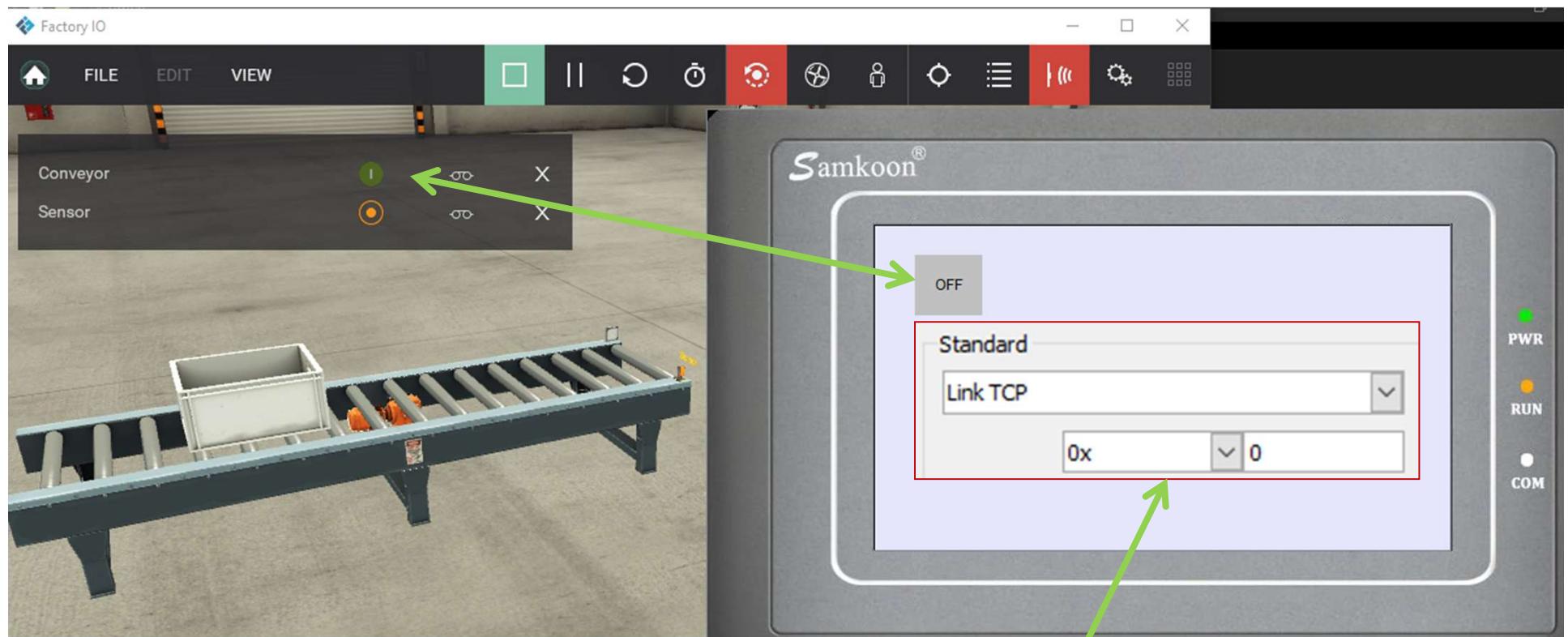
- FACTORY I/O (Paused) ■
- FACTORY I/O (Reset) ■
- FACTORY I/O (Running) ■
- FACTORY I/O (Time Scale) ■
- Sensor ■
- FACTORY I/O (Running) ■
- Sensor ■

ACTUATORS

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

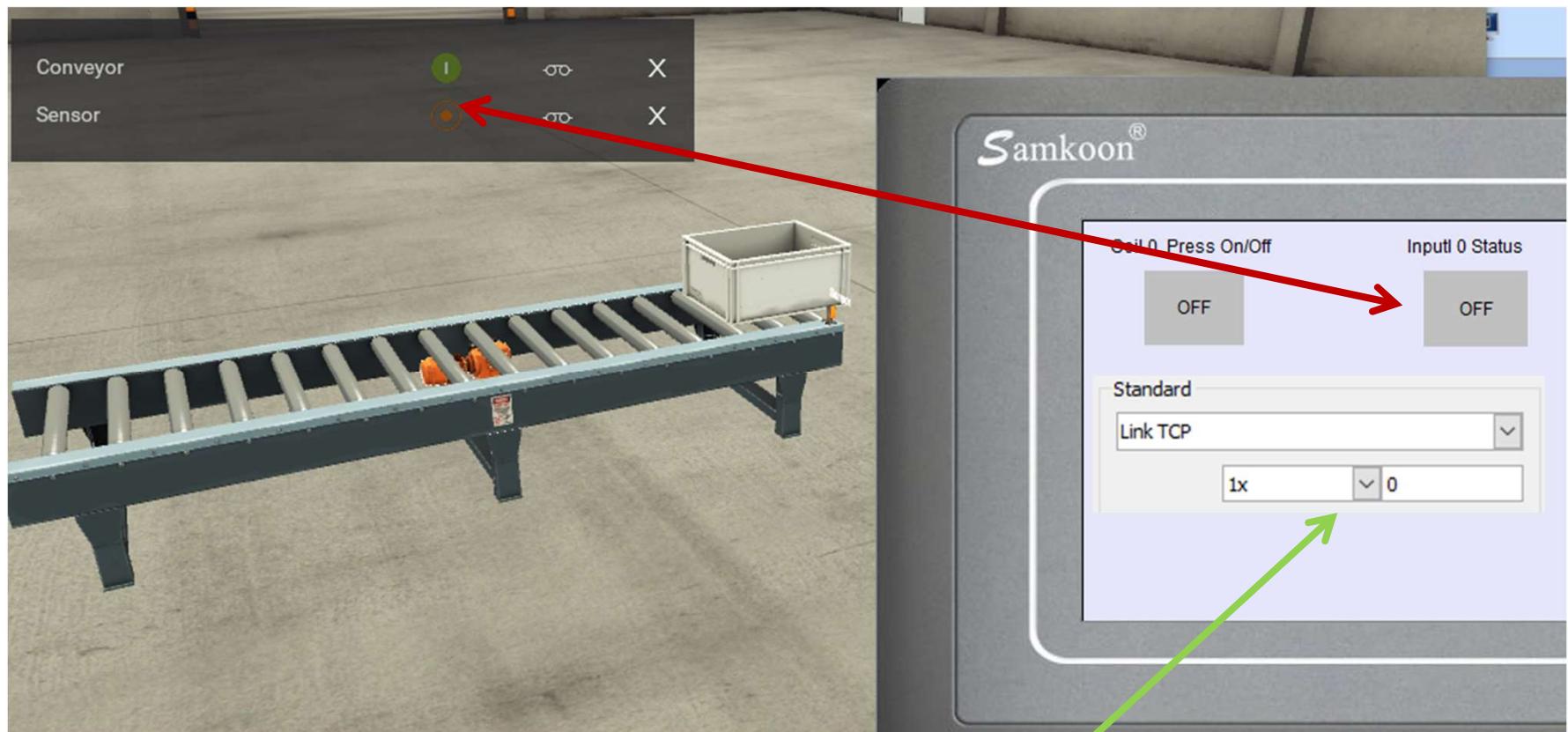
(192.168.1.50:502)
Slave ID:1

Input 0	Coil 0	Conveyor
Input 1		



The screenshot shows the Factory IO software interface with the following details:

- DRIVER:** Modbus TCP/IP Server
- SENSORS:**
 - FACTORY I/O (Paused)
 - FACTORY I/O (Reset)
 - FACTORY I/O (Running)
 - FACTORY I/O (Time Scale)
 - Sensor (FACTORY I/O (Running))
 - Sensor (FACTORY I/O (Run))
- ACTUATORS:**
 - Conveyor (Coil 0)
 - FACTORY I/O (Camera Position)
 - FACTORY I/O (Pause)
 - FACTORY I/O (Reset)
 - FACTORY I/O (Run)
- Central Panel:** A configuration box for a Modbus connection with the IP address '(192.168.1.50:502)', Slave ID:1, Input 0, Input 1, and Coil 0.
- Status:** STOP, CONFIGURATION, CLEAR



DRIVER Modbus TCP/IP Server ✓ STOP CONFIGURATION CLEAR

SENSORS

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

ACTUATORS

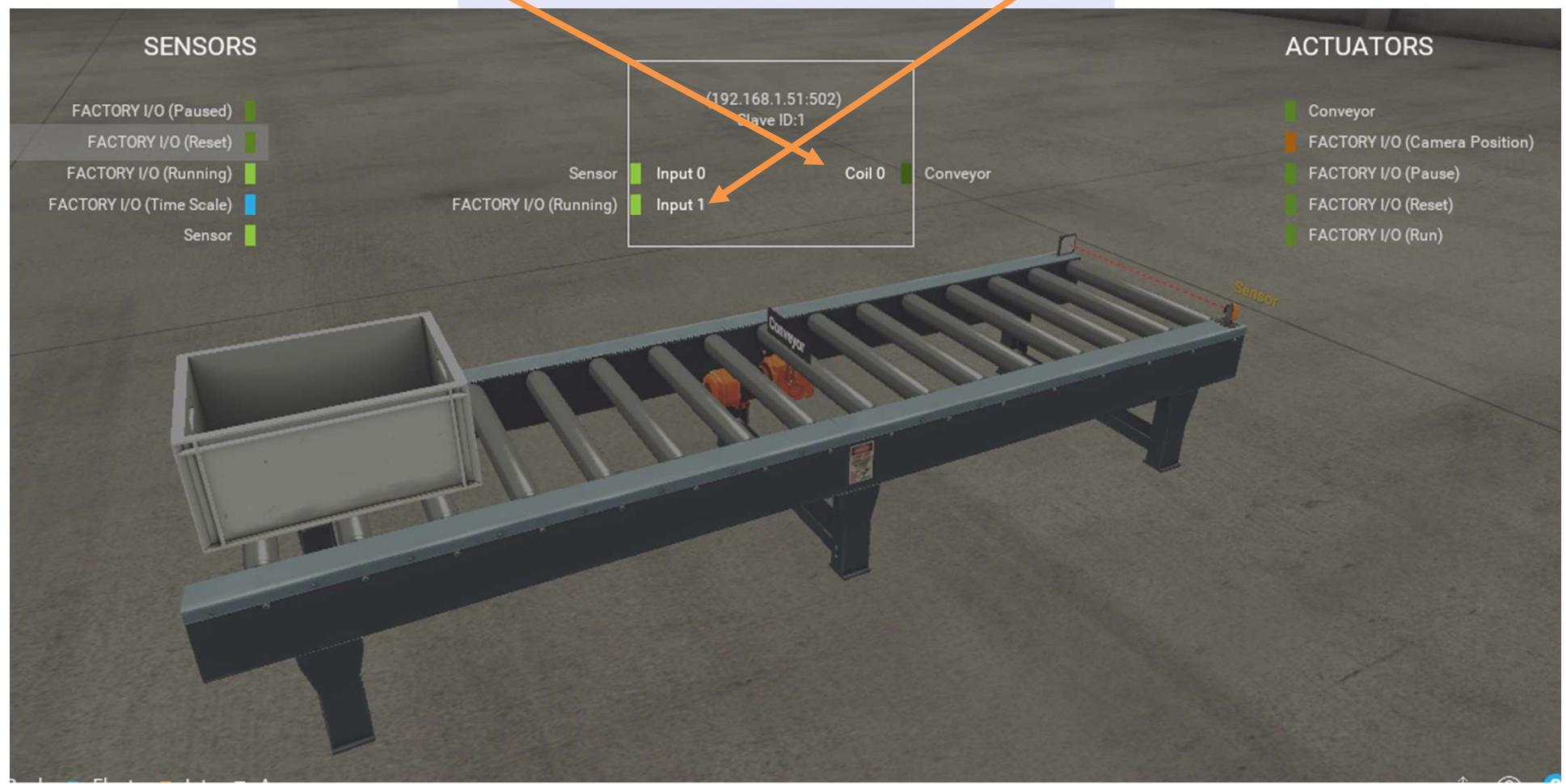
- Conveyor
- FACTORY I/O (Camera Position)
- FACTORY I/O (Pause)
- FACTORY I/O (Reset)
- FACTORY I/O (Run)

Configuration Screen (Top Right):

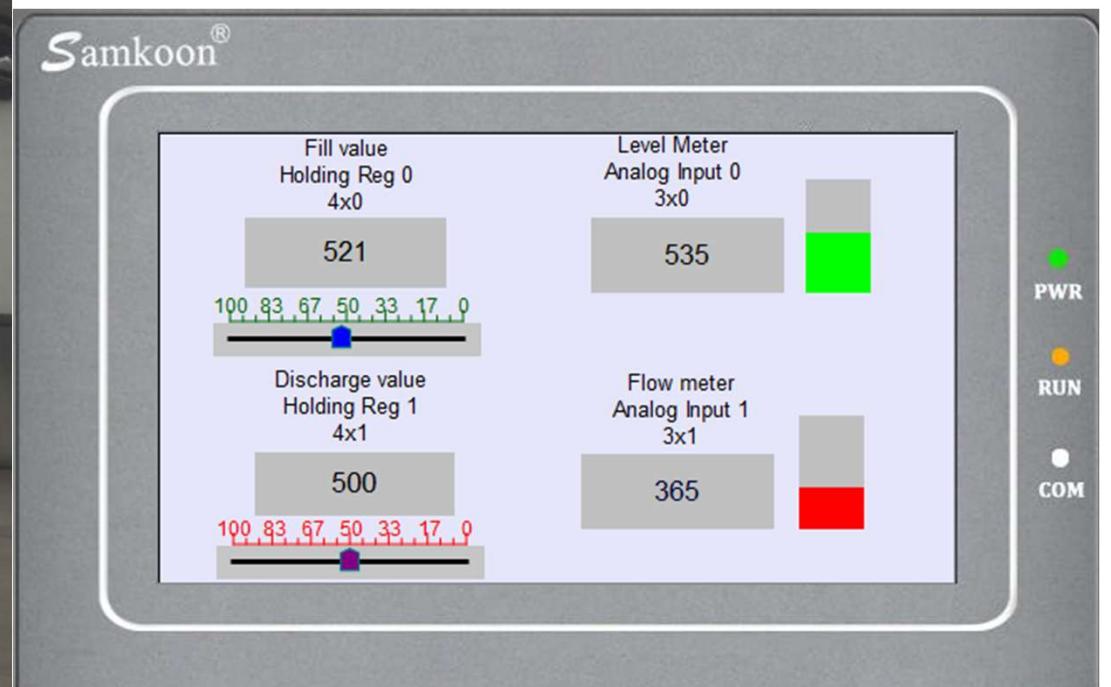
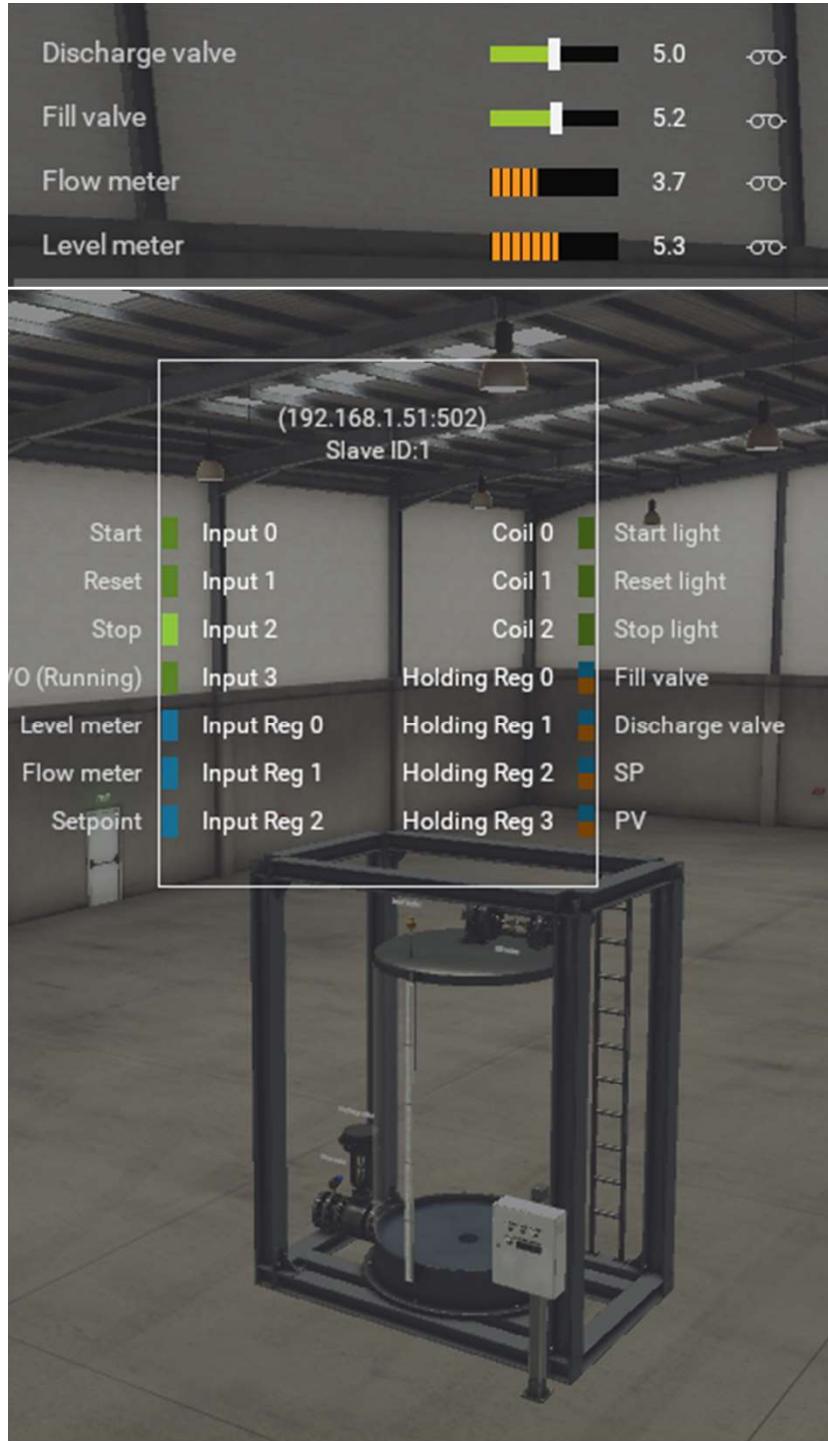
- Coil 0: Press On/Off: OFF
- Input 0 Status: OFF
- Standard: Link TCP
- Link TCP: 1x 0

Legend:

- Green square: Conveyor
- Orange square: FACTORY I/O (Camera Position)
- Green square: FACTORY I/O (Pause)
- Green square: FACTORY I/O (Reset)
- Green square: FACTORY I/O (Run)



YouTube Modbus – 06 HMI Advance



UTCC IIOT (Industrial Internet of Thing) Starter Kid Training

แสดงสถานะการ
ทำงานปั๊จจบัน



อุณหภูมิ
ความชื้น
บรรยายกาศ



อุณหภูมิควบคุมเครื่องจักร
อุณหภูมิเครื่องจักรทำงาน



สั่งงานเครื่องจักร
ทำงาน (On/Off)



Modbus RTU



Modbus TCP

Wire to Wireless

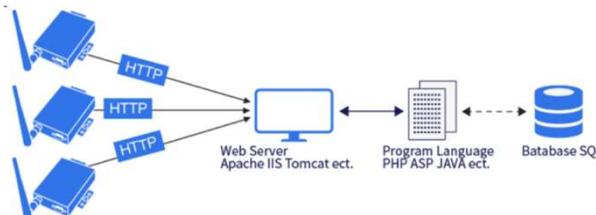
จัดเก็บข้อมูลเข้าระบบฐานข้อมูล
เพื่อนำมาวิเคราะห์ และประมวลผล

IIOT Server (SCADA)



(Mysql , NodeRed
and Docker)

Book1 - Microsoft Excel							
A	B	C	D	E	F	G	H
1 Weight Values							
2	12.562						
3	12.455						
4	12.367						
5	11.905						
6	11.903						
7	10.657						
8	6.433						
9							
10							
11							
12							
13							
14							
15							



Open Source / Hardware Technologies

