

The Project of Capacity Building for Industry Demand-based TVET in Uganda

In-service Training Program (Manual) for Instructors

Electrics

January, 2025



Contents

Day	Activity		Details
Day 1	Job Sheet 1	Preparing Download	➤ Meeting regarding as training plan ➤ Prepare for training. - Set up the program - Sequence design software - PLC software (XG5000) - HMI software (XP builder) ➤ Programming and simulation
Day 2	Job Sheet 2	3phase IM FR control	➤ Control Panel working ➤ 3Phase IM direction FR control by Program * Sequence installation * PLC installation.
Day 3	Job Sheet 3	Auto Level Control	➤ Control Panel working ➤ Auto Level Control system * Sequence installation * PLC installation.
Day 4	Job Sheet 4	HMI design	➤ Human & Machine Interface 1) XP builder setup and design 2) Design by HMI
Day 5	Job Sheet 5	PLC & HMI	➤ PLC & HMI simulation 1) PLC Programming 2) Design by XP builder 3) Installation
Day 6	Job Sheet 6	PLC & VFD	➤ PLC & VFD setting 1) 7step speed control setting
Day 7	Job Sheet 7	PLC & VFD	➤ Multistep speed control by VFD 1) Reverse control
Day 8	Job Sheet 8	PLC & VFD	➤ Multistep control 1) 7step 2) Forward and Reverse
Day 9	Job Sheet 9	Arduino setting& LED control	➤ LED control by Arduino 1) LED blinking & 3LED control
Day 10	Job Sheet 10	RGB control	▶ RGB control by Arduino 1) PWM.

Job Sheet 1

Day	1	Task No.	EE-PLC-01	Time	8h
Title	Preparing Download				
Material	Name	Specification	Q'ty		
	Sequence software	ESS	1		
	PLC software	XG5000	1		
	HMI software	XP builder	1		
1. Purpose ☐ Software download 2. Download 1) Sequence software - Download and design 2) PLC software - Download and design 3) HMI software - Download 4) Arduino software - Download and design 3. System setting ☐ PLC, Control panel, HMI, VFD and Materials					
☐ Information Window 10, 64bit over					
☐ Requirements Prepare the Equipment, tools and Materials including Software					

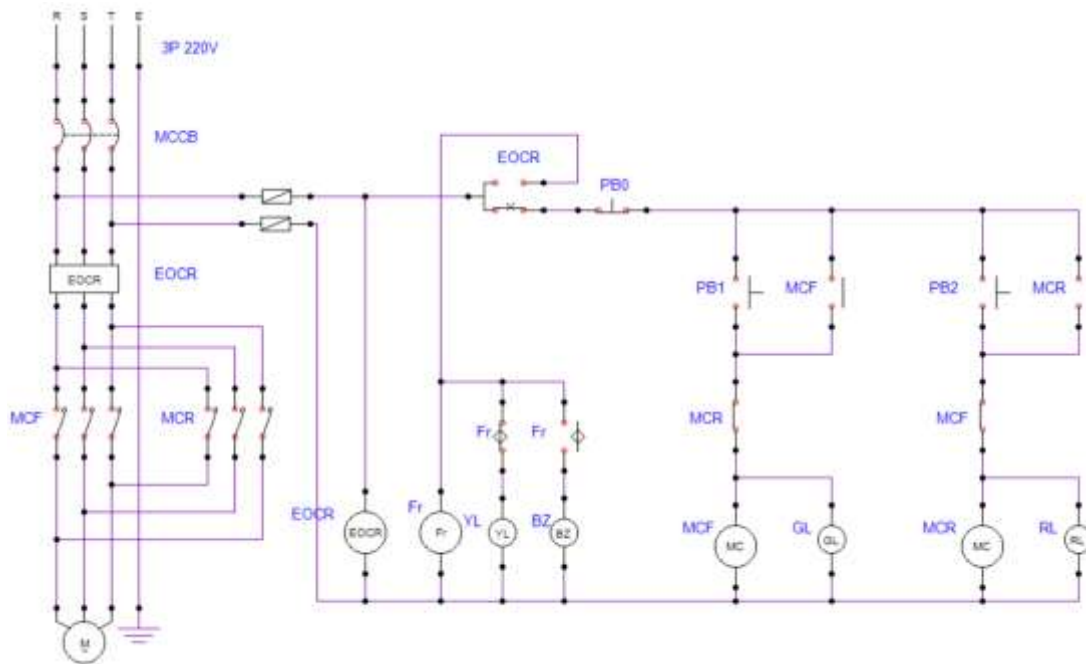
Job Sheet 2

Day	2	Task No.	EE-PLC-02	Time	8h
Title	3phase IM FR control				
Material	Name	Spec	Q'ty		
	PLC	XBC-DR30SU	1		
	Sequence board		1		
	MC, EOCR	Industrial type	2		

1. Purpose

- Design Sequence circuit of 3phase IM FR control with EOCR
- Install PLC and Sequence

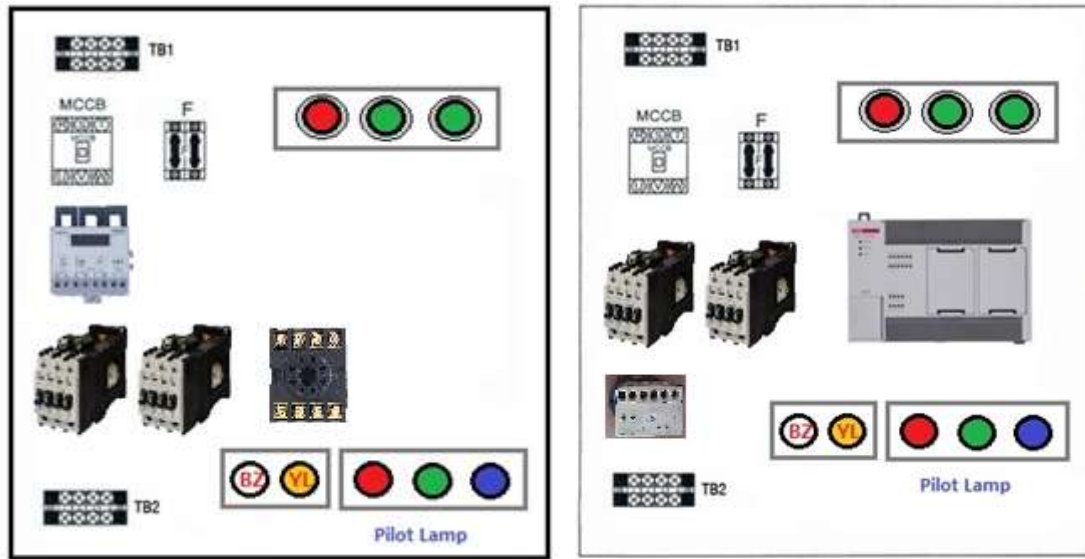
2. Diagram



3. Practice

- 1) 3phase IM FR control with EOCR and Flicker by Sequence
 - Step 1. Programming by sequence software
 - Step 2. Simulation
- 2) 3phase IM FR control with EOCR and Flicker by PLC
 - Step 1. PLC programming and simulation
 - Step 2. Program writes to PLC
 - Step 3. Installation
 - Step 4. Operation

4. Layout



□ Information

□ Requirements

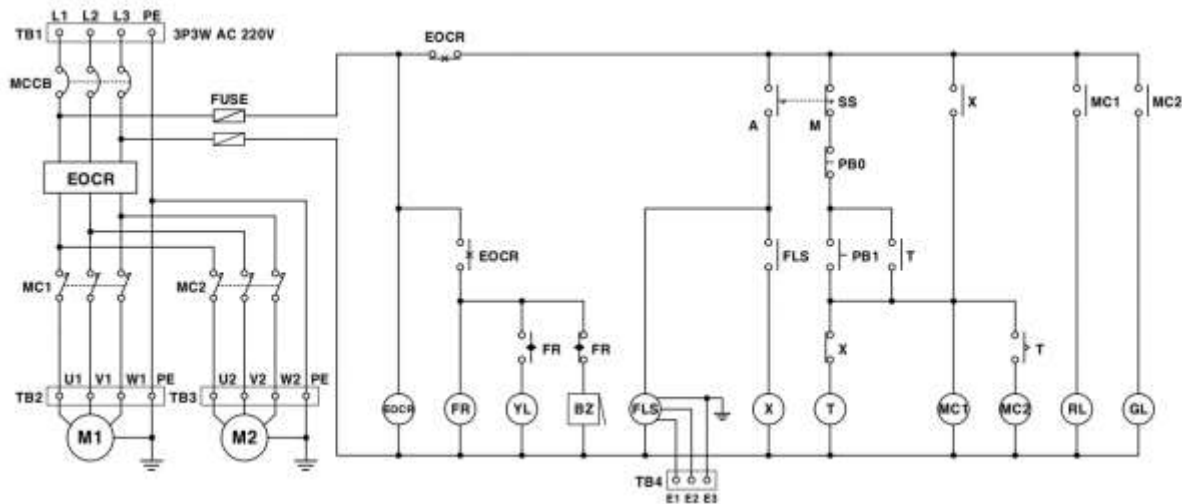
Job Sheet 3

Day	3	Task No.	EE-PLC-03	Time	8h
Title	Auto Level Control				
Material	Name		Spec	Q'ty	
	PLC		XBC-DR30SU	1	
	Floatless relay		8pin	1	
	Flicker relay		8pin	1	

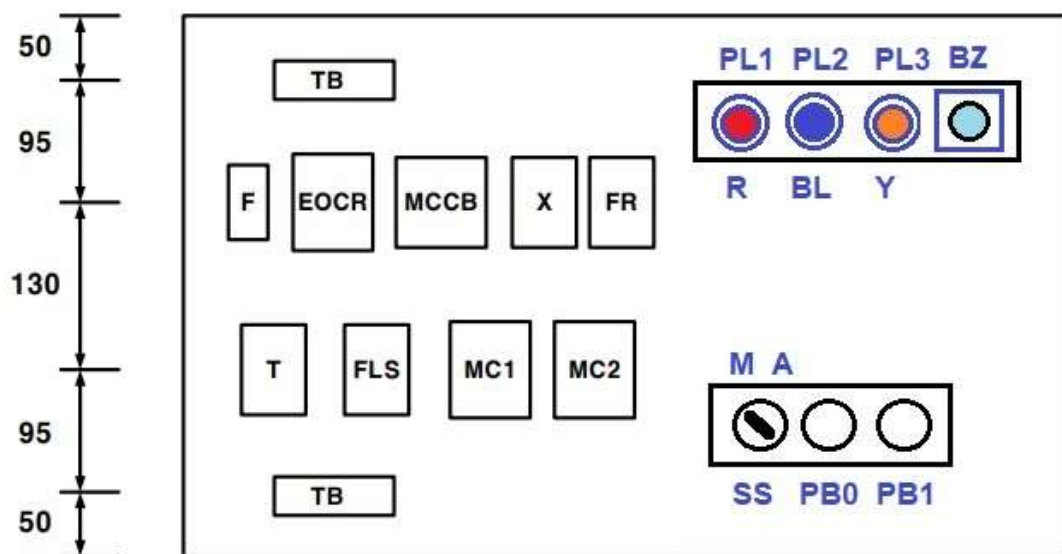
1. Purpose

- Wiring Sequence circuit of Auto level control system
- Wiring PLC control of Auto level control system

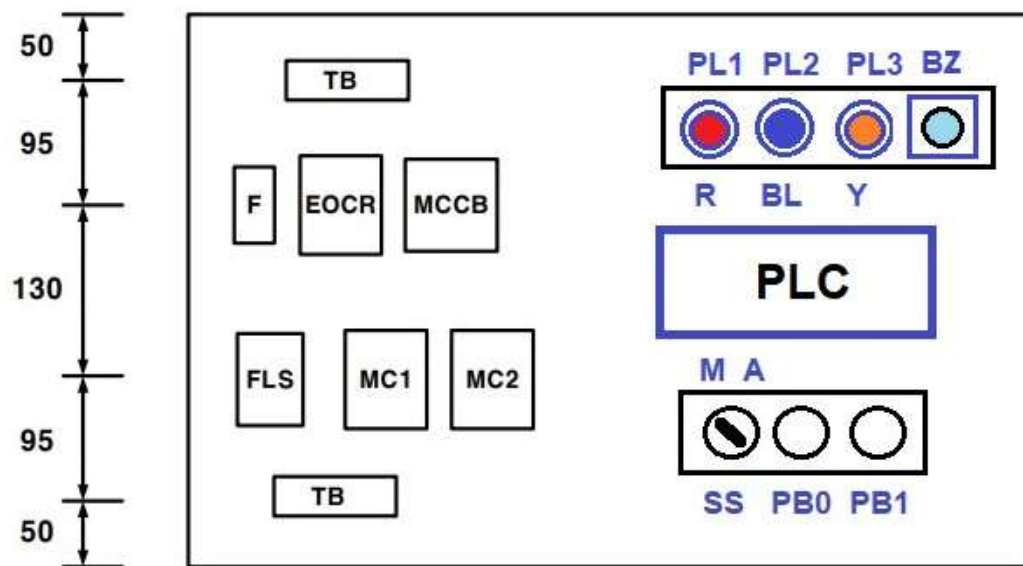
2. Circuit diagram



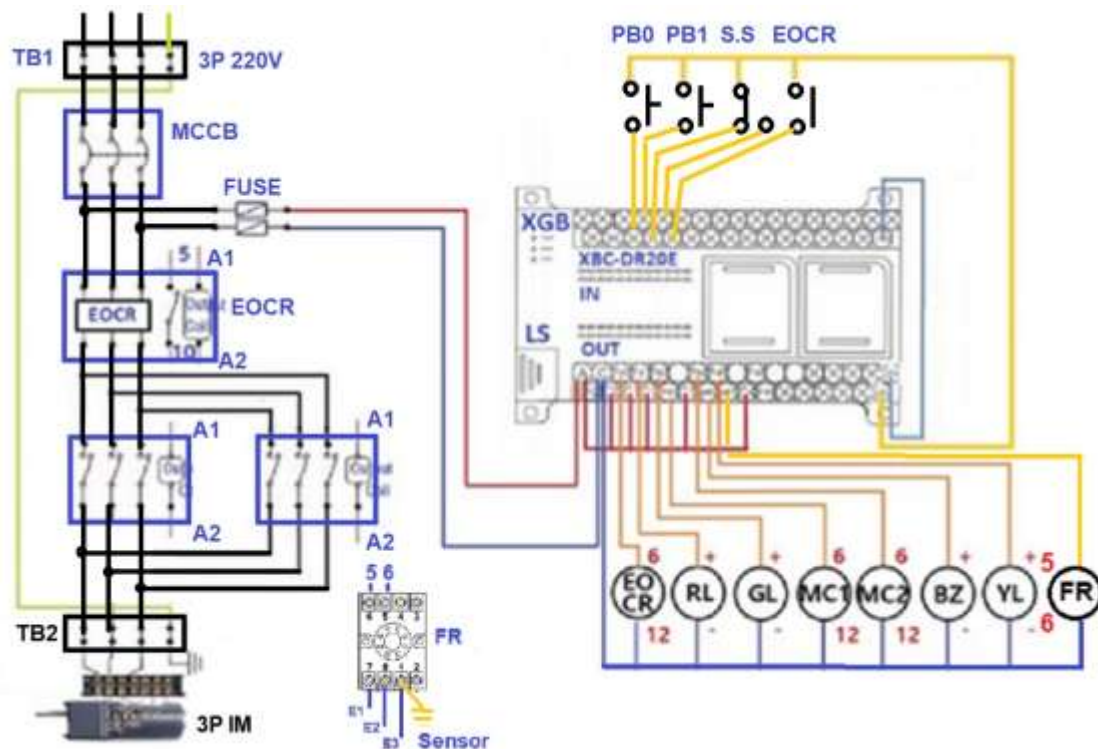
3. Layout by sequence



4. Layout by PLC



5. Connection diagram by PLC



□ Information

□ Requirements

Job Sheet 4

Day	4	Task No.	EE-HMI-04	Time	8h
Title	HMI design				
Material	Name	Spec	Q'ty		
	PLC	XBC-DR30SU	1		
	HMI	XP-Builder	1		
	computer		1		

1. Purpose

- Practice to design the XP-Builder software

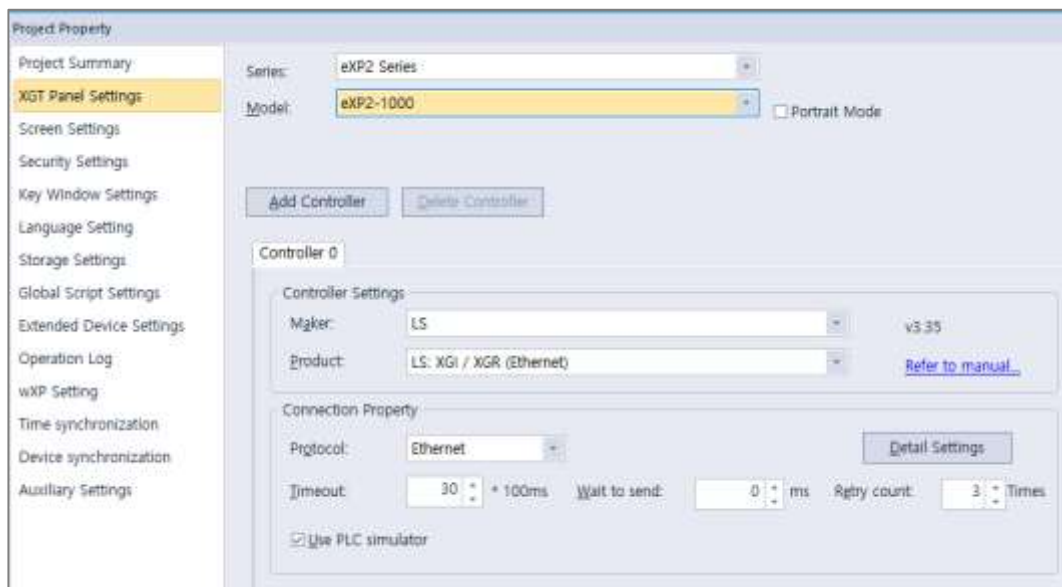
2. Design

- 1) XP builder Program [RUN]
- 2) Create Project



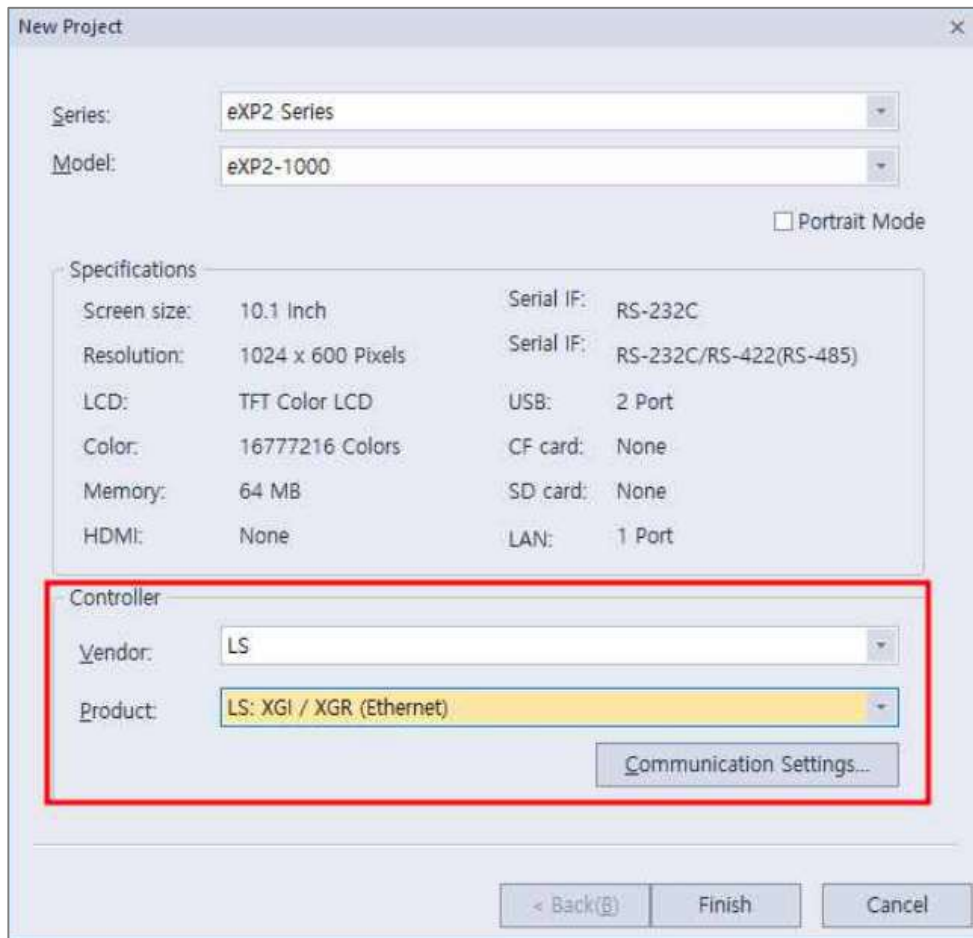
<Create Project>

- 3) Select the HMI series and model name: eXP2series, eXP2-1000



4) Controller: to be connected

- Product: LS, XGI/XGR(Ethernet)

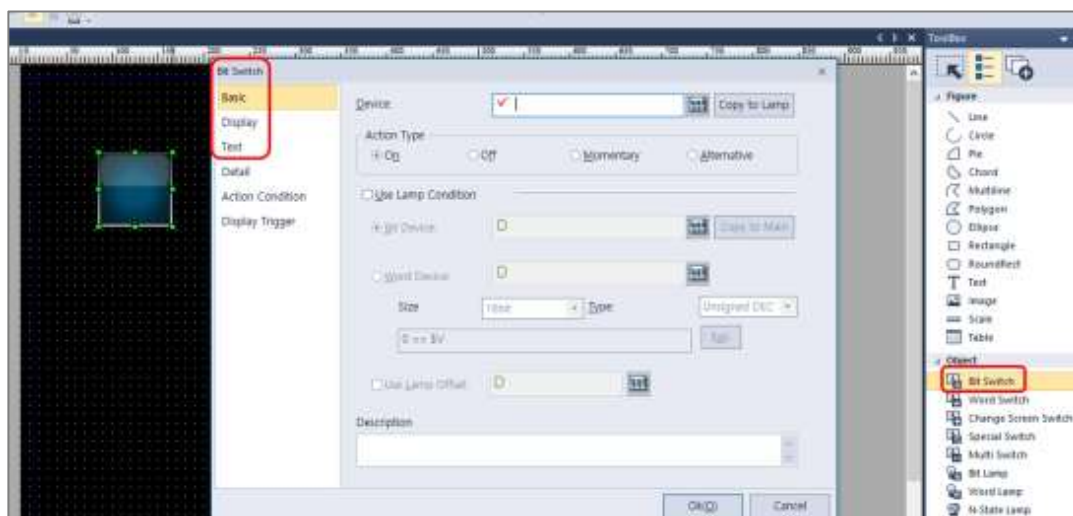


- Choose [Finish] to create your project

3. Design the XP builder

1) Design the [Bit switch] on [Object] right

- Click the [Bit switch] on [Object]



- Basic: M100, M200, M300 for INPUT as Bit Switch

Bit Switch

Basic

Display

Text

Detail

Action Condition

Display Trigger

Device: D %MX100

Action Type

☐ On ☐ Off ☒ Momentary ☐ Alternative

☒ Use Lamp Condition

☒ Bit Device: D %MX100

☐ Word Device: D

Size: 16bit Type: Unsigned DEC

0 == \$V Exp.

☐ Use Lamp Offset: D

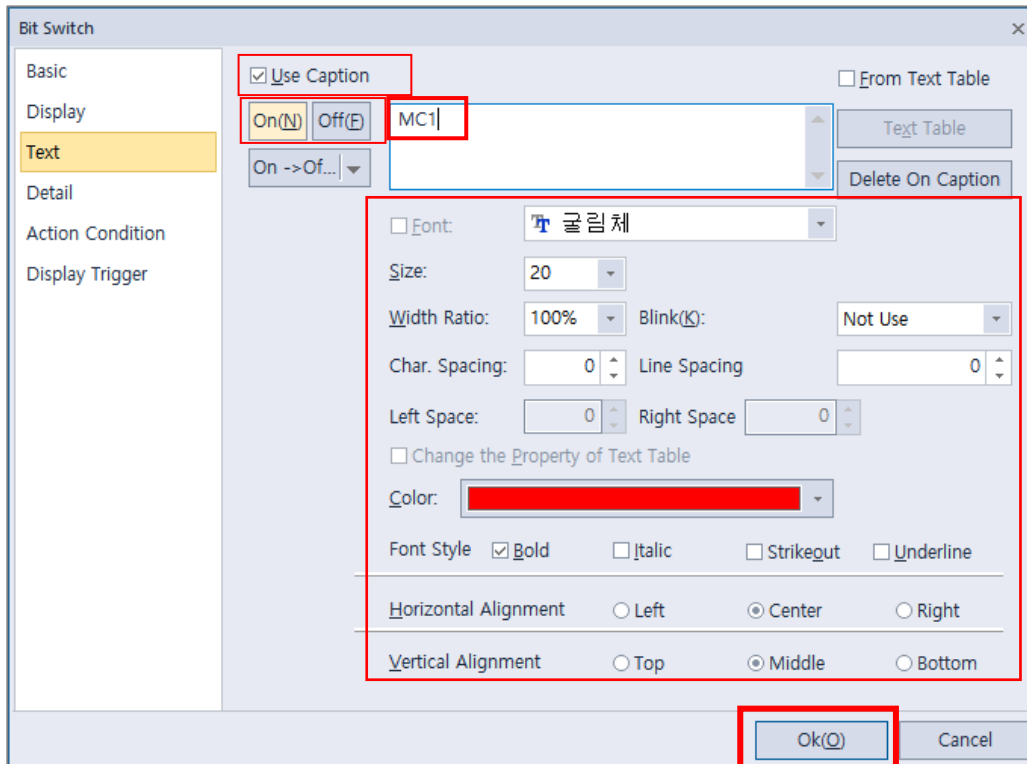
Description

Ok(O) Cancel

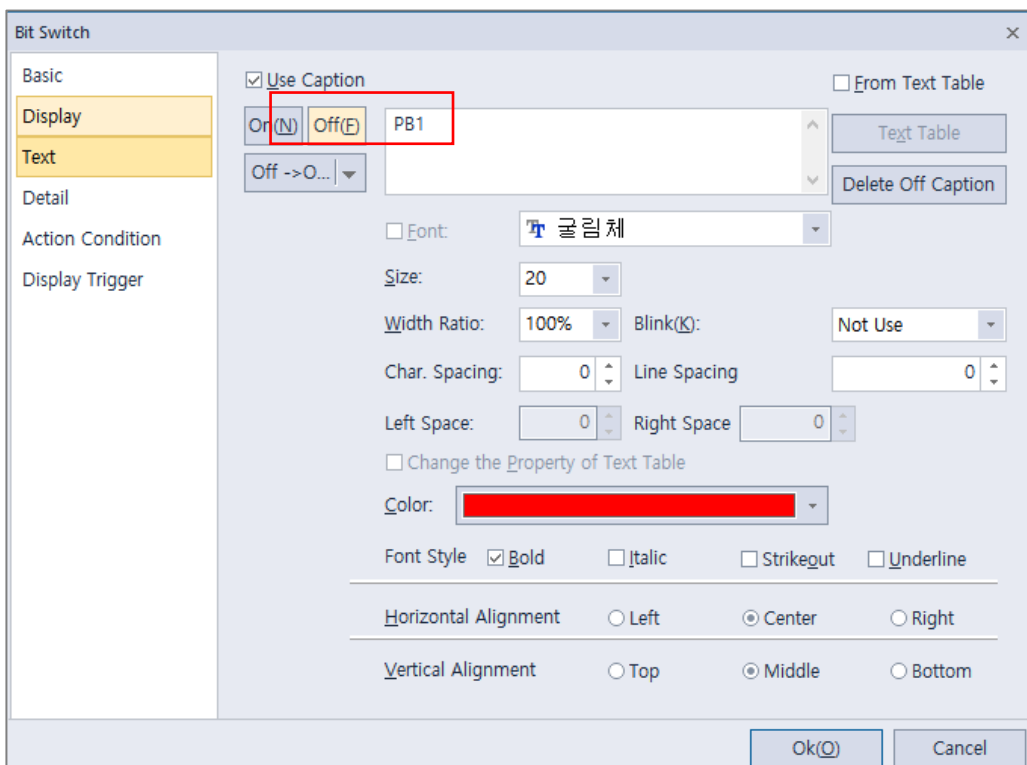
- **Display:** [ON], [OFF]

□ Text: Click the [Use Caption] - M100, M200, M300 each one

□ [ON]: ON, Size, color, Font size, --- [OK]

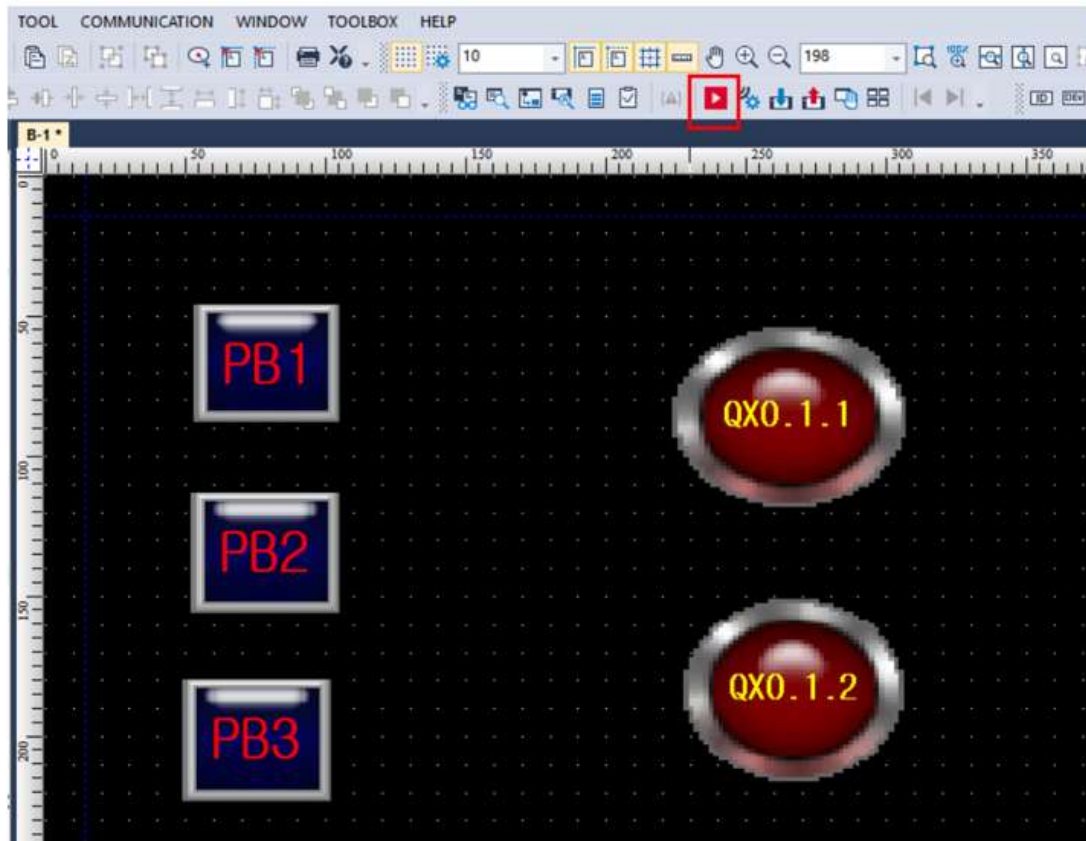


□ [OFF]- M100, M200, M300 display--- [OK]



<Text-OFF>

2) Click the [OK]



☐ Information

☐ Requirements

Job Sheet 5

Day	5	Task No.	EE-HMI-05	Time	8h
Title	HMI simulation				
Material	Name	Spec	Q'ty		
	PLC	XBC-DR30SU	1		
	HMI	XP-Builder	1		
	computer		1		

1. Purpose

- Practice to simulate the XP-Builder software and PLC program

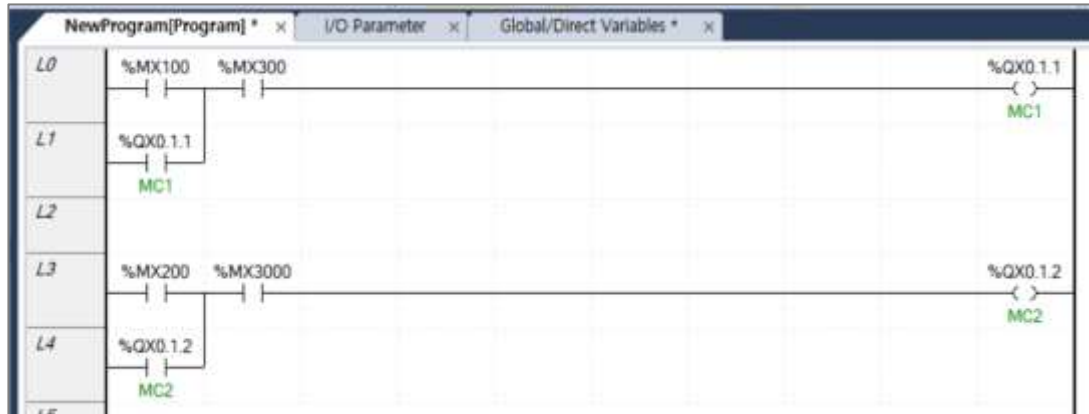
2. PLC-HMI programming and Simulation

1) PLC Programming

- PLC I/O address

Input address	Device	Output address	Device
M100	PB1_Forward	%QX0.1.1	MC1
M200	PB2_Reverse	%QX0.1.2	MC2
M300	PB3_Stop		

- Ladder diagram



2) Hidden the PLC diagram

3) Click the [XP builder] icon on screen to be [RUN]

4) Click [Create Project]

- New project
 - Series: eXP2 Series
 - Model: eXP2-1000
- Controller
 - Maker: LS
 - Product: LS: XGI/XGR(Ethernet) / [OK]

- Double click the [XGT Panel setting]
- Detailed Setting

IP address: 192. 168.1.2

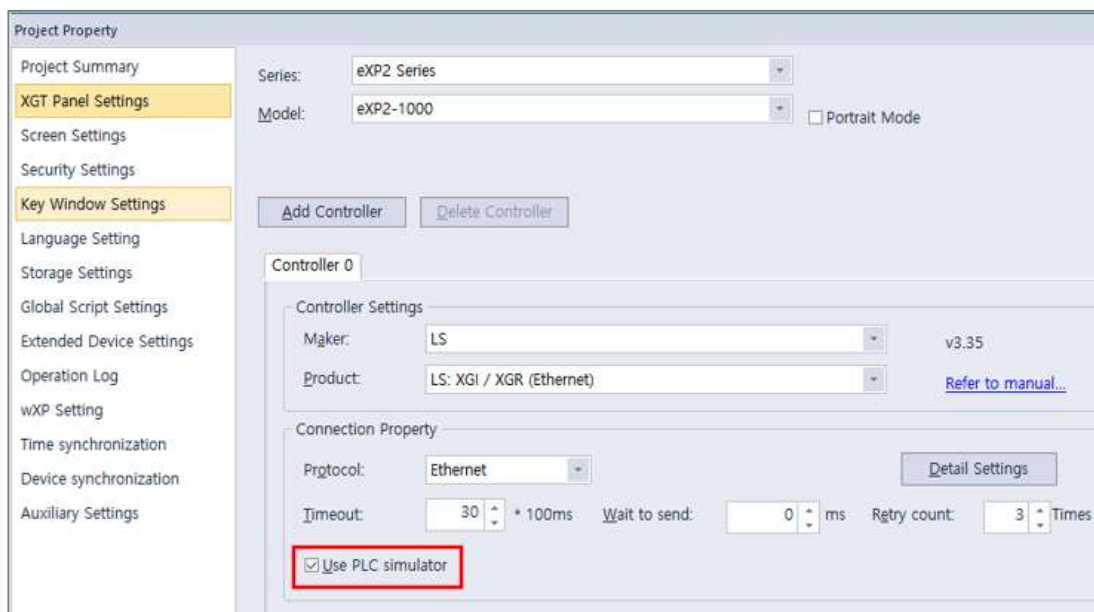
Destination port:2004

Click the [OK]

- [OK]

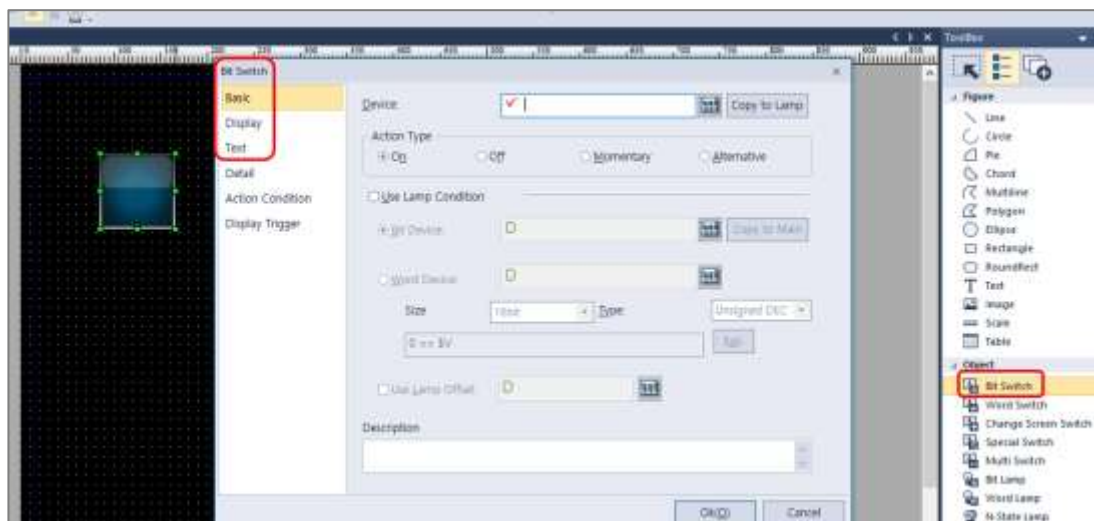
5) Simulation setting

- Double click the [XGT Panel setting]
- [Use PLC simulator] is checked [v]: for only simulation with PLC



6) Design the XP builder

- Bit switch [PB1], [PB2], [PB3], and Lamp [MC1], [MC2]
- Design the [Bit switch] on [Object] right
- Click the [Bit switch] on [Object]



< Object>

- Basic: M100, M200, M300 for INPUT as Bit Switch

Bit Switch

Basic

Display

Text

Detail

Action Condition

Display Trigger

Device: D %MX100

Action Type

☐ On ☐ Off ☒ Momentary ☐ Alternative

☒ Use Lamp Condition

☒ Bit Device: D %MX100

☐ Word Device: D

Size: 16bit Type: Unsigned DEC

0 == \$V Exp.

☐ Use Lamp Offset: D

Description

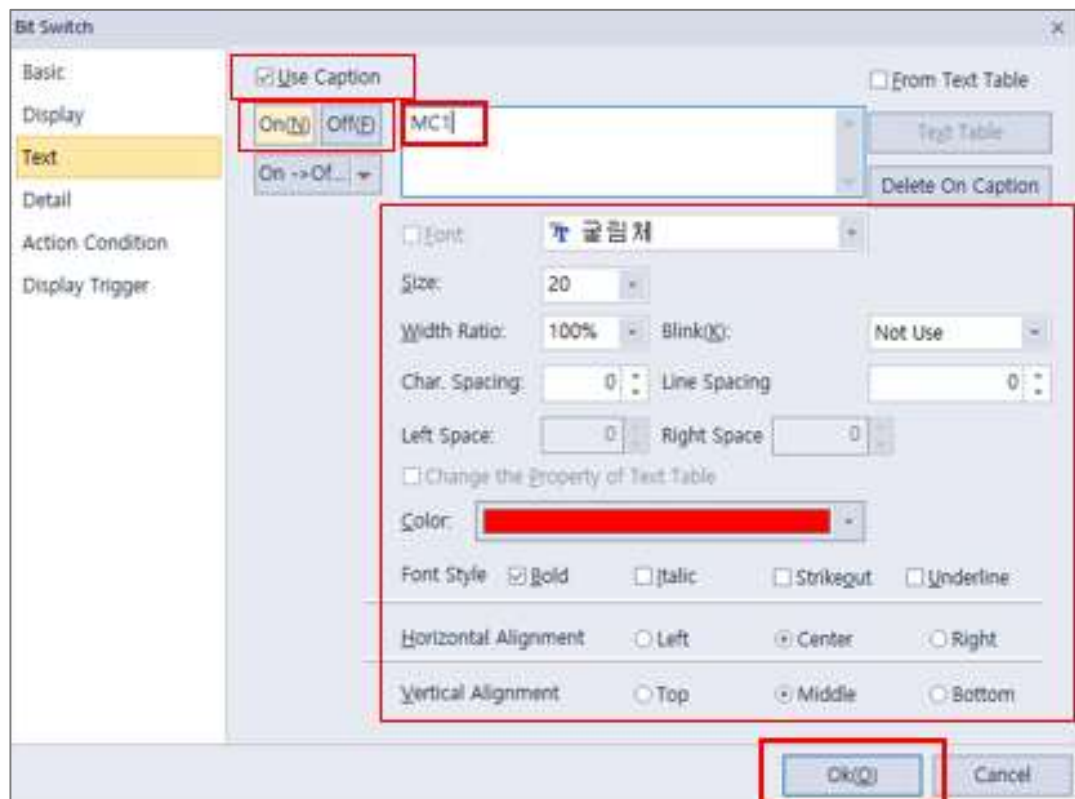
Ok(Q) Cancel

- Display: [ON], [OFF]

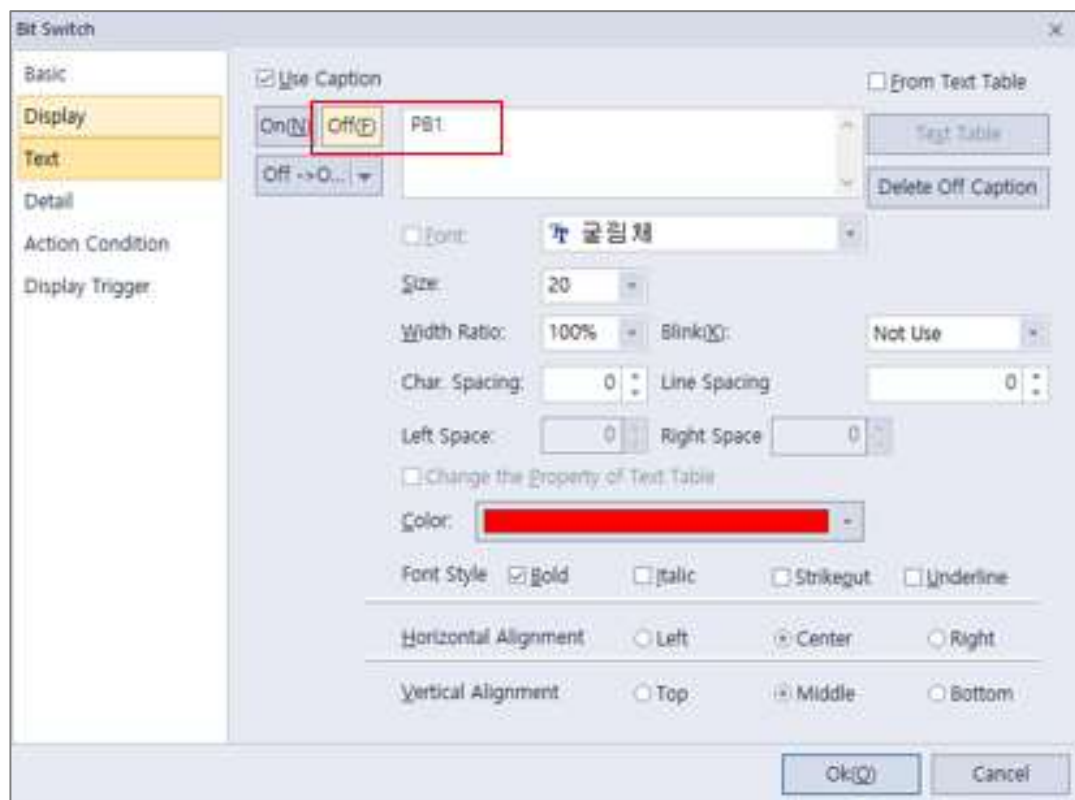
ON

OFF

- Text: Click the [**Use Caption**] - M100, M200, M300 each one
- [**ON**]: ON, Size, color, Font size, --- [**OK**]



- [**OFF**]- M100, M200, M300 display--- [**OK**]



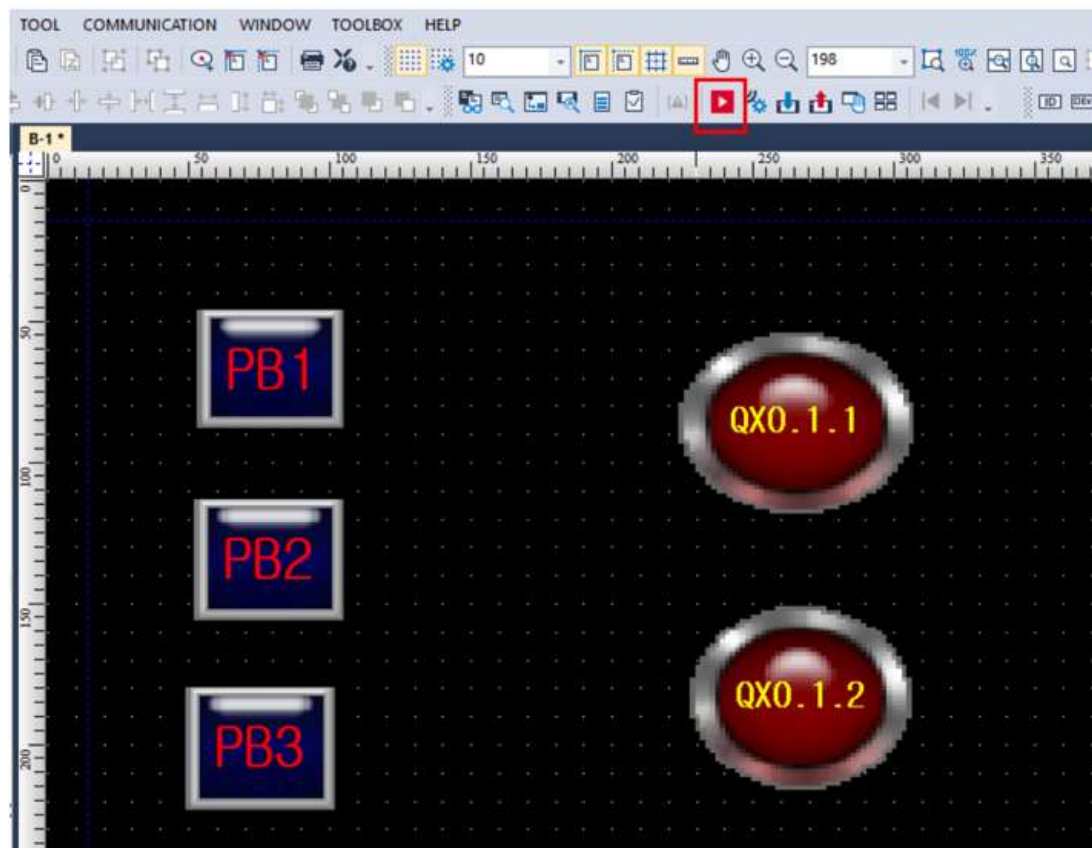
- Click the [**OK**]

7) PLC simulation

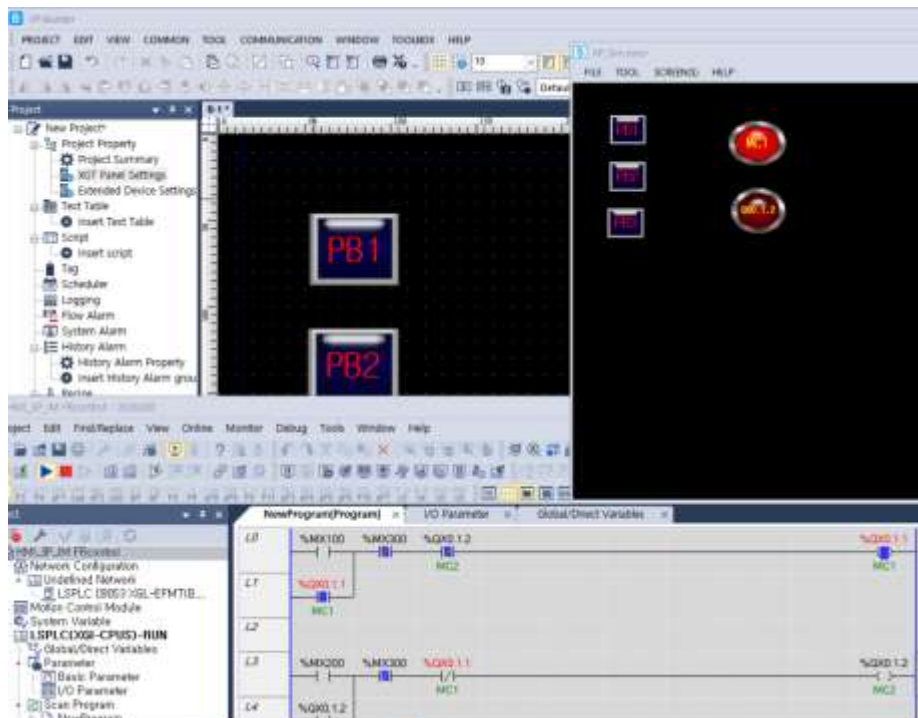


8) HMI simulation

- Click the Simulation

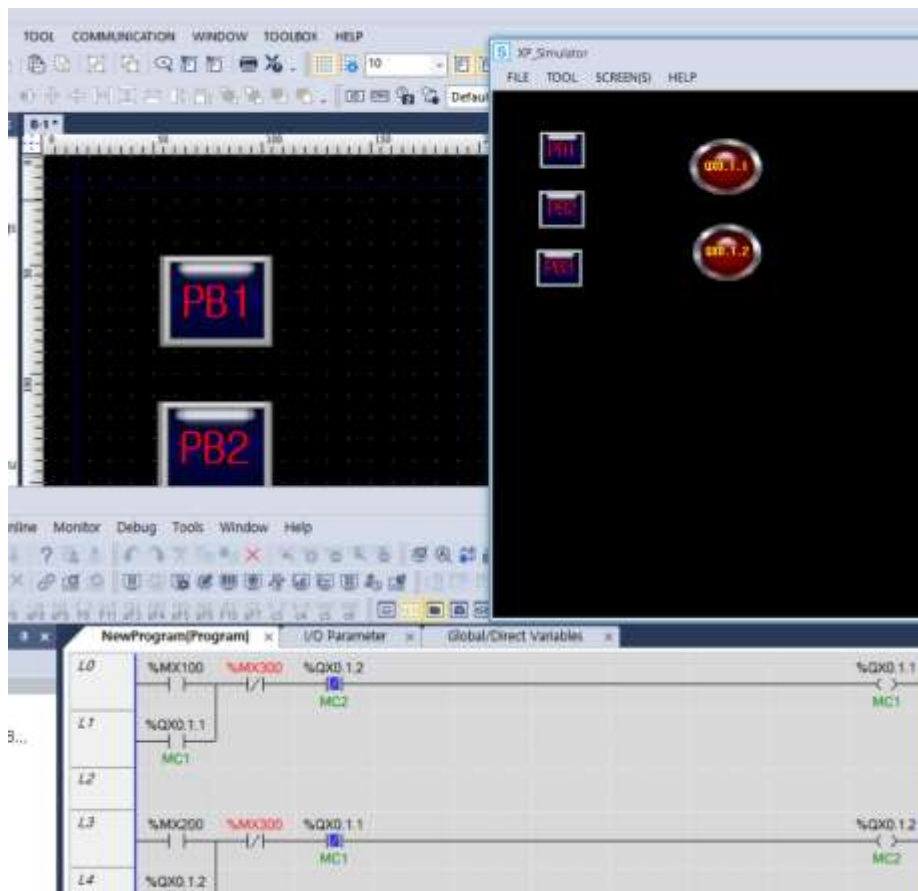


□ PB1-MC1



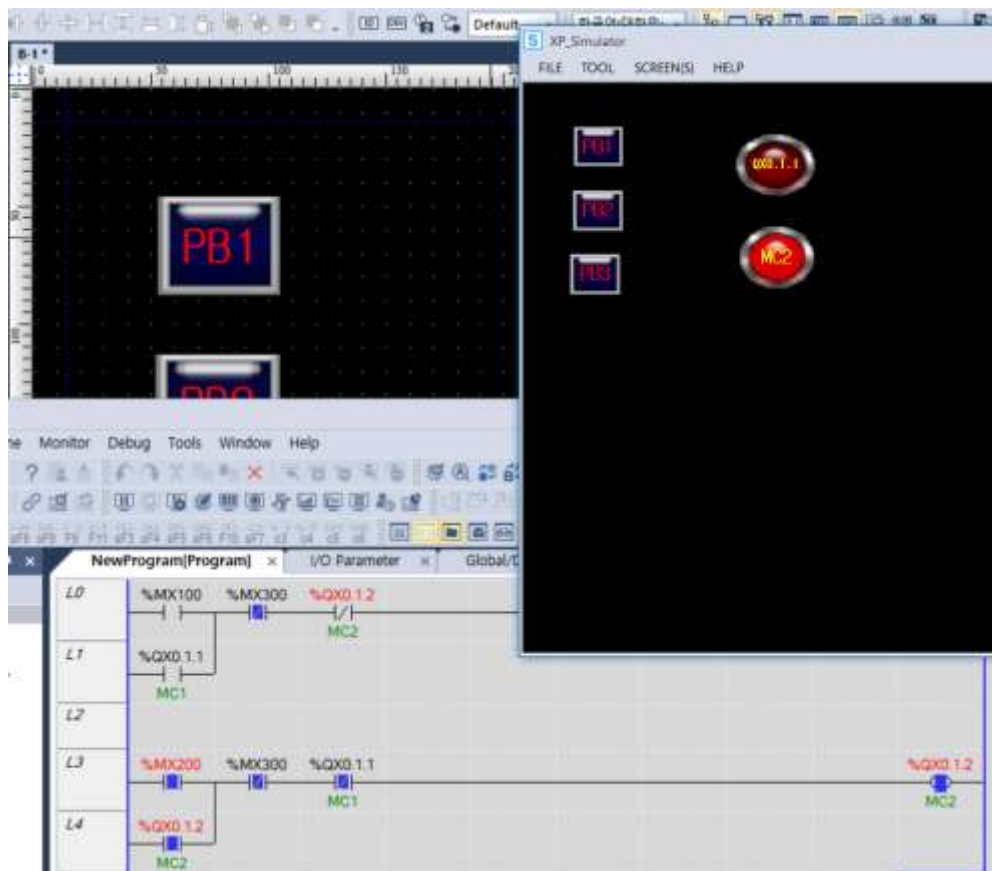
< PB1 simulation>

□ PB0-STOP



< PB0 simulation>

(4) PB2-MC2



□ Information

□ Requirements

Job Sheet 6

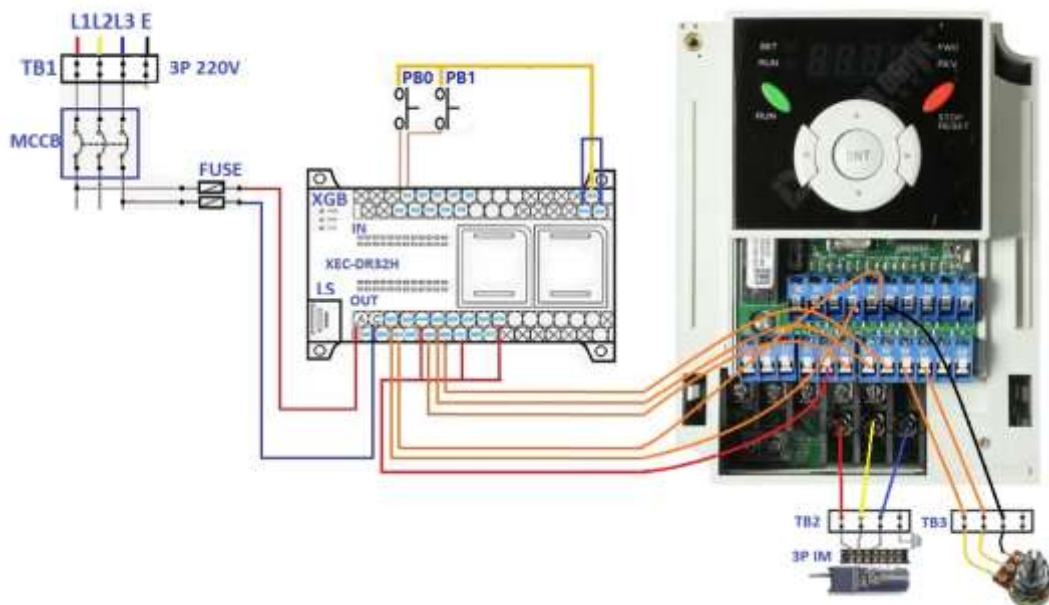
Day	6	Task No.	EE-VFD-06	Time	8h
Title	PLC&VFD setting				
Material	Name	Spec	Q'ty		
	PLC	XBC-DR30SU	1		
	VFD	IG5A	1		
	Computer		1		

1. Purpose

- ☐ Wiring PLC and VFD on the Control panel board with 3phase IM
- ☐ Frequency setting for Forward control
- ☐ Reference table

2. Diagram

1) Setting PLC and VFD



2) PLC setting

CPU	P1	P2	P6	P7	P8
	FX	RX	drv group	I group	
			ST1, ST2, ST3	I30, I31, I32, I33, I34	
XGI	%QX0.0.0	%QX0.0.1	%QX0.2.4	%QX0.2.5	%QX0.2.6
XBC	P41	P42	P44	P45	P46



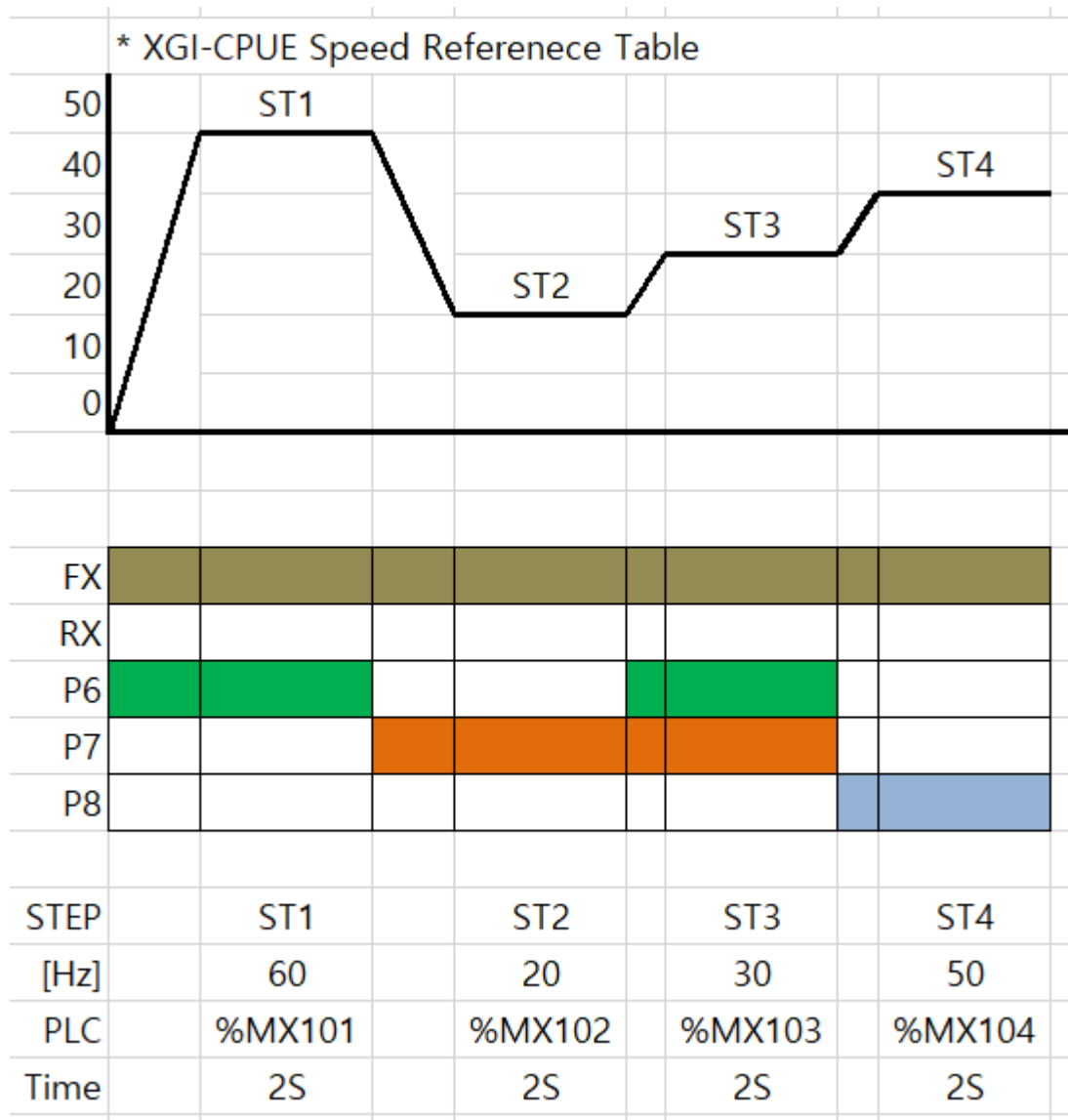
☐ Information

☐ Requirements

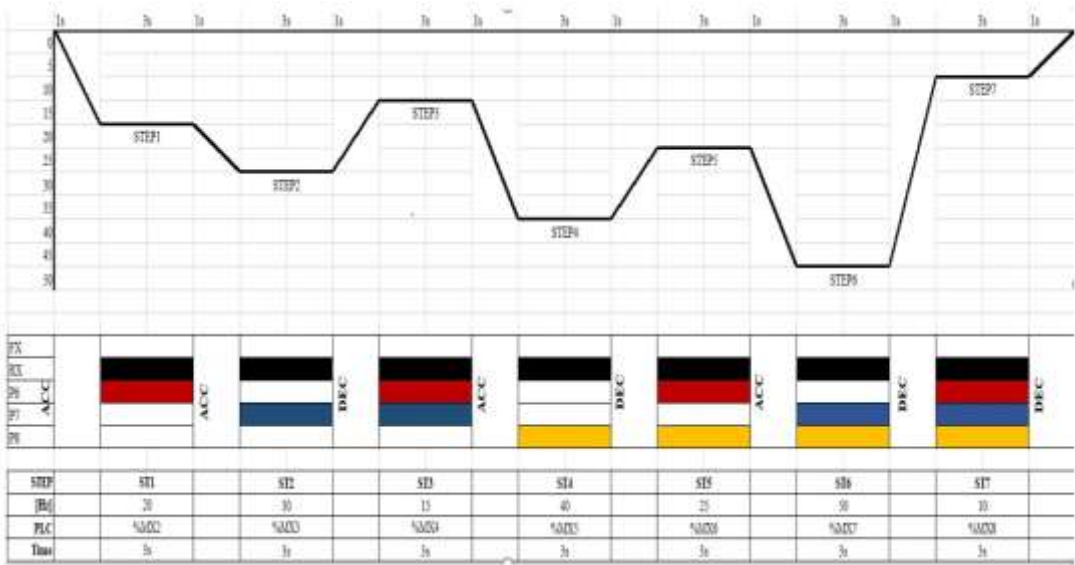
Job Sheet 7

Day	8	Task No.	EE-VFD-07	Time	8h
Title	Forward /Reverse control				
Material	Name	Spec	Q'ty		
	PLC	XBC-DR30SU	1		
	VFD	IG5A	1		
	Computer		1		

1. Forward control



2. Reverse control



□ Information

□ Requirements

Job Sheet 8

Day	9	Task No.	EE-VFD-08	Time	8h
Title	Multi step control				
Material	Name		Spec	Q'ty	
	PLC		XBC-DR30SU	1	
	VFD		IG5A	1	
	Computer			1	

1. Purpose

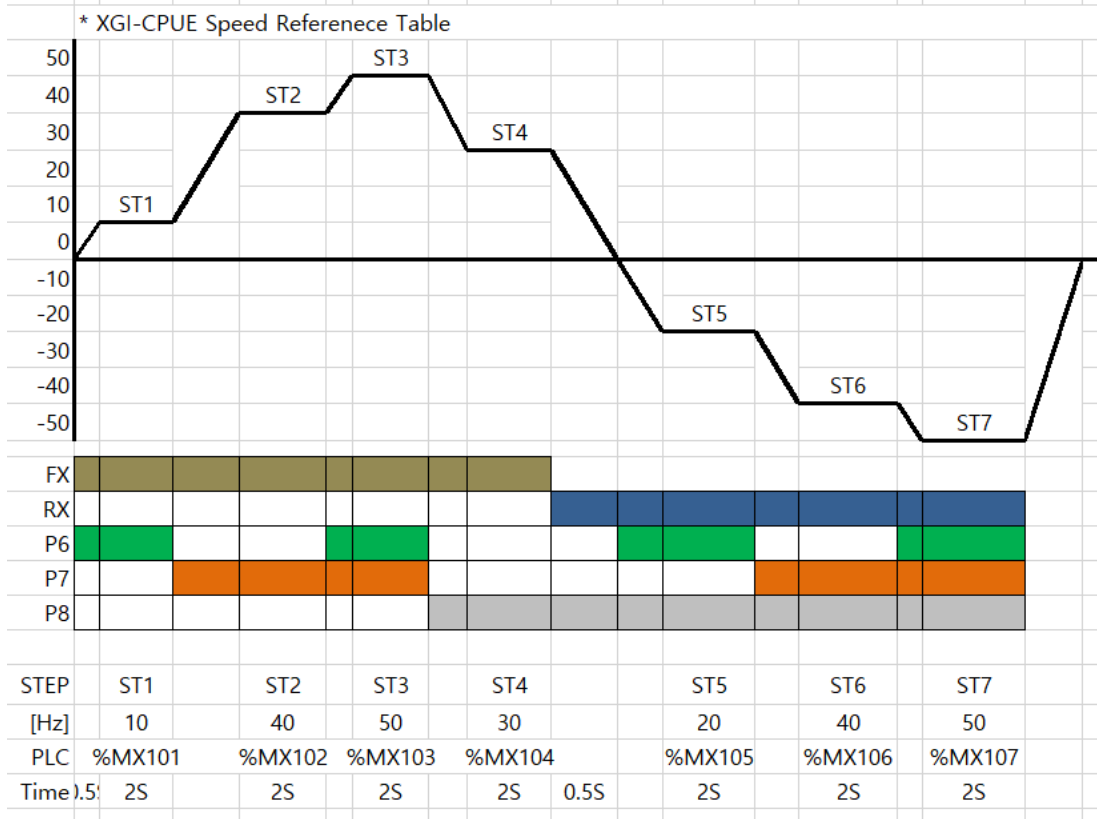
- ☐ Wiring PLC and VFD on the Control panel board with 3phase IM
- ☐ Frequency setting for Forward & Reverse control
- ☐ Reference table

2. Timetable

Time Table															
T \ PT	1S	1S	1S	1S	1S	1S	1S	1S	1S	1S	1S	1S	1S	1S	1S
T0															
T1															
T2															
T3															
T4															
T5															
T6															
T7															

PT	T0	T1	T2	T3	T4	T5	T6	T7
PT	T#1S	T#2S	T#2S	T#2S	T#2S	T#2S	T#2S	T#2S

3. Reference table



□ Information

□ Requirements

Job Sheet 9

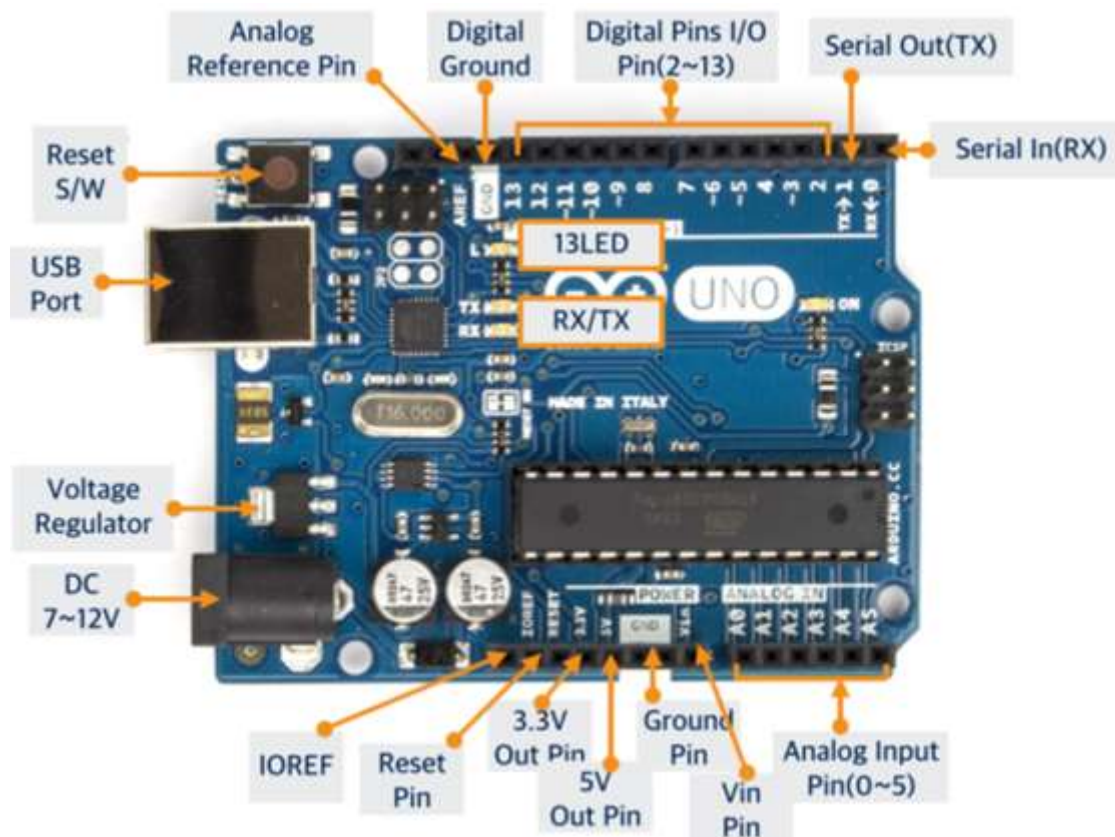
Day	09	Task No.	EE-Arduino-09	Time	8h
Title	Arduino setting and LED control				
Material	Name	Spec	Q'ty		
	Arduino	UNO	1		
	Computer		1		
	LED		3		

1. Purpose

- Wide use of automatic control
- About the structure of Arduino
- Arduino program control

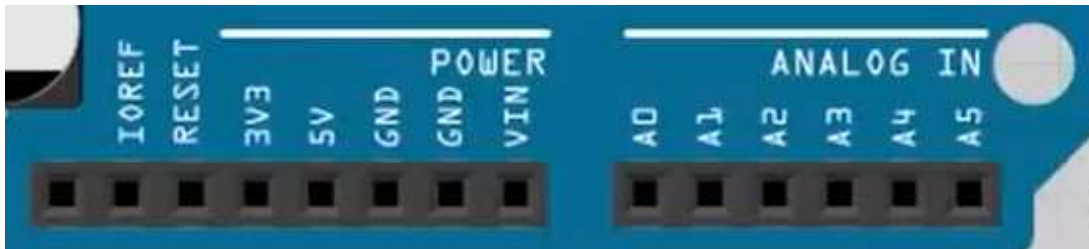
2. Structure of Arduino

1) Name of Arduino board parts



2) 2. Arduino I/O

□ Arduino Input & Power: DI, AI



□ Arduino Digital output & PWM: DO, AO

- PWM Mode Output: ~ displayed address



3. LED control

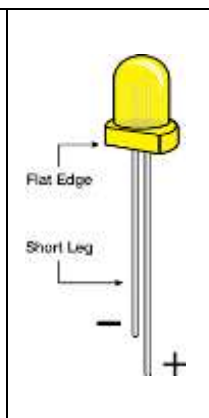
1) LED

LED is a type of semiconductor called "Light Emitting Diode" which is an electronic device made of semiconductor materials (silicon, selenium, germanium, etc.).

It is dubbed indicator, digital and word display in circuit and device.

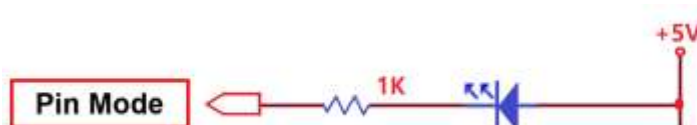
It has positive and negative poles.

The short leg is negative pole, the long one is positive pole.



LED : Light Emitting Diode

2) Sink (Anode common)



Output (V)	LED
0 (0V)	ON
1 (5V)	OFF

3) Source (Cathode common)



4) Sketch program for blinking

LED_ON | Arduino IDE 2.3.2

File Edit Sketch Tools Help

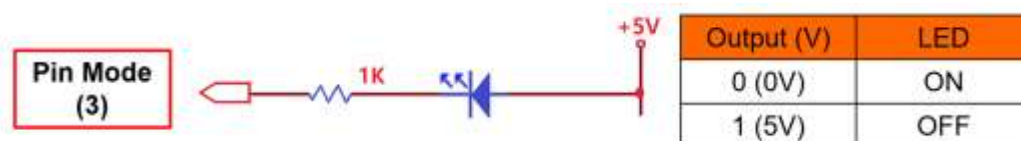
Arduino Uno

LED_ON.ino

```

1 void setup()
2 {
3   pinMode(3,OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(3,LOW);
9 }
10

```



5) 3LED control

□ Sketch program

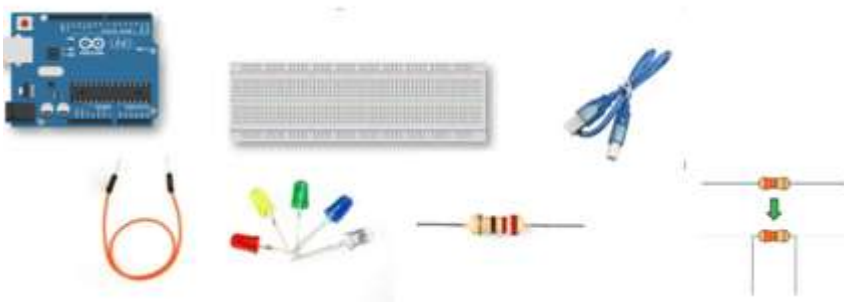
sketch_jan8b | Arduino IDE 2.3.2

File Edit Sketch Tools Help

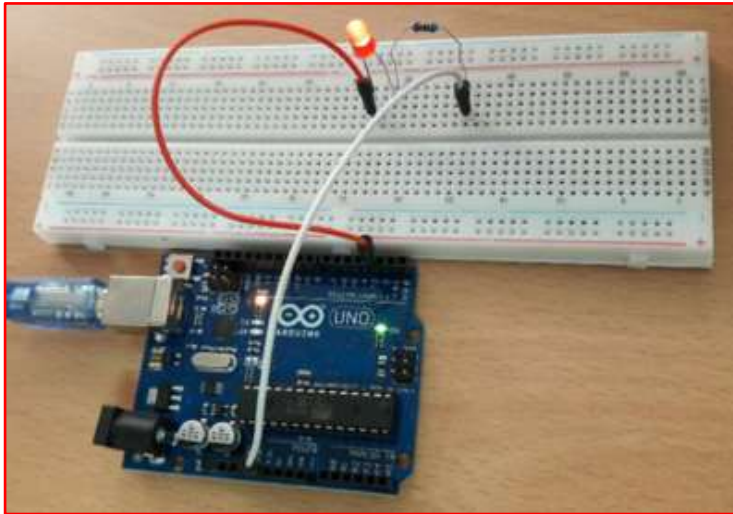
```
✓ → ⚙️ Arduino Uno

sketch_jan8b.ino
1  #define LED1 10
2  #define LED2 9
3  #define LED3 8
4
5  void setup() {
6      pinMode(LED1,OUTPUT);
7      pinMode(LED2,OUTPUT);
8      pinMode(LED3,OUTPUT);
9  }
10
11 void loop() {
12     digitalWrite(LED1, HIGH);
13     digitalWrite(LED2, LOW);
14     digitalWrite(LED3, HIGH);
15     delay(1000);
16     digitalWrite(LED1, LOW);
17     digitalWrite(LED2, HIGH);
18     digitalWrite(LED3, LOW);
19     delay(1000);
20 }
21
```

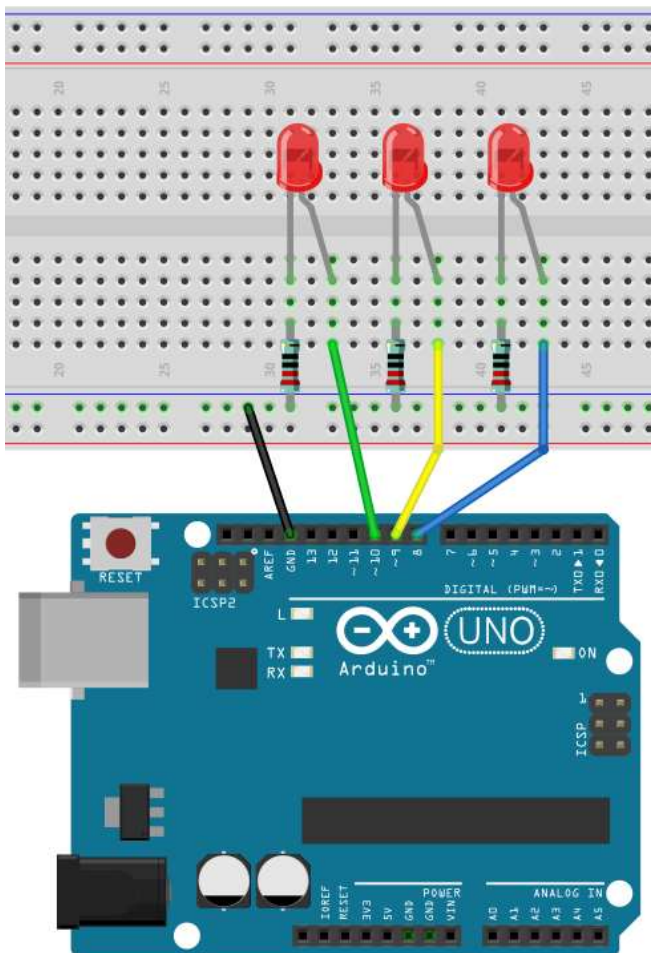
6) Material



7) Installation for blinking



8) Installation for 3 LED control



□ Information

□ Requirements

Job Sheet 10

Day	10	Task No.	EE-Arduino10	Time	8h
Title	RGB control				
Material	Name		Spec	Q'ty	
	Arduino		UNO	1	
	Computer			1	
	LED		RGB	1	

1. Purpose

- ☐ PWM (Pulse Width Modulation)
- ☐ Color code table
- ☐ How to control an RGB LED with Arduino

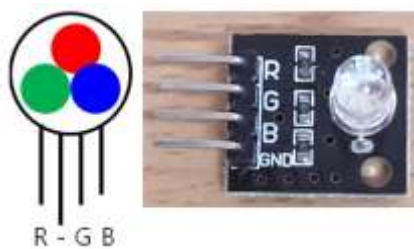
2. RGB

1) Definition

An RGB LED (**Red, Green, Blue** Light Emitting Diode) is a specialized type of LED that can emit light in three primary colors: red, green, and blue. By controlling the intensity of each of these colors, you can produce a wide range of colors, including white when all three are at full intensity

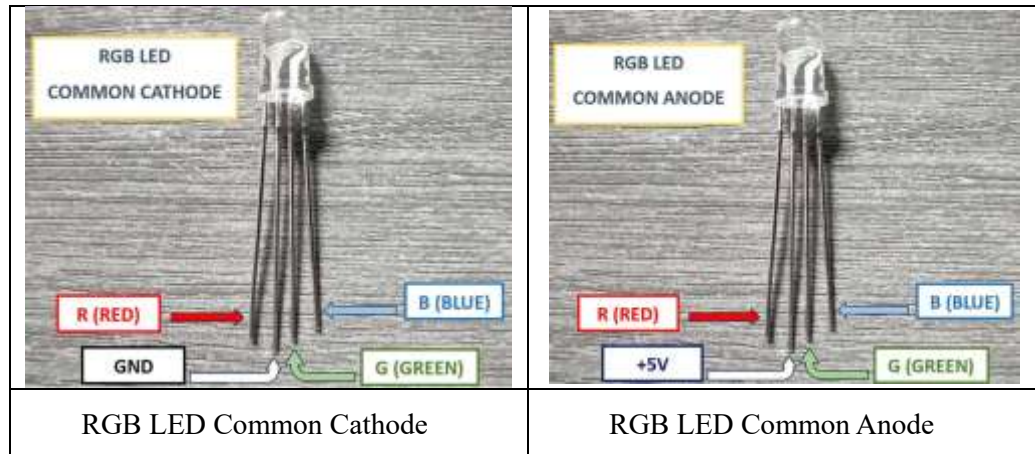
1) Basic structure

- ☐ An RGB LED contains three individual LEDs within a single package: one red, one green, and one blue.
- ☐ Each of these LEDs has a separate anode (positive terminal) and a common cathode (negative terminal) or a common anode and separate cathodes.



2) Pin

- ☐ If you have a common cathode RGB LED, one of the pins will be the cathode (usually the longest pin or one indicated by a flat edge on the package) which is connected to the common negative (-) terminal.
- ☐ If you have a common anode RGB LED, one of the pins will be the anode (usually the longest pin) which is connected to the common positive (+) terminal.



<Fig 7-2 RGB pin>

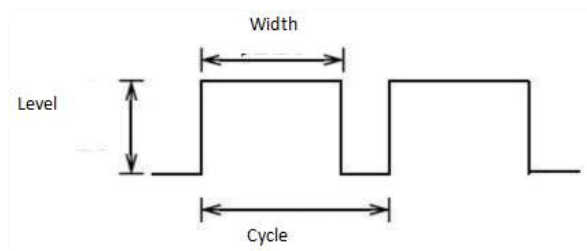
- The R, G, and B leads of the RGB LED should be connected to any PWM pin (digital pins 3,5,6,9,10,11).

3) Controlling the colors

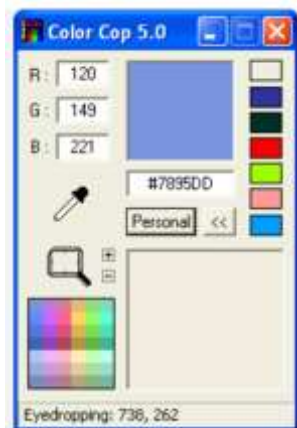
- To illuminate a specific color, you need to apply a voltage to the corresponding anode pin (for common anode) or cathode pin (for common cathode).
- For example, to make a red light, you apply a voltage across the red anode and the common cathode (or red cathode and common anode).

4) PWM

- Arduino can control the intensity of the LEDs by rapidly toggling them on and off.
- This is called PWM. It simulates an analog voltage by rapidly switching the LED on and off, effectively changing the average voltage.
- For example, if you want a dim red, you might PWM the red pin at a lower duty cycle.



5) Colorcop (<http://colorcop.net/>)



3. Sketch program of RGB

1) RGB LED – Common cathode

```
// This sketch is for the common cathode type of RGB LED
// for common cathode, 255 means full color display and 0 means color is not
displayed

int r_led= 9;
int g_led = 10;
int b_led = 11;

void setup()
{
  pinMode(r_led, OUTPUT);
  pinMode(g_led, OUTPUT);
  pinMode(b_led, OUTPUT);
}

void loop()
{
  // Red color (full red, no green, no blue)
  analogWrite(r_led, 255)
  analogWrite(g_led, 0);
  analogWrite(b_led, 0);

  delay(1000); analogWrite

  // Green color (no red, full green, no blue)
  analogWrite (r_led, 0);
  analogWrite(g_led, 255);
  analogWrite(b_led, 0);

  delay(1000);
```

```

// Blue color (no red, no green, full blue)
analogWrite(r_led, 0);
analogWrite(g_led, 0);
analogWrite(b_led, 255);

delay(1000);

// Purple color (full red, no green, full blue)
analogWrite(r_led, 255);
analogWrite(g_led, 0);
analogWrite(b_led, 255);

delay(1000);
}

```

2) RGB LED – Common Anode

```

// This sketch is for the common anode type of RGB LED
// for common anode, 0 means full color display and means color is not displayed
int r_led= 9;
int g_led = 10;
int b_led = 11;

void setup()
{
  pinMode(r_led, OUTPUT);
  pinMode(g_led, OUTPUT);
  pinMode(b_led, OUTPUT);
}

```

```
void loop()
{
  // Red color (full red, no green, no blue)
  analogWrite(r_led, 0)
  analogWrite(g_led, 255);
  analogWrite(b_led, 255);

  delay(1000); analogWrite

  // Green color (no red, full green, no blue)
  analogWrite (r_led, 255);
  analogWrite(g_led, 0);
  analogWrite(b_led, 255);

  delay(1000);

  // Blue color (no red, no green, full blue)
  analogWrite(r_led, 255);
  analogWrite(g_led, 255);
  analogWrite(b_led, 0);

  delay(1000);
  // Purple color (full red, no green, full blue)
  analogWrite(r_led, 0);
  analogWrite(g_led, 255);
  analogWrite(b_led, 0);

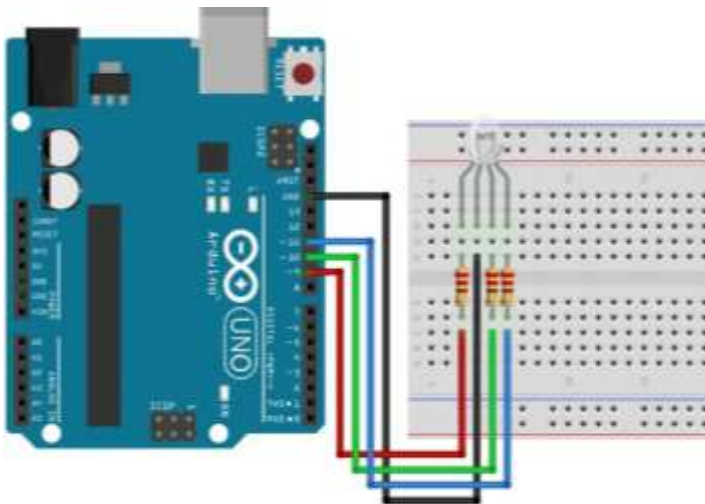
  delay(1000);
}
```

4. Wiring of RGB

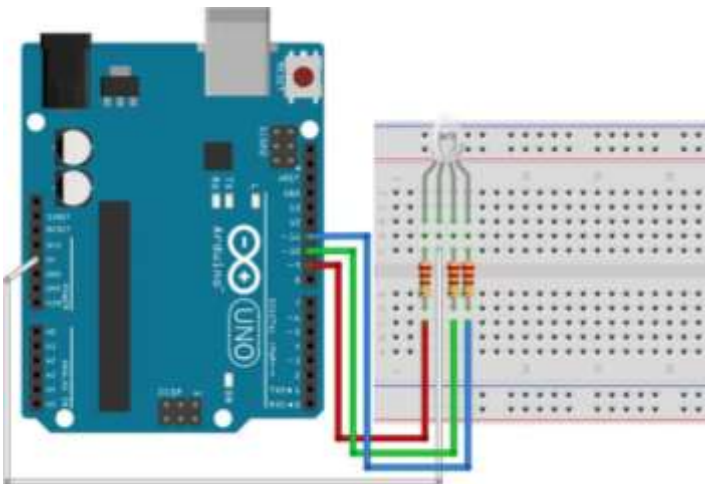
1) Equipment and material

- ☐ Arduino board
- ☐ Breadboard
- ☐ USB-B cable
- ☐ Jumper wire
- ☐ RGB- 3
- ☐ 220ohm- 3

2) Installation_ RGB LED circuit for common cathode



3) Installation_ RGB LED circuit for common Anode



- ☐ Information

- ☐ Requirements