

번호: 8번 학번: 21101395 이름: 권준혁 제출 일시: 5월 11일

- 마감 시간을 지키지 않으면 미제출로 간주한다. (확인 여부 표시) - 확인
- 이메일에 파일이 첨부되지 않으면 미제출로 간주한다. (확인 여부 표시) - 확인
- E-mail 주소를 잘못 적어서 제출하지 못한 경우는 미제출로 간주한다. (확인 여부 표시) - 확인
- 문서에 그림을 포함시키지 않으면 미제출로 간주한다. (확인 여부 표시) - 확인

1. 실습1

PS2 Joystick Module의 VRX - Arduino UNO R3 A4

PS2 Joystick Module의 VRY - Arduino UNO R3 A5

PS2 Joystick Module의 SW - Arduino UNO R3 digital pin 8

LED Module의 R - Arduino UNO R3 9

LED Module의 Y - Arduino UNO R3 10

LED Module의 G - Arduino UNO R3 11

serial data rate는 19200 baud로 설정한다.

- Serial monitor를 통해서 하나의 문자를 수신할 때만 동작한다. 수신된 문자가 'x' or 'X' 인 경우 PS2 Joystick Module의 VRX pin 전압의 크기를 0에서 1023 사이의 값으로 변환해서 읽는다. 그 값을 0에서 255 사이의 값으로 변환하고 변환된 값에 비례하는 PWM 파형을 빨간색 LED에 가한다. 수신된 문자가 'y' or 'Y' 인 경우 PS2 Joystick Module의 VRY pin 전압의 크기를 0에서 1023 사이의 값으로 변환해서 읽는다. 그 값을 0에서 255 사이의 값으로 변환하고 변환된 값에 비례하는 PWM 파형을 노란색 LED에 가한다. 수신된 문자가 's' or 'S' 인 경우 PS2 Joystick Module의 스위치가 눌렸는지를 읽는다. 스위치가 눌렸다면 초록색 LED가 현재 OFF 상태이면 ON 시키고 ON 상태이면 OFF 시킨다. 모든 경우마다 아래 결과 그림에서 보이는 것과 같은 문장을 serial monitor로 출력한다. 맨 마지막에 1초간 delay를 추가한다. 다른 문자가 수신되면 아무 동작도 수행하지 않는다.

1.1 하드웨어 구성 결과

그림 1 실습1 하드웨어 구성 결과

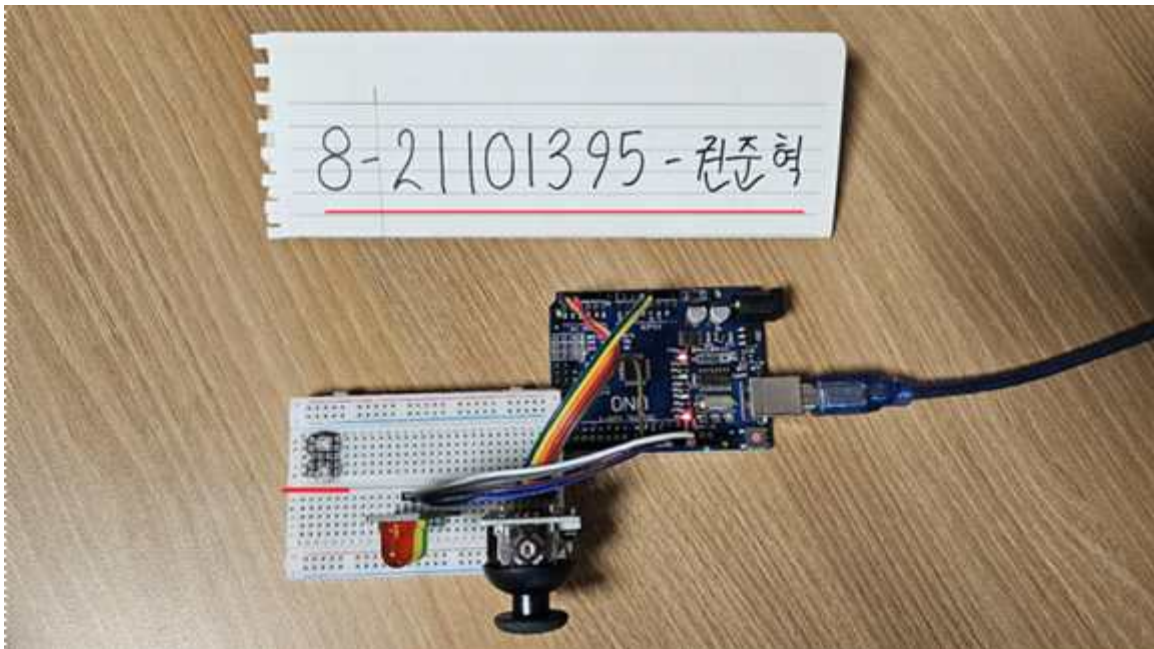


그림 1은 실습1 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.

1.2 소프트웨어 작성 결과 (새 페이지)

<p>그림 2 실습1 소프트웨어 작성 결과</p> <pre>8_21101395_Arduino_Report2_Lab1.ino 1 // 8-21101395-권준혁 Arduino과제2 실습1 2 #define VRX_PIN A4 3 #define VRY_PIN A5 4 #define SW_PIN 8 5 #define LED_RED_PIN 9 6 #define LED_YELLOW_PIN 10 7 #define LED_GREEN_PIN 11 8 9 void setup() { 10 Serial.begin(19200); 11 pinMode(VRX_PIN, INPUT); 12 pinMode(VRY_PIN, INPUT); 13 pinMode(SW_PIN, INPUT_PULLUP); 14 pinMode(LED_RED_PIN, OUTPUT); 15 pinMode(LED_YELLOW_PIN, OUTPUT); 16 pinMode(LED_GREEN_PIN, OUTPUT); 17 Serial.println("8-21101395-권준혁> setup()"); 18 } 19 20 void loop() { 21 if (Serial.available() > 0) { 22 char receivedChar = Serial.read(); 23 int value; 24 if (receivedChar == 'x' receivedChar == 'X') { 25 int vrxValue = analogRead(VRX_PIN); 26 value = map(vrxValue, 0, 1023, 0, 255); 27 analogWrite(LED_RED_PIN, value); 28 Serial.print("VRX = "); 29 Serial.print(vrxValue); 30 Serial.print(", PWM Value of VRX = "); 31 Serial.println(value);</pre>	<p>그림 3 실습1 소프트웨어 작성 결과 계속</p> <pre>32 } else if (receivedChar == 'y' receivedChar == 'Y') { 33 int vryValue = analogRead(VRY_PIN); 34 value = map(vryValue, 0, 1023, 0, 255); 35 analogWrite(LED_YELLOW_PIN, value); 36 Serial.print("VRY = "); 37 Serial.print(vryValue); 38 Serial.print(", PWM Value of VRY = "); 39 Serial.println(value); 40 } else if (receivedChar == 's' receivedChar == 'S') { 41 if (digitalRead(SW_PIN) == LOW) { 42 if (digitalRead(LED_GREEN_PIN) == LOW) { 43 digitalWrite(LED_GREEN_PIN, HIGH); 44 Serial.println("SW pressed : LED ON"); 45 } else { 46 digitalWrite(LED_GREEN_PIN, LOW); 47 Serial.println("SW pressed : LED OFF"); 48 } 49 } 50 } 51 delay(1000); 52 } 53 } 54 }</pre>
<p>그림 2, 3은 실습1 소프트웨어 작성 결과를 보여준다.</p> <ul style="list-style-type: none">- 출석번호 = 8번- 파일 이름은 반드시 번호_학번_Arduino_Report2_Lab1- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함	

1.3 프로그램 실행 결과 1 (새 페이지)

그림 4 실습1 프로그램 실행 결과



그림 4는 실습1 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 순서로 출력되도록 동작시켜서 결과를 보인다.

=> setup() 함수에서 첫줄 출력
=> VRX가 최대가 되도록 PS2 Joystick Module을 조절하고 serial monitor에서 x (X)를 입력
=> VRY가 최대가 되도록 PS2 Joystick Module을 조절하고 serial monitor에서 y (Y)를 입력
=> PS2 Joystick Module의 스위치를 누르고 serial monitor에서 s (S)를 입력
=> VRX가 최소가 되도록 PS2 Joystick Module을 조절하고 serial monitor에서 x (X)를 입력
=> VRY가 최소가 되도록 PS2 Joystick Module을 조절하고 serial monitor에서 y (Y)를 입력
=> PS2 Joystick Module의 스위치를 누르고 serial monitor에서 s (S)를 입력

1.4 프로그램 실행 결과 2

그림 5 실습1 프로그램 실행 결과

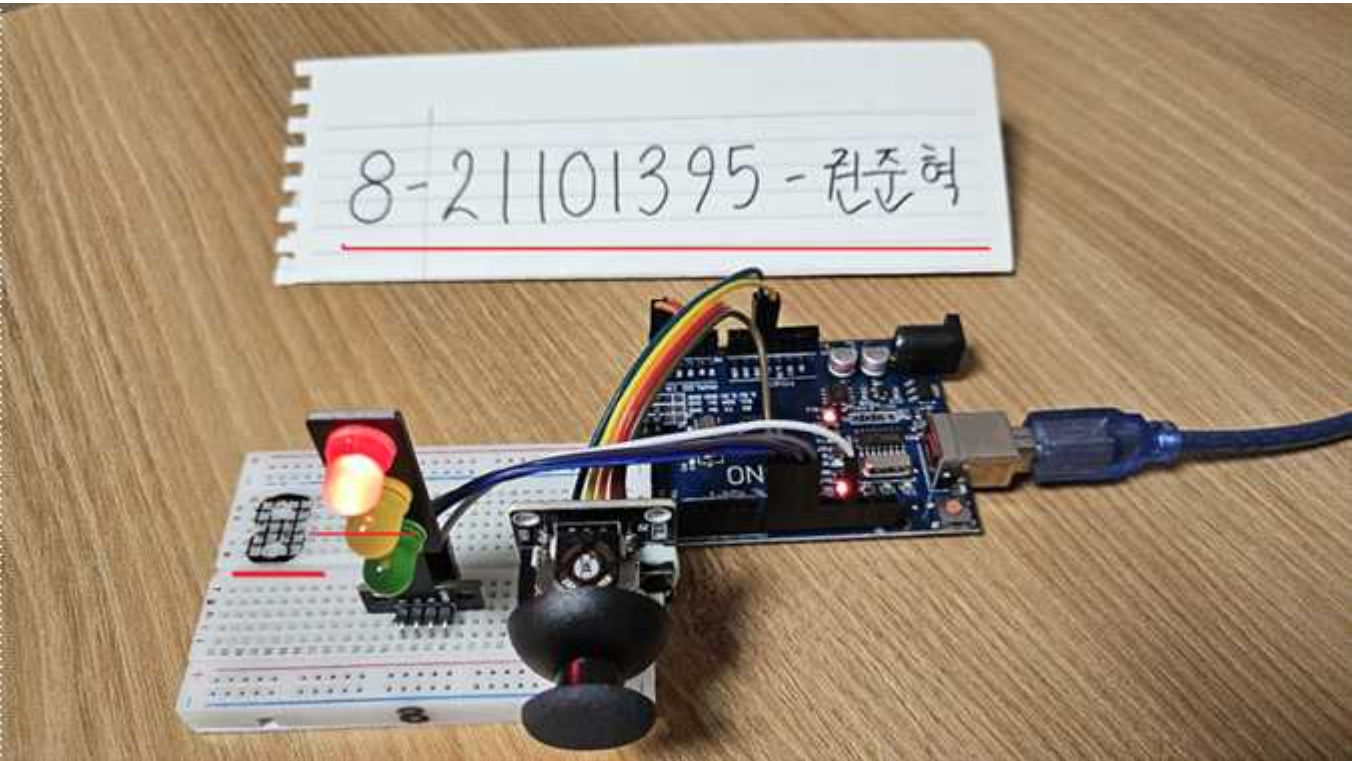


그림 5는 실습1 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- VRX 값이 최대가 되도록 PS2 Joystick Module을 조절한 결과 빨간색 LED가 최대 밝기가 된다.

그림 6 실습1 프로그램 실행 결과



그림 6은 실습1 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- VRY 값이 최대가 되도록 PS2 Joystick Module을 조절한 결과 노란색 LED가 최대 밝기가 된다.

1.6 프로그램 실행 결과 4 (새 페이지)

그림 7 실습1 프로그램 실행 결과

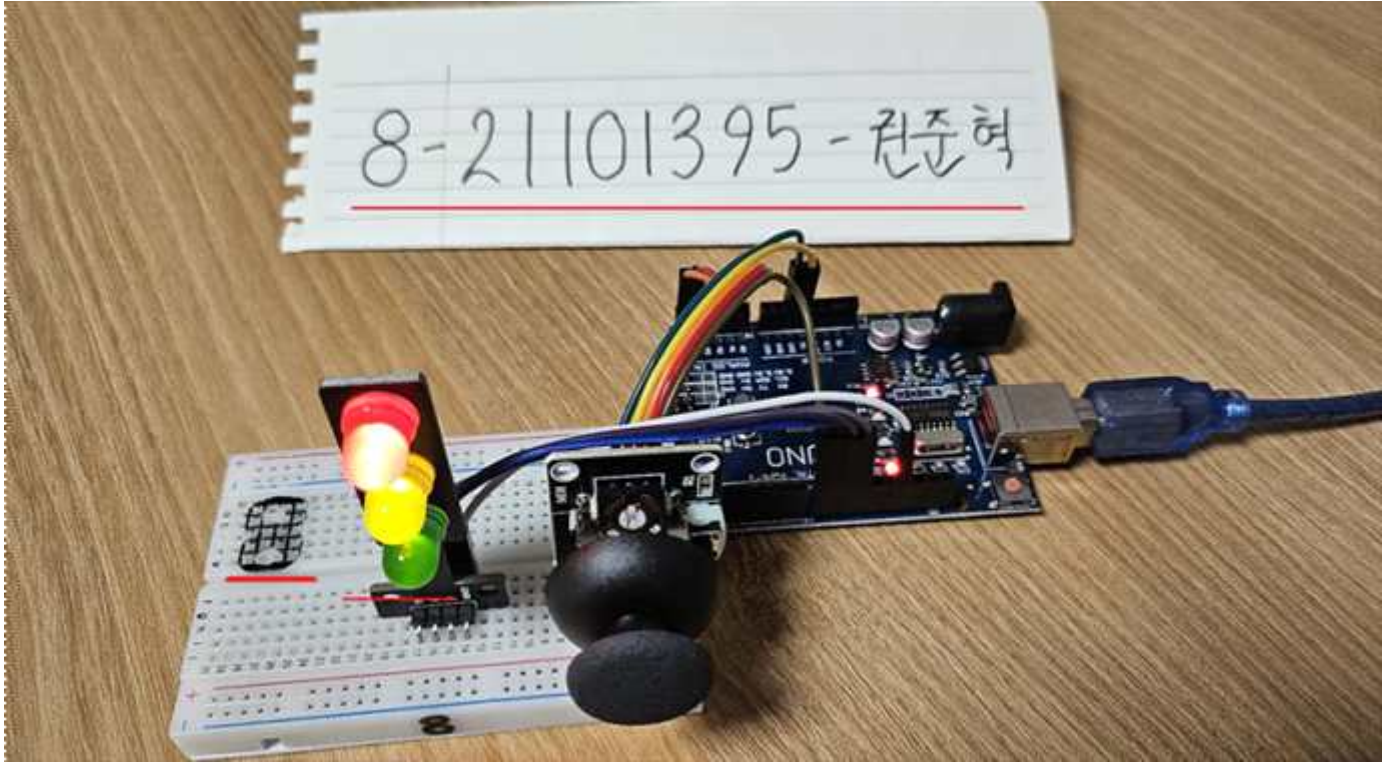


그림 7은 실습1 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- 처음 PS2 Joystick Module의 스위치를 누른 결과 꺼져있던 초록색 LED가 켜진다.

2. 실습2 (새 페이지)

Single Channel Relay Module의 OUT - Arduino UNO R3 digital PIN 12

Single Channel Relay Module의 COM - Arduino UNO R3 3.3V

Single Channel Relay Module의 NO - LED Module의 G

Single Channel Relay Module의 NC - LED Module의 R

IR Sensor Module의 OUT - Arduino UNO R3 digital PIN 13

serial data rate는 31250 baud로 설정한다.

- IR Sensor Module에서 물체가 감지되지 않으면 Single Channel Relay Module을 사용해서 초록색 LED를 켜다. IR Sensor Module에서 물체가 감지되면 Single Channel Relay Module을 사용해서 빨간색 LED를 켜다. loop() 함수를 종료하기 전에 100[ms] delay를 추가한다. 물체 감지 상태가 변하는 경우에만 LED를 켜야 한다.

2.1 하드웨어 구성 결과

그림 8 실습2 하드웨어 구성 결과

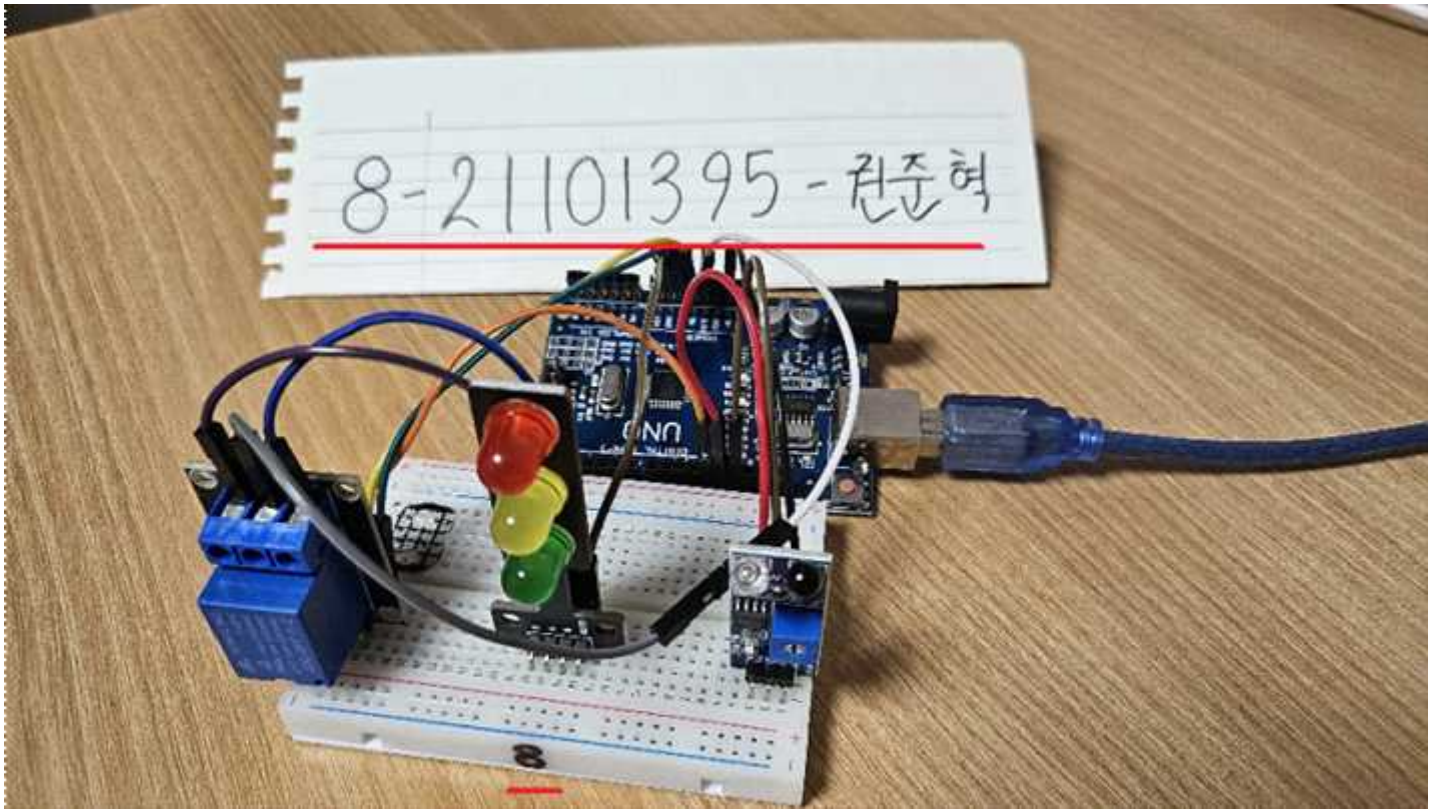


그림 8은 실습2 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.

그림 9 실습2 소프트웨어 작성 결과

```

8_21101395_Arduino_Report2_Lab2.ino
1  // 8-21101395-권준혁 Arduino과제2 실습2
2  #define RELAY_SIG_PIN 12
3  #define IR_OUT_PIN 13
4
5  bool objectDetected = false;
6  bool previousState = false;
7
8  void setup() {
9      Serial.begin(31250);
10     pinMode(RELAY_SIG_PIN, OUTPUT);
11     pinMode(IR_OUT_PIN, INPUT);
12     Serial.println("8-21101395-권준혁> setup()");
13 }
14
15 void loop() {
16
17     bool currentState = digitalRead(IR_OUT_PIN);
18
19     if (currentState != previousState) {
20         if (currentState == HIGH) {
21             digitalWrite(RELAY_SIG_PIN, HIGH);
22             Serial.println("IR Dection Sensor = 0 = Detected!");
23         } else {
24             digitalWrite(RELAY_SIG_PIN, LOW);
25             Serial.println("IR Dection Sensor = 1 = Not Detected!");
26         }
27         objectDetected = currentState;
28     }
29     previousState = currentState;
30     delay(100);
31 }

```

그림 9는 실습2 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호_학번_Arduino_Report2_Lab2
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함

2.3 프로그램 실행 결과 1 (새 페이지)

그림 10 실습2 프로그램 실행 결과



그림 10은 실습2 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
 - 반드시 그림과 같은 순서로 출력되도록 동작시켜서 결과를 보인다.
- => setup() 함수에서 첫줄 출력
- => 물체를 IR Sensor Module에 가까이 위치시켜서 감지되도록 한다.
- => 물체를 IR Sensor Module에서 치운다.
- => 물체를 IR Sensor Module에 가까이 위치시켜서 감지되도록 한다.
- => 물체를 IR Sensor Module에서 치운다.

2.4 프로그램 실행 결과 2

그림 11 실습2 프로그램 실행 결과

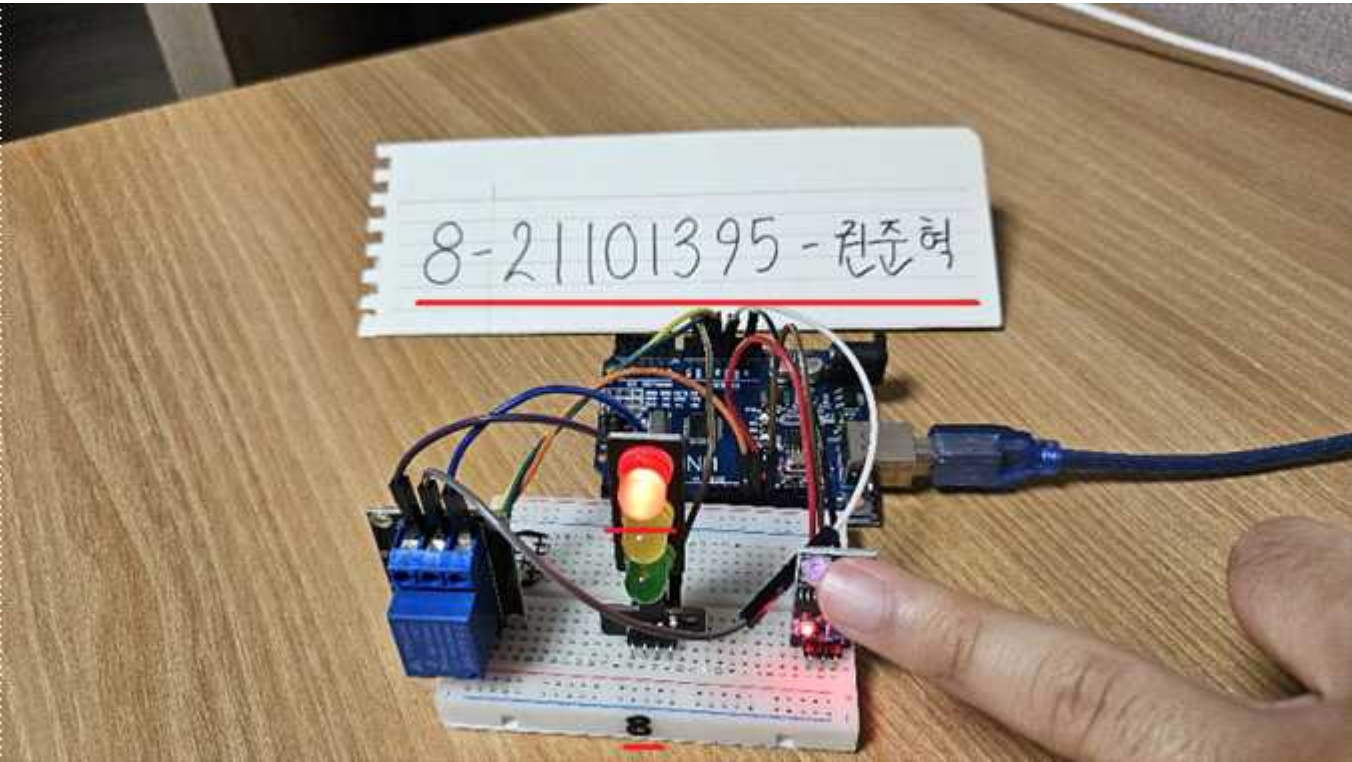


그림 11은 실습2 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- 물체를 IR Sensor Module에 가까이 위치시켜서 감지되도록 한 결과 빨간색 LED가 켜진다.

2.5 프로그램 실행 결과 3 (새 페이지)

그림 12 실습2 프로그램 실행 결과

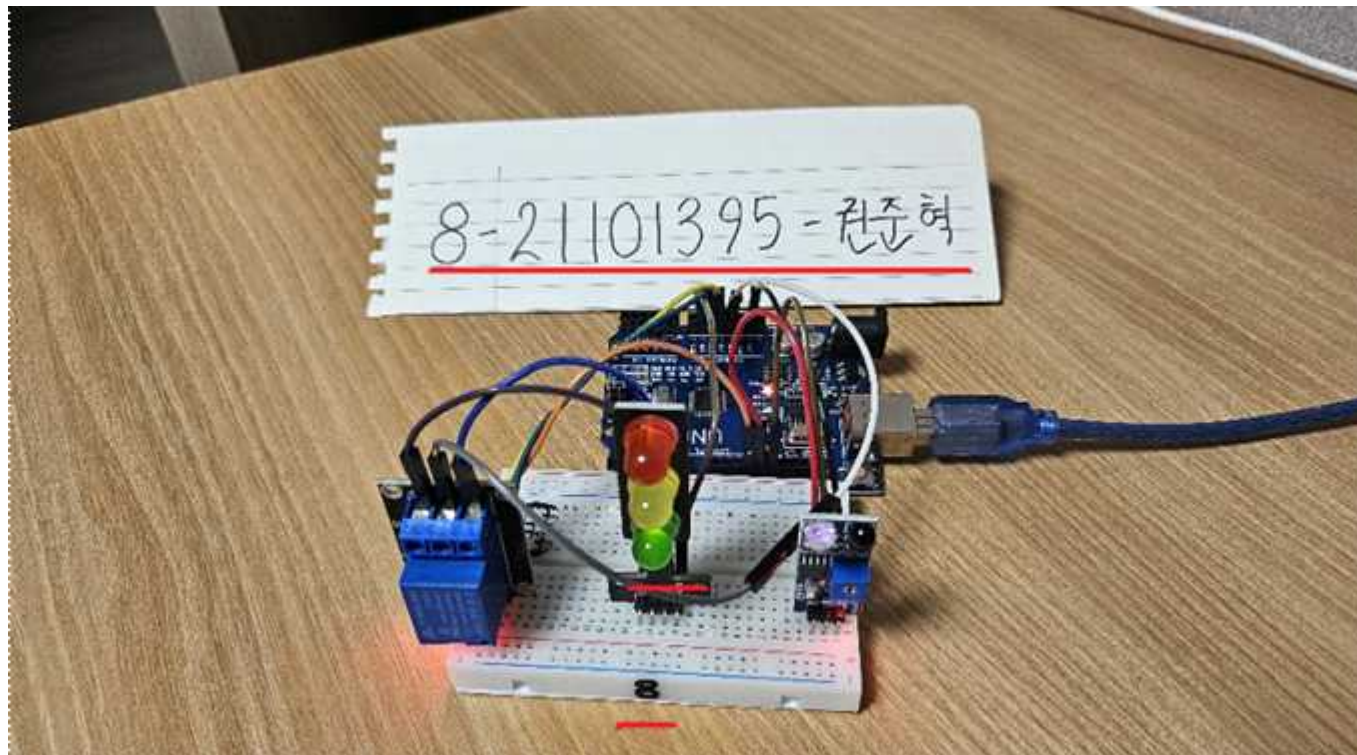


그림 12는 실습2 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- 물체를 IR Sensor Module에서 치운 결과 초록색 LED가 켜진다.

3. 실습3 (새 페이지)

Ultrasonic Sensor Module HC-SR04의 Trig - Arduino UNO R3 digital PIN 8

Ultrasonic Sensor Module HC-SR04의 Echo - Arduino UNO R3 digital PIN 9

LED Module의 R - Arduino UNO R3 10

LED Module의 Y - Arduino UNO R3 11

LED Module의 G - Arduino UNO R3 12

serial data rate는 19200 baud로 설정한다.

- Serial monitor를 통해서 하나의 문자를 수신할 때만 동작한다. 수신된 문자가 'm' or 'M' 인 경우 Ultrasonic Sensor Module HC-SR04을 사용해서 물체와의 거리를 측정한다. 다만, 마지막 측정 후로부터 1초 이상의 시간이 지난 경우에만 측정을 수행한다. 측정된 거리가 15[cm] 이상이면 빨간색 LED를 켜다. 측정된 거리가 10[cm] 이상 15[cm] 미만이면 노란색 LED를 켜다. 측정된 거리가 10[cm] 미만이면 초록색 LED를 켜다. LED는 하나만 켜지도록 해야 한다. 다른 문자가 수신되면 아무 동작도 수행하지 않는다.

3.1 하드웨어 구성 결과

그림 13 실습3 하드웨어 구성 결과

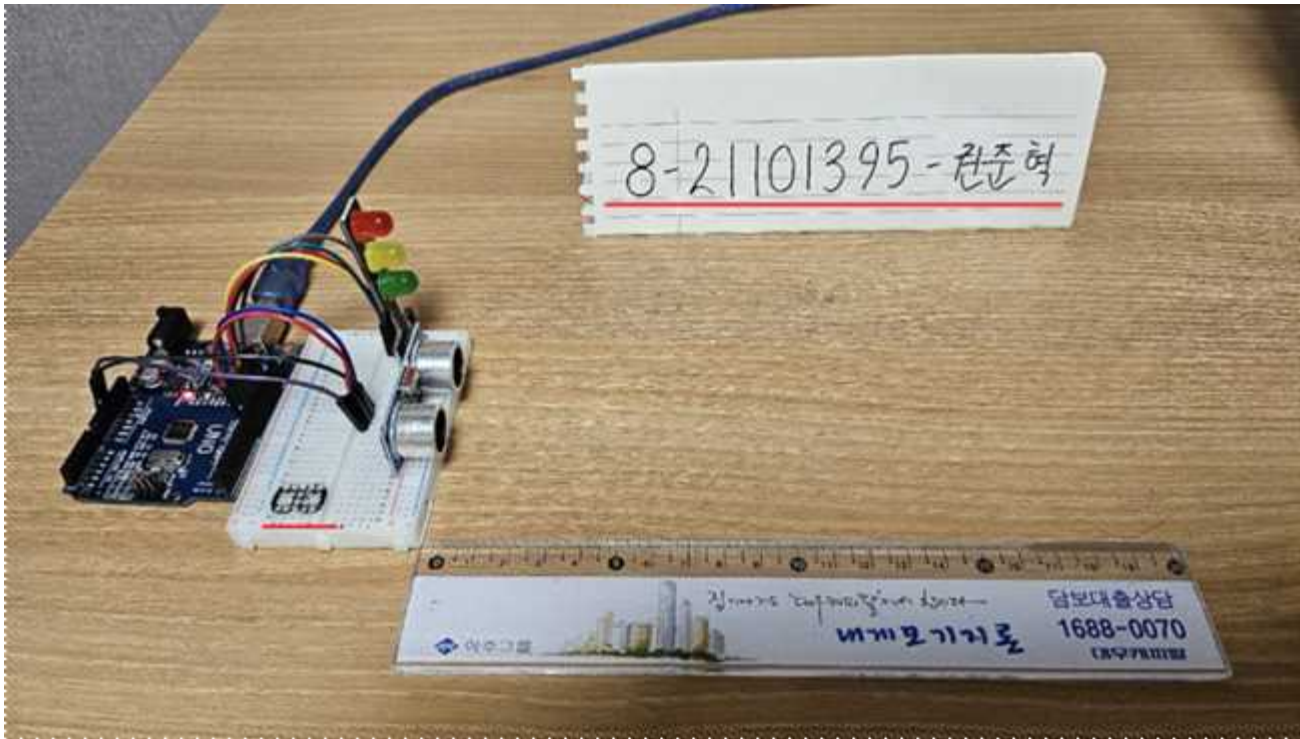


그림 13은 실습3 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.

그림 14 실습3 소프트웨어 작성 결과

```
8_21101395_Arduino_Report2_Lab3.ino
1 //8-21101395-권준혁 Arduino과제2 실습3
2 #define LED_RED_PIN 10
3 #define LED_YELLOW_PIN 11
4 #define LED_GREEN_PIN 12
5 #define TRIG_PIN 8
6 #define ECHO_PIN 9
7 #define MEASURE_INTERVAL 1000
8 #define SPEED_OF_SOUND 343
9
10 unsigned long previousMillis = 0;
11
12 void setup() {
13   Serial.begin(19200);
14   pinMode(LED_RED_PIN, OUTPUT);
15   pinMode(LED_YELLOW_PIN, OUTPUT);
16   pinMode(LED_GREEN_PIN, OUTPUT);
17   pinMode(TRIG_PIN, OUTPUT);
18   pinMode(ECHO_PIN, INPUT);
19   Serial.println("8-21101395-권준혁> setup()");
20 }
21
22 void loop() {
23   if (Serial.available() > 0) {
24     char receivedChar = Serial.read();
25     if (receivedChar == 'm' || receivedChar == 'M') {
26       unsigned long currentMillis = millis();
27       if (currentMillis - previousMillis >= MEASURE_INTERVAL) {
28         previousMillis = currentMillis;
29         float duration, distance;
30         unsigned long measureTime = millis();
31         duration = measureDistance(&distance);
32         Serial.print(" measureTime = ");
33         Serial.print(measureTime);
```

그림 15 실습3 소프트웨어 작성 결과

```
34   Serial.print(" duration = ");
35   Serial.print(duration);
36   Serial.print(" distance = ");
37   Serial.print(distance);
38   Serial.println("cm");
39   if (distance >= 15) {
40     turnOnLED(LED_RED_PIN);
41     turnOffLED(LED_YELLOW_PIN);
42     turnOffLED(LED_GREEN_PIN);
43   } else if (distance >= 10) {
44     turnOffLED(LED_RED_PIN);
45     turnOnLED(LED_YELLOW_PIN);
46     turnOffLED(LED_GREEN_PIN);
47   } else {
48     turnOffLED(LED_RED_PIN);
49     turnOffLED(LED_YELLOW_PIN);
50     turnOnLED(LED_GREEN_PIN);
51   }
52 }
53 }
54 }
55 }
56
57 float measureDistance(float *distance) {
58   digitalWrite(TRIG_PIN, LOW);
59   delayMicroseconds(2);
60   digitalWrite(TRIG_PIN, HIGH);
61   delayMicroseconds(10);
62   digitalWrite(TRIG_PIN, LOW);
63   float duration = pulseIn(ECHO_PIN, HIGH);
64   *distance = duration/2.0*SPEED_OF_SOUND*100/1000000;
65   return duration;
66 }
67
68 void turnOnLED(int pin) {
69   digitalWrite(pin, HIGH);
70 }
71
72 void turnOffLED(int pin) {
73   digitalWrite(pin, LOW);
74 }
```

그림 14, 15는 실습3 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호_학번_Arduino_Report2_Lab3
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함

3.3 프로그램 실행 결과 1 (새 페이지)

그림 16 실습3 프로그램 실행 결과

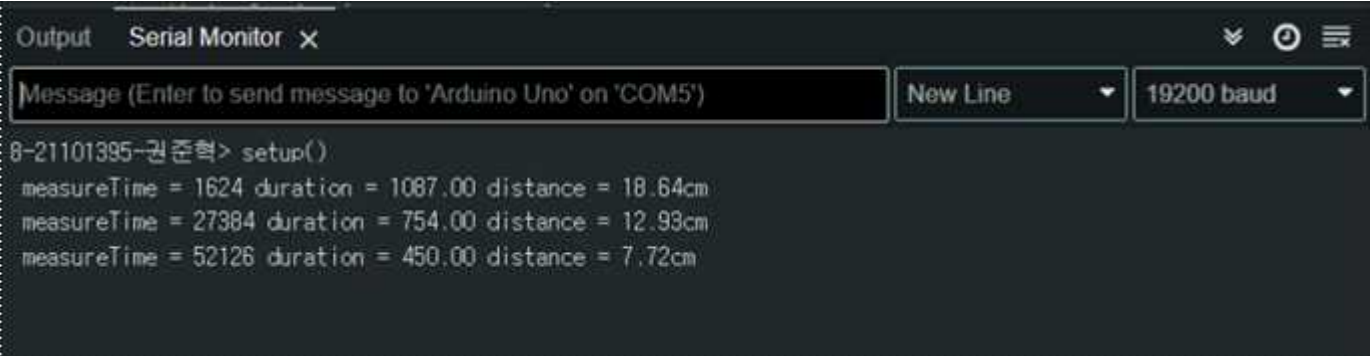


그림 16은 실습3 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
 - 반드시 그림과 같은 순서로 출력되도록 동작시켜서 결과를 보인다.
- => setup() 함수에서 첫줄 출력
- => 물체를 15[cm] 보다 먼 거리에 위치시키고 거리를 측정한 결과로 거리는 18.64[cm]가 된다.
- => 물체를 10[cm]와 15[cm] 사이에 위치시키고 거리를 측정한 결과로 거리는 12.93[cm]가 된다.
- => 물체를 10[cm]와 15[cm] 사이에 위치시키고 거리를 측정한 결과로 거리는 7.72[cm]가 된다.

3.4 프로그램 실행 결과 2

그림 17 실습3 프로그램 실행 결과

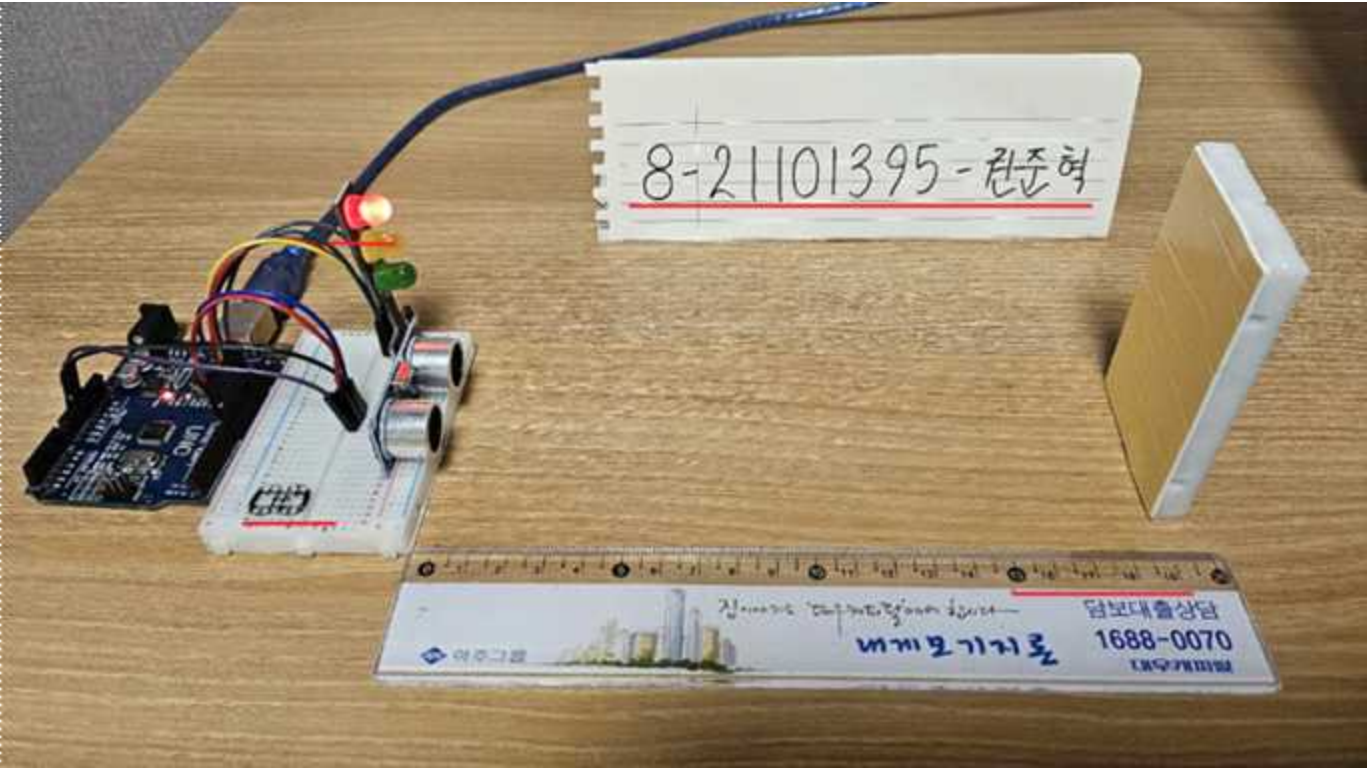


그림 17은 실습3 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- 물체를 15[cm] 보다 먼 거리에 위치시키고 거리를 측정한 결과로 빨간색 LED가 켜진다.
- 반드시 자의 눈금이 선명히 보여야 한다.

3.5 프로그램 실행 결과 3 (새 페이지)

그림 18 실습3 프로그램 실행 결과

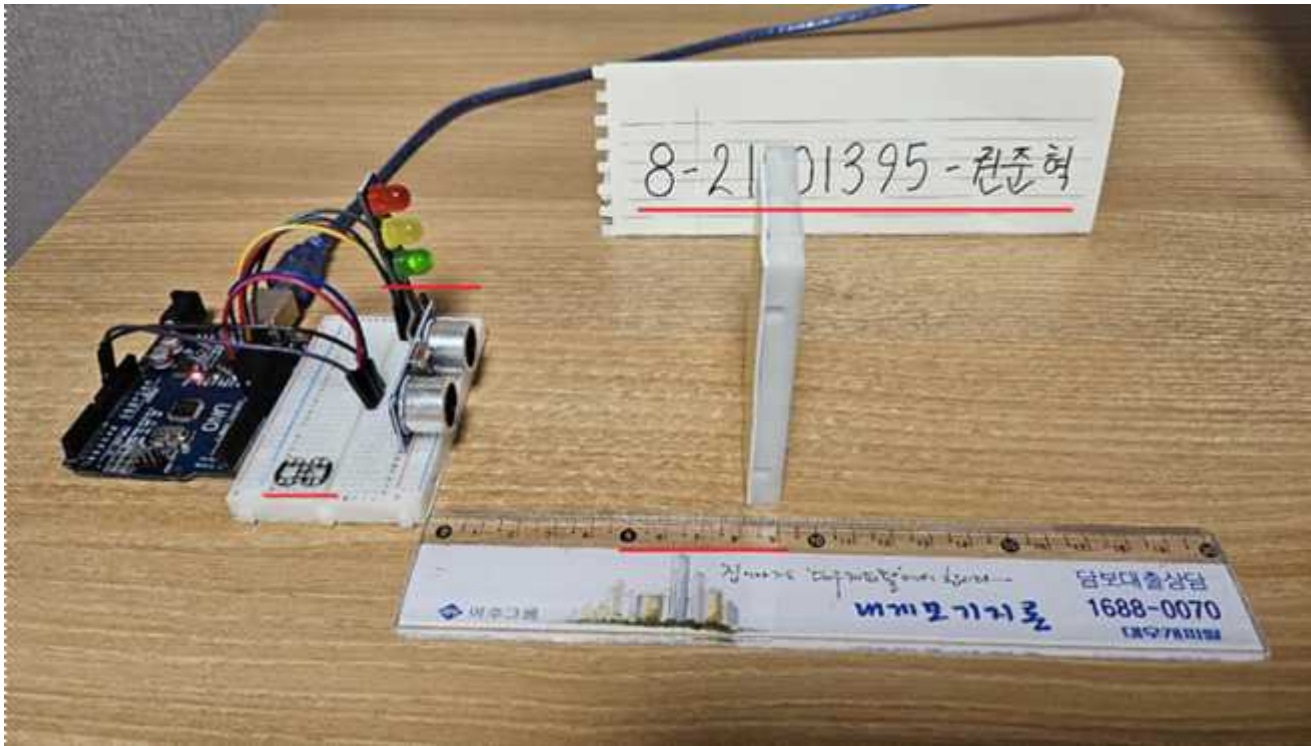


그림 18은 실습3 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- 물체를 10[cm]와 15[cm] 사이에 위치시키고 거리를 측정한 결과로 노란색 LED가 켜진다.
- 반드시 자의 눈금이 선명히 보여야 한다.

그림 19 실습3 프로그램 실행 결과

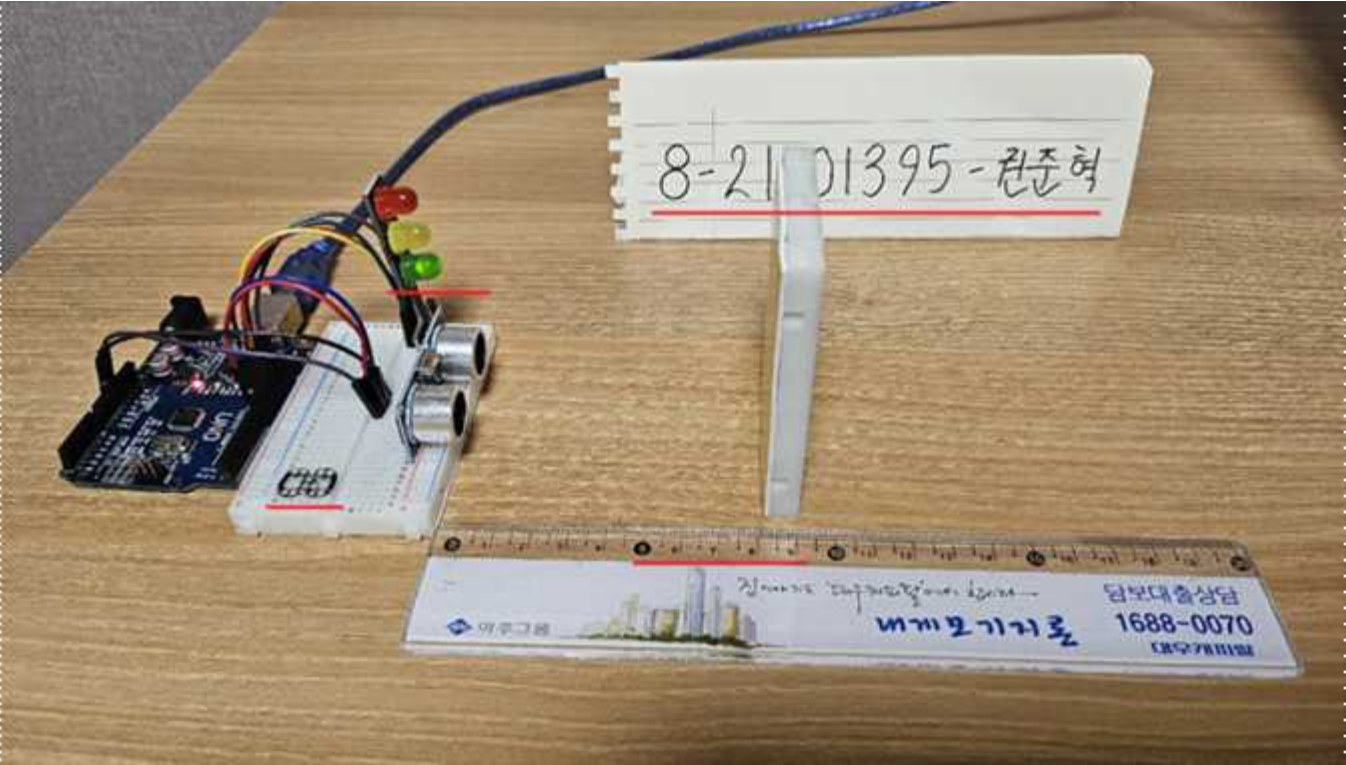


그림 19는 실습3 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- 물체를 10[cm] 보다 작은 거리에 위치시키고 거리를 측정한 결과로 초록색 LED가 켜진다.
- 반드시 자의 눈금이 선명히 보여야 한다.

4. 실습4 (새 페이지)

BLE Module AT-09의 TXD - Arduino UNO R3 digital PIN 4

BLE Module AT-09의 RXD - Arduino UNO R3 digital PIN 5

LED Module의 R - Arduino UNO R3 10

LED Module의 Y - Arduino UNO R3 9

LED Module의 G - Arduino UNO R3 8

Switch Module 1의 OUT - Arduino UNO R3 11

Switch Module 2의 OUT - Arduino UNO R3 12

BLE Module AT-09의 이름은 BT05-출석번호 (예, 150번인 경우 BT05-150)로 설정한다. (setup() 함수에서)

BLE Module AT-09와의 serial data rate는 19200 baud로 설정한다. (다른 프로그램 사용)

Serial monitor와의 serial data rate는 31250 baud로 설정한다.

- 프로그램이 실행되면 선택된 Led는 빨간색 Led가 되고 빨간색 Led를 켜다. Switch Module 2 (오른쪽 Switch Module)의 스위치가 눌릴 때마다 선택된 Led가 노란색 LED -> 초록색 LED -> 빨간색 LED -> ... 순서로 변경된다. 선택된 Led는 켜고 그 외의 Led는 끈다. Switch Module 1 (왼쪽 Switch Module)의 스위치가 눌릴 때마다 선택된 Led의 상태를 변화시킨다. 즉 현재 켜져 있으면 끄고 꺼져있으면 켜다. BLE App을 사용해서 연결하고 숫자 2를 전송하면 Switch Module 2 (오른쪽 Switch Module)의 스위치가 눌릴 때와 같은 동작을 수행한다. 숫자 1을 전송하면 Switch Module 1 (왼쪽 Switch Module)의 스위치가 눌릴 때와 같은 동작을 수행한다.

4.1 하드웨어 구성 결과

그림 20 실습4 하드웨어 구성 결과를 보여준다.



그림 20은 실습4 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.

그림 21 실습4 소프트웨어 작성 결과

```
8_21101395_Arduino_Report2_Lab4.ino
1 // 8-21101395-권준혁 Arduino21102 실습4
2 #include <SoftwareSerial.h>
3 #define SOFTWARE_SERIAL_RX_PIN 4
4 #define SOFTWARE_SERIAL_TX_PIN 5
5 #define SWITCH_2_OUT 12
6 #define SWITCH_1_OUT 11
7 #define LED_RED_PIN 10
8 #define LED_YELLOW_PIN 9
9 #define LED_GREEN_PIN 8
10 #define BLE_BAUD_RATE 9600
11
12 SoftwareSerial bleSerial(SOFTWARE_SERIAL_RX_PIN, SOFTWARE_SERIAL_TX_PIN);
13 bool isLedOn = false;
14 enum LedColor {RED, YELLOW, GREEN};
15 LedColor currentLed = RED;
16
17 void printCurrentLed(byte currentLed) {
18     String msg;
19     switch (currentLed) {
20         case RED:
21             msg = "Red Led";
22             digitalWrite(LED_RED_PIN, HIGH);
23             digitalWrite(LED_YELLOW_PIN, LOW);
24             digitalWrite(LED_GREEN_PIN, LOW);
25             break;
26         case YELLOW:
27             msg = "Yellow Led";
28             digitalWrite(LED_RED_PIN, LOW);
29             digitalWrite(LED_YELLOW_PIN, HIGH);
30             digitalWrite(LED_GREEN_PIN, LOW);
31             break;
32         case GREEN:
33             msg = "Green Led";
```

그림 22 실습4 소프트웨어 작성 결과

```
34     digitalWrite(LED_RED_PIN, LOW);
35     digitalWrite(LED_YELLOW_PIN, LOW);
36     digitalWrite(LED_GREEN_PIN, HIGH);
37     break;
38 }
39 Serial.println(msg);
40 bleSerial.println(msg);
41 }
42
43 void setup() {
44     pinMode(SWITCH_1_OUT, INPUT_PULLUP);
45     pinMode(SWITCH_2_OUT, INPUT_PULLUP);
46     pinMode(LED_RED_PIN, OUTPUT);
47     pinMode(LED_YELLOW_PIN, OUTPUT);
48     pinMode(LED_GREEN_PIN, OUTPUT);
49     Serial.begin(31250);
50     bleSerial.begin(BLE_BAUD_RATE);
51     Serial.println("8-21101395-권준혁> setup()");
52 }
53
54 void loop() {
55     if (digitalRead(SWITCH_1_OUT) == HIGH) {
56         Serial.println("8-21101395-권준혁> switch 1 pressed");
57         bleSerial.println("8-21101395-권준혁> switch 1 pressed");
58         if (isLedOn == false) {
59             isLedOn = true;
60             Serial.print("8-21101395-권준혁> Led On = ");
61             bleSerial.print("8-21101395-권준혁> Led On = ");
62             printCurrentLed(currentLed);
63         }
64         else {
65             isLedOn = false;
66             Serial.print("8-21101395-권준혁> Led Off = ");
```

그림 21, 22는 실습4 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호_학번_Arduino_Report2_Lab4
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함

그림 23 실습4 소프트웨어 작성 결과 계속

```
67 bleSerial.print("8-21101395-권준혁> Led Off = ");
68 printCurrentLed(currentLed);
69 }
70 switch(currentLed) {
71 case RED:
72     digitalWrite(LED_RED_PIN, !digitalRead(LED_RED_PIN));
73     break;
74 case YELLOW:
75     digitalWrite(LED_YELLOW_PIN, !digitalRead(LED_YELLOW_PIN));
76     break;
77 case GREEN:
78     digitalWrite(LED_GREEN_PIN, !digitalRead(LED_GREEN_PIN));
79     break;
80 }
81 delay(200);
82 }
83
84 if (digitalRead(SWITCH_2_OUT) == HIGH) {
85     isLedOn = true;
86     switch(currentLed) {
87 case RED:
88         currentLed = YELLOW;
89         break;
90 case YELLOW:
91         currentLed = GREEN;
92         break;
93 case GREEN:
94         currentLed = RED;
95         break;
96     }
97     Serial.println("8-21101395-권준혁> switch 2 pressed");
98     bleSerial.println("8-21101395-권준혁> switch 2 pressed");
99     Serial.print("8-21101395-권준혁> select = ");
```

그림 24 실습4 소프트웨어 작성 결과 계속

```
100 bleSerial.print("8-21101395-권준혁> select = ");
101 printCurrentLed(currentLed);
102 delay(200);
103 }
104
105 bleSerial.listen();
106 while (bleSerial.available() > 0) {
107     char cmd = bleSerial.read();
108     Serial.print(cmd);
109     if (cmd == '1') {
110         Serial.println("8-21101395-권준혁> BLE 1 pressed");
111         bleSerial.println("8-21101395-권준혁> BLE 1 pressed");
112         if (isLedOn == false) {
113             isLedOn = true;
114             Serial.print("8-21101395-권준혁> Led On = ");
115             bleSerial.print("8-21101395-권준혁> Led On = ");
116             printCurrentLed(currentLed);
117         }
118         else {
119             isLedOn = false;
120             Serial.print("8-21101395-권준혁> Led Off = ");
121             bleSerial.print("8-21101395-권준혁> Led Off = ");
122             printCurrentLed(currentLed);
123         }
124         switch(currentLed) {
125 case RED:
126             digitalWrite(LED_RED_PIN, !digitalRead(LED_RED_PIN));
127             break;
128 case YELLOW:
129             digitalWrite(LED_YELLOW_PIN, !digitalRead(LED_YELLOW_PIN));
130             break;
131 case GREEN:
132             digitalWrite(LED_GREEN_PIN, !digitalRead(LED_GREEN_PIN));
133             break;
134         }
135         delay(200);
136     }
137     else if (cmd == '2') {
138         isLedOn = true;
139         switch(currentLed) {
140 case RED:
141             currentLed = YELLOW;
142             break;
143 case YELLOW:
144             currentLed = GREEN;
145             break;
146 case GREEN:
147             currentLed = RED;
148             break;
149         }
150         Serial.println("8-21101395-권준혁> BLE 2 pressed");
151         bleSerial.println("8-21101395-권준혁> BLE 2 pressed");
152         Serial.print("8-21101395-권준혁> select = ");
153         bleSerial.print("8-21101395-권준혁> select = ");
154         printCurrentLed(currentLed);
155         delay(200);
156     }
157 }
158 }
159 }
```

그림 23, 24는 실습4 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호_학번_Arduino_Report2_Lab4
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함

그림 25 실습4 프로그램 실행 결과



그림 25는 실습4 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
 - 반드시 그림과 같은 순서로 출력되도록 동작시켜서 결과를 보인다.
- => setup() 함수에서 첫줄 출력
- BLE Module AT-09의 이름을 수정할 때 이름을 수정한 후에 시간이 조금 지나야 AT-09가 정상 동작을 수행할 수 있다.
- => 정상 동작 수행은 왼쪽 아래 Led가 깜빡거리는 것으로 확인한다.
- => 만일 왼쪽 아래 Led가 깜빡거리지 않으면 Arduino board를 reset 시켜 본다.
- => 위의 결과는 그렇게 한 후에 동작을 수행한 결과이다.

그림 26 실습4 프로그램 실행 결과

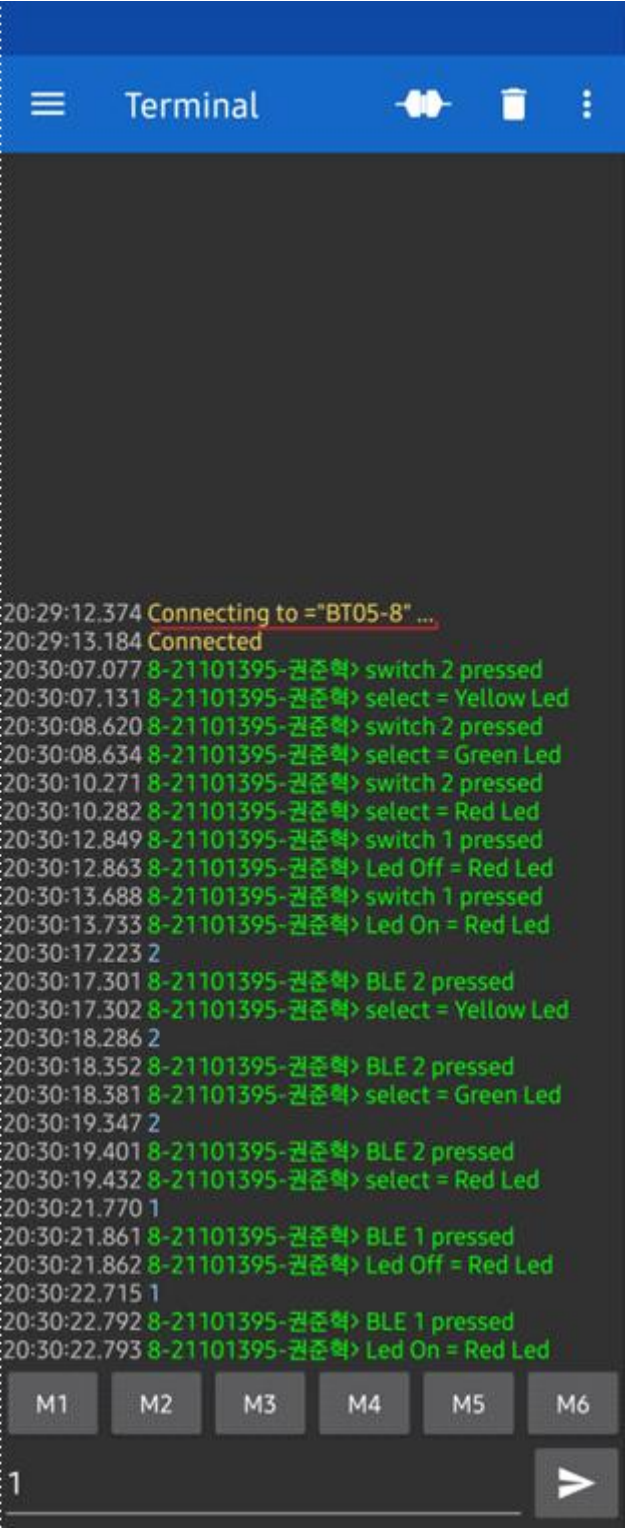


그림 26은 실습4 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- BLE Module AT-09의 이름 = BT05-8 (반드시 본인의 출석번호와 일치해야 한다.)
- 반드시 그림과 같은 순서로 출력되도록 동작시켜서 결과를 보인다.
- 한글이 일부 정상 출력되지 않는 것을 볼 수 있다. (이런 경우 그대로 캡처하면 된다.)

4.5 프로그램 실행 결과 3 (새 페이지)

그림 27은 실습4 프로그램 실행 결과



그림 27은 실습4 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- 스위치 2 (오른쪽 스위치)를 눌러서 노란색 LED가 선택된 결과를 보여준다.

4.6 프로그램 실행 결과 4

그림 28 실습4 프로그램 실행 결과



그림 28은 실습4 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 브레드보드에 출석번호를 직접 적는다.
- 스위치 2 (오른쪽 스위치)를 눌러서 초록색 LED가 선택된 결과를 보여준다.