

번호: 8번 학번: 21101395 이름: 권준혁 제출 일시: 5월 1일

- 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

- 빨간색 LED를 digital pin에 연결하고 같은 간격으로 on/off 시키는 동작을 반복하는 프로그램을 작성하고 실행한다.

1.1 하드웨어 구성 결과

그림 1 실습1 하드웨어 구성 결과

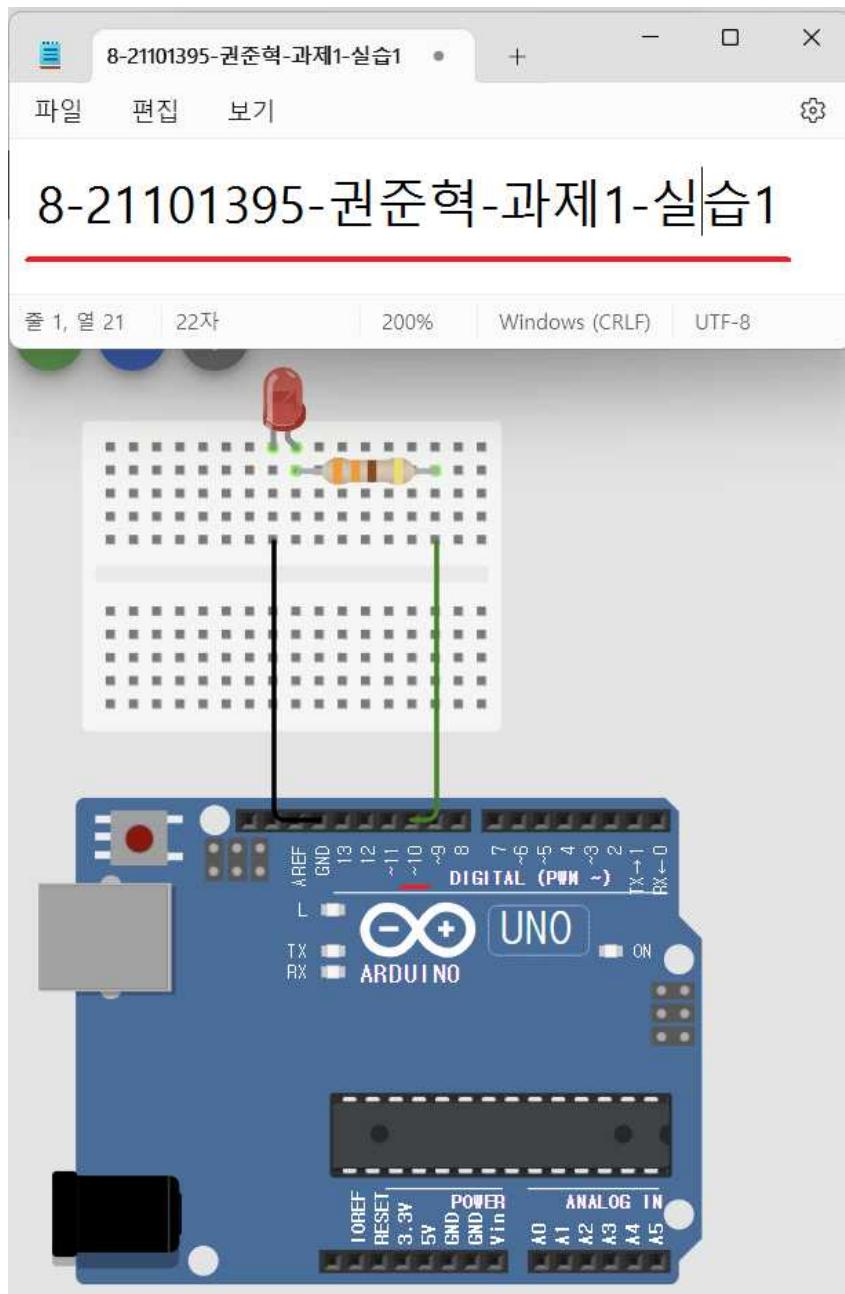


그림 1은 실습1 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 그림과 같이 구성한다. (LED의 색과 저항값을 일치시킨다.)
- pin number = 출석번호 끝자리 + 2 => pin number = 8 + 2 => pin number = 10

그림 2 실습1 소프트웨어 작성 결과

```
1 // 8-21101395-권준혁 과제1 실습1
2 int ledPin = 10;
3 void setup() {
4     pinMode(ledPin, OUTPUT);
5 }
6
7 void loop() {
8     digitalWrite(ledPin, HIGH);
9     delay(900);
10    digitalWrite(ledPin, LOW);
11    delay(900);
12 }
```

그림 2는 실습1 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함
- pin number = 출석번호 끝자리 + 2 => pin number = 8 + 2 => pin number = 10
- led on (off) 시간 = 출석번호 끝자리x100[ms] + 100[ms] (0번 = 100[ms], 9번 = 1000[ms])

그림 3 실습1 프로그램 실행 결과

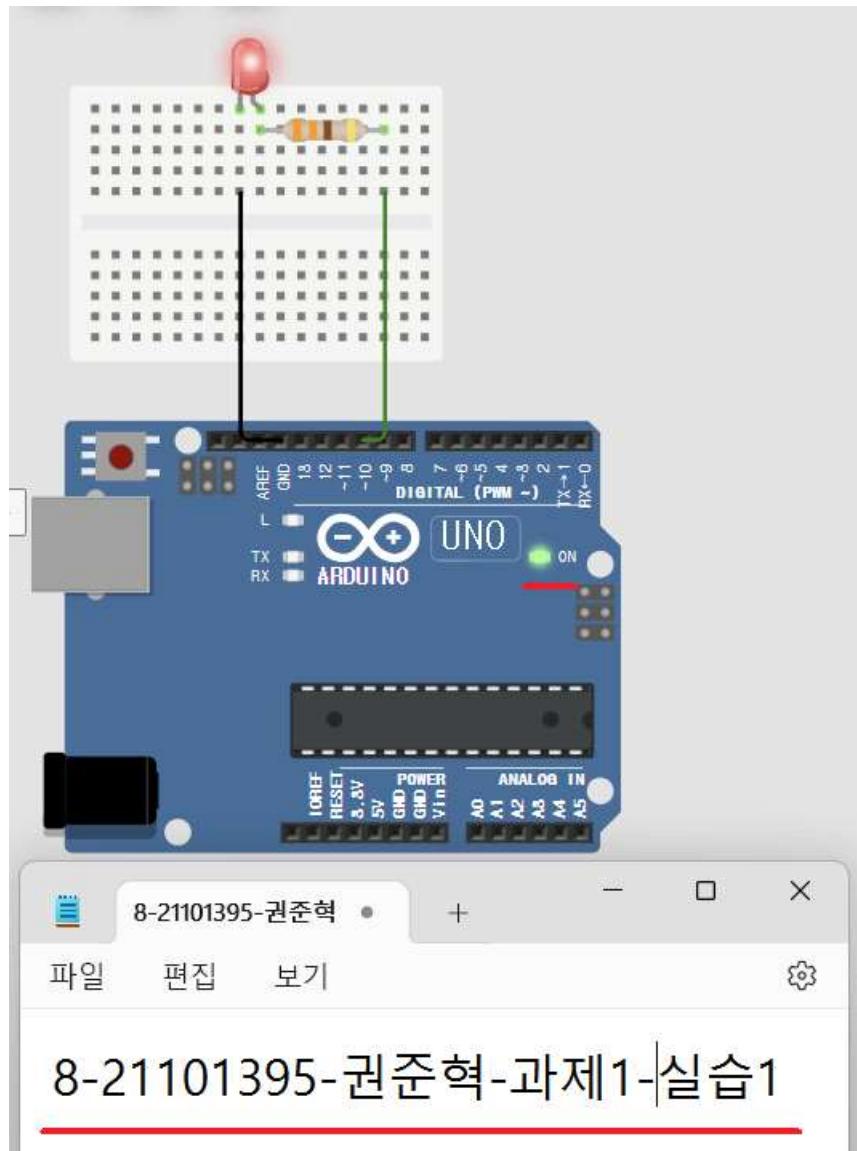


그림 3은 실습1 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.

2. 실습2 (새 페이지)

- 노란색 LED를 digital pin에 연결하고 Morse code로 S.O.S. 문자를 LED로 표시하는 프로그램을 작성하고 실행한다.

2.1 하드웨어 구성 결과

그림 4 실습2 하드웨어 구성 결과



그림 4는 실습2 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 그림과 같이 구성한다. (LED의 색과 저항값을 일치시킨다.)
- pin number = 출석번호 끝자리 + 2 => pin number = 8 + 2 => pin number = 10

2.2 소프트웨어 작성 결과 (새 페이지)

그림 5 실습2 소프트웨어 작성 결과

```
1 // 8-21101395-권준혁 과제1 실습2
2 int ledPin = 10;
3 void setup()
4 {
5     pinMode(ledPin, OUTPUT);
6 }
7
8 void loop()
9 {
10    for (int x=0; x<3; x++) {
11        digitalWrite(ledPin, HIGH); //sets the LED on
12        delay(150); //waits for 150ms
13        digitalWrite(ledPin, LOW); //sets the LED off
14        delay(150); //waits for 150ms
15    }
16    delay(100);
17    for (int x=0; x<3; x++) {
18        digitalWrite(ledPin, HIGH); //sets the LED on
19        delay(400); //waits for 150ms
20        digitalWrite(ledPin, LOW); //sets the LED off
21        delay(100); //waits for 150ms
22    }
23    delay(100);
24    for (int x=0; x<3; x++) {
25        digitalWrite(ledPin, HIGH); //sets the LED on
26        delay(150); //waits for 150ms
27        digitalWrite(ledPin, LOW); //sets the LED off
28        delay(100); //waits for 150ms
29    }
30    delay(5000);
31 }
```

그림 5는 실습2 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함
- pin number = 출석번호 끝자리 + 2 => pin number = 8 + 2 => pin number = 10

2.3 프로그램 실행 결과 (새 페이지)

그림 6 실습2 프로그램 실행 결과

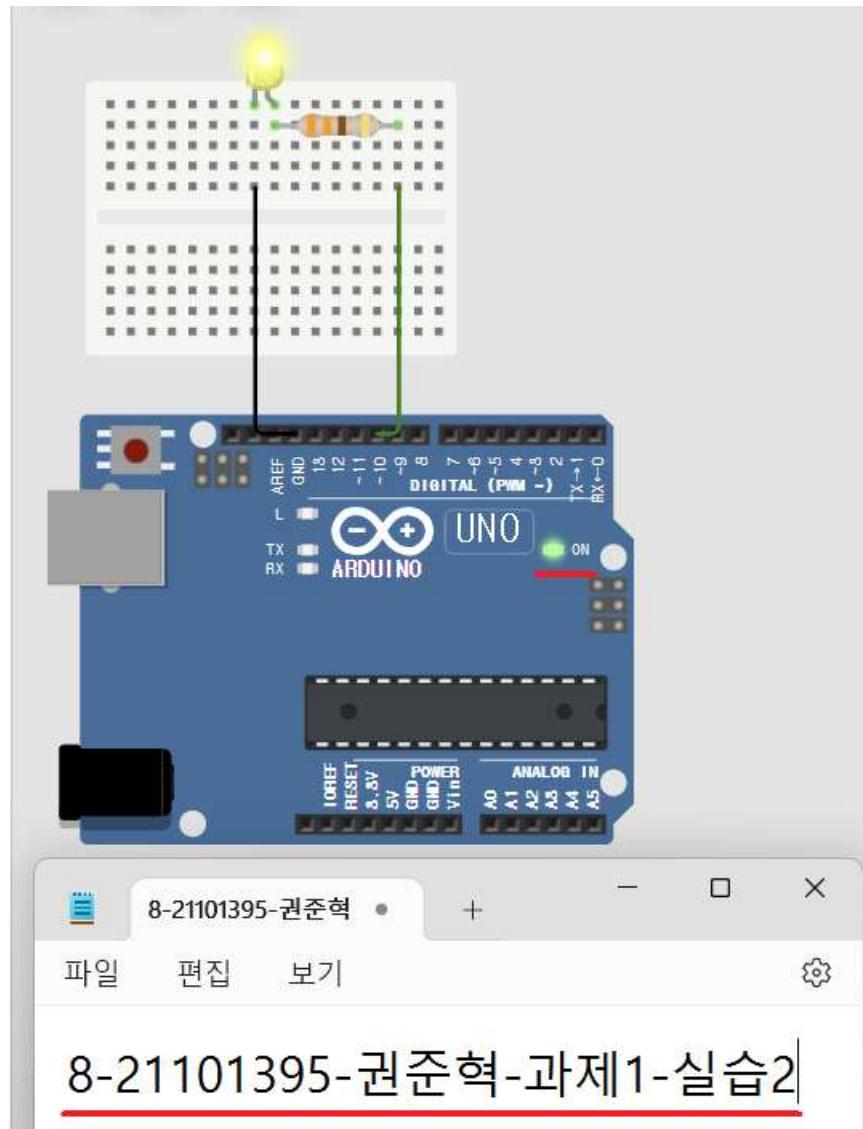
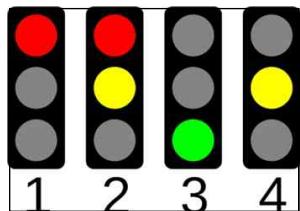


그림 6은 실습2 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.

3. 실습3 (새 페이지)

- 3개의 LED를 이용해서 그림과 같이 4가지 상태를 가지는 교통 신호등 시스템을 구성해 본다.



- 1번 상태 = 빨간색만 켜진다.
- 2번 상태 = 빨간색과 노란색만 켜진다.
- 3번 상태 = 초록색만 켜진다.
- 4번 상태 = 노란색만 켜진다.
- 1번 상태 → 2번 상태 → 3번 상태 → 4번 상태 → 1번 상태 → ... 와 같은 순서로 반복된다.

3.1 하드웨어 구성 결과

그림 8 실습3 하드웨어 구성 결과

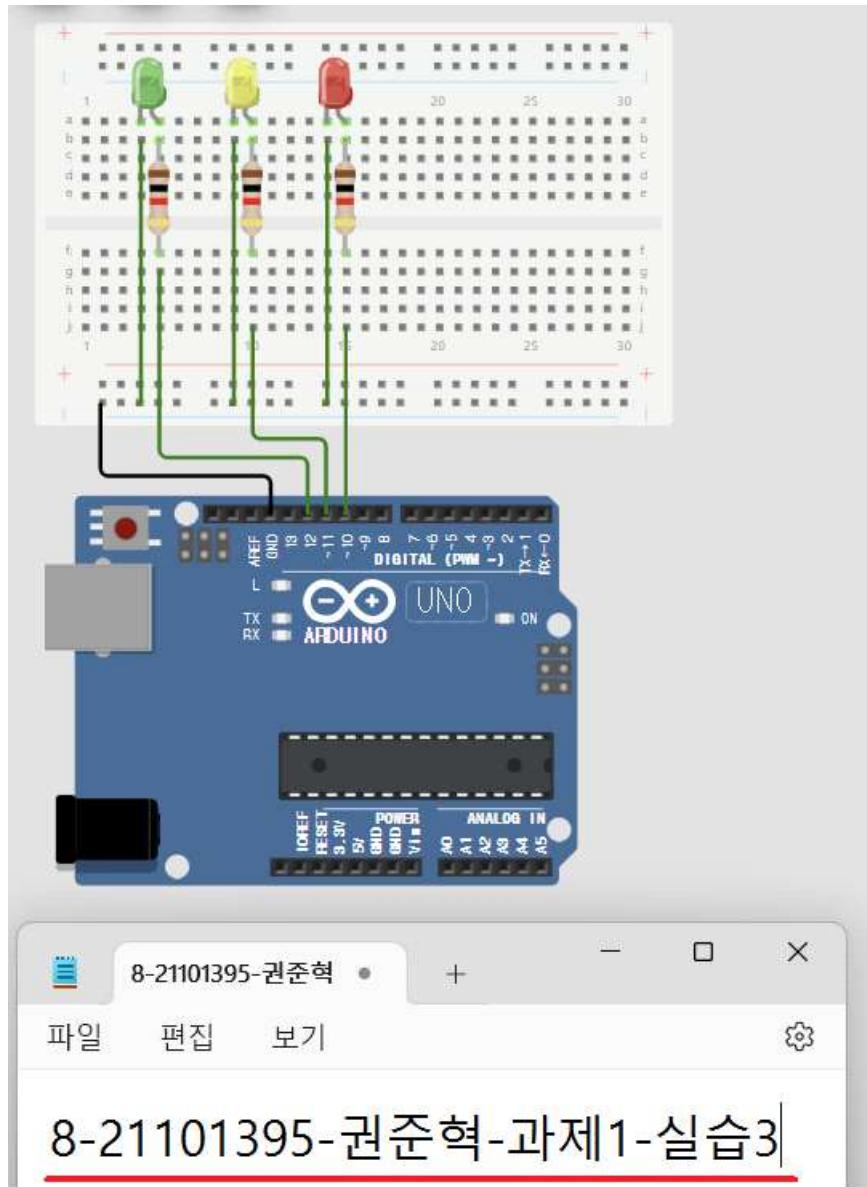


그림 8은 실습3 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 그림과 같이 구성한다. (LED의 색과 순서 그리고 저항값을 일치시킨다.)
- pin number = 출석번호 끝자리 + 2 => pin number = 8 + 2 => pin number = 10부터 시작
=> red led = 10, yellow led = 11, green led = 12 (출석번호에 따라서 수정한다.)

3.2 소프트웨어 작성 결과

그림 9 실습3 소프트웨어 작성 결과

```
1 // 8-21101395-권준혁 과제1 실습3
2 int ledDelay = 5000;
3 int redPin = 10;
4 int yellowPin = 11;
5 int greenPin = 12;
6
7 void setup() {
8     pinMode(redPin, OUTPUT);
9     pinMode(yellowPin, OUTPUT);
10    pinMode(greenPin, OUTPUT);
11 }
12
13 void loop() {
14     digitalWrite(redPin, HIGH); // turn the red light on
15     delay(ledDelay); // wait 5 seconds
16     digitalWrite(yellowPin, HIGH); // turn on yellow
17     delay(2000); // wait 2 seconds
18     digitalWrite(greenPin, HIGH); // turn green on
19     digitalWrite(redPin, LOW); // turn red off
20     digitalWrite(yellowPin, LOW); // turn yellow off
21     delay(ledDelay); // wait 5 seconds
22     digitalWrite(yellowPin, HIGH); // turn yellow on
23     digitalWrite(greenPin, LOW); // turn green off
24     delay(2000); // wait 2 seconds
25     digitalWrite(yellowPin, LOW); // turn yellow off
26 }
```

그림 9는 실습3 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함
- pin number = 출석번호 끝자리 + 2 => pin number = 8 + 2 => pin number = 10부터 시작
=> red led = 10, yellow led = 11, green led = 12 (출석번호에 따라서 수정한다.)

3.3 프로그램 실행 결과

그림 10 실습3 프로그램 실행 결과

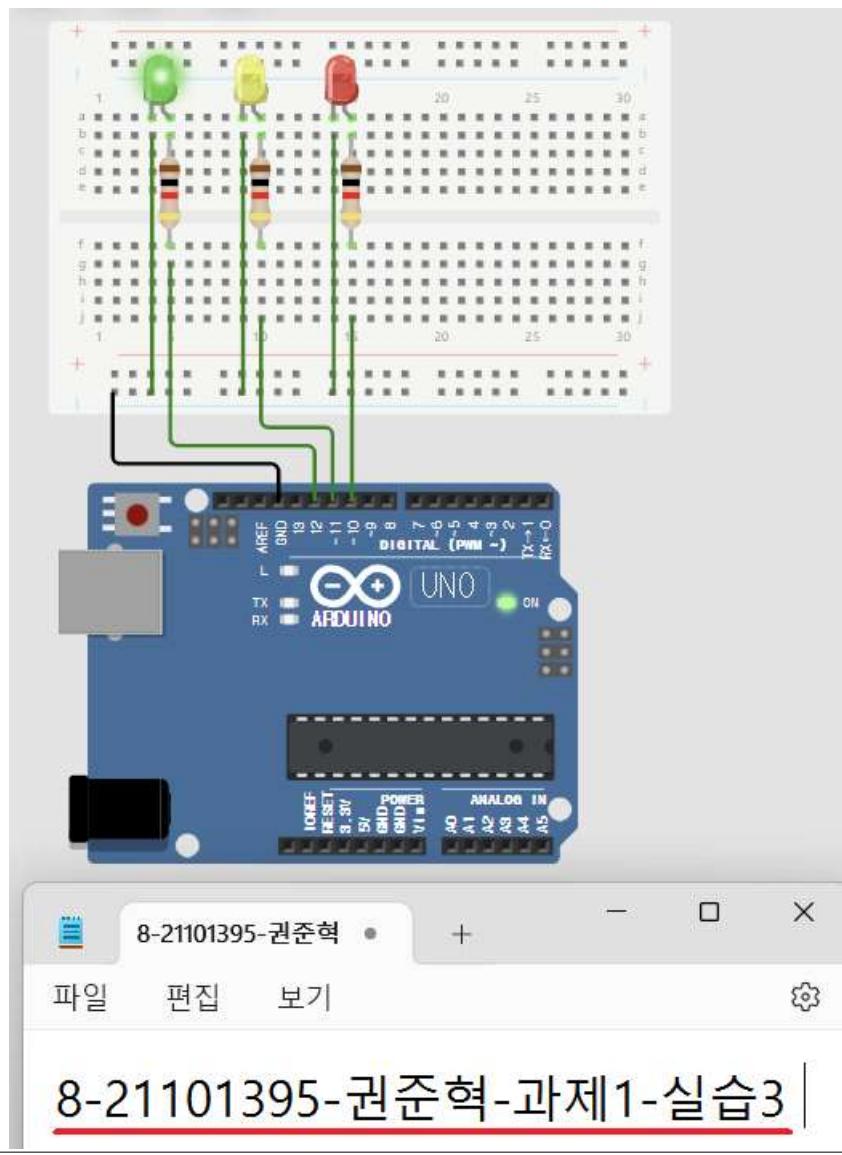


그림 10은 실습3 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.

4. 실습4 (새 페이지)

그림과 같은 교통 신호등 시스템을 구성해 본다. 버튼 왼쪽의 3개의 LED는 자동차용 신호등을 나타내고 오른쪽 2개의 LED는 보행자용 신호등을 나타낸다. 버튼이 눌리기 전에는 자동차용 신호등은 초록색 LED가 켜지고 보행자용 신호등은 빨간색 LED가 켜진다. 버튼을 누르면 자동차용 신호등은 노란색 LED가 켜진다. 일정 시간 후에 자동차용 신호등은 빨간색 LED가 켜지고 보행자용 신호등은 초록색 LED가 켜진다. 일정 시간 후에 보행자용 신호등의 초록색 LED가 깜빡거리고 빨간색 LED가 켜진다. 이때 자동차용 신호등은 노란색 LED가 켜진다. 일정 시간 후에 자동차용 신호등의 초록색 LED가 켜진다.

4.1 하드웨어 구성 결과

그림 11 실습4 하드웨어 구성 결과

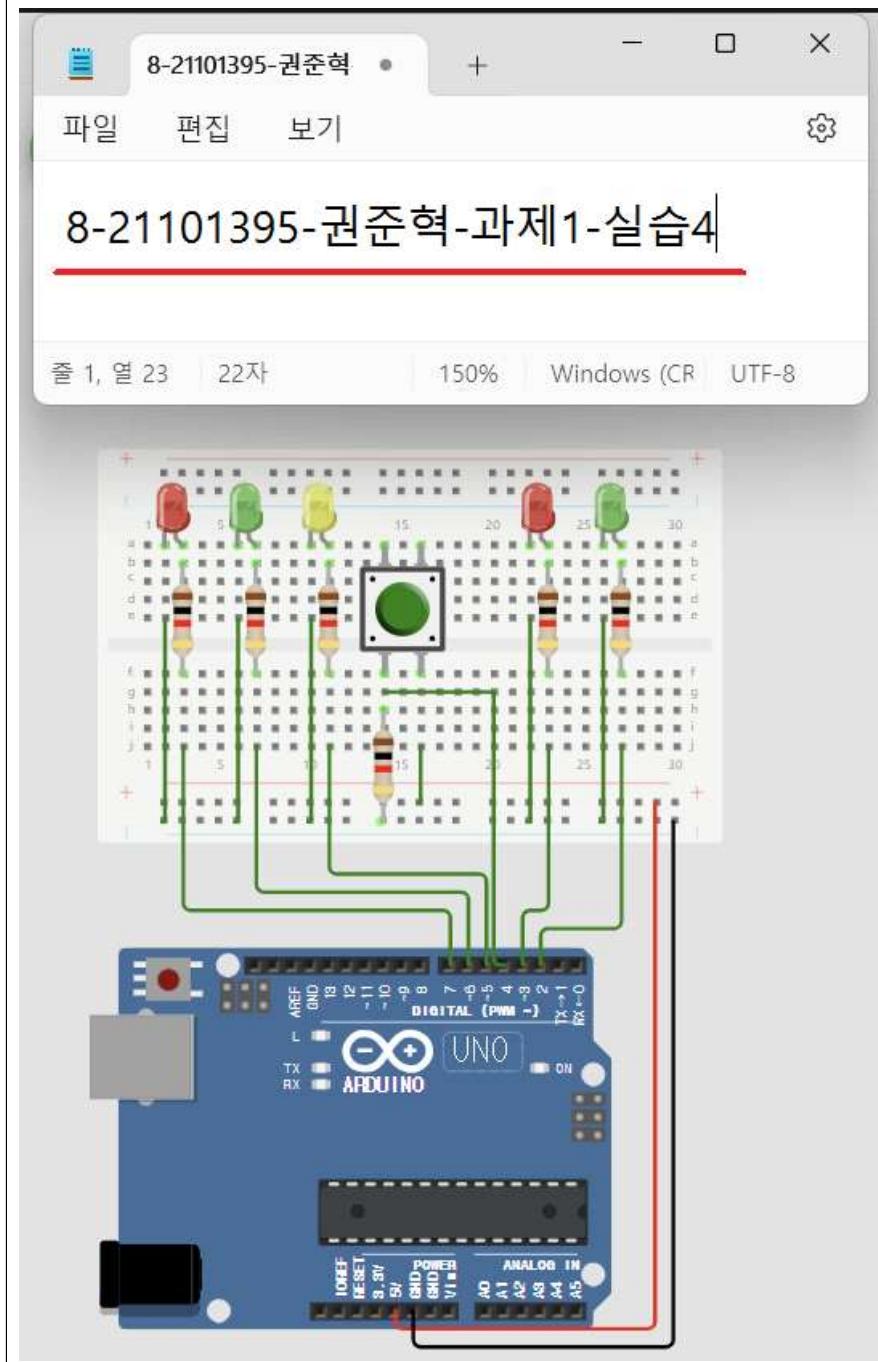


그림 11은 실습4 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 그림과 같이 구성한다. (LED의 색과 순서 그리고 저항값을 일치시킨다.)

4.2 소프트웨어 작성 결과

그림 12 실습4 소프트웨어 작성 결과

```
1 // 8-21101395-권준혁 과제1 실습4
2 int carRed = 7; // assign the car lights
3 int carGreen = 6;
4 int carYellow = 5;
5 int pedRed = 3; // assign the pedestrian lights
6 int pedGreen = 2;
7 int button = 4; //button pin
8 int crossTime = 5000; // time allowed to cross
9 unsigned long changeTime = 0; // time last pedestrian cycle completed
10
11 void changeLights() {
12     digitalWrite(carGreen, LOW); // green off
13     digitalWrite(carYellow, HIGH); // yellow on
14     delay(2000); // wait 2 seconds
15     digitalWrite(carYellow, LOW); // yellow off
16     digitalWrite(carRed, HIGH); // red on
17     delay(1000); // wait 1 second till its safe
18     digitalWrite(pedRed, LOW); // ped red off
19     digitalWrite(pedGreen, HIGH); // ped green on
20     delay(crossTime); // wait for preset time period
21     // flash the ped green
22     for (int x=0; x<10; x++) {
23         digitalWrite(pedGreen, HIGH);
24         delay(250);
25         digitalWrite(pedGreen, LOW);
26         delay(250);
27     }
28     // turn ped red on
29     digitalWrite(pedRed, HIGH);
30     delay(500);
31     digitalWrite(carYellow, HIGH); // yellow on
32     digitalWrite(carRed, LOW); // red off
33     delay(1000);
34     digitalWrite(carGreen, HIGH);
35     digitalWrite(carYellow, LOW); // yellow off
36     // record the time since last change of lights
37     changeTime = millis();
38     // then return to the main program loop
39 }
40 }
```

그림 12 실습4 소프트웨어 작성 결과 계속

```
41 ↘ void setup() {
42     pinMode(carRed, OUTPUT);
43     pinMode(carYellow, OUTPUT);
44     pinMode(carGreen, OUTPUT);
45     pinMode(pedRed, OUTPUT);
46     pinMode(pedGreen, OUTPUT);
47     pinMode(button, INPUT); // button on pin 2
48     // turn on the green light
49     digitalWrite(carGreen, HIGH);
50     digitalWrite(pedRed, HIGH);
51 }
52
53 ↘ void loop() {
54     int state = digitalRead(button);
55     // check if button is pressed and it is over 5 seconds since last button press
56     if (state == HIGH && (millis() - changeTime) > 5000 ) {
57     ↗ // Call the function to change the lights
58     | changeLights();
59     }
60 }
```

그림 12는 실습4 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함
- 프로그램이 한 페이지에 포함되지 않으면 두 개의 테이블로 나누어서 포함시킨다.

4.3 프로그램 실행 결과

그림 13 실습4 프로그램 실행 결과

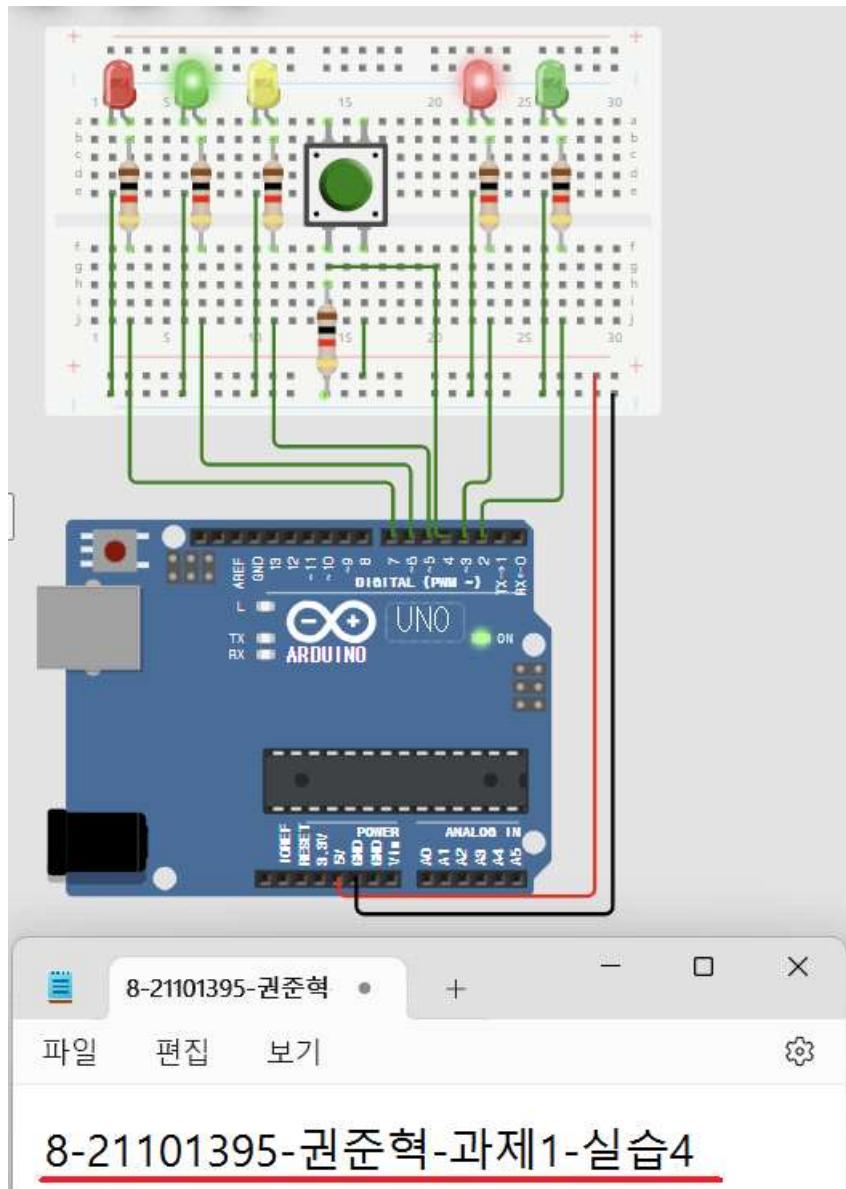


그림 13은 실습4 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
 - 반드시 그림과 같은 번호-학번-이름을 포함한다.

5. 실습5 (새 페이지)

그림과 같이 6개의 LED로 회로를 구성한다. 맨 오른쪽의 빨간색 LED부터 시작해서 왼쪽으로 차례로 LED가 일정 시간 동안 켜졌다 꺼진다. 맨 왼쪽 초록색 LED까지 도달하면 오른쪽으로 차례로 LED가 일정 시간 동안 켜졌다 꺼진다. 이 과정을 계속 반복한다.

5.1 하드웨어 구성 결과

그림 14 실습5 하드웨어 구성 결과

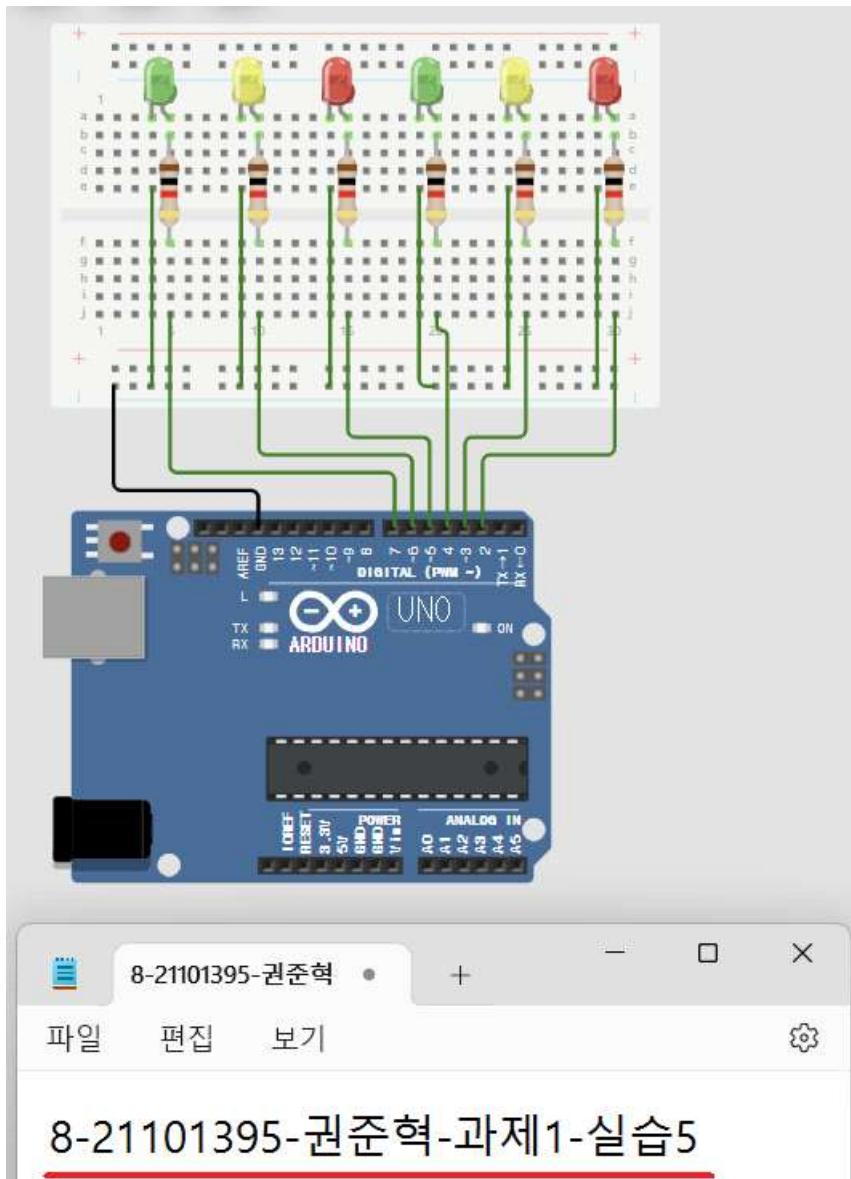


그림 14는 실습5 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 그림과 같이 구성한다. (LED의 색과 순서 그리고 저항값을 일치시킨다.)

5.2 소프트웨어 작성 결과

그림 15 실습5 소프트웨어 작성 결과

```
1 // 8-21101395-권준혁 과제1 실습5
2 byte ledPin[] = {2,3,4,5,6,7}; // Create array for LED pins
3 byte ledPinNumber = 6; // total number of LED
4 int ledDelay = 1800; // delay between changes
5 int direction = 1;
6 int currentLED = 0;
7 unsigned long changeTime;
8 void changeLED();
9
10 void setup() {
11     for (int x=0; x<ledPinNumber; x++) { // set all pins to output
12         pinMode(ledPin[x], OUTPUT);
13     }
14     changeTime = millis();
15 }
16
17 void loop() {
18     if ((millis() - changeTime) > ledDelay) { // if it has been ledDelay ms since last change
19         changeLED();
20         changeTime = millis();
21     }
22 }
23
24 void changeLED() {
25     for (int x=0; x<ledPinNumber ; x++) { // turn off all LED's
26         digitalWrite(ledPin[x], LOW);
27     }
28     digitalWrite(ledPin[currentLED], HIGH); // turn on the current LED
29     currentLED += direction; // increment by the direction value
30     // change direction if we reach the end
31     if (currentLED == (ledPinNumber-1)) {direction = -1;}
32     if (currentLED == 0) {direction = 1;}
```

그림 15는 실습5 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함
- LED가 켜지는 시간 = 출석번호 끝자리*200[ms] + 200[ms]
=> 끝자리가 0번 = 0 + 200[ms] => LED가 켜지는 시간 = 200[ms]
=> 끝자리가 9번 = 9*200[ms] + 200[ms] => LED가 켜지는 시간 = 2000[ms]

5.3 프로그램 실행 결과

그림 16 실습5 프로그램 실행 결과

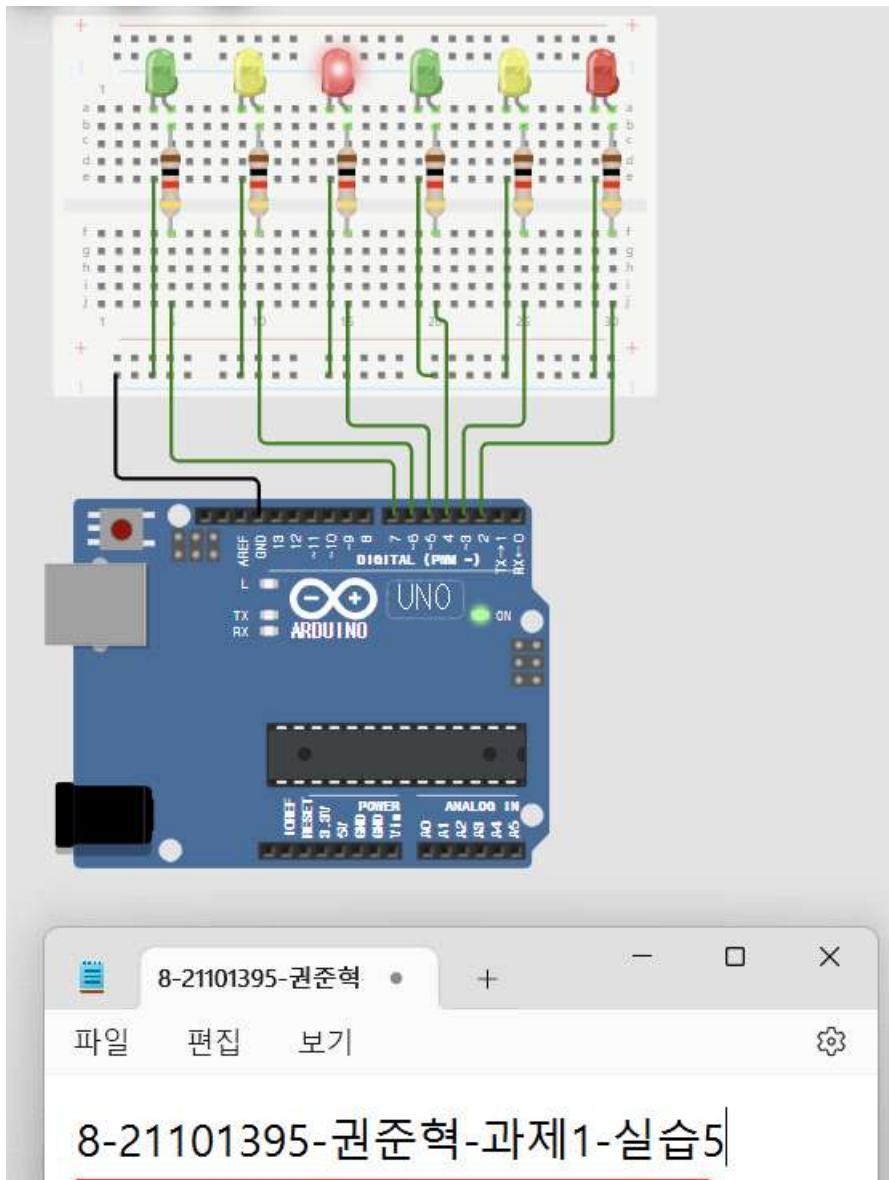


그림 16은 실습5 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.

6. 실습6 (새 페이지)

그림과 같이 6개의 LED로 회로를 구성한다. 맨 오른쪽의 빨간색 LED부터 시작해서 왼쪽으로 차례로 LED가 일정 시간 동안 켜졌다 꺼진다. 맨 왼쪽 초록색 LED까지 도달하면 오른쪽으로 차례로 LED가 일정 시간 동안 켜졌다 꺼진다. 이 과정을 계속 반복한다. 이때 LED가 켜지는 시간 간격은 potentiometer의 값을 읽어서 설정한다. Potentiometer의 값을 analogRead() 함수를 사용해서 읽으면 0에서 1023 사이의 값이 반환되는데 그 값을 실습5의 ledDelay 변수의 값으로 설정한다.

6.1 하드웨어 구성 결과

그림 17 실습6 하드웨어 구성 결과

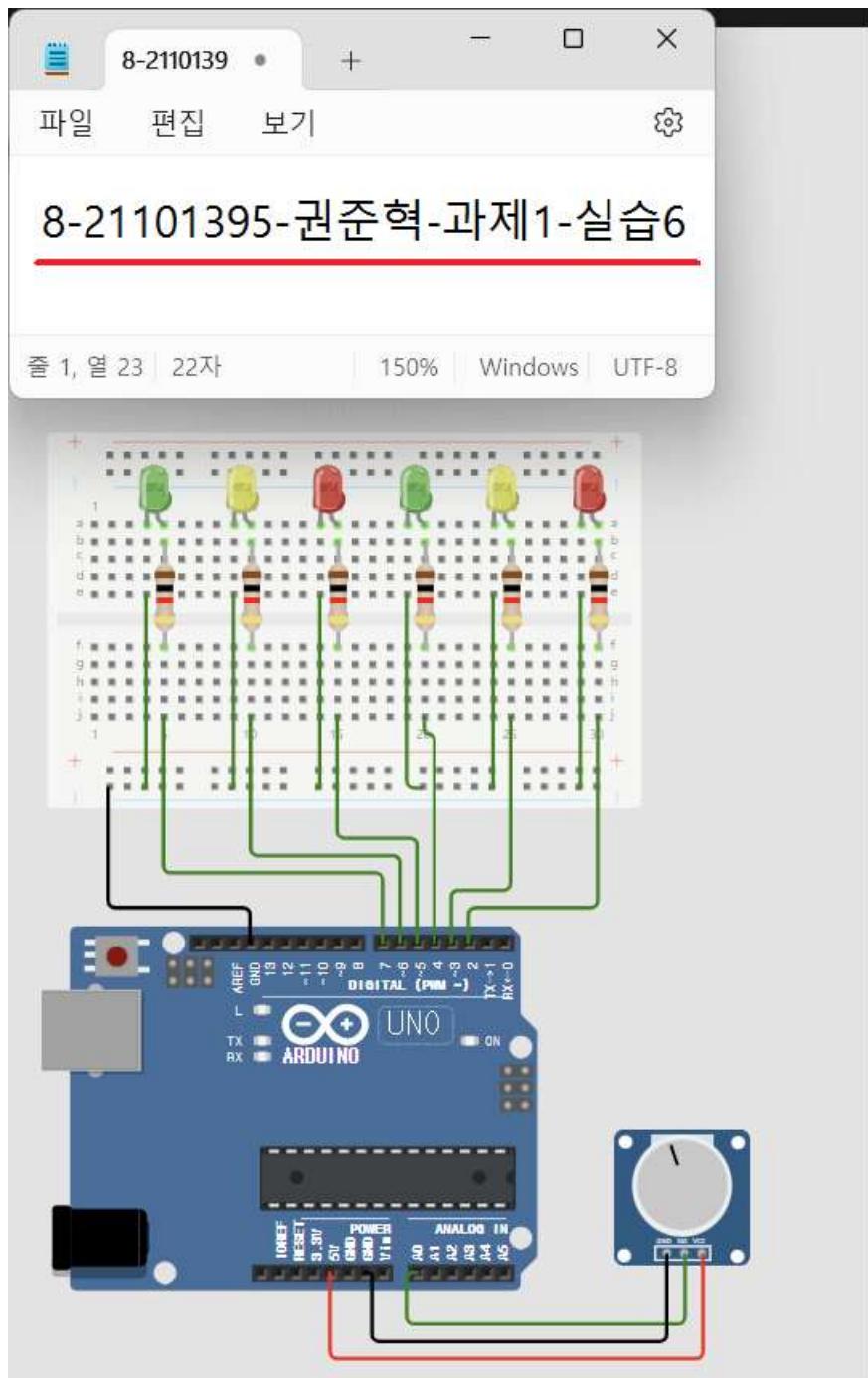


그림 17은 실습6 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 그림과 같이 구성한다. (LED의 색과 순서 그리고 저항값을 일치시킨다.)

6.2 소프트웨어 작성 결과

그림 18 실습6 소프트웨어 작성 결과

```
1 // 8-21101395-권준혁 과제1 실습6
2 byte ledPin[] = {2,3,4,5,6,7}; // Create array for LED pins
3 byte ledPinNumber = 6; // total number of LED
4 byte potentiometerPin = A0; // potentiometer pin number
5 int ledDelay; // delay between changes
6 int direction = 1;
7 int currentLED = 0;
8 unsigned long changeTime;
9 void changeLED();
10
11 void setup() {
12     for (int x=0; x<ledPinNumber; x++) { // set all pins to output
13         pinMode(ledPin[x], OUTPUT);
14     }
15     pinMode(A0, INPUT);
16     changeTime = millis();
17 }
18
19 void loop() {
20     ledDelay = analogRead(potentiometerPin); // read the value from the potentiometer
21     if ((millis() - changeTime) > ledDelay) { // if it has been ledDelay ms since last change
22         changeLED();
23         changeTime = millis();
24     }
25 }
26
27 void changeLED() {
28     for (int x=0; x<ledPinNumber; x++) { // turn off all LED's
29         digitalWrite(ledPin[x], LOW);
30     }
31     digitalWrite(ledPin[currentLED], HIGH); // turn on the current LED
32     currentLED += direction; // increment by the direction value
33     // change direction if we reach the end
34     if (currentLED == (ledPinNumber-1)) {direction = -1;}
35     if (currentLED == 0) {direction = 1;}
36 }
```

그림 18은 실습6 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함
- LED가 켜지는 시간 = potentiometer에서 읽은 값[ms] (0[ms]에서 1023[ms]까지의 값을 가질 수 있다.)

6.3 프로그램 실행 결과 ---

그림 19 실습6 프로그램 실행 결과

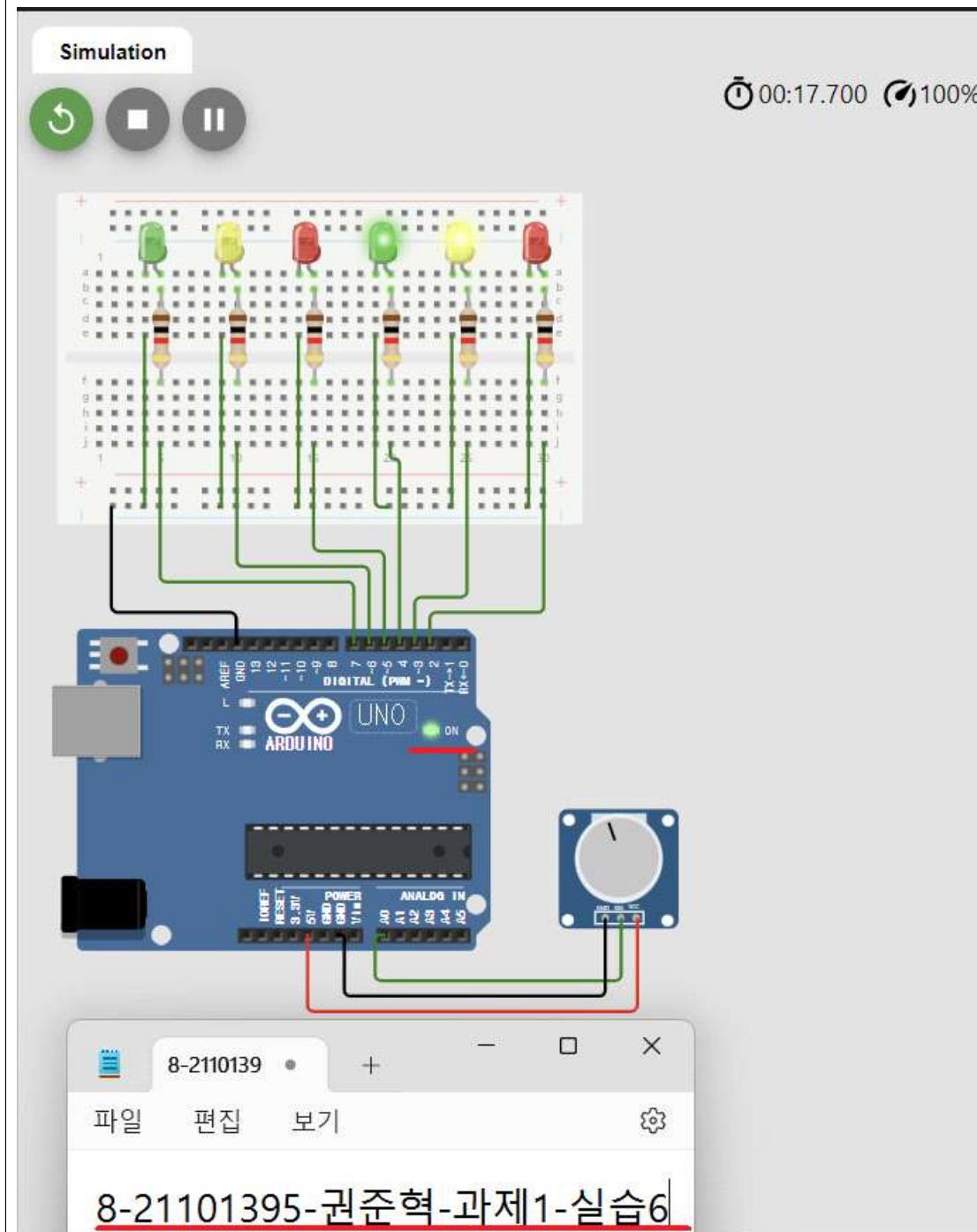


그림 19는 실습6 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.

7. 실습7 (새 페이지)

그림과 같이 회로를 구성한다. PWM 파형을 사용해서 빨간색 LED의 밝기를 조절한다.

7.1 하드웨어 구성 결과

그림 20 실습7 하드웨어 구성 결과

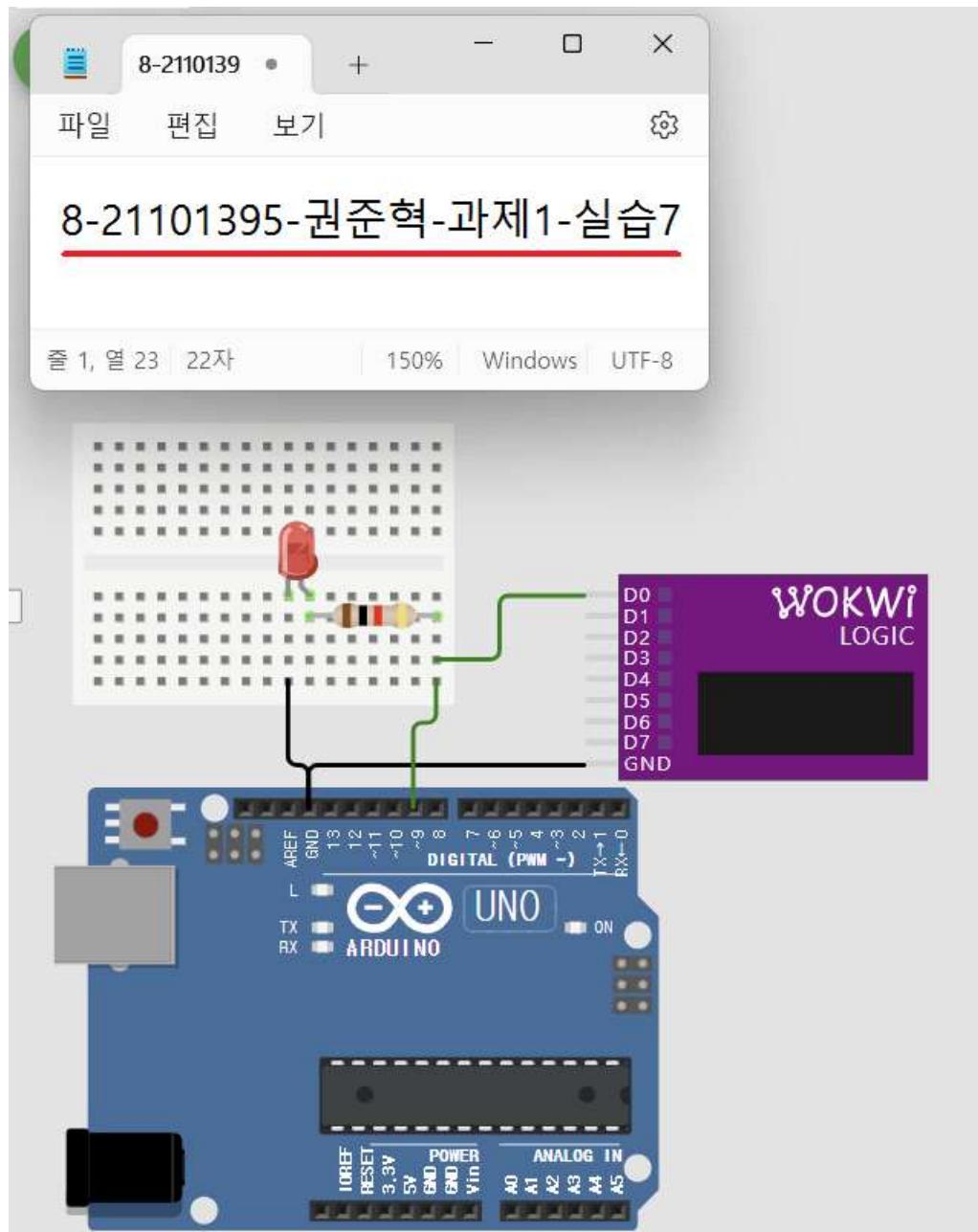


그림 20은 실습7 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 그림과 같이 구성한다. (LED의 색과 저항값을 일치시킨다.)

7.2 소프트웨어 작성 결과

그림 21 실습7 소프트웨어 작성 결과

```
1 // 8-21101395-권준혁 과제1 실습7
2 int ledPin = 9;
3 float sinVal;
4 int ledVal;
5
6 void setup() {
7     pinMode(ledPin, OUTPUT);
8 }
9
10 void loop() {
11     for (int x=0; x<100; x=x+30) {
12         // convert degrees to radians then obtain sin value
13         sinVal = (sin(x*(3.1412/180)));
14         ledVal = int(sinVal*255);
15         analogWrite(ledPin, ledVal);
16         delay(5);
17     }
18 }
```

그림 21은 실습7 소프트웨어 작성 결과를 보여준다.

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

7.3 프로그램 실행 결과 1

그림 22 실습7 프로그램 실행 결과

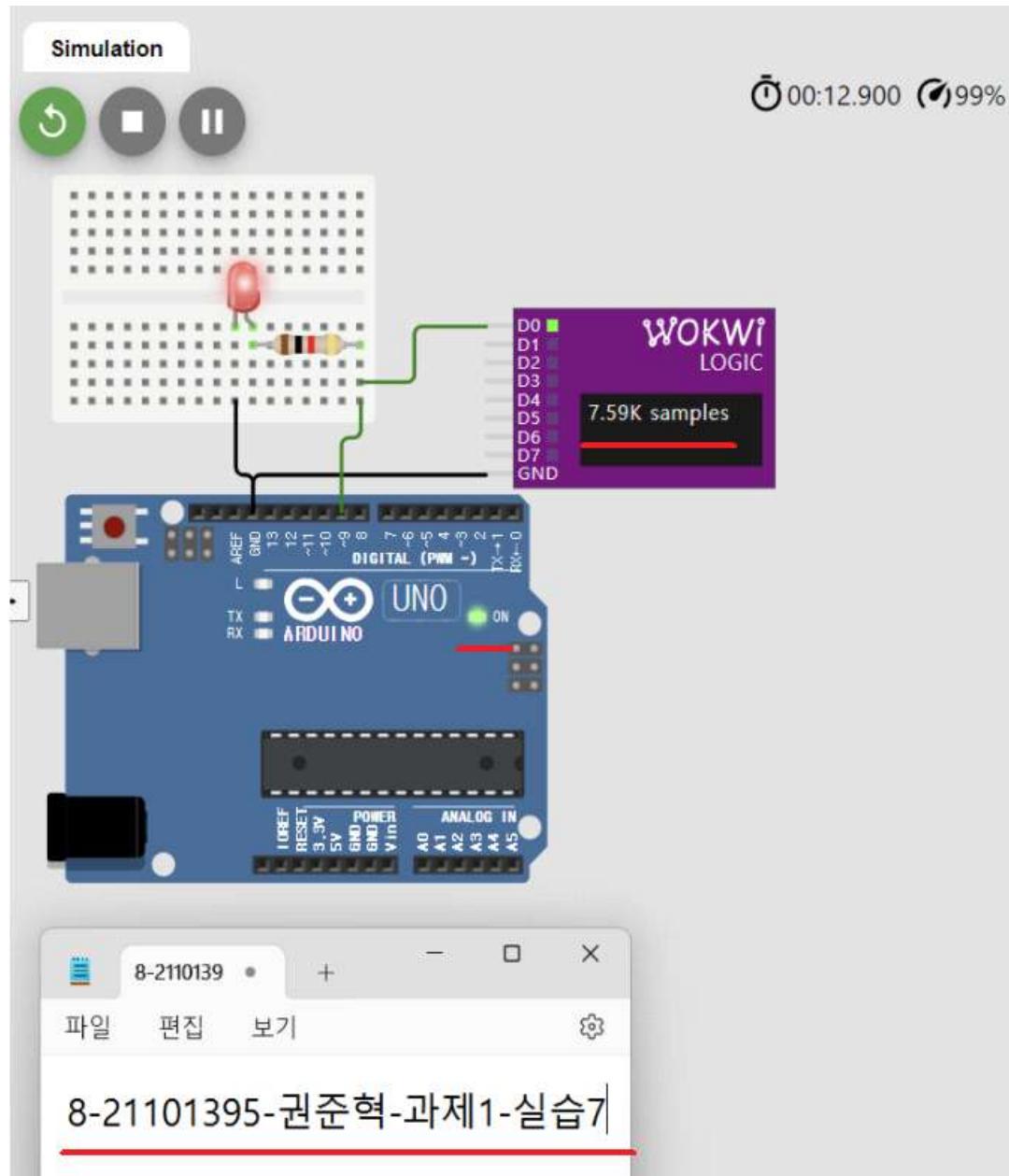


그림 22는 실습7 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.

7.4 프로그램 실행 결과 2

그림 23 실습7 프로그램 실행 결과

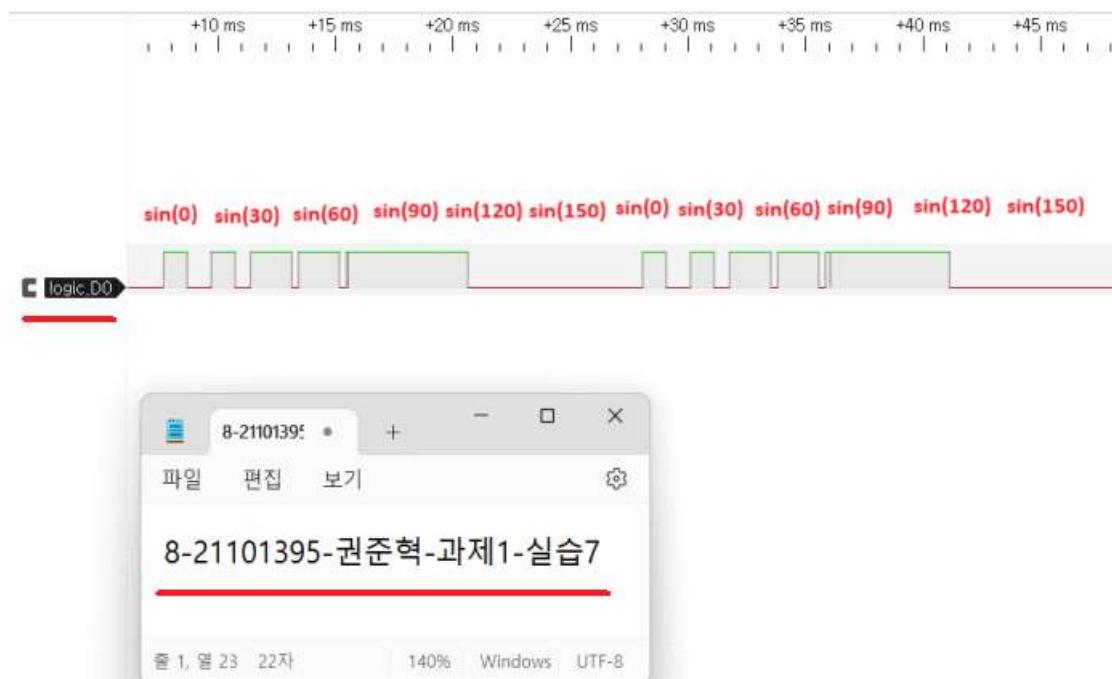


그림 23은 실습7 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.