

번호: 8번 학번: 21101395 이름: 권준혁 제출 일시: 5월 25일

- 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

Switch Module 2의 OUT - NodeMCU-32S GPIO12 pin (board에는 G12로 적혀있다.)

Switch Module 1의 OUT - NodeMCU-32S GPIO14 pin (board에는 G14로 적혀있다.)

LED Module의 R - NodeMCU-32S GPIO27 pin (board에는 G27로 적혀있다.)

LED Module의 Y - NodeMCU-32S GPIO26 pin (board에는 G26로 적혀있다.)

LED Module의 G - NodeMCU-32S GPIO25 pin (board에는 G25로 적혀있다.)

PIR Motion Sensor HC-SR501의 Output - NodeMCU-32S GPIO32 pin (board에는 G32로 적혀있다.)

serial data rate는 115200 baud로 설정한다.

- 3가지 mode로 동작한다.
- Mode 0이 시작되면 빨간색 led만 켜진다. Mode 0에서는 움직임이 감지되어도 led를 변화시키지 않는다.
- Mode 1이 시작되면 노란색 led만 켜진다. 움직임이 감지되면 노란색 led는 꺼지고 빨간색 led가 0.2초 간격으로 on/off 된다. 움직임 감지가 끝나면 빨간색 led가 꺼지고 mode 1의 처음 상태로 돌아가서 다시 노란색 led가 켜진다.
- Mode 2가 시작되면 초록색 led만 켜진다. 움직임이 감지되면 초록색 led는 꺼지고 0.2초 간격으로 빨간색 led는 on/off 노란색 led는 off/on 된다. 움직임 감지가 끝나도 계속 led들을 on/off 시킨다. 스위치 2 (오른쪽 스위치)를 누르면 led들의 on/off가 중지되고 mode 2의 처음 상태로 돌아가서 초록색 led만 켜진다.
- Mode 선택은 스위치 1 (왼쪽)을 한 번 누를 때마다 mode가 변하는 방식으로 수행한다.
- Mode 변경은 움직임 감지가 끝난 상태에서만 실행된다. 예로, mode = 0에서 motion detected가 되고 끝나기 전에 mode가 변경되면 mode = 1에서 motion ended가 될 수 있다. 이렇게 되지 않아야 한다는 의미이다.
- Mode 2에서 움직임 감지가 된 후에 아직 switch 2 (오른쪽 스위치)를 누르지 않은 상태면 mode 변경이 될 수 없다.
- 프로그램이 처음 시작되면 mode 0으로 동작한다.
- Mode 1, 2에서 led를 on/off 시키는 것은 timer interrupt를 사용한다.
- 스위치 1, 2의 입력을 위해서 각각 external interrupt를 사용한다.
- PIR Motion Sensor HC-SR501의 출력을 판단하는 것은 external interrupt를 사용한다.
- PIR Motion Sensor HC-SR501의 delay와 sensitivity는 최소로 설정한다.
- delay() 함수는 사용하지 않는다.
- interrupt service routine은 최대로 짧게 작성하고 그 안에서 serial monitor와의 입출력을 수행하지 않는다.
- 뒤의 결과 그림의 예와 같이 serial monitor에 메시지를 Serial.printf()를 사용해서 출력한다.

## 1.1 하드웨어 구성 결과 (새 페이지)

그림 1 실습1 하드웨어 구성 결과

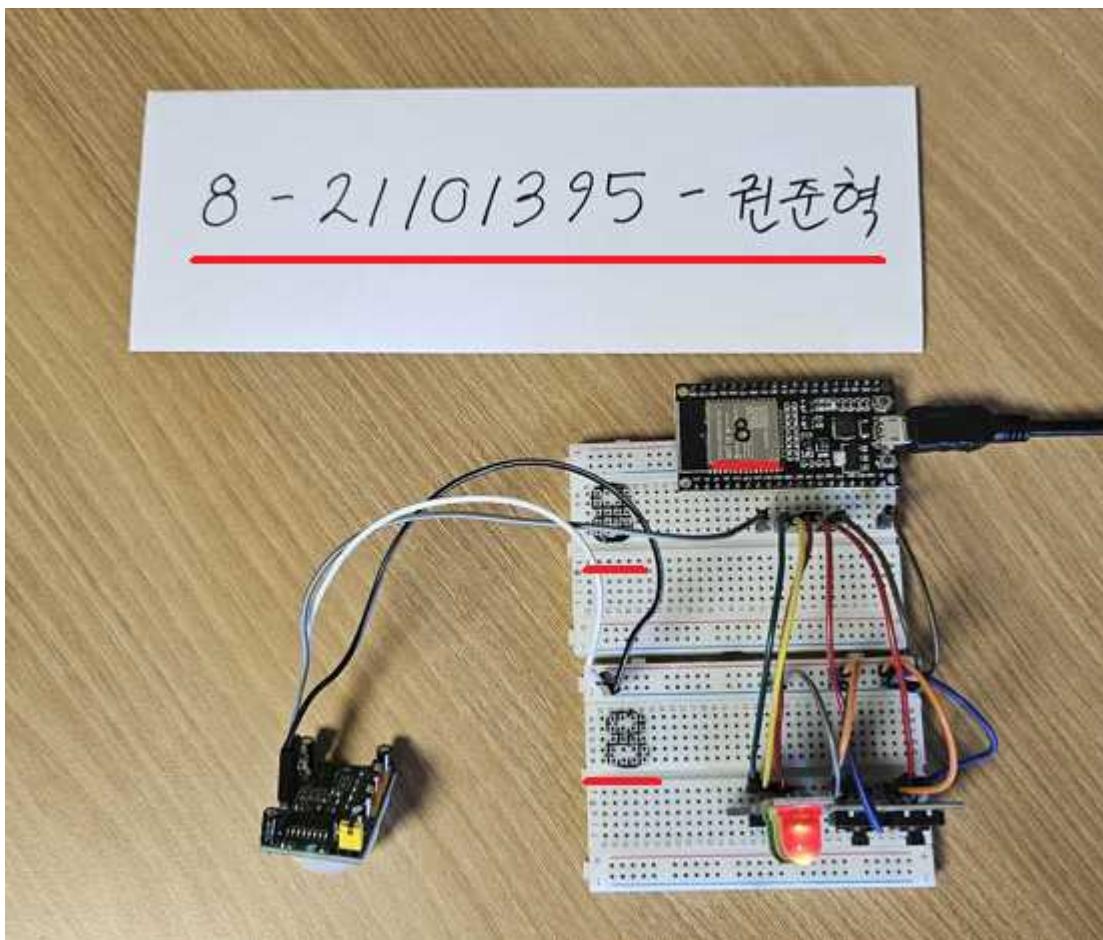


그림 1은 실습1 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 그림과 같이 브레드보드와 ESP32 NodeMCU-32S 보드에 출석번호를 직접 적는다.  
=> 출석번호를 확실하게 확인할 수 있어야 한다.
- 요구되는 표시가 하나라도 없으면 0점 처리한다.

## 그림 2 실습1 소프트웨어 작성 결과

```

8_21101395_ESP32_Report1_Lab1.ino
1 // 8-21101395-권준혁 ESP32과제1 실습1
2 // 각종 상수 정의
3 #define SWITCH2_OUT_PIN 12
4 #define SWITCH1_OUT_PIN 14
5 #define LED_RED_PIN 27
6 #define LED_YELLOW_PIN 26
7 #define LED_GREEN_PIN 25
8 #define HC_SR501_OUTPUT_PIN 32
9 #define LED_ON_OFF_INTERVAL 200000
10
11 // 각종 변수 정의
12 byte ledPinArray[] = {LED_RED_PIN, LED_YELLOW_PIN, LED_GREEN_PIN};
13 byte pirState = LOW;
14 volatile bool isPirMotionDetected = false;
15 volatile long int pirMotionDetectionTime = 0;
16 volatile long int lastDetection = 0;
17 volatile long int currentDetection = 0;
18 hw_timer_t *ledTimer = NULL;
19 volatile bool isButton2Pressed = false;
20 volatile bool isButton1Pressed = false;
21 volatile bool isLedTimerExpired = false;
22 bool isMode2LedOnOffStarted = false;
23 byte mode = 0;
24 byte currentMode = 0;
25
26 void IRAM_ATTR button2Isr() {
27     isButton2Pressed = true;
28 }
29
30 void IRAM_ATTR button1Isr() {
31     isButton1Pressed = true;
32 }
33
34 void IRAM_ATTR ledTimerIsr() {
35     isLedTimerExpired = true;
36 }
37
38 void IRAM_ATTR pirSensorIsr() {
39     if (digitalRead(HC_SR501_OUTPUT_PIN) == HIGH) {
40         isPirMotionDetected = true;
41         lastDetection = currentDetection;
42         currentDetection = millis();
43         pirMotionDetectionTime = currentDetection - lastDetection;
44     } else {
45         isPirMotionDetected = false;
46     }
47 }
48

```

그림 2는 실습1 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab1
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함
- #define 부분은 같게 작성한다.
- setup() 전까지의 부분을 캡처한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 그림 3 실습1 소프트웨어 작성 결과

8\_21101395\_ESP32\_Report1\_Lab1.ino

```

49 void setup() {
50     Serial.begin(115200);
51
52     pinMode(SWITCH2_OUT_PIN, INPUT_PULLUP);
53     pinMode(SWITCH1_OUT_PIN, INPUT_PULLUP);
54     pinMode(LED_RED_PIN, OUTPUT);
55     pinMode(LED_YELLOW_PIN, OUTPUT);
56     pinMode(LED_GREEN_PIN, OUTPUT);
57     pinMode(HC_SR501_OUTPUT_PIN, INPUT);
58
59     attachInterrupt(digitalPinToInterrupt(SWITCH2_OUT_PIN), button2Isr, FALLING);
60     attachInterrupt(digitalPinToInterrupt(SWITCH1_OUT_PIN), button1Isr, FALLING);
61     attachInterrupt(digitalPinToInterrupt(HC_SR501_OUTPUT_PIN), pirSensorIsr, CHANGE);
62
63     ledTimer = timerBegin(0, 80, true);
64     timerAttachInterrupt(ledTimer, &ledTimerIsr, true);
65     timerAlarmWrite(ledTimer, LED_ON_OFF_INTERVAL, true);
66     timerAlarmEnable(ledTimer);
67
68     Serial.printf("\n8-21101395-권준혁> Setup() ended\n");
69 }
70

```

그림 3은 실습1 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab1
- 파일 이름과 함께 캡처한다.
- setup() 부분을 캡처한다.
- Serial.printf() 부분이 있으면 표시를 한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 그림 4 실습1 소프트웨어 작성 결과

```

8_21101395_ESP32_Report1_Lab1.ino
71 void loop() {
72
73     if (isButton1Pressed) {
74         if (!isPirMotionDetected && !isMode2LedOnOffStarted) {
75             currentMode = mode;
76             mode = (mode + 1) % 3;
77             Serial.printf("8-21101395-권준혁> mode changed from mode = %d to mode = %d\n", currentMode, mode);
78         }
79         isButton1Pressed = false;
80     }
81
82     if (isButton2Pressed) {
83         if (mode == 2 && isMode2LedOnOffStarted) {
84             isMode2LedOnOffStarted = false;
85             digitalWrite(LED_RED_PIN, LOW);
86             digitalWrite(LED_YELLOW_PIN, LOW);
87             digitalWrite(LED_GREEN_PIN, HIGH);
88             Serial.printf("8-21101395-권준혁> mode 2 led on/off stopped by button 2\n");
89         }
90         isButton2Pressed = false;
91     }
92
93     if (isLedTimerExpired) {
94         switch (mode) {
95             case 1:
96                 if (isPirMotionDetected) {
97                     static bool toggle = false;
98                     toggle = !toggle;
99                     digitalWrite(LED_RED_PIN, toggle ? HIGH : LOW);
100                    digitalWrite(LED_YELLOW_PIN, LOW);
101                    Serial.printf("8-21101395-권준혁> mode = %d ledTimer interrupt serviced\n", mode);
102                } else {
103                    digitalWrite(LED_RED_PIN, LOW);
104                    digitalWrite(LED_YELLOW_PIN, HIGH);
105                }
106                break;
107
108            case 2:
109                if (isPirMotionDetected || isMode2LedOnOffStarted) {
110                    timerAlarmEnable(ledTimer);
111                    isMode2LedOnOffStarted = true;
112                    static bool toggle = false;
113                    toggle = !toggle;
114                    digitalWrite(LED_RED_PIN, toggle ? HIGH : LOW);
115                    digitalWrite(LED_YELLOW_PIN, toggle ? LOW : HIGH);
116                    digitalWrite(LED_GREEN_PIN, LOW);
117                    Serial.printf("8-21101395-권준혁> mode = %d ledTimer interrupt serviced\n", mode);
118                }
119                break;
120            }
121        isLedTimerExpired = false;
122    }
123

```

그림 4는 실습1 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab1
- 파일 이름과 함께 캡처한다.
- loop() 부분을 캡처한다. (될 수 있는 대로 내용이 이어지는 부분까지 캡처한다.)
- Serial.printf() 부분이 있으면 표시를 한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 그림 5 실습1 소프트웨어 작성 결과

```

8_21101395_ESP32_Report1_Lab1.ino
124     switch (mode) {
125         case 0:
126             digitalWrite(LED_RED_PIN, HIGH);
127             digitalWrite(LED_YELLOW_PIN, LOW);
128             digitalWrite(LED_GREEN_PIN, LOW);
129             break;
130         case 1:
131             if (!isPirMotionDetected) {
132                 digitalWrite(LED_RED_PIN, LOW);
133                 digitalWrite(LED_YELLOW_PIN, HIGH);
134                 digitalWrite(LED_GREEN_PIN, LOW);
135             }
136             break;
137         case 2:
138             if (!isMode2LedOnOffStarted) {
139                 digitalWrite(LED_RED_PIN, LOW);
140                 digitalWrite(LED_YELLOW_PIN, LOW);
141                 digitalWrite(LED_GREEN_PIN, HIGH);
142             }
143             break;
144     }
145
146     static byte lastPirState = LOW;
147     if (lastPirState != isPirMotionDetected) {
148         if (isPirMotionDetected) {
149             Serial.printf("8-21101395-권준혁> mode = %d : Motion detected!\n", mode);
150         } else {
151             Serial.printf("8-21101395-권준혁> mode = %d : Motion ended! : Detection Time = %d\n", mode, pirMotionDetectionTime);
152         }
153         lastPirState = isPirMotionDetected;
154     }
155 }
156

```

그림 5는 실습1 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab1
- 파일 이름과 함께 캡처한다.
- loop() 부분을 캡처한다. (될 수 있는 대로 내용이 이어지는 부분까지 캡처한다.)
- Serial.printf() 부분이 있으면 표시를 한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 그림 7 실습1 프로그램 실행 결과

8\_21101395\_ESP32\_Report1\_Lab1.ino

```

151 } else {
152     Serial.printf("8-21101395-권준혁> mode = %d : Mo
153     //Serial.printf("\n8-21101395-권준혁> Setup() en

```

Output Serial Monitor X

Message (Enter to send message to 'NodeMCU-32S' on 'COM6')

```

8-21101395-권준혁> Setup() ended
8-21101395-권준혁> mode changed from mode = 0 to mode = 1
8-21101395-권준혁> mode changed from mode = 1 to mode = 2
8-21101395-권준혁> mode changed from mode = 2 to mode = 0
8-21101395-권준혁> mode = 0 : Motion detected!
8-21101395-권준혁> mode = 0 : Motion ended! : Detection Time = 17718
8-21101395-권준혁> mode changed from mode = 0 to mode = 1
8-21101395-권준혁> mode = 1 : Motion detected!
8-21101395-권준혁> mode = 1 ledTimer interrupt serviced
8-21101395-권준혁> mode = 1 : Motion ended! : Detection Time = 6144
8-21101395-권준혁> mode changed from mode = 1 to mode = 2
8-21101395-권준혁> mode = 2 : Motion detected!
8-21101395-권준혁> mode = 2 ledTimer interrupt serviced
8-21101395-권준혁> mode = 2 : Motion ended! : Detection Time = 11834
8-21101395-권준혁> mode = 2 ledTimer interrupt serviced
8-21101395-권준혁> mode = 2 led on/off stopped by button 2

```

mode 변경결과 0-&gt;1-&gt;2-&gt;0

mode 0에서 움직임 감지 시작과 끝  
mode1에서 움직임 감지 시작

mode 1에서 timer interrupt 발생

mode 1에서 움직임 감지 끝  
mode 1에서 mode 2로 변경  
mode 2에서 움직임 감지 시작

mode 2에서 timer interrupt 발생

mode 2에서 움직임 감지 끝

mode 2에서 움직임 감지 끝났지만  
계속 timer interrupt 발생

mode 2에서 button 2로 led on/off 중지

그림 7은 실습1 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 순서와 내용으로 출력되도록 동작시켜서 결과를 보인다. (번호-학번-이름은 수정한다.)
- 반드시 그림과 같이 표시하고 설명을 적어야 한다. (없으면 0점 처리한다.)
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab1
- 파일 이름과 프로그램 일부와 함께 캡처한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

#### 1.4 프로그램 실행 결과 2

- 다음 순서로 동작을 수행한 것을 촬영해서 그 동영상을 보고서와 같이 제출한다.
  - 동영상의 이름은 “2024-1-회로이론(1)-ESP32과제1-보고서-출석번호-학번-이름-실습1”(확장자 제외하고)로 설정해야 한다.
  - 동영상 시간 3분 미만, 크기는 20[MB] 미만이 되도록 한다. 이 이상이 되면 크게 감점한다.
  - 다음의 순서로 동작을 진행하고 확인한다.
  - 각 동작의 요구 조건을 완전히 만족시켜야 한다. 일부만 만족하는 경우는 구현되지 않은 것으로 간주한다.
- 1) 처음 시작되면 mode=0이고 빨간색 led만 on 된다. ( )
  - 2) 스위치 1 (왼쪽 스위치)을 순차적으로 눌러서 mode를 1->2->0 순서로 변경한다. ( )
  - 3) 움직임을 감지시킨다. Led는 변화가 없음을 확인한다. ( )
  - 4) 스위치 1을 눌러서 mode=1로 변경한다. 노란색 led만 on 된다. ( )
  - 5) 움직임을 감지시킨다. 노란색 led는 꺼지고 빨간색 led가 약 0.2초 간격으로 on/off 된 후에 노란색 led만 on 된다. ( )
  - 6) 스위치 1을 눌러서 mode=2로 변경한다. 초록색 led만 on 된다. ( )
  - 7) 움직임을 감지시킨다. 초록색 led는 꺼지고 빨간색 led가 약 0.2초 간격으로 on/off 되고 ( ), 노란색 led는 약 0.2초 간격으로 off/on 된다. ( )
  - 8) 스위치 1을 눌러도 7)의 상태가 변화가 없음을 확인한다. ( )
  - 9) 스위치 2를 누르면 초록색 led만 on 된다. ( )
  - 10) 스위치 1을 눌러서 mode=0으로 변경한다. 빨간색 led만 on 된다. ( )

## 1.5 프로그램 실행 결과 3 (새 페이지)

그림 8 프로그램을 사용해서 동영상 파일의 크기를 줄인 결과

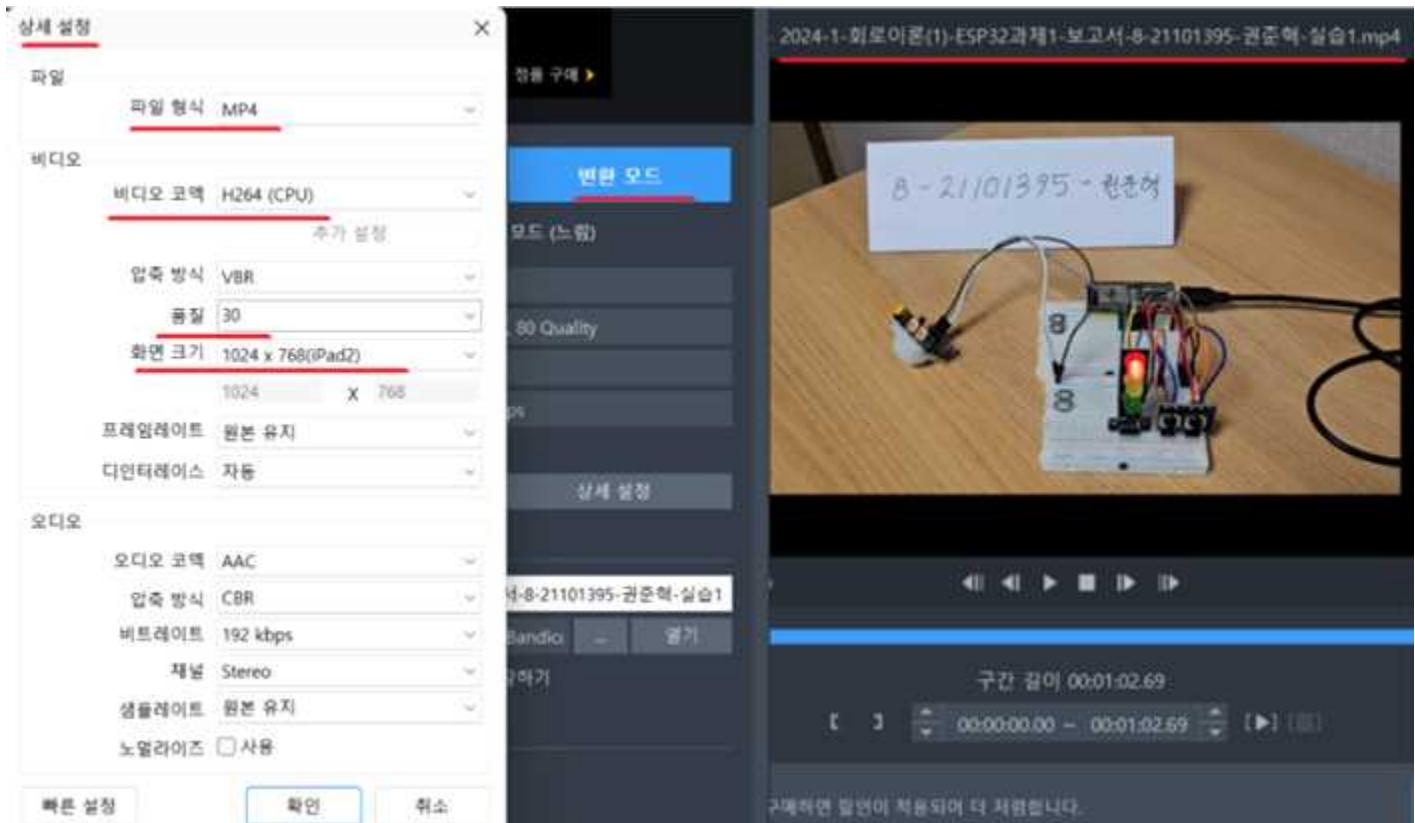


그림 9 프로그램을 사용해서 동영상 파일의 크기를 줄인 결과

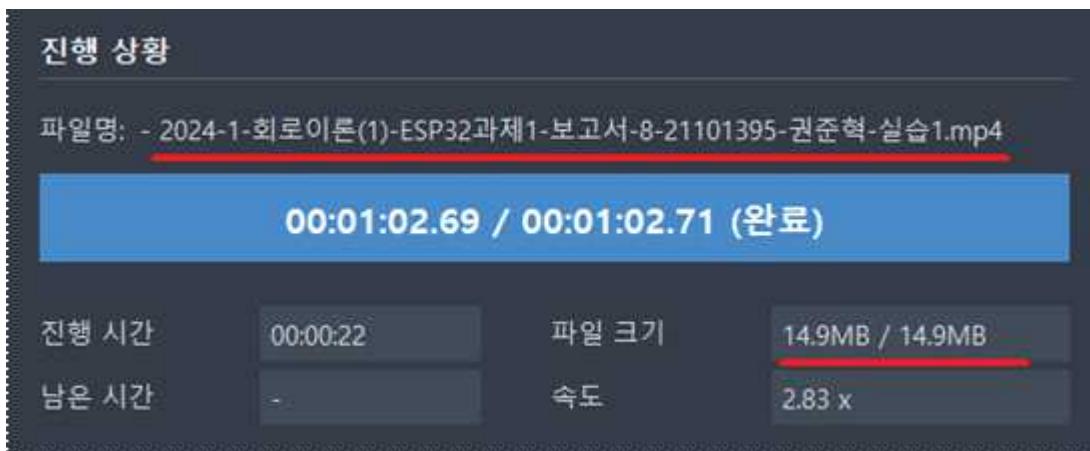


그림 8, 9는 프로그램을 사용해서 동영상 파일의 크기를 줄인 결과를 보여준다.

- 출석번호 = 8번
- => 반디컷 실행하고 자르기 선택 => 동영상 파일 불러온다 => 시작 => 변환 모드 => 상세설정
- => 그림과 같이 설정하고 확인 => 시작
- 품질의 값을 작게 하면 파일 크기를 줄일 수 있다.
- 반드시 변환된 동영상의 시간이 3분 미만이고 크기가 20[MB] 미만인지와 동영상의 내용을 확인한다.
- 다른 프로그램을 사용하는 경우 비슷한 과정의 그림을 캡처해서 보면 된다.

## 2. 실습2 (새 페이지)

Max7219 based Dot Matrix Module의 GND - NodeMCU-32S GND

Max7219 based Dot Matrix Module의 Vcc - NodeMCU-32S 5V (+5[V]) (Dot Matrix Module의 brightness를 최소로 설정해서 전류 소모가 최소가 되도록 해야 한다. 밝기를 최대로 하면 큰 전류가 소모되므로 외부 전원을 사용하는 것이 좋다.)

Max7219 based Dot Matrix Module의 DIN - NodeMCU-32S GPIO23 pin (board에는 G23으로 적혀있다.)

Max7219 based Dot Matrix Module의 CS - NodeMCU-32S GPIO5 pin (board에는 G5로 적혀있다.)

Max7219 based Dot Matrix Module의 CLK - NodeMCU-32S GPIO18 pin (board에는 G18로 적혀있다.)

Temperature and Humidity Sensor AM2302 (wired DHT22)의 - (GND) pin - NodeMCU-32S GND

Temperature and Humidity Sensor AM2302 (wired DHT22)의 + (Vcc) pin - NodeMCU-32S 3V3 (+3.3[V])

Temperature and Humidity Sensor AM2302 (wired DHT22)의 Out pin - NodeMCU-32S GPIO17 pin (board에는 G17로 적혀있다.)

PS2 Joystick Module의 GND - NodeMCU-32S GND

PS2 Joystick Module의 Vcc - NodeMCU-32S 3V3 (+3.3[V])

PS2 Joystick Module의 VRX - NodeMCU-32S GPIO0 pin (board에는 G0으로 적혀있다. ADC2\_1)

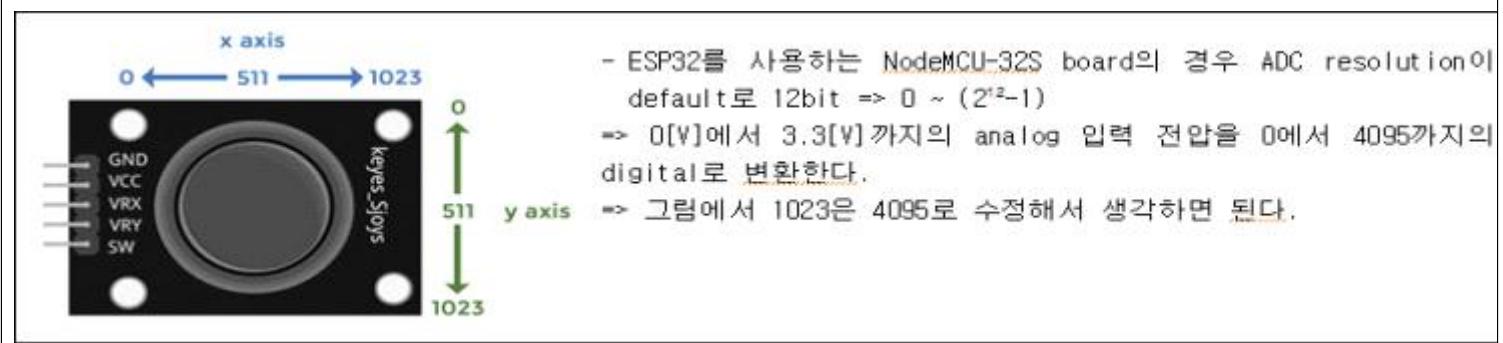
PS2 Joystick Module의 VRY - NodeMCU-32S GPIO4 pin (board에는 G4로 적혀있다. ADC2\_0)

PS2 Joystick Module의 SW - NodeMCU-32S GPIO22 pin (board에는 G22로 적혀있다.)

serial data rate는 115200 baud로 설정한다.

- 2가지 mode로 동작한다.
- Mode 0에서는 10초 간격으로 timer interrupt를 사용해서 온도와 습도를 읽는다. 온도 센서 reading이 정상적이면 그 온도가 기준값을 초과하면 dot matrix에 H를 출력한다. 그 온도가 기준값 이하이면 dot matrix에 L을 출력한다.
- Mode 0에서는 온도 센서 reading이 잘못되면 error message를 출력한다.
- Mode 1에서는 timer interrupt를 사용하지 않는다.
- Mode 1에서는 온도와 습도를 읽고 joystick을 통해서 scroll 방향과 속도를 설정하고 dot matrix에 온도와 습도를 출력해서 scroll 시킨다.
- Mode 1에서는 joystick의 VRX, VRY pin의 전압값을 읽어서 0에서 4095까지의 digital 값으로 변환한다.
- Mode 1에서는 joystick knob을 오른쪽으로 이동하면 scroll 방향을 오른쪽으로, 왼쪽으로 이동하면 scroll 방향을 왼쪽으로 설정한다.
- Mode 1에서는 joystick knob을 위로 이동하면 scroll 속도를 빠르게, 아래로 이동하면 scroll 속도를 느리게 설정한다. 최종 scroll 속도는 100[ms] 이상 500[ms] 이하의 값으로 설정되도록 한다.
- Mode 1에서는 joystick을 움직이지 않으면 이전 설정이 적용된다.

그림 10 Joystick module 기준 방향



- Mode 선택은 joystick switch를 한 번 누를 때마다 mode가 변하는 방식으로 수행한다.
- Joystick switch가 눌렸는지를 external interrupt를 사용해서 감지한다. 스위치가 눌렸으면 mode를 변화시킨다. 초기 mode = 0으로 설정한다.
- Joystick switch debouncing을 간단히 구현한다. Interrupt 발생 횟수를 기록해서 그 값이 예로 10 이상이 되면 interrupt를 처리하는 방법을 사용할 수 있다.
- Serial monitor를 통해서 문자를 수신하면 문자에 따른 동작을 수행한다.
- Serial monitor를 통해서 수신된 문자가 'p'(or 'P')이고 현재 dot matrix display가 정지상태가 아니고 mode 1이라면 dot matrix display를 정지한다. (MD\_Parola library displaySuspend() 참고)
- Serial monitor를 통해서 수신된 문자가 'r'(or 'R')이고 현재 dot matrix display가 정지상태이고 mode 1이라면 현재 dot matrix display를 재시작한다.(MD\_Parola library displaySuspend() 참고)
- Serial monitor를 통해서 수신된 문자가 '0'이고 mode = 1이면 mode = 0으로 변경한다.
- Serial monitor를 통해서 수신된 문자가 '1'이고 mode = 0이면 mode = 1로 변경한다.
- Serial monitor를 통해서 수신된 문자가 't'이면 그다음 문자들을 실숫값으로 읽는다. 현재 mode = 0이고 실숫값이 20.0 이상 30.0 이하이면 mode 0에서의 온도 기준값으로 설정한다.
- Serial monitor를 통해서 수신된 문자가 's'이면 그다음 문자들을 정숫값으로 읽는다. 현재 mode = 1이고 정숫값이 100 이상 500 이하이면 dot matrix scroll 속도 값으로 설정한다.
- Serial monitor를 통해서 수신된 문자가 'd'이면 그다음 문자를 읽는다. 현재 mode = 1이고 그다음 문자가 'L' 이면 dot matrix scroll 방향을 PA\_SCROLL\_LEFT, mode = 1이고 그다음 문자가 'R' 이면 PA\_SCROLL\_RIGHT로 설정한다.
- delay() 함수는 사용하지 않는다.
- interrupt service routine은 최대로 짧게 작성하고 그 안에서 serial monitor와의 입출력을 수행하지 않는다.
- 뒤의 결과 그림의 예와 같이 serial monitor에 메시지를 Serial.printf()를 사용해서 출력한다.

## 2.1 하드웨어 구성 결과 (새 페이지)

그림 11 실습2 하드웨어 구성 결과

8 - 21101395 - 권준혁

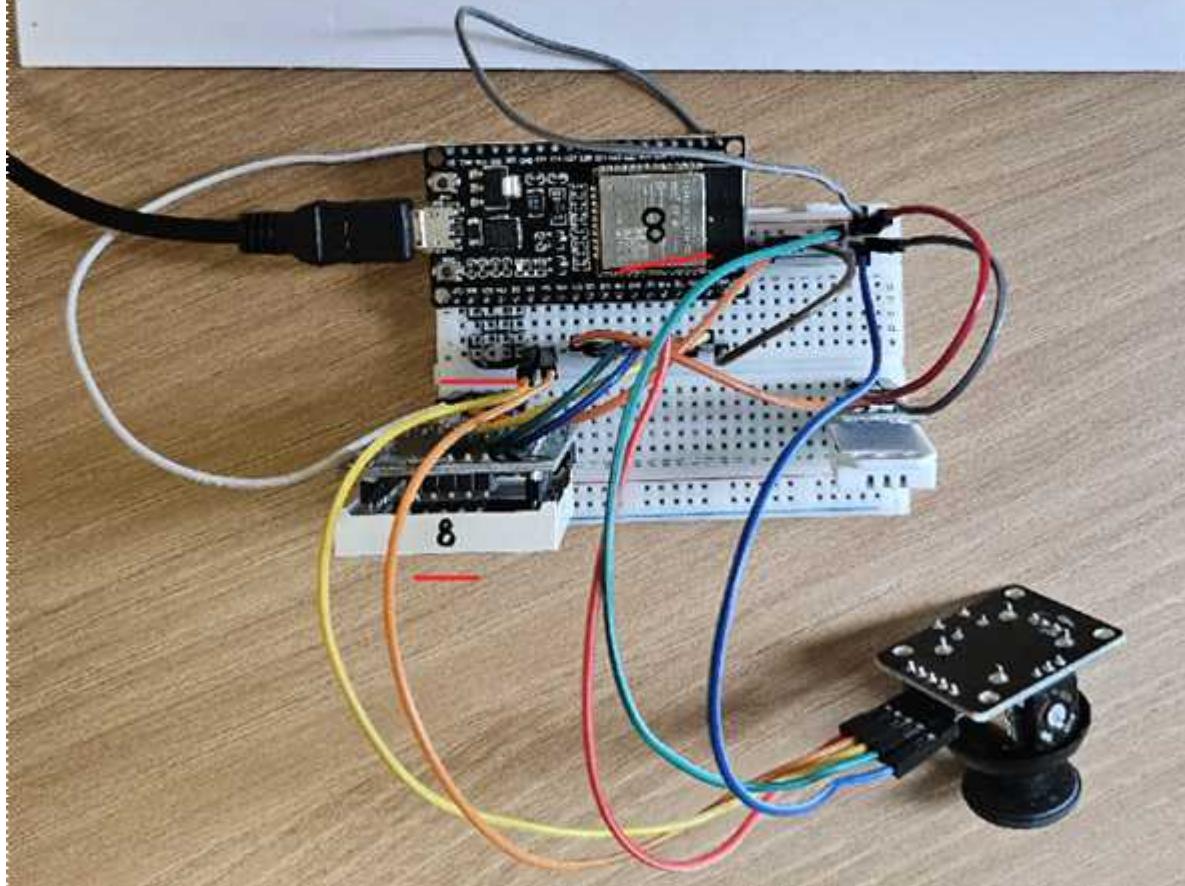


그림 11은 실습2 하드웨어 구성 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 번호-학번-이름을 포함한다.
- 반드시 그림과 같이 브레드보드와 ESP32 NodeMCU-32S 보드에 출석번호를 직접 적는다.  
=> 출석번호를 확실하게 확인할 수 있어야 한다.
- 요구되는 표시가 하나라도 없으면 0점 처리한다.

## 2.2 소프트웨어 작성 결과 1 (새 페이지)

### 그림 12 실습2 소프트웨어 작성 결과

#### 8\_21101395\_ESP32\_Report1\_Lab2.ino

```
1 // 8-21101395-권준혁 ESP32과제1 실습2
2 #include "DHT.h"
3 #include <MD_Parola.h>
4 #include <MD_MAX72xx.h>
5 #include <SPI.h>
6
7 #define DHT_PIN 17
8 #define DHT_TYPE DHT22
9 #define DHT_READ_INTERVAL 10000000
10 #define HARDWARE_TYPE MD_MAX72XX::GENERIC_HW
11 #define MAX_DEVICES 1
12 #define CS_PIN 5
13 #define JOYSTICK_PS2_SW_PIN 22
14 #define JOYSTICK_PS2_VRY_PIN 4
15 #define JOYSTICK_PS2_VRX_PIN 0
16 #define JOYSTICK_SWITCH_INTERRUPT_COUNT_MAX 10
17 #define MESSAGE_BUFFER_SIZE 20
18
19 char messageBuffer[MESSAGE_BUFFER_SIZE];
20 DHT dht(DHT_PIN, DHT_TYPE);
21 float humidity, temperature;
22 float temperatureLimit = 26.5;
23 bool isDhtTimerExpired = false;
24 uint16_t scrollSpeed = 300;
25 textEffect_t scrollEffect = PA_SCROLL_LEFT;
26 uint16_t scrollPause = 500;
27 bool isPaused = false;
28 textPosition_t scrollAlign = PA_LEFT;
29 MD_Parola Display = MD_Parola(HARDWARE_TYPE, CS_PIN, MAX_DEVICES);
30 byte mode = 0;
31 unsigned int joystickPS2Vrx, joystickPS2Vry;
32 volatile unsigned int joystickSwitchInterruptCount = 0;
```

그림 12는 실습2 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab2
- 첫 줄은 반드시 그림과 같은 번호-학번-이름 포함
- #define 부분은 같게 작성한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 그림 13 실습2 소프트웨어 작성 결과

```

8_21101395_ESP32_Report1_Lab2.ino
33 volatile bool isJoystickSwitchPressed = false;
34
35 hw_timer_t *timer = NULL;
36 portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;
37
38 void IRAM_ATTR onTimer() {
39     portENTER_CRITICAL_ISR(&timerMux);
40     isDhtTimerExpired = true;
41     portEXIT_CRITICAL_ISR(&timerMux);
42 }
43
44 void IRAM_ATTR handleJoystickSwitch() {
45     if (++joystickSwitchInterruptCount >= JOYSTICK_SWITCH_INTERRUPT_COUNT_MAX) {
46         isJoystickSwitchPressed = true;
47     }
48 }
49
50 void setup() {
51     Serial.begin(115200);
52     pinMode(JOYSTICK_PS2_SW_PIN, INPUT_PULLUP);
53     attachInterrupt(digitalPinToInterrupt(JOYSTICK_PS2_SW_PIN), handleJoystickSwitch, FALLING);
54     dht.begin();
55     display.begin();
56     display.setIntensity(0);
57     display.displayClear();
58
59     timer = timerBegin(0, 80, true);
60     timerAttachInterrupt(timer, &onTimer, true);
61     timerAlarmWrite(timer, DHT_READ_INTERVAL, true);
62     timerAlarmEnable(timer);
63     Serial.printf("8-21101395-권준혁> Setup ended\n");
64 }

```

그림 13은 실습2 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab2
- 파일 이름과 함께 캡처한다.
- setup() 함수까지 캡처한다.
- Serial.printf() 부분이 있으면 표시를 한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 2.2 소프트웨어 작성 결과 3 (새 페이지)

### 그림 14 실습2 소프트웨어 작성 결과

```
8-21101395_ESP32_Report1_Lab2.ino
 66 void loop() {
 67   if (isJoystickSwitchPressed) {
 68     isJoystickSwitchPressed = false;
 69     mode = !mode;
 70     Serial.printf("8-21101395-권준혁> mode changed from mode=%d to mode=%d : joystickSwitchInterruptCount = %d\n", mode, !mode, joystickSwitchInterruptCount);
 71     joystickSwitchInterruptCount = 0;
 72     Display.displayClear();
 73     if (mode == 1) {
 74       timerAlarmDisable(timer);
 75     } else {
 76       timerAlarmEnable(timer);
 77     }
 78   }
 79   if (mode == 0 && isLightTimerExpired) {
 80     isLightTimerExpired = false;
 81     sensorDisplay();
 82   } else if (mode == 1) {
 83     joystickInput();
 84     scrolltest();
 85   }
 86
 87   if (Serial.available() > 0) {
 88     inputSerial();
 89   }
 90 }
 91
 92 void sensorDisplay() {
 93   temperature = dht.readTemperature();
 94   humidity = dht.readHumidity();
 95   if (isnan(temperature) || isnan(humidity)) {
 96     Serial.printf("8-21101395-권준혁> Failed to read from DHT sensor!\n");
 97   } else {
 98     if (Display.displayAnimate()) {
 99       Serial.printf("8-21101395-권준혁> Mode = %d : temperature = %.1f[°C], humidity = %.1f[%]\n", mode, temperature, humidity);
100     if (temperature > temperatureLimit) {
101       display.drawText("H", PA_CENTER, 0, 0, PA_PRINT, PA_NO_EFFECT);
102       Serial.printf("8-21101395-권준혁> Mode = %d : Temperature = %.1f[°C]\n", mode, temperature, temperatureLimit);
103     } else if (temperature < temperatureLimit) {
104       display.drawText("L", PA_CENTER, 0, 0, PA_PRINT, PA_NO_EFFECT);
105       Serial.printf("8-21101395-권준혁> Mode = %d : Temperature = %.1f[°C]\n", mode, temperature, temperatureLimit);
106     }
107   }
108 }
109
110 }
```

그림 14는 실습2 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab2
- 파일 이름과 함께 캡처한다.
- loop() 함수부터 될 수 있는 대로 내용이 이어지는 부분까지 캡처한다.
- Serial.printf() 부분이 있으면 표시를 한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 2.2 소프트웨어 작성 결과 4 (새 페이지)

### 그림 15 실습2 소프트웨어 작성 결과

```
8_21101395_ESP32_Report1_Lab2.ino
111 void scrolltest() {
112     if (!isPaused) {
113         temperature = dht.readTemperature();
114         humidity = dht.readHumidity();
115         snprintf(messageBuffer, MESSAGE_BUFFER_SIZE, isnan(temperature) && isnan(humidity) ? "%lf, %lf" : "ERROR", temperature, humidity);
116
117         if (Display.displayAnimate()) {
118             Display.displayText(messageBuffer, scrollAlign, scrollSpeed, scrollPause, scrollEffect, scrollEffect);
119             Serial.printf("8-21101395-권준혁> Mode = %d : temperature = %.1f[℃], humidity = %.1f[%]\n", mode, temperature, humidity);
120             Serial.printf("8-21101395-권준혁> Mode = %d : Joystick PS2 VRX = %d, VRY = %d\n", mode, joystickPS2Vrx, joystickPS2Vry);
121
122         if (joystickPS2Vrx >= 4095 && joystickPS2Vry >= 4095) {
123             Serial.printf("8-21101395-권준혁> Mode = %d : scrollEffect = PA_SCROLL_RIGHT \n", mode);
124             Serial.printf("8-21101395-권준혁> Mode = %d : scrollSpeed = 500 \n", mode);
125         }
126         else if (joystickPS2Vrx >= 4095 && joystickPS2Vry <= 0) {
127             Serial.printf("8-21101395-권준혁> Mode = %d : scrollEffect = PA_SCROLL_RIGHT \n", mode);
128             Serial.printf("8-21101395-권준혁> Mode = %d : scrollSpeed = 100 \n", mode);
129         }
130         else if (joystickPS2Vrx <= 0 && joystickPS2Vry >= 4095) {
131             Serial.printf("8-21101395-권준혁> Mode = %d : scrollEffect = PA_SCROLL_LEFT \n", mode);
132             Serial.printf("8-21101395-권준혁> Mode = %d : scrollSpeed = 500 \n", mode);
133         }
134         else if (joystickPS2Vrx <= 0 && joystickPS2Vry <= 0) {
135             Serial.printf("8-21101395-권준혁> Mode = %d : scrollEffect = PA_SCROLL_LEFT \n", mode);
136             Serial.printf("8-21101395-권준혁> Mode = %d : scrollSpeed = 100 \n", mode);
137         }
138     }
139 } else {
140     Display.displayClear();
141 }
142 }
```

그림 15는 실습2 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab2
- 파일 이름과 함께 캡처한다.
- 될 수 있는 대로 내용이 이어지는 부분까지 캡처한다.
- Serial.printf() 부분이 있으면 표시를 한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 2.2 소프트웨어 작성 결과 5 (새 페이지)

### 그림 16 실습2 소프트웨어 작성 결과

```
8_21101395_ESP32_Report_Lab2.ino
144 void joystickInput() {
145     joystickPS2Vrx = analogRead(JOYSTICK_PS2_VRX_PIN);
146     joystickPS2Vry = analogRead(JOYSTICK_PS2_VRY_PIN);
147     scrollEffect = joystickPS2Vrx < 1000 ? PA_SCROLL_LEFT : joystickPS2Vrx > 3000 ? PA_SCROLL_RIGHT : scrollEffect;
148     scrollSpeed = joystickPS2Vry < 1000 ? min(scrollSpeed + 10, 100) : joystickPS2Vry > 3000 ? max(scrollSpeed - 10, 500) : scrollSpeed;
149     Display.setTextEffect(scrollEffect, PA_NO_EFFECT);
150     Display.setSpeed(scrollSpeed);
151 }
152
153 void inputSerial() {
154     char input = Serial.read();
155     if (input >= 'a' && input <= 'z') {
156         input -= 32;
157     }
158
159     if (input == 'P') {
160         inputPause();
161     } else if (input == 'R') {
162         inputDisplay();
163     } else if (input == 'e' || input == 'i') {
164         inputCount(input);
165     } else if (input == 'T') {
166         inputTemp();
167     } else if (input == 'S') {
168         inputSpeed();
169     } else if (input == 'O') {
170         inputDirection();
171     }
172 }
173
174 void inputPause() {
175     if (mode == 1 && !isPaused) {
176         isPaused = true;
177     }
178     Serial.printf("8-21101395-%d> Mode = %d : suspend display updates (from serial monitor)\n", mode);
179 }
180
181 void inputDisplay() {
182     if (mode == 1 && isPaused) {
183         isPaused = false;
184     }
185     Serial.printf("8-21101395-%d> Mode = %d : resume display updates (from serial monitor)\n", mode);
186 }
187
188 void inputCount(char input) {
189     mode = input - 'e';
190     if (mode == 0) {
191         timerAlarmEnable(timer);
192     } else {
193         timerAlarmDisable(timer);
194     }
195     Display.displayClear();
196     Serial.printf("8-21101395-%d> change mode to %d (from serial monitor)\n", mode);
197 }
198
199 void inputTemp() {
200     if (mode == 0) {
201         float newTempLimit = Serial.parseFloat();
202         if (newTempLimit >= 20.0 && newTempLimit <= 30.0) {
203             temperatureLimit = newTempLimit;
204             Serial.printf("8-21101395-%d> Mode = %d : temperatureLimit = %.1f (from serial monitor)\n", mode, temperatureLimit);
205         }
206     }
207 }
```

그림 16은 실습2 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab2
- 파일 이름과 함께 캡처한다.
- 될 수 있는 대로 내용이 이어지는 부분까지 캡처한다.
- Serial.printf() 부분이 있으면 표시를 한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 그림 17 실습2 소프트웨어 작성 결과

```

8_21101395_ESP32_Report1_Lab2.ino
209 void inputSpeed() {
210     if (mode == 1) {
211         int newScrollSpeed = Serial.parseInt();
212         if (newScrollSpeed >= 100 && newScrollSpeed <= 500) {
213             scrollSpeed = newScrollSpeed;
214             Display.setSpeed(scrollSpeed);
215             Serial.printf("8-21101395-권준혁> Mode = %d : scrollSpeed = %d (from serial monitor)\n", mode, scrollSpeed);
216         }
217     }
218 }
219
220 void inputDirection() {
221     if (mode == 1) {
222         char direction = Serial.read();
223         if (direction == 'L') {
224             scrollEffect = PA_SCROLL_LEFT;
225             Serial.printf("8-21101395-권준혁> Mode = %d : scrollEffect = PA_SCROLL_LEFT (from serial monitor)\n", mode);
226         } else if (direction == 'R') {
227             scrollEffect = PA_SCROLL_RIGHT;
228             Serial.printf("8-21101395-권준혁> Mode = %d : scrollEffect = PA_SCROLL_RIGHT (from serial monitor)\n", mode)
229         }
230     }
231 }

```

그림 17은 실습2 소프트웨어 작성 결과를 보여준다.

- 출석번호 = 8번
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab2
- 파일 이름과 함께 캡처한다.
- 될 수 있는 대로 내용이 이어지는 부분까지 캡처한다.
- Serial.printf() 부분이 있으면 표시를 한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 2.3 프로그램 실행 결과 1 (새 페이지)

### 그림 18 실습2 프로그램 실행 결과

The screenshot shows the Arduino Serial Monitor window for the file "8\_21101395\_ESP32\_Report1\_Lab2.ino". The monitor displays a series of text messages from the serial port. The text is color-coded with red annotations on the right side of the screen:

- 3)번 동작: 현재 온도가 기준 온도를 초과 => H 출력
- 4)번 동작: mode=1 설정
- Joystick 움직이기 전의 출력
- 5)번 동작: Joystick 우상방향
- Joystick 가운데 위치: 이전 5번 동작 계속
- 6)번 동작: Joystick 좌하 방향
- 7)번 동작: Joystick 우하 방향
- 8)번 동작: Joystick 좌상 방향
- Joystick 가운데 위치: 이전 8번 동작 계속
- 9)번 동작: serial monitor에서 0을 입력

The main text area shows the following log entries:

```
8-21101395-권준혁> Setup ended
8-21101395-권준혁> Mode = 0 : temperature = 28.5[°C], humidity = 48.3[%]
8-21101395-권준혁> Mode = 0 : Temperature = 28.5[°C] > 26.5[°C]
8-21101395-권준혁> mode changed from mode = 1 to mode = 0 : joystickSwitchInterruptCount = 10
8-21101395-권준혁> Mode = 1 : temperature = 28.6[°C], humidity = 48.0[%]
8-21101395-권준혁> Mode = 1 : Joystick PS2 VRX = 2685, VRY = 1818
8-21101395-권준혁> Mode = 1 : temperature = 28.6[°C], humidity = 47.9[%]
8-21101395-권준혁> Mode = 1 : Joystick PS2 VRX = 4085, VRY = 0
8-21101395-권준혁> Mode = 1 : scrollIEffect = PA_SCROLL_RIGHT
8-21101395-권준혁> Mode = 1 : scrollISpeed = 100
8-21101395-권준혁> Mode = 1 : temperature = 28.6[°C], humidity = 47.9[%]
8-21101395-권준혁> Mode = 1 : Joystick PS2 VRX = 2681, VRY = 1819
8-21101395-권준혁> Mode = 1 : temperature = 28.6[°C], humidity = 47.7[%]
8-21101395-권준혁> Mode = 1 : Joystick PS2 VRX = 0, VRY = 4085
8-21101395-권준혁> Mode = 1 : scrollIEffect = PA_SCROLL_LEFT
8-21101395-권준혁> Mode = 1 : scrollISpeed = 500
8-21101395-권준혁> Mode = 1 : temperature = 28.6[°C], humidity = 47.6[%]
8-21101395-권준혁> Mode = 1 : Joystick PS2 VRX = 4085, VRY = 4085
8-21101395-권준혁> Mode = 1 : scrollIEffect = PA_SCROLL_RIGHT
8-21101395-권준혁> Mode = 1 : scrollISpeed = 500
8-21101395-권준혁> Mode = 1 : temperature = 28.6[°C], humidity = 47.4[%]
8-21101395-권준혁> Mode = 1 : Joystick PS2 VRX = 0, VRY = 0
8-21101395-권준혁> Mode = 1 : scrollIEffect = PA_SCROLL_LEFT
8-21101395-권준혁> Mode = 1 : scrollISpeed = 100
8-21101395-권준혁> Mode = 1 : temperature = 28.6[°C], humidity = 47.4[%]
8-21101395-권준혁> Mode = 1 : Joystick PS2 VRX = 2682, VRY = 1821
8-21101395-권준혁> Mode = 1 : temperature = 28.6[°C], humidity = 48.0[%]
8-21101395-권준혁> Mode = 1 : Joystick PS2 VRX = 2683, VRY = 1819
8-21101395-권준혁> change mode to 0 (from serial monitor)
```

그림 18은 실습2 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 순서와 내용으로 출력되도록 동작시켜서 결과를 보인다. (번호-학번-이름은 수정한다.)
- 반드시 그림과 같이 표시하고 설명을 적어야 한다. (없으면 0점 처리한다.)
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab2
- 파일 이름과 프로그램 일부와 함께 캡처한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 2.3 프로그램 실행 결과 1 (새 페이지)

그림 19 실습2 프로그램 실행 결과

```
8_21101395_ESP32_Report1_Lab2.ino
1 // 8_21101395-권준혁 ESP32과제1 실습2

Output Serial Monitor ×
Not connected. Select a board and a port to connect automatically. New Line 115200 baud

9-21101395-권준혁> Node = 0 : temperature = 28.5[°C], humidity = 40.3[%]
9-21101395-권준혁> Node = 0 : Temperature = 29.5[°C] > 26.5[°C]
9-21101395-권준혁> Node = 0 : temperature_limit = 29.9 (from serial monitor)
9-21101395-권준혁> Node = 0 : temperature = 28.5[°C], humidity = 40.2[%]
9-21101395-권준혁> Node = 0 : Temperature = 28.5[°C] > 23.9[°C]
9-21101395-권준혁> Node = 0 : temperature_limit = 29.9 (from serial monitor)
9-21101395-권준혁> Node = 0 : temperature = 28.5[°C], humidity = 40.1[%]
9-21101395-권준혁> Node = 0 : Joystick PS2 VRX = 2675, VRY = 1821
9-21101395-권준혁> Node = 0 : suspend display updates (from serial monitor)
9-21101395-권준혁> Node = 0 : resume display updates (from serial monitor)
9-21101395-권준혁> Node = 1 : temperature = 28.4[°C], humidity = 40.2[%]
9-21101395-권준혁> Node = 1 : Joystick PS2 VRX = 2678, VRY = 1818
9-21101395-권준혁> Node = 1 : scrollSpeed = 500 (from serial monitor)
9-21101395-권준혁> Node = 1 : temperature = 28.4[°C], humidity = 40.1[%]
9-21101395-권준혁> Node = 1 : Joystick PS2 VRX = 2679, VRY = 1819
9-21101395-권준혁> Node = 1 : scrollSpeed = 100 (from serial monitor)
9-21101395-권준혁> Node = 1 : temperature = 28.4[°C], humidity = 40.0[%]
9-21101395-권준혁> Node = 1 : Joystick PS2 VRX = 2678, VRY = 1818
9-21101395-권준혁> Node = 1 : temperature = 28.5[°C], humidity = 40.0[%]
9-21101395-권준혁> Node = 1 : Joystick PS2 VRX = 2679, VRY = 1819
9-21101395-권준혁> Node = 1 : scrollEffect = PA_SCROLL_RIGHT (from serial monitor)
9-21101395-권준혁> Node = 1 : temperature = 28.5[°C], humidity = 40.1[%]
9-21101395-권준혁> Node = 1 : Joystick PS2 VRX = 2679, VRY = 1821
9-21101395-권준혁> Node = 1 : scrollEffect = PA_SCROLLLEFT (from serial monitor)
9-21101395-권준혁> Node = 1 : temperature = 28.5[°C], humidity = 40.2[%]
9-21101395-권준혁> Node = 1 : Joystick PS2 VRX = 2677, VRY = 1819
```

9)번 동작: serial monitor에서 0을 입력

10)번 동작: serial monitor에서 t23.9를 입력

11)번 동작: serial monitor에서 t29.9를 입력

12)번 동작: serial monitor에서 1을 입력  
입력이 없는 경우 이전 설정에 따라서 동작 계속

13)번 동작: serial monitor에서 p를 입력

14)번 동작: serial monitor에서 r를 입력

입력이 없는 경우 이전 설정에 따라서 동작 계속

15)번 동작: serial monitor에서 s500을 입력

16)번 동작: serial monitor에서 s100을 입력

17)번 동작: serial monitor에서 dR을 입력

18)번 동작: serial monitor에서 dl을 입력

그림 19는 실습2 프로그램 실행 결과를 보여준다.

- 출석번호 = 8번
- 반드시 그림과 같은 순서와 내용으로 출력되도록 동작시켜서 결과를 보인다. (번호-학번-이름은 수정한다.)
- 반드시 그림과 같이 표시하고 설명을 적어야 한다. (없으면 0점 처리한다.)
- 파일 이름은 반드시 번호\_학번\_ESP32\_Report1\_Lab2
- 파일 이름과 프로그램 일부와 함께 캡처한다.
- 요구되는 표시와 내용이 하나라도 없으면 0점 처리한다.

## 2.4 프로그램 실행 결과 2

- 다음 순서로 동작을 수행한 것을 촬영해서 그 동영상을 보고서와 같이 제출한다.
  - 동영상의 이름은 “2024-1-회로이론(1)-ESP32과제1-보고서-출석번호-학번-이름-실습2”(확장자 제외하고)로 설정해야 한다.
  - 시간 8분 미만, 크기는 50[MB] 미만이 되도록 한다. 이 이상이 되면 크게 감점한다.
  - 다음의 순서로 동작을 진행하고 확인한다.
    - 각 동작의 요구 조건을 완전히 만족시켜야 한다. 일부만 만족하는 경우는 구현되지 않은 것으로 간주한다.
- 1) 보드의 RST switch를 눌러서 serial monitor에 Setup() ended message를 출력시킨다.
  - 2) dot matrix module에 “L” 이 출력됨을 확인한다. ( )
  - 3) 온도 센서를 손으로 잡고 약 10초 후에 dot matrix module에 “H” 가 출력됨을 확인한다. ( )
  - 4) Joystick 스위치를 눌러서 mode=1로 변경한다. ( )
  - 5) Joystick을 우상 방향으로 움직여서 dot matrix module의 text가 오른쪽으로 빠르게 scroll 되는 것을 확인한다. ( )
  - 6) Joystick을 좌하 방향으로 움직여서 dot matrix module의 text가 왼쪽으로 느리게 scroll 되는 것을 확인한다. ( )
  - 7) Joystick을 우하 방향으로 움직여서 dot matrix module의 text가 오른쪽으로 느리게 scroll 되는 것을 확인한다. ( )
  - 8) Joystick을 좌상 방향으로 움직여서 dot matrix module의 text가 왼쪽으로 빠르게 scroll 되는 것을 확인한다. ( )
  - 9) serial monitor에서 0을 입력해서 mode=0으로 변경한다. ( )
  - 10) serial monitor에서 t23.9를 입력해서 dot matrix module에 “H” 가 출력됨을 확인한다. ( )
  - 11) serial monitor에서 t29.9를 입력해서 dot matrix module에 “L” 가 출력됨을 확인한다. ( )
  - 12) serial monitor에서 1을 입력해서 mode=1로 변경한다. ( )
  - 13) serial monitor에서 p를 입력해서 dot matrix module의 display가 중지되는 것을 확인한다. ( )
  - 14) serial monitor에서 r을 입력해서 dot matrix module의 display가 다시 시작되는 것을 확인한다. ( )
  - 15) serial monitor에서 s500을 입력해서 dot matrix module의 text가 왼쪽으로 느리게 scroll 되는 것을 확인한다. ( )
  - 16) serial monitor에서 s100을 입력해서 dot matrix module의 text가 왼쪽으로 빠르게 scroll 되는 것을 확인한다. ( )
  - 17) serial monitor에서 dR을 입력해서 dot matrix module의 text가 오른쪽으로 빠르게 scroll 되는 것을 확인한다. ( )
  - 18) serial monitor에서 dL을 입력해서 dot matrix module의 text가 왼쪽으로 빠르게 scroll 되는 것을 확인한다. ( )

## 2.5 프로그램 실행 결과 3 (새 페이지)

그림 21 프로그램을 사용해서 동영상 파일의 크기를 줄인 결과



그림22



그림 21은 프로그램을 사용해서 동영상 파일의 크기를 줄인 결과를 보여준다.

- 출석번호 = 8번
- => 반디컷 실행하고 자르기 선택 => 동영상 파일 불러온다 => 시작 => 변환 모드 => 상세설정
- => 그림과 같이 설정하고 확인 => 시작
- 품질의 값을 작게 하면 파일 크기를 줄일 수 있다.
- 반드시 변환된 동영상의 시간이 8분 미만이고 크기가 50[MB] 미만인지와 동영상의 내용을 확인한다.
- 다른 프로그램을 사용하는 경우 비슷한 과정의 그림을 캡처해서 보이면 된다.