

## [마이크로프로세서및실습] 과제3 2020136129 최수연

### 1. 과제 제목

- 4-Digit 7-Segment

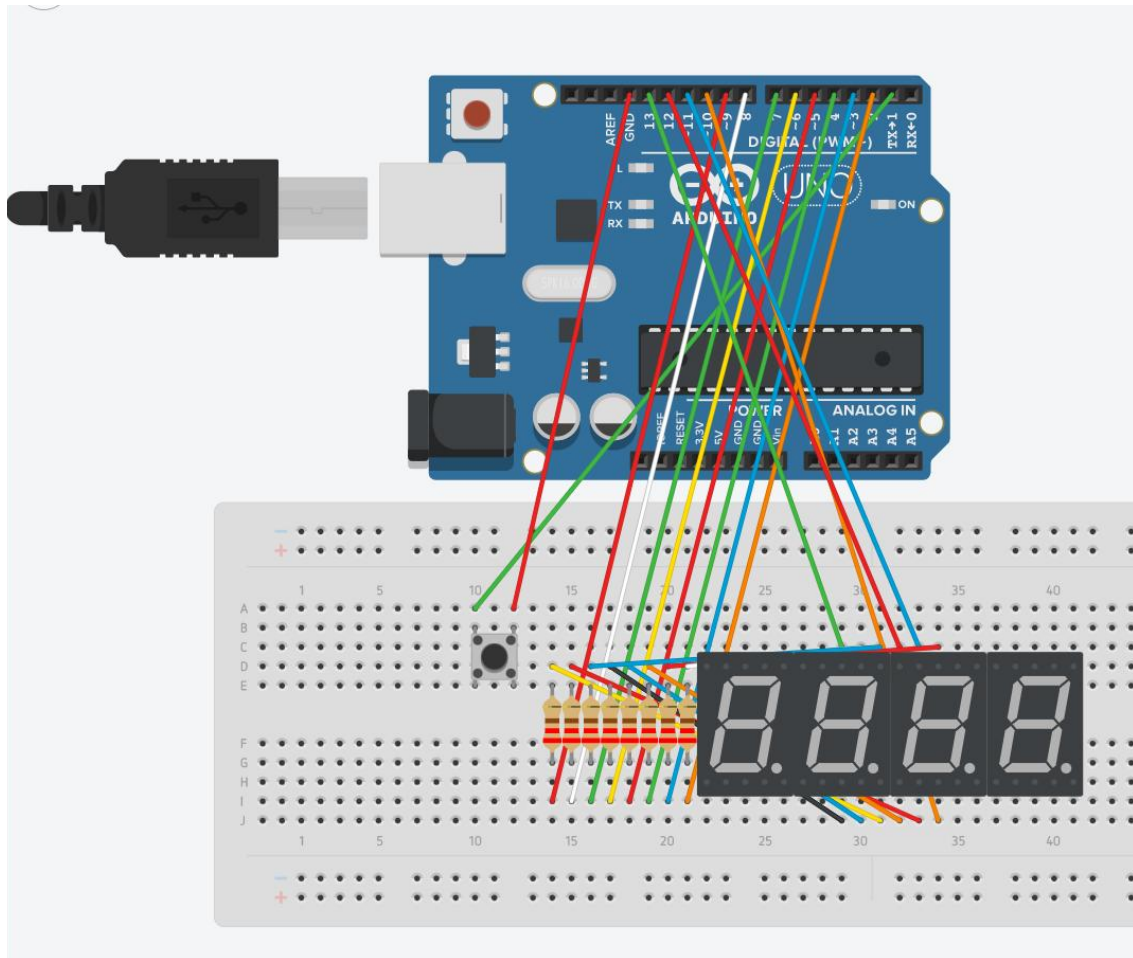
### 2. 부품 리스트

- 아두이노 우노 R3
- 830 포인트 브레드보드
- 숫자 점퍼 케이블
- 아두이노 USB 케이블
- 4-Digit 7-Segment
- 저항 220옴

### 3. 구현하고자 하는 기능 및 설명

- 4-RoW 세그먼트와 Switch를 아두이노에 연결한다.
- 다음 조건을 만족하는 타이머를 설계한다.
  - 시스템이 Reset 되면 화면에 "--.--"로 표시한다.
  - Switch 를 누르면 타이머가 시작된다.
  - 시간표시는 "MM.SS"로 한다. MM은 분단위 표시. SS는 초단위 표시. MM과 SS 사이는 DP(decimal point)를 사용하여 초단위로 점멸(blink)하게 한다.
  - 타이머가 동작상태에서 switch를 누르면 타이머가 정지한다.
  - 정지상태에서 switch를 누르면 타이머가 계속되고, 5초이상 길게 누르면 clear (화면에 00.00 표시) 한다.
  - Clear 상태 switch를 누르면 재시작한다.

#### 4. 부품과 아두이노 우노 보드와의 연결도



## 5. 프로그램 소스

```
// pin 2-9 A-H
// pin 10-13 S0-S3
int seg[][8] = { {0, 0, 0, 0, 0, 0, 1, 1}, // 0
                 {1, 0, 0, 1, 1, 1, 1, 1}, // 1
                 {0, 0, 1, 0, 0, 1, 0, 1}, // 2
                 {0, 0, 0, 0, 1, 1, 0, 1}, // 3
                 {1, 0, 0, 1, 1, 0, 0, 1}, // 4
                 {0, 1, 0, 0, 1, 0, 0, 1}, // 5
                 {0, 1, 0, 0, 0, 0, 0, 1}, // 6
                 {0, 0, 0, 1, 1, 1, 1, 1}, // 7
                 {0, 0, 0, 0, 0, 0, 0, 1}, // 8
                 {0, 0, 0, 0, 1, 0, 0, 1} }; // 9
int seg1[][8] = { {0, 0, 0, 0, 0, 0, 1, 0}, // 0
                  {1, 0, 0, 1, 1, 1, 1, 0}, // 1
                  {0, 0, 1, 0, 0, 1, 0, 0}, // 2
                  {0, 0, 0, 0, 1, 1, 0, 0}, // 3
                  {1, 0, 0, 1, 1, 0, 0, 0}, // 4
                  {0, 1, 0, 0, 1, 0, 0, 0}, // 5
                  {0, 1, 0, 0, 0, 0, 0, 0}, // 6
                  {0, 0, 0, 1, 1, 1, 1, 0}, // 7
                  {0, 0, 0, 0, 0, 0, 0, 0}, // 8
                  {0, 0, 0, 0, 1, 0, 0, 0} }; // 9
int seg_pin[] = {2, 3, 4, 5, 6, 7, 8, 9};
int sel_pin[] = {10, 11, 12, 13}; // 어디 부분의 segment사용할지 결정. 연결 부분 확인
int cnt = 0;                       // 각 sel의 시간 조정을 위한 cnt 변수 초기화
int n[4] = {0, 0, 0, 0};          // 배열 n[]에 따라 sel에 표시될 숫자 조정
int sw_pin = 1;
int start = 0;
int time_cnt = 0;
boolean curr_sw = LOW;
boolean last_sw = LOW;
boolean sw_state = HIGH;
boolean stop_r = false; //추가
void setup() {
    // put your setup code here, to run once:
    int i;
    pinMode(sw_pin, INPUT_PULLUP);
    for (i = 0; i < 8; i++) pinMode(seg_pin[i], OUTPUT);
    for (i = 0; i < 4; i++) pinMode(sel_pin[i], OUTPUT);
    for (i = 0; i < 4; i++) digitalWrite(sel_pin[i], HIGH);
}
```

```

for (i = 0; i < 8; i++) {
    if (i == 6) digitalWrite(seg_pin[i], LOW); // - 가 켜짐
    else digitalWrite(seg_pin[i], HIGH);
}
}

void loop() {
    // put your main code here, to run repeatedly:
    int i, j;
    curr_sw = digitalRead(1);
    if (curr_sw == LOW){
        start = 1;
        stop_r = false;//추가
    }
    if (last_sw == LOW && curr_sw == HIGH) sw_state = ! sw_state; //HIGH -> LOW
    else if (last_sw == LOW && curr_sw == LOW) time_cnt++;
    if (curr_sw == HIGH) time_cnt = 0;
    if (time_cnt >= 500){
        curr_sw = HIGH;
        for(i = 0; i < 4; i++) n[i] = 0;
        cnt = 0;
        stop_r = true;//추가
    }
    last_sw = curr_sw;
    if (start == 1) {//추가
        for (i = 0; i < 4; i++) {
            digitalWrite(sel_pin[i], HIGH);
            for (j = 0; j < 8; j++) {
                if (i == 2 && j == 7) digitalWrite(seg_pin[j], 0);
                else digitalWrite(seg_pin[j], seg[n[i]][j]);
            } delayMicroseconds(2500);

            digitalWrite(sel_pin[i], LOW);
        }
        int a = cnt;
        if (sw_state == LOW && stop_r == false) cnt++; //추가
        // 전체 delayMicroseconds(2500) -> 2.5ms * 4 = 10 ms 1/100초
        if(a != cnt){
            if (cnt % 100 == 0) {
                n[0]++;
                if (n[0] == 10) n[0] = 0;
            }
        }
    }
}

```

```
}
if (cnt % 1000 == 0) {
    n[1]++;
    if (n[1] == 6) n[1] = 0;
}
if (cnt % 6000 == 0) {
    n[2]++;
    if (n[2] == 10) n[2] = 0;
}
if (cnt % 60000 == 0) {
    n[3]++;
    if (n[3] == 6) {
        n[3] = 0;
        cnt = 0; // cnt가 int범위 벗어날 것을 방지하기 위해 초기값으로 되돌림
    }
}
}
}
}
```