



ATmega128A를 이용한 선풍기 제어



ATmega128A, FNDx1, BarLed, SG90 서보모터, 텍트스위치 사용, DC Motor

조원 : 이홍재, 김준태, 한상진, 김경환

목차



1. 목적



2. 기능



3. R&R



4. TIME SCHEDULE



5. 기능 설명



6. 향후 계획



1. 목적



- 학습 과정 중에 습득한 기술 스택의 활용과 개념 설계 검증.

2. 기능



- 팬을 버튼으로 3단계 제어 (정지/느림/빠름), 7세그먼트로 표시
- 서보모터 회전을 버튼으로 2단계 제어 (Bar 3개/4개), LED Bar 8세그먼트로 표시

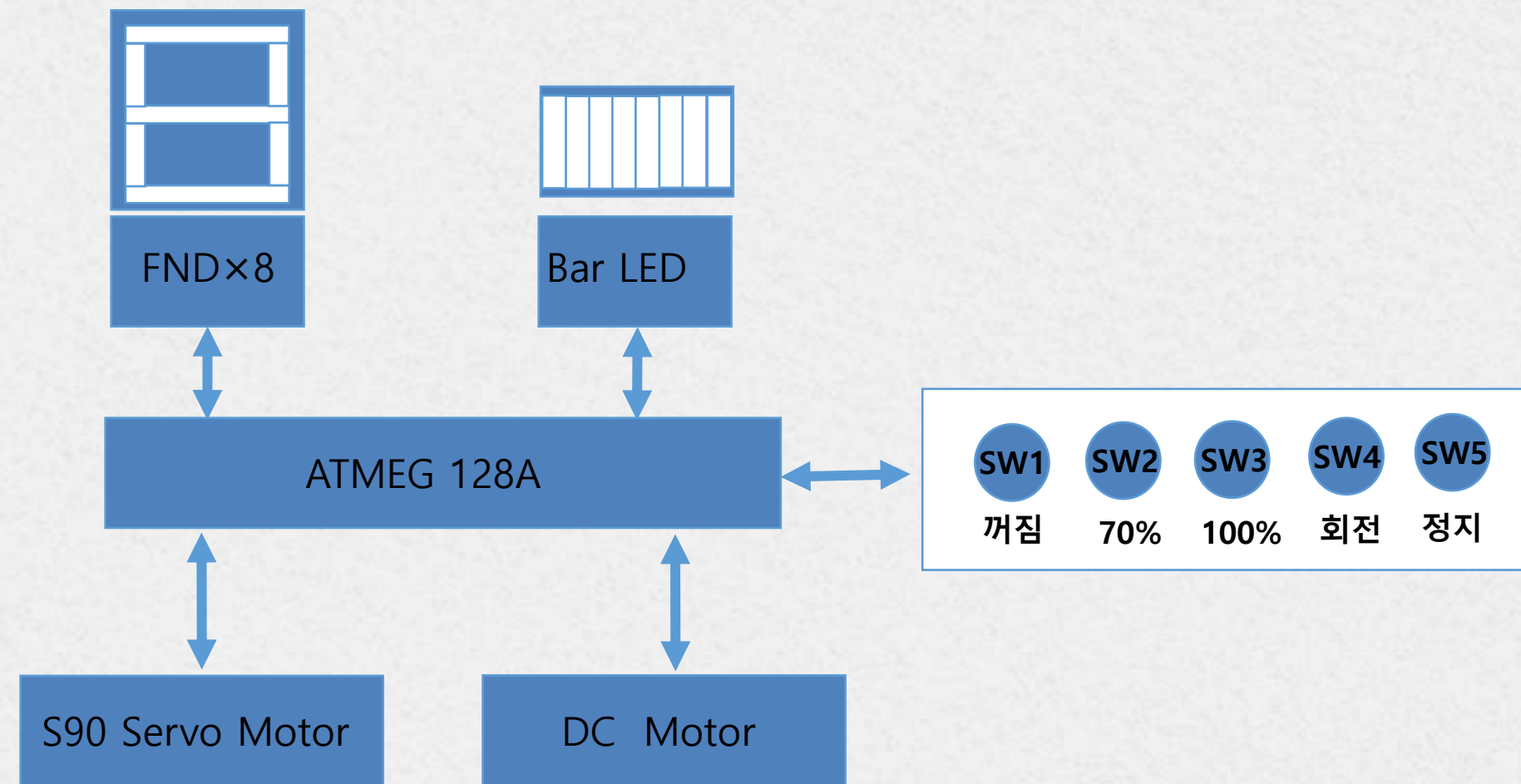
3.R&R



- 한상진:SW 작성 및 검증
- 김경환:PPT 작성
- 이홍재:HW 설계
- 김준태:동작 검증

	2(월)	3	4	5	6	7	8	9(월)	비고
제품 기획 설계팀 구성					휴 무				
설계 및 동작 점검									
품평회									
발표 준비									
발표									

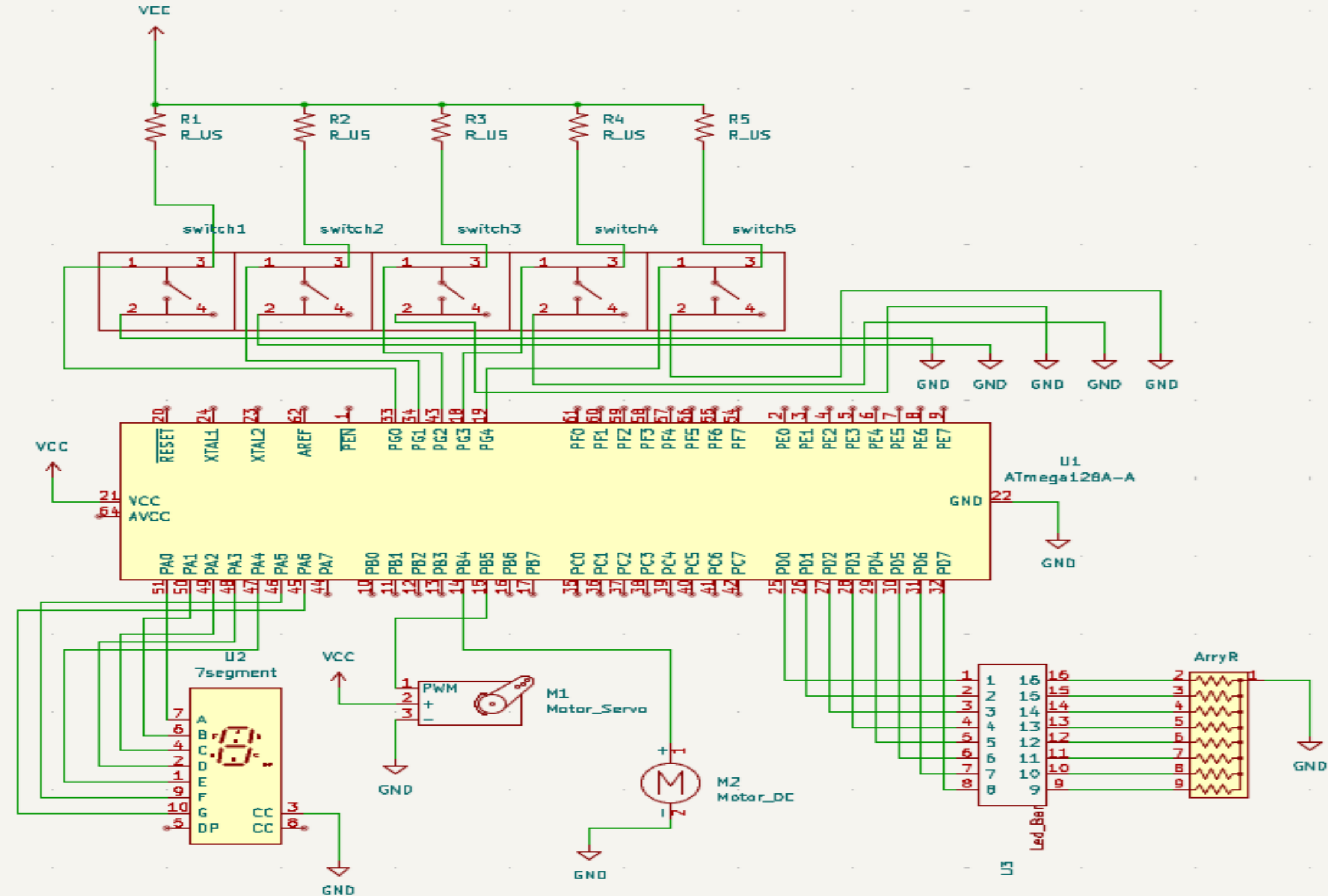
*5. 기능 설명-블록 다이어그램



동작 로직

SW1 FND×8: 0, BarLED×8: ALL전원 OFF
 SW2 FND×8: 1, BarLED×8 : 7EA 전원 ON
 SW3 FND×8: 2, BarLED×8: ALL전원 OFF
 SW4 FND×8: 3, Fan 회전
 SW5 FND×8: 4, Fan 회전 정지

* 5. 기능설명 - 회로도



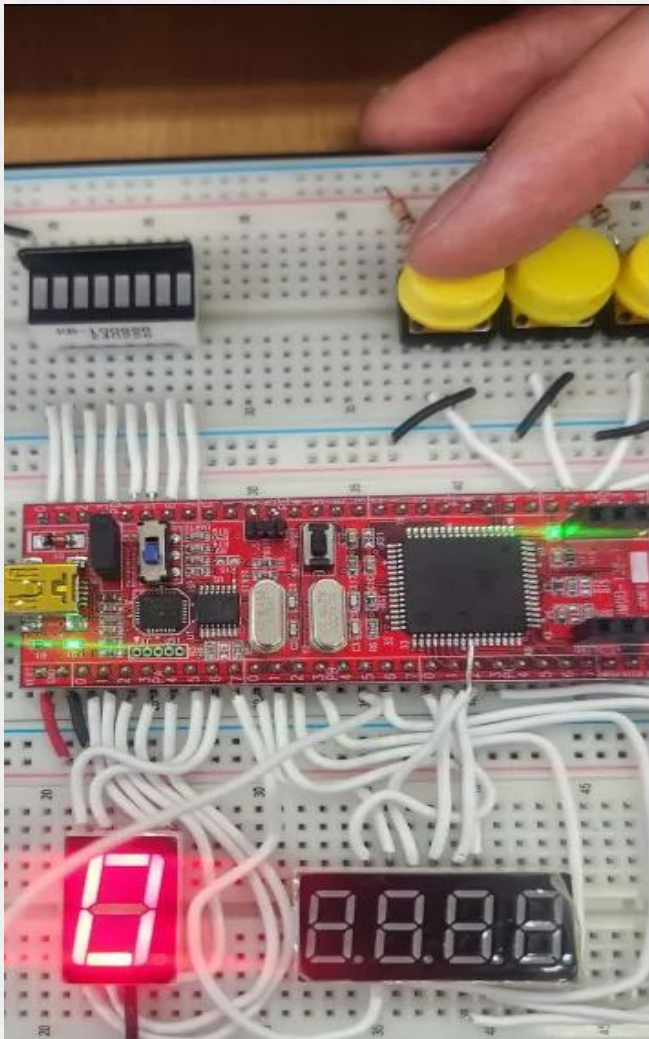
* 5. 기능설명 - 일반

Atmega128a 마이크로컨트롤러로 제어하는 팬 및 서보 모터 제어 SW를 c로 작성, 하드웨어로 구현

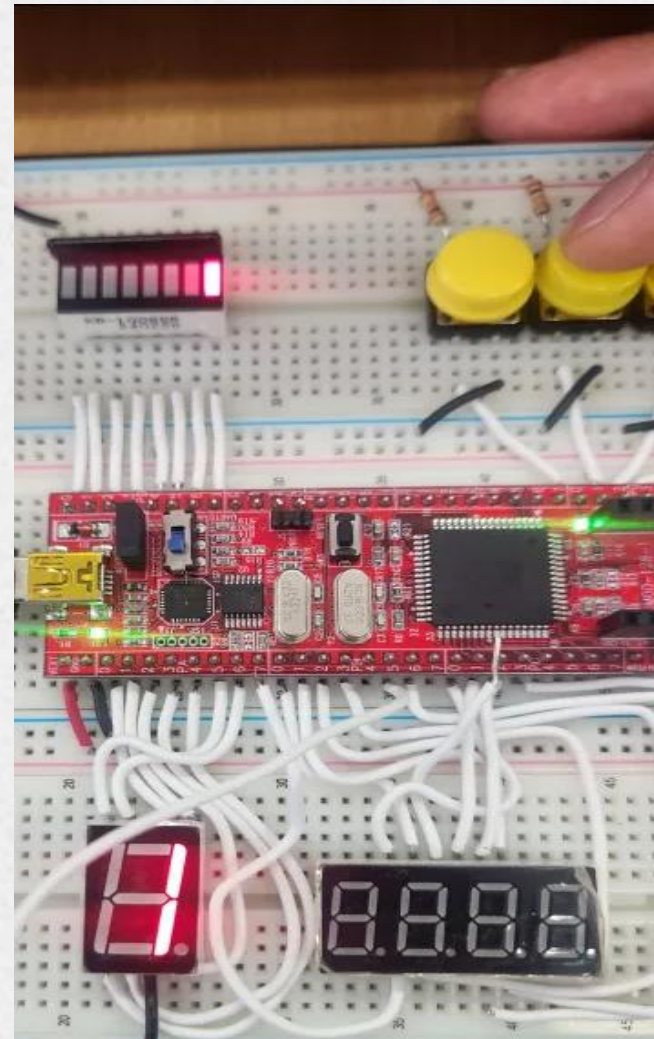
5-2.팬 제어 기능

- 팬 속도를 0%, 70%, 100%로 설정할 수 있음.
- 팬 속도 설정에 따라 FND에 숫자 0, 1, 2 표시.

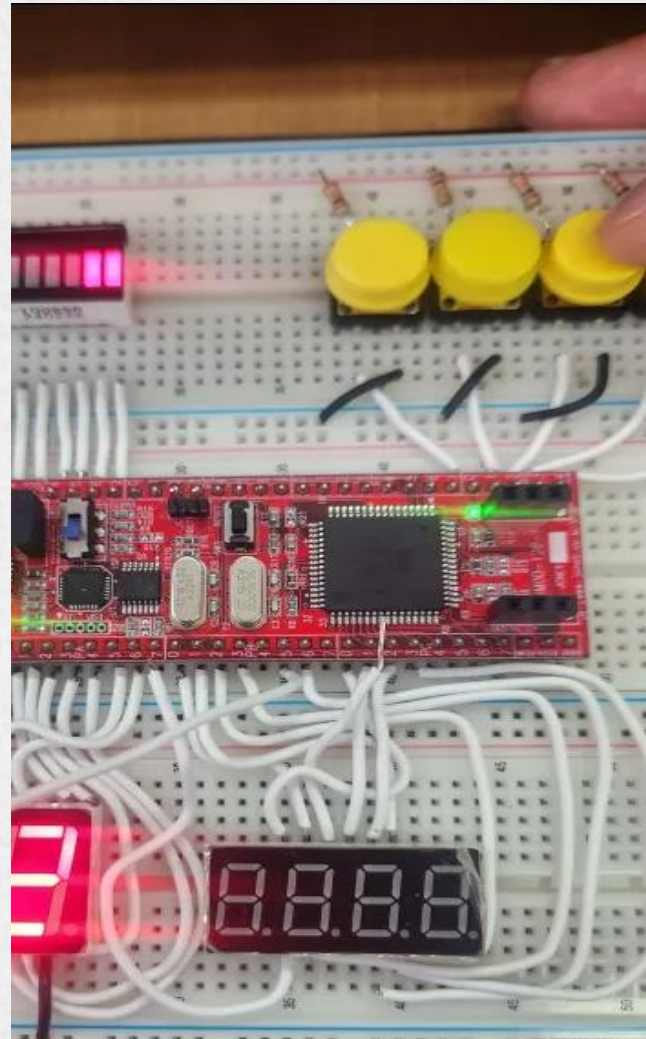
0번 버튼



1번 버튼



2번 버튼



구현코드

```
// 팬 정지 버튼
if (buttonGetState(&btnFanStop) == ACT_PUSH)
{
    PORTD = 0x00; // FND off
    _delay_ms(50);
    PORTA = fndNumber[0]; // 숫자 0 표시
    _delay_ms(50);
    OCR1A = 0; // 서보 정지
    OCR0 = 0; // 팬 정지
    _delay_ms(30);
}

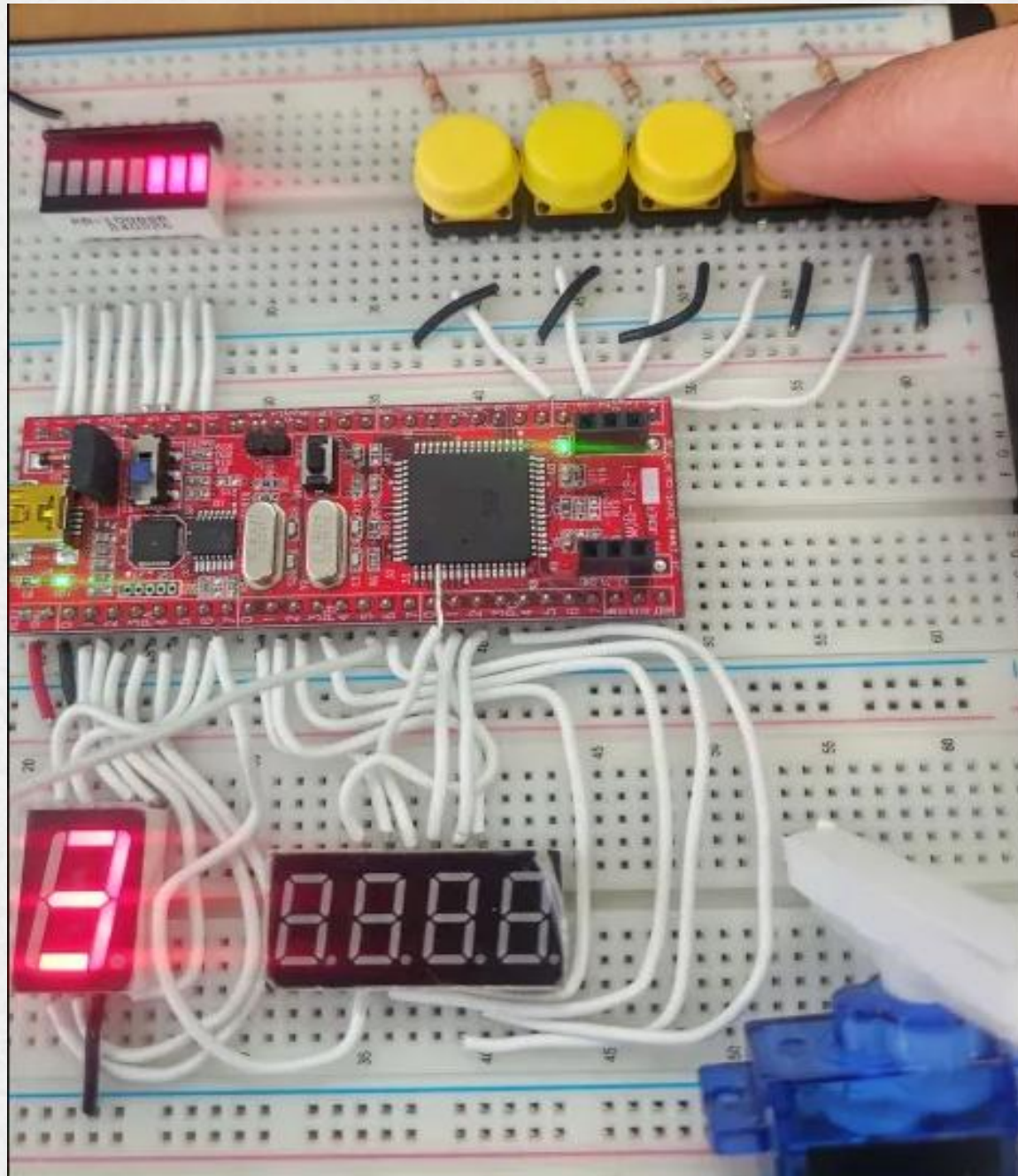
// 팬 저속 버튼
if (buttonGetState(&btnFanSlow) == ACT_PUSH)
{
    OCR0 = 180; // 팬 PWM 70%
    _delay_ms(50);
    PORTD = 0b10000000; // FND 표시
    _delay_ms(50);
    PORTA = fndNumber[1]; // 숫자 1 표시
    _delay_ms(50);
}

// 팬 고속 버튼
if (buttonGetState(&btnFanFast) == ACT_PUSH)
{
    PORTD = 0b11000000; // FND 표시
    _delay_ms(50);
    PORTA = fndNumber[2]; // 숫자 2 표시
    _delay_ms(50);
    OCR0 = 255; // 팬 PWM 100%
    _delay_ms(50);
}
```

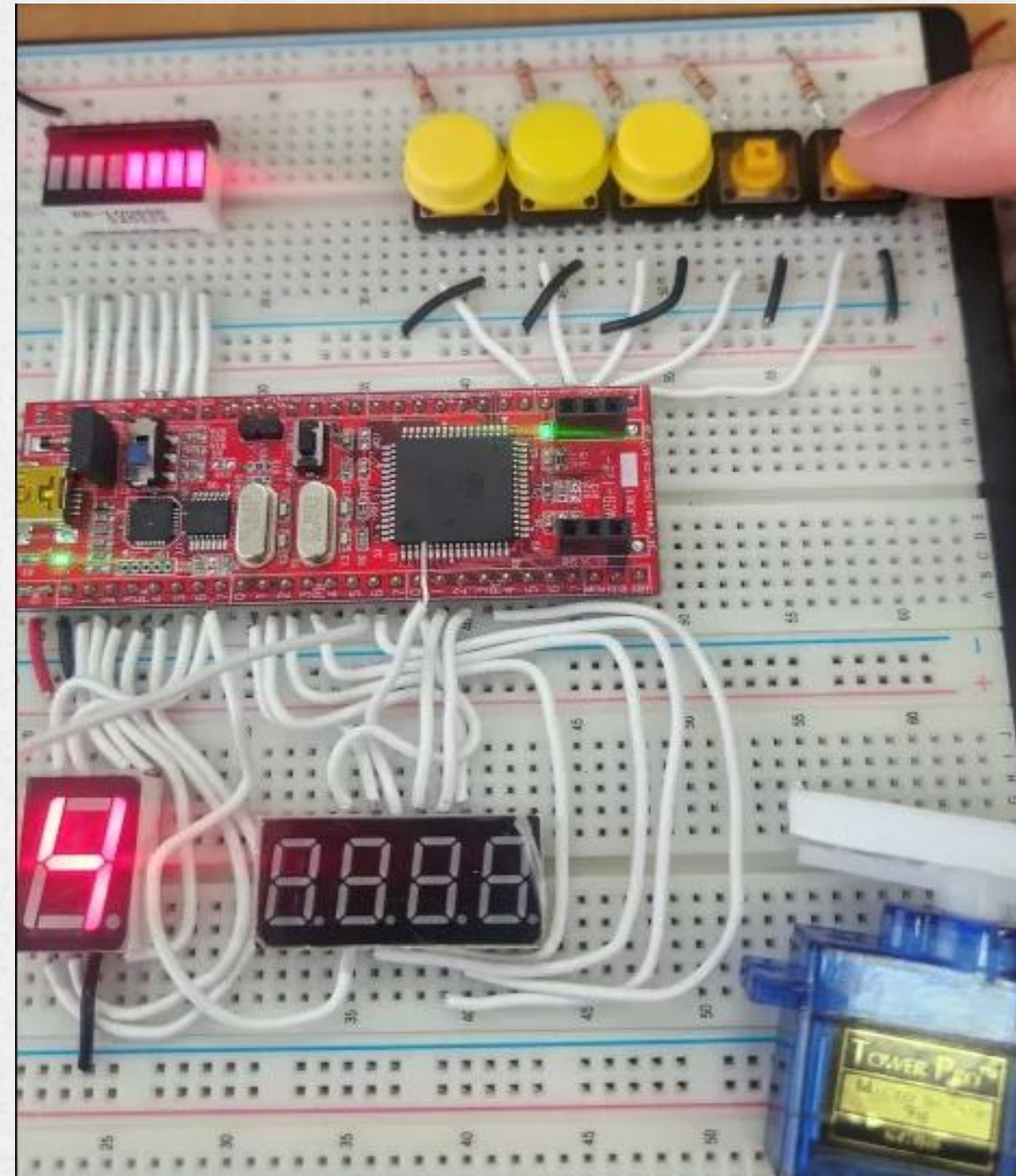

* 5. 기능설명 일반

5-3서보 모터 제어 기능
서보 모터를 작동 및 정지할 수 있음.
서보 모터 작동 중 FND에 숫자 3, 4가 LED BAR 3개, 4개 표시.

3번 버튼



4번 버튼



구현 코드

```
// 서보 동작 시작 버튼
if (buttonGetState(&btnServOn) == ACT_PUSH)
{
    PORTD = 0b11100000; // FND 표시
    _delay_ms(50);
    PORTA = fndNumber[3]; // 숫자 3 표시
    _delay_ms(50);

    servoEnalbed = 1; // 서보 동작 시작
}

// 서보 동작 정지 버튼
if (buttonGetState(&btnServOff) == ACT_PUSH)
{
    PORTD = 0b11100000; // FND 표시
    _delay_ms(50);
    PORTA = fndNumber[4]; // 숫자 4 표시
    _delay_ms(50);

    servoEnalbed = 0; // 서보 동작 정지

    OCR1A = 0; // 서보 PWM 0
    _delay_ms(10);
}

if (servoEnalbed == 1) {
    servoPos += servoDiret;

    if(servoPos >= 635) {
        servoPos = 635;
        servoDiret = -5;
    }
    else if (servoPos < 105) {
        servoPos = 105;
        servoDiret = 5;
    }

    OCR1A = servoPos;

    _delay_ms(10);
}
```

// 서보모터 상태가 1(켜짐)이면
// 서보모터 듀티값을 증감값만큼 더해서 저장(초기값 370 + 5)

// 현재 서보모터 각도가 -90도를 찍었거나 넘었으면
// -90도로 설정해 준 다음
// 증감값 초기(+5)에서 -5로 변경

// 현재 서보모터 각도가 +90도를 넘었으면
// +90도로 설정해 준 다음
// 증감값을 +5로 변경

//위 연산값을 16비트 PWM OC레지스터에 듀티값에 넣고

// 부드러운 연산(회전)을 위해 딜레이 걸어줌

* 5. 기능설명 - 레지스터 제어

(1) 팬 제어 (PWM 제어 - 8bit, OCR0)

타이머0 (Timer0) 사용
PWM 출력을 이용해 팬의 속도 제어
PB4 핀에서 출력
속도 조절:
OCR0 = 0: 정지
OCR0 = 128: 중간 속도
OCR0 = 255: 최고 속도

(2) 서보모터 제어 (16bit PWM - OCR1A)

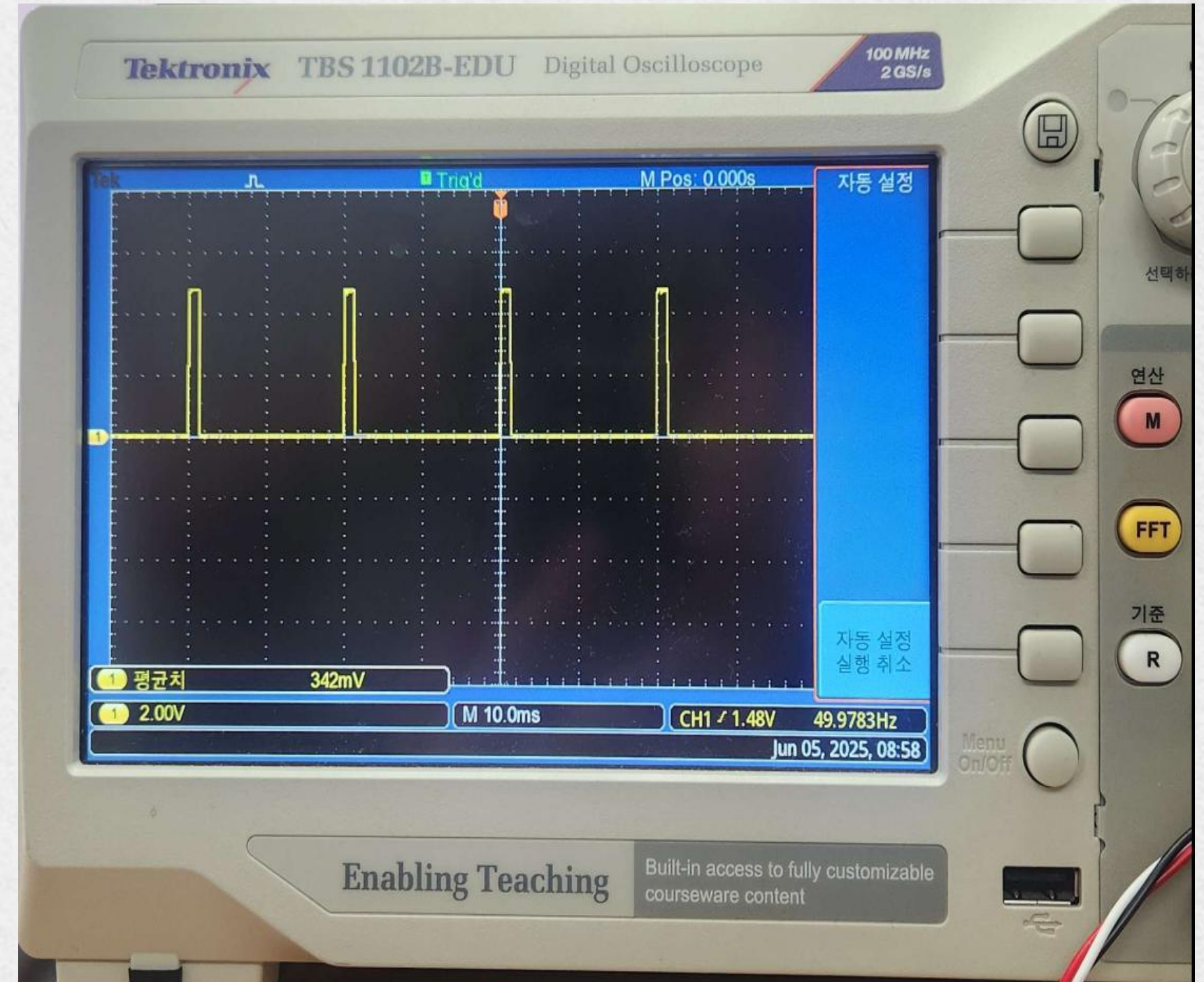
타이머1 (Timer1) 사용
TOP 값 ICR1 = 4999 설정으로 50Hz 주기 설정 (서보 표준)
OCR1A 값을 조절하여 회전 각도 조절
Servo On 상태일 때 OCR1A 값을 반복증가, 감소시켜 회전
Servo Off 시 OCR1A = 0으로 회전 정지

(3) 버튼 제어

버튼 5개 각각 초기화 (buttonInit)
buttonGetState()로 눌림 감지 (ACT_PUSH)
매 버튼 입력마다 관련 동작 수행 및 FND 숫자 갱신, LED Bar 갯수 증가

(4) FND 출력 (7세그먼트 디스플레이)

PORTD: 자리 수 선택 (간단히 전체 포트 출력으로 처리)
PORTA: 표시할 숫자의 세그먼트 코드 (0~9)
숫자 테이블 fndNumber[] 사용해 매핑



* 6. 문제 해결

1. FND 오작동

Solution) build 후에 출력되어야할 숫자가 뜨질 않아서 확인
→ 해당 코드 부분 확인 후 이상 없음을 확인
→ 하드웨어가 잘못되었나 접근하여 부품을 바꿈

2. Servo90 Motor의 작동

Solution) 1번의 FND 처럼 하드웨어가 문제인가 해서
역순으로 해당 서보 모터를 먼저 교체
→ 교체해도 작동 안함을 확인
→ 해당 코드가 for문으로 작성된 것을 if문으로 교체

```
if (buttonGetState(&btnFour) == ACT_PUSH)
{
    PORTD = 0b11110000;
    _delay_ms(50);
    PORTA = fndNumber[4];
    _delay_ms(50);

    // 회전 반복문
    for (int i = 0; i < 500; i++)
    {
        for (int j = 0; j < 500; j++)
        {
            OCR1A = j;
            _delay_ms(10);
        }
        if (buttonGetState(&btnStop) == ACT_PUSH)
        {
            PORTD = 0b00000000;
            _delay_ms(50);
            // 1. 숫자 0으로 표기
            PORTA = fndNumber[0];
            OCR1A = 0;
        }
    }
}
```




* 향후 계획

리모콘 기능	IR 리시브 센서를 이용한 리모콘 기능
자동 중지모드	자이로센서를 이용한 선풍기 넘어질 시 자동 중지
블루투스	HC-06 블루투스 Slave 모듈을 이용한 블루투스 기능

*QnA

THANK YOU
