

2021113490 이현서  
2021111808 박태영

## 아두이노를 통한 선풍기 제작

## <목차>

1. 선풍기 제작 배경
2. 회로 구성 설명
3. 소스코드 설명
4. 느낀점

## <선풍기 제작 배경>

버튼에 따라 미풍, 약풍, 강풍이 달라지는 기존 선풍기의 기능을 착안하여, 우리도 이런 선풍기의 기능을 비슷하게 구현하고자 하는 목표를 가지고 만들게 되었다.

## <회로 구성 설명>

포텐시오미터를 활용해 미풍(빨), 약풍(노), 강풍(초) 3단계로 나누어 모터가 작동해 선풍기가 돌아가도록 하였고, 다른 단계가 될 때마다 해당 단계의 LED가 켜지고, 또한 부저가 울리도록해서 단계가 바뀌었다는 신호를 시각적 신호는 물론, 청각적 신호까지 들을 수 있도록 하였다. 게다가 부저의 소리 신호도 각 단계마다 주파수를 다르게 해 소리의 높낮이를 구분할 수 있도록 하였다.

아두이노 우노 보드에 스텝 모터 드라이버와 연결해 스텝 모터를 조절하도록 하고, 브레드보드에 포텐시오미터와 LED 3개, 부저를 연결해 아두이노 우노 보드와 연결해 작동하도록 하였다.

## <소스코드 설명>

```
// 스텝모터를 활용하기 위한 헤더 파일
#include <Stepper.h>

// GREEN, YELLOW, RED색의 LED를 각각 8, 9, 10번 핀에 연결하기 위한
// 전처리 코드
#define GREEN 8
#define YELLOW 9
#define RED 10

// 포텐시오미터를 아날로그 핀인 A0에 연결하기 위한 전처리 코드
#define PO A0

// 스텝 모터를 돌아가게 하기 위한 IN1, IN2, IN3, IN4핀을 정의한 전처리 코드
#define IN1 2
#define IN2 3
#define IN3 4
#define IN4 5

// 모터의 속도에 따른 효과음을 내주기 위해 적절한 Note의 높낮이를 각각의
```

```

// NOTE_RED, NOTE_YELLOW, NOTE_GREEN음을 RED, YELLOW, GREEN색의 LED에 대응시켜주는 전처리
코드

// 숫자가 높아질 수록 음이 높아지게끔 (속도가 높아지면 음이 높아지게끔 처리)
#define NOTE_RED 600
#define NOTE_YELLOW 650
#define NOTE_GREEN 700

// 음을 얼마나 지속시켜줄지 정해주는 DURATION전처리기
#define DURATION 100

// 부저를 디지털 13번 핀에 연결하기 위한 전처리 코드
#define BUZZER 13

// 2048을 스텝모터에서 1바퀴로 표현하기 위해 정한 전처리 코드
#define STEPS 2048

// 또한 포텐시옴터의 MAX 값인 1024를 3으로 나누어 미풍 약풍 강풍을 나눌 수 있게끔
// threshold(한계점)을 정해줌
int threhole = 1024 / 3;

// 스텝 모터 객체 인스턴스를 만들기 위한 생성자 코드를 작성해 준다
// 즉 스텝모터 인스턴스를 myStepper에 넣음
Stepper myStepper = Stepper(STEPS, IN4, IN3, IN2, IN1);

// checkBefore는 미풍 약풍 강풍에서 바람의 세기를 조절 할 때에 다른 바람 세기로
// 넘어왔다는 것을 알 수 있게끔 하는 일종의 디버깅 변수
int checkBefore = 0;

void setup() {
    // 디버깅을 하기 위해 Serial(신호)를 연결
    Serial.begin(9600);
    // RED(10), YELLOW(9), GREEN(8) 핀을 출력(OUTPUT)핀으로 설정해 준다.
    pinMode(RED, OUTPUT);
    pinMode(YELLOW, OUTPUT);
    pinMode(GREEN, OUTPUT);
}

// 반복문
void loop() {
    // A0에 연결된 포텐시옴터의 값을 아날로그 값으로 받아 들인 다음에 value에 저장한다.
    int value = analogRead(A0);

    // 포텐시옴터의 값을 표시 하기 위한 디버깅 코드
    Serial.println(value);
}

```

```

// 포텐시오미터의 값에 따라 LED를 상이하게 키고 끌 수 있게 하는 조건문을 작성한다.

// CASE1: 첫번째 한계값 ( 1 * threhole ) -> 미풍
if(value <= threhole) {

    // checkBefore확인 변수를 활용하여 checkBefore != 0 즉 checkBefore를 0으로 처리하지 않았다면
    // ( 0으로 초기화 되지 않았다는 의미는 다른 바람 세기에서 넘어오지 않았다는 것 (미풍 -> 미풍이 아님)을
    의미 하므로 소리를 낸다.)
    if(checkBefore != 0) {
        // tone을 통해 부저를 NOTE_RED에 맞는 소리를 울린다.
        tone(BUZZER, NOTE_RED, DURATION);

        // checkBefore변수를 0으로 초기화 한다.
        checkBefore = 0;
    }
    // myStepper객체를 이용하여 speed(속도)를 2로 설정한다.
    myStepper.setSpeed(2);

    // RED색의 LED를 키고 YELLOW색의 LED를 끄고 GREEN색의 LED를 끈다.
    digitalWrite(RED, HIGH);
    digitalWrite(YELLOW, LOW);
    digitalWrite(GREEN, LOW);

    // 또한 STEPS / 10을 한 이유는 CASE1의 경우인 미풍의 경우는 속도가 매우 느리기 때문에
    // 그 돌아가는 거리를 적게 하여 바로바로 미풍, 약풍, 강풍으로 넘어갈 수 있게끔 설정하는 것이다.
    myStepper.step(STEPS / 10);
    // CASE2: 두번째 한계값 ( 2 * threhole )
} else if(value <= 2 * threhole) {
    // (약풍 -> 약풍)이 아님을 확인해 주는 코드
    if(checkBefore != 1) {
        // 만약 다른 세기에서 약풍으로 들어왔다면 소리를 내준다.
        tone(BUZZER, NOTE_YELLOW, DURATION);

        // 또한 checkBefore의 값을 1로 변경해 준다.
        checkBefore = 1;
    }
    // myStepper객체를 이용하여 speed(속도)를 5로 설정한다.
    myStepper.setSpeed(5);

    // RED색의 LED를 끄고 YELLOW색의 LED를 키고 GREEN색의 LED를 끈다.
    digitalWrite(RED, LOW);
    digitalWrite(YELLOW, HIGH);
    digitalWrite(GREEN, LOW);

    // 미풍보다는 바람의 세기가 크므로 이 보다는 돌아가는 길이를 증가 시키기 위해 5로 나누어 준다.

```

```

myStepper.step(STEPS / 5);

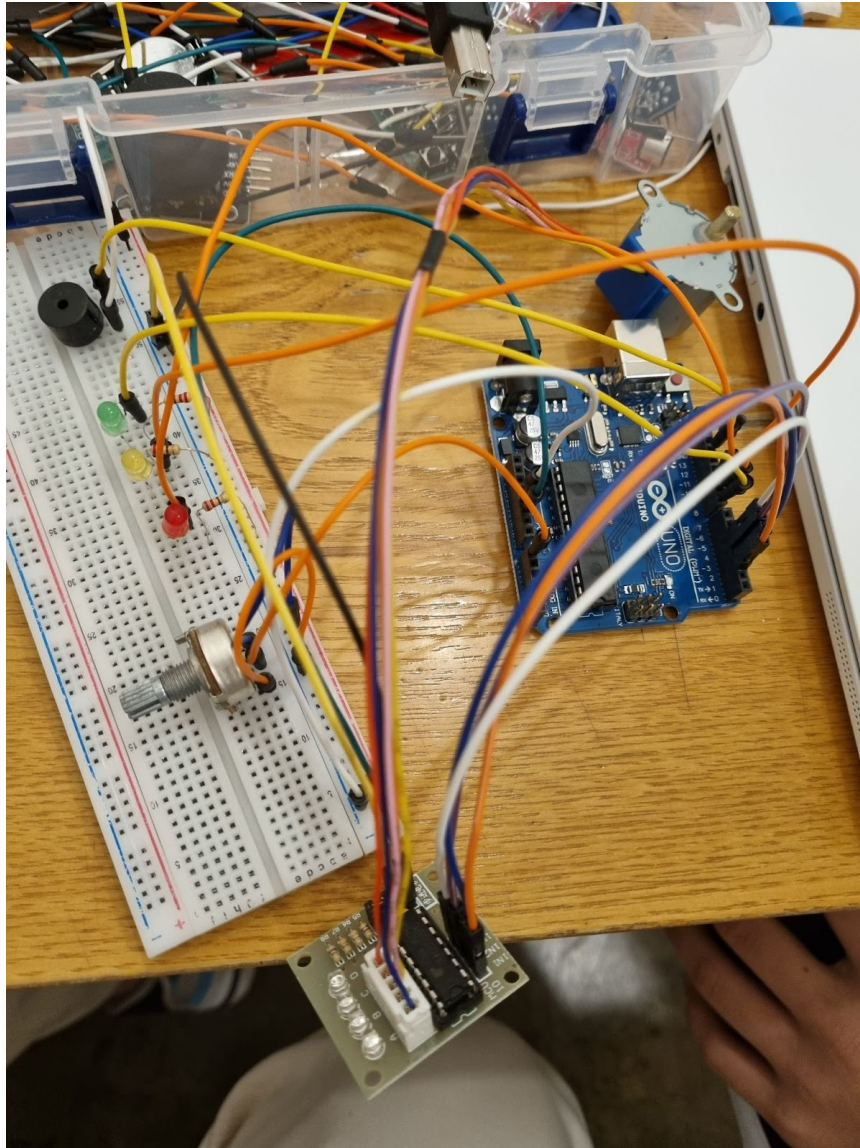
// CASE3: 마지막째 한계값 ( 3 * threhold = 1024)
} else {
    // (강풍 -> 강풍)이 아님을 확인해 주는 코드
    if(checkBefore != 2) {
        // 만약 다른 세기에서 강풍으로 들어왔다면 이에 맞는 소리를 내준다.
        tone(BUZZER, NOTE_GREEN, DURATION);

        // 또한 checkBefore의 값을 2로 변경해준다.
        checkBefore = 2;
    }
    // myStepper객체를 이용하여 speed(속도)를 9로 설정한다.
    myStepper.setSpeed(9);

    // RED색의 LED를 끄고 YELLOW색의 LED를 끄고 GREEN색의 LED를 켜다.
    digitalWrite(RED, LOW);
    digitalWrite(YELLOW, LOW);
    digitalWrite(GREEN, HIGH);

    // 약풍보다는 바람의 세기가 크므로 이 보다는 돌아가는 길이를 증가 시키기 위해 3으로 나누어 준다.
    myStepper.step(STEPS / 3);
}
}

```



## <느낀점>

### 박태영

: 저번주에 새로 받아본 아두이노 키트로 선풍기라는 실생활 해결 문제를 받고, 어떻게 구현해야할지 고민을 하는 시간을 가졌고, 직접 생각하면서 한 단계씩 차근차근 해결하니 비교적 순조롭게 완성을 하였다. 중간에 코드가 잘 작동하지 않아 디버깅 시간을 길게 가졌지만 근본적으로 생각하며 접근하여 해결을 할 수 있었다.

### 이현서

: 강의때 처음 써본 아두이노를 통해 어떠한 결과물을 처음 만들어 보았습니다. 처음에는 생소했던 아두이노를 통해 의미있는 결과물을 만들어내니 이에 대한 쾌감은 말로 표현할 수 없었습니다. 또한, 친구와 협력해서 코드를 짜고 회로를 구성하고나니 원초적인 코드를 작성하고 디버깅하는 작업보다 뭔가 더 의미있고 재미있었던 것 같습니다.