**Lab4. 아날로그 입력 및 라이브러리 작성**

# **예제1. 가변저항수치에 따른 LED 발광주기 조절하기**

|  |  |  |
| --- | --- | --- |
| **구분** | **설명** | **코드** |
| 핀 정의  핀 세팅 | 핀 번호를 LED\_BULITIN으로 정의합니다.  핀 번호를 LED\_SENSOR로 정의합니다.  빌트인 LED를 출력으로 사용합니다. | Int LED\_BULITIN = 13;  Int sensorPin=0;  Void setup(){  pinMode(LED\_BUILTIN, OUTPUT)  } |
| 루프 | 프로그램의 동작을 루프문 안에 기술합니다.  센서값을 정의합니다. | void loop(){  int sensorValue;  sensorValue = analogRead(sensorPin) |
|  | 센사값에 입력된 시간만큼 delay가 발생합니다. | digitalWrite(LED\_BUILTIN, HIGH);  delay(sensorValue); |
|  |  | digitalWrite(LED\_BUILTIN, HIGH);  delay(sensorValue); |
|  |  | } |

**예제2. 라이브러리 분리하기,**

|  |  |  |
| --- | --- | --- |
| **구분** | **설명** | **코드** |
| 헤더파일 생성  변수와 함수 선언  cpp파일 생성  생성자 생성 | arduino라이브라리를 호출하여 아두이노의 함수를 사용합니다.  클래스에서 사용할 변수와 함수를 선언합니다.  헤더파일을 호출하여 필요한 변수를 가져옵니다.  생성자를 만듭니다. | #include<Arduino>  class LED13{  public:  LED13():  void on();  void off();    #include “LED13.h”  LED13::LEC13(){  pinMode(LED\_PIN,OUTPUT) |
| 함수 생성 | 함수에 필요한 동작을 설정합니다. | Void LED13::on(){  digitalWrite(LED\_PIN,HIGH);  } |
|  |  |  |

=========================================================================================

**1.가변저항수치에 따른 LED발광주기 조절하기**

**소스 코드**

=========================================================================================

int sensorPin = 0;

int ledPin=13;

void setup(){

pinMode(ledPin,OUTPUT);

}

void loop(){

int sensorValue;

sensorValue = analogRead(sensorPin);

digitalWrite(ledPin,HIGH);

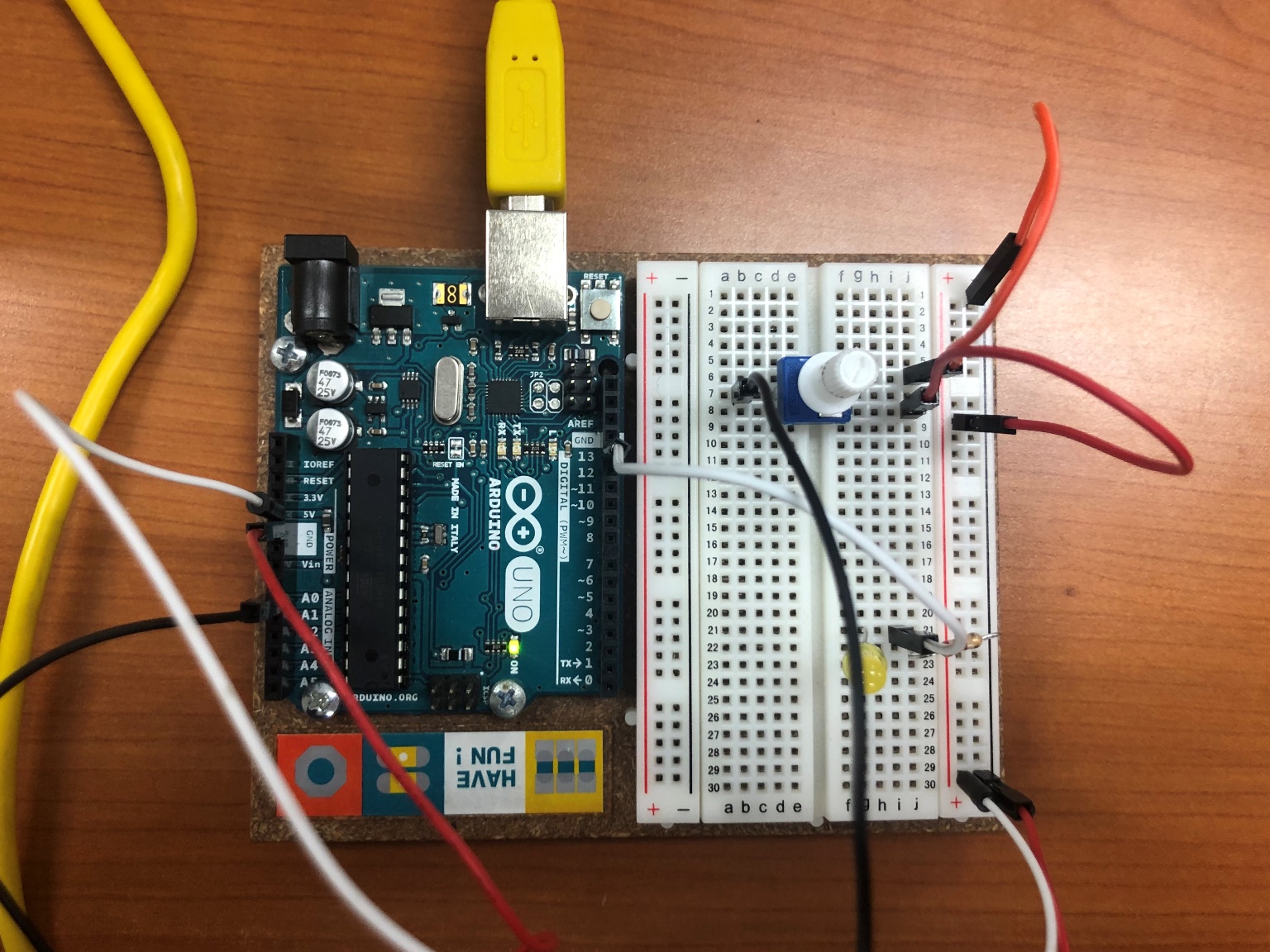
delay(sensorValue);

digitalWrite(ledPin,LOW);

delay(sensorValue)

}

=========================================================================================



=========================================================================================

**2.라이브러리 분리하기(GPIO에 대한 초기화 및 입출력)**

**소스 코드**

========================================== LED13.h ======================================

#include <Arduino.h>

class LED13 {

public:

LED13();

void on();

void off();

void blink(int time);

};

============================================LED13.cpp======================================

#include "LED13.h"

const byte LED\_PIN = 13;

LED13::LED13(){

pinMode(LED\_PIN,OUTPUT);

}

void LED13::on(){

digitalWrite(LED\_PIN,HIGH);

}

void LED13::off(){

digitalWrite(LED\_PIN,LOW);

}

void LED13::blink(int time){

on();

delay(time);

off();

delay(time);

}

==============================================Main=========================================

#include<LED13.h>

LED13 led;

void setup(){

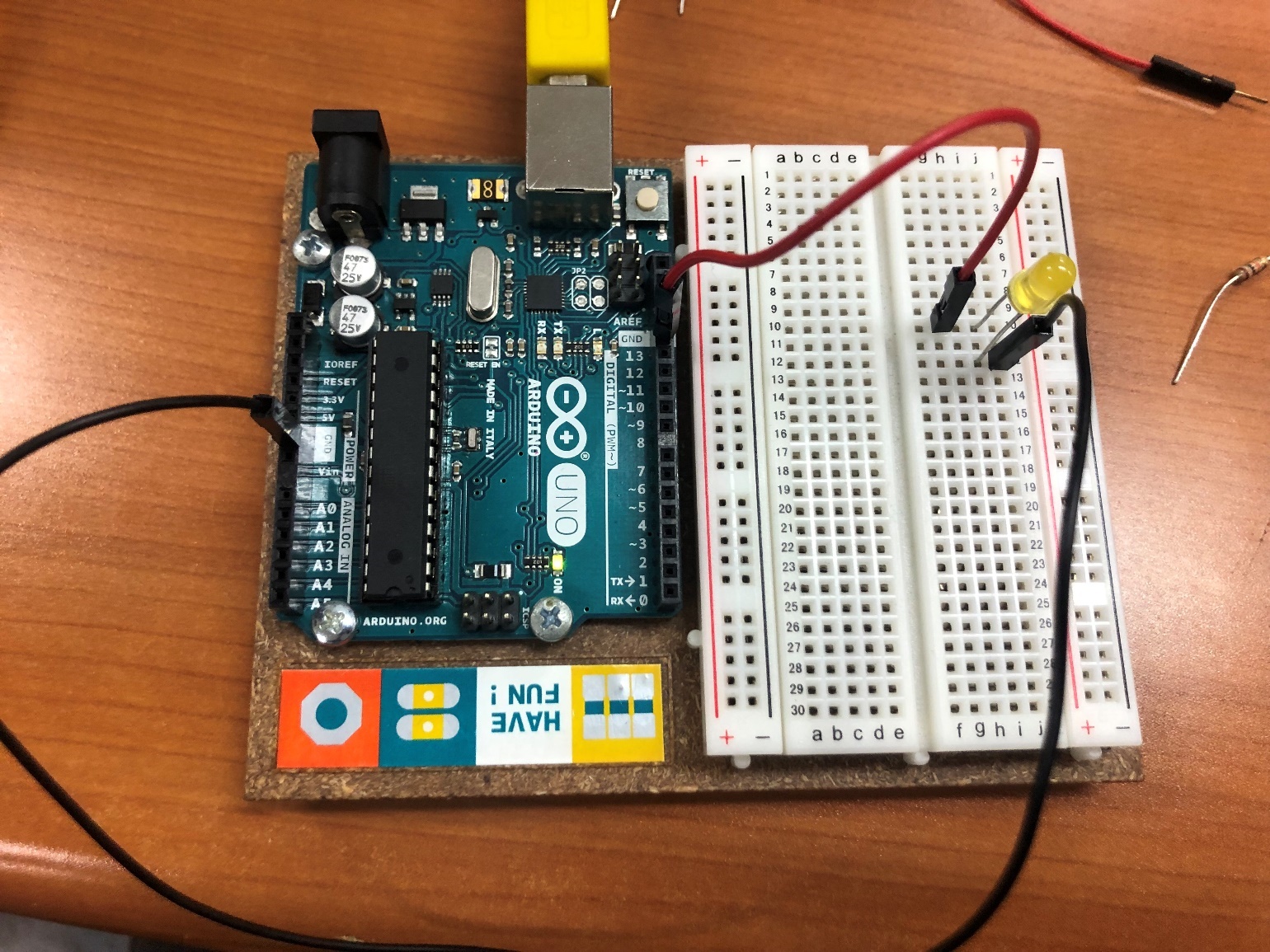
}

void loop(){

led.blink(2000);

}

=========================================================================================



=========================================================================================

**3.가변 저항을 이용한 아날로그 input 값을 이용해서 부저 소리 높낮이 제어하기**

**소스 코드**

=========================================================================================

int sensorPin = 0;

int speakerpin = 13;

void setup(){

}

void loop(){

int sensorValue;

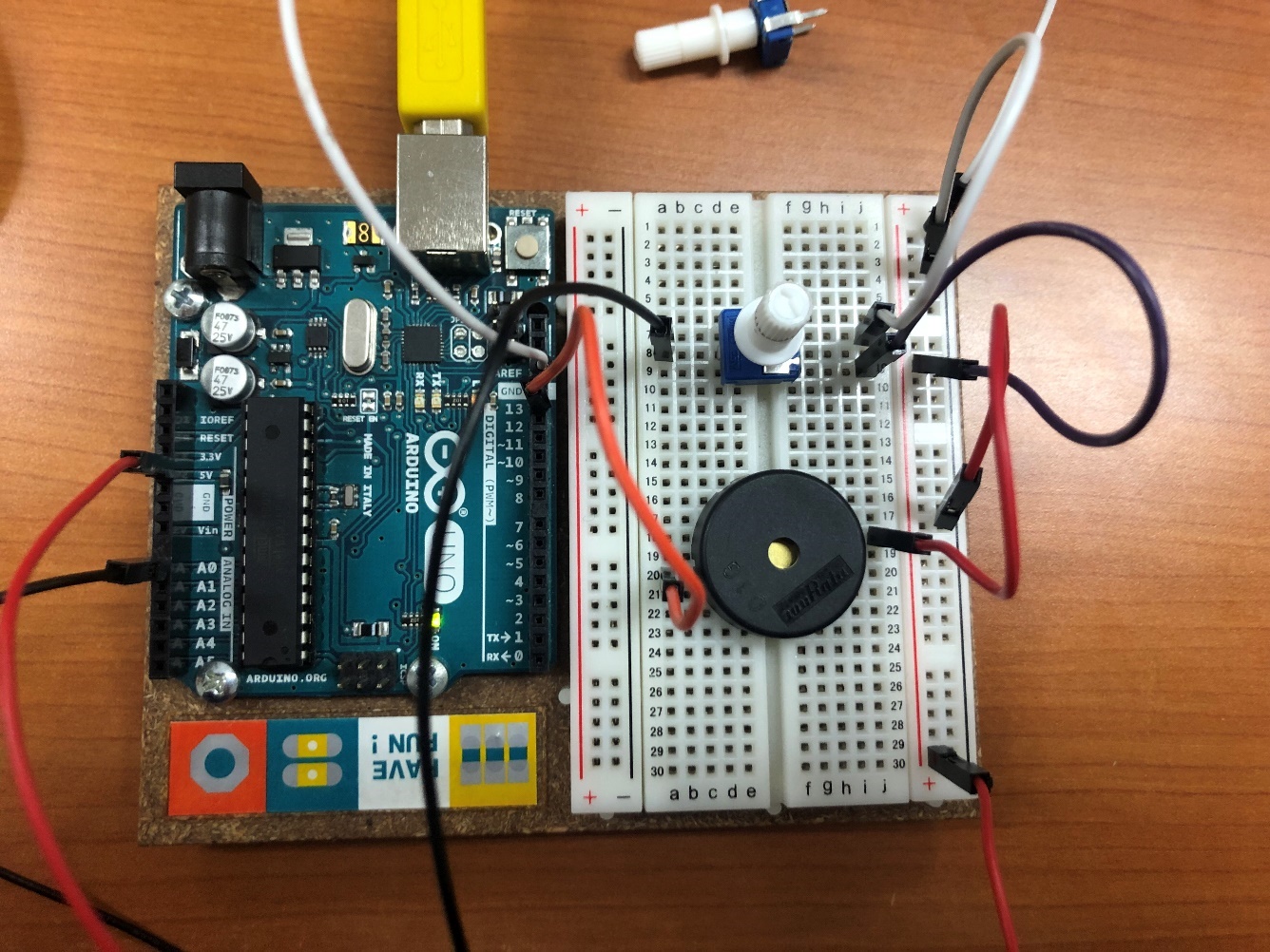
sensorValue = analogRead(sensorPin);

int mapped = map(sensorValue,0,1023,20,4000);

tone(speakerpin,mapped,500);

}

=========================================================================================



|  |
| --- |
| **고찰 및 실험 평가**  **이번 실험은 가변저항을 이용하여 LED핀의 발광 주기를 제어하는 것과 라이브러리를 분리하는 법, 가변 저항을 이용하여 아날로그 input값을 이용해서 부저의 소리 높낮이를 제어하는 3가지 실험을 하였다.**  **먼저, 첫번째 실험에서는 디지털 핀 한 개와 아날로그 핀 한 개를 선언한 후, 디지털 핀은 LED에 , 아날로그 핀은 가변저항에 연결하였고, 아날로그 핀에 입력되는 신호를 가변저항을 통해 가변저항 신호를 조절하게 된다. 그렇기 때문에 가변저항에서 나오는 아날로그 신호를 새변수로 선언한 후, 이를 delay에 값으로 넣게 되면 딜레이 간격이 가변저항에서 나오는 아날로그 신호값에 따라 바뀌게 된다. 두번째 실험애서는 라이브러리를 분리하는 법을 하였는데, cpp에서 사용할 때와 마찬가지로 헤더파일 선언후, 필요한 함수명과 변수명들을 정의하고, cpp파일에서 헤더파일을 불러온 뒤 함수들의 기능들을 선언하였다. 그리고 메인에서 다시 헤더파일을 불러오게 되면 필요한 기능들을 사용할 수 있었다. cpp를 공부했기 떄문에, 아두이노에도 쉽게 적응할 수 있었다. 세번째 실험에서는 가변저항을 이용하여 부저 소리의 높낮이를 제어하는 실험이였다. 부저의 높낮이를 제어하기 위해서는 부저의 주파수와 가변저항의 주파수를 맞춰줘야 했다. 왜냐하면 부저는 20~4000까지 변할 수 있지만 가변저항 값은 0~ 1023까지 변하기 때문이다. 이를 변환 시켜주기 위해서, map함수를 사용하였다. 가변저항 값을 map함수를 이용하여 20~4000까지 바꾼 후 새로운 변수에 저장한다. 그리고 tone함수를 이용하여 변환된 가변저항 값과 주파수를 설정하여 부저의 높낮이를 나타낼 수 있었다. 이번 실험에도 동작하지 않는 핀이 있었기 때문에 실험전에 핀이 제대로 동작하는 지 확인하고 실험했어야 했다. 항상 실험에서 주의하면서 임해야겠다.** |