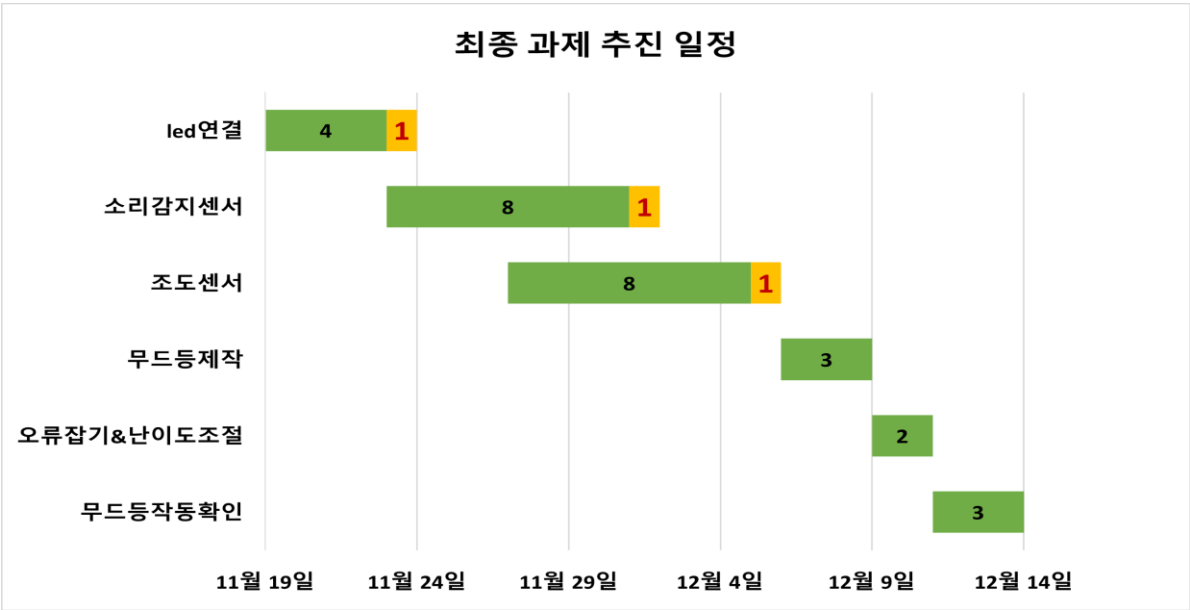


임베디드시스템 이해 및 활용 기말보고서

1. 프로젝트 기획(기획 의도 및 배경)

프로젝트 기획에 앞서 아두이노로 평소 사용할 수 있는 것을 만들면 좋겠다고 생각했다. 나의 일상 생활에서 꼭 필요하고 계속 사용하고 있는 것이 무엇인가 생각해보니 무드 등이었다. 개인적인 생각이지만 무드 등을 잠자기 전에 켜 놓으면 심적으로 안정되고 포근한 느낌을 받았었다. 다양한 센서를 사용하여 나에게 맞춘 맞춤형 무드 등을 제작하면 좋을 것 같아서 이 프로젝트를 기획하게 되었다. 기획한 무드 등의 가장 큰 메리트는 누워서 박수로 빛을 조절할 수 있다는 것이다.

2. 프로젝트 개발 계획 대비 개발상황



A PLAN	B PLAN	C PLAN	D PLAN
RGB LED연결	소리감지센서	조도센서	무드등 제작
100%	99%	100%	99%
4 →5 days	8→9 days	8→9 days	4→3 days

기존에 개발을 계획한 순서는 무드 등 제작, RGB led 연결, 소리 감지 센서 연결, 조도 센서 연결, 오류 잡기 및 난이도 조절, 마지막으로 무드 등 작동 확인하기였다. 기존 계획을 넉넉하게 잡아 계획을 진행하는데 크게 어려움은 없었다. 첫 번째 계획이었던 무드 등 제작은 투명 플라스틱 컵을 사용하여 만들어보았다. 그러나 무드 등의 외적 퀄리티가 매우 떨어지고 빛이 은은하게 퍼질 거라고 생각한 것과 다르게 눈이 부신다는 단점이 생겼다. 따라서 무드 등 제작을 프로젝트 계획 가장 마지막으로 변경하게 되었다.

RGB led 연결과 센서들을 연결하는 데 있어 기존대비 하루가 더 걸렸다. 각각의 센서의 코드를 작성했을 때 오류도 없고 잘 실행되었다. 그러나 3가지를 동시에 실행시키는 과정에서 계속 오류가 나서 이를 해결하기 위해 시간이 조금 더 걸렸다. 무드 등 제작은 육각형 모양의 투명 플라스틱 통을 구매하여 만들었고 모든 센서 작동을 완료시킨 뒤 무드 등과 연결했기 때문에 3일 정도 걸렸다. 기존에 계획했던 오류 잡기는 센서들을 연결하면서 바로바로 수정했기 때문에 기존 계획보다 빠르게 오류를 잡을 수 있었다. 무드 등은 빛에 민감하기 때문에 주변 환경의 빛의 크기를 시리얼 모니터로 확인하고 코드를 수정하면서 무드 등 작동을 확인했다.

3. 프로젝트 소개

이 프로젝트에서 사용된 RGB led는 LED 한 개로 3가지 색을 표현할 수 있다. 3가지 색인 빨강, 초록, 파랑을 잘 조합하면 다양한 색을 낼 수 있다. 색의 양은 전압을 조절해 변경할 수 있다. 소리 감지 센서는 마이크로폰을 사용하여 소리의 크기를 측정하는 센서다. 공기의 진동을 인식해서 전기 신호를 출력하는 장치다. 조도 센서는 주변 밝기를 측정하는 센서다. 극성은 없으나 빛의 양에 따라 전도율이 변하며 주위가 어두울수록 저항값이 증가하여 전기가 적게 통한다.

개발한 프로젝트는 계획에서 정한 순서대로 진행했다. 첫 번째, RGB led의 전압을 조절하여 기존 색을 조합해 노란색, 보라색, 하늘색으로 개발했다. 마지막 색은 전압을 주지 않아 아무런 색이 나타나지 않도록 제작했다. 두 번째, 소리 감지 센서는 박수 소리를 인식하여 RGB led의 색상을 변경해 주도록 개발했다. 박수를 총 4번 치는데 3번은 led의 색상을 변경해주고 마지막 박수에는 led를 꺼지도록 제작했다. 세 번째, 조도 센서 연결은 주변이 어두워졌을 때 자동으로 무드 등을 켜주고 주변이 밝아졌을 때는 자동으로 무드 등이 꺼지도록 개발했다. 마지막으로 무드 등 제작은 기존에 계획했던 플라스틱 컵이 아니라 투명 플라스틱 통을 사용했다. 스티커로 무드 등의 겉을 꾸며주었고 펄 스타핑을 사용해 무드 등의 빛이 시각적으로 예쁘게 퍼지도록 꾸며주었다. 무드 등의 뒷부분을 플라스틱 막이 아닌 종이를 사용해서 RGB led를 무드 등 중앙에 부착 시켜 주었다. 개발한 무드 등의 동작 환경은 어두울 때 (빛 밝기: 700 이상일 때) 작동된다. 즉, 해가 졌을 때나 빛이 없을 때 유용하게 사용할 수 있다.

4. 프로젝트 구현 핵심 코드 설명

```
final_project
int b;

void setup() {
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(blueLed, OUTPUT);
  pinMode(sound, INPUT);
  Serial.begin(9600);
}

void loop() {
  a = analogRead(sound);
  b = analogRead(cds);
  Serial.println(b);

  if(b > 700){
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, HIGH);
    digitalWrite(blueLed, HIGH);
    if(a > threshold){
      count +=1;
      count %=4;
    }
    if( count == 1){
      set(255, 255, 0); //노란색
    }
    else if(count == 2){
      set(255, 0, 255); // 보라색
    }
    else if(count == 3){
      set(0, 255, 255); // 하늘색
    }
    else{
      digitalWrite(redLed, LOW);
      digitalWrite(greenLed, LOW);
      digitalWrite(blueLed, LOW);
    }
  }

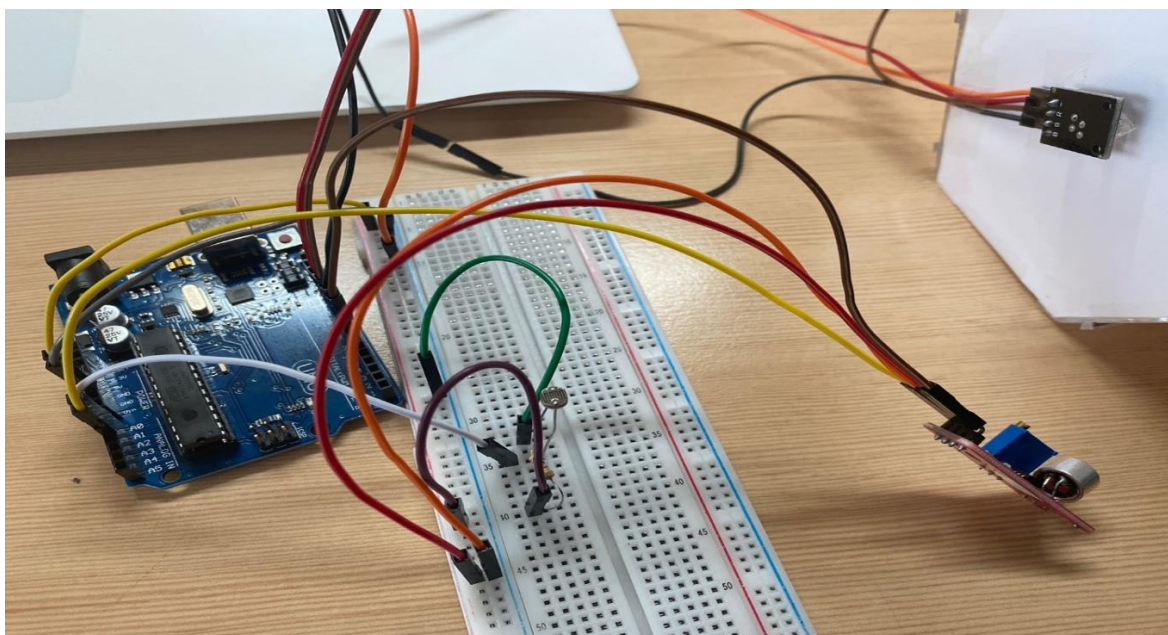
  void set(int R, int G, int B){
    analogWrite(redLed, R);
    analogWrite(greenLed, G);
    analogWrite(blueLed, B);
  }
}
```

RGB led의 red는 PWM 11번, green은 PWM10번, blue는 PWM 9번 핀에 연결해주었다. 소리 감지 센서(sound)는 A1 아날로그 핀에 연결해주었다. 소리 감지 센서의 한계점(threshold)를 100으로 지정하여 박수 소리를 감지하도록 만들어주었다. 조도센서(cds)는 아날로그 핀 A0에 연결했다. 박수에 따른 색을 변경하기 위해 count 변수의 값을 3으로 정했다.

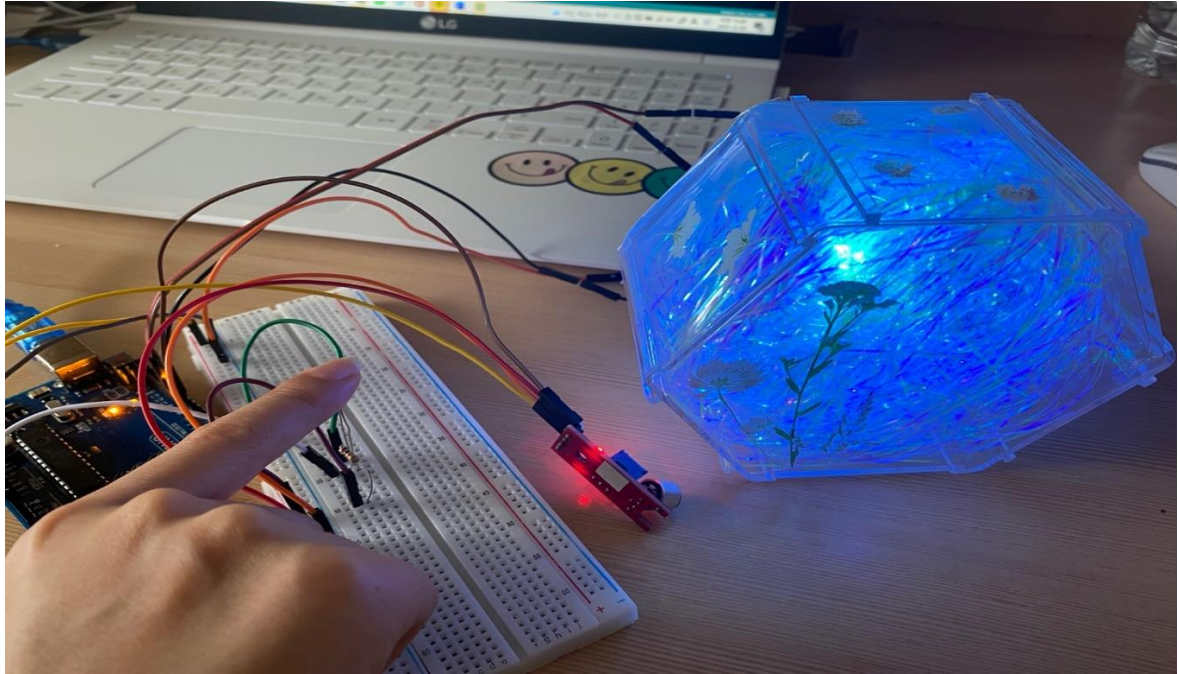
함수 setup에서 소리 감지 센서(sound)를 INPUT, RGB led를 OUTPUT으로 설정해 소리 감지 센서에서 박수 소리가 입력되었을 때 RGB led가 출력되도록 작성했다. 시리얼 모니터에서 센서들이 잘 작동되고 있는지 볼 수 있는 코드를 작성해주었다.

이 프로젝트에서 가장 중요한 부분인 loop 함수를 작성해주었다. 변수 a에는 소리 감지 센서의 아날로그 입력을 저장해주었고 변수 b에는 조도 센서의 아날로그 입력을 저장했다. if 문을 사용하여 센서들이 동시에 작동되도록 만들어 주었다. 시리얼 모니터로 빛 밝기의 크기를 확인하여 적당한 값인 700을 찾아냈다. 조도 센서가 700보다 클 때 RGB led가 켜지도록 디지털 출력을 HIGH로 작성했다. 조도 센서 If 문 속에 소리 감지 센서의 if 문을 작성해주었다. 소리 감지 센서가 지정한 threshold보다 클 때 count를 1씩 더해주어 색을 변경해주고 4로 나누어 다시 처음으로 돌아가도록 코드를 작성했다. Count가 1일 때 (첫 번째 박수일 때) 빨강과 초록색에 255 전압을 주어 노란색을 만들었다. Count가 2일 때 (두 번째 박수일 때) 빨강과 파란색에 255 전압을 주어 보라색을 만들었다. Count가 3일 때 (세 번째 박수일 때) 초록과 파란색에 255 전압을 주어 하늘색을 만들었다. 마지막 박수 즉, 지정한 박수(count)가 아닐 때 전압을 주지 않아 아무 색도 나지 않도록 만들어 주었다. 조도 센서가 700 이상이 아닐 때 꺼지도록 디지털 출력을 LOW로 작성했다. 함수 set에 상수 R, G, B를 지정하여 RGB 핀들이 아날로그 출력되도록 제작했다.

<부품 연결>



<조도센서를 가렸을 때(어두울 때)>



<박수소리로 RGB led 색 변경>



5. 자체평가

이 프로젝트에서 두 센서를 동시에 사용한다는 것은 나에게 큰 도전이었다. 코드를 작성할 때 센서 하나가 작동되면 다른 하나가 작동되지 않아 이 둘을 함께 작동시키기 위한 코드를 작성하는데 가장 많은 시간을 사용했다. 이를 해결하기 위해 많은 예시와 수업자료들을 찾아보면서 사용된 부품들의 작동원리와 기능을 정확하게 파악할 수 있었다. 프로젝트를 진행하면서 수업내용을 다시 한번 복습하게 되는 계기가 되어서 좋았다.

이 프로젝트의 한계점은 소리 감지 센서의 작동이다. 소리가 정확히 감지되지 않아 박수 소리도 제대로 인식하지 못하였다. 따라서 박수를 엄청 크게 쳐야 인식되는 문제점이 생겼다. 소리 감지 센서를 손가락으로 쳐 색을 변경하는 것이 박수로 색을 변경하는 것보다 인식이 잘되는 것을 볼 수 있었다. 편하게 누워서 박수로 무드 등을 작동시키는 것이 이 프로젝트의 최종 목표였는데 소리 감지 센서의 감도가 좋지 못해 센서 가까이에서 큰 박수로 쳐야 한다는 것이 조금 아쉬웠다. 이를 보완하기 위해 다른 고성능 소리 감지 센서를 구매해 사용하거나 다른 센서를 사용한다면 이 문제점을 보완할 수 있을 것 같다. 충분히 다른 센서를 사용할 수 있었지만 기획한 프로젝트를 계획대로 진행하고 싶었기 때문에 소리 감지 센서 그대로 진행하게 되었다.

추후에 이 프로젝트를 발전 시켜 조도 센서와 소리 감지 센서가 아닌 다른 센서나 부품으로 다른 맞춤형 무드 등을 만들어 보고 싶다. 이번에는 피에조 부저와 터치센서를 사용하여 노래가 나오는 무지개 무드 등을 제작하고 싶다. RGB led가 조도 센서로 인해 어두워졌을 때 자동으로 켜지도록 만들어 준다. 그 후 무지개색 순서로 색이 서서히 변하도록 제작하고 터치센서를 눌렀을 때 지정한 노래가 나오도록 추후 발전 계획을 세워보았다.