



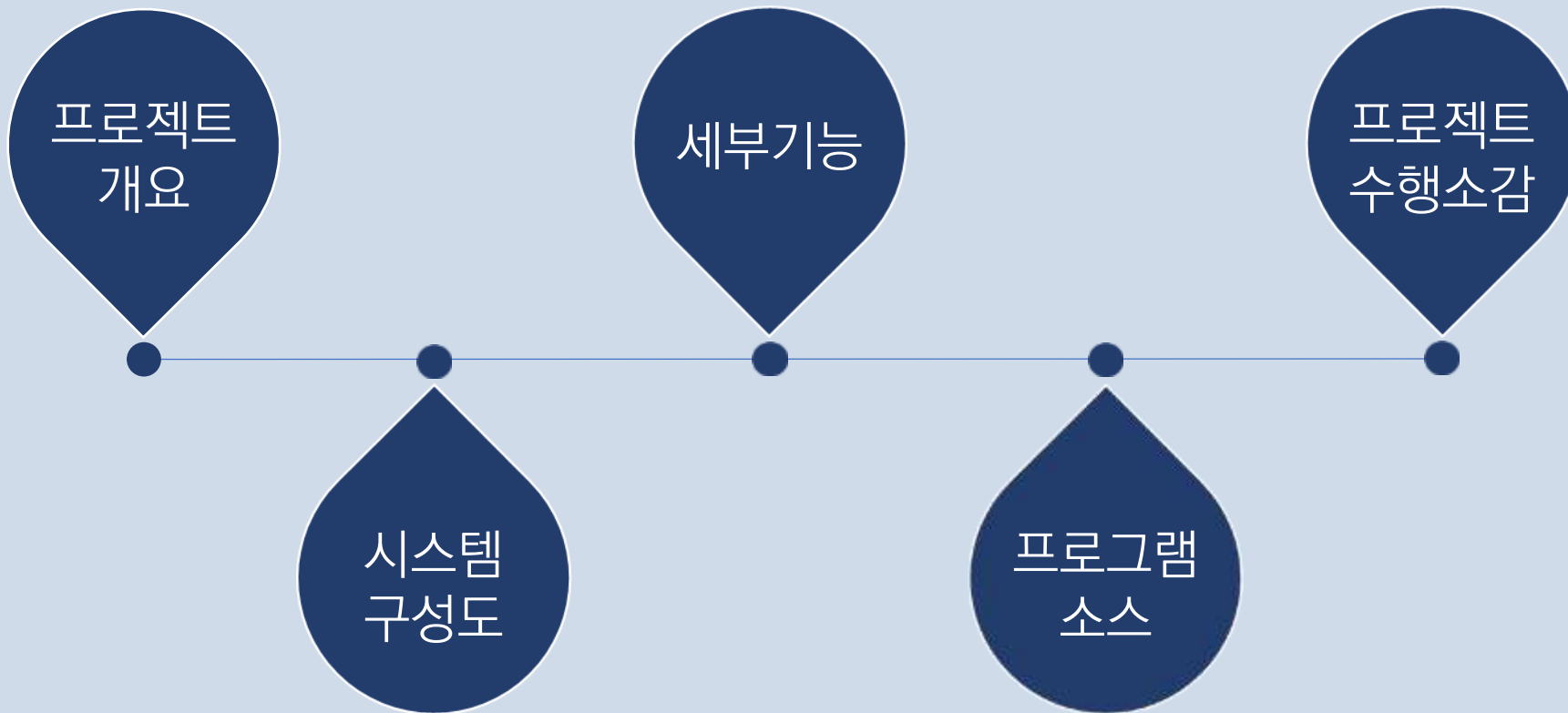
화재 감지기

센서활용 프로그래밍

20180646 구혜진



목차



프로젝트 개요

화염 감지 센서를 이용한 화재 감지기 구현



시스템 구성도

1. 필요한 부품

		
아두이노 UNO	브레드보드	화염 감지 센서
		
Serial I2C LCD	330옴 저항 x4	수동 부저
		
LED x3		온습도 센서

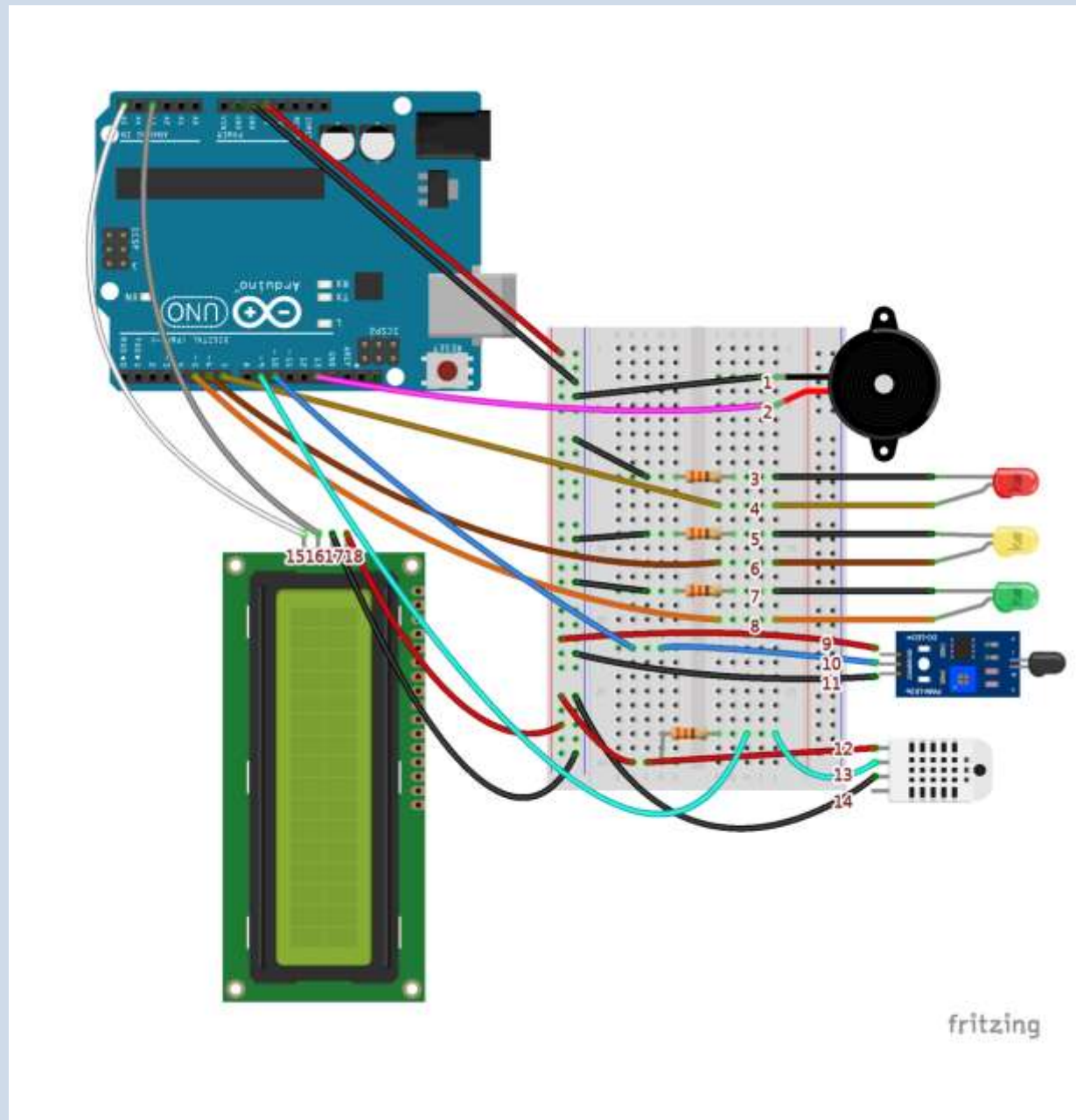
시스템 구성도

2. 배선도

부품		UNO	부품		UNO
부저	1	GND	화염 센서	9	5V
	2	13		10	10
적색 LED *	3	GND		11	GND
	4	7	온습도 센서 *	12	5V
황색 LED *	5	GND		13	9
	6	6		14	GND
녹색 LED *	7	GND	LCD	15	A5
	8	5		16	A4
				17	GND
				18	5V

※ LED는 보드에 직접 꼽아도 됨

※ 저항이 필요한 부품은 * 표시



세부기능

1. setup()



구동 화면 1

로딩 화면 처럼 “*” 특수문자가 3바퀴를 돈다.



구동 화면 2

멜로디가 울린 후, 화면이 뜬다.

세부기능

2. loop() - 1



평소(~45 ° C)
녹색 LED가 켜지고,
“It's safe.” 문구가 뜬다.



에러 화면

세부기능

3. loop() - 2



일정 온도 이상 올라갔을 때($45.1 \sim ^\circ \text{C}$)
노란색 LED가 켜지고, “Be careful.” 문장과
“Warning!” 문장이 번갈아가며 뜬다.

화재 감지

빨간색 LED가 켜지고, “!
SIREN !” 문장과 “It's
dangerous” 문장이 뜬다.

세부기능

4. 영상



프로그램 소스

```
#include < pitches.h> // 멜로디 라이브러리(슬라이드 13번)
#include < Wire.h>
#include < LiquidCrystal_I2C.h> // lcd 라이브러리
#include < DHT11.h> // 온습도 센서 라이브러리
#define PI 3.14592 // 사이렌 소리에 필요한 변수 정의

LiquidCrystal_I2C lcd(0x27, 16, 2); // lcd 핀
Int Greenled = 5;
Int Yellowled = 6;
Int Redled = 7; // led 핀
Int DHT = 9;
DHT11 dht11(DHT); // 온습도 센서 핀
int flame = 10; // 화재 감지 센서 핀
int buzzer = 13; // 부저 핀
int fire = 0; // 화재 감지 센서 상태
int melody[] = {NOTE_G5, NOTE_C6, NOTE_E6, NOTE_C6,
NOTE_D6, NOTE_G6};
int noteDus[] = {8,8,8,8,8,8}; // setup 시작음
float sinVI;
int VI; // 사이렌 소리 관련 변수
int bright, col, i; // for문에 사용할 변수
bool blinking = true; // lcd 커서 상태
int mintemp = 45; // 최소 온도
```

```
void setup() {
    // put your setup code here, to run once:
    pinMode(Greenled, OUTPUT);
    pinMode(Yellowled, OUTPUT);
    pinMode(Redled, OUTPUT);
    pinMode(flame, INPUT);
    pinMode(buzzer, OUTPUT); // 입출력 모드 지정

    lcd.init();
    lcd.noBacklight();
    delay(1000);
    lcd.backlight();
    delay(1000); // lcd 백라이트 꺼졌다 켜짐

    for(int i = 1; i<=3; i++) {
        for(col = 0; col <= 15; col++) {
            lcd.setCursor(col,0);
            lcd.print("*");
            delay(50);
            lcd.clear();
        }
        for(col = 15; col >= 0; col--) {
            lcd.setCursor(col,1);
            lcd.print("*");
            delay(50);
            lcd.clear();
        }
    } // *가 시계방향으로 3바퀴 도는 반복문
```

프로그램 소스

```
for (int thisNote = 0; thisNote < 6; thisNote++) {
    int noteDu = 1000 / noteDus[thisNote];
    tone(buzzer, melody[thisNote], noteDu);

    int pauseBetweenNotes = noteDu * 1.30;
    delay(pauseBetweenNotes);
    noTone(buzzer);
} // 시작음 반복문

lcd.setCursor(7,0);
lcd.print("Hi");
lcd.setCursor(6,1);
lcd.print("@_@/");
delay(3000); // 인사(시작 화면)

lcd.backlight();
}

void loop() {
    // put your main code here, to run repeatedly:
    int err; // 에러 변수
    float temp, humi; // 온습도 변수 (= double)
    lcd.noCursor();
    fire = digitalRead(flame);
```

```
if(blinking) {

    if((err=dht11.read(humi, temp))==0) {
        lcd.clear();
        lcd.setCursor(2,0);
        lcd.print(temp, 1); // 소수점 1자리까지 보임
        lcd.print("C");
        lcd.setCursor(9,0);
        lcd.print(humi, 1);
        lcd.print("%");

        if(temp <= mintemp) { // ~ 45도까지의 상태
            digitalWrite(Greenled, HIGH);
            lcd.setCursor(1,1);
            lcd.print("It's safe. ^-^");
        } else digitalWrite(Greenled, LOW);

        if(temp > mintemp) { // 45도가 넘었을 때의 상태
            digitalWrite(Greenled, LOW);
            digitalWrite(Yellowled, HIGH);
            for(i = 1; i<=2; i++) {
                lcd.setCursor(3,1);
                lcd.print("Be careful.");
                delay(2000);
                lcd.setCursor(2,1);
                lcd.print(" Warning! "); // 공백으로 겹치는 부분을 조절
                delay(2000);
            }
            i = 1; // 반복을 위해 i값을 1로 변경
```

프로그램 소스

```
} else digitalWrite(Yellowled, LOW);

if(fire != 1) { // 화재 감지 시
    for(bright = 0; bright<=1; bright++) {
        digitalWrite(Greenled, LOW);
        digitalWrite(Redled, HIGH);
        delay(100);
        digitalWrite(Redled, LOW);
    }
    bright--; // 반복을 위함

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" ! SIREN ! ");
    lcd.setCursor(0,1);
    lcd.print("It's dangerous!!"); // lcd 출력

    for(int i=0; i<180; i++) {
        sinVI = sin(i*PI/180); // 사인 함수로 주파수 조절
        VI = (int) (3000+3000*sinVI);
        tone(buzzer, VI);
        delay(5); // 부저 사이렌 음 출력
    }
```

```
} else {
    digitalWrite(Redled, LOW);
    noTone(buzzer); // 화재를 감지하지 못했을 때 즉시 꺼짐
}
} else { // 에러 시 에러 코드 출력
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Error No: ");
    lcd.print(err);
    lcd.blink(); blinking = true;
}
}
delay(2000);
}
```

프로그램 소스

```
/*  
 * Public Constants  
 * (pitches.h)  
 */  
#define NOTE_B0 31  
#define NOTE_C1 33  
#define NOTE_CS1 35  
#define NOTE_D1 37  
#define NOTE_DS1 39  
#define NOTE_E1 41  
#define NOTE_F1 44  
#define NOTE_FS1 46  
#define NOTE_G1 49  
#define NOTE_GS1 52  
#define NOTE_A1 55  
#define NOTE_AS1 58  
#define NOTE_B1 62  
#define NOTE_C2 65  
#define NOTE_CS2 69  
#define NOTE_D2 73  
#define NOTE_DS2 78  
#define NOTE_E2 82  
#define NOTE_F2 87  
#define NOTE_FS2 93  
#define NOTE_G2 98  
  
#define NOTE_GS2 104  
#define NOTE_A2 110  
#define NOTE_AS2 117  
#define NOTE_B2 123  
#define NOTE_C3 131  
#define NOTE_CS3 139  
#define NOTE_D3 147  
#define NOTE_DS3 156  
#define NOTE_E3 165  
#define NOTE_F3 175  
#define NOTE_FS3 185  
#define NOTE_G3 196  
#define NOTE_GS3 208  
#define NOTE_A3 220  
#define NOTE_AS3 233  
#define NOTE_B3 247  
#define NOTE_C4 262  
#define NOTE_CS4 277  
#define NOTE_D4 294  
#define NOTE_DS4 311  
#define NOTE_E4 330  
#define NOTE_F4 349  
#define NOTE_FS4 370  
#define NOTE_G4 392  
#define NOTE_GS4 415  
  
#define NOTE_A4 440  
#define NOTE_AS4 466  
#define NOTE_B4 494  
#define NOTE_C5 523  
#define NOTE_CS5 554  
#define NOTE_D5 587  
#define NOTE_DS5 622  
#define NOTE_E5 659  
#define NOTE_F5 698  
#define NOTE_FS5 740  
#define NOTE_G5 784  
#define NOTE_GS5 831  
#define NOTE_A5 880  
#define NOTE_AS5 932  
#define NOTE_B5 988  
#define NOTE_C6 1047  
#define NOTE_CS6 1109  
#define NOTE_D6 1175  
#define NOTE_DS6 1245  
#define NOTE_E6 1319  
#define NOTE_F6 1397  
#define NOTE_FS6 1480  
#define NOTE_G6 1568  
#define NOTE_GS6 1661  
#define NOTE_A6 1760  
  
#define NOTE_AS6 1865  
#define NOTE_B6 1976  
#define NOTE_C7 2093  
#define NOTE_CS7 2217  
#define NOTE_D7 2349  
#define NOTE_DS7 2489  
#define NOTE_E7 2637  
#define NOTE_F7 2794  
#define NOTE_FS7 2960  
#define NOTE_G7 3136  
#define NOTE_GS7 3322  
#define NOTE_A7 3520  
#define NOTE_AS7 3729  
#define NOTE_B7 3951  
#define NOTE_C8 4186  
#define NOTE_CS8 4435  
#define NOTE_D8 4699
```

프로젝트 수행소감

기능이 많지는 않아서 실습을 해보지 않은 부품도 이용해보고 싶었지만 그러지 못한 데에 아쉬움을 느꼈습니다. 소스를 짤 때 루프 함수에 몰아서 적었는데, 다음에는 다른 함수를 만들어서 이용하는 편이 좋겠다는 생각도 들었습니다. 평소에 아두이노를 어려워 했던 것 보다 과정에서는 흥미가 생겨서 다른 주제를 가지고도 만들어 보고 싶습니다.

- 이 프로젝트에서 개선하고 싶은 점

단순히 화재 감지기의 기능만을 구현해서 그 기능까지는 문제가 없지만, 블루투스 모듈이나 텍스트 스위치를 이용해서 시간을 알려주는 화면으로 전환하는 기능도 추가해보고 싶습니다. 시간의 설정은 버튼이나 스마트폰의 앱을 이용해서(리모콘처럼) 현재 시간을 입력해준 후, 그 시간이 0:00가 되면 요일이나 월/일이 전환이 되는 형식이었습니다. 생각을 조금 더 해보면 조도센서로 백라이트 밝기를 조절하는 등 또 다른 센서들도 이용이 가능했을 것 같습니다.



감사합니다