

project-3

```
#include <LiquidCrystal_I2C.h>          // LiquidCrystal_I2C의 라이브러리를 불러옵니다. (I2C LCD)
#include<Servo.h> //Servo 라이브러리를 추가
LiquidCrystal_I2C lcd(0x27, 16, 2);    // lcd(LCD의 I2C 슬레이브 주소, lcd 1줄당 출력할 글자수, lcd 줄의 수)
Servo servo;      //Servo 클래스로 servo객체 생성

int value = 0;    // 각도를 조절할 변수 value
const int gasPin = A0;
int FAN = A1;
int laser = A2;
int red = 3;
int green = 5;
int blue = 6;
int flame = 11;  // 불꽃감지 센서 핀 번호 11
int state = 0;   // 불꽃감지 센서의 상태 값 저장 변수
int buzzerPin = 9;  // BUZZER 핀 번호 9

void setup(){
  pinMode(laser, OUTPUT); // define the digital output interface 13 feet
  digitalWrite(laser, LOW);
  pinMode(FAN, OUTPUT);
  digitalWrite(FAN, LOW);
  pinMode(buzzerPin, OUTPUT); // BUZZER를 출력으로 설정
  pinMode(flame, INPUT); // 불꽃 감지 센서를 입력으로 설정
  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(blue, OUTPUT);
  servo.attach(7);      //멤버함수인 attach : 핀 설정
  lcd.begin();          // LCD_I2C 통신을 시작합니다.
  lcd.backlight();      // LCD backlight를 ON
  Serial.begin(9600);   // 시리얼 통신, 속도는 9600
}
```

```

void loop()
{
    state = digitalRead(flame); //불꽃 감지 센서값을 입력받음

    Serial.println(analogRead(gasPin));
    delay(500);

    lcd.clear();           //lcd 화면을 지웁니다.
    lcd.home();            //lcd 커서 위치를 0,1로 위치시킵니다.

    if(analogRead(gasPin) > 500 || state == 0){
        digitalWrite(FAN, LOW);
        digitalWrite(laser, LOW); // open the laser head
        lcd.print("Fire!!");
        digitalWrite(buzzerPin, LOW);
        value = 100;             //각도를 100도 증가시킨다.
        servo.write(value); //value값의 각도로 회전. ex) value가 100이라면 100도 회전
        digitalWrite(red, LOW);
        digitalWrite(green, HIGH);
        digitalWrite(blue, LOW);

    }else {
        digitalWrite(FAN, HIGH);
        digitalWrite(laser, HIGH); // turn off the laser head
        lcd.print("No worries");

        digitalWrite(buzzerPin, HIGH);
        value = 0;             //각도를 0으로 초기화
        servo.write(value); //value값의 각도로 회전.
        digitalWrite(red, LOW);
        digitalWrite(green, LOW);
        digitalWrite(blue, LOW);
        delay(1000); // delay one second
    }
}

```