

Fire detection System

화재 감지 시스템

CONTENTS

001 조 활동

- 활동
- 주제 선정 과정

002 제작

- 회로 조립 & 코딩 구현
- 작동

003 활용도

- 활용방안

조 활동

활동 및 주제선정과정



001 | 회차별 활동

01

2018.05.01

PM 8-9:30

아두이노 프로그램 설치 및 코딩예제 실습 (~57p)

02

2018.05.08

PM 8-10

PPT로 교재 학습 및 센서 실습, (~)

03

2018.05.17

Pm 7-9:30

PPT로 교재 학습 및 센서 실습, (~)

04

2018.05.29

PM 7-9:30

조별 발표 주제를 구상하다가 가스센서를 중점으로 생각해서 구현하게 됨.

001 | 회차별 활동

05

2018.05.31

PM 8-9:30

구현에 쓰일 헤더파일, 센서, 자료조사

06

2018.06.02 - 03

1차 발표자료 준비

07

2018.06.00

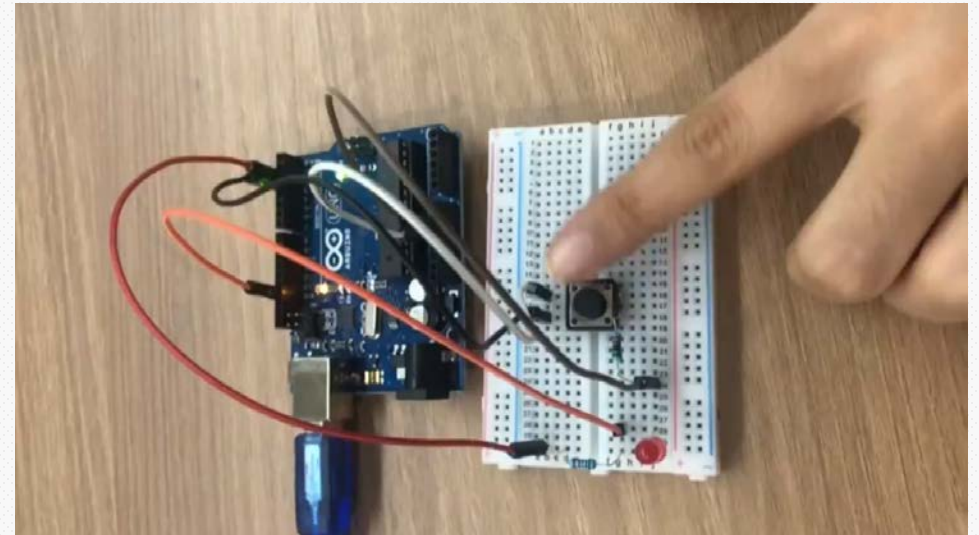
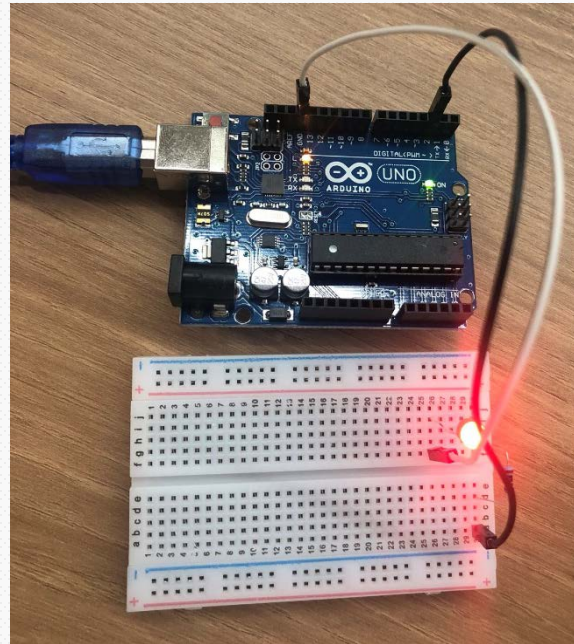
오피셜아워 후 수정예정

08

2018.06.00

최종 발표자료 준비

001 | 회차별 활동



제작

조립 - 코딩 - 작동



002 | 제작

필요한 준비물

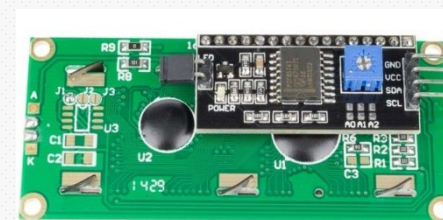
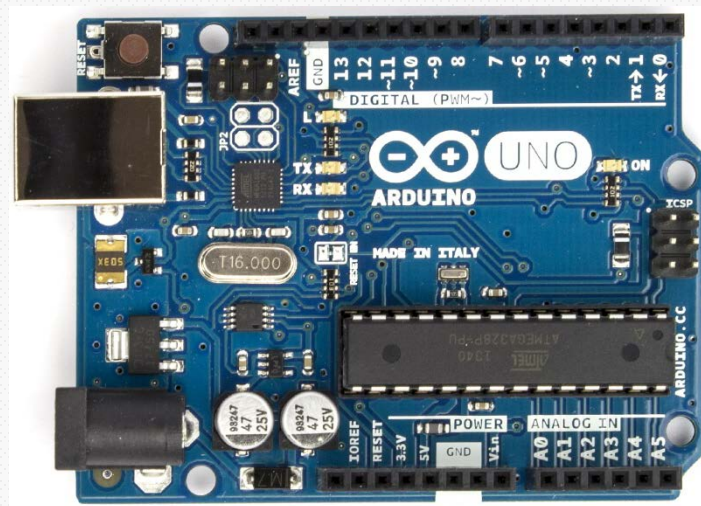
입력



처리 및 연산

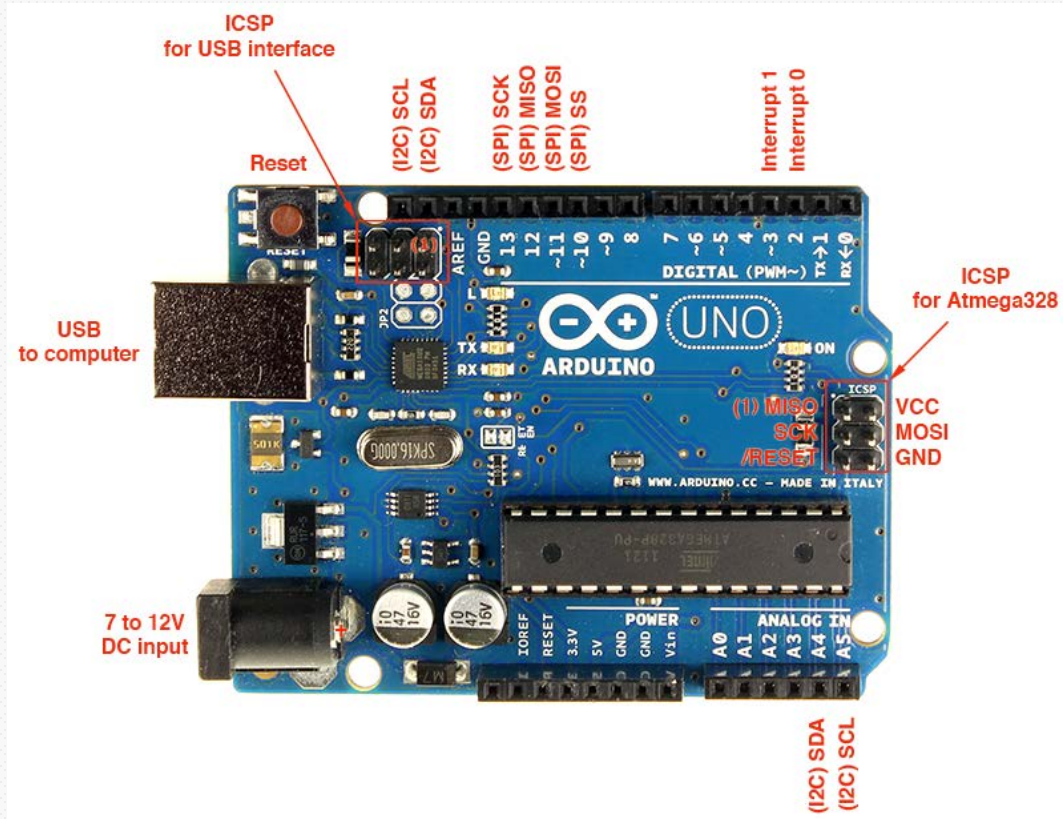


출력



002 | 제작

Arduino UNO

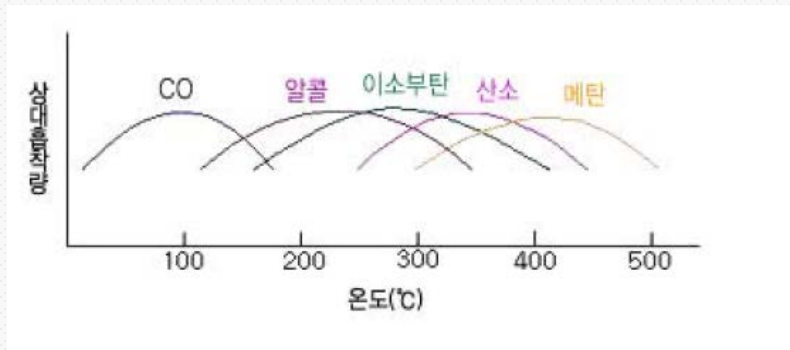


아두이노 우노(Arduino UNO)

- 오픈소스를 기반으로 한 단일 보드 마이크로컨트롤러이다.
- 아두이노의 표준 보드
- 원리
 - 아두이노는 다수의 스위치나 센서로부터 값을 받아들이며,
 - LED나 모터와 같은 외부 전자 장치들을 통제함으로써 환경과 상호 작용이 가능한 물건을 만들어 낼 수 있다.
- 특징
 - 마이크로 컨트롤러를 쉽게 동작 시킬 수 있다.
 - 일반적으로 AVR 프로그래밍이 WinAVR로 컴파일하여, ISP장치를 통해 업로드를 해야 함
 - 컴파일된 펌웨어를 USB를 통해 업로드
 - 윈도우를 비롯해 맥OS, 리눅스 등 여러 OS지원

002 | 제작

가스센서MQ2

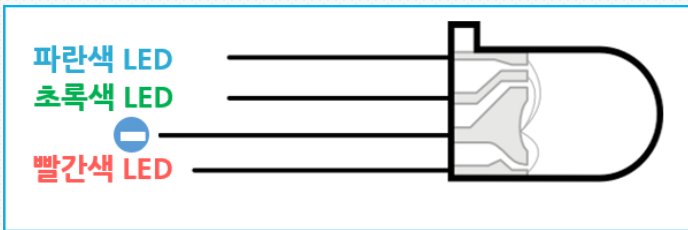


가스센서 MQ-2

- 내부에 포함된 히터와 센서로 LPG, 부탄 메탄, 알코올 등을 검출해낼 수 있다.
- 원리
 - 센서 내부에 있는 히터가 가열이 되면 센서내부의 금속막에 공기중의 성분이 달라붙게 되는데,
 - 이에 따라 저항 값이 낮아지고,
 - 이를 전압 분배의 법칙을 통해 아날로그 값을 아두이노에 보내주게 된다.
 - (기체의 종류에 따라 흡착량도 달라진다.)
- 특징
 - 내부에 포함된 히터나 적정온도로 올라가서 안정적인 결과를 얻기 위해서는 대기시간이 필요하다.
 - 모듈의 종류별로 사용법이 약간씩 틀리다.
 - 특히 MQ-2는 빠른 response time을 갖고 있어서 거의 즉시 값을 한다.
 - 사용할 때 모듈의 발열은 고장이 아닌 정상적인 작동

002 | 제작

RGB LED



RGB LED

- 빨간색, 초록색, 파란색 3개의 LED를 하나의 LED로 합친 LED
- 원리
 - 한쪽 방향으로 전류가 흐르도록 제어하는 반도체 소자를 다이오드라 하는데,
 - 이 중 전기 에너지를 빛 에너지로 변환 하는 것을 LED(Light Emitting Diode)라고 한다.
- 특징
 - 여러 색을 낼 수 있다.
 - 합성도 가능

002 | 제작

I2C LCD



I2C 캐릭터 LCD 모듈

- LCD란 Liquid Crystal Display의 약자로 액정 표시 장치
- 16*2 Character LCD로서 1줄에 16개의 문자씩 총 2줄에 문자를 표기
- 특징
 - 기존 - 6개의 digital Pin을 사용
 - → I2C LCD 모듈 (I2C Converter)을 사용하면 I2C interface를 사용해 LCD 제어가 가능해지며, 총 4개의 배선으로 아두이노에 연결이 가능
 - 아두이노에 연결하여 제어할 경우 두 개의 AnalogPin(SDA, SCL)으로 제어가 가능
 - 대부분의 16*2 LCD는 5V전원을 통해 Backlight를 동작 시킬 수 있다.
 - 가변 저항을 통해 문자의 명암(선명도)를 조절

002 | 제작

코드 구현

```
fire_program | 아두이노 1.8.5
파일 편집 스케치 툴 도움말

fire_program

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);

const int gasPin = A0 ;
int redPin = 11;
int greenPin = 10;
int bluePin = 9;

int a;

void setup()
{
  lcd.init();
  lcd.begin(16,2);
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

저장 완료.

스케치는 프로그램 저장 공간 4618 바이트(14%)를 사용, 최대 32256 바이트,
전역 변수는 동적 메모리 430바이트(20%)를 사용, 1618바이트의 지역변수가 남.
```

```
fire_program | 아두이노 1.8.5
파일 편집 스케치 툴 도움말

fire_program

void loop()
{
  a = analogRead(gasPin);
  Serial.print(a);
  lcd.backlight();

  // green
  if (60 >= a){
    setColor(0, 100, 0);
    Serial.println("green");
    lcd.init();
    lcd.setCursor(6, 0);
    lcd.print("s"); lcd.print("a"); lcd.print("f"); lcd.print("e");
    lcd.blink();
  }
  // yellow
  else if (80 >= a && a > 60){
    setColor(100, 100, 0);
    Serial.println("yellow");
    lcd.init();
    lcd.setCursor(6, 0);
    lcd.print("w"); lcd.print("a"); lcd.print("r"); lcd.print("n");
    lcd.blink();
  }
  // red
  else if (a > 80){
    setColor(100, 0, 0);
    Serial.println("red");
    lcd.init();
    lcd.setCursor(5, 1);
    lcd.print("f"); lcd.print("l"); lcd.print("r"); lcd.print("e"); lcd.print("l"); lcd.print("l");
    lcd.blink();
  }
  delay(1000);
}

저장 완료.

스케치는 프로그램 저장 공간 4618 바이트(14%)를 사용, 최대 32256 바이트,
전역 변수는 동적 메모리 430바이트(20%)를 사용, 1618바이트의 지역변수가 남, 최대는 2048 바이트.
```

```
fire_program | 아두이노 1.8.5
파일 편집 스케치 툴 도움말

fire_program

void setColor(int red, int green, int blue)
{
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}

저장 완료.

스케치는 프로그램 저장 공간 4618 바이트(14%)를 사용, 최대 32256 바이트,
전역 변수는 동적 메모리 430바이트(20%)를 사용, 1618바이트의 지역변수가 남, 최대는 2048 바이트.
```


002 | 제작

Wire.h

Wire.h

아두이노에서는 I2C 통신을 쉽게 사용할 수 있도록 하기 위해 “Wire” 라는 객체를 제공하고 있다. “Wire” 객체는 “Wire.h” 파일에 선언되어 있으므로 “#include” 구문을 이용해 추가해줘야 한다. Wire 객체는 전역으로 이미 선언된 객체이므로 따로 선언해 줄 필요는 없다.

역할

- I2C 통신을 위한 핀으로 SDA, SCL 핀을 하나씩 제공한다. 우노 (UNO)의 경우에는 SDA, SCL 핀이 따로 표시되어 있지는 않기 때문에 핀 순서에 조금 유의해야한다.
- 우노 (UNO)는 ‘A4’ 번 핀이 SDA, ‘A5’ 번 핀이 SCL 핀이다.

002 | 제작

LiquidCrystal_I2C.h

LiquidCrystal_I2C.h

기본 LCD회로를 보다 편리하게 사용하기 위해 만든 라이브러리

역할

- LiquidCrystal_I2C lcd(0x27,16,2); //주소
- Lcd.init(); //초기화
- Lcd.begin(16,2); //시작
- Lcd.backlight(); //백라이트 설정
- Lcd.setCursor(6, 0); //화면출력 위치
- Lcd.print(" s "); //문자출력
- Lcd.blink(); //화면깜박이기

002 | 제작

코드 해설1

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

```
const int gasPin = A0;
int redPin = 11;
int greenPin = 10;
int bluePin = 9;
```

```
int a;
```

```
void setup()
{
  lcd.init();
  lcd.begin(16,2);
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}
```

<Wire.h>, <LiquidCrystal_I2C.h>
불러오기

LiquidCrystal_I2C 타입 객체 선언

연결할 핀(번호)을 미리 설정

gasPin = 가스센서
red/green/blue = LED 모듈

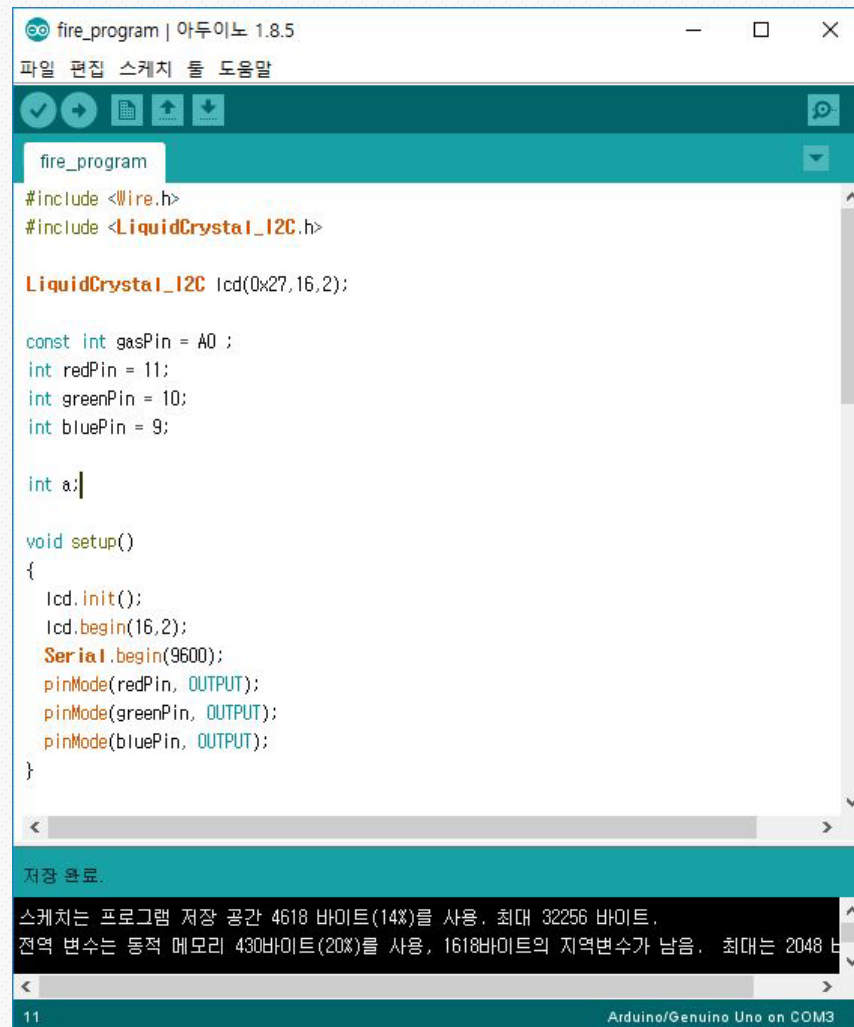
a :이산화탄소 측정값 변수선언

void setup() //초기화

{
 LCD 초기화
 LCD 시작점 지정
 전송시작

 설정된 핀으로 디지털 신호 출력

}



002 | 제작

코드해설2

```
fire_program | 아두이노 1.8.5
파일 편집 스케치 툴 도움말

fire_program

void loop()
{
  a = analogRead(gasPin);
  Serial.println(a);
  lcd.backlight();

  // green
  if(60 >= a){
    setColor(0, 100, 0);
    Serial.println("green");
    lcd.init();
    lcd.setCursor(6, 0);
    lcd.print("s"); lcd.print("a"); lcd.print("f"); lcd.print("e");
    lcd.blink();
  }

  // yellow
  else if(80 >=a && a > 60){
    setColor(100, 100, 0);
    Serial.println("yellow");
    lcd.init();
    lcd.setCursor(6, 0);
    lcd.print("w"); lcd.print("a"); lcd.print("r"); lcd.print("n");
    lcd.blink();
  }

  // red
  else if(a > 80){
    setColor(100, 0, 0);
    Serial.println("red");
    lcd.init();
    lcd.setCursor(5, 1);
    lcd.print("f"); lcd.print("i"); lcd.print("r"); lcd.print("e"); lcd.print("!"); lcd.print("!");
    lcd.blink();
  }
  delay(1000);
}

저장 완료.

스케치는 프로그램 저장 공간 4618 바이트(14%)를 사용, 최대 32256 바이트.
전역 변수는 동적 메모리 430바이트(20%)를 사용, 1618바이트의 지역변수가 남음. 최대는 2048 바이트.
```

65 Arduino/Genuino Uno on COM3

002 | 제작

코드해설2

```
void loop()
{
  a = analogRead(gasPin);
  Serial.println(a);
  lcd.backlight();
}
```

```
// green
if(60 >= a){

  setColor(0, 100, 0);
  Serial.println("green");
}
```

```
lcd.init();
```

```
lcd.setCursor(6, 0);
```

```
lcd.print("s"); lcd.print("a");
lcd.print("f"); lcd.print("e");
lcd.blink();
}
```

```
void loop()
{
  a = 이산화탄소 측정량
  시리얼 창에 측정량 출력
  LED on
}
```

```
// green
if(60 >= a){
  // 측정량에 따라 조건설정
  setColor 함수로 색깔 설정
  시리얼창에 출력
}
```

```
LCD초기화
```

```
LCD 시작점 지정
```

```
출력 문자 입력
```

```
깜박거림 설정
```

```
}
```



```
fire_program | 아두이노 1.8.5
파일 편집 스케치 툴 도움말

fire_program

void loop()
{
  a = 이산화탄소 측정량
  Serial.println(a);
  lcd.backlight();

  // green
  if(60 >= a){
    setColor(0, 100, 0);
    Serial.println("green");
    lcd.init();
    lcd.setCursor(6, 0);
    lcd.print("s"); lcd.print("a"); lcd.print("f"); lcd.print("e");
    lcd.blink();
  }
}
```

002 | 제작

코드해설2

```
delay(1000);  
}
```

```
1초 기다리기  
}
```

fire_program | 아두이노 1.8.5

파일 편집 스케치 툴 도움말

```
void loop()  
{  
  a = analogRead(gasPin);  
  Serial.println(a);  
  lcd.backlight();  
  
  // green  
  if(60 >= a){  
    setColor(0, 100, 0);  
    Serial.println("green");  
    lcd.init();  
    lcd.setCursor(6, 0);  
    lcd.print("s"); lcd.print("a"); lcd.print("f"); lcd.print("e");  
    lcd.blink();  
  }  
  // yellow  
  else if(80 >= a && a > 60){  
    setColor(100, 100, 0);  
    Serial.println("yellow");  
    lcd.init();  
    lcd.setCursor(6, 0);  
    lcd.print("w"); lcd.print("a"); lcd.print("r"); lcd.print("n");  
    lcd.blink();  
  }  
  // red  
  else if(a > 80){  
    setColor(100, 0, 0);  
    Serial.println("red");  
    lcd.init();  
    lcd.setCursor(5, 1);  
    lcd.print("f"); lcd.print("i"); lcd.print("r"); lcd.print("e"); lcd.print("!"); lcd.print("!");  
    lcd.blink();  
  }  
  delay(1000);  
}
```

저장 완료.

스케치는 프로그램 저장 공간 4618 바이트(14%)를 사용, 최대 32256 바이트.
전역 변수는 동적 메모리 430바이트(20%)를 사용, 1618바이트의 지역변수가 남음. 최대는 2048 바이트.

65 Arduino/Genuino Uno on COM3

002 | 제작

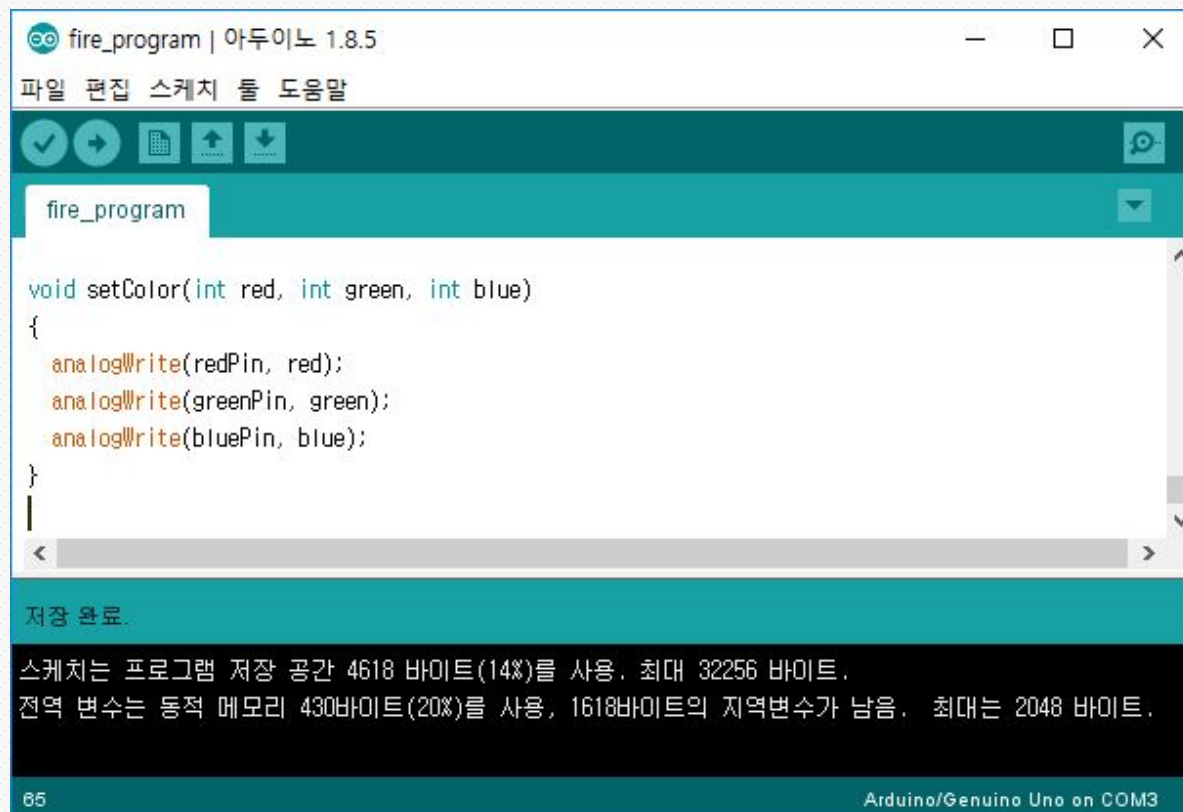
코드해설3

```
void setColor(int red, int green, int blue)
{
    analogWrite(redPin, red);
    analogWrite(greenPin, green);
    analogWrite(bluePin, blue);
}
```

```
void setColor(int red, int green, int blue)
{ //색 크기를 인수로 받기

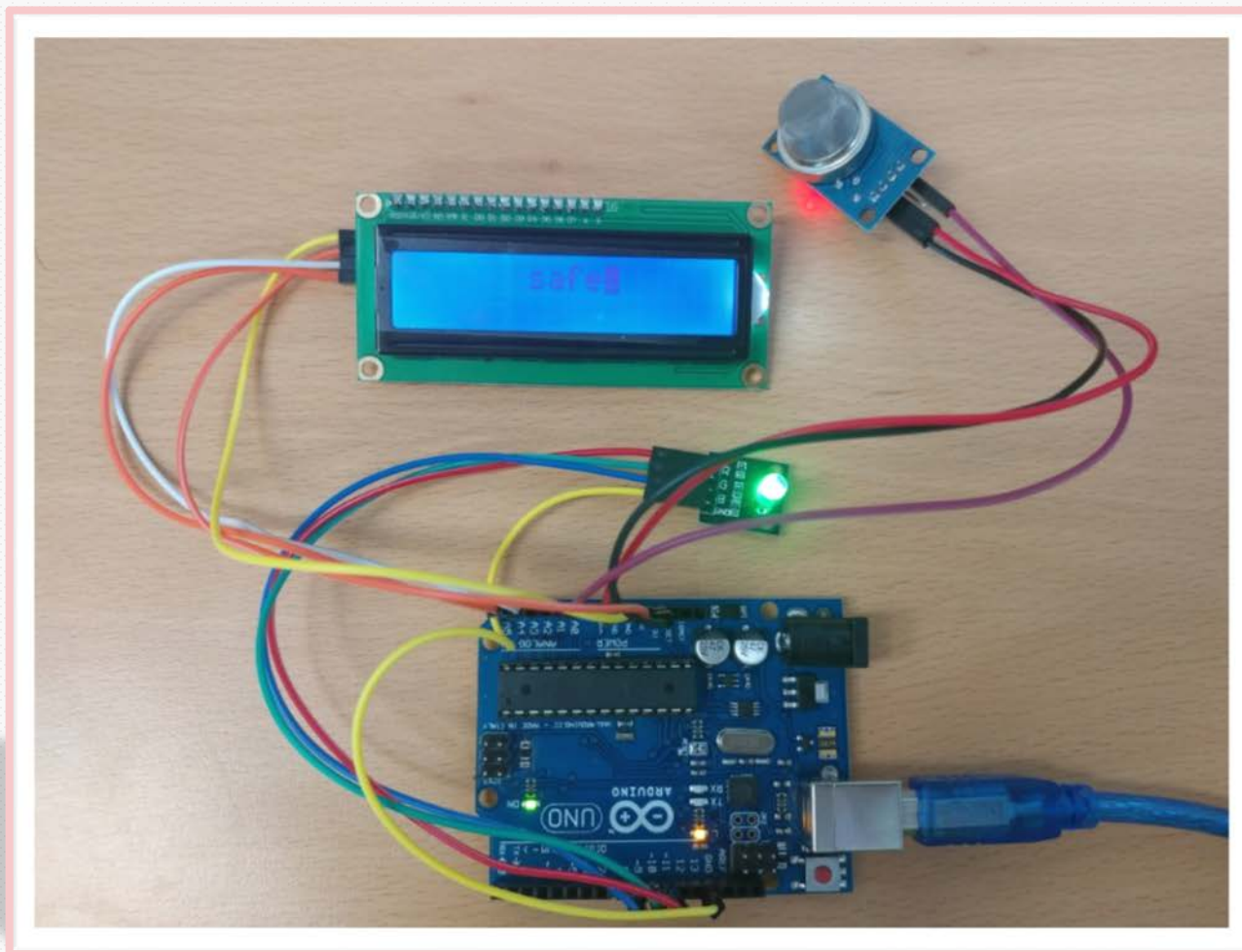
    핀으로 출력

}
```



002 | 제작

완성품



002 | 제작

완성품

$\text{CO}_2 \leq 60$



$60 < \text{CO}_2 \leq 80$



$80 < \text{CO}_2$



활용도

활용방안



003 | 활용도

활용방안1

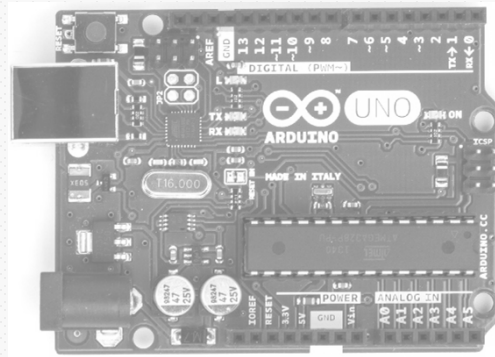
입력



처리 및 연산



출력



입력 센서의 활용

조도/온습도/인체감지 등등 여러 센서가 있고, 동시에 여러가지 센서를 사용해도 되지만,
입력센서를 가스센서로 고정 했을 때,
상황/측정가스종류에 따라 나뉠 수 있다.

또다른 강점 → 여러상황에 동시에 쓰일 수 있다.

감지 가스의 종류에 따라
산소+이산화탄소 = 실내 환경측정, 공기청정기
이산화탄소 등등 = 화재감지 or 자살방지

003 | 활용도

활용방안2

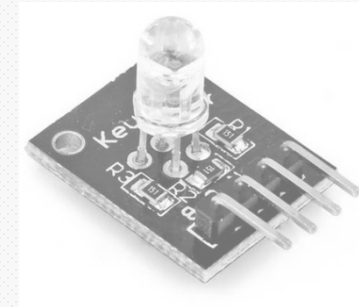
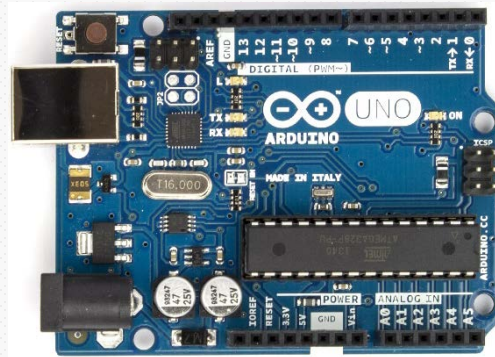
입력



처리 및 연산



출력



출력 단자의 활용

사람이 입력 받을 수 있는 오감으로 나뉘었을 때,
시각 - 색, 디스플레이
청각 - 소리, 버저
후각 - 냄새
촉각 - 진동

기계가 입력 받아야 한다면,
통신센서 - 블루투스 신호, 진동 등



004 | 참고문헌

참고사이트

가스센서

<https://m.blog.naver.com/roboholic84/220526674407>

RGB LED

<https://kocoafab.cc/tutorial/view/725>

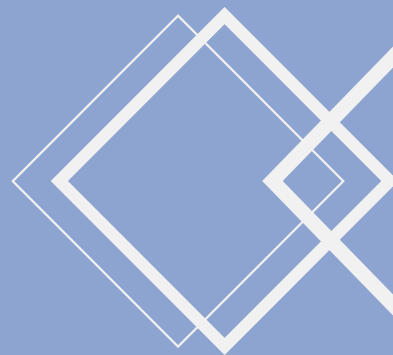
LCD I2C

<https://blog.naver.com/jmerrier/221057114197>

<https://kocoafab.cc/tutorial/all?level=1&page=0>

Wire.h

<https://m.blog.naver.com/PostView.nhn?blogId=yuyyulee&logNo=220325361752&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F>



Thank you
For Watching

Any Question...?