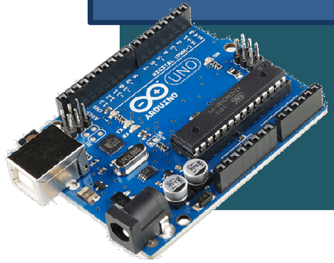
2022년 IoT기반 스마트 솔루션 개발자 양성과정



Firmware [펌웨어]

6-Analog Sensor

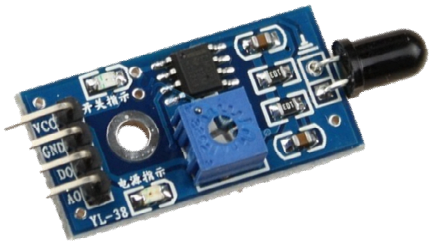
담당 교수 : 유근택
010-5486-5376
rgt3340@naver.com
http://cafe.naver.com/cbdsp

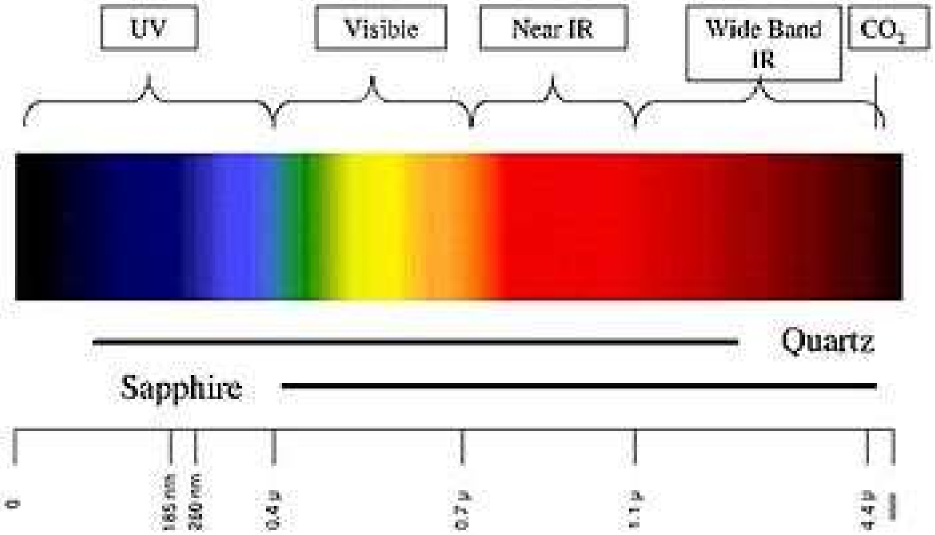
충북대학교 공동훈련센터

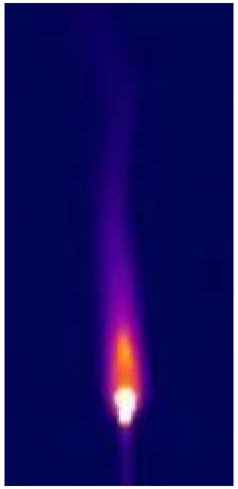
Flame Sensor

- Features
  - Fast response time
  - High photo sensitivity
  - This product itself will remain within RoHS compliant version.

- Descriptions
  - YG1006 is a high speed and high sensitive NPN silicon phototransistor in a standard 5mm package.
  - Due to its black epoxy the device is sensitive to infrared radiation.



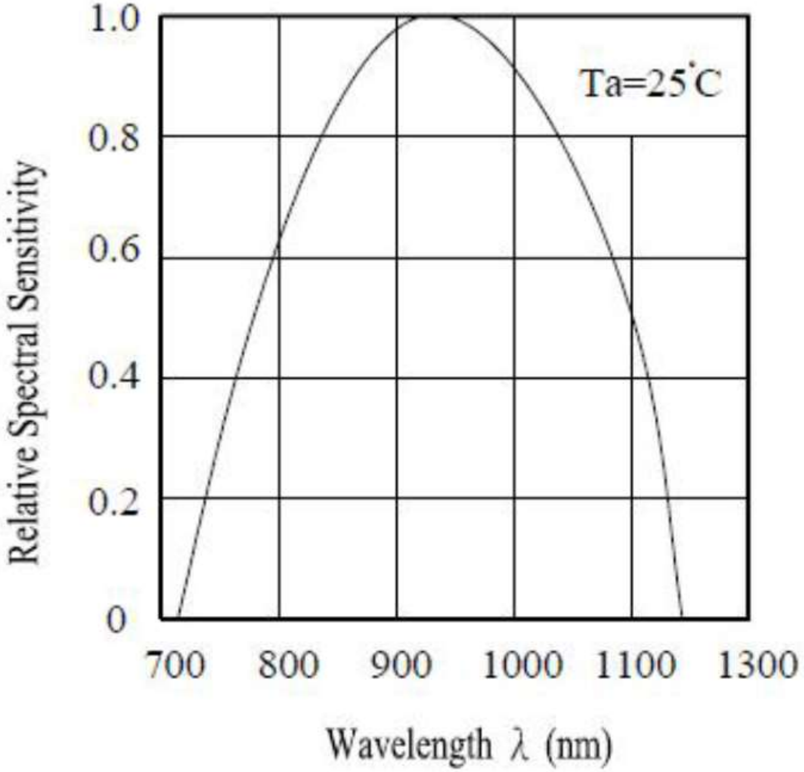

UV
Visible
Near IR
Wide Band IR
CO_2
Quartz
Sapphire
0
185 nm
280 nm
0.4 μ
0.7 μ
1.1 μ
4.4 μ

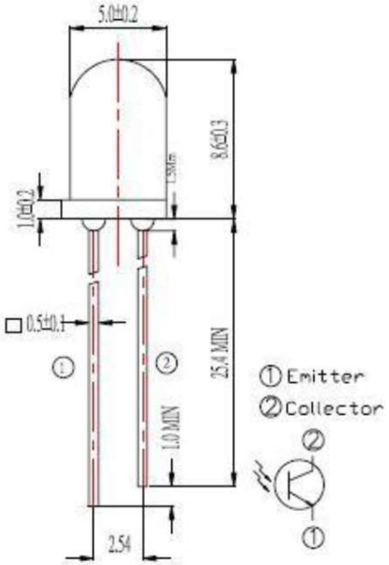


Flame sensor specifications.

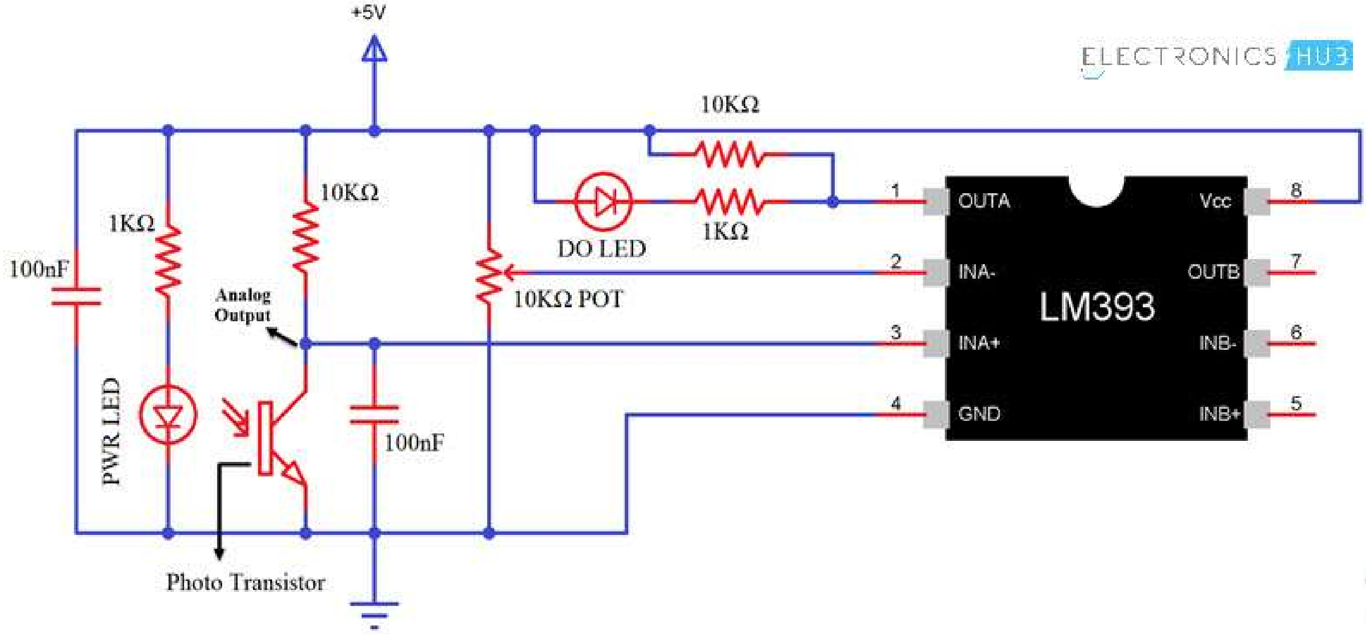
- **Overview**
  - Sensitive to flame spectrum
  - Features wide range voltage comparator LM393
  - Adjustable sensitivity
  - Signal output indicator
- **Specifications**
  - Spectrum range: 760nm ~ 1100nm
  - Detection angle: 0 - 60 degree
  - Power: 3.3V ~ 5.3V
  - Operating temperature: -25℃ ~ 85℃
  - Dimension: 27.3mm * 15.4mm
  - Mounting holes size: 2.0mm
- **Applications**
  - Fire detection
  - Fire fighting robot
  - Fire alarm
- **How to Use**
  - In the case of working with a MCU:
  - VCC ↔ 3.3V ~ 5.3V
  - GND ↔ power supply ground
  - AOUT ↔ MCU.IO (analog output)
  - DOUT ↔ MCU.IO (digital output)

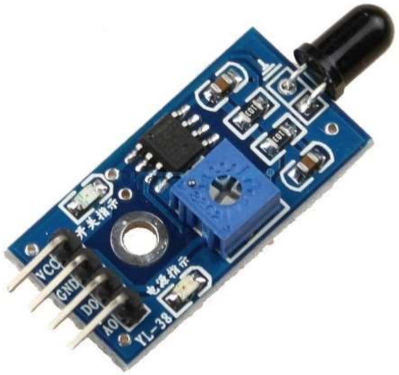
Typical Electro-Optical Characteristics Curves



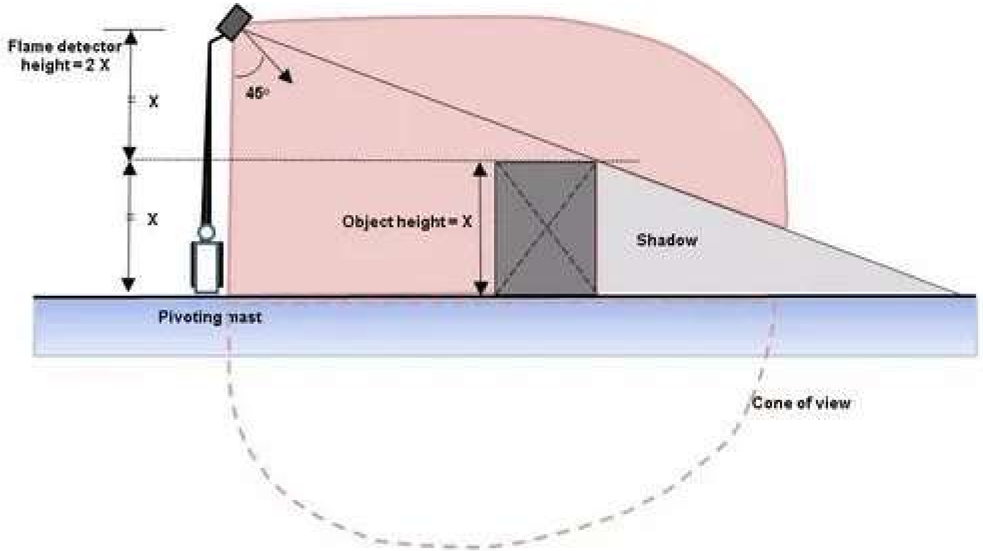


Sensor Module Circuit





Detection and application



- 산업용 가스 터빈.
- 수소 스테이션.
- 화재 감지 장치.
- 화재 경보.
- 가정용 및 산업용 난방 시스템.
- 소방 로봇.
- 산업용 용광로.
- 건조 시스템.
- 가스 구동 식 조리 장치.
- 방사선 연구실.
- 광산.
- 점화 시스템.

system configuration

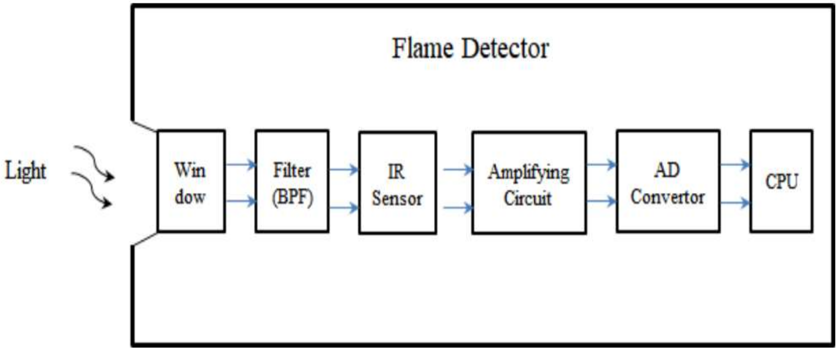
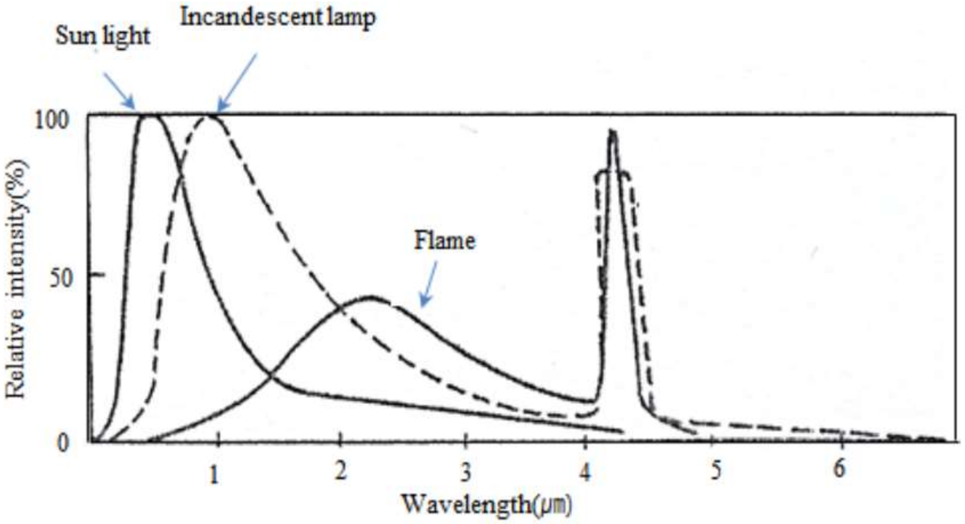
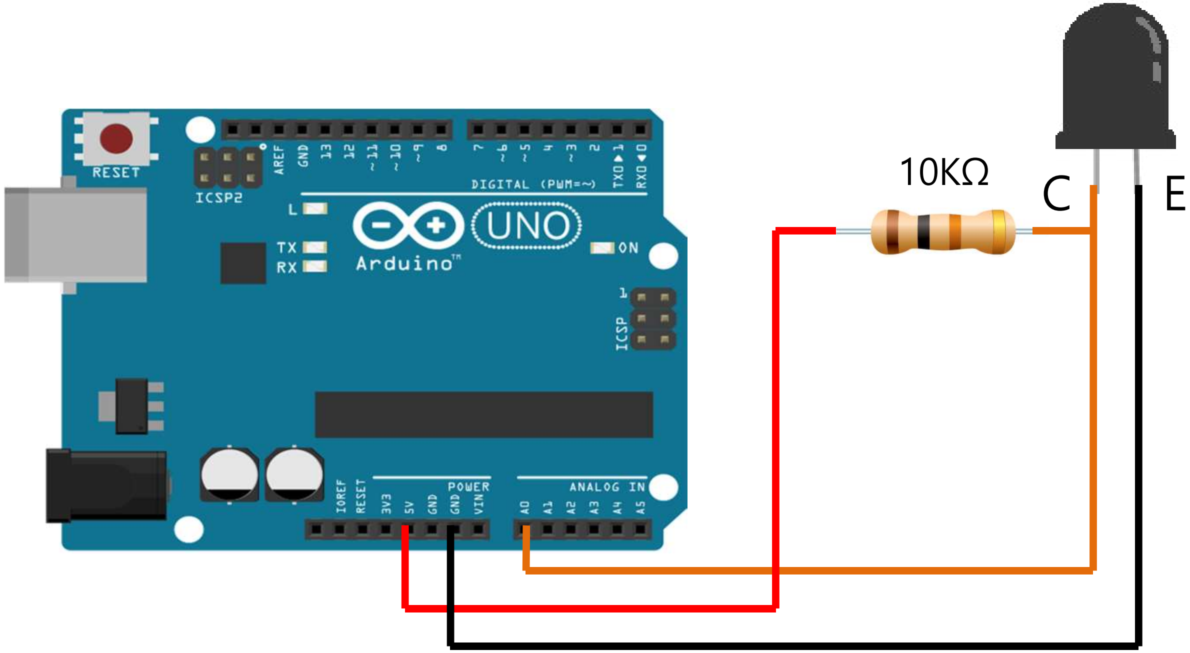


Diagram of flame detector



Spectral distribution chart comparing different lighting

Wiring


10KΩ
C
E
RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino™
ON
ICSP
1
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

FlameSensor-1

```
#define FlameSensor A0

void setup( ) {
  Serial.begin(9600);
}

void loop( ) {
  int FlameValue=analogRead(FlameSensor);
  Serial.println(FlameValue);
}
```

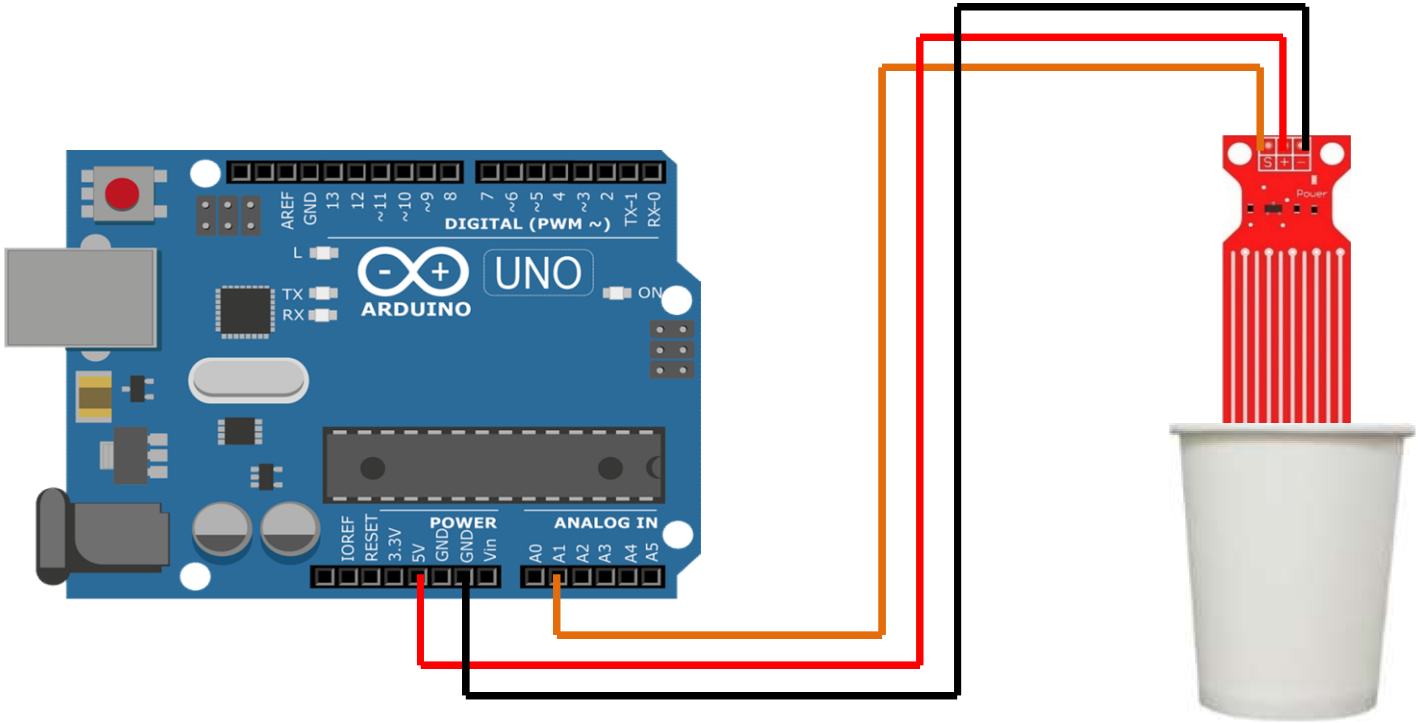
FlameSensor-2

```
// lowest and highest sensor readings:
const int sensorMin = 0;  // sensor minimum
const int sensorMax = 1024; // sensor maximum

void setup() {
// initialize serial communication @ 9600 baud:
Serial.begin(9600);
}
void loop() {
// read the sensor on analog A0:
    int sensorReading = analogRead(A0);
// map the sensor range (four options):
// ex: 'long int map(long int, long int, long int, long int, long int)'
    int range = map(sensorReading, sensorMin, sensorMax, 0, 3);

// range value:
switch (range)
{
        case 0:     // A fire closer than 1.5 feet away.
          Serial.println("** Close Fire **");
          break;
        case 1:    // A fire between 1-3 feet away.
          Serial.println("** Distant Fire **");
          break;
        case 2:    // No fire detected.
          Serial.println("No Fire");
          break;
 }
delay(1);            // delay between reads
}
```

Water Sensor

- Working voltage: 5V
- Working Current: < 20mA
- Interface: Analog
- Width of detection: 40mm×16mm
- Working Temperature: 10℃~30℃
- Weight: 3g
- Size: 65mm×20mm×8mm
- Arduino compatible interface
- Low power consumption
- High sensitivity
- Output voltage signal: 0~4.2V

Wiring

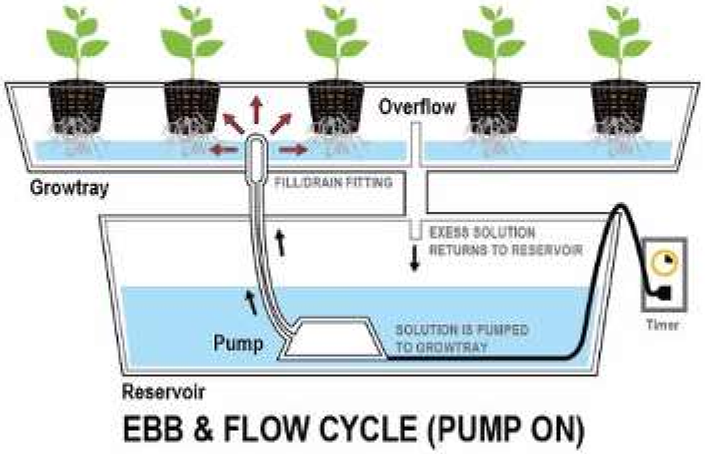

AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX-1
RX-0
DIGITAL (PWM ~)
L
TX
RX
ARDUINO
UNO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
S
+
−
Power

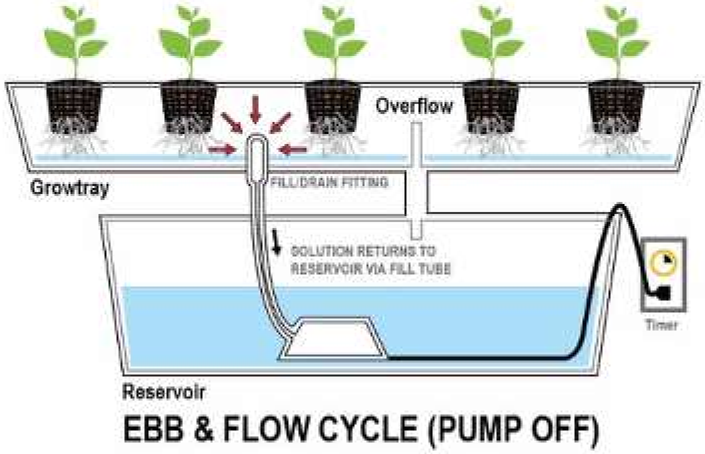
WaterSensor-1

```
#define WaterSensor A1

void setup( ) {
  Serial.begin(9600);
}

void loop( ) {
  int WaterValue=analogRead(WaterSensor);
  Serial.println(WaterValue);
}
```

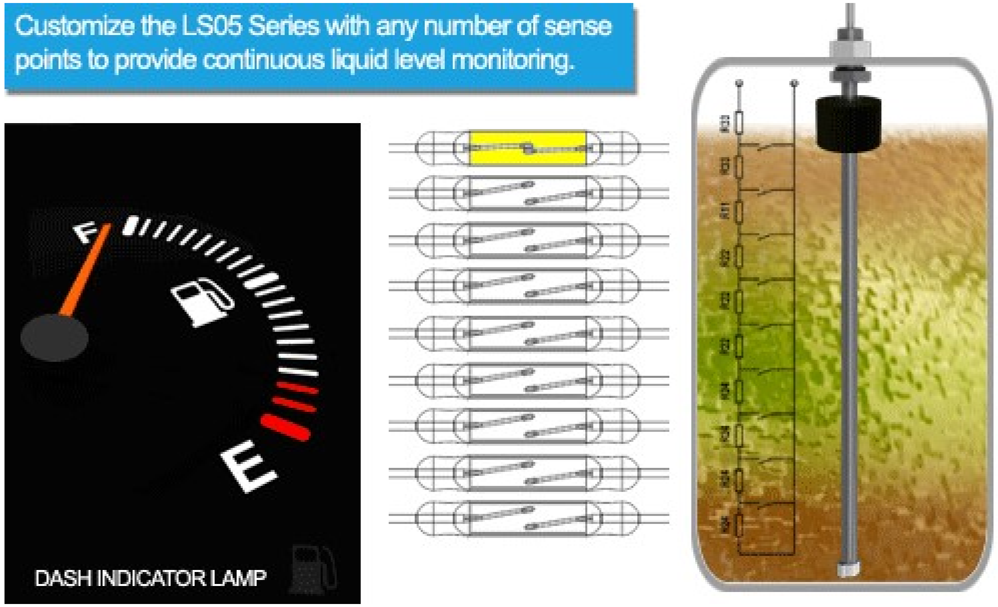
WaterSensor Application

- 수경재배





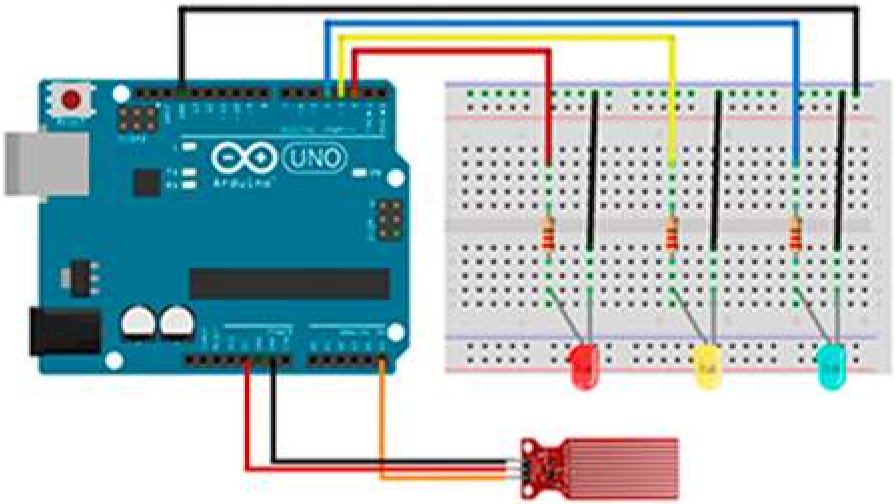
차량 연료탱크 센서


Customize the LS05 Series with any number of sense points to provide continuous liquid level monitoring.

WaterSensor-2

```
int resval = 0;  // holds the value
int respin = A5; // sensor pin used

void setup() {
   // start the serial console
  Serial.begin(9600);
}

void loop() {

  resval = analogRead(respin); //Read data from analog pin and store it to resval variable

  if (resval<=100){ Serial.println("Water Level: Empty"); }
  else if(resval>100 && resval<=300){ Serial.println("Water Level: Low"); }
  else if(resval>300 && resval<=330){ Serial.println("Water Level: Medium"); }
  else if(resval>330){     Serial.println("Water Level: High");  }
  delay(1000);
}
```

WaterSensor-3



아두이노 우노보드	수분 수위 센서
A0	S
5V	+
GND	-

아두이노 우노보드	3색 LED
2	빨강(긴 다리)
3	노랑(긴 다리)
4	파랑(긴 다리)
GND (-)	GND(짧은 다리)

WaterSensor-3

```
int water_pin = A0;      //수분수위센서 A0에 연결
int LED1 = 2;            // LED를 각각 디지털핀 2,3,4에 연결
int LED2 = 3;
int LED3 = 4;

void setup() {
  Serial.begin(9600);   // Serial monitor 구동 전원입력
  pinMode( A0,   INPUT); // A0핀을 입력으로 설정
  pinMode( 2,   OUTPUT); // 디지털핀 2,3,4를 출력으로 설정
  pinMode( 3,   OUTPUT);
  pinMode( 4,   OUTPUT);
}

void loop(){
  Serial.println(analogRead(A0));  // Serial monitor로 A0값을 보여줌
  delay(100);                     // 입력값을 보여주는데 0.1초 설정
  if(analogRead(A0) < 400){      // A0가 400이하면 모든 LED OFF
    digitalWrite( 2, LOW);
    digitalWrite( 3, LOW);
    digitalWrite( 4, LOW);
    delay(100);
  }
   else if( 400 < analogRead(A0) && analogRead(A0) < 500 ){
    digitalWrite( 2, HIGH);
    digitalWrite( 3, LOW);
    digitalWrite( 4, LOW);        // 400<A0<500이면 빨간 LED ON
    delay(100);
  }
   else if( 500 < analogRead(A0) && analogRead(A0) < 600){
    digitalWrite( 2, HIGH);
    digitalWrite( 3, HIGH);
    digitalWrite( 4, LOW);        // 500<A0<600이면 빨,노 LED ON
    delay(100);
  }
   else {
    digitalWrite( 2, HIGH);
    digitalWrite( 3, HIGH);
    digitalWrite( 4, HIGH);
    delay(100);                   // 이외 모든 경우 빨,노,파 LED ON
  }
}
```

WaterSensor-4

```
#include <LiquidCrystal.h>     // include the library code:

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //initialise the library with the numbers of the interface pins

int resval = 0;  // holds the value
int respin = A5; // sensor pin used

void setup() {
   lcd.begin(16, 2);                         // set up the LCD's number of columns and rows:
   lcd.print("WATER LEVEL: ");        // Print a message to the LCD.

}

void loop() {
  lcd.setCursor(0, 1);                   // set the cursor to column 0, line 1
  resval = analogRead(respin);   //Read data from analog pin and store it to resval variable

  if (resval<=100){ lcd.println("Empty "); }
  else if (resval>100 && resval<=300){ lcd.println("Low "); }
  else if (resval>300 && resval<=330){ lcd.println("Medium "); }
  else if (resval>330){      lcd.println("High                ");
  }
  delay(1000);
}
```