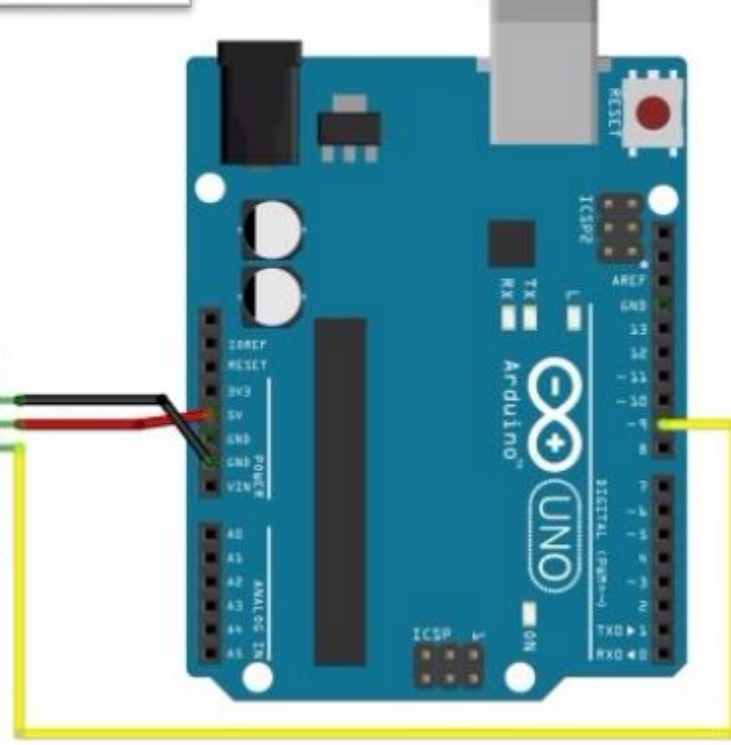
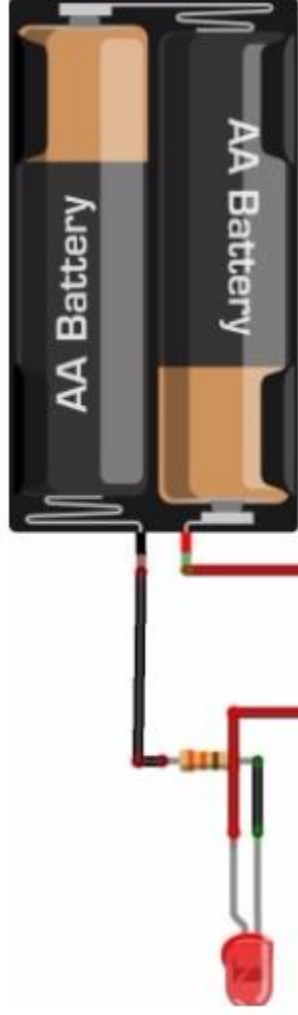


센서 3

릴레이/조이스틱/배터리/
토양수분센서/빛 방울 센서/
GPS센서/블루투스/컬러 센서

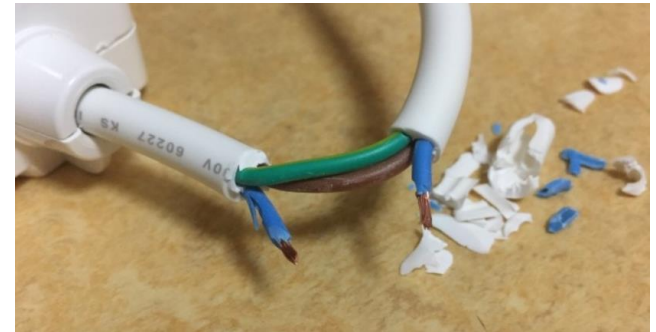


```
int relay = 9;
void setup ()
{
  pinMode (relay, OUTPUT);
}
void loop ()
{
  digitalWrite (relay, HIGH);
  delay (1000);
  digitalWrite (relay, LOW);
  delay (1000);
}
```

릴레이 연결의 기초 예제!

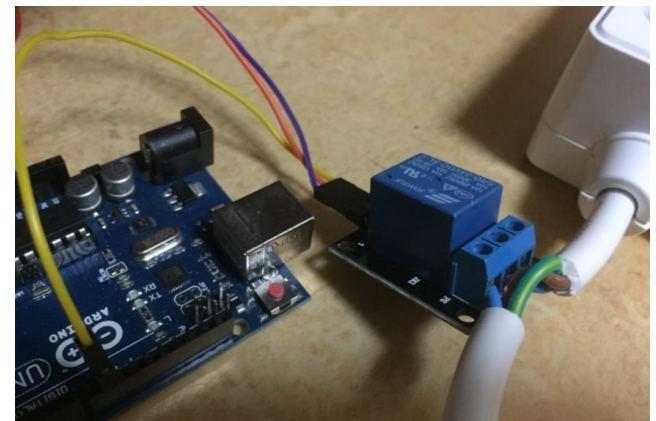
- 위 이미지를 기준으로 전원연결측 단자(왼쪽)과 아두이노 연결 단자(오른쪽)으로 나눔
- 아두이노 연결단자는 모듈에 적혀있는대로 Vcc는 5V로 연결하고 Gnd 는 Gnd로, S는 신호를 받는 단자로 소스코드와 같은 핀에 연결
- 외부전원 연결단자는 가운데 붉은 단자는 공통 단자 이며 ON일때 연결하려면 위쪽 단자에 OFF일때 연결하려면 아래쪽단자에 연결하면 됨
- 다시말해서 위쪽단자와 함께 연결할 경우 아두이노에서 신호를 줄때(HIGH) LED가 켜지게 됨

- 릴레이는 낮은전압으로 높은 전압의 기기를 컨트롤할 때 사용하는 전자부품
- 예를 들어서, 아두이노의 5V전압으로 220V의 헤어드라이기나 전등, 선풍기 등 전제제품을 제어 할 수 있음

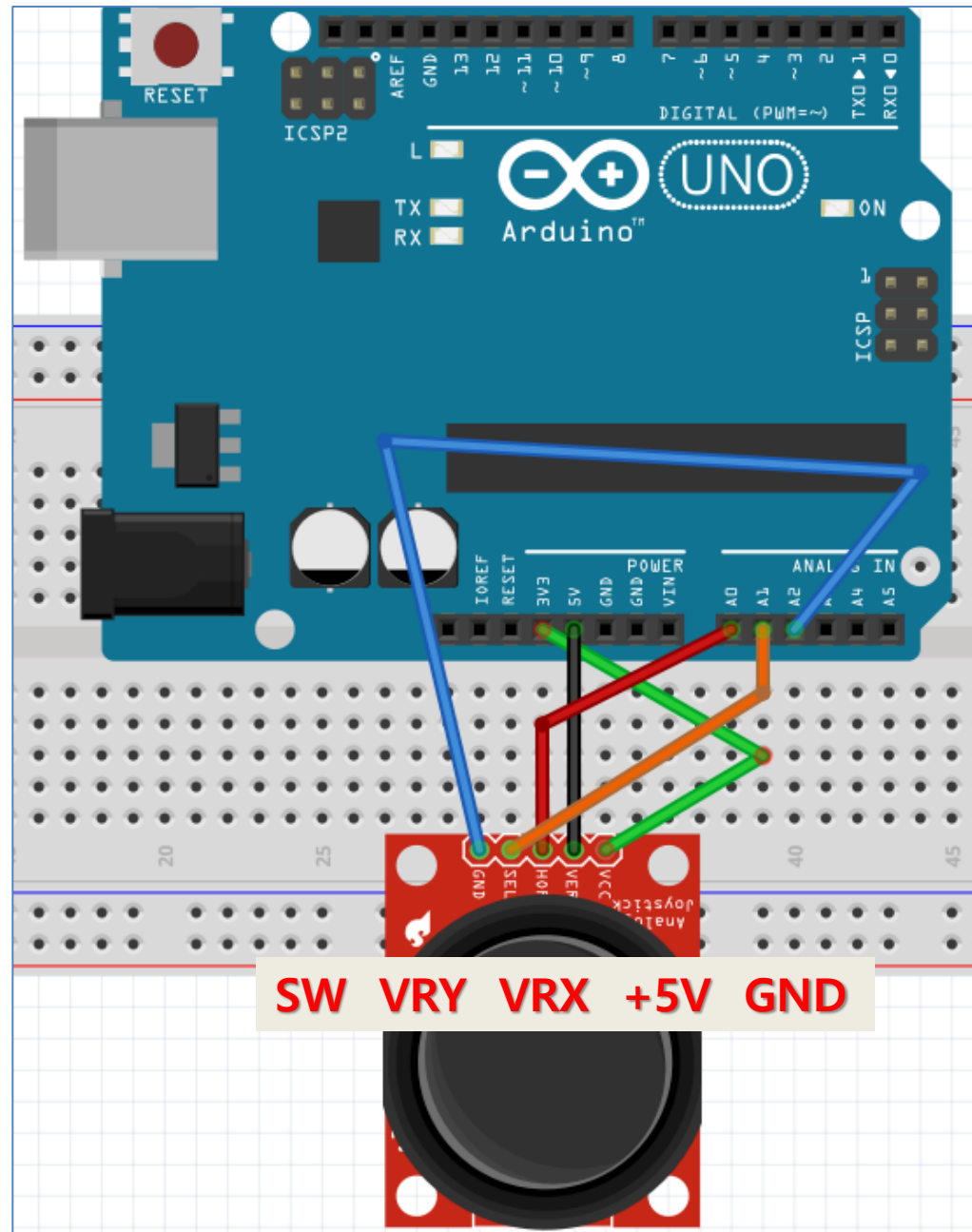


아두이노 1채널 릴레이 모듈 5V용 E4-P26

<https://blog.naver.com/3demp/221057404554>



조이스틱



JoyStic

```
int JoyStic_X = A0;
int JoyStic_Y = A1;
int pin_SW = A2;

void setup() {
  pinMode(pin_SW, INPUT_PULLUP); //스위치가 Pullup상태에서 해제될때 인식
  Serial.begin(9600);
}

void loop() {
  int x, y, z;
  x=analogRead(JoyStic_X);
  y=analogRead(JoyStic_Y);

  Serial.print(x,DEC);
  Serial.print(",");
  Serial.print(y,DEC);
  Serial.print(",");

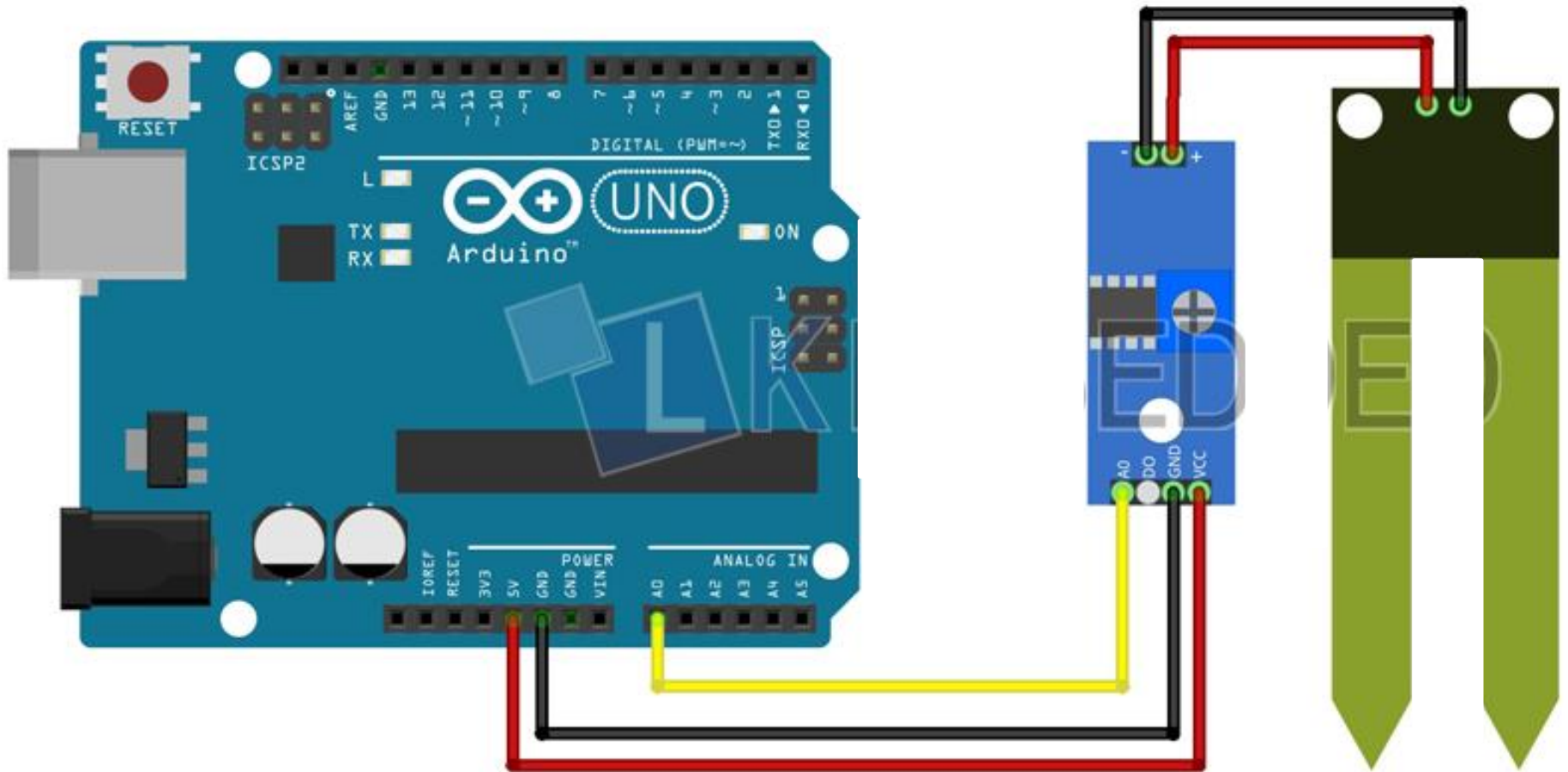
  if(digitalRead ( pin_SW ))
  {
    Serial.println("OFF");
  }else {
    Serial.println("ON");
  }
  delay(100);
}
```

COM3 (Arduino/Genuino Uno)

517,488,OFF
517,488,OFF
517,488,OFF
517,488,OFF
517,488,OFF
517,488,OFF
517,488,OFF
517,488,OFF
517,488,OFF
517,488,OFF
517,489,OFF
517,488,OFF

☒ 자동 스크롤

토양 수분 센서



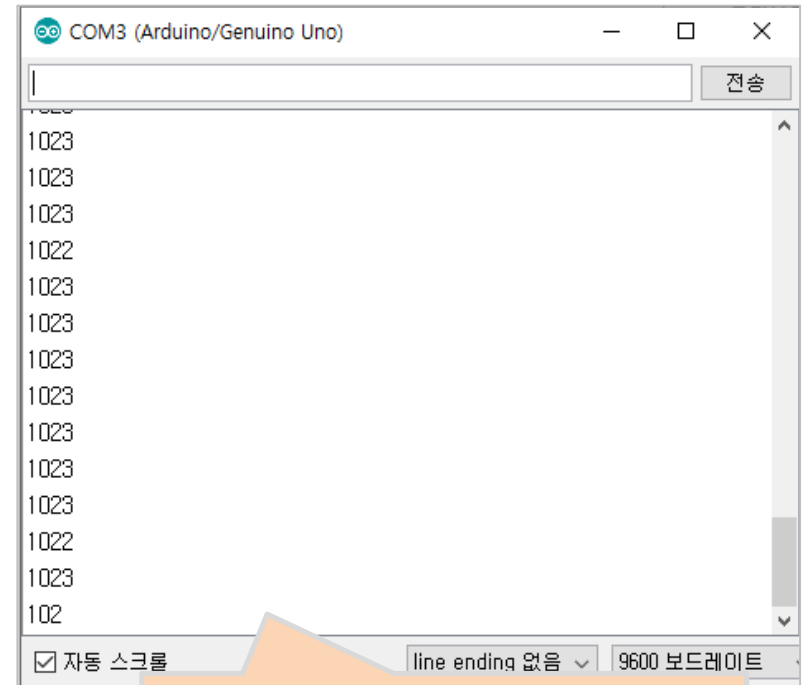
토양 수분 센서

```
unsigned int val;

void setup() {
  Serial.begin(9600);
  pinMode(A0, INPUT);
}

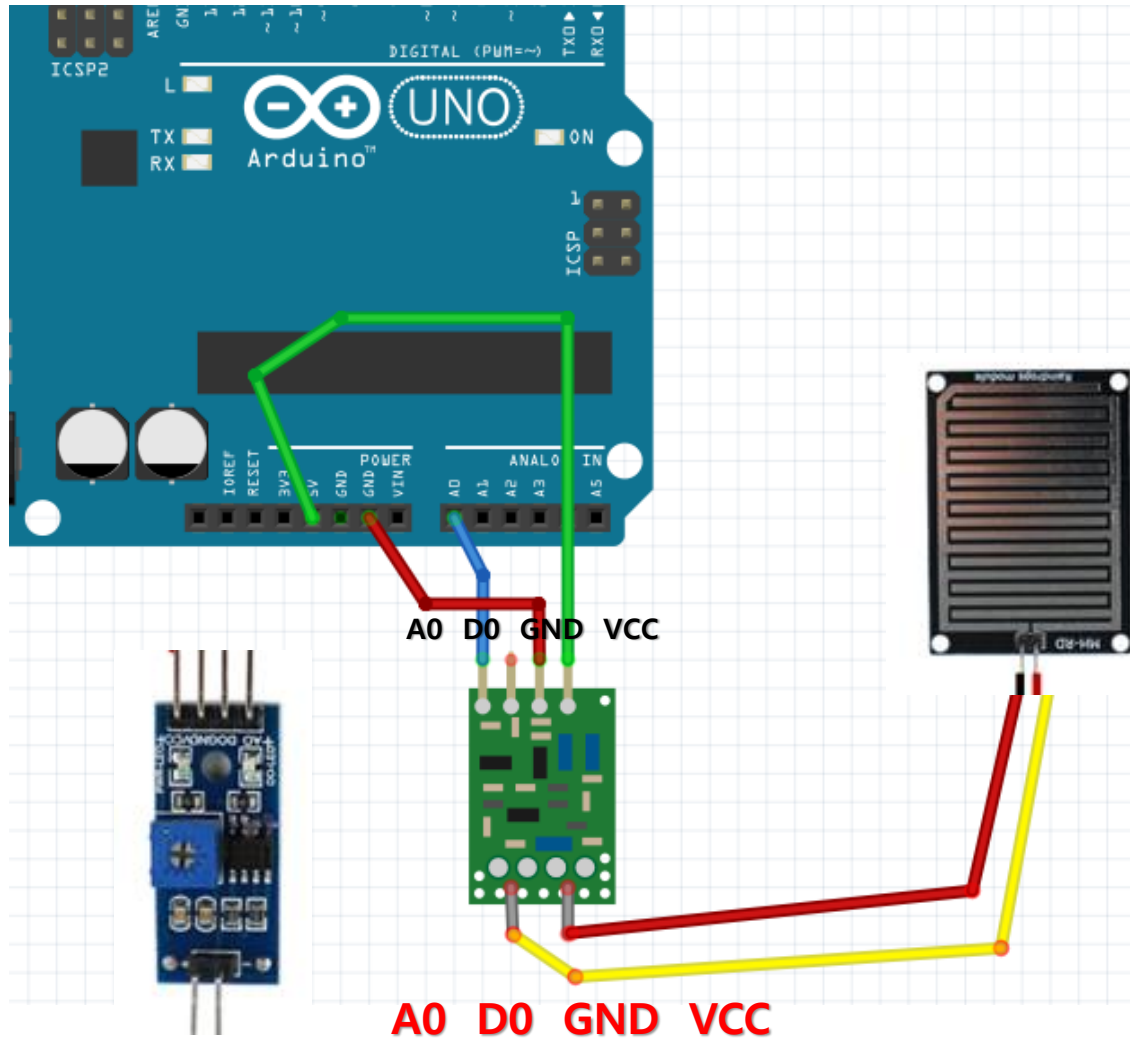
void loop() {
  val = analogRead(A0);
  Serial.println(val);
  delay(100);
}

/*0 ~ 300 : 높은 습도 상태
  300 ~ 700 : 보통 습도 상태
  700 ~ 950 : 건조 상태*/
```



0~300 : 높은 습도
300~700 : 보통 습도
700 ~ 1024 : 건조

빛방울 센서

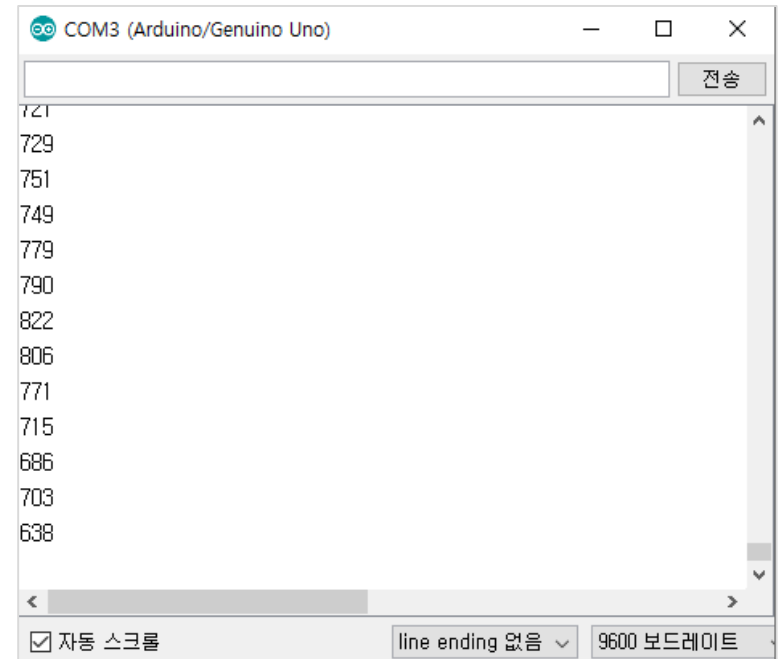


빗방울 센서

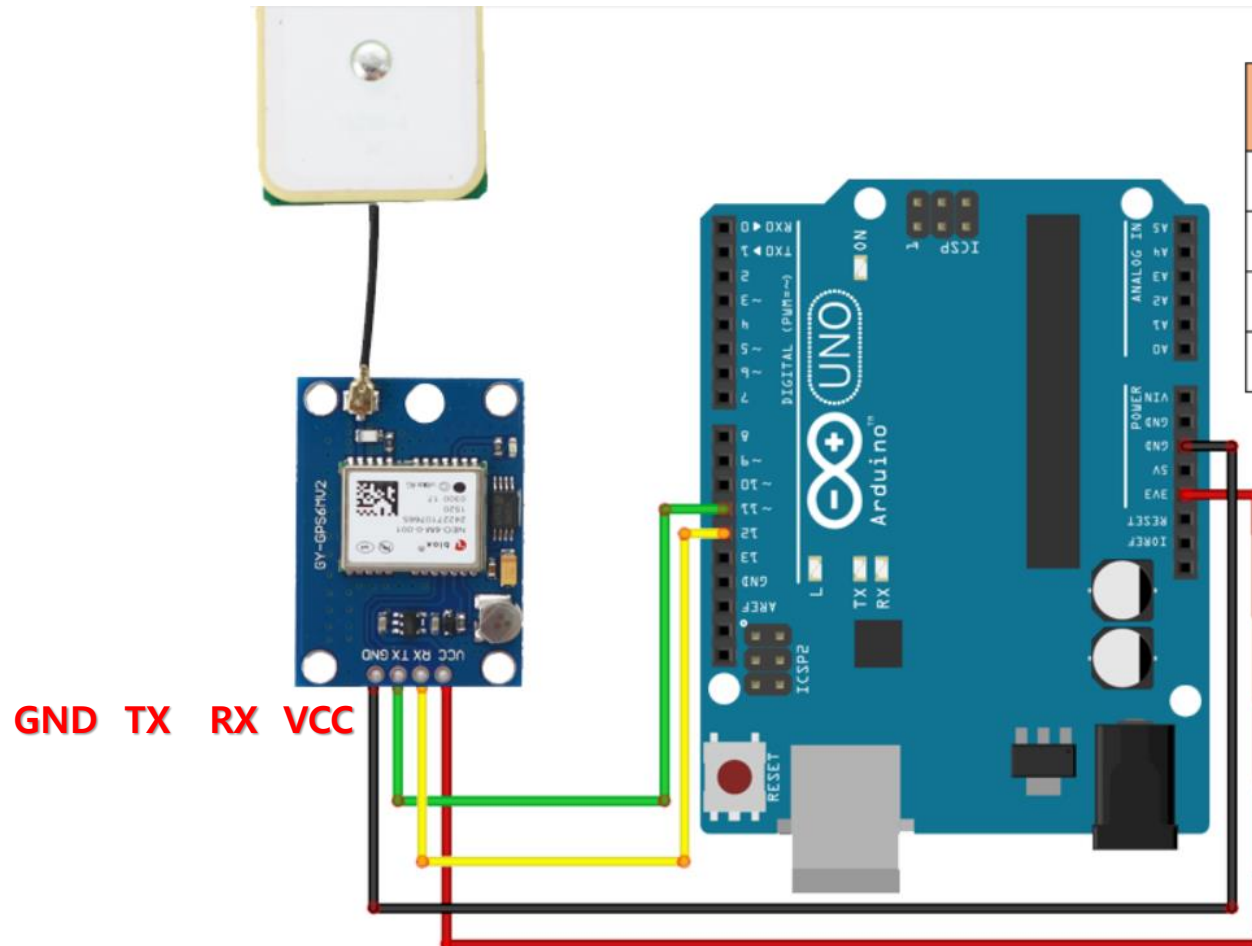
```
int Raindrops_pin = A0;

void setup() {
  Serial.begin(9600);
  pinMode(A0, INPUT);
}

void loop() {
  int val = analogRead(Raindrops_pin);
  Serial.println(val);
  delay(100);
}
```



GPS 모듈



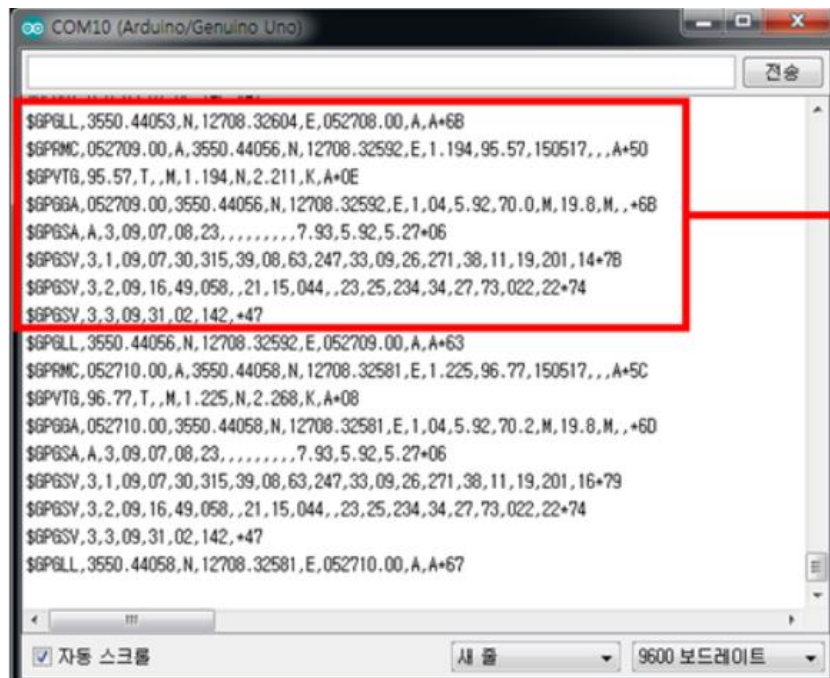
아두이노 우노 (Uno)	GPS 수신 모듈
5V	VCC
GND	GND
D12	RX
D11	TX

GPS 모듈

```
#include <SoftwareSerial.h>
SoftwareSerial gps(11,12);
```

```
void setup() {
  Serial.begin(9600);
  gps.begin(96000);
}
```

```
void loop() {
  if(gps.available()){
    Serial.write(gps.read());
  }
}
```



GPS 신호값

\$GPBOD	Bearing, origin to destination
\$GPBWC	Bearing and distance to waypoint, great circle
• \$GPGGA	Global Positioning System Fix Data
• \$GPGLL	Geographic position, latitude / longitude
• \$GPGSA	GPS DOP and active satellites
• \$GPGSV	GPS Satellites in view
\$GPHDT	Heading, True
\$GPROO	List of waypoints in currently active route
\$GPRMA	Recommended minimum specific Loran-C data
\$GPRMB	Recommended minimum navigation info
• \$GPRMC	Recommended minimum specific GPS / Transit data
\$GPRTE	Routes
\$GPTRF	Transit Fix Data
\$GPSTN	Multiple Data ID
\$GPVBW	Dual Ground / Water Speed
• \$GPVTG	Track made good and ground speed
\$GPWPL	Waypoint location
\$GPXTE	Cross-track error, Measured
\$GPZDA	Date & Time

· 시리얼 모니터를 통해 확인 할 수 있는 신호들은 다음과 같습니다.

\$GPGGA	GPS 의 고정값(기준값)
\$GPGLL	지리적 위치의 위도, 경도
\$GPGSA	GPS 시스템 소프트웨어와 통신위성
\$GPGSV	GPS 위성이 본 경관
\$GPRMC	//권장되는 최소한의 GPS / 데이터 교환
\$GPVTG	지구 표면상의 항공 궤도와 대지 속도

2) 위도와 경도 확인하기

- \$GPGGA는 GPS 신호의 위치, 시간, 속도 등의 다양한 정보를 축약하여 담고 있습니다.
그렇기 때문에 \$GPGGA의 값을 그대로 지도에서 읽어내려 한다면 제대로 된 위치가 나오지 않습니다.
아래 사이트를 통해 \$GPGGA값을 우리도 쉽게 알아 볼 수 있는 위도와 경도값으로 바꿔보겠습니다.

사이트 - <http://www.gonmad.co.uk/nmea.php>

Gonma
www.gonmad.co.uk

Seakoon Online Judge
<http://www.acmicpc.net>

Advertisement loading
...please support our advertisers...

Home | Serious ... | Fun ... | Software ...

Serious Content
[SeakoonOnlineJudge](#)
[SeakoonPC](#)
[Seakoon](#)
[GPS module in Seakoon](#)
[Create nmea from your GPS data](#)
[My Seakoon](#)

The Fun Stuff
[Seakoon](#)
[Seakoon](#)
[Seakoon](#)
[Seakoon](#)
[Seakoon](#)
[Seakoon](#)
[Seakoon](#)
[Seakoon](#)

Software / Downloads
[GPS Receiver](#)
[Seakoon](#)
[GPS](#)
[GPS](#)
[GPS](#)
[GPS](#)
[GPS](#)
[GPS](#)
[GPS](#)
[GPS](#)

GonMad Services
[GPS](#)
[GPS](#)
[GPS](#)

GPS NMEA data to Google Map converter (v4.0)

The page allows you to paste captured raw NMEA data from your GPS device, and generates a Google Map with track markers from that data.
The conversion is all performed by JavaScript and is therefore all client side. That means that your GPS data is NOT sent to our server for processing.
The fact that all processing is performed client-side means that clicking the 'Filter' button below is pretty processor intensive, and during the filtering your CPU use may hit 100% for a while. This is normal.
Once your raw data is filtered, pressing the 'Generate Map' button will generate the map and track. (This may also take a while depending on how many track points need to be plotted)

GPS NMEA Data to Google Maps Converter

NMEA Data Capture

Paste your raw NMEA data here...

\$GPGGA 값을 붙여넣기

Minimum satellites required for good fix: 4

Minimum time between fixes: 1 secs

Filter data | Clear data | Paste from Clipboard

Filter information:
Essential fix data found: 0

Essential Fix data (filtered from above):
Paste your GPGGA sentences here (filtered data)...


\$GPGGA 값을 붙여넣기

Generate Map | Clear data | Copy to clipboard

Latitude =

Longitude =

- 각각의 노란 박스가 있는 곳에 \$GPGGA를 붙여넣기 하고 **Filter data**를 클릭합니다.
 'Fix data값을 찾았다'는 표시가 뜬다면 **Generate Map**을 클릭합니다.



www.gonmad.co.uk

Advertisement loading
...please support our advertisers.

[Home](#)
[Serious ...](#)
[Fun ...](#)
[Software ...](#)

Services / Content

- [BicycleGPS](#)
- [GPS Data](#)
- [GPS NMEA to Google](#)
- [GPS Data from your](#)
- [GPS Data](#)
- [My Account](#)

The Fun Stuff

- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)

Software / Downloads

- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)

GoMad Services

- [GPS Data](#)
- [GPS Data](#)
- [GPS Data](#)

GPS NMEA data to Google Map converter (v4.0)

This page allows you to paste captured raw NMEA data from your GPS device, and generates a Google Map with track markers from that data. The conversion is all performed by JavaScript and is therefore all client-side. That means that your GPS data is NOT sent to our server for processing. The fact that all processing is performed client-side means that clicking the 'Filter' button below is pretty processor intensive, and during the filtering your CPU use may hit 100% for a while. This is normal. Once your raw data is filtered, pressing the 'Generate Map' button will generate the map and track. (This may also take a while depending on how many track points need to be plotted)

NMEA Data Capture

\$GPGGA,05.2750.00,3990.44080,N,12700.33374,E,1,04,S,09.7,1.0,M,19.0,M,0.0,00000000.0

Filter information:
Essential fix data found: 1 (1/1)

Essential Fix data (Filtered from above):

\$GPGGA,05.2750.00,3990.44080,N,12700.33374,E,1,04,S,09.7,1.0,M,19.0,M,0.0,00000000.0

Calculating:
Latitude = 35.84068166666667
Longitude = 127.138779

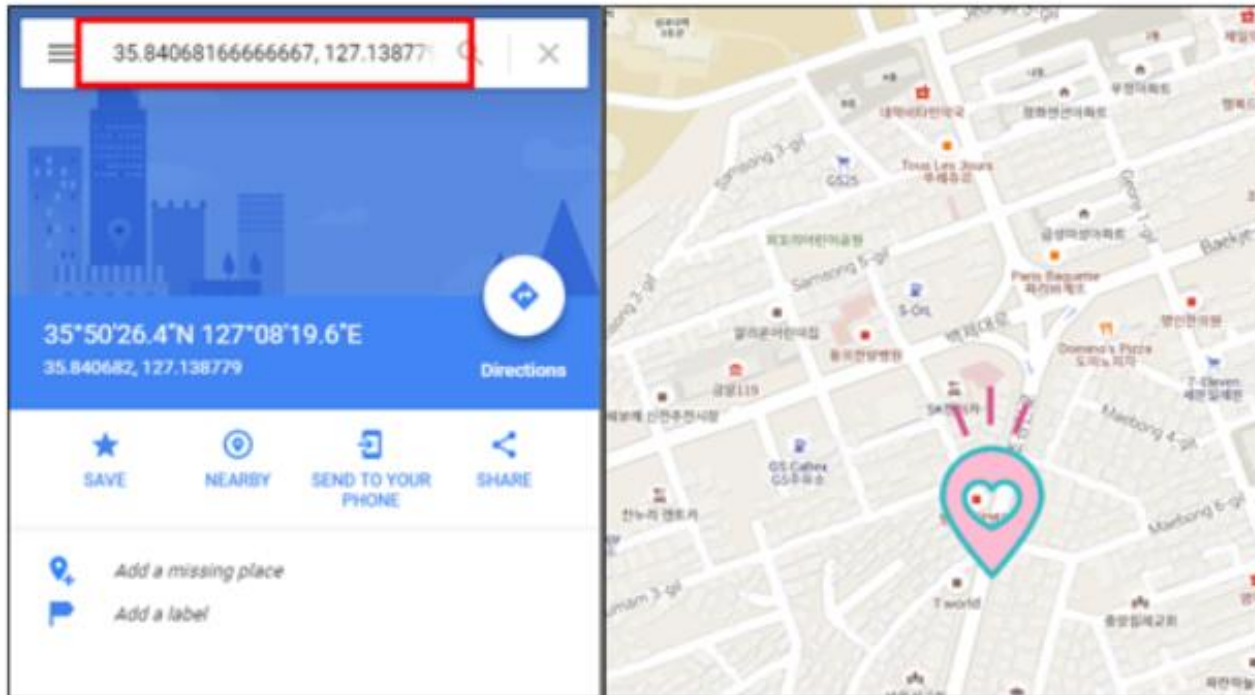
Calculating:
Latitude = 35.84068166666667
Longitude = 127.138779

☒ Show marker for selected track point? (Click marker for full info)
☒ Hide selected track point marker while animating?

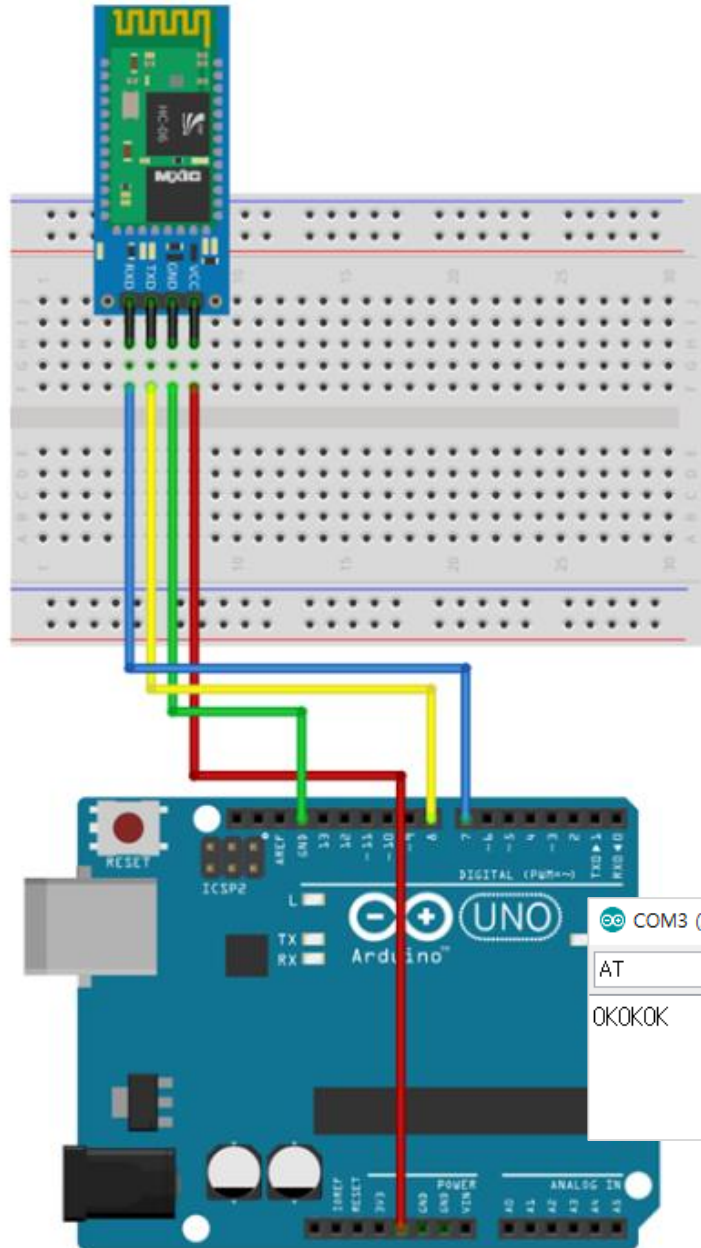
Latitude : 위도
Longitude : 경도

3) 구글맵에서 위치 검색하기

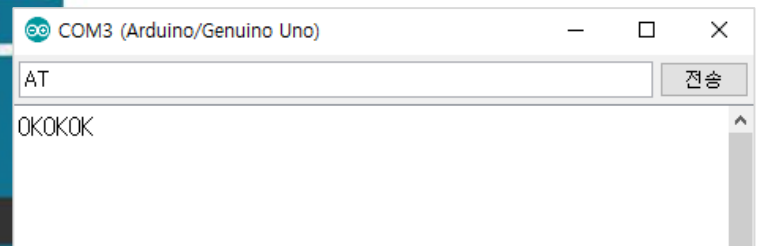
- 위도와 경도를 구글맵에 붙여넣기하여 현재 위치를 확인합니다.
- 실내에서 테스트했음에도 1~2m에 불과한 오차를 보입니다.
(GPS는 일반적으로 실외보다 실내에서의 오차 범위가 크게 나타납니다.)



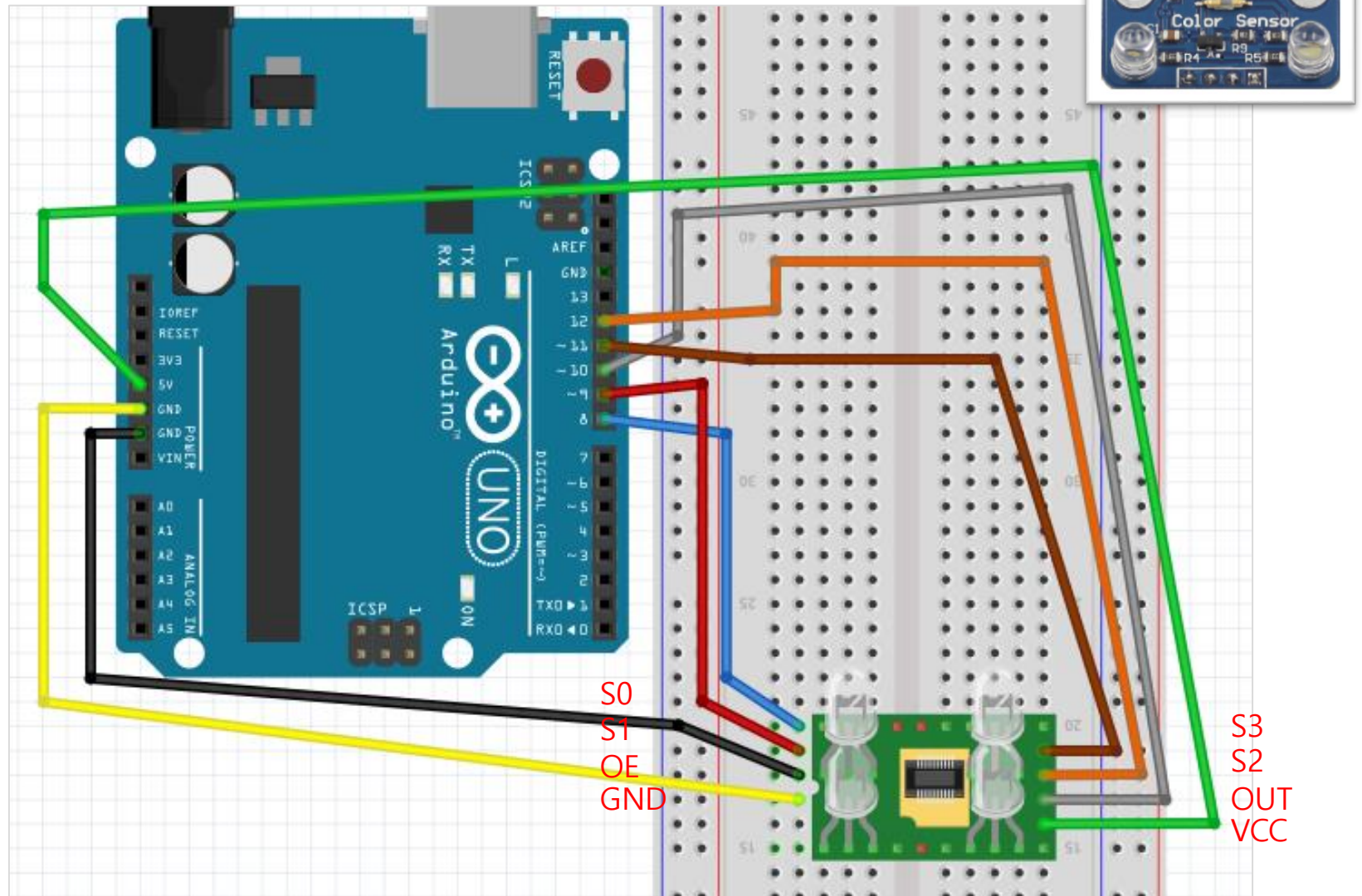
블루투스



아두이노 우노보드	HC-06
5V	VCC
GND	GND
D8	TXD
D7	RXD



컬러 센서



컬러센서

```
1 const int s0 = 8;
2 const int s1 = 9;
3 const int s2 = 12;
4 const int s3 = 11;
5 const int out = 10;
6
7 int red = 0;
8 int green = 0;
9 int blue = 0;
10
11 void setup()
12 {
13   pinMode(s0, OUTPUT);
14   pinMode(s1, OUTPUT);
15   pinMode(s2, OUTPUT);
16   pinMode(s3, OUTPUT);
17   pinMode(out, INPUT);
18
19   Serial.begin(9600);
20   digitalWrite(s0, HIGH);
21   digitalWrite(s1, HIGH);
22 }
```

```
23
24 void loop()
25 {
26   color(); //Chama a rotina que le as cores
27   Serial.print("Red :");
28   Serial.print(red, DEC);
29   Serial.print(" Green :");
30   Serial.print(green, DEC);
31   Serial.print(" Blue :");
32   Serial.print(blue, DEC);
33   Serial.println();
34   delay(2000);
35 }
36 void color()
37 {
38   digitalWrite(s2, LOW);
39   digitalWrite(s3, LOW);
40   red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
41   digitalWrite(s3, HIGH);
42   blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
43   digitalWrite(s2, HIGH);
44   green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
45 }
```

S0	S1	S2	S3	
HIGH	HIGH	LOW	LOW	→ RED
HIGH	HIGH	LOW	HIGH	→ BLUE
HIGH	HIGH	HIGH	HIGH	→ GREEN

