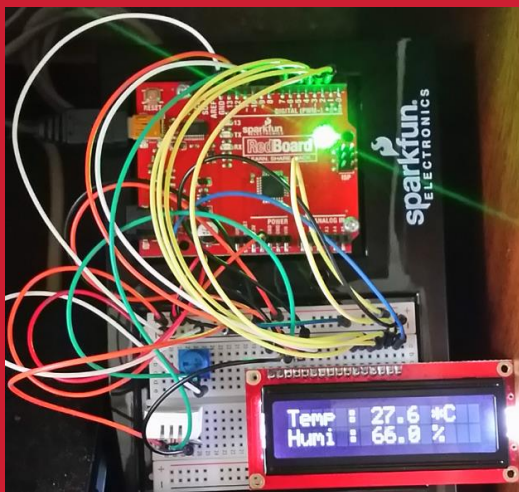
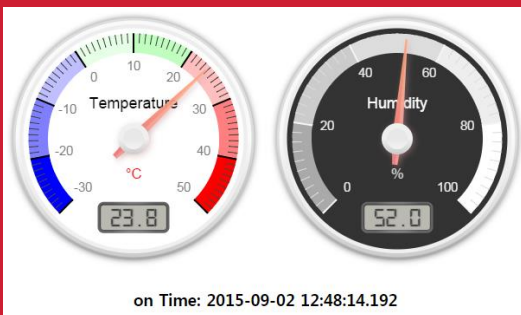




HW-SW-Connectivity

[wk06]

Arduino Sensors

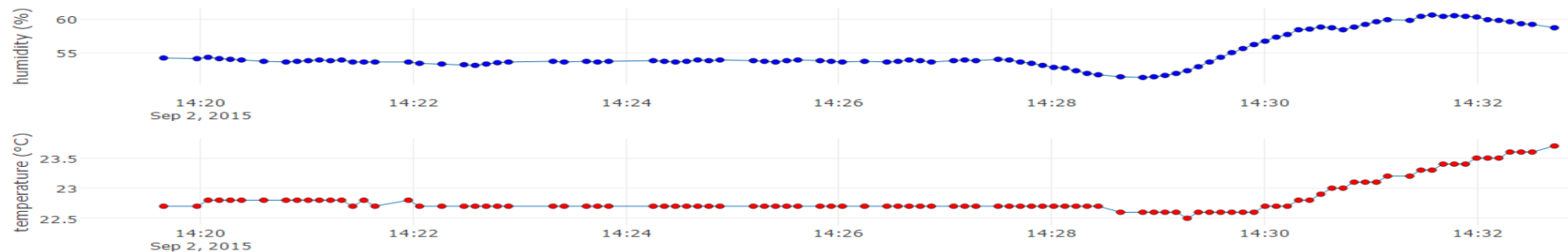


Basic HW and SW Integration using
Arduino & Javascript

COMSI, INJE University

2nd semester, 2017

Email : yish@inje.ac.kr



실기시험

1. 일시: 10월 18일(수) 오후 4시 ~ 5시
2. 장소: E531 (전산실습실)
3. Express server 설치/구동 및 라우팅 화면 구성
4. 범위 : node.js, express server, Sublime Text 3

[Tip] USB 메모리에 portable SW를 설치해서 준비.

[PC 사용법] local hard로 로그인

필기시험

1. 일시: 10월 18일(수) 오후 5시 ~ 6시
2. 장소: E429
3. 20문제 (객관식 및 단답형)
4. 범위 : node.js ~ arduino



Arduino



Home

Buy

Download

Products ▼

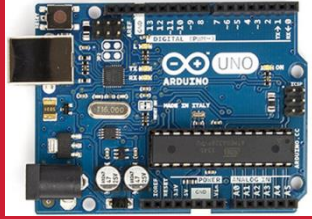
Learning ▼

Forum

Support ▼

Blog

<https://www.arduino.cc/>



[Practice]

◆ [wk05]

- **Arduino basic circuits**
- **Complete your project**
- **Upload file name : AAnn_Rpt04.zip**

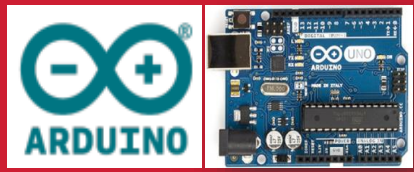
◆ [Target of this week]

- Complete your projects
- Save your outcomes and compress 4 figures and 2 codes.

제출파일명 : AAnn_Rpt04.zip

- 압축할 파일들

- ① **AAnn_Blink.png**
- ② **AAnn_2Leds.ino**
- ③ **AAnn_4Leds.ino**
- ④ **AAnn_Sawtooth.png**
- ⑤ **AAnn_AnalogVoltage.png**
- ⑥ **AAnn_AnalogVoltage_Plot.png**

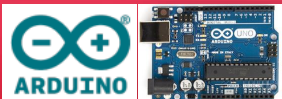


1. Arduino SW: IDE



HOME BUY SOFTWARE PRODUCTS LEARNING FORUM SUPPORT BLOG

<https://www.arduino.cc/>



A1.1 Arduino IDE

Download the Arduino IDE



ARDUINO 1.8.4

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Windows app

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

[Release Notes](#)

[Source Code](#)

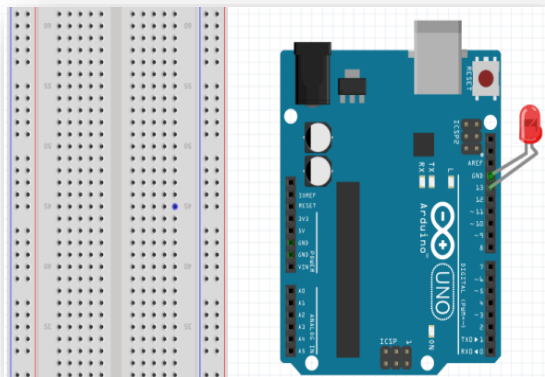
[Checksums \(sha512\)](#)



LED



A2.1.2 blink [modified your code, save it]



**Connect LED to
D13 & GND**



AA00_Blink

```
1 /*
2  Blink by AAnn
3  Turns an LED on for one second, then off for one second, repeatedly.
4  */
5  int pinNum = 13;  // D13
6
7  // the setup function runs once when you press reset or power the board
8  void setup() {
9    // initialize digital pin LED_BUILTIN as an output.
10    pinMode(pinNum, OUTPUT);
11  }
12
13  // the loop function runs over and over again forever
14  void loop() {
15    digitalWrite(pinNum, HIGH);  // turn the LED on (HIGH is the voltage level)
16    delay(1000);                  // wait for a second
17    digitalWrite(pinNum, LOW);   // turn the LED off by making the voltage LOW
18    delay(1000);                  // wait for a second
19  }
```

**Save as
AAnn_Blink.png**



Serial monitor & plotter



DIY: Sawtooth wave

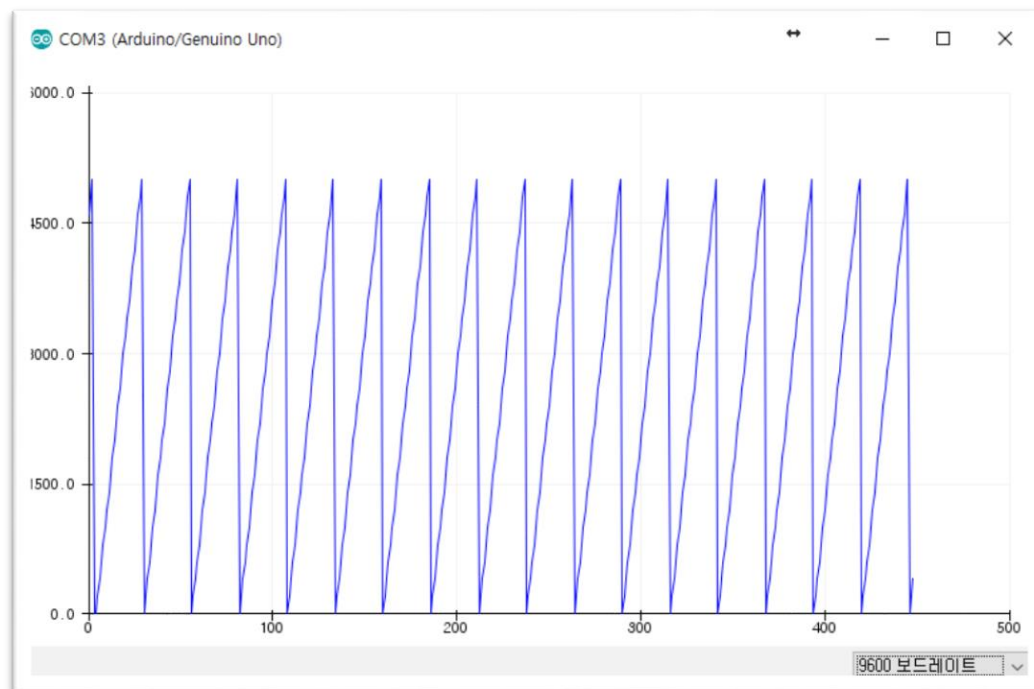
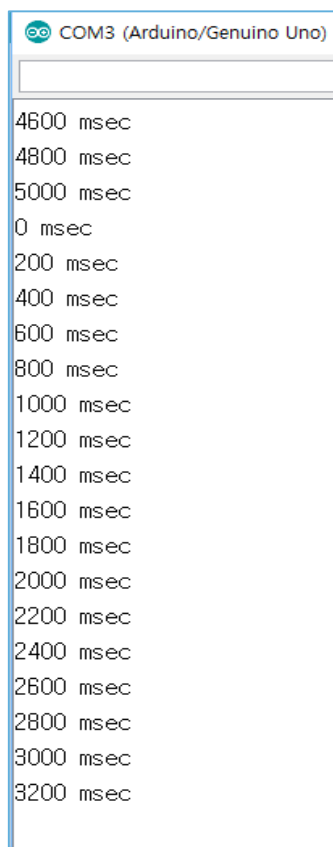
[DIY]

delay를 0.2초로 설정후

5초 마다 number를 초기화하여

시리얼플로터로 톱니파를 발생.

시간은 ms로 계산해서 출력



Save as

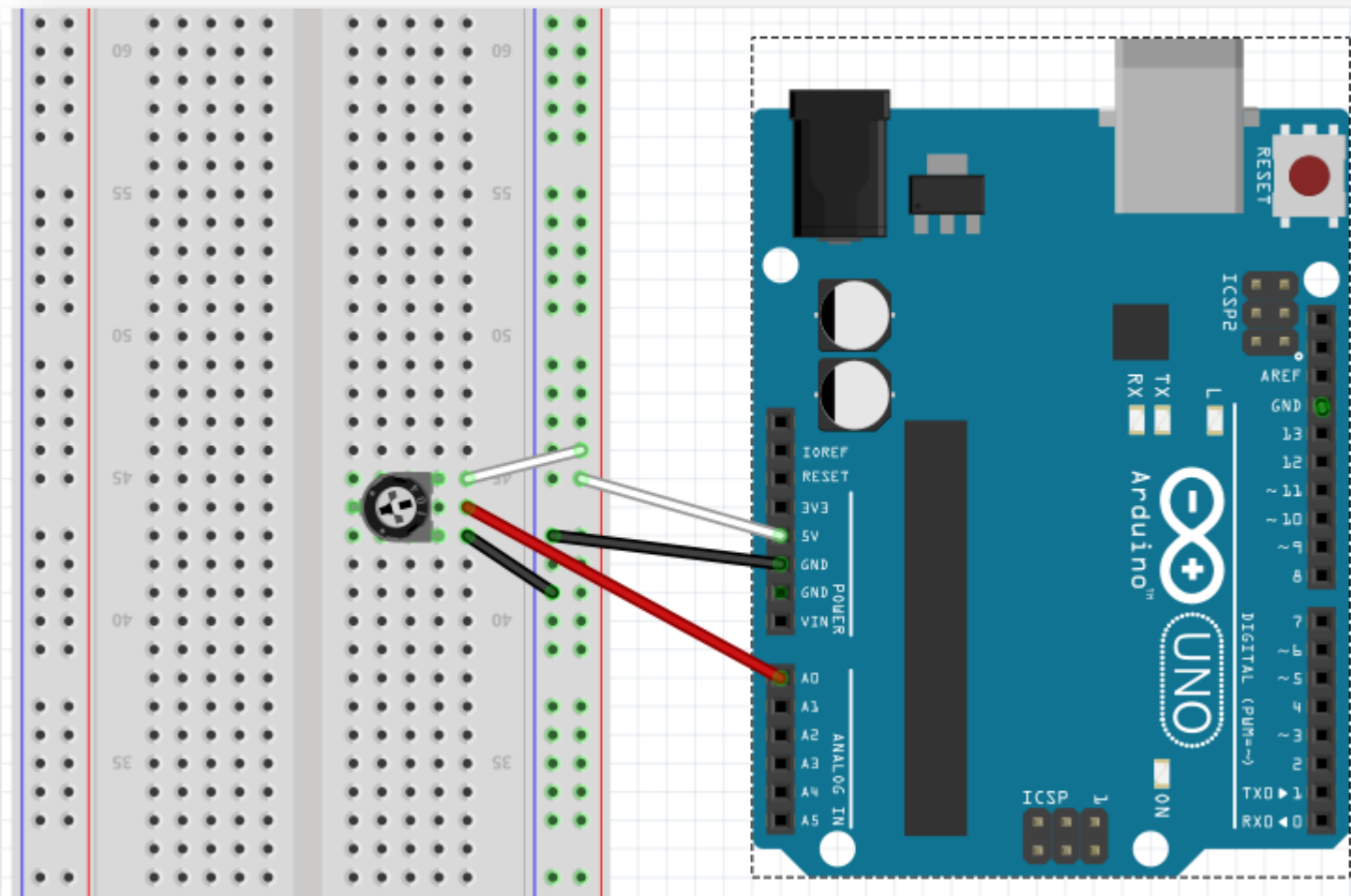
AAnn_Sawtooth.png



Analog Signal

A2.5.1 AnalogReadSerial (circuit)

Standard potentiometer (가변 저항기)





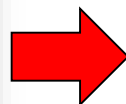
A2.5.4 ReadAnalogVoltage

Serial monitor

```
COM7 (Arduino/Genuino Uno)

Present value (0 ~ 1023) : 0
Present value (0 ~ 1023) : 0
Present value (0 ~ 1023) : 6
Present value (0 ~ 1023) : 14
Present value (0 ~ 1023) : 82
Present value (0 ~ 1023) : 199
Present value (0 ~ 1023) : 300
Present value (0 ~ 1023) : 429
Present value (0 ~ 1023) : 525
Present value (0 ~ 1023) : 634
Present value (0 ~ 1023) : 819
Present value (0 ~ 1023) : 989
Present value (0 ~ 1023) : 1023
Present value (0 ~ 1023) : 1023
Present value (0 ~ 1023) : 1023
```

☒ 자동 스크롤 No line ending 9600 보드 레이아웃



```
COM4

4A00, Present voltage (0.0 ~ 5.0) : 5.00
4A00, Present voltage (0.0 ~ 5.0) : 3.68
4A00, Present voltage (0.0 ~ 5.0) : 2.42
4A00, Present voltage (0.0 ~ 5.0) : 1.37
4A00, Present voltage (0.0 ~ 5.0) : 0.00
4A00, Present voltage (0.0 ~ 5.0) : 0.00
4A00, Present voltage (0.0 ~ 5.0) : 0.00
4A00, Present voltage (0.0 ~ 5.0) : 0.88
4A00, Present voltage (0.0 ~ 5.0) : 1.47
4A00, Present voltage (0.0 ~ 5.0) : 2.11
4A00, Present voltage (0.0 ~ 5.0) : 2.79
4A00, Present voltage (0.0 ~ 5.0) : 3.38
4A00, Present voltage (0.0 ~ 5.0) : 3.99
4A00, Present voltage (0.0 ~ 5.0) : 4.91
4A00, Present voltage (0.0 ~ 5.0) : 5.00
4A00, Present voltage (0.0 ~ 5.0) : 5.00
4A00, Present voltage (0.0 ~ 5.0) : 4.68
4A00, Present voltage (0.0 ~ 5.0) : 3.88
4A00, Present voltage (0.0 ~ 5.0) : 3.35
```



A2.5.6 ReadAnalogVoltage

Hint code

AA00_AnalogRead

```
1 /*
2  AnalogReadSerial
3
4  Reads an analog input on pin 0, prints the result to the Serial Monitor.
5  Attach the center pin of a potentiometer to pin A0,
6  and the outside pins to +5V and ground.
7  */
8
9  // the setup routine runs once when you press reset:
10 void setup() {
11   // initialize serial communication at 9600 bits per second:
12   Serial.begin(9600);
13 }
14
15 // the loop routine runs over and over again forever:
16 void loop() {
17   // read the input on analog pin 0:
18   int sensorValue = analogRead(A0);
19   //float voltage = map(sensorValue, 0, 1023, 0.0, 5.0); // map 0~1023 to 0~5
20   float voltage = sensorValue*(5.0/1023.0);
21   // print out the value you read:
22   Serial.print("AA00, Present voltage (0.0 ~ 5.0) : ");
23   Serial.println(voltage);
24   delay(500); // delay in between reads for stability
25 }
```

COM4

```
AA00, Present voltage (0.0 ~ 5.0) : 5.00
AA00, Present voltage (0.0 ~ 5.0) : 5.00
AA00, Present voltage (0.0 ~ 5.0) : 3.91
AA00, Present voltage (0.0 ~ 5.0) : 2.76
AA00, Present voltage (0.0 ~ 5.0) : 1.59
AA00, Present voltage (0.0 ~ 5.0) : 0.00
AA00, Present voltage (0.0 ~ 5.0) : 0.00
AA00, Present voltage (0.0 ~ 5.0) : 0.00
AA00, Present voltage (0.0 ~ 5.0) : 0.81
AA00, Present voltage (0.0 ~ 5.0) : 1.89
AA00, Present voltage (0.0 ~ 5.0) : 2.92
AA00, Present voltage (0.0 ~ 5.0) : 3.57
AA00, Present voltage (0.0 ~ 5.0) : 4.29
AA00, Present voltage (0.0 ~ 5.0) : 5.00
AA00, Present voltage (0.0 ~ 5.0) : 4.99
AA00, Present voltage (0.0 ~ 5.0) : 4.62
AA00, Present voltage (0.0 ~ 5.0) : 3.21
AA00, Present voltage (0.0 ~ 5.0) : 1.82
AA00, Present voltage (0.0 ~ 5.0) : 1.80
AA00, Present voltage (0.0 ~ 5.0) : 1.80
AA00, Present voltage (0.0 ~ 5.0) : 1.80
```

☒ 자동 스크롤

A2.5.7 ReadAnalogVoltage

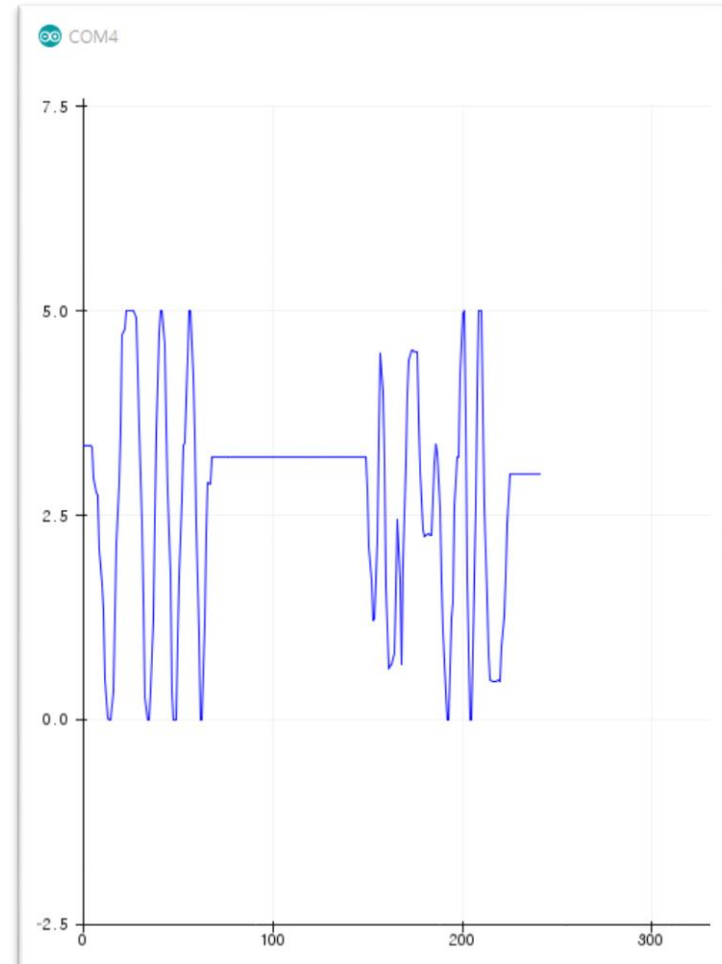
Result

COM4

```

%A00, Present voltage (0.0 ~ 5.0) : 5.00
%A00, Present voltage (0.0 ~ 5.0) : 3.68
%A00, Present voltage (0.0 ~ 5.0) : 2.42
%A00, Present voltage (0.0 ~ 5.0) : 1.37
%A00, Present voltage (0.0 ~ 5.0) : 0.00
%A00, Present voltage (0.0 ~ 5.0) : 0.00
%A00, Present voltage (0.0 ~ 5.0) : 0.00
%A00, Present voltage (0.0 ~ 5.0) : 0.88
%A00, Present voltage (0.0 ~ 5.0) : 1.47
%A00, Present voltage (0.0 ~ 5.0) : 2.11
%A00, Present voltage (0.0 ~ 5.0) : 2.79
%A00, Present voltage (0.0 ~ 5.0) : 3.38
%A00, Present voltage (0.0 ~ 5.0) : 3.99
%A00, Present voltage (0.0 ~ 5.0) : 4.91
%A00, Present voltage (0.0 ~ 5.0) : 5.00
%A00, Present voltage (0.0 ~ 5.0) : 5.00
%A00, Present voltage (0.0 ~ 5.0) : 4.68
%A00, Present voltage (0.0 ~ 5.0) : 3.88
%A00, Present voltage (0.0 ~ 5.0) : 3.35
  
```

Save as
AAnn_AnalogVoltage.png



Save as
AAnn_AnalogVoltage_Plot.png

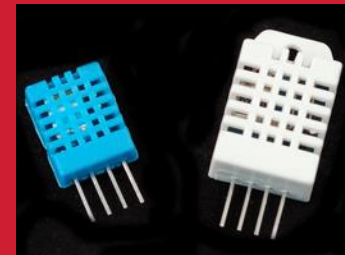
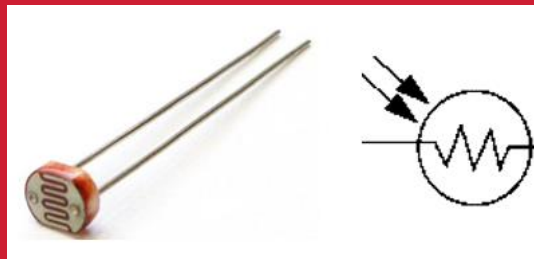
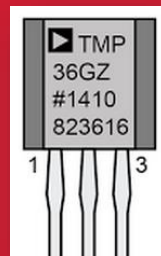
A2.5.6-1 ReadAnalogVoltage using f_map()

Hint code : f_map() instead of map()

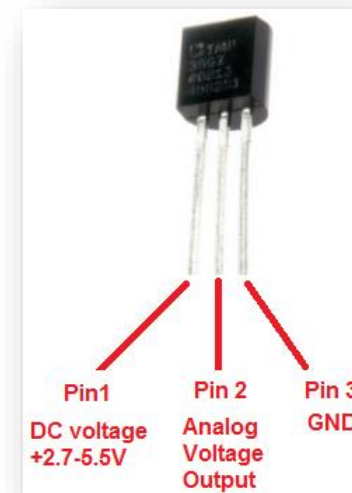
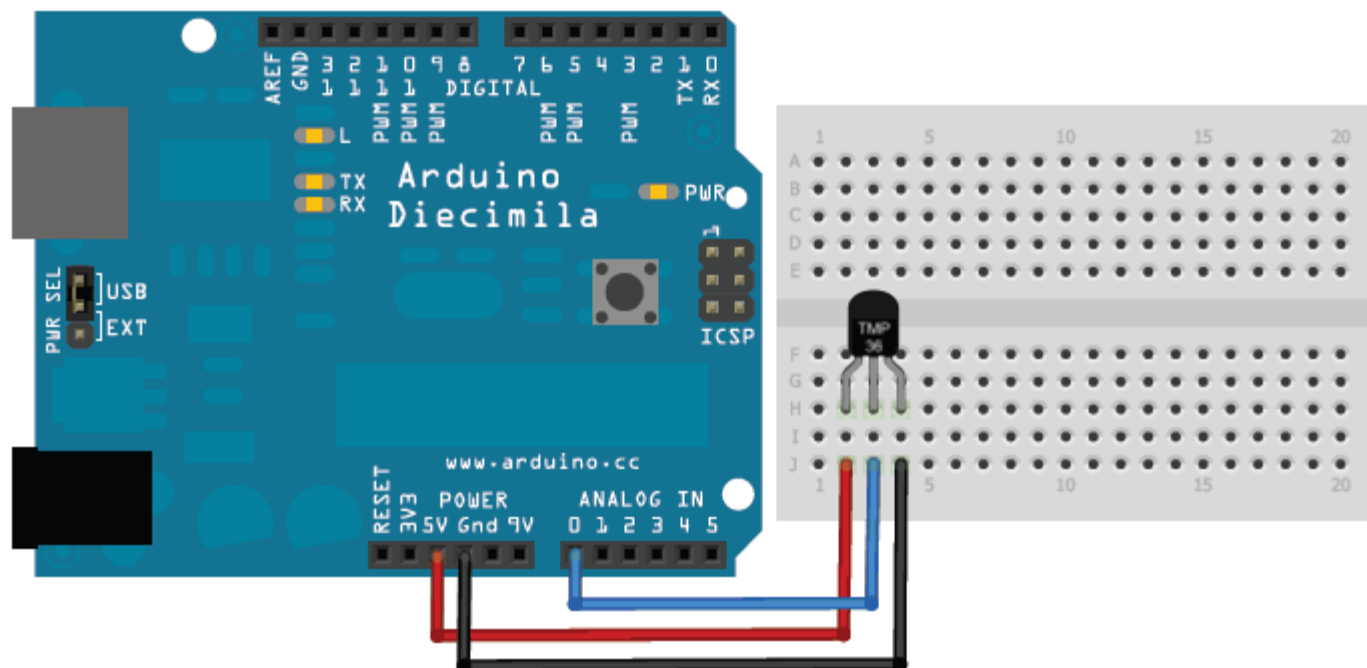
```
AA00_AnalogRead_fmap$
/*
8
9 // the setup routine runs once when you press reset:
10 void setup() {
11   // initialize serial communication at 9600 bits per second:
12   Serial.begin(9600);
13 }
14
15 // the loop routine runs over and over again forever:
16 void loop() {
17   // read the input on analog pin 0:
18   int sensorValue = analogRead(A0);
19   //float voltage = map(sensorValue, 0, 1023, 0.0, 5.0); // map 0~1023 to 0~5
20   //float voltage = sensorValue*(5.0/1023.0);
21   float voltage = f_map(sensorValue, 0, 1023, 0.0, 5.0); // map 0~1023 to 0~5
22   // print out the value you read:
23   Serial.print("AA00, Present voltage (0.0 ~ 5.0) : ");
24   Serial.println(voltage);
25   delay(500);          // delay in between reads for stability
26 }
27
28 float f_map(long x, long in_min, long in_max, float out_min, float out_max)
29 {
30   return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
31 }
```



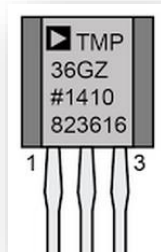
Arduino Sensors



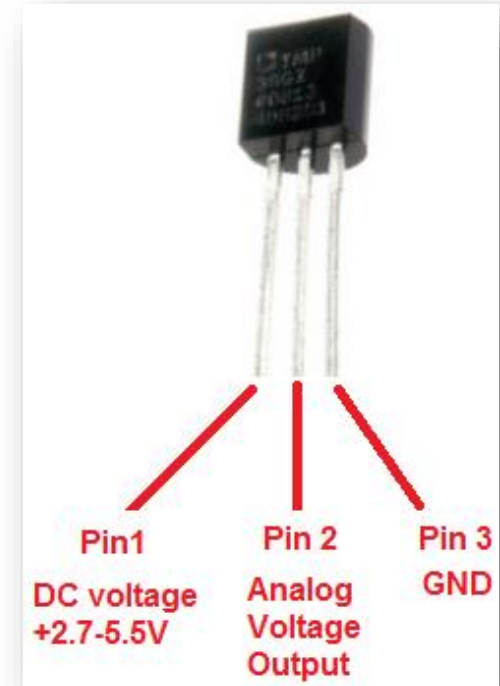
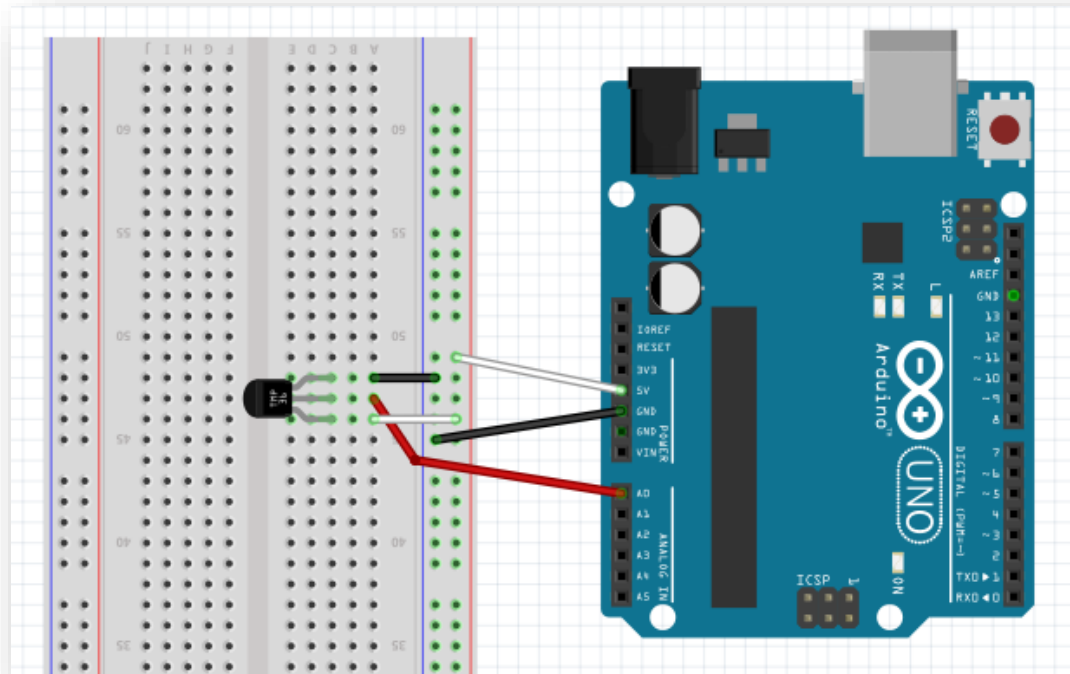
A3.1.1 Temperature sensor [TMP36]



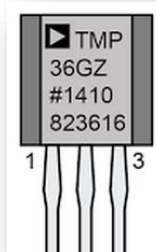
Parts : TMP36



A3.1.2 Temperature sensor [TMP36]



Parts : TMP36



- **Size:** TO-92 package (about 0.2" x 0.2" x 0.2") with three leads
- **Price:** \$2.00 at the [Adafruit shop](#)
- **Temperature range:** -40°C to 150°C / -40°F to 302°F
- **Output range:** 0.1V (-40°C) to 2.0V (150°C) but accuracy decreases after 125°C
- **Power supply:** 2.7V to 5.5V only, 0.05 mA current draw

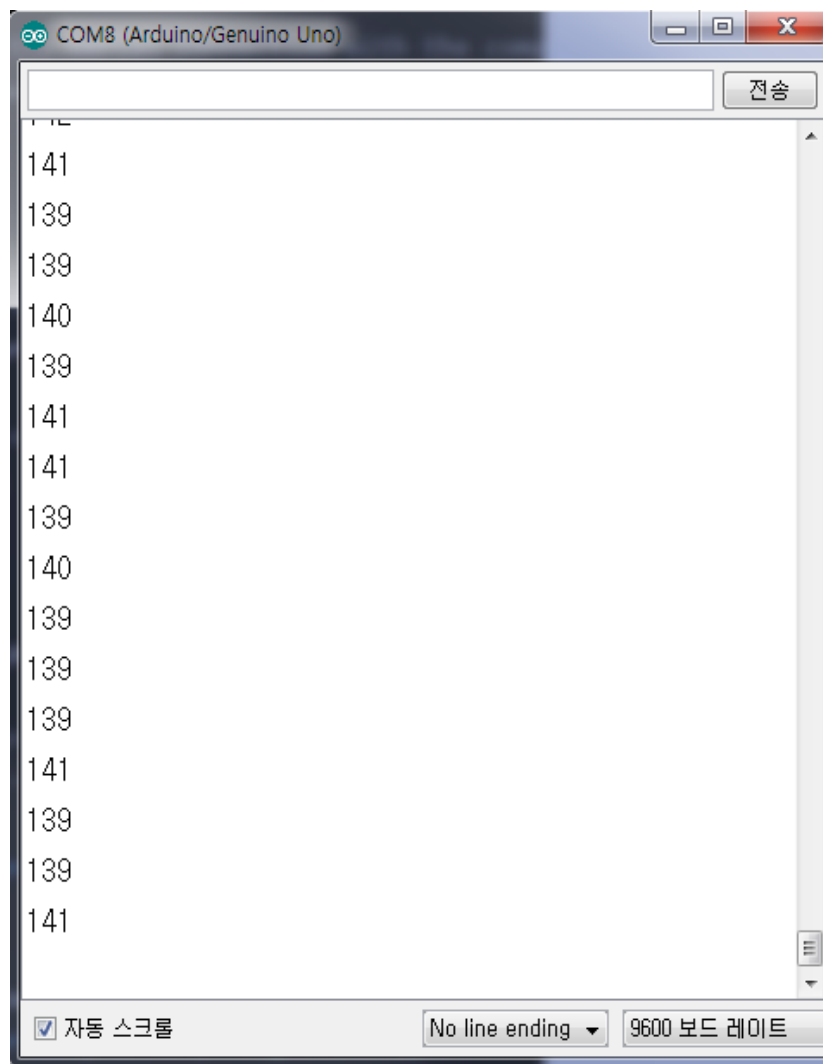


A3.1.3 Temperature sensor [TMP36]

Simple code

```
TMP36 $
1 //
2 //  AA00, TMP36 sensor
3 //
4
5 #define TEMP_INPUT 0
6 // or  int TEMP_INPUT = 0;
7
8 void setup() {
9   Serial.begin(9600);
10 }
11
12 void loop() {
13
14   int value = analogRead(TEMP_INPUT);
15   Serial.println(value);
16
17   delay(1000);
18 }
```

Serial output (0 ~ 1023)



A3.1.4 Temperature sensor [TMP36]

Sensor property

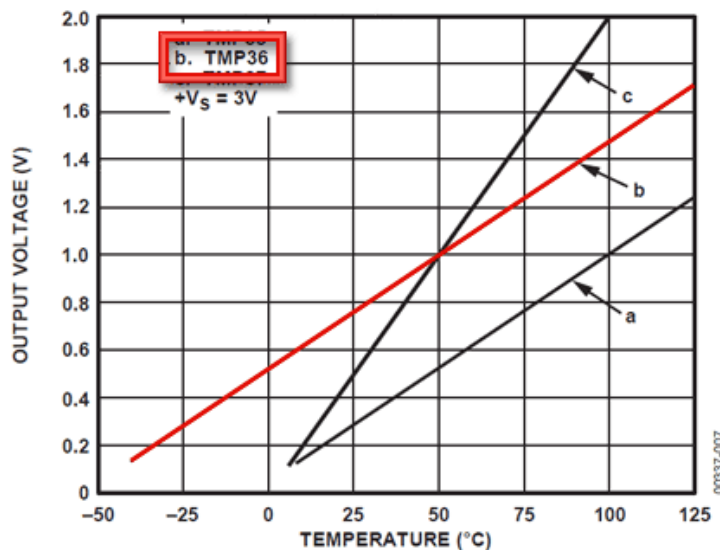


Figure 6. Output Voltage vs. Temperature

Temperature conversion

$$\text{Temp (}^{\circ}\text{C)} = (\text{Vout} - 500) / 10$$

$$\text{Vout (mV)} = \text{value} * (5000 / 1023)$$

$$(0 \leq \text{value} \leq 1023)$$



A3.1.5 Temperature sensor [TMP36]

Working code

```
TMP36
10 }
11
12 void loop() {
13   //getting the voltage reading from the temperature sensor
14   int value = analogRead(TEMP_INPUT);
15   Serial.print("AA00, value = ");
16   Serial.print(value);
17   Serial.print(" : ");
18
19   // converting that reading to voltage
20   float voltage = value * 5.0 * 1000; // in mV
21   voltage /= 1023.0;
22
23   // print out the voltage
24   Serial.print(voltage);
25   Serial.print(" mV, ");
26
27   // now print out the temperature
28   float temperatureC = (voltage - 500) / 10 ;
29   Serial.print(temperatureC);
30   Serial.println(" degrees C");
31
32   delay(1000);
33 }
```

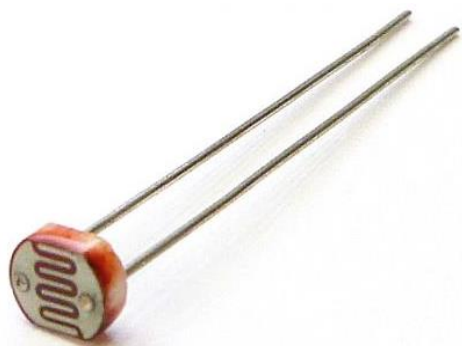
Serial output (°C)

```
COM4
AA00, value = 131 : 640.27 mV, 14.03 degrees C
AA00, value = 130 : 635.39 mV, 13.54 degrees C
AA00, value = 132 : 645.16 mV, 14.52 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 129 : 630.50 mV, 13.05 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 130 : 635.39 mV, 13.54 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 132 : 645.16 mV, 14.52 degrees C
AA00, value = 129 : 630.50 mV, 13.05 degrees C
AA00, value = 132 : 645.16 mV, 14.52 degrees C
AA00, value = 129 : 630.50 mV, 13.05 degrees C
AA00, value = 130 : 635.39 mV, 13.54 degrees C
AA00, value
AA00, value
```

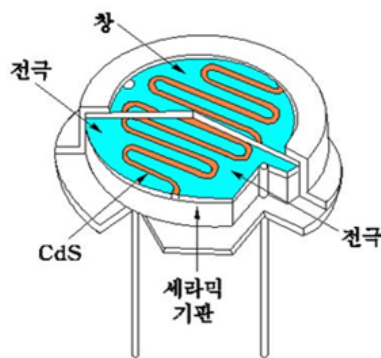
Save as

AAnn_TMP36.png

CdS 센서- photoresistor

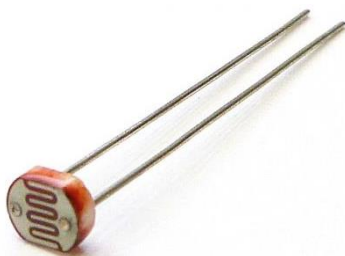


CDS특성



1. 감도
- 빛의 파장에 따라 감도가 다름
2. 허용손실
- 비교적 큰 전류를 흘릴 수 있음
3. 암 전류
- 빛이 없어도 약간의 전류가 흐름
4. 명 전류
- 빛을 비추면 흐르는 전류
5. 응답특성
- 응답 시간 지연
- 빛의 세기에 따라 응답시간 다름
6. 가변저항
- 빛에 따른 가변저항

CdS 센서 - photoresistor



- ✓ CdS 분말을 세라믹 기판 위에 압축하여 제작
- ✓ 빛이 강할 수록 저항 값이 감소
- ✓ ADC를 이용하여 변화된 저항에 전압을 인가하여 전압의 변화를 감지
- ✓ 자동 조명장치, 조도 측정 등에 사용

럭스

다른 뜻에 대해서는 [Lux](#) 문서를 참조하십시오.

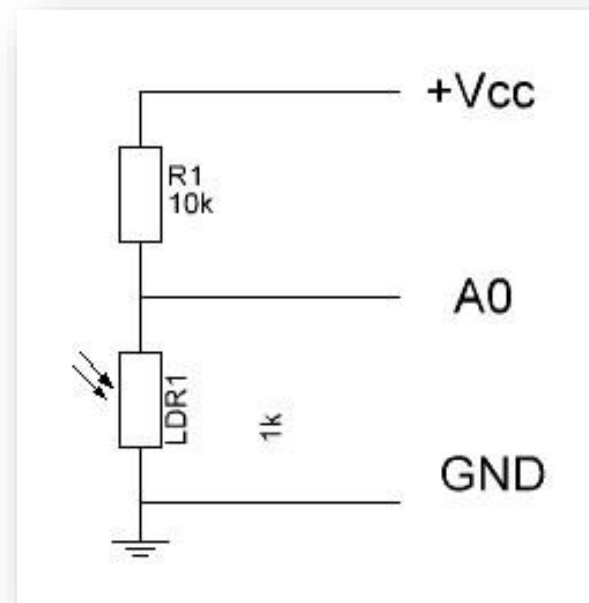
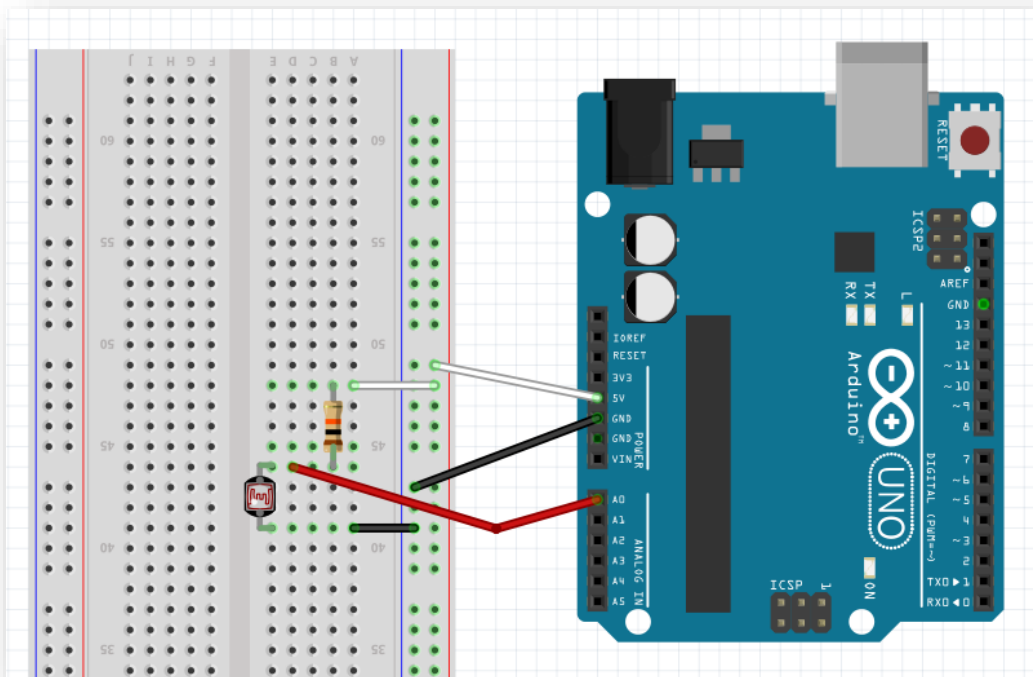
럭스(lux, 기호 **lx**)는 빛의 **조명도**를 나타내는 SI 단위이다. 럭스는 루멘에서 유도

$$1 \text{ lx} = 1 \text{ lm/m}^2 = 1 \text{ cd}\cdot\text{sr}\cdot\text{m}^{-2}$$

럭스의 예 [\[편집\]](#)

I 밝기차	예
10 ⁻⁵ lux	가장 밝은 별(시리우스)의 빛 ^[1]
10 ⁻⁴ lux	하늘을 덮은 완전한 별빛 ^[1]
0.002 lux	대기광이 있는 달 없는 맑은 밤 하늘 ^[1]
0.01 lux	초승달
0.27 lux	맑은 밤의 보름달 ^{[1][2]}
1 lux	열대 위도를 덮은 보름달 ^[3]
3.4 lux	맑은 하늘 아래의 어두운 황혼 ^[4]
50 lux	거실 ^[5]
80 lux	복도/화장실 ^[6]
100 lux	매우 어두운 낮 ^[1]
320 lux	권장 오피스 조명 (오스트레일리아) ^[7]
400 lux	맑은 날의 해뜰이 뜨는 해넘이
1000 lux	인공 조명 ^[1] ; 일반적인 TV 스튜디오 조명
10,000–25,000 lux	낮 (직사광선이 없을 때) ^[1]
32,000–130,000 lux	직사광선

CdS 센서 회로



Parts : 20 mm photocell LDR, R (10 kΩ X 1)

광센서에서의 전압 강하 값을 **A0**로 측정



빛 입력 (1/3)

실습목표

CdS 셀을 이용하여 조도를 측정해 보자.

1. CdS 셀로 측정된 조도를 아날로그 핀을 통하여 0~1023 범위로 읽는다.
2. ADC 값을 시리얼모니터로 0~100%의 범위로 출력한다. (빛의 밝기가 아니고 단지 밝기 비율)
(0 ~ 1023) → (100 ~ 0)%
3. ADC 값을 시리얼모니터로 lux 값으로 출력한다.

Hardware

1. CdS셀과 10k Ω 저항을 연결한 뒤 저항의 한쪽 끝은 5V에 CdS셀의 한쪽 끝은 GND에 연결한다.
2. 저항과 CdS셀 사이를 아날로그입력핀 A0에 연결한다.

A3.2.4 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 1.

```

AAnn_CdS
1 #define CDS_INPUT 0
2
3 void setup() {
4   Serial.begin(9600);
5 }
6
7 void loop() {
8
9   int value = analogRead(CDS_INPUT);
10  Serial.println(value);
11
12  delay(1000);
13 }
14

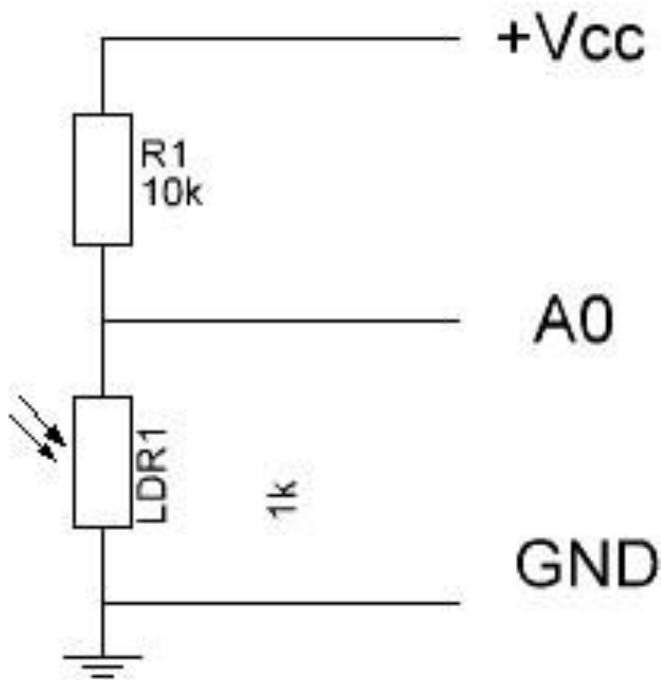
```

COM4

233
234
235
237
235
235
236
241
386
975
965
964
964
967

어두우면 측정 값이 커지고 밝을수록 값이 작아진다 ???

CdS 센서 회로 분석 (1/2)



LDR's (Light dependent resistors) have a low resistance in bright light and a high resistance in the darkness.

If you would use the LDR as the lower part of a voltage divider, then in darkness there would be a high voltage over the LDR, while in bright light, there would be a low voltage over that resistor.

어두우면 측정 값이 작아지고 밝을수록 값이 커져야 된다.
그리고 측정 값은 **lux**로 표현된다.

$$V_{out} = \frac{R_{ldr}}{R_1 + R_{ldr}} * V_{cc}$$

A0에서 측정되는 LDR
양단의 전압 = V_{out}

CdS 센서 회로 분석 (2/2)

Doing that on an Arduino Analog port, would give a reading between 0 and 1024, which of course are really non-descriptive numbers. What you would want is an output in Lux or Lumen?

$$R_{ldr} = 500 / \text{Lux}, \text{ or}$$

$$\text{Lux} = 500 / R_{ldr} \text{ (in kOhm)}$$

as R_{ldr} is related to the voltage measured over it, reading the Voltage over it, can be used to calculate the R_{ldr} and thus the Lux level

If the LDR is the lower part of a 5 Volt Voltage divider and a 10kOhm resistor the upper part, the Voltage will be:

$$V_{out} = (5 / (10 + R_{ldr})) * R_{ldr}$$

$$V_{out} = 5 * R_{ldr} / (10 + R_{ldr}) \text{ (remember: multiplication before division)}$$

as we do not measure a voltage, but a value between 0 and 1024, every step can be defined by $5/1024 = 0.0048828125$.

$$\Rightarrow V_{out} = \text{Analogreading} * 0.0048828125$$

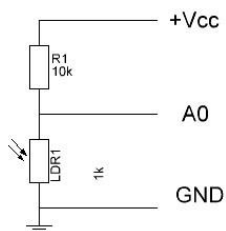
as $R_{ldr} = (10 * V_{out}) / (5 - V_{out})$ (remember R_{ldr} is expressed in kOhm)

$$\Rightarrow \text{Lux} = (500 * (5 - V_{out})) / (10 * V_{out})$$

$$\Rightarrow \text{Lux} = (2500 - 500 * V_{out}) / (10 * V_{out})$$

$$\Rightarrow \text{Lux} = (2500 / V_{out} - 500) / 10$$

$$\Rightarrow \text{Lux} = (2500 / ((\text{AnalogRead} * 0.0048828125) - 500)) / 10$$



$$V_{out} = \frac{R_{ldr}}{R_1 + R_{ldr}} * V_{cc}$$

<https://arduino diy.wordpress.com/2013/11/03/measuring-light-with-an-arduino/>

CdS 센서 회로 - 측정 2.

AA00_CdS

```
1 // lux
2 #define CDS_INPUT 0
3
4 void setup() {
5   Serial.begin(9600);
6 }
7 void loop() {
8   int value = analogRead(CDS_INPUT);
9   Serial.println(int(luminosity(value)));
10  delay(1000);
11 }
12
13 //Voltage to LuxLux
14 double luminosity (int RawADC0){
15   double Vout=RawADC0*0.0048828125; // 5/1024 (Vin = 5 V)
16   int lux=(2500/Vout-500)/10; // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
17   return lux;
18 }
```

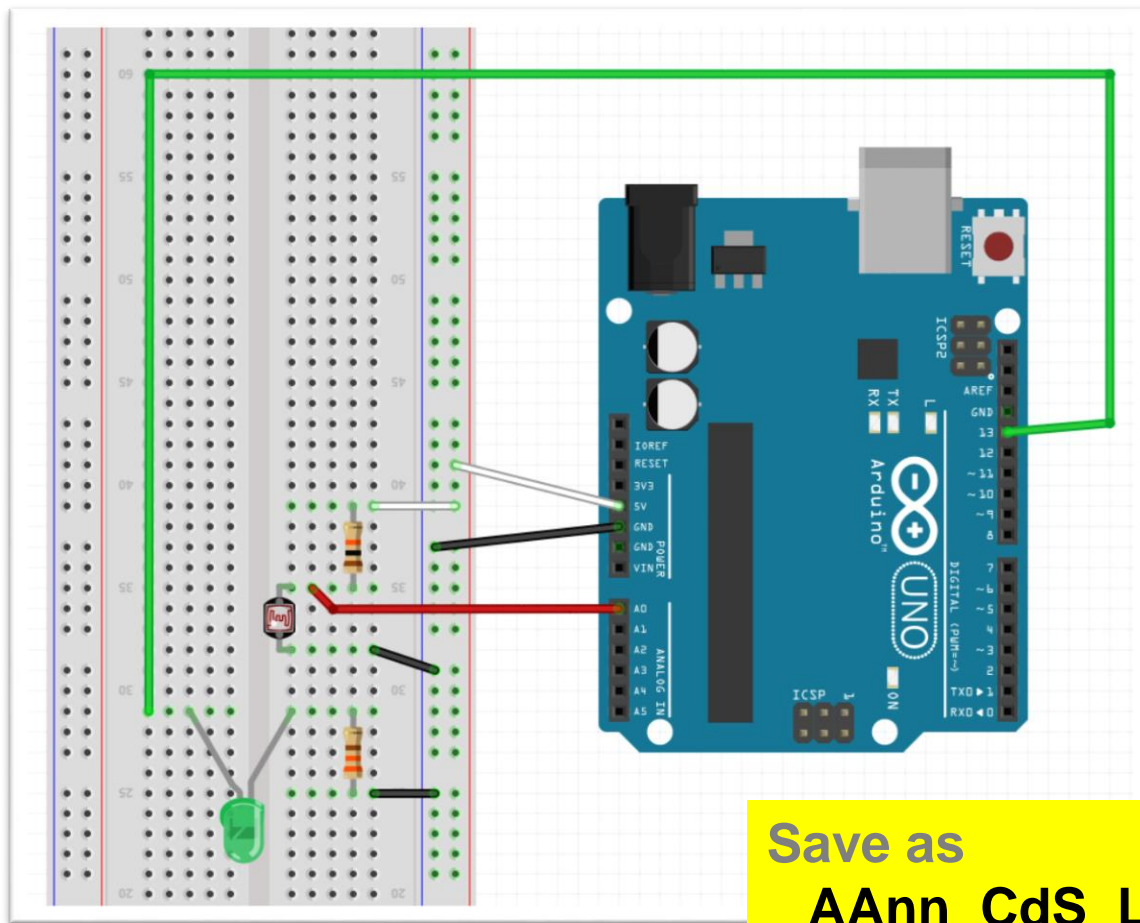
COM4



밝을수록 측정 값이 커지고 어두울수록 값이 작아진다 !!!

DIY 조도 값에 따라 LED를 켜고 끄는 코드를 만드시오.

- 단색 LED의 anode를 D13번, cathode를 330 Ω 저항에 연결 후 **GND**에 연결하시오.
- 조도 값이 문턱 값 이상이면 LED를 OFF, 그렇지 않으면 ON.



Save as
AAnn_CdS_LED.fzz



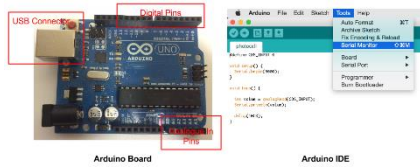
A3.2.7 Luminosity sensor [Photocell CdS LDR]

DIY Code

Write down your code here to complete the task that turns on LED when luminosity of ambient light becomes lower than a threshold.

조도 값이 문턱 값 이상이면 LED를 OFF, 그렇지 않으면 ON.

Save as
AAnn_CdS_LED.ino



IOT: HSC

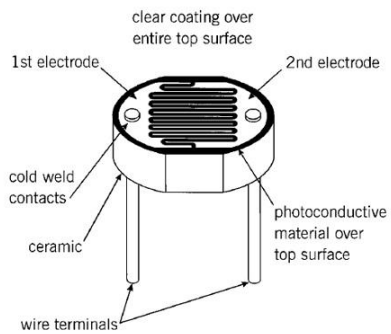
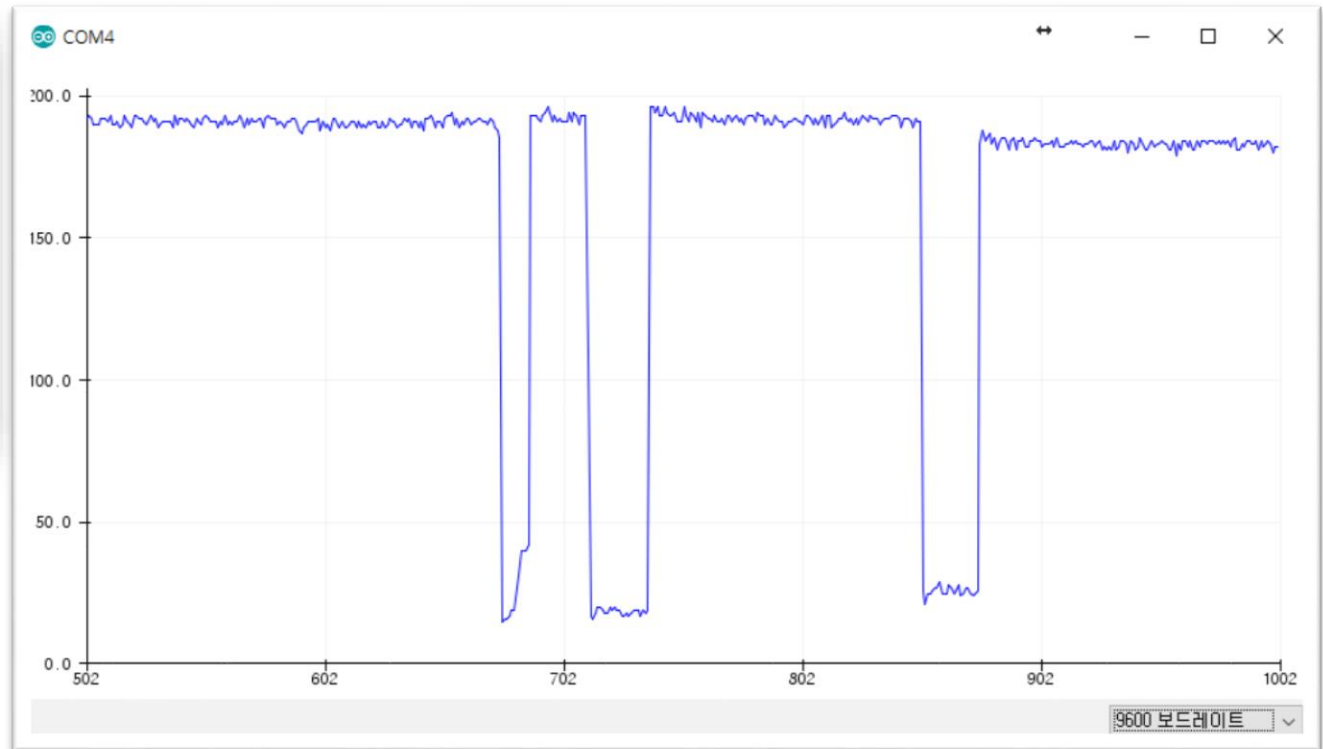
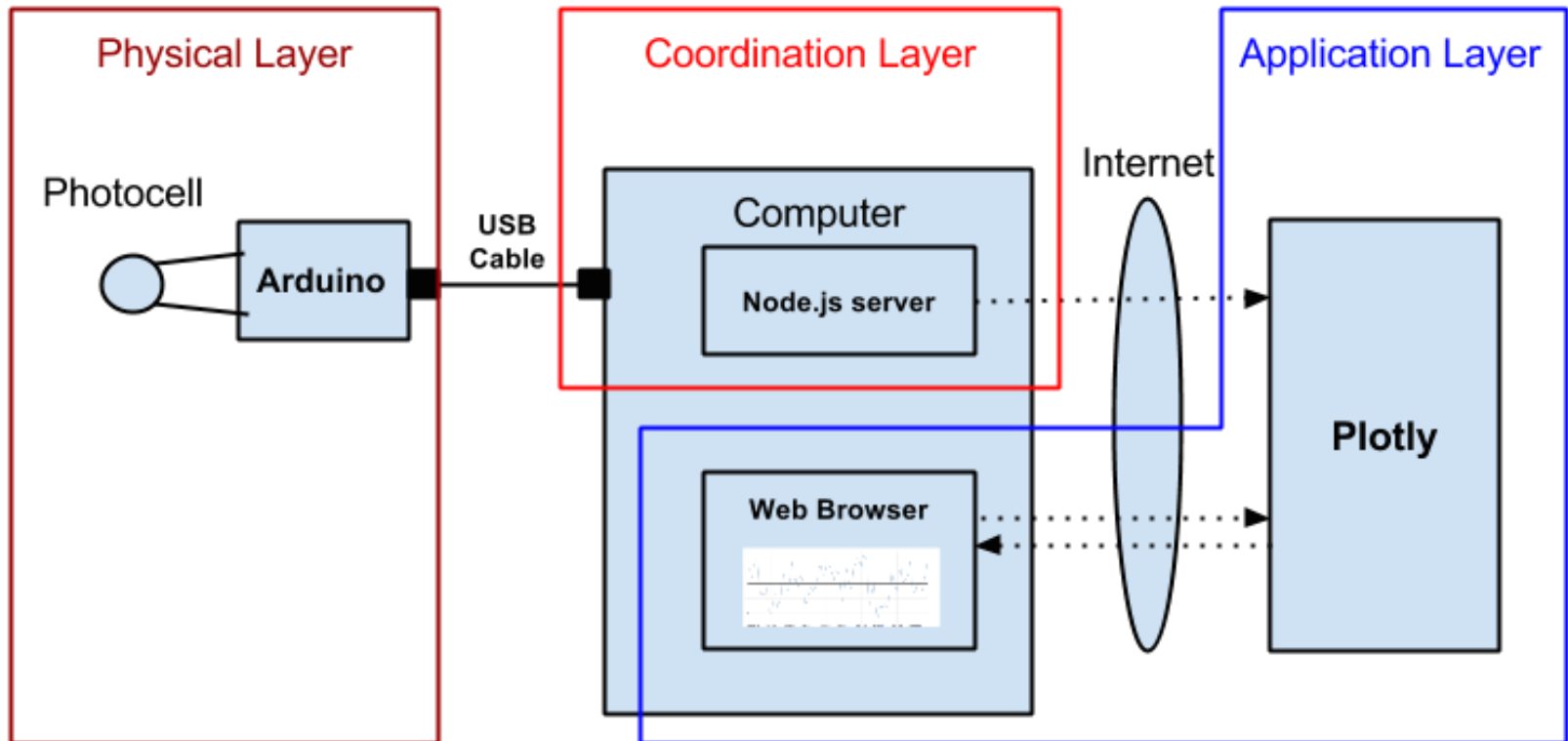


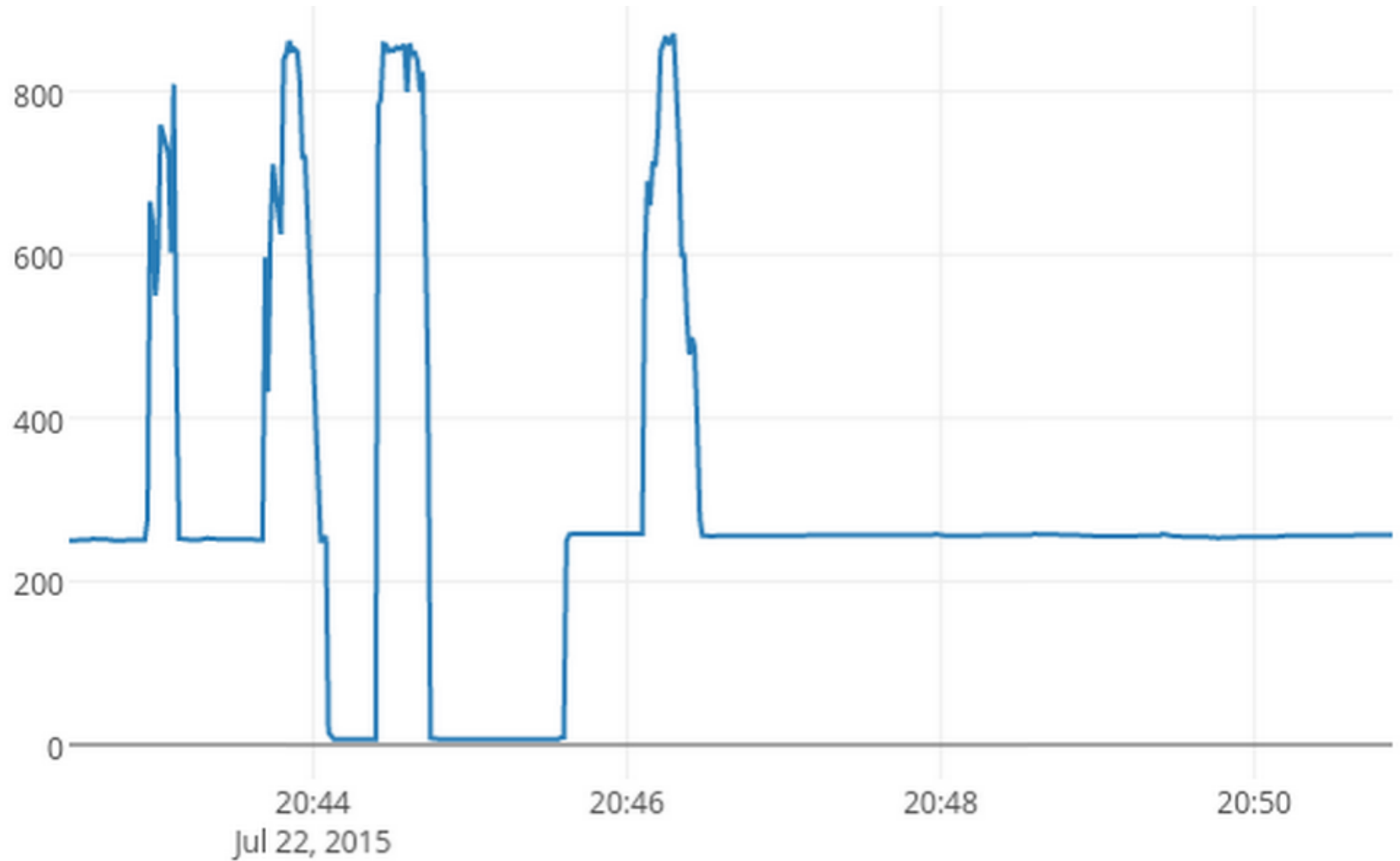
Figure 3
Typical Construction of a Plastic Coated Photocell



Layout [H S C]



Arduino: node.js + plotly





[Practice]

◆ [wk06]

- **Arduino sensors**
- **Complete your project**
- **Upload file name : AAnn_Rpt05.zip**

◆ [Target of this week]

- Complete your projects
- Save your outcomes and compress 3 figures

제출파일명 : **AAnn_Rpt05.zip**

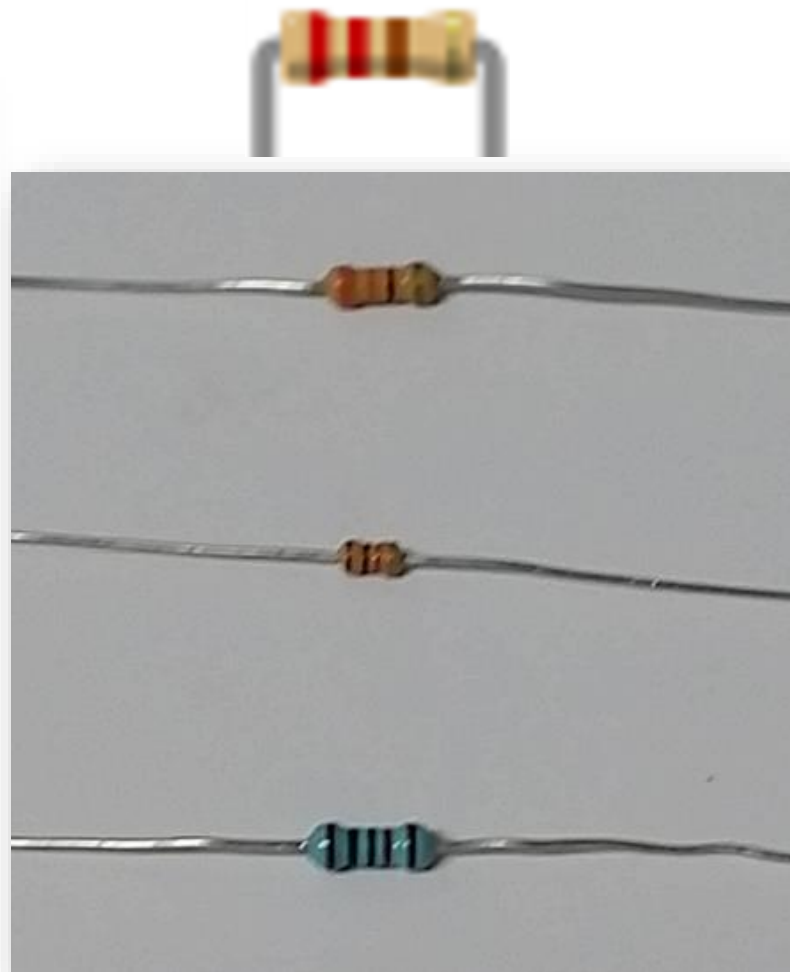
- 압축할 파일들

- ① **AAnn_TMP36.png**
- ② **AAnn_CdS_LED.fzz**
- ③ **AAnn_CdS_LED.ino**

[참고 : 저항 값 읽기]



Color	First	Second	Third	Multiplier	Tolerance
Black	0	0	0	x1	
Brown	1	1	1	x10	1%
Red	2	2	2	x100	2%
Orange	3	3	3	x1000	
Yellow	4	4	4	x10 000	
Green	5	5	5	x100 000	0,50%
Blue	6	6	6	x1 000 000	0,25%
Violette	7	7	7	x10 000 000	0,10%
Gray	8	8	8		
White	9	9	9		
Silver				x0,01	10%
Gold				x0,1	5%



● References & good sites

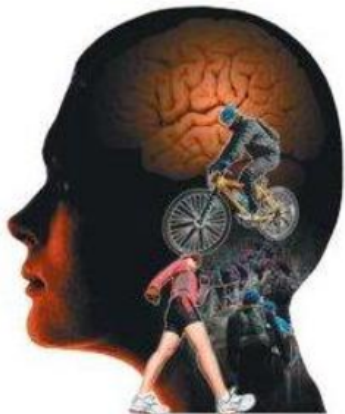
- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling



[Features](#) [Business](#) [Explore](#) [Marketplace](#) [Pricing](#)

Search GitHub

[Sign in](#) or [Sign up](#)



Redwoods Yi

Redwoods

Block or report user

📍 GimHae, Republic of Korea

Overview

Repositories 5

Stars 2

Followers 0

Following 0

Pinned repositories

dht22-iot-project

lot project to monitor data streaming from DHT22 wired at Arduino.

● HTML

Lec

All lectures by Redwoods in Inje University

arduino-nodejs-plotly-streaming

This repo introduces a simple and efficient way to plot the streaming data from Arduino with Easy Pulse ppg sensor or DHT11 sensor.

● HTML

hw-coding

Resource for lecture of Hardware Programming (2017, Inje university)

● Arduino

