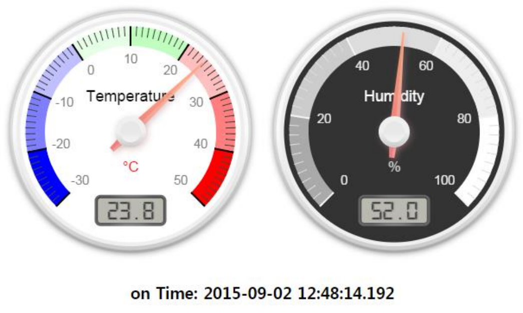
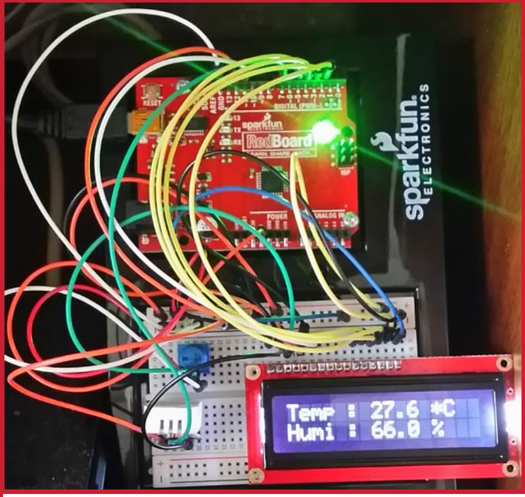
HW-SW-Connectivity

[wk06]

Arduino Sensors



on Time: 2015-09-02 12:48:14.192

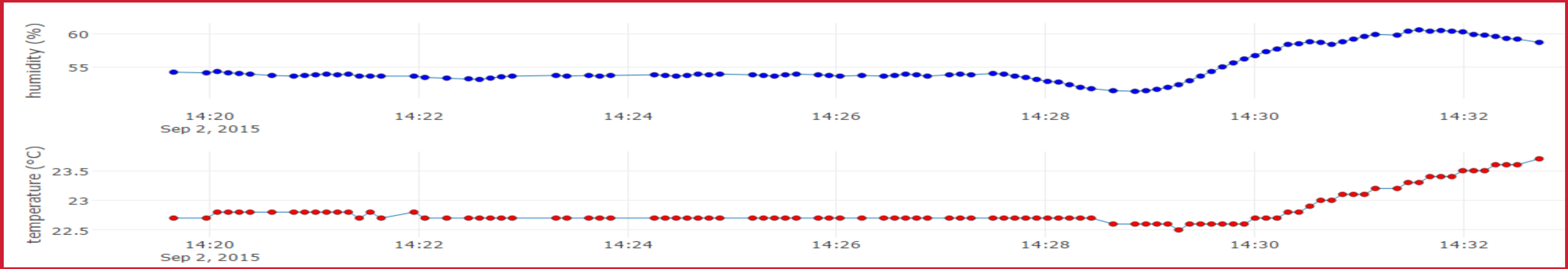


Basic HW and SW Integration using
Arduino & Javascript

COMSI, INJE University

2nd semester, 2017

Email : yish@inje.ac.kr



wk06 : Mid-term Examination

실기시험

1. 일시**: 10**월 **18**일**(**수**)** 오후 **4**시 **~ 5**시

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 SW**를 설치해서 준비**.**

[PC 사용법**] local hard**로 로긴

wk06 : Mid-term Examination

필기시험

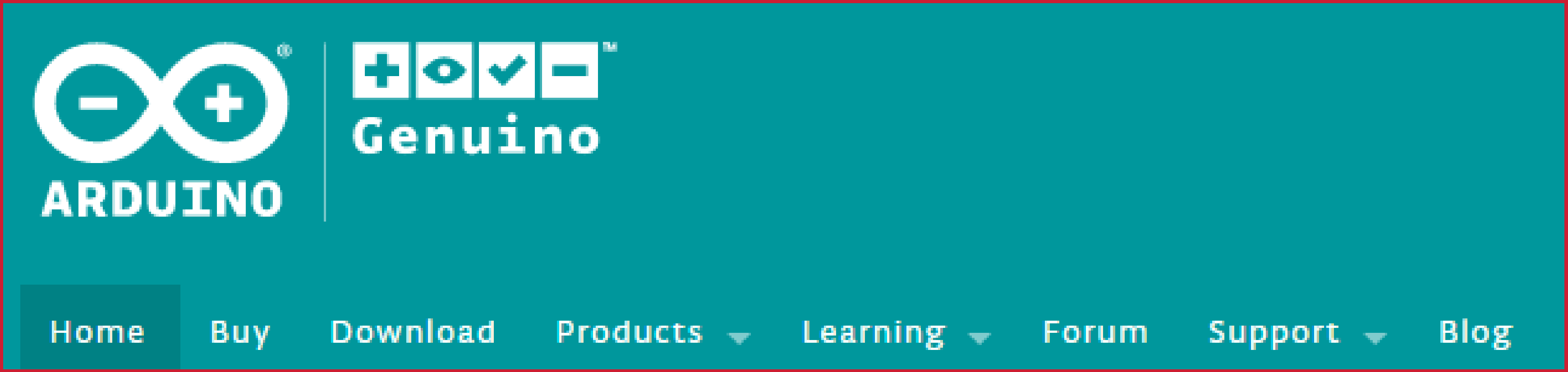
1. 일시: 10월 18일(수) 오후 5시 ~ 6시

2. 장소: E429

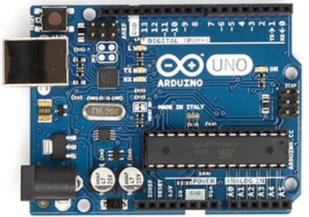
3. 20문제 (객관식 및 단답형)

4. 범위 : node.js ~ arduino

Arduino



https://www.arduino.cc/



[Practice]

- [wk05]
  - Arduino basic circuits
  - Complete your project
  - Upload file name : AAnn_Rpt04.zip

wk05 : Practice-04 : 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**

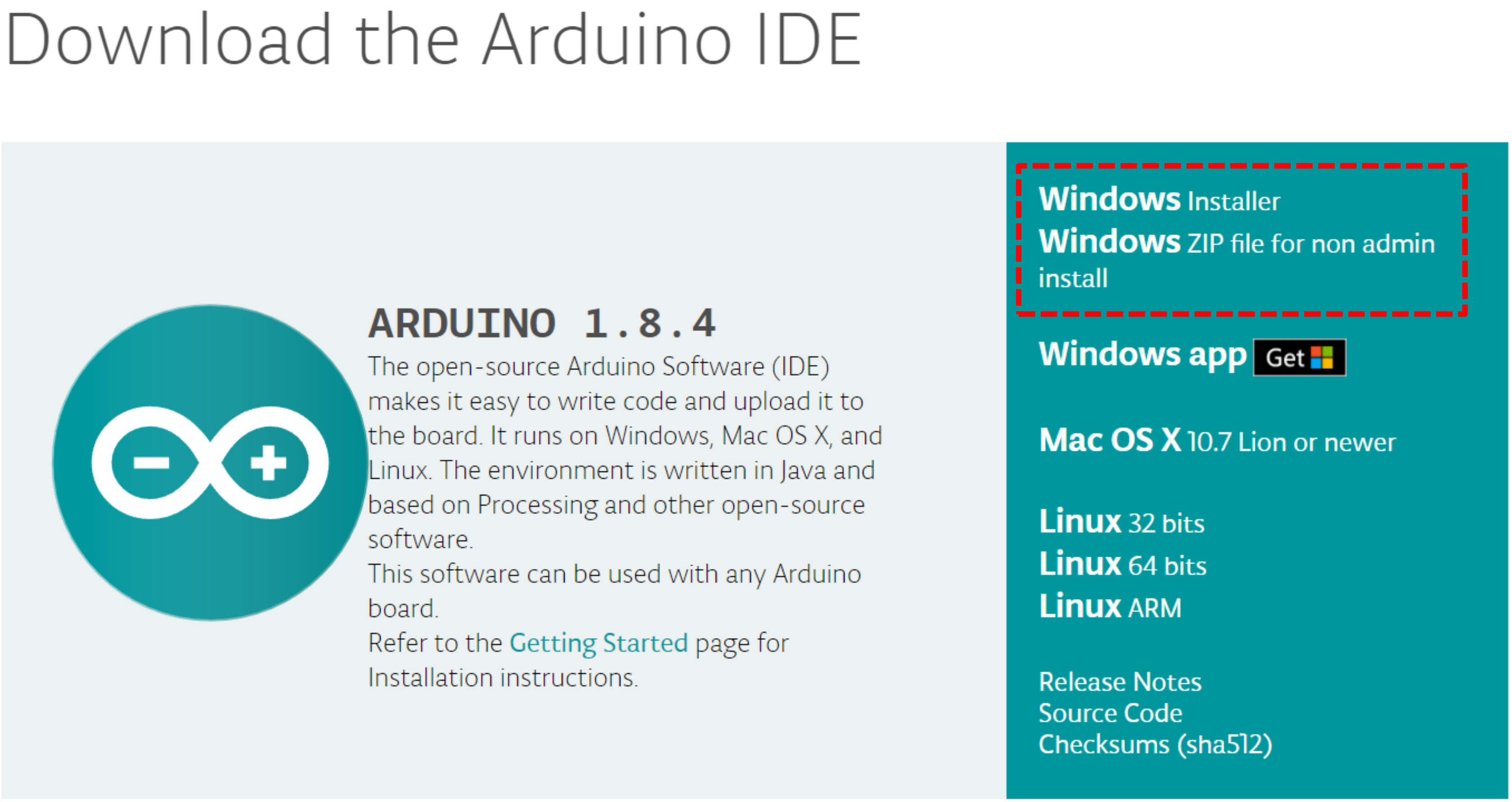
⑤ **AAnn_AnalogVoltage.png**

⑥ **AAnn_AnalogVoltage_Plot.png**

1. Arduino SW: IDE

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 Get

Mac OS X 10.7 Lion or newer

Linux 32 bits

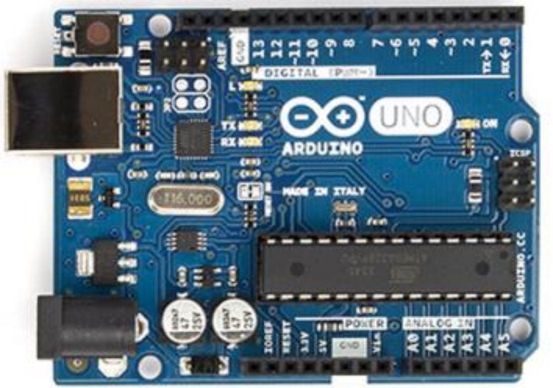
Linux 64 bits

Linux ARM

Release Notes

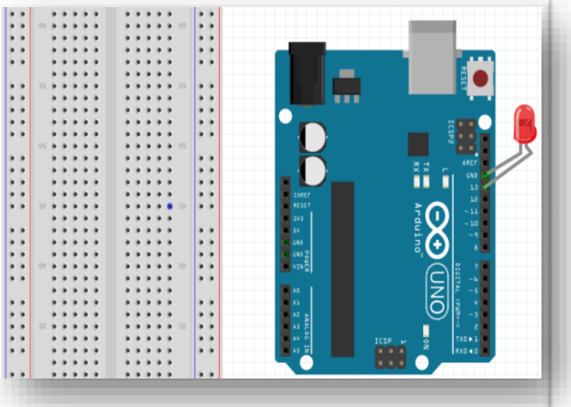
Source Code

Checksums (sha512)

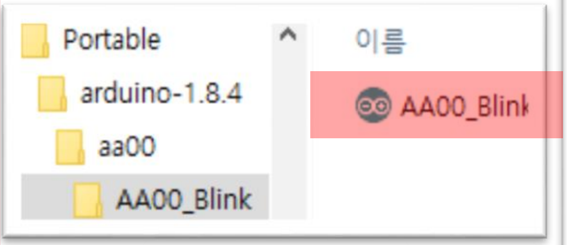


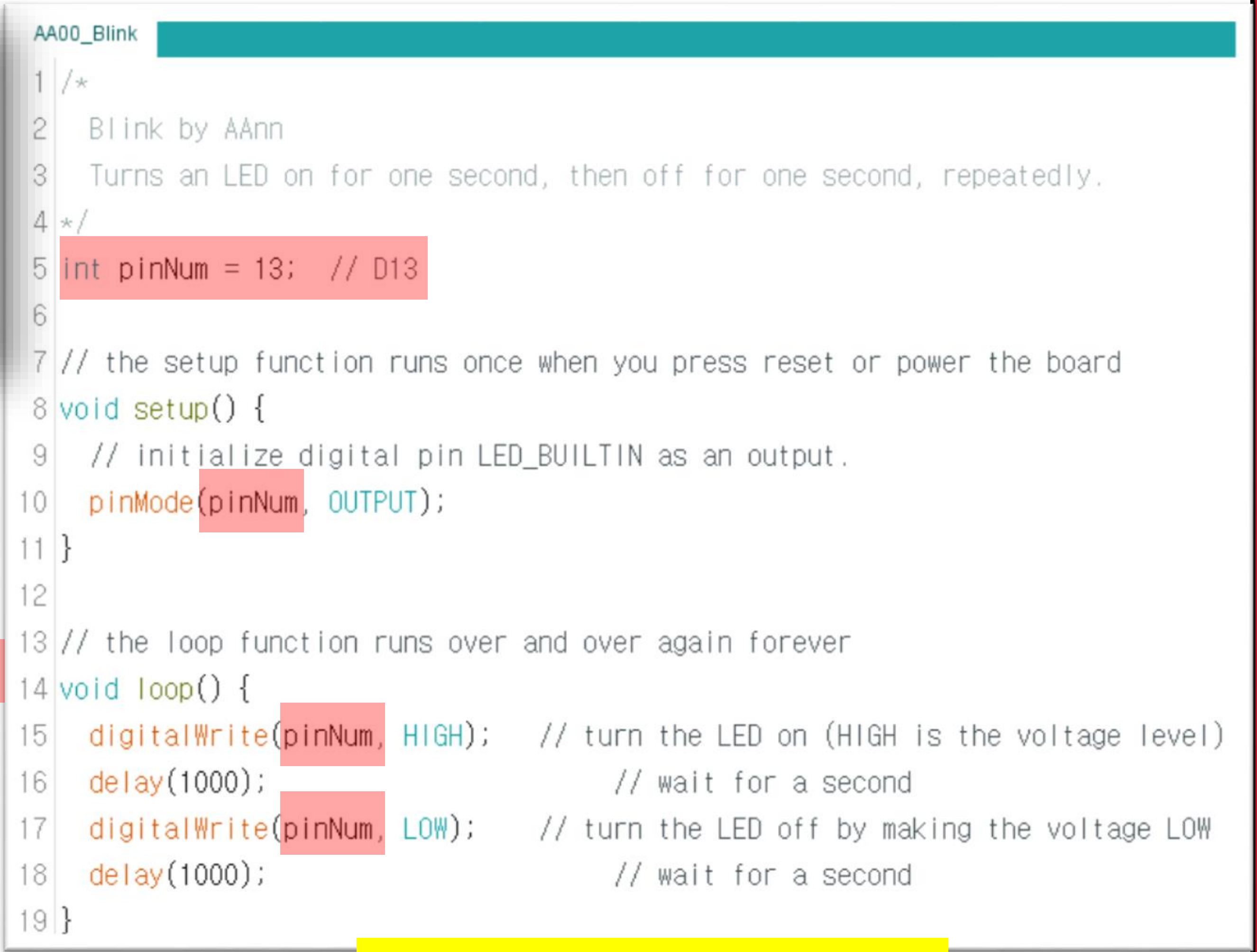
LED

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



Connect LED to D13 & GND





```
/*
  Blink by AAnn
  Turns an LED on for one second, then off for one second, repeatedly.
*/
int pinNum = 13;   // D13

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(pinNum, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(pinNum, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(pinNum, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Save as

AAnn_Blink.png

Serial monitor & plotter

DIY: Sawtooth wave

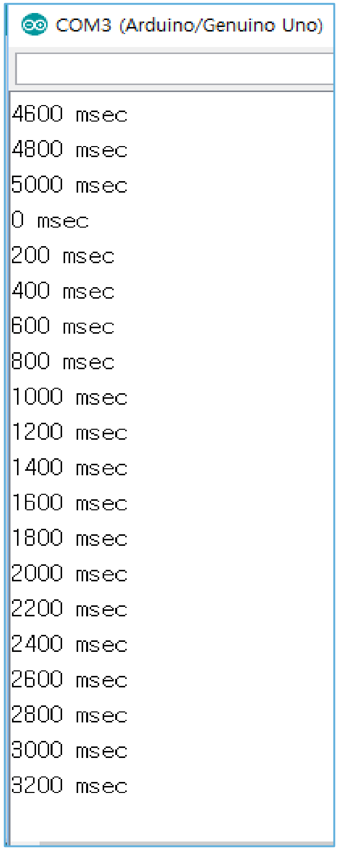
[DIY]

delay를 0.2초로 설정후

5초 마다 number를 초기화하여

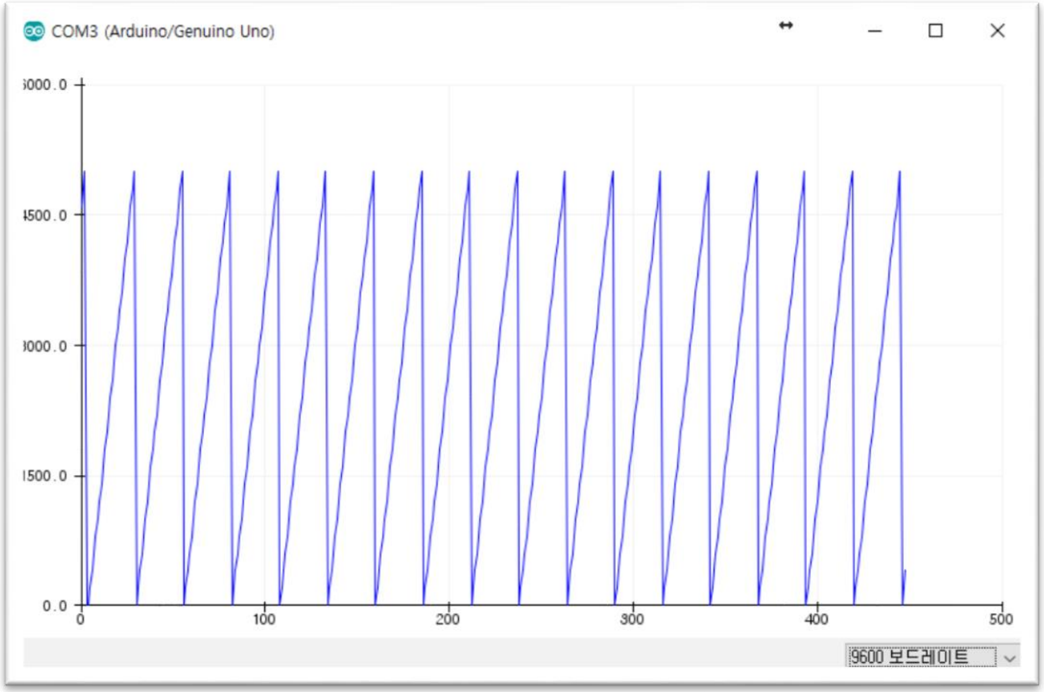
시리얼플로터로 톱니파를 발생.

시간은 ms로 계산해서 출력



COM3 (Arduino/Genuino Uno)

```
4600 msec
4800 msec
5000 msec
0 msec
200 msec
400 msec
600 msec
800 msec
1000 msec
1200 msec
1400 msec
1600 msec
1800 msec
2000 msec
2200 msec
2400 msec
2600 msec
2800 msec
3000 msec
3200 msec
```



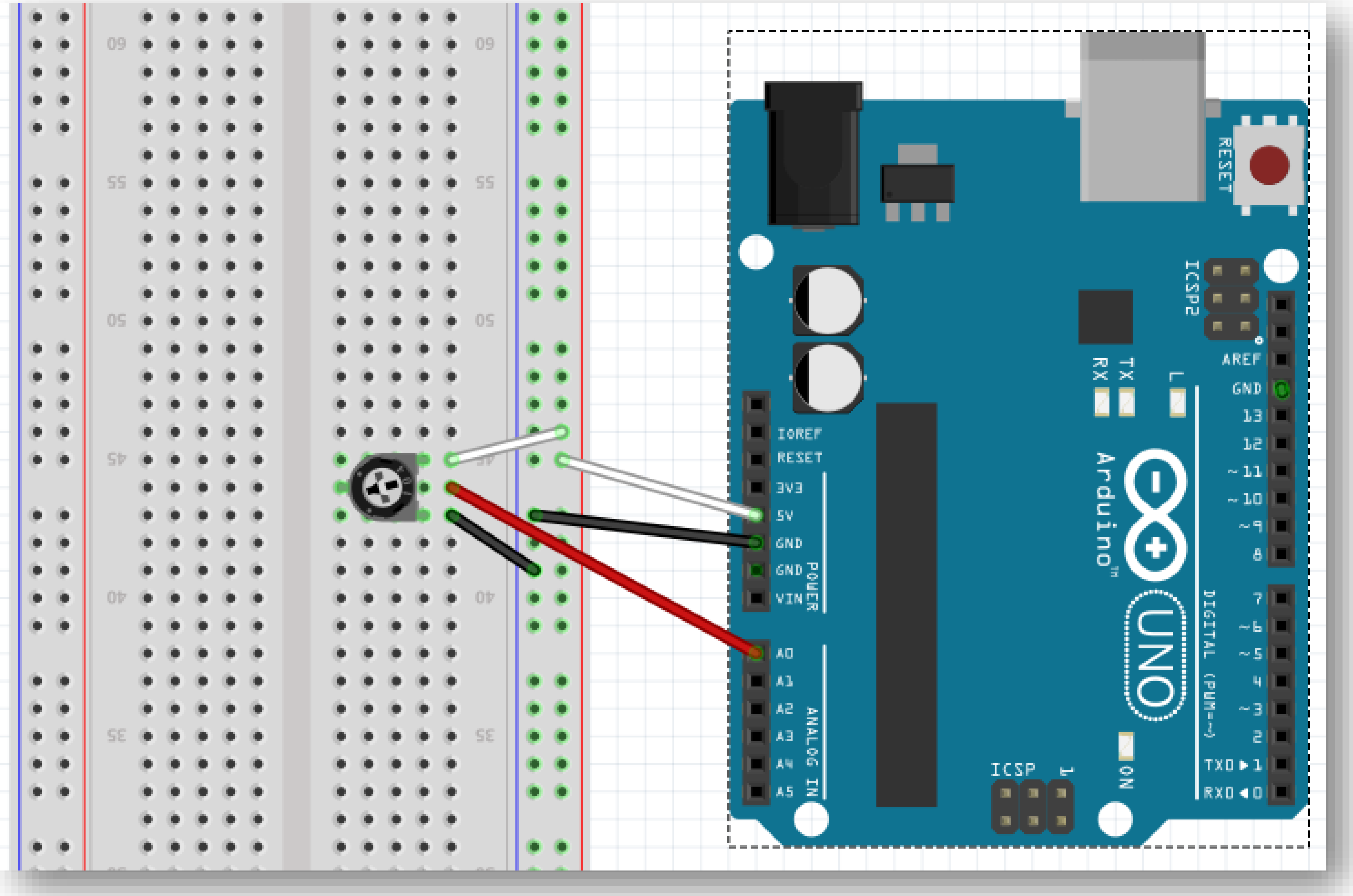
Save as

AAnn_Sawtooth.png

Analog Signal

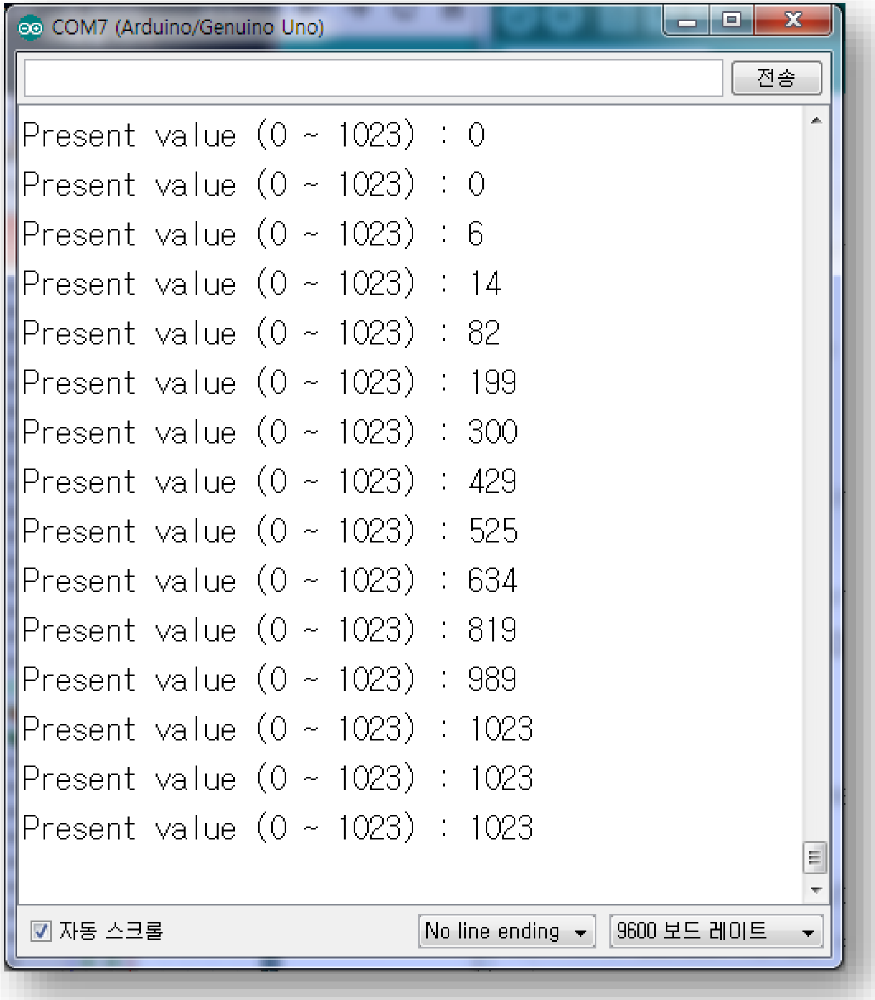
A2.5.1 AnalogReadSerial (circuit)

Standard potentiometer (가변 저항기)



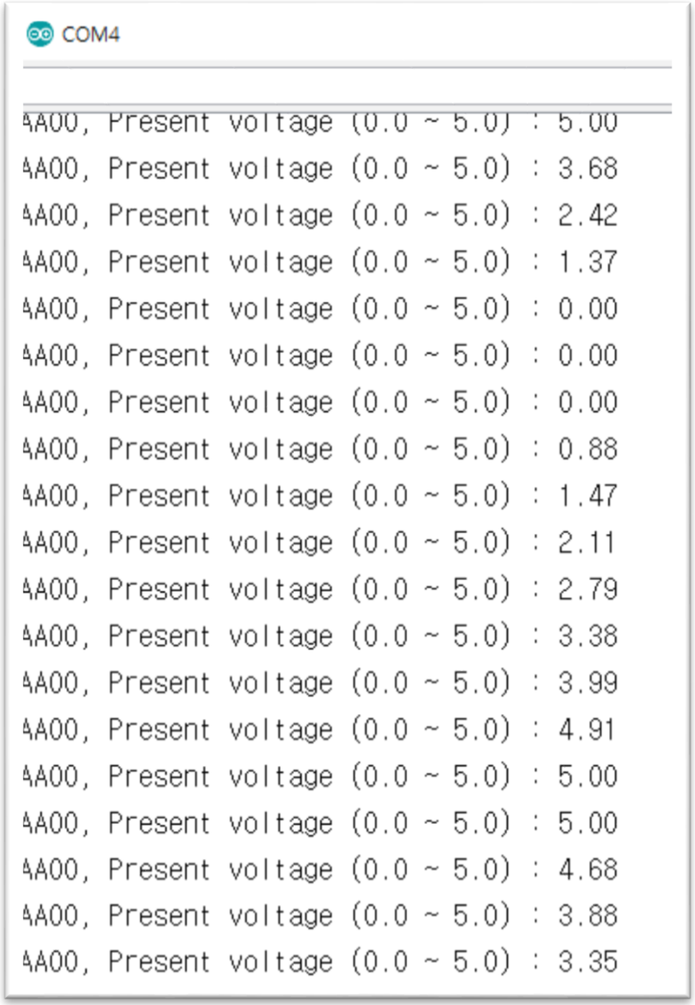
A2.5.4 ReadAnalogVoltage

Serial monitor



```
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
```

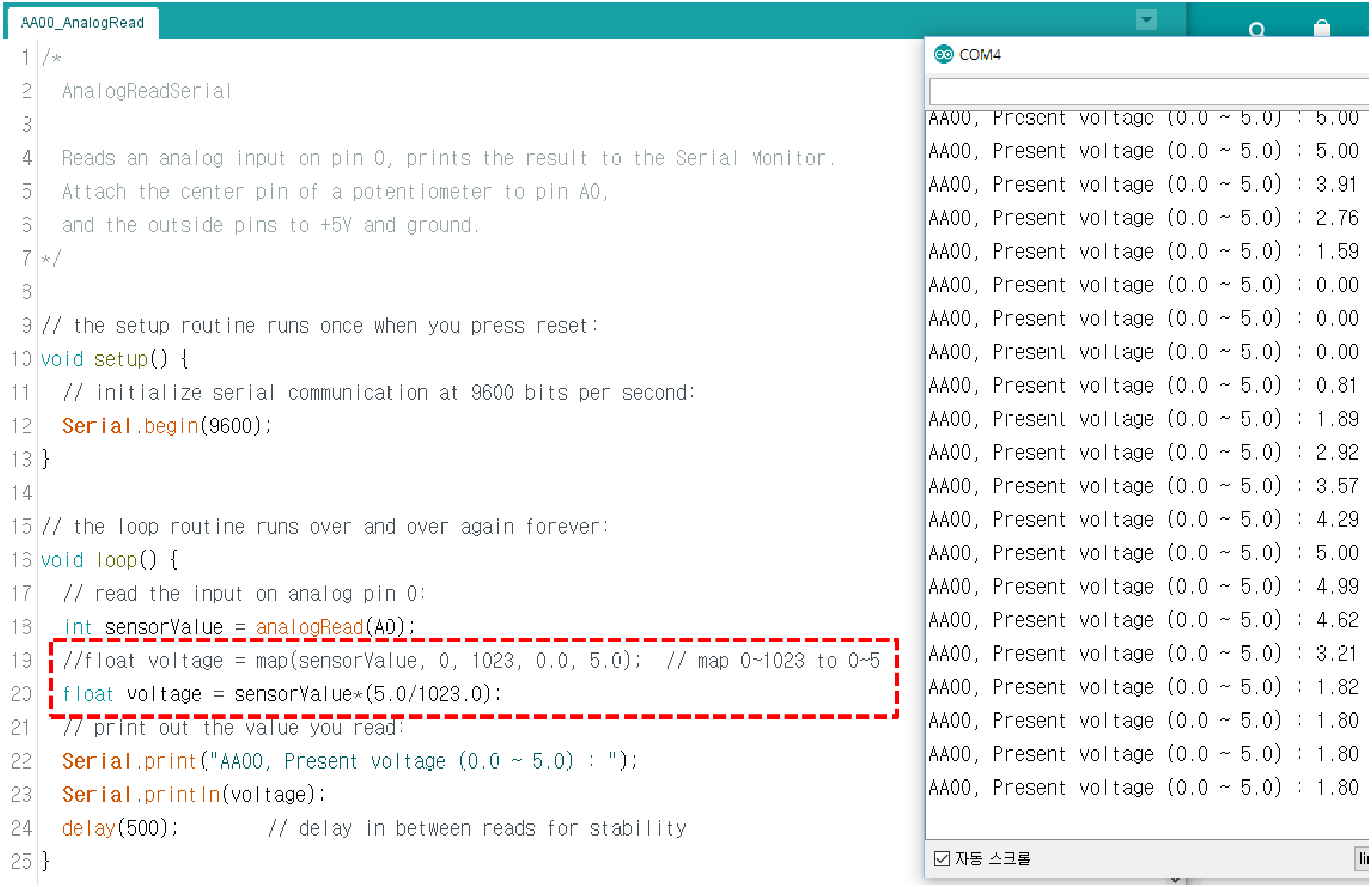
☑ 자동 스크롤 — No line ending — 9600 보드 레이트



```
AA00, Present voltage (0.0 ~ 5.0) : 5.00
AA00, Present voltage (0.0 ~ 5.0) : 3.68
AA00, Present voltage (0.0 ~ 5.0) : 2.42
AA00, Present voltage (0.0 ~ 5.0) : 1.37
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) : 0.88
AA00, Present voltage (0.0 ~ 5.0) : 1.47
AA00, Present voltage (0.0 ~ 5.0) : 2.11
AA00, Present voltage (0.0 ~ 5.0) : 2.79
AA00, Present voltage (0.0 ~ 5.0) : 3.38
AA00, Present voltage (0.0 ~ 5.0) : 3.99
AA00, Present voltage (0.0 ~ 5.0) : 4.91
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) : 4.68
AA00, Present voltage (0.0 ~ 5.0) : 3.88
AA00, Present voltage (0.0 ~ 5.0) : 3.35
```

A2.5.6 ReadAnalogVoltage

Hint code

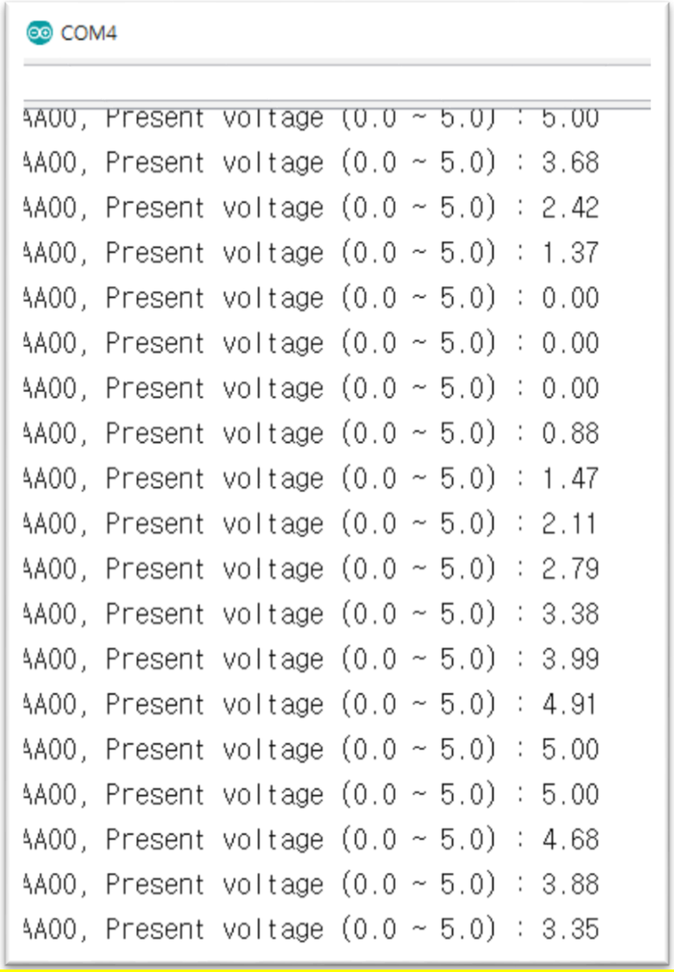


```
/*
  AnalogReadSerial

  Reads an analog input on pin 0, prints the result to the Serial Monitor.
  Attach the center pin of a potentiometer to pin A0,
  and the outside pins to +5V and ground.
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  //float voltage = map(sensorValue, 0, 1023, 0.0, 5.0);  // map 0~1023 to 0~5
  float voltage = sensorValue*(5.0/1023.0);
  // print out the value you read:
  Serial.print("AA00, Present voltage (0.0 ~ 5.0) : ");
  Serial.println(voltage);
  delay(500);        // delay in between reads for stability
}
```

```
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
```

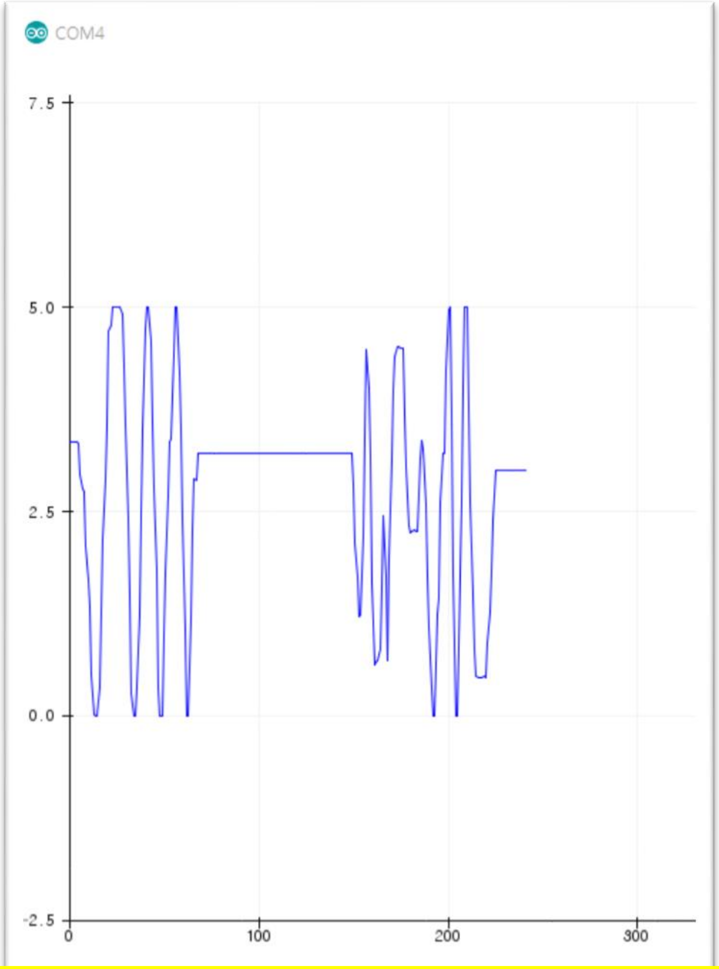
A2.5.7 ReadAnalogVoltage

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
```

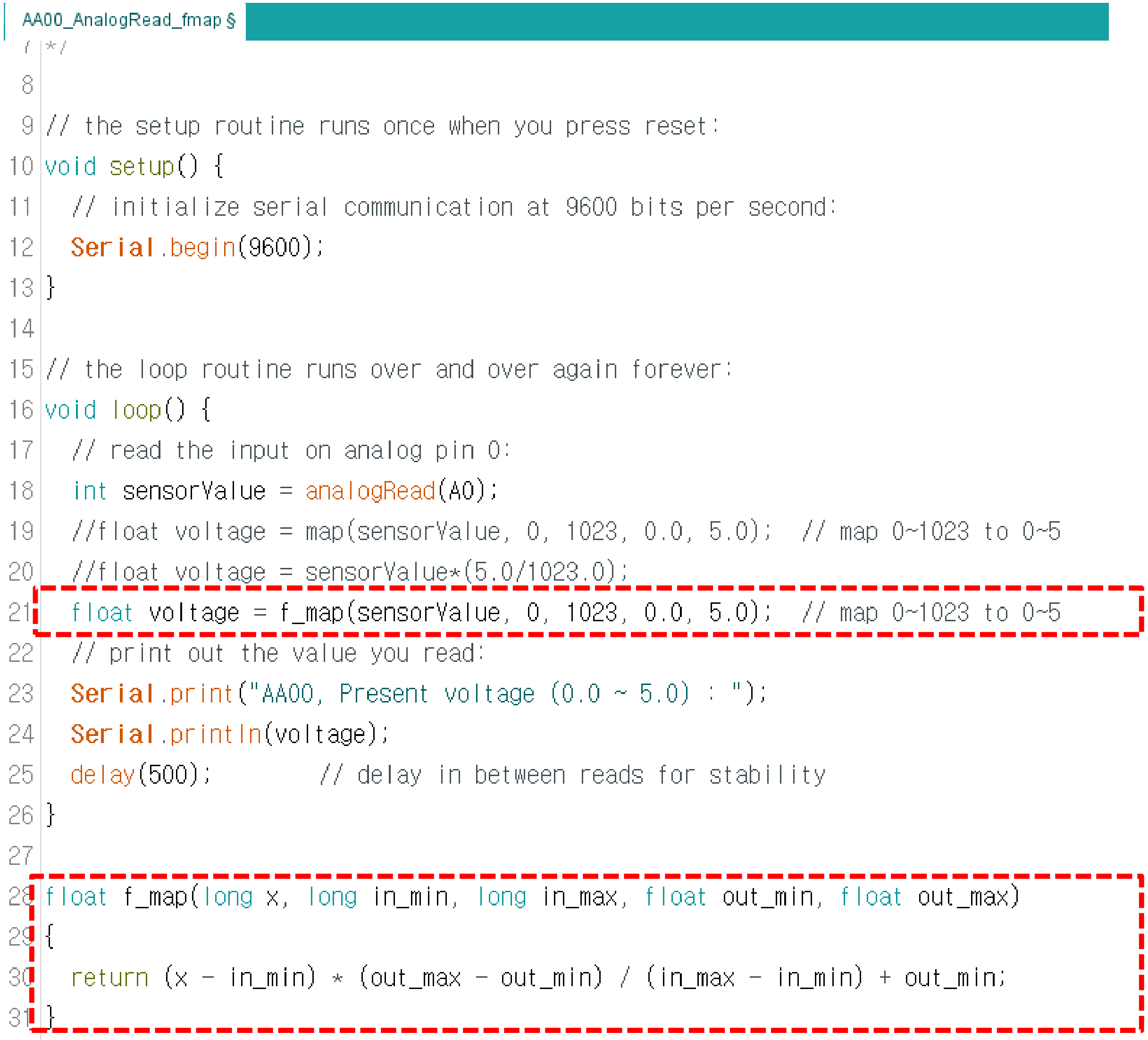


Save as **AAnn_AnalogVoltage.png**

Save as **AAnn_AnalogVoltage_Plot.png**

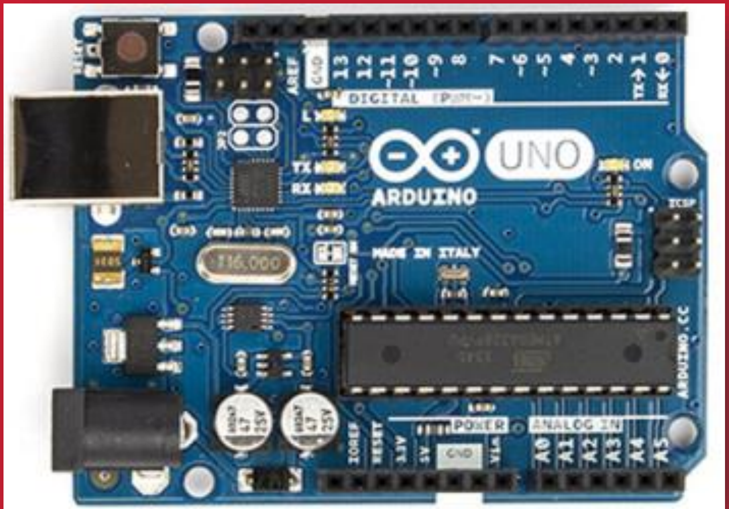
A2.5.6-1 ReadAnalogVoltage using f_map()

Hint code : f_map() instead of map()



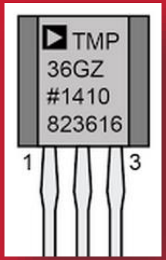
AA00_AnalogRead_fmap §

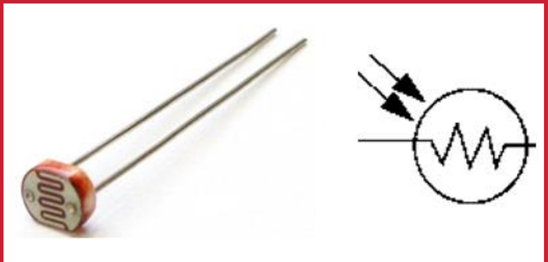
```
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  //float voltage = map(sensorValue, 0, 1023, 0.0, 5.0);  // map 0~1023 to 0~5
  //float voltage = sensorValue*(5.0/1023.0);
  float voltage = f_map(sensorValue, 0, 1023, 0.0, 5.0);  // map 0~1023 to 0~5
  // print out the value you read:
  Serial.print("AA00, Present voltage (0.0 ~ 5.0) : ");
  Serial.println(voltage);
  delay(500);        // delay in between reads for stability
}

float f_map(long x, long in_min, long in_max, float out_min, float out_max)
{
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
```

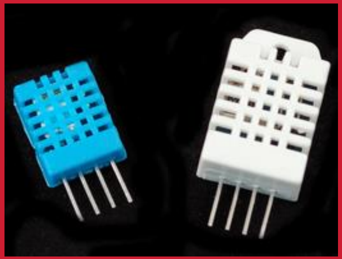


Arduino

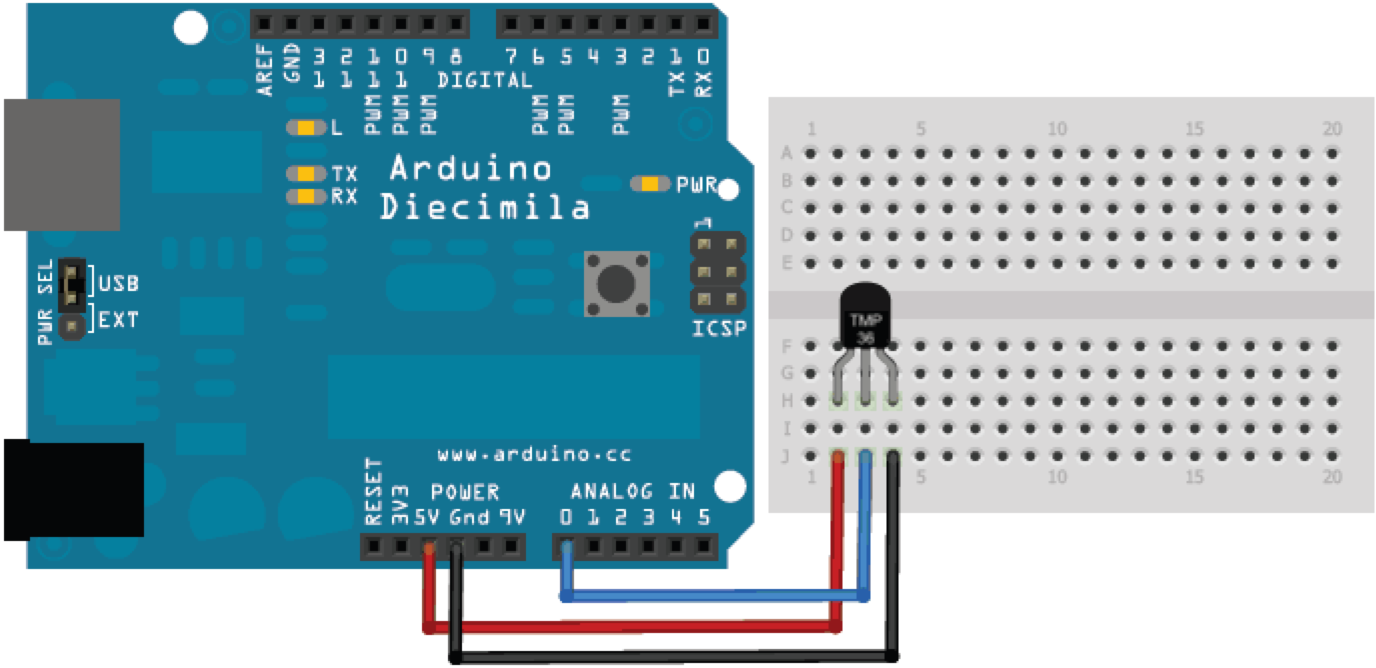
Sensors

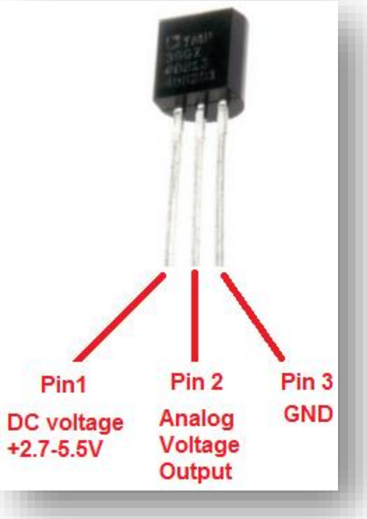




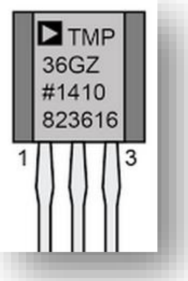


A3.1.1 Temperature sensor [TMP36]

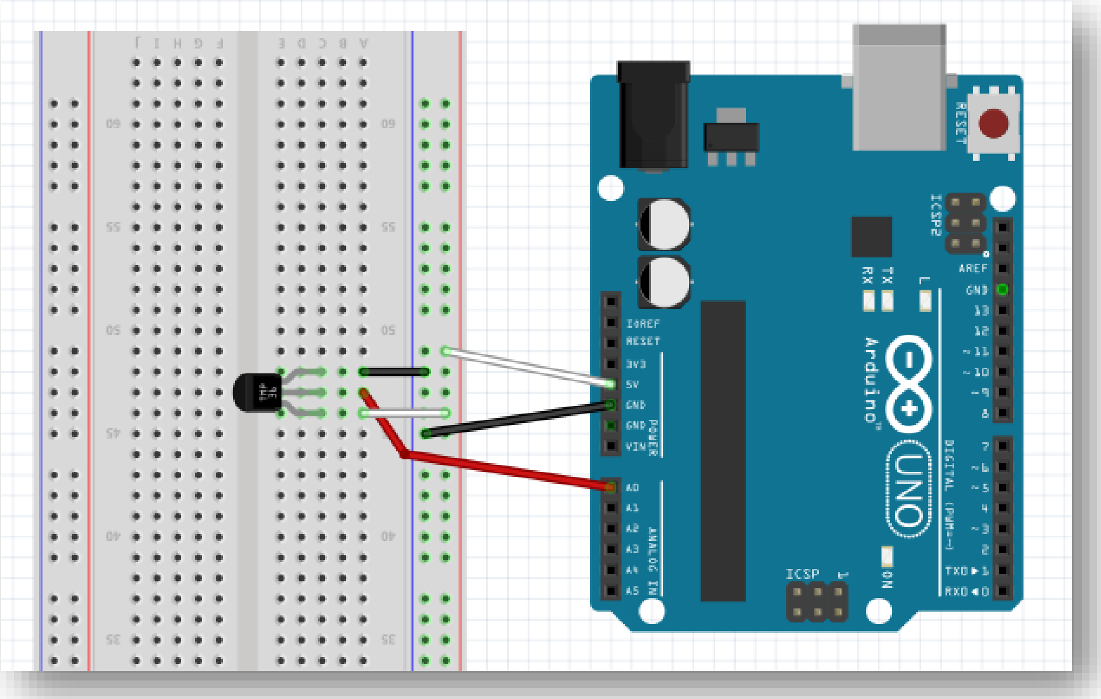


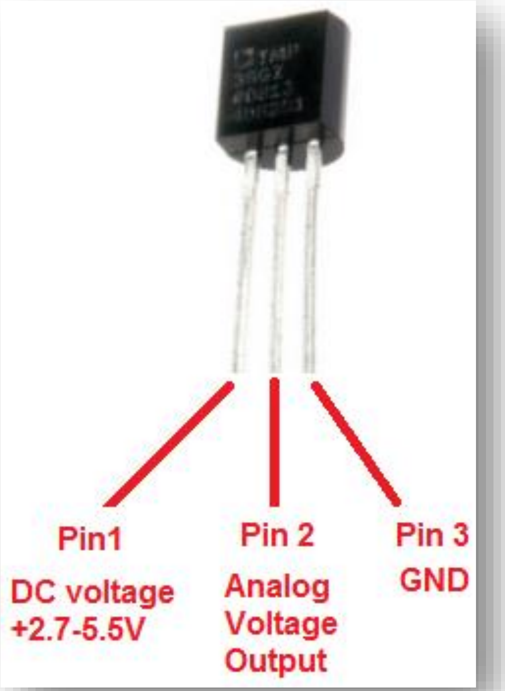


Parts : TMP36

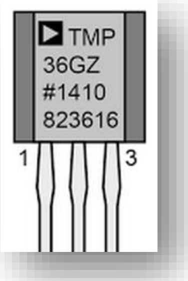


A3.1.2 Temperature sensor [TMP36]





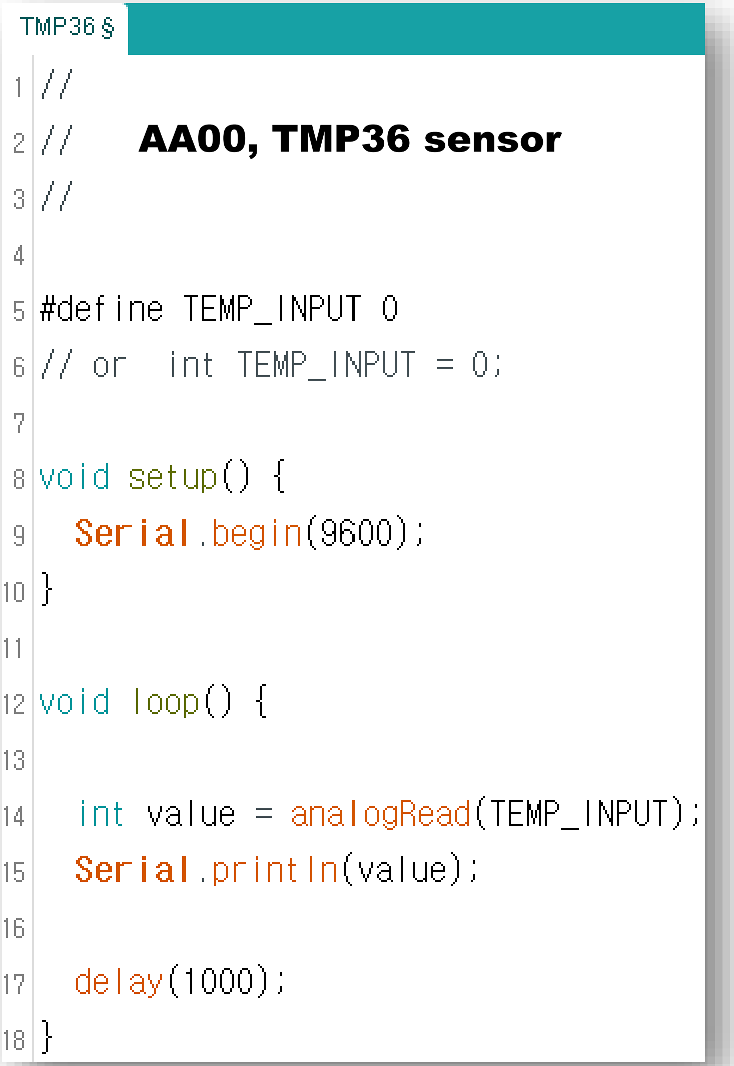
Parts : TMP36



- **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

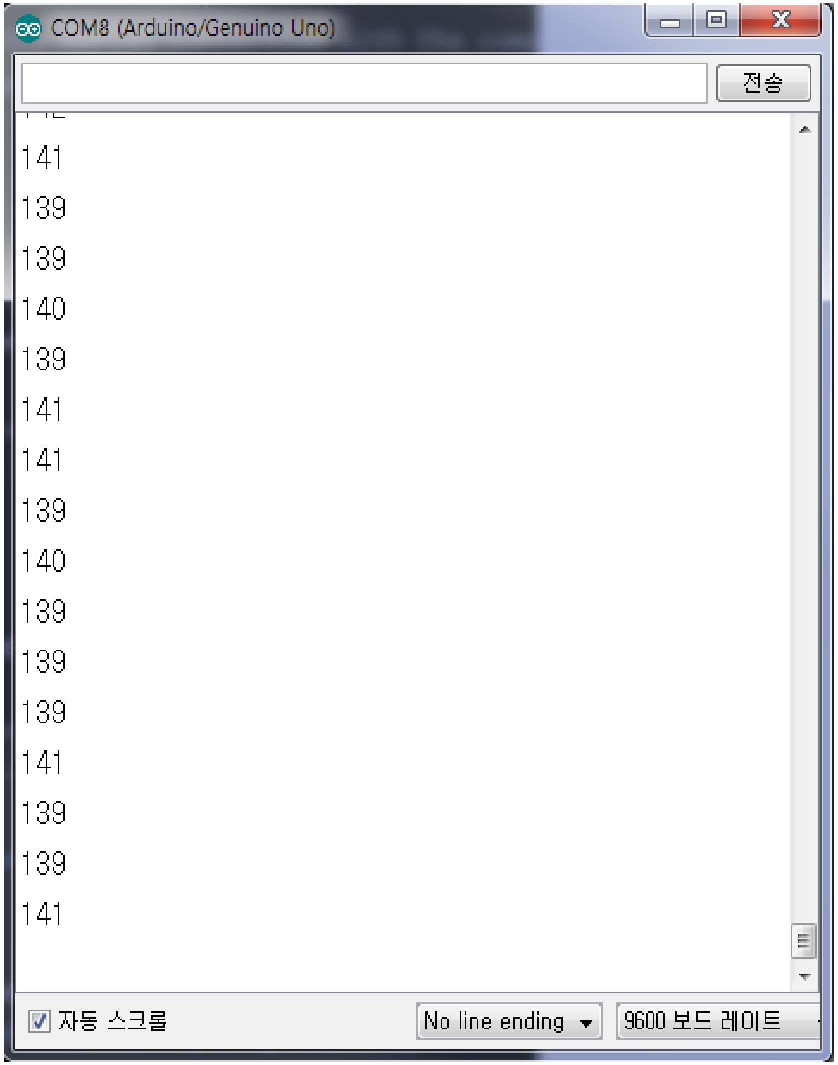
A3.1.3 Temperature sensor [TMP36]

Simple code



```
//
//     AA00, TMP36 sensor
//

#define TEMP_INPUT 0
// or  int TEMP_INPUT = 0;

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

void loop() {

  int value = analogRead(TEMP_INPUT);
  Serial.println(value);

  delay(1000);
}
```

Serial output (0 ~ 1023)



```
141
139
139
140
139
141
141
139
140
139
139
139
141
139
139
141
```

☑ 자동 스크롤   No line ending   9600 보드 레이트

A3.1.4 Temperature sensor [TMP36]

Sensor property

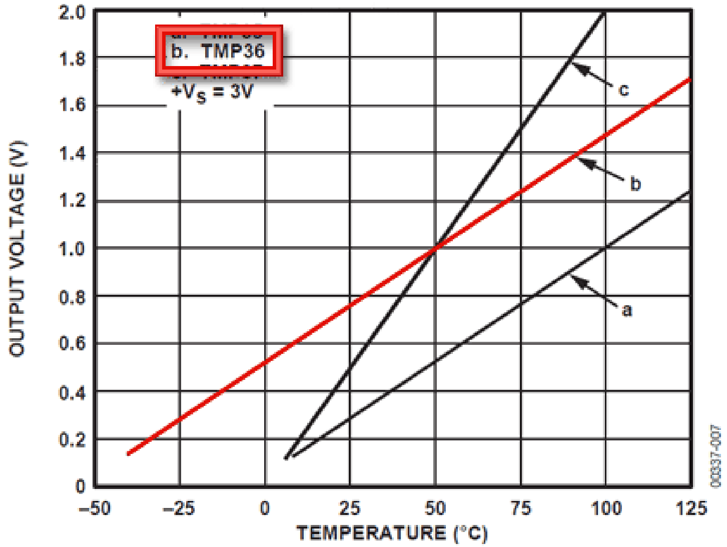


Figure 6. Output Voltage vs. Temperature

Temperature conversion

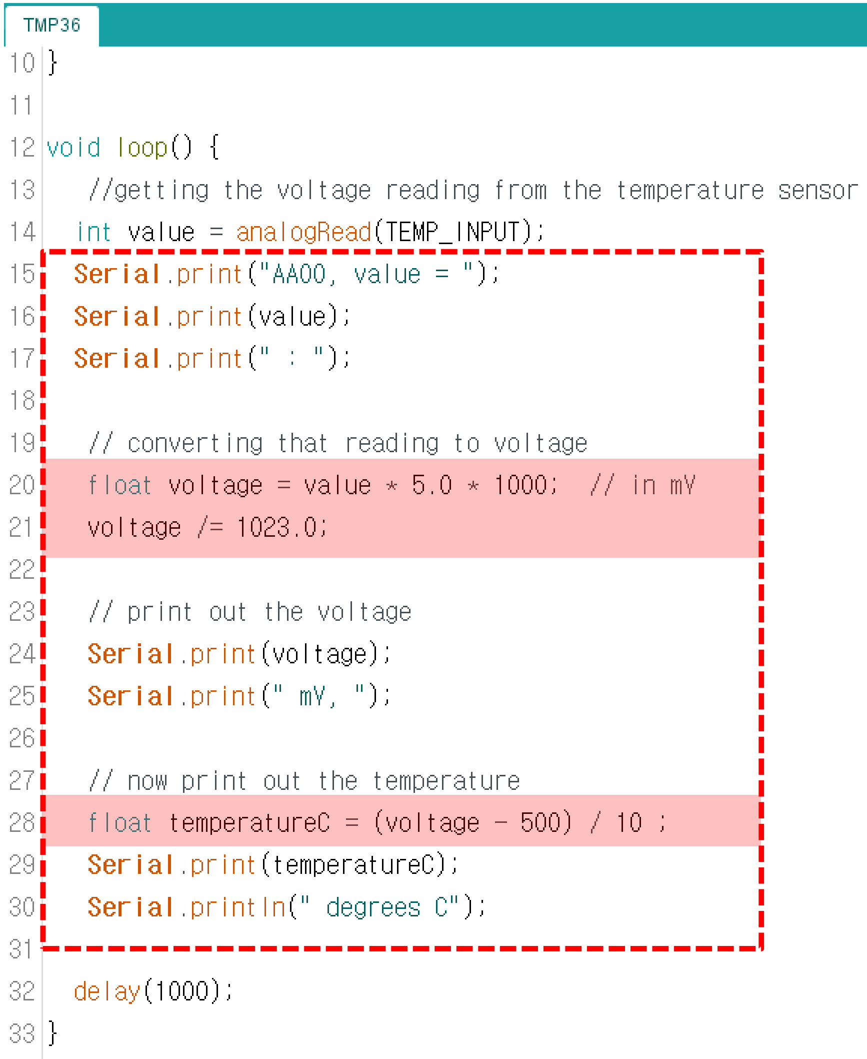
Temp (° C) = (Vout – 500) / 10

Vout (mV) = value * (5000 / 1023)

(0 <= value <= 1023)

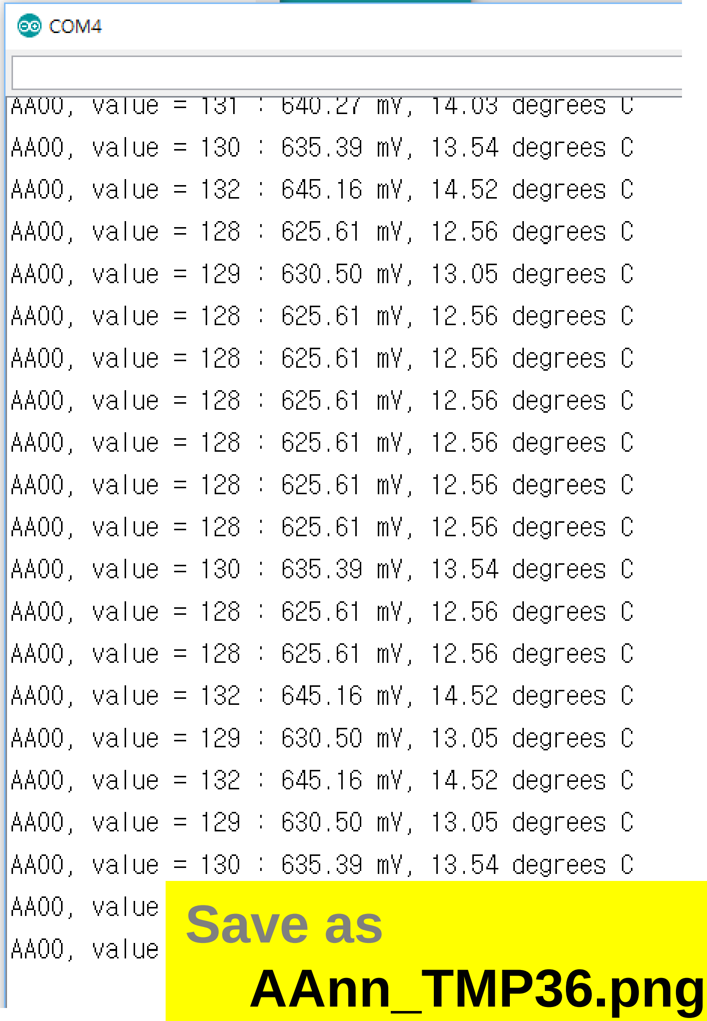
A3.1.5 Temperature sensor [TMP36]

Working code



```
}

void loop() {
   //getting the voltage reading from the temperature sensor
  int value = analogRead(TEMP_INPUT);
  Serial.print("AA00, value = ");
  Serial.print(value);
  Serial.print(" : ");

   // converting that reading to voltage
   float voltage = value * 5.0 * 1000;  // in mV
   voltage /= 1023.0;

   // print out the voltage
   Serial.print(voltage);
   Serial.print(" mV, ");

   // now print out the temperature
   float temperatureC = (voltage - 500) / 10 ;
   Serial.print(temperatureC);
   Serial.println(" degrees C");

  delay(1000);
}
```

Serial output (°C)

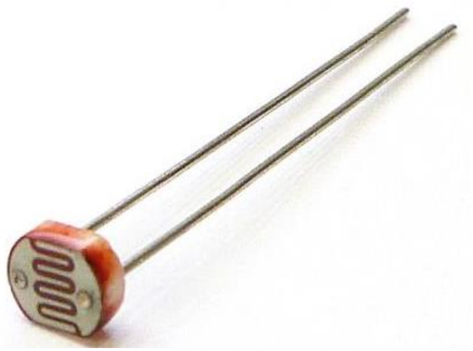


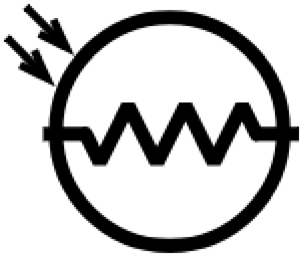
```
AA00, value = 131 : 640.27 mV, 14.03 degrees C
AA00, value = 130 : 635.39 mV, 13.54 degrees C
AA00, value = 132 : 645.16 mV, 14.52 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 129 : 630.50 mV, 13.05 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128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

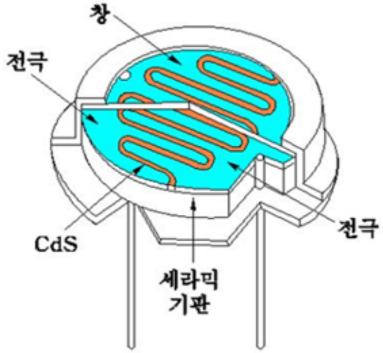
A3.2 Luminosity sensor [Photocell LDR]

CdS 센서 - photoresistor





CDS특성

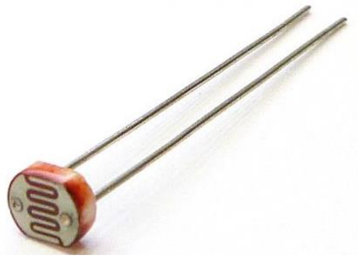


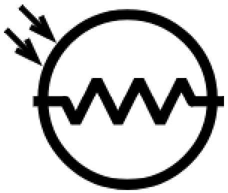
1. 감도
   - 빛의 파장에 따라 감도가 다름
2. 허용손실
   - 비교적 큰 전류를 흘릴 수 있음
3. 암 전류
   - 빛이 없어도 약간의 전류가 흐름
4. 명 전류
   - 빛을 비추면 흐르는 전류
5. 응답특성
   - 응답 시간 지연
   - 빛의 세기에 따라 응답시간 다름
6. 가변저항
   - 빛에 따른 가변저항

https://en.wikipedia.org/wiki/Photoresistor

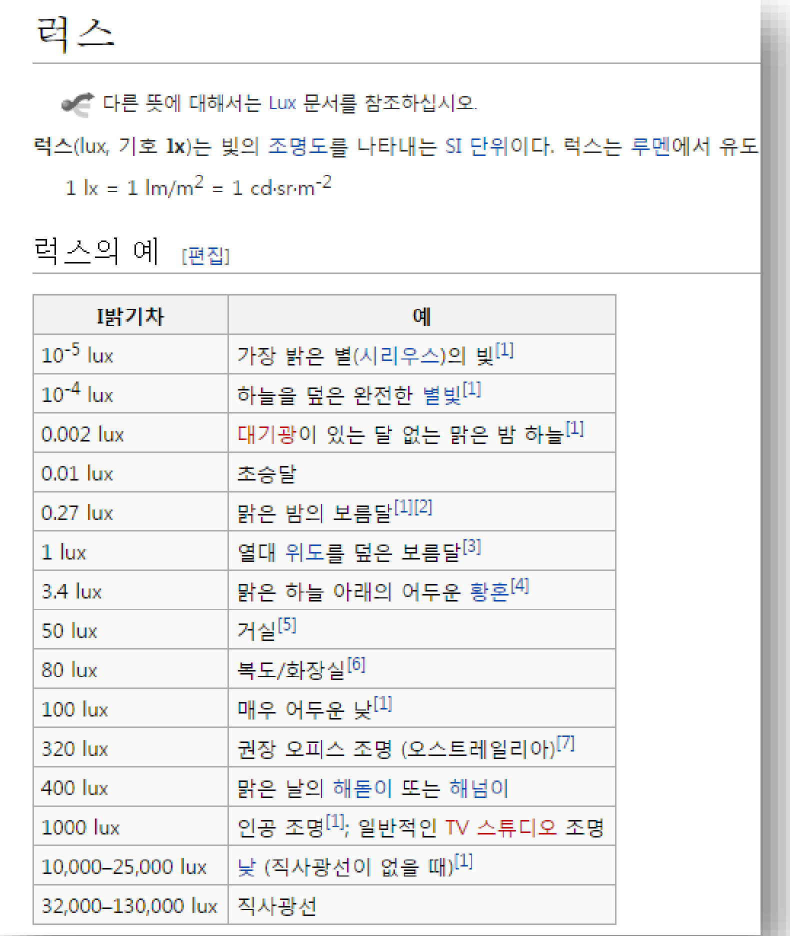
A3.2.1 Luminosity sensor [Photocell LDR]

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 조명도를 나타내는 SI 단위이다. 럭스는 루멘에서 유도

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

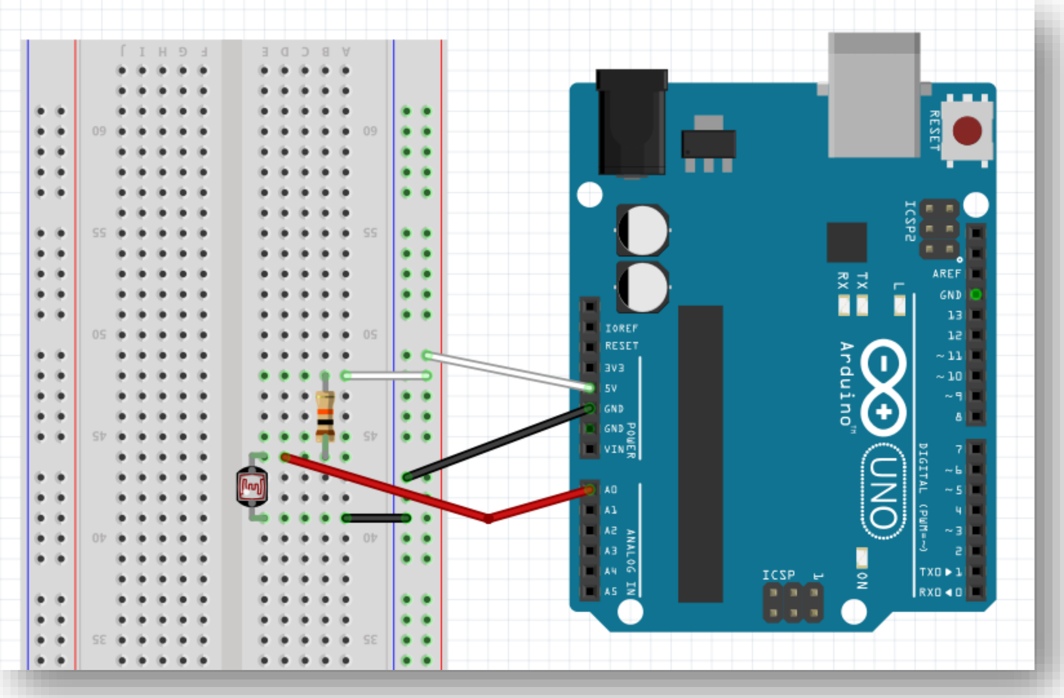
럭스의 예 [편집]

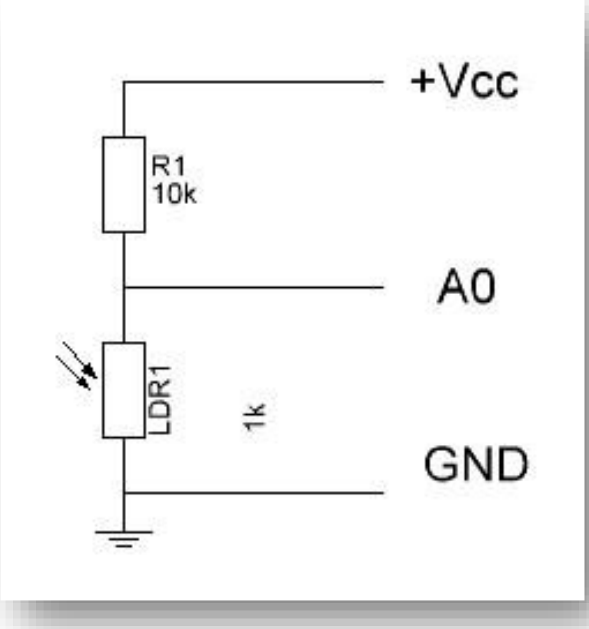
I밝기차	예
10^{-5} lux	가장 밝은 별(시리우스)의 빛[1]
10^{-4} lux	하늘을 덮은 완전한 별빛[1]
0.002 lux	대기광이 있는 달 없는 맑은 밤 하늘[1]
0.01 lux	초승달
0.27 lux	맑은 밤의 보름달[1][2]
1 lux	열대 위도를 덮은 보름달[3]
3.4 lux	맑은 하늘 아래의 어두운 황혼[4]
50 lux	거실[5]
80 lux	복도/화장실[6]
100 lux	매우 어두운 낮[1]
320 lux	권장 오피스 조명 (오스트레일리아)[7]
400 lux	맑은 날의 해돋이 또는 해넘이
1000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

https://en.wikipedia.org/wiki/Photoresistor

A3.2.2 Luminosity sensor [Photocell LDR]

CdS 센서 회로





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

광센서에서의 전압 강하 값을 **A0**로 측정

A3.2.3 Luminosity sensor [Photocell LDR]

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)%

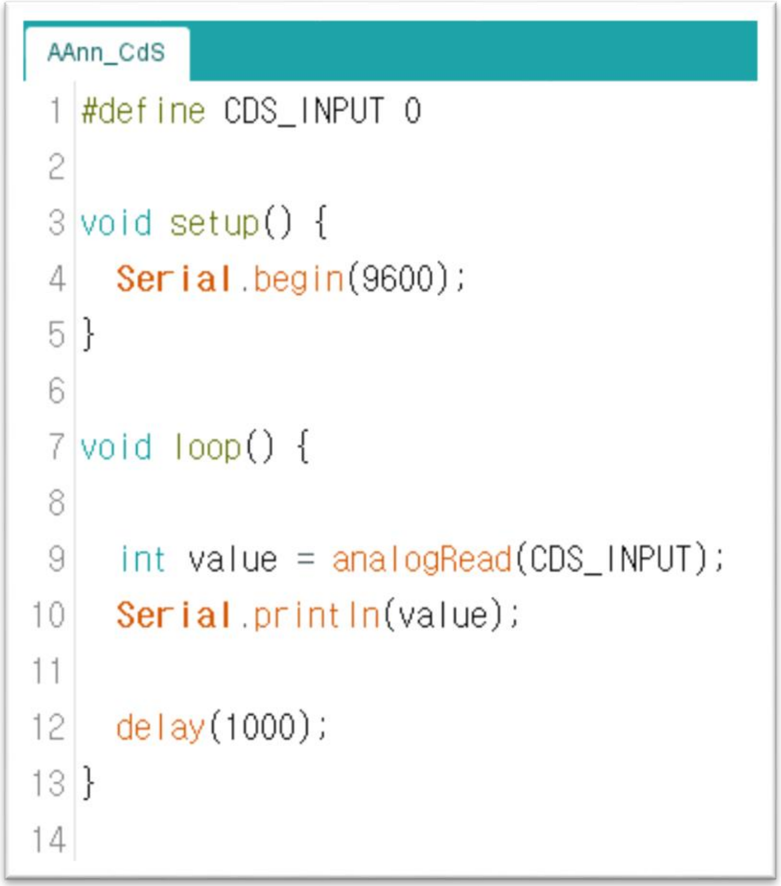
3. ADC 값을 시리얼모니터로 lux 값으로 출력한다.

Hardware

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

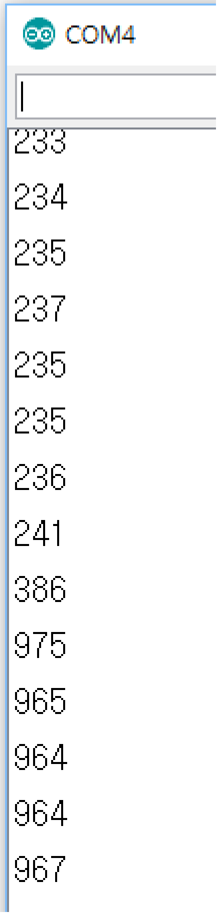
A3.2.4 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 1.



AAnn_CdS

```
#define CDS_INPUT 0

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

void loop() {

  int value = analogRead(CDS_INPUT);
  Serial.println(value);

  delay(1000);
}

```

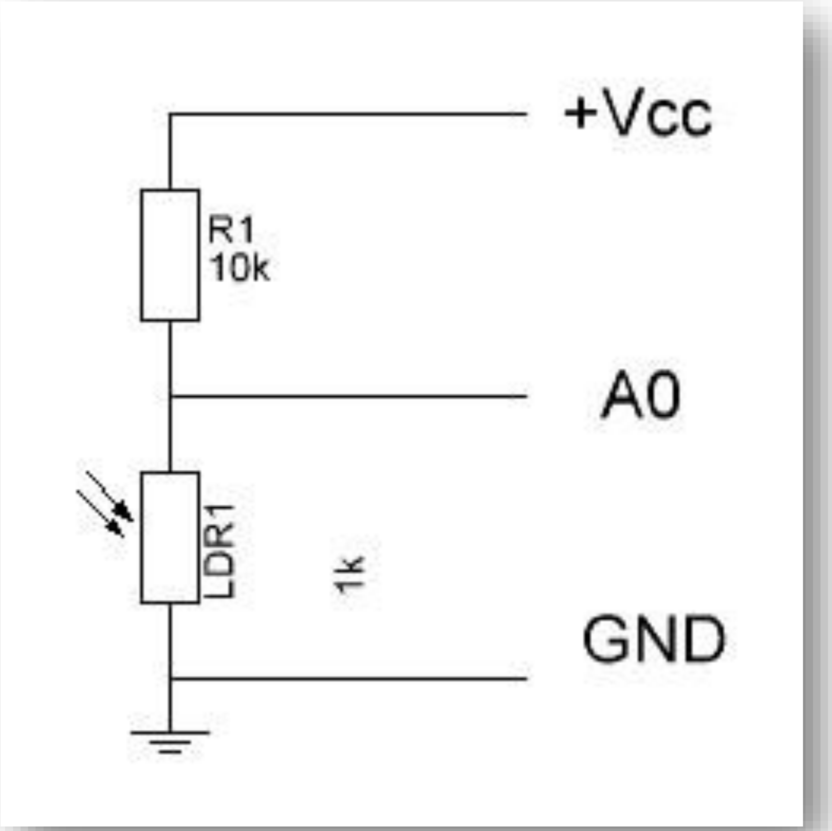


COM4

```
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 us 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.

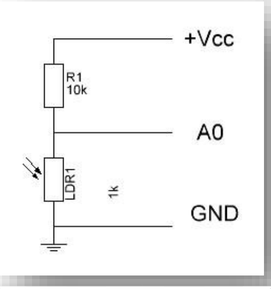
어두우면 측정 값이 작아지고 밝을수록 값이 커져야 된다.
그리고 측정 값은 lux로 표현된다.

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

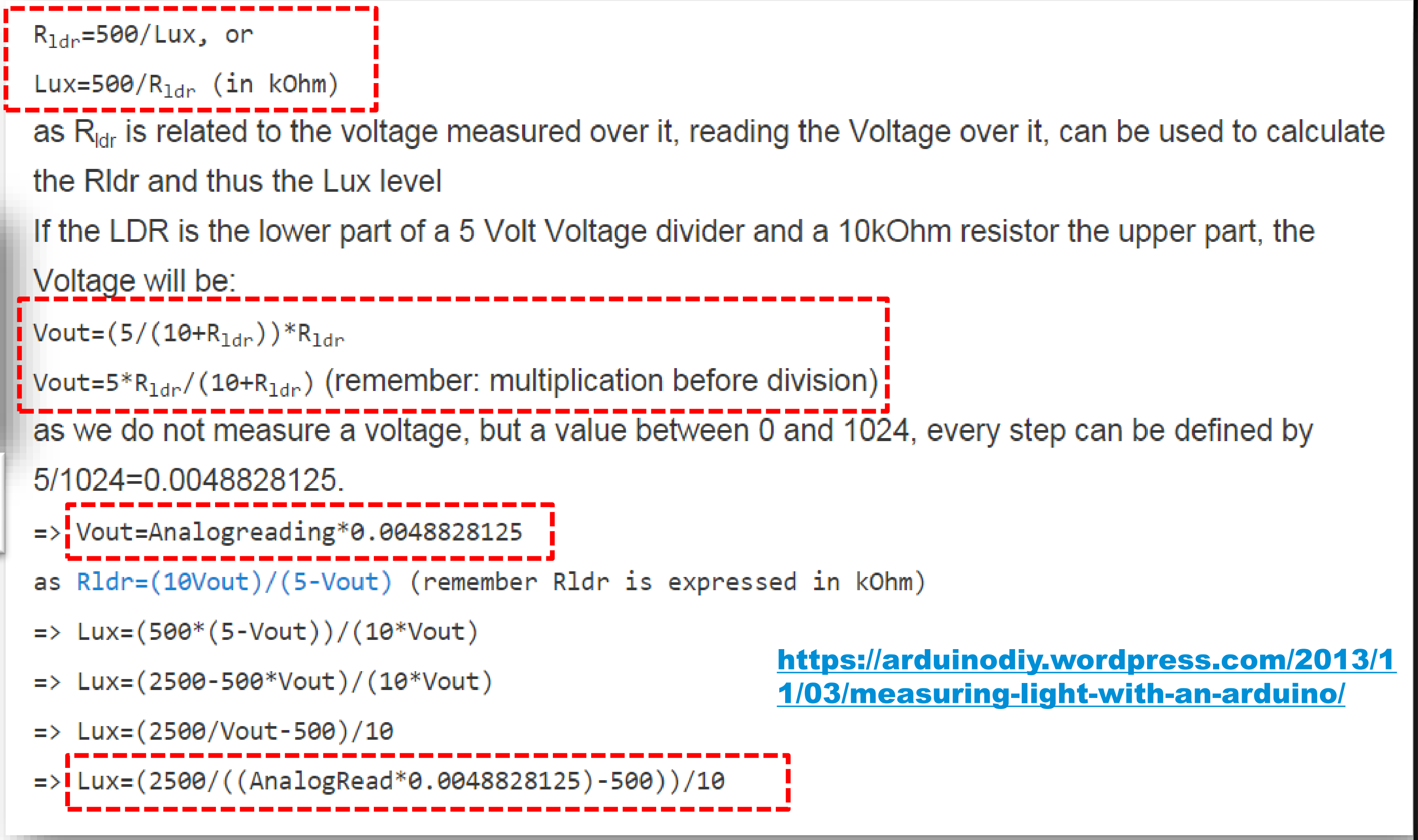
A0에서 측정되는 LDR 양단의 전압 = V_{out}

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?



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



R_{ldr}=500/Lux, or

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

as R_{ldr} is related to the voltage measured over it, reading the Voltage over it, can be used to calculate the Rldr 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:

Vout=(5/(10+R_{ldr}))*R_{ldr}

Vout=5*R_{ldr}/(10+R_{ldr}) (remember: multiplication before division)

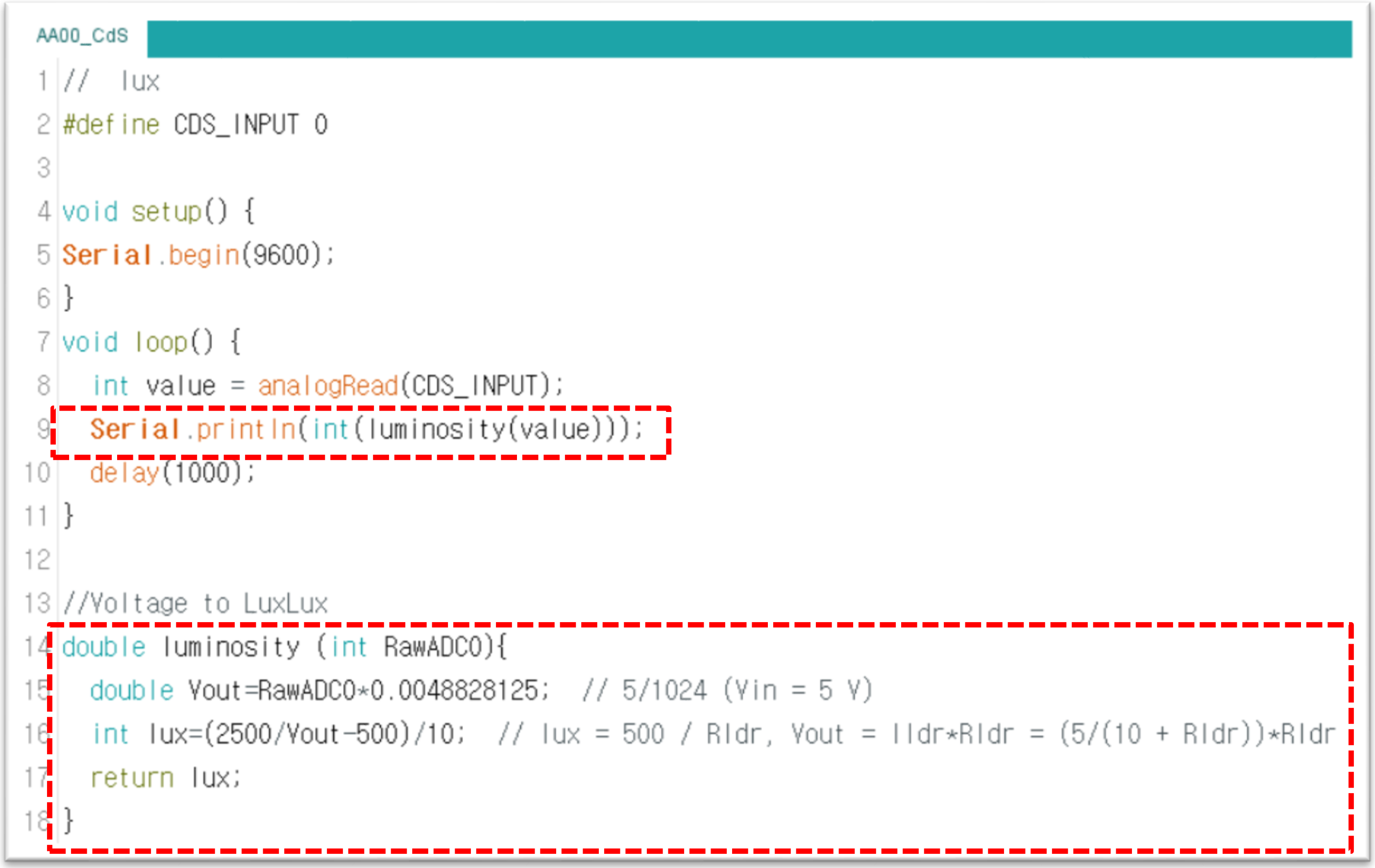
as we do not measure a voltage, but a value between 0 and 1024, every step can be defined by 5/1024=0.0048828125.

```
=> Vout=Analogreading*0.0048828125
as Rldr=(10Vout)/(5-Vout) (remember Rldr is expressed in kOhm)
=> Lux=(500*(5-Vout))/(10*Vout)
=> Lux=(2500-500*Vout)/(10*Vout)
=> Lux=(2500/Vout-500)/10
=> Lux=(2500/((AnalogRead*0.0048828125)-500))/10
```

https://arduinodiy.wordpress.com/2013/11/03/measuring-light-with-an-arduino/

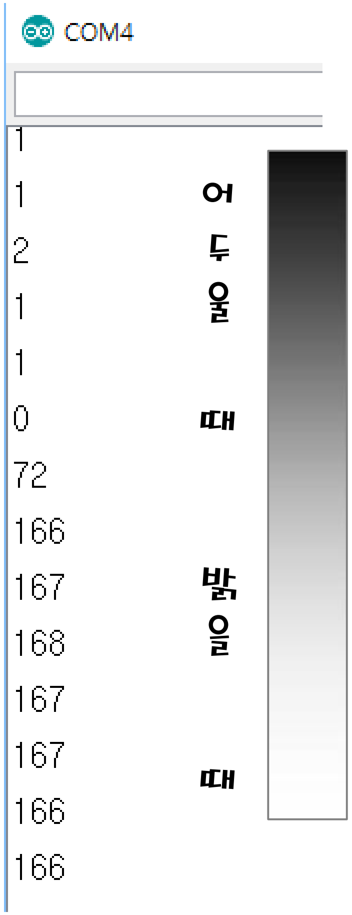
A3.2.5 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 2.



AA00_CdS

```
//  lux
#define CDS_INPUT 0

void setup() {
Serial.begin(9600);
}
void loop() {
  int value = analogRead(CDS_INPUT);
  Serial.println(int(luminosity(value)));
  delay(1000);
}

//Voltage to LuxLux
double luminosity (int RawADC0){
  double Vout=RawADC0*0.0048828125;   // 5/1024 (Vin = 5 V)
  int lux=(2500/Vout-500)/10;   // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
  return lux;
}
```



COM4

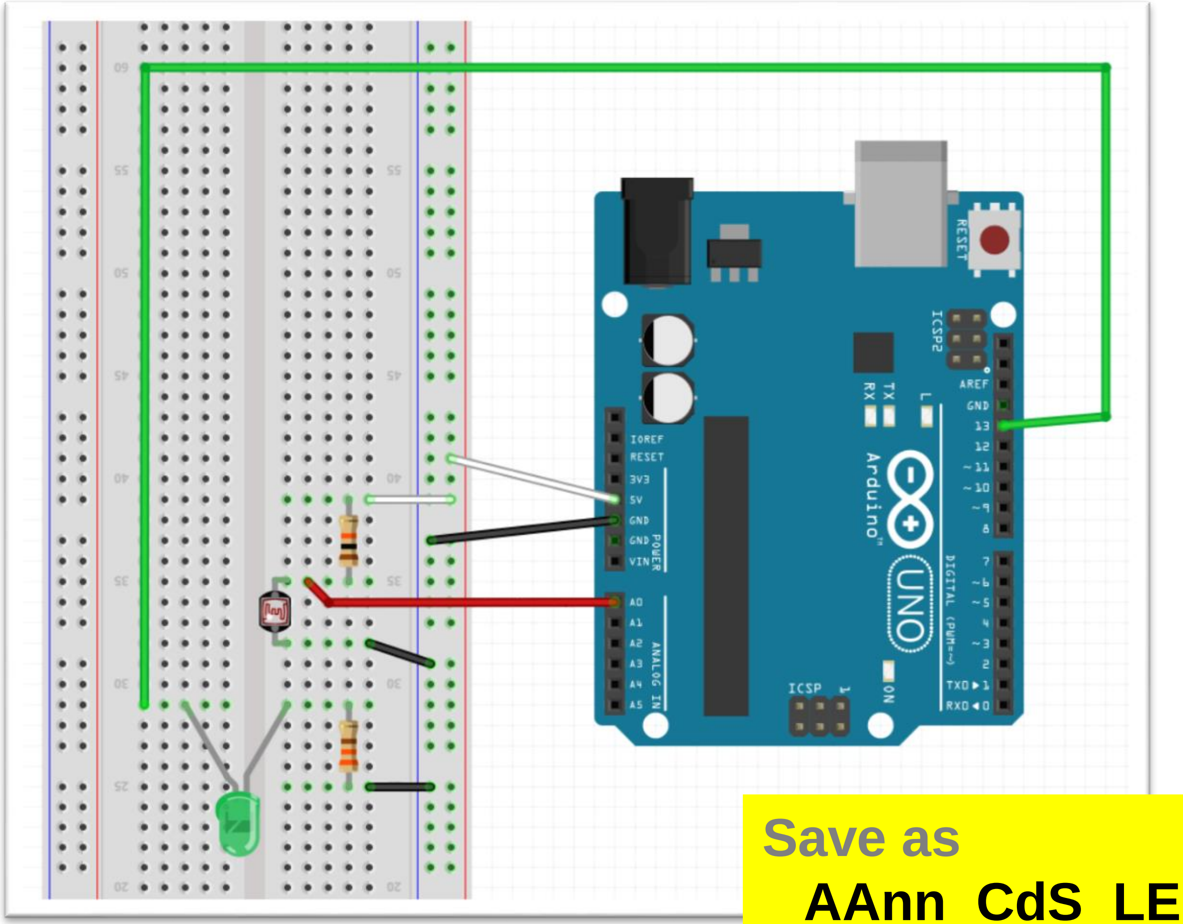
```
1
1
2
1
1
0
72
166
167
168
167
167
166
166
```

밝을수록 측정 값이 커지고 어두울수록 값이 작아진다 !!!

A3.2.6 Luminosity sensor [Photocell LDR]

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

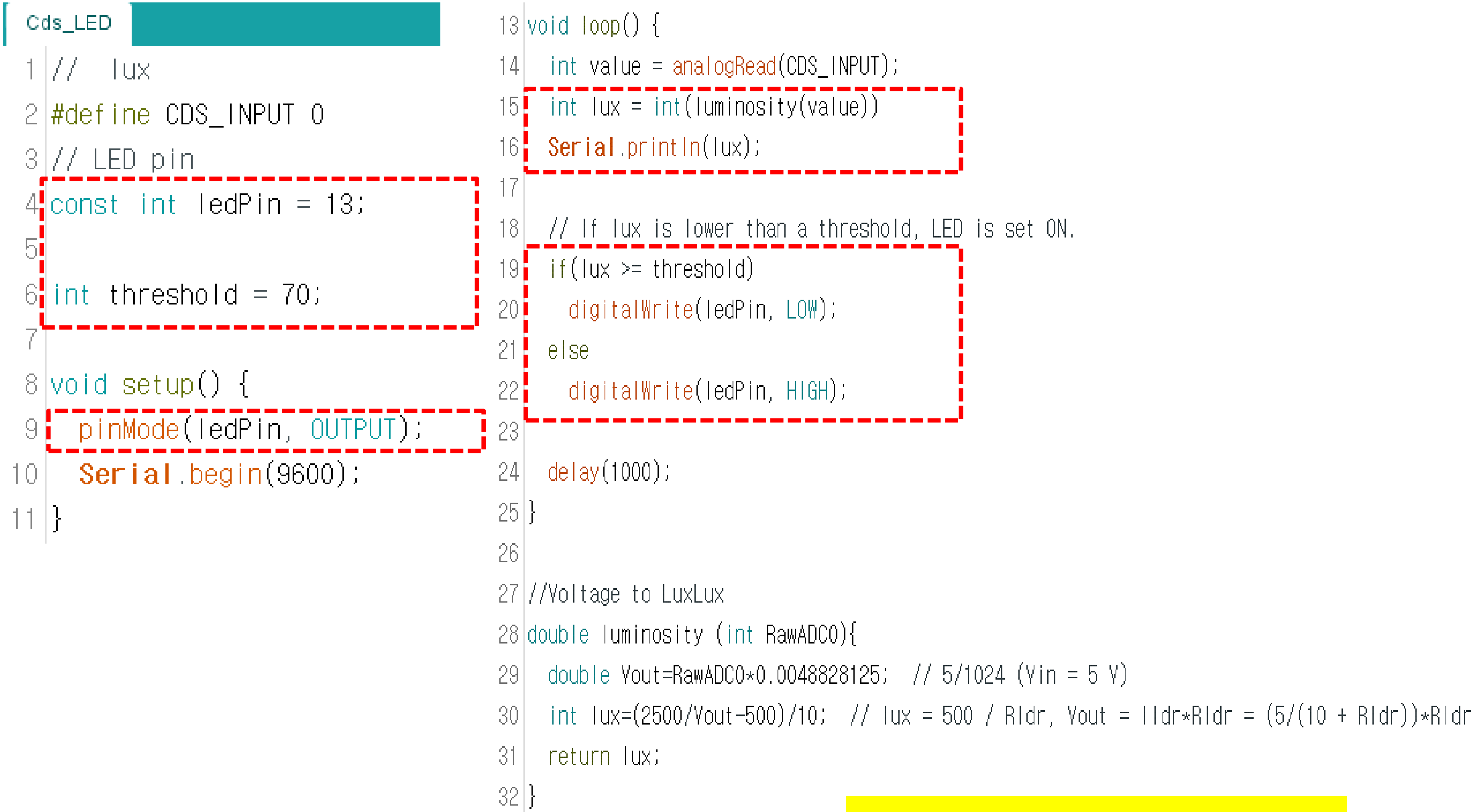
Write down your code here to complete the task that turns on LED when luminosity of ambient light becomes lower than a threshold.

조도 값이 문턱 값 이상이면 LED를 OFF, 그렇지 않으면 ON.

Save as
AAnn_CdS_LED.ino

A3.2.7 Luminosity sensor [Photocell CdS LDR]

DIY Code



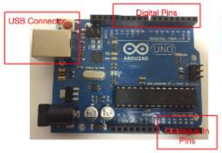
Cds_LED

```
//   lux
#define CDS_INPUT 0
// LED pin
const int ledPin = 13;

int threshold = 70;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  int value = analogRead(CDS_INPUT);
  int lux = int(luminosity(value))
  Serial.println(lux);

  // If lux is lower than a threshold, LED is set ON.
  if(lux >= threshold)
    digitalWrite(ledPin, LOW);
  else
    digitalWrite(ledPin, HIGH);

  delay(1000);
}

//Voltage to LuxLux
double luminosity (int RawADC0){
  double Vout=RawADC0*0.0048828125;  // 5/1024 (Vin = 5 V)
  int lux=(2500/Vout-500)/10;  // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
  return lux;
}
```

Save as

AAnn_CdS_LED.ino

IOT: HSC



Arduino Board

```
#define [illegible]_INPUT 0

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

void loop() {

  int value = analogRead([illegible]_INPUT);
  Serial.println(value);

  delay([illegible]);
}
```

Arduino IDE

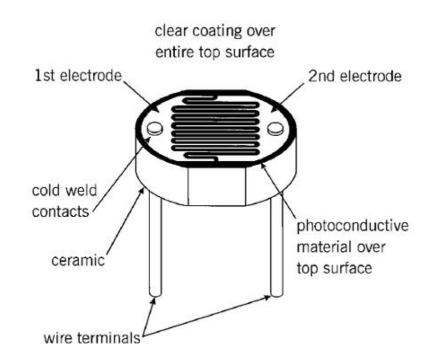
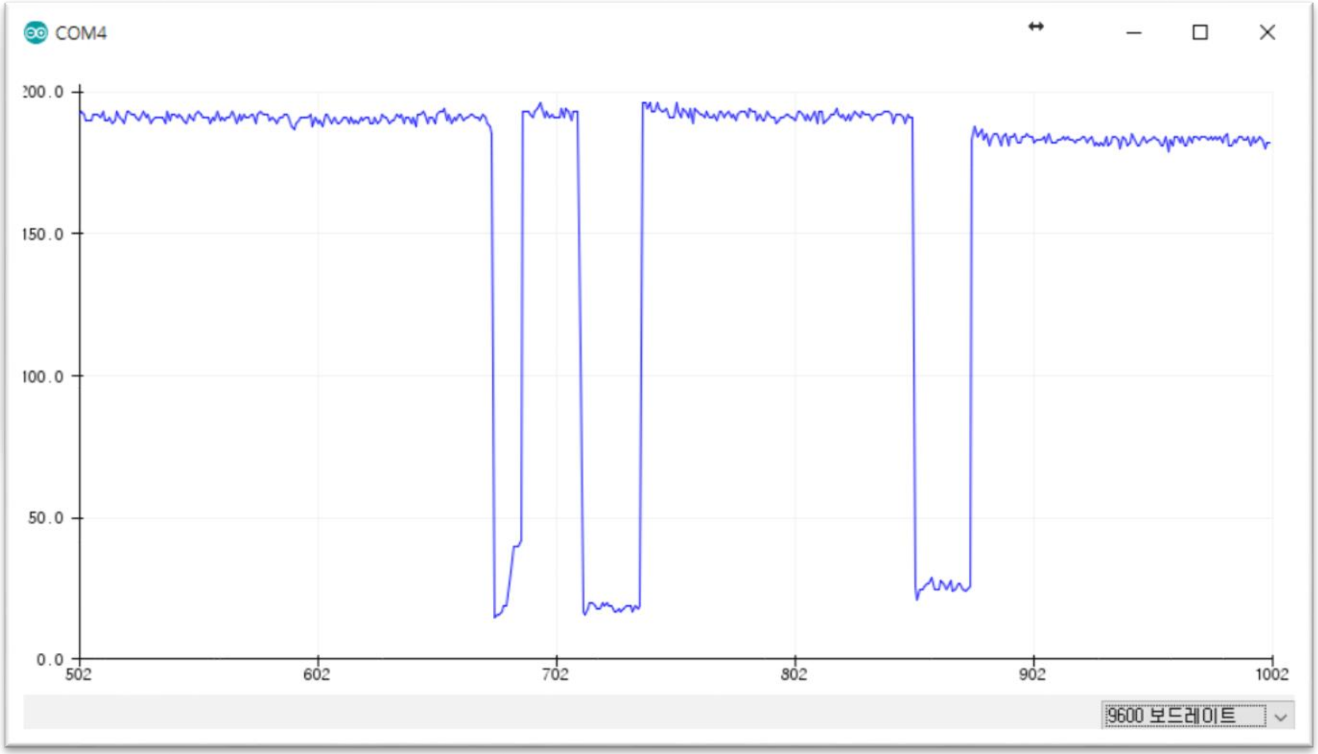
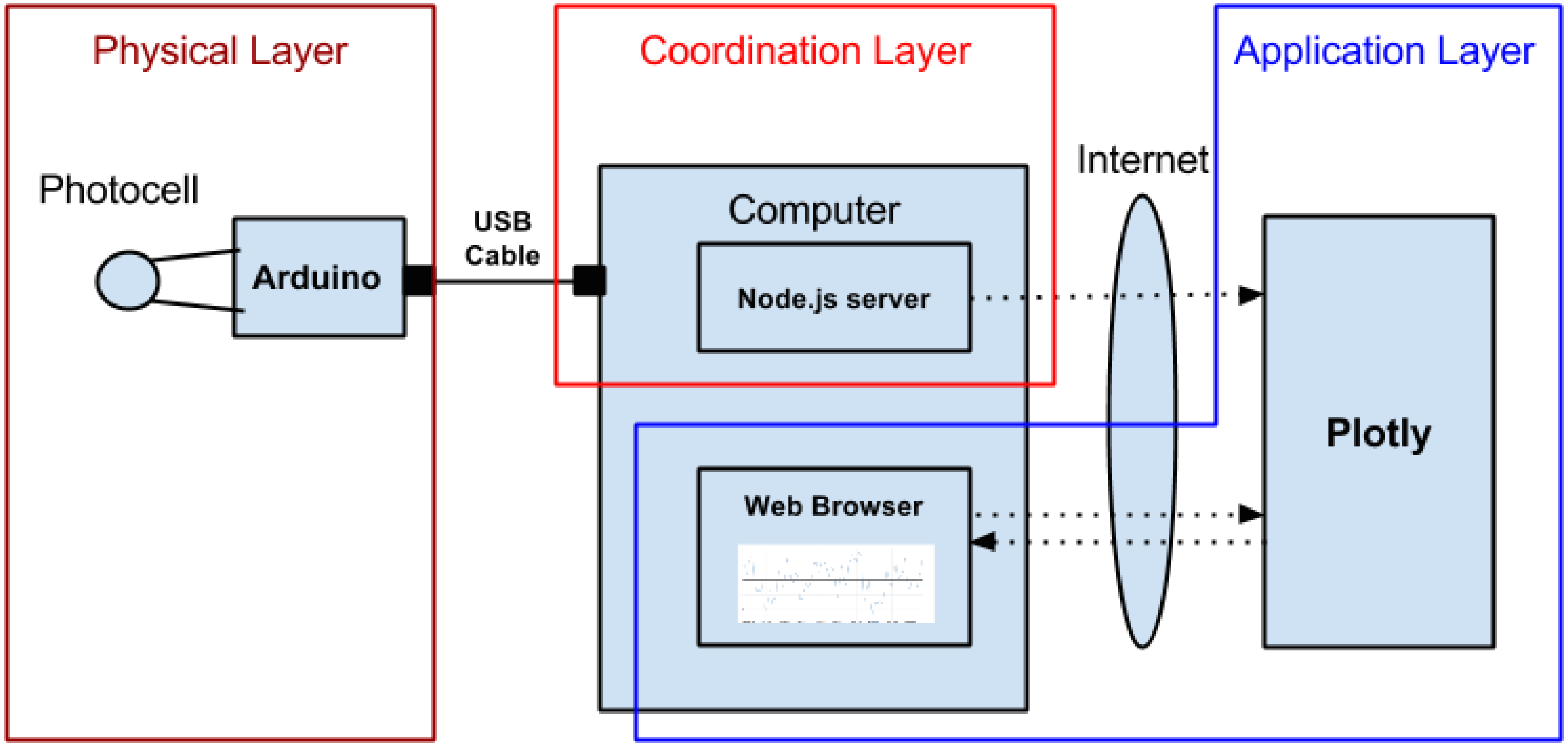


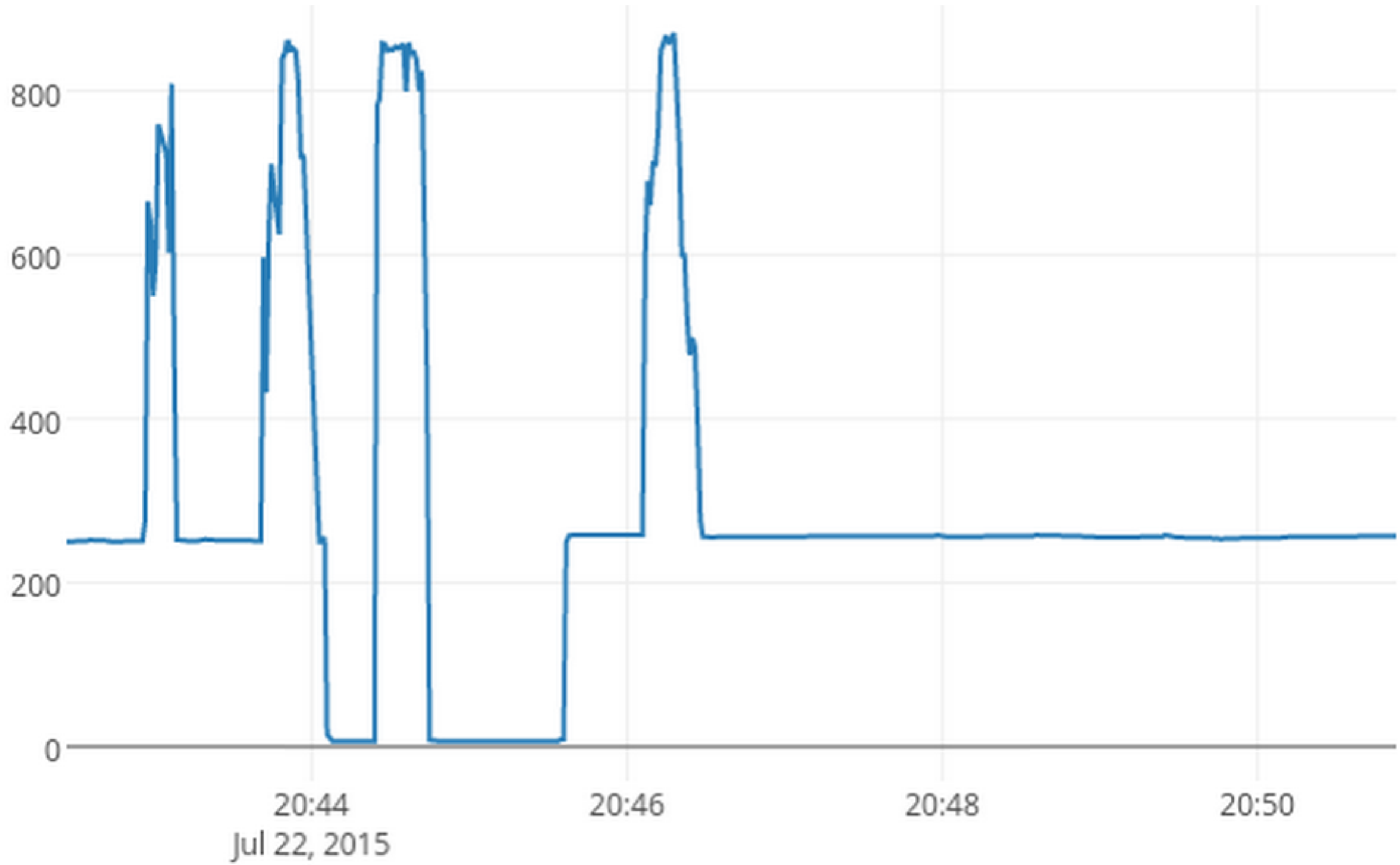
Figure 3
Typical Construction of a Plastic Coated Photocell

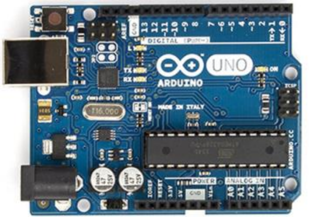


Layout [H S C]


Physical Layer
Coordination Layer
Application Layer
Photocell
Arduino
USB
Cable
Computer
Node.js server
Internet
Plotly
Web Browser

Arduino: node.js + plotly





[Practice]

- [wk06]
  - Arduino sensors
  - Complete your project
  - Upload file name : AAnn_Rpt05.zip

wk06 : Practice-05 : AAnn_Rpt05.zip

◆ **[Target of this week]**

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

제출파일명 : **AAnn_Rpt05.zip**

- 압축할 파일들

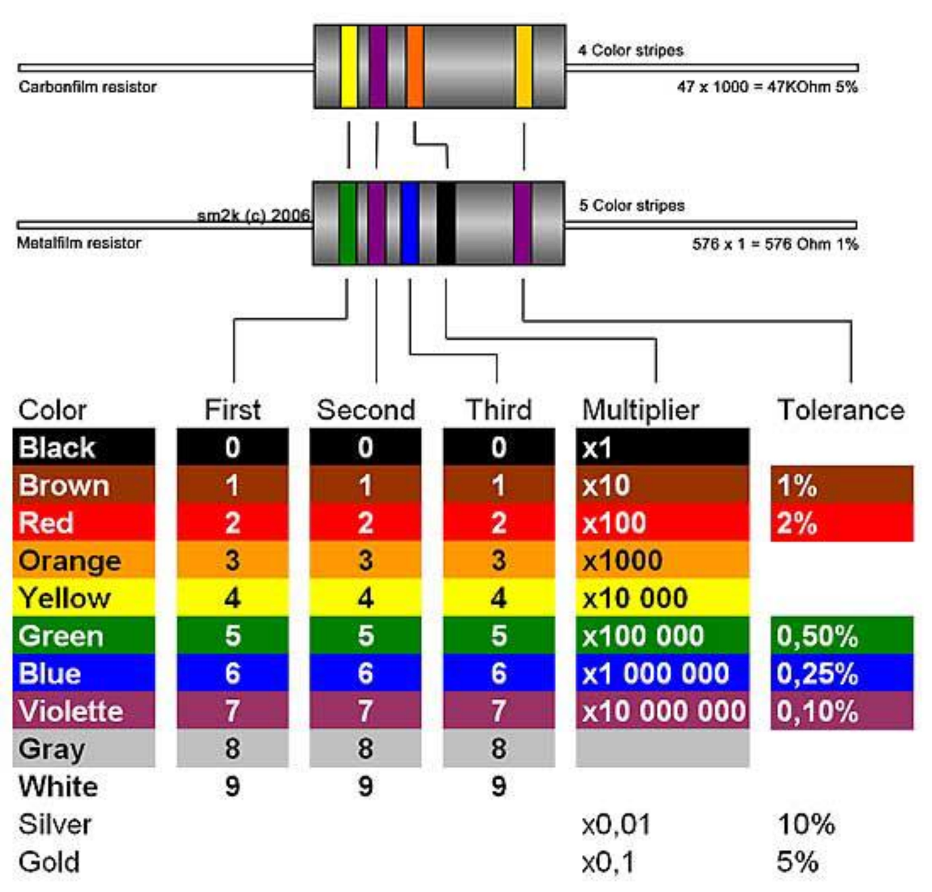
① **AAnn_TMP36.png**

② **AAnn_CdS_LED.fzz**

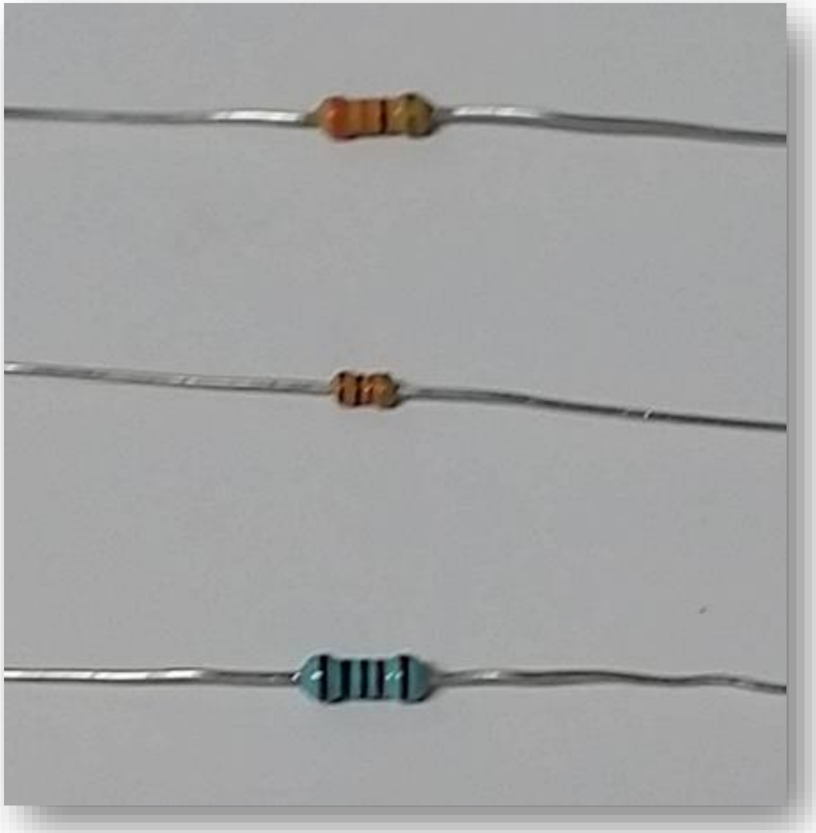
③ **AAnn_CdS_LED.ino**

Email : chaos21c@gmail.com

[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%

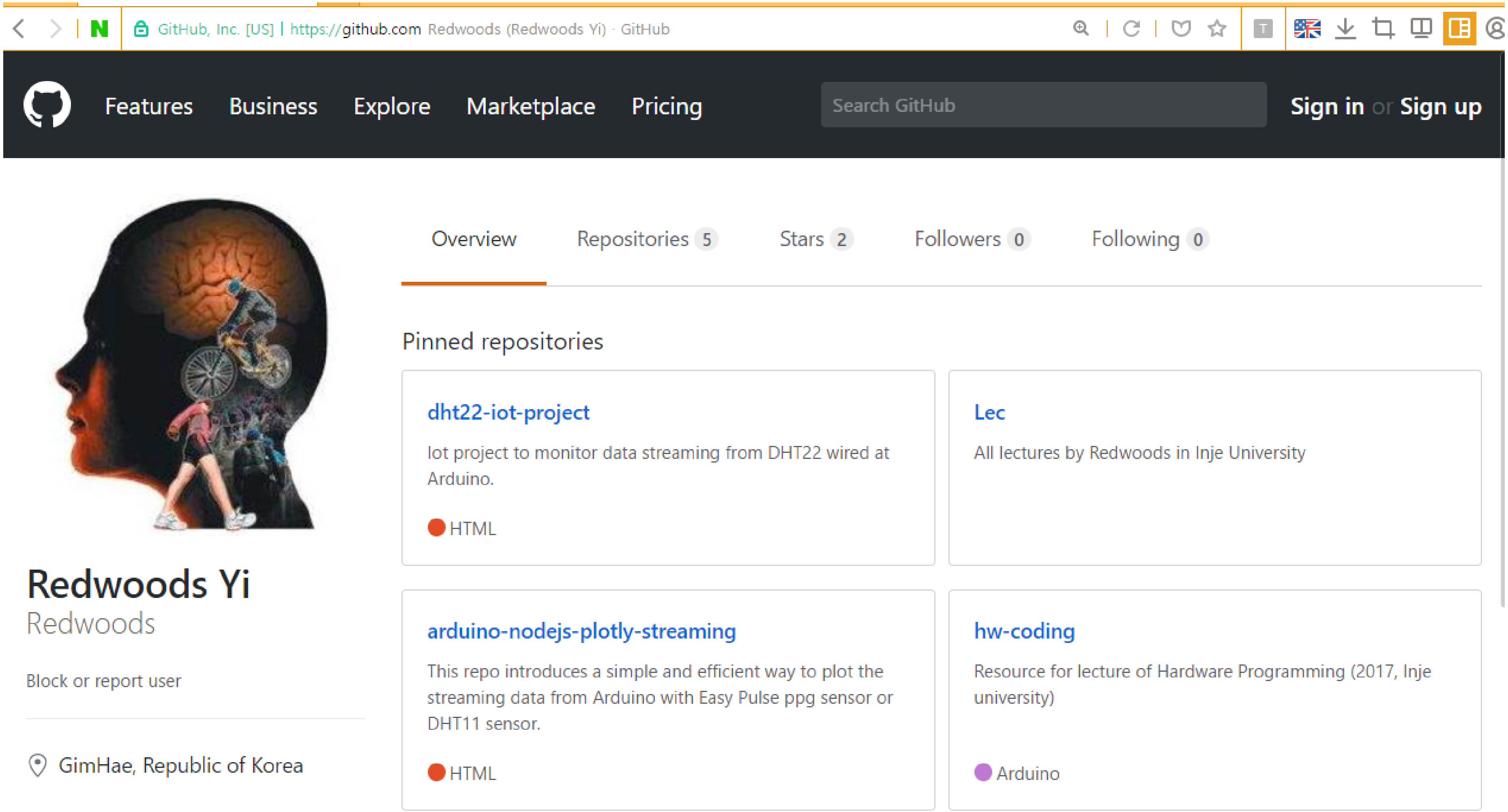


Lecture materials

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

Github.com/Redwoods



GitHub, Inc. [US] | https://github.com Redwoods (Redwoods Yi) · GitHub

Overview | Repositories 5 | Stars 2 | Followers 0 | Following 0

Pinned repositories

dht22-iot-project
Iot 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

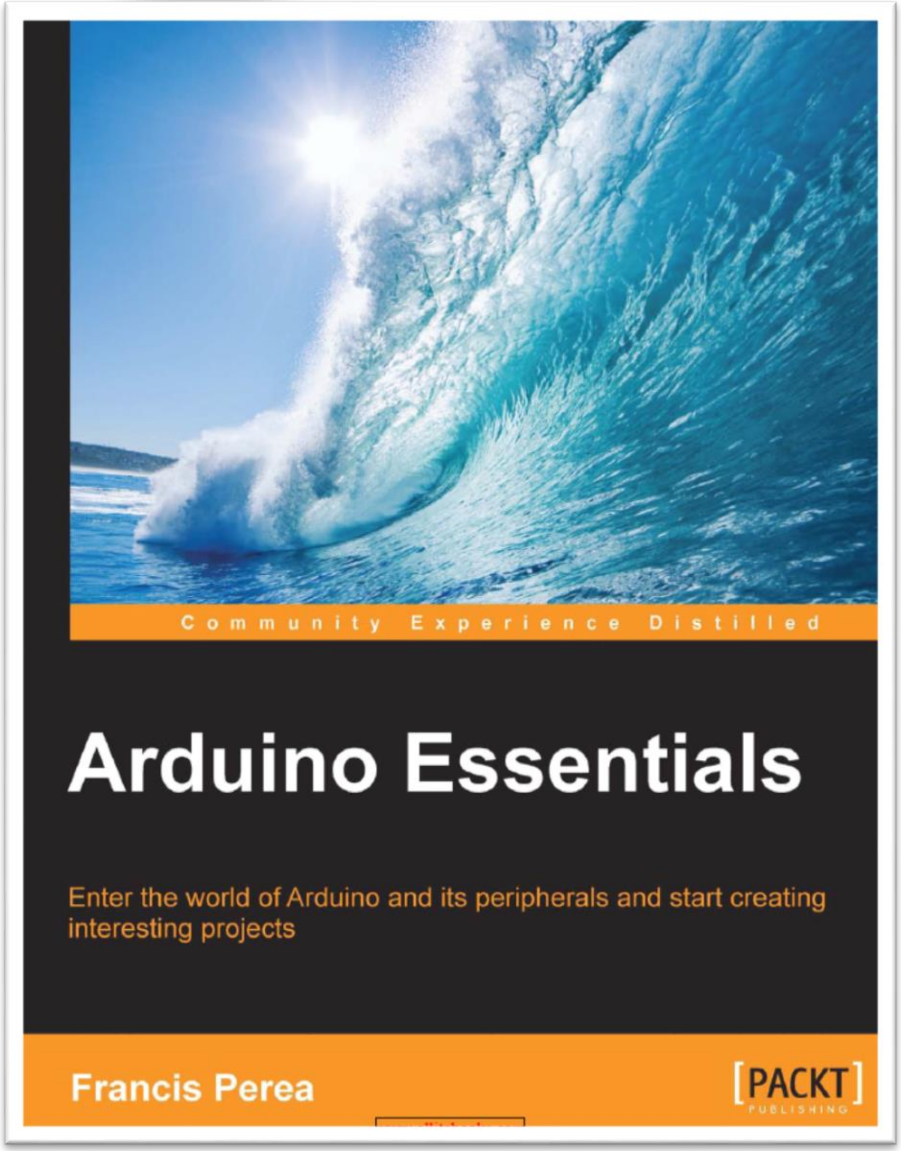
hw-coding
Resource for lecture of Hardware Programming (2017, Inje university)
● Arduino

Redwoods Yi
Redwoods

Block or report user

GimHae, Republic of Korea

References



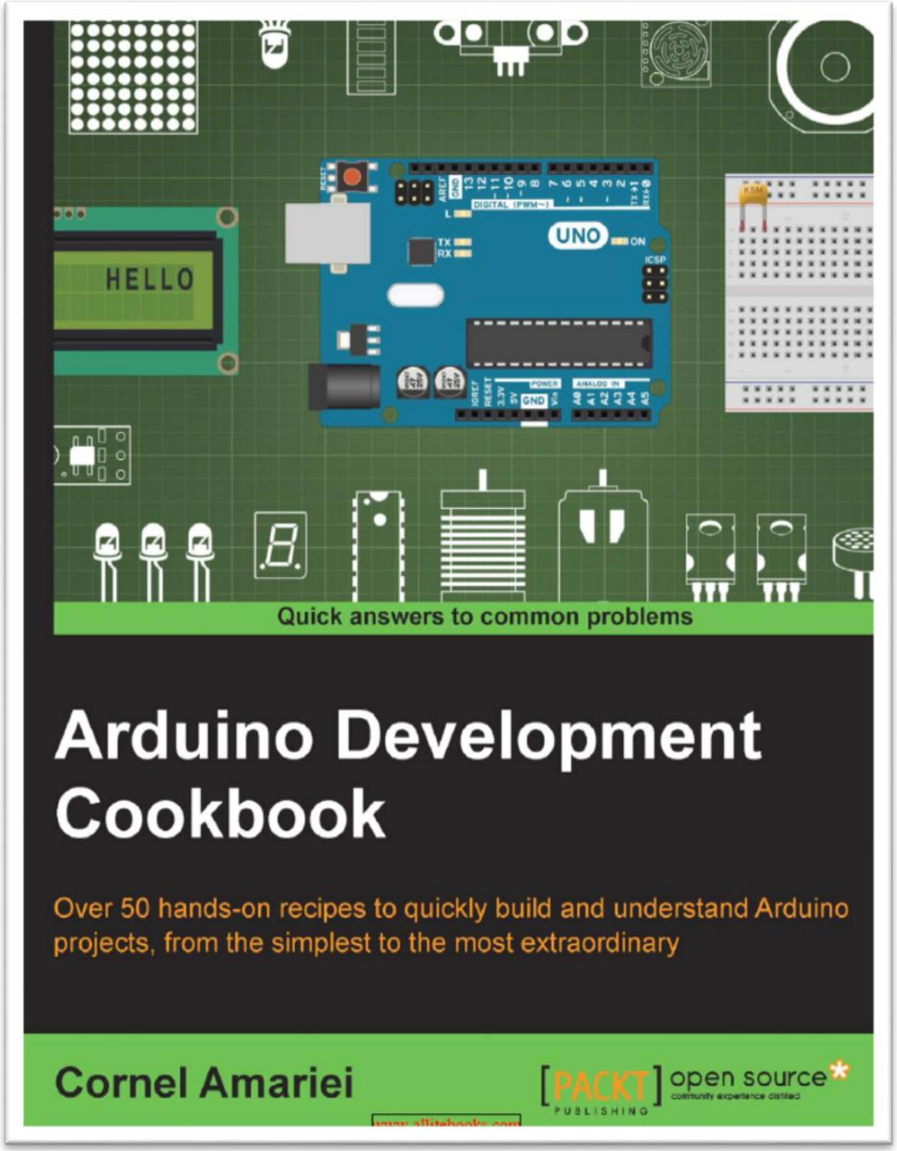
Community Experience Distilled

Arduino Essentials

Enter the world of Arduino and its peripherals and start creating interesting projects

Francis Perea

[PACKT] PUBLISHING



Quick answers to common problems

Arduino Development Cookbook

Over 50 hands-on recipes to quickly build and understand Arduino projects, from the simplest to the most extraordinary

Cornel Amariei

[PACKT] PUBLISHING open source* community experience distilled