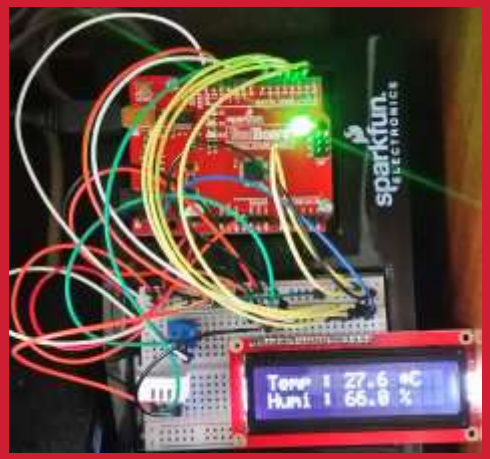


Arduino-IOT

[wk05]

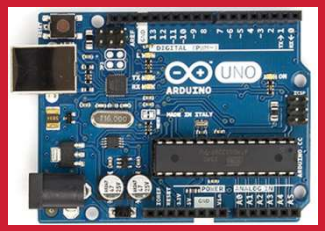
Arduino sensors

Visualization of Signals using Arduino,
Node.js & Storing Signals in MongoDB
& Mining Data using Python

Comsi, INJE University
2nd semester, 2019
Email : chaos21c@gmail.com

My ID

ID	성명
AA01	김관용
AA02	백동진
AA03	김도훈
AA04	김희찬
AA05	류재현
AA06	문민규
AA07	박진석
AA08	이승협
AA09	표혜성
AA10	김다영
AA11	성소진
AA12	김해인
AA13	신송주
AA14	윤지훈



[Review]

- [wk04]
  - Arduino basic circuits
  - Complete your project
  - Submit folder : AAnn_Rpt04

wk04 : Practice-03 : AAnn_Rpt04

- **[Target of this week]**
  - Complete your works
  - Save your outcomes and upload 3 figures in github

제출폴더명 : **AAnn_Rpt04**

- 제출할 파일들

  ① **AAnn_Monitoring.png**

  ② **AAnn_multi_Monitoring.png**

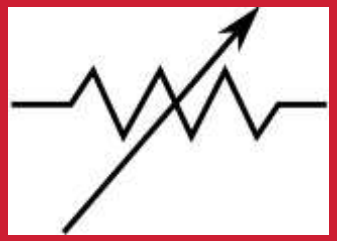
  ③ **AAnn_multi_Signals.png**

  ④ **AAnn_AnalogVoltage.png**

  ⑤ ***.ino**

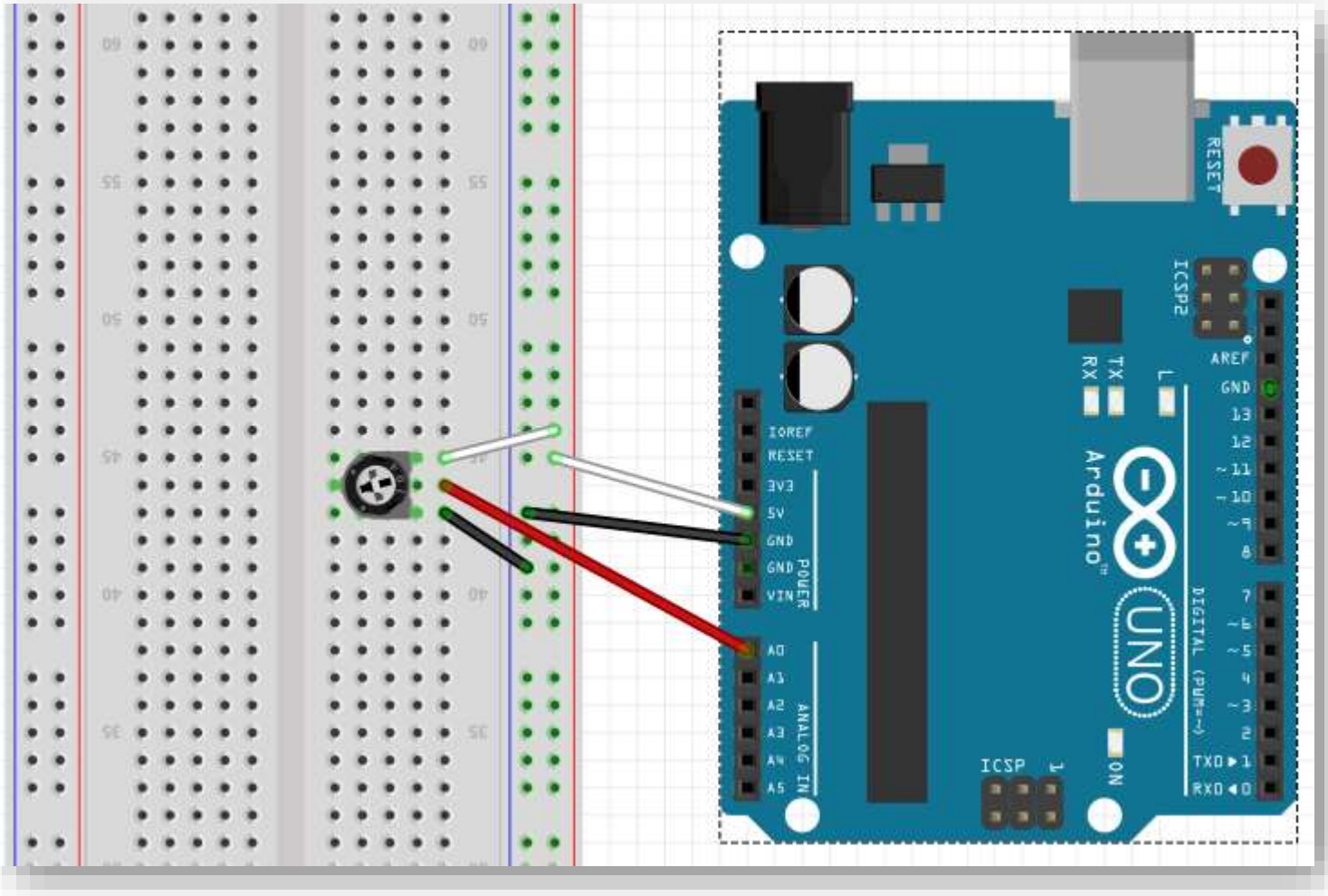
Email : chaos21c@gmail.com

Analog Signal



A2.5.1 AnalogReadSerial (circuit)

Standard potentiometer (가변 저항기)



A2.5.2 AnalogReadSerial (code)

▸ 스케치 구성 (코드 4-1)

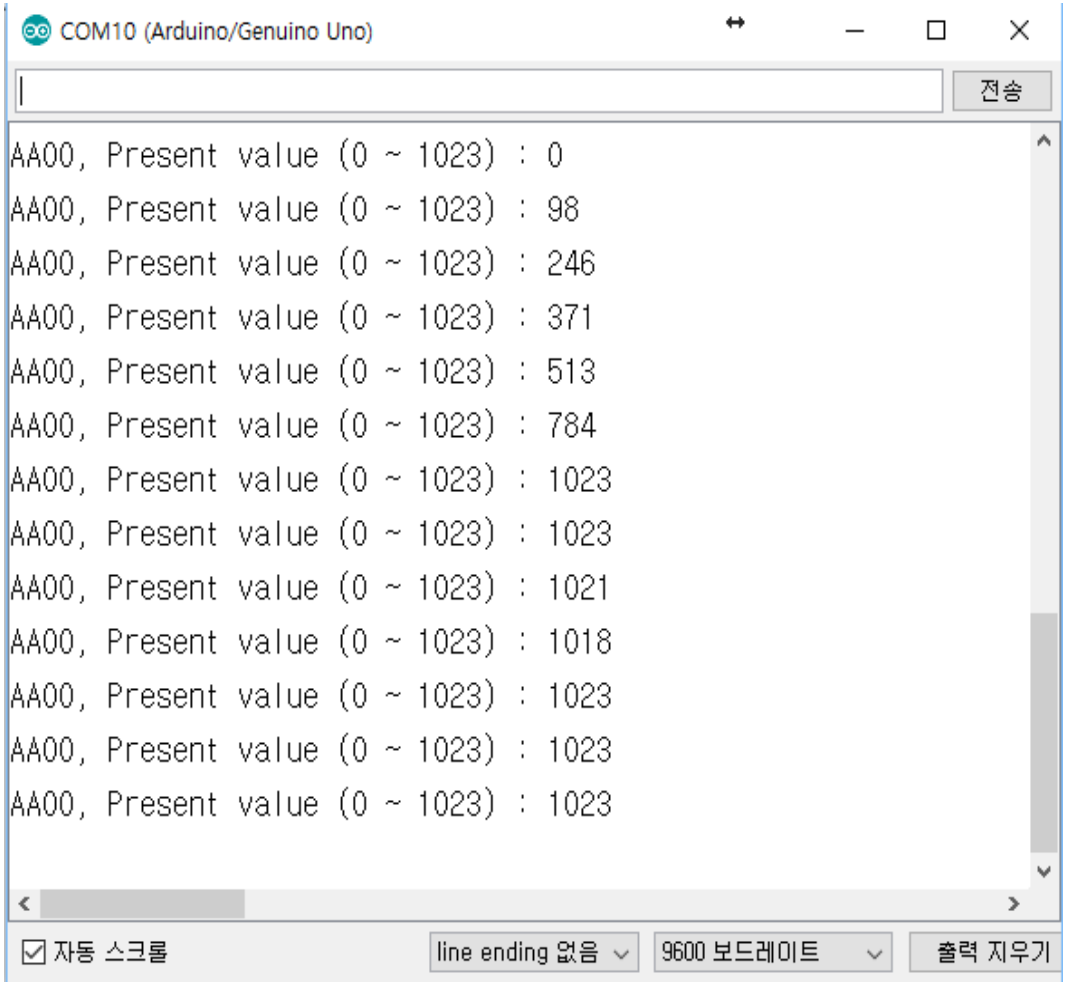
1. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
2. loop()에서 **analogRead()** 함수로 A0 핀에서 측정되는 값을 읽어 들인다.
3. 직렬 통신으로 A0 측정값을 한 줄로 0.5 초 마다 컴퓨터로 전송한다.

▸ 아두이노 코드 : **sketch06_analog_read.ino**

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  Serial.print("AA00, Present value (0 ~ 1023) : ");
  Serial.println(sensorValue);
  delay(500);        // 2 Hz sampling
}
```

A2.5.3 ReadAnalogValue

Serial monitor : 0 < value < 1023



A2.5.4 Analog value to Resistance or Voltage

아날로그 값을 저항 및 전압으로 변환

▶ 저항 또는 전압 환산

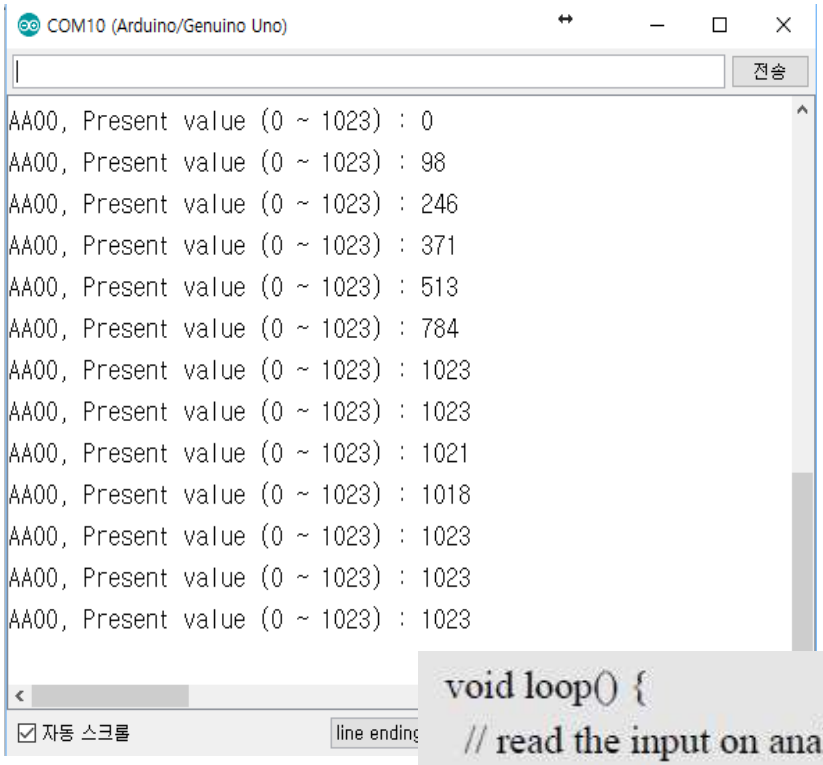
1. 저항 = 10.0 * A0 / 1023 (kΩ)

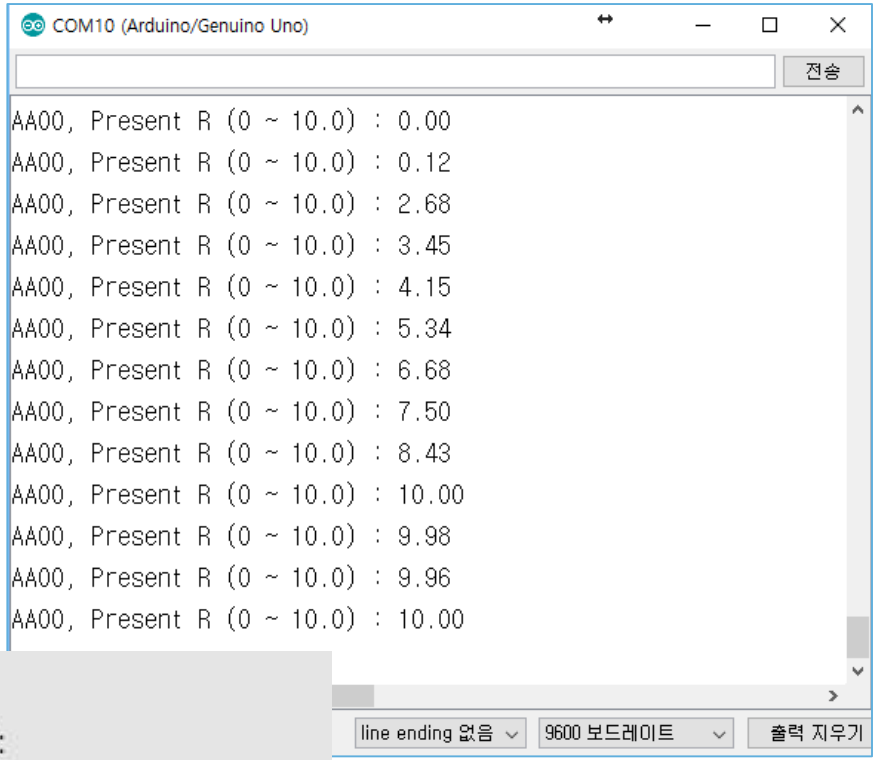
2. 전압 = 5.0 * A0 / 1023 (V)

A0: 아날로그 핀 A0에서의 측정값 (0 ~ 1023)

A2.5.5 Analog value to Resistance

Serial monitor : Resistance (0 < R < 10 kΩ)

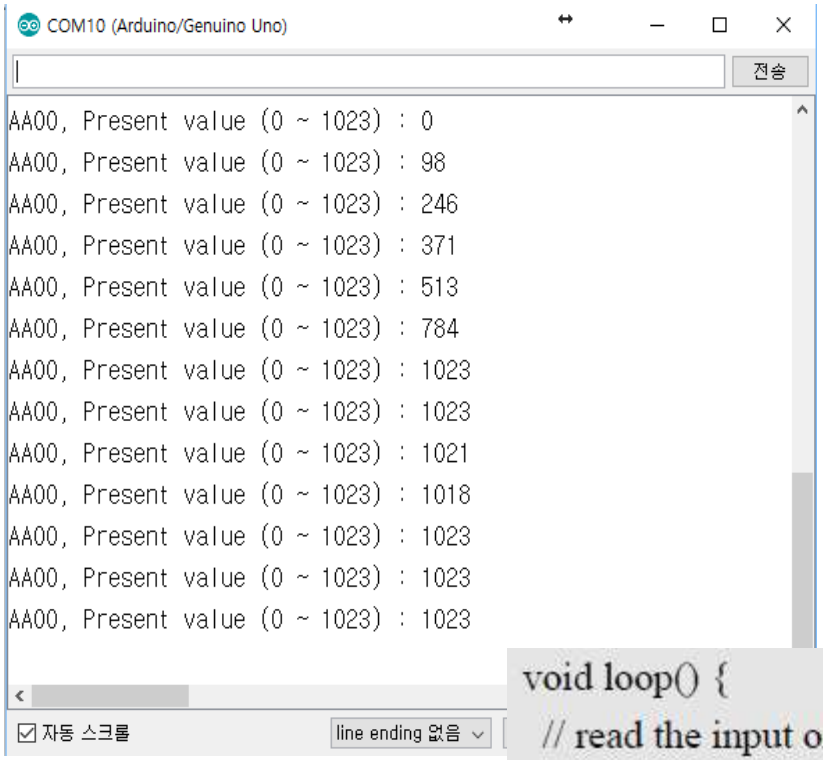


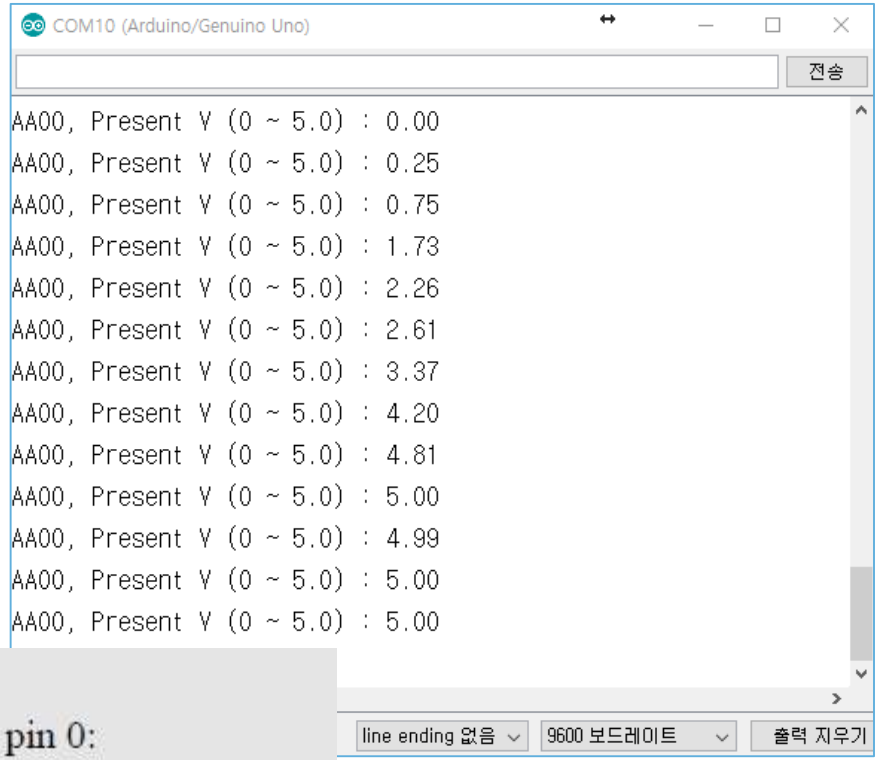



```
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.print("AA00, Present R (0 ~ 10.0) : ");
  float resistance = sensorValue*(10.0/1023.0); // kΩ
  Serial.println(resistance);
  delay(500);        // 2 Hz sampling
}
```

A2.5.6 Analog value to Voltage

Serial monitor : Voltage (0 < V < 5 V)





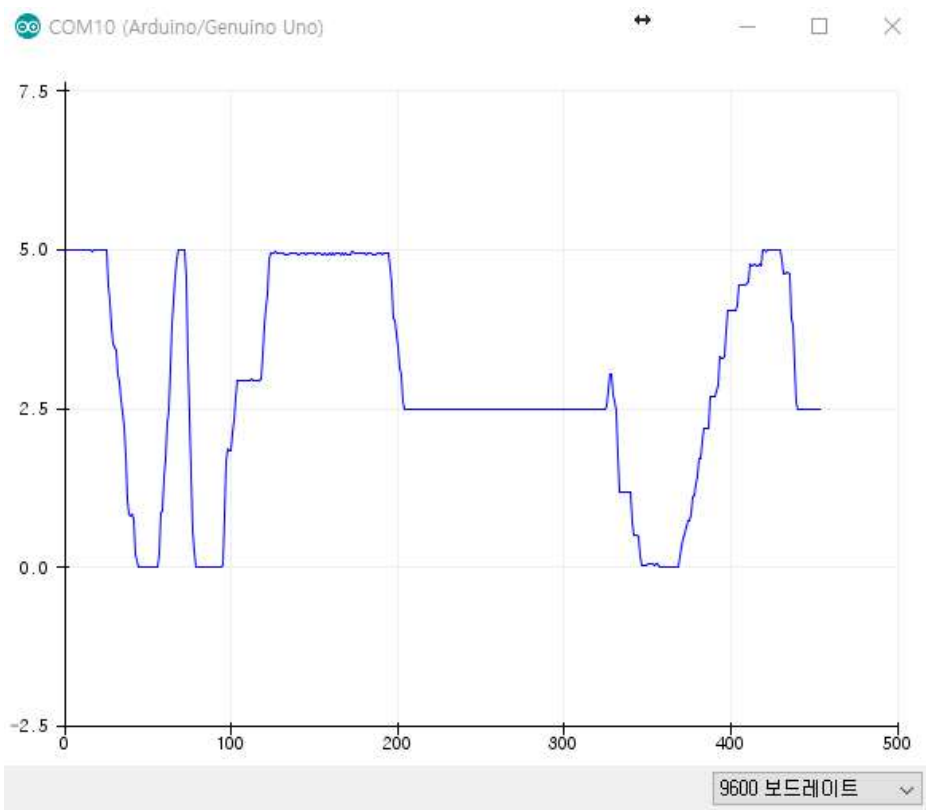

```
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.print("AA00, Present V (0 ~ 5.0) : ");
  float voltage= sensorValue*(5.0/1023.0);  // V
  Serial.println(voltage);
  delay(500);        // 2 Hz sampling
}
```

A2.5.7 ReadAnalogVoltage

Result

COM4

```
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
```



Save as

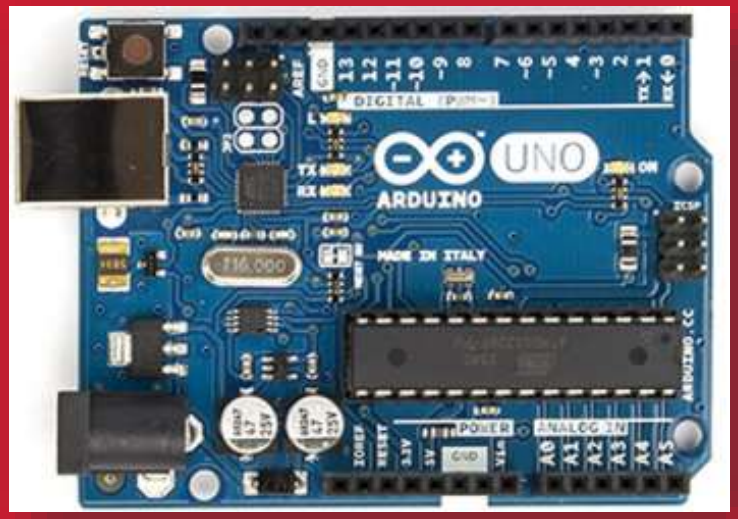
AAnn_AnalogVoltage.png

A2.5.8 ReadAnalogVoltage using f_map()

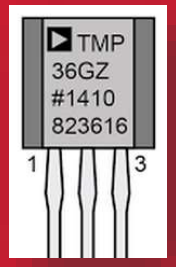
Hint code : f_map() instead of map()

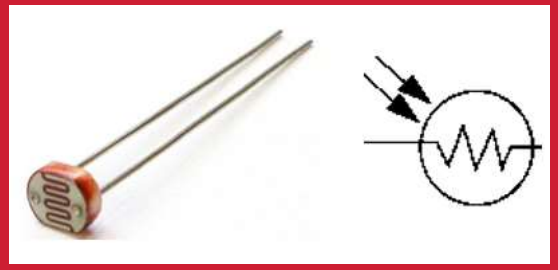
AAnn_AnalogRead_fmap §

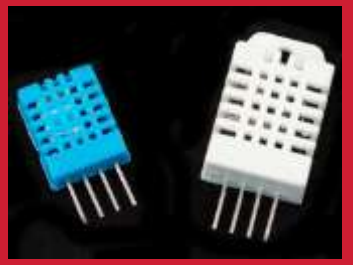
```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// 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;
}
```



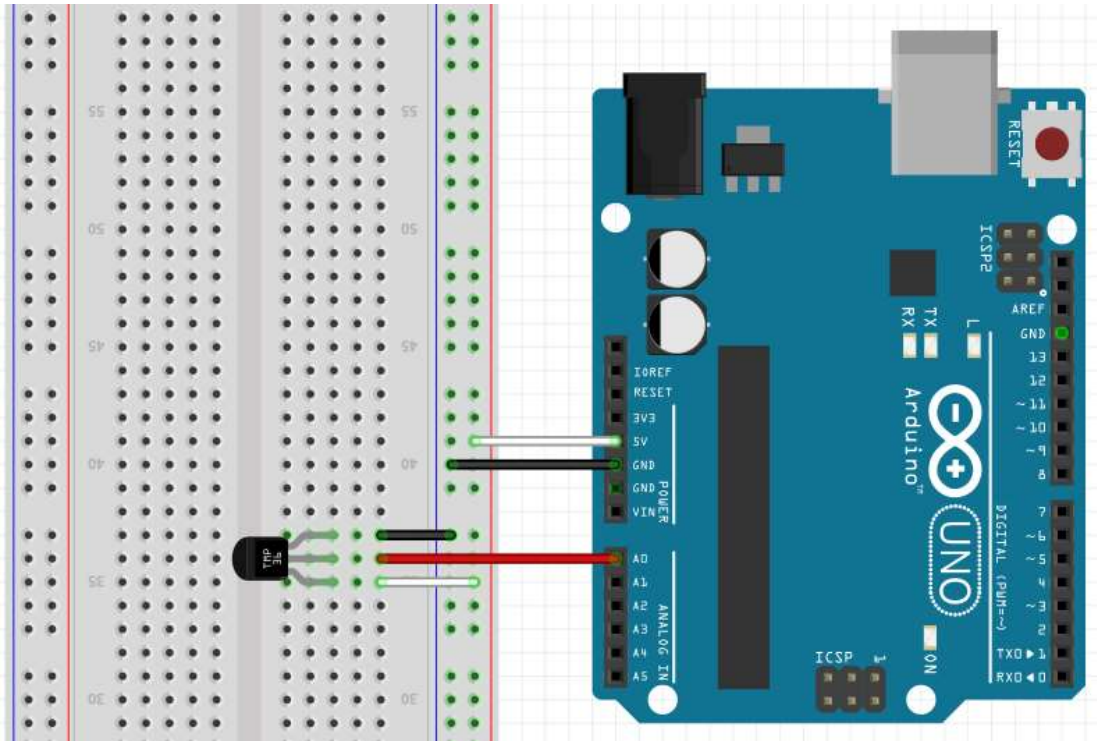
Arduino Sensors

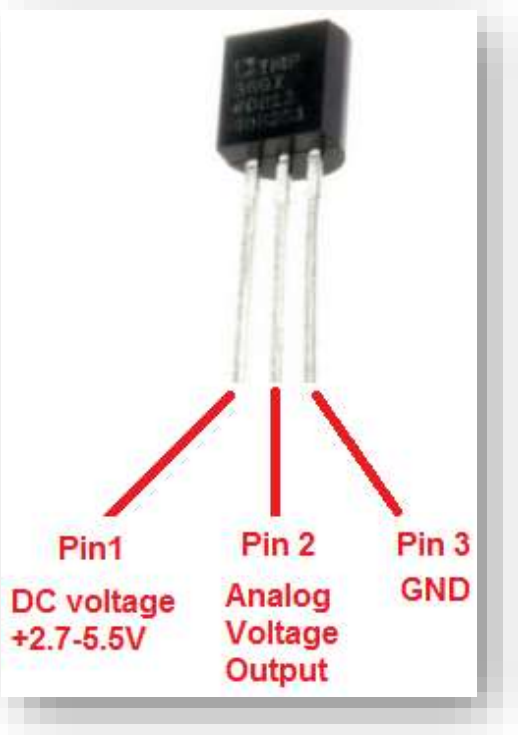




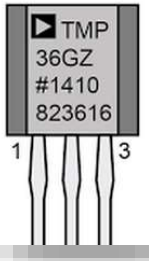


A3.1.1 Temperature sensor [TMP36]





Parts : TMP36



- **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

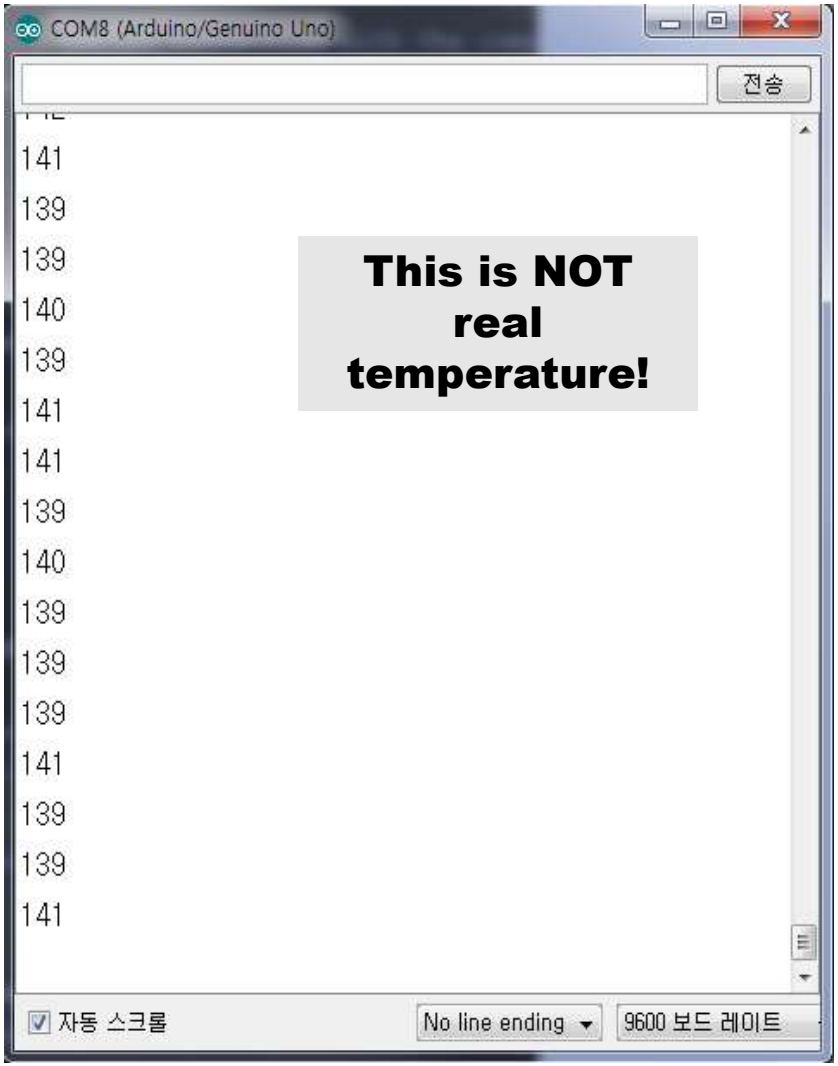
A3.1.2 Temperature sensor [TMP36]

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)



A3.1.3 Temperature sensor [TMP36]

Sensor property

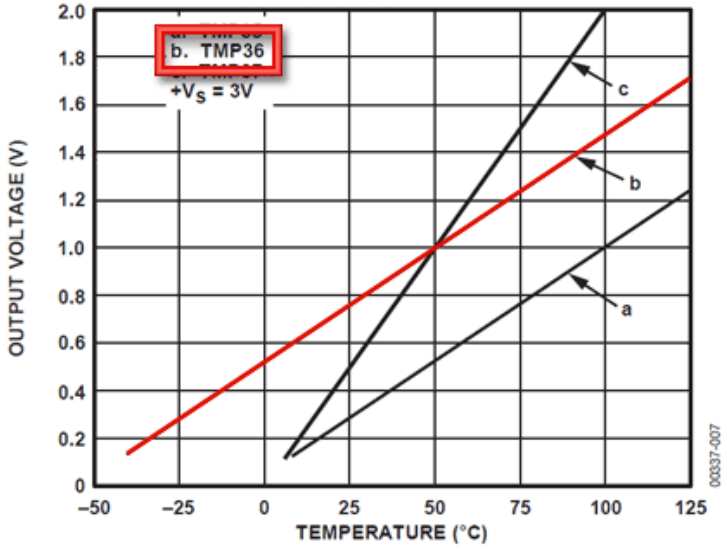


Figure 6. Output Voltage vs. Temperature

Output Voltage (mV) vs. Temperature (°C)			
V	0	500	1000
T	-50	0	50

https://github.com/Redwoods/Arduino/blob/master/ar-iot/py-ml/tmp36_LR.ipynb

Temperature conversion

Temp (° C) = (Vout – 500) / 10

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

(0 <= value <= 1023)

```
// converting that reading to voltage
float voltage = value * 5.0 * 1000;  // in mV
voltage /= 1023.0;
float temperatureC = (voltage - 500) / 10 ;
```

A3.1.4 Temperature sensor [TMP36]

Working code

TMP36

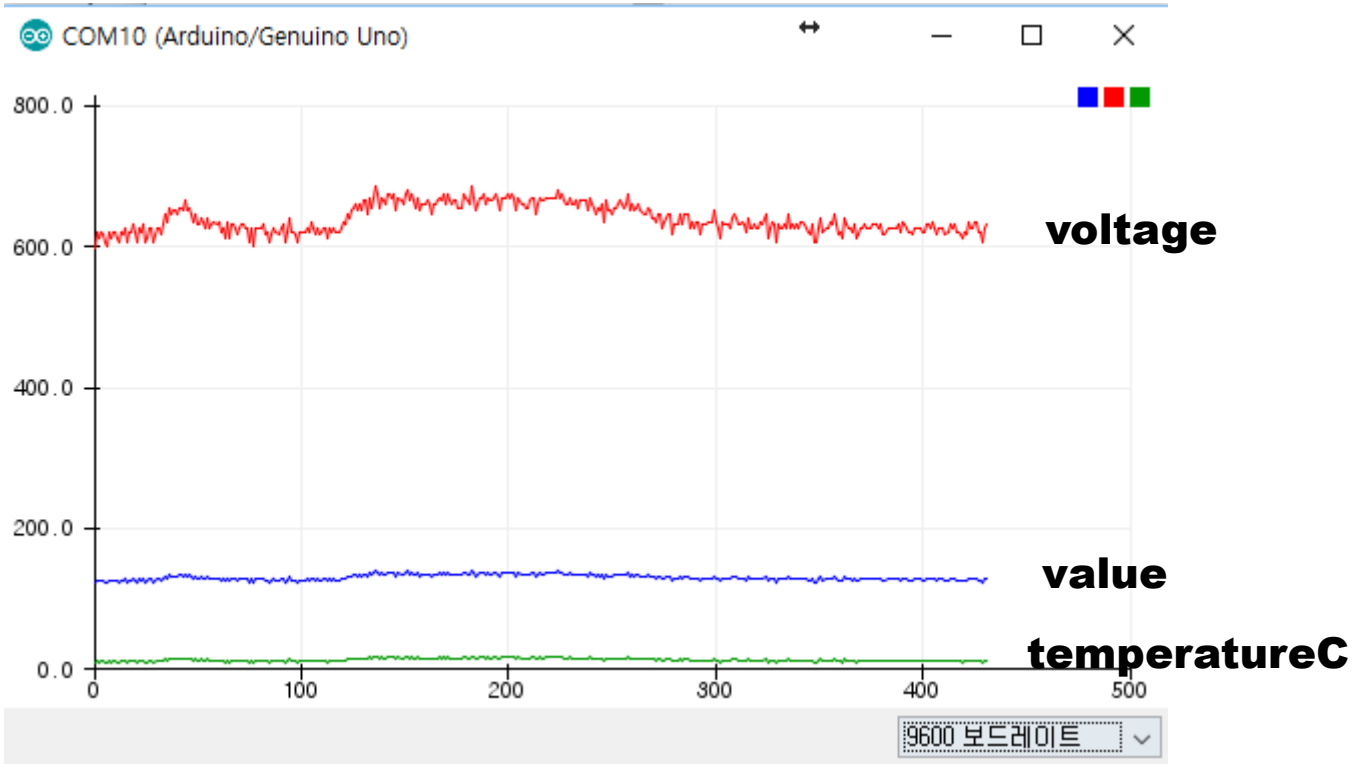
```
10 }
11
12 void loop() {
13    //getting the voltage reading from the temperature sensor
14   int value = analogRead(TEMP_INPUT);
15   Serial.print("AA00, value = ");
16   Serial.print(value);
17   Serial.print(" : ");
18
19    // converting that reading to voltage
20    float voltage = value * 5.0 * 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 (℃)

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 = 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 = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
```

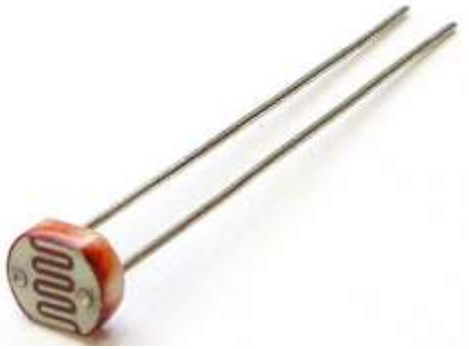
A3.1.5 Temperature sensor [TMP36]

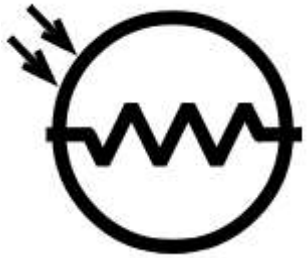


Save as
AAnn_TMP36.png

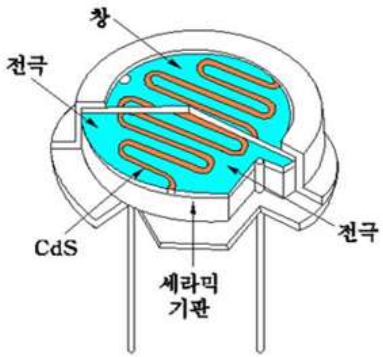
A3.2 Luminosity sensor [Photocell LDR]

CdS 센서- photoresistor





CDS특성

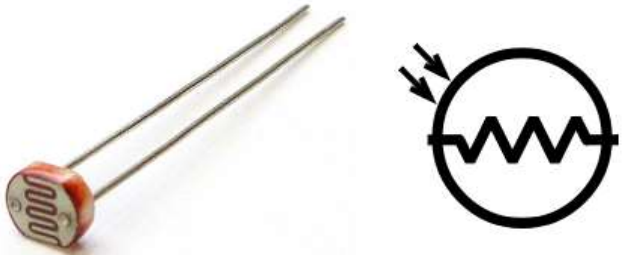


1. 감도
   - -빛의 파장에 따라 감도가 다름
2. 허용손실
   - -비교적 큰 전류를 흘릴 수 있음
3. 암 전류
   - -빛이 없어도 약간의 전류가 흐름
4. 명 전류
   - - 빛을 비추면 흐르는 전류
5. 응답특성
   - - 응답 시간 지연
   - - 빛의 세기에 따라 응답시간 다름
6. 가변저항
   - -빛에 따른 가변저항

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

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

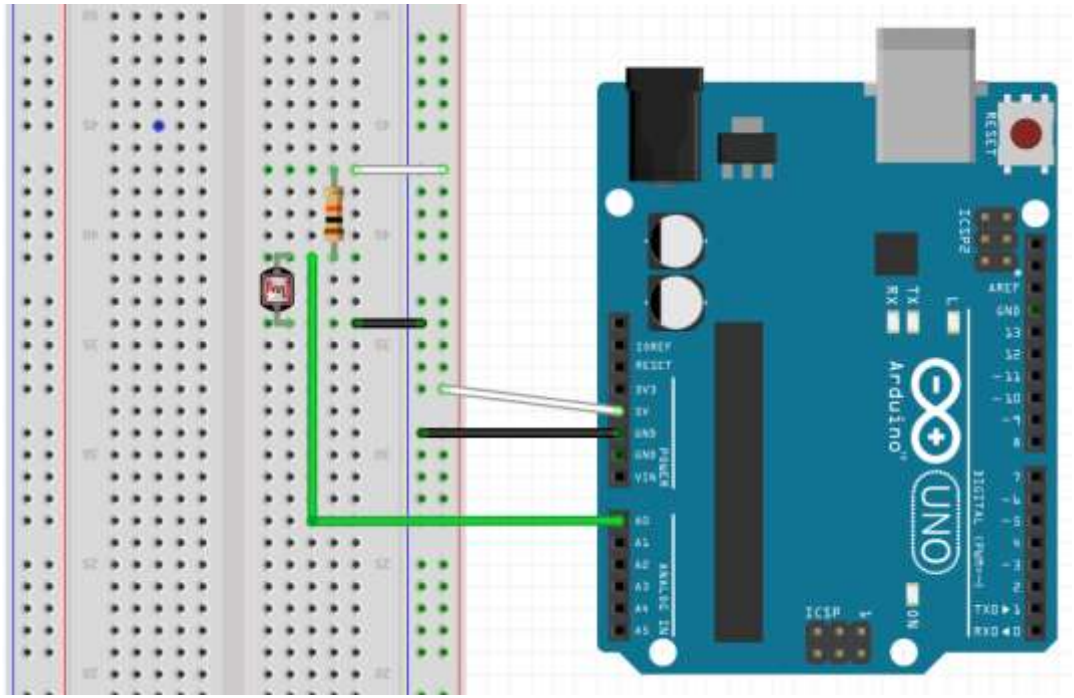
럭스의 예 [편집]

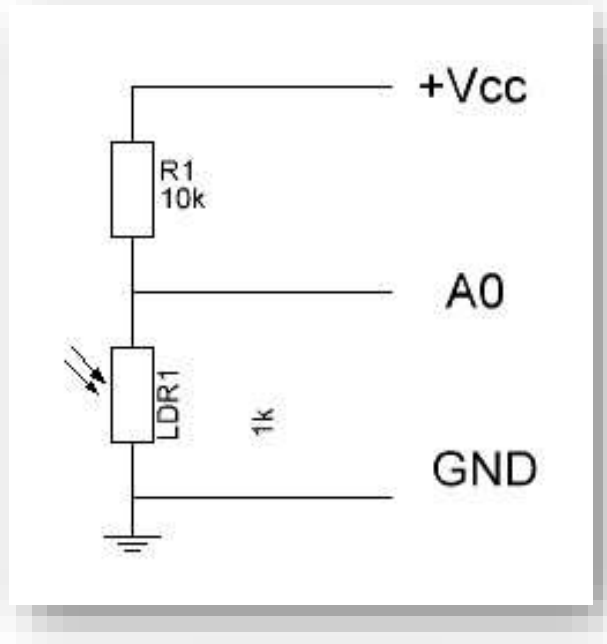
밝기차	예
10^{-5} lux	가장 밝은 별(시리우스)의 빛[1]
10^{-4} lux	하늘을 덮은 완전한 별빛[1]
0.002 lux	대기광이 있는 달 없는 맑은 밤 하늘[1]
0.01 lux	초승달
0.27 lux	맑은 밤의 보름달[1][2]
1 lux	열대 위도를 덮은 보름달[3]
3.4 lux	맑은 하늘 아래의 어두운 황혼[4]
50 lux	거실[5]
80 lux	복도/화장실[6]
100 lux	매우 어두운 낮[1]
320 lux	권장 오피스 조명 (오스트레일리아)[7]
400 lux	맑은 날의 해돋이 또는 해넘이
1000 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 [sketch-1]

▶ 스케치 구성

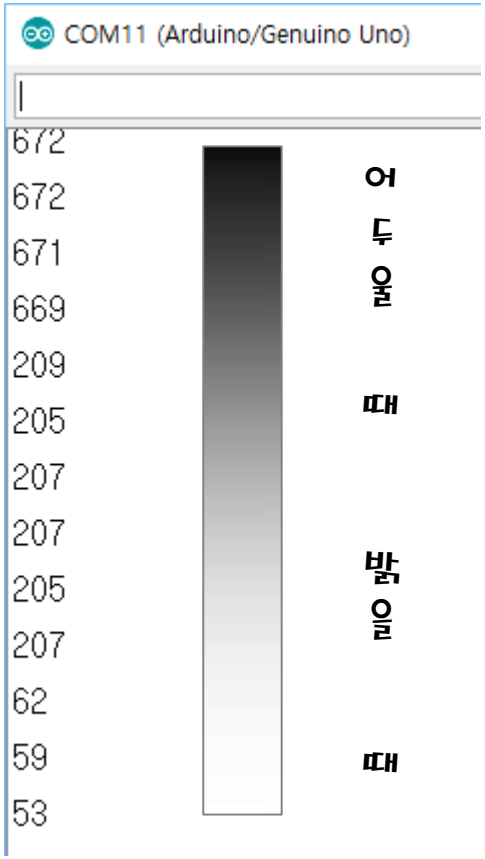
1. A0 핀을 CdS 조도 센서의 입력으로 설정한다.
2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
3. loop()에서 **analogRead()** 함수로 A0 핀에서 측정되는 값을 읽어 들인다.

A3.2.4 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 1.

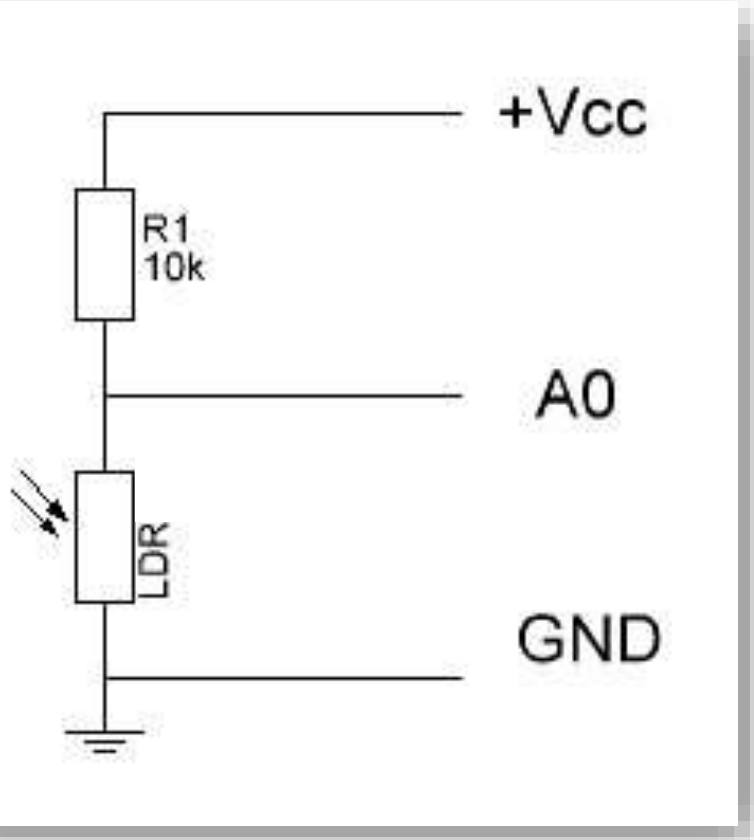
CdS_start

```
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
```



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.

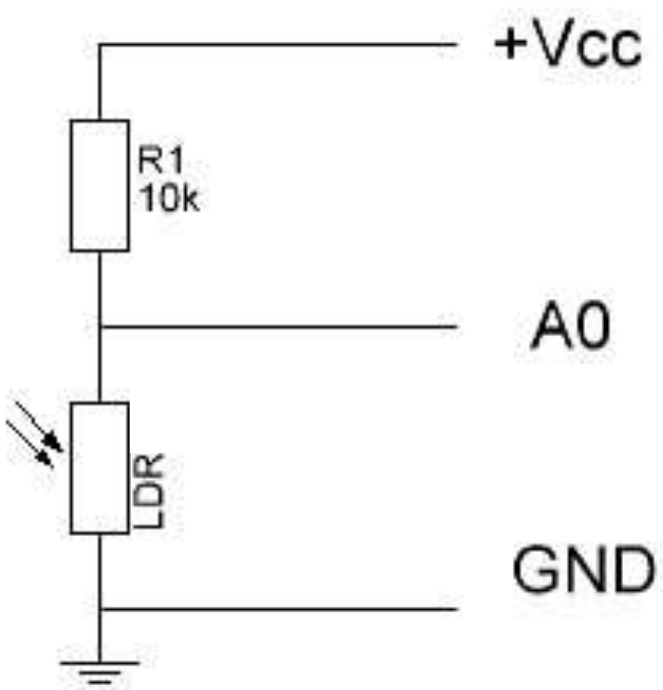
If you would us the LDR as the lower part of a voltage divider, then in darkness there would be a high voltage over the LDR, while in bright light, there would be a low voltage over that resistor.

어두우면 측정 값이 작아지고 밝을수록 값이 커져야 된다.
그리고 측정 값은 **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)



$$(a)\ V_{out} = \frac{R_{ldr}}{(R_1 + R_{ldr})} * V_{CC}\ ,$$

$$(b)\ R_{ldr} = \frac{10 * V_{out}}{(5 - V_{out})}\ (k\Omega)\ ,$$

$$(c)\ V_{out} = value * V_{CC}/1023\ ,$$

$$(d)\ Lux = \frac{500}{R_{ldr}}\ ,$$

$$(e)\ Lux = (\frac{2500}{V_{out}} - 500)/10\ (lux).$$

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

A0에서 측정되는 **LDR** 양단의 전압 = **Vout**

A3.2.5 Luminosity sensor [sketch-2]

▶ 스케치 구성

1. A0 핀을 CdS 조도 센서의 입력으로 설정한다.
2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
3. loop()에서 **analogRead()** 함수로 A0 핀에서 측정되는 값을 읽어 들인다.
4. A0 측정값 (0 ~ 1023)을 전압 (0 ~ 5 V)으로 환산한다.
5. 전압 (V)을 온도 (°C)로 환산한 후, A0 측정값, 환산 전압, 환산 조도를 한 줄로 1 초마다 컴퓨터로 전송한다.

A3.2.6 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 2.

sketch08_CdS2

```
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 Lux
14 double luminosity (int RawADC0){
15   double Vout=RawADC0*5.0/1023;  // 5/1023 (Vin = 5 V)
16   double lux=(2500/Vout-500)/10;
17   // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
18   return lux;
19 }
```

COM11 (Arduino/Genuino Uno)

```
23
24
23
183
191
183
185
531
1169
1262
1198
1039
1063
```

어두울 때

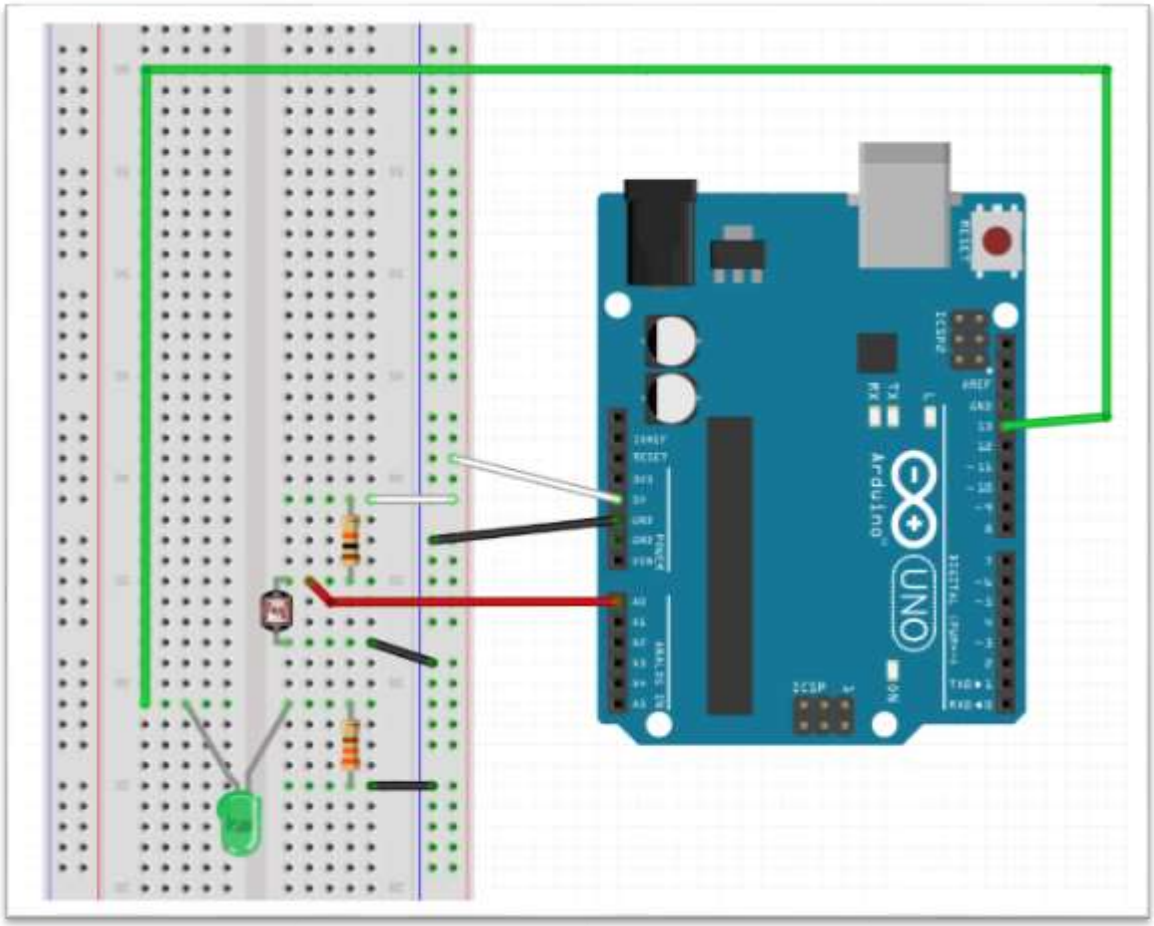
밝을 때

밝을수록 측정 값이 커지고
어두을수록 값이 작아진다 !!!

[DIY] 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.



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.

DIY Code

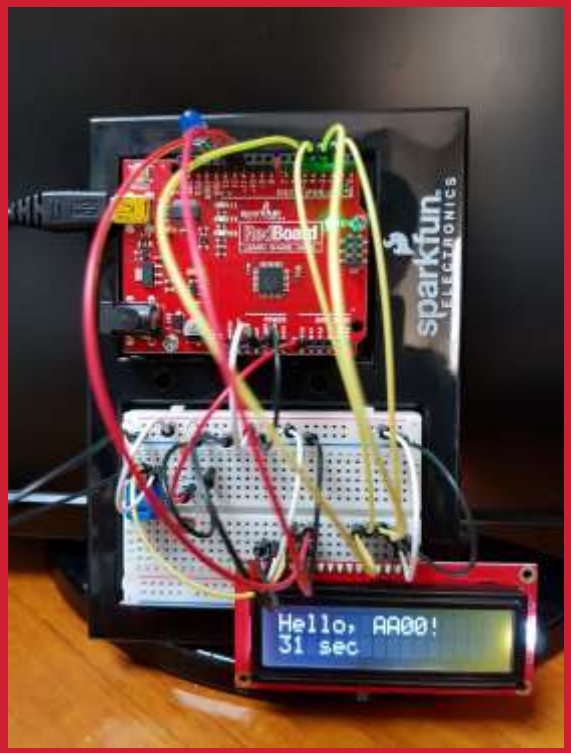
Cds_LED

```
1  //  lux
2  #define CDS_INPUT 0
3  // LED pin
4  const int ledPin = 13;
5
6  int threshold = 70;
7
8  void setup() {
9    pinMode(ledPin, OUTPUT);
10   Serial.begin(9600);
11 }
```

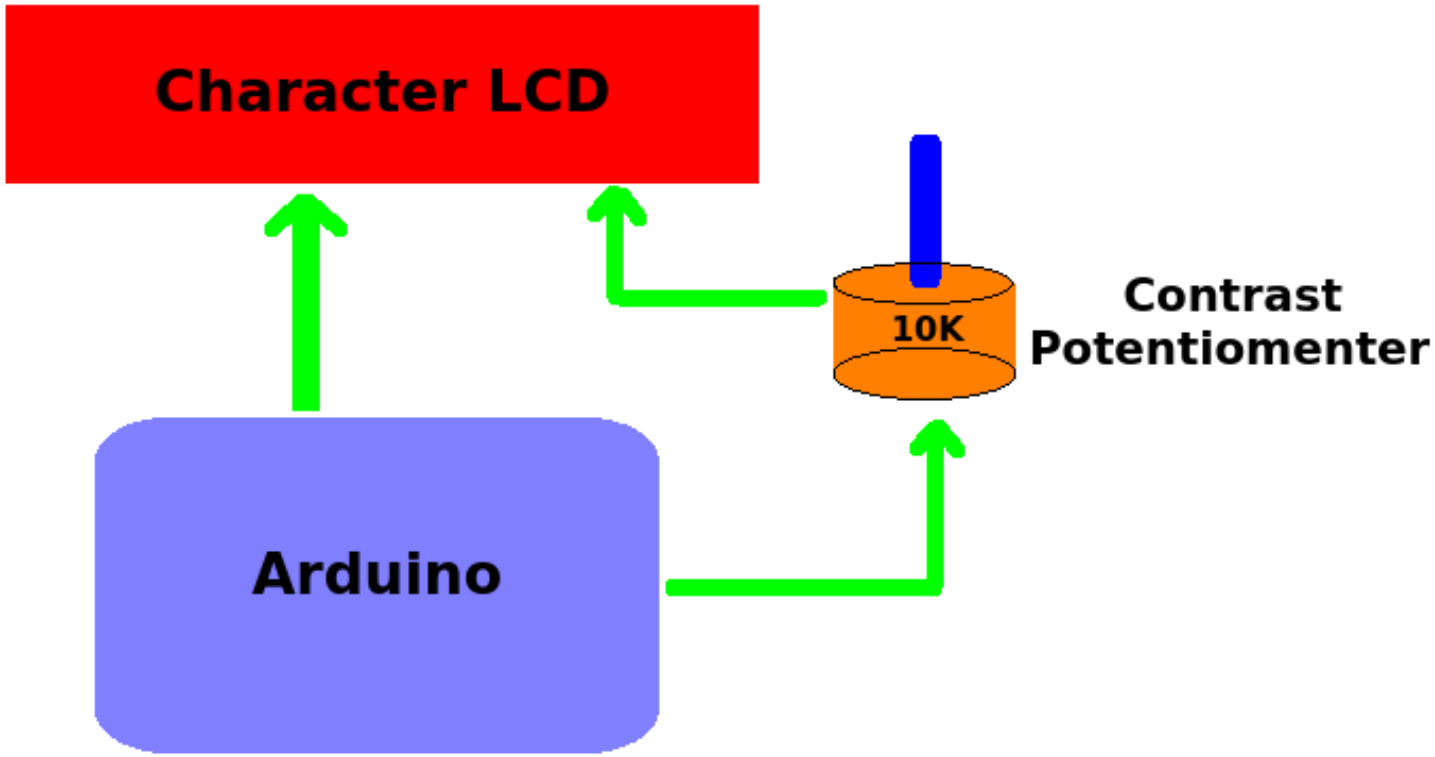
```
13 void loop() {
14   int value = analogRead(CDS_INPUT);
15   int lux = int(luminosity(value))
16   Serial.println(lux);
17
18   // If lux is lower than a threshold, LED is set ON.
19   if(lux >= threshold)
20     digitalWrite(ledPin, LOW);
21   else
22     digitalWrite(ledPin, HIGH);
23
24   delay(1000);
25 }
26 //Voltage to Lux
27 double luminosity (int RawADC0){
28   double Vout=RawADC0*5.0/1023;  // 5/1023 (Vin = 5 V)
29   double lux=(2500/Vout-500)/10;
30   // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
31   return lux;
32 }
```

AAnn_CdS_LED.ino

Signal Monitoring via LCD



Introduction to LCD

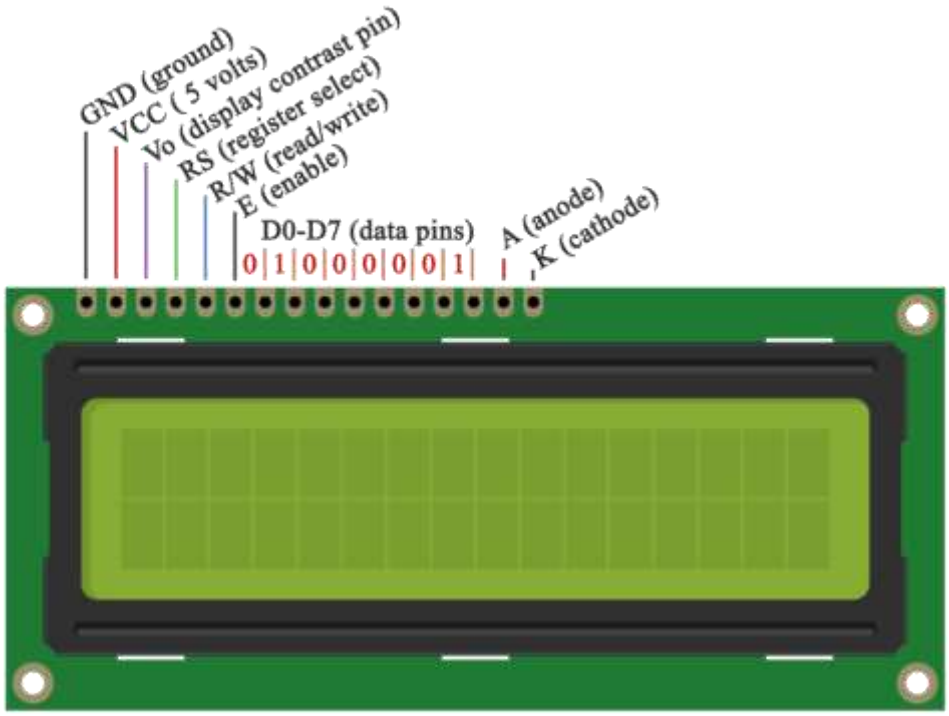


Liquid crystal display

1. 입출력 핀을 이용하여 LCD 모듈에 표시하기
2. I²C를 이용한 LCD 출력

1. Introduction to LCD

LCD (Liquid Crystal Display, 16 X 2)



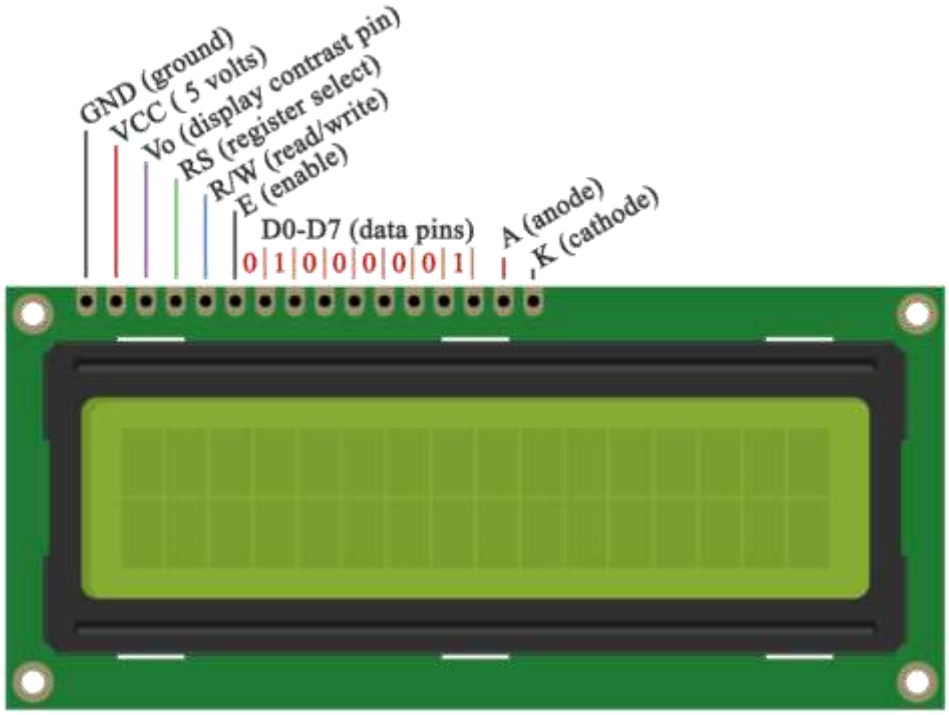
(1,2, ... 15,16)

1. GND
2. VCC (+5V)
3. Vo (contrast, 가변저항기 연결)
4. RS
5. R/W
6. E

- D0 ~ D7 (data, 7~14)
- A (15, Backlight+, 220 or 330 Ω)
- K (16, Backlight-)

Introduction to LCD

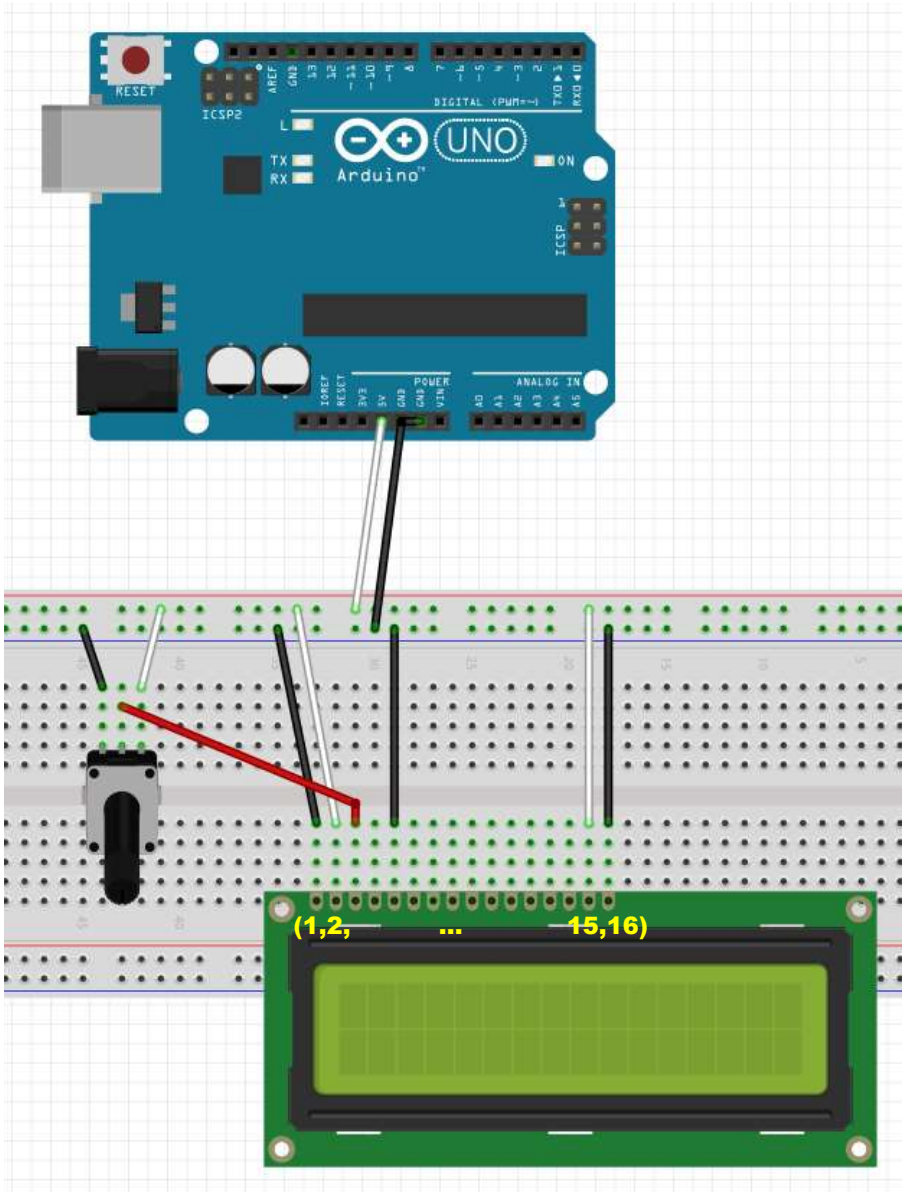
LCD (Liquid Crystal Display, 16 X 2)



(1,2, ... 15,16)

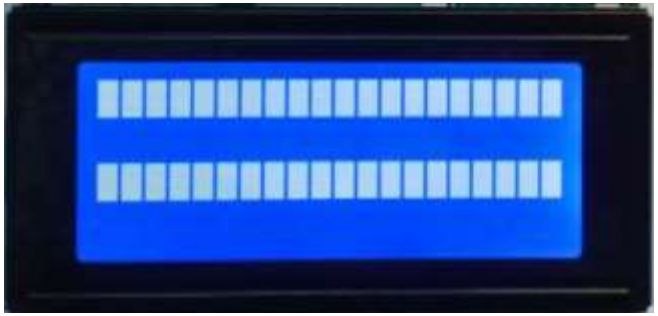
Pin 1 to Arduino GND
Pin 2 to Arduino +5V
Pin 3 to wiper
Pin 4 to Arduino pin D12
Pin 5 to Arduino GND
Pin 6 to Arduino pin D11
Pin 11 to Arduino pin D5
Pin 12 to Arduino pin D4
Pin 13 to Arduino pin D3
Pin 14 to Arduino pin D2
Pin 15 to +5V (with 220 or 330 Ω)
Pin 16 to GND

LCD 초기화 (pin-1, 2, 3, 5, 15,16)

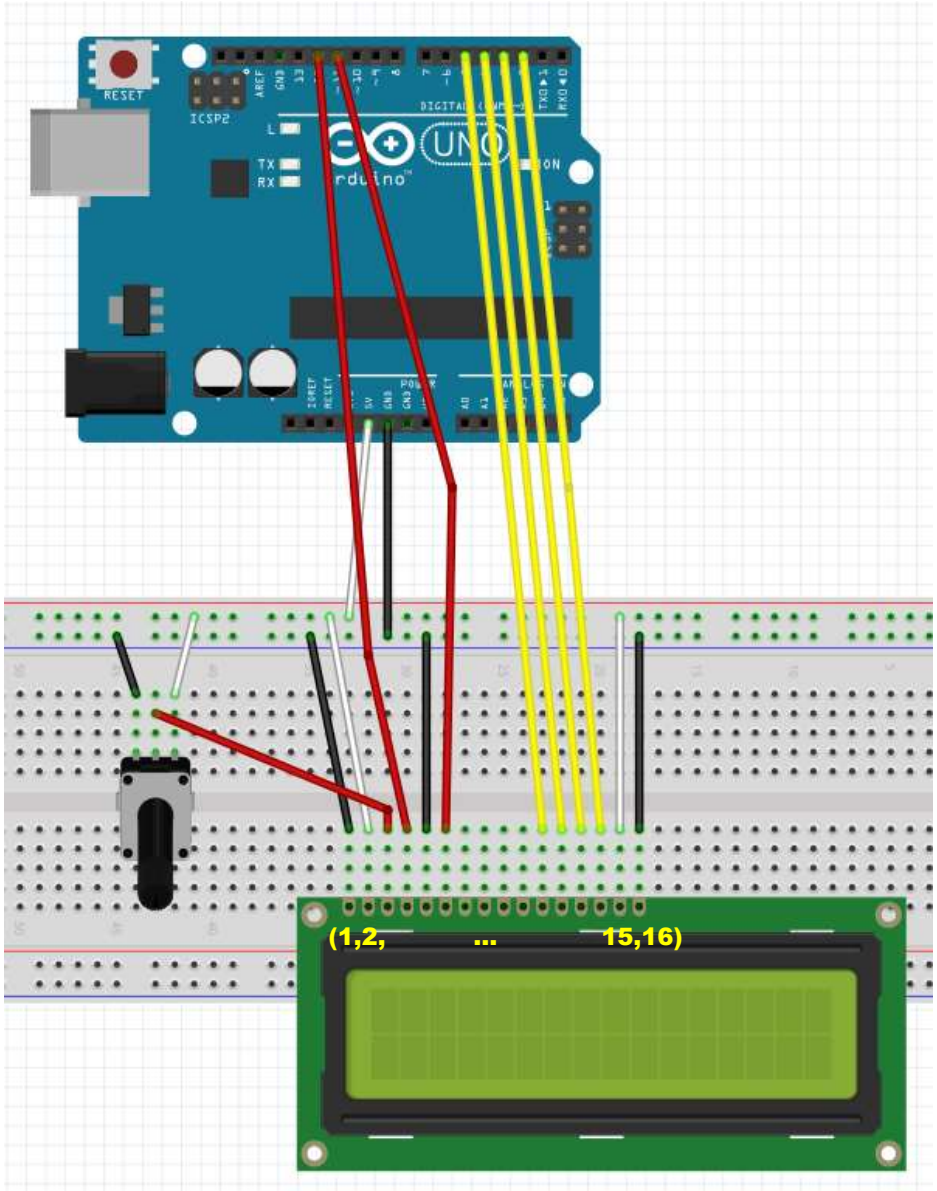


Pin 1 to Arduino GND
Pin 2 to Arduino +5V
Pin 3 to wiper (potentiometer)
Pin 5 to Arduino GND
Pin 15 to +5V
Pin 16 to GND

전원 연결 후
LCD 초기화



데이터 입력 초기화 (pin-4, 6, 11,12,13,14)



Pin 1 to Arduino GND

Pin 2 to Arduino 5V

Pin 3 to wiper

Pin 4 to Arduino pin D12

Pin 5 to Arduino GND

Pin 6 to Arduino pin D11

Pin 11 to Arduino pin D5

Pin 12 to Arduino pin D4

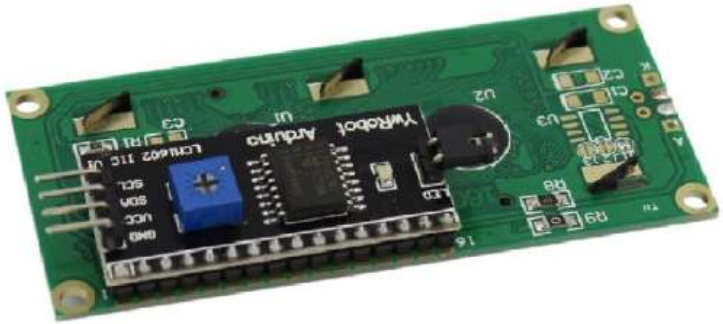
Pin 13 to Arduino pin D3

Pin 14 to Arduino pin D2

Pin 15 to +5V

Pin 16 to GND

2. I²C를 이용한 LCD 출력



I²C(아이스퀘어드시, **Inter-Integrated Circuit**)는 필립스에서 개발한 직렬 버스이다. 마더보드, 임베디드 시스템, 휴대 전화 등에 저속의 주변 기기를 연결하기 위해 사용된다.

I²C 는 풀업 저항이 연결된 직렬 데이터(**SDA**)와 직렬 클럭(**SCL**)이라는 두 개의 양 방향 오픈 컬렉터 라인을 사용한다. 최대 전압은 **+5 V** 이며, 일반적으로 **+3.3 V** 시스템이 사용되지만 다른 전압도 가능하다.

https://ko.wikipedia.org/wiki/I%C2%B2C

http://www.ifuturetech.org/product/16x2-lcd-i2c-lcd/

I²C를 이용한 LCD 출력

I²C (Inter Integrated Circuit)

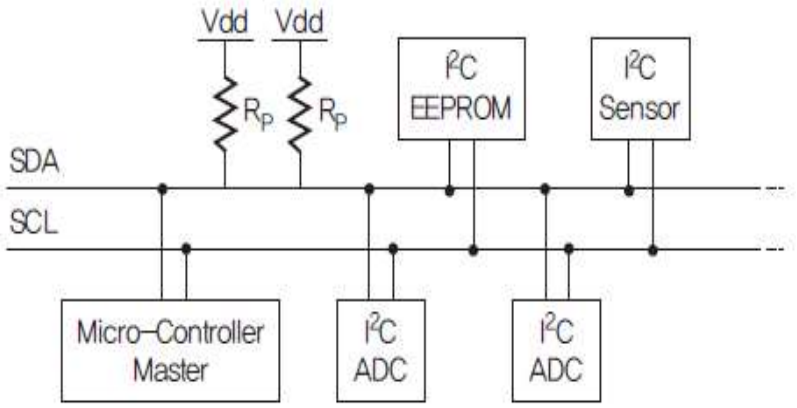


그림 3.2 I²C를 이용한 네트워크

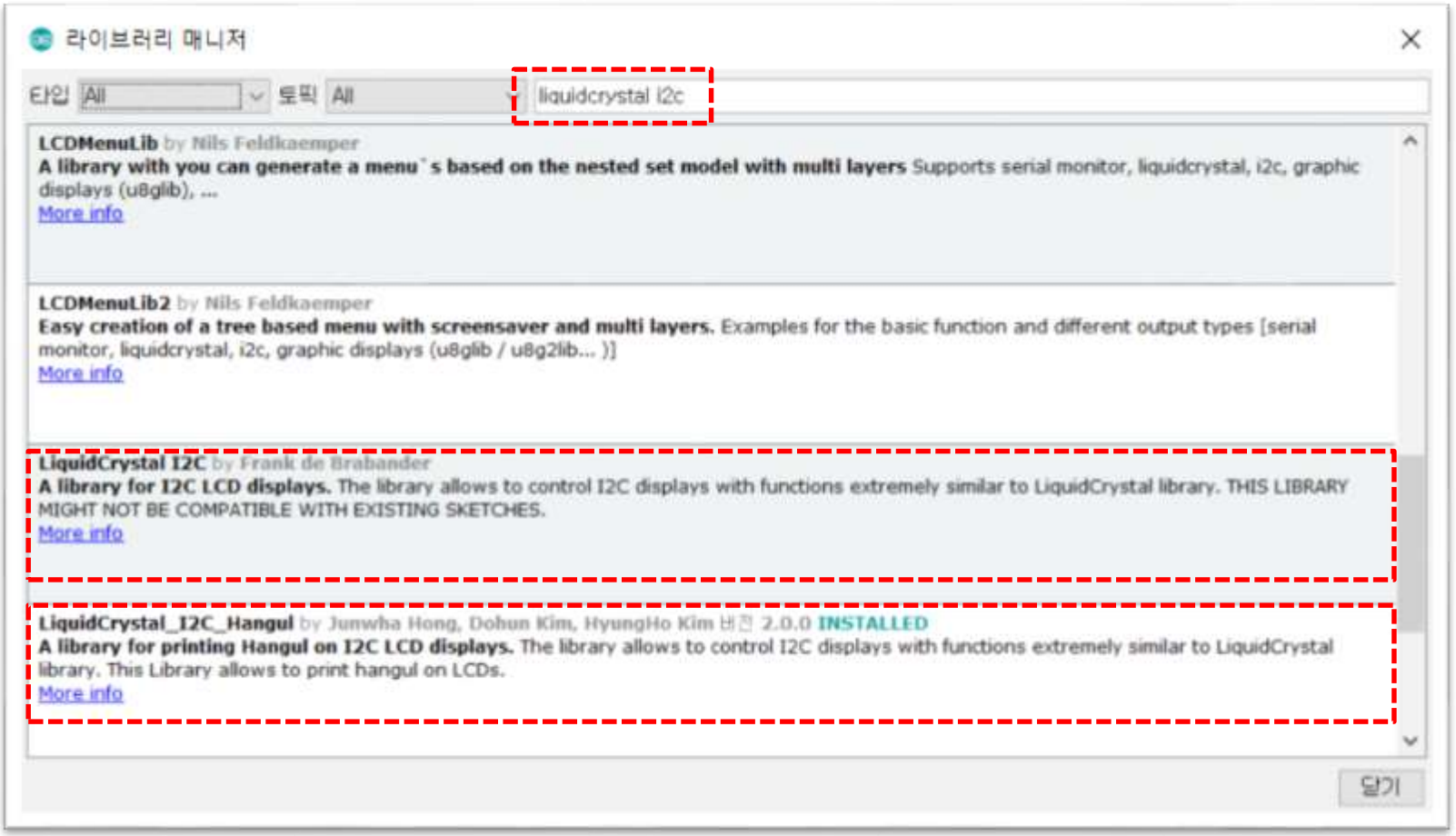
- ✓ Phillips사에서 개발된 규격이며 TWI라고도 함.
- ✓ SDA(Serial Data line), SCL(Serial Clock Line) 두 선으로 통신
- ✓ Master와 Slave로 구분되어 Master에서 통신을 주관
- ✓ 최대 112개의 노드를 연결 가능하고 최고 3.4Mbps의 속도

- ✓ LCD 모듈을 I²C 통신으로 제어하기 위해선 PCF8574 IC를 사용
- ✓ SDA, SCL 두 개의 입출력 핀만 필요

I²C를 이용한 LCD 출력 - 라이브러리 설치

라이브러리 매니저를 이용하여 I²C LCD용 라이브러리(LiquidCrystal I2C)를 설치

스케치 > 라이브러리 포함하기 > 라이브러리 관리



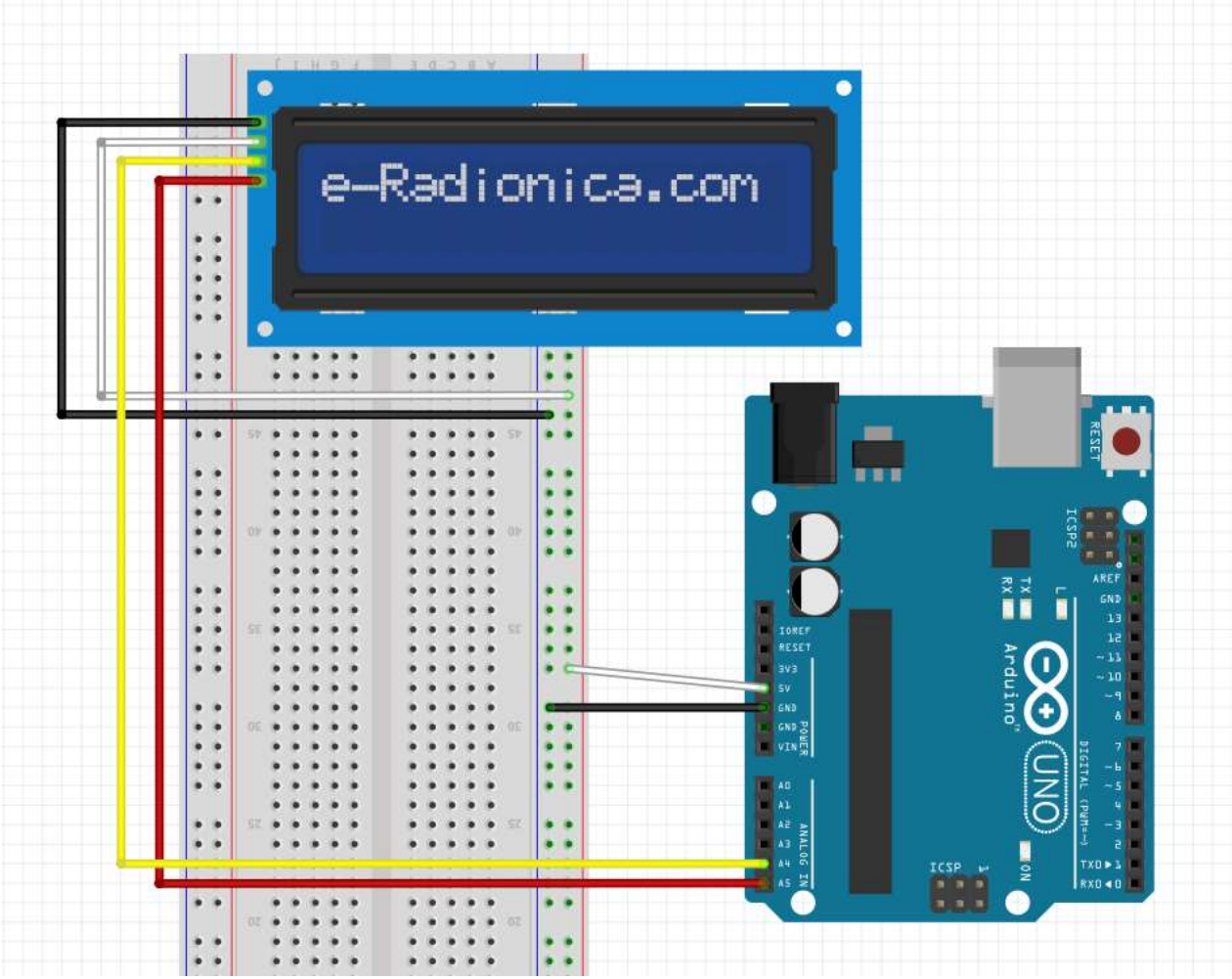
https://github.com/junwha0511/LiquidCrystal_I2C_Hangul

I²C를 이용한 LCD 출력 회로

Hardware

1. I²C LCD 모듈과 Arduino는 전원핀 Vcc, GND와 I²C 통신핀 SDA, SCL이 연결되어야 한다.
2. I²C LCD 모듈의 Vcc와 GND를 Arduino의 5V와 GND에 연결한다.
3. SDA는 A4에, SCL은 A5에 연결한다.

I²C를 이용한 LCD 출력 회로



I²C를 이용한 LCD 출력

Commands

- LiquidCrytral_I2C(I2C 주소, 가로 글자수, 세로 글자수)

  LCD 모듈이 연결된 I2C 주소와 LCD의 가로, 세로 글자수를 설정한다.

- lcd.init( ); LCD 모듈을 설정한다.
- lcd.clear( ): lcd란 이름의 LCD 모듈의 화면의 모든 표시를 지우고 커서를 왼쪽 위로 옮긴다.
- lcd.home( ): lcd란 이름의 LCD 모듈의 커서를 왼쪽 위로 옮긴다.
- lcd.setCursor(행, 열): lcd란 이름의 LCD 모듈의 커서를 원하는 위치로 이동시킨다.
- lcd.print(데이터): lcd란 이름의 LCD 모듈에 데이터를 출력한다.
- lcd.noBacklight(): lcd란 이름의 LCD 모듈의 백라이트를 소등한다.
- lcd.backlight(); lcd란 이름의 LCD 모듈의 백라이트를 점등한다.

I²C를 이용한 LCD 출력

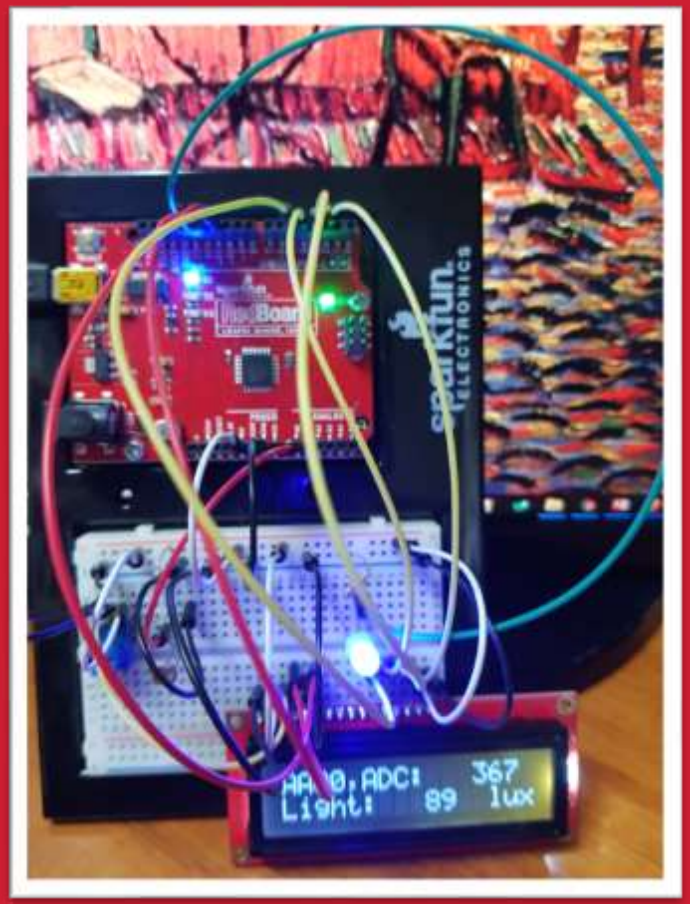
Take a photo of LCD screen.

Save photo as
AAnn_LCD_hello.png

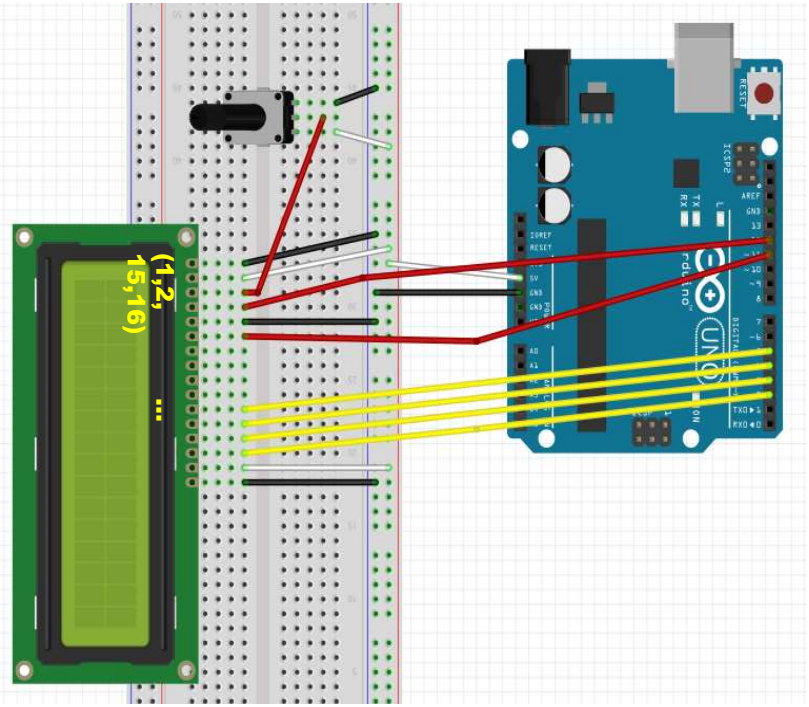
Save code: AAnn_LCD.ino

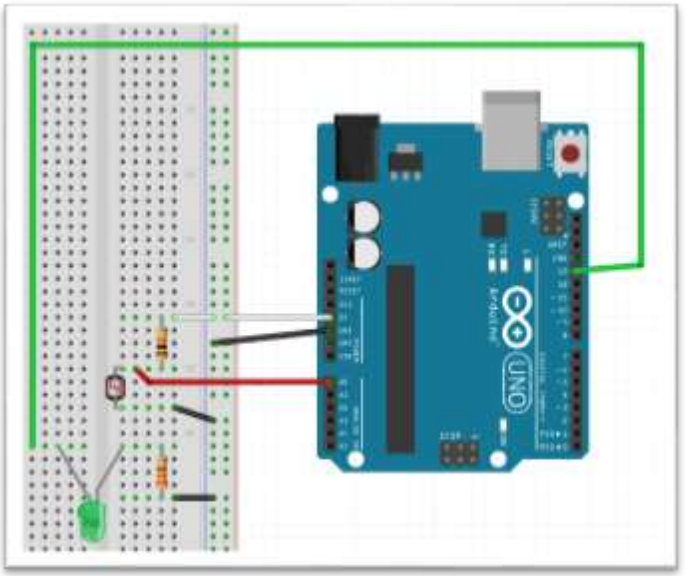
CdS LCD Project

LCD에 조도 값을
표시하면서
조도에 따라 LED를
ON/OFF



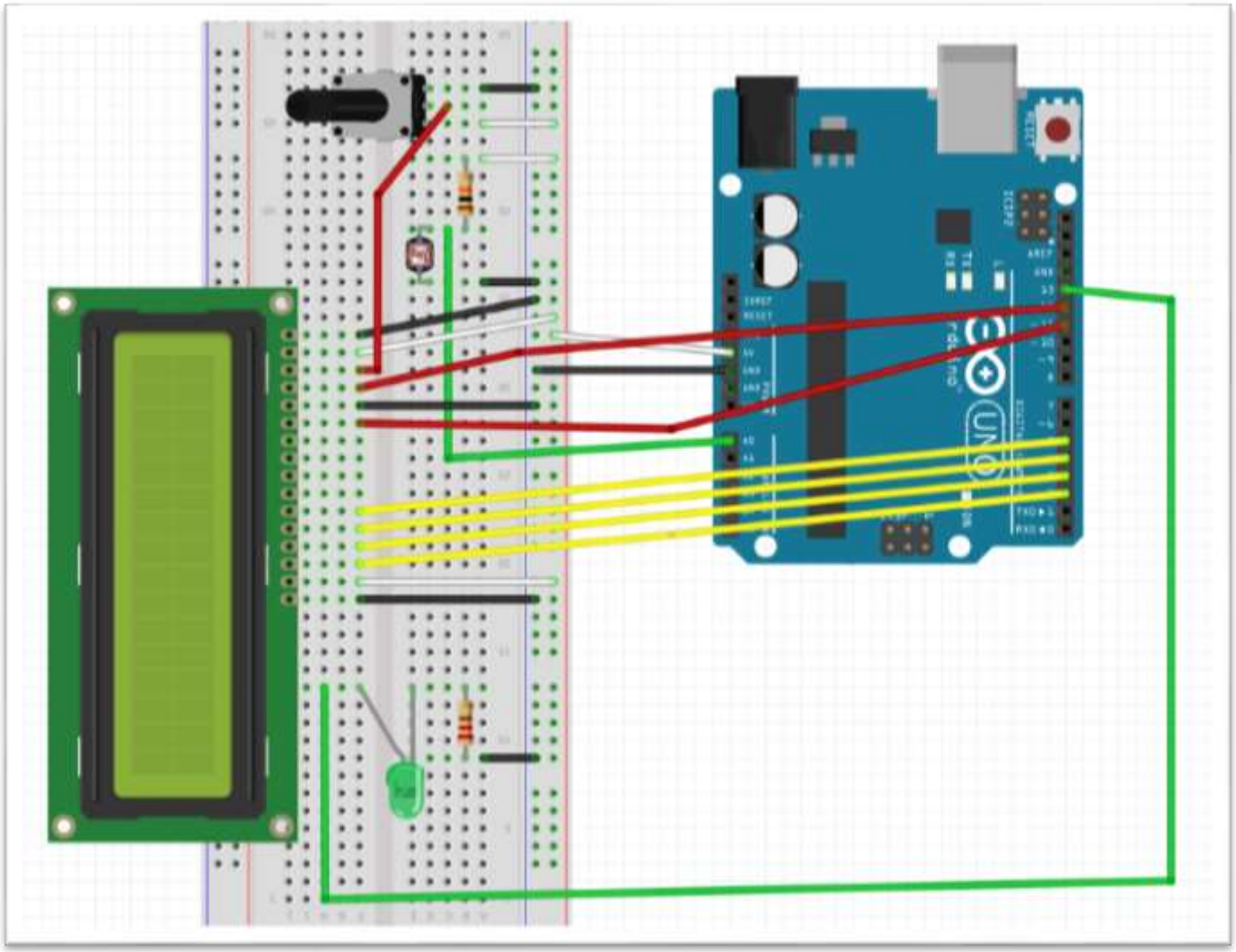
CdS-LCD project





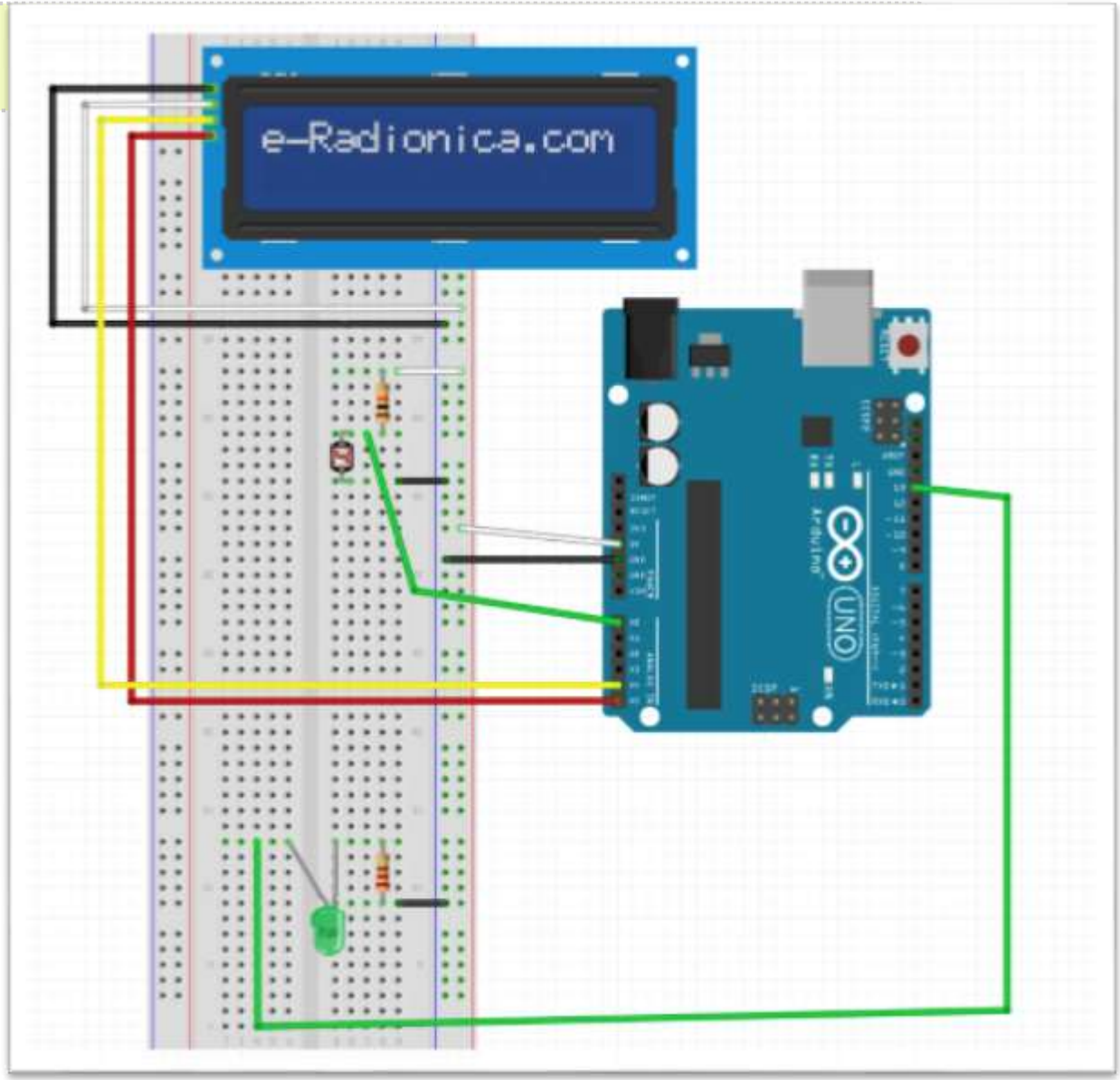
CdS-LCD project : fzz circuit

CdS_LCD_LED.fzz



CdS-LCD project : fzz circuit

CdS_LCD_I2C_LED.fzz



CdS-LCD project

Set CdS-LCD project

Project

CdS 셀을 이용하여 조도를 측정해 보자.

1. CdS 셀로 측정된 조도를 아날로그 핀을 통하여 0~1023 범위로 읽는다.
2. ADC 값을 LCD 모듈로 lux로 출력한다. (빛의 밝기)
3. lux 값에 따라 D13에 연결된 단색 LED의 ON/OFF를 조정한다.

Hardware

1. LCD를 연결한다.
2. CdS셀과 10kΩ 저항을 연결한 뒤 저항의 한쪽 끝은 5V에 CdS셀의 한쪽 끝은 GND에 연결한다.
3. 저항과 CdS셀 사이를 아날로그 입력핀 A0에 연결한다.
4. 단색 LED를 330 Ω 저항을 연결해서 디지털 입력핀 D13과 GND에 연결한다.

CdS-LCD project : new code

CdS 센서 LCD 회로 - code: AAnn_LCD_lux.ino

AAnn_LCD_lux_start §

```
/*
 빛 입력 LCD 모니터링 및 제어
*/
 // LCD 라이브러리 설정
#include <LiquidCrystal_I2C.h>
#include<Wire.h>
// LCD 설정
LiquidCrystal_I2C lcd(0x27,16,2); // 0x3F
// 0번 아날로그핀을 CdS 셀 입력으로 설정한다.
const int CdSPin = 0;     // CdS => A0
const int ledPin = 13;    // LED pin => D13

  // LED OFF above threshold lux

void setup() {
  pinMode(ledPin, OUTPUT);
// 16X2 LCD 모듈 설정하고 백라이트를 켠다.
  lcd.init();
  lcd.backlight();
// 모든 메세지를 삭제한 뒤
// 숫자를 제외한 부분들을 미리 출력시킨다.
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("AA00,ADC: ");
  lcd.setCursor(0,1);
  lcd.print("Light:  ");
  lcd.setCursor(13,1);
  lcd.print("lux");  //
}
```

```
void loop(){
  int adcValue; // 실제 센서로부터 읽은 값 (0~1023)
  int illuminance; // 현재의 밝기. 0~100%
  int lux;         // 현재의 밝기. lux

  // CdS cell을 통하여 입력되는 전압을 읽는다.
  adcValue = analogRead(CdSPin);
  // luminosity() 함수를 이용해서 Lux 를 계산한다.
  lux = int(luminosity(adcValue));

  // 전에 표시했던 내용을 지운다.
  lcd.setCursor(12,0);
  lcd.print("    ");
  // ADC 값을 표시한다
  lcd.setCursor(12,0);
  lcd.print(adcValue);
  // 전에 표시했던 내용을 지운다.
  lcd.setCursor(9,1);
  lcd.print("   ");
  // 밝기를 표시한다
  lcd.setCursor(9,1);
  lcd.print(lux);

  // On/Off LED by threshold


  delay(1000);
}
```

LED ON/OFF

기능을 추가해서

Code를 완성 후,

AAnn_LCD_lux.ino

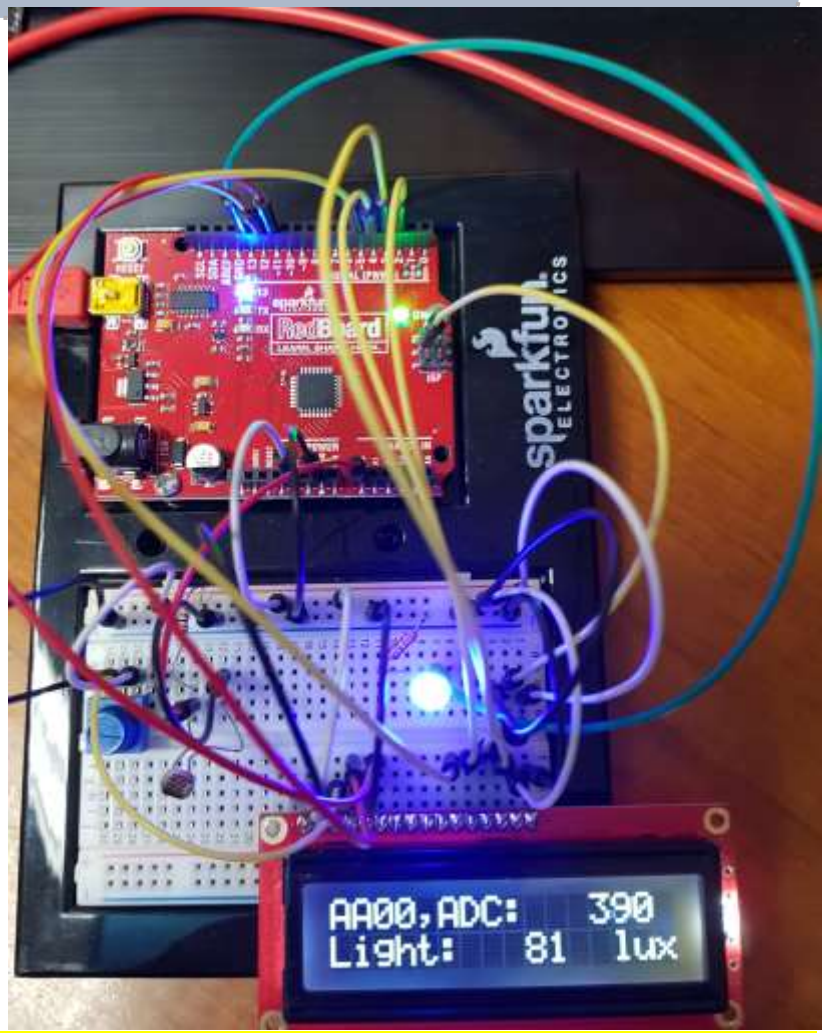
로 저장...

CdS-LCD project : result

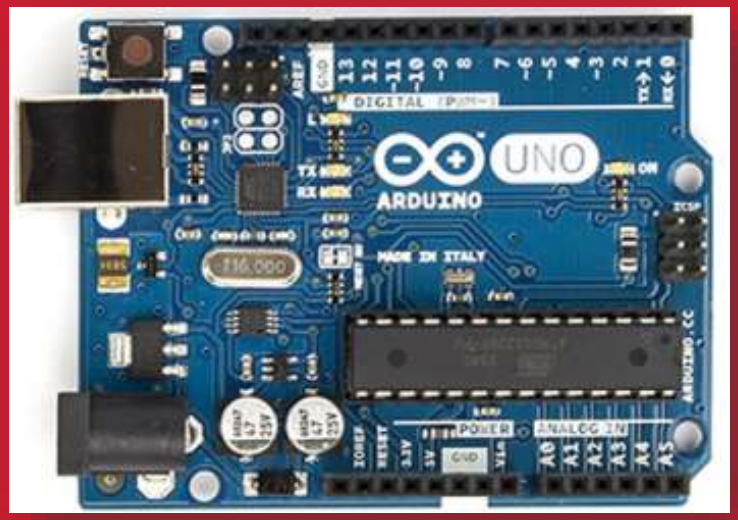
CdS 센서 LCD 회로 - 측정 결과

주변의 조도에 따라
어두우면 LED가
켜지고, 밝으면
LED가 꺼지도록
코드를 수정하시오.

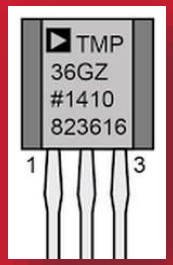
LED가 켜진
화면을 폰으로
촬영해서 그림을
제출하시오.

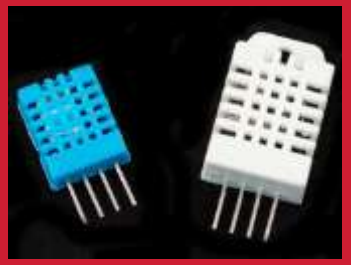


조도에 따라 LED가 ON/OFF 되는 것을 확인 받고
결과 화면 촬영: AAnn_LCD_lux.png 로 저장...



Arduino & Node.js





IOT: HSC

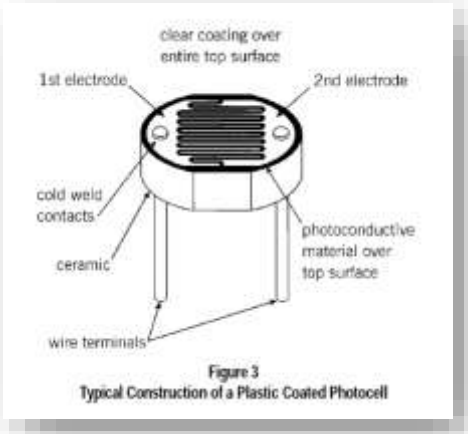
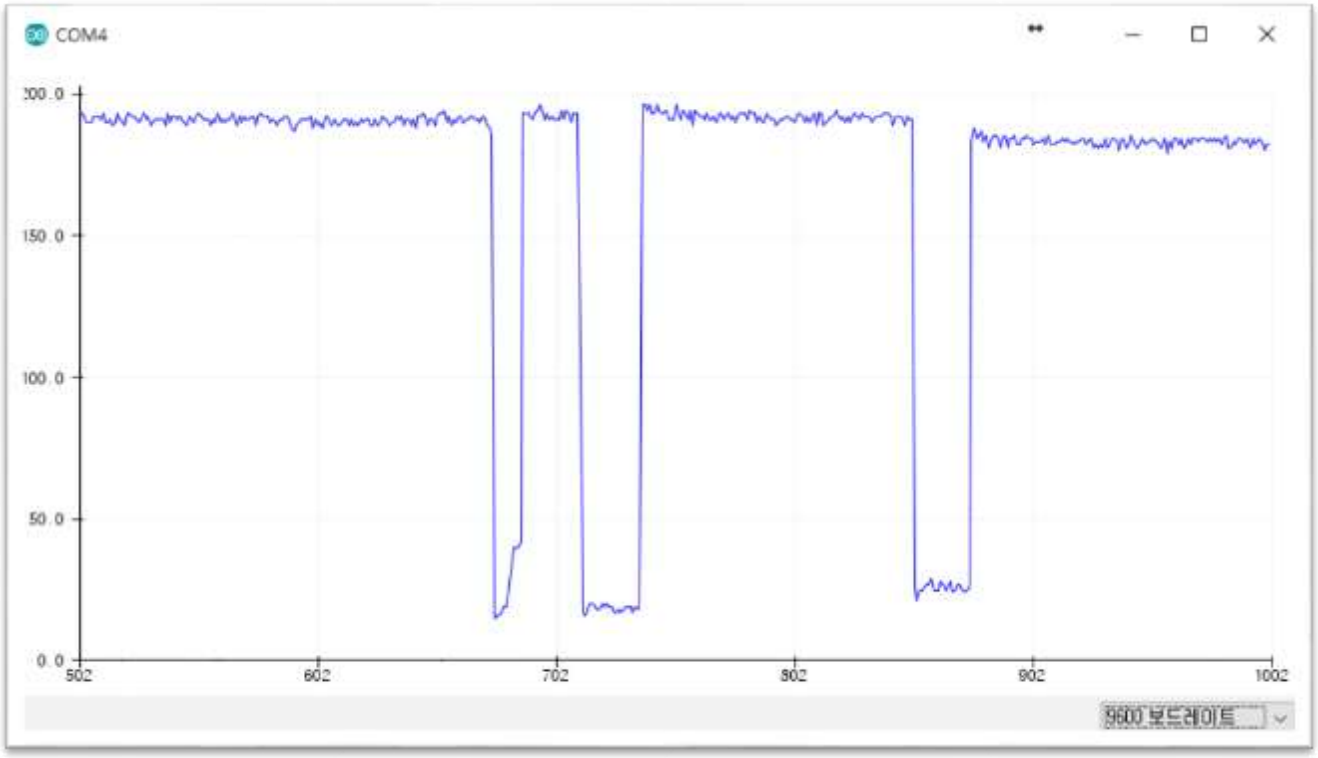
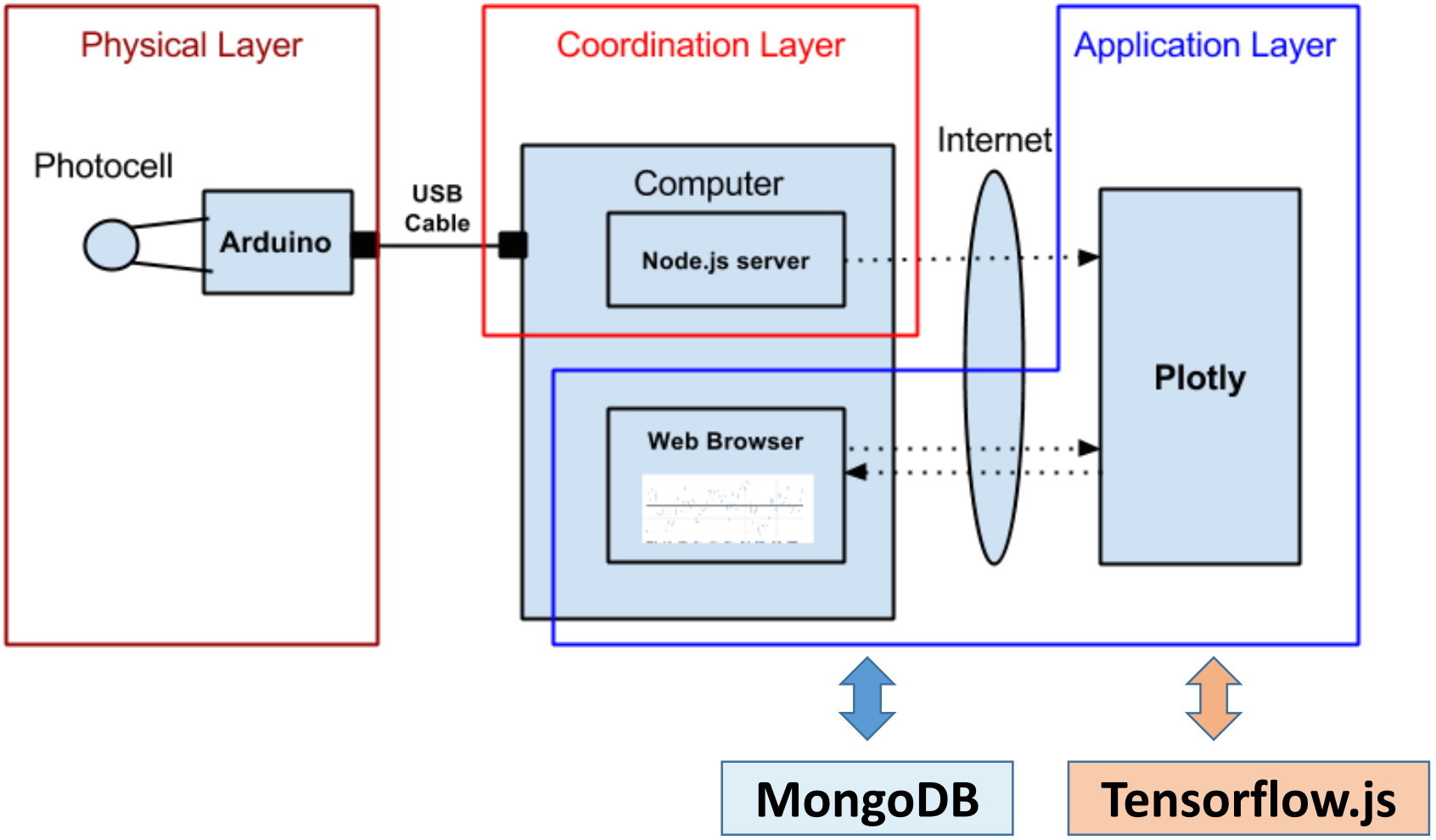


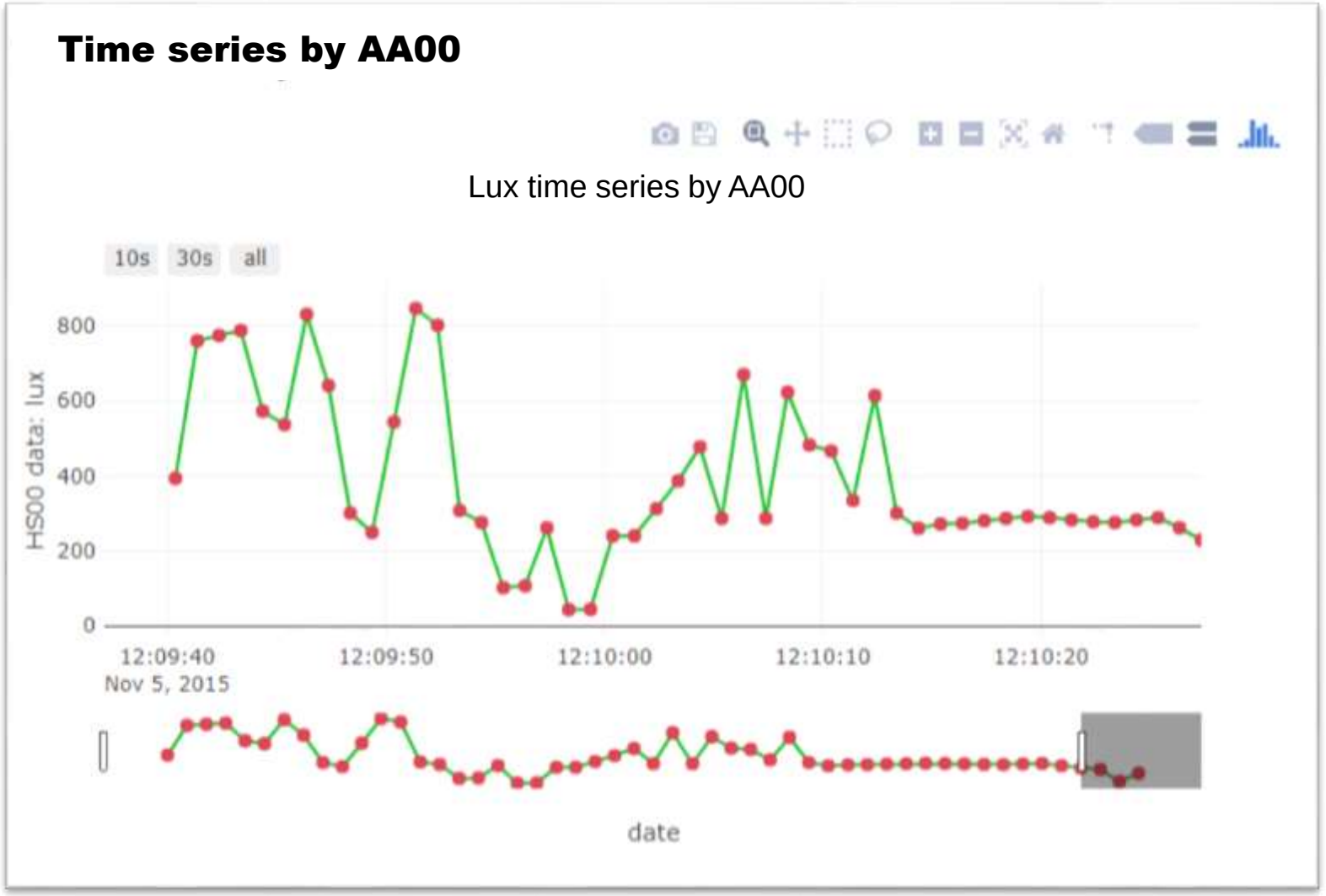
Figure 3
Typical Construction of a Plastic Coated Photocell



Layout [H S C]

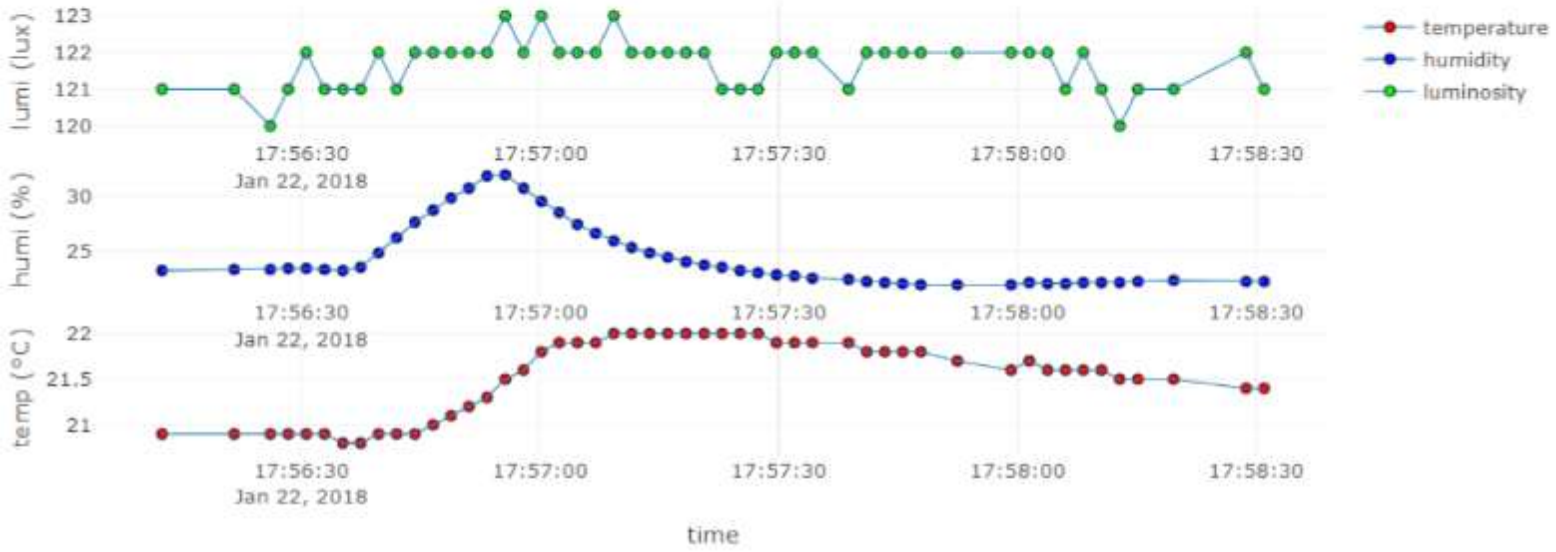

Physical Layer
Photocell
Arduino
USB Cable
Coordination Layer
Computer
Node.js server
Web Browser
Internet
Application Layer
Plotly
MongoDB
Tensorflow.js

Arduino data + plotly

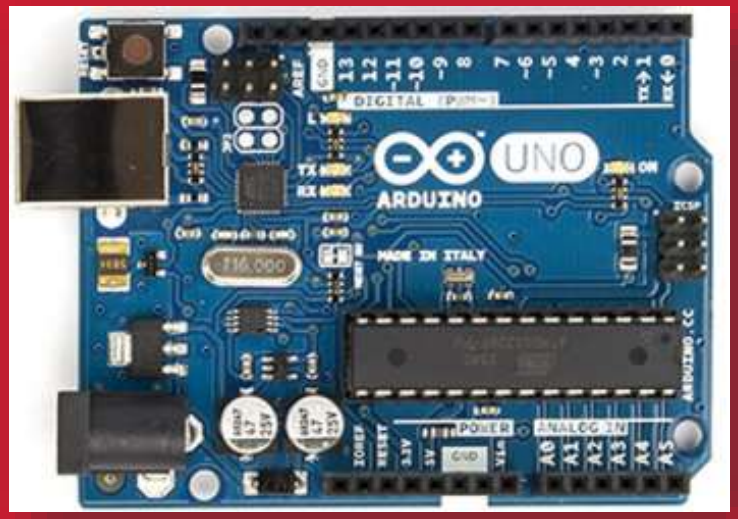


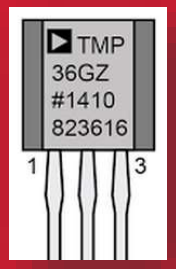
Real-time Weather Station from sensors

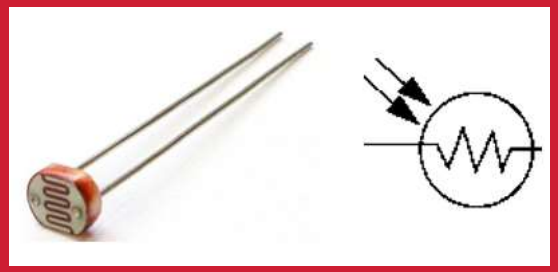
on Time: 2018-01-22 17:58:31.012

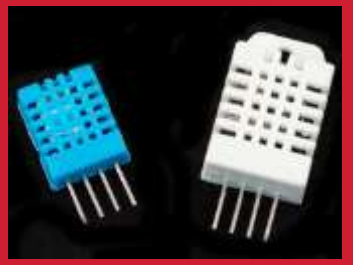


Arduino Sensors + Node.js

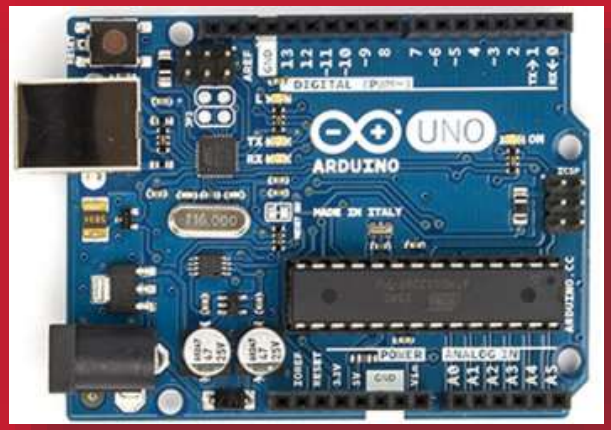






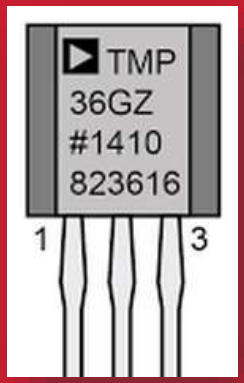


Single sensor: tmp36



TMP36

Node project



A4.1.1 tmp36 node project

Start tmp36-node project

1. **Go to my working folder**
2. **md iot & cd iot**
3. **md tmp36**
4. **cd tmp36**
5. **dir**

```
npm
D:\Portable\NodeJSPortable\Data>cd aann

D:\Portable\NodeJSPortable\Data\aann>dir
 D 드라이브의 볼륨: DATA
 볼륨 일련 번호: 7A01-106A

 D:\Portable\NodeJSPortable\Data\aann 디렉터리

2018-09-10  오후 04:12    <DIR>          .
2018-09-10  오후 04:12    <DIR>          ..
2018-09-10  오후 04:17    <DIR>          aa00App
2018-09-10  오후 03:47    <DIR>          express
2018-09-10  오후 03:07    <DIR>          expressTest
2018-09-03  오후 04:33    <DIR>          server
2018-09-03  오후 05:37    <DIR>          start
               0개 파일                   0 바이트
               7개 디렉터리  848,410,902,528 바이트 남음

D:\Portable\NodeJSPortable\Data\aann>md iot

D:\Portable\NodeJSPortable\Data\aann>cd iot

D:\Portable\NodeJSPortable\Data\aann\iot>md tmp36

D:\Portable\NodeJSPortable\Data\aann\iot>cd tmp36

D:\Portable\NodeJSPortable\Data\aann\iot\tmp36>dir
 D 드라이브의 볼륨: DATA
 볼륨 일련 번호: 7A01-106A

 D:\Portable\NodeJSPortable\Data\aann\iot\tmp36 디렉터리

2018-10-20  오후 03:02    <DIR>          .
2018-10-20  오후 03:02    <DIR>          ..
               0개 파일                   0 바이트
               2개 디렉터리  848,410,902,528 바이트 남음

D:\Portable\NodeJSPortable\Data\aann\iot\tmp36>
```

A4.1.2 tmp36 node project

Set tmp36-node project

1. **npm init**
2. **description**

   tmp36-node project
3. **entry point**

   tmp36_node.js
4. **author**

   your id : aann

```
npm
package name: (tmp36)
version: (1.0.0)
description: tmp36-node project
entry point: (index.js) tmp36_node.js
test command:
git repository:
keywords: tmp36 node.js
author: aa00
license: (ISC) MIT
About to write to D:\Portable\NodeJSPortable\Data\aann\iot\

{
  "name": "tmp36",
  "version": "1.0.0",
  "description": "tmp36-node project",
  "main": "tmp36_node.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "tmp36",
    "node.js"
  ],
  "author": "aa00",
  "license": "MIT"
}


Is this OK? (yes) y

D:\Portable\NodeJSPortable\Data\aann\iot\tmp36>
```

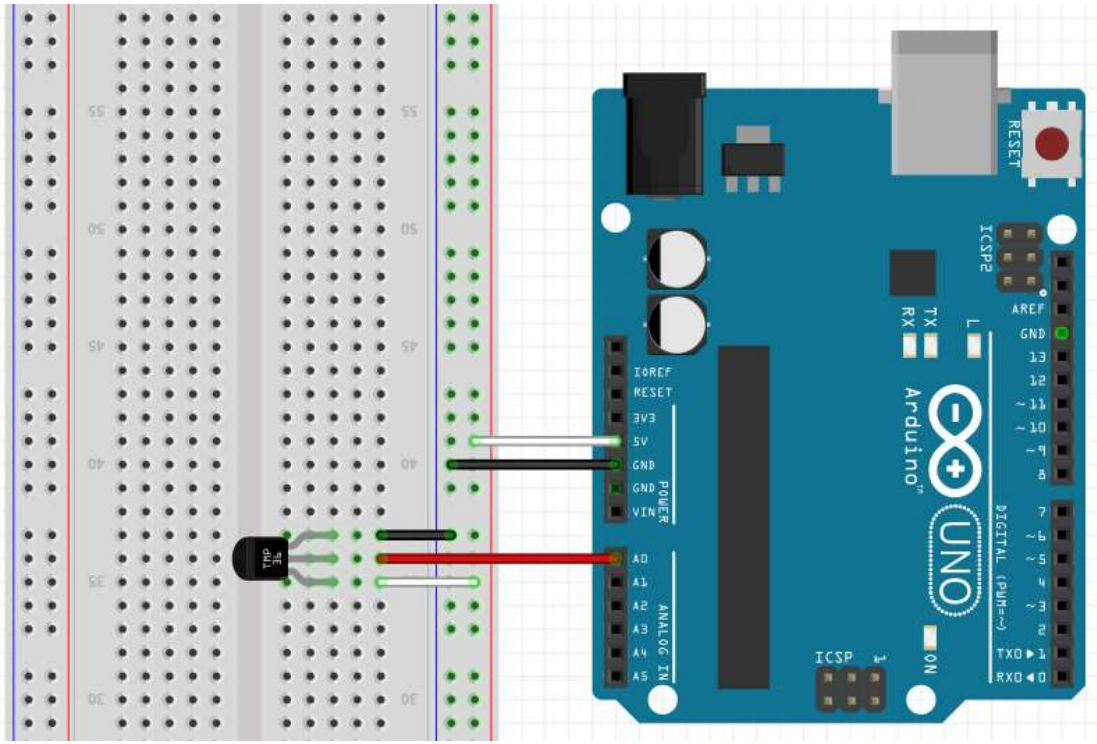
A4.1.3 tmp36 node project

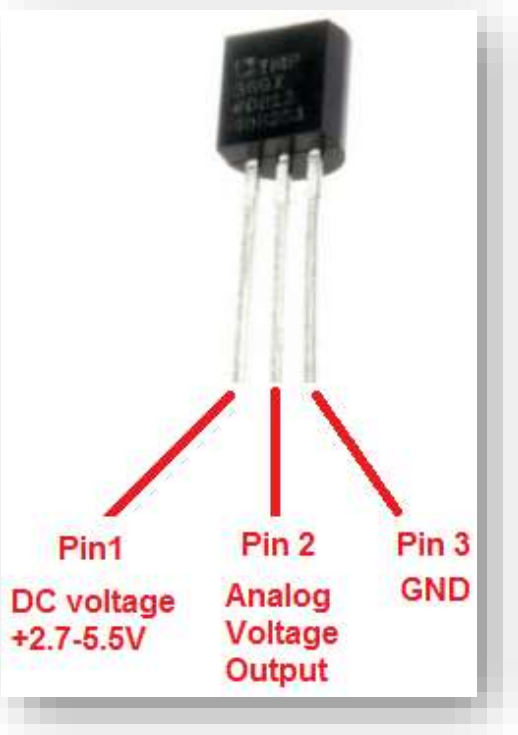
package.json

package.json

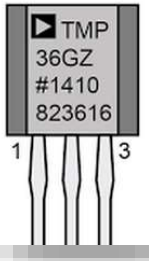
```
{
  "name": "tmp36",
  "version": "1.0.0",
  "description": "tmp36-node project",
  "main": "tmp36_node.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "tmp36",
    "node.js"
  ],
  "author": "aa00",
  "license": "MIT"
}
```

A3.1.1 Temperature sensor [TMP36]





Parts : TMP36



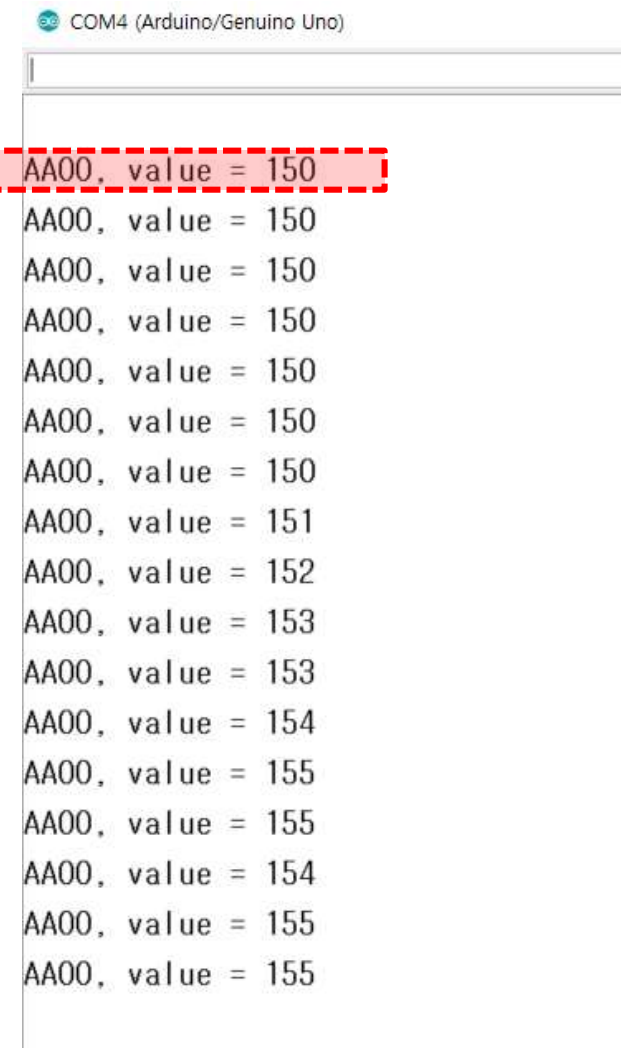
- **Size:** TO-92 package (about 0.2" x 0.2" x 0.2") with three leads
- **Price:** $2.00 at the Adafruit shop
- **Temperature range:** -40°C to 150°C / -40°F to 302°F
- **Output range:** 0.1V (-40°C) to 2.0V (150°C) but accuracy decreases after 125°C
- **Power supply:** 2.7V to 5.5V only, 0.05 mA current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

A4.1.4 tmp36 node project

AAnn_TMP36_NodeJS_start.ino

```
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.println(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 }
```

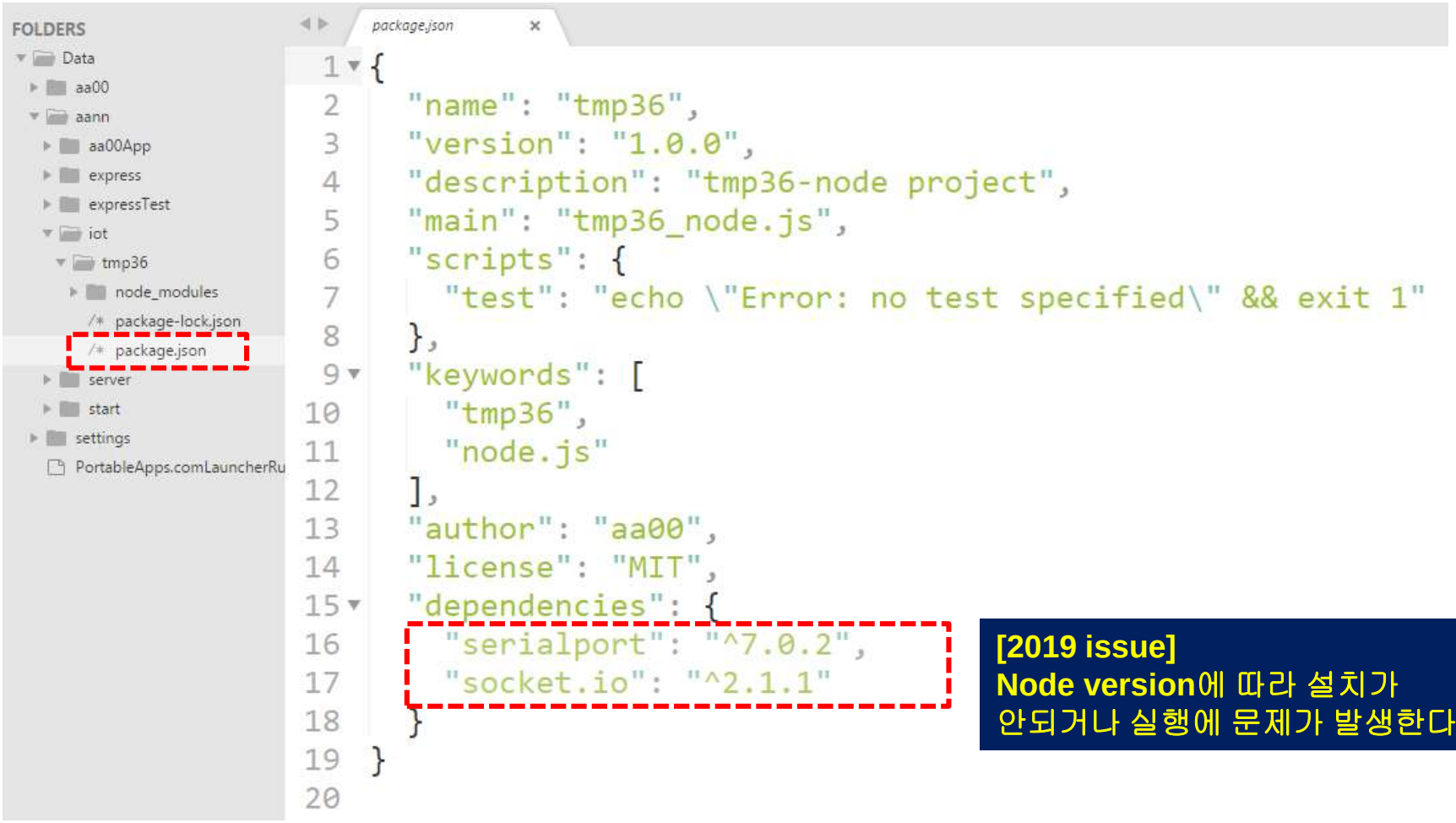
Serial output (A0, 0 ~ 1023)



A4.1.5 tmp36 node project

Go to tmp36 subfolder

- **npm install –save serialport**
- **npm install –save socket.io**



[2019 issue]
Node version에 따라 설치가
안되거나 실행에 문제가 발생한다.

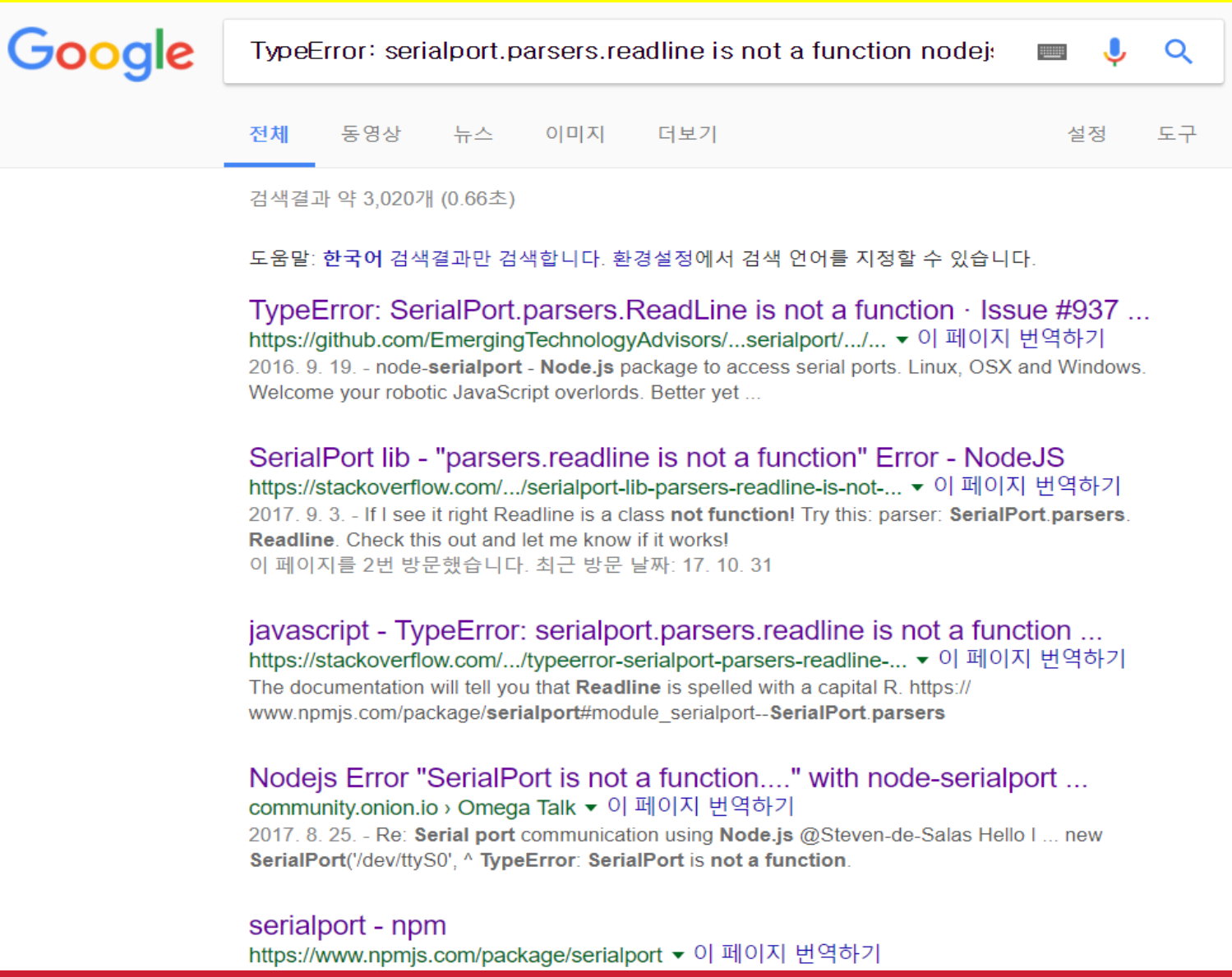
Error & Bug ---

serialport 6.x 버전의 API 변화로 오류 발생, 버전 downgrade

```
D:\Portable\NodeJSPortable\Data\aann\iot\tmp36\node_modules\@serialport\bindings\lib\win32.js
:9
class WindowsBinding extends AbstractBinding {
^^^^^

SyntaxError: Block-scoped declarations (let, const, function, class) not yet supported
outside strict mode
    at exports.runInThisContext (vm.js:53:16)
    at Module._compile (module.js:387:25)
    at Object.Module._extensions..js (module.js:422:10)
    at Module.load (module.js:357:32)
    at Function.Module._load (module.js:314:12)
    at Module.require (module.js:367:17)
    at require (internal/module.js:20:19)
    at Object.<anonymous> (D:\Portable\NodeJSPortable\Data\aann\iot\tmp36\node_modules\@seria
    lport\bindings\lib\index.js:6:22)
    at Module._compile (module.js:413:34)
    at Object.Module._extensions..js (module.js:422:10)
[Finished in 0.3s]
```

Error & Bug ---

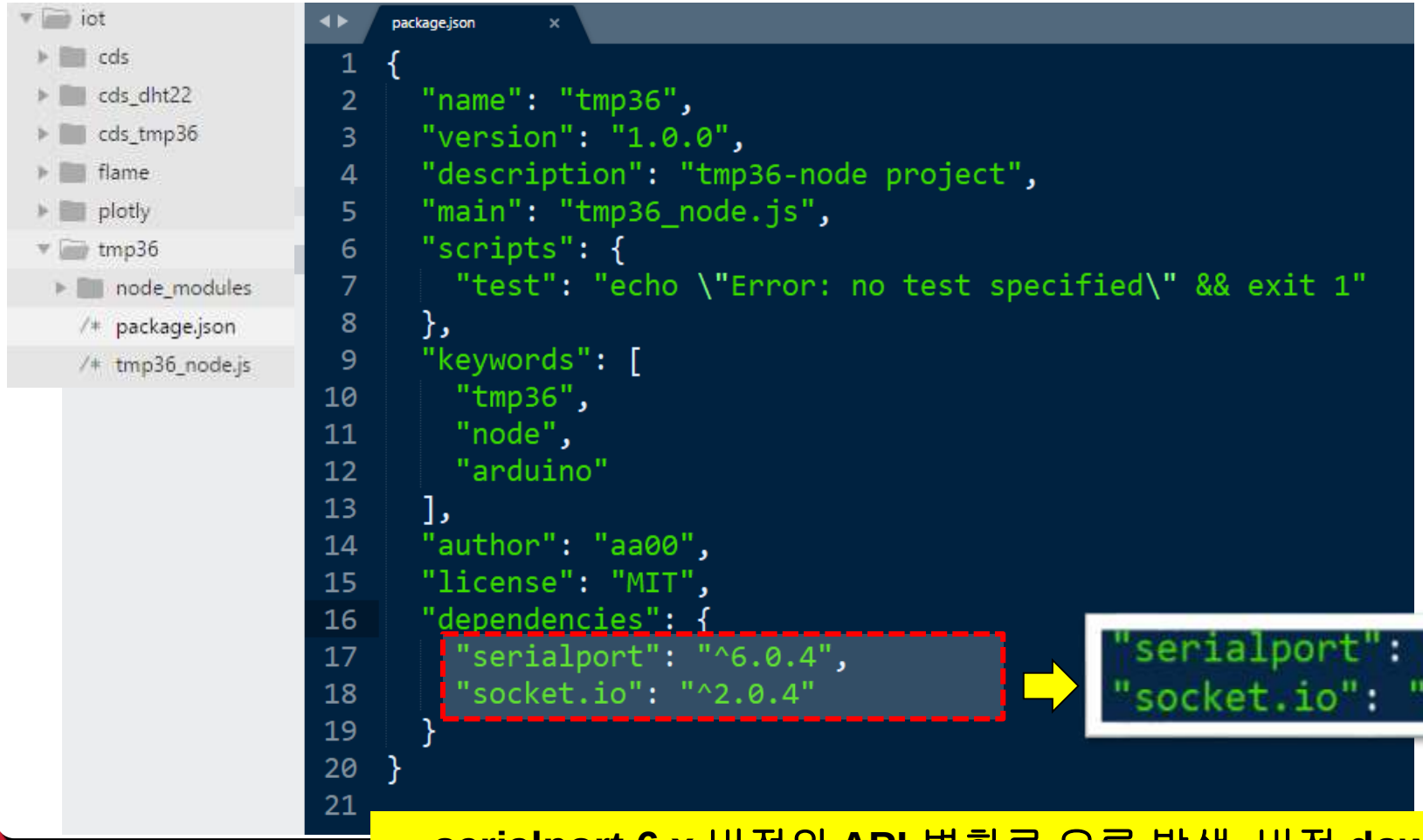
serialport 6.x 버전의 API 변화로 오류 발생, 버전 downgrade



A4.1.6 tmp36 node project → downgrade

Go to tmp36 subfolder (after deleteing node_modules subfolder)

- **"dependencies" 속성의 버전을 아래와 같이 변경**
- **npm install**



serialport 6.x 버전의 API 변화로 오류 발생, 버전 downgrade

A4.1.7 tmp36 node project : code-1

tmp36_node_start.js

```
1  // tmp36_node.js
2
3  var serialport = require('serialport');
4  var portName = 'COM10';  // check your COM port!!
5  var port     =   process.env.PORT || 3000;
6
7  var io = require('socket.io').listen(port);
8
9  // serial port object
10 var sp = new serialport(portName,{
11     baudRate: 9600,   // 9600  38400
12     dataBits: 8,
13     parity: 'none',
14     stopBits: 1,
15     flowControl: false,
16     parser: serialport.parsers.readline('\r\n')  // new serialport.pars
17 });
18
19 var tdata = [];  // Array
20
21 sp.on('data', function (data) { // call back when data is received
22     // raw data only
23         //console.log(data);
24         tdata = data;  // data
25         console.log("AA00," + tdata);
26         io.sockets.emit('message', tdata);  // send data to all clients
27 });
```

serialport 6.x 버전의 API 변화로 오류 발생, 버전 downgrade

A4.1.7 tmp36 node project : code-2

tmp36_node_start.js

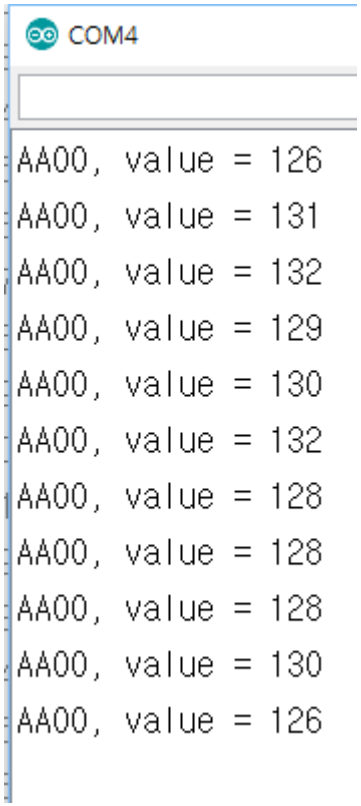
```
33 io.sockets.on('connection', function (socket) {
34     // If socket.io receives message from the client browser then
35     // this call back will be executed.
36     socket.on('message', function (msg) {
37         console.log(msg);
38     });
39     // If a web browser disconnects from Socket.IO then this callback is called.
40     socket.on('disconnect', function () {
41         console.log('disconnected');
42     });
43 });
44
```

serialport 6.x 버전의 API 변화로 오류 발생, 버전 downgrade 후 해결.

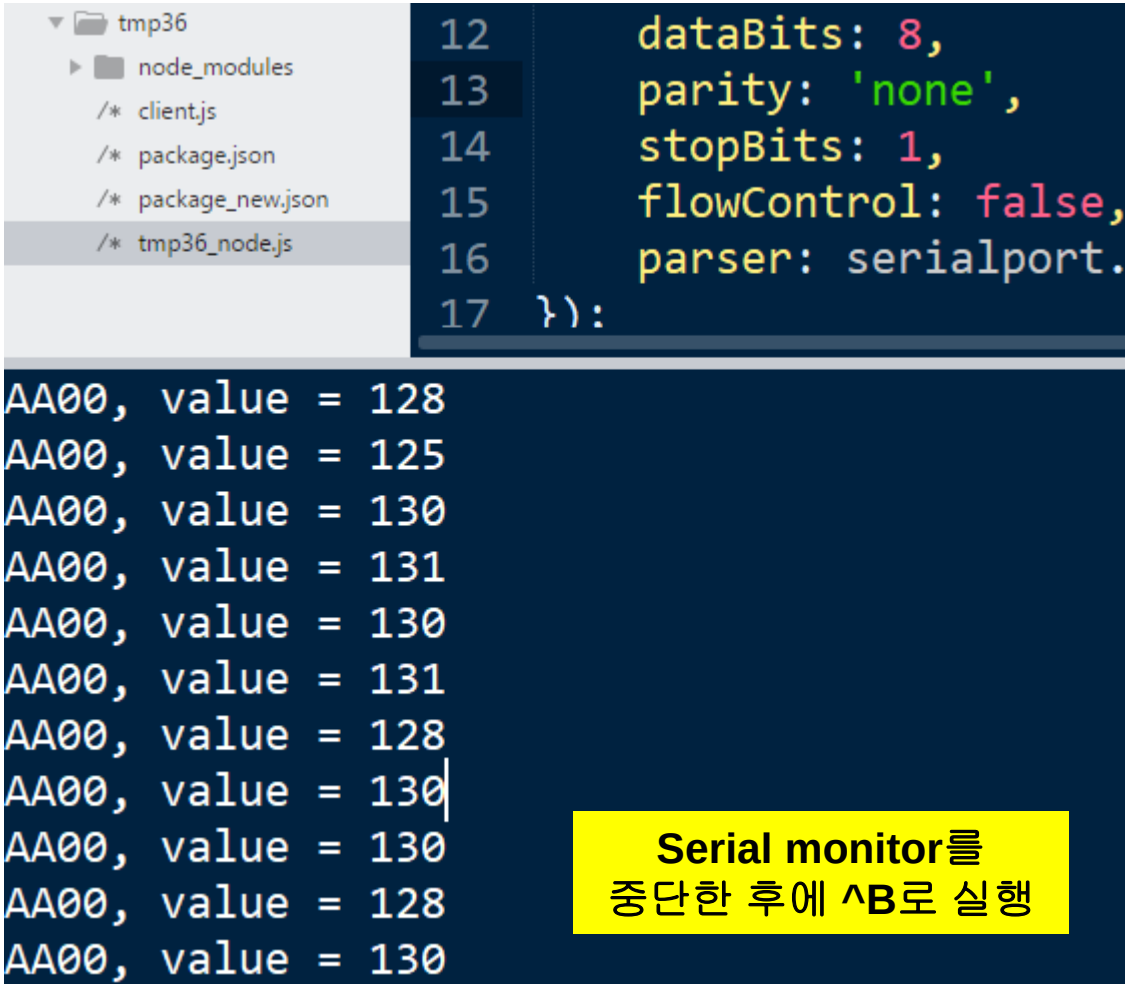
TypeError: SerialPort.parsers.ReadLine is not a function · Issue #937 ...
https://github.com/EmergingTechnologyAdvisors/.../... ▾ 이 페이지 번역하기
2016. 9. 19. - node-**serialport** - **Node.js** package to access serial ports. Linux, OSX and Windows. Welcome your robotic JavaScript overlords. Better yet ...

A4.1.8 tmp36 node project (after downgrade)

Serial output
(A0 in Arduino)



tmp36_node.js (^B로 실행)



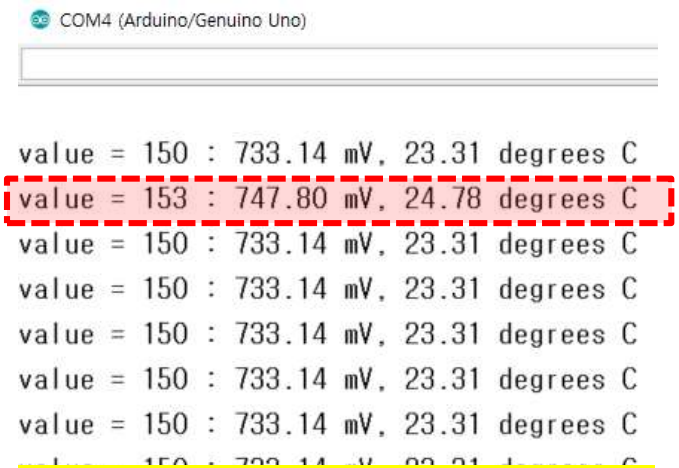
Serial monitor를
중단한 후에 ^B로 실행

A4.1.9 tmp36 node project (all messages)

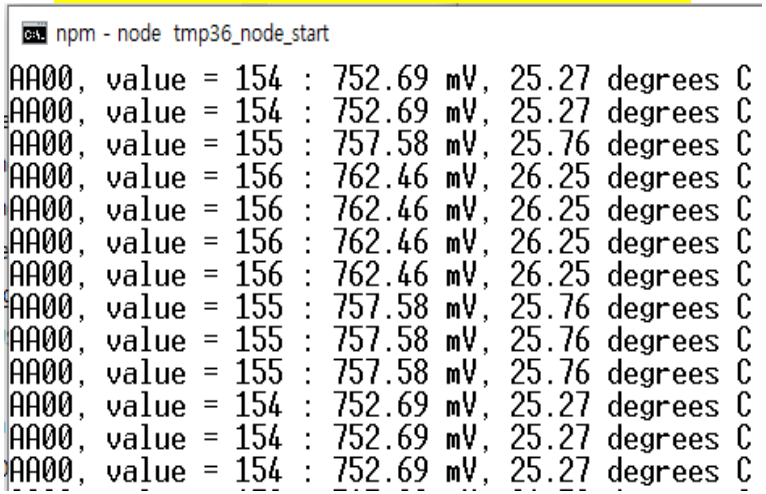
AAnn_TMP36_NodeJS.ino

```
12 void loop() {
13   //getting the voltage reading from the temperature sensor
14   int value = analogRead(TEMP_INPUT);
15   Serial.print("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 monitor



Node cmd



A4.1.9 tmp36 node project (all messages)

tmp36_node.js

```
19 var dStr = '';
20 var tdata = [];  // Array
21
22 sp.on('data', function (data) { // call back when data is
23     // raw data only
24         //console.log(data);
25         dStr = getDateString();
26         tdata[0] = dStr;  // date
27         tdata[1] = data;  // data
28         console.log('AA00,' + tdata);
29          io.sockets.emit('message', tdata);  // send data
30 });
31
32 // helper function to get a nicely formatted date string
33 function getDateString() {
34     var time = new Date().getTime();
35     // 32400000 is (GMT+9 Korea, GimHae)
36     // for your timezone just multiply +/-GMT by 3600000
37     var datestr = new Date(time +32400000).
38     toISOString().replace(/T/, ' ').replace(/Z/, '');
39     return datestr;
40 }
```

Node cmd에서 node tmp36_node 로 실행

```
D:\Portable\NodeJSPortable\Data\aa00\iot\tmp36a>node tmp36_node
AA00,2019-10-02 11:53:33.119,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:34.119,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:35.122,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:36.122,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:37.126,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:38.125,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:39.128,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:40.127,value = 149 : 728.25 mV, 22.83 degrees C
AA00,2019-10-02 11:53:41.131,value = 149 : 728.25 mV, 22.83 degrees C
AA00,2019-10-02 11:53:42.134,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:43.133,value = 151 : 738.03 mV, 23.80 degrees C
AA00,2019-10-02 11:53:44.138,value = 149 : 728.25 mV, 22.83 degrees C
AA00,2019-10-02 11:53:45.137,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:46.139,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:47.140,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:48.143,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:49.142,value = 149 : 728.25 mV, 22.83 degrees C
AA00,2019-10-02 11:53:50.146,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:51.145,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:52.148,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:53.153,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:54.152,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:55.155,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:56.155,value = 150 : 733.14 mV, 23.31 degrees C
AA00,2019-10-02 11:53:57.158,value = 151 : 738.03 mV, 23.80 degrees C
```

AAnn_tmp36_message.png 로 저장

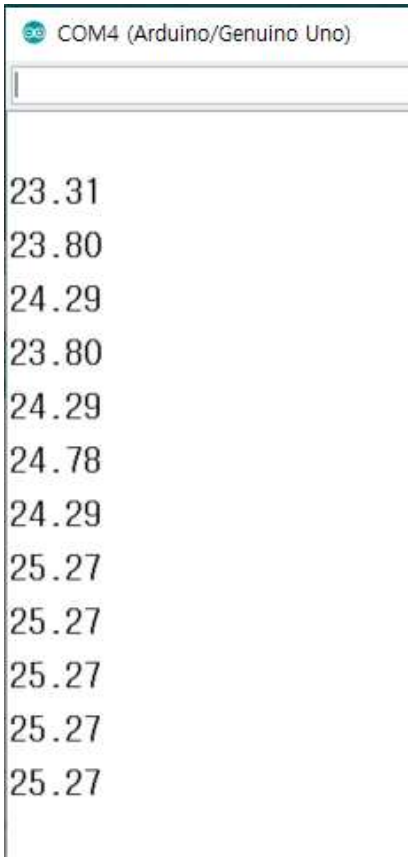
A4.1.10 tmp36 node project (only data)

AAnn_TMP36_NodeJS.ino 수정

AA00_TMP36_NodeJS

```
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(" Temperature, ");
30   Serial.println(temperatureC);
31 //   Serial.println(" degrees C");
32
33   delay(1000);
34 }
```

실행 결과



A4.1.11 tmp36 node project (date & data → IOT)

tmp36_node.js

```
19 var dStr = '';
20 var tdata = [];  // Array
21
22 sp.on('data', function (data) { // call back when data is
23     // raw data only
24         //console.log(data);
25         dStr = getDateString();
26         tdata[0] = dStr;  // date
27         tdata[1] = data;  // data
28         console.log('AA00,' + tdata);
29         io.sockets.emit('message', tdata);  // send data
30 });
31
32 // helper function to get a nicely formatted date string
33 function getDateString() {
34     var time = new Date().getTime();
35     // 32400000 is (GMT+9 Korea, GimHae)
36     // for your timezone just multiply +/-GMT by 3600000
37     var datestr = new Date(time +32400000).
38     toISOString().replace(/T/, ' ').replace(/Z/, '');
39     return datestr;
40 }
```

Node cmd에서
node tmp36_node

IOT data format
시간, data
시간, 온도

```
AA00,2019-10-02 11:59:32.529,23.31
AA00,2019-10-02 11:59:33.528,23.31
AA00,2019-10-02 11:59:34.527,23.31
AA00,2019-10-02 11:59:35.531,23.31
AA00,2019-10-02 11:59:36.530,23.80
AA00,2019-10-02 11:59:37.529,24.29
AA00,2019-10-02 11:59:38.534,25.76
AA00,2019-10-02 11:59:39.533,24.78
AA00,2019-10-02 11:59:40.532,24.78
AA00,2019-10-02 11:59:41.536,24.78
AA00,2019-10-02 11:59:42.535,24.78
```

시간 , 온도

공백없이 “,”로
시간과 온도 구분

A4.1.12 tmp36 node project (실행 결과)

▸ Sublime Text 3에서 실행

```
AA00,2018-10-21 10:44:18.278,16.96
AA00,2018-10-21 10:44:19.278,17.45
AA00,2018-10-21 10:44:20.276,16.96
AA00,2018-10-21 10:44:21.276,16.96
AA00,2018-10-21 10:44:22.276,17.45
AA00,2018-10-21 10:44:23.279,16.96
AA00,2018-10-21 10:44:24.277,16.96
AA00,2018-10-21 10:44:25.278,17.45
AA00,2018-10-21 10:44:26.277,17.45
AA00,2018-10-21 10:44:27.276,16.47
AA00,2018-10-21 10:44:28.280,17.45
```

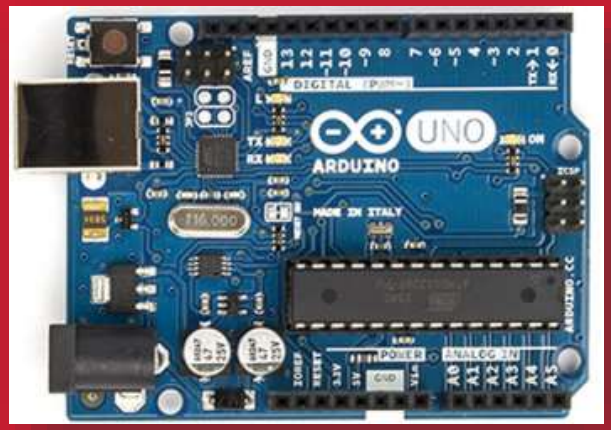
▸ Node cmd에서 실행

node tmp36_node

```
npm - node  tmp36_node
^C
D:\Portable\NodeJSPortable\Data\aann\iot\tmp36>node tmp36_node
AA00,2018-10-21 11:07:38.784,16.47
AA00,2018-10-21 11:07:39.784,17.45
AA00,2018-10-21 11:07:40.783,17.45
AA00,2018-10-21 11:07:41.782,17.45
AA00,2018-10-21 11:07:42.782,17.45
AA00,2018-10-21 11:07:43.785,17.94
AA00,2018-10-21 11:07:44.784,17.94
AA00,2018-10-21 11:07:45.784,16.96
```

AAnn_tmp36_IOT_data.png
로 저장

Single sensor: CdS

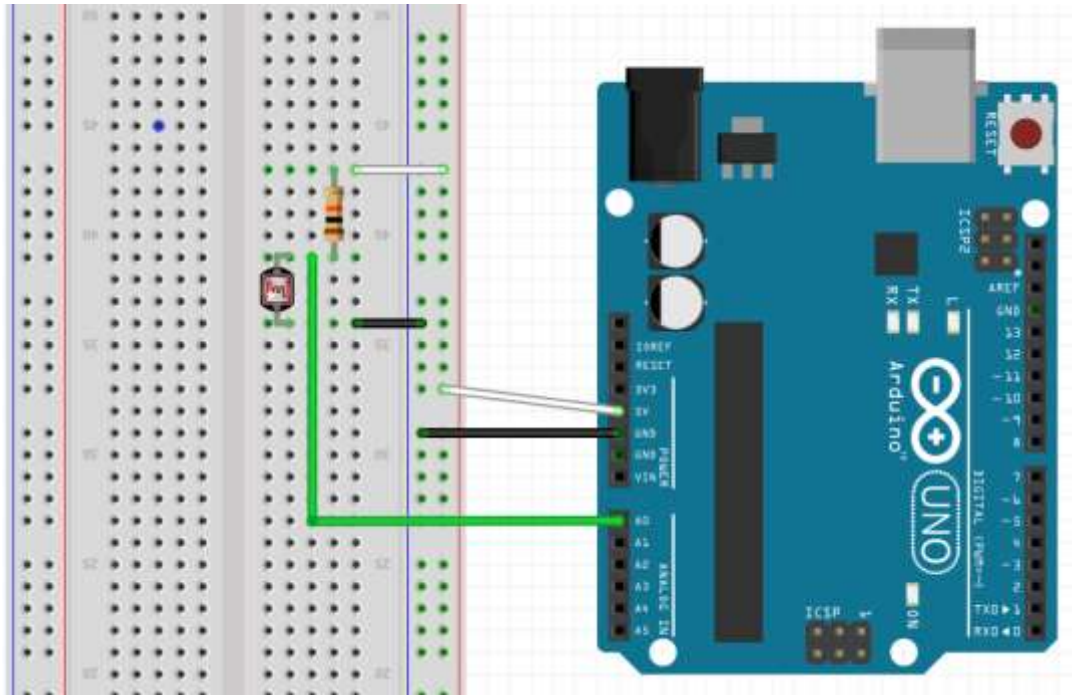


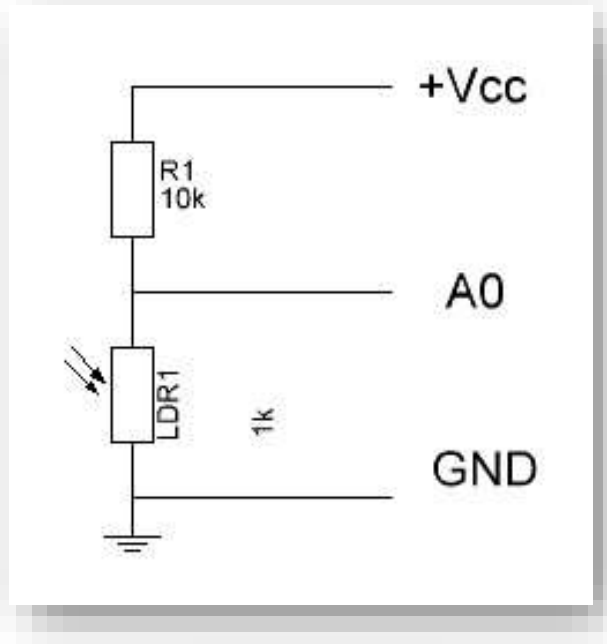
CdS (LDR)

Node project

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

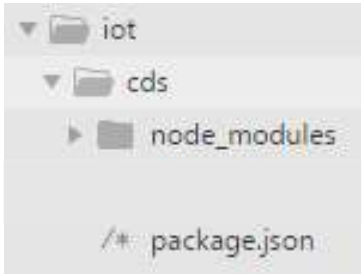
A4.2.1 Luminosity sensor [Photocell LDR]

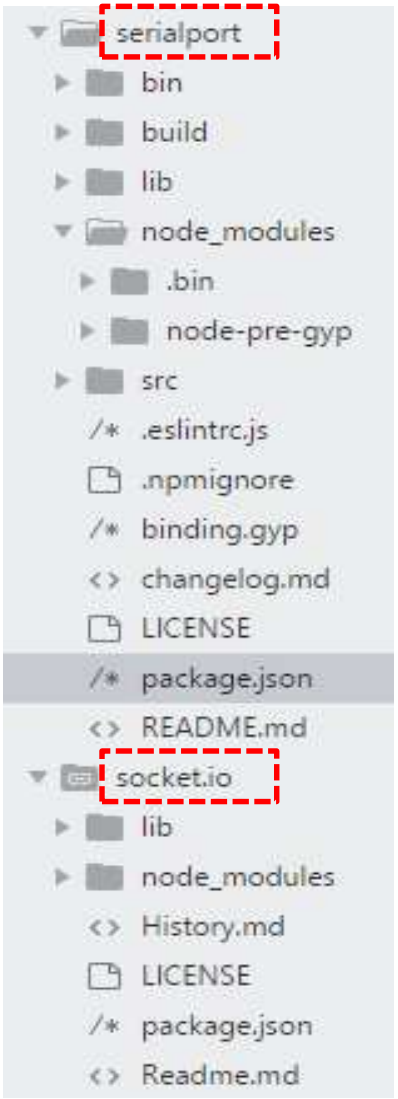
1. Make cds node project
   - ➢ md cds in iot folder
   - ➢ cd cds
2. Go to cds subfolder
   - ➢ npm init

"main": "cds_node.js"
"author": “aann"

A4.2.2 Luminosity sensor [Photocell LDR]

1. Make cds node project
   - md cds in iot folder
   - cd cds
2. Go to cds subfolder
   - npm init
   - **npm install –save serialport@4.0.7**
   - **npm install –save socket.io@1.7.3**





You can check version of each module by browing package.json in each module subfolder.

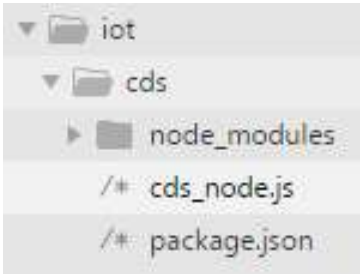
A4.2.3 Luminosity sensor [Photocell LDR]

1. **Make cds node project**
   - **md cds**
   - **cd cds**
2. **Go to cds subfolder**
   - **npm init**
   - **npm install –save serialport@4.0.7**
   - **npm install –save socket.io@1.7.3**

package,json

```
{
  "name": "cds",
  "version": "1.0.0",
  "description": "cds-node project",
  "main": "cds_node.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "aa00",
  "license": "MIT",
  "dependencies": {
    "serialport": "^4.0.7",
    "socket.io": "^1.7.3"
  }
}
```

A4.2.4 Luminosity sensor [Photocell LDR]



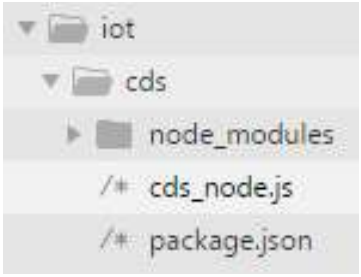
Save tmp36_node.js as cds_node.js

```
var dStr = '';
var tdata = [];

sp.on('data', function (data) { // call back when data is received
    // raw data only
        //console.log(data);
        dStr = getDateString();
        tdata[0] = dStr;  // date
        tdata[1] = data;  // data
        console.log("AA00," + tdata);
        io.sockets.emit('message', tdata);  // send data to all clients
});

// helper function to get a nicely formatted date string
function getDateString() {
    var time = new Date().getTime();
    // 32400000 is (GMT+9 Korea, GimHae)
    // for your timezone just multiply +/-GMT by 3600000
    var datestr = new Date(time +32400000).
    toISOString().replace(/T/, ' ').replace(/Z/, '');
    return datestr;
}
```

A4.2.5 cds_ node project (실행 결과)

▸ Sublime Text 3에서 실행

```
AA00,2018-01-14 19:12:42.037,86
AA00,2018-01-14 19:12:43.035,36
AA00,2018-01-14 19:12:44.039,54
AA00,2018-01-14 19:12:45.038,175
AA00,2018-01-14 19:12:46.042,175
AA00,2018-01-14 19:12:47.041,174
```

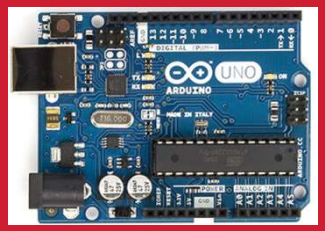


▸ Node cmd에서 실행

```
node cds_node
```

```
NodeJS - node  cds_node

D:\Portable\NodeJSPortable\Data\aa00\iot\cds>node cds_node
AA00,2018-01-14 19:15:33.602,176
AA00,2018-01-14 19:15:34.601,45
AA00,2018-01-14 19:15:35.601,35
AA00,2018-01-14 19:15:36.604,33
AA00,2018-01-14 19:15:37.604,175
```

AAnn_cds_IOT_data.png
로 저장



[Practice]

- [wk05]
  - Arduino sensors
  - Complete your project
  - Submit folder : AAnn_Rpt05

wk05 : Practice-04 : AAnn_Rpt05

- [Target of this week]
  - Complete your works
  - Save your outcomes and upload outputs in giyhub

제출폴더명 : **AAnn_Rpt05**

- 제출할 파일들
  1. AAnn_TMP36.png
  2. AAnn_LCD_hello.png
  3. AAnn_LCD_lux.png
  4. AAnn_tmp36_message.png
  5. AAnn_tmp36_IOT_data.png
  6. AAnn_cds_IOT_data.png
  7. All *.ino

[Upload to github]

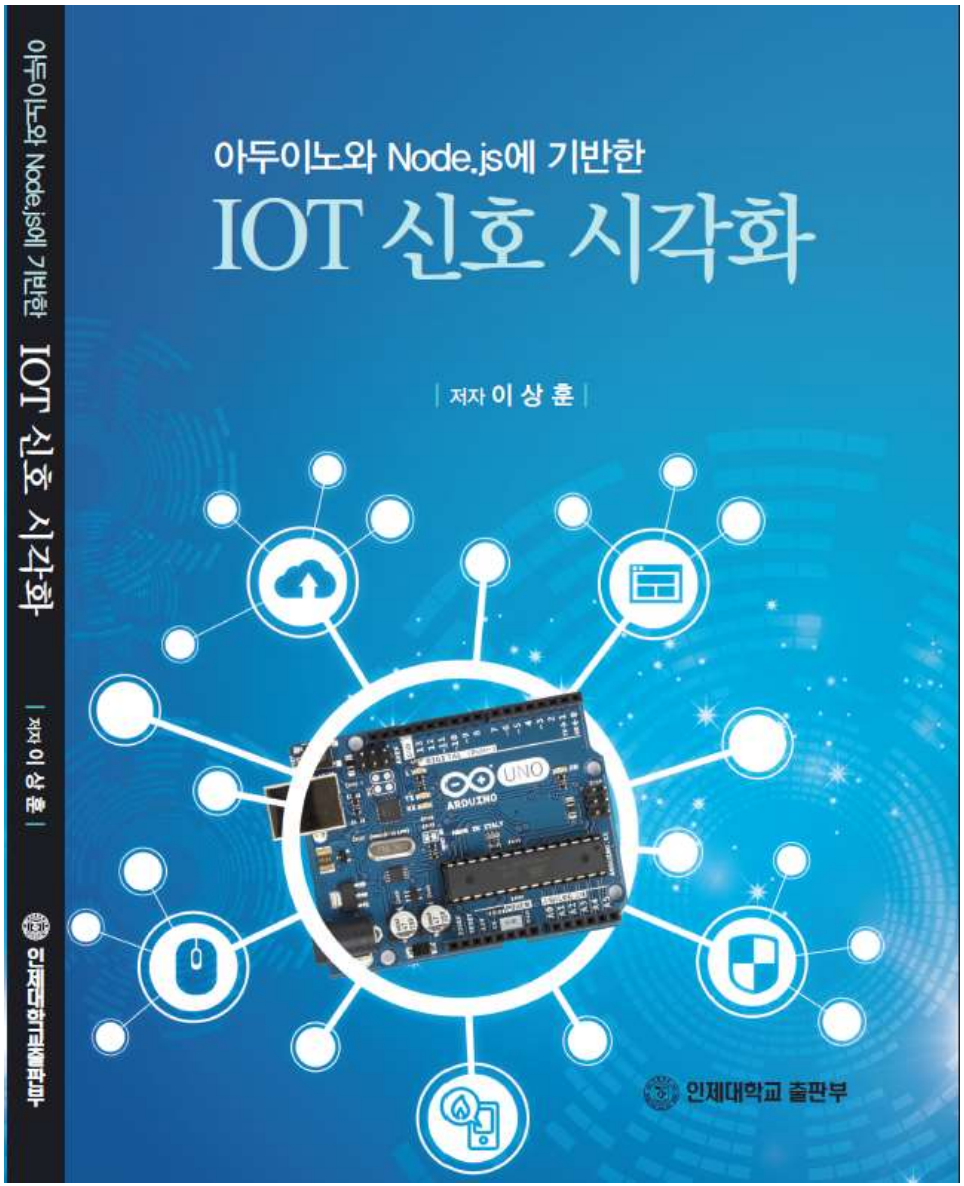
- [wk05]
  - upload all work of this week
  - Use repo “aann” in github
  - upload folder “aann_rpt05” in your github.

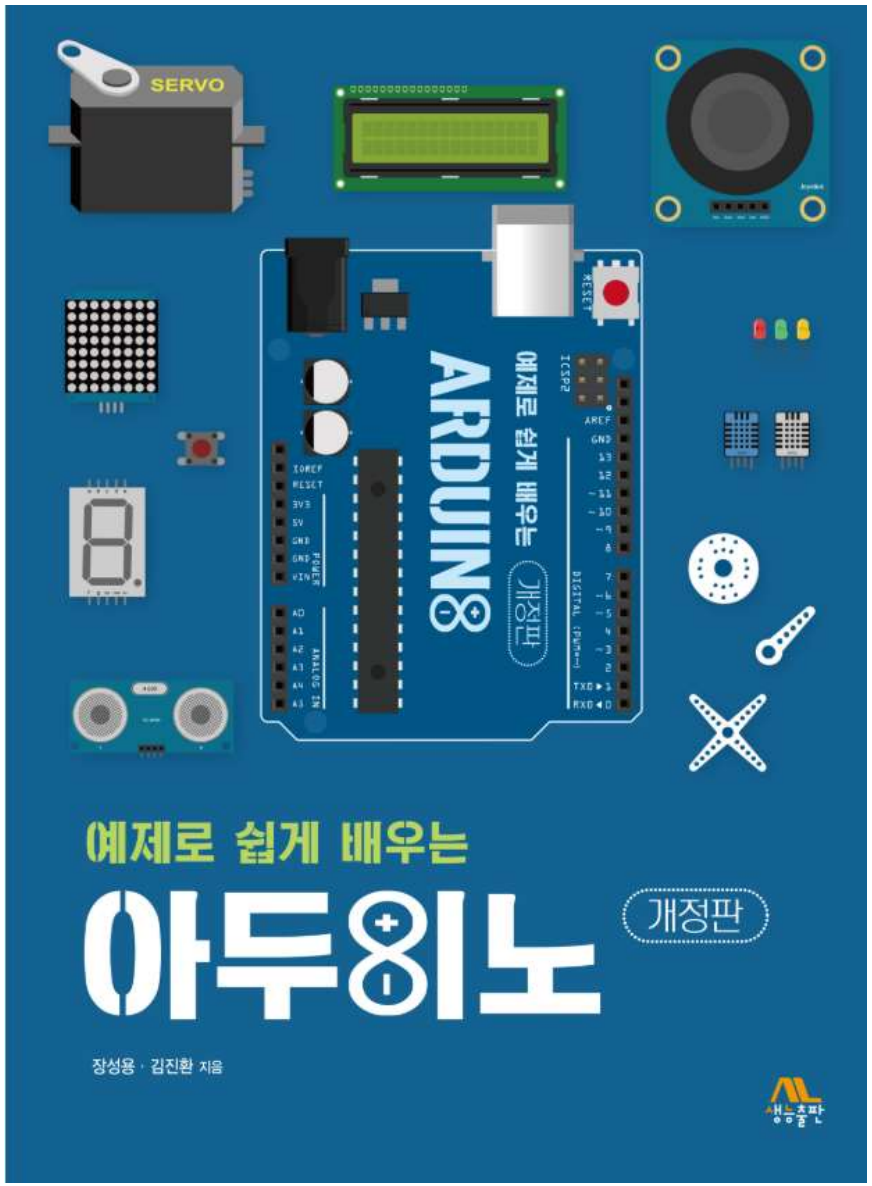
Lecture materials

References & good sites

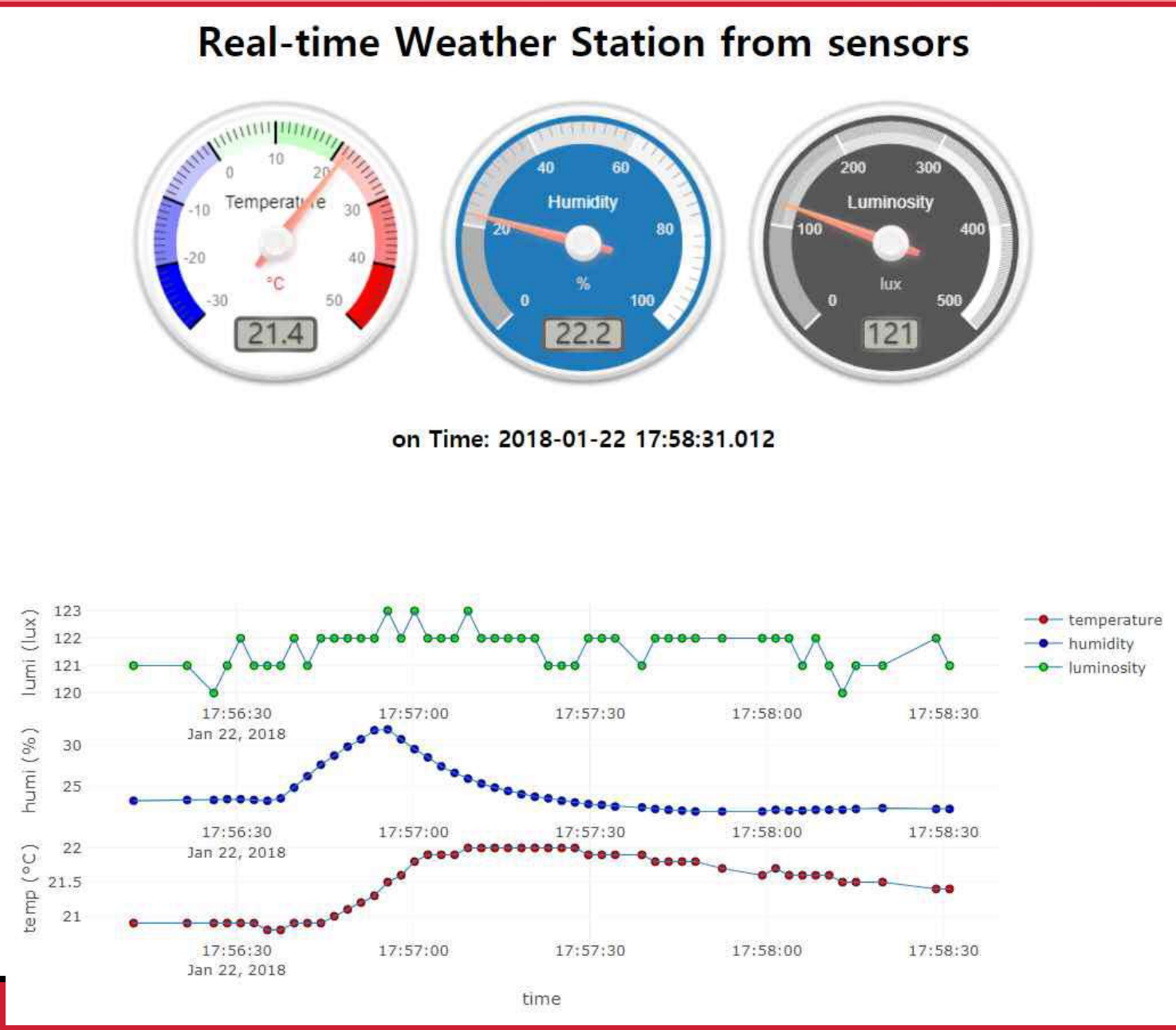
- ✓ http://www.arduino.cc Arduino Homepage
- ✓ http://www.nodejs.org/ko Node.js
- ✓ https://plot.ly/ plotly
- ✓ https://www.mongodb.com/ MongoDB
- ✓ http://www.w3schools.com By w3schools
- ✓ http://www.github.com GitHub

주교재 및 참고도서


아두이노와 Node.js에 기반한
IOT 신호 시각화
| 저자 이 상 훈 |
인제대학교 출판부
아두이노와 Node.js에 기반한 IOT 신호 시각화
| 저자 이 상 훈 |
인제대학교 컴퓨터공학과


SERVO
ARDUINO
예제로 쉽게 배우는
개정판
예제로 쉽게 배우는
아두이노
개정판
장성용 · 김진환 지음

Target of this class



Another target of this class