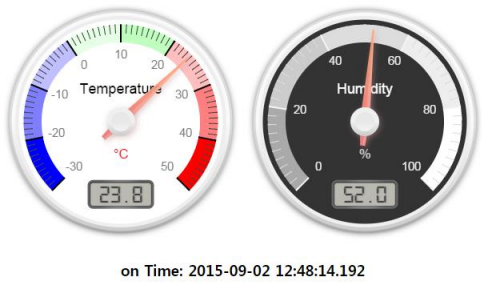
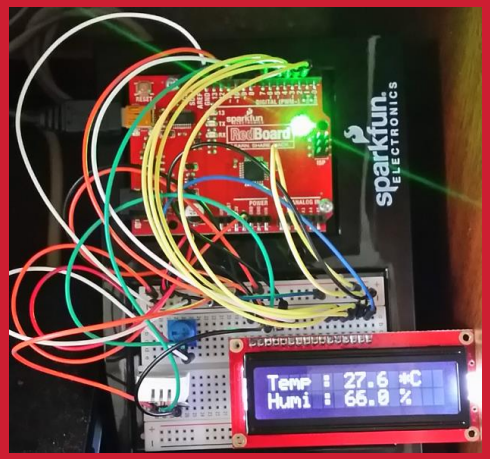
# Healthcare-IOT

## [wk05]




# Arduino Sensors

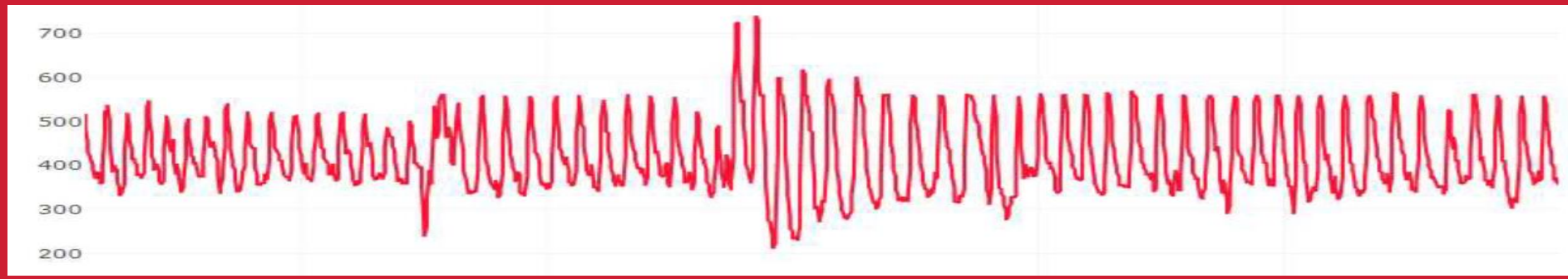



Visualization of Healthcare Signals using Arduino & Node.js

HCit, INJE University

1st semester, 2018

Email : chaos21c@gmail.com



# My ID

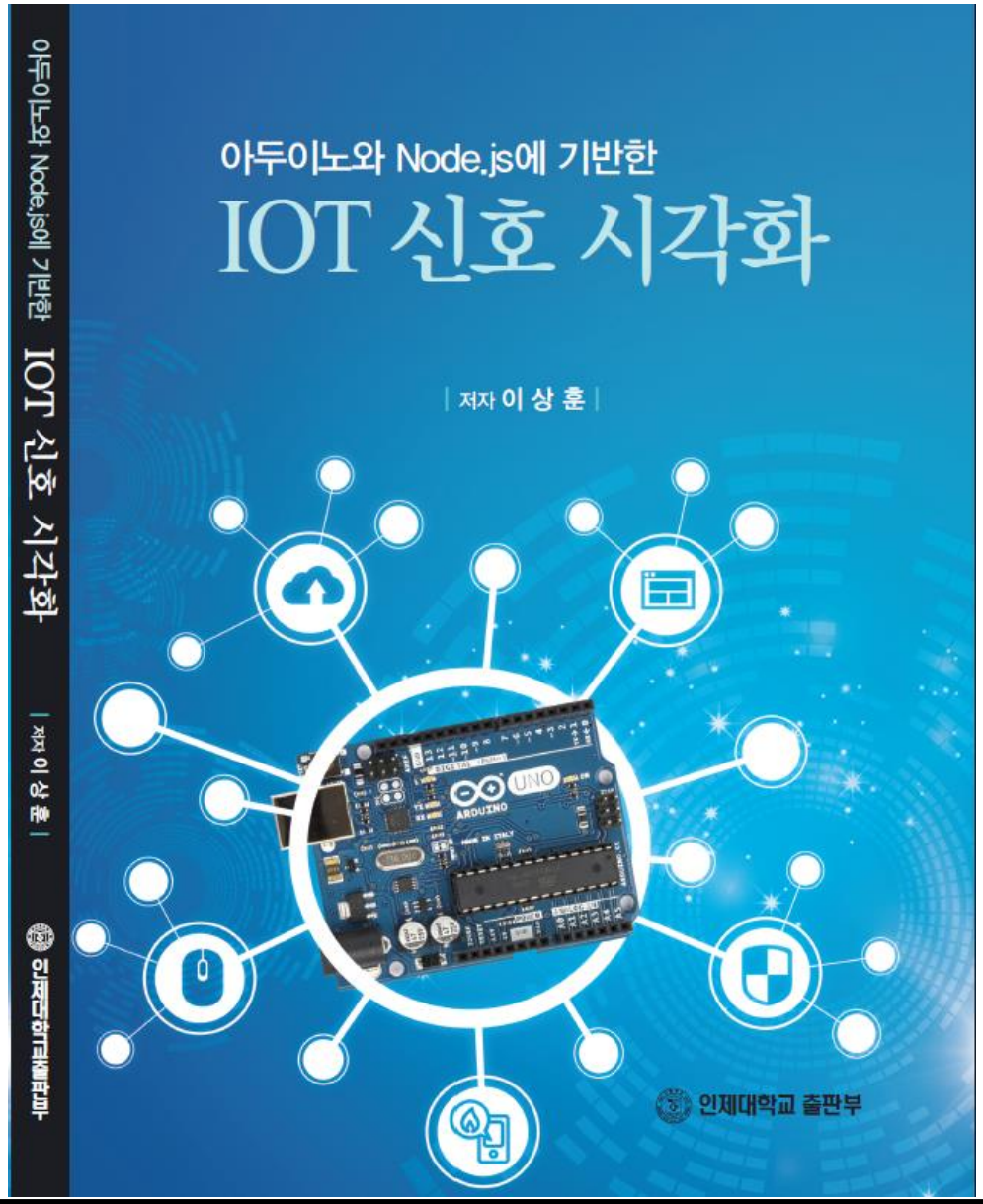
오전

| 성명 | ID |
|---|---|
| 김민선 | HS01 |
| 김영걸 | HS02 |
| 김주란 | HS03 |
| 김주현 | HS04 |
| 김태민 | HS05 |
| 여준하 | HS06 |
| 이수민 | HS07 |
| 정민지 | HS08 |
| 정유현 | HS09 |
| 정재은 | HS10 |
| 주하영 | HS11 |
| 한준영 | HS12 |

오후

| 성명 | ID |
|---|---|
| 신영주 | HS21 |
| 오가영 | HS22 |
| 윤민수 | HS23 |
| 윤진아 | HS24 |
| 이진영 | HS25 |
| 임상은 | HS26 |
| 임재형 | HS27 |
| 최민영 | HS28 |
| 황유빈 | HS29 |

# 주교재


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 출판부

# 주간계획서

| 주간계획서 | | | |
|---|---|---|---|
| 주차 | 수업방법 | 수업내용 | 과제물 |
| 1 | 강의/실습 | 수업 및 실습 안내<br>-포터블 소프트웨어 설치 | |
| 2 | 강의/실습 | Node.js I<br>- Node.js 코드의 기본 구조<br>- 기초 Node 서버 및 클라이언트 | 실습확인 |
| 3 | 강의/실습 | Node.js II<br>- Node.js Express 서버 | 실습확인 |
| 4 | 강의/실습/발표 | Arduino I<br>- 아날로그 신호 회로<br>- LCD를 이용한 센서 신호 모니터링 | 실습확인 |
| 5 | 강의/실습 | Arduino II<br>- 단일 센서 회로와 Node.js 연결<br>- 다중 센서 회로와 Node.js 연결 | 실습확인 |
| 6 | 강의/실습 | 프로젝트1<br>- 생체 센서 회로와 Node.js 연결<br>- 생체 신호 소개 | 프로젝트1 |
| 7 | 강의/실습/발표 | IOT 데이터 시각화 I (Plotly.js)<br>- 데이터 및 시계열 차트<br>- 데이터 스트리밍 | 실습확인 |
| 8 | 시험 | 중간고사 | |
| 9 | 강의/실습 | IOT 데이터 시각화 II (Plotly.js)<br>- 다중 센서 데이터 시각화<br>- 다중 센서 데이터 스트리밍 | 실습확인 |
| 10 | 강의/실습/발표 | 프로젝트II<br>- 생체 센서 데이터 시각화<br>- 생체 센서 데이터 스트리밍 | 프로젝트II |
| 11 | 강의/실습 | IOT 데이터 저장과 처리<br>- MongoDB 설치 및 Mongo shell<br>- MongoDB와 Node.js 연결 및 데이터 저장 | 실습확인 |
| 12 | 강의/실습 | 프로젝트III<br>- MongoDB에 IOT 데이터 저장 및 모니터링<br>- 생체 센서 데이터 저장 및 시각화 | 프로젝트III |
| 13 | 강의/실습 | IOT 데이터 마이닝<br>- 아두이노에서 발생된 데이터 관리<br>- 데이터마이닝 소개 | 실습확인 |
| 14 | 강의/실습/발표 | 프로젝트IV<br>- 생체 센서 데이터 관리<br>- 생체 센서 데이터 마이닝 | 프로젝트IV |
| 15 | 시험 | 기말고사 | |

# Purpose of HS

주요 수업 목표는 다음과 같다.

1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리
4. 생체 센서 발생 신호 처리, 시각화 및 저장
5. 생체 센서 발생 신호 저장 및 분석
6. 생체 신호 장비 활용 능력

# [Review]

- [wk04]
  - Arduino basic circuits
  - Complete your project
  - Submit file : HSnn_Rpt03.zip

# wk04 : Practice-03 : HSnn_Rpt03.zip

- **[Target of this week]**
  - Complete your works
  - Save your outcomes and compress 4 figures

제출파일명 : **HSnn_Rpt03.zip**

- 압축할 파일들

① **HSnn_Blink.png**

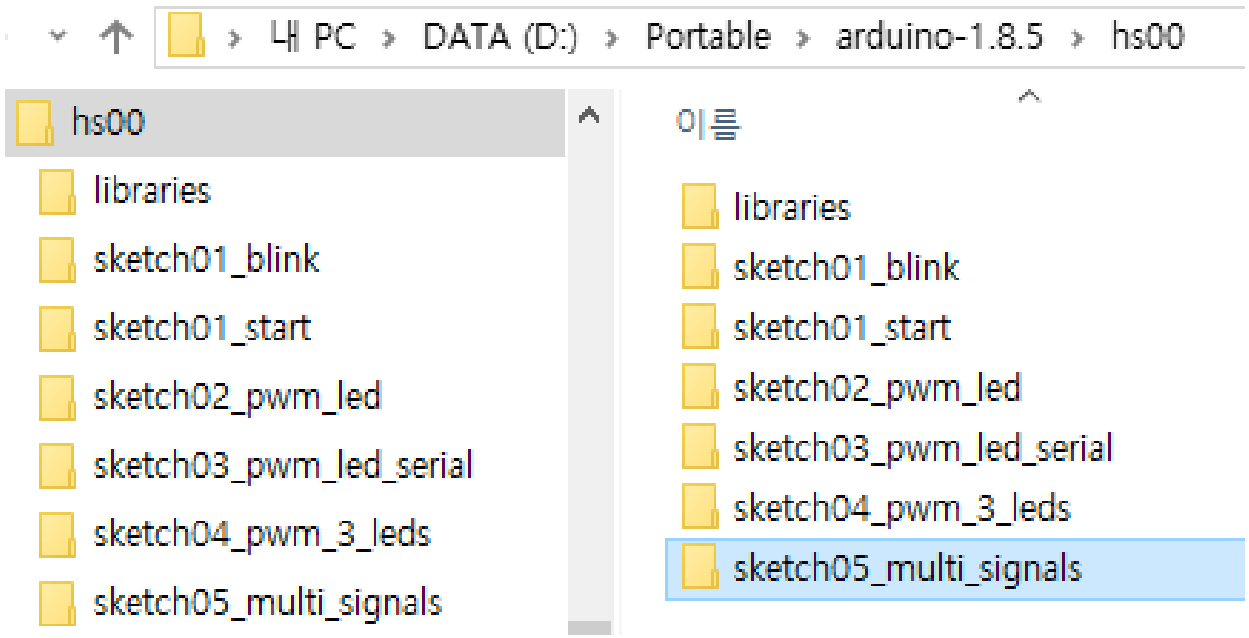
② **HSnn_Monitoring.png**

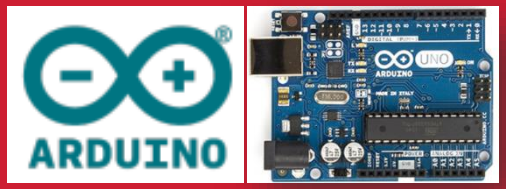
③ **HSnn_multi_Monitoring.png**

④ **HSnn_multi_Signals.png**

**Email : chaos21c@gmail.com**
**[ 제목 : id, 이름 (수정) ]**

# [My working folder]

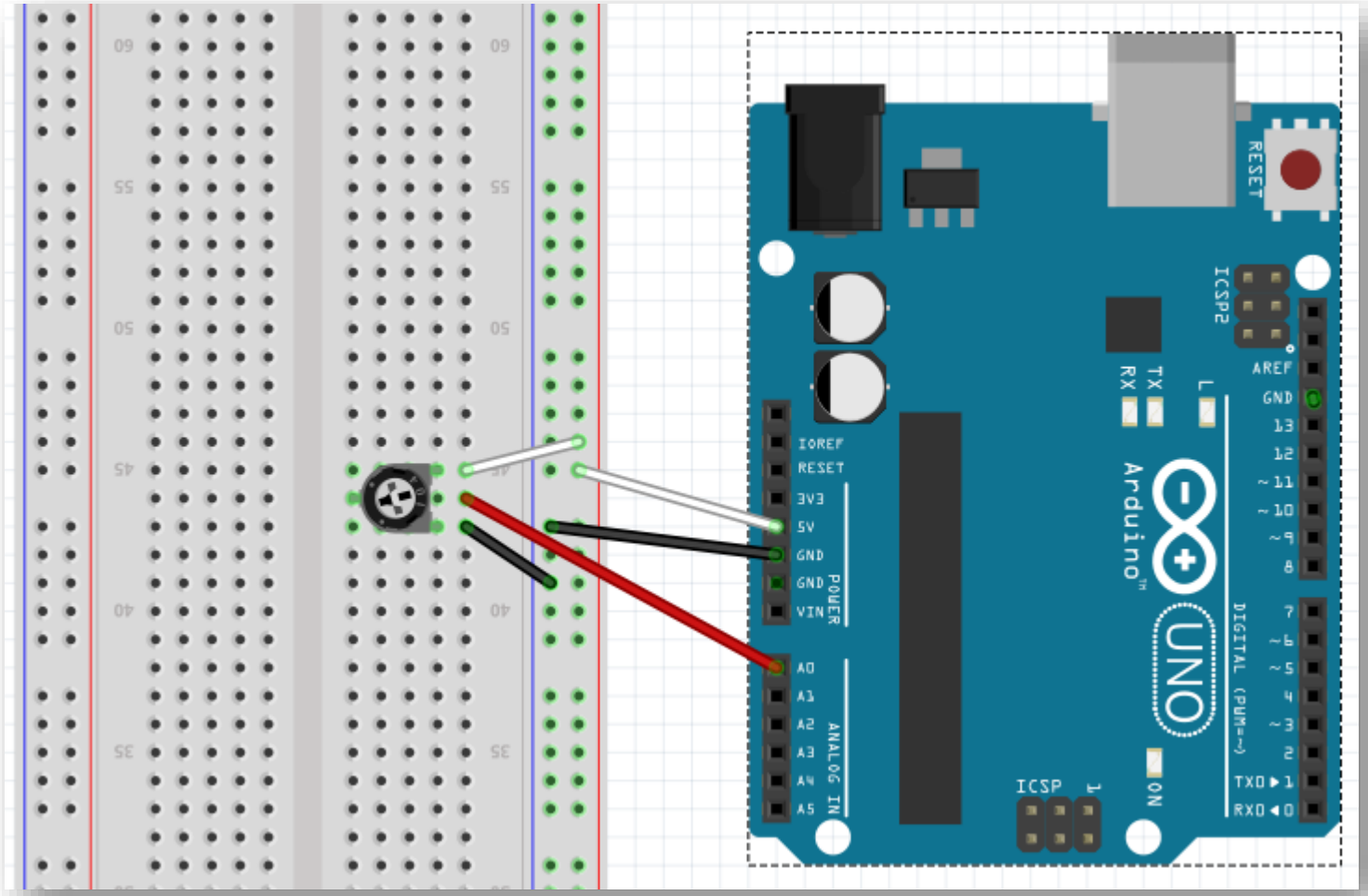





# Analog Signal

# A2.5.1 AnalogReadSerial (circuit)

**Standard potentiometer (가변 저항기)**



# A2.5.2 AnalogReadSerial (code)

▸ 스케치 구성 (코드 4-1)

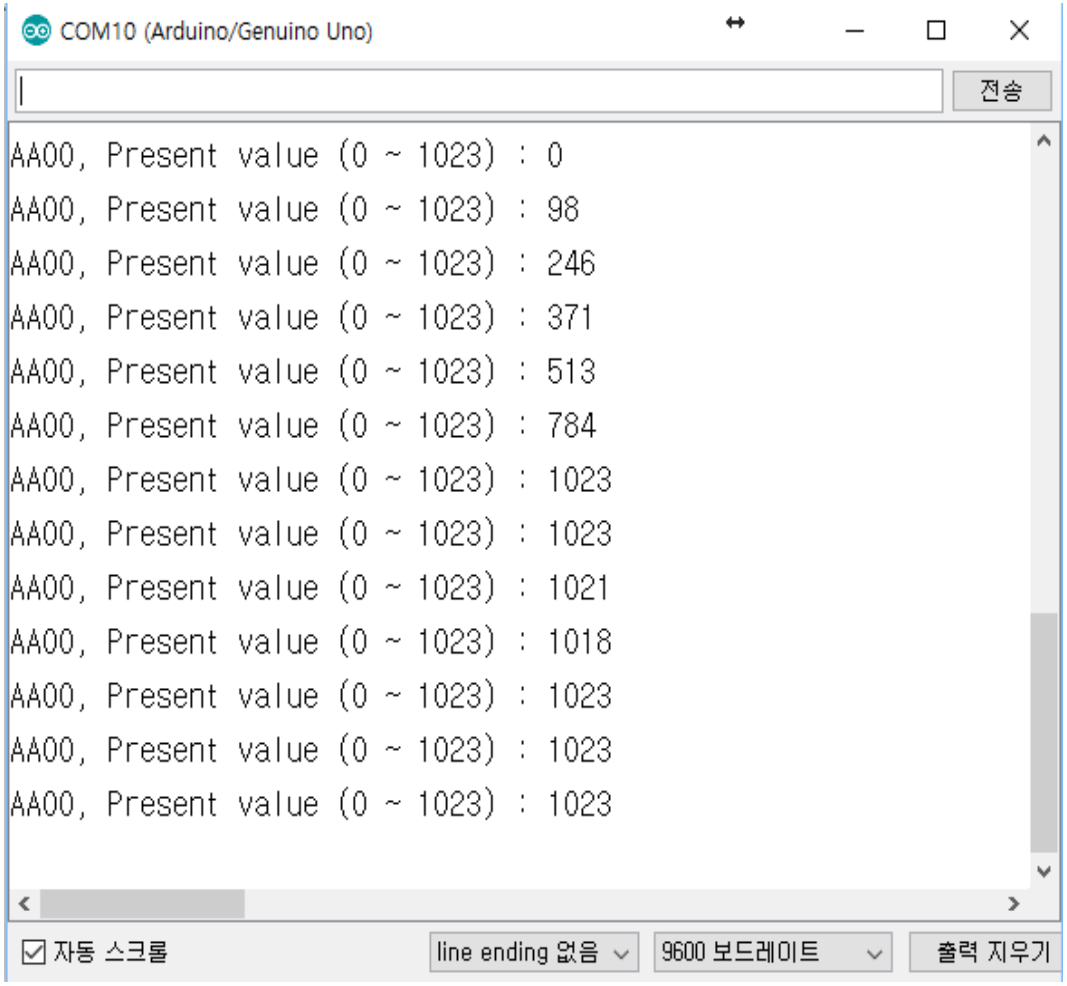
1. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
2. loop()에서 **analogRead**() 함수로 A0 핀에서 측정되는 값을 읽어 들인다.
3. 직렬 통신으로 A0 측정값을 한 줄로 0.5 초 마다 컴퓨터로 전송한다.

▸ 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 )

COM10 (Arduino/Genuino Uno)

전송

```
AA00, Present value (0 ~ 1023) : 0
AA00, Present value (0 ~ 1023) : 98
AA00, Present value (0 ~ 1023) : 246
AA00, Present value (0 ~ 1023) : 371
AA00, Present value (0 ~ 1023) : 513
AA00, Present value (0 ~ 1023) : 784
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1021
AA00, Present value (0 ~ 1023) : 1018
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1023
```

자동 스크롤 line ending

COM10 (Arduino/Genuino Uno)

전송

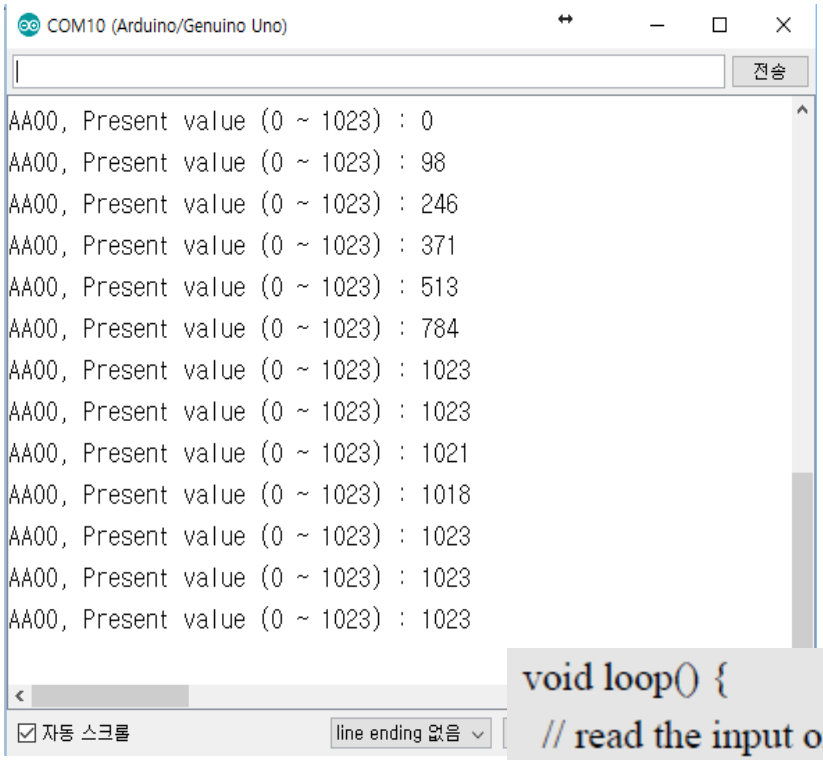
```
AA00, Present R (0 ~ 10.0) : 0.00
AA00, Present R (0 ~ 10.0) : 0.12
AA00, Present R (0 ~ 10.0) : 2.68
AA00, Present R (0 ~ 10.0) : 3.45
AA00, Present R (0 ~ 10.0) : 4.15
AA00, Present R (0 ~ 10.0) : 5.34
AA00, Present R (0 ~ 10.0) : 6.68
AA00, Present R (0 ~ 10.0) : 7.50
AA00, Present R (0 ~ 10.0) : 8.43
AA00, Present R (0 ~ 10.0) : 10.00
AA00, Present R (0 ~ 10.0) : 9.98
AA00, Present R (0 ~ 10.0) : 9.96
AA00, Present R (0 ~ 10.0) : 10.00
```

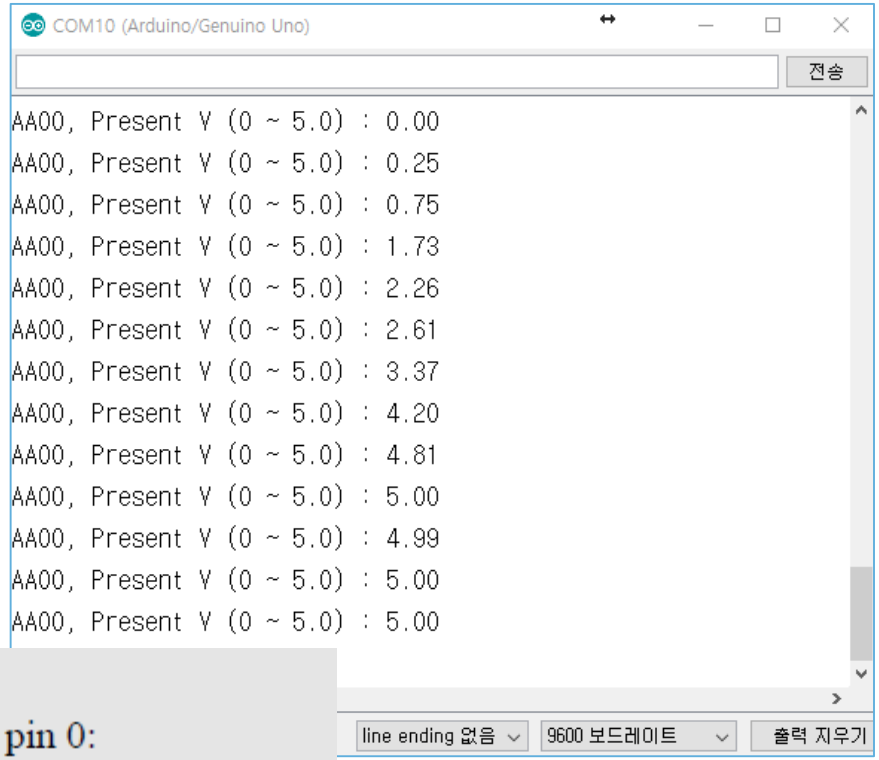
line ending 없음 9600 보드레이트 출력 지우기

```
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)



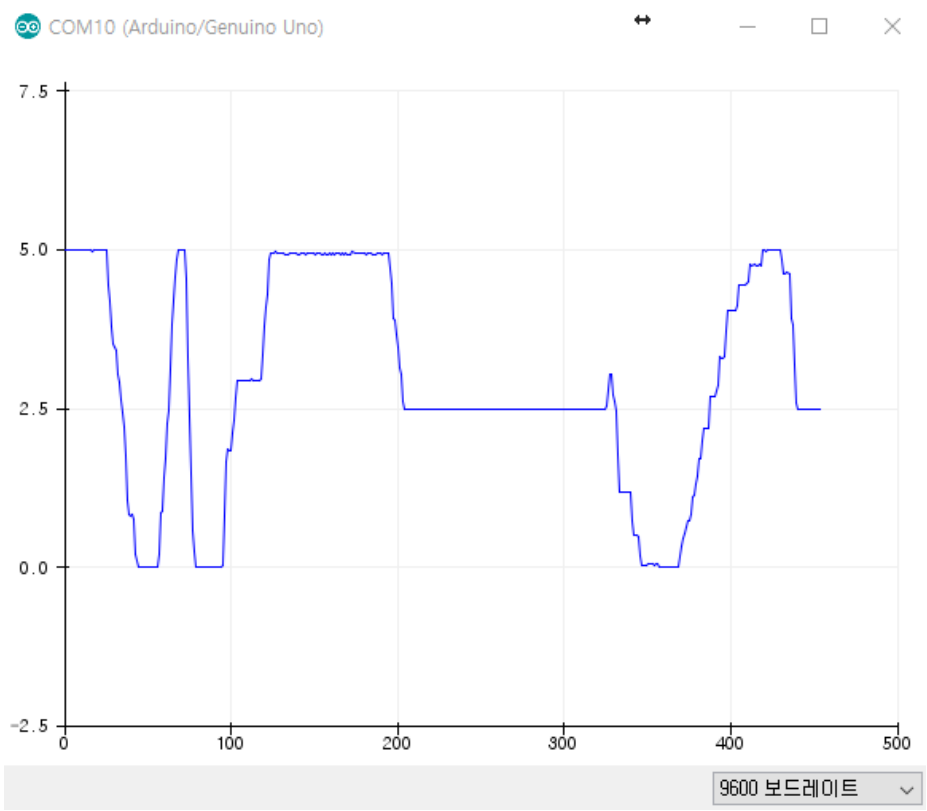

```
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.print("AA00, Present V (0 ~ 5.0) : ");
  float voltage= sensorValue*(5.0/1023.0);  // V
  Serial.println(voltage);
  delay(500);        // 2 Hz sampling
}
```

# A2.5.7 ReadAnalogVoltage

## Result

COM4

```
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

**HSnn_AnalogVoltage.png**

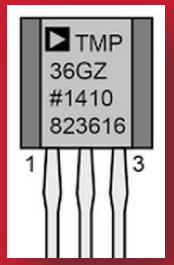
# A2.5.8 ReadAnalogVoltage using f_map()

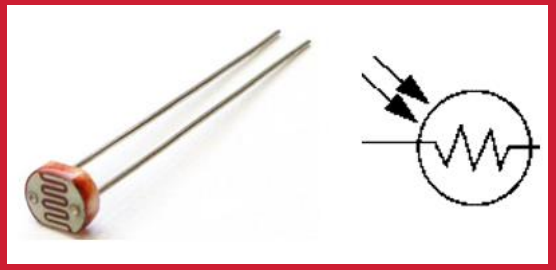
**Hint code : f_map() instead of map()**

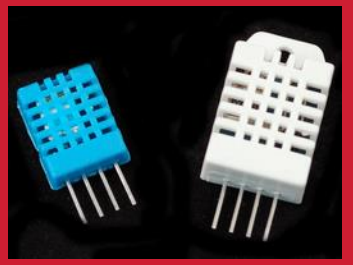
HS00_AnalogRead_fmap

```
// 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("HS00, 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;
}
```

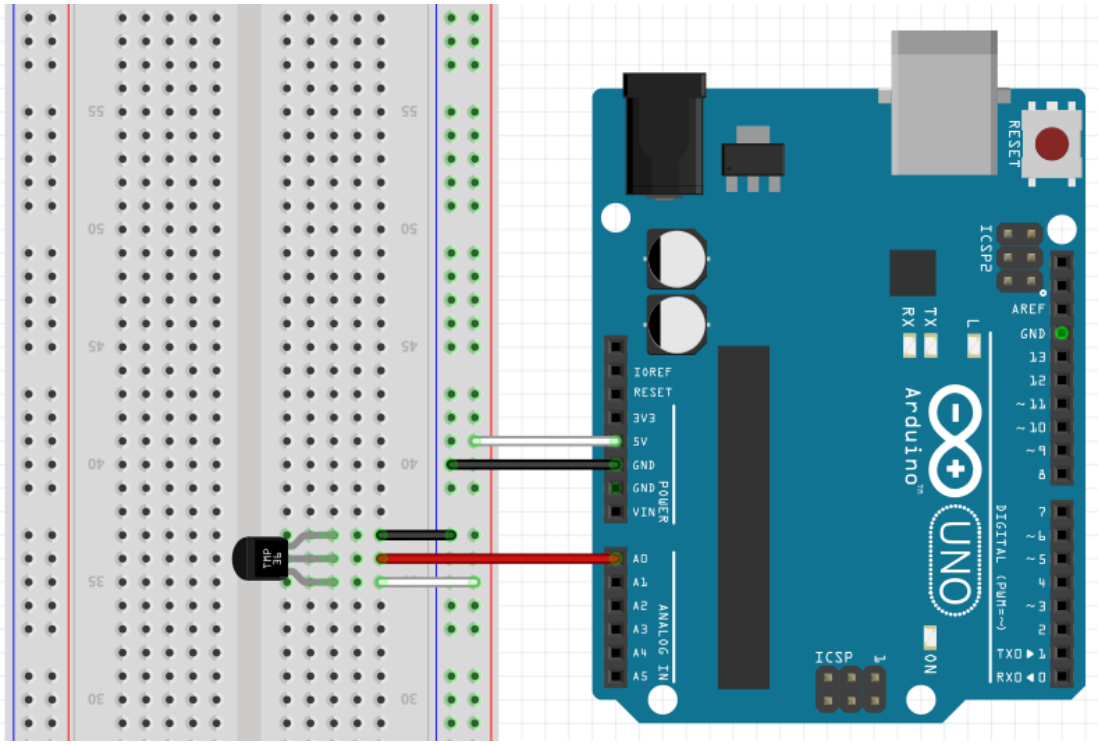

# Arduino Sensors

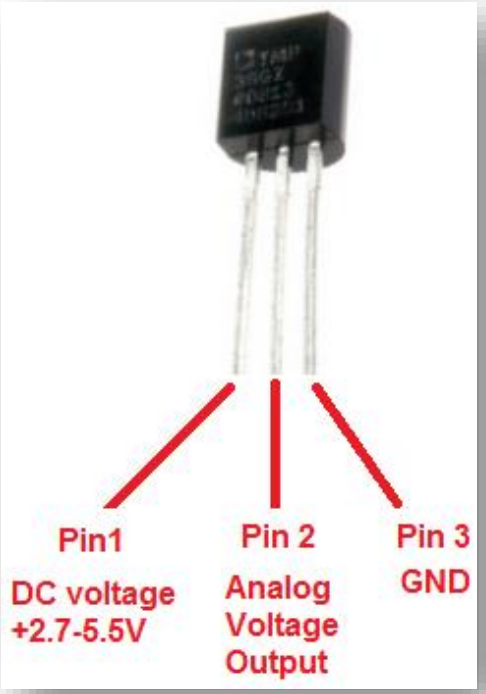


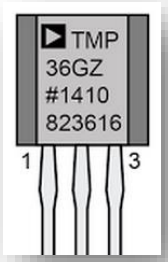


# A3.1.1 Temperature sensor [ TMP36]




## 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

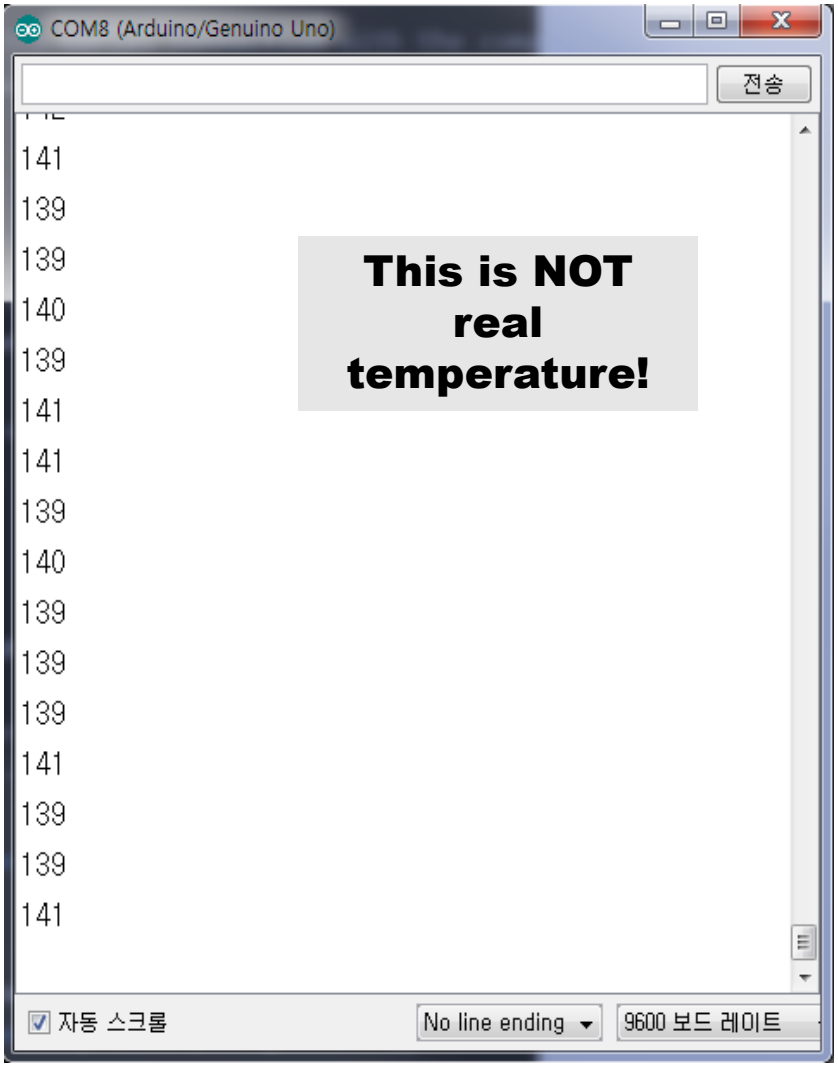
# A3.1.2 Temperature sensor [ TMP36]

## Simple code

TMP36 §

```
//
//    HS00, 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)



# A3.1.3 Temperature sensor [ TMP36]

## Sensor property

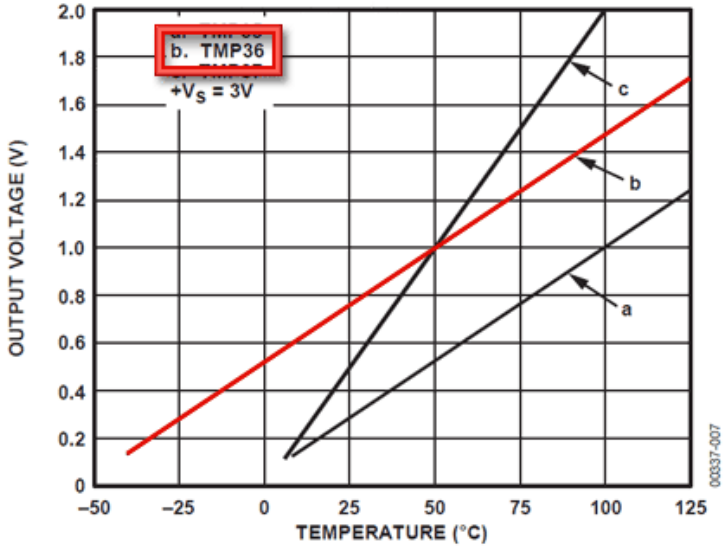



*Figure 6. Output Voltage vs. Temperature*

## Temperature conversion

**Temp (° C ) = (Vout – 500) / 10**

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

**( 0 <= value <= 1023)**

```
// 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

```
}

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);
}
```

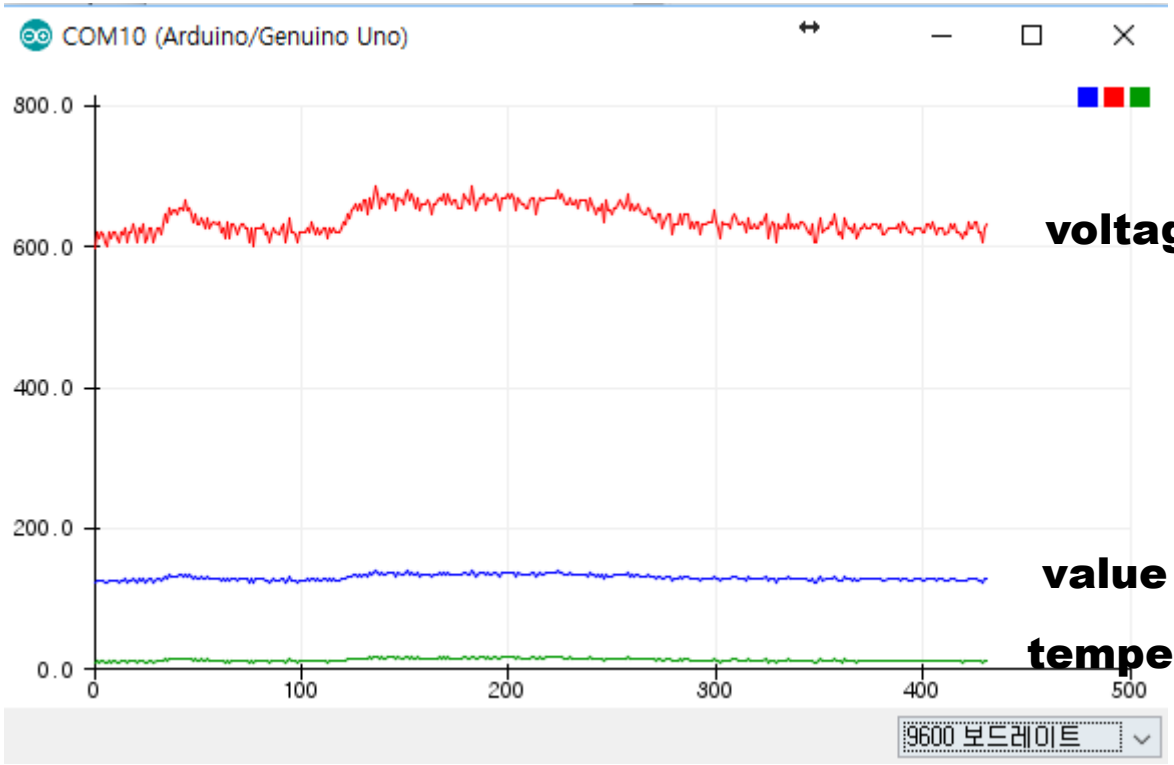
**Serial output ( ˚C)**

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
AA00, value
```

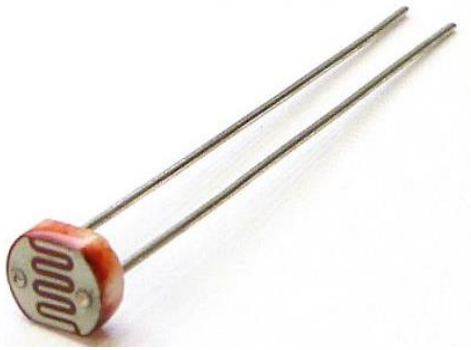
**Save as HSnn_TMP36.png**

# A3.1.5 Temperature sensor [ TMP36]

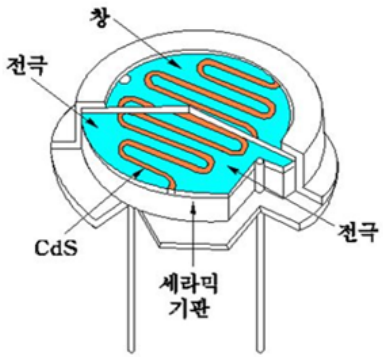


# A3.2 Luminosity sensor [ Photocell LDR]

## CdS 센서- photoresistor



### CDS특성

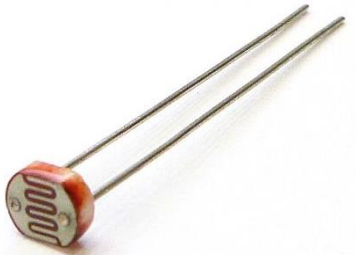



1. 감도
   - -빛의 파장에 따라 감도가 다름
2. 허용손실
   - -비교적 큰 전류를 흘릴 수 있음
3. 암 전류
   - -빛이 없어도 약간의 전류가 흐름
4. 명 전류
   - - 빛을 비추면 흐르는 전류
5. 응답특성
   - - 응답 시간 지연
   - - 빛의 세기에 따라 응답시간 다름
6. 가변저항
   - -빛에 따른 가변저항

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

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

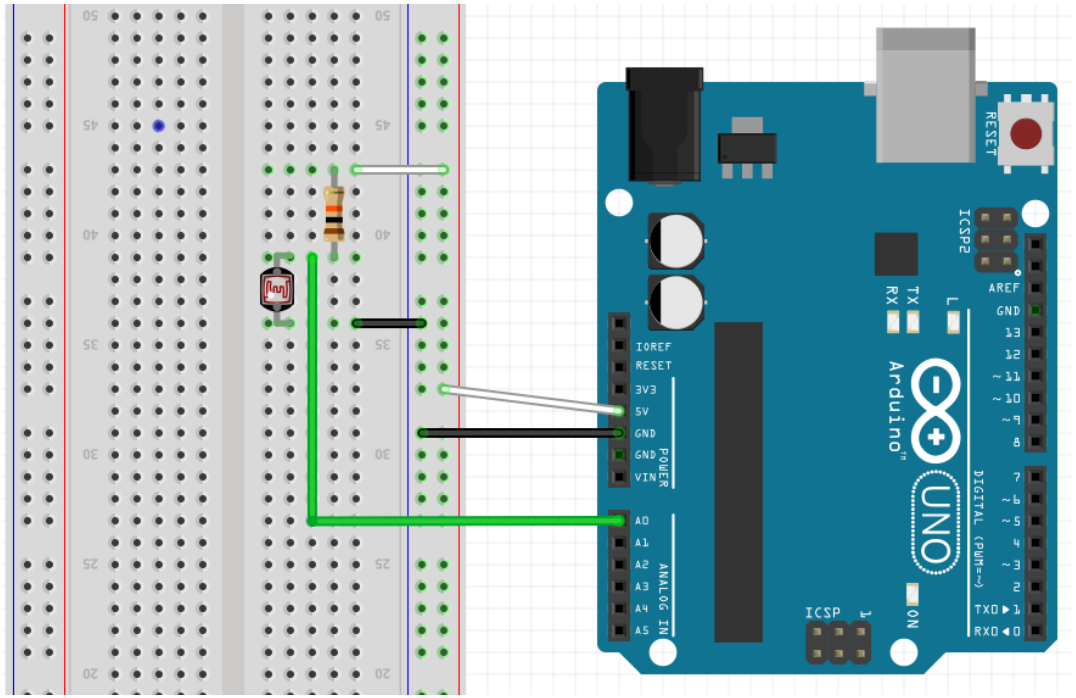
### 럭스의 예 [편집]

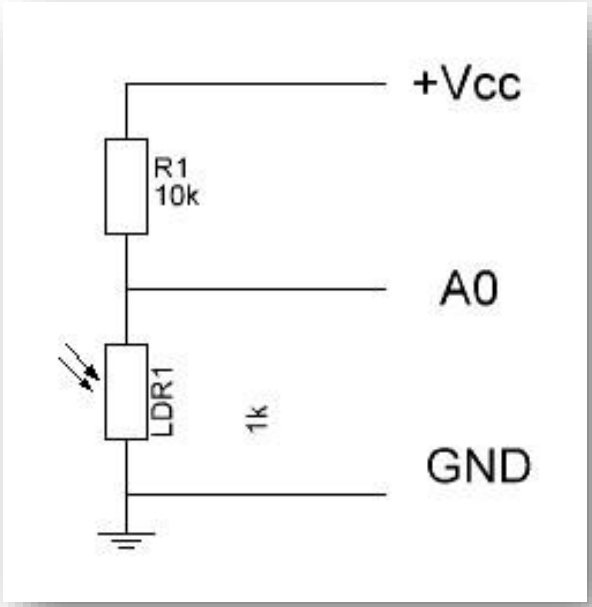
| 밝기차 | 예 |
|---|---|
| $10^{-5}$ lux | 가장 밝은 별(시리우스)의 빛[1] |
| $10^{-4}$ lux | 하늘을 덮은 완전한 별빛[1] |
| 0.002 lux | 대기광이 있는 달 없는 맑은 밤 하늘[1] |
| 0.01 lux | 초승달 |
| 0.27 lux | 맑은 밤의 보름달[1][2] |
| 1 lux | 열대 위도를 덮은 보름달[3] |
| 3.4 lux | 맑은 하늘 아래의 어두운 황혼[4] |
| 50 lux | 거실[5] |
| 80 lux | 복도/화장실[6] |
| 100 lux | 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

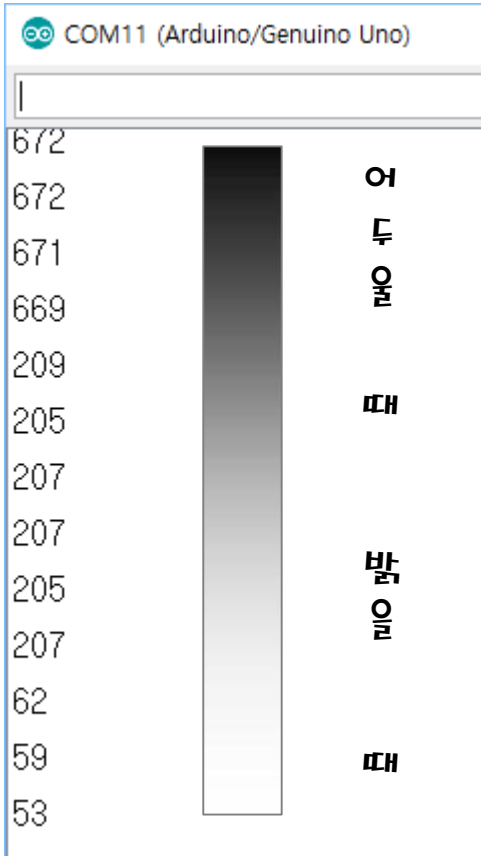
1. A0 핀을 CdS 조도 센서의 입력으로 설정한다.
2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
3. loop()에서 **analogRead()** 함수로 A0 핀에서 측정되는 값을 읽어 들인다.

# A3.2.4 Luminosity sensor [ Photocell LDR]

## CdS 센서 회로 - 측정 1.

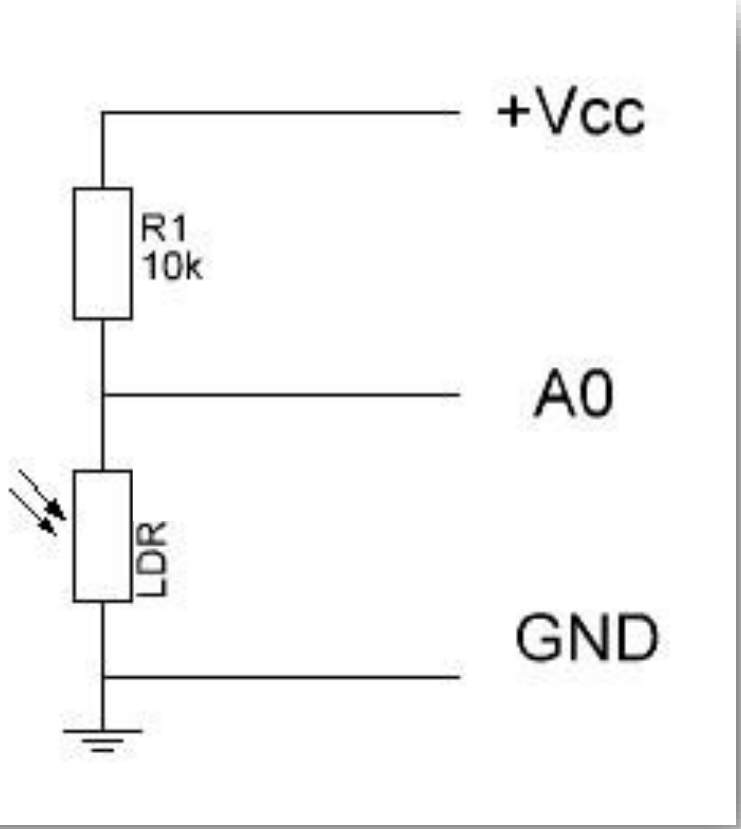
AAnn_CdS

```
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
```




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.**

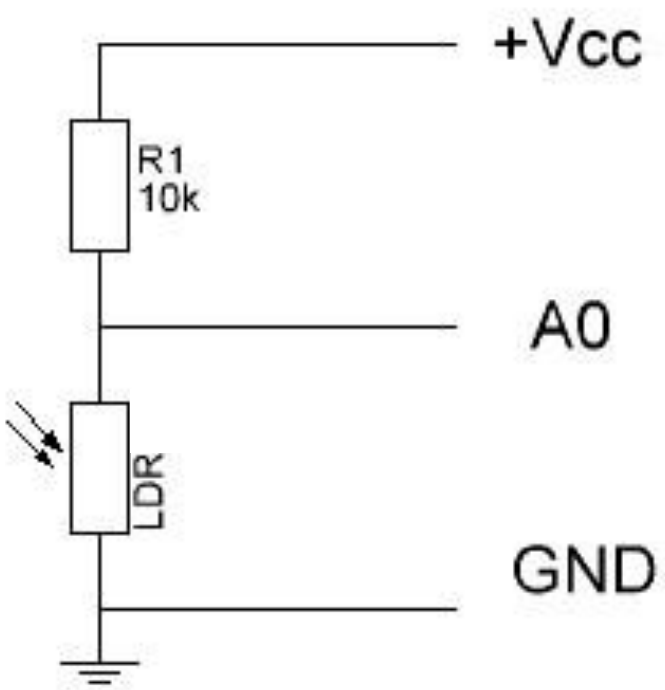
**If you would us the LDR as the lower part of a voltage divider, then in darkness there would be a high voltage over the LDR, while in bright light, there would be a low voltage over that resistor.**

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

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

**A0**에서 측정되는 **LDR** 양단의 전압 = **Vout**

# 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

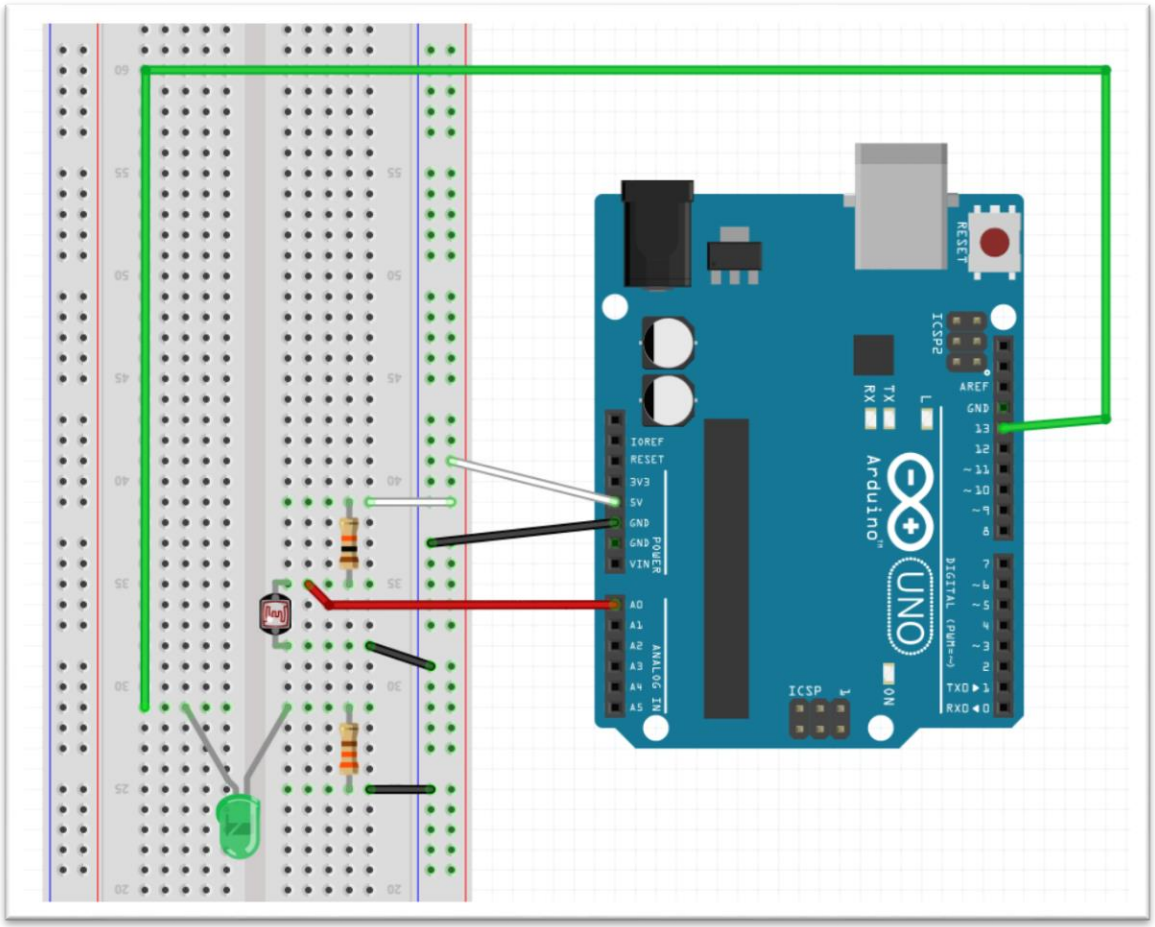
밝을 때

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.

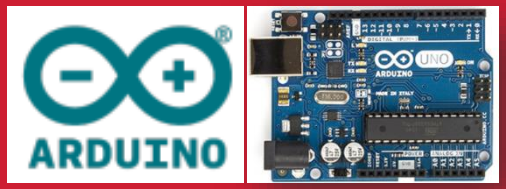
## DIY Code

Cds_LED

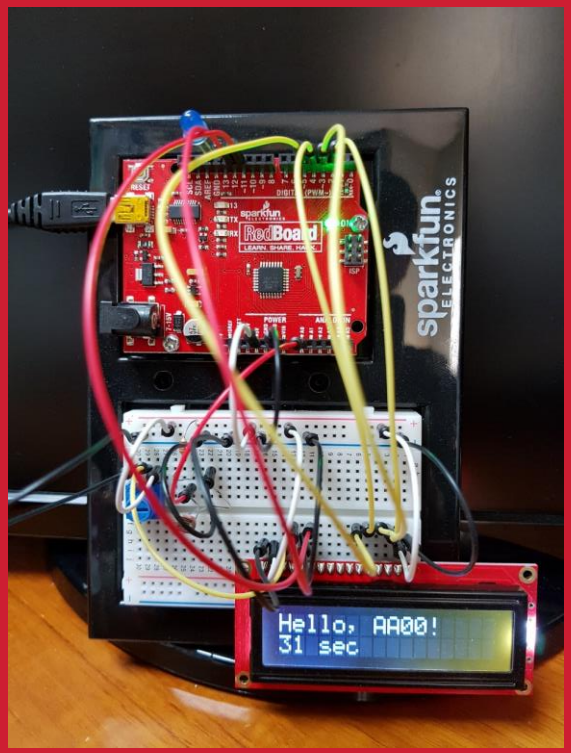
```
1  //  lux
2  #define CDS_INPUT 0
3  // LED pin
4  const int ledPin = 13;
5
6  int threshold = 70;
7
8  void setup() {
9    pinMode(ledPin, OUTPUT);
10   Serial.begin(9600);
11 }
```

```
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
27 //Voltage to LuxLux
28 double luminosity (int RawADC0){
29   double Vout=RawADC0*0.0048828125;  // 5/1024 (Vin = 5 V)
30   int lux=(2500/Vout-500)/10;  // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
31   return lux;
32 }
```

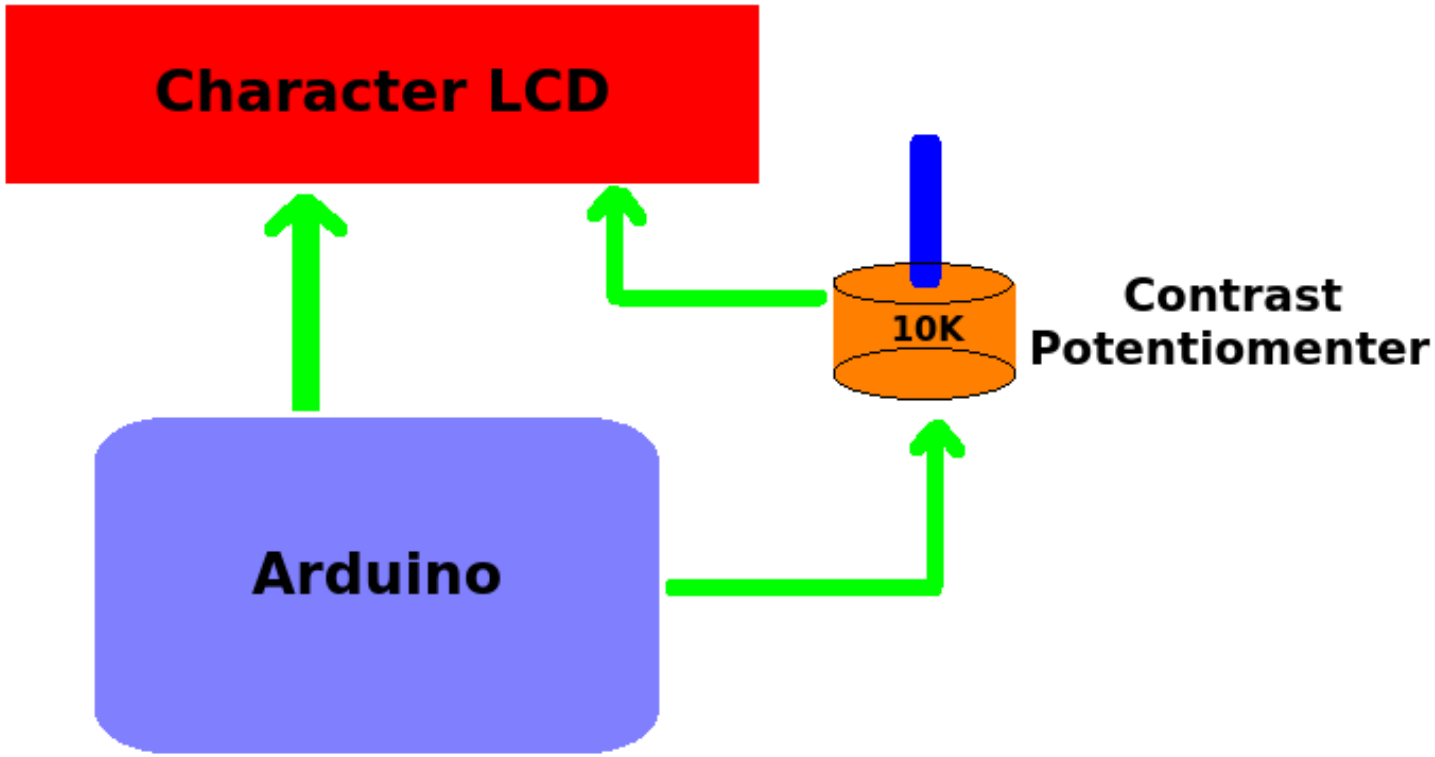
**HSnn_CdS_LED.ino**




# Signal Monitoring via LCD

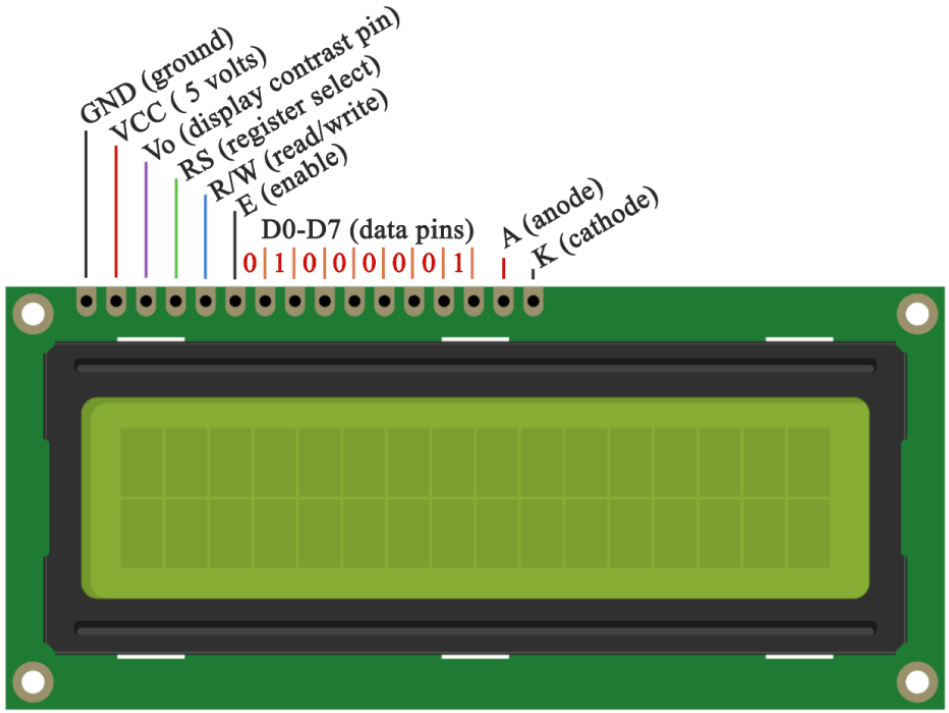



# Introduction to LCD



# Introduction to LCD

## LCD (Liquid Crystal Display, 16 X 2)



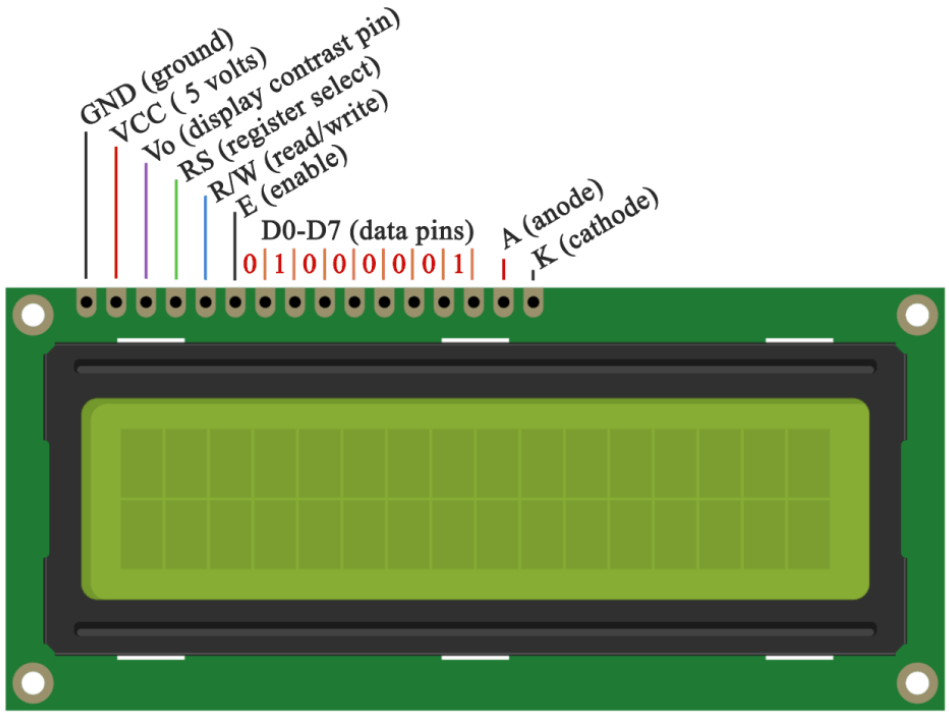

**(1,2, ... 15,16)**

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

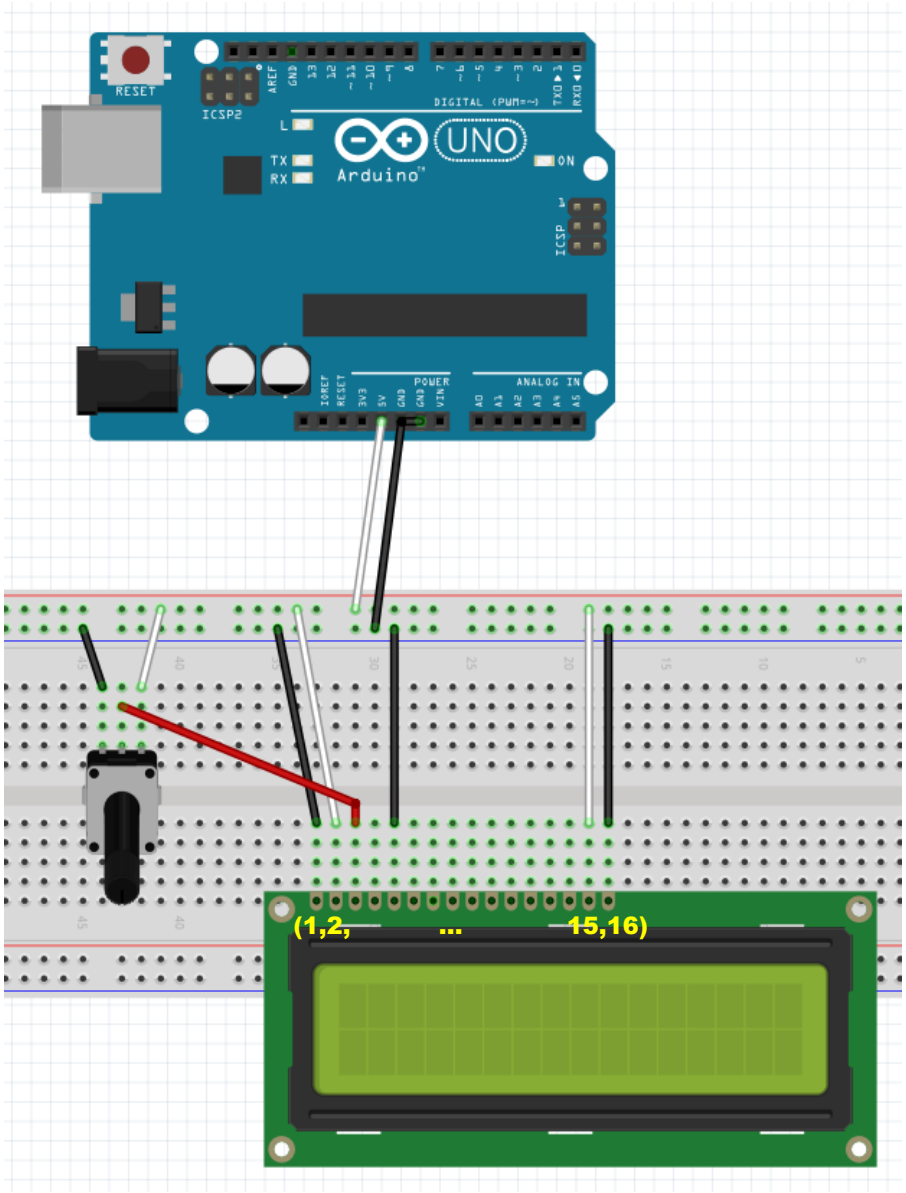
## LCD (Liquid Crystal Display, 16 X 2)




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

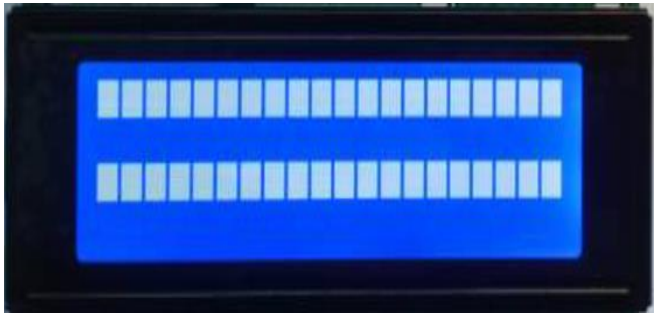
Pin 1 to Arduino GND
Pin 2 to Arduino +5V
Pin 3 to wiper
Pin 4 to Arduino pin D12
Pin 5 to Arduino GND
Pin 6 to Arduino pin D11
Pin 11 to Arduino pin D5
Pin 12 to Arduino pin D4
Pin 13 to Arduino pin D3
Pin 14 to Arduino pin D2
Pin 15 to +5V (with 220 or 330 Ω)
Pin 16 to GND

# LCD 초기화 (pin-1, 2, 3, 5, 15,16)

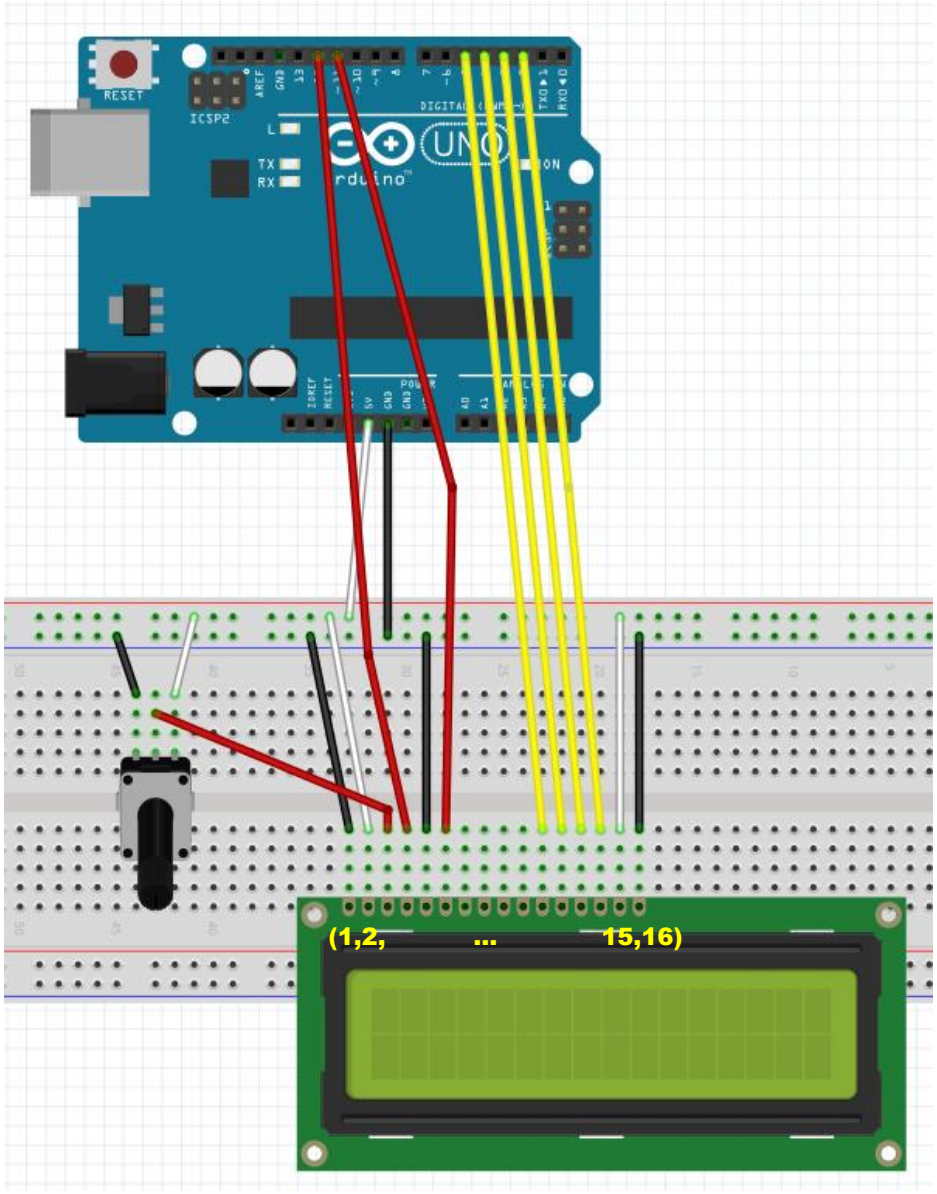



Pin 1 to Arduino GND
Pin 2 to Arduino +5V
Pin 3 to wiper (potentiometer)
Pin 5 to Arduino GND
Pin 15 to +5V
Pin 16 to GND

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

# Introduction to LCD - code “Hello AAnn”

- LiquidCrytral lcd(rs, en, d4, d5, d6, d7)

lcd란 이름으로 I2C에 연결된 LCD 모듈 객체.

- lcd.begin(행, 열)

lcd란 이름의 LCD 모듈의 크기를 정의한다.

- 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();

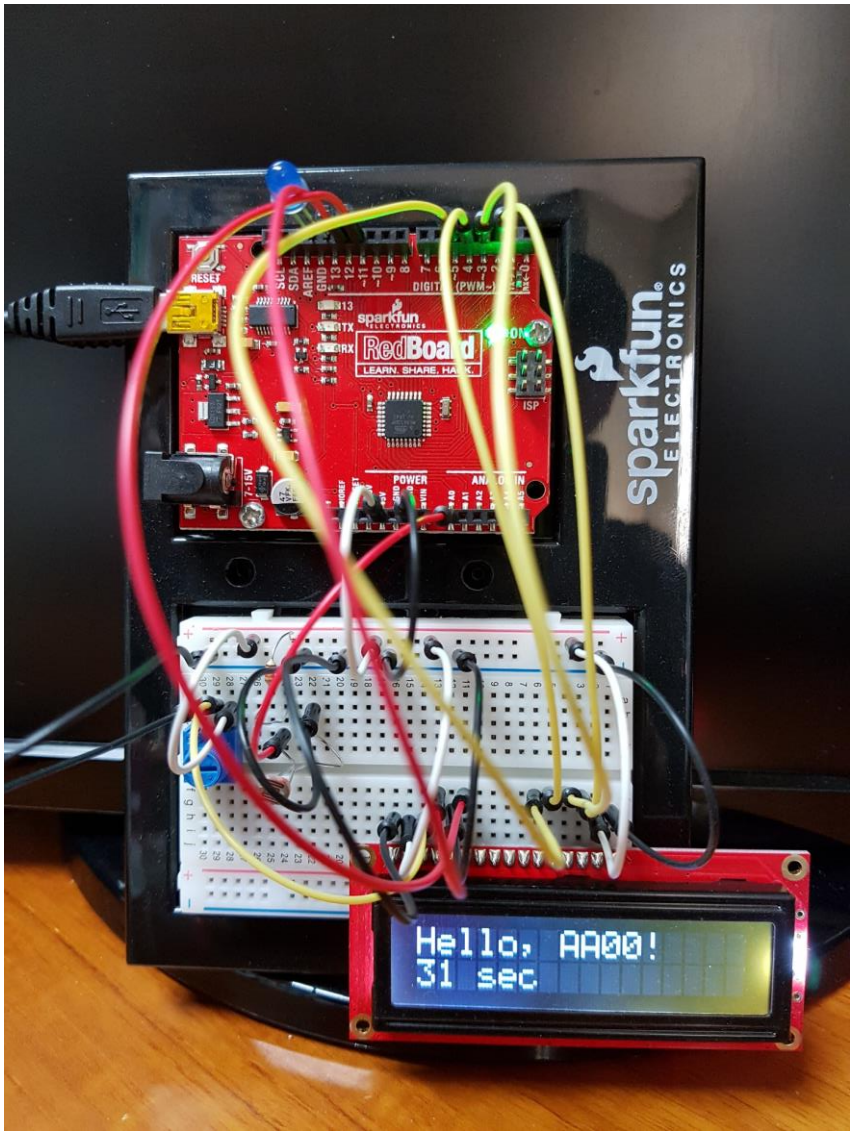
lcd란 이름의 LCD 모듈의 백라이트를 점등한다.

# Introduction to LCD - code “Hello HSnn”

sketch09_hello_LCD

```
6
7  // include the library code:
8  #include <LiquidCrystal.h>
9
10 // initialize the library with the numbers of the interface pins
11 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
12
13 void setup() {
14   // set up the LCD's number of columns and rows:
15   lcd.begin(16, 2);
16   // Print a message to the LCD.
17   lcd.print("Hello, HS00!");
18 }
19
20 void loop() {
21   // set the cursor to column 0, line 1
22   lcd.setCursor(0, 1);  // second line, first column
23   // print the number of seconds since reset:
24   lcd.print(millis() / 1000);
25   lcd.print(" sec");
26 }
```

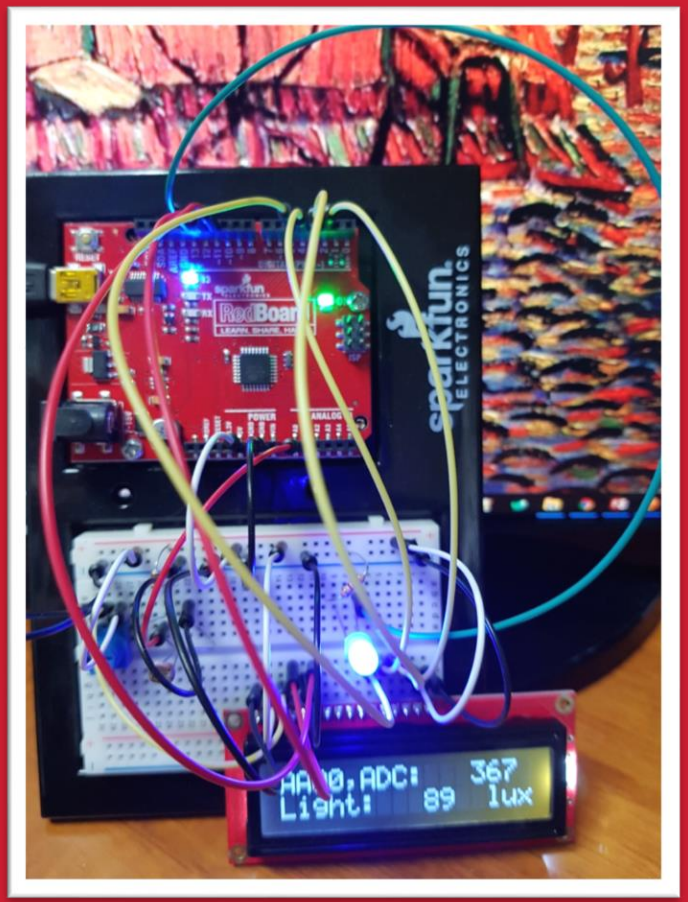
# Introduction to LCD - “Hello HSnn”



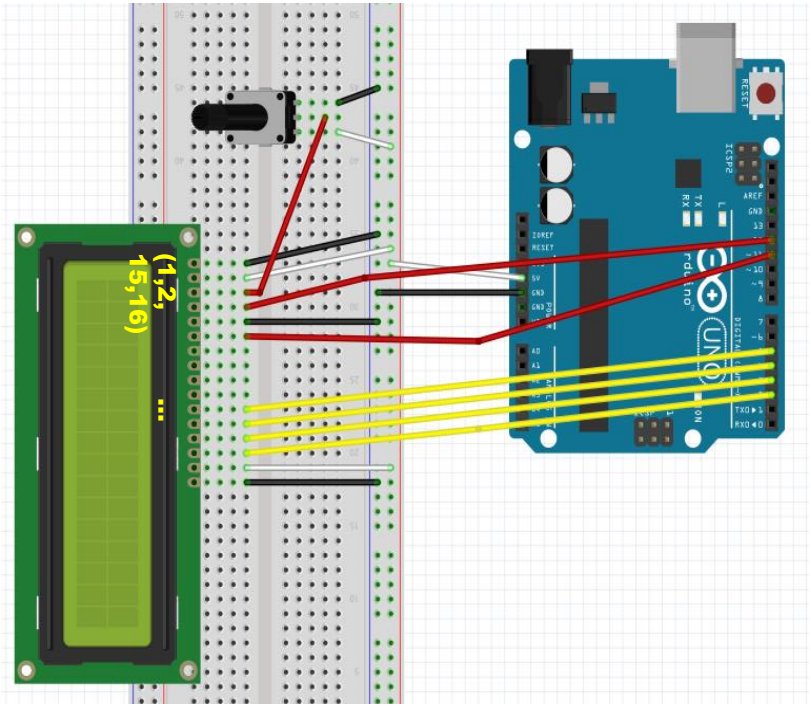

결과 화면 촬영: HSnn_Hello_LCD.png 로 저장...

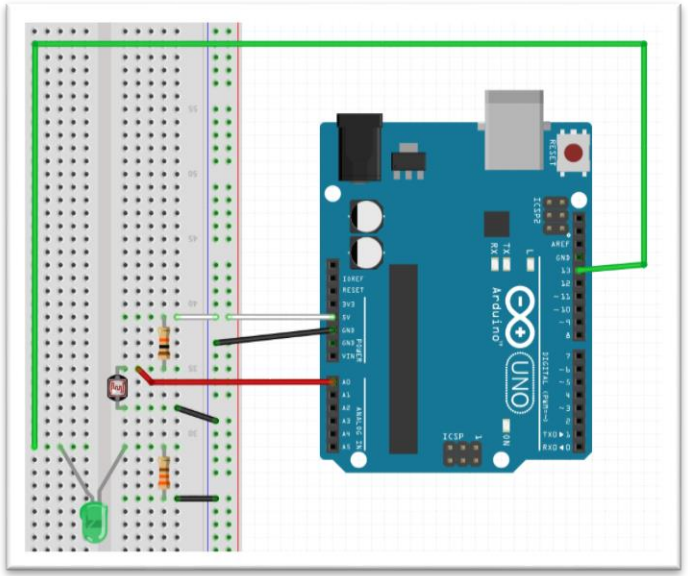
# CdS LCD Project

LCD에 조도 값을
표시하면서
조도에 따라 LED를
ON/OFF



# CdS-LCD project



# CdS-LCD project

## Set CdS-LCD project

**Project**

CdS 셀을 이용하여 조도를 측정해 보자.

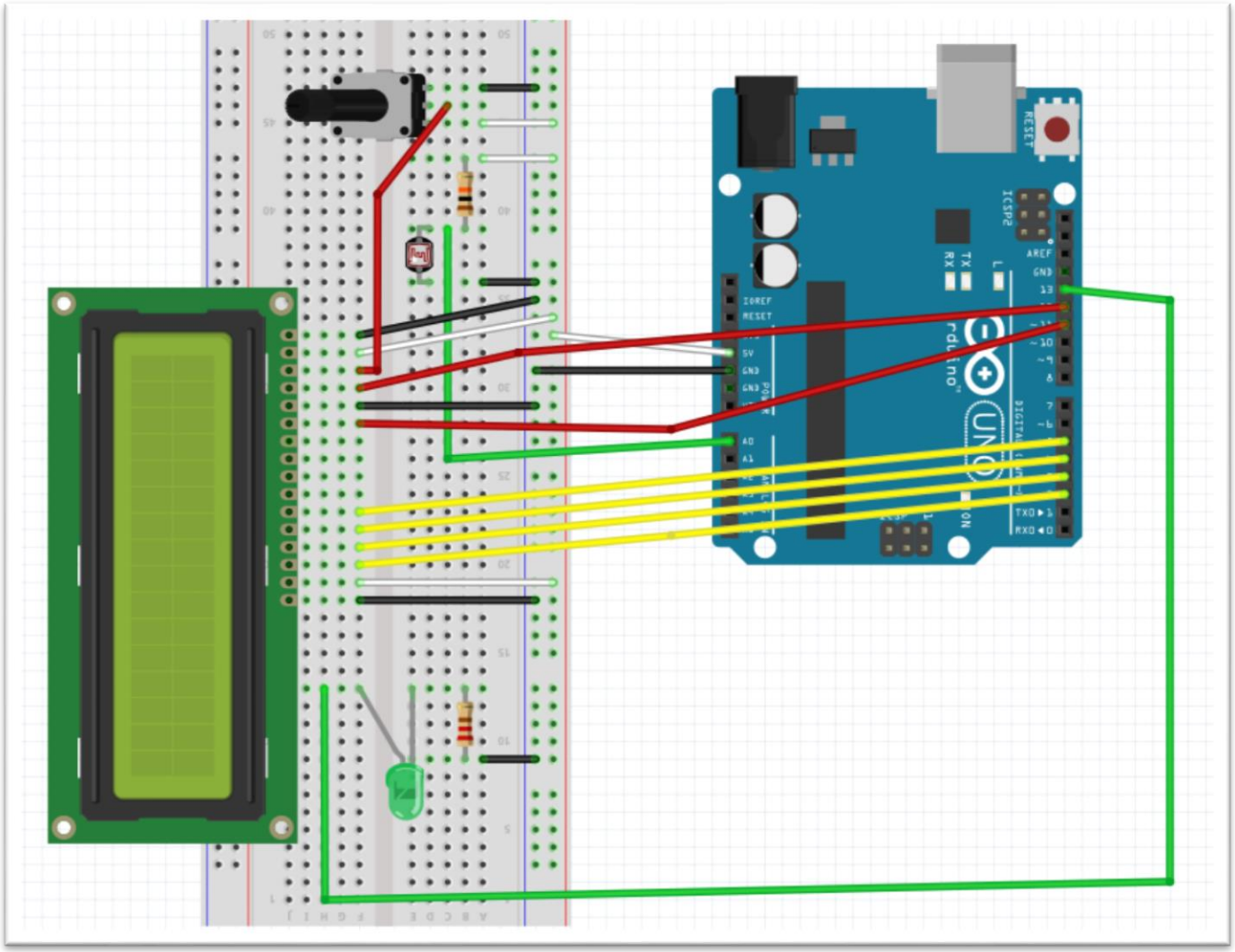
1. CdS 셀로 측정된 조도를 아날로그 핀을 통하여 0~1023 범위로 읽는다.
2. ADC 값을 LCD 모듈로 lux로 출력한다. (빛의 밝기)
3. lux 값에 따라 D13에 연결된 단색 LED의 ON/OFF를 조정한다.

**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 : fzz circuit

circuit06_CdS_LCD_LED.fzz



# CdS-LCD project : new code

## CdS 센서 LCD 회로 - code: HSnn_LCD_lux.ino

sketch10_CdS_LCD_lux_start

```
// LCD 라리브러리 설정
#include <LiquidCrystal.h>
// LCD 설정
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //  rs,en,d4,d5,d6,d7
// 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.begin(16,2);
// 모든 메세지를 삭제한 뒤
// 숫자를 제외한 부분들을 미리 출력시킨다.
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("HS00,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,**

**HSnn_LCD_lux.ino**

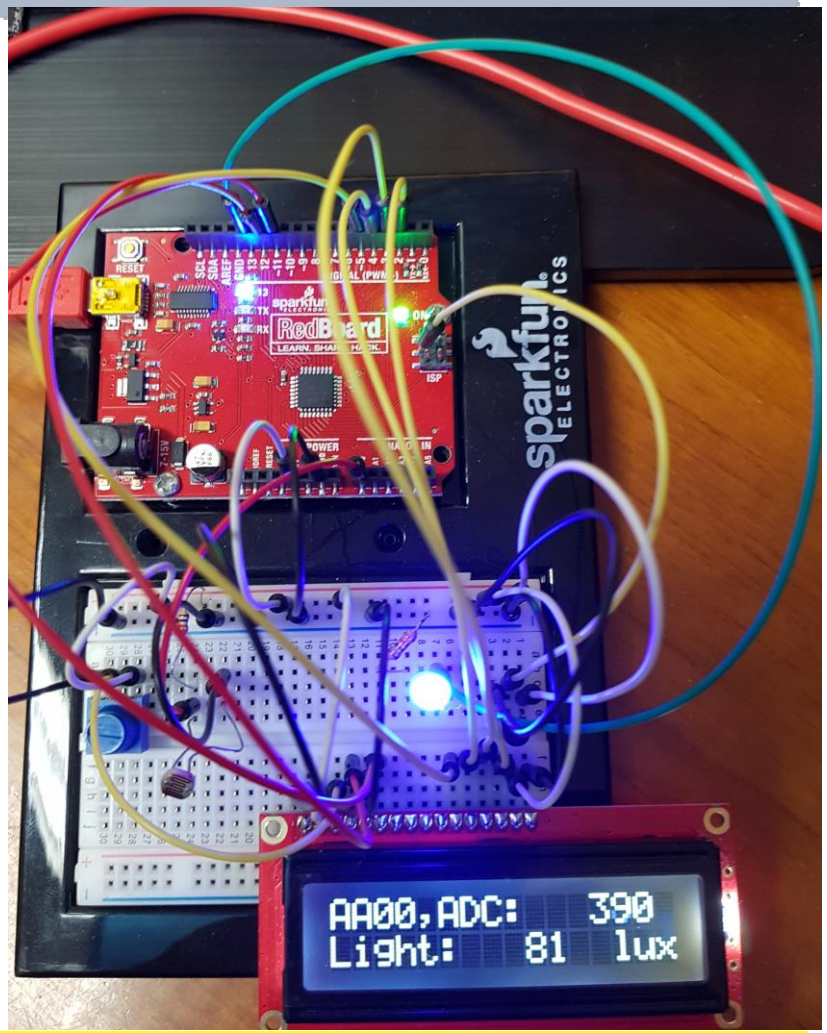
로 저장...

# CdS-LCD project : result

## CdS 센서 LCD 회로 - 측정 결과

주변의 조도에 따라
어두우면 **LED**가
켜지고, 밝으면
**LED**가 꺼지도록
코드를 수정하시오.

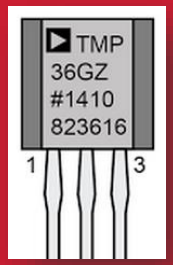
**LED**가 켜진
화면을 폰으로
촬영해서 그림을
제출하시오.

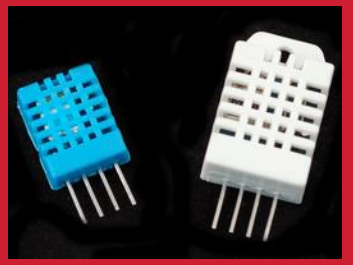


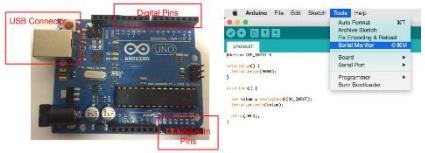

조도에 따라 **LED**가 **ON/OFF** 되는 것을 확인 받고
결과 화면 촬영: **HSnn LCD lux.png** 로
저장...

# Next week


# Arduino & Node.js






# IOT: HSC

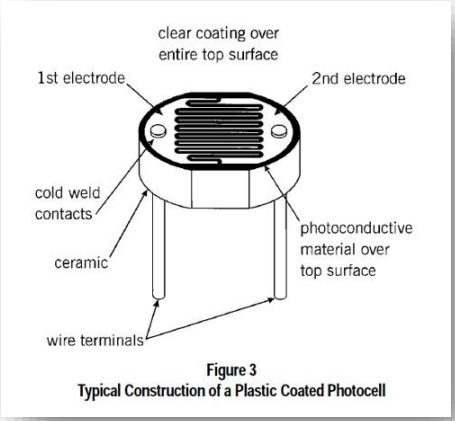
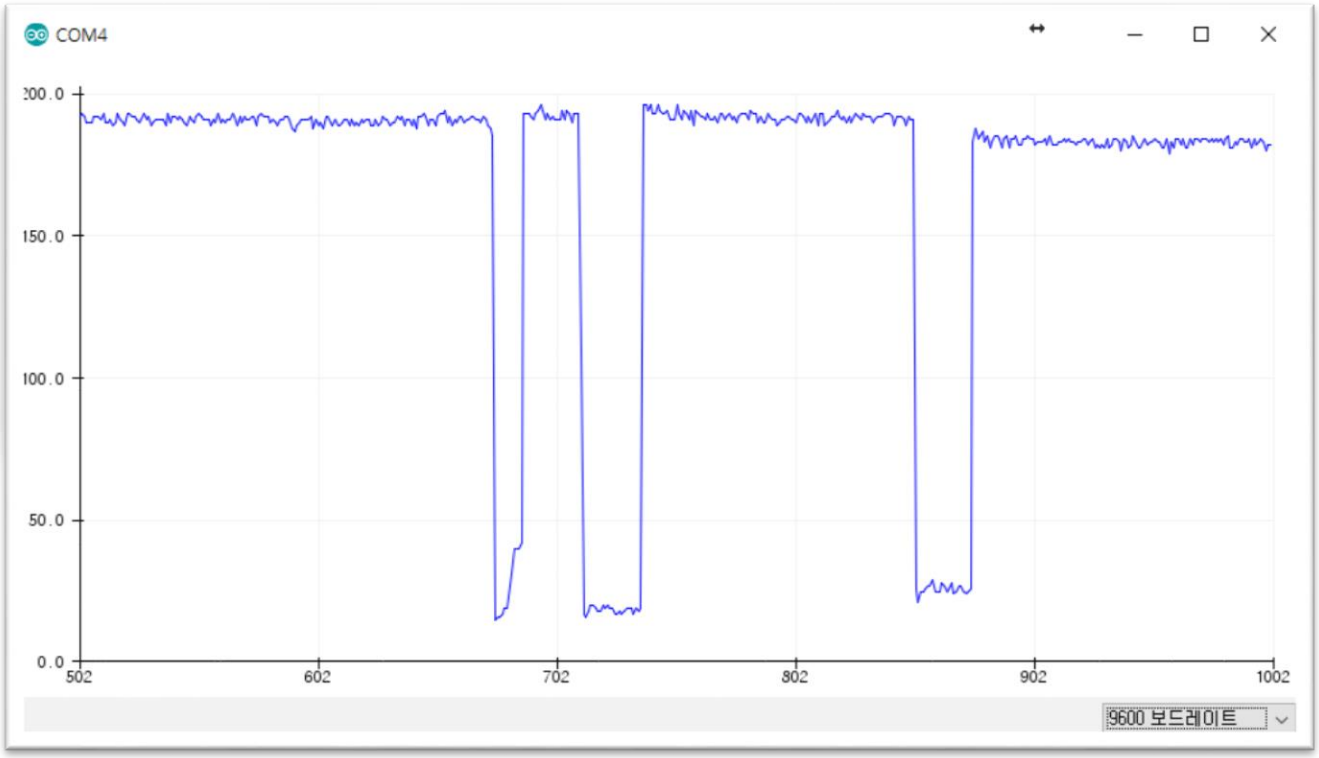
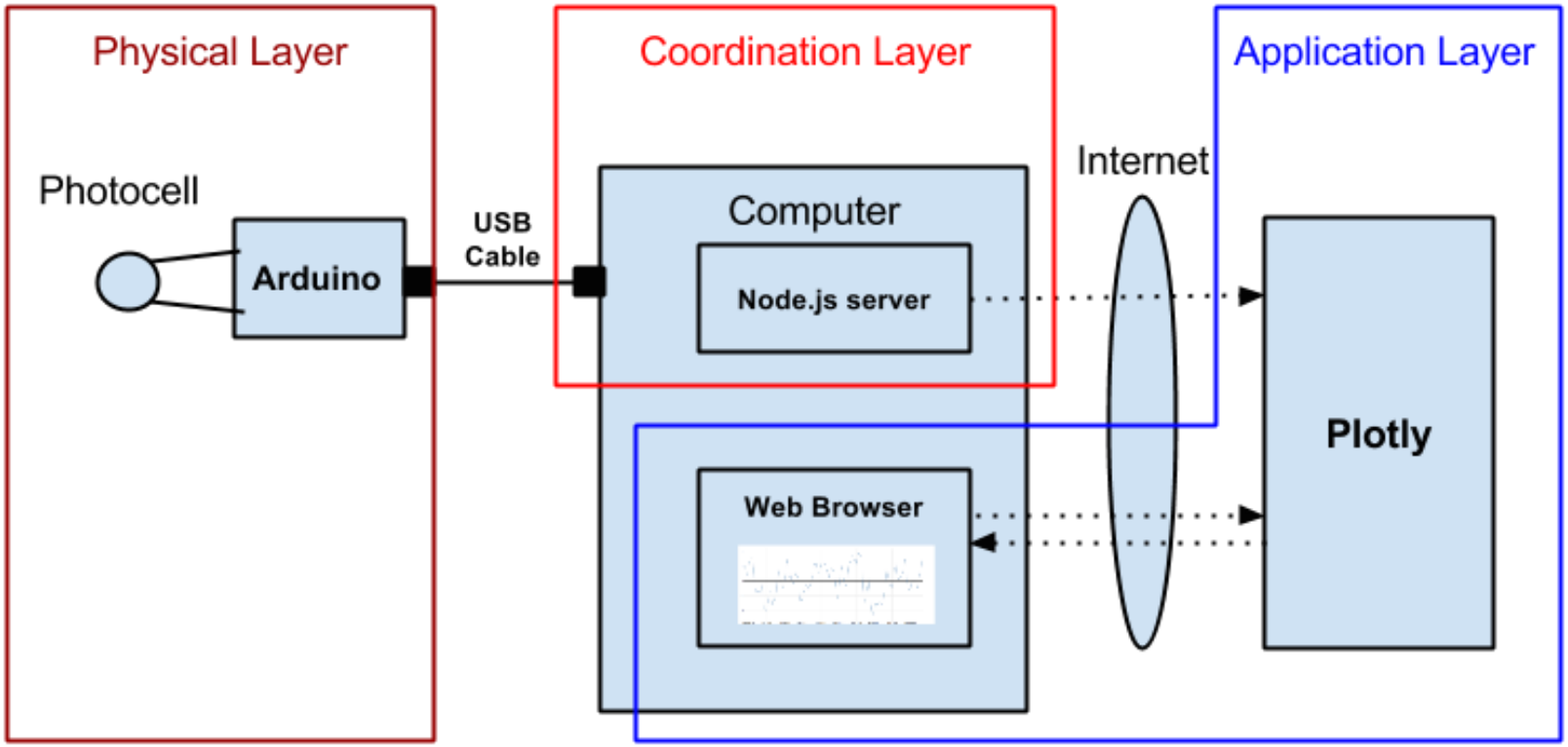


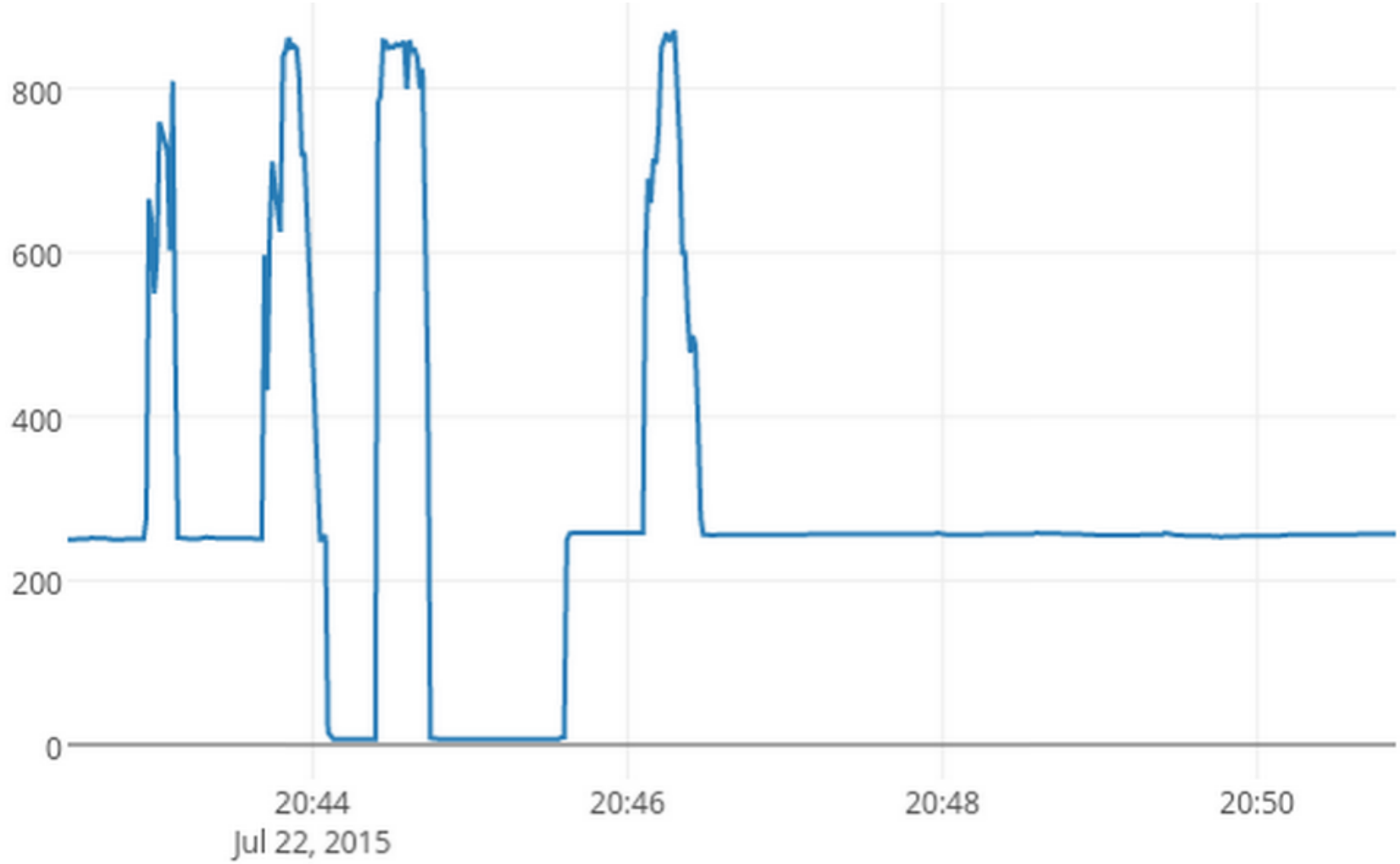
Figure 3
Typical Construction of a Plastic Coated Photocell



# Layout [H S C]


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

# Arduino: node.js + plotly



# [Practice]

- [wk05]
  - Arduino sensors
  - Complete your project
  - Submit file : HSnn_Rpt04.zip

# wk05 : Practice-04 : HSnn_Rpt04.zip

- [Target of this week]
  - Complete your works
  - Save your outcomes and compress 5 outputs

제출파일명 : **HSnn_Rpt04.zip**

- 압축할 파일들

① **HSnn_AnalogVoltage.png**

② **HSnn_TMP36.png**

③ **HSnn_Hello_LCD.png**

④ **HSnn_LCD_lux.ino**

⑤ **HSnn_LCD_lux.png**

**Email : chaos21c@gmail.com**

**[ 제목 : id, 이름 (수정) ]**

# Lecture materials

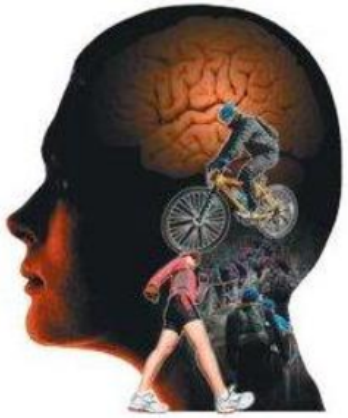
## References & good sites

- ✓ 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

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**

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

# Github.com/Redwoods/healthcare-signal-iot


Redwoods / Lec

Unwatch 1

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

All lectures by Redwoods in Inje University

Add topics

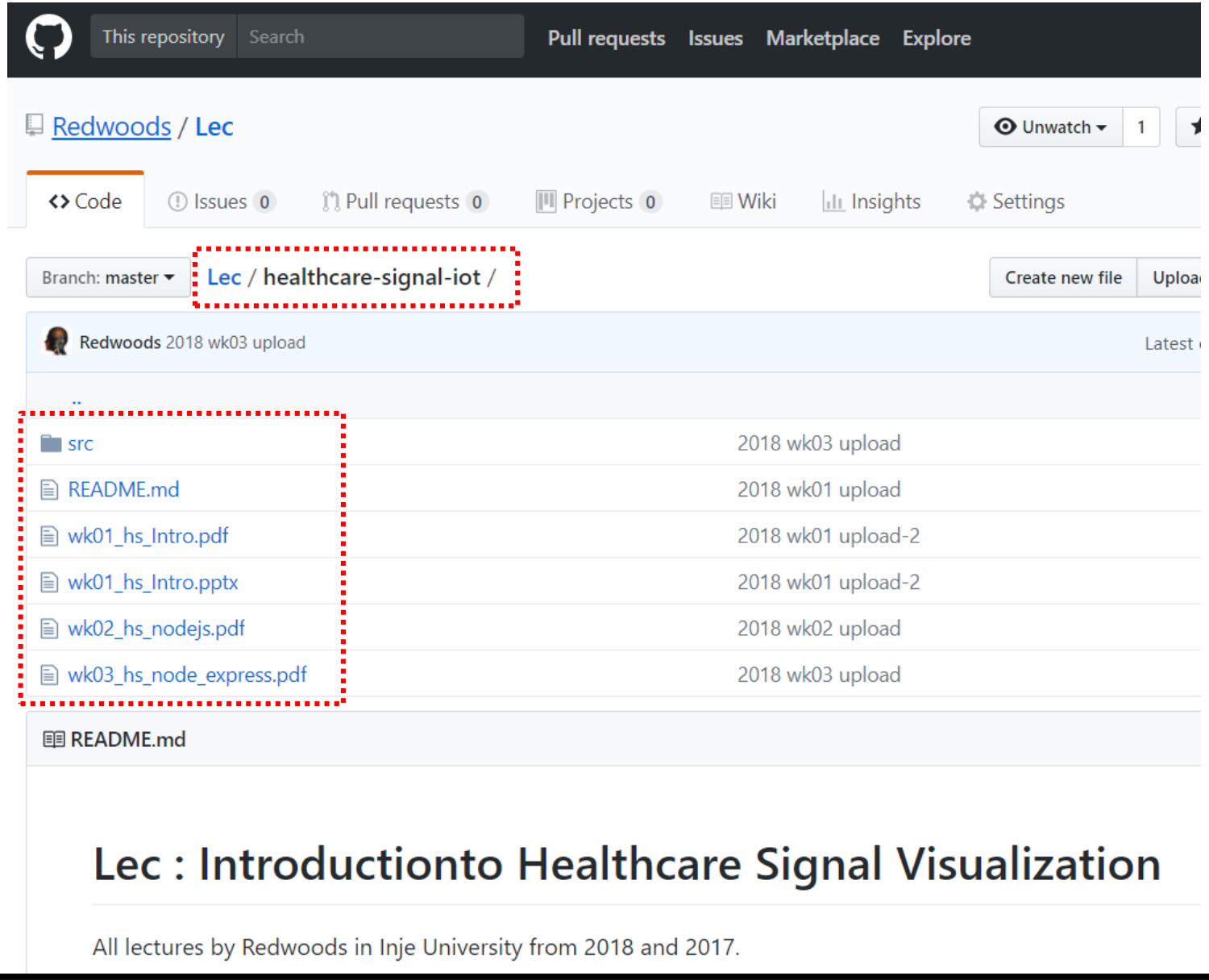
81 commits | 1 branch | 0 releases

Branch: master | New pull request | Create new file | Upload files

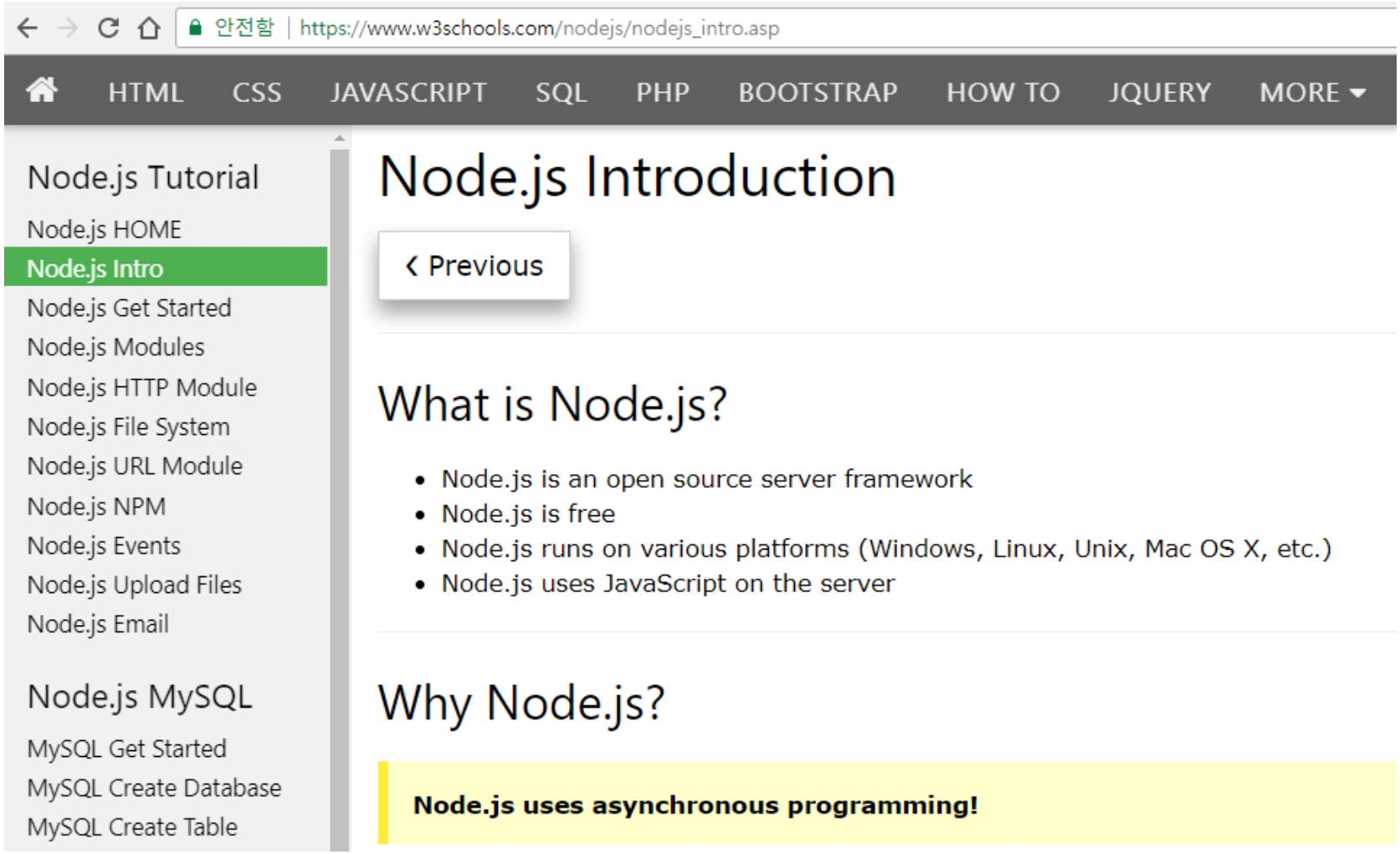
Redwoods 2018 wk01 upload

| | |
|---|---|
| advanced-Arduino-iot | wk16 exam upload |
| ev3 | wk16 final exam. answers |
| healthcare-signal-iot | 2018 wk01 upload |
| html5-basic | 2018 wk01 upload |
| html5-mobile-simulation | wk15 lec upload |
| Lec.Rproj | 2018 wk01 upload |
| README.md | wk03 upload and fix links |

# Github.com/Redwoods/healthcare-signal-iot

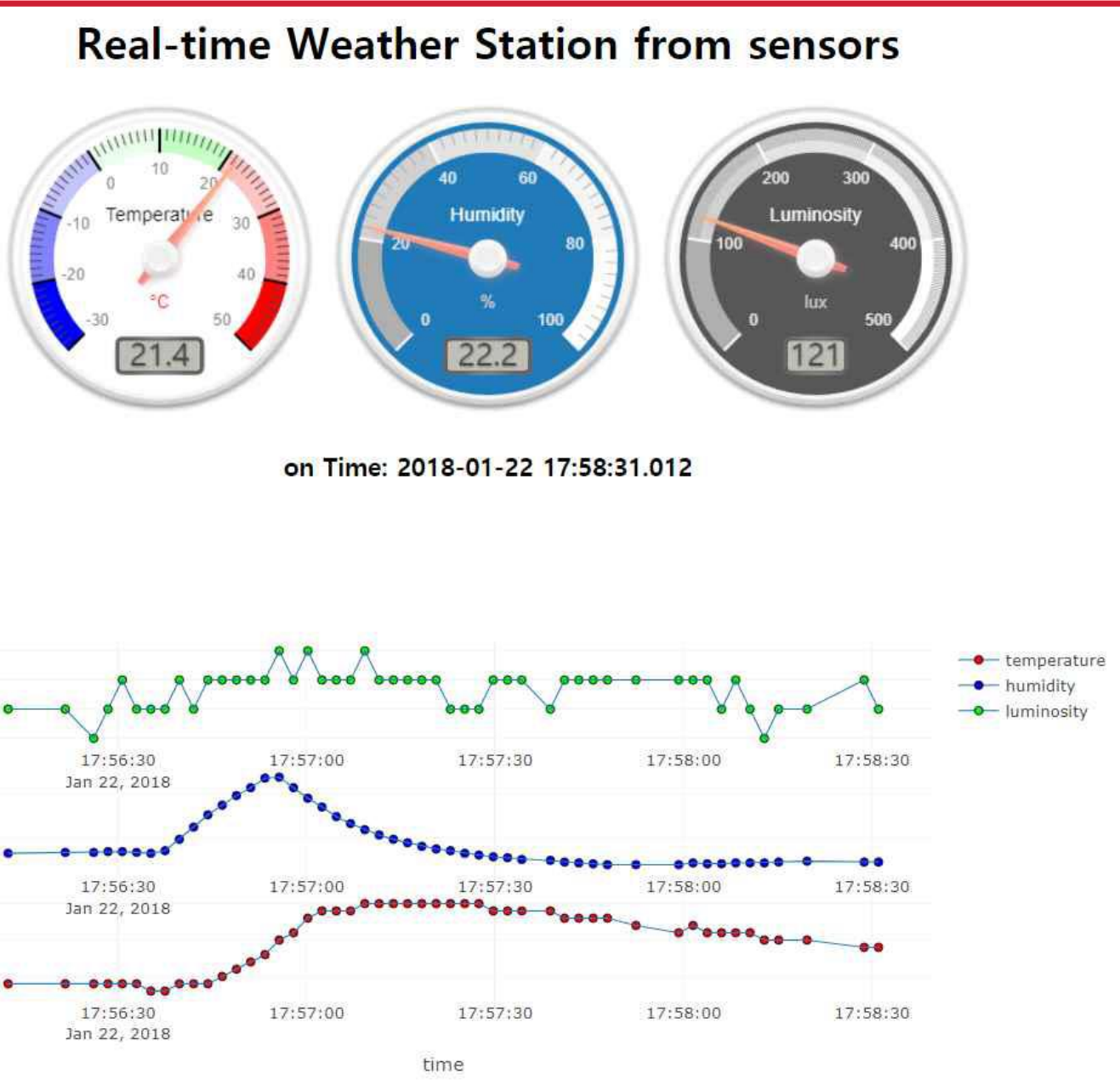


# 1.0 What is node.js?




https://www.w3schools.com/nodejs/nodejs_intro.asp

# Target of this class



# Project of this class