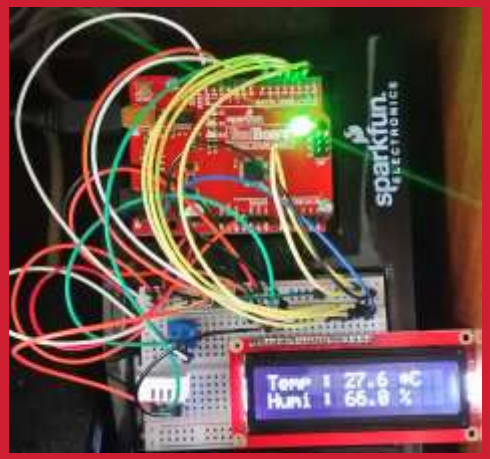


Arduino-IOT

[wk05]

Arduino + Node

Visualization of Signals using Arduino, Node.js & storing signals in MongoDB & mining data using Python

Drone-IoT-Comsi, INJE University

2nd semester, 2023

Email : chaos21c@gmail.com

My ID

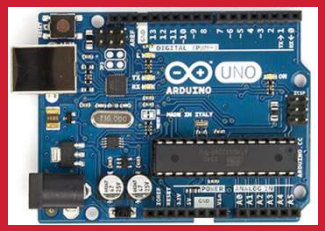
ID를 확인하고 github에 repo 만들기

ID	성명
AA01	강동하
AA02	고서진
AA03	김민재
AA04	김예원
AA05	김주호
AA06	김창욱
AA07	김현서
AA08	박종혁
AA09	서명진
AA10	유동기
AA11	
AA12	이근보
AA13	정호기

위의 id를 이용해서 github에 repo를 만드시오.

Option: 아두이노응용 실습 과제 – AAnn

Public, README.md check



[Review]

- [wk04]
  - Arduino sensors
  - Complete your project
  - Upload folder: aann-rpt05
  - Use repo “aann” in github

wk04 : Practice-04 : aann-rpt04

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

Upload folder : aann-rpt04

- 제출할 파일들

  ① AAnn_multi_Signals.png

  ② All *.ino

Email : chaos21c@gmail.com

wk04 : Practice-05 : aann-rpt05

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

Upload folder : aann-rpt05

- 제출할 파일들
  - ① AAnn_AnalogVoltage.png
  - ② AAnn_TMP36.png
  - ③ AAnn_LCD_lux.png
  - ④ All *.ino

Email : chaos21c@gmail.com

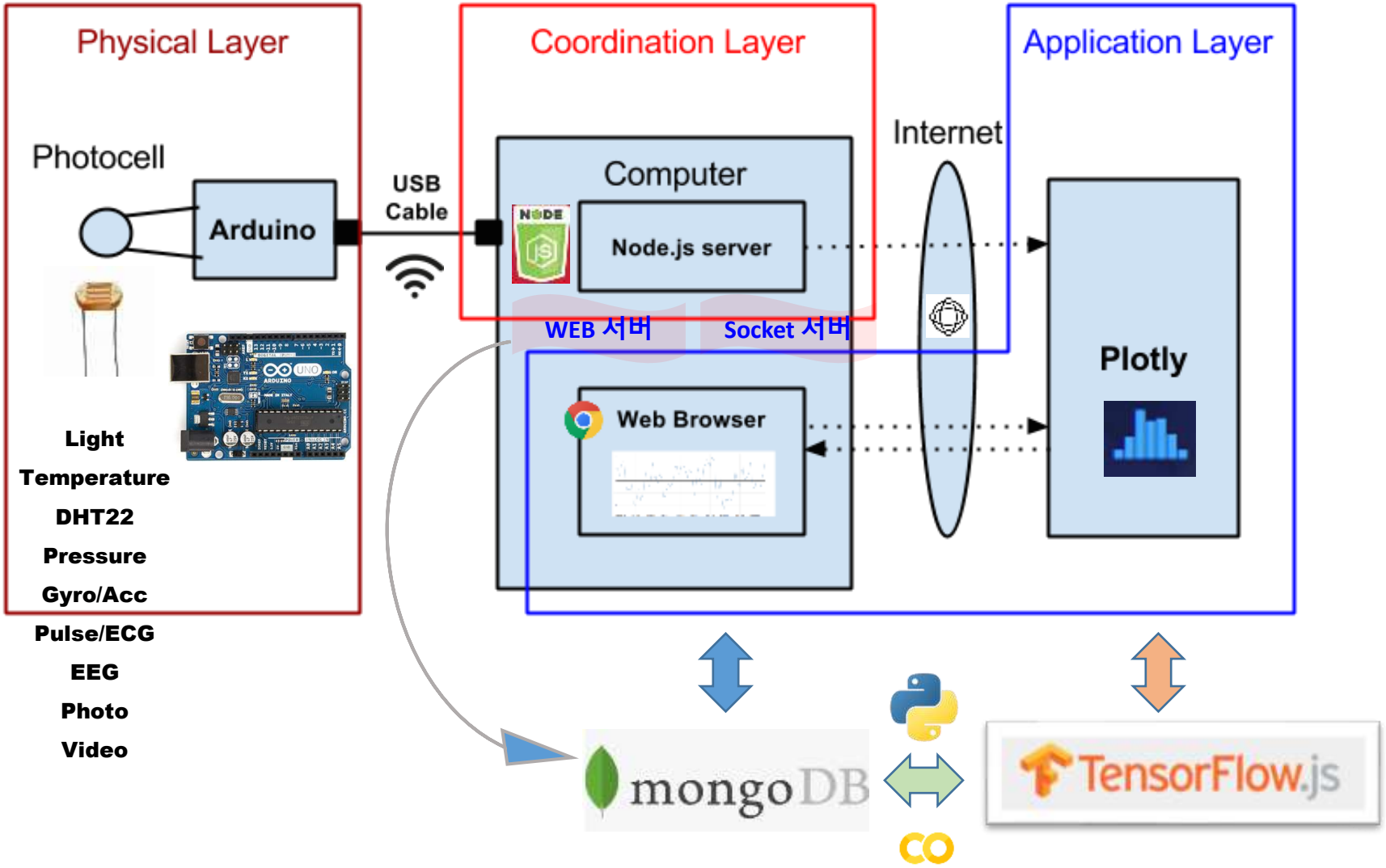
Purpose of AA

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. 저장된 IoT 데이터의 마이닝 (파이썬 코딩)

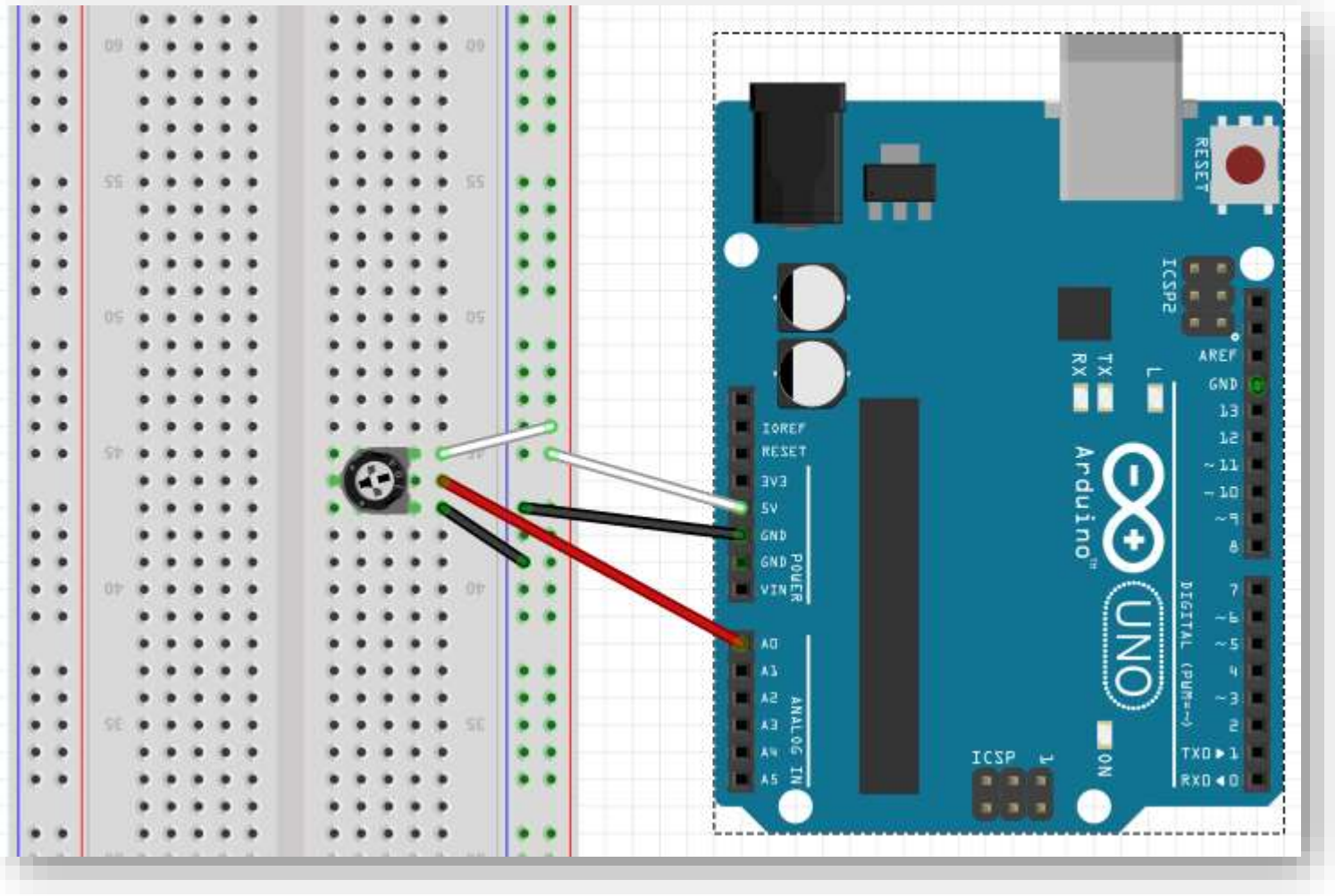
Layout [H S C]

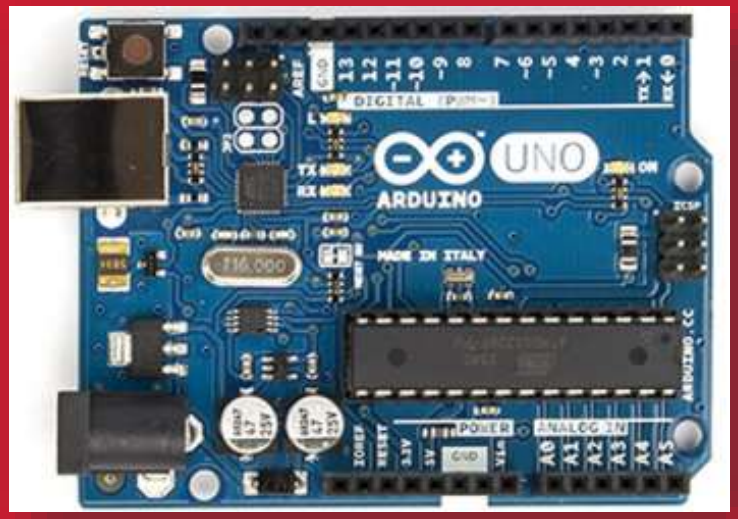

Physical Layer
Photocell
Arduino
USB Cable
Light
Temperature
DHT22
Pressure
Gyro/Acc
Pulse/ECG
EEG
Photo
Video
Coordination Layer
Computer
Node.js server
WEB 서버
Socket 서버
Web Browser
Internet
Application Layer
Plotly
mongoDB
TensorFlow.js

Analog Signal

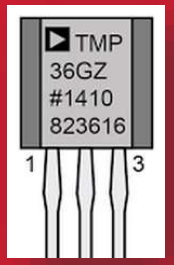
A2.5.1 AnalogReadSerial (circuit)

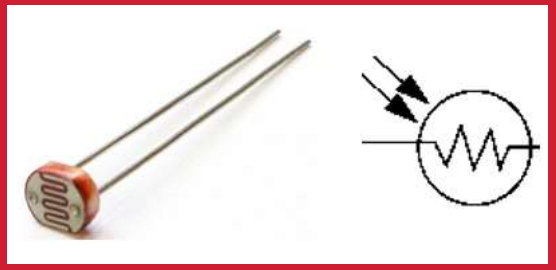
Standard potentiometer (가변 저항기)

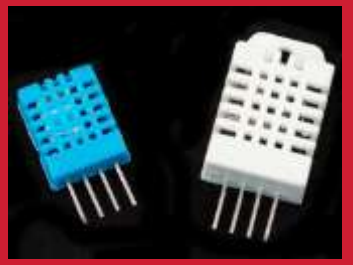




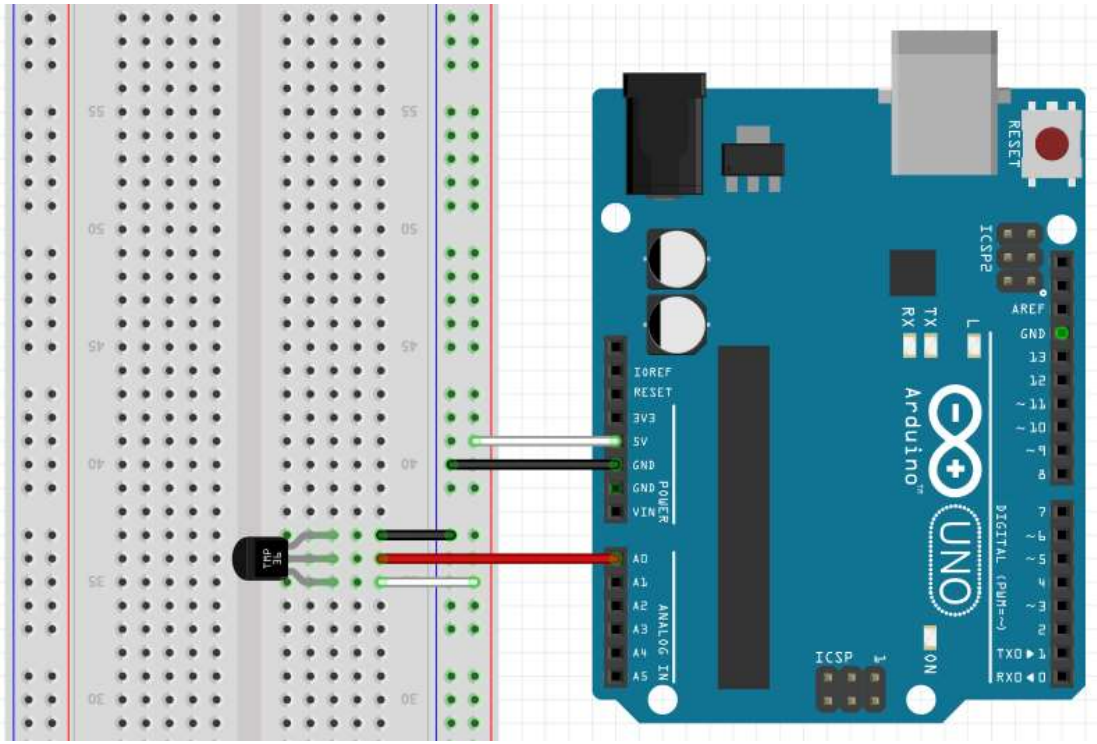
Arduino Sensors

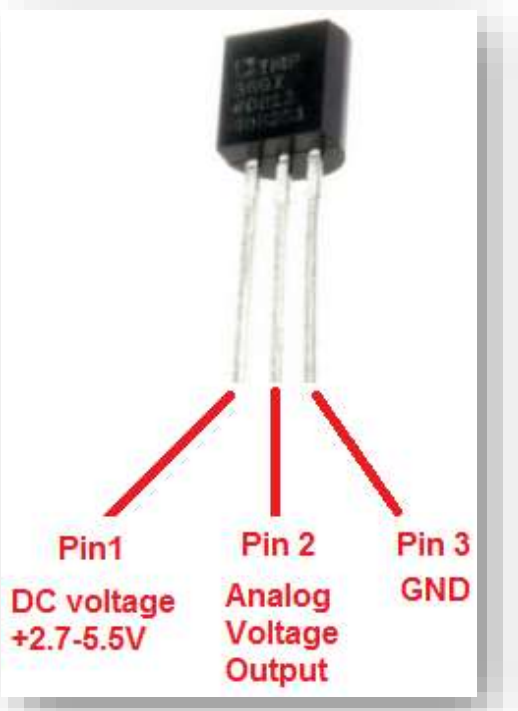




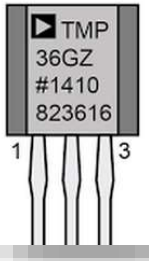


A3.1.1 Temperature sensor [TMP36]





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

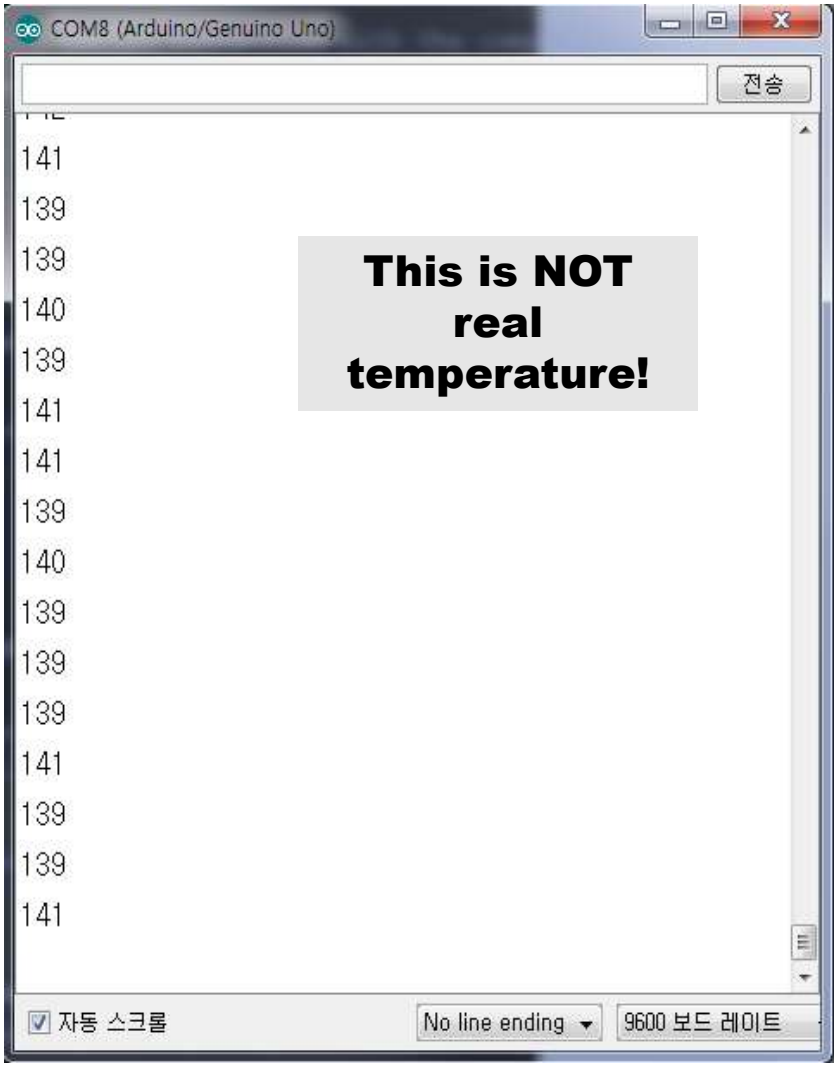
A3.1.2 Temperature sensor [TMP36]

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

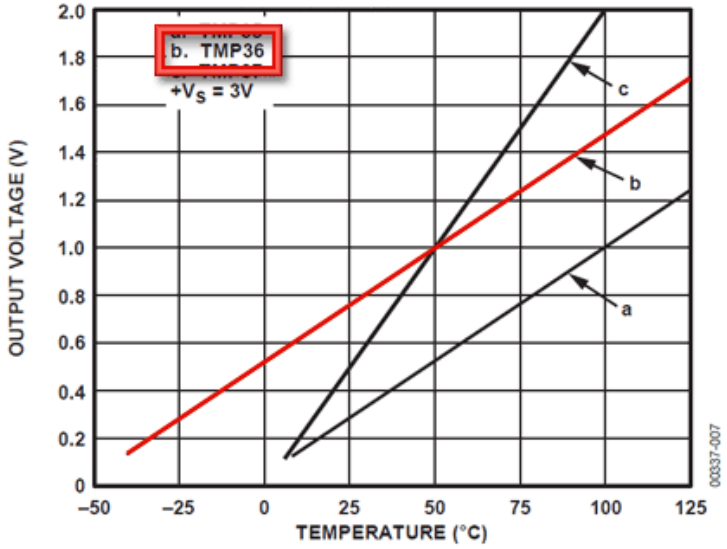


Figure 6. Output Voltage vs. Temperature

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

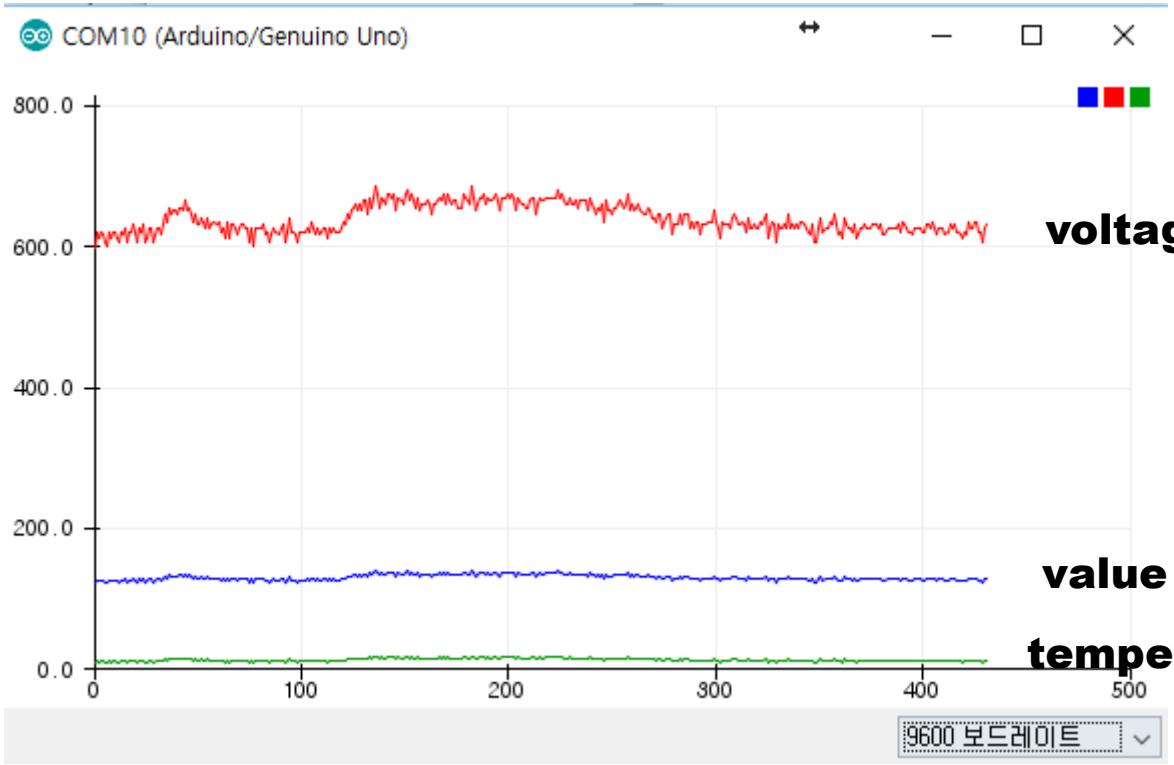
```
10 }
11
12 void loop() {
13    //getting the voltage reading from the temperature sensor
14   int value = analogRead(TEMP_INPUT);
15   Serial.print("AA00, value = ");
16   Serial.print(value);
17   Serial.print(" : ");
18
19    // converting that reading to voltage
20    float voltage = value * 5.0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

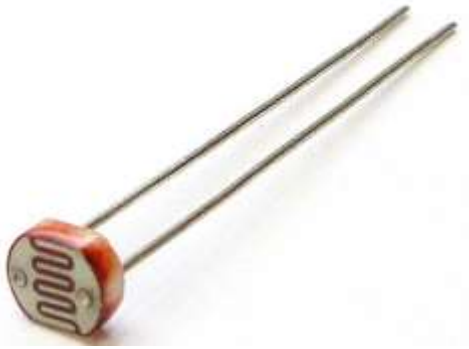
A3.1.5 Temperature sensor [TMP36]

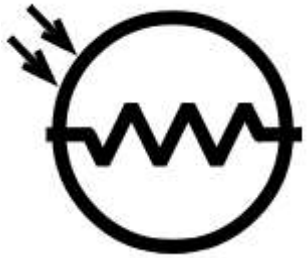


Save as
AAnn_TMP36.png

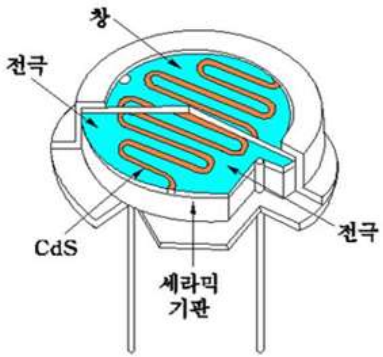
A3.2 Luminosity sensor [Photocell LDR]

CdS 센서– photoresistor





CDS특성

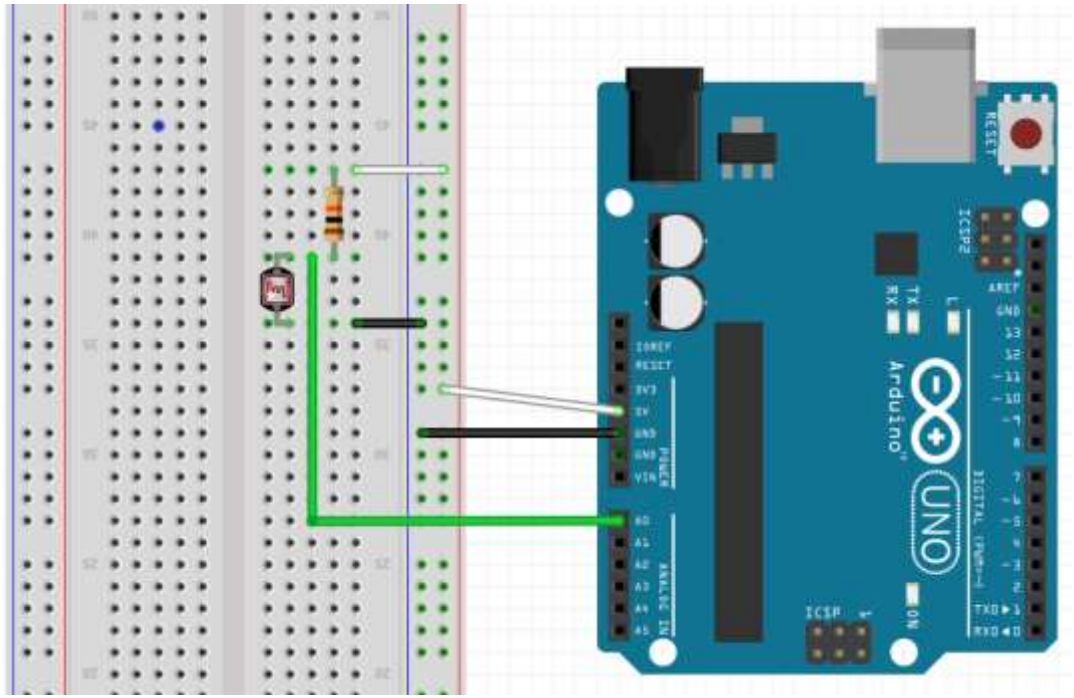


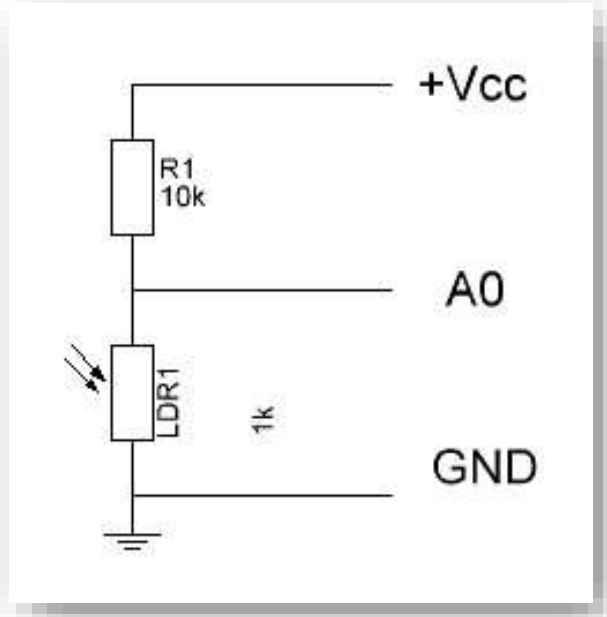
1. 감도
   - -빛의 파장에 따라 감도가 다름
2. 허용손실
   - -비교적 큰 전류를 흘릴 수 있음
3. 암 전류
   - -빛이 없어도 약간의 전류가 흐름
4. 명 전류
   - - 빛을 비추면 흐르는 전류
5. 응답특성
   - - 응답 시간 지연
   - - 빛의 세기에 따라 응답시간 다름
6. 가변저항
   - -빛에 따른 가변저항

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

A3.2.2 Luminosity sensor [Photocell LDR]

CdS 센서 회로





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

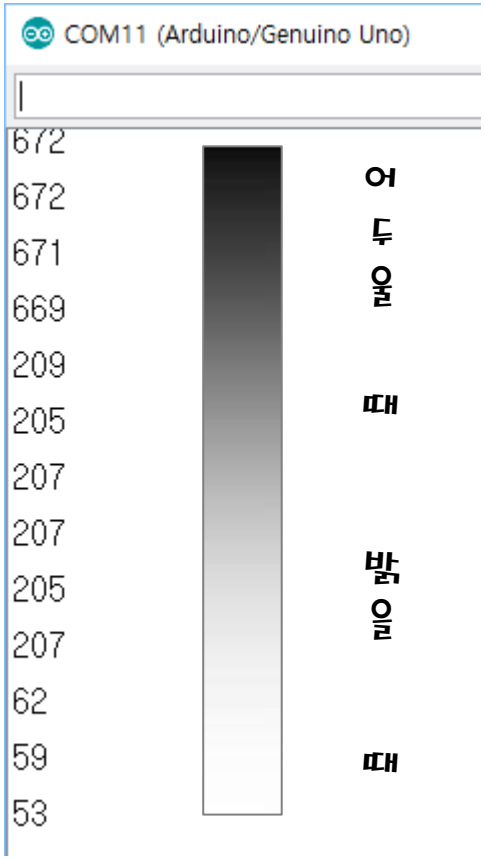
광센서에서의 전압 강하 값을 **A0**로 측정

A3.2.4 Luminosity sensor [Photocell LDR]

CdS 센서 회로 – 측정 1.

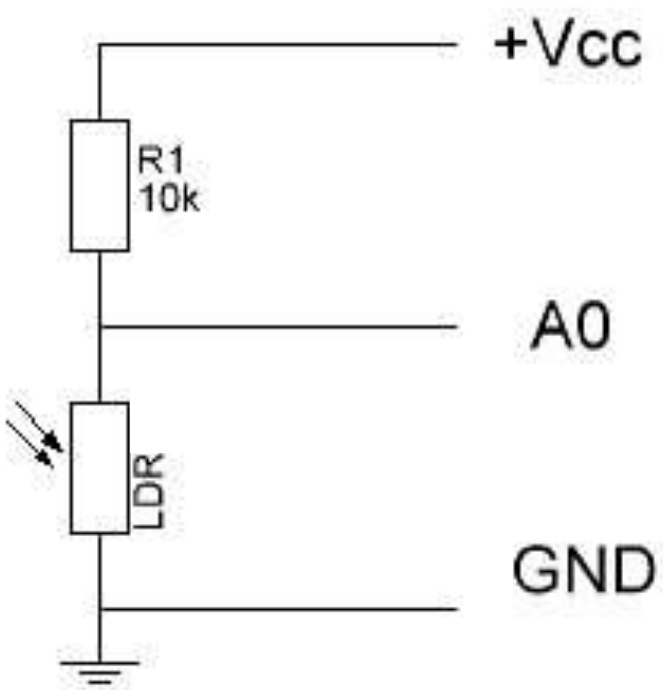
CdS_start

```
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 (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**

A3.2.6 Luminosity sensor [Photocell LDR]

CdS 센서 회로 – 측정 2.

sketch08_CdS2

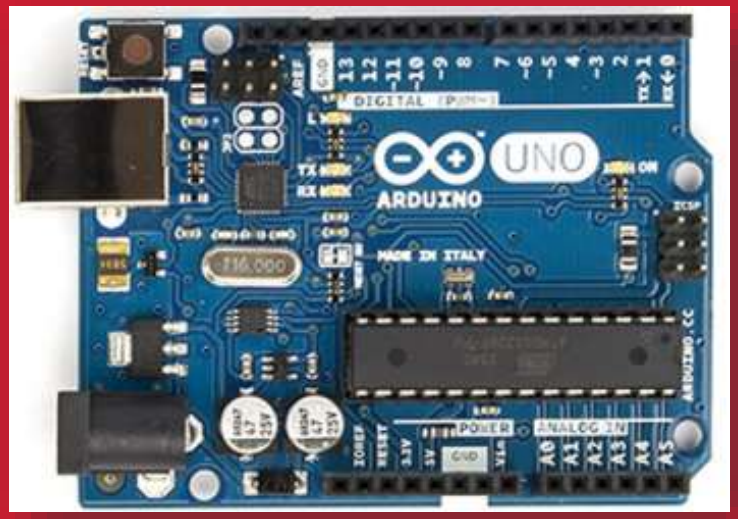
```
1  //  lux
2  #define CDS_INPUT 0
3
4  void setup() {
5  Serial.begin(9600);
6  }
7  void loop() {
8    int value = analogRead(CDS_INPUT);
9    Serial.println(int(luminosity(value)));
10   delay(1000);
11 }
12
13 //Voltage to Lux
14 double luminosity (int RawADC0){
15   double Vout=RawADC0*5.0/1023;  // 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 }
```

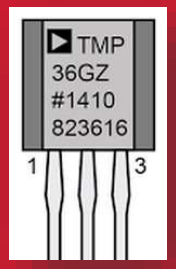
COM11 (Arduino/Genuino Uno)

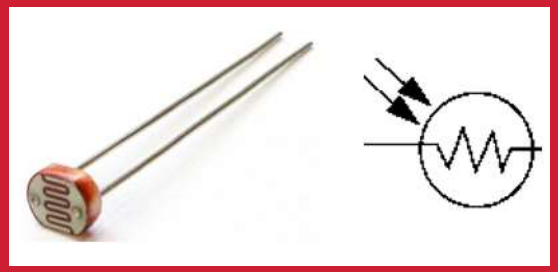
```
23
24
23
183
191
183
185
531
1169
1262
1198
1039
1063
```

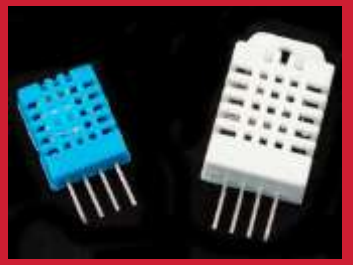
밝을수록 측정 값이 커지고
어두을수록 값이 작아진다 !!!

Arduino Sensors + Node.js









IOT: HSC

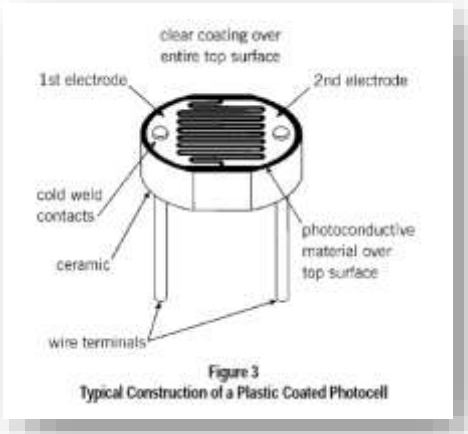
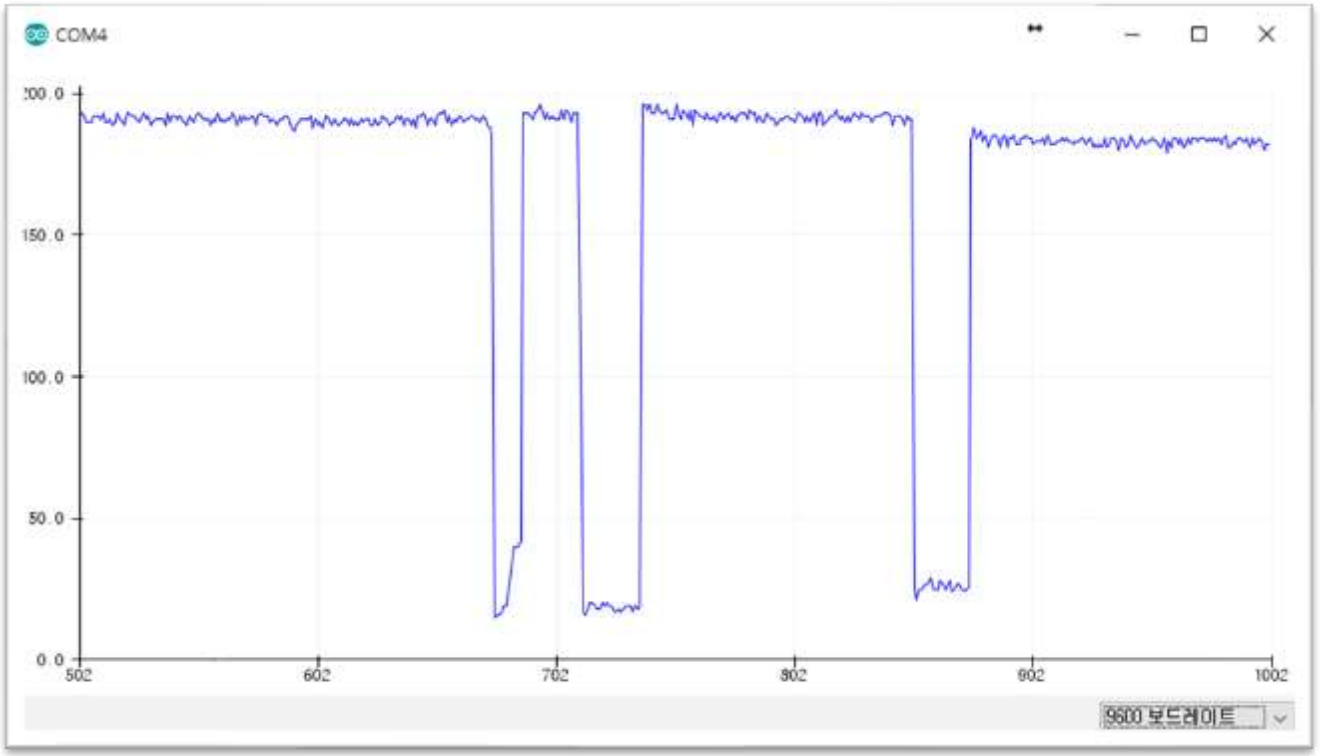
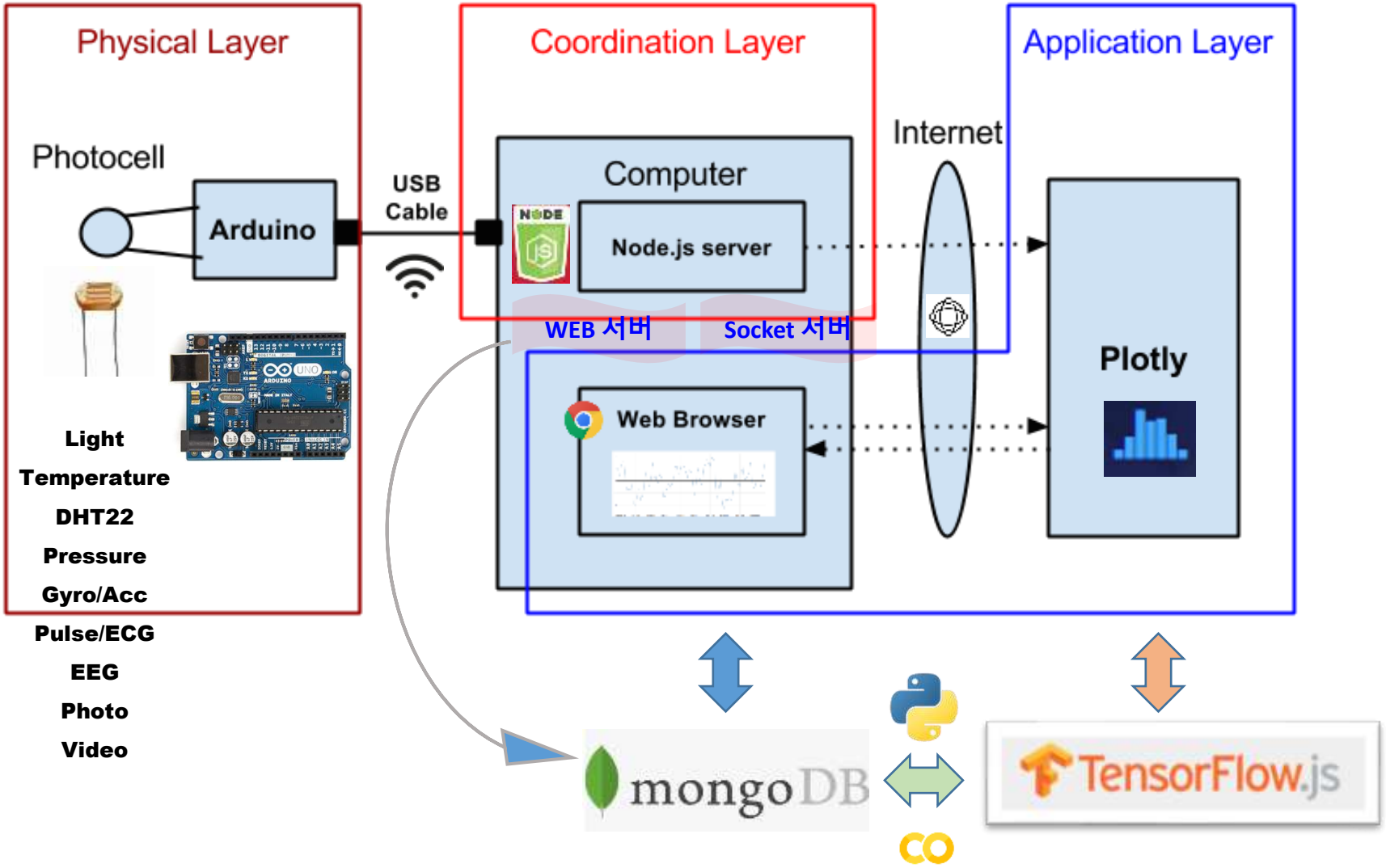


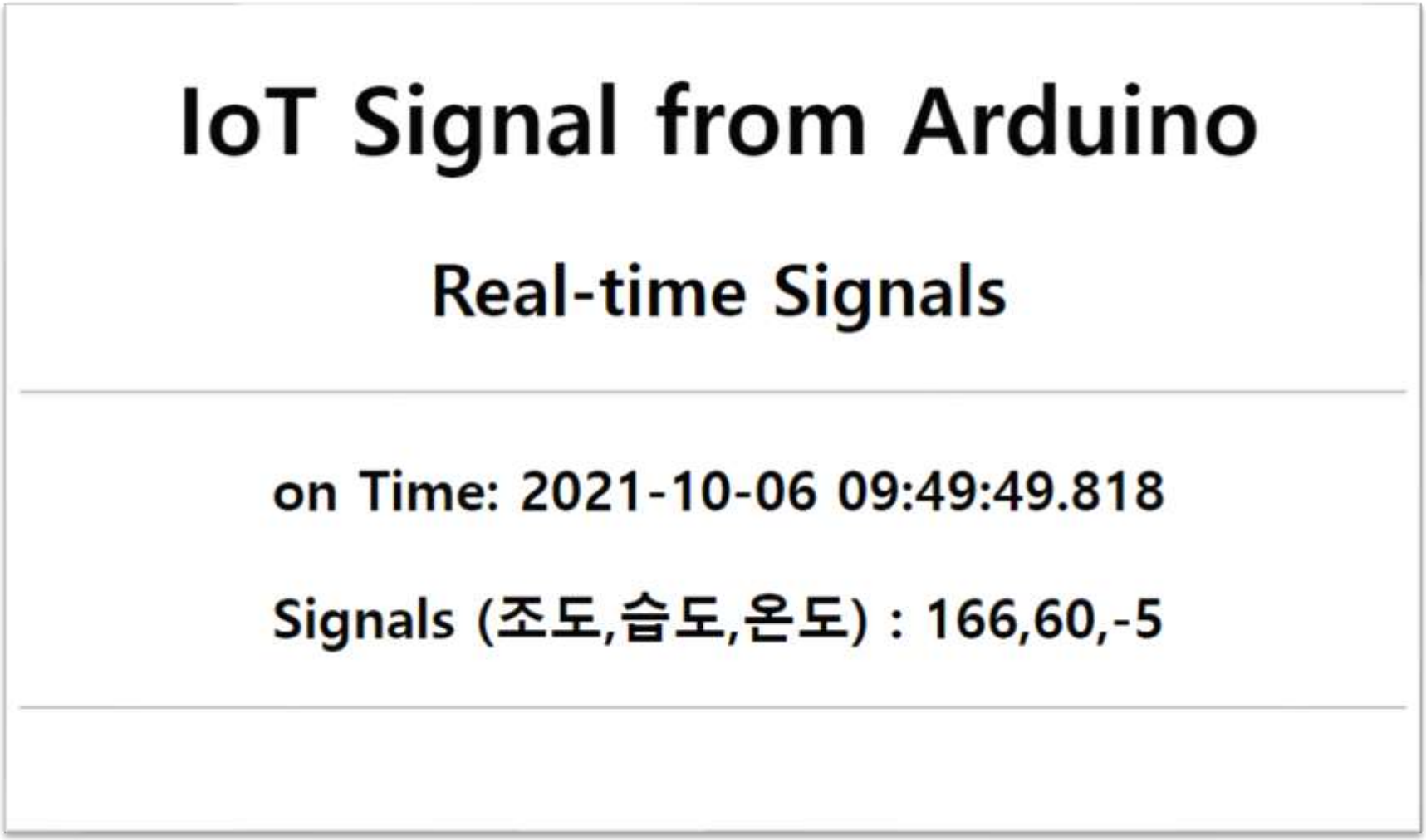
Figure 3
Typical Construction of a Plastic Coated Photocell



Layout [H S C]

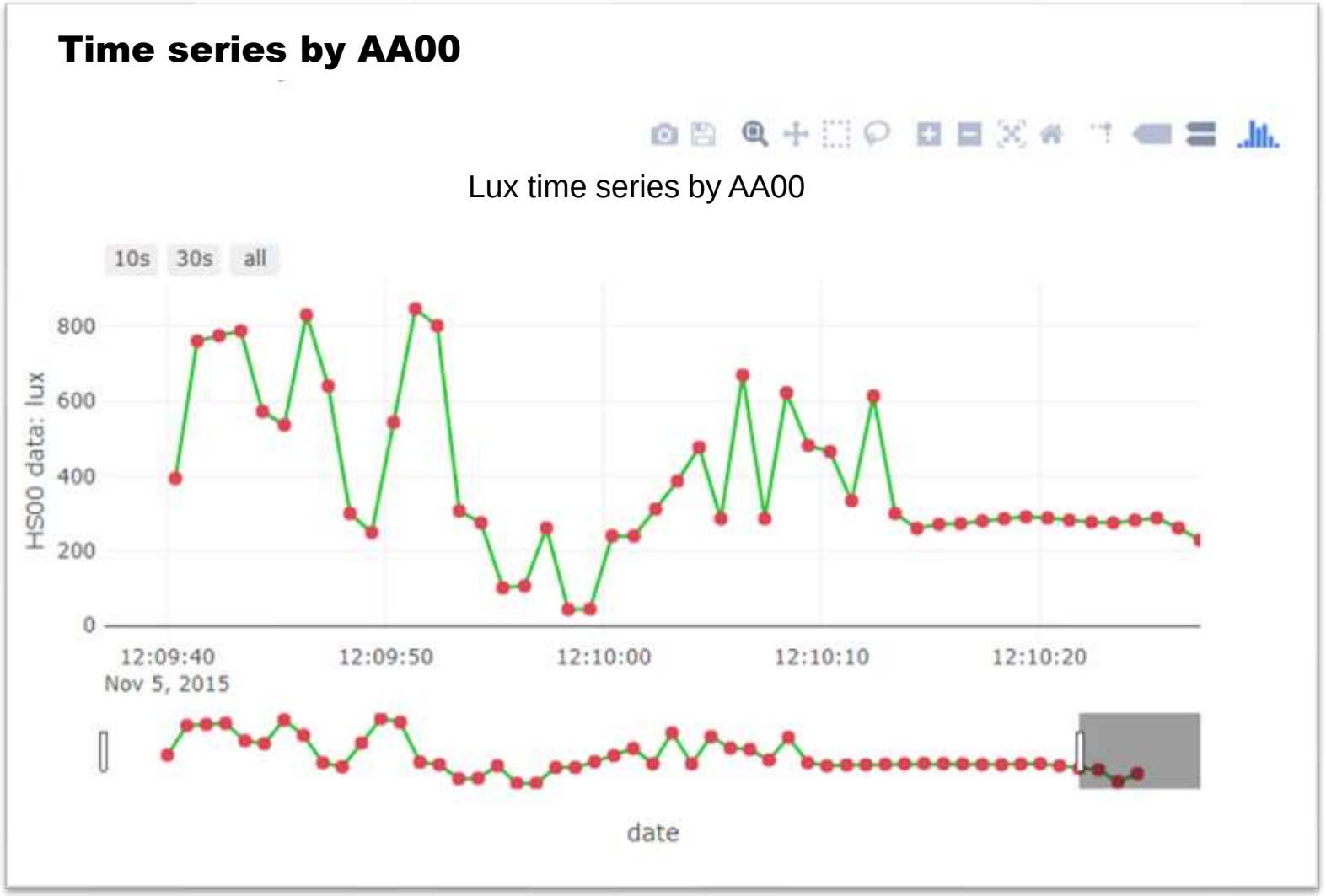

Physical Layer
Photocell
Arduino
USB Cable
Light
Temperature
DHT22
Pressure
Gyro/Acc
Pulse/ECG
EEG
Photo
Video
Coordination Layer
Computer
Node.js server
WEB 서버
Socket 서버
Web Browser
Internet
Application Layer
Plotly
mongoDB
TensorFlow.js

on WEB monitoring Arduino data

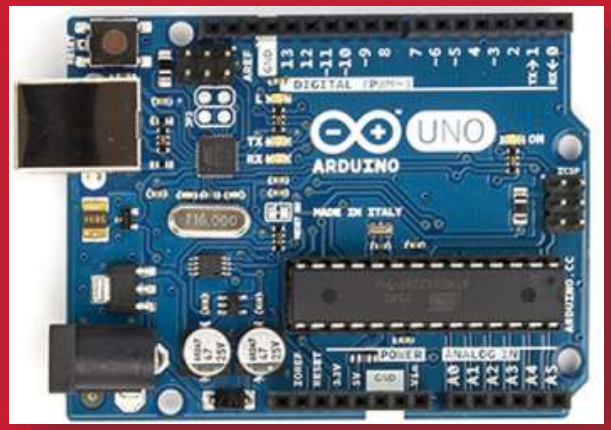


http://chaos.inje.ac.kr:3030/iot_multi.html

Arduino data + plotly

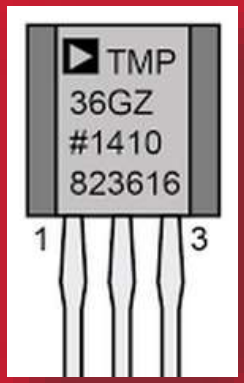


Single sensor: tmp36

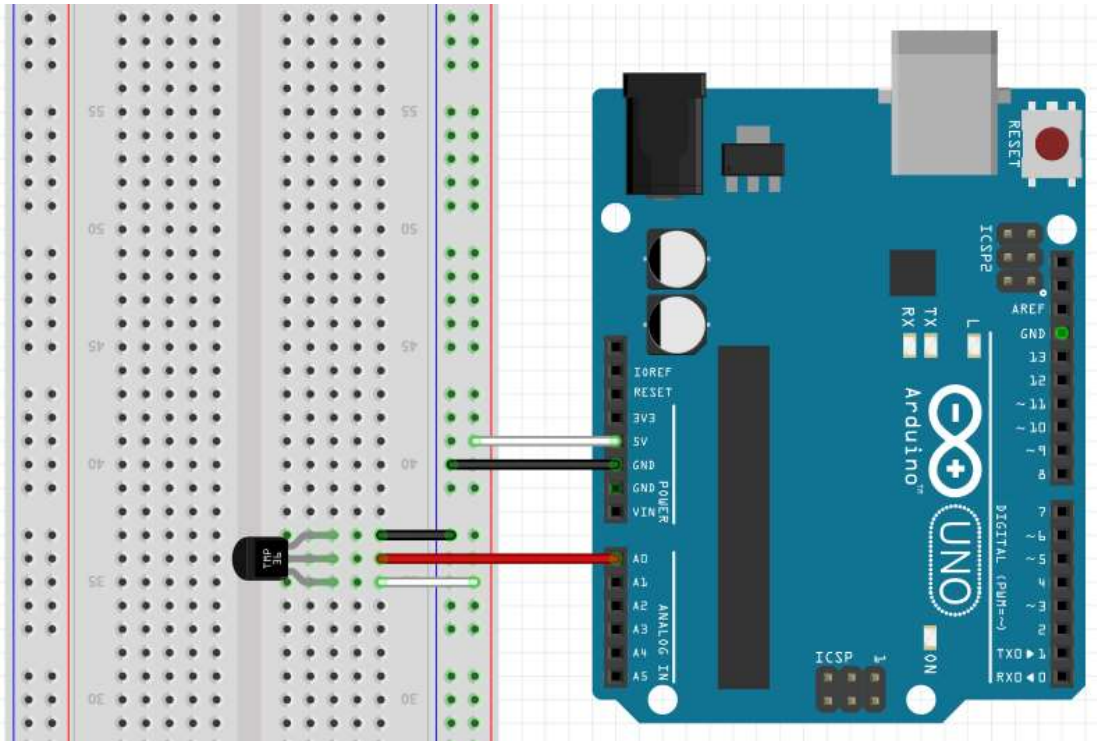


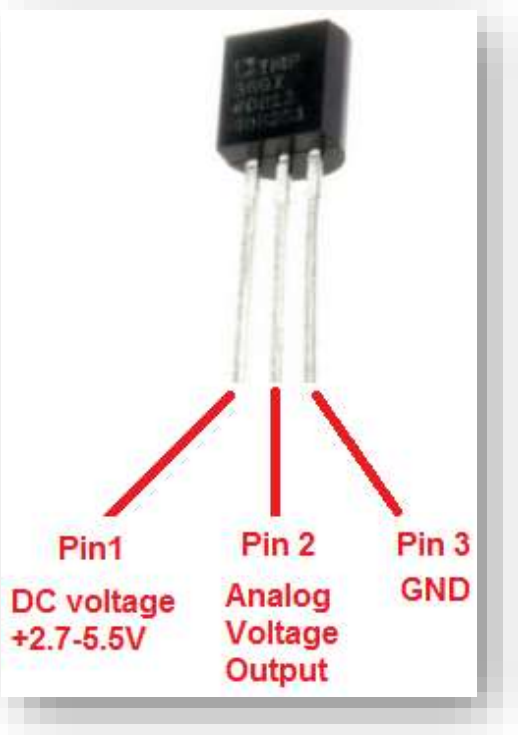
TMP36

Node project

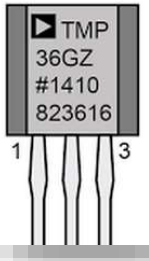


A3.1.1 Temperature sensor [TMP36]





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

A4.1.1 tmp36 node project

Start tmp36-node project

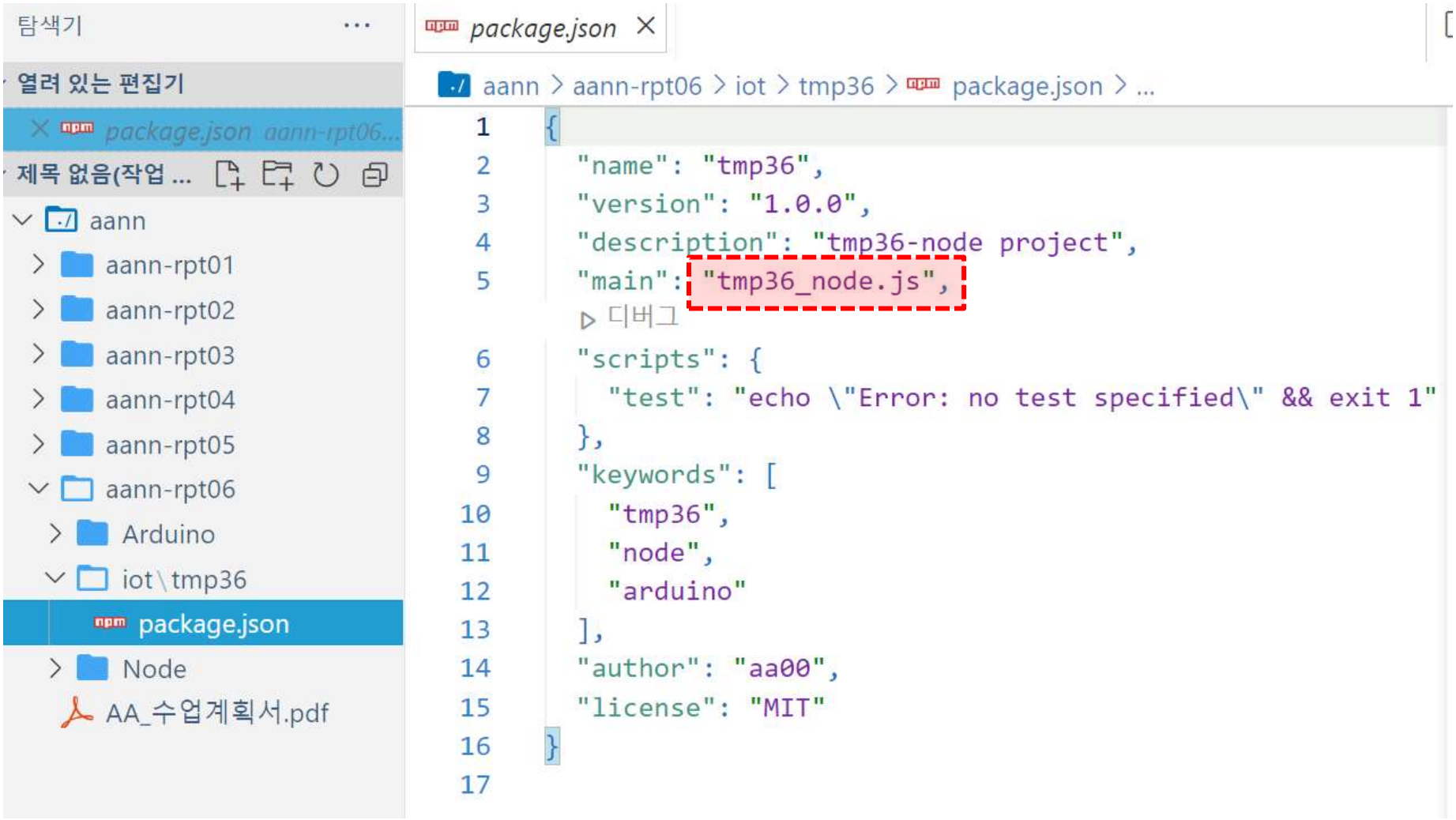
1. **Go to my working folder: aann-rpt06**
2. **md iot & cd iot**
3. **md tmp36**
4. **cd tmp36**
5. **Open terminal**
6. **npm init**

A4.1.2 tmp36 node project: npm init

문제 출력 디버그 콘솔 터미널 node

```
D:\aann\aann-rpt06\iot\tmp36>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaul
ts.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
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 arduino
author: aa00
license: (ISC) MIT
```

A4.1.3 tmp36 node project: package.json



A4.1.4 tmp36 node project: install modules

npm install --save serialport

```
D:\aann\aann-rpt06\iot\tmp36>npm install --save serialport

> @serialport/bindings@9.2.4 install D:\aann\aann-rpt06\iot\tmp36\node_m
odules\@serialport\bindings
> prebuild-install --tag-prefix @serialport/bindings@ || node-gyp rebuil
d

npm notice created a lockfile as package-lock.json. You should commit th
is file.
npm WARN tmp36@1.0.0 No repository field.

+ serialport@9.2.4
added 74 packages from 45 contributors and audited 74 packages in 11.774
s

17 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

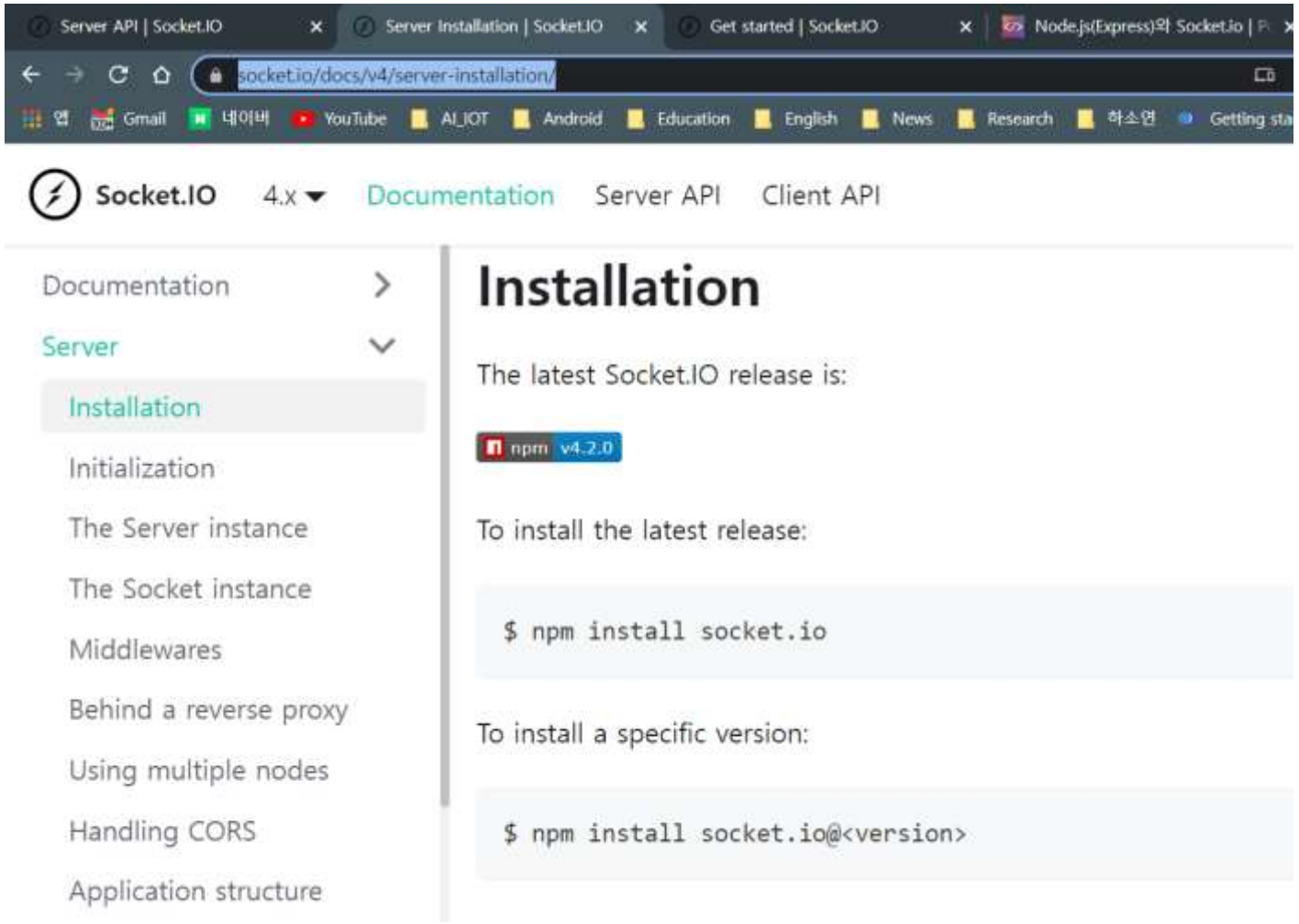
A4.1.4 tmp36 node project: install modules

npm install --save serialport

```
D:\aann\aann-rpt06\Node>npm install serialport

added 21 packages, and audited 22 packages in 5s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

A4.1.4 tmp36 node project: install modules

socket.io



https://socket.io/docs/v4/server-installation/

A4.1.4 tmp36 node project: install modules

npm install --save socket.io@2.3.0

```
D:\aann\aann-rpt06\iot\tmp36>npm install --save socket.io@2.3.0
npm WARN tmp36@1.0.0 No repository field.

+ socket.io@2.3.0
added 52 packages from 33 contributors and audited 126 packages in 3.878
s

17 packages are looking for funding
  run `npm fund` for details

found 4 vulnerabilities (2 moderate, 1 high, 1 critical)
  run `npm audit fix` to fix them, or `npm audit` for details

  D:\aann\aann-rpt06\iot\tmp36 디렉터리

 2021-10-05  오전 10:23    <DIR>          .
 2021-10-05  오전 10:23    <DIR>          ..
 2021-10-05  오전 10:23    <DIR>          node_modules
 2021-10-05  오전 10:23            28,477 package-lock.json
 2021-10-05  오전 10:23               367 package.json
               2개 파일              28,844 바이트
               3개 디렉터리  2,424,474,251,264 바이트 남음
```

A4.1.4 tmp36 node project: install modules

npm install --save socket.io@2.3.0

```
found 4 vulnerabilities (2 moderate, 1 high, 1 critical)
  run `npm audit fix` to fix them, or `npm audit` for details

D:\aann\aann-rpt06\iot\tmp36>npm audit fix
npm WARN tmp36@1.0.0 No repository field.

+ socket.io@2.4.1
added 1 package, removed 11 packages, updated 11 packages and moved 1 pa
ckage in 1.33s

17 packages are looking for funding
  run `npm fund` for details

fixed 1 of 4 vulnerabilities in 126 scanned packages
  1 package update for 3 vulnerabilities involved breaking changes
  (use `npm audit fix --force` to install breaking changes; or refer to
`npm audit` for steps to fix these manually)
```

A4.1.4 tmp36 node project: install modules

npm install --save socket.io [N.A.]

4.x 버전 설치는 좀 더 검토가 필요.

```
D:\aann\aann-rpt06\iot\tmp36>npm install --save socket.io
npm WARN tmp36@1.0.0 No repository field.

+ socket.io@4.2.0
added 20 packages from 66 contributors and audited 94 packages in 2.046s

17 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities


  D:\aann\aann-rpt06\iot\tmp36 디렉터리

 2021-10-05  오전 10:23    <DIR>          .
 2021-10-05  오전 10:23    <DIR>          ..
 2021-10-05  오전 10:23    <DIR>          node_modules
 2021-10-05  오전 10:23            28,477 package-lock.json
 2021-10-05  오전 10:23               367 package.json
               2개 파일              28,844 바이트
               3개 디렉터리  2,424,474,251,264 바이트 남음
```

A4.1.4 tmp36 node project: install modules

정상 동작 버전을 설치!

npm install --save serialport@9.2.4

npm install --save socket.io@2.4.1

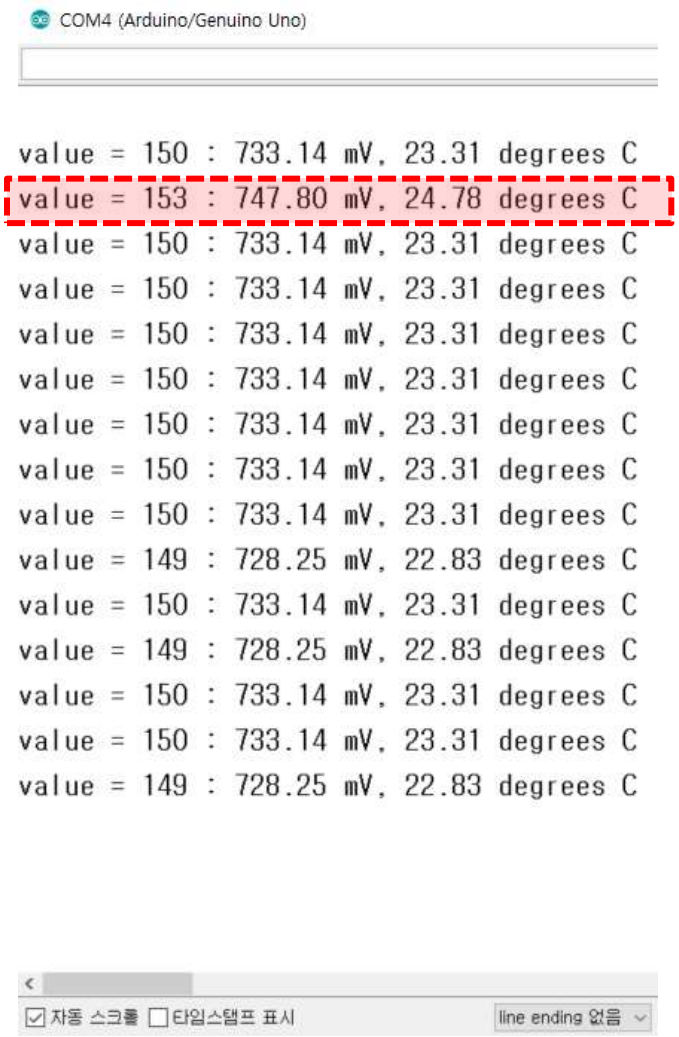
```
"author": "aa00",
"license": "MIT",
"dependencies": {
  "serialport": "^9.2.4",
  "socket.io": "^2.4.1"
}
```

A4.1.5 tmp36 node project (Arduino code)

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



A4.1.6 tmp36 node project (node code)

tmp36_node_start.js

```
3   var serialport = require("serialport");
4   var portName = "COM3"; // check your COM port!!
5   var port = process.env.PORT || 3000;
6
7   var io = require("socket.io").listen(port);
8
9   const Readline = require("@serialport/parser-readline");
10  // serial port object
11  var sp = new serialport(portName, {
12    baudRate: 9600, // 9600  38400
13    dataBits: 8,
14    parity: "none",
15    stopBits: 1,
16    flowControl: false,
17    parser: new Readline("\r\n"),
18  });
```

```
20  const parser = sp.pipe(new Readline({ delimiter: "\r\n" }));
21
22  // Read the port data
23  sp.on("open", () => {
24    console.log("serial port open");
25  });
26
27  var tdata = []; // Array
28
29  parser.on("data", (data) => {
30    // call back when data is received
31    // raw data only
32    //console.log(data);
33
34    tdata = data; // data
35    console.log("AA00," + tdata);
36    io.sockets.emit("message", tdata); // send data to all clie
37  });
```

A4.1.7 tmp36 node project (node cmd message)

[Terminal] node tmp36_node.js

```
D:\aann\aann-rpt06\iot\tmp36>node tmp36_node
serial port open
67.35 mV, 26.74 degrees C
67.35 mV, 26.74 degrees C
7 : 767.35 mV, 26.74 degrees C
7 : 767.35 mV, 26.74 degrees C
AA00, value = 157 : 767.35 mV, 26.74 degrees C
AA00, value = 157 : 767.35 mV, 26.74 degrees C
AA00, value = 157 : 767.35 mV, 26.74 degrees C
AA00, value = 157 : 767.35 mV, 26.74 degrees C
AA00, value = 155 : 757.58 mV, 25.76 degrees C
AA00, value = 155 : 757.58 mV, 25.76 degrees C
AA00, value = 156 : 762.46 mV, 26.25 degrees C
AA00, value = 156 : 762.46 mV, 26.25 degrees C
AA00, value = 156 : 762.46 mV, 26.25 degrees C
AA00, value = 156 : 762.46 mV, 26.25 degrees C
AA00, value = 155 : 757.58 mV, 25.76 degrees C
AA00, value = 155 : 757.58 mV, 25.76 degrees C
AA00, value = 156 : 762.46 mV, 26.25 degrees C
AA00, value = 156 : 762.46 mV, 26.25 degrees C
```

A4.1.8 tmp36 node project (all messages)

tmp36_node.js

```
var dStr = "";
var tdata = []; // Array

parser.on("data", (data) => {
  // call back when data is received
  // raw data only
//   console.log(data);

  dStr = getDateString();
  tdata[0] = dStr;
  tdata[1] = data; // data
  console.log("AA00," + tdata.toString());
  io.sockets.emit("message", tdata); // se
});

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;
}
```

[Terminal] node tmp36_node

```
D:\aann\aann-rpt06\iot\tmp36>node tmp36_node
serial port open
AA00,2021-10-05 11:21:24.062,lue = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:24.062,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:24.062,value = 157 : 767.35 mV, 26.74 degrees C
AA00,2021-10-05 11:21:24.062,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:24.063,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:24.063,value = 157 : 767.35 mV, 26.74 degrees C
AA00,2021-10-05 11:21:25.644,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:26.648,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:27.651,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:28.651,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:29.655,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:30.658,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:31.662,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:32.661,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:33.665,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:34.669,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:35.672,value = 156 : 762.46 mV, 26.25 degrees C
AA00,2021-10-05 11:21:36.676,value = 155 : 757.58 mV, 25.76 degrees C
AA00,2021-10-05 11:21:37.675,value = 156 : 762.46 mV, 26.25 degrees C
```

AAnn_tmp36_message.png
로 저장

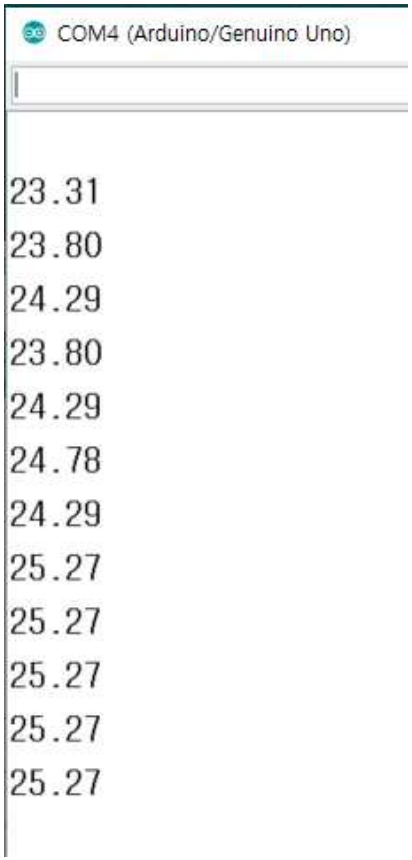
A4.1.9 tmp36 node project (only data)

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



A4.1.10 tmp36 node project (date & data → IOT)

[Terminal] node tmp36_node

```
D:\aann\aann-rpt06\iot\tmp36>node tmp36_node
serial port open
AA00,2021-10-05 11:31:03.941,26.25
AA00,2021-10-05 11:31:04.944,26.25
AA00,2021-10-05 11:31:05.945,26.25
AA00,2021-10-05 11:31:06.948,26.25
AA00,2021-10-05 11:31:07.951,26.25
AA00,2021-10-05 11:31:08.951,26.25
AA00,2021-10-05 11:31:09.954,25.76
AA00,2021-10-05 11:31:10.954,26.25
AA00,2021-10-05 11:31:11.958,26.25
AA00,2021-10-05 11:31:12.957,26.25
AA00,2021-10-05 11:31:13.961,26.25
AA00,2021-10-05 11:31:14.964,26.25
AA00,2021-10-05 11:31:15.964,26.25
```

시간 , 온도

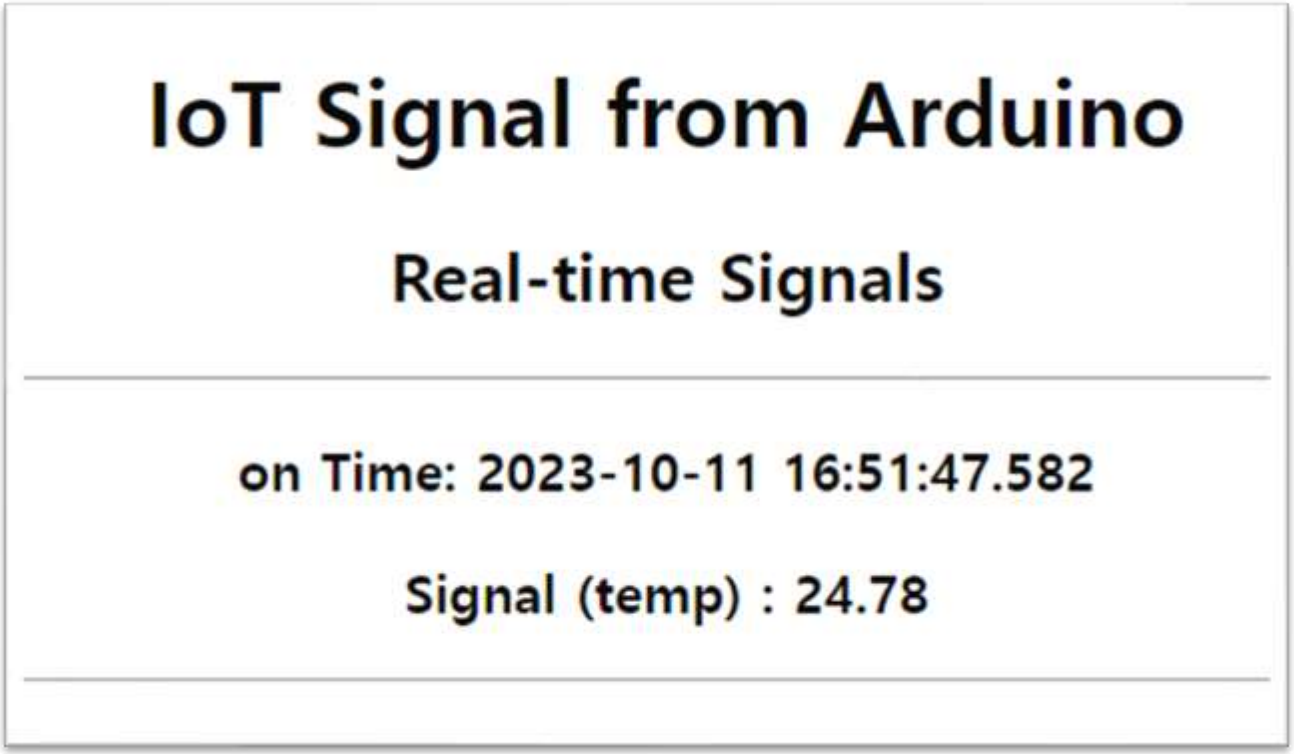
IOT data format
시간, **data**
시간, 온도

AAnn_tmp36_IOT_data.png
로 저장

공백없이 “,”로
시간과 온도 구분

A4.1.11 tmp36 node project (web monitoring)

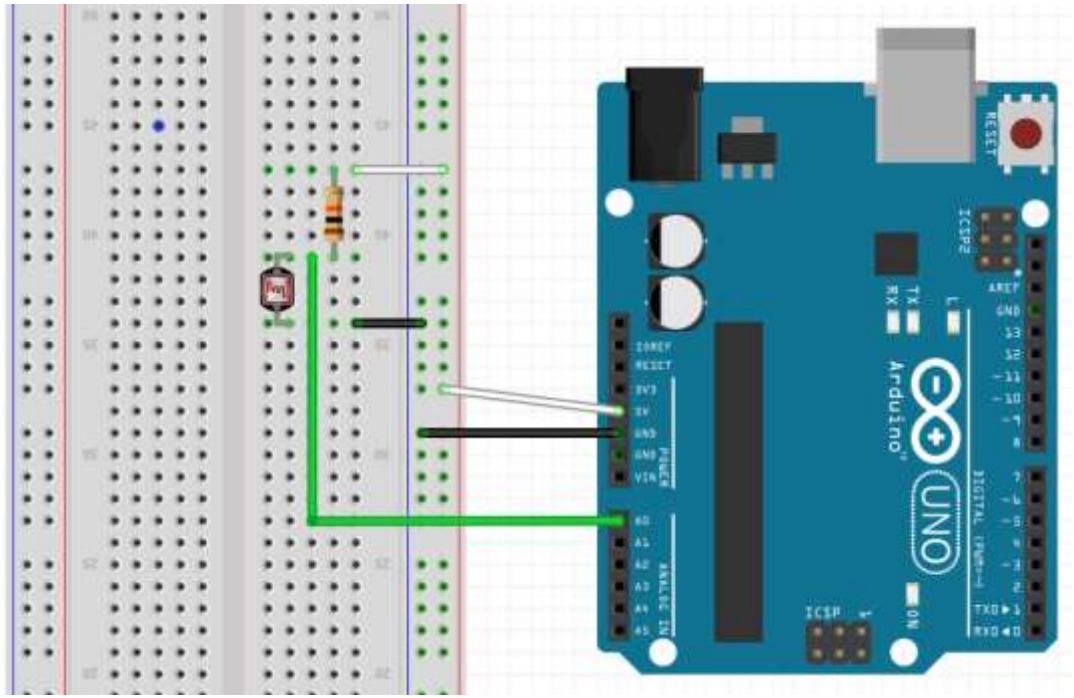
[Web monitoring] client_signal_tmp36.html

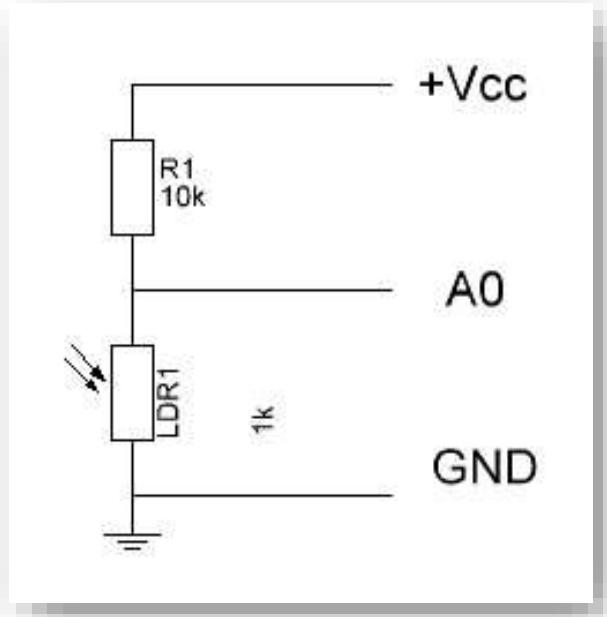


AAnn_tmp36_IOT_WEB.png
로 저장

A3.2.2 Luminosity sensor [Photocell LDR]

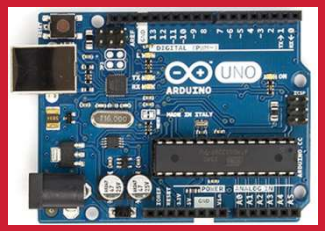
CdS 센서 회로





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

광센서에서의 전압 강하 값을 **A0**로 측정



[Practice]

- [wk05]
  - Arduino sensors + Node.js
  - Complete your project
  - Upload folder: aann-rpt06
  - Use repo “aann” in github

wk05 : Practice : aann-rpt06

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

제출폴더명 : **aann-rpt06**

- 압축할 파일들

① AAnn_tmp36_message.png

② AAnn_tmp36_IOT_data.png

③ AAnn_tmp36_IOT_WEB.png

④ AAnn_cds_IOT_data.png

⑤ AAnn_cds_IOT_WEB.png

⑥ All *.ino

⑦ All *.js

⑧ NO node_modules folder

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

Target of this class