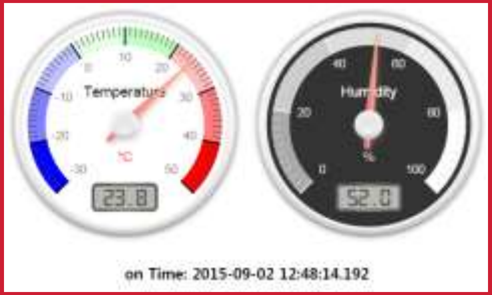
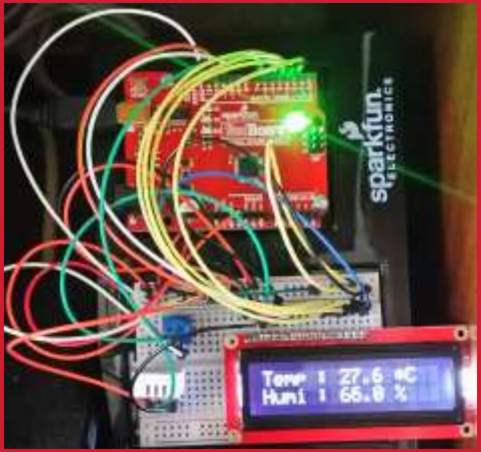
Arduino-IOT

[wk09]

Arduino + Node

I. Multi signals





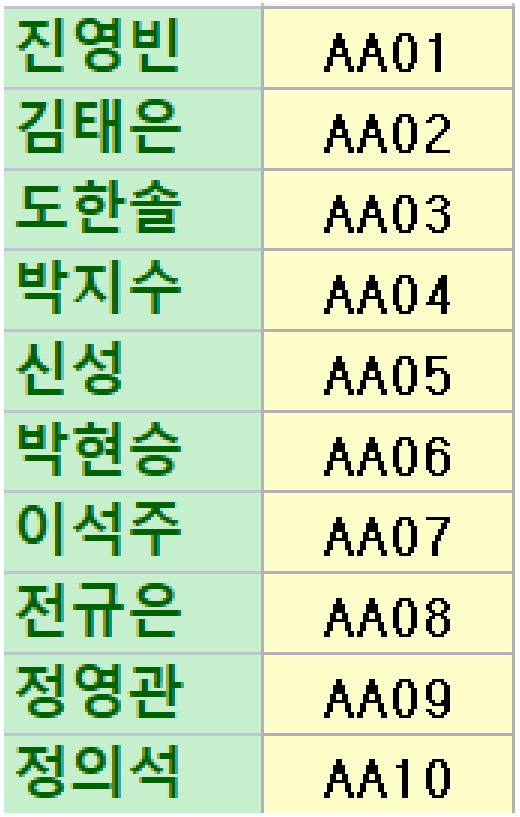
Visualization of Signals using Arduino, Node.js & storing signals in MongoDB

Comsi, INJE University

2nd semester, 2018

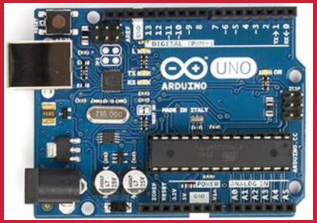
Email : chaos21c@gmail.com

My ID



진영빈	AA01
김태은	AA02
도한솔	AA03
박지수	AA04
신성	AA05
박현승	AA06
이석주	AA07
전규은	AA08
정영관	AA09
정의석	AA10
이근재	**AA11**

[Review]



- [wk05]
  - Arduino sensors
  - Complete your project
  - Submit file : AAnn_Rpt04.zip

wk05 : Practice-04 : AAnn_Rpt04.zip

◆ **[Target of this week]**

- Complete your works
- **Save your outcomes and compress 6 outputs**

제출파일명 : **AAnn_Rpt04.zip**

- 압축할 파일들

① **AAnn_AnalogVoltage.png**

② **AAnn_TMP36.png**

③ **AAnn_CdS_LED.ino**

④ **AAnn_Hello_LCD.png**

⑤ **AAnn_LCD_lux.ino**

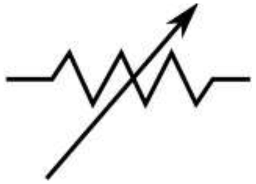
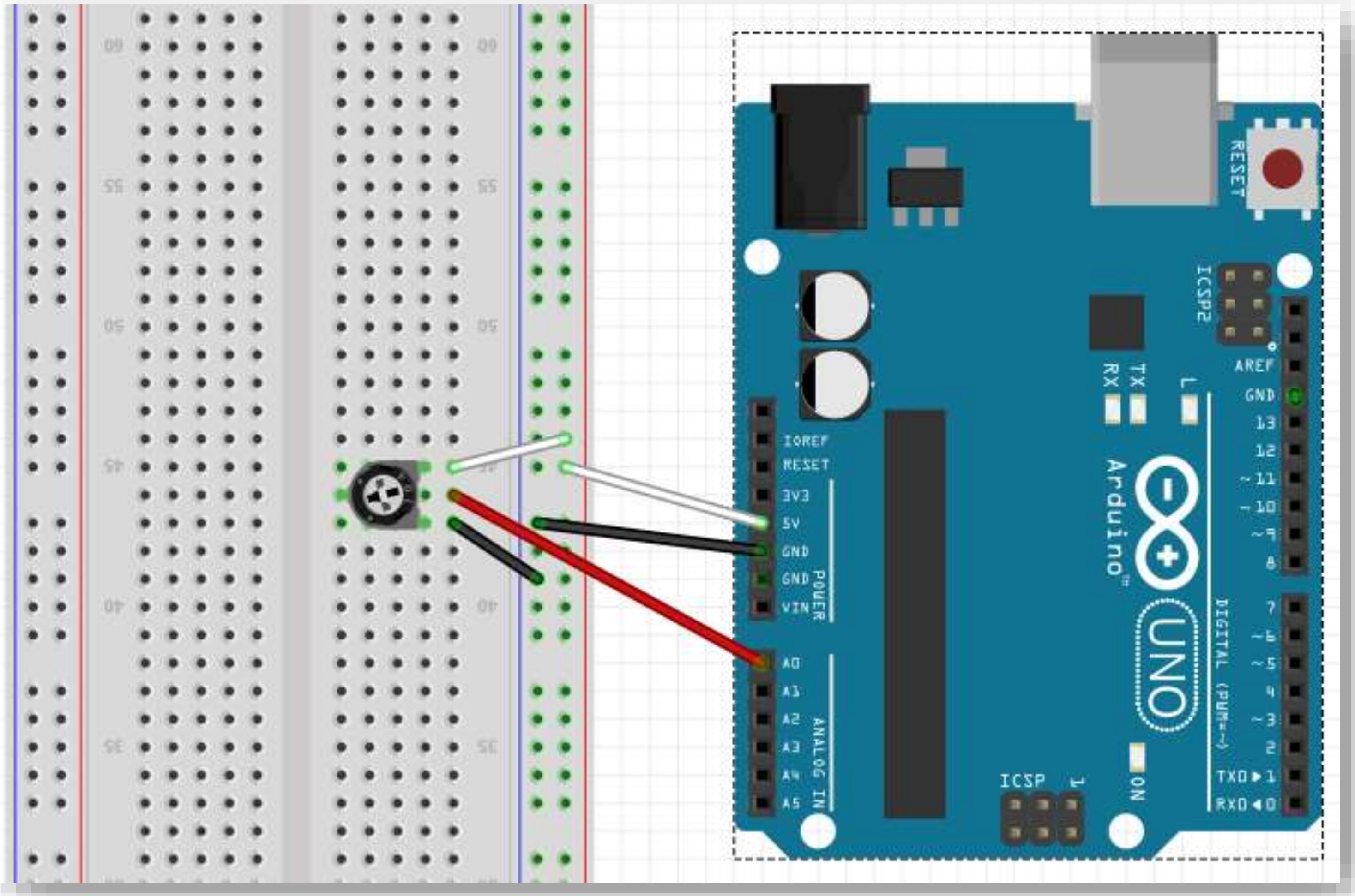
⑥ **AAnn_LCD_lux.png**

[제목 **: id,** 이름 **(**수정**)]**

Analog Signal

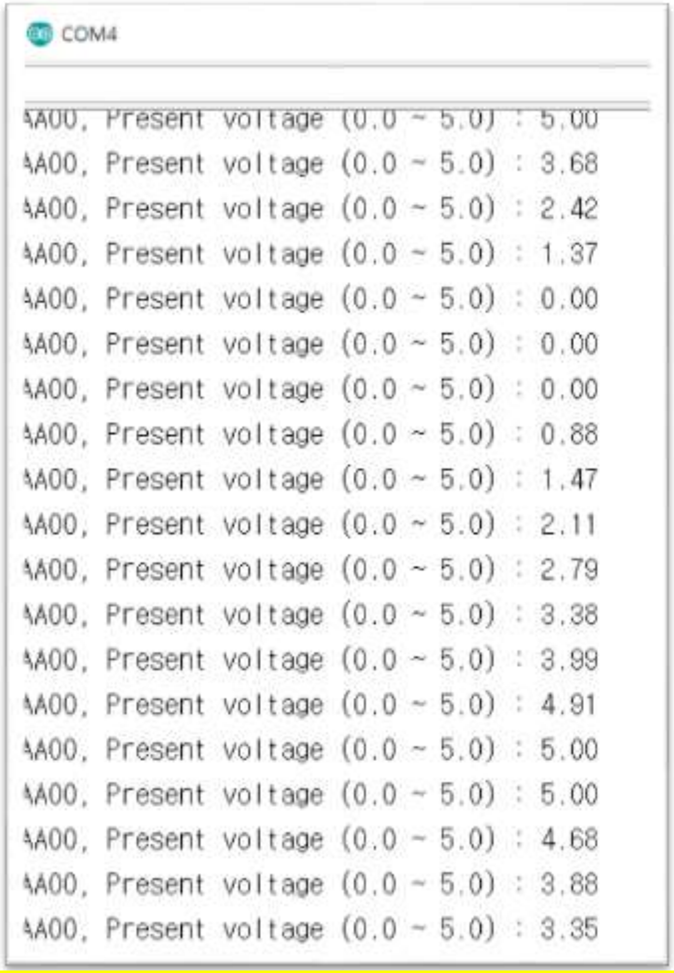
A2.5.1 AnalogReadSerial (circuit)

Standard potentiometer (가변 저항기)

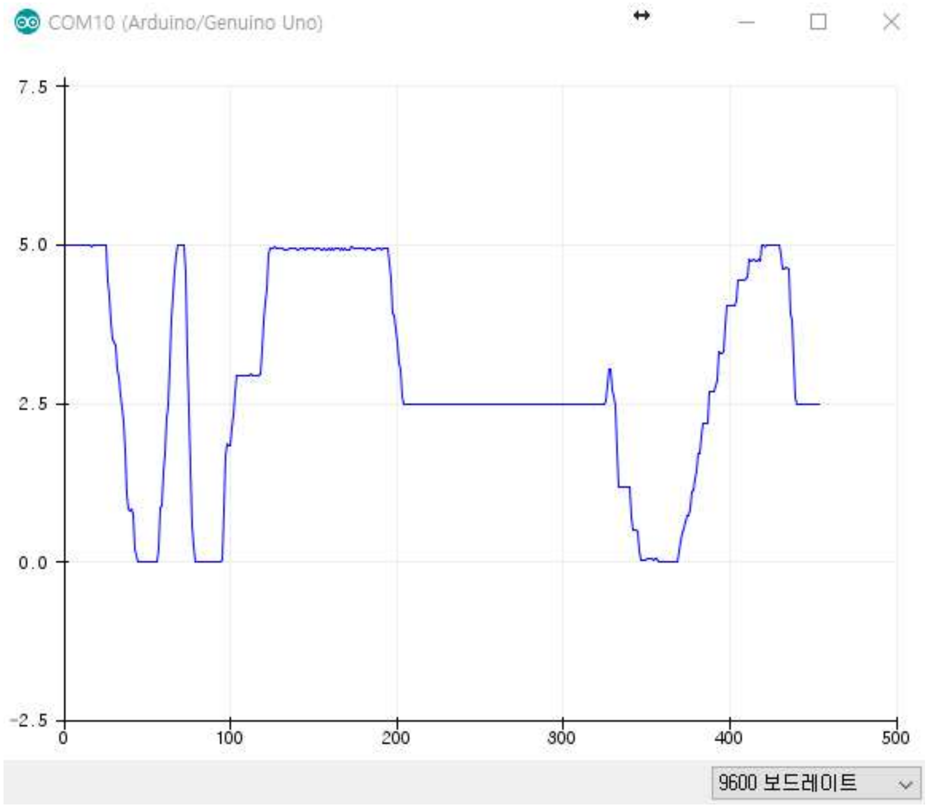
A2.5.7 ReadAnalogVoltage

Result



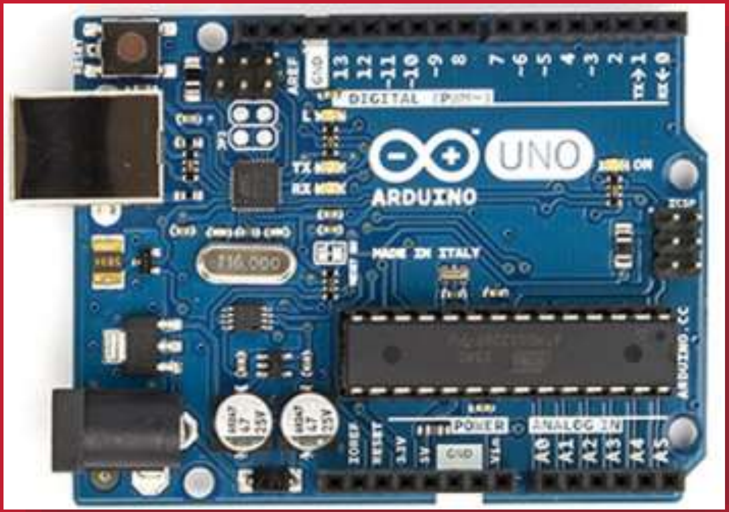
COM4

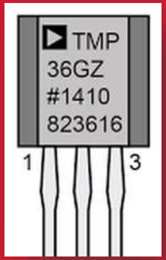
```
AA00, Present voltage (0.0 ~ 5.0) : 5.00
AA00, Present voltage (0.0 ~ 5.0) : 3.68
AA00, Present voltage (0.0 ~ 5.0) : 2.42
AA00, Present voltage (0.0 ~ 5.0) : 1.37
AA00, Present voltage (0.0 ~ 5.0) : 0.00
AA00, Present voltage (0.0 ~ 5.0) : 0.00
AA00, Present voltage (0.0 ~ 5.0) : 0.00
AA00, Present 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
```

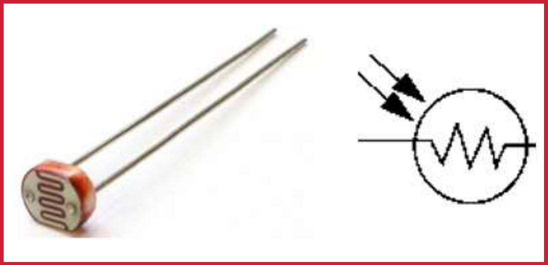
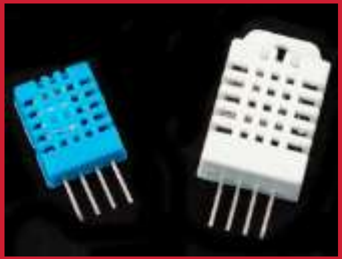


Save as
AAnn_AnalogVoltage.png

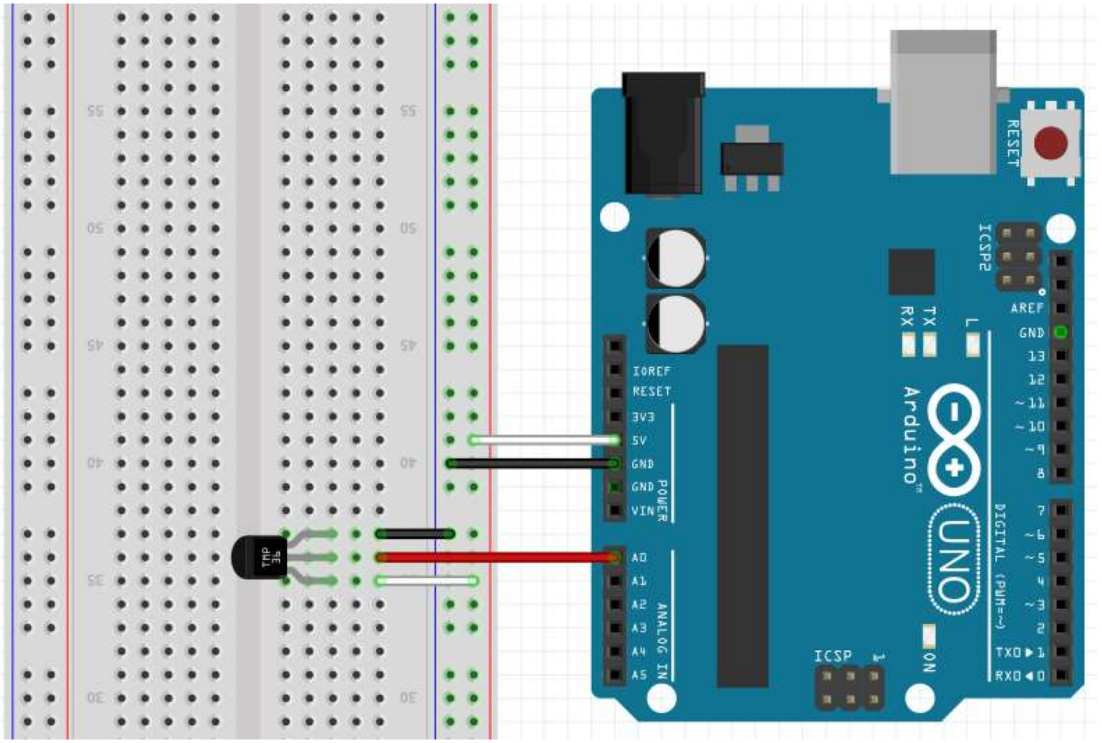
Arduino Sensors

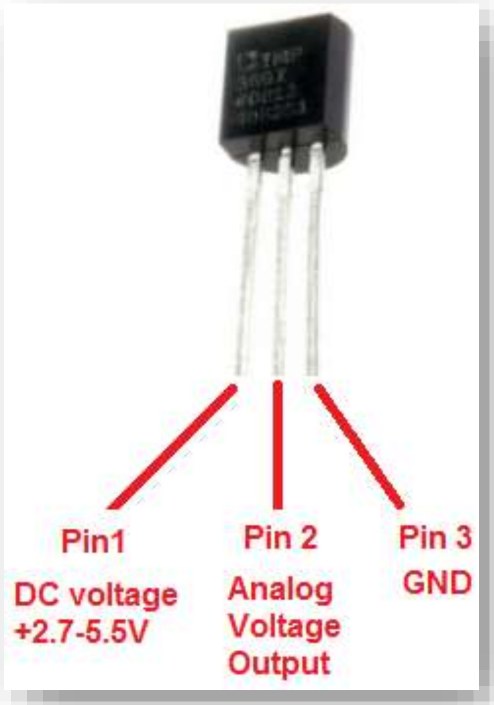




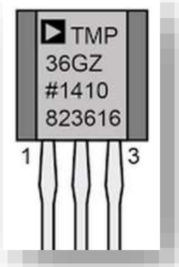



A3.1.1 Temperature sensor [TMP36]



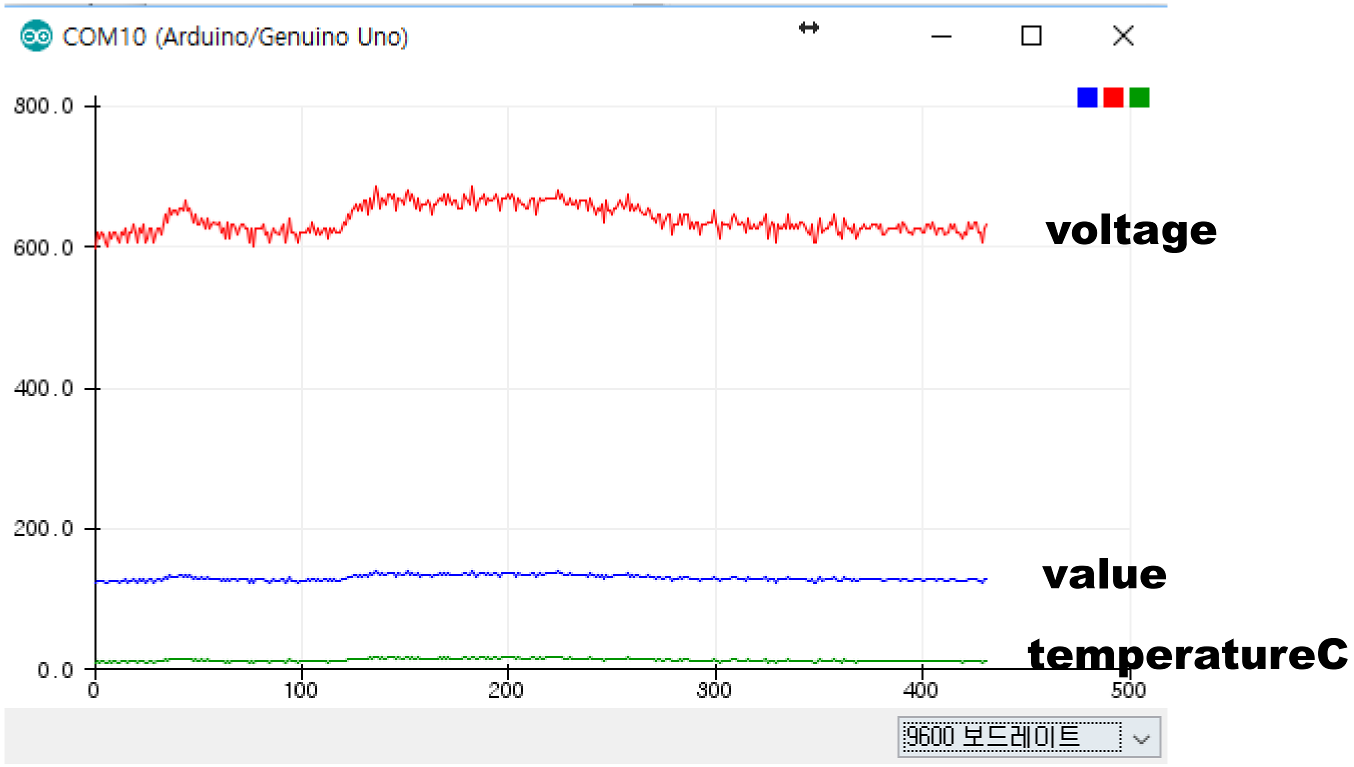


Parts : TMP36



- **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

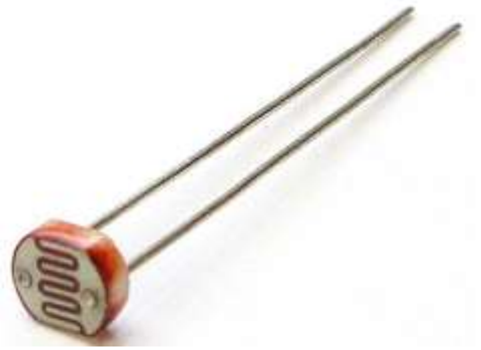
A3.1.5 Temperature sensor [TMP36]

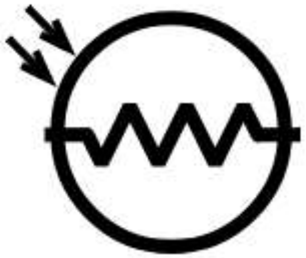


Save as
AAnn_TMP36.png

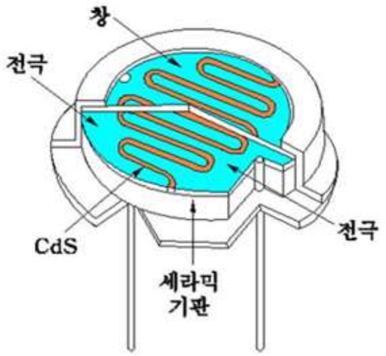
A3.2 Luminosity sensor [Photocell LDR]

CdS 센서- photoresistor





CDS특성

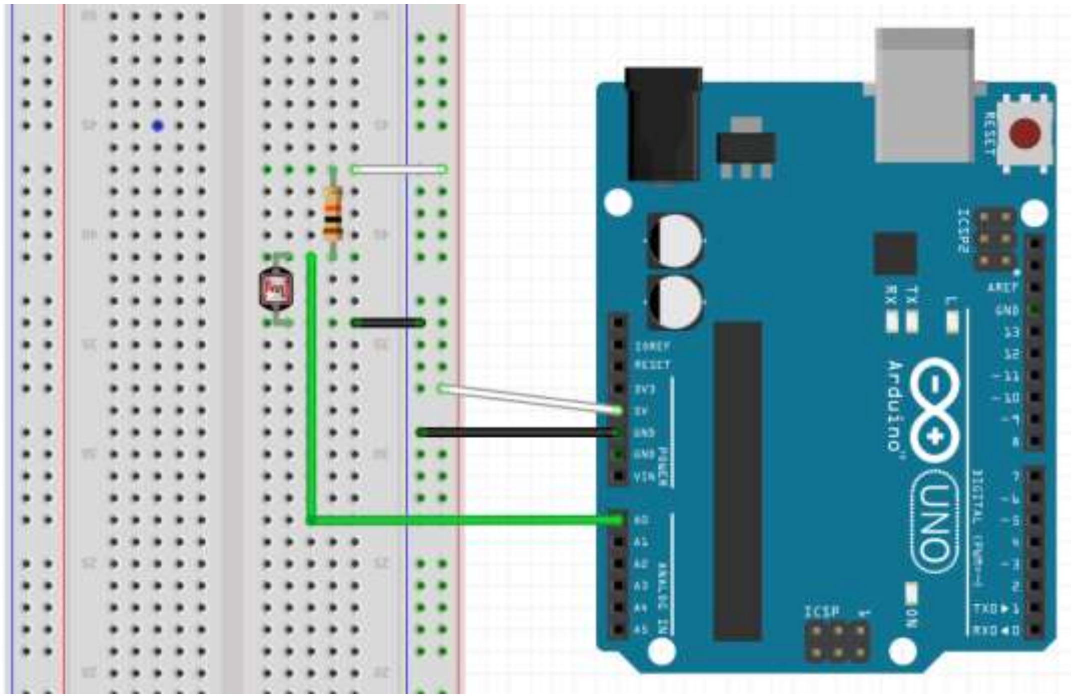


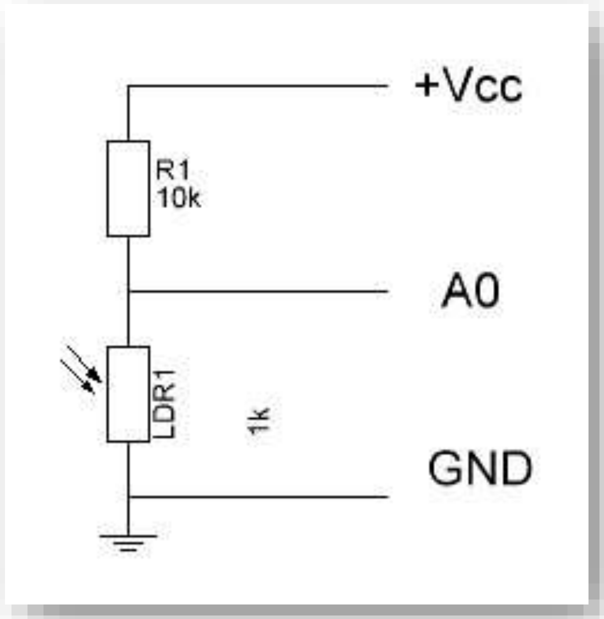
1. 감도
   - 빛의 파장에 따라 감도가 다름
2. 허용손실
   - 비교적 큰 전류를 흘릴 수 있음
3. 암 전류
   - 빛이 없어도 약간의 전류가 흐름
4. 명 전류
   - 빛을 비추면 흐르는 전류
5. 응답특성
   - 응답 시간 지연
   - 빛의 세기에 따라 응답시간 다름
6. 가변저항
   - 빛에 따른 가변저항

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

A3.2.2 Luminosity sensor [Photocell LDR]

CdS 센서 회로



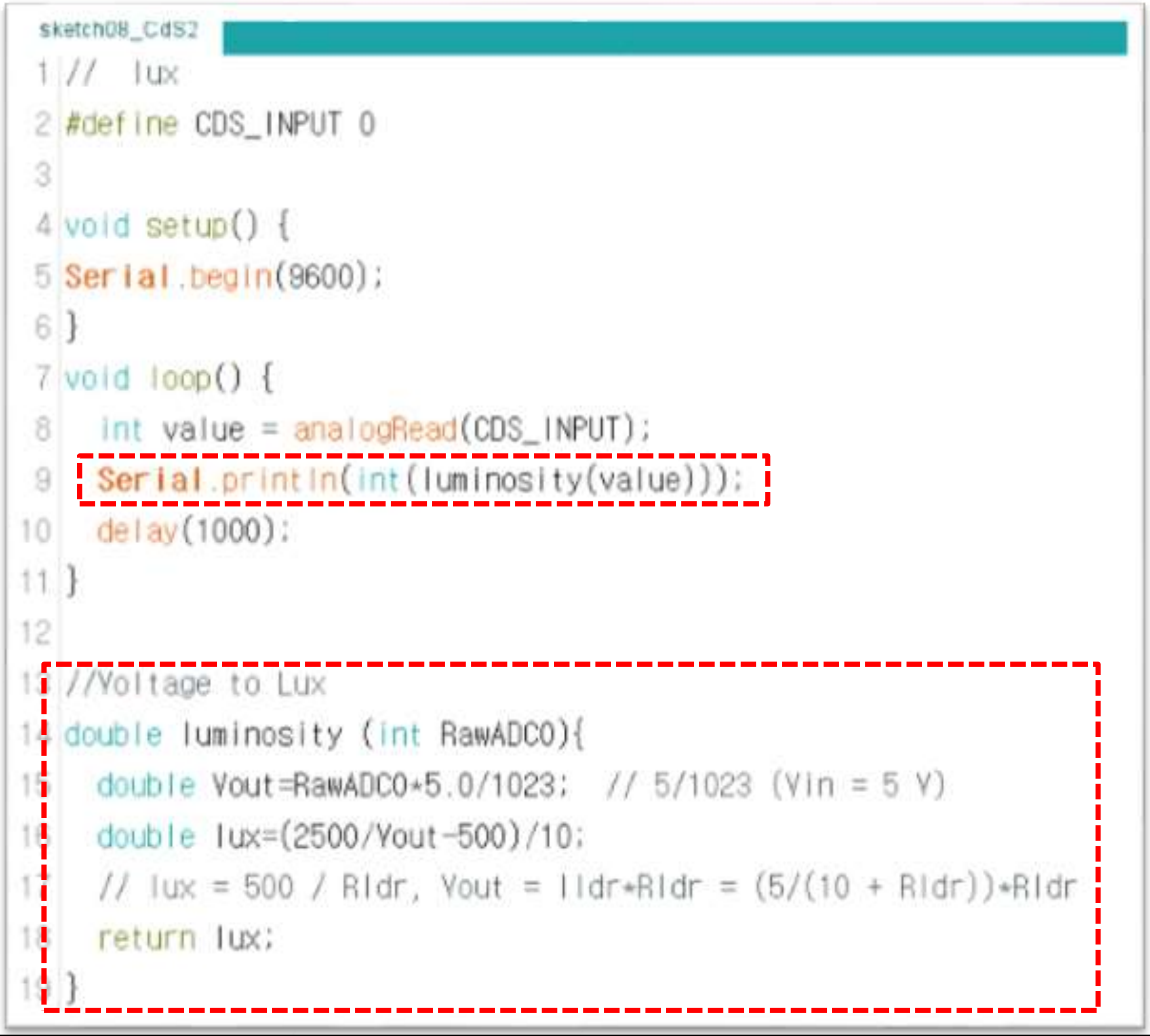


Parts : 20 mm photocell LDR, R (10 kΩ X 1)

광센서에서의 전압 강하 값을 **A0**로 측정

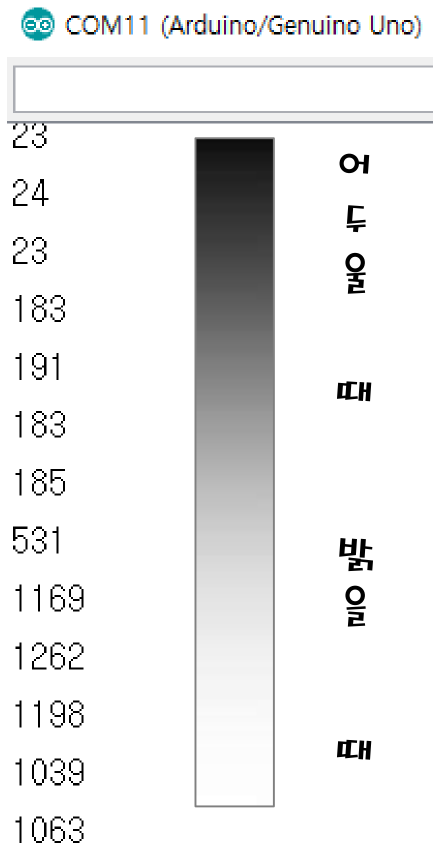
A3.2.6 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 2.



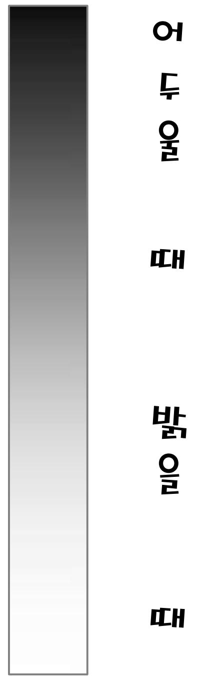
sketch08_CdS2

```
// lux
#define CDS_INPUT 0

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

//Voltage to Lux
double luminosity (int RawADC0){
  double Vout=RawADC0*5.0/1023;   // 5/1023 (Vin = 5 V)
  double lux=(2500/Vout-500)/10;
  // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
  return lux;
}
```



COM11 (Arduino/Genuino Uno)

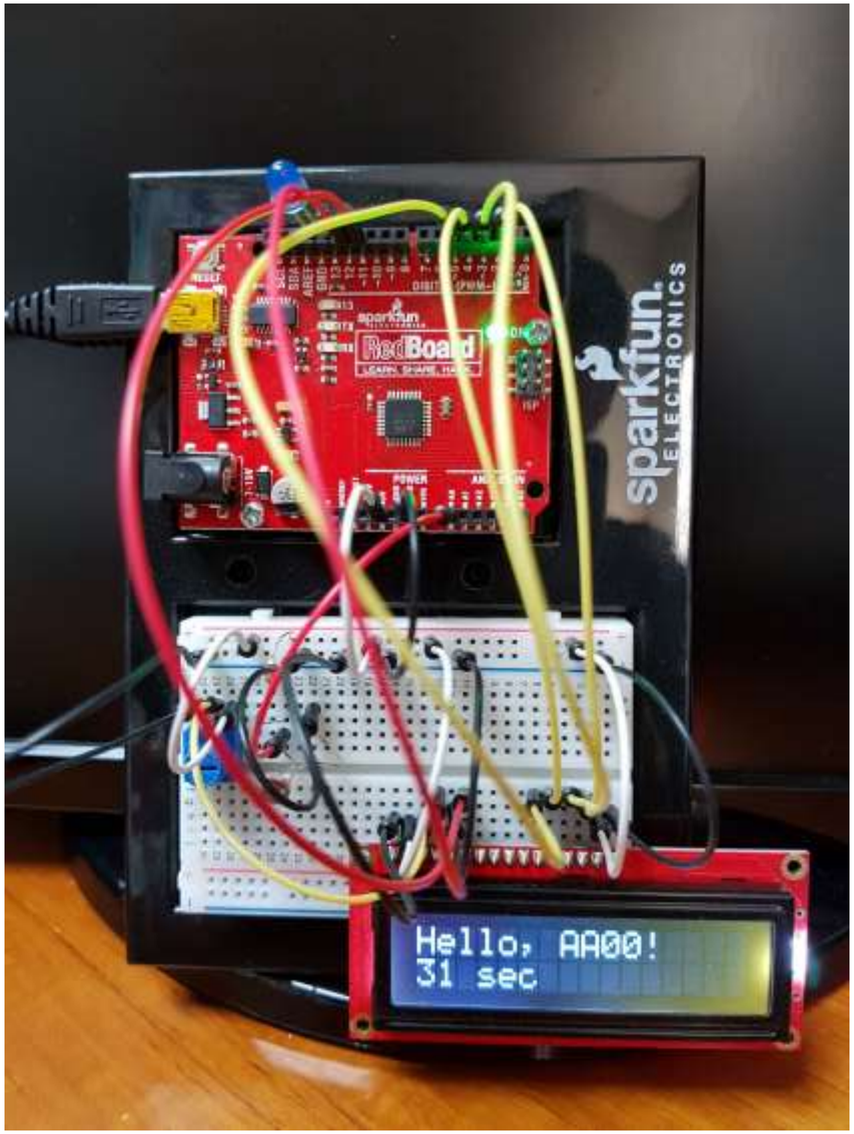
```
23
24
23
183
191
183
185
531
1169
1262
1198
1039
1063
```



밝을수록 측정 값이 커지고
어두울수록 값이 작아진다 !!!

Signal Monitoring via LCD

Introduction to LCD - “Hello AAnn”

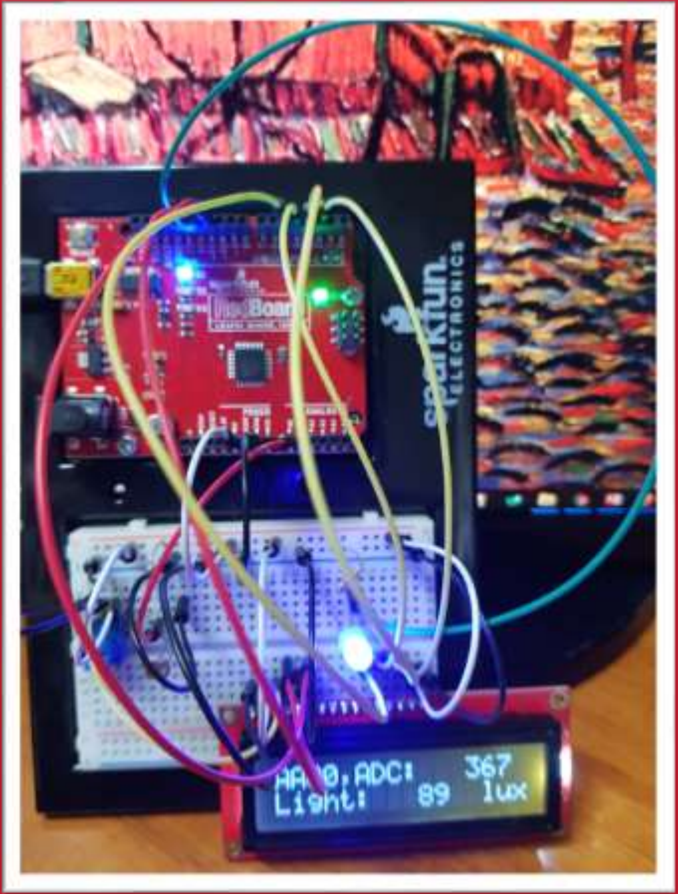


결과 화면 촬영: **AAnn_Hello_LCD.png** 로 저장...

CdS LCD Project

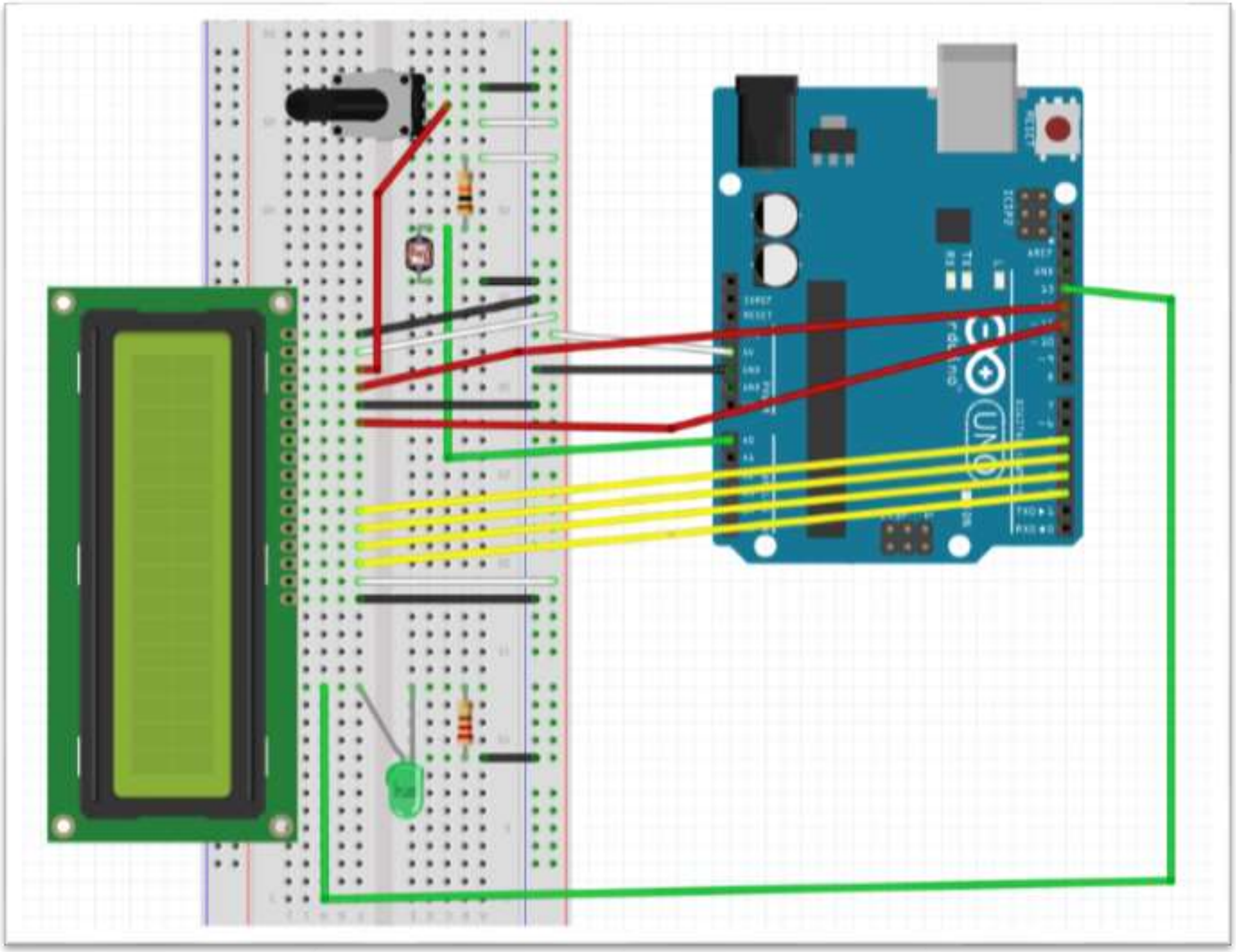
LCD에 조도 값을 표시하면서

조도에 따라 LED를 ON/OFF



CdS-LCD project : fzz circuit

CdS_LCD_LED.fzz

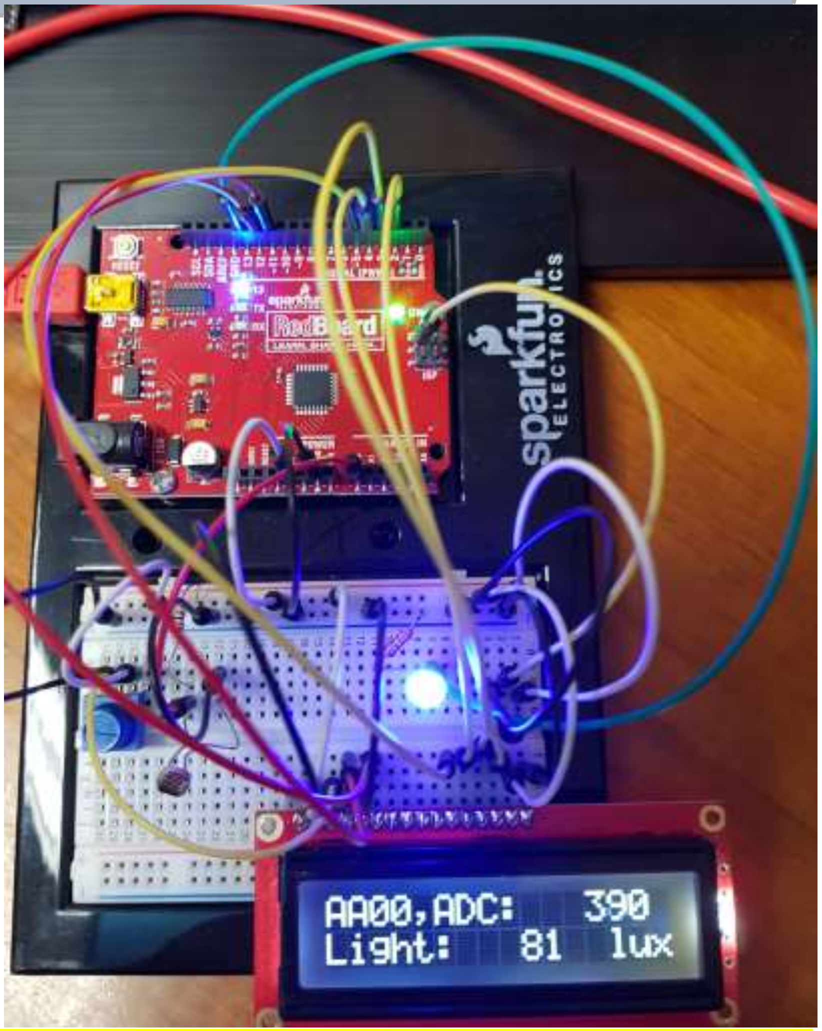


CdS-LCD project : result

CdS 센서 LCD 회로 - 측정 결과

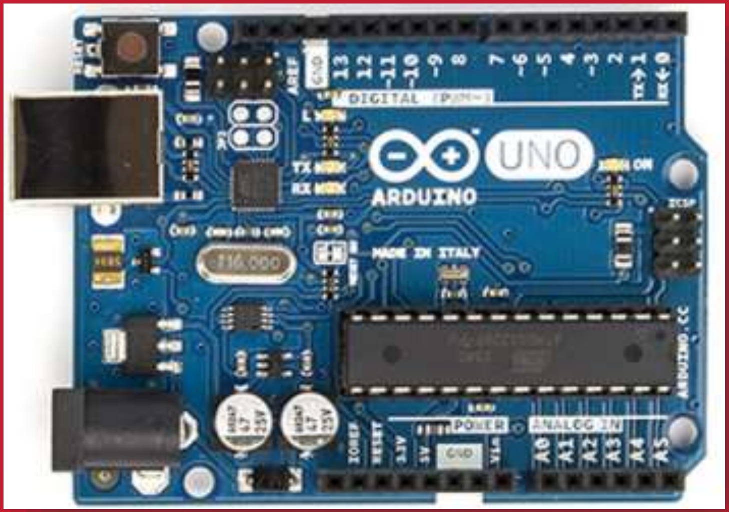
주변의 조도에 따라 어두우면 LED가 켜지고, 밝으면 LED가 꺼지도록 코드를 수정하시오.

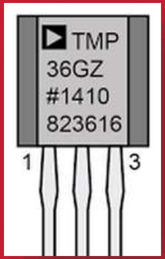
LED가 켜진 화면을 폰으로 촬영해서 그림을 제출하시오.

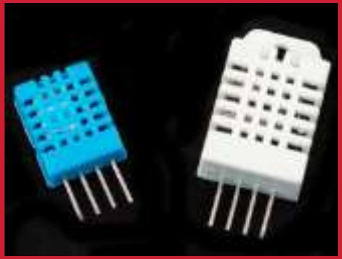


조도에 따라 LED가 ON/OFF 되는 것을 확인 받고
결과 화면 촬영: AAnn_LCD_lux.png 로 저장...

Arduino & Node.js







IOT: HSC

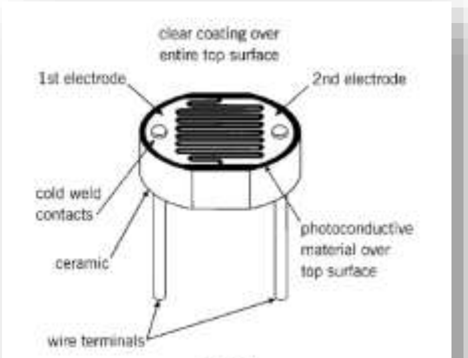
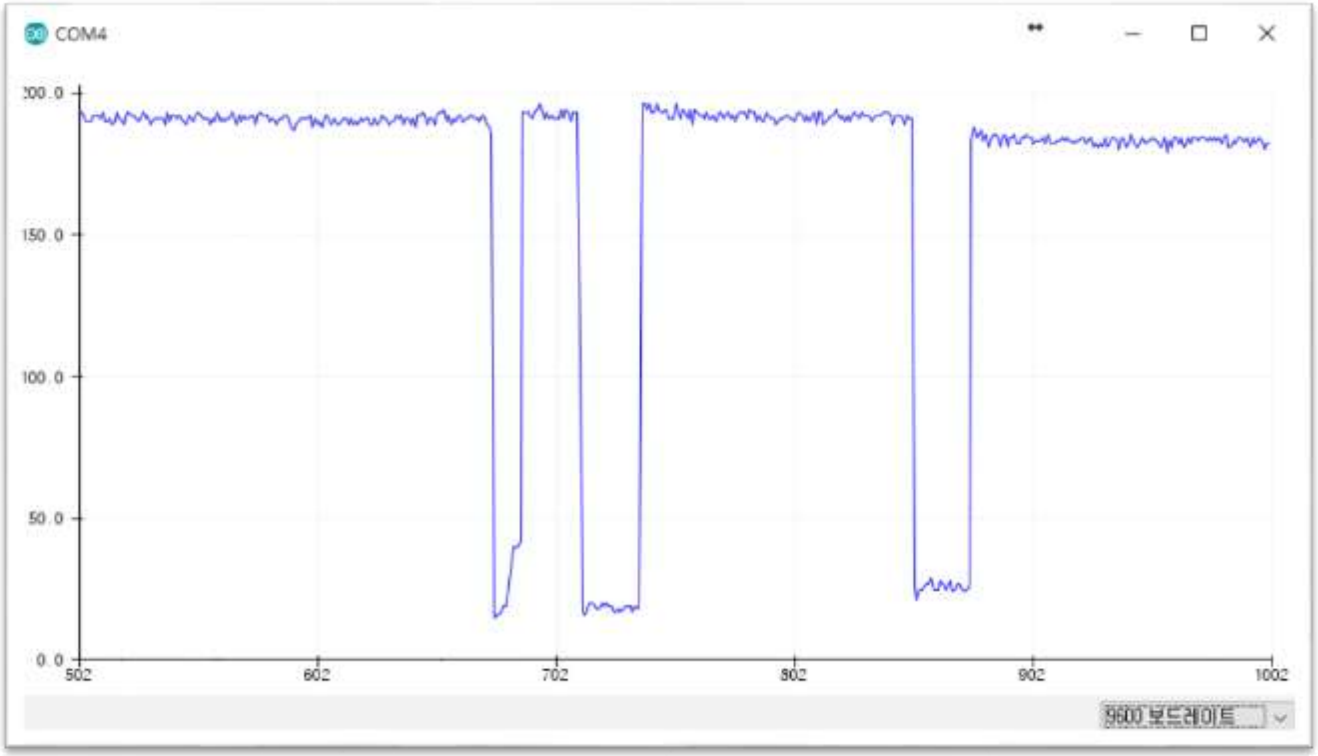
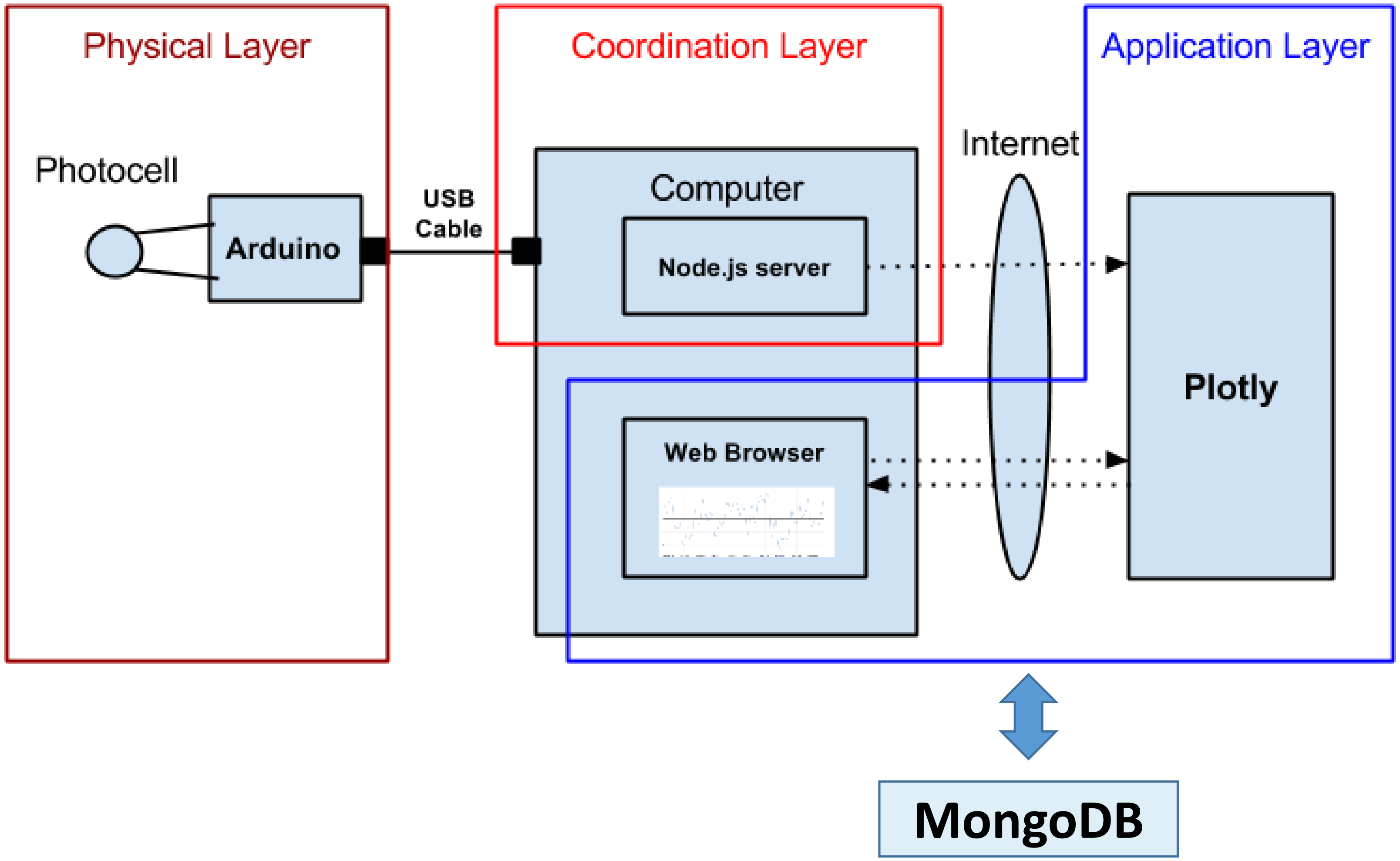


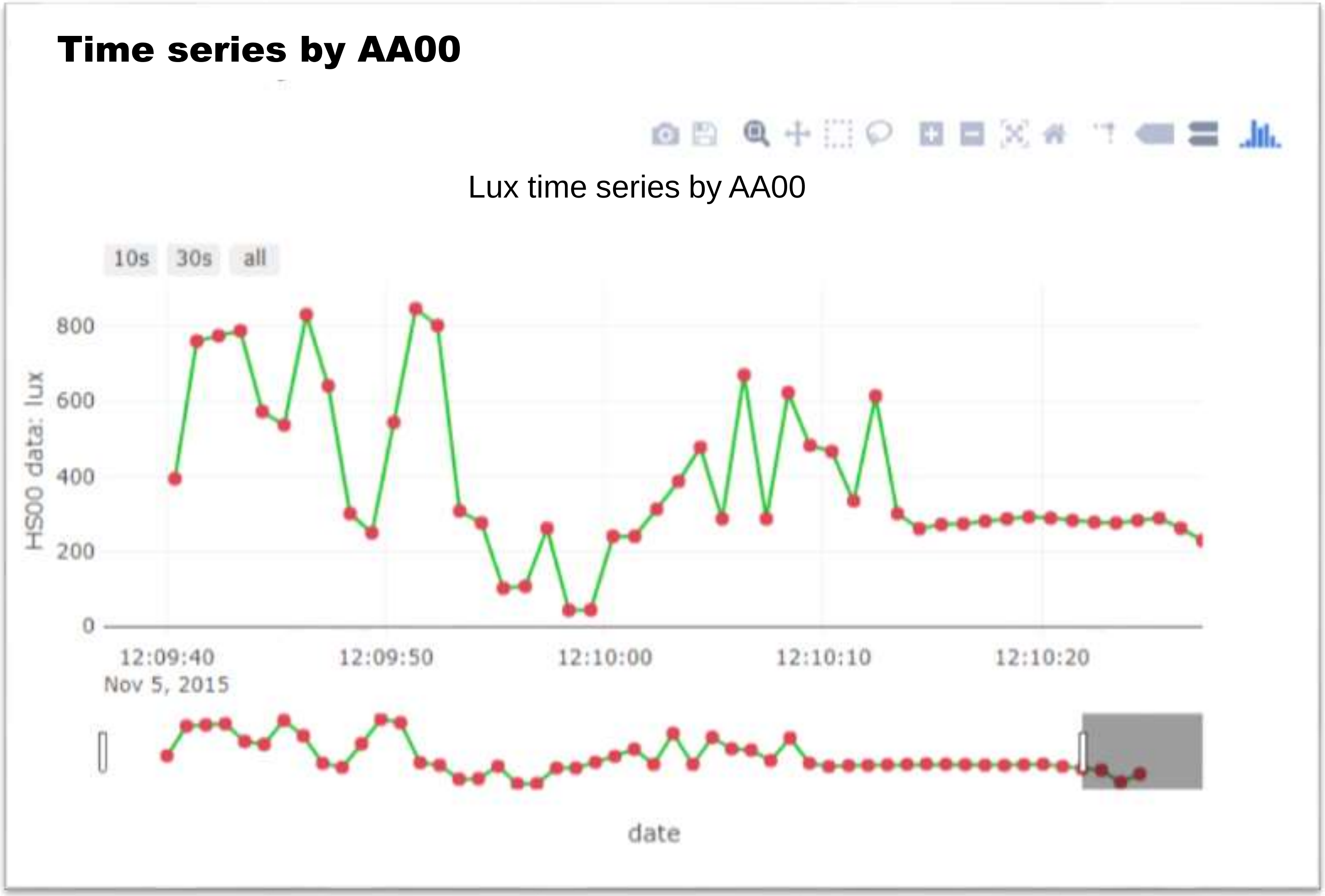
Figure 3
Typical Construction of a Plastic Coated Photocell



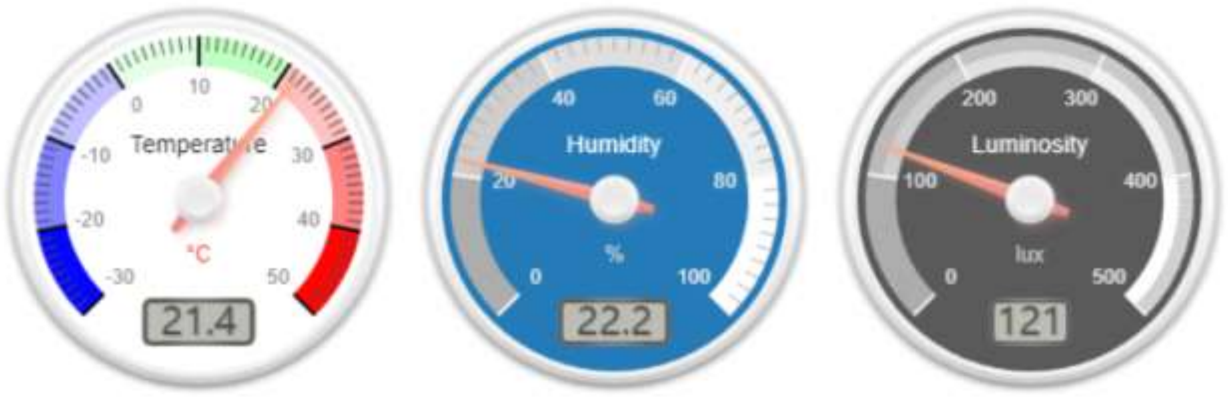
Layout [H S C]


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

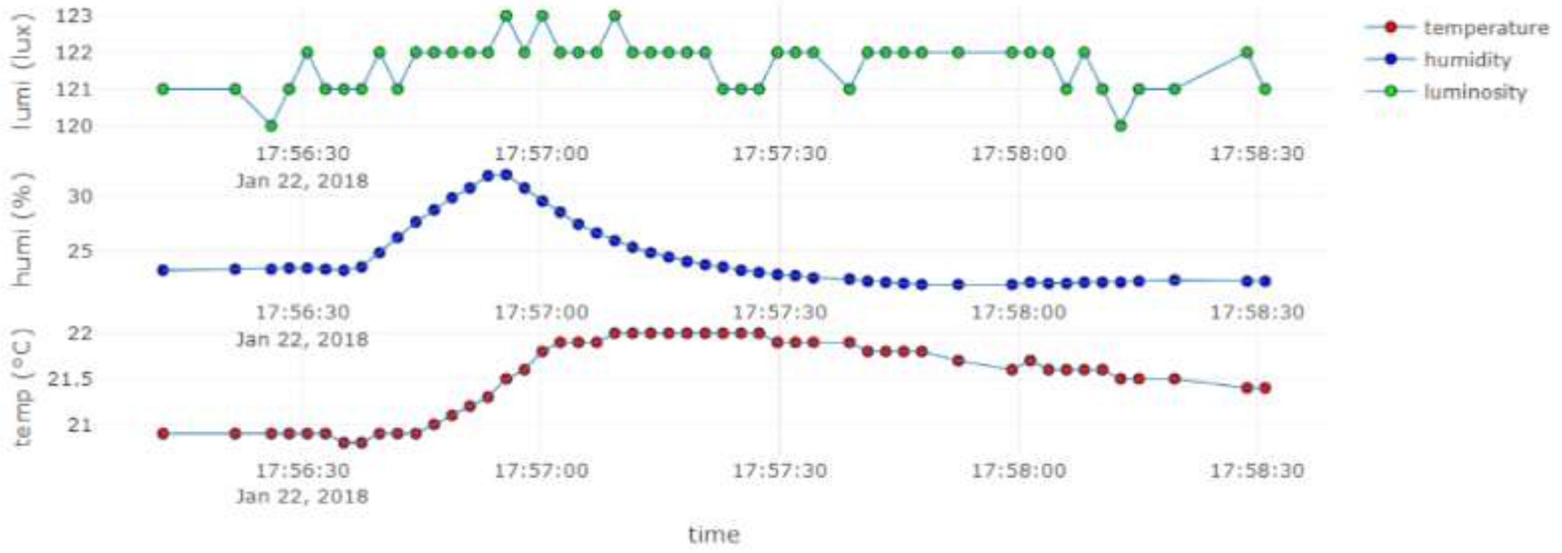
Arduino data + plotly



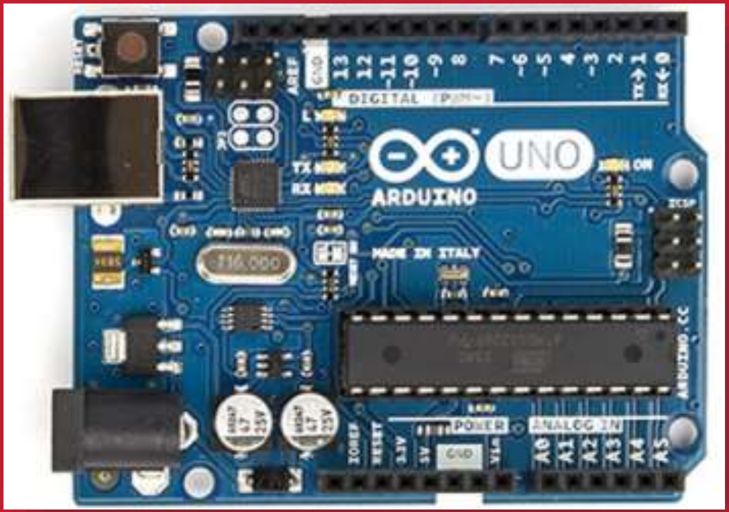
Real-time Weather Station from sensors

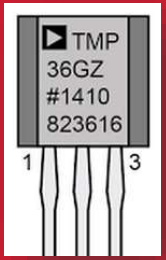


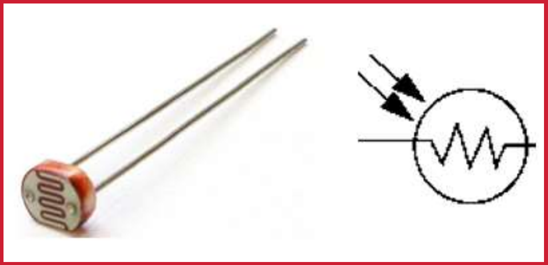
on Time: 2018-01-22 17:58:31.012

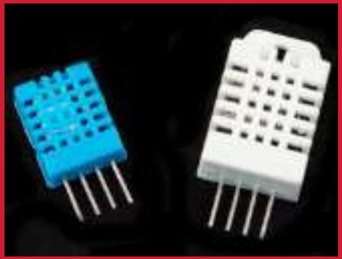


Arduino Sensors + Node.js

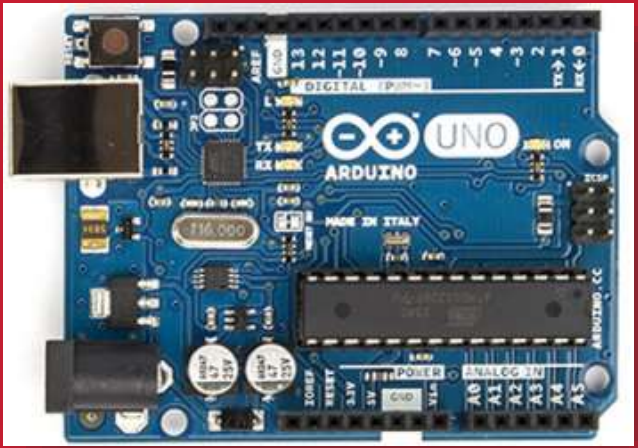




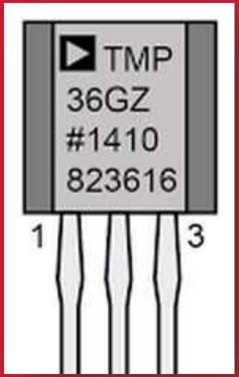




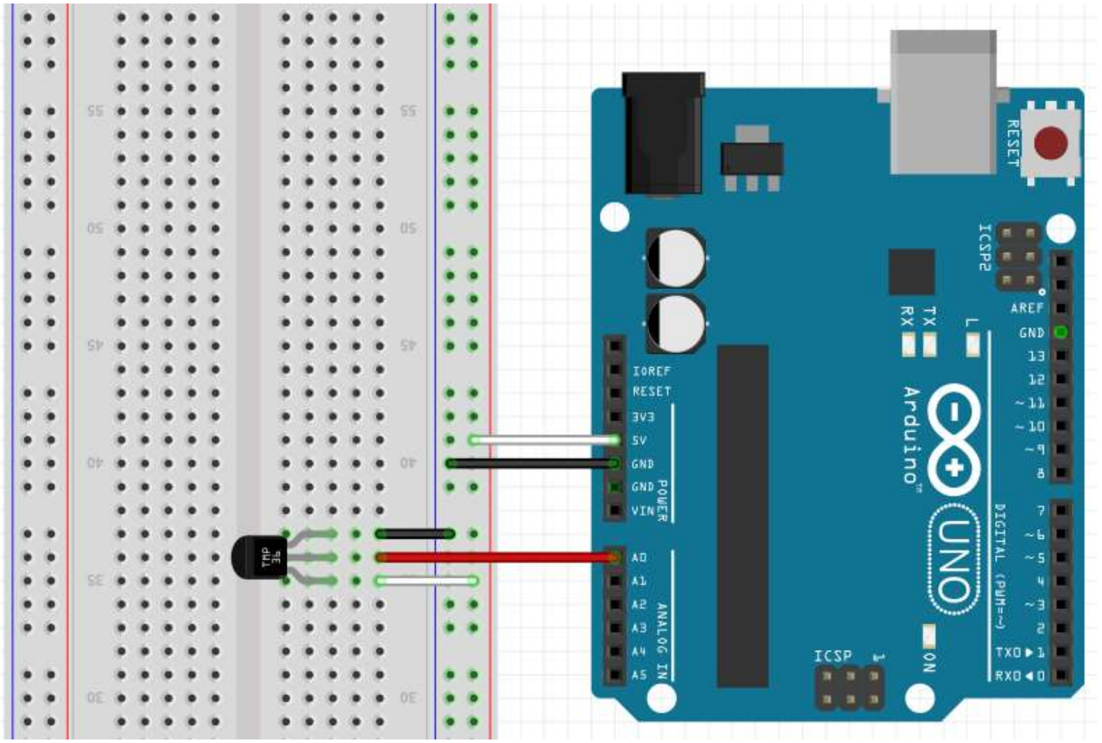
Single sensor: TMP36

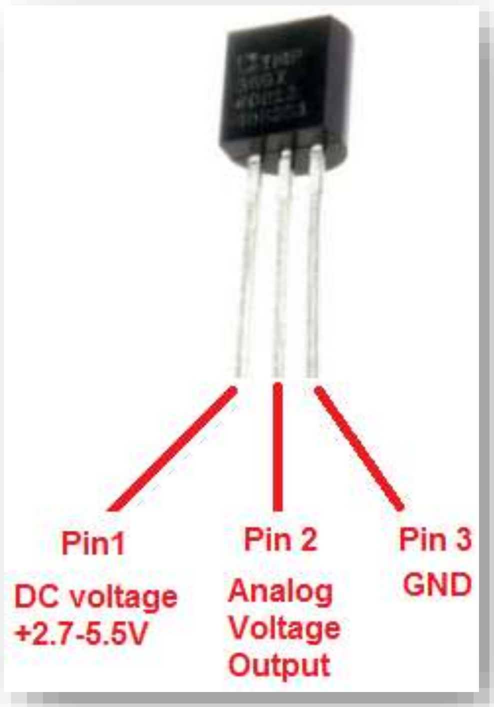


Arduino + Node.js

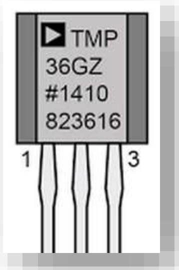


A3.1.1 Temperature sensor [TMP36]





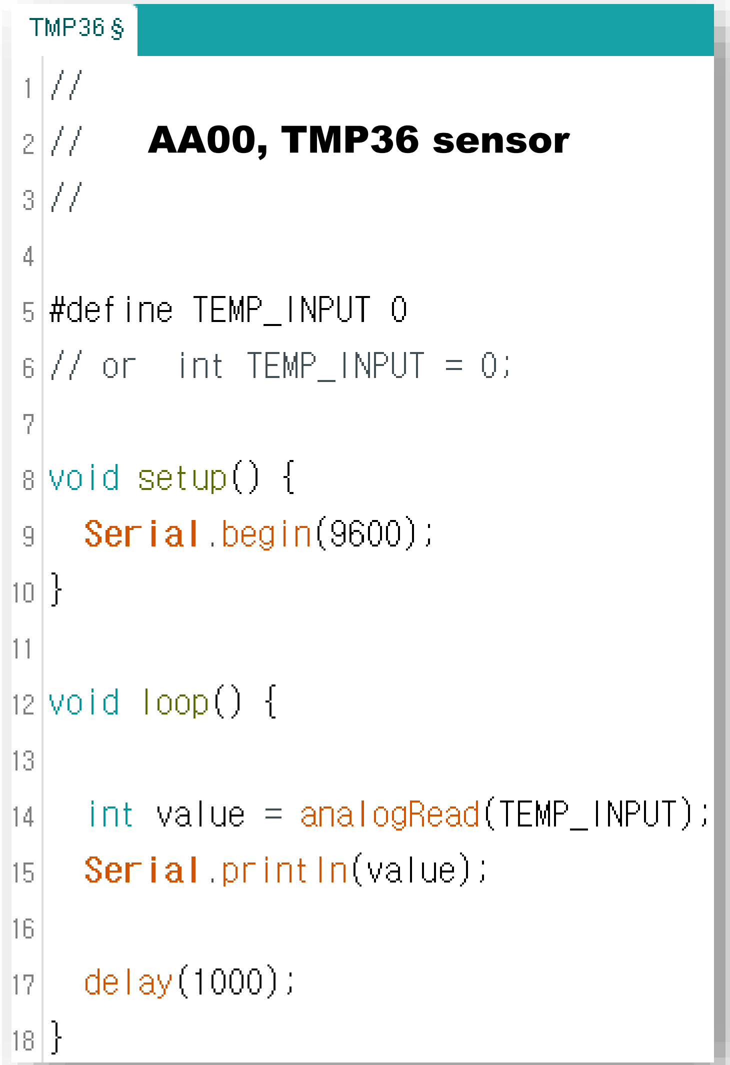
Parts : TMP36



- **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]

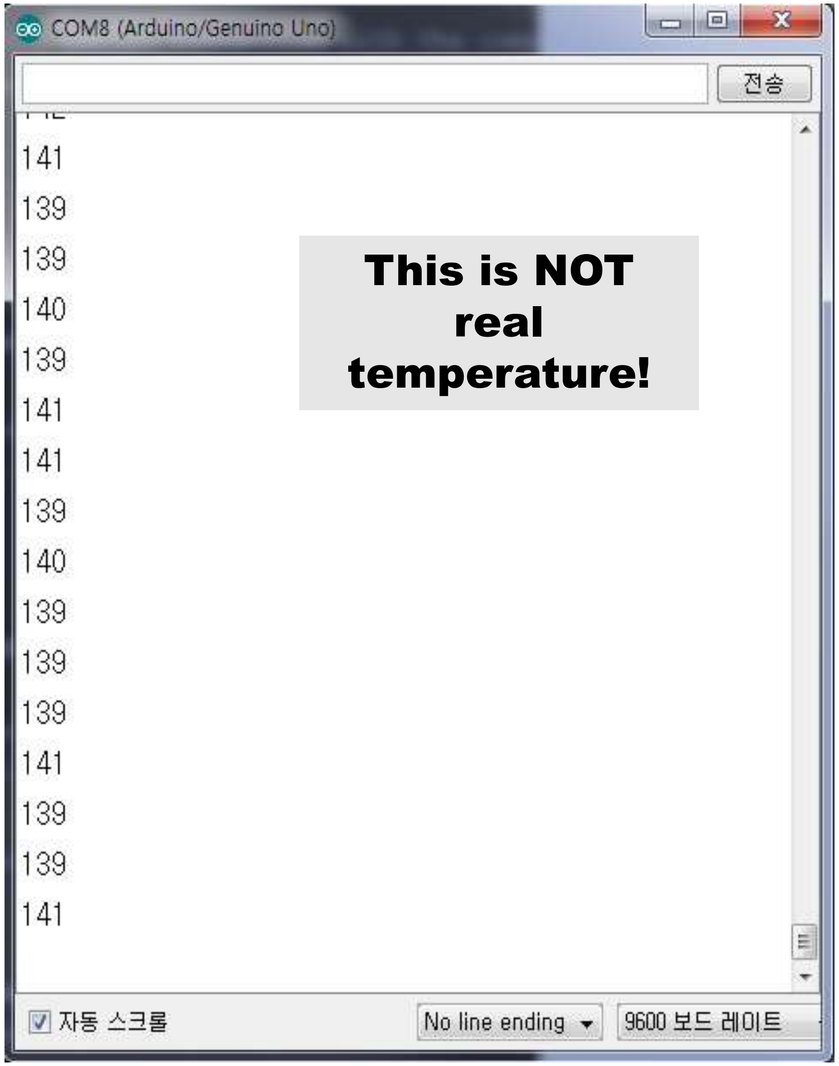
Simple code



TMP36 §

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

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

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

void loop() {

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

  delay(1000);
}
```

Serial output (0 ~ 1023)



COM8 (Arduino/Genuino Uno)

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

This is NOT real temperature!

A3.1.3 Temperature sensor [TMP36]

Sensor property

Temperature conversion

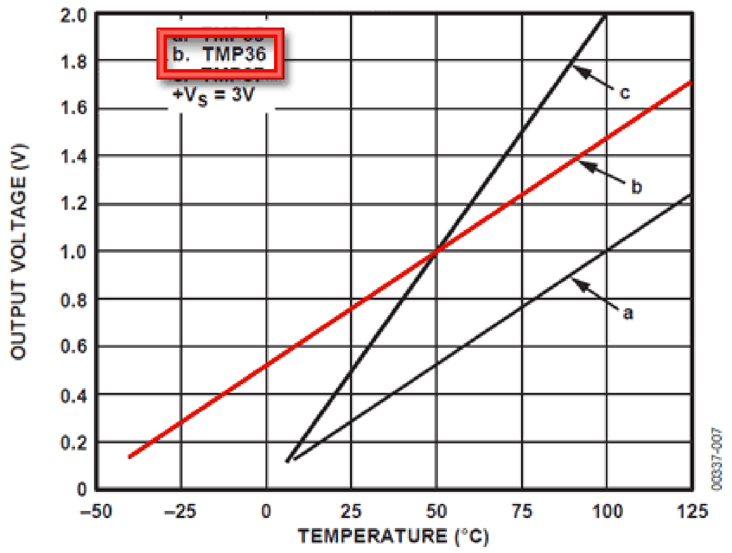


Figure 6. Output Voltage vs. Temperature

Temp (° C) = (Vout – 500) / 10

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

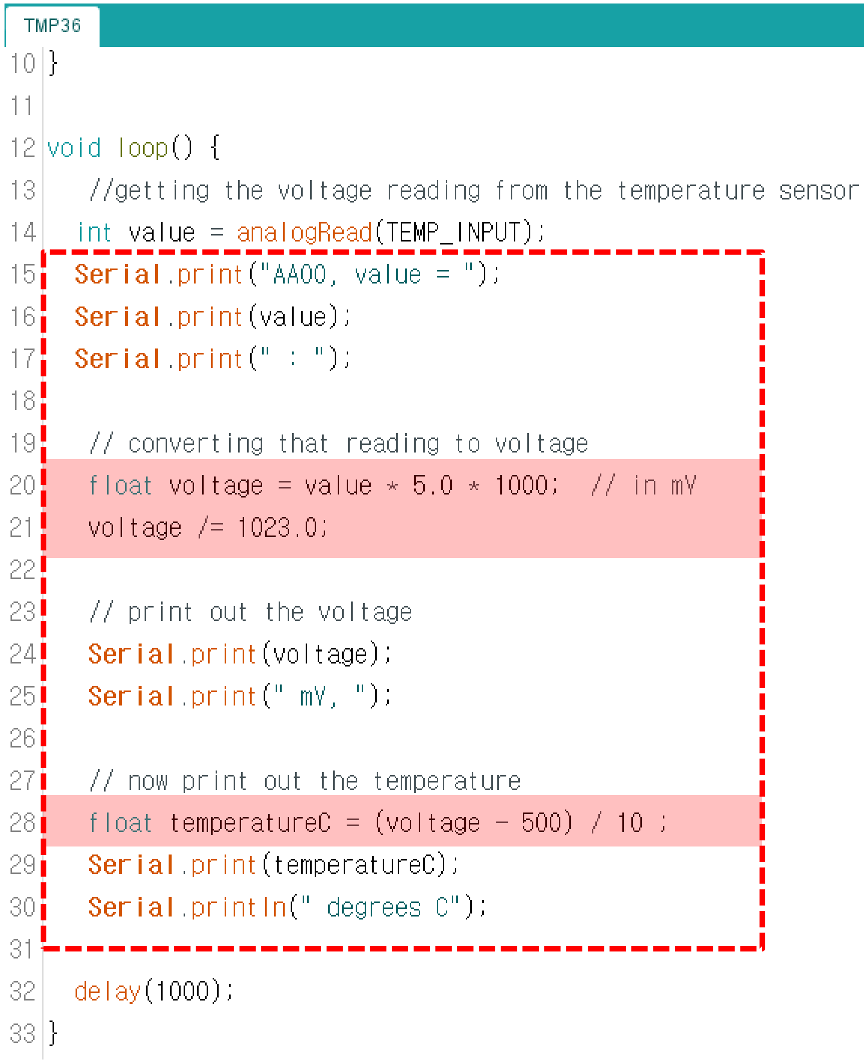
(0 <= value <= 1023)

```
// 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]

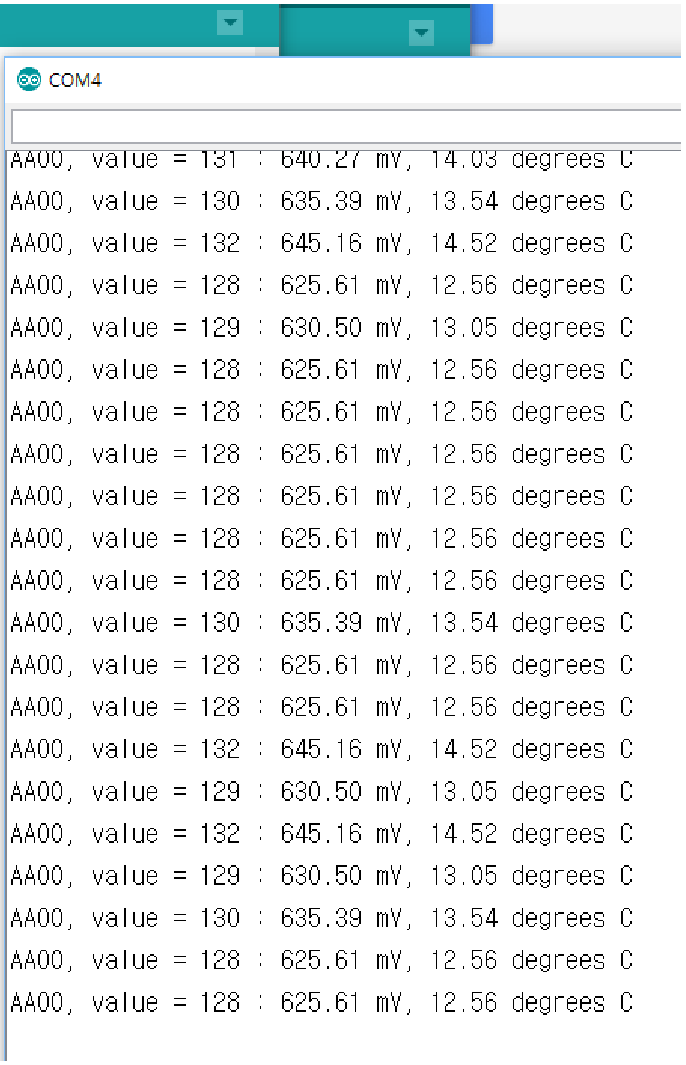
Working code

Serial output (°C)



TMP36

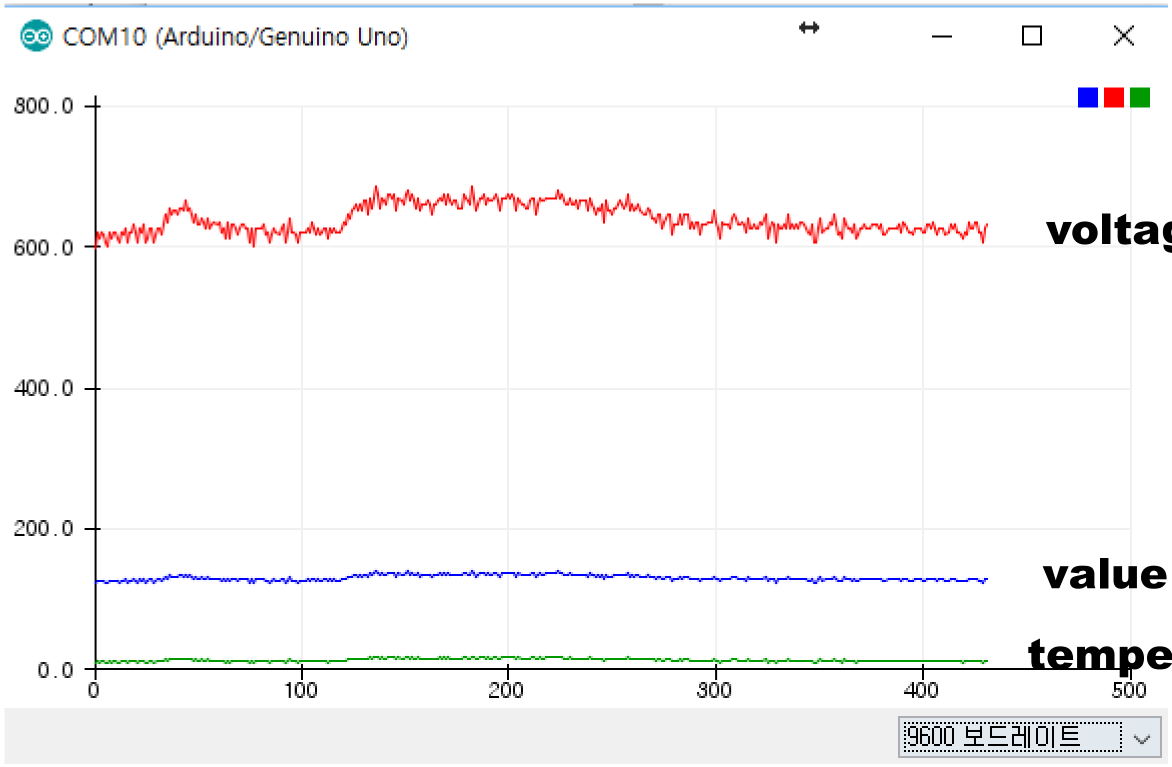
```
}

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);
}
```



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]

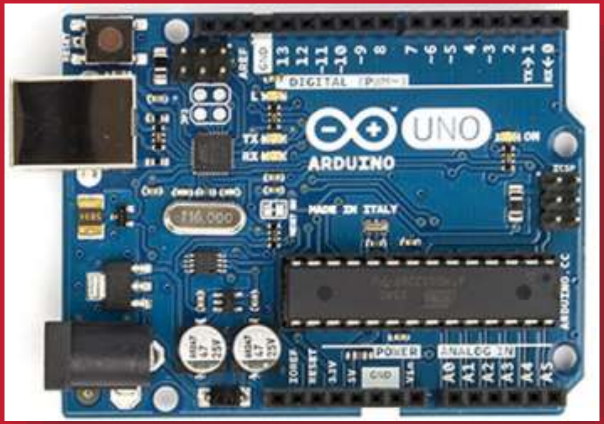


voltage

value

temperatureC

Single sensor: tmp36



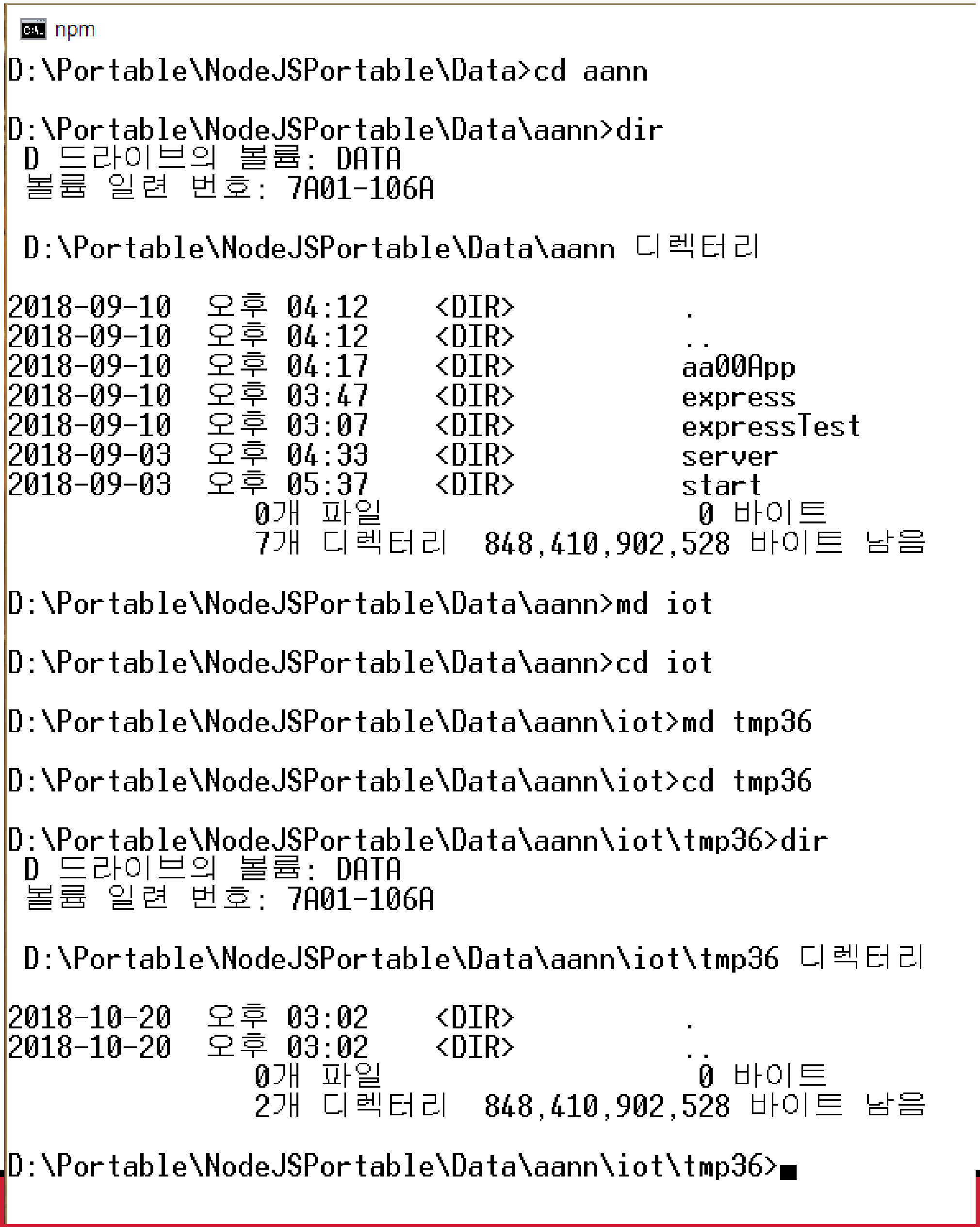
TMP36

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

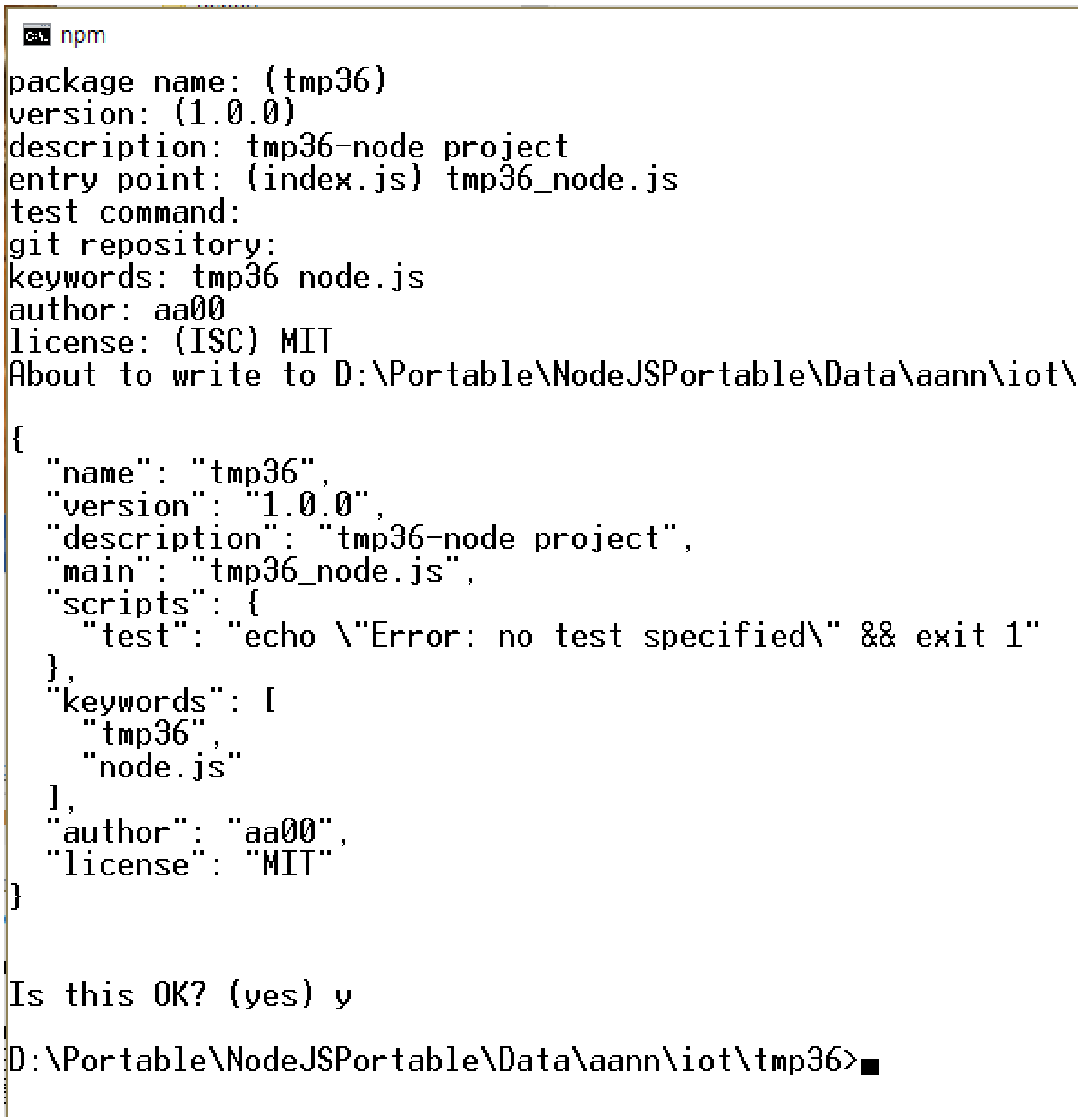
Set tmp36-node project

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

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

   tmp36_node.js
4. **author**

   your id : aann

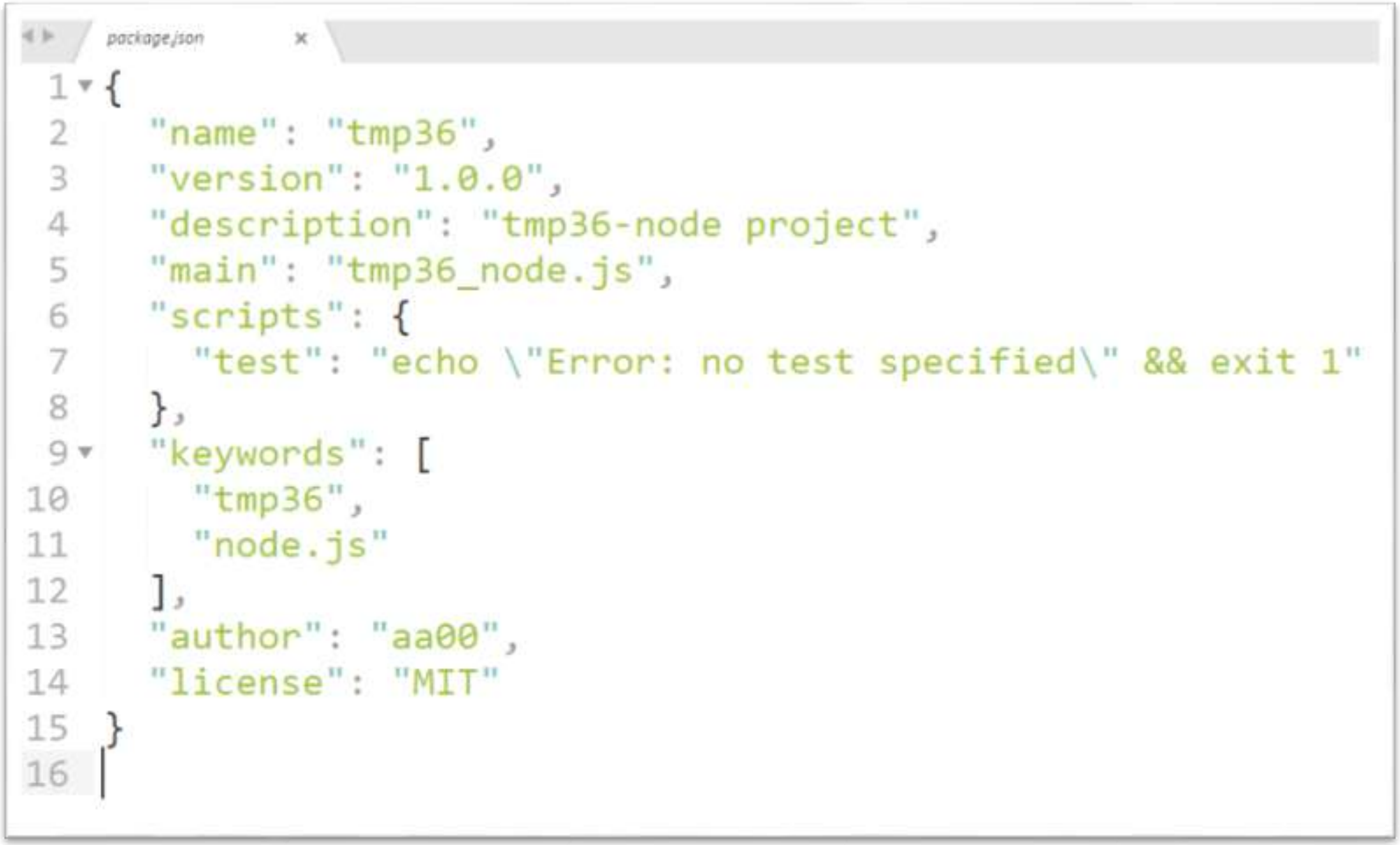


npm

```
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

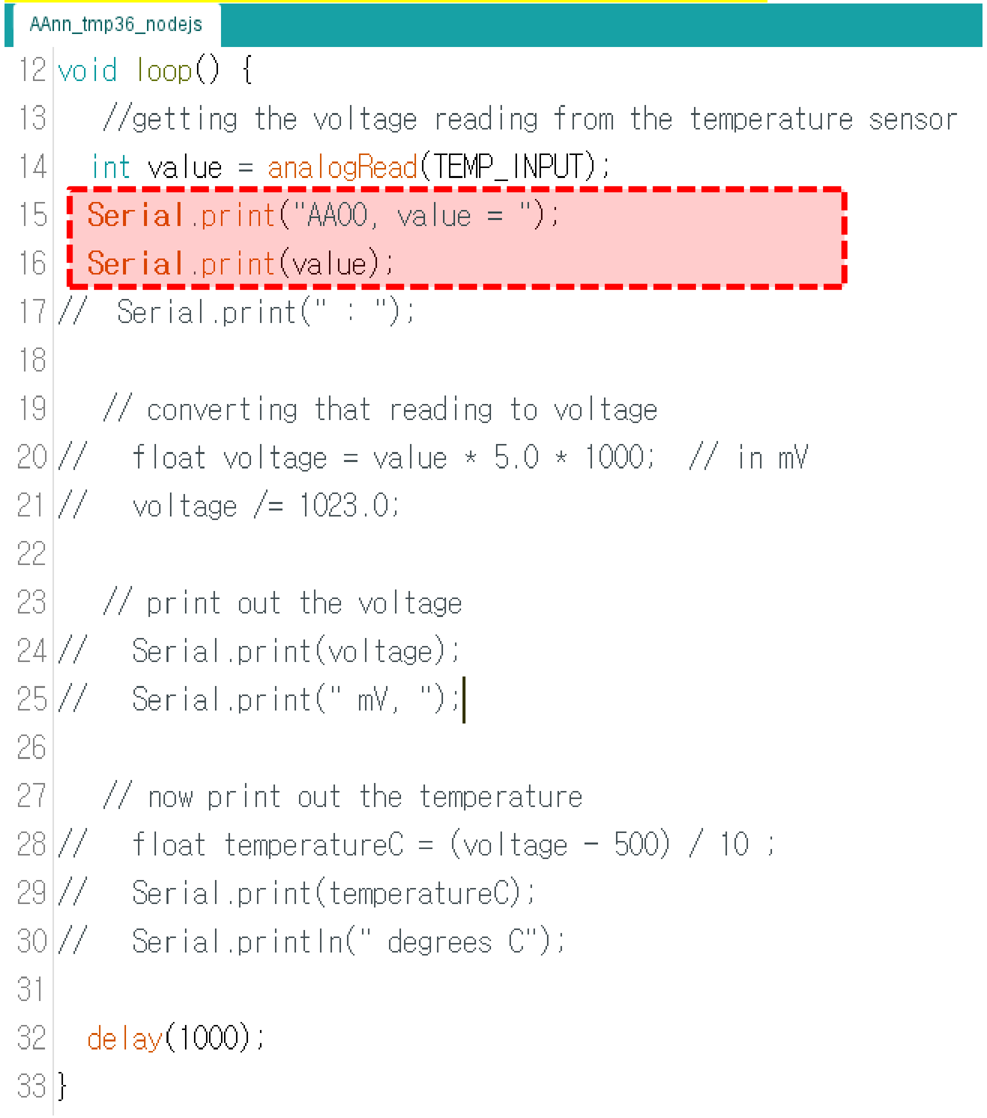
package.json


package.json

```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"
}

```

A4.1.4 tmp36 node project

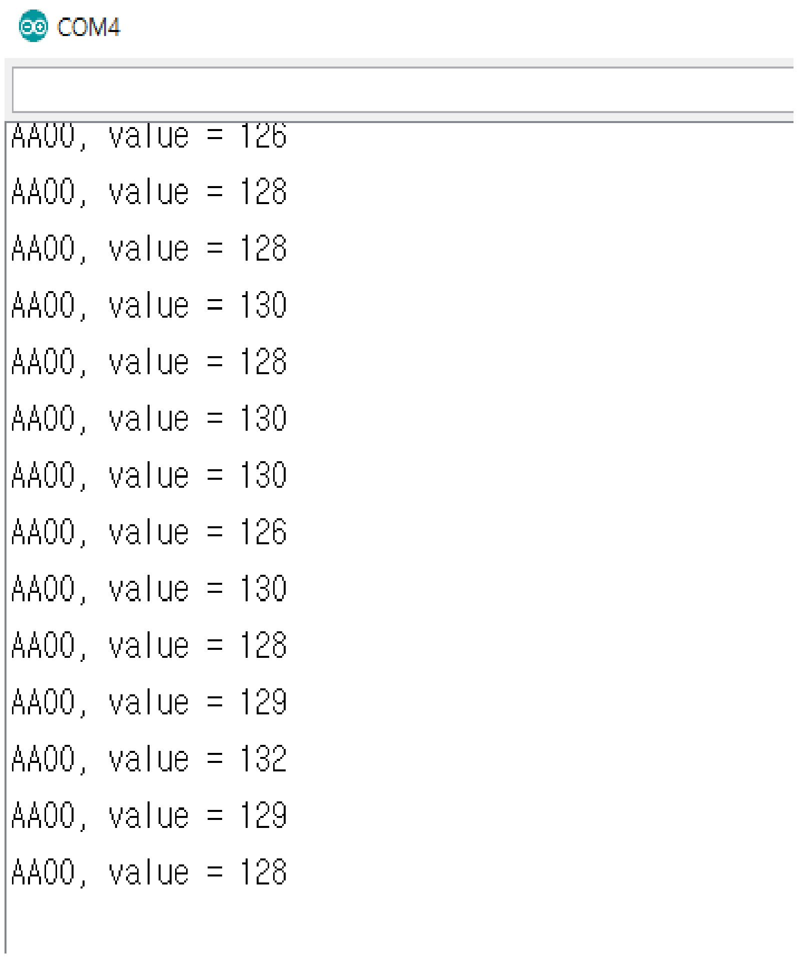
AAnn_TMP36_NodeJS.ino



AAnn_tmp36_nodejs

```
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 (A0, 0 ~ 1023)



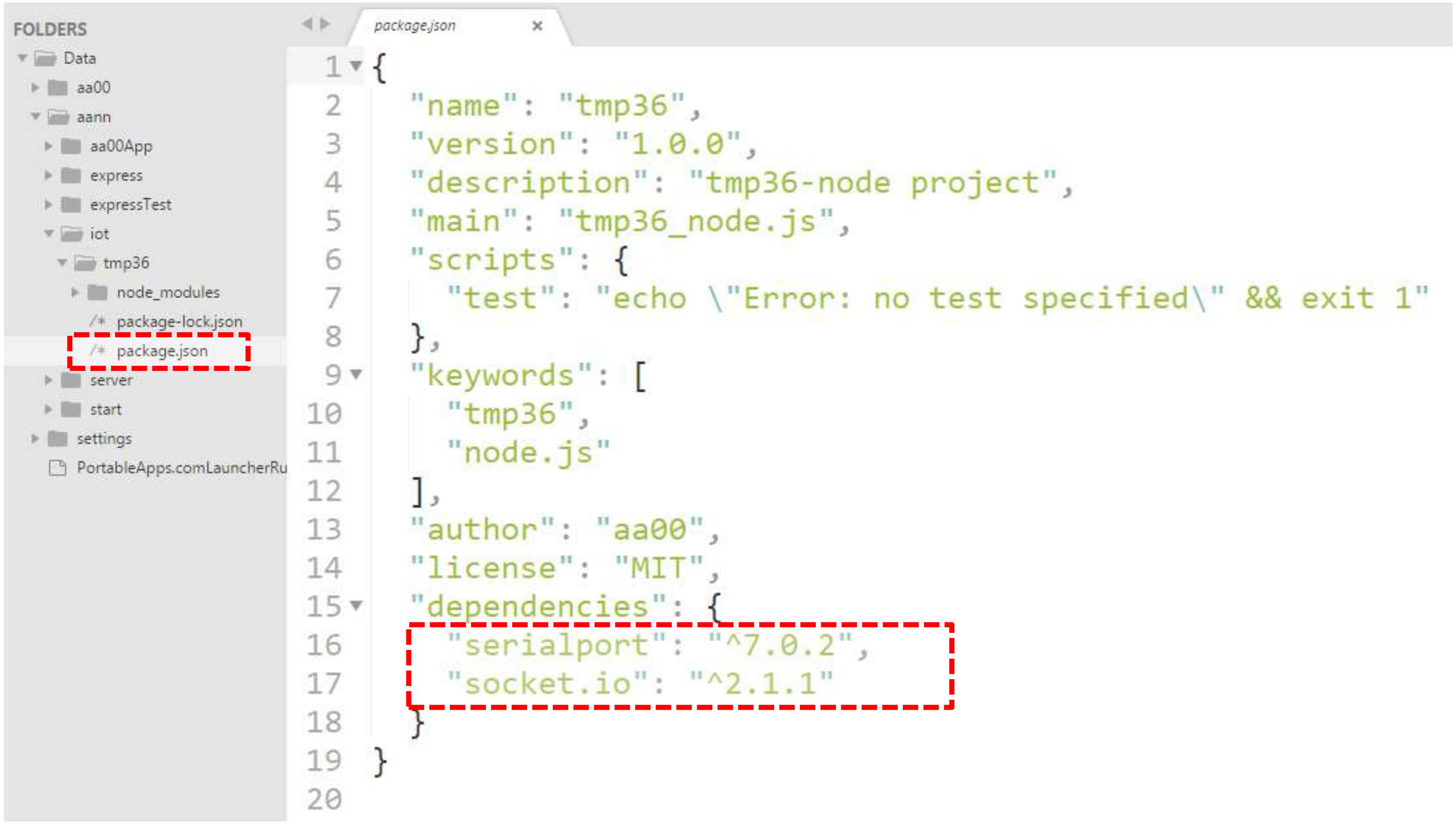
COM4

```
AA00, value = 126
AA00, value = 128
AA00, value = 128
AA00, value = 130
AA00, value = 128
AA00, value = 130
AA00, value = 130
AA00, value = 126
AA00, value = 130
AA00, value = 128
AA00, value = 129
AA00, value = 132
AA00, value = 129
AA00, value = 128
```

A4.1.5 tmp36 node project

Go to tmp36 subfolder

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

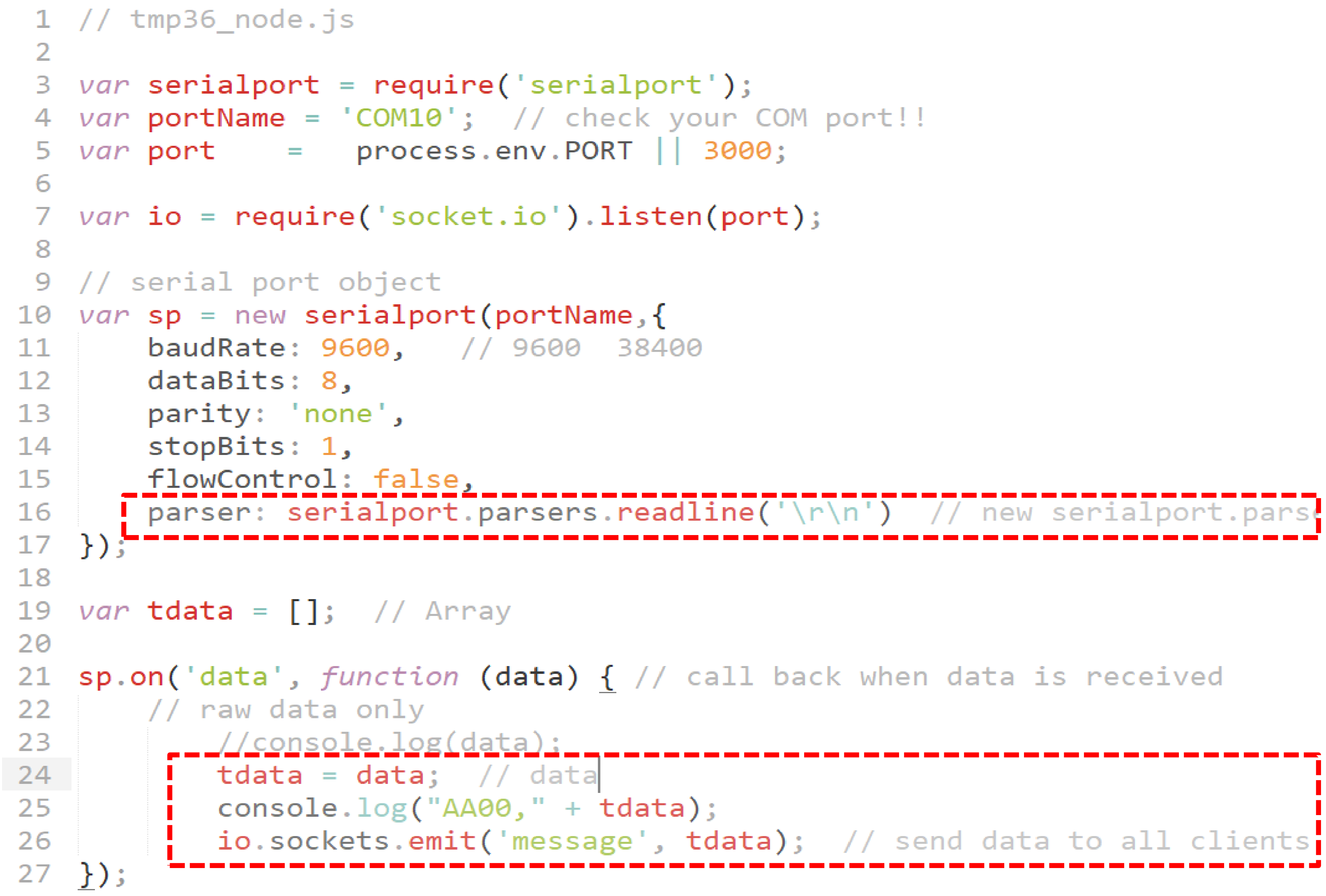

FOLDERS
- Data
  - aa00
  - aann
    - aa00App
    - express
    - expressTest
    - iot
      - tmp36
        - node_modules
        - package-lock.json
        - package.json
    - server
    - start
  - settings
  - PortableApps.comLauncherRu

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",
  "dependencies": {
    "serialport": "^7.0.2",
    "socket.io": "^2.1.1"
  }
}
```

A4.1.6 tmp36 node project : code-1

tmp36_node_start.js

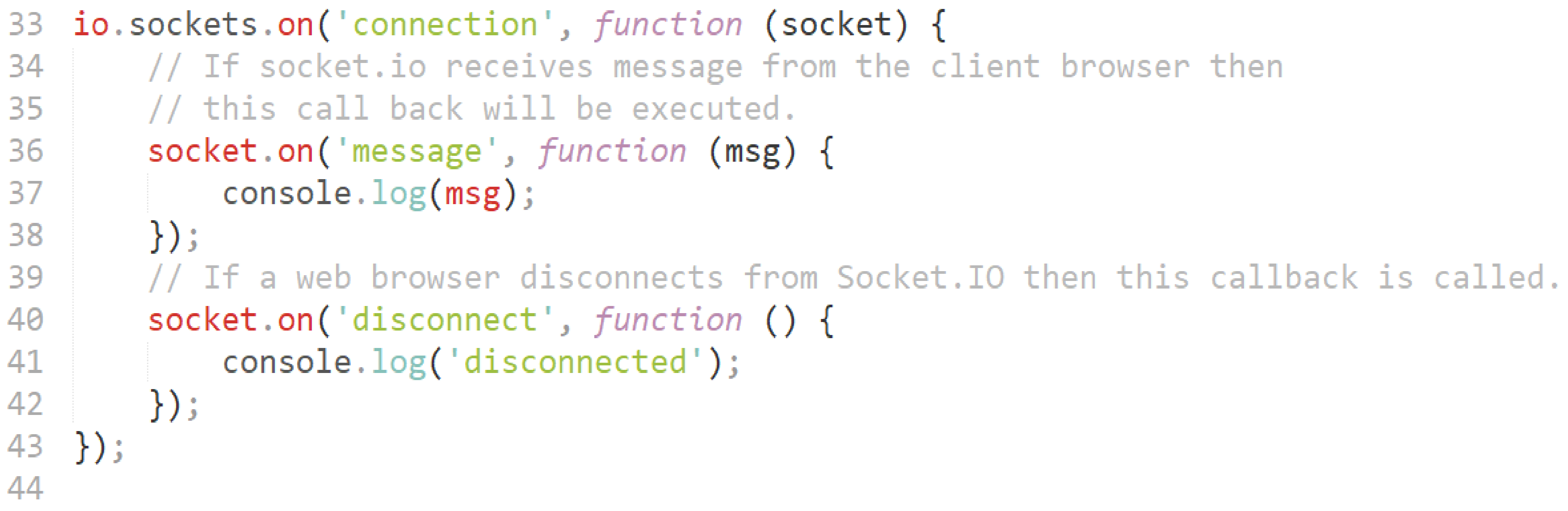


```
// tmp36_node.js

var serialport = require('serialport');
var portName = 'COM10';  // check your COM port!!
var port     =   process.env.PORT || 3000;

var io = require('socket.io').listen(port);

// serial port object
var sp = new serialport(portName,{
    baudRate: 9600,    // 9600  38400
    dataBits: 8,
    parity: 'none',
    stopBits: 1,
    flowControl: false,
    parser: serialport.parsers.readline('\r\n')  // new serialport.pars
});

var tdata = [];  // Array

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

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

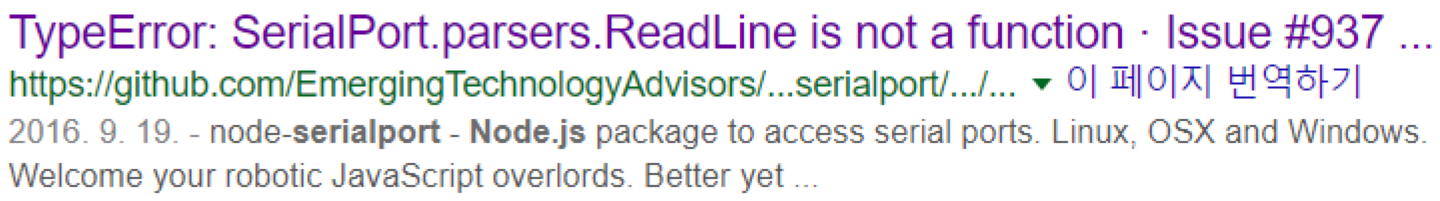
A4.1.7 tmp36 node project : code-2

tmp36_node.js



```javascript
io.sockets.on('connection', function (socket) {
    // If socket.io receives message from the client browser then
    // this call back will be executed.
    socket.on('message', function (msg) {
        console.log(msg);
    });
    // If a web browser disconnects from Socket.IO then this callback is called.
    socket.on('disconnect', function () {
        console.log('disconnected');
    });
});

```

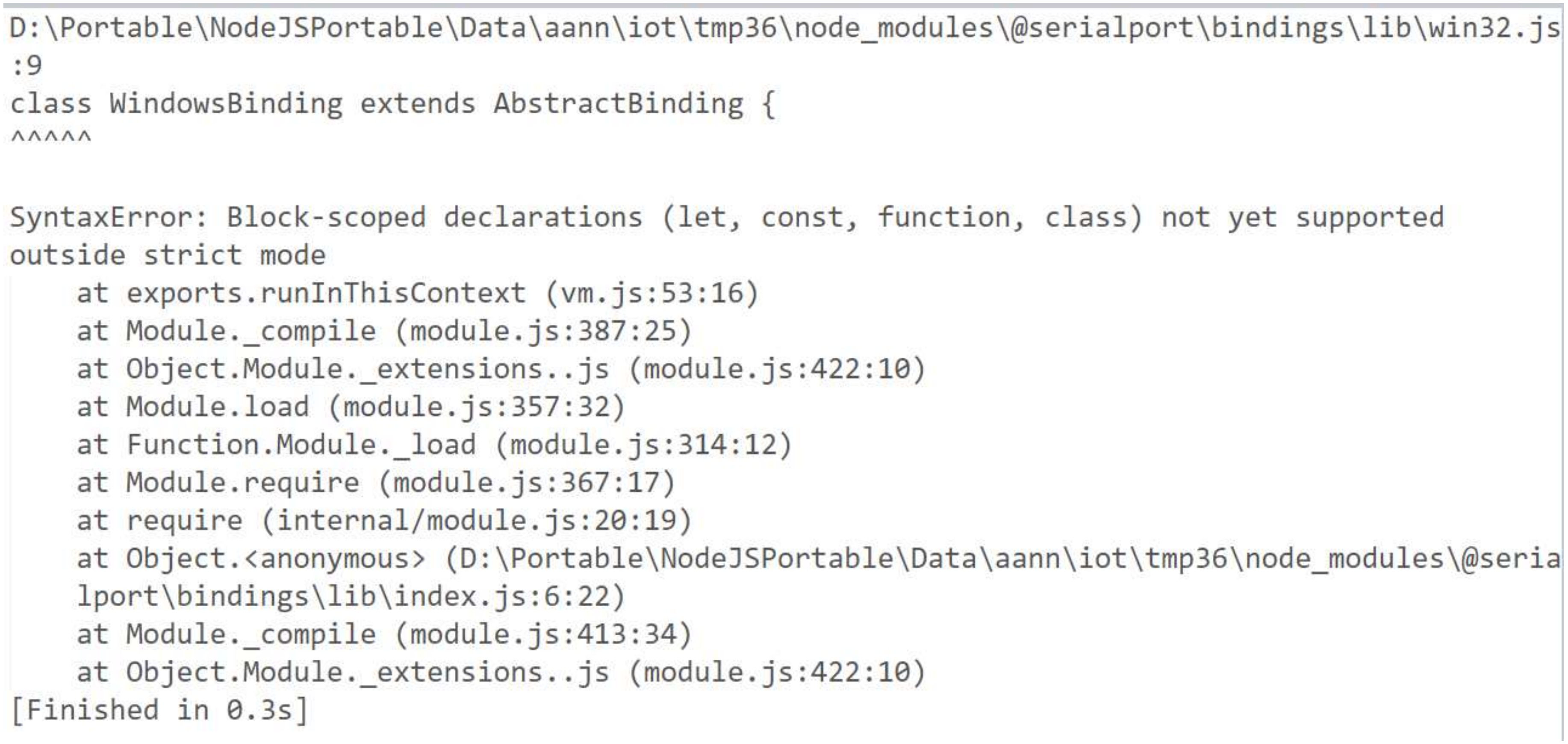
serialport 6.x 버전의 API 변화로 오류 발생, 버전 downgrade



TypeError: SerialPort.parsers.ReadLine is not a function · Issue #937 ...
https://github.com/EmergingTechnologyAdvisors/...serialport/.../... ▾ 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 ...

Error & Bug ---

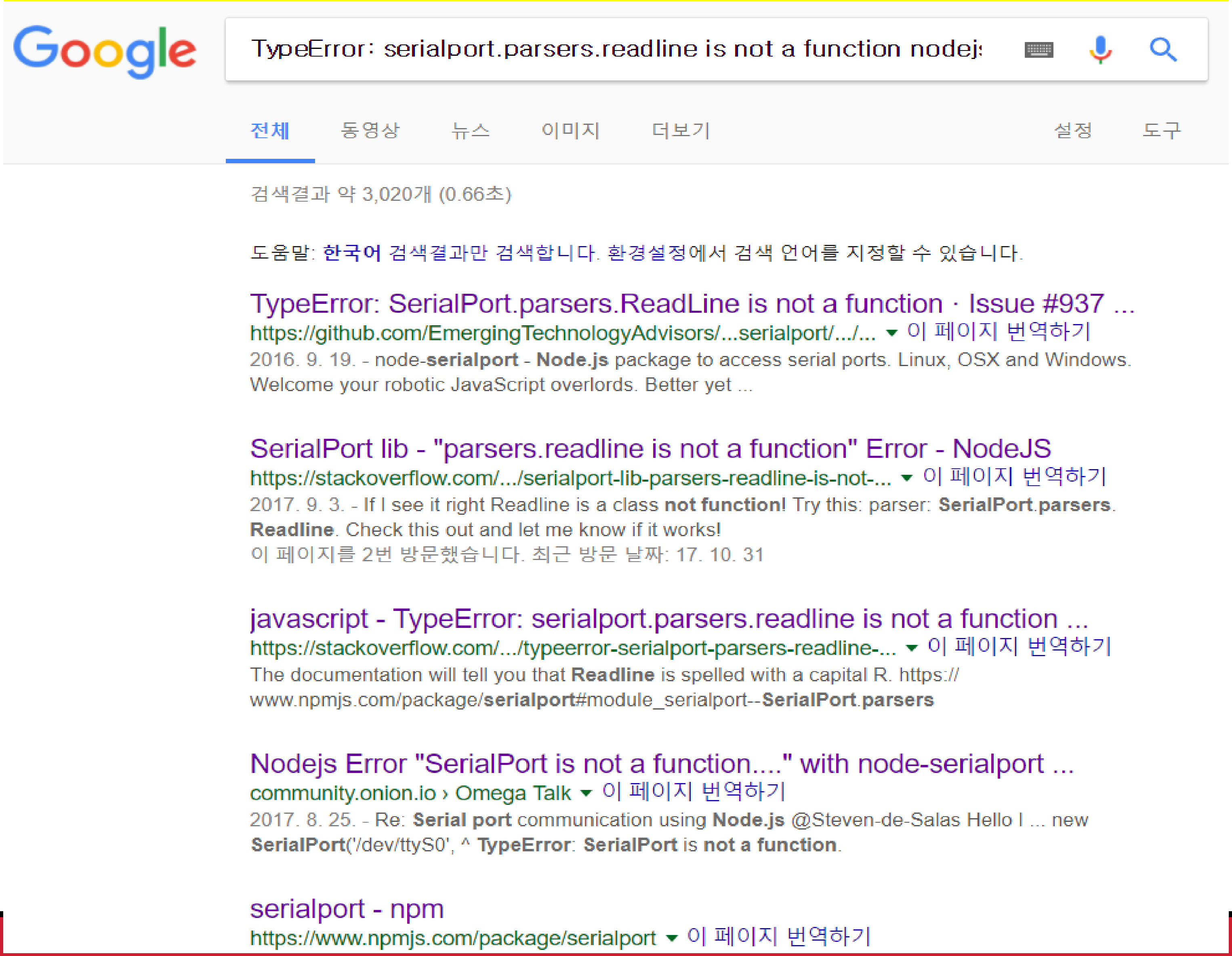
serialport 6.x 버전의 API 변화로 오류 발생, 버전 downgrade



```
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



TypeError: serialport.parsers.readline is not a function nodej:

검색결과 약 3,020개 (0.66초)

도움말: 한국어 검색결과만 검색합니다. 환경설정에서 검색 언어를 지정할 수 있습니다.

TypeError: SerialPort.parsers.ReadLine is not a function · Issue #937 ...
https://github.com/EmergingTechnologyAdvisors/...serialport/.../... ▾ 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 ...

SerialPort lib - "parsers.readline is not a function" Error - NodeJS
https://stackoverflow.com/.../serialport-lib-parsers-readline-is-not-... ▾ 이 페이지 번역하기
2017. 9. 3. - If I see it right Readline is a class **not function!** Try this: parser: **SerialPort.parsers**. **Readline**. Check this out and let me know if it works!
이 페이지를 2번 방문했습니다. 최근 방문 날짜: 17. 10. 31

javascript - TypeError: serialport.parsers.readline is not a function ...
https://stackoverflow.com/.../typeerror-serialport-parsers-readline-... ▾ 이 페이지 번역하기
The documentation will tell you that **Readline** is spelled with a capital R. https://www.npmjs.com/package/**serialport**#module_serialport--**SerialPort.parsers**

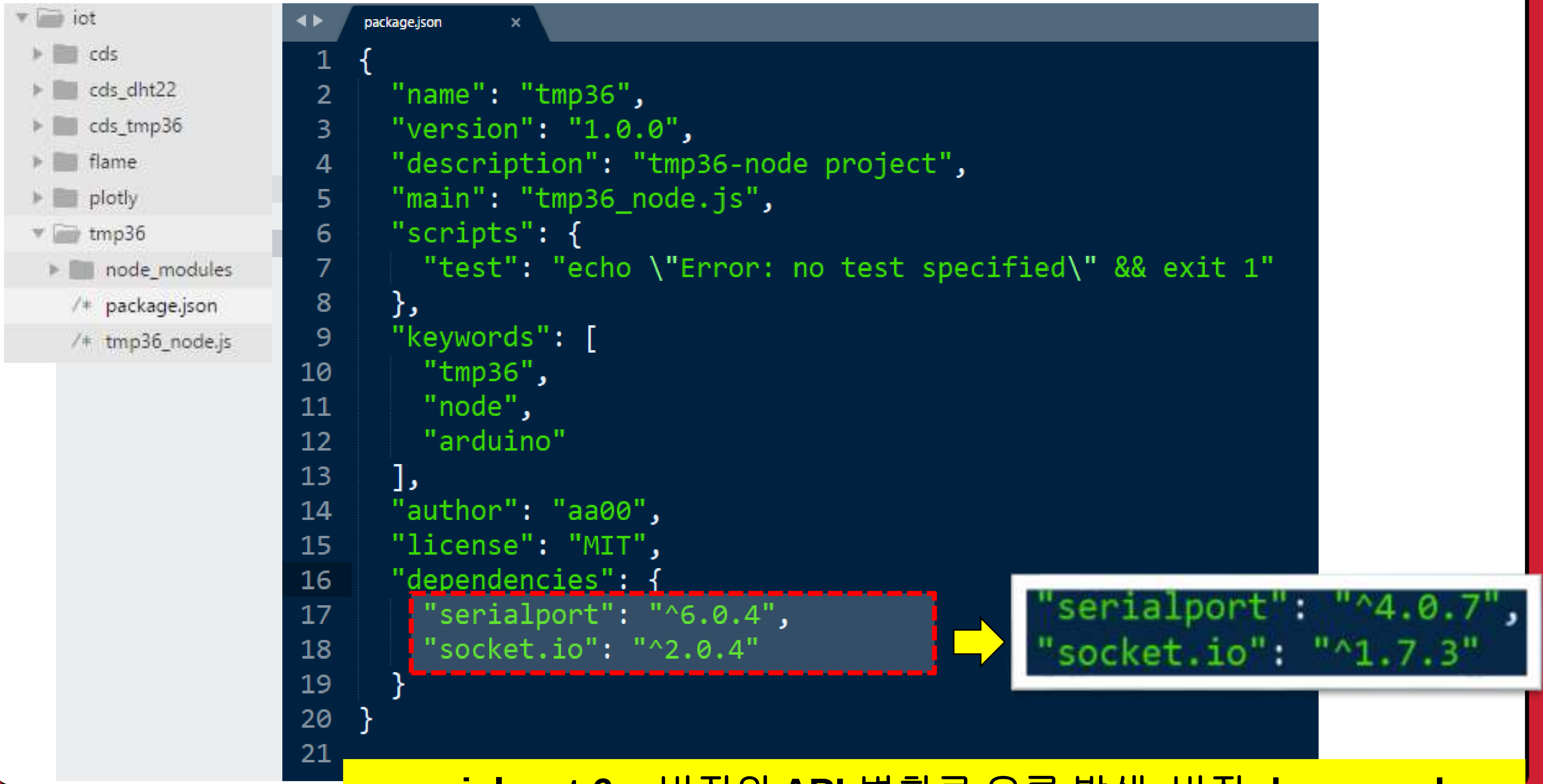
Nodejs Error "SerialPort is not a function...." with node-serialport ...
community.onion.io › Omega Talk ▾ 이 페이지 번역하기
2017. 8. 25. - Re: **Serial port** communication using **Node.js** @Steven-de-Salas Hello I ... new **SerialPort**('/dev/ttyS0', ^ **TypeError**: **SerialPort** is **not a function**.

serialport - npm
https://www.npmjs.com/package/serialport ▾ 이 페이지 번역하기

A4.1.6A tmp36 node project → downgrade

Go to tmp36 subfolder (after deleteing node_modules subfolder)

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



iot / cds / cds_dht22 / cds_tmp36 / flame / plotly / tmp36 / node_modules / package.json / tmp36_node.js

package.json

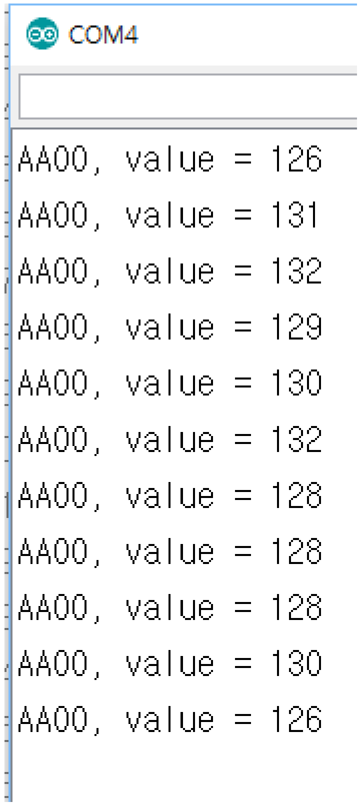
```
{
  "name": "tmp36",
  "version": "1.0.0",
  "description": "tmp36-node project",
  "main": "tmp36_node.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "tmp36",
    "node",
    "arduino"
  ],
  "author": "aa00",
  "license": "MIT",
  "dependencies": {
    "serialport": "^6.0.4",
    "socket.io": "^2.0.4"
  }
}
```

```
"serialport": "^4.0.7",
"socket.io": "^1.7.3"
```

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

A4.1.8 tmp36 node project (after downgrade)

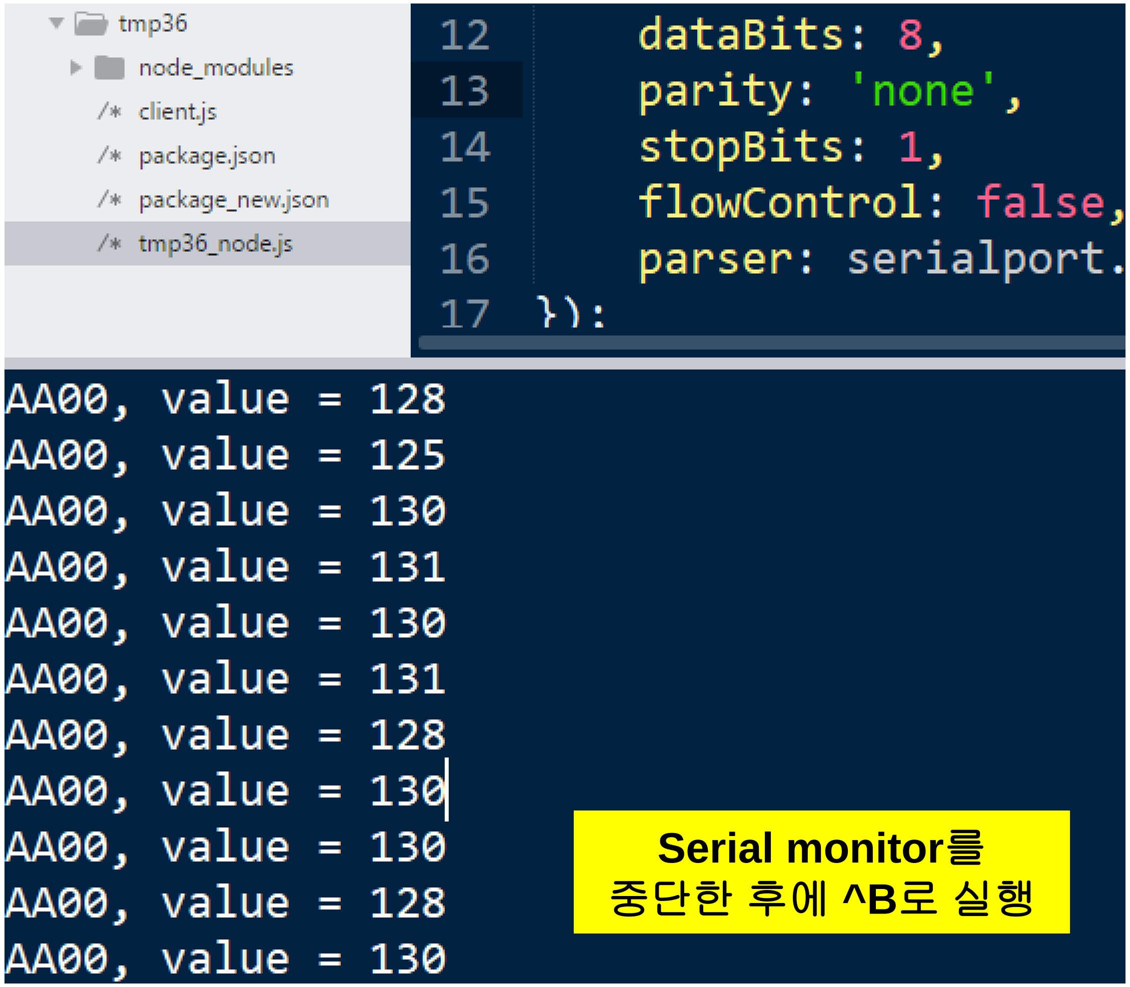
Serial output (A0 in Arduino)



COM4

```
AA00, value = 126
AA00, value = 131
AA00, value = 132
AA00, value = 129
AA00, value = 130
AA00, value = 132
AA00, value = 128
AA00, value = 128
AA00, value = 128
AA00, value = 130
AA00, value = 126
```

tmp36_node.js (^B로 실행)



```
    dataBits: 8,
    parity: 'none',
    stopBits: 1,
    flowControl: false,
    parser: serialport.
});
```

```
AA00, value = 128
AA00, value = 125
AA00, value = 130
AA00, value = 131
AA00, value = 130
AA00, value = 131
AA00, value = 128
AA00, value = 130
AA00, value = 130
AA00, value = 128
AA00, value = 130
```

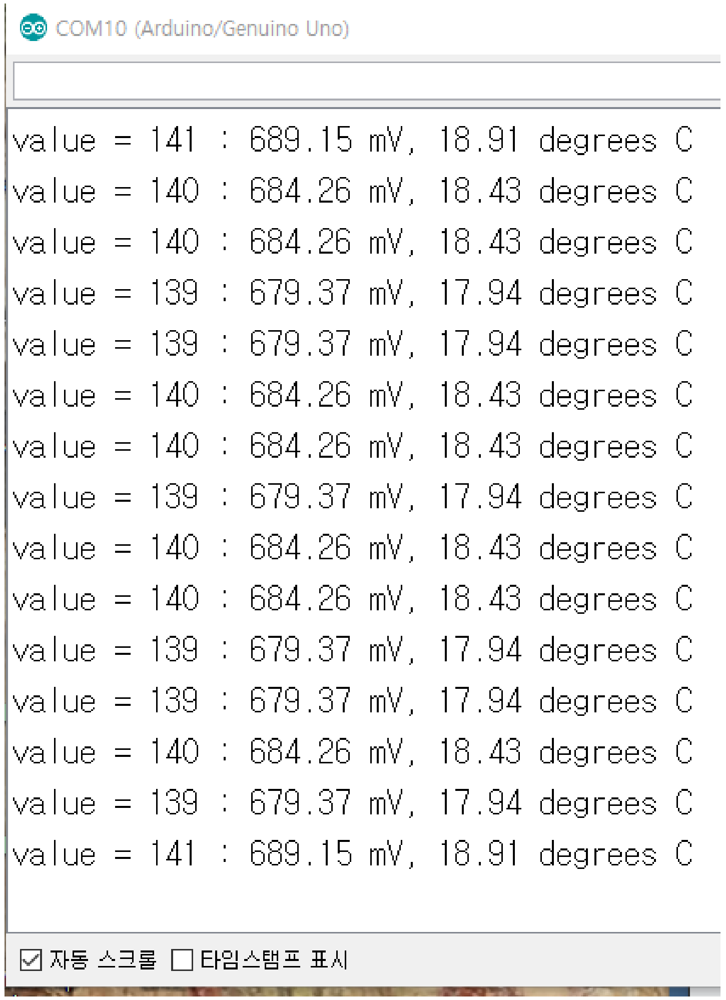
Serial monitor를 중단한 후에 ^B로 실행

A4.1.9 tmp36 node project (all messages)

AAnn_TMP36_NodeJS.ino

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

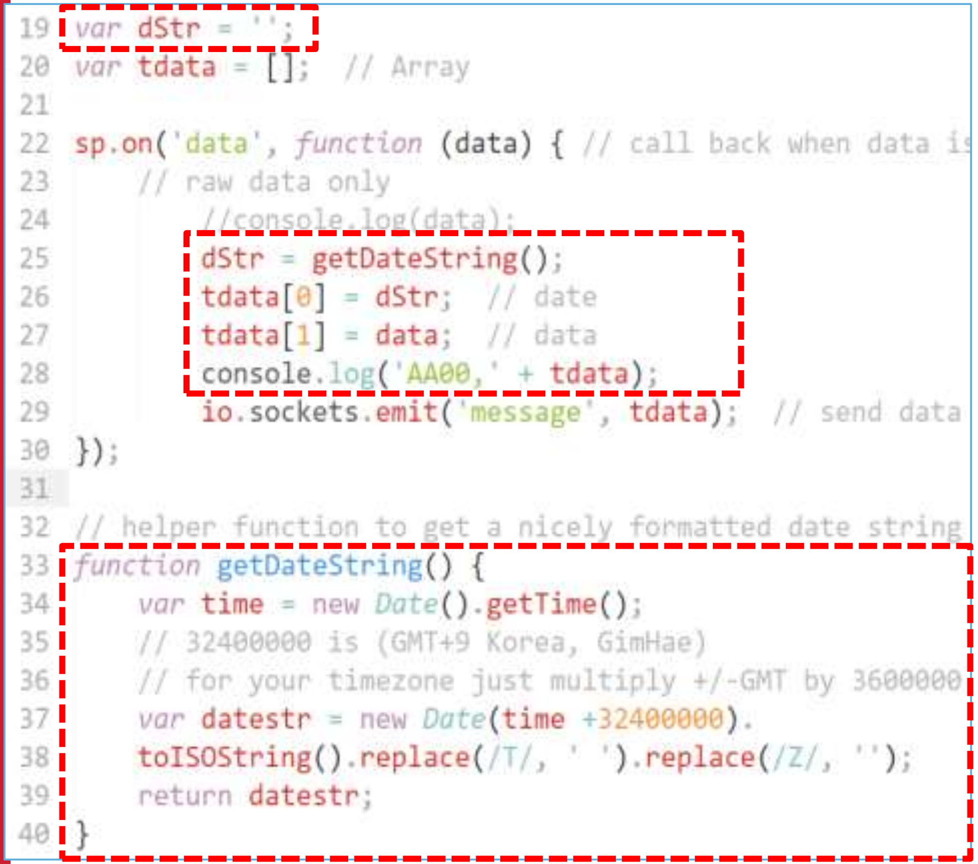

COM10 (Arduino/Genuino Uno)

```
value = 141 : 689.15 mV, 18.91 degrees C
value = 140 : 684.26 mV, 18.43 degrees C
value = 140 : 684.26 mV, 18.43 degrees C
value = 139 : 679.37 mV, 17.94 degrees C
value = 139 : 679.37 mV, 17.94 degrees C
value = 140 : 684.26 mV, 18.43 degrees C
value = 140 : 684.26 mV, 18.43 degrees C
value = 139 : 679.37 mV, 17.94 degrees C
value = 140 : 684.26 mV, 18.43 degrees C
value = 140 : 684.26 mV, 18.43 degrees C
value = 139 : 679.37 mV, 17.94 degrees C
value = 139 : 679.37 mV, 17.94 degrees C
value = 140 : 684.26 mV, 18.43 degrees C
value = 139 : 679.37 mV, 17.94 degrees C
value = 141 : 689.15 mV, 18.91 degrees C
```

[✓ 자동 스크롤] ☐ 타임스탬프 표시

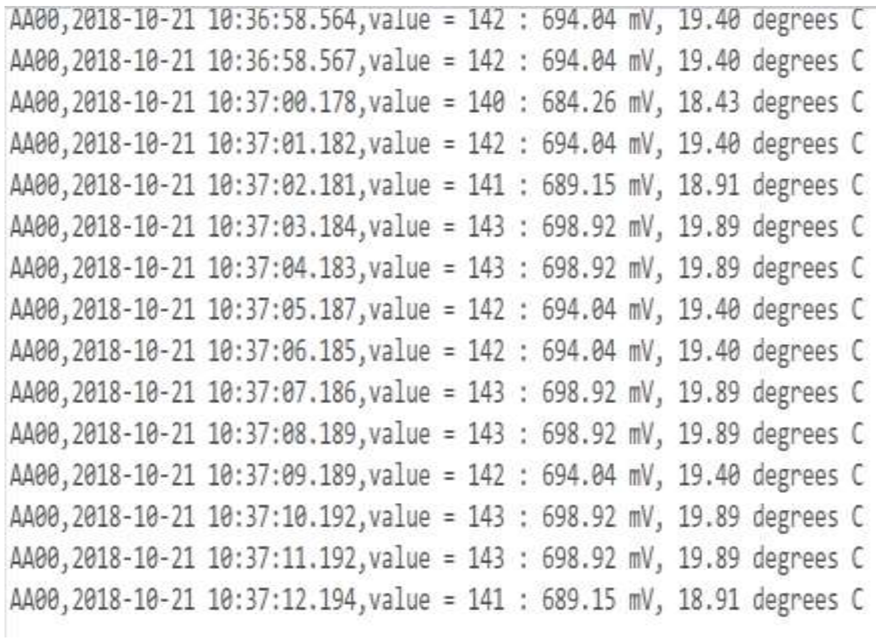
A4.1.9 tmp36 node project (all messages)

tmp36_node.js



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

sp.on('data', function (data) { // call back when data is
    // 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
});

// 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;
}
```

SB3에서 tmp36_node.js를 ^B로 실행

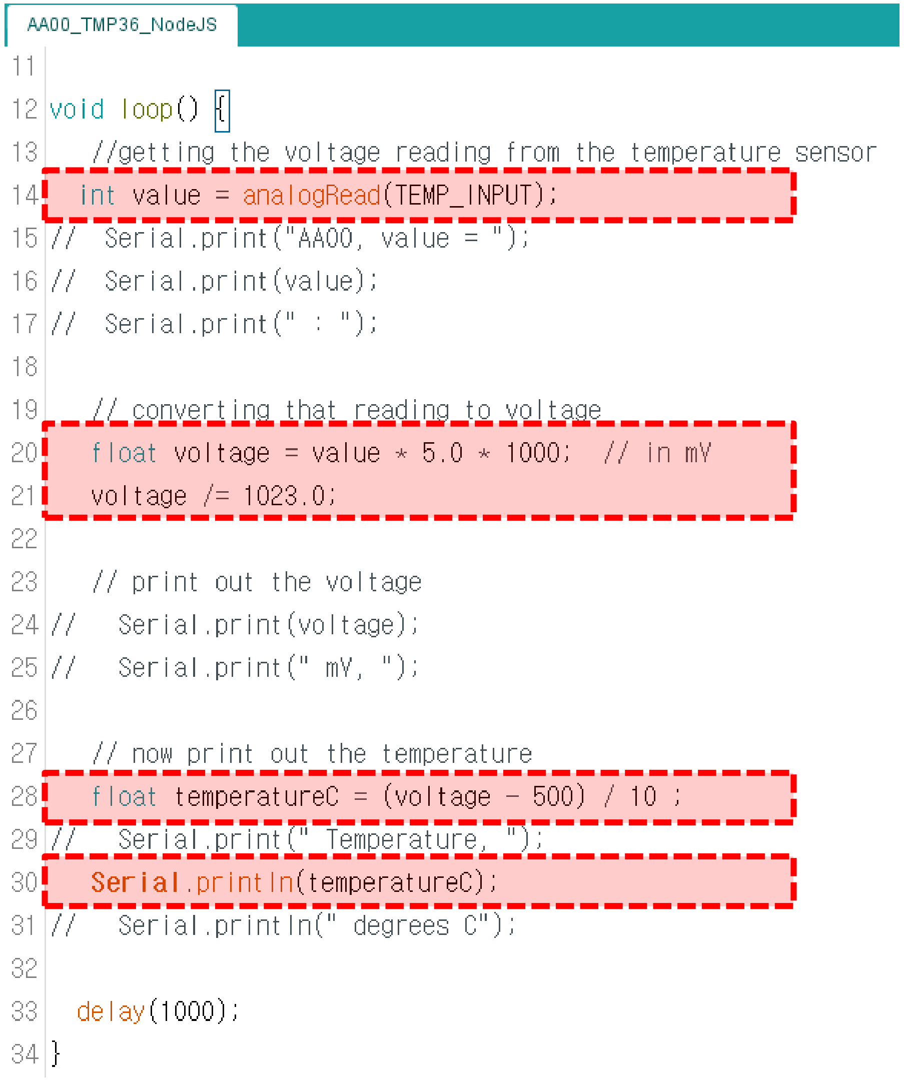


```
AA00,2018-10-21 10:36:58.564,value = 142 : 694.04 mV, 19.40 degrees C
AA00,2018-10-21 10:36:58.567,value = 142 : 694.04 mV, 19.40 degrees C
AA00,2018-10-21 10:37:00.178,value = 140 : 684.26 mV, 18.43 degrees C
AA00,2018-10-21 10:37:01.182,value = 142 : 694.04 mV, 19.40 degrees C
AA00,2018-10-21 10:37:02.181,value = 141 : 689.15 mV, 18.91 degrees C
AA00,2018-10-21 10:37:03.184,value = 143 : 698.92 mV, 19.89 degrees C
AA00,2018-10-21 10:37:04.183,value = 143 : 698.92 mV, 19.89 degrees C
AA00,2018-10-21 10:37:05.187,value = 142 : 694.04 mV, 19.40 degrees C
AA00,2018-10-21 10:37:06.185,value = 142 : 694.04 mV, 19.40 degrees C
AA00,2018-10-21 10:37:07.186,value = 143 : 698.92 mV, 19.89 degrees C
AA00,2018-10-21 10:37:08.189,value = 143 : 698.92 mV, 19.89 degrees C
AA00,2018-10-21 10:37:09.189,value = 142 : 694.04 mV, 19.40 degrees C
AA00,2018-10-21 10:37:10.192,value = 143 : 698.92 mV, 19.89 degrees C
AA00,2018-10-21 10:37:11.192,value = 143 : 698.92 mV, 19.89 degrees C
AA00,2018-10-21 10:37:12.194,value = 141 : 689.15 mV, 18.91 degrees C
```

AAnn_tmp36_message.png로 저장

A4.1.10 tmp36 node project (only data)

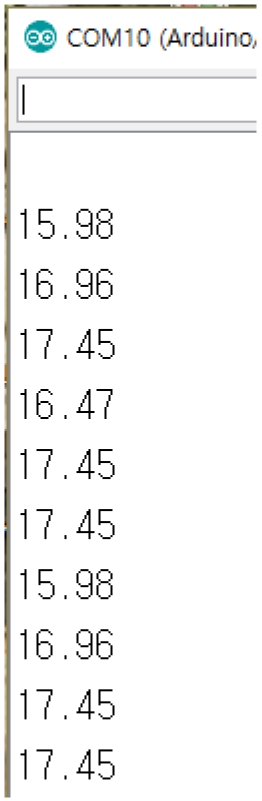
AAnn_TMP36_NodeJS.ino 수정



AA00_TMP36_NodeJS

```
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(" Temperature, ");
   Serial.println(temperatureC);
//   Serial.println(" degrees C");

  delay(1000);
}
```

실행 결과

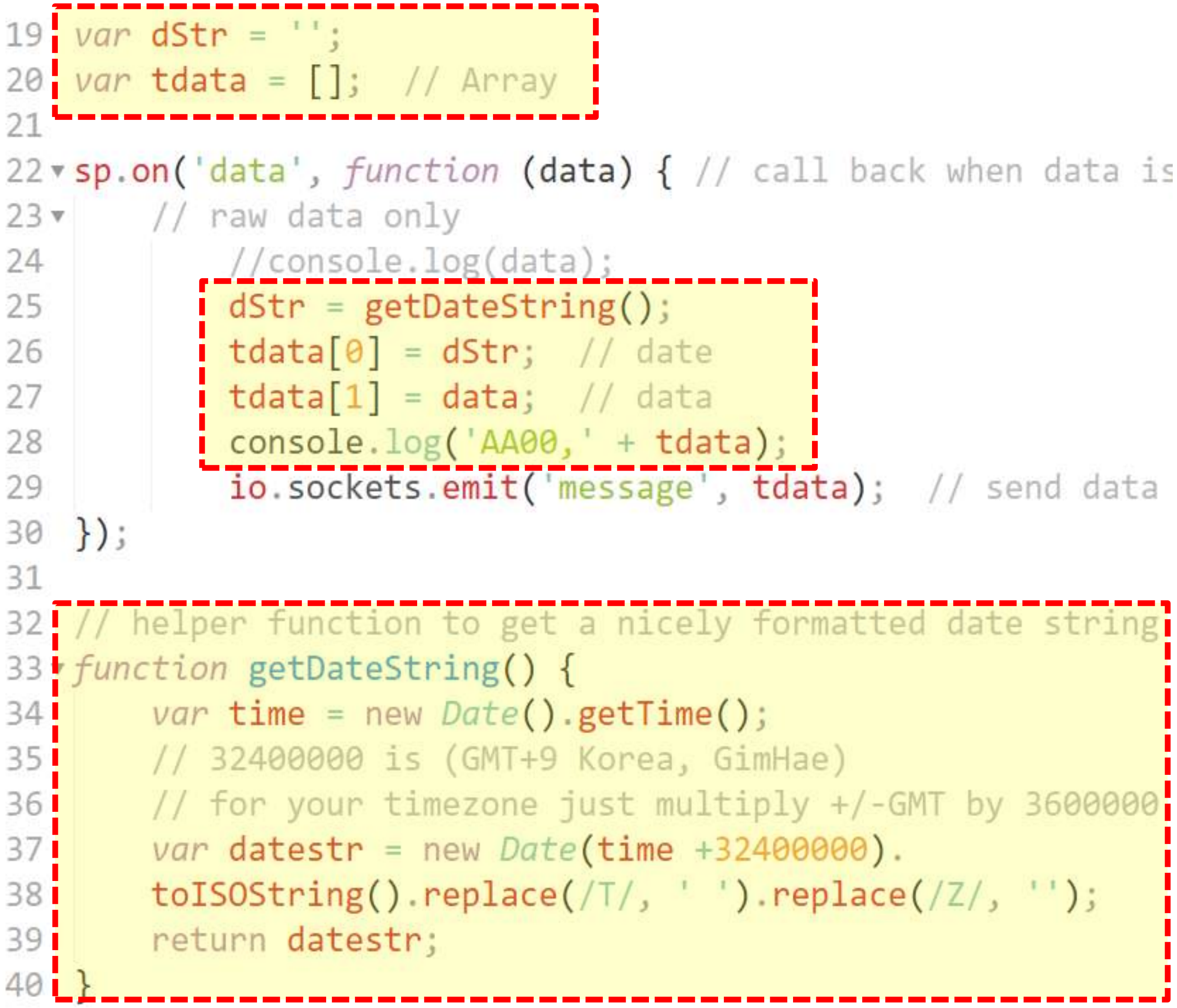


COM10 (Arduino,

```
15.98
16.96
17.45
16.47
17.45
17.45
15.98
16.96
17.45
17.45
```

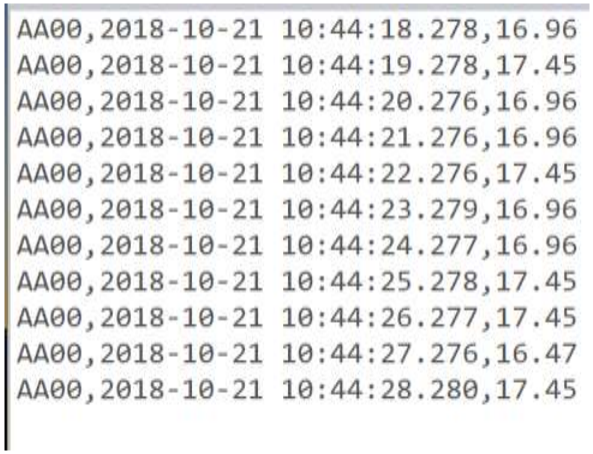
A4.1.11 tmp36 node project (date & data → IOT)

tmp36_node.js



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

sp.on('data', function (data) { // call back when data is
    // 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
});

// 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;
}
```

IOT data format
시간, data
시간, 온도

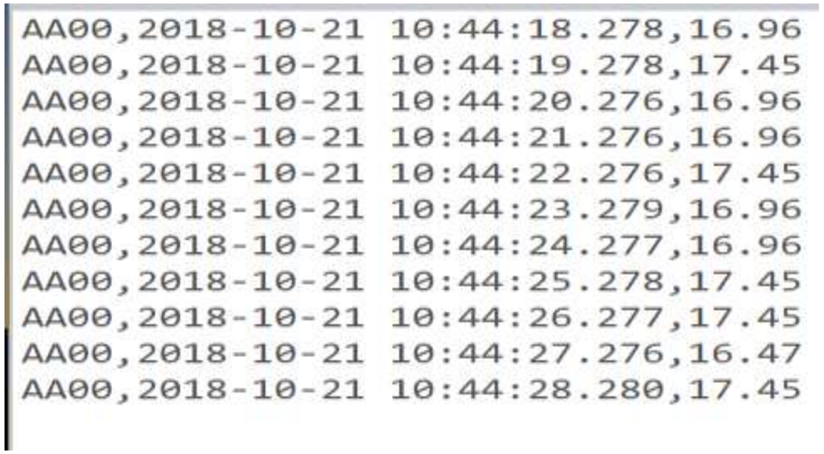


```
AA00,2018-10-21 10:44:18.278,16.96
AA00,2018-10-21 10:44:19.278,17.45
AA00,2018-10-21 10:44:20.276,16.96
AA00,2018-10-21 10:44:21.276,16.96
AA00,2018-10-21 10:44:22.276,17.45
AA00,2018-10-21 10:44:23.279,16.96
AA00,2018-10-21 10:44:24.277,16.96
AA00,2018-10-21 10:44:25.278,17.45
AA00,2018-10-21 10:44:26.277,17.45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

시간 , 온도

A4.1.12 tmp36 node project (실행 결과)

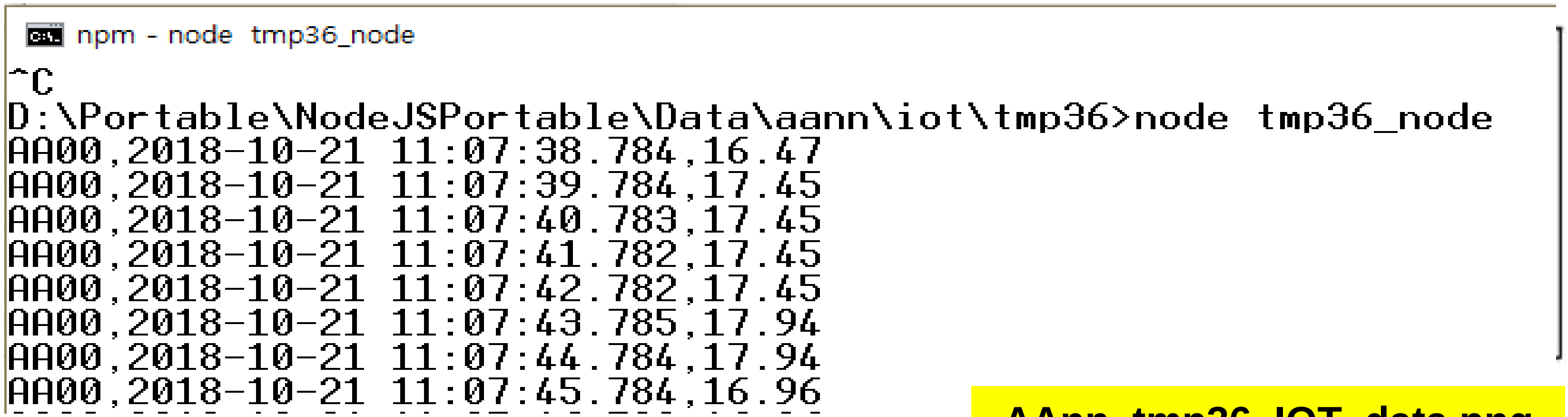
- Sublime Text 3에서 실행



```
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

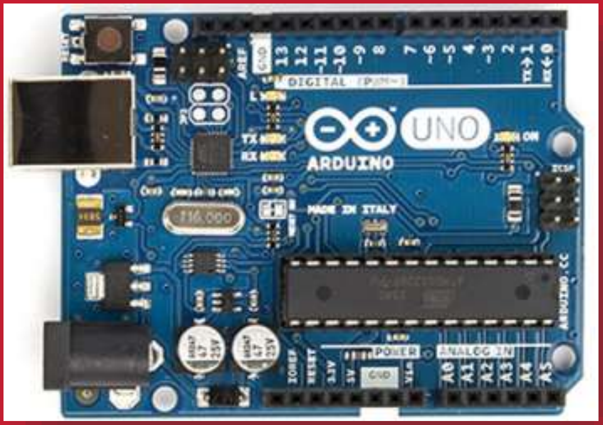
```
^C
D:\Portable\NodeJSPortable\Data\aann\iot\tmp36>node tmp36_node
AA00,2018-10-21 11:07:38.784,16.47
AA00,2018-10-21 11:07:39.784,17.45
AA00,2018-10-21 11:07:40.783,17.45
AA00,2018-10-21 11:07:41.782,17.45
AA00,2018-10-21 11:07:42.782,17.45
AA00,2018-10-21 11:07:43.785,17.94
AA00,2018-10-21 11:07:44.784,17.94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

AAnn_tmp36_IOT_data.png
로 저장

Single sensor: CdS

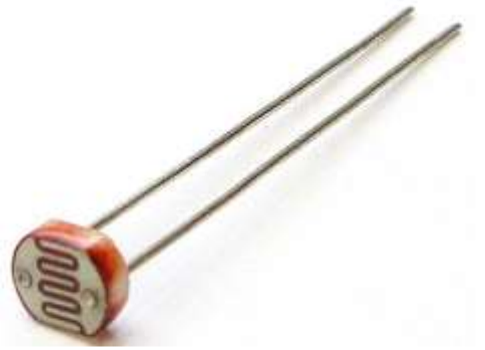
Arduino

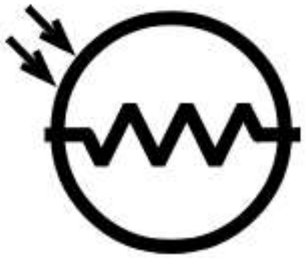
+ Node.js



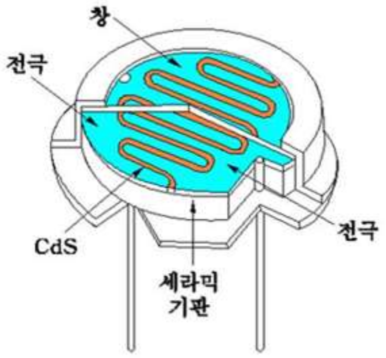
A3.2 Luminosity sensor [Photocell LDR]

CdS 센서- photoresistor





CDS특성

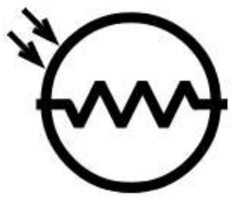


1. 감도
   - 빛의 파장에 따라 감도가 다름
2. 허용손실
   - 비교적 큰 전류를 흘릴 수 있음
3. 암 전류
   - 빛이 없어도 약간의 전류가 흐름
4. 명 전류
   - 빛을 비추면 흐르는 전류
5. 응답특성
   - 응답 시간 지연
   - 빛의 세기에 따라 응답시간 다름
6. 가변저항
   - 빛에 따른 가변저항

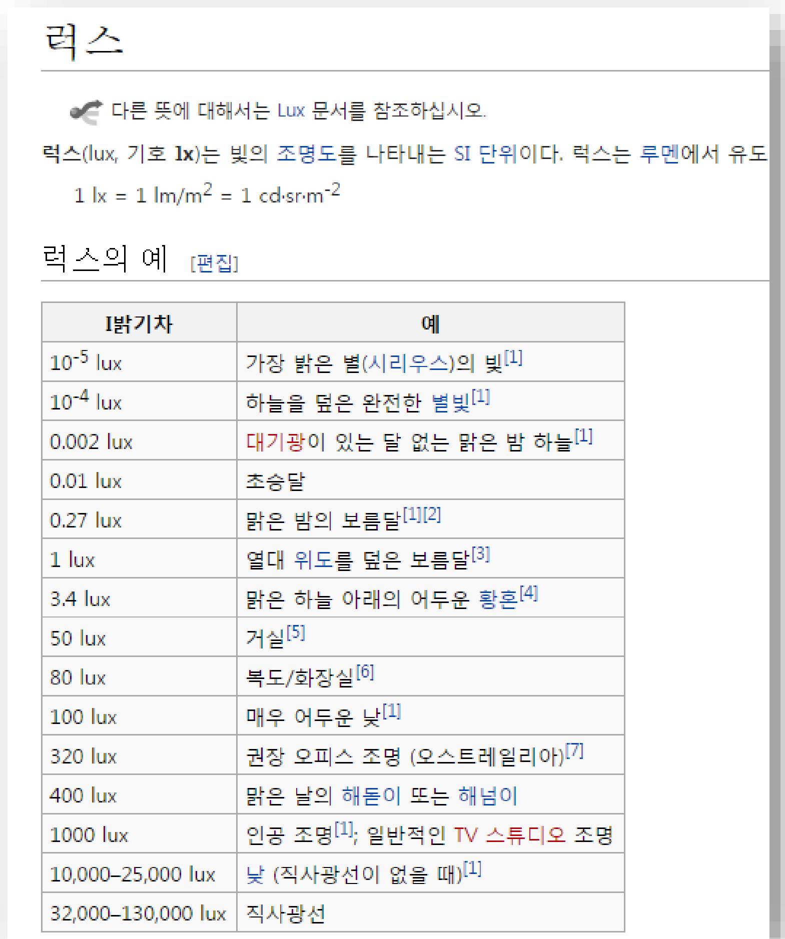
https://en.wikipedia.org/wiki/Photoresistor

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 lx = 1 lm/m^2 = 1 cd·sr·m^{-2}

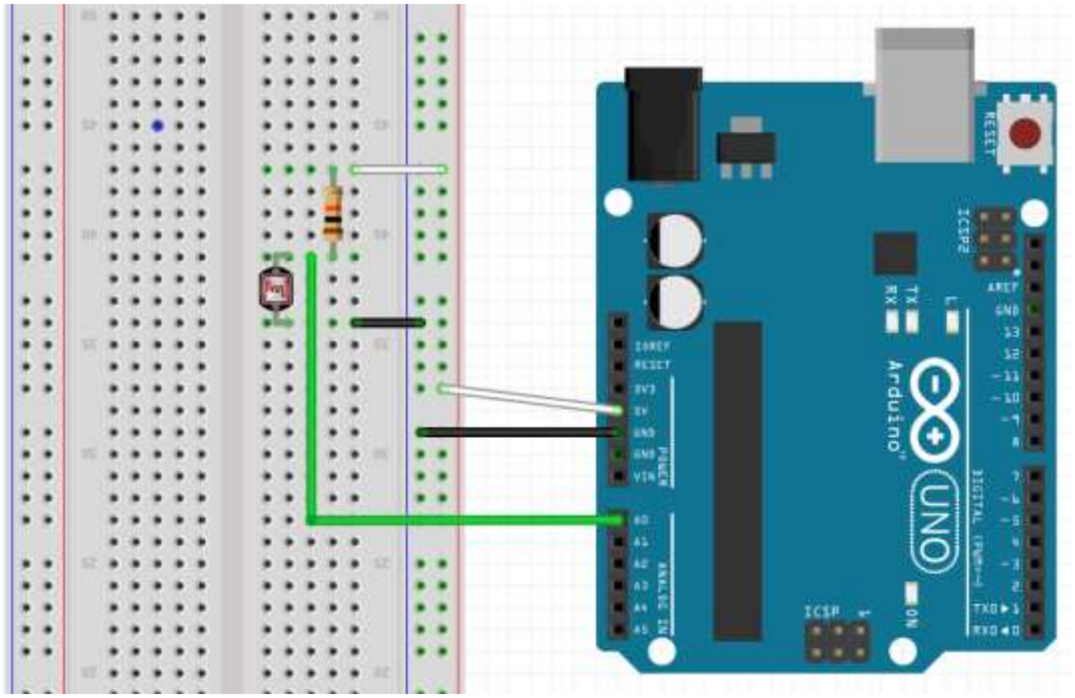
럭스의 예 [편집]

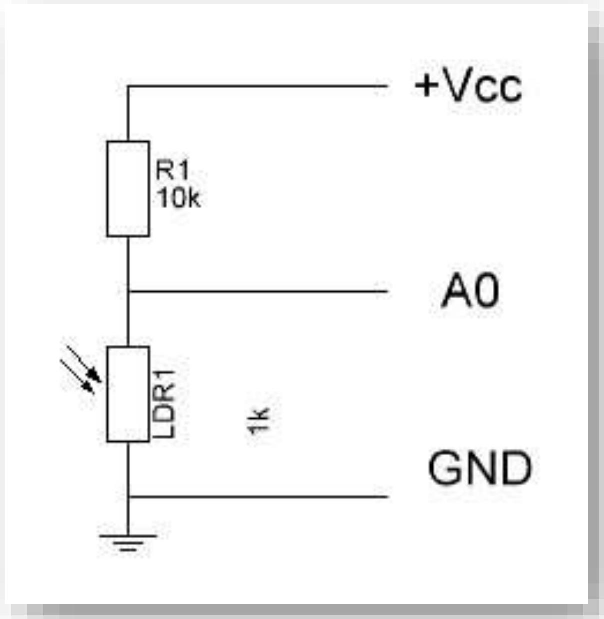
l밝기차	예
10^{-5} lux	가장 밝은 별(시리우스)의 빛[1]
10^{-4} lux	하늘을 덮은 완전한 별빛[1]
0.002 lux	대기광이 있는 달 없는 맑은 밤 하늘[1]
0.01 lux	초승달
0.27 lux	맑은 밤의 보름달[1][2]
1 lux	열대 위도를 덮은 보름달[3]
3.4 lux	맑은 하늘 아래의 어두운 황혼[4]
50 lux	거실[5]
80 lux	복도/화장실[6]
100 lux	매우 어두운 낮[1]
320 lux	권장 오피스 조명 (오스트레일리아)[7]
400 lux	맑은 날의 해돋이 또는 해넘이
1000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

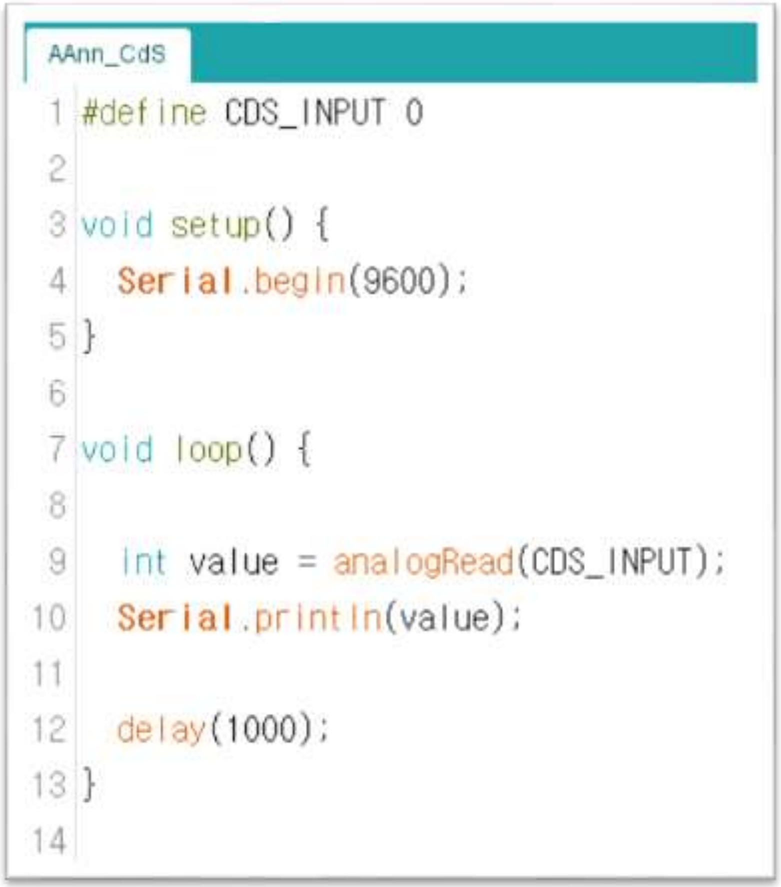
1. A0 핀을 CdS 조도 센서의 입력으로 설정한다.

2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.

3. loop()에서 **analogRead**() 함수로 A0 핀에서 측정되는 값을 읽어 들인다.

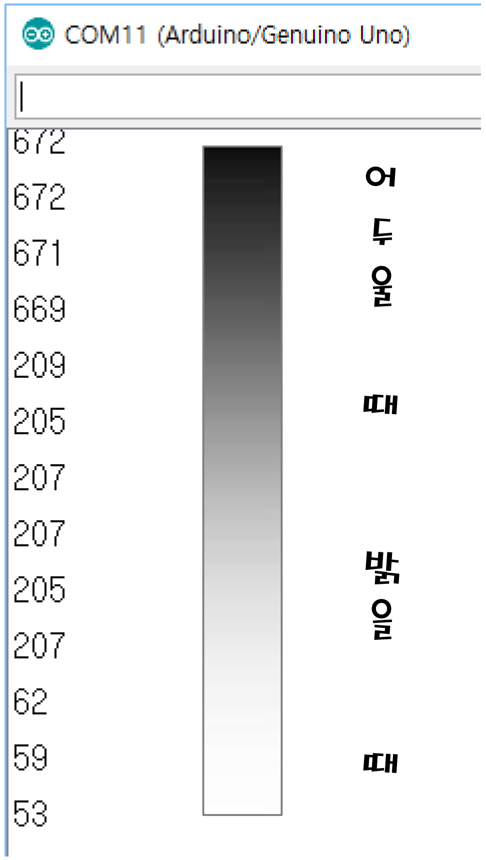
A3.2.4 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 1.



AAnn_CdS

```
#define CDS_INPUT 0

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

void loop() {

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

  delay(1000);
}

```

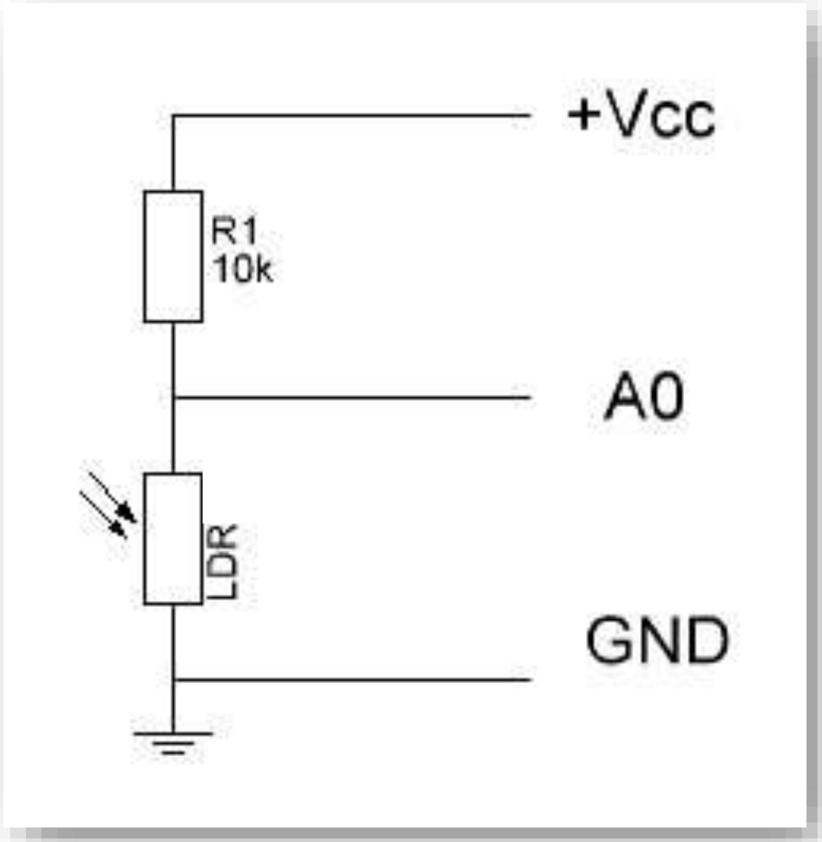


COM11 (Arduino/Genuino Uno)

```
672
672
671
669
209
205
207
207
205
207
62
59
53
```

어두우면 측정 값이 커지고 밝을수록 값이 작아진다 ???

CdS 센서 회로 분석 (1/2)



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

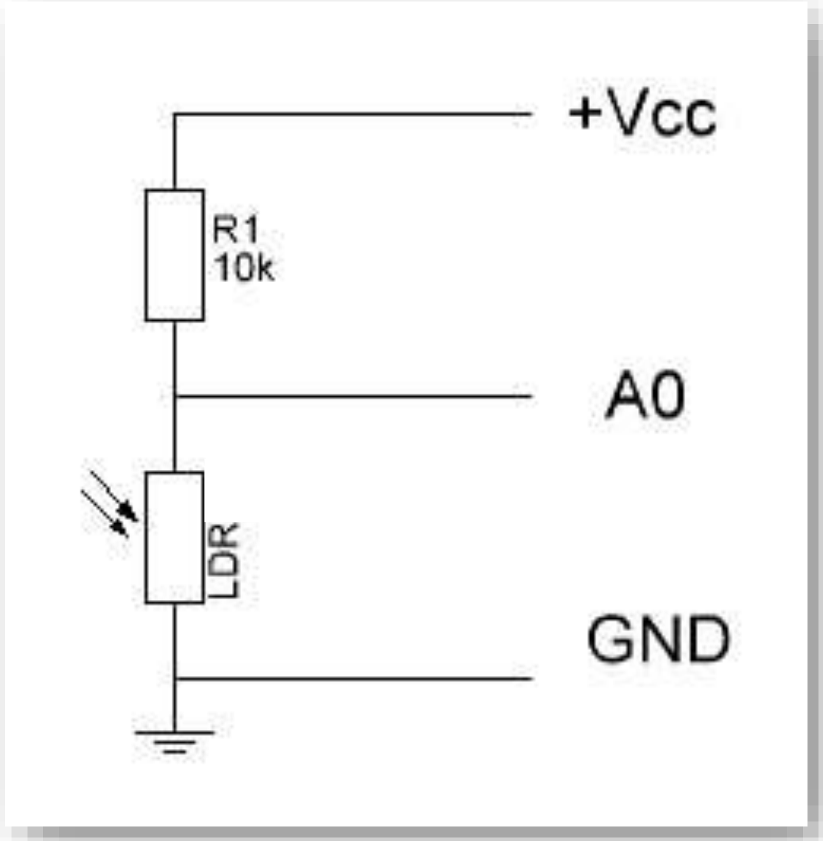
If you would us the LDR as the lower part of a voltage divider, then in darkness there would be a high voltage over the LDR, while in bright light, there would be a low voltage over that resistor.

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

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

A0에서 측정되는 LDR 양단의 전압 = V_{out}

CdS 센서 회로 분석 (2/2)



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

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

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

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

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

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

A0에서 측정되는 LDR 양단의 전압 = V_{out}

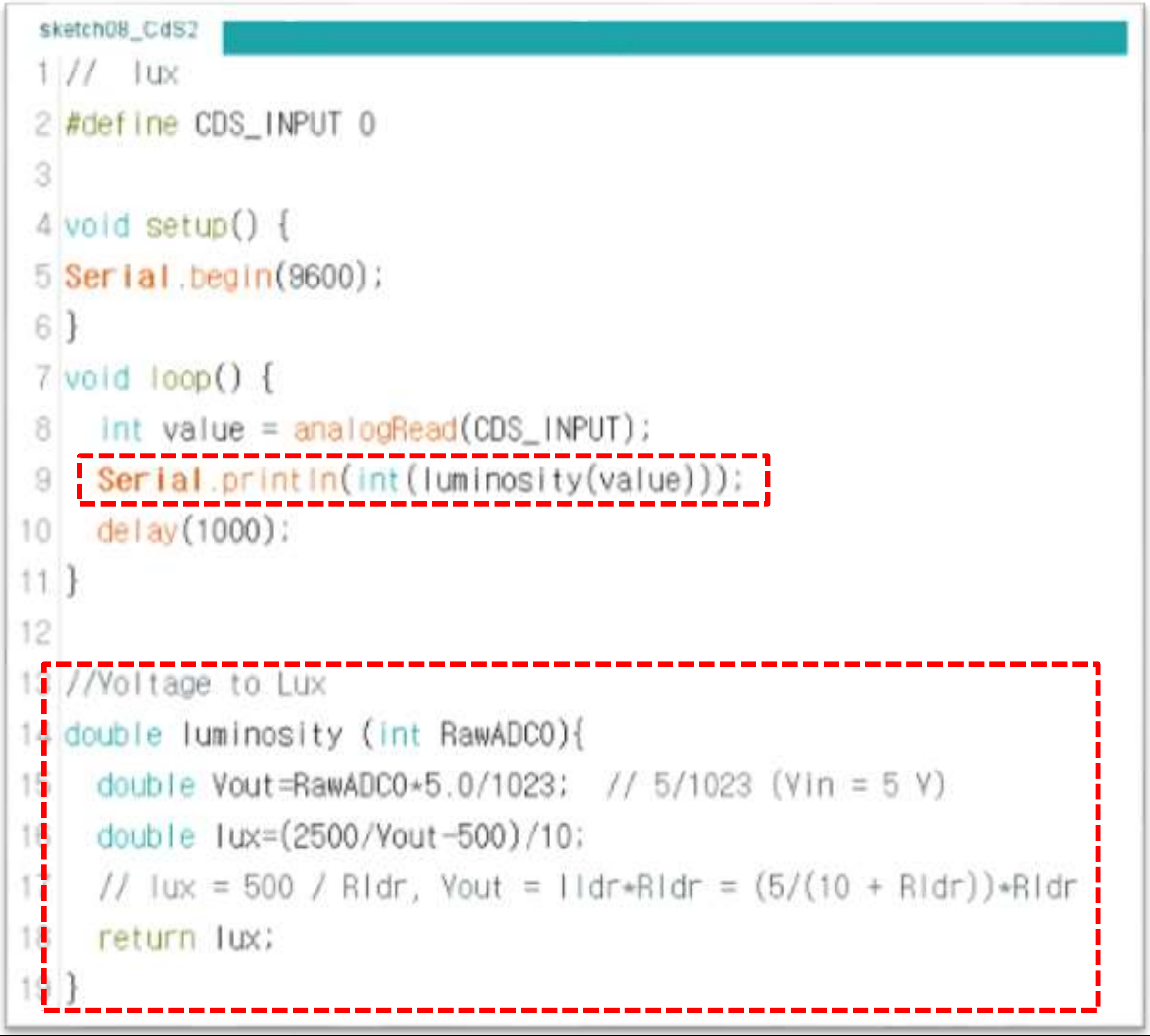
A3.2.5 Luminosity sensor [sketch-2]

▸ 스케치 구성

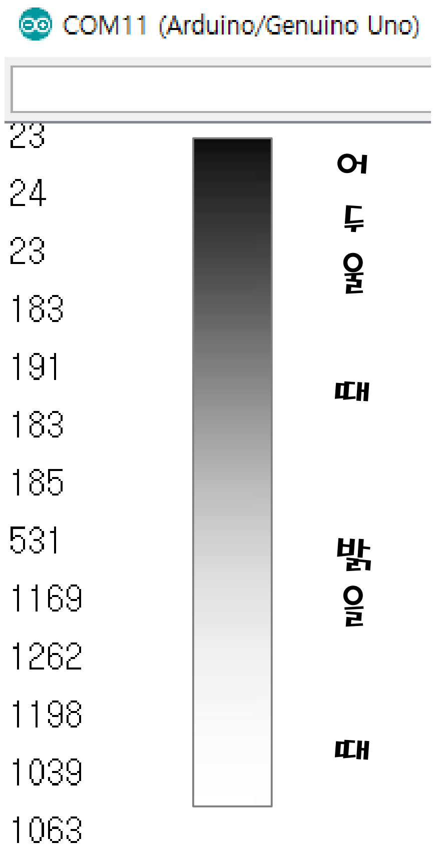
1. A0 핀을 CdS 조도 센서의 입력으로 설정한다.
2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
3. loop()에서 **analogRead**() 함수로 A0 핀에서 측정되는 값을 읽어 들인다.
4. A0 측정값 (0 ~ 1023)을 전압 (0 ~ 5 V)으로 환산한다.
5. 전압 (V)을 온도 (°C)로 환산한 후, A0 측정값, 환산 전압, 환산 조도를 한 줄로 1 초마다 컴퓨터로 전송한다.

A3.2.6 Luminosity sensor [Photocell LDR]

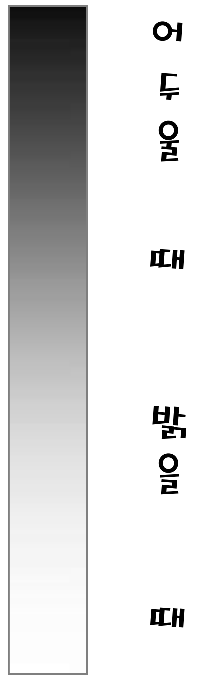
CdS 센서 회로 - 측정 2.



```
sketch08_CdS2
// lux
#define CDS_INPUT 0

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

//Voltage to Lux
double luminosity (int RawADC0){
  double Vout=RawADC0*5.0/1023;  // 5/1023 (Vin = 5 V)
  double lux=(2500/Vout-500)/10;
  // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
  return lux;
}
```



```
COM11 (Arduino/Genuino Uno)

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

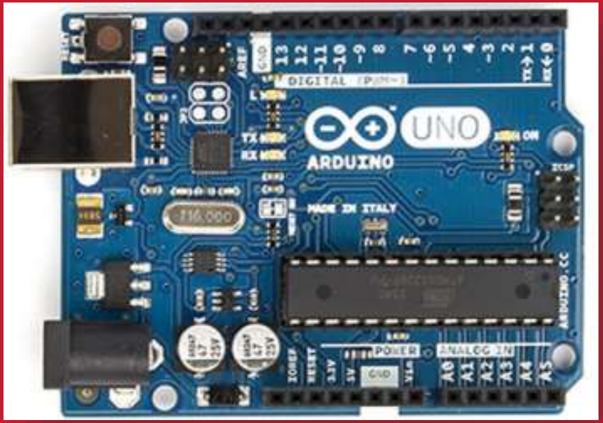


밝을수록 측정 값이 커지고
어두울수록 값이 작아진다 !!!

Single sensor: CdS

CdS (LDR)

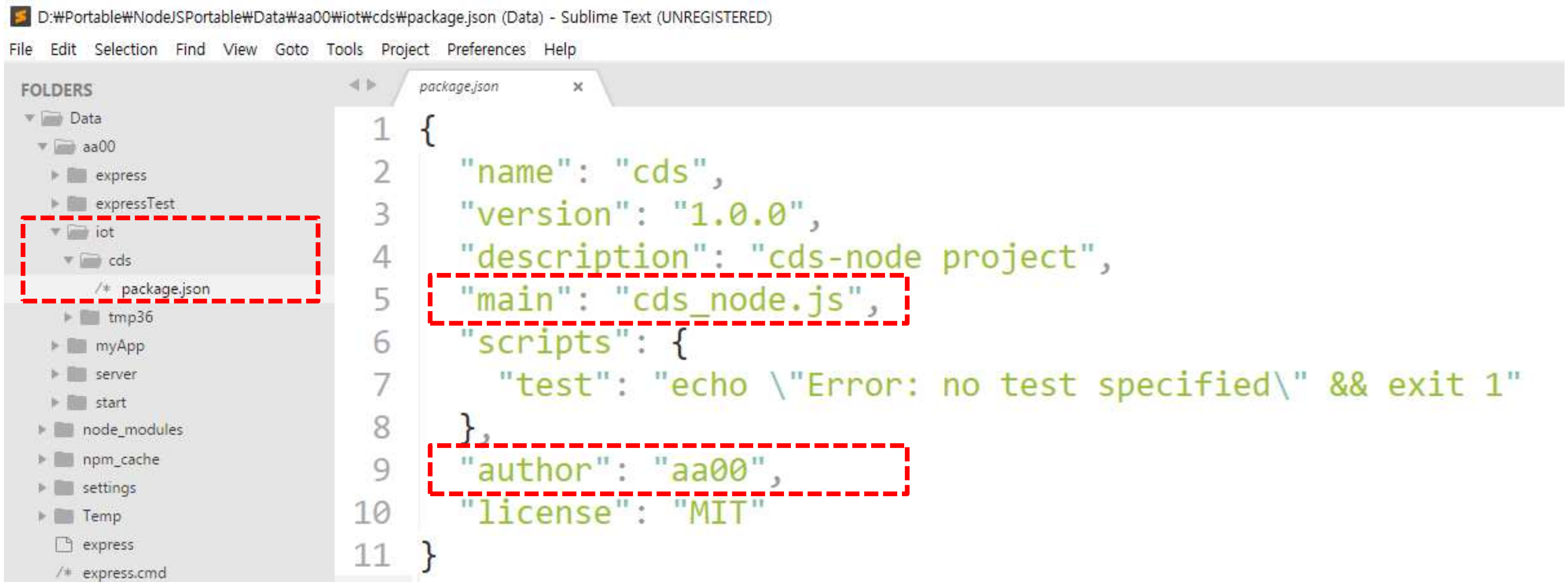
Node project



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"

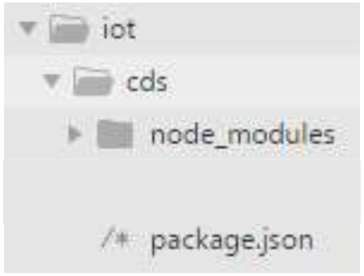


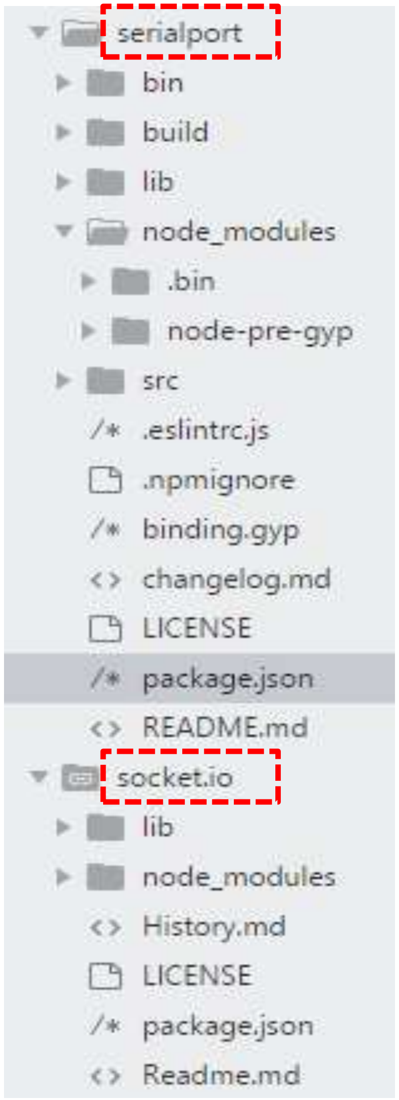
D:₩Portable₩NodeJSPortable₩Data₩aa00₩iot₩cds₩package.json (Data) - Sublime Text (UNREGISTERED)

```
{
  "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"
}
```

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]

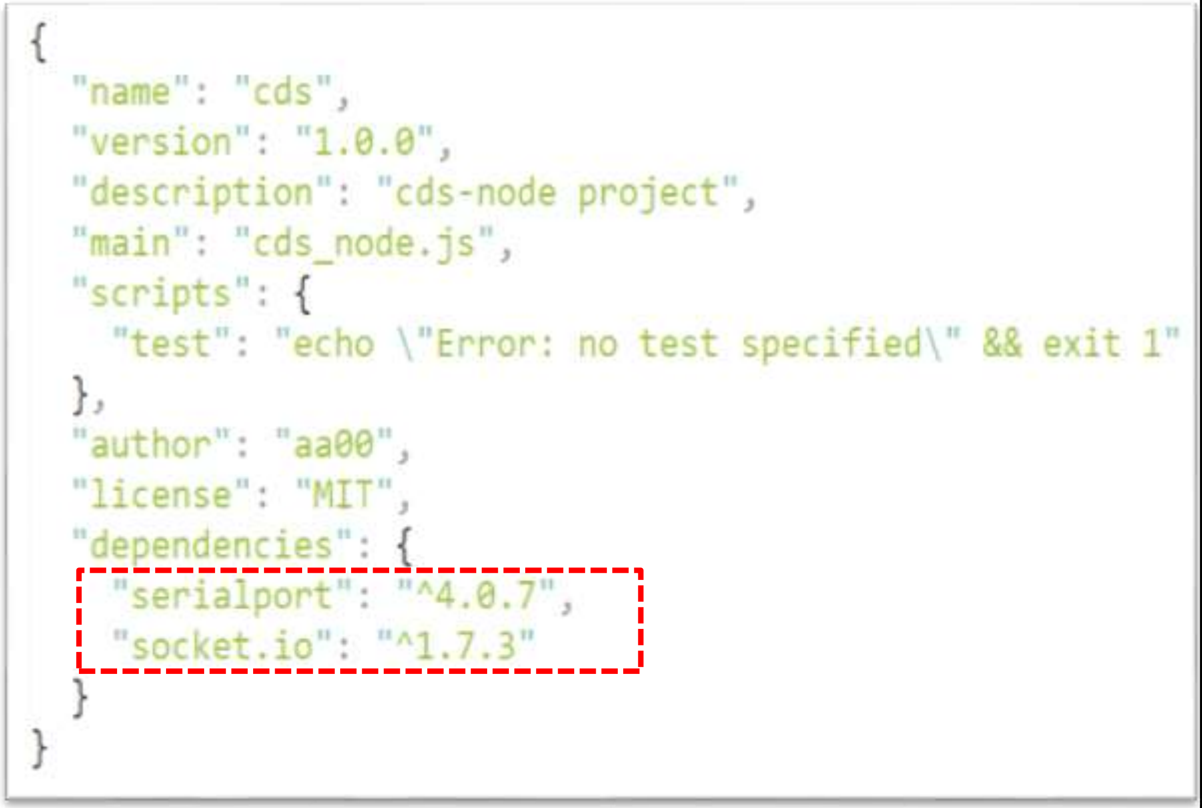
1. **Make cds node project**

- ➢ **md cds**
- ➢ **cd cds**

2. **Go to cds subfolder**

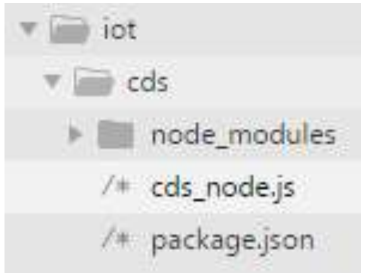
- ➢ **npm init**
- ➢ **npm install –save serialport@4.0.7**
- ➢ **npm install –save socket.io@1.7.3**

package,json

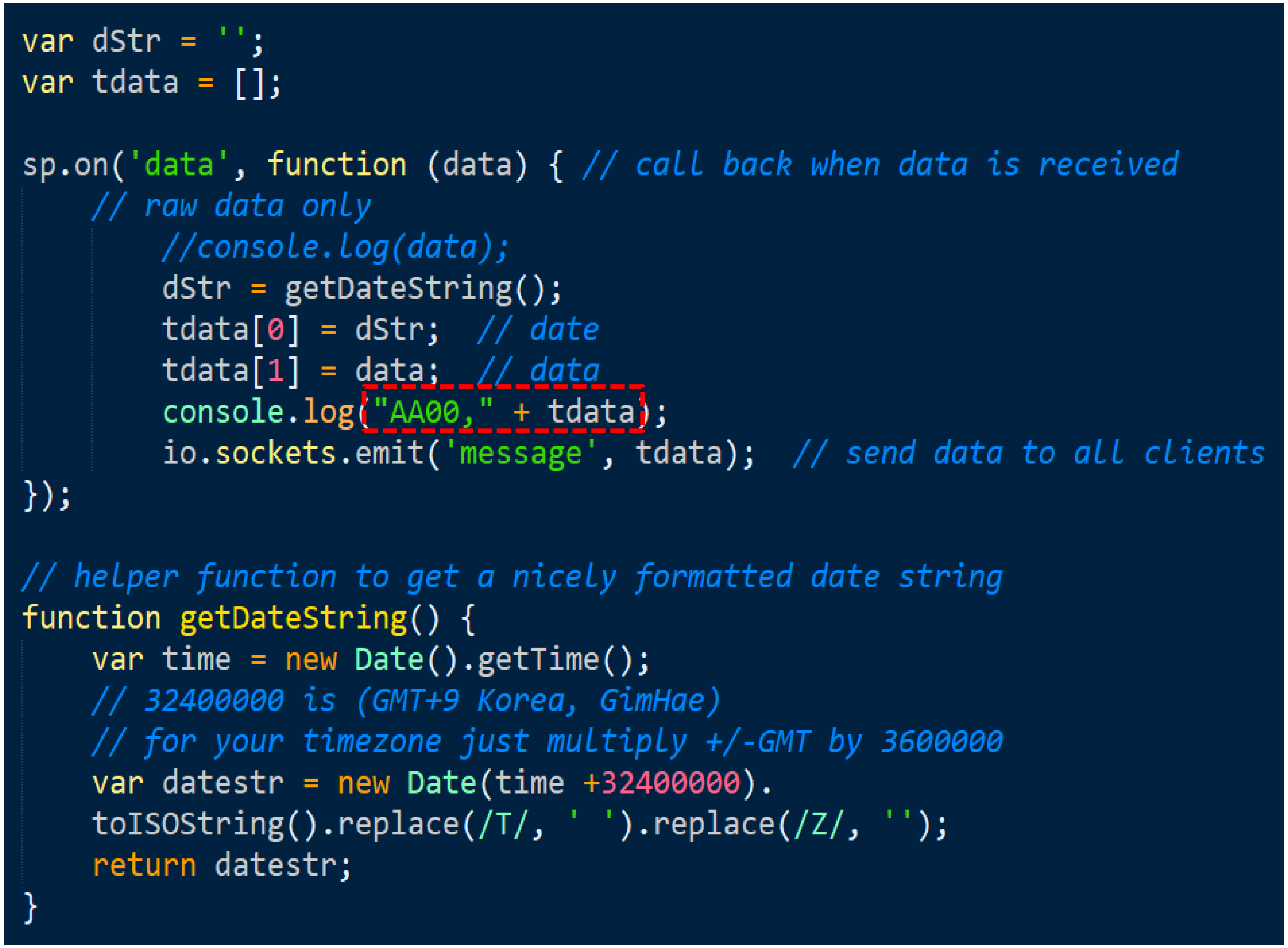


```
{
  "name": "cds",
  "version": "1.0.0",
  "description": "cds-node project",
  "main": "cds_node.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "aa00",
  "license": "MIT",
  "dependencies": {
    "serialport": "^4.0.7",
    "socket.io": "^1.7.3"
 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

A4.2.4 Luminosity sensor [Photocell LDR]



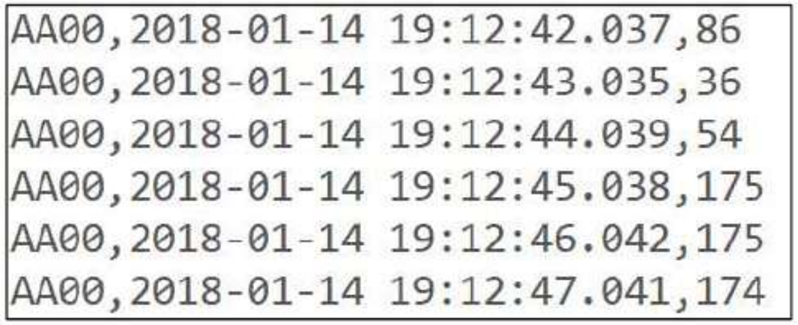
Save tmp36_node.js as cds_node.js



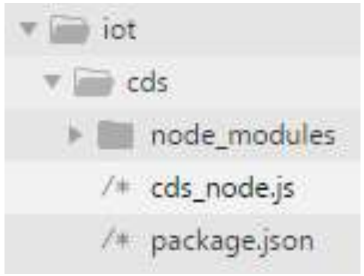
```
var dStr = '';
var tdata = [];

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



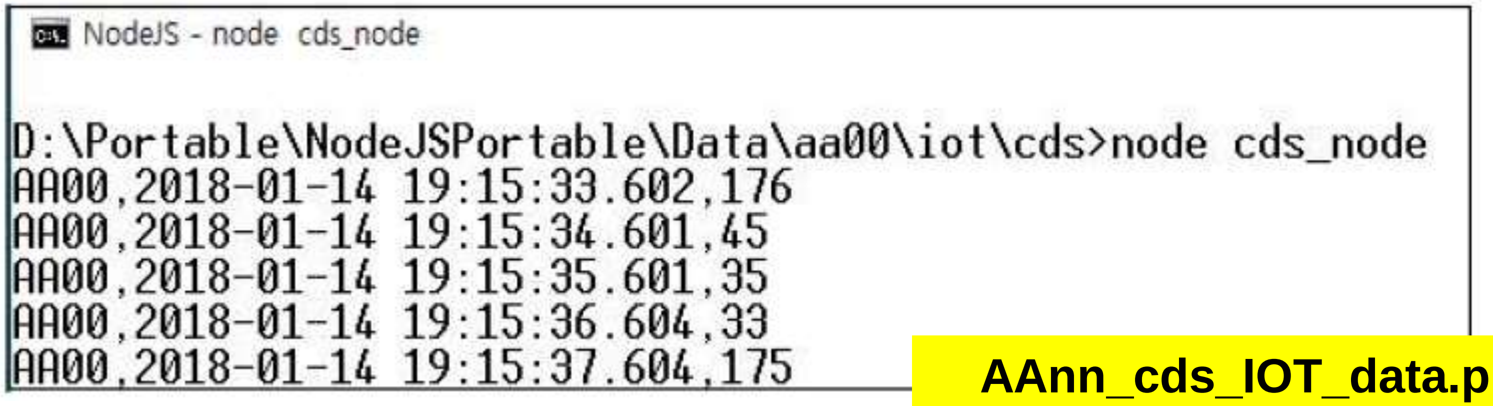
```
AA00,2018-01-14 19:12:42.037,86
AA00,2018-01-14 19:12:43.035,36
AA00,2018-01-14 19:12:44.039,54
AA00,2018-01-14 19:12:45.038,175
AA00,2018-01-14 19:12:46.042,175
AA00,2018-01-14 19:12:47.041,174
```



```
▼ iot
  ▼ cds
    ▸ node_modules
      cds_node.js
      package.json
```

▸ Node cmd에서 실행

```
node cds_node
```



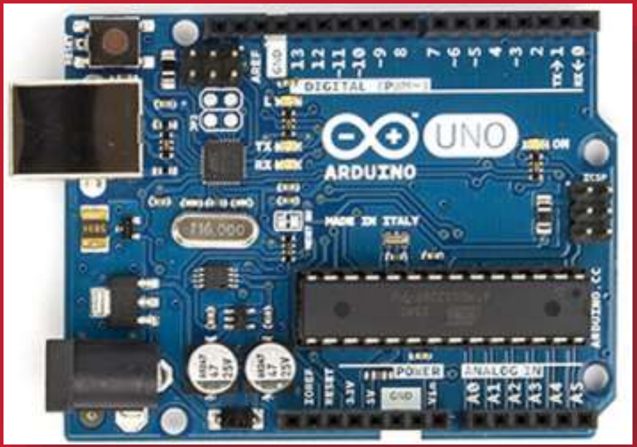
```
D:\Portable\NodeJSPortable\Data\aa00\iot\cds>node cds_node
AA00,2018-01-14 19:15:33.602,176
AA00,2018-01-14 19:15:34.601,45
AA00,2018-01-14 19:15:35.601,35
AA00,2018-01-14 19:15:36.604,33
AA00,2018-01-14 19:15:37.604,175
```

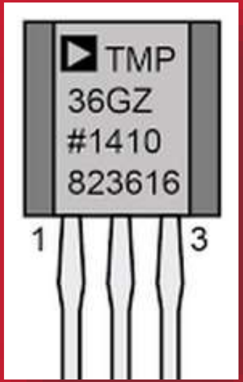
AAnn_cds_IOT_data.png 로 저장

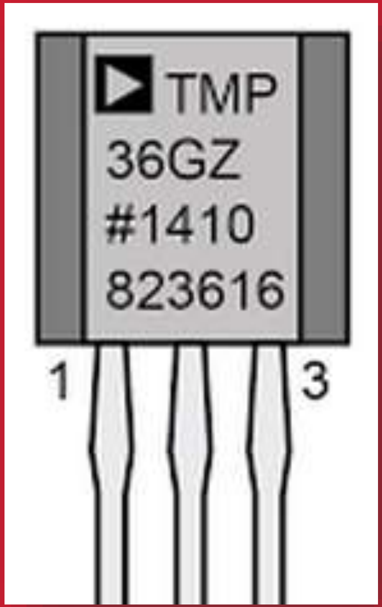
Multiple sensors

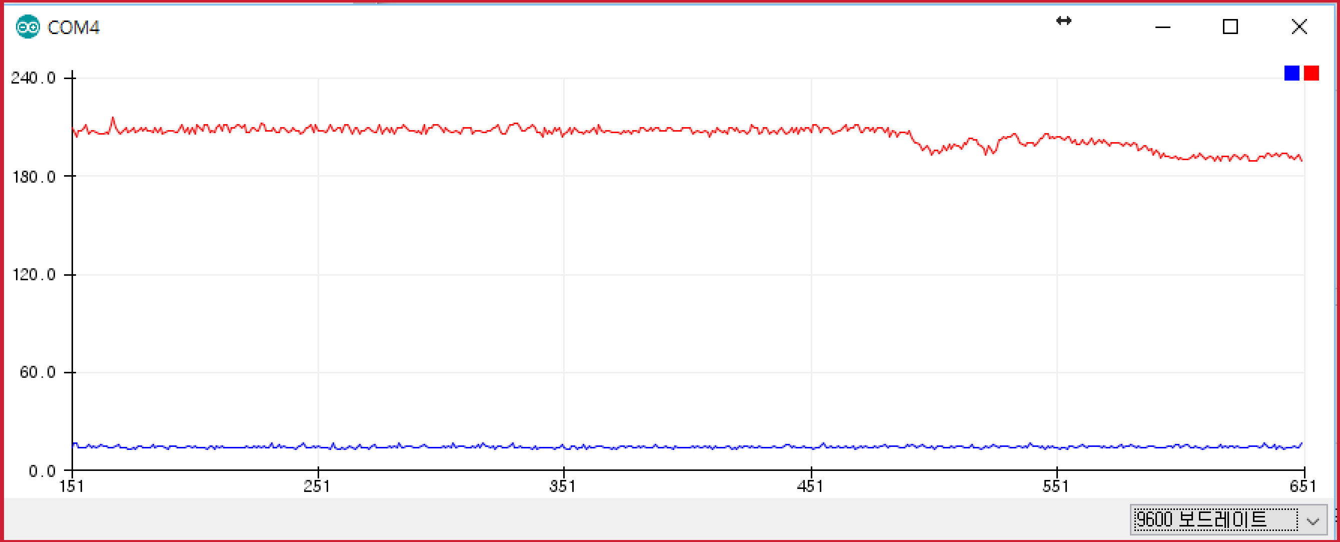
Arduino

+ Node.js

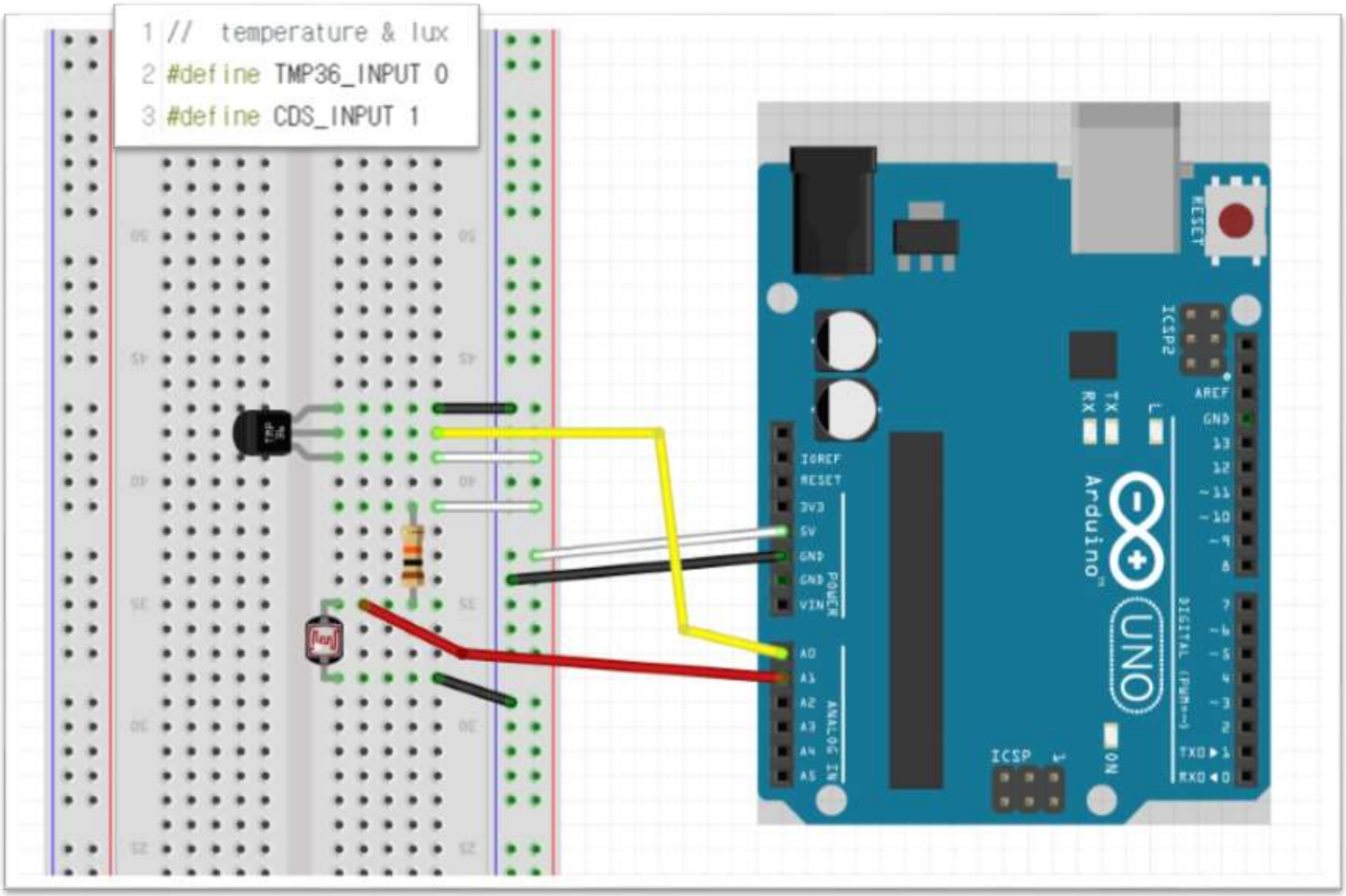





TMP
36GZ
#1410
823616
1
3

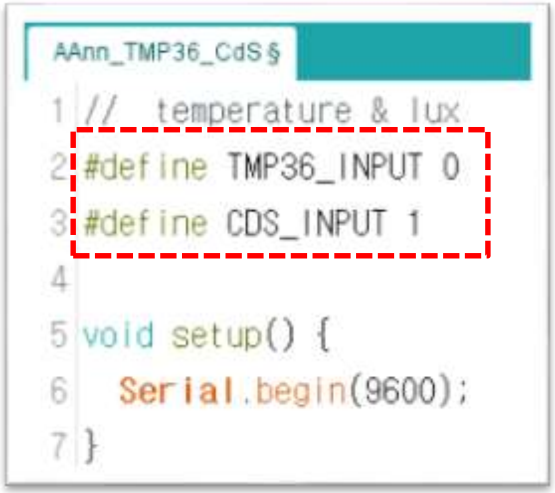

COM4
240.0
180.0
120.0
60.0
0.0
151
251
351
451
551
651
9600 보드레이트

A4.3.1 TMP36 + CdS : circuit



```
// temperature & lux
#define TMP36_INPUT 0
#define CDS_INPUT 1
```

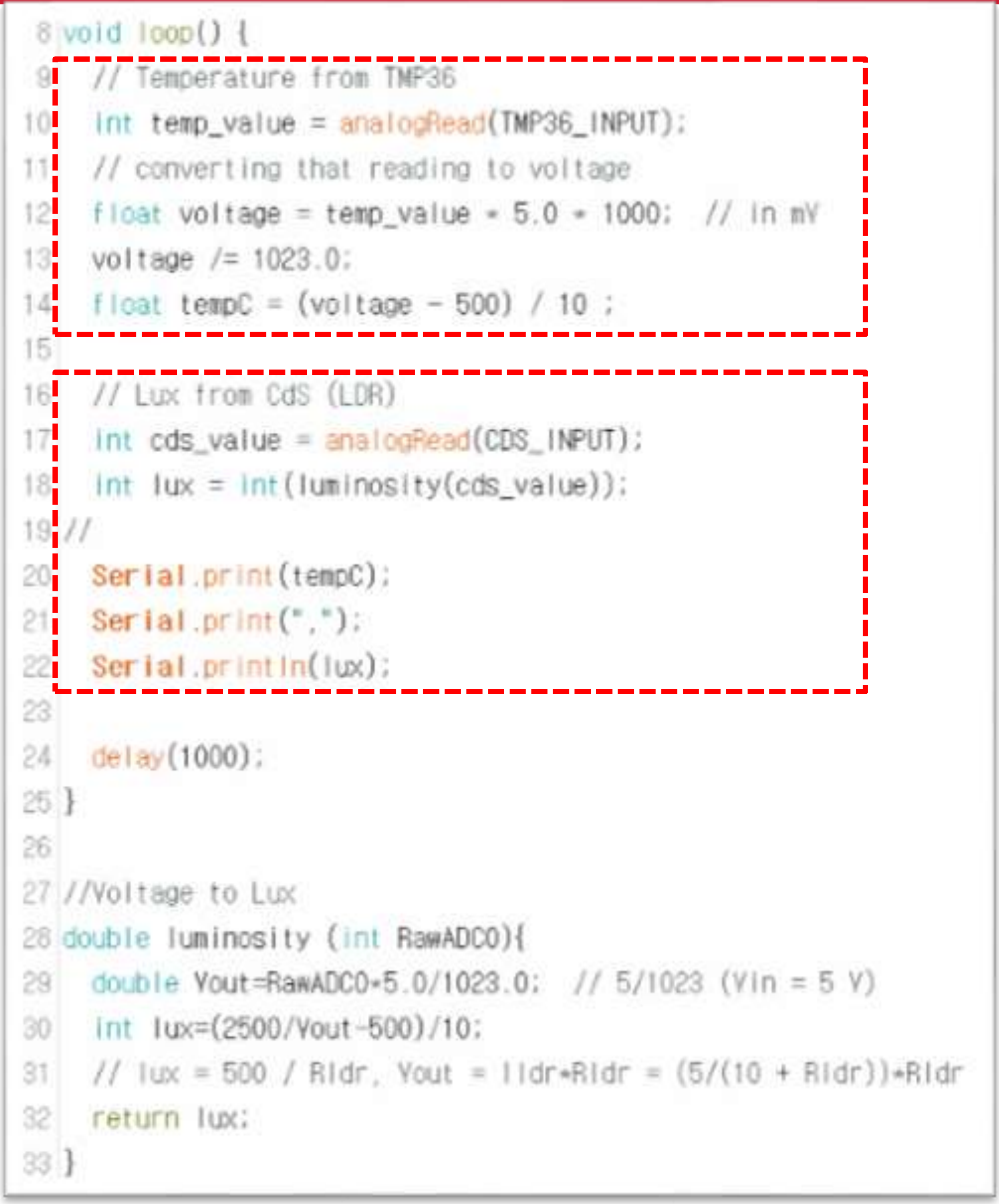
A4.3.2 TMP36 + CdS : code



AAnn_TMP36_CdS §

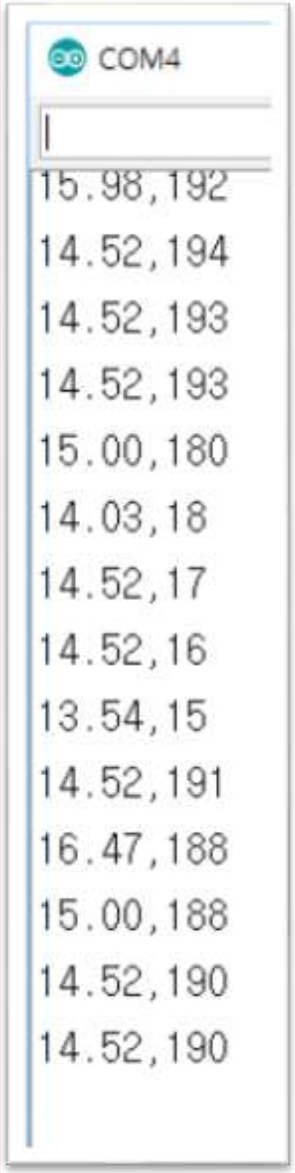
```
//  temperature & lux
#define TMP36_INPUT 0
#define CDS_INPUT 1

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

AAnn_tmp36_cds.ino

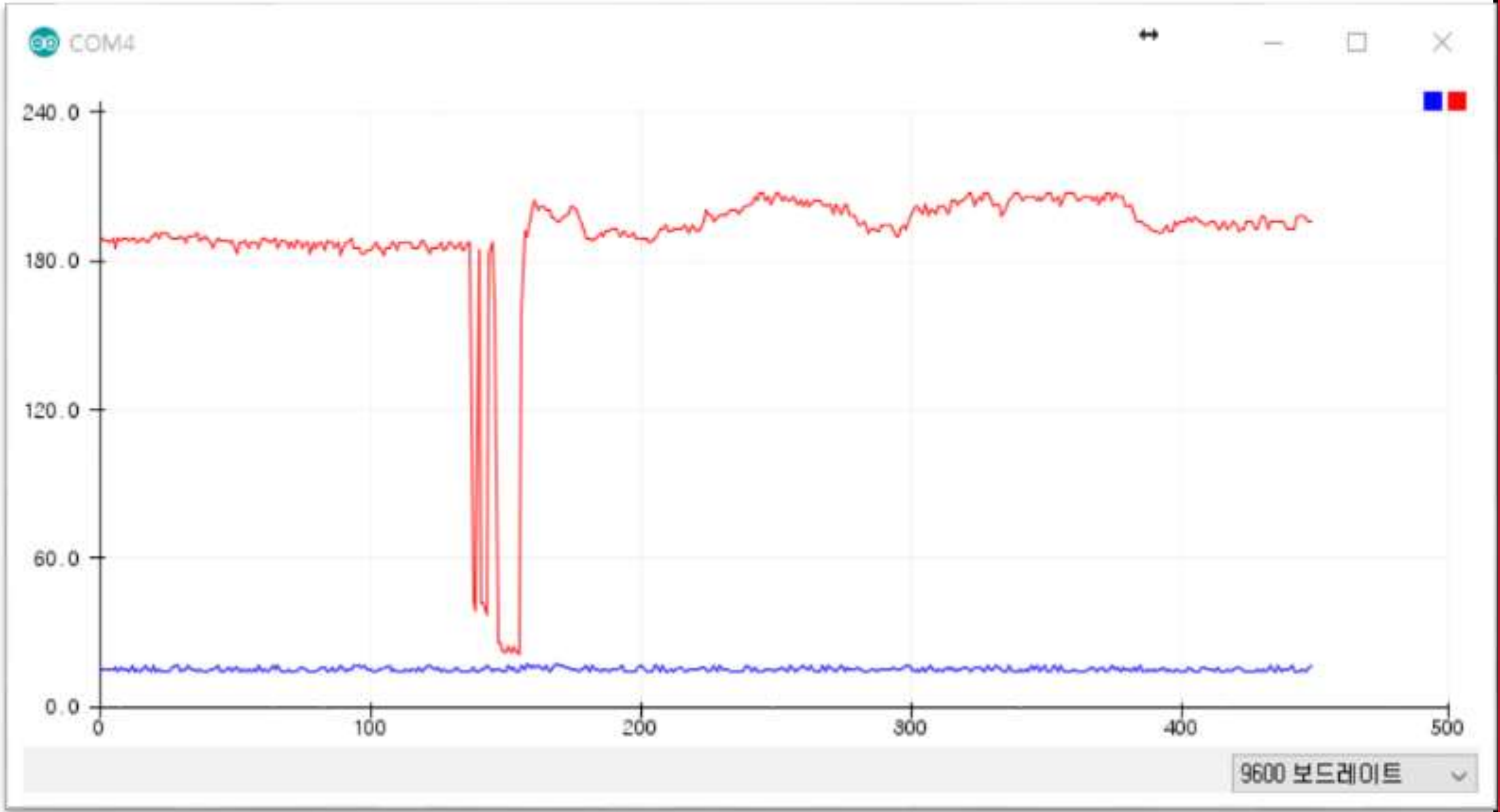


```
void loop() {
  // Temperature from TMP36
  int temp_value = analogRead(TMP36_INPUT);
  // converting that reading to voltage
  float voltage = temp_value * 5.0 * 1000;   // in mV
  voltage /= 1023.0;
  float tempC = (voltage - 500) / 10 ;

  // Lux from CdS (LDR)
  int cds_value = analogRead(CDS_INPUT);
  int lux = int(luminosity(cds_value));
//
  Serial.print(tempC);
  Serial.print(",");
  Serial.println(lux);

  delay(1000);
}

//Voltage to Lux
double luminosity (int RawADC0){
  double Vout=RawADC0*5.0/1023.0;  // 5/1023 (Vin = 5 V)
  int lux=(2500/Vout-500)/10;
  // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
  return lux;
}
```

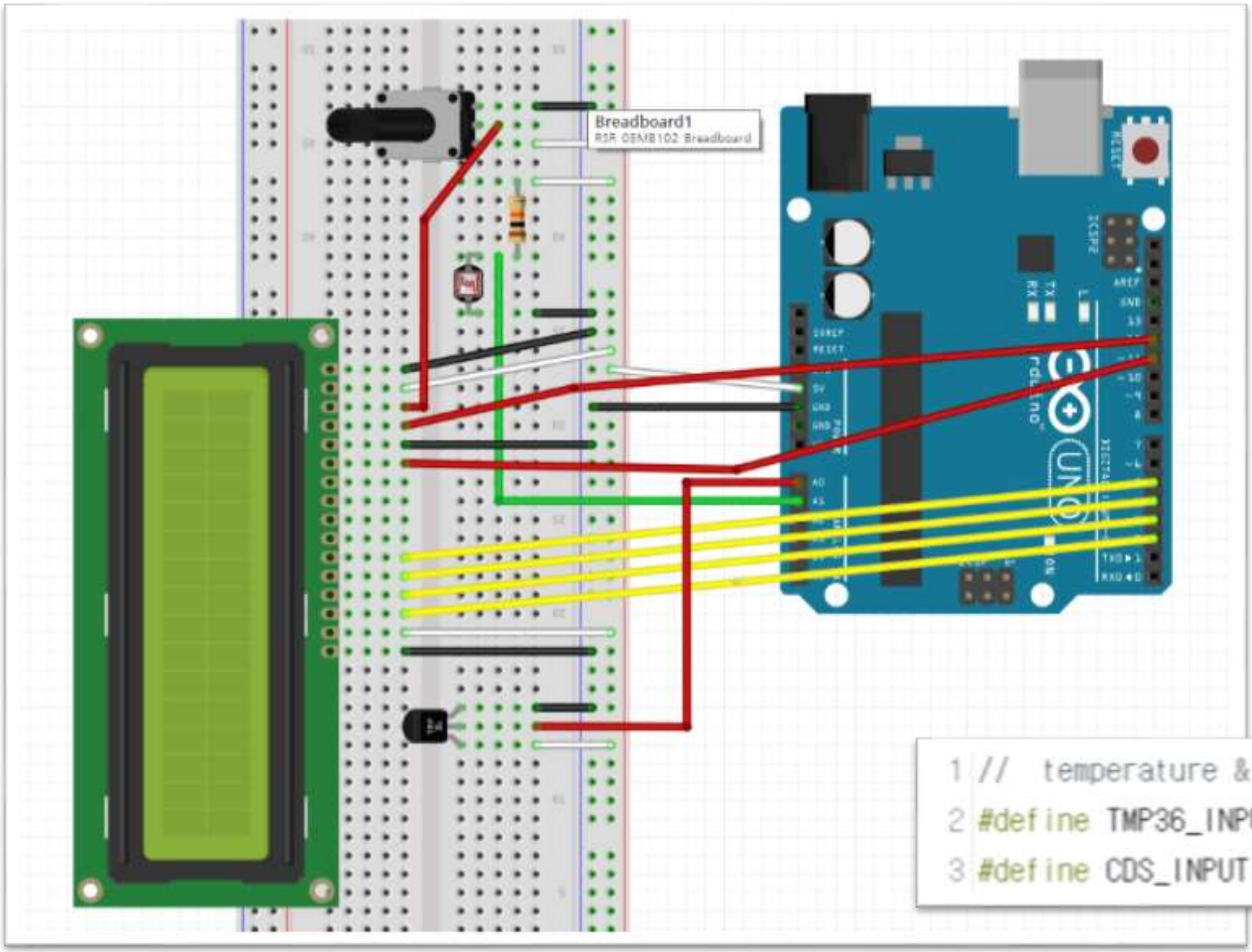
A4.3.3 TMP36 + CdS : result



```
COM4

15.98,192
14.52,194
14.52,193
14.52,193
15.00,180
14.03,18
14.52,17
14.52,16
13.54,15
14.52,191
16.47,188
15.00,188
14.52,190
14.52,190
```

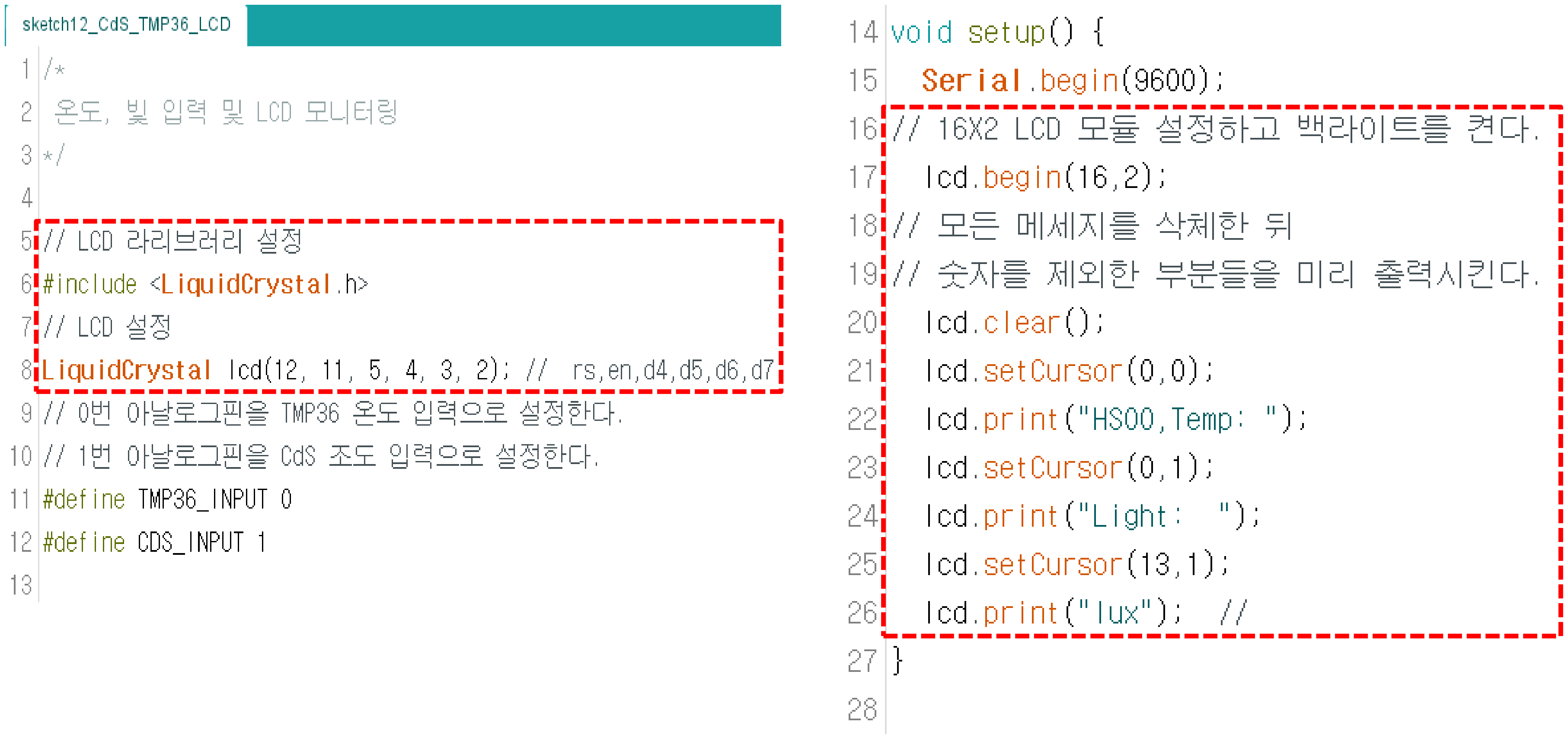


A4.4.1 TMP36 + CdS + LCD : circuit



```
// temperature & lux
#define TMP36_INPUT 0
#define CDS_INPUT 1
```

A4.4.2 TMP36 + CdS + LCD : code-1



sketch12_CdS_TMP36_LCD

```
/*
 온도, 빛 입력 및 LCD 모니터링
*/

// LCD 라리브러리 설정
#include <LiquidCrystal.h>
// LCD 설정
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //  rs,en,d4,d5,d6,d7
// 0번 아날로그핀을 TMP36 온도 입력으로 설정한다.
// 1번 아날로그핀을 CdS 조도 입력으로 설정한다.
#define TMP36_INPUT 0
#define CDS_INPUT 1

void setup() {
  Serial.begin(9600);
// 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,Temp: ");
  lcd.setCursor(0,1);
  lcd.print("Light:  ");
  lcd.setCursor(13,1);
  lcd.print("lux");  //
}

```

A4.4.3 TMP36 + CdS + LCD : code-2

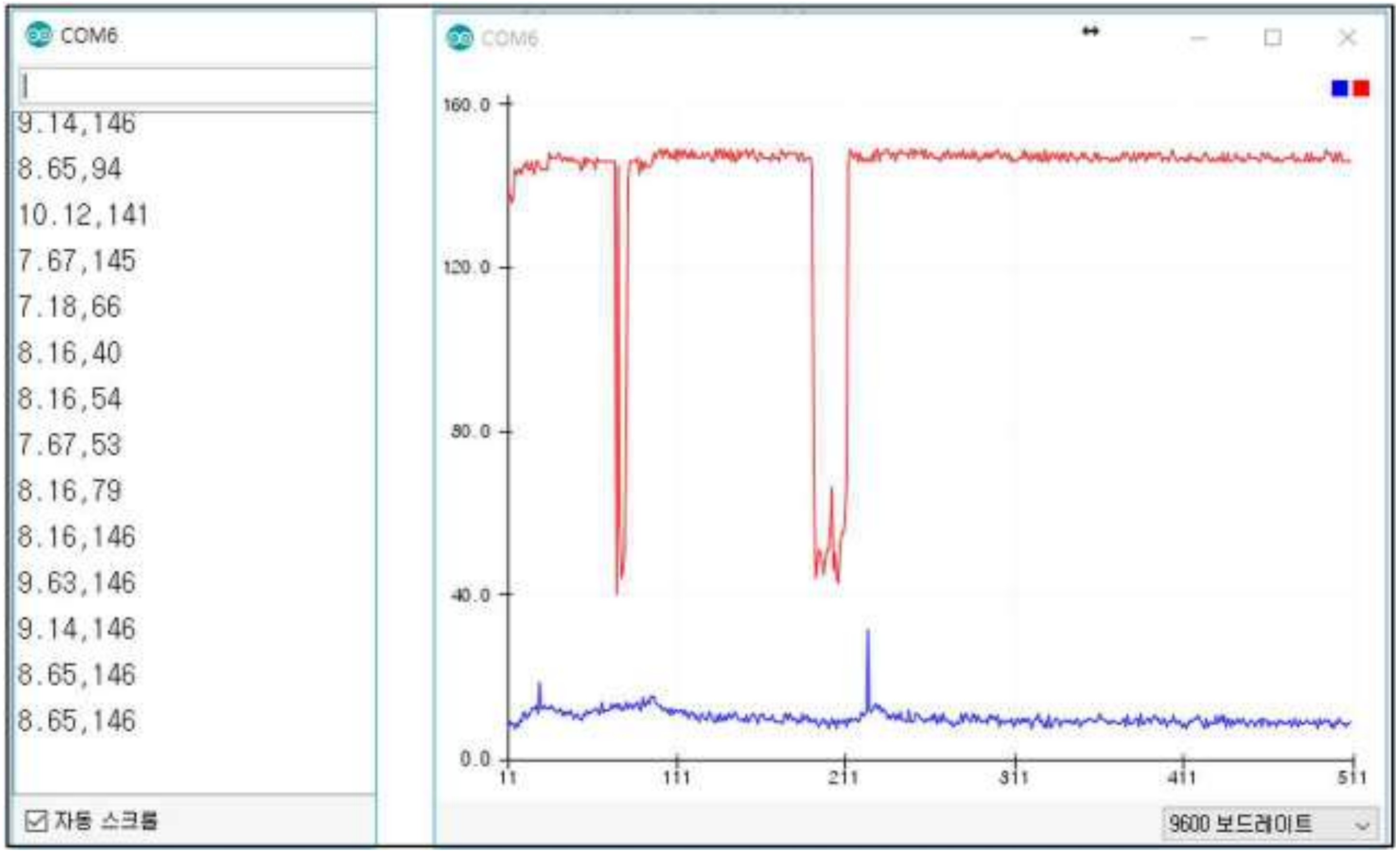
```
void loop(){
  // Temperature from TMP36
  int temp_value = analogRead(TMP36_INPUT);
  // converting that reading to voltage
  float voltage = temp_value * 5.0 * 1000;  // in mV
  voltage /= 1023.0;
  float tempC = (voltage - 500) / 10 ;

  // Lux from CdS (LDR)
  int cds_value = analogRead(CDS_INPUT);
  int lux = int(luminosity(cds_value));

  // 전에 표시했던 내용을 지운다.
  lcd.setCursor(12,0);
  lcd.print("    ");
  // 온도를 표시한다
  lcd.setCursor(12,0);
  lcd.print(tempC);
  // 전에 표시했던 내용을 지운다.
  lcd.setCursor(9,1);
  lcd.print("   ");
  // 조도를 표시한다
  lcd.setCursor(9,1);
  lcd.print(lux);
```

LCD output

```
  // Serial output --> 온도,조도
  Serial.print(tempC);
  Serial.print(",");
  Serial.println(lux);
  delay(1000);
}

//Voltage to Lux
double luminosity (int RawADC0){
  double Vout=RawADC0*5.0/1023.0;  // 5/1023 (Vin = 5 V)
  double lux=(2500/Vout-500)/10.0;
  // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
  return lux;
}
```

Serial output

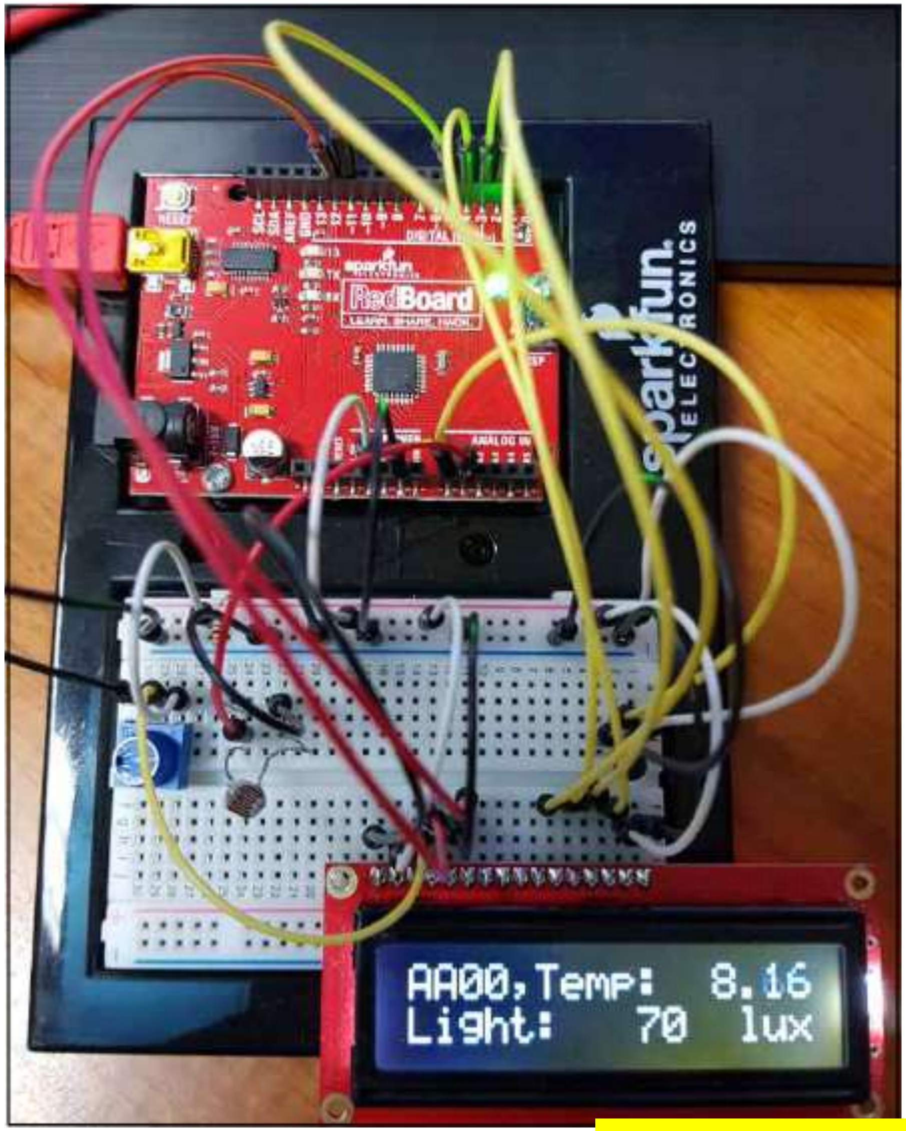
A4.4.4 TMP36 + CdS + LCD : result-1


COM6

```
9.14,146
8.65,94
10.12,141
7.67,145
7.18,66
8.16,40
8.16,54
7.67,53
8.16,79
8.16,146
9.63,146
9.14,146
8.65,146
8.65,146
```

☑ 자동 스크롤

A4.4.5 TMP36 + CdS + LCD : result-2

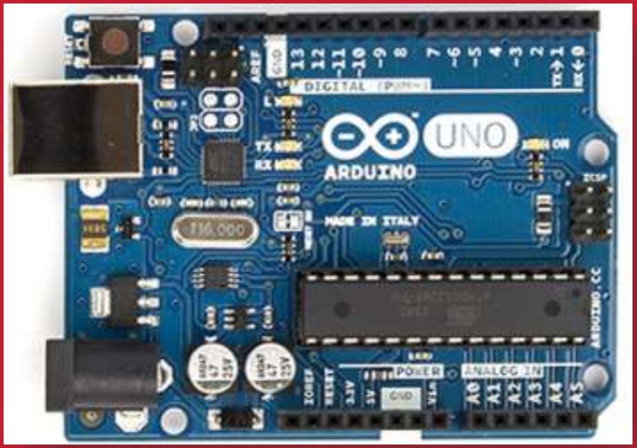


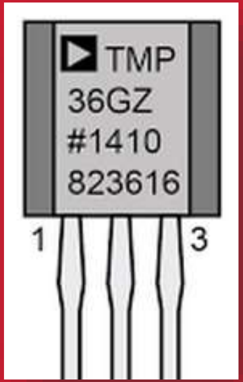
Save as
AAnn_cds_tmp36_lcd.png

Multiple sensors

CdS + TMP36

Node project





A4.5.1 CdS + TMP36 + Node project

1. Make cds_tmp36 node project
   - md cds_tmp36 in iot folder
   - cd cds_tmp36
2. Go to cds_tmp36 subfolder
   - npm init

"main": "cds_tmp36_node.js"
"author": “aann"

name : cds_tmp36

description : cds-tmp36-node project

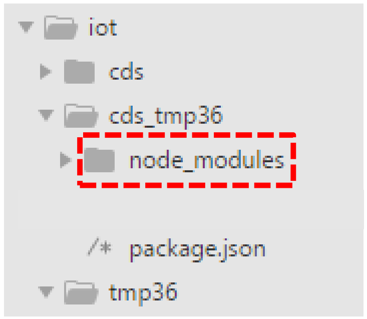
entry point : cds_tmp36_node.js

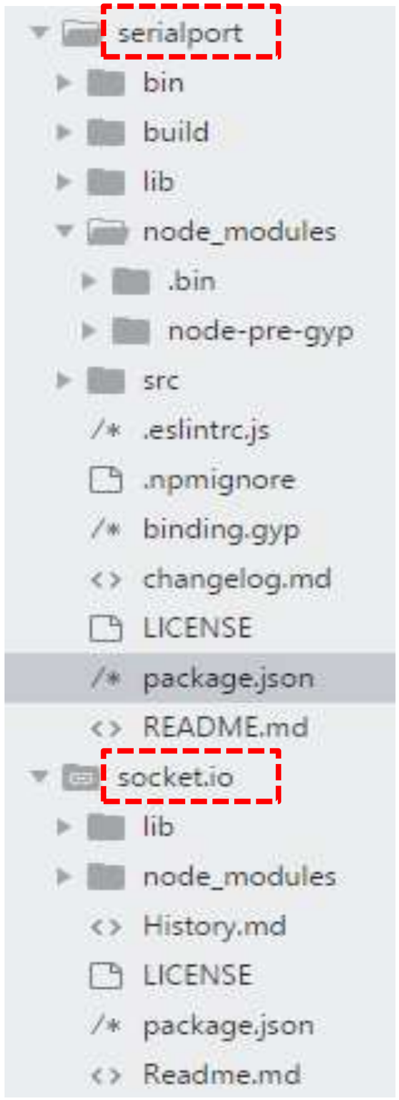
author : hsnn

A4.5.2 CdS + TMP36 + Node project

1. Make cds_tmp36 node project
- ➢ md cds_tmp36 in iot folder
- ➢ cd cds_tmp36

2. Go to cds_tmp36 subfolder
- ➢ npm init
- ➢ **npm install –save serialport@4.0.7**
- ➢ **npm install –save socket.io@1.7.3**


```
▼ iot
  ▶ cds
  ▼ cds_tmp36
    ▶ node_modules
      package.json
  ▼ tmp36
```

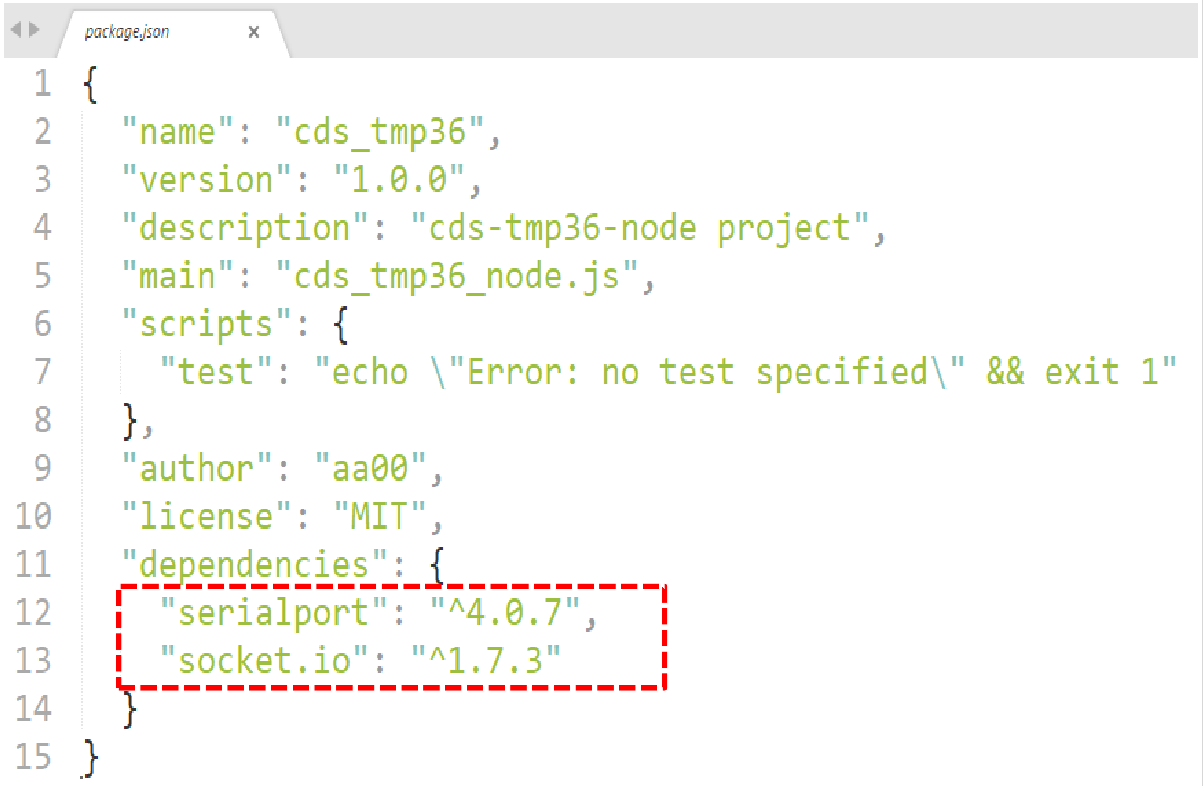

```
▼ serialport
  ▶ bin
  ▶ build
  ▶ lib
  ▼ node_modules
    ▶ .bin
    ▶ node-pre-gyp
  ▶ src
    .eslintrc.js
    .npmignore
    binding.gyp
    changelog.md
    LICENSE
    package.json
    README.md
▼ socket.io
  ▶ lib
  ▶ node_modules
    History.md
    LICENSE
    package.json
    Readme.md
```

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

A4.5.3 CdS + TMP36 + Node project

1. Make cds_tmp36 node project
   - ➢ md cds_tmp36
   - ➢ cd cds_tmp36
2. Go to cds_tmp36 subfolder
   - ➢ npm init
   - ➢ npm install –save serialport@4.0.7
   - ➢ npm install –save socket.io@1.7.3

package,json

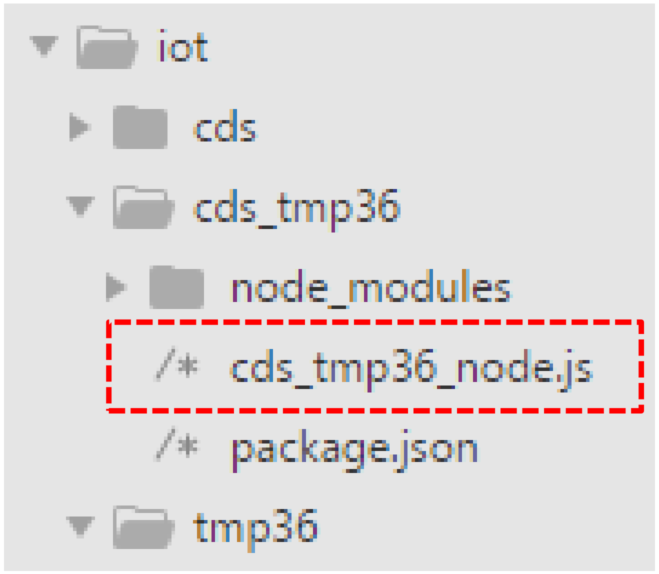


```
{
  "name": "cds_tmp36",
  "version": "1.0.0",
  "description": "cds-tmp36-node project",
  "main": "cds_tmp36_node.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "aa00",
  "license": "MIT",
  "dependencies": {
    "serialport": "^4.0.7",
    "socket.io": "^1.7.3"
  }
}
```

A4.5.4 CdS + TMP36 + Node project

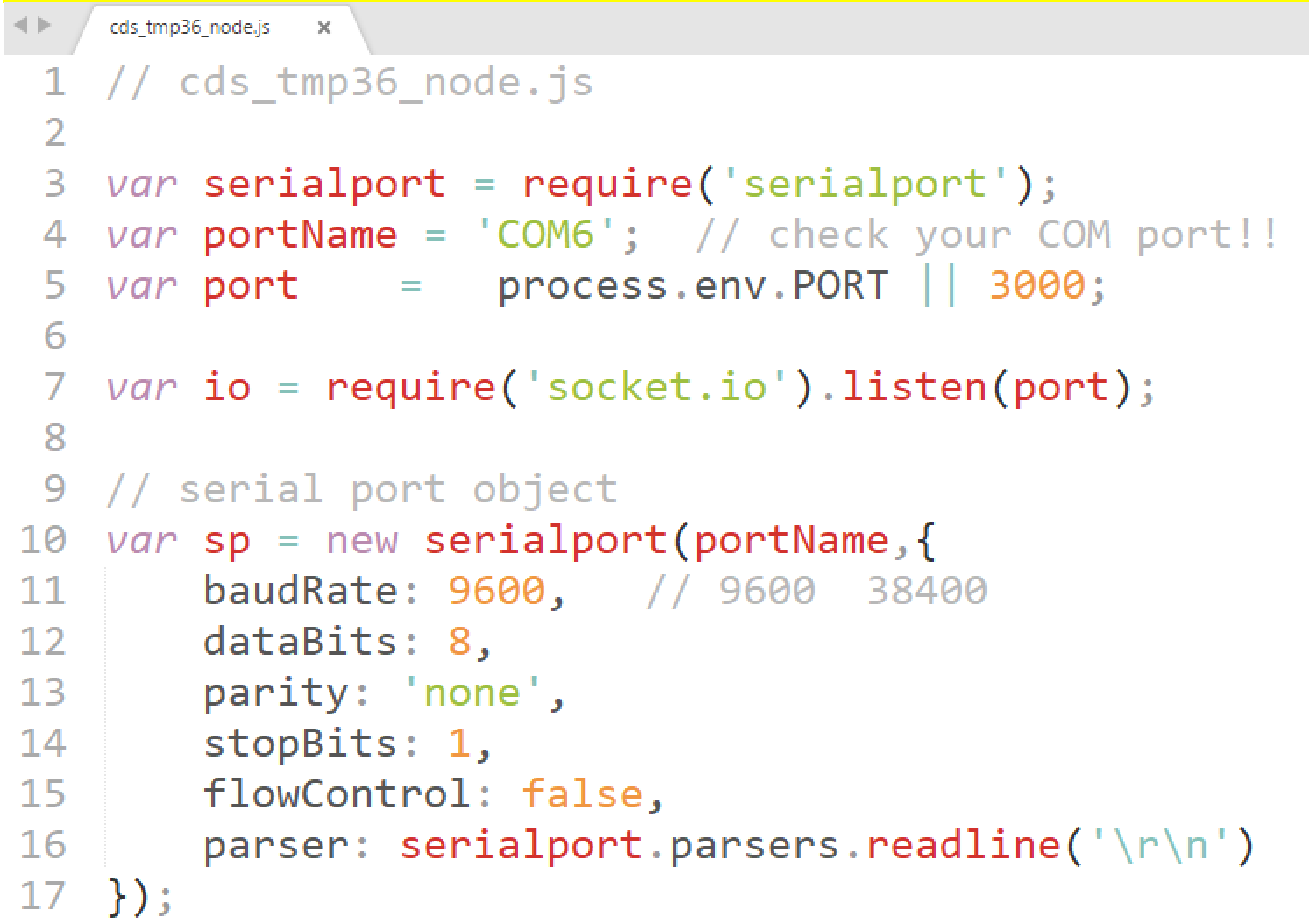
Recycling code:

Save cds_node.js as
cds_tmp36_node.js



A4.5.5.1 CdS + TMP36 + Node project : code-1

cds_tmp36_node.js

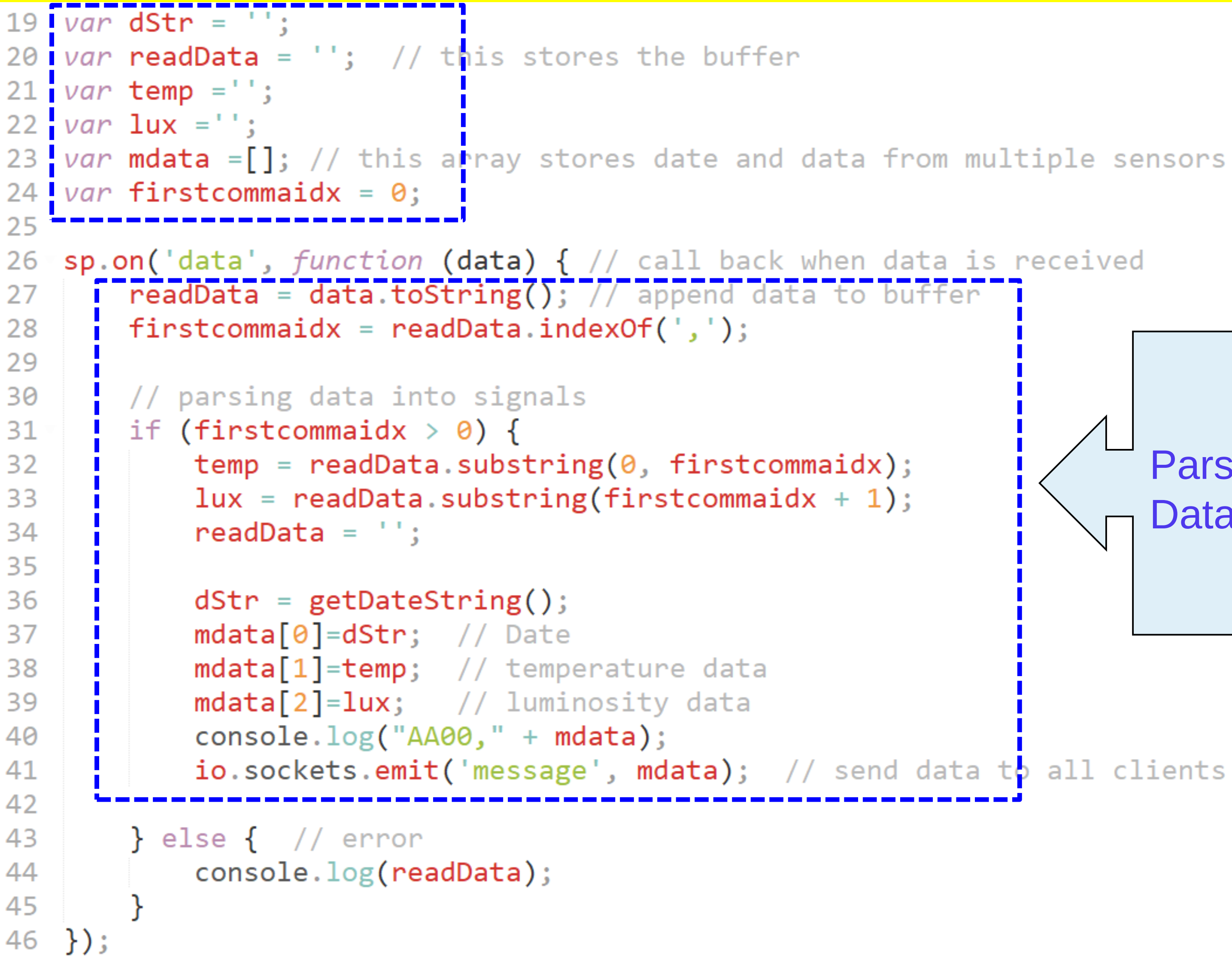


cds_tmp36_node.js ×

```
// cds_tmp36_node.js

var serialport = require('serialport');
var portName = 'COM6';  // check your COM port!!
var port     =    process.env.PORT || 3000;

var io = require('socket.io').listen(port);

// serial port object
var sp = new serialport(portName,{
    baudRate: 9600,   // 9600  38400
    dataBits: 8,
    parity: 'none',
    stopBits: 1,
    flowControl: false,
    parser: serialport.parsers.readline('\r\n')
});
```

A4.5.5.2 CdS + TMP36 + Node project : code-2

cds_tmp36_node.js – parsing data

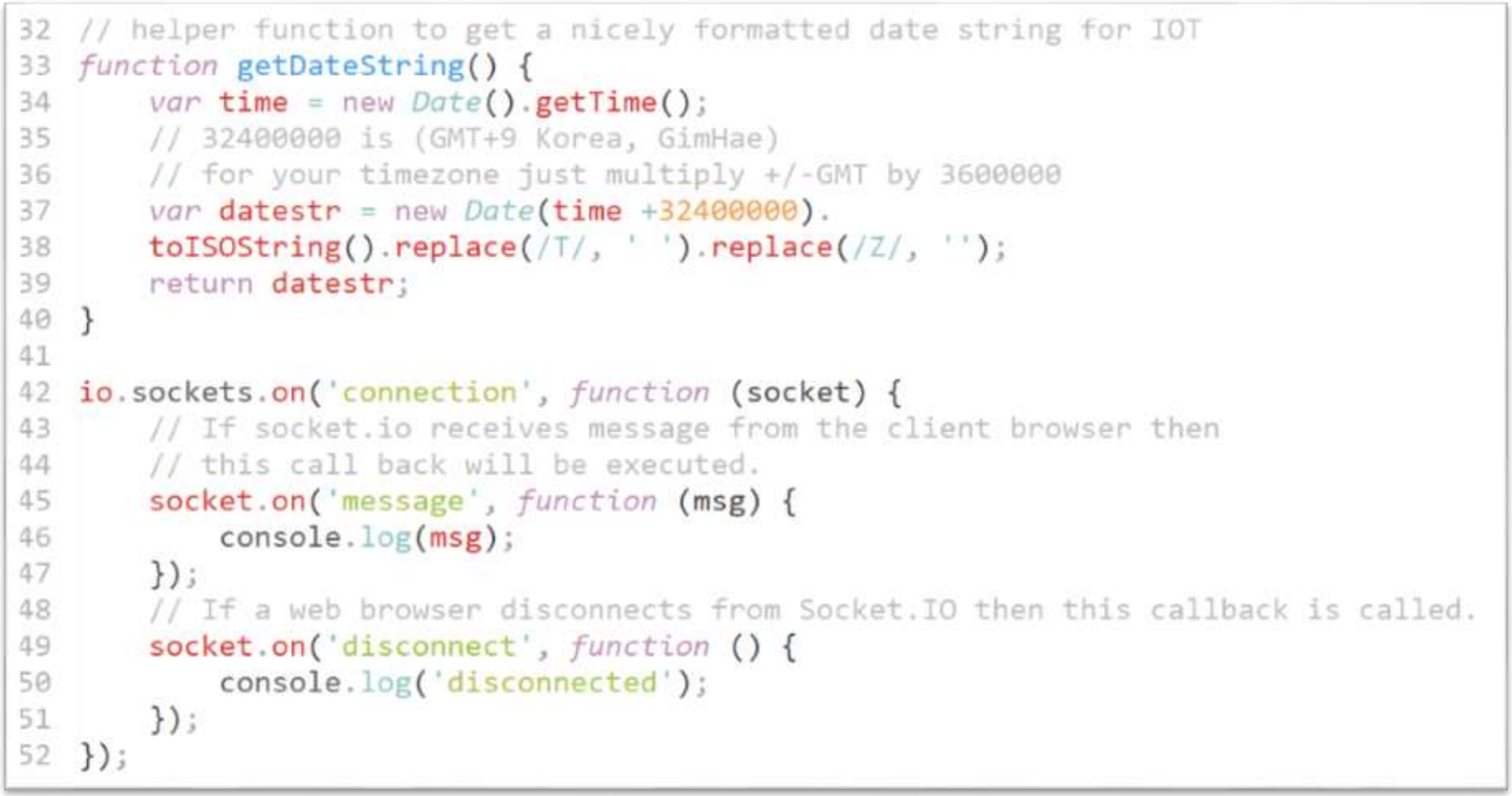


```
var dStr = '';
var readData = '';  // this stores the buffer
var temp ='';
var lux ='';
var mdata =[]; // this array stores date and data from multiple sensors
var firstcommaidx = 0;

sp.on('data', function (data) { // call back when data is received
    readData = data.toString(); // append data to buffer
    firstcommaidx = readData.indexOf(',');

    // parsing data into signals
    if (firstcommaidx > 0) {
        temp = readData.substring(0, firstcommaidx);
        lux = readData.substring(firstcommaidx + 1);
        readData = '';

        dStr = getDateString();
        mdata[0]=dStr;   // Date
        mdata[1]=temp;   // temperature data
        mdata[2]=lux;    // luminosity data
        console.log("AA00," + mdata);
        io.sockets.emit('message', mdata);  // send data to all clients

    } else {  // error
        console.log(readData);
    }
});
```

Parsing Data

A4.5.5.3 CdS + TMP36 + Node project : code-3

cds_tmp36_node.js

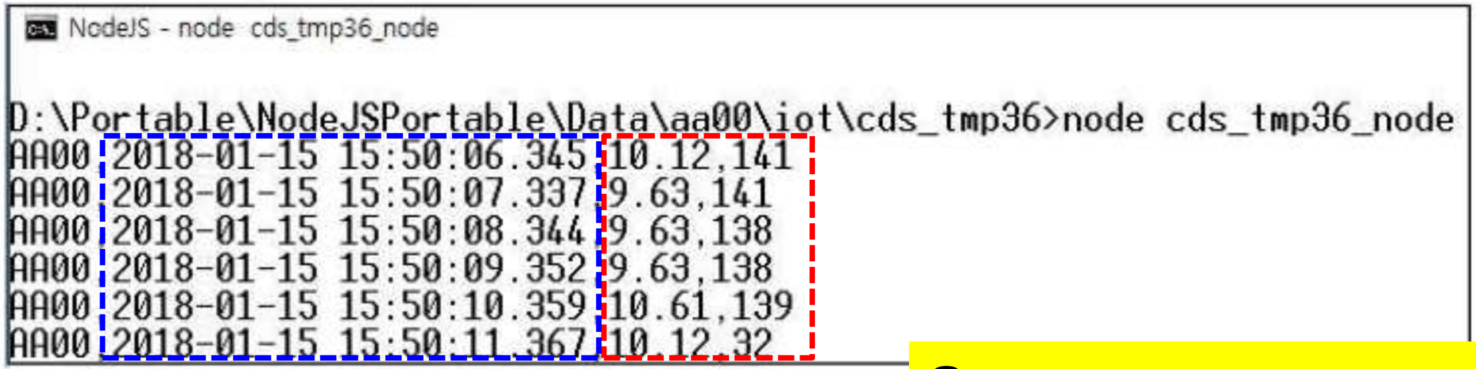


```
// helper function to get a nicely formatted date string for IOT
function getDateString() {
    var time = new Date().getTime();
    // 32400000 is (GMT+9 Korea, GimHae)
    // for your timezone just multiply +/-GMT by 3600000
    var datestr = new Date(time +32400000).
    toISOString().replace(/T/,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

io.sockets.on('connection', function (socket) {
    // If socket.io receives message from the client browser then
    // this call back will be executed.
    socket.on('message', function (msg) {
        console.log(msg);
    });
    // If a web browser disconnects from Socket.IO then this callback is called.
    socket.on('disconnect', function () {
        console.log('disconnected');
    });
});
```

A4.5.6 CdS + TMP36 + Node project : result

Node cmd 에서 실행

```
node cds_tmp36_node
```



NodeJS - node cds_tmp36_node

```
D:\Portable\NodeJSPortable\Data\aa00\iot\cds_tmp36>node cds_tmp36_node
AA00 2018-01-15 15:50:06.345 10.12,141
AA00 2018-01-15 15:50:07.337 9.63,141
AA00 2018-01-15 15:50:08.344 9.63,138
AA00 2018-01-15 15:50:09.352 9.63,138
AA00 2018-01-15 15:50:10.359 10.61,139
AA00 2018-01-15 15:50:11.367 10.12,32
```

IOT data format

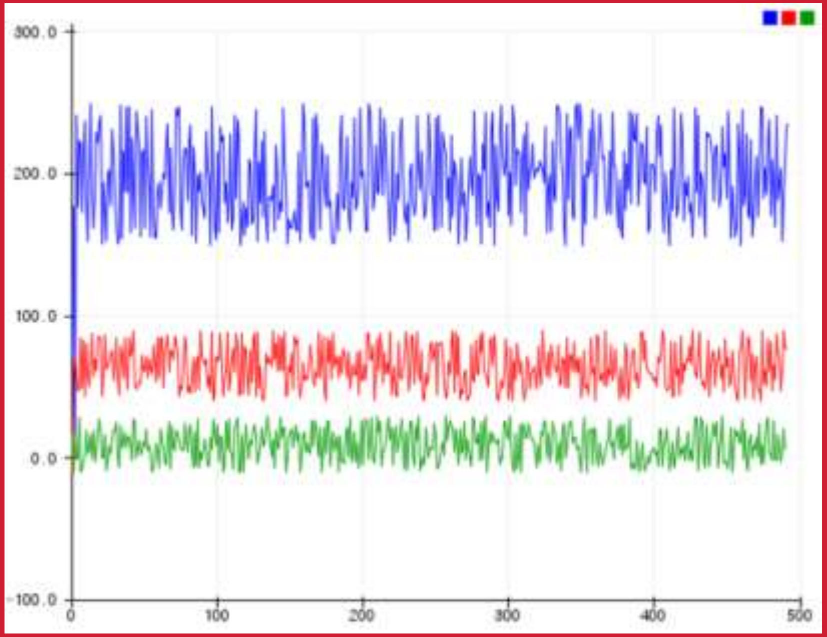
시간, 온도,조도

Save as
AAnn_cds_tmp36_IOT.png

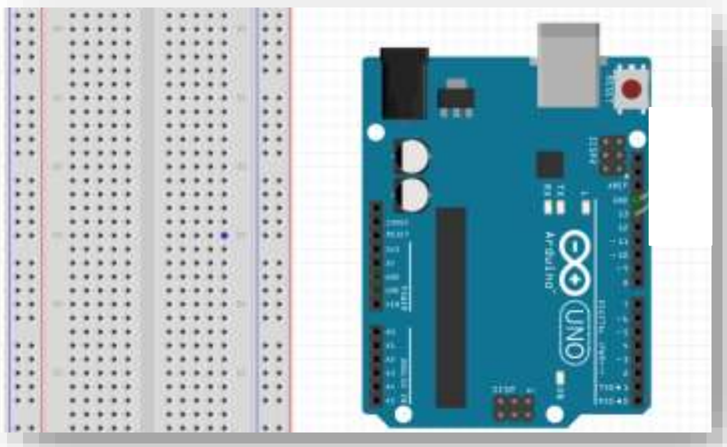
[DIY] Multi-signals

다중신호 시뮬레이션

+ node.js



DIY – 스케치



아두이노에서 LED와 저항을 모두 제거하고 USB만 컴퓨터와 연결한다.

전자 소자 연결 없이 마구잡이 수 생성 함수를 이용해서 조도, 습도, 온도에 해당되는 3개의 신호를 만든다.

온도는 값의 범위를 –10 ~ 30, 습도는 40 ~ 90, 그리고 조도는 150 ~ 250 으로 가상적 으로 설정한다.

직렬통신 모니터링을 이용해서 세 개의 신호의 변화를 모니터링 하는 코드를 만들어 결과를 확인한다.

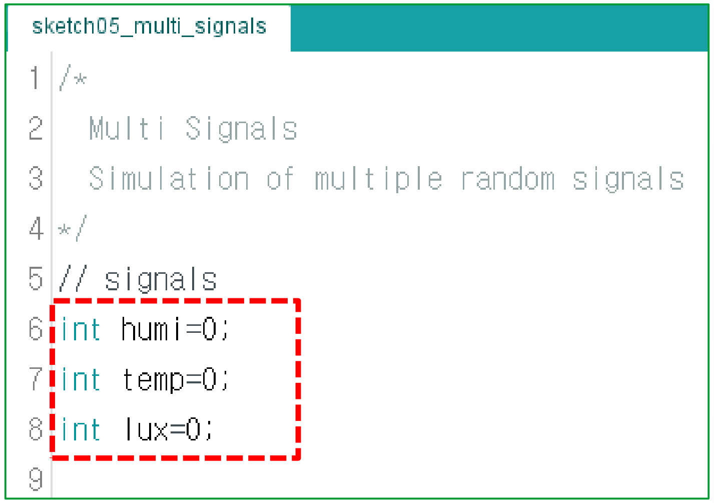
▸ 스케치 구성

1. 3 개의 신호를 담을 변수를 초기화한다.

2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.

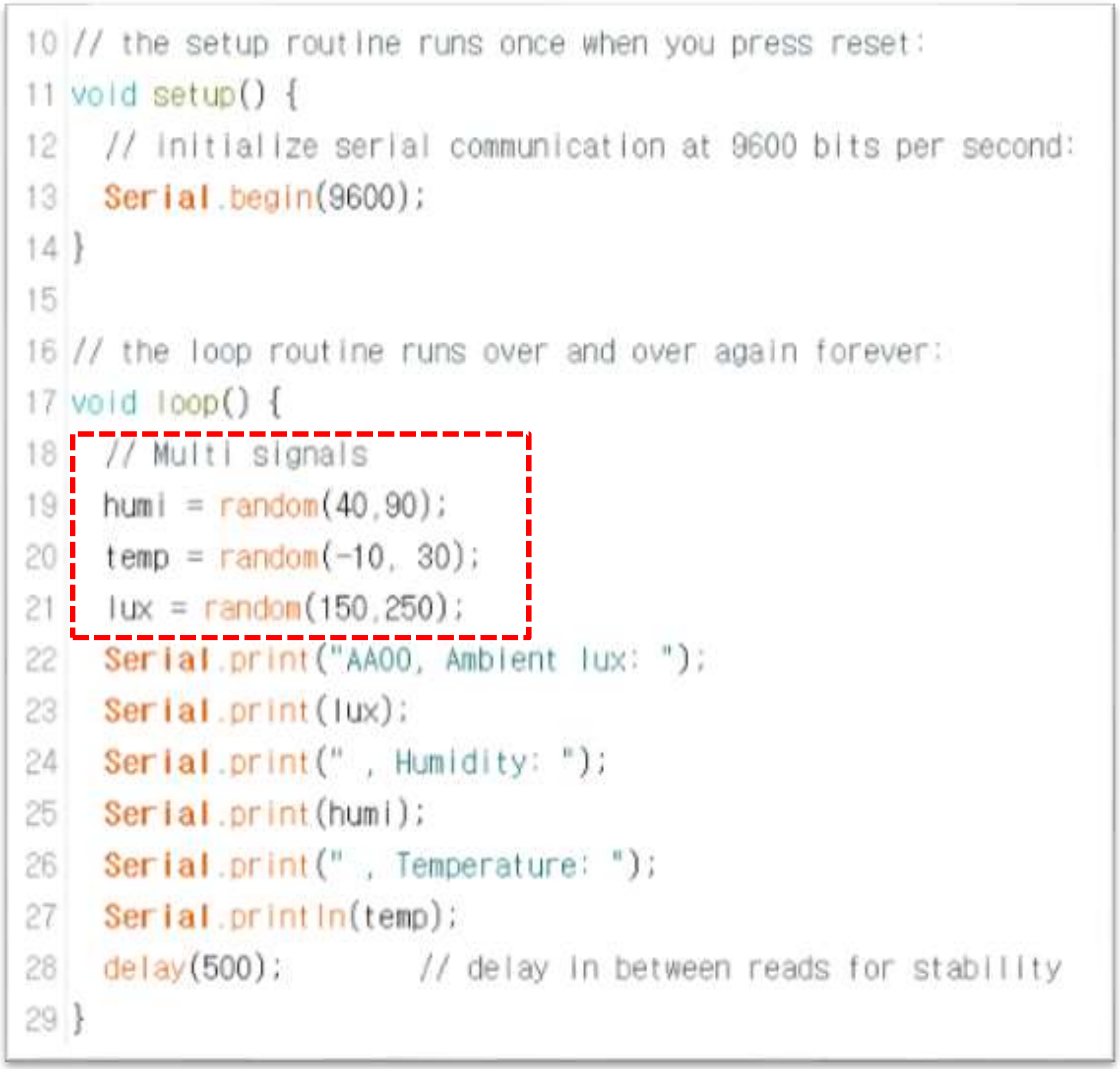
3. loop()에서 마구잡이 수를 세 개 발생시켜서 직렬 통신으로 3 개의 pwm 값을 각각 컴퓨터로 전송한다.

DIY – code



sketch05_multi_signals

```
/*
  Multi Signals
  Simulation of multiple random signals
*/
// signals
int humi=0;
int temp=0;
int lux=0;

```

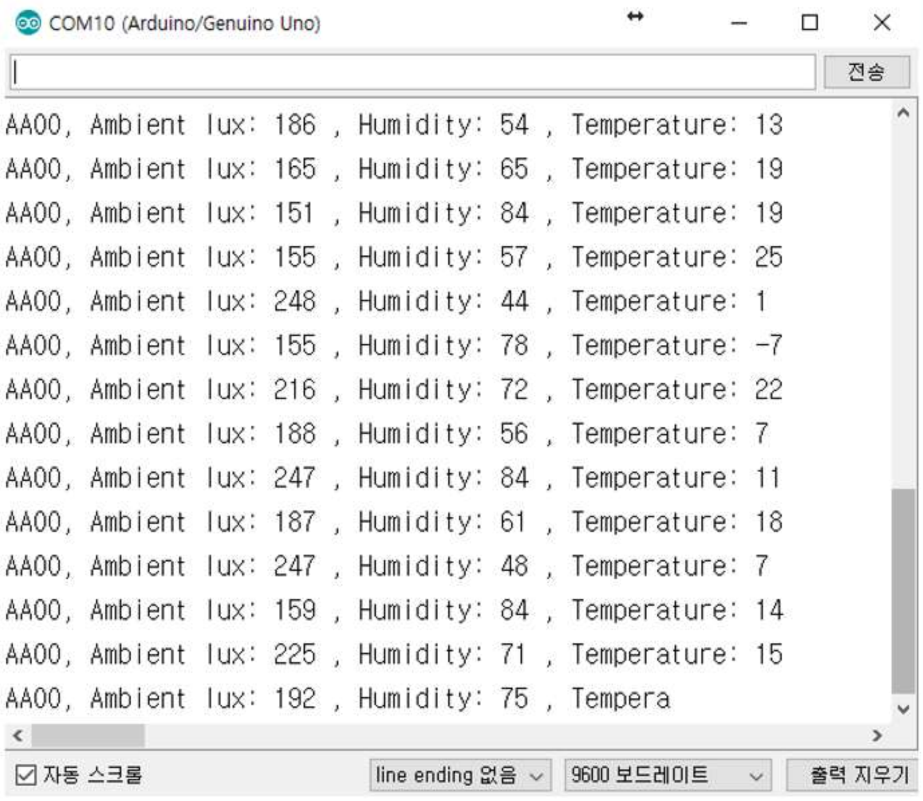


```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // Multi signals
  humi = random(40,90);
  temp = random(-10, 30);
  lux = random(150,250);
  Serial.print("AA00, Ambient lux: ");
  Serial.print(lux);
  Serial.print(" , Humidity: ");
  Serial.print(humi);
  Serial.print(" , Temperature: ");
  Serial.println(temp);
  delay(500);        // delay in between reads for stability
}
```

DIY – result

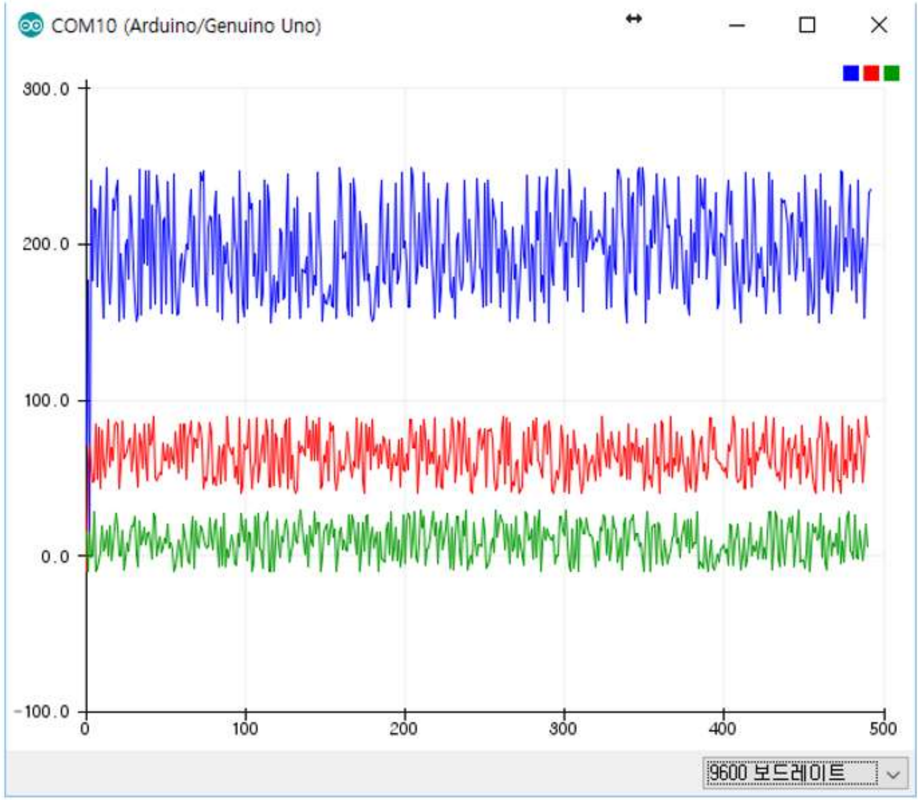
DIY 결과

가상적인 세 개의 센서신호 시뮬레이션: 조도(위), 습도(중간), 온도(아래).



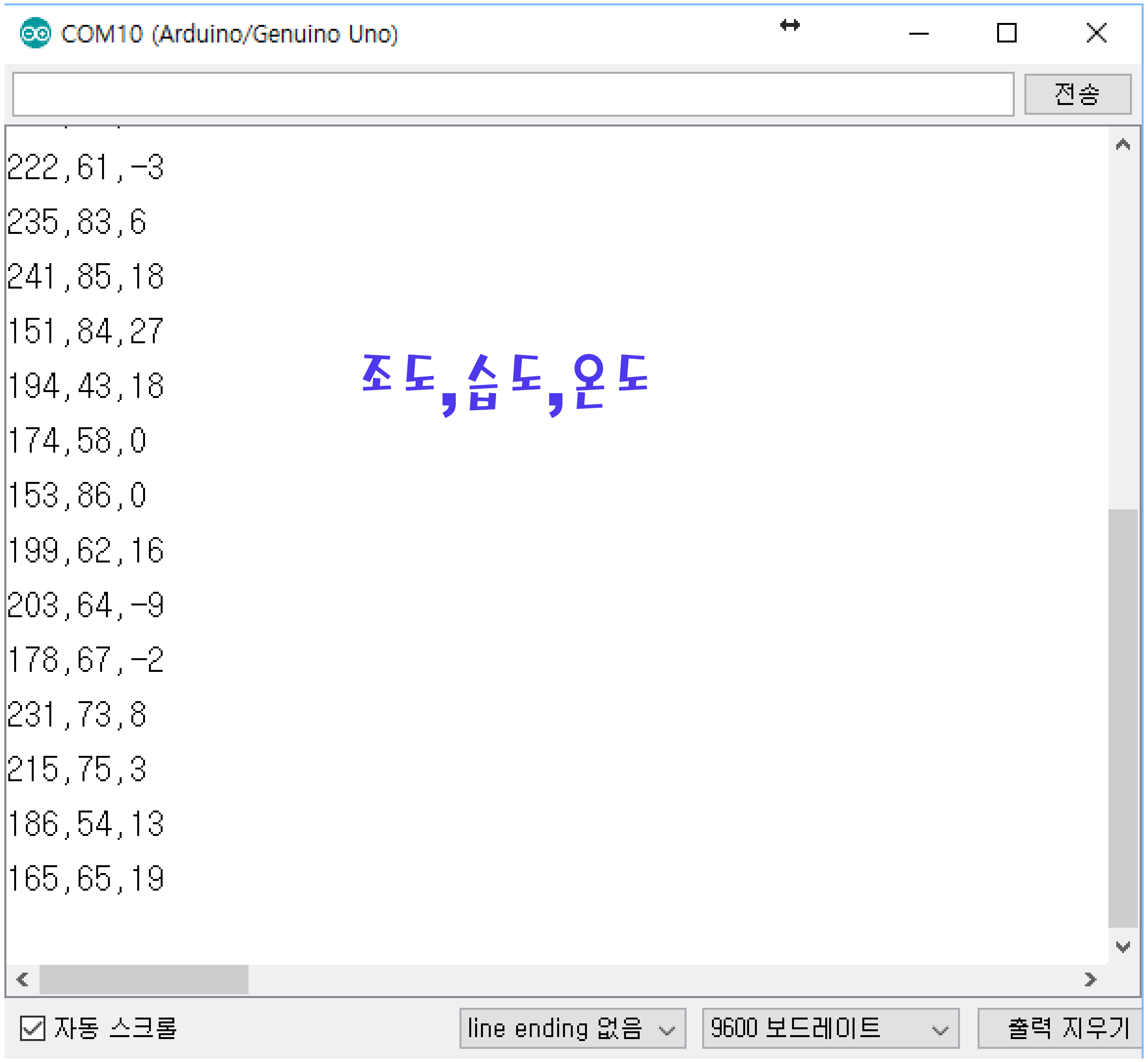
COM10 (Arduino/Genuino Uno)

```
AA00, Ambient lux: 186 , Humidity: 54 , Temperature: 13
AA00, Ambient lux: 165 , Humidity: 65 , Temperature: 19
AA00, Ambient lux: 151 , Humidity: 84 , Temperature: 19
AA00, Ambient lux: 155 , Humidity: 57 , Temperature: 25
AA00, Ambient lux: 248 , Humidity: 44 , Temperature: 1
AA00, Ambient lux: 155 , Humidity: 78 , Temperature: -7
AA00, Ambient lux: 216 , Humidity: 72 , Temperature: 22
AA00, Ambient lux: 188 , Humidity: 56 , Temperature: 7
AA00, Ambient lux: 247 , Humidity: 84 , Temperature: 11
AA00, Ambient lux: 187 , Humidity: 61 , Temperature: 18
AA00, Ambient lux: 247 , Humidity: 48 , Temperature: 7
AA00, Ambient lux: 159 , Humidity: 84 , Temperature: 14
AA00, Ambient lux: 225 , Humidity: 71 , Temperature: 15
AA00, Ambient lux: 192 , Humidity: 75 , Tempera
```



DIY – New result 1

DIY 결과 [1] : 가상적인 세 개의 센서신호 시뮬레이션 → 조도,습도,온도



```
222,61,-3
235,83,6
241,85,18
151,84,27
194,43,18
174,58,0
153,86,0
199,62,16
203,64,-9
178,67,-2
231,73,8
215,75,3
186,54,13
165,65,19
```

조도,습도,온도

DIY – New result 2-1

DIY 결과 [2] : 가상적인 세 개의 센서신호 시뮬레이션 → 조도,습도,온도를 Node.js로 처리

[1 단계] Node cmd

1. Make multi_signals node project

➢ md multi_signals

➢ cd multi_signals

2. Go to multi_signals subfolder

➢ npm init

name : multi_signals

description : multi-signals-node project

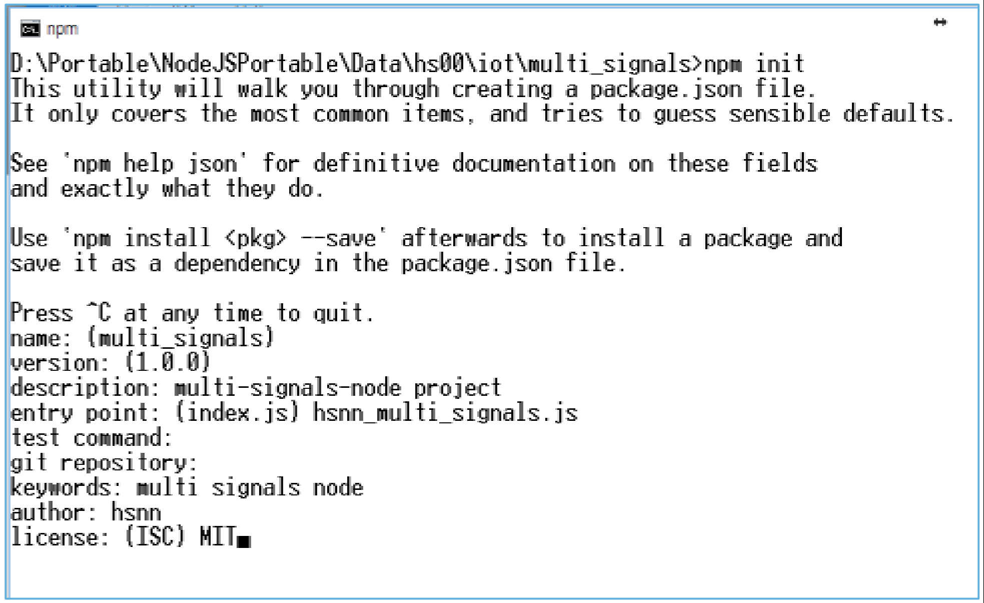
entry point : aann_multi_signals.js

author : aann

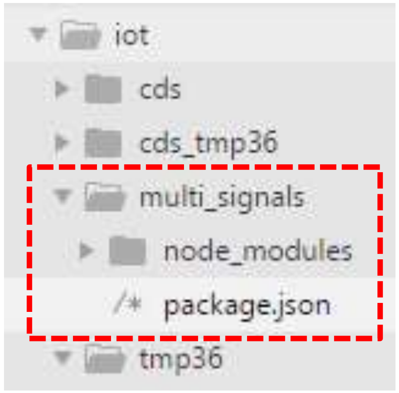
3. Install node modules

➢ npm install –save serialport@4.0.7

➢ npm install –save socket.io@1.7.3



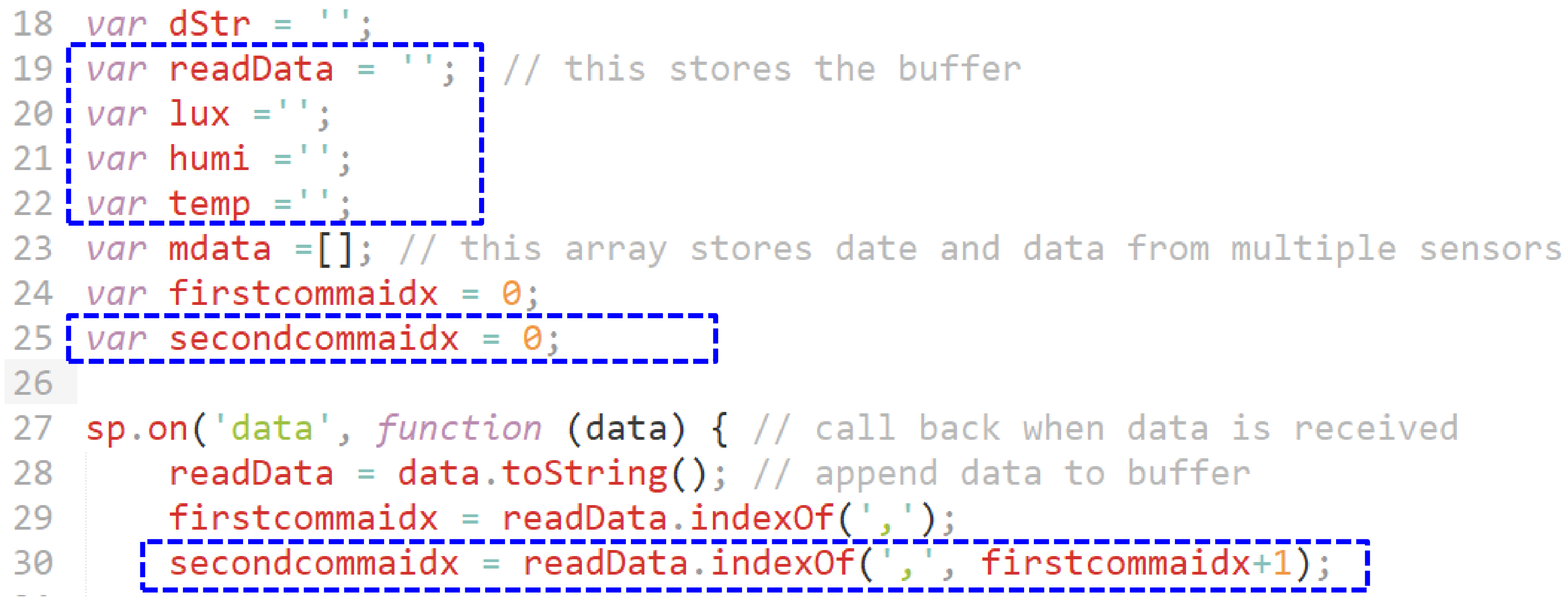
```
D:\Portable\NodeJSPortable\Data\hs00\iot\multi_signals>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

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

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

Press ^C at any time to quit.
name: (multi_signals)
version: (1.0.0)
description: multi-signals-node project
entry point: (index.js) hsnn_multi_signals.js
test command:
git repository:
keywords: multi signals node
author: hsnn
license: (ISC) MIT
```



DIY – New result 2-2

DIY 결과 [2] : 가상적인 세 개의 센서신호 시뮬레이션 → 조도,습도,온도를 **Node.js**로 처리

Recycling code:
Save cds_tmp36_node.js as
AAnn_multi_signals.js in multi_signals subfolder



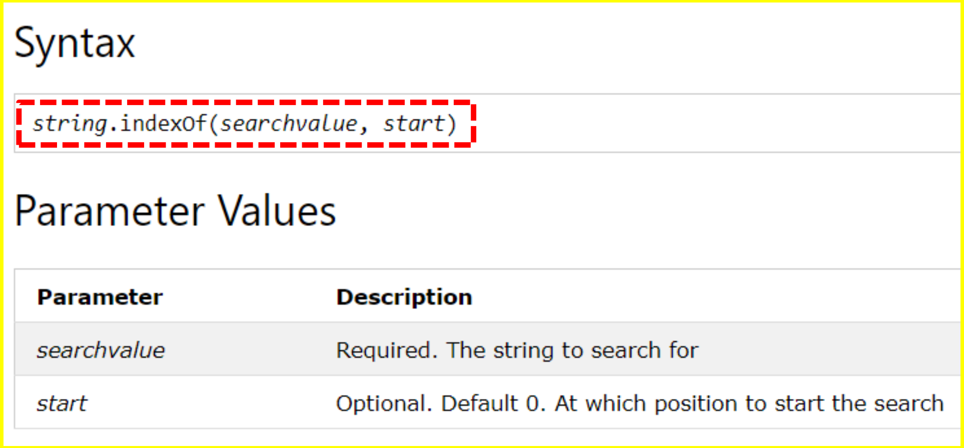
```
var dStr = '';
var readData = ''; // this stores the buffer
var lux ='';
var humi ='';
var temp ='';
var mdata =[]; // this array stores date and data from multiple sensors
var firstcommaidx = 0;
var secondcommaidx = 0;

sp.on('data', function (data) { // call back when data is received
    readData = data.toString(); // append data to buffer
    firstcommaidx = readData.indexOf(',');
    secondcommaidx = readData.indexOf(',', firstcommaidx+1);
```

DIY – New result 2-3

DIY 결과 [2] : 가상적인 세 개의 센서신호 시뮬레이션 → 조도,습도,온도를 Node.js로 처리

Hint:

javascript function : indexOf() https://www.w3schools.com/jsref/jsref_indexof.asp



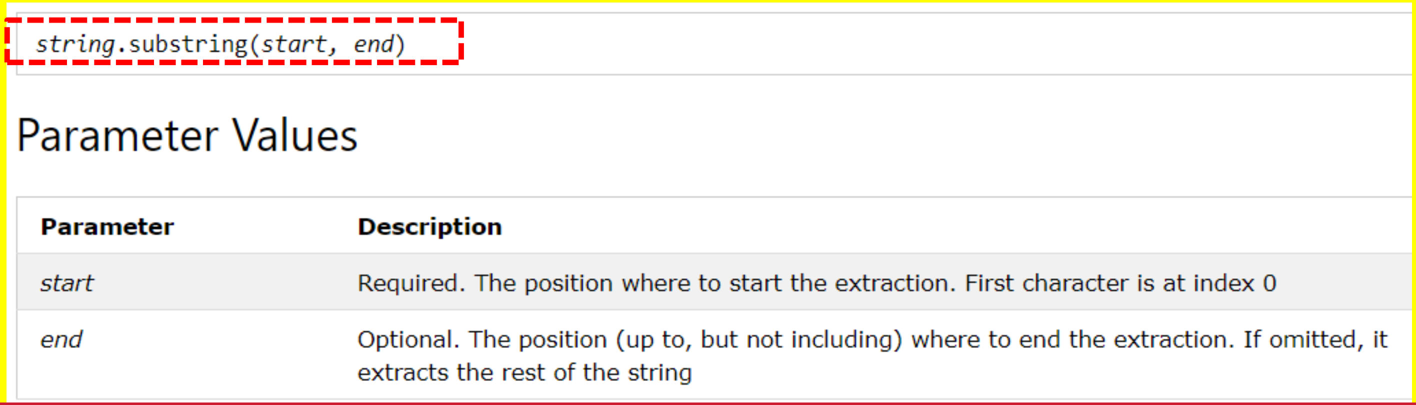
Syntax

```
string.indexOf(searchvalue, start)
```

Parameter Values

Parameter	Description
searchvalue	Required. The string to search for
start	Optional. Default 0. At which position to start the search

javascript function : substring()



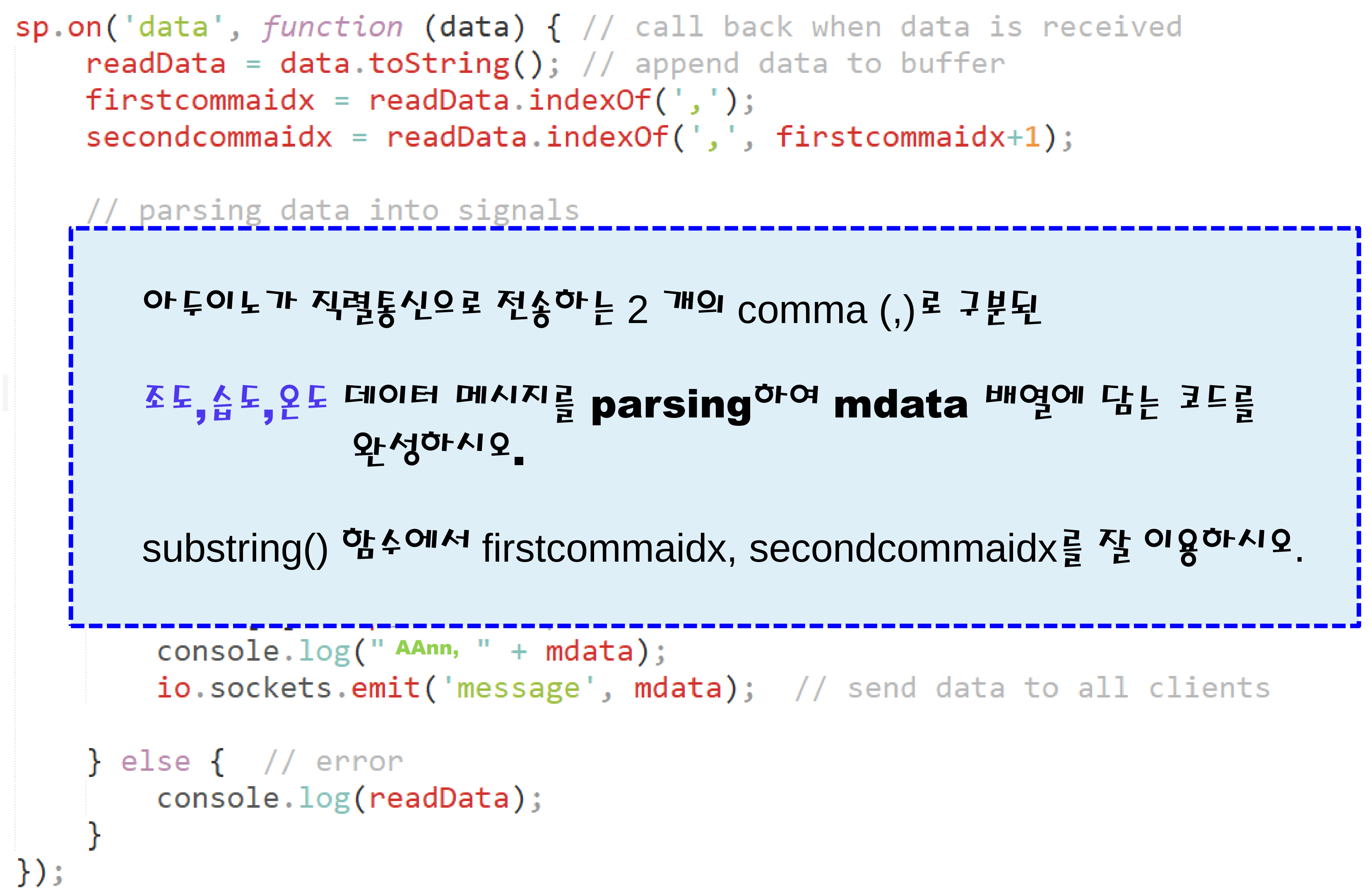
```
string.substring(start, end)
```

Parameter Values

Parameter	Description
start	Required. The position where to start the extraction. First character is at index 0
end	Optional. The position (up to, but not including) where to end the extraction. If omitted, it extracts the rest of the string

DIY – New result 2-4

DIY 결과 [2] : 가상적인 세 개의 센서신호 시뮬레이션 → **조도,습도,온도를 Node.js로 처리**



```
sp.on('data', function (data) { // call back when data is received
    readData = data.toString(); // append data to buffer
    firstcommaidx = readData.indexOf(',');
    secondcommaidx = readData.indexOf(',', firstcommaidx+1);

    // parsing data into signals
```

아두이노가 직렬통신으로 전송하는 2 개의 comma (,)로 구분된

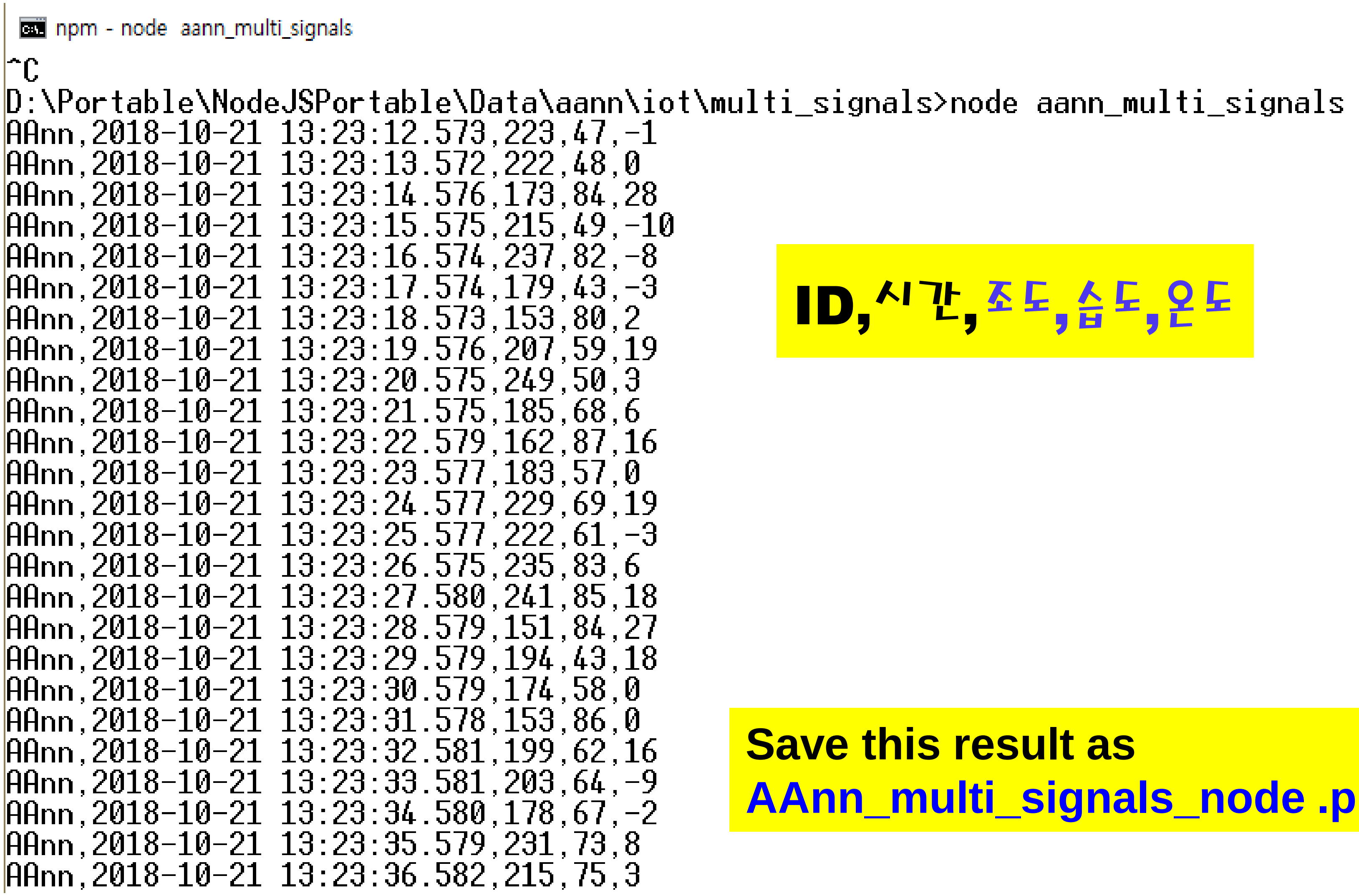
조도,습도,온도 데이터 메시지를 **parsing**하여 **mdata** 배열에 담는 코드를 완성하시오.

substring() 함수에서 firstcommaidx, secondcommaidx를 잘 이용하시오.

```
        console.log(" AAnn, " + mdata);
        io.sockets.emit('message', mdata);  // send data to all clients

    } else {  // error
        console.log(readData);
    }
});
```

DIY – New result 2-5

DIY 결과 [2] : 가상적인 세 개의 센서신호 시뮬레이션 → 조도,습도,온도를 Node.js로 처리



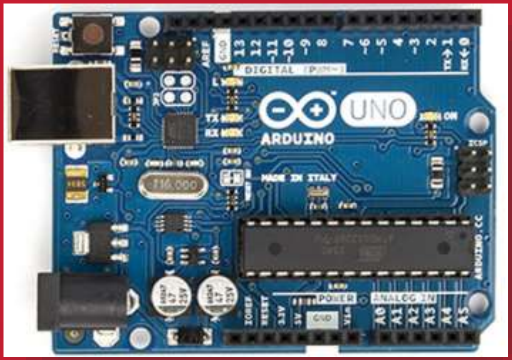
```
npm - node  aann_multi_signals
^C
D:\Portable\NodeJSPortable\Data\aann\iot\multi_signals>node aann_multi_signals
AAnn,2018-10-21 13:23:12.573,223,47,-1
AAnn,2018-10-21 13:23:13.572,222,48,0
AAnn,2018-10-21 13:23:14.576,173,84,28
AAnn,2018-10-21 13:23:15.575,215,49,-10
AAnn,2018-10-21 13:23:16.574,237,82,-8
AAnn,2018-10-21 13:23:17.574,179,43,-3
AAnn,2018-10-21 13:23:18.573,153,80,2
AAnn,2018-10-21 13:23:19.576,207,59,19
AAnn,2018-10-21 13:23:20.575,249,50,3
AAnn,2018-10-21 13:23:21.575,185,68,6
AAnn,2018-10-21 13:23:22.579,162,87,16
AAnn,2018-10-21 13:23:23.577,183,57,0
AAnn,2018-10-21 13:23:24.577,229,69,19
AAnn,2018-10-21 13:23:25.577,222,61,-3
AAnn,2018-10-21 13:23:26.575,235,83,6
AAnn,2018-10-21 13:23:27.580,241,85,18
AAnn,2018-10-21 13:23:28.579,151,84,27
AAnn,2018-10-21 13:23:29.579,194,43,18
AAnn,2018-10-21 13:23:30.579,174,58,0
AAnn,2018-10-21 13:23:31.578,153,86,0
AAnn,2018-10-21 13:23:32.581,199,62,16
AAnn,2018-10-21 13:23:33.581,203,64,-9
AAnn,2018-10-21 13:23:34.580,178,67,-2
AAnn,2018-10-21 13:23:35.579,231,73,8
AAnn,2018-10-21 13:23:36.582,215,75,3
```

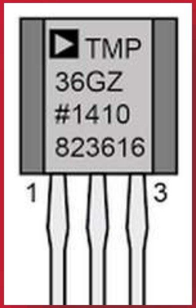
ID,시간,조도,습도,온도

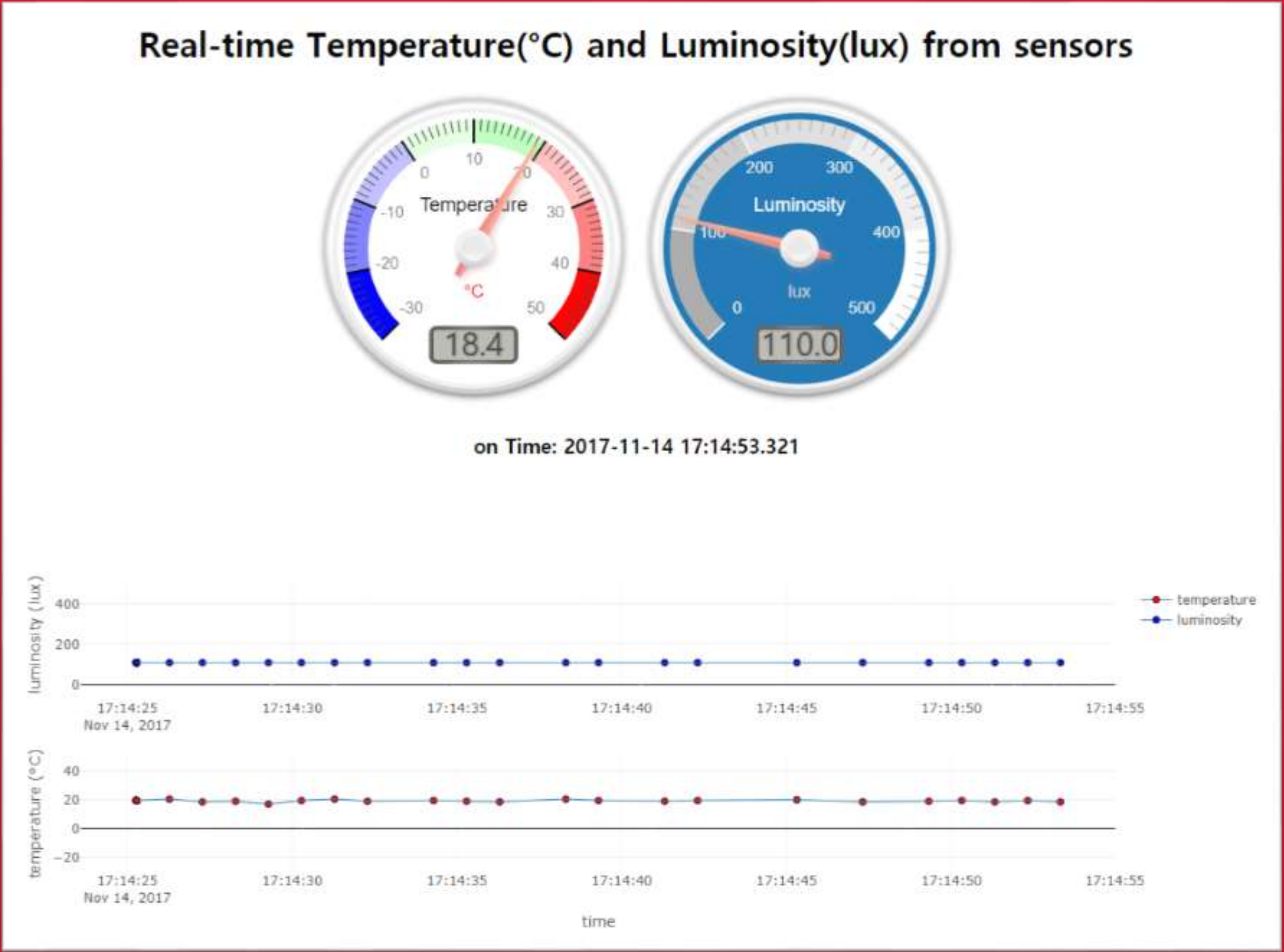
Save this result as
AAnn_multi_signals_node .png

Next week

Data visualization using ploy.ly

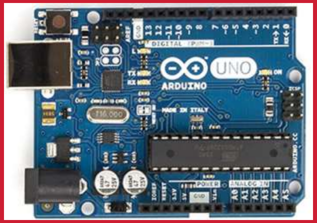






Real-time Temperature(°C) and Luminosity(lux) from sensors

on Time: 2017-11-14 17:14:53.321



[Practice]

- [wk09]
  - Arduino + Node.js I. sensors
  - Complete your project
  - Submit file : AAnn_Rpt05.zip

wk09 : Practice : AAnn_Rpt05.zip

[Target of this week]

- Complete your works
- **Save your outcomes and compress 5 outputs**

제출파일명 : **AAnn_Rpt05.zip**

- 압축할 파일들

① **AAnn_tmp36_message.png**

② **AAnn_tmp36_IOT_data.png**

③ **AAnn_cds_IOT_data.png**

④ **AAnn_cds_tmp36_lcd.png**

⑤ **AAnn_cds_tmp36_IOT.png**

⑥ **AAnn_multi_signals_node.png**

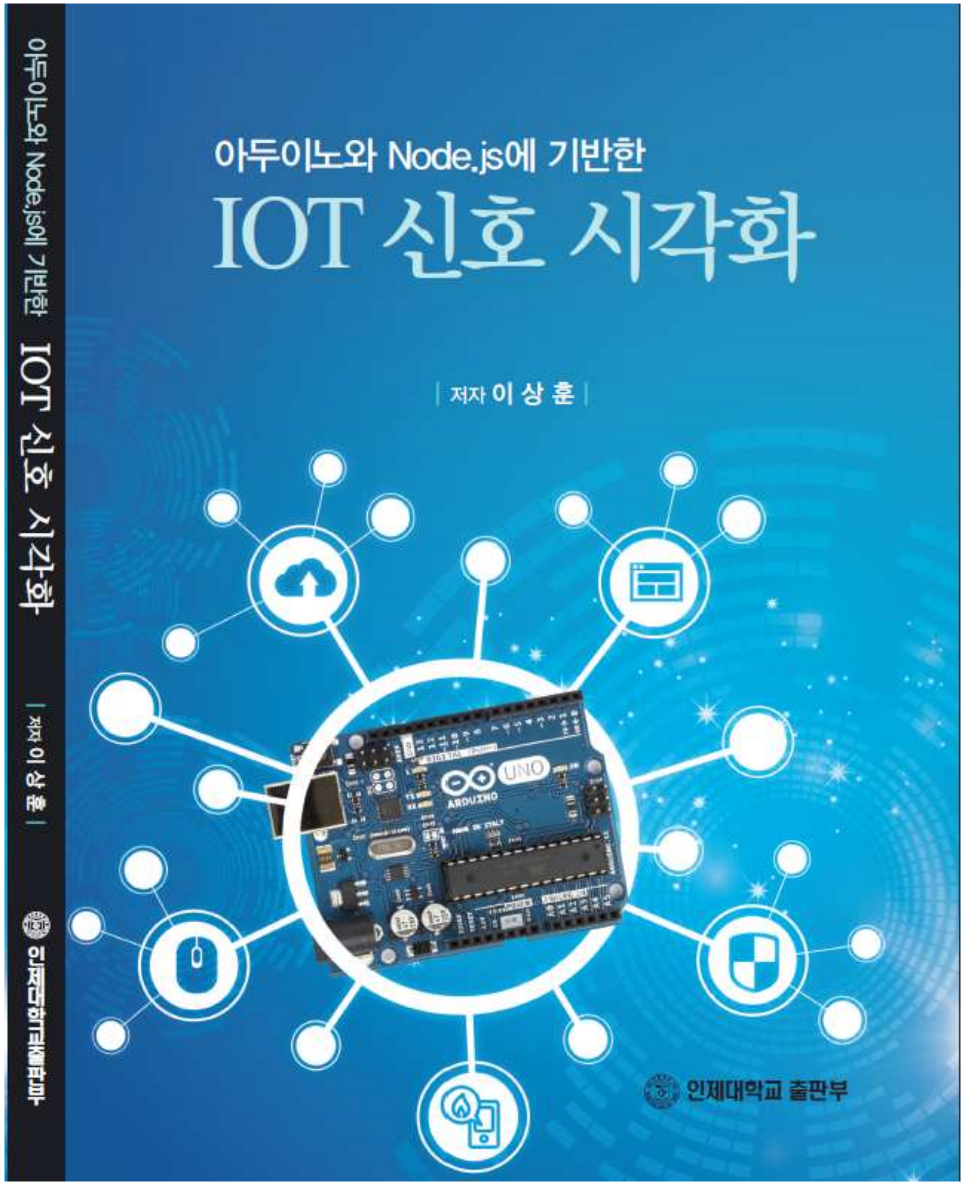
[제목 **: id,** 이름 **(**수정**)]**

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

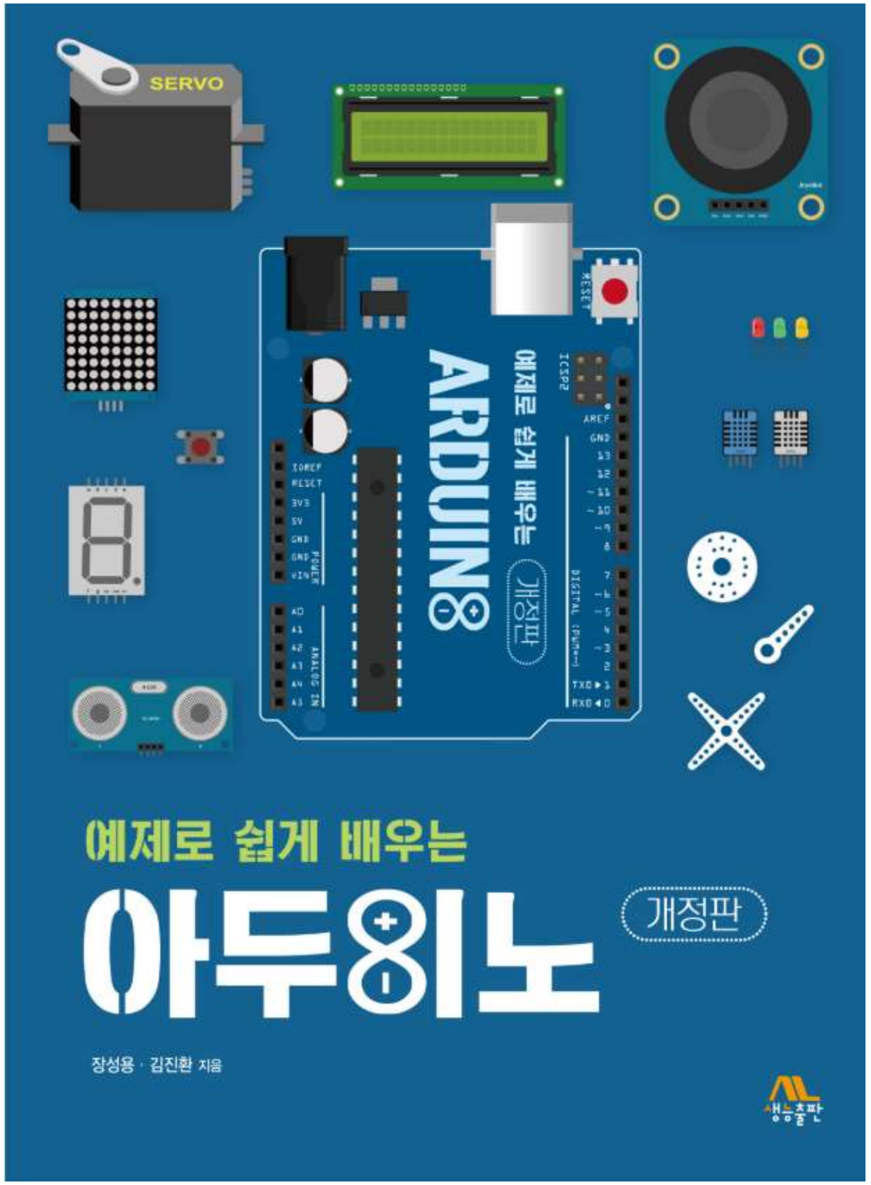


아두이노와 Node.js에 기반한

IOT 신호 시각화

| 저자 이 상 훈 |

인제대학교 출판부

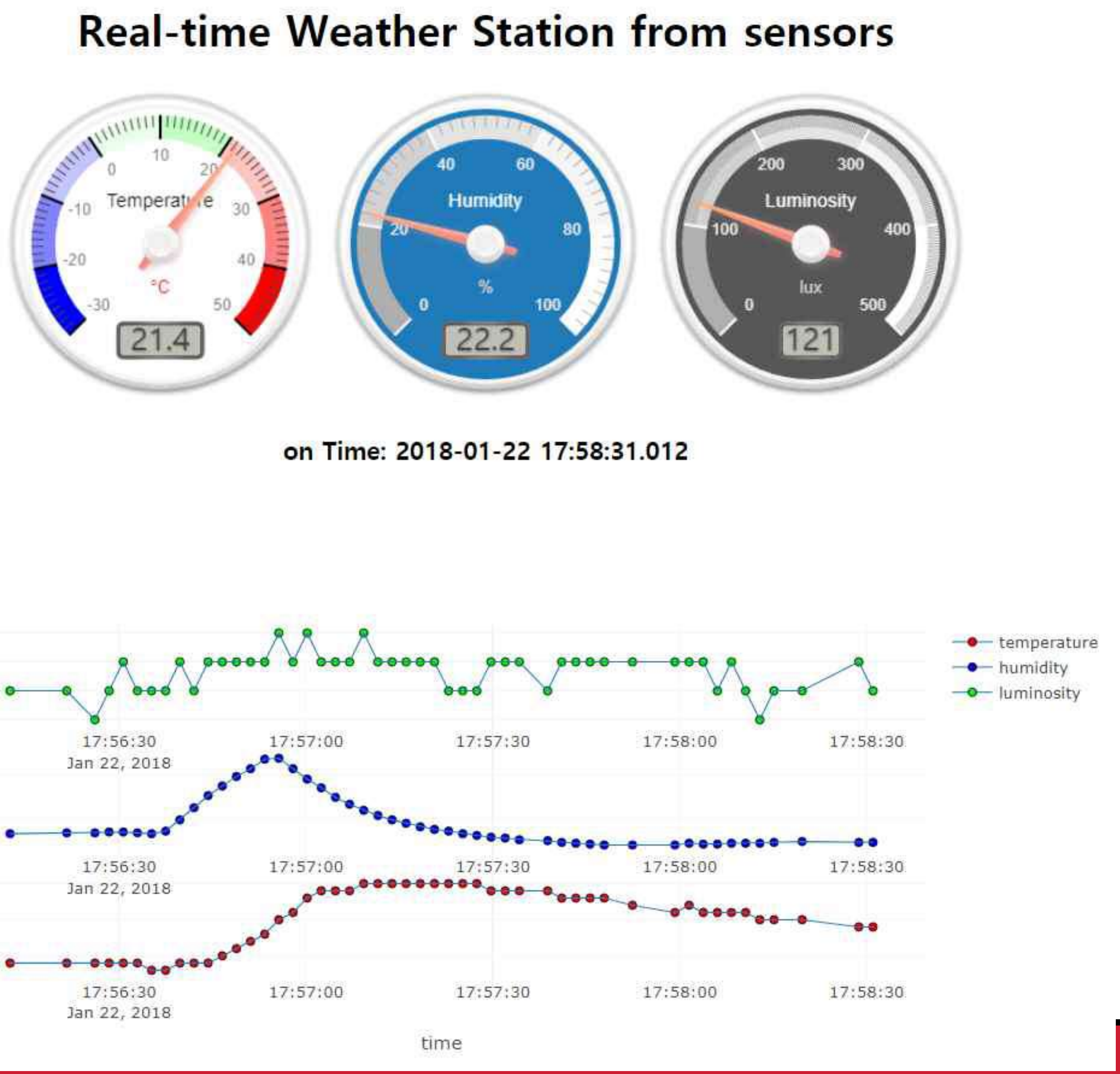


예제로 쉽게 배우는

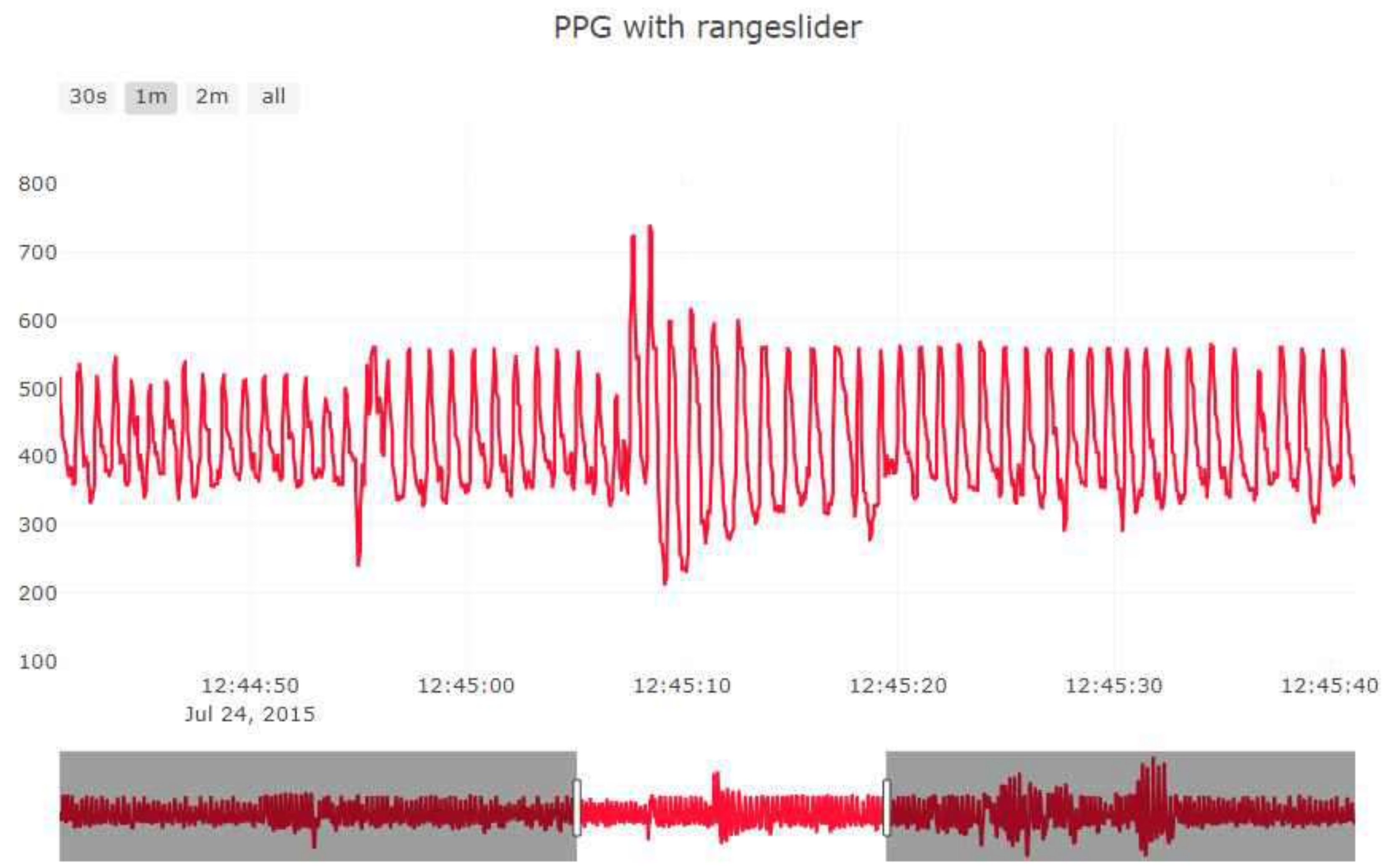
아두이노 개정판

장성용 · 김진환 지음

Target of this class



Another target of this class


PPG with rangeslider
30s
1m
2m
all
12:44:50
Jul 24, 2015
12:45:00
12:45:10
12:45:20
12:45:30
12:45:40