

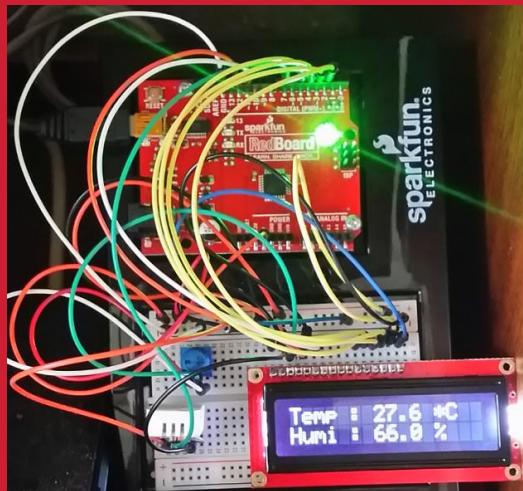


HW-SW-Connectivity

[wk11]

Arduino & NodeJS II

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

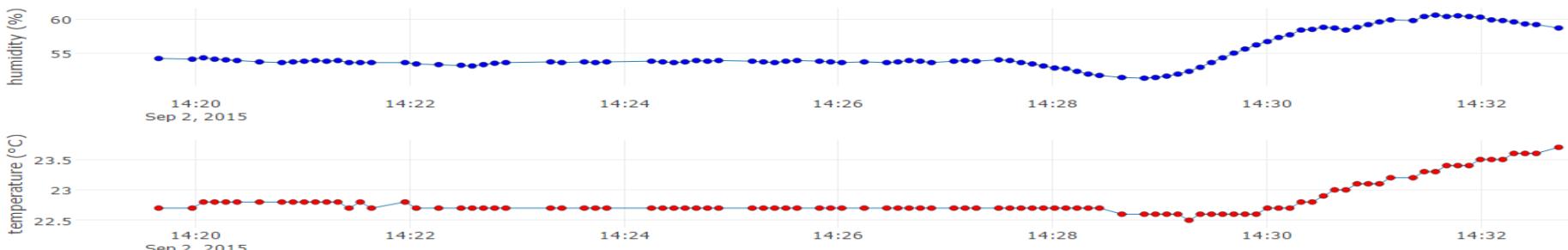


Basic HW and SW Integration using
Arduino & Javascript

COMSI, INJE University

2nd semester, 2017

Email : yish@inje.ac.kr





[Practice]

◆ [wk10]

- **Arduino sensors & node.js**
- **Complete your TMP36 project**
- **Upload file name : AAnn_Rpt07.zip**

wk10 : Practice-07 : AAnn_Rpt07.zip

◆ [Target of this week]

- Complete your projects
- Save your outcomes and compress one ino file and 2 figures

제출파일명 : **AAnn_Rpt07.zip**

- 압축할 파일들

- ① **AAnn_random_PWM.ino**
- ② **AAnn_tmp36_message.png**
- ③ **AAnn_tmp36_IOT_data.png**

Email : **chaos21c@gmail.com**

wk10 : Practice-07 : AAnn_Rpt07.zip

AA07_random_pwm

```
1
2 float pwm1 = 0;
3 float pwm2 = 0;
4 float pwm3 = 0;
5
6 int ledR = 3;
7 int ledG = 5;
8 int ledB = 6;
9
10 void setup() {
11   // put your setup code here, to run once:
12   Serial.begin(9600);
13 }
```

```
15 void loop() {
16   // put your main code here:
17   pwm1 = random(0,255);
18   pwm2 = random(0 ,255);
19   pwm3 = random(0,255);
20   pwmLed(ledR , pwm1);
21   pwmLed(ledG , pwm2);
22   pwmLed(ledB , pwm3);
23
24   Serial.print("AA07, LED_R: ");
25   Serial.print(pwm1);
26   Serial.print(" , LED_G: ");
27   Serial.print(pwm2);
28   Serial.print(" , LED_B: ");
29   Serial.println(pwm3);
30   delay(10);
31 }
32
33 void pwmLed(int led, int pwmValue) {
34   analogWrite(led, pwmValue);
35   delay(10);
36 }
```



Arduino

[Home](#)[Buy](#)[Download](#)[Products](#) ▾[Learning](#) ▾[Forum](#)[Support](#) ▾[Blog](#)

<https://www.arduino.cc/>

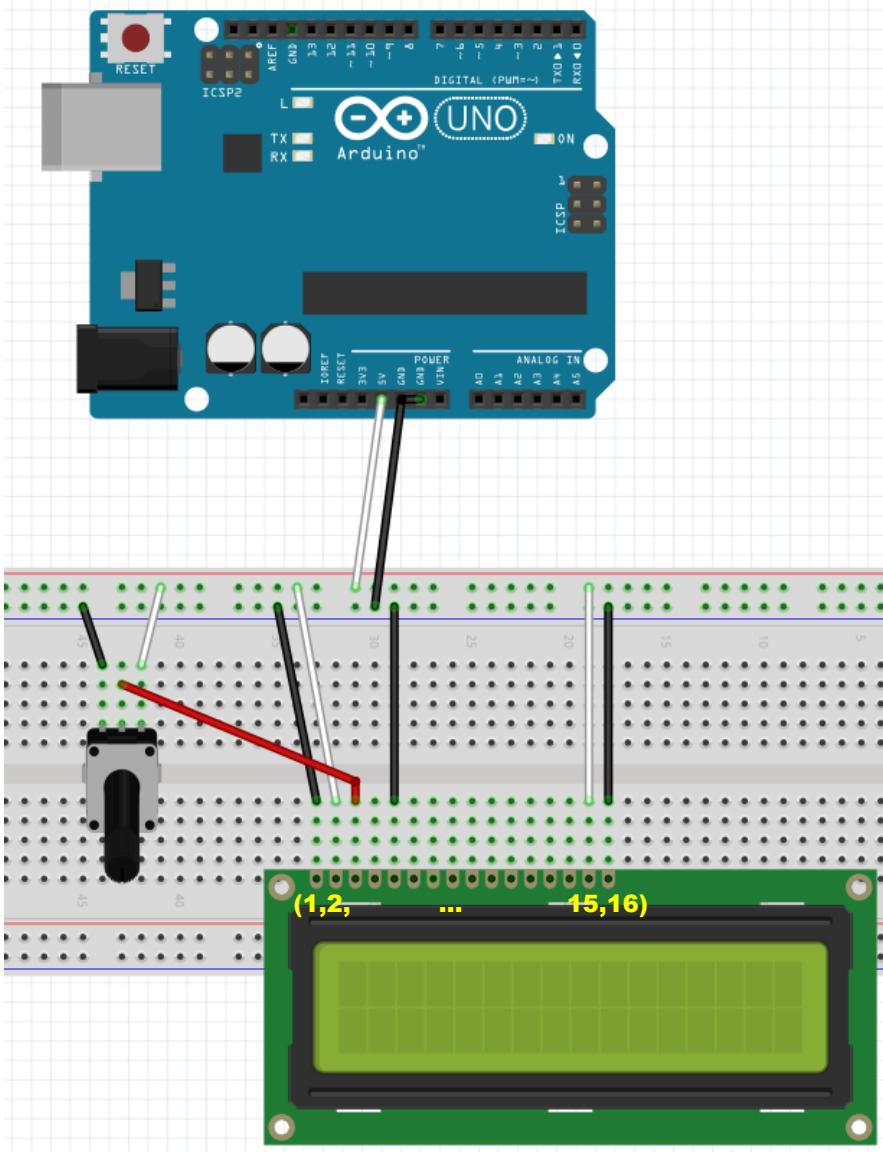


Introduction to LCD



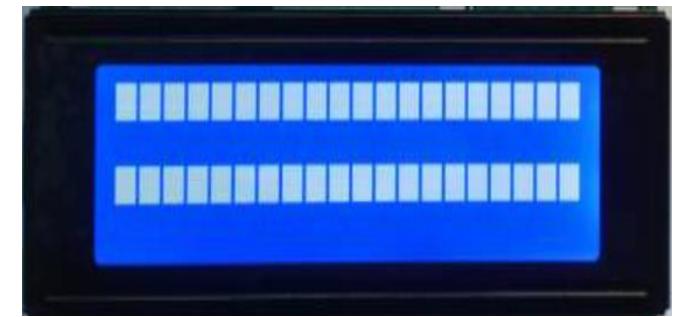


LCD 초기화 (pin-1, 2, 3, 5, 15,16)



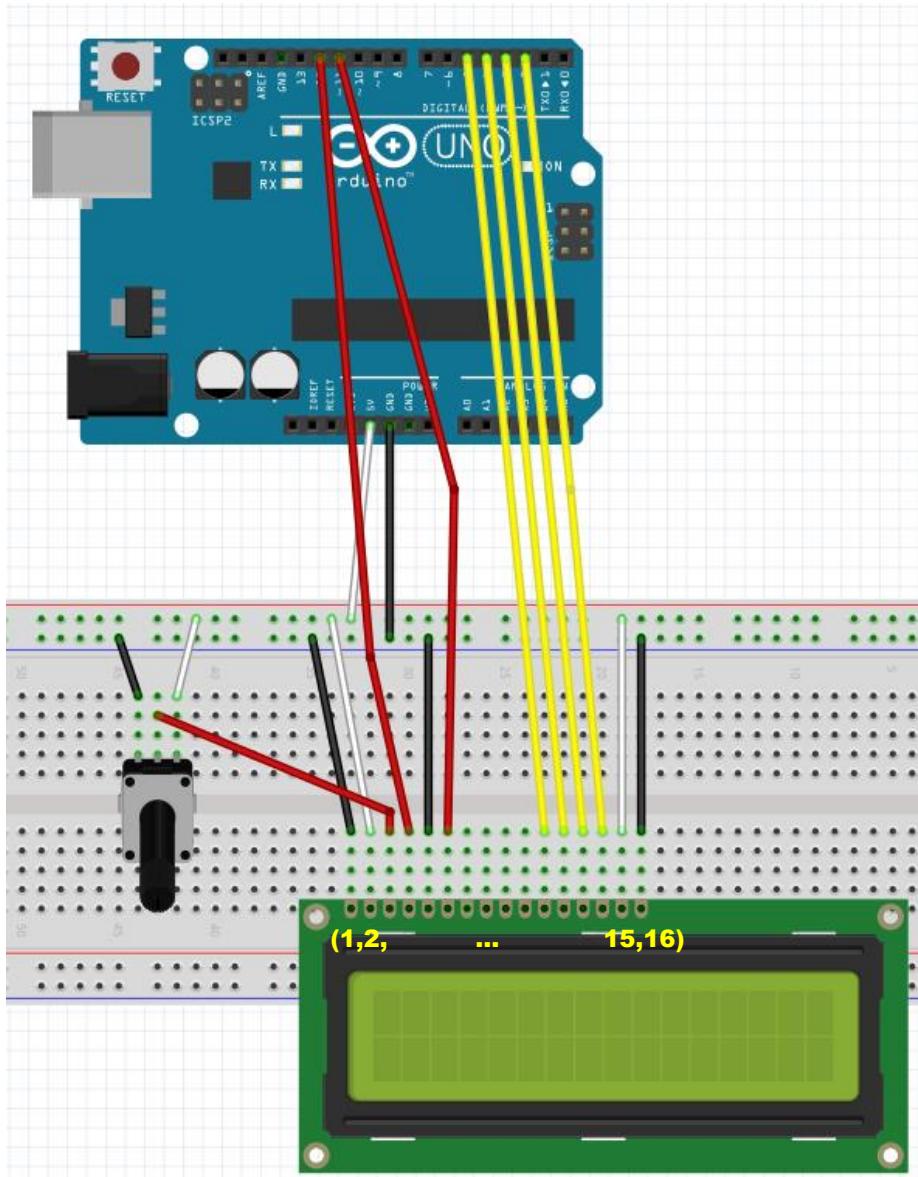
Pin 1 to Arduino GND
Pin 2 to Arduino +5V
Pin 3 to wiper (potentiometer)
Pin 5 to Arduino GND
Pin 15 to +5V
Pin 16 to GND

전원 연결 후
LCD 초기화





데이터 입력 초기화 (pin-4, 6, 11,12,13,14)



Pin 1 to Arduino GND

Pin 2 to Arduino 5V

Pin 3 to wiper

Pin 4 to Arduino pin D12

Pin 5 to Arduino GND

Pin 6 to Arduino pin D11

Pin 11 to Arduino pin D5

Pin 12 to Arduino pin D4

Pin 13 to Arduino pin D3

Pin 14 to Arduino pin D2

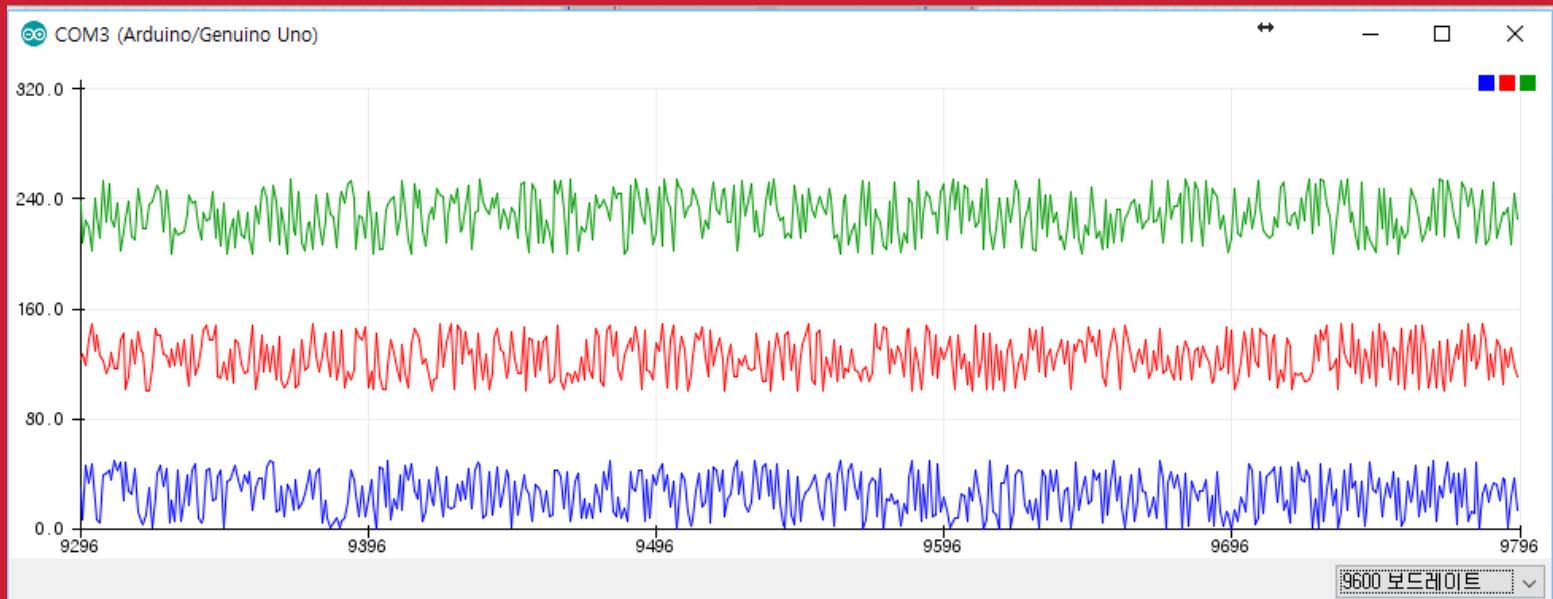
Pin 15 to +5V

Pin 16 to GND



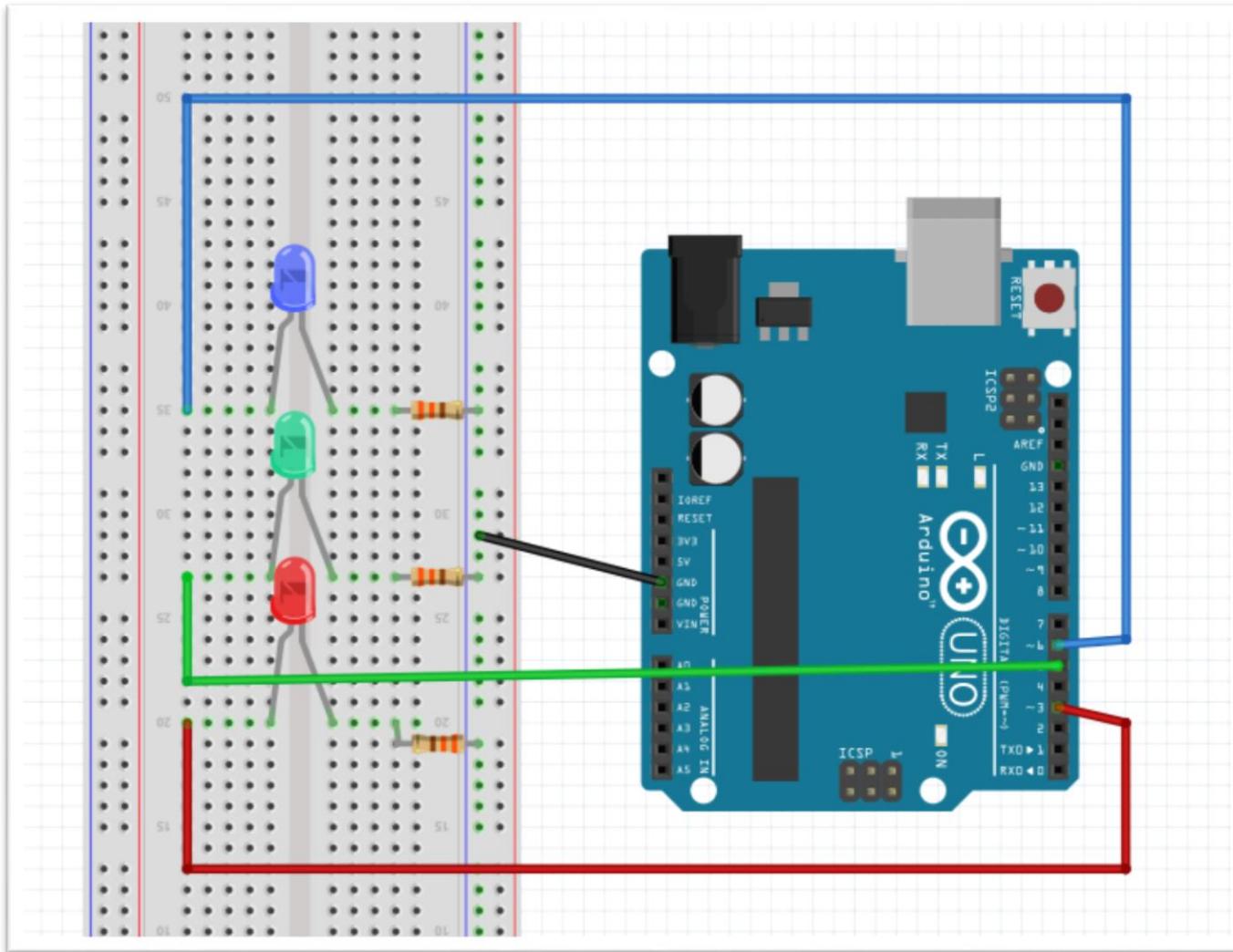
Arduino

Random Data Simulation



A4.2.1 [DIY] Random PWM simulation

2.1 Multiple random signals – Circuit of 3 LEDs at PWM pins (3,5,6)





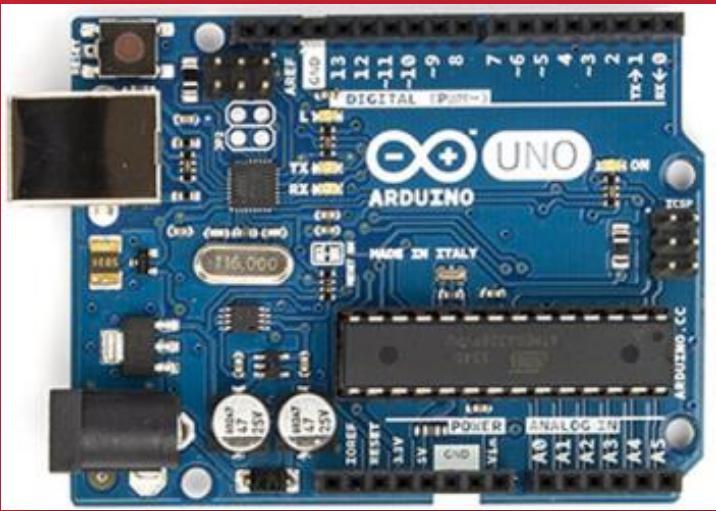
A4.2.2 [DIY] Random PWM simulation

2.2 Multiple random signals – Randomly changing the brightness of 3 led's

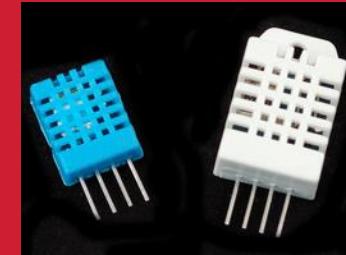
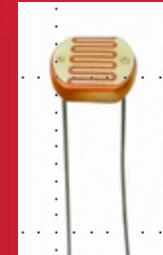
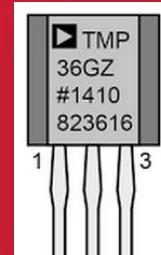
```
AA00_Three_random_pwms
1 /*
2 Multi Signals
3
4 Simulation of multiple random signals to control three LED's
5 */
6
7 float pwm1=0; // Blue LED
8 float pwm2=0; // Red LED
9 float pwm3=0; // Green LED
10
11 int ledR = 3;
12 int ledG = 5;
13 int ledB = 6;
14
15 // the setup routine runs once when you press reset:
16 void setup() {
17   // initialize serial communication at 9600 bits per second:
18   Serial.begin(9600);
19 }
```

```
21 // the loop routine runs over and over again forever:
22 void loop() {
23
24   // Multi signals
25   pwm1 = random(0,50);
26   pwm2 = random(100, 150);
27   pwm3 = random(200,255);
28   Serial.print("AA00, LED_B: ");
29   Serial.print(pwm1);
30   Serial.print(" , LED_R: ");
31   Serial.print(pwm2);
32   Serial.print(" , LED_G: ");
33   Serial.println(pwm3);
34   delay(100);           // delay in between reads for stability
35 }
```

[DIY] 무작위 수인 세 개의 **pwm** 값을 이용해서 **R, G ,B led**의 밝기를 무작위로 변경하는 아두이노 코드를 작성하시오. (5주차 **dimming code**를 참조하시오)
완성된 스케치 **code**를 **AAnn_random_PWM.ino** 로 저장해서 제출.

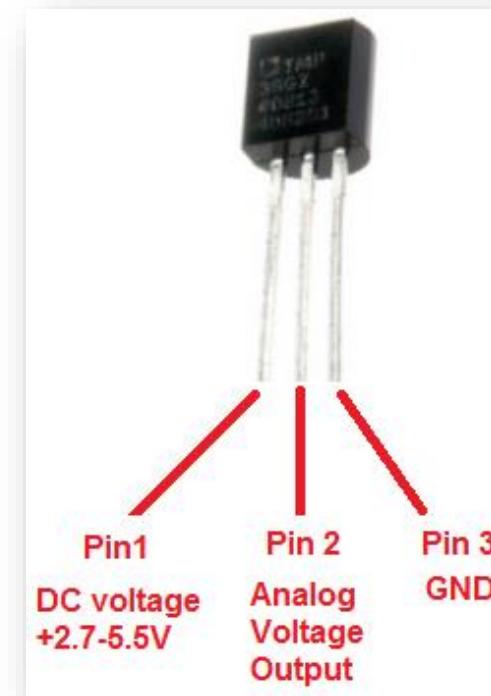
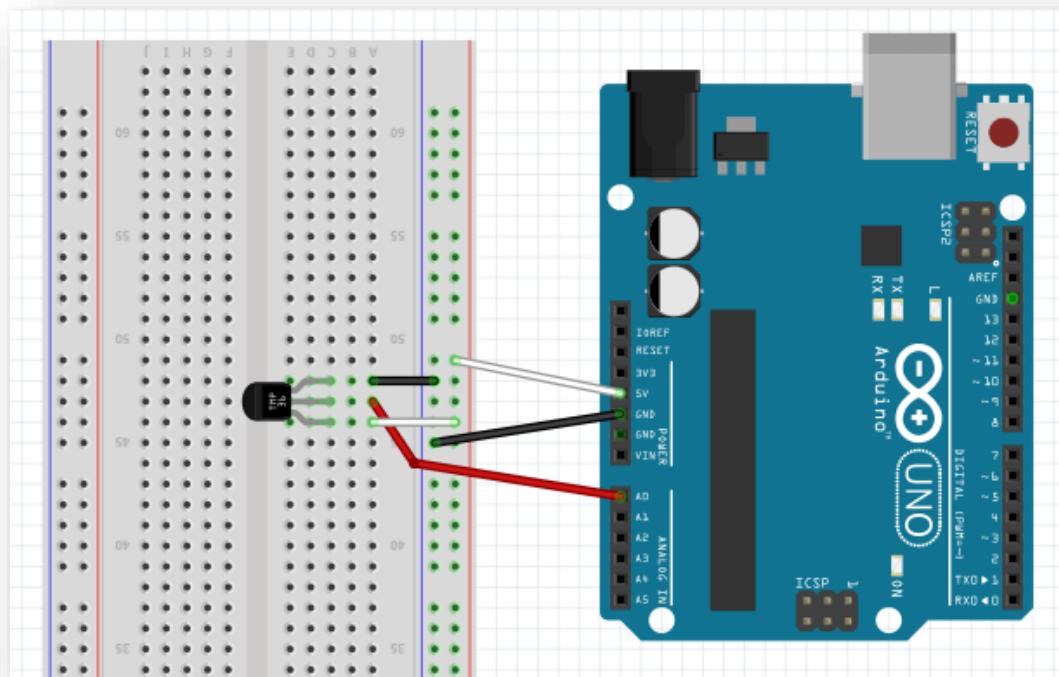


Arduino & Node.js

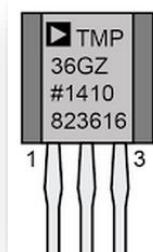




A3.1.2 Temperature sensor [TMP36]



Parts : TMP36



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



A3.1.5 Temperature sensor [TMP36]

Working code

```
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 }
```

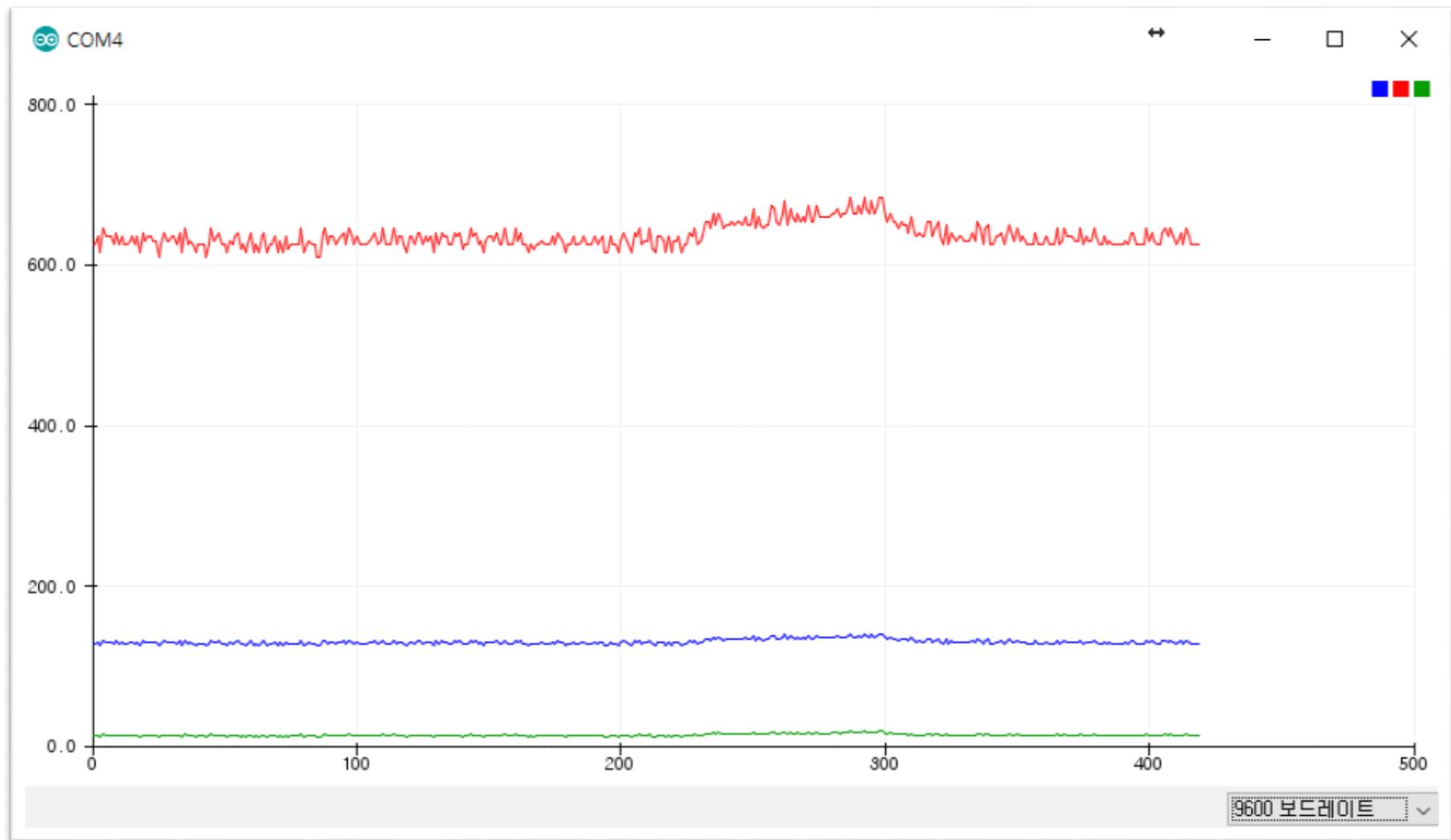
Serial output (°C)

The screenshot shows the Arduino Serial Monitor window titled "COM4". It displays a series of temperature readings printed in a repeating loop. Each reading consists of the string "AA00, value = ", followed by a numerical value, and then "mV, " and "degrees C". The values fluctuate between approximately 12.56°C and 14.52°C.

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



A3.1.5 Temperature sensor [TMP36]





A5.1.2 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

```
cmd NodeJS
D:\Portable\NodeJSPortable\Data\aa00\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 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: (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
About to write to D:\Portable\NodeJSPortable\Data\aa00\tmp36\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"
}

Is this ok? (yes)
D:\Portable\NodeJSPortable\Data\aa00\tmp36>
```



A5.1.3 tmp36 node project

package.json

```
▶ package.json ×
1 {
2   "name": "tmp36",
3   "version": "1.0.0",
4   "description": "tmp36-node project",
5   "main": "tmp36_node.js",
6   "scripts": {
7     "test": "echo \\\"Error: no test specified\\\" && exit 1"
8   },
9   "keywords": [
10     "tmp36",
11     "node",
12     "arduino"
13   ],
14   "author": "aa00",
15   "license": "MIT"
16 }
```



A5.1.5 tmp36 node project

Go to tmp36 subfolder

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

```
1 {  
2   "name": "tmp36",  
3   "version": "1.0.0",  
4   "description": "tmp36-node project",  
5   "main": "tmp36_node.js",  
6   "scripts": {  
7     "test": "echo \\\"Error: no test specified\\\" && exit 1"  
8   },  
9   "keywords": [  
10     "tmp36",  
11     "node",  
12     "arduino"  
13   ],  
14   "author": "aa00",  
15   "license": "MIT",  
16   "dependencies": {  
17     "serialport": "^6.0.4",  
18     "socket.io": "^2.0.4"  
19   }  
20 }  
21
```



A5.1.6 tmp36 node project : code-1

tmp36_node.js

```
1 // tmp36_node.js
2
3 var serialport = require('serialport');
4 var portName = 'COM4'; // check your COM port!!
5 var port      = process.env.PORT || 3000;
6
7 var io = require('socket.io').listen(port);
8
9 // serial port object
10 var sp = new serialport(portName,{
11     baudRate: 9600, // 9600 38400
12     dataBits: 8,
13     parity: 'none',
14     stopBits: 1,
15     flowControl: false,
16     parser: serialport.parsers.readline('\r\n') // new serialport.parsers.ReadLine
17 });
18
19 var tdata = 0;
20
21 sp.on('data', function (data) { // call back when data is received
22     // raw data only
23     //console.log(data);
24     tdata=data; // data
25     console.log(tdata);
26     io.sockets.emit('message', tdata); // send data to all clients
27 });
```

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



A5.1.7 tmp36 node project : code-2

tmp36_node.js

```
28
29 io.sockets.on('connection', function (socket) {
30     // If socket.io receives message from the client browser then
31     // this call back will be executed.
32     socket.on('message', function (msg) {
33         console.log(msg);
34     });
35     // If a web browser disconnects from Socket.IO then this callback is called.
36     socket.on('disconnect', function () {
37         console.log('disconnected');
38     });
39 });
40
```

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 ...



A5.1.6A tmp36 node project → downgrade

Go to tmp36 subfolder

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

```
1 {  
2   "name": "tmp36",  
3   "version": "1.0.0",  
4   "description": "tmp36-node project",  
5   "main": "tmp36_node.js",  
6   "scripts": {  
7     "test": "echo \\\"Error: no test specified\\\" && exit 1"  
8   },  
9   "keywords": [  
10     "tmp36",  
11     "node",  
12     "arduino"  
13   ],  
14   "author": "aa00",  
15   "license": "MIT",  
16   "dependencies": {  
17     "serialport": "^6.0.4",  
18     "socket.io": "^2.0.4"  
19   }  
20 }  
21
```

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

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



A5.1.11 tmp36 node project (date & data → IOT)

tmp36_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(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;
}
```

Serial output (°C)

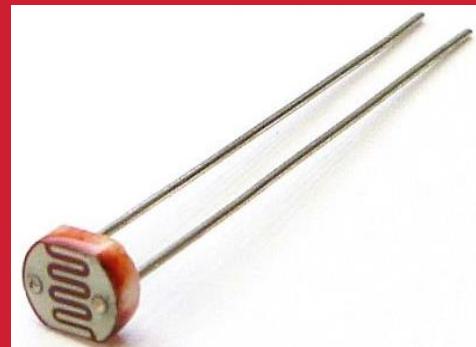
IOT data format

```
[ '2017-11-01 12:46:20.033', '15.49' ]
[ '2017-11-01 12:46:21.042', '15.49' ]
[ '2017-11-01 12:46:22.034', '13.54' ]
[ '2017-11-01 12:46:23.026', '14.03' ]
[ '2017-11-01 12:46:24.035', '15.00' ]
[ '2017-11-01 12:46:25.027', '14.52' ]
[ '2017-11-01 12:46:26.035', '16.47' ]
[ '2017-11-01 12:46:27.028', '15.98' ]
[ '2017-11-01 12:46:28.020', '15.98' ]
[ '2017-11-01 12:46:29.028', '15.49' ]
[ '2017-11-01 12:46:30.021', '13.05' ]
[ '2017-11-01 12:46:31.013', '15.49' ]
[ '2017-11-01 12:46:32.021', '15.00' ]
```

AAnn_tmp36_IOT_data.png
로 저장

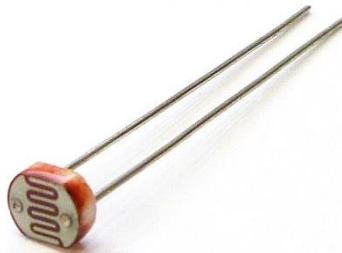


Arduino & Node.js



A3.2.1 Luminosity sensor [Photocell LDR]

CdS 센서 - photoresistor



- ✓ CdS 분말을 세라믹 기판 위에 압축하여 제작
- ✓ 빛이 강할 수록 저항 값이 감소
- ✓ ADC를 이용하여 변화된 저항에 전압을 인가하여 전압의 변화를 감지
- ✓ 자동 조명장치, 조도 측정 등에 사용

럭스

다른 뜻에 대해서는 Lux 문서를 참조하십시오.

럭스(lux, 기호 lx)는 빛의 조명도를 나타내는 SI 단위이다. 럭스는 루멘에서 유도
 $1 \text{ lx} = 1 \text{ lm/m}^2 = 1 \text{ cd}\cdot\text{sr}\cdot\text{m}^{-2}$

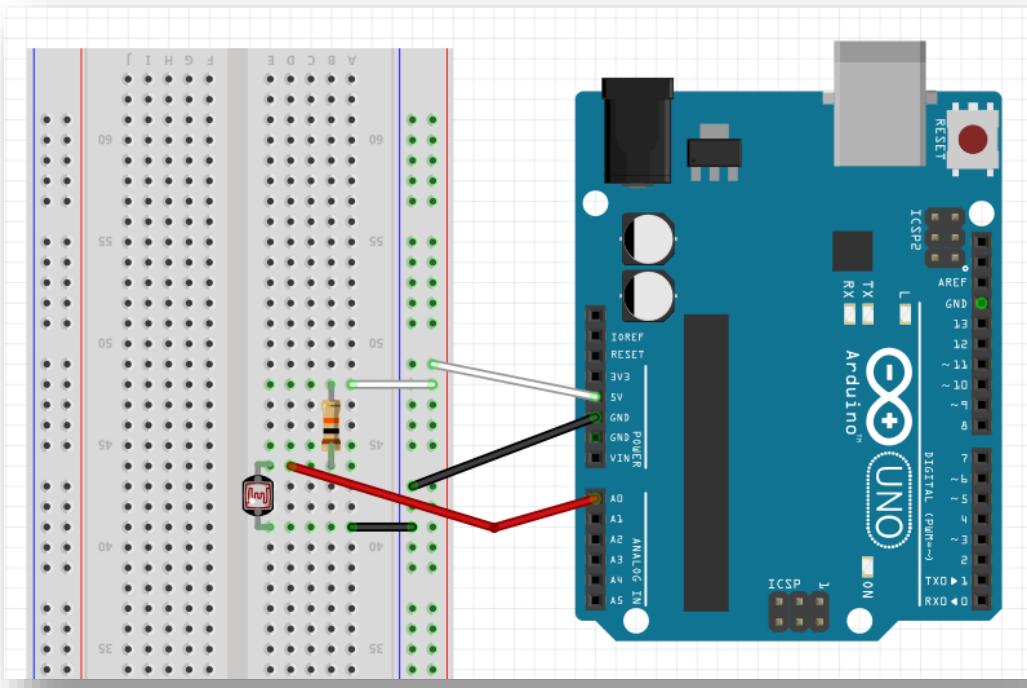
럭스의 예 [편집]

I밝기자	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	절대 위도를 덮은 보름달 ^[3]
3.4 lux	맑은 하늘 아래의 어두운 황혼 ^[4]
50 lux	거실 ^[5]
80 lux	북도/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



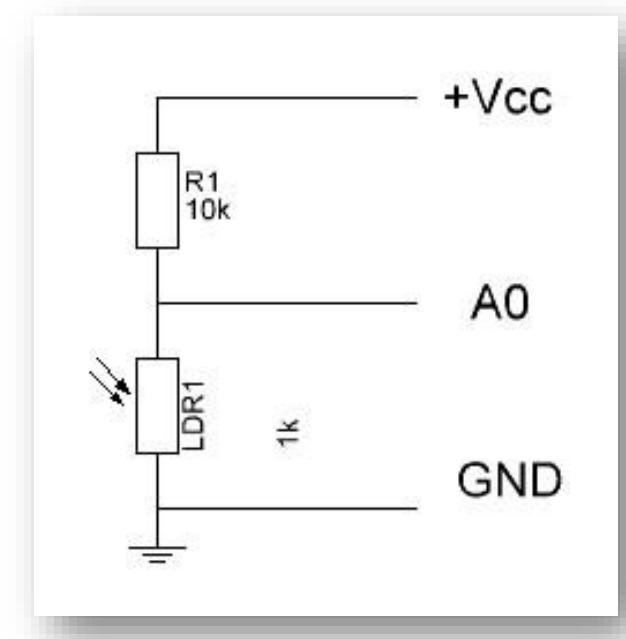
A3.2.2 Luminosity sensor [Photocell LDR]

CdS 센서 회로



Parts : 20 mm photocell LDR, R ($10\text{ k}\Omega \times 1$)

광센서에서의 전압 강하 값을 A0로 측정





A3.2.4 Luminosity sensor [Photocell LDR]

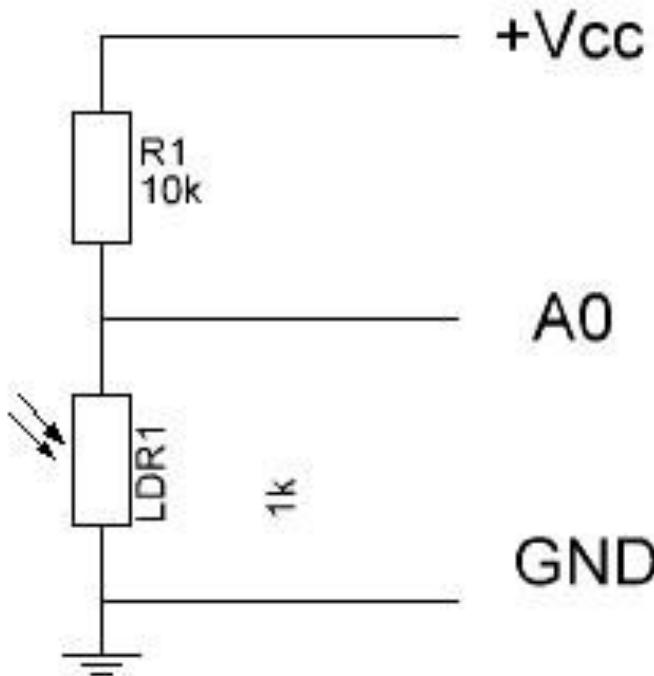
CdS 센서 회로 - 측정 1.

```
AAnn_CdS
1 #define CDS_INPUT 0
2
3 void setup() {
4   Serial.begin(9600);
5 }
6
7 void loop() {
8
9   int value = analogRead(CDS_INPUT);
10  Serial.println(value);
11
12  delay(1000);
13 }
14
```

```
COM4
|
233
234
235
237
235
235
235
236
241
386
975
965
964
964
967
```

어두으면 측정 값이 커지고 밝을수록 값이 작아진다 ???

CdS 센서 회로 분석 (1/2)



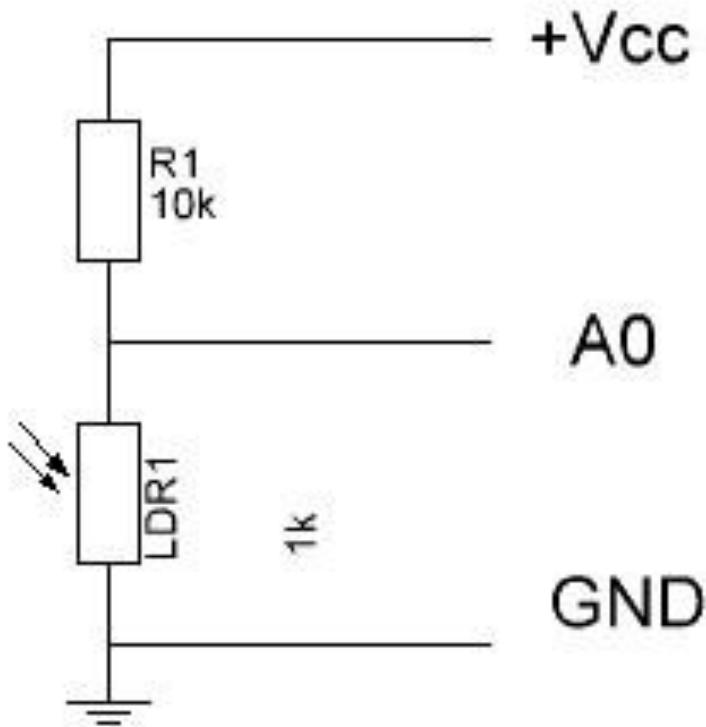
LDR's (Light dependent resistors) have a low resistance in bright light and a high resistance in the darkness.

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}



$$A_o \rightarrow V_o \rightarrow lux$$

$$lux = 500 / R_{ldr}$$

$$V_o = I_{ldr} * R_{ldr}$$

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

$$R_{ldr} = 10 * V_o / (5 - V_o)$$

$$lux = 250/V_o - 50$$

$$V_o = 5.0 * A_o / 1023.0$$

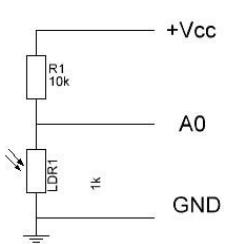
CdS 센서 회로 분석 (3/3)

**Doing that on an Arduino Analog port, would give a reading between 0 and 1024, which of course are really non-descriptive numbers.
What you would want is an output in Lux or Lumen?**

$$R_{ldr} = 500/\text{Lux}, \text{ or}$$

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

as R_{ldr} is related to the voltage measured over it, reading the Voltage over it, can be used to calculate the R_{ldr} and thus the Lux level



If the LDR is the lower part of a 5 Volt Voltage divider and a 10kOhm resistor the upper part, the Voltage will be:

$$V_{out} = (5/(10+R_{ldr})) * R_{ldr}$$

$$V_{out} = 5 * R_{ldr} / (10+R_{ldr}) \text{ (remember: multiplication before division)}$$

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

$$5/1024 = 0.0048828125.$$

$$\Rightarrow V_{out} = \text{Analogreading} * 0.0048828125$$

$$\text{as } R_{ldr} = (10V_{out}) / (5 - V_{out}) \text{ (remember } R_{ldr} \text{ is expressed in kOhm)}$$

$$\Rightarrow \text{Lux} = (500 * (5 - V_{out})) / (10 * V_{out})$$

$$\Rightarrow \text{Lux} = (2500 - 500 * V_{out}) / (10 * V_{out})$$

$$\Rightarrow \text{Lux} = (2500 / V_{out} - 500) / 10$$

$$\Rightarrow \text{Lux} = (2500 / ((\text{AnalogRead} * 0.0048828125) - 500)) / 10$$

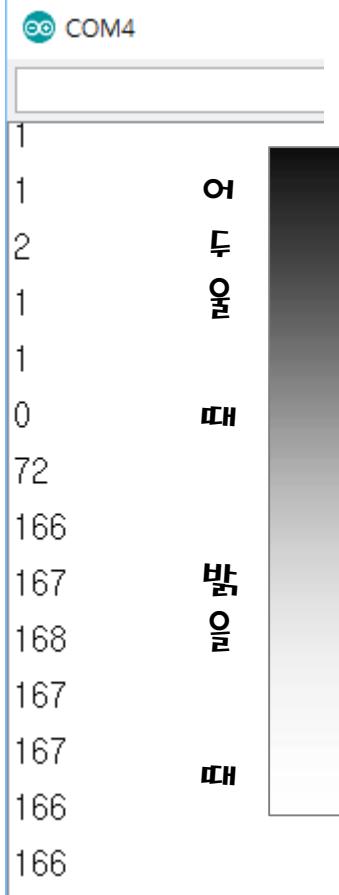
<https://arduinodiy.wordpress.com/2013/1/03/measuring-light-with-an-arduino/>



A3.2.5 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 2.

```
AA00_CdS
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 LuxLux
14 double luminosity (int RawADC0){
15 double Vout=RawADC0*0.0048828125; // 5/1024 (Vin = 5 V)
16 int lux=(2500/Vout-500)/10; // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
17 return lux;
18 }
```



밝을수록 측정 값이 커지고 어두울수록 값이 작아진다 !!!



A5.2.1 Luminosity sensor [Photocell LDR]

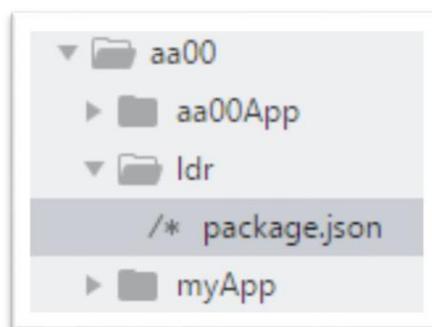
1. Make ldr node project

➤ **md ldr**

➤ **cd ldr**

2. Go to ldr subfolder

➤ **npm init**



```
{  
  "name": "ldr",  
  "version": "1.0.0",  
  "description": "",  
  "main": "ldr_node.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "keywords": [  
    "cds",  
    "ldr",  
    "node",  
    "arduino"  
  ],  
  "author": "aa00",  
  "license": "MIT"  
}
```

"main": "ldr_node.js"
"author": "aann"



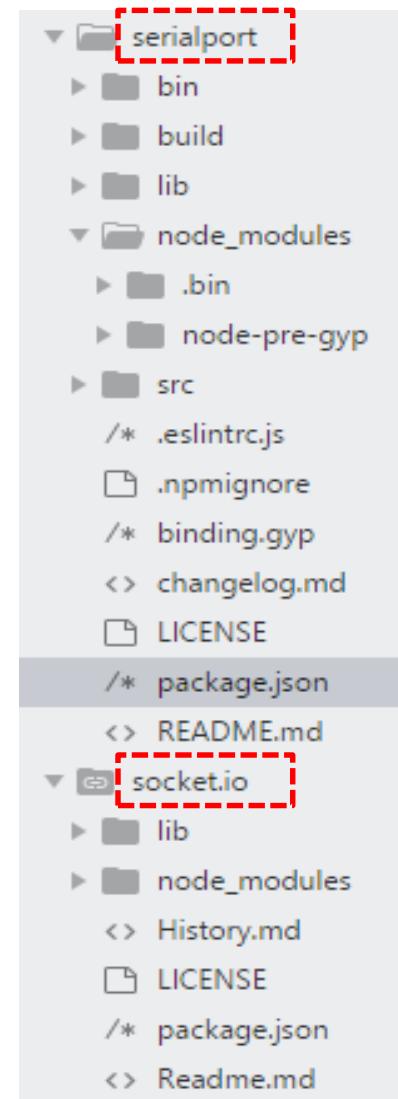
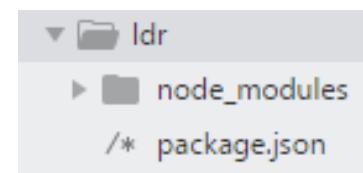
A5.2.2 Luminosity sensor [Photocell LDR]

1. Make ldr node project

- `md ldr`
- `cd ldr`

2. Go to ldr 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 browsing package.json in each module subfolder.



A5.2.3 Luminosity sensor [Photocell LDR]

1. Make ldr node project

➤ `md ldr`

➤ `cd ldr`

2. Go to ldr subfolder

➤ `npm init`

➤ `npm install --save serialport@4.0.7`

➤ `npm install --save socket.io@1.7.3`

package.json

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



A5.2.4 Luminosity sensor [Photocell LDR]

```
▼ └── ldr
    ├── node_modules
    └── /* ldr_node.js
        └── package.json
```

Save tmp36_node.js as ldr_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
});

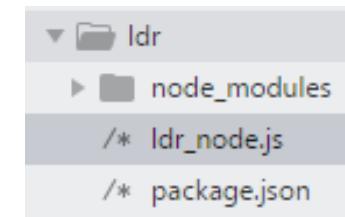
// helper function to get a nicely formatted date string
function getDateString() {
    var time = new Date().getTime();
    // 32400000 is (GMT+9 Korea, GimHae)
    // for your timezone just multiply +/-GMT by 3600000
    var datestr = new Date(time +32400000).
        toISOString().replace(/\T/, '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



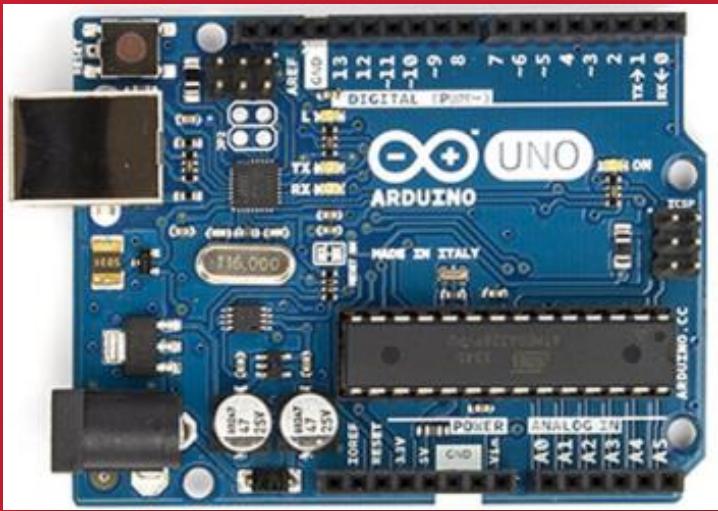
A5.2.5 Luminosity sensor [Photocell LDR]

Run ldr_node.js (^B)

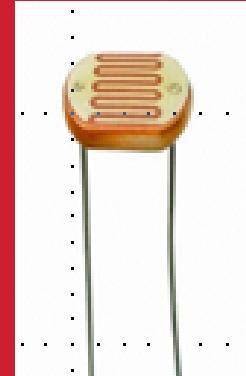
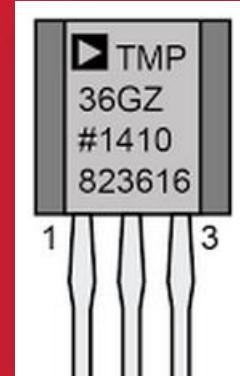
```
AA00,2017-11-08 08:49:54.597,171  
AA00,2017-11-08 08:49:55.589,171  
AA00,2017-11-08 08:49:56.598,173  
AA00,2017-11-08 08:49:57.589,173  
AA00,2017-11-08 08:49:58.596,172  
AA00,2017-11-08 08:49:59.588,171  
AA00,2017-11-08 08:50:00.580,173  
AA00,2017-11-08 08:50:01.588,173  
AA00,2017-11-08 08:50:02.579,171  
AA00,2017-11-08 08:50:03.586,172  
AA00,2017-11-08 08:50:04.578,173  
AA00,2017-11-08 08:50:05.571,172
```

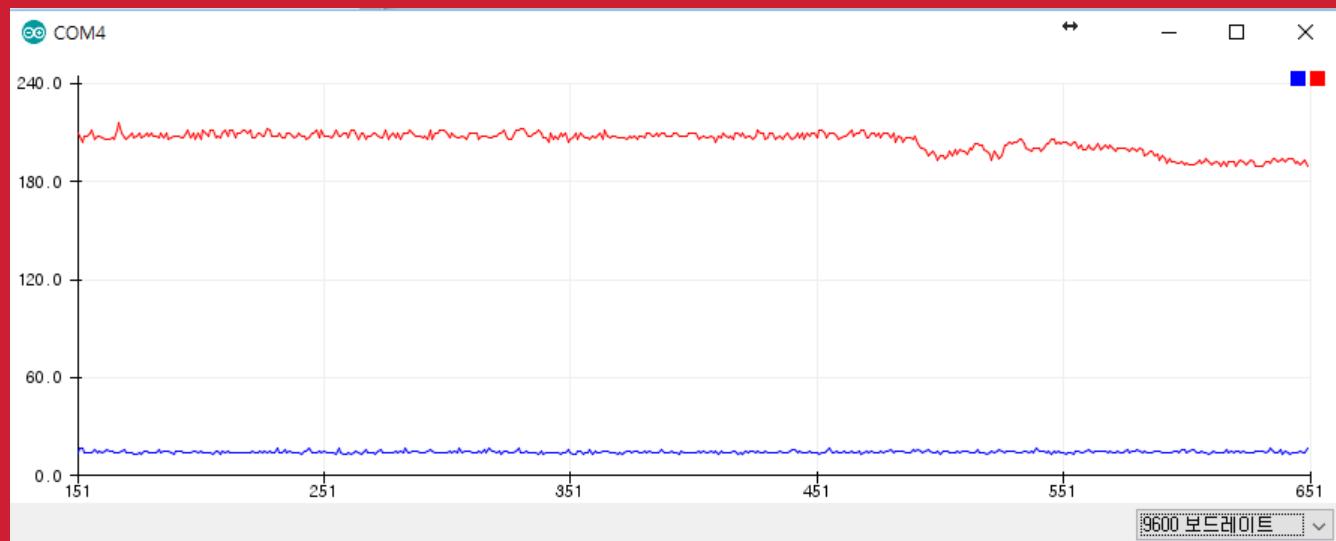
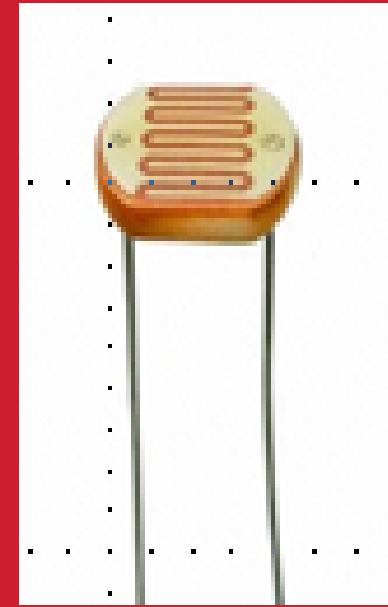
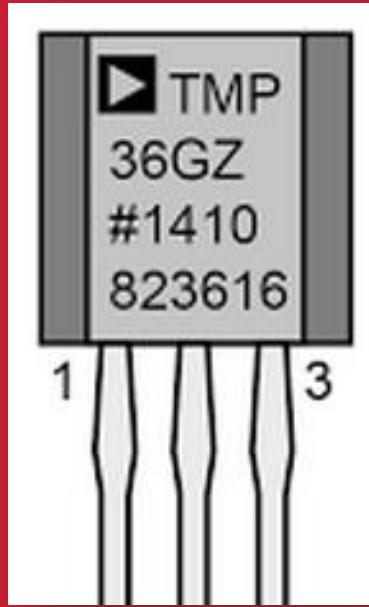


Save as
AAnn_lux_data.png



Arduino & Node.js

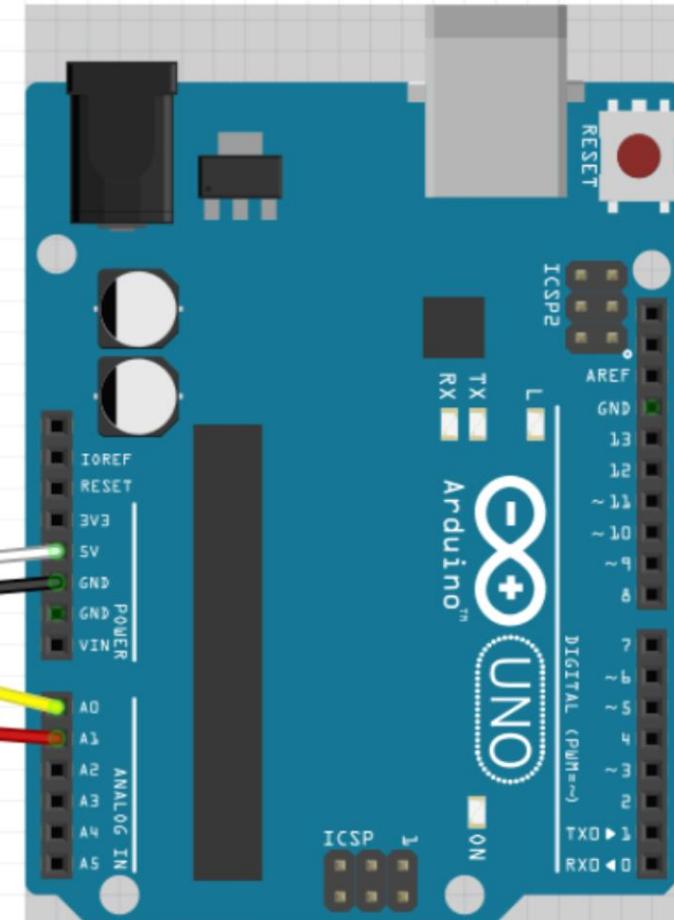
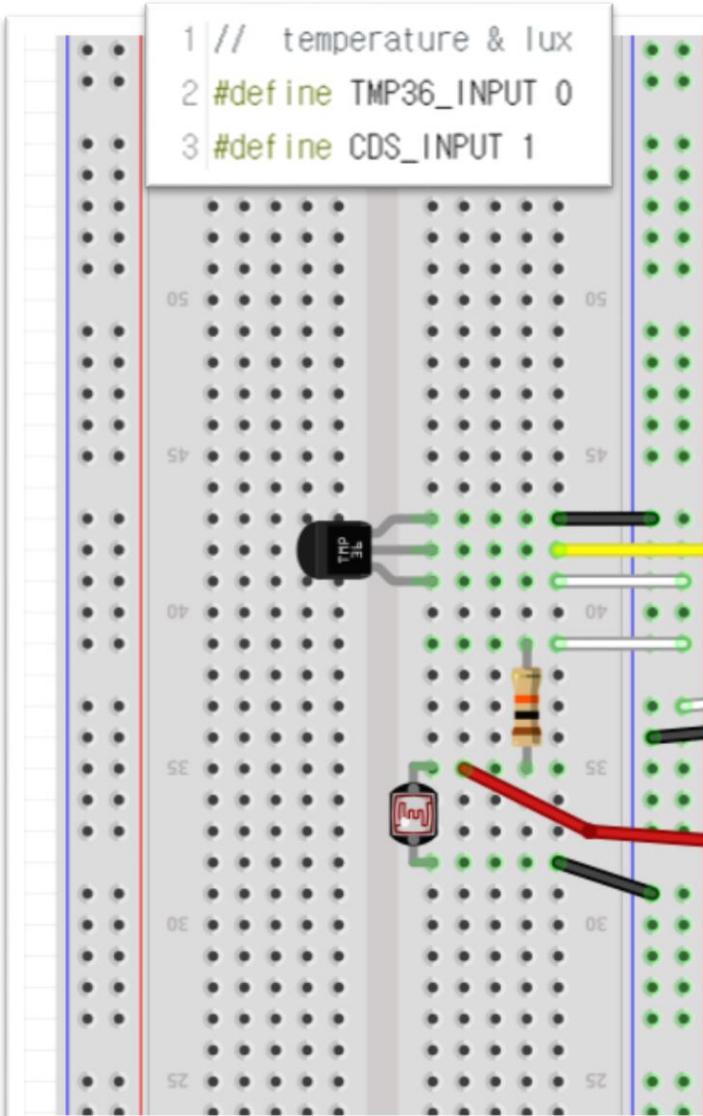






A5.3.1 TMP36 + CdS : circuit

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





A5.3.2 TMP36 + CdS : code

AAnn_TMP36_Cds \$

```
1 // temperature & lux
2 #define TMP36_INPUT 0
3 #define CDS_INPUT 1
4
5 void setup() {
6   Serial.begin(9600);
7 }
```

```
8 void loop() {
9   // Temperature from TMP36
10  int temp_value = analogRead(TMP36_INPUT);
11  // converting that reading to voltage
12  float voltage = temp_value * 5.0 * 1000; // in mV
13  voltage /= 1023.0;
14  float tempC = (voltage - 500) / 10 ;
15
16  // Lux from CdS (LDR)
17  int cds_value = analogRead(CDS_INPUT);
18  int lux = int(luminosity(cds_value));
19
20  Serial.print(tempC);
21  Serial.print(",");
22  Serial.println(lux);
23
24  delay(1000);
25 }
26
27 //Voltage to LuxLux
28 double luminosity (int RawADC0){
29   double Vout=RawADC0*0.0048828125; // 5/1024 (Vin = 5 V)
30   int lux=(2500/Vout-500)/10;
31   // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
32   return lux;
33 }
```



A5.3.3 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

COM4

240.0

180.0

120.0

60.0

0.0

0

100

200

300

400

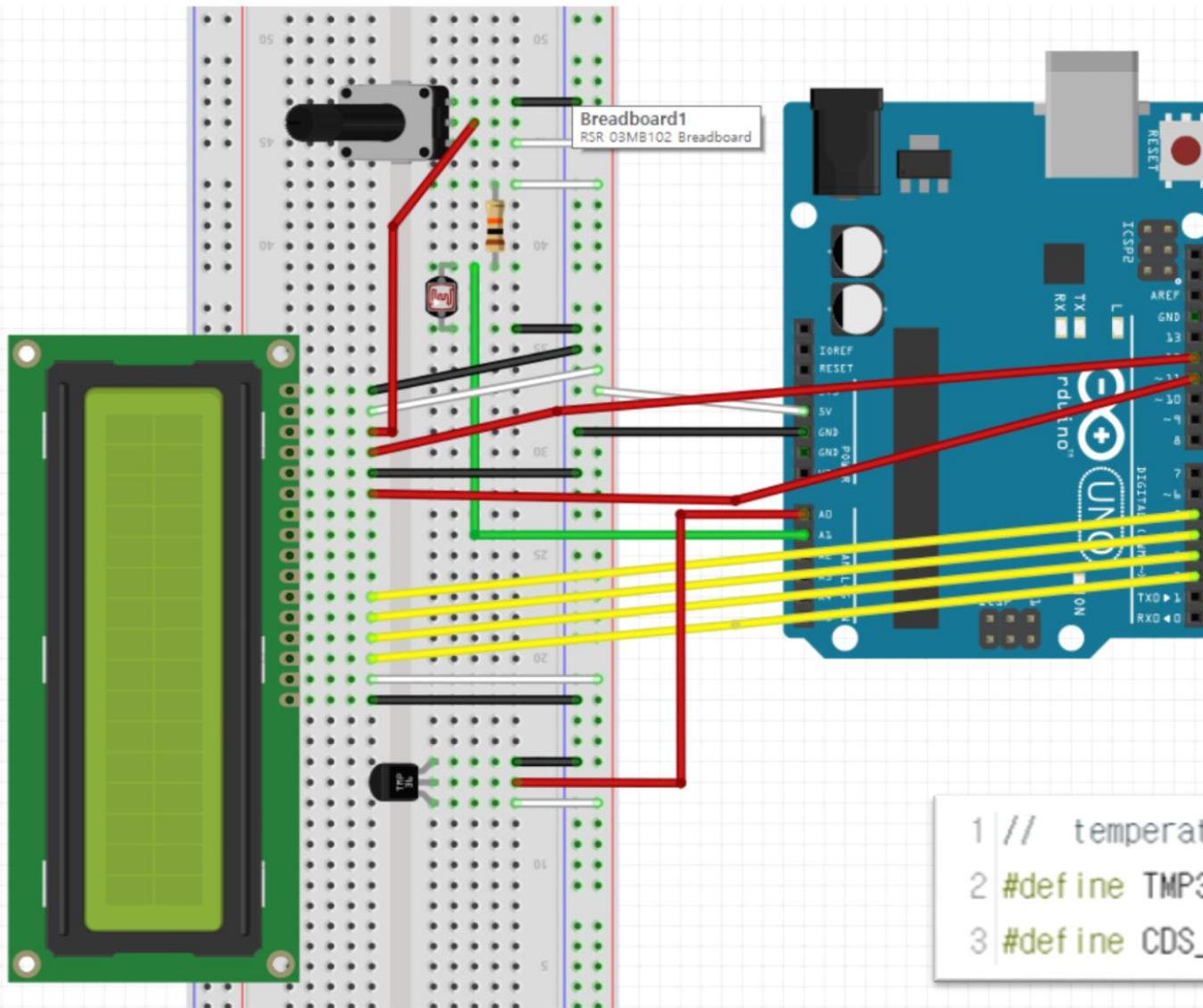
500

9600 보드레이트





A5.3.4 TMP36 + CdS + LCD : circuit



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



A5.3.5 TMP36 + CdS + LCD : code

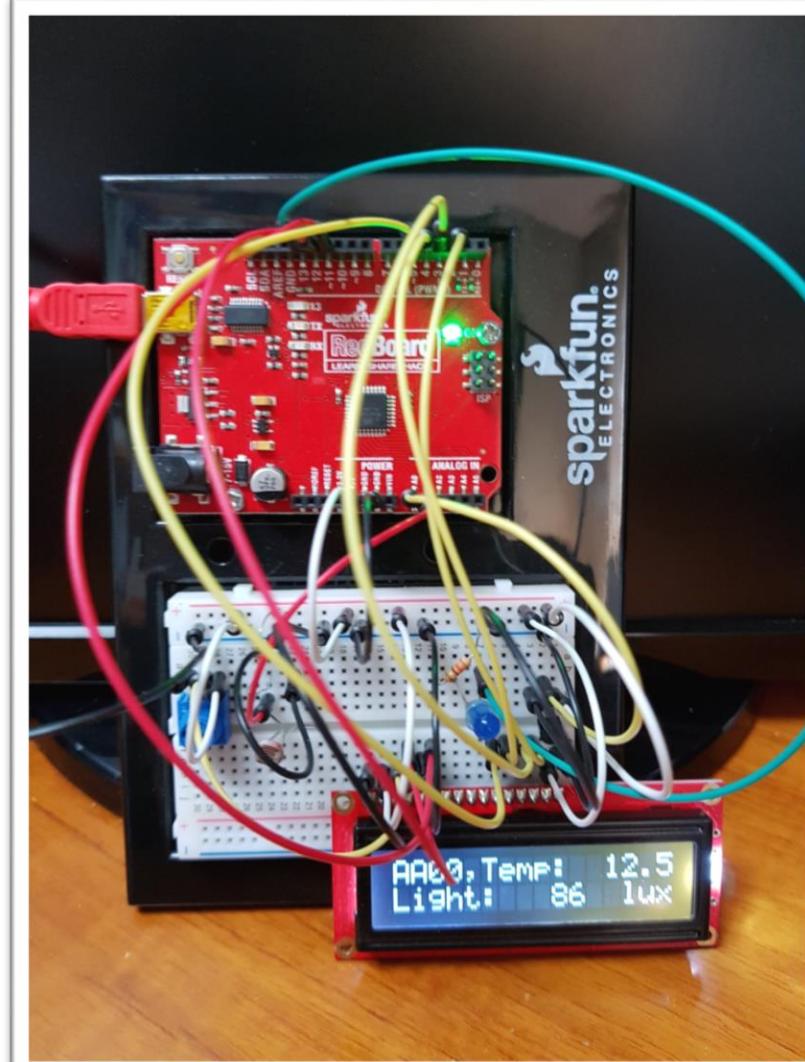
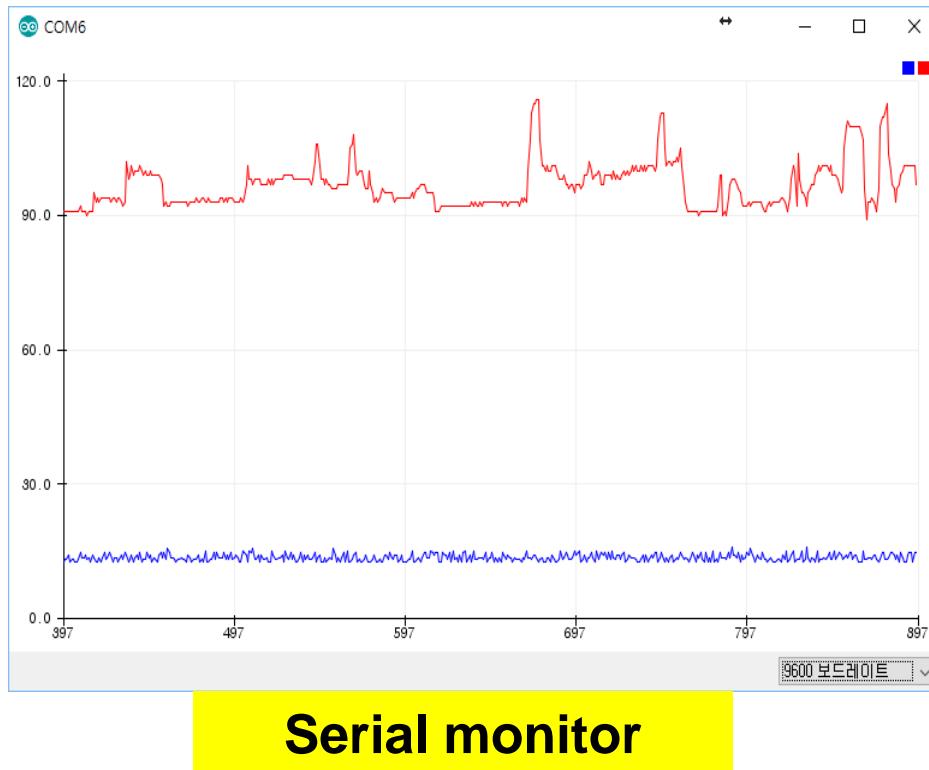
AA00_TMP36_Cds_LCD

```
1 /*  
2 온도, 빛 입력 및 LCD 모니터링  
3 */  
4  
5 // LCD 라이브러리 설정  
6 #include <LiquidCrystal.h>  
7 // LCD 설정  
8 LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // rs,en,d4,d5,d6,d7  
9 // 0번 아날로그핀을 TMP36 온도 입력으로 설정한다.  
10 // 1번 아날로그핀을 CdS 조도 입력으로 설정한다.  
11 #define TMP36_INPUT 0  
12 #define CDS_INPUT 1  
13  
14 void setup() {  
15   Serial.begin(9600);  
16   // 16X2 LCD 모듈 설정하고 백라이트를 켠다.  
17   lcd.begin(16,2);  
18   // 메세지를 표시한다.  
19   lcd.print("AA00");  
20   lcd.setCursor(0,1);  
21   lcd.print("TMP36 & CdS Test");  
22   // 3초동안 메세지를 표시한다.  
23   delay(3000);  
24   // 모든 메세지를 삭제한 뒤  
25   // 숫자를 제외한 부분들을 미리 출력시킨다.  
26   lcd.clear();  
27   lcd.setCursor(0,0);  
28   lcd.print("AA00,Temp: ");  
29   lcd.setCursor(0,1);  
30   lcd.print("Light: ");  
31   lcd.setCursor(13,1);  
32   lcd.print("lux"); //  
33 }
```

```
35 void loop(){  
36   // Temperature from TMP36  
37   int temp_value = analogRead(TMP36_INPUT);  
38   // converting that reading to voltage  
39   float voltage = temp_value * 5.0 * 1000; // in mV  
40   voltage /= 1023.0;  
41   float tempC = (voltage - 500) / 10 ;  
42  
43   // Lux from CdS (LDR)  
44   int cds_value = analogRead(CDS_INPUT);  
45   int lux = int(luminosity(cds_value));  
46  
47   // 전에 표시했던 내용을 지운다.  
48   lcd.setCursor(12,0);  
49   lcd.print("    ");  
50   // ADC 값을 표시한다  
51   lcd.setCursor(12,0);  
52   lcd.print(tempC);  
53   // 전에 표시했던 내용을 지운다.  
54   lcd.setCursor(9,1);  
55   lcd.print("    ");  
56   // 밝기를 표시한다  
57   lcd.setCursor(9,1);  
58   lcd.print(lux); // 수정  
59  
60   // Serial output  
61   Serial.print(tempC);  
62   Serial.print(",");  
63   Serial.println(lux);  
64   delay(1000);  
65 }
```



A5.3.6 TMP36 + CdS + LCD: result



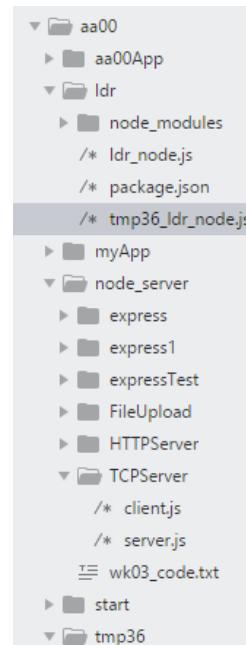
Save as
AAnn_tmp36_lux_LCD.png



A5.3.7 TMP36 + CdS + LCD + NodeJS

Recycling code:

Save ldr_node.js as
tmp36_ldr_node.js
and run (^B)



```
18
19 var dStr = '';
20 var tdata = [];
21
22 sp.on('data', function (data) { // call ba
23   // raw data only
24   //console.log(data);
25   dStr = getDateString();
26   tdata[0] = dStr; // date
27   tdata[1] = data; // data
28   console.log("AA00," + tdata);
29   io.sockets.emit('message', tdata);
30 });
31
32 // helper function to get a nicely formatt
33 -function getDateString() {
34   var d = new Date();
35   var dd = d.getDate();
36   var mm = d.getMonth() + 1;
37   var yy = d.getFullYear();
38   var hh = d.getHours();
39   var mm1 = d.getMinutes();
40   var ss = d.getSeconds();
41
42   if (dd < 10) dd = '0' + dd;
43   if (mm < 10) mm = '0' + mm;
44
45   return yy + '-' + mm + '-' + dd + ' ' + hh + ':' + mm1 + ':' + ss;
46 }
```

```
AA00,2017-11-08 10:54:25.377,13.54,98
AA00,2017-11-08 10:54:26.385,13.54,98
AA00,2017-11-08 10:54:27.392,12.56,98
AA00,2017-11-08 10:54:28.401,13.05,99
AA00,2017-11-08 10:54:29.407,14.03,98
AA00,2017-11-08 10:54:30.414,13.54,98
AA00,2017-11-08 10:54:31.406,14.03,98
AA00,2017-11-08 10:54:32.414,13.54,99
AA00,2017-11-08 10:54:33.421,13.54,99
AA00,2017-11-08 10:54:34.429,14.03,99
AA00,2017-11-08 10:54:35.437,13.05,98
AA00,2017-11-08 10:54:36.428,14.03,99
AA00,2017-11-08 10:54:37.435,12.56,99
AA00,2017-11-08 10:54:38.443,15.00,100
AA00,2017-11-08 10:54:39.450,12.56,97
```



A5.3.7 TMP36 + CdS + LCD + NodeJS

Recycling code:

Save ldr_node.js as
tmp36_ldr_node.js

and run in NodeJs

> node tmp36_ldr_node

Save as
**AAnn_tmp36_lux_data.
png**

NodeJS - node tmp36_ldr_node

```
D:\Portable\NodeJSPortable\Data\aa00\ldr>node tmp36_ldr_node
AA00,2017-11-08 11:17:49.040,14.52,95
AA00,2017-11-08 11:17:50.047,12.56,95
AA00,2017-11-08 11:17:51.055,14.03,95
AA00,2017-11-08 11:17:52.063,14.52,94
AA00,2017-11-08 11:17:53.070,14.03,95
AA00,2017-11-08 11:17:54.063,14.03,95
AA00,2017-11-08 11:17:55.069,13.54,95
AA00,2017-11-08 11:17:56.077,12.56,95
AA00,2017-11-08 11:17:57.084,13.54,95
AA00,2017-11-08 11:17:58.092,13.54,95
AA00,2017-11-08 11:17:59.085,12.56,95
AA00,2017-11-08 11:18:00.091,13.54,95
AA00,2017-11-08 11:18:01.098,13.05,95
AA00,2017-11-08 11:18:02.105,13.54,95
AA00,2017-11-08 11:18:03.113,13.54,95
AA00,2017-11-08 11:18:04.121,14.52,94
AA00,2017-11-08 11:18:05.112,15.00,95
AA00,2017-11-08 11:18:06.120,14.52,93
AA00,2017-11-08 11:18:07.129,13.05,92
AA00,2017-11-08 11:18:08.134,15.00,91
AA00,2017-11-08 11:18:09.142,12.56,91
AA00,2017-11-08 11:18:10.134,12.56,91
AA00,2017-11-08 11:18:11.142,13.05,91
AA00,2017-11-08 11:18:12.150,12.56,91
AA00,2017-11-08 11:18:13.158,13.05,91
AA00,2017-11-08 11:18:14.165,13.54,91
AA00,2017-11-08 11:18:15.172,13.54,91
AA00,2017-11-08 11:18:16.163,13.54,91
AA00,2017-11-08 11:18:17.170,15.00,92
AA00,2017-11-08 11:18:18.178,12.56,91
AA00,2017-11-08 11:18:19.187,14.03,91
AA00,2017-11-08 11:18:20.195,14.03,91
AA00,2017-11-08 11:18:21.186,12.56,91
AA00,2017-11-08 11:18:22.193,13.05,92
AA00,2017-11-08 11:18:23.199,13.54,91
AA00,2017-11-08 11:18:24.207,13.54,91
AA00,2017-11-08 11:18:25.216,12.56,92
AA00,2017-11-08 11:18:26.223,15.49,91
```



[Practice]

◆ [wk11]

- **Arduino sensors & node.js**
- **Complete your TMP36-CdS project**
- **Upload file name : AAnn_Rpt08.zip**

wk11 : Practice-08 : AAnn_Rpt08.zip

◆ [Target of this week]

- Complete your projects
- Save your outcomes and compress 4 figures

제출파일명 : **AAnn_Rpt08.zip**

- 압축할 파일들

- ① **AAnn_lux_data.png**
- ② **AAnn_tmp36_lux_LCD.png**
- ③ **tmp36_ldr_node.js**
- ④ **AAnn_tmp36_lux_data.png**

Email : **chaos21c@gmail.com**



[참고 : 저항 값 읽기]

Carbonfilm resistor

4 Color stripes

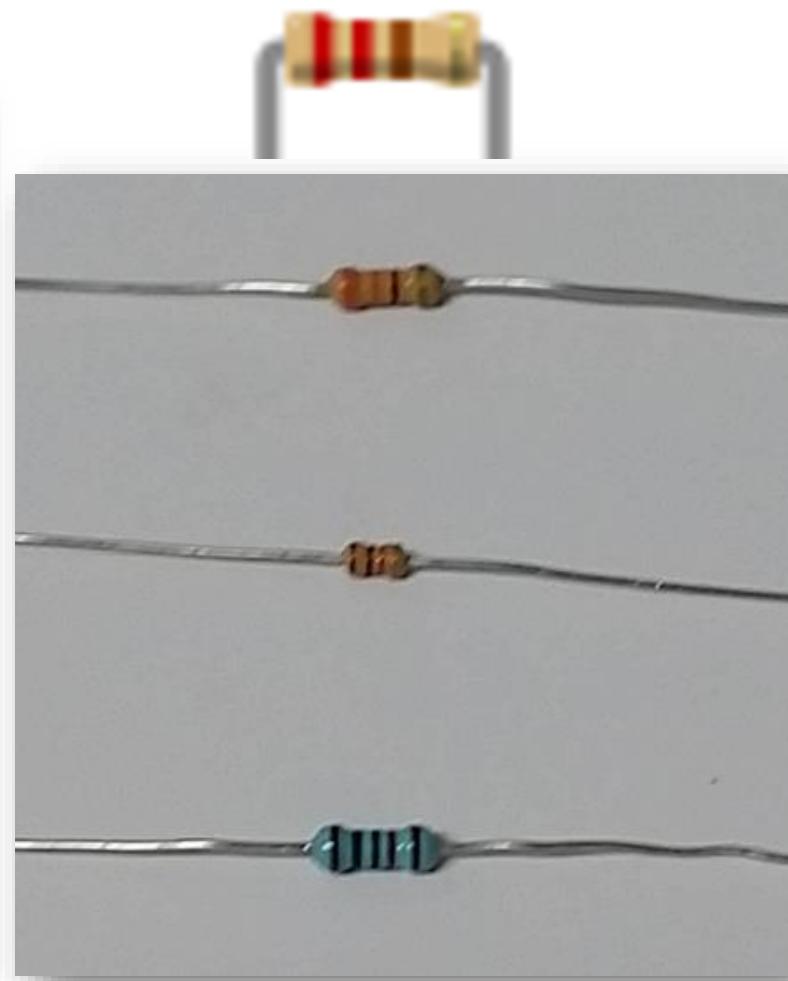
$$47 \times 1000 = 47\text{KOhm } 5\%$$

sm2k (c) 2006

5 Color stripes

$$576 \times 1 = 576 \text{ Ohm } 1\%$$

Color	First	Second	Third	Multiplier	Tolerance
Black	0	0	0	x1	
Brown	1	1	1	x10	1%
Red	2	2	2	x100	2%
Orange	3	3	3	x1000	
Yellow	4	4	4	x10 000	
Green	5	5	5	x100 000	0,50%
Blue	6	6	6	x1 000 000	0,25%
Violette	7	7	7	x10 000 000	0,10%
Gray	8	8	8		
White	9	9	9		
Silver				x0,01	10%
Gold				x0,1	5%

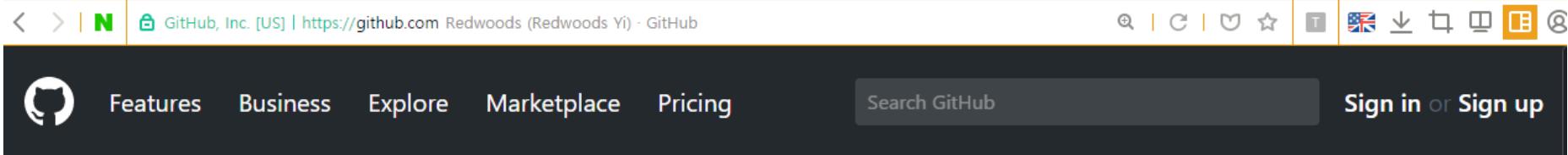


Lecture materials

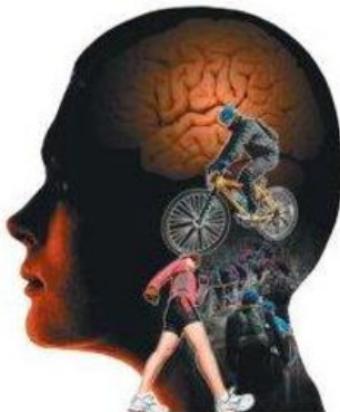


● References & good sites

- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling



A screenshot of a GitHub user profile page. At the top, there's a dark header with a navigation bar containing icons for back, forward, and search, along with links for GitHub features, business, explore, marketplace, and pricing. A search bar labeled "Search GitHub" is on the right, followed by "Sign in or Sign up".



Redwoods Yi

Redwoods

[Block or report user](#)

 GimHae, Republic of Korea

[Overview](#)

[Repositories 5](#)

[Stars 2](#)

[Followers 0](#)

[Following 0](#)

Pinned repositories

[dht22-iot-project](#)

Iot project to monitor data streaming from DHT22 wired at Arduino.

 HTML

[Lec](#)

All lectures by Redwoods in Inje University

[arduino-nodejs-plotly-streaming](#)

This repo introduces a simple and efficient way to plot the streaming data from Arduino with Easy Pulse ppg sensor or DHT11 sensor.

 HTML

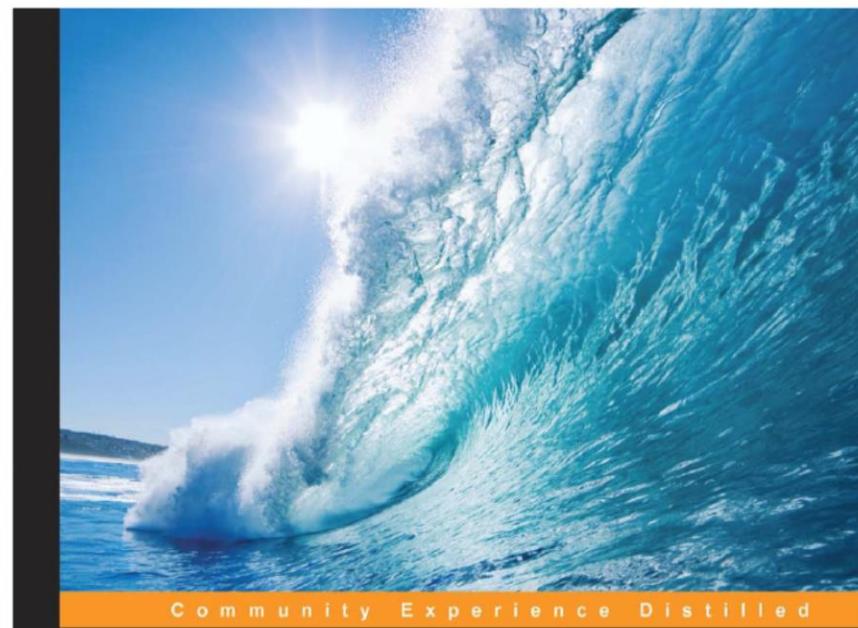
[hw-coding](#)

Resource for lecture of Hardware Programming (2017, Inje university)

 Arduino



References

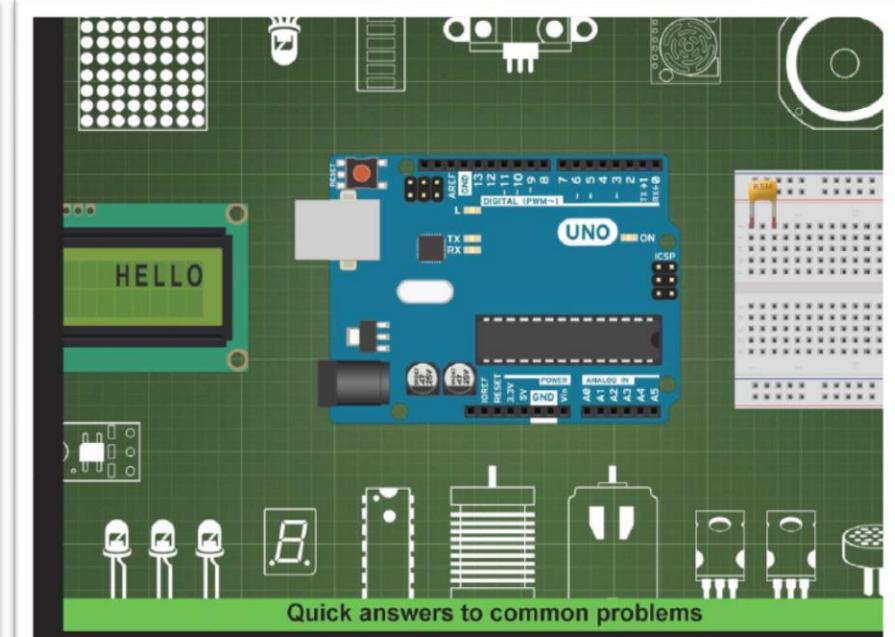


Arduino Essentials

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

Francis Perea

[PACKT]
PUBLISHING



Arduino Development Cookbook

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

Cornel Amariei

[PACKT] open source★
PUBLISHING