

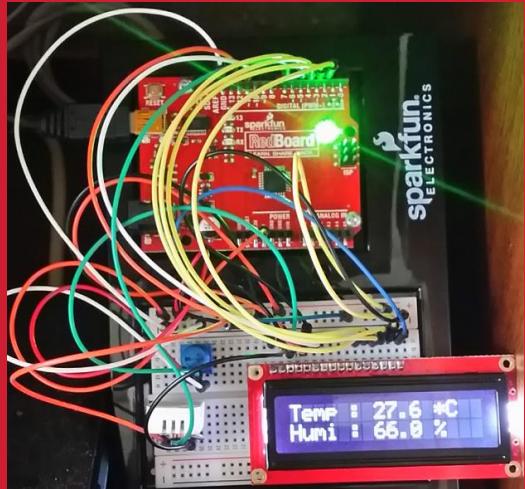


HW-SW-Connectivity

[wk10]

Arduino & NodeJS I

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

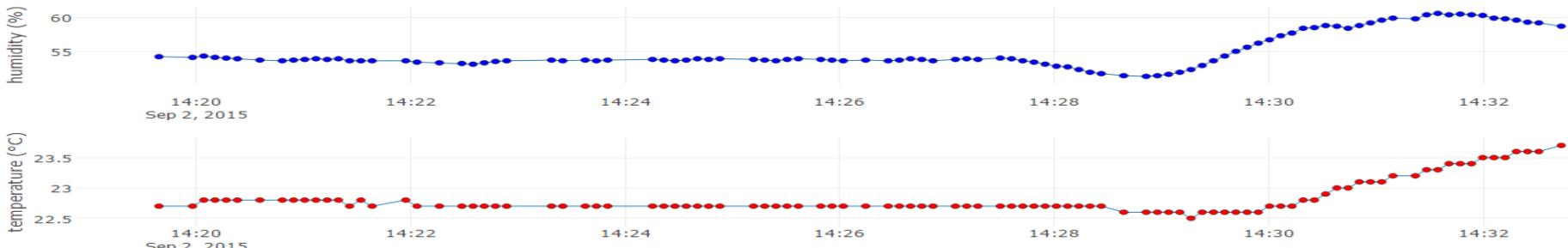


Basic HW and SW Integration using
Arduino & Javascript

COMSI, INJE University

2nd semester, 2017

Email : yish@inje.ac.kr





[Project]

◆ [wk09]

- Arduino LCD-CdS project
- Complete your project
- Upload file name : AAnn_Rpt09.zip

wk09 : Practice-06 : AAnn_Rpt06.zip

◆ [Target of this week]

- Complete your projects
- Save your outcomes and compres.

제출파일명 : **AAnn_Rpt05.zip**

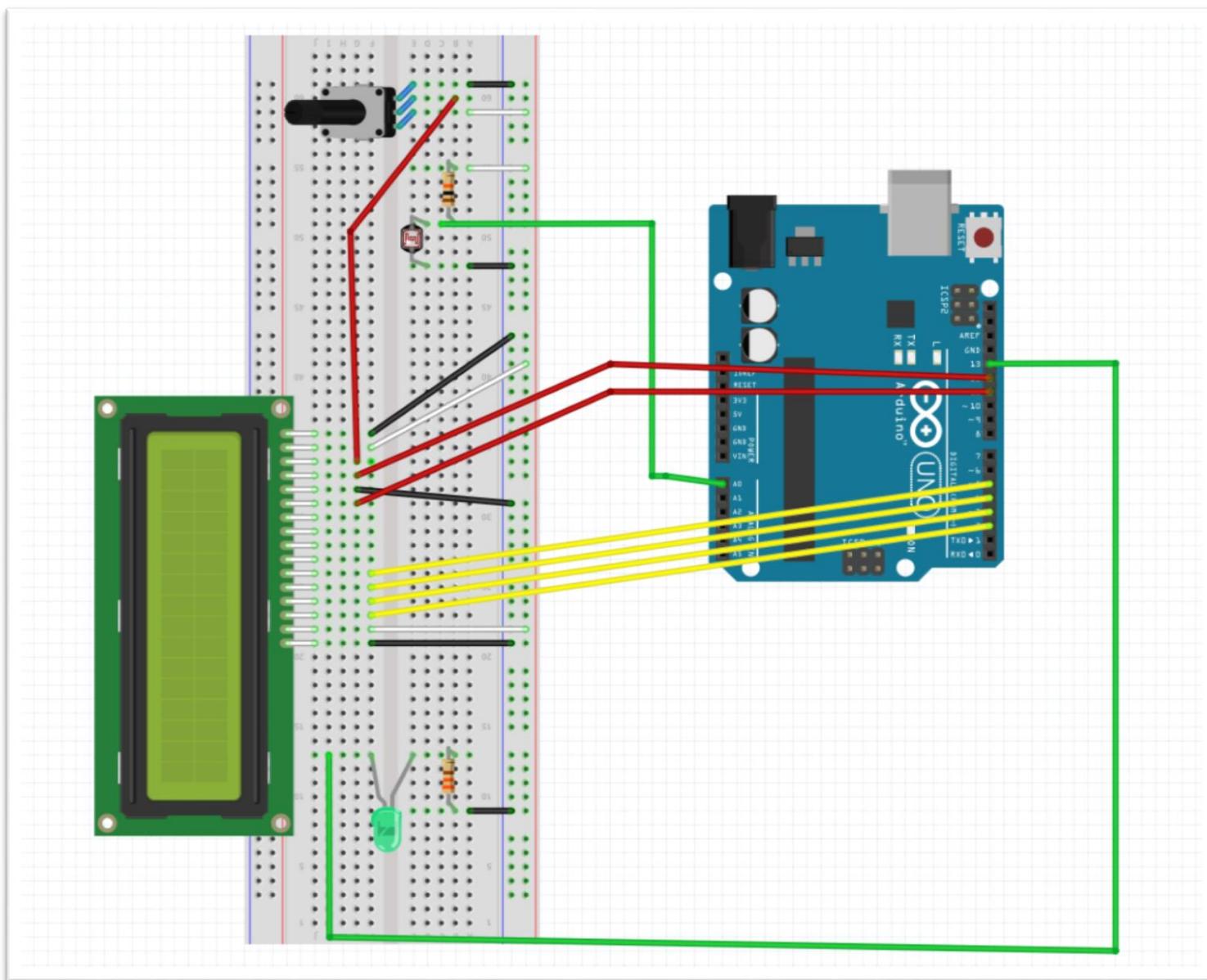
- 압축할 파일들

- ① **AAnn_Hello_LCD.png**
- ② **AAnn_LCD_lux.fzz**
- ③ **AAnn_LCD_lux.ino**
- ④ **AAnn_LCD_lux.png**



Email : **chaos21c@gmail.com**

wk09 : Practice-06 : AAnn_Rpt06.zip





Arduino

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

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



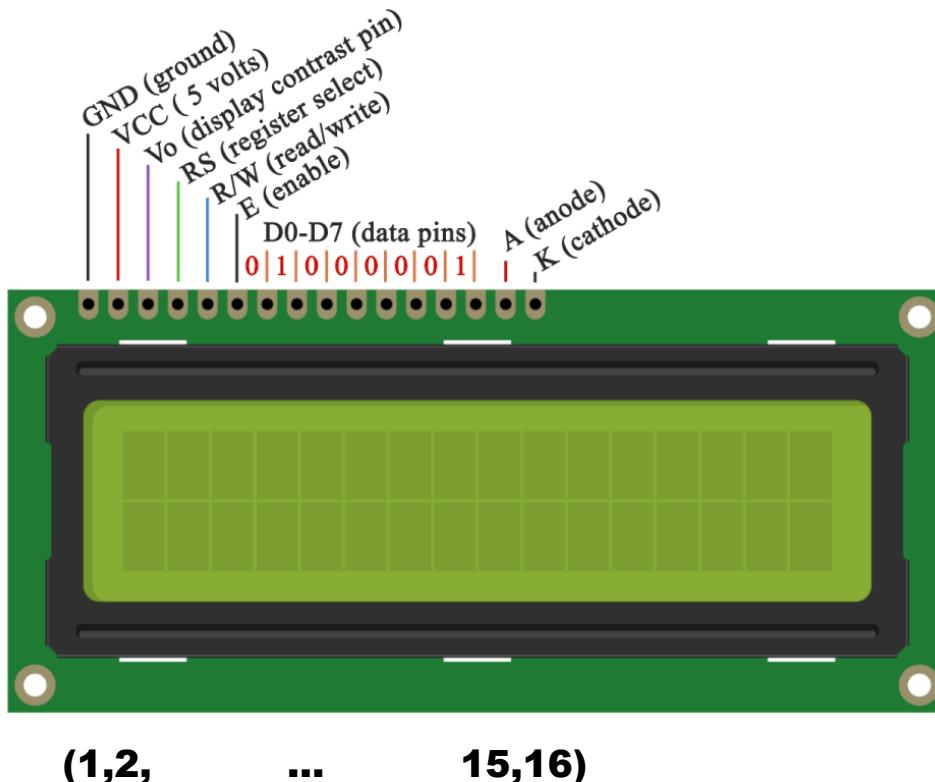
Introduction to LCD





Introduction to LCD

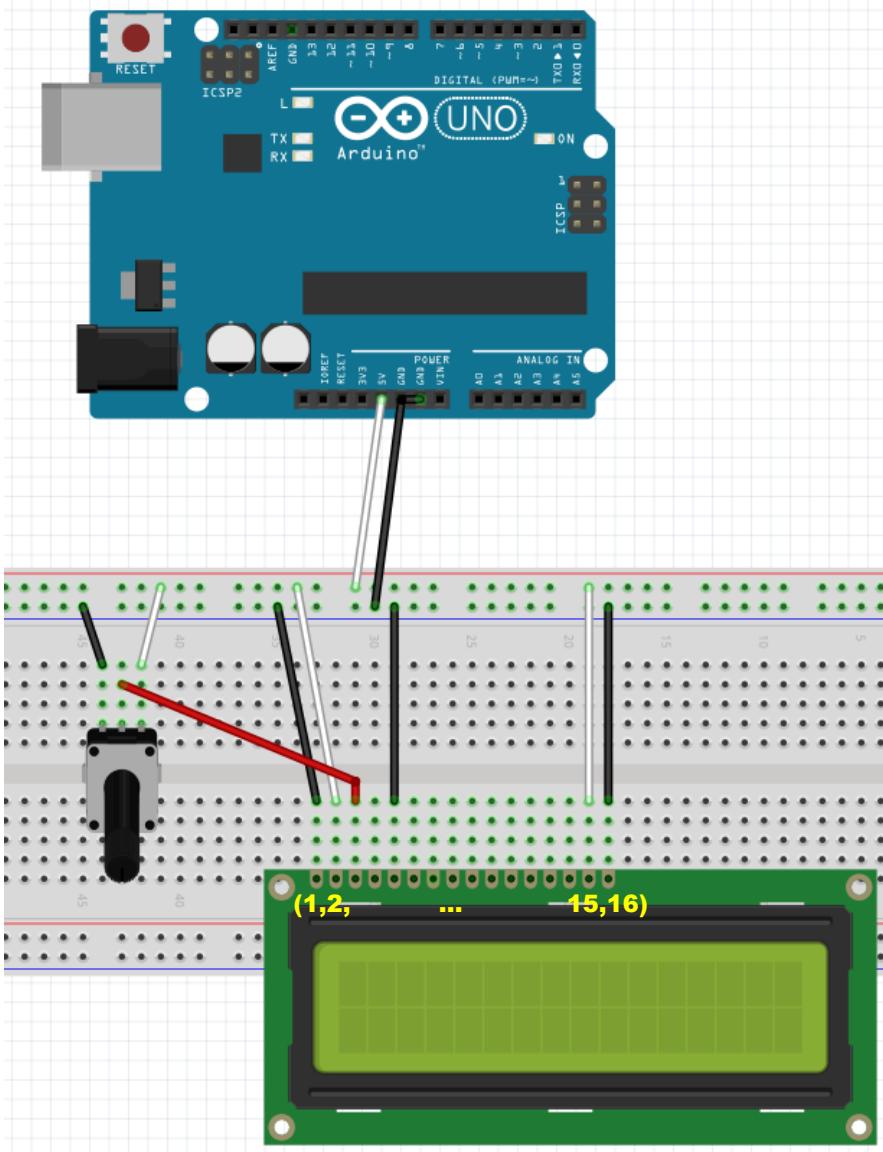
LCD (Liquid Crystal Display, 16 X 2)



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

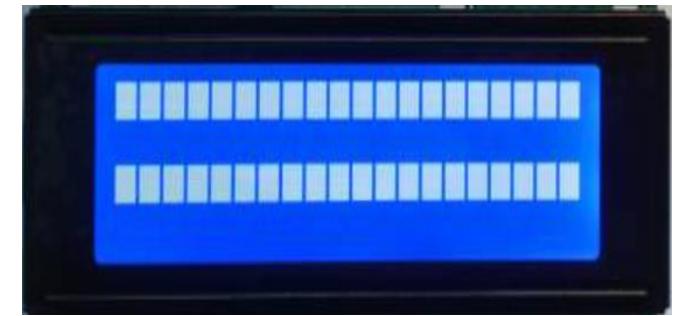


LCD 초기화 (pin-1, 2, 3, 5, 15,16)



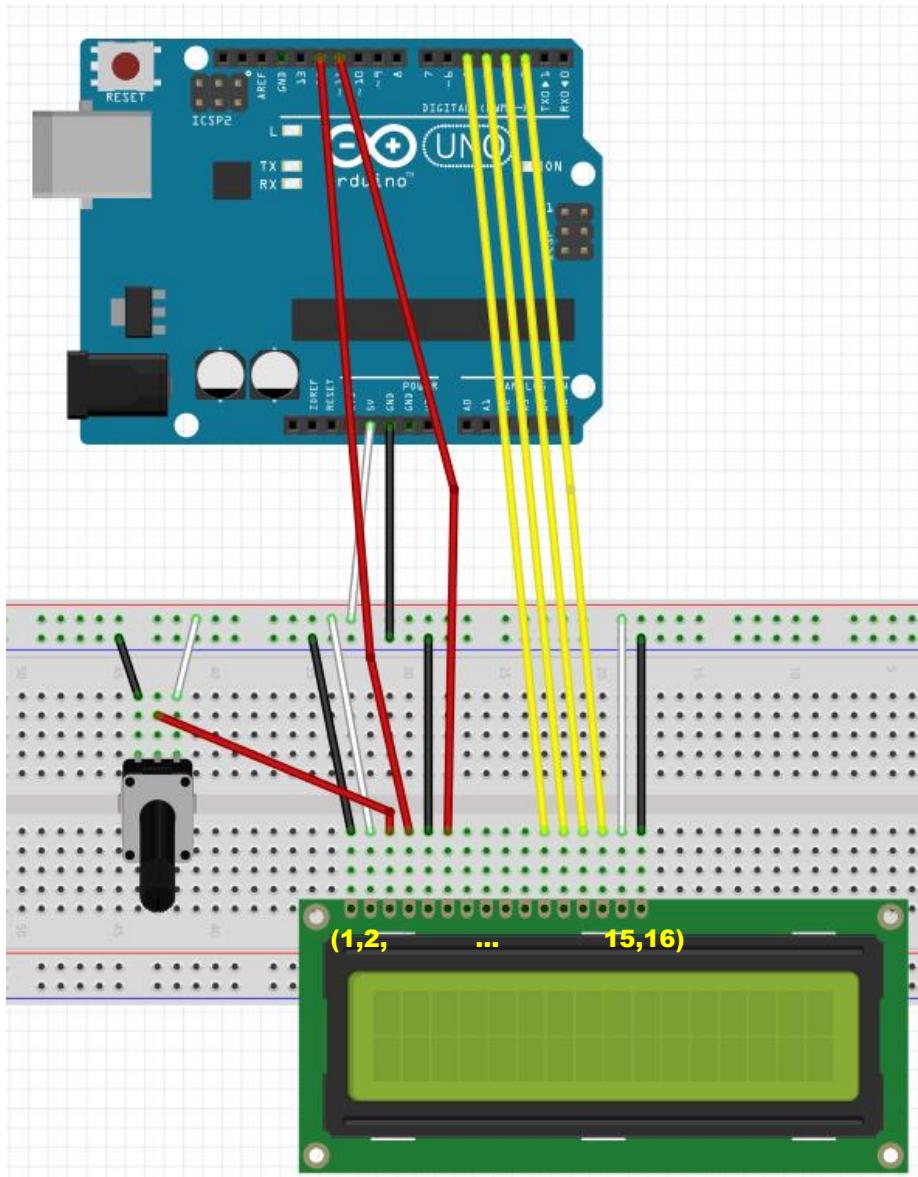
Pin 1 to Arduino GND
Pin 2 to Arduino +5V
Pin 3 to wiper (potentiometer)
Pin 5 to Arduino GND
Pin 15 to +5V
Pin 16 to GND

전원 연결 후
LCD 초기화





데이터 입력 초기화 (pin-4, 6, 11,12,13,14)



Pin 1 to Arduino GND

Pin 2 to Arduino 5V

Pin 3 to wiper

Pin 4 to Arduino pin D12

Pin 5 to Arduino GND

Pin 6 to Arduino pin D11

Pin 11 to Arduino pin D5

Pin 12 to Arduino pin D4

Pin 13 to Arduino pin D3

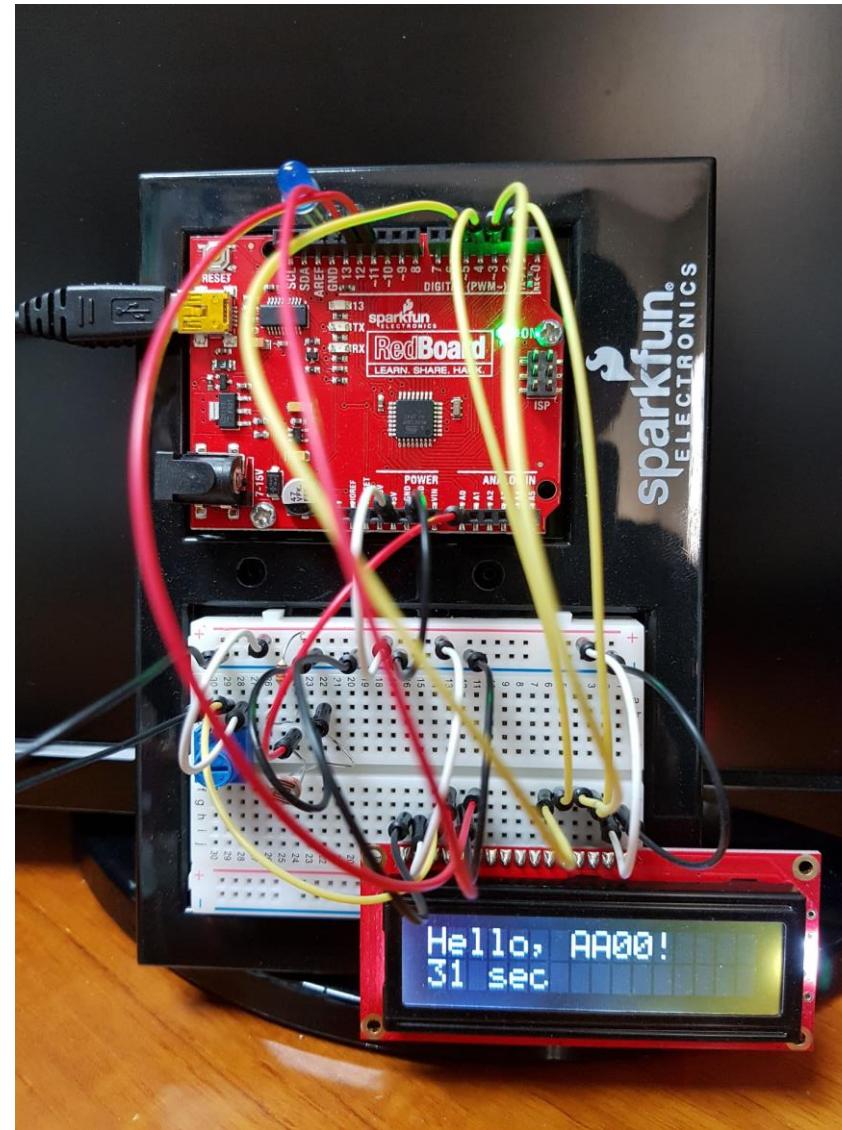
Pin 14 to Arduino pin D2

Pin 15 to +5V

Pin 16 to GND



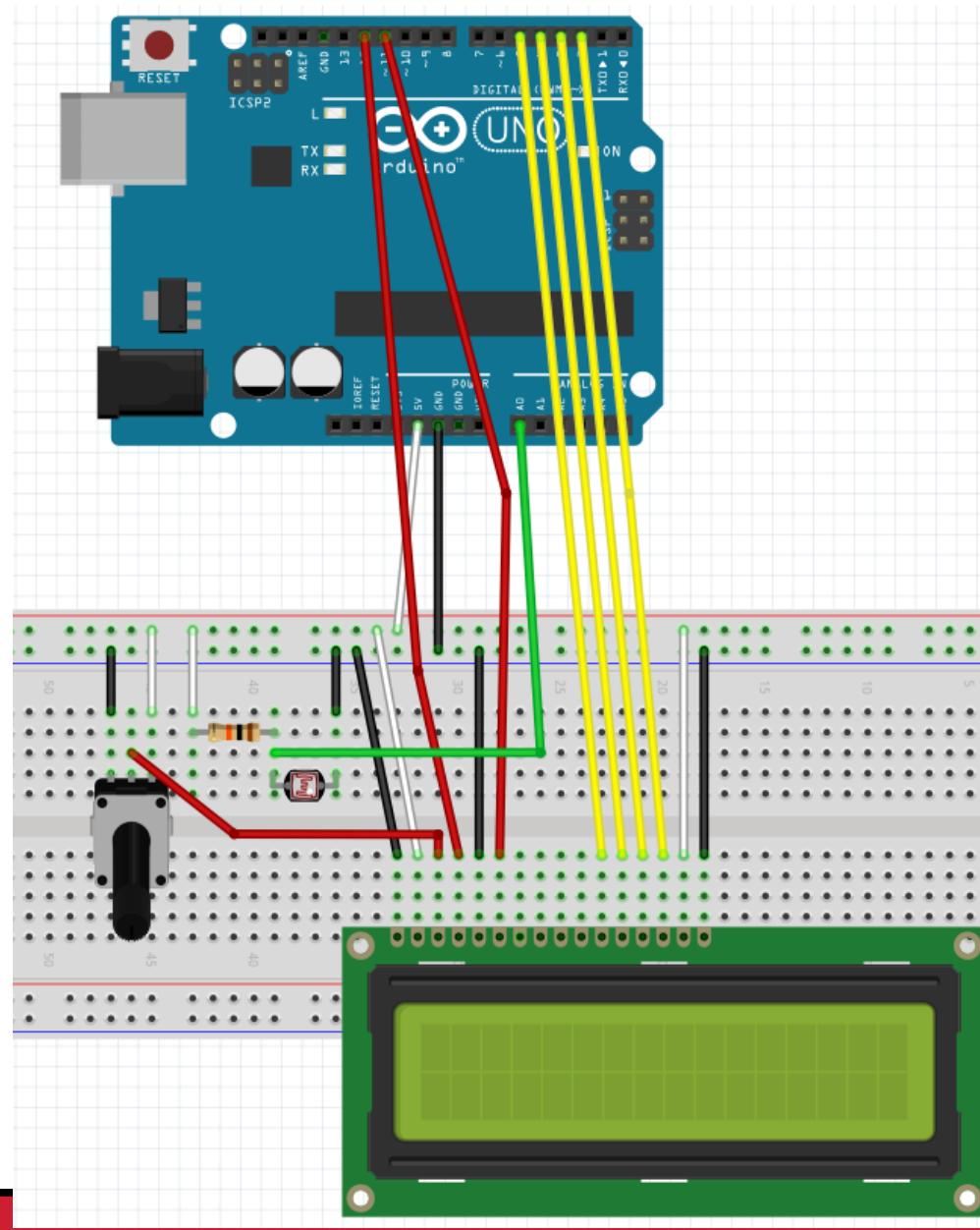
Introduction to LCD - “Hello AAnn”



결과 화면 찰영: **AAnn Hello LCD.png** 로
저장...

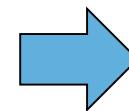
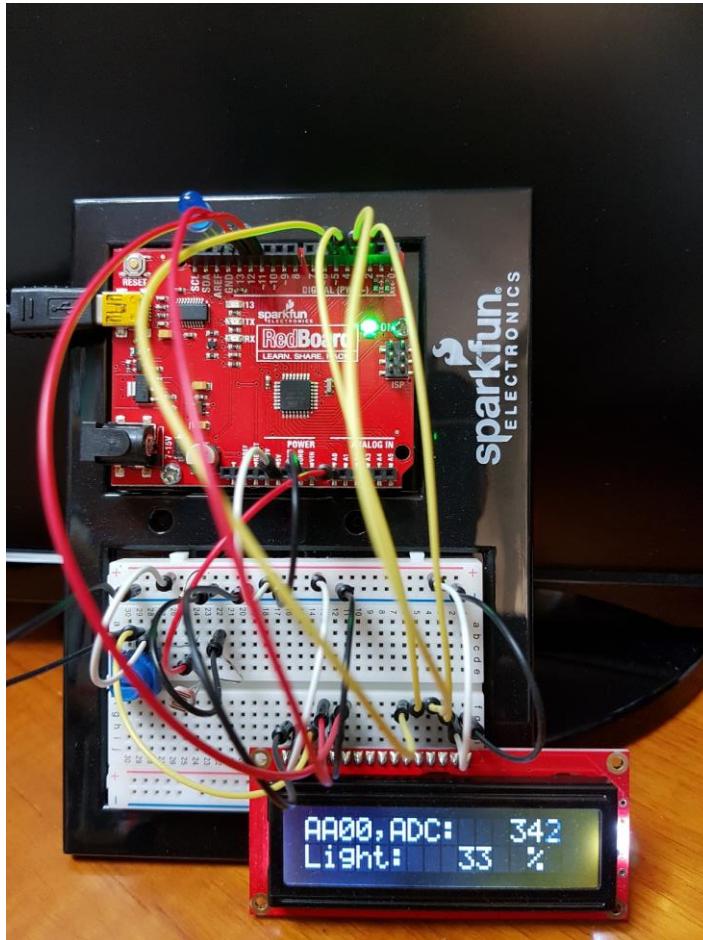


A3.3.3 Luminosity monitoring using LCD



A3.3.6 Luminosity monitoring using LCD

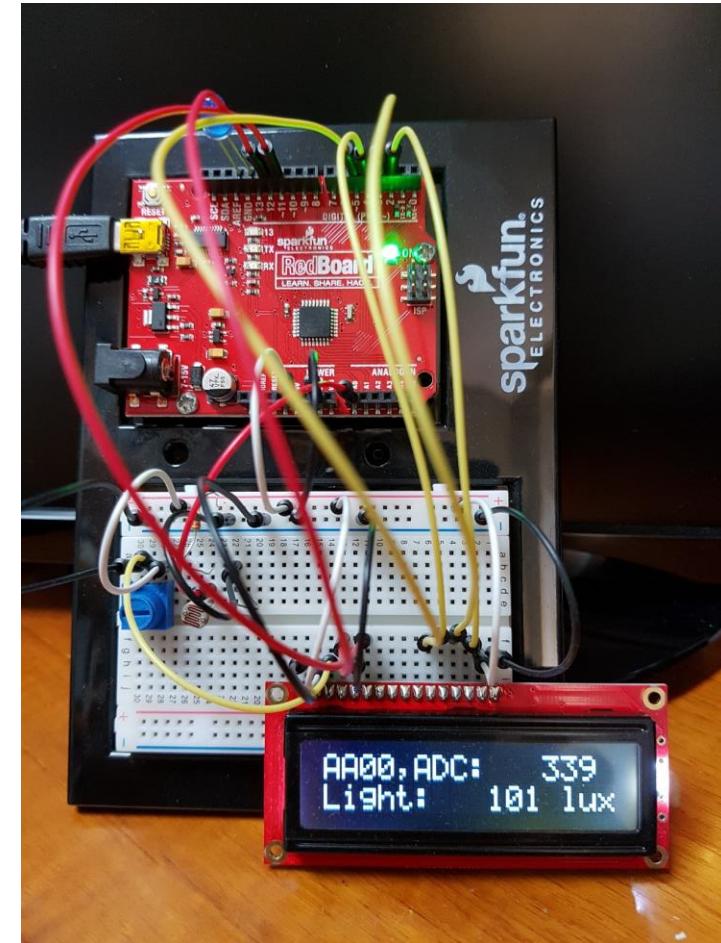
[DIY] CdS 센서 LCD 회로 - 측정



실제 밝기(가
lux로
표시되도록
코드를
수정하시오.

luminosity
함수 이용

수정된 화면을
폰으로
촬영해서
그림을
제출하시오.

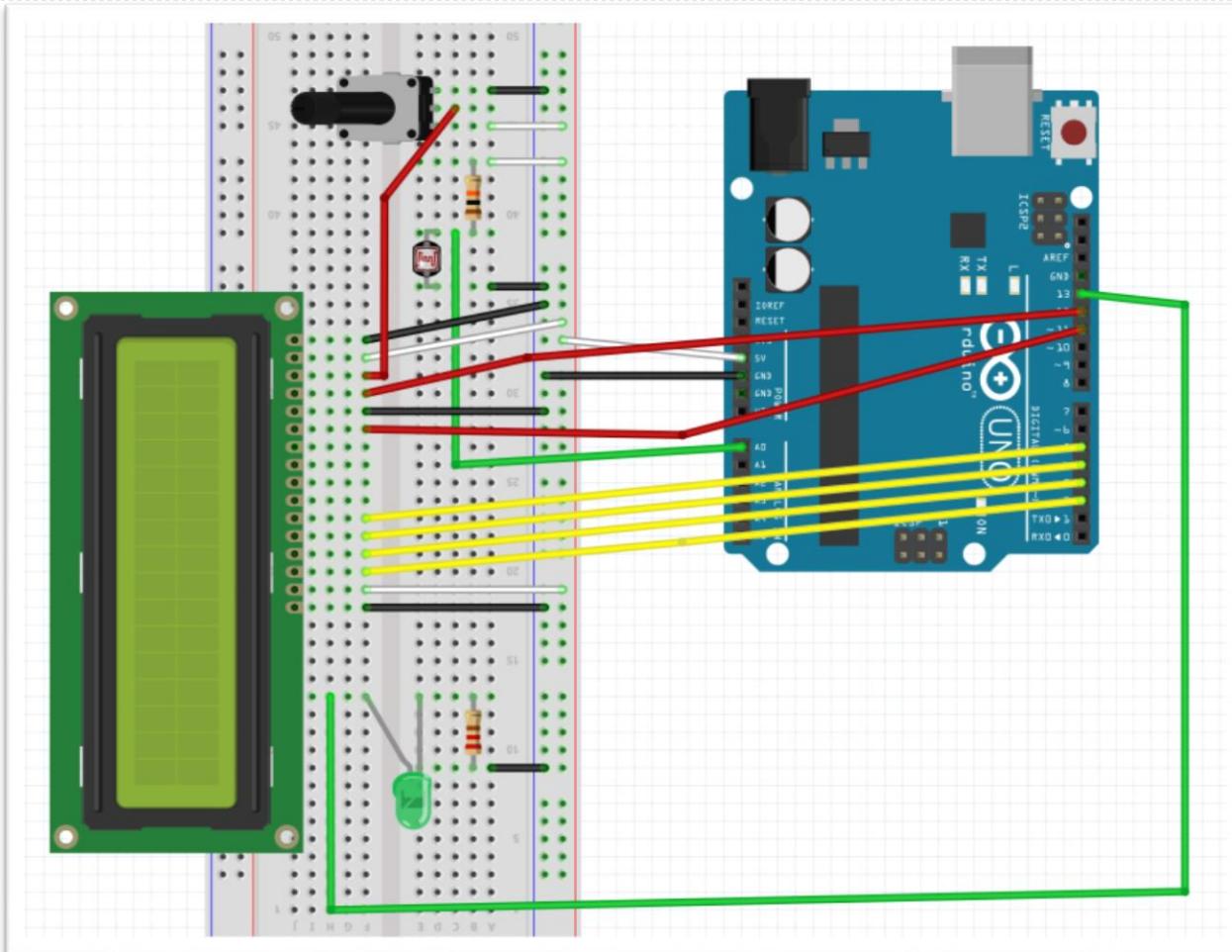


결과 화면 촬영: [AAnn_LCD_lux.png](#) 로 저장...



CdS-LCD project : fzz circuit

[DIY] AAAnn_LCD_lux.fzz로 저장.



Fritzing 회로도 소스를 AAnn_LCD lux.fzz로 저장...

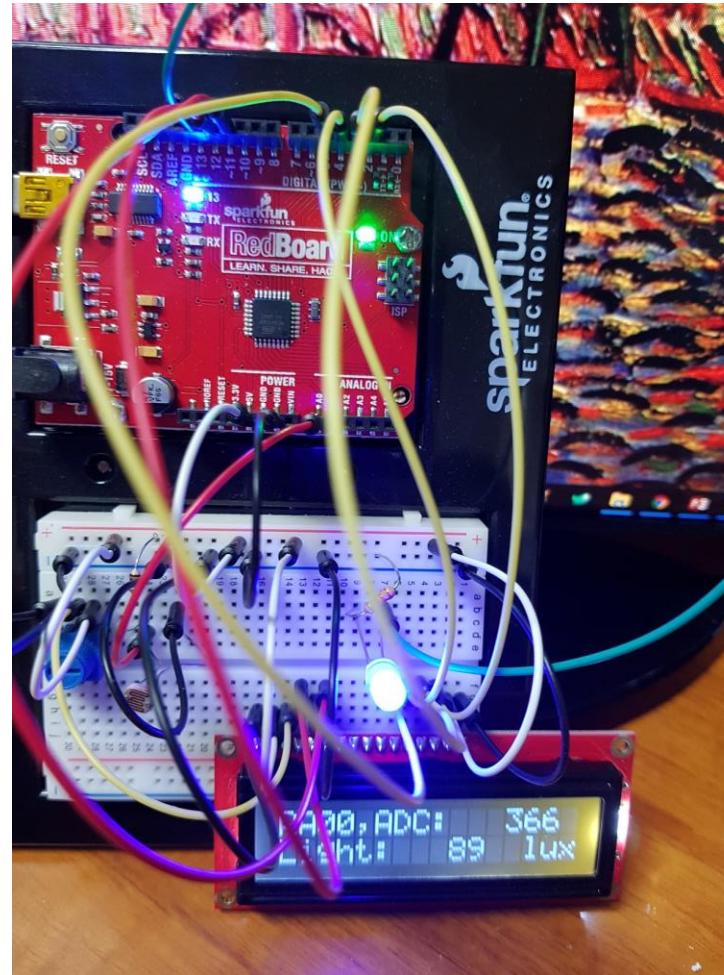


CdS-LCD project : new code

CdS 센서 LCD 회로 - 측정 2

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

LED가 켜진
화면을 폰으로
촬영해서 그림을
제출하시오.



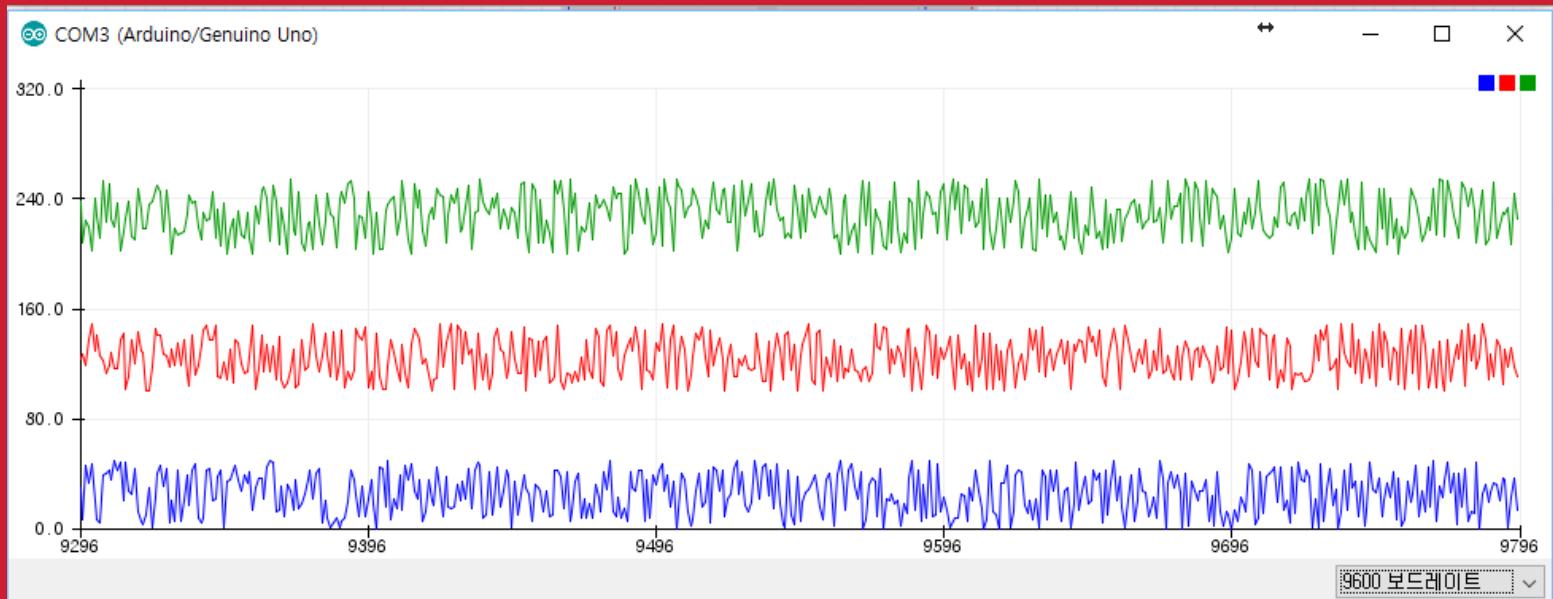
조도에 따라 **LED**가 **ON/OFF** 되는 것을 확인 받고
결과 화면 촬영: AAann LCD lux.png 로

저장...



Arduino

Random Data Simulation





A4.1.1 Random data simulation

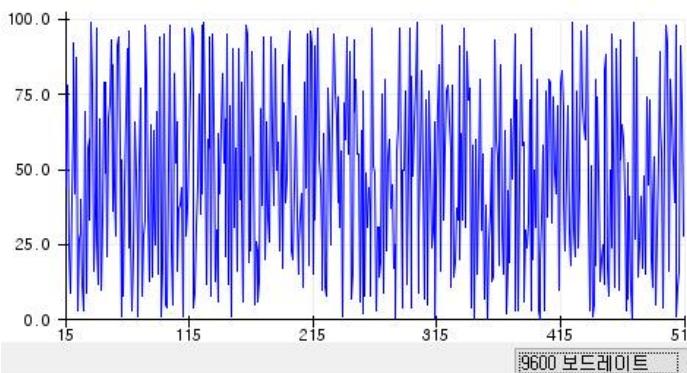
1. Single random signals

random(min, max)

COM3 (Arduino/Genuino Uno)

```
|  
AA00, Current signal: 72.00  
AA00, Current signal: 44.00  
AA00, Current signal: 78.00  
AA00, Current signal: 23.00  
AA00, Current signal: 9.00  
AA00, Current signal: 40.00  
AA00, Current signal: 65.00
```

COM3 (Arduino/Genuino Uno)



```
AA00_Random_signal  
1 /*  
2  Single random signals  
3  
4  Simulation of random signal  
5 */  
6  
7 float sig1=0;  
8  
9 // the setup routine runs once when you press reset:  
10 void setup() {  
11   // initialize serial communication at 9600 bits per second:  
12   Serial.begin(9600);  
13 }  
14  
15 // the loop routine runs over and over again forever:  
16 void loop() {  
17  
18   sig1 = random(0,100);  
19  
20   Serial.print("AA00, Current signal: ");  
21   Serial.println(sig1);  
22   delay(100);           // Sampling rate = 10 Hz  
23 }
```



A4.1.2 Random data simulation – multiple

2. Multiple random signals

AA00_Multi_signals

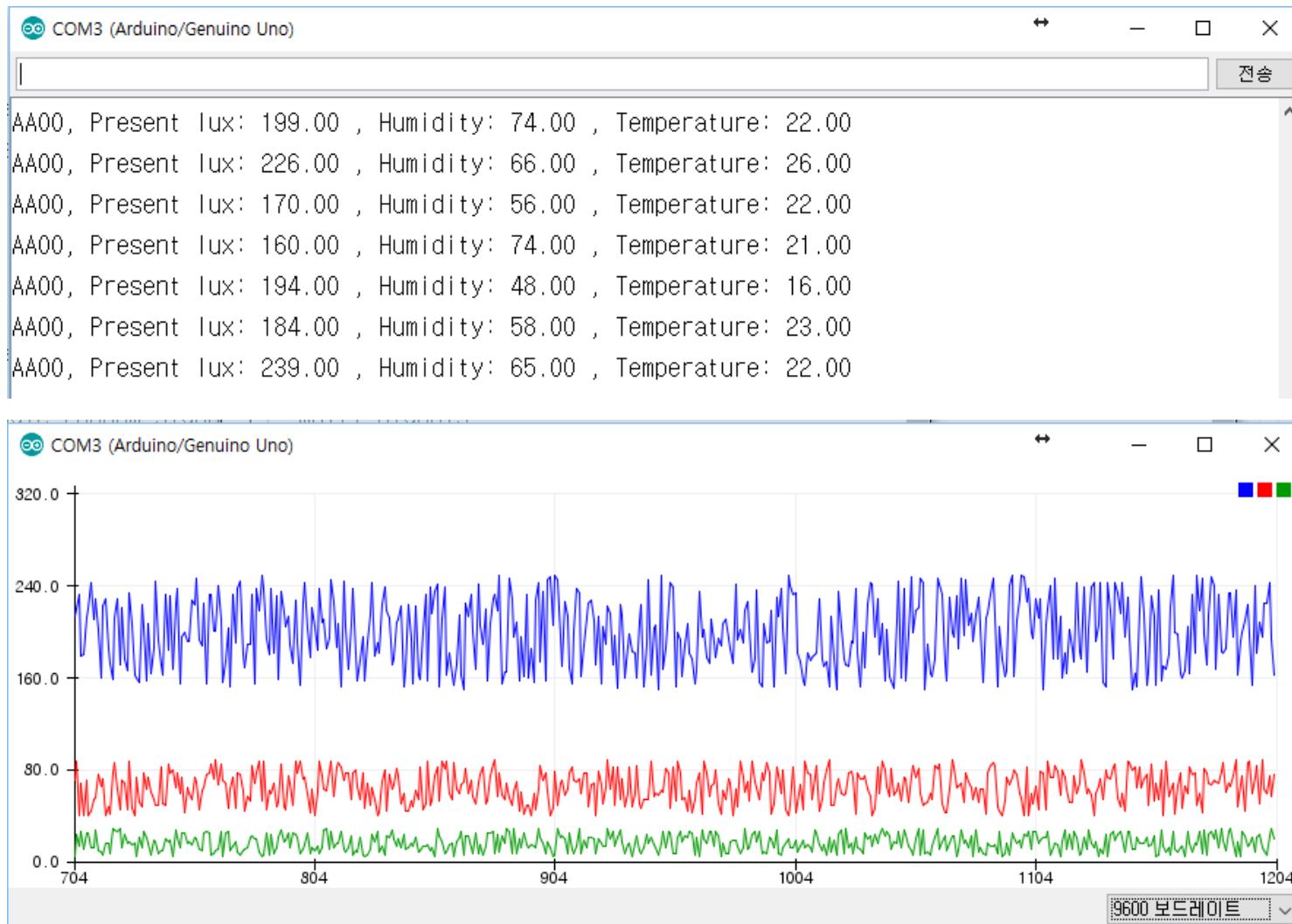
```
1 /*  
2 Multi Signals  
3  
4 Simulation of multiple random signals  
5 */  
6  
7 float humi=0;  
8 float temp=0;  
9 float lux=0;  
10
```

```
11 // the setup routine runs once when you press reset:  
12 void setup() {  
13   // initialize serial communication at 9600 bits per second:  
14   Serial.begin(9600);  
15 }  
16  
17 // the loop routine runs over and over again forever:  
18 void loop() {  
19  
20   // Multi signals  
21   humi = random(40,90);  
22   temp = random(5, 30);  
23   lux = random(150,250);  
24   Serial.print("AA00, Present lux: ");  
25   Serial.print(lux);  
26   Serial.print(" , Humidity: ");  
27   Serial.print(humi);  
28   Serial.print(" , Temperature: ");  
29   Serial.println(temp);  
30   delay(100);      // delay in between reads for stability  
31 }
```



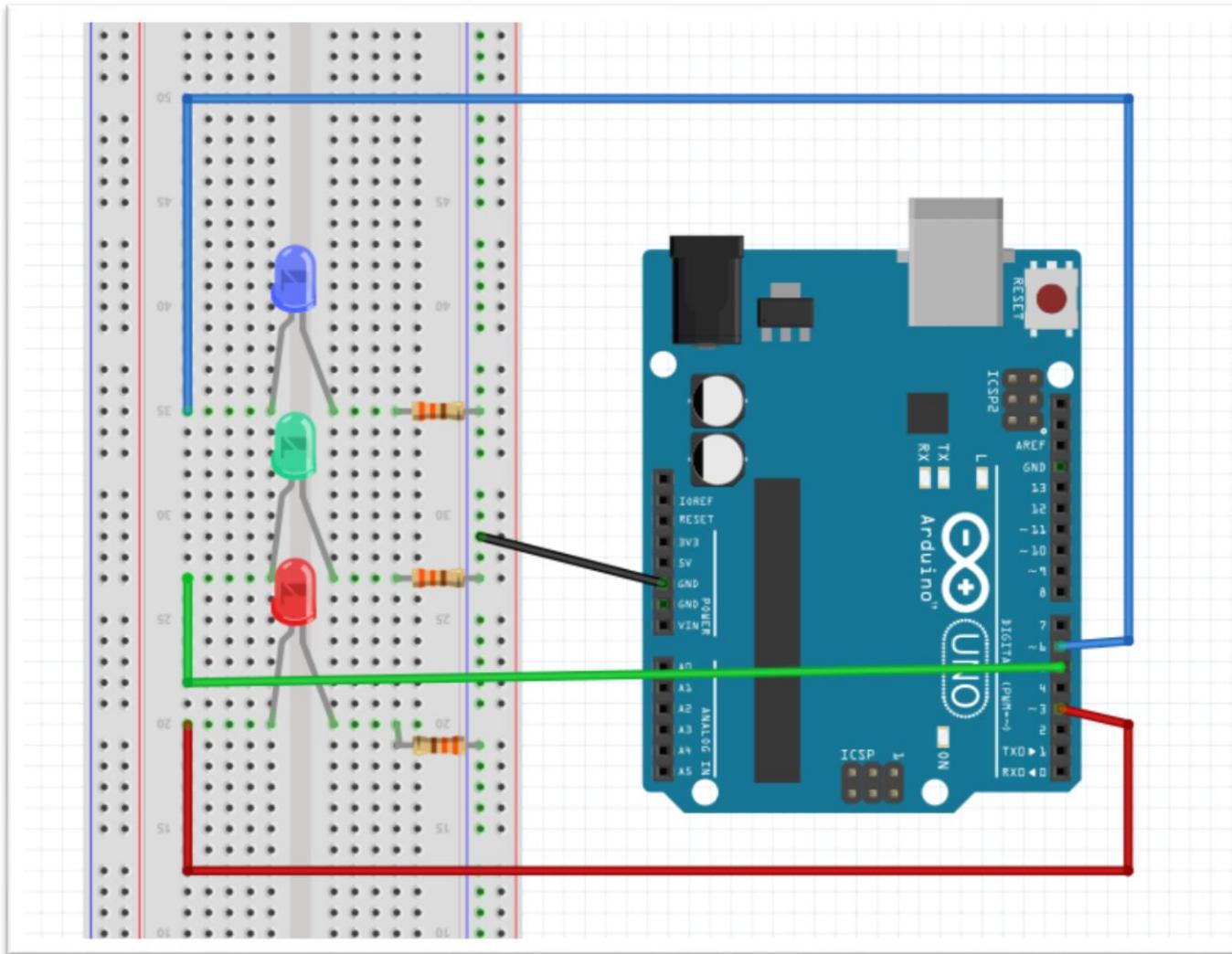
A4.1.2 Random data simulation – multiple

2. Multiple random signals (checkpoint: range of each data)



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.

[참고] LED control – 밝기 조절: PWM

PWM (Pulse Width Modulation)

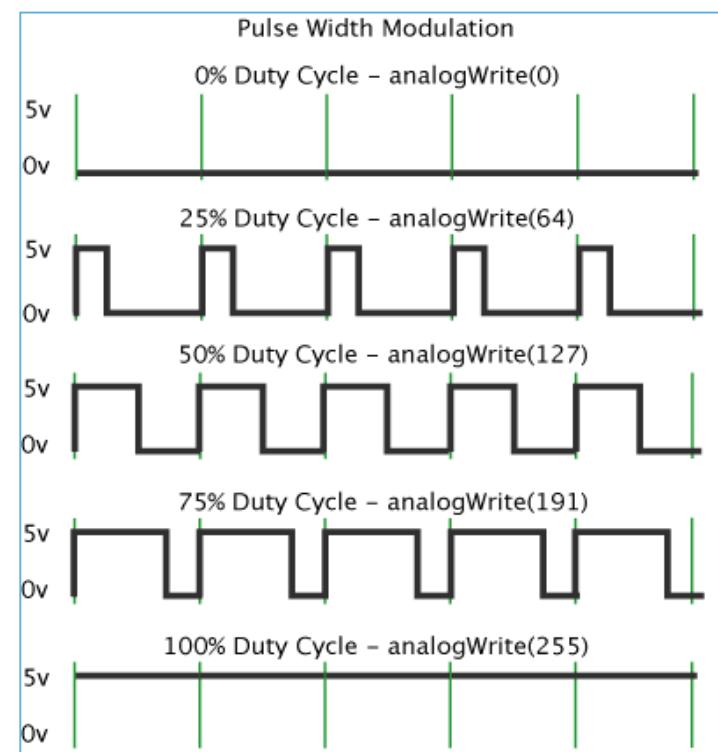
Using [analogWrite\(pin, pwm_value\)](#) function in fading an LED off and on.

AnalogWrite uses [pulse width modulation \(PWM\)](#), turning a digital pin on and off very quickly with different ratio between on and off, to create a fading effect.

A call to [analogWrite\(\)](#) is on a scale of **0 - 255**, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time)

PWM frequency = 500 Hz

<https://www.arduino.cc/en/Tutorial/PWM>

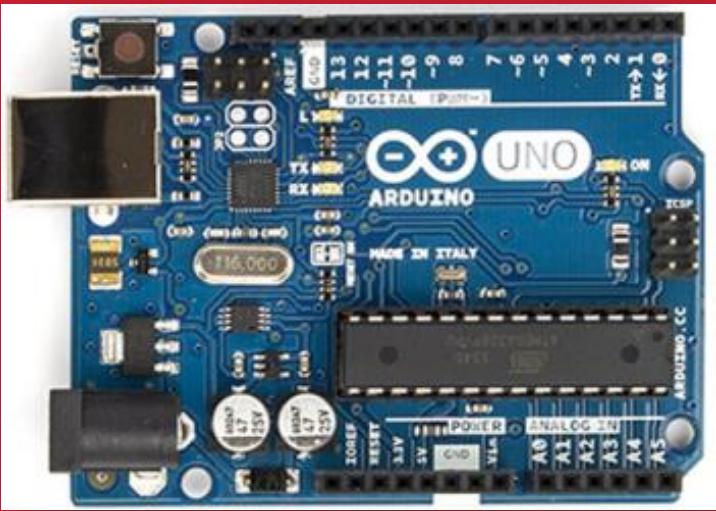




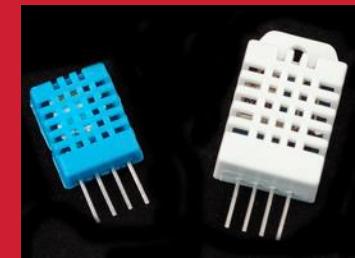
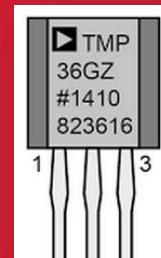
[참고] – 밝기 조절: PWM → code-1

```
1 /*  
2 Dimming 4 leds  
3 */  
4  
5 int ledR = 3; // LED connected to digital pin 3  
6 int ledG = 5;  
7 int ledB = 6;  
8 int ledY = 9;  
9  
10 int dimTime = 20;  
11  
12 void setup() {  
13 // nothing happens in setup  
14 }
```

```
16 void loop() {  
17 // fade in from min to max in increments of 5 points:  
18 for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {  
19 // sets the value (range from 0 to 255):  
20 analogWrite(ledR, fadeValue);  
21 // wait for 30 milliseconds to see the dimming effect  
22 delay(dimTime);  
23 }  
24  
25 // fade out from max to min in increments of 5 points:  
26 for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {  
27 // sets the value (range from 0 to 255):  
28 analogWrite(ledR, fadeValue);  
29 // wait for 30 milliseconds to see the dimming effect  
30 delay(dimTime);  
31 }  
32 // 각 led에 동일한 dimming code 적용  
33 for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {  
34 // sets the value (range from 0 to 255):  
35 analogWrite(ledG, fadeValue);  
36 // wait for 30 milliseconds to see the dimming effect  
37 delay(dimTime);  
38 }
```

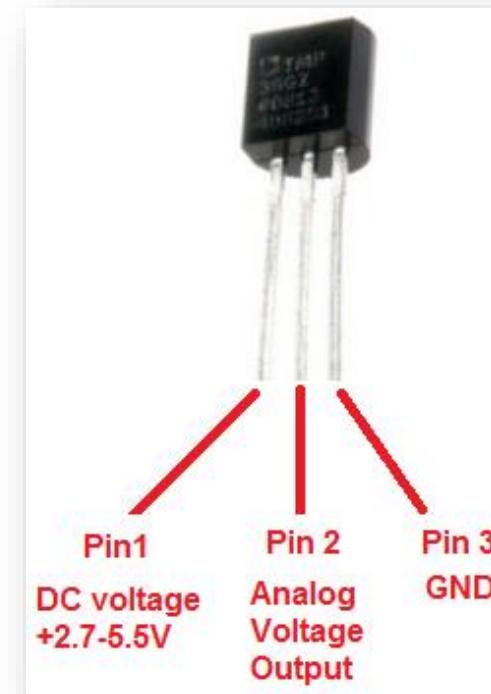
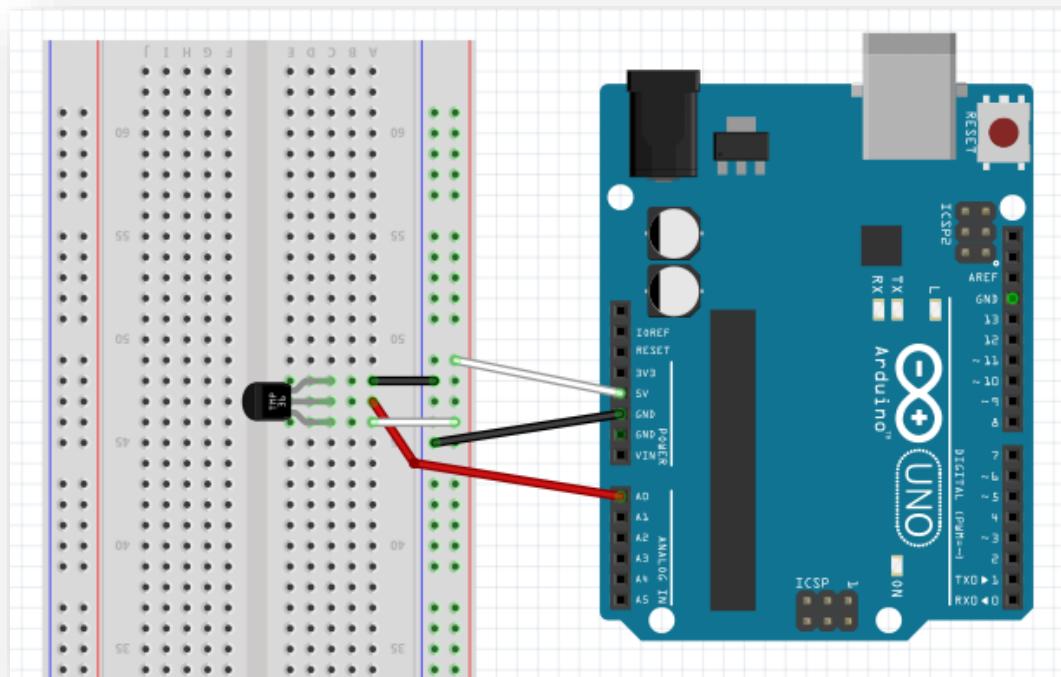


Arduino & Node.js

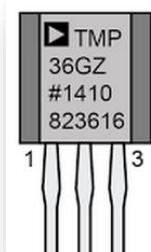




A3.1.2 Temperature sensor [TMP36]



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



A3.1.3 Temperature sensor [TMP36]

Simple code

```
TMP36$  
1 //  
2 // AA00, TMP36 sensor  
3 //  
4  
5 #define TEMP_INPUT 0  
6 // or int TEMP_INPUT = 0;  
7  
8 void setup() {  
9   Serial.begin(9600);  
10 }  
11  
12 void loop() {  
13  
14   int value = analogRead(TEMP_INPUT);  
15   Serial.println(value);  
16  
17   delay(1000);  
18 }
```

Serial output (0 ~ 1023)

The screenshot shows the Arduino Serial Monitor window titled "COM8 (Arduino/Genuino Uno)". The window displays a series of temperature readings in degrees Celsius, ranging from 139 to 141. The text area contains the following data:

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

At the bottom of the window, there are several status indicators: a checked checkbox for "자동 스크롤" (Auto Scroll), a dropdown menu for "No line ending", and a text input field for "9600 보드 레이트" (Board Rate).

A3.1.4 Temperature sensor [TMP36]

Sensor property

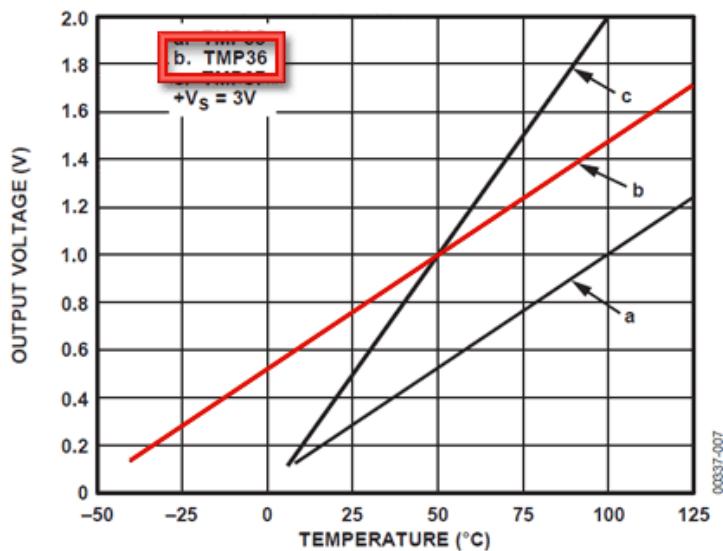


Figure 6. Output Voltage vs. Temperature

Temperature conversion

$$\text{Temp } (\text{° C}) = (\text{Vout} - 500) / 10$$

$$\text{Vout (mV)} = \text{value} * (5000 / 1023)$$
$$(0 \leq \text{value} \leq 1023)$$



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 }
```

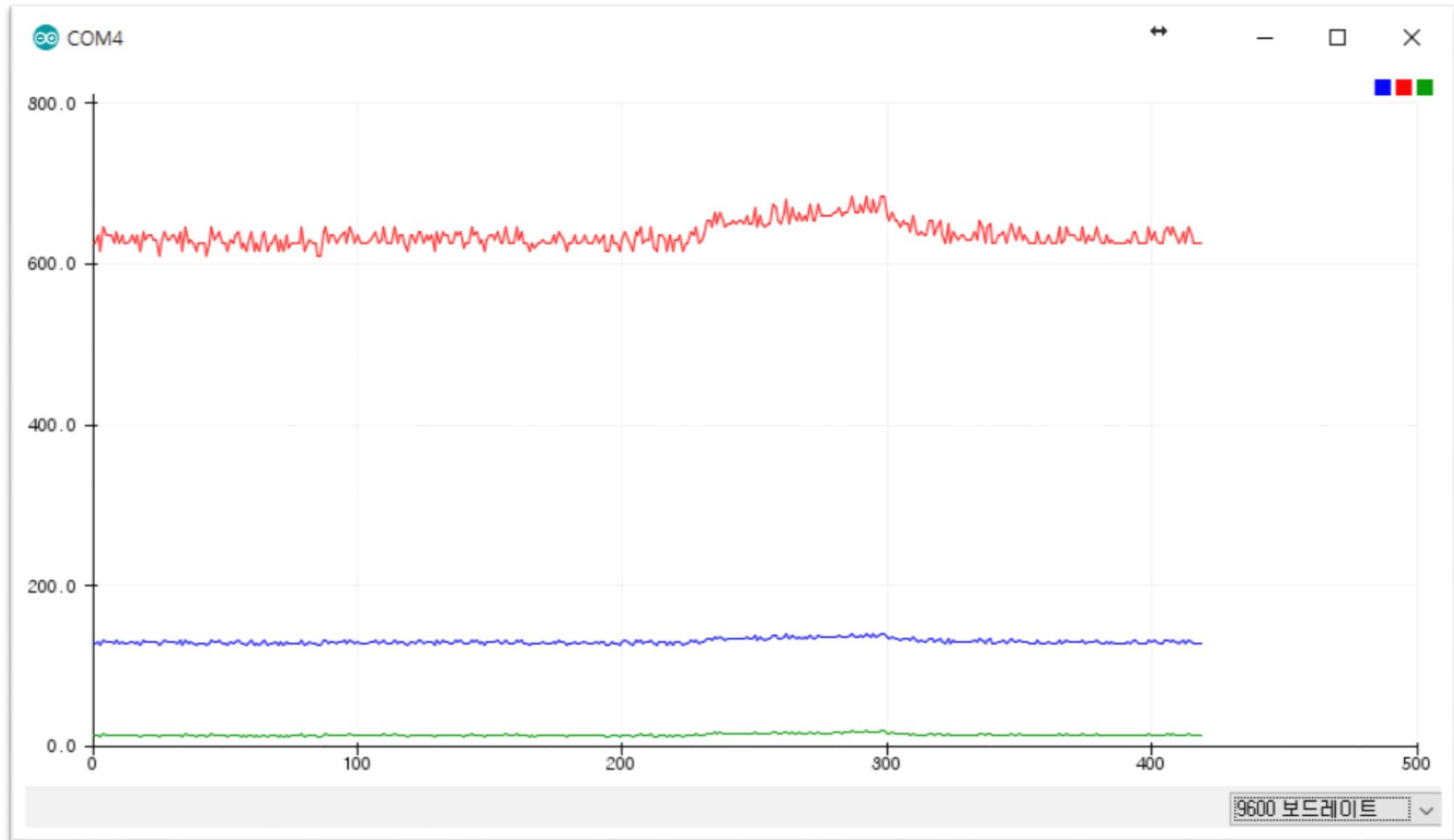
Serial output (°C)

The screenshot shows the Arduino Serial Monitor window titled "COM4". It displays a series of temperature readings in Celsius, each preceded by the string "AA00, value = ". The data is as follows:

```
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 = 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]





A5.1.1 tmp36 node project

Start tmp36-node project

1. Go to my working folder
2. md tmp36
3. cd tmp36
4. dir

```
ca: NodeJS
D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data 디렉터리

2017-11-01 오전 10:10 <DIR> .
2017-11-01 오전 10:10 <DIR> ..
2016-02-26 오전 04:36 <DIR> 6,148 .DS_Store
2017-10-18 오전 10:15 <DIR> 332 aa00
2017-09-13 오후 12:18 <DIR> 240 express
2017-09-13 오후 12:18 <DIR> 240 express.cmd
2017-09-15 오후 04:41 <DIR> 112 myApp
2017-09-13 오후 12:18 <DIR> 112 node_modules
2015-10-15 오전 07:21 <DIR> 296 npm
2015-10-15 오전 07:21 <DIR> 204 npm.cmd
2017-09-15 오후 04:32 <DIR> 204 npm_cache
2017-11-01 오전 10:10 <DIR> 89 PortableApps.comLau
e.ini
2017-09-06 오후 01:40 <DIR> 7,309 바이트
    6개 파일
    7개 디렉터리 910,031,147,008 바이트 남음

D:\Portable\NodeJSPortable\Data>cd aa00

D:\Portable\NodeJSPortable\Data\aa00>md tmp36

D:\Portable\NodeJSPortable\Data\aa00>cd tmp36

D:\Portable\NodeJSPortable\Data\aa00\tmp36>dir
D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data\aa00\tmp36 디렉터리

2017-11-01 오전 10:39 <DIR> .
2017-11-01 오전 10:39 <DIR> ..
    0개 파일
    2개 디렉터리 0 바이트
    910,031,147,008 바이트 남음

D:\Portable\NodeJSPortable\Data\aa00\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 }
```



A5.1.4 tmp36 node project

AAnn_TMP36_NodeJS.ino

```
AA00_TMP36_NodeJS
11
12 void loop() {
13     //getting the voltage reading from the temperature sensor
14     int value = analogRead(TEMP_INPUT);
15     Serial.print("AA00, value = ");
16     Serial.println(value);
17     // Serial.print(" : ");
18     //
19     //    // converting that reading to voltage
20     //    float voltage = value * 5.0 * 1000; // in mV
21     //    voltage /= 1023.0;
22     //
23     //    // print out the voltage
24     //    Serial.print(voltage);
25     //    Serial.print(" mV, ");
26     //
27     //    // now print out the temperature
28     //    float temperatureC = (voltage - 500) / 10 ;
29     //    Serial.print(" Temperature, ");
30     //    Serial.print(temperatureC);
31     //    Serial.println(" degrees C");
32
33     delay(1000);
34 }
```

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
```



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 ...



Error & Bug ---

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



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

전체

동영상

뉴스

이미지

더보기

설정

도구

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](https://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



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"



A5.1.8 tmp36 node project (after downgrade)

tmp36_node.js

```
tmp36
├── node_modules
└── /* client.js
    /* package.json
    /* package_new.json
    /* tmp36_node.js

12   dataBits: 8,
13   parity: 'none',
14   stopBits: 1,
15   flowControl: false,
16   parser: serialport.
17 }::

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 output (A0 in node)

```
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
```



A5.1.9 tmp36 node project (all message)

AAnn_TMP36_NodeJS.ino

```
AA00_TMP36_NodeJS
11
12 void loop() {
13     //getting the voltage reading from the temperature sensor
14     int value = analogRead(TEMP_INPUT);
15     Serial.print("AA00, value = ");
16     Serial.print(value);
17     Serial.print(" : ");
18
19     // converting that reading to voltage
20     float voltage = value * 5.0 * 1000; // in mV
21     voltage /= 1023.0;
22
23     // print out the voltage
24     Serial.print(voltage);
25     Serial.print(" mV, ");
26
27     // now print out the temperature
28     float temperatureC = (voltage - 500) / 10 ;
29     Serial.print(" Temperature, ");
30     Serial.print(temperatureC);
31     Serial.println(" degrees C");
32
33     delay(1000);
34 }
```

Serial output

```
AA00, value = 128 : 625.61 mV, Temperature, 12.56 degrees C
AA00, value = 132 : 645.16 mV, Temperature, 14.52 degrees C
AA00, value = 128 : 625.61 mV, Temperature, 12.56 degrees C
AA00, value = 128 : 625.61 mV, Temperature, 12.56 degrees C
AA00, value = 129 : 630.50 mV, Temperature, 13.05 degrees C
AA00, value = 130 : 635.39 mV, Temperature, 13.54 degrees C
AA00, value = 131 : 640.27 mV, Temperature, 14.03 degrees C
AA00, value = 127 : 620.72 mV, Temperature, 12.07 degrees C
AA00, value = 129 : 630.50 mV, Temperature, 13.05 degrees C
AA00, value = 128 : 625.61 mV, Temperature, 12.56 degrees C
AA00, value = 128 : 625.61 mV, Temperature, 12.56 degrees C
```

AAnn_tmp36_message.png
로 저장



A5.1.10 tmp36 node project (only data)

AAnn_TMP36_NodeJS.ino

```
AA00_TMP36_NodeJS
11
12 void loop() {
13     //getting the voltage reading from the temperature sensor
14     int value = analogRead(TEMP_INPUT);
15     // Serial.print("AA00, value = ");
16     // Serial.print(value);
17     // Serial.print(" : ");
18
19     // converting that reading to voltage
20     float voltage = value * 5.0 * 1000; // in mV
21     voltage /= 1023.0;
22
23     // print out the voltage
24     // Serial.print(voltage);
25     // Serial.print(" mV, ");
26
27     // now print out the temperature
28     float temperatureC = (voltage - 500) / 10 ;
29     // Serial.print(" Temperature, ");
30     Serial.println(temperatureC);
31     // Serial.println(" degrees C");
32
33     delay(1000);
34 }
```

Serial output

COM4

13.54
11.58
12.56
12.56
12.56
12.56
14.03
11.58
13.54
13.54
12.56
11.58

12.56
12.56
13.05
12.56
12.56
12.56
15.49
13.54
13.54
14.52
15.00
16.47
15.49



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



[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**

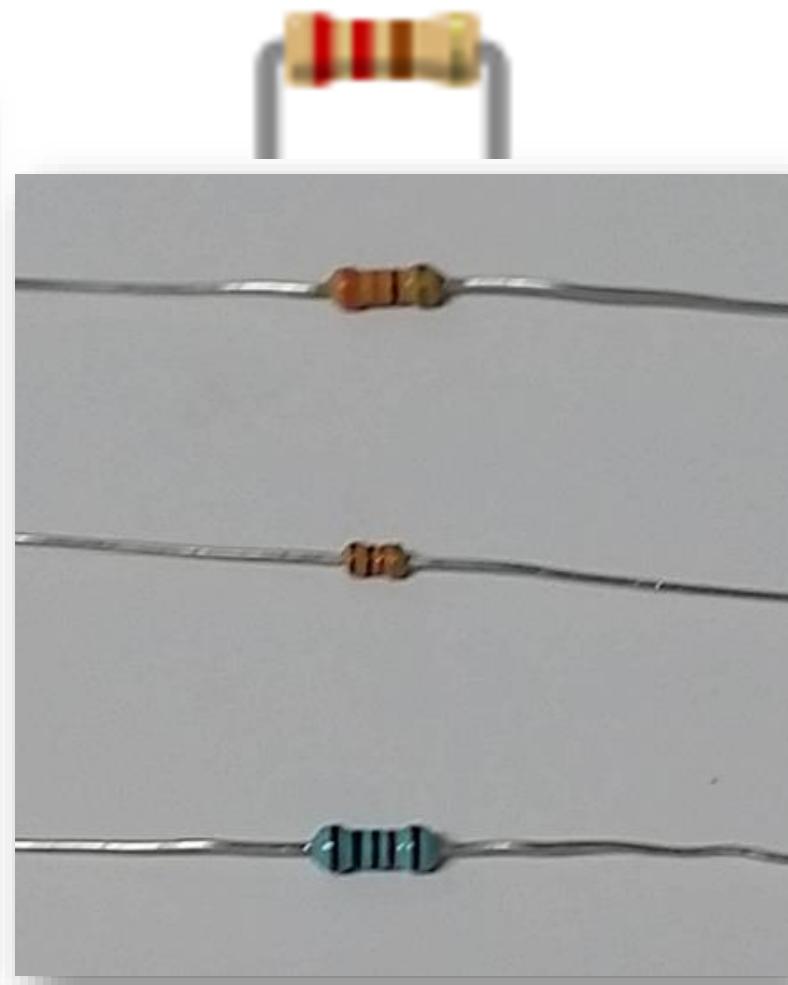


[참고 : 저항 값 읽기]

Carbonfilm resistor

sm2k (c) 2006

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%

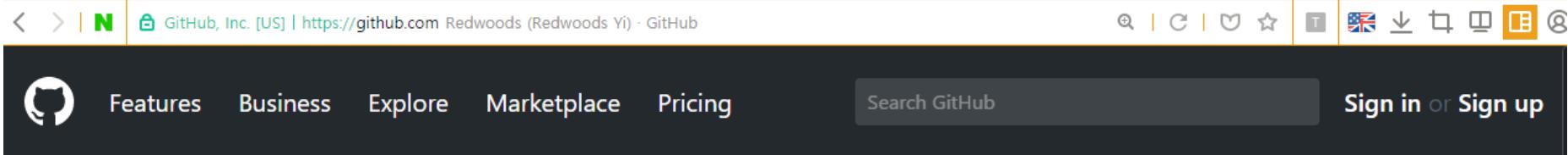


Lecture materials

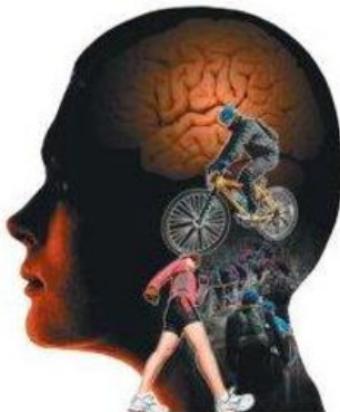


● References & good sites

- ✓ <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. To the right is a search bar labeled "Search GitHub" and buttons for "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

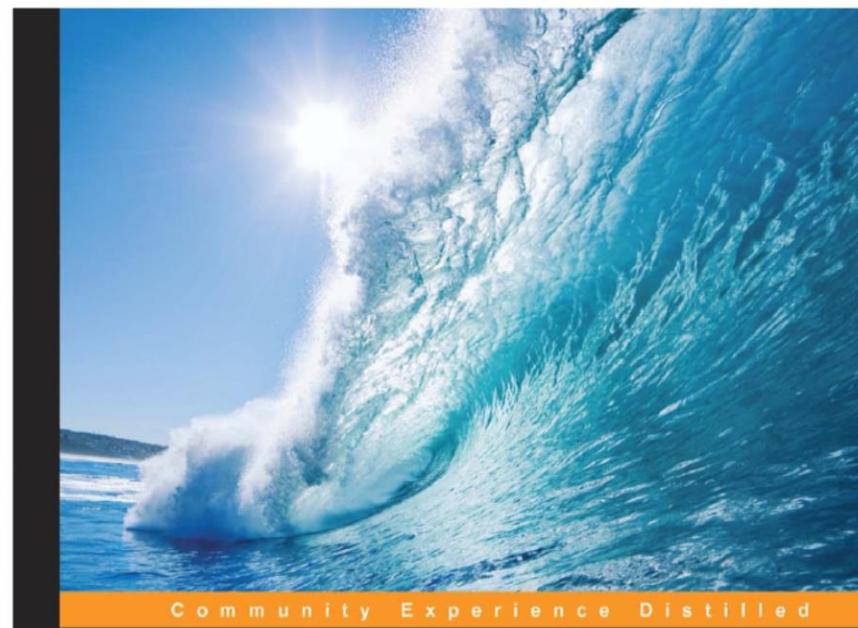
[hw-coding](#)

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

 Arduino

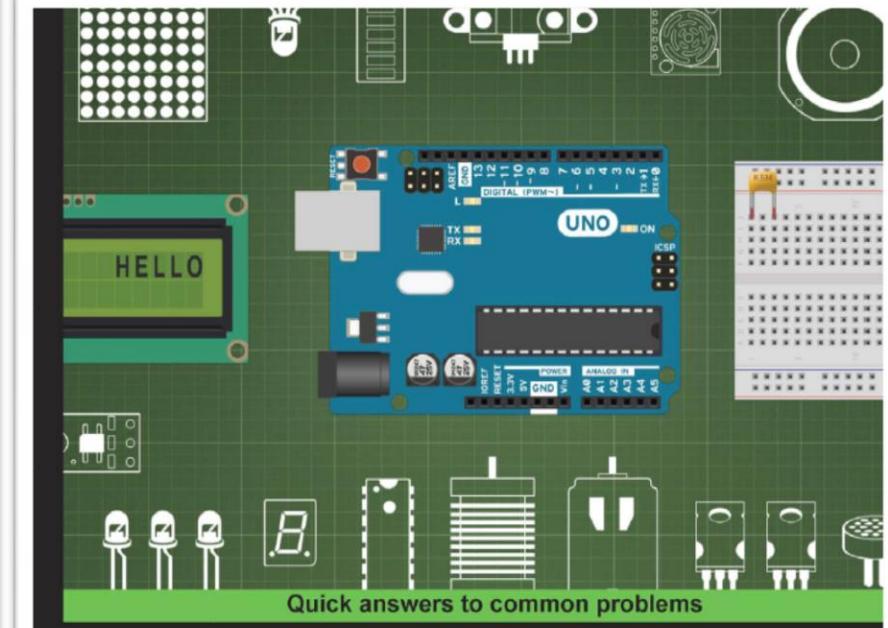


References



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

www.allaboutcircuits.com