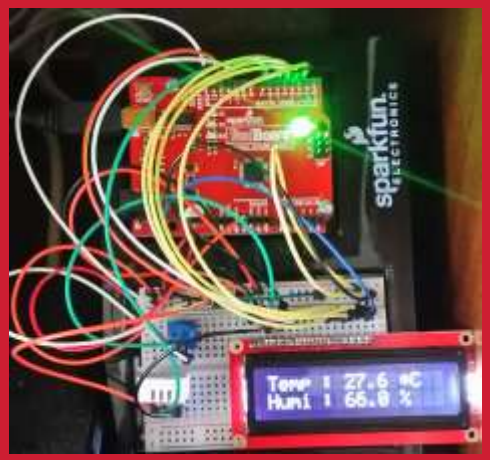
Arduino-IOT

[wk10]



Arduino + Node
Data visualization I

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

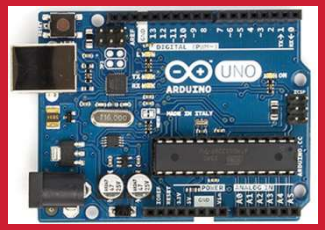
Comsi, INJE University

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]

- [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**

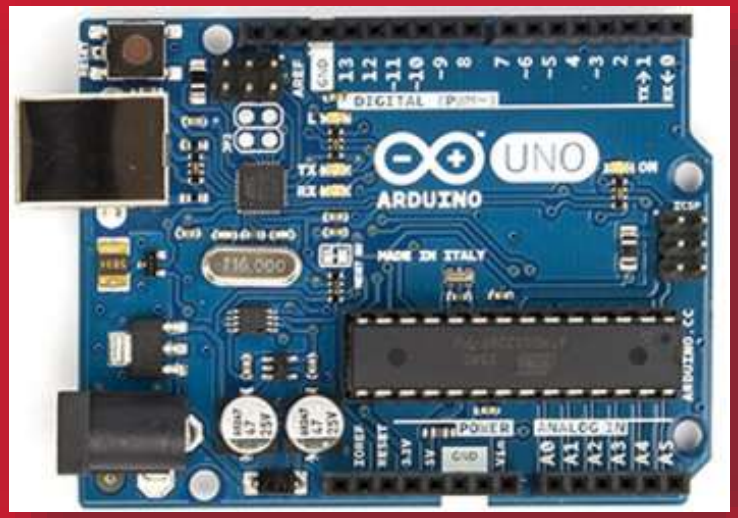
③ **AAnn_cds_IOT_data.png**

④ **AAnn_cds_tmp36_lcd.png**

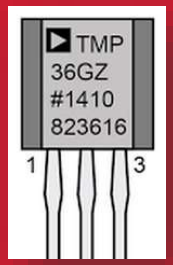
⑤ **AAnn_cds_tmp36_IOT.png**

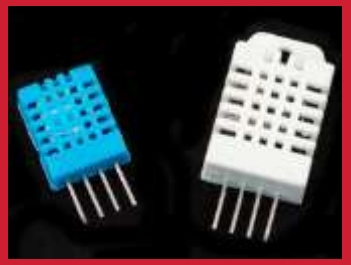
Email : chaos21c@gmail.com

[제목 : id, 이름 (수정)]



Arduino & Node.js





IOT: HSC

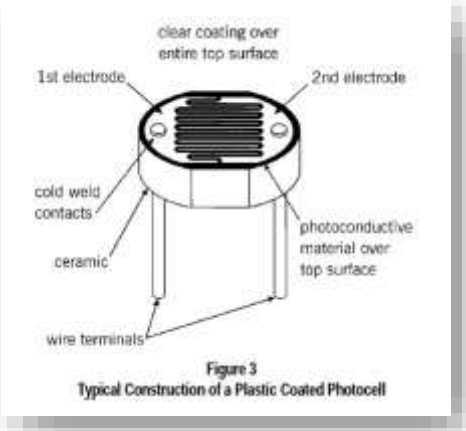
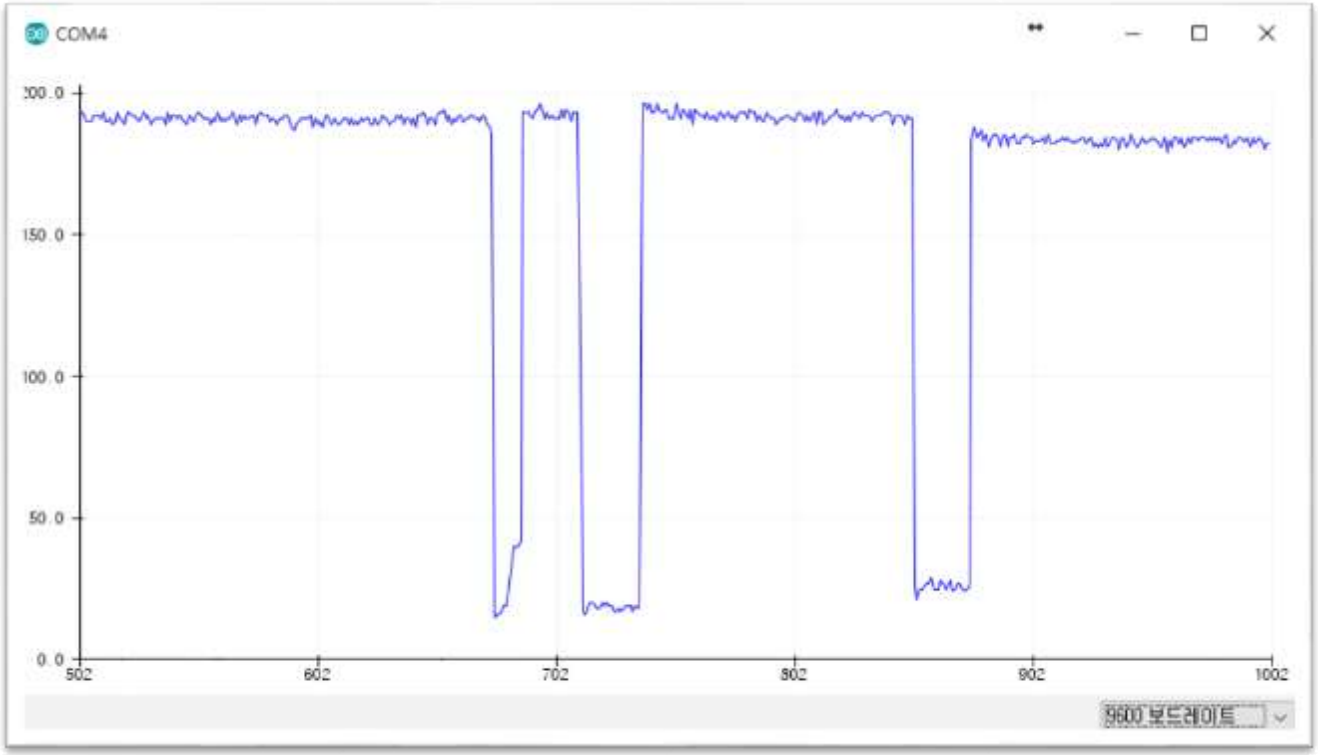
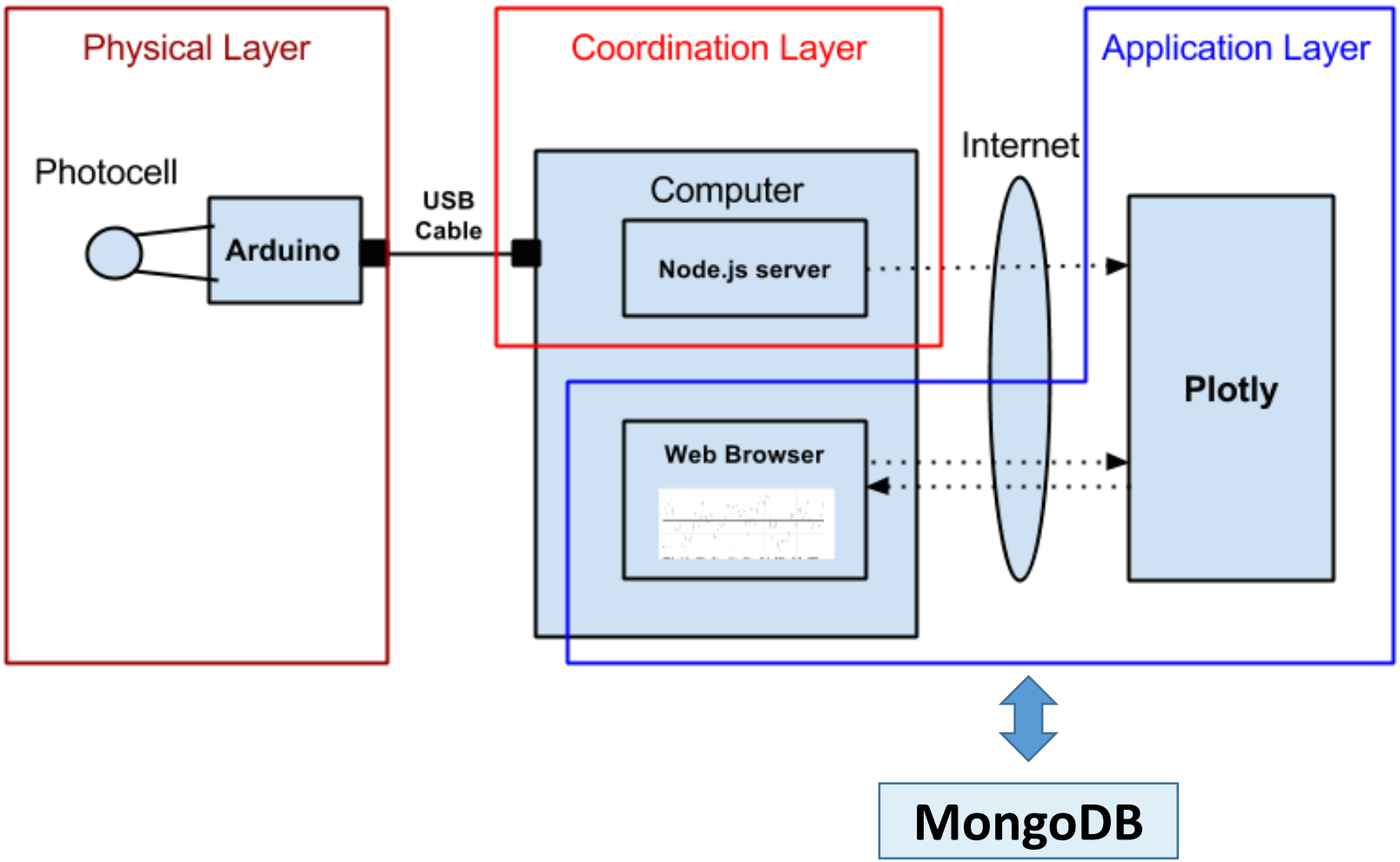


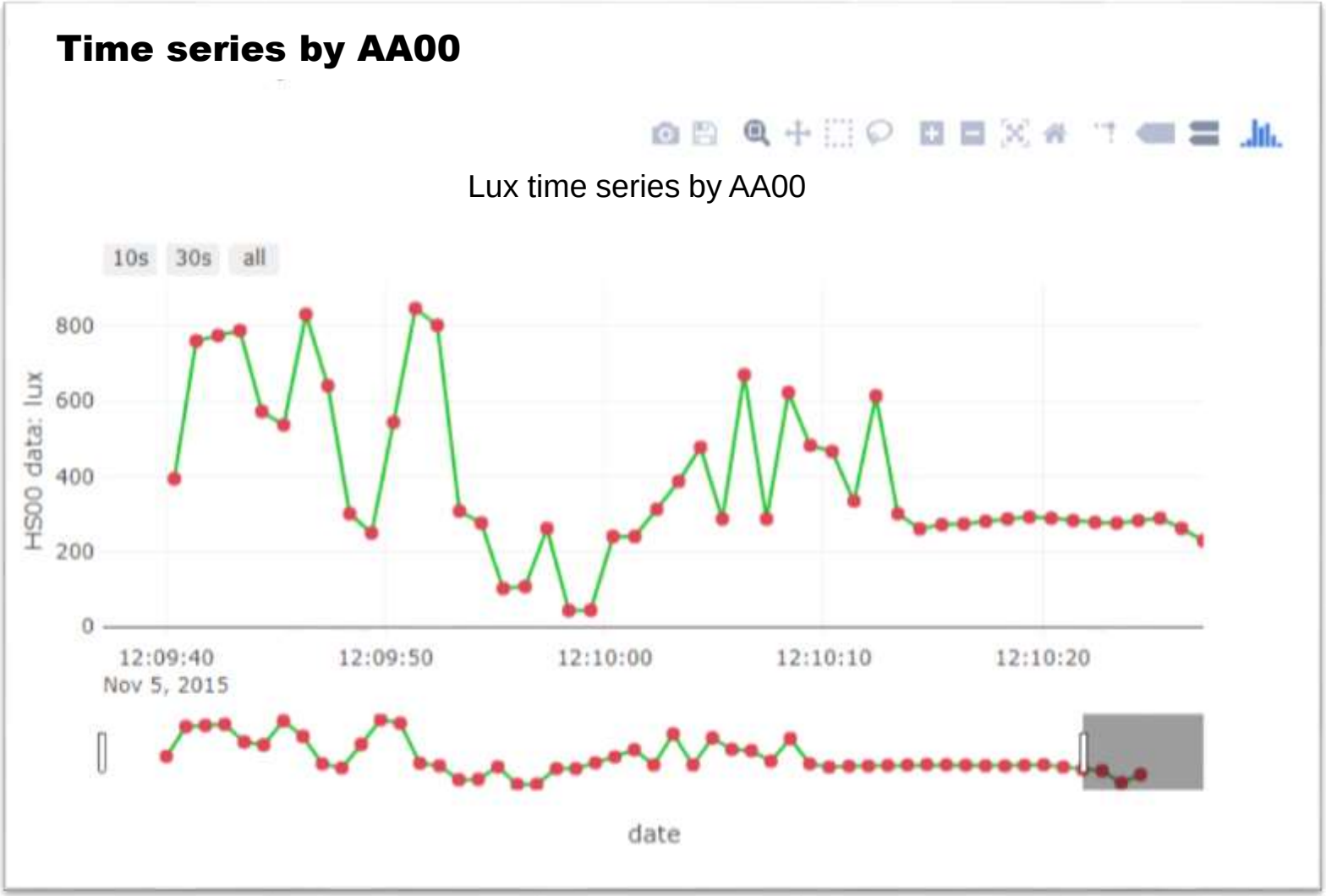
Figure 3
Typical Construction of a Plastic Coated Photocell



Layout [H S C]

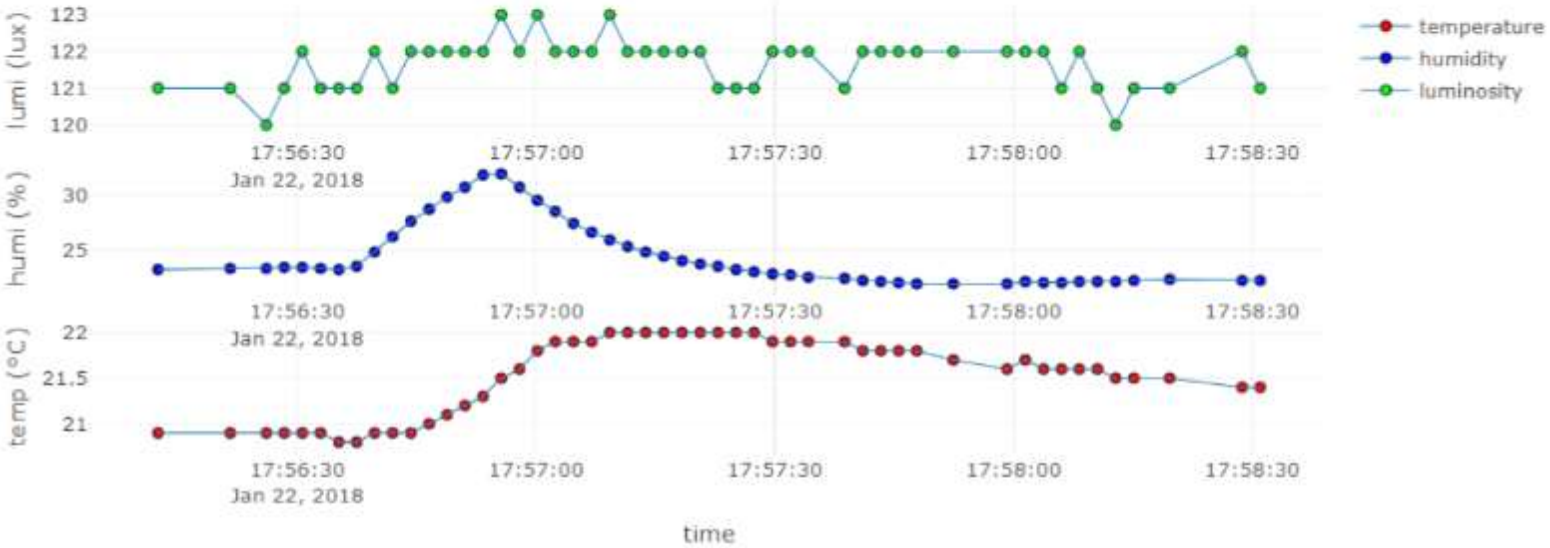

Physical Layer
Photocell
Arduino
USB Cable
Coordination Layer
Computer
Node.js server
Web Browser
Internet
Application Layer
Plotly
MongoDB

Arduino data + plotly

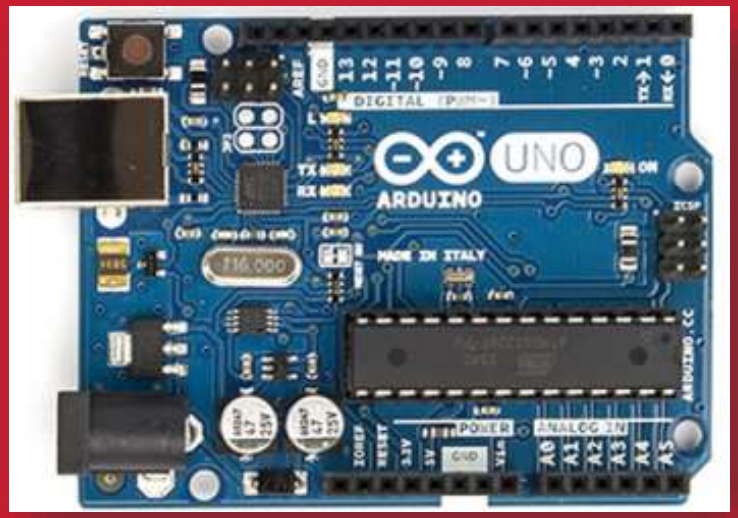


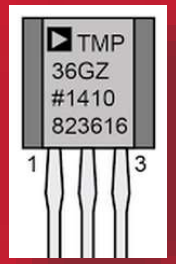
Real-time Weather Station from sensors

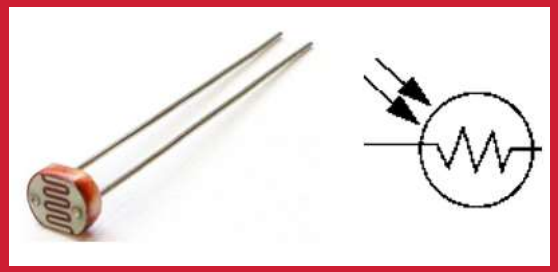
on Time: 2018-01-22 17:58:31.012

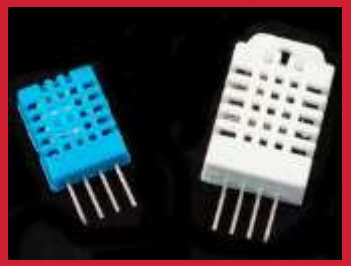


Arduino Sensors + Node.js

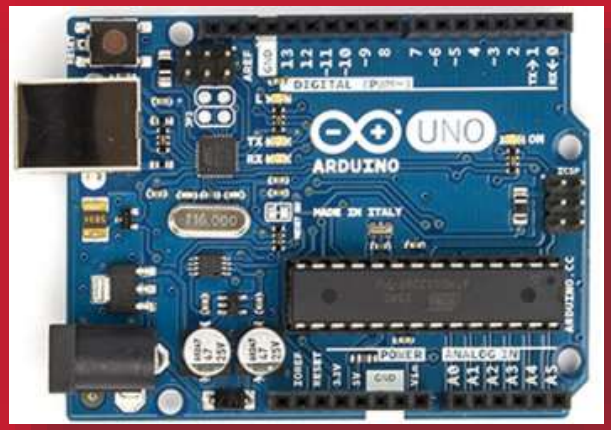








Single sensor: tmp36



TMP36

Node project

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
```

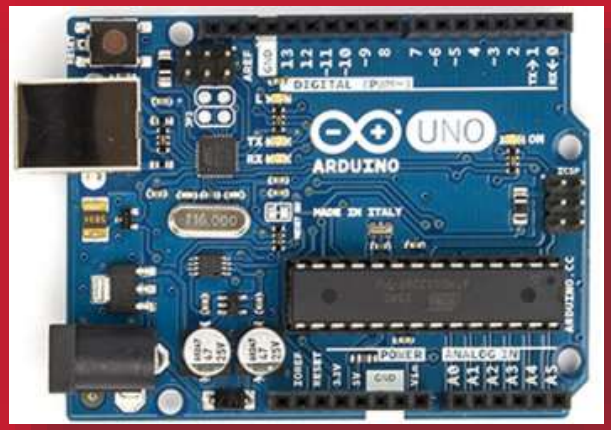
▶ Node cmd에서 실행

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



CdS (LDR)

Node project

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

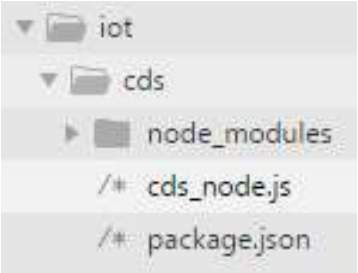
package,json

```
{
  "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"
  }
}
```

A4.2.5 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
```



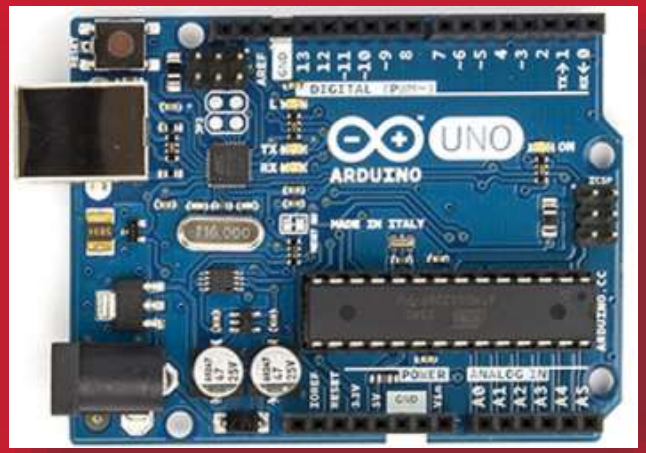
▶ Node cmd에서 실행

```
node cds_node
```

```
NodeJS - 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
```

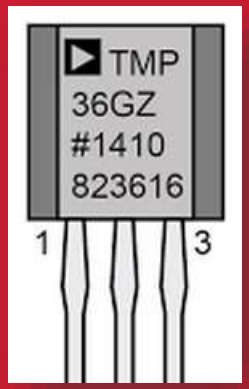
AAnn_cds_IOT_data.png
로 저장

Multiple sensors

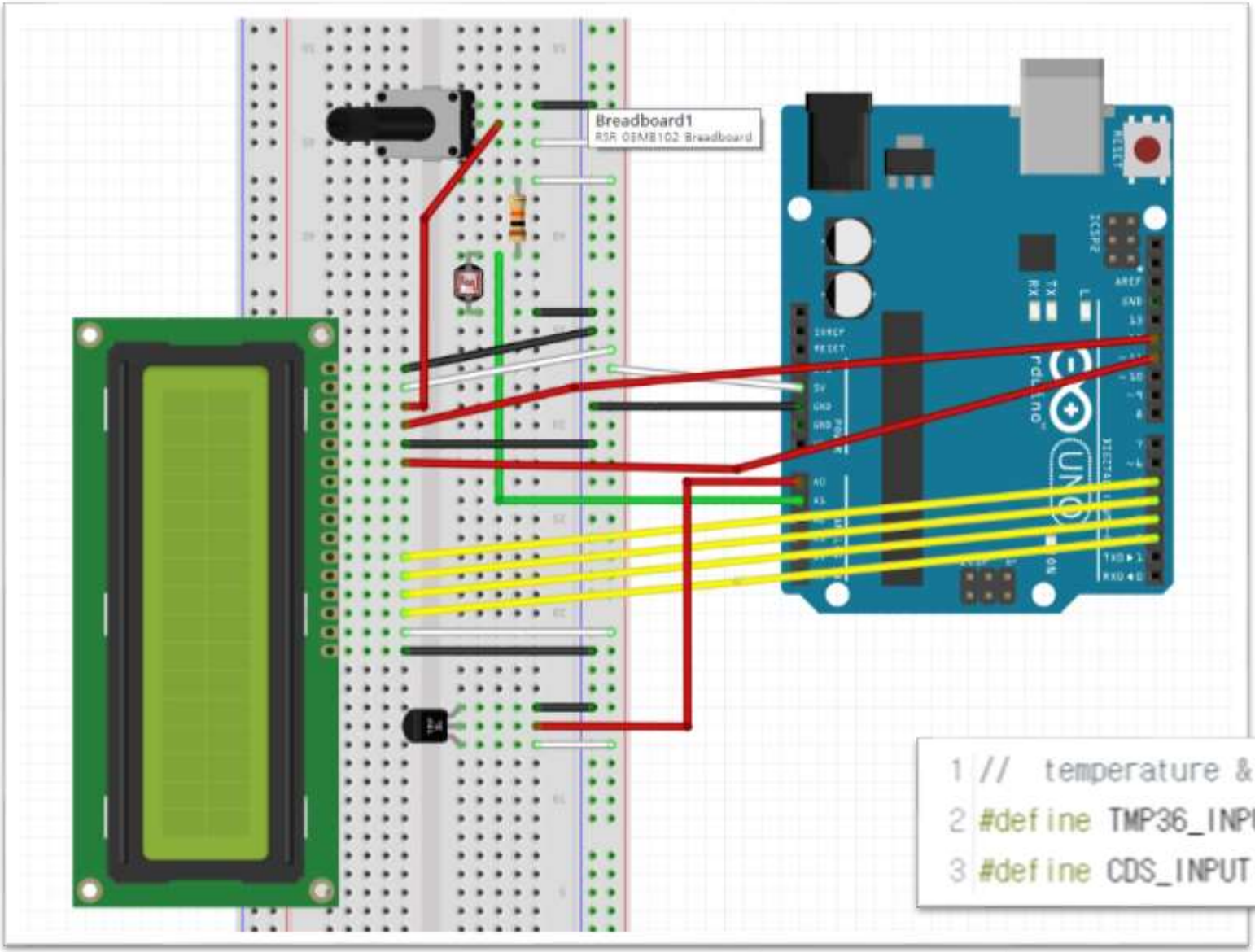


CdS + TMP36

Node project



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 라이브러리 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

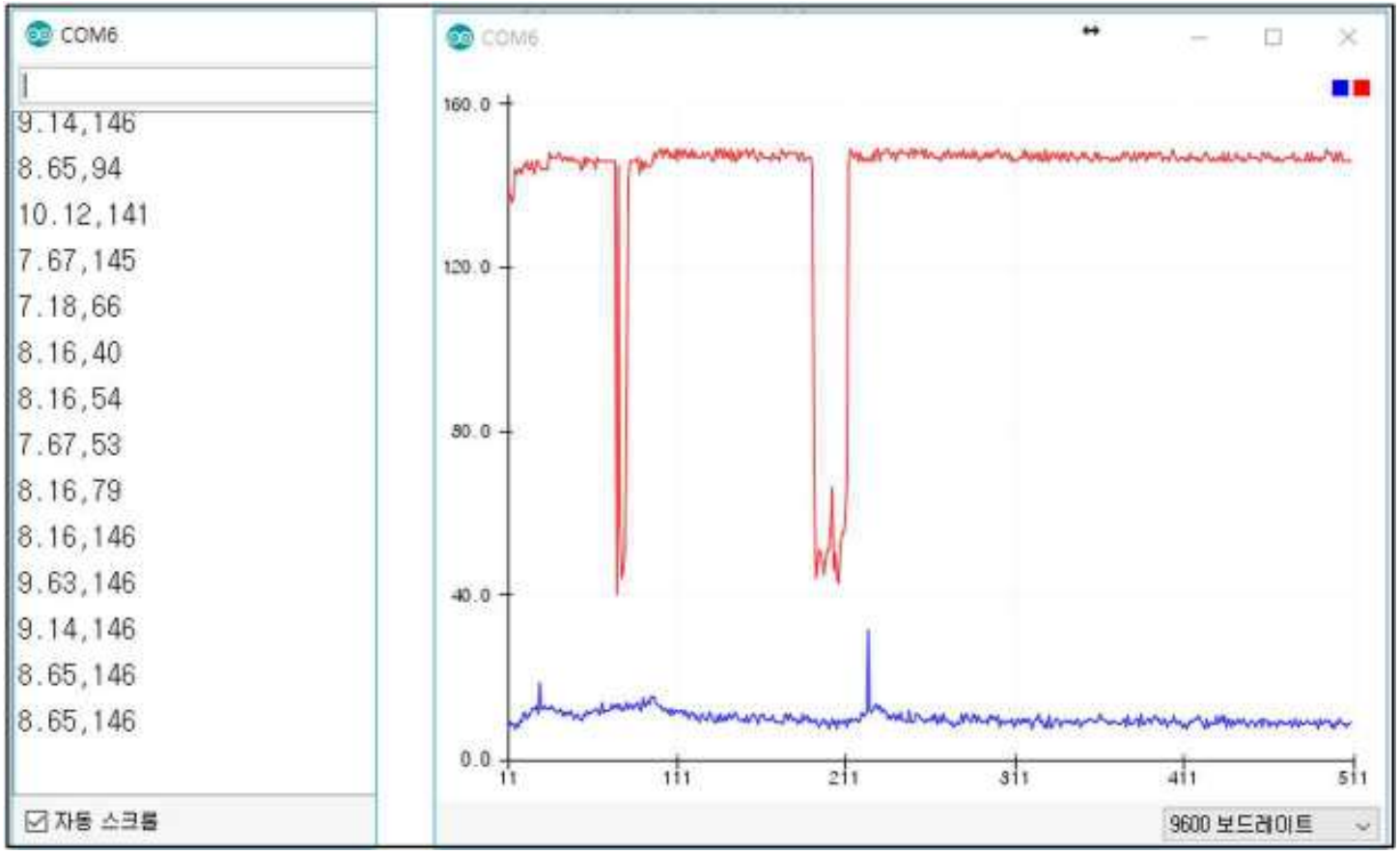
```
29 void loop(){
30   // Temperature from TMP36
31   int temp_value = analogRead(TMP36_INPUT);
32   // converting that reading to voltage
33   float voltage = temp_value * 5.0 * 1000;  // in mV
34   voltage /= 1023.0;
35   float tempC = (voltage - 500) / 10 ;
36
37   // Lux from CdS (LDR)
38   int cds_value = analogRead(CDS_INPUT);
39   int lux = int(luminosity(cds_value));
40
41   // 전에 표시했던 내용을 지운다.
42   lcd.setCursor(12,0);
43   lcd.print("    ");
44   // 온도를 표시한다
45   lcd.setCursor(12,0);
46   lcd.print(tempC);
47   // 전에 표시했던 내용을 지운다.
48   lcd.setCursor(9,1);
49   lcd.print("   ");
50   // 조도를 표시한다
51   lcd.setCursor(9,1);
52   lcd.print(lux);
```

LCD output

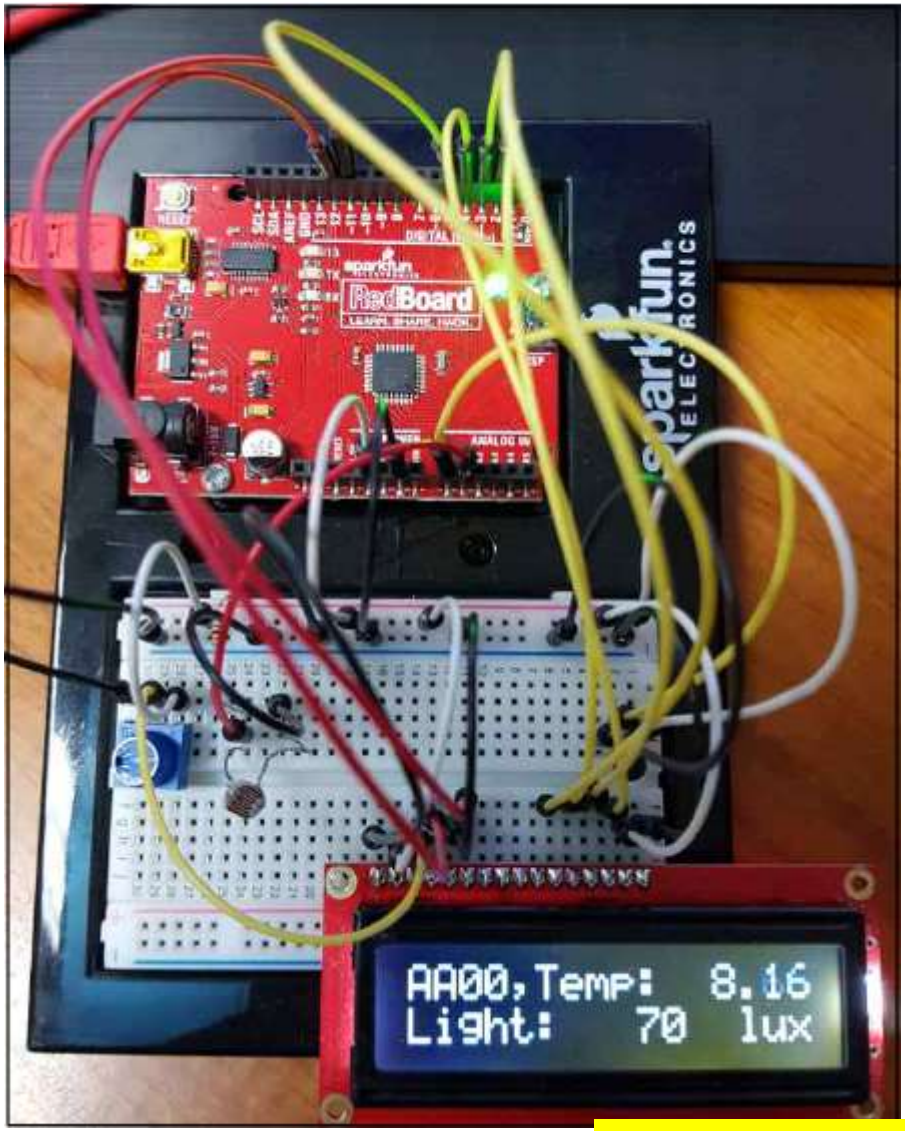
```
54   // Serial output --> 온도,조도
55   Serial.print(tempC);
56   Serial.print(",");
57   Serial.println(lux);
58   delay(1000);
59 }
60
61 //Voltage to Lux
62 double luminosity (int RawADC0){
63   double Vout=RawADC0*5.0/1023.0;  // 5/1023 (Vin = 5 V)
64   double lux=(2500/Vout-500)/10.0;
65   // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
66   return lux;
67 }
```

Serial output

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



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



Save as
AAnn_cds_tmp36_lcd.png

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

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"
  }
}
```

A4.5.5.1 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

A4.5.6 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

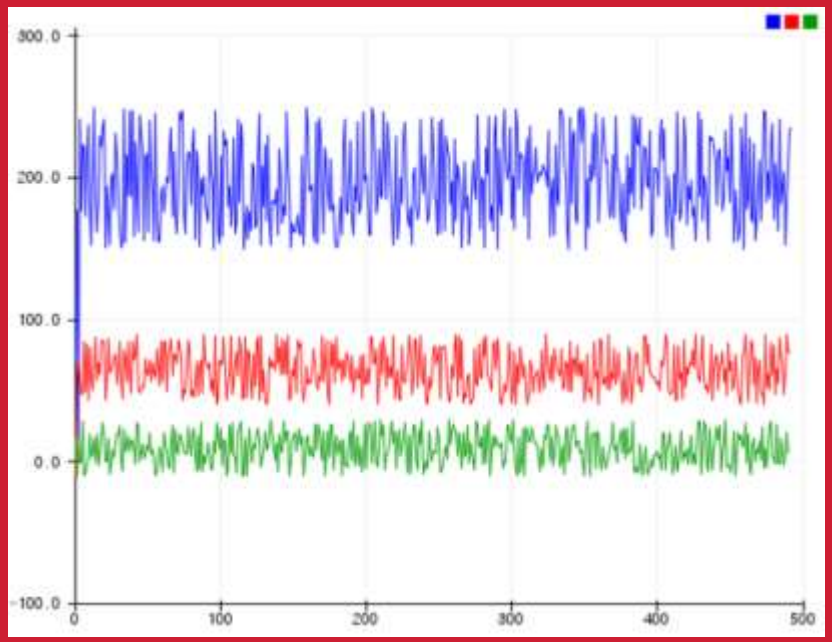
시간, 온도,조도

Save as
AAnn_cds_tmp36_IOT.png

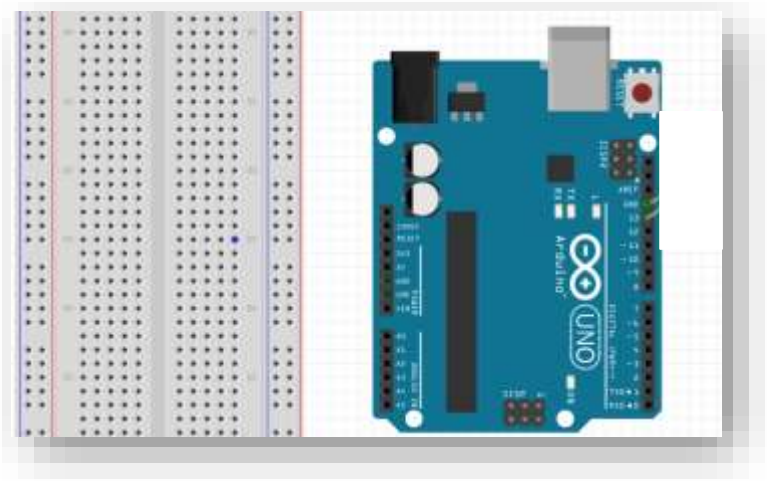
[DIY] Multi-signals

다중신호 시뮬레이션

+ node.js



DIY – 스케치



아두이노에서 **LED**와 저항을 모두 제거하고 **USB**만 컴퓨터와 연결한다.

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

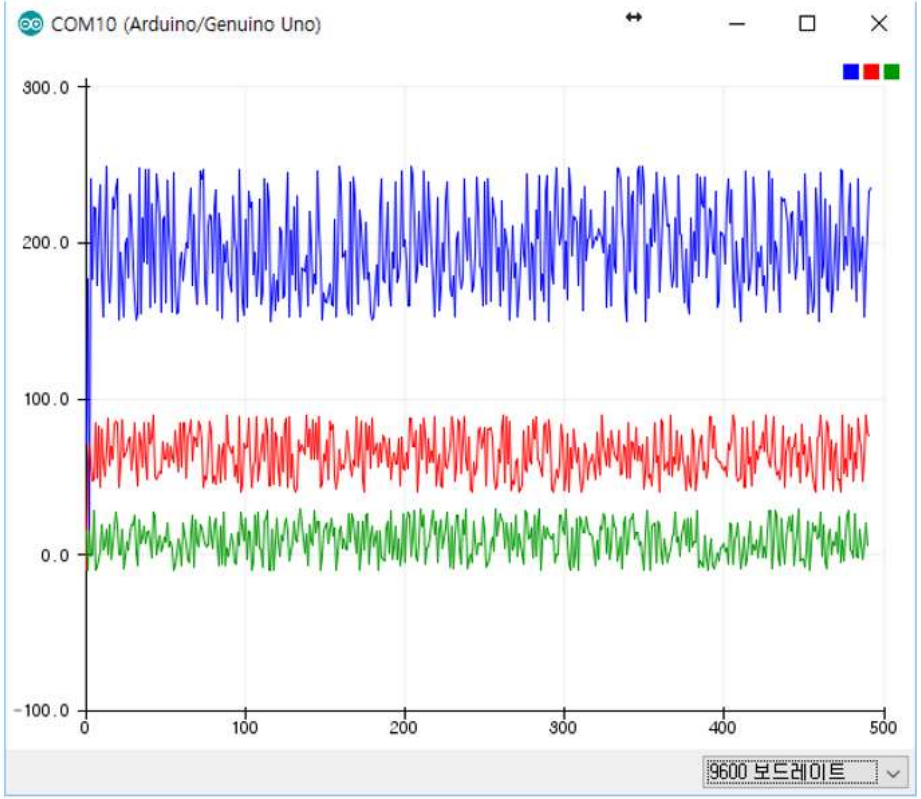
가상적인 세 개의 센서신호 시뮬레이션:조도(위), 습도(중간), 온도(아래).

COM10 (Arduino/Genuino Uno)

전송

```
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
```

자동 스크롤 | line ending 없음 | 9600 보드레이트 | 출력 지우기



DIY – New result 1

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

COM10 (Arduino/Genuino Uno)

전송

```
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

자동 스크롤 | line ending 없음 | 9600 보드레이트 | 출력 지우기

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

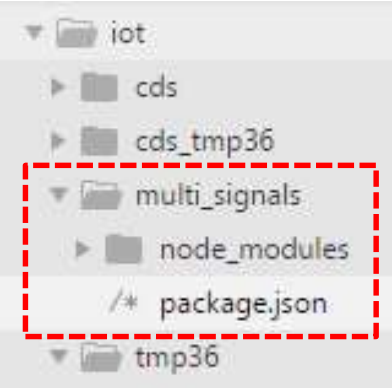
entry point : aann_multi_signals.js

author : aann

3. Install node modules

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

```
npm
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.js**로 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

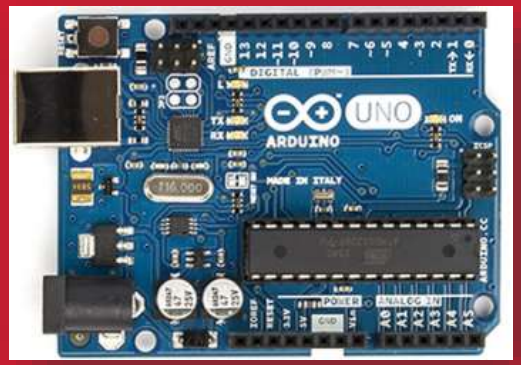
DIY 결과 [2] : 가상적인 세 개의 센서신호 시뮬레이션 → 조도,습도,온도를 **Node.js**로 처리

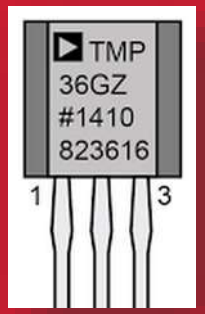
npm - node aann_multi_signals

```
^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

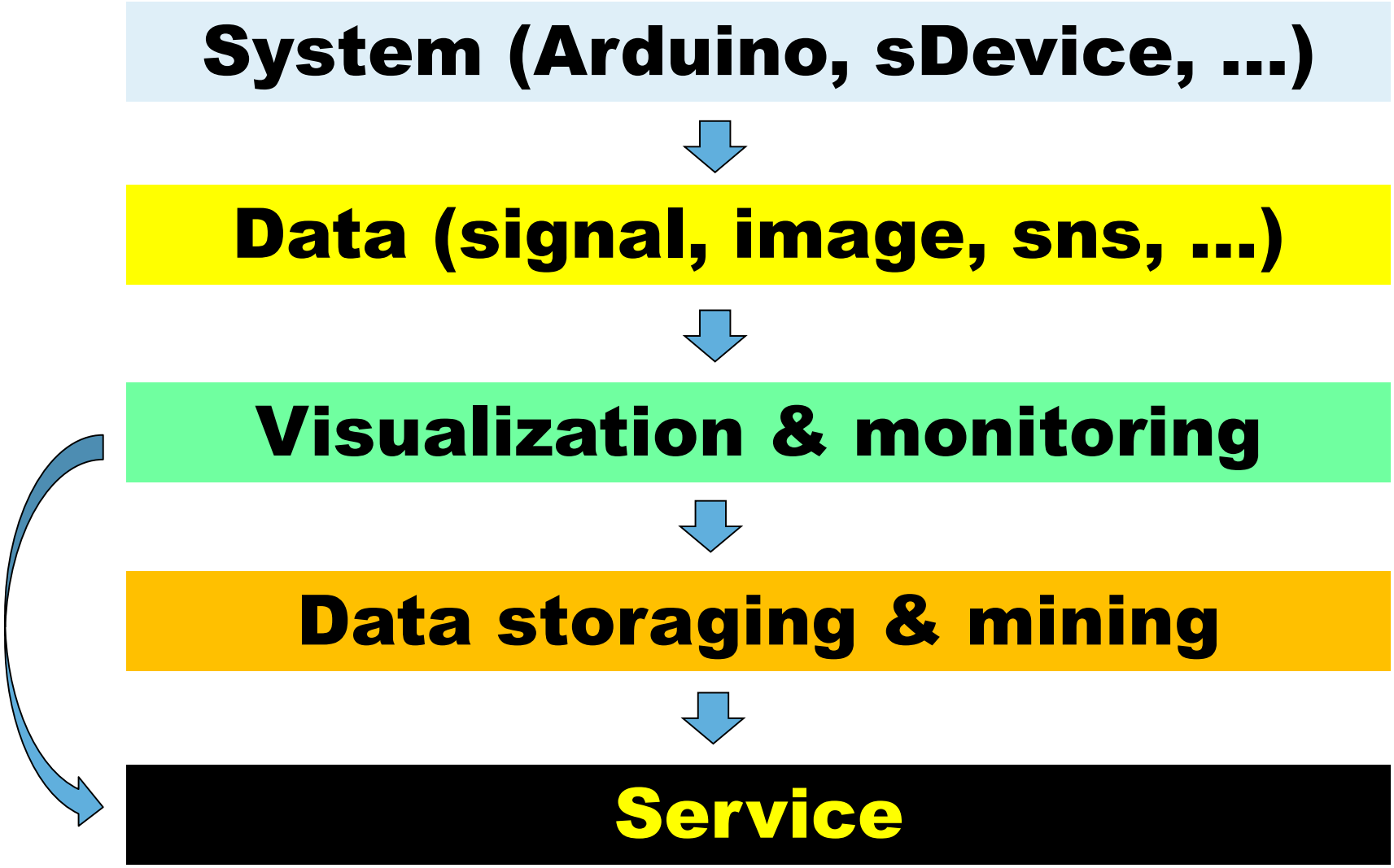
Save this result as AAnn_multi_signals_node .png





Data visualization using ploy.ly

A5. Introduction to visualization



A5.1 Introduction to data visualization

아두이노 센서 회로

↓

직렬모니터/플로터 모니터링

↓

LCD 모니터링

Node.js → ↓ ← Plotly.js

웹 모니터링

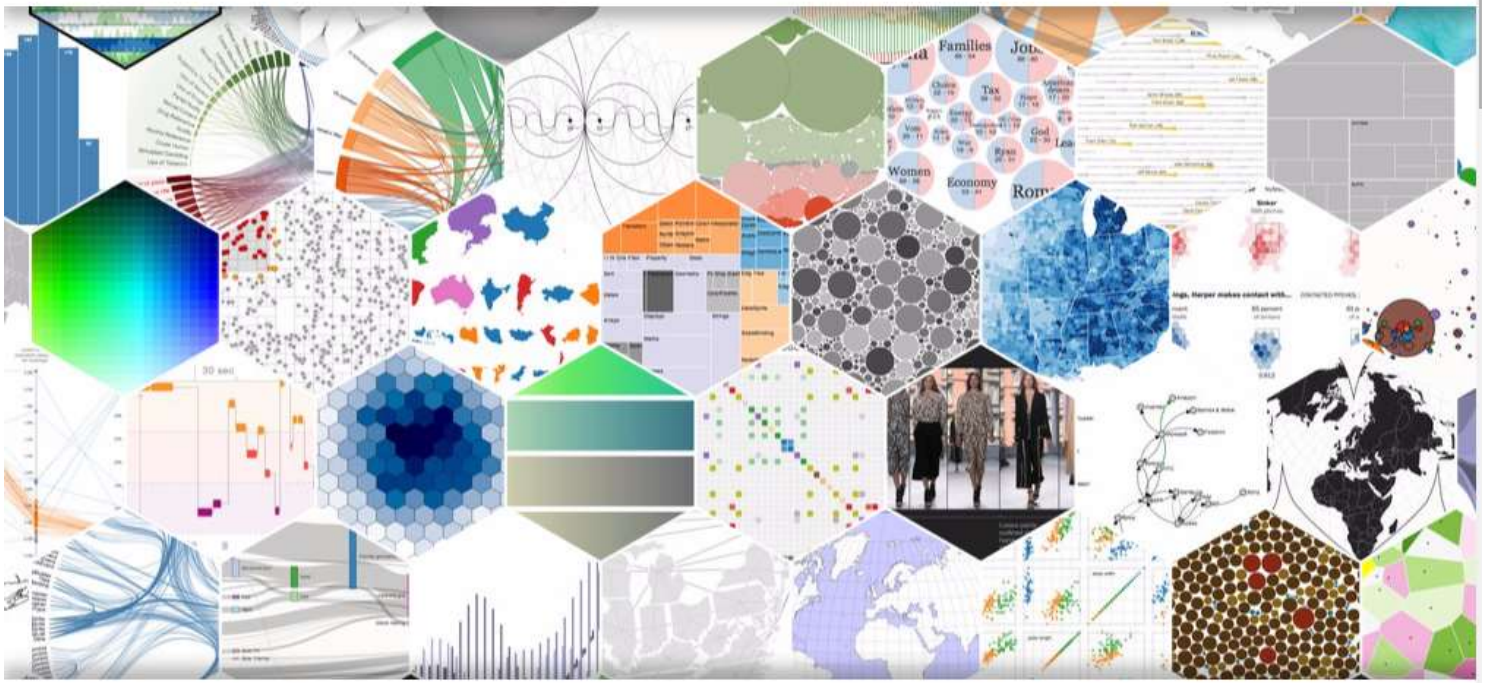
A5.1.1 D3.js

d3js.org

Overview Examples Documentation Source

Fork me on GitHub

Data-Driven Documents

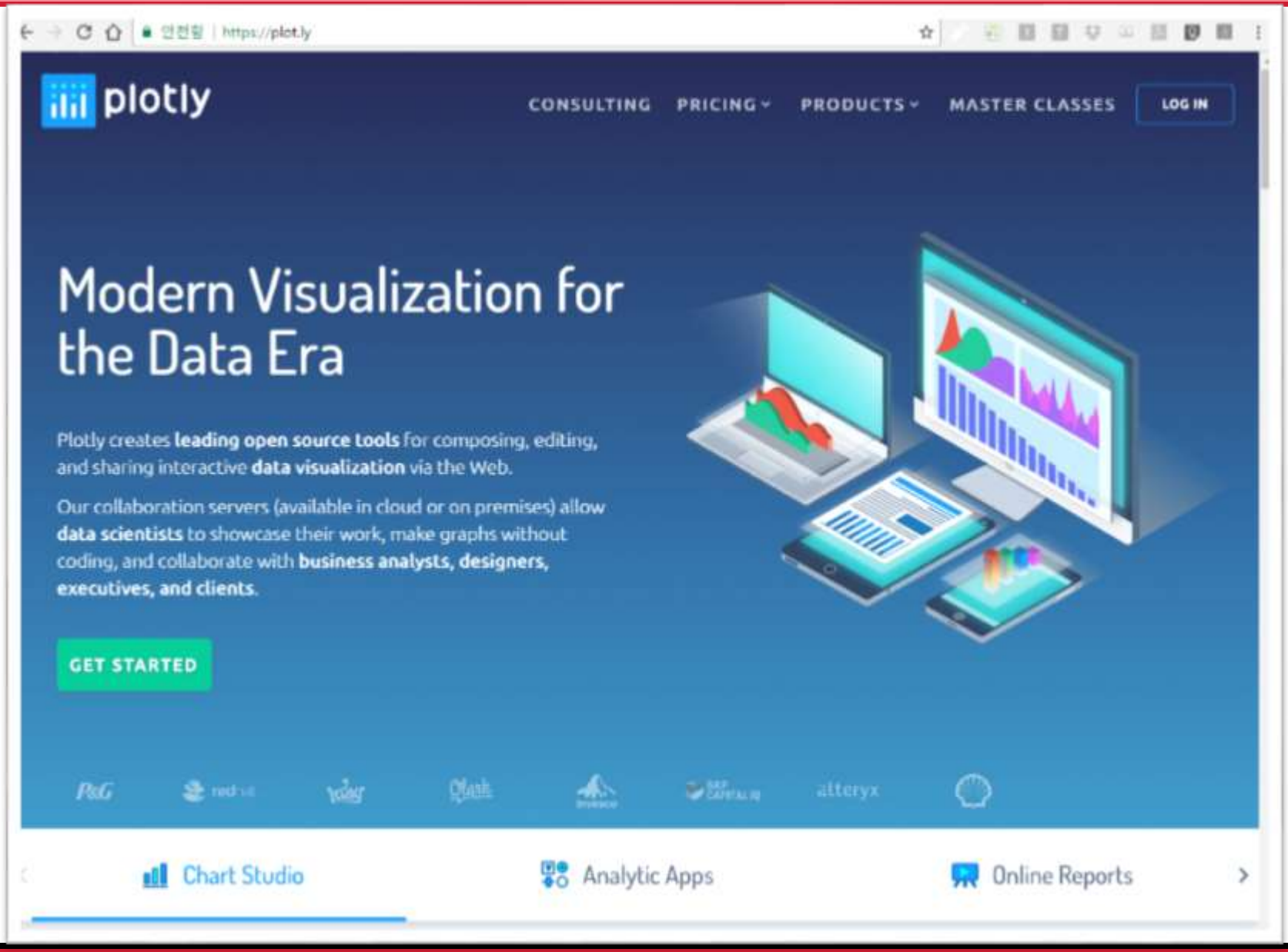


D3.js is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

See more examples.

http://d3js.org/

A5.1.2 plot.ly



https://plot.ly/

A5.1.3 plotly.js

plotly

plotly.js is Plotly's client-side, interactive JavaScript charting library, built on top of D3.js, stack.gl, and jQuery.

https://plot.ly/javascript/

A5.1.4 Introduction to plotly.js

https://plotly/javascript/

plotly Developer Support PLOTCON Consulting SIGN IN SIGN UP REQUEST DEMO

plotly.js

The open source JavaScript graphing library that powers Plotly

Help API Libraries Plotly.js Fork on Github

Quick Start

Getting Started

Cheat Sheet

CDN

Download

Full Reference

Event Reference

Function Reference

Configuration Options

Examples

Plotly Fundamentals

Basic

Statistical

Scientific

JS

What is plotly.js?

Built on top of d3.js and stack.gl, plotly.js is a high-level, declarative charting library. plotly.js ships with 20 chart types, including 3D charts, statistical graphs, and SVG maps.

Learn about why we open sourced plotly.js or view the source on GitHub.

▸ Plotly.js Features

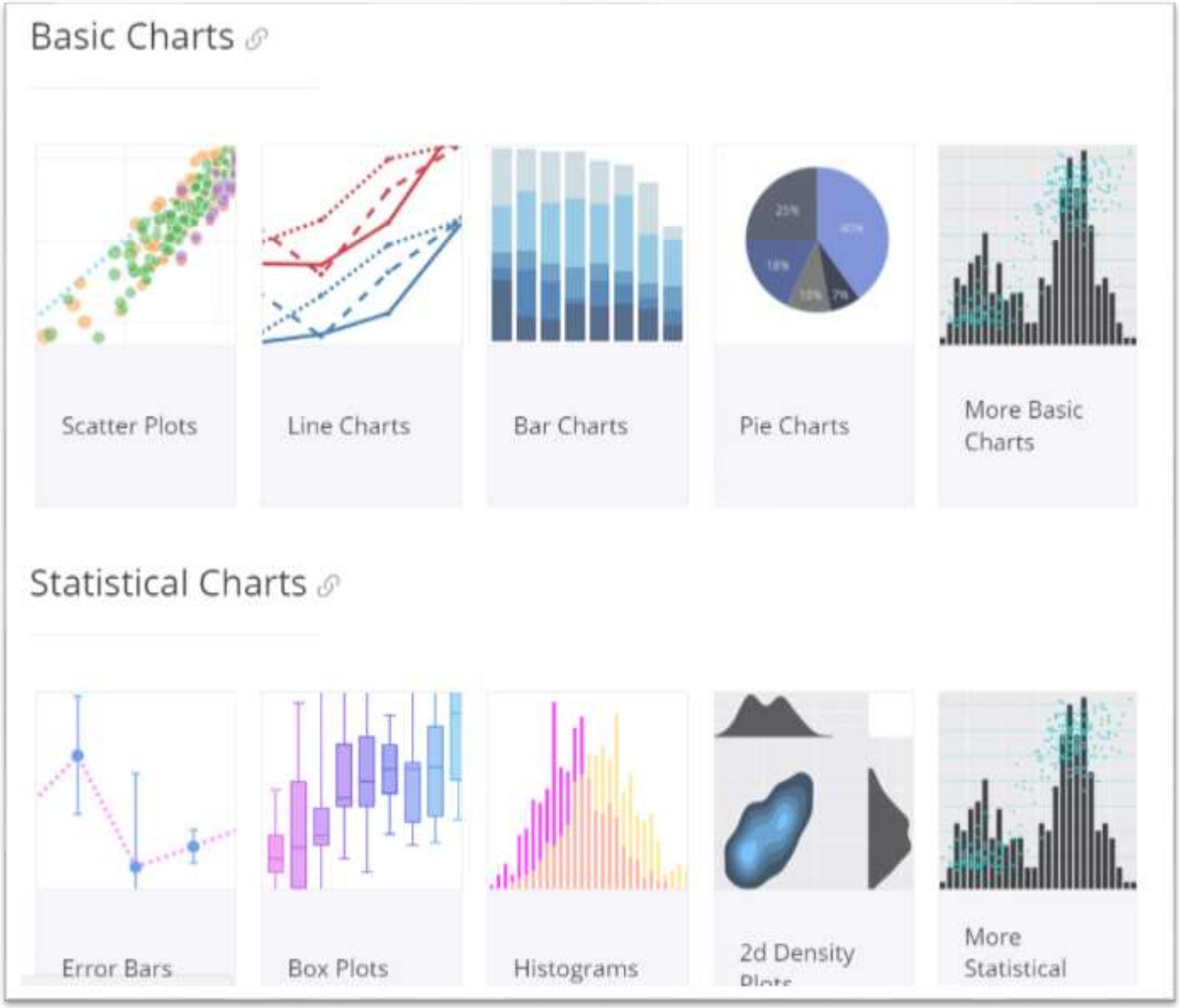
Search

Search Plotly.js Docs

Plotly Fundamentals

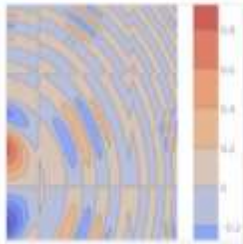
https://plot.ly/javascript/getting-started/#download

A5.1.5 Introduction to plotly.js charts

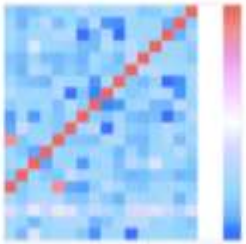


A5.1.6 Introduction to plotly.js charts

Scientific Charts

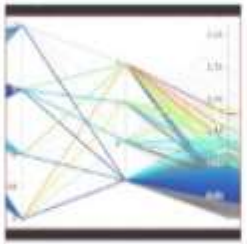


Contour Plots

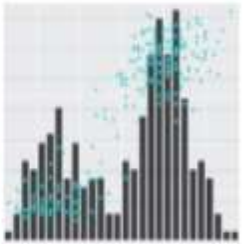


Heatmaps

Ternary Plots

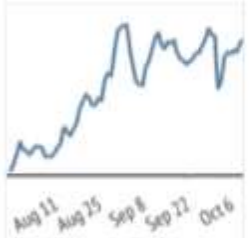


Parallel Coordinates Plot

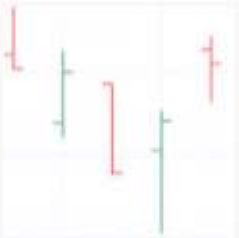


More Scientific Charts

Financial Charts



Time Series



OHLC Charts

Candlestick Charts

A5.1.7 Introduction to plotly.js charts

Maps

Choropleth Maps

Scatter Plots on Maps

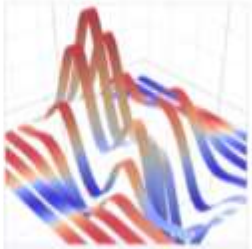
Bubble Maps

Lines on Maps

Scatter Plots on Mapbox

3D Charts

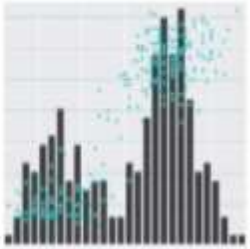
3D Scatter Plots



Ribbon Plots

3D Surface Plots

3D Mesh Plots



More 3D Charts

A5.1.8 plotly.js: time series & streaming

https://plot.ly/javascript/time-series/

https://plot.ly/javascript/streaming/

A5.1.9 Getting started: plotly.js

https://plot.ly/javascript/getting-started/

A5.1.10 Getting started: plotly.js

plotly.js CDN

You can also use the ultrafast plotly.js CDN link. This CDN is graciously provided by the incredible team at Fastly.

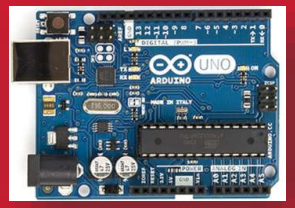
```
<head>
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
</head>
```

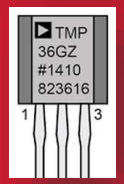
Else, if you want to get a specific version of plotly.js, say 1.2.0:

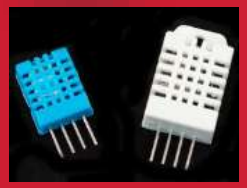
```
<head>
    <script src="https://cdn.plot.ly/plotly-1.2.0.min.js"></script>
</head>
```

```
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
```

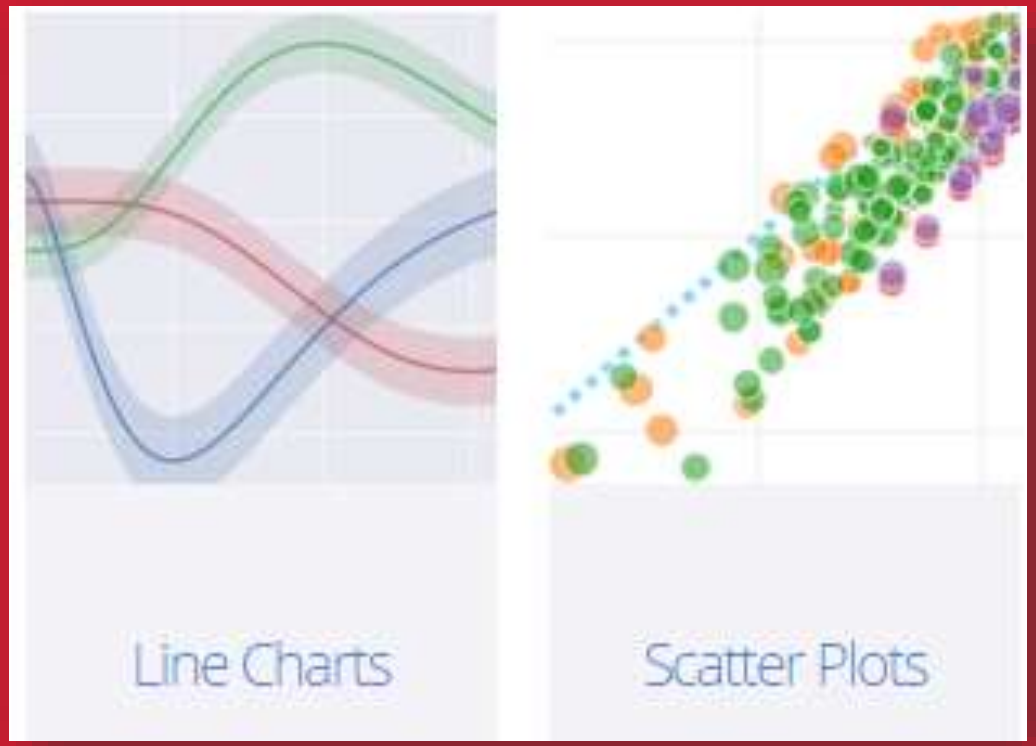
https://plot.ly/javascript/getting-started/







Data charts using plotly.js



A5.2 Data charts

Navigation

Basic Line Plot

Line and Scatter Plot

Adding Names to Line and Scatter Plot

Line and Scatter Styling

Styling Line Plot

Colored and Styled Scatter Plot

Line Shape Options for Interpolation

Graph and Axes Titles

Line Dash

Connect Gaps Between Data

Labelling Lines with Annotations

Back To Plotly.Js

Line Charts in plotly.js

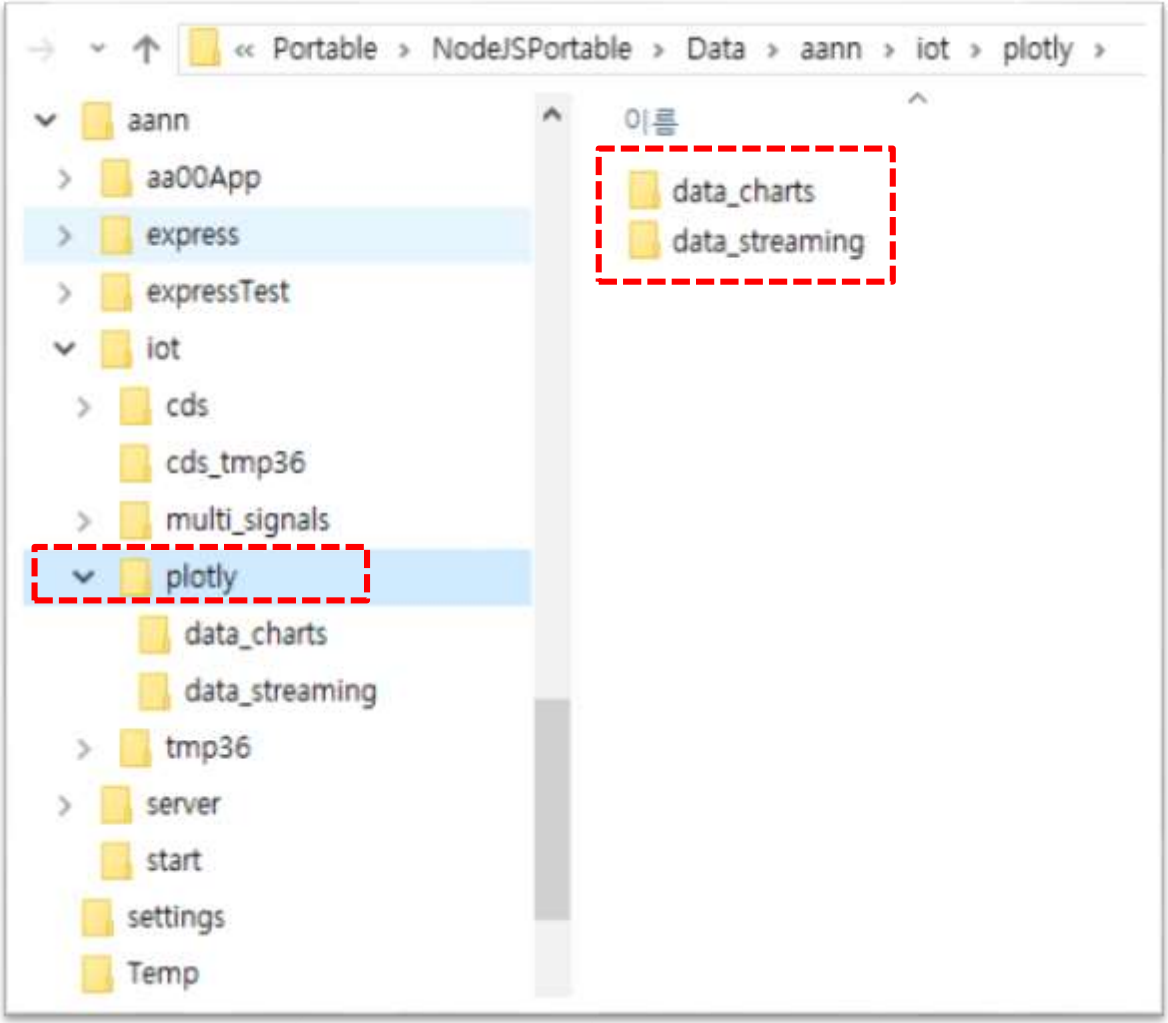
How to make D3.js-based line charts in JavaScript.

R | plotly.js | Python | Pandas

Basic Line Plot

```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  type: 'scatter'
};

var trace2 = {
  x: [1, 2, 3, 4],
  y: [16, 5, 11, 9],
  type: 'scatter'
};
```

A5.2.1 Working folders



A5.2.2.1 Starting plotly basic chart

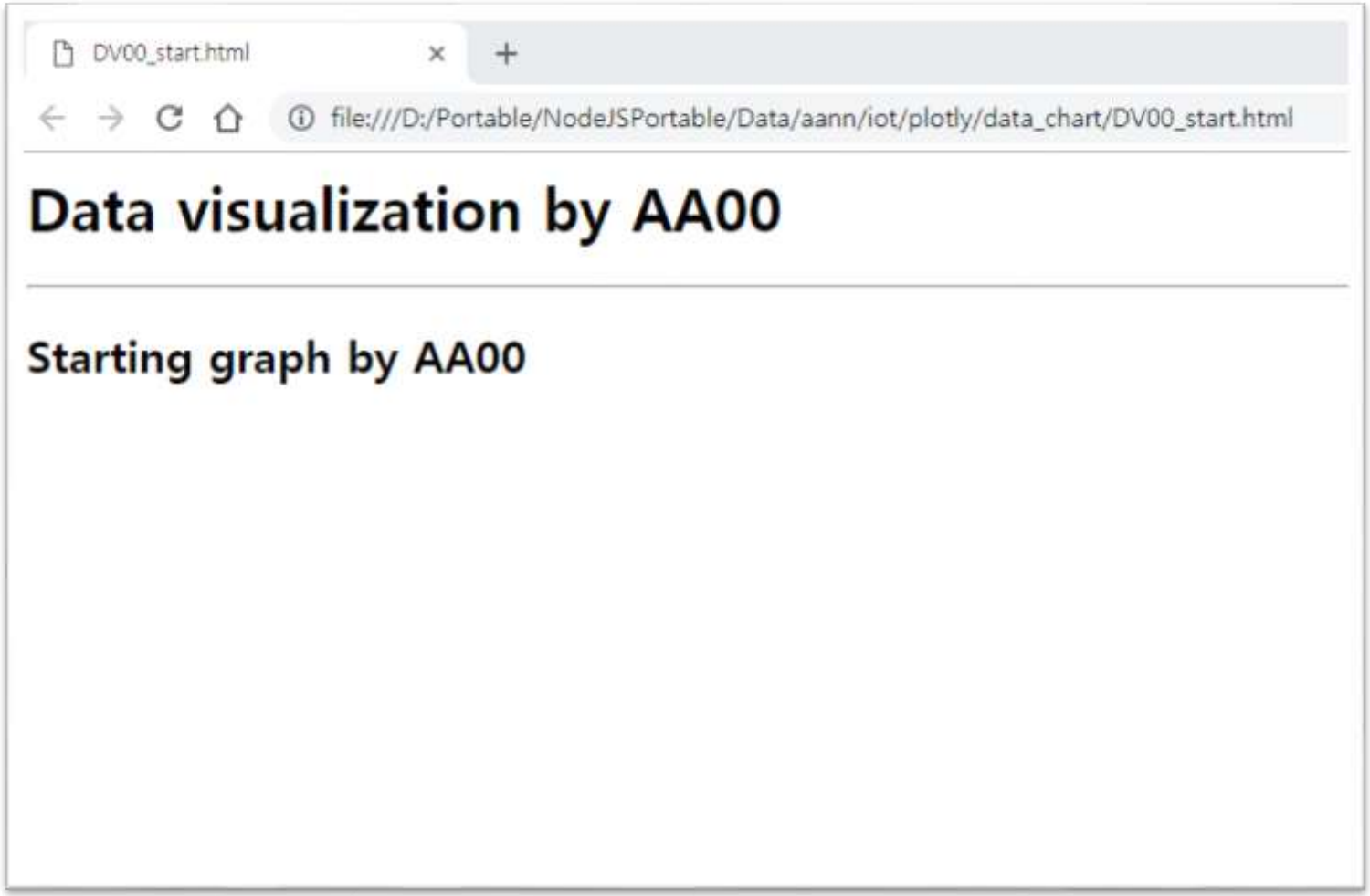
DV00_start.html

Starting chart!

```html
<html>
<head>
    <meta charset="utf-8">
    <!-- Plotly.js -->
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
</head>
<body>
    <h1>Data visualization by AA00</h1>
    <hr>
    <h2>Starting graph by AA00</h2>

    <!-- Plotly chart will be drawn inside this DIV -->
    <div id="myDiv" style="width: 500px;height: 300px"></div>

    <script>
        <!-- JAVASCRIPT CODE GOES HERE -->


    </script>
</body>
</html>
```

A5.2.2.2 Starting plotly basic chart

SB3, ^B



[Tip] Using WEB browser in SB text3

[Tool] Sublime Text – 현재 작업 중인 파일을 웹브라우저로 열기

1. Tool > Developer > New Plugin을 실행 한 후 아래 내용으로 덮어 씌운 후 **'open_browser'**으로 저장한다**.**

```
import sublime, sublime_plugin
import webbrowser

class OpenBrowserCommand(sublime_plugin.TextCommand):
    def run(self,edit):
        url = self.view.file_name()
        webbrowser.open_new(url)
```

2. Preferences -> Key Bindings - User로 이동한 후 단축키를 할당한다**.**

```
{ "keys": ["f10"], "command": "open_browser" }
```

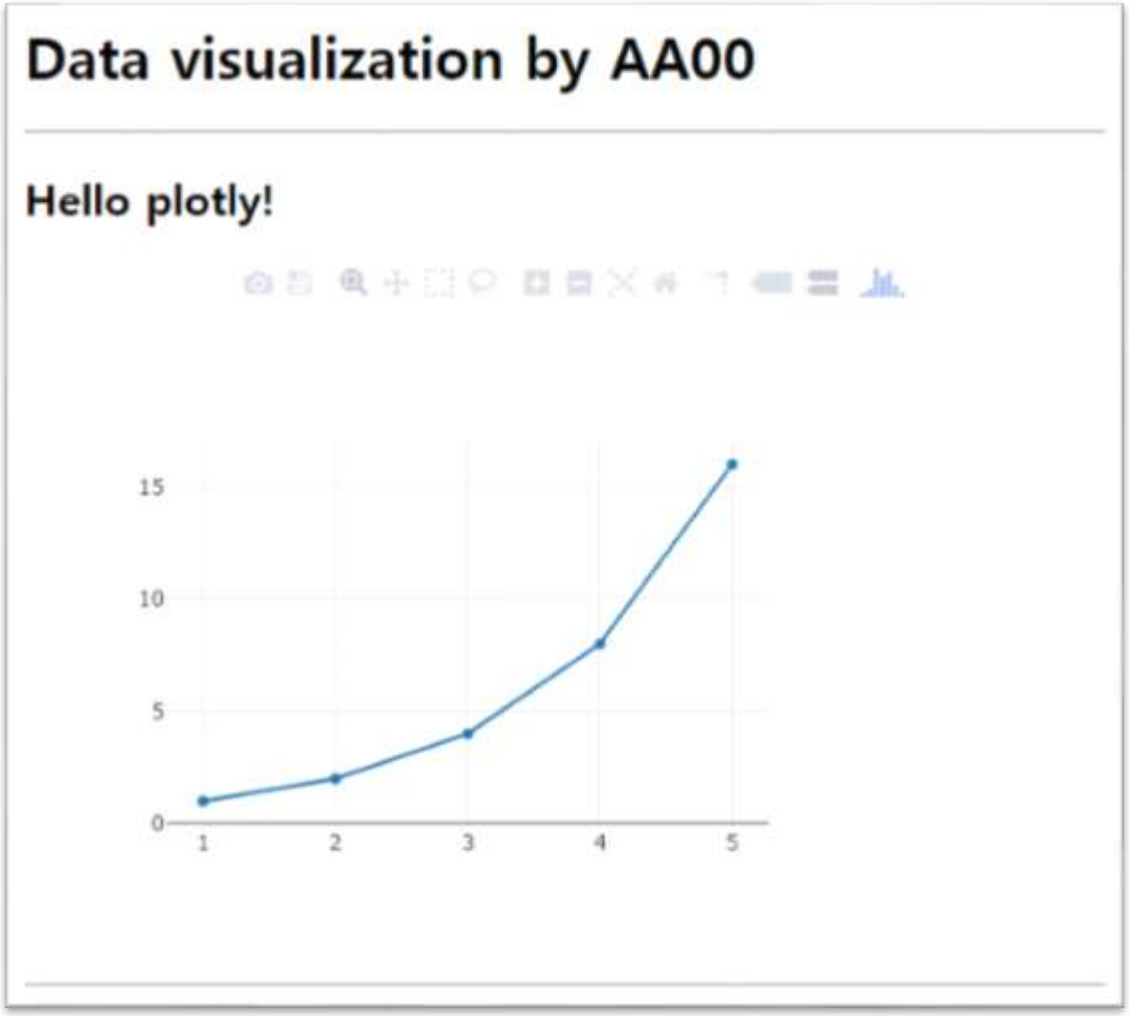
A5.2.3.1 Hello plotly basic chart

Hello plotly data chart!

```html
<html>
<head>
    <meta charset="utf-8">
    <!-- Plotly.js -->
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
</head>
<body>
    <h1>Data visualization by AA00</h1>
    <hr>
    <h2>Hello plotly!</h2>
    <!-- Plotly chart will be drawn inside this DIV -->
    <div id="myDiv" style="width: 500px;height: 400px"></div>
    <hr>
    <script>
        <!-- JAVASCRIPT CODE GOES HERE -->
        var data = [
        {
            x: [1, 2, 3, 4, 5],
            y: [1, 2, 4, 8, 16],
            type: 'scatter'
        }];

        Plotly.newPlot('myDiv', data);

    </script>
</body>
</html>
```

A5.2.3.2 Hello plotly basic chart

Graph : Hello plotly chart!

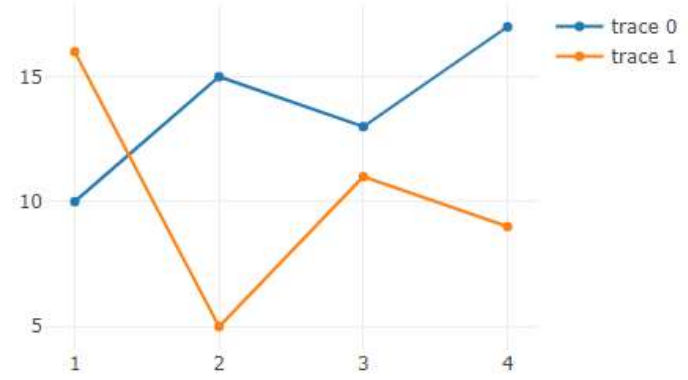


A5.2.4 plotly.js: Line Charts

[1] Basic multi-line charts

```
<script>
    <!-- JAVASCRIPT CODE GOES HERE -->

    var trace1 = {
        x: [1, 2, 3, 4],
        y: [10, 15, 13, 17],
        type: 'scatter'
    };

    var trace2 = {
        x: [1, 2, 3, 4],
        y: [16, 5, 11, 9],
        type: 'scatter'
    };

    var data = [trace1, trace2];

    Plotly.newPlot('myDiv', data);

</script>
```

Line charts by AA00

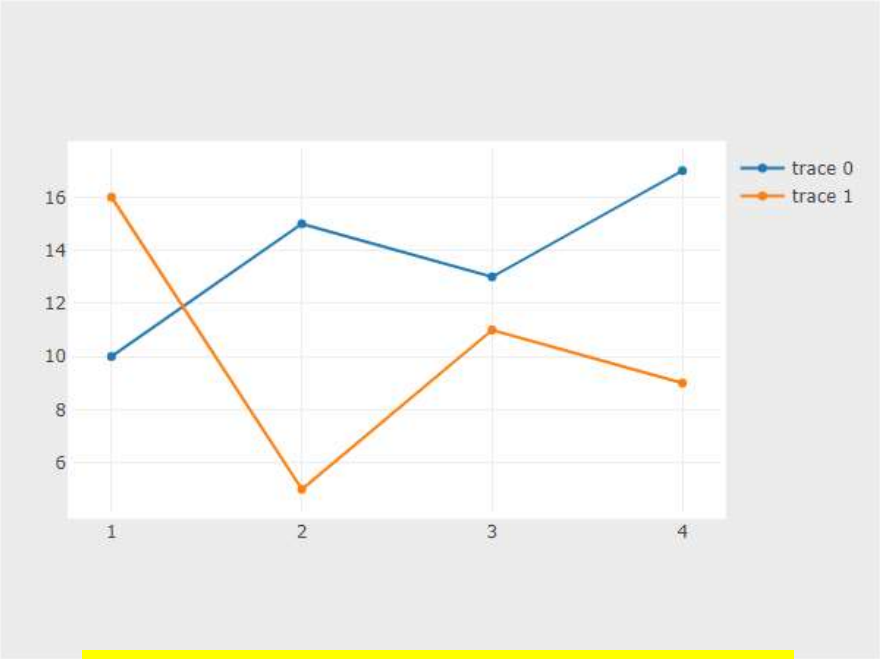


A5.2.5 plotly.js: Line Charts

[2] Basic line charts with layout

```
var layout = {
    autosize: false,
    width: 600,
    height: 450,
    margin: {
        l: 50,  // left
        r: 50,  // right
        b: 100, // bottom
        t: 100, // top
        pad: 4  // padding
    },
    paper_bgcolor: '#ececec',
    plot_bgcolor: '#ffffff' //'#rrggbb'
};

Plotly.newPlot('myDiv', data, layout);
```

Line charts with layout by AA00



AAnn_Chart_Layout.png

Test: pad → 40

A5.2.6.1 plotly.js: Line & Scatter plot

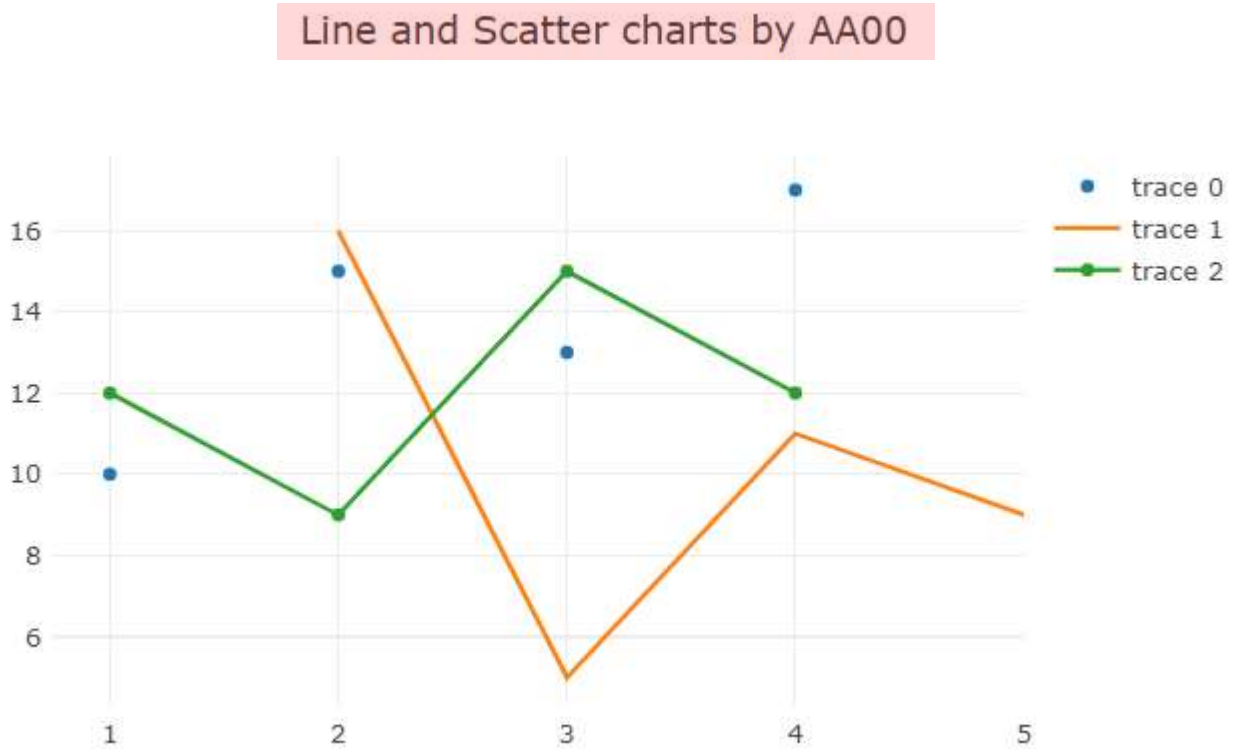
[3] Line & scatter plot

```
var trace1 = {
    x: [1, 2, 3, 4],
    y: [10, 15, 13, 17],
    mode: 'markers'
};

var trace2 = {
    x: [2, 3, 4, 5],
    y: [16, 5, 11, 9],
    mode: 'lines'
};

var trace3 = {
    x: [1, 2, 3, 4],
    y: [12, 9, 15, 12],
    mode: 'lines+markers'
};
```

```
var data = [ trace1, trace2, trace3 ];

var layout = {
    title:'Line and Scatter charts by AA00',
    width: 600,
    height: 450,
    margin: {
        l: 50,
        r: 50,
        b: 100,
        t: 100,
        pad: 4
    },
};

Plotly.newPlot('myDiv', data, layout);
```

A5.2.6.2 plotly.js: Line & Scatter plot

[3.1] Line & scatter plot with title



A5.2.6.3 plotly.js: Line & Scatter plot

[3.2] Line & scatter plot with axis name

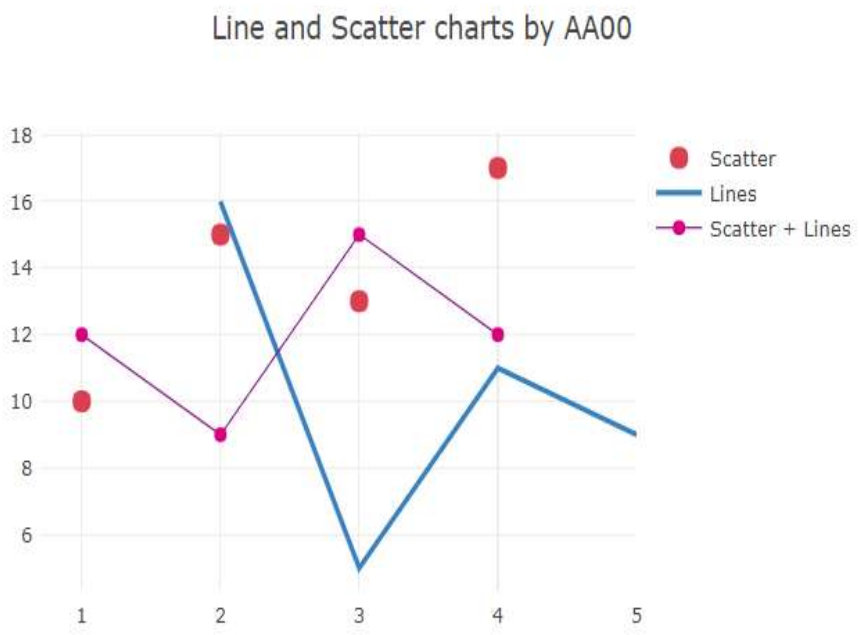
```
var trace1 = {
    x: [1, 2, 3, 4],
    y: [10, 15, 13, 17],
    mode: 'markers',
    name: 'Scatter'
};

var trace2 = {
    x: [2, 3, 4, 5],
    y: [16, 5, 11, 9],
    mode: 'lines',
    name: 'Lines'
};

var trace3 = {
    x: [1, 2, 3, 4],
    y: [12, 9, 15, 12],
    mode: 'lines+markers',
    name: 'Scatter + Lines'
};
```

A5.2.6.4 plotly.js: Line & Scatter plot

[3.3] Line & scatter plot with style

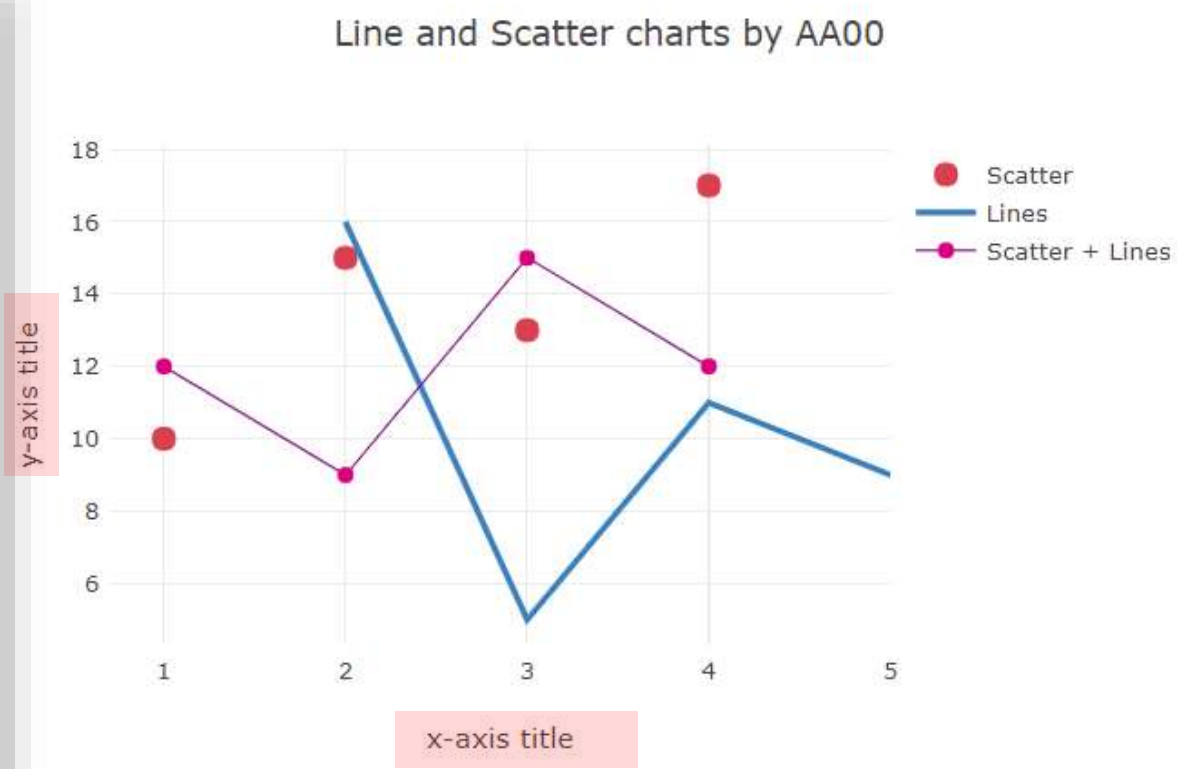
```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  mode: 'markers',
  name: 'Scatter',
  marker: {
    color: 'rgb(219, 64, 82)',
    size: 12
  }
};

var trace2 = {
  x: [2, 3, 4, 5],
  y: [16, 5, 11, 9],
  mode: 'lines',
  name: 'Lines',
  line: {
    color: 'rgb(55, 128, 191)',
    width: 3
  }
};
```

```
var trace3 = {
  x: [1, 2, 3, 4],
  y: [12, 9, 15, 12],
  mode: 'lines+markers',
  name: 'Scatter + Lines',
  marker: {
    color: 'rgb(128, 0, 128)',
    size: 8
  },
  line: {
    color: 'rgb(128, 0, 128)',
    width: 1
  }
};
```



[3.4] Line & scatter plot with axis titles

```
var layout = {
  title:'Line and Scatter Plot',
  width: 600, height: 450,
  margin: {
    l: 50,
    r: 50,
    b: 100,
    t: 100,
    pad: 4
  },
  xaxis: {
    title: 'x-axis title'
  },
  yaxis: {
    title: 'y-axis title'
  }
};
```



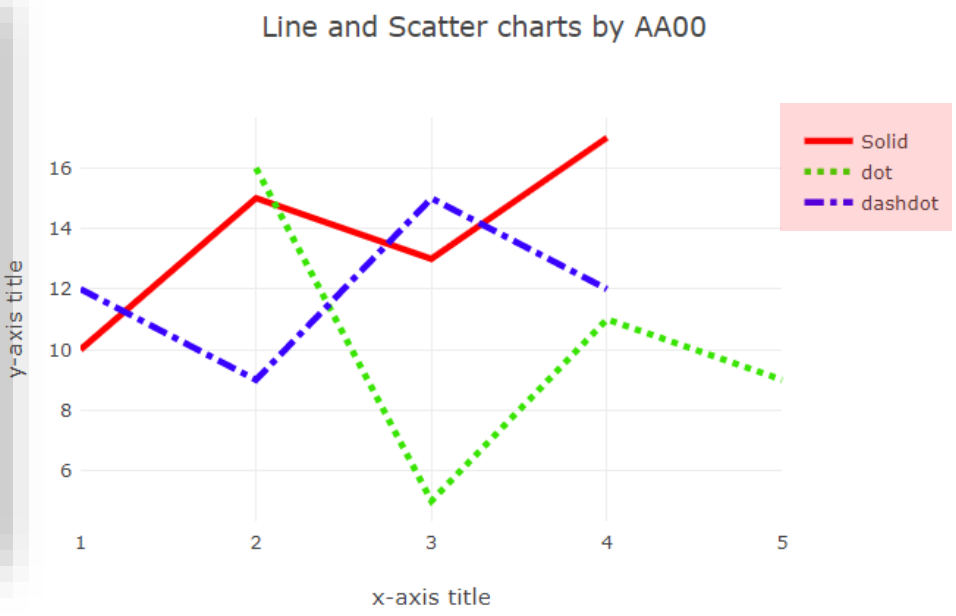
AAnn_Axis_Title.png

A5.2.6.6 plotly.js: Line & Scatter plot

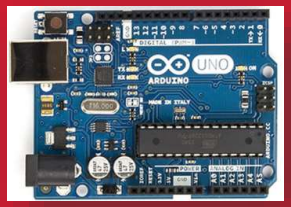
[3.5] Line & scatter plot with dash and dot

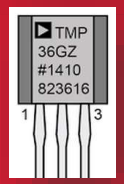
```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  mode: 'lines',
  name: 'Solid',
  line: {
    color: 'rgb(255, 0, 0)',
    dash: 'solid',
    width: 4
  }
};

var trace2 = {
  x: [2, 3, 4, 5],
  y: [16, 5, 11, 9],
  mode: 'lines',
  name: 'dot',
  line: {
    color: 'rgb(55, 228, 0)',
    dash: 'dot',
    width: 4
  }
};
```

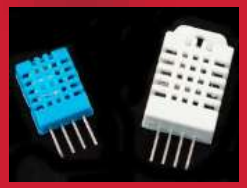
```
var trace3 = {
  x: [1, 2, 3, 4],
  y: [12, 9, 15, 12],
  mode: 'lines',
  name: 'dashdot',
  line: {
    color: 'rgb(55, 0, 255',
    dash: 'dashdot',
    width: 4
  }
};
```



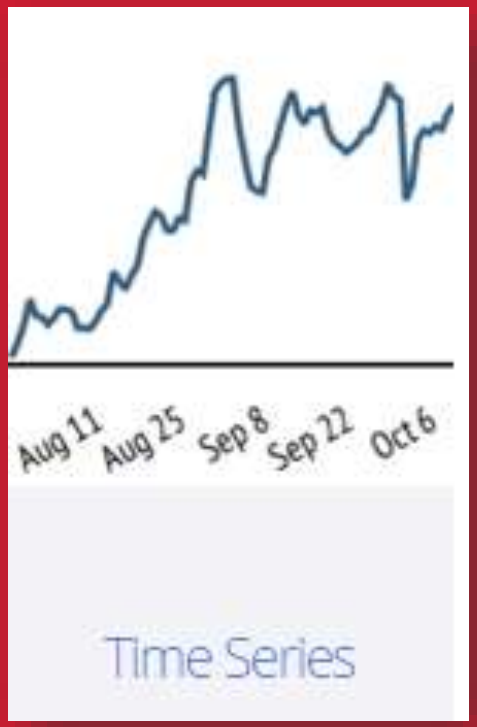
AAnn_Line_Dash_Dot.png







Data visualization using plotly.js



A5.3. Time series

Navigation

Date Strings

Basic Time Series

Manually Set Range

Time Series with Rangeslider

Back To Plotly.js

Time Series in plotly.js

How to plot D3.js-based date and time in Plotly.js. An example of a time-series plot.

R | Python | matplotlib | plotly.js | Pandas | node.js | MATLAB

Date Strings

```
var data = [
  {
    x: ['2013-10-04 22:23:00', '2013-11-04 22:23:00', '2013-12-04 22:23:00'],
    y: [1, 3, 6],
    type: 'scatter'
  }
];

Plotly.newPlot('myDiv', data);
```

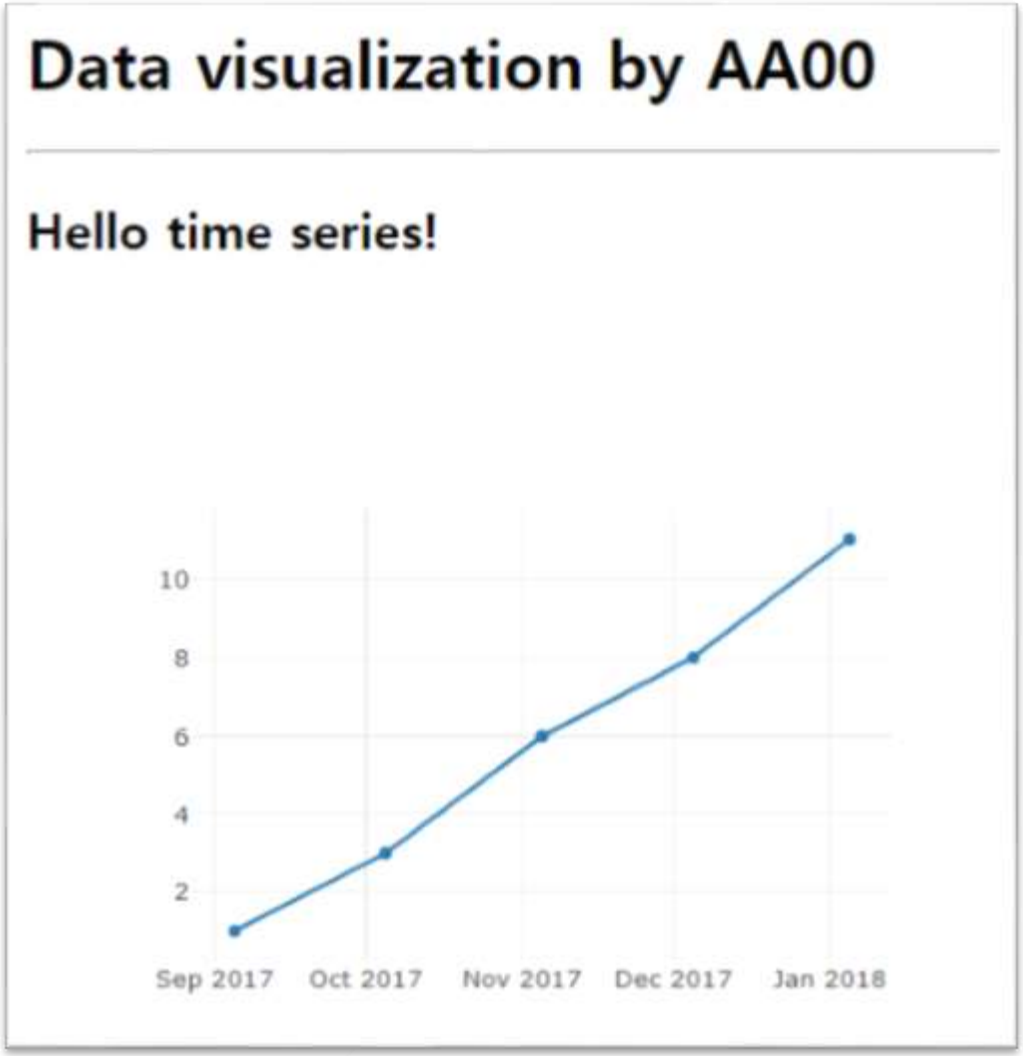
A5.3.1 plotly.js: Time series

[1] Time series : date strings

```
<!-- Plotly chart will be drawn inside this DIV -->
<div id="myDiv" style="width: 500px;height: 400px"></div>

<script>
    <!-- JAVASCRIPT CODE GOES HERE -->

    var data = [
    {
        x: ['2017-9-04 22:23:00',
        '2017-10-04 22:23:00',
        '2017-11-04 22:23:00',
        '2017-12-04 22:23:00'],
        y: [1, 3, 6, 8],
        type: 'scatter'
    }
    ];

    Plotly.newPlot('myDiv', data);

</script>
```

A5.3.2 plotly.js: Time series

Time series : date strings – result



A5.3.3.1 plotly.js: Time series

[2] Time series : financial data strings – AAPL stock price

안전함 | https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-a...

```
Date,AAPL.Open,AAPL.High,AAPL.Low,AAPL.Close,AAPL.Volume,AAPL.Adjusted,dn,mavg,up,direction
2015-02-17,127.489998,128.880005,126.919998,127.830002,63152400,122.905254,106.7410523,117.9276669,129.1142814,Increasing
2015-02-18,127.629997,128.779999,127.449997,128.720001,44891700,123.760965,107.842423,118.9403335,130.0382439,Increasing
2015-02-19,128.479996,129.029999,128.330002,128.449997,37362400,123.501363,108.8942449,119.8891668,130.8840887,Decreasing
2015-02-20,128.619995,129.5,128.050003,129.5,48948400,124.510914,109.7854494,120.7635001,131.7415509,Increasing
2015-02-23,130.020004,133,129.660004,133,70974100,127.876074,110.3725162,121.7201668,133.0678174,Increasing
2015-02-24,132.940002,133.600006,131.169998,132.169998,69228100,127.078049,111.0948689,122.6648335,134.2347981,Decreasing
2015-02-25,131.559998,131.600006,128.149994,128.789993,74711700,123.828261,113.2119183,123.6296667,134.0474151,Decreasing
2015-02-26,128.789993,130.869995,126.610001,130.419998,91287500,125.395469,114.1652991,124.2823333,134.3993674,Increasing
2015-02-27,130,130.570007,128.240005,128.460007,62014800,123.510987,114.9668484,124.8426669,134.7184854,Decreasing
2015-03-02,129.25,130.279999,128.300003,129.089996,48096700,124.116706,115.8770904,125.4036668,134.9302432,Decreasing
2015-03-03,128.960007,129.520004,128.089996,129.360001,37816300,124.376308,116.9535132,125.9551669,134.9568205,Increasing
2015-03-04,129.100006,129.559998,128.320007,128.539993,31666300,123.587892,118.0874253,126.4730002,134.8585751,Decreasing
2015-03-05,128.580002,128.75,125.760002,126.410004,56517100,121.539962,119.1048311,126.848667,134.5925029,Decreasing
2015-03-06,128.399994,129.369995,126.260002,126.599998,72842100,121.722637,120.190797,127.2288335,134.26687,Decreasing
2015-03-09,127.959999,129.570007,125.059998,127.139999,88528500,122.241834,121.6289771,127.631167,133.6333568,Decreasing
2015-03-10,126.410004,127.220001,123.800003,124.510002,68856600,119.71316,123.1164763,127.9235004,132.7305246,Decreasing
2015-03-11,124.75,124.769997,122.110001,122.239998,68939000,117.530609,123.592756,128.0033337,132.4139113,Decreasing
2015-03-12,122.309998,124.900002,121.629997,124.449997,48362700,119.655466,123.4894559,127.9813337,132.4732114,Increasing
2015-03-13,124.400002,125.400002,122.580002,123.589996,51827300,118.828598,123.045606,127.8490003,132.6523946,Decreasing
2015-03-16,123.879997,124.949997,122.870003,124.949997,35874300,120.136203,122.6967016,127.7283335,132.7599655,Increasing
2015-03-17,125.900002,127.32,125.650002,127.040001,51023100,122.145688,122.616033,127.6680002,132.7199674,Increasing
2015-03-18,127,129.160004,126.370003,128.470001,65270800,123.520597,122.6064498,127.652167,132.6978842,Increasing
2015-03-19,128.75,129.25,127.400002,127.5,45809500,122.587966,122.5939029,127.6245004,132.6550979,Decreasing
2015-03-20,128.25,128.399994,125.160004,125.900002,68695100,121.049608,122.4865925,127.4980004,132.5094083,Decreasing
2015-03-23,127.120003,127.849998,126.519997,127.209999,37709700,122.309137,122.6741703,127.2633335,131.8524968,Increasing
2015-03-24,127.230003,128.039993,126.559998,126.690002,32842300,121.809174,123.0410183,127.0025001,130.9639818,Decreasing
2015-03-25,126.540001,126.82,123.379997,123.379997,51655200,118.626689,122.8276392,126.7531667,130.6786943,Decreasing
2015-03-26,122.760002,124.879997,122.599998,124.239998,47572900,119.453558,122.5538523,126.4835001,130.4131478,Increasing
2015-03-27,124.57,124.699997,122.910004,123.25,39546200,118.5017,122.2826504,126.2099998,130.1373491,Decreasing
2015-03-30,124.050003,126.400002,124,126.370003,47099700,121.501502,122.346906,126.0283332,129.7097604,Increasing
2015-03-31,126.089996,126.489998,124.360001,124.43,42090600,119.63624,122.395242,125.8334998,129.2717577,Decreasing
2015-04-01,124.82,125.120003,123.099998,124.25,40621400,119.463174,122.3761274,125.6009999,128.8258723,Decreasing
```

A5.3.3.2 plotly.js: Time series

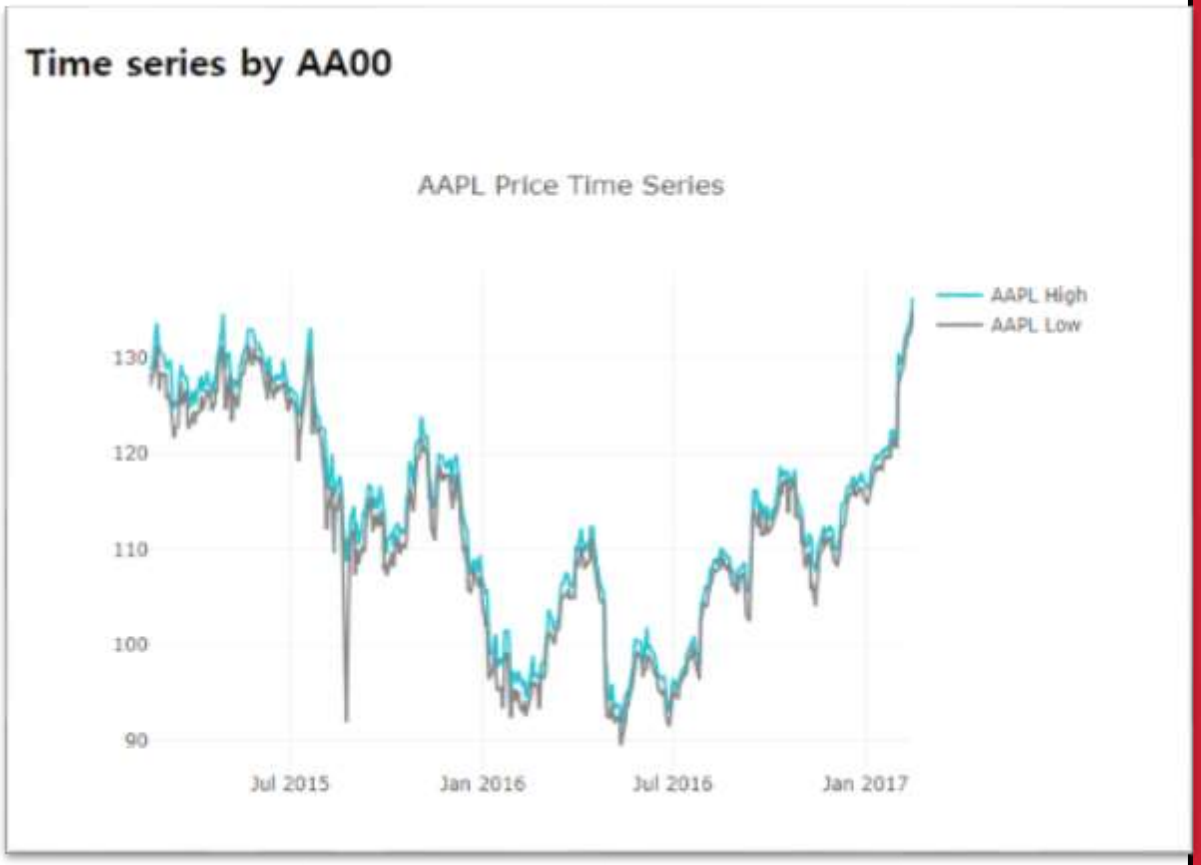
[2] Time series : financial data strings – AAPL stock price

```
Plotly.d3.csv("https://raw.githubusercontent.com/plotly/datasets/master/
    finance-charts-apple.csv", function(err, rows){

    function unpack(rows, key) {
        return rows.map(function(row) { return row[key]; });
    }


    var trace1 = {
        type: "scatter",
        mode: "lines",
        name: 'AAPL High',
        x: unpack(rows, 'Date'),
        y: unpack(rows, 'AAPL.High'),
        line: {color: '#17BECF'}
    }

    var trace2 = {
        type: "scatter",
        mode: "lines",
        name: 'AAPL Low',
        x: unpack(rows, 'Date'),
        y: unpack(rows, 'AAPL.Low'),
        line: {color: '#7F7F7F'}
    }

    var data = [trace1,trace2];
```

A5.3.3.3 plotly.js: Time series

[2] Time series : financial data strings – AAPL stock price

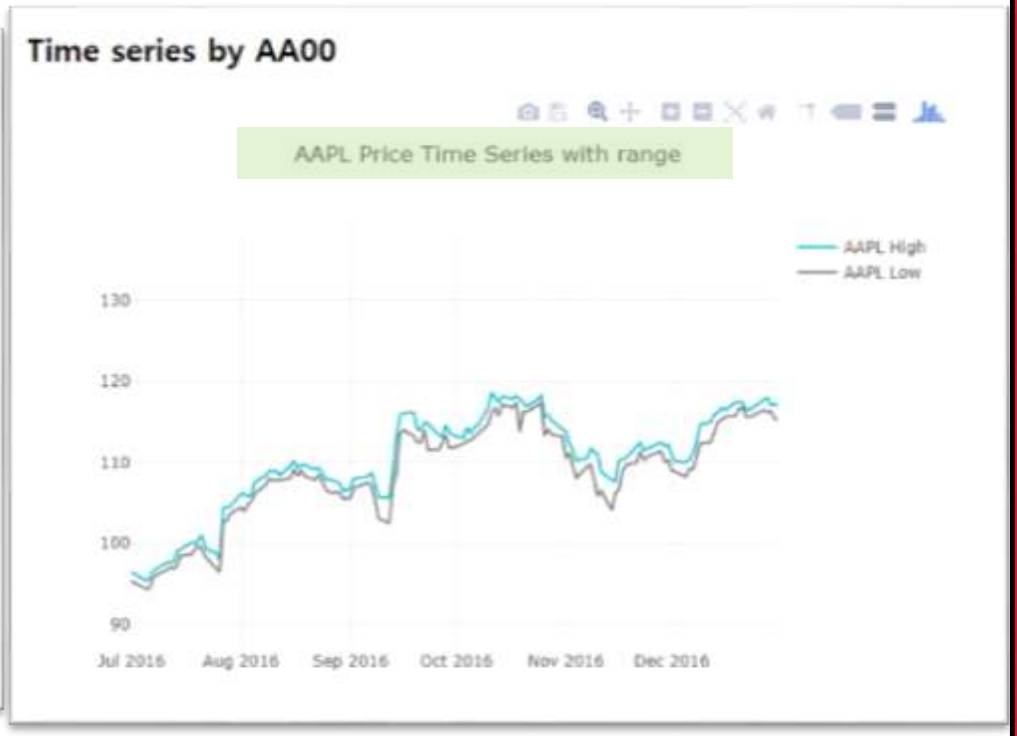
```
var data = [trace1,trace2];

var layout = {
    title: 'AAPL Price Time Series',
};

Plotly.newPlot('myDiv', data, layout);
```



A5.3.3.4 plotly.js: Time series

[2] Time series : financial data strings – set range

```
var data = [trace1,trace2];

var layout = {
    title: 'AAPL Price Time Series with range',
    xaxis: {
        range: ['2016-07-01', '2016-12-31'],
        type: 'date'
    },
    yaxis: {
        autorange: true,
        range: [86.8700008333, 138.870004167],
        type: 'linear'
    }
};

Plotly.newPlot('myDiv', data, layout);
```



날짜와 주가의 범위를 지정

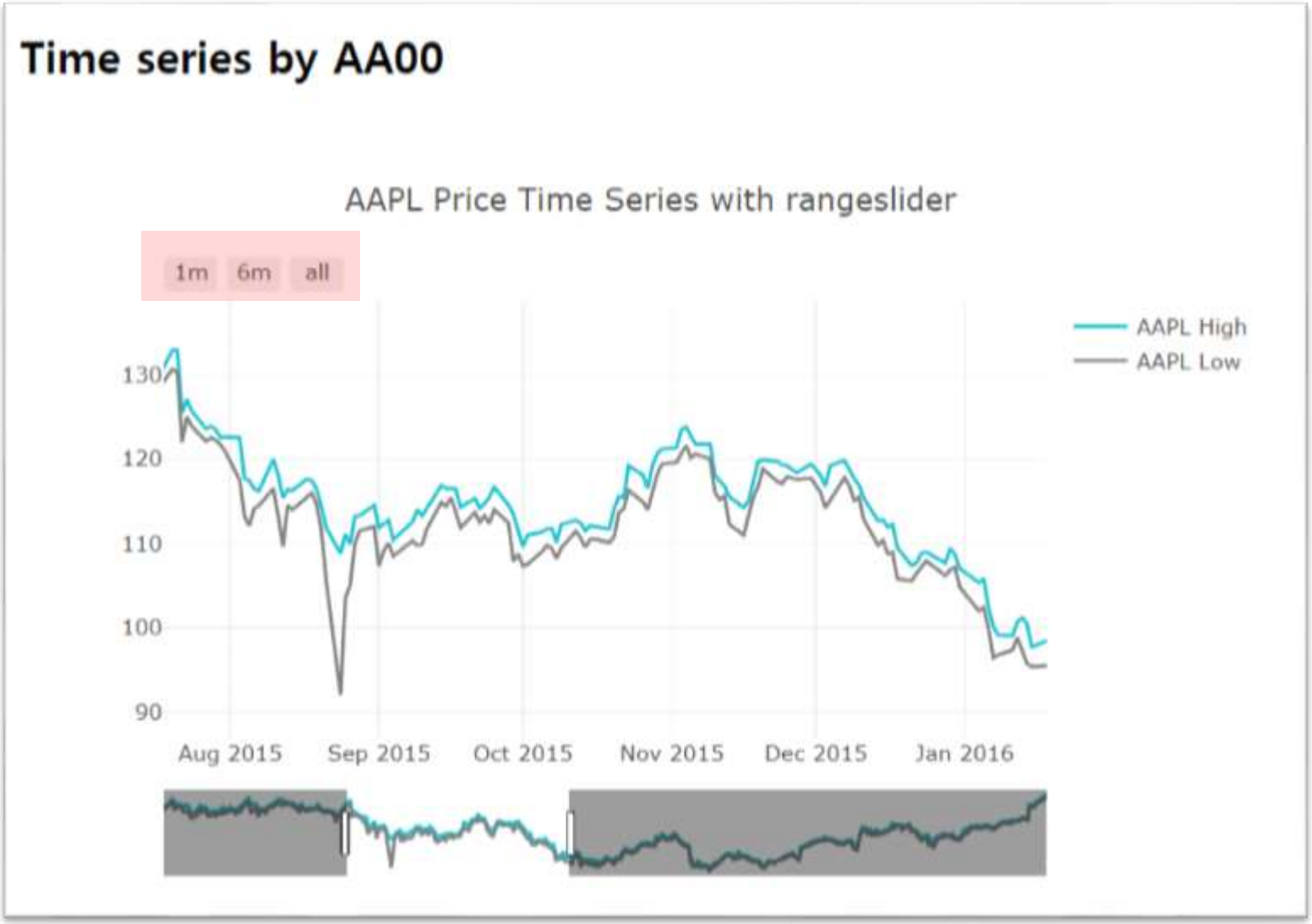
A5.3.3.5 plotly.js: Time series

[2] Time series : financial data strings – Range slider

```
var layout = {
    title: 'AAPL Price Time Series with rangeslider',
    xaxis: {
        autorange: true,
        range: ['2015-02-17', '2017-02-16'],
        rangeselector: {buttons: [
            {
                count: 1,
                label: '1m',
                step: 'month',
                stepmode: 'backward'
            },
            {
                count: 6,
                label: '6m',
                step: 'month',
                stepmode: 'backward'
            },
            {step: 'all'}
            ]},
            rangeslider: {range: ['2015-02-17', '2017-02-16']},
            type: 'date'
        },
        yaxis: {
            autorange: true,
            range: [86.8700008333, 138.870004167],
            type: 'linear'
        }
    };
```

A5.3.3.6 plotly.js: Time series

[2] Time series : financial data strings – Range slider



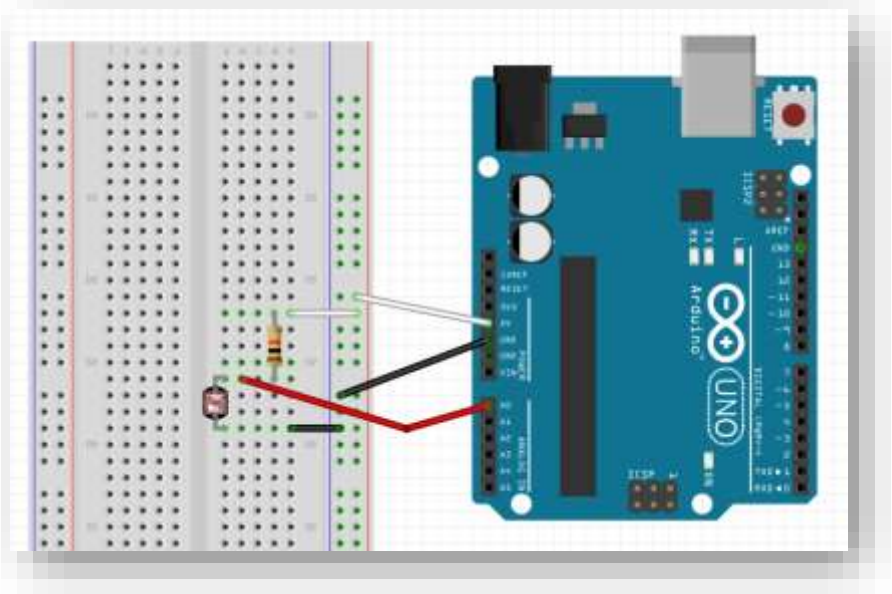
A5.3.4.1 plotly.js: Sensor time series

[3] Time series : my lux data

```
'2015-11-05 12:09:41.382',
'2015-11-05 12:09:42.380',
'2015-11-05 12:09:43.378',
'2015-11-05 12:09:44.377',
'2015-11-05 12:09:45.375',
'2015-11-05 12:09:46.389',
'2015-11-05 12:09:47.388',
'2015-11-05 12:09:48.386',
'2015-11-05 12:09:49.384',
'2015-11-05 12:09:50.383',
'2015-11-05 12:09:51.381',
'2015-11-05 12:09:52.380',
'2015-11-05 12:09:53.394',
'2015-11-05 12:09:54.392',
'2015-11-05 12:09:55.391',
'2015-11-05 12:09:56.389',
'2015-11-05 12:09:57.387',
'2015-11-05 12:09:58.386',
'2015-11-05 12:09:59.384',
'2015-11-05 12:10:00.398',
'2015-11-05 12:10:01.397',
```

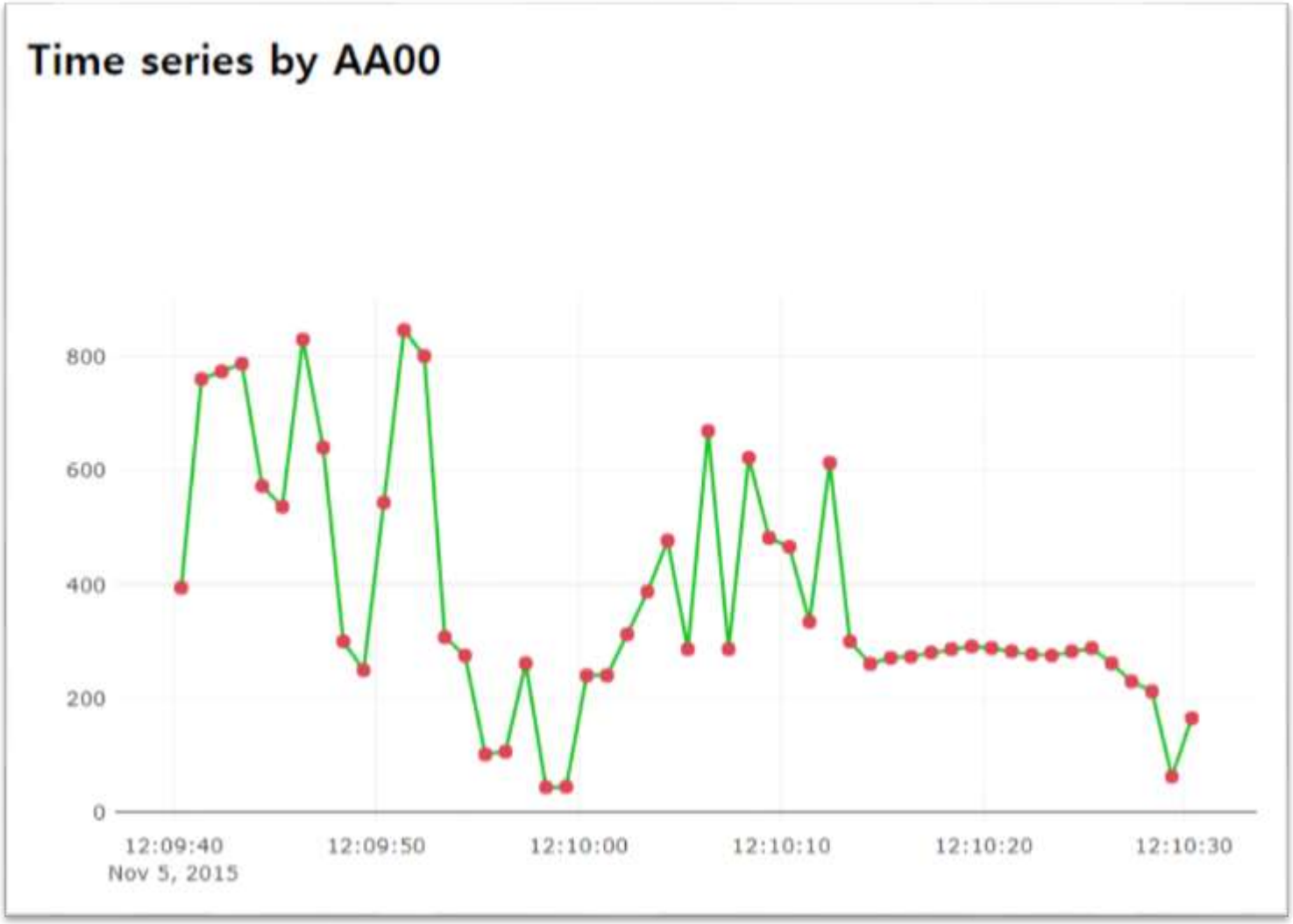
Data :

date,value



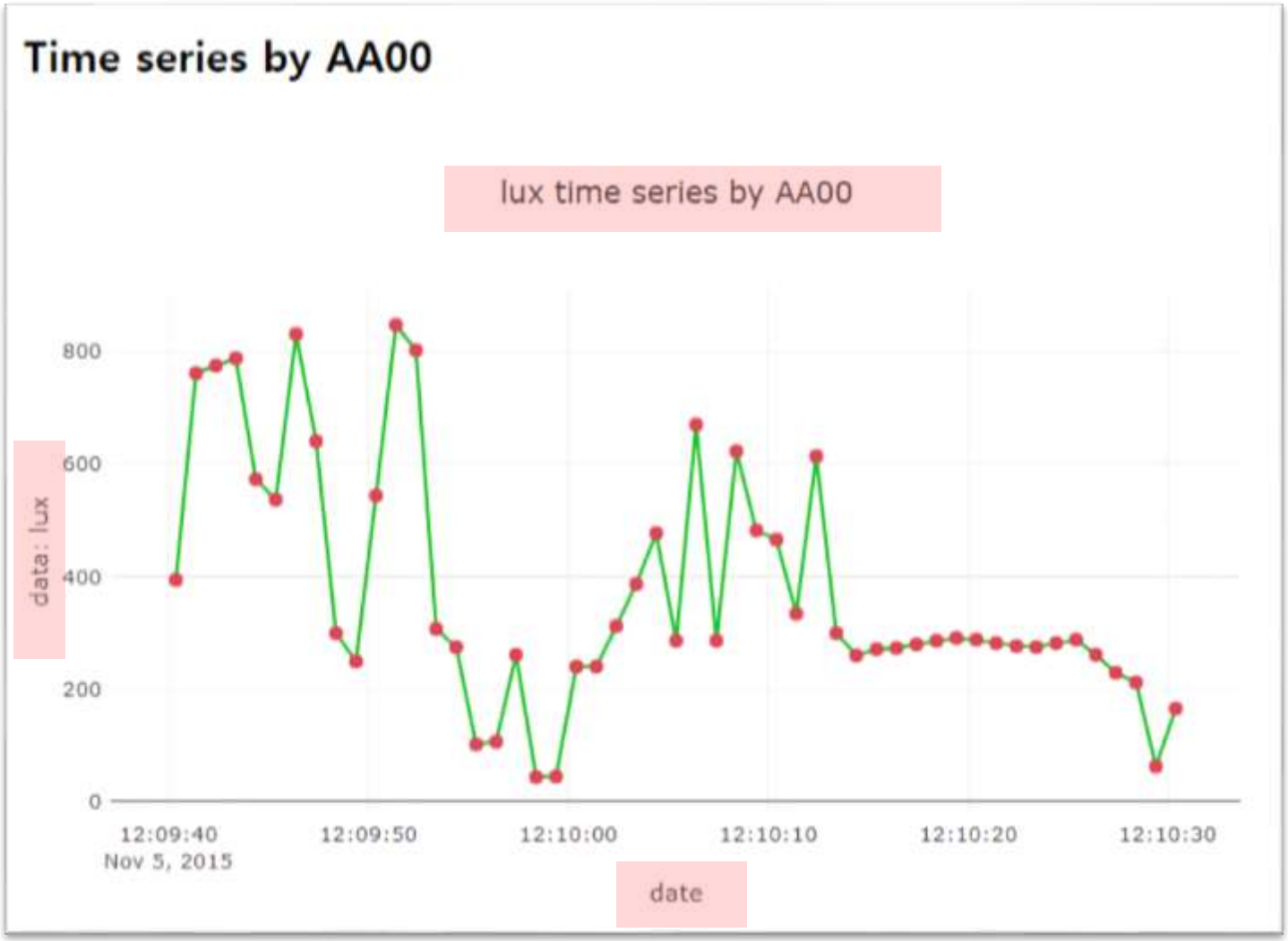
A5.3.4.2 plotly.js: Time series

[3] Time series : my lux data → DV_ts03_sensor_chart.html



A5.3.4.3 plotly.js: Time series

[3] Time series : my lux data – [DIY] → Set title and axis title

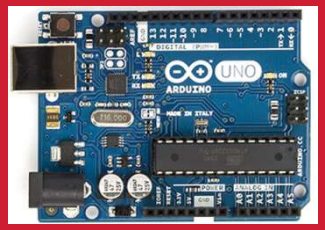


AAnn_lux_Time_Series.png

Project: Time series with Rangeslider

[Project-DIY] AAnn_lux_Rangelslider.html

AAnn_lux_Rangelslider.png



[Practice]

- [wk10]
  - Charts by plotly
  - Complete your plotly chart project
  - Upload file name : AAnn_Rpt07.zip

[wk10] Practice-07 AAnn_Rpt07.zip

- [Target of this week]
  - Complete your charts
  - Save your outcomes and compress them.

제출파일명 : **AAnn_Rpt07.zip**

- 압축할 파일들

① **AAnn_multi_signals_node.png**

② **AAnn_Chart_Layout.png**

③ **AAnn_Axis_Title.png**

④ **AAnn_Line_Dash_Dot.png**

⑤ **AAnn_lux_Time_Series.png**

⑥ **AAnn_lux_Rangeslider.png**

Email : chaos21c@gmail.com

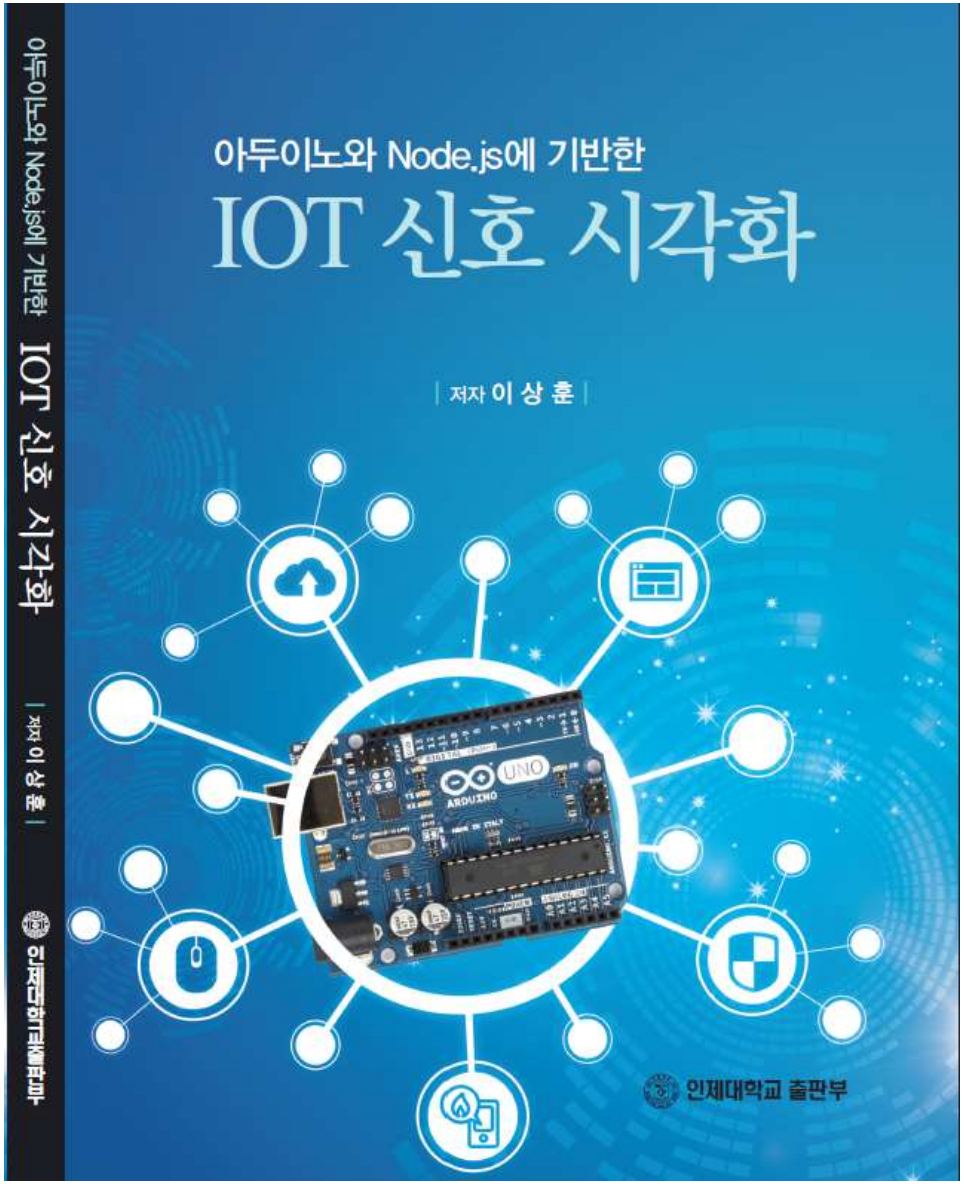
[제목 : id, 이름 (수정)]

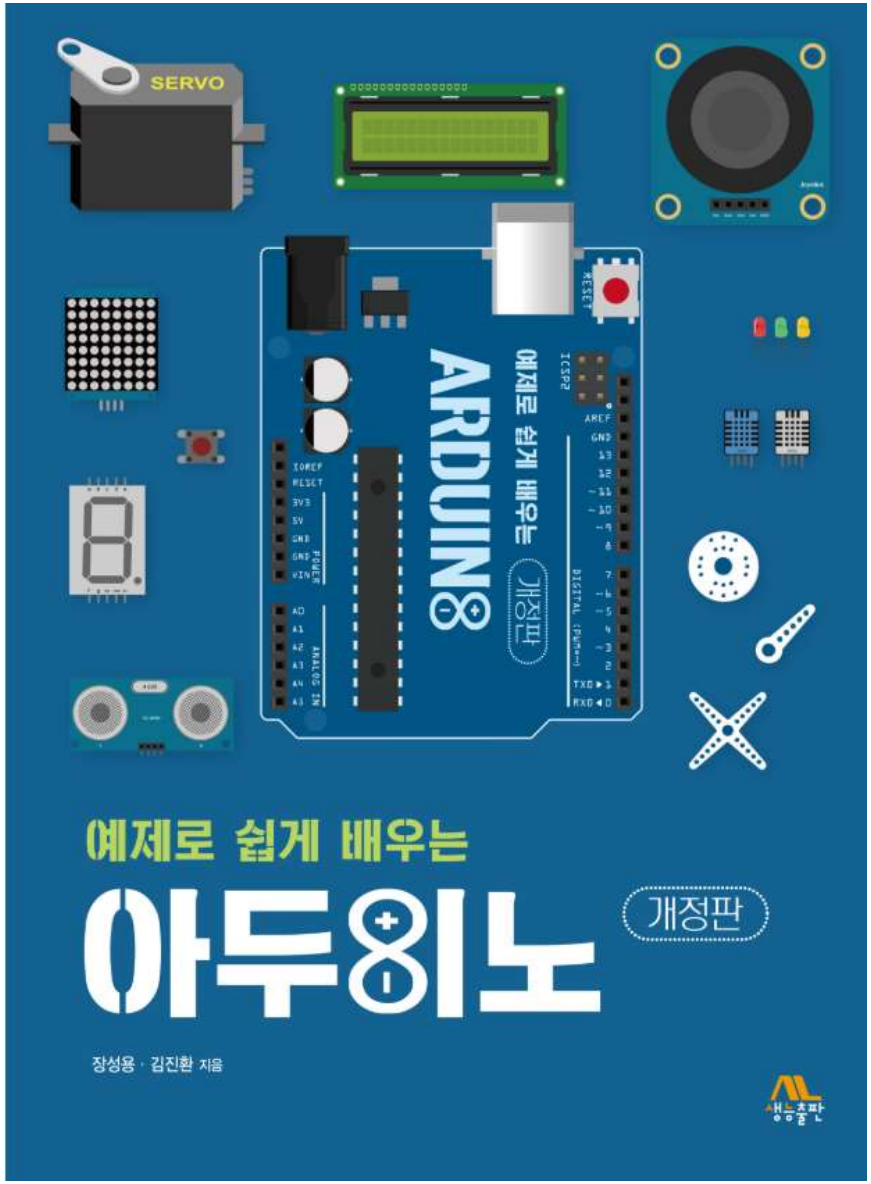
Lecture materials

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
아두이노와 Node.js에 기반한 IOT 신호 시각화
| 저자 이 상 훈 |
인제대학교출판부


SERVO
ARDUINO
예제로 쉽게 배우는
개정판
예제로 쉽게 배우는
아두이노
개정판
장성용 · 김진환 지음

Target of this class

Real-time Weather Station from sensors

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

Another target of this class