

HW-SW-Connectivity

[wk04]

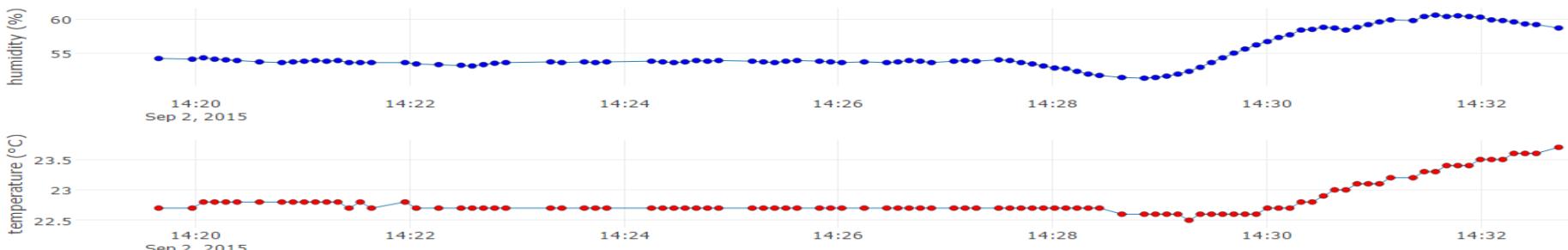
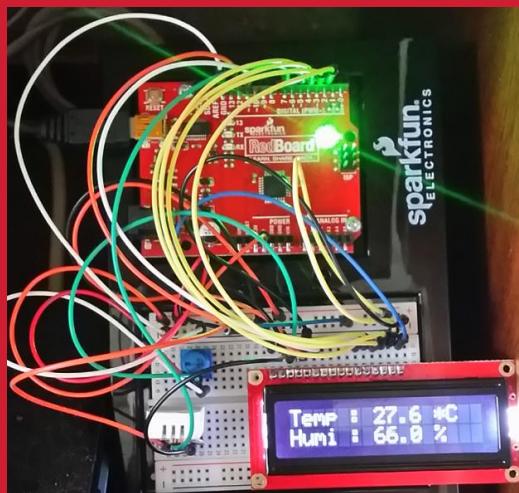
myApp server & Arduino intro

Basic HW and SW Integration using
Arduino & Javascript

COMSI, INJE University

2nd semester, 2017

Email : yish@inje.ac.kr





주간계획서

2017학년도 2학기

주간계획서		수업내용	과제물
주차	수업방법	수업내용	과제물
1	이론	교육과정 안내	가상강의 등록
2	이론/실습	모바일 서버 기초 : node.js 설치	
3	이론/실습	모바일 서버 프로그래밍 : node.js 응용	점검과제
4	이론/실습	기초 하드웨어: 아두이노 소개	점검과제
5	이론/실습	기초 하드웨어 프로그래밍: 아두이노 센서	프로젝트 1
6	이론/실습	시각화 프로그래밍: plotly.js	점검과제
7	이론/실습	아두이노 센서 신호 처리 및 시각화	프로젝트 1
8	시험 또는 실습과제	중간고사	
9	이론/실습	아두이노 센서 신호 시각화 고급 프로그래밍	점검과제
10	이론/실습	모바일 데이터베이스 I: mongoDB 설치	점검과제
11	이론/실습	모바일 데이터베이스 II: mongoDB 응용 프로그래밍, Compass 활용	프로젝트 1
12	이론/실습	모바일 클라인언트 프로그래밍 I: Angular.js 기초	점검과제
13	이론/실습	모바일 클라인언트 프로그래밍 II: Angular.js 응용 프로그래밍	점검과제
14	이론/실습	하드웨어와 소프트웨어 융합 IOT 프로젝트	프로젝트 IV
15	이론/실습	기말고사	

Weekly schedule of HSC– 2nd semester, 2017



- **wk01 : Introduction to class and enrollment in cyber class (Sublime text 3 install)**
- **wk02 : Basic mobile server : node.js install and test**
- **wk03 : Mobile server programming : node.js App**
- **wk04 : Basic HW : Arduino I. – circuit & programming, Arduino SW install**
- **wk05 : Basic HW : Arduino II. – sensor circuit & programming**
- **wk06 : Visualization using Javascript – ployly.js and gauge.js**
- **wk07 : Project-1 : Handling and visualization of signals from various sensors**
- **wk08 : Mid-term exam.**
- **wk09 : Advanced programming to visualize signals from sensors**
- **wk10 : Mobile database I : Mongo DB install**
- **wk11 : Mobile database II : Mongo DB App.**
- **wk12 : Mobile client programming I : Angular.js install**
- **wk13 : Mobile client programming II : Angular.js App.**
- **wk14 : Project-2 : IOT project fusing HW & SW**
- **wk15 : Final exam.**





[Practice]

◆ [wk03]

- NodeJS Server
- Save your results
- File name : AAnn_Rpt02.zip

wk03 : Practice-02 : AAnn_Rpt02.zip

◆ [Target of this week]

- Check examples of NodeJS server
- Save your outcomes and compress 5 figures

제출파일명 : **AAnn_Rpt02.zip**

- 압축할 파일들

- ① **AAnn_HTTP.png**
- ② **AAnn_TCP.png**
- ③ **AAnn_Upload.png**
- ④ **AAnn_Express.png**
- ⑤ **AAnn_Express_App.png**

email: chaos21c@gmail.com



What is targets of this week?

Node.js Server

- 1. http, tcp, file**
- 2. Express**



6. Node Server

Node Server I.

- 1. HTTP server**
- 2. TCP server**
- 3. File upload**



7. Node Server

Node Server II.

1. Express server

2. Full Express App

3. My Express App



7.1.1 Express server test

Step 1 : npm init

Step 2 : npm install --save express

Step 3 : Write Express code

Step 4 : Run app.js

Step 5 : http://localhost:3000



7.1.6 Express server test: run app.js

^B (실행 전 Tools > Cancel Build 해제 확인!)

D:\Portable\NodeJS\Portable\Data\aa00\node_server\expressTest\app.js (aa00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

aa00
 node_server
 expressTest
 node_modules
 app.js
 package.json
 FileUpload
 file_server.js
 HTTPServer
 index.js
 TCPServer
 client.js
 server.js
 wk03_code.txt
start

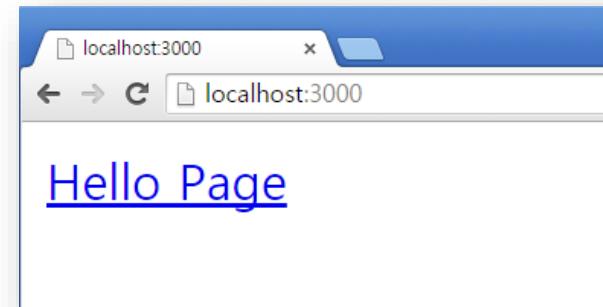
```
1 // app.js, express server
2 var express = require('express');
3 var app = express();
4 var port = 3000;
5
6 app.get('/', function(req, res) {
7   res.send('<a href="/hello">Hello Page</a>');
8 });
9
10 app.get('/hello', function(req, res) {
11   res.send('Hello HSC (하소연)');
12 });
13
14 var server = app.listen(port, function() {
15   console.log('Listening on port %d', server.address().port);
16 });
17
```

Listening on port 3000

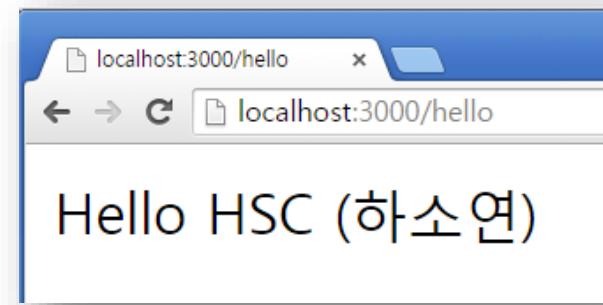


7.1.7 Express server test: test server routing

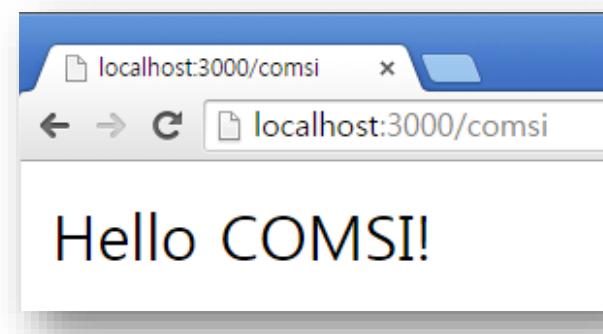
localhost:3000



localhost:3000/hello



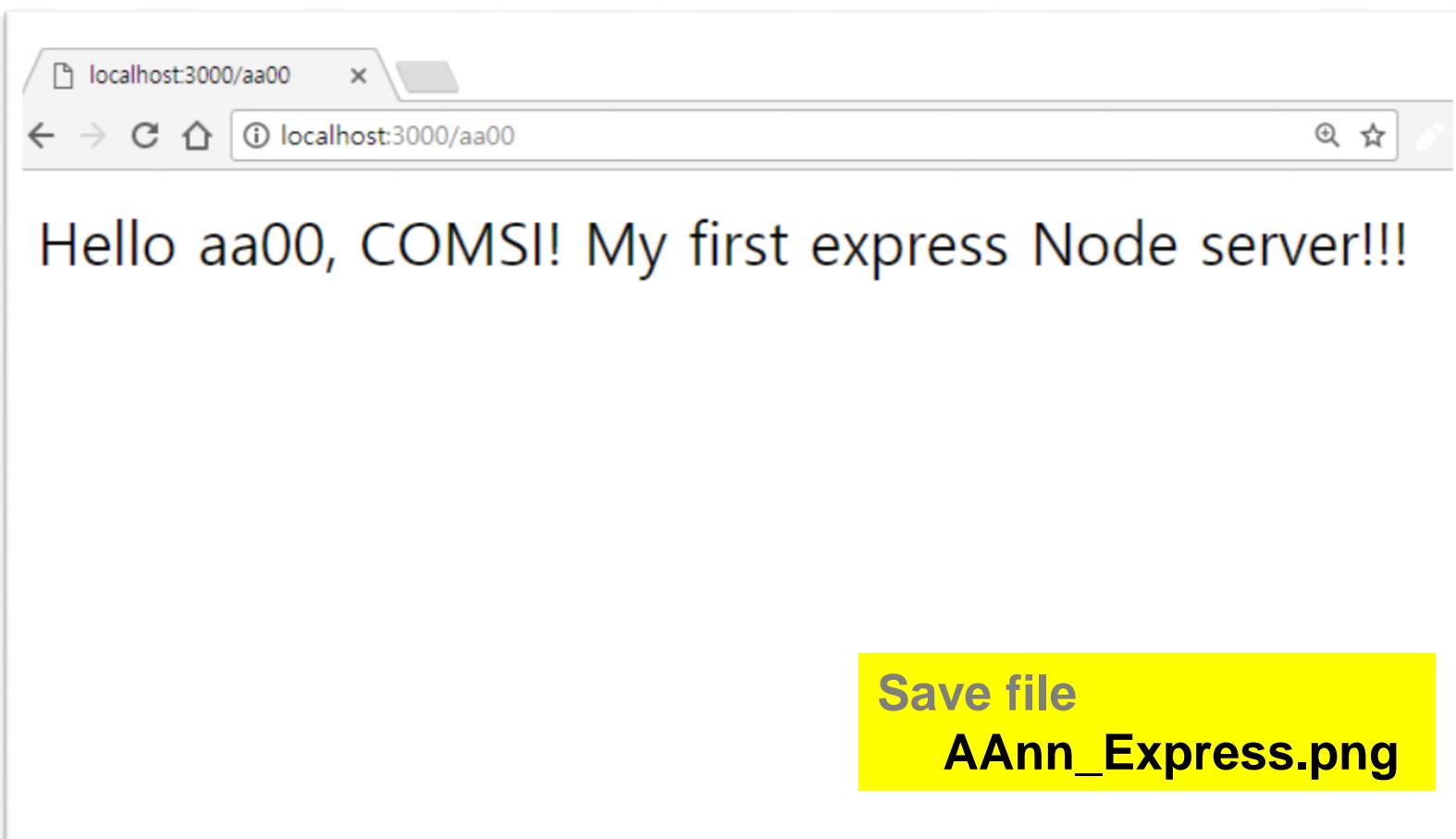
localhost:3000/comsi



^B (실행 전 Tools > Cancel Build 해제 확인!)



7.1.8 Express server test: run app.js



Routing: 라우팅은 애플리케이션 엔드 포인트(**URI**)의 정의, 그리고 **URI**가 클라이언트 요청에 응답하는 방식



7.2.1 Full Express App using generator

Step 1 : make folder ‘express’

Step 2 : go to the folder

Step 3 : `npm install -g express-generator`

Step 4 : `express`

Step 5 : `npm install`

Step 6 : `node ./bin/www`

Step 6 : <http://localhost:3000>



7.2.6 run bin/www

```
npm
D:\Portable\NodeJSPortable\Data\aa00\node_server\express>node ./bin/www
Listening on port 3000
^C
D:\Portable\NodeJSPortable\Data\aa00\node_server\express>npm start
> express@0.0.0 start D:\Portable\NodeJSPortable\Data\aa00\node_server\express
> node ./bin/www
Listening on port 3000
```

- 1. Install node modules that is defined in package.json**
- 2. npm install**
- 3. Modules were installed in node_modules subfolder.**

- 4. Run node express web**

node ./bin/www

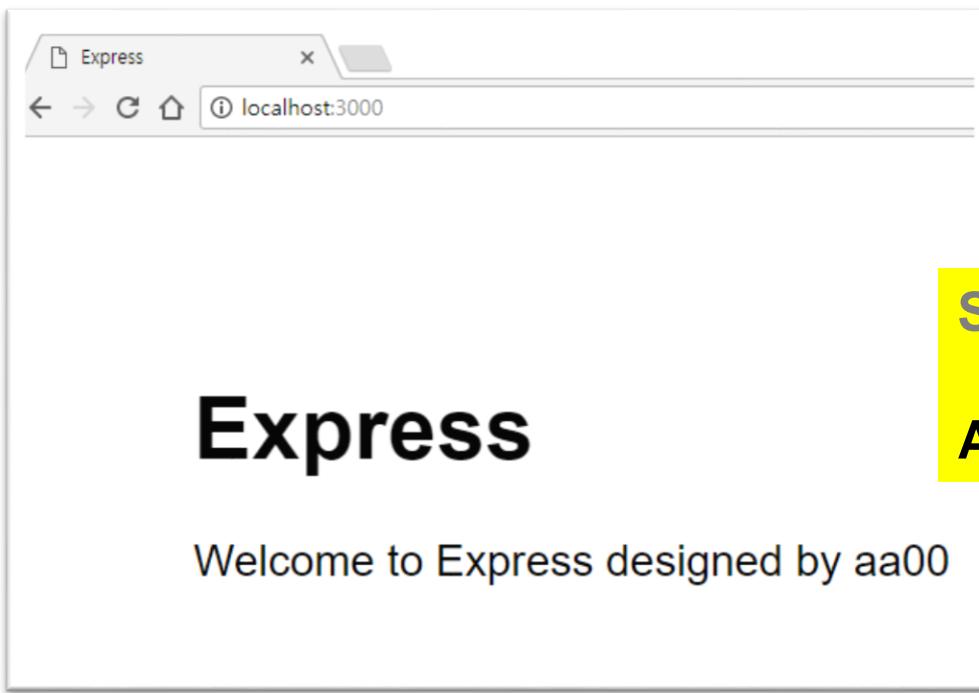
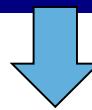
or

npm start



7.2.7 test app

```
node ./bin/www  
D:\Portable\NodeJSPortable\Data\aa00\node_server\express>node ./bin/www  
Listening on port 3000  
GET / 200 304.727 ms - 187  
GET /stylesheets/style.css 304 1.898 ms --  
GET / 304 18.247 ms --  
GET /stylesheets/style.css 304 0.640 ms --  
GET / 304 22.868 ms --  
GET /stylesheets/style.css 304 0.658 ms --
```



Save file

AAnn_Express_App.png



Node Server II.

- 1. Express server**
- 2. Full Express App**
- 3. My Express App**



7.3. Templating using Express-generator

Express options

-e, --ejs add ejs engine support (defaults to jade)

--hbs add handlebars engine support

-H, --hogan add hogan.js engine support

-c, --css <engine> add stylesheet <engine> support (less|stylus|compass)

-f, --force force on non-empty directory<engine></engine>

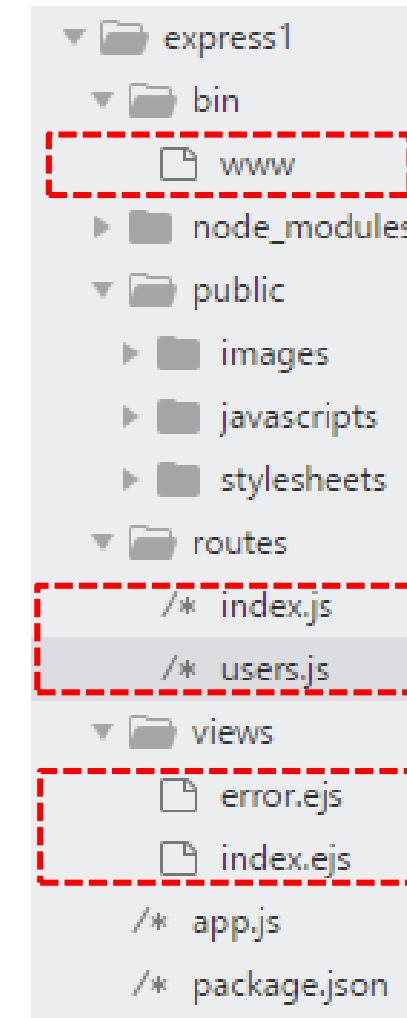


7.3.1 Templating using Express-generator

express express1 -e -c less

```
{  
  "name": "express1",  
  "version": "0.0.0",  
  "private": true,  
  "scripts": {  
    "start": "node ./bin/www"  
  },  
  "dependencies": {  
    "body-parser": "~1.17.1",  
    "cookie-parser": "~1.4.3",  
    "debug": "~2.6.3",  
    ["ejs": "~2.5.6"],  
    "express": "~4.15.2",  
    ["less-middleware": "~2.2.0"],  
    "morgan": "~1.8.1",  
    "serve-favicon": "~2.4.2"  
  }  
}
```

npm install





7.3.2 Run template: ejs

npm start or node ./bin/www

```
D:\Portable\NodeJSPortable\Data\aa00\node_server\express1>npm start
> express1@0.0.0 start D:\Portable\NodeJSPortable\Data\aa00\node_server\express1
> node ./bin/www

GET / 200 18.958 ms - 207
GET /stylesheets/style.css 200 26.509 ms - 87
GET /favicon.ico 404 2.619 ms - 1266
GET / 304 1.658 ms - -
GET /stylesheets/style.css 304 1.834 ms - -
```

The screenshot shows a code editor window with a dark theme. The file is named 'index.ejs'. The code is an EJS template:

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <h1><%= title %></h1>
    <p>Welcome to <%= title %></p>
  </body>
</html>
```

A yellow box highlights the file name 'index.ejs' at the top right of the editor window.

Express

Welcome to Express



7.4. myApp using Express-generator

myApp
using Express-generator



7.4.1 myApp using Express-generator

[root folder] express myApp

NodeJS

```
D:\Portable\NodeJSPortable\Data>express myApp
```

```
warning: the default view engine will not be jade in future releases
warning: use '--view=jade' or '--help' for additional options
```

```
create : myApp
create : myApp/package.json
create : myApp/app.js
create : myApp/public
create : myApp/views
create : myApp/views/index.jade
create : myApp/views/layout.jade
create : myApp/views/error.jade
create : myApp/routes
create : myApp/routes/index.js
create : myApp/routes/users.js
create : myApp/bin
create : myApp/bin/www
create : myApp/public/javascripts
create : myApp/public/images
create : myApp/public/stylesheets
create : myApp/public/stylesheets/style.css
```

```
install dependencies:
> cd myApp && npm install
```

```
run the app:
> SET DEBUG=myapp:* & npm start
```

NodeJS

```
D:\Portable\NodeJSPortable\Data>dir
D 드라이브의 폴더: DATA
폴더 일련 번호: 7A01-106A
```

```
D:\Portable\NodeJSPortable\Data 디렉터리
```

날짜	시간	크기	내용
2017-09-15	오후 04:41	<DIR>	.
2017-09-15	오후 04:41	<DIR>	.
2016-02-26	오전 04:36		6,148 .DS_Store
2017-09-09	오전 11:42	<DIR>	aa00
2017-09-13	오후 12:18		332 express
2017-09-13	오후 12:18		240 express.cmd
2017-09-15	오후 04:41	<DIR>	myApp
2017-09-13	오후 12:18	<DIR>	node_modules
2015-10-15	오전 07:21		296 npm
2015-10-15	오전 07:21		204 npm.cmd
2017-09-15	오후 04:32	<DIR>	npm_cache
2017-09-15	오후 04:28		89 PortableApps.com
e.ini			
2017-09-06	오후 01:40	<DIR>	settings
		6개 파일	7,309 바이트
		7개 디렉터리	925,147,037,696 바이트
			남음

```
D:\Portable\NodeJSPortable\Data>cd myApp
```

```
D:\Portable\NodeJSPortable\Data\myApp>dir
D 드라이브의 폴더: DATA
폴더 일련 번호: 7A01-106A
```

```
D:\Portable\NodeJSPortable\Data\myApp 디렉터리
```

날짜	시간	크기	내용
2017-09-15	오후 04:41	<DIR>	.
2017-09-15	오후 04:41	<DIR>	.
2017-09-15	오후 04:41		1,257 app.js
2017-09-15	오후 04:41	<DIR>	bin
2017-09-15	오후 04:41		325 package.json
2017-09-15	오후 04:41	<DIR>	public
2017-09-15	오후 04:41	<DIR>	routes
2017-09-15	오후 04:41	<DIR>	views
		2개 파일	1,582 바이트



7.4.2 myApp : package.json

D:\Portable\NodeJSPortable\Data\aa00\myApp\package.json (aa00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

```
aa00
└── MyApp
    ├── bin
    │   └── www
    ├── node_modules
    └── public
        ├── images
        ├── javascripts
        └── stylesheets
    └── routes
        ├── index.js
        └── users.js
    └── views
        ├── error.jade
        ├── index.jade
        ├── layout.jade
        ├── app.js
        └── package.json
└── node_server
└── start
```

package.json

```
1 {  
2     "name": "myapp",  
3     "version": "0.0.0",  
4     "private": true,  
5     "scripts": {  
6         "start": "node ./bin/www"  
7     },  
8     "dependencies": {  
9         "body-parser": "~1.17.1",  
10        "cookie-parser": "~1.4.3",  
11        "debug": "~2.6.3",  
12        "express": "~4.15.2",  
13        "jade": "~1.11.0",  
14        "morgan": "~1.8.1",  
15        "serve-favicon": "~2.4.2"  
16    }  
17}  
18}
```



7.4.3 myApp : app.js

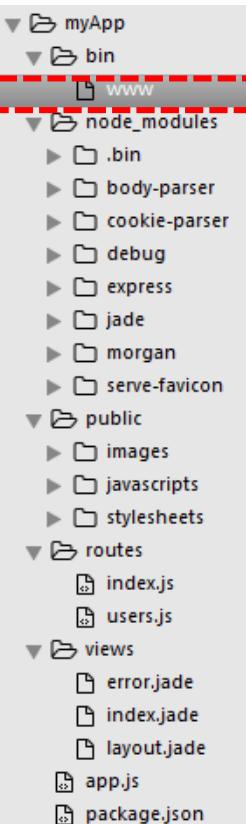
```
myApp
└── bin
    └── www
    └── node_modules
        ├── .bin
        ├── body-parser
        ├── cookie-parser
        ├── debug
        ├── express
        ├── jade
        ├── morgan
        └── serve-favicon
    └── public
        ├── images
        ├── javascripts
        └── stylesheets
    └── routes
        ├── index.js
        └── users.js
    └── views
        ├── error.jade
        ├── index.jade
        └── layout.jade
    └── app.js
    └── package.json
```

```
1 var express = require('express');
2 var path = require('path');
3 var favicon = require('serve-favicon');
4 var logger = require('morgan');
5 var cookieParser = require('cookie-parser');
6 var bodyParser = require('body-parser');
7
8 var routes = require('./routes/index');
9 var users = require('./routes/users');
10
11 var app = express();
12
13 // view engine setup
14 app.set('views', path.join(__dirname, 'views'));
15 app.set('view engine', 'jade');
16
17 // uncomment after placing your favicon in /public
18 //app.use(favicon(path.join(__dirname, 'public', 'favicon')));
19 app.use(logger('dev'));
20 app.use(bodyParser.json());
21 app.use(bodyParser.urlencoded({ extended: false }));
22 app.use(cookieParser());
23 app.use(express.static(path.join(__dirname, 'public')));
24
25 // Routes config
26 app.use('/', routes);
27 app.use('/users', users);
```

```
28 // catch 404 and forward to error handler
29 app.use(function(req, res, next) {
30     var err = new Error('Not Found');
31     err.status = 404;
32     next(err);
33 });
34
35 // error handlers
36 // development error handler
37 // will print stacktrace
38 if (app.get('env') === 'development') {
39     app.use(function(err, req, res, next) {
40         res.status(err.status || 500);
41         res.render('error', {
42             message: err.message,
43             error: err
44         });
45     });
46 }
47
48 // production error handler
49 // no stacktraces leaked to user
50 app.use(function(err, req, res, next) {
51     res.status(err.status || 500);
52     res.render('error', {
53         message: err.message,
54         error: {}
55     });
56 });
57
58 module.exports = app;
```



7.4.4 myApp : bin/www



```
1  #!/usr/bin/env node
2
3  /**
4   * Module dependencies.
5   */
6
7  var app = require('../app');
8  var debug = require('debug')('express:server');
9  var http = require('http');
10
11 /**
12  * Get port from environment and store in Express.
13 */
14
15 var port = normalizePort(process.env.PORT || '3000');
16 app.set('port', port);
17
18 /**
19  * Create HTTP server.
20 */
21
22 var server = http.createServer(app);
23 /**
24  * Listen on provided port, on all network interfaces.
25 */
26
27
28 server.listen(port);
29 server.on('error', onError);
30 server.on('listening', onListening);
31
32 console.log("Server Running on " + port +
33     "\nLaunch http://localhost:" + port);
```

```
Server Running on 3000.
Launch http://localhost:3000
```



7.4.5 routes/index.js (user-defined module)

routes/index.js

FOLDERS

- ▼ aa00
 - ▶ myApp
 - ▶ bin
 - ▶ www
 - ▶ node_modules
 - ▶ public
 - ▶ images
 - ▶ javascripts
 - ▶ stylesheets
 - ▶ routes
 - ▶ index.js
 - ▶ users.js
 - ▶ views
 - ▶ error.jade
 - ▶ index.jade
 - ▶ layout.jade
 - ▶ app.js
 - ▶ package.json

```
index.js *  
1 var express = require('express');  
2 var router = express.Router();  
3  
4 /* GET home page. */  
5 router.get('/', function(req, res, next) {  
6   res.render('index', { title: 'Express' });  
7 });  
8  
9 module.exports = router;  
10
```



7.4.6 routes/users.js (user-defined module)

routes/users.js

The screenshot shows a code editor with a sidebar containing a file tree. The tree includes 'aa00', 'myApp' (selected), 'bin', 'www', 'node_modules', 'public', 'routes' (selected), 'index.js', 'users.js' (selected), 'views', 'app.js', 'package.json', and 'node_server'. The main pane displays the 'users.js' file content:

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET users listing. */
5 router.get('/', function(req, res, next) {
6   res.send('respond with a resource');
7 });
8
9 module.exports = router;
10
```

A red dashed box highlights the line 'res.send('respond with a resource');'.



7.4.6.1 Define routes : app.use()

app.use()

FOLDERS

- aa00
- myApp
 - bin
 - www
- node_modules
- public
 - images
 - javascripts
 - stylesheets
- routes
 - index.js
 - users.js
- views
 - error.jade
 - index.jade
 - layout.jade
- app.js
- package.json

```
app.js
1 var index = require('./routes/index');
2 var users = require('./routes/users');
3
4 var app = express();
5
6 // view engine setup
7 app.set('views', path.join(__dirname, 'views'));
8 app.set('view engine', 'jade');
9
10 // uncomment after placing your favicon in /public
11 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
12 app.use(logger('dev'));
13 app.use(bodyParser.json());
14 app.use(bodyParser.urlencoded({ extended: false }));
15 app.use(cookieParser());
16 app.use(express.static(path.join(__dirname, 'public')));
17
18 app.use('/', index);
19 app.use('/users', users);
```

WEB routes로 **index, users**를 설정



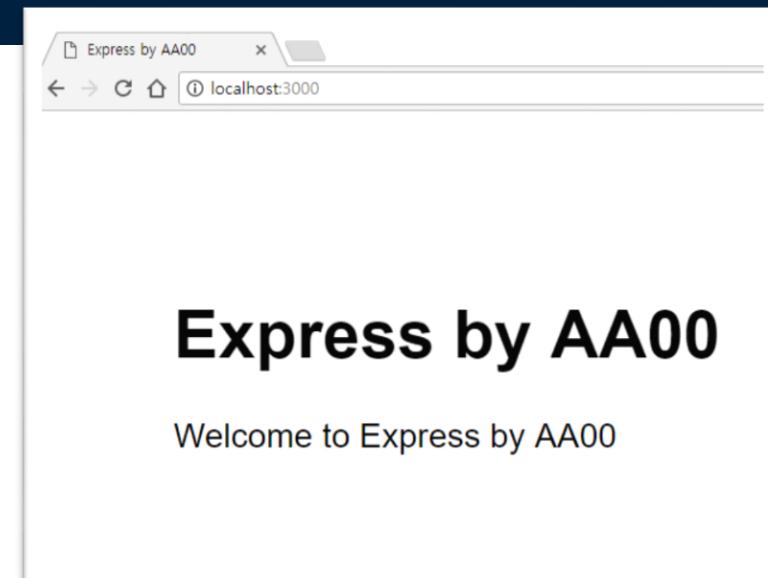
7.4.6.2 Define routes : app.use('/', index)

- ▼ aa00
- ▼ myApp
- ▼ bin
- www
- node_modules
- public
- ▼ routes
- index.js
- users.js
- views
- app.js
- package.json
- ▼ node_server

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function(req, res, next) {
6   res.render('index', { title: 'Express by AA00' });
7 });
8
9 module.exports = router;
10 |
```

NodeJS - node ./bin/www

```
D:\Portable\NodeJSPortable\Data>cd aa00
D:\Portable\NodeJSPortable\Data\aa00>cd myApp
D:\Portable\NodeJSPortable\Data\aa00\myApp>node ./bin/www
Server running on 3000.
Launch http://localhost:3000
GET / 304 996.710 ms -
GET /stylesheets/style.css 304 2.569 ms -
GET /favicon.ico 404 25.794 ms - 1235
```





7.4.6.3 Define routes : app.use('/users', users)

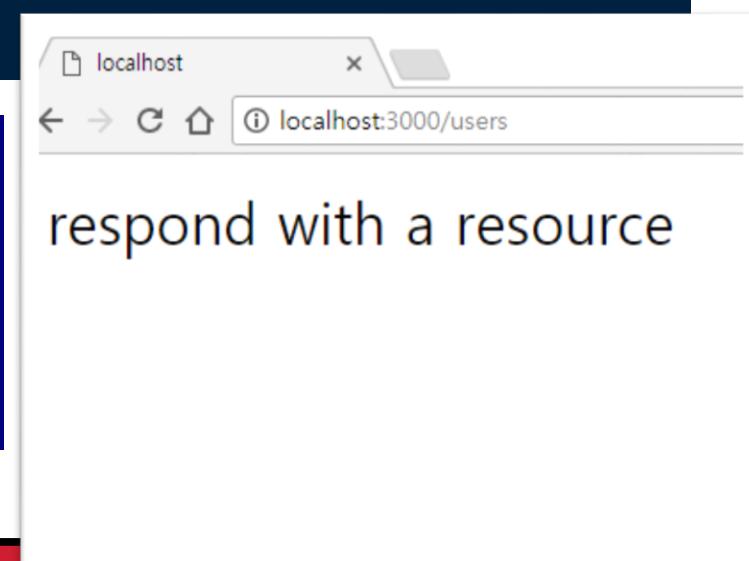
FOLDERS

- ▼ aa00
 - ▼ myApp
 - ▼ bin
 - www
 - node_modules
 - public
 - images
 - javascripts
 - stylesheets
 - routes
 - index.js
 - users.js
 - views
 - error.jade
 - index.jade
 - layout.jade
 - app.js
 - package.json

```
users.js
```

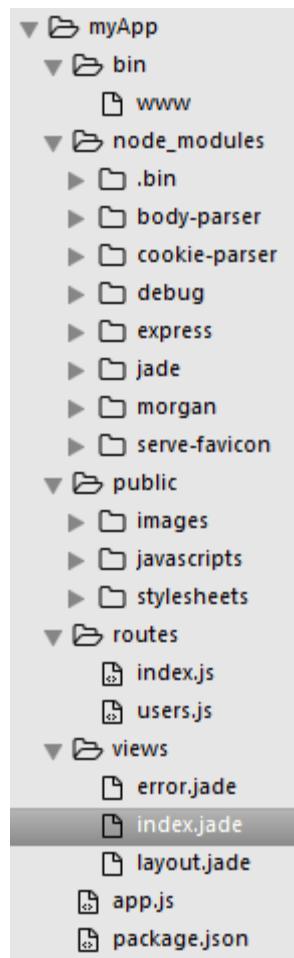
```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET users listing. */
5 router.get('/', function(req, res, next) {
6   res.send('respond with a resource');
7 });
8
9 module.exports = router;
10
```

```
D:\Portable\NodeJSPortable\Data\aa00\myApp>node ./bin/www
Server running on 3000.
Launch http://localhost:3000
GET / 304 996.710 ms -
GET /stylesheets/style.css 304 2.569 ms -
GET /favicon.ico 404 25.794 ms - 1235
GET /users 304 1.428 ms -
```





7.4.7 views/index.jade (jade html engine)



views/index.jade

```
1 extends layout
2
3 block content
4   h1= title
5   p Welcome to #{title}
```

views/layout.jade

```
1 doctype html
2 html
3   head
4     title= title
5     link(rel='stylesheet', href='/stylesheets/style.css')
6   body
7     block content
8
```



7.4.8 myApp : run myApp

1. npm install

```
cmd: npm
myapp@0.0.0 D:\Portable\NodeJSPortable\Data\aa00\myApp
+-- body-parser@1.17.2
| +-- bytes@2.4.0
| +-- content-type@1.0.4
| +-- debug@2.6.7
| +-- depd@1.1.1
| +-- http-errors@1.6.2
| | +-- inherits@2.0.3
| | +-- ionic-lite@0.4.15
| | +-- on-finished@2.3.0
| | | +-- ee-first@1.1.1
| | +-- qs@6.4.0
| | +-- raw-body@2.2.0
| | | +-- unpipe@1.0.0
| | | +-- type-is@1.6.15
| | | | +-- media-types@0.3.0
| | | | +-- mime-db@1.30.0
| +-- cookie-parser@1.4.3
| | +-- cookie@0.3.1
| | +-- cookie-signature@1.0.6
+-- debug@2.6.8
+-- ms@2.0.0
+-- express@4.15.4
| +-- accepts@1.3.4
| | +-- negotiator@0.6.1
+-- array-flatten@1.1.1
+-- content-disposition@0.5.2
+-- encodeurl@1.0.1
+-- escape-html@1.0.3
+-- etag@1.8.1
+-- finalhandler@1.0.4
+-- fresh@0.5.0
+-- merge-descriptors@1.0.1
+-- methods@1.1.2
+-- parseurl@1.3.2
+-- path-to-regexp@0.1.7
+-- proxy-addr@1.1.5
| +-- forwarded@0.1.2
+-- ipaddr.js@1.4.0
```

2. Run myApp !

- **^B on www (in SB3)**

```
Server running on 3000.  
Launch http://localhost:3000
```

- **node bin/www (in cmd)**

```
cmd: NodeJS - node ./bin/www
D:\Portable\NodeJSPortable\Data\aa00\myApp>node ./bin/www
Server running on 3000.
Launch http://localhost:3000
```

- **npm start (in cmd)**

```
cmd: npm
D:\Portable\NodeJSPortable\Data\aa00\myApp>npm start
> myapp@0.0.0 start D:\Portable\NodeJSPortable\Data\aa00\myApp
> node ./bin/www

Server running on 3000.
Launch http://localhost:3000
```



7.4.9 myApp : run myApp

D:\Portable\NodeJSPortable\Data\aa00\myApp\routes#index.js (aa00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

```
▼ aa00
  ▼ myApp
    ▼ bin
      www
    ▶ node_modules
    ▼ public
      images
      javascripts
      stylesheets
    ▼ routes
      index.js
      users.js
    ▼ views
      error.jade
      index.jade
      layout.jade
    app.js
    package.json
```

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function(req, res, next) {
6   res.render('index', { title: 'Express by AA00' });
7 });
8
9 module.exports = router;
```



Server running on 3000.

Launch <http://localhost:3000>

```
ESC [0mGET / ESC [32m200 ESC [0m659.076 ms - 19
ESC [0mGET /stylesheets/style.css ESC [32m200
ESC [0mGET / ESC [36m304 ESC [0m13.718 ms - -ES
ESC [0mGET /stylesheets/style.css ESC [36m304
```

Express by AA00

Welcome to Express by AA00

Save as
AAnn_App.png



[Practice]

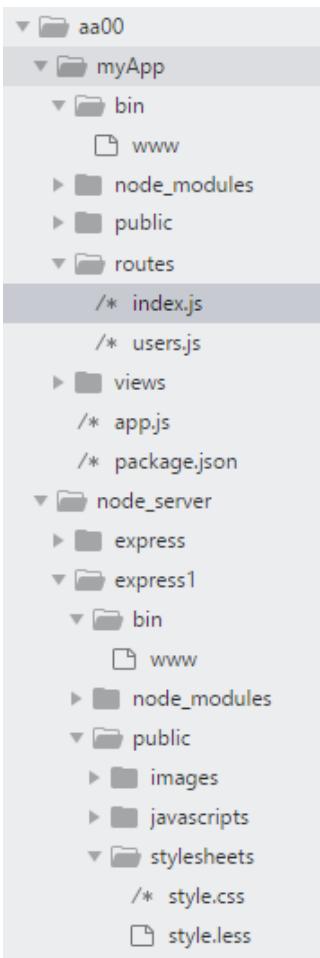
◆ [wk04]

- My Express App: multi-routing
- Add a new route in index.js
- Upload file name : AAnn_Rpt03.zip



[Practice-1] Modify routes/index.js

Add a new route '**/aann**' in index.js, multi-routing.



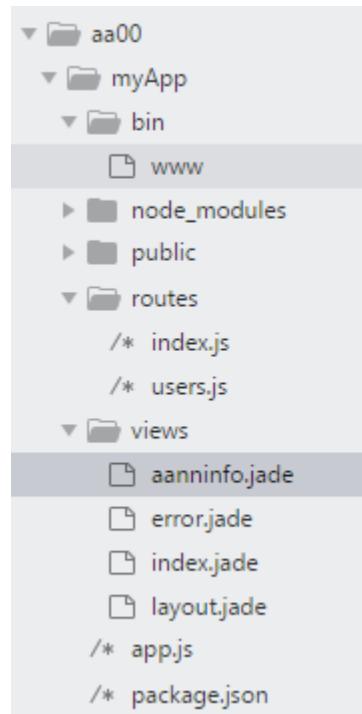
```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function(req, res, next) {
6   res.render('index', { title: 'Express by AA00' });
7 });
8
9 /* GET myInfo page, by /aann -> multi-routing */
10 router.get('/aann', function(req, res, next) {
11   res.render('aanninfo', { title: 'Express App by AA00',
12                         id: 'AAnn',
13                         name: 'COMSI' });
14   // views/aanninfo.jade
15 });
16
17
18 module.exports = router;
19
```

Use renderer aanninfo.jade in views folder



[Practice-2] Add aanninfo.jade in views folder

Add aanninfo.jade in views folder



```
1 extends layout
2
3 block content
4   h1= title
5   p Welcome to #{title}
6   p My ID : #{id}
7   center Developed by #{name}
8
```

Save as
AAnn_MyInfo_jade.png



[Practice-3] Result: your Info

Rerun node ./bin/www

localhost:3000/aann

The screenshot shows a web browser window titled "Express App by AA00". The address bar displays "localhost:3000/aann". The main content area of the browser shows the following text:

Express App by AA00

Welcome to Express App by AA00

My ID : AAnn

Developed by COMSI

Save as

AAnn_MyInfo.png

wk04 : Practice-03 : AAnn_Rpt03.zip

◆ [Target of this week]

- Complete your projects
- Save your outcomes and compress 3 figures

제출파일명 : **AAnn_Rpt03.zip**

- 압축할 파일들

- ① **AAnn_MyApp.png**
- ② **AAnn_MyInfo_jade.png**
- ③ **AAnn_MyInfo.png**

email: chaos21c@gmail.com

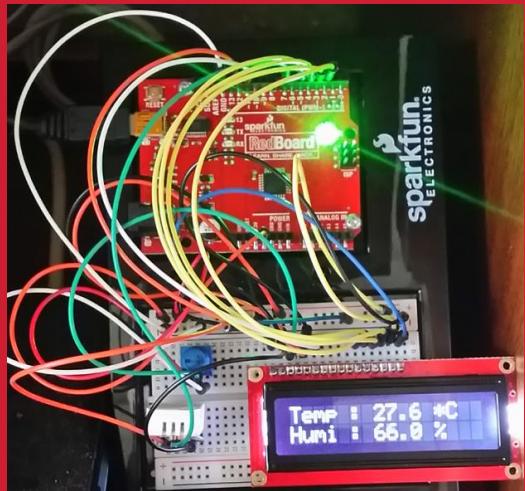


HW-SW-Connectivity

[wk04]

Arduino intro

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

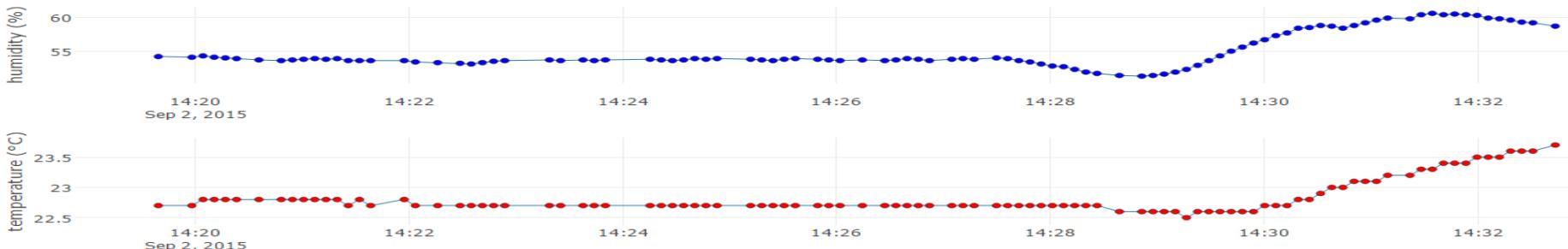


Basic HW and SW Integration using
Arduino & Javascript

COMSI, INJE University

2nd semester, 2017

Email : yish@inje.ac.kr





Arduino

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

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



A0.0 Arduino products

ENTRY LEVEL	ARDUINO/GENUINO UNO	ARDUINO PRO	ARDUINO PRO MINI	ARDUINO/GENUINO MICRO		
	ARDUINO NANO	ARDUINO/GENUINO STARTER KIT	ARDUINO BASIC KIT			
	ARDUINO MOTOR SHIELD					
ENHANCED FEATURES	ARDUINO/GENUINO MEGA	ARDUINO ZERO	ARDUINO DUE	ARDUINO PROTO SHIELD		
INTERNET OF THINGS	ARDUINO YÚN	ARDUINO ETHERNET SHIELD	ARDUINO GSM SHIELD	ARDUINO WIFI SHIELD 101		
WEARABLE	ARDUINO GEMMA	LILYPAD ARDUINO USB	LILYPAD ARDUINO MAIN BOARD			
	LILYPAD ARDUINO SIMPLE	LILYPAD ARDUINO SIMPLE SNAP				
3D PRINTING	MATERIA 101					

BOARDS

MODULES

SHIELDS

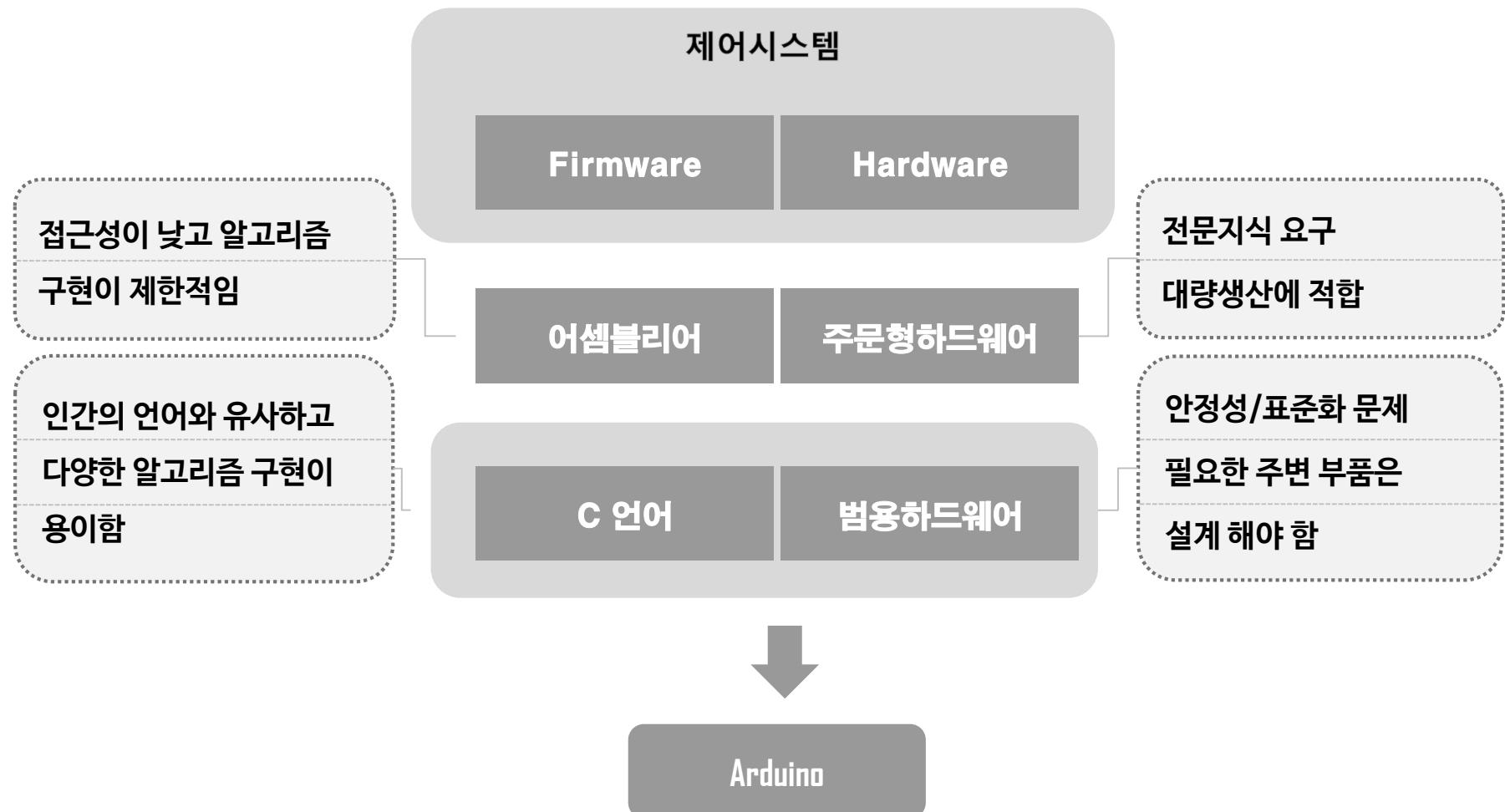
KITS

ACCESSORIES

COMING NEXT



0.1 Arduino 란?



0.1 Arduino 란?

2005년 Italy의 Massimo Banzi & David Cuartielles에 의해 개발

예술가
취미생활
학생

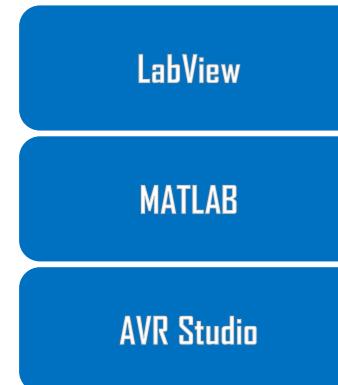
전자공학
교육

누구나
쉽게
사용
가능한
제어장치

오픈소스
하드웨어

GSM Wifi Ethernet
Motor drive
등의 월드 제공

다양한
라이브러리



범용
하드웨어
의
표준

0.2 Arduino hardware

Arduino Board



Shield (통신모듈)

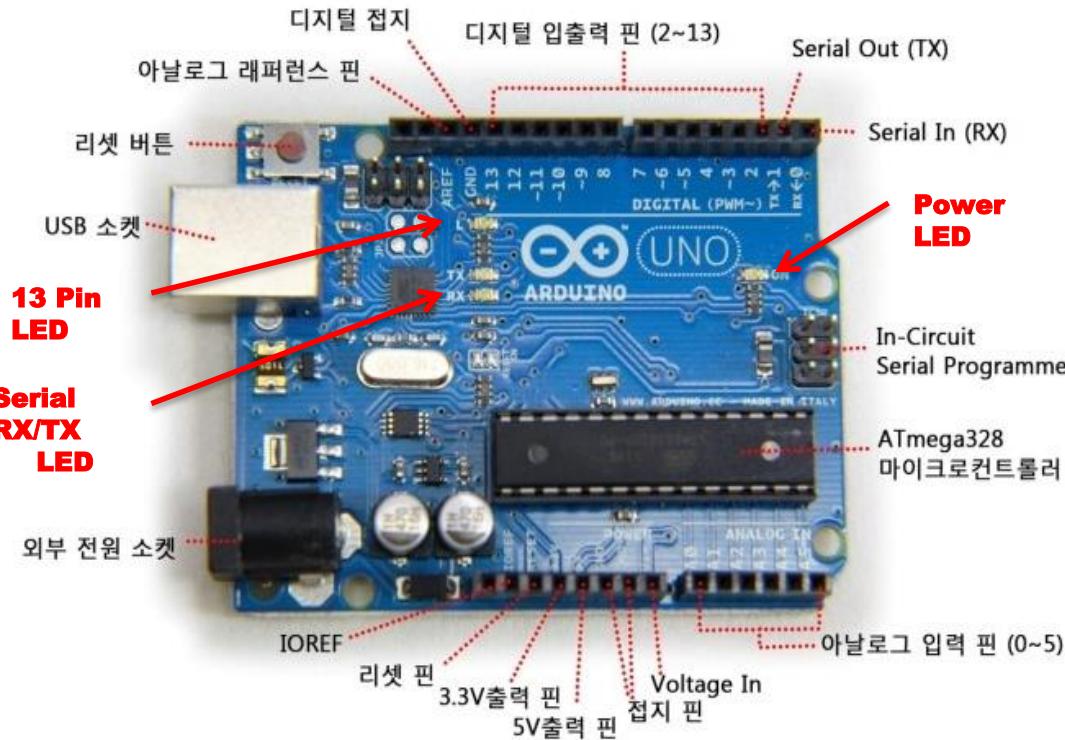


Sensor
광,온도,
습도...

Actuator
LED,모터,
스피커...

Accessory
스크린,
USB...

0.2 Arduino hardware



✓ Arduino UNO R3

- ATmega328 microcontroller
- Input voltage: 7~12V
- 14 Digital I/O Pins (6 PWM outputs)
- 6 Analog Inputs
- 32KB Flash Memory
- 16Mhz Clock Speed



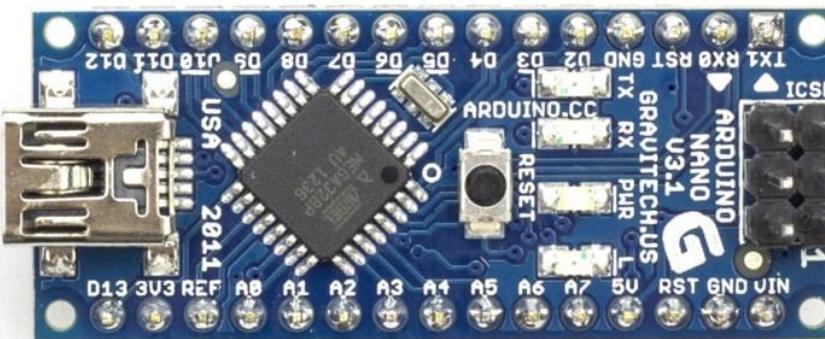
0.2 Arduino hardware



✓ Arduino MEGA 2560

- ATmega2560 microcontroller
- Input voltage: 7~12V
- 54 Digital I/O Pins (6 PWM outputs)
- 16 Analog Inputs
- 4 UARTs
- 256KB Flash Memory
- 16Mhz Clock Speed

0.2 Arduino hardware



✓ Arduino Pro NANO

- ATmega168/328 microcontroller
- Input voltage: 7~12V
- 14 Digital I/O Pins (6 PWM outputs)
- 8 Analog Inputs
- 16KB Flash Memory
- 16Mhz Clock Speed

0.2 Arduino hardware - IOT



ARDUINO YUN



ARDUINO ETHERNET SHIELD



ARDUINO GSM SHIELD



ARDUINO WIFI SHIELD 101



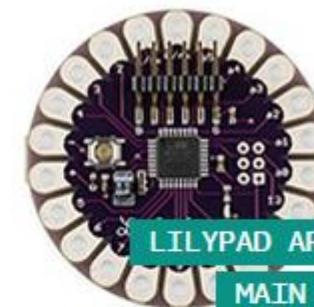
0.2 Arduino hardware – Wearable



ARDUINO GEMMA



LILYPAD ARDUINO
SIMPLE



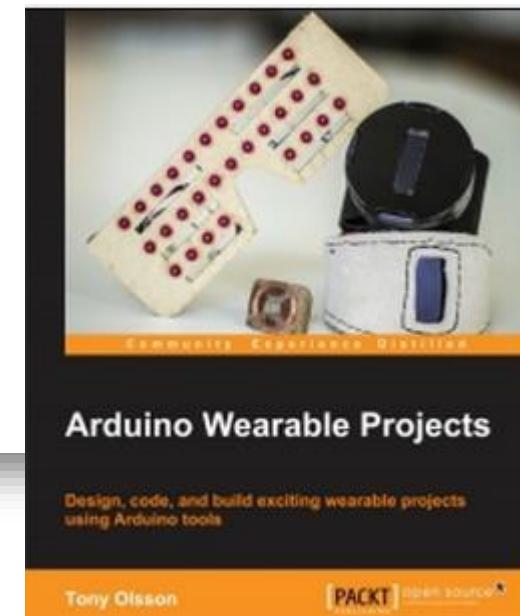
LILYPAD ARDUINO
MAIN BOARD



LILYPAD ARDUINO USB



LILYPAD ARDUINO
SIMPLE SNAP



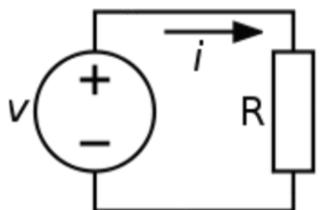
Arduino Wearable Projects

Design, code, and build exciting wearable projects using Arduino tools

Tony Olsson

PACKT open source

0.3 전압, 전류, 저항



전압
[V]

- ✓ 전위가 높은 쪽과 낮은 쪽의 차이

- ✓ 1쿨롱(coulomb: 전하의 단위)의 전하가 갖고 있는 에너지
- ✓ Arduino에서는 직류 3.3[V]와 5[V]를 지원

전류
[A]

- ✓ 1초당 1쿨롱의 전하가 단위 면적을 통과했을 때를 1[A]로 정의

- ✓ Arduino에서는 $1/1000$ [A] 단위인 [mA]를 사용

저항
[Ω]

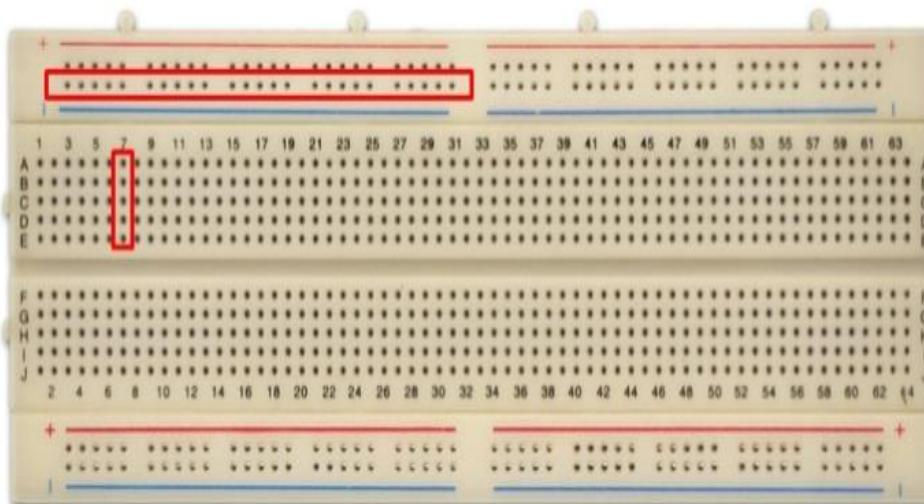
- ✓ 전류의 흐름을 방해하는 정도를 나타냄

- ✓ 색 띠나 숫자로 값을 표시

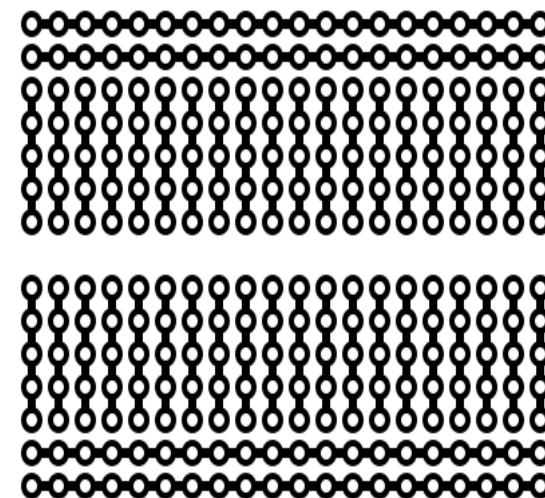
- ✓ Arduino에서는 칩(chip) 형태의 저항이 사용

0.4 브레드 보드 (Bread board)

시제품 제작이나 실험용 와이어를 보드에 꽂아 사용

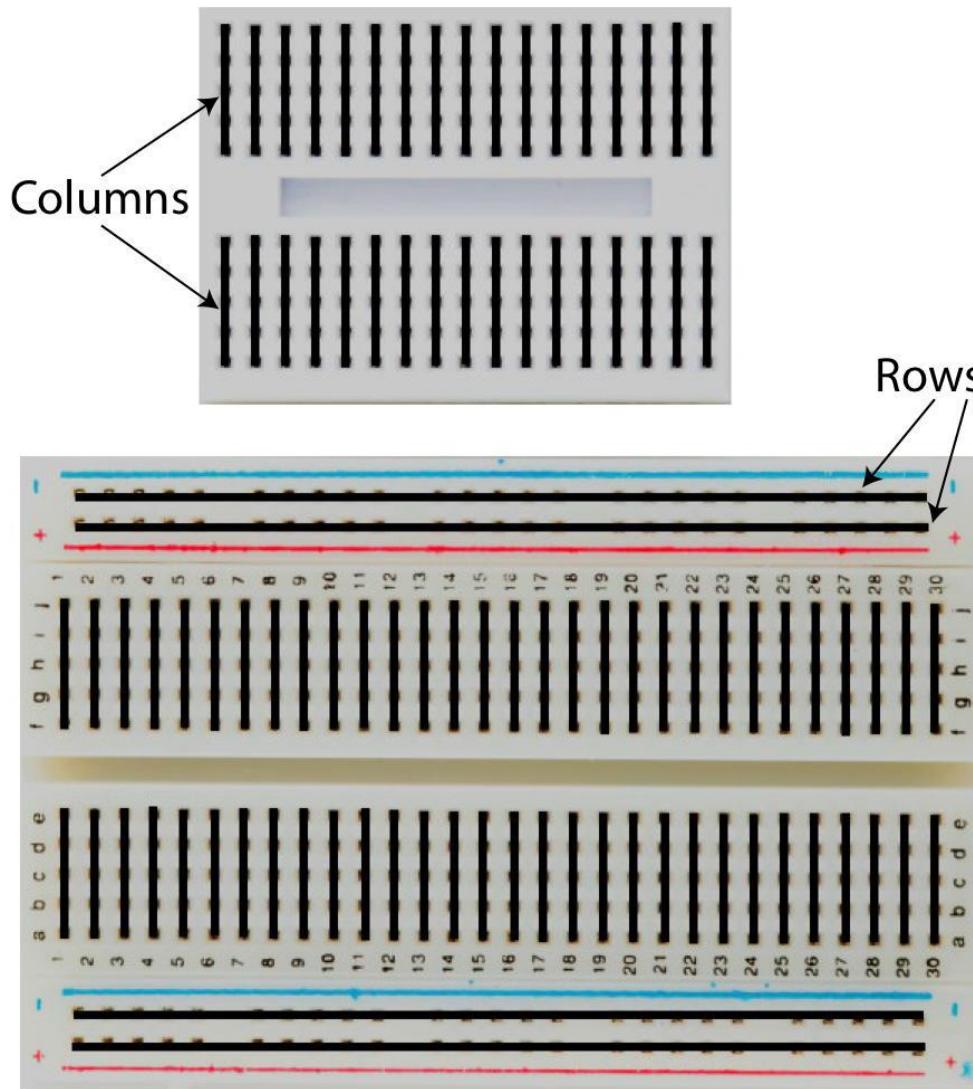


빨간색 묶음 훌끼리 내부회로가 연결되어 있음



내부 결선

0.4 브레드 보드 (Bread board)





0.5 Arduino 3D printing

3D Printing

The Arduino approach to 3D printing is represented by Materia 101, a printer that allows you to start experimenting with this amazing technology in the easiest way.





Arduino

- **Inexpensive**
- **Cross-platform**
- **Simple & clear programming**
- **Open source & extensible software**
- **Open source & extensible hardware**
- **Simple HW-SW Connectivity**

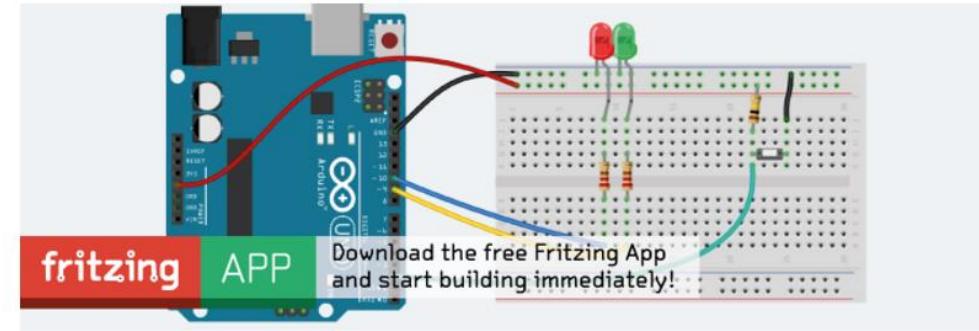


Arduino SW

[fritzing.org Fritzing](http://fritzing.org/home/)

fritzing electronics made easy

Projects Parts Download Learning Services Contribute FORUM FAB



fritzing APP Download the free Fritzing App and start building immediately!

Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone. We offer a software tool, a community website and services in the spirit of Processing and Arduino, fostering a creative ecosystem that allows users to document their prototypes, share them with others, teach electronics in a classroom, and layout and manufacture professional pcbs.

Download and Start
Download our latest version 0.9.3b released on June 2, 2016 and start right away.

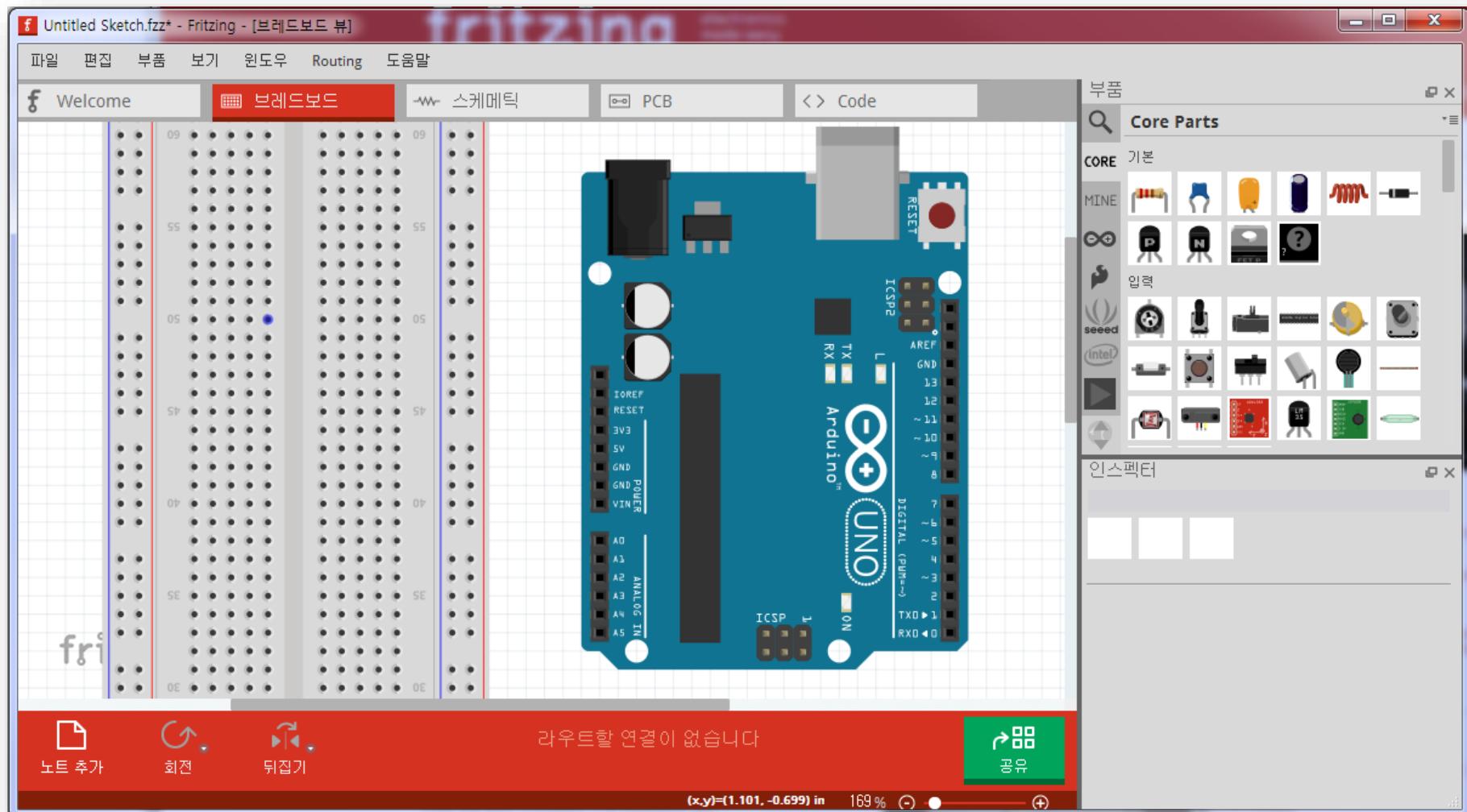
Produce your own board
With Fritzing Fab you can easily and inexpensively turn your circuit into a real, custom-made PCB. Try it out now!

Participate
Fritzing can only act as a creative platform if many



Fritzing configuration

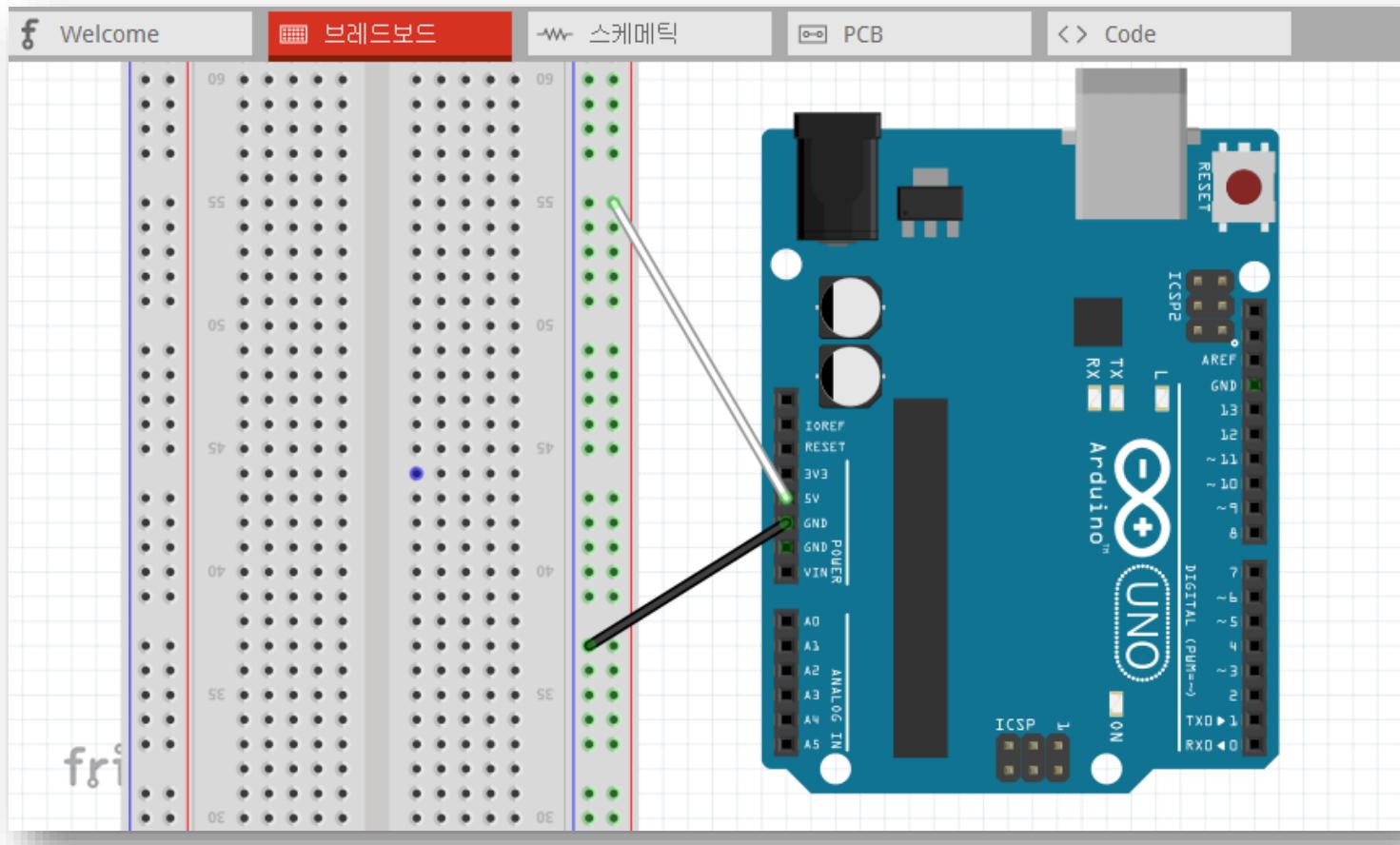
[Fritzing] configuration





Fritzing configuration

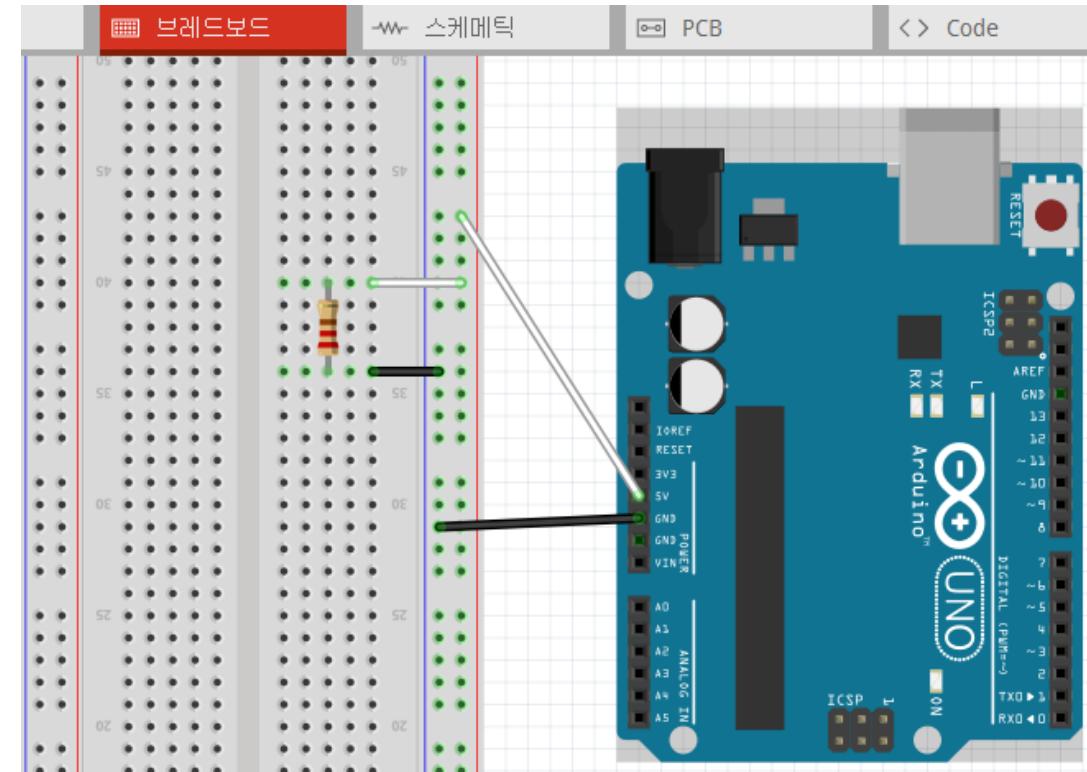
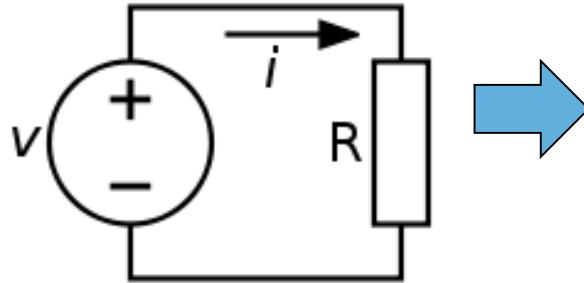
[Fritzing] power





DIY : my 1st circuit

[Fritzing] Simple R circuit

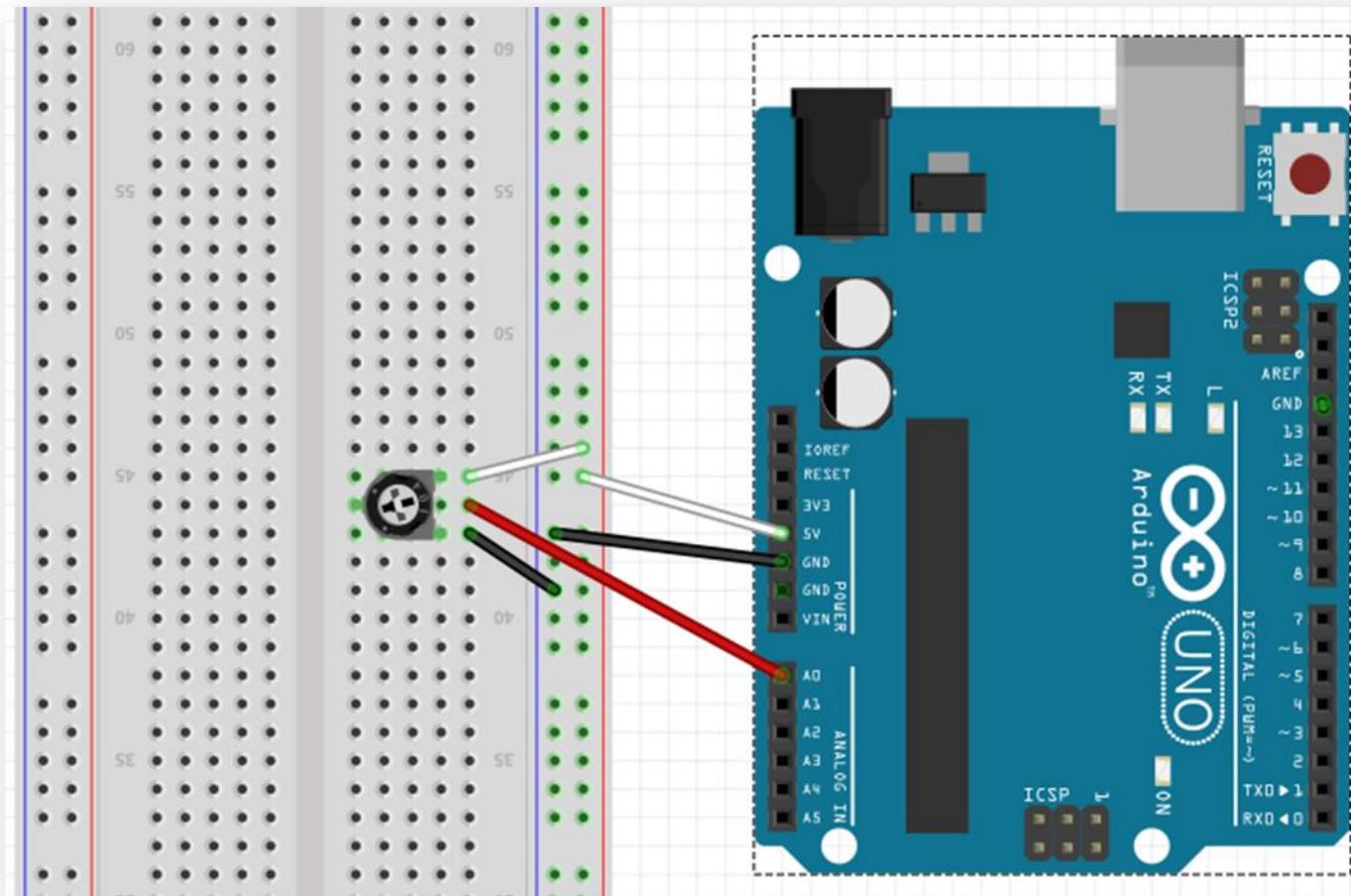




Arduino circuits



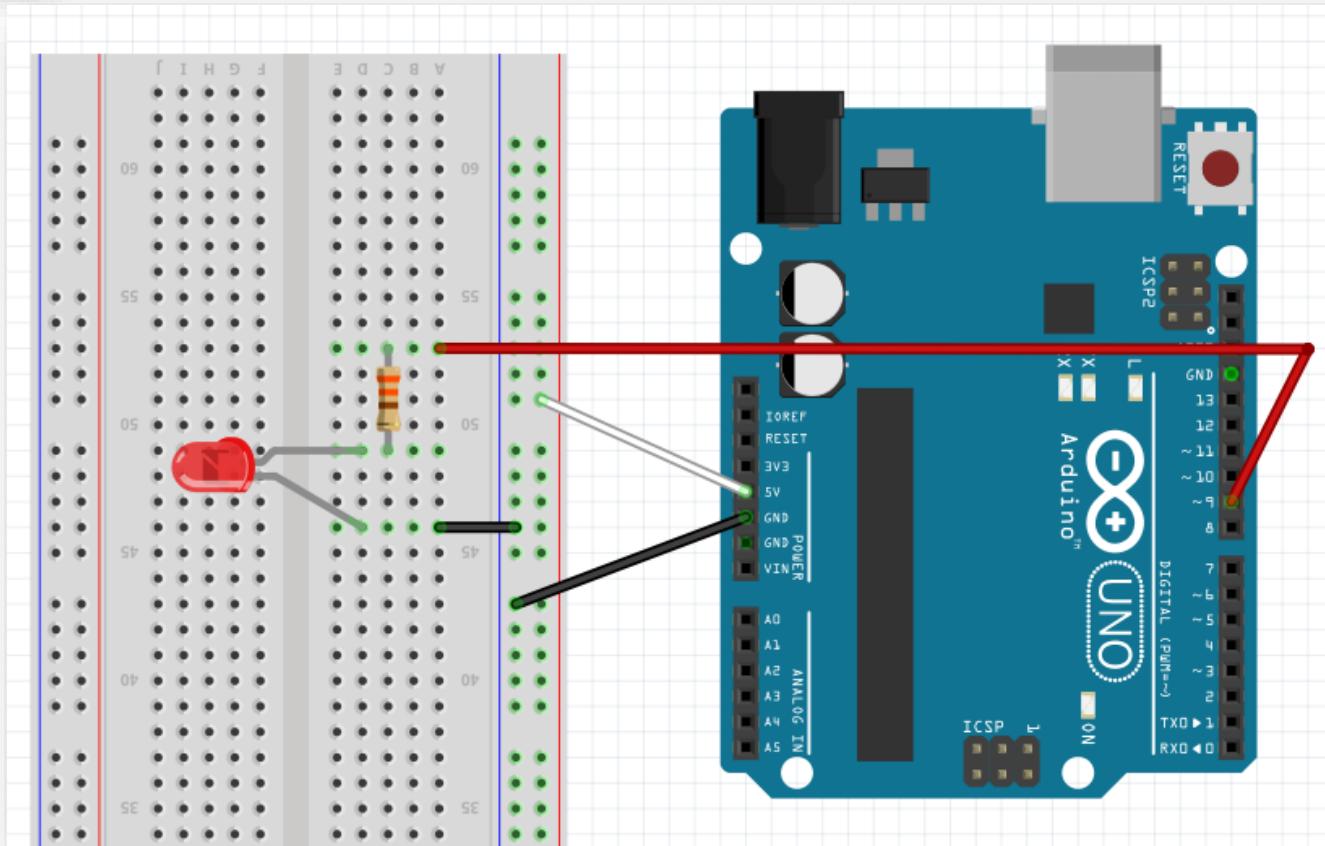
0.6.A1 Potentiometer (가변 저항기)



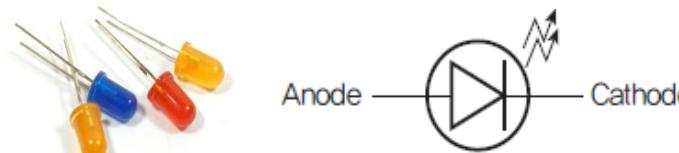
Parts : 가변저항기



0.6.A2 single LED



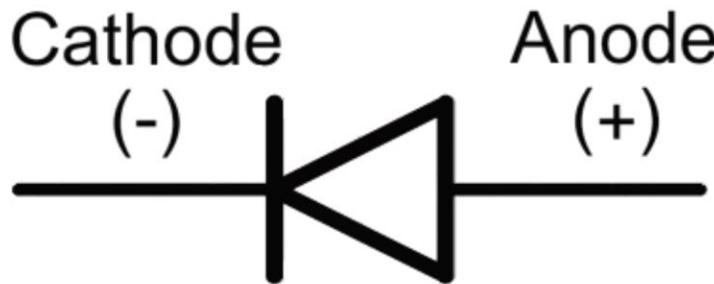
Parts : LED (1), R ($330 \Omega \times 1$)



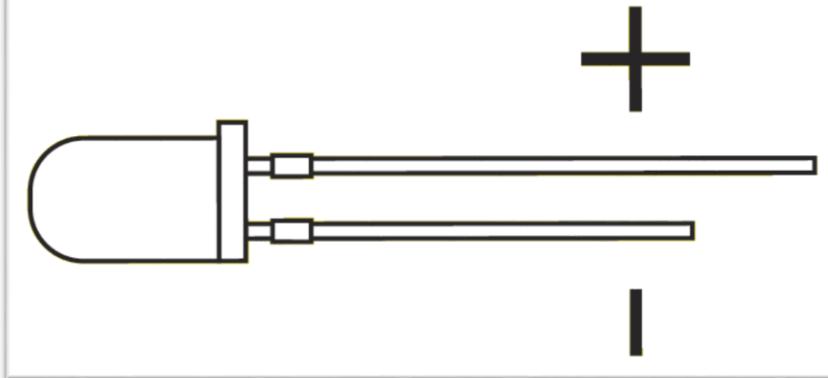


0.6.A2 single LED

Polarity of Diode and LED



The diode circuit symbol, with the anode and cathode marked.

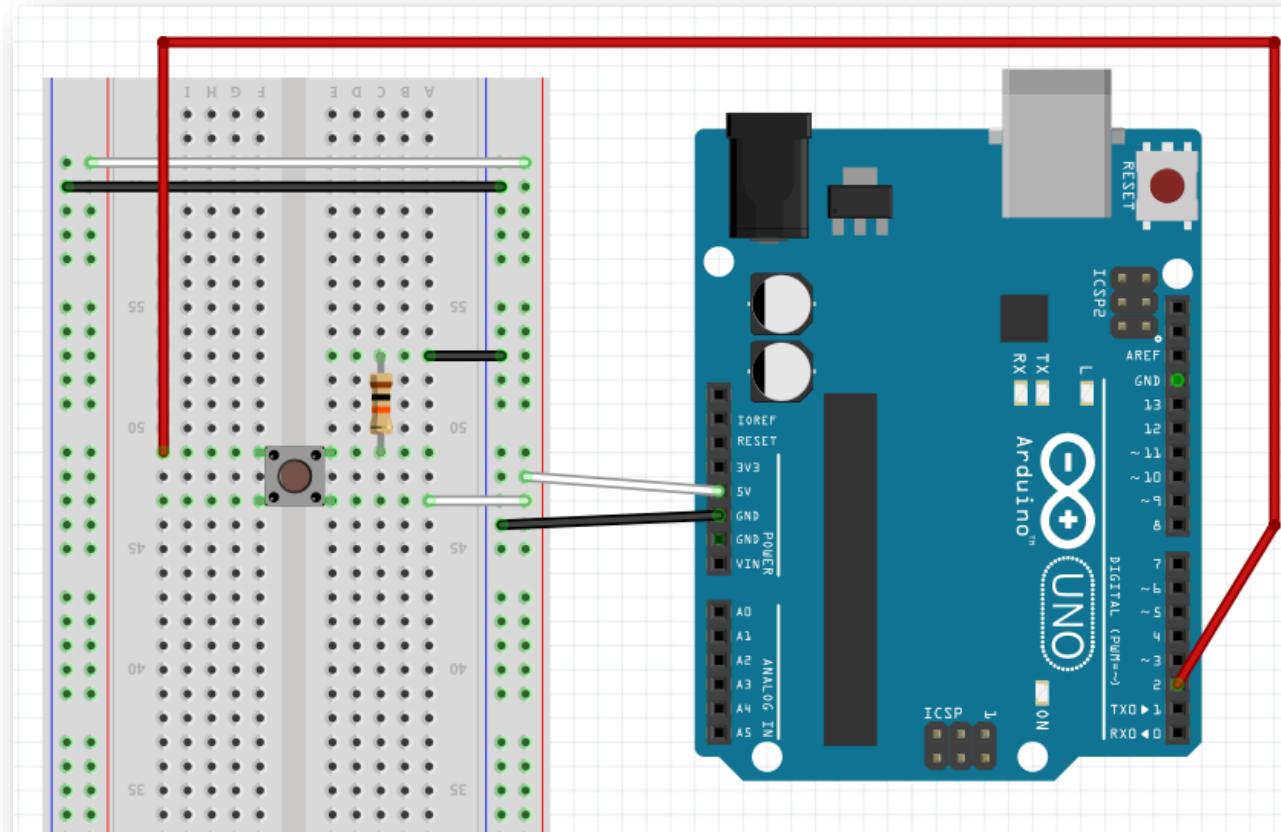


Find the longer leg, which should indicate the positive, anode pin.

<https://learn.sparkfun.com/tutorials/polarity/diode-and-led-polarity>



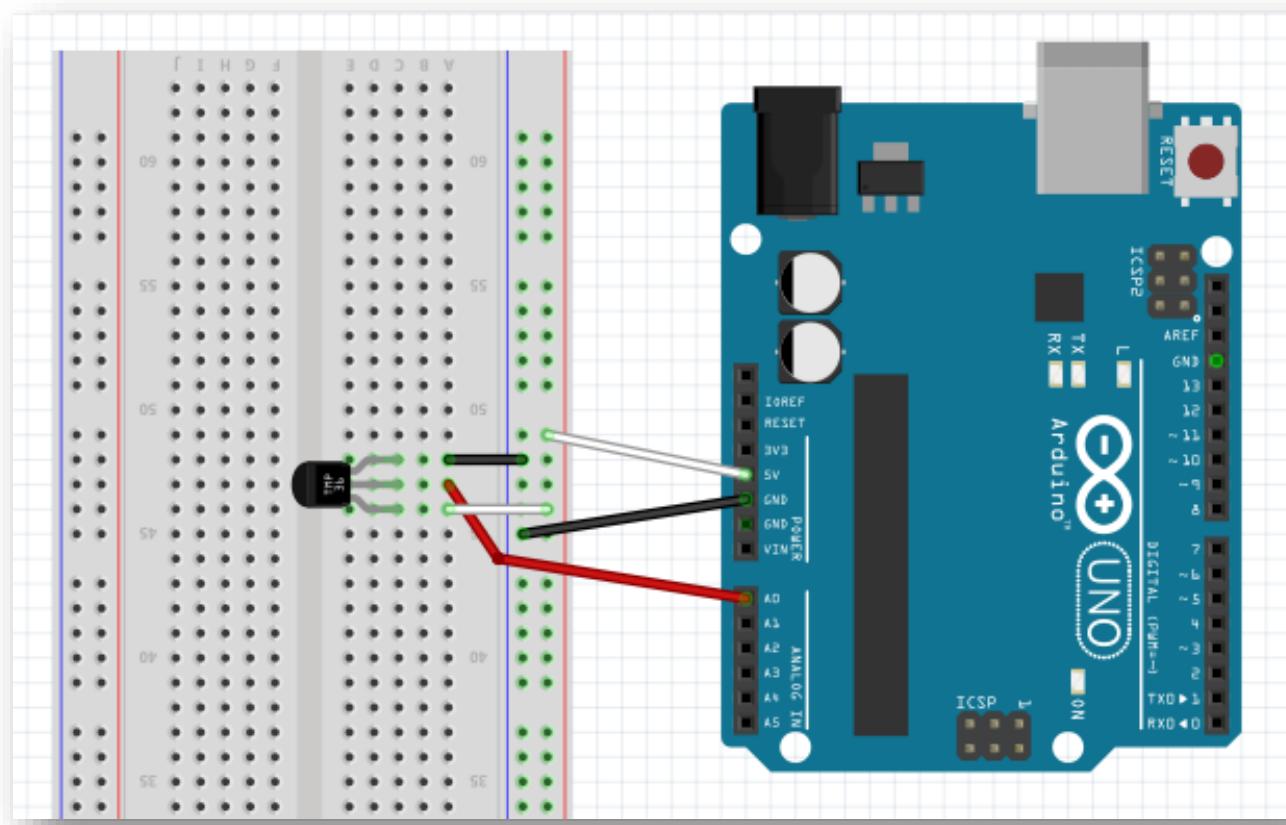
0.6.A3 button Switch



Parts : Button switch (1), R ($10\text{ k}\Omega \times 1$)



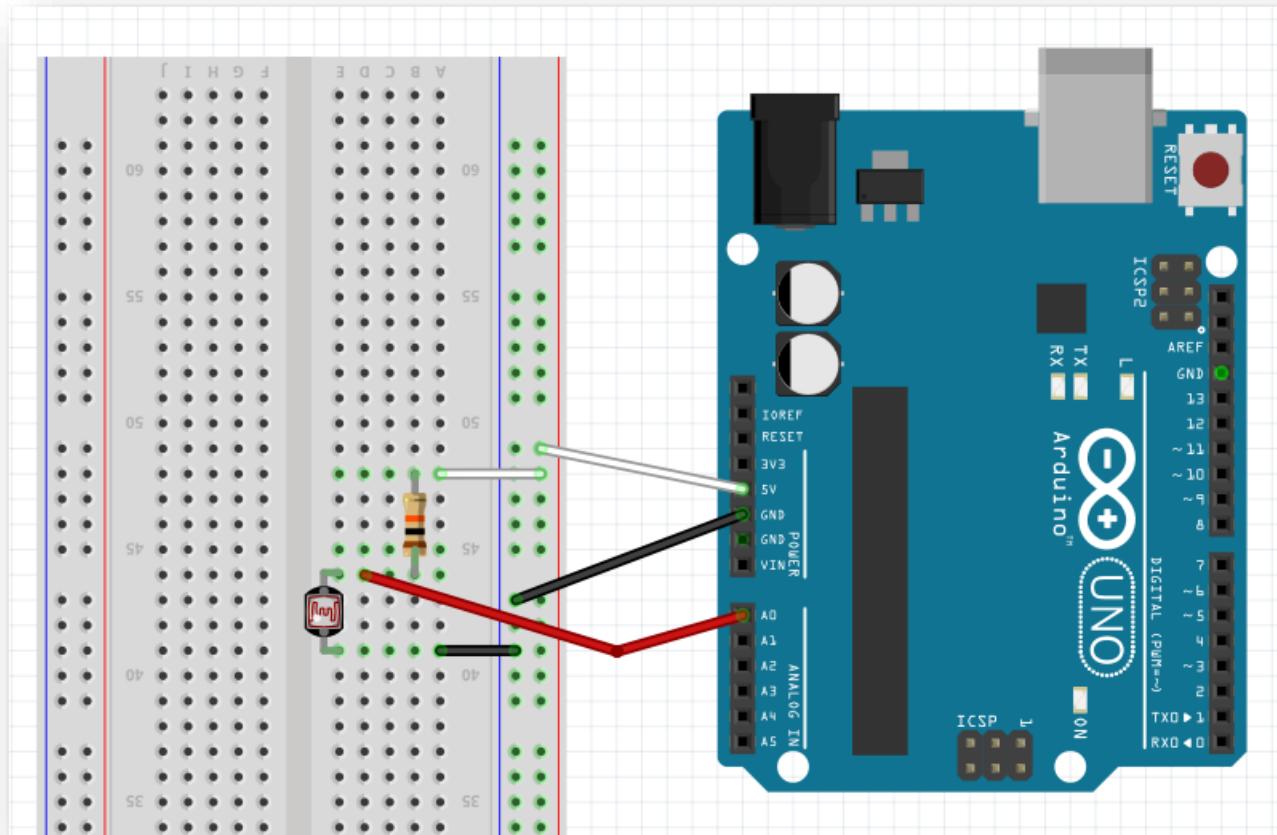
0.6.A4 Temperature sensor (TMP36)



Parts : Temperature sensor (TMP36)
A0 : analog signal input



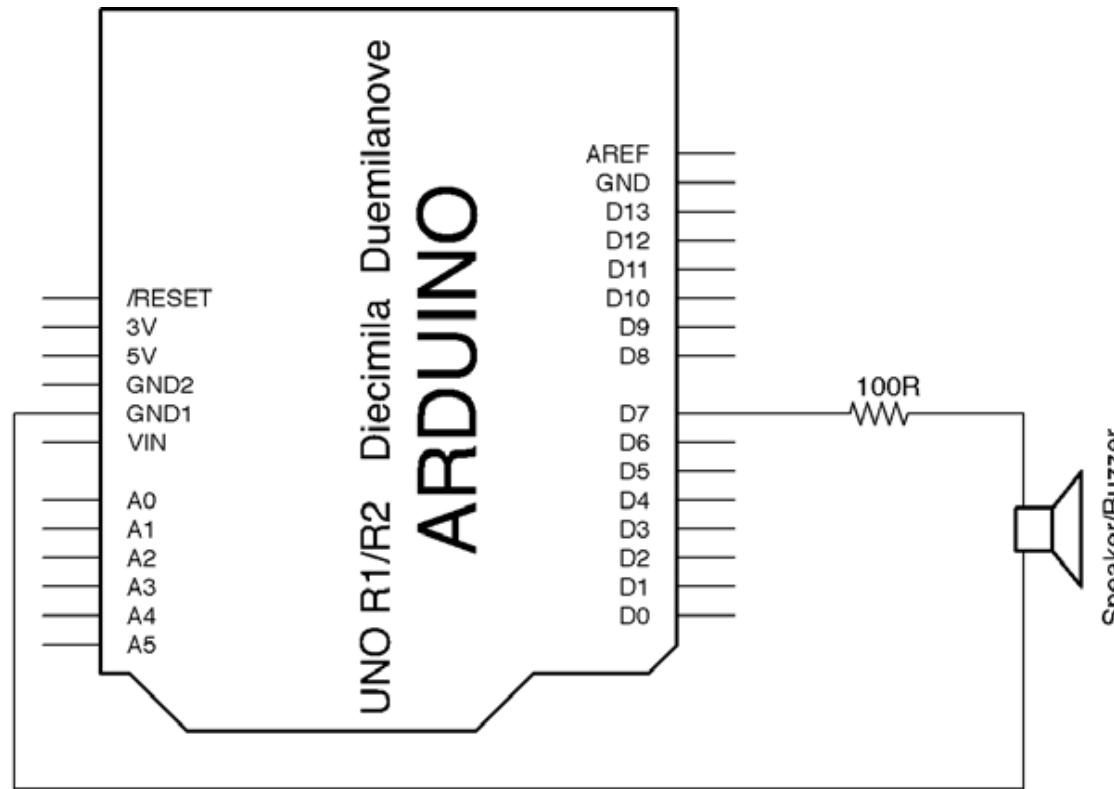
0.6.A5 Luminosity sensor : photo cell LDR



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



0.6.A6 Speaker/Buzzer



Parts : Speaker/Buzzer, R ($100 \Omega \times 1$)

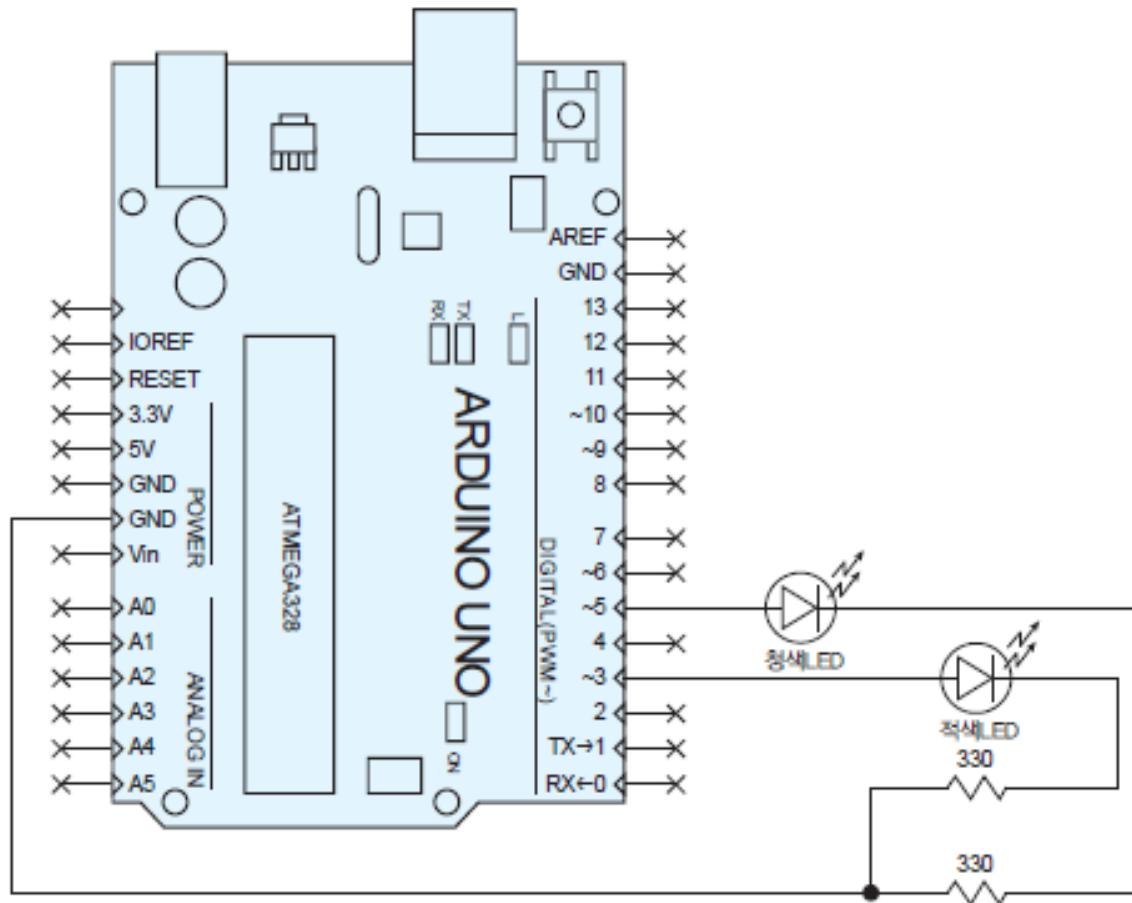


Arduino circuits

DIY



0.6.DIY1 Two LEDs

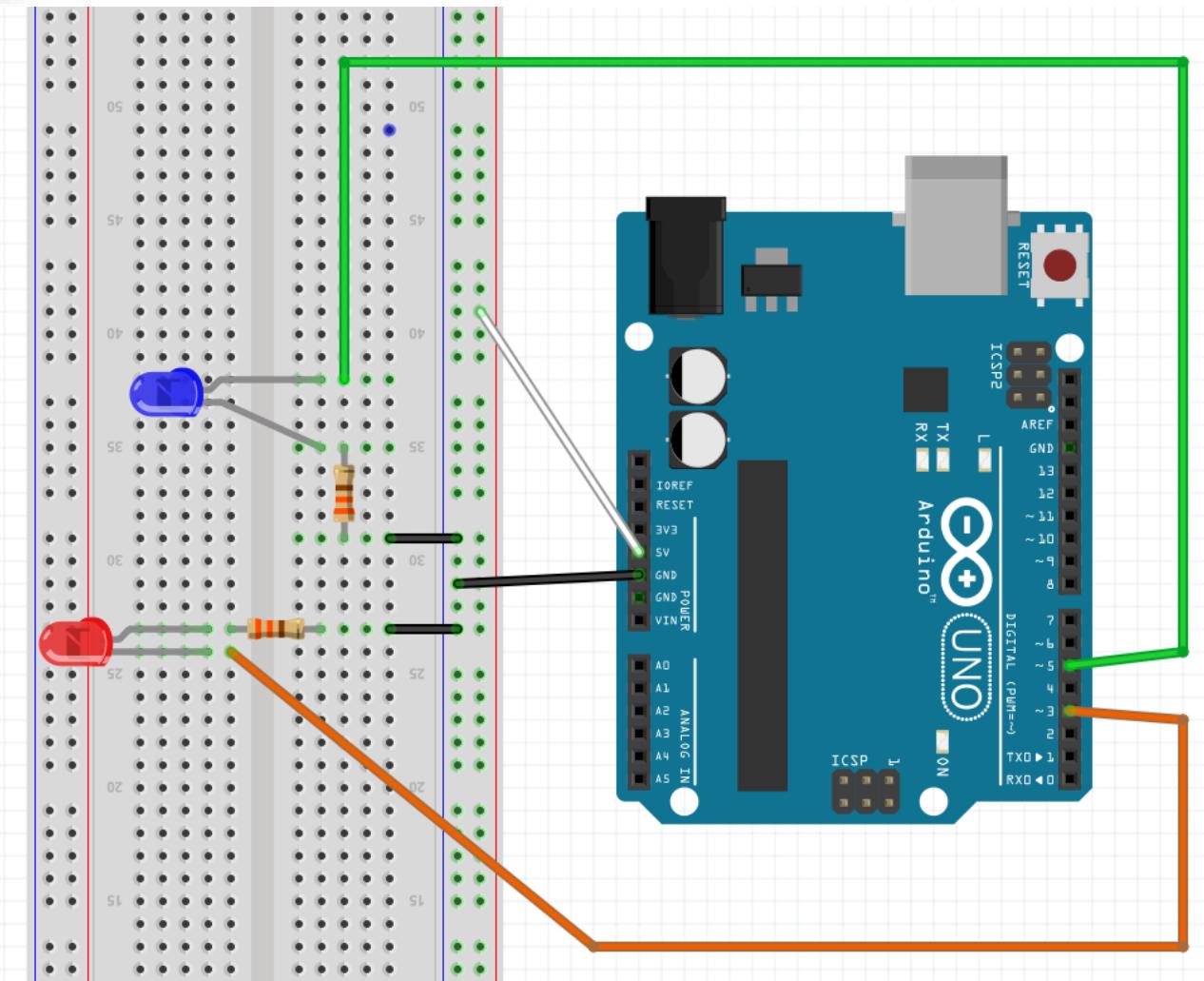


Parts : LED(청, 적), R (330 Ω X 2)

AAnn_2Led.fzz



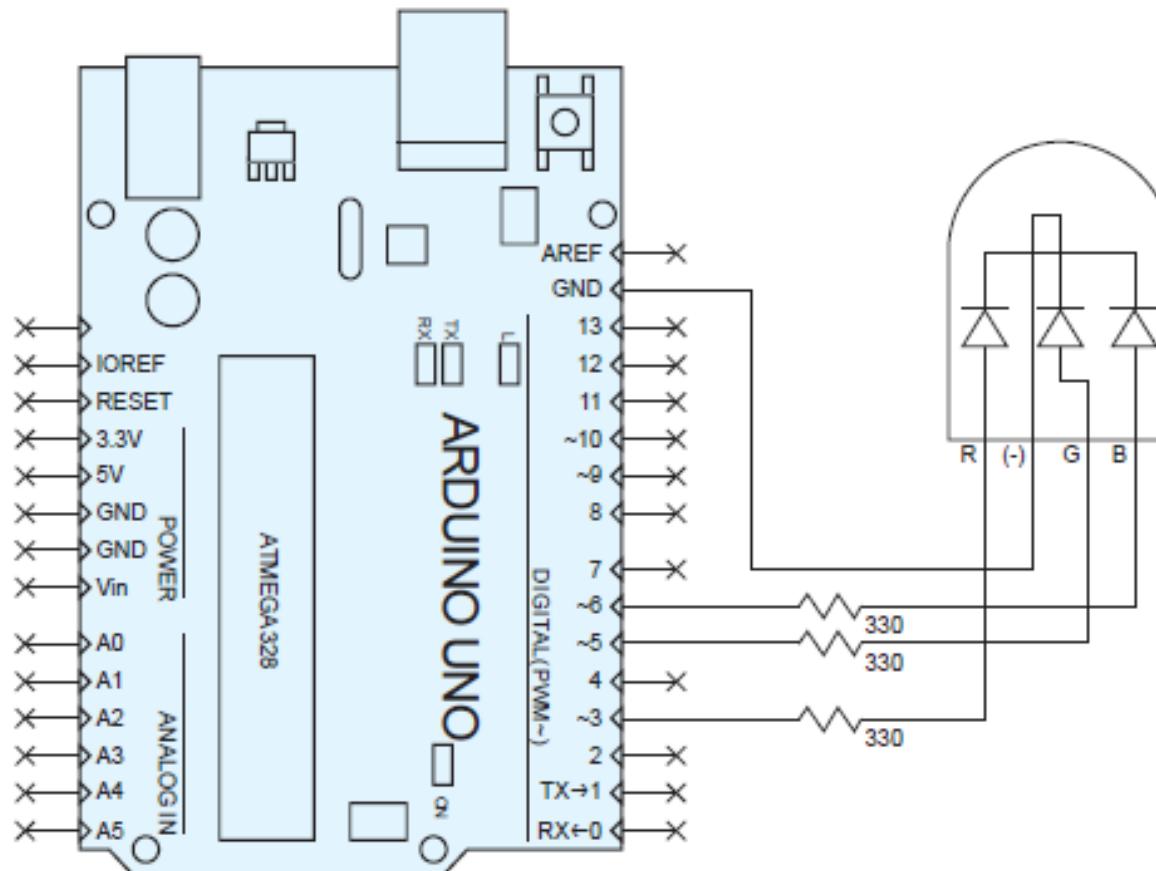
0.6.DIY1 Two LEDs



AAnn_2Led.fzz



0.6.DIY2 RGB LED

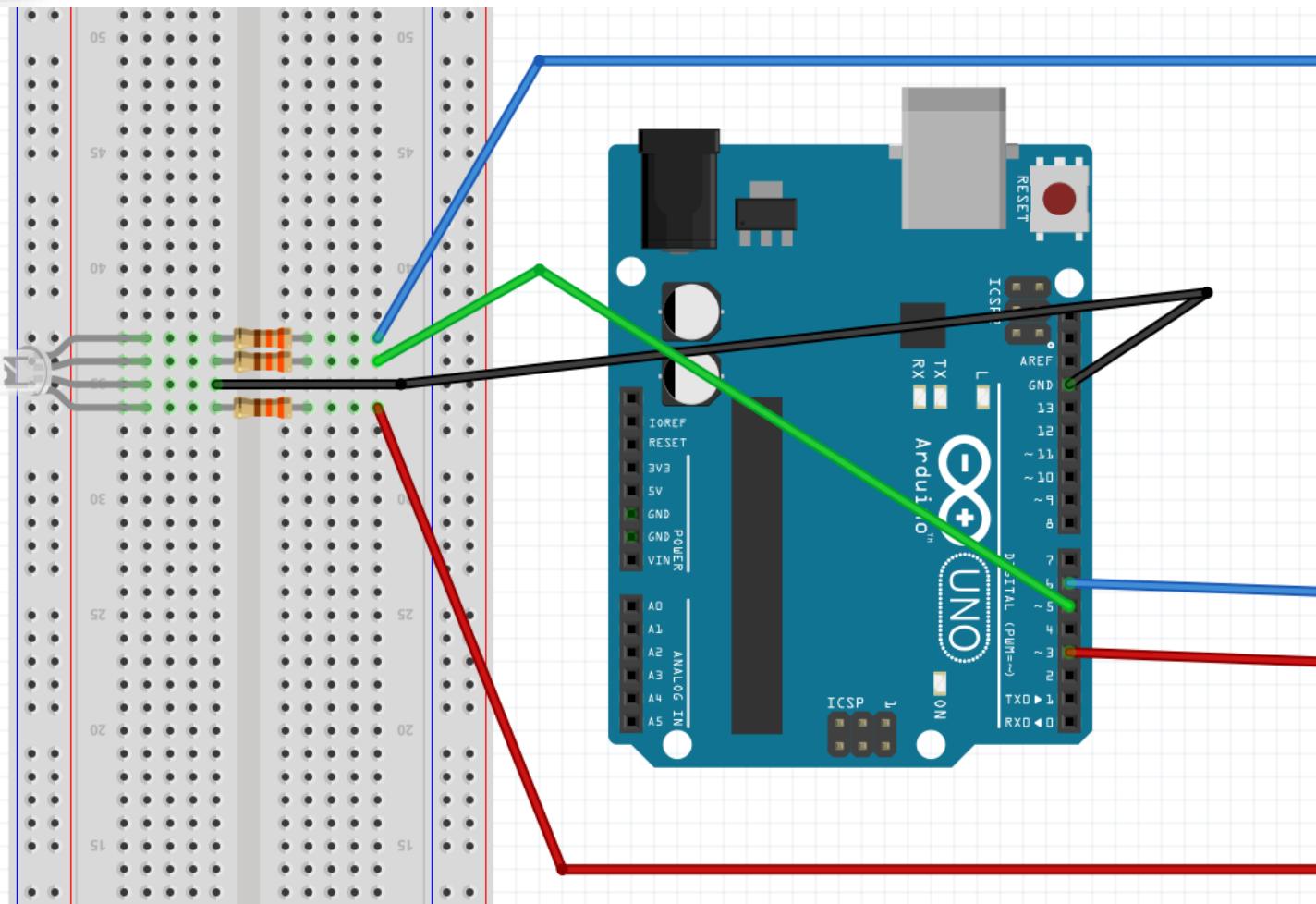


Parts : RGB LED , R (330 Ω X 3)

AAnn_rgbLed.fzz



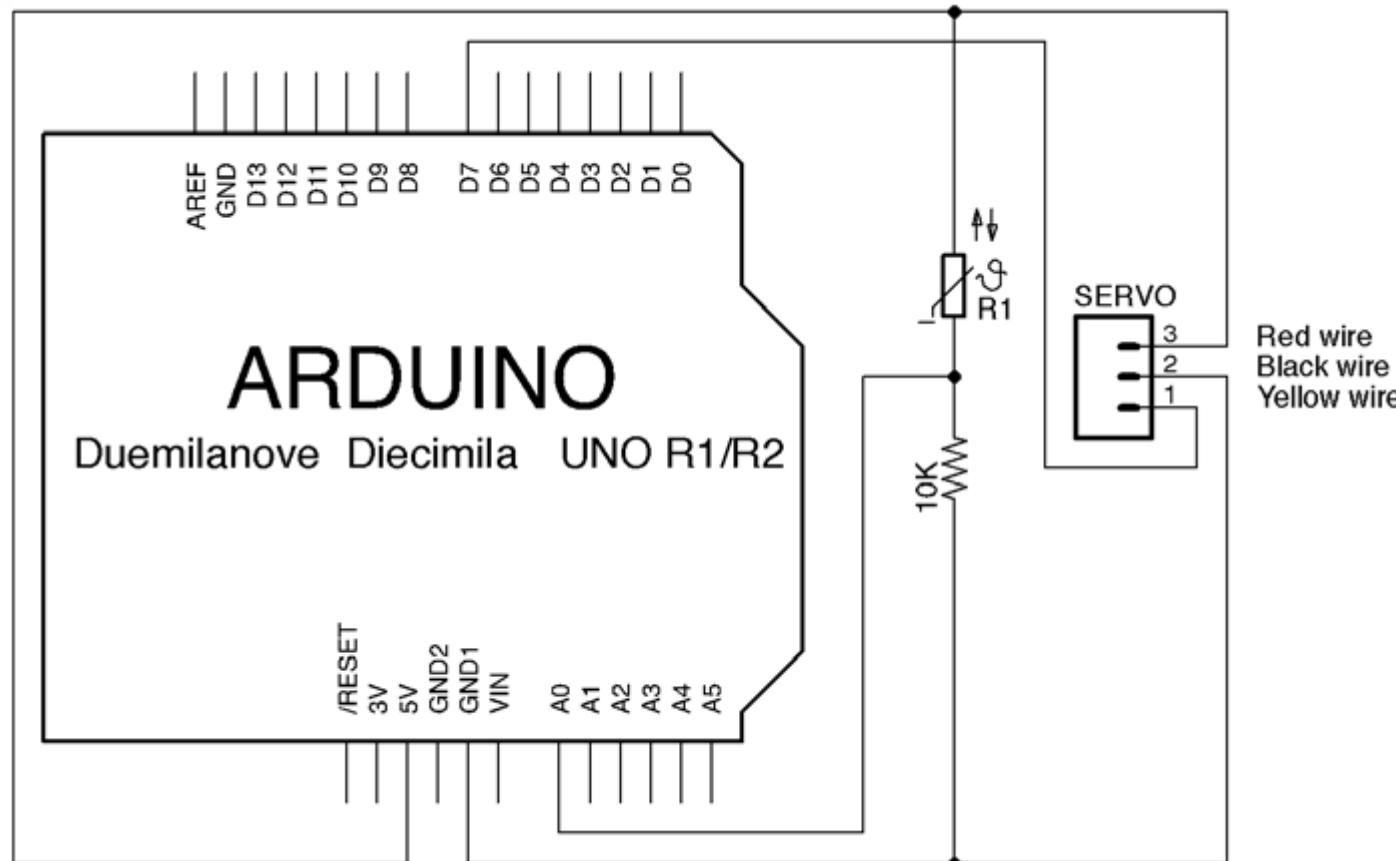
0.6.DIY2 RGB LED



AAnn_rgbLed.fzz



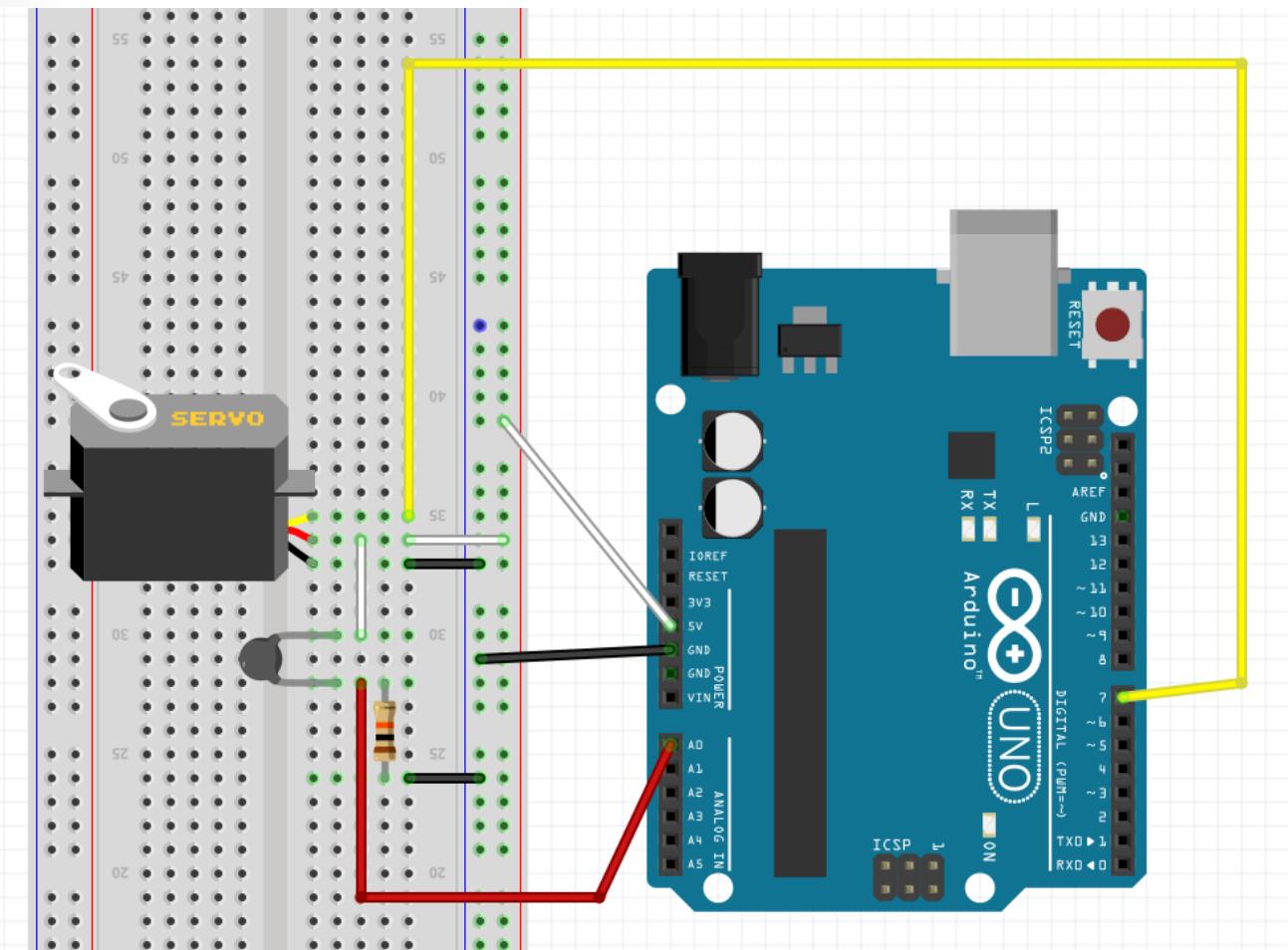
0.6.DIY3 Servo



Parts : Servo motor, thermistor, R (10 kΩ X 1)



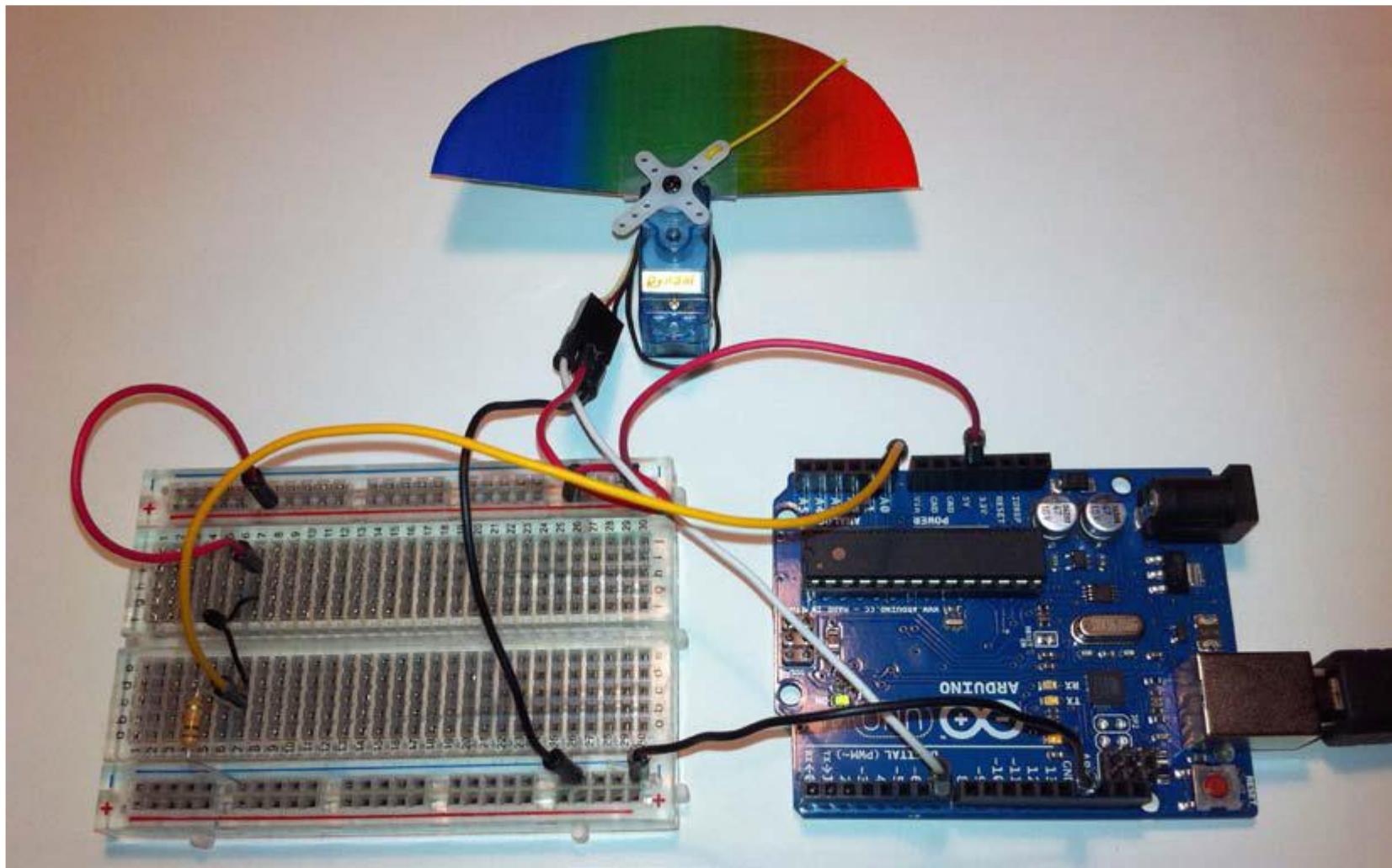
0.6.DIY3 Servo



AAnn_servo.fzz

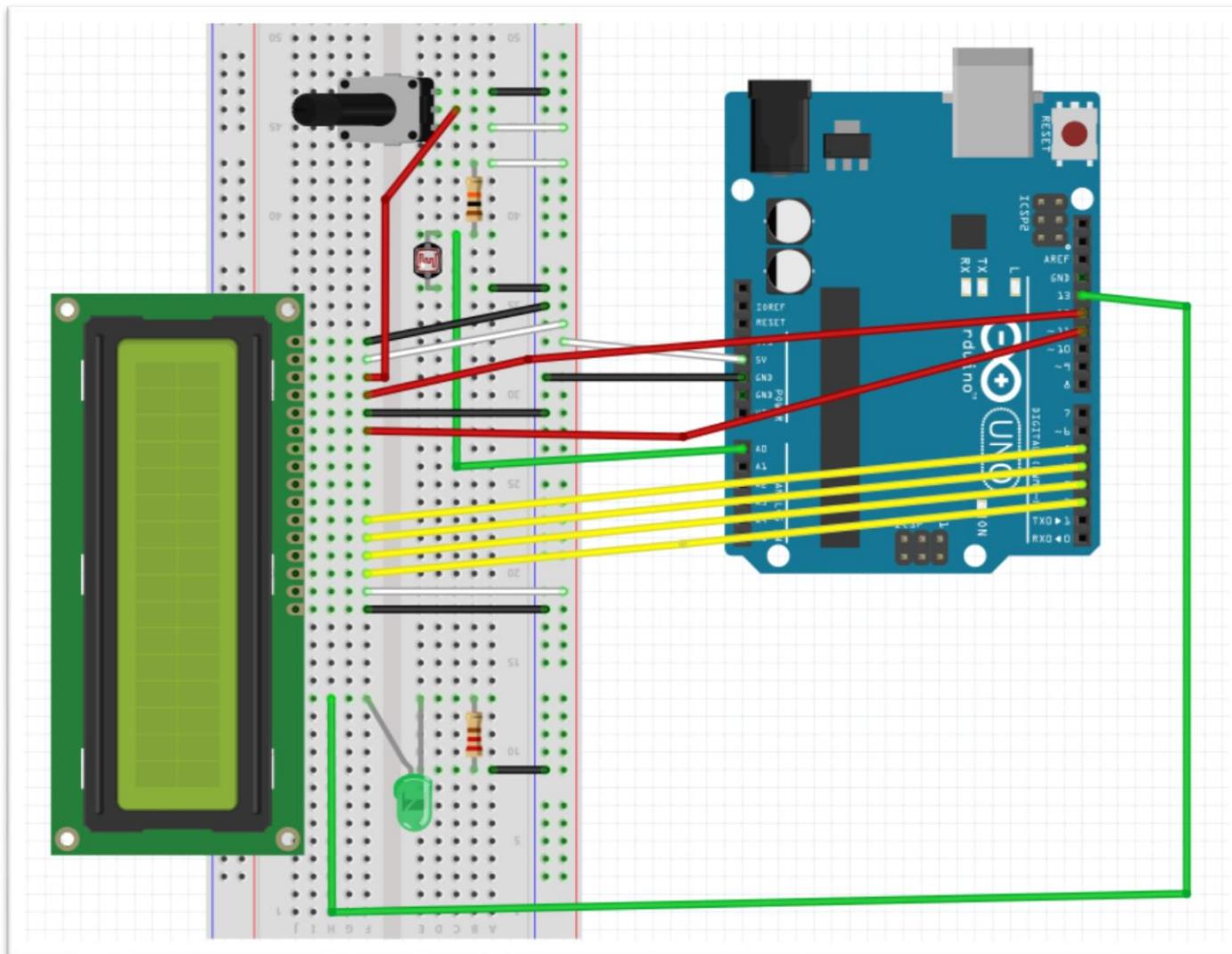


0.6.DIY3 Servo



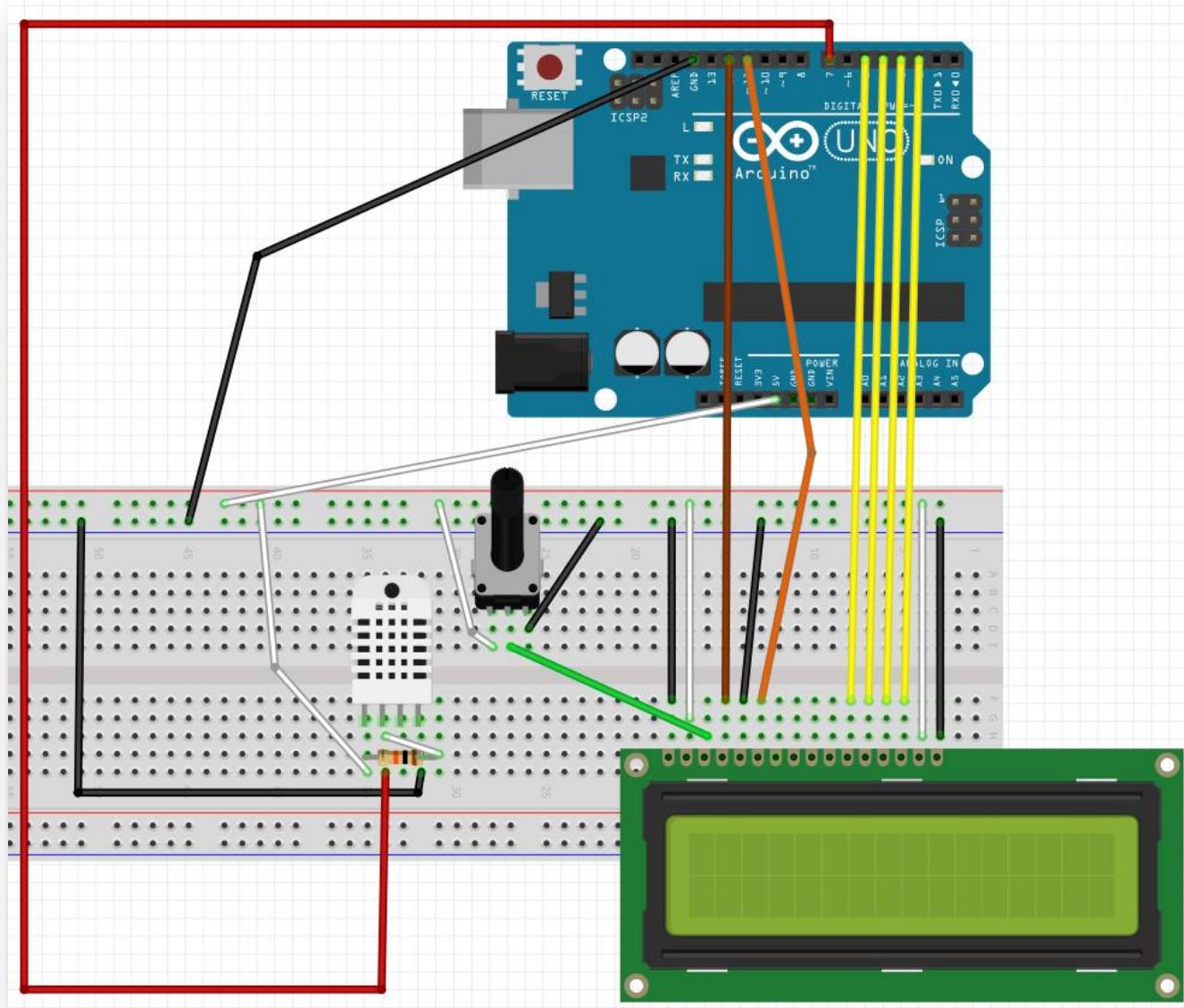


0.7.DIY4 Display of luminosity





0.8.DIY5 Display of Temperature & Humidity





1. Arduino SW: IDE



The screenshot shows the top navigation bar of the Arduino website. On the left is the Arduino logo. To its right are several menu items: HOME, BUY, SOFTWARE (which is highlighted in a darker teal color), PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG. The background of the bar is a teal gradient.

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



1.1 Arduino IDE

Download the Arduino IDE



ARDUINO 1.8.4

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Windows app [Get](#)

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

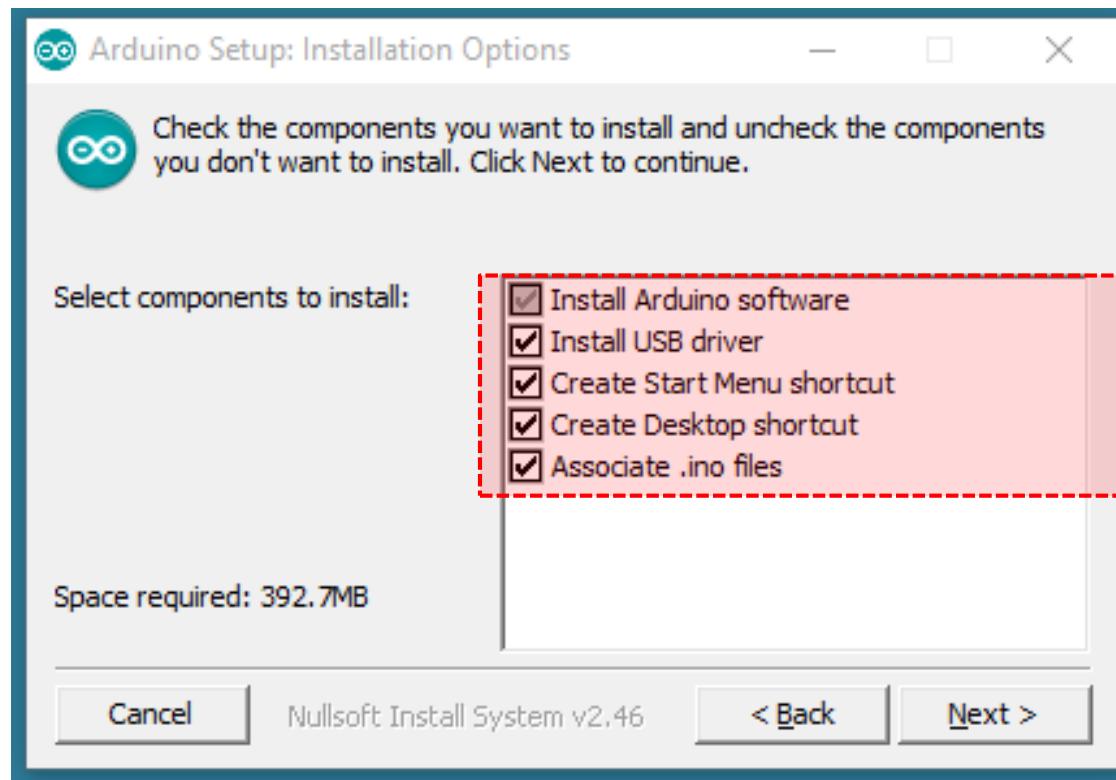
[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

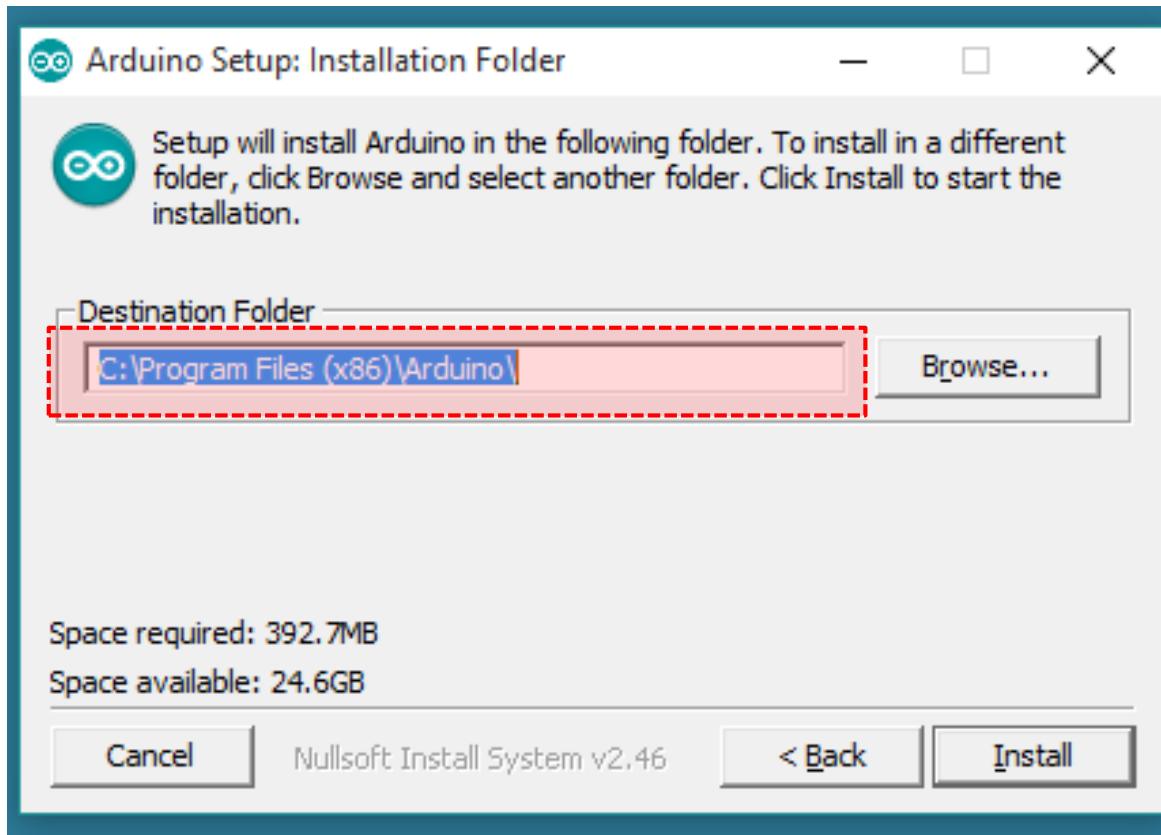


1.2 Arduino IDE





1.3 Arduino IDE





1.4 Arduino IDE

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch_may01a | 아두이노 1.8.2
- Menu Bar:** 파일 편집 스케치 둘 도움말
- Toolbar:** Includes icons for save, open, print, and upload.
- Sketch Editor:** Displays the code for sketch_may01a. The code consists of two main functions:
 - setup():** A function to run once at the start. It contains a comment: // put your setup code here, to run once:.
 - loop():** A function to run repeatedly. It contains a comment: // put your main code here, to run repeatedly:.
- Code Syntax Highlighting:** The `setup()` and `loop()` blocks are highlighted with red dashed boxes.
- Status Bar:** Shows Arduino/Genuine Uno on COM3.



1.5 Arduino IDE

sketch_may01a | 아두이노 1.8.2

파일 편집 스케치 툴 도움말

- 새 파일 Ctrl+N
- 열기... Ctrl+O
- 최근 파일 열기 >
- 스케치북 >
- 예제 >
- 닫기 Ctrl+W
- 저장 Ctrl+S
- 다른 이름으로 저장... Ctrl+Shift+S
- 페이지 설정 Ctrl+Shift+P
- 인쇄 Ctrl+P
- 환경설정 Ctrl+Comma
- 종료 Ctrl+Q

환경설정

설정 네트워크

스케치북 위치: C:\Users\yish-HC\Documents\Arduino

찾아보기

에디터 언어: 시스템 기본설정 (마우스를 재시작해야 함)

에디터 글꼴 크기: 14

Interface scale: 자동 100 % (마우스를 재시작해야 함)

다음 동작중 자세한 출력 보기: 캠패일 업로드

캠패일러 경고: None

줄 번호 표시

코드 풀딩 사용하기

업로드 후 코드 확인하기

외부 에디터 사용

Aggressively cache compiled core

시작시 업데이트 확인

스케치 파일을 저장할 때 새로운 확장자(.pde -> .ino)로 업데이트

검증 또는 업로드 할 때 저장하기

추가적인 보드 매니저 URLs

추가적인 환경 설정은 파일에서 직접 편집할 수 있습니다
C:\Users\yish-HC\AppData\Local\Arduino15\preferences.txt
(마우스가 실행되지 않는 경우에만 수정 가능)

확인 취소



[참고 : 저항 값 읽기]

Carbonfilm resistor

4 Color stripes

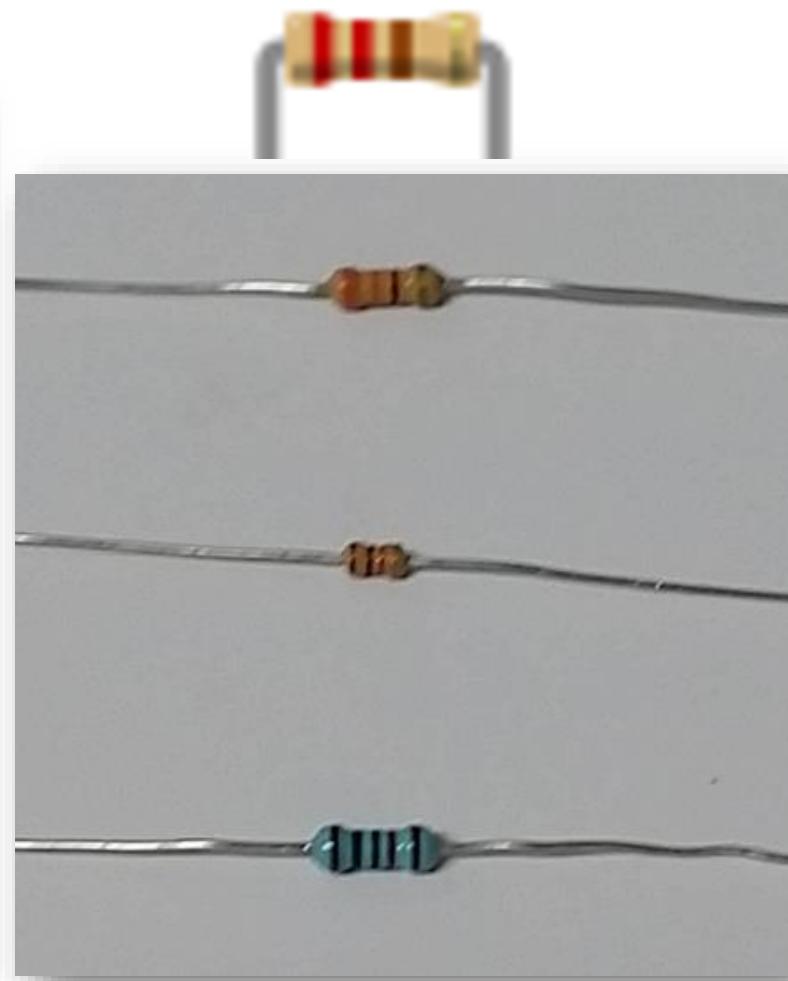
$$47 \times 1000 = 47\text{KOhm } 5\%$$

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%

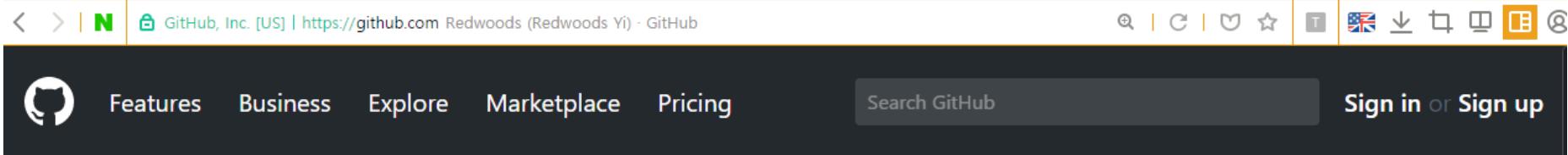


Lecture materials

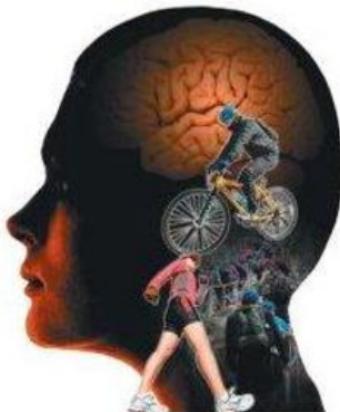


● References & good sites

- ✓ <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. 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

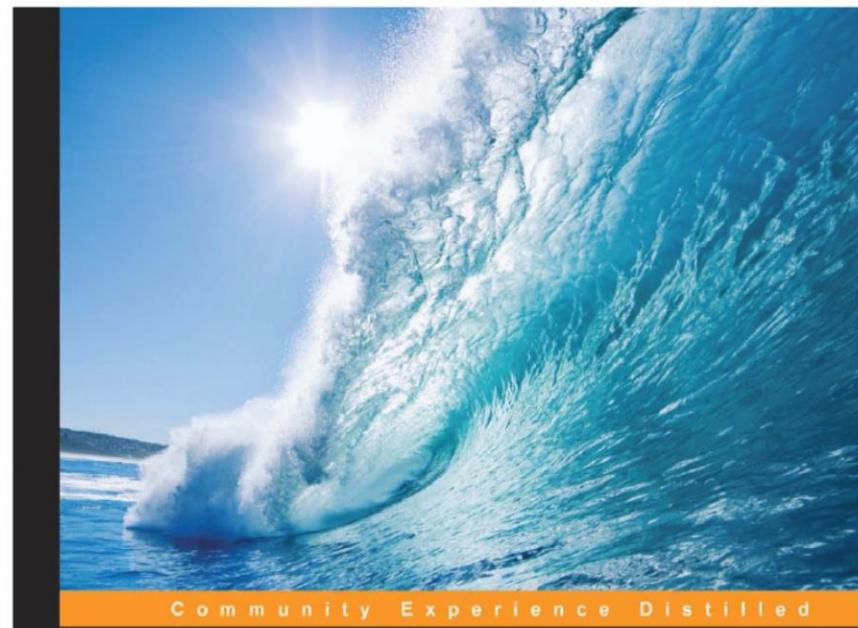
[hw-coding](#)

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

 Arduino



References

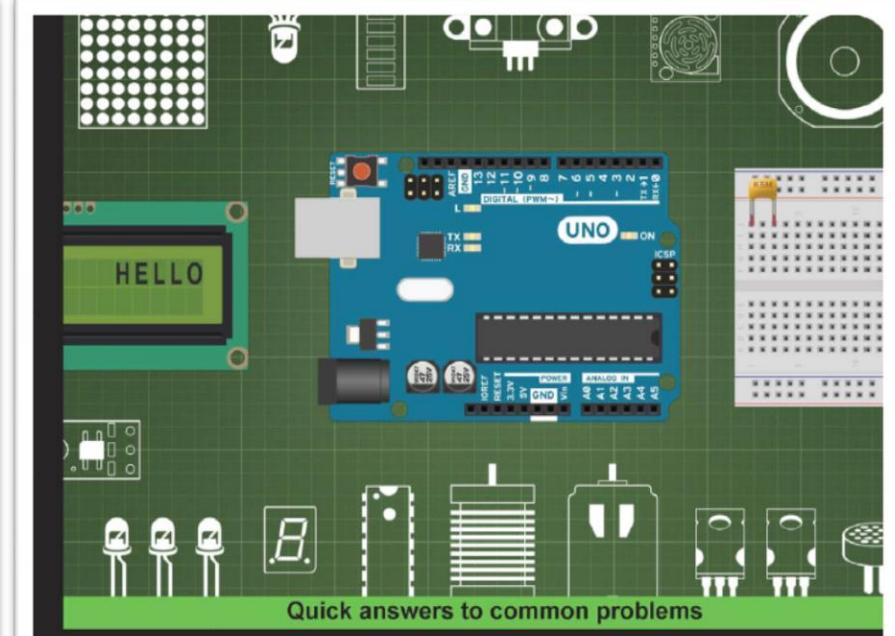


Arduino Essentials

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