

Physical Computing

박종화(suakii@gmail.com) | 2017.09.26

1

프로그램 소개

아두이노 무드 램프



프로그램 소개

무드 램프

시각적인 효과

RGB LED 이용

+

Software(Processing + Library)

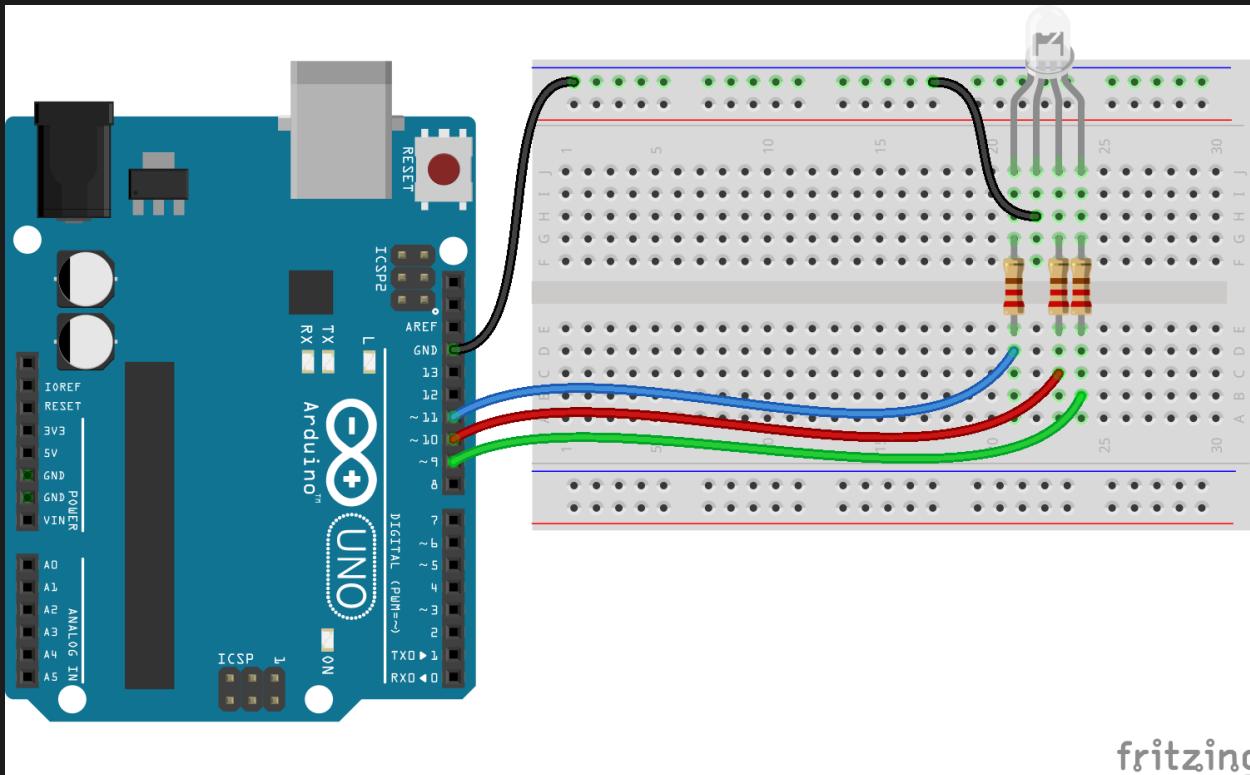
프로그램 소개

무드 램프 - Hardware



프로그램 소개

무드 램프 - Hardware



프로그램

기존 코드

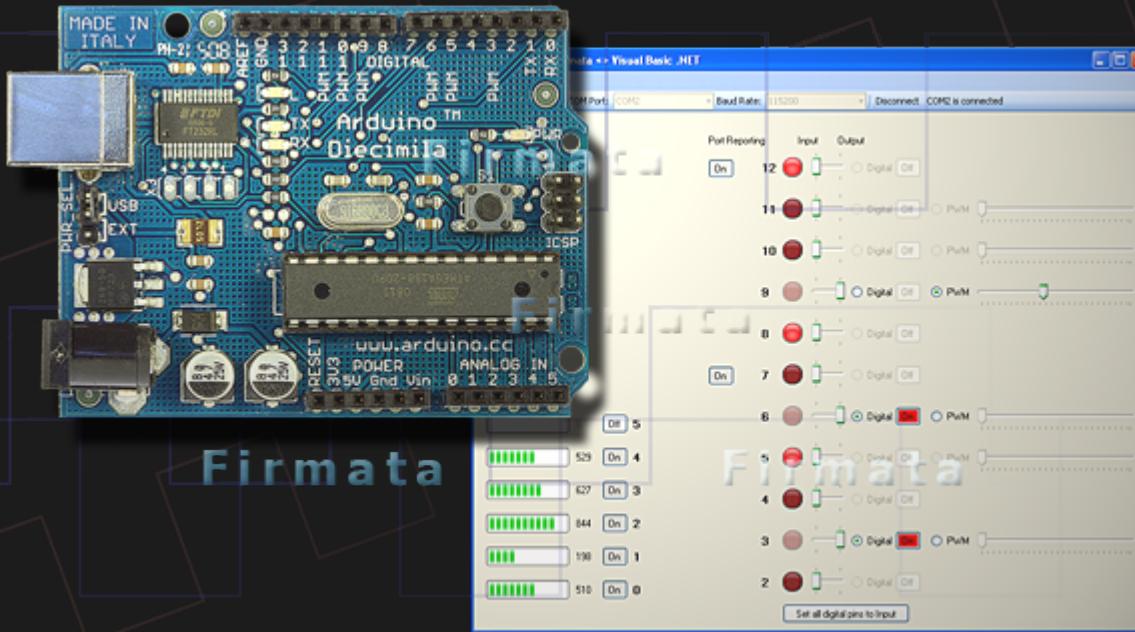
```
int REDPin = 9;  
int GREENPin = 10;  
int BLUEPin = 11;  
int brightness = 0;  
int increment = 5;  
void setup()  
{  
    pinMode(REDPin, OUTPUT);  
    pinMode(GREENPin, OUTPUT);  
    pinMode(BLUEPin, OUTPUT);  
    Serial.begin(9600);  
}
```

```
void loop() {  
    brightness = brightness + increment;  
    if (brightness <= 0 || brightness >= 255)  
        increment = -increment;  
    brightness = constrain(brightness, 0, 255);  
    analogWrite(REDPin, brightness);  
    analogWrite(GREENPin, brightness);  
    analogWrite(BLUEPin, brightness);  
    delay(20);  
}
```

Firmata

Firmata

What?



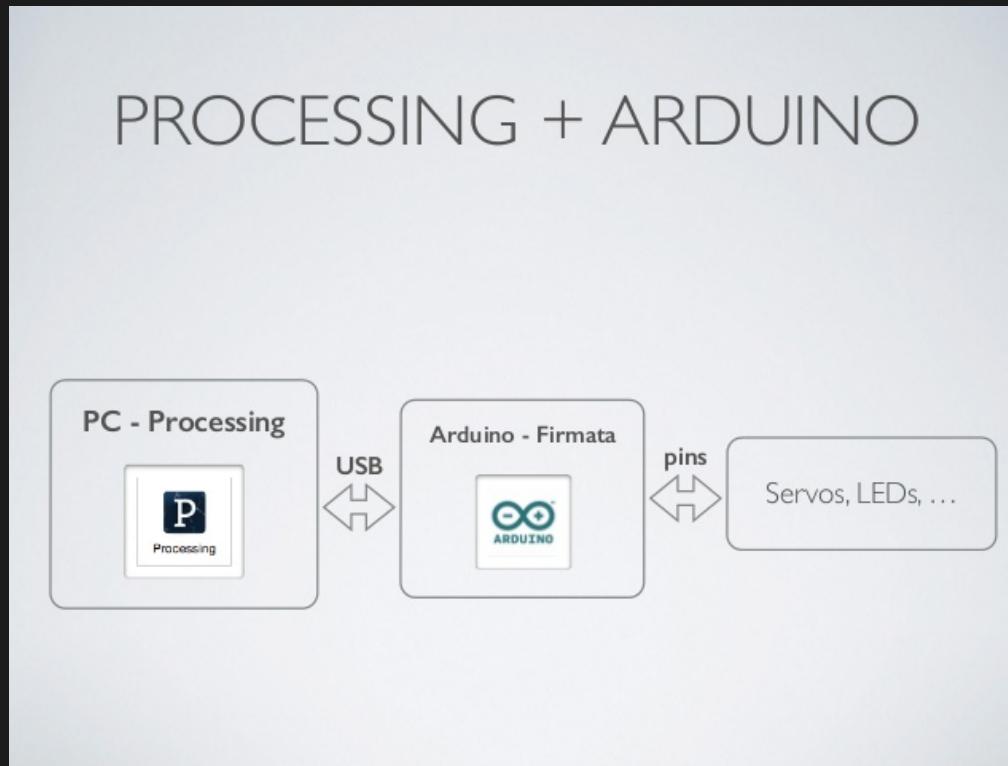
Firmata

Firmata

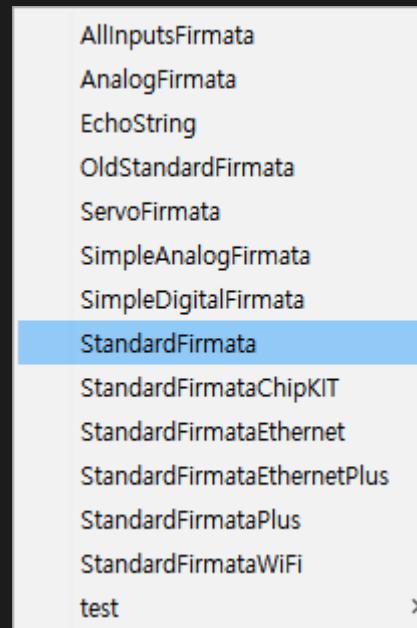
Firmata is a protocol for communicating with microcontrollers from software on a host computer.

Firmata

What?

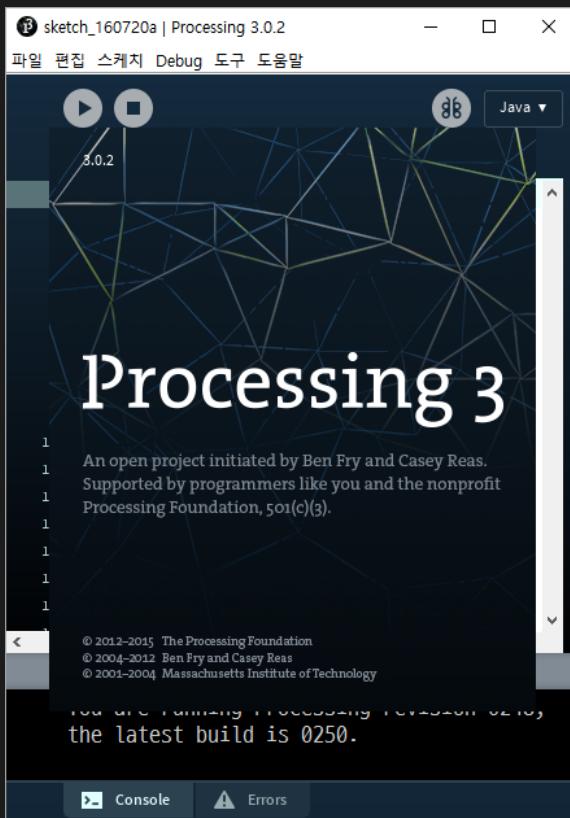


Upload to Arduino



Processing

Processing language



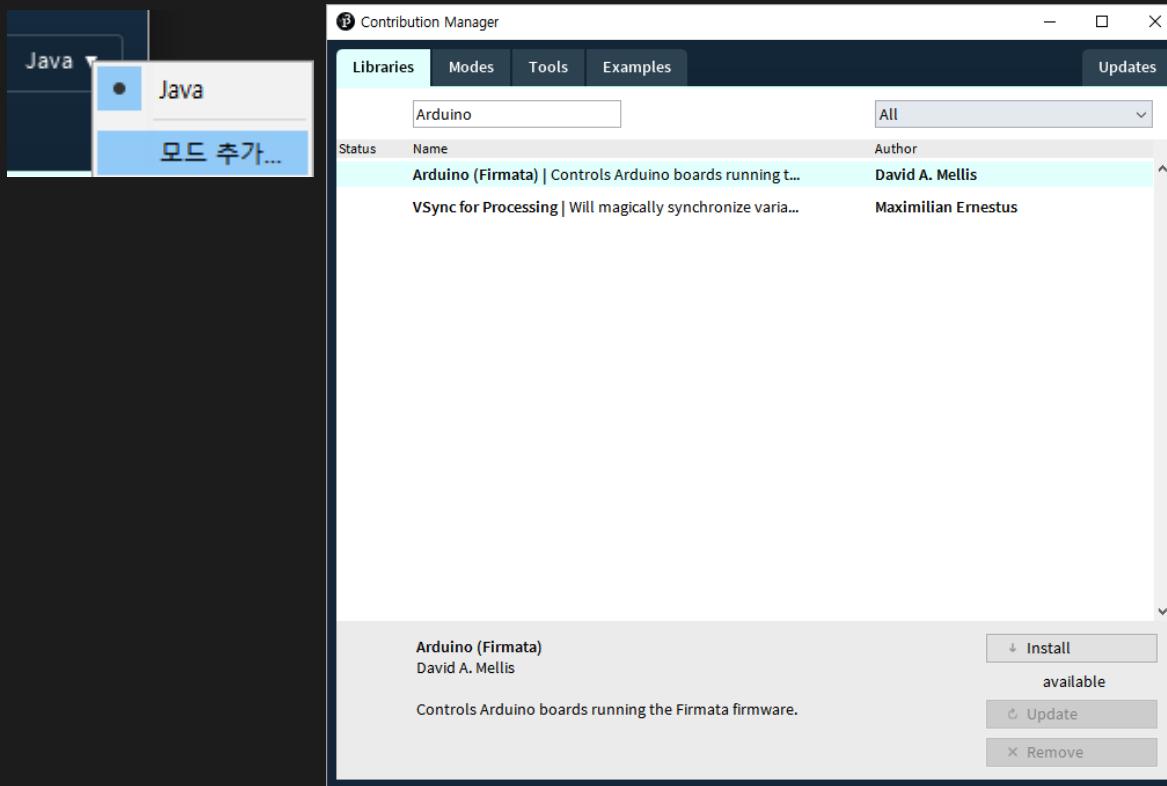
Processing

www.processing.org

Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts

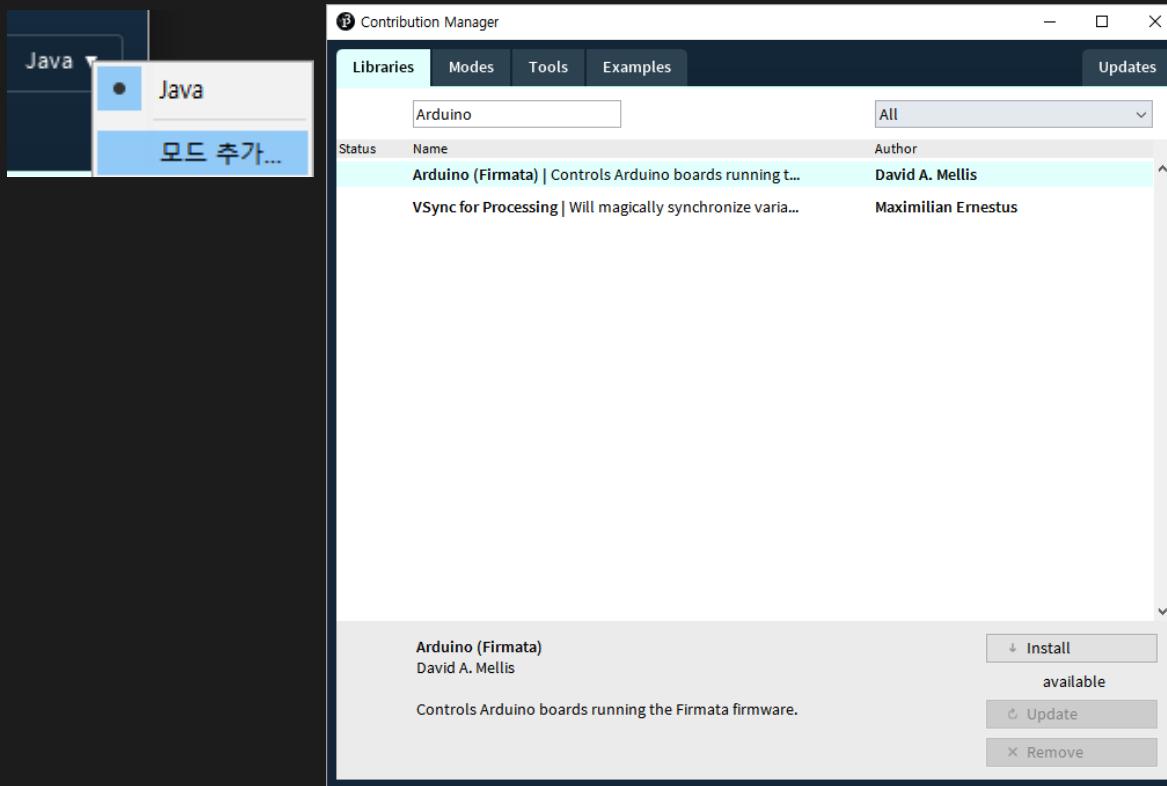
Processing

Install Library



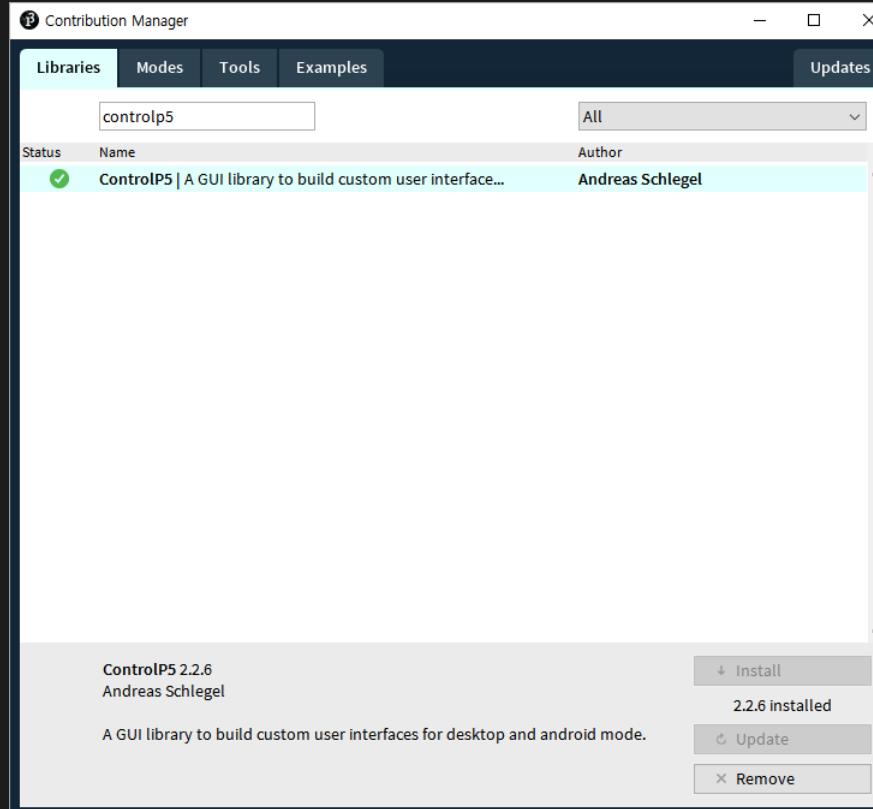
Processing

Firmata Library Install



Processing

ControlP5 - GUI



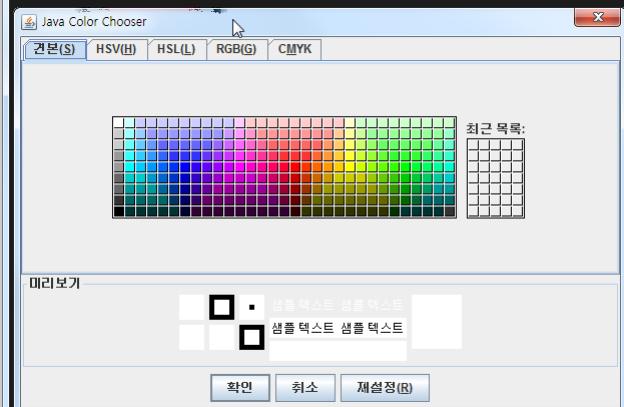
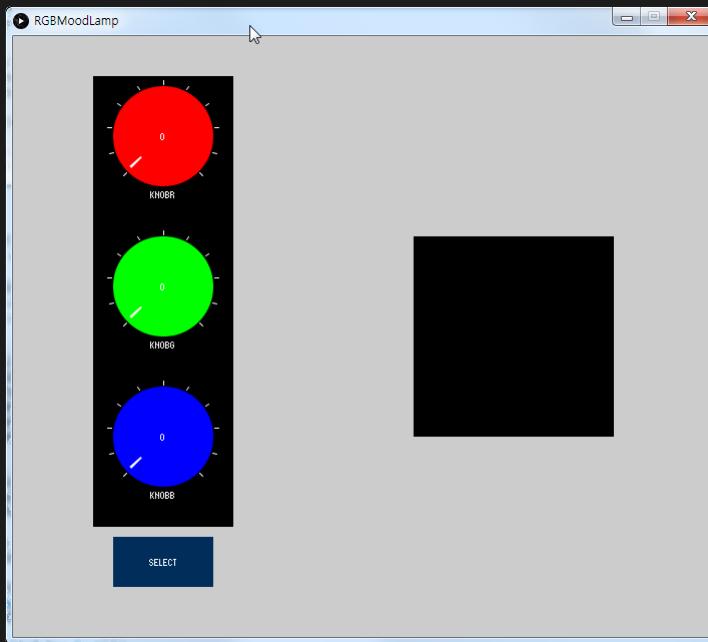
Coding

Processing

<https://github.com/suakii/2017GSArduinoAd>

Coding

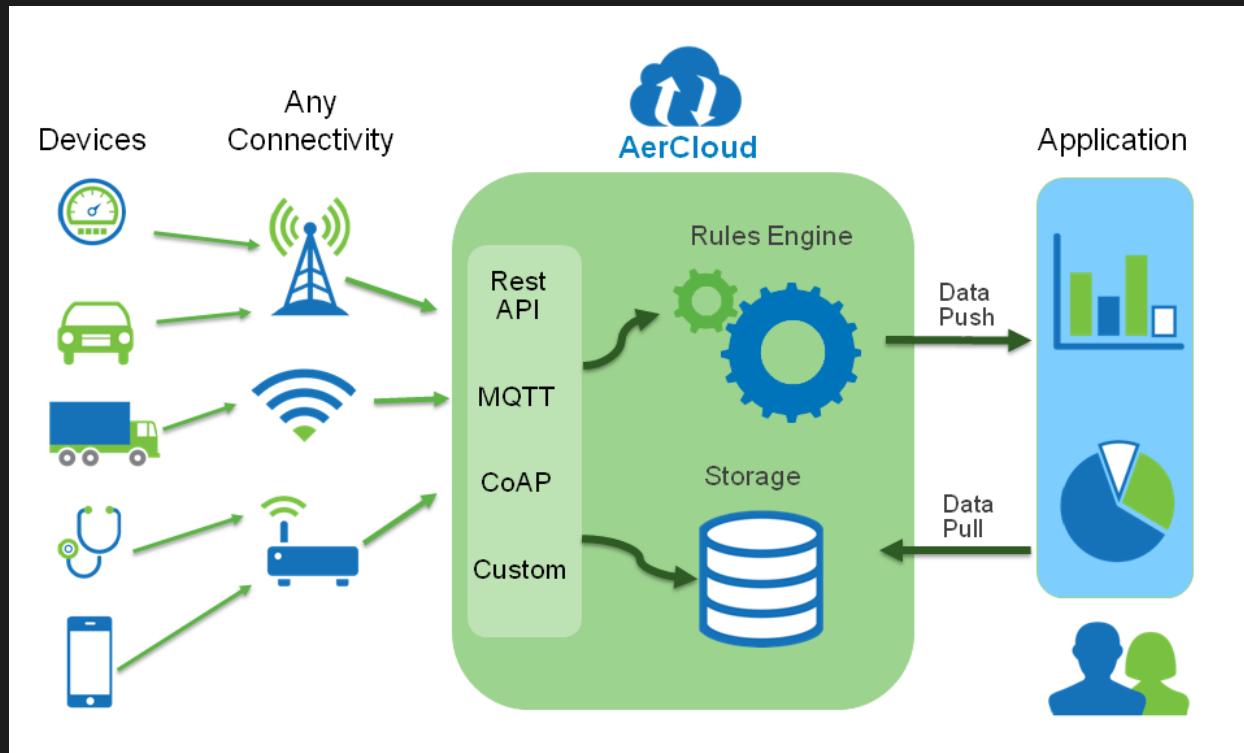
Result



2

프로그램 소개

Simple IOT



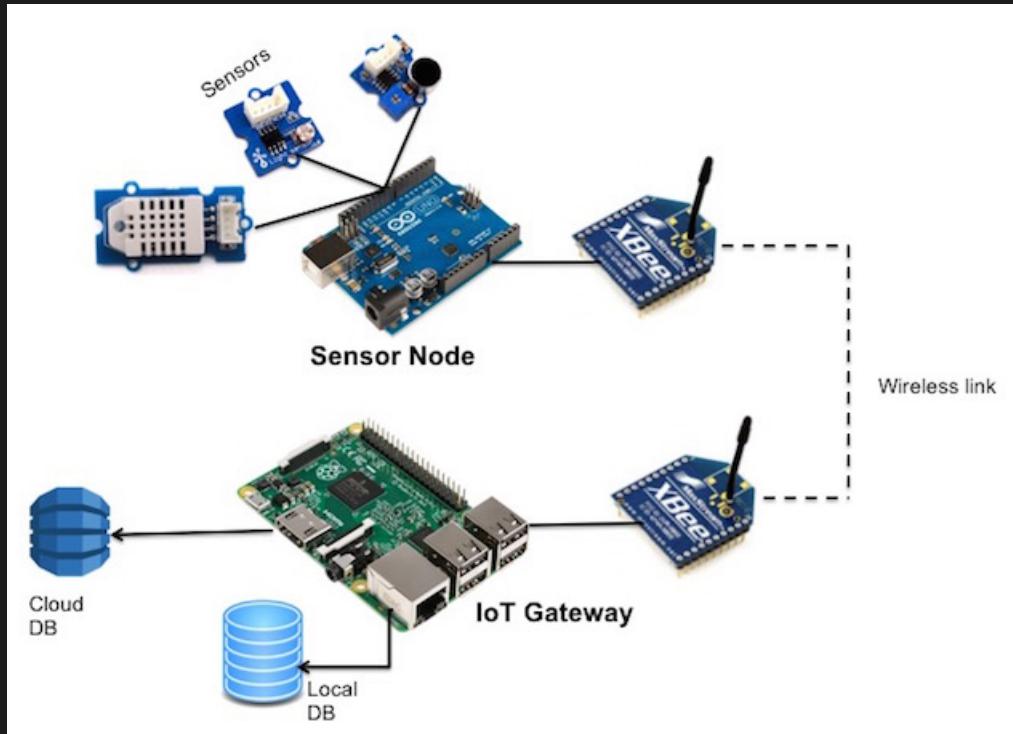
IOT

IOT

사물 인터넷(Internet of Things, 약어로 IoT)은 각종 사물에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술을 의미한다.

IOT

IOT



IOT

IOT

But..

Processing + Arduino

IOT

IOT Cloud Platform

www.thingspeak.com

The screenshot shows the homepage of ThingSpeak. At the top, there's a blue header bar with the logo "ThingSpeak™", navigation links for "Channels", "Apps", "Blog", and "Support", and account links for "Account" and "Sign Out". The main background image is a dark night sky with stars and silhouettes of trees at the bottom. Centered on the page is the text "Billions and Billions" in large white letters, followed by "The open data platform for the Internet of Things" in smaller white letters. Below this text are two buttons: a green "Get Started" button and a white "Contact Us" button. At the bottom of the page, there's a white footer section with three main sections: "Collect" (with a cloud icon), "Analyze" (with a bar chart icon), and "Act" (with a gear icon). Each section has a brief description: "Send sensor data to the cloud.", "Analyze and visualize your data.", and "Trigger a reaction." respectively. Below these sections, there are two columns of features: "ThingSpeak Features" and "Works With".

ThingSpeak Features

- Real-time data collection and storage
- MATLAB® analytics and visualizations
- Alerts
- Device communication
- Open API
- Geolocation data

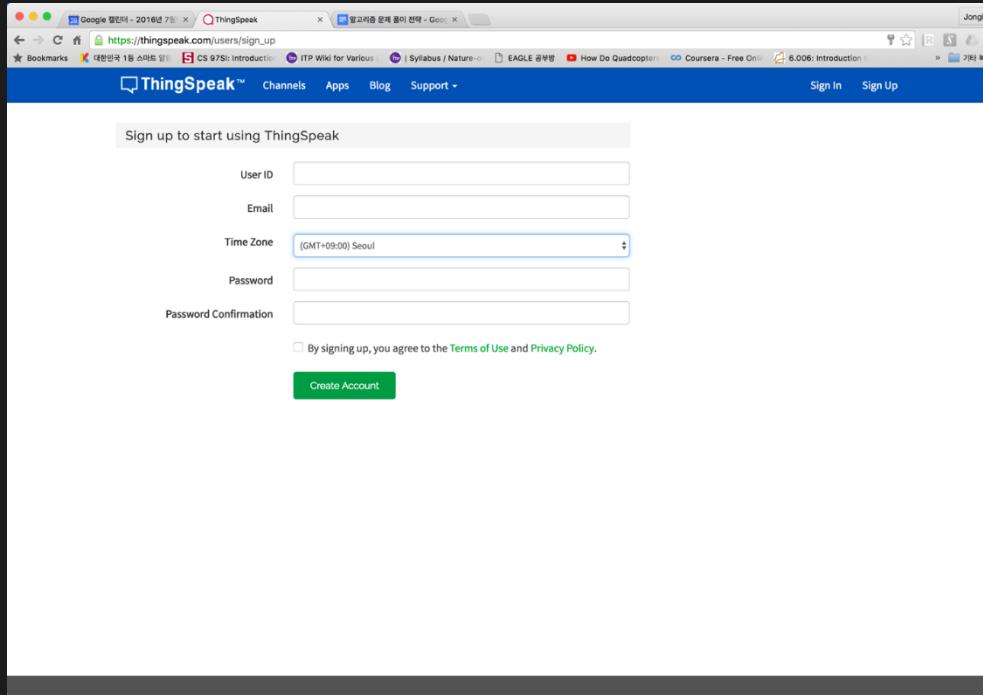
Works With

- Arduino®
- Particle Photon and Core
- Raspberry Pi™
- Mobile and web apps
- Twitter®
- Twilio®

IOT

Thingspeak

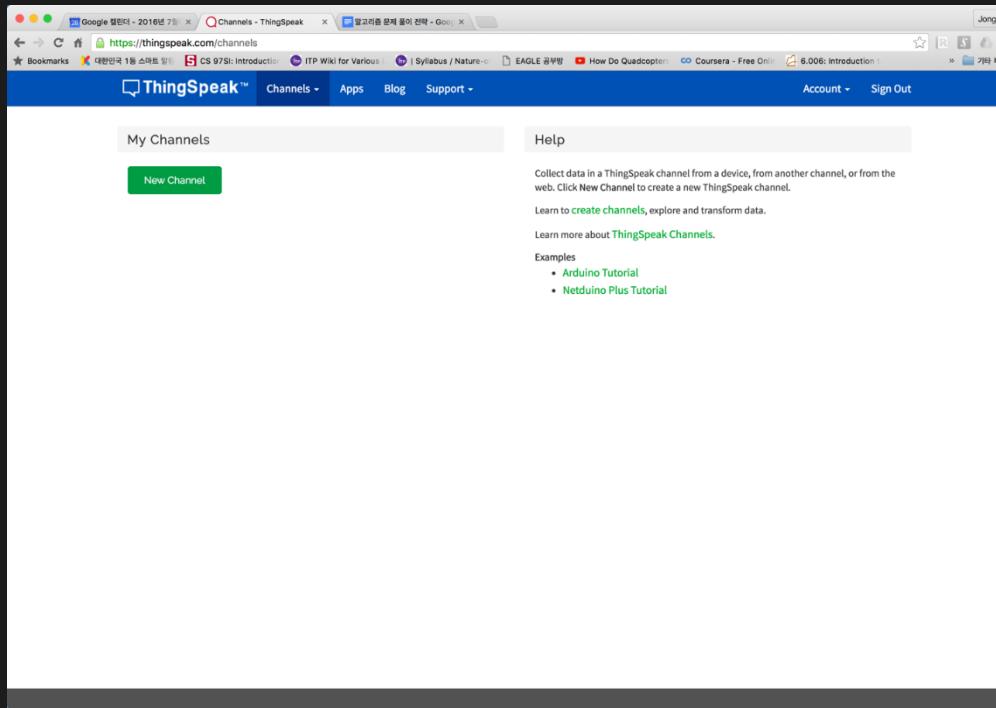
회원가입 및 채널 개설



IOT

Thingspeak

회원가입 및 채널 개설



IOT

Thingspeak

회원가입 및 채널 개설

The screenshot shows the 'New Channel' form on the ThingSpeak website. The 'Name' field contains 'Test' and the 'Description' field also contains 'Test'. There are eight 'Field' input boxes labeled 'Field 1' through 'Field 8', each with a 'Field Label' dropdown menu. Below these is a 'Metadata' input box and a 'Tags' input box with the placeholder '(Tags are comma separated)'. At the bottom, there is a 'Make Public' checkbox and a 'URL' input box. To the right of the form is a 'Help' sidebar with sections for 'ThingSpeak Channel' (describing channels as data stores), 'Channel Settings' (listing configuration options like name, description, fields, metadata, tags, location, and public status), and 'Using the Channel' (explaining how to get data into a channel). A note at the bottom of the sidebar says 'See Tutorial: ThingSpeak and MATLAB for an example of measuring dew point from a'.

New Channel

Name: Test

Description: Test

Field 1: Field Label 1

Field 2:

Field 3:

Field 4:

Field 5:

Field 6:

Field 7:

Field 8:

Metadata:

Tags: (Tags are comma separated)

Make Public:

URL: _____

Help

ThingSpeak Channel

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Channel Name: Enter a unique name for the ThingSpeak channel.
- Description: Enter a description of the ThingSpeak channel.
- Field#: Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata: Enter information about channel data, including JSON, XML, or CSV data.
- Tags: Enter keywords that identify the channel. Separate tags with commas.
- Latitude: Specify the position of the sensor or thing that collects data in decimal degrees. For example, the latitude of the city of London is 51.5072.
- Longitude: Specify the position of the sensor or thing that collects data in decimal degrees. For example, the longitude of the city of London is -0.1275.
- Elevation: Specify the position of the sensor or thing that collects data in meters. For example, the elevation of the city of London is 35.052.
- Make Public: If you want to make the channel publicly available, check this box.
- URL: If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Video ID: If you have a YouTube® or Vimeo® video that displays your channel information, specify the full path of the video URL.

Using the Channel

You can get data into a channel from a device, website, or another ThingSpeak channel. You can then visualize data and transform it using [ThingSpeak Apps](#).

See [Tutorial: ThingSpeak and MATLAB](#) for an example of measuring dew point from a

IOT

Thingspeak

API & Upload Method

The screenshot shows the Thingspeak API Keys page. At the top, it displays the URL https://thingspeak.com/channels/128820/api_keys. The page has a navigation bar with links for Channels, Apps, Blog, and Support. On the left, there's a sidebar with the author information: Author: suakii and Access: Private. Below the sidebar, there are tabs for Private View, Public View, Channel Settings, API Keys (which is selected), and Data Import / Export.

Write API Key

Key: GFYKT9IUN7C0VFQB
Generate New Write API Key

Read API Keys

Key: 3LECTVLQV9GDHCBN
Note:
Save Note Delete API Key
Generate New Read API Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- Write API Key: Use this key to write data to a channel. If you feel your key has been compromised, click Generate New Write API Key.
- Read API Keys: Use this key to allow other people to view your private channel feeds and charts. Click Generate New Read API Key to generate an additional read key for the channel.
- Note: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

Create a Channel

```
POST https://api.thingspeak.com/channels.json  
api_key=5G98R8230IZOBIN2  
name=My New Channel
```

Update a Channel

```
PUT https://api.thingspeak.com/channels/128820  
api_key=5G98R8230IZOBIN2  
name=updated Channel
```

Clear a Channel

```
DELETE https://api.thingspeak.com/channels/128820/feeds.json  
api_key=5G98R8230IZOBIN2
```

Delete a Channel

```
DELETE https://api.thingspeak.com/channels/128820  
api_key=5G98R8230IZOBIN2
```

IOT

Thingspeak

API & Upload Method

The screenshot shows the Thingspeak API interface for channel 128820. The top navigation bar includes 'ThingSpeak', 'Channels', 'Apps', 'Blog', and 'Support'. The main page title is 'Test'. It displays channel details: Channel ID: 128820, Author: suakii, and Access: Private. Below this are tabs for 'Private View', 'Public View', 'Channel Settings', 'API Keys' (which is selected), and 'Data Import / Export'. The 'Import' section allows CSV file upload with a green 'Upload' button. The 'Export' section provides a green 'Download' button for CSV format. On the right side, several API endpoints are listed with their corresponding URLs:

- Update Channel Feed - GET: `GET https://api.thingspeak.com/update?api_key=GYK791UN7C0VFQ8&file=`
- Update Channel Feed - POST: `POST https://api.thingspeak.com/update.json`
 `api_key=GYK791UN7C0VFQ8`
 `field1=73`
- Get a Channel Feed: `GET https://api.thingspeak.com/channels/128820/feeds.json?results=`
- Get a Channel Field Feed: `GET https://api.thingspeak.com/channels/128820/fields/1.json?resul`
- Get Status Updates: `GET https://api.thingspeak.com/channels/128820/status.json`

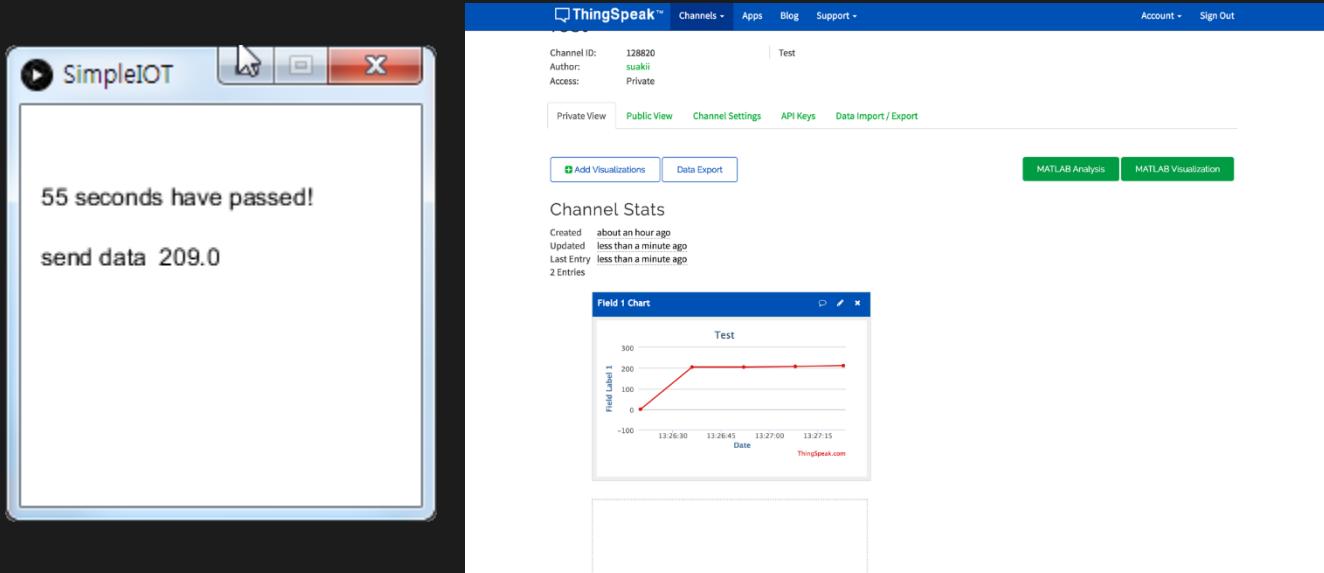
At the bottom, there's a note about selecting a CSV file and a 'Help' section with a 'Learn More' link.

Coding

Processing

<https://github.com/suakii/2017GSArduinoAd>

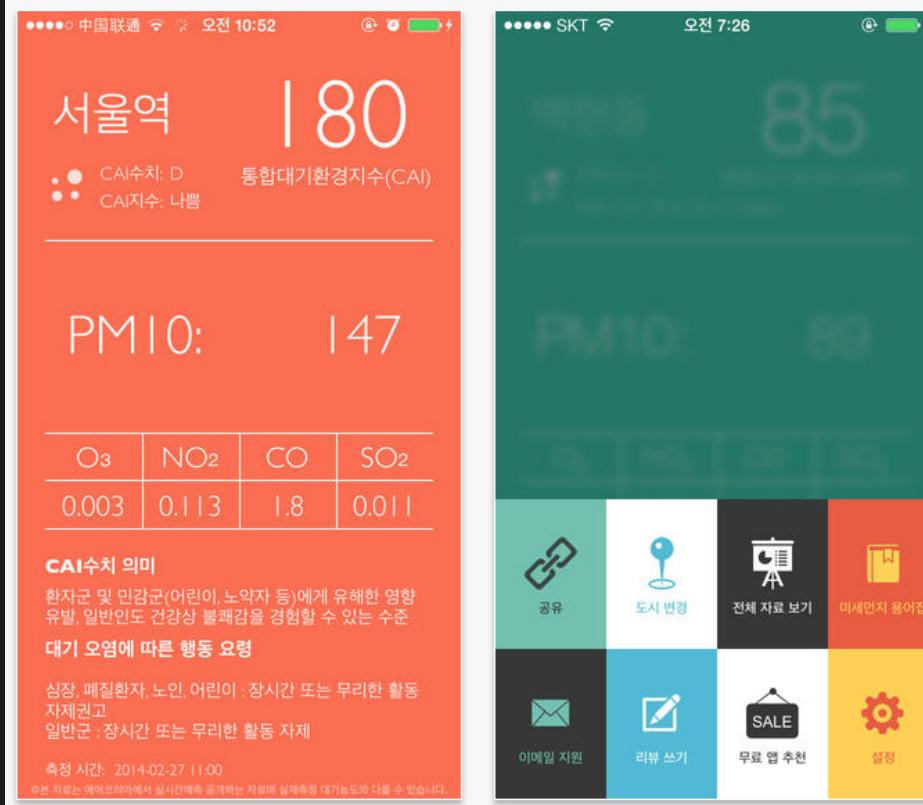
Coding Result



3

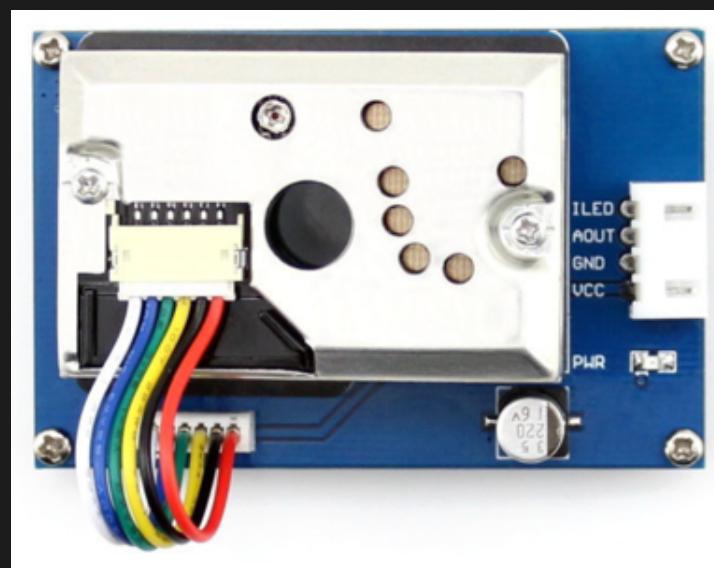
프로그램 소개

미세먼지



프로그램 소개

Hardware



미세먼지

미세먼지

But..

공공데이터 + Processing + Arduino

프로그램 소개

<http://www.airkorea.or.kr>

The screenshot shows the homepage of the Airkorea website. At the top, there's a navigation bar with links for Bookmarks, Korean version, ITP Wiki for Various, Syllabus / Nature, EAGLE 공부방, How Do Quadcopter, Coursera - Free Only, and a search bar for '6.006: Introduction'. The main header features the 'Airkorea' logo and the text '에이코리아판' (Airkorea Edition). Below the header, there are tabs for NEWS, 실시간 자료조회 (Real-time Data Inquiry), 대기질 예·경보 (Air Quality Forecast/Warning), 통계정보 (Statistical Information), 배움터 (Learning Center), and 고객의 소리 (Customer Voice). A news banner at the top left says '현재 황사, 오존 경보 발령내역 없음' (No current dust or ozone warning). The main content area has several sections: '기상정보' (Weather Information) with a map showing weather icons; '실시간 대기정보' (Real-time Air Information) with a map of South Korea showing green dots representing monitoring stations; '시도별 대기정보' (Regional Air Information); '대기질 예·경보' (Air Quality Forecast/Warning); and '우리동네 대기질' (Local Air Quality) with a detailed table for Seoul. At the bottom, there are links for various government services like the National Environmental Research Institute and the Ministry of Environment, along with footer information about copyright and language settings.

프로그램 소개

<https://www.data.go.kr>

The screenshot displays the homepage of the www.data.go.kr website. At the top, there is a navigation bar with tabs for '일반 사용자' (General User), '로그아웃 | 사이트맵 | 마이페이지 | ENGLISH |' (Logout | Site Map | My Page | English), and social media links for Facebook, Twitter, and YouTube. The main header features the '정부 3.0 DATA' logo and a search bar with the placeholder '미세먼지' (PM2.5). Below the header, a blue banner titled 'STANDARD DATA' contains icons for various government sectors: Education, National Security, Public Finance, Employment, Social Welfare, Consumer Safety, Culture, Health Services, Environment, Science and Technology, Agriculture, Energy, and Justice.

Below the banner, there are three featured sections:

- 국가 중점개방 데이터**: A circular graphic showing various data icons like a smartphone, globe, and charts, with the text "국민의 손으로 직접 선정한 '국가 중점개방 데이터' 36대 분야를 대용량 데이터로 개방합니다!" (36 key fields selected by citizens directly, opening as large data sets).
- 공공데이터 활용사례**: A thumbnail for "ARCHiTIES" featuring an image of a modern building and the text "건축, 시각부터 끝까지, 건설업체, 자재 등 건축 분야의 다양한 정보를 제공하는 협업서비스입니다..." (A collaboration service providing various information from architecture to construction, construction companies, materials, etc.).
- 인기 데이터**: A section showing four cards: 1. 제주도 유동인구 현황(관광객...) (Jeju Island tourism statistics); 2. 제주도 유동인구 현황(도민...) (Jeju Island resident statistics); 3. 교통사고통계 (Traffic accident statistics); and 4. 상가(상권)정보 (Business/Commercial area information).

프로그램 소개

<https://www.data.go.kr>

The screenshot shows the homepage of the DATA.GOV.KR website. The URL in the address bar is <https://www.data.go.kr/oossearch.do>. The page title is "데이터셋 > 오픈API | 공공데이터포털". The main navigation menu includes "데이터셋", "활용사례", "참여마당", "정보공유", and "전체메뉴". A search bar at the top has the placeholder "미세먼지". Below the search bar, there are four tabs: "전체(12)", "파일데이터(9)", "오픈API(3)", and "표준데이터(0)". The "오픈API(3)" tab is selected. The main content area displays three search results for "오픈API [3건]".

- 대기오염정보 조회 서비스** (조회수: 5,032, 활용신청수: 1,999)
수정일: 2016.06.17 기관: 한국환경공단 서비스유형: REST
각 층정소별 대기오염정보를 조회하기 위한 서비스로 기간별, 시도별 대기오염 정보와 민간군 이상 층정소 내역, 미세먼지...
[XML](#)
- 한국동서발전 발전소 주변 대기환경(기상, 대기질)** (조회수: 303, 활용신청수: 11)
수정일: 2014.12.24 기관: 한국동서발전(주) 서비스유형: REST
발전소 주변지역의 대기질 정보 및 기상정보를 제공
[XML](#)
- 대기질 실시간 층정자료** (조회수: 789, 활용신청수: 98)
수정일: 2016.01.22 기관: 부산광역시 서비스유형: REST
부산연안의 21개 층정소의 대기층정자료
[XML](#)

On the right side, there is a sidebar with filters and a search bar:

- 인기검색어**
 - 제주도
 - 버스
 - 미세먼지
 - 출렁복도
 - 날씨
- 국가충정데이터**
 - 서비스유용필터(OPENAPI)
 - 제공기관필터
 - 분류체계필터
 - 이용허락범위필터
 - 태그
 - 학자자

프로그램 소개

<https://www.data.go.kr>

The screenshot shows the homepage of the DATA.go.kr website. At the top, there is a navigation bar with links for '데이터셋', '활용사례', '참여마당', '정보공유', and '트렌체메뉴'. Below the navigation bar, there is a search bar and a link to '이달의 추천 데이터'. The main content area features a sidebar on the left with sections for '제공기관', '한국환경공단', '등록일', '2013-12-05', '키워드', '보유근거', '행정업무', and '공유'. A red banner labeled '환경기부' is overlaid on the sidebar. To the right of the sidebar, there is a section titled '대기오염정보 조회 서비스' with a sub-section for 'XML 대기오염정보 조회 서비스'. Below this, there is a section titled '연관 데이터셋' with several listed datasets. At the bottom of the page, there is a footer with links for '개인정보처리방침', '공공데이터포털 약관', '사이트맵', '문영자에게 예길보내기', '개인정보분야별책임자: 정부만', '관련 사이트', '이동', and 'FAQ'.

프로그램 소개

<https://www.data.go.kr>

The screenshot shows the DATA GO KR website's 'My Page' section. On the left sidebar, there are links for 'Open API' (including 'Development Account', 'Usage History', 'Operation Log', and 'API Key Management'), 'DATA', 'My Profile', 'My Address', 'Membership Information', and 'Public Data Submission'. The main content area is titled '개발계정 상세보기' (Development Account Detail View). It displays detailed information about a service named '대기오염정보 조회 서비스' (Real-time Air Pollution Inquiry Service). The service is a REST API with 0 daily requests and 0 concurrent users. It is used by '자동승인' (Automated Approval) with '신청유형' (Application Type) set to '개발계정 | 변경신청' (Development Account | Change Application) and '처리상태' (Status) set to '승인' (Approved). The service was active from June 20, 2016, to June 20, 2018. Below this, there are sections for 'Service Information' (including 'UTF-8' encoding, 'End Point' at <http://openapi.airkorea.or.kr/>, and 'Data Format' as XML), 'Technology Document' (link to IROS3_OA_DV_0701_OpenAPI 활용 가이드), and 'Detailed Information' (two rows of data with columns for NO, 상세기능명 (Detail Function Name), 설명 (Description), 활용제한여부 (Usage Limitation), 일일 트래픽 (Daily Traffic), 처리결과 (Processing Result), and 미리보기다운로드 (Preview Download)). Row 1 is for '측정소별 실시간 측정데이터 기간(일, 한달, 3개월)으로 해당 측정소의 일반량 및 측정정보 조회 기능 제공' (Provide real-time measurement data for specific locations by day, month, or 3 months, including general quantities and measurement information). Row 2 is for '통합마기환경지수 미감금이상 측정 통합마기환경지수(CAI)가 나를 이상인 측정소의 빠른 조회 기능 제공' (Provide a quick search function for CAI values exceeding the limit, specifically for locations where CAI is above the limit).

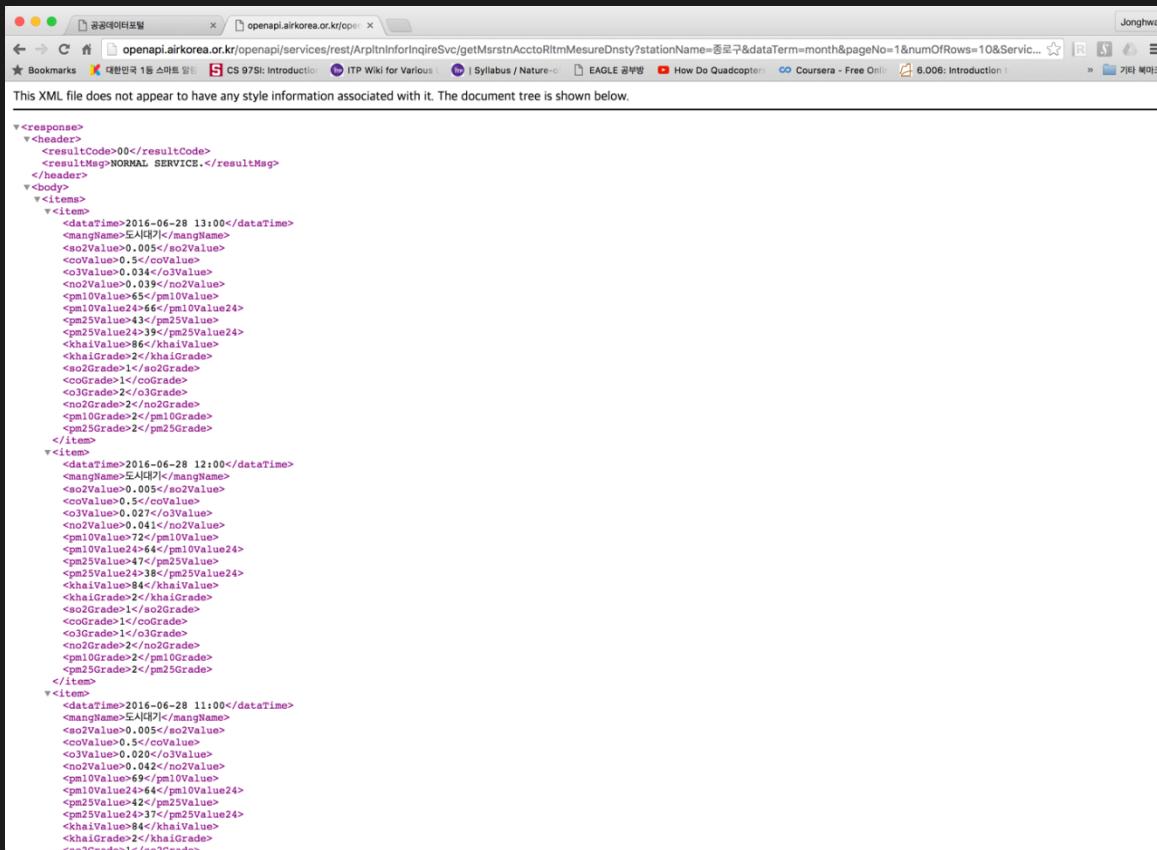
프로그램 소개

인증키 Test

<http://openapi.airkorea.or.kr/openapi/services/rest/ArpltnInforInqireSvc/getMsrstnAcctoRltmMeasureDnsty?stationName=종로구&dataTerm=month&pageNo=1&numOfRows=10&ServiceKey=8n%2BI3gJJn6o2pN%2BwlFHj6i9anqdFAjpD97TWOc%2FeV4zynQPsgQ3TCxhgua6DYylpMGG%2Fn1%2BH7rX%2Fya6uDkcltA%3D%3D&ver=1.2>

프로그램 소개

인증키 Test



The screenshot shows a web browser window with the URL openapi.airkorea.or.kr/openapi/services/rest/ArpltnInforInquireSvc/getMsrstrnAcctoRltmMesureDnsty?stationName=종로구&dataTerm=month&pageNo=1&numOfRows=10&ServiceKey=CS_97Si. The page displays an XML document representing air quality data for Seoul, Jongno-gu. The XML structure includes headers, body, items, and various data values for different pollutants and grades.

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <header>
        <resultCode>00</resultCode>
        <resultMsg>NORMAL SERVICE.</resultMsg>
    </header>
    <body>
        <items>
            <item>
                <dateTime>2016-06-28 13:00</dateTime>
                <mangName>도시락기</mangName>
                <so2Value>0.005</so2Value>
                <coValue>0.5</coValue>
                <j3Value>0.034</j3Value>
                <no2Value>0.039</no2Value>
                <pm10Value>59</pm10Value>
                <pm25Value>46</pm25Value>
                <pm25Value>43</pm25Value>
                <pm25Value>39</pm25Value>
                <khaiValue>86</khaiValue>
                <khaiGrade>2</khaiGrade>
                <so2Grade>1</so2Grade>
                <coGrade>1</coGrade>
                <j3Grade>2</j3Grade>
                <no2Grade>2</no2Grade>
                <pm10Grade>2</pm10Grade>
                <pm25Grade>2</pm25Grade>
            </item>
            <item>
                <dateTime>2016-06-28 12:00</dateTime>
                <mangName>도시락기</mangName>
                <so2Value>0.005</so2Value>
                <coValue>0.5</coValue>
                <j3Value>0.027</j3Value>
                <no2Value>0.041</no2Value>
                <pm10Value>72</pm10Value>
                <pm25Value>44</pm25Value>
                <pm25Value>47</pm25Value>
                <pm25Value>38</pm25Value>
                <khaiValue>84</khaiValue>
                <khaiGrade>2</khaiGrade>
                <so2Grade>1</so2Grade>
                <coGrade>1</coGrade>
                <j3Grade>1</j3Grade>
                <no2Grade>2</no2Grade>
                <pm10Grade>2</pm10Grade>
                <pm25Grade>2</pm25Grade>
            </item>
            <item>
                <dateTime>2016-06-28 11:00</dateTime>
                <mangName>도시락기</mangName>
                <so2Value>0.5</so2Value>
                <coValue>0.5</coValue>
                <j3Value>0.020</j3Value>
                <no2Value>0.042</no2Value>
                <pm10Value>69</pm10Value>
                <pm10Value>64</pm10Value>
                <pm25Value>44</pm25Value>
                <pm25Value>37</pm25Value>
                <khaiValue>80</khaiValue>
                <khaiGrade>2</khaiGrade>
                <so2Grade>1</so2Grade>
                <coGrade>1</coGrade>
```

프로그램 소개

영통동 조회

```
<response>
<header>
<resultCode>00</resultCode>
<resultMsg>NORMAL SERVICE.</resultMsg>
</header>
<body>
<items>
<item>
<dateTime>2016-06-28 13:00</dateTime>
<so2Value>-</so2Value>
<coValue>-</coValue>
<o3Value>-</o3Value>
<no2Value>-</no2Value>
<pm10Value>-</pm10Value>
<khaiValue>-</khaiValue>
<khaiGrade/>
<so2Grade/>
<coGrade/>
<o3Grade/>
<no2Grade/>
<pm10Grade/>
</item>
.....
<numOfRows>10</numOfRows>
<pageNo>1</pageNo>
<totalCount>742</totalCount>
</body>
</response>
```

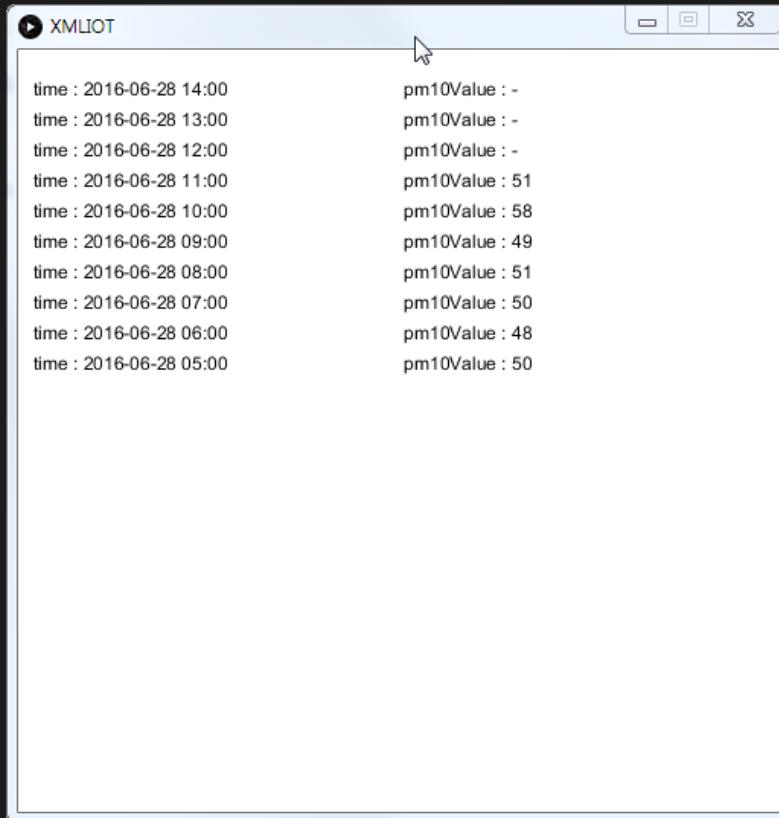
Coding

Processing - XMLIOT

<https://github.com/suakii/2017GSArduinoAd>

Coding

Result



The screenshot shows a terminal window titled "XMLIOT". The window contains a list of 12 entries, each consisting of a timestamp followed by a "pm10Value" field. The timestamps are in descending order from top to bottom, starting at 14:00 and ending at 05:00 on June 28, 2016. The "pm10Value" field contains either a dash "-" or a numerical value (51, 58, 49, 51, 50, 48, 50).

time	pm10Value
2016-06-28 14:00	-
2016-06-28 13:00	-
2016-06-28 12:00	-
2016-06-28 11:00	51
2016-06-28 10:00	58
2016-06-28 09:00	49
2016-06-28 08:00	51
2016-06-28 07:00	50
2016-06-28 06:00	48
2016-06-28 05:00	50

미세먼지

미세먼지

예보	농도
좋은	0~30 $\mu\text{g}/\text{m}^3$
보통	31~80 $\mu\text{g}/\text{m}^3$
약간 나쁨	81~120 $\mu\text{g}/\text{m}^3$
나쁨	121~200 $\mu\text{g}/\text{m}^3$
매우 나쁨	201~300 $\mu\text{g}/\text{m}^3$ 이상

Coding

Processing - XMLIOT-Motor

<https://github.com/suakii/2017GSArduinoAd>

Coding

Result

