

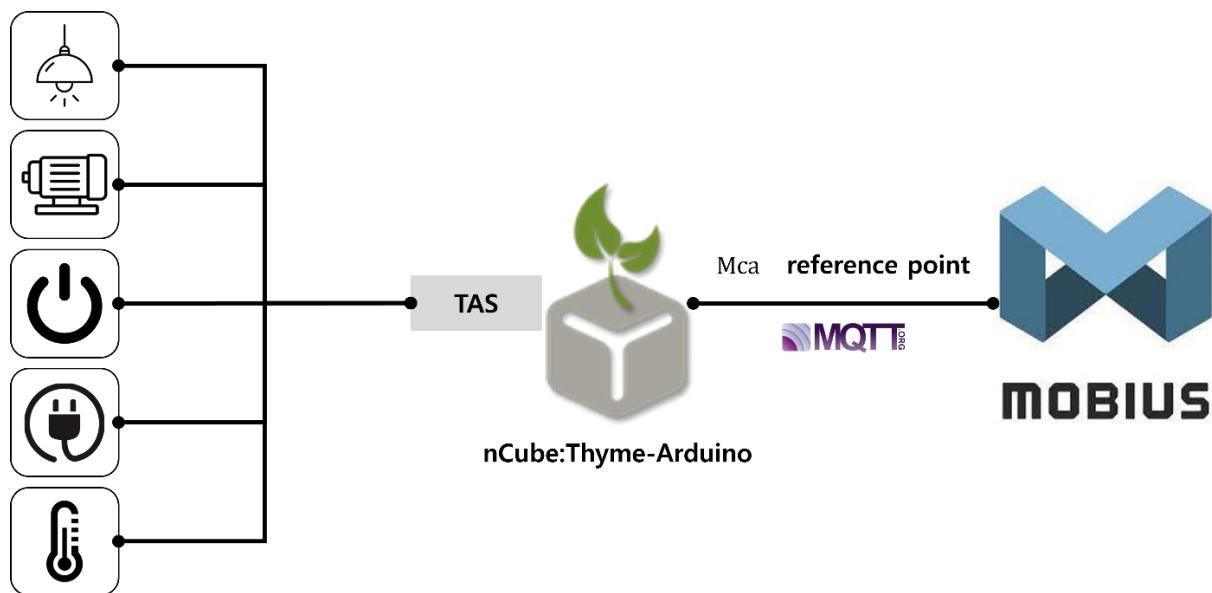


OCEAN

nCube-Thyme-Arduino

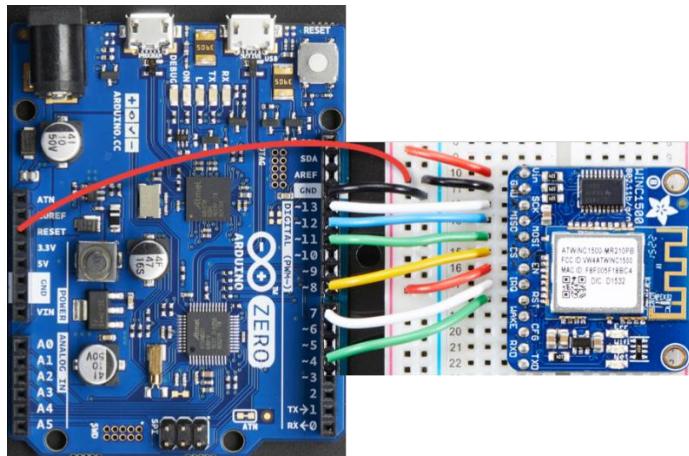
Ver.1.0

nCube-Thyme-Arduino



nCube-Thyme-Arduino는 전자부품연구원에서 개발한 oneM2M IoT 표준을 통해 개발된 CSE(Common Service Entity, 사물인터넷 서비스 플랫폼)인 Mobius와 oneM2M 표준에 정의된 Mca reference point를 통해 연결된다. nCube-Thyme-Arduino는 아두이노를 사용하여 oneM2M IoT 표준의 AE(Application Entity)를 구현한 것이며, 다양한 센서 데이터 측정 및 업로드와 사용자의 명령에 따른 액추에이터 동작을 할 수 있다. 개발 언어는 Arduino IDE에서 지원하는 C, C++이며 MQTT를 사용하여 사물인터넷 서비스 플랫폼과 연결된다.

nCube-Thyme-Arduino 가이드



<Arduino zero>



<Feather M0>

본 가이드는 nCube-Thyme-Arduino는 cortex-M0를 기반으로 작동하는 Arduino Zero와 WiFi 모듈을 포함하고 있는 Adafruit 사의 Feather M0를 기반으로 제작되었다. Mobius IoT Platform에 생활속에서 수집될 수 있는 다양한 정보를 업로드하고, 사용자의 동작 제어를 제공하는 과정에 대한 설명을 목적으로 하고 있다. 또한, nCube-Thyme-Arduino를 활용하여 Feather M0 보드에서 Co2 농도 측정 및 업로드하고 LED 제어를 해보는 예제를 통해 설명한다.

목차

| | | |
|-------|---------------------------------|----|
| 1 | Introduction | 6 |
| 1 - 1 | Adafruit Feather M0..... | 6 |
| 1 - 2 | CO ₂ 센서..... | 7 |
| 1 - 3 | RGB-LED..... | 7 |
| 2 | 환경설정 | 8 |
| 2 - 1 | Visual Studio Code 설치..... | 8 |
| 2 - 2 | Arduino IDE 환경설정..... | 9 |
| 2 - 3 | Adafruit Feather M0 동작 테스트..... | 14 |
| 3 | nCube-Mint 개발 환경 | 16 |
| 3 - 1 | nCube-Mint 라이브러리 다운로드..... | 16 |
| 3 - 2 | OneM2MClient 라이브러리 소개 | 17 |
| 3 - 3 | nCube-Mint 불러오기 | 19 |
| 3 - 4 | nCube-Mint 스케치 실행..... | 20 |
| 3 - 5 | nCube-Mint 기본 Resource 조회..... | 22 |
| 4 | 센서별 Cube-Mint 응용 개발 | 24 |

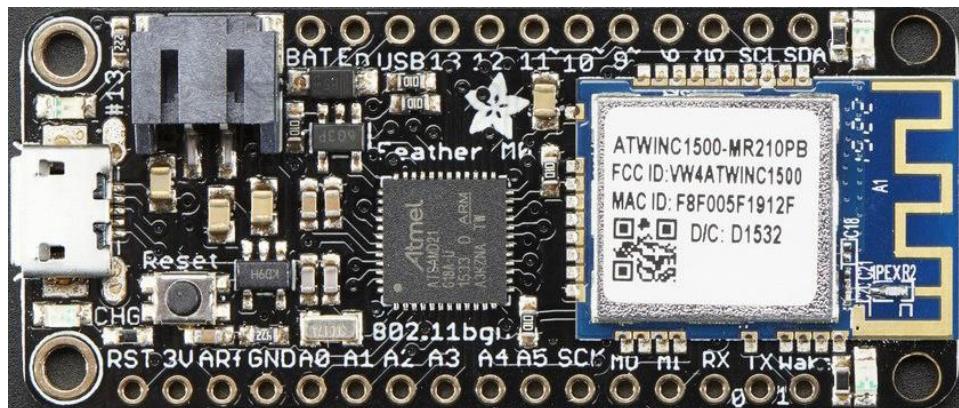
nCube-Thyme-Arduino 가이드

| | |
|---|----|
| 4 - 1 nCube-Mint + Co2 Sensor..... | 24 |
| 4 - 1 - 1 Co2 센서 연결..... | 24 |
| 4 - 1 - 2 TasCO2 라이브러리 소개 | 25 |
| 4 - 1 - 3 nCube-Mint + Co2 Sensor 스케치 | 26 |
| 4 - 1 - 4 nCube-Mint Co2 동작과 데이터 조회..... | 28 |
| 4 - 2 RGB-LED 연결 | 30 |
| 4 - 2 - 1 RGB-LED 연결 | 30 |
| 4 - 2 - 2 nCube-Mint + LED 스케치 | 31 |
| 4 - 2 - 3 nCube-Mint LED 동작과 제어 | 34 |
| 4 - 3 Adafruit Powre Relay FeatherWing 연결 | 37 |
| 4 - 3 - 1 Adafruit Powre Relay FeatherWing 연결 | 37 |
| 4 - 3 - 2 nCube-Mint + Powre Relay 스케치..... | 38 |
| 4 - 3 - 3 nCube-Mint Power Relay 동작과 제어 | 40 |
| 5 전체 센서 통합 nCube-Mint 개발 및 테스트 | 43 |
| 5 - 1 nCube-Mint 회로 구성..... | 43 |
| 5 - 2 nCube-Mint 스케치..... | 44 |
| 5 - 3 nCube-Mint 동작과 데이터 조회, 제어 | 48 |
| Appendix A | 50 |
| Appendix B | 52 |

1 Introduction

1장에서는 본 가이드에서 사용된 하드웨어에 대해 간단히 설명한다. 기본이 되는 Adafruit Feather M0에 대한 소개를 시작으로, Co2 센서, RGB-LED 센서에 대해서 설명한다.

1 - 1 Adafruit Feather M0



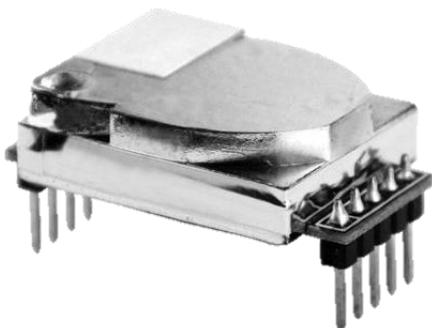
Adafruit 사에 의해 제작 및 판매되고 있다. Cortex M0 프로세서를 포함하고 있으며 I/O pin 들과, 여러개의 12-bit ADC(아날로그-디지털 변환회로), 하나의 10-bit DAC(디지털- 아날로그 변환회로), SPI, I2C, UART 통신을 지원하는 총 6 개의 SERCOM, native USB 지원, 802.11bgn을 지원하는 Atmel WINC1500 WiFi 모듈을 포함하는 등의 특징을 갖는다.

Adafruit Feather M0는 저전력 설계가 적용되어 IoT Device로 사용되기에 적절하다.

WiFi 모듈은 전력관리칩을 포함하고 있으며 12mA의 전력 소비량을 가지고, Cortex M0 칩은 10mA의 전력 소비량을 가진다.

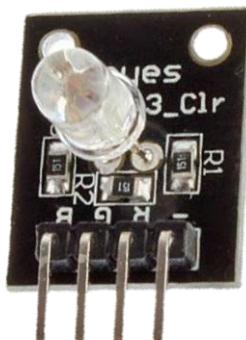
Micro USB를 통한 전원 입력과, 3.7V 리튬 폴리머 배터리를 이용한 전원 입력이 모두 가능하며, 배터리가 연결된 상태에서 Micro USB를 통한 전원이 입력되면 배터리를 충전한다. 또한 위의 보드사진에서 좌측 하단에 위치한 3V 핀을 통해서 최대 600mA의 전력 출력을 지원한다.

1 - 2 Co2 센서



CM1106 Co2 센서는 대기중의 Co2 농도를 측정하여 UART 포트를 이용해 Arduino로 업로드한다. 4 바이트(0x11, 0x01, 0x01, 0xED) 입력값을 통해 8 바이트의 출력값(0x16, 0x05, 0x01, 0x02, 0x72, 0x01, 0xD6, 0x9)을 얻을 수 있으며, 5, 6 바이트의 출력값 0x02, 0x72 가 Co2 농도를 나타내는 수치로써, 위의 예에서 0x0272 = 626, 즉 Co2 농도가 626ppm임을 나타낸다.

1 - 3 RGB-LED



FE-RGB 3 색 LED 보드는 각 R, G, B 핀의 입력에 따라 Red, Green, Blue 3가지 색을 나타낸다. 기본적인 RGB 3 가지 불빛 이외에도, 입력값의 조합에 따라 하나 이상의 빛($2(\text{LED On, OFF})^3(\text{LED 3 개}) - 1(\text{RGB off case}) = 7$ 가지)을 낼 수 있다.

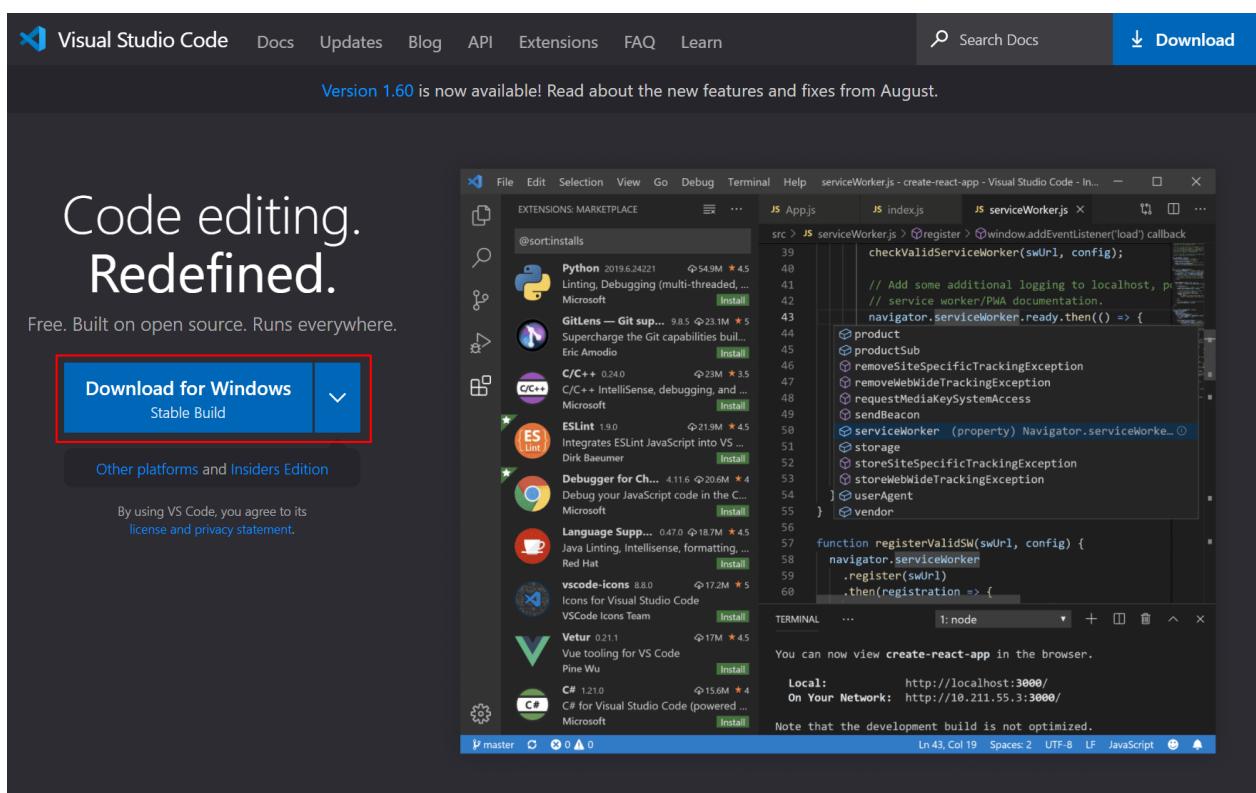
2 환경설정

2 장에서는 nCube-Thyme-Arduino를 사용하기 위한 환경설정에 대해 설명한다.

Visual Studio Code(VSCode)와 Platform IO IDE의 설치를 진행하고 간단한 예제를 통해 Feather M0 보드를 테스트해보는 과정을 소개하겠다.

단, 본 문서는 Window OS를 기반으로 제작되었으며, Mac OS 도 기본적인 과정은 비슷하다.

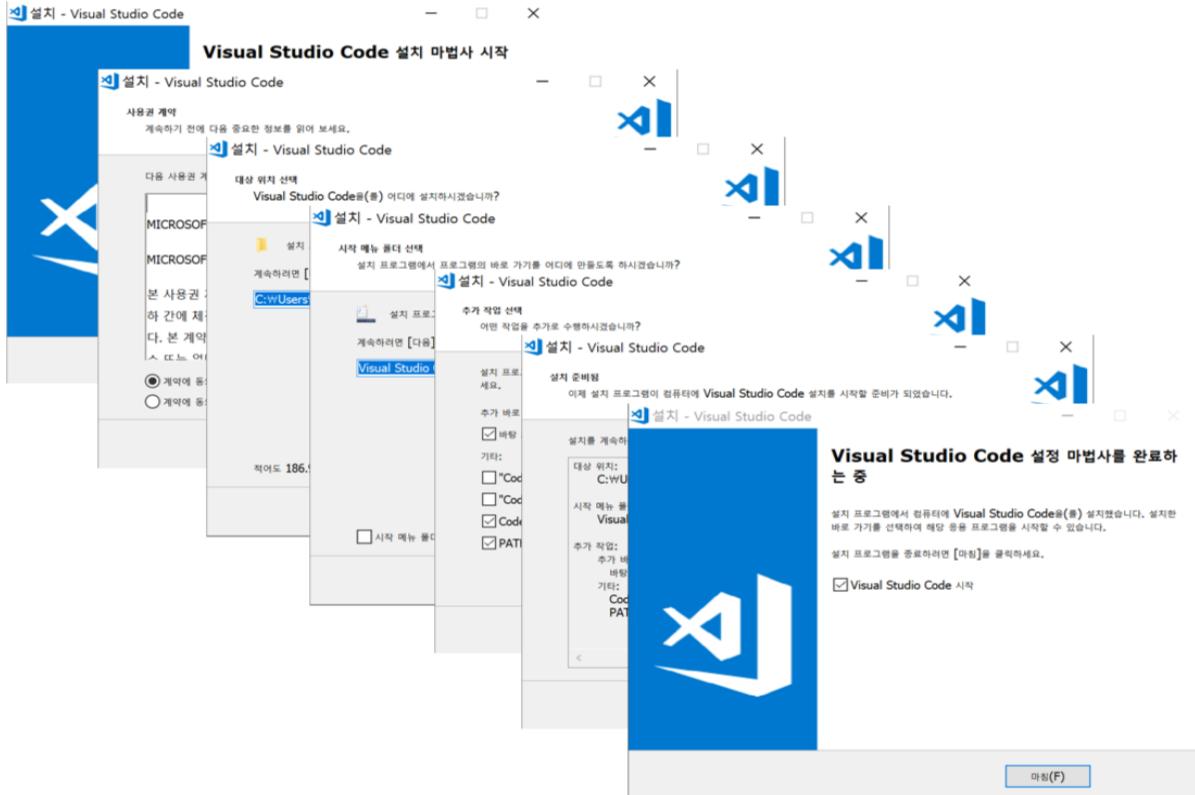
2 - 1 Visual Studio Code 설치



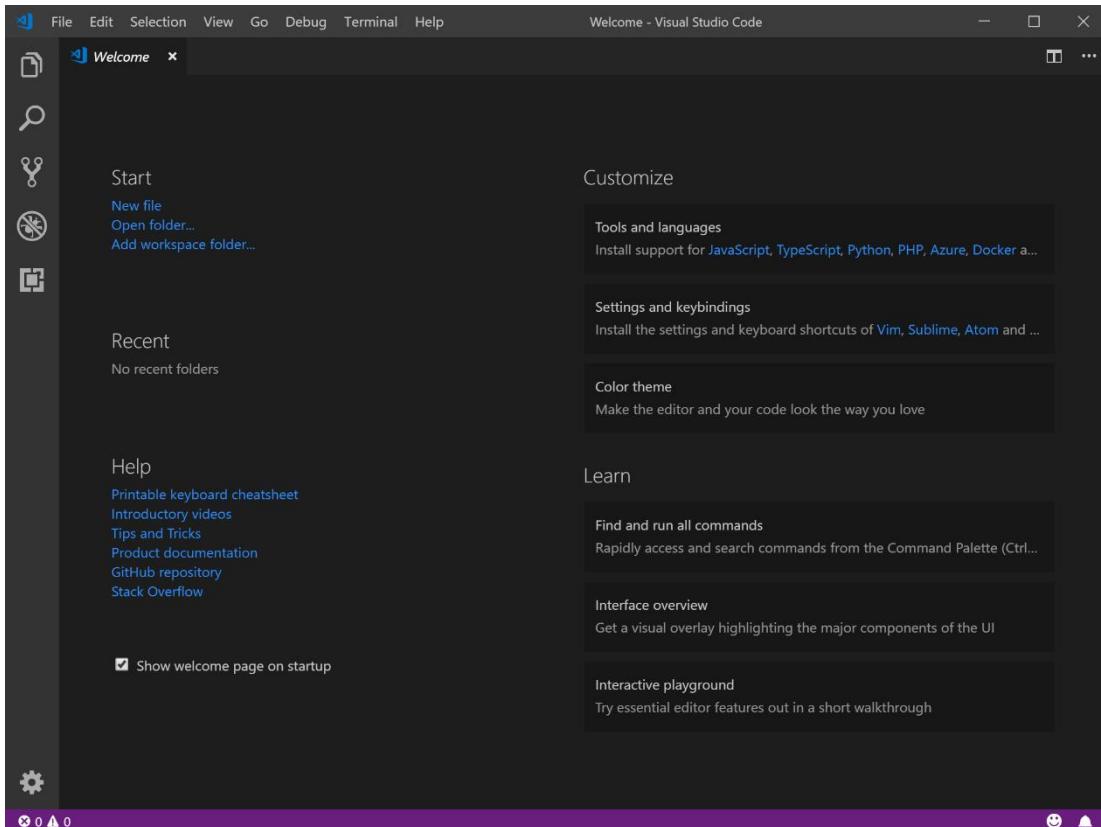
<https://code.visualstudio.com/>

위의 URL로 들어가여 Visual Studio Code를 다운받는다. 기본적으로 본인의 OS에 맞게 설치파일이 다운로드 되나, 간혹 오류로 인해 다른OS의 설치파일이 다운로드 될 수 있으니 확인하고 다운로드 한다.

nCube-Thyme-Arduino 가이드

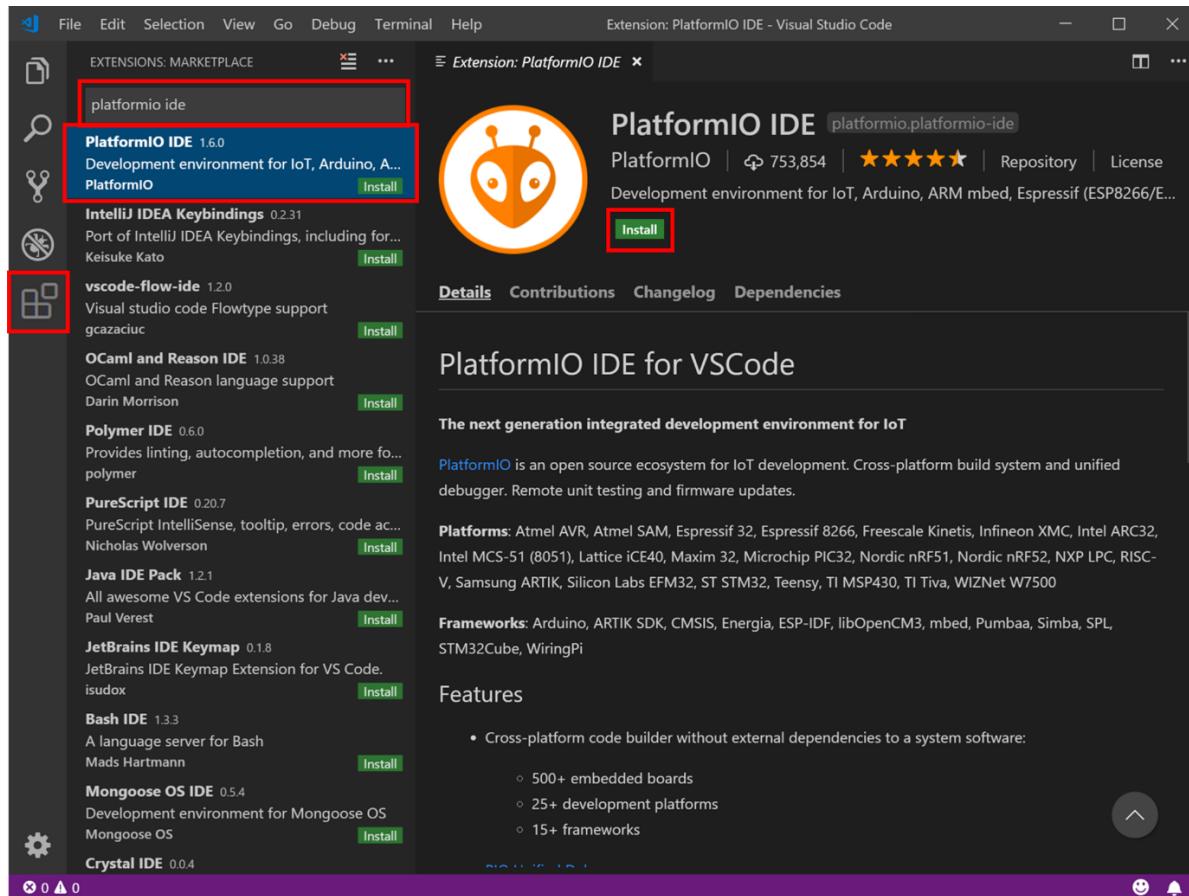


다운로드 한 설치파일을 실행시켜 VS Code를 설치한다.



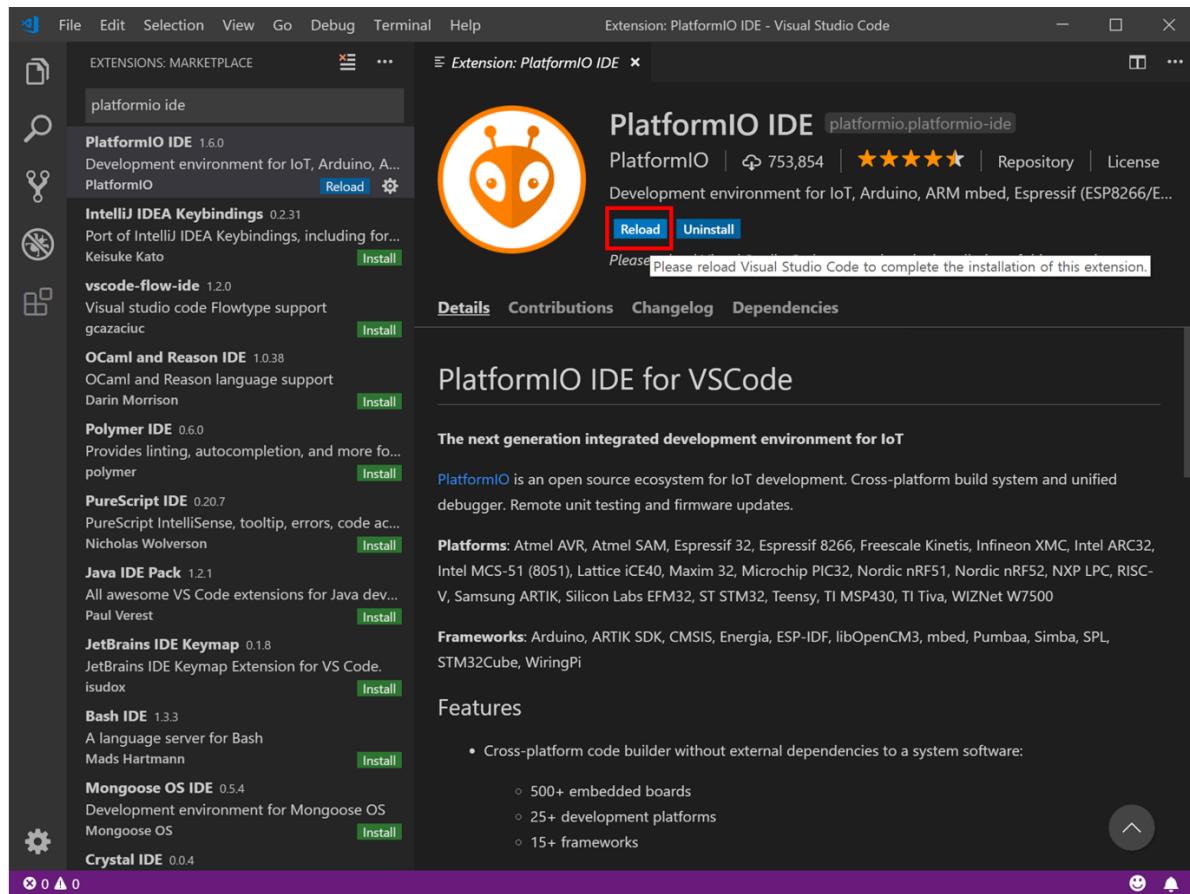
설치가 완료된 후 VS Code를 실행한 화면이다.

2 - 2 PlatformIO IDE 설치



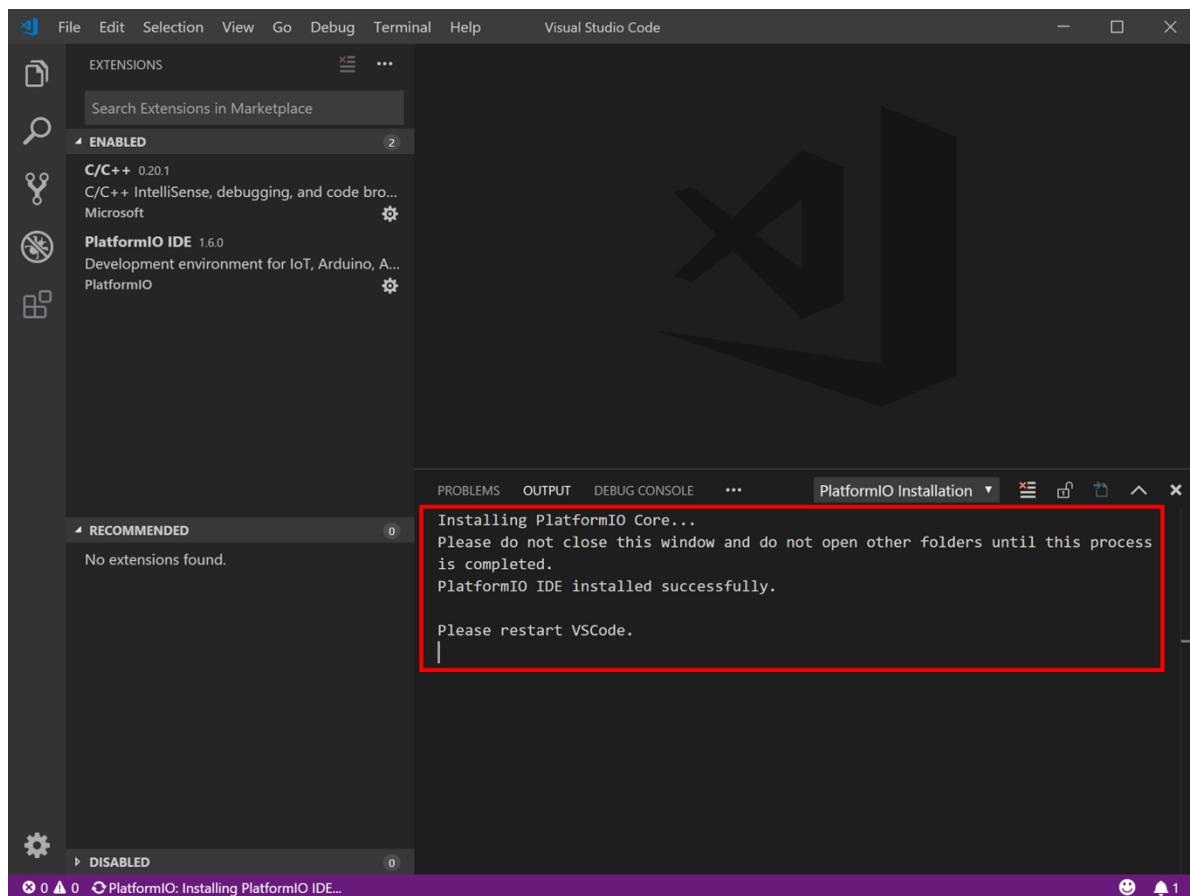
VS Code의 왼쪽 사이드바에서 Extensions(Ctrl+Shift+X) 탭을 열고 `Platform IO IDE`를 검색하여 install 버튼을 눌러 설치한다.

nCube-Thyme-Arduino 가이드



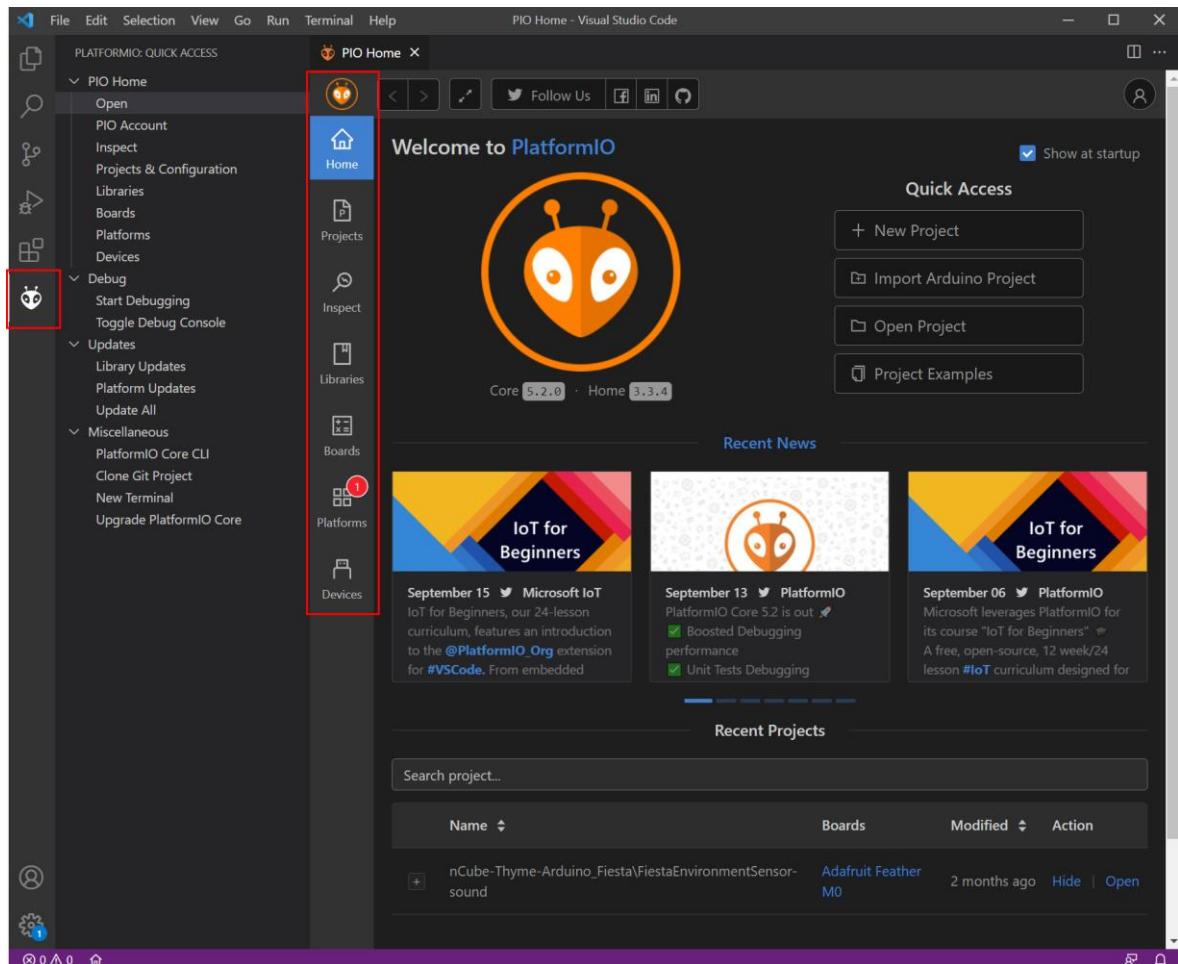
PlatformIO IDE 확장기능이 설치된 후 Reload 버튼이 보이면 이 버튼을 누른다.

nCube-Thyme-Arduino 가이드



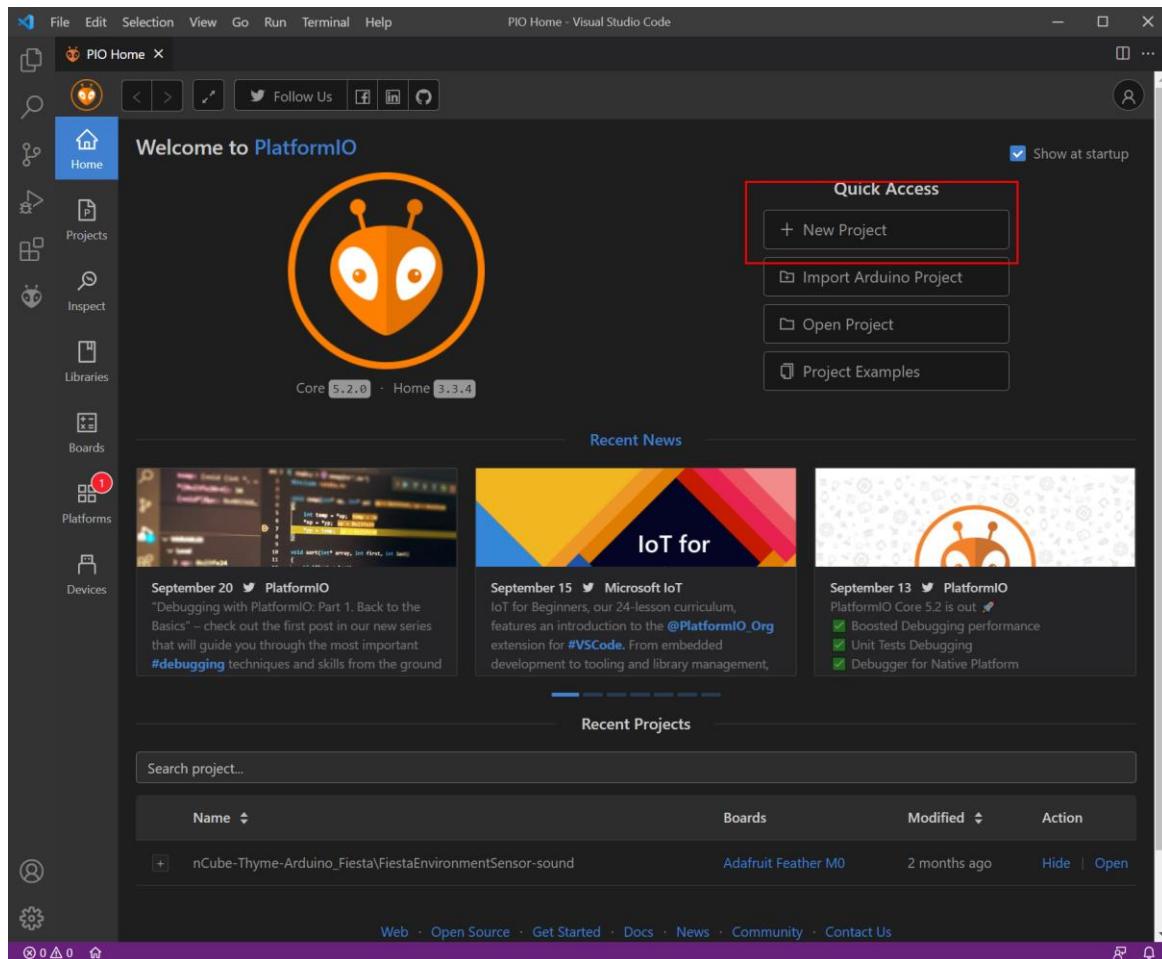
이후 VS Code가 재실행 된 뒤에 PlatformIO IDE가 추가적으로 필요한 패키지를 자동으로 설치하며, 'Please restart VSCode'라는 문구가 나타나면 다시 VS Code를 재실행한다.

nCube-Thyme-Arduino 가이드



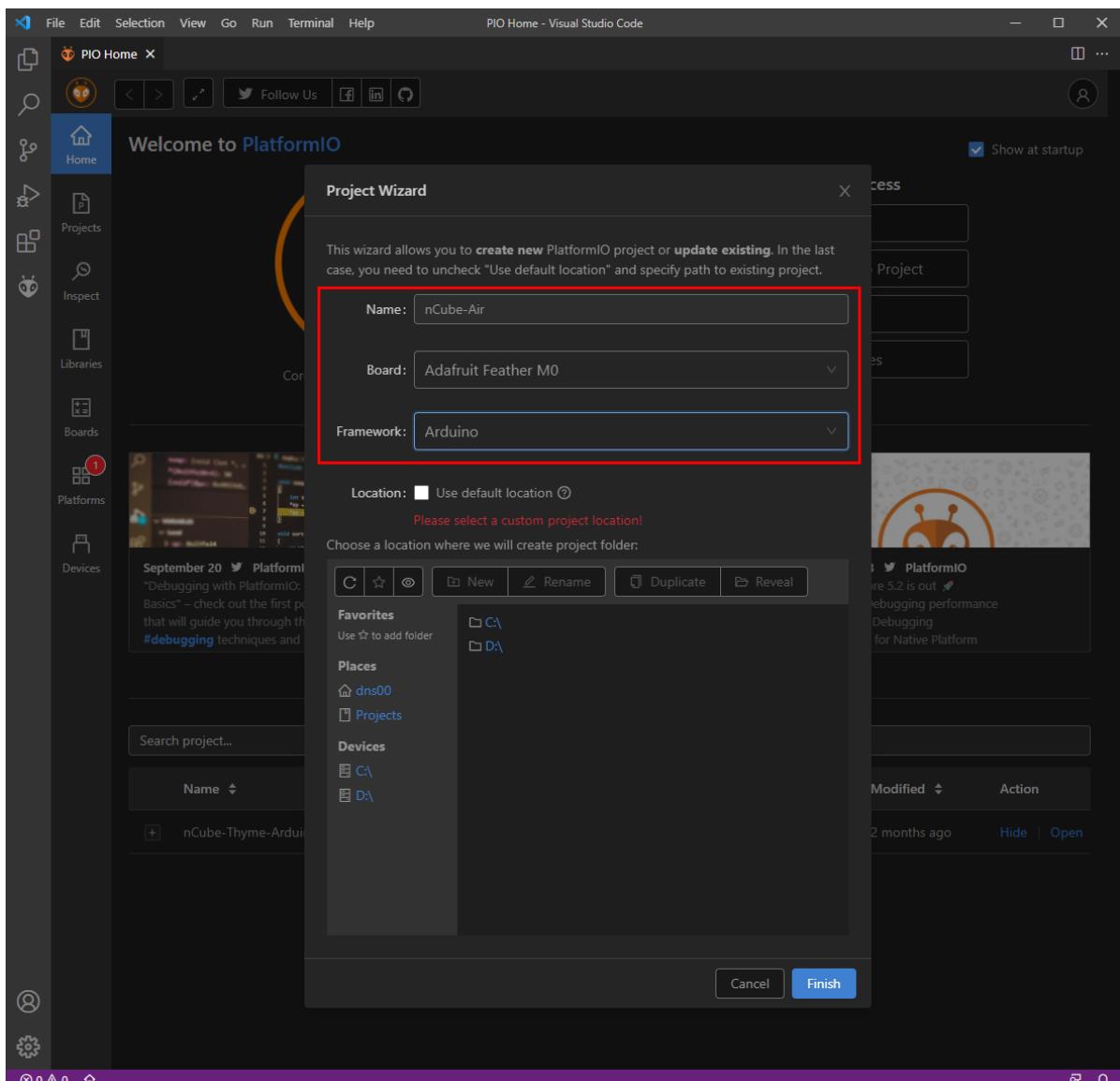
VS Code가 재실행 된 이후에 VS Code의 사이드바에 PlatformIO IDE버튼이 추가되며, PlatformIO IDE의 홈 화면이 나타난다.

2 - 3 nCube-Thyme-Arduino 환경설정



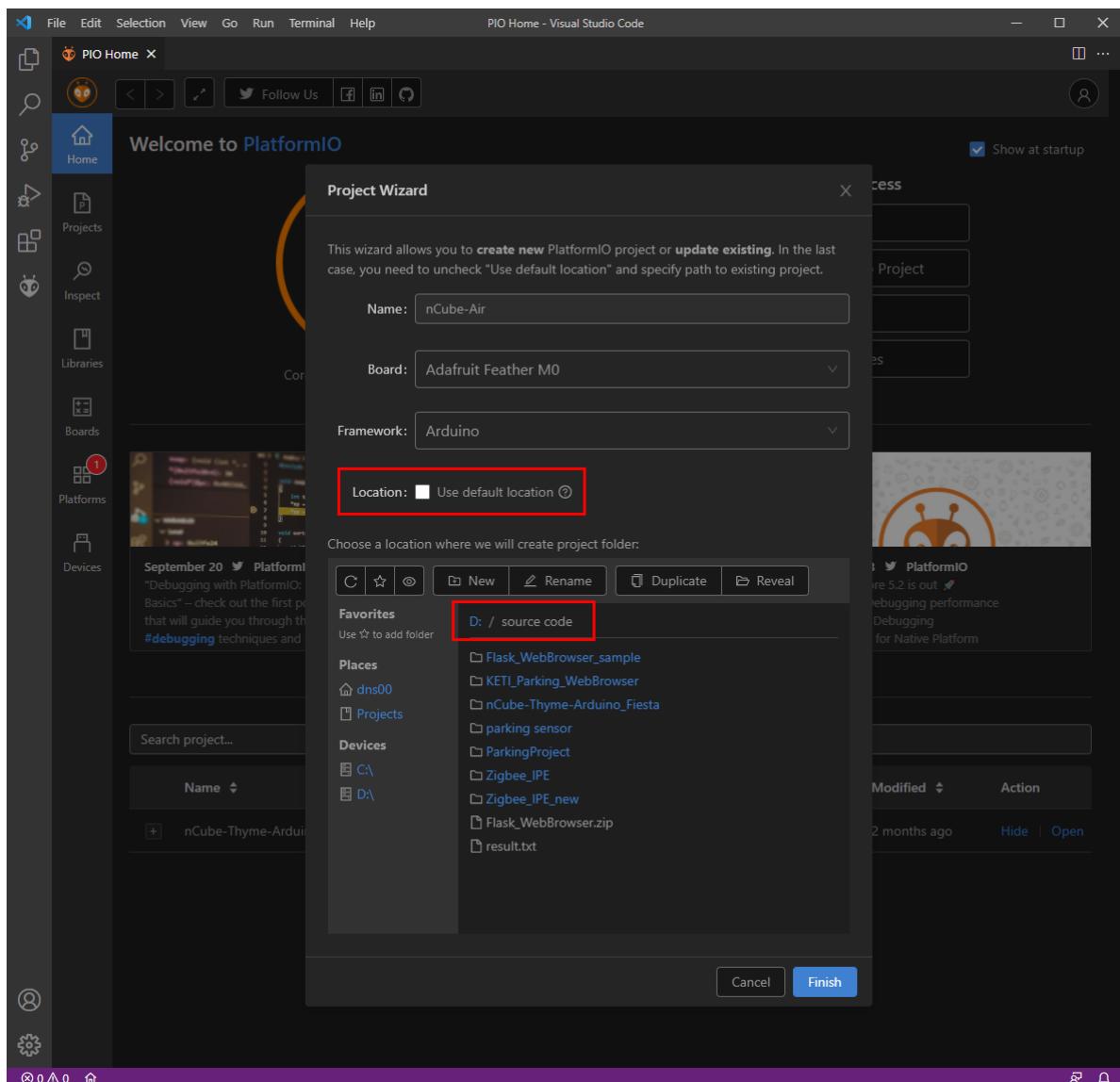
PlatformIO IDE의 홈 화면에서 New Project 버튼을 클릭한다.

nCube-Thyme-Arduino 가이드



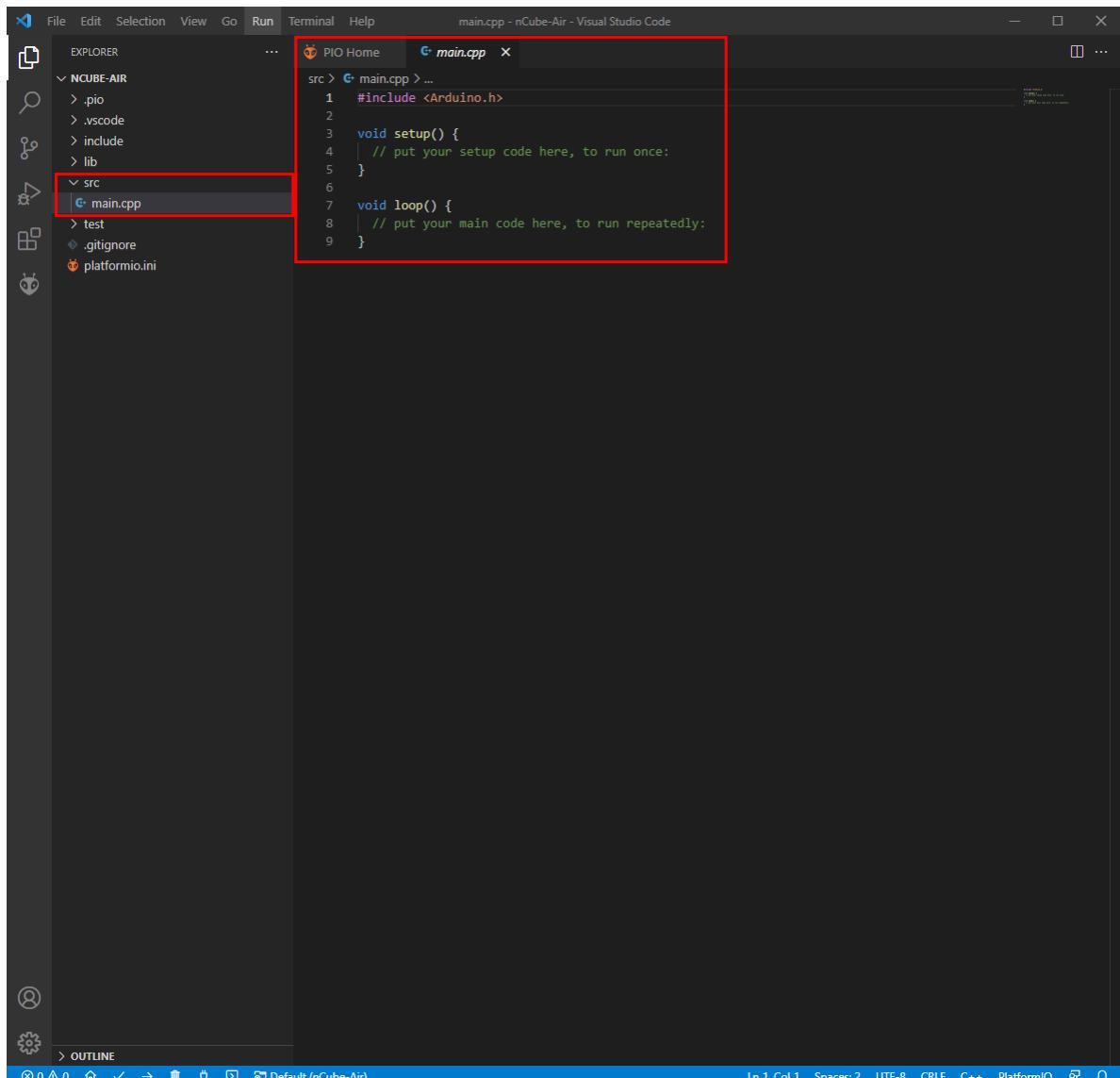
Name에는 원하는 프로젝트 이름, Board에는 'Adafruit Feather M0', Framework에는 'Arduino'로 설정한다.

nCube-Thyme-Arduino 가이드



Location의 Use default location을 해제하고, 원하는 프로젝트 폴더 생성 장소를 선택한 뒤에 Finish버튼을 누른다.

nCube-Thyme-Arduino 가이드



The screenshot shows the Visual Studio Code interface with the PlatformIO extension open. The Explorer sidebar on the left displays the project structure under 'NCUBE-AIR': .pio, .vscode, include, lib, src, test, .gitignore, and platformio.ini. The 'src' folder is expanded, and the 'main.cpp' file is selected and highlighted with a red box. The main editor area shows the contents of main.cpp:

```
src > main.cpp > ...
1 #include <Arduino.h>
2
3 void setup() {
4     // put your setup code here, to run once:
5 }
6
7 void loop() {
8     // put your main code here, to run repeatedly:
9 }
```

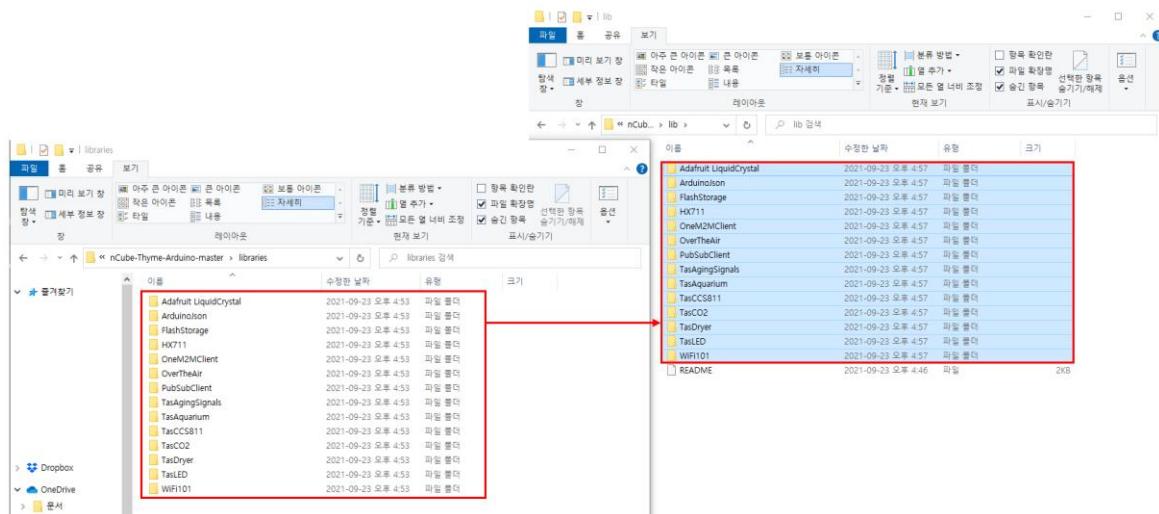
The status bar at the bottom indicates the file is 'Default (nCube-Air)' with line 1, column 1, spaces: 2, and encoding: UTF-8.

프로젝트가 정상적으로 생성되면 위의 그림과 같이 프로젝트 폴더가 열림을 알 수 있으며, 아두이노를 동작시키는 main.cpp 파일이 src 폴더 아래에 있음을 확인할 수 있다.

2 - 4 nCube-Thyme-Arduino 다운로드 및 import

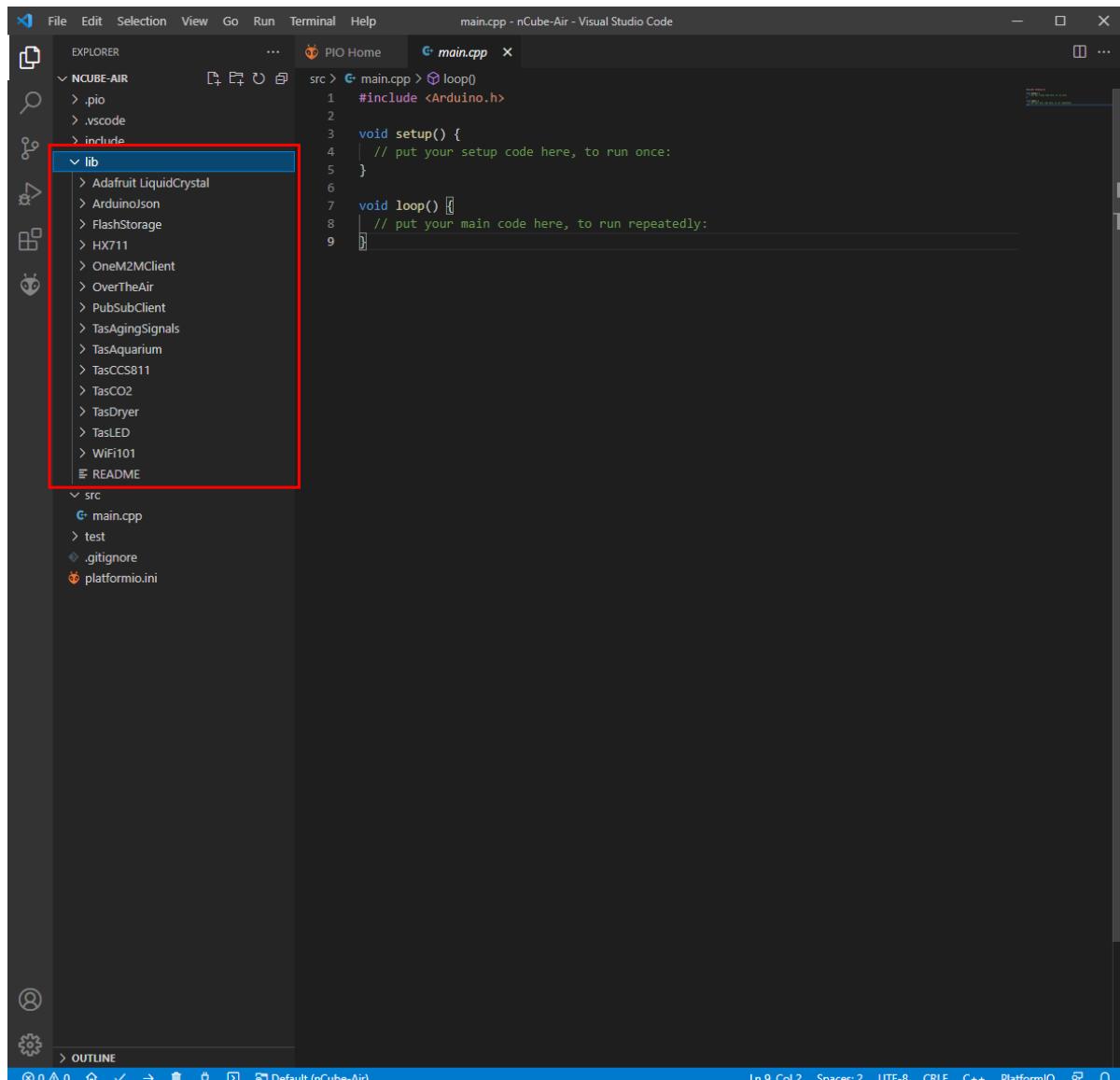
The screenshot shows the GitHub repository page for 'nCube-Thyme-Arduino'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below that, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. In the center, there's a list of files: 'IoTKETI Add files via upload', 'libraries', '.gitignore', and 'README.md'. On the right, there's a sidebar with options like 'Clone' (via HTTPS, SSH, or GitHub CLI), 'Open with GitHub Desktop', 'Open with Visual Studio', and a prominent 'Download ZIP' button, which is also highlighted with a red box.

<https://github.com/IoTKETI/nCube-Thyme-Arduino> 링크에서 nCube-Thyme-Arduino 라이브러리들을 다운로드 받고, 압축을 풀어준다.



nCube-Thyme-Arduino-master/libraries 폴더의 라이브러리들을 생성한 프로젝트 폴더의 lib 폴더로 복사한다

nCube-Thyme-Arduino 가이드



nCube-Thyme-Arduino의 라이브러리들을 생성한 프로젝트 폴더의 lib 폴더로 복사된 것을 확인할 수 있다.

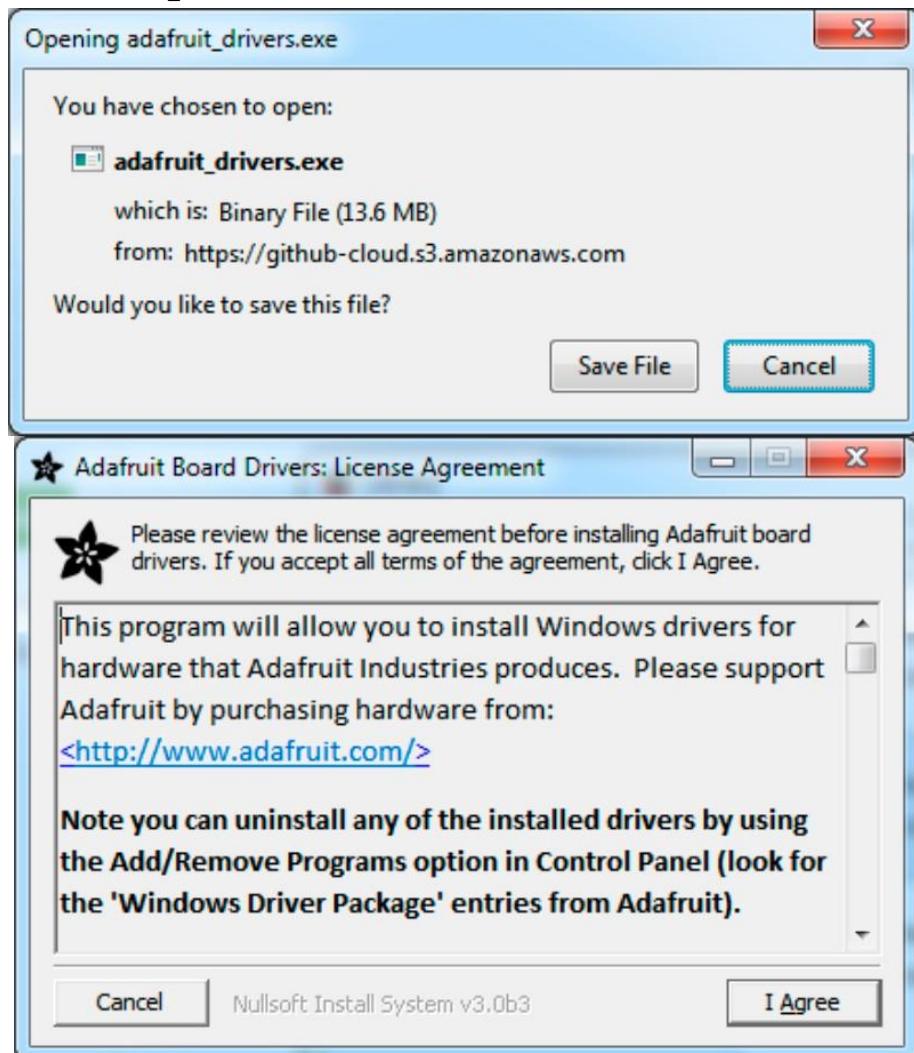
이로서 nCube-Thyme-Arduino를 개발하기 위한 환경설정을 완료하였다.

2 - 5 Adafruit Windows Driver 설치

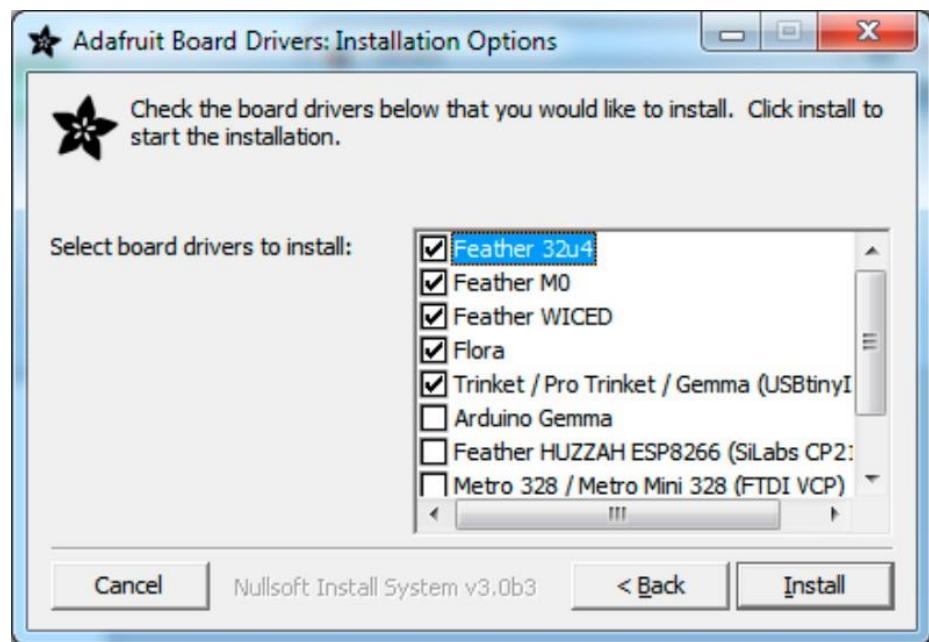
윈도우 운영체제 환경에서는 추가적으로 Adafruit WindowsDriver 를 설치해야한다.

<http://adafru.it/mai>

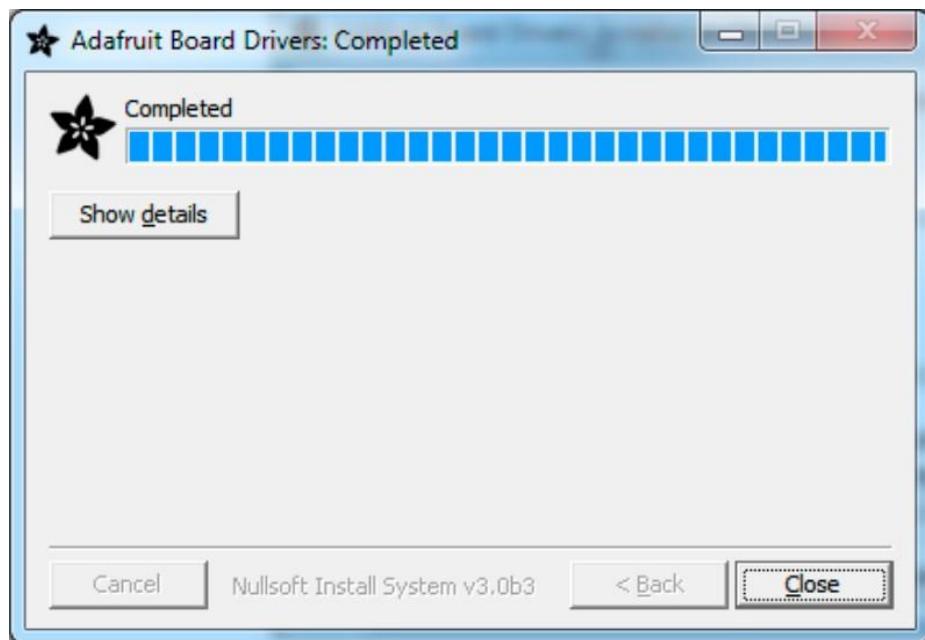
위의 URL 을 통해 adafruit_drivers.exe 파일을 다운로드 받고 실행시킨다.



installer 를 실행시키고, 라이선스 이용 내용을 확인한 뒤에 I Agree 버튼을 눌러 설치를 진행한다.



위의 화면에서 드라이버의 설치를 원하는 보드의 종류를 체크할 수 있다. 본 문서의 예제를 위해 Feather M0 항목에 체크한 뒤 Install 버튼을 눌러준다.



윈도우 운영체제를 위한 adafruit drivers 설치가 완료되었다.

3 보드 및 센서 환경설정

이번 장에서는 nCube-Thyme-Arduino를 구성하기 위해 필수적으로 요구되는 라이브러리를 다운로드받고, OneM2MClient 라이브러리의 예제파일로 제공되는 nCube-Air를 이용하여 보드를 구성한다.

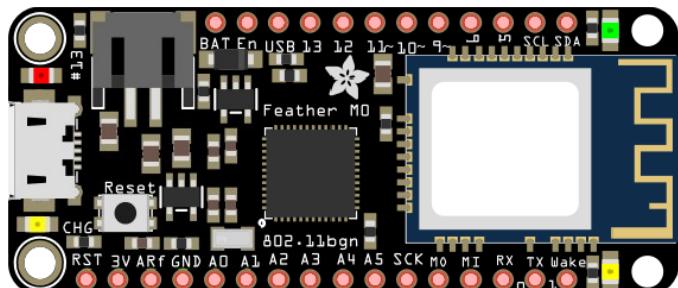
3 - 1 보드 및 센서 준비



- Adafruit Feather M0 WiFi 1개
- Adafruit CCS811 Air Quality Sensor 1개
- RGB 3Color LED 1개
- Bread Board 1개
- Jumper Cable M/M 한 세트
- USB Type-A to Micro-B Cable 1개

3 - 2 보드와 센서 개요

Adafruit Feather M0



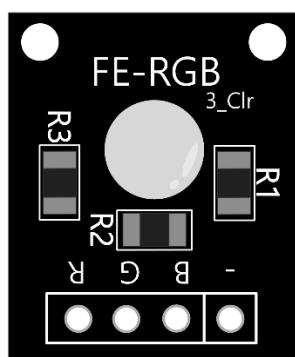
- Cortex M0 processor
- SPI, I2C, UART 통신지원
- 802.11bgn을 지원하는 Atmel WINC1500 WiFi chip

CCS811 Air Quality Sensor



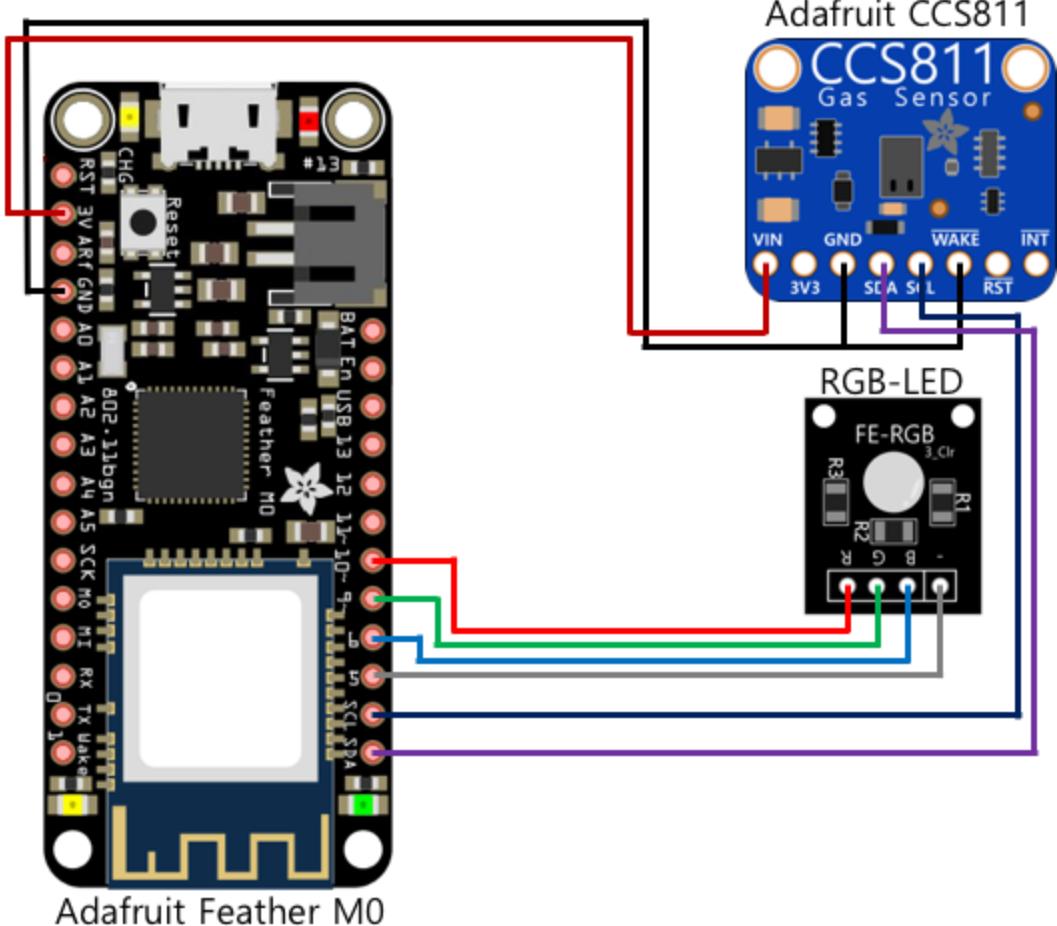
- 대기중의 CO₂, TVOC(TOTAL Volatile Organic Compounds) 측정
- I2C 통신을 이용하여 Arduino에 센서 값을 전송

RGB-LED



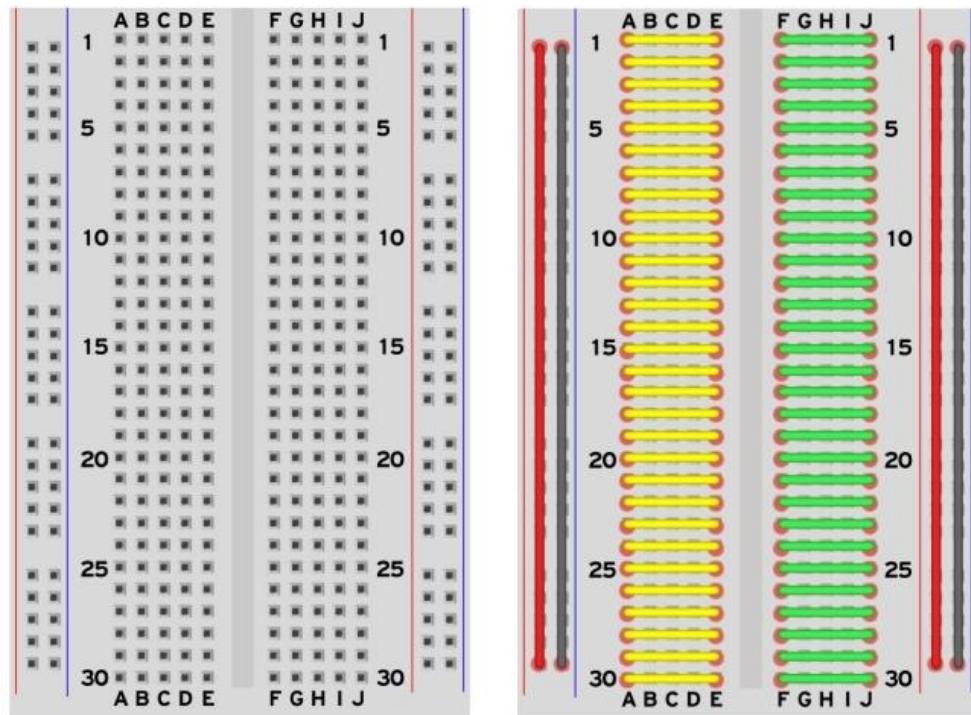
- RGB 3-color LED 보드는 빨간색, 녹색, 파란색 LED를 포함
- RGB 3-color LED의 LED들은 각각의 R, G, B핀의 전원 입력에 따라 점멸

3 - 3 센서와 보드 연결



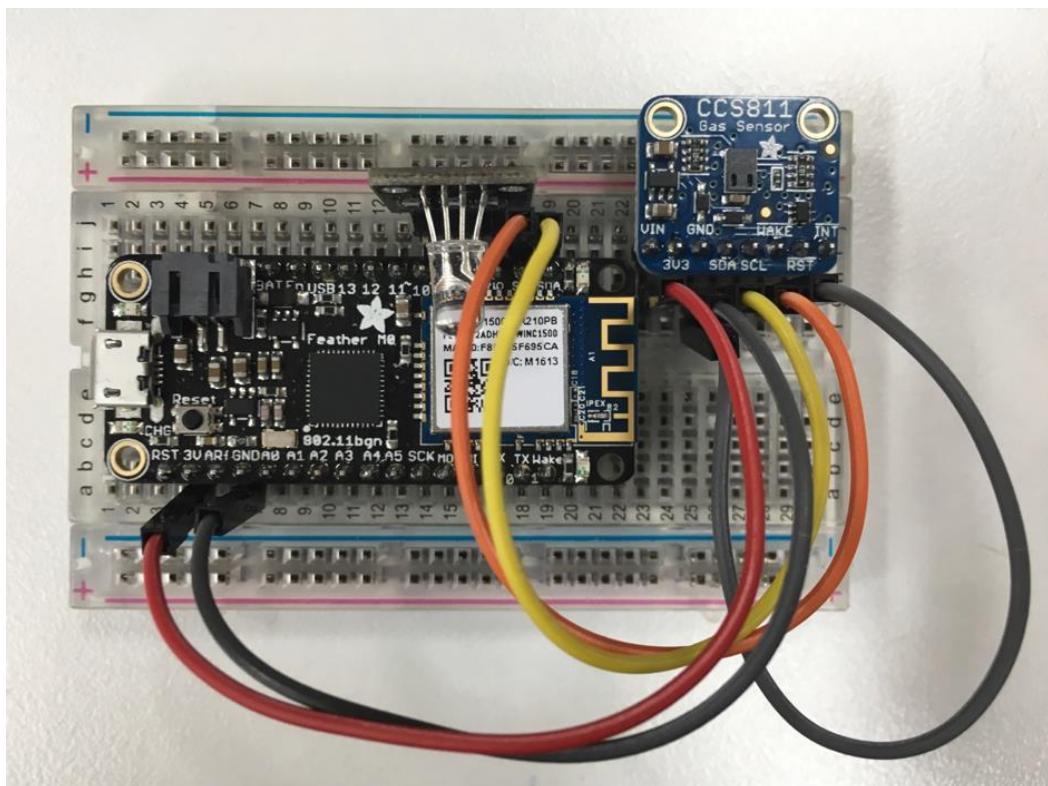
위의 그림과 같이 Adafruit Feather M0 보드와 CCS811 Air Quality Sensor, RGB-LED를 연결한다.

nCube-Thyme-Arduino 가이드



이 때 사용하는 Bread board는, 위와 같은 구조를 가지고 있다.

가장 오른쪽의 2줄과 왼쪽의 2줄은 각각 세로로 연결되어 있으며, a~e열은 각각 1~30행까지 가로로 연결되어 있고, f~j열 또한 각각 1~30행까지 가로로 연결되어 있다.

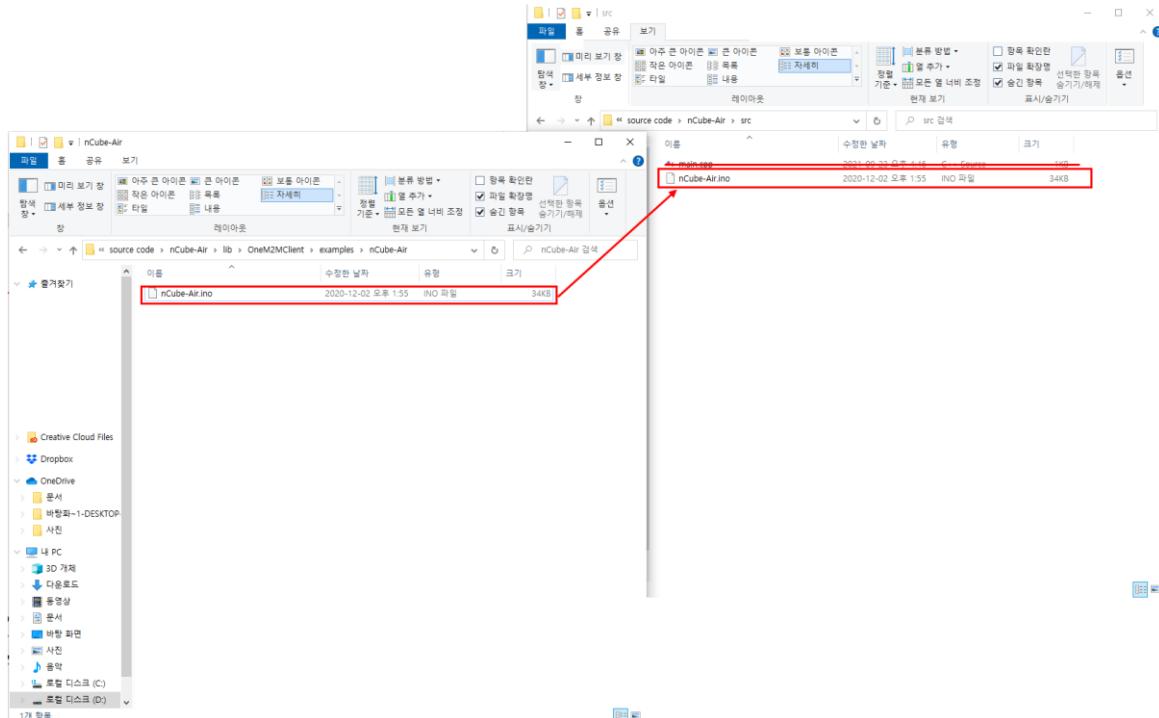


Bread board의 구조를 고려하여, Adafruit Feather M0 보드와 CCS811 Air Quality Sensor, RGB-LED를 연결한 모습은 위와 같다.

4 nCube-Air 설정 및 실행

이번 장에서는 테스트를 위해 생성한 nCube-Air 프로젝트를 설정하고 해당 코드를 실행해보는 과정을 설명한다.

4 - 1 nCube-Air 소스코드 설정



lib/OneM2MClient/examples/nCube-Air 디렉토리에 있는 nCube-Air.ino 소스 코드를 src 디렉토리에 복사하고, 기존 src 디렉토리에 main.cpp 파일을 삭제한다.

nCube-Thyme-Arduino 가이드

The screenshot shows the Visual Studio Code interface with the following details:

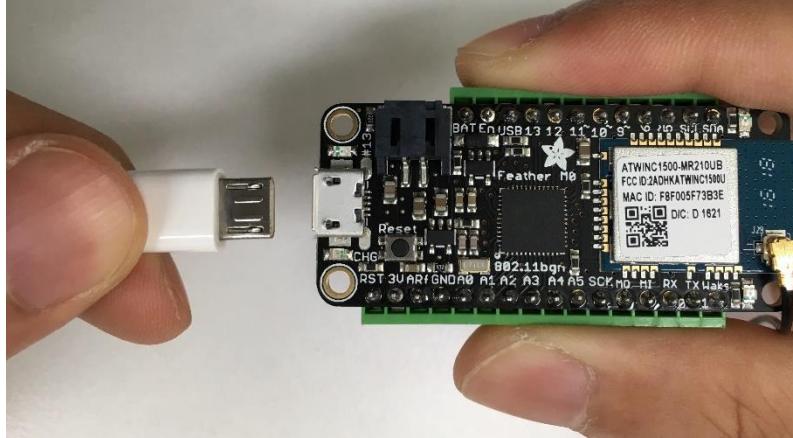
- File Explorer (Left):** Shows the project structure for "ncube-AIR". The "src" folder contains the main file "nCube-Air.ino". Other files like ".pio", "vscode", "c.cpp_properties.json", "extensions.json", "launch.json", "include", "lib", and various sensor and utility files are also listed.
- Code Editor (Center):** Displays the content of "nCube-Air.ino". A red box highlights the configuration section:

```
128 // Information of CSE as Mobius with MQTT
129 const String FIRMWARE_VERSION = "1.0.0.0";
130 String AE_NAME = "air1";
131 String AE_ID = "S" + AE_NAME;
132 const String CSE_ID = "/Mobius2";
133 const String CB_NAME = "Mobius";
134 const char* MOBIUS_MOTT_BROKER_IP = "203.253.128.161";
135 const uint16_t MOBIUS_MOTT_BROKER_PORT = 1883;
```
- Terminal (Bottom Left):** Shows the command "Default (nCube-Air)".
- Status Bar (Bottom Right):** Shows the line number (Ln 126), column (Col 55), and other file-related information.

nCube-Air.ino 소스 코드에서 120~127 라인 중 122 라인은 AE NAME을 설정하는 부분이다.

이 부분을 자신만의 AE 이름으로 바꾸어 준다.

4 – 2 nCube-Air 소스코드 업로드 및 실행



USB(micro 5pin)케이블을 사용하여 Adafruit Feather M0보드와 PC를 연결한다.

```

src > nCube-Air.ino > nCube
114     unsigned long temp_generate_interval = base_generate_interval;
115     unsigned long tvoc_generate_previousMillis = 0;
116     unsigned long tvoc_generate_interval = base_generate_interval;
117     unsigned long co2_generate_previousMillis = 0;
118     unsigned long co2_generate_interval = base_generate_interval;
119
120     // Information of CSE as Mobius with MQTT
121     const String FIRMWARE_VERSION = "1.0.0.0";
122     String AE_NAME = "airi";
123     String AE_ID = "S" + AE_NAME;
124     const String CSE_ID = "/Mobius2";
125     const String CB_NAME = "Mobius";
126     const char* MOBIUS_MOTT_BROKER_IP = "203.253.128.161";
127     const uint16_t MOBIUS_MOTT_BROKER_PORT = 1883;
128
129     OneM2MClient nCube;
130
131     // add TAS(Thing Adaptation Layer) for Sensor
132     #include "TasLED.h"
133     TasLED tasLed;
134
135     #include "TasCCS811.h"
136     TasCCS811 TasCCSSensor;
137
138
139     // build tree of resource of oneM2M
140     void buildResource() {
141         nCube.configResource(2, "/" +CB_NAME+, AE_NAME); // AE resource
142
143         nCube.configResource(3, "/" +CB_NAME+ "/" +AE_NAME, "update"); // Container resource
144         nCube.configResource(3, "/" +CB_NAME+ "/" +AE_NAME, "co2"); // Container resource
145         nCube.configResource(3, "/" +CB_NAME+ "/" +AE_NAME, "led"); // Container resource
146         nCube.configResource(3, "/" +CB_NAME+ "/" +AE_NAME, "temp"); // Container resource
147         nCube.configResource(3, "/" +CB_NAME+ "/" +AE_NAME, "tvoc"); // Container resource
148
149         nCube.configResource(23, "/" +CB_NAME+ "/" +AE_NAME+ "/update", "sub"); // Subscription resource
150         nCube.configResource(23, "/" +CB_NAME+ "/" +AE_NAME+ "/led", "sub"); // Subscription resource
151     }
152

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

[=====] 83% (1024/1228 pages)
[=====] 83% (1088/1228 pages)
[=====] 94% (1152/1228 pages)
[=====] 99% (1216/1228 pages)
[=====] 100% (1228/1228 pages)
done in 0.598 seconds

Verify 78040 bytes of flash with checksum.
Verify successful
done in 0.053 seconds
CPU reset.

===== [SUCCESS] Took 17.44 seconds =====

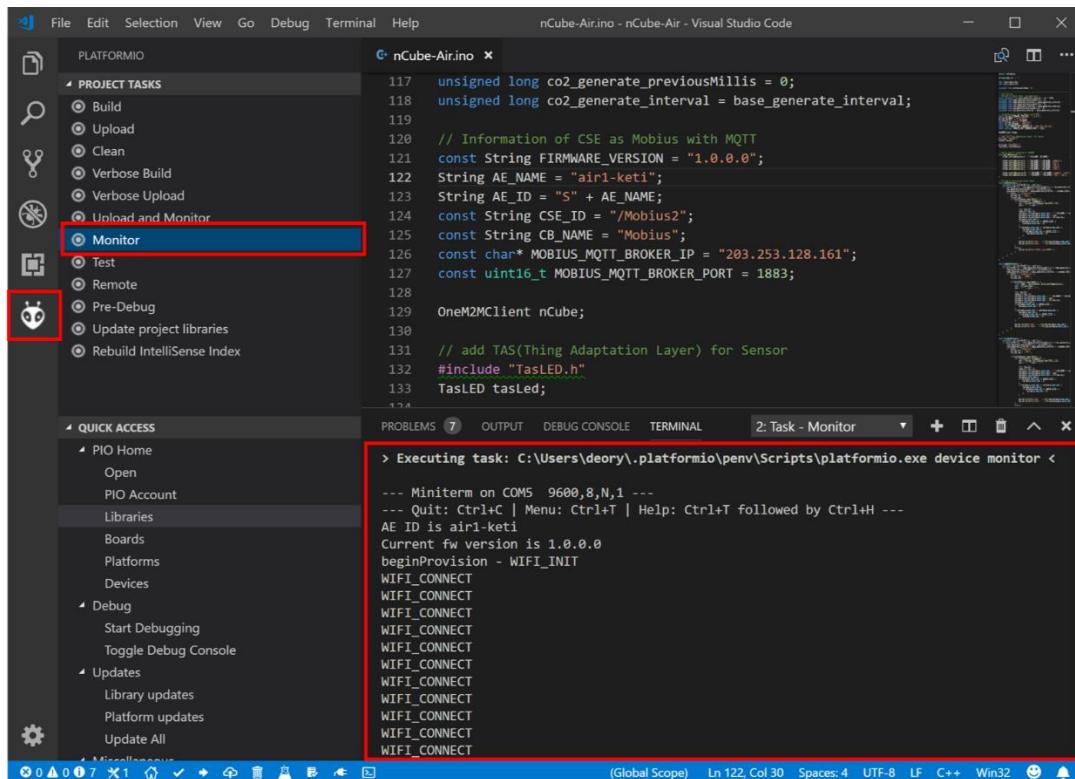
Terminal will be reused by tasks, press any key to close it.

VS Code의 왼쪽 사이드바에서 PlatformIO IDE 버튼을 누르고, PROJECT TASKS탭의 Upload 버튼을 클릭한다.

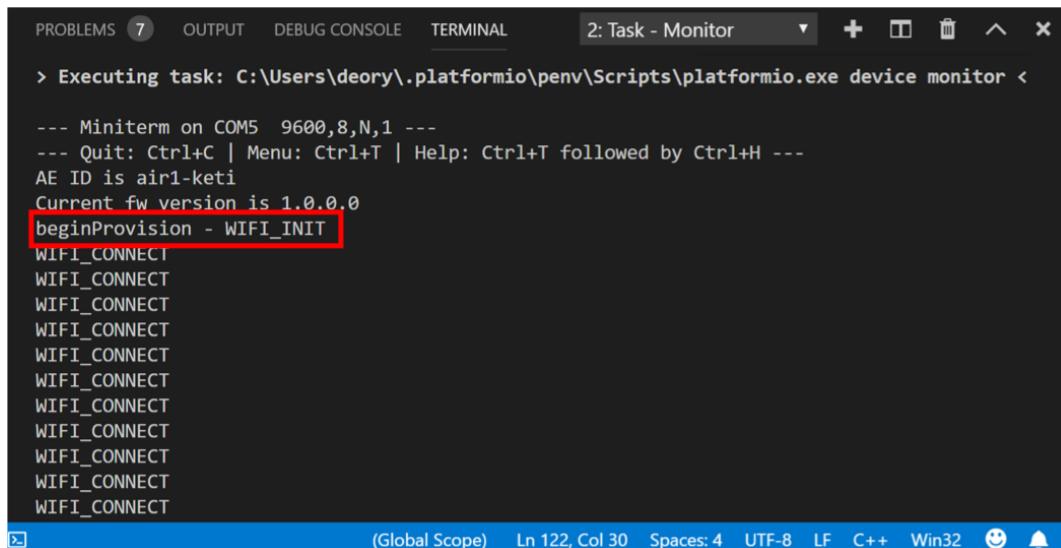
Upload 버튼을 누르면 PlatformIO IDE는 소스 코드를 컴파일하고 연결된 보드로 업로드 한다.

업로드가 모두 완료되고 난 뒤에 상태 창에 SUCCESS 메시지가 나타난다.

nCube-Thyme-Arduino 가이드



VS Code의 왼쪽 사이드바에서 PlatformIO IDE 버튼을 누르고, PROJECT TASKS 탭의 Monitor 버튼을 클릭한다.

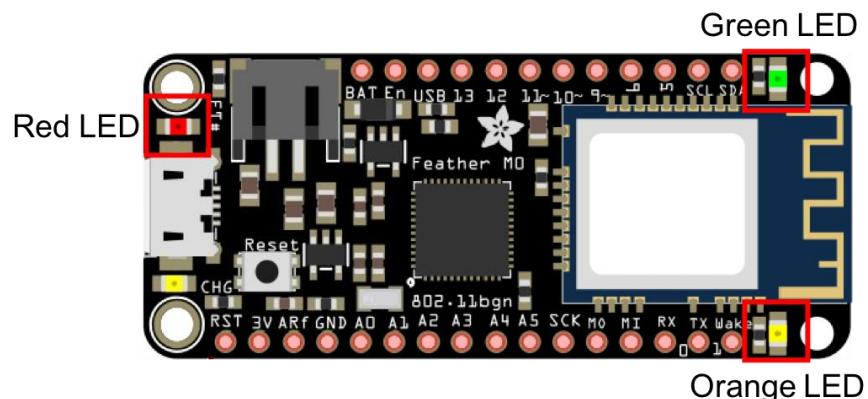


시리얼 모니터를 보면, 현재 nCube-Air가 Provision 모드임을 확인할 수 있다.

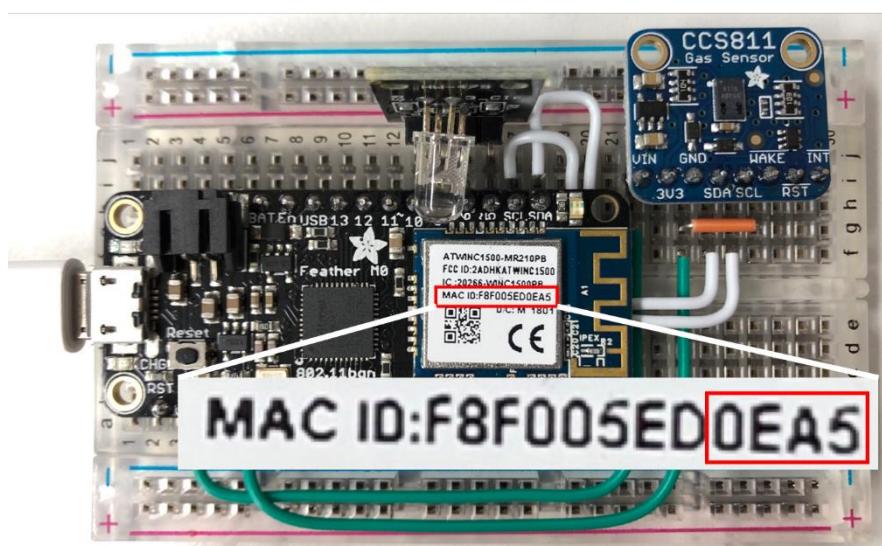
nCube-Air는 WiFi 연결을 통해서 MQTT 프로토콜로 Mobius와 통신해야 한다. 그러나, 소스 코드를 업로드 한 직후의 nCube-Air는 사용할 수 있는 WiFi AP를 알 수 없다.

따라서 nCube-Air가 스스로 WiFi AP로 동작하면서 웹서버를 구동함으로써, 사용자가 nCube-Air가 이용할 수 있는 WiFi AP 정보를 입력할 수 있도록 동작하는것이 Provision 모드이다. Provision 모드에 있는 nCube-Air에 WiFi 로 접속하여, nCube-Air의 웹 서버를 통해서 nCube-Air가 사용할 수 있는 WiFi AP 정보를 입력하면, nCube-Air가 WiFi AP 모드를 해제하고, 사용자가 입력한 WiFi 정보를 이용하여 WiFi AP에 접속할 수 있다.

nCube-Thyme-Arduino 가이드



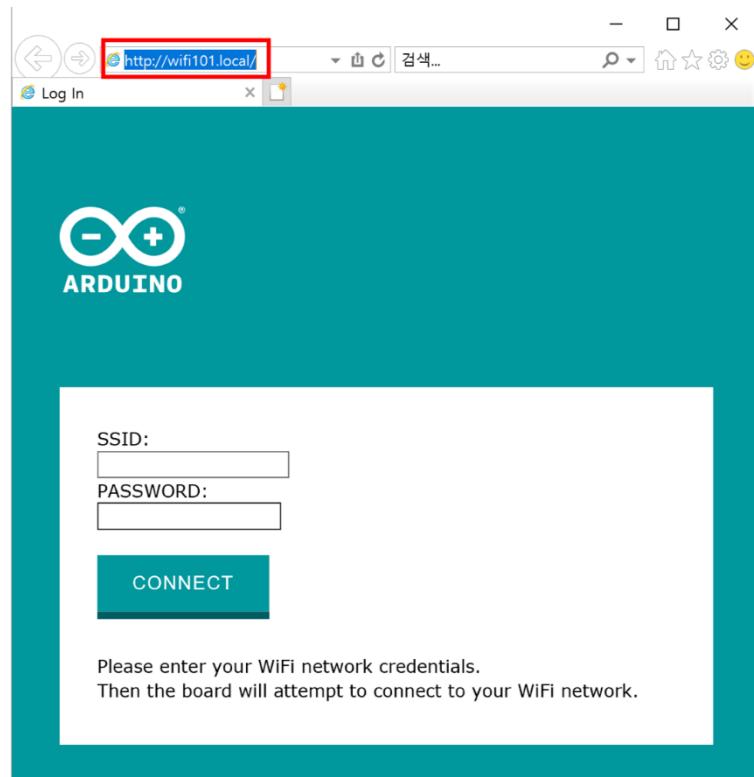
Provision 모드에서는 Red LED가 빠르게 점멸하고, Green LED가 점등된 상태가 유지된다.



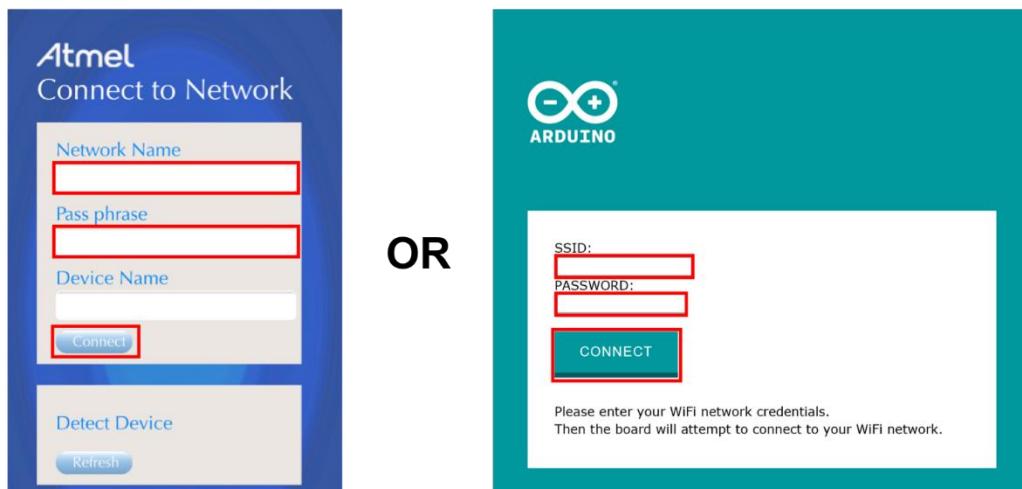
Adafruit Feather M0 보드의 WiFi 모듈 위에 붙어있는 스티커의 MAC 주소 마지막 네 자리의 문자를 기억한다.

그리고 랩탑 또는 스마트폰의 WiFi를 이용하여 [wifi101-\[MAC 주소 마지막 네 자리\].local](http://wifi101-[MAC 주소 마지막 네 자리].local)로 검색되는 WiFi AP에 연결한다.

nCube-Thyme-Arduino 가이드



WiFi AP에 연결한 뒤에, 웹 브라우저를 열어 <http://wifi101.local> URL로 접속하여 아두이노의 WiFi 설정 페이지에 접속한다.



WiFi 모듈의 펌웨어 버전에 따라 WiFi 설정 페이지의 모양에 차이가 날 수 있다.

Network Name 또는 SSID에는 nCube-Air가 이용 할 WiFi AP의 SSID를 입력하고, Pass phrase 또는 PASSWORD에는 AP의 패스워드를 입력한 후 connect 버튼을 클릭한다

nCube-Thyme-Arduino 가이드

```
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL

WIFI_CONNECT
WIFI_CONNECT
SSID: FI-LAB2G
IP Address: 192.168.0.87
signal strength (RSSI):-60 dBm
_MQTT_CONNECT
_MQTT_CONNECT
_MQTT_CONNECT
Attempting MQTT connection...connected
/oneM2M/resp/Sair1-keti/Mobius2/json Successfully subscribed
/oneM2M/req/Mobius2/Sair1-keti/json Successfully subscribed
create_ae - 0 - xI64VypS
Request [/oneM2M/req/Sair1-keti/Mobius2/json] ----> 136
{"op":"1","to":"/Mobius?rcn=0","fr":"Sair1-keti","rqi":"xI64VypS","ty":"2","pc":{"m2m:ae":
>{"rn":"air1-keti","api":"3.14","rr":"true"}}}
<---- 2001
create_cnt - 0 - GjrLodiS
Request [/oneM2M/req/Sair1-keti/Mobius2/json] ----> 119
{"op":"1","to":"/Mobius/air1-keti?rcn=0","fr":"Sair1-keti","rqi":"GjrLodiS","ty":"3","pc":
 {"m2m:cnt":{"rn":"update"}}}
<---- 2001
create_cnt - 1 - 4YNypNAt
Request [/oneM2M/req/Sair1-keti/Mobius2/json] ----> 116
{"op":"1","to":"/Mobius/air1-keti?rcn=0","fr":"Sair1-keti","rqi":"4YNypNAt","ty":"3","pc":
 {"m2m:cnt":{"rn":"co2"}}}
<---- 2001
create_cnt - 2 - LG1xx50z
Request [/oneM2M/req/Sair1-keti/Mobius2/json] ----> 116
```

입력한 WiFi AP에 nCube-Air가 연결되고 난 이후에, 시리얼 모니터를 통해서 접속한 WiFi AP의 이름과 nCube-Air가 할당받은 IP주소를 확인할 수 있다.

이후, nCube-Air는 Mobius에 리소스를 생성하고 센서의 데이터를 업로드하며, 시리얼 모니터를 통해서 이 과정을 확인할 수 있다.

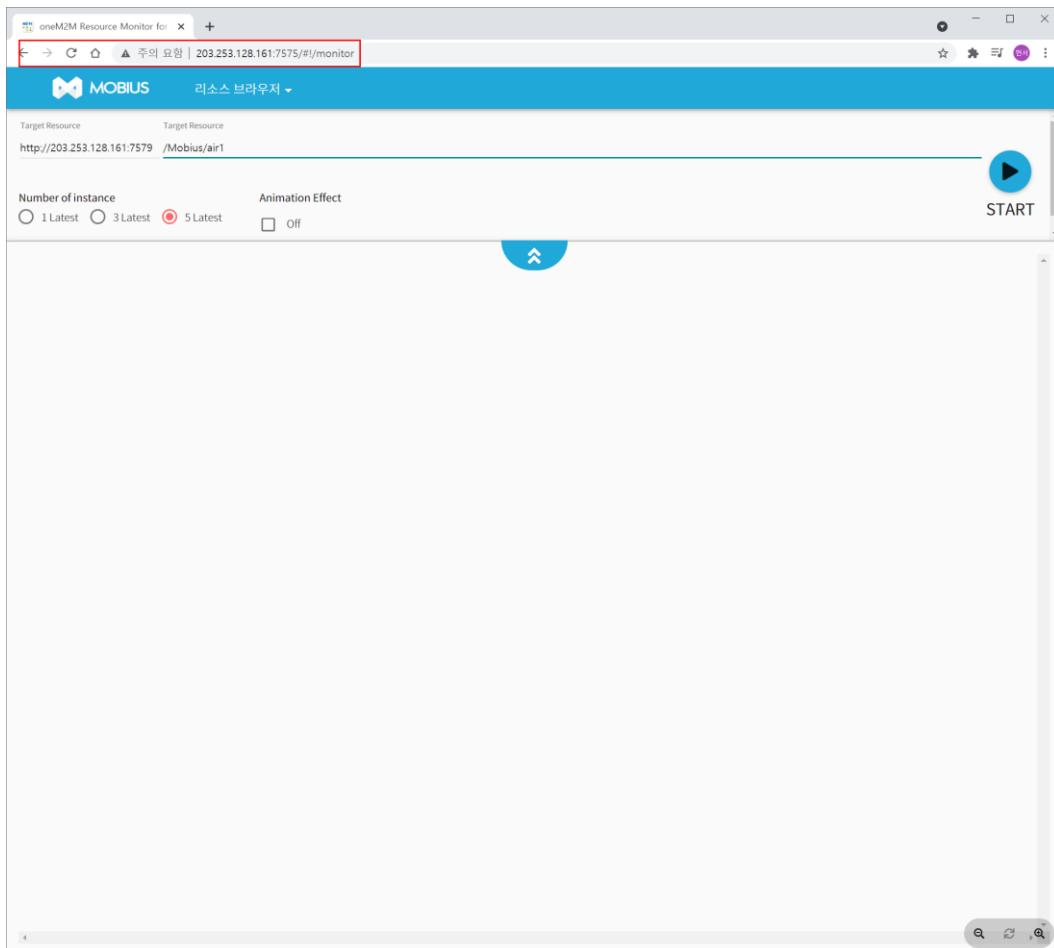
WiFi 연결이 된 이후에는 Red LED가 소등되고, Green LED가 점등되며, 데이터가 발생하여 WiFi를 통해서 Mobius로 업로드 할 때 Orange LED가 점멸한다.

이로서 nCube-Air의 소스 코드를 nCube-Air에 업로드 하고, 동작시키는 것이 완료되었다.

5 nCube-Air 데이터 검색 및 LED 제어

이번 장에서는 Mobius 리소스 브라우저를 통해 nCube-Air 프로젝트에서 생성한 데이터를 조회하고 LED를 제어하는 과정을 살펴본다.

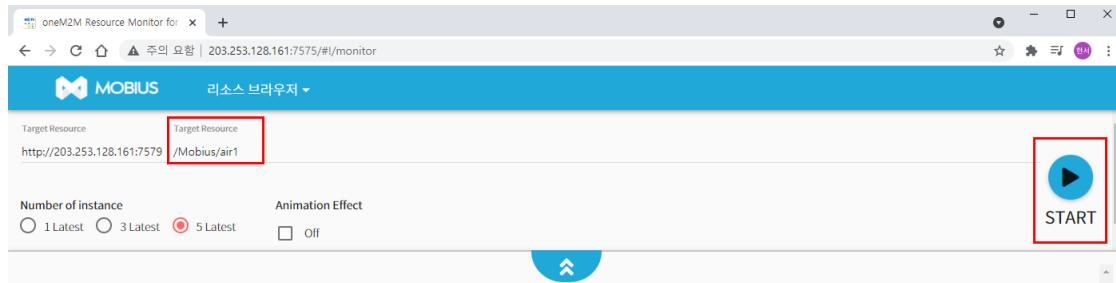
5 - 1 Mobius 리소스 모니터



oneM2M Resource Monitor for Mobius는 Mobius에 생성되는 리소스를 실시간으로 보여주고, 리소스의 생성과 삭제 기능을 지원한다.

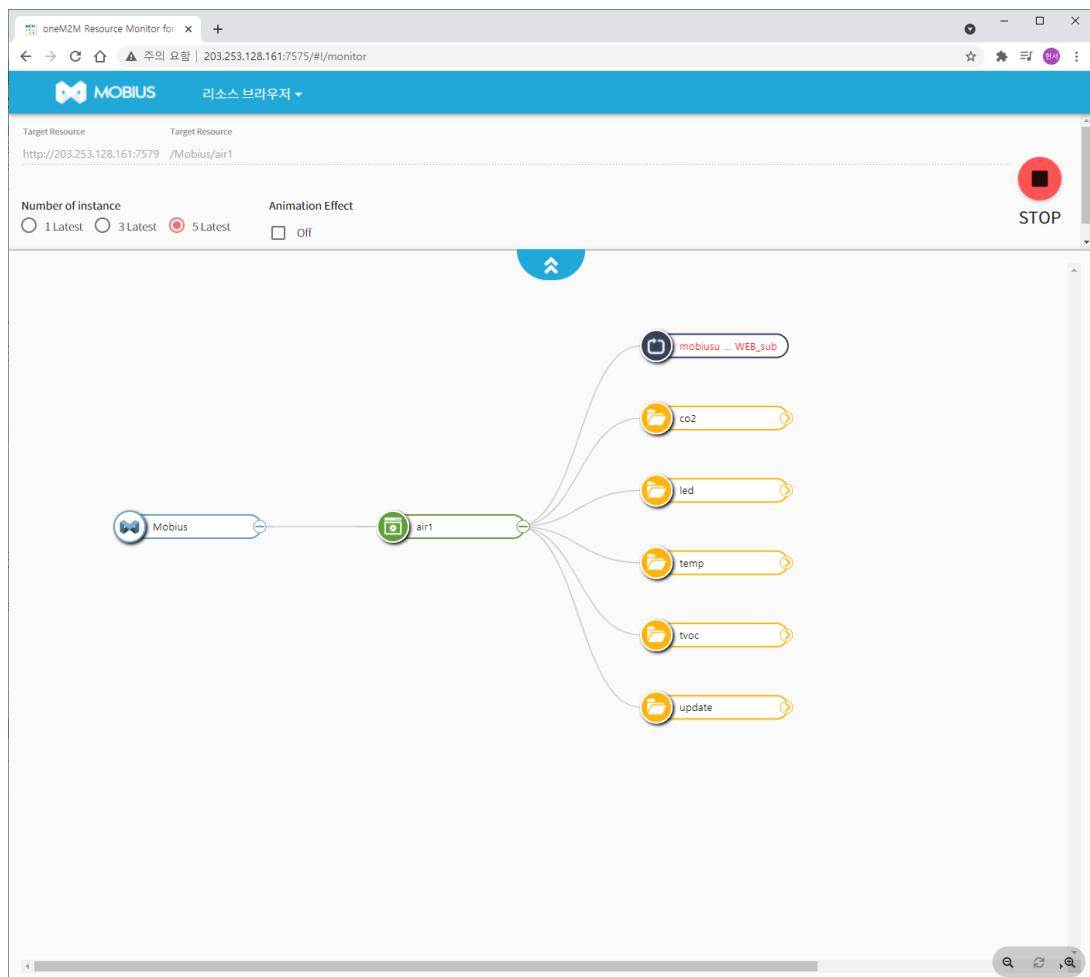
oneM2M Resource Monitor for Mobius는 <http://203.253.128.161:7575>에 접속하여 이용할 수 있다.

5 – 2 nCube-Air 데이터 조회



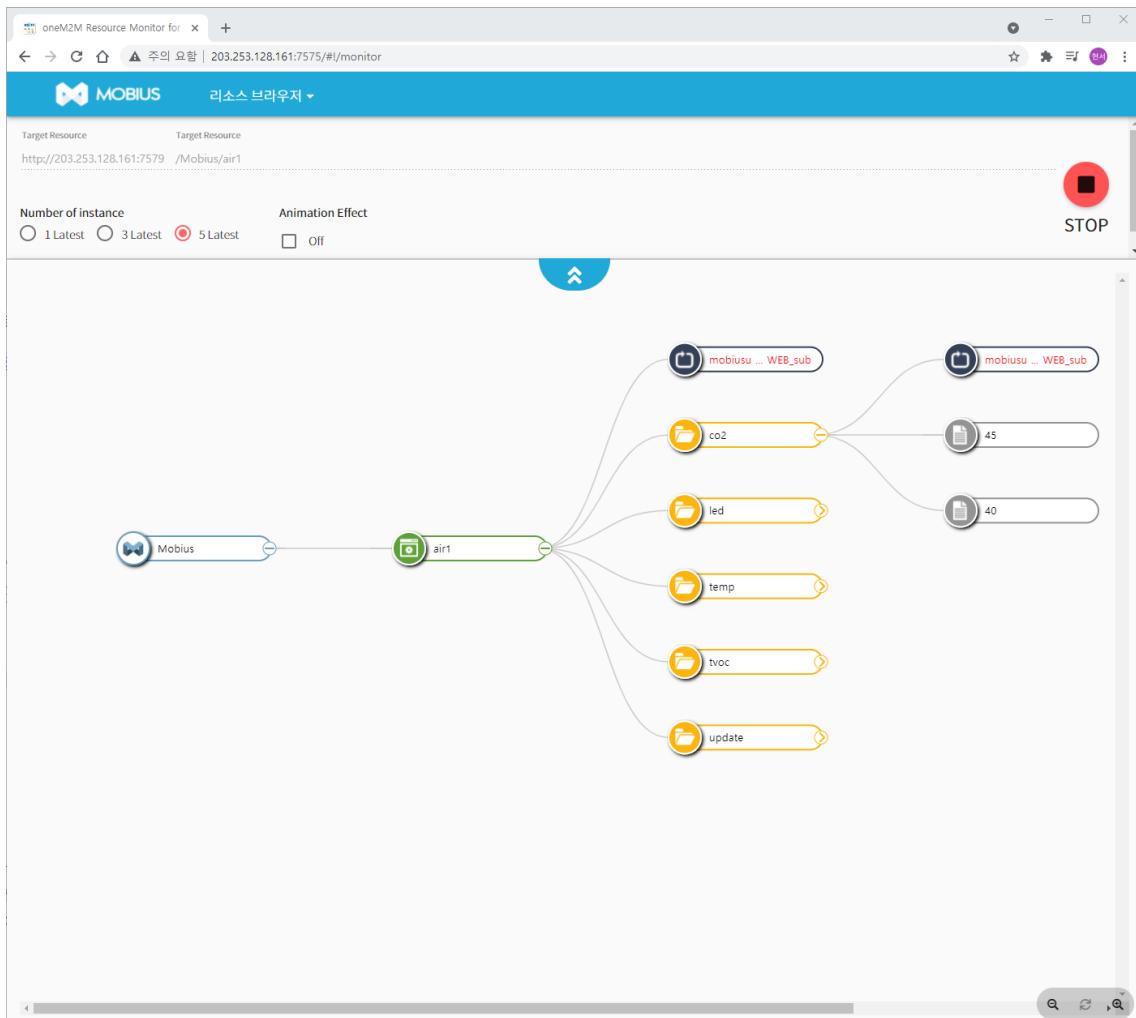
Target Resource에 nCube-Air 소스 코드에서 설정한 AE_NAME의 값(page 27 참고)을 입력하고 Start 버튼을 클릭한다.

Number of instance는 최근 몇 개의 데이터를 볼지 선택하는 항목이다. 5 Latest에 체크하면 최근 5개 데이터를 확인할 수 있다.



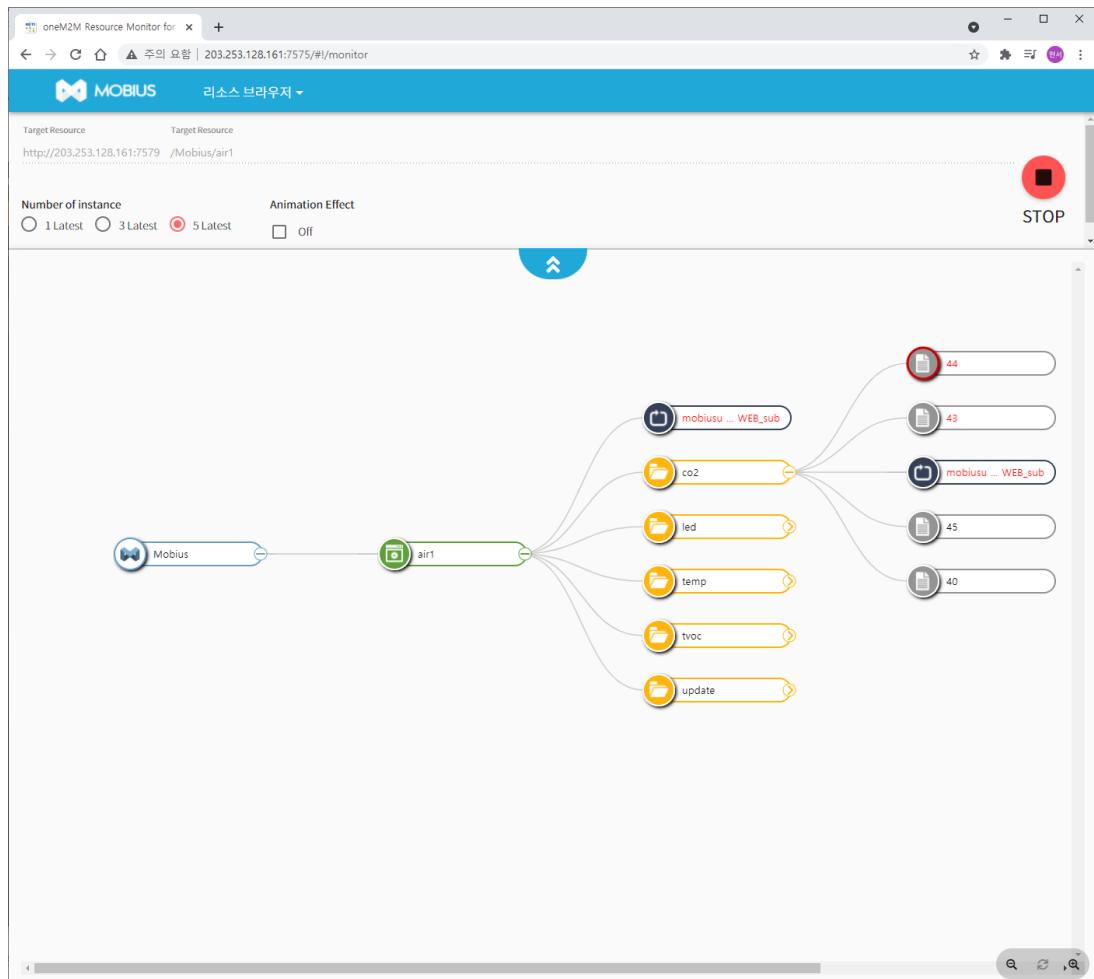
위의 그림과 같이 nCube-Air가 Mobius에 생성한 리소스 구조를 볼 수 있다.

nCube-Thyme-Arduino 가이드



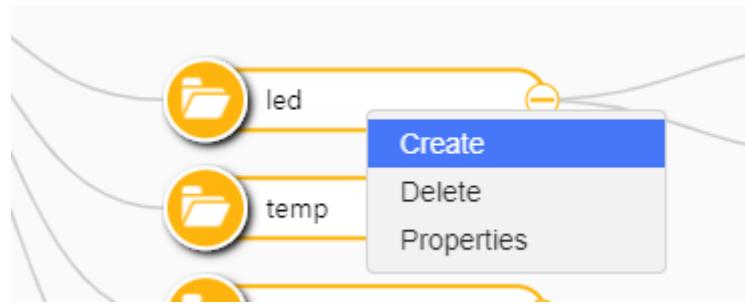
노란색으로 표시된 container(cnt) 리소스 아이콘 오른쪽의 + 버튼을 클릭하면 cnt 리소스 하위의 cin 리소스 데이터를 볼 수 있다.

nCube-Thyme-Arduino 가이드

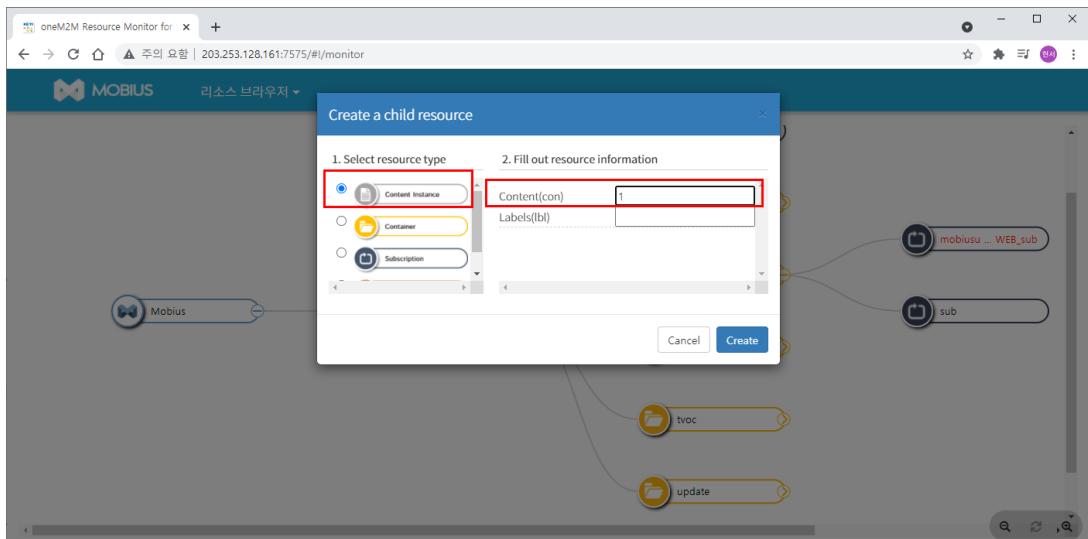


co2, 리소스에 생성된 데이터와 실시간으로 생성되는 데이터를 확인할 수 있다.

5 – 3 nCube-Air LED 제어



nCube-Air의 RGB-LED를 제어하기 위해서 led cnt 리소스 아래에 contentInstance(cin)리소스를 생성한다. led cnt 리소스 위에 마우스를 위치시키고, 우클릭 하여 드롭다운 메뉴를 띄운 뒤에 Create 버튼을 클릭한다.



Create a child resource 창에서 'Content Instance' 리소스를 선택하고, Content(con)의 값에 0~7 사이의 값을 입력하여 Create 버튼을 클릭한다.



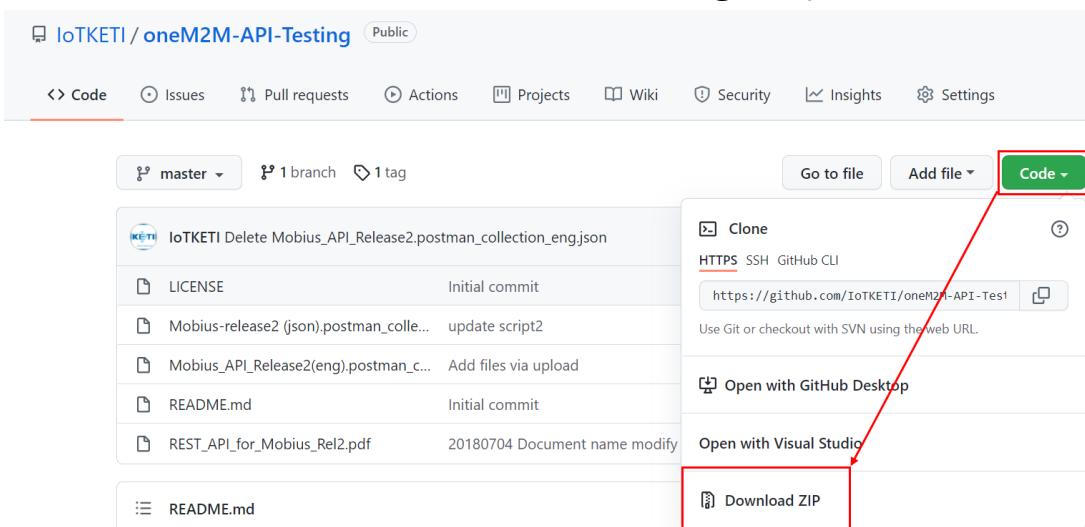
nCube-Air의 RGB-LED가 con의 값에 따라서 점멸한다.

이로서 nCube-Air가 측정하여 Mobius에 업로드한 공기 질 데이터를 조회하고, nCube-Air의 RGB-LED를 제어를 완료하였다.

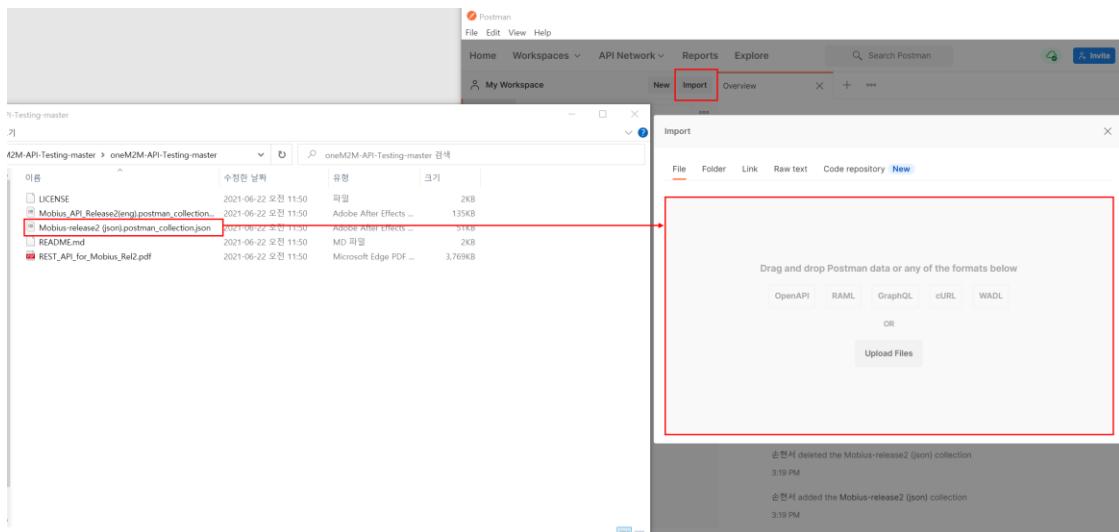
6 oneM2M API를 통한 nCube-Air 데이터 검색

oneM2M API를 이용하면, oneM2M Resource Monitor for Mobius를 이용하는 것 보다 더욱 다양한 조건을 이용하여 nCube-Air가 Mobius에 생성한 데이터를 조회하는것이 가능하다. 따라서 이번 장에서는 oneM2M API를 이용하여 nCube-Air 프로젝트에서 생성한 데이터를 조회하고 LED를 제어하는 과정을 살펴본다.

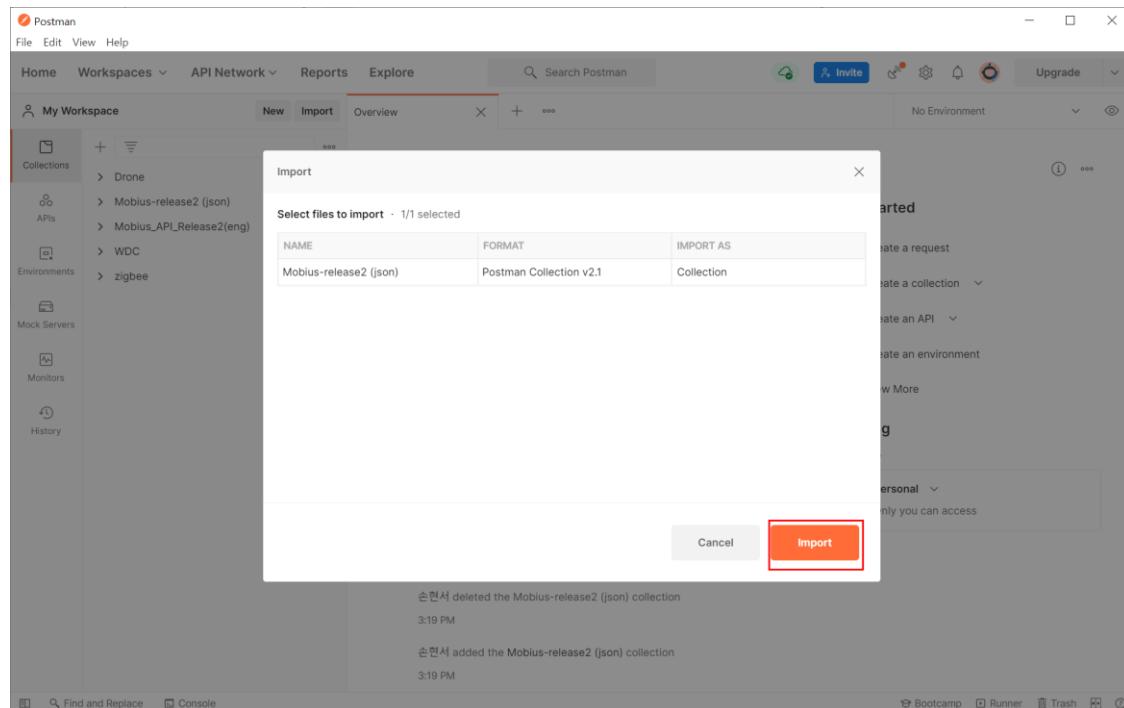
6 - 1 Postman에 oneM2M-API-Testing import하기



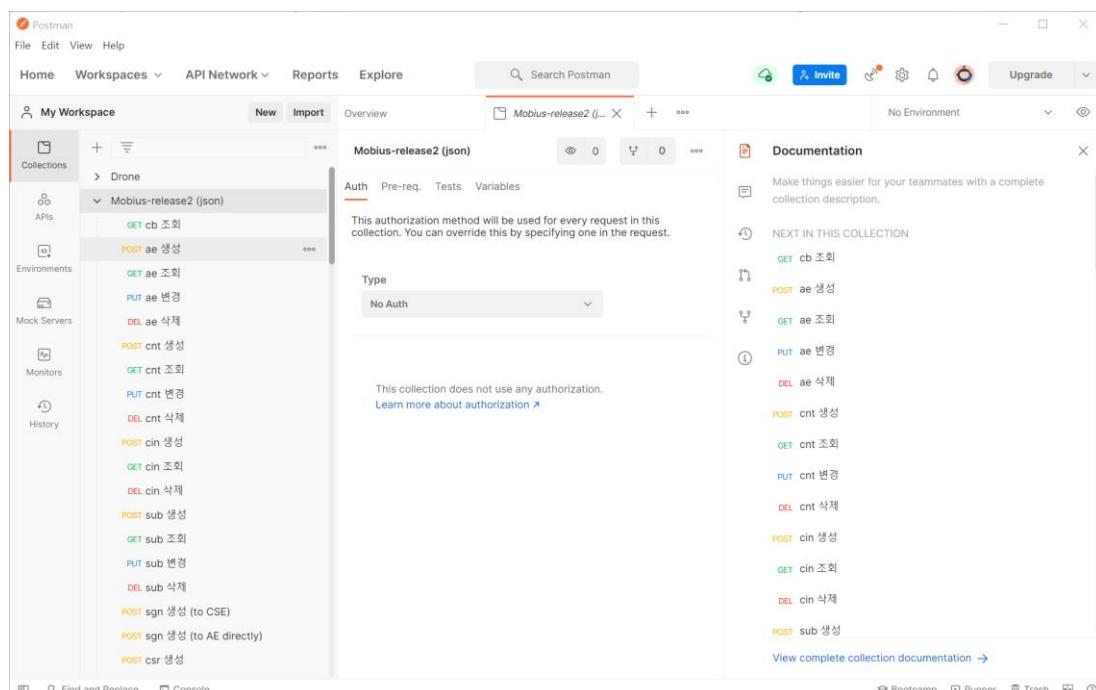
<https://github.com/IoTKETI/oneM2M-API-Testing>에 접속하여 API Collections을 다운로드 받고 압축을 해제한다.



nCube-Thyme-Arduino 가이드

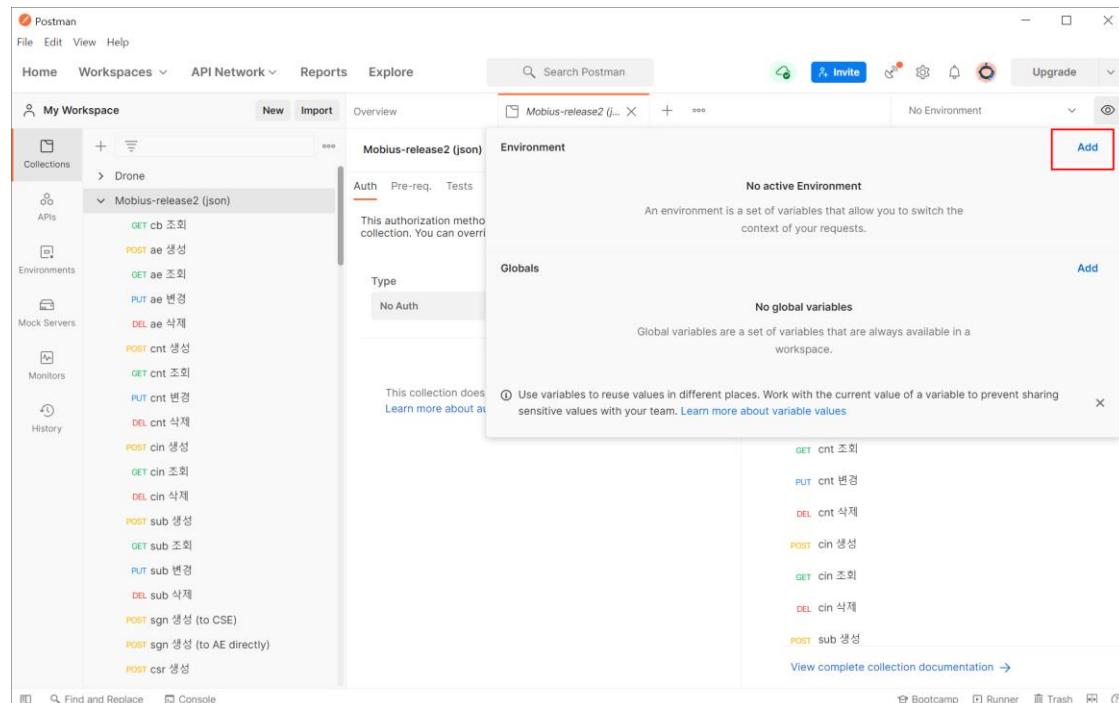


압축을 해제한 폴더에서 Mobius-release2 (json).postman_collection.json 파일을 postman app에 드래그하여 import 한다.

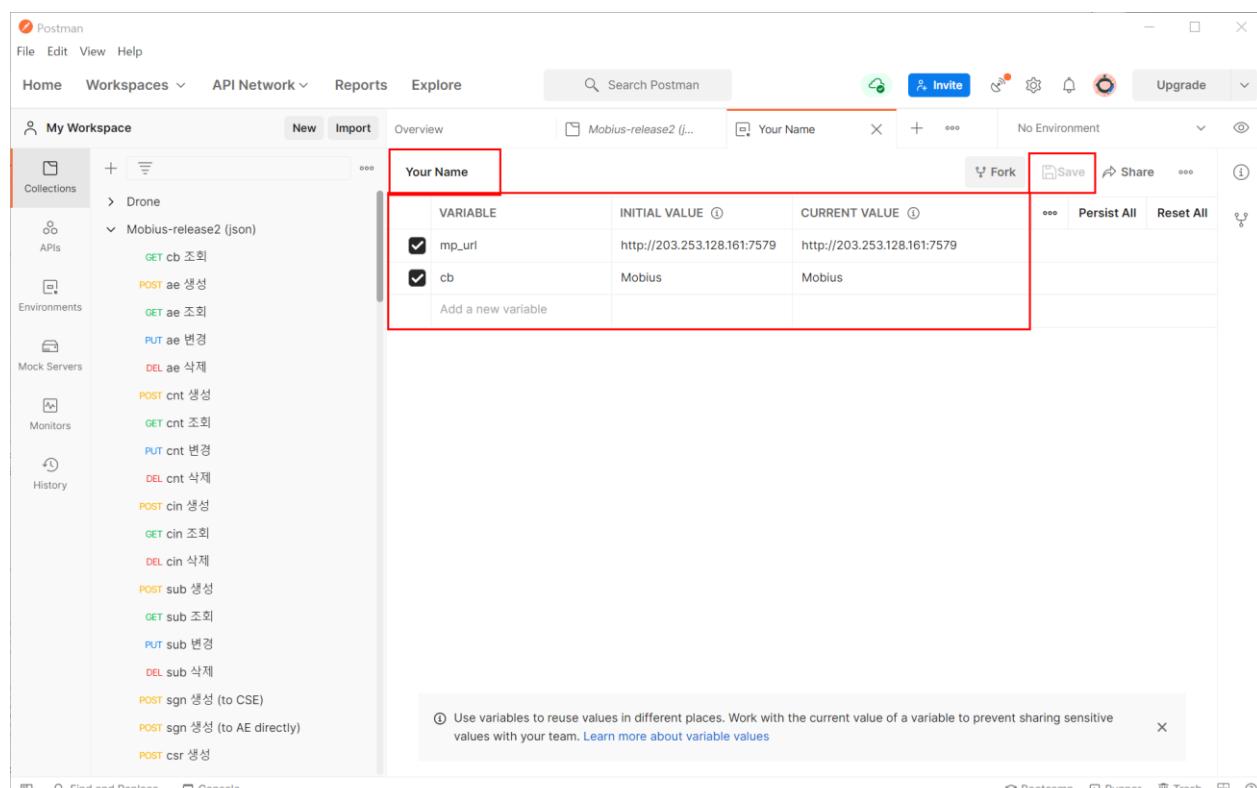


Postman의 메인 화면 왼쪽의 Collections탭을 클릭하면 Mobius release2 (json) API들이 성공적으로 추가된것을 확인할 수 있다.

6 – 2 Postman에 환경설정하기



Mobius release2 (json) API를 사용하기 위해서 추가적으로 환경설정을 해주어야한다. Postman의 우측 상단에 위치한 눈모양 아이콘을 클릭하고, Environments창의 Add 버튼을 클릭한다.



New Environment 창이 나타나면, 위와 같이 입력하고 **Save** 버튼을 누른다.

6 – 3 Cin 조회하기

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'Mobius-release2 (json)' which contains an item 'cin 조회'. The main workspace shows a GET request for 'cin 조회' with the path {{mp_url}}/{{cb}}/air1/co2/la. The 'Headers' tab is selected, showing three headers: Accept (application/json), X-M2M-Ri (12345), and X-M2M-Origin (SOrigin). The 'Body' tab is selected, showing a JSON response. The response body is a JSON object with several properties, including 'm2m:cin' which has a 'con' field highlighted with a red box containing the value '44'. At the bottom right of the main workspace, the 'Send' button is also highlighted with a red box.

Mobius release2 (json) 콜렉션에서 cin 조회 항목을 선택한 뒤에, 조회하고자 하는 cin 리소스의 path를 입력하고 Send 버튼을 클릭한다.

이 예제에서 path의 구성은 다음과 같다.

{{mp_url}}/{{cb}}/[AE Name]/[container Name]/la

{{mp_url}}과 {{cb}}는 위에서 설정한 환경의 값에 따라서 다음과 같이 변환되어 Postman이 보내는 API의 목적지가 된다.

[http://203.253.128.161:7579/Mobius/\[AE Name\]/\[container Name\]/la](http://203.253.128.161:7579/Mobius/[AE Name]/[container Name]/la)

path의 가장 마지막의 la는 latest의 oneM2M 표준상 short name이며, 상황에 따라 short name 혹은 full name을 사용할 수 있다.

API의 목적지 path를 container 하위의 latest로 설정함으로써, Mobius가 container 하위에 있는 가장 최신의 리소스를 반환하도록 하였다

이로서 oneM2M API와 Postman을 이용하여 nCube-Air가 업로드 한 최신 데이터를 조회하여 보았다.

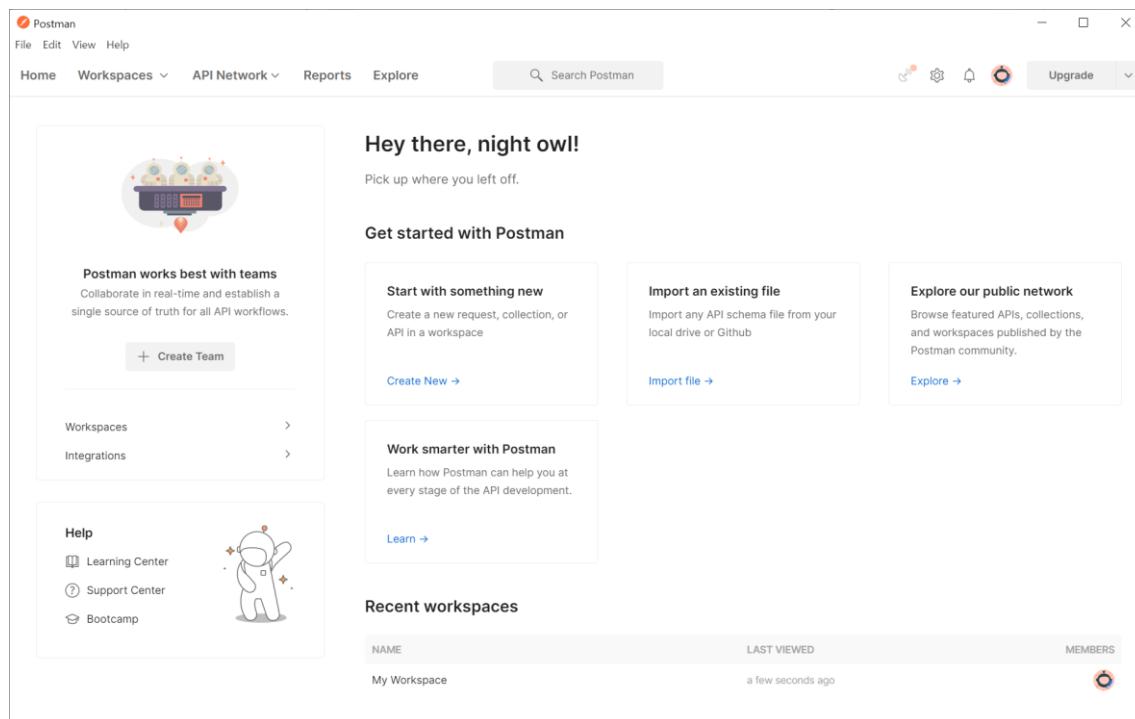
Appendix A.

Postman 설치

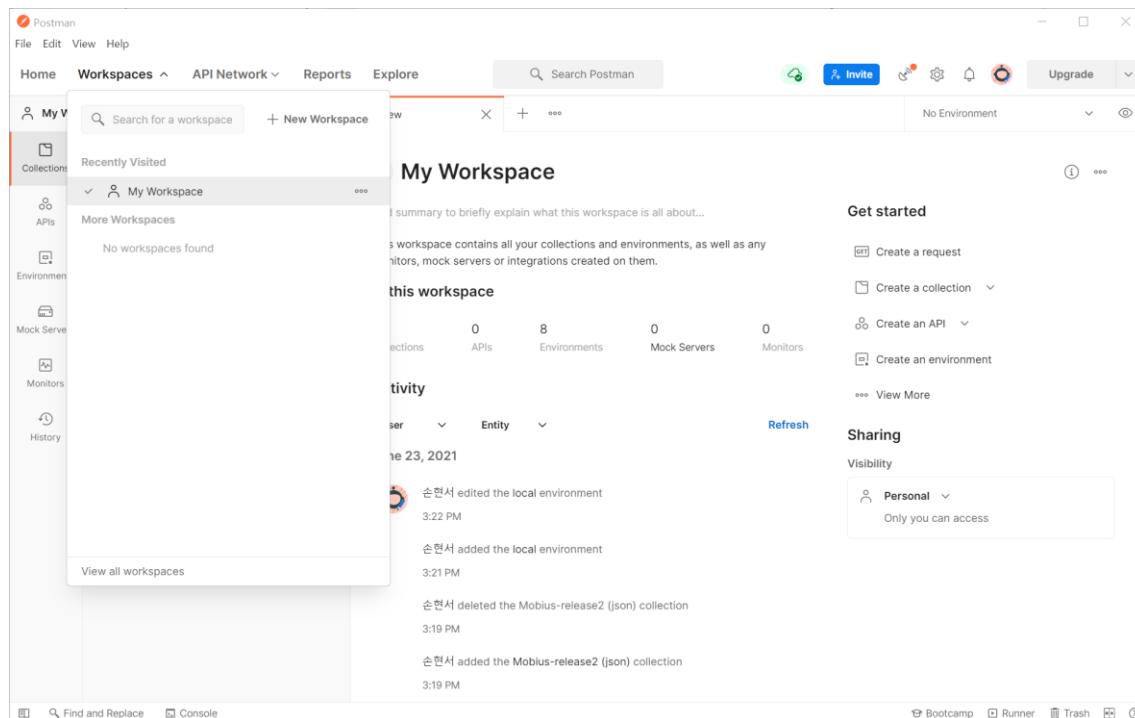
The screenshot shows the Postman homepage with the Twitter API v2 collection selected. The interface includes a sidebar with sections like Collections, Environments, Mock Servers, Monitors, and History. The main area displays the Twitter API v2 / Tweet Lookup / Single Tweet endpoint. The request method is GET, and the URL is https://api.twitter.com/2/tweets/:id. The request body contains the parameter 'id' with the value '140321829861628420'. The response status is 200 OK, with a duration of 468 ms and a size of 734 B. The response body is a JSON object with a single key 'data': { "id": "140321829861628420", "text": "Dioscore Mitchell went down after a collision with Paul George toward the end of Game 2. https://t.co/V9jVhob0Jh" }. The right side of the screen shows the Documentation for the endpoint, Request params, and Path Variables sections. At the bottom, there's a cartoon illustration of three robots holding monitors.

<https://www.postman.com/>에서 본인의 OS에 맞게 네모박스안의 버튼을 클릭하여 Postman 설치파일을 다운받는다.

nCube-Thyme-Arduino 가이드



다운로드한 설치파일을 실행하여 정상적으로 설치가 완료되면 위와 같이 postman app이 실행된다.



Workspaces에 API를 사용할 수 있는 환경이 구성되어 있다.