



OCEAN

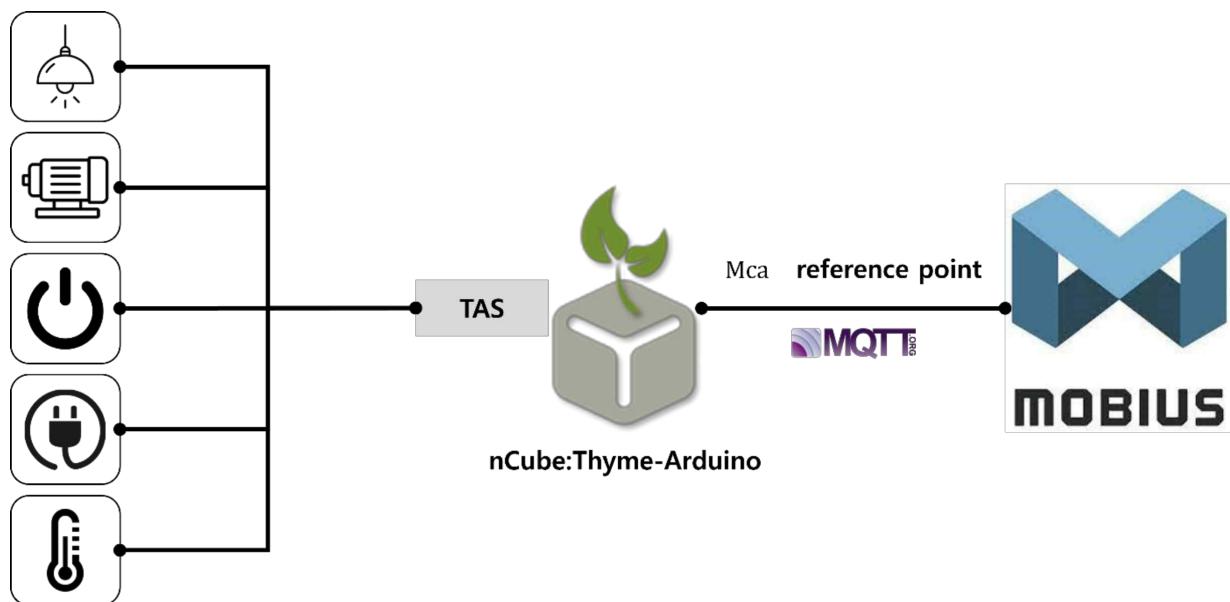
nCube-Thyme-Arduino

Ver.1.1.0

Revision History

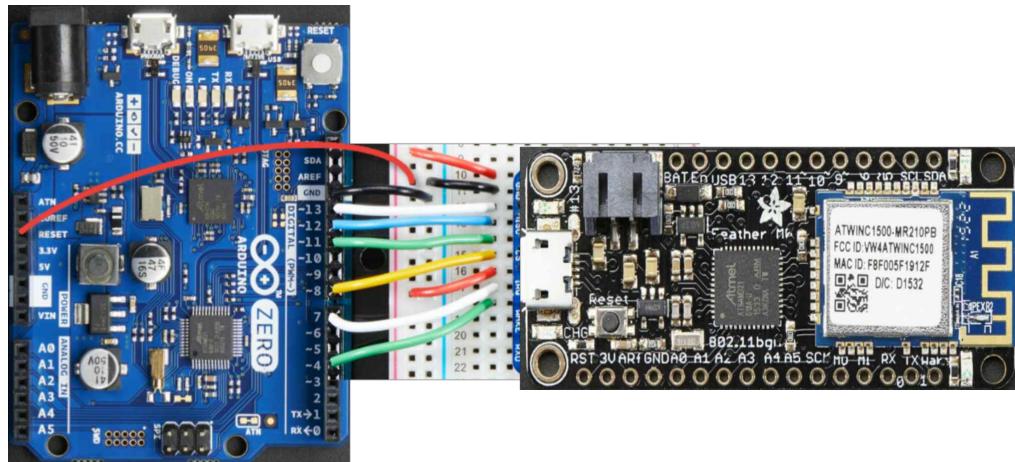
버전	변경일자	제·개정 내역
1.0.0		초안
1.1.0	2022-05-27	- 내용 변경에 따른 업데이트 eunbeelee@keti.re.kr / 이은비 monto8790@gmail.com / 박윤기 sm.jeong@keti.re.kr / 정승명 iyahn@keti.re.kr / 안일엽

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.....
1 - 1	Adafruit Feather M0.....
1 - 2	CO ₂ 센서.....
1 - 3	RGB-LED.....
2	환경설정.....
2 - 1	Arduino IDE 설치.....
2 - 2	Arduino IDE 환경설정.....
3	nCube-Air 개발 환경.....
3 - 1	nCube-Air 라이브러리 다운로드.....
3 - 2	nCube-Air 불러오기.....
4	보드 및 센서 환경 설정.....
4 - 1	보드 및 센서 준비.....
4 - 2	보드와 센서 개요.....
4 - 3	센서와 보드 연결.....

nCube-Thyme-Arduino 가이드

5 nCube-Air 설정 및 실행

5 - 1 nCube-Air 소스코드 설정

5 - 2 nCube-Air 소스코드 업로드 및 실행

5 - 3 네트워크 설정

5 - 4 nCube-Air 실행

6 nCube-Air 데이터 검색 및 LED 제어

6 - 1 Mobius 리소스 모니터

6 - 2 nCube-Air 데이터 조회

6 - 3 nCube-Air LED 제어

7 oneM2M API를 통한 nCube-Air 데이터 검색

7 - 1 Postman에 OneM2M-API-Testing import하기

7 - 2 Postman에 환경 설정하기

7 - 3 Cin 조회하기

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



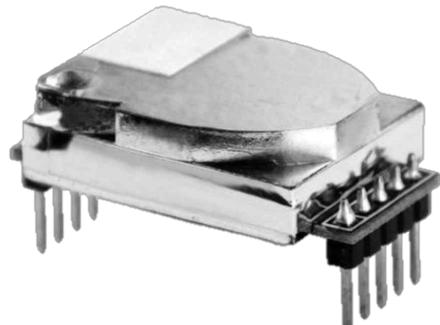
Adafruit 사에 의해 제작 및 판매되고 있다. Cortex M0 프로세서를 포함하고 있으며 I/O pin 들과, 여려개의 12-bit ADC(아날로그-디지털 변환회로), 하나의 10-bit DAC(디지털- 아날로그 변환회로), SPI, I2C, UART 통신을 지원하는 총 6 개의 SERCOM, native USB 지원, 802.11bgn을 지원하는 Atmel WINC1500 WiFi 모듈을 포함하는 등의 특징을 갖는다.

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.

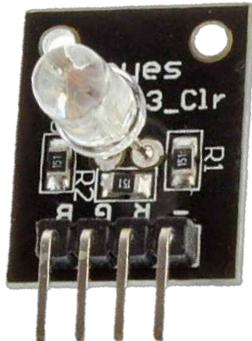
Micro USB를 통한 전원 입력과, 3.7V 리튬 폴리머 배터리를 이용한 전원 입력이 모두 가능하며, 배터리가 연결된 상태에서 Micro USB를 통한 전원이 입력되면 배터리를 충전한다. 또한 위의 보드사진에서 좌측 하단에 위치한 3V 핀을 통해서 최대 600mA의 전력 출력을 지원한다.

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(LED On, OFF) ^ 3(LED 3 개) - 1(RGB off case) = 7 가지)을 낼 수 있다.

2. 환경설정

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

Arduino IDE의 설치를 진행한다.

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

2-1. Arduino IDE 설치

Downloads

The screenshot shows the Arduino IDE download page. On the left, there's a logo and the text "Arduino IDE 1.8.19". Below it is a brief description of what the software does and a link to the "Getting Started" page. On the right, under "DOWNLOAD OPTIONS", there are links for Windows, Linux, and Mac OS X. The "Windows app" link is highlighted with a red box. Below each platform section are links for "Release Notes" and "Checksums (sha512)".

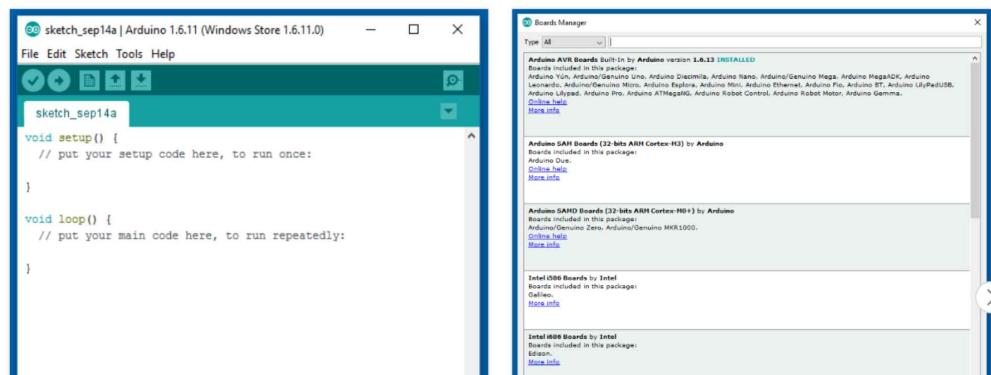
<https://www.arduino.cc/en/Main/Software> Arduino IDE의 설치를 위해 해당 링크로 접속한다.

Windows app 옆에 있는 Get 마크를 선택한다.

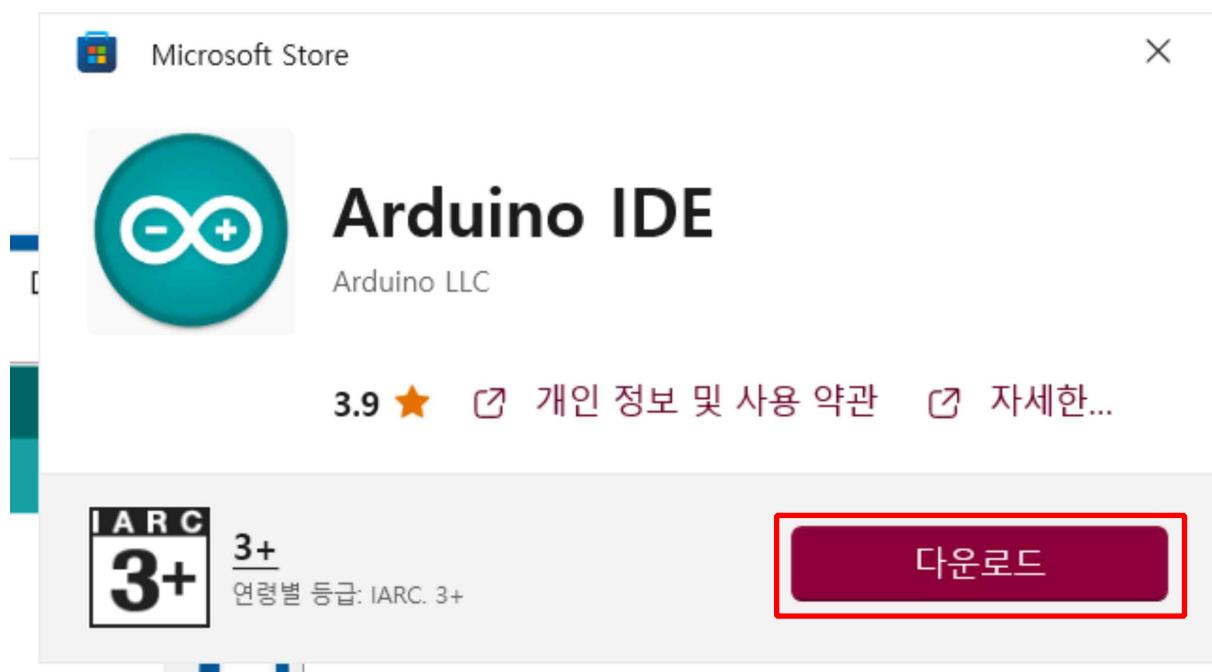


JUST DOWNLOAD를 선택한다.

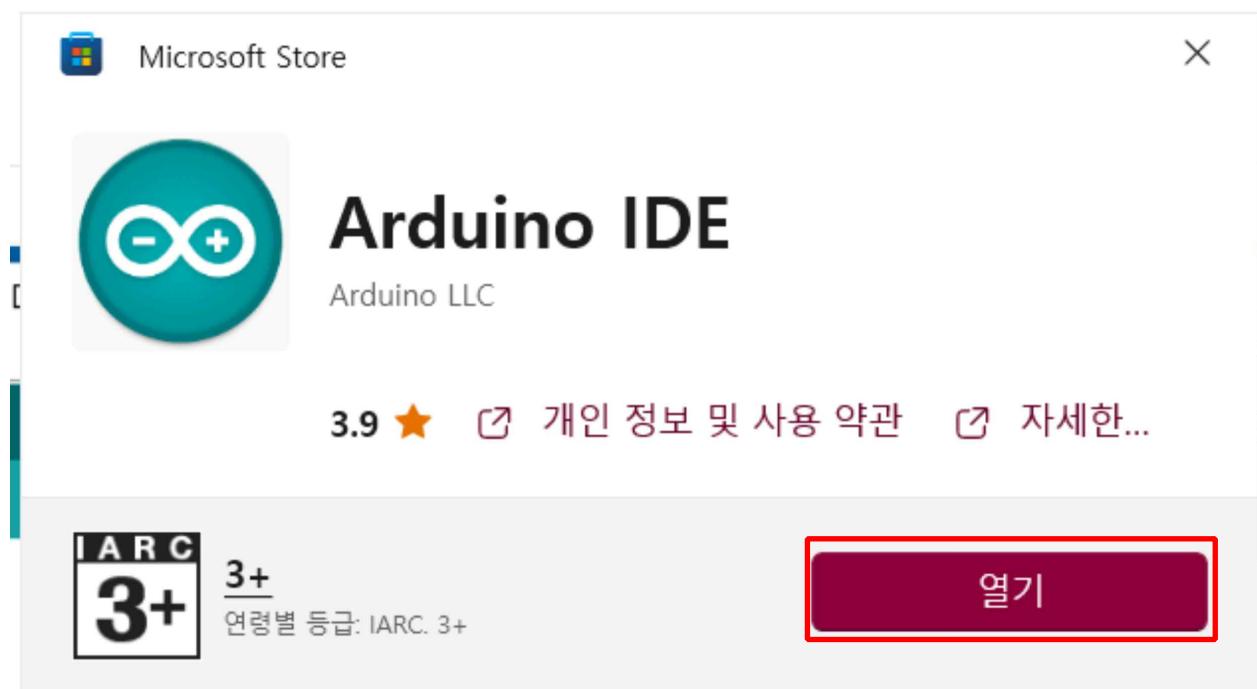
홈 / 개발자 도구 / Arduino IDE



스토어 앱에서 다운로드를 선택한다.



다운로드를 선택한다.

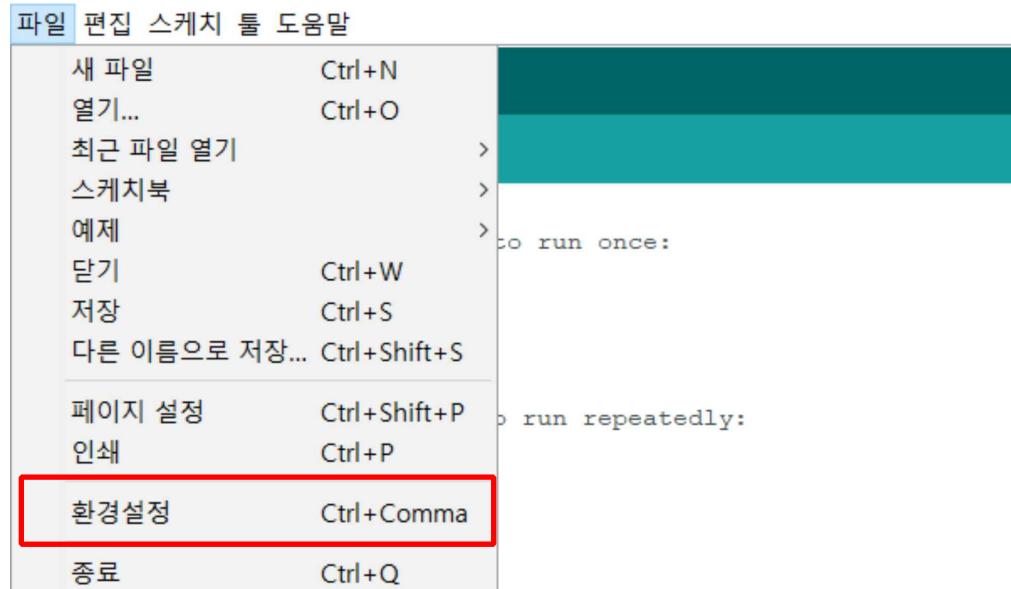


열기를 선택한다.

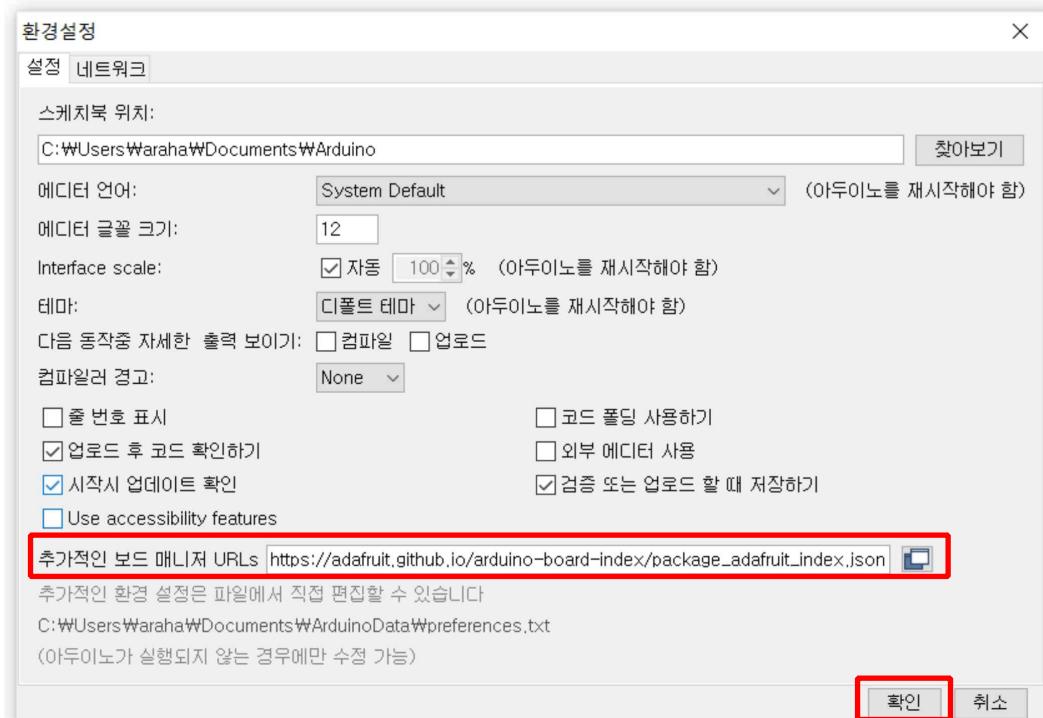
nCube-Thyme-Arduino 가이드

2-2. Arduino IDE 환경설정

sketch_may26a | 아두이노 1.8.19 (Windows Store 1.8.57.0)



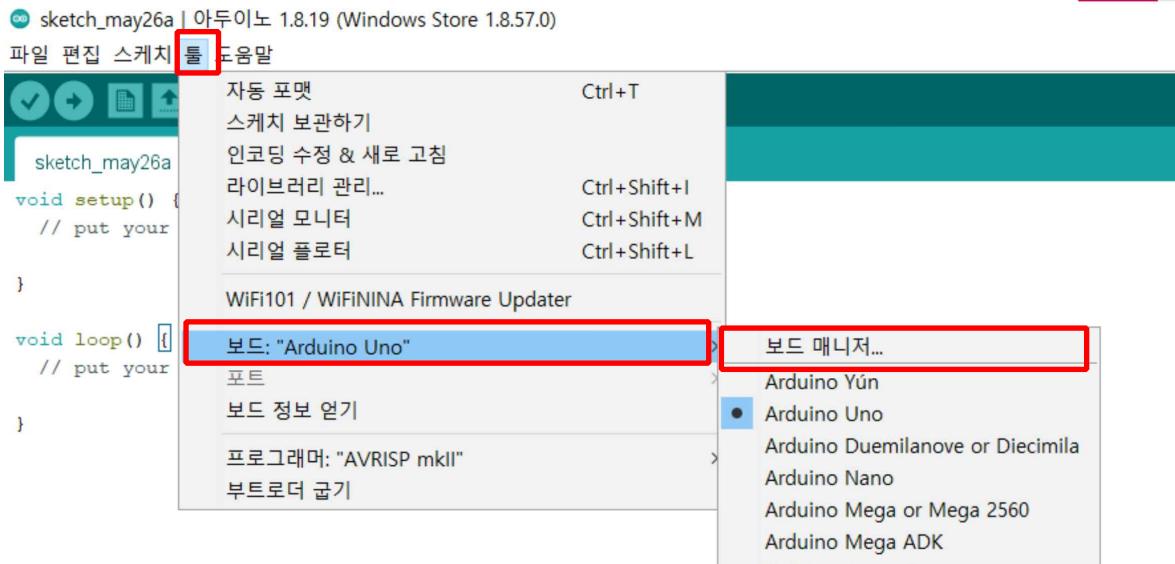
파일에서 환경설정을 선택한다.



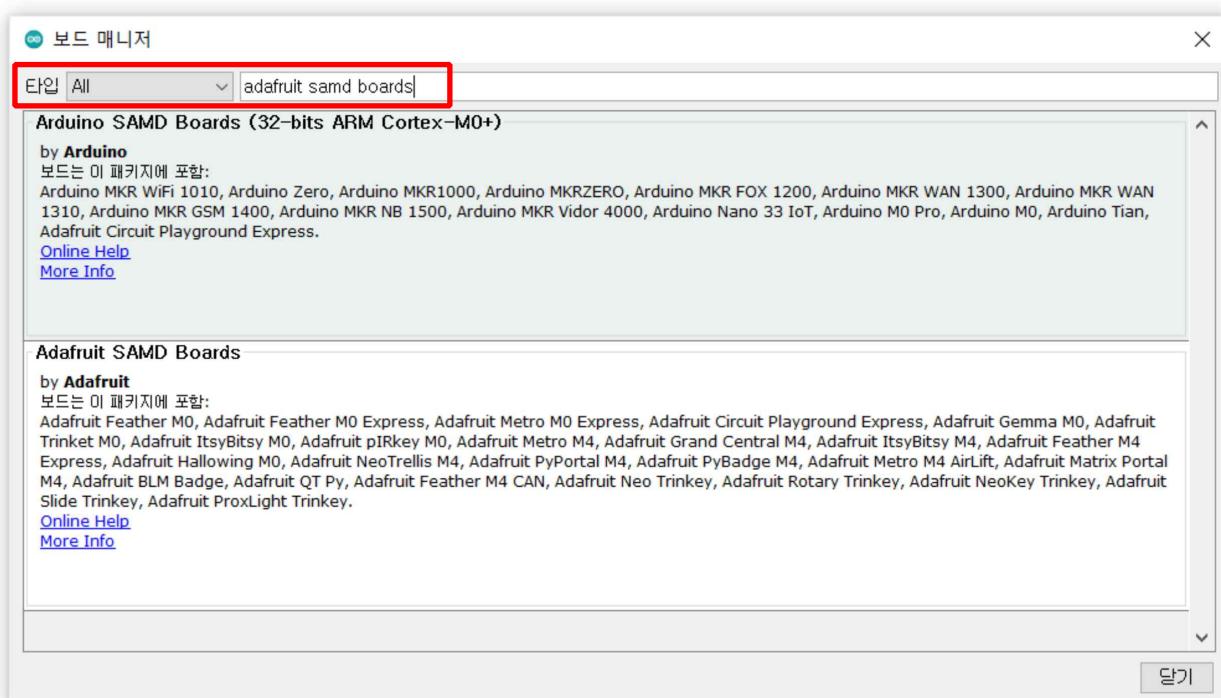
Adafruit Feather M0 보드매니저를 설치하기 위해 보드매니저에 https://adafruit.github.io/arduino-board-index/package_adafruit_index.json를 입력한다.

입력 후, 확인을 선택한다.

nCube-Thyme-Arduino 가이드

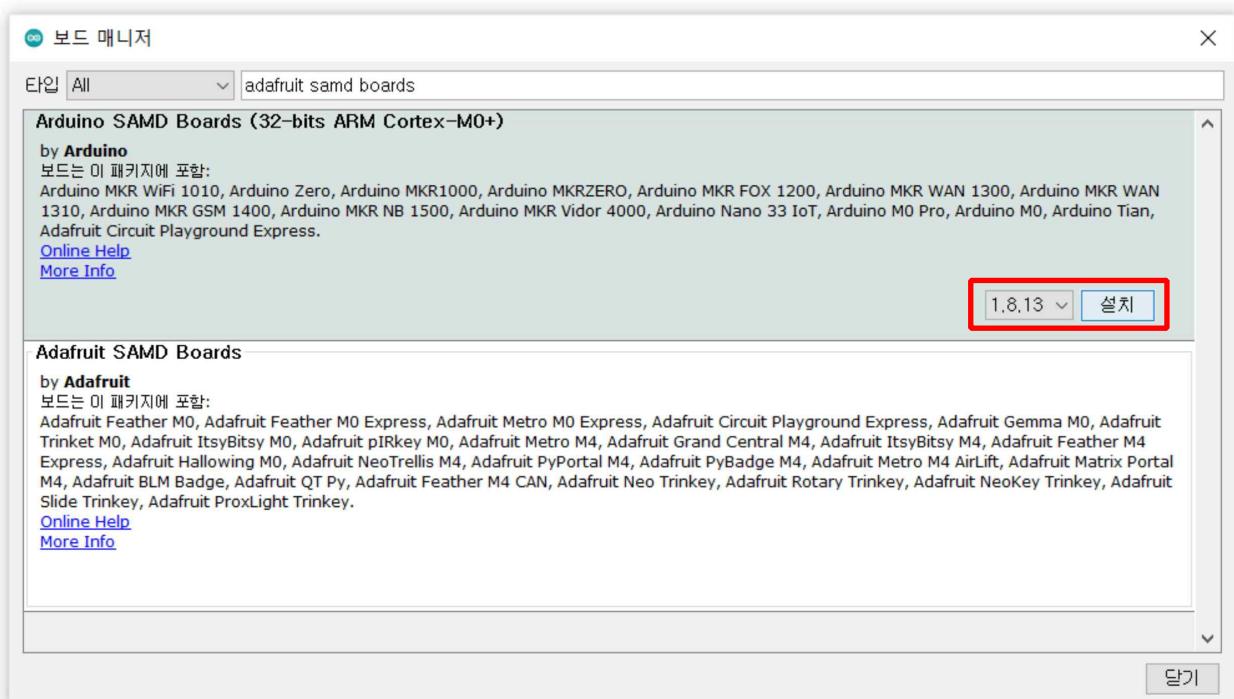


툴 메뉴를 선택 후, 보드 메뉴에 있는 보드 매니저를 선택한다.

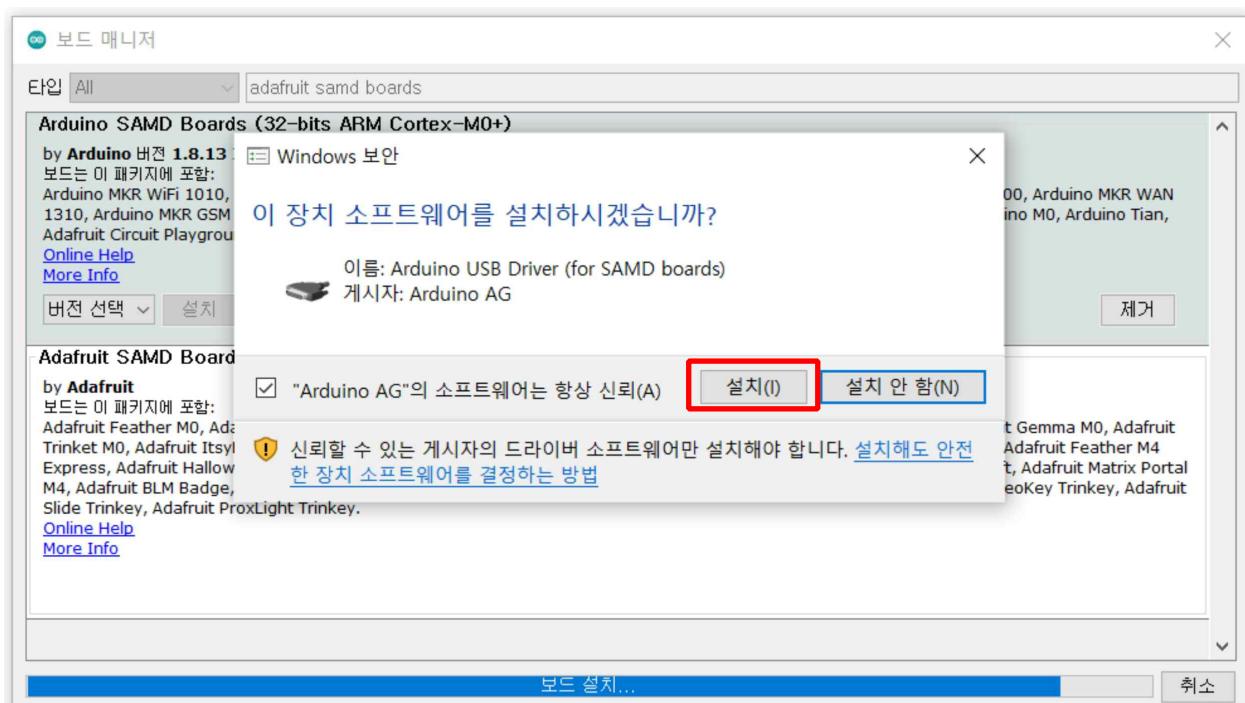


타입에 'adafruit samd boards'를 검색한다.

nCube-Thyme-Arduino 가이드

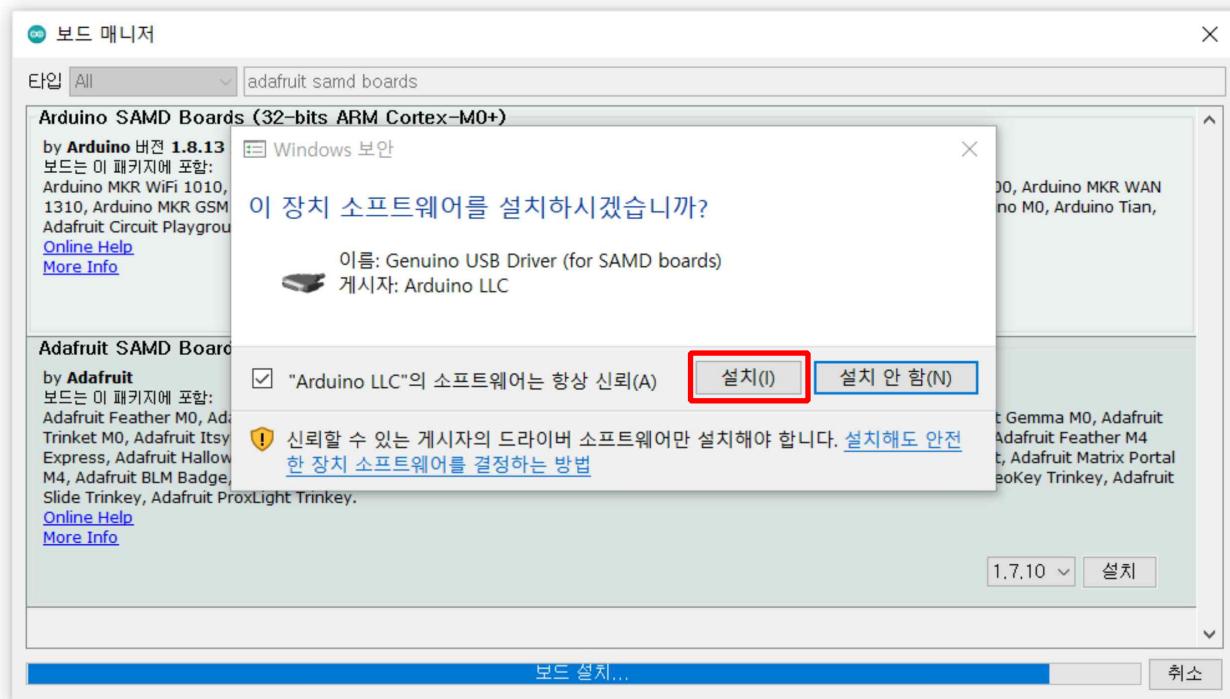


검색 후 나오는 모든 패키지를 설치해야 한다. 먼저, 위에 있는 패키지를 설치한다.

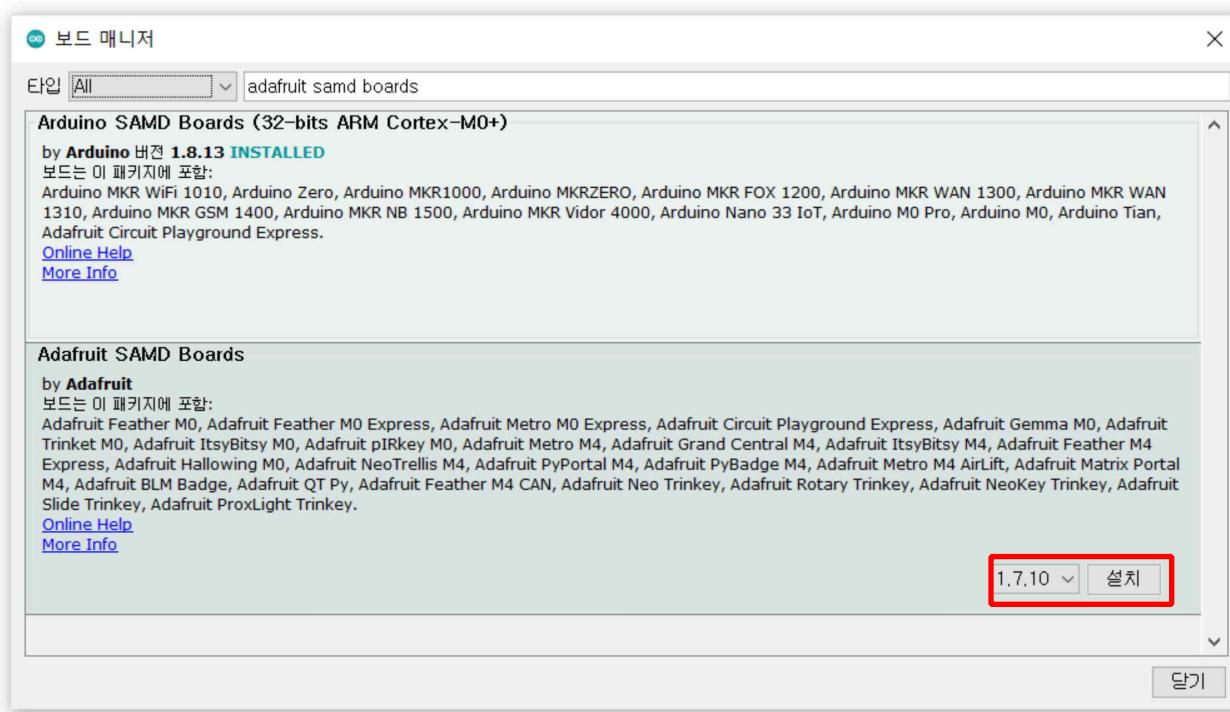


설치를 선택한다.

nCube-Thyme-Arduino 가이드

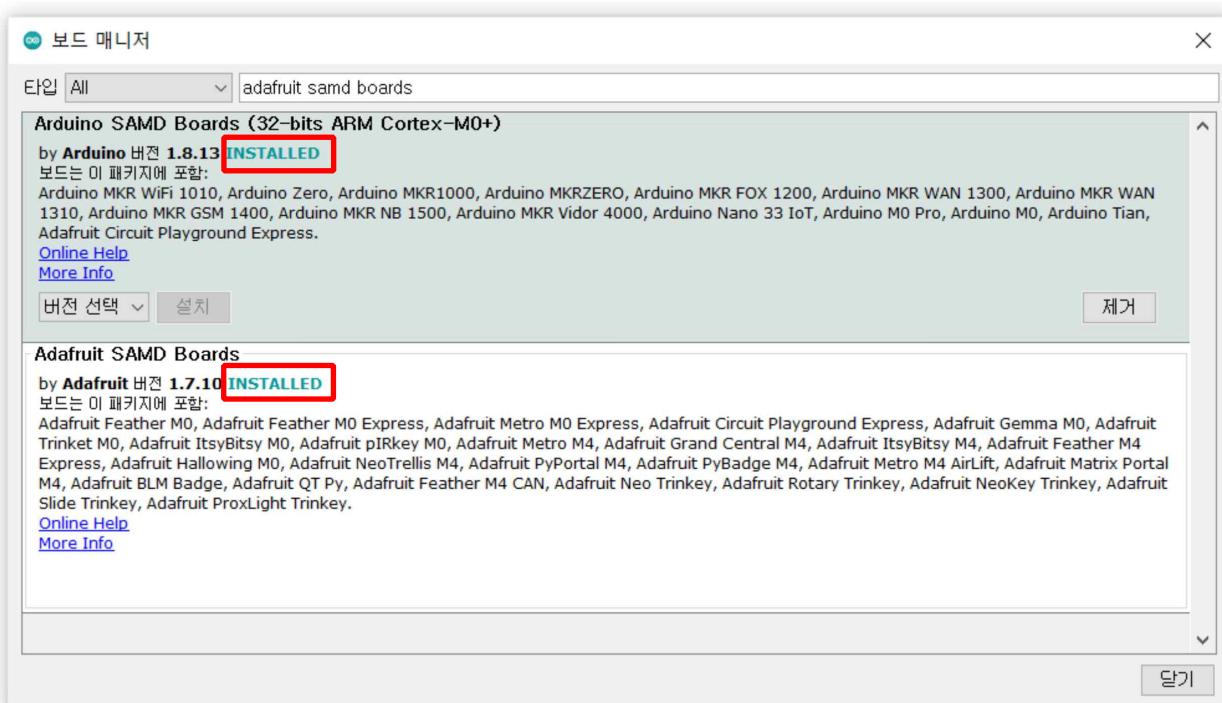


설치를 선택한다.



아래에 있는 패키지도 설치한다.

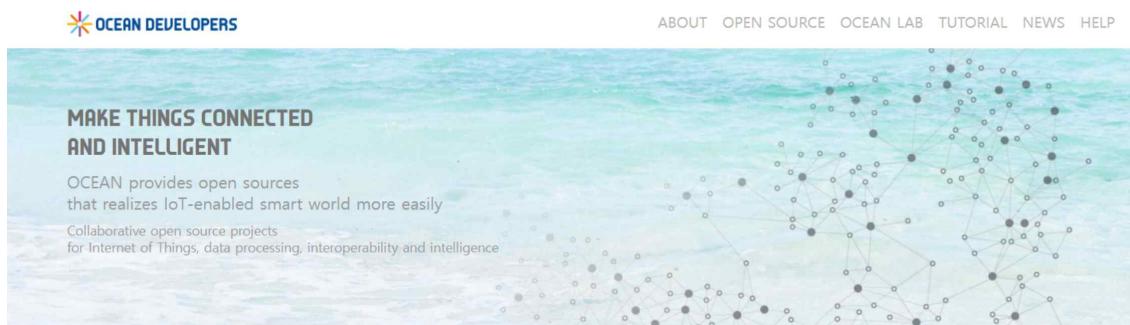
nCube-Thyme-Arduino 가이드



설치가 완료되면 패키지 옆에 'INSTALLED' 라고 뜬다.

3. nCube-Air 개발 환경

3-1. nCube-Air 라이브러리 다운로드



nCube-Air 라이브러리를 다운로드하기 위해 <http://developers.iotocean.org> 사이트에 접속한다.

The screenshot shows the "ITEMS REGISTERED" section of the OCEAN Developers website. It displays a grid of software components categorized by type (SERVER, ETC, NODE) and status (REGISTERED). The components include:

- IoT**: Standard compliant IoT platform implemenatations (12 items registered)
- oneM2M JavaScript SDK**: oneM2M Javascript API v1.0 (March 15, 2021)
- Mobius Open Platform**: 전자부품연구원에서 개발한 ... (v0.1, April 29, 2020)
- Mobius**: IoT Server Platform based o... (v2.4.36, March 13, 2020)
- Performance Test for Mobius Platform.**: Performance Test Result. v1.0 (December 17, 2019)
- Logchain-S**: Blockchain platform for inter... (v1.0, May 26, 2022)
- OASIS**: SI oneM2M Server is a serve... (v2.1.0, July 6, 2018)
- nCube Thyme Nodejs**: Application Dedicated Node... (v2.3.2, July 2, 2018)
- nCube Lavender**: Application Service Node D... (v2.4.0, May 30, 2018)
- nCube Rosemary**: IoT Middle Node Device Pla... (v2.4.0, May 31, 2018)
- nCube Thyme Arduino**: Application Dedicated Node... (v2.1.6, May 2, 2018)

nCube Thyme Arduino를 선택한다.

nCube-Thyme-Arduino 가이드

The screenshot shows the Ocean Developers website with the nCube Thyme Arduino project details. Key information includes:

- OPEN SOURCE** badge
- Type**: Node Device
- Category**: IoT
- Version**: v2.1.6 (May 2, 2018)
- Info**: xuehu0000@keti.re.kr (Chen Nan)
- RELATED ITEMS** button
- Download** button (highlighted with a red box)
- OVERVIEW**, **DOCUMENTATION**, **INSTANCES**, **TERMS&CONDITIONS** buttons
- Git Hub Project Link** section with a red box around the [nCube-Thyme-Arduino GitHub](#) link.

Download를 선택 후, nCube-Thyme-Arduino GitHub를 선택한다.

The screenshot shows the GitHub repository page for nCube-Thyme-Arduino. Key elements include:

- Code** tab (highlighted with a green box)
- Branches**: master (2 branches)
- Tags**: 7 tags
- Clone** section with [HTTPS](https://github.com/IoTKETI/nCube-Thyme-Arduino), [SSH](ssh://github.com/IoTKETI/nCube-Thyme-Arduino), and [GitHub CLI](#) options
- Code** dropdown menu with [Download ZIP](#) option (highlighted with a red box)
- File List**: README.md, .gitignore, libraries, Document
- README.md Content**: nCube:Thyme for Arduino
- Version**: 2.1.6 (20180514)

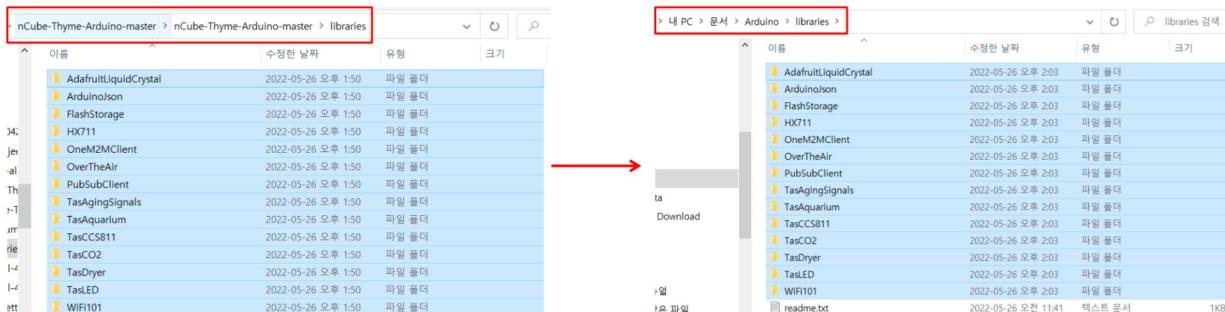
Code를 선택 후, Download ZIP을 선택하여 다운로드 받는다. 그리고 압축을 푼다.

nCube-Thyme-Arduino 가이드

nCube-Thyme-Arduino-master > nCube-Thyme-Arduino-master > libraries >			
이름	수정한 날짜	유형	크기
AdafruitLiquidCrystal	2022-05-26 오후 1:50	파일 폴더	
ArduinoJson	2022-05-26 오후 1:50	파일 폴더	
FlashStorage	2022-05-26 오후 1:50	파일 폴더	
HX711	2022-05-26 오후 1:50	파일 폴더	
OneM2MClient	2022-05-26 오후 1:50	파일 폴더	
OverTheAir	2022-05-26 오후 1:50	파일 폴더	
PubSubClient	2022-05-26 오후 1:50	파일 폴더	
TasAgingSignals	2022-05-26 오후 1:50	파일 폴더	
TasAquarium	2022-05-26 오후 1:50	파일 폴더	
TasCCS811	2022-05-26 오후 1:50	파일 폴더	
TasCO2	2022-05-26 오후 1:50	파일 폴더	
TasDryer	2022-05-26 오후 1:50	파일 폴더	
TasLED	2022-05-26 오후 1:50	파일 폴더	
WiFi101	2022-05-26 오후 1:50	파일 폴더	

압축을 푼 후, nCube-Thyme-Arduino-master/libraries 폴더의 Adafruit LiquidCrystal 폴더명을 수정한다.

Adafruit LiquidCrystal 폴더명에서 공백을 제거하여 AdafruitLiquidCrystal로 수정한다.

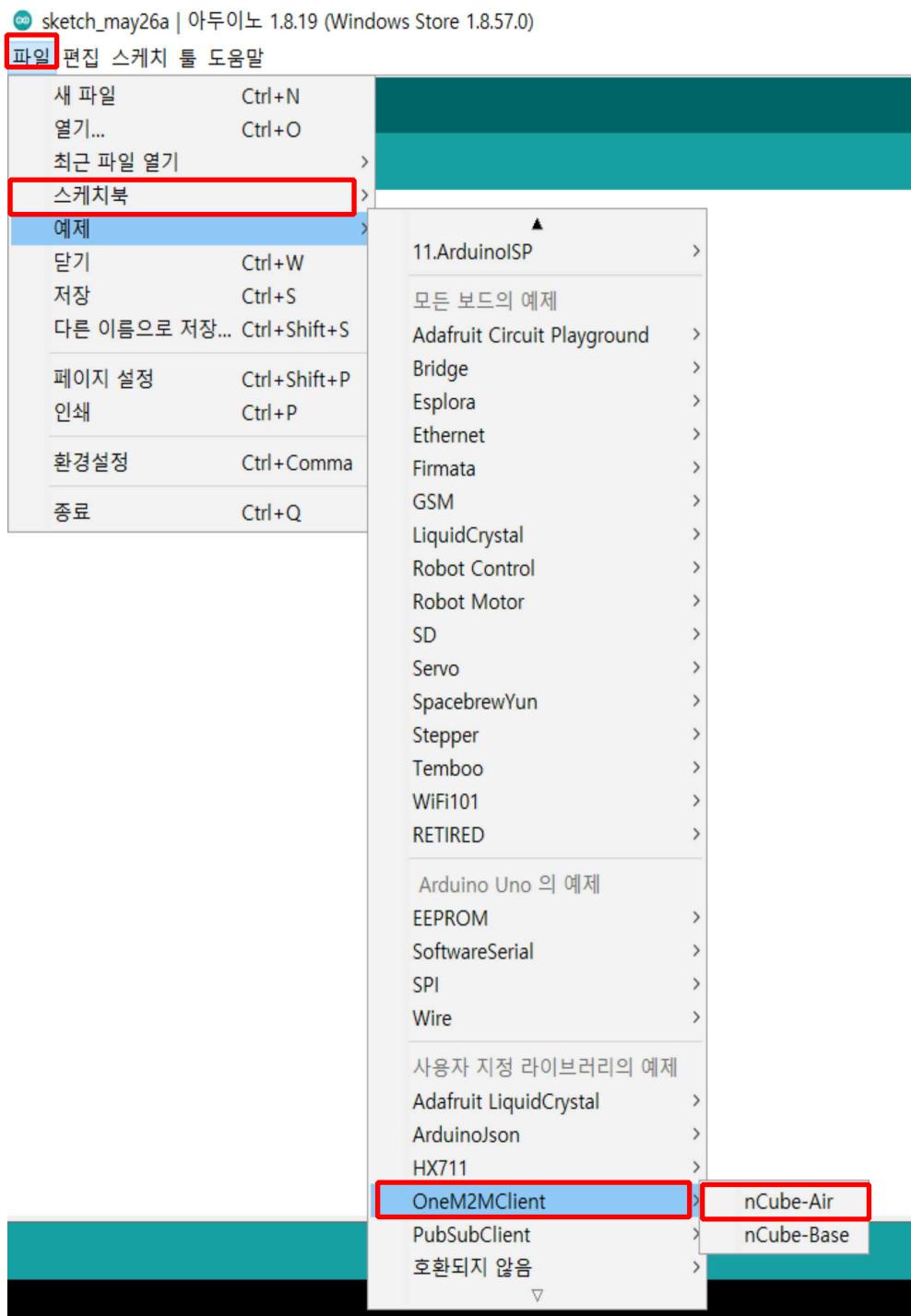


nCube-Thyme-Arduino-master > nCube-Thyme-Arduino-master > libraries			
이름	수정한 날짜	유형	크기
AdafruitLiquidCrystal	2022-05-26 오후 1:50	파일 폴더	
ArduinoJson	2022-05-26 오후 1:50	파일 폴더	
FlashStorage	2022-05-26 오후 1:50	파일 폴더	
HX711	2022-05-26 오후 1:50	파일 폴더	
OneM2MClient	2022-05-26 오후 1:50	파일 폴더	
OverTheAir	2022-05-26 오후 1:50	파일 폴더	
PubSubClient	2022-05-26 오후 1:50	파일 폴더	
TasAgingSignals	2022-05-26 오후 1:50	파일 폴더	
TasAquarium	2022-05-26 오후 1:50	파일 폴더	
TasCCS811	2022-05-26 오후 1:50	파일 폴더	
TasCO2	2022-05-26 오후 1:50	파일 폴더	
TasDryer	2022-05-26 오후 1:50	파일 폴더	
TasLED	2022-05-26 오후 1:50	파일 폴더	
WiFi101	2022-05-26 오후 1:50	파일 폴더	

내 PC > 문서 > Arduino > libraries			
이름	수정한 날짜	유형	크기
AdafruitLiquidCrystal	2022-05-26 오후 2:03	파일 폴더	
ArduinoJson	2022-05-26 오후 2:03	파일 폴더	
FlashStorage	2022-05-26 오후 2:03	파일 폴더	
HX711	2022-05-26 오후 2:03	파일 폴더	
OneM2MClient	2022-05-26 오후 2:03	파일 폴더	
OverTheAir	2022-05-26 오후 2:03	파일 폴더	
PubSubClient	2022-05-26 오후 2:03	파일 폴더	
TasAgingSignals	2022-05-26 오후 2:03	파일 폴더	
TasAquarium	2022-05-26 오후 2:03	파일 폴더	
TasCCS811	2022-05-26 오후 2:03	파일 폴더	
TasCO2	2022-05-26 오후 2:03	파일 폴더	
TasDryer	2022-05-26 오후 2:03	파일 폴더	
TasLED	2022-05-26 오후 2:03	파일 폴더	
WiFi101	2022-05-26 오후 2:03	파일 폴더	
readme.txt	2022-05-26 오후 11:41	텍스트 문서	1KB

nCube-Thyme-Arduino-master/libraries 폴더의 라이브러리들을 Documents/Arduino/libraries 폴더로 복사한다.

3-2. nCube-Air 불러오기



Arduino IDE 툴에서 파일을 선택 후, 예제를 선택하여 OneM2M Client를 선택한다. 그 후, nCube-Air를 선택한다.

4. 보드 및 센서 환경설정

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

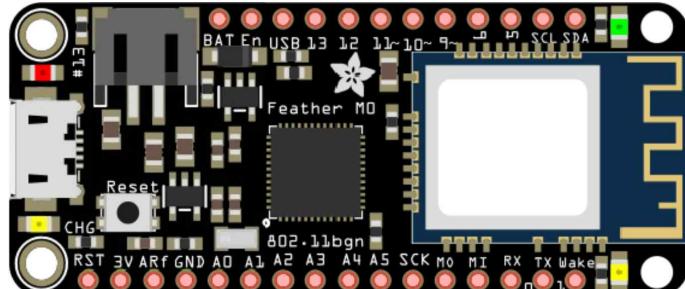
4-1. 보드 및 센서 준비



- 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

4-2. 보드와 센서 개요

Adafruit Feather M0



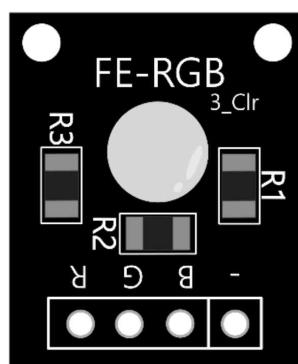
- Cortex M0 processor
- SPI, I2C, UART 통신지원
- 802.11bgn을 지원하는 Atmel WINC1500 WiFi chip

CCS811 Air Quality Sensor



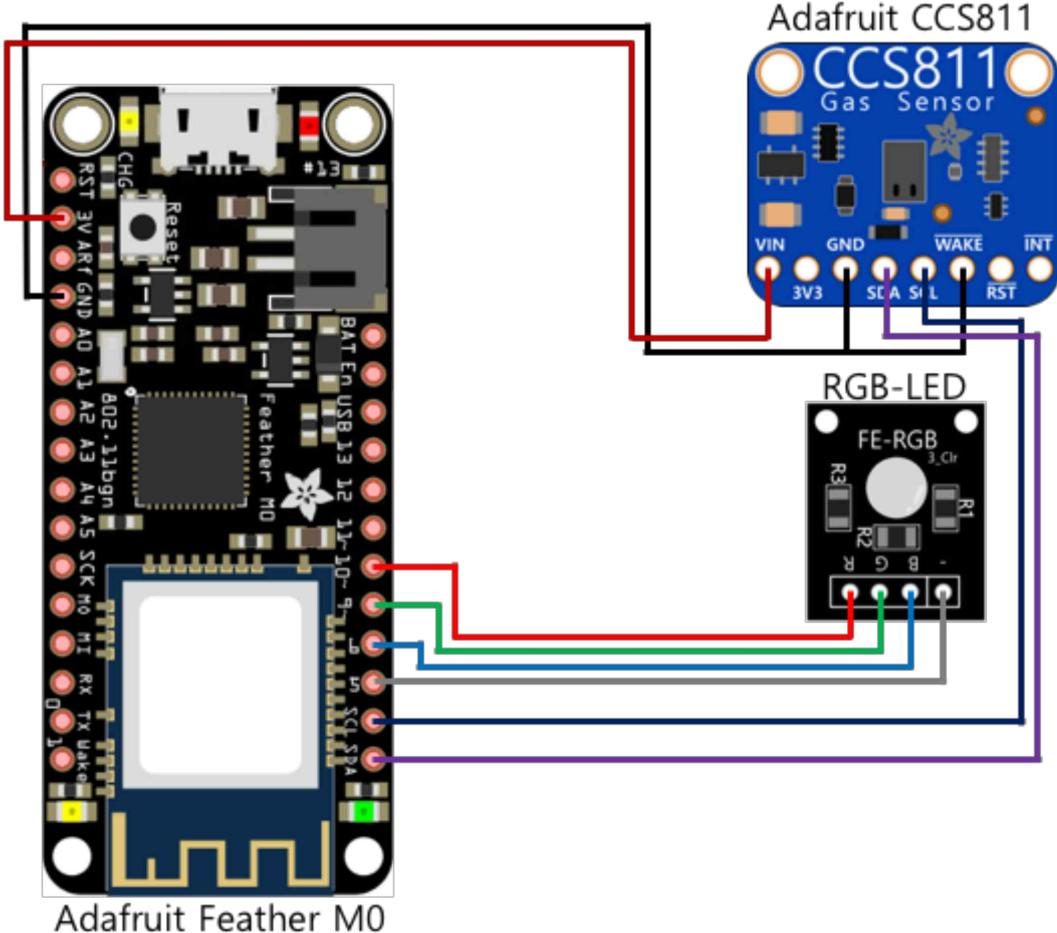
- 대기중의 CO₂, TVOC(TOTAL Volatile Organic Compounds) 측정
- I2C 통신을 이용하여 Arduino에 센서 값을 전송

RGB-LED



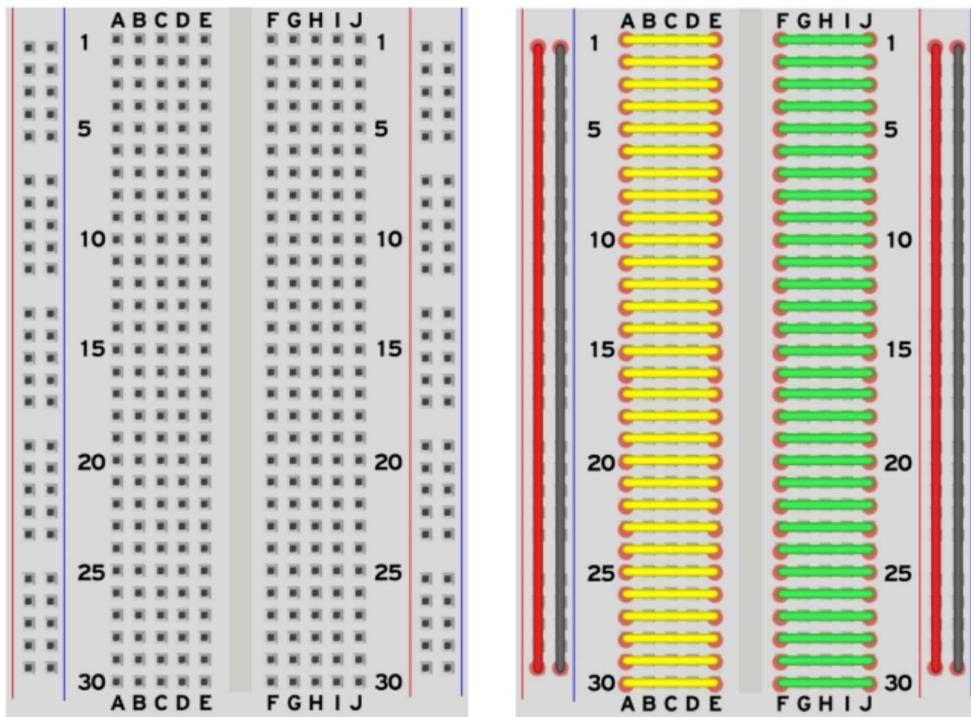
- RGB 3-color LED 보드는 빨간색, 녹색, 파란색 LED를 포함
- RGB 3-color LED의 LED들은 각각의 R, G, B핀의 전원 입력에 따라 점멸

4-3. 센서와 보드 연결



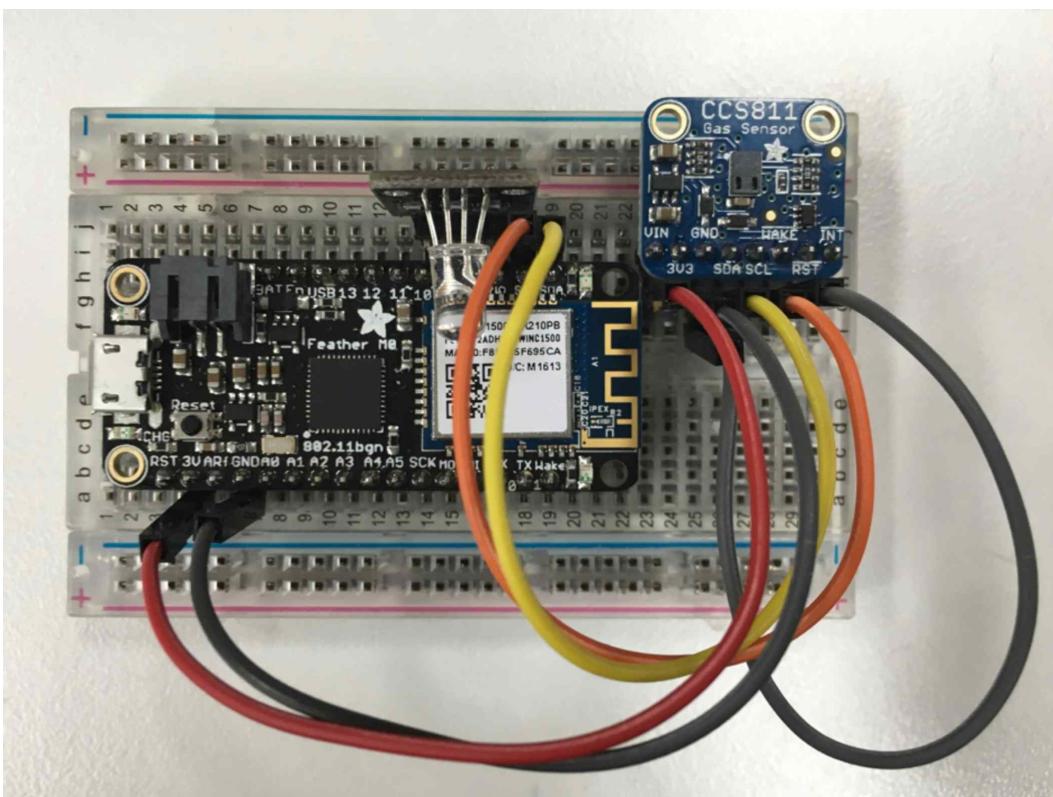
위의 그림과 같이 Adafruit Feather M0 보드와 CCS811 Air Quality Sensor, RGB-LED를 연결한다.

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.

5. nCube-Air 설정 및 실행

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

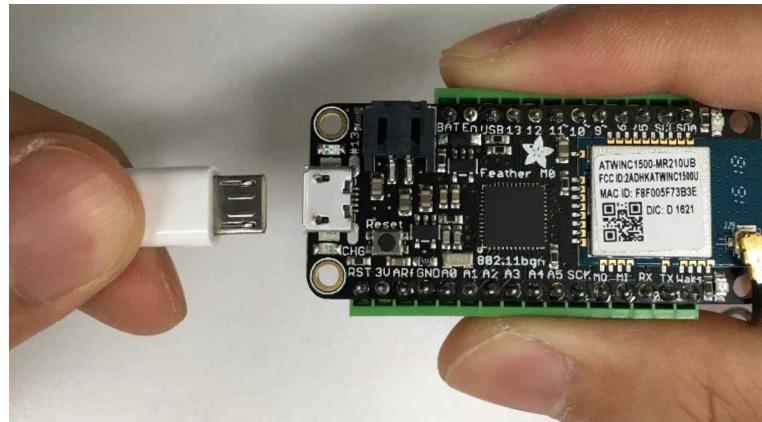
5-1. nCube-Air 소스코드 설정

```
nCube-Air$  
  uint8_t push_idx;  
  String ref[QUEUE_SIZE];  
  String con[QUEUE_SIZE];  
  String rqi[QUEUE_SIZE];  
  unsigned long* previousMillis[QUEUE_SIZE];  
 } queue_t;  
  
queue_t noti_q;  
queue_t upload_q;  
  
String body_str = "";  
  
char resp_topic[48];  
char noti_topic[48];  
  
unsigned long system_watchdog = 0;  
  
// -----  
// User Define  
// Period of Sensor Data, can make more  
const unsigned long base_generate_interval = 20 * 1000;  
unsigned long temp_generate_previousMillis = 0;  
unsigned long temp_generate_interval = base_generate_interval;  
unsigned long tvoc_generate_previousMillis = 0;  
unsigned long tvoc_generate_interval = base_generate_interval;  
unsigned long co2_generate_previousMillis = 0;  
unsigned long co2_generate_interval = base_generate_interval;  
  
// Information of CSE as Mobius with MQTT  
const String FIRMWARE_VERSION = "1.0.0.0";  
String AE_NAME = "air5";  
String AE_ID = "S" + AE_NAME;  
const String CSE_ID = "/Mobius2";  
const String CB_NAME = "Mobius";  
const char* MOBIUS_MQTT_BROKER_IP = "203.253.128.161"; //"192.168.33.86";  
const uint16_t MOBIUS_MQTT_BROKER_PORT = 1883;  
  
OneM2MClient nCube;  
  
// add TAS(Thing Adaptation Layer) for Sensor  
#include "TasLED.h"  
TasLED tasLed;  
  
#include "TasCCS811.h"  
TasCCS811 TasCCSSensor;
```

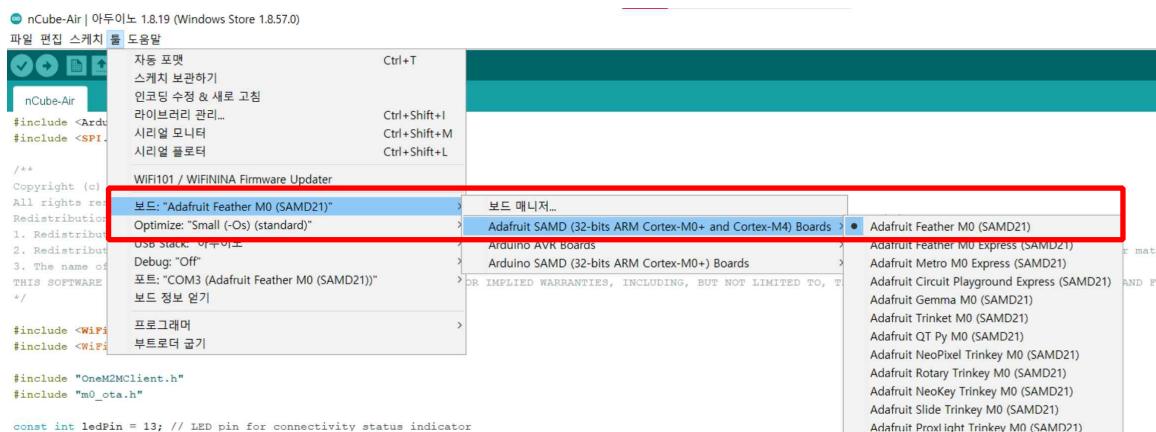
nCube-Air.ino 소스 코드에서 120~127 라인 중 122 라인은 AE NAME을 설정하는 부분이다. 이 부분을 자신만의 AE 이름으로 바꾸어 준다.

nCube-Thyme-Arduino 가이드

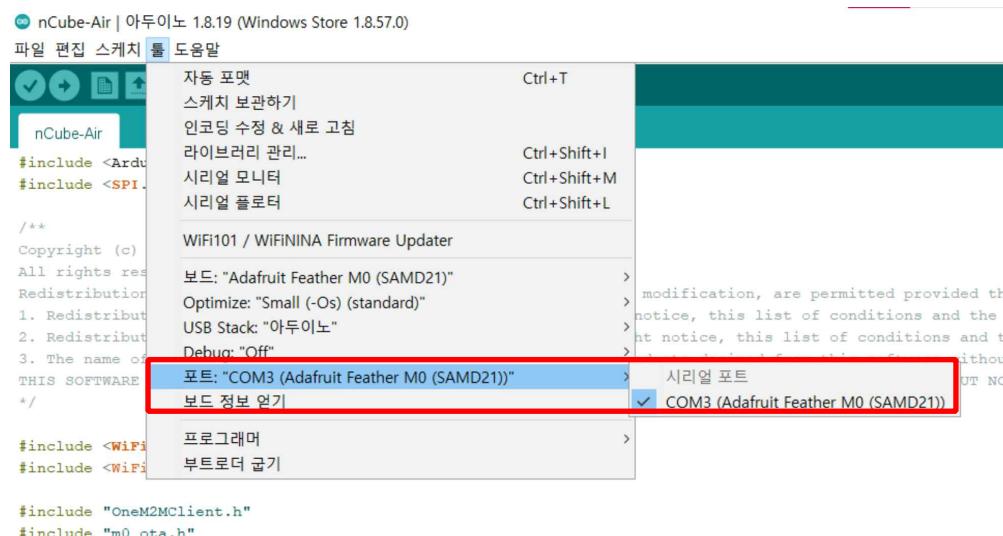
5-2. nCube-Air 소스코드 업로드 및 실행



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



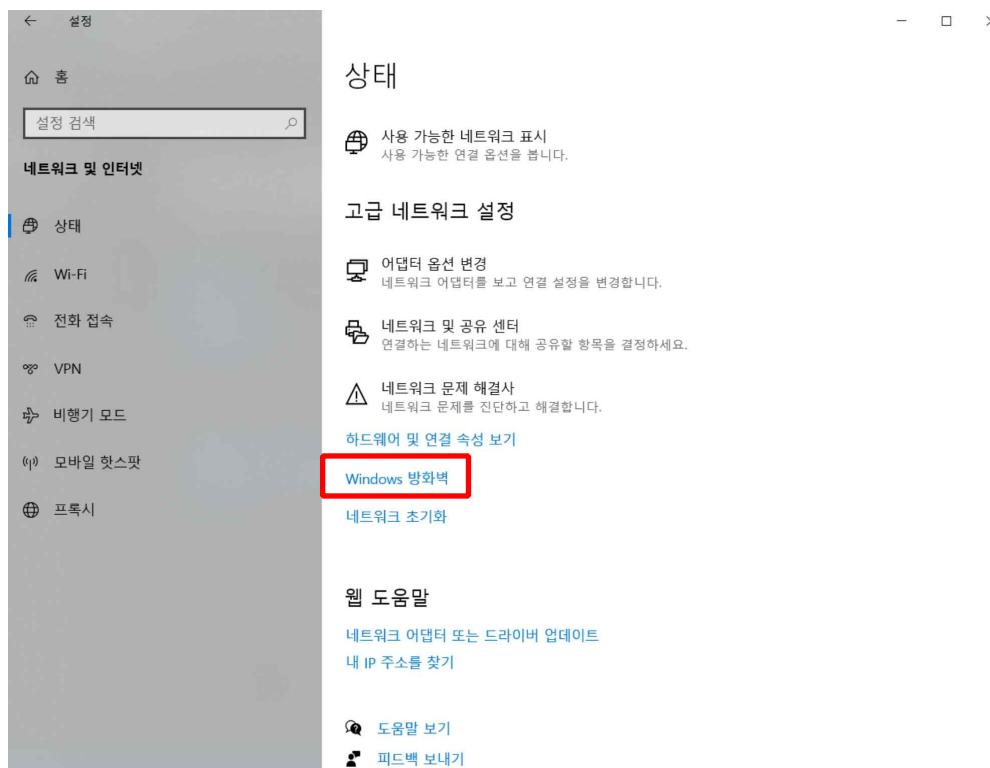
툴을 선택 후, 보드 메뉴에서 Adafruit Feather M0을 선택한다.



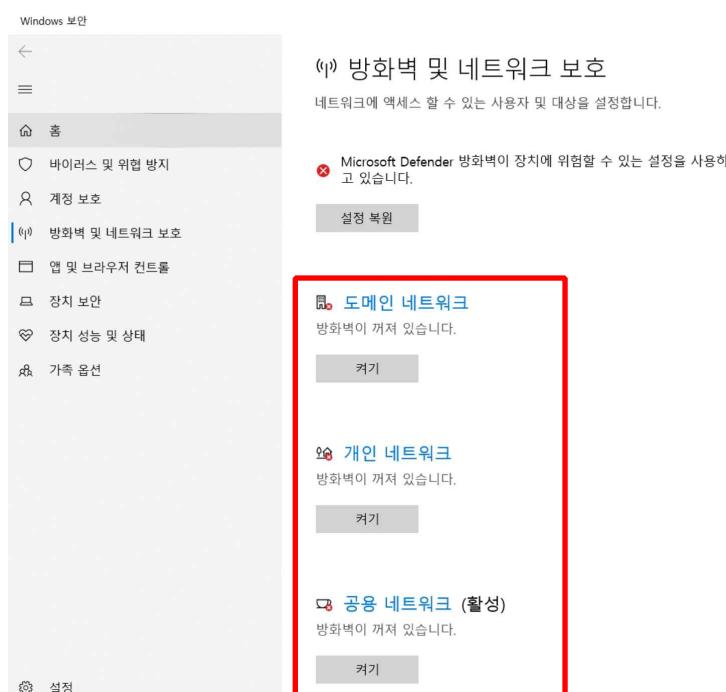
포트를 선택한다.

nCube-Thyme-Arduino 가이드

5-3. 네트워크 설정



환경설정 - Windows 방화벽을 선택한다.



모든 방화벽을 다 끈다.

nCube-Thyme-Arduino 가이드

The screenshot shows a Windows File Explorer window with the following path highlighted by a red box: C:\Program Files\mosquitto. The table below lists the contents of this folder.

이름	수정한 날짜	유형	크기
devel	2022-05-26 오후 3:32	파일 폴더	
aclfile.example	2021-11-17 오전 9:28	EXAMPLE 파일	1KB
ChangeLog	2021-11-17 오전 9:28	텍스트 문서	129KB
edl-v10	2021-11-17 오전 9:28	파일	2KB
epl-v20	2021-11-17 오전 9:28	파일	14KB
libcrypto-1_1-x64.dll	2021-08-24 오후 6:57	응용 프로그램 확장	3,338KB
libssl-1_1-x64.dll	2021-08-24 오후 6:57	응용 프로그램 확장	668KB
mosquitto	2022-05-26 오후 8:51	CONF 파일	40KB
mosquitto.dll	2021-11-22 오전 5:13	응용 프로그램 확장	86KB
mosquitto	2021-11-22 오전 5:14	응용 프로그램	372KB
mosquitto_ctrl	2021-11-22 오전 5:13	응용 프로그램	75KB
mosquitto_dynamic_security.dll	2021-11-22 오전 5:13	응용 프로그램 확장	118KB
mosquitto_passwd	2021-11-22 오전 5:13	응용 프로그램	22KB
mosquitto_pub	2021-11-22 오전 5:13	응용 프로그램	51KB
mosquitto_rr	2021-11-22 오전 5:13	응용 프로그램	78KB

C:\Program Files\mosquitto 경로에 있는 mosquito.conf 파일을 연다.

The screenshot shows a code editor displaying the mosquito.conf configuration file. The file contains various configuration options, with the following lines highlighted by a red box:

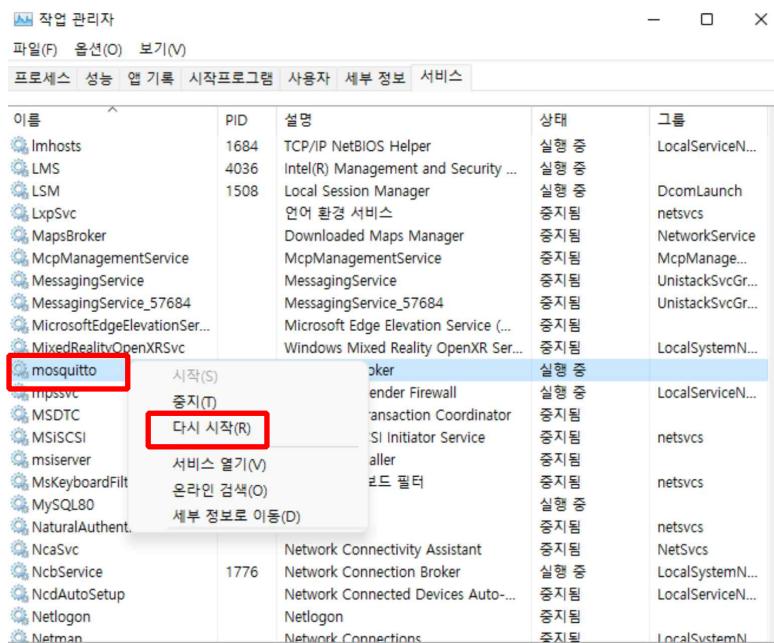
```
220 # address and so restrict access to the associated network and
221 # interface. By default, mosquitto will listen on all interfaces.
222 # Note that for a websockets listener it is not possible to bind to a host
223 # name.
224 #
225 # On systems that support Unix Domain Sockets, it is also possible
226 # to create a Unix socket rather than opening a TCP socket. In
227 # this case, the port number should be set to 0 and a unix socket
228 # path must be provided, e.g.
229 # listener 0 /tmp/mosquitto.sock
230 #
231 # listener port-number [ip address/host name/unix socket path]
232 listener 1883
233 protocol mqtt
234 #
235 # By default, a listener will attempt to listen on all supported IP protocol
236 # versions. If you do not have an IPv4 or IPv6 interface you may wish to
237 # disable support for either of those protocol versions. In particular, note
238 # that due to the limitations of the websockets library, it will only ever
239 # attempt to open IPv6 sockets if IPv6 support is compiled in, and so will fail
240 # if IPV6 is not available.
241 #
242 # Set to `ipv4` to force the listener to only use IPv4, or set to `ipv6` to
243 # force the listener to only use IPv6. If you want support for both IPv4 and
244 # IPv6, then do not use the socket_domain option.
245 #
246 #socket_domain
247 #
248 # Bind the listener to a specific interface. This is similar to
```

주석 처리 되어있는 listener 코드의 주석을 해제하고 포트번호인 1883을 작성한다.
그 줄 아래에 protocol mqtt를 작성한다.

nCube-Thyme-Arduino 가이드

```
Get Started mosquitto.conf x
C: > Program Files > mosquitto > mosquitto.conf
514
515 # If set, only clients that have a matching prefix on their
516 # clientid will be allowed to connect to the broker. By default,
517 # all clients may connect.
518 # For example, setting "secure-" here would mean a client "secure-"
519 # client" could connect but another with clientid "mqtt" couldn't.
520 #clientid_prefixes
521
522 # Boolean value that determines whether clients that connect
523 # without providing a username are allowed to connect. If set to
524 # false then a password file should be created (see the
525 # password_file option) to control authenticated client access.
526 #
527 # Defaults to false, unless there are no listeners defined in the configuration
528 # file, in which case it is set to true, but connections are only allowed from
529 # the local machine.
530 allow_anonymous true
531
532 # -----
533 # Default authentication and topic access control
534 # -----
535
536 # Control access to the broker using a password file. This file can be
537 # generated using the mosquitto_passwd utility. If TLS support is not compiled
538 # into mosquitto (it is recommended that TLS support should be included) then
539 # plain text passwords are used, in which case the file should be a text file
540 # with lines in the format:
541 # username:password
542 # The password (and colon) may be omitted if desired, although this
543 # offers very little in the way of security.
544 #
545 # See the TLS client require_certificate and use_identity_as_username options
546 # for alternative authentication options. If a plugin is used as well as
547 # password_file, the plugin check will be made first.
```

주석 처리 되어있는 allow_anonymous 코드의 주석을 해제하고 true로 바꿔준다.



mosquitto 설정 변경 후, 작업관리자를 연다.

서비스 탭에서 mosquitto를 다시 시작해준다.

5-4. nCube-Air 실행

```
C:\Users\Webleem\Desktop\Mobius-master>node mobius.js
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
CPU Count: 12
select_ri_lookup /Mobius: 2.842ms
update_cb_poa_csi /Mobius: 2.816ms
"""

Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
Production Mode
Garbage collection is not exposed
mobius server (10.252.73.149) running at 7579 port
sgn_mqtt_client is connected
Production Mode
Garbage collection is not exposed
select_ri_lookup /Mobius: 2.631ms
update_cb_poa_csi /Mobius: 0.877ms
Production Mode
"""
mobius server (10.252.73.149) running at 7579 port
Garbage collection is not exposed
Production Mode
Garbage collection is not exposed
Production Mode
sgn_mqtt_client is connected
select_ri_lookup /Mobius: 2.04ms
update_cb_poa_csi /Mobius: 0.984ms
"""


```

nCube-Air를 실행하기 위해서는 mobius 서버를 열어놓아야 한다.

Mobius 폴더로 이동 후, node mobius.js를 입력하여 mobius를 실행한다.



```
● nCube-Air | 아두이노 1.8.19 (Windows Store 1.8.57.0)
파일 편집 스케치 를 도움말
[+] 업로드
nCube-Air
#include <Arduino.h>
#include <SPI.h>

/*
Copyright (c) 2018, OCEAN
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#include <WiFi101.h>
#include <WiFiMDNSResponder.h>

#include "OneM2MClient.h"
#include "m0_cta.h"

const int ledPin = 13; // LED pin for connectivity status indicator

#define R + DSR_WTET = 1.
```

이제 연결된 보드에 nCube-Air를 업로드 할 차례이다. 업로드 버튼인 화살표 버튼을 선택한다.

nCube-Thyme-Arduino 가이드

The screenshot shows the Arduino IDE interface. The top menu bar indicates "nCube-Air | 아두이노 1.8.19 (Windows Store 1.8.57.0)" and "파일 편집 스케치 툴 도움말". The code editor contains the following code:

```
/*  
 * nCube-Air | 아두이노 1.8.19 (Windows Store 1.8.57.0)  
 * 파일 편집 스케치 툴 도움말  
 */  
  
#include <Arduino.h>  
#include <SPI.h>  
  
/*  
Copyright (c) 2010, OCEAN  
All rights reserved.  
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:  
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.  
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.  
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.  
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
*/  
  
#include <WiFi101.h>  
#include <WiFiManager.h>  
  
< />  
  
[터미널 창]  
스케치는 프로그램 저장 공간 76664 바이트(29%)를 사용. 최대 262144 바이트.  
Device : ATSAMD21G18  
Version : v2.0 [Arduino:XYZ] Mar 5 2016 17:46:52  
Address : 0x0  
Pages : 4096  
Page Size : 64 bytes  
Total Size : 256KB  
Planes : 1  
Lock Regions : 16  
Locked : none  
Security : false  
ECD : true  
SRK : true  
Write 77592 bytes to flash (1213 pages)  
[=====] 100% (1213/1213 pages)  
Done in 15.050 seconds  
Verify 77592 bytes of flash  
[=====] 100% (1213/1213 pages)  
Verify successful  
Done in 1.236 seconds
```

업로드 버튼을 누르면 소스 코드를 컴파일하고 연결된 보드로 업로드 한다.
업로드가 모두 완료되고 난 뒤에 상태창에 SUCCESS 메시지가 나타난다.

The screenshot shows the Serial Monitor window titled "COM3". The right side of the window has a red box around the "시리얼 모니터" button. The text area displays the following serial communication:

```
("top":1,"to":"/Mobius/air1/co2?rcn=0","fr":"Sair1","rqci":"SNvYX3e1","ty":4,"pc":{"m2m:cin":{"con":40}}}  
Message arrived [/oneM2M/rep/Sair1/Mobius2/json] <---- 37  
("rcn":2001,"rqci":"SNvYX3e1","pc":{})  
<---- 2001  
pop : 4  
push : 5  
pop : 4  
push : 5  
Request [/oneM2M/reg/Sair1/Mobius2/json] -----> 114  
("top":1,"to":"/Mobius/air1/temp?rcn=0","fr":"Sair1","rqci":"xU7RMXzV","ty":4,"pc":{"m2m:cin":{"con":26.23}}}  
Message arrived [/oneM2M/rep/Sair1/Mobius2/json] <---- 37  
("rcn":2001,"rqci":"xU7RMXzV","pc":{})  
<---- 2001  
MQTTPINGREQ ----->  
<---- MQTTINGRESP
```

At the bottom, there are checkboxes for "자동 스크롤" and "타임스탬프 표시", and a dropdown for "波特率" set to "9600 보드레이트".

스케치는 프로그램 저장 공간 76664 바이트(29%)를 사용. 최대 262144 바이트.
Device : ATSAMD21G18
Version : v2.0 [Arduino:XYZ] Mar 5 2016 17:46:52
Address : 0x0
Pages : 4096
Page Size : 64 bytes
Total Size : 256KB
Planes : 1

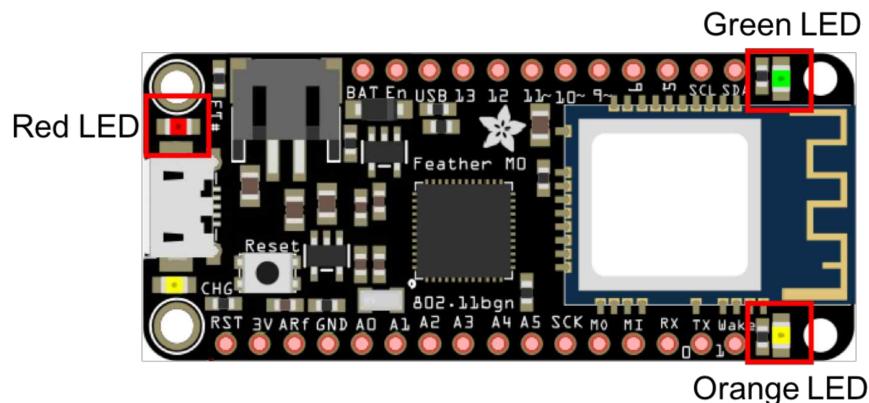
모니터링하기 위해 시리얼 모니터 버튼을 클릭한다.

```
|  
AE ID is air5  
Current fw version is 1.0.0.0  
beginProvision - WIFI INIT  
WIFI_CONNECT  
WIFI_CONNECT  
WIFI_CONNECT  
WIFI_CONNECT  
WIFI_CONNECT  
WIFI_CONNECT  
WIFI_CONNECT  
WIFI_CONNECT  
WIFI_CONNECT
```

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 WiFi로 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.



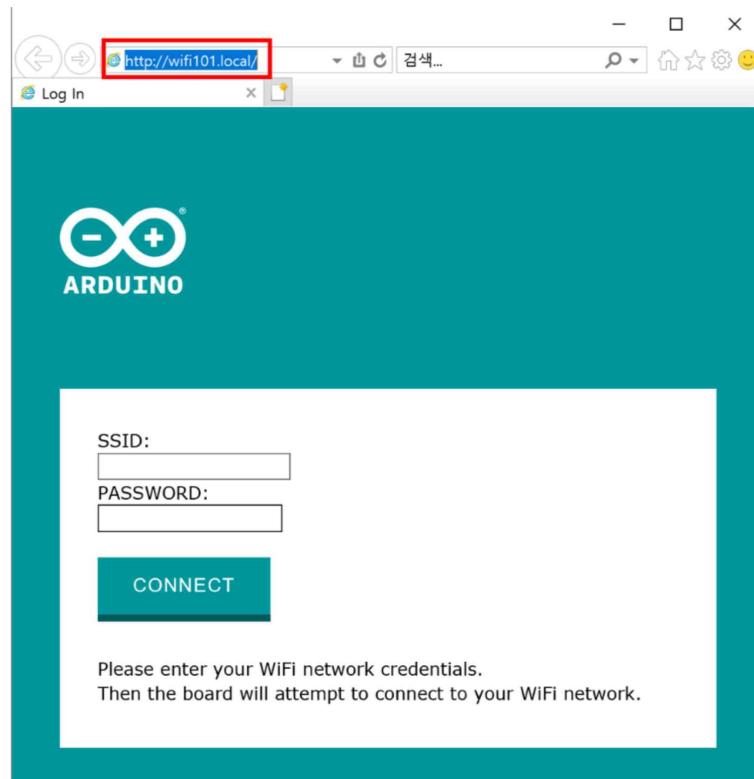
Provision 모드에서는 Red LED가 빠르게 점멸하고, Green LED가 점등된 상태가 유지된다.

nCube-Thyme-Arduino 가이드



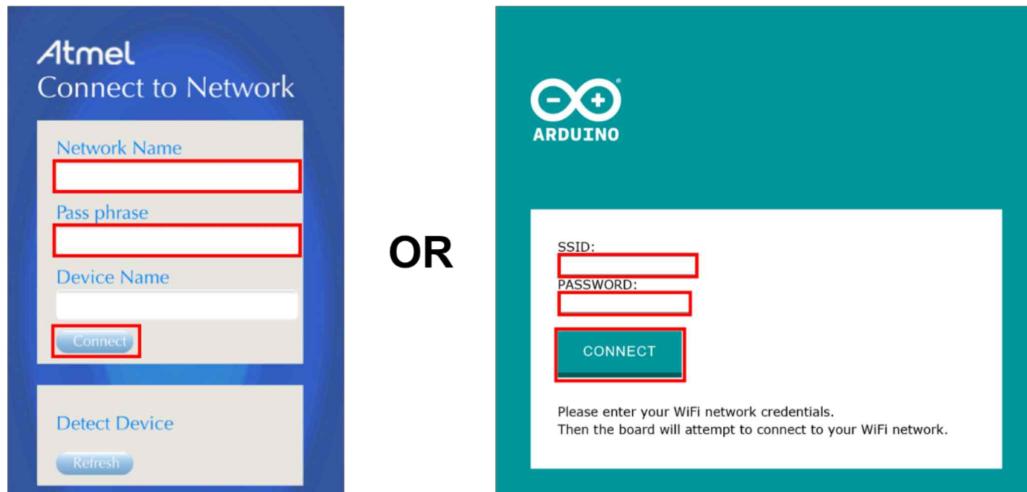
Adafruit Feather M0 보드의 WiFi 모듈 위에 붙어있는 스티커의 MAC 주소 마지막 네 자리의 문자를 기억한다.

그리고 랩탑 또는 스마트폰의 WiFi를 이용하여 **wifi101-[MAC 주소 마지막 네 자리]**로 검색되는 WiFi AP에 연결한다.



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

nCube-Thyme-Arduino 가이드



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

```
_MQTT_CONNECT
Attempting MQTT connection...connected
/oneM2M/resp/Sair1/Mobius2/json Successfully subscribed
/oneM2M/req/Sair1/Mobius2/json Successfully subscribed
create_ae - 0 - YraCyTaZ
Request [/oneM2M/req/Sair1/Mobius2/json] ----> 126
{"op":"1","to":"/Mobius?rcn=0","fr":"Sair1","rqi":"YraCyTaZ","ty":2,"pc":{"m2m:ae":{"rn":"air1","api":3.14,"rr":true}}
create_ae - 0 - kyG9V4HF
Request [/oneM2M/req/Sair1/Mobius2/json] ----> 126
{"op":"1","to":"/Mobius?rcn=0","fr":"Sair1","rqi":"kyG9V4HF","ty":2,"pc":{"m2m:ae":{"rn":"air1","api":3.14,"rr":true}}
create_ae - 0 - X5siw4LO
Request [/oneM2M/req/Sair1/Mobius2/json] ----> 126
{"op":"1","to":"/Mobius?rcn=0","fr":"Sair1","rqi":"X5siw4LO","ty":2,"pc":{"m2m:ae":{"rn":"air1","api":3.14,"rr":true}}
MQTTPINGREQ ---->
```

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.

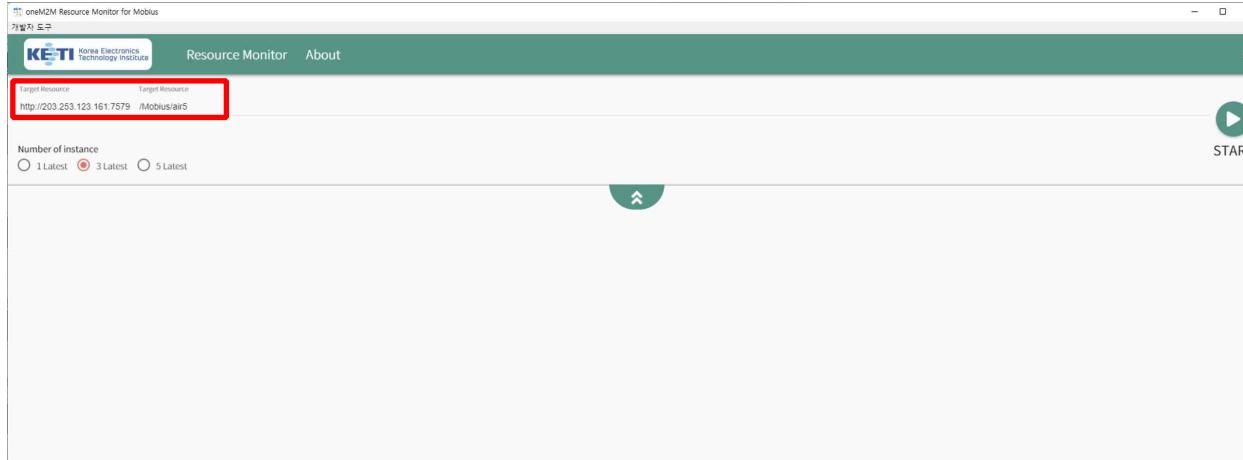
WiFi 연결이 된 이후에는 Red LED가 소등되고, Green LED가 점등되며, 데이터가 발생하여 WiFi를 통해서 Mobius로 업로드 할 때 Orange LED가 점멸한다.

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

6. nCube-Air 데이터 검색 및 LED 제어

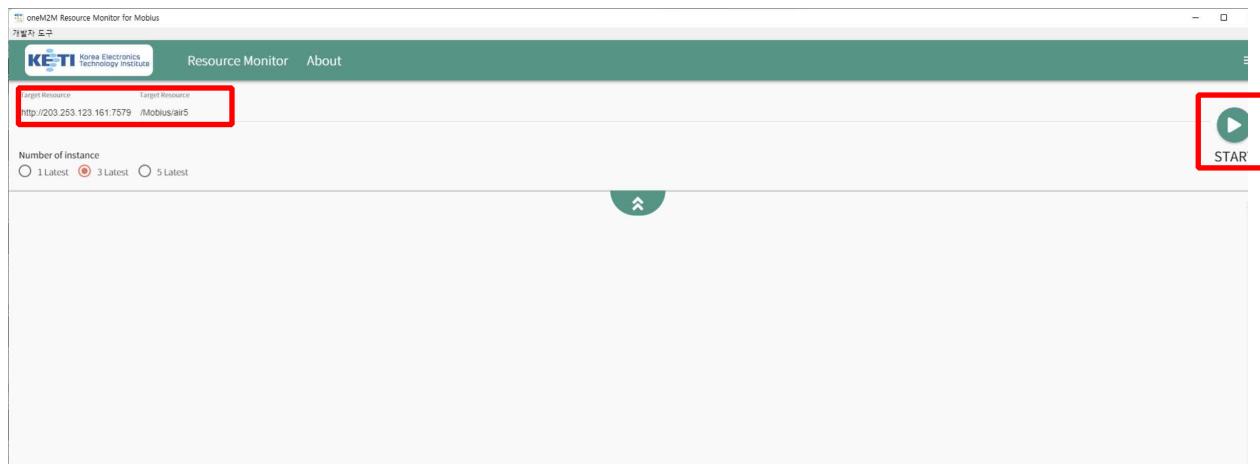
이번 장에서는 Mobius 리소스 모니터를 통해 nCube-Air 프로젝트에서 생성한 데이터를 조회하고 LED를 제어하는 과정을 살펴본다.

6-1. Mobius 리소스 모니터



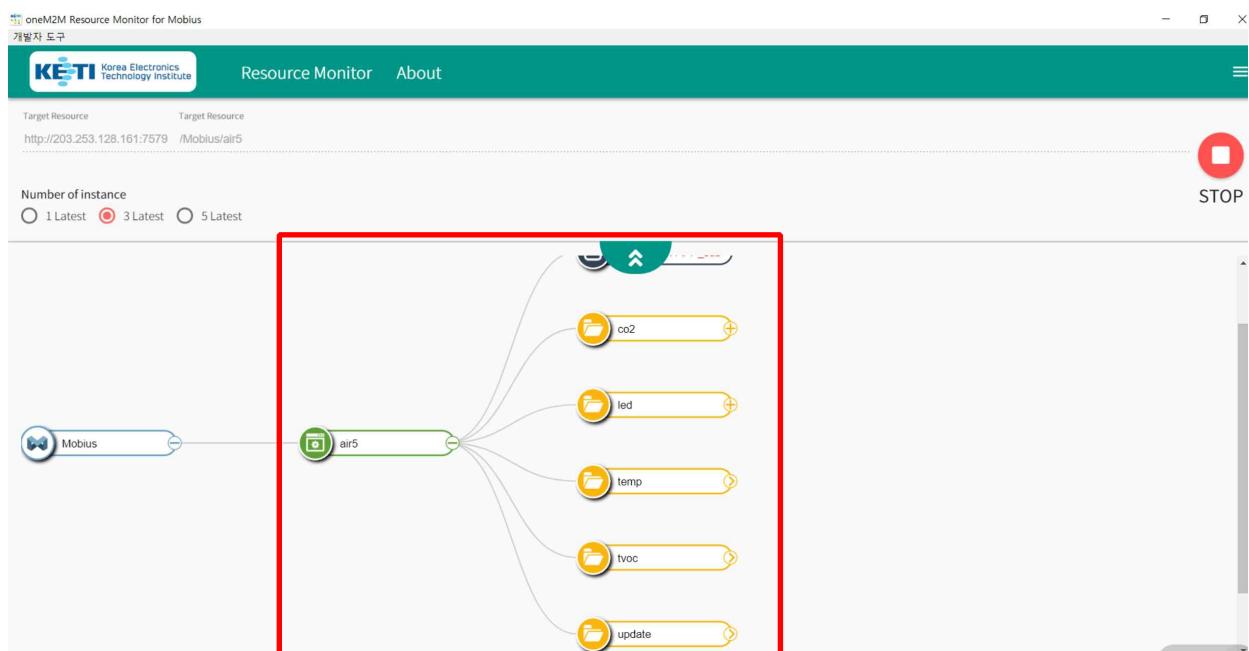
oneM2M Resource Monitor for Mobius는 Mobius에 생성되는 리소스를 실시간으로 보여주고, 리소스의 생성과 삭제 기능을 지원한다.

6-2. nCube-Air 데이터 조회



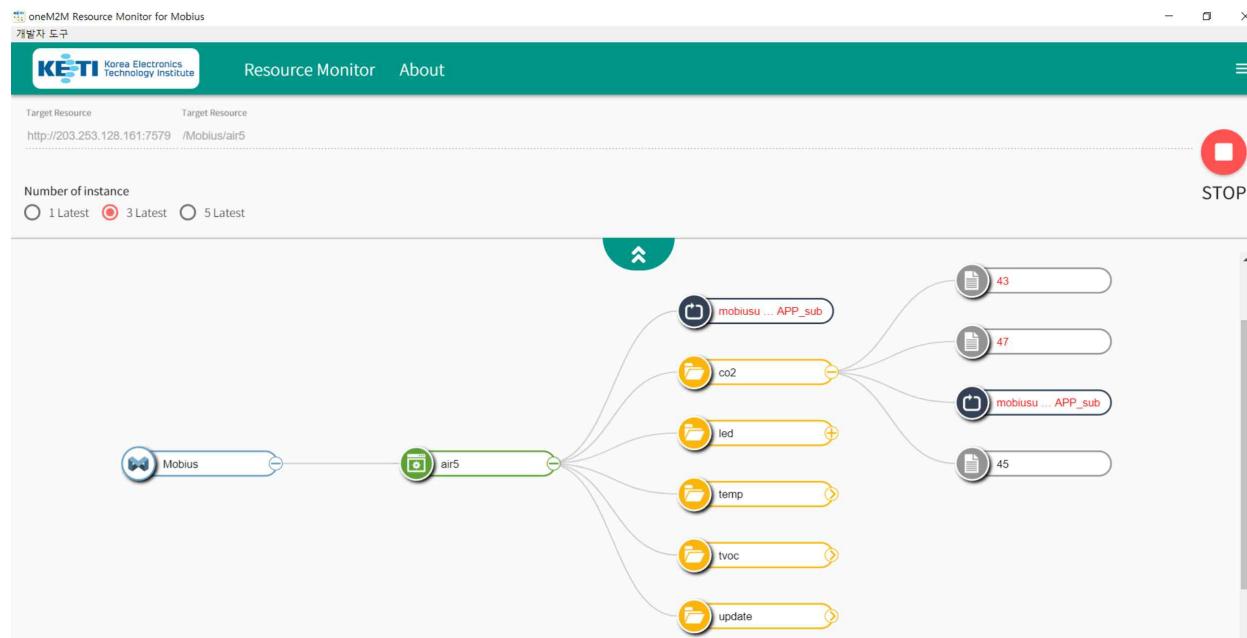
Target Resource에 nCube-Air 소스 코드에서 설정한 **AE_NAME**의 값(page 27 참고)을 입력하고 Start 버튼을 클릭한다.

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

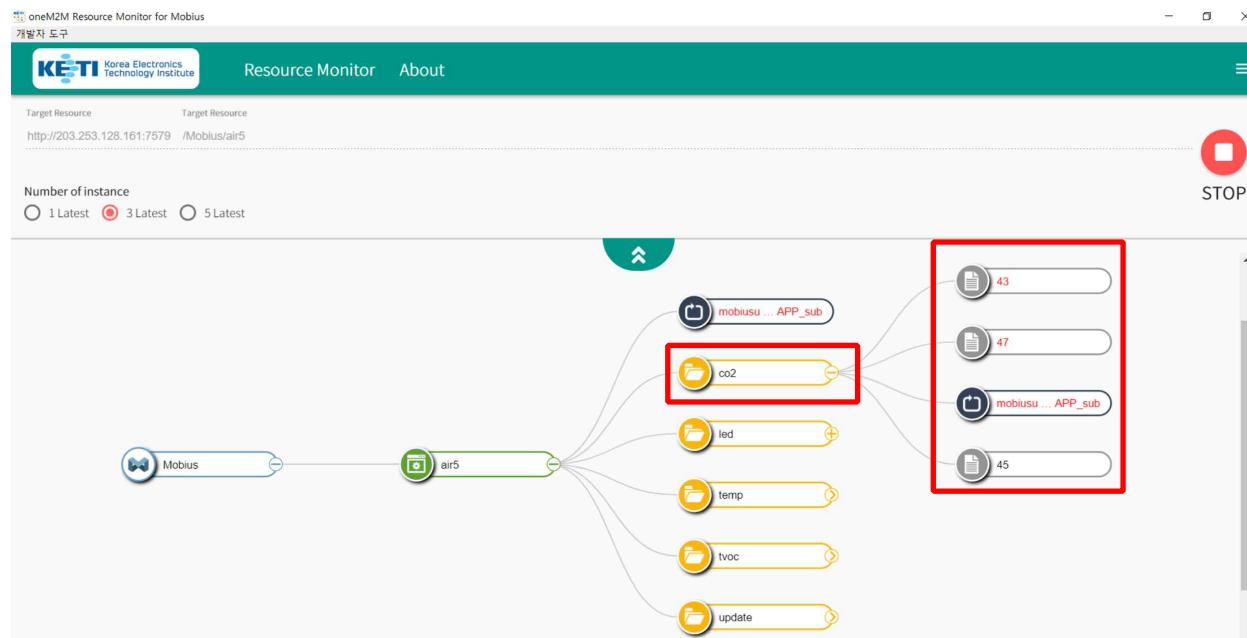


위의 그림과 같이 nCube-Air가 Mobius에 생성한 리소스 구조를 볼 수 있다.

nCube-Thyme-Arduino 가이드

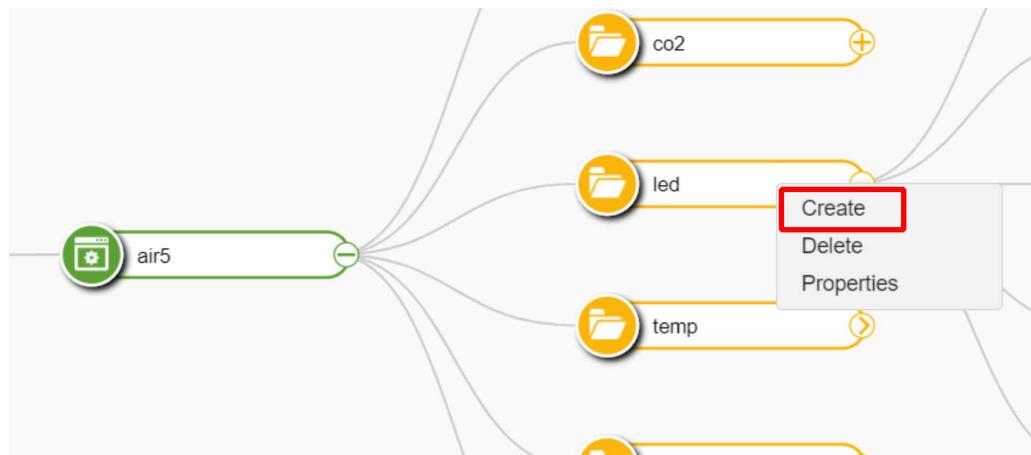


노란색으로 표시된 container(cnt) 리소스 아이콘 오른쪽의 + 버튼을 클릭하면 cnt 리소스 하위의 cin 리소스 데이터를 볼 수 있다.

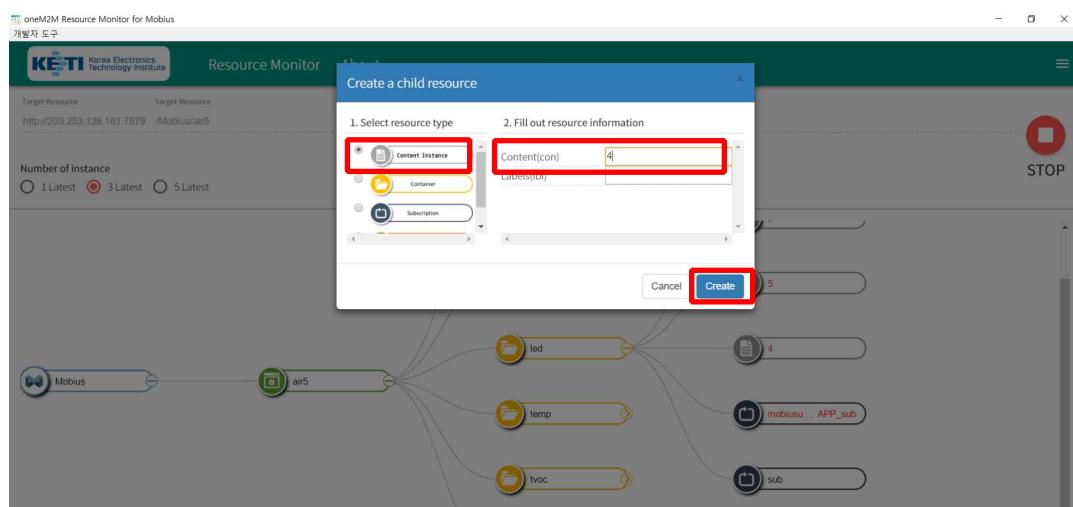


co2, 리소스에 생성된 데이터와 실시간으로 생성되는 데이터를 확인할 수 있다.

6-3. 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.

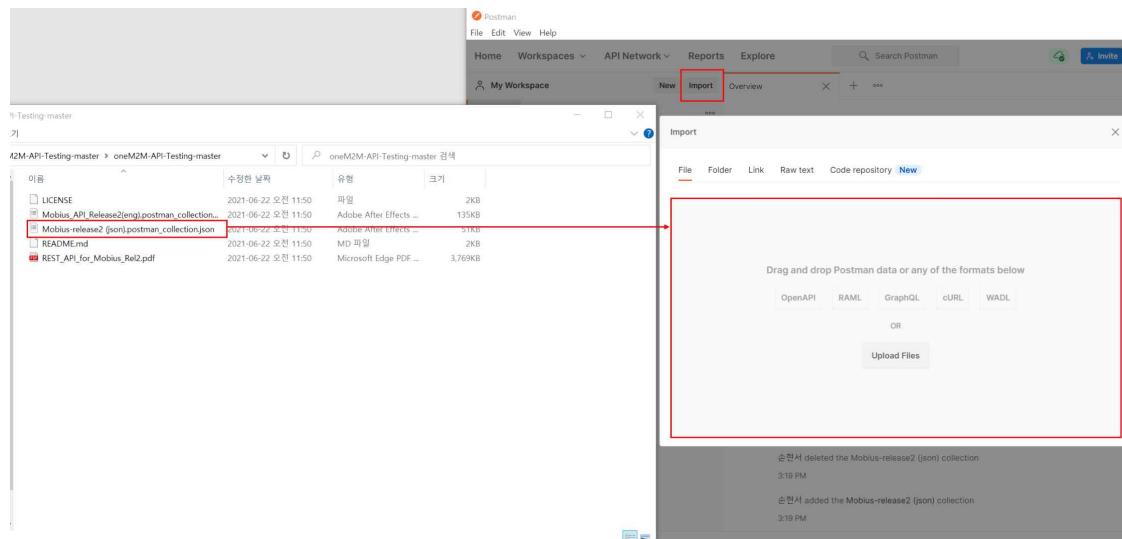
7. oneM2M API를 통한 nCube-Air 데이터 검색

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.

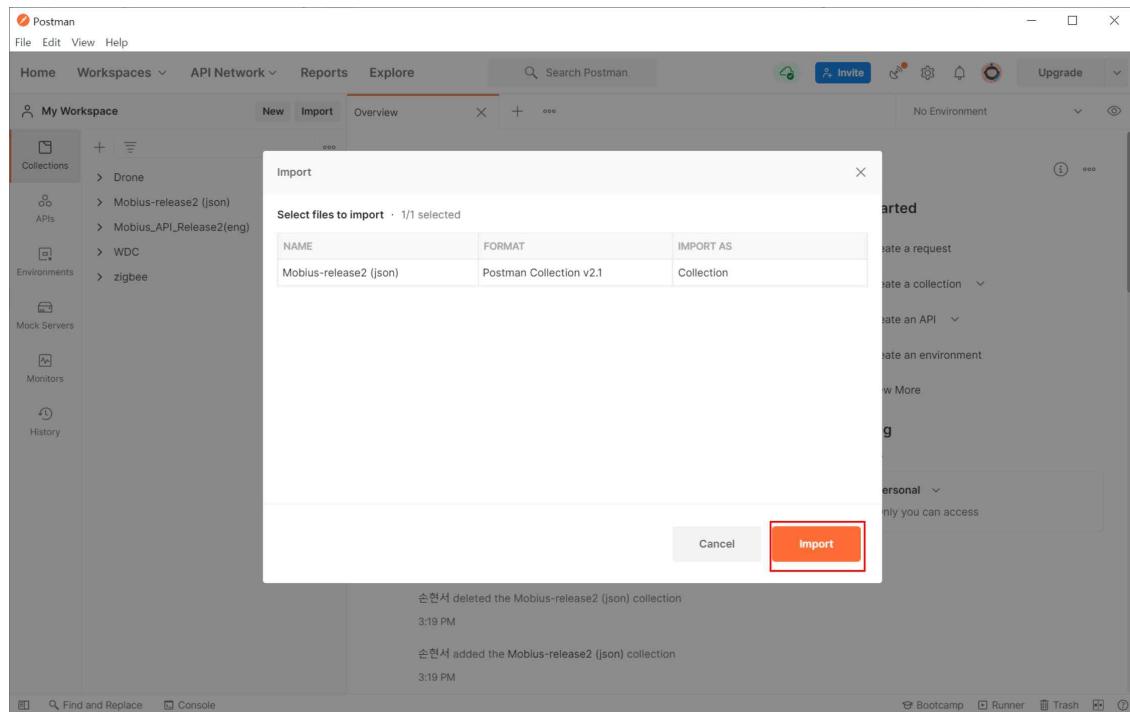
7-1. Postman에 oneM2M-API-Testing import하기

The screenshot shows a GitHub repository named 'IoTKETI / oneM2M-API-Testing'. The repository has 1 branch and 1 tag. In the sidebar, there is a 'Code' dropdown menu with a red arrow pointing to the 'Download ZIP' button. Other options in the sidebar include 'Clone' (HTTPS, SSH, GitHub CLI), 'Open with GitHub Desktop', and 'Open with Visual Studio'.

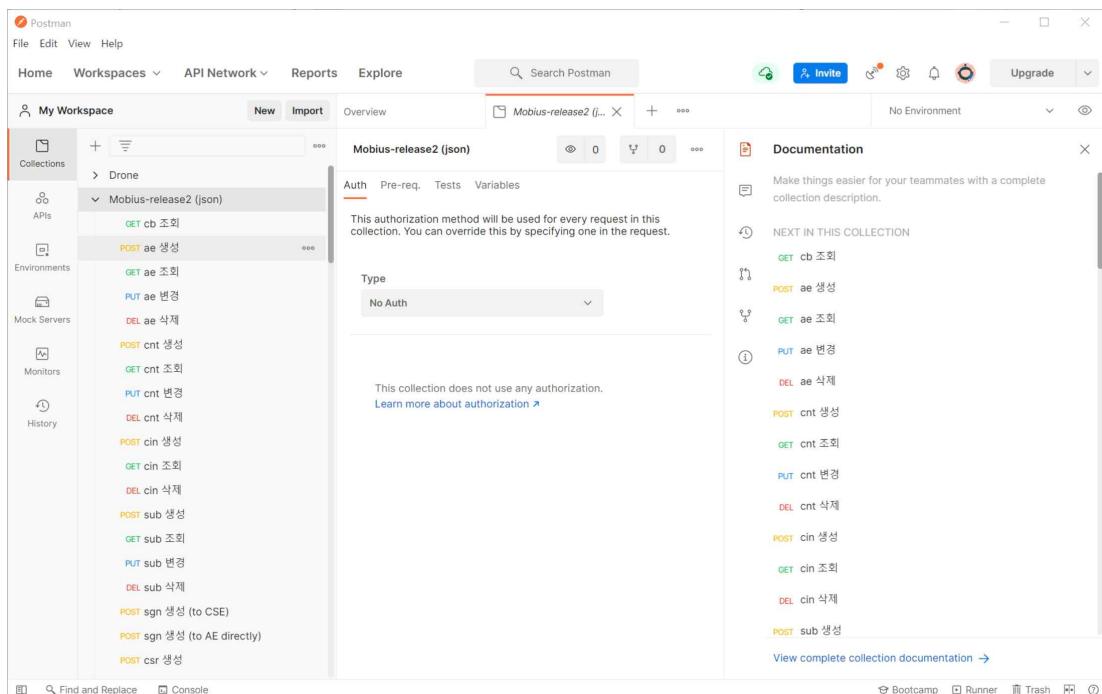
<https://github.com/IoTKETI/oneM2M-API-Testing>에 접속하여 API Collections을 다운로드 받고 압축을 해제한다.



nCube-Thyme-Arduino 가이드



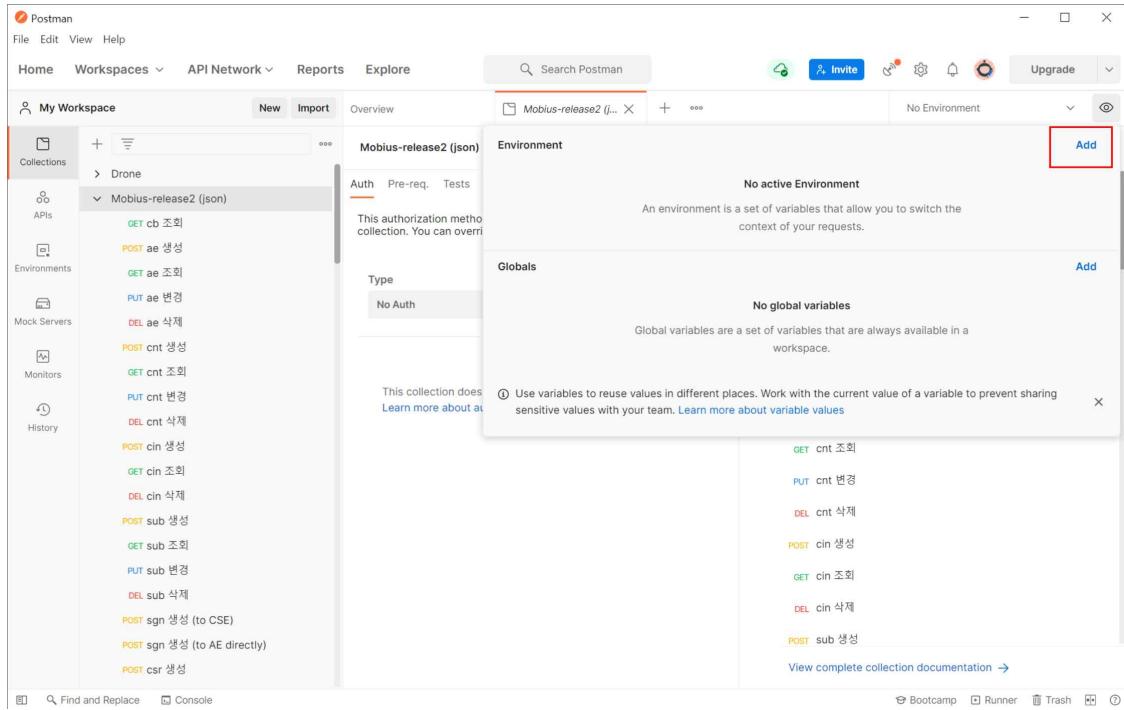
압축을 해제한 폴더에서 Mobius-release2 (json).postman_collection.json 파일을 postman app에 드래그 하여 import 한다.



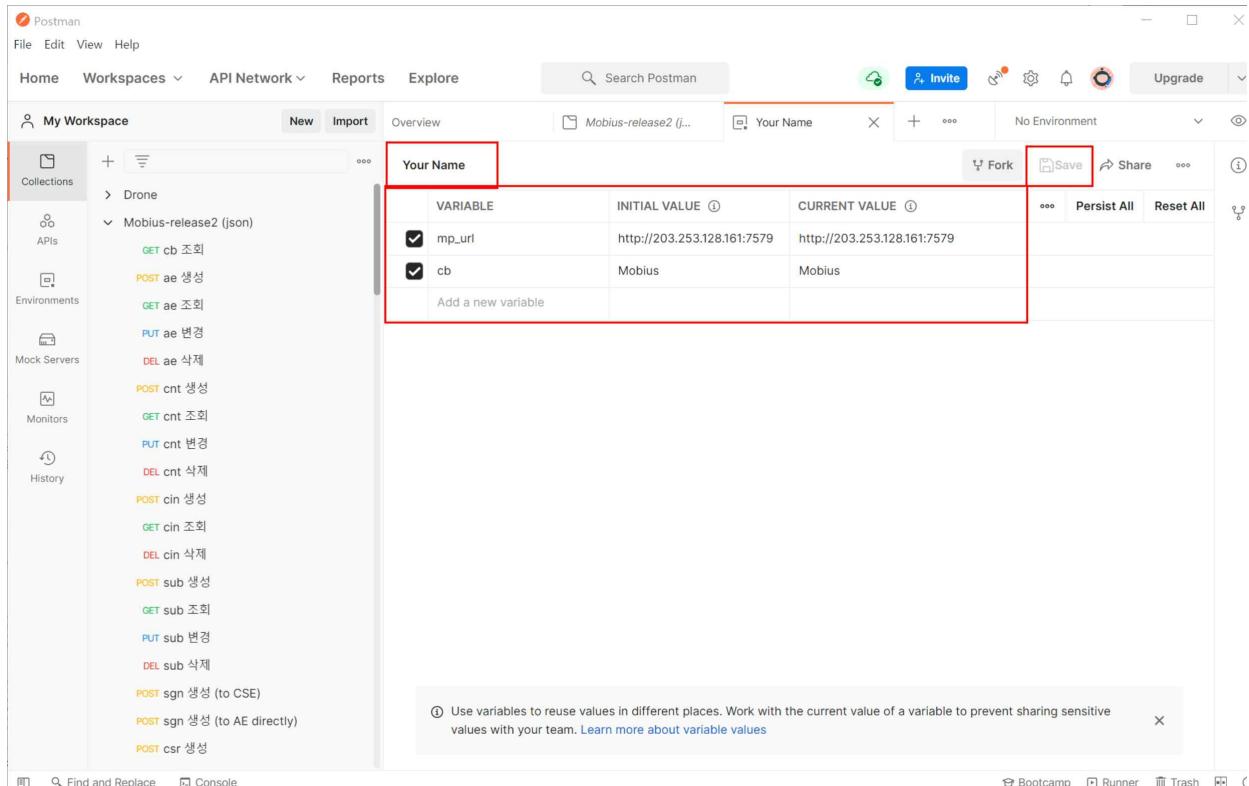
Postman의 메인 화면 왼쪽의 Collections탭을 클릭하면 Mobius release2 (json) API들이 성공적으로 추가된것을 확인할 수 있다.

nCube-Thyme-Arduino 가이드

7-2. 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.

nCube-Thyme-Arduino 가이드

7-3. Cin 조회하기

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- API:** Mobius-release2 (json)
- Request Type:** GET
- Path:** {{mp_url}}/{{cb}}/air1/co2/la
- Headers:** Accept: application/json, X-M2M-Ri: 12345, X-M2M-Origin: SOrigin
- Body:** JSON response (Pretty) showing a single object with a 'con' field.

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.