



OCEAN

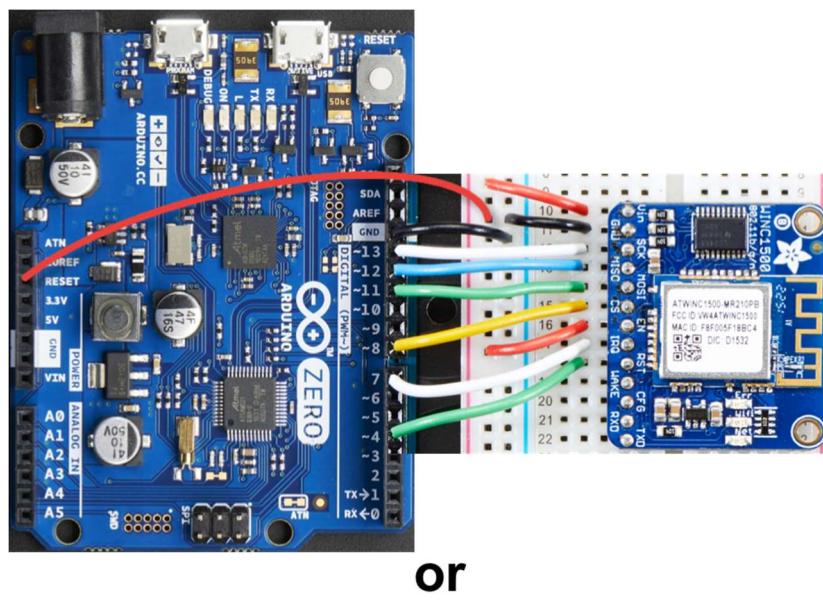
nCube-Mint

Ver.1.0

nCube-Mint

nCube-Mint 는 oneM2M AE 를 아두이노를 이용하여 구현한 버전이다. 개발 언어는 Arduino IDE 에서 지원하는 C, C++이다. nCube-Mint 는 nCube:thyme 과 기능적으로 동일하게 동작하며, MQTT 프로토콜 통신을 사용한다.

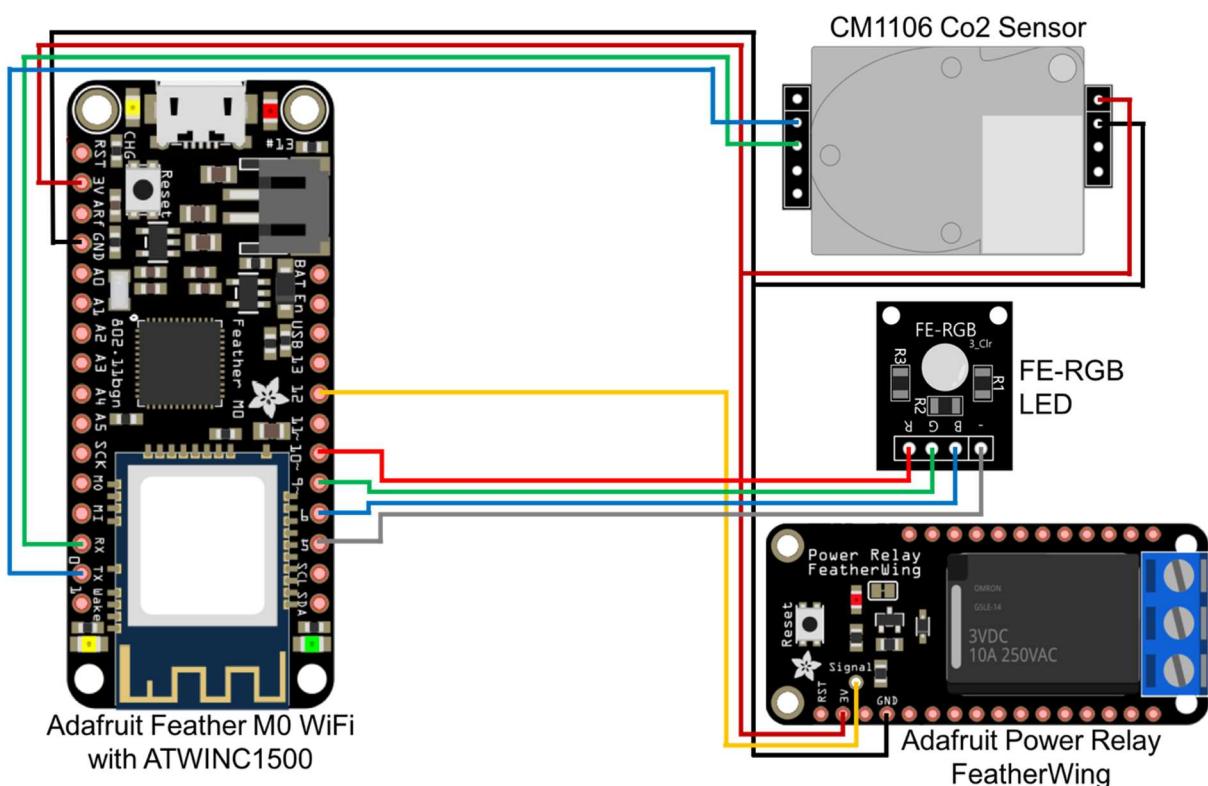
nCube-Mint 의 하드웨어는 cortex-M0 를 기반으로 작동하는 Arduino Zero 를 타겟으로 하며, 더불어 무선인터넷을 통한 MQTT 통신을 사용하기 위해 WiFi 모듈을 필요로 한다. 본 문서는 Arduino Zero 의 Cortex-M0 와 WiFi 모듈을 하나의 모듈에 모두 포함하고 있는 Adafruit 사의 Feather M0 를 기반으로 제작하였다.



본 문서는 Mobius IoT Platform 에 생활속에서 수집될 수 있는 다양한 정보를 업로드하고, 사용자의 동작 제어를 제공하는 nCube-Mint 보드를 제작하는 과정의 설명을 목적으로 하며, nCube-Mint 를 활용한 Co2 농도의 측정 및 업로드, LED 와 Relay 제어를 제공하는 예제를 통해 설명한다.

nCube-Mint 제작 가이드

본 문서의 내용을 바탕으로 nCube-Mint 에 다양한 센서를 연결하여 생활 속 곳곳에서 온도, 습도, Co2 농도, 적외선을 이용한 특정 공간에서의 사람의 존재 유무나 이동방향 등을 탐지할 수 있는 저전력 IoT Device 제작이 가능하다.



위의 사진은 완성되어 동작하는 nCube-Mint 의 사진과, 3 개의 센서와 연결한 그림이다.

목차

1	하드웨어	6
1 - 1	Adafruit Feather M0	6
1 - 2	CO ₂ 센서	7
1 - 3	RGB-LED	7
1 - 4	Adafruit Power Relay FeatherWing.....	7
2	환경설정	8
2 - 1	Arduino IDE 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

nCube-Mint 제작 가이드

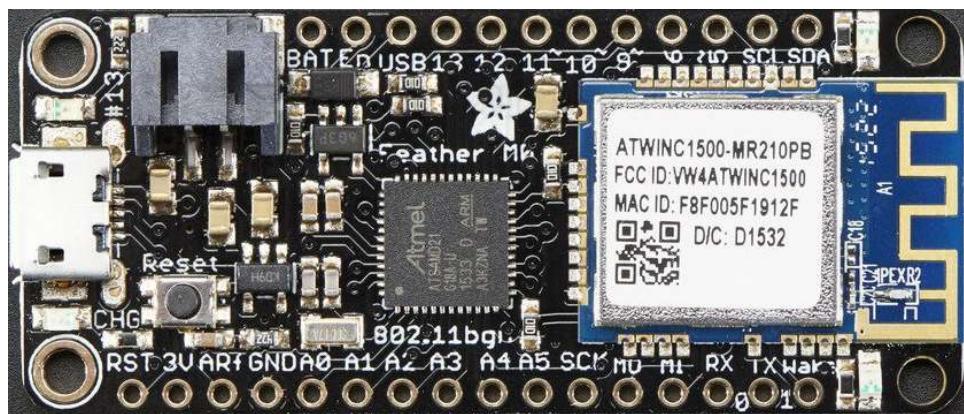
4 센서별 Cube-Mint 응용 개발	24
4 - 1 nCube-Mint + Co2 Sensor.....	24
4 - 1 - 1 Co2 센서 연결.....	24
4 - 1 - 2 TasCO2 라이브러리 소개.....	25
4 - 1 - 3 nCube-Mint + Co2 Sensor 스케치.....	26
4 - 1 - 4 nCube-Mint Co2 동작과 데이터 조회.....	28
4 - 2 RGB-LED 연결	30
4 - 2 - 1 RGB-LED 연결	30
4 - 2 - 2 nCube-Mint + LED 스케치	31
4 - 2 - 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 하드웨어

1장에서는 nCube-Mint 를 구성하기 위한 하드웨어에 대해 간단히 설명한다.

기본이 되는 Adafruit Feather M0 에 대한 소개를 시작으로, Co2 센서, RGB-LED 센서, Adafruit Power Relay FeatherWing 에 대해 소개한다.

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, 802.11bgn 을 지원하는 Atmel WINC1500 WiFi 칩을 포함하는 등의 특징을 갖는다.

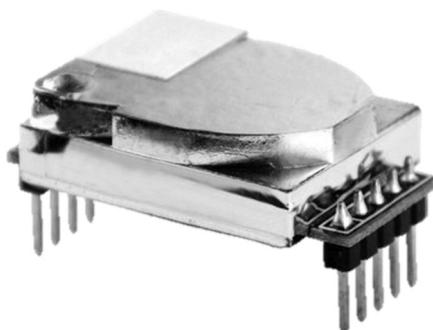
Adafruit Freather M0 는 저전력 설계가 적용되어 IoT Device 로 사용되기에 적절하다.

WiFi 칩은 전력관리칩을 포함하고 있으며 12mA 의 전력 소비량을 가지고, Cortex M0 칩은 10mA 의 전력 소비량을 가진다.

MicroUSB 를 통한 전원 입력과, 3.7V 리튬 폴리머 배터리를 이용한 전원 입력이 모두 가능하며, 배터리가 연결된 상태에서 MicroUSB 를 통한 전원이 입력되면 배터리를 충전한다.

또한 위의 보드사진에서 좌측 하단에 위치한 3V 핀을 통해서 최대 600mA 의 전력 출력을 지원한다.

1 - 2 Co₂ 센서



CM1106 Co₂ 센서는 대기중의 Co₂ 농도를 측정하여 UART 포트를 이용해 Arduino로 업로한다. 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 3색 LED 보드는 각 R, G, B 핀의 입력에 따라 Red, Green, Blue LED를 동작시킨다. 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.

1 - 4 Adafruit Power Relay FeatherWing



Adafruit Power Relay FeatherWing 보드는 3V의 입력과 Digital 입력 신호를 받아, 2개의 외부 전원 출력부에 선택적으로 전류가 흐르도록 한다. 외부 전원 출력부에 TV, Radio 등의 가전제품의 전원 제어가 가능하다.

2 환경설정

2 장에서는 nCube-Mint 보드를 제작하기 위한 환경설정에 대해 설명한다.

Arduino IDE의 설치, Adafruit Feather M0 보드매니저 설치, 간단한 예제를 통한 Adafruit Feather M0 테스트의 순서로 소개하겠다.

단, 본 문서는 MacOS를 기반으로 제작되었으며, Windows OS도 기본적인 과정은 같지만 추가적인 작업이 필요한 경우 별도로 표시하였다.

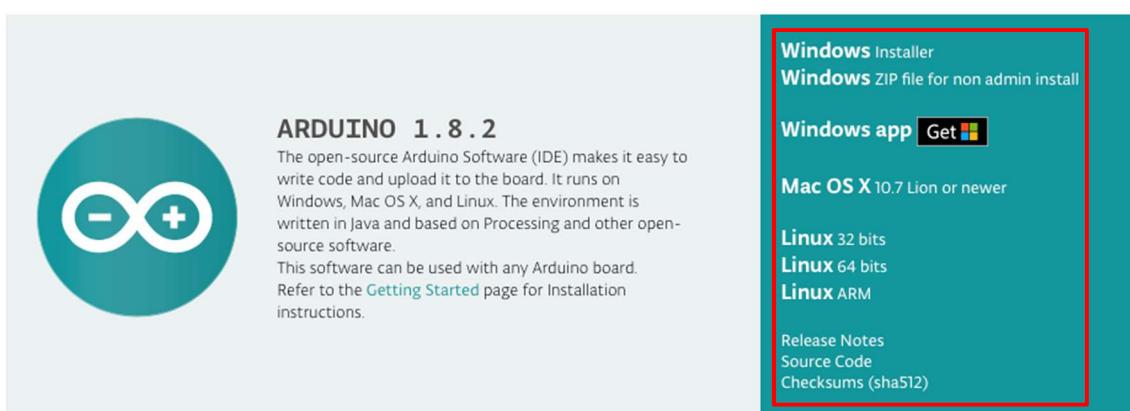
2 - 1 Arduino IDE 설치



<https://www.arduino.cc/en/Main/Software>

위의 URL을 통해 Arduino 홈페이지의 SOFTWARE 탭에 접속한다. 별도의 로그인이나 회원가입은 필요로 하지 않는다.

Download the Arduino IDE



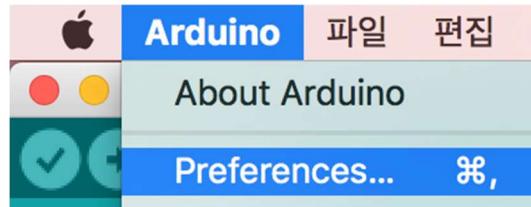
아래로 스크롤 하여 Download the Arduino IDE 항목의 우측에서 사용하는 OS에 맞는 Arduino IDE를 다운로드 받은 후 설치한다.

위에서 소개하는 방법 이외의 경로를 통해 Arduino IDE를 설치해도 무관하다.

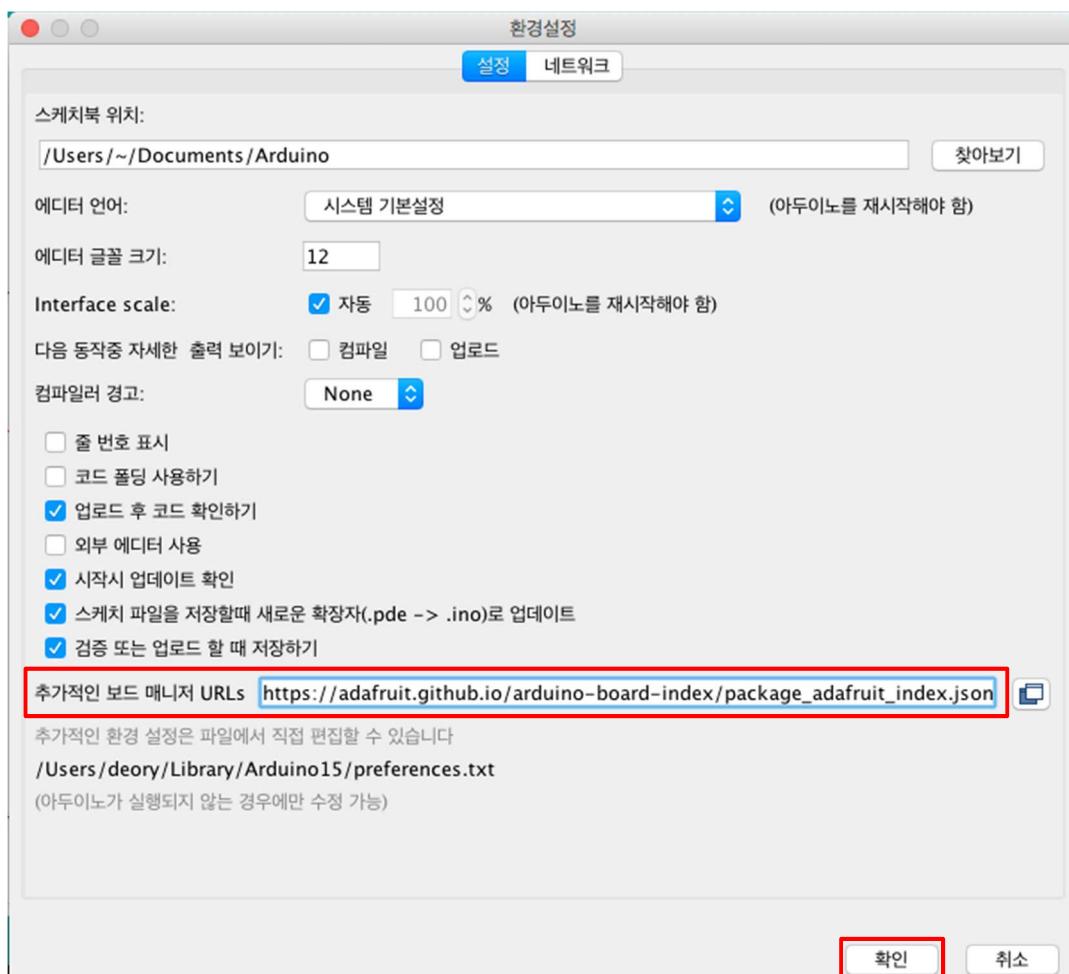
단, Arduino IDE 버전은 v1.6.4 또는 이후 버전을 권장한다.

2 - 2 Arduino IDE 환경설정

Arduino IDE 를 이용해 nCube-Mint 스케치를 컴파일하고, Adafruit Feather M0 로 업로드하기 위한 환경설정을 진행한다.



Arduino IDE 를 실행시킨 후 환경설정에 들어간다.

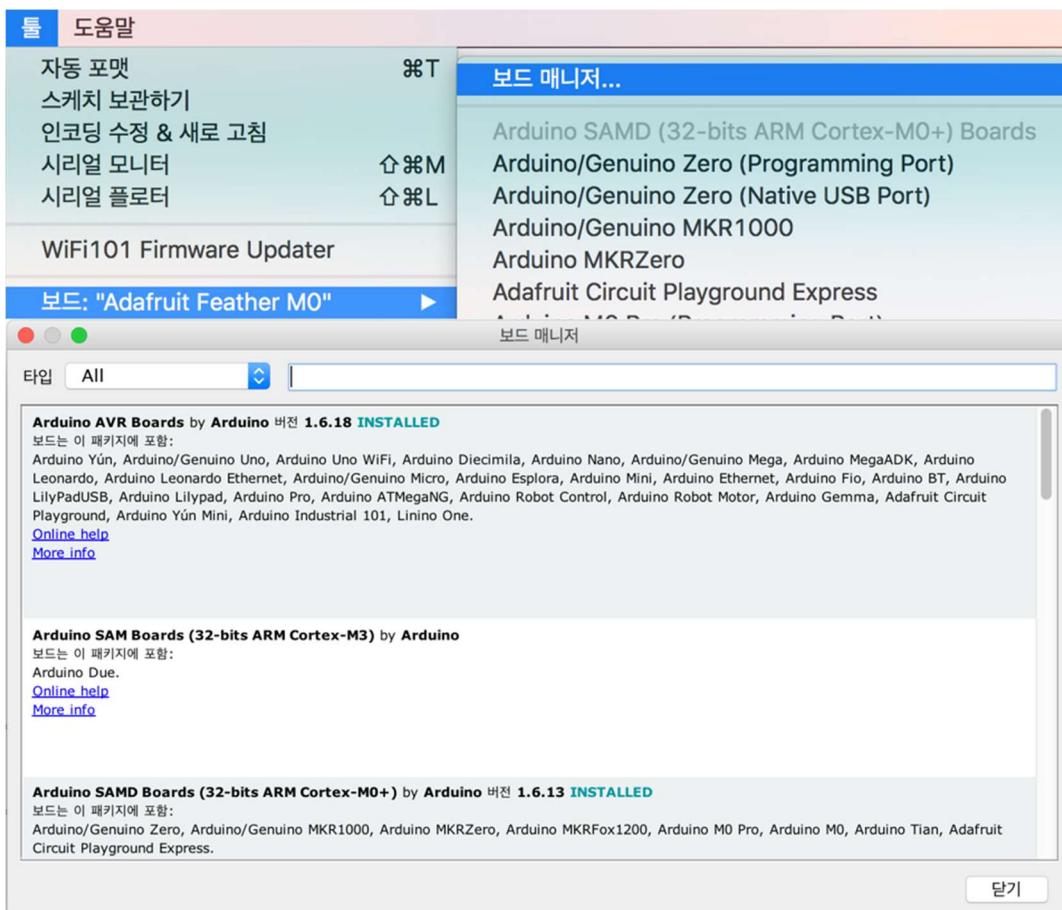


Additional Boards Manager URLs 항목에

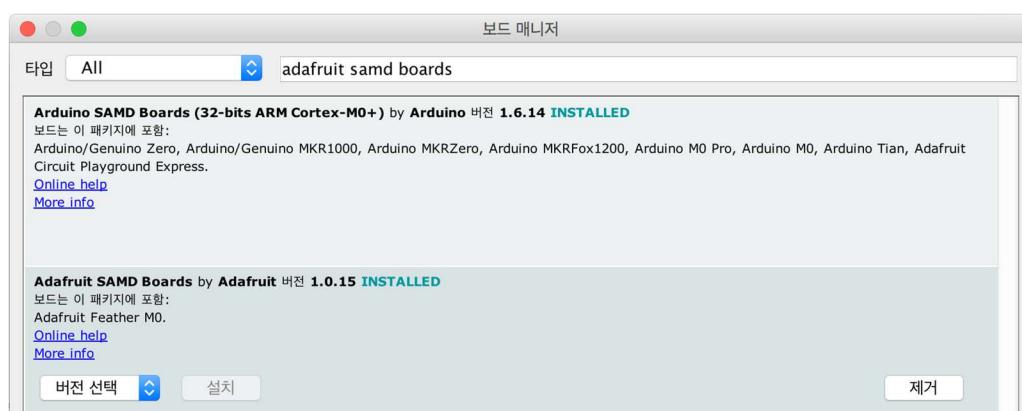
https://adafruit.github.io/arduino-board-index/package_adafruit_index.json

위의 URL 을 추가하고, 확인 버튼을 눌러 빠져나온다.

nCube-Mint 제작 가이드



Tools 탭을 클릭하고 Board 탭의 하위메뉴에서 Boards Manager... 을 클릭하여 보드 매니저를 실행시킨다.



adafruit samd boards 를 검색하여 나오는 두개의 패키지를 모두 설치한다.

nCube-Mint 제작 가이드

툴	도움말	Arduino Boards
자동 포맷	⌘T	Arduino Pro or Pro Mini
스케치 보관하기		Arduino NG or older
인코딩 수정 & 새로 고침		Arduino Robot Control
시리얼 모니터	⇧⌘M	Arduino Robot Motor
시리얼 플로터	⇧⌘L	Arduino Gemma
WiFi101 Firmware Updater		Adafruit Circuit Playground
보드: "Adafruit Feather M0"	▶	Arduino Yún Mini
포트	▶	Arduino Industrial 101
보드 정보 얻기		Linino One
프로그램어: "AVRISP mkII"	▶	Arduino Uno WiFi
부트로더 굽기		Adafruit SAMD (32-bits ARM Cortex-M0+) Boards
		✓ Adafruit Feather M0
		Adafruit Feather M0 Express

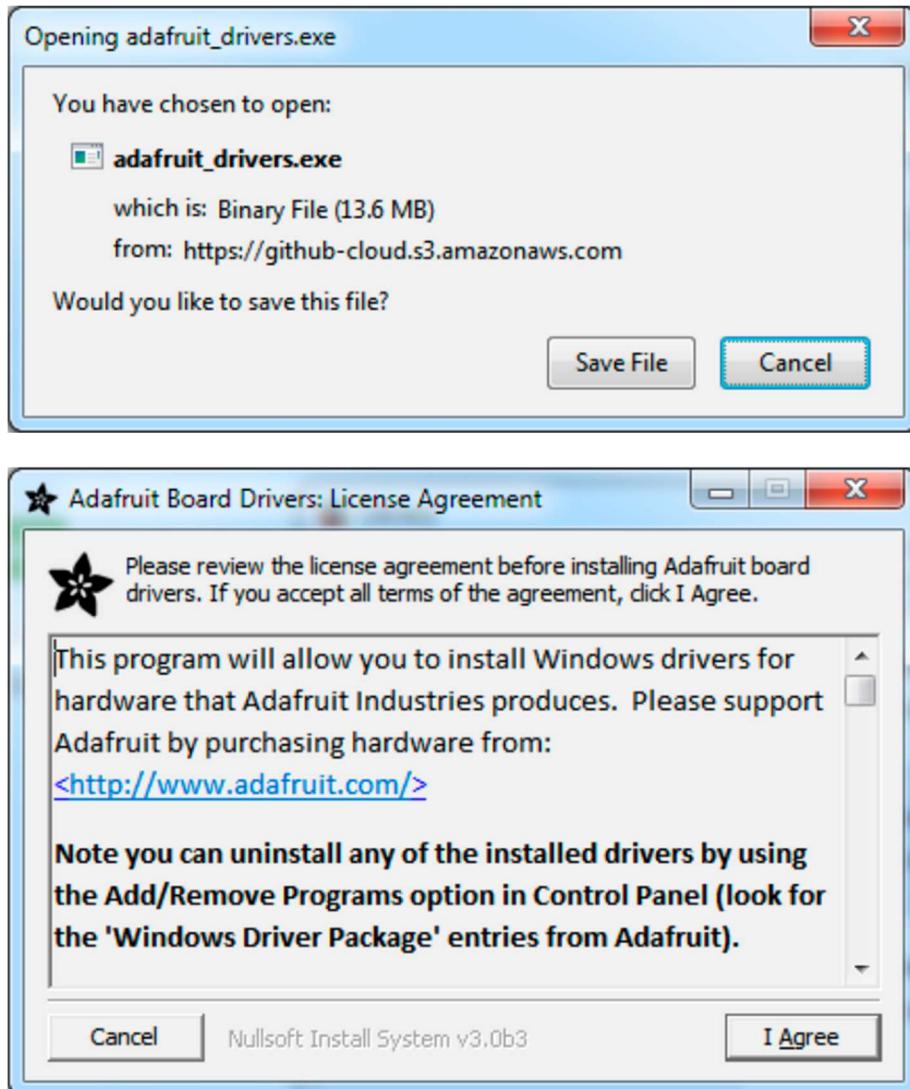
마지막으로, Tools 탭에서 Board 항목 아래에 Adafruit Feather M0 항목이 보이고, 선택 가능한 것을 확인함으로써 Arduino IDE Setup 을 마친다.

* Adafruit Windows Driver 설치

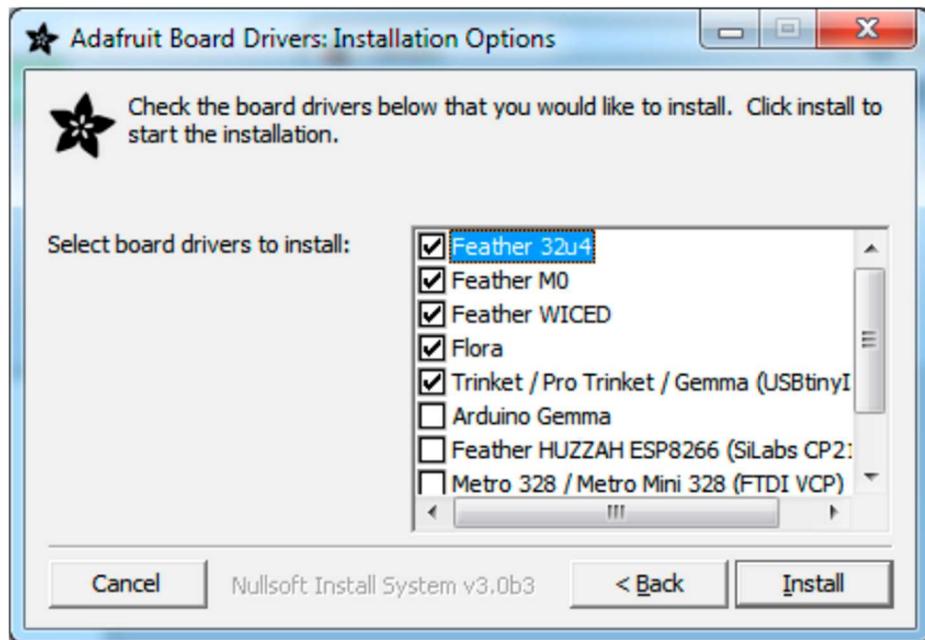
윈도우 운영체제 환경에서는 추가적으로 Adafruit WindowsDriver 를 설치해야한다.

<http://adafru.it/mai>

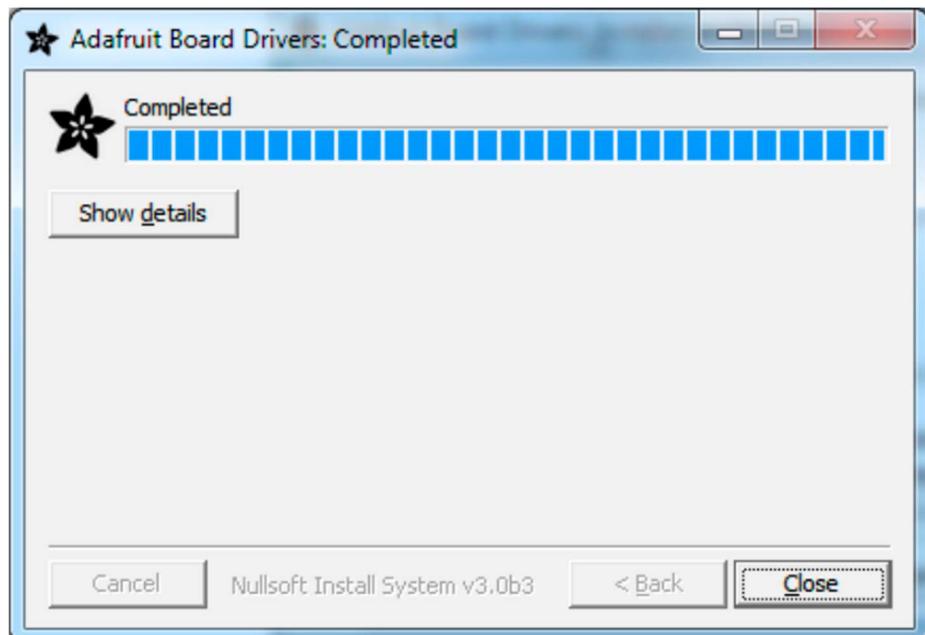
위의 URL 을 통해 adafruit_drivers.exe 파일을 다운로드 받고 실행시킨다.



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

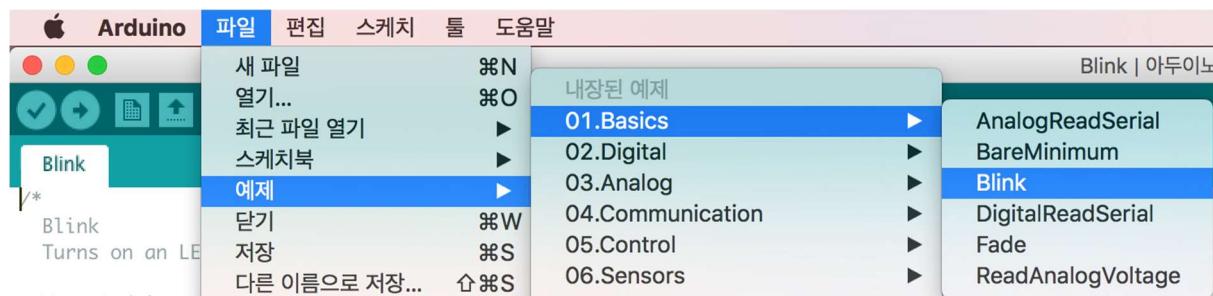


위의 화면에서 드라이버의 설치를 원하는 보드의 종류를 체크할 수 있다. 본 문서의 예제를 위해 Feather M0 항목에 체크한 뒤 Install 버튼을 눌러준다.



윈도우 운영체제를 위한 adafruit drivers 설치가 완료되었다.

2 - 3 Adafruit Feather M0 동작 테스트

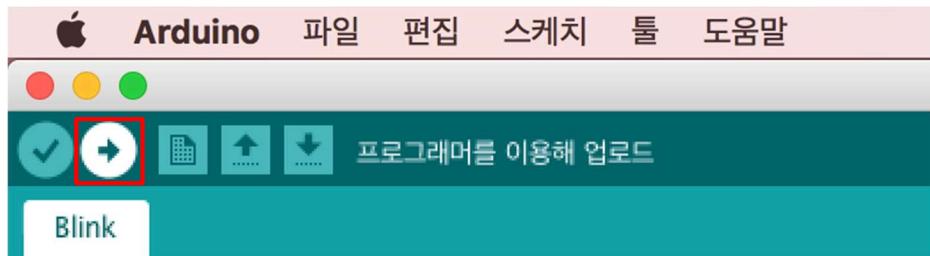


Arduino IDE의 File 탭의 Examples 탭, 01.Basics 탭 아래의 Blink 항목을 클릭하여 Blink 예제를 불러온다.

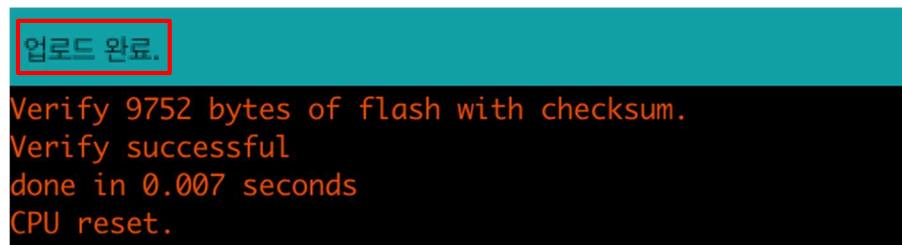


Adafruit Feather M0 와 컴퓨터를 USB-5pin Micro USB 케이블로 연결하고, Tool 탭의 보드 항목과, 포트 항복이 모두 Adafruit Feather M0 로 설정되도록 각각 클릭하여 설정하여준다. 기본적으로 설정이 되어있다면, 넘어가도록 한다.

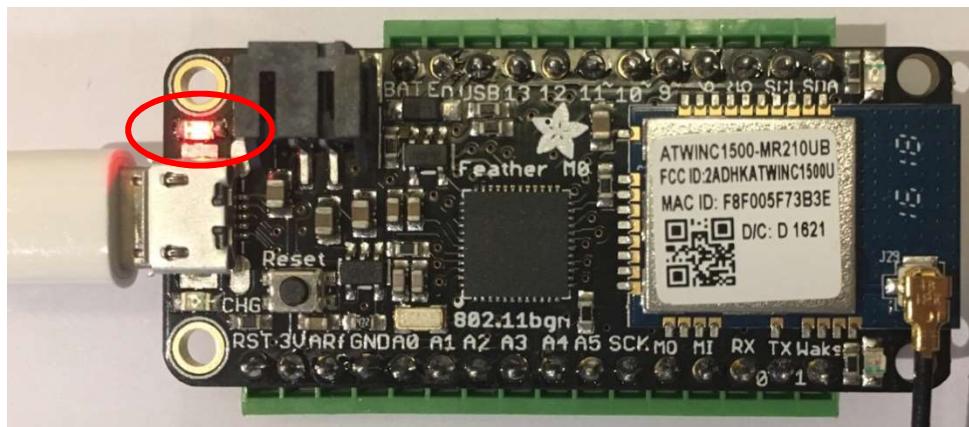
nCube-Mint 제작 가이드



위의 그림에서 하얀색으로 표시된 업로드 버튼을 눌러,
Blink 예제를 컴파일하여 Adafruit Feather M0로 업로드 한다.



업로드가 완료되면 Arduino IDE 하단에 ‘업로드 완료.’라는 글자가 나타난다.



Blink 예제를 다운로드 한 Adafruit Feather M0에 기본으로 내장된 빨간색 LED가 1초 간격으로 점멸하는것을 확인할 수 있다.

이로서 nCube-Mint 보드를 제작하기 위한 환경설정과 Adafruit Feather M0의 연결, 동작 테스트가 모두 완료되었다.

3 nCube-Mint 개발 환경

이번 장에서는 nCube-Mint 를 구성하기 위해 필수적으로 요구되는 라이브러리를 다운로드받고, OneM2MClient 라이브러리의 예제파일로 제공되는 nCube-Mint 스케치를 이용하여 Adafruit Feather M0 를 이용한 nCube-Mint 보드를 구성한다.

3 - 1 nCube-Mint 라이브러리 다운로드

<https://github.com/IoTKETI/nCube-Thyme-Arduino>

No description or website provided.

onem2m iot device arduino-zero ota Manage topics

2 commits 1 branch 0 releases 1 contributor BSD-3-Clause

Branch: master New pull request Create new file Upload files Find file Clone or download

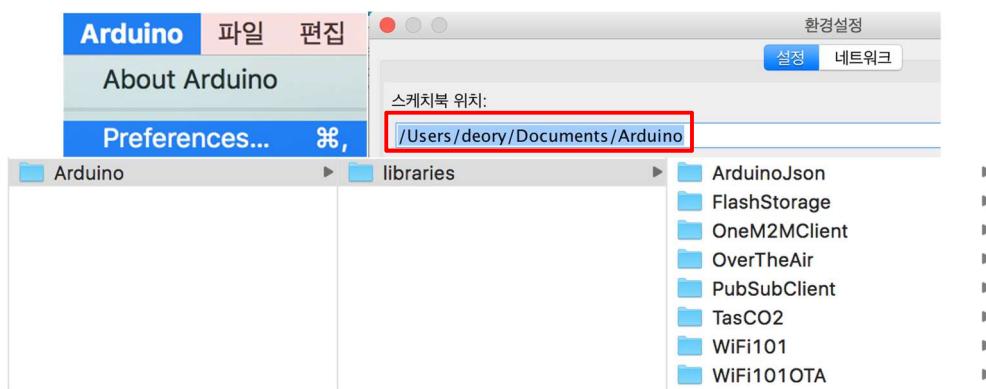
coffeenan Initial commit Latest commit 9d79d8b 4 hours ago

nCube-Mint-libraries Initial commit 4 hours ago

LICENSE Initial commit 7 days ago

위의 nCube-Mint 를 다운로드 받고, 압축을 풀어준다.

libraries-nCube-Mint 내부의 라이브러리 디렉토리들은 Arduino IDE 의 환경설정내의 스케치북 위치 아래의 libraries 디렉토리 내에 위치시켜야 한다.



libraries-nCube-Mint 에 포함된 라이브러리들은 위의 그림에서 확인할 수 있으며, 해당 위치로 옮긴 후에 Arduino IDE 가 불러올 수 있도록 Arduino IDE 를 재시작 시켜주어야 한다.

3 - 2 OneM2MClient 라이브러리 소개

OneM2MClient 라이브러리는 nCube-Mint 보드를 제작하기 위해 사용되는 가장 기본적인 라이브러리로, nCube-Mint 의 핵심적인 동작들을 함수로 사용할 수 있도록 지원한다. 객체를 만들어 아래에서 소개하는 변수, 구조체에 접근하고, 함수를 호출한다.

resource

resource_count

nCube-Mint 에서는 OneM2MClient 라이브러리의 *resource* 구조체를 이용하여 AE, Cnt, Cin 정보를 저장하며, *resource_count* 는 구조체 배열의 크기를 저장하기 위해 사용한다.

```
createAE(String rqi, int index, String api)  
createCnt(String rqi, int index)  
deleteSub(String rqi, int index)  
createSub(String rqi, int index)  
createCin(String rqi, int index)
```

위의 함수들을 호출하여 AE 생성, Cnt 생성, Subscription 제거 및 생성, Cin 생성을 한다. 매개변수로서 기본적으로 Request ID, 해당하는 동작을 위한 정보를 담고 있는 *resource* 구조체 배열의 *index* 를 주어야 하며, *createAE* 의 경우 추가로 Application ID 값을 매개변수로 전달해 준다.

response(body_buff)

LED 나 Relay 제어 패킷을 처리하고난 뒤에, 응답 패킷을 보내기 위한 함수이다. 매개변수로 응답 패킷의 페이로드 데이터를 준다한다.

setCallback(resp_callback, noti_callback)

response 패킷이나, notification 패킷이 왔을 때, 동작 할 callback 함수를 지정하는 함수이다. 매개변수는 좌측 부터 순서대로 response callback 함수, notification callback 함수의 이름으로 준다.

begin()

Flash 메모리에 저장된 WiFi SSID, Password 값이 있는 경우에, 해당하는 정보로 WiFi 연결을 시도하며, 저장된 값이 없는 경우에 Adafruit Feather M0 를 WiFi Access Point 로 동작하게끔 하여 접속하고자 하는 WiFi 의 SSID 와 Password 를 입력할 수 있도록 한다.

chkInitProvision()

chkInitProvision2()

WiFi reset 버튼이 짧게 1회 클리된 경우에 Flash 메모리에 저장된 WiFi SSID, Password 값을 시리얼 모니터를 통해 출력해 주며,
길게 1회 클릭된 경우에 Flash 메모리에 저장된 WiFi SSID, Password 값을 지우고, Adafruit Fetather M0 를 WiFi Access Point 로 동작하도록 한다.

*chkInitProvision()*와 *chkInitProvision2()*는 Flash 메모리의 값을 지우기 위해 지속해야 하는 버튼 클릭 시간의 차이를 가지며, 상대적으로 각각 길고, 짧게 설정하였다.

chkConnect()

WiFi 연결이 되어있는지 확인하고, MQTT 연결이 되어있지 않은 경우에 MQTT Subscript 를 수행하며 True 값을 반환한다.

WiFi 연결이 되어있지 않을 경우에는, WiFi 제접속을 시도하며 False 값을 반환한다.

getAeid()

resource 구조체에 저장되어 있는 AE-ID 를 반환해 주는 함수이다.

3 - 3 nCube-Mint 불러오기

nCube-Mint 예제를 이용하여 IoT Platform과 연결을 위한 설정과, 필요한 리소스 구조를 만드는 작업을 진행한다.



Arduino IDE를 재실행 한 뒤에, File 탭, Examples 탭, OneM2MCClient 탭의 nCube-Mint 를 선택하여 불러온다.

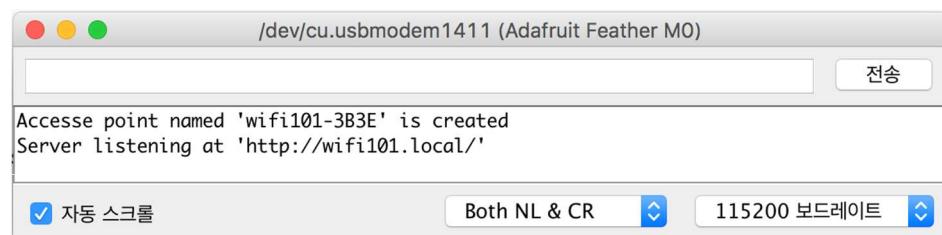
```
const String AE_ID = "your-ae";           // guide: this is same with AE name
const String MQTT_BROKER_IP = "203.253.128.161"; // guide: set IP address of MQTT broker for Mobius as IoT Platform
const uint16_t MQTT_BROKER_PORT = 1883;      // guide: set MQTT port used
OneM2MCClient nCube(MQTT_BROKER_IP, MQTT_BROKER_PORT, AE_ID); // AE-ID
```

MQTT_BROKER_IP 와 MQTT_BROKER_PORT 변수의 값은 Mobius 서버의 MQTT Broker의 IP 와 Port 번호로 바꾸어 준다.

3 - 4 nCube-Mint 스케치 실행

Configuration 과 Resource Build 를 마친 뒤에 업로드 버튼을 눌러 nCube-Mint 스케치를 Adafruit Feather M0 에 업로드하고, Arduino IDE 의 좌측 윗쪽에 위치한

시리얼 모니터  버튼을 클릭하여 시리얼 모니터를 연다.



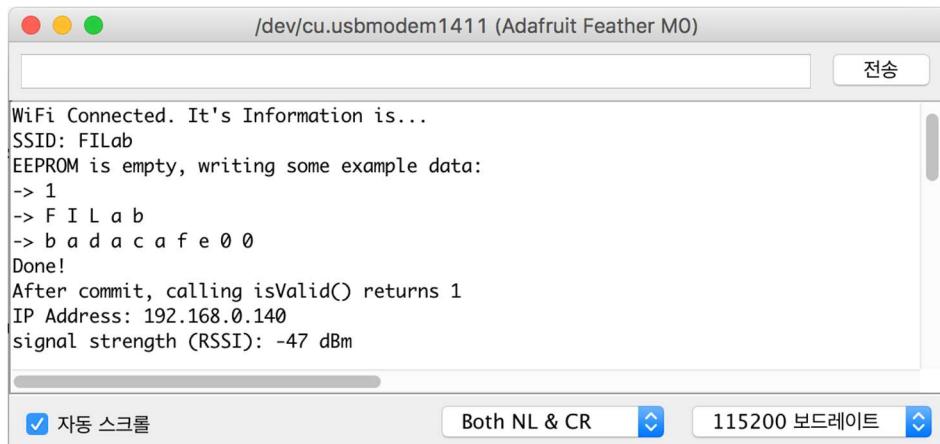
스케치 컴파일과 업로드가 완료된 뒤, 시리얼 모니터를 통해 위와같은 모습을 확인할 수 있다. 스마트폰의 WiFi 기능을 이용해 시리얼 모니터에 표시된 ‘wifi101-XXXX’(Adafruit Feather M0 보드마다 이름이 다름) AP에 접속하면 아래와 같은 화면을 볼 수 있다.



nCube-Mint 제작 가이드

Refresh 버튼을 눌러 접속이 가능한 WiFi AP 목록을 확인할 수 있다.

Network Name 아래에 접속할 AP의 SSID를, Pass phrase 아래에 AP에 해당하는 Password를 입력한 뒤에 Connect 버튼을 눌러 WiFi에 접속한다.



The screenshot shows a terminal window titled "/dev/cu.usbmodem1411 (Adafruit Feather M0)". The window displays the following text:

```
WiFi Connected. It's Information is...
SSID: FILab
EEPROM is empty, writing some example data:
-> 1
-> F I L a b
-> b a d a c a f e 0 0
Done!
After commit, calling isValid() returns 1
IP Address: 192.168.0.140
signal strength (RSSI): -47 dBm
```

At the bottom of the terminal window, there are three buttons: "자동 스크롤" (Auto Scroll) checked, "Both NL & CR", and "115200 보드레이트" (Board Rate).

nCube-Mint 보드가 WiFi AP에 접속하면 시리얼 모니터에 현재 접속한 AP의 정보와 nCube-Mint 보드의 로컬 IP 주소, WiFi 신호의 강도를 보여준다.



The screenshot shows a terminal window titled "/dev/cu.usbmodem1421 (Adafruit Feather M0)". The window displays the following text:

```
{"rsc":4004,"rqi":"m1R6SrqB","pc":{"m2m:dbg":"resource does not exist"}},0
Receive published data
/onem2m/resp/your-ae/mobius-yt/json - RESP_TOPIC receive a message.
delete_sub
create_sub
create_sub

{"op":1,"to":"/mobius-yt/your-ae/your-cnt-ctrl?rcn=0","fr":"your-ae","rqi":"VForEVWj","ty":23,"pc":{"m2m:sub":{}}
REQUEST OK!

{"rsc":2001,"rqi":"VForEVWj","pc":{}},0
Receive published data
/onem2m/resp/your-ae/mobius-yt/json - RESP_TOPIC receive a message.
create_sub
create_cin
```

At the bottom of the terminal window, there are three buttons: "자동 스크롤" (Auto Scroll) checked, "Both NL & CR", and "115200 보드레이트" (Board Rate).

이후에는 nCube-Mint 보드가
mint AE 생성 → co2, led, relay Container 생성 → led, relay Subscript 제거 → led, relay
Subscript 생성
작업하는 것을 시리얼 모니터를 통해 확인할 수 있다.

3 - 5 nCube-Mint 기본 Resource 조회

Mobius에 생성된 리소스들을 조회할 수 있는 Resource Monitor가 있다. C#을 이용한 윈도우즈 응용 프로그램 버전과, 웹을 통해 서비스되는 버전이 존재한다. 본 가이드 문서에서는 웹을 통해 서비스되는 Resource Monitor를 사용하여 nCube-Mint 보드가 생성한 리소스를 조회해 본다.

```
void buildResource() {
    // temperally build resource structure into Mobius as oneM2M IoT Platform

    // AE resource
    uint8_t index = 0;
    nCube.resource[index].ty = "2";
    nCube.resource[index].to = "/mobius-yt";
    nCube.resource[index].rn = AE_ID;
    nCube.resource[index++].status = 0;

    // Container resource
    nCube.resource[index].ty = "3";
    nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn;
    nCube.resource[index].rn = "your-cnt";
    nCube.resource[index++].status = 0;

    nCube.resource[index].ty = "3";
    nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn;
    nCube.resource[index].rn = "your-cnt-ctrl";
    nCube.resource[index++].status = 0;

    // Subscription resource
    nCube.resource[index].ty = "23";
    nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn + '/' + nCube.resource[2].rn;
    nCube.resource[index].rn = "your-sub";
    nCube.resource[index++].status = 0;

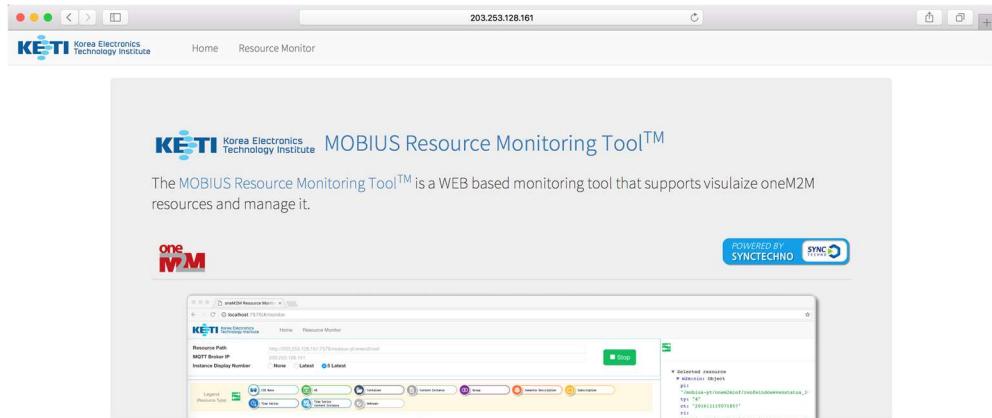
    nCube.resource_count = index;
}
```

nCube-Mint 기본 예제 스케치의 buildResource() 함수를 보면 nCube-Mint가 생성할 리소스에 대해 파악할 수 있다.

nCube-Mint 기본 예제 스케치는
mobius-yt cse 아래에 your-ae를 생성하고,
mint ae 아래에 your-cnt, your-cnt-ctrl을 생성하며,
led cnt 아래에 your-sub를 생성한다.

nCube-Mint 제작 가이드

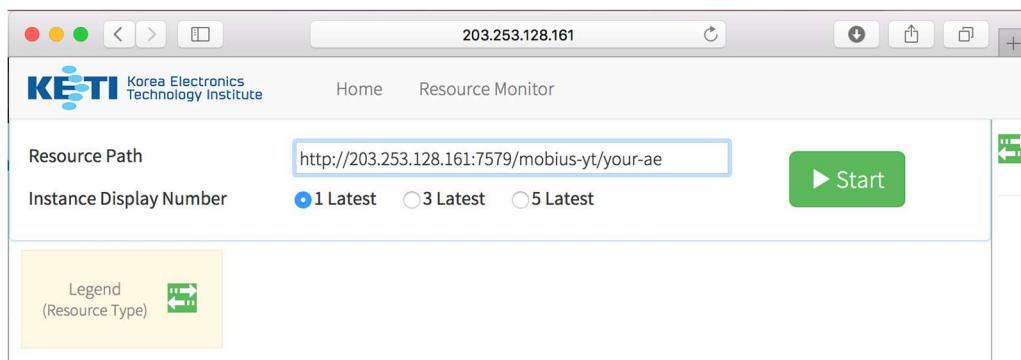
<http://203.253.128.161:7575/>



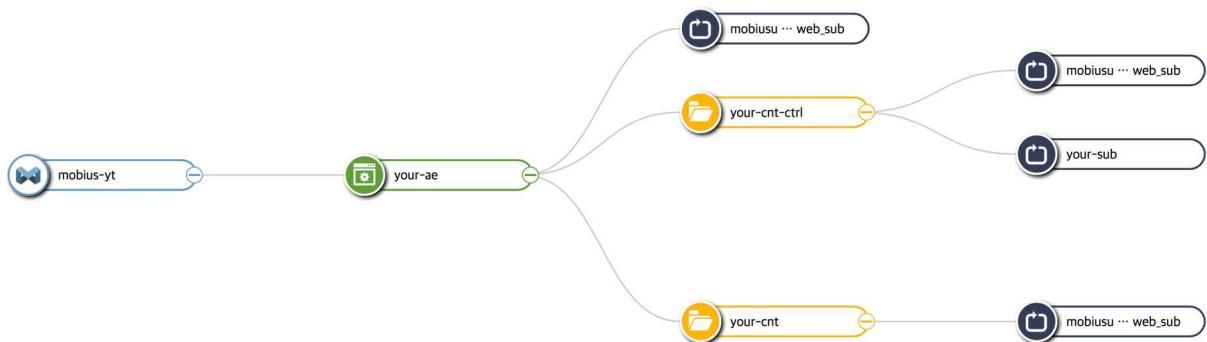
위의 URL을 통해 웹버전의 Mobius Resource Monitor에 접속한다.

Resource Monitor 기능을 이용하기 위해 상단의 메뉴에서 Resource Monitor 탭을 클릭하여 Resource Monitor 페이지에 접속한다.

<http://203.253.128.161:7579/mobius-xt/your-ae>



Resource Path에 위의 주소를 넣고 Start 버튼을 클릭하면, nCube-Mint 보드가 생성한 Resource Tree를 볼 수 있다.



Mobius-xt CSE 아래에 mint ae, co2 cnt, led cnt가 잘 생성된 것을 확인할 수 있다.

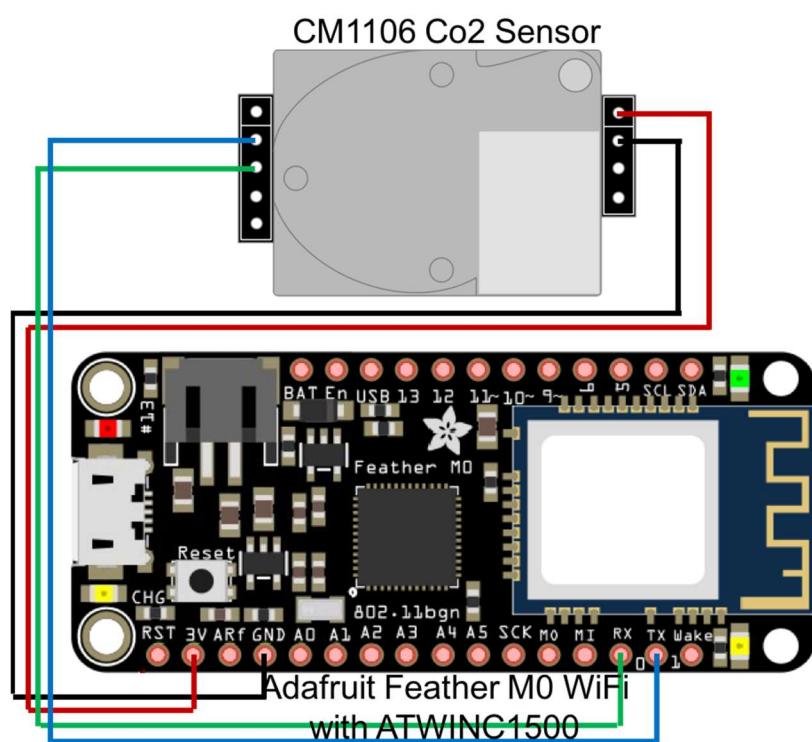
4 센서별 Cube-Mint 응용 개발

이번 장에서는 nCube-Mint 보드에 Co2 센서, RGB-LED, Relay 를 부착하여, Co2 농도를 IoT Platform 에 업로드 하고, IoT Platform 을 이용하여 nCube-Mint 보드에 부착된 RGB-LED 와 Relay 를 제어한다.

4 - 1 nCube-Mint + Co2 Sensor

nCube-Mint 보드에 Co2 센서를 연결시킨다. TasCO2 라이브러리를 이용해 현재 대기중의 Co2 농도를 측정하고, co2_sensing_interval 변수의 값을 주기로 IoT Platform 에 업로드하도록 한다.

4 - 1 - 1 Co2 센서 연결



Adafruit Feather M0 와 Co2 센서의 연결은 위의 그림을 참조하면 된다.

4 - 1 - 2 TasCO2 라이브러리 소개

TAS(Thing Adeption Software)란 nCube-Mint 와 센서를 연결시키기 위한 소프트웨어이다.

본 문서 내에서 사용되는 TAS 는 Co2 센서를 지원하기 위해 제작된 TasCO2 이다.

객체를 만들어서 라이브러리 내의 변수에 접근하고, 함수를 호출한다.

Init()

Co2 센서와 nCube-Mint 를 URAT 를 이용해 연결하기 위한 함수이다.

Co2 센서를 이용하기 위해 최초로 한번 꼭 호출해 주어야 한다.

setCallback(measure_callback)

Co2 센서를 이용하여 대기중의 Co2 농도 측정이 완료된 후에 동작할 callback 함수를 지정하는 함수이다.

chkCO2Data()

Co2 센서가 측정한 현재 대기중의 Co2 농도를 등록된 Callback 함수의 매개변수로 전달해 주는 함수이다.

requestData()

Co2 센서가 현재 대기중의 Co2 농도를 측정하도록 명령하는 함수이다.

4 - 1 - 3 nCube-Mint + Co2 Sensor 스케치

nCube-Mint 보드에 Co2 센서를 연결시키고 이용하기 위해서 TasCO2 라이브러리를 사용하여 nCube-Mint 기본 예제 스케치에 추가적인 코드를 작성해서 넣어야한다.

```
19 const String AE_ID = "mint"; // guide: this is same with AE name
20 const String MQTT_BROKER_IP = "203.253.128.161"; // guide: set IP address of MQTT broker for Mobius as IoT Platform
21 const uint16_t MQTT_BROKER_PORT = 1883; // guide: set MQTT port used
22 OneM2MClient nCube(MQTT_BROKER_IP, MQTT_BROKER_PORT, AE_ID); // AE-ID
23
24 unsigned long req_previousMillis = 0;
25 const long req_interval = 2000;
26
27 // guide: set sensing period, modify or add for your sensors
28
29 unsigned long sensing_previousMillis = 0;
30 const long sensing_interval = (1000 * 5);
```

ae-id 를 mint 로 수정해 주었으며, sensing_interval 값을 통해 Co2 농도 업로드 주기가 5초로 설정되어있는 것을 확인할 수 있다.

```
102 void buildResource() {
103     // temporally build resource structure into Mobius as oneM2M IoT Platform
104
105     // AE resource
106     uint8_t index = 0;
107     nCube.resource[index].ty = "2";
108     nCube.resource[index].to = "/mobius-yt";
109     nCube.resource[index].rn = AE_ID;
110     nCube.resource[index++].status = 0;
111
112     // Container resource
113     nCube.resource[index].ty = "3";
114     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn;
115     nCube.resource[index].rn = "co2"; // guide: change to co2
116     nCube.resource[index++].status = 0;
117
118     nCube.resource_count = index;
119 }
```

현재 동작시키고자 하는 nCube-Mint의 경우, ae 와, Co2 농도를 저장할 cnt 가 1개 필요하므로 buildResource()함수의 코드를 위와같이 바꿔준다. Co2 농도 데이터를 저장할 cnt 의 이름은 co2 로 수정하였다.

nCube-Mint 제작 가이드

```
17 #include "TasCO2.h"  
25 TasCO2 TasCO2Sensor;
```

TasCO2 라이브러리를 추가시키고, TasCO2Sensor 이라는 이름으로 객체를 생성하였다.

```
272 TasCO2Sensor.init();  
273 TasCO2Sensor.setCallback(upload_callback);
```

init()함수를 호출하여 초기화를 시켜주고, setCallback()함수에 upload_callback 함수를 매개변수로 전달하여 호출하였다.

해당 코드는 nCube-Mint 가 동작할 때 최초로 1회 실행되면 충분하므로, nCube-Mint 의 기본 예제 스케치의 setup()함수 내에 위치시킨다.

```
287 if (state == "create_cin") {  
288     // guide: in here generate sensing data  
289     // if get sensing data directly, assign curValue sensing data and set sensing_flag to 1  
290     // if request sensing data to sensor, set sensing_flag to 0, in other code of receiving  
291     TasCO2Sensor.requestData();  
292     sensing_flag = 0;  
293 }  
294 }
```

nCube-Mint 기본 예제 스케치의 loop() 함수 내에 if (state == “create_cin”) 조건문 안에서 requestData() 함수를 호출하여, 대기중의 Co2 농도를 측정할 수 있도록 해준다.

```
66 void upload_callback(String con) {  
67     if (state == "create_cin") {  
68         curValue = con;  
69         sensing_flag = 1;  
70     }  
71 }
```

Co2 농도 측정이 완료되면, 매개변수인 con 에 현재 대기의 Co2 농도가 저장되어 upload_callback 함수가 호출된다. 현재 대기의 Co2 농도값인 con 값을 curValue 값에 저장시켜주고, Co2 측정이 완료되었음을 표시하기 위해 sensing_flag 값을 1로 바꾸어 주도록 한다.

```
317     TasCO2Sensor.chkCO2Data();
```

nCube-Mint 기본 예제 스케치의 loop()함수 안에 위치한 if (nCube.chkConnect()) 조건문 안에서 chkCO2Data() 함수를 호출하도록 한다.

4 - 1 - 4 nCube-Mint Co2 동작과 데이터 조회

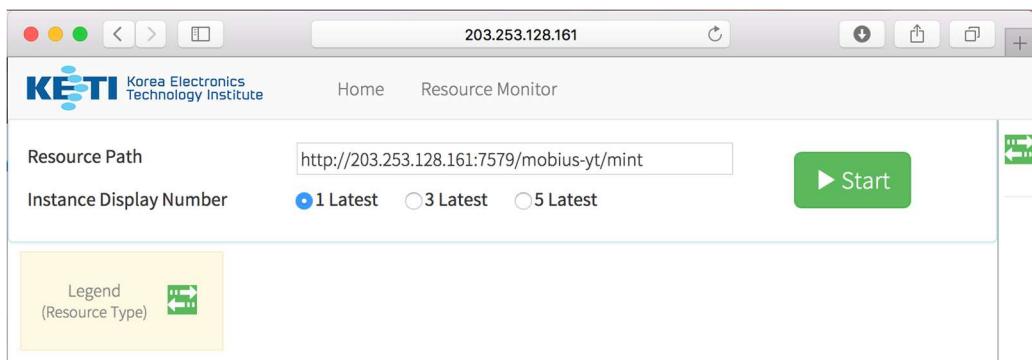
스케치를 업로드 하고, 스마트폰을 이용하여 nCube-Mint 보드가 이용할 WiFi AP의 SSID 와 Password 를 다시 입력하여 준다.



```
/dev/cu.usbmodem1421 (Adafruit Feather M0)
[REDACTED] 전송
REQUEST OK!
{"rsc":2001,"rqi":"8wAdCZx9","pc":{"m2m:cin":{"rn":"4-20170710041805205s7fa","ty":4,"pi":"rkGfukYLBZ","ri":"HJGBbet
Receive published data
/oneM2M/resp/mint/mobius-yt/json - RESP_TOPIC receive a message.
Read: <A>
{"op":1,"to":"/mobius-yt/mint/co2","fr":"mint","rqi":"Q9yP8Vgy","ty":4,"pc":{"m2m:cin":{"con":"2000.00"}}
REQUEST OK!
{"rsc":2001,"rqi":"Q9yP8Vgy","pc":{"m2m:cin":{"rn":"4-20170710041810209Z2La","ty":4,"pi":"rkGfukYLBZ","ri":"r1M9-LK
Receive published data
/oneM2M/resp/mint/mobius-yt/json - RESP_TOPIC receive a message.
Read: <A>
{"op":1,"to":"/mobius-yt/mint/co2","fr":"mint","rqi":"yXs5iy63","ty":4,"pc":{"m2m:cin":{"con":"2000.00"}}
REQUEST OK!
{"rsc":2001,"rqi":"yXs5iy63","pc":{"m2m:cin":{"rn":"4-20170710041815210Z90A","ty":4,"pi":"rkGfukYLBZ","ri":"SkG1Ggt
Receive published data
/oneM2M/resp/mint/mobius-yt/json - RESP_TOPIC receive a message.

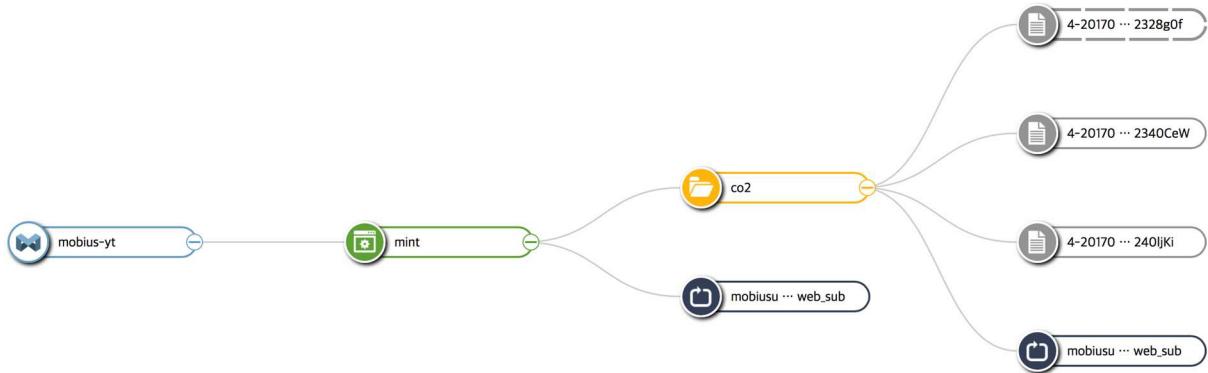
[REDACTED]
자동 스크롤 Both NL & CR 115200 보드레이트
```

시리얼 모니터를 통해 5 초 간격으로 위와같이 Co2 농도를 측정하여 IoT Platform 으로 업로드 하는 것을 볼 수 있다.



Mobius Resource Monitor 를 통해 nCube-Mint 가 업로드한 Co2 농도 데이터를 조회해보자. nCube-Mint 의 ae-id 가 mint 로 변경 되었으므로, Resource Path 는 http://203.253.128.161:7579/mobius-yt/mint 로 하여 조회한다.

nCube-Mint 제작 가이드

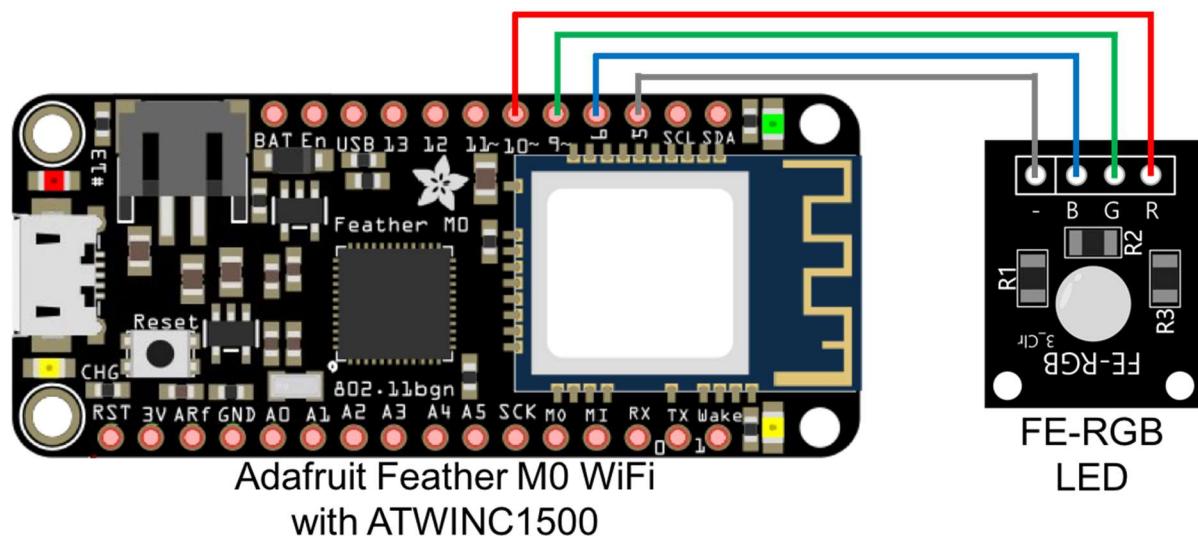


위와같이 5 초 주기로 Co2 농도 데이터가 업로드 되고 있음을 볼 수 있다.

4 - 2 RGB-LED 연결

nCube-Min 보드 t에 RGB-LED를 연결시키고, Mobius Resource Monitor를 이용하여 LED를 제어해 본다.

4 - 2 - 1 RGB-LED 연결



Adafruit Feather M0 와 RGB-LED 의 연결은 위의 그림을 참조하면 된다.

4 - 2 - 2 nCube-Mint + LED 스케치

Mobius로 부터 LED 제어 noti를 받아서 LED를 제어하기 위해, 아래와 같은 스케치의 수정이 필요하다. 스케치는 nCube-Mint 기본 예제 스케치를 바탕으로 수정한다.

```
19 const String AE_ID = "mint"; // guide: this is same with AE name
20 const String MQTT_BROKER_IP = "203.253.128.161"; // guide: set IP address of MQTT broker for Mobius as IoT Platform
21 const uint16_t MQTT_BROKER_PORT = 1883; // guide: set MQTT port used
22 OneM2MClient nCube(MQTT_BROKER_IP, MQTT_BROKER_PORT, AE_ID); // AE-ID
```

ae-id 를 mint 로 수정해준다.

```
102 void buildResource() {
103     // temporally build resource structure into Mobius as oneM2M IoT Platform
104
105     // AE resource
106     uint8_t index = 0;
107     nCube.resource[index].ty = "2";
108     nCube.resource[index].to = "/mobius-yt";
109     nCube.resource[index].rn = AE_ID;
110     nCube.resource[index++].status = 0;
111
112     // Container resource
113     nCube.resource[index].ty = "3";
114     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn;
115     nCube.resource[index].rn = "led-ctrl";
116     nCube.resource[index++].status = 0;
117
118     // Subscription resource
119     nCube.resource[index].ty = "23";
120     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn + '/' + nCube.resource[1].rn;
121     nCube.resource[index].rn = "led-sub";
122     nCube.resource[index++].status = 0;
123
124     nCube.resource_count = index;
125 }
```

현재 동작시키고자 하는 nCube-Mint의 경우, ae 와, LED 제어 데이터를 저장할 cnt 가 1개, LED 제어 데이터를 noti 하여줄 sub 가 1개 필요하므로, buildResource()함수의 코드를 위와같이 바꾸었다. LED 제어 데이터를 저장할 cnt 는 led-ctrl, LED 제어 데이터를 noti 하여줄 sub 는 led-sub 로 각각 이름을 변경하였다.
Subscription 리소스를 생성하는 부분에서 to 값의 하위 리소스를 가리키는 nCube.resource[1].rn 가 nCube-Mint 기본 예제 스케치와는 달리 [2]가 아닌 [1]로 수정해 주어야 한다.

nCube-Mint 제작 가이드

```
86 void noti_callback(String topic, JsonObject& root) {  
87     if (state == "create_cin") {  
88         if (root["pc"]["sgn"]["sur"] == (nCube.resource[2].to + "/" + nCube.resource[2].rn)) { // guide: uri of subscription re:  
89             String con = root["pc"]["sgn"]["nev"]["rep"]["m2m:cin"]["con"];  
90             noti_con = con;  
91  
92             String rqi = root["rqi"];  
93             String resp_body = "";  
94             resp_body += "{\"rsc\":\"2000\", \"to\":\"\", \"fr\":\"\" + nCube.getAeid() + "\", \"pc\":\"\", \"rqi\":\"\" + rqi + "\"}";  
95  
96             resp_body.toCharArray(body_buff, resp_body.length() + 1);  
97             control_flag = 1;  
98         }  
99     }  
100 }
```

noti_callback()함수 내의 if 문의 조건식 내에서 리소스에 접근하는 부분의 코드를 앞서 수정한 리소스구조에 맞춰 nCube.resource[3]을 nCube.resource[2]로 수정하여 준다.

```
19 #define LED_RED_PIN 10  
20 #define LED_GREEN_PIN 9  
21 #define LED_BLUE_PIN 6  
22 #define LED_GND_PIN 5
```

RGB-LED 의 Red, Green, Blue LED 와 접지 핀과 Adafruit Feather M0 에 연결된 핀을 정의한다. 10, 9 와 같이 숫자로 표현하는 것 보다 LED_RED_PIN, LED_GREEN_PIN 과 같이 문자로 표현하는 것이 가독성을 높이는데 도움이 된다.

```
267 pinMode(LED_BLUE_PIN, OUTPUT);  
268 pinMode(LED_GREEN_PIN, OUTPUT);  
269 pinMode(LED_RED_PIN, OUTPUT);  
270 pinMode(LED_GND_PIN, OUTPUT);  
271  
272 digitalWrite(LED_BLUE_PIN, LOW);  
273 digitalWrite(LED_GREEN_PIN, LOW);  
274 digitalWrite(LED_RED_PIN, LOW);  
275 digitalWrite(LED_GND_PIN, LOW);
```

setup()함수 내에서 RGB-LED 와 연결된 Adafruit Feather M0 의 핀들이 digital 신호(0 or 1)를 출력하도록 설정하고 있다.

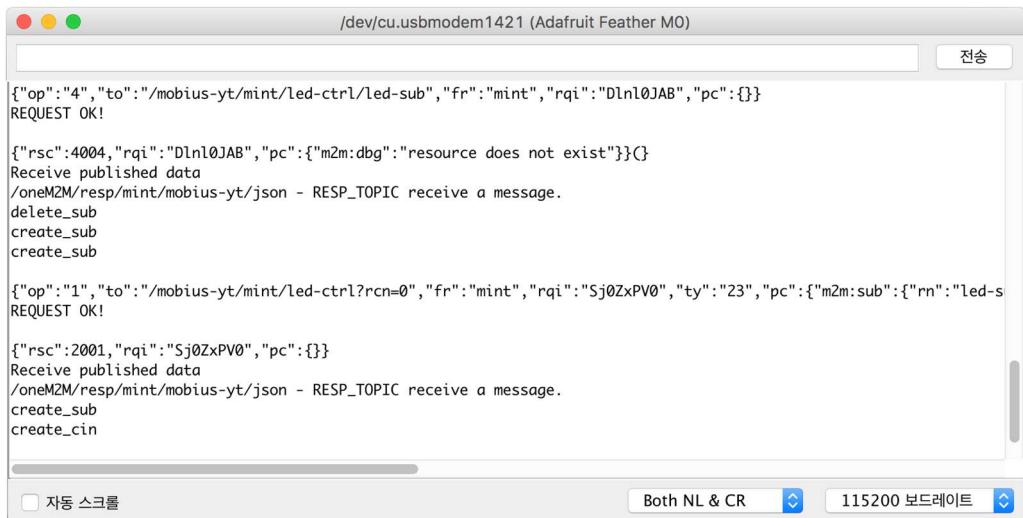
nCube-Mint 제작 가이드

```
324     if (noti_con == "0") {  
325         digitalWrite(LED_BLUE_PIN, LOW);  
326         digitalWrite(LED_GREEN_PIN, LOW);  
327         digitalWrite(LED_RED_PIN, LOW);  
328     }  
329     else if (noti_con == "1") {  
330         digitalWrite(LED_BLUE_PIN, LOW);  
331         digitalWrite(LED_GREEN_PIN, LOW);  
332         digitalWrite(LED_RED_PIN, HIGH);  
333     }  
334     else if (noti_con == "2") {  
335         digitalWrite(LED_BLUE_PIN, LOW);  
336         digitalWrite(LED_GREEN_PIN, HIGH);  
337         digitalWrite(LED_RED_PIN, LOW);  
338     }  
339     else if (noti_con == "3") {  
340         digitalWrite(LED_BLUE_PIN, HIGH);  
341         digitalWrite(LED_GREEN_PIN, LOW);  
342         digitalWrite(LED_RED_PIN, LOW);  
343     }  
344     else if (noti_con == "4") {  
345         digitalWrite(LED_BLUE_PIN, HIGH);  
346         digitalWrite(LED_GREEN_PIN, HIGH);  
347         digitalWrite(LED_RED_PIN, LOW);  
348     }  
349     else if (noti_con == "5") {  
350         digitalWrite(LED_BLUE_PIN, HIGH);  
351         digitalWrite(LED_GREEN_PIN, LOW);  
352         digitalWrite(LED_RED_PIN, HIGH);  
353     }  
354     else if (noti_con == "6") {  
355         digitalWrite(LED_BLUE_PIN, LOW);  
356         digitalWrite(LED_GREEN_PIN, HIGH);  
357         digitalWrite(LED_RED_PIN, HIGH);  
358     }  
359     else if (noti_con == "7") {  
360         digitalWrite(LED_BLUE_PIN, HIGH);  
361         digitalWrite(LED_GREEN_PIN, HIGH);  
362         digitalWrite(LED_RED_PIN, HIGH);  
363     }
```

nCube-Mint 기본 예제 스케치의 loop() 함수 내에 if (control_flag == 1) 조건문 안에서 Mobius 로 부터 받은 noti 값에 따라 Red, Green, Blue LED 를 각각 끄고 켜도록 해준다.

4 - 2 - 3 nCube-Mint LED 동작과 제어

스케치를 업로드 하고, 스마트폰을 이용하여 nCube-Mint 보드가 이용할 WiFi AP의 SSID 와 Password 를 다시 입력하여 준다.



```
"/dev/cu.usbmodem1421 (Adafruit Feather M0) 전송

{"op": "4", "to": "/mobius-yt/mint/led-ctrl/led-sub", "fr": "mint", "rqi": "Dlnl0JAB", "pc": "{}"} REQUEST OK!
{"rsc": "4004", "rqi": "Dlnl0JAB", "pc": {"m2m:dbg": "resource does not exist"}}

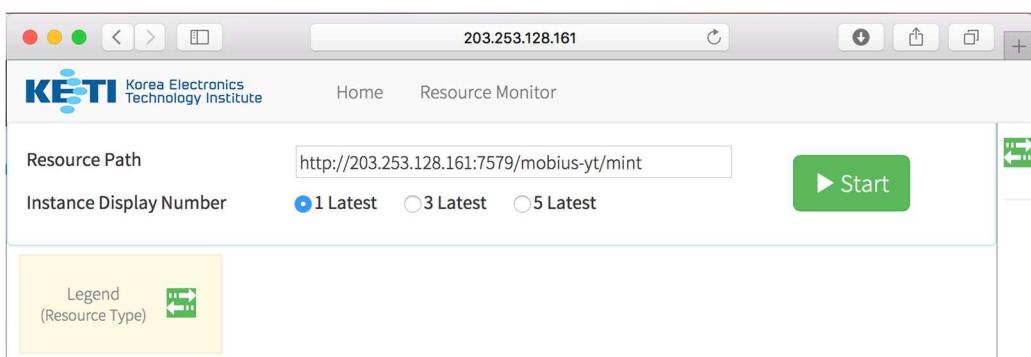
Receive published data
/onem2m/resp/mint/mobius-yt/json - RESP_TOPIC receive a message.
delete_sub
create_sub
create_sub

{"op": "1", "to": "/mobius-yt/mint/led-ctrl?rcn=0", "fr": "mint", "rqi": "Sj0ZxPV0", "ty": "23", "pc": {"m2m:sub": {"rn": "led-s"}}} REQUEST OK!
{"rsc": "2001", "rqi": "Sj0ZxPV0", "pc": {}}

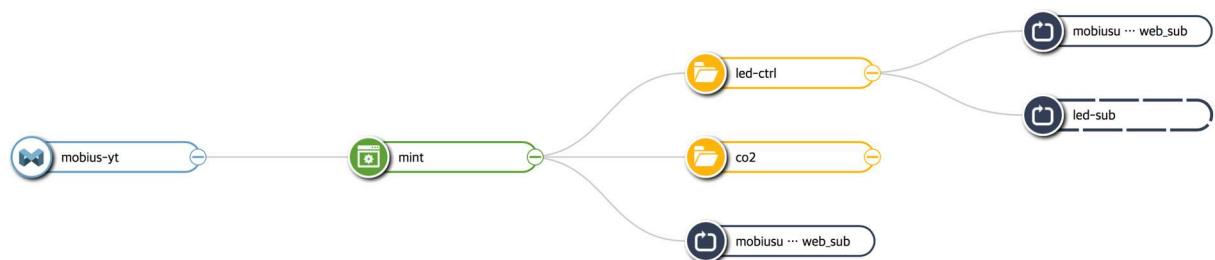
Receive published data
/onem2m/resp/mint/mobius-yt/json - RESP_TOPIC receive a message.
create_sub
create_cin

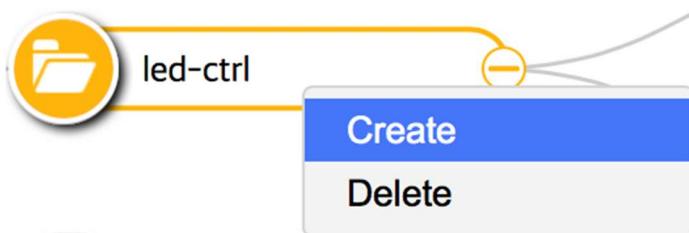
Both NL & CR 115200 보드레이트
```

시리얼 모니터의 출력이 create_cin 에서 멈추는지 확인하고, 다음 단계로 넘어간다.

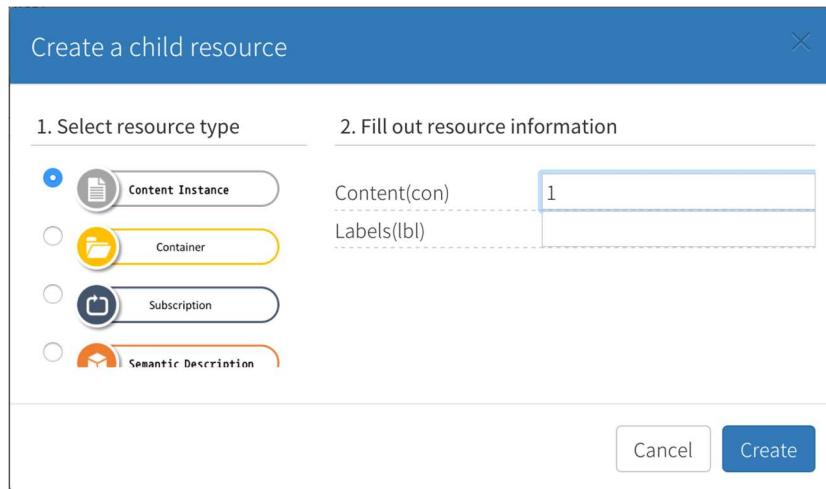


Mobius Resource Monitor 를 통해 nCube-Mint 보드에 연결된 RGB-LED 를 제어하기에 앞서, nCube-Mint 가 만든 리소스 구조를 조회해보자. Resource Path 는 <http://203.253.128.161:7579/mobius-yt/mint> 로 이전 예제와 같고, 조회되는 리소스 구조는 아래의 그림과 같다.

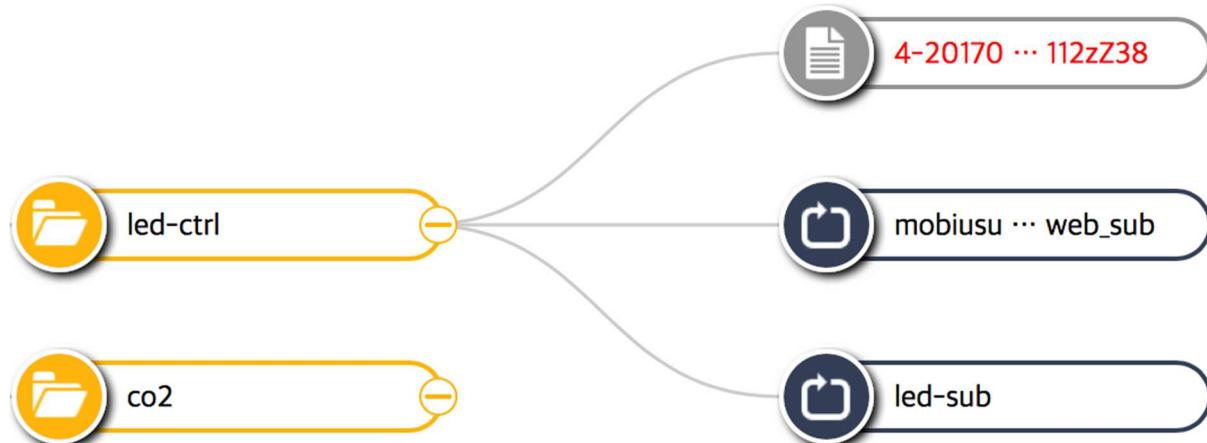




led-ctrl cnt 위에서 마우스 오른쪽을 클릭하여 메뉴를 열고, Create 버튼을 클릭한다.



Create a child resource 창에서 좌측의 Select resource type 에 Content Instance 가 체크되어 있는것을 확인하고, 우측의 Fill out resource information 에 Content 에 1 을 넣고 Create 버튼을 클릭한다.



cin 이 새로 생성된 모습을 바로 확인할 수 있다.

nCube-Mint 제작 가이드



Arduino IDE의 시리얼 모니터를 통해 noti 를 수신하고 Response 도 완료한 것을 확인할 수 있다.

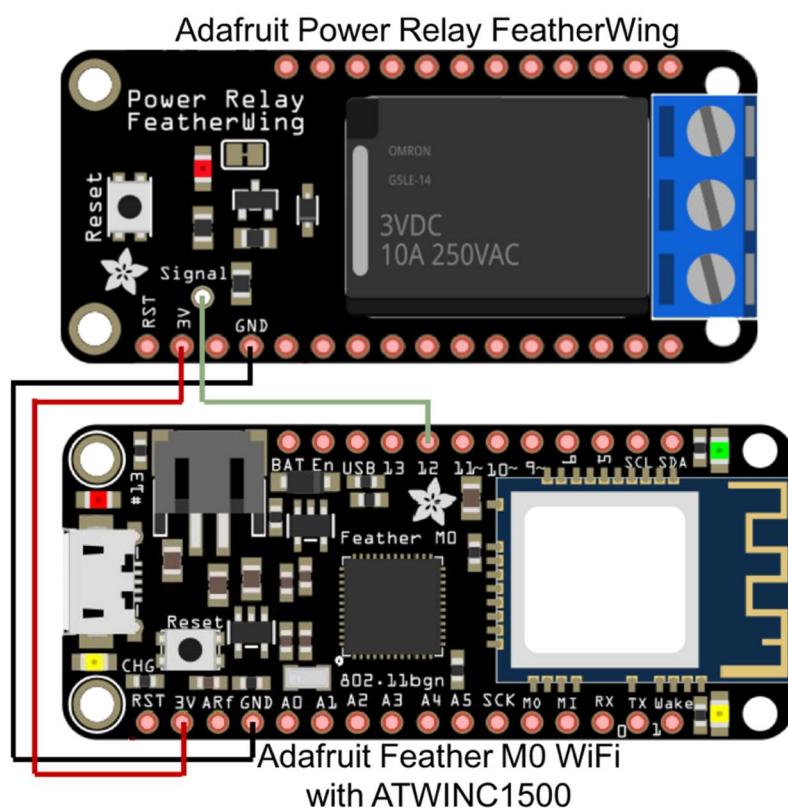


cin 의 content 값에 따라 다양한 LED 빛을 확인할 수 있다.

4 - 3 Adafruit Powre Relay FeatherWing 연결

nCube-Mint 보드에 Adafruit Power Relay FeatherWing 를 연결시키고, Mobius Resource Monitor 를 이용하여 Relay 를 제어해 본다.

4 - 3 - 1 Adafruit Powre Relay FeatherWing 연결



Adafruit Feather M0 와 Adafruit Power Relay FeatherWing 의 연결은 위의 그림을 참조하면 된다.

4 - 3 - 2 nCube-Mint + Powre Relay 스케치

Mobius로 부터 Relay 제어 noti를 받아서 Relay를 제어하기 위해, 아래와 같은 스케치의 수정이 필요하다. 스케치는 nCube-Mint 기본 예제 스케치를 바탕으로 수정한다.

```
19 const String AE_ID = "mint"; // guide: this is same with AE name
20 const String MQTT_BROKER_IP = "203.253.128.161"; // guide: set IP address of MQTT broker for Mobius as IoT Platform
21 const uint16_t MQTT_BROKER_PORT = 1883; // guide: set MQTT port used
22 OneM2MCClient nCube(MQTT_BROKER_IP, MQTT_BROKER_PORT, AE_ID); // AE-ID
```

이전 예제와 마찬가지로 ad-id를 mint로 바꿔준다.

```
102 void buildResource() {
103     // temperally build resource structure into Mobius as oneM2M IoT Platform
104
105     // AE resource
106     uint8_t index = 0;
107     nCube.resource[index].ty = "2";
108     nCube.resource[index].to = "/mobius-yt";
109     nCube.resource[index].rn = AE_ID;
110     nCube.resource[index++].status = 0;
111
112     // Container resource
113     nCube.resource[index].ty = "3";
114     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn;
115     nCube.resource[index].rn = "relay-ctrl"; // relay-ctrl
116     nCube.resource[index++].status = 0;
117
118     // Subscription resource
119     nCube.resource[index].ty = "23";
120     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn + '/' + nCube.resource[1].rn; // relay-sub
121     nCube.resource[index].rn = "relay-sub";
122     nCube.resource[index++].status = 0;
123
124     nCube.resource_count = index;
125 }
```

ae와, Relay 제어 데이터를 저장할 cnt가 1개, Relay 제어 데이터를 noti하여줄 sub가 1개 필요하므로, buildResource()함수의 코드를 위와같이 바꾸었다. cnt와 sub의 이름은 각각 relay-cnt, relay-sub이다. Subscription 리소스를 생성하는 부분에서 이전 예제와 마찬가지로 nCube.resource[1].rn로 수정되어야 한다.

nCube-Mint 제작 가이드

```
86 void noti_callback(String topic, JsonObject& root) {  
87     if (state == "create_cin") {  
88         if (root["pc"]["sgn"]["sur"] == (nCube.resource[2].to + "/" + nCube.resource[2].rn)) { // guide: uri of subscription re  
89             String con = root["pc"]["sgn"]["nev"]["rep"]["m2m:cin"]["con"];  
90             noti_con = con;  
91         }  
92         String rqi = root["rqi"];  
93         String resp_body = "";  
94         resp_body += "{\"rsc\":\"2000\", \"to\":\"\", \"fr\":\"\" + nCube.getAeid() + "\", \"pc\":\"\", \"rqi\":\"\" + rqi + "\"};  
95         resp_body.toCharArray(body_buff, resp_body.length() + 1);  
96         control_flag = 1;  
97     }  
98 }  
99 }  
100 }
```

이전 예제와 마찬가지로 noti_callback()함수에서 nCube.resource[3].to, nCube.resource[3].rn 이 각각 nCube.resource[2].to, nCube.resource[2].rn 으로 수정되어야 한다.

19 #define RELAY_PIN 12

Adafruit Power Relay FeatherWing 의 signal 핀과 Adafruit Feather M0 에 연결된 핀을 정의한다.

```
264     pinMode(RELAY_PIN, OUTPUT);  
265  
266     digitalWrite(RELAY_PIN, LOW);
```

setup()함수 내에서 Relay 와 연결된 핀이 digital 신호(0 or 1)를 출력하도록 설정하고 있다.

```
315     if (noti_con == "0") {  
316         digitalWrite(RELAY_PIN, LOW);  
317     }  
318     else if (noti_con == "1") {  
319         digitalWrite(RELAY_PIN, HIGH);  
320     }
```

nCube-Mint 기본 예제 스케치의 loop() 함수 내에 if (control_flag == 1) 조건문 안에서 Mobius 로 부터 받은 noti 값에 따라 Relay signal 값을 0 또는 1로 바꾸도록 해준다.

4 - 3 - 3 nCube-Mint Power Relay 동작과 제어

스케치를 업로드 하고, 스마트폰을 이용하여 nCube-Mint 보드가 이용할 WiFi AP의 SSID 와 Password 를 다시 입력하여 준다.

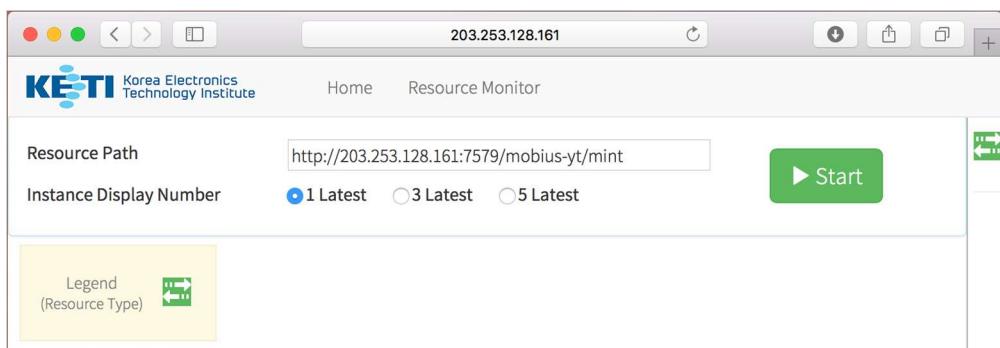


The screenshot shows a serial monitor window titled "/dev/cu.usbmodem1421 (Adafruit Feather M0)". The text area contains the following message exchange:

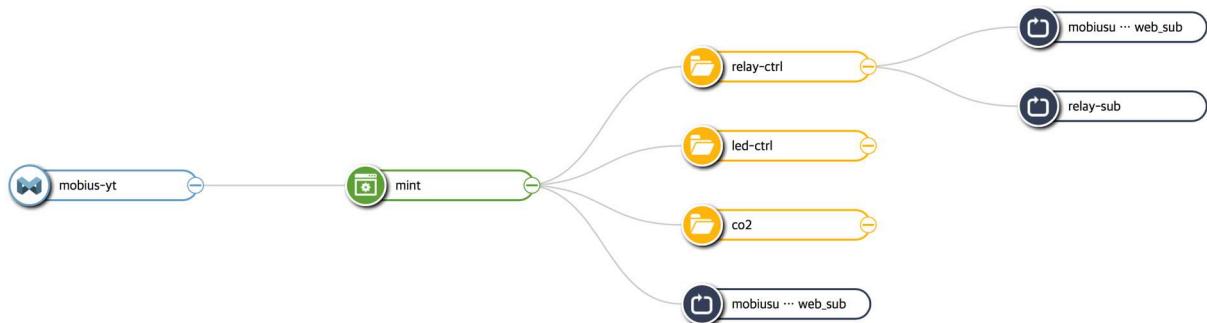
```
create_sub
create_sub
{"op": "1", "to": "/mobius-yt/mint/relay-ctrl?rcn=0", "fr": "mint", "rqi": "m1R6SrqB", "ty": "23", "pc": {"m2m:sub": {"rn": "relay-ctrl", "f": "2001"}, "m2m:cin": {"rn": "relay-state", "f": "0"}}, "txid": 1
REQUEST OK!
{"rsc": "2001", "rqi": "m1R6SrqB", "pc": {}}◆
Receive published data
/oneM2M/resp/mint/mobius-yt/json - RESP_TOPIC receive a message.
create_sub
create_cin
```

At the bottom, there are checkboxes for "자동 스크롤" (Auto Scroll) and "Both NL & CR", and a baud rate selector set to "115200 보드레이트" (115200 Baudrate).

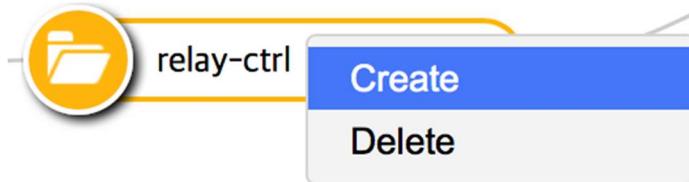
시리얼 모니터의 출력이 create_cin 에서 멈추는지 확인하고, 다음 단계로 넘어간다.



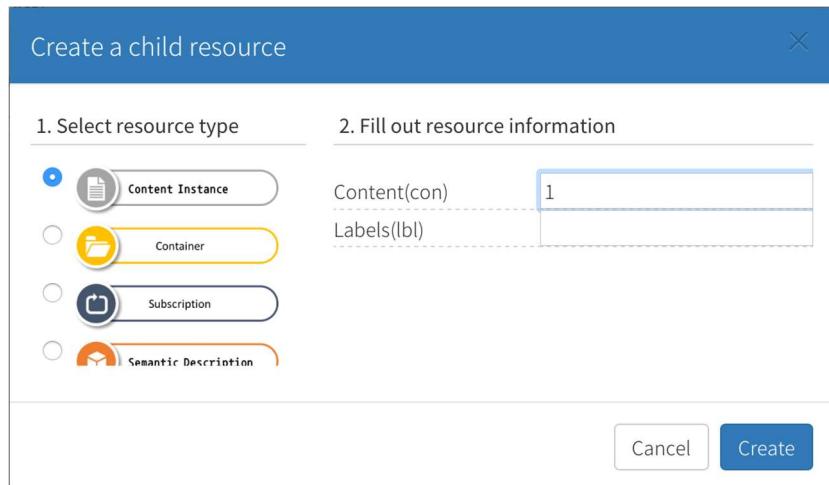
Mobius Resource Monitor 를 통해 nCube-Mint 보드에 연결된 Power Relay 를 제어하기에 앞서, 리소스 구조를 조회해보자. Resource Path 는 <http://203.253.128.161:7579/mobius-yt/mint> 이다.



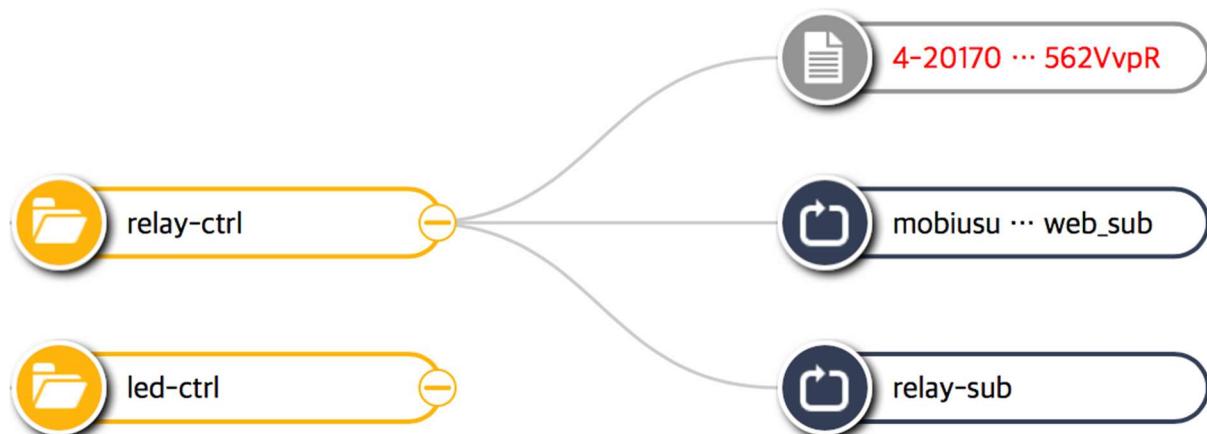
조회된 리소스 구조는 위의 그림과 같다.



relay-ctrl cnt 위에서 마우스 오른쪽을 클릭하여 메뉴를 열고, Create 버튼을 클릭한다.



Create a child resource 창에서 좌측의 Select resource type에 Content Instance 가 체크되어 있는것을 확인하고, 우측의 Fill out resource information에 Content에 1을 넣고 Create 버튼을 클릭한다.



cin 이 생성된 모습을 바로 확인할 수 있다.

nCube-Mint 제작 가이드



The screenshot shows the Arduino IDE's Serial Monitor window. The title bar reads "/dev/cu.usbmodem1421 (Adafruit Feather M0)". The main text area displays a JSON message and a response:

```
{"op":5,"net":"3","fr":"/mobius-yt","rqi":"BkIx0o5lBW","pc":{"sgn":{"net":"3","sur":"/mobius-yt/mint/relay-ctrl/rel1/oneM2M/resp/mobius-yt/mint/json\nRESPONSE OK!"}}
```

 Below the text area, there are two buttons: "자동 스크롤" (Auto Scroll) and "전송" (Send). At the bottom, the baud rate is set to "115200 보드레이트" (115200 Baud Rate).

Arduino IDE 의 시리얼 모니터를 통해 noti 를 수신하고 Response 도 완료한 것을 확인할 수 있다.

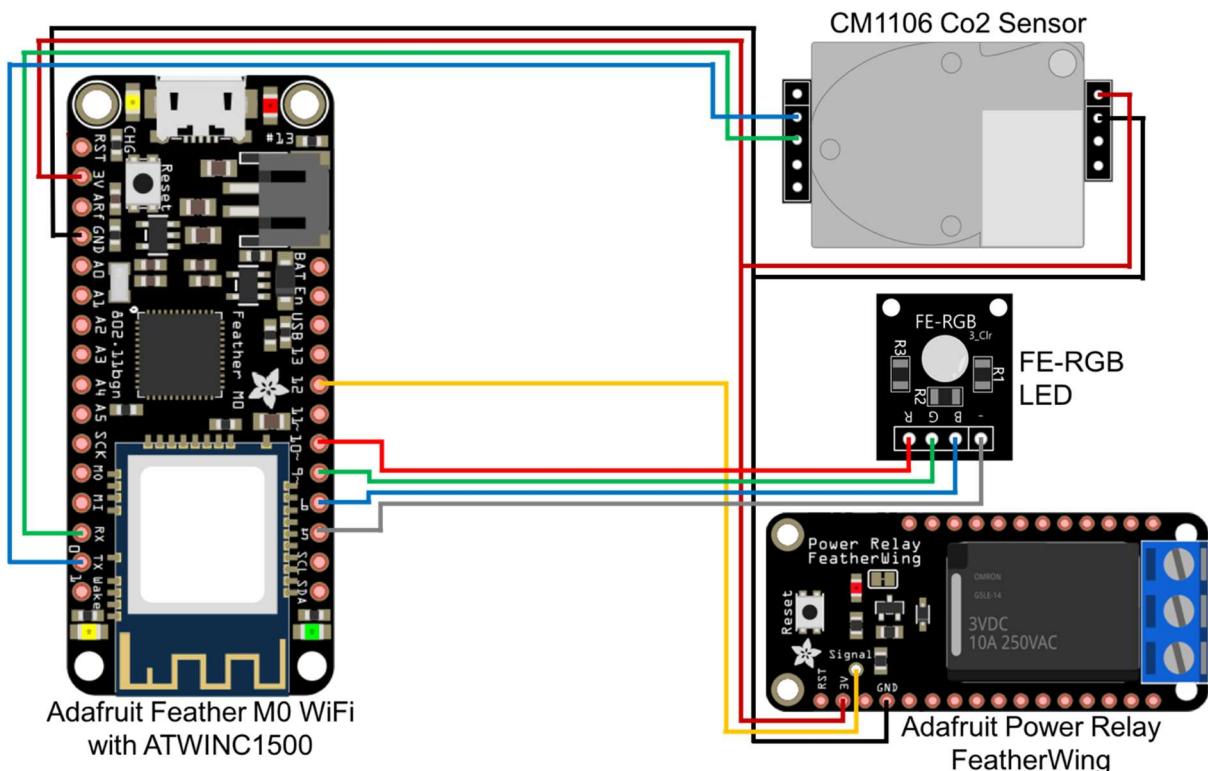


Power Relay 에 내장된 빨간 LED 가 점등되면 작동 표시를 하는것을 알 수 있다.

5 전체 센서 통합 nCube-Mint 개발 및 테스트

5 장에서는 4 장에서 보인 Co2 센서, RGB-LED, Power Relay 를 모두 Adafruit Feather M0 와 연결시키고, 업로드 되는 Co2 농도 데이터 조회와 LED, Relay 제어를 한다.

5 - 1 nCube-Mint 회로 구성



Adafruit Feather M0 와 Co2 센서, RGB-LED, Adafruit Power Relay FeatherWing 를 연결한 모습은 위의 그림과 같다.

5 - 2 nCube-Mint 스케치

CO₂ 센서, RGB-LED, Power Relay 를 모두 연결한 nCube-Mint 의 최종적인 스케치는 nCube-Mint 기본 예제 스케치에서 다음과 같은 코드의 수정 및 추가가 필요하다.

윗 라인부터 추가가 필요한 코드를 소개한다.

```
17 #include "TasCO2.h"
18
19 #define LEDPIN 13
20
21 #define LED_RED_PIN 10
22 #define LED_GREEN_PIN 9
23 #define LED_BLUE_PIN 6
24 #define LED_GND_PIN 5
25
26 #define RELAY_PIN 12

28 const String AE_ID = "mint";           // guide: this is same with AE name
29 const String MQTT_BROKER_IP = "203.253.128.161"; // guide: set IP address of MQTT broker
30 const uint16_t MQTT_BROKER_PORT = 1883;    // guide: set MQTT port used
31 OneM2MCClient nCube(MQTT_BROKER_IP, MQTT_BROKER_PORT, AE_ID); // AE-ID
32 TasCO2 TasCO2Sensor;

73 void upload_callback(String con) {
74     if (state == "create_cin") {
75         curValue = con;
76         sensing_flag = 1;
77     }
78 }
```

nCube-Mint 제작 가이드

```
97 void noti_callback(String topic, JsonObject& root) {
98     if (state == "create_cin") {
99         if (root["pc"]["sgn"]["sur"] == (nCube.resource[4].to + "/" + nCube.resource[4].rn)) { // guide: uri of subscription res
100            String con = root["pc"]["sgn"]["nev"]["rep"]["m2m:cin"]["con"];
101            noti_con = con;
102
103            String rqi = root["rqi"];
104            String resp_body = "";
105            resp_body += "{\"rsc\":\"2000\", \"to\":\"\", \"fr\":\"\" + nCube.getAeid() + "\", \"pc\":\"\", \"rqi\":\"\" + rqi + "\"}";
106
107            resp_body.toCharArray(body_buff, resp_body.length() + 1);
108            control_flag = 1;
109        }
110    else if(root["pc"]["sgn"]["sur"] == (nCube.resource[5].to + "/" + nCube.resource[5].rn)) {
111        String con = root["pc"]["sgn"]["nev"]["rep"]["m2m:cin"]["con"];
112        noti_con = con;
113
114        String rqi = root["rqi"];
115        String resp_body = "";
116        resp_body += "{\"rsc\":\"2000\", \"to\":\"\", \"fr\":\"\" + nCube.getAeid() + "\", \"pc\":\"\", \"rqi\":\"\" + rqi + "\"}";
117
118        resp_body.toCharArray(body_buff, resp_body.length() + 1);
119        control_flag = 2;
120    }
121 }
122 }
```

```
124 void buildResource() {
125     // temperally build resource structure into Mobius as oneM2M IoT Platform
126
127     // AE resource
128     uint8_t index = 0;
129     nCube.resource[index].ty = "2";
130     nCube.resource[index].to = "/mobius-yt";
131     nCube.resource[index].rn = AE_ID;
132     nCube.resource[index++].status = 0;
133
134     // Container resource
135     nCube.resource[index].ty = "3";
136     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn;
137     nCube.resource[index].rn = "co2";
138     nCube.resource[index++].status = 0;
139
140     nCube.resource[index].ty = "3";
141     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn;
142     nCube.resource[index].rn = "led-ctrl";
143     nCube.resource[index++].status = 0;
144
145     nCube.resource[index].ty = "3";
146     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn;
147     nCube.resource[index].rn = "relay-ctrl";
148     nCube.resource[index++].status = 0;
149
150     // Subscription resource
151     nCube.resource[index].ty = "23";
152     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn + '/' + nCube.resource[2].rn;
153     nCube.resource[index].rn = "led-sub";
154     nCube.resource[index++].status = 0;
155
156     nCube.resource[index].ty = "23";
157     nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn + '/' + nCube.resource[3].rn;
158     nCube.resource[index].rn = "relay-sub";
159     nCube.resource[index++].status = 0;
160
161     nCube.resource_count = index;
162 }
```

```
299 pinMode(LED_BLUE_PIN, OUTPUT);
300 pinMode(LED_GREEN_PIN, OUTPUT);
301 pinMode(LED_RED_PIN, OUTPUT);
302 pinMode(LED_GND_PIN, OUTPUT);
303 pinMode(RELAY_PIN, OUTPUT);
304
305 digitalWrite(LED_BLUE_PIN, LOW);
306 digitalWrite(LED_GREEN_PIN, LOW);
307 digitalWrite(LED_RED_PIN, LOW);
308 digitalWrite(LED_GND_PIN, LOW);
309 digitalWrite(RELAY_PIN, LOW);

324 TasCO2Sensor.init();
325 TasCO2Sensor.setCallback(upload_callback);

339 if (state == "create_cin") {
340     // guide: in here generate sensing data
341     // if get sensing data directly, assign curValue sensing data and set sensing_flag to 1
342     // if request sensing data to sensor, set sensing_flag to 0, in other code of receiving
343     TasCO2Sensor.requestData();
344     sensing_flag = 0;
345 }
346 }
```

nCube-Mint 제작 가이드

```
359 if(control_flag == 1) {  
360     control_flag = 0;  
361  
362     if (noti_con == "0") {  
363         digitalWrite(LED_BLUE_PIN, LOW);  
364         digitalWrite(LED_GREEN_PIN, LOW);  
365         digitalWrite(LED_RED_PIN, LOW);  
366     }  
367     else if (noti_con == "1") {  
368         digitalWrite(LED_BLUE_PIN, LOW);  
369         digitalWrite(LED_GREEN_PIN, LOW);  
370         digitalWrite(LED_RED_PIN, HIGH);  
371     }  
372     else if (noti_con == "2") {  
373         digitalWrite(LED_BLUE_PIN, LOW);  
374         digitalWrite(LED_GREEN_PIN, HIGH);  
375         digitalWrite(LED_RED_PIN, LOW);  
376     }  
377     else if (noti_con == "3") {  
378         digitalWrite(LED_BLUE_PIN, HIGH);  
379         digitalWrite(LED_GREEN_PIN, LOW);  
380         digitalWrite(LED_RED_PIN, LOW);  
381     }  
382     else if (noti_con == "4") {  
383         digitalWrite(LED_BLUE_PIN, HIGH);  
384         digitalWrite(LED_GREEN_PIN, HIGH);  
385         digitalWrite(LED_RED_PIN, LOW);  
386     }  
387     else if (noti_con == "5") {  
388         digitalWrite(LED_BLUE_PIN, HIGH);  
389         digitalWrite(LED_GREEN_PIN, LOW);  
390         digitalWrite(LED_RED_PIN, HIGH);  
391     }  
392     else if (noti_con == "6") {  
393         digitalWrite(LED_BLUE_PIN, LOW);  
394         digitalWrite(LED_GREEN_PIN, HIGH);  
395         digitalWrite(LED_RED_PIN, HIGH);  
396     }  
397     else if (noti_con == "7") {  
398         digitalWrite(LED_BLUE_PIN, HIGH);  
399         digitalWrite(LED_GREEN_PIN, HIGH);  
400         digitalWrite(LED_RED_PIN, HIGH);  
401     }  
  
410     if (noti_con == "0") {  
411         digitalWrite(RELAY_PIN, LOW);  
412     }  
413     else if (noti_con == "1") {  
414         digitalWrite(RELAY_PIN, HIGH);  
415     }  

```

420 | TasCO2Sensor.chkCO2Data();

5 - 3 nCube-Mint 동작과 데이터 조회, 제어

스케치를 업로드 하고, 스마트폰을 이용하여 nCube-Mint 보드가 이용할 WiFi AP의 SSID 와 Password 를 다시 입력한다.



```
REQUEST OK!
{"rsc":2001,"rqi":"tv0x2mCi","pc":{"m2m:cin":{"rn":"4-20170710071259960dhE7","ty":4,"pi":"rkGfukYlBZ","ri":"rkMEbKi
Receive published data
/oneM2M/resp/mint/mobius-yt/json - RESP_TOPIC receive a message.
Read: ♦♦

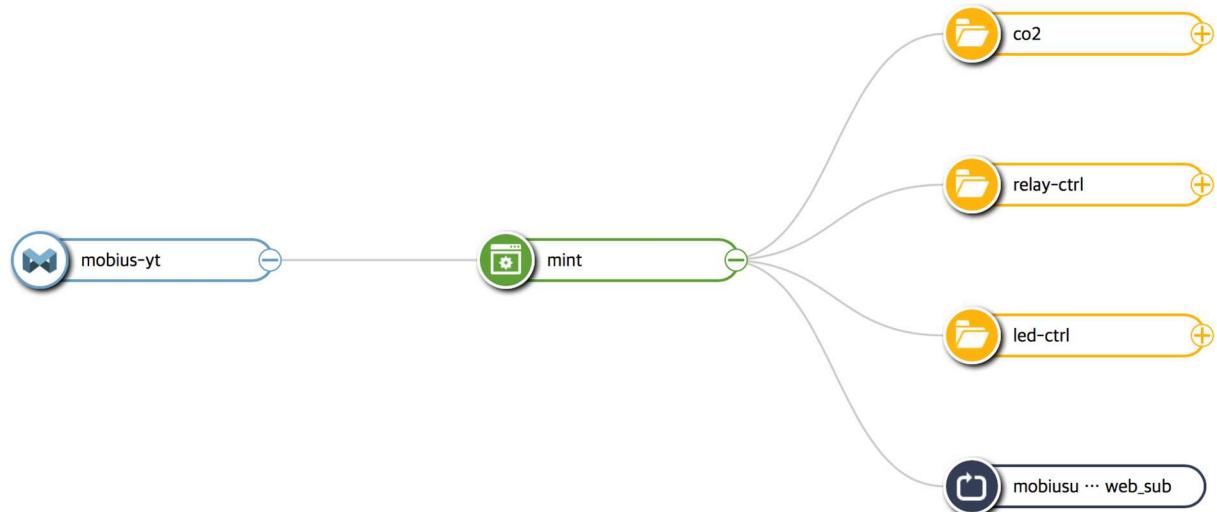
{"op":1,"to":"/mobius-yt/mint/co2","fr":"mint","rqi":"eW5NegPr","ty":4,"pc":{"m2m:cin":{"con":"2000.00"}}
REQUEST OK!

{"rsc":2001,"rqi":"eW5NegPr","pc":{"m2m:cin":{"rn":"4-20170710071304955VbPH","ty":4,"pi":"rkGfukYlBZ","ri":"S1fY-to
Receive published data
/oneM2M/resp/mint/mobius-yt/json - RESP_TOPIC receive a message.
Read: ♦♦

{"op":1,"to":"/mobius-yt/mint/co2","fr":"mint","rqi":"6JkyCVXd","ty":4,"pc":{"m2m:cin":{"con":"2000.00"}}
REQUEST OK!

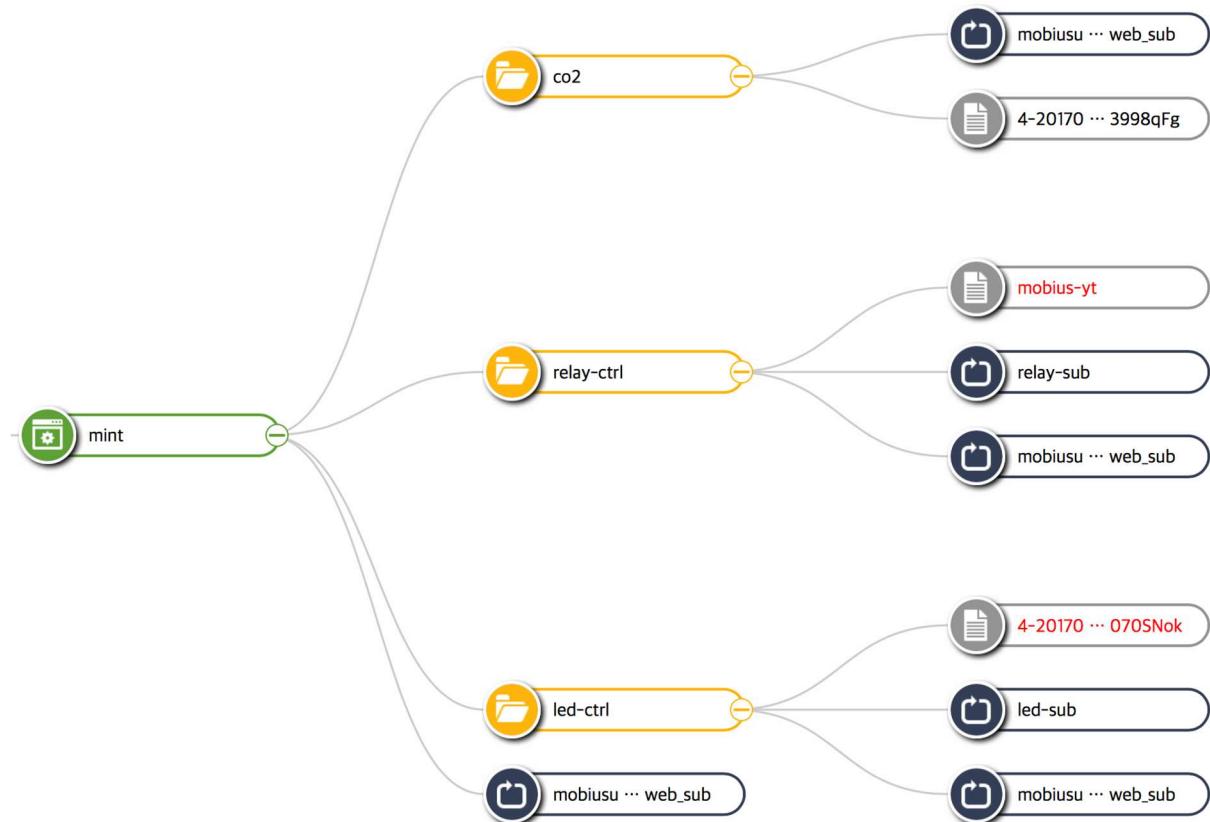
{"rsc":2001,"rqi":"6JkyCVXd","pc":{"m2m:cin":{"rn":"4-20170710071309945d0WY","ty":4,"pi":"rkGfukYlBZ","ri":"HyzR-Fi
Receive published data
/oneM2M/resp/mint/mobius-yt/json - RESP_TOPIC receive a message.
```

시리얼 모니터를 통해 5 초 간격으로 Co2 농도를 측정하여 업로드 하는 보습을 볼 수 있다.

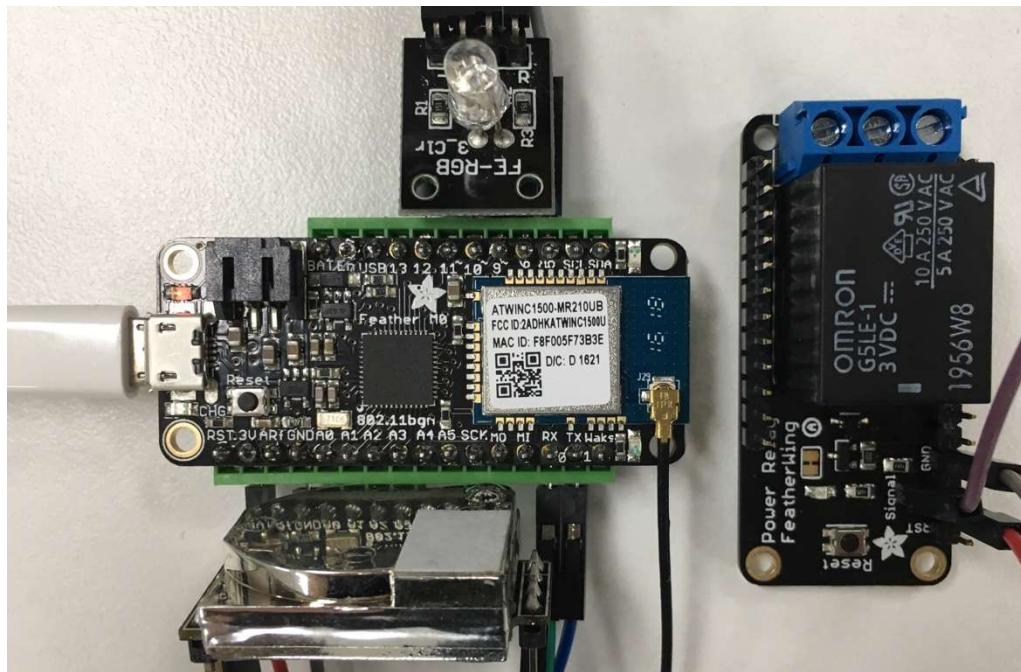


Web 버전의 Mobius Resource Monitor 를 통해 확인한 최종적인 nCube-Mint 의 리소스 구조는 위의 그림과 같다.

nCube-Mint 제작 가이드



co2 cnt 에 현재 대기의 Co2 농도를 측정 데이터가 업로드 되는것을 확인할 수 있으며,
led-ctrl, relay-ctrl cnt 에 cin 을 생성하여 RGB-LED 와 Power Relay 를 제어할 수 있다.



Appendix A.

WiFi101, PubSubClient 라이브러리 업데이트 후,
nCube-Mint 개발을 위해 변경해야할 부분

WiFi101 라이브러리 업데이트 후,
Arduino, libraries, WiFi101, src 경로상의

WiFi.cpp 에서 추가되어야 할 부분은 다음과 같다.

```
139     memset(WiFi._password, 0, M2M_MAX_SSID_LEN);
140     memcpy(WiFi._password, (char *)pstrProvInfo->au8Password, strlen((char *)pstrProvInfo->au8Password));

561     uint8_t WiFiClass::beginProvision2()
562     {
563         return beginProvision2(1);
564     }
565
566     uint8_t WiFiClass::beginProvision2(uint8_t channel)
567     {
568         uint8_t mac[6];
569         char provSsid[13];
570
571         // get MAC address for provisioning SSID
572         macAddress(mac);
573         sprintf(provSsid, "wifi101-%.2X%2X", mac[1], mac[0]);
574
575         // start provisioning mode
576         startProvision(provSsid, "wifi101", channel);
577
578         return status();
579     }

750     char* WiFiClass::PASSWORD()
751     {
752         if (_status == WL_CONNECTED || _status == WL_AP_LISTENING || _status == WL_AP_CONNECTED) {
753             return _password;
754         }
755         else {
756             return 0;
757         }
758     }
```

WiFi101.h 에서 추가되어야 할 부분은 다음과 같다.

```
95     char _password[M2M_MAX_SSID_LEN];
```

```
135     uint8_t beginProvision2();  
136     uint8_t beginProvision2(uint8_t channel);
```

```
156     char* PASSWORD();
```

PubSubClient 라이브러리 업데이트 후,

Arduino, libraries, PubSubClient, src 경로상의

PubSubClient.h 에서 수정되어야 할 부분은 다음과 같다.

```
26 #define MQTT_MAX_PACKET_SIZE 128
```



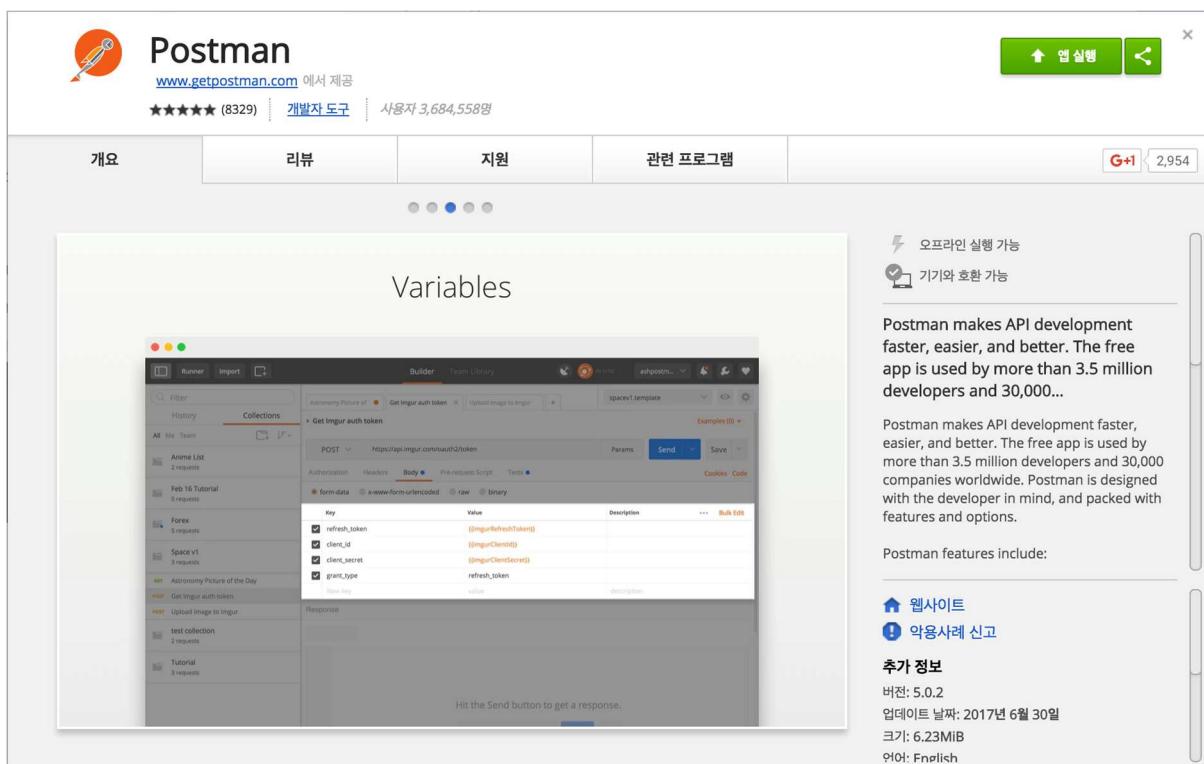
```
26 #define MQTT_MAX_PACKET_SIZE 400
```

Appendix B.

Postman 을 활용한 nCube-Mint Resource 조회와 제어

Google Chrome 의 Postman 앱을 이용해서 nCube-Mint 보드가 업로드 한 Co2 데이터를 확인하고, LED 와 Relay 를 제어해본다. 이외에도 API 를 활용한 AE, CNT, CIN 생성 및 조회가 가능하다.

https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcdcbncddomop?utm_source=chrome-ntp-icon



Google Chrome 을 이용해서 위의 URL 에 접속해 Postman 앱을 다운로드 및 설치하고, 실행시킨다.

nCube-Mint 제작 가이드

The screenshot shows the OCEAN website's download section for the Mobius Yellow Turtle software. The left sidebar lists various partners and projects, including KETI, IIIP, and the Ministry of Science, ICT and Future Planning. The main content area displays the 'Yellow Turtle' version 2.3.8, which is described as an open source software for oneM2M-based IoT Server Platform based on Node.js Java Script. A table lists several files, with the 'POSTMAN Environment Script' file highlighted by a red box.

Name	Download Link
Mobius Installation Guide Korea v2.3.4_KR.pdf	Installation_Guide_Mobius_Yellow_Turtle_v2.3.4_KR.pdf
Mobius Installation Guide English v2.0_ENG.pdf	Installation_Guide_Mobius_Yellow_Turtle_v2.0_ENG.pdf
REST API for Mobius Yellow Turtle REST_API_for_Mobius_Yellow_Turtle_v2.0_ENG.pdf	REST_API_for_Mobius_Yellow_Turtle_v2.0_ENG.pdf
Mobius Yellow Turtle Source mobius-yt-v2.3.8.zip	mobius-xt-v2.3.8.zip
Mysql DB Script Mobiusdb_v2.2.20.sql (TableSpace Regeneration_Pending)	Mobiusdb_v2.2.20.sql (TableSpace Regeneration_Pending)
POSTMAN Environment Script mobius-xt-7579.postman_environment.json	mobius-xt-7579.postman_environment.json
POSTMAN Script mobius_xt_release1.postman_collection.json	mobius_xt_release1.postman_collection.json
	test-as-virtual-device.postman_collection.json
	test-for-group-resource-fanoutpoint.postman_collection.json

<http://www.itocean.org> 의 Download, Mobius Yellow Turtle, POSTMAN Environment Script 아래의 mobius-xt-7579.postman_environment.json 을 클릭하여 Postman API collection 을 다운로드 받는다.

The screenshot shows the Postman application interface. The left sidebar displays a collection named 'mobius-xt-release1' with 41 requests. The main workspace shows the 'Builder' tab with a 'GET' request to 'Enter request URL'. The 'Authorization' tab is selected, showing 'No Auth'. The 'Response' tab shows a placeholder message: 'Hit the Send button to get a response.' At the bottom, there is a note: 'Work faster with Postman Watch and learn'.

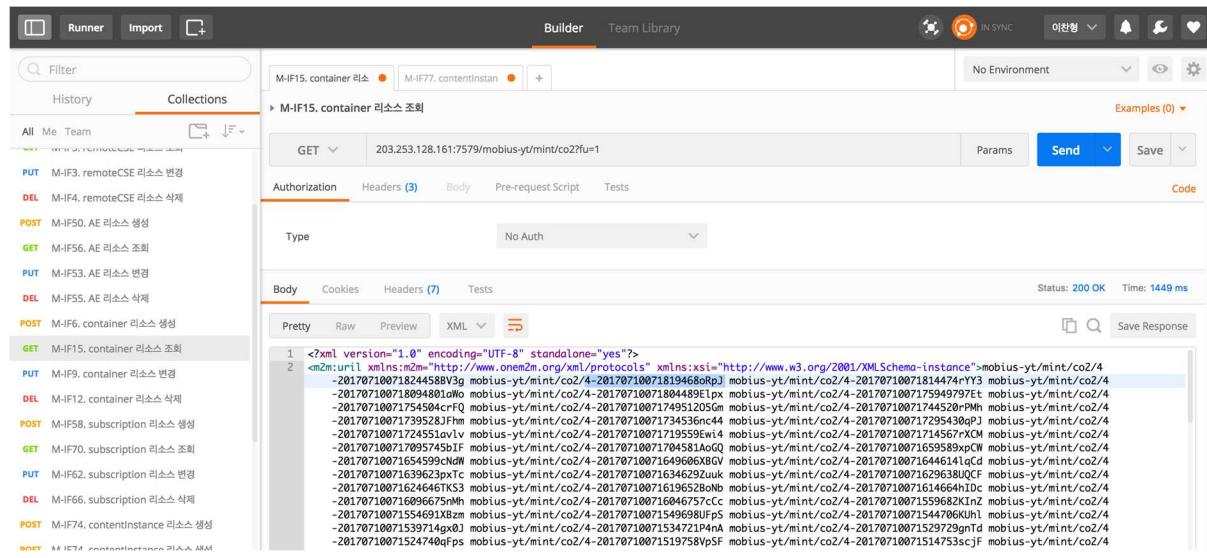
Postman에서 import → import file → choose files 를 클릭하여 위에서 다운로드 받은 API collection 을 열면 Postman 좌측의 Collections 목록 아래에 API 들이 추가된 것을 볼 수 있다.

nCube-Mint 제작 가이드

추가된 API를 이용하여 Co2 데이터를 담고 있는 cin 조회, RGB-LED, Power Relay를 제어하기 위한 cin 생성을 할 수 있다.

좌측 목록에서 container 리소스 조회를 선택하고, URL 부분에 다음과 같이 입력하고, 파란색 Send 버튼을 누르면, co2 cnt 내에 생성된 cin들의 목록을 볼 수 있다.

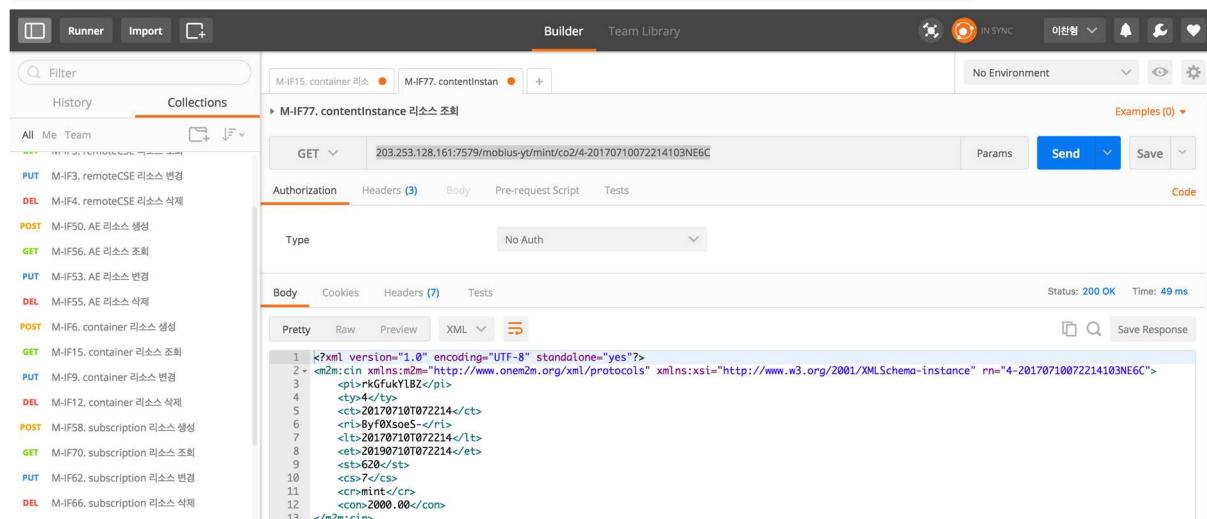
203.253.128.161:7579/mobius-xt/mint/co2?fu=1



The screenshot shows the Postman application interface. The 'Builder' tab is selected. A GET request is made to the URL `203.253.128.161:7579/mobius-xt/mint/co2?fu=1`. The response status is `200 OK` and the time taken is `1449 ms`. The response body is a large XML document representing the CO2 sensor data, which is partially visible in the screenshot.

하나의 cin 이름을 복사하여두고, 좌측 목록에서 contentinstance 리소스 조회를 선택하고, URL 부분에 다음과 같이 입력하고 Send 버튼을 누르면 cin 내의 con 필드 아래에 업로드된 Co2 농도를 확인할 수 있다.

203.253.128.161:7579/mobius-xt/mint/co2/[contentinstance resource name]

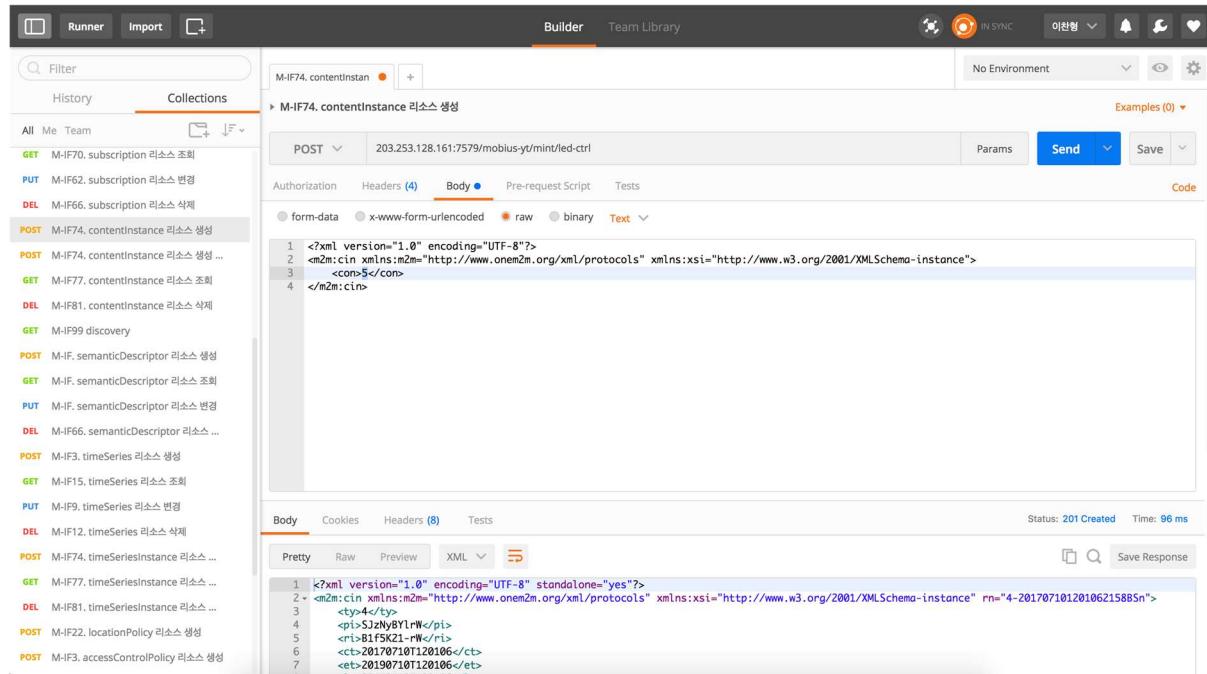


The screenshot shows the Postman application interface. The 'Builder' tab is selected. A GET request is made to the URL `203.253.128.161:7579/mobius-xt/mint/co2/4-20170710072214103NE6C`. The response status is `200 OK` and the time taken is `49 ms`. The response body is a single XML document representing the CO2 sensor data.

nCube-Mint 제작 가이드

RGB-LED 와 Power Relay 를 제어하기 위해 좌측 목록에서 contentinstance 리소스 생성 을 선택하고, URL 부분에 각각 다음과 같이 입력하고 Body 의 con 항목 아래에 적절한 값을 입력하여 Send 버튼을 누르면 RGB-LED 와 Power Relay 제어가 가능하다..

203.253.128.161:7579/mobius-xt/mint/led-ctrl



The screenshot shows the Postman application interface. The left sidebar lists various API endpoints under the 'Collections' tab. The main area displays a POST request to the URL '203.253.128.161:7579/mobius-xt/mint/led-ctrl'. The 'Body' tab is selected, showing the following XML code:

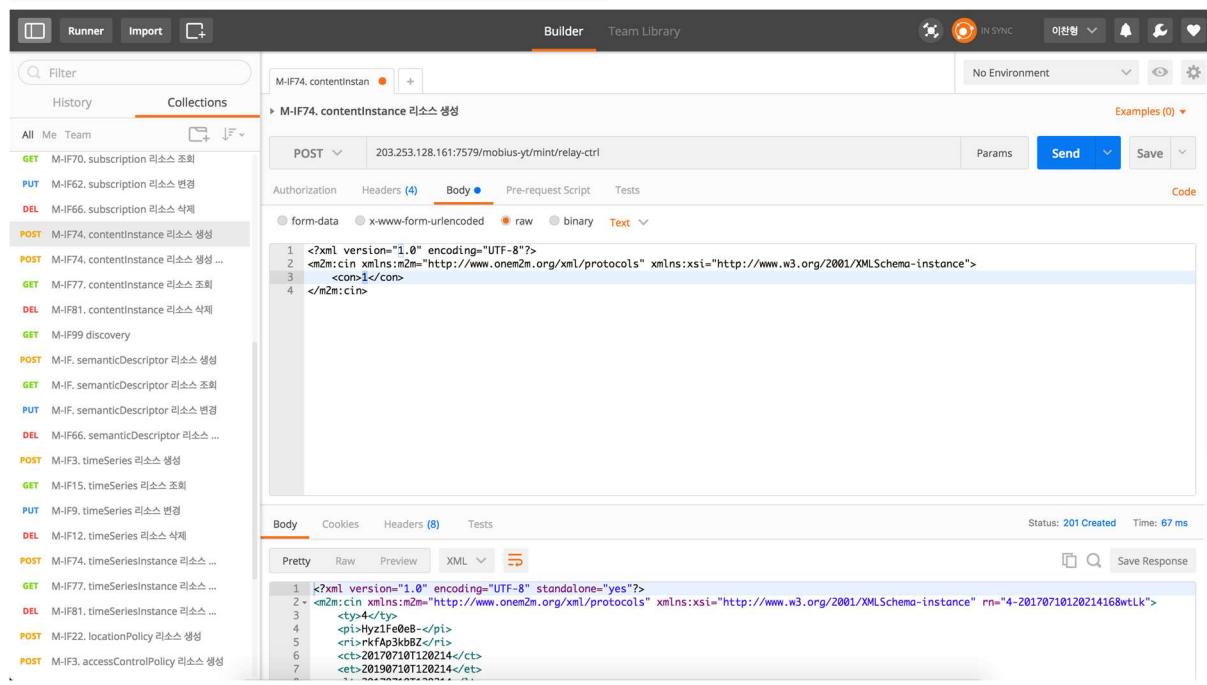
```
<?xml version="1.0" encoding="UTF-8"?>
<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<con>5</con>
</m2m:cin>
```

Below the code, the 'Pretty' tab is selected, showing the expanded XML structure:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" rn="4-2017071012010621588n">
3   <ty>4</ty>
4   <pi>SJZnX8YlW</pi>
5   <ri>B1F5K21-rh</ri>
6   <ct>20170710T120106</ct>
7   <et>20190710T120106</et>
```

The status bar at the bottom indicates 'Status: 201 Created' and 'Time: 96 ms'.

203.253.128.161:7579/mobius-xt/mint/relay-ctrl



The screenshot shows the Postman application interface. The left sidebar lists various API endpoints under the 'Collections' tab. The main area displays a POST request to the URL '203.253.128.161:7579/mobius-xt/mint/relay-ctrl'. The 'Body' tab is selected, showing the following XML code:

```
<?xml version="1.0" encoding="UTF-8"?>
<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<con>1</con>
</m2m:cin>
```

Below the code, the 'Pretty' tab is selected, showing the expanded XML structure:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" rn="4-20170710120214168wLk">
3   <ty>4</ty>
4   <pi>JzTf0eB</pi>
5   <ri>4TAq3kbZ-zr1>
6   <ct>20170710T120214</ct>
7   <et>20190710T120214</et>
```

The status bar at the bottom indicates 'Status: 201 Created' and 'Time: 67 ms'.