OCEAN

nCube-Thyme (Arduino)

Version: 1.0

Developer Guide v1.0

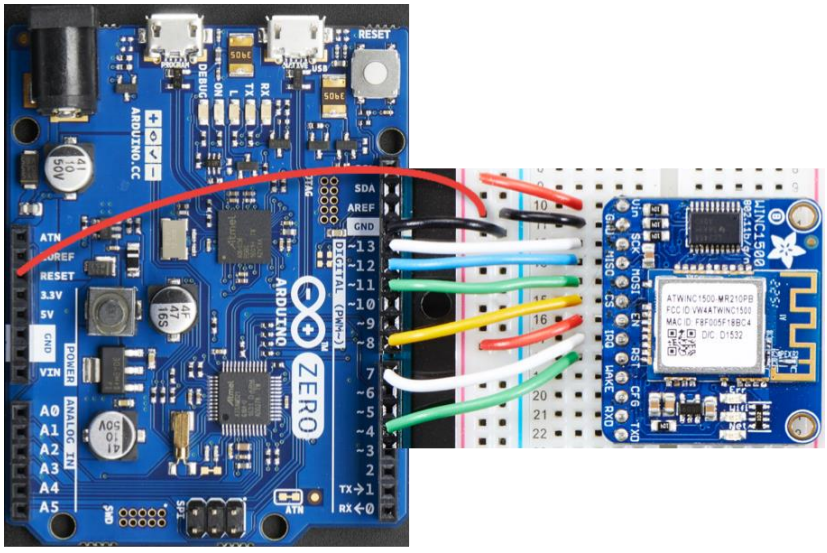
Document Release Date: July 20, 2017

OCEAN

nCube-Mint

nCube-Mint is an Arduino version of oneM2M Application Entity(AE) developed in C, C++ and supported by Arduino IDE. nCube-Mint is functionally identical with nCube: thyme and uses MQTT protocol.

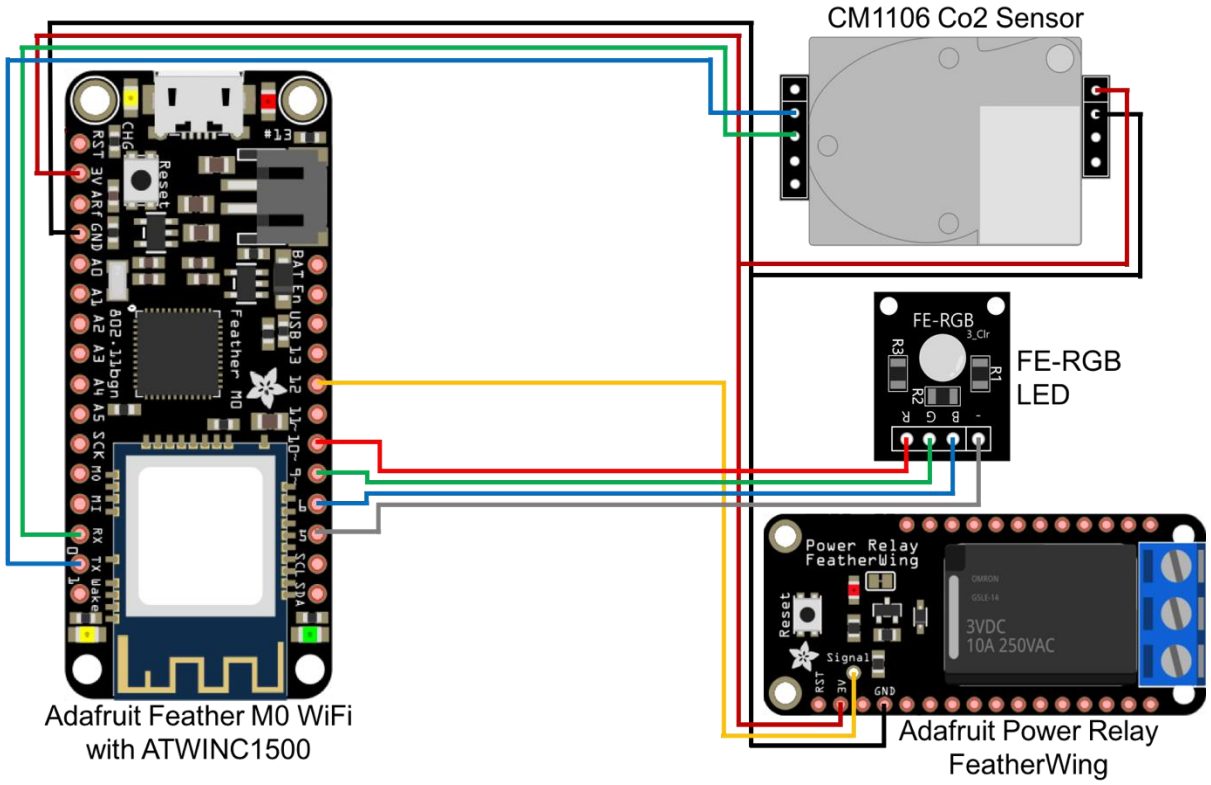
nCube-Mint hardware is targeted for Arduino Zero which is based on cortex-M0. WiFi modules are required for MQTT communication using wireless internet. This document is based on Feather M0 of Adafruit corporation which includes Cortex-M0 of Arduino Zero and WiFi modules as a one module.



or

This document is intended to explain the process of uploading various information in everyday life using Mobius IoT Platform and making nCube-Mint that provides motion control of the user. The applications of nCube-Mint such as measurement of Co2 and upload data, LED and Relay control are provided with an example.

Based on this document it is possible to create various low-power IoT Devices to detect temperature, humidity, Co2 concentration and the infrared sensor that captures the presence or the direction of movements of the people.



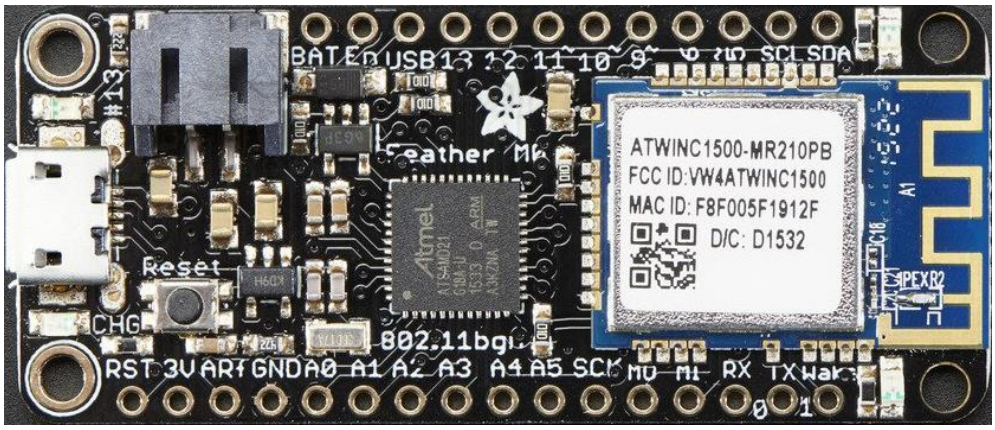
The figure above is the working prototype nCube-Mint and the connection details of the three sensors.

Contents

1. Hardware

In chapter 1, hardware specifications for the development of nCube-Mint will be introduced in order of Adafruit Feather M0, Co2 sensor, RGB-LED sensor and Adafruit Power Relay FeatherWing.

1.1. Adafruit Feather M0



Adafruit includes Cortex M0 processor and produced by Adafruit ©. A total of six SERCOM that supports I/O pins, several 12-bit ADC (analog-digital converting circuit), one 10-bit DAC (digital-analog converting circuit), SPI, I2C, UART communication is included and native USB, Atmel WINC1500 WiFi chip that supports 802.11bgn is included as well.

Adafruit Feather M0 is suitable for the IoT devices with low power designs. The WiFi chip includes a power management chip and has a power consumption of 12mA and 10mA for the Cortex M0 chip. The battery will be charged when it is connected with MicroUSB or 3.7V Lithium Polymer battery. From the figure above, 3V pin located on the bottom left of the chip, maximum of 600mA power output is supported.

1.2. Co2 Sensor

The M1106 Co2 sensor measures the Co2 values from the air and uploads the value to Arduino using the ART port. Output values of 8 bytes (0x16, 0x05, 0x01, 0x02, 0x72, 0x01, 0xD6, 0x9) can be obtained from the input values of 4 bytes (0x11, 0x01, 0x01, 0xED). The

output value of 5, 6 bytes (0x02, 0x72) indicates the Co2 concentration. For instance, 0x0272 = 626 is identical with 626ppm value of the Co2 concentration.

1.3. RGB-LED

FE-RGB 3-color LED board operates red, green and blue LEDs depending on the input of R, G, B pins. The combination of input values of lights (2(LED On, Off)^3(LED three types)-1(RGB off case) = 7 types) including three basic red, green and blue can be created.

1.4. Adafruit Power Relay FeatherWing

Adafruit Power Relay FeatherWing board selectively flows electric current into two external power supply from the input of 3V and digital input signal. The external power supply supports electric devices controls including TV and radio.

2. Environment

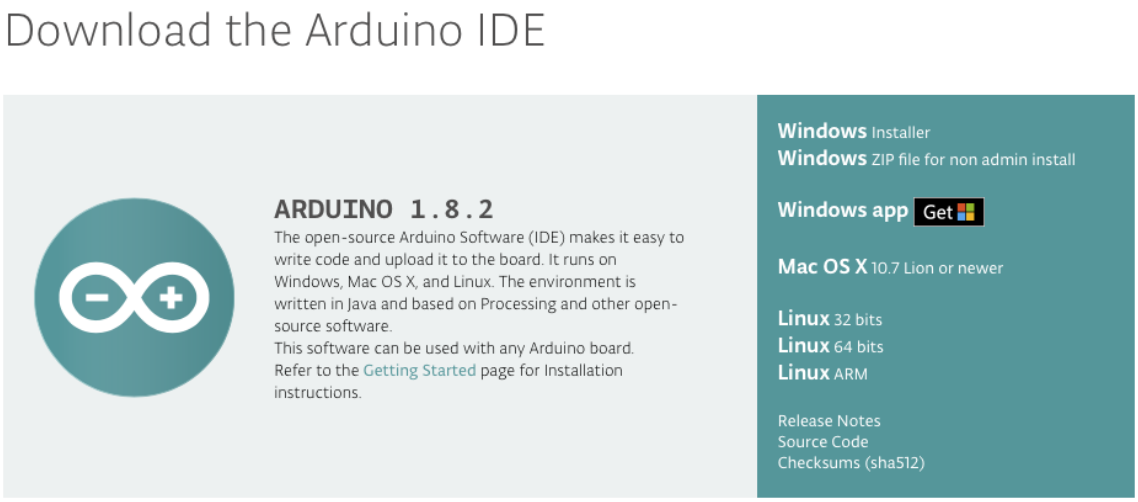
In chapter 2, environmental setting for nCube-Mint development is explained including the installation of Arduino IDE, Adafruit Feather M0 board manager, and Adafruit Feather M0 testing with the examples.

This document is based on MacOS, basic features of installation would be similar in Windows OS, however, additional explanations will also be included if necessary.

2.1. Arduino IDE Installation

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

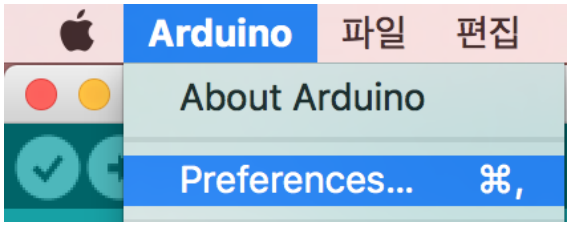
Go to Arduino homepage 'SOFTWARE' tab through the link above. Either login or sign up is not necessary.



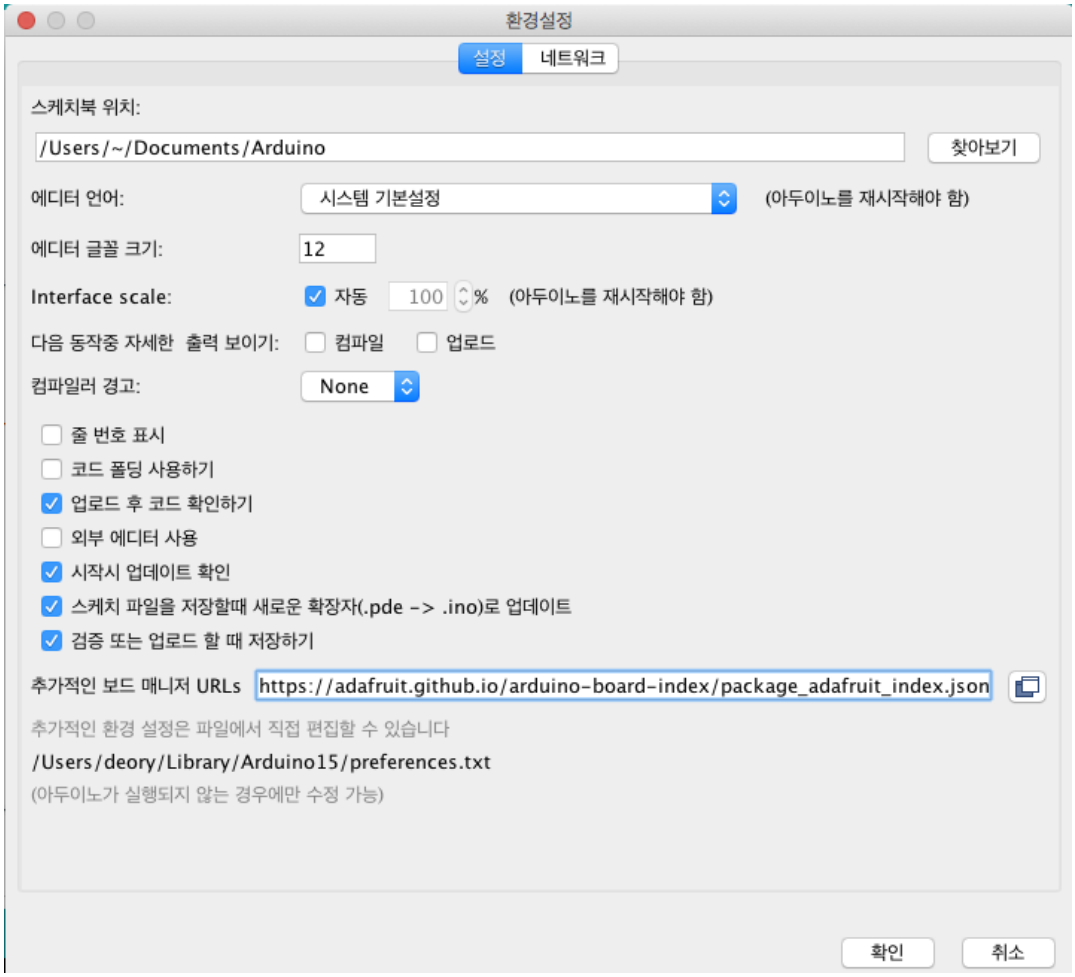
Select compatible OS on the right side of 'Download the Arduino IDE' tab and download Arduino IDE. We recommend to download Arduino IDE v1.6.4. or the later releases.

2.2. Arduino IDE Environment Setting

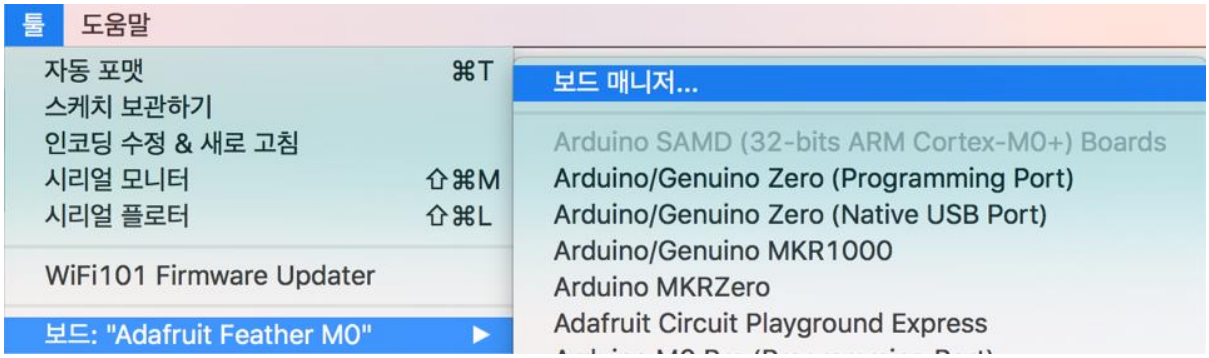
The environment setting for Arduino IDE to compile nCube-Mint Sketch and to upload Adafruit Feather M0 is explained as follows.

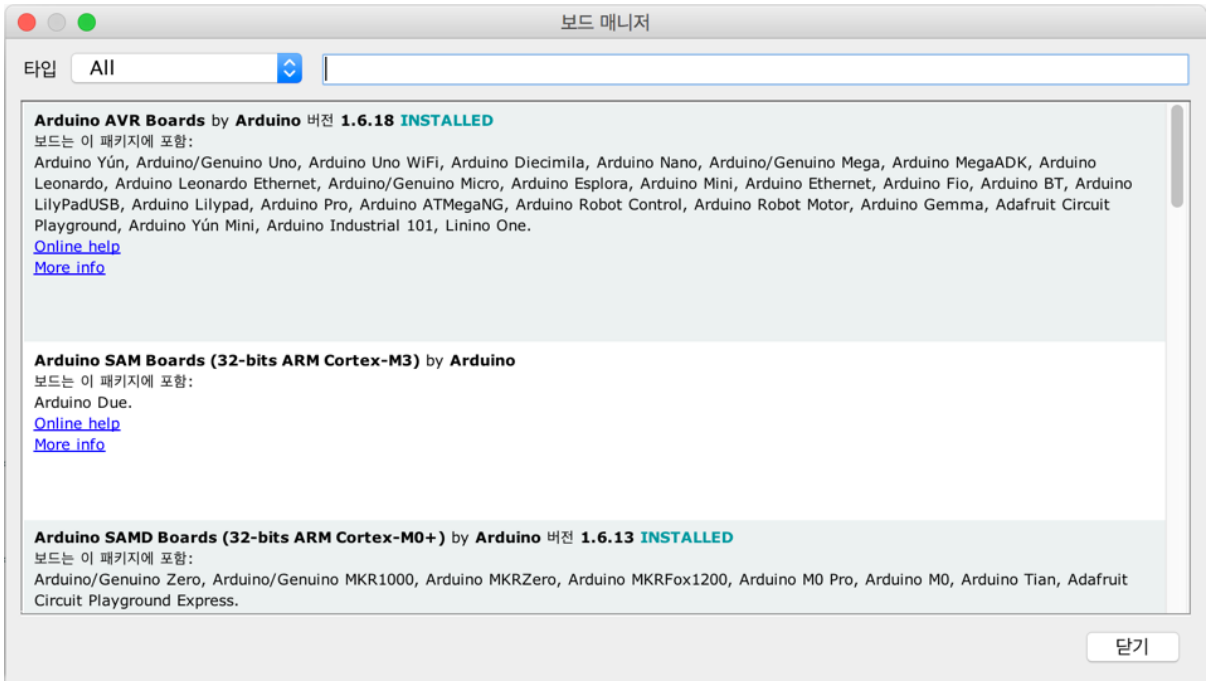


Run Arduino IDE and go to *Preferences* from the *Arduino* tab.

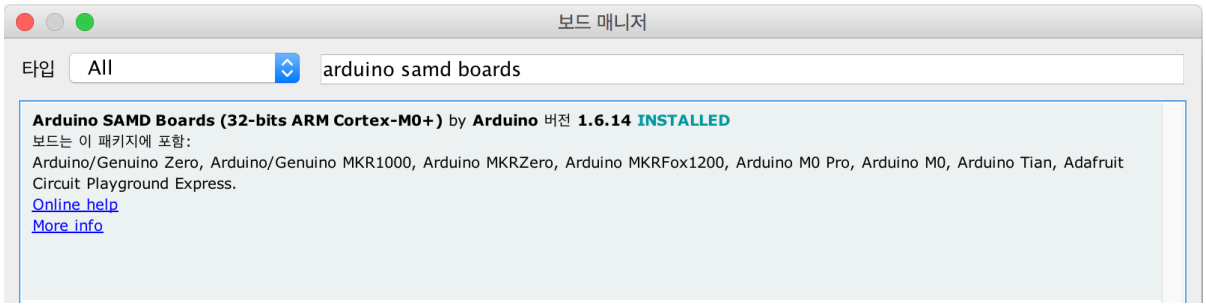


Add *https://adafruit.github.io/arduino-board-index/package_adafruit_index.json* in *Additional Boards Manager* URLs then save and quit.





Execute *Boards Manager* in Tools>Board>BoardsManager...



Search *Arduino SAMD Boards* and install the package. The version mush be 1.6.2 or the later releases.

Restart *Arduino IDE* and execute *Board Manager*. Search *Adafrut SAMD Boards* and install

the package.

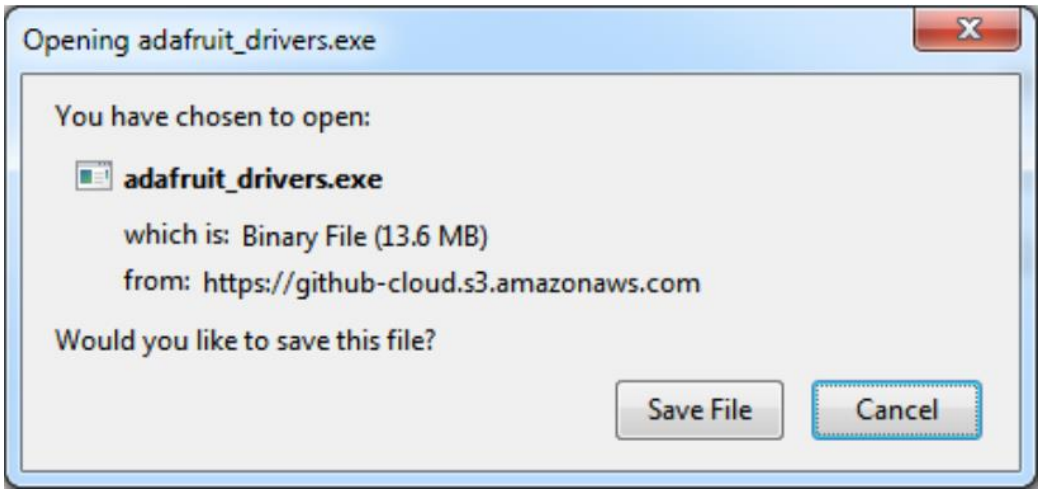
If *Adafruit Feather M0* appears in the *Tool* tab, Arduino IDE Setup is successfully compeleted.

* Adarfruit Windows Driver Installation

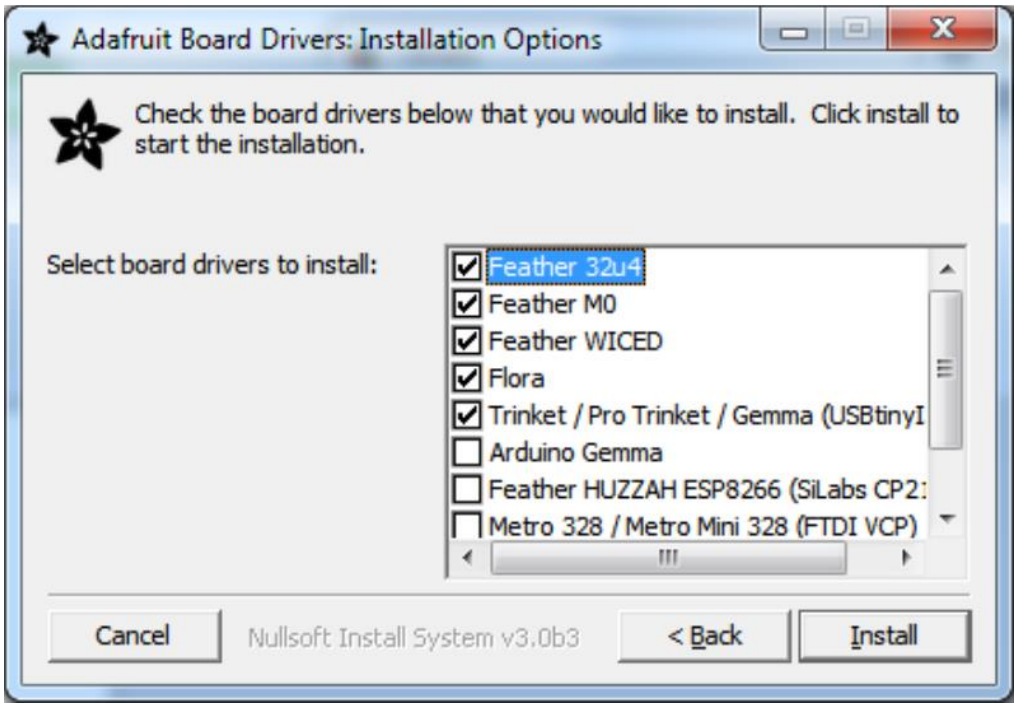
In Windows OS environment, *Adafruit WindowsDriver* should be additionally installed.

go to http://adafru.it/mai

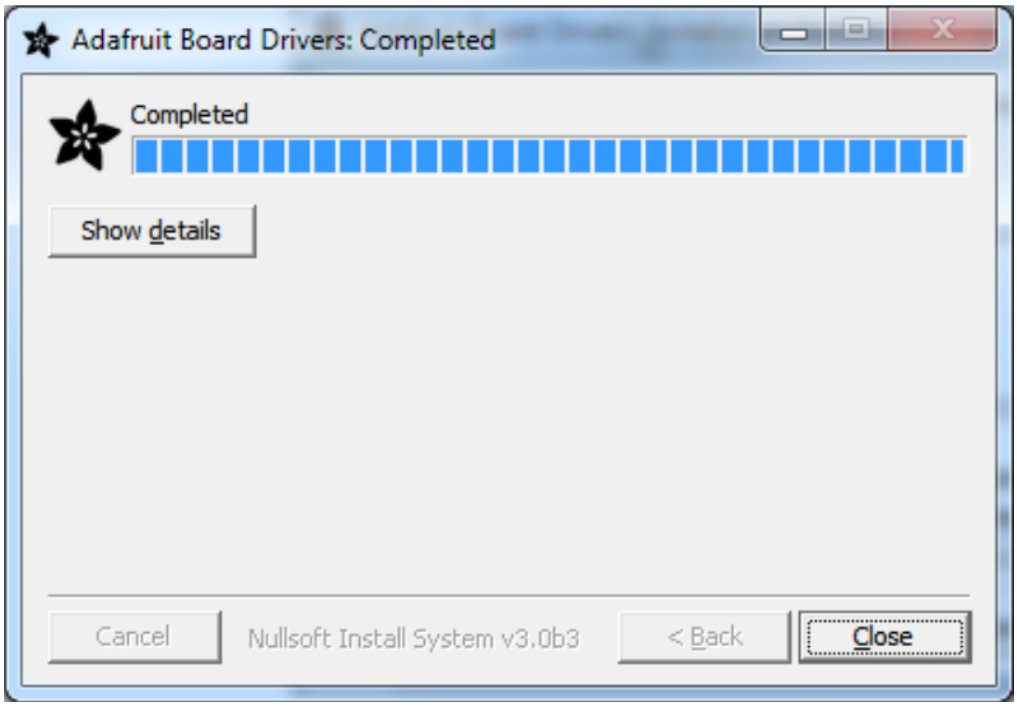
From the URL above, download *adafruit_driver.exe* file and execute.



Execute *installer* and click agree.

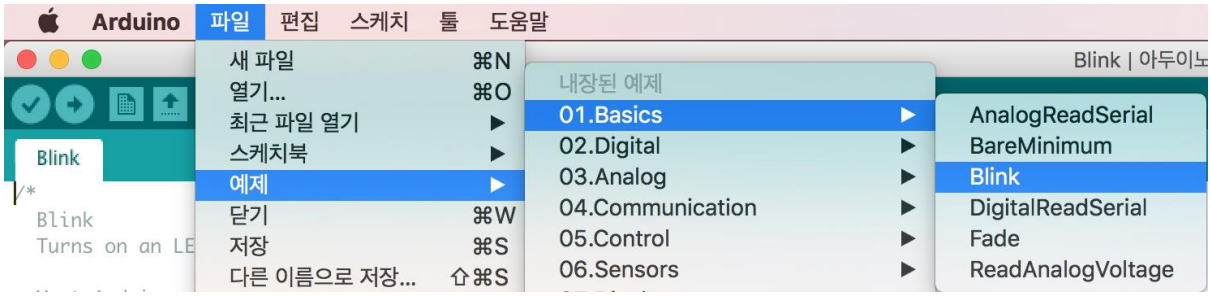


From the figure above, select preferable board types. In this document select *Feather M0* and click install.

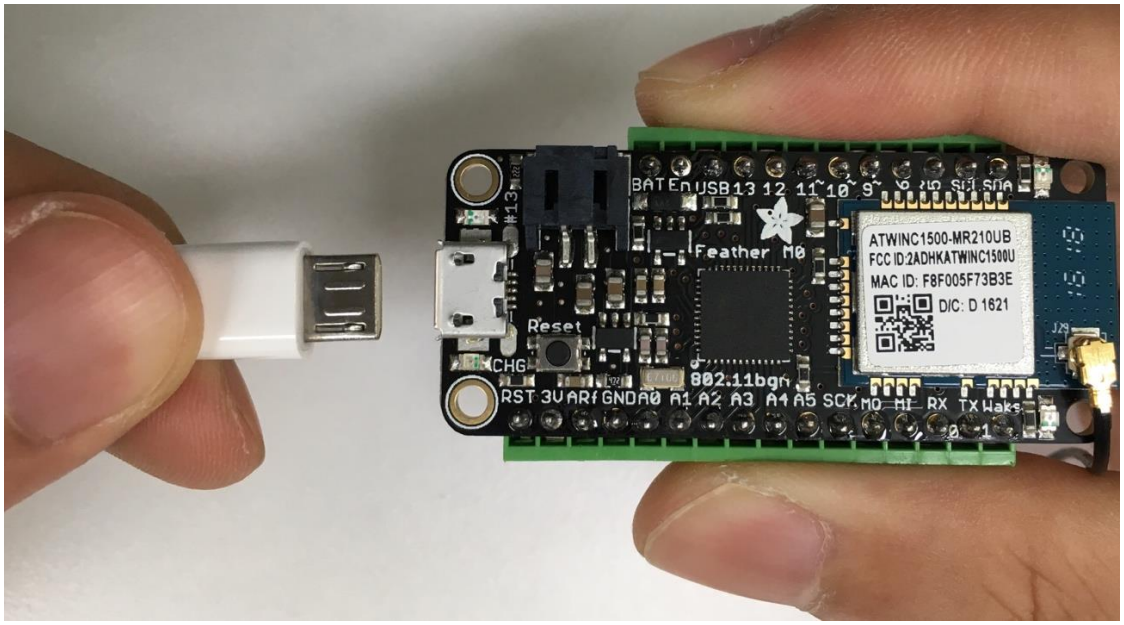


Adafruit Drivers installation is successfully completed.

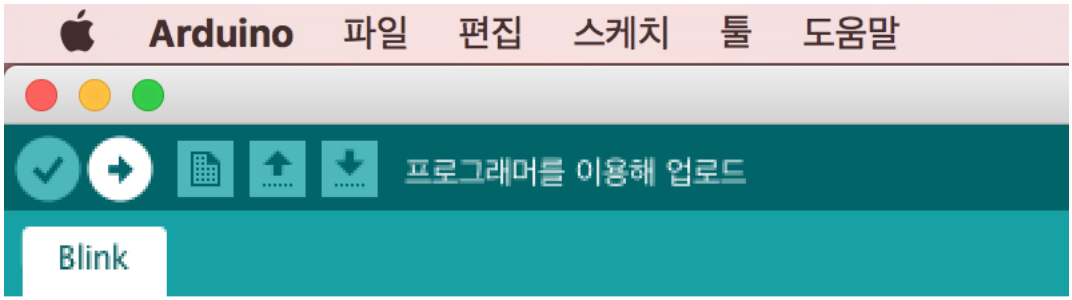
2.3. Adafruit Feather M0 Testing



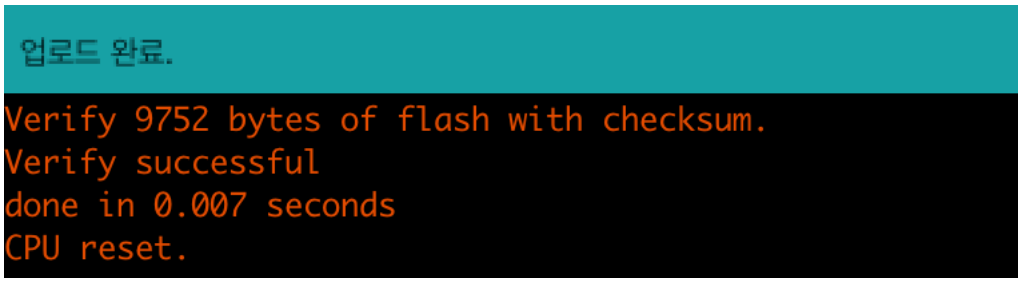
Open *Blink* example from the tab *File>Examples>01.Basics*.



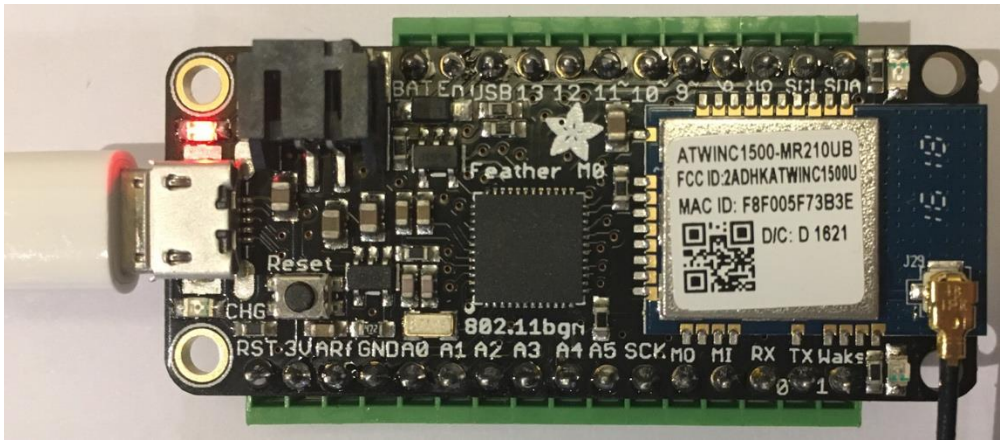
Connect Adafruit Feather M0 and computer with USB-5pinMicro USB cable. Set *Tool>Board* and *Tool>Port* with Adafruit Feather M0. If it is already set as the figure above, then go to the next step.



As the figure above, click upload button as shown in white button. Compile the *Blink* example and upload to *Adafruit Feather M0*.



The figure above will be shown if uploaded successfully.



The red LED in Adafruit Feather M0 will blink in 1 second intervals.

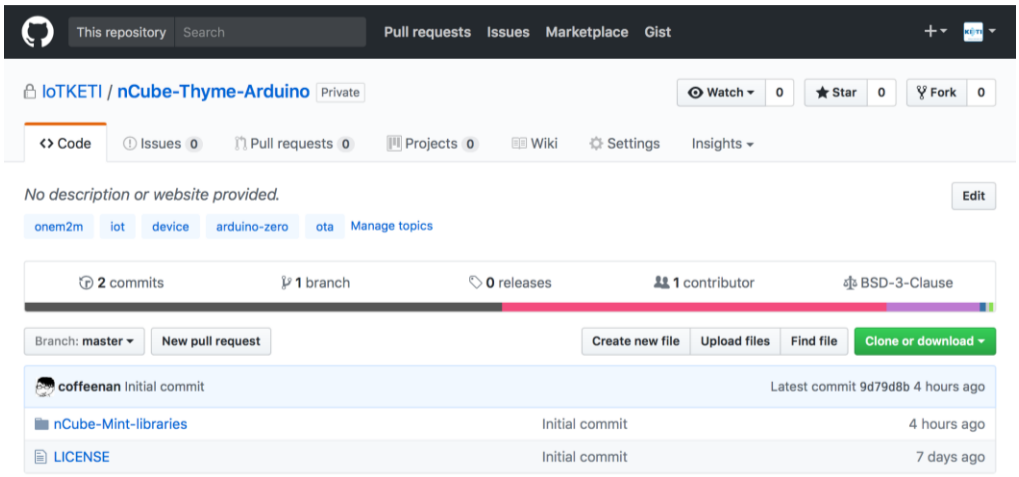
This is the end of Adafruit Feather M0 installation and testing for nCube-Mint development environment setting.

3. nCube-Mint Development Environment

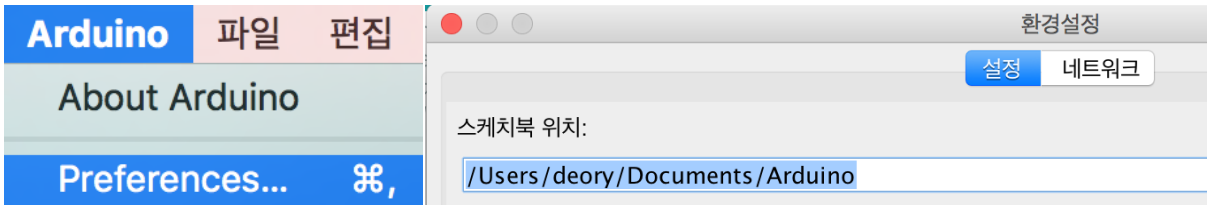
In chapter 3, nCube-Mint board set up with Adafruit Feather M0 installation using nCube-Mint Sketch examples under oneM2MClient library is explained. Moreover, essential libraries to set up nCube-Mint develop environment will be explained.

3.1. nCube-Mint Library Download

http://github.com/IoTKETI/nCube-Thyme-Arduino



From the figure above, download nCube-Mint and unzip. Locate library directories in *Libraries>nCube>Mint* under *Arduino>Preferences>Sketchbook>Libraries* directory.



Libraries under *Libraries>nCube>Mint* are shown as the figure above. After locating the libraries, restart Arduino IDE.

3.2. oneM2MClient Library

oneM2MClient is the basic library for nCube-Mint development that supports fundamental actions in nCube-Mint using functions.

It creates objects as follows to access variables, structures and to call functions.

resource
resource_count

AE, Cnt, Cin information for nCube-Mint is stored using *resource* structure in oneM2MClient library and the size of arrays in a structure is stored using *resource_count*.

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)

Create AE, create Cnt, create/delete Subscription, and create Cin actions are available by calling the functions above.

Variable String rqi refers to Request ID and variable int index refers to the resource structure that holds the information of corresponding function.

For the case of createAE, additional parameter values should be added as Application ID value.

response(body_buff)

Response function is for sending response packet after conducting LED or Relay control packet. Payload data of the response packet should be specified.

setCallback(resp_callback, noti_callback)

setCallback function is for designating callback function when response packet of notification packet arrives. Parameter orders are as above: response callback function, notification callback function name.

begin()

If the values for WiFi SSID and Password are stored in Flash memory, WiFi connection trial will begin to the corresponding information. If there is no value, WiFi SSID and Password values can be added for the connection of WiFi Access Point with Adafruit Feather M0.

chkInitProvision()
chkInitProvision2()

When WiFi reset button is clicked once in a short term, the values of stored WiFi SSID and Password will be presented in Serial Monitor. If it is clicked once in a long term, it deletes WiFi SSID and Password values in Flash Memory, and actuates Adafruit Feather M0 with WiFi Access Point.

chkInitProvision() and chkInitProvision2() is for the time interval of button clicks for deleting Flash memory values. The default setting for two different actions are set as relative click terms - short and long.

chkConnect()

Check on the WiFi connection and if MQTT connection is not available, return True value by conducting MQTT Subscript.

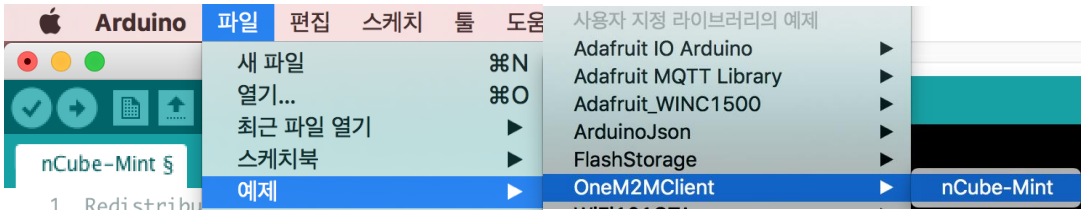
If WiFi is not connected, reconnect WiFi and return False value.

getAeid()

getAeid function returns stored AE-ID value from the resource structure.

3.3. nCube-Mint Setting

Setting and resource structure building for IoT Platform connection using nCube-Mint example will be done in this section.



Restart Arduiono IDE and open nCube-Mint from the tab File>Examples>OneM2MClient>n-Cube-Mint.

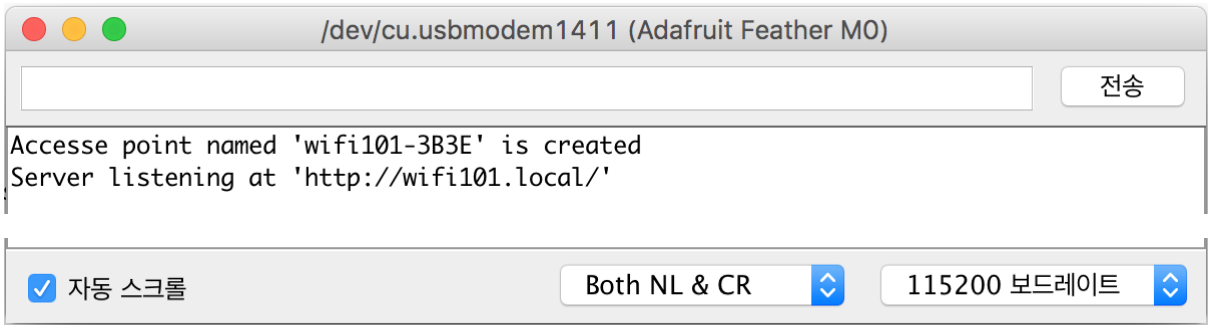
```
const String AE_ID = "your-ae";                  // guide: this is same with AE name
const String MQTT_BROKER_IP = "203.253.128.161";  // guide: set IP address of MQTT broker for Mobius as IoT Platform
const uint16_t MQTT_BROKER_PORT = 1883;           // guide: set MQTT port used
OneM2MClient nCube(MQTT_BROKER_IP,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

unsigned long req_previousMillis = 0;
const long req_interval = 2000;

// guide: set sensing period, modify or add for your sensors

unsigned long sensing_previousMillis = 0;
const long sensing_interval = (1000 * 5);
```

Change the variable values of *MQTT_BROKER_IP* and *MQTT_BROKER_PORT* into the values of *MQTT Broker IP* and *Port* number of Mobius server.

The variable sensing_interval determines the uploading interval of the values of Co2 into IoT Platform. Default setting is 5 seconds interval and a unit of milli-second.

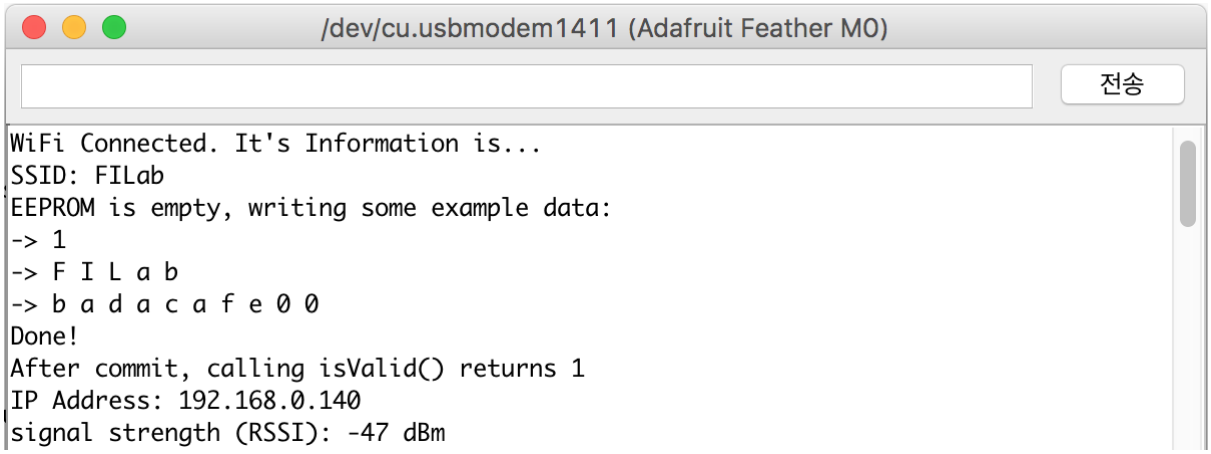
3.4. nCube-Mint Sketch Execution

Upload nCube-Mint Sketch into Adafruit Feather M0 after configuration and resource building. Click *Serial Monitor* button from the upper left of Arduino IDE.

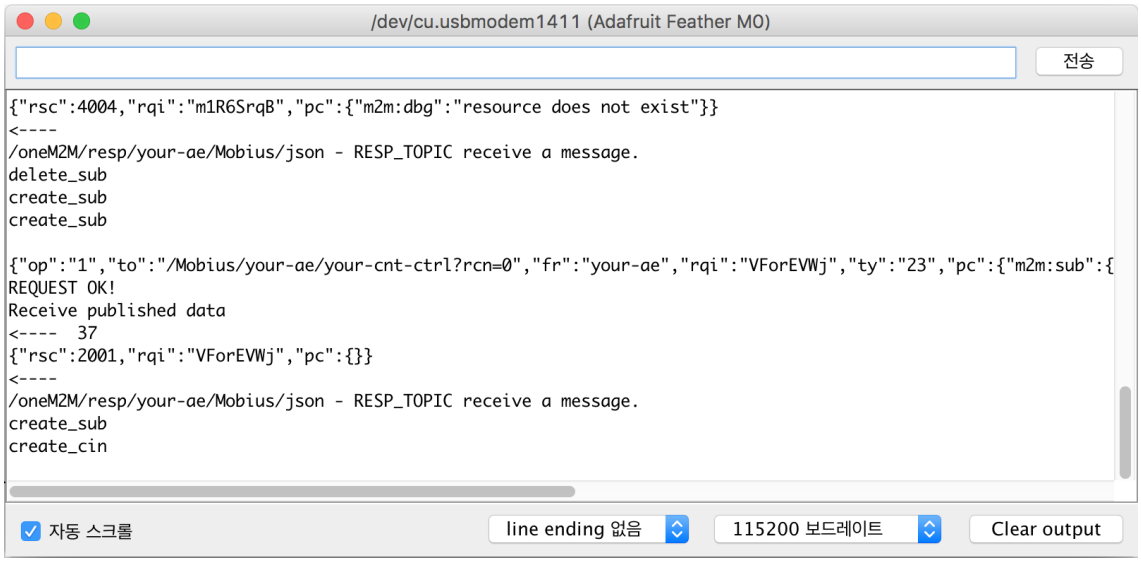


After Sketch compile and upload, if successfully conducted, serial monitor will show the status as above. The figure below shows the access status of AP 'wifi101-XXXX'(The name varies according to the board of Adafruit Feather M0) using WiFi from the smart phone.

Click *Refresh* button to find available WiFi AP lists. Input *SSID AP* under *Network Name* and *Password*. By clicking *Connect* button, *WiFi* connection is done.



After the WiFi AP access of nCube-Mint, Serial Monitor will present current AP information, local IP address and the signal strength of WiFi.



The operation of nCube-Mint including

mint AE creation → co2, led, relay Container creation → led, relay Subscript deletion → led, relay Subscrip creation

will be presented in Serial Monitor.

3.5. nCube-Mint Basic Resource Inquiry

The Resource Monitor for Mobius is available in versions of *C#, Windows OS* program and the *Web*. In this document, nCube-Mint resource inquiry using Resource Monitor *Web* version will be presented.

```
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";
  nCube.resource[index].rn = AE_ID;
  nCube.resource[index++].status = 0;

  // Container resource
  nCube.resource[index].ty = "3";
  nCube.resource[index].to = "/Mobius/" + 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/" + 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/" + nCube.resource[0].rn + '/' + nCube.resource[2].rn;
  nCube.resource[index].rn = "your-sub";
  nCube.resource[index++].status = 0;

  nCube.resource_count = index;
}
```

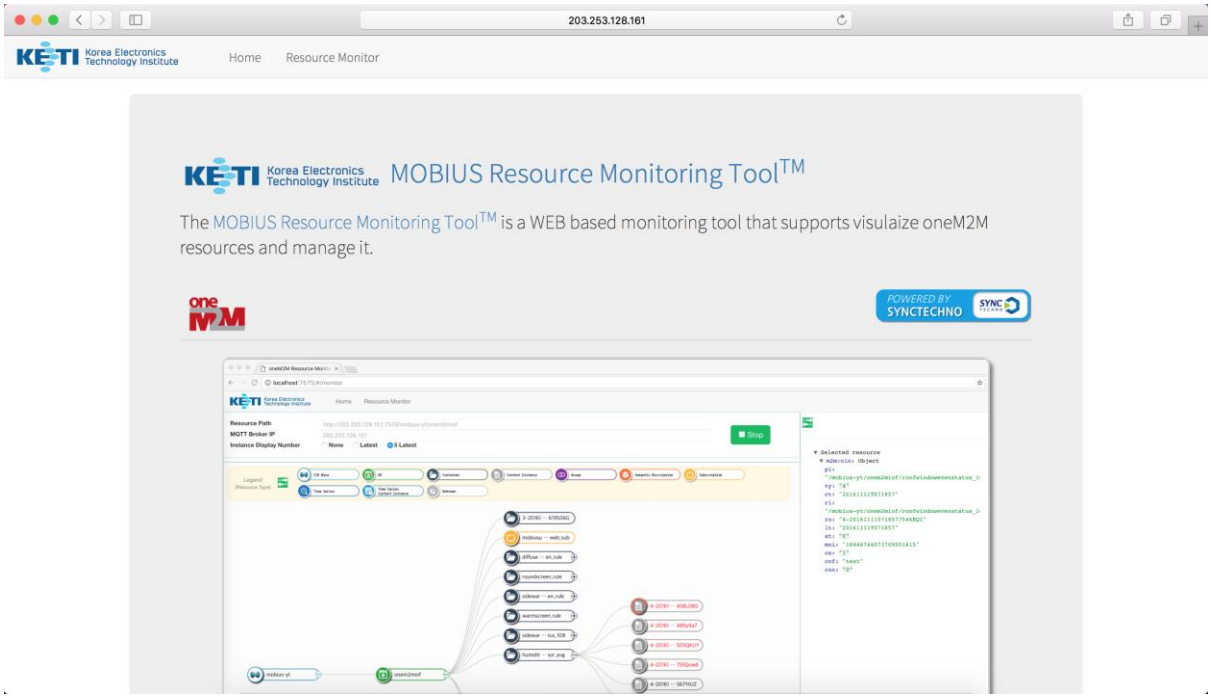
From the buildResource() function in nCube-Mint basic example Sketch, available resources to be created are shown.

The nCube-Mint basic example Sketch creates

your-ae under *mobius-yt cse*

your-cnt, *your-cnt-ctrl* under *mint-ae*

your-sub under *led cnt.*

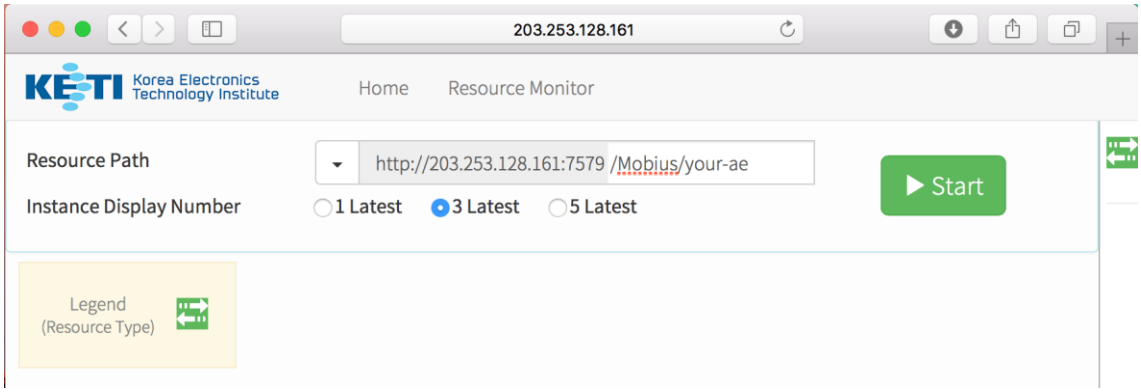


http://203.253.128.161:7575/

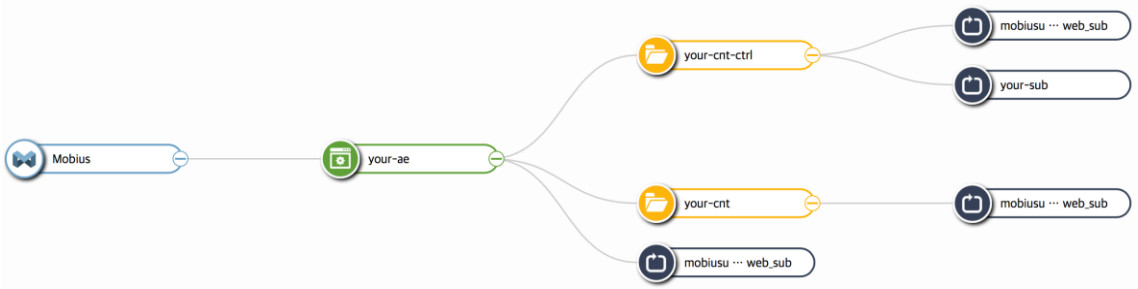
Access to Mobius Resource Monitor using the URL above.

Access to Resource Monitor page by click the Resource Monitor tap and use Resource Monitor function.

http://203.253.128.161:7579/mobius-yt/tour-ae



Put the above address into Resource Path and click Start button.



The resource tree that nCube-Mint created will be shown as above.

mint ae, co2, cnt, led cnt are also successfully created under Mobius-yt CSE.

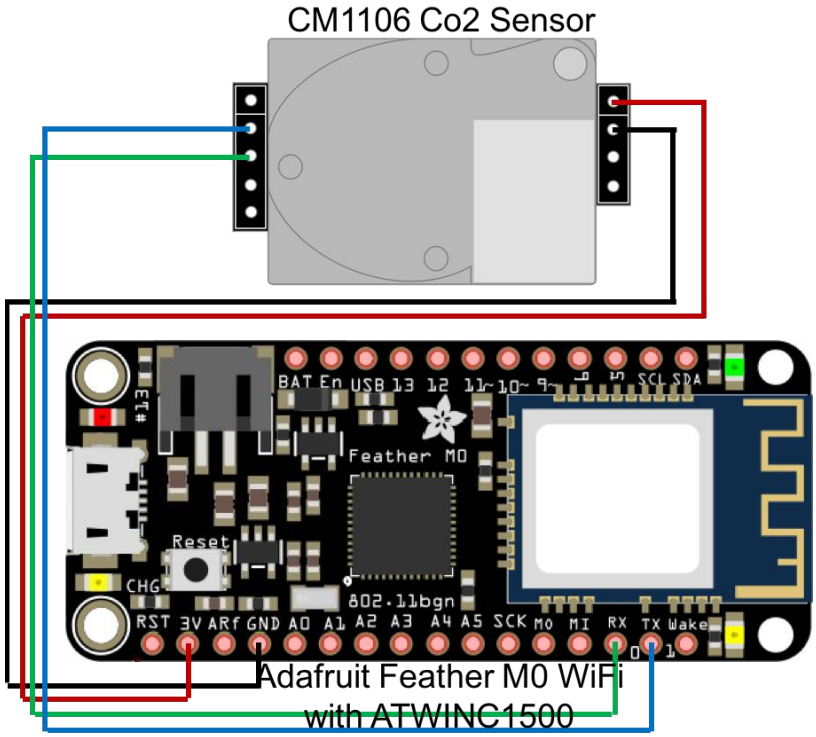
4. nCube-Mint Application Development

In this chapter, connect Co2 Sensor, RGB-LED, Power Relay to the nCube-Mint board, then upload Co2 concentration to IoT Platform, and control RGB-LED and Power Relay.

4.1. nCube-Mint + Co2 Sensor

First, Connect Co2 sensor to nCube-Mint board. Next, measure Co2 concentration and upload to IoT Platform.

4.1.1. Co2 Sensor Connection



Connect Co2 sensor to Adafruit Feather M0 as above picture.

4.1.2. TasCO2 Library Introduction

TAS (Thing Adeption Software) is a software for the communication between nCube-Mint and IoT sensor. In this document, TasCO2 is only for the TAS development of Co2 sensor. To use the library functions variables, first create TasCO2 object, call and use TasCO2 library function.

init()

This function is used to connect nCube-Mint and Co2 sensor. Before use Co2 sensor, this function must be called.

setCallback(measure_callback)

This function is used to set callback function be called after measure Co2 concentration.

chkCO2Data()

After this function called, Co2 concentration will passed to registered callback function's

parameter.

If the function above is called, Co2 concentration value will be transferred to the Callback function parameter.

reauestData()

This function is for the execution of the Co2 sensor measurement in the air.

4.1.3. nCube-Mint + Co2 Sensor Sketch

To make nCube-Mint get Co2 concentration measured by Co2 sensor, some code to be added to nCube-Mint example sketch.

```
const String AE_ID = "mint";                  // 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

OneM2MClient nCube(MQTT_BROKER_IP, MQTT_BROKER_PORT, AE_ID); // AE-ID

unsigned long req_previousMillis = 0;
const long req_interval = 2000;

// guide: set sensing period, modify or add for your sensors
unsigned long sensing_previousMillis = 0;
const long sensing_interval = (1000 * 5);
```

ae-id is modified to mint, and we can check Co2 sensing interval is set 5 seconds.

```
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";
    nCube.resource[index].rn = AE_ID;
    nCube.resource[index++].status = 0;

    // Container resource
    nCube.resource[index].ty = "3";
    nCube.resource[index].to = "/Mobius/" + nCube.resource[0].rn;
    nCube.resource[index].rn = "co2";
    nCube.resource[index++].status = 0;

    nCube.resource_count = index;
}
```

In this section, needs only a ae, cnt used to contain Co2 concentration. Modify buildResrouce() function's code like above picture. And set cnt name to co2.

```
21 #include "TasCO2.h"
31 TasCO2 TasCO2Sensor;
```

Now include TasCO2.h library, and create TasCO2 object named TasCO2Sensor.

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

In setup() function, initialize Co2 Sensor, set upload_callback() to callback function via calling setCallback() function.

```
288             if (state == "create_cin") {
289                 // guide: in here generate sensing data
290                 // if get sensing data directly, assign curValue sensing data and set sensing_flag to 1
291                 // if request sensing data to sensor, set sensing_flag to 0, in other code of receiving
292                 TasCO2Sensor.requestData();
293                 sensing_flag = 0;
294             }
295         }
```

In loop() function's if (state == "create_cin") conditional statement, call requestData() function to make Co2 sensor measuring Co2 concentration.

```
70 void upload_callback(String con) {
71     if (state == "create_cin") {
72         curValue = con;
73         sensing_flag = 1;
74     }
75 }
```

After measuring Co2 concentration, it will be passed to upload_callback() function's con parameter. curValue variable is made to contain data to upload, so copy con data to curValue.

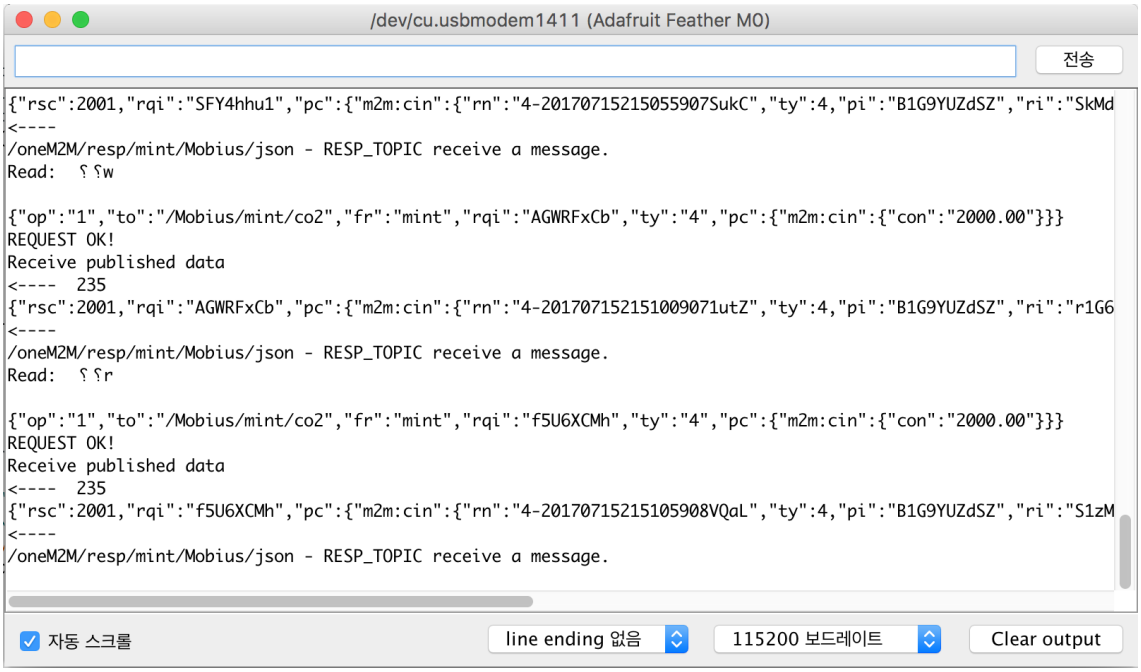
```
317             TasCO2Sensor.chkCO2Data();
```

In loop() function's if (nCube.chkConnect()) conditional statement, call chkCO2Data()

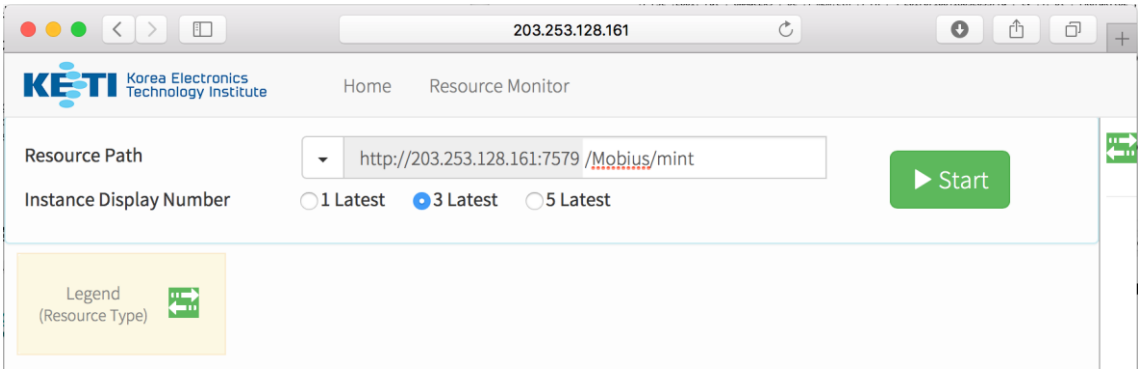
function.

4.1.4. nCube-Mint + Co2 Sensor Operation and Data Query

Upload sketch to nCube-Mint board, input WiFi SSID and Password information used by nCube-Mint board via smart phone.

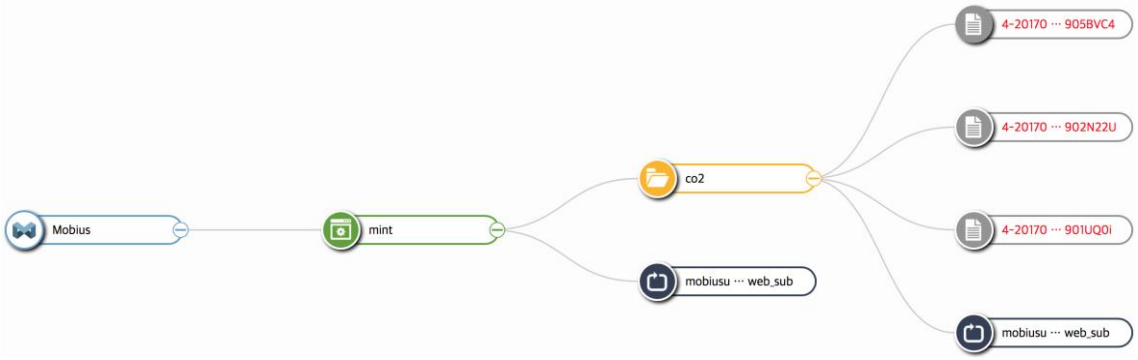


In serial monitor, Co2 concentration uploading status shown in 5 seconds terms.



Query Co2 data uploaded by nCube-Mint via Mobius Resource Monitor. Because nCube-

Mint ae-id is modified to mint, Resource path is http://203.153.128.161:7579/Mobius/mint.

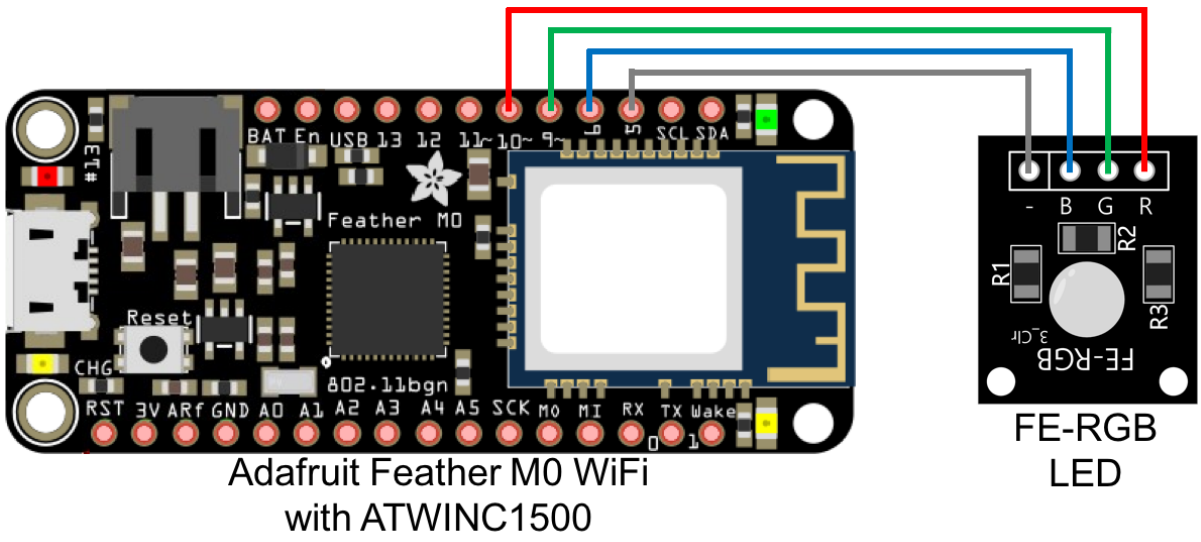


A new cin containing Co2 data is shown in 5 seconds terms.

4.2. nCube-Mint RGB-LED Connection

Connect RGB-LED to nCube-Mint board, control the LED via Mobius Resource Monitor.

4.2.1. nCube-Mint RGB-LED Connection



Connect RGB-LED to Adafruit Feather M0 like above picture.

4.2.2. nCube-Mint + LED Sketch

To control LED according to notification packet, some code to be added and modified like below. nCube-Mint + LED sketch is written based on nCube-Mint example sketch.

```
23 const String AE_ID = "mint";                  // guide: this is same with AE name
24 const String MQTT_BROKER_IP = "203.253.128.161";  // guide: set IP address of MQTT broker for Mobius as IoT Platform
25 const uint16_t MQTT_BROKER_PORT = 1883;          // guide: set MQTT port used
```

Modify AD_ID to mint.

```
103 void buildResource() {
104     // temperally build resource structure into Mobius as oneM2M IoT Platform
105
106     // AE resource
107     uint8_t index = 0;
108     nCube.resource[index].ty = "2";
109     nCube.resource[index].to = "/Mobius";
110     nCube.resource[index].rn = AE_ID;
111     nCube.resource[index++].status = 0;
112
113     // Container resource
114     nCube.resource[index].ty = "3";
115     nCube.resource[index].to = "/Mobius/" + nCube.resource[0].rn;
116     nCube.resource[index].rn = "led-ctrl";
117     nCube.resource[index++].status = 0;
118
119     // Subscription resource
120     nCube.resource[index].ty = "23";
121     nCube.resource[index].to = "/Mobius/" + nCube.resource[0].rn + '/' + nCube.resource[1].rn;
122     nCube.resource[index].rn = "led-sub";
123     nCube.resource[index++].status = 0;
124
125     nCube.resource_count = index;
126 }
```

In this section, needs ae, cnt, sub under cnt used to contain LED control data and notify the data to nCube-Mint concentration. Modify buildResource() function's code like above picture then set each cnt, sub names to co2.

```
90 void noti_callback(String topic, JsonObject &root) {
91     if (state == "create_cin") {
92         if (root["pc"]["sgn"]["sur"] == (nCube.resource[2].to + "/" + nCube.resource[2].rn)) { // guide: uri of subscription
93             String con = root["pc"]["sgn"]["nev"]["rep"]["m2m:cin"]["con"];
94             noti_con = con;
95
96             const char *rqi = root["rqi"];
97             resp_rqi = String(rqi);
98             control_flag = 1;
99         }
100     }
101 }
```

In noti_callback() function's if conditional statement, modify 'nCube.resource[3].to + "/" + nCube.resource[3].rn' to 'nCube.resource[2].to + "/" + nCube.resource[2].rn' to reference right sub resource.

```
21 #define LED_RED_PIN 10
22 #define LED_GREEN_PIN 9
23 #define LED_BLUE_PIN 6
24 #define LED_GND_PIN 5
```

Above codes define LED pins. It will make better readability.

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

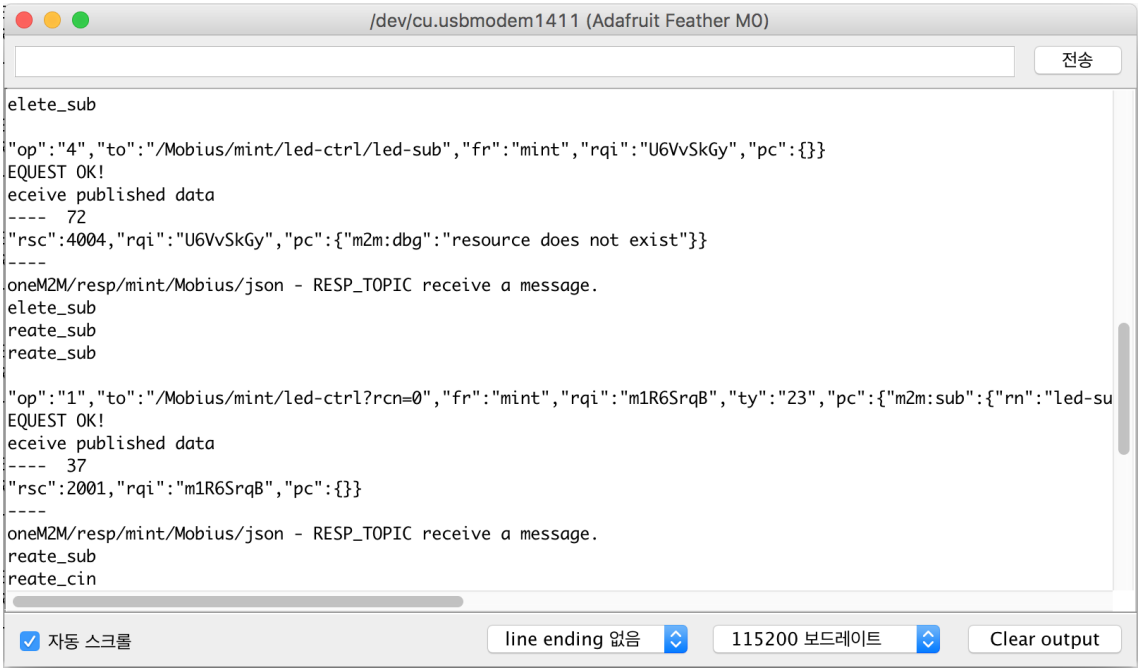
In setup() function, set LED pins to output digital signal 0 or 1.

```
320                     if (noti_con == "0") {
321                       digitalWrite(LED_RED_PIN, LOW);
322                       digitalWrite(LED_GREEN_PIN, LOW);
323                       digitalWrite(LED_BLUE_PIN, LOW);
324                     }
325                     else if (noti_con == "1") {
326                       digitalWrite(LED_RED_PIN, HIGH);
327                       digitalWrite(LED_GREEN_PIN, LOW);
328                       digitalWrite(LED_BLUE_PIN, LOW);
329                     }
330                     else if (noti_con == "2") {
331                       digitalWrite(LED_RED_PIN, LOW);
332                       digitalWrite(LED_GREEN_PIN, HIGH);
333                       digitalWrite(LED_BLUE_PIN, LOW);
334                     }
335                     else if (noti_con == "3") {
336                       digitalWrite(LED_RED_PIN, LOW);
337                       digitalWrite(LED_GREEN_PIN, LOW);
338                       digitalWrite(LED_BLUE_PIN, HIGH);
339                     }
340                     else if (noti_con == "4") {
341                       digitalWrite(LED_RED_PIN, HIGH);
342                       digitalWrite(LED_GREEN_PIN, HIGH);
343                       digitalWrite(LED_BLUE_PIN, LOW);
344                     }
345                     else if (noti_con == "5") {
346                       digitalWrite(LED_RED_PIN, HIGH);
347                       digitalWrite(LED_GREEN_PIN, LOW);
348                       digitalWrite(LED_BLUE_PIN, HIGH);
349                     }
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       }
```

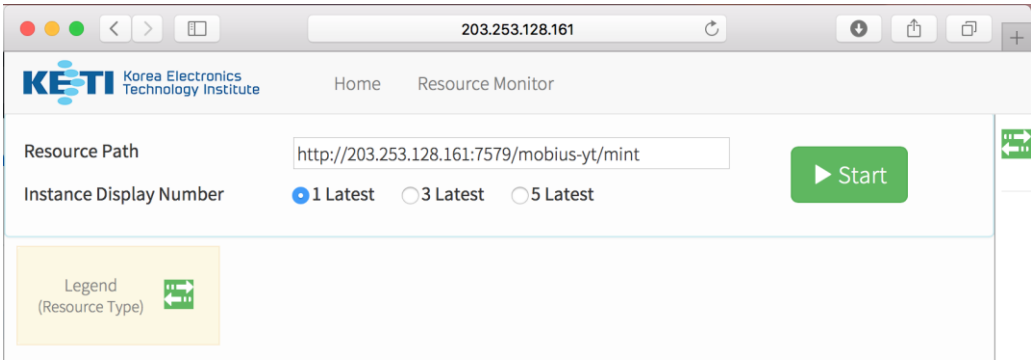
in loop() function's if (control_flag == 1) conditional statement, write code like above picture to turn on or off according to notification instance.

4.2.3. nCube-Mint LED Operation and Control

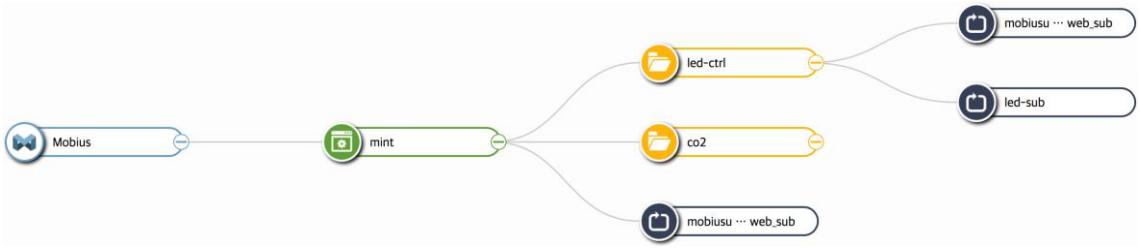
Upload sketch to nCube-Mint board, input WiFi SSID and Password information used by nCube-Mint board via smart phone.



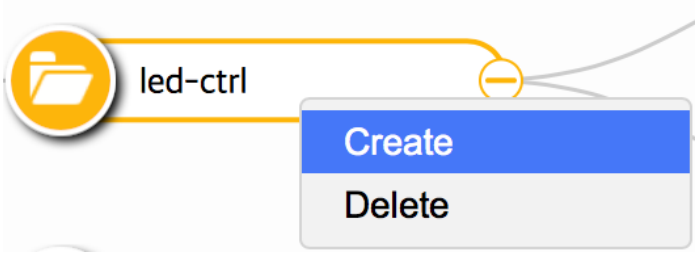
After checking serial monitor shows create_cin, go to next step.



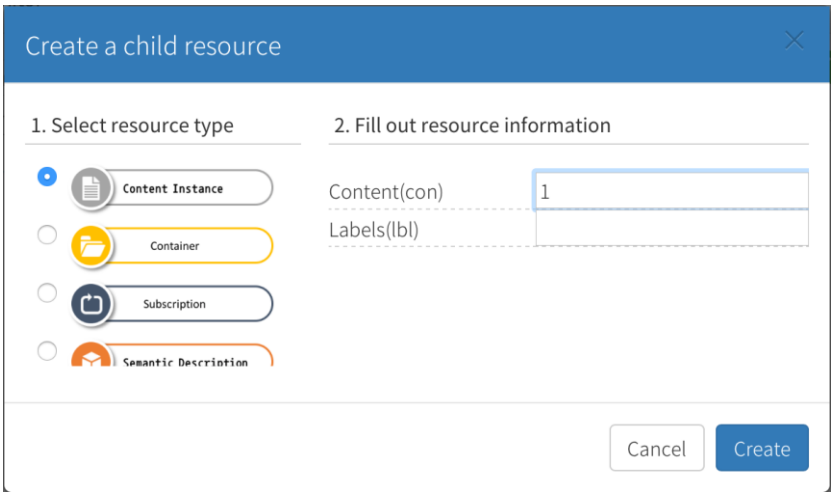
Before control nCube-Mint's RGB-LED, check resource structure made by nCube-Mint. Resource path is http://203.253.128.161:7579/Mobius/mint.



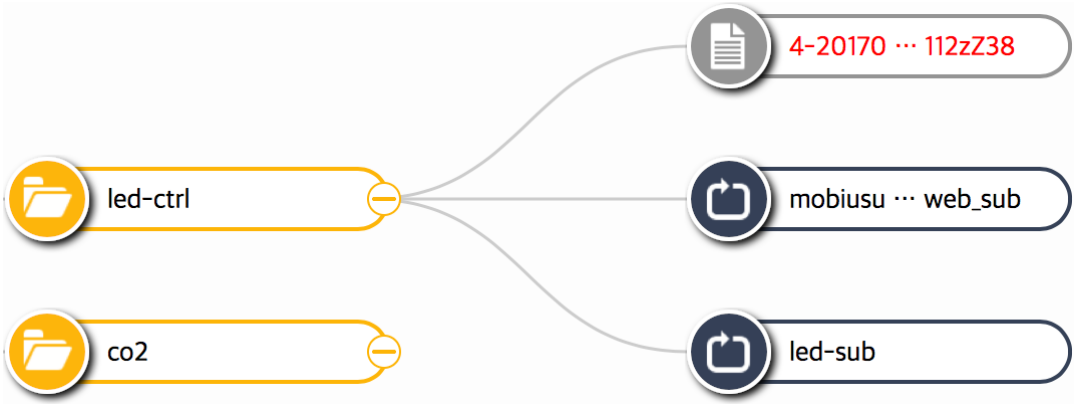
Resource structure will be shown like above picture.



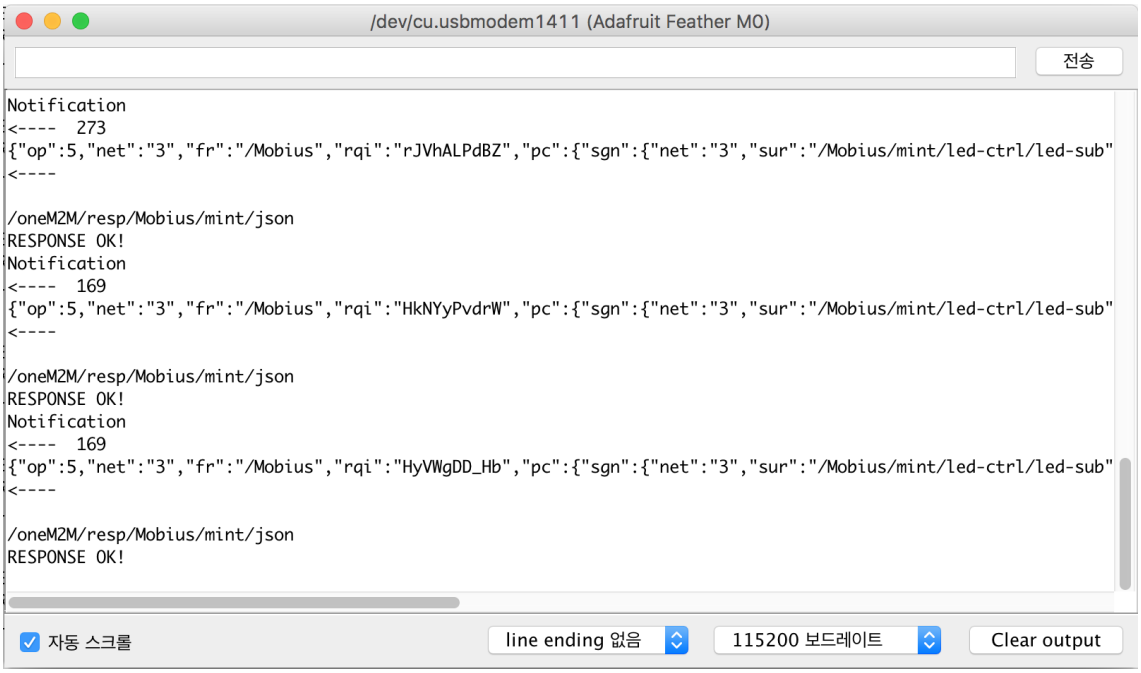
Move mouse pointer to the led-ctrl cnt, click right button and choose Create menu.



In Create a child resource window, first check Content Instance menu selected, fill Content(con) black with '1', and click Create button.



Then directly cin is created and shown under led-ctrl cnt.



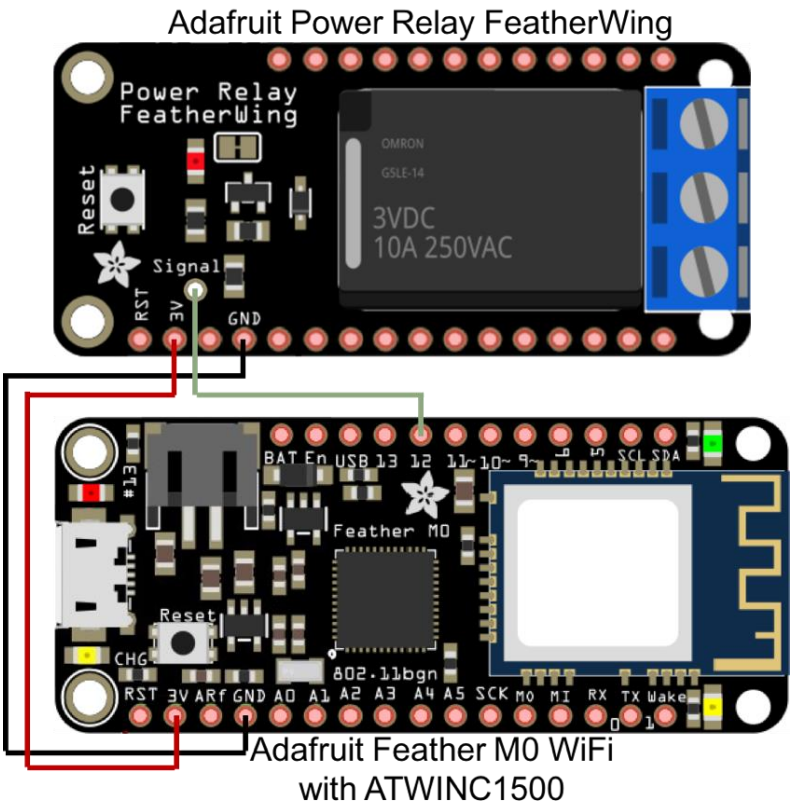
In serial monitor, can see receive notification packet and send response packet.

Depending on cin's con data, each or all of RGB LEDs will shine.

4.3. Adafruit Power Relay FeatherWing

Connect Adafruit Power Relay FeatherWing to nCube-Mint board, control the LED via Mobius Resource Monitor.

4.3.1. Adafruit Power Relay FeatherWing Connection



Connect Adafruit Power Relay FeatherWing to Adafruit Feather M0 like above picture.

4.3.2. nCube-Mint + Power Relay Sketch

For the Relay control from the Mobius Relay notification, the modification of the sketch should be conducted. The modification is based on nCube-Mint basic example sketch.

```
19 #define RELAYPIN 12
```

Define the signal pin connected to Adafruit Power Relay FeatherWing and the pin connected to Adafruit Feather M0.

```
269   pinMode(RELAYPIN, OUTPUT);
270
271   digitalWrite(RELAYPIN, LOW);
```

Inside the setup() function, the pin connected to Relay is set to print out digital signal (0 or 1).

```
320       if (noti_con == "0") {
321         digitalWrite(RELAYPIN, LOW);
322       }
323       else if (noti_con == "1") {
324         digitalWrite(RELAYPIN, HIGH);
325       }
```

The if statement (control_flag == 1) inside loop() function in nCube-Mint basic example, set the Relay signal value whether 0 or 1 according to the notification value from Mobius.

4.3.3. nCube-Mint Power Relay Operation and Control

Before the execution, rename ae-id→mint, cnt→relay-ctrl that contains the Relay control data, and sub→relay-sub that contains the Relay control data.

```
23 const String AE_ID = "mint";                  // guide: this is same with AE name
24 const String MQTT_BROKER_IP = "203.253.128.161";  // guide: set IP address of MQTT broker for Mobius as IoT Platform
25 const uint16_t MQTT_BROKER_PORT = 1883;          // guide: set MQTT port used
```

The figure above shows how to modify ae-id to mint.

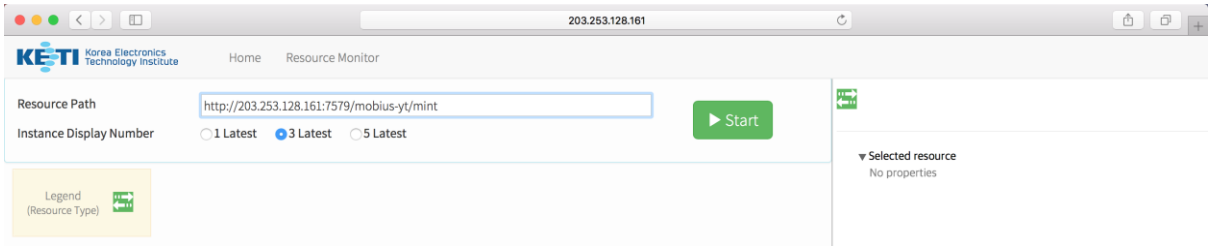
```
103 void buildResource() {
104     // temperally build resource structure into Mobius as oneM2M IoT Platform
105 
106     // AE resource
107     uint8_t index = 0;
108     nCube.resource[index].ty = "2";
109     nCube.resource[index].to = "/Mobius";
110     nCube.resource[index].rn = AE_ID;
111     nCube.resource[index++].status = 0;
112 
113     // Container resource
114     nCube.resource[index].ty = "3";
115     nCube.resource[index].to = "/Mobius/" + nCube.resource[0].rn;
116     nCube.resource[index].rn = "relay-ctrl";
117     nCube.resource[index++].status = 0;
118 
119     // Subscription resource
120     nCube.resource[index].ty = "23";
121     nCube.resource[index].to = "/Mobius/" + nCube.resource[0].rn + '/' + nCube.resource[1].rn;
122     nCube.resource[index].rn = "relay-sub";
123     nCube.resource[index++].status = 0;
124 
125     nCube.resource_count = index;
126 }
```

buildResource() function should be modified to execute nCube-Mint, which includes *ae*, *cnt* that contains Relay control data, and the *sub* that notifies Relay control data.

```
90 void noti_callback(String topic, JsonObject &root) {
91     if (state == "create_cin") {
92         if (root["pc"]["sgn"]["sur"] == (nCube.resource[2].to + "/" + nCube.resource[2].rn)) { // guide: uri of subscription
93             String con = root["pc"]["sgn"]["nev"]["rep"]["m2m:cin"]["con"];
94             noti_con = con;
95 
96             const char *rqi = root["rqi"];
97             resp_rqi = String(rqi);
98             control_flag = 1;
99         }
100     }
101 }
```

Modify if statement in noti_callback() function as above: nCube.resource[3] to nCube.resouce[2].

Upload Sketch and input SSID of WiFi AP and Password for the nCube-Mint execution with smartphone.

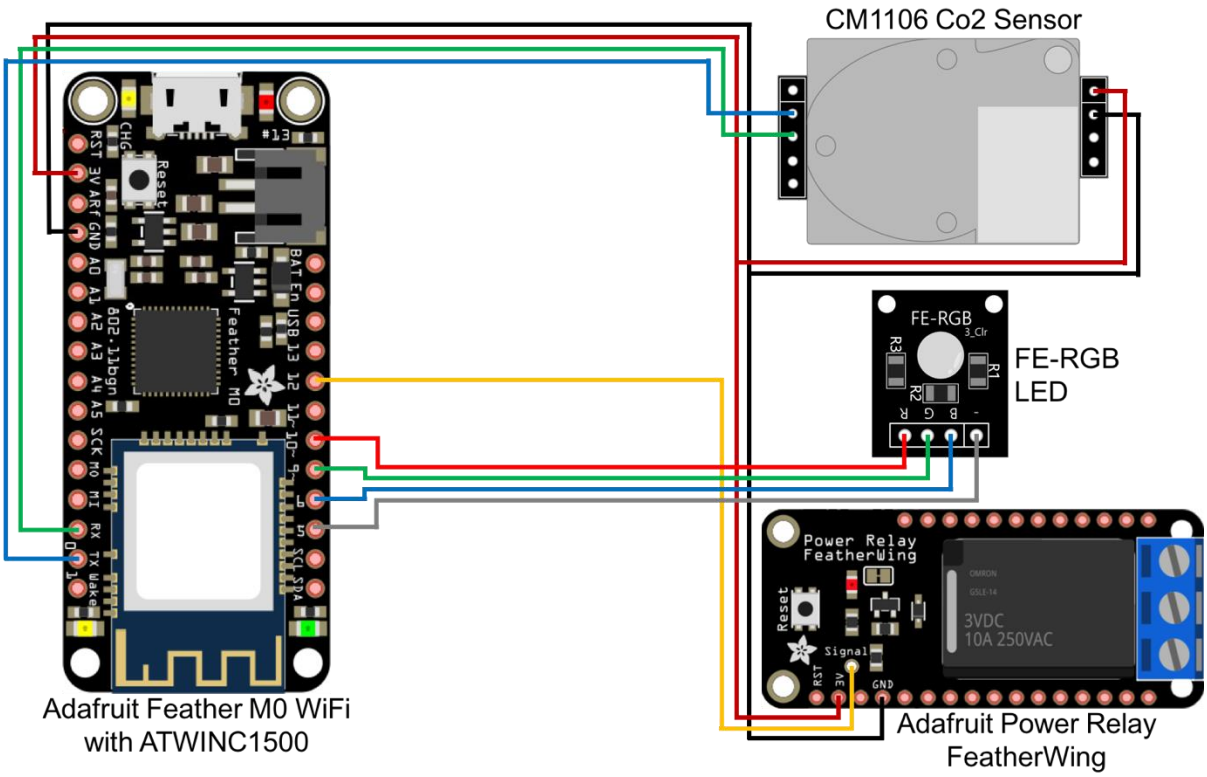


Before controlling Power Relay connected to nCube-Mint with Mobius Resource Monitor, query the resource structure created with nCube-Mint. The resource Path is identical with the previous example: http://203.253.128.161:7579/mobius-yt/mint

5. Sensor Integration, nCube-Mint Execution and Testing

In this chapter, the connection of Co2 sensor, RGB-LED, Power Relay with Adafruit Feather M0 and the querying of Co2 concentration and LED, Relay control will be explained.

5.1. nCube-Mint Circuit Composition



The figure above presents Adafruit Feather M0 connection to Co2 sensor, RGB-LED, Adafruit Power Relay FeatherWing.

5.2. nCube-Mint Sketch

```
#include "TasCO2.h"

#define LEDPIN 13

#define LED_RED_PIN 10
#define LED_GREEN_PIN 9
#define LED_BLUE_PIN 6
#define LED_GND_PIN 5

#define RELAY_PIN 12
```

```
const String AE_ID = "mint";                    // guide: this is same with AE name
const String MQTT_BROKER_IP = "203.253.128.161";  // guide: set IP address of MQTT
const uint16_t MQTT_BROKER_PORT = 1883;           // guide: set MQTT port used

OneM2MClient nCube(MQTT_BROKER_IP, MQTT_BROKER_PORT, AE_ID); // AE-ID

TasCO2 TasCO2Sensor;
```

```
void upload_callback(String con) {
    if (state == "create_cin") {
        curValue = con;
        sensing_flag = 1;
    }
}
```

```
void noti_callback(String topic, JsonObject &root) {
    if (state == "create_cin") {
        if (root["pc"]["sgn"]["sur"] == (nCube.resource[4].to + "/" + nCube.resource[4].rn)) { // guide: uri of subscription
            String con = root["pc"]["sgn"]["nev"]["rep"]["m2m:cin"]["con"];
            noti_con = con;

            const char *rqi = root["rqi"];
            resp_rqi = String(rqi);
            control_flag = 1;
        }
        if (root["pc"]["sgn"]["sur"] == (nCube.resource[5].to + "/" + nCube.resource[5].rn)) { // guide: uri of subscription
            String con = root["pc"]["sgn"]["nev"]["rep"]["m2m:cin"]["con"];
            noti_con = con;

            const char *rqi = root["rqi"];
            resp_rqi = String(rqi);
            control_flag = 2;
        }
    }
}
```

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

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

320     TasCO2Sensor.init();
321     TasCO2Sensor.setCallback(upload_callback);
```

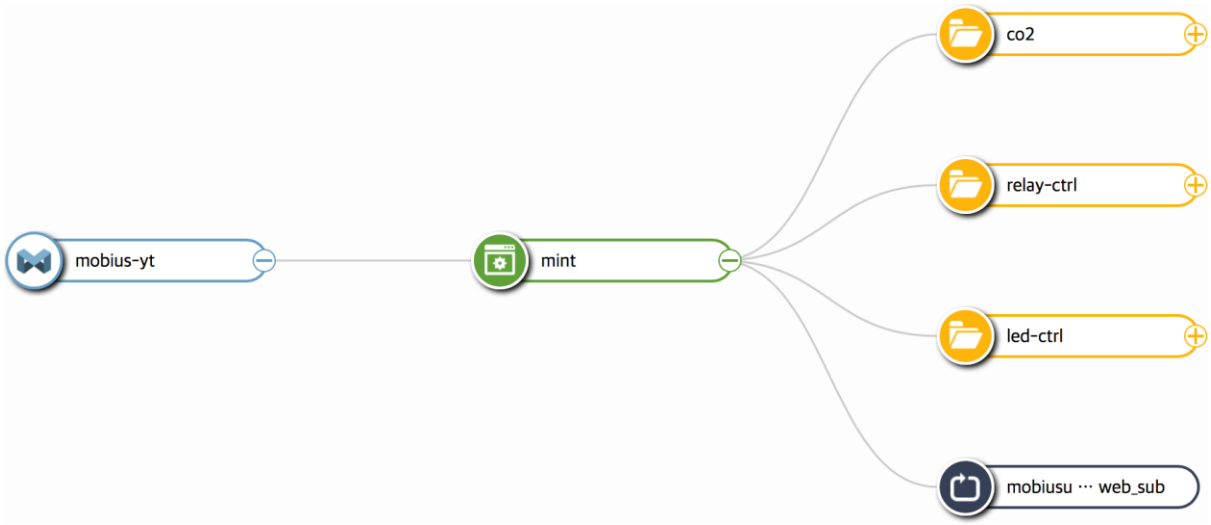
```
        if (state == "create_cin") {
            // guide: in here generate sensing data
            // if get sensing data directly, assign curValue sensing data and set sensing_flag to 1
            // if request sensing data to sensor, set sensing_flag to 0, in other code of receiving s
     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

        if (control_flag == 1) {
            control_flag = 0;

            if (noti_con == "0") {
              digitalWrite(LED_RED_PIN, LOW);
              digitalWrite(LED_GREEN_PIN, LOW);
              digitalWrite(LED_BLUE_PIN, LOW);
            }
            else if (noti_con == "1") {
              digitalWrite(LED_RED_PIN, HIGH);
              digitalWrite(LED_GREEN_PIN, LOW);
              digitalWrite(LED_BLUE_PIN, LOW);
            }
            else if (noti_con == "2") {
              digitalWrite(LED_RED_PIN, LOW);
              digitalWrite(LED_GREEN_PIN, HIGH);
              digitalWrite(LED_BLUE_PIN, LOW);
            }
            else if (noti_con == "3") {
              digitalWrite(LED_RED_PIN, LOW);
              digitalWrite(LED_GREEN_PIN, LOW);
              digitalWrite(LED_BLUE_PIN, HIGH);
            }
            else if (noti_con == "4") {
              digitalWrite(LED_RED_PIN, HIGH);
              digitalWrite(LED_GREEN_PIN, HIGH);
              digitalWrite(LED_BLUE_PIN, LOW);
            }
            else if (noti_con == "5") {
              digitalWrite(LED_RED_PIN, HIGH);
              digitalWrite(LED_GREEN_PIN, LOW);
              digitalWrite(LED_BLUE_PIN, HIGH);
            }

            else if (noti_con == "6") {
              digitalWrite(LED_RED_PIN, LOW);
              digitalWrite(LED_GREEN_PIN, HIGH);
              digitalWrite(LED_BLUE_PIN, HIGH);
            }
            else if (noti_con == "7") {
              digitalWrite(LED_RED_PIN, HIGH);
              digitalWrite(LED_GREEN_PIN, HIGH);
              digitalWrite(LED_BLUE_PIN, HIGH);
            }

            String resp_body = "";
            resp_body += "{\"rsc\":\"2000\",\"to\":\"\",\"fr\":\"" + nCube.getAeid() + "\",\"pc\":\"\",\"rqi\":\"" + resp_rqi + "\"}";
            resp_body.toCharArray(body_buff, resp_body.length() + 1);
            nCube.response(body_buff);
        }

        else if (control_flag == 2) {
            control_flag = 0;

            if (noti_con == "0") {
              digitalWrite(RELAY_PIN, LOW);
            }
            else if (noti_con == "1") {
              digitalWrite(RELAY_PIN, HIGH);
            }

            String resp_body = "";
            resp_body += "{\"rsc\":\"2000\",\"to\":\"\",\"fr\":\"" + nCube.getAeid() + "\",\"pc\":\"\",\"rqi\":\"" + resp_rqi + "\"}";
            resp_body.toCharArray(body_buff, resp_body.length() + 1);
            nCube.response(body_buff);
        }
```

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

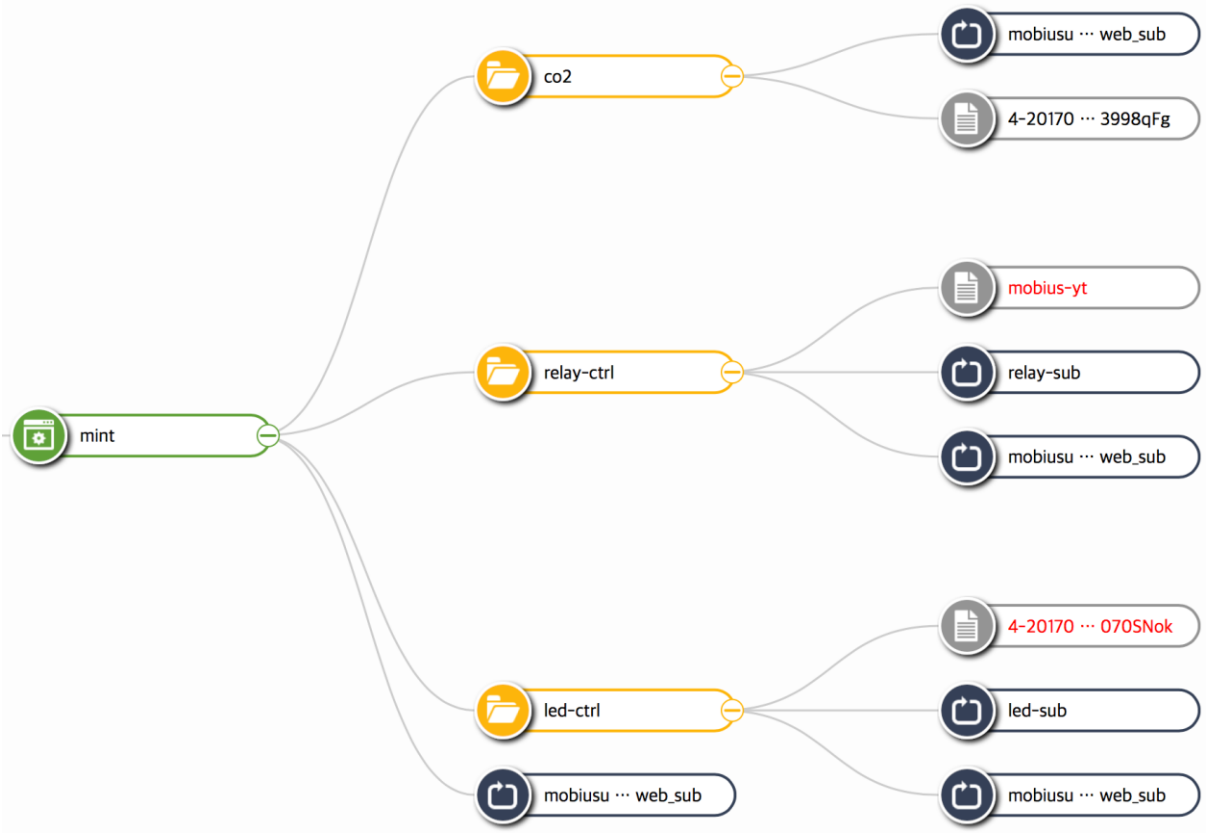
5.3. nCube-Mint Operation, Data Query and Control

Upload sketch and input WiFi AP SSID and password for nCube-Mint board using mobile phone.

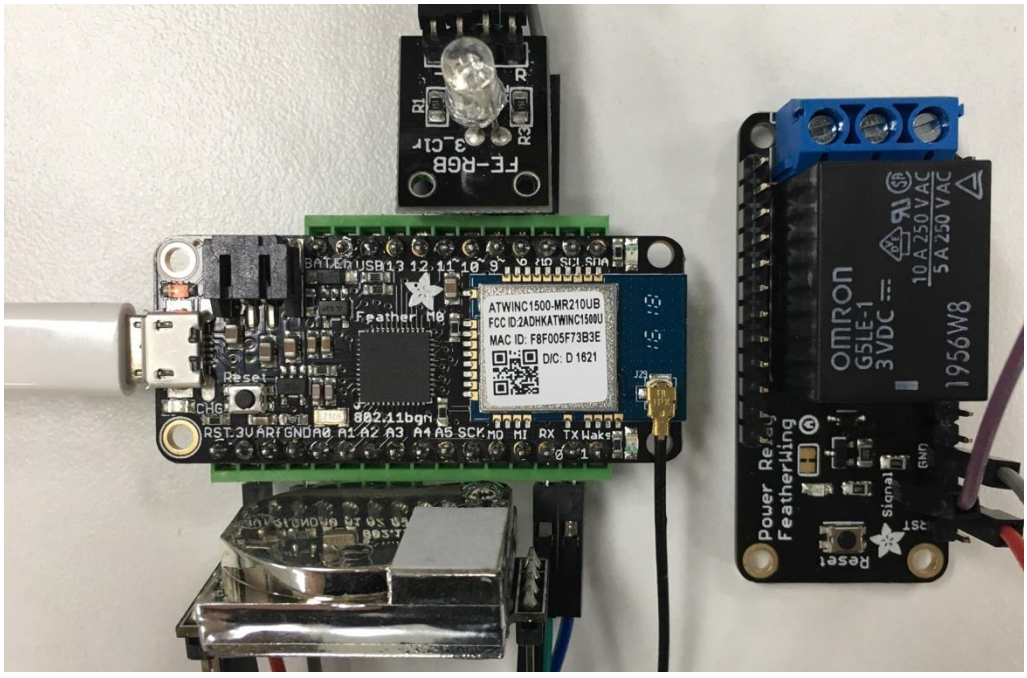
Co2 concentration values are uploaded in 5 seconds interval as above from the serial monitor.



The final resource structure of nCube-Mint from the Mobius Resource Monitor web version is as above.



Co2 concentration values are uploaded in *co2 cnt.* Control of RGB-LED and Power Relay is also available by creating *cin* under *led-ctrl, relay-ctrl cnt.*



Appendix A

Essential modifications for nCube-Mint development after WiFi101, PubSubClient library update

After conducting WiFi101 library update,

The code for *WiFi.cpp* modification is as below.

Arduino>Libraries>WiFi101>src

```
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   }
```

The code for *WiFi101.h* modification is as below.

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

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

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

After *PubSubClient* library update, from the root Arduino>Libraries>PubSubClient>src

The code for *PubSubClient.h* modification is as below.

```
26  #define MQTT_MAX_PACKET_SIZE 128
```

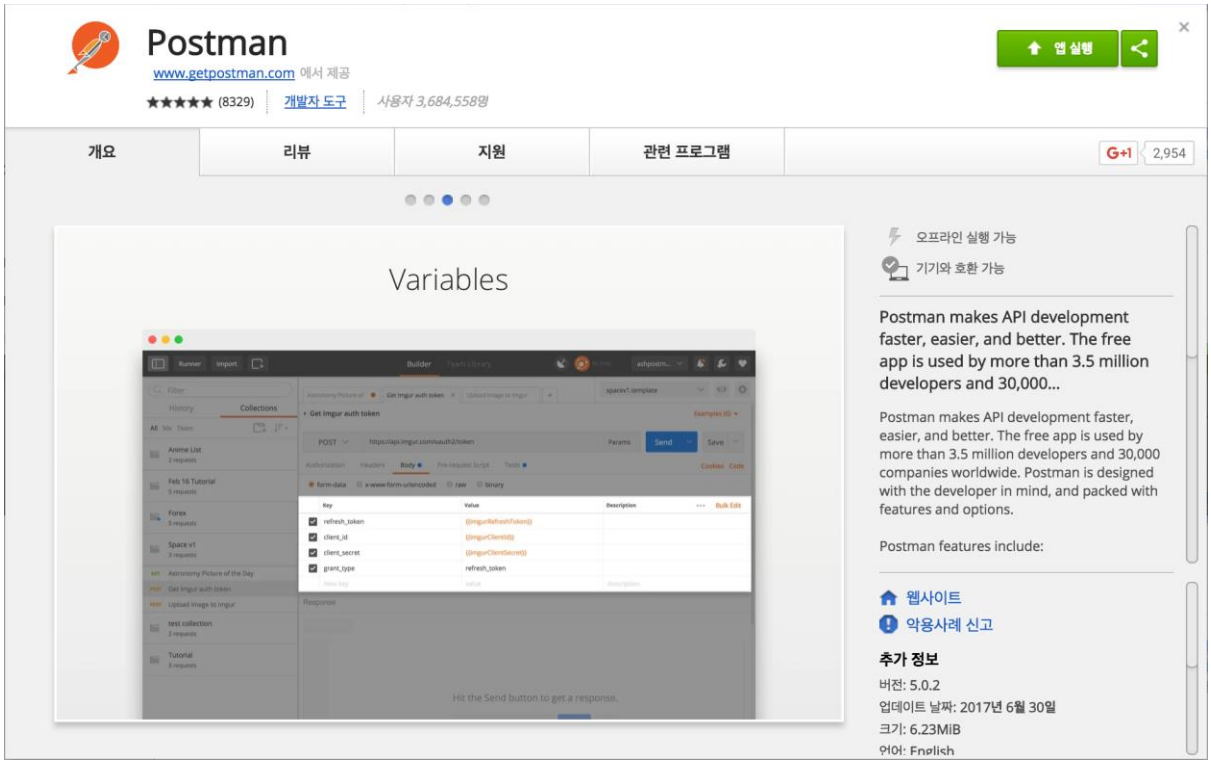
```
26  #define MQTT_MAX_PACKET_SIZE 400
```

Appendix B

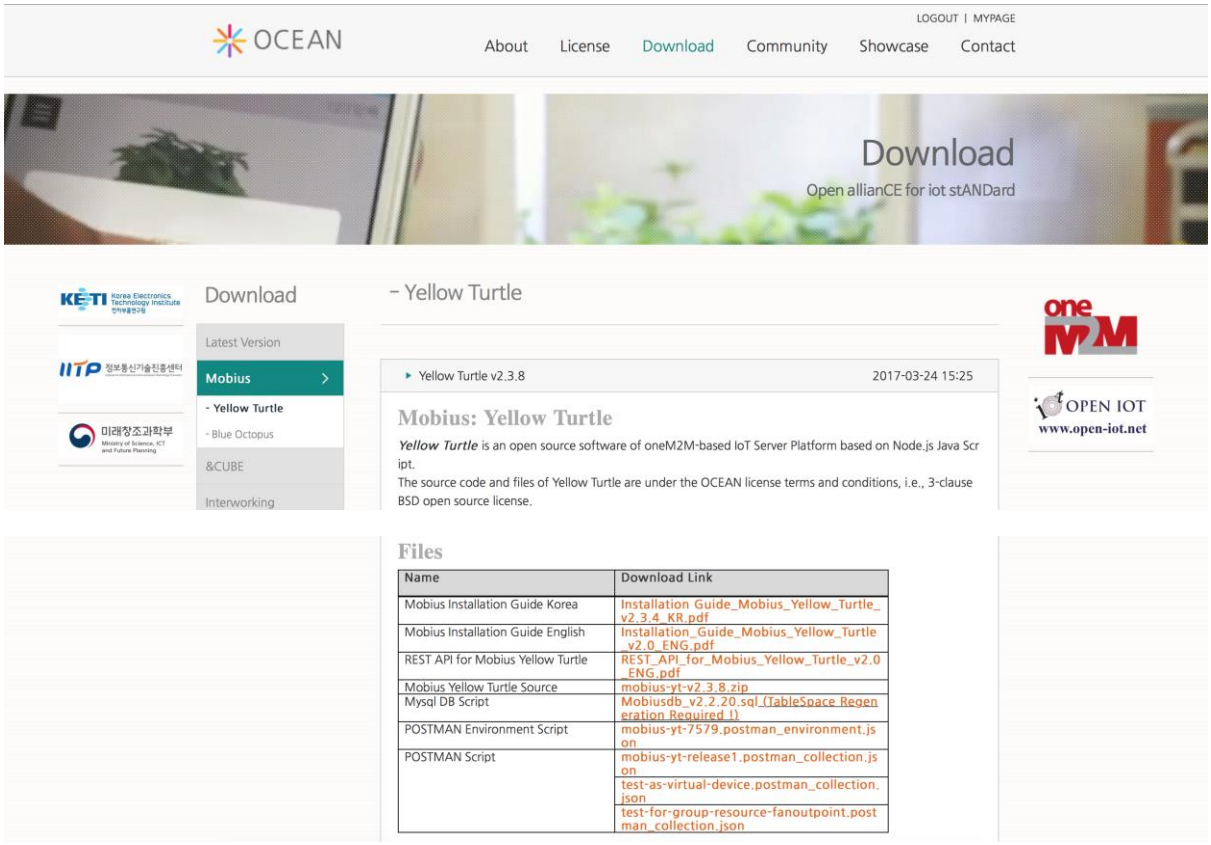
nCube-Mint Resource Query and Control using Postman

Browsing Co2 data updated with nCube-Mint, LED and controlling Relay is available with Postman Application in Google Chrome. Moreover, AE, CNT, CIN creation and querying with API is also available.

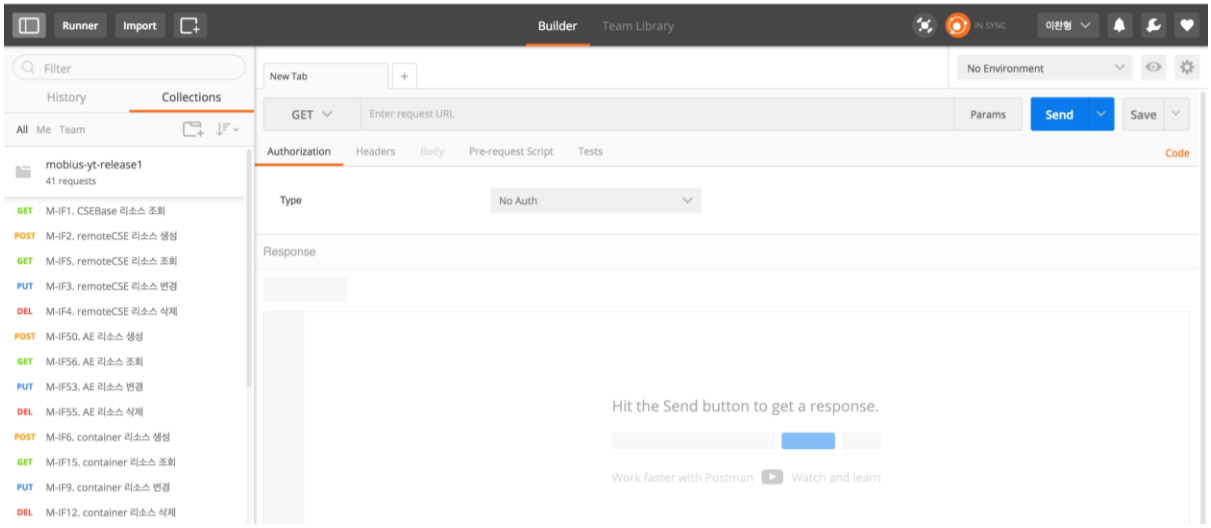
https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?utm_source=chrome-ntp-icon



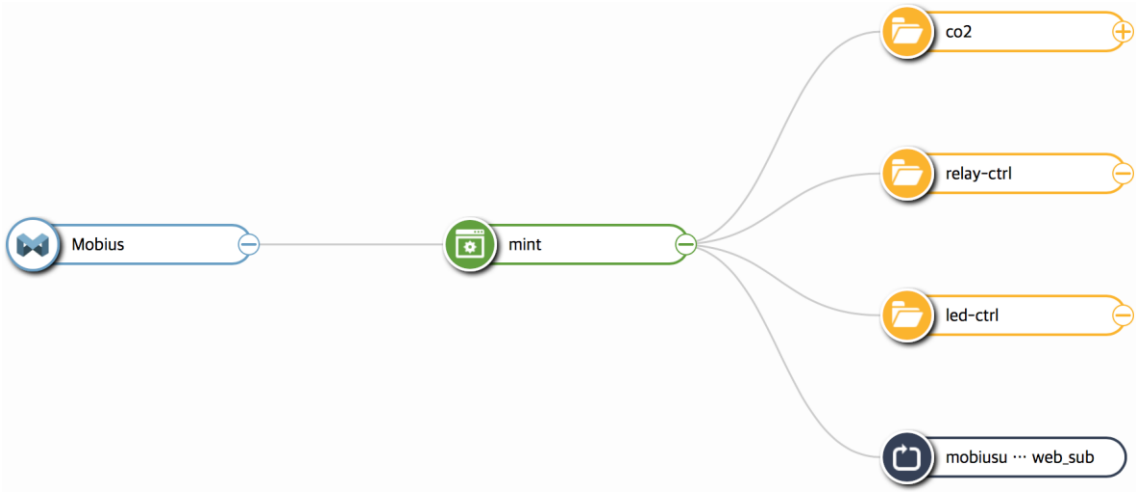
Download Postman Application in Google Chrome from the URL link above, and execute.



Go to the ocean official website (http://www.iotocean.org), Download>Mobius Yellow Turtle>POSTMAN Environment Scrip and click mobius-yt-7579.postman_environment.json and download Postman API collection.



Open API collection in Postman by import → import file → choose files, then APIs are added under Collections list on the left.


co2
relay-ctrl
Mobius
mint
led-ctrl
mobiusu ··· web_sub