



Software Version: 1.0.0

-Base on &Cube-Mint 1.2.1

Developer Guide v1.0

Document Release Date: July 7, 2017

Software Release Date: July 7, 2017



Copyright (c) 2014, OCEAN
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

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

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

Contents

1. Introduction.....	4
1.1. Arduino OTA(Over The Air).....	4
1.2. &Cube-Mint OTA	4
1.3. &Cube-Mint OTA Working flow.....	5
2. Develop Environment.....	6
2.1. Download and Configuration	6
2.2. Change the code	8
3. Execution.....	9
3.1. Upload first source	9
3.2. Compile new version software	10
3.3. Upload the HEX to OTA Server	11
3.4. Upgrade.....	12
4. Notice.....	13

1.Introduction

1.1. Arduino OTA(Over The Air)

OTA (Over The Air) means the programmer can code on local and send it to device to make it upgrade or replace with new firmware through the cloud. In Arduino, some developer finds a new way to update the Arduino sketch online. It makes developer who install the wireless Arduino device in a place where is difficult to touch can be update without uninstall.

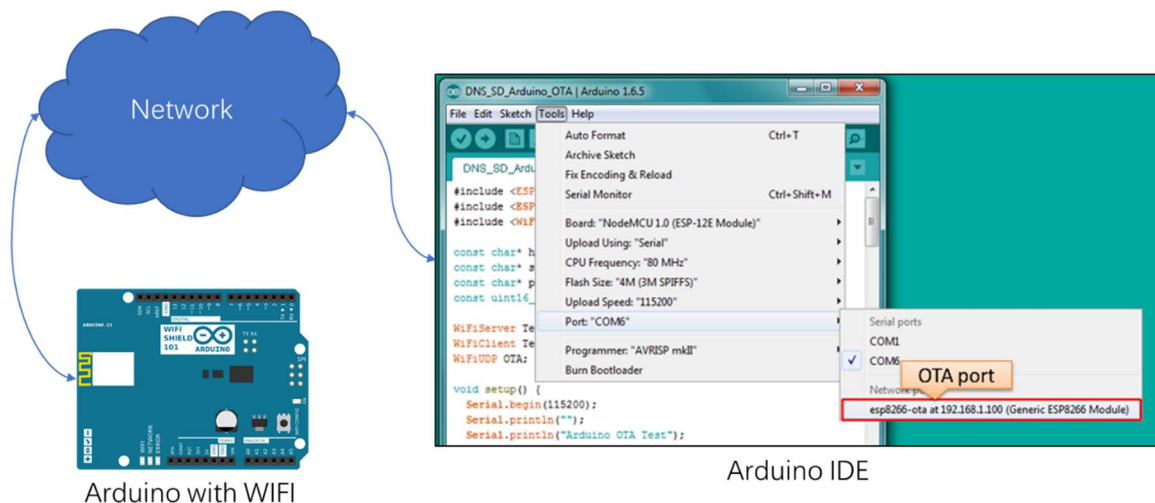


Figure 1 Arduino OTA

1.2. &Cube-Mint OTA

In some case user have a kind of light weight WIFI device. Just like Arduino Yun or Arduino UNO with WIFI shield. And they can access internet directly without any gateway's help. That is reason why we make &Cube-Mint software. It is a light weight C++ program for make a Adafruit Feather M0 WIFI board (A kind of Arduino M0 board) connect with Mobius-yt IoT Server Platform (More info in the [IoT-Ocean Site](#)). As mentioned above, the

user maybe need to upgrade the program without touching. So, we support another software witch named &Cube-Mint OTA to help them upgrading the software throw the internet.

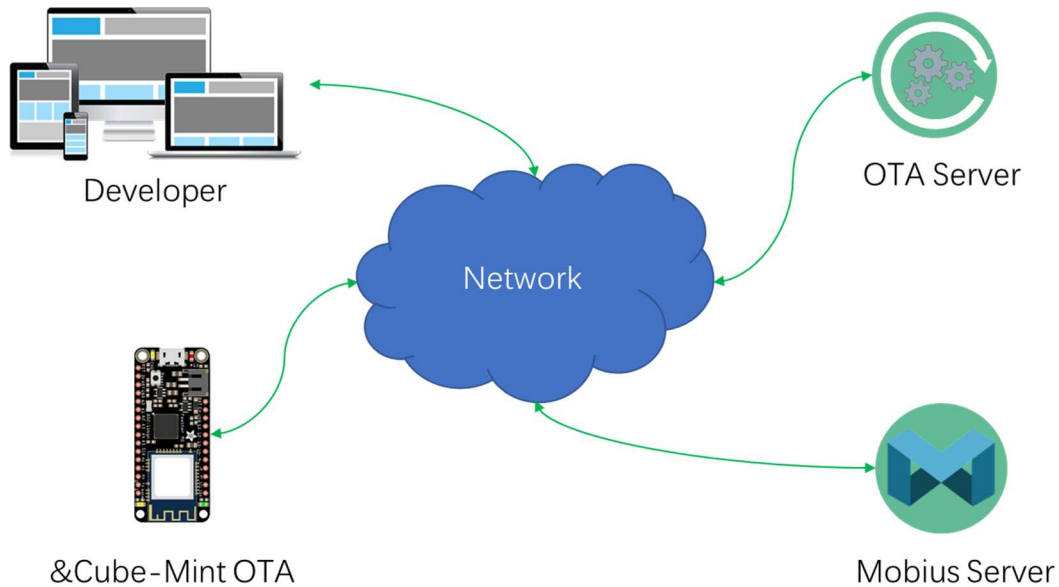


Figure 2 &Cube-Mint OTA System

1.3. &Cube-Mint OTA Working flow

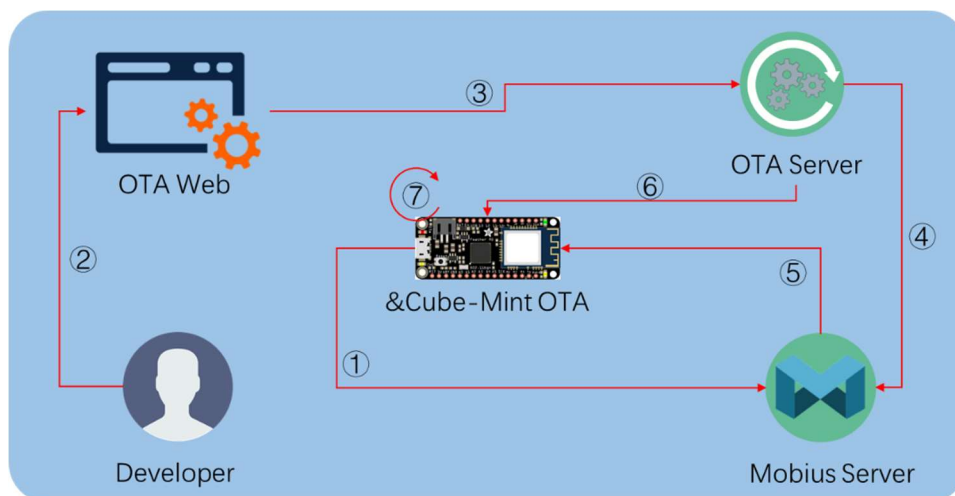


Figure 3 &Cube-Mint OTA Working flow

Figure 3 shows the &Cube-Mint OTA Working flow.

1. &Cube-Mint OTA start running. It will access the Mobius Server and create AE, Container and Subscription (oneM2M Resource).
2. Developer open the OTA web page and upload a new version software.
3. Send new version software to OTA Server and create or update the update profile.
4. OTA Server send update command to Mobius Server.
5. Mobius Server send update command to &Cube-Mint OTA as a notification message.
6. &Cube-Mint OTA receive the update command then request a new version software from OTA Server.
7. &Cube-Mint OTA receive the new software then write it to flash rom and reboot.

2. Develop Environment

Our &Cube-Mint OTA example source just works on the [Adafruit Feather M0 WIFI](#) board. It not native Arduino board but it is made base on Arduino M0 open hardware. So normal Arduino IDE can be used to build the source project and compiling with a little bit configuration.

2.1. Download and Configuration

1. Download Arduino IDE from [Arduino official site](#).
2. Configure the Arduino IDE follow the [Adafruit Feather M0 WIFI guide page](#).
3. Download the &Cube-Mint source from [IoTKETI github site](#).
4. Unzip nCube-Mint-OTA.zip file and cope all folder under libraries folder to the Arduino IDE libraries folder (C:\Users\[user]\Documents\Arduino\libraries in window PC).

5. Connect Adafruit Feather M0 WIFI board to PC with USB.
6. Open the Arduino IDE. Click menu "File>Examples>oneM2MClient>nCube-Mint-OTA".

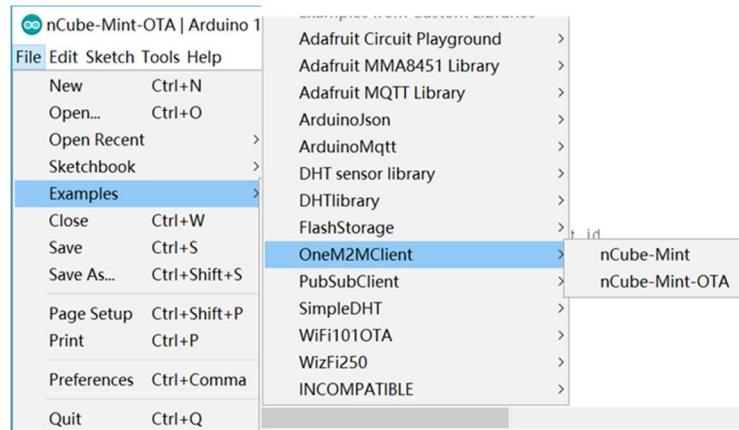


Figure 4 Open sample source

7. Click menu "Tools" to select "Board" and "Port".

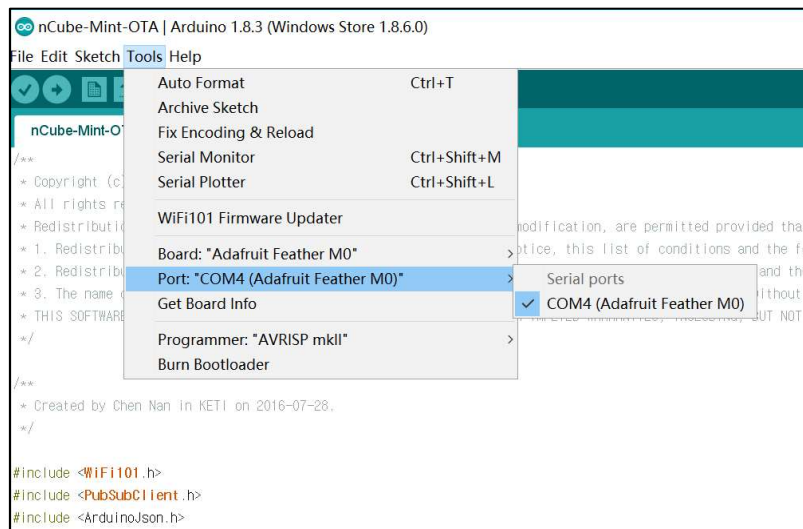


Figure 5 Configure the debug information

2.2. Change the code

The &Cube-Mint OTA sample source is an example for other developer understanding how to use OTA library in &Cube-Mint software. So, if you want to make it work as you wish you should check some parameter in the source before compiling.

1. "FIRMWARE_VERSION": this source version.
2. "AE_ID": this device AE (oneM2M Resource Type) name.

```
#include <WiFi101.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

#include "OneM2MClient.h"
#include "m0_ota.h"

#define LEDPIN 13

const String FIRMWARE_VERSION = "1.0.0.0"; // guide: this is your source version which it will compare with cloud
const String AE_ID = "your-ae"; // guide: this is same with AE name
const String MQTT_BROKER_IP = "203.259.126.151"; // 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;

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

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

////////////////////////////////////

short action_flag = 0;
short sensing_flag = 0;
short control_flag = 0;
```

3. "buildResource()": check the container name and subscription name, make sure OTA container("update") and subscription are existed.

```
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 = "update"; // fix container name for OTA
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-sensor-container";
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-control-container";
nCube.resource[index++].status = 0;

// Subscription resource
nCube.resource[index].ty = "23";
nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn + '/' + nCube.resource[1].rn;
nCube.resource[index].rn = "sub"; // fix subscription name for OTA
nCube.resource[index++].status = 0;

nCube.resource[index].ty = "23";
nCube.resource[index].to = "/mobius-yt/" + nCube.resource[0].rn + '/' + nCube.resource[3].rn;
nCube.resource[index].rn = "your-control-subscription";
nCube.resource[index++].status = 0;
```


3. Execution

3.1. Upload first source

Before OTA working, we need make first version of &Cube-Mint OTA running on the board. In order to facilitate observation, we connect a RGB LED light with this board and turn on the blue light when software begins running. And then click the "upload" button on top of Arduino IDE.

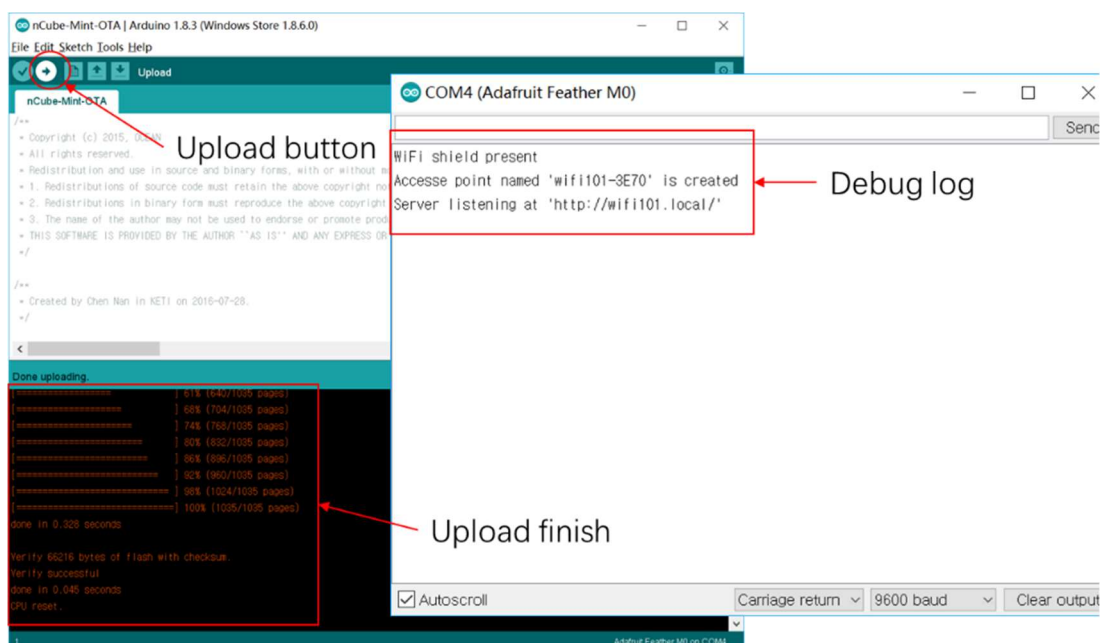


Figure 6 Upload first software

When uploading is finished, Using the smart phone or PC's WIFI to connect SSID "wifi101-****" that you can find in the debug log. Then open a browser in your device and move to "<http://wifi101.local/>" page.

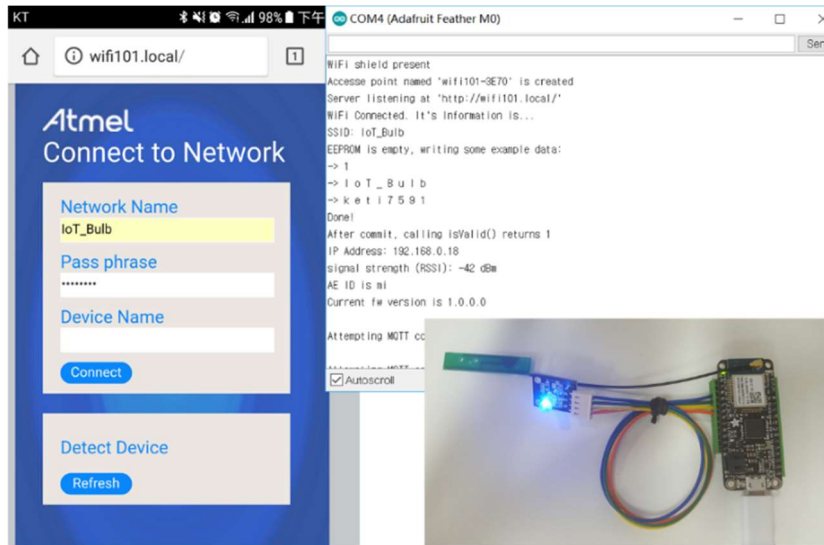


Figure 7 First software running

3.2. Compile new version software

Then we raise the version number to "1.0.0.1" in the source and turn on the red light when software begins running. Then click the "verify" button to compile it without upload.

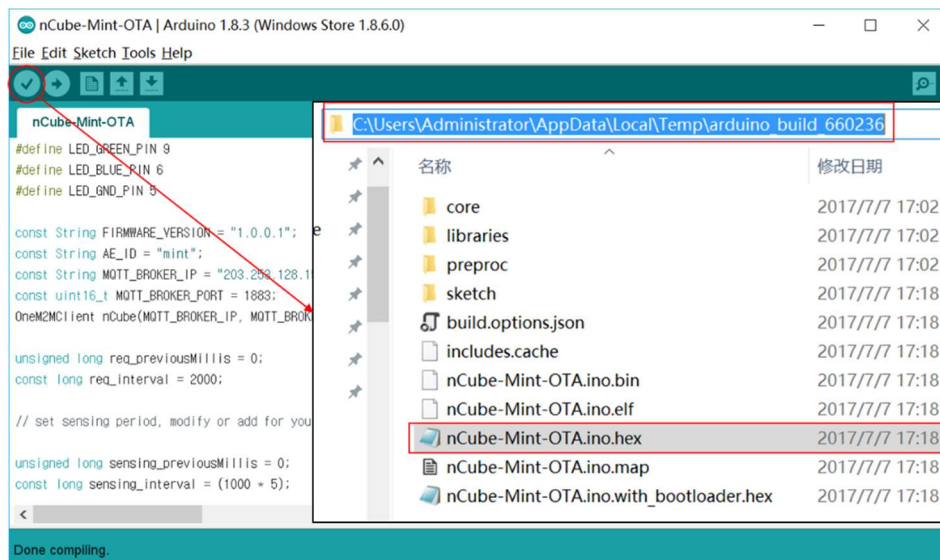


Figure 8 Compile new version software

When compiling is done you can find HEX file in below folder:

- "C:\Users\W[user]\AppData\Local\Temp\arduino_build_*****"

3.3. Upload the HEX to OTA Server

Now we have a new version "1.0.0.1" HEX file. Then we need upload this HEX file to OTA server and we have provided a web site for this. Open a browser application and type the URL "<http://203.253.128.161:8730>" you will see a page like Figure 9. Input the AE name (must same to your source), software version (must same to your source). Then load HEX file (must be *.hex file) and Pass Code (must remember it for uploading new version software of same AE device).

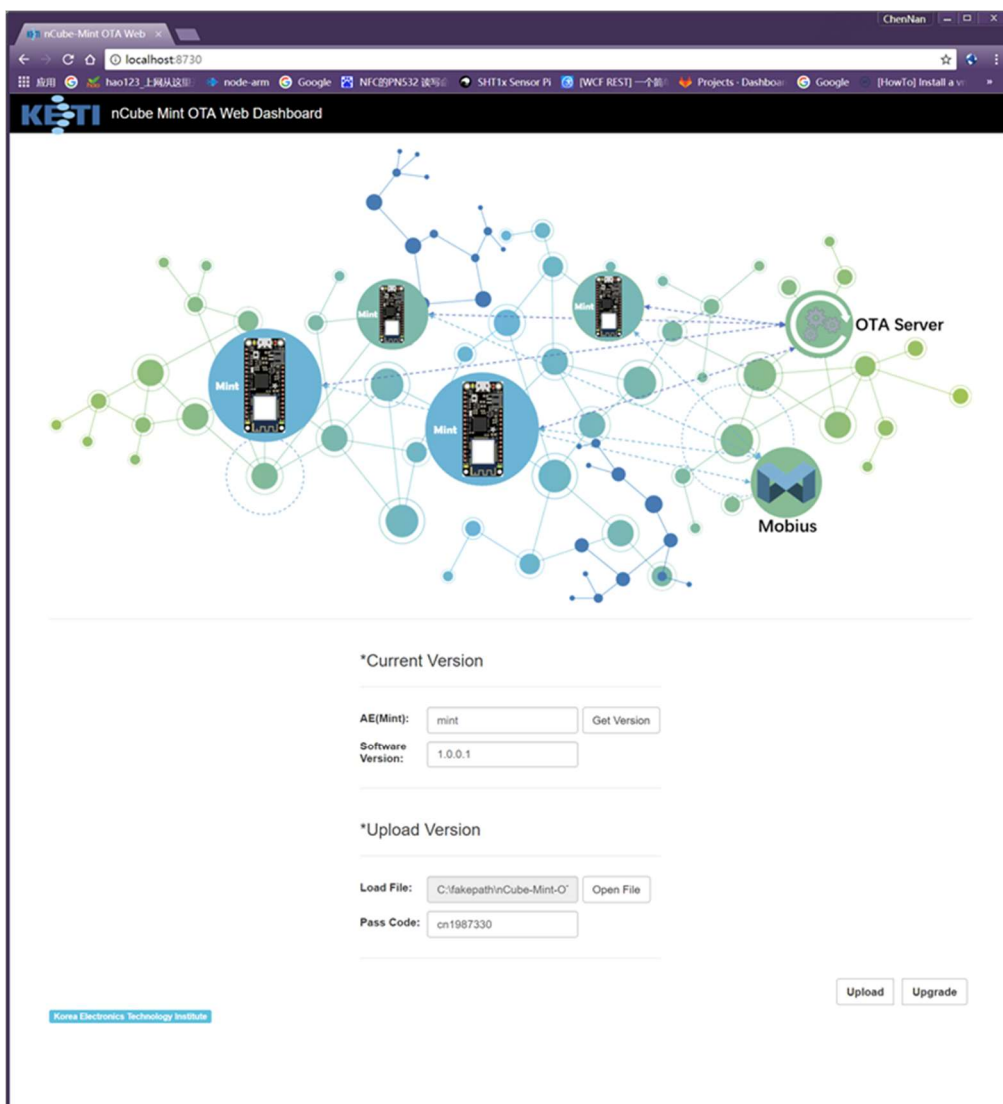


Figure 9 OTA web

Click "Upload" button to send the version "1.0.0.1" software HEX file to the OTA Server.

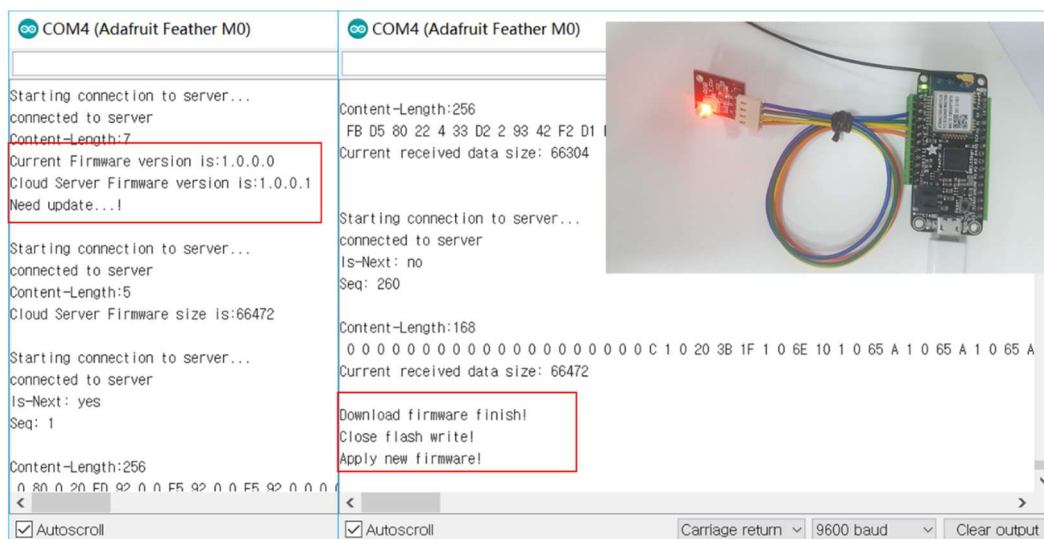
3.4. Upgrade

In sample code, device will wait for upgrading to the new version software until the developer send the command to it. For this, we have created an "update" container and subscription under it for each device. Developer only need to send content instance with content value "active" to the "update" container then software will check update from server. The following example is the update command messages.

```
POST /mobius-yt/[your-ae]/update
Host: 203.253.128.161:7579
Accept: application/json
X-M2M-RI: 12345
X-M2M-Origin: [your-ae]
Content-Type: application/vnd.onem2m-res+json; ty=4

{"m2m:cin":{"con": "active"}}
```

Also, we provide the update command sending function on the OTA web page. Instead of directly send a message, the developer just need to click "Upgrade" button to send an update command to the Mobius Server and then the Mobius Server send a notification message to this device. When device receives the update command it will query the OTA Server about target software version. And then compare it with the version number of software which is running inside. If inside version number is low it will request binary file of new version software.



4. Notice

In our source, the OTA Server IP address was fixed in "OverTheAir" library. If you want to change it to another server please open the "m0_ota.h" file in the "OverTheAir" library folder and then modify the "`FIRMWARE_SERVER_HOST`" and "`FIRMWARE_SERVER_PORT`" parameters.