

Document Number	BQW_02_0020.001
-----------------	-----------------

MiniHub Pro

User Guide

Version 1.6



BROWAN

Browan Communications Inc.

No.15-1, Zhonghua Rd., Hsinchu Industrial Park,

Hukou, Hsinchu, Taiwan, R.O.C. 30352

Tel: +886-3-6006899

Fax: +886-3-5972970

Index

Index	2
Release History	3
About this Document	4
Required Equipment	4
Wiring Diagram	5
Software Package Requirement	6
Erase Flash and Upgrade Firmware	11
Erase flash	12
Firmware Upgrade	13
Monitor	14
Web Provision	15
Connect to Web GUI	15
AWS & LoRa Setting	17
WiFi Setting	19
Station EUI Writing	20
LED Behavior	21
Reset to Default	22
OTA	23
Q&A	31

Release History

Date	Version	Author	Comment
2020/01/21	1.1	Jason Andrew Lin Andrew Shiu Crux	• First release.
2020/04/16	1.2	Jason Andrew Lin Andrew Shiu Crux	• Add Web GUI for AWS IoT provision. • Add Web GUI for Basic Station provision. • Add writing station EUI command. • Add LED behavior.
2020/04/16	1.3	Jason Andrew Lin Andrew Shiu Crux	• Fix some typos.
2020/05/27	1.4	Jason Andrew Lin Andrew Shiu Crux	• Add Q&A.
2020/06/22	1.5	Jason Joey	• Add OTA flow.
2020/06/30	1.6	Jason	• Correct some wording • Browan Official Release, add Document Number

About this Document

This document explains how to erase the MiniHub Pro flash (Model Name: TBMH110), how to upgrade new firmware, and the WiFi behaviors after powering up. The Web GUI usage for AWS IoT provision and Basic Station provision.

Required Equipment

- MiniHub Pro

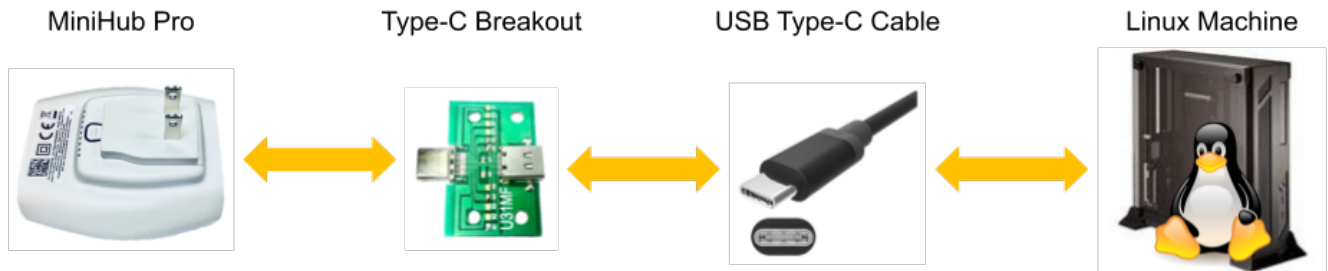


- USB Type C Breakout Board



- USB Type C Cable
- Linux machine (We use Ubuntu 14.04 LTS in the document)

Wiring Diagram



You must use the Type-C breakout-board provided by Browan to erase the flash or upgrade the firmware.

Please pay attention to the direction. The label of the Type-C breakout board is aligned with the word “RESET”.



Software Package Requirement

The original development tools are from Amazon FreeRTOS GitHub. To make it easier for non-developers to update the firmware, Browan extracts the required parts and repackages them into browan_esptool.

Please refer to `browan_esptool-v [X]. [Y] .tar.gz` which attached in this document and copy it to your Linux machine.

```
$ ls -l
-rw-rw-r-- 1 vagrant vagrant 8896515 Jan 21 08:23 browan_esptool-v2.0.tar.gz
```

Extract the tarball using the following command:

```
$ tar -zxvf browan_esptool-v2.0.tar.gz
$ ls -l
drwxrwxr-x 4 vagrant vagrant 4096 Jan 21 08:21 browan_esptool-v2.0
-rw-rw-r-- 1 vagrant vagrant 8896515 Jan 21 08:23 browan_esptool-v2.0.tar.gz
```

The folders/files in browan_esptool/:

```
browan_esptool-v2.0/
|-- browan_esptool.sh
|-- esptool
| |-- CONTRIBUTING.md
| |-- LICENSE
| |-- MANIFEST.in
| |-- README.md
| |-- ecdsa
| |-- espefuse.py
| |-- espsecure.py
| |-- esptool.py
| |-- flasher_stub
| |-- idf_monitor.py
| |-- pyaes
| |-- setup.cfg
| |-- setup.py
```

```
| `-- test
|-- images
| |-- aws_demos
| |-- aws_demos.bin
| |-- bootloader.bin
| |-- ecdsasigner.crt
| |-- ecdsasigner.key
| |-- ota_data_initial.bin
| |-- partition-table.bin
| `-- storage.bin
|-- install_pkg.sh
`-- requirements.txt
```

- browan_esptool.sh is a script to invoke esptool commands
- esptool/ provided by Amazon FreeRTOS
- images/ includes the firmware images released by Browan. New firmware updates only need to update this folder.
- `install_pkg.sh` is a script to install the required packages
- requirements.txt is the python module require list for esptool

In Ubuntu 14.04 LTS, you need to install the following packages:

```
* libssl-dev
* libffi-dev
* python
* python-pip
* python-setuptools
* python-serial
* python-pyparsing
```

And install the following packages using pip:

```
* testresources
* setuptools
* pyserial>=3.0
* future>=0.15.2
```

* cryptography>=2.1.4
* pyparsing>=2.0.3,<2.4.0

MiniHub Pro uses the CP2102 chip as a USB-to-serial interface. Your system needs a correct driver.

Connect the MiniHub to the type-C breakout board and then connect it to your host Linux. Use the following commands to check whether the system automatically mounts the CP2102 driver.

```
$ lsmod | grep cp210x
cp210x          28672  0
usbserial      49152  2 pl2303,cp210x
```

If UART does not work well, download the "CP210x USB to UART Bridge VCP Drivers" from Silicon Labs official website, and install the device driver following the guideline.

- <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

If not, you need to install the following packages:

```
* linux-modules-extra-`uname -r`
```

If you need to monitor the MiniHub Pro terminal screen on Linux, you would need to add the user to the `dialout` group.

```
$ sudo usermod -a -G dialout $USER
```

The above installation and settings operations can be completed automatically using the `install_pkg.sh` provided by Browan. To apply the installed drivers, please restart the system after executing `install_pkg.sh`.

```
$ cd browan_esptool-v2.0/

$ ls -l
total 20
-rwxrwxr-x 1 andrew andrew 1081 Jan 21 16:21 browan_esptool.sh
drwxrwxr-x 6 andrew andrew 4096 Jan 21 16:19 esptool
drwxrwxr-x 2 andrew andrew 4096 Apr 16 11:44 images
-rwxrwxr-x 1 andrew andrew 620 Jan 21 15:59 install_pkg.sh
-rw-rw-r-- 1 andrew andrew 434 Jan 21 14:37 requirements.txt

$ ./install_pkg.sh
```

\$ reboot

Erase Flash and Upgrade Firmware

If your system installed correct CP2012 driver, when MiniHub Pro connects to the host Linux through USB, Linux should automatically generate the corresponding /dev/ttyUSB? character-device. You can use the following command to verify:

```
$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Bridge / myAVR mySmartUSB light

                ^^^^^^

Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

$ ls -l /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Jan 21 09:03 /dev/ttyUSB0
```

You can use `browan_esptool.sh` to do flash erasing and programming operations. Please note that before executing an erase or program, you need to power off the device, then hold down the SETUP button and reconnect the USB power supply. The device will switch to flash download mode before executing `browan_esptool.sh`



Erase flash

1. Remove the USB power cable.
2. Press and hold the SETUP button and then reconnect the USB power cable.
3. Execute `./browan_esptool.sh` and pass in the ttyUSB path and operation command "erase" as parameter s.
4. Power recycle after finish step 3.

Here is the output of browan_esptool.sh:

```
$ ./browan_esptool.sh /dev/ttyUSB0 erase
esptool.py v2.8-dev
Serial port /dev/ttyUSB0
Connecting.....
Detecting chip type... ESP32
Chip is ESP32D0WDQ5 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 24:6f:28:3f:2e:18
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Erasing flash (this may take a while)...
Chip erase completed successfully in 8.2s
Hard resetting via RTS pin...
$
```

Firmware Upgrade

1. Remove the USB power cable.
2. Press and hold the SETUP button and then reconnect the USB power cable.
3. Execute `./browan_esptool.sh` and pass in the ttyUSB path and operation command "flash" as parameters.
4. Power recycle after finish the step 3.

Here is the output of browan_esptool.sh:

```
$ ./browan_esptool.sh /dev/ttyUSB0 flash
esptool.py v2.8-dev
Serial port /dev/ttyUSB0
Connecting.....
Detecting chip type... ESP32
Chip is ESP32D0WDQ5 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 24:6f:28:3f:2e:18
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Compressed 28224 bytes to 16081...
Wrote 28224 bytes (16081 compressed) at 0x00001000 in 0.4 seconds (effective 618.0 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 132...
Wrote 3072 bytes (132 compressed) at 0x00008000 in 0.0 seconds (effective 2176.0 kbit/s)...
Hash of data verified.
Compressed 8192 bytes to 31...
Wrote 8192 bytes (31 compressed) at 0x00016000 in 0.0 seconds (effective 14256.4 kbit/s)...
Hash of data verified.
Compressed 1336640 bytes to 807124...
Wrote 1336640 bytes (807124 compressed) at 0x00020000 in 19.0 seconds (effective 563.3 kbit/s)...
Hash of data verified.
```

Leaving...
Hard resetting via RTS pin...
\$

Monitor

1. Power recycle.
2. Execute `./browan_esptool.sh` and pass in the ttyUSB path and operation command "monitor" as parameters.

Here is the output of browan_esptool.sh:

```
$ ./browan_esptool.sh /dev/ttyUSB0 monitor
--- idf_monitor on /dev/ttyUSB0 115200 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

esp32>
```

If you don't want to use the monitor command provided by esptool. You can use any terminal tool such as minicom, screen, picocom on Linux, or use putty on Windows.

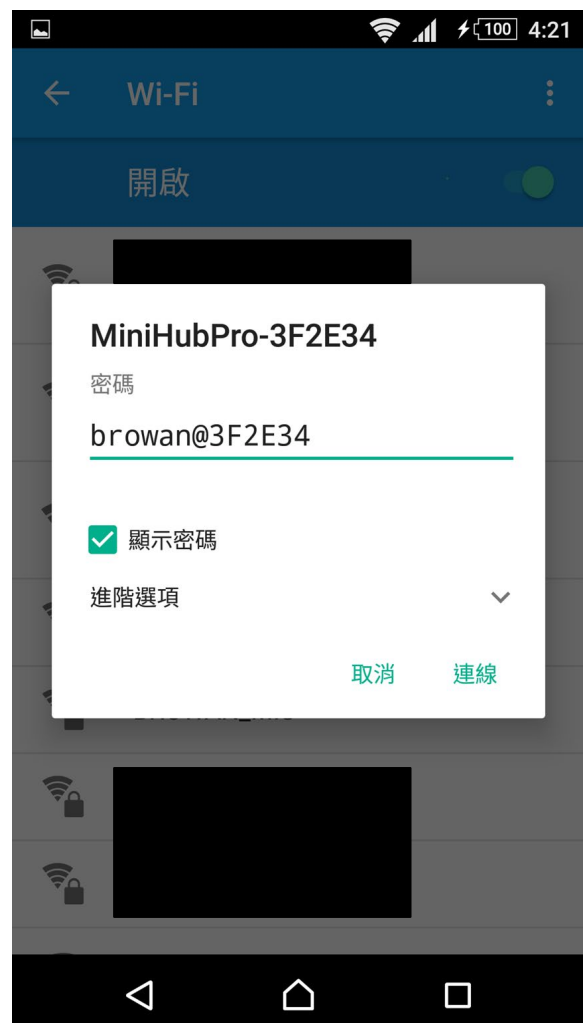
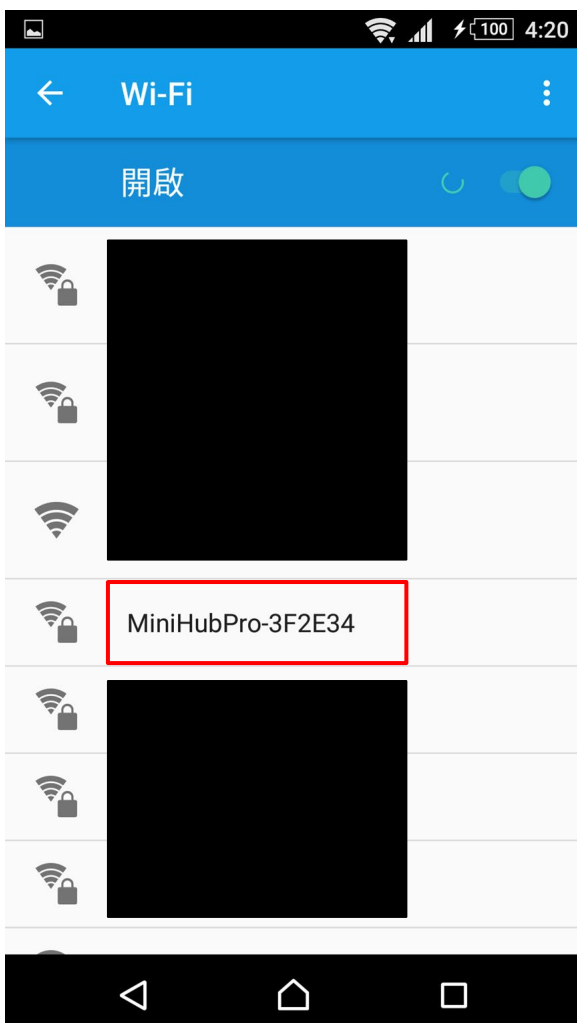
The UART connection settings is: **115200-8-N-1**

Web Provision

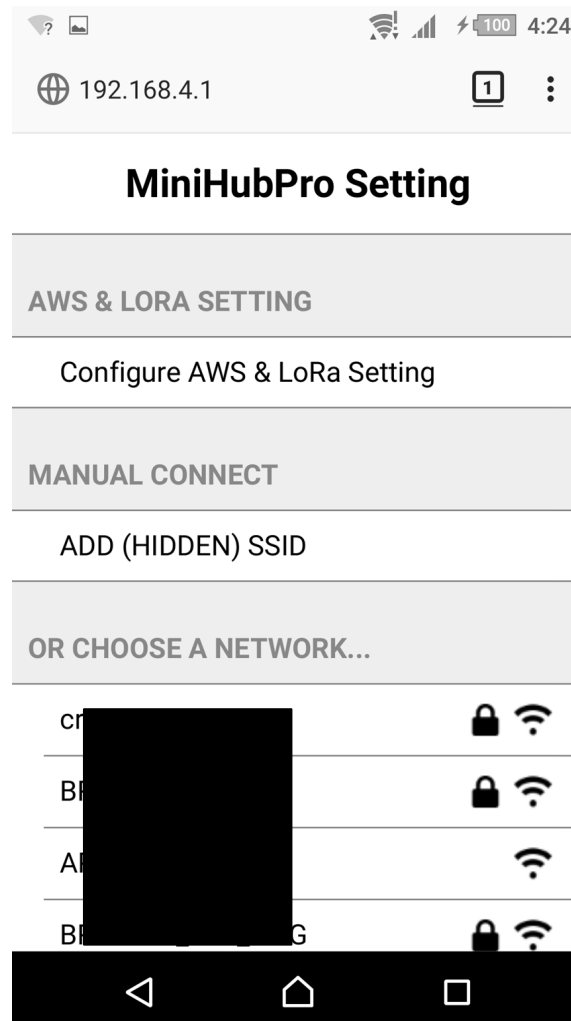
Connect to Web GUI

The device can run as WiFi AP mode or WiFi Station mode. When the device in the initial state, such as first boot-up time or after reset-to-default. It will run with the WiFi AP mode. That means it accepts any WiFi client to connect to it.

You can find the SSID `MiniHubPro-XXXXXX` in the WiFi site-survey list. The suffix 6 characters are the last 6 hex string of WiFi MAC address. The password is in the back label.

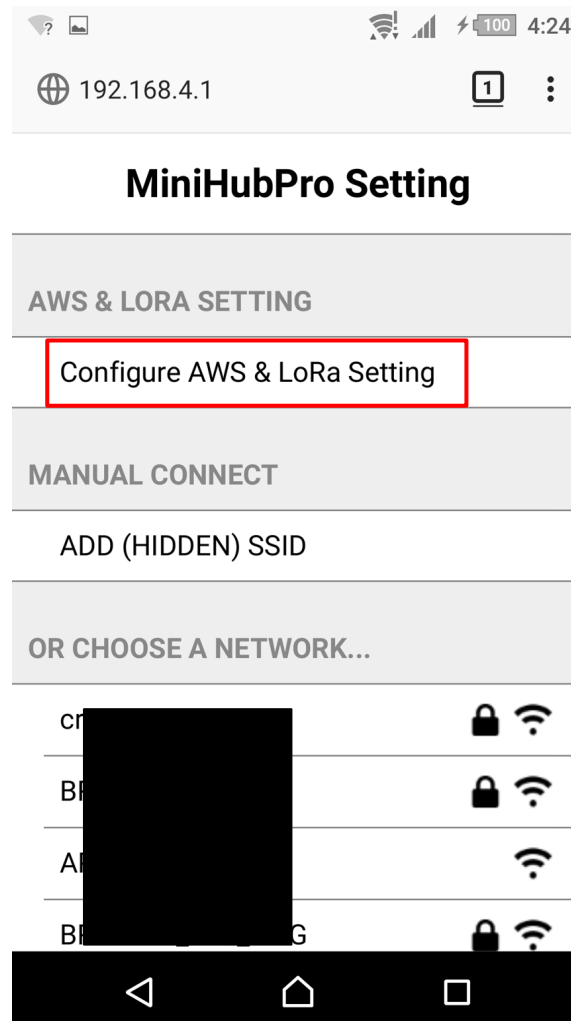


After connected to **MiniHubPro-XXXXXX** AP, it will open the setup page. If web page doesn't open automatic ally, please using **Firefox or Chrome** to open **192.168.4.1** manually.



AWS & LoRa Setting

Click "Configure AWS & LoRa Setting" to open setting page.



There are two parts, one is for AWS, and another one is for LoRa. Please configure your setting and click the "Save" button at the bottom. If you don't want to change any setting, please click the "Cancel" button at the bottom.

192.168.4.1



AWS & LNS Setting

GATEWAY MAC

246F283F2E34

AMAZON WEB SERVICES (AWS)

AWS IoT Endpoint URI:

a2bpldx3pawckp-ats.iot.us-east-1.am

AWS IoT Endpoint Thing Name:

7EDB626E1B

Certificate: (*.der)

Browse... No file selected.

Private Key: (*.der)

Browse... No file selected.

LORA NETWORK SERVER (LNS)

CUPS Enable: ☒

CUPS

Type: ☒ Boot ☐ Regular

CUPS URI:

https://s2.sm.tc:7007

CUPS Trust: (installed)

Browse... No file selected.

CUPS CRT: (installed)

Browse... No file selected.

CUPS Key: (installed)

Browse... No file selected.

LNS

LNS URI:

Address:Port

LNS Trust: (non-install)

Browse... No file selected.

LNS CRT: (non-install)

Browse... No file selected.

LNS Key: (non-install)

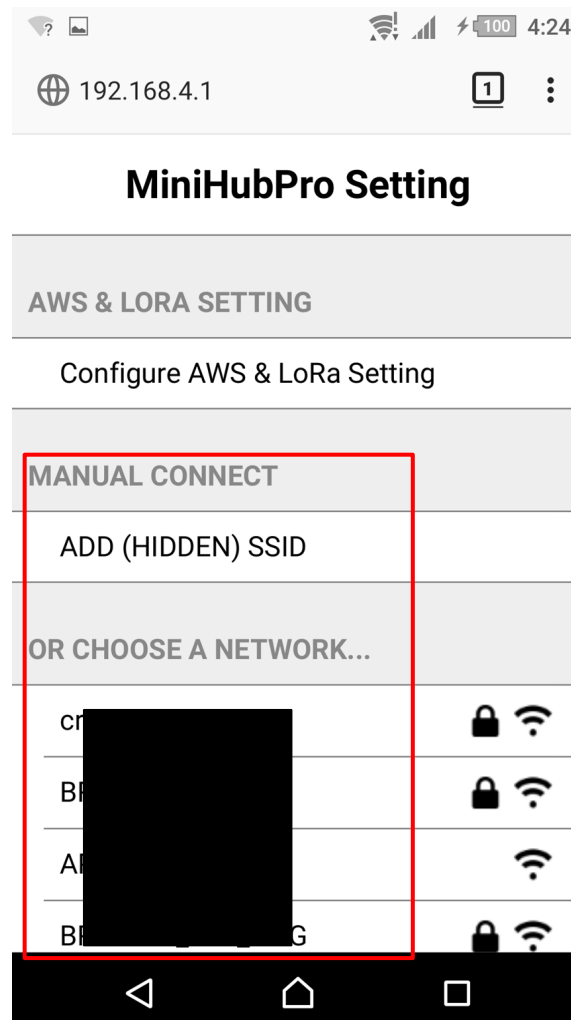
Browse... No file selected.

Cancel

Save

WiFi Setting

Choose one of the WiFi AP which you prefer to connect to the internet. You also can add SSID manually by yourself at this page. After that, the MiniHub Pro will store the connection information and switch to the WiFi Station mode.



You can also set connection information and switch to WiFi Station mode in CLI prompt:

```
esp32> wifi_conf my_ssid my_pwd
esp32> wifi_info
WiFi MAC   : 24:6F:28:3F:2E:34
WiFi SSID  : MiniHubPro-3F2E34
WiFi PWD   : browan@3F2E34
Station SSID: my_ssid
Station PWD : my_pwd
esp32> opmode_conf 1
esp32> restart
```

*Note:

Station SSID is the connected WiFi Router's SSID

Station PWD is the connected WiFi Router's Password

Station EUI Writing

You can write a Station EUI (6 bytes in HEX) in CLI prompt:

```
esp32> lora_sync 80029cXXXXXX
esp32> lora_sync
loramac from nvs= 80029cXXXXXX
```

After writing the EUI, you should restart the device to apply the setting.

LED Behavior

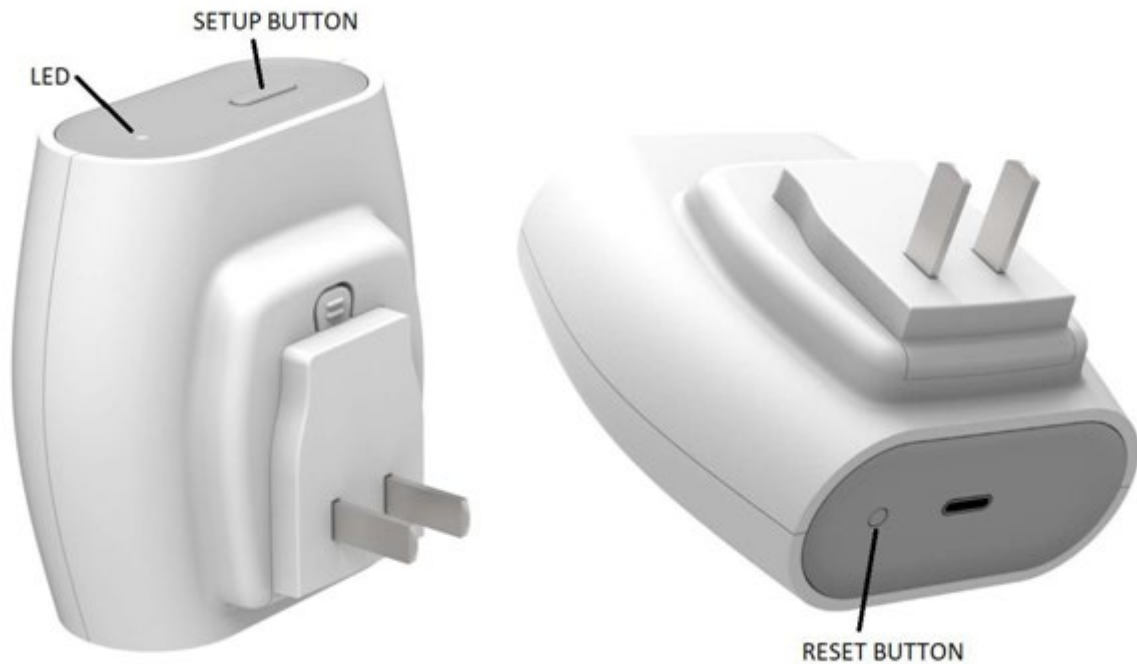
Colors	Blink Pattern	Mode	Status
Green	Blinking 1 sec	WIFI_STA	WiFi station not connected
Green	Blinking 1/4 sec	WIFI_STA	WiFi station connected, establishing the connection to LNS, configuring radio
Green	Solid	WIFI_STA	WiFi station connected, Sta is connected to LNS, radio listening
Green/ Orange	Blinking 1/4 sec	WIFI_STA	WiFi station connected, CUPS transaction in progress *Note: Do not unplug device in this state
Orange	Blinking 1/4 sec	CONFIG	Scanning WiFi networks, setting up configuration AP
Orange	Blinking 1 sec	CONFIG	Configuration AP active

*Note:

WIFI_STA is WiFi Station Mode

CONFIG is Configuration Mode

Reset to Default



Press the reset button over 5 seconds to reset the system to default status. After reset to default, the orange LED will blink every 1 second. You can also use `restore_default` command to reset to default in CLI prompt:

```
esp32> restore_default
Restore to default...
Restore to default finished.
esp32>
```

OTA

1. Please register things for MiniHub Pro on the AWS IoT and configure the AWS & LNS Setting.

AWS & LNS Setting

GATEWAY MAC
246F283D807C
AMAZON WEB SERVICES (AWS)
AWS IoT Endpoint URI:
aawdy7dhw9u1-ats.iot.ap-northeast-1.amazonaws.com:8883
AWS IoT Endpoint Thing Name:
MiniHubPro-3D807C
Certificate: (*.der)
Choose File cert.der
Private Key: (*.der)
Choose File privatekey.der

2. Create an Amazon S3 bucket to store your update.

REF: <https://docs.aws.amazon.com/freertos/latest/userguide/dg-ota-bucket.html>

3. Create an OTA Update service role.

REF: <https://docs.aws.amazon.com/freertos/latest/userguide/create-service-role.html>

4. Create an OTA user policy.

REF: <https://docs.aws.amazon.com/freertos/latest/userguide/create-ota-user-policy.html>

*If you use the "Administrator" user, you can skip this step.

5. Create a FreeRTOS OTA update job

- a. Go to "IoT Core" service.
- b. Go to "Manage Jobs" and click the "Create" button.
- c. Select "Create OTA update job"

- d. Select the things name which configured to MiniHub Pro. And click "Next".

Select devices to update

Browse and select the devices you want to include in this job.

1 thing(s) and 0 thing group(s) selected.

Close

Things	Thing groups	Summary
☒	MiniHubPro-3D807C	

- e. Select the "MQTT" protocol

Select the protocol for firmware image transfer

HTTP and MQTT protocols are supported for firmware updates. [Learn more](#)

- ☐ HTTP [?](#)
- ☒ MQTT

- f. Select the "Sign a new firmware image for me."

Select and sign your firmware image

Code signing ensures that devices only run code published by trusted authors and that the code has not been altered or corrupted since it was signed. You have three options for code signing. [Learn more](#)

- ☒ Sign a new firmware image for me
- ☐ Select a previously signed firmware image
- ☐ Use my custom signed firmware image

- g. Create a new Code signing profile

Code signing profile [Learn more](#)

No code signing profile selected

CreateSelect

- Click "Create"
- Input the "Profile name"
- Select hardware platform: **ESP-WROVER-KIT**
- Import the "Certificate"

Certificate:
<https://drive.google.com/file/d/1SFUX11uqm3OWOhDs5TyDo62jlqlksmGO/view?usp=sharing>

Certificate private key:

<https://drive.google.com/file/d/1EavG36gmL3cdkQxqTrTZIWTjDPm4Mmz4/view?usp=sharing>

- Input the Pathname of code signing certificate on device: P11_CSK
- Click "Create"

*Next time you can select this profile directly.

h. Upload the firmware

Select your firmware image in S3 or upload it

Image not selected Select

- Click "Select"
- Choose the bucket which store the firmware image.
- Click "Upload an image" Upload the image file: aws_demos.bin

i. Input the pathname of firmware image on the device: P11_CSK

j. Select "IAM role for OTA update job" (Created at step.2)

k. Input the OTA job unique ID and click the "Create" button.

l. You can find the successfully created job message.

Successfully created job. [View Job.](#) ✕

m. The OTA job status is "Queued"

JOB

AFR_OTA-minihubpro_ota_demo_0001

IN PROGRESS

Actions ▾

Overview

Details

Resource Tags

Last updated Jun 16, 2020 8:24:41 PM +0800

All Statuses Refresh

1 Queued	0 In progress	0 Timed out	0 Failed	0 Succeeded	0 Rejected	0 Canceled	0 Removed
-------------	------------------	----------------	-------------	----------------	---------------	---------------	--------------

Resource	Last updated	Status
> MiniHubPro-3D807C	Jun 16, 2020 8:24:38 PM +0800	Queued ...

6. Now please power on the MiniHub Pro. To trigger the OTA job process.

Current App Version: 20200601_TB-300_release

```
* Application information:
* Project name:    esp-idf
* App version:    20200601_TB-300_release
* Compile time:   Jun  1 2020 17:55:15
* ELF file SHA256: 6208adb82674992c...
* ESP-IDF:       v3.3-163-g601a03e
```

Current OTA Version: 0.9.2

```
12 621 [iot_thread] INFO: NVS> iot_thing_name = [MiniHubPro-3D807C]
13 621 [iot_thread] OTA Version 0.9.2
```

Start OTA Job:

```
28 1111 [OTA Agent Task] [prvOTAAgentTask] Called handler. Current State [Ready] Event [Start] New state [RequestingJob]
29 1111 [OTA Agent Task] [INFO ][MQTT][11110] (MQTT connection 0x3ffef418) SUBSCRIBE operation scheduled.
30 1111 [OTA Agent Task] [INFO ][MQTT][11110] (MQTT connection 0x3ffef418, SUBSCRIBE operation 0x3fffbbbc) Waiting for operation completion.
31 1121 [OTA Agent Task] [INFO ][MQTT][11210] (MQTT connection 0x3ffef418, SUBSCRIBE operation 0x3fffbbbc) Wait complete with result SUCCESS.
32 1121 [OTA Agent Task] [prvSubscribeToJobNotificationTopics] OK: $aws/things/MiniHubPro-3D807C/jobs/$next/get/accepted
33 1121 [OTA Agent Task] [INFO ][MQTT][11210] (MQTT connection 0x3ffef418) SUBSCRIBE operation scheduled.
34 1121 [OTA Agent Task] [INFO ][MQTT][11210] (MQTT connection 0x3ffef418, SUBSCRIBE operation 0x3fffbbbc) Waiting for operation completion.
35 1131 [OTA Agent Task] [INFO ][MQTT][11300] (MQTT connection 0x3ffef418, SUBSCRIBE operation 0x3fffbbbc) Wait complete with result SUCCESS.
36 1131 [OTA Agent Task] [prvSubscribeToJobNotificationTopics] OK: $aws/things/MiniHubPro-3D807C/jobs/notify-next
37 1131 [OTA Agent Task] [prvRequestJob_Mqtt] Request #0
38 1131 [OTA Agent Task] [INFO ][MQTT][11310] (MQTT connection 0x3ffef418) MQTT PUBLISH operation queued.
```

39 1131 [OTA Agent Task] [INFO][MQTT][11310] (MQTT connection 0x3ffef418, PUBLISH operation 0x3fffbbbc) Waiting for operation completion.

40 1138 [OTA Agent Task] [INFO][MQTT][11380] (MQTT connection 0x3ffef418, PUBLISH operation 0x3fffbbbc) Wait complete with result SUCCESS.

41 1138 [OTA Agent Task] [prvOTAAgentTask] Called handler. Current State [RequestingJob] Event [RequestJobDocument] New state [WaitingForJob]

42 1139 [OTA Agent Task] [prvParseJobDoc] Size of OTA_FileContext_t [64]

43 1139 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [clientToken: 0:MiniHubPro-3D807C]

44 1139 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [jobId: AFR_OTA-minihubpro_ota_demo_0001]

45 1139 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [protocols: ["MQTT"]]

46 1139 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [streamname: AFR_OTA-7bd6fc8c-d14f-4a3c-8789-08aee20d6cc4]

47 1139 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [filepath: P11_CSK]

48 1139 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [filesize: 1312384]

49 1139 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [fileid: 0]

50 1139 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [certfile: P11_CSK]

51 1139 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [sig-sha256-ecdsa: MEUCIQC1luQ8fw/5qJbMeVJYGVbvXULR...]

52 1139 [OTA Agent Task] [prvParseJobDoc] Job was accepted. Attempting to start transfer.

Downloaded and verified:

1182 3338 [OTA Agent Task] [prvIngestDataBlock] Received final expected block of file.

1183 3338 [OTA Agent Task] [prvStopRequestTimer] Stopping request timer.

1184 3341 [OTA Agent Task] [INFO][DEMO][33410] Entering get_item_from_nvs with [P11_CSK]

1185 3341 [OTA Agent Task] [INFO][DEMO][33410] Non-Volatile Storage (NVS) handle...[22]

1186 3341 [OTA Agent Task] [INFO][DEMO][33410] Length of the [P11_CSK] is: [365]

1187 3351 [OTA Agent Task] [INFO][DEMO][33510] Leaving get_item_from_nvs

1188 3351 [OTA Agent Task] [prvIngestDataBlock] File receive complete and signature is valid.

1189 3351 [OTA Agent Task] [prvStopRequestTimer] Stopping request timer.

1190 3351 [OTA Agent Task] [prvUpdateJobStatus_Mqtt] Msg: {"status":"IN_PROGRESS","statusDetails":{"self_test":"ready","updatedBy":"0x90002"}}

Upgraded App Version: 20200616_TB-300_release

* Application information:

* Project name: esp-idf
* App version: 20200616_TB-300_release
* Compile time: Jun 16 2020 11:31:12
* AWS APP Version: 0.9.4
* ELF file SHA256: 2af7743e892cd34e...
* ESP-IDF: v3.3-163-g601a03e

Upgraded OTA Version: 0.9.4

12 773 [iot_thread] INFO: NVS> iot_thing_name = [MiniHubPro-3D807C]
13 773 [iot_thread] OTA Version 0.9.4

Update the "SUCCEDED" message to AWS IoT:

42 1293 [OTA Agent Task] [prvParseJobDoc] Size of OTA_FileContext_t [64]
43 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [clientToken: 0:MiniHubPro-3D807C]
44 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [jobId: AFR_OTA-minihubpro_ota_demo_0001]
45 1294 [OTA Agent Task] [prvParseJSONbyModel] Identified parameter [self_test]
46 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [updatedBy: 589826]
47 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [protocols: ["MQTT"]]
48 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [streamname: AFR_OTA-7bd6fc8c-d14f-4a3c-8789-08aee20d6cc4]
49 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [filepath: P11_CSK]
50 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [filesize: 1312384]
51 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [fileid: 0]
52 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [certfile: P11_CSK]
53 1294 [OTA Agent Task] [prvParseJSONbyModel] Extracted parameter [sig-sha256-ecdsa: MEUCIQCl1uQ8fw/5qJbMeVJYGVbvXULR...]
54 1294 [OTA Agent Task] [prvParseJobDoc] In self test mode.
W (13260) ota_pal: Set image as testing!
55 1305 [OTA Agent Task] [prvUpdateJobStatus_Mqtt] Msg: {"status":"IN_PROGRESS","statusDetails":{"self_test":"active","updatedBy":"0x90004"}}
56 1305 [OTA Agent Task] [INFO][MQTT][13050] (MQTT connection 0x3fff979c) MQTT PUBLISH operation queued.
57 1306 [OTA Agent Task] [INFO][MQTT][13060] (MQTT connection 0x3fff979c, PUBLISH operation 0x3fffc000) Waiting for operation completion.

```

58 1312 [OTA Agent Task] [INFO ][MQTT][13120] (MQTT connection 0x3fff979c, PUBLISH operation 0x
3fffc000) Wait complete with result SUCCESS.
59 1312 [OTA Agent Task] [prvUpdateJobStatus_Mqtt] 'IN_PROGRESS' to $aws/things/MiniHubPro-3D80
7C/jobs/AFR_OTA-minihubpro_ota_demo_0001/update
60 1313 [OTA Agent Task] [prvOTA_Close] Context->0x0x3fffe90
61 1313 [OTA Agent Task] [prvOTAAgentTask] Called handler. Current State [WaitingForJob] Event [Recei
vedJobDocument] New state [CreatingFile]
62 1313 [OTA Agent Task] [prvInSelfTestHandler] prvInSelfTestHandler, platform is in self-test.
63 1314 [OTA Agent Task] [prvStartSelfTestTimer] Starting OTA_SelfTest timer.
64 1314 [OTA Agent Task] Received eOTA_JobEvent_StartTest callback from OTA Agent.
65 1322 [OTA Agent Task] [prvStopSelfTestTimer] Stopping the self test timer.
66 1322 [OTA Agent Task] [prvUpdateJobStatus_Mqtt] Msg: {"status":"SUCCEEDED","statusDetails":{"rea
son":"accepted v0.9.4"}}
  
```

7. Go to AWS IoT to check the status. The status is "Succeeded"

Jobs > AFR_OTA-minihubpro_ota_demo_0001

JOB

AFR_OTA-minihubpro_ota_demo_0001

COMPLETED

Actions ▾

Last updated Jun 16, 2020 8:30:40 PM +0800 [All Statuses](#) [Refresh](#)

0	0	0	0	1	0	0	0
Queued	In progress	Timed out	Failed	Succeeded	Rejected	Canceled	Removed

Resource	Last updated	Status
MiniHubPro-3D807C Jun 16, 2020 8:29:49 PM +0800 Queued at Jun 16, 2020 8:24:38 PM +0800 Updated at Jun 16, 2020 8:30:22 PM +0800 View thing details	Jun 16, 2020 8:30:22 PM +0800	Succeeded

Q&A

Where is the LoRa gateway library?

LoRa gateway library is a library to control Semtech LoRa multi-channel RF receiver. The path of the LoRa gateway library is:

```
$TOP_DIR/vendors/browan/boards/tb-300/components/lora_gateway
```

Where is Web GUI's source code?

Web GUI is base on esp32-wifi-manager (<https://github.com/tonyp7/esp32-wifi-manager>). The path of esp32-wifi-manager is:

```
$TOP_DIR/vendors/browan/boards/tb-300/components/esp32-wifi-manager
```

Where is the entry point for LoRa Basic Station?

LoRa Basic Station's entry point is `browan_cmd_station()`. You can find this function call at the end of `lora_basic_station_demo.c`. The path of `lora_basic_station_demo.c` is:

```
$TOP_DIR/demos/lora_basic_station/lora_basic_station_demo.c
```

Federal Communication Commission Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

IMPORTANT NOTE:

Radiation Exposure Statement:

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Country Code selection feature to be disabled for products marketed to the US/CANADA