Microhomie Documentation

Release 1.0.0

Microhomie Team

Jul 13, 2019

Contents

Welcome! This is the documentation for Microhomie v1.0.0.

Microhomie is a MicroPython framework for Homie, a lightweight MQTT convention for the IoT.

The main target device is the ESP8266 and we deliver pre-build firmware images. For ESP32 devices see the *Install Microhomie on the ESP32* section.

CHAPTER 1

Install Microhomie on the ESP8266

The first thing you need to do is to load the Microhomie firmware, a modified MicroPython firmware, onto your ESP8266 device. You can download the firmware from the GitHub release page.

If you just started with MicroPython, a good start is the Getting started with MicroPython on the ESP8266 from the MicroPython documentation.

CHAPTER 2

Configuration

To configure your Microhomie device create a `settings.py` file from the `settings.example.py` file, make your changes and copy the file to your ESP8266 device.

CHAPTER 3

Get started with a simple LED node

The LED example in this guide use the on-board LED, so you don't need to do any wiring to get started.

Copy the `main.py` file from the `examples/led` directory to your ESP8266, reset the device and watch the incoming MQTT messages.

To turn the on-board LED from your ESP8266 on or off send `true`, `false` or `toggle` to the property topic. For example:

```
mosquitto_pub -h HOST -u USER -P PASSWORD -t "homie/DEVICE-ID/led/power/set" -m true
```

CHAPTER 4

Build your own Microhomie ESP8266 firmware

If you want to build your own Microhomie firmware, maybe for helping us with development, learning or just for the fun to build your own firmware, follow the next steps.

First clone the Microhomie repository:

```
git clone https://github.com/microhomie/microhomie.git
```

The next step is to setup the build environment, build the esp-open-sdk (Requirements and Dependencies), get the MicroPython source, prepare it for the Microhomie firmware and download the required MicroPython modules:

```
cd microhomie
make bootstrap
```

Now you can build your Microhomie firmware and load it to your ESP8266:

```
make
```

Erase and flash:

```
make delpoy PORT=/dev/ttyUSBX
```

Just flash:

```
make flash PORT=/dev/ttyUSBX
```

If you want to help with development, please use our linting:

```
make lint
```

CHAPTER 5

Install Microhomie on the ESP32

For the ESP32 you can just copy all requirements and Microhomie to your device.

Flash MicroPython to your your ESP23 with the official the Firmware for ESP32 boards.

Clone the Microhomie repository and get all the requirements.

```
git clone https://github.com/microhomie/microhomie.git
make requirements
```

The requirements will be downloaded to the path `./lib`.

To save some RAM on the ESP8266 we use the minimal version from mqtt_as.py. You need to download the file with ESP32 support and overwrite the existing:

```
curl -s -o lib/mqtt_as.py https://raw.githubusercontent.com/kevinkk525/micropython-
↪mqtt/master/mqtt_as.py
```

Next copy `lib` and `homie` from your host to the device. `homie` should be copied to the device `lib` directory.

```
lib/
├── aswitch.py
├── asyn.py
├── homie
│   ├── constants.py
│   ├── device.py
│   ├── __init__.py
│   ├── node.py
│   ├── property.py
│   └── utils.py
├── mqtt_as.py
└── uasyncio
    ├── core.py
    └── __init__.py
```

For example we have an mpfshell script `esp32_install.mpf` to automate the delpoyment:

```
mpfshell ttyUSB0 -s esp32_install.mpf
```

Continue with the *Configuration* and the *Get started with a simple LED node*.