

Sonoff Mini with firmware 3.6: Flashing to Tasmota

by Lukas Kukacka

September 9, 2020

in [SmartHome](#)

I guess you know the magic of **Sonoff Mini** and want to **flash it to Tasmota** (or any other alternative firmware like ESPHome should work - however, not tested!).

You have seen articles, tutorials and will follow them. I was mostly following How-to on Reddit ([Part 1](#), [Part 2](#), [Part 3](#)).

I recommend you to read them, but not follow them yet. If you do, you will probably not succeed and when googling around, likely end up on this [Github issue: Sonoff Mini doesn't go into DIY mode with 3.5.0 update](#).

What is the problem then?

The guides / how-tos will probably contain following steps:

1. Set up the Sonoff Mini the standard way to add it official *eWeLink* app.
2. Update Sonoff to latest firmware.

There it goes. Latest firmware at the time of writing this article is **3.6.x**. However, most of the guides on flashing Sonoff I saw are relying on official [Sonoff Devices DIY Tools](#).

The problem with official tooling?

- Since firmware **3.5.x** the API of Sonoff devices has changed to [protocol v2.0](#)
- It seems the official tools are still made only for version **3.3.0** (see [/tool/](#)), so **the tooling will probably not work**. I did even try because...
- Tools is Windows-only exe

Any good news?

Actually, YES! With the latest firmware using protocol v2.0 (i.e. 3.6.x) there is **no need to open the device and install DIY jumper**. DIY mode to allow OTA update can be enabled just using API calls.

Steps for flashing to Tasmota

⚠ Disclaimer: I don't take any responsibility for any damages or other harm caused by anything in this article. Please make sure you know what you are doing and stay safe. You are working with electric device. Make sure it is disconnected from power when manipulating with the device. Improper manipulation can cause significant harm or injuries! ⚡🔧⚡🤖

Below are the extra steps I had to make to have Sonoff Mini successfully flashed to Tasmota.

All the API calls are well documented in [SONOFF DIY MODE Protocol Doc v2.0 Doc.pdf](#).

There is few valuable comments in the [Github issue](#) which pointed me to the right direction.

Please be aware that I have followed the guides first, but got stuck at the point to use Itead Windows-only tooling. All steps before still apply (i.e. you need to power on Sonoff Mini, find its IP address, etc.). I made this articles from memory and mostly as notes for myself, didn't have more Mini's to try it on. Please let me know if you find some issues.

(Optional) Postman requests collection

I have used [Postman](#) to send all requests to Sonoff Mini while flashing. You can **download my Postman collection** at [Github Gist](#).

The collection has all the requests you will need. You just have to:

- Specify `baseURL` in environment (or just rewrite in URL) to `IPAddress:8081` of Sonoff Mini (e.g. `192.168.1.123:8081`)
- Update URL and SHA256 hash of firmware binary (see below).

Enable DIY mode

Send `POST` request to `http://{{baseURL}}/zeroconf/ota_unlock` with payload

```
{
  "deviceid": "",
  "data": { }
}
```

Yes, really with that payload. The protocol requires it.

Validate OTA is unlocked

Send `POST` request to `http://{{baseURL}}/zeroconf/info` with payload

```
{
  "deviceid": "",
  "data": { }
}
```

Yes, really `POST` request. It is not very REST-like, but that is according to the docs.

You should get answer containing values like this:

```
{
  ...
  "otaUnlock": true,
  "fwVersion": "3.6.0",
  ...
}
```

Cool, **OTA flashing just got enabled**.

Download firmware

I downloaded latest *tasmota-lite.bin* from [Tasmota releases](#). See the other articles for details.

Please note that following steps are specific for macOS and some details might be different for you.

For this article I will expect you to place *tasmota-lite.bin* to `~/Downloads/sonoff-mini-flash` folder. Please adjust accordingly.

Start serving the firmware file

Note: Not sure if pointing the firmware URL directly to Github raw binary (and therefore skipping this step) would work. I didn't try. If anyone did, please let me know.

I used [Docker-based Nginx server](#) to quickly serve the file *the right way* (look at the end of the article for details on why I chose Nginx server).

```
docker run -p 8000:80 \
-v ~/Downloads/sonoff-mini-flash/:/opt/www/files/ \
mohamnag/nginx-file-browser
```

You can just quickly check <http://localhost:8000> if the firmware file is available.

Start flashing

Send POST request to `http://{{baseUrl}}/zeroconf/ota_flash` with payload

```
{
  "deviceid": "",
  "data": {
    "downloadUrl": "http://FILE_SERVER_IP/files/tasmota-lite.bin",
    "sha256sum": "FIRMWARE_SHA256_HASH"
  }
}
```

where:

- `FILE_SERVER_IP` is IP address of your computer (i.e. the server where you have *tasmota-lite.bin* served from)
- `FIRMWARE_SHA256_HASH` is SHA-256 hash of the *tasmota-lite.bin* file
 - On macOS you can get it using `sha256sum tasmota-lite.bin`

When you execute this API call, you should get success response immediately.

You can now observe log from Docker Nginx server and see how Sonoff Mini is fetching firmware chunks.

Wait a moment and when you stop seeing logs in Nginx server, Sonoff Mini will restart.

Done 🎉

Congratulations, all should be done and your **Sonoff Mini is now running Tasmota firmware**.

You can follow Tasmota specific articles on how to set it up.

Don't forget to set [Template](#) in Tasmota config to get the external switches working.

(Optional) More technical details

Server providing firmware binary must support Range header

⚠️ **This is extremely important** and it took me a while to find out what is wrong.

Docs for URL: `http://[ip]:[port]/zeroconf/ota_flash` actually mention:

“ The download address of the new firmware, only supports the HTTP protocol, **the HTTP server must support the Range request header.**

I was first using just a simple Python server (`python -m SimpleHTTPServer 8080`) to serve firmware binary and that is where my main issue was. It does not support Range header. It took me a while of debugging and detailed reading through documentation to find the note about Range header.

Sonoff is downloading the firmware binary from its HTTP server in 4KB chunks using Range header (I guess that is because of the limited memory on the device). The header of request looks something like this:

Range: bytes=0-4095

Without server supporting the Range header, the request to upload firmware was successful, but Sonoff ended up in infinite loop trying to fetch the first 4KB of the firmware binary.

And the end I have decided to go for the simplest and safest option for me: Nginx server in Docker (as mentioned above).

Like the article, got a comment or found an issue? Get in touch via  [Twitter](#) .

Tags: [smart-home](#), [tasmota](#), [sonoff](#)

Subscribe via [RSS](#)

Share: 

Lukas Kukacka

Site Map

[About me](#)

[Posts](#)

[Privacy policy](#)

Contact

 blog@lukaskukacka.com

 [lukaskukacka](#)

 [lukas-kukacka](#)

 [lukaskukacka](#)

 [lukas-kukacka](#)

Subscribe via [RSS](#)

Articles about Swift, iOS development and anything related