Brew Bubbles

Release 2.2.3

Feb 21, 2021

Contents

1 Contributions **3**

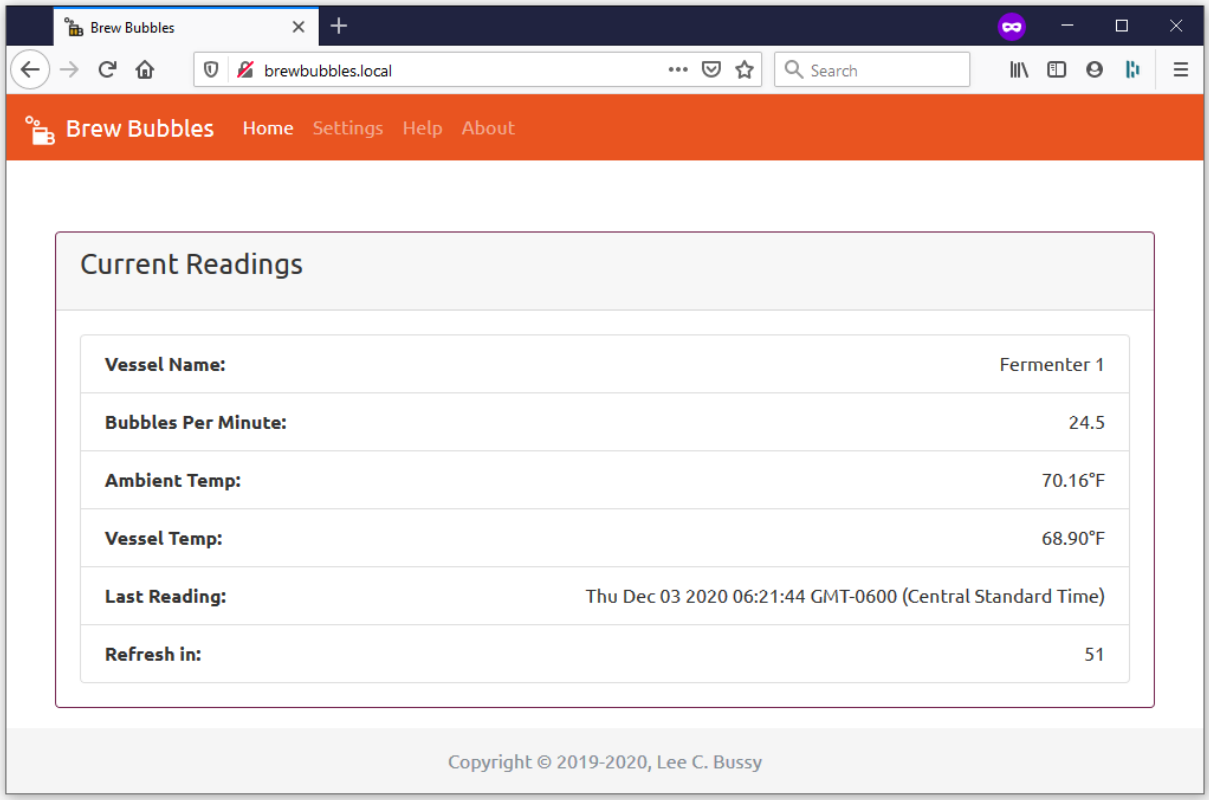
2 Table of Contents **5**

2.1 Project Prerequisites 5
2.2 Device Assembly 5
2.3 Installing the Firmware 16
2.4 Set Up Networking 19
2.5 Operation and Configuration 28
2.6 Firmware Update 41
2.7 WiFi Reset 41
2.8 Brew Bubbles API 43
2.9 Troubleshooting 52

One of the very first things a homebrewer does in the morning after a brew day is rush to the fermenter and look to see if the airlock is bubbling. Just in case you are not quite at the point you take a day off afterward, or even remember to brew on Saturday, this project is for you.

Brew Bubbles is a combination hardware/software solution for homebrewers. Brew Bubbles intends to monitor the fermentation process by counting and reporting bubbles in an S-type airlock as "Bubbles per Minute" (BPM.) Also, since not everyone has a fancy fermentation chamber and temperature control, Brew Bubbles optionally reports your fermenter's temperature as well as that of which the room in which the fermenter rests. All of these reports in a compact mobile-responsive web page served by Brew Bubbles.

If you use temperature control already, such as BrewPi Remix or Fermentrack, Brew Bubbles reports that data to display in that solution. If you use a cloud dashboard for your brews, such as Brewer's Friend, you can display the data there as well.



CHAPTER 1

Contributions

Documentation should always be considered a work in progress; in that, I depend on *you* to help me make sure it is relevant and easy to read. *I* know how to work with Brew Bubbles. My goal is to make you an expert too. Anything that is confusing is a bug, and I want to know about it, please.

For spelling/grammatical errors, or if you would like to improve the documentation, please feel free to issue a pull request with the update. For technical issues with the documentation, please open an issue on GitHub.

CHAPTER 2

Table of Contents

2.1 Project Prerequisites

To use Brew Bubbles, you need to use an S-lock type fermentation lock. These locks are inexpensive and readily available. There are always discussions about the ability to clean these locks. It's my experience that a warm soap and water soak followed by a good rinse handles the nasties. For the price of the airlocks, you could replace them every few batches if you were worried.

These are not perfect airlocks. I would have preferred to devise a way to use the three-piece airlocks; however, in testing, these proved to be exceedingly dependant upon the airlock fill level to work reliably. Still, Brew Bubbles should take its signal from any input. The future may hold additional options.

If you use a blow-off-tube, you would remove that and replace it with the S-lock when blowoff danger has subsided. I have also had excellent results using FermCap-S and not using a blowoff tube despite vigorous fermentation.

You need an ESP8266 controller and some other parts detailed on the Device Assembly page in addition to the S-lock. The *Device Assembly* page has a complete BOM for Brew Bubbles.

Add a tiny bit of time and some skill you can pick up making a couple of these devices, and you have yourself an easy to make, low-cost, and rewarding DIY homebrew project.

2.2 Device Assembly

Brew Bubbles runs on an ESP8266 controller. To detect the information to be logged, you need to connect specific devices to the controller. I have provided a circuit board design, sometimes called a "shield," to make this easier.

Assembly is not difficult, but it does require some basic soldering.

Contents

- *Device Assembly*

- *Materials Required*
  - *Controller*
  - *Printed Circuit Board*
  - *General Parts*
- *Component Installation*
- *Sensors*
- *Light Filter*
- *Bracket and Mounting*
- *Power*

2.2.1 Materials Required

This shield uses one or two each of some widely-available and inexpensive components. You may find you are better off buying a resistor assortment, for instance. Buying 10 or 20 of a piece at a time for a nominal cost from China is also possible. Make several!

Gather the following parts and pieces:

Controller

The ESP8266 controller is paired with many different "developer boards" to make connections easier. Do yourself a favor and do not buy a bare ESP8266 chip. I can't help you if you do.

The developer board used in this project is the **Wemos D1 Mini**. Wherever you purchase it, make sure it is an ESP-12F (if that information is listed) and that it says 4MB (or sometimes shown as 32Mb, which is 4 megaBYTES converted to megaBITS.) Also, be sure it comes with the 8-pin male and female headers, or else you will need to get them elsewhere. You can find the Wemos D1 Mini in many places; here are a few:

1. **The "Legitimate" Way**: The Chinese have knocked the Wemos D1 Mini has mercilessly. This situation is ironic since Espressif makes the ESP8266 controller, and they are a Chinese company. In the past, wemos.cc was the way to buy the "original" Wemos D1 Mini. However, they no longer appear to sell directly, and instead, the website is a wiki-type information source.
2. **The Quick Way**: Why Amazon, of course. You can search for "Wemos D1 Mini" (it has to be "mini" and not "pro") and find a large number of different ways to buy. My preference is to get a handful at a time since they are cheaper that way. This link gives you five for $17.99 in a couple of days with free Prime shipping ($3.60/ea.)
3. **The Cheap Way**: AliExpress has everything anyone would ever need, so long as that person can wait a month or more for delivery. For $1.99 each, this way is as cheap as it gets. Be aware that there are different sellers on there, and each ship independently. Sometimes you save by paying less on shipping if you are buying multiple things from one seller.

Printed Circuit Board

The pcb directory in the repository contains the Eagle files for the printed circuit board shield supporting Brew Bubbles. The shield provides the necessary component connections and circuitry for the ESP8266 controller used in this project. It is nearly identical in size to the Wemos D1 mini we use, providing a very compact and lightweight footprint. A sub-directory also contains Gerber files for people who need those.

Everyone likes to save a dollar, including me. There used to be other recommendations for PCB manufacturers, but quality, responsiveness, and dependability have forced me to limit my recommendations to OSH Park. You can buy three boards for $5.70 (plus shipping) at this link.

Displayed below is a view of the top of the printed circuit board, as rendered by OSH Park.

Below is a view of the bottom of the printed circuit board, as rendered by OSH Park.

If you would like to personalize these board designs, you may modify them with Autodesk's EAGLE. EAGLE is a scriptable electronic design automation (EDA) application with schematic capture, printed circuit board (PCB) layout, auto-router, and computer-aided manufacturing (CAM) features. EAGLE stands for Easily Applicable Graphical Layout Editor and is developed by CadSoft Computer GmbH.

General Parts

These are the parts which you can get pretty much anywhere.

The items marked with an *asterisk below are optional. They are in the design to provide a means to monitor and trend the ambient temperature where you place the fermenter and the fermenting liquid's temperature via a thermowell or insulated in contact with the fermenter. If you choose not to use these, the firmware automatically skips reporting these readings.

Quan	Description	Placement
3	0.1uF 10V Ceramic Capacitor	C1, C2, C3
2*	2.2k ohm 1/4W 5% Axial resistor	R1, R2
1	150 ohm 1/4W 5% Axial resistor	R3
2*	3-Pin 90-degree Header	VESSEL, ROOM
1	GP1A57HR Transmissive Photointerrupter	U1
2*	DS18B20 Temperature sensor and lead	VESSEL, ROOM

This BOM is available on Mouser. You can find these parts in just about any of the usual places. Please make sure they are the proper rating and form factor.

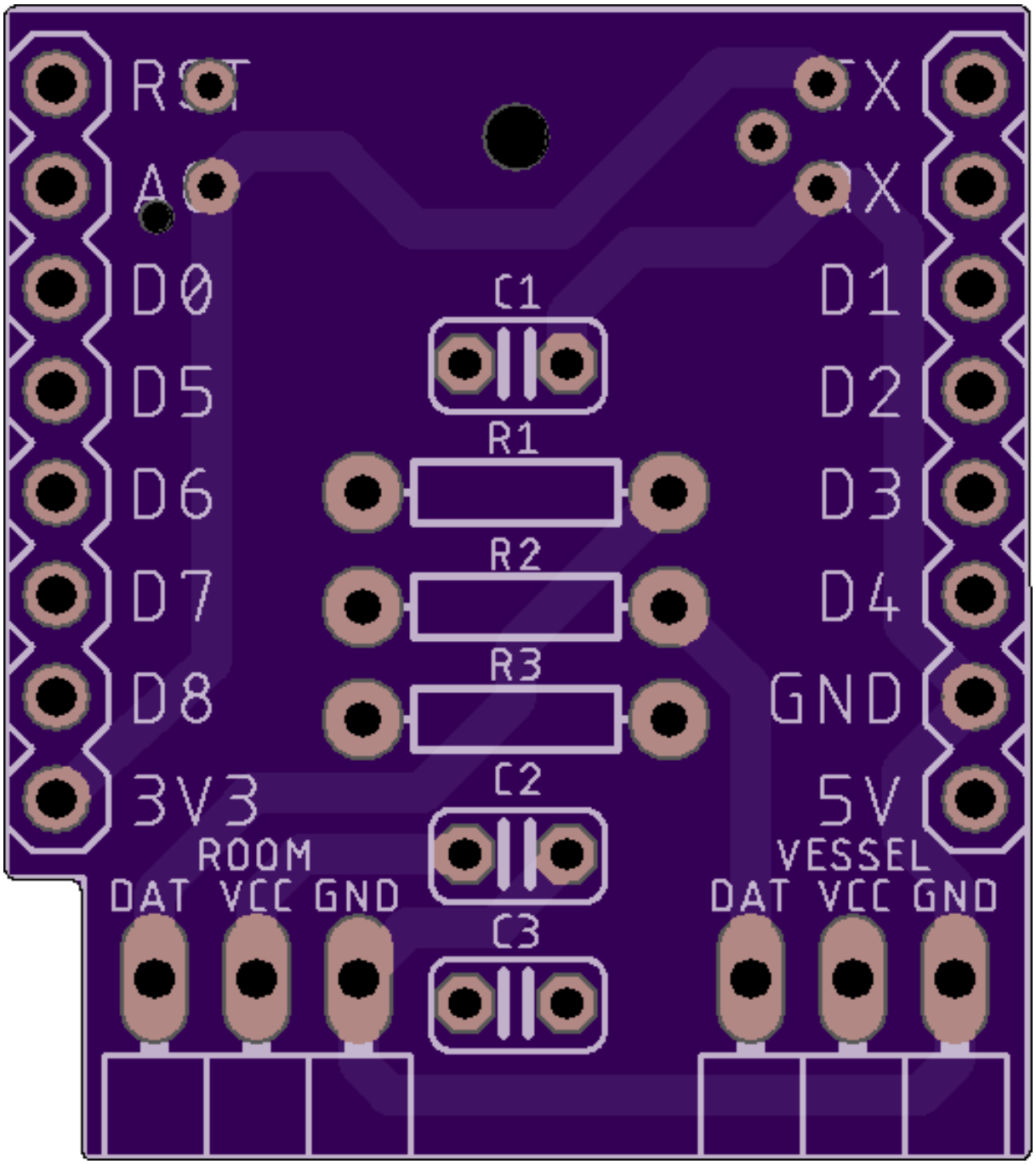
If you choose to use the temperature sensors, you need to obtain two (2) Waterproof DS18B20 Temperature Sensor with leads. These are the same as used with the various fermenter temperature control projects such as BrewPi Remix. You also need to crimp on a three-terminal female plug. You can get a kit on Amazon with more terminals than you ever need, including the crimping tool, for about $25 or save by doing some careful AliExpress shopping.

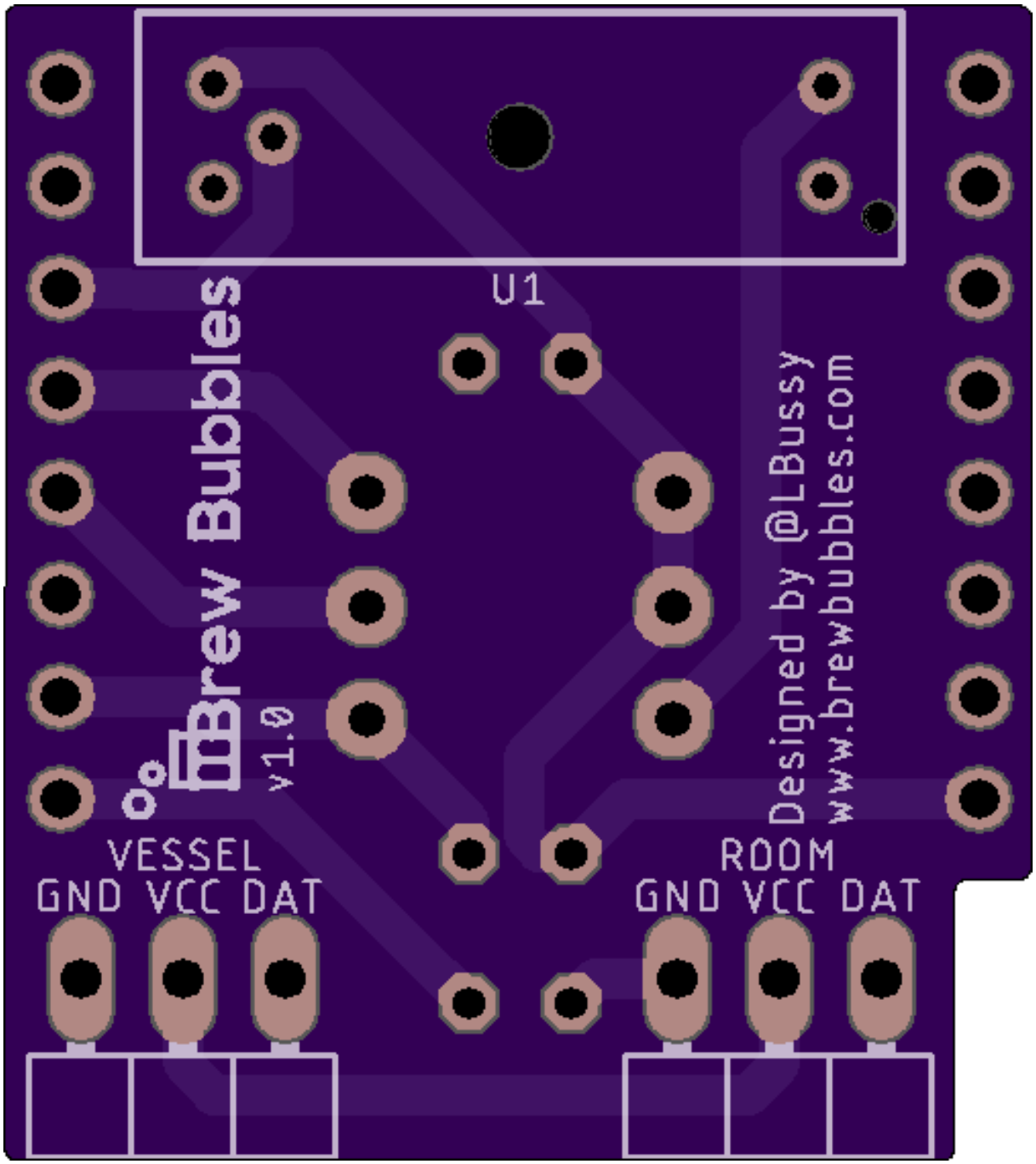
2.2.2 Component Installation

You are going to have to solder. If you have legitimately never soldered anything before, I recommend you spend a few minutes on YouTube and watch a few videos. Sparkfun also has a very nice tutorial. It is not hard at all once you get the hang of it. While the shield is comparatively small, the components chosen are simple through-hole parts, which may be easily soldered by a beginner with a little patience.

I do not intend to provide a step-by-step on how to solder here. Still, I recommend the following part installation order for ease of assembly:

1. Resistors - As the shortest mounted components, soldering the three resistors to the board first is the least challenging. They are also some of the most tolerant parts, so these grant you some experience to get you going.
2. 3-Pin headers - These components are not sensitive to the heat except for the plastic.
3. Capacitors - These are mounted next. Be sure to get them as close to the board as possible since having them stick up changes their intended impact on the circuit.


RST
A0
TX
RX
D0
D1
D5
D2
D6
D3
D7
D4
D8
GND
3V3
5V
C1
R1
R2
R3
C2
C3
ROOM
DAT VCC GND
VESSEL
DAT VCC GND


U1
Brew Bubbles
v1.0
Designed by @LBussy
www.brewbubbles.com
VESSEL
GND VCC DAT
ROOM
GND VCC DAT

4. 8-pin female headers - These are the tallest items on the front side of the board and are the last pieces to go on this side. Lightly tack on one pin and make sure the header is straight. When you have it positioned correctly, start from the other end, and solder the pins correctly. If you have a D1 laying around with the pin headers soldered on it already, using that to steady the parts helps. This process is a chicken or the egg choice with the next item. The first part to be soldered, either the controller or shield is the most difficult. After that, you can use the other to steady the headers of the first. If you have a breadboard, you may also employ that to steady the parts.
5. 8-pin male headers - These need to be soldered on the controller board. See note on #4 above.
6. GP1A57HR photo-interrupter - If the controller is still attached, take it off temporarily. The photo-interrupter goes on the *back* side of the circuit board in the outline provided. Therefore you solder it from the top side. If you put it on the wrong side, you can remove the solder (more YouTube work), but I'm not going to lie: it is frustrating. Be careful to do it right the first time.

Once you have finished soldering the shield, make sure to clean off the flux. You can use cheap vodka or Everclear, or a commercially available flux solvent.

It should be apparent by now that the Wemos should plug into the shield. There is a notch in the shield, which corresponds to the notch in the Wemos. The controller should be on the same side as the components as shown:

2.2.3 Sensors

Obtain some Dupont headers and a crimper from any of the usual places. Crimp a 3-pin female header on your sensors and plug them in. Some have asked about the pin order and the potential to change it to facilitate soldering a DS18B20 sensor directly to the PCB. That configuration was how I did my early prototypes, and it was a nicer form factor. However, I quickly found out that the ESP8266 would heat the surrounding parts, and the sensor read about 10°F higher than it should. Other projects get away with this, I believe, because they use sleep mode on the controller. As a web-delivered application, that was not an option for me.

2.2.4 Light Filter

These widely-used sensors are everywhere. Using them for this application sometimes takes a little tweaking to make it perfect. When developing Brew Bubbles, I got lucky and ended up with a couple of sensors that worked perfectly out of the box. After release, I discovered that most people were not so fortunate. Because the airlocks and water are clear, these other sensors failed to register bubbles correctly.

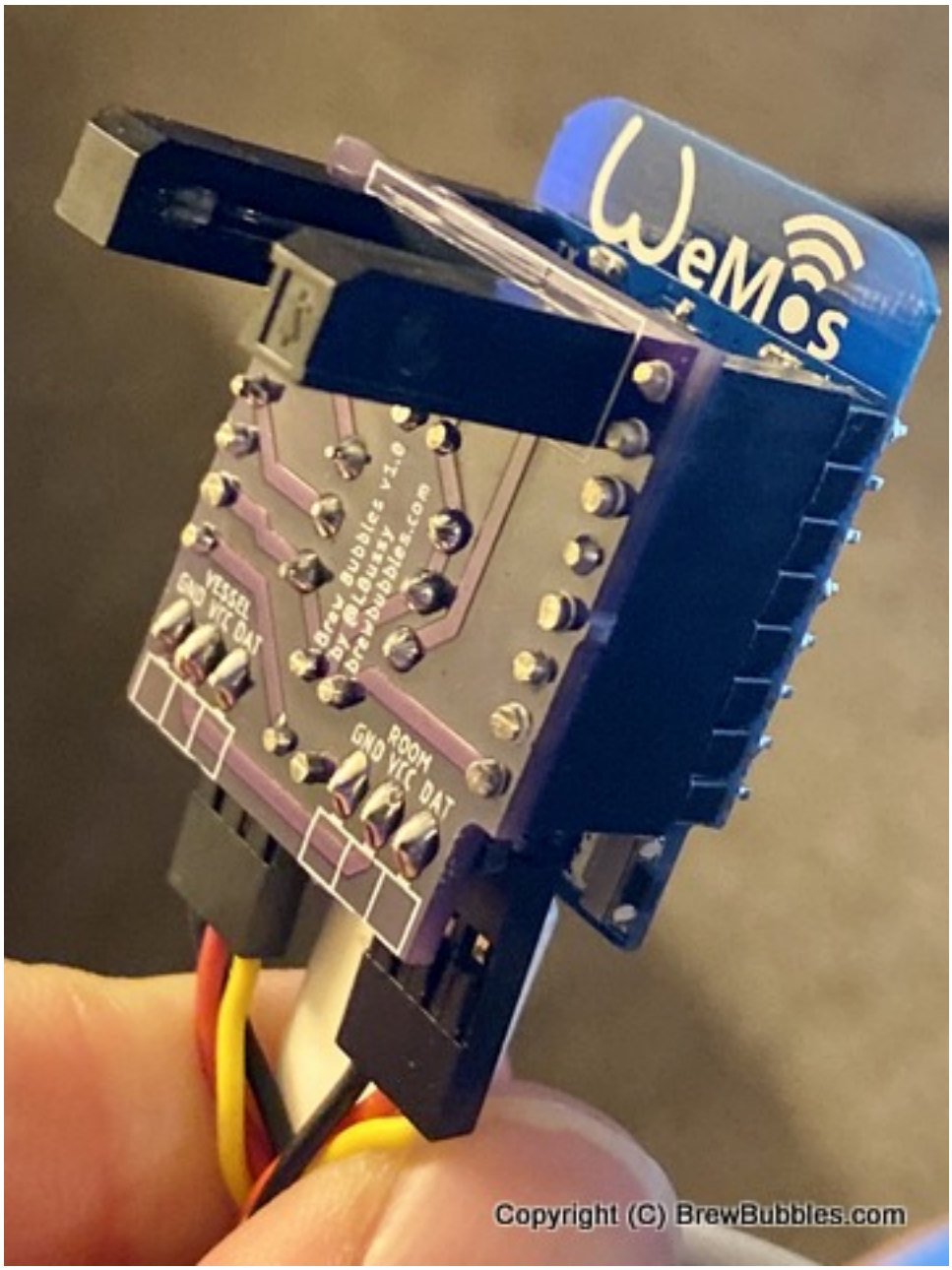
The Sharp GP1A57HRJ00Fit emits 950 nm light, and the receptor is sensitive to 400 through 1200 nm. That ranges from blue/violet through infrared. The sensor is most sensitive to IR light (700 nm – 1 mm) at 900 nm. Water absorbs infrared far better than the lower visible wavelengths:

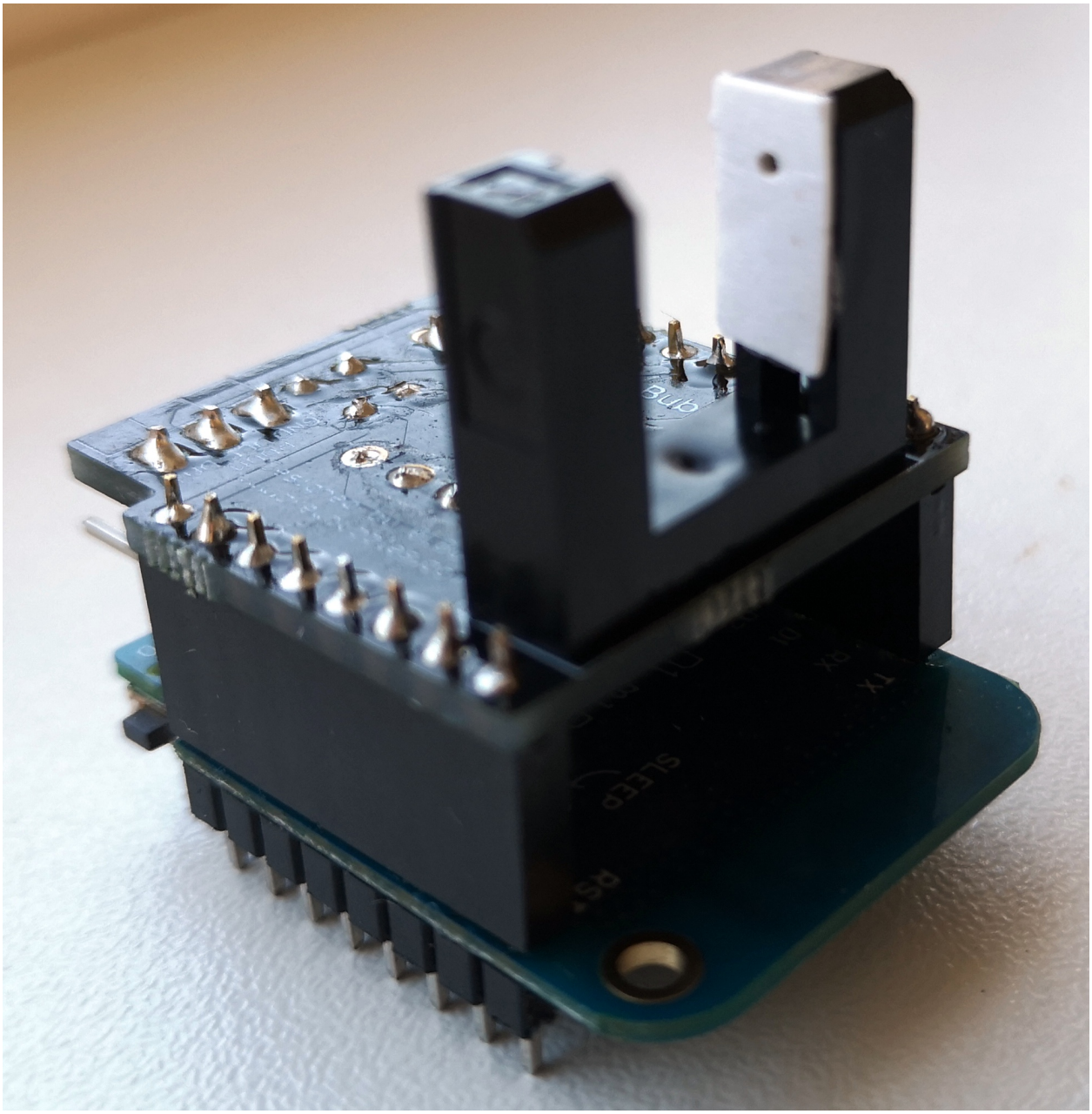
IR filters are standard in photography but possibly expensive to purchase. Unless you have one you can cannibalize, this approach might be pricey. What's not pricey are gels used for lighting in photography, stage, and film making. There are infrared gels, but these too are expensive and more difficult to find. What we used to do in the Army to make a flashlight for night-vision devices was to take our angle-head flashlights and put on the red and green filters. Red blocks all but red and green blocks all but green. The combination blocks pretty much all visible light but allows IR to pass.

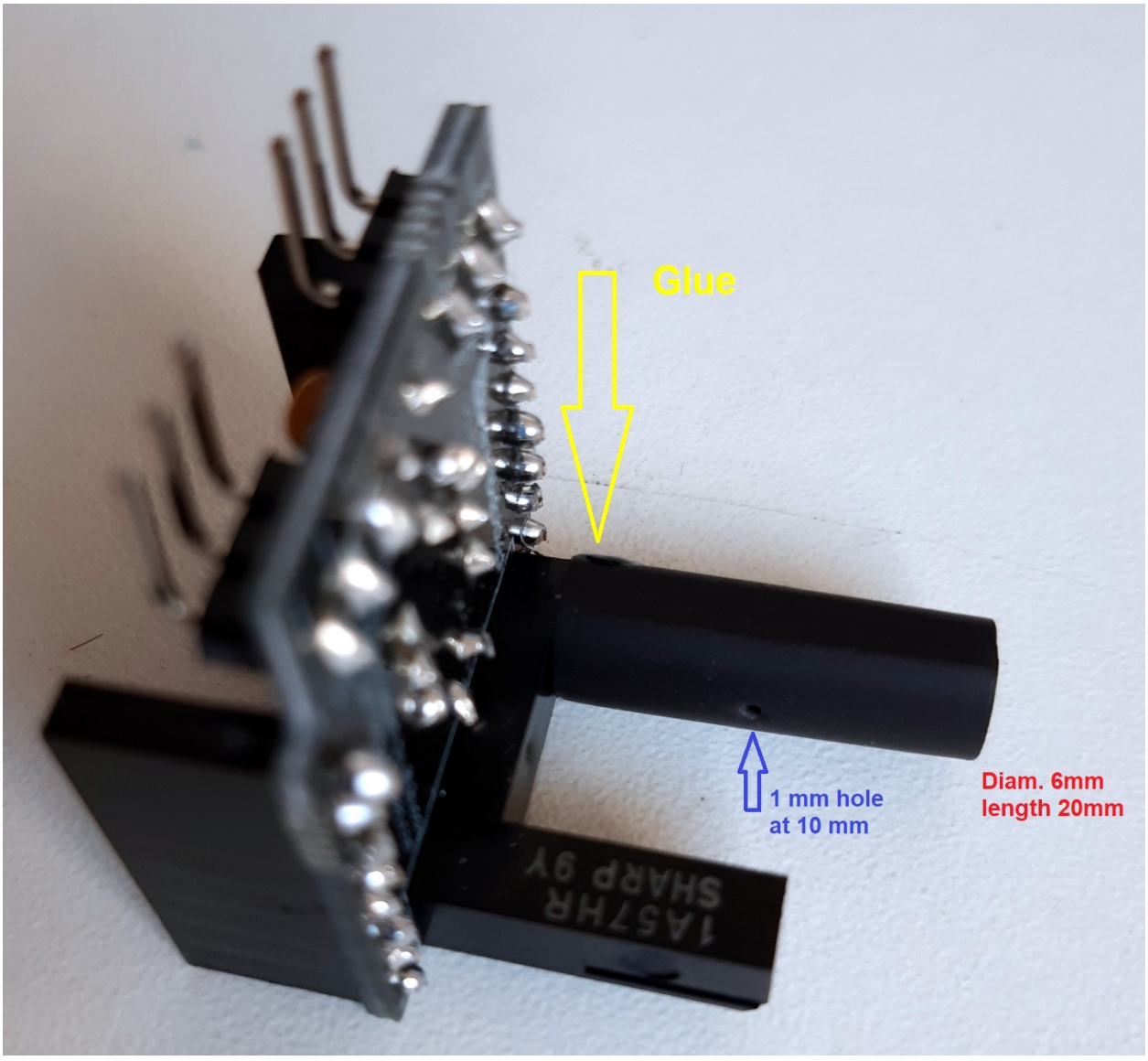
I tested this with gels I purchased online, which did not turn out to be as effective as I thought. I could also see through the gels when I looked around the room, so I suspect they were not optimal for this purpose. If you have an IR filter available, that might be a good thing to try. Otherwise, users @ChrisThomas and @wd16261 on Homebrewtalk came up with a very practical solution:

Here, ChrisThomas uses double-stick tape and a small piece of cardboard with a pinhole:

And here, wd16261 uses a piece of shrink-tubing with a dot of glue to hold his pinhole camera in place:


WeMos
Brew Bubbles v1.0
by @LaBussy
BrewBubbles.com
VESSEL
GND VCC DAT
ROOM
GND VCC DAT
Copyright (C) BrewBubbles.com




Glue
1 mm hole
at 10 mm
Diam. 6mm
length 20mm
1A57HR
SHARP 9Y

Both of these form collimators that restrict the amount of light entering the sensor. Both seem to be very effective and likely simpler than messing with light filters.

2.2.5 Bracket and Mounting

Position the photo-receptor such that the U-gap surrounds the bottom of a fermentation S-Lock. In this way, when bubbles pass by, they are registered and counted.



You should certainly feel free to use duct tape or a rubber-band or whatever suits you to affix Brew Bubbles to the airlock. I have included a 3-D printable bracket design in the project for those who desire a more finished approach. The completed controller & shield combo is slid into the top to allow the temperature sensor connectors to point up. The airlock is then passed through the hole in the bracket and into the carboy stopper. The hole may need to be adjusted larger or drilled out depending on the size of your airlock. If it is too loose around the airlock, a drill stop or even tape may be used on the tube under the bracket to hold it in place.

There is a hole in the bracket side intended to allow using a pop-rivet or small screw to secure the temperature sensor cable(s) with an R-type cable clamp. I recommend this to avoid strain on the small wires.

2.2.6 Power

Power the device via its USB port with a standard 5V cell phone type charger or power supply.


Copyright © Brewbubbles.com

2.3 Installing the Firmware

Contents

- *Installing the Firmware*
  - *Flashing Firmware - Initial*
    - *Preferred Method for Windows and Macs*
    - *Unix-Based Platforms (or Windows with Python)*
    - *Mac Platforms*
    - *Windows-Based Platforms*
  - *Erase Flash*
  - *Firmware Updates*

There are two critical files in this project:

File	Description	Address
firmware.bin	The Brew Bubbles application layer	0x00000
littlefs.bin	The Brew Bubbles web file system	0x300000

These files represent the program that you upload to your controller. The *firmware.bin* file is the C-based program that handles serving web pages, processing configurations, detecting pulses (bubbles), scheduling, and monitoring the temperature. The *littlefs.bin* file is how the controller's web file system is delivered. It is analogous to a directory on your computer. It contains web pages and images served via the web interface.

2.3.1 Flashing Firmware - Initial

Flashing the firmware may be done from many platforms with a variety of tools. The following are some methods that have I have tested. Other methods may work; however, I have no experience with them.

Important Note: The ESP8266 has a memory section which is not erased or written over by flashing firmware. If you have previously used your controller for any other tasks, I recommend that you wipe the flash memory before you begin. The method to do that differs by the tool used, and more information follows in the sections below.

Before proceeding, connect your controller via the USB port to your workstation.

Preferred Method for Windows and Macs

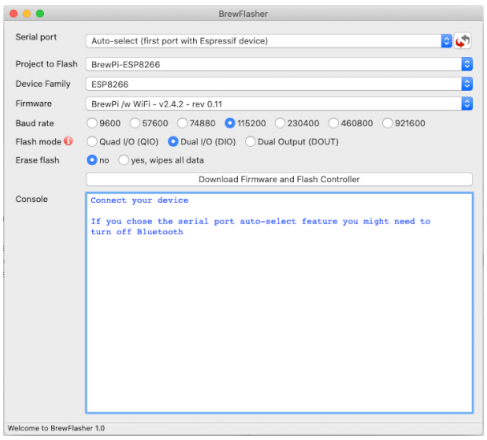
BrewFlasher is a stand-alone desktop application for Windows and macOS designed to simplify flashing Brew Bubbles (And other brewing-related firmware) to your controller.

It handles everything - locating the correct firmware, downloading it, setting the correct flash options/offsets, and flashing the firmware. There is no fumbling with the command line or worrying about esptool options. Select the project you want to flash, click a button, and you have finished.

You may download BrewFlasher from its GitHub release page or BrewFlasher.com.

You will use the following settings:

- Serial port: Auto-select



- Project: Brew Bubbles
- Device Family: ESP8266
- Firmware: Brew Bubbles (latest version)
- Baud rate: 921600 (any should work, this is faster)
- Flash mode: Dual I/O (DIO)
- Erase flash: yes, wipes all data (this avoids problems later on - it will wipe wifi settings if any)

Now click the button that says, "Download Firmware and Flash Controller."

Here is a short video produced by the author of BrewFlasher.

Unix-Based Platforms (or Windows with Python)

Espressif, the makers of the ESP8266, have adopted a python-based tool named esptool. Assuming you have either Python 2.7 or 3.4+ on your system, you can install *esptool* with *pip*:

```
pip install esptool
```

Note: With some Python installations, you may receive an error. Try *python -m pip install esptool* or *pip2 install esptool* (the latter, especially if you are on Python3).

After installing, esptool.py is available in the default Python executables directory. For manual installation instructions, please visit the GitHub repository.

Once *esptool* is installed, you may use the following command line to flash the firmware (assuming the firmware is in the local directory):

```
esptool write_flash 0x00000 firmware.bin 0x300000 littlefs.bin
```

Please note that this takes advantage of *esptool*'s capability to auto-detect the controller attached via USB. If you have other devices directly attached to your system, this may fail, and you will need to specify the port manually. For example: *-p /dev/ttyUSB0* (or *-p COM3* on Windows.)

If you desire to erase your controller, you may also use *esptool*'s erase flash option:

```
esptool erase_flash
```

Mac Platforms

I do not have access to a Mac. However, anecdotally, I believe Mac users may follow the "*Unix-Based Platforms*" instructions above. If you have firsthand knowledge of this process, please let me know.

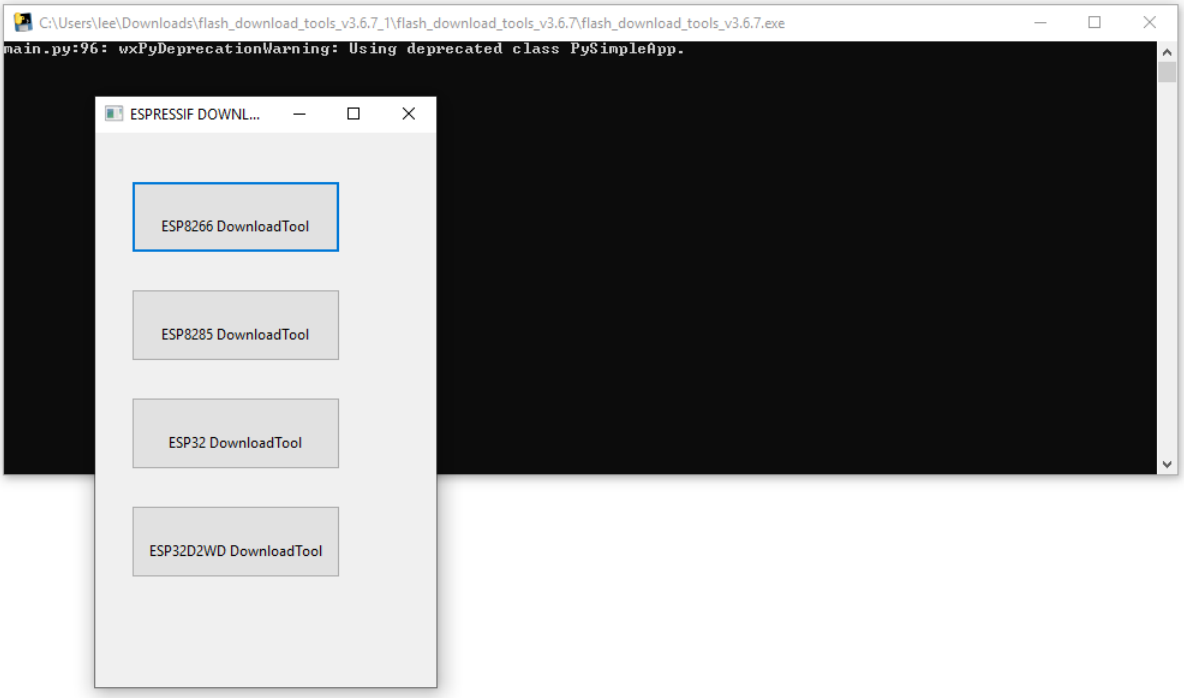
Windows-Based Platforms

There are two methods for uploading the firmware files to your controller. The preferred method is BrewFlasher; the controller vendor supplies another available way:

Espressif Tools

Espressif's Flash Download tool is the tool that the makers of the ESP8266 have released. Download it directly from Espressif's website. Unzip the tool to a convenient folder and execute the application (named *flash_download_tools_v3.6.7.exe* at the time of writing.)

Here you see the console window and the main screen:



Select "ESP8266 DownloadTool." Setup as follows:

- Add the firmware file - Check the first checkbox - Click the ellipsis (. . .) next to the text field - Navigate to the firmware directory, select *firmware.bin* and click "*Open*" - In the right-most text field after the "@" symbol, enter the address *0x00000* (zero, the lower-case letter "X", followed by five zeros)
- Add the SPIFFS (LittleFS) file - Check the second checkbox - Click the ellipsis (. . .) next to the text field - Navigate to the firmware directory, select *littlefs.bin* and click "*Open*" - In the right-most text field after the "@"* symbol, enter the address *0x300000* (zero, the lower-case letter "X", followed by the number "3" and five zeros)
- Set the CrystalFreq to *26M*

- Set the SPI Speed to *40MHz*
- Set the SPI MODE to *QIO* (you may use *DIO* if you experience issues flashing the firmware)
- Set the FLASH SIZE to *32Mbit-C1* (32 Megabits = 4 Megabytes)
- Select the proper COM port
- Set BAUD to *460800* (you may use a lower speed if you experience issues flashing the firmware)

When setup is complete, click on the "*START*" button underneath the green box. The darker green box will move across the bottom of the window, and when complete, the bright green box changes to "*FINISH*".

At this point, you may close the tool and the selection screen and proceed with setup.

2.3.2 Erase Flash

If you desire to erase your controller, you may leverage the "*ERASE*" button within the Flash Download tool.

2.3.3 Firmware Updates

The web application provides Over The Air (OTA) update functionality for upgrades. Navigate to the *Settings* page and scroll down to the *Update Firmware* section.

2.4 Set Up Networking

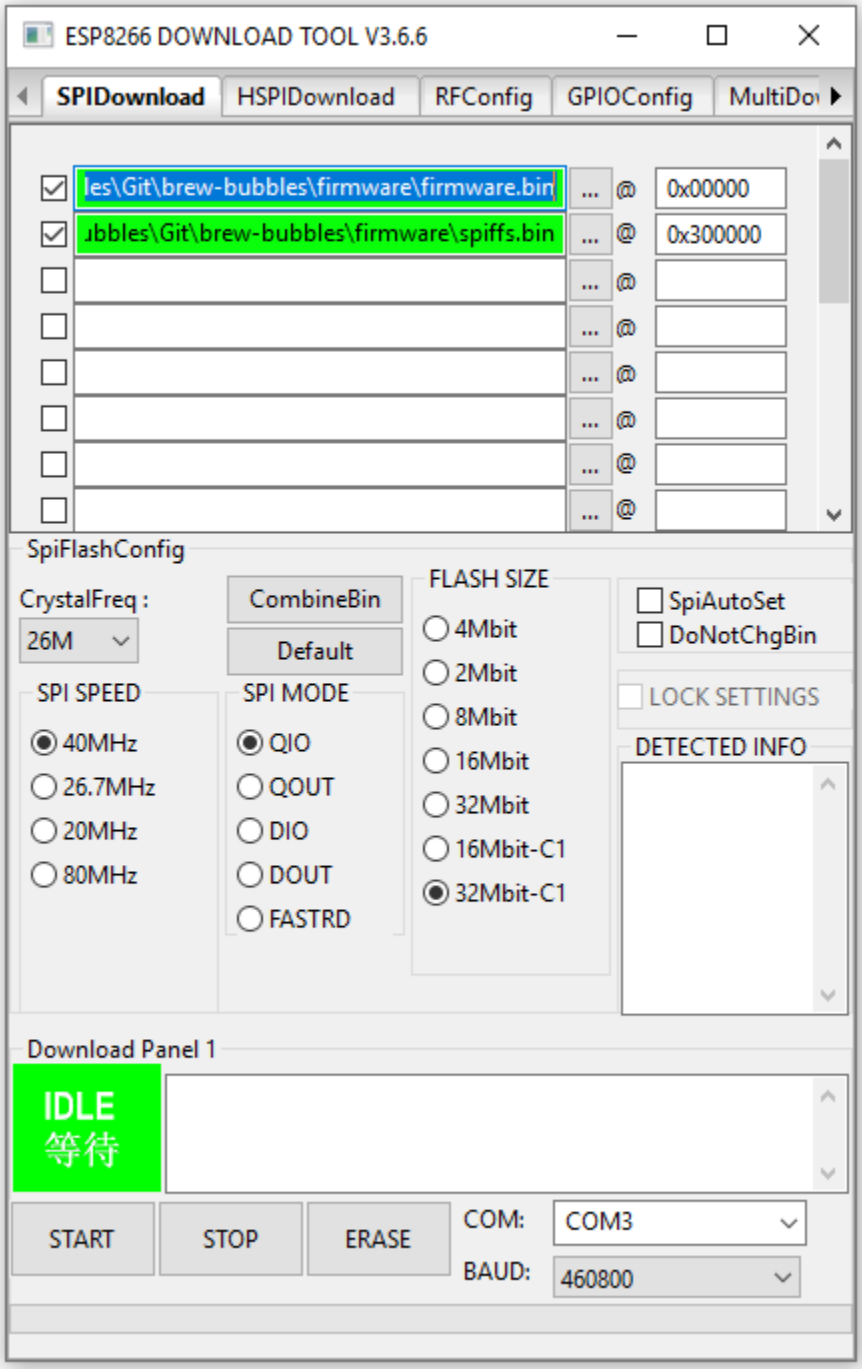
Once you have flashed the firmware and littlefs files to your controller, it starts in Access Point mode with a captive portal. A captive portal is similar to what you get when you connect to free WiFi with a login web page. Any network communication will be redirected to the portal page, allowing you to log in. This captive portal is how you initially configure Brew Bubbles, allowing it to connect to your local WiFi.

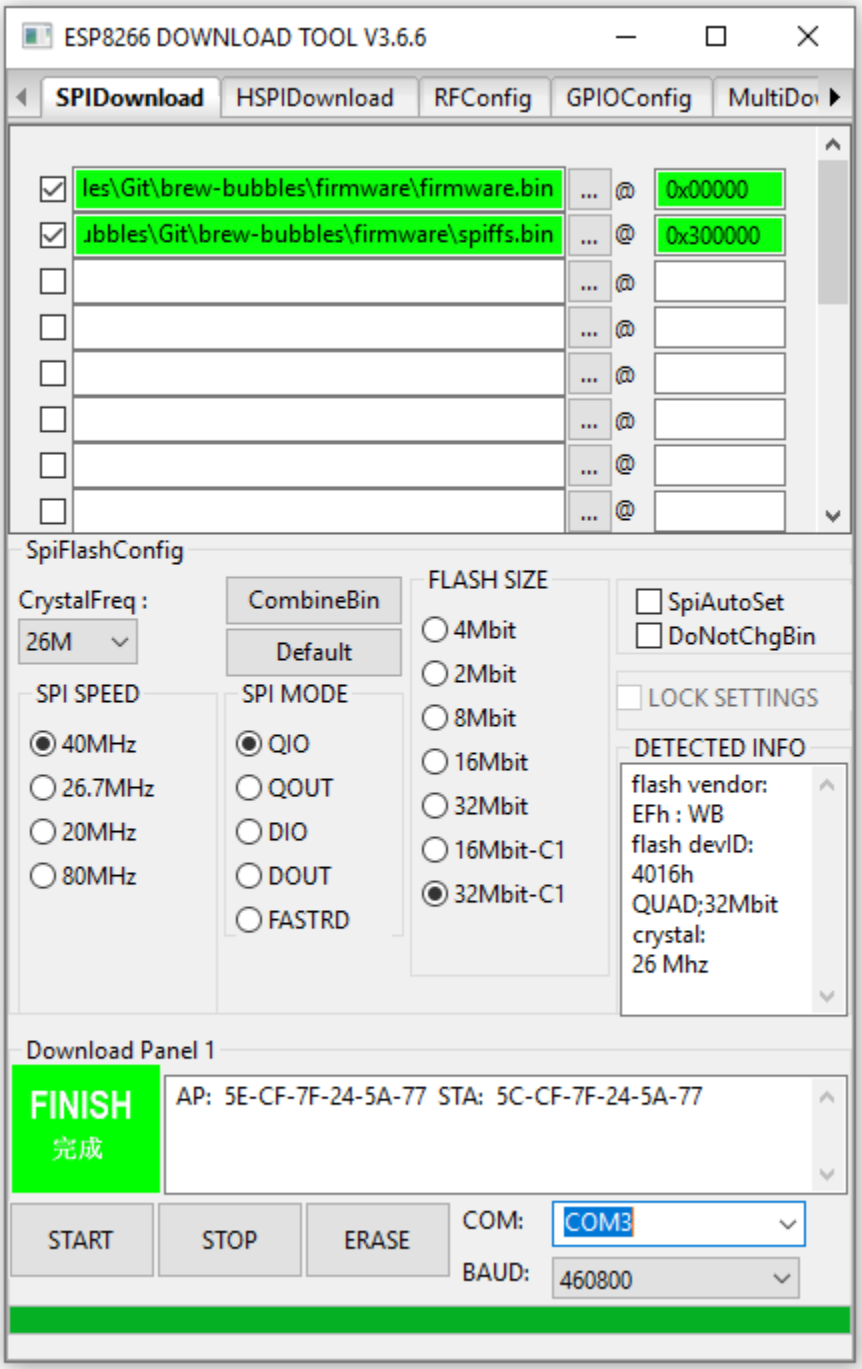
This process works best through a phone in most cases. There are some peculiarities in the ESP8266 libraries, which sometimes cause the process to act a little flakey. It may not open the portal page or not issue an IP address to your client. In testing, these problems did not come up using a phone. If you don't have a phone handy, it is possible to do this work with your computer. However, I point out some caveats below.

Blink Mode: The LED on the ESP8266 is an indicator of various modes during operation. When in AP mode, the LED blinks on and off at 0.5Hz. That is, it is on for a second and off for a second. When in this mode, the access point shows up in the list of available access points on your client.

Contents

- *Set Up Networking*
  - *Connect to Captive Portal*
    - *Captive Portal Page does not Open*
    - *Connection Timed Out or Similar Errors*
  - *Portal Configuration*


ESP8266 DOWNLOAD TOOL V3.6.6
SPIDownload
HSPIDownload
RFConfig
GPIOConfig
les\Git\brew-bubbles\firmware\firmware.bin
0x00000
ubbles\Git\brew-bubbles\firmware\spiffs.bin
0x300000
SpiFlashConfig
CrystalFreq :
26M
CombineBin
Default
FLASH SIZE
4Mbit
2Mbit
8Mbit
16Mbit
32Mbit
16Mbit-C1
32Mbit-C1
SpiAutoSet
DoNotChgBin
LOCK SETTINGS
DETECTED INFO
SPI SPEED
40MHz
26.7MHz
20MHz
80MHz
SPI MODE
QIO
QOUT
DIO
DOUT
FASTRD
Download Panel 1
IDLE
等待
START
STOP
ERASE
COM:
COM3
BAUD:
460800


ESP8266 DOWNLOAD TOOL V3.6.6
SPIDownload
HSPIDownload
RFConfig
GPIOConfig
les\Git\brew-bubbles\firmware\firmware.bin
0x00000
ubbles\Git\brew-bubbles\firmware\spiffs.bin
0x300000
SpiFlashConfig
CrystalFreq :
26M
CombineBin
Default
SPI SPEED
40MHz
26.7MHz
20MHz
80MHz
SPI MODE
QIO
QOUT
DIO
DOUT
FASTRD
FLASH SIZE
4Mbit
2Mbit
8Mbit
16Mbit
32Mbit
16Mbit-C1
32Mbit-C1
SpiAutoSet
DoNotChgBin
LOCK SETTINGS
DETECTED INFO
flash vendor:
EFh : WB
flash devID:
4016h
QUAD;32Mbit
crystal:
26 Mhz
Download Panel 1
FINISH
完成
AP: 5E-CF-7F-24-5A-77 STA: 5C-CF-7F-24-5A-77
START
STOP
ERASE
COM:
COM3
BAUD:
460800

2.4.1 Connect to Captive Portal

Connect to the access point as follows:

Login: Access Point Name (SSID): *brewbubbles*

Password: *brewbubbles*

See the documentation for your particular phone, device, or computer OS if you don't know how to access this list.

Select the Brew Bubbles access point and enter the password: "brewbubbles" (without the quotes:)

Once you are connected, depending on the platform you are working on, a web page should open, displaying the portal. With iOS, this should happen automatically. While testing on various PCs, this was not 100% successful. I noted two issues:

Captive Portal Page does not Open

1. If your default web browser is not opening the portal page automatically, open the browser of your choice. It is likely that, at this point, the captive portal displays correctly.
2. If the portal does not open automatically, a bar may appear at the top of your browser with a button indicating that the network login page needs to open. Clicking this button should open the portal.
3. If the portal still does not open, enter the address "192.168.4.1" (without quotes) in the address bar. The Brew Bubbles controller uses this address in portal mode.

Connection Timed Out or Similar Errors

If you cannot open the captive portal web page by following the above instructions, the controller may not have issued your device an IP address. Follow instructions for your particular platform to set the following parameters:

IP Configuration: IP Address: 192.168.4.2

Subnet Mask: 255.255.255.0

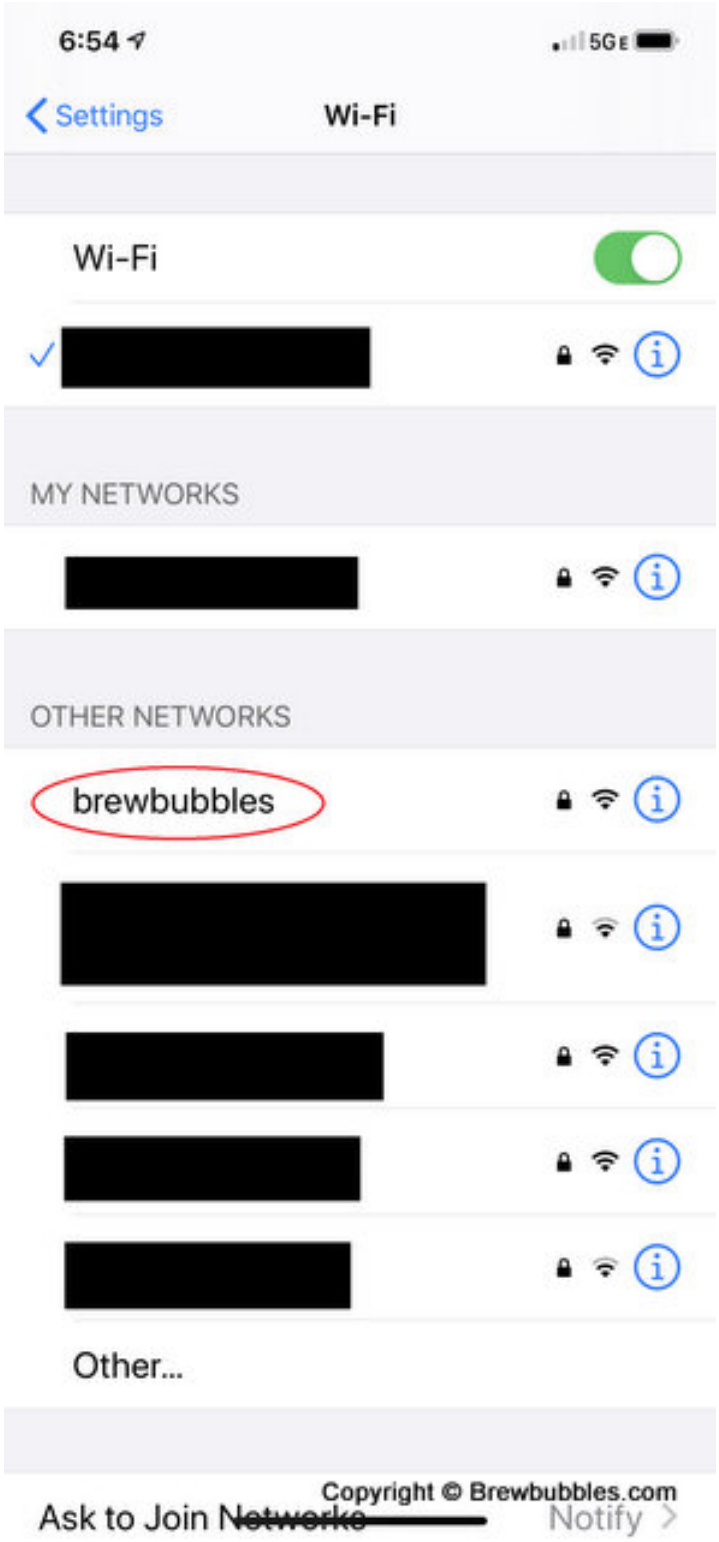
Gateway: 192.168.4.1

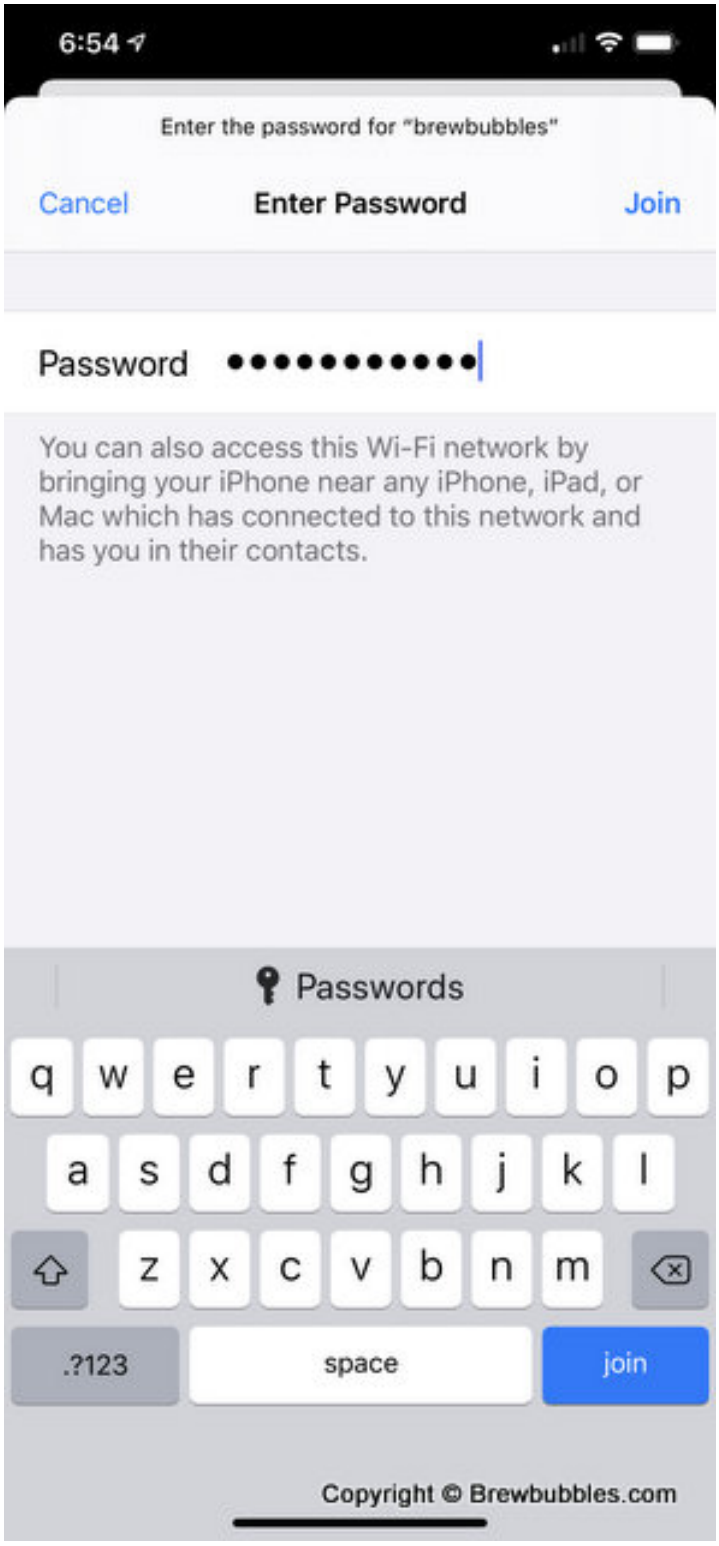
These methods should allow you to configure the portal. If you still cannot connect to the portal, please try another system or method before logging an issue.

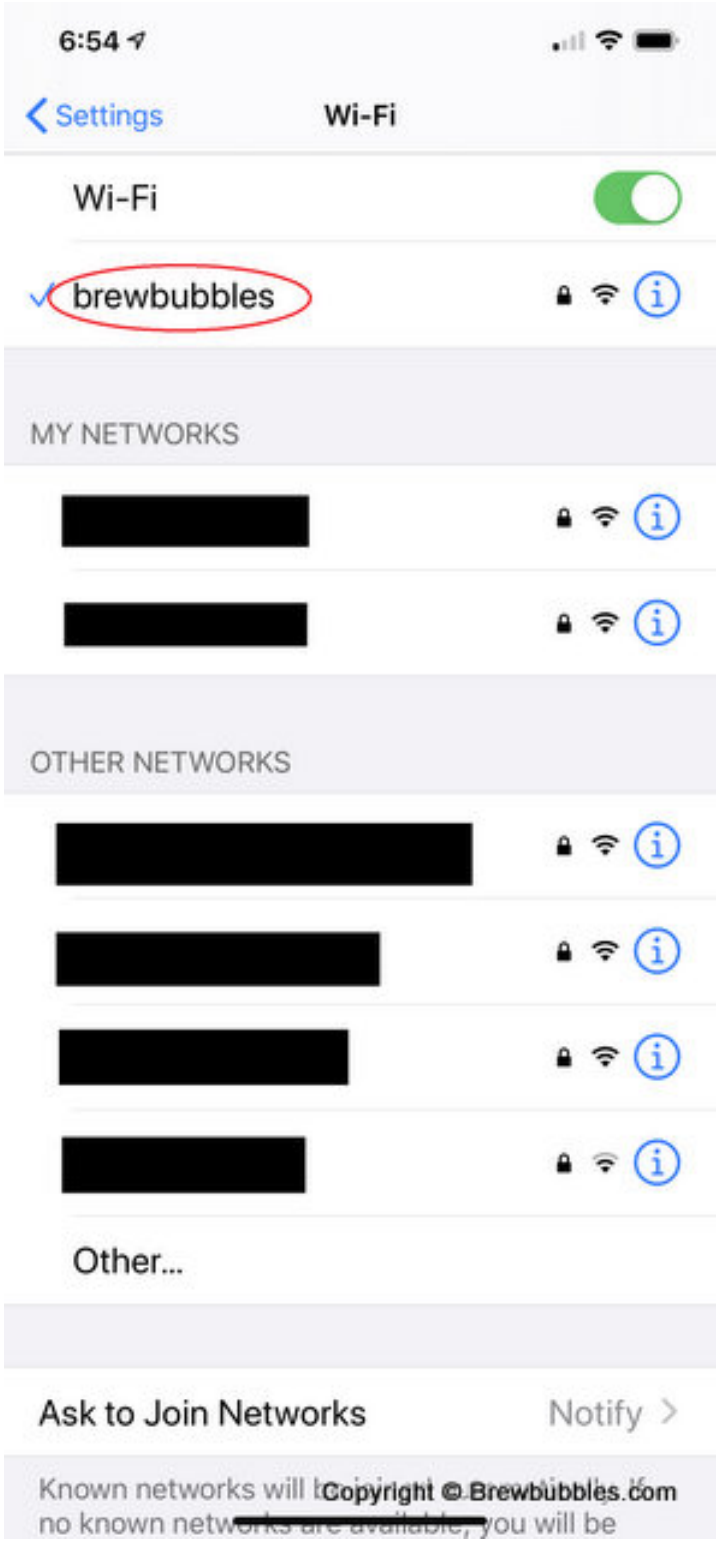
2.4.2 Portal Configuration

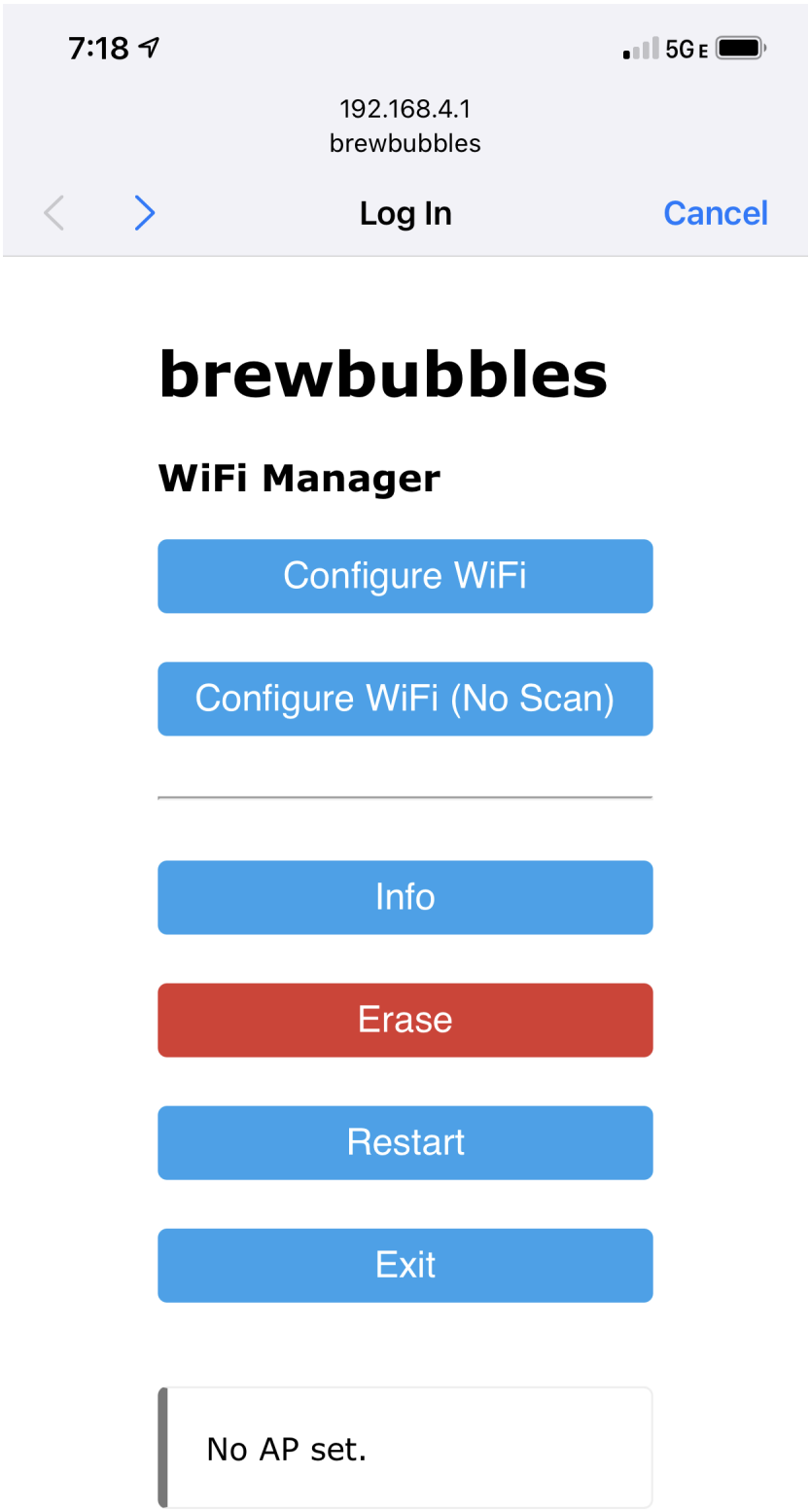
When you access the captive portal, you have six choices:

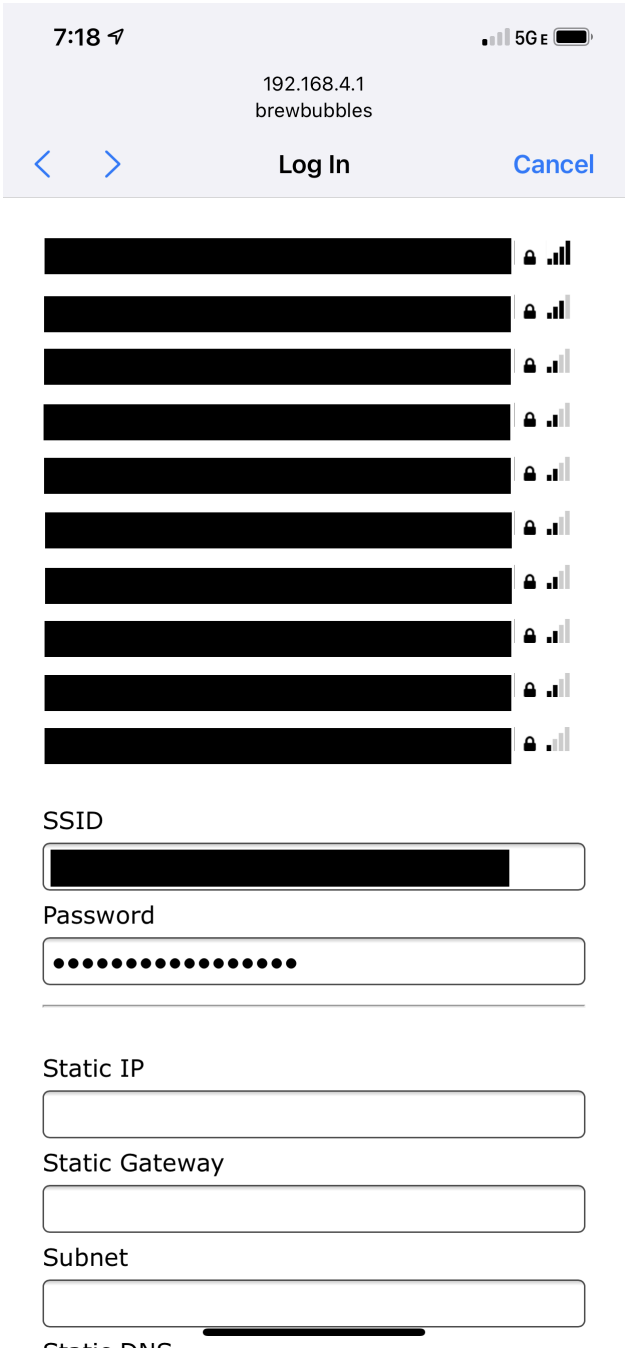
1. Configure WiFi
2. Configure WiFi (No Scan)
3. Info
4. Erase
5. Restart
6. Exit


6:54
5G
Settings
Wi-Fi
Wi-Fi
MY NETWORKS
OTHER NETWORKS
brewbubbles
Other...
Ask to Join Networks
Notify
Copyright © Brewbubbles.com


6:54
Enter the password for "brewbubbles"
Cancel
Enter Password
Join
Password
You can also access this Wi-Fi network by bringing your iPhone near any iPhone, iPad, or Mac which has connected to this network and has you in their contacts.
Passwords
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
.?123
space
join
Copyright © Brewbubbles.com


6:54
Settings
Wi-Fi
Wi-Fi
brewbubbles
MY NETWORKS
[redacted]
[redacted]
OTHER NETWORKS
[redacted]
[redacted]
[redacted]
[redacted]
Other...
Ask to Join Networks
Notify
Known networks will
no known networks are available, you will be
Copyright © Brewbubbles.com


7:18
5GE
192.168.4.1
brewbubbles
Log In
Cancel
brewbubbles
WiFi Manager
Configure WiFi
Configure WiFi (No Scan)
Info
Erase
Restart
Exit
No AP set.


7:18
5G E
192.168.4.1
brewbubbles
Log In
Cancel
[redacted]
SSID
[redacted]
Password
••••••••••••••••••
Static IP
Static Gateway
Subnet

Select option 1, “Configure WiFi.” You see the following configuration screen:

Your access point should be visible in the list on top. Selecting it populates the name in the SSID field below. If your access point does not show up, you can re-scan or enter the AP name manually. Note this name is case-sensitive. Next, enter your access point password in the “Password” field.

If you need to use a static IP address, you must fill out the following fields:

1. Static IP
2. Static Gateway
3. Subnet
4. Static DNS

If you have questions about these fields, consult the documentation for your access point. You need not fill out these fields to use an automatically assigned IP address since you may access the device by its name once connected to WiFi.

You may also change the device name, which is required if you have more than one device on the network. No two devices should have the same name.

Once you have filled out at least the SSID and Password, click on the “Save” button.

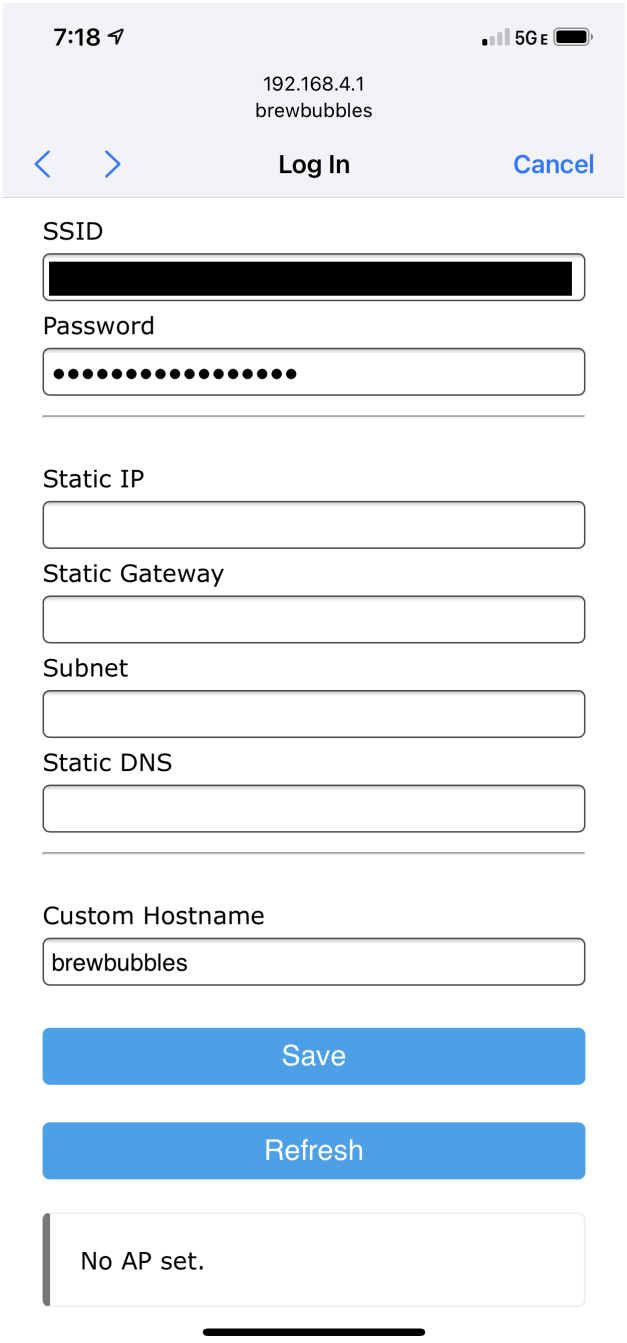
The controller will now connect to the wireless access point you have configured. On most phones, the configuration page will close automatically.

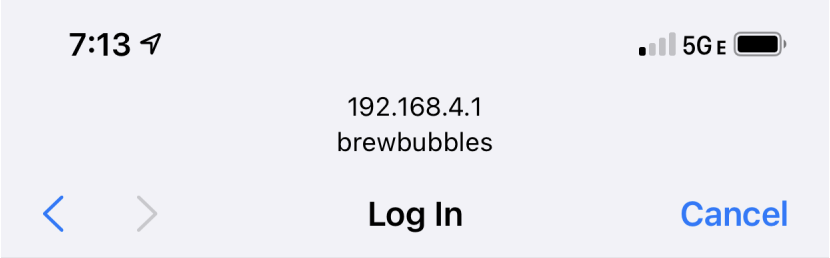
2.5 Operation and Configuration

Brew Bubbles is presented to the user for monitoring and configuration as a set of web pages. Below are the different information and settings available.

Contents

- *Operation and Configuration*
  - *mDNS Support for Client OS*
  - *Menu*
  - *Home Page*
  - *Settings Page*
    - *Controller Settings*
    - *Temperature Settings*
    - *URL Target Settings*
    - *Brewer’s Friend Settings*
    - *Brewfather Settings*
    - *ThingSpeak Settings*
    - *Advanced*


7:18
5G E
192.168.4.1
brewbubbles
Log In
Cancel
SSID
[redacted]
Password
••••••••••••••••••
Static IP
Static Gateway
Subnet
Static DNS
Custom Hostname
brewbubbles
Save
Refresh
No AP set.



Saving Credentials
Trying to connect ESP to network.
If it fails reconnect to AP to try again.

2.5.1 mDNS Support for Client OS

The Brew Bubbles device leverages a multicast Domain Naming System (mDNS) to make it easier to connect to your device. You may also have heard of zeroconf, which includes mDNS; or Avahi, which is a different implementation of mDNS. For our purposes, all of them work together.

Since I designed Brew Bubbles to be accessed and configured via a web page, you need to know its address. Your local WiFi automatically assigns an IP address if you did not enter a static address in the WiFi configuration. You can always type in something like *192.168.4.100*, but that's not as easy to remember as *brewbubbles.local.*

macOS and Linux (including Raspberry Pi's) have implementations of mDNS, which will natively allow them to use the *.local name. Windows requires that you install a small program to support mDNS. Review the table below for OS support of mDNS:

Operating System	mDNS Support
MacOS	Built-in
iOS	Built-in
Linux	Built-in most versions; If not, install Avahi
Android	Not natively supported. No Android avail for testing.
Windows	Not natively supported. Install Bonjour from Apple

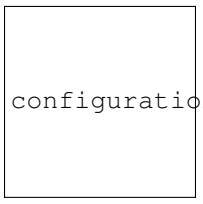
You can undoubtedly work with Brew Bubbles without mDNS. However, it makes your life easier. If you do not implement mDNS, of course, you either need to use an OS hosts file or bookmark (or remember) the IP address.

You may access Brew Bubbles by entering the name *brewbubbles.local* (or the IP address) in your browser.

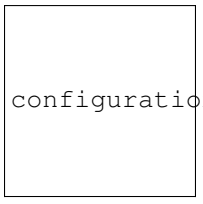
If you have an issue accessing a *.local address, it may be that your router does not support multicasting. Very old routers and some routers which are supplied by your cable company may be in this category. If this is the case, you need to use the IP address.

2.5.2 Menu

Across the top of every page, a menu displays. The standard desktop page looks like the below:



The mobile header looks like the below:



Clicking on the three vertically stacked lines, called a "triple bar," activates the main menu:

The page layout is responsive; the web browser's display ratio determines the header type. If you change your browser window layout to be more narrow on your desktop, it displays the mobile header.



Either across the top in desktop mode or as a dropdown when you click the ≡ character, you see the following choices:

Home The main page, with the current values displayed.

Settings Configuration and maintenance choices.

Help Where to get help.

About Information about the author.

You may review each of these below.

2.5.3 Home Page

Accessing the main page of Brew Bubbles provides you with ready access to all functional monitoring points:

You will see the following items:

Vessel Name Vessel Name is a label that you may assign to help you keep track of multiple devices. It defaults to "Fermenter 1," but you may change it in the settings.

Bubbles per Minute Brew Bubbles internally polls the device for approximately one minute. It then reports the bubbles per minute in exact terms, meaning the number may be a decimal. Brew Bubbles also uses a sliding window to average the readings to help filter noise. The sliding window is set at 15, meaning as the device is in operation, it reports Bubbles per Minute as an average of up to the last 15 readings. In effect, this is a 15-minute moving average. This window is not configurable via the interface.

Ambient Temp If you have an ambient (room) temperature sensor installed, this reports the temperature in the configured temperature format (default is Fahrenheit.) This temperature reports in a 5-minute sliding window. This window is not configurable via the interface.

Vessel Temp If you have a vessel temperature sensor installed, this reports the temperature in the configured temperature format (default is Fahrenheit.) This temperature reports in a 5-minute sliding window. This window is not configurable via the interface.



Bubble Sensor

Current Readings

Vessel Name:	Fermenter 1
Bubbles Per Minute:	0.0
Ambient Temp:	64.85°F
Vessel Temp:	64.33°F
Last Reading:	Sat Dec 14 2019 20:33:00 GMT-0600 (CST)
Refresh in:	51

Last Reading The date and time of the most recently calculated reading set within the controller. Internally the device refreshes its values approximately every 60 seconds.)

Refresh In The web page refreshes its displayed values every 60 seconds. This field shows the time remaining until that refresh.

2.5.4 Settings Page

The settings page contains all configurable items for configuration and control of Brew Bubbles.

Note: Each setting page as an "Update" button. Be sure to save any updates before leaving a page. There will be no reminder if you selct another link without saving.

Controller Settings

The first settings frame is the Controller Settings. This section deals with the overall device configuration.

You may configure two items here:

mDNS ID: The default mDNS name for Brew Bubbles is *brewbubbles.* This name forms the name portion of the mDNS name *brewbubbles.local.* The mDNS name needs to be unique on the local network. If you have more than one Brew Bubbles device, you should change these names to be unique. Should you forget and have two with the same name, you must access the controller via the IP address to change the name.

The name should be 3 to 24 characters in length, begin with a letter, and contain only ASCII letters 'a' through 'z' (case-insensitive), the digits '0' through '9', and the hyphen-minus character ('-'). Do not include the *.local* portion of the mDNS name.

Bubble ID: Bubble ID is an additional field that can help distinguish between different Brew Bubbles devices reporting to a shared system.

Temperature Settings

Configure temperature format and calibration in this section:

Temperature Format: Select either Fahrenheit or Celsius with the radio button. Conversion happens internal to the controller and reports in the proper format.

Temperature Calibration: In this section, you may enter calibration offsets to either sensor independently. Enter any decimal-based number from -25.0 to 25.0 and click "Update." The compensation applies internally, and the corrected temperatures are displayed.

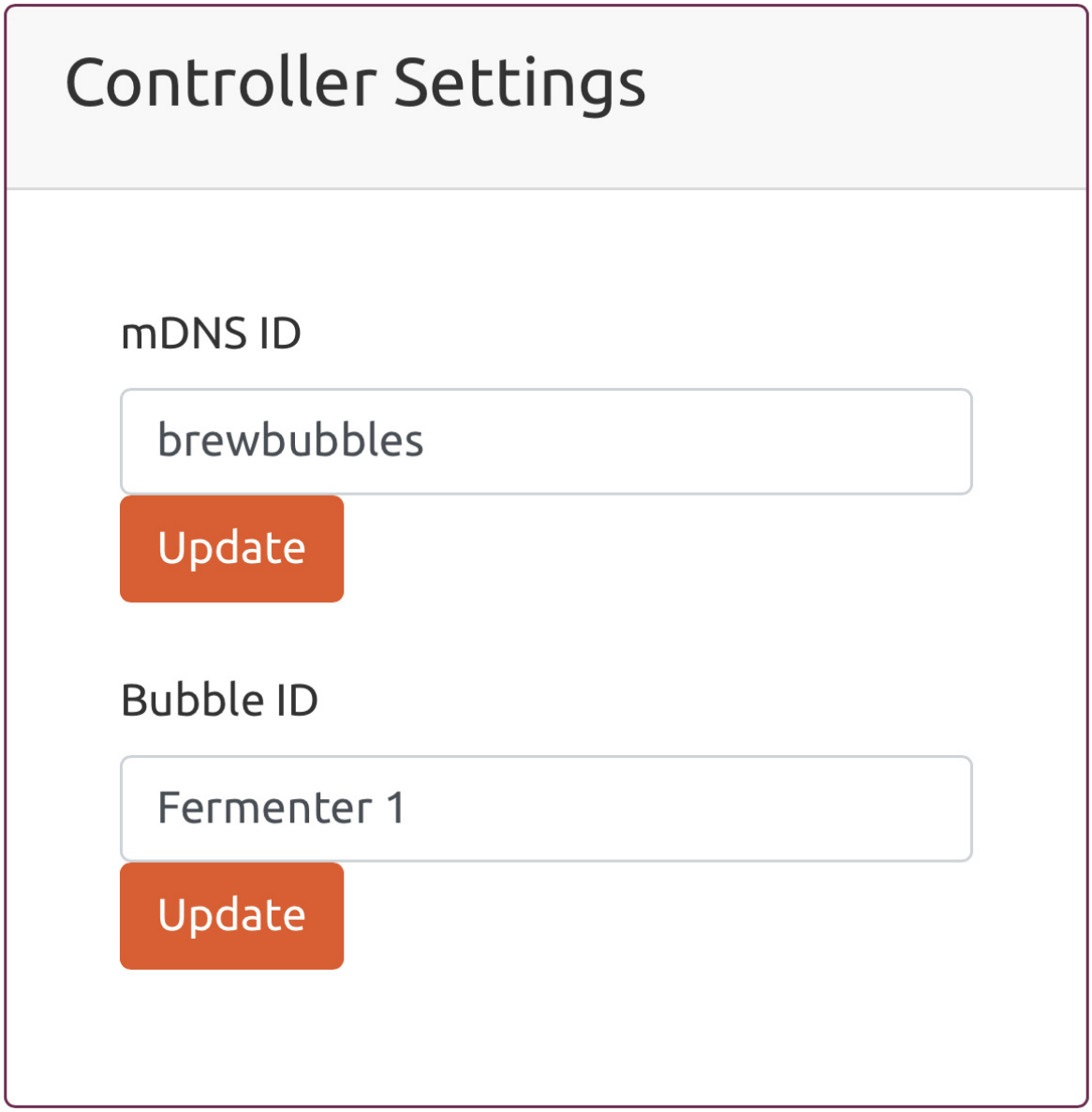
URL Target Settings

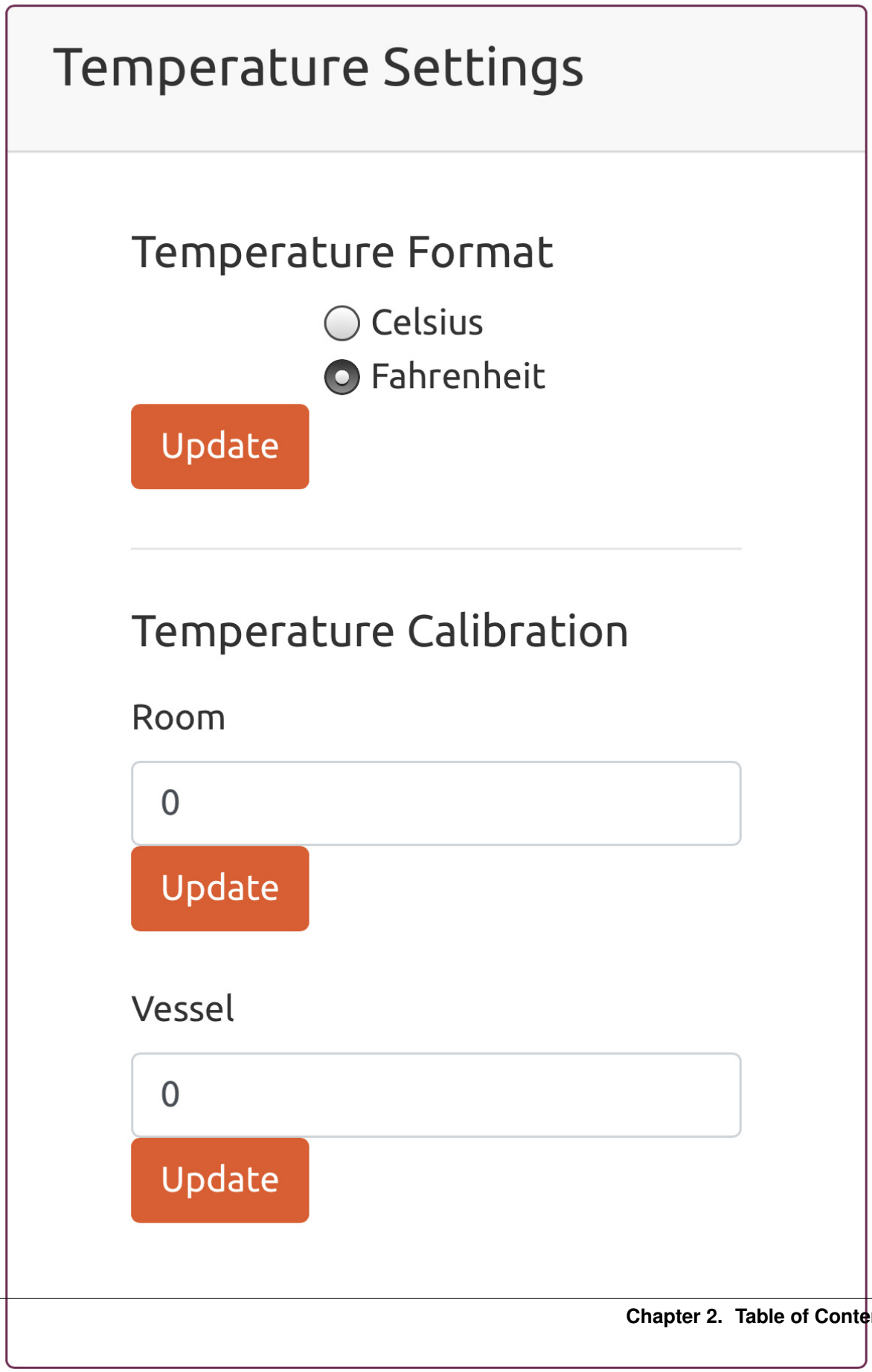
Target settings control how Brew Bubbles reports to HTML endpoints such as BrewPi Remix or Fermentrack. BrewPi Remix automatically begins to report on Brew Bubbles' data once received at its endpoint.

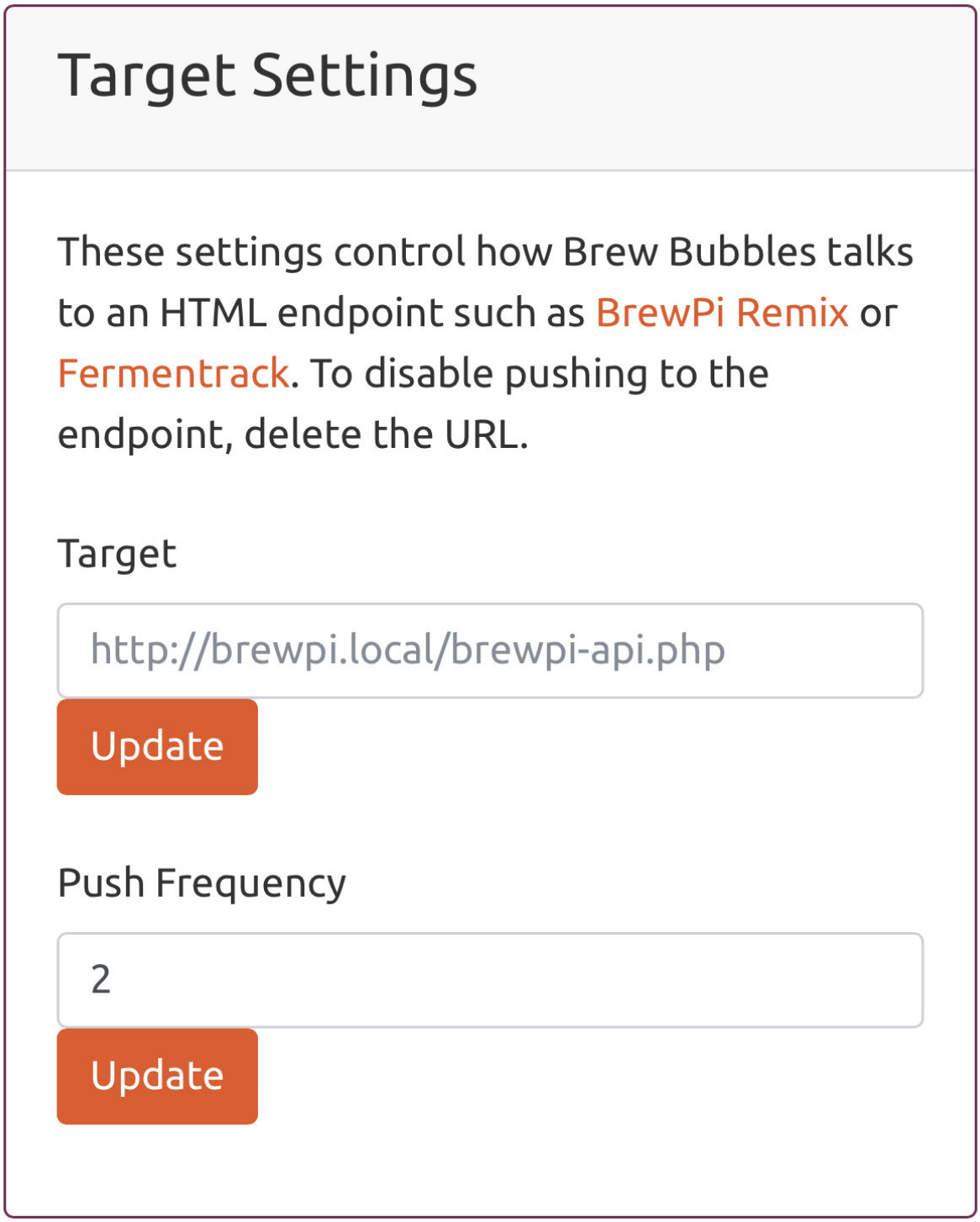
Target: The target may be any DNS or mDNS name. If you are using mDNS, be sure to include the ".local" portion. The address should be a complete URI, including the target page and port if needed. For BrewPi Remix, the name will be *http://{hostname}.local/brewpi-api.php*.

If you are unable to access Brew Bubbles using the *.local name, you are not able to use a target with a .local name either. In this case, use the IP address of your target.

Only HTTP (not HTTPS) is supported. Support for SSL on controllers is extremely resource-intensive as well as unstable at this time. If the libraries improve in the future, I will consider https support.


Controller Settings
mDNS ID
brewbubbles
Update
Bubble ID
Fermenter 1
Update


Temperature Settings
Temperature Format
Celsius
Fahrenheit
Update
Temperature Calibration
Room
0
Update
Vessel
0
Update


Target Settings
These settings control how Brew Bubbles talks to an HTML endpoint such as BrewPi Remix or Fermentrack. To disable pushing to the endpoint, delete the URL.
Target
http://brewpi.local/brewpi-api.php
Update
Push Frequency
2
Update

If a port number is required, it comes immediately after the hostname. Basic URI rules are:

```
http:[//authority]/path[?query][#fragment]
```

Authority is made up of:

```
authority = [userinfo@]host[:port]
```

For more information, please review the Wikipedia article.

Push Frequency: Enter the push frequency in minutes. Be sure to check your target system's requirements and restrictions so that you do not flood the target. For BrewPi Remix, I recommend setting it at 2 minutes, which matches the default charting granularity. Valid settings for this field are 1 to 60 minutes.

Brewer's Friend Settings

The friendly folks at Brewer's Friend have added "BPM" (Bubbles per Minute) to their API. Adding Brew Bubbles to your Fermentation Chart is done on the Fermentation Chart page for your brew under "Devices." Select "Link Devices," choose a "Stream" device, and select your Bubble ID. The device must have reported to Brewer's Friend at least once to be listed.

Brewer's Friend Key: Find your API key from your Profile dropdown in the top-right corner of the web page under "Integrations." Towards the top of the page is a section labeled "API Key." Copy the API Key and enter it into this section and click "Update." The key is a long hexadecimal key which will look like *c6e88f70f575c4ecdca3dcb686381185*.

Push Frequency: Enter the push frequency in minutes. Brewer's Friend requires that you push readings no more than once every 15 minutes. Valid settings for this field are 15 to 120 minutes.

Brewfather Settings

Brewfather integration is also supported. Adding Brew Bubbles to your Fermentation Chart is done in Settings where you will enable "Custom Stream." The device must have reported to Brewer's Friend at least once to be listed.

Brewfather Key: Log into your Brewfather account and go to Settings > Custom Stream. Your API key will be a 10 to 64-character string of letters and numbers on the line following "URL http://log.brewfather.net/stream?id=" (do not include the the URL.) e.g. you may see: http://log.brewfather.net/stream?id=q4F3wPfooBa3X3, from which you will enter *q4F3wPfooBa3X3* as your key.

Push Frequency: Enter the push frequency in minutes. Brewfather requires that you push readings no more than once every 15 minutes. Valid settings for this field are 15 to 120 minutes.

ThingSpeak Settings

ThingSpeak allows posting custom data streams in order to collect and report upon it. To enable this functionality, you must create a channel with the following:

- **Name (optional):** Any you prefer, such as "Brew Bubbles | Fermenter 1"
- **Description (optional):** How you would like to present this, such as "Brew Bubbles data channel for Fermenter 1."
- **Field 1:** "BPM" and check enabled
- **Field 2:** "Ambient °F" (or "Ambient °C") and check the box to enable
- **Field 3:** "Vessel °F" (or "Vessel °C") and check the box to enable

Brewer's Friend Settings

These settings control how Brew Bubbles talks to Brewer's Friend. With a premium account, Brewer's Friend supports logging brew-related data. To find your API key; log into your Brewer's Friend account and go to Profile > Account. Your API key will be a 40-character string of letters and numbers (not a URL).

Brewer's Friend Key

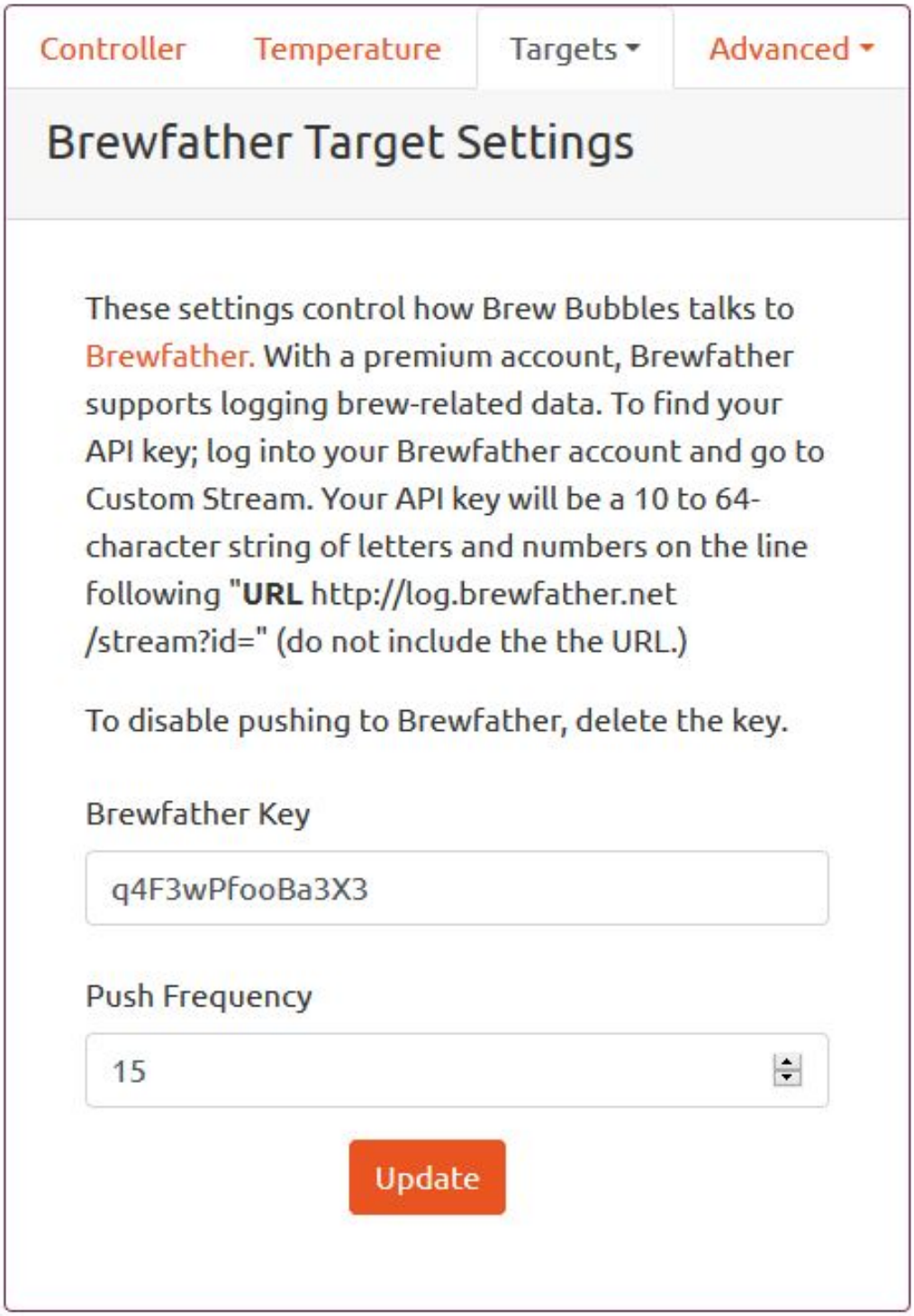
000fa2ac454c1abf14ffa3

Update

Push Frequency

15

Update


Controller
Temperature
Targets
Advanced
Brewfather Target Settings
These settings control how Brew Bubbles talks to Brewfather. With a premium account, Brewfather supports logging brew-related data. To find your API key; log into your Brewfather account and go to Custom Stream. Your API key will be a 10 to 64-character string of letters and numbers on the line following "URL http://log.brewfather.net /stream?id=" (do not include the the URL.)
To disable pushing to Brewfather, delete the key.
Brewfather Key
q4F3wPfooBa3X3
Push Frequency
15
Update

- **Link to External Site (optional):** https://www.brewbubbles.com
- **Link to GitHub (optional):** https://github.com/lbussy/brew-bubbles/

After you create your channel, you may optionally go into "Sharing" and allow people to view your channel. The public URL may be discovered by selecting the "Public View" tab.

Channel ID: Go to 'My Channels.' Select the 'API Keys' tab for your channel. The channel ID and write API key will be displayed. The Channel ID is a number towards the top in bolded characters.

Channel Write Key: Go to 'My Channels.' Select the 'API Keys' tab for your channel. The channel ID and write API key will be displayed. You must use the "Write" key, the "Read" key will not allow posting data. If you ever wish to generate a new write API key, you must re-enter it into Brew Bubbles or else posting will fail.

Push Frequency: Enter the push frequency in minutes. Users of a free account are limited to sending no more than 3 million messages each year to the ThingSpeak service. This works out to approximately 5 posts per minute. Users of the free license will also be limited to 4 channels. Since Brew Bubbles only allows you to send once per minute, and ThingSpeak limits you to four free channels, you are unlikely to find a way to exceed your quota.

Advanced

I will cover Firmware Update and WiFi reset in subsequent sections.

2.6 Firmware Update

After you initially flash the controller, you may update Brew Bubbles via the web page when new versions are released. You may do this without erasing any of the Brew Bubbles' settings. If you instead choose to flash new firmware and LittleFS manually, your application configuration is lost. This page displays your controller's current version, as well as the latest version available.

If you wish to proceed, clicking "Update Firmware" button begins the process.

A page displays while the upgrade is in progress.

Warning: Do not close this page. If you close the page before the update is complete, you may lose your application settings.

During this process, both the firmware and the LittleFS updates apply. When the update is complete, the application settings are re-applied, and a completion message is displayed.

The controller's onboard LED may flicker during the upgrade. Once complete, the LED remains steady on or off, depending on whether the sensor is blocked.

The page redirects after this message and opens the main page, or you may navigate there manually.

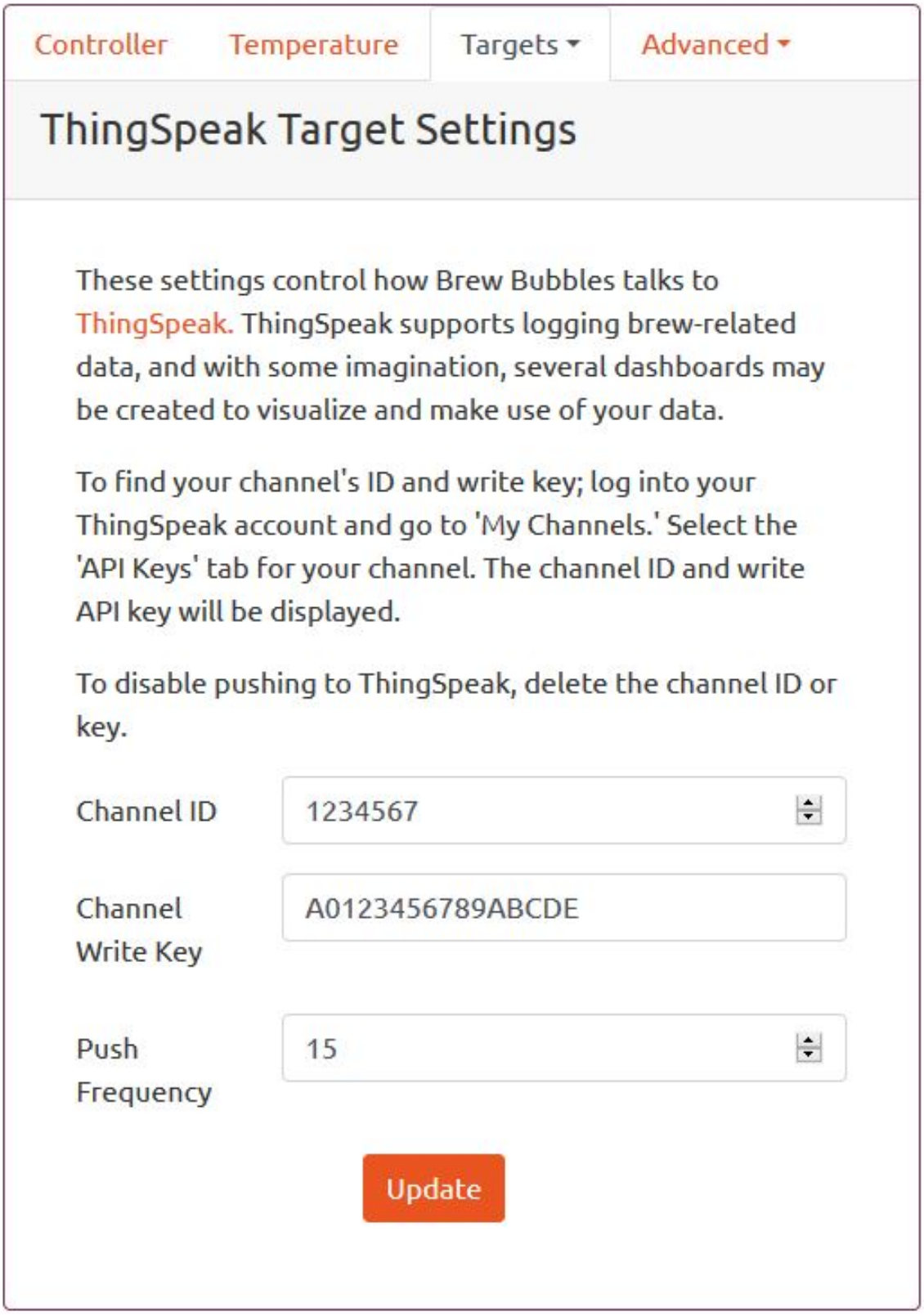
2.7 WiFi Reset

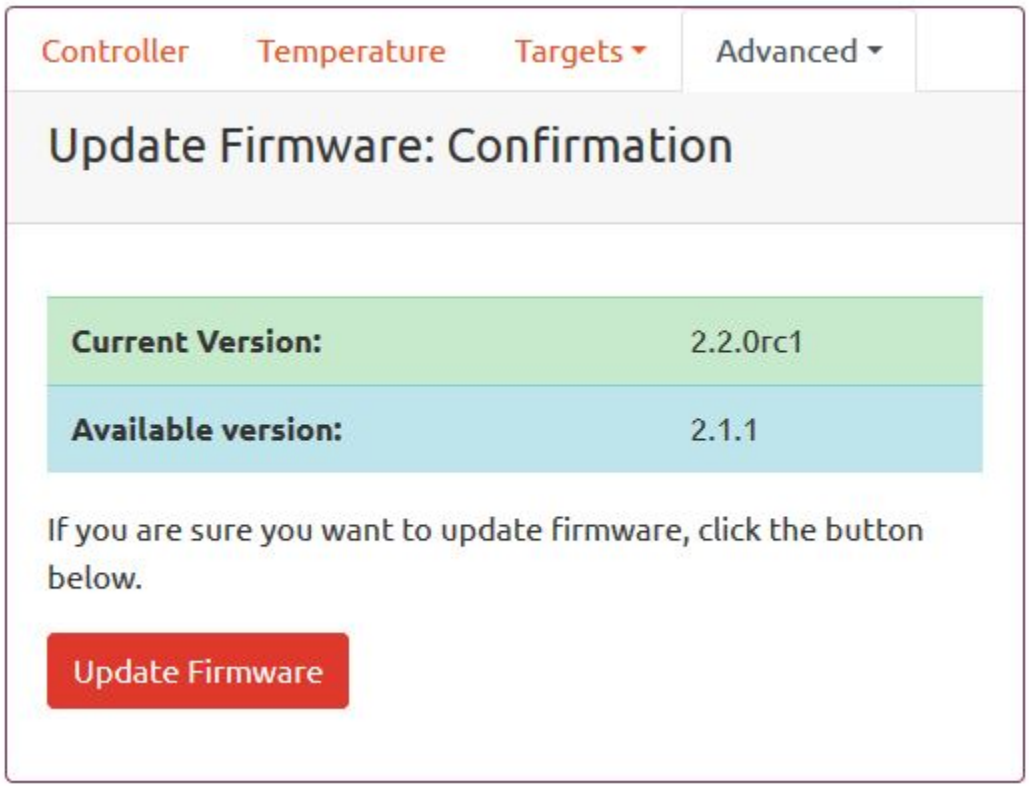
I've provided the WiFi reset function to erase the WiFi settings from the controller if you wish to move the device to a new network.

If you click the "Reset WiFi" button, the controller presents you with a confirmation page:

Clicking "Reset WiFi" again confirms the process. You are presented with a page informing you that the settings have reset.

At this point, you cannot control the device again until you configure the WiFi. The controller resets, and the Access Point activates where you may connect and configure it via the captive portal.


Controller
Temperature
Targets
Advanced
ThingSpeak Target Settings
These settings control how Brew Bubbles talks to ThingSpeak. ThingSpeak supports logging brew-related data, and with some imagination, several dashboards may be created to visualize and make use of your data.
To find your channel's ID and write key; log into your ThingSpeak account and go to 'My Channels.' Select the 'API Keys' tab for your channel. The channel ID and write API key will be displayed.
To disable pushing to ThingSpeak, delete the channel ID or key.
Channel ID
1234567
Channel Write Key
A0123456789ABCDE
Push Frequency
15
Update



Be aware that it is exceedingly easy to retrieve the WiFi password from the controller unless you erase it. If you give a controller to another person (any controller, any application), be sure to wipe the settings, or possibly the entire device.

2.8 Brew Bubbles API

Brew Bubbles stores and communicates with external agents via JSON and POST methods. This document details those mechanics.

Contents

- *Brew Bubbles API*
  - *System Configuration*
    - */config/*
  - *Outbound API*
    - */bubble/*
    - */thisVersion/*
    - */thatVersion/*
  - *Inbound API*
    - *Configuration Ingestion*

Please Wait ...

Your Brew Bubbles' firmware is being updated to the latest version. This can take up to 5 minutes, during which your Brew Bubbles will be unresponsive. Please do not disconnect the power or reset your Brew Bubbles while this process is taking place.

If you wish to observe your controller's LED during the process, the LED will flash as the update is in progress. As soon as the LED is steady - either off if blocked or on if not blocked, the process is complete.

When the update is complete, the controller will re-load your original application settings. If this step fails, you will need to manually reconfigure all application settings. WiFi settings will not be affected.

Do not refresh this page. If you do, you will not be able to track the upgrade process.

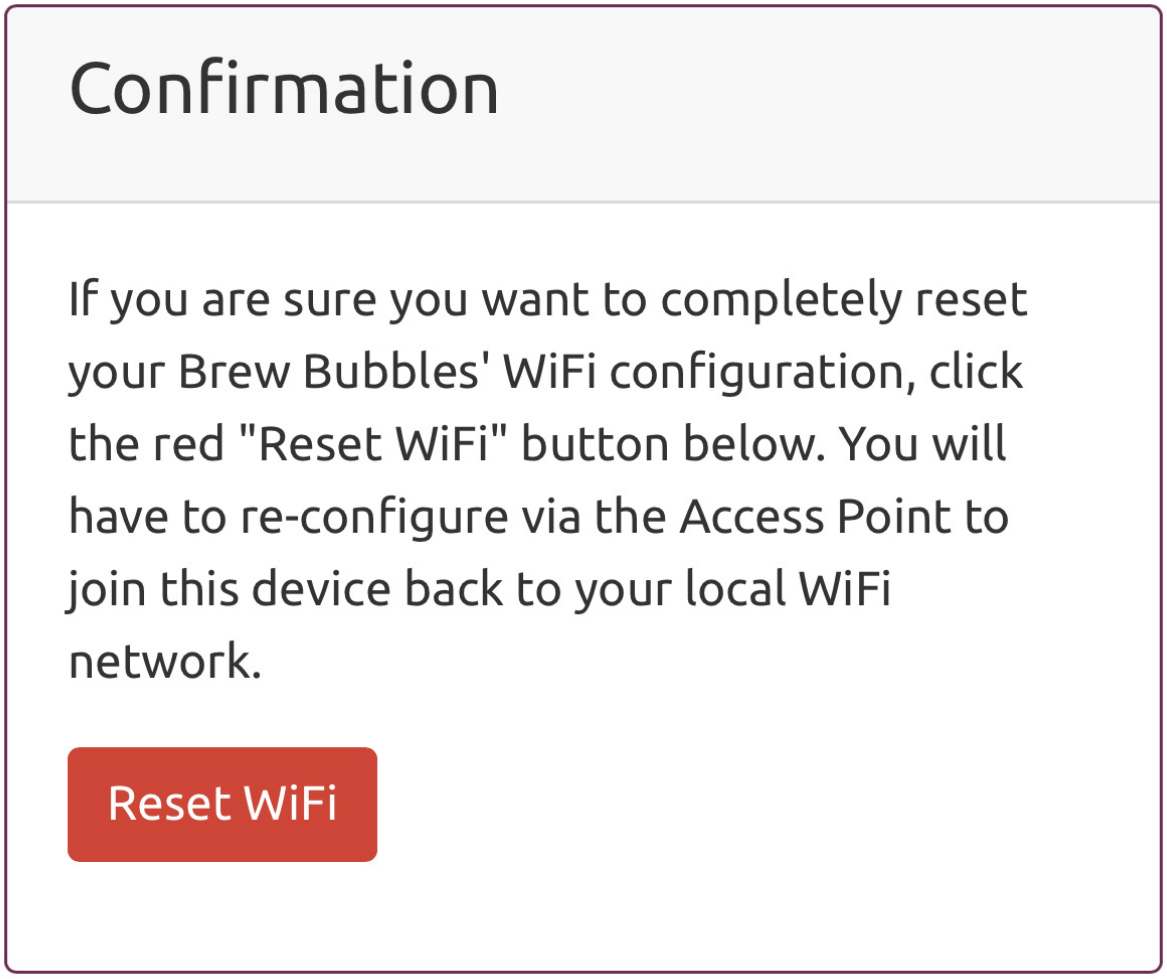
Redirect Pending

The firmware update is complete. You will be redirected momentarily.

Reset WiFi

If you plan on relocating your Brew Bubbles to a new network, you can reset the WiFi connection settings by clicking "Reset WiFi". You will need to perform the the original network setup again by connecting to this device as an access point.

Reset WiFi


Confirmation
If you are sure you want to completely reset your Brew Bubbles' WiFi configuration, click the red "Reset WiFi" button below. You will have to re-configure via the Access Point to join this device back to your local WiFi network.
Reset WiFi

Resetting Configuration

Your Brew Bubbles' WiFi connection settings are being reset which will cause it to forget your network configuration. After this is complete, it will reboot, and will create the configuration Access Point. To reconnect your Brew Bubbles to your network you will need to connect to the configuration Access Point and provide new WiFi settings.

        * *Control Points*
    - *Downstream Targets*
        * *General HTTP Targets*
        * *Brewer's Friend*
        * *Brewfather*
        * *ThingSpeak*

2.8.1 System Configuration

The device stores its configuration in JSON within LittleFS. Web pages retrieve the configuration data at the following endpoint:

/config/

Triggers the controller to send all of its configuration items

```
{
"apconfig": {
    "ssid": "brewbubbles",
    "passphrase": "brewbubbles"
},
"hostname": "brewbubbles",
"bubble": {
    "name": "Fermenter 1",
    "tempinf": true
},
"calibrate": {
    "room": 0,
    "vessel": 0
},
"urltarget": {
    "url": "",
    "freq": 2,
    "update": false
},
"brewersfriend": {
    "channel": 0,
    "key": "c6e88f70f575c4ecdca3dcb686381185",
    "freq": 15,
    "update": false
},
"brewfather": {
    "channel": 0,
    "key": "q4F3wPfooBa3X3",
    "freq": 15,
    "update": false
},
"thingspeak": {
    "channel": 1244893,
    "key": "AB6C1ME2NWS1MSDS",
    "freq": 15,
```

(continues on next page)

(continued from previous page)

```
    "update": false
},
"dospiffs1": false,
"dospiffs2": false,
"didupdate": false
}
```

These keys represent the following settings:

apconfig:

- ssid: The AP name (SSID) which broadcasts when the controller is in AP mode
- appwd: The AP password required to connect to the AP

hostname:

- The mDNS hostname used by the controller

bubble:

- name: The fermenter name
- tempinf: Controls whether the device reports in Fahrenheit (true) or Celcius (false)

calibrate:

- room: The offset applied to the room sensor
- vessel: The offset applied to the vessel sensor

urltarget:

- url: The HTML endpoint to which we send a JSON POST
- freq: The frequency (in minutes) at which the application sends a status POST
- update: Internal semaphore to trigger a reconfiguration

brewersfriend:

- channel: Not used
- key: The hexadecimal key provided by Brewer's Friend to allow POSTing data to your brew
- freq: The frequency (in minutes) at which the application sends a status POST to Brewer's Friend
- update: Internal semaphore to trigger a reconfiguration

brewfather:

- channel: Not used
- key: The hexadecimal key provided by Brewfather to allow POSTing data
- freq: The frequency (in minutes) at which the application sends a status POST to Brewfather
- update: Internal semaphore to trigger a reconfiguration

thingspeak:

- channel: A numeric channel ID
- key: The hexadecimal key provided by ThingSpeak to allow writing data to your channel
- freq: The frequency (in minutes) at which the application sends a POST to ThingSpeak
- update: Internal semaphore to trigger a reconfiguration

dospiffs1:

- Sets a semaphore indicating that the controller has reset one time after the firmware update

dospiffs2:

- Sets a semaphore indicating that the controller has reset two times after firmware update and that LittleFS update may begin

didupdate:

- An indication that both firmware and LittleFS OTA has completed

2.8.2 Outbound API

/bubble/

Triggers the controller to send the last status payload:

```
{
    "api_key": "Brew Bubbles",
    "device_source": "Brew Bubbles",
    "name": "Fermenter 1",
    "bpm": 123.456,
    "ambient": 72.5,
    "temp": 68.1,
    "temp_unit": "F",
    "datetime": "2020-12-03T20:44:40Z"
}
```

/thisVersion/

This endpoint returns the current controller firmware and LittleFS version in JSON format:

```
{
    "version": "2.2.0rc1",
    "branch": "devel",
    "build": "8ec3d68"
}
```

/thatVersion/

This endpoint returns the currently available controller firmware and LittleFS version from the Brew Bubbles website in the same format as the local version.

2.8.3 Inbound API

The controller uses inbound web page access to configure and control Brew Bubbles.

Configuration Ingestion

The controller uses inbound POST endpoints to configure Brew Bubbles:

/settings/controller/

- mdnsid: Sets the hostname of the controller
- bubname: A display name for the controller

/settings/temperature/

- calroom: A floating-point number by which the room/ambient sensor is offset
- calvessel: A floating-point number by which the room/ambient sensor is offset
- tempformat: String, celsius or fahrenheit to configure temperature reporting

/settings/urltarget/

- urltargeturl: The URL to which reports are posted
- urlfreq: Frequency for reports in minutes

/settings/brewersfriendtarget/

- brewersfriendkey: Brewer's Friend key
- brewersfriendfreq: Frequency for reports in minutes

/settings/brewfathertarget/

- brewfatherkey: Brewfather key
- brewfatherfreq: Frequency for reports in minutes

/settings/thingspeaktarget/

- thingspeakchannel: Channel to which the data will be posted
- thingspeakkey: Write key for the channel
- thingspeakfreq: Frequency for reports in minutes

/clearupdate/

- Clears all update related semaphores.

Control Points

The following pages take action upon access:

/wifi2/: Accessing this page resets all WiFi configuration items and resets the controller.

/otastart/: Accessing this page begins the OTA update process independent of web page processes.

2.8.4 Downstream Targets

Downstream targets are systems to which Brew Bubbles sends data on a schedule. Sending data to various targets is done in similar yet specific formats.

Note that since the temperature probes are optional, they report as -100 in either temperature format when not connected. A sensor failure also results in this reading.

General HTTP Targets

General targets are targets that take an unqualified HTTP post. Currently, systems that are known to support Brew Bubbles are BrewPi Remix and Fermentrack.

Brew Bubbles makes the post with no authentication nor key, and in the following format:

```
{
    "api_key":"Brew Bubbles",
    "device_source":"Brew Bubbles",
    "name":"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "bpm":99.999,
    "ambient":70.3625,
    "temp":-196.6,
    "temp_unit":"F",
    "datetime":"2019-11-16T23:59:01.123Z"
}
```

Brewer's Friend

Brew Bubbles natively and specifically supports posting data to Brewer's Friend. The payload sent to Brewer's Friend is according to the following format:

```
{
    "api_key":"Brew Bubbles",
    "device_source":"Brew Bubbles",
    "name":"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "bpm":99.999,
    "ambient":70.3625,
    "temp":-196.6,
    "temp_unit":"F",
    "datetime":"2019-11-16T23:59:01.123Z"
}
```

Brewfather

Brew Bubbles natively and specifically supports posting data to Brewfather. The payload sent to Brewer's Friend is according to the following format:

```
{
    "api_key":"Brew Bubbles",
    "device_source":"Brew Bubbles",
    "name":"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "bpm":99.999,
    "aux_temp":70.3625,
    "temp":-196.6,
    "temp_unit":"F",
    "datetime":"2019-11-16T23:59:01.123Z"
}
```

ThingSpeak

ThingSpeak receives POST reports according to their provided library.

2.9 Troubleshooting

The easiest way to get help with the project is to join the Brew Bubbles discussion on Homebrewtalk.com. You may also open an issue on Github.

This page includes various bits and pieces that do not fit elsewhere, which may help you if things are not going as expected.

Contents

- *Troubleshooting*
  - *LED Flashing*
  - *Double-Reset Detect*
  - *Emergency AP mode*
  - *Serial Debug*
  - *Telnet Debug*
  - *Controller Reset*
  - *About page*
  - *FAQ*

2.9.1 LED Flashing

In typical operation, the blue LED on the controller indicates the photo-receptor status. The LED lights when the photo-receptor is not blocked (or when a bubble passes), and is dark when the photo-receptor is blocked (no bubble).

During non-operating mode, the LED flashes at different frequencies as an indicator of the process underway.

0.5Hz: The LED blinks at 0.5Hz or 1 second on, one second off when in Access Point mode

0.66Hz: The LED blinks at 1.5Hz (33 seconds on, .33 seconds off) while the controller attempts to connect to your WiFi. After trying for 30 seconds, the connection attempt fails, and the controller enters AP mode. AP mode times out after 120 seconds and resets the controller.

10Hz: The controller needs to access an Internet Network Time Protocol (NTP) server to set the correct time. While attempting to get the time, the LED flashes at 10Hz or .05 seconds on, .05 seconds off. Occasionally, the controller cannot get network time, in which case it remains in that loop indefinitely since the controller cannot operate without the correct time. If you see this, you can reset the controller with the reset button located within the cutout and try again.

2.9.2 Double-Reset Detect

If you need to access the AP configuration portal without resetting the WiFi settings via the web page, press the reset button twice within 10 seconds. The controller resets and starts AP mode, where the LED flashes at 0.5Hz. If the LED does not start flashing at the 0.5Hz rate, repeat the double reset until it does. You need not treat the resets like a double-mouse click; once per second is usually sufficient.

2.9.3 Emergency AP mode

If you cannot access the AP by any other means, you may do so by grounding D5 on the controller and resetting it via the reset button or with a power cycle. The controller then starts in AP mode, where the LED flashes at 0.5Hz.

2.9.4 Serial Debug

Serial debug has been left enabled in the firmware. You may connect the device to your computer and use a terminal program set at a baud rate of 74880 to review the debug messages. Rudimentary commands are allowed:

2.9.5 Telnet Debug

A simple Telnet server is available to connect with PuTTY or another terminal emulator over port 23. There is no security to this connection; the connection will only emulate the serial connection display as if it was directly connected to your computer. The same serial commands are available as described above.

2.9.6 Controller Reset

You may perform a reset on the controller by navigating to Settings > Advanced > Reset. The controller will reset as if you had pressed the reset button on the controller itself.

2.9.7 About page

The "About" page shows some diagnostic information which may be helpful in certain circumstances:



- **Header:** The header displays the firmware's tagged version, followed by the branch, followed by the commit. This information may be helpful where confusion exists about which version is installed.
- **Uptime:** Uptime shows how long the controller has been running without a restart. The controller may reset itself for several reasons, including a 42-day reboot cycle.
- **Reset Reason:** The reason for the last reboot according to the core libs.
- **Heap Information:** Memory information for the controller. Low memory conditions may cause crashes or occasionally self-reboots to attempt to heal the issue.

2.9.8 FAQ

No FAQ yet; I'll capture that as time goes on.