
Fermentrack Documentation

John Beeler

Sep 03, 2023

Contents:

1	What is Fermentrack?	3
1.1	Included with Fermentrack	3
1.2	Why Use Fermentrack? (New Features)	4
1.3	Requirements	4
1.4	Getting Started with Fermentrack	4
1.5	Other Notes	5
2	Installing Fermentrack	7
2.1	Preparing a Raspberry Pi for Fermentrack	7
2.2	Installing Fermentrack	8
2.3	Installing Fermentrack without Docker	9
2.4	Coexistence with Other Apps	10
2.5	Installing Raspbian from Windows 7	11
3	Setting up Fermentrack	15
3.1	User Setup	15
3.2	Site Settings	15
3.3	Notes for Advanced Users	16
4	Adding a BrewPi Controller	17
4.1	Flashing a Controller	17
4.2	Adding a BrewPi Controller to Fermentrack	18
4.3	BrewPi Controller Configuration	20
5	Adding a Gravity Sensor	23
5.1	Tilt Setup	23
5.2	iSpindel Setup	26
5.3	Manual Sensor Setup	26
6	Adding a TiltBridge	27
6.1	TiltBridge Setup	27
7	Changelog, Licensing, and Development	29
7.1	Fermentrack Architecture	29
7.2	Changelog	30
7.3	Included Components & Licensing	42
7.4	Contributing to Fermentrack	43

7.5	“Push” Support	44
7.6	License	47
8	Supported Hardware	49
8.1	BrewPi Controllers	49
8.2	Specific Gravity Sensors	50
9	Frequently Asked Questions	51
9.1	Can I change the temperature format on a beer I’ve started logging?	51
9.2	Help - I forgot my Fermentrack login/password!	51
9.3	What happens to my beer logs/active profiles/other data if I change the Fermentrack “Preferred Time-zone”?	52
10	Documentation To-Do List	53



F E R M E N T R A C K

CHAPTER 1

What is Fermentrack?

Fermentrack is a web interface for controlling and monitoring fermentation temperature and progress. It interfaces with BrewPi controllers as well as specific gravity sensors like the Tilt and iSpindel in order to help you make better beer. Fermentrack is designed to be run on a Raspberry Pi, but can be used in most environments capable of running Docker.

Fermentrack is a complete replacement for the web interface used by BrewPi written in Python using the Django web framework. In addition to the features bundled with brewpi-www, Fermentrack provides an easy install process, simple flashing of both Arduino-based and ESP8266-based BrewPi controllers, and the ability to control multiple chambers at once. Fermentrack also supports features such as WiFi and Bluetooth connectivity to help reduce clutter in your brewery.

To help track your wort's progress on it's journey to become beer, Fermentrack also supports multiple specific gravity sensors both on a standalone basis as well as in conjunction with BrewPi-based fermentation temperature controllers. Currently, Fermentrack supports the [Tilt Hydrometer](#) and [iSpindel](#) devices for automated gravity logging in addition to providing the ability to log manual gravity readings taken with traditional hydrometers or refractometers.

1.1 Included with Fermentrack

- **Fermentrack** - The Django-based replacement for brewpi-www. Licensed under MIT license.
- **brewpi-script** - Installed alongside Fermentrack is brewpi-script. Licensed under GPL v3.
- **BrewPi Firmware (Various)** - Fermentrack can be used to install various versions of the BrewPi firmware. Most of these are licensed under GPL v3, though other licenses may apply.
- **Other Firmware (Various)** - Fermentrack also can be used to install firmware such as iSpindel and BrewPiLess to compatible devices. Licenses may vary.

Other components used in or bundled with Fermentrack may have their own licensing requirements. These components can be referenced at [Included Components & Licensing](#)

1.2 Why Use Fermentrack? (New Features)

Fermentrack adds a number of features that are missing in the official BrewPi web interface. The following are just some of the features that have been added or enhanced:

- Single Command Installation (See: [Installing Fermentrack](#))
- Native multi-chamber support
- Cleaner, more intuitive controller setup
- Integrated support for ESP8266-based controllers
- Official support for “legacy” controllers
- Native support (including mDNS autodetection) for WiFi controllers
- Robust device detection for serial controllers
- Support for Tilt and iSpindel specific gravity sensors
- Support for Python 3
- Support for Docker-based installations

1.3 Requirements

All Fermentrack installations require the following:

For Raspberry Pi-based Installations: * Raspberry Pi 2B, 3, 4, 400, (or newer) /w Internet Connection * Raspberry Pi OS (Raspbian) install (Buster or later) * 2GB of free space available

Note: Fermentrack does NOT support installing to armv6-based Pis (e.g. Pi Zero, Pi Zero W, Pi 1).

For x86/x64-based Installations: * A Debian Buster or later-based-OS (e.g. Ubuntu, Debian, RPi OS for x86) * 2GB of free space available

Additionally, a Bluetooth receiver is required for [Tilt Hydrometer](#) support. (For Raspberry Pi installations, the adapter built into a Pi 3+ is sufficient)

1.4 Getting Started with Fermentrack

Getting started with Fermentrack is incredibly easy! All you need to do is:

1. Install Raspbian on your Raspberry Pi
2. Install Fermentrack (one command!)
3. Configure Fermentrack from your web browser
4. Connect & configure your BrewPi temperature controllers or specific gravity sensors

It can be done from start to finish in a bit under an hour, assuming all your hardware is assembled & ready to go. To learn how, read [Installing Fermentrack](#).

1.5 Other Notes

Fermentrack is currently designed for “legacy” BrewPi firmware running on ESP8266 and Arduino hardware, and does not support “modern” firmware such as that included with current official BrewPi controllers.

A full table of controllers/expected hardware availability is available in [Supported Hardware](#).

Warning: Fermentrack is currently intended to be installed on a fresh installation of Raspbian. It is **not** intended to be installed alongside brewpi-www and will conflict with the apache server brewpi-www installs. If you intend to use Fermentrack alongside an installation of Raspberry Pints or another PHP-based application, read [installing/Apache_and_PHP_Support](#).

Installing Fermentrack

2.1 Preparing a Raspberry Pi for Fermentrack

Prior to installing Fermentrack, you need to install Raspbian and set everything up. Click the link below to watch a video showing how to prepare the Raspberry Pi using a Mac, or read the linked instructions below for your operating system.

2.1.1 Prepare the Raspberry Pi - [Video]

1. Download the latest version of Raspbian from [here](#). I recommend the Lite version as I prefer headless installations, but the full version works as well.
2. Burn Raspbian to your SD card using [these instructions](#).
3. [Enable SSH](#) on your Raspberry Pi by writing an empty file named “ssh” to the root of the SD card.
4. *Optional* - Pre-configure WiFi - See the note below if you want to configure WiFi now, thus preventing having to find an ethernet cable
5. Plug the SD card into your Raspberry Pi, connect the Pi to ethernet (if you did not configure WiFi), and plug in power.
6. Locate the IP address for your Raspberry Pi This can generally be done by executing `arp -a | grep raspberry` however you can also locate your Raspberry Pi by logging into your router and looking for the device.
7. Update the Raspberry Pi software by running `sudo apt-get update` and `sudo apt-get upgrade`.
8. Run `raspi-config` and configure the Pi. At a minimum, expand the filesystem (option 1).
9. Update the default password for the pi user using `passwd`
10. *Optional* - If you didn't do it earlier, [Configure WiFi](#) on your Raspberry Pi (if needed)

2.1.2 Additional Info about Install-Time WiFi Configuration

To configure WiFi on a headless install (or otherwise configure it at setup) prior to the initial boot on a newly flashed Raspbian installation, create a `wpa_supplicant.conf` file in the `/boot` directory of the SD card with the following contents (adjusting to match your network configuration):

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
country=US
network={
    ssid="YOUR_SSID"
    psk="YOUR_PASSWORD"
    key_mgmt=WPA-PSK
}
```

Note - In the above, `ssid` is the name of your wireless network, `psk` is the password for your wireless network (if applicable), and `key_mgmt` is the password management protocol (which, for most home networks these days is WPA-PSK) You will also need to select the appropriate 2 letter [country code](#) for where you plan on using the Raspberry Pi.

2.2 Installing Fermentrack

Fermentrack is intended to be installed as a Docker container as part of a Docker Compose stack. There are three main ways to install Fermentrack:

- One-line fully automated
- Semi-automated
- Manual

Regardless of what method you choose, all of these expect that your Raspberry Pi has been properly set up with a working copy of Raspberry Pi OS. If Raspberry Pi OS is not yet installed, please read and complete [Preparing a Raspberry Pi for Fermentrack](#).

The installation will take around 30 minutes, depending on which Raspberry Pi you are using.

2.2.1 Fully Automated

The easiest way to install Fermentrack is via a single command executed on your Raspberry Pi. To install via this method, SSH into your Raspberry Pi and execute:

```
curl -L install.fermentrack.com | bash
```

Follow the prompts as needed, and once the script completes, you're done!

You can also watch a video of this process on YouTube at: XXXX

Todo: Come back and re-add the video link once the video of the docker install is available.

2.2.2 Semi-Automated

If you prefer a slightly less automatic installation method, you can download the **'fermentrack-tools'** repo from git and use the install script contained therein. To install using this script, do the following:

1. Log into your Raspberry Pi
2. install git (Type `sudo apt-get install git-core`)
3. Clone the repo (`git clone https://github.com/thorak/fermentrack-tools.git`)
4. Run the install script (`cd fermentrack-tools && sudo ./install.sh`)
5. Follow the prompts on screen. Relaunch the install script if prompted.
6. Once installation completes, open a web browser and connect to Fermentrack to complete the installation process.

2.2.3 Manual

If you want to install Fermentrack manually, simply open the `install.sh` script from the **‘fermentrack-tools’** repo and follow the commands in the order they are executed by the script.

2.3 Installing Fermentrack without Docker

The use of Docker is recommended for all Fermentrack installations that are not being used for active development, as Docker ensures a consistent environment and simple upgrades.

There are two main ways to install Fermentrack without Docker:

- Semi-automated
- Manual

Regardless of what method you choose, all of these expect your Raspberry Pi to be adequately set up with a working copy of Raspbian. If you have not yet installed Raspbian please read and complete [Preparing a Raspberry Pi for Fermentrack](#)

Warning: The overwhelming number of issues with Fermentrack are caused by environmental problems Docker helps prevent.

2.3.1 Semi-Automated

The easiest way to install without Docker is to run the legacy non-docker install script. To do this, simply complete the following:

1. Log into your Raspberry Pi
2. install git (Type `sudo apt-get install git-core`)
3. Clone the repo (`git clone https://github.com/thorak/fermentrack-tools.git`)
4. Run the install script (`sudo fermentrack-tools/non_docker_install/install.sh`)
5. Follow the prompts on screen. Relaunch the install script if prompted.
6. Once installation completes, open a web browser and connect to Fermentrack to complete the installation process.

This script will:

- Install the software/packages necessary for Fermentrack to run

- Install Fermentrack
- Install Nginx
- Set up Nginx to access Fermentrack

2.3.2 Manual

If you want to install Fermentrack manually, simply open the `non_docker_install/install.sh` script from the **'fermentrack-tools'** repo and follow the commands in the order they are executed by the script.

Note: Many of these commands require being executed as `sudo` to work.

2.4 Coexistence with Other Apps

Unlike apps such as RaspberryPints and BrewPi-www, which use [Apache](#) to directly serve webpages, Fermentrack uses [Nginx](#) as a reverse proxy to pass connections to the Fermentrack application running on your Pi. Fermentrack can coexist peacefully with other apps - including those that use Apache - but some additional configuration will be required.

Note: This document assumes that your installation of Fermentrack is running inside Docker

2.4.1 Port Utilization

For Docker-based installations, port utilization is the primary issue preventing coexistence with other apps. Fermentrack uses several, all of which must be free for the installation to be successful. The most likely of these ports to be occupied - port 80 - can be changed at the time of installation by running the installation script manually (see below).

- 80 - **Configurable** - The primary port through which Fermentrack is accessed
- 5432 - Used by Postgres
- 6379 - Used by Redis
- 7555 - The socket through which the Fermentrack app is directly accessed
- 7556 - Used by Circus
- 7557 - Used by Circus

BrewPi-script instances may use additional ports not listed above - these are configurable through the Fermentrack interface.

2.4.2 Changing Fermentrack away from Port 80

Port 80 is the standard port for most web applications - including Fermentrack - which also causes it to be the most likely to have conflicts with other applications. If you need to change the port Fermentrack is accessible through this can be accomplished by re-running the `install.sh` script included with [fermentrack-tools](#).

1. Log into your Raspberry Pi via `ssh`
2. Change to the `fermentrack-tools` directory

3. Run `./install.sh`
4. When prompted, select a port other than 80.

2.5 Installing Raspbian from Windows 7

here

2.5.1 Download and Install Raspbian

1. Download the latest version of Raspbian [here](#). You can download either the Lite or Full version - the Lite version is good for “headless” setups where you won’t have a monitor & keyboard hooked up to your Raspberry Pi, the full version includes a graphical interface for use with a monitor/keyboard.
2. Download and install [Etcher](#) as recommended [here](#).
3. Burn Raspbian to your SD card using Etcher:
 - Connect the SD card you will be installing Raspbian onto to your Windows PC using a removable SD card adaptor
 - Select the proper Raspbian .zip file
 - Select the proper removable drive to flash (Etcher only allows you to select removable drives)
 - Flash to the SD Card
 - Navigate to and open the SD Card to verify files were flashed
4. Enable SSH on your Raspberry Pi by writing an empty file named “ssh” to the root of the SD card via the Notepad Windows Program:
 - Run Notepad
 - In a new file, put in one space and nothing else
 - Click File > Save As, and save the file to the root (lowest level) directory on the SD Card:
 - Name the file `ssh`
 - NOTE - Be certain to Save as type: All Types (*.*)
 - Close the file
5. Configure WiFi (Optional, but required if running the Lite/Headless Version and you do not plan to connect the Raspberry Pi via Ethernet Cable):
 - Run Notepad
 - In a new file, paste the contents below. (Inserting the proper country code, network name, and network password) Network Names with some symbols may be problematic. If you have issues connecting, eliminate your SSID from having any unusual symbols. When entering your network name and password include the Quotes.

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="NETWORK-NAME"
```

(continues on next page)

(continued from previous page)

```

    psk="NETWORK-PASSWORD"
}

* Click File > Save As, and save the file to the root (lowest level) directory on the
  ↳ SD Card:
* Name the file ``wpa_supplicant.conf``
* Be certain to Save as type: All Types (* *)
* Close the file

```

1. Verify that the `ssh` and `wpa_supplicant.conf` files were created in the Root Directory.

- Open My Computer (File Explorer)
- Navigate and open the removable SD Card containing Raspbian
- Confirm that the `ssh` file and the `wpa_supplicant.conf` file can be seen
- Navigate off the SD Card file.
- Eject SD Card properly.

2. Remove the SD Card and plug the SD Card into your Raspberry Pi and plug in power to the Raspberry Pi. (If you did not configure WiFi, connect the Pi to Ethernet, plug in power) **NOTE - Give time to boot, this can take 60 seconds or longer**

2.5.2 Configure your Raspberry Pi

8. Download and install Putty (<http://www.putty.org/>). (This next step assumes that `ssh` is enabled on the Raspbian Image (Step 4) and that you properly created the `wpa_supplicant.conf` file.(Step 5))

Step 9. Login over WiFi/Ethernet using Putty. (Your Windows device MUST be on the same WiFi Network as you configured your Pi for) 9.1 Launch Putty 9.2 Set the Host Name (or IP address) Field to: `raspberrypi.local` (you can also log into your router, look for the device, and enter the correct IP address into Putty) 9.3 By default the Port should be set to 22 and the connection type should be set to SSH. 9.4 Click Open 9.5 If you see a Security Alert, select Yes. 9.6 A new terminal window will pop open prompting you for a user name. 9.7 The default user name is: `pi` 9.8 The default password is: `raspberrypi`

(You can now access your Pi via WiFi)

10. Update the Raspberry Pi Software. 10.1 Run the command `sudo apt-get update -y` on the Raspberry Pi using SSH (Putty) 10.2 Run the command `sudo apt-get upgrade -y`

11. Configure Raspberry Pi (RPi 3 B+ with Raspbian Lite shown) 11.1 Run the command `sudo Raspi-config` 11.2 Change User Password from default `raspberrypi` • Option #1 • Enter new password 11.3 (Optional) Change Hostname from default `raspberrypi`. • Option #2 • Option N1 • Enter new Hostname • (NOTE - Changing the Hostname will alter how you login via Putty. If Hostname is changed, in step 9.2 you will need to enter the new hostname similar to `newhostname.local` and on step 9.8 you would need to enter your new password. 11.4 (Optional, if needed & not done earlier using `wpa_supplicant.conf`) Configure WiFi • Option #2 • Option N2 • Follow Prompts 11.5 Reboot

Raspberry Pi is now ready for Fermentrack Install.

2.5.3 (Optional) Set up additional, optional networks

::todo Change this to the main Raspi Setup file

If you move your Raspberry Pi around often, or potentially need to connect to multiple networks, you can configure the `wpa_supplicant.conf` file to contain multiple network options. To do so, do the following:

1. Connect Raspberry Pi and Windows based hardware on the same network as original installation.
2. Launch Putty and login to Raspberry Pi.
3. Determine additional Networks to configure to. (If you don't know, run the following command to locate local networks) 3.1 Run the command `sudo iwlist wlan0 scan` 3.2 Record the ESSID you want to connect to (you will need to know the password)
4. Add the network details to the Raspberry Pi 4.1 open the `wpa_supplicant.conf` file • Run the command `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf` 4.2 Find the following from Installation Step 5.2:

```
network={
    ssid="NETWORK-NAME"
    psk="NETWORK-PASSWORD"
}
```

- 4.3 Add priority and Network ID to original network configuration.

```
network={
    ssid="NETWORK-NAME"
    psk="NETWORK-PASSWORD"
    priority=1
}
```

- 4.4 Add additional Networks under your main and set priority:

```
network={
    ssid="additional-network-name"
    psk="additional-network-password"
    priority=2
}

network={
    ssid="Secondary-Network-Name"
    psk="Secondary-Network-Password"
    priority=3
}
```

- 4.5 Save your New Network Configuration. (Press the following) • Ctrl + x • Y • Enter 4.6 Reboot the Pi by running `sudo shutdown -r now`

5. Confirm that Raspberry Pi is on Priority 1 Network 5.1 Launch Putty and login in to Raspberry Pi • If connected successfully, congratulations! • If unsuccessful: • Make certain RaspberryPi and Windows hardware are on the same network. • Log Windows/Pi devices into the original network to see if connection can be made. • If Raspberry Pi is lost and can't be connected to, wipe SD card and start the installation process over.

Fermentrack is now ready to be configured.

Setting up Fermentrack

Once you have finished installing Fermentrack, you are ready to configure it. You will be guided through the configuration process when you first connect to Fermentrack. An overview of this configuration procedure is below.

3.1 User Setup

When you first access a new installation of Fermentrack you will be asked to set up a user account. This account will enable you to configure devices, configure the Fermentrack application, and view brew logs.

Setting up the user account is extremely straightforward:

1. From the root Fermentrack page, click “Continue to guided installation”
2. **On the next page, entering the following:**
 - **Username** - The username used to log into Fermentrack
 - **Password** - The password for the user account
 - **Email Address** - Currently unused, but may potentially be used later
3. Click “Create new user account”
4. Done!

You’re now ready to move on to configuring the site settings.

3.2 Site Settings

After configuring a user account, the next step is to configure Fermentrack. The following are the available configuration options:

- **Brewery Name** - The name displayed in the upper left of each Fermentrack page
- **Date time format display** - The date format used on the dashboard of each device

- **Require login for dashboard** - *Currently Unused* - Should users be required to be logged in to *view* the dashboard/logs. Login will still be required to change temperature settings, configuration options, etc.
- **Temperature format** - The preferred (default) temperature format. Used when setting fermentation temperature profiles. Can be overridden per device.
- **Preferred timezone** - The preferred timezone you would like graphs displayed in. *Note* - All data is stored in UTC (GMT) - you are only selecting how the data will be *displayed*. Feel free to change this at any time with no issues.
- **Git Update Type** - Which releases you will be prompted to upgrade to. Can be set to “tagged versions” (which will generally be tested and stable), “all commits” which will include all new code releases, and “None”.
- **Sentry mode** - Collect code-related information from crashes and send to the developer for fixing.

All of these can be updated at any time by clicking on the “gear” icon in the upper right of any Fermentrack page.

3.3 Notes for Advanced Users

3.3.1 User Accounts

Currently, Fermentrack has limited access control implemented, and is not yet designed for multiple user installations. This feature is planned for a later release - once it becomes available, different levels of access will be able to be specified on a per-user basis.

In the mean time if you need multiple user accounts they can be configured using the Django admin interface (accessible via the “gear” icon). Each account will need “superuser” access to be able to use the full functionality of Fermentrack. Again - please keep in mind - multiple user access is not officially supported. When access control functionality is implemented, any users created previously through this method will retain full access/control of Fermentrack.

3.3.2 Advanced Site Settings

There are a handful of additional site settings which are intended for advanced users and developers only. These settings can only be accessed via the Django admin. These settings include:

- **Allow Git Branch Switching** - Allows switching to a different git branch from within Fermentrack
- **User Has Completed Configuration** - Determines if the user has completed the initial configuration workflow, or should be prompted to set Fermentrack up on next login.

Additionally, graph colors can be configured via the Django admin. The graph color options include:

- **Graph Beer Set Color** - The color of the “Beer Setting” line
- **Graph Beer Temp Color** - The color of the “Beer Temperature” line
- **Graph Fridge Set Color** - The color of the “Fridge Setting” line
- **Graph Fridge Temp Color** - The color of the “Fridge Temp” line
- **Graph Room Temp Color** - The color of the “Room Temp” line
- **Graph Gravity Color** - The color of the “Specific Gravity” line

Adding a BrewPi Controller

4.1 Flashing a Controller

Fermentrack includes software which allows you to easily download & flash firmware to controllers. This feature is currently supported for the ESP8266 and Arduino architectures, and supports multiple families of firmware including BrewPi, BrewPiLess, and iSpindel.

All devices will need be flashed via serial over USB - including devices you ultimately want to use over WiFi.

See also:

A video of this process can be seen on YouTube at <https://youtu.be/vpm-8k8tnGk>

4.1.1 Accessing the Flash Workflow

Flashing a new controller is accomplished through the controller flashing interface. This can be accessed directly or through the guided device setup workflow. From the device menu, choose “Flash Controller”.

4.1.2 Flashing a Controller

1. Once in the flash workflow, click “Refresh Firmware List from Fermentrack.com”. This downloads a list of available, screened firmware from Fermentrack.com.
2. After the list downloads, select your device family and click “Submit”.
3. On the next screen, select the board (hardware variant) your device is based on. For some families, there may only be one option.
4. Double check your device family & board on the following screen and then ensure that the device you want to flash is **not** connected to the computer/device running Fermentrack. Once this is done click “Scan Devices”
5. Review the “Preexisting Devices” list to ensure that the device you want to flash is *not* listed. Assuming this is the case, connect the device you want to flash to the device running Fermentrack via a USB cable. Click “Scan for New Devices”

6. Your device should now be detected and displayed in the list. If it isn't, return to an earlier step and restart the process. Click the "Set Up" button next to your device.
7. On the following screen, select the firmware you want to flash to the device. Click "Flash to Device".
8. You will be redirected to a page that tracks the status of your flash. Once the most recent status changes from 'running' to 'finished', the flashing of your device is complete. Your device is ready to use!

4.1.3 Additional Controller Configuration

Fermentrack is designed to work with any implementation of a BrewPi controller including the ESP8266, Arduino, and ESP32 as well as the python-based Fuscus project. While specific information related to each hardware device is beyond the scope of this document, you can view additional information on each of these on the [Fermentrack GitHub Wiki Page](#)

4.2 Adding a BrewPi Controller to Fermentrack

Setting up a controller that is running the BrewPi firmware is easy, and Fermentrack will guide you every step of the way. If you prefer to jump straight in and set the controller up manually, Fermentrack supports that too. Just jump ahead to *Setting up a controller using the Advanced (Manual) Workflow*.

All of these instructions assume that you have already flashed the relevant firmware to your controller. If you just built it and need to flash it, read *Flashing a Controller* and complete that process before continuing.

Warning: Prior to setting up a controller with Fermentrack, please read any notes specific to your controller's hardware on the :doc:hardware page.

4.2.1 Configuring an ESP8622 controller for WiFi

If you have an ESP8622 controller and wish to connect to Fermentrack via WiFi, follow these steps to associate the ESP8622 board with your LAN wifi network.

1. After flashing an ESP8622, open your host computer's WiFi network selector and look for an access point that begins with "ESP_" and then a string of numbers.
2. Connect to this access point and navigate to any web page- you should be automatically redirected to the ES8266 configuration web page (If not, the configuration page usually runs at <http://192.168.4.1>).
3. Select your LAN wifi network from the networks identified, as well as your wifi password and mDNS name (a text string to identify your ESP8622 board by)
4. Click 'Save'. The ESP8622 will reboot and then connect to your LAN wifi network.

4.2.2 Adding a WiFi-configured ESP8266 controller to Fermentrack using the Guided Workflow

Follow these steps if you wish to add a BrewPi controller that supports WiFi.

1. With the controller disconnected from the Raspberry Pi, launch guided setup and select "Add New Device (Guided)" from the "Select Device to Control" dropdown
2. Select the correct board type from the dropdown and click "Submit"

3. If your device is already flashed, choose “Yes - Already Flashed”. If it isn’t, read [Flashing a Controller](#) before continuing.
4. Select the correct board type (ESP8266) from the dropdown and click “Submit”
5. Select “WiFi” on the left, and then click “Scan WiFi via mDNS”
6. Select the appropriate device from the “Available (Uninstalled) Devices” list, and click “Set Up”
7. Enter a name for the device, adjust settings as needed, and click “Submit”

4.2.3 Adding a serial-connected controller using the Guided Workflow

Note: When setting up a controller to connect via serial, the selected “port” is incredibly important, but is prone to change on reboot or as other devices are connected. Please read [About Serial Port Autodetection](#) for information on how Fermentrack handles this issue.

Follow these steps if you wish to add a BrewPi controller via a serial (USB) connection.

1. With the controller disconnected from the Raspberry Pi, launch guided setup and select “Add New Device (Guided)” from the “Select Device to Control” dropdown
2. Select the correct board type from the dropdown and click “Submit”
3. If your device is already flashed, choose “Yes - Already Flashed”. If it isn’t, read [Flashing a Controller](#) before continuing.
4. If setting up any device other than an ESP8266 click “Begin Serial Autodetection”. If setting up an ESP8266, select “Serial” on the left, and then click “Begin Serial Autodetection”
5. Ensure that the controller **is not** connected to the Raspberry Pi, and click “Scan Devices”
6. Connect the controller to the Raspberry Pi
7. Click “Scan for New Devices”
8. Choose the device that corresponds to your Arduino and click “Set Up”
9. Enter a name for the device, adjust settings as needed, and click “Submit”

About Serial Port Autodetection

By default, Linux assigns serial ports to devices like BrewPi controllers based on the order they are connected. If you have multiple devices connected to your Raspberry Pi this can mean that Fermentrack could potentially mistake one device for another. To prevent this, Fermentrack includes a feature which will disregard the specified serial port when connecting to the controller and will instead connect to the device with the USB serial number that matches the device you set up.

If you wish to disable this feature and instead only connect to the specified serial port, uncheck the “Prefer Connecting Via Udev” option in the “Manage Device” screen.

This feature only works on Linux-based operating systems (including Raspbian for Raspberry Pi), and may not work if Fermentrack is installed on a Mac or Windows-based computer.

4.2.4 Setting up a controller using the Advanced (Manual) Workflow

Note: When setting up a controller to connect via serial, the selected “port” is incredibly important, but is prone to change on reboot or as other devices are connected. Please read [About Serial Port Autodetection](#) for information on how Fermentrack handles this issue.

Setting up a controller using Advanced Workflow

1. Connect the controller to the Raspberry Pi
2. Launch guided setup and select “Add New Device (Advanced)”
3. Enter the configuration options associated with the device.

Todo: Rewrite the Manual Workflow section to document the available options, etc.

4.3 BrewPi Controller Configuration

Once your BrewPi controller is set up within Fermentrack it’s ready for configuration. If you’ve already configured your controller on either another installation of Fermentrack or an installation of brewpi-www, then you may already be ready to go. Otherwise, just follow these instructions to get on your way.

1. Go to your Fermentrack installation’s main page, and click “View Dashboard” underneath the device you wish to configure.
2. From the menu at the top with your device’s name, choose “Configure Sensors/Pins/Devices”
3. A screen will appear showing you the configuration status of your device. At a minimum, you need to have a cooling pin and chamber sensor set for your device to operate.
4. Configure each of the devices listed under “Available Devices” that you plan to use.

4.3.1 Heating/Cooling/Door Sensor Pin

Heating/cooling are configured by pin number. In most cases the correct pin will be prescribed, with the function listed out in the name. (On ESP8266 devices, for example, you will see a header similar to “Pin 16 (D0 (Heat))”. This is generally the “heat” pin.)

To configure, simply choose the device function from the drop down menu (Heating Relay, Cooling Relay, etc.), set whether or not your relay requires the pin to be inverted (for most mechanical relays, you want Invert Pin to be “Inverted”) and click “Assign”

Warning: While the door sensor pin will generally appear as available to configure, you should not set this pin unless you have an actual door sensor connected to your controller. If you set this up incorrectly your controller may think the door to your fermentation chamber is always open, and fail to trigger heating/cooling as expected.

4.3.2 Temperature Sensors

Fermentrack supports the DS18B20 temperature sensors, which use the OneWire protocol. This allows multiple sensors to be connected via a single pin on the controller board. For the ESP8622 board, this is the D6 pin. To ensure

stable data transfers, you will also want to include a 4.7K resistor between the voltage source and the data pin. An example of how to gang multiple temperature sensors together, with resistor, looks like this:

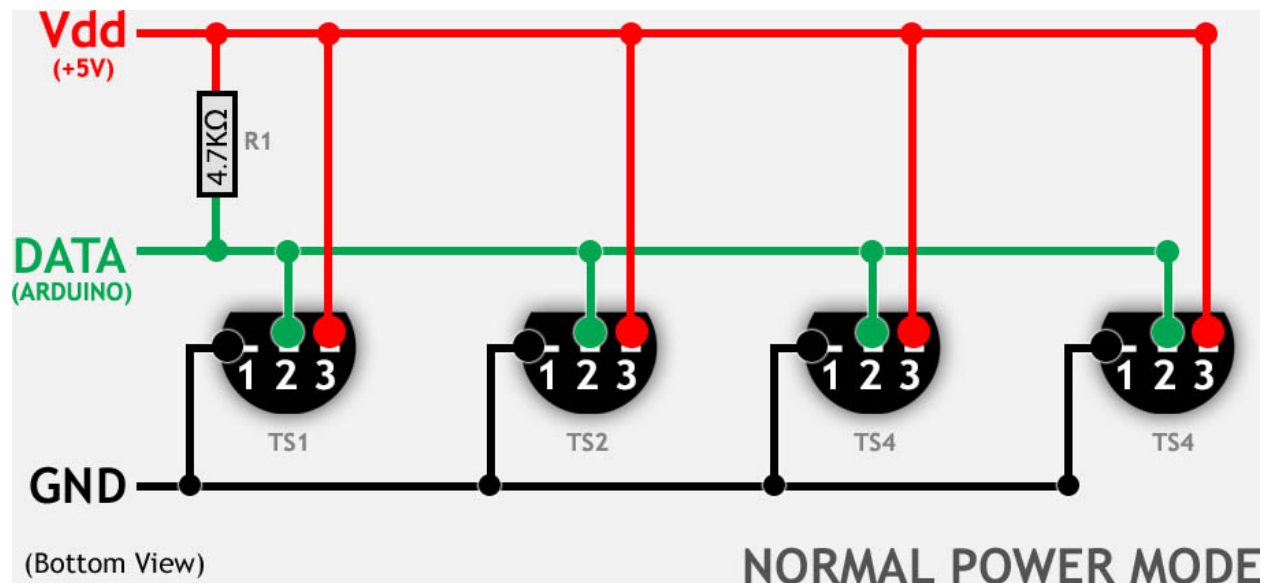


Fig. 1: Image courtesy of <https://www.tweaking4all.com/hardware/arduino/arduino-ds18b20-temperature-sensor/>

Each temperature sensor will be configured by its own hard-coded device address once it is connected to Fermentrack. You should see all connected temperature sensors listed under either Available Devices or Installed Devices.

To install an available temperature sensor, simply choose the device function (Chamber Temp, Room Temp, Beer Temp, etc.) and click “Assign”. Your BrewPi controller will configure the device appropriately.

4.3.3 Other Options (Chamber fan, chamber light, etc.)

While the BrewPi controller firmware allows for other options to be selected such as chamber fan, chamber light, etc. these options don’t actually do anything and should be ignored.

Adding a Gravity Sensor

A major benefit of the Fermentrack framework is that gravity sensors are integrated as a central feature of the fermentation system, on an equal level with the temperature sensors used to control the fermentation process. Similar to Fermentrack's agnostic view towards BrewPi controllers (any controller using the BrewPi firmware can be integrated), Fermentrack provides the ability to integrate a variety of gravity sensors.

5.1 Tilt Setup

The [Tilt Hydrometer](#) is a Bluetooth specific gravity sensor which operates by measuring the angle of a sensor floating in your fermenting beer. Fermentrack supports all versions of the Tilt Hydrometer both on a standalone basis and in conjunction with a BrewPi-based temperature controller.

5.1.1 Connecting a Tilt to Fermentrack

Tilt Hydrometers can be connected to Fermentrack either through a [TiltBridge](#) Tilt-to-WiFi bridge or using a Bluetooth adapter connected to the device running Fermentrack (including the built-in Bluetooth adapter on Raspberry Pi 3B or newer).

Bluetooth

To utilize the Bluetooth adapter connected to the device running Fermentrack simply place the tilt near the device at an angle (not vertical or horizontal) to wake it up. After a few seconds the device will begin broadcasting and can be detected within Fermentrack. Then, to add the device:

1. From the main Fermentrack dashboard, click on "Add new gravity sensor"
2. Click on the Tilt icon
3. Enter a name for the Tilt hydrometer you are adding, select the temperature units you wish to use, and select the color of the circuit board on the Tilt. You also want to specify if you are using a direct Bluetooth connection (default, included) to your Pi, or using a TiltBridge bluetooth to WiFi adapter (requires additional hardware).

You will return to the dashboard and Fermentrack will notify you that a new tilt sensor was added.

TiltBridge

Todo: Write the TiltBridge instructions

5.1.2 Tilt Settings

After adding the Tilt to Fermentrack, you are returned to the Gravity Sensors dashboard. When first added the Tilt is running in unassociated mode in Fermentrack, meaning it is handled as its own device - not integrated with a BrewPi controller. Clicking on the device will bring you into a screen where you can further configure and customize that Tilt sensor. From this page, you can [Manage Sensor](#), [Load Prior Log](#), and [Assign Device](#).

Manage Sensor

From the sensor dashboard, click on Manage Sensor. This page will show you the current settings running on the Tilt sensor, allow you to add manual calibration points, as well as allow you to perform a [Guided Calibration](#). The B, C, and D coefficients shown are automatically applied/updated after running a guided calibration. You can also remove this sensor from Fermentrack at the bottom of the page.

Load Prior Log

This button allows you to view prior logs generated by the Tilt sensor. This will bring up a window with available logs. You can click on the log name to view the graph in the browser, or click “View Full CSV” to download the csv file that holds the raw data for the graph. You can also choose “Delete” to delete the log. Viewing a previous graph does not stop Fermentrack from collecting data from the Tilt on the active log. You can return to the active log by clicking the “Load Prior Log” button and then “Return to Active Log”.

Assign Device

See [Integrating with BrewPi](#)

5.1.3 Logging/Visualization

To start logging data points from your Tilt, click below the device name in the orange box. Enter a name for your log in the text field and click “Create Log & Start Logging”. This will create a blank graph where your data points will begin to populate after a few moments of being collected. The page does not refresh automatically so you will have to manually refresh to see updates.

Add Data Point

This button allows you to manually add a data point to the graph. What you enter here will appear on the graph at the time you enter it. You can use these data points to annotate times that you perform specific actions, such as racking to secondary, adding dry hops, etc.

Control Logging

This button will give you the option to stop logging data and return you to the dashboard.

5.1.4 Guided Calibration

This section should be prefixed by stating that the Tilt sensor is very well calibrated out of the box and these steps are completely optional. This could be useful if you suspect your Tilt of falling out of calibration and want to reconfirm accuracy.

Although Tilt devices seem to output specific gravity readings, they work by internally measuring the angle (tilt) of the device and applying an equation to convert this angle to the expected specific gravity. While we do not have the ability to edit or calibrate this conversion equation directly, we can apply a correction factor to the readings it provides to add additional accuracy to the specific gravity we log. For the mathematically inclined, this is done by determining the coefficients for an equation of a line that best fits your Tilt's measurements ($bx^2 + cx + d$).

This process will guide you through creating sugar/water mixtures of known specific gravity and then seeing what specific gravity the Tilt reports from each. By the end of this process you will have collected 5 data points which Fermentrack will then use to derive your Tilt's conversion function. This data will be used to mathematically determine a coefficient to apply to your Tilt which result in the most accurate reading possible for future use.

You will need a few pieces of equipment to complete this process:

- 4 lbs (2 kg) White Table Sugar
- Water
- 2 cup (500 ml) measuring cup (for mixing)
- 2+ gallon (or 6+ liter) bucket (does not need to be sanitized)
- Tilt Hydrometer
- Gram scale

Once you have assembled these supplies, click "Begin guided calibration".

5.1.5 Integrating with BrewPi

To integrate your Tilt logging with a BrewPi controller:

1. Select the "Assign Device" button from the Dashboard view
2. From the dropdown list, select the name of the controller you wish to associate the Tilt sensor with
3. Click "Attach sensor to controller"

Now, to view your Tilt sensor readings, navigate to the BrewPi controller dashboard. You will have to restart a log if you had one running before associating the Tilt with the BrewPi sensor. Now, any wort you ferment with this controller will incorporate the Tilt's temperature and gravity readings onto your graph. Once the Tilt (or any gravity sensor) is attached to a BrewPi controller, that controller dashboard will become the main method with which to interact with the Tilt, specifically for things like logging.

5.1.6 Troubleshooting Tilt Support

Tilt Hydrometer support relies on a number of components beyond those used for other functions in Fermentrack, and as a result is particularly sensitive to changes in the program environment on the device on which Fermentrack is

installed. Testing has been added to Fermentrack to help diagnose some of these environmental issues if they happen to impact an installation.

Fixing Missing System Packages

If there are system packages missing, you will unfortunately need to fix them manually. For Raspberry Pis running Raspbian, here is how to fix this issue. For other OS's, please adapt these instructions as necessary

1. Log into your Raspberry Pi via as the *pi* user
2. Type `sudo apt-get update` and allow the package system to update
3. Type `sudo apt-get upgrade` and follow the prompts to upgrade all installed packages
4. For each missing package identified by the test script, type `sudo apt-get install -y {package name}`
5. Allow each package to install. Repeat the previous step for all missing packages.

Fixing Missing/Incorrect Python Packages

Although all Python packages should be automatically installed as part of the installation script, it is possible that packages come out of sync for a variety of reasons. If you are missing packages they will need to be installed for Fermentrack to properly interface with your Tilt.

A manual refresh of the Python packages can be triggered from the GitHub upgrade page without updating Fermentrack from GitHub. To trigger a refresh:

1. Log into Fermentrack
2. Click the 'gear' icon in the upper right hand corner of the page
3. Click 'Update from GitHub'
4. Click the 'Update/Install Missing Python Packages' button

5.2 iSpindel Setup

Todo: Write the iSpindel Sensor Setup page

5.3 Manual Sensor Setup

Todo: Write the Manual Sensor Setup page

CHAPTER 6

Adding a TiltBridge

TiltBridge is a standalone piece of hardware which is designed to allow your Tilt hydrometer to connect directly to WiFi without the need for an old cell phone, tablet, or Raspberry Pi. TiltBridge remains connected 24/7 and allows you to keep logging your specific gravity - even when you aren't next to your fermenters. It is cheap, easy to use, and open source.

This can also be particularly useful if your Tilt hydrometer is inside a steel conical or thickly insulated fridge, where the Bluetooth signal may have a hard time propagating to your Raspberry Pi. When placed near the Tilt hydrometer, the Tiltbridge can rebroadcast the Tilt's relatively weak bluetooth signal as a stronger WiFi signal direct to your WiFi router, allowing you to place your Raspberry Pi receiver further away from your fermentation unit.

6.1 TiltBridge Setup

Todo: Finish the Tiltbridge Setup page

Changelog, Licensing, and Development

7.1 Fermentrack Architecture

Todo: Rewrite/fix architecture.rst to match the mkdocs/markdown version

The Fermentrack stack is based on a front end application, a controller, and a firmware running on the device that handles reading temperatures, switching cooling and heating etc. Everything but the firmware part is running under a process manager which takes care of launching the front end and brewpi.py controller scripts.

![[Fermentrack Architecture]](img/fermentrack.png)

See [components](components.md) documentation for links and licenses.

The webserver nginx and chaussette WSGI server

Used to proxy http requests to chaussette over WSGI to the Fermentrack django application.

cron

Used to start the Fermentrack stack, it starts the Circus process manager via a @reboot job, it also checks the status of circus every 10 seconds, if it not running it will start it. All this is handled by a script: *updateCronCircus.sh*

Supports the following arguments: *{start|stop|status|startifstopped|add2cron}* where:

- *start* - will start circusd and all the services
- *stop* - will quit circusd and all processes (note it would be started again in 10 minutes)
- *status* - will output a status of all processes running (see below)
- *startifstopped* - will start the process manager if stopped (called from cron every 10 minutes)
- *add2cron* - if crontab entries are missing, it will add them back.

Crontab entries added with *add2cron*:

```
@reboot ~/fermentrack/brewpi-script/utils/updateCronCircus.sh start
*/10 * * * * ~/fermentrack/brewpi-script/utils/updateCronCircus.sh startifstopped
```

Example *status* output:

```
$ ~/fermentrack/brewpi-script/utils/updateCronCircus.sh status Fermentrack: active brewpi-spawner: ac-
tive circusd-stats: active dev-brewpi1: active
```

The process manager *circus*

Fermentrack is started at boot with the help of cron (see *cron*), the process manager handles all the different processes needed by Fermentrack.

- **Fermentrack** - The django application (web interface) runs under chaussette
- **brewpi-spawner** - An internal Fermentrack process for spawning controller scripts for controlling controllers like brewpi-esp8266.
- **circusd-stats** - An Internal circus process for stats, not used yet.
- **dev-brewpi1** - Is a controller script spawned by brewpi-spawner, handing a controller.

Circus documentation can be found [here](#).

Logging

- **Circus process manager logs:**
 - `/home/fermentrack/fermentrack/log/circusd.log`
- **Controller script (brewpi.py) log:**
 - `/home/fermentrack/fermentrack/log/dev-[name]-stdout.log`
- **Controller script (brewpi.py) error/info log:**
 - `/home/fermentrack/fermentrack/log/dev-[name]-stderr.log`
- **Controller script spawner:**
 - `/home/fermentrack/fermentrack/log/fermentrack-brewpi-spawner.log`
- **Fermentrack django application:**
 - `/home/fermentrack/fermentrack/log/fermentrack.log`

Logs are rotated every 2MB and the last 5 are saved with a number suffix.

7.2 Changelog

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](<http://keepachangelog.com/>) because it was the first relatively standard format to pop up when I googled “changelog formats”.

7.2.1 [Unreleased] - Additional Bugfixes

Changed

- Changed BrewPi-Script process manager to log data points via an HTTP API rather than directly querying the database

- Add additional cleanup steps to BrewPi-Script process manager
- Clarify links to BrewPi Process Spawner logs on help page

Fixed

- Resolved issue that would prevent the “invert” flag from being properly applied to BrewPi-ESP devices running recent firmware

7.2.2 [2023-08-20] - Additional BrewPi-Script Bugfixes

Changed

- Changed BrewPi-Script process manager to get list of devices via an HTTP API rather than directly querying the database
- Force BrewPi-Script process manager to restart in certain cases where a deadlock is detected

7.2.3 [2023-07-04] - BrewPi-Script Bugfixes

Changed

- Changes the message when a BrewPi device cannot be reached
- Only allow BrewPi-Script to check for profile updates once every 30 seconds
- Prevents two BrewPi devices from being able to have the same cached IP address

Fixed

- Resolves an error that would cause debugging connections to controllers to periodically raise an exception
- Resolves an error where a missing schema on a generic push target would not get corrected to <http://>
- Allow GravityLog.device to be None in backups, fixing errors that could prevent backups from being properly generated
- BrewPi-Script processes are now launched in a manner that allows StopIteration exceptions to be properly handled

7.2.4 [2023-06-23] - ESP32 Support, TiltBridge Jr & new BrewPi-Script

NOTE - This is the first version of Fermentrack that explicitly DOES NOT support “dockerless” installation. Installation is intended to be via the official Fermentrack images on Docker Hub.

Added

- BrewPi-Script instances are now controlled via a custom process manager
- Officially added arm64v8 to supported platforms
- Added support for new versions of the BrewPi Firmware for certain ESP-based BrewPi Controllers
- Added support for extended settings on new BrewPi Firmware for certain ESP-based BrewPi Controllers

- Added board types for ESP32, ESP32-C3, and ESP32-S2
- Added examples for HTTPS support (Thanks @HuggableShark)
- Added UUIDs to most exportable objects to allow for easier import/export
- Added “TiltBridge Jr.” (new Tilt bluetooth daemon) support

Changed

- Removed Circus support in favor of managing processes via Supervisor
- Removed libzmq requirement
- Removed environment tests for Tilt hydrometers (Obviated via Docker)
- Switch to use latest LTS version of Django
- Update to Python 3.9
- Changed most references to ESP8266 to reference ESP32 as appropriate
- Rewrote backup & restore functions to no longer rely on Django management code
- Changed restoration of legacy backups to explicitly utilize the staging folder
- Removed GitHub upgrade workflow from UI

Removed

- Removed Fermentrack-specific Bluetooth daemon - now handled by TiltBridge Jr.

7.2.5 [2022-04-05] - Last Dockerless Version & Backup & Restore

NOTE - This is the last version of Fermentrack that will explicitly support “dockerless” installation. All future installations are encouraged to use (and all existing users are encouraged to migrate) to docker-based installations of Fermentrack to ensure compatibility going forward.

Added

- Backup & Restore functionality (Gear Menu > Manage Backups)

Changed

- Removed “upgrade” notice from non-docker installs
- Added warning to non-docker installs about a lack of future support

Fixed

- Adjusted elements that were preventing dark mode from being as dark as it could be (Thanks @carjed)

7.2.6 [2021-12-08] - Compatibility Enhancements

Added

- i386 (i686) build target for Docker images
- Custom color scheme support (dark mode!) (Thanks @calandryll!)

Changed

- Change to prefer caching IPv4 addresses in BrewPiDevice.wifi_host_ip
- Removed PyBluez requirement to enable wider platform support

Fixed

- Properly allow blanks for BrewPiDevice.wifi_host_ip

7.2.7 [2021-04-05] - Docker Support

Added

- Added support for Docker installations
- Added environment/container version number to GitHub page
- Tilt “version” is now displayed on the Tilt Configuration page (Bluetooth only)
- Tilt battery level now shows on the Tilt Configuration page (Bluetooth v3/Tilt Pro only)
- “Last Check-in” time now added to Tilt configuration page (Bluetooth only)
- Added link to view Huey logs inside the Fermentrack UI
- Added link to view Circusd logs inside the Fermentrack UI

Changed

- Removed instances where BrewPi-Script would write to the database
- Adjusted feedback loop for Circus to eliminate a potential race condition with transactional databases
- Redesigned available firmware list to reduce confusion
- Stale gravity check-in points will now not be displayed in the gravity dashboard panels
- Added support for latest TiltBridge firmware
- Remove unimplemented “TCP Socket” external push option
- Added Beer Setting, Fridge Setting, and Controller State to generic external push targets
- Upgraded TiltBridge support for v1.0.0 TiltBridges (earlier TiltBridge versions will now require manual configuration)
- External push target attempts now log to huey stdout
- Added battery to the GenericPushTarget message for pushed iSpindel devices

Fixed

- Fermentrack now works with properly transactional databases (e.g. Postgres)
- Resolved issue causing false failures of the connectivity test (Thanks postalbunny!)
- Fixed issue preventing renaming of BrewPi controllers
- Dashes now allowed in TiltBridge mDNS IDs
- Corrected issue where iSpindel data couldn't be loaded if a data point wasn't available in Redis

7.2.8 [2020-12-19] - Tilt Pro

Added

- iSpindel temperature readings can now be calibrated for accuracy
- Add properly sized app icon for apple devices

Changed

- Added support for the new Tilt Pro
- Updated Sentry target
- Round iSpindel readings to four decimal places

Fixed

- Updated requirements to support the new pip

7.2.9 [2020-11-07] - Temp Profile Tweaks & Docker Prep

Added

- Added optional “notes” field to fermentation profile models
- Added error message when attempting to create a point with an invalid temp/ttl in a temp profile
- Added support for configurable gravity units (e.g. plato, specific gravity)

Changed

- Changed link to sqlite database to allow for a subdirectory in Docker installs
- Fermentation profile points can now be deleted for in-use fermentation profiles
- Remove Python 3.7 warning (everyone should have upgraded by now)
- If a Grainfather, Brewfather, or generic push target logging URL doesn't begin with <http://> it is now automatically appended

Fixed

- Bug causing errors when enabling beer profile mode
- Can now properly push to BrewFather when a fully populated temperature controller isn't linked
- Don't prompt new installs to run the script to fix old sqlite files

7.2.10 [2020-08-22] - Bugfixes**Added**

- Added pin definitions for esp32 BrewPi firmware
- Added Linux networking capability test to Tilt connectivity test suite

Changed

- Added ability for BrewFather push target to push temps from BrewPi temp sensors

Fixed

- Fixed bug that prevents reloading of cached controller objects
- Properly detect/force temperature conversion for iSpindel

7.2.11 [2020-06-05] - Django 3.0 Support**Added**

- Added button to controller settings page to restart a controller

Changed

- Updated code to be Django 3.0 Compatible
- Changed from Django 1.11 to Django 3.0
- Toggling display of a graph line on a temp controller's dashboard now clears the data point displayed in the legend
- Refactored brewpi-script to accept device IDs rather than names

Fixed

- Properly catch exception when Redis test cannot connect to server
- Gravity and gravity temp colors when graphed on temp controller graphs will now display the correct color in the legend
- Links to CSVs from the beer log list now properly generate if the CSV exists
- Correct error detection/logging when a temp controller with an attached gravity sensor attempts to log before the gravity sensor logs its first point

- Properly check that a temperature setting is provided when setting a Beer or Fridge Constant mode for temp controllers
- Temp controller name uniqueness checks are now properly enforced in all add controller workflows

7.2.12 [2020-04-11] - Bugfixes & Tilt Troubleshooting

Added

- Added explicit support for LBussy's BrewPi-Remix I2C Board
- Exposed upgrade.log from the help screen
- Store the exact last time that a message was received from a Tilt to Redis
- Add sentry support to tilt_monitor_aio.py
- Added "debug" scripts for bluetooth Tilt connections
- Added TiltBridge connection settings to Tilt management page

Changed

- Removed legacy Python 2 code
- Reduced gravity sensor temp precision to 0.1 degrees
- Locked pybluez, aioblescan, and redis versions to prevent undesired format changes going forward

Fixed

- Fix display of TiltBridge mDNS settings on Tilt settings page

7.2.13 [2020-02-17] - Improved ESP32 Flashing Support

Added

- Added support for flashing a bootloader and otadata partition to ESP32 devices

Changed

- SPIFFS partitions can now be flashed to ESP8266 devices

7.2.14 [2020-02-15] - ThingSpeak and Grainfather Support

Added

- Added support for pushing data to ThingSpeak (thanks @johndoyle!)
- Added support for pushing data to Grainfather (thanks @mp-se!)

Changed

- Gravity sensors attached to BrewPi controllers will now send those controller's temps to Brewfather
- An explicit error message will now be displayed when a user attempts to manually access the ispindel endpoint

Fixed

- Fixed where Fahrenheit readings coming from an iSpindel could be improperly reconverted to Fahrenheit
- Lock temperature display on dashboard panels to one decimal place
- Allow updates to controller settings when controller name isn't changing (for real this time)
- Fix bug that would default all Tilts to 'Bluetooth' even when a TiltBridge was selected
- Fixed issue where Tilt readings were not being properly decoded (Thanks NecroBrews!)
- Fixed issue where dashboard panels were not being updated (Thanks NecroBrews!)

7.2.15 [2019-12-15] - Brewer's Friend, Brewfather, and MacOS BLE Support

Added

- Added support for pushing gravity sensor data to Brewer's Friend
- Added support for pushing gravity sensor data to Brewfather
- Added BLE support for MacOS (thanks corbinstreehouse!)

Changed

- Adding an external push target now triggers data to always be sent within 60 seconds regardless of push frequency

Fixed

- Disable "View Full CSV" button if gravity/beer logs don't exist
- Properly cleanse booleans when changing site settings for Constance
- Allow updates to controller settings when controller name isn't changing
- Remove requirement for TiltBridge value in the TiltBridge model definition
- Ignore Tilt diagnostic codes that cause erroneous temperature/gravity readings

7.2.16 [2019-10-24] - Miscellaneous Bugfixes

Fixed

- Prompt for reconfiguration if pin/address assignment fails on BrewPi Controller
- Validate device name uniqueness when adding a new BrewPi Controller
- Warn user when empty temperature is submitted alongside Fridge or Beer Constant mode

- Return debugging info when a connection to a WiFi BrewPi Controller is refused
- Properly handle errors in the first step of the firmware flash process
- When logging beer points on a gravity-enabled log, make sure the gravity sensor exists (or stop logging)
- Properly handle empty TiltBridge check-ins
- Before adding a Tilt that uses a TiltBridge, make sure the TiltBridge exists
- Return an error if a TiltBridge doesn't pass properly formed JSON
- Enforce uniqueness of a Beer name/logging device combination when the Beer is created
- Cause brewpi-script to terminate if the controller returns invalid control settings
- Return empty JSON for annotations if Beer doesn't exist

7.2.17 [2019-03-31] - TiltBridge Support

Added

- Added support for TiltBridge Tilt-to-WiFi devices

Changed

- Removed Hex SHA display on GitHub update
- Tweaked backup count for log files to reduce clutter

Fixed

- Fixed hostname lookup in connection debug when running on a nonstandard port
- Fixed multipart firmware flashing
- Remove Git branch switching prompt during initial setup
- Remove links to defunct Tilt logs
- Fixed OneWire address display on BrewPi "Assign Pin/Device" page
- Fix link to "load beer log" modal on device dashboard when no beer is loaded

7.2.18 [2019-03-17] - Firmware Flash Changes

Added

- Added support for flashing multi-part firmware (eg partition tables)

Changed

- Updated firmware_flash models to support additional device families
- Changed to version 2 of firmware_flash models

7.2.19 [2019-02-17] - External Push (Remote Logging) Support

Added

- Fermentrack can now periodically “push” readings out to an external device/app
- Added “new control constants” support for “modern” controllers

Fixed

- Explicitly linked Favicon from template
- Fixed BrewPi-Script error when attempting to use feature not available in Python 3.4
- Properly catch error in BrewPi-Script when pidfile already exists
- Added filesize check for gravity sensor & brewpi-device logfiles
- Add support for temperature calibration offsets

7.2.20 [2019-02-17] - External Push (Remote Logging) Support

Added

- Fermentrack can now periodically “push” readings out to an external device/app
- Added “new control constants” support for “modern” controllers

Fixed

- Explicitly linked Favicon from template
- Fixed BrewPi-Script error when attempting to use feature not available in Python 3.4
- Properly catch error in BrewPi-Script when pidfile already exists
- Added filesize check for gravity sensor & brewpi-device logfiles
- Add support for temperature calibration offsets

7.2.21 [2018-10-24] - Tilt Monitor Refactoring

Changed

- The Tilt Hydrometer monitor now uses aioblescan instead of beacontools for better reliability
- Added support for smaller screen sizes

Fixed

- Tilt Hydrometers will now properly record temperatures measured in Celsius

7.2.22 [2018-08-05] - Gravity Refactoring

Added

- DS18B20 sensors can now have temperature offsets added to each reading to correct for calibration errors
- ESP8266 controllers can now have their WiFi settings reset via the “manage sensor” web interface
- Control constants form now supports both “new” (OEM BrewPi) and “old” (“Legacy” branch) control constants
- Tilt hydrometers can now have their specific gravity readings calibrated
- “Heat/Cool State” will now be shown on temperature graphs
- Fermentrack logo added as favicon

Changed

- The iSpindel endpoint can now be accessed at either /ispindel or /ispindle
- Specific gravity will now be shown on graphs with 3 decimal places
- Beer log format has been changed to add state information

Fixed

- Removed constant LCD polling for “modern” controllers
- Gravity support will now be properly disabled when the correct flag is set at setup
- iSpindel devices that do not report all ‘extras’ will no longer throw errors when reporting gravity

7.2.23 [2018-04-27] - “v1.0 release”

Added

- Added fermentation controller “Manage Device” page
- Upgrades are now logged to upgrade.log
- Controller “stdout” and “stderr” logs are now saved/accessible
- Support for serial devices
- Support for Arduino-based devices
- Support for in-app git branch switching
- Autodetection of serial devices
- Huey (delayed/scheduled task) support (currently unused)
- Controllers connected via serial can now have their serial port autodetected using the udev serial number
- Beer profiles are now displayed in graph form
- Firmware can now be flashed to new Arduino & ESP8266-based controllers from within the app
- Preferred timezone can now be selected for use throughout Fermentrack
- Beer log management (deletion/downloading)

- Added configuration options for graph line colors
- Graph lines can be toggled by clicking the icon in the legend
- Added support for specific gravity sensors
- Added support for Tilt Hydrometers
- Added support for iSpindel specific gravity sensors

Changed

- Inversion flag for installed devices is now shown on the “configure pins/sensors” page
- Form errors are now displayed on “configure pins/sensors” page
- Beer logs are no longer deleted along with the parent device (but they will become inaccessible from within Fermentrack)
- GitHub updates are no longer triggered automatically by visiting the update page, and must now be manually triggered by clicking a button
- The IP address of a BrewPiDevice is now cached, and can be used if mDNS stops working
- At end of a fermentation profile the controller will now be switched to beer constant mode
- All data points are now explicitly recorded in UTC
- Added icon to graph legend to display line color
- Updated to Django v1.11 (Long term support version)
- Changed from supporting Python 2 to Python 3

Fixed

- Inversion state no longer improperly defaults
- Minimum graph size adjusted to account for smaller displays
- Changed on_delete behavior to allow deletion of fermentation controllers
- Git update check will now properly wait between checks if up to date
- GIT_UPDATE_TYPE of ‘none’ will now properly disable update checks
- BrewPi controllers now accept unicode names
- “View Room Temp” link on Dashboard now functions
- Room temp now included in legend for graphs

7.2.24 [2017-03-17] - “v0.1 release”

Added

- First release!

7.3 Included Components & Licensing

Fermentrack is licensed under the MIT License, the terms of which can be read here: [License](#)

Fermentrack contains a number of JavaScript, Python packages, CSS, and other components to help provide additional functionality which are provided under their own licenses. Some of these packages are listed below.

7.3.1 Python Packages

In addition to Django, this app utilizes a number of Python packages. These packages include:

Package	License
Django	BSD 3 Clause
configobj	BSD 3 Clause
pyserial	BSD 3 Clause
huey	MIT (Expat)
sentry-python	BSD 2 Clause
django-constance	BSD 3 Clause
GitPython	BSD 3 Clause
pytz	MIT (Expat)
redis	MIT (Expat)
zeroconf	LGPL v2
pyudev	LGPL v2.1
circus	Apache Public License v2
circus-web	Apache Public License v2
chaussette	Apache Public License v2
pid	Apache Public License v2
aioblescan	MIT (Expat)
django-envron	MIT (Expat)
whitenoise	MIT (Expat)
psycopg2	LGPL v3
numpy	BSD 3 Clause
scipy	BSD 3 Clause

7.3.2 JavaScript Packages

Fermentrack provides some of its functionality using JavaScript. Some of the third party JavaScript packages used within Fermentrack include:

Package	License
jQuery	MIT (Expat)
vue.js	MIT (Expat)
Dygraph	MIT (Expat)
Moment	MIT (Expat)
Moment Timezone	MIT (Expat)
Respond.js	MIT (Expat)
html5shiv.js	MIT (Expat)

7.3.3 UI (CSS, Fonts, etc.) Packages

In addition to Python and JavaScript packages, Fermentrack utilizes a handful of third party images, icon packs, CSS files, and fonts in rendering the user interface. Some of these include:

Package	License
FontAwesome (Font)	SIL OFL 1.1
FontAwesome (CSS)	MIT (Expat)
Bootstrap	MIT (Expat)
FlatUI	MIT (Expat)
5x8 LCD Font	SIL OFL 1.1

7.3.4 Separate Applications

Fermentrack is designed to help install or manage communications with certain key applications. These applications are not incorporated into Fermentrack but may be bundled with the software as a convenience.

Package	License
BrewPi-Script	GPL v3
BrewPi Firmware	GPL v3 (and potentially others)
Avrdude	GPL v2
Fuscus	GPL v3
esptool.py	GPL v2

7.4 Contributing to Fermentrack

Want to help? Great! There are a number of ways you can get involved in this project. To help you get started, some areas in which you can contribute are listed below along with additional resources for getting started.

7.4.1 Reporting Bugs/Issues

Found a bug or issue (or have a suggestion for how to improve)? Awesome!

The best way to reach out is to either open an issue on [GitHub](#) or discuss your issue or idea in the HomeBrewTalk thread (or - ideally - both). Both of these are actively monitored, and will help to keep track of progress towards a resolution.

7.4.2 Developing Fermentrack

Interested in helping develop Fermentrack (or improving its look & feel)? Fantastic - Fermentrack is an open source project, and all help is welcome.

Fermentrack is a Python-based application which uses the Django framework. It is open source and is managed on GitHub. To help you get started, take a look at the developer documentation located within [Changelog](#), [Licensing](#), and [Development](#). After reading, if you have questions don't hesitate to reach out. I do recommend reaching out (either on HomeBrewTalk or by raising an issue on GitHub) prior to developing a feature that you ultimately would like merged back in.

7.4.3 Documenting Fermentrack

A project like Fermentrack is only as good as its documentation. Documentation for Fermentrack is (currently) written in [reStructuredText](#) with an eye towards [Sphinx](#). The source for the documentation is located in the `docs/source` folder in the Fermentrack repo on [GitHub](#).

If you are familiar with GitHub, pull requests that include documentation fixes are always welcome. If not, reach out on GitHub or the HomeBrewTalk forums and we'll be happy to help get things updated.

7.5 “Push” Support

Although Fermentrack is focused on the “fermentation” phase of your brewing operation, Fermentrack is designed to integrate with your brewing operation as a whole. To support the use of data collected by Fermentrack in other applications, Fermentrack allows for data to be “pushed” on a periodic basis via HTTP requests (and will - in the future - support pushing via TCP (sockets)).

7.5.1 Supported “Push” Targets

Fermentrack currently supports five push targets:

- **“Generic” Push Target** - Fermentrack’s “native” push format - Pushes both specific gravity & temperature data
- **Brewer’s Friend** - Pushes both specific gravity & temperature data associated with gravity sensors
- **Brewfather** - Pushes both specific gravity & temperature data associated with gravity sensors, as well as temperature data from BrewPi controllers with gravity sensors attached
- **ThingSpeak** - Pushes temperature data
- **Grainfather** - iSpindel push format - Pushes both specific gravity & temperature data from gravity sensors

Generic Push Target Format

The “generic” push target format is the one recommended for use by developers who are adding native Fermentrack support to their apps. This format contains temperature/gravity data collected by Fermentrack for each available specific gravity sensor and BrewPi controller.

This format is supported by the [Fermentrack Push Target app](#) for testing/development purposes.

```
{'api_key': 'abcde',
 'brewpi_devices': [{ 'control_mode': 'f',
                      'internal_id': 1,
                      'name': 'Legacy 2',
                      'temp_format': 'F'},
                    { 'beer_temp': 31.97,
                      'control_mode': 'f',
                      'fridge_temp': 36.26,
                      'internal_id': 2,
                      'name': 'Kegerator',
                      'temp_format': 'F'}],
 'gravity_sensors': [{ 'gravity': 1.247,
                      'internal_id': 1,
                      'name': 'Pinky',
                      'sensor_type': 'tilt',
                      'temp': 78.0,
```

(continues on next page)

(continued from previous page)

```

        'temp_format': 'F'},
    {'internal_id': 3,
     'name': 'Spindly',
     'sensor_type': 'ispindel',
     'temp': 86.225,
     'temp_format': 'F'}],
    'version': '1.0'}

```

Brewer's Friend Support

Fermentrack supports pushing data from specific gravity sensors to [Brewer's Friend](#) using the "Device Stream" API. To configure:

1. Log into Fermentrack and click the "gear" icon in the upper right
2. Click "Add Brewer's Friend Push Target" at the bottom of the page
3. Log into your Brewer's Friend account and go to [Profile > Integrations](#)
4. Copy the API key listed at the top of the page in Brewer's Friend (the string of letters/numbers)
5. Within Fermentrack, paste the API key you just copied into the "Api key" field
6. Set the desired push frequency and select the gravity sensor from which you want to push data
7. Click "Add Push Target"

Within 60 seconds, Fermentrack will begin sending data from your gravity sensor to Brewer's Friend. This data can be seen on the [Device Settings](#) page. (*Note* - You may need to click "Show All" on the right of this page to see the data for newly added devices)

Brewfather Support

Fermentrack supports pushing data from specific gravity sensors or temperature data from a brewpi to Brewfather using the "Custom Stream" API.

The following values in the Brewfather API is used;

```

{ "name": "",          // Will be either brewpi or gravitysensor
  "temp": 0,          // Temperature from gravity sensor or beer temp from attached_
  ↪ brewpi
  "aux_temp": 0,      // Fridge temperature from brewpi
  "ext_temp": 0,      // Room temperature from brewpi
  "temp_unit": "C"    //
  "gravity": 0,       // Gravity from sensor
  "gravity_unit": "G",
  "pressure": 0,      // not used
  "pressure_unit": "", // not used
  "ph": 0,           // not used
  "bpm": 0,          // not used
  "comment": "",     // not used
  "beer": ""         // Name of active beer log
}

```

To configure:

1. Log into Fermentrack and click the "gear" icon in the upper right

2. Click “Add Brewfather Push Target” at the bottom of the page
3. Log into your Brewfather account and go to [Settings](#)
4. At the bottom of the page, under “Power-ups” click the “switch” next to “Custom Stream”
5. Copy the Logging URL (starting with http and ending with a string of letters/numbers) listed under “Custom Stream”
6. Within Fermentrack, paste the Logging URL you just copied into the “Logging URL” field
7. From the drop down list, select either Gravity or BrewPi as source.
8. Set the desired push frequency and select the gravity sensor/brewpi from which you want to push data
9. Click “Add Push Target”

Within 60 seconds, Fermentrack will begin sending data from your gravity sensor to Brewfather. This data can be seen on the [Devices](#) page.

NOTE - If your gravity sensor is attached to a BrewPi controller, the temperature readings from that controller will be used instead of the ones from the gravity sensor.

ThingSpeak Support

Fermentrack supports pushing data from specific sensors to a ThingSpeak Channel. The Channel Speak API is fixed to receive fields in the channel, so the designation of each channel is already defined. This means that Field 1 is always Beer Name, Field 2 is Sensor Name, etc. To configure:

1. Log into Fermentrack and click the “gear” icon in the upper right
2. Click “Add ThingSpeak Push Target” at the bottom of the page
3. Now log into your ThingSpeak account and on the My Channels Page select New Channel
4. Enter the data below in

```
Name - Give your Channel a Name
Description - Give your Channel a Description
Field 1 - Beer Name
Field 2 - Sensor Name
Field 3 - Temp Format
Field 4 - Beer Temp
Field 5 - Fridge Temp
Field 6 - Room Temp
Field 7 - Beer Gravity
```

Feel free to fill out the optional elements but only the ‘field’ values above are sent. The values entered are just labels for the data sent and can be customised. For example you can change ‘Beer Temp’ to ‘My Beer (°C)’.

1. At the bottom of the page, select ‘Save Channel’
2. Copy the “Write API Key” from the “API Keys” section
3. Within Fermentrack, paste the API Key you just copied into the “API Key” field
4. Set the desired push frequency and select the gravity sensor from which you want to push data
5. Click “Add Push Target”

Within 60 seconds, Fermentrack will begin sending data from to the ThingSpeak Channel. This data can be seen on the ThingSpeak ‘Private View’ tab in the channel page.

Grainfather Support

Fermentrack supports pushing data from specific gravity sensors (Gravity & Temperature) to Grainfather using the brew tracking API. To configure:

1. Log into your Grainfather account and select Equipment.
2. Add a Fermentation device and select iSpindel as device type. Fermentrack will push data in this format independent of what your device is. Copy the logging URL.
3. The second thing you need to do is to go to an active brew and link the device to a brew session. This is done under the headline fermentation tracking and the function “Add Tracking Device”. Make note of the Name value (this is the brew ID).
4. Log into Fermentrack and click the “gear” icon in the upper right
5. Click “Add Grainfather Push Target” at the bottom of the page
6. Within Fermentrack, paste the Logging URL you just copied into the “Logging URL” field and enter the name (brew id) under the “gf_name” field.
7. Set the desired push frequency and select the gravity sensor from which you want to push data
8. Click “Add Push Target”

Within 60 seconds, Fermentrack will begin sending data from your gravity sensor to Grainfather. This data can be seen in your Grainfather account under Equipment or the Brew Session.

7.5.2 Implementation Notes

Push support within Fermentrack is implemented through the use of a “helper script” which currently is launched once every minute. The helper script polls the defined push targets to determine which (if any) are overdue for data to be pushed, and - for those targets - then executes the push based on how those targets are configured. Fermentrack polls Redis for the latest available data point for specific gravity sensors, and polls BrewPi controllers for the latest data point directly. This data is then encoded based on the selected push format and sent downstream to the requested target.

Push requests are handled asynchronously. Due to the way that the polling script is implemented, push “frequencies” may be up to one polling cycle (currently 1 minute) later than expected. For 1 minute push cycles, this means that the actual frequency could be as high as 2 minutes.

7.5.3 Feedback

Push support was designed to support future applications that do not yet exist, and as such, may not be perfect for *your* application. That said, feedback is always appreciated and welcome. Feel free to reach out (HBT forums, GitHub, Reddit...) if you have something in mind that you’d like to integrate Fermentrack into, and don’t think the existing push options will quite work.

7.6 License

MIT License

Copyright (c) 2016–2020 John Beeler
Copyright (c) 2016–2020 Fredrik Steen

(continues on next page)

(continued from previous page)

Permission **is** hereby granted, free of charge, to **any** person obtaining a copy of this software **and** associated documentation files (the "**Software**"), to deal **in** the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

The above copyright notice **and** this permission notice shall be included **in all** copies **or** substantial portions of the Software.

THE SOFTWARE IS PROVIDED "**AS IS**", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Supported Hardware

Fermentrack supports both BrewPi-based temperature controllers as well as various specific gravity sensors. Support for each family of hardware varies, but is expanding with each release.

8.1 BrewPi Controllers

8.1.1 ESP8266-Based Controllers

Fermentrack was explicitly built to support ESP8266-based controllers regardless of connection method (WiFi or Serial).

For more information on ESP8266-based firmware, please refer to one of the following:

- [BrewPi-ESP8266 GitHub](#)
- [BrewPi-ESP8266 HomeBrewTalk Thread](#)

If connecting an ESP8266-based controller via serial, please note that by default Fermentrack will detect the USB serial number associated with your ESP8266 when initially configured, and will use that *instead of the specified serial port* to connect. For more information, read [About Serial Port Autodetection](#).

8.1.2 Arduino-Based Controllers

Fermentrack currently has native support for Arduino-based controllers connected via Serial (USB).

By default, Fermentrack will detect the USB serial number associated with your Arduino when initially configured, and will use that - instead of the specified serial port - to connect. For more information, read [About Serial Port Autodetection](#).

Bluetooth Support for BrewPi-Controllers

There is a third party project which looks to add bluetooth support for Arduino (and similar) controllers to BrewPi. While this project is not natively supported from within Fermentrack, at the end the project presents the BrewPi controller as a device connected via serial which allows it to be set up within Fermentrack via the Manual Setup workflow.

8.1.3 Native Python (Fuscus)

[Fuscus](#) is a project which implements the legacy (Arduino) BrewPi featureset directly on the Raspberry Pi with no need for an external controller.

As of now, serial connections are supported by Fermentrack, and therefore it is expected that Fuscus should be Fermentrack compatible. Fuscus support *has not been tested* and should be considered experimental.

Due to the nature of the serial ports used by Fuscus, the serial autodetection process cannot be used to set up a Fuscus-based controller. To set up, please follow the instructions in [Setting up a controller using the Advanced \(Manual\) Workflow](#).

Note: When setting up a Fuscus-based controller in the manual workflow, make sure to set “`prefer_connecting_via_udev`” to False. If it is set to true, BrewPi-Script may either not connect or connect to the wrong device.

Further down the development path are other features involving Fuscus such as direct installation and configuration management - though these are expected in v3 and beyond.

8.1.4 Spark Core/OEM Controllers

Currently, Fermentrack does not support Spark-based controllers. Support for Spark based controllers is planned, but will be implemented at a much later date. Once implemented, Fermentrack will support controlling both legacy (Arduino/Fuscus) and modern (Spark) controllers from the same installation.

8.2 Specific Gravity Sensors

8.2.1 Tilt Hydrometer

The [Tilt Hydrometer](#) is supported natively by Fermentrack which will assist with device setup. Tilt Hydrometers are easy to use, and can be installed either alongside or independent of a temperature controller. Natively, Tilt Hydrometers communicate via Bluetooth, however they can also be connected via the [TiltBridge](#) Bluetooth-to-WiFi adapter.

8.2.2 iSpindel

[iSpindel](#) devices are directly supported by Fermentrack which can assist with device configuration & calibration, as well as the initial flashing of the iSpindel firmware.

Frequently Asked Questions

9.1 Can I change the temperature format on a beer I've started logging?

No. To prevent inconsistency the log format is permanently set when logging begins to the temperature format associated with the device. If you would like to change the format and restart logging, do the following:

1. Update the temperature format in control constants to the desired format
2. Stop logging the existing beer
3. Start logging a new beer

9.2 Help - I forgot my Fermentrack login/password!

Thankfully, this is a pretty easy issue to overcome. Django provides the `manage.py` command line script which contains the `createsuperuser` command. For Docker-based installations, there is a script bundled alongside **'fermentrack-tools'** that leverages this command to create a new user. To use this script:

1. Log into your Raspberry Pi via SSH
2. Change to the `fermentrack-tools` directory (e.g. `cd fermentrack-tools`)
3. Run `./docker-create-superuser.sh`
4. Follow the prompts to create a new superuser account
5. Log into the Fermentrack admin panel and delete/modify the old account. The Fermentrack admin panel can be accessed through the `Settings` page (the gear in the upper right) and clicking the "Django Admin" button.

For non-docker installs, there are a few more steps but it's still pretty easy:

1. Log into your Raspberry Pi via ssh and switch to the user you installed Fermentrack to (generally this can be done with the command `sudo -u fermentrack -i` assuming you installed to the `fermentrack` user)
2. Change to the user's home directory (`cd ~`)

3. Enable the virtualenv (`source venv/bin/activate`)
4. Change to the Fermentrack directory (`cd fermentrack`)
5. Run the `createsuperuser` command (`./manage.py createsuperuser`)
6. Follow the prompts to create a new superuser account
7. Log into the Fermentrack admin panel and delete/modify the old account. The Fermentrack admin panel can be accessed through the `Settings` page (the gear in the upper right) and clicking the “Django Admin” button.

9.3 What happens to my beer logs/active profiles/other data if I change the Fermentrack “Preferred Timezone”?

Not much. To prevent this being an issue Fermentrack uses UTC (GMT) internally and converts times to your local timezone on the fly. Feel free to update your preferred timezone as you move, travel, or are otherwise inclined without worrying about how this might impact your existing logs or active profiles.

CHAPTER 10

Documentation To-Do List

This file is automatically generated, and contains a list of everything that has been noted as a “To Do” in the documentation.

If you’re looking for a way to help with the Fermentrack documentation, this is a great place to start. You can help by either:

1. Directly addressing & clearing items from the list below
2. Adding new items (using the `.. todo ::` directive) where additional documentation is needed.

Below are the items which need to be addressed:

Todo: Rewrite the Manual Workflow section to document the available options, etc.

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/dev/docs/source/controllers/c` line 88.)

Todo: Rewrite/fix `architecture.rst` to match the `mkdocs/markdown` version

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/dev/docs/source/develop/arch` line 4.)

Todo: Write the iSpindel Sensor Setup page

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/dev/docs/source/gravitysenso` line 6.)

Todo: Write the Manual Sensor Setup page

(The [original entry](#) is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/dev/docs/source/gravitysensor line 6.)

Todo: Write the TiltBridge instructions

(The [original entry](#) is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/dev/docs/source/gravitysensor line 36.)

Todo: Figure out how to include the text in about.rst on this page in the HTML version only

(The [original entry](#) is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/dev/docs/source/index.rst, line 55.)

Todo: Come back and re-add the video link once the video of the docker install is available.

(The [original entry](#) is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/dev/docs/source/installing/ins line 28.)

Todo: Finish the Tiltbridge Setup page

(The [original entry](#) is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/dev/docs/source/tiltbridge.rst line 19.)

Todo: Figure out how to include the text in about.rst on this page in the HTML version only
