

HestiaPi Owners Manual

HestiaPi Community

December 7, 2023

Contents

1 Software	2
1.1 Quick Installation	2
1.1.1 Prepare a new SD card	2
1.2 Manual Installation	2
1.3 Software Architecture	2
2 Hardware	2
2.1 Bill of Materials (BOM)	2
2.2 PCB Soldering	3
2.2.1 Instructions	3
2.2.2 Video	7
2.2.3 Hints and Tips	7
2.2.4 Troubleshooting	7
2.3 Printing the Case	7
2.3.1 Files	7
2.3.2 Filament	7
2.4 Wall Installation ONE	8
3 Usage	9
3.1 Touchscreen	9
3.2 Webpage	9
3.2.1 Basic UI	9
3.2.2 Paper UI	12
3.3 Mobile app	12
4 Configuration	12
4.1 First boot	12
4.2 Boot Sequence	13
4.3 Connect WiFi	13
4.4 Advanced Settings	13
4.4.1 Remote Temperature Sensors	13
4.4.2 Ruuvi Temperature Sensors	15
4.4.3 Easy Remote Access	16
4.4.4 Update Your DynDNS Automatically	16
4.5 Setting a Static IP Address	16
4.5.1 Configuring dhcpcd	16
4.5.2 Configuring Networking Service	17
4.6 Set up TLS	17
4.6.1 Static IP	18
4.6.2 DNS Servers	18
4.6.3 Obtain TLS certificate	18
4.6.4 Configure HestiaPi to use new Certificate	18
4.7 File Structure & Paths	19
5 FAQ	20
5.1 Configuration	20
5.1.1 Default SSH Username and Password	20

5.1.2	MQTT Configuration	21
5.1.3	How to Access My HestiaPi From Outside My House	21
5.2	Troubleshooting	21
5.2.1	Taps do not register on the touchscreen	21
5.2.2	Screen doesn't turn on	22
5.2.3	Screen is slow to respond to taps	22
5.2.4	How to edit files via SSH	22
5.2.5	Start OpenHAB2 in Debug Mode	22

1 Software

1.1 Quick Installation

To make it easier for new users, HestiaPi offers ready-to-burn image files for your SD card.

If you bought your HestiaPi with an SD card, skip this step.

1.1.1 Prepare a new SD card

With the image file downloaded, you need to use an image writing tool (we prefer Etcher from below links) to install it on your SD card. You can't simply copy-paste it. If you downloaded a ZIP version, unzip the .img file first before the next step.

For experienced Linux users, just use dd to write to the block device (SD card) and use the conv=fsync to make sure everything gets written. For a more detailed set of instructions, choose the right guide for your system below (courtesy of Raspberry Pi website – thanks):

- [Linux](#)
- [Mac OS](#)
- [Windows](#) (avoid if you can as people have reported issues flashing their card from Windows)

1.2 Manual Installation

For people who want to install everything step-by-step from their terminal we include all commands here. To be added...

1.3 Software Architecture

To be added...

2 Hardware

2.1 Bill of Materials (BOM)

This is the bill of materials to make your own HestiaPi. The AC/DC converter allows your HestiaPi to be powered by the HVAC unit, but it requires that your HVAC system has a common wire. If your system does not have a common wire (e.g., if it only has Red, Green, White, and Yellow wires), the AC/DC converter will not work. In that case, you will need a 5V/1A DC power supply, and your hestia will need to be plugged into a wall outlet.

The LCD headers need to be 15mm long, with 12mm of that being on the top of the board. Standard headers are not long enough to allow the LCD to clear the relays.

1. Printed Circuit Board – Send the [Gerber Files](#) to a shop like [JLCPCB](#) (or use them to etch your own board)
2. Raspberry Pi Zero W [AdaFruit](#) [Amazon](#)
3. 3.5 inch touch screen for Raspberry Pi [Amazon](#)
4. 4 relays capable of handling 24V / 2A – G3MC-202P-DC5 [Mouser](#)
5. At least 40 standard male 0.1" headers – [AdaFruit](#)
6. Reset switch – TL1105WF160Q [Mouser](#)
7. A six-post terminal block – TB003-500-P06BE [Mouser](#)
8. BME280 sensor – SEN0372 [Mouser](#)
9. Cable for BME280 sensor [Amazon](#)
10. LCD headers, 2x13 extra long headers (total length should be 15mm, 12mm from board to tip) [AdaFruit](#) (requires adjusting pins)
11. AC/DC converter – [Amazon](#)

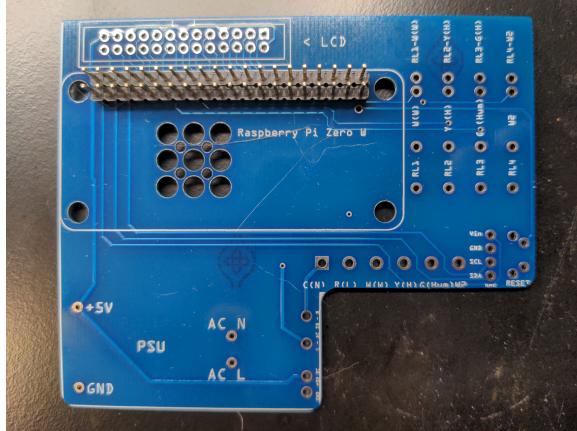


Figure 1: Headers for the pi, don't forget the reset pin like I did here!

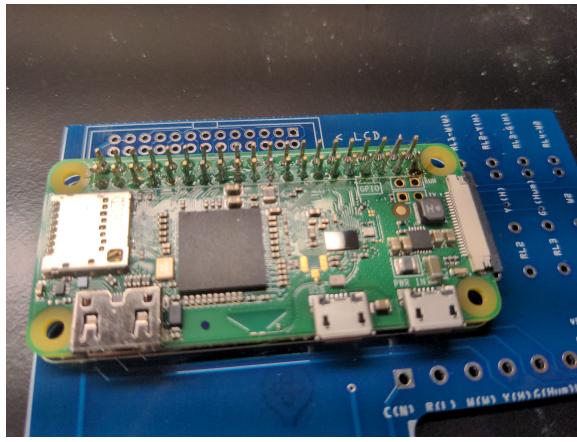


Figure 2: Pi soldered on, sans the reset pin that I had to go back and patch up later

2.2 PCB Soldering

2.2.1 Instructions

This guide describes the steps required to solder and assemble HestiaPi Touch ONE from parts. Assembly with the case and wall is covered in Section 2.4.

1. Solder standard headers onto PCB for the pi (figure 1) Make sure you don't miss the reset pin at the right, just below the block of 2x20 pins
2. Solder the pi onto those headers (figure 2)
3. Put terminal block and sendor headers in place and solder them on (figures 3 and 4)
4. Solder relays in place (figure 5)
5. Solder the headers onto the BME board (figures 6 and 7)
6. Solder switch in place (figure 8)
7. Pull the long headers so the top of the pin is 12mm off of the board, do this for all of the pins in the LCD headers (figures 9 and 10)
8. Solder the LCD headers into place and connect the BME280 board (11)
9. If using the AC/DC converter, solder that on

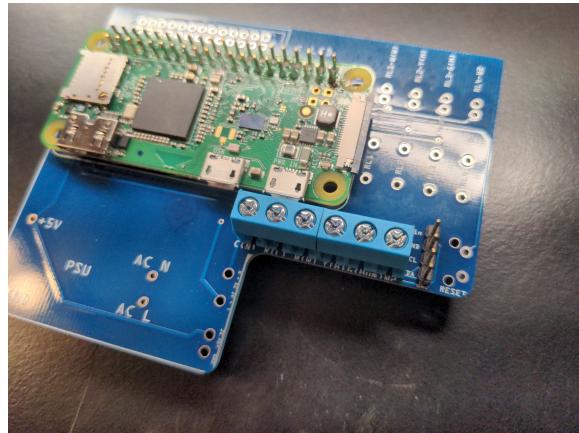


Figure 3: Terminal block and sensor headers in place

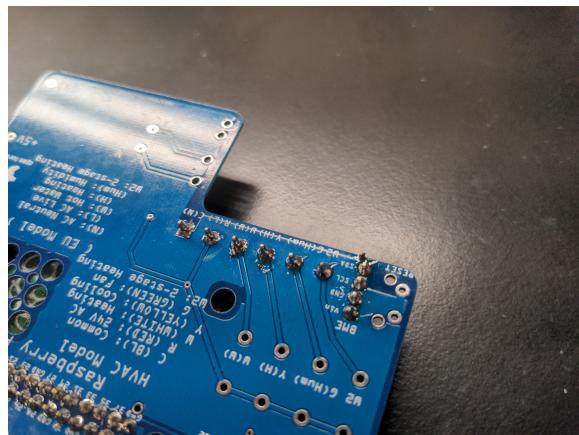


Figure 4: Terminal block and sensor headers soldered on

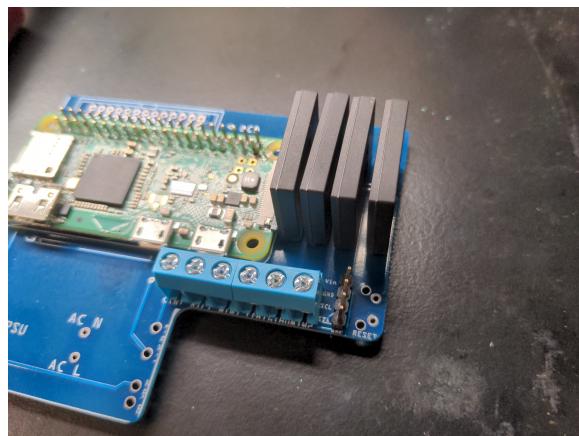


Figure 5: Relays soldered into place

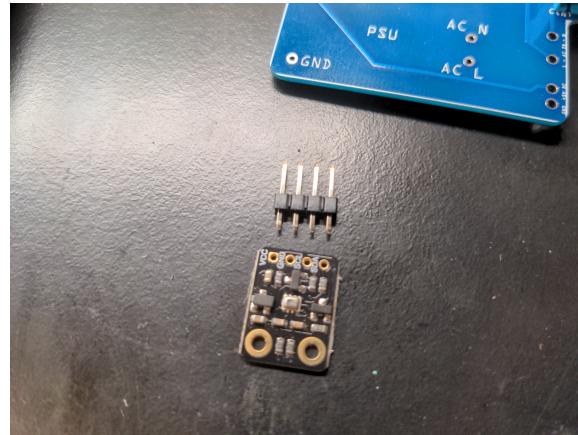


Figure 6: BME needs to have headers attached

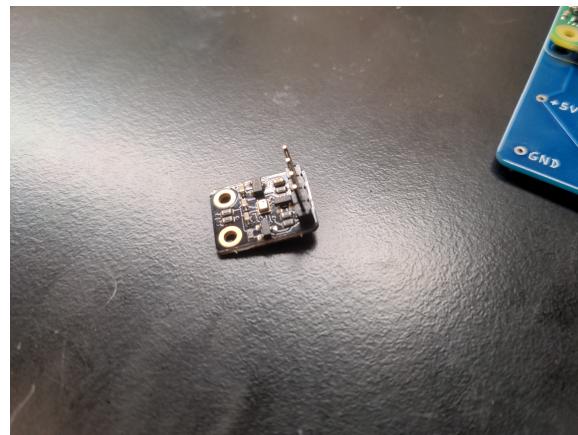


Figure 7: BME with headers soldered on



Figure 8: Solder switch into place



Figure 9: Pin placement adjusted to have 12mm above the board



Figure 10: All headers adjusted to be 12mm above the board

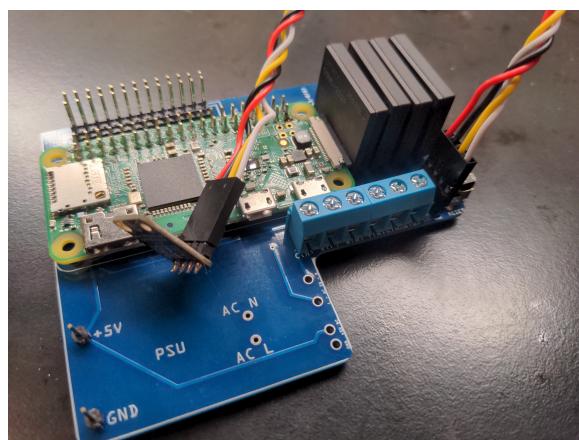


Figure 11: Fully assembled HestiaPi with headers for a DC power supply

2.2.2 Video



2.2.3 Hints and Tips

The LCD needs to be connected before powering HestiaPi as it initialises on boot only (otherwise it looks blank-white and touch events do not register) and it may also cause a freeze or reboot due to power spike.

If you cannot control mains, that is having it off during all the time of installation, our advise is to leave the SD card and LCD out, connect all wires, partly (not fully) insert the SD and finish off case installation with the LCD attached to the cover.

Once all is done, from outside of the case, push first the SD all the way in (it does not lock-click in place) and then insert a non-metallic tool and press the reset button from the right side. HestiaPi will boot and in about 10-15sec the LCD will show some of the boot messages.

2.2.4 Troubleshooting

After assembling your HestiaPi, put in an SD card that has been flashed, attach the touch screen and try booting it up.

If you get a blank screen, the pi might not be booting properly. Make sure the SD card was flashed properly. A common mistake is to flash the image onto the partition instead of the block device (e.g., /dev/sdb1 instead of /dev/sdb).

Test your reset button. If it doesn't work, odds are it's either a faulty component, or more likely a cold solder joint. The former can be fixed by replacing the part, and the latter by re-soldering the component onto the board. Use a multimeter to verify the switch works. If it does, trace the line to the pin on the pi to see if there is connectivity there.

2.3 Printing the Case

Printing the case really depends on your own printer but here are some basic guidelines that you can adjust accordingly. The power supply for HVAC - US is too high but because we use the same design for both US and EU models, you would need to clip off one of the 3 LCD hooks. Facing the cover from the outside, cut the bottom left hook. Doesn't need to be flush.

2.3.1 Files

[Download](#) the latest (set of 2) .STL files (BaseONE*.stl and CoverONE*.stl).

2.3.2 Filament

Choose a filament that stays rigid enough in the max temperature your house may reach on a hot Summer day without the AC on :)

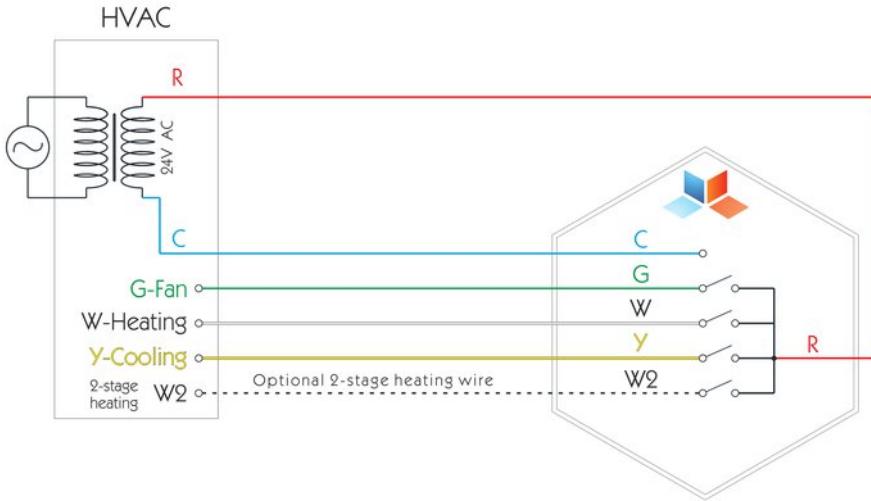


Figure 12: US Wiring Diagram

We use nGen filament for this reason but also because it prints easily and reliably. Check the same [download](#) page for printing instructions and tips.

2.4 Wall Installation ONE

HestiaPi's case comes in 2 parts. The backplate that goes to the wall and should not be visible and the front cover. The backplate should have 4 small holes, 3 larger holes and an opening for the wires coming from the wall.

If you bought HestiaPi, all internal screws are replaced with plastic rivets. Otherwise you would need:

- 4 x 2.5Mx25mm hex screws
- 4 x 2.5M hex nuts

For attaching to the wall you need:

- 4 x 3.5Mx40mm non-countersunk screws

Place the hex screws through the 4 small holes entering from the side facing the wall. Secure them in the hex slot and make sure they are sit flush. Remove the LCD from the PCB and insert the PCB alone guiding the 4 screws through the 4 corner holes of the Pi and secure with the nuts. Avoid using a large tool. You can simply tighten them by hand. Don't overtighten.

With the remaining 3 larger holes mark your wall and drill according to the location of the wires. The opening of the backplate should match the location of the wires. Secure the backplate and PCB with the 3 larger screws.

Complete wiring according to your model instructions (for US see Figure 12; for EU see Figure 13)

Remove any protective film from the LCD if present and lock the LCD on the cover from the inside making sure the LCD's header is at the top.

Guide the 4 wires through the slit of bottom partition of the cover and secure the sensor in it so that it is thermally protected from the rest of the circuit.

If you installed the bottom screw it may block the cover to fully insert. Clip off part of the sensor partition to allow enough clearance.

Hold the front cover aligned to the backplate and bring closer while you make sure the pin header of the PCB is aligned to the header of the LCD. Push firmly from the sides of the cover and not from the LCD till it locks in place. Make sure no wires are caught in between as this may block the cover from locking in place securely.

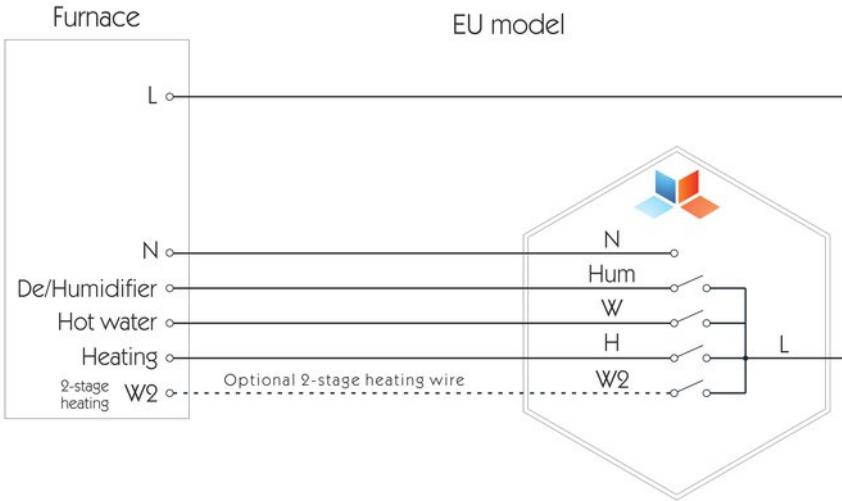


Figure 13: EU Wiring Diagram

3 Usage

3.1 Touchscreen

Once the HestiaPi Touch boots, the interface will have a simple interface to adjust the heating, cooling, or fan. It also displays the current temperature and humidity, as well as an icon in the upper right to get technical information about the Pi. The interface immediately after boot is shown in figure 14a.

Pressing the flame icon in the upper left corner displays the heating menu. In figure 14b, the status message at the bottom left corner shows that the heating system is currently turned off.

Pressing the flame icon again, will activate the menu to turn on the heat (see figure 14c). After turning on the heat, the set temperature (displayed on the left) and the + and - buttons to change the set point of the temperature will turn orange. Figure 14d shows the interface after turning up the temperature to a reasonable point.

Finally, one of the menu options in the menu to turn on the heat is “boost”. This is used to turn on the heat for a specified duration (default is 10 minutes) when the heating is scheduled to be off. Boost will still respect the temperature set point, so it can not be used to give the heat a little boost while in normal operation. When the system is in boost mode, this will be indicated along with the duration of the boost remaining in the bottom left corner of the heating menu, as shown in figure 14e.

3.2 Webpage

The web server runs on port 8080, which means to access it, you'll need to navigate to [http://\[YOUR_HESTIA_IP\]:8080/](http://[YOUR_HESTIA_IP]:8080/). When there, there will be two options for which user interface (UI) to use: Basic UI or Paper UI. The Basic UI is documented in section 3.2.1 while the Paper UI is covered by section 3.2.2. This choice can be seen in figure 15a.

3.2.1 Basic UI

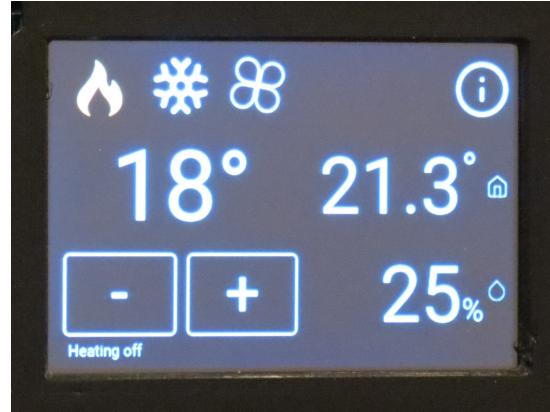
The basic UI is for controlling the thermostat from your device (phone, laptop, tablet, desktop, etc.). This UI covers the same functionality as the touchscreen (see section 3.1 for more details), plus the ability to do maintenance tasks such as taking backups, updating the software, and shutting down the HestiaPi in case you need to do hardware maintenance.

Figures 15b and 15c show the main menu. The heating and cooling can be adjusted similar to the way as is done using the touch screen on the HestiaPi. In the settings menu (figure 15d), there are some additional features which are not accessible via the touchscreen interface. This includes switching from Celsius to Fahrenheit, setting your time zone and if your system has second stage heating it can be configured here (as seen in figure 15e).

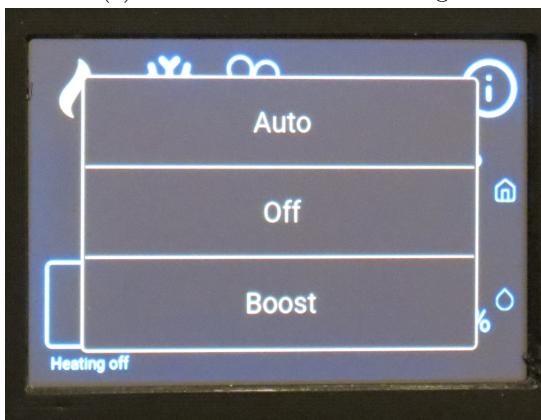
The maintenance menu allows for backups, updates, and shutting down the pi, which is typically only necessary when taking off the LCD screen to do physical maintenance or upgrades. When the backup button is pressed, it will update both the backup field to show that it is done as well as the latest backup field with the date and time of the latest backup, as shown in figure 15f.



(a) User Interface After Booting



(b) Heating Interface - Heat is off



(c) Heating Interface - Toggle heat on/off



(d) Heating Interface - The heat is on



(e) Heating Interface - Boosting the heat

Figure 14: fig:UI Menus



(a) User Interface Choices

Main Menu

Temperature

- Temperature Setpoint: 22.0 °C (sliders)
- Temperature: 22.0 °C

Heating

- Heating Mode: AUTO (button)
- Heating Boost Time: 10 min (sliders)

Cooling

- Cooling Mode: AUTO (button)
- Cooling Boost Time: 10 min (sliders)

Fan

- Fan Mode: ON (button)

Configuration

- Settings
- Info

(c) Bottom of Main Menu

Settings

- Maintenance
- 2nd Stage Heating
- Temperature Unit: °F (selected)
- System Type: HVAC
- Timezone

(d) Settings

2nd Stage Heating

- 2nd Stage Heating: ON (button)
- Threshold: 0.0 °C (sliders)
- Delay: 0 min (sliders)

(e) 2nd Stage Heating

Maintenance

- MainSwitch: ON
- Backup: Backup complete!
- Last Backup: 2020-01-24 19:41
- Restore
- Installed version
- Latest version
- Update & restart
- Reboot
- Shutdown

(f) Maintenance Menu After a Backup

Figure 15: fig:Web UI

3.2.2 Paper UI

The paper UI is used for advanced configuration settings. This section will be expanded in the future, but in the meantime, you can read the [OpenHAB documentation](#).

3.3 Mobile app

The interface for the mobile app is almost identical to the basic UI of the webpage (covered in section 3.2). The only significant differences are getting the application and connecting to the HestiaPi.

The application can be downloaded from the [Google Play store](#), or [Apple's App store](#). Ideally, as long as your mobile device is connected to the same network as the HestiaPi, the app should automatically find the HestiaPi's OpenHAB server. If this works as expected, everything should look similar to the screenshots shown in figure 15.

If the server is not found, the hamburger menu in the top left (three horizontal lines) will bring up a menu that allows access to the settings. In the settings menu, there is a Local section which allows connecting to a local OpenHAB server. The app refuses to connect to unencrypted web servers when the URL is entered manually, so the URL should be slightly different than described in the web UI: `https://[YOUR_HESTIA_IP]:8443/` Once the URL is entered, return to the settings screen and in the local section, it should say "Insecurely connected to YOUR_HESTIA_IP". It says the connection is insecure because the app has no way to verify that the server is actually the correct one. The app will work fine when connected "insecurely" and to get a secure connection requires quite a bit of effort and technical know-how. For instructions on how to get the app to say it's a secure connection, see the "[Set up TLS](#)" section (4.6).

Now that the application is connected to the server, click the back arrow to return to the Main Menu.

4 Configuration

4.1 First boot

Fix your HestiaPi's case to the wall first. If you simply want to test-drive HestiaPi before committing to it, connect the LCD first and then plug in a Micro USB cable to the Pi's port.

1. Insert the MicroSD card back in the Raspberry Pi. Just push it in. It does not click. It does not lock in place. A tiny part of it will stick out just enough to grab and pull it if needed.
2. Insert the LCD in the cover. Remove the protective film if present. Turn and push it in place. It should feel firm in place.
3. Take all necessary precautions before applying mains voltage so cut off the power now!
4. Connect all Heating, Cooling, Fan and Hot Water (depending on model) control lines on the terminal block's contacts.
5. Connect remaining wires, marked C(N) and R(L).
6. Place the sensor at the bottom compartment of the cover and fit the 4 wires in the vertical slit. Note that the sensor, the little shiny square, should be placed facing outwards and ideally not be blocked by any plastic piece of the case. The red wire (Vin) goes to the top pin (Vin) on the PCB.
7. Align and push evenly the cover against the base aligning at the same times the pins with the LCD connector. The cover should lock when pushed all the way in. Step back and enjoy the new looks of your wall :)
8. If you cannot cut off the power on the cables, you are risking of HestiaPi booting before the LCD is connected. In such a scenario the LCD will not display anything but a blank white screen and you would need to restart as it is not "plug and play" like HDMI. We would advise to leave the SD card out before applying mains voltage and just before you are about to close the case, insert it but don't restart. It shouldn't boot. Once you close the case, there is a chance that it will restart. Close the case and wait 20 seconds. If nothing shows up on the screen, it didn't restart. Press reset button from the right side.
9. If at any time you want to remove the top cover, select "Shutdown" from the App. When HestiaPi Touch is completely shut down, simply pull the cover outwards.
10. You should soon see the HestiaPi boot sequence and the loading screen at the end with a countdown. Follow [these steps](#) to connect your new HestiaPi to your WiFi.
11. After a few seconds the screen will show if the WiFi is connected and what is the local IP it got (DHCP)
12. The full installation may take up to 20 minutes for the very first time and a few restarts are normal. Just leave it alone. You can always SSH to it. Use `pi/hestia`

13. The SD card image expands automatically to occupy the complete size of the card if available.
14. While waiting, head over to the [downloads](#) section and download the smartphone app on your phone. Under settings set Local OpenHAB URL as [http://\[hestiapi_IP\]:8080](http://[hestiapi_IP]:8080) and close the application
15. The LCD UI starts with 0 values or blank fields. This is normal until it gets ready.
16. Once the LCD is showing the UI with temperature values, try and load the app again or simply use your laptop and navigate to:
[http://\[hestiapi_IP\]:8080/start/index](http://[hestiapi_IP]:8080/start/index) and select “Basic UI”
17. You should now be able to control the basic functions from either the App or your browser
18. Please note that the UI of the app, web and LCD may change with software updates so back up your customisations before running an update.
19. OpenHAB2 has a great [forum](#) with so much information from fellow users. Salivate at what you want to make now with it.
20. Feel free to explore the files under `/etc/openhab2/names/default.*` in folders items, rules, sitemaps and things.

4.2 Boot Sequence

On startup, the HestiaPi will run a number of services, which does take some time due to the low computational power of the pi. However, booting is something which very infrequently done, which is why the boot times of five minutes or more are acceptable in order to keep the size and cost of the hardware low.

Understanding the startup process requires understanding what components are running. The components are documented in section [1.3](#).

Systemd starts up a number of services, including:

1. mosquitto
2. hcuart
3. dhcpcd
4. openhab2

The status of all of these services can be checked with "`systemctl status SERVICENAME`" where SERVICENAME is replaced with the name of the service of interest. For example, to check the status of Mosquitto: `systemctl status mosquitto`

In addition to things started by systemd, there are also scripts which are run from `/etc/rc.local`.

4.3 Connect WiFi

Follow the on-screen instructions on the LCD when it prompts to connect your phone to the “HESTIAP” network with HESTIAPI as the password. Once connected you will automatically be prompted on your phone to select your WiFi network from a list (no hidden SSID supported yet) and enter the password.

Your HestiaPi will then restart to connect to your network and the HESTIAPI network will not be shown again if you entered the correct the details.

4.4 Advanced Settings

4.4.1 Remote Temperature Sensors

Sometimes the thermostat wires are poorly located. This may mean near a door that throws off the temperature, in a room that gets a lot of sun and therefore runs hotter than the other rooms, or any number of other problems. If this sounds familiar, you may be interested in having a remote temperature sensor.

There are many different models of remote temperature sensors to choose from. The main feature that you will want to look for when selecting a sensor is that it supports MQTT. MQTT is the protocol that will allow the temperature sensor to send the current temperature to HestiaPi.

To get the IP address of your HestiaPi, click the information icon (i) in the top right of your HestiaPi. Consult the documentation of your temperature sensor on how to configure it to publish temperature data to the HestiaPi. It should be publishing to port 1883 on the HestiaPi, which is a standard port for MQTT. There are some notes on how to set this up for Ruuvi tags in the section on [4.4.2](#).

After you've done this, the data is being sent to the HestiaPi, but we need to reconfigure the thermostat to use this new data. This will be done via the web interface, and specifically the PaperUI. You can get to this by navigating to [http://\[YOUR-HESTIAPPI-IP\]:8080/paperui](http://[YOUR-HESTIAPPI-IP]:8080/paperui) (where you insert the IP address of your HestiaPi in labeled location).

We will walk through an example which sets up a temperature sensor in the living room. You can use a different names for your sensor if appropriate.

First, we need to make an Item to store the data. Go to Configuration -> Items and click the (+). Enter the following settings, leaving anything not mentioned at its default value:

- Name (can't have spaces): livingroomtemp
- Label (human readable, can have spaces): Living Room Temperature
- Type: Number
- Dimension: Temperature

Click the plus in the blue circle at the top to save. For more information about what an "item" is in OpenHAB terminology, see the OpenHAB documentation at: <https://v2.openhab.org/docs/concepts/items.html>

Second, we need to make a Thing to represent the sensor. Go to Configuration -> Things -> (+) -> MQTT Binding -> Add manually -> Generic MQTT Thing.

- Name (human readable, can have spaces): Living room sensor
- Thing ID: livingroomsensor
- Location: Sensors
- Bridge selection: Mosquitto MQTT Broker

Click the checkmark in the blue circle at the top to save. For more information about what a "thing" is in OpenHAB terminology, see the OpenHAB documentation at: <https://v2.openhab.org/docs/concepts/things.html>

Third, we need to add a Channel to the Thing. We will just go over how to make one channel for temperature, but additional channels could be made for humidity, barometric pressure and anything else that your sensor can measure. Go to Configuration -> Things -> Living room sensor -> Channels (+)

- Channel type: Number Value
- Channel ID (no spaces): temp
- Label (human readable): Temperature
- MQTT State Topic: <see notes below>

The MQTT State Topic will depend on what topic your specific temperature sensor is using when it is publishing the data. For example, the RuuviBridge will use a prefix of "ruuvi" and then the BlueTooth MAC address of the Ruuvi tag. So the MQTT State Topic would be something like "ruuvi/AA:BB:CC:DD:EE:FF".

Each MQTT topic **should** only have one value. For example, it would be nice if the RuuviBridge published temperature data to the topic: "ruuvi/AA:BB:CC:DD:EE:FF/temperature". Then the humidity could go to "ruuvi/AA:BB:CC:DD:EE:FF/humidity" and so forth. Unfortunately, that's not how it currently works. Instead, RuuviBridge publishes all of the fields to a single topic in JSON format. There are other temperarure sensors that also follow this pattern.

To convert the entire JSON string to just the temperature data, click "Show More" and enter "JSONPATH:\$.temperature" in the "Incoming Value Transformations" field. This will extract just the temperature field. The same can be done for "humidity" and "pressure" if you are interested in making that information available to your thermostat.

If you configured the Channel to use the JSONPATH transformation, you'll need to install that add-on. Go to Add-On -> Transformations tab, and click install next to "JSONPath Transformation". It will take several minutes to install. Please be patient.

Next, we want to create an Item to hold the temperature that is being recorded by the temperature sensor on the HestiaPi, just in case we decide we want to see it or use it in the future. Configuration -> Items -> (+)

- Name (can't have spaces): localtemp
- Label (human readable, can have spaces): Local Temperature
- Category (optional, leave blank)

- Type: String

Click the plus in the blue circle at the top to save.

Now we want to configure the local temperature sensor to send data to this new Item and stop sending it to the “MyTemp” Item that is displayed on the HestiaPi and is used to determine when to turn the heating and cooling on and off. Go to Configuration -> Things -> Temperature Sensor

Under channels, click the blue concentric circles to the left of the “Output” channel, then click the trash can icon to the right of the linked item “Temperature (MyTemp)”. This will cause the temperature on the HestiaPi to stop updating (until we reconfigure it).

While still in the Output channel of the Temperature Thing: Next to Linked Items -> (+)

- Profile: Default
- Item to link: Local Temperature (localtemp)

Click the Link button to save. Now the temperature sensor on the HestiaPi is going to this localtemp Item.

Now we need to change the type of the MyTemp Item from a String to a Number. Go to Configuration -> Items -> Temperature (MyTemp) -> Pencil icon

- Type: Number

Click the check mark in the blue circle at the top to save.

It's important to leave the Dimension blank here. Setting it to Temperature will result in an incorrect conversion from Celsius to Fahrenheit (so it would display 32F when it is actually 0F).

Finally, we want to use a Channel to our new sensor Thing to the Item where the data is stored. Specifically, we want our Living room sensor to update both the livingroomsensor item as well as the MyTemp item.

Go to Configuration -> Things -> Living room sensor.

Under channels, click the blue concentric circles to the left of the Temperature channel.

- Profile: Follow
- Item to link: Temperature (MyTemp)

Then click on the concentric circles and edit the linked item.

- Profile: Default

The reason we have to choose Follow first is because MyTemp is not available under the Default profile. However we can get to it under Follow and then switch it back to the correct profile type of “Default”.

Repeat this process to also link the Living room sensor to the “Living Room Temperature (livingroomtemp)” item.

If you'd like to learn more about linking things and items with channels, the OpenHAB documentation covers that here: <https://v2.openhab.org/docs/configuration/paperui.html>

At this point, we should be all done. Go to Control -> SENSORS and you should see the items you added (Local temperature, Living room temperature) and the Temperature (which is what is used to actually control the HVAC) should match the Living room temperature. The LCD on the HestiaPi should also be displaying the Temperature, which should now be coming from the remote sensor.

Congratulations, you now have configured your HestiaPi to use your remote temperature sensor!

4.4.2 Ruuvi Temperature Sensors

The Ruuvi tags require a Ruuvi Gateway device in order to support MQTT. If you do not have a Ruuvi Gateway, there is a project which will effectively make one out of a Raspberry Pi here: <https://github.com/Scrin/ruuvi-go-gateway>

Unfortunately, the Ruuvi Gateway does not publish the data in a usable format and the ruuvi-go-gateway outputs the data in that same unusable format in order to maintain compatibility. This means a bridge is required to decode the data and then output it in a usable format. The code for the bridge can be found at: <https://github.com/Scrin/RuuviBridge/>

Both of these can be installed on a Raspberry Pi to get the data into a usable format. They can even be installed on the HestiaPi itself, if you're so inclined. Consult the documentation for these projects on how to install and configure them.

A typical configuration is to have the ruuvi-go-gateway publish the encoded data to a local MQTT server and have RuuviBridge subscribe to that local MQTT server and then publish the properly formatted data to the HestiaPi.

After you have ruuvi-go-gateway and RuuviBridge configured, the temperature data will be going to your HestiaPi. The next step is to configure the HestiaPi to use this data. See [4.4.1](#) for more information on how to do this.

4.4.3 Easy Remote Access

All latest releases of HestiaPi offer very easy remote access to your home without touching your network modem/router or even knowing HestiaPi's IP! Does not depend on port forwarding or DynDNS! Woohooo!

Please note that this is an externally hosted service not controlled by you or us but by OpenHAB itself.

[Instructions video](#) (if you prefer video to text)

To activate it (shipped disabled by default for obvious reasons) go to [http://\[YOUR-HESTIAPI-IP\]:8080/paperui](http://[YOUR-HESTIAPI-IP]:8080/paperui) and select Add-ons > MISC and make sure “openHAB Cloud Connector” is installed.

Once installed SSH into your HestiaPi (username: pi and password: hestia) and type:

```
cat /var/lib/openhab2/uuid
```

copy the output somewhere. Then type:

```
cat /var/lib/openhab2/openhabcloud/secret
```

copy this output too. Reboot your HestiaPi

```
sudo reboot
```

Then go to <https://myopenhab.org> and create an account using your details and the above information (UUID and secret).

You can now access your HestiaPi Touch from a browser or your mobile app

Hint: Enter <https://myopenhab.org> as a remote url and your myopenHAB account username and password as credentials

4.4.4 Update Your DynDNS Automatically

[getpublicip.sh](#) does just that.

Stored with instructions here:

```
/home/pi/scripts/getpublicip.sh
```

You would need an account depending on the service you choose to use inside the script.

4.5 Setting a Static IP Address

Setting a static IP address on a Raspberry Pi can be done with either the dhcpcd or the networking service. These are documented in more detail [on this stackexchange post](#), so this guide only goes over the basics. The dhcpcd option has been thoroughly tested with the HestiaPi in particular, but the networking option should also work, as that has been tested with Raspberry Pis in general.

A third option is to get a dynamic IP address every time the HestiaPi starts up, but configure your router to look at the MAC address that the request came from and always assign the same dynamic address. This process will be different for every router and can not be covered in this documentation. For instructions on how to do this, consult your router's documentation.

4.5.1 Configuring dhcpcd

Using dhcpcd to configure a static IP address requires editing `/etc/dhcpcd.conf` and restarting the dhcpcd service. The configuration file contains some commented out examples, which look similar to this:

```
interface wlan0
static ip_address=10.1.1.31/24
static routers=10.1.1.1
static domain_name_servers=10.1.1.1
```

Then run `sudo systemctl restart dhcpcd` to start using the new configuration. The system will then have two IP addresses, the DHCP assigned one that it already had, and the newly assigned one. This is because dhcpcd doesn't remove any IP addresses. When the HestiaPi is rebooted, it will only have the static IP address.

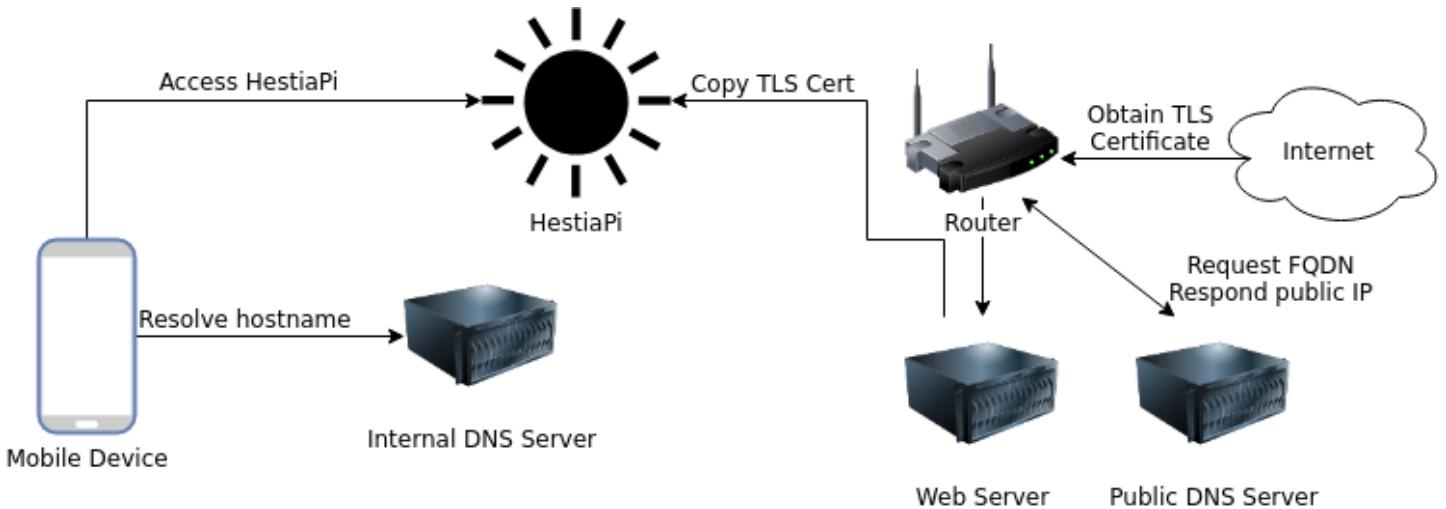


Figure 16: Complicated TLS Setup Overview

4.5.2 Configuring Networking Service

The other option is to just disable DHCP all together and set the static IP address in `/etc/network/interfaces`. The configuration would look similar to the example below:

```
iface wlan0 inet static
    address 10.1.1.31
    netmask 255.255.255.0
    gateway 10.1.1.1
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Once this is set, run `sudo systemctl disable dhcpcd` to make sure it doesn't cause any issues, then run `sudo systemctl enable networking`. This should reconfigure the interface to use only the statically assigned IP address. Unlike the `dhcpcd` option, this means your SSH connection will be terminated and you will have to SSH into the machine again, but this also avoids a reboot (however rebooting is still recommended to verify that the IP address is set properly on boot).

4.6 Set up TLS

Setting up Transport Layer Security (TLS) is an advanced topic which requires owning a domain, having control over a DNS server, the ability to forward ports on the edge router, setting up web servers, and using the command line. As such, it is recommended that only people who are at least somewhat familiar with these technologies attempt to set this up. For the vast majority of users, setting up TLS is not necessary, and it is safe to skip this step.

At the core, setting up TLS just means giving the server a host/domain name, getting a trusted certificate for that name, and accessing the server by name instead of by IP address. The rest of this section assumes the reader is familiar with how to SSH into their HestiaPi and switch user to be root.

The example provided here is just one way to get a TLS certificate, and the method described prioritizes making sure the HestiaPi is never directly accessible from the Internet. This ensures that random people in the internet will not be able to modify your home automation system.

Figure 16 shows the overall setup. In this example, we will use the domain `example.org`. You will need to replace this domain with one that you control. The process involves setting up a web server which will publicly be known as `hestia1.example.org`. This web server will be directly connected to the internet and is what will obtain the TLS certificate. The public DNS server will resolve `hestia1.example.org` to your public IP address. This is needed because the certificate authority (Let's Encrypt in our example) will reach out to the hostname to verify that it really is who it claims to be. Once the web server has the TLS certificate, it can be copied over to the HestiaPi to be used there. The final component is an internal DNS server, which is used for devices on the LAN to connect to the HestiaPi by name and it needs to be given the internal IP address. It is possible to configure some DNS servers to give out different results depending on who it asking (e.g. the public IP if the router asks, but the LAN IP if the request comes from anywhere else on the LAN), but this documentation chose to have two separate servers in an attempt to make the configurations less complex.

4.6.1 Static IP

To assign the HestiaPi a static IP address, see section [4.5](#).

4.6.2 DNS Servers

Setting up a DNS server is beyond the scope of this document. There are many guides which have been written on the topic, such as [this](#). If you have a bind9 server, adding an entry for the HestiaPi in the internal DNS server might look like this:

```
hestia1      IN      A      10.1.1.31
```

The external entry would look the same, but it would be your public IP address instead of the HestiaPi's internal IP address. The internal DNS server will also need to be configured to do recursive DNS lookups. This is typically set in `/etc/bind/named.conf.options` using something like the following inside the `options` section:

```
allow-recursion { 10.1.1.0/24; localnets; };
allow-query { 10.1.1.0/24; localnets; };
allow-query-cache { 10.1.1.0/24; localnets; };
recursion yes;
```

For more details on how this works, read [documentation about bind9](#) or [this stackoverflow post](#).

Updating your computers and mobile devices to point to this name server is best done at the router. Instructions for making this change will vary from one router to the next, but your router's documentation should explain how to change the nameservers that the DHCP server is assigning.

Once the entry has been added to the nameserver and your computer or device is using that nameserver, you should be able to access the HestiaPi by name. In the example above, the host name would be `hestia1` and the domain used in the linked blog post is `debuntu.foo`. This means opening a browser and going to: <http://hestia1.debuntu.foo:8080/>

4.6.3 Obtain TLS certificate

In order to avoid connecting the HestiaPi to the internet where anyone could interact with it, this guide shows how to obtain a trusted TLS certificate using a web server. At this point it's expected that a public DNS server has been configured to point to your public IP address (see section [4.6.2](#)).

The next step is to set up a Linux server which will act as the web server. Once Linux is installed, you'll need to set up a web server, such as [nginx](#). Consult your router's documentation on how to forward port 80 from your router to the web server. At this point you should be able to access the webpage from the internet. An easy way to test this is to use a mobile phone or tablet with cell data service. Disconnect from wifi and attempt to go to `hestia1.example.org` and verify there's a web server running. This verifies that the DNS, port forwarding and web server are all working correctly.

Follow the [instructions](#) on how to use certbot to obtain a TLS certificate. The [LetsEncrypt Documentation](#) is rather comprehensive and can provide additional context to how their system works.

Once complete, there should be a directory in `/etc/letsencrypt/live` for your hostname which contains links to the TLS certificates.

4.6.4 Configure HestiaPi to use new Certificate

In order to use your new certificate, it needs to be converted from PEM format to pkcs12 format and imported to the Java keystore, after deleting the previous certificate. At this point you should have two files: one with the private key for your certification, and the other should be your public certificate and any intermediate certificates that browsers will need to verify your certificate. With letsencrypt, these files are named `privkey.pem` and `fullchain.pem`.

First, we convert these to pkcs12 format and put them into a single file, with the password that the HestiaPi is going to expect ("openhab").

```
openssl pkcs12 -export -inkey privkey.pem -in fullchain.pem -out openhab.p12 -passout pass:openhab
```

Next, we want to stop openhab so we aren't modifying a file that is in use, and we delete the key that is there (whose alias is `mykey`), and add the .p12 file that we just created.

```
# Lets not modify a keystore that is in use...
sudo systemctl stop openhab2
# Delete the old key
sudo /opt/jdk/zulu8.40.0.178-ca-jdk1.8.0_222-linux_aarch32hf/bin/keytool \
    -keystore /var/lib/openhab2/etc/keystore \
```

```

-v -storepass openhab -delete -alias mykey
# Import the new key
sudo /opt/jdk/zulu8.40.0.178-ca-jdk1.8.0_222-linux_aarch32hf/bin/keytool \
    -keystore /var/lib/openhab2/etc/keystore -importkeystore \
    -srckeystore ~/openhab.p12 -srcstoretype PKCS12 \
    -destkeystore /var/lib/openhab2/etc/keystore -deststoretype jks \
    -destalias mykey -srcalias 1 -srcstorepass openhab -deststorepass openhab

```

In the event you want to look to see what is in the keystore, you can do so with the following command:

```

# List keys (for debugging)
/opt/jdk/zulu8.40.0.178-ca-jdk1.8.0_222-linux_aarch32hf/bin/keytool \
    -keystore /var/lib/openhab2/etc/keystore \
    -v -storepass openhab -list

```

Finally, we start openhab2 and it may take a while for the web server to start, but when it does, it should be using the new certificate.

```
sudo systemctl start openhab2
```

Navigate to your hostname (e.g., `http://hestia1.debuntu.foo:8443/`) and you should have an encrypted connection to your HestiaPi!

4.7 File Structure & Paths

WiFi details

```
/etc/wpa_supplicant/wpa_supplicant.conf
```

OpenHAB Items

```
/etc/openhab2/items/default.items
```

Rules

```
/etc/openhab2/rules/default.rules
```

Sitemaps

```
/etc/openhab2/sitemaps/default.sitemap
```

Things

```
/etc/openhab2/things/default.things
```

Logs

```
/var/log/openhab2/events.log
```

```
/var/log/openhab2/openhab.log
```

LCD UI The LCD UI is an HTML-based page loaded on a fullscreen browser. All HTML, CSS, JS, fonts and icon files are in here

```
/home/pi/scripts/oneui
```

The vue framework is used.

Scripts In /home/pi/scripts

There are AdafruitDHTHum.py

AdafruitDHTTemp.py

Read sensor data from DHT sensors.

```
C2F.sh
```

```
F2C.sh
```

Change HestiaPi from Celcius to Fahrenheit and vice versa.

```
getBMEhumi.sh
```

```
getBMETemp.sh
```

```
getBMEpress.sh
```

Read sensor data from BME sensors (calling bme280.py).

getcputemperature.sh

Returns RasPi CPU temperature.

getssid.sh

Returns WiFi SSID name.

gettz.sh

Returns system Timezone.

getuseddiskspace.sh

Returns used SD card space.

getwifiinfo.sh

Returns WiFi signal strength.

getwlan0ip.sh

Returns WiFi IP.

getwlan0mac.sh

Returns WiFi MAC address.

netcheck.sh

Cron script that checks WiFi connectivity by pinging its gateway. If no response is received at the first time, the WiFi interface is restarted and a DHCP (dynamic) IP is requested. If no response is received again RaspberryPi, the reboot command is sent. Please note this script is not enabled by default and you will need to follow the instructions supplied at the top of the file. Please also note that restarting the Pi will stop any current task and will not resume after restart.

openhabloader.sh

Loads the Touch LCD UI.

getpublicip.sh

Checks current public IP and if it matches with previous reading, it does nothing else. If current public IP is different, the latest value is sent to your account (manual and free account registration needed).

Web UI

`http://[YOUR_HESTIA_IP]:8080/basicui/app`

or simply

`http://[YOUR_HESTIA_IP]:8080`

and then select Basic UI and default

Smartphone App Under Settings > Local server settings

`http://[YOUR_HESTIA_IP]:8080`

5 FAQ

5.1 Configuration

5.1.1 Default SSH Username and Password

Username: pi

Password: hestia

SSH port: 22

5.1.2 MQTT Configuration

All the topics are defined in the [.things file](#).

Confirm by subscribing from another laptop to all (#) MQTT IDs and listen for published messages while you play with your HestiaPi. For Linux users, run this in a terminal:

```
mosquitto_sub -h [HESTIA_PI_IP] -d -t hestia/#
```

5.1.3 How to Access My HestiaPi From Outside My House

You will need a WiFi router with port forwarding feature (most routers do these days) and if you don't have a static IP (or if you don't know what this is), you will need to use a free Dynamic DNS service called [NoIP](#). Don't worry – although we can't offer support on individual routers, we can certainly point you in the right direction. Installation instructions on the above link. Alternatively you can use [my.openhab.org](#) which is a service hosted externally and is not controlled by us or you but by OpenHAB itself. Go to:

```
http://[YOUR-HESTIAPI-IP]:8080/paperui
```

and select Add-ons > MISC and install "openHAB Cloud Connector" if not installed. Once installed SSH into your HestiaPi (username: pi and password: hestia) and type:

```
cat /var/lib/openhab2/uuid
```

write the output down. Then type:

```
cat /var/lib/openhab2/openhabcloud/secret
```

write this output down too.

Then go to <https://myopenhab.org> and create an account using your details and the above information (UUID and secret). You can now access your HestiaPi Touch from a browser or your mobile app (enter "<https://myopenhab.org>" as a remote url and your myopenHAB account username and password as credentials). The above steps are also available in youtube format [here](#) too.

5.2 Troubleshooting

5.2.1 Taps do not register on the touchscreen

If the screen doesn't turn on, see the entry below about that. These steps are to troubleshoot situations where the display is working correctly, but just not responding to touches.

This most often is caused by a pinched screen. This can be either the tabs of the case pinching the screen too tightly or the case pinching down on the front of the screen. First, look for any wires that may have gotten pinched between the relays or power supply (the tall components) and the screen when the case was snapped into place. This is the most common cause of this issue and fortunately it is easily resolved.

To confirm that it is a pinch, connect the screen without the shell of the case and power on the HestiaPi. If it works correctly, this tells us that the case is to blame.

To determine if it's the shell pinching down when it is closed, remove the HestiaPi from the case's backplate, then connect the screen while it is in the case's shell. Do not snap on the backplate. If the touchscreen works in this configuration, we know the tabs that hold the LCD to the shell are not the problem.

If the screen does not work with the backplate off, we know that the tabs are pinching the screen too tightly. In this case, the tabs can be very carefully scraped down with a hobby knife, box cutter, file or something of this nature. Take care not to break of the tabs or hurt yourself.

If you are in a situation where the shell of the case is pinching down on the screen only when it's connected with the backplate (and there are no wires getting in between the screen and the rest of the HestiaPi), it means the case is a bad fit. One option is to not quite snap the case completely shut. The LCD pins will hold the shell in place and it should work fine. The next option is to run without the shell on, which will give you a very punk look to your thermostat. Or you can get a custom printed shell for your case that is slightly taller.

If you are an advanced user and want to try to resolve this issue, the tails of the headers on the back of the main PCB can sometimes be trimmed to get just a tiny bit more room, which is often all that is needed. The PCB should generally be as close to the backplate as possible. Another possibility is to trim all of the LCD headers to make them just a little bit shorter. This will make it even more difficult to align the pins when putting the case on, but people have successfully repaired their thermostats by doing this.

Other causes of the touchscreen not registering taps include: the Raspberry Pi was powered on when the LCD screen was connected improperly, someone pushed on the screen too hard, or only some of the pins are fully connected to the screen. If the touch part of the screen is broken, the above tests will all have the same results (no response to taps). At this point, your options are to replace the screen or just use the screen for viewing information and make all the changes to the thermostat over wifi.

5.2.2 Screen doesn't turn on

If you see lights illuminated on the pi, but the screen doesn't turn on at all, the screen is not receiving power, which means it is likely not connected properly. Look in through the side of the case and see if you can tell if the pins are aligned correctly. If you can not determine this, try removing the shell of the case and connect the screen on without the shell of the case. The screen should light up slightly when powered on. If this still doesn't work, it is likely a defective screen. If you have another Raspberry Pi available, you can connect it to the top left pins of the Pi (when the SD card is on the left) and see if the screen lights up when that Pi is powered on with the HestiaPi's SD card in it.

If the screen lights up slightly, but is blank, this is a different issue. In this case, the screen is likely working and it's an issue with the software. You should see the screen flicker within 60 seconds and you should see the boot messages after a few minutes.

The fastest and easiest fix is to just copy the image onto the SD card again and try again. If you are an advanced user and want to spend time troubleshooting the issue to determine the root cause, we encourage you to share whatever you find with the community at <https://community.hestiapicom>.

5.2.3 Screen is slow to respond to taps

The HestiaPi should respond to touches within a second. If it's responding to taps, but it taking a long time, it's likely a software issue. In this case, copying a fresh image onto the SD card will likely fix it. The other thing that can be done to get it to respond faster to touches is to get the fastest SD card available.

A video has been made to demonstrate how quickly the screen should respond to taps. A stylus is used here to make it clear when the tap is occurring versus when the screen is updating, but it should work the same when tapping with your finger.
<https://peertube.gsgambit.com/w/1AhKQByTrAg38HKSQUJFR3>

5.2.4 How to edit files via SSH

If you are very new to command line interface we would advise you taking a short online course by searching for "linux command line interface" on your favourite website.

To edit a file while you are inside SSH use the command

```
sudo nano /path/to/your/file
```

Then leave your mouse alone as it does not control you cursor anymore

Use only your keyboard and once you are done, press Ctrl+O to save and Ctrl+X to close.

5.2.5 Start OpenHAB2 in Debug Mode

For OpenHAB2 (v10.x image – July 2018) To monitor the OpenHAB logs without stopping the service run

```
openhab-cli showlogs
```

To start OpenHAB manually after stopping the service run

```
openhab-cli start
```

For older OpenHAB installations: Stop OpenHAB first

```
sudo service openhab2 stop
```

and when it is stopped, start it manually

```
/usr/share/openhab2/start_debug.sh
```

once (if) loaded type inside the OpenHAB session

```
log:tail
```

and notice any issues.