HestiaPi Owners Manual

HestiaPi Community

December 23, 2024

Contents

1 Compatibility

This section will help you determine if your heating, venting and air conditioning (HVAC) system is compatible with a HestiaPi. HVAC standards vary around the world, but the HestiaPi has been designed to be compatible with the standard systems in the United States of America and in Europe.

It is important to note that while the vast majority of systems use the standard controls for the country they are in, some have proprietary controls. The sections below will help you determine if your home uses the standard HVAC controls.

If you have additional wires that are not listed in the sections below, the HestiaPi may not be able to control all the functionality of your system. You will want to check the owner's manual for your HVAC system to determine what the additional wires do and whether you would be comfortable with losing that functionality.

1.1 USA

If you open your existing thermostat and see wires labeled R, G, W, and Y, then your system is compatible.

If you also have a wire labeled C, then the HestiaPi can be powered by the HVAC system, eliminating the need to plug the thermostat into an outlet. A wiring diagram can be found in Figure 1.

If you have two stage heating, which is more common with heat pumps than gas, oil, or electric furnaces, you should also see a W2 wire. This is also supported by the HestiaPi.

1.2 Europe

If you open your existing thermostat and see wires labeled N, Hum, W, and H then your system will be compatible with the HestiaPi and can be powered directly from the HVAC system, eliminating the need for plugging the thermostat into a power outlet. A wiring diagram can be found in Figure 2.

2 Initial Setup

2.1 Preparing for Installation

These instructions will help you prepare to install your thermostat onto the wall.

1. If you didn't buy your HestiaPi with a pre-flashed a MicroSD card, follow the instructions from section 6.1.1 to flash one.
2. Insert the MicroSD card into 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.
3. Turn off the breaker for your HVAC system to make sure you are not working with live electrical wires!
4. Open your old thermostat, mark the wires so you know which goes where, and then disconnect everything and remove the old thermostat from the wall.

At this point you are ready to install the thermostat using the instructions in section 2.2.

2.2 Physical Installation

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.

- 4x 2.5Mx8mm hex screws
- 4x 2.5M hex nuts

For attaching to the wall you need:

- 3x 3.5Mx40mm non-countersunk screws (or whatever you have that fits)

Place the hex nuts in the 4 small recesses on the side of the backplate that will face the wall. Bolt the HestiaPi PCB to the backplate using the hex screws. They should just be snug. Do not overtighten.

Plave the backplate on the wall with the opening of the backplate lining up with the location of the wires from your HVAC system. With the larger three holes, mark your wall and drill holes smaller than your remaining screws. Secure the backplate to the wall with the 3 larger screws.

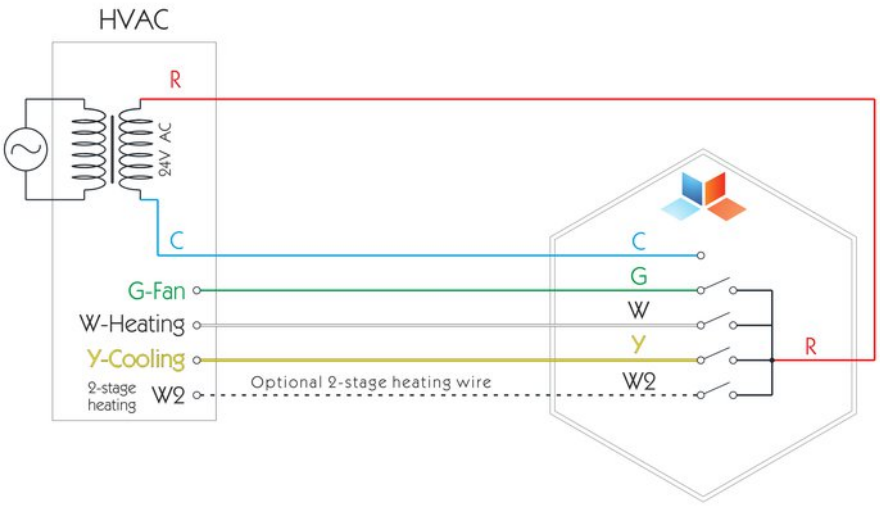


Figure 1: US Wiring Diagram

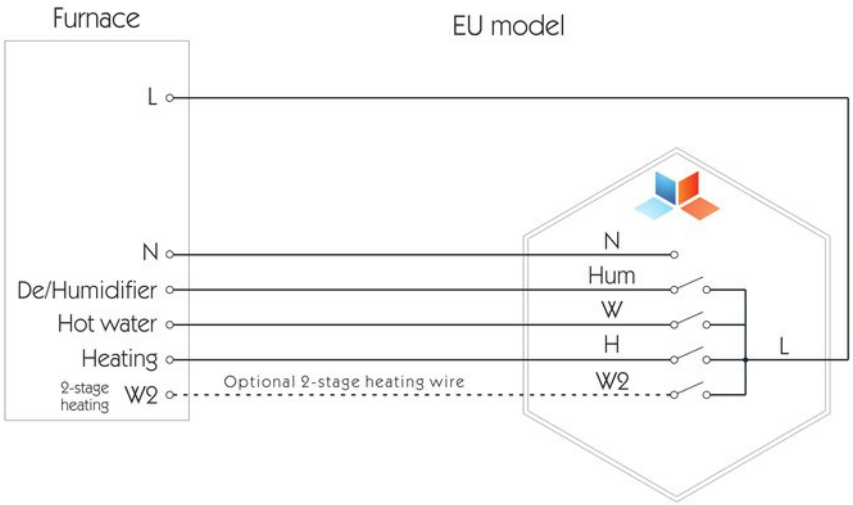


Figure 2: EU Wiring Diagram

Complete wiring according to your model instructions (for US see Figure 1; for EU see Figure 2)

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.

Attach the 4 wire cable to the temperature sensor, making sure that the red wire is connected to the Vin pin.

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. Note that the sensor, the little shiny square, should ideally be placed facing outwards and not be blocked by any plastic piece of the case. The red wire (Vin) goes to the top pin (Vin) on the PCB.

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 (not on the LCD screen) until it locks in place.

Make sure no wires are squished between as this may block the cover from locking in place securely, or could cause issues with the touch screen.

It is critical that the pins for the LCD are lined up correctly before powering on your HVAC system. If they are offset, there's a risk of damaging the screen.

Once everything is connected, turn on the breaker to your HVAC system. If your HVAC unit can not power the HestiaPi and need to plug the thermostat into an outlet, plug it into the wall now.

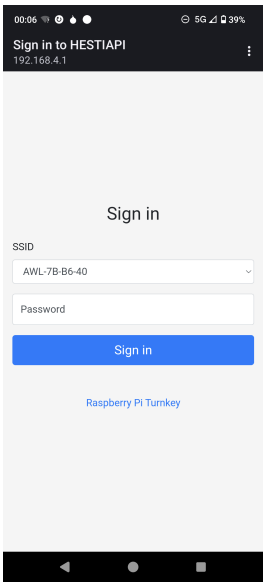


Figure 3: Screen for setting up the HestiaPi's wifi access

2.3 First Boot Setup

The first time you boot your HestiaPi, it will go through a setup process which requires it reboot several times to set everything up.

Follow the on-screen instructions on the LCD when it prompts to connect your phone to the "HESTIAPI" network with HESTIAPI as the password. Once connected you will automatically be prompted on your phone (or computer) to select your WiFi network from a list (no hidden SSID supported yet) and enter the password. If you are not automatically taken you to the wifi setup screen, open `http://192.168.4.1/` in a web browser. This should look similar to what is in figure 3.

If you entered the correct wifi info, your HestiaPi will restart and connect to your network. You should no longer see the HESTIAPI network when you look for wifi networks on your phone (or computer). The loading screen should show you the IP address of the HestiaPi while it is booting, and that should be an IP address from your home network (as opposed to being 192.168.4.1, which is what it will be if the thermostat is not connected to your wifi network).

When selecting your access point, you will need to choose a 2.4GHz network. Some routers use both 2.4GHz and 5GHz, however the Rasberry Pi inside of the HestiaPi does not support 5GHz. This should not be a problem as routers are configured to be compatible with both, however it's important to know if you have changed your router settings to disable the 2.4GHz band.

From the time you first turn it on to the time that you are ready to use the thermostat can take up to 20 minutes. While you are waiting, you may want to download a mobile app to access the thermostat without having to navigate to the HestiaPi using a web browser. See section 3.3 for more details.

When the HestiaPi first boots, the LCD UI may start with zeros for the temperature and humidity. This is normal and the data will update in a minute or two. If your HestiaPi is using a temperature sensor that does not also have the ability to sense humidity, the humidity reading will remain as 0%.

Pressing the "i" button in the upper right of the screen of the HestiaPi will display information such as the WiFi that it is connected to, the signal strength, the MAC and IP address of the thermostat, and so forth.

Once the LCD is showing the UI with temperature values, try and load the mobile app or use your phone or laptop and navigate to: http://[hestiapi_IP]:8080/start/index (substituting in your HestiaPi's IP address where appropriate) and select "Basic UI".

You should now be able to control the basic functions from either the app or your browser.

OpenHAB2 has a great forum with loads of information from fellow users who are willing to help you customize your system to suit your needs.

Please note that the UI of the app, web and LCD may change with software updates so be sure to back up any customizations you make before running an update.

3 Everyday 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 4a.

Pressing the flame icon in the upper left corner displays the heating menu. In figure 4b, 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 4c). 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 4d 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 indiacted along with the duration of the boost remaining in the bottom left corner of the heating menu, as showin in figure 4e.

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/` 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 5a.

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 maintainence tasks such as taking backups, updating the software, and shutting down the HestiaPi in case you need to do hardware maintainence.

Figures 5b and 5c 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 5d), there are some additional features which are not accessible via the touchscreen interface. This includes switching from Celsius to Fahrenheit, settings your time zone and if your system has second stage heating it can be configured here (as seen in figure 5e).

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 maintenaince 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 5f.

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

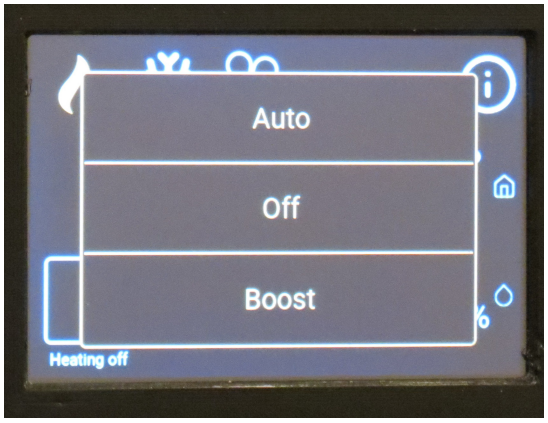
The interface for the mobile app is almost identical to the basic UI of the webpage (covered in section 3.2). The only signficiant differences are getting the application and connecting to the HestiaPi.

The application can be downloaded from F-Droid, 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 5.

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 (5.4).

(a) User Interface After Booting

(b) Heating Interface - Heat is off

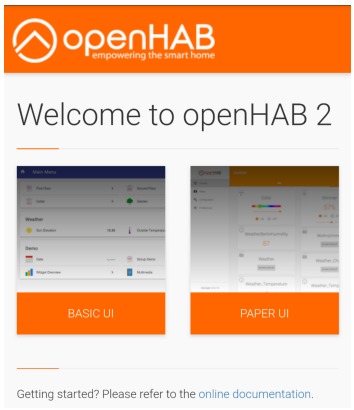


(c) Heating Interface - Toggle heat on/off

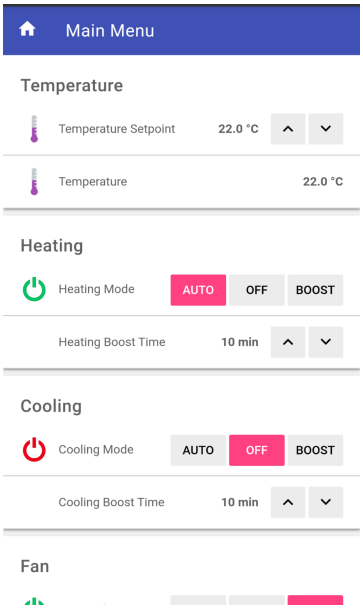
(d) Heating Interface - The heat is on

(e) Heating Interface - Boosting the heat

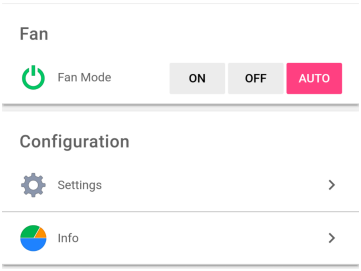
Figure 4: fig:UI Menus



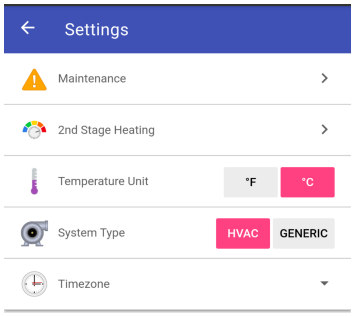
(a) User Interface Choices



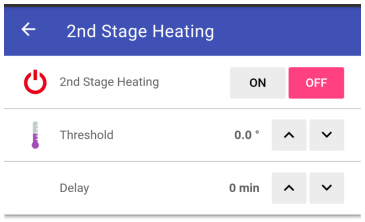
(b) Main Menu



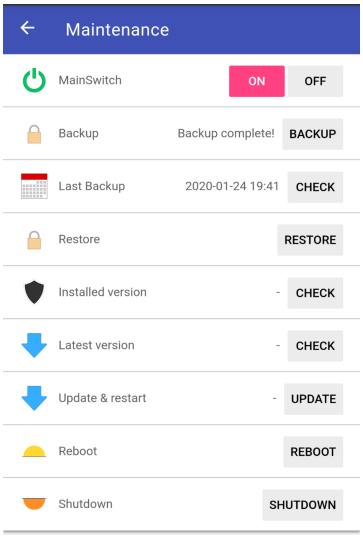
(c) Bottom of Main Menu



(d) Settings



(e) 2nd Stage Heating



(f) Maintenance Menu After a Backup

Figure 5: fig:Web UI

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

4 Troubleshooting

4.1 General issues

For those who are having problems and don't want to read the entire troubleshooting section, just try reflashing the SD card with the latest image from https://hestiapi.com/downloads. This really will fix nearly every problem! So if your issue isn't listed, or you just want a quick fix, re-flash that SD card and you'll likely be back in business in no time. If it doesn't fix your issue, use a different SD card and if that doesn't work, read the rest of this section or post on the "Get Help" section of the community forum.

4.2 Verifying relay status

If you are unsure if the relays are turning on or off when they should, this can be checked in software. To do so, you will need to be able to SSH into the HestiaPi.

To determine if a relay on, SSH into the pi and then look at the GPIO value. The mappings of which GPIO pins are connected to which relays are in section 7.3.1.

As an example, relay 1 is GPIO12, which is the fan for HVAC systems and humidity when in Generic (EU) mode. After SSHing into the HestiaPi, the command below would display a "1" if that function was on, or a "0" if it was off.

```
cat /sys/class/gpio/gpio12/value
```

If this reports that the relay is on, but the the fan (or humidity for EU systems) is not on, it indicates that there's likely an issue with wiring or with your HVAC system.

If this reports that the functionality is off, when you think it should be on, you may want to manually turn it on. After SSHing into the HestiaPi, you will need to become root, and then set the GPIO value. To continue with the fan example on HVAC systems, here are commands you would run after SSHing into the pi in order to turn the fan on, and then turn it back off again.

```
sudo su -
echo 1 | /sys/class/gpio/gpio12/value # turn on
echo 0 | /sys/class/gpio/gpio12/value # turn off
```

As soon as you run the "echo 1" command, the fan should turn on. There should not be any delay. The same should be true when running the command to turn off the relay.

Turning on relays in this manner bypasses the thermostat logic, which means that the function will never turn off. That's why it's important to make sure to manually turn off the functionality after you manually turn it on. If you forgot to turn off the cooling system after manually turning it on, it'd likely burn out the compressor, which would be a very expensive repair.

Running these commands should help you in determining if there is an issue with the hardware or with the software. If these command do not cause the functions to turn on, it points to a hardware issue, such as incorrect wiring.

If these commands work as expected but the heat does not turn on when the heating set point is higher than the temperature, and the heating system is turned on, it indicates that something is going wrong in the software (more specifically, in OpenHAB). If that's the case, you can get help on the forum at https://community.hestiapi.com/. Please be patient, as we are all volunteers and it may be a few days before we see your message and have time to investigate and reply.

4.3 Temperature doesn't show up on the LCD screen

The most common cause of this problem is that the Pi can't get the temperature value from the temperature sensor. This will likely cause the heating and cooling to not turn on, but the fan may turn on (e.g. when using boost). For information about how to determine if your relays are working, see section 4.2. Confirming that the relays work properly and that you are only dealing with an issue of temperature sensing can be helpful in keeping troubleshooting as simple as possible.

The two reasons we've seen for this problem are both hardware failures. It's always been either a faulty sensor, or a faulty i2c controller on the Pi. Unfortunately there's no way to tell which it is other than to replace one of these two components and see if it fixes the issue. However, if you are comfortable using SSH to log onto the HestiaPi, you can run some commands to confirm that the problem has been correctly diagnosed before buying additional hardware.

The command `i2cdetect -y 1` should show you a grid which should have the number 76 or 77 in it. If the grid is all dashes, it indicates that you are running into this problem of faulty hardware. In that case, running the command `/home/pi/scripts/getBMEtemp.sh` is expected to result in an I/O error. As a final check, if you look at `/var/log/openhab2/openhab.log`, you should see the same error message you saw when you ran `getBMEtemp.sh` manually.

Of the failures we have seen, it seems to be evenly split between the problem being on the sensor side versus being on the pi side. Our suggestion would be to replace whichever component is cheaper first.

4.4 Eternal loading screen

If your thermostat ever reboots and the loading countdown on the screen goes all the way down to the point where it just says "Loading..." and stays there for hours, it's either a software issue or a slow micro SD card.

First, reflash the SD card and this will almost certainly have you back up and running.

If that does not work, the only other potential cause we have seen is using an old, generic, or slow SD card. Using a card rated at 10MB/sec should be sufficient, and 30MB/sec would be ideal. Unlabeled cards, or ones rated at lower speeds may work. We have not found a card that will consistently cause this issue, so we can't say for sure what the minimum acceptable speed is.

An icon that looks like a 10 inside a circle, a 1 inside a letter U, or a V10 are all icons that mean it's 10MB/sec. A 3 inside a U, or V30 indicated it is a 30MB/sec card. There are faster cards, labeled V60 and V90, but getting a card that fast is not necessary.

4.5 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 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.

4.6 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.hestiapi.com.

4.7 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.gsugambit.com/w/1AhKQByTrAg38HKSQUJFR3

4.8 Boot loop

We have seen faulty SD cards cause a boot loop. First, try re-flashing the latest HestiaPi image onto the SD card. If that doesn't solve the issue, try doing the same with a different SD card. This has always resolved the problems that we've seen thus far.

4.9 Resets when heat turns on

This was only ever reported by one user and the issue turned out to be a faulty SD card. Using a different SD card and flashing it with the latest image got them back up and running smoothly.

4.10 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.

4.11 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.

5 Customization

5.1 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 5.1.1.

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-HESTIAPI-IP]:8080/paperui/index.html (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 publishs 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 avilable 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!

5.1.1 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 mean 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 5.1 for more information on how to do this.

5.2 Remote Access

5.2.1 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/index.html 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

5.2.2 Traditional remote access

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/index.html`

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.

5.2.3 Update Your DynDNS Automatically

By default, the HestiaPi does not connect to any external servers. If you want the thermostat to tell a dynamic DNS service about the its publis IP address, the getpublicip.sh script can help you do just that.

The file is already on your HestiaPi at:

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

You will need an account with whatever service you choose to use inside the script, and the will need to be modified to specify your username, password and possibly domain name. There are some example in the script near the bottom which are commented out. Customize and uncomment them and then rename the script to `getpubliship.sh` and you should be good to go. This file is already run periodically by OpenHAB.

Note: Old versions of the HestiaPi software automatically reached out to ipinfo.io to determine your public IP address. For these versions, the script is already named `getpublicip.sh` and so you just have to edit that file to get it to report the public IP to your dynamic DNS provider.

5.3 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 HestiPi 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.

5.3.1 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.

5.3.2 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:

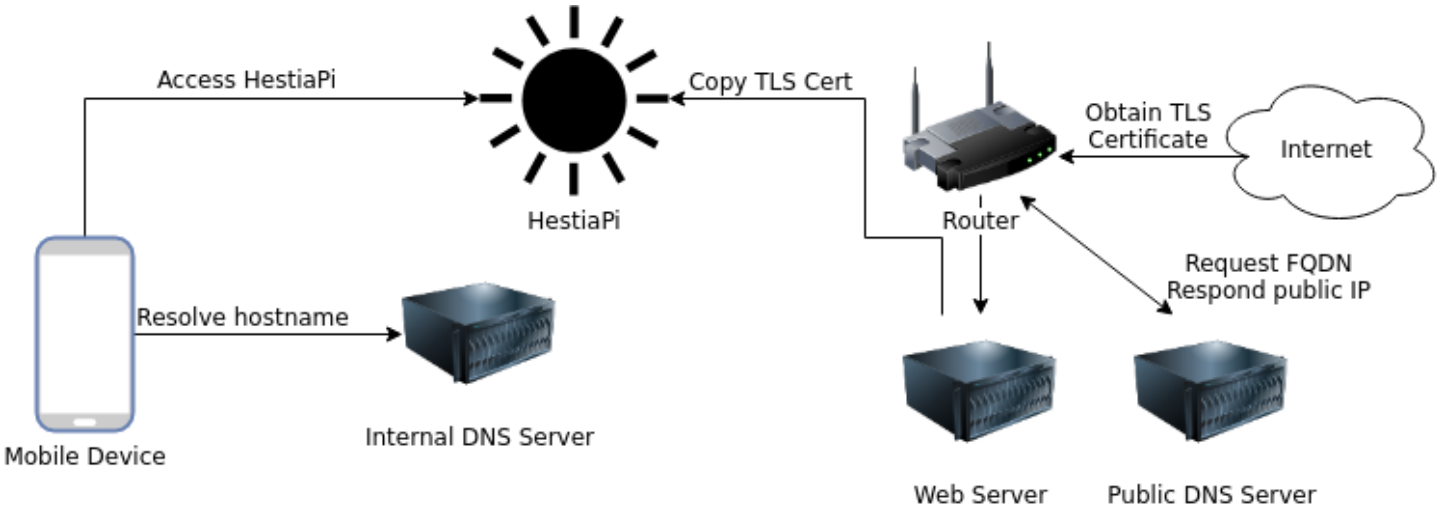


Figure 6: Complicated TLS Setup Overview

```
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 staticly 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).

5.4 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 6 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 (LetsEncrypt 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.

5.4.1 Static IP

To assign the HestiaPi a static IP address, see section 5.3.

5.4.2 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 instide 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/`

5.4.3 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 5.4.2).

The next step is to set up a Linux server which will act as the web server. Once Linux is intalled, 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.

5.4.4 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 encryped connection to your HestiPi!

6 Build Your Own HestiaPi

6.1 Software

6.1.1 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.

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)

6.1.2 Manual Installation

For people who want to install everything step-by-step, or just see how it is done, the process is automated in https://gitlab.hax0rbana.org/public-repos/raspberrypi-automation/.

That repo is not specific to the HestiaPi, it automates building multiple Raspberry Pi images that someone might want for their projects. The main file that is relevant to the HestiaPi project is hestiapi.yml.

The README file in that repo has the latest information on exactly how to build the image, but the general process to build this at the time of writing is as follows:

```
# Install the necessary software
apt-get update && apt-get install -y ansible xz-utils ansible

# Get a fully patched Raspberry Pi image
PLAYBOOK=upgrade.yml ./build_emulator.sh arm1176 bullseye wrapper_ansible.sh

# Configure that image to be a HestiaPi
PLAYBOOK=hestiapi.yml EXTRA_VARS="hestiapi_version=v1.4-dev" \
        ./build\_emulator.sh arm1176 bullseye wrapper_ansible.sh

# At this point you will have your HestiaPi image at the following location
ls -l qemu-rpi/2022-09-22-raspios-bullseye-armhf-lite.img
```

These instructions were tested on Ubuntu and should work on any Debian-based Linux distribution.

6.2 Hardware

6.2.1 Bill of Materials (BOM)

This is the bill of materials to make your own HestiaPi. The AC/DC converter allows your HestiPi 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 OSH Park
2. Raspberry Pi Zero W AdaFruit Amazon or Raspberry Pi Zero 2 W AdaFruit
3. 3.5 inch touch screen for Raspberry Pi Amazon
4. 4 relays capabile of handling 24V / 2A – Panasonic AQG22205 Mouser
5. At least 45 standard male 0.1" headers – AdaFruit
6. Reset switch – TL1105WF160Q Mouser
7. A six-post terminal block – TB003-500-P06BE Mouser
8. Temperature sensor – a BME280 sensor (includes humidity sensing) SparkFun or a BMP280 sensor (does not include humidity) Mouser
9. Cable for temperature sensor Amazon
10. LCD headers, 2x13 extra long headers Mouser
11. AC/DC converter – Amazon

6.2.2 PCB Soldering

6.2.3 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 2.2.

1. Solder standard headers onto PCB for the pi (figure 7) 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 8)
3. Put terminal block and sendor headers in place and solder them on (figures 9 and 10)
4. Solder relays in place (figure 11)
5. Solder the headers onto the BME board (figures 12 and 13)
6. Solder switch in place (figure 14)
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 15 and 16)
8. Solder the LCD headers into place and connect the BME280 board (17)
9. If using the AC/DC converter, solder that on

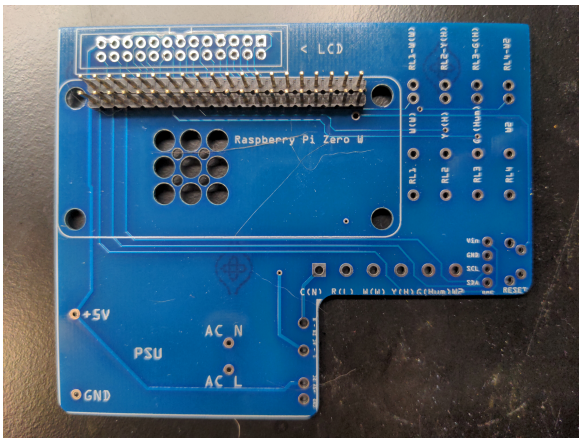


Figure 7: Headers for the pi, don't forget the reset pin like I did here!

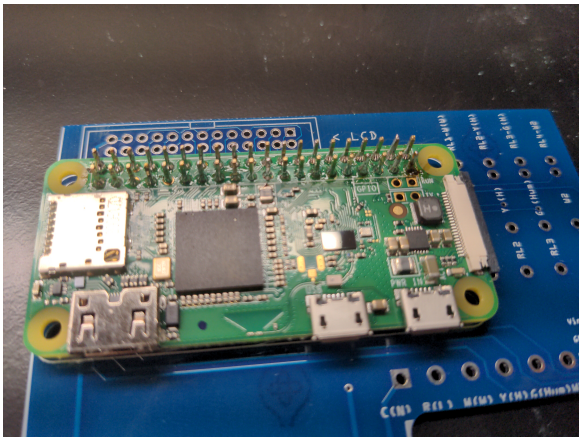


Figure 8: Pi soldered on, sans the reset pin that I had to go back and patch up later

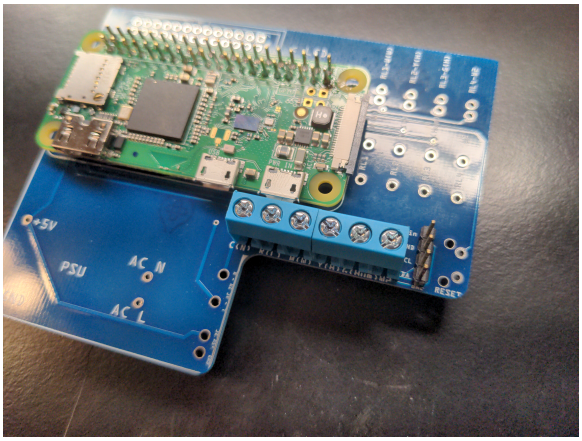


Figure 9: Terminal block and sensor headers in place

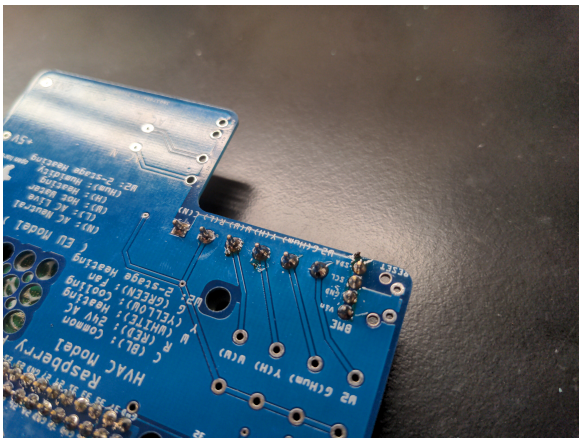


Figure 10: Terminal block and sensor headers soldered on

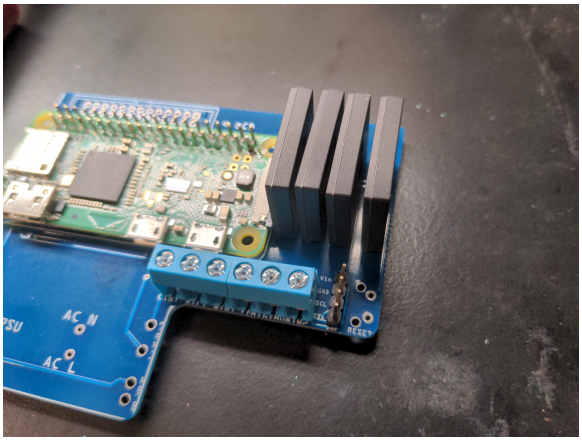


Figure 11: Relays soldered into place

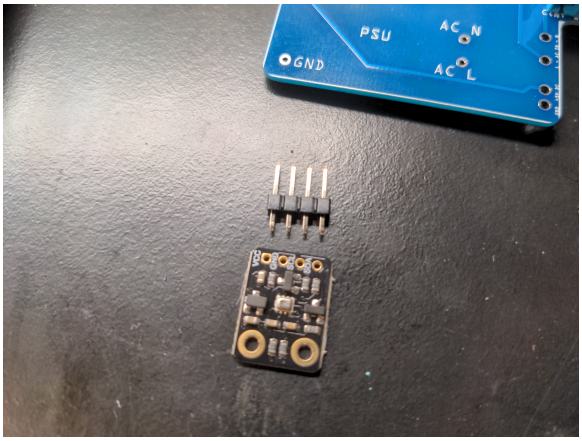


Figure 12: BME needs to have headers attached

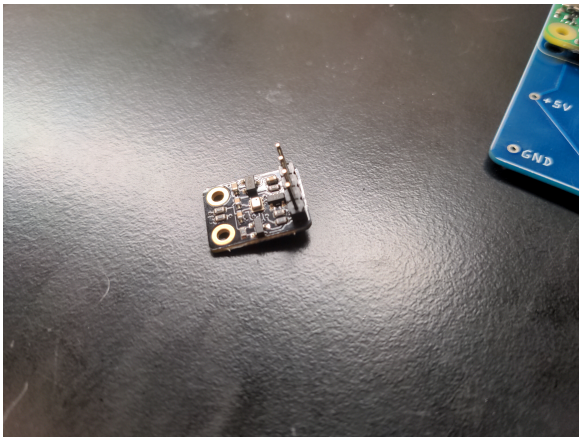


Figure 13: BME with headers soldered on

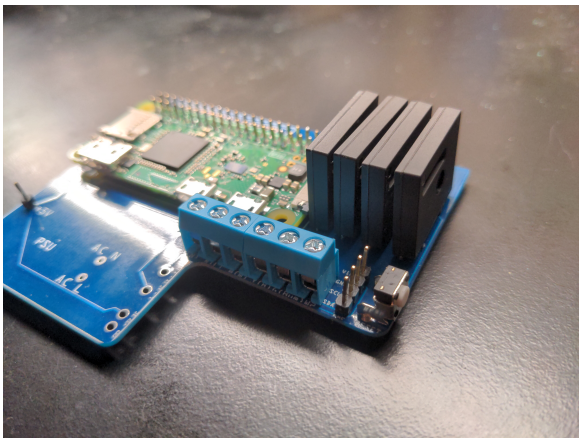


Figure 14: Solder switch into place

Figure 15: Pin placement adjusted to have 12mm above the board

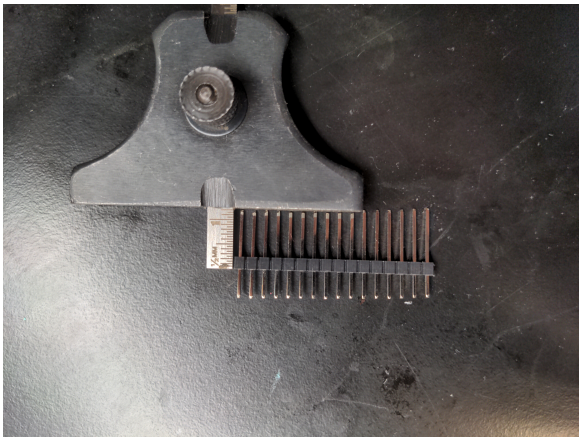


Figure 16: All headers adjusted to be 12mm above the board

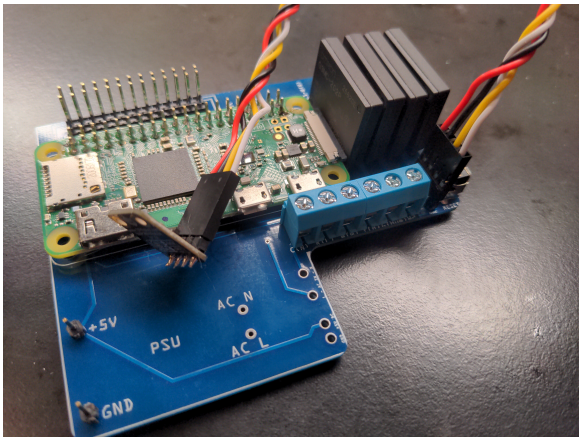


Figure 17: Fully assembled HestiaPi with headers for a DC power supply

6.2.4 Video

6.2.5 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.

6.2.6 Troubleshooting

After assembling you 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.

6.2.7 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.

6.2.8 Files

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

6.2.9 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 :)

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.

7 Reference Information

7.1 SSH and MQTT Information

7.1.1 Default SSH Username and Password

Username: pi

Password: hestia

SSH port: 22

If you are not familiar with how to SSH into a server, there's a guide available here: https://www.howtogeek.com/311287/how-to-connect-to-an-ssh-server-from-windows-macos-or-linux/

7.1.2 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/#
```

7.2 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.

Systemd starts up a number of services, including:

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

The status of sll 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.

7.3 File Structure & Paths

7.3.1 GPIO pin mappings

Relays

The relays are used for different things depending on whether the thermostat is in HVAC mode (aka US mode) or Generic mode (aka EU mode). The default is US mode, so unless you changed it, this is what you can expect.

	Pin	**GPIO**	US	EU
Relay 1	32	GPIO12	Heating (W)	Hot Water (W)
Relay 2	16	GPIO23	Cooling (Y)	Heating (H)
Relay 3	12	GPIO18	Fan (G)	Humitity (Hum)
Relay 4	36	GPIO16	2nd stage heating (W2)	2nd stage heating (W2)

The letters in parenthesis is the symbol that is used in the wiring diagrams for climate control systems. For more information on that, see section 2.2 about installation.

Temperature sensor

The BME or BMP sensor can be at either I2C address 0x76 or 0x77. The software will check both addresses, in this order.

	Pin	GPIO
SDA	3	GPIO02
SCL	5	GPIO03

LCD touchscreen pinout

Pin no.	Symbol	Description
1, 17	3.3V	Power positive (3.3V power input)
2, 4	5V	Power positive (5V power input)
3, 5, 7, 8, 10, 12, 13, 15, 16	NC	Not connected
6, 9, 14, 20, 25	GND	Ground
11	TP_IRQ	Touch Panel interrupt, low level while the Touch Panel detects touching
18	LCD_RS	Instruction/Data Register selection
19	LCD_SI/TP_SI	SPI data input of LCD/Touch Panel
21	TP_SO	SPI data output of Touch Panel
22	RST	Reset
23	LCD_SCK/TP_SCK	SPI clock of LCD/Touch Panel
24	LCD_CS	LCD chip selection, low active
26	TP_CS	Touch Panel chip selection, low active

7.3.2 Configuration files

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`