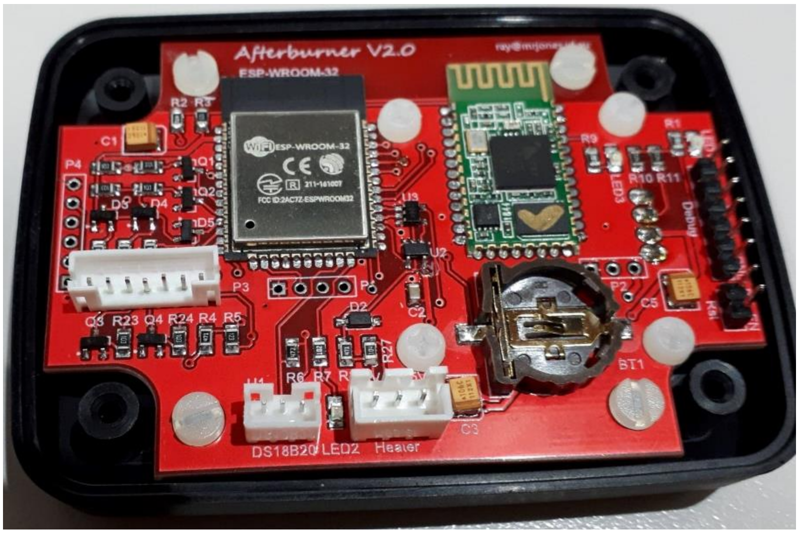
The Afterburner *(powered by chilly)*

USER MANUAL



A Bluetooth / Wi-Fi enabled controller for Chinese Diesel Heaters

Congratulations on your acquisition of the Afterburner.
This device allows you unprecedented remote controllability of your compatible Chinese diesel heater, and detailed examination of the run time conditions.

DISCLAIMER

The software of this design is released under the GPL V3 licence and is offered under the following disclaimer:

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

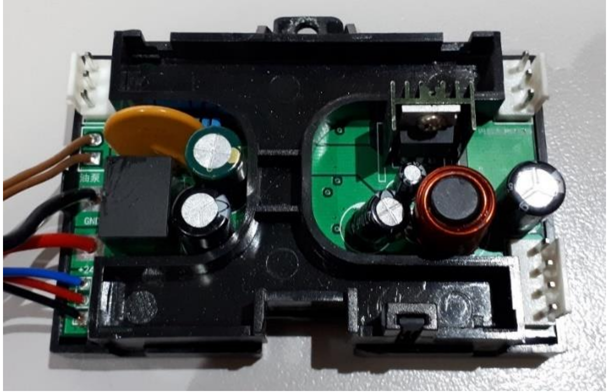
You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

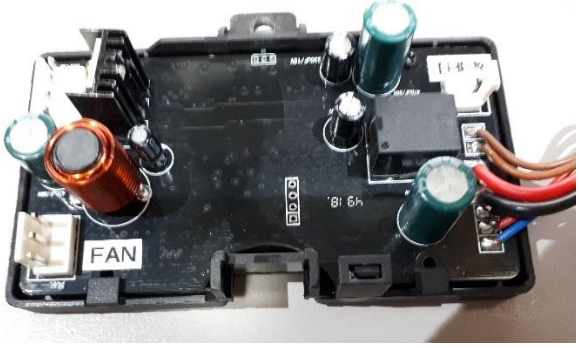
No responsibility will be taken for any damages caused by operating the Afterburner with your diesel heater system.

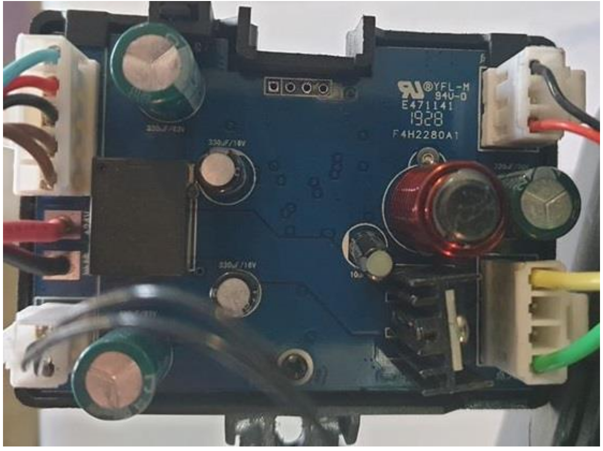
COMPATIBILITY

The Afterburner only works with compatible digitally controlled Chinese Diesel heaters.
Not all heaters are digitally controlled.

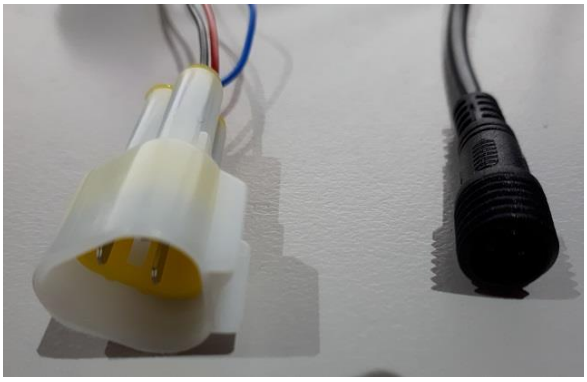
The best discriminator is to inspect the control board housed within the heater unit.
If it looks like either of the following, then your heater is compatible and will work with this controller.



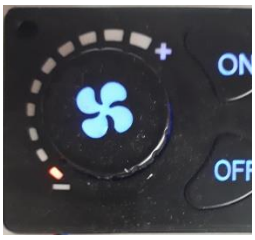


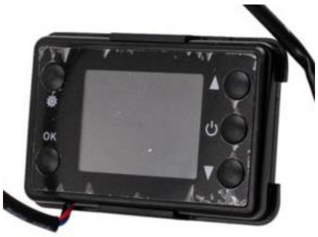


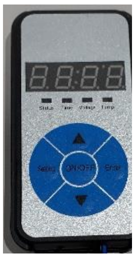
Note that they only have 3 wires to the head control unit, typically being red black and blue, which ultimately terminate in either a triangular or round waterproof connector:

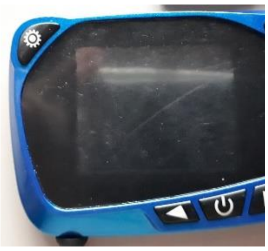


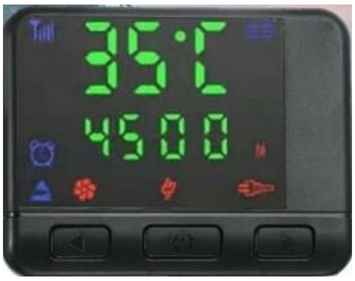
The following controllers are compatible.

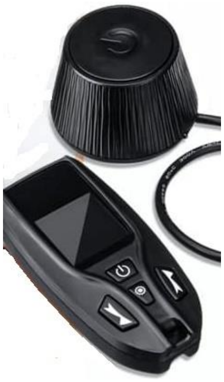


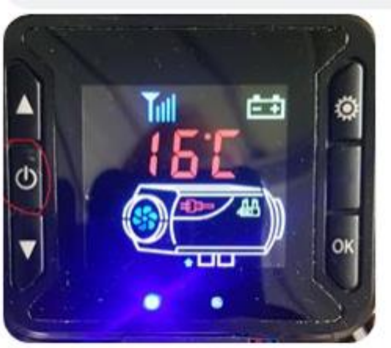




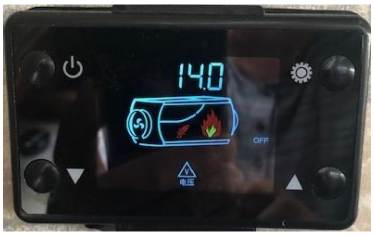


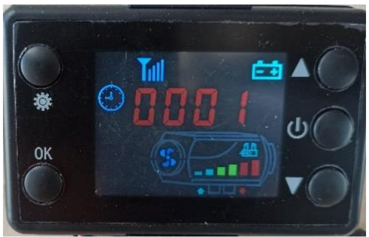


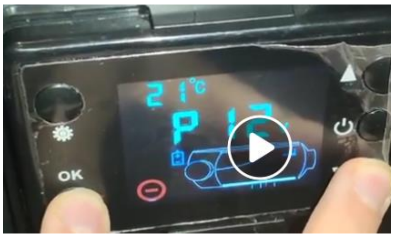


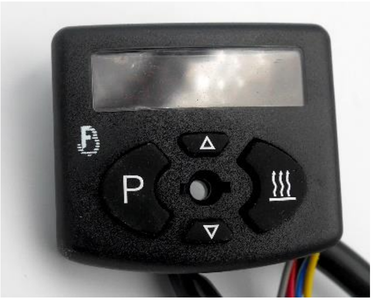


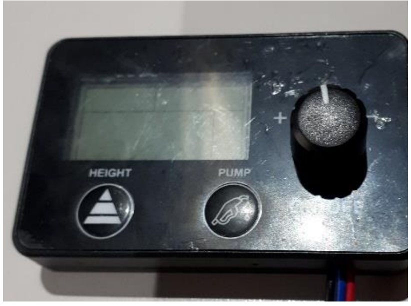
The following are NOT compatible

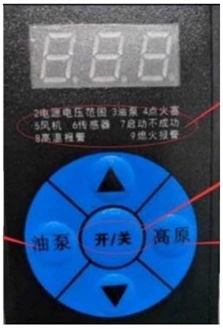


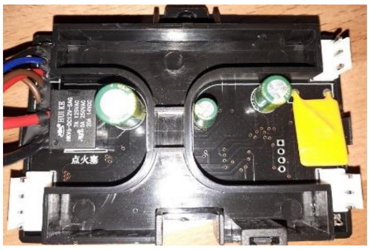


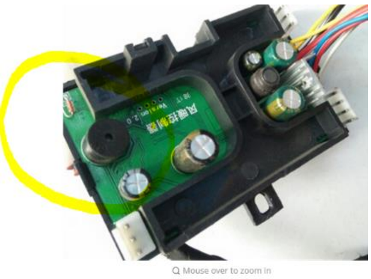






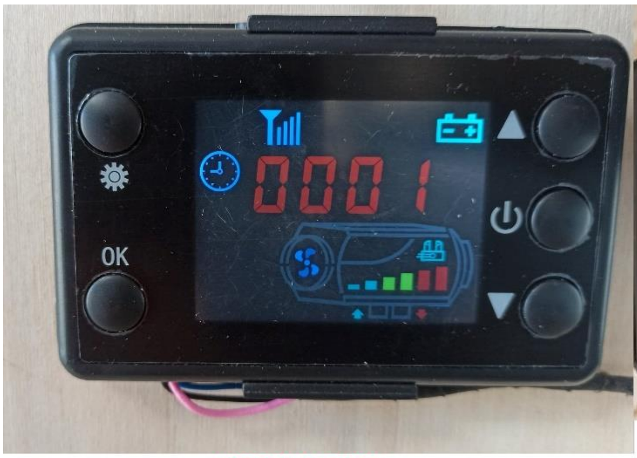






Black LCD uncertainty

Sadly, the astute will note there is a black LCD in both the compatible and not compatible lists. Please refer to the following side by side comparison:

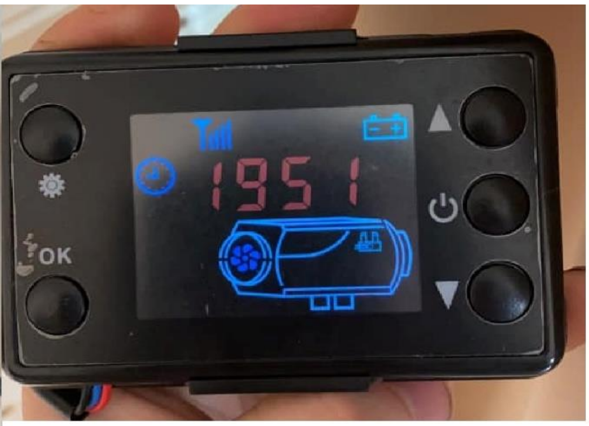


NOT COMPATIBLE

OK is slender thinner stroke

Triangles are more like an equilateral. (Bases are wider)

On/Off symbol vertical bar goes down further



COMPATIBLE

OK is sqatter, more rotund

Triangles are pointier

On/Off symbol is a shorter bar

Contents

Introduction

The software of the Afterburner is open source, leveraging upon the Arduino platform and Espressif ESP32 IDF add on.
The source code repository: https://gitlab.com/mrjones.id.au/bluetoothheater

Two variants of PCB have been produced.

1. The 1st production PCB has a green solder mask and is open source.
2. The new Mk2 PCB, and later variants, was designed to fit within an off the shelf ABS case.
   The solder mask of this PCB is red.
   These PCBs are proprietary and only available as completely assembled and tested units.

The full software capabilities can be fully exercised with either PCB.

The primary difference of the later proprietary PCBs is additional circuitry to provide buffered input and output signals for the purposes of user expansion:

- 2 contact closure inputs (to ground).
- 2 open collector transistor outputs, to drive indicators or relays.
- A single analogue input.
- Header position for a 433MHz receiver module
- Header position for a 433MHz transmitter module

- Later PCBs (V3.1+) provide a location to readily expand the I2C bus

With either PCB variant, the digital inputs are made active by connecting the exposed sense line to ground.

Mk1 PCB –software and hardware expansion – JTAG debug & development

If you are keen to expand the functionality and contribute code, the spare unused pins are exposed to a 7-pin header on the Mk1 PCB.

CAUTION: these are raw unconditioned ESP32 pins.

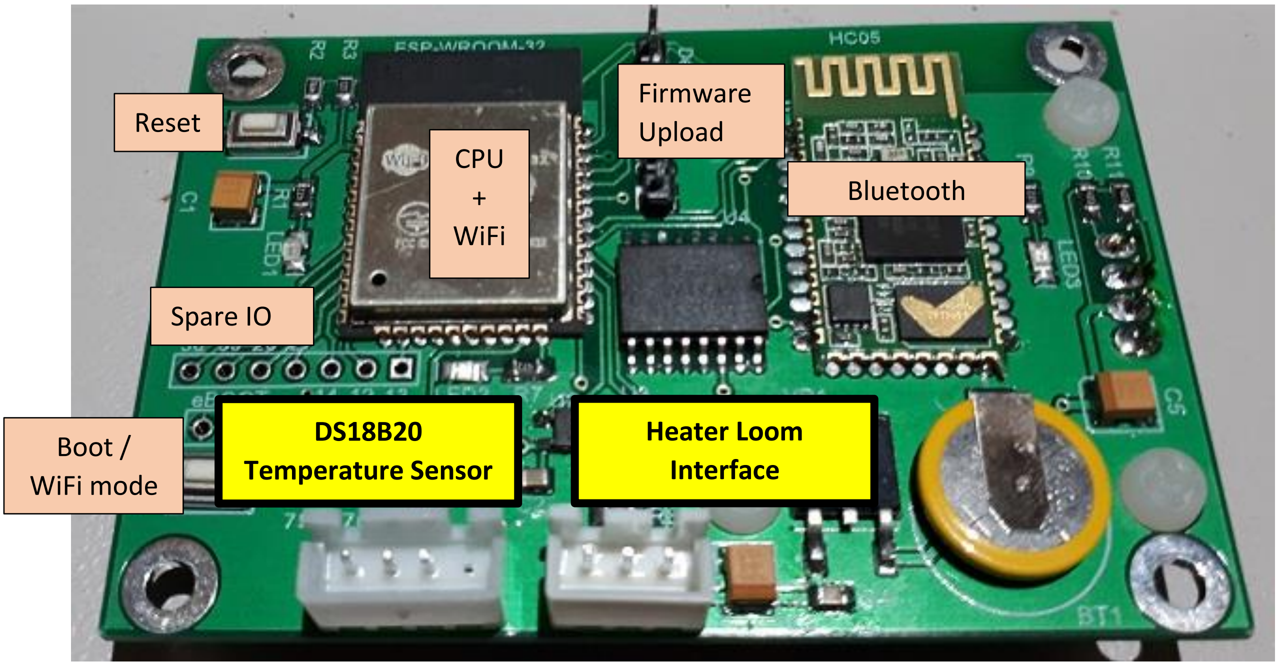
These uncommitted pins allow currently unsupported hardware to potentially be added, eg a GSM data module.
Keeping in mind that the ESP32 also features a peripheral pin multiplexer, any internal peripheral can generally be steered to these pins, even redirected from their existing purpose on the design, eg the debug serial port or Bluetooth module.
Further information can be found in the ESP32 technical guides by Espressif.

The raw exposed pins also allow the use of a JTAG debug tool, but some rework of the DS18B20 temperature sensor circuit is required to achieve this.

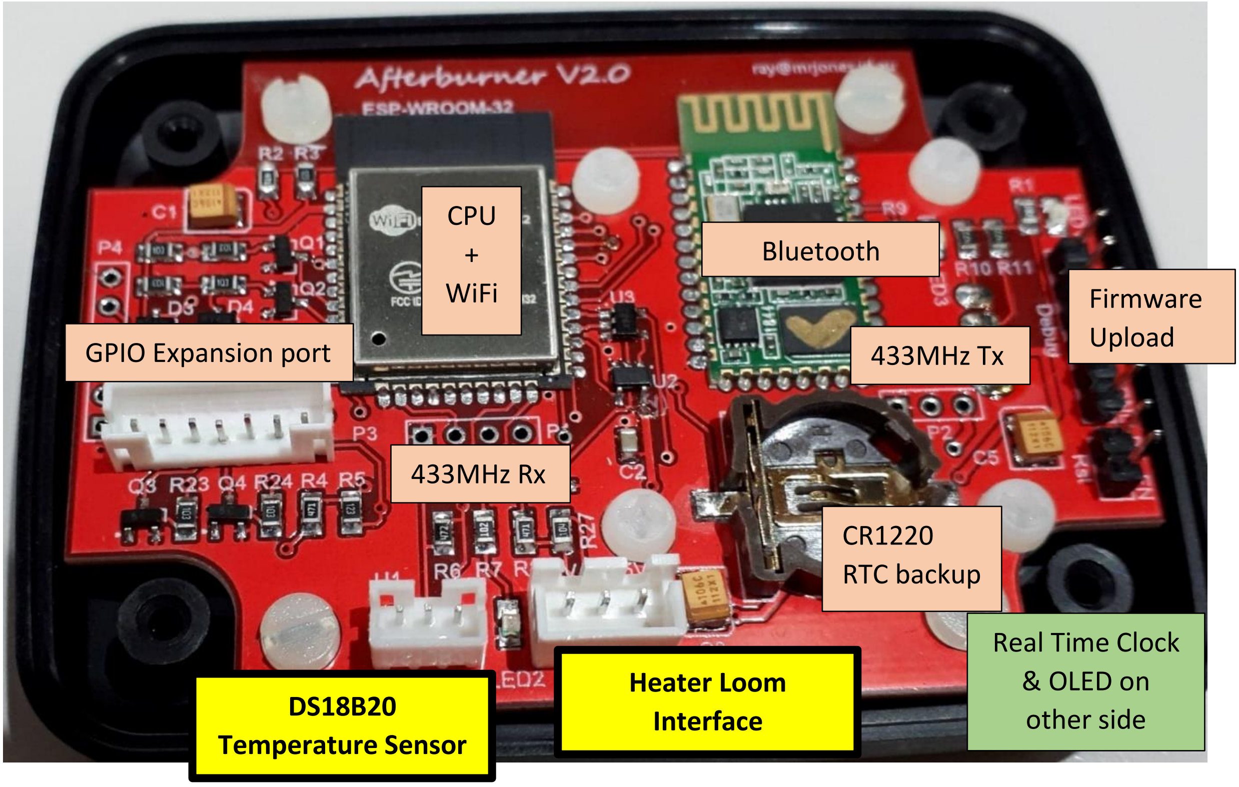
The use of GPIO and JTAG is mutually exclusive.

Hardware overview

Mk1 PCB



Mk2 PCB



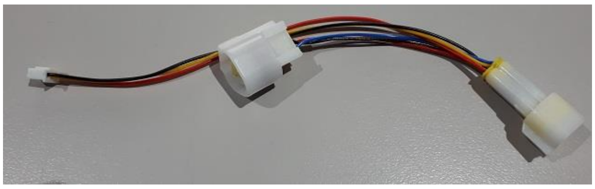
Both PCBS have identical core componentry. The physical locations of parts has been shifted to accommodate the case standoffs in the Mk2 PCB.

Extra components on the Mk2 PCB exist for the GPIO expansion port buffering.

The Afterburner has two primary connections to the outside world:

- Heater communications
- Temperature Sensor

Heater Loom interface



The smaller triangular male connection plugs into the existing heater wiring loom in place of the OEM controller.
The larger female connector is only needed for initial installation to learn the settings from the previous controller.

The Afterburner termination is a 3-way JST-XH (2.54mm pitch) connector.

Temperature sensor

An encapsulated DS18B20 is used as the temperature sensor.

This discrete unit can be mounted wherever desired away from the Afterburner.

The Afterburner termination is:

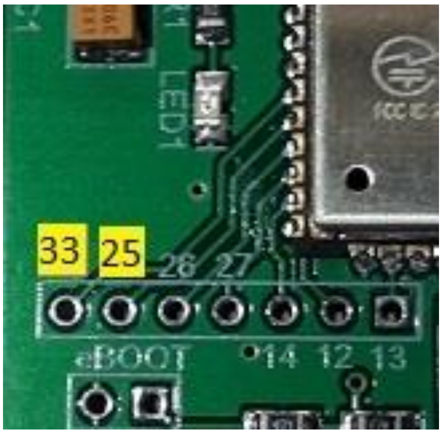
Mk1: 4-way JST-XH (2.54mm pitch).

Mk2: 3-way JST-PH (2mm pitch).

It is not recommended to install this sensor within the case as heat radiated from the ESP32 will distort the temperature readings.

MK1 PCB - Unused ESP32 pins

Spare IO



Seven unused pins of the ESP32 are brought out to a 7-way header location.
These can be used for customised extensions if so desired.

A neat feature of the ESP32 is it hosts a pin multiplexer. Generally, any internal peripheral can be steered to these pins!

These pins are also most of the signals, bar one required to use the JTAG debug tool.

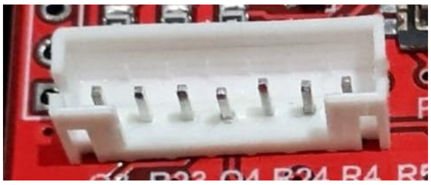
Note:
The silk screen overlay on the initial production of Mk1 Afterburner PCBs erroneously labelled the first two pins, the correct numbering is the highlighted numbers in the image above.
The functionality of the circuit is not affected.

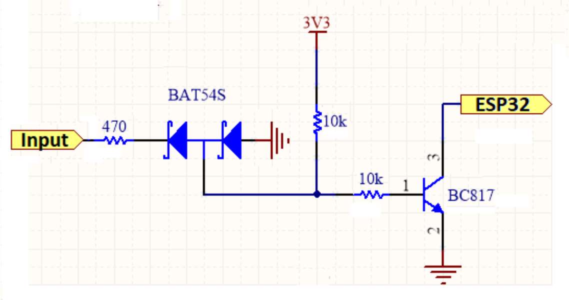
Mk2 GPIO Expansion port

A 7-pin JST PH header is provided to allow

- 2 contact closure inputs
- 2 open collector outputs
- 1 analogue input
- 5V Power & ground sources

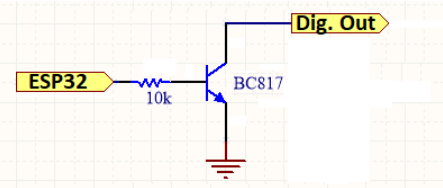
Wire Colour	**Purpose**
Blue	Digital input #1
Green	Digital input #2
Orange	Digital output #1
Yellow	Digital output #2
Black	Ground, 0V
White	Analogue input
Red	Power, +5V (from heater ECU)





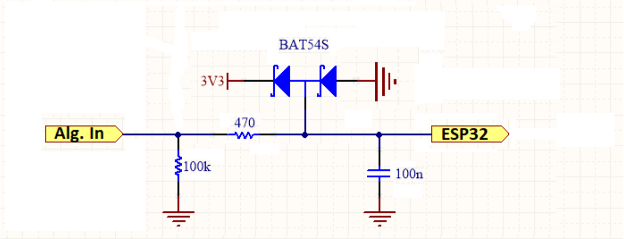
Diode array:
Blocks +ve external voltages.
Shunts -ve external voltages to 0V.

External input is pulled up by 10kΩ resistor; transistor conducts, logic 0 at ESP.
Contact closure to 0V on input; transistor stops conducting, logic 1 at ESP32.
(ESP32 uses internal pullup)



ESP32 drives pin high to turn transistor on.

Open collector can drive a relay coil (back EMF protection required) or LED.
Either powered from 5V or external power source. (0V commoned)



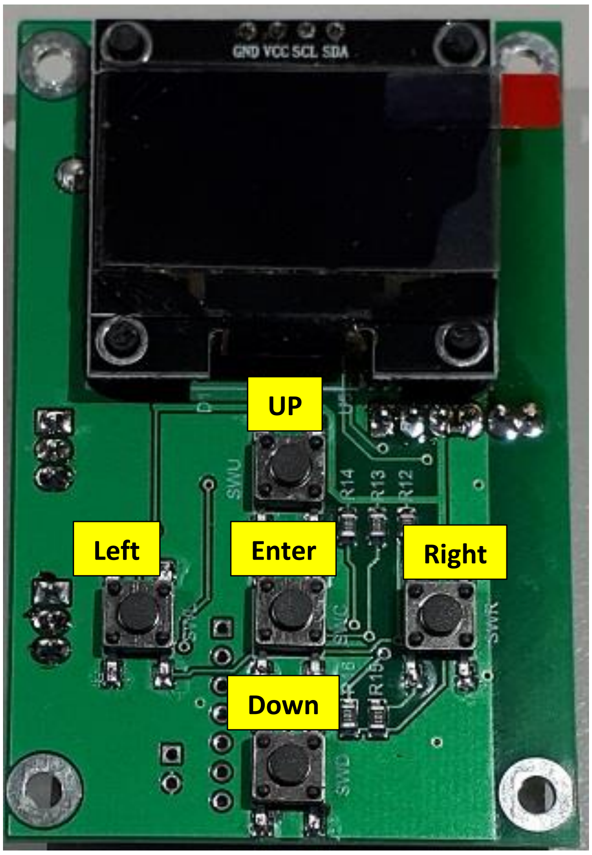
Diode array clamps to 3.3V or 0V.

100k pull down only exists on V2.1+ PCBs.

Series 470Ω resistor limits clamping current via diodes, and minor filtering with 100n capacitor.

User interface

The physical user interface is on the rear of the PCB, hosting a 1.3” 128 x 64 OLED display, and a 5 button key pad.



The OLED display is above the keypad, the buttons forming an Up/Down, Left/Right, Enter configuration.

Menu System

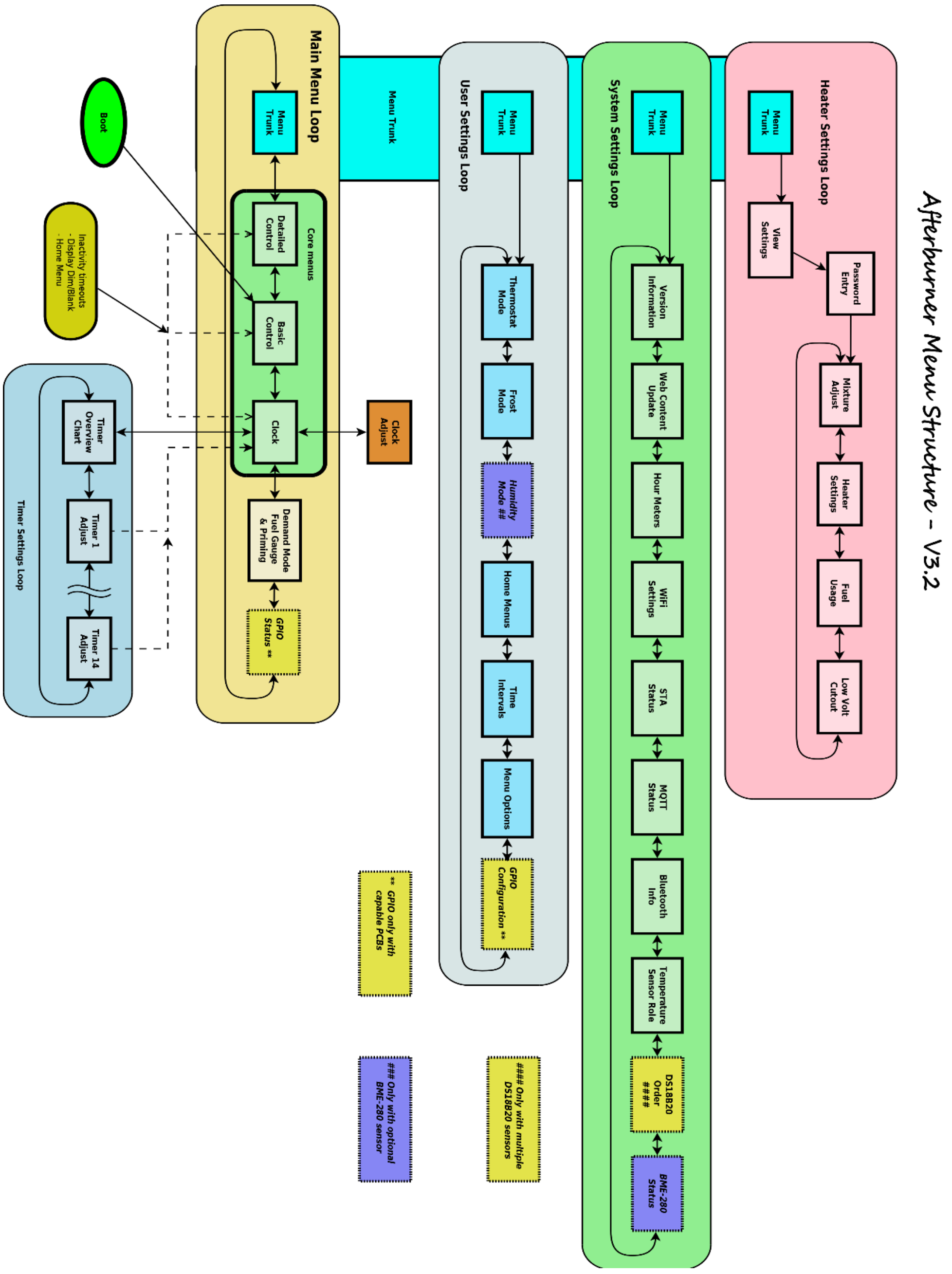
OLED Menu Structure

The Afterburner hosts a local user interface comprising of a 1.3" 128x64 resolution OLED display, and a 5 button keypad beneath the OLED display.

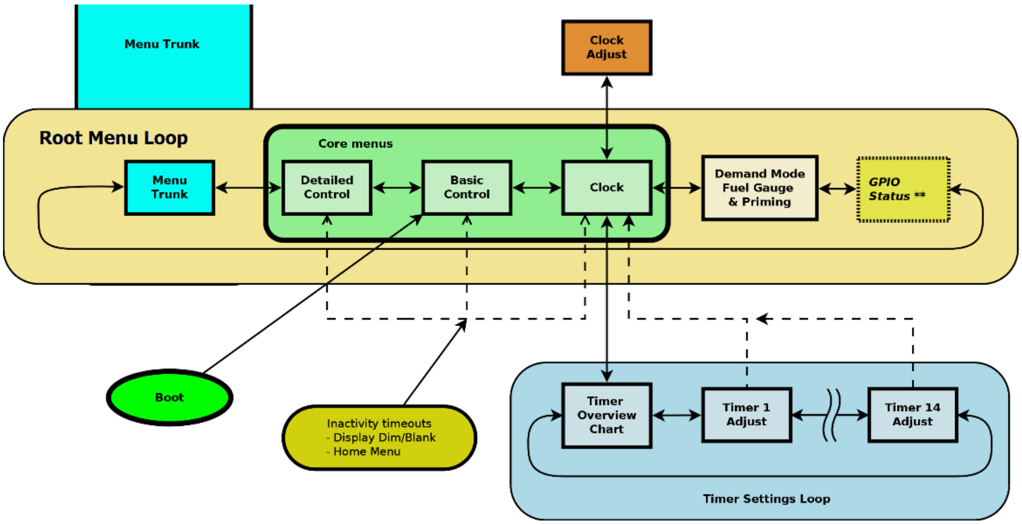
The keypad buttons are referred to in the menu descriptions as **LEFT**, **RIGHT**, **UP**, **DOWN** & **CENTRE**.

The OLED display can present numerous menus used to monitor/control the heater, or alter settings such as fuel mixture, timers etc.

The general topology of the menu structure is as follows:



Root Menu Loop



The normal user Interaction with the heater system occurs via the menus available in the *Root Menu Loop*.
The primary menus to start and stop the heater are the **Basic Control** and **Detailed Control** menus.

The Afterburner always boots into the *Loop* and selects the **Basic Control** menu.
This is indicated by the bright green bubble at lower left of the menu diagram.

The **Basic Control** menu provides simple control of the heater and will be described later.

Menu Navigation

Each menu in the *Root Menu Loop*, and other menu loops, can be navigated to by using the **LEFT** or **RIGHT** keys.

The menu trunk screen allows navigation away from the *Root Menu Loop* into another menu loop.

The **Clock** menu screen has its own associated menu loop.
e.g.: pressing **DOWN** from the **Clock** menu will shift control to the *Timer Settings Loop*.
Pressing **LEFT** and **RIGHT** now selects the available menus within that loop.

Applying settings

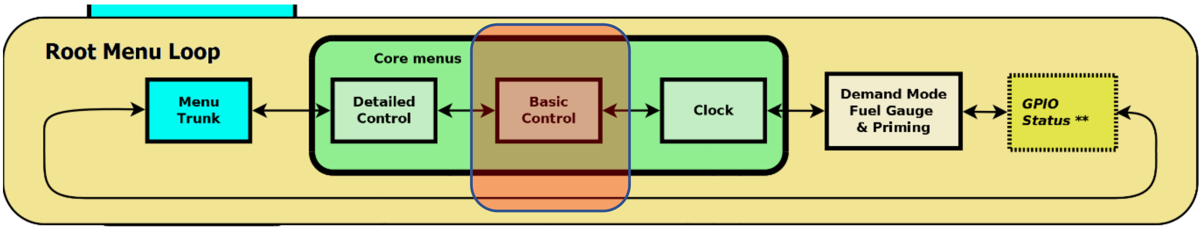
If values need to be applied, this is generally achieved by pressing the **CENTRE** keypad button.

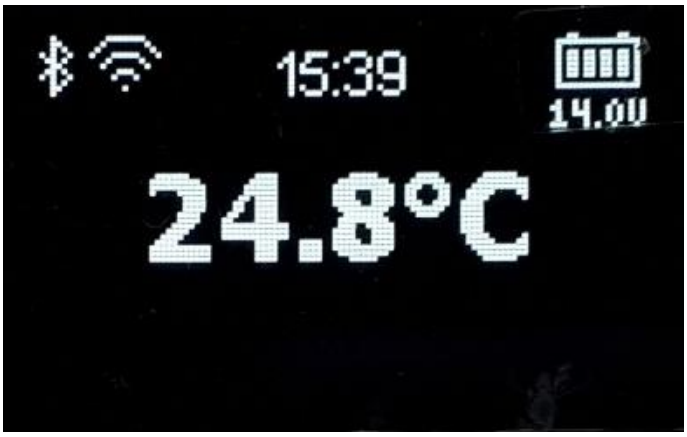
Inactivity timeout

By default, if keypad has been left idle for a minute, the display is dimmed.
Additionally if the currently selected menu is not associated with one of the core root menus (circled in green on the *Root Menu Loop*) (**Detailed Control**, **Basic Control**, or **Clock**), the menu selection will revert to one of those three core root menus.
The timeout behaviour can be altered within the *User Settings Menu Loop*.

Importantly, once the display has dimmed (or disabled), the initial press of any button is not passed onto the user interface.
Instead upon the initial keypress the display is immediately restored to normal brightness.
Consequential keystrokes will then be delivered to the user interface unless the inactivity timeout once again elapses.

Basic Control menu





This menu is always the first menu presented upon powering up the controller, and the default timeout menu unless menu selection has branched from the other core root menus.

The **Basic Control** menu allows basic operation of the heater to take place:

- Heater On/Off
- Temperature adjustment
- Basic status and fault monitoring
- Thermostat or Fixed power modes
- Celsius or Fahrenheit display
- User mode GPIO output manipulation

GPIO output manipulation

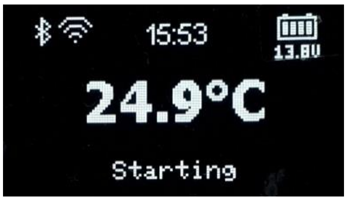
You can manipulate the user mode GPIO outputs by holding the **LEFT** or **RIGHT** keys down for 1 second.

Heater On / Off

The heater can be set to run, or stop, by holding down the **CENTRE** keypad button for around 1 second.

Starting the heater

If the heater is off, hold down the **CENTRE** keypad button.
A "***Starting***" message will then appear at the bottom of the screen.



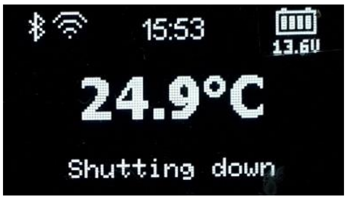
It is possible the heater will not start and a message will be brought up accordingly.
Typical causes being:

- Ambient temperature is already higher than the desired thermostat setting.
- The battery voltage is lower than the Low Voltage Cutout setting.
- The accumulated fuel usage suggests the fuel tank is nearly empty.

Once fully ignited and running, ***"Starting"*** will change to "***Running***", which indicates the heater is now obeying the user's desired thermostat setting, or fixed power demand.

Stopping the heater

If the heater is on, hold down the **CENTRE** keypad button.
A "***Shutting down***" message will then appear at the bottom of the screen.



15:53 13.6V

24.9°C

Shutting down

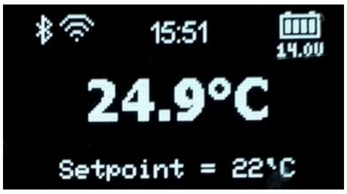
During this time, the heater will stop the fuel delivery and reheat the glow plug to help expel unburnt fuel. The fan will continue to run and may speed up.
This process takes about a minute to complete.

The message then changes to "***Cooling***" and the fan will continue to run until the heater has cooled sufficiently *(@ 55°C body temperature)*.

Finally, the messages will disappear once the heater has completed the shutdown sequence.

Temperature adjustment

The desired thermostat temperature, or fixed power setting, can be altered by briefly pressing **UP** or **DOWN** to increase or decrease the setting which is shown at the bottom of the screen.
The current setting can also be observed by briefly pressing the **CENTRE** button.



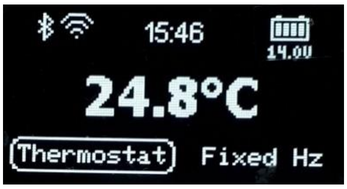
15:51 14.0V

24.9°C

Setpoint = 22°C

Thermostat or Fixed power selection

Holding the **DOWN** keypad button will enable the selection of Thermostat or Fixed power modes. The active mode will be surrounded by a selection loop, which can then be changed by using the **LEFT** or **RIGHT** keypad buttons.

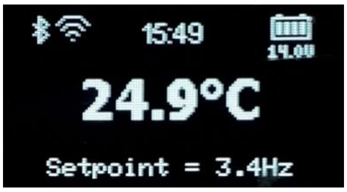


15:46 14.0V

24.8°C

Thermostat Fixed Hz

If Fixed mode is selected, the power demand is the frequency the fuel pump will run at, in Hertz. eg:



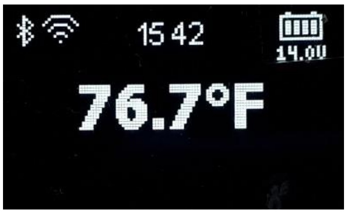
15:49 14.0V

24.9°C

Setpoint = 3.4Hz

Fahrenheit or Celsius

The displayed temperature can be shown as degrees Fahrenheit, or degrees Celsius.
Toggle the presentation by holding the **UP** button for around 1 second.

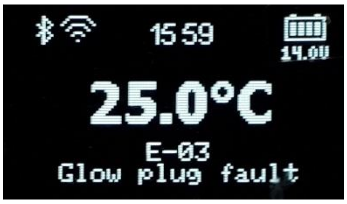


15 42 14.0V

76.7°F

Fault reporting

Should a fault be detected by the heater, an error message will be presented at the bottom of the screen, in two lines.
The 1st line will indicate the error code, eg E-03.
The 2nd line a description of the fault, eg: Glow Plug Fault:

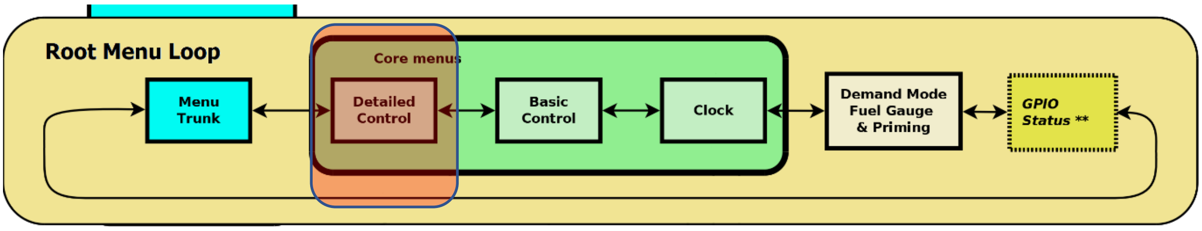


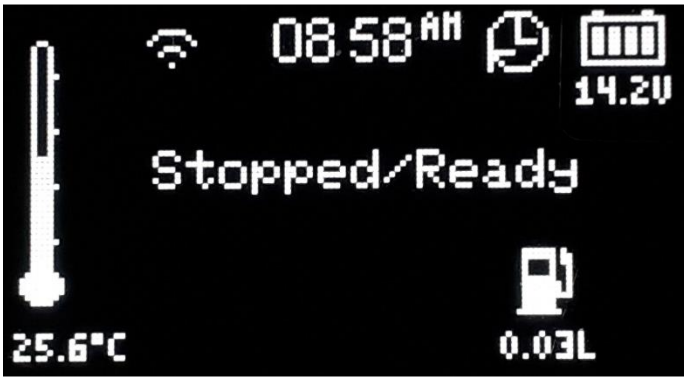
15 59 14.0V

25.0°C

E-03
Glow plug fault

Detailed Control menu



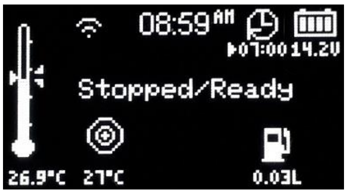


The detailed control menu allows the operation of the heater to be monitored in close detail. Here the actual pump speed, fan RPM, heat exchanger temperature and glow plug activity can be observed.

Temperature adjustment

The desired temperature, or fixed heat demand can be altered by briefly pressing **UP** / **DOWN**. The current settings can be viewed by briefly pressing **CENTRE**.

When in thermostat mode, the desired target value is placed as a small triangle to the left of the measured ambient "temperature bulb" on the left-hand side of the screen.
Small half markers may appear to the right of the bulb to illustrate the Cyclic Mode zone.
The actual temperature is shown below the bulb, the height of the “mercury” likewise changes.



Once started, the desired setting is reported beneath the "target" icon as °C,°F or Hz, depending upon thermostat or fixed demand modes.

GPIO output manipulation

You can manipulate the user mode GPIO outputs by holding the **LEFT** or **RIGHT** keys down for 1 second.
LEFT hold toggles digital output#1,
RIGHT hold toggles digital output#2.

Fuel usage

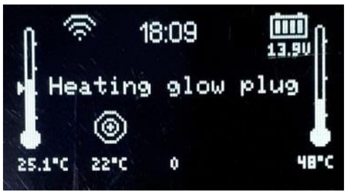
Accumulated fuel usage since the last fuel gauge reset is shown beneath the fuel bowser icon.

Heater On / Off

The heater can be turned on or off by holding down the **CENTRE** keypad button for 1 second.

Starting the heater

Hold the **CENTRE** key for a second or so.
The heater should respond, reflected in the status message:

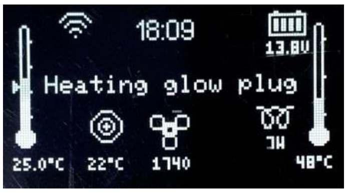


Heating glow plug

It is possible the heater will not start and a message will be brought up accordingly.
Typical causes being:

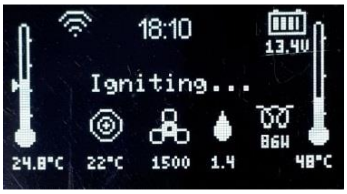
- Ambient temperature is already higher than the desired thermostat setting.
- The battery voltage is lower than the Low Voltage Cutout setting.
- The accumulated fuel usage suggests the fuel tank is nearly empty.

If started, the fan icon will appear and begin spinning, showing with the measured RPM.
The power applied to the glow plug will steadily rise.



Heating glow plug

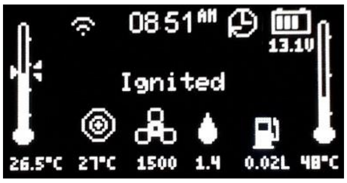
It takes about a minute for the glow plug to receive full power, at which point the pump then starts.
The Pump being active is indicated by an animated fuel droplet, along with the actual pump rate:



Igniting...

The pump starts at a slow speed and steadily rises, along with fan speed to get the heater ignited.
Ignition tends to be sensed by the heater's ECU once the temperature of the heat exchanger has risen by about 5-10°C.
The heat exchanger temperature is reported by the right hand side "thermometer bulb", the measurement value is shown below.

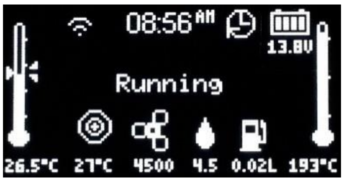
Once the body temperature rises above 65°C, the glow plug is shut down, all the while the fuel pump and fan speeds are progressively increased until at full speed.
The heater runs in this state for a few minutes getting the internals nice and hot:



Ignited

After running at full speed for a while, the heater transitions to the running state, and now obeys the desired setting being either a thermostat setting, or a fixed heat demand.

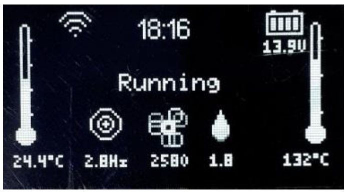
This occurs about 5 minutes after the initial start was demanded:



Thermostat or Fixed demand modes

Hold **DOWN** to toggle from Thermostat to Fixed demand modes.

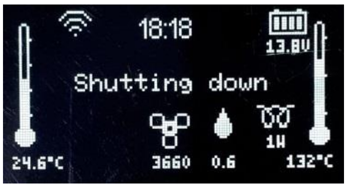
The changed operating mode is reflected in the units reported below the target icon.
°C or °F for thermostat, Hz for fixed demand.
The set temperature triangular indicator also disappears from the left-hand thermometer when in fixed demand mode:



If Cyclic Mode is enabled, markers will appear to the right of the ambient temperature thermometer to indicate the start and stop thresholds.
These markers will persist even when in Fixed mode as Cyclic Mode is always active if enabled and the heater is running.

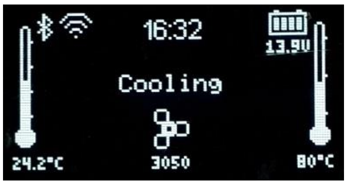
Stopping the heater

Heater shutdown is initiated by holding the **CENTRE** key for a second or so.
The status will then change to Shutting down:



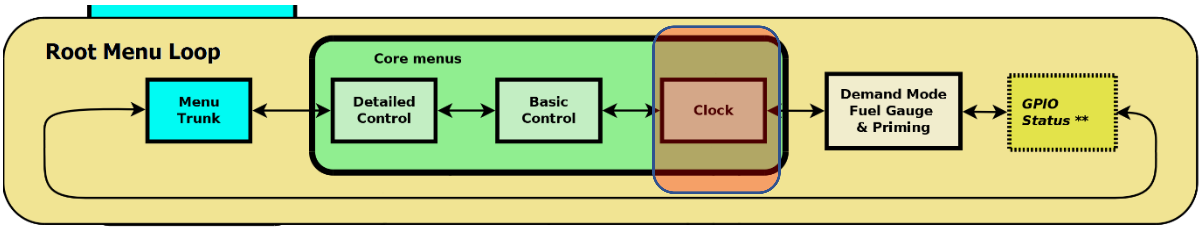
The pump speed is quickly lowered and eventually shut off.
The glow plug is re-powered to assist in drying the heat exchanger of unburnt fuel.

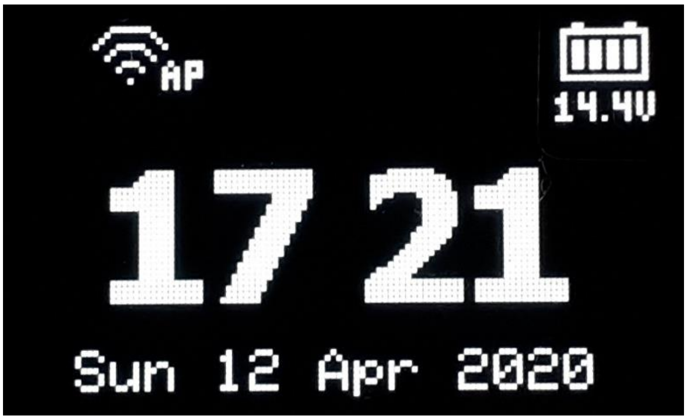
About a minute later power is removed from the glow plug and the heater transitions to cooldown mode.
Only the fan is now running:



It stays in cooldown mode until the heat exchanger is observed to have dropped below 55°C, at which point it then returns to the initial idle state.

Clock menu




AP
14.4V
17 21
Sun 12 Apr 2020

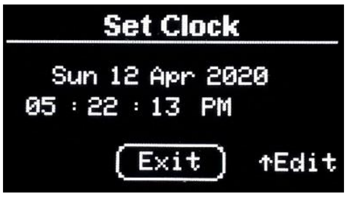
The clock menu presents the current time and date, as maintained by the Real Time Clock (RTC). This time is non-volatile, time is maintained by a lithium coin cell if the main power is off.

This menu is basically passive, just showing the time, but from here the clock can be set, or timers defined by pressing the **UP** or **DOWN** keypad buttons respectively.

You can manipulate the user mode GPIO outputs by holding the **LEFT** or **RIGHT** keys down for 1 second.

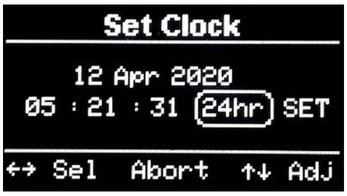
Set Time

Press the **UP** keypad button from the Clock menu.
Control branches from the *Root Menu Loop* and the Set Time menu is presented.


Set Clock
Sun 12 Apr 2020
05 : 22 : 13 PM
Exit ↑Edit

Pressing **CENTRE** will return to the *Root Menu Loop's* Clock menu.
Press **UP** again to enter clock setting mode.

The time format field will first be surrounded by a loop.


Set Clock
12 Apr 2020
05 : 21 : 31 24hr SET
↔ Sel Abort ↑↓ Adj

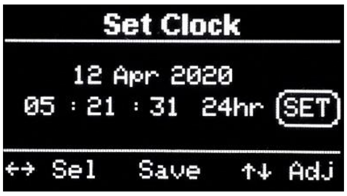
When any field other than SET is selected:

- **UP** or **DOWN** adjusts the selected value.

- **LEFT** or **RIGHT** will move to another edit field.
- **CENTRE** will abort the clock setting mode.

Applying the new Time and Date

To apply the new time/date setting, press **LEFT** or **RIGHT** until **SET** is selected.



```
      Set Clock

    12 Apr 2020
 05 : 21 : 31  24hr (SET)

↔ Sel    Save    ↑↓ Adj
```

Pressing the **CENTRE** button will then upload the new settings into the RTC chip.
Note that the time settings are maintained during power down by the CR1220 lithium battery.

Returning to parent menu (Clock)

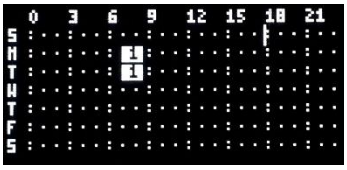
To return to the Root Loop's Clock menu, we must not be in edit mode. i.e. "Exit" will the selected at the bottom of the screen.
To abort edit mode, not altering the actual time, press the **CENTRE** key when over any edit field.
To set the time, you must press **CENTRE** when the **SET** field is selected.

Either usage of the **CENTRE** key will then select the "Exit" field.
Press the **CENTRE** keypad button once again to return the Root Menu Loop's Clock Menu.

Timers

Pressing the **DOWN** keypad button from the *Root Menu Loop's* **Clock** menu will enter the *Set Timer Menu Loop*.

The initial menu *Set Timer Menu Loop* is a graphic chart showing the coverage for the entire week of any timers that are enabled:

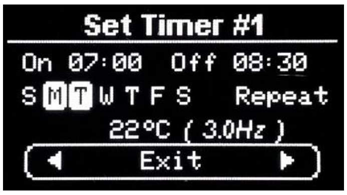


In this example, Timer 1 is enabled for Monday and Tuesday, starting around 07:00, stopping around 08:30.
The bar being solid indicates this timer repeats.
The numeric within the bar shows which timer definition applies.

There are 14 timers available, each with their own menu screen.
They can be navigated to by pressing the **LEFT** or **RIGHT** keypad buttons.

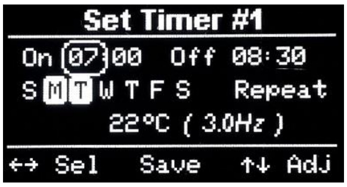
Selecting and setting fields

After navigating to a timer, a screen something like the following will appear.
The actual timer being identified in the **Set Timer** header.



Set Timer #1
On 07:00 Off 08:30
S M T W T F S Repeat
22°C (3.0Hz)
◀ Exit ▶

To return to the *Root Menu Loop's* **Clock** menu, press the **CENTRE** button.
To start editing the timer, press the **UP** button.

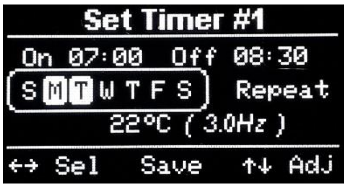


Set Timer #1
On 07:00 Off 08:30
S M T W T F S Repeat
22°C (3.0Hz)
↔ Sel Save ↑↓ Adj

- **LEFT** and **RIGHT** keys to select the field to be edited.
- **UP** or **DOWN** keys to alter the value.
  Note that holding these buttons down will enable auto repeat mode.

Enabling or Disabling the timer

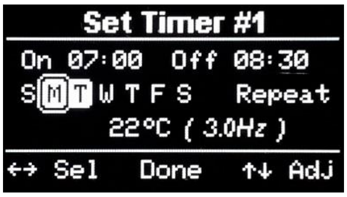
Using the **LEFT** / **RIGHT** keys, navigate to the Timer Enable field:



Set Timer #1
On 07:00 Off 08:30
S M T W T F S Repeat
22°C (3.0Hz)
↔ Sel Save ↑↓ Adj

The timer can operate in simple or advanced mode.
Simple mode allows you to just enable or disable the timer by using the **UP** / **DOWN** keys.

Advanced mode allows you to select **which specific days** the timer will be enabled for.
Advanced mode is entered by holding down the **UP** key until a list of days appears instead of just **Enabled** or **Disabled**:



```
   Set Timer #1
On 07:00  Off 08:30
S M T W T F S  Repeat
   22°C ( 3.0Hz )
←→ Sel   Done   ↑↓ Adj
```

When “per day mode” is enabled, the selection loop shrinks to only select one day. Monday is selected in the above image.

- **LEFT** or **RIGHT** select the day.
- **UP** or **DOWN** enable or disable the day.
  A day is enabled when shown in inverse text.
- **CENTRE** leaves per day edit mode.

Assuming specific days were selected, the base selection menu now indicates which days are enabled.

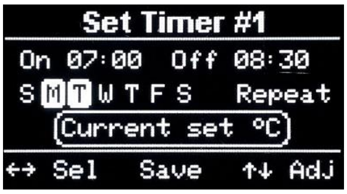
Timer Repeat or One Shot

Timers can be set to repeat every day, or self-cancel once expired.
Navigate to the bottom right edit field then press **UP** or **DOWN** to toggle the setting.

Timer Temperature

The temperature to use when the timer starts can be specified in the last field.
Note that the equivalent Fixed Hz value is also shown against this setting if the user prefers the unit running in that mode.

The new demand is applied to the heater when the timer becomes active.
It is also possible to reduce this setting below the minimum allowed (8°C). Then when the timer starts the present thermostat setting is left unchanged:



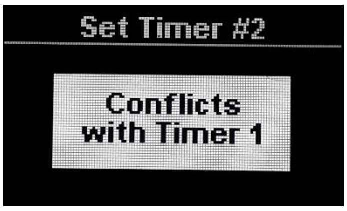
```
   Set Timer #1
On 07:00  Off 08:30
S M T W T F S  Repeat
  Current set °C
←→ Sel   Save   ↑↓ Adj
```

Saving the timer

Press the **CENTRE** key to save the timer’s settings, there is no abort function.

Before the Afterburner accepts the new timer settings, a check is made against all other timers that may have already been enabled.
It is illegal for two timers to overlap.
If the new settings do conflict with another timer, the current timer is forcibly disabled, and a

notification appears about which timer it conflicted with.

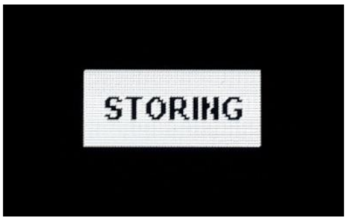


Set Timer #2

Conflicts
with Timer 1

Irrespective of the timer being accepted, the settings are always stored to non-volatile memory. The timer may however end up being disabled due to a conflict.

If no conflict occurs the following message appears:

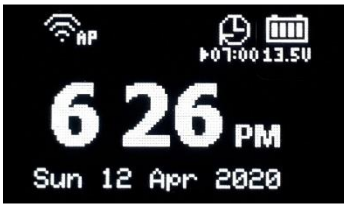


STORING

Returning to Root Menu Loop (Clock menu)

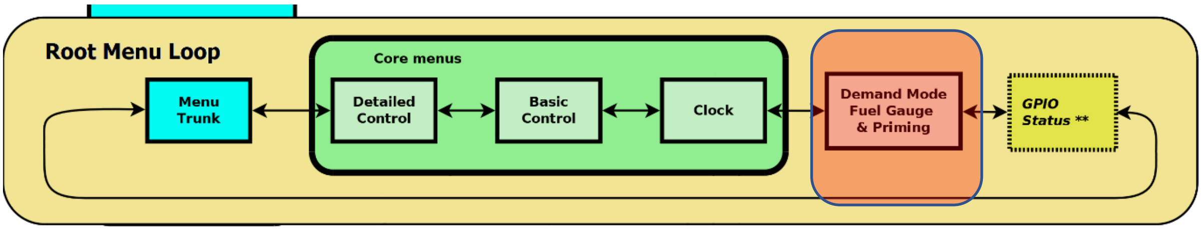
When the navigation line is selected, press the **CENTRE** key to return to the *Root Menu Loop's* **Clock** menu.
Note: If the keypad left idle for minute, the Root Menu Loop's Clock menu will be activated.

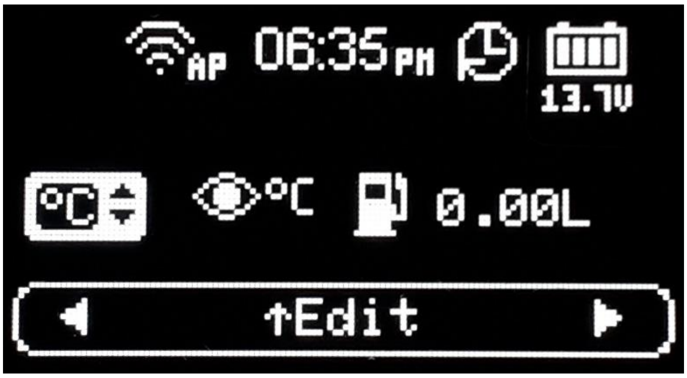
When back in the *Root Menu Loop*, at the top of the screen; if any timers have been set, a timer icon will appear along with the start time of the next scheduled timer:



If the timer is set to repeat, the icon gains an arrow at the lower left of the icon.

Mode Selection menu





The Mode Selection menu allows:

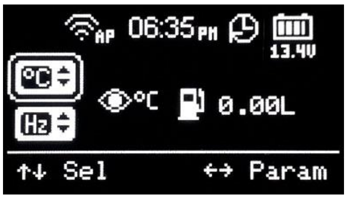
- Thermostat / Fixed Hz mode to be toggled
- Temperature to be reported as °C or °F
- Fuel pump to be primed.
- Fuel usage to be reset

Initially enter the options by pressing **UP**.

- **LEFT** / **RIGHT** selects the other options.
- **UP** / **DOWN** alter the selected option.

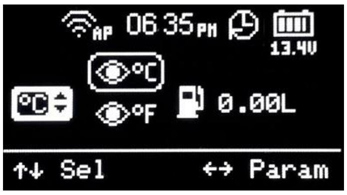
Thermostat / Fixed Hz selection

The first option allows Thermostat or Fixed Hz heater mode to be used:



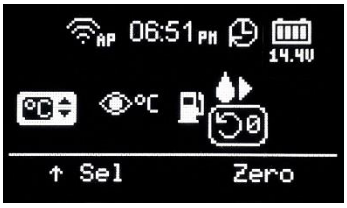
Celsius or Fahrenheit

The next option is whether temperature should be viewed as degrees Celsius or degrees Fahrenheit:

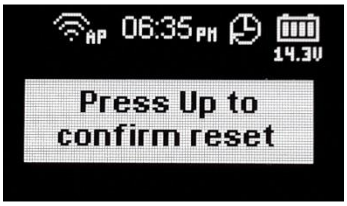


Fuel Usage Reset

The next option is whether to start the fuel pump for priming, or reset the accumulated fuel usage:



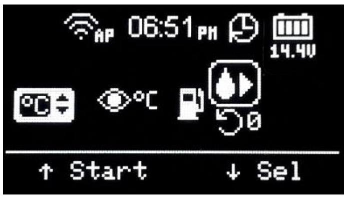
Pressing **CENTRE** will prompt for confirmation you want to reset the fuel usage.
Fuel usage reset must be performed after filling the tank if you enable the Low Fuel Shutdown feature.



Press Up to
confirm reset

Fuel Pump Priming

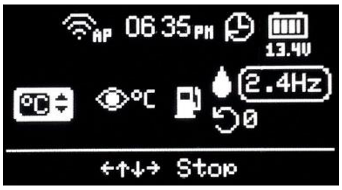
The top selection allows fuel pump priming to be performed:



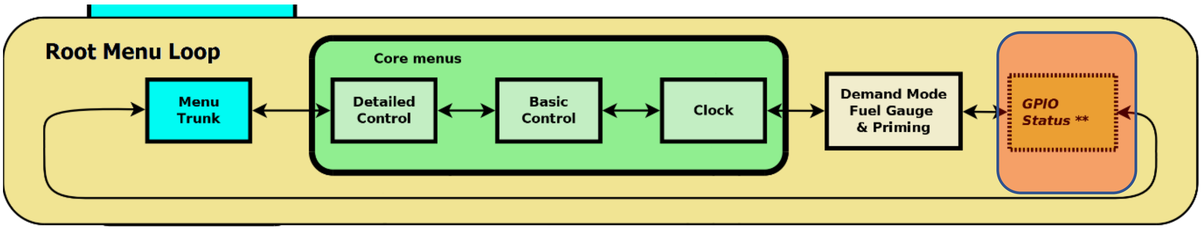
Pump priming is always disabled by default and can only be enabled if the heater is **not already running**.

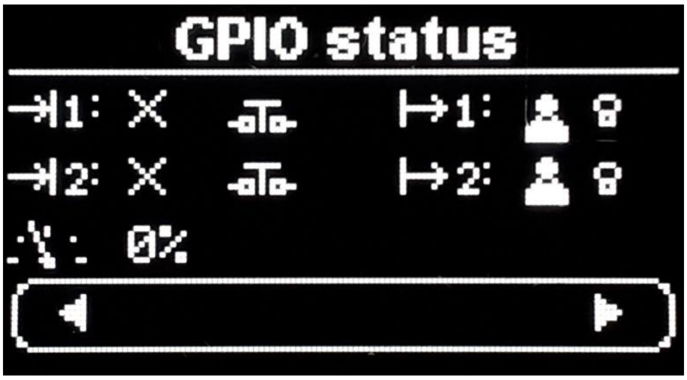
Press **UP** or **CENTRE** to start the fuel pump, the current pump speed is then reported.

Stop the fuel pump by pressing ANY key.
The fuel pump will also be automatically stopped after 1 minute, and control will be returned to the **Basic Control** menu due to the inactivity timeout.



GPIO Status menu





The GPIO status menu allows the current signal states of the GPIO port to be visualised.

This menu only appears if GPIO hardware has been installed to the Afterburner PCB.

Digital Inputs

The 2 digital inputs are represented by the incoming arrow icons to the top left of the screen.

Depending upon how the GPIO inputs have been configured, the icons presented adjacent to the inputs will be one of the following:

✕	Disabled	Changes at the input are ignored
▶	Start	A momentary closure will start the heater, a long closure will stop.
■	Stop	A momentary closure will stop the heater.
▶■	Start / Stop	Alternate momentary closures will start or stop the heater
🏃	Run	The heater will run whilst the input remains closed
δT	External Thermostat	The heater will cycle high/low power according closed/open

The switch icon shows the state of the digital input:

-o⊤o-	The input is open circuit
-▬⊤▬-	The input is closed to ground

Digital Outputs

The 2 digital outputs are represented by the outgoing arrow icons to the top right of the screen.

✕	Disabled	The output remains inactive
💡	LED Status	An attached LED will dynamically indicate the heater run state

👤	User	The output is controlled by user command
⇌	Temperature threshold	The output is active according to the defined temperature threshold
⏻	Heater On status	The output is active whenever the heater is not is the standby/stopped condition

The state of the output is reflected by the ‘bulb’ icon:

💡	The output is inactive
💡	The output is active

Analogue input

The input voltage applied to the Analogue input is presented as a percentage between 0.0V and 3.3V.

Presently the analogue value is purely for user interpretation.

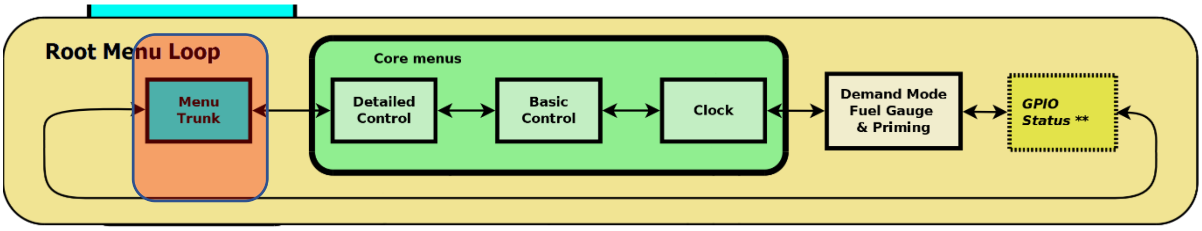
User interaction

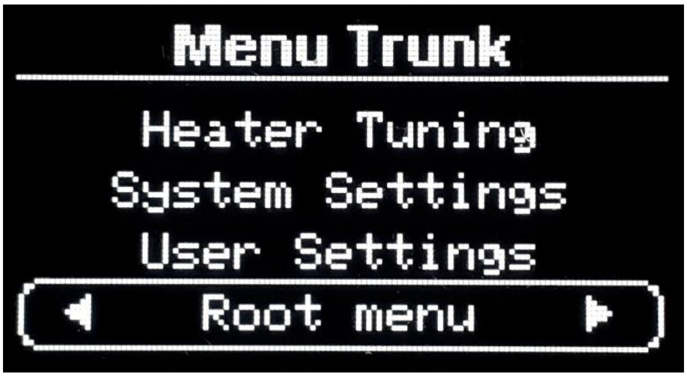
If a digital output has been defined as User Controlled, the current state can be toggled by holding down the **LEFT** or **RIGHT** keys:

The **LEFT** key toggles digital output #1

The **RIGHT** key toggles digital output #2

Menu Trunk





The menu trunk allows navigation into the other major menu loops than the *Root Menu Loop*.

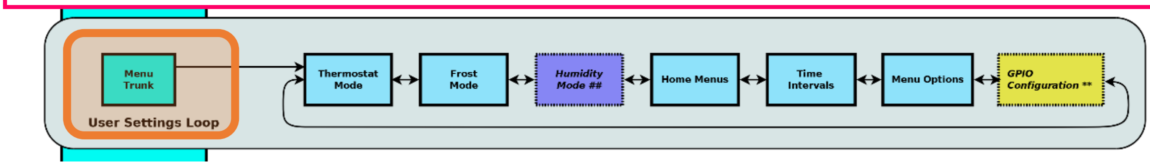
Use **UP** or **DOWN** to select a new menu loop.

Navigate into the selected menu loop using **LEFT** or **RIGHT**.

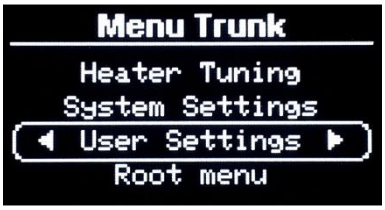
Immediately select the *Root Menu Loop* by pressing **CENTRE** when not on the bottom line.

Pressing **CENTRE** when already on the “Root menu” selection will jump to the user specified default timeout menu.
If the default is not specified, the **Basic Control** menu will be selected.

User Settings Menu loop



The User Settings menu loop can be accessed via the Menu Trunk of the Root Menu loop.

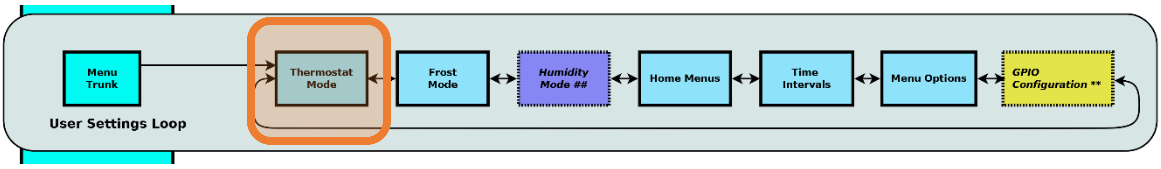


Use **UP**/**DOWN** to navigate to User Settings line, then pressing **LEFT** or **RIGHT** enters the **User Settings menu loop**.

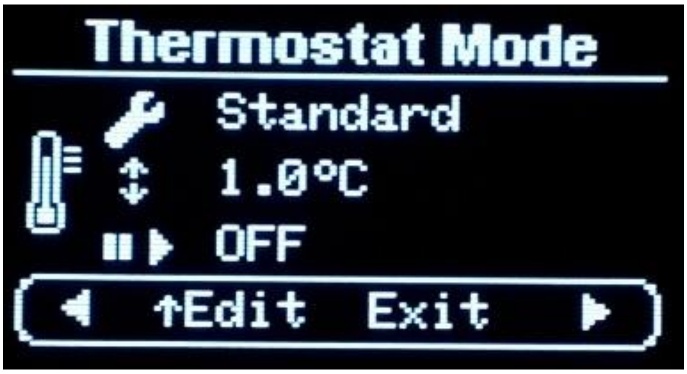
As the name implies this menu loop holds settings that are predominantly user options, not so much heater system settings.

Thermostat Mode menu (User Settings menu loop)

The Thermostat Mode menu is always the first shown when departing from the Menu Trunk into the User Settings menu loop.



Thermostat Mode menu:



Thermostat Mode

🔧 Standard

↕ 1.0°C

▮▶ OFF

◄ ↑Edit Exit ►

Press **UP** to enter edit mode and select a setting to change.

- **LEFT** / **RIGHT** adjust the selected setting.
- **UP** / **DOWN** select another setting.
- **CENTRE** to accept and save the setting.

The available options are indicated by the graphic icons:

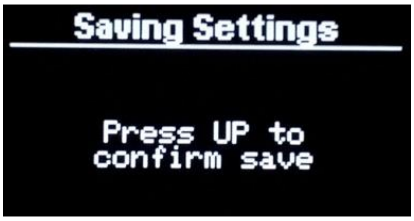
🔧 Thermostat algorithm selection *(Standard, Dead-band, Linear Hz, External)*

↕ Thermostat window size *(0.2°C – 6.3°C range)*

▮▶ Cyclic Mode thresholds *(automatic overheat shutdown/restart)*

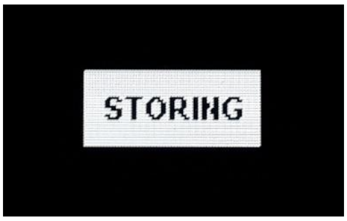
Abort by pressing **DOWN** until the navigation line is re-selected, then press the **CENTRE** key.

To set and save the settings, the **CENTRE** key must be pressed whilst one of the settings is selected. You must then confirm the changes by pressing the **UP** key.
STORING will then appear as the settings are saved to non-volatile memory:



Saving Settings

Press UP to confirm save



STORING

Thermostat Modes

The Afterburner offers 4 Inbuilt Thermostat Modes and can optionally support an external thermostat:

- Standard
- Deadband
- Linear Hz
- Stop / Start
- An external dry contact thermostat can be used *(only if GPIO capability is available)*

In addition, a special supervisor mode (Cyclic Mode) can operate atop all thermostat modes. Cyclic Mode shuts the heater down if the temperature keeps on increasing despite being at minimum power.

If ambient temperature is already above the desired temperature, attempts to start the heater will be ignored

Standard Thermostat mode

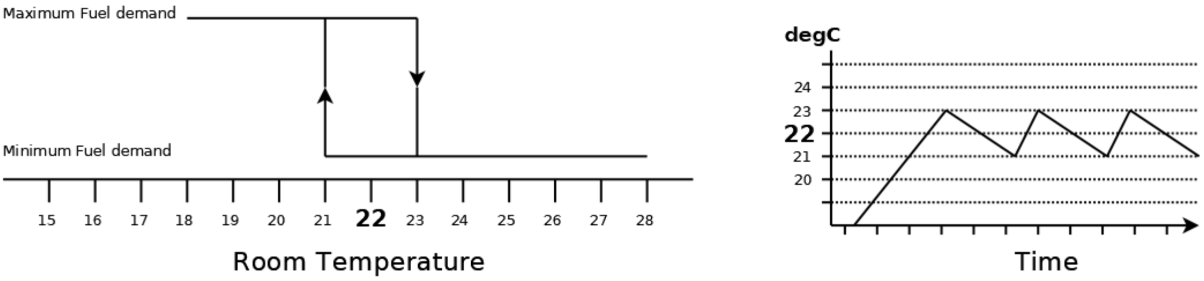
Uses the standard thermostat algorithm built into the heater ECU.

Both the current measured temperature and desired setpoint are sent to the heater, at 1°C resolution.

Starting from cold, the heater will run at the maximum defined fuel rate until the ambient temperature exceeds the setpoint by 1°C. Once this threshold is crossed, the heater then changes to run at the minimum fuel rate.
When the ambient temperature later drops 1°C below the setpoint, the heater once again switches to run at the maximum fuel rate.

i.e. a 2°C hysteresis.

Standard thermostat mode graphic, demonstrating fuel transitions about a 22°C setpoint:

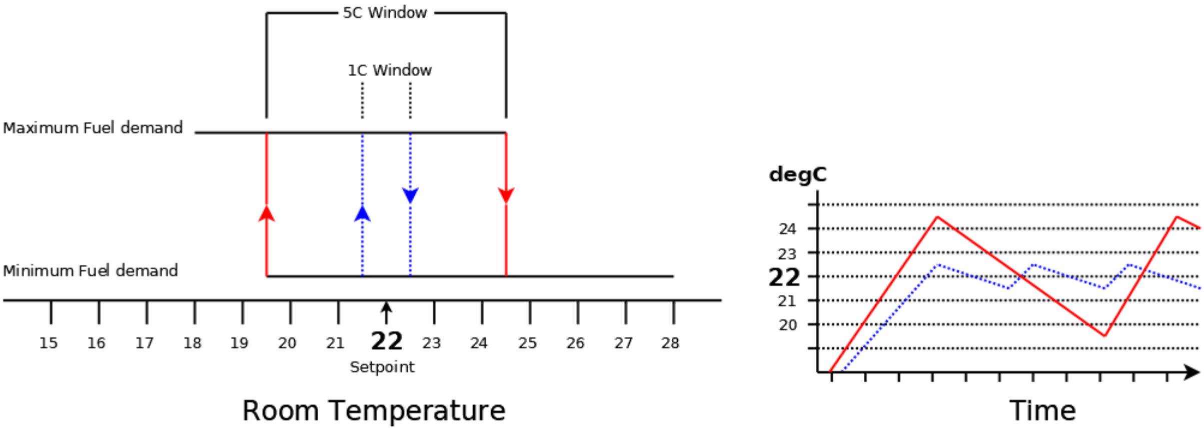


Dead-band mode - modified hysteresis

Dead-band mode still uses the heater ECU’s internal thermostat algorithm, but the Afterburner distorts the ambient temperature value that is advertised to the heater.
If the true ambient temperature lies within the user definable window about the setpoint, the setpoint temperature is sent to the heater.
If the temperature lies outside the window, the temperature sent to the heater will be at least 1°C higher or lower than the setpoint.
Thereby the dead-band, or hysteresis of the ECU’s thermostat is controlled over a narrower or wider

range.
A dead-band window of 2°C is equivalent to the standard thermostat function.

Dead-band thermostat mode graphic:



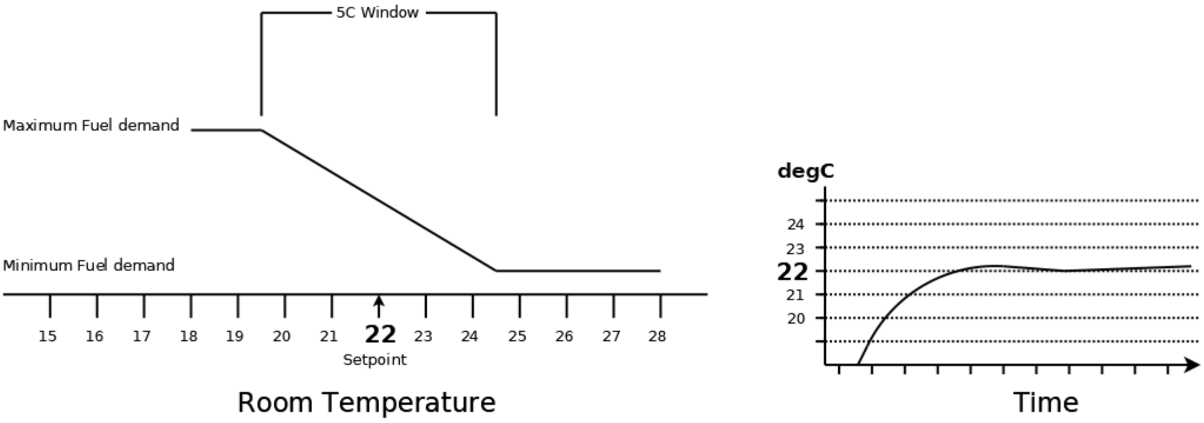
Linear Hz

Linear Hz mode is a hybrid method that removes the thermostat function from the heater.
The heater is told to run using Fixed Hz, or “heat demand” mode.

If the ambient temperature is below the bounds of the user defined window about the setpoint, maximum fuel rate is requested by the Afterburner.
If the ambient temperature is above the window’s bounds, minimum fuel rate is requested.
Within the window, the fuel rate demand is linearly adjusted between the maximum and minimum fuel rates according to the deviation from the setpoint.

Accordingly, the ambient temperature is automatically maintained within a narrow band.
As the fuel rate is gradually tapered, the initial arrival at the setpoint will be more exponential.
Note that the exact average temperature may be slightly above or below the desired setpoint, but should remain fairly steady.

Linear Hz thermostat mode graphic:



Stop / Start Thermostat Mode

Stop / Start thermostat mode operates by running the heater at full power until the desired temperature is reached, plus half the window.
When the ambient temperature reaches the desired temperature + ½ the window temperature, the heater is instructed to shut down.

eg:

Desired setting is 22°C, Window is defined as 1°C.
Heater will be shut down @ 22.5°C.

Once shut down, the Afterburner considers the heater to be suspended, meaning it will instruct the heater to start when the ambient temperature falls back below desired – ½ Window.
eg:

Desired setting is 22°C, Window is defined as 1°C.
Heater will be asked to restart @ 21.5°C.

This mode will cause the glow plug to the be used during both start and stop cycles, so consideration should be made in regards extra possible power consumption.

It usually takes 5 minutes for a heater to reach normal running condition, and a further 3 minutes or so to shut down.
The heater denies restart attempts whilst performing the shutdown procedure.

External Thermostat Mode (requires GPIO input)

A typical thermostat used for domestic heating that provides a contact closure for heat demand can be attached the Afterburner's GPIO input.
Before you can select External Thermostat mode in this menu, you must first enable the External Thermostat mode in the GPIO Configuration menu.

The heater is forced in Fixed Hz mode, minimum and maximum fuel demand are commanded according to if the thermostat's contact being open or closed respectively.

Cyclic Mode can still supervise this mode, shutting down the heater if it should get too warm.

Cyclic Mode (overheat supervisor)

When enabled, Cyclic mode operates irrespective of the thermostat mode in use.

Cyclic Mode is a supervisor mode that can be used to shut the heater down when a maximum temperature margin is exceeded.
This margin creates a threshold that is always relative to the desired setpoint.

When stopped due to exceeding this threshold, the Afterburner then considers the heater is "Suspended" as opposed to "Stopped".

When "Suspended", the Afterburner is continually monitoring the ambient temperature.
It if falls below the restart temperature threshold the heater is once again started.

e.g.:
The Cyclic stop margin is set to 2°C, the Cyclic start margin is set to 0°C.
The desired setpoint is 22°C.
If the ambient temperature rises over 24°C, the heater is requested to stop running.
When ambient falls below 22°C the heater is restarted.

The allowed range for the stop margin is +2°C to +10°C, setting to 0°C disables the function.
The allowed range for the start margin is 0°C to -20°C.

The web page temperature gauge graphically illustrates the Cyclic stop and start thresholds as red bars about the desired temperature.

Suspended Mode

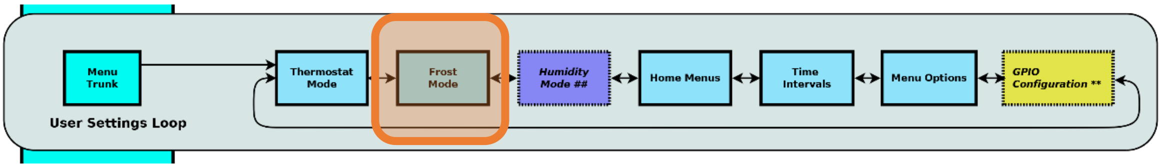
If you try to restart the heater when the Afterburner has entered suspended mode, it will instead transition directly to Standby mode, cancelling the Cyclic restart.
A further manual attempt to start the heater will then allow the heater to start.

Starting in elevated ambient temperature

If an attempt is made to start the heater when ambient temperature already exceeds the Cyclic Mode upper threshold, the heater is immediately switched to Suspended mode.

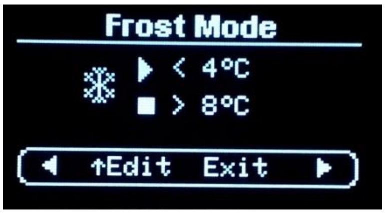
Caution
Starting and stopping the heater causes the glow plug to be cycled during both shutdown and restart phases.
Consideration should be given to the higher electrical energy load required when using a battery powered system if you expect the Cyclic Mode thresholds to be frequently encountered.

Frost Mode *(User Settings menu loop)*



Frost mode allows the heater to automatically start if the sensed thermostat temperature falls below a defined value.

Termination of Frost Mode can automatically happen after a desired temperature rise has been observed, at which point the heater is commanded to shut down again.
Alternatively a rise of 0 indicates that once started, it should then target the most recently used desired temperature.



Frost Mode

▶ < 4°C
■ > 8°C

◀ ↑Edit Exit ▶

To alter the setting, press UP from the base line.
Use LEFT or RIGHT to change the setting.
Use CENTRE to save the settings, confirming as prompted.

To disable

To disable Frost Mode, adjust the start temperature down to 0°C.
DISABLED is then presented as the start temperature.

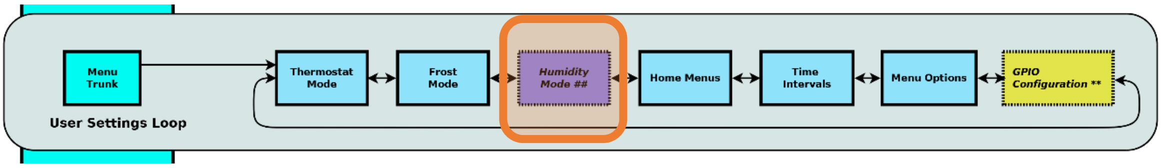
To run continuously once started

To allow the heater to operate up to the normal desired temperature once started, reduce the stop temperature to match the start temperature.
User Stop is then displayed as the stop temperature.

IMPORTANT:
When enabled, Frost Mode will always engage if the start condition is met irrespective of whether you have already turned off the heater.
Ensure the heating ducts remain free of obstructions when unattended.

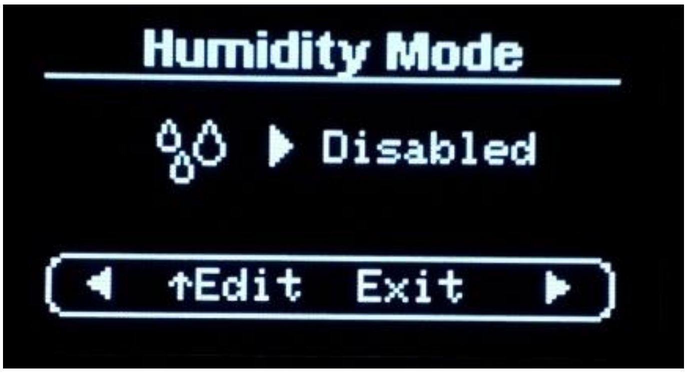
Humidity Start *(User Settings menu loop)*

– Requires BME-280 environmental sensor

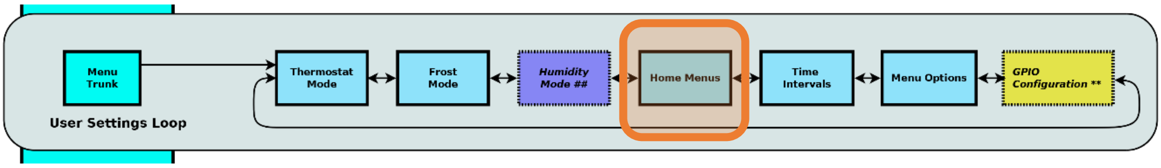


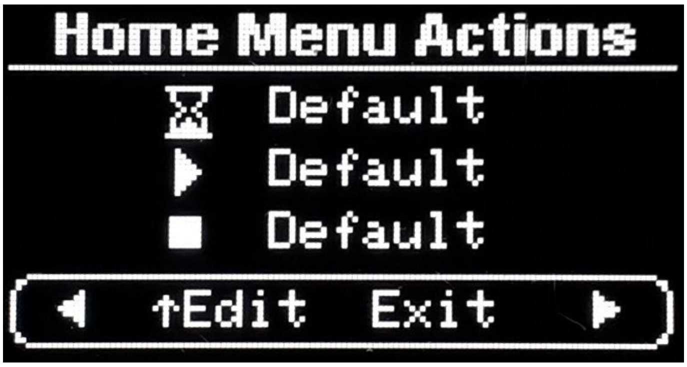
Humidity start will start the heater if the humidity rises above a defined threshold.

Presently once started, the heater will remain running until shut down by the user.



Home Menu Actions *(User Settings menu loop)*




Home Menu Actions

⌛ Default

▶ Default

■ Default

◀ ↑Edit Exit ▶

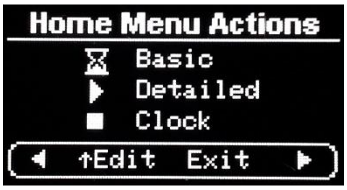
The Home Menu Actions menu allows the user to customise which default Root Menu is shown for the following events:

• Inactivity timeout	*(hourglass icon)*
• Heater started	*(start icon)*
• Heater Stopped	*(stop icon)*

The available options for each event being:

- Default menu (typically **Basic Control**, maybe **Clock** when setting timers)
- **Detailed Control** menu
- **Basic Control** menu
- **Clock** menu

The following example shows the settings for menu timeout being the **Basic Control** menu, starting the heater will bring up the Detailed status menu, and stopping the heater will bring up the Clock menu:


Home Menu Actions

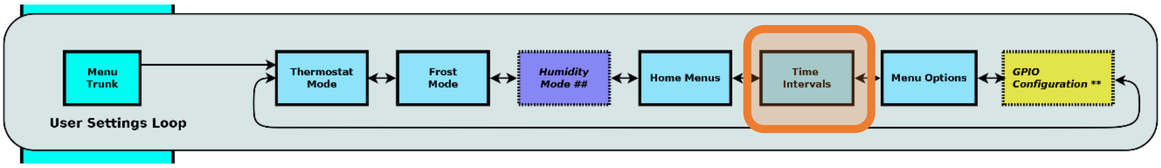
⌛ Basic

▶ Detailed

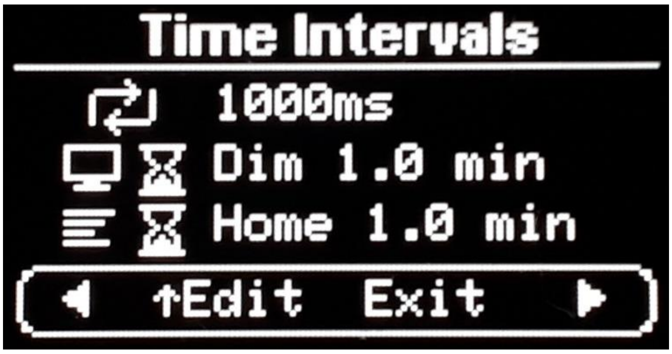
■ Clock

◀ ↑Edit Exit ▶

Time Intervals *(User Settings menu loop)*



The Time intervals menu allows the polling rate of the heater to be adjusted and sets the timeout values for default menu and display dimming or blanking.



Time Intervals

1000ms

Dim 1.0 min

Home 1.0 min

◀ ↑Edit Exit ▶

Heater Polling Rate

The looping arrows icon sets the rate that the controller sends new information to the heater and receives the current status in response.
The default rate is 1000ms (1 second) which allows co-operation with an OEM controller. Once running stand-alone you may choose to speed the update rate for more frequent status updates.

Display Timeout

The display timeout is shown as a monitor icon paired with an hourglass icon.
After the nominated time, the display can be made to:

- Stay illuminated
- Dimmed
- Blanked

To toggle from Dim or Blank modes, ensure the 2nd line is selected (by pressing **UP**/**DOWN**), then hold **LEFT** or **RIGHT** to toggle the action between Dim or Blank.

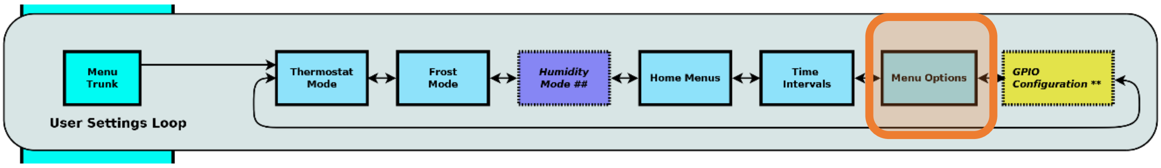
To keep the display always on, reduce the time out to 0, using presses of the **LEFT** button in Dim mode, or **RIGHT** button in Blank mode.

Home Menu Timeout

The Home Menu timeout is shown as a monitor icon paired with an hourglass icon.
After the nominated time, the menu system will be forced to the timeout menu selected in the Home Menu Actions menu *(User Settings menu loop)*

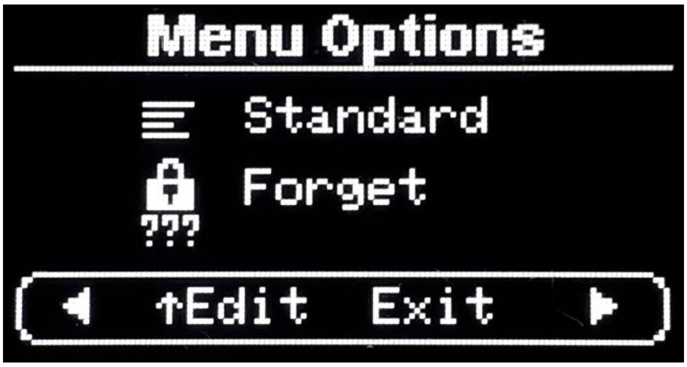
To prevent the menu timeout, simply reduce the time to 0 here.

Menu Options menu *(User Settings menu loop)*

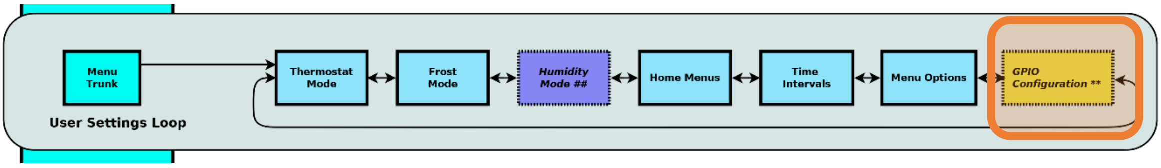


The Menu Options menu allows the menu system to be simplified to the essential operating functions.

You can also choose whether the password required for protected menus is remembered for 24 hours or needs to be re-entered every time you enter such a menu.

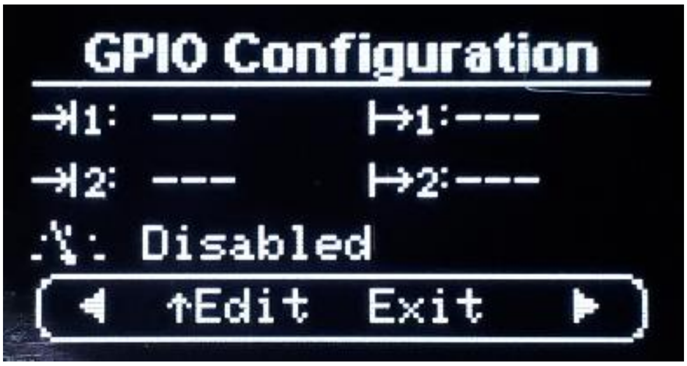


GPIO configuration *(User Settings menu loop)*



The GPIO Configuration menu is only available if GPIO capability is detected on the PCB.

By default, the functions of the GPIO pins is disabled. They must be enabled via this menu in the User Settings menu loop:



```
GPIO Configuration
⇥1: ---        ↦1:---
⇥2: ---        ↦2:---
.\: Disabled
( ◀  ↑Edit  Exit  ▶ )
```

The top 2 left hand entries show the input mode of the 2 digital inputs.
The 2 right hand entries show the output mode of the 2 digital outputs.
The 3rd line shows the analogue input mode.

The IO line to be configured is selected by using the **UP/DOWN** keys.
The mode is then changed by using the **LEFT/RIGHT** keys.

Some modes have a further option, in which case a long hold of the **LEFT** or **RIGHT** keys will allow you to switch focus to the side parameter:

External Thermostat hold time

Temperature Threshold adjustment

For the Temperature Threshold output mode, hold the **RIGHT** key to allow adjusting the threshold temperature.
Once the selection is expanded, short presses of **LEFT**/**RIGHT** will adjust the value.
Holding the **RIGHT** key down defines the output to be active when the temperature exceeds the set value. Holding the **LEFT** key down makes the output active when below the set value.

Digital Input configuration

The 2 digital inputs can be configured as:

Input #1 – Blue wire

• Disabled	
• Start – *time sensitive*	A short close (<1.5 seconds) will **start** the heater. A long close will **stop** the heater
• Run	Holding the input closed will run the heater, opening will stop the heater.
• Start / Stop	If stopped, a momentary close will start the heater, a subsequent momentary close will then stop the heater.

Input #2 – Green wire

• Disabled	
• Stop	A momentary close will stop the heater.
• External Thermostat	Closed: heater runs at high speed Open: heater runs at low speed. An optional open time can be defined to stop the heater after being held open for longer than the chosen interval. Heater is always started upon initial contact closure.
• Zero Fuel Usage	Fuel usage is zeroed if held closed > 1 second.

Digital Output configuration

The two digital outputs can be configured as:

Output #1 – Orange wire

• Disabled	
• LED Status Indicator	Standby – OFF Heater starting – “Breathe up” illumination Heater running – Steady illumination Heater starting – “Breathe down” illumination
• User controlled	Can be made active upon user demand
• Temperature Threshold	Output active when either above or below a defined temperature threshold
• Run Status	Output is active when heater is in any running condition -from start through to final cooling phases

Output #2 – Yellow wire

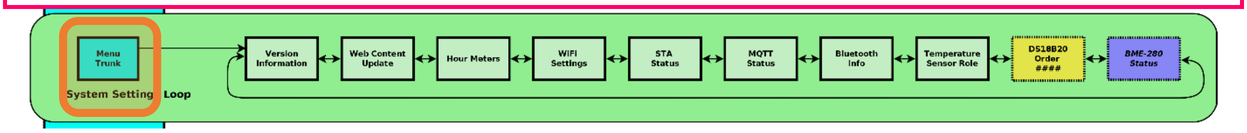
• Disabled	
• User controlled	Can be made active upon user demand
• Temperature Threshold	Output can be made active when either above or below a defined temperature
• Run Status	Output is active when heater is in any running condition -from start through to final cooling phases

Analogue input configuration

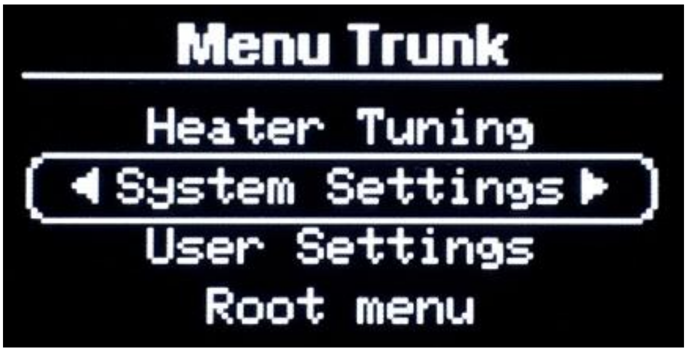
The single analogue input can be configured as:

• Disabled	
• Enabled	Input voltage applied to white wire is digitised and presented as a percentage of the usable range (0 – 3.3V)

System Settings menu loop



The System Settings menu loop can be accessed via the Menu Trunk of the Root Menu loop.



Menu Trunk

Heater Tuning

◀ System Settings ▶

User Settings

Root menu

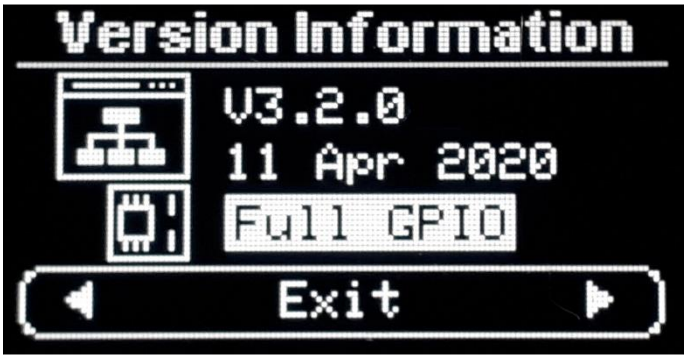
Use **UP**/**DOWN** to navigate to System Settings line, then pressing **LEFT** or **RIGHT** enters the **System Settings** menu loop.

Version Information *(System Settings menu loop)*

The version information menu allows the current installed software version to be inspected and potentially automatically updated.

From this menu you can:

- Authorise automatic web updating of the firmware.
- Restore factory default values to the stored non-volatile parameters.



Version Information
V3.2.0
11 Apr 2020
Full GPIO
Exit

The first two lines show the current installed version, and its release date.

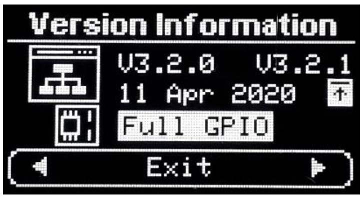
The third line shows the GPIO capabilities of the installed PCB.

- Full GPIO
- No Analog
- No GPIO

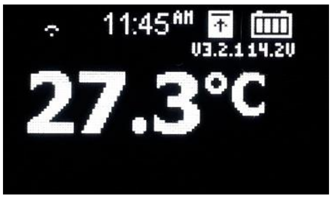
IMPORTANT:
PCB versioning is automatically determined the very first time a board is programmed from scratch, or immediately after performing a factory default.

To ensure the PCB version is correctly detected following a factory default, it is strongly recommended to temporarily remove the 7-pin expansion port cable if it is fitted.

Provided STA mode WiFi is connected to an Internet enabled network, when the **Version Information** menu is opened a check is made for a possible firmware update being available from the Afterburner web server; http://afterburner.www.mrjones.id.au .
If there is a new update available, the new version is flashed up at the top right of the **Version Information** screen, it will also flash up upon the home menu screens:



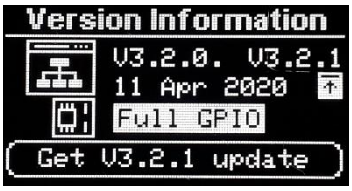
Version Information
V3.2.0 V3.2.1
11 Apr 2020
Full GPIO
Exit



11:45 AM
V3.2.1 14.2V
27.3°C

Firmware Web Update

To download and install an available firmware update from the Afterburner website, press **UP**. The bottom status line will then indicate the update can be performed by pressing the **CENTRE** button:



Version Information

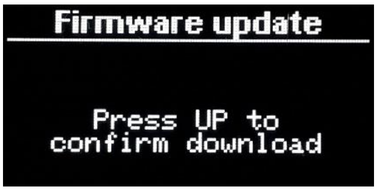
V3.2.0. V3.2.1

11 Apr 2020

Full GPIO

Get V3.2.1 update

If the **CENTRE** button is pressed, you then need to press **UP** again to confirm the download.



Firmware update

Press UP to confirm download

Once started, the download of new firmware will immediately take place, progress being shown on the OLED.

Upon the download completion, the new firmware's checksum is verified and the system reboots into the new version.

Should the download fail, the current version will be maintained.

Factory default

Pressing UP twice from the base Version menu begins the ability to perform a factory default upon all stored, non-volatile, parameters.

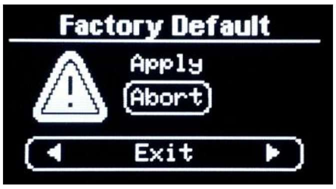
CAUTION:

Factory defaulting will reset the heater's fuel mixture tuning and any other customisations that may have been performed.

Most importantly it restores the default passwords for WiFi Access Point mode and firmware updates via a browser, and indeed the SSID of the Afterburner's Access Point:

- **SSID:** "Afterburner"
- **SSID password:** "thereisnospoon"
- **Browser update password:** "BurnBabyBurn"

After 2 **UP** presses:



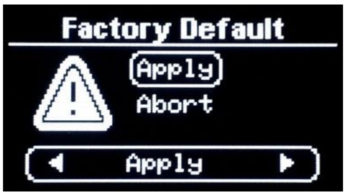
Factory Default

Apply

Abort

Exit

After 3rd **UP** press:



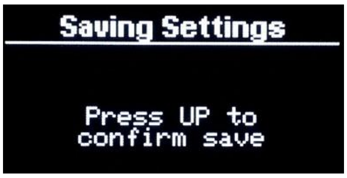
Factory Default

Apply

Abort

Apply

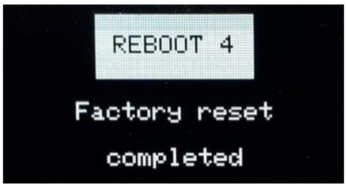
Pressing the **CENTRE** key after the 3rd **UP** press then presents a confirmation screen:



Saving Settings

Press UP to confirm save

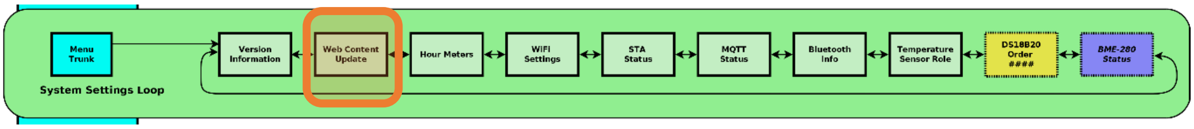
If **UP** is pressed, the factory default values are installed and the system is rebooted:



REBOOT 4

Factory reset completed

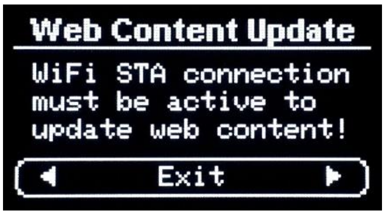
Web Content Update *(System Settings menu loop)*



Providing the Afterburner has an active STA WiFi connection established *(which provides access to the Internet)* the Afterburner can contact the Afterburner website http://afterburner.mrjones.id.au and retrieve the latest available web content and automatically save it to SPIFFS.

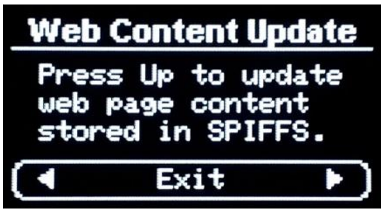
This is in lieu of using the Afterburner's inbuilt **/update** web page to upload new content.
Using /update can be problematic for some when using a mobile device to extract index.html.gz from index.html.zip, that is then required to be uploaded to SPIFFS.

If the STA connection is not available, the following message will be presented:

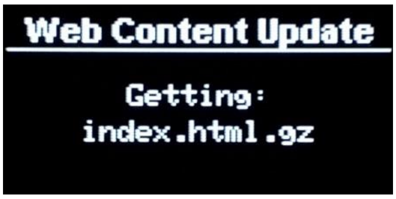

Web Content Update

WiFi STA connection must be active to update web content!

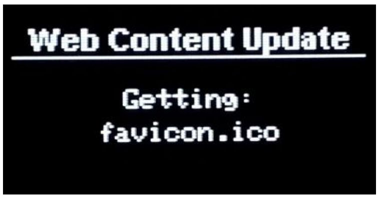
Otherwise the following sequence will occur when you press UP


Web Content Update

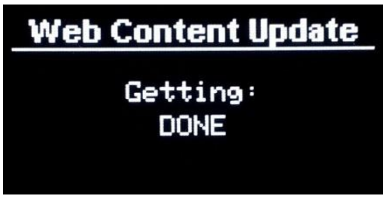
Press Up to update web page content stored in SPIFFS.


Web Content Update

Getting:
index.html.gz

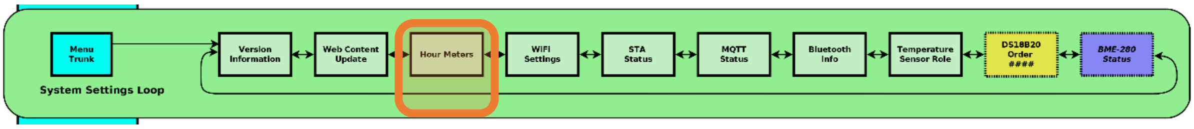

Web Content Update

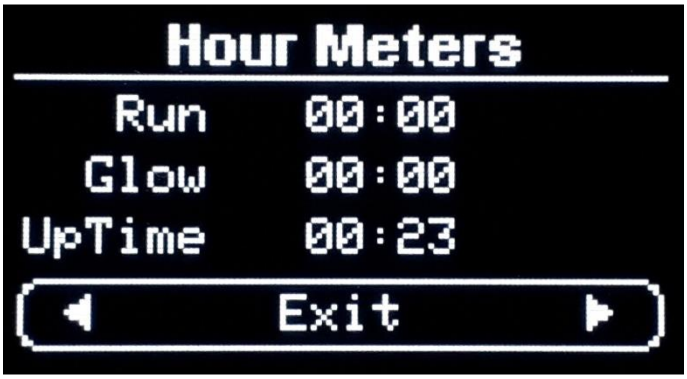
Getting:
favicon.ico


Web Content Update

Getting:
DONE

Hour Meters (System Settings menu loop)

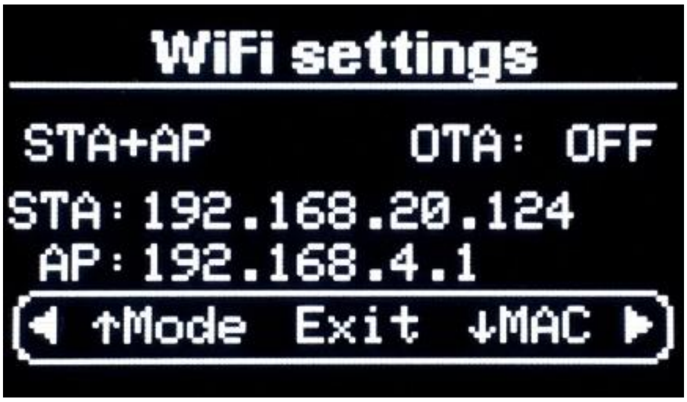




```
Hour Meters
Run       00:00
Glow      00:00
UpTime    00:23
◀      Exit      ▶
```

The Hour Meters menu gives an insight into heater usage and how long the Afterburner has been running since the last reboot.

Wi-Fi Settings *(System Settings menu loop)*



```
WiFi settings
STA+AP        OTA: OFF
STA: 192.168.20.124
 AP: 192.168.4.1
```

The Wi-Fi setting menu presents the current operating state of the Wi-Fi Interface and the associated web server interface.

IP addresses

The Wi-Fi interface host's its own AP (Access Point).
It can also connect to another network via that network's Access Point. This known as Station mode (STA).

ie **AP only**, **STA+AP** or **STA only** modes.

If Wi-Fi is enabled, the IP address for the inbuilt Access Point is shown, always 192.168.4.1.
If also connected to another network, the IP address for that network is shown against **STA**.

Web Server

Two possible web pages can be presented when a user browses port 80 (default HTTP port):

- Heater control page
- Configuration Portal to connect to an existing Wi-Fi network.

Therefore, there are 6 possible web server modes:

1. **CFG AP only:** STA configuration portal. Only available on 192.168.4.1.
2. **AP only**: Heater control. Only available on 192.168.4.1.
3. **CFG STA+AP:** STA configuration portal. Available on both IP addresses.
4. **STA+AP:** Heater control. Available on both IP addresses.
5. **CFG STA only:** STA configuration portal. Only available on STA IP address.
6. **STA only:** Heater control. Only available on STA IP address.

IMPORTANT:
If it's not possible to establish an STA connection, the Afterburner will fail back to AP only mode.

Selecting Wi-Fi operation mode

To change the Wi-Fi operation mode, press the **UP** key.
Then use the **LEFT** / **RIGHT** keys to select which mode to use in future.

To disable Wi-Fi entirely, keep pressing left until **DISABLED** appears.

> **NOTE:**
> You cannot select an **STA** mode if no credentials are stored to connect to another network.
>
> Selecting an **AP only** mode will **erase any stored credentials** for another network.
> (i.e. STA capability will be lost).

If you wish to connect to another network, or change existing credentials, you must select a Configuration Portal enabled mode; **CFG AP only, CFG STA+AP** *or* **CFG STA only**

To switch to the selected Wi-Fi mode, press the **CENTRE** key.
The new Wi-Fi setting will be saved to non-volatile memory and the system will reboot after approximately 5 seconds.

Enabling / Disabling "Over The Air" (OTA) firmware updates.

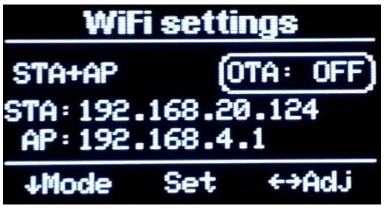
The standard method of uploading new firmware into the ESP32 uses the processor's debug serial port.
For convenience, firmware can also be updated from the Arduino environment via the Wi-Fi Interface if the OTA feature is enabled. *Note that serial upload is generally faster, but OTA does not require a physical connection to the Afterburner from the development PC.*

By default, OTA mode is disabled.
In the interests of security it is preferable to only enable OTA mode when new firmware is to be uploaded via Arduino.

Always leaving OTA enabled runs the risk of a 3rd party uploading unwanted firmware rendering the Afterburner unserviceable. *Firmware can however always be installed using the serial port option.*

As a reminder, "OTA" is reported against the Wi-Fi icon at the top of the screen when enabled.

To change the OTA setting, press **UP** from the Wi-Fi Mode selection mode:



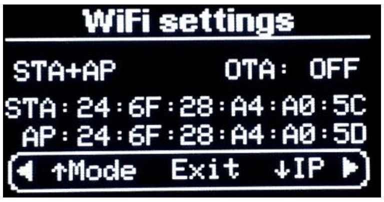
WiFi settings
STA+AP OTA: OFF
STA: 192.168.20.124
AP: 192.168.4.1

Press:

- **RIGHT** to enable OTA
- **LEFT** to disable OTA
- **CENTRE** to save the OTA mode to non-volatile memory, the system will reboot in 5 seconds.
- **DOWN** to return to the Wi-Fi Mode selection mode.

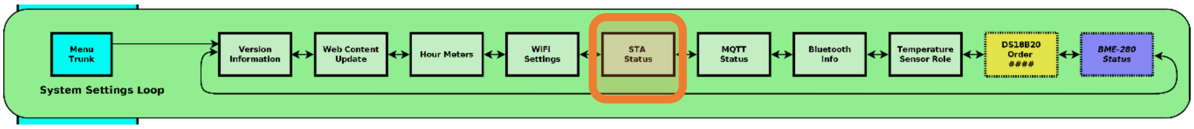
MAC address discovery

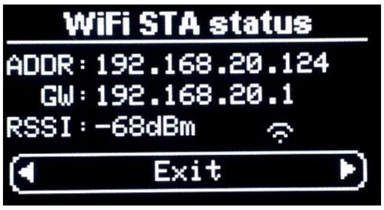
In some circumstances it is useful to know the MAC address of the TCP/IP interfaces.
Especially useful if you wish to assign a fixed IP address on the STA network for the Afterburner controller.



```
     WiFi settings
STA+AP        OTA: OFF
STA:24:6F:28:A4:A0:5C
 AP:24:6F:28:A4:A0:5D
```

STA Status *(System Settings menu loop)*



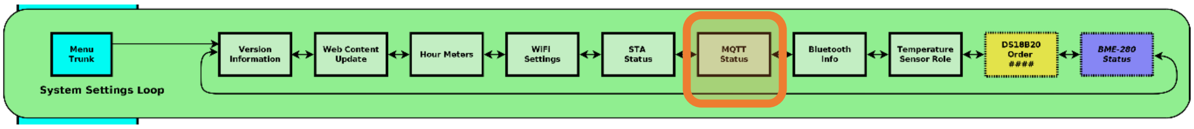


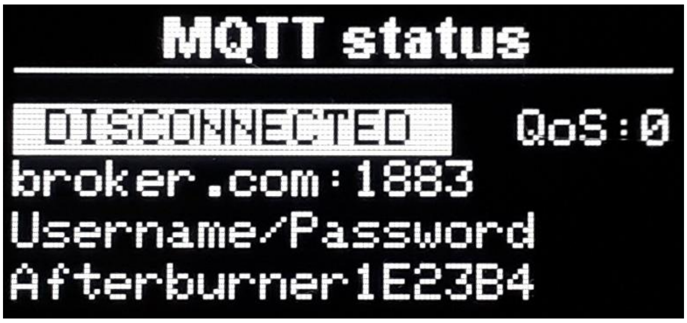
```
   WiFi STA status
ADDR:192.168.20.124
  GW:192.168.20.1
RSSI:-68dBm
◀       Exit        ▶
```

This menu provides information about the STA connection:

- IP address of Afterburner
- IP address of STA gateway
- Received Signal Strength Indicator (RSSI) of STA connection

MQTT Status *(System Settings menu loop)*





```
MQTT status
DISCONNECTED   QoS:0
broker.com:1883
Username/Password
Afterburner1E23B4
```

This menu provides status information about the MQTT connection to a broker.

Configuration of the MQTT connection can be made via the Debug serial connection or Telnet.

The latest web page content also allows configuration of MQTT, provided a direct web client connection to the Afterburner is used.

Connection Status

The 1st line shows the current connection state of the Afterburner to the MQTT broker.
Note that an active STA connection is required to allow MQTT connectivity.

Quality of Service

The QoS of the MQTT connection is shown here. It is recommended to leave this as 0.

Broker IP address

The IP address of the broker and the port for standard MQTT protocol connections is shown on the 2nd line.
If this line exceeds 20 characters, the text is automatically scrolled to avoid wrapping and overlaying the next display line.

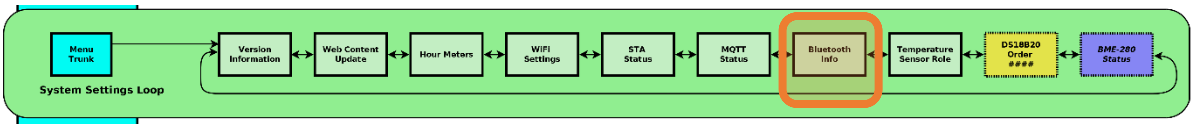
Broker Username/Password

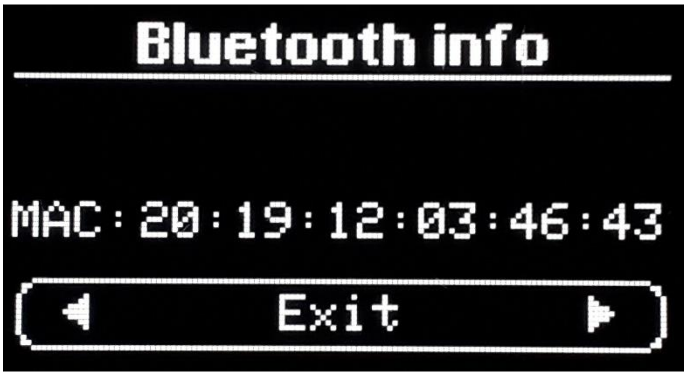
The optional username & password required for the broker is shown on the 3rd line.
If no username or password is required, they must be left blank.
If this line exceeds 20 characters, the text is automatically scrolled to avoid wrapping and overlaying the next display line.

MQTT topic prefix

The topic prefix (last line) is created by appending the last 3 octants of the STA MAC address to the string “Afterburner”. This ensures any data published by you Afterburner is genuinely unique, especially important if using a public MQTT broker.
Any MQTT client you configure must ensure the exact same prefix is used, especially maintaining Case Sensitivity.

Bluetooth information (System Settings menu loop)

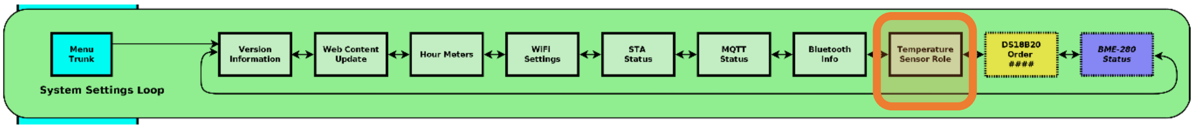


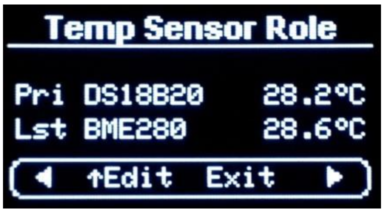

Bluetooth info

MAC:20:19:12:03:46:43

Nothing more than providing the MAC address of the HC-05 Bluetooth module!

Temperature Sensor Role *(System Settings menu loop)*





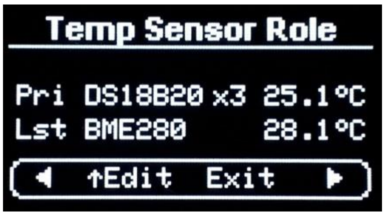
```
Temp Sensor Role
Pri DS18B20      28.2°C
Lst BME280       28.6°C
◀  ↑Edit  Exit  ▶
```

This password protected menu (code = 1688) allows you select which temperature sensor type is the sensor used for the thermostat function.

It is generally recommended to use the external DS18B20 sensor for the thermostat as it will be less affected by heat generated within the plastic case of the Afterburner.

Notice that the term “sensor type” was used earlier.
Up to 3 DS18B20 sensors can be paralleled on the same “one-wire” interface (which refers to the data signal, ignoring the extra 2 wires required to power and run the sensor).
If more than one DS18B20 sensor exists, the available sensors can be prioritised in a related menu, the DS18B20 Sensor Role menu.

If more than one DS18B20 exists, the main Temp Sensor Role display subtly changes as below, now showing how many DS18B20 sensors are present.

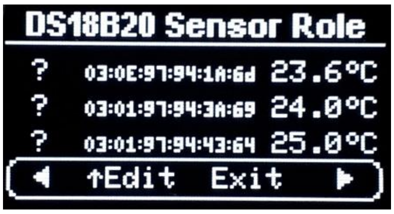


```
Temp Sensor Role
Pri DS18B20 x3   25.1°C
Lst BME280       28.1°C
◀  ↑Edit  Exit  ▶
```

IMPORTANT:
If you change the number of DS18B20 sensors to the system, make sure you reboot the Afterburner so the one-wire bus is fully re-scanned for all available sensors.

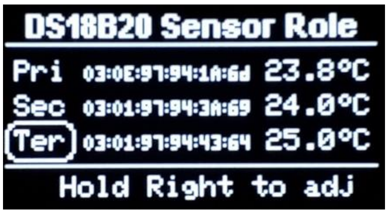
To see the detected DS18B20 sensors, press **UP** when the DS18B20 line of the Sensor Role menu is selected.

If a sensor has never been introduced to the Afterburner before, it will have no priority assigned against its serial number:



```
DS18B20 Sensor Role
?   03:0E:97:94:1A:6d 23.6°C
?   03:01:97:94:3A:69 24.0°C
?   03:01:97:94:43:64 25.0°C
◀  ↑Edit  Exit  ▶
```

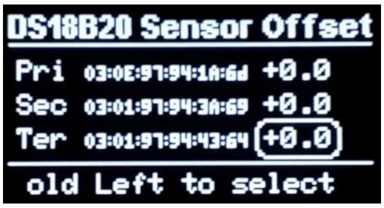
Priority assignment →



```
DS18B20 Sensor Role
Pri 03:0E:97:94:1A:6d 23.8°C
Sec 03:01:97:94:3A:69 24.0°C
Ter 03:01:97:94:43:64 25.0°C
  Hold Right to adJ
```

Each sensor can be assigned the priority in which they will be used, remembering the primary sensor here will become the thermostat sensor when the parent Sensor Role menu also selects DS18B20 as

the primary sensor.
Select a sensor using the **UP**/**DOWN** keys, then change the priority using the **LEFT**/**RIGHT** keys.



```
DS18B20 Sensor Offset
 Pri 03:0E:97:94:1A:6d +0.0
 Sec 03:01:97:94:3A:69 +0.0
 Ter 03:01:97:94:43:64 +0.0
 old Left to select
```

An offset can be applied to each sensor by holding down the **RIGHT** key.
Switch back to sensor priority by holding down the **LEFT** key.
Press **CENTRE** when not on the base line to save the priority and offset settings.
Return to the parent Sensor Role menu by pressing the **CENTRE** key when on the base line.

Identifying multiple DS18B20 sensors

If I have more than 1 DS18B20 sensor all hooked onto the same wire, how does the Afterburner differentiate between them?

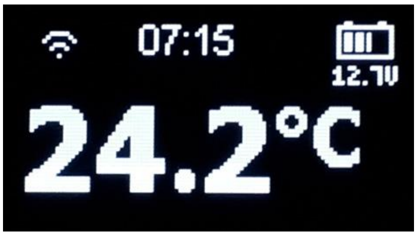
The answer is as shown in the DS18B20 Sensor Role menu. Each sensor holds a unique serial number.
Upon reboot, the Afterburner scans the one wire bus for DS18B20 sensors, learning their unique serial numbers in the process. These serial numbers are then matched against the previously configured priorities.
If you physically change a sensor, you must repeat the priority assignment procedure.

How do I work out which sensor is which?

It's as simple as grabbing hold of one sensor and observing the temperature reading rise against a sensor in the DS18B20 Sensor Role menu.
Note the serial number and location of the sensor, use old school pen and paper!

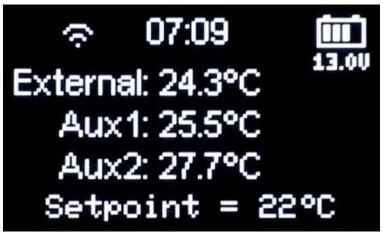
Once each sensor is identified you can then assign the order you would prefer them to appear in. When saved, the serial numbers are then stored against the chosen priority in the Afterburner's configuration memory. Next time it boots, and finds the same sensors, it will recreate the list order as previously defined.

The ultimate primary (thermostat) sensor will always be shown as the current temperature.
The extra sensors are normally not visible on the standard control menus.
You can view them when in the Basic control menu:
Press the **CENTRE** key, the current set point will be shown, along with the extra sensors.
The next sensor in the priority chain will be named "External", then "Aux1" & "Aux2".
The sensor readings are only shown if they are physically present.



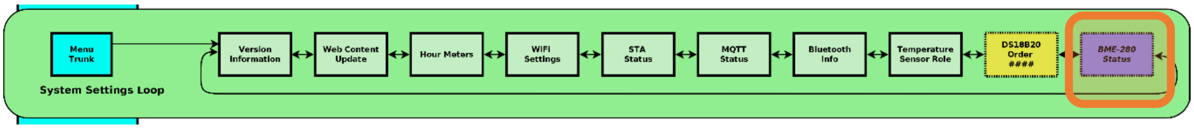
```
07:15
12.7V
24.2°C
```

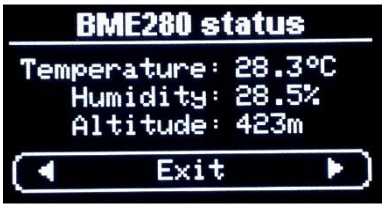
CENTRE press →



```
07:09
13.0V
External: 24.3°C
Aux1: 25.5°C
Aux2: 27.7°C
Setpoint = 22°C
```

BME-280 Status *(System Settings menu loop)*





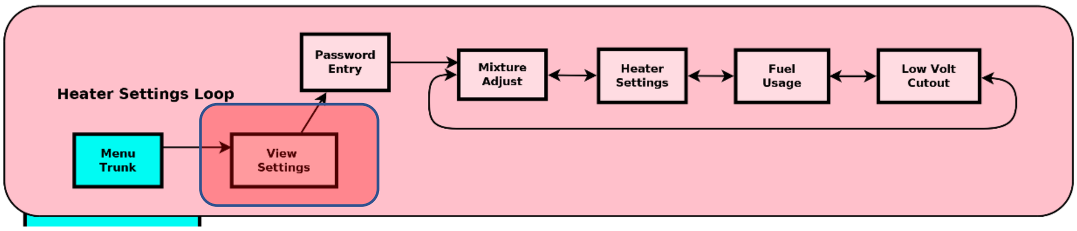
BME280 status

Temperature: 28.3°C
Humidity: 28.5%
Altitude: 423m

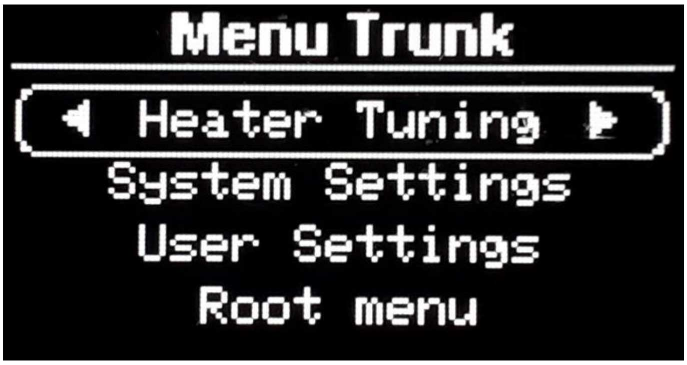
This menu provides readings from the optional BME-280 environmental sensor.

If the BME-280 sensor is not installed or fitted this menu will not be presented.

Heater Tuning menu loop



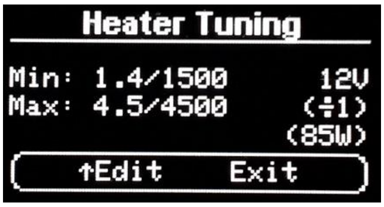
The Heater Tuning menu loop can be accessed via the Menu Trunk of the Root Menu loop.



```
Menu Trunk
◄ Heater Tuning ►
System Settings
User Settings
Root menu
```

Use **UP**/**DOWN** to navigate to Heater Tuning line, then pressing **LEFT** or **RIGHT** enters the **Heater Tuning** menu loop.

The initial Heater Tuning menu simply shows the current fuel mixture and heater settings:

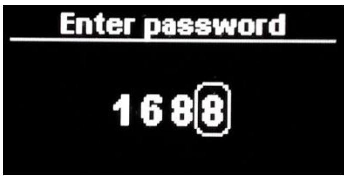


```
Heater Tuning
Min: 1.4/1500      12V
Max: 4.5/4500     (÷1)
                 (85W)
  ↑Edit      Exit
```

The currently active heater tuning settings are shown on the display.
To de-mystify SN-x and PF-x, their actual effect is toggled into view on a regular basis.

To edit the heater settings, you need to enter a password to gain access.
The password is "**1688**".

Press **UP**, a password entry menu is then presented:



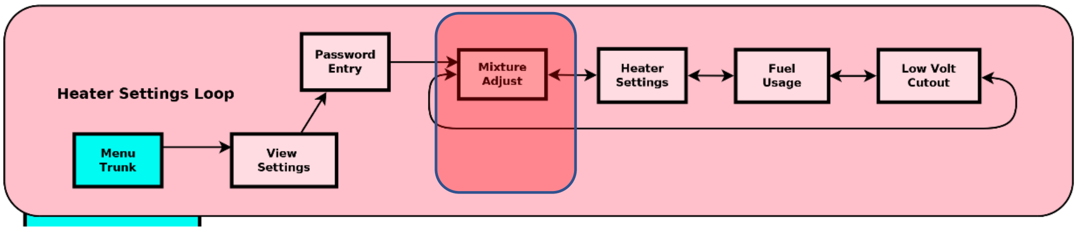
```
Enter password
1688
```

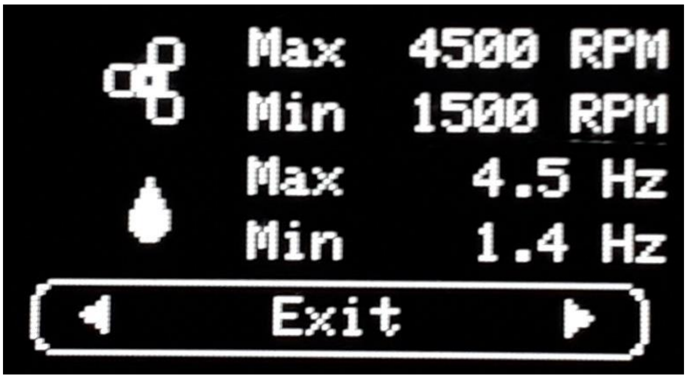
- **UP** / **DOWN** adjust each digit's value.
- **LEFT** / **RIGHT** select the next digit.

Only when "1688" is displayed, press the **CENTRE** key.
If any other value is shown, control returns to the Heater Tuning menu.

When the password is accepted, we leave the Root Menu Loop and gain access to the Heater Settings Loop.

Fuel Mixture edit menu *(Heater Settings Menu Loop)*



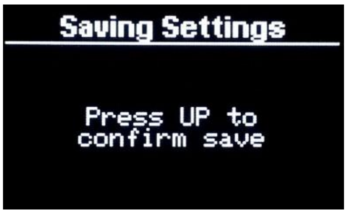

Max 4500 RPM
Min 1500 RPM
Max 4.5 Hz
Min 1.4 Hz
Exit

Press **UP** to enable edit mode:

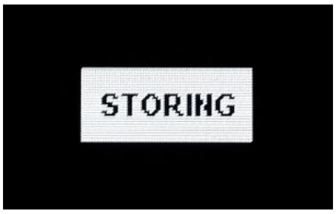
- **LEFT** / **RIGHT** adjust each setting.
- **UP** / **DOWN** select another setting.

Abort by pressing **DOWN** until the navigation line is re-selected, then move away using **LEFT** or **RIGHT**.

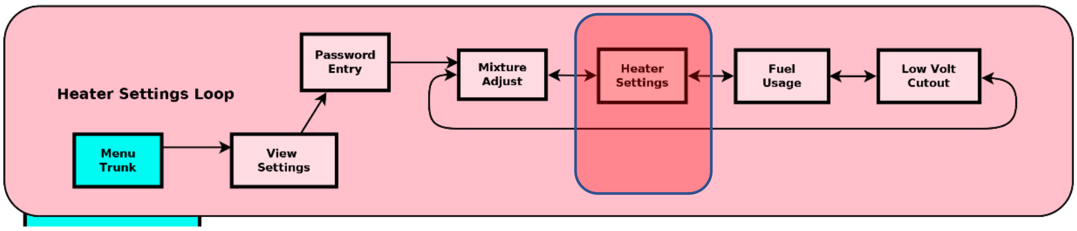
To set, save and apply the settings, the **CENTRE** key must be pressed whilst one of the adjustable settings is selected.
You must then confirm the changes by pressing the **UP** key.
STORING will then appear as the settings are saved to non-volatile memory:


Saving Settings

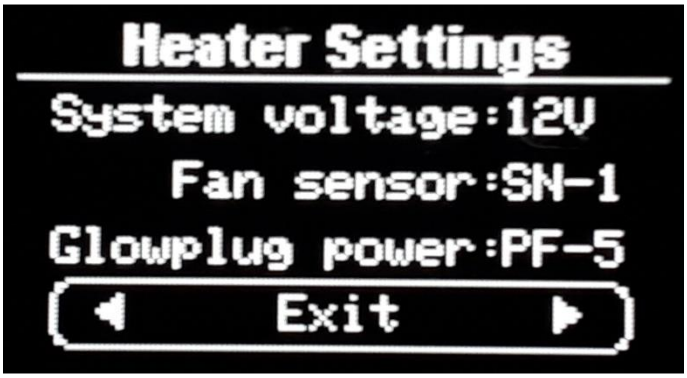
Press UP to confirm save


STORING

Heater Settings edit menu *(Heater Settings Menu Loop)*



When at the navigation line of the Fuel Mixture menu, press **LEFT** or **RIGHT** to access the Heater Settings menu:



Heater Settings

System voltage:12V

Fan sensor:SN-1

Glowplug power:PF-5

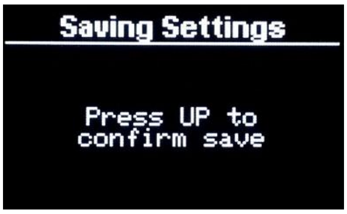
◀ Exit ▶

Press **UP** to enter edit mode.

- **LEFT** / **RIGHT** adjust the selected setting.
- **UP** / **DOWN** select another setting.

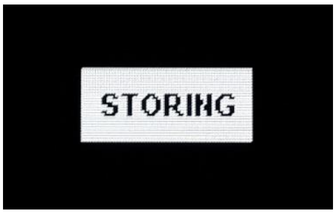
Abort by pressing **DOWN** until the navigation line is selected then move away using **LEFT** or **RIGHT**.

To set, save and apply the settings, the **CENTRE** key must be pressed whilst one of the adjustable settings is selected.
You must then confirm the changes by pressing the **UP** key.
STORING will then appear as the settings are saved to non-volatile memory:



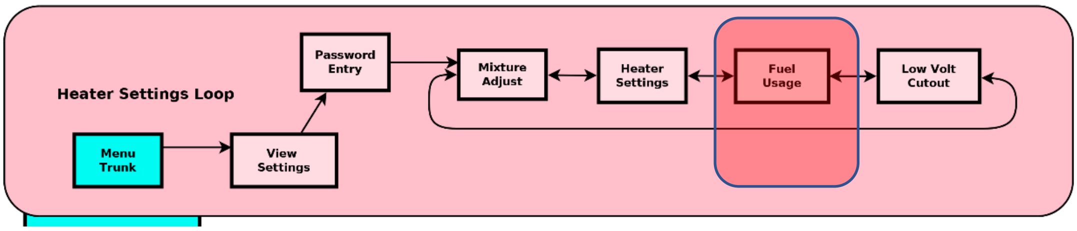
Saving Settings

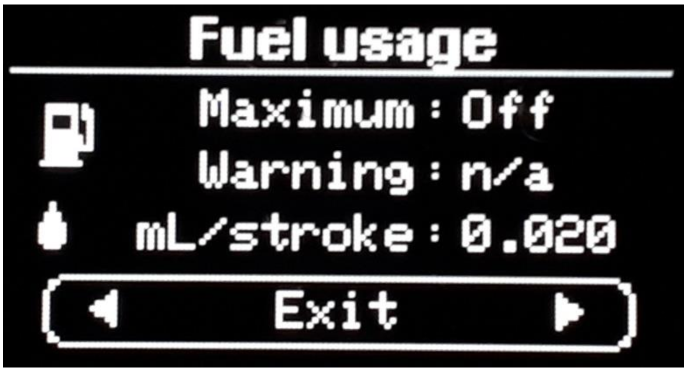
Press UP to
confirm save



STORING

Fuel Usage menu *(Heater Settings Menu Loop)*




Fuel usage

Maximum : Off

Warning : n/a

mL/stroke : 0.020

The Afterburner can estimate fuel usage by accumulating the pump cycles over time.

The typical pumps average around 0.02mL / stroke.
The pump frequency (Hz) determines how may strokes per second the pump performs.

Excess Fuel Usage Shutdown

Fully exhausting the fuel tank leads to air being introduced into the fuel line, which can be troublesome to clear depending upon the location of the pump.

To avoid this the Afterburner can be configured to shut down the heater when an excess quantity of fuel has been consumed, since the last fuel gauge reset.

A warning can be configured when the last remaining fuel is being consumed, prior to shut down.

Maximum Fuel Usage

The Maximum Fuel usage setting determines how much fuel is allowed to be consumed before shutting down the heater. The value is can be changed in 100mL increments.
Once the accumulated usage exceeds this figure, the heater will be shut down.
If fuel usage already exceeds this value, attempts to start the heater will be denied.

Fuel Usage Reset

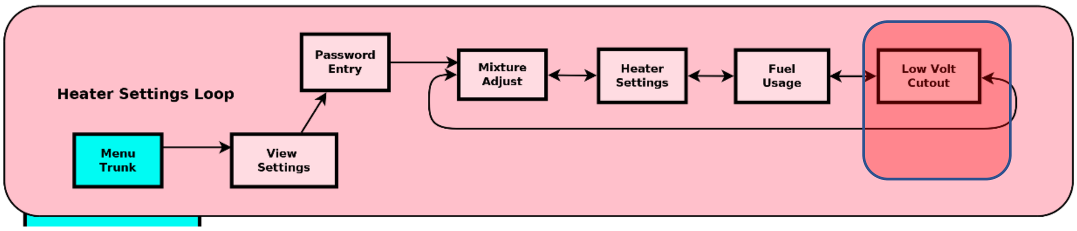
When the tank is refilled, the fuel usage must also be reset.
Fuel usage can be reset from the Mode Selection Menu in the Root Menu loop, or the web page.

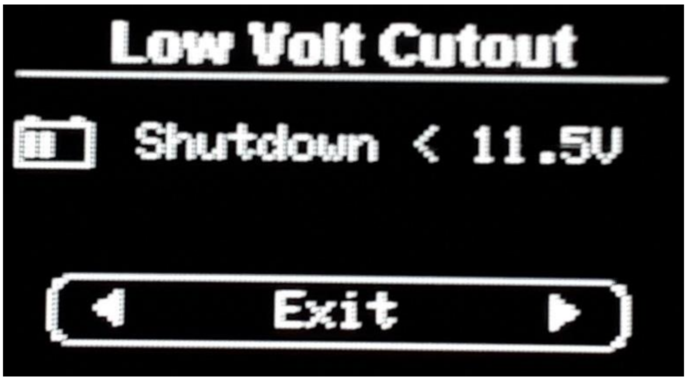
A GPIO input can also be configured so a 1 second press of a push button will reset the fuel usage. Such a button mounted near the fuel tank is a convenient way to deal with the reset.

Fuel Usage Warning

To give advance warning of the shutdown potentially occurring, a warning level can be set for the last remaining quantity of fuel. The value can be changed in 100mL increments.
This value is the headroom before empty, not absolute fuel usage.

Low Voltage Cutout menu *(Heater Settings Menu Loop)*




Low Volt Cutout

Shutdown < 11.5V

Exit

To afford protection of the battery from excess depth of discharge, the Afterburner can be configured to shut down the heater if the voltage reported by the heater falls less than the threshold defined in this menu.
The voltage must remain less than the threshold for at least 10 seconds to guard against momentary dips such as when starting a vehicle.

Adjustment of Low Voltage Cutout

Select the edit field by pressing **UP**.

Adjust the value by using **LEFT** or **RIGHT**.

If the value is reduced below 10V *for a 12V system* or 20V *for a 24V system*, the Low Voltage Cutout feature is disabled.

Save the setting by pressing **CENTRE**.

Communications

Th great advantage the Afterburner offers is the ability to have full control and monitoring of your heater from your smart phone device.

Connection methods

The primary methods being:

- Classic SPP Bluetooth *(Serial Port Protocol)*
- WiFi

Bluetooth access

Android devices support classic SPP Bluetooth.

iOS devices use BLE (Bluetooth Low Energy) which is not supported by the Afterburner hardware.
WiFi connectivity is fully supported on iOS devices.

WiFi access

WiFi connectivity can be hosted directly by an inbuilt access point, or preferably configuring the Afterburner to connect to an existing WiFi network that allows Internet access.

Allowed WiFi modes are:

AP only	The Afterburner hosts an Access Point – NO Internet connectivity
AP+STA	The After hosts its own AP, and is connected to a WiFi network
STA only	The Afterburner is only connected to a WiFi network

The last two modes allow the Afterburner to gain Internet access.

Having Internet access available allows the Afterburner to detect and on demand update the operating firmware.
It can also readily obtain the latest web page content.

Internet access is essential for MQTT functionality.

WiFi access also allows debug monitoring and some setups best achieved with a real alphanumeric keyboard.

WiFi capabilities

WiFi connections allow the following control and monitoring methods

- Direct web page
- MQTT for remote Internet access control and monitoring.
- Telnet for some setup and internal monitoring.

Classic Bluetooth SPP Connectivity *(Serial Port Protocol)*

Using an Android application, you can readily control and monitor your heater via a Bluetooth connection.

To do so, you must:

1. Pair your Android device with the Afterburner.
2. Launch the Diesel Heater application on your Android device.

Bluetooth pairing

1. Using your Android device's Settings menu, locate the Connections menu.
2. If not already turned on, enable Bluetooth.
3. Press upon the Bluetooth text header.
4. The device will now typically scan for any Bluetooth devices, also showing previously paired devices.
5. Scroll to the bottom of the Android display and select the Diesel Heater device from the Available Devices list.
6. To pair you need to enter the correct passcode, which is "1234".
7. The Afterburner device should now appear in the paired devices list.

Android Application

The Android application is a custom design, created using MIT App Inventor 2:
http://ai2.appinventor.mit.edu/

The Android application itself is available as part of the Open Source repository hosted on GitLab:
https://gitlab.com/mrjones.id.au/bluetoothheater

It is not available via Google Play.

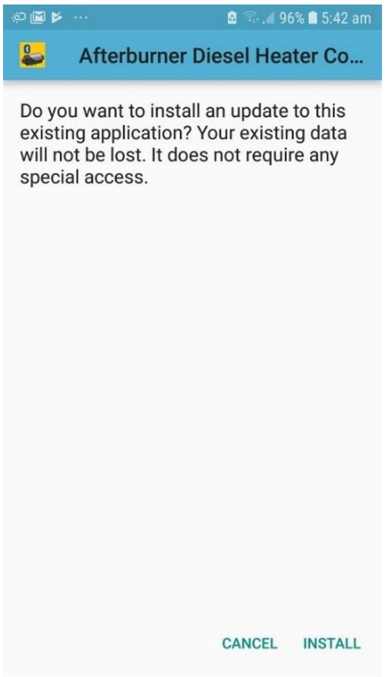
Installing the Android application

Using App Inventor can be somewhat intimidating, especially with the application design as it now stands.

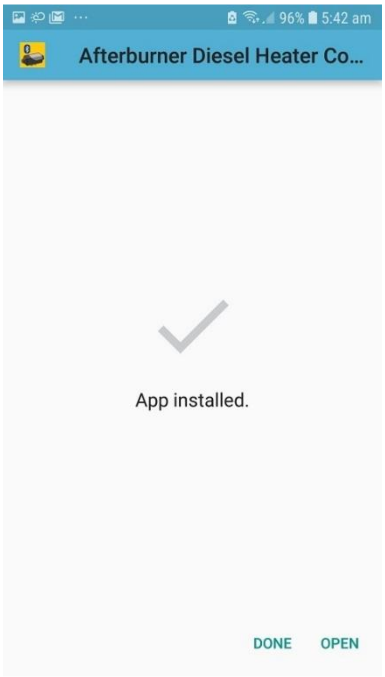
To ease the installation on your Android device, you can download the latest application that was compiled and saved as a .APK file from App Inventor.
Download and save the **Android Bluetooth application** to your Android device from http://www.mrjones.id.au/afterburner/firmware.html.

Once downloaded it is then simply a case of installing the application of your Android device, which you do by opening the .APK file you have downloaded.

You may be greeted with a warning about installing from an untrusted source, you will need to accept this before proceeding. If you cannot bring yourself to do this, you can always use App Inventor and upload from there directly to your device.



Do you want to install an update to this existing application? Your existing data will not be lost. It does not require any special access.



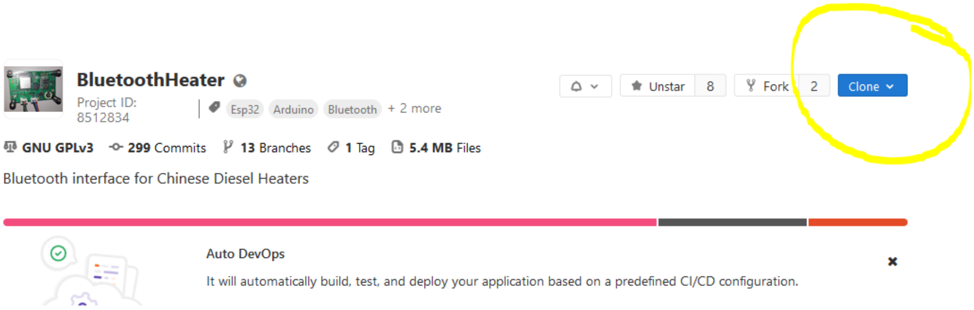
App installed.

Modifying the Android application

You must Clone the repository, open MIT App Inventor (requires Google login), upload the design from the repository to your App Inventor login, then upload to your Android device.

Whilst this may seem tedious, remember this is an Open Source design and you are free to make changes as you see fit! A word of caution though, the block based design is a mongrel to come to grips with, especially with the number of code blocks already present.
Rest assured though to just upload the design to your device the process is not all that difficult:

1. Clone the GitLab repository from https://gitlab.com/mrjones.id.au/bluetoothheater

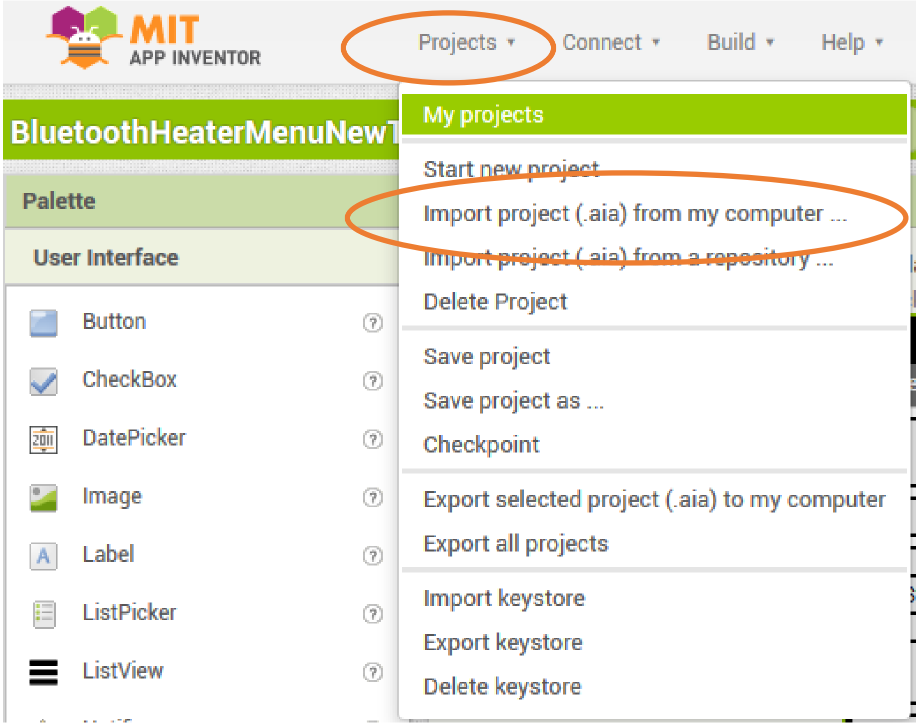


Bluetooth interface for Chinese Diesel Heaters

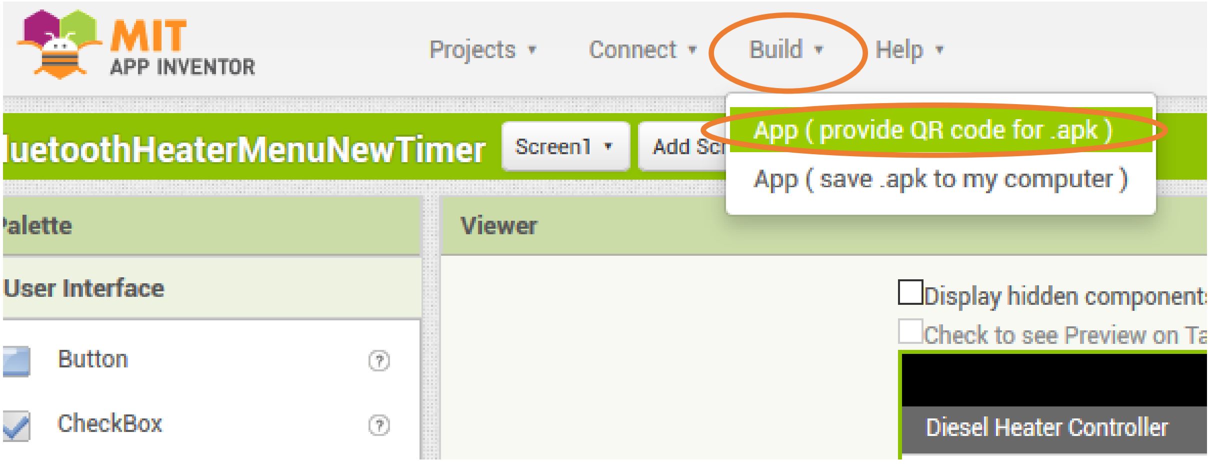
It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

2. Unless you want to push back proposed design changes, it may be easiest to clone as a ZIP file.

3. In the downloaded repository, locate the "AppInventor" subdirectory which is under the repository's root directory.
4. The file for the Android design will have a ".aia" extension, presently it is BluetoothHeaterMenuNewTimer.aia
5. Log into MIT App Inventor; http://ai2.appinventor.mit.edu/
6. Select a Google account to use, preferably the one associated with your Android device!
7. Once logged into App Inventor, select the "Projects" menu, then click upon "Import project (.aia) from my computer…"

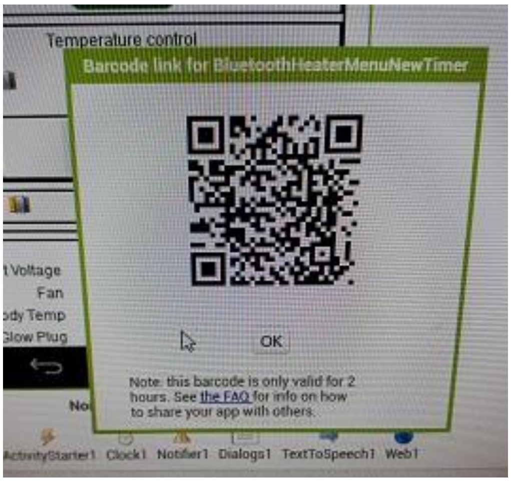


8. You will now need to install the App Inventor Companion on your Android device.
9. Visit Google Play, and search for "MIT AI2 Companion".
10. Select and install the MIT AI2 Companion.
11. Open the AI2 Companion application once installed.
12. **IMPORTANT: your Android device must be connected to the same network as your PC.**
13. Click upon the "Scan QR Code" on the AI2 Companion.
14. On your PC, in App Inventor, select; "Build", "App (provide QR code for .apk)"


MIT APP INVENTOR
Projects ▾
Connect ▾
Build ▾
Help ▾
luetoothHeaterMenuNewTimer
Screen1 ▾
Add Scr
App (provide QR code for .apk)
App (save .apk to my computer)
alette
Viewer
User Interface
Button
CheckBox
□ Display hidden component
□ Check to see Preview on Ta
Diesel Heater Controller

15. The design will then be compiled, and eventually a unique QR code for this instance will then be presented on your PC's screen.

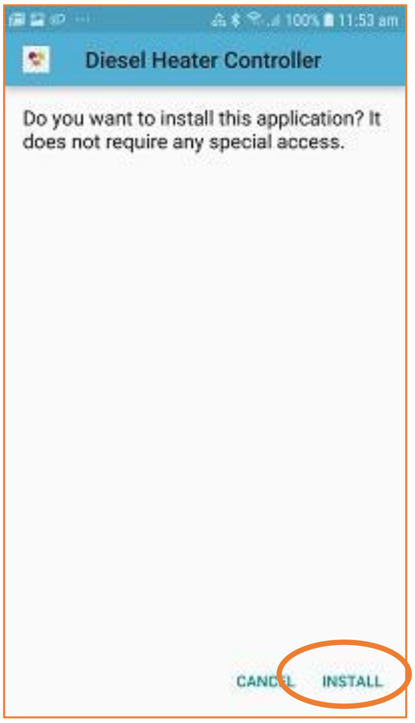
This QR code must be scanned using AI2 Companion on your Android device.



Barcode link for BluetoothHeaterMenuNewTimer

Note: this barcode is only valid for 2 hours. See the FAQ for info on how to share your app with others.

16. The application will then be uploaded to your Android device, but first you must authorise.



Diesel Heater Controller

Do you want to install this application? It does not require any special access.

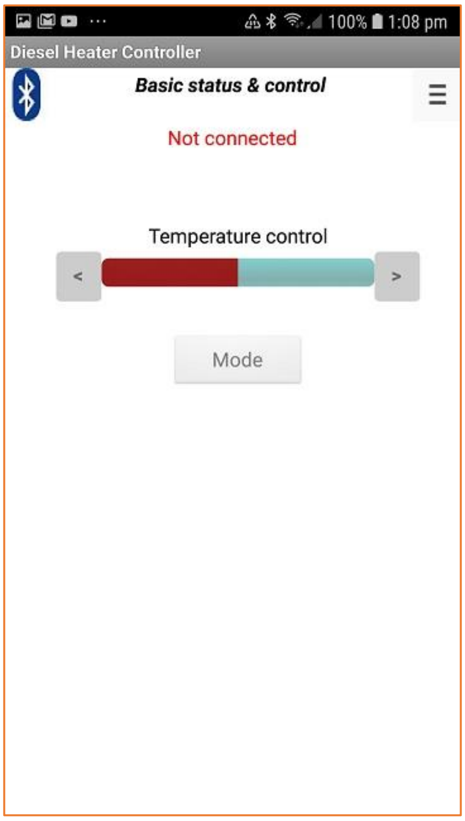
17. The application will now be installed to your Android device.

Using the Bluetooth Application

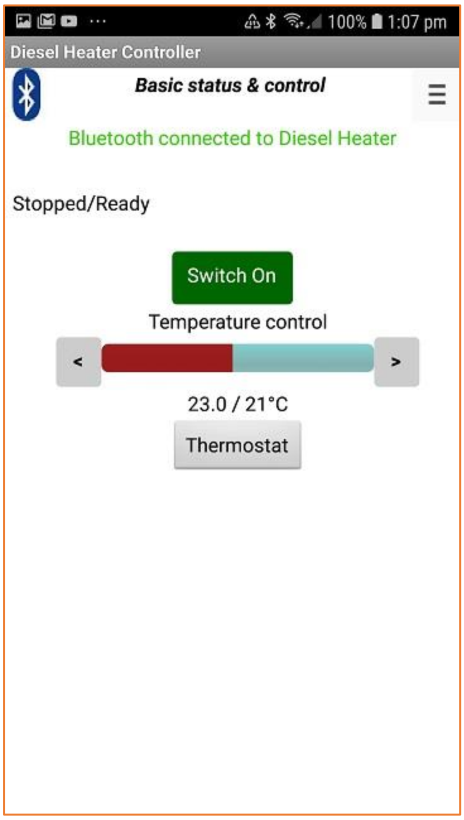
1. Once installed on your Android device, the Bluetooth application can now be opened:

2. You now need to connect to the previously paired Diesel Heater connection:
    a. Press upon the Bluetooth icon at the top left of the screen.
    b. Select the Diesel Heater connection from the provided list

3. A connection with the Afterburner should now be established and the screen will fill in with further information.



Not connected

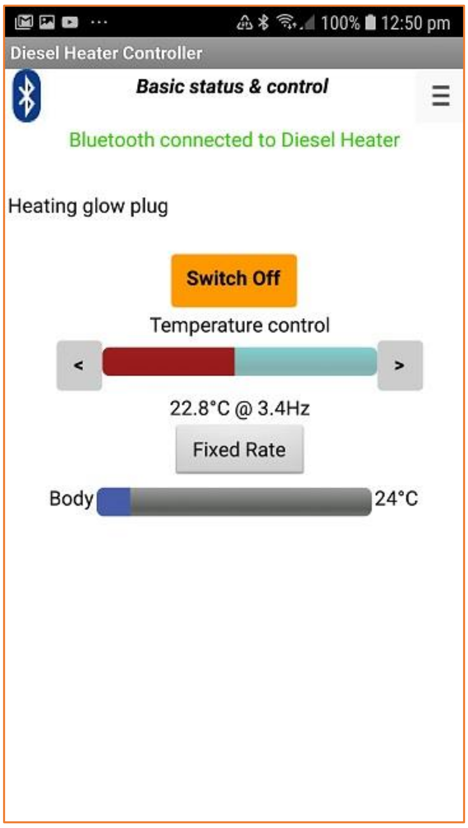


Bluetooth connected to Diesel Heater

Stopped/Ready

23.0 / 21°C

Go on, Press the big green button!



Diesel Heater Controller

Basic status & control

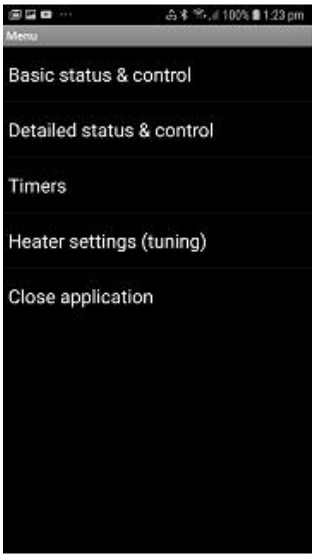
Bluetooth connected to Diesel Heater

Heating glow plug

Menus

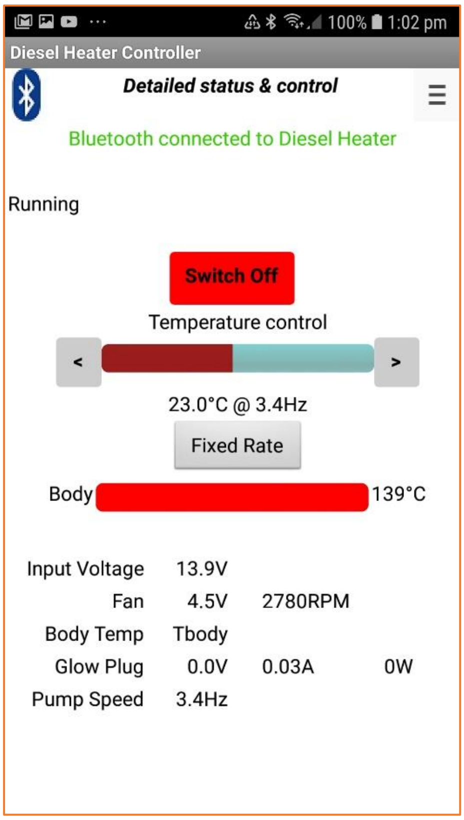
The application always starts in Basic Status & Control mode

To change the mode, press the menu at the top right corner:



Detailed Status & Control mode

This mode allows the detailed information about the heater to be presented.
Below the basic control information, a table of the running state is additionally shown.
This information should be useful for fault diagnosis, and heater tuning.



Diesel Heater Controller

Detailed status & control

Bluetooth connected to Diesel Heater

Running

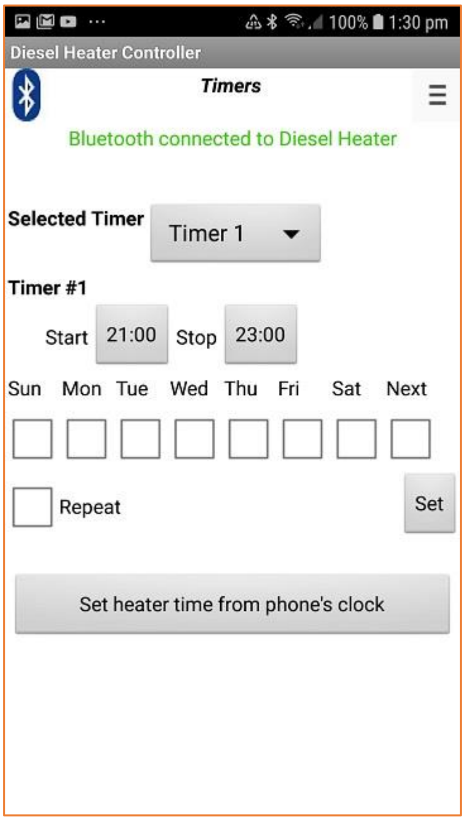
Input Voltage	13.9V		
Fan	4.5V	2780RPM	
Body Temp	Tbody		
Glow Plug	0.0V	0.03A	0W
Pump Speed	3.4Hz		

Timers mode

The timers provided by the Afterburner are extremely flexible.
14 individual timers can be defined. Each timer can be associated with any day desired, be a one-shot affair or repeat on a regular basis.

Essentially a Start and Stop time is defined, then which day of the week it should run.
The concept of next day becomes every day if repeat is enabled.

Select Timers from the menu list:



Diesel Heater Controller

Timers

Bluetooth connected to Diesel Heater

Selected Timer: Timer 1

Timer #1

Start 21:00 Stop 23:00

100% 1:30 pm
Sun Mon Tue Wed Thu Fri Sat Next
□ □ □ □ □ □ □ □
□ Repeat
Set
Set heater time from phone's clock

Which timer is to be programmed is selected from the “Selected Timer” drop-list.

Pressing the “Start“ or “Stop” times brings up a time setting dialog where the time can be adjusted.

The days the timer operates can be selected from the row of check boxes.
If none are selected the timer is disabled.
“Next” means the timer runs the next time the selected time occurs.

If we want the timer to always operate using the selected days, select the “Repeat” checkbox.
If “Next” and “Repeat” are selected the timer runs every day.

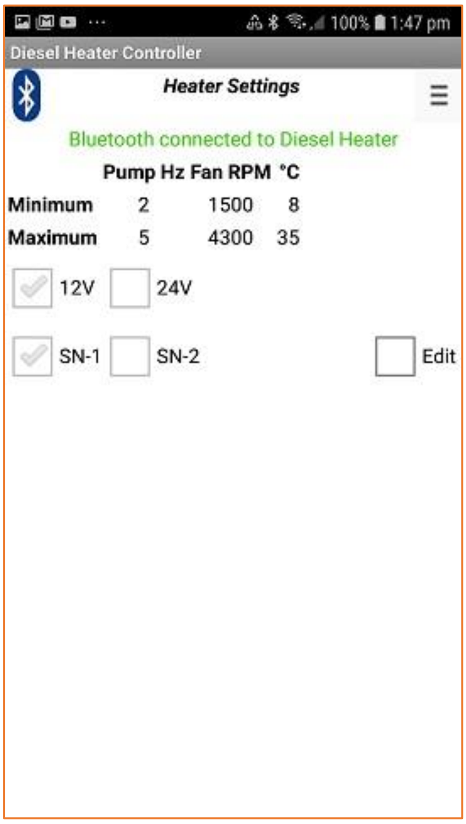
Importantly you must save you new timer definition into the Afterburner.

Press the “Set” button.
The proposed timer parameters are then sent to the Afterburner. It checks that the defined interval does not conflict with any other defined timer.
If OK the timer is set.
If in conflict, the timer is disabled (all days are cleared), and the timer that caused the conflict is advised via a popup message.

You can examine that timer by selecting it from the “Selected Timer” drop-list to determine the cause of the conflict.

Heater Settings mode

Heater settings mode allows the fuel mixture and running conditions to be observed and optionally altered.


100% 1:47 pm

Diesel Heater Controller

Heater Settings

Bluetooth connected to Diesel Heater

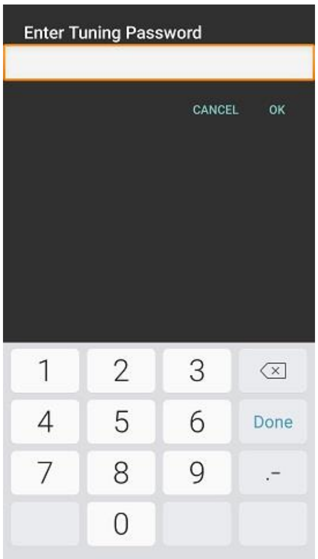
	Pump Hz	Fan RPM	°C
Minimum	2	1500	8
Maximum	5	4300	35

☑ 12V □ 24V

☑ SN-1 □ SN-2 □ Edit

Upon initial entry the display is purely passive, simply reporting the settings as observed from the communications protocol.

To change the settings you select the “Edit” checkbox, upon which you will then need to enter a password to gain access.


Enter Tuning Password

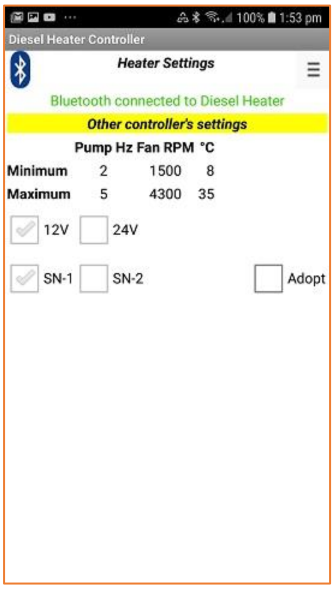
The password required is “1688”.

If entered correctly, the settings now appear in text edit boxes.

The values can now be changed by selecting and entering new values.

New settings are saved into the Afterburner by pressing the “Apply” button.

If another controller is in parallel with the Afterburner, the settings of that controller are shown instead, as they are dominant.


100% 1:53 pm

Diesel Heater Controller

Heater Settings

Bluetooth connected to Diesel Heater

Other controller's settings

	Pump Hz	Fan RPM	°C
Minimum	2	1500	8
Maximum	5	4300	35

☑ 12V □ 24V

☑ SN-1 □ SN-2 □ Adopt

In such a situation, it becomes possible to inherit those settings by pressing the “Adopt” button checkbox.

Once again the password “1688” needs to be entered to allow the inheritance to take place.

Wi-Fi Connectivity

The Afterburner can communicate over a Wi-Fi connection.

Available features currently being:

- Web server (port 80)
- Websocket connection (port 81, JSON status and control)
- Telnet debug stream (port 23)

The ESP32 processor allows the following Wi-Fi modes:

- AP (Inbuilt Access Point) - a Wi-Fi device can directly connect to the Afterburner.
- STA (Station) mode – The Afterburner connects to an existing Wi-Fi network.

The web server interface operates as either:

- An interface to control and monitor the heater
- A Configuration Portal to enable connection to another Wi-Fi network.

Connecting to the inbuilt Access Point

A mobile device is best used to connect to the Afterburner's AP (Access Point).
When browsing for available Wi-Fi networks, it will be named "Afterburner".

- Select the "Afterburner" network.
- Enter the password "**thereisnospoon**"

The connecting device should then be given an IP address.
The Afterburner's AP IP address will always be 192.168.4.1.
Your device will also probably warn there is no Internet connectivity (which there isn't).

Wi-Fi Configuration Portal

To allow the Afterburner to connect to an already existing Wi-Fi network the credentials required to connect to that network need to be defined to the Afterburner.
The Wi-Fi Configuration Portal is used to define and save these credentials.

By default, when shipped, there will not be any credentials stored to connect to another Wi-Fi network.
The Afterburner can only offer its own inbuilt AP (Access Point).

Over this inbuilt AP, the Afterburner's web server can either:

- Serve the heater controller web pages
- Serve the Wi-Fi Configuration Portal

The web server operation mode is defined using the Wi-Fi Settings menu.

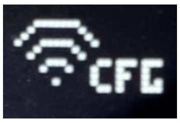
Enabling the Configuration Portal

Using the OLED Wi-Fi Settings menu, change the WiFi mode to **CFG AP only.**

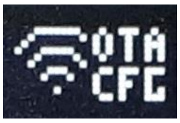
When the new mode is selected and accepted, the Afterburner will reboot and change web server to serve the STA Configuration Portal.

Note: you cannot select STA (Station) modes when no credentials for another Wi-Fi network are stored.

After reboot, the Afterburner advises the web server is using the Configuration Portal by showing CFG against the Wi-Fi icon.



or

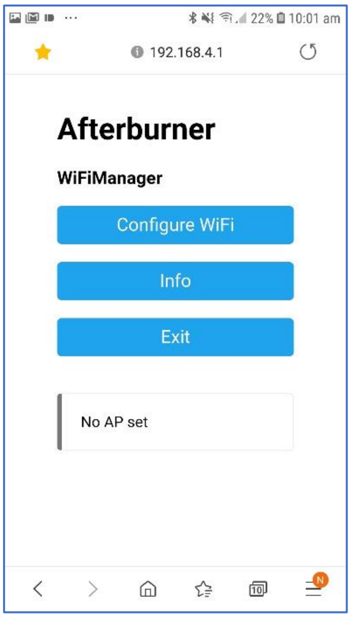


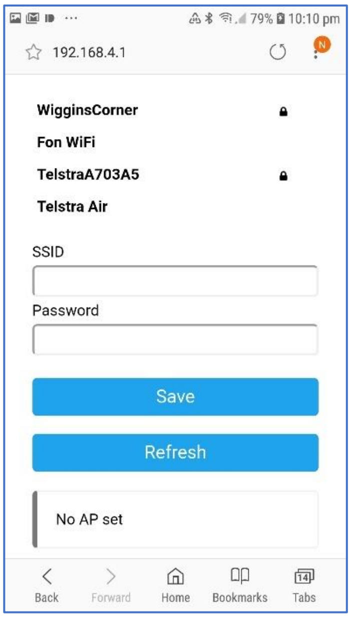
Connecting to the Configuration Portal

Connect the device to the Afterburner's Access Point.

Open a web browser on the device and enter the URL http://192.168.4.1

The Configuration Portal will then appear.
Press upon **Configure WiFi**:





A list of available access points is presented.

- Select the appropriate SSID for your network
- Enter the password for that network.
- Press **Save**

The selected SSID and the password credentials are then saved into the Afterburner's non-volatile memory.

- The ESP32 will reboot.
- The web server will now serve the Diesel Heater Web Control page.

Web server connectivity is now available via:

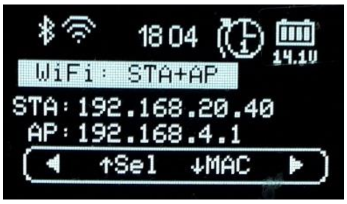
- The Afterburner's inbuilt access point (192.168.4.1)
- The IP address that was allocated by your local Wi-Fi network.

The IP addresses are visible via the OLED Wi-Fi Settings menu, beside the **STA:** and **AP:** entries:

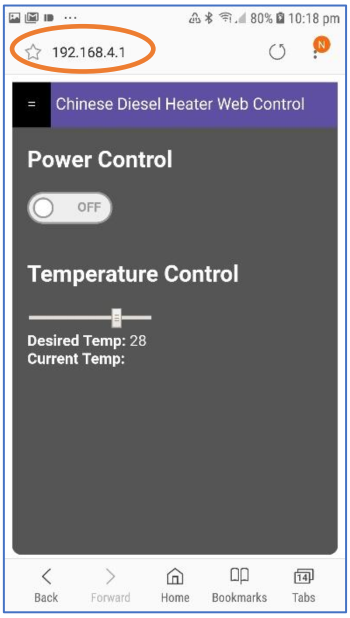
STA only mode

If you wish to switch to STA only, you can now do so in the WiFi Settings menu by changing the mode from **AP+STA** to **STA only**.
The Afterburner will attempt to connect to the STA network if it is present, otherwise it will fall back to AP only mode.

STA only mode reduces power consumption as it is not required to regularly send the AP beacon.



WiFi: STA+AP
STA: 192.168.20.40
AP: 192.168.4.1



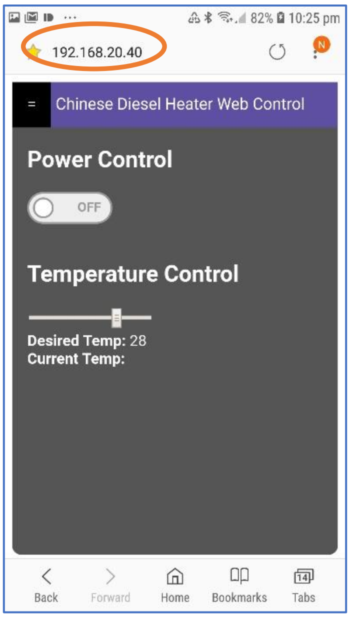
192.168.4.1

Chinese Diesel Heater Web Control

Power Control

Temperature Control

Desired Temp: 28
Current Temp:



192.168.20.40

Chinese Diesel Heater Web Control

Power Control

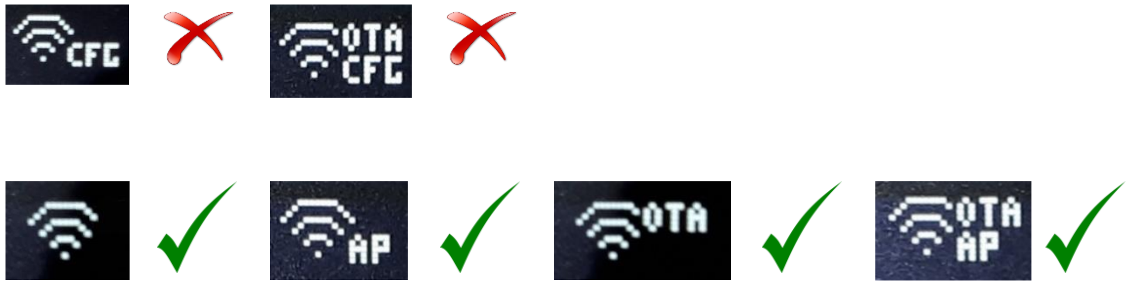
Temperature Control

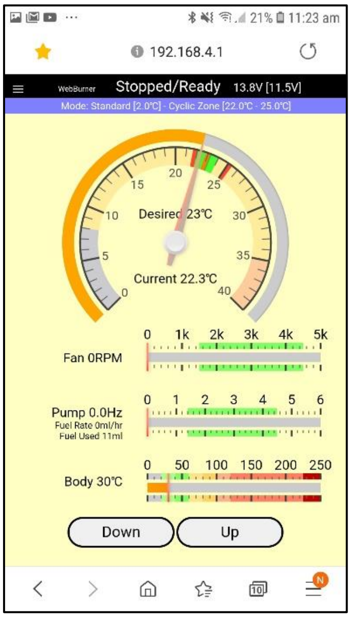
Desired Temp: 28
Current Temp:

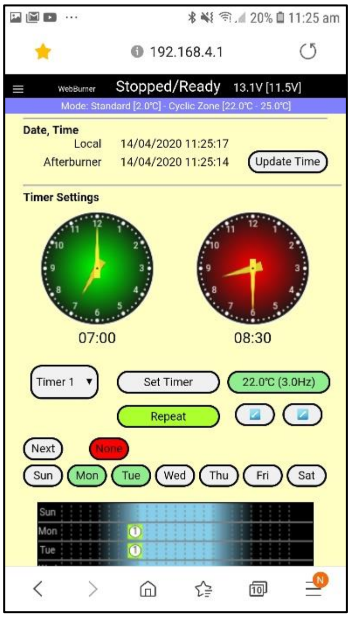
Heater Web Control

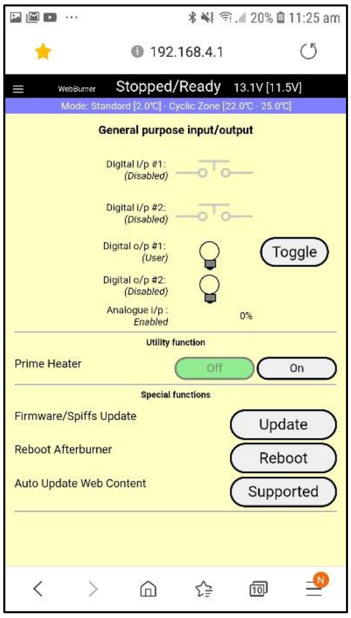
The primary purpose for Wi-Fi connectivity to the Afterburner is to be able to control and monitor your heater via a web browser.

To connect to the heater control web pages, the Afterburner must not be configured to serve the Configuration Portal.
On the OLED screen, the Wifi icon must not indicate CFG:




21% 11:23 am
192.168.4.1
WebBurner Stopped/Ready 13.8V [11.5V]
Mode: Standard [2.0℃] - Cyclic Zone [22.0℃ - 25.0℃]
Desired 23℃
Current 22.3℃
Fan 0RPM
Pump 0.0Hz
Fuel Rate 0ml/hr
Fuel Used 11ml
Body 30℃


20% 11:25 am
192.168.4.1
WebBurner Stopped/Ready 13.1V [11.5V]
Mode: Standard [2.0℃] - Cyclic Zone [22.0℃ - 25.0℃]
Date, Time
Local 14/04/2020 11:25:17
Afterburner 14/04/2020 11:25:14
Timer Settings
07:00
08:30
Timer 1
22.0℃ (3.0Hz)


20% 11:25 am
192.168.4.1
WebBurner Stopped/Ready 13.1V [11.5V]
Mode: Standard [2.0℃] - Cyclic Zone [22.0℃ - 25.0℃]
General purpose input/output
Digital i/p #1: *(Disabled)*
Digital i/p #2: *(Disabled)*
Digital o/p #1: *(User)*
Digital o/p #2: *(Disabled)*
Analogue i/p : *Enabled* 0%
Utility function
Prime Heater
Special functions
Firmware/Spiffs Update
Reboot Afterburner
Auto Update Web Content

The initial heater control web page allows basic heater control; on/off, temperature demand, fuel reset. To toggle to thermostat mode you need to use the Thermostat menu which you can conveniently access by clicking upon the blue bar atop the screen.

Several menus are available via the menu bar at the top left:

- Home
- Thermostat, Cyclic and Frost modes
- Environmental sensors
- Fuel mixture settings
- Date/time & timer settings
- System settings
- System functions
- Comms settings

The web pages are retrieved from the ESP32's SPIFFS partition.
The files in the SPIFFS partition can be updated using by browsing the Afterburner's **/update** inbuilt web page.
Details on how to upload are in the Firmware updates section.

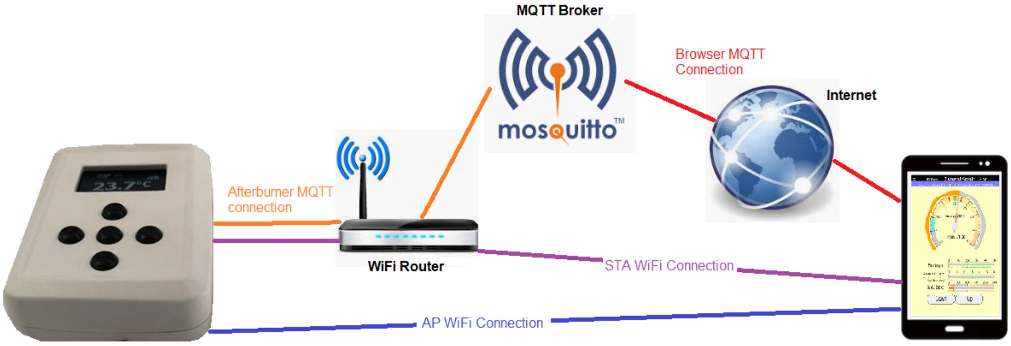
Using web page as an MQTT client

The web content has the smarts built in to allow gathering of the information that populates all the fields from an MQTT broker, instead of directly from the Afterburner's web server.

The Afterburner must firstly be setup to publish to a MQTT broker. Then you point your browser to the same MQTT broker in the web page's Comms Settings menu.

Once configured, you will have access to your heater where ever you have internet access!

MQTT Connectivity



The Afterburner can connect to an MQTT broker allowing the ability to integrate information from the Afterburner into a larger home automation scheme, or the exciting possibility of controlling and monitoring your heater via a remote connection over the Internet.

The MQTT broker can be hosted on your own home network, or elsewhere on the Internet.
Internet based brokers obviously allow remote access easily as they are already beyond any local firewall, but nothing prevents you making your own private broker visible from the Internet by adding it to your exposed connections.
The details of exposing a private broker are beyond the scope of this document as it is very dependent upon what site equipment is in use.

Topics

MQTT operates by using topics to provide or collect information to/from elsewhere.
In MQTT speak, this is Publishing and Subscribing.

Publishing is pushing topics to the broker.
Clients express interest in a specific topic by subscribing to a topic
The broker maintains a list of clients and the topics they would like to see.
When new topics are published, they are then relayed by the broker to the clients subscribed to it.

It is quite feasible a topic may simply be lost at the broker if no one has subscribed to it.

In the afterburner, there are two conventions used for the topic names:
“Global” topic that provides detailed information within a variable number of named entities, JSON encapsulated payload.
“Specific” topics that only provide JSON encapsulated information about one specific named entity.

The specific topics are only a subset of the full information available via the JSON topic, but are sufficient to provide remote control and monitoring of the Afterburner by simple MQTT clients.

Topic prefix

It is important that the information to/from the Afterburner is uniquely identified.
This is especially important should you have multiple Afterburner’s attached to the same broker, or indeed the final MQTT client.
It is vital for the prefix to be unique if you use a public broker.

Earlier versions, preceded the output topics with the prefix ***Afterburner***.
This prefix could be manually altered, but was at risk of not being changed to something unique.

A guaranteed unique name can be achieved by including the last 3 hex octets of the MAC address in the topic prefix.
(The first 3 MAC octets are constant and identify the vendor/device, ie Espressif ESP32)

Eg: if your MAC is 24:0A:C4:C0:AB:95, set the topic prefix to be ***AfterburnerC0AB95***.
Do not include the colons, they may lead to unpredictable behaviour.

V3.2+ firmware now enforces a fixed topic prefix of ***Afterburner*** suffixed with last 3 octets of the STA MAC address. i.e. ***AfterburnerC0AB95*** as per the example above.

The following descriptions use ***topicPrefix*** to indicate this unique name.

Connection State

Upon connection to a broker, it is generally recommended to publish a retained topic that indicates the connection is available to the Afterburner.

Likewise, should the connection be lost, indication should also be provided the unit is not available. The broker offers a “last will and testament” feature that allows clients to be informed upon loss of the Afterburner’s connection.

The same topic name is used for both instances.

Online Notification

When the Afterburner establishes the broker connection, the following topic is published and retained by the broker:

Topic	Payload
topicPrefix/status	“online”

Last Will and Testament

When the Afterburner establishes the broker connection, the last will & testament is alos defined. Should the broker connection unexpectedly disappear, the broker will then publish the following:

Topic	Payload
topicPrefix/status	“offline”

Control and Status

The Global Topic is /JSONout
The Specific Topics start with /sts *(after the topicPrefix)*

Publish

The Afterburner publishes to the following topics (outgoing status):

Topic	Payload
topicPrefix/JSONout	JSON encapsulated string providing full detailed status of the heater state
topicPrefix/sts/Run	0=heater OFF, 1=heater ON
topicPrefix/sts/RunState	An integer enumeration indicating the current running state of the heater

topicPrefix/sts/RunString	String that describes the running state of the heater
topicPrefix/sts/TempDesired	Desired temperature in °C
topicPrefix/sts/TempCurrent	Current temperature in °C – from sensor used for thermostat
topicPrefix/sts/Temp2Current	Current temperature in °C – from optional second sensor
topicPrefix/sts/Temp3Current	Current temperature in °C – from optional third sensor
topicPrefix/sts/TempBody	Current temperature in °C of the heater's casing
topicPrefix/sts/ErrorState	An integer enumeration indicating the current error state of the heater
topicPrefix/sts/ErrorString	String that describes the error state of the heater
topicPrefix/sts/Thermostat	0=fixed pump demand, 1=thermostat mode active
topicPrefix/sts/PumpFixed	Desired pump delivery rate, in Hz
topicPrefix/sts/PumpActual	Current pump delivery rate, in Hz
topicPrefix/sts/FanRPM	Current fan RPM
topicPrefix/sts/InputVoltage	Current input voltage to heater unit
topicPrefix/sts/GlowVoltage	Current voltage applied to glow plug
topicPrefix/sts/GlowCurrent	Current amperage applied to glow plug
topicPrefix/sts/GPalg	Analogue input value applied to analogue input as a percentage
topicPrefix/sts/FuelUsage	Accumulated fuel consumption since last reset, mL
topicPrefix/sts/FuelRate	Current rate of fuel consumption, mL/hr

Subscribe

The Afterburner subscribes to the following two topics (incoming control):

Topic	Payload
topicPrefix/JSONin	JSON encapsulated command, using defined dictionary names
topicPrefix/cmd/#	# = wildcard wildcard can be any valid JSON dictionary name and will be treated as if it were part of a JSONin command string

A client can choose whether to publish commands using either /JSONin or /cmd/*JSONname*.
eg. The both the following are valid usages:

Topic	Payload
topicPrefix/JSONin	{"TempDesired":22}
topicPrefix/cmd/TempDesired	22

The /JSONin topic can support multiple names, whereas the /cmd/xxx is singular

eg. This is valid for /JSONin, but must be sent as individual /cmd's:

Topic	Payload
topicPrefix/JSONin	{"TempDesired":22,"Run":1,"Thermostat":1}

JSON dictionary

The JSON names are defined in the JSON dictionary document.

Configuring MQTT

First and foremost, the Afterburner must be able to connect to a broker, which means it must be configured to use STA WiFi mode.
STA mode being when the Afterburner is connected to a local WiFi network.

Once connected, discover the Afterburner's STA IP address using the OLED menus.

Using a telnet client, connect to the Afterburner (telnet port 23).
This accesses the general debug information output, from there you can now access a MQTT configuration menu.

As debug information freely flows, it is of benefit to suppress the notifications that appear when JSON data is packaged for delivery.
Hit 'J' to toggle off JSON reporting. This avoids unwanted distraction whilst using the MQTT menu.

Hit 'M' to present the MQTT setup menu

Keystroke	Function
1	Enter broker's IP address The IP address must be entered using dotted notation
2	Enter broker's port *(default 1883)*
3	Enter the username to access the broker
4	Enter the password to access the broker
	The enforced topic prefix.
6	Enter the desired Quality of Service
7	Enable or disable MQTT connection
<ENTER>	Save and exit the parameters
<ESC>	Abort any changes

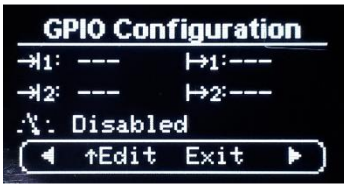
GPIO Capability

The Mk2+ PCBs provide extra input and output conditioning circuitry to allow the ready addition of external switches or status indicators/relays.

The Mk1 PCB can still provide these GPIO functions, but care needs to be taken to protect the fragile inputs of the ESP32 processor. (eg 3.3V logic, intolerant to errant voltages).

A digital input is considered as being active when the external pin is connected to 0V.
The Mk2 PCB however due to the input transistor inverts this to a high level at the ESP32.
During initial birthing, the PCB version is identified so the correct digital input behaviour occurs for either PCB, in relation to the applied external signal input point.

By default, the GPIO functionality is disabled. It must be enabled via this menu in the User Settings menu loop:



```
GPIO Configuration
⇥1: ---      ↦1:---
⇥2: ---      ↦2:---
∿: Disabled
◀  ↑Edit  Exit  ▶
```

The top 2 left hand entries show the mode of the 2 digital inputs.
The 2 right hand entries show the mode of the 2 digital outputs.
The 3rd line shows the analogue input mode.

Navigation of the menu is explained in the User Settings Menu Loop section.

Digital Inputs

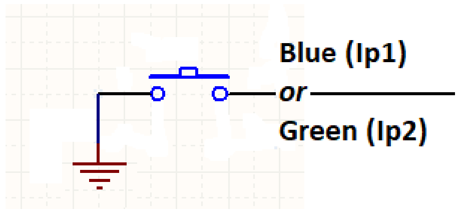


Figure 1: User furnished equipment

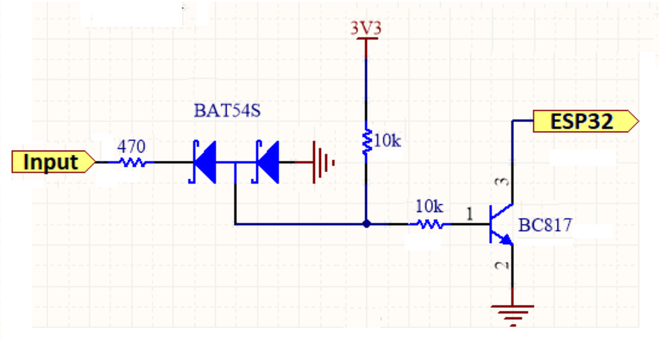


Figure 2: Digital Input circuit of Afterburner

The 2 digital inputs can be configured as:

Input #1 – Blue wire

• Disabled	
• Start – time sensitive	A short close (<1.5 seconds) will **start** the heater. A long close will **stop** the heater
• Run	Holding the input closed will run the heater, opening will stop the heater.
• Start/Stop	If stopped, a momentary close will start the heater, a subsequent momentary close will then stop the heater.

Input #2 – Green wire

• Disabled	
• Stop	A momentary close will stop the heater.
• External Thermostat	Closed: heater runs at high speed Open: heater runs at low speed. An optional open time can be defined to stop the heater after being held open for longer than the chosen interval. Heater is always started upon initial contact closure.
• Zero Fuel Usage	Fuel usage is zeroed if held closed > 1 second.

Input mode detailed descriptions

Start – time sensitive

A contact closure less than 1.5 seconds will request the heater to start.
A contact closure longer than 1.5 seconds will request the heater to stop.

This mode may be useful for those wishing to use GSM relay to start or stop the heater. By generating appropriately timed pulses (eg 1 second or 2 seconds) a single input can be used to start or stop the heater.

Stop

A contact closure will request the heater to stop.

Start / Stop

Alternate contact closures will request the heater to start or stop. No time sensitivity exists.

Run

The heater will run whilst the input remains closed.
If opened the heater will be requested to stop, performing the usual shutdown sequence.

An example usage may be using a conventional “dry contact“ furnace thermostat, with the heater starting and stopping according to the condition of the thermostat’s output. Bear in mind it takes over 5 minutes to start the heater, and a couple of minutes to shutdown the heater.

External Thermostat

This mode is specifically intended for use with an external “dry contact“ thermostat.
When the contact is closed, the heater will be made to run at maximum power.
When the contact is open the heater will be made to run at minimum power.

If the heater is off upon the initial closure, it is requested to start.

Additionally, a time period can be associated with this mode.
If the contact remains open longer than the selected time interval, the heater is requested to stop.

This allows operation solely using the wall thermostat. Initial operation will see the heater start, then cycle according to the setting. Finally disabling the thermostat will see the heater eventually stop once the time period elapse.
If the time period is set to zero, the heater will always run unless commanded to turn off.

Fuel Usage Reset.

This input allows a switch to be mounted nearby the fuel tank, to be used to reset the accumulated fuel usage when the tank is re-filled.
Resetting the accumulated fuel usage is important if you are using the Excess Fuel Usage shutdown feature to avoid running the tank dry, which usually requires the fuel line to be re-primed.

Digital Outputs

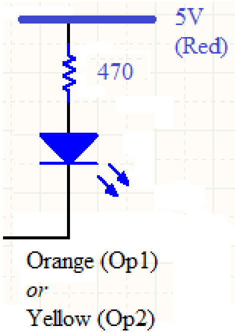


Figure 4: User furnished equipment: Driving an LED

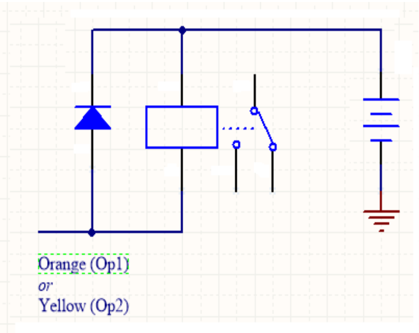


Figure 5: User furnished equipment : Driving a relay, diode must be included.

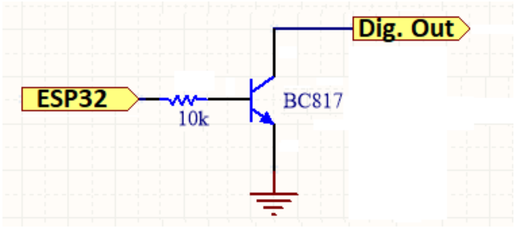


Figure 3: Afterburner digital output circuitry

The 2 digital outputs are offered as open collector devices, effectively providing a path to ground when active.

The load can be a 12V device, eg relay, if desired.
When driving a relay, a diode must be installed across the relay's coil as shown to manage the back EMF when the relay is turned off. Not doing so may harm the open collector output with the excess voltage spike generated as the magnetic field rapidly collapses.

The digital outputs can be configured as:

Output #1 – Orange wire

• Disabled	
• LED Status Indicator	Standby – OFF Heater starting – "Breathe up" illumination Heater running – Steady illumination Heater starting – "Breathe down" illumination
• User controlled	Can be made active upon user demand
• Temperature Threshold	Output active when either above or below a defined temperature threshold
• Run Status	Output is active when heater is in any running condition -from start through to final cooling phases

Output #2 – Yellow wire

• Disabled	
• User controlled	Can be made active upon user demand
• Temperature Threshold	Output can be made active when either above or below a defined temperature
• Run Status	Output is active when heater is in any running condition -from start through to final cooling phases

Output mode detailed descriptions

User controlled

The Output can be made active upon request by the user.
The state of the line can be toggled by holding the left or right keypad switches when in the GPIO Status, Detailed Status, Basic Status or Clock menus.
It is also possible to control these via Bluetooth or WiFi applications, eg web browser.

LED Status

This mode intends a LED to be connected from 5V via a resistor to the output line to indicate the heater's operating state:

- When the heater is off, the LED will be off.
- When the heater is starting, the LED will cycle in an increasing brightness manner.

- When the heater is running, the LED will be constantly on.
- When the heater is shutting down, the LED will cycle in decreasing brightness manner.
- If the heater is suspended due to the cyclic mode threshold being crossed, the LED will momentarily flash every second or so.

Temperature Threshold

The output will monitor the ambient temperature and activate as required.
The output can be defined to be active when above or below the threshold.
When defining the GPIO function to be temperature threshold, either:

- Hold the right button to enable editing the temperature threshold, and making the output become active if **above** the defined temperature. OR
- Hold the left button to enable editing the temperature threshold, and making the output become active if **below** the defined temperature.
- Adjust the new temperature threshold using the **UP/DOWN** keys.
- Accept the threshold by pressing the **CENTRE** key

Run Status

The output is made active whenever the heater is not in standby, ie starting, running, or shutting down.
If the heater is suspended (in standby) due to Cyclic Mode, the output is not active.

Analogue Input – White wire

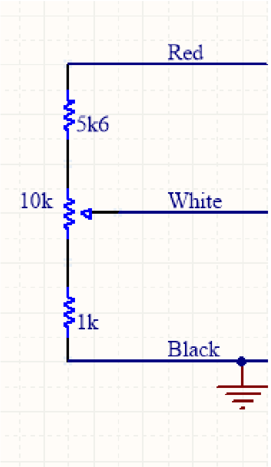


Figure 6: Suggested input circuit for a potentiometer

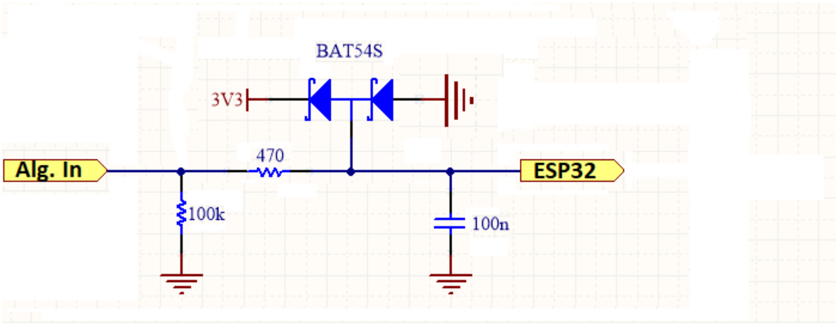


Figure 7: Analogue input circuit of the Afterburner

The analogue input accepts a voltage in the range from 0.0V to 3.3V.
The power supplied from the Afterburner on the GPIO connector is 5V so if using a pot you must pad the top end with a resistor to limit the maximum expected voltage, as shown above.
Note that there are clamping diodes in the input circuit, but these are a safety feature and should not be relied upon.

The single analogue input can be configured as:

• Disabled	
• Enabled	Input voltage applied to white wire is digitised and presented as a percentage of the usable range (0 – 3.3V)

Arduino Environment

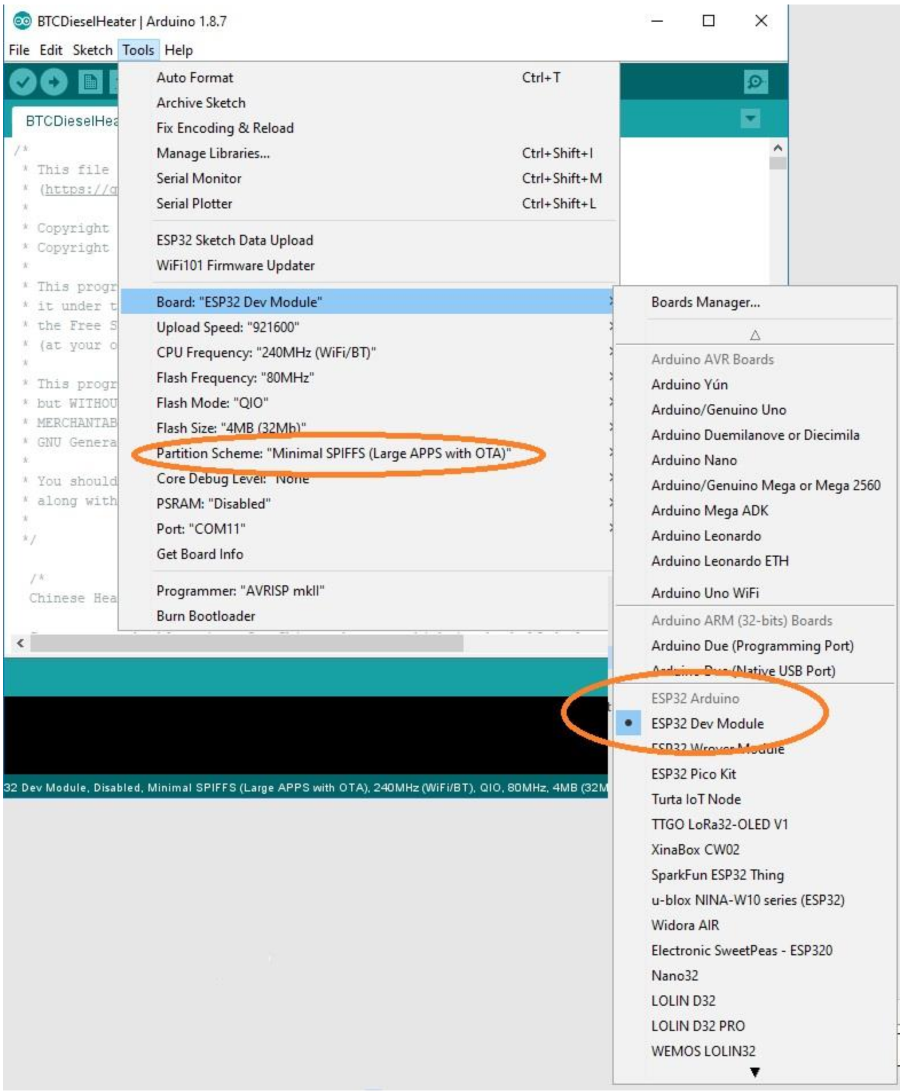
ESP32 build settings

Arduino Board Configuration

The ESP32 processor we are targeting in this design is housed within an ESP-WROOM-32 module.

Once the ESP32 environment package is installed into the Arduino IDE, select the **Standard ESP32 Dev kit board** configuration from the Arduino IDE.

This important so we can select the correct SPIFFS partition scheme.



Board: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Minimal SPIFFS (Large APPS with OTA)"
Core Debug Level: "None"
PSRAM: "Disabled"
Port: "COM11"
Programmer: "AVRISP mkII"

32 Dev Module, Disabled, Minimal SPIFFS (Large APPS with OTA), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32M

Minimal SPIFFS Partition Scheme

The compiled binary size with Wi-Fi code approaches 1Mbyte.
For some weird obtuse reason the Default Partition Scheme has produced weird behaviour despite 1.4MB being available; eg:

- Watchdog timeouts shortly after reboot, causing a constant reboot cycle (visible via serial port debug)
- Web pages not being served

- TelnetSpy not responding to telnet connections

For reliable operation, you must select the Minimal SPIFFS Partition Scheme from the Arduino board configuration options.
This option is only available with the standard ESP32 Dev Board setting.

Required libraries for Arduino environment.

ESP32 environment package

https://github.com/espressif/arduino-esp32 *or*
https://dl.espressif.com/dl/package_esp32_index.json (V1.0.1)

- Wi-Fi Manager : The one from stock library manager didn't work with esp32, try https://github.com/tzapu/Wi-Fi Manager/tree/development
- TelnetSpy : https://github.com/yasheena/telnetspy
- OneWire: https://github.com/stickbreaker/OneWire (for ESP32)
- DallasTemperature: https://github.com/milesburton/Arduino-Temperature-Control-Library
- WebSockets: https://github.com/Links2004/arduinoWebSockets
- RTClib: https://github.com/adafruit/RTClib
- ArduinoJSON: https://github.com/bblanchon/ArduinoJson

  The Arduino IDE insists upon installing the latest version. This will not work properly.
  Please refer to https://arduinojson.org/v5/faq/

- Adafruit_GFX: https://github.com/adafruit/Adafruit-GFX-Library
- Adafruit SH1106 (for ESP32): https://github.com/nhatuan84/esp32-sh1106-oled

Firmware updates

As the Afterburner software is still undergoing continual development, it may be necessary to upload a new firmware version in the future.

The following methods are available in V2.3.0 firmware onwards: *(ordered easiest to hardest)*

- Prompted automatic update from the Afterburner web site
- Browser upload
- OTA update, via a batch file
- Serial port upload, via a batch file.
- OTA or serial upload using the Arduino IDE

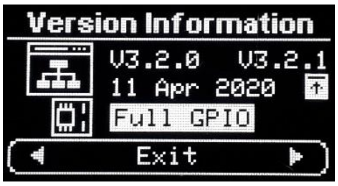
Automatic update

Provided the Afterburner is connected to a WiFi network, which itself is connected to the Internet, the Afterburner can interrogate the Afterburner project web site (http://afterburner.mrjones.id.au) and determine if a new firmware version is available.

Initially the Afterburner will animate an "Update available" icon upon the display's header area.
It will NOT automatically start the new firmware download unless authorised by the user.

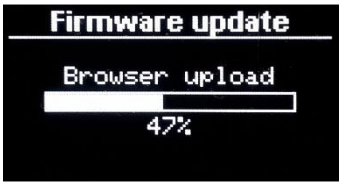
To authorise the upgrade, navigate to the Version Information menu, located in the System Settings menu loop.

Presuming an update is available, a larger animation takes place to the top right of the Version Information menu:


Version Information
V3.2.0 V3.2.1
11 Apr 2020
Full GPIO
Exit

To authorise the download, press **UP**, **CENTRE**, **UP**.

The Afterburner will then download the new firmware, showing the progress on the display:


Firmware update
Browser upload
47%

Once the download completes, the checksum is verified and the system reboots into the new firmware.
Should the download fail, the Afterburner still reboots, but the existing version will be retained.

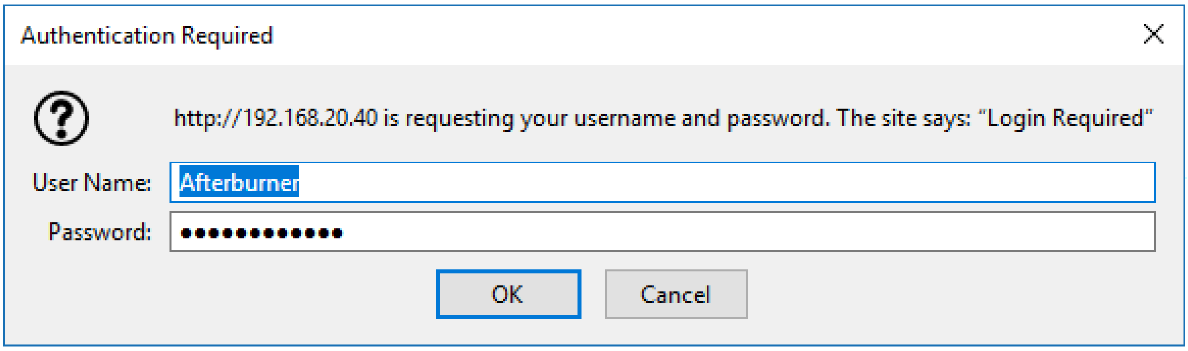
Browser update

Download the desired **pre-compiled firmware (binary image)** from http://afterburner.mrjones.id.au/firmware.html and save to the local PC, or even mobile phone.

Using a web browser, open Afterburner unit's browser update web page.
Note the Afterburner can be accessed using the inbuilt access point (192.168.4.1) or via a WiFi network, provided both devices are registered on the same WiFi network.

Browse to the Afterburner's IP address, eg if using AP mode http://192.168.4.1/update

This will then present a password confirmation page:



Authentication Required

http://192.168.20.40 is requesting your username and password. The site says: "Login Required"

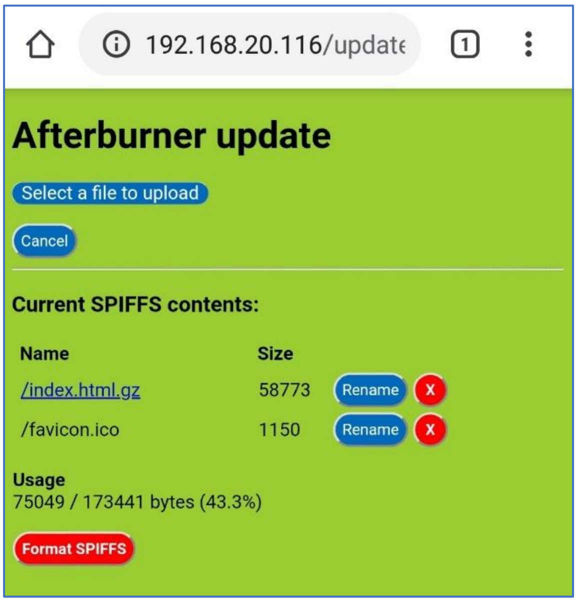
User Name: Afterburner

Password: ●●●●●●●●●●●●

The required credentials are:

- **Username**: Afterburner
- **Password**: BurnBabyBurn

The following screen should then be presented:



192.168.20.116/update

Afterburner update

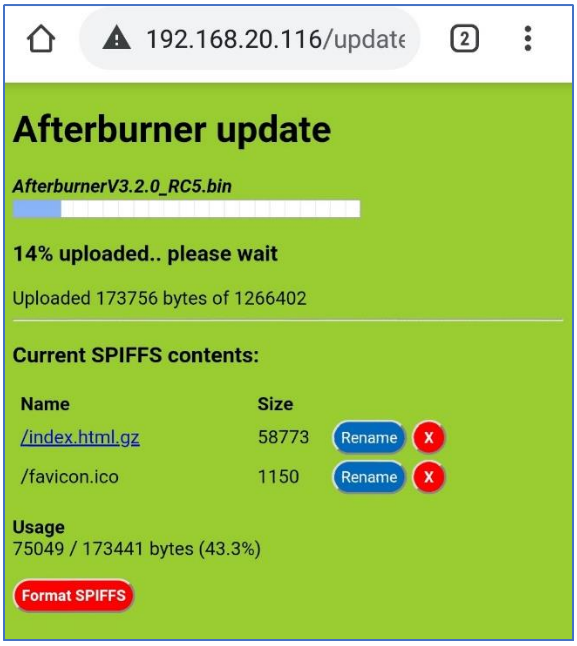
Current SPIFFS contents:

Name	Size		
/index.html.gz	58773	Rename	X
/favicon.ico	1150	Rename	X

Usage
75049 / 173441 bytes (43.3%)

Click upon **Select a file to upload** and locate the binary image previously downloaded from http://www.mrjones.id.au/afterburner/firmware.html.

Click upon flashing **Upload** button to start the upload of new firmware to the Afterburner, the Afterburner display will show progress, as will the web browser:



192.168.20.116/update

Afterburner update

AfterburnerV3.2.0_RC5.bin

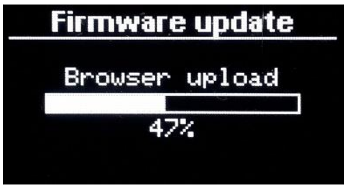
14% uploaded.. please wait

Uploaded 173756 bytes of 1266402

Current SPIFFS contents:

Name	Size		
/index.html.gz	58773	Rename	X
/favicon.ico	1150	Rename	X

Usage
75049 / 173441 bytes (43.3%)



Firmware update

Browser upload

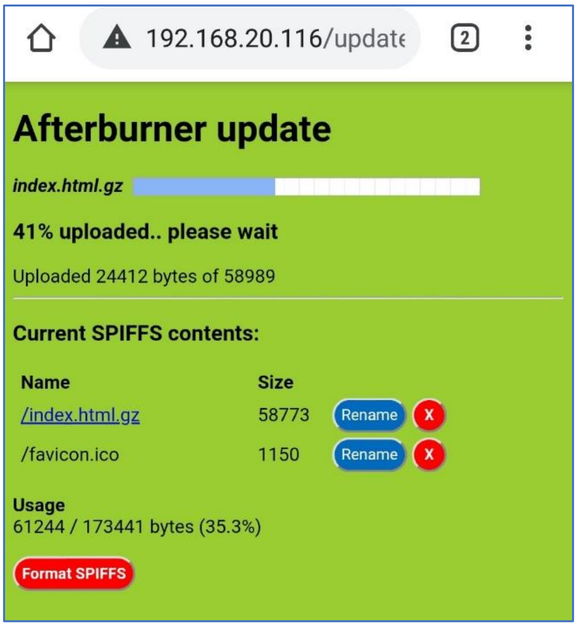
47%

Once the download completes, the checksum is verified and the system reboots into the new firmware.
Should the download fail, the Afterburner still reboots, but the existing version will be retained.

SPIFFS file uploads

Browser update can also be used to upload new web page content to the SPIFFS partition, instead of using the Arduino IDE.

The same procedure is followed, but instead of selecting the binary image, select the new *(typically .gz compressed HTML web page content)* file to upload:



192.168.20.116/update

Afterburner update

index.html.gz

41% uploaded.. please wait

Uploaded 24412 bytes of 58989

Current SPIFFS contents:

Name	Size		
/index.html.gz	58773	Rename	X
/favicon.ico	1150	Rename	X

Usage
61244 / 173441 bytes (35.3%)

OTA update (batch file)

Download the desired **pre-compiled firmware (binary image)** from http://www.mrjones.id.au/afterburner/firmware.html and save to a local Windows PC.

Download the **Firmware uploading scripts**, also available from the above web link.

Extract the uploading scripts to a convenient directory on the PC.

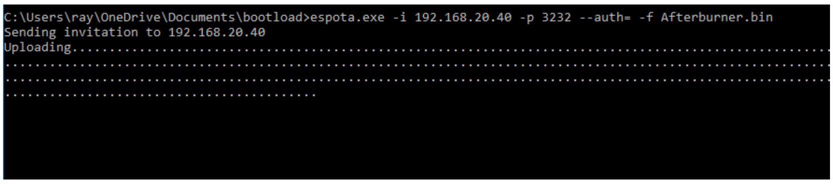
Modify the OTA.bat file to correctly reflect the Afterburner's IP address you will use to access the Afterburner via the Over The Air protocol:

```
espota.exe -i 192.168.20.100 -p 3232 --auth= -f Afterburner.bin
```

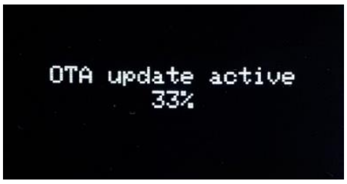
Copy the downloaded binary to the same directory.
Rename the binary image to **Afterburner.bin.**

Execute **OTA.bat.**

The upload to the Afterburner will now take place over the WiFi connection:



```
C:\Users\ray\OneDrive\Documents\bootload>espota.exe -i 192.168.20.40 -p 3232 --auth= -f Afterburner.bin
Sending invitation to 192.168.20.40
Uploading.....................................................................................................
..............................................................................................................
..............................................................................................................
.........................................
```



```
OTA update active
33%
```

Once the download completes, the checksum is verified and the system reboots into the new firmware.
Should the OTA download fail, the Afterburner still reboots, but the existing version will be retained.

Serial update (batch file)

Plug a FTDI USB adapter, **that has been jumpered for 3.3V operation**, into a Windows PC.
Using Device Manager determine the correct COM port of the USB adapter.

Connect the Tx, Rx and GND signals between the FTDI adapter and the matching Afterburner's debug serial port pins.

Prepare the Afterburner for serial bootload by performing the following sequence:

1. Press and HOLD the **EN** button *(connect **En** pin to 0V on Mk2 PCB)*.
2. Press and HOLD the **BOOT** button *(connect **Bt** pin to 0V on Mk2 PCB)*.
3. Release the **EN** button.
4. Release the **BOOT** button.

The ESP32 is now ready to receive firmware over the debug serial port.

Download the desired **pre-compiled firmware (binary image)** from http://www.mrjones.id.au/afterburner/firmware.html and save to a local Windows PC.

Download the **Firmware uploading scripts**, also available from the above web link.

Extract the uploading scripts to a convenient directory on the PC.

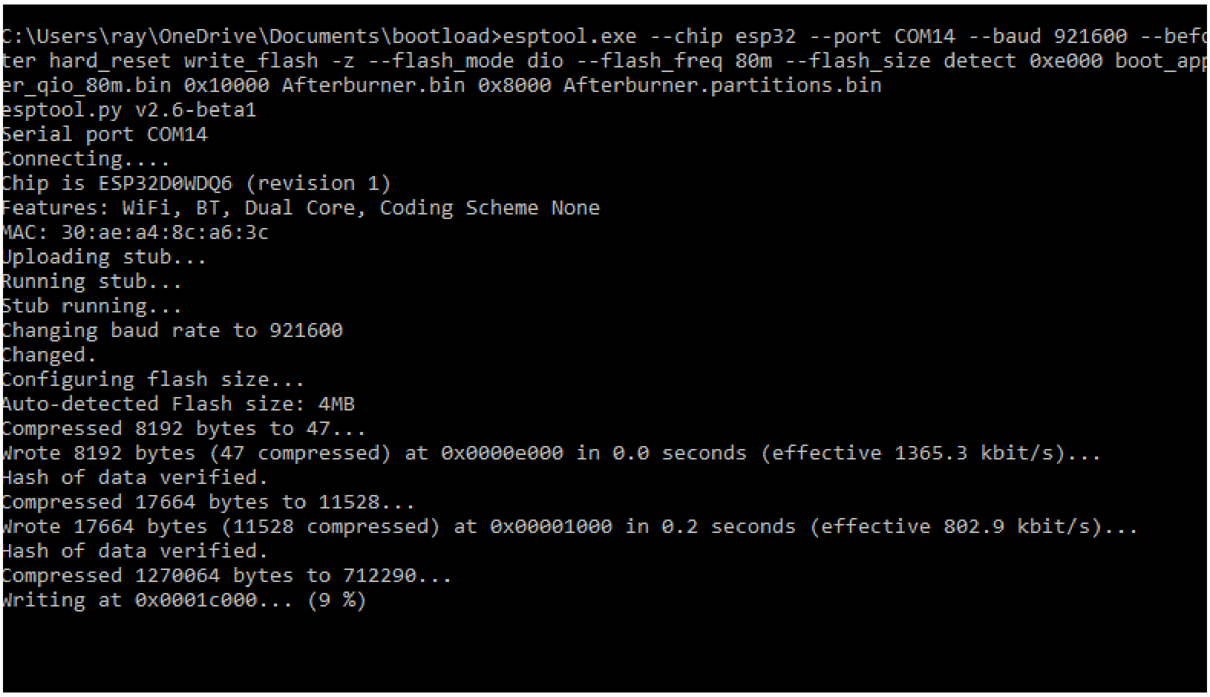
Copy the downloaded binary image to the same directory.
Rename the binary image to **Afterburner.bin.**

Modify the COM.bat file to correctly reflect the COM port that will be used to upload new firmware to the Afterburner *(immediately following –port)* **DO NOT CHANGE ANYTHING ELSE**:

```
esptool.exe --chip esp32 --port COM11 --baud 921600 --before
default_reset --after hard_reset write_flash -z --flash_mode dio -
-flash_freq 80m --flash_size detect 0xe000 boot_app0.bin 0x1000
bootloader_qio_80m.bin 0x10000 Afterburner.bin 0x8000
Afterburner.partitions.bin
```

Execute **COM.bat.**

The upload to the Afterburner will now take place over the serial connection:



```
C:\Users\ray\OneDrive\Documents\bootload>esptool.exe --chip esp32 --port COM14 --baud 921600 --befo
ter hard_reset write_flash -z --flash_mode dio --flash_freq 80m --flash_size detect 0xe000 boot_app
er_qio_80m.bin 0x10000 Afterburner.bin 0x8000 Afterburner.partitions.bin
esptool.py v2.6-beta1
Serial port COM14
Connecting....
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, Coding Scheme None
MAC: 30:ae:a4:8c:a6:3c
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.0 seconds (effective 1365.3 kbit/s)...
Hash of data verified.
Compressed 17664 bytes to 11528...
Wrote 17664 bytes (11528 compressed) at 0x00001000 in 0.2 seconds (effective 802.9 kbit/s)...
Hash of data verified.
Compressed 1270064 bytes to 712290...
Writing at 0x0001c000... (9 %)
```

Once the download completes, you will need to press (ground) the **EN** switch again to reboot the ESP32.

Note that whilst this procedure is the most tedious to perform, it is the only way to recover a bricked system, in the unlikely event that should ever occur.

Rebuilding from source code

The source code for the Afterburner is released under the GPLv3 Open Source licence.
You are free to make changes for your own personal use or even sell changes, but should you decide to sell your changed code it is also a requirement to likewise share your source code changes when requested under the terms of the GPLv3 licence.

The Git repository for the project is hosted at https://gitlab.com/mrjones.id.au/bluetoothheater.
You can either clone the repository to your own machine, or simply download a zip file.
It is beyond this document to go into the intricacies of Git usage, but it is well worth learning should you like to develop the code further.

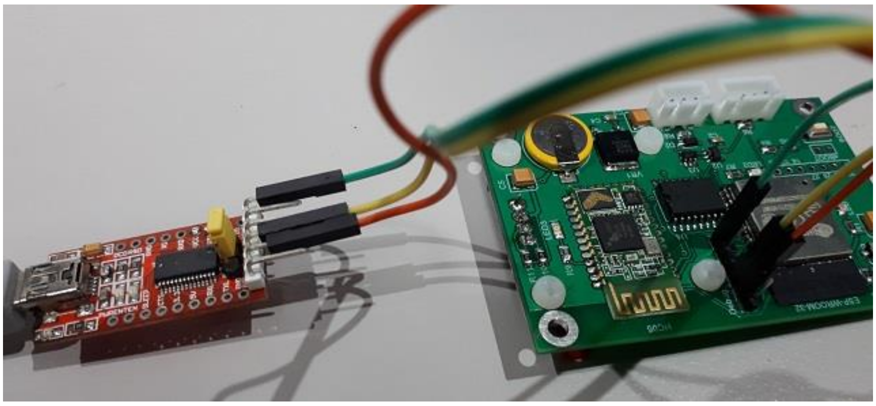
You are also welcome create and "**Push**" your branch back to the above repository incorporating the "next great thing" and subsequently issuing a "**Pull Request**" to have the changes evaluated and potentially merged into the master branch.
You will not be able to push to the GitLab master branch.

Native internal bootloader

The Afterburner PCB does include any bootloader assistance hardware as fitted to the dev boards. Ie. USB to serial device.
The required serial port to perform bootloading of firmware is provided as a 6-pin header
The PCB provides pushbuttons for the **EN** and **BOOT** pins.
The Mk2 PCB also brings the **EN** and **BOOT** pins to a header adjacent to the serial port connection.
The process to prepare for serial boot load is quite simple to deal with once understood.

Required Hardware

New firmware is uploaded to the ESP32 via its debug/programming serial port.
No USB to serial support is provided on the Afterburner PCB.
To provide a USB to serial connection, a **FTDI232 USB adapter** (or similar) is required.
Only the GND, power, Tx & Rx pins should be connected to the Afterburner's debug/programming port.
Note that the ESP32 is a 3.3V processor, so ensure the USB to Serial adapter board is set to 3.3V mode.



Arduino IDE

The Arduino IDE along with the ESP32 tools and required Arduino libraries is used to compile and subsequently upload firmware to the ESP32 module.
Advanced users may choose to use better development environments like Microsoft's VSCode with the Arduino plug in, or PlatformIO.

For the Arduino IDE the general procedure is:

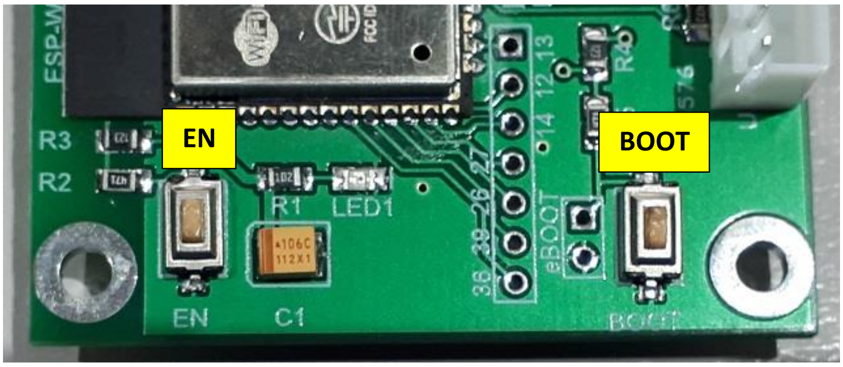
- Load the BTCDieselHeater.ino project into the Arduino IDE.
- Select the correct virtual COM port for the USB to serial adapter.
- Compile and upload the design in the usual manner.

Serial Bootload Process

Prepare the ESP for bootload.

The ESP32 processor needs to be prepared for serial bootloading by using the following sequence:

1. Press and HOLD the **EN** button.
2. Press and HOLD the **BOOT** button.
3. Release the **EN** button.
4. Release the **BOOT** button.

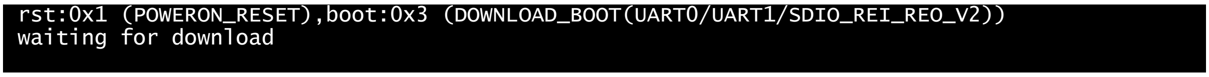


The above sequence can be performed at any time prior to the firmware upload.
It does not need to happen at the moment the upload is being attempted.

When EN is held, power to the ESP32 is removed.
When EN is released and BOOT is held down, the ESP32's internal bootloader will be started.
*The ESP32 will then simply wait in the bootloader until triggered by **ESPtool** during the firmware upload process.*

Note: these switches are also brought out next to the debug port header fields and can instead be used to perform the correct EN & BOOT sequencing.

If a serial debug session is active, the following message should then appear:



```
rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REO_V2))
waiting for download
```

IMPORTANT
Bootloading will fail if a serial debug session remains active.

ESPtool will be unable to use the already occupied serial port.
Ensure the serial debug session is closed before the upload takes place.

Serial Upload

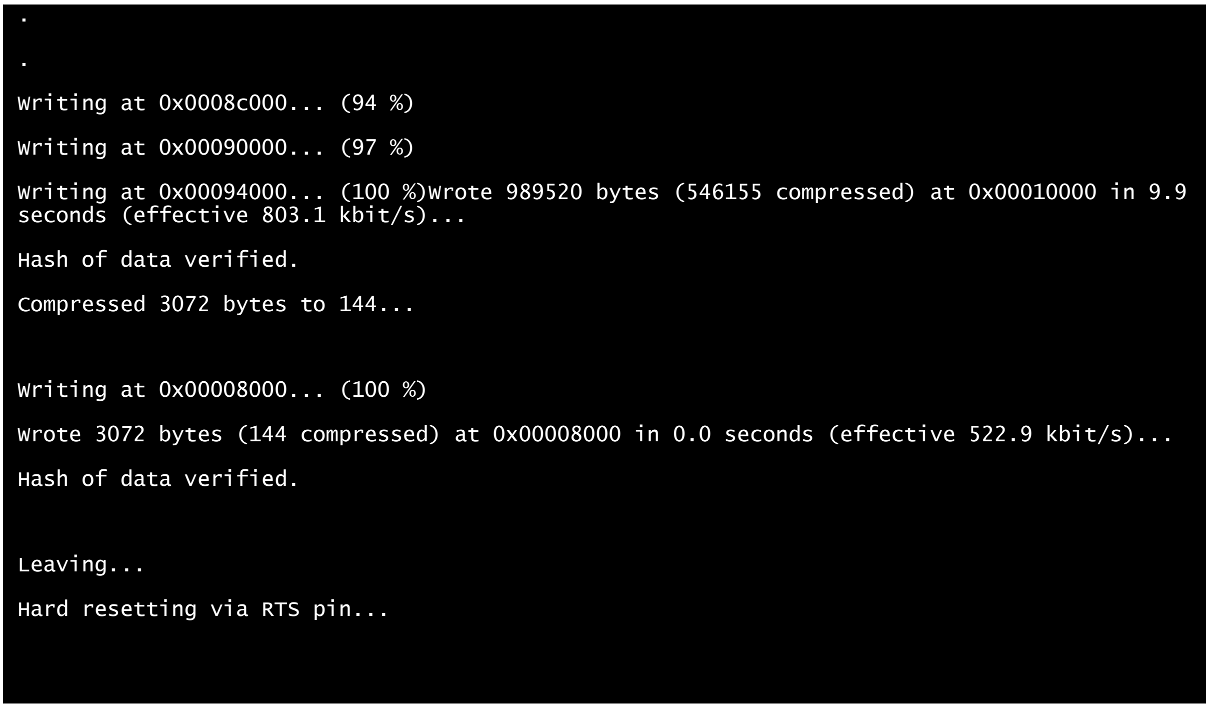
After completion of code compilation, the Arduino environment automatically launches ***ESPtool.exe*** to upload the generated binary file to the ESP32.
An example (automatic) sequence follows:

```
Uploading...
esptool.py v2.5.0
Serial port COM11
Connecting......
Chip is ESP32D0WDQ6 (revision 1)
Features: Wi-Fi , BT, Dual Core
MAC: 30:ae:a4:8c:a6:3c
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...

Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.1 seconds (effective 1092.3 kbit/s)...
Hash of data verified.
Compressed 16720 bytes to 10825...

Writing at 0x00001000... (100 %)
Wrote 16720 bytes (10825 compressed) at 0x00001000 in 0.2 seconds (effective 727.0
kbit/s)...
Hash of data verified.
Compressed 989520 bytes to 546155...

Writing at 0x00010000... (2 %)
Writing at 0x00014000... (5 %)
Writing at 0x00018000... (8 %)
Writing at 0x0001c000... (11 %)
Writing at 0x00020000... (14 %)
Writing at 0x00024000... (17 %)
.
```



```
.

.

Writing at 0x0008c000... (94 %)

Writing at 0x00090000... (97 %)

Writing at 0x00094000... (100 %)Wrote 989520 bytes (546155 compressed) at 0x00010000 in 9.9
seconds (effective 803.1 kbit/s)...

Hash of data verified.

Compressed 3072 bytes to 144...


Writing at 0x00008000... (100 %)

Wrote 3072 bytes (144 compressed) at 0x00008000 in 0.0 seconds (effective 522.9 kbit/s)...

Hash of data verified.


Leaving...

Hard resetting via RTS pin...
```

Completion of serial upload

Despite ***ESPtool*** advising it has rebooted the processor “Hard resetting via RTS pin..”, this did not happen because only the data lines are connected. RTS is not brought through.

Once uploading is completed, YOU finally need to **press and release the EN button** again to reboot the ESP32 processor.
The BOOT switch must not be operated at this time.

OTA (Over The Air) bootload process

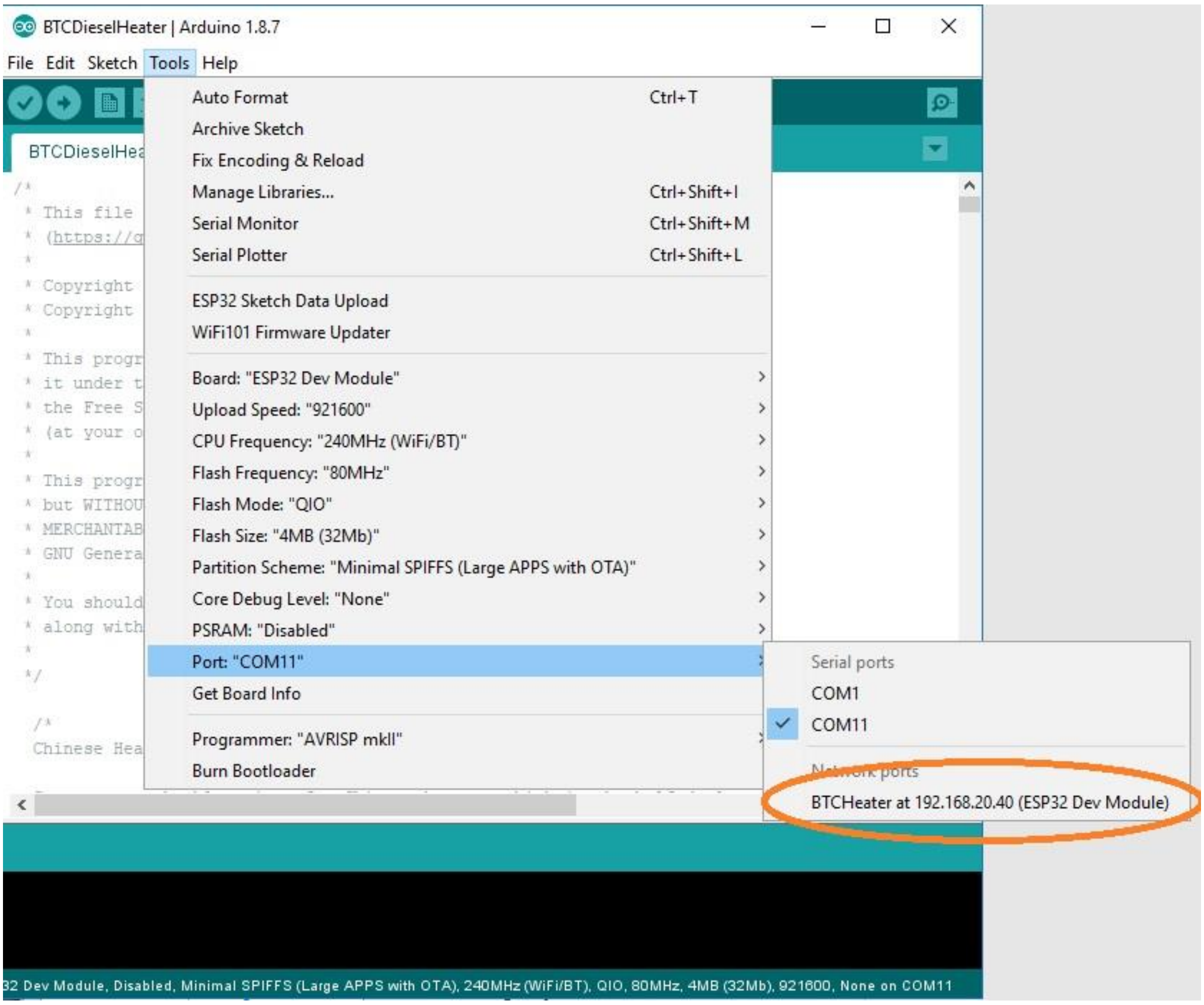
Firmware can also be uploaded to the ESP32 using the OTA (Over The Air) protocol, using the Wi-Fi capabilities of the ESP32.

This requires the Afterburner to be connected to the same Wi-Fi network as the PC running the Arduino software. This achieved by using the Configuration Portal.
You must ensure OTA is enabled on the Afterburner.
The text OTA must appear adjacent to the Wi-Fi icon on the OLED display:

If not, you will also need to use the Wi-Fi settings menu to enable OTA.
This is the same menu as used to enable the Wi-Fi Configuration Portal.

Once prepared, the IP address of the Afterburner on your home network can be discovered via the, Wi-Fi settings menu. It is the STA address.

The Arduino IDE needs to then be configured to use this address instead of a COM port in the menu **Tools… Port…** configuration.
The Arduino IDE is smart enough to automatically discover the OTA port on the network!



Board: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Minimal SPIFFS (Large APPS with OTA)"
Core Debug Level: "None"
PSRAM: "Disabled"
Port: "COM11"
Programmer: "AVRISP mkII"

BTCHeater at 192.168.20.40 (ESP32 Dev Module)

32 Dev Module, Disabled, Minimal SPIFFS (Large APPS with OTA), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None on COM11

SPIFFS upload

The pages served by the web server to control the diesel heater are stored in a special area of flash memory in the ESP32 known as the SPIFFS partition (Serial Peripheral Interface Flash File System).

It is very easy to upload the web pages to SPIFFS.
Like firmware updates, SPIFFS upload works over either OTA or a serial connection using the ESP32's internal bootloader.

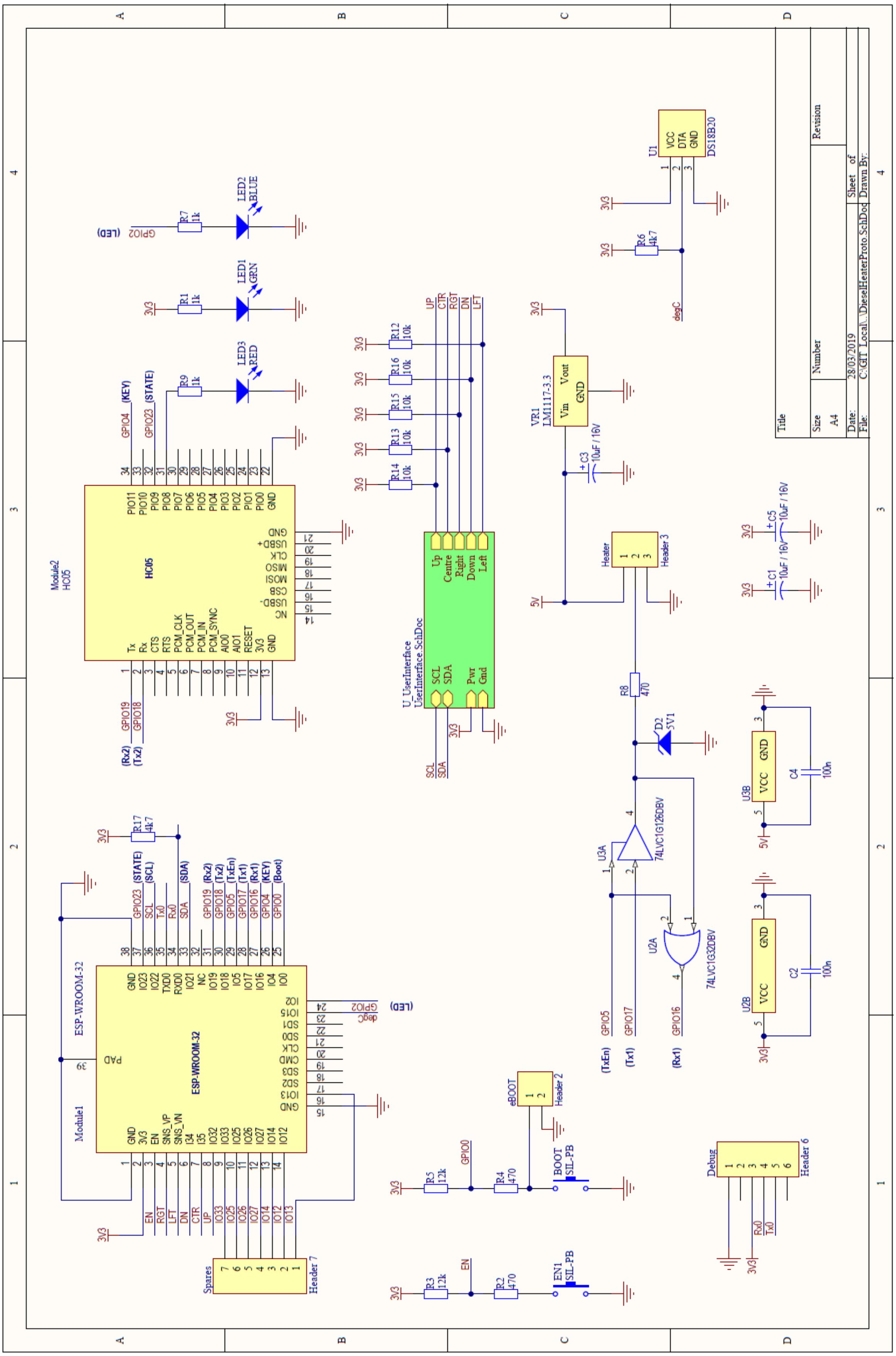
If using serial bootloading, ensure the USB to serial adapter is connected to the Afterburner's debug/programming port, and the ESP32 is ready for bootloading by performing the EN/BOOT button sequence.

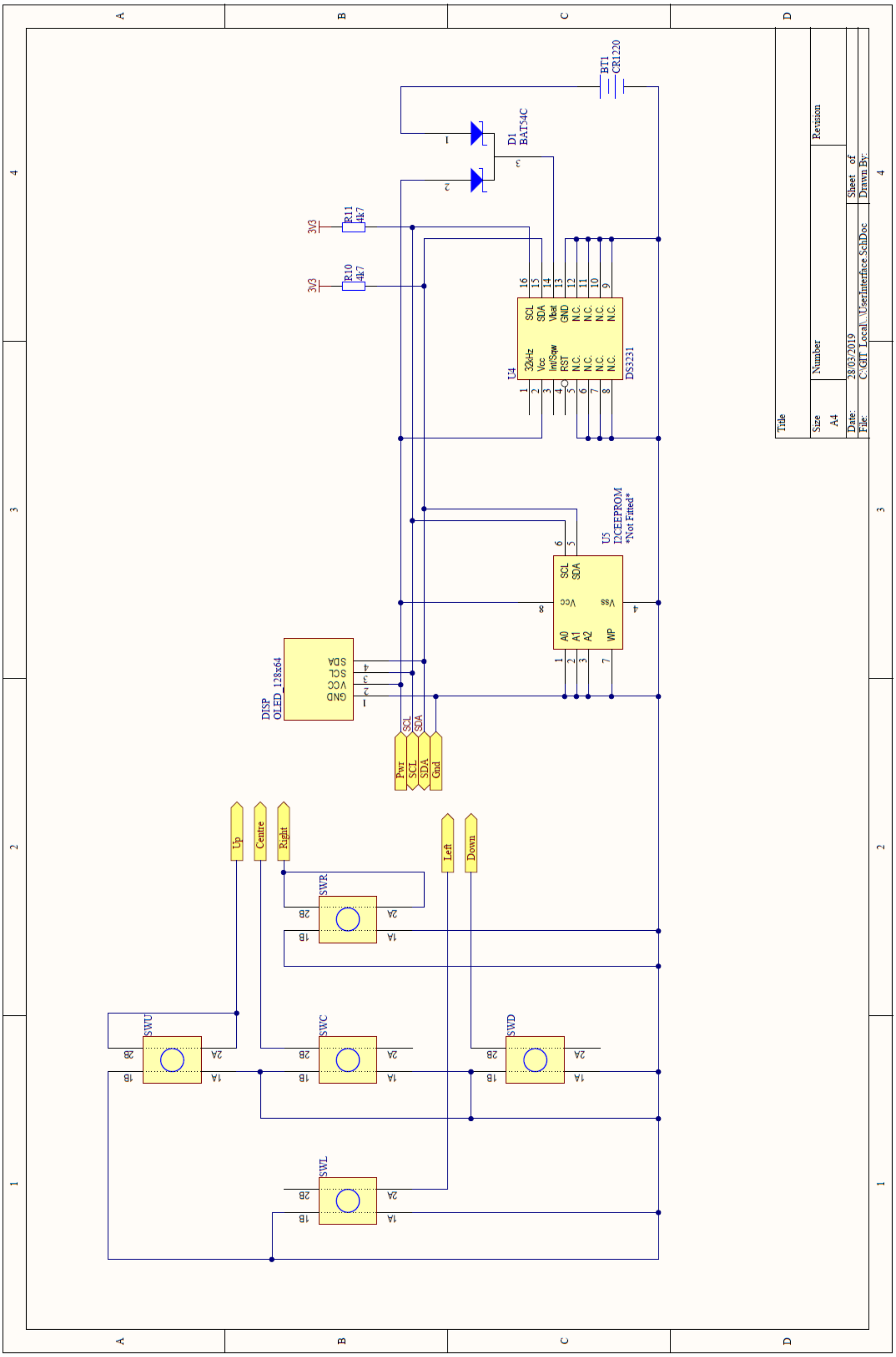
In the Arduino IDE, select **Tools… ESP32 Sketch Data Upload**.

The contents of the `Afterburner/data` directory will then be uploaded to the ESP32's SPIFFS partition.

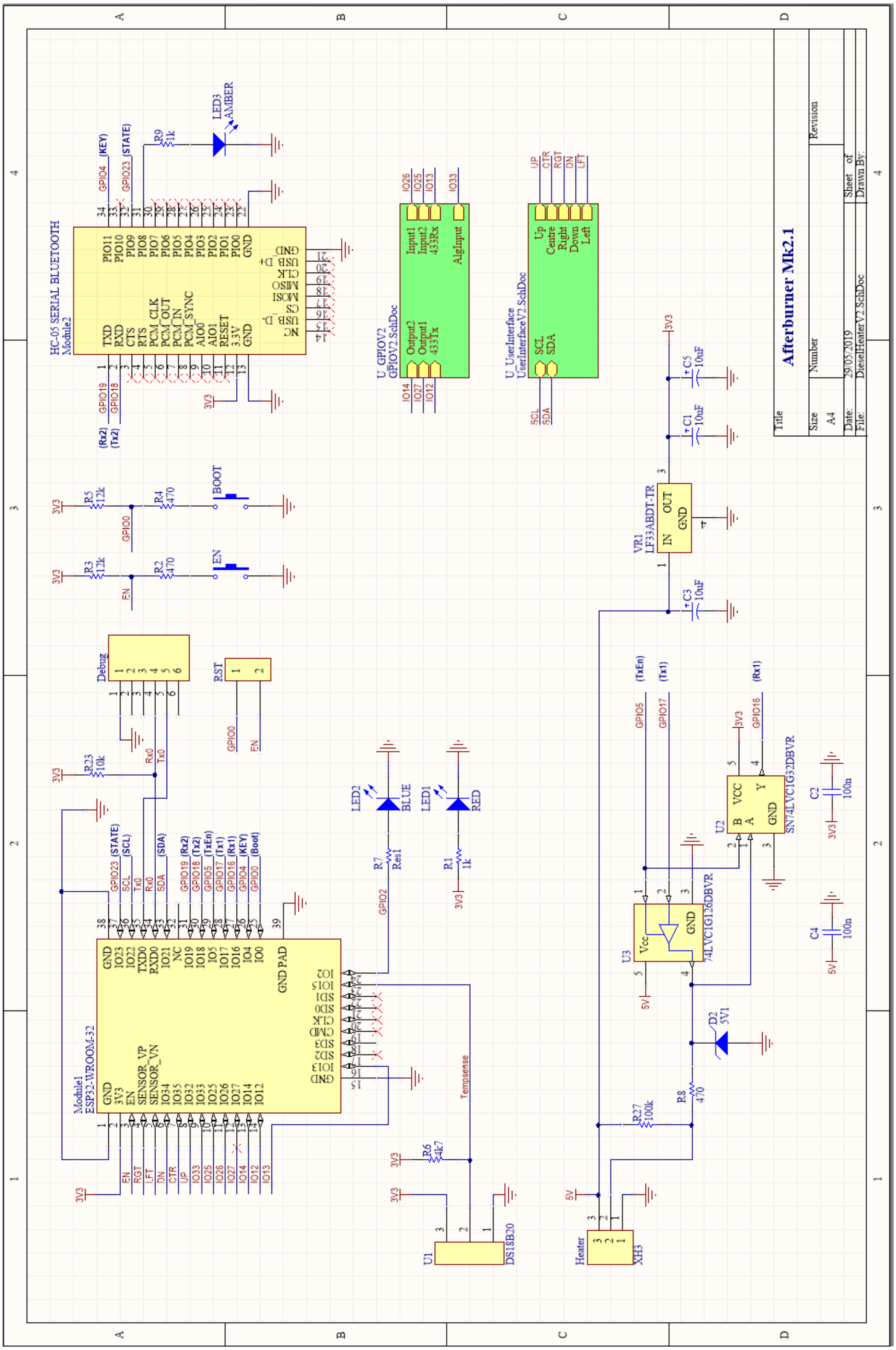
Schematics

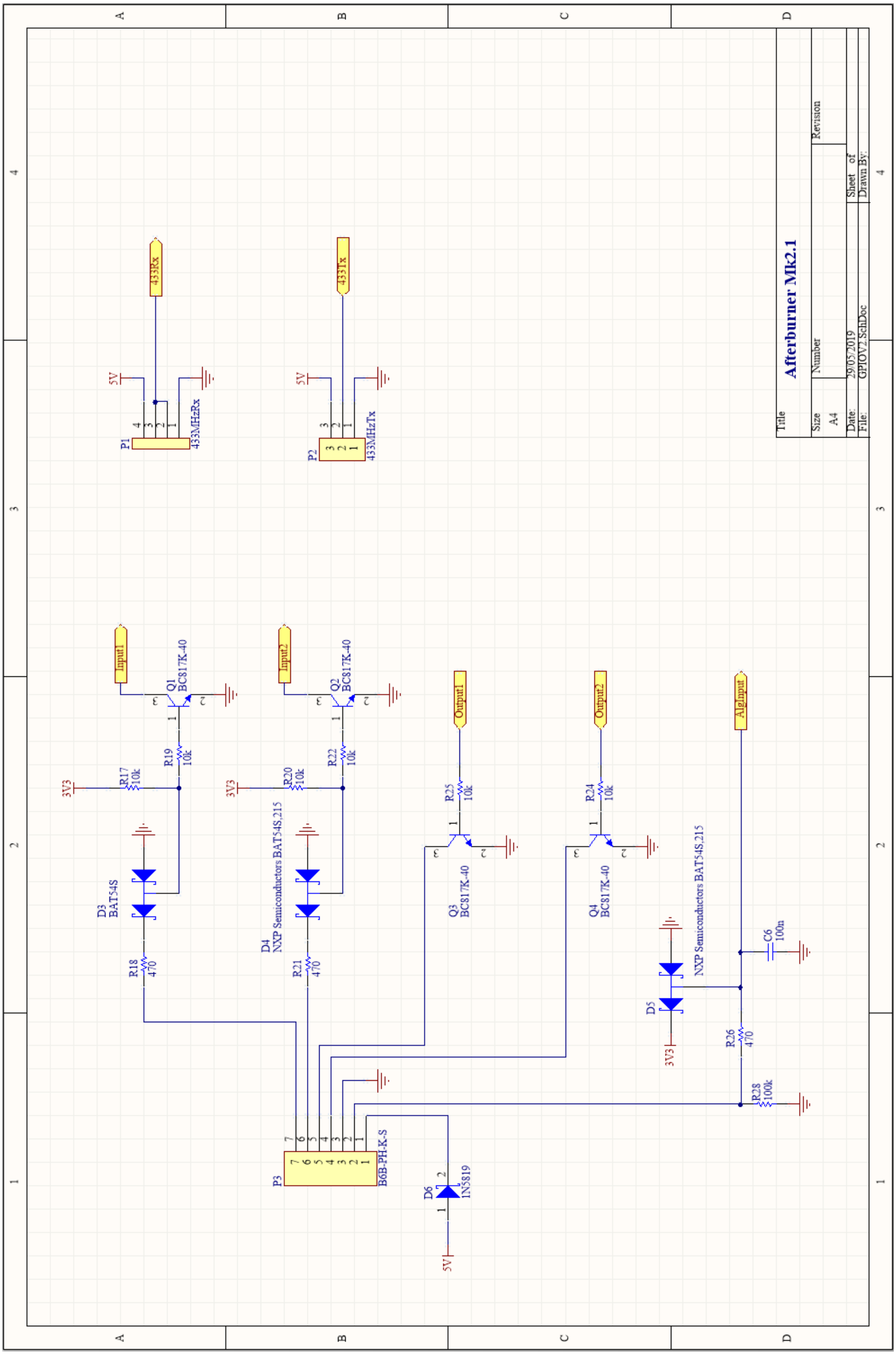
Mk1 PCB

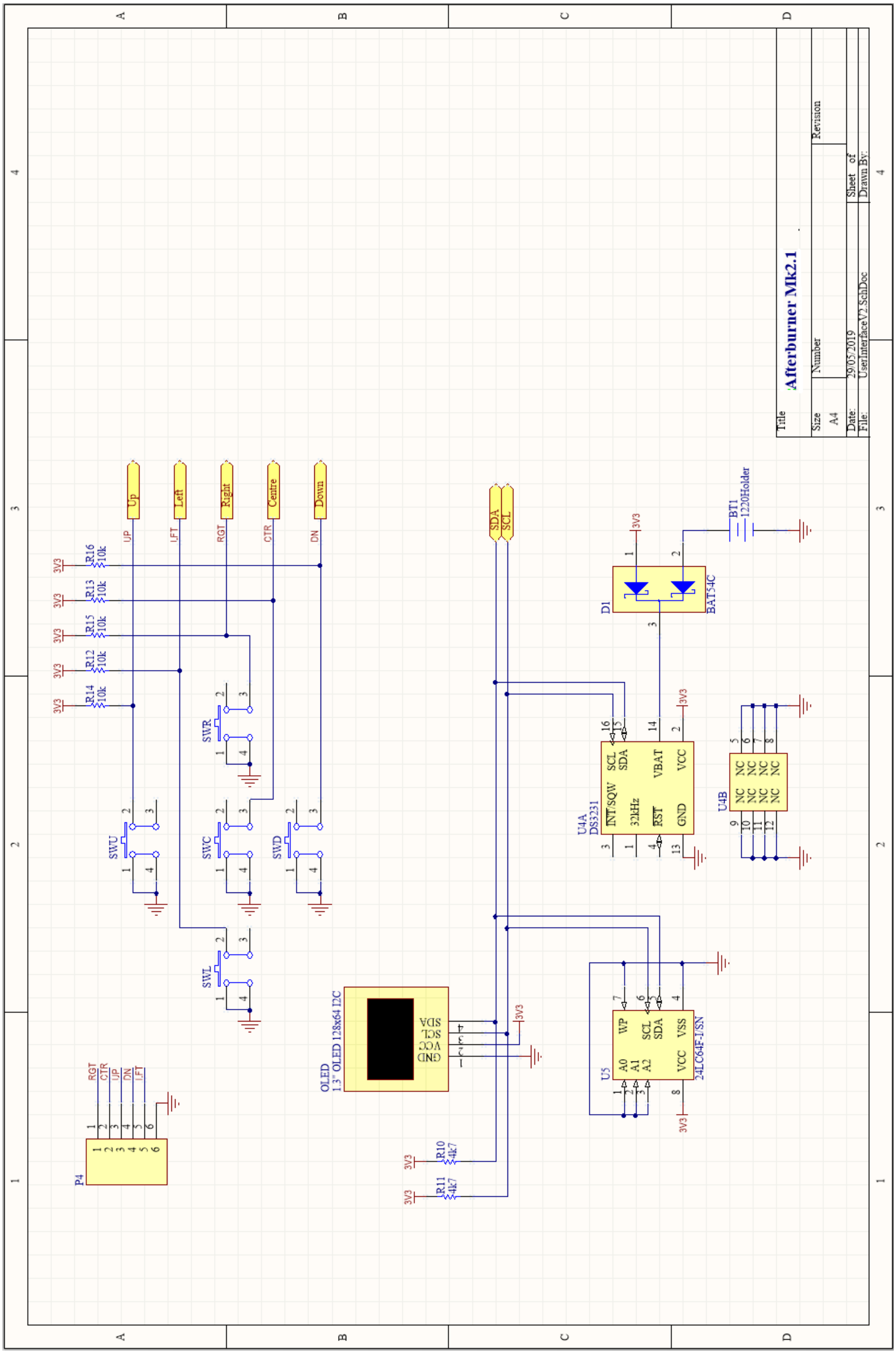

Module1
ESP-WROOM-32
Module2
HC05
U_UserInterface
UserInterface.SchDoc
Up
Centre
Right
Down
Left
SCL
SDA
Pwr
Gnd
VR1
LM1117-3.3
Vin
Vout
GND
U1
DS18B20
VCC
DTA
GND
LED1
GRN
LED2
BLUE
LED3
RED
Heater
Header 3
D2
5V1
U3A
74LVC1G126DBV
U2A
74LVC1G32DBV
U3B
U2B
VCC
GND
C4
100n
C2
100n
C1
10uF / 16V
C5
10uF / 16V
C3
10uF / 16V
eBOOT
Header 2
BOOT
SIL-PB
EN1
SIL-PB
Debug
Header 6
Spares
Header 7
R1 1k
R7 1k
R9 1k
R6 4k7
R17 4k7
R8 470
R2 470
R4 470
R3 12k
R5 12k
R12 10k
R13 10k
R14 10k
R15 10k
R16 10k
3V3
5V
GPIO2
GPIO4
GPIO23
GPIO19
GPIO18
GPIO5
GPIO17
GPIO16
GPIO0
(LED)
(KEY)
(STATE)
(Rx2)
(Tx2)
(TxEn)
(Tx1)
(Rx1)
(Boot)
(SCL)
(SDA)
degC
Title
Size
A4
Number
Revision
Date:
28/03/2019
Sheet of
File:
C:\GIT Local\..\DieselHeaterProto.SchDoc
Drawn By:


BT1
CR1220
D1
BAT34C
1
2
3
R11
4k7
R10
4k7
3V3
3V3
U4
DS3231
32kHz
Vcc
Int/Sqw
RST
N.C.
SCL
SDA
Vbat
GND
U5
I2CEEPROM
Not Fitted
SCL
SDA
Vcc
Vss
A0
A1
A2
WP
DISP
OLED_128x64
GND
VCC
SCL
SDA
SCL
SDA
Pwr
SCL
SDA
Gnd
Up
Centre
Right
Left
Down
SWR
SWU
SWC
SWD
SWL
1A
1B
2A
2B
A
B
C
D
1
2
3
4
Title
Size
A4
Number
Revision
Date:
28/03/2019
Sheet of
File:
C:\GIT Local\..\UserInterface.SchDoc
Drawn By:

Mk2 PCB


Afterburner Mk2.1
Title
Size A4
Number
Revision
Date: 29/05/2019
File: DieselHeaterV2.SchDoc
Sheet of
Drawn By:
HC-05 SERIAL BLUETOOTH
Module2
LED3 AMBER
R9 1k
U_GPIOV2
GPIOV2.SchDoc
Input1 Input2 433Rx AlgInput
Output2 Output1 433Tx
IO26 IO25 IO13 IO33
IO14 IO27 IO12
U_UserInterface
UserInterfaceV2.SchDoc
Up Centre Right Down Left
SCL SDA
UP CTR RGT DN LFT
VR1
LF33ABDT-TR
IN OUT GND
C1 10uF
C3 10uF
C5 10uF
3V3
BOOT
EN
R5 12k
R4 470
R3 12k
R2 470
GPIO0
Debug
RST
R23 10k
Rx0 Tx0
LED2 BLUE
LED1 RED
R7 Res1
R1 1k
GPIO2
Module1
ESP32-WROOM-32
GND PAD
Tempsense
R6 4k7
U1
DS18B20
U2
SN74LVC1G32DBVR
VCC B A GND Y
U3
74LVC1G126DBVR
Vcc GND
C2 100n
C4 100n
5V
D2 5V1
R8 470
R27 100k
Heater
XH3
GPIO5 (TxEn)
GPIO17 (Tx1)
GPIO16 (Rx1)
GPIO23 (STATE)
GPIO4 (KEY)
GPIO19 (Rx2)
GPIO18 (Tx2)
GPIO0 (Boot)
A B C D
1 2 3 4


433Rx
5V
P1
4 3 2 1
433MHzRx
433Tx
5V
P2
3 2 1
433MHzTx
Input1
Q1
BC817K-40
R17
10k
R19
10k
3V3
D3
BAT54S
R18
470
Input2
Q2
BC817K-40
R20
10k
R22
10k
3V3
D4
NXP Semiconductors BAT54S,215
R21
470
Output1
R25
10k
Q3
BC817K-40
Output2
R24
10k
Q4
BC817K-40
AlgInput
NXP Semiconductors BAT54S,215
D5
C6
100n
3V3
R26
470
R28
100k
P3
7 6 5 4 3 2 1
B6B-PH-K-S
D6
1N5819
5V
Title
Afterburner Mk2.1
Size
A4
Number
Revision
Date:
29/05/2019
Sheet of
File:
GPIOV2.SchDoc
Drawn By:
A
B
C
D
1
2
3
4


Title
Afterburner Mk2.1
Size
A4
Number
Revision
Date: 29/03/2019
Sheet of
File: UserInterfaceV2.SchDoc
Drawn By:
3V3
R14 10k
R12 10k
R15 10k
R13 10k
R16 10k
UP
LFT
RGT
CTR
DN
Up
Left
Right
Centre
Down
SWU
SWC
SWD
SWR
SWL
P4
RGT
CTR
UP
DN
LFT
OLED
1.3" OLED 128x64 I2C
SDA
SCL
VCC
GND
R10 4k7
R11 4k7
SDA
SCL
D1
BAT54C
BT1
1220Holder
U4A
DS3231
INT/SQW
32kHz
RST
GND
SCL
SDA
VBAT
VCC
U4B
NC
U5
24LC64F-I/SN
A0
A1
A2
VCC
WP
SCL
SDA
VSS
A
B
C
D
1
2
3
4