

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.

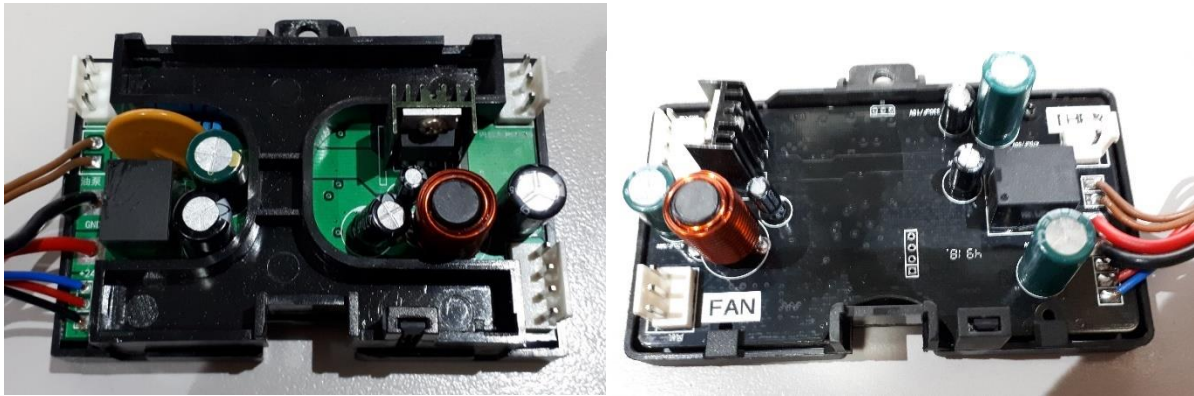
You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

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

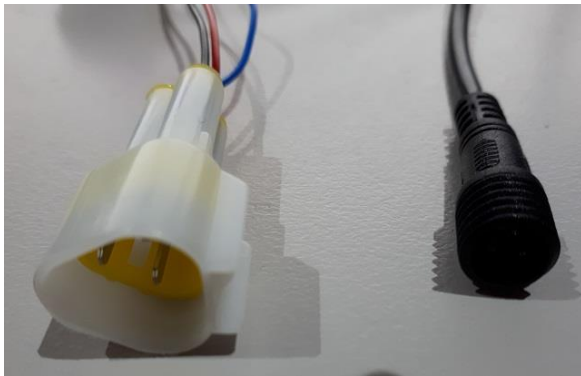
COMPATIBILITY

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 both 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:



If your existing controller looks like any of the following, it is compatible.



Contents

PCB overview	6
User interface.....	10
OLED Menu Structure	11
Menu Navigation.....	11
Inactivity timeout.....	11
Basic Control menu	12
Heater On / Off	12
Temperature adjustment.....	13
Thermostat or Fixed power selection	13
Fault reporting	13
Detailed Control menu.....	14
Starting the heater	14
Thermostat or Fixed demand modes.....	15
Stopping the heater	15
Clock menu.....	17
Set Time	17
Applying the new Time and Date.....	17
Returning to parent menu (Clock)	17
Timers	18
Selecting and setting fields	18
Enabling or Disabling the timer.....	19
Timer Repeat or One Shot	19
Saving the timer	19
Returning to Root Menu Loop (Clock menu).....	20
Mode Selection menu.....	21
Wi-Fi Mode menu	22
IP addresses	22
Web Server.....	22
Selecting Wi-Fi operation mode.....	22
Enabling / Disabling “Over The Air” (OTA) firmware updates.	23
MAC address discovery	23
Heater Settings menu	24
Fuel Mixture edit menu	24
Heater Settings edit menu	25
Advanced Thermostat modes.....	26

Afterburner user manual

Dead-band mode - modified hysteresis.....	27
Linear Hz.....	27
Settings Adjustment.....	28
Bluetooth Connectivity	29
Bluetooth pairing	36
Android Application	36
Using the Bluetooth Application.....	40
Menus	41
Detailed Status & Control mode	42
Timers mode	43
Heater Settings mode	44
Arduino Environment.....	45
ESP32 build settings.....	50
Arduino Board Configuration.....	50
Minimal SPIFFS Partition Scheme	50
Required libraries for Arduino environment.	51
Firmware updates.....	52
Introduction	58
Required Hardware.....	58
Arduino IDE	59
Bootload Process	59
Prepare the ESP for bootload.	59
Upload.....	60
Completion of upload	60
Schematic.....	63

Introduction

The Afterburner is an open source software design, using 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.
2. Due to popular demand a new Mk2 PCB was designed to fit within an off the shelf ABS case. The solder mask of this PCB is red.

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

The primary difference of the Mk2 PCB is additional circuitry was added to provide buffered input and output signals for the purposes of 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

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

Mk1 PCB –software and hardware expansion

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.

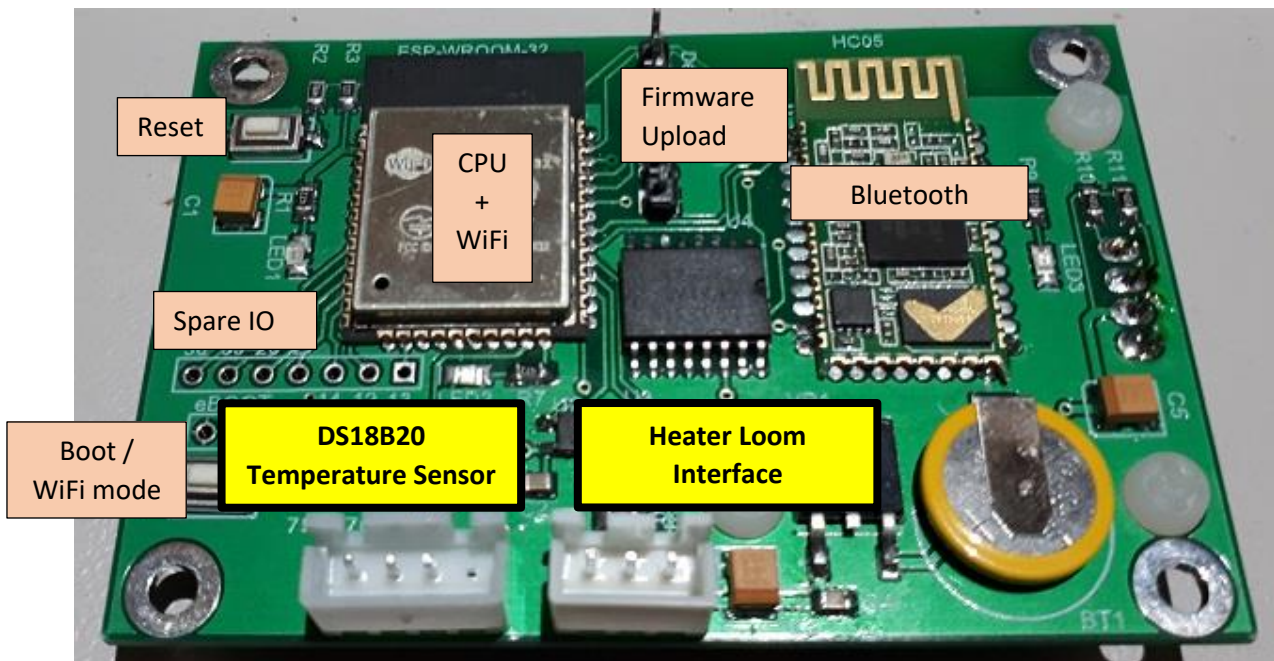
These uncommitted pins allow currently unsupported hardware to potentially be added, eg a GSM data module.

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.

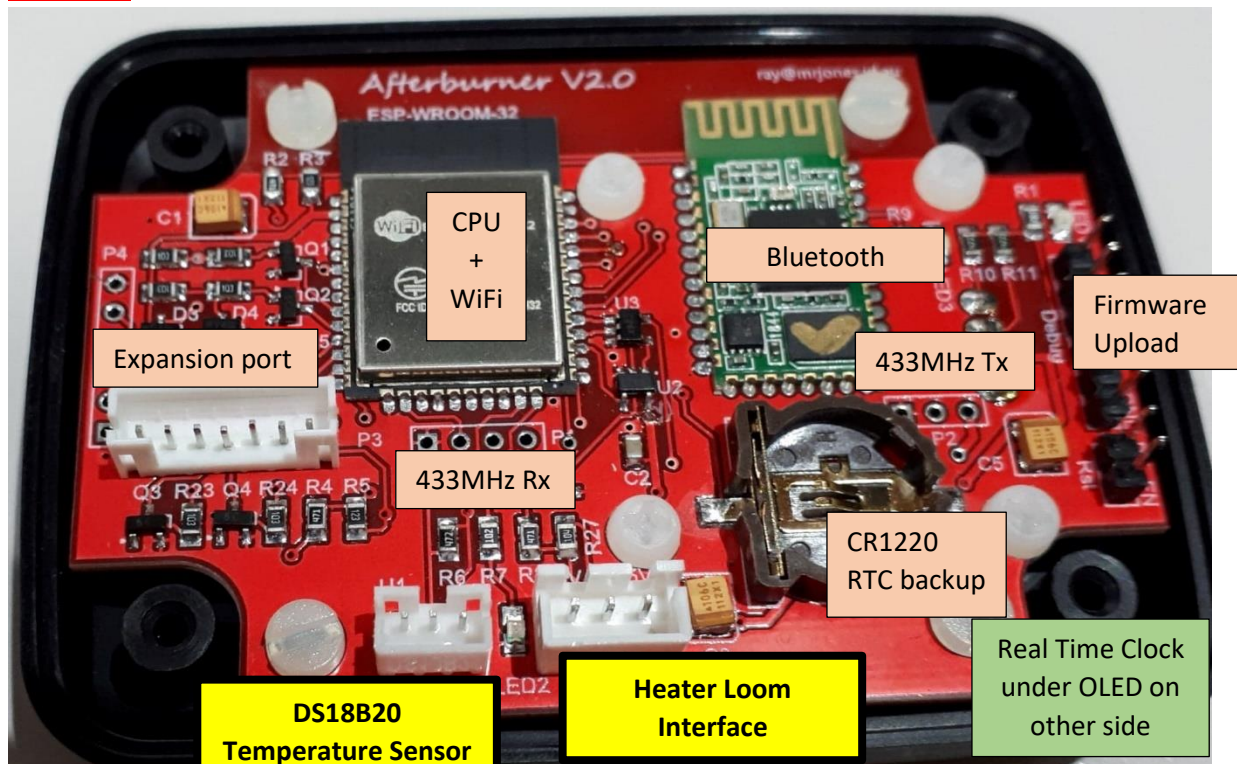
Further information can be found in the ESP32 technical guides by Espressif.

PCB overview

Mk1 PCB



Mk2 PCB



Both PCBs have identical core componentry, just the physical location has been shifted to accommodate the case standoffs in the Mk2 PCB.

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

Afterburner user manual

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 used for initial installation to learn the settings from the previous controller.



The controller 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 controller termination is:

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

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



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!



Note:

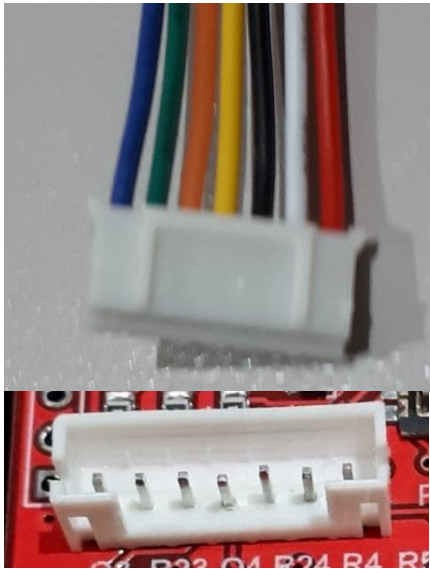
The silk screen overlay on the initial production of Afterburner PCBs erroneously labelled the first two pins, the correct numbering is the highlighted numbers in the image above.

The functionality of the circuit is not affected.

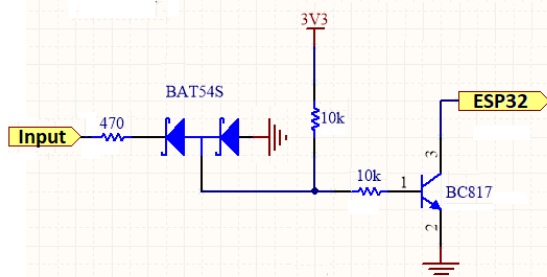
Mk2 Expansion port

A 7-pin JST PH header is provided to allow

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

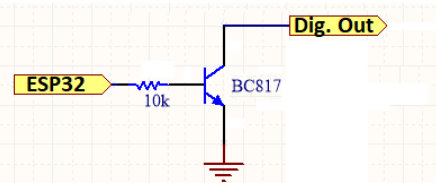


<u>Wire Colour</u>	<u>Purpose</u>
Blue	Digital input #1
Green	Digital input #2
Orange	Digital output #1
Yellow	Digital output #2
Black	Ground, 0V
White	Analogue input
Red	Power, +5V



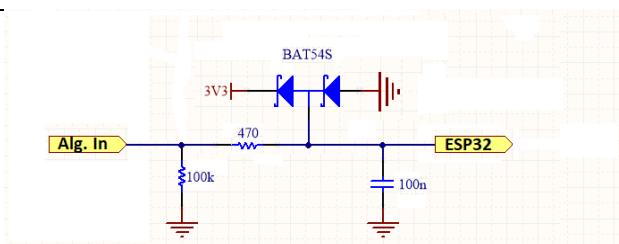
Diode array:
Blocks +ve external voltages.
Shunts -ve external voltages to 0V.

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.

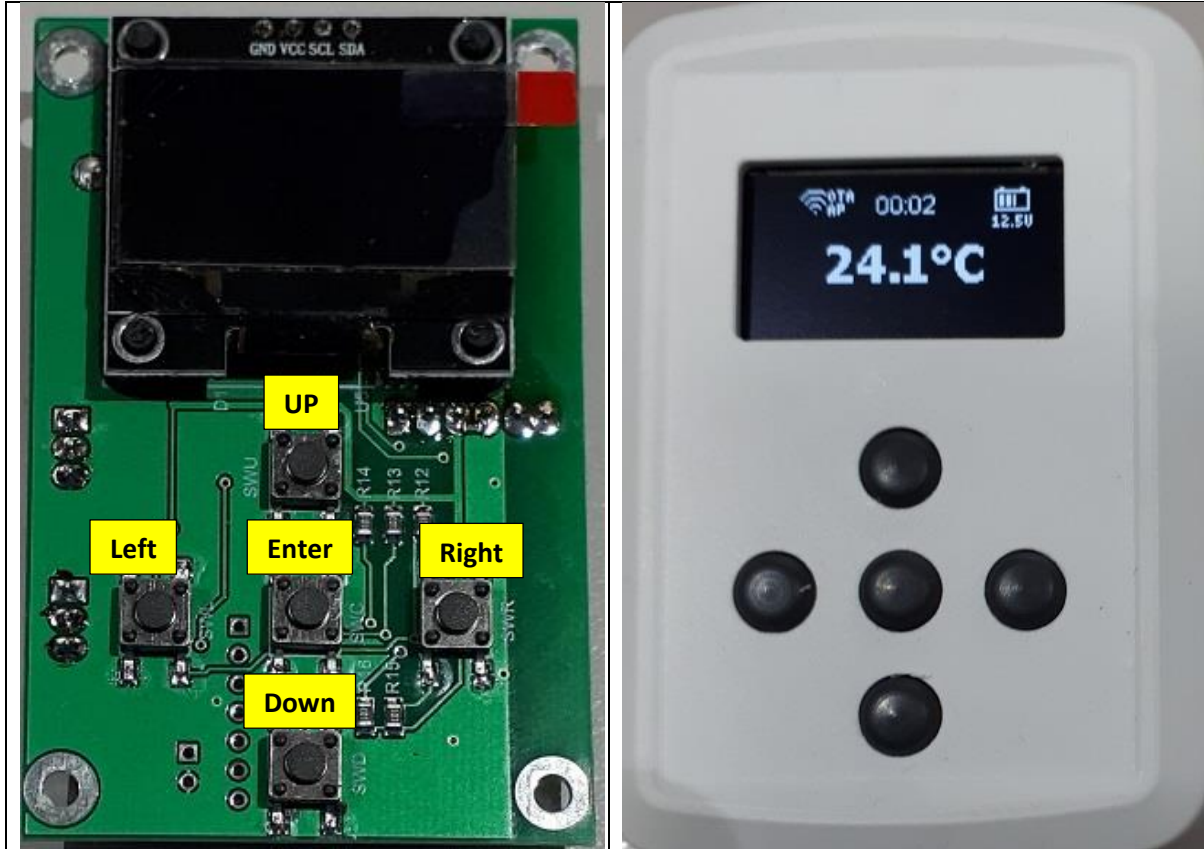
Open collector can drive a relay coil (back
EMF protection required) or LED.
Either powered from 5V or external power
source. (0V commoned)



Diode array clamps to 3.3V or 0V
100k pull down only exists on V2.1 PCB (not
in production)
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.

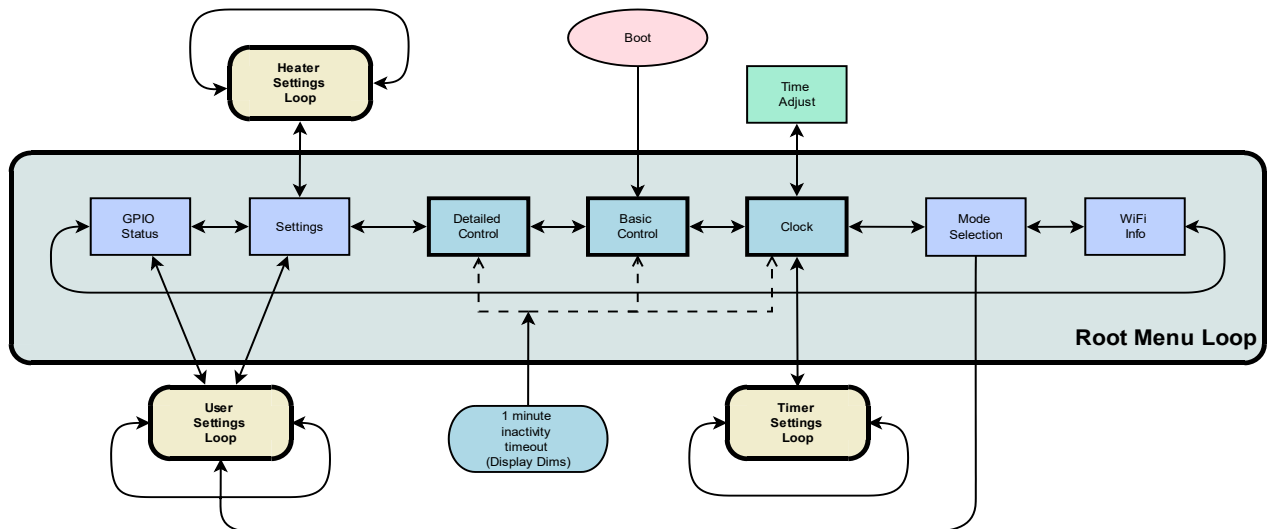
OLED Menu Structure

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 several menus used to monitor/control the heater, or alter settings such as fuel mixture, timers etc.

The general topology of the menu flow is as follows:



When first powered, the initial menu is always the Basic Control menu. This menu provides basic control of the heater and will be described later.

Menu Navigation

The Basic Control menu lies within the Root Menu Loop.

Each screen in the loop can be navigated to by using the **LEFT** or **RIGHT** keys.

From some screens, navigation can move away from the Root Menu Loop, and enter a different loop.

e.g.: pressing **DOWN** from the **Clock** menu will shift control to the Timer Settings Loop. Pressing **LEFT** and **RIGHT** now will select the available menus within that loop.

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

Inactivity timeout

By default, if keypad has been idle for a minute, the display is dimmed.

Additionally if the currently selected menu is not associated with one of the core root menus (**Detailed Control**, **Basic Control**, or **Clock**), menu selection will revert to a core root menu.

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 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

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.



Once fully ignited and running, this 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.



During this time, the heater will stop the fuel delivery and reheat the glow plug to 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 (*about 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 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.



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.



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



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.



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:



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

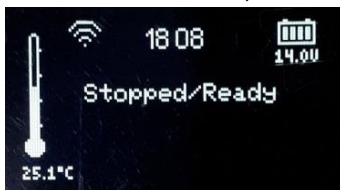
The desired temperature, or fixed heat demand can be altered by pressing the **UP** / **DOWN** keys.

When in thermostat mode, the desired value is placed as a small triangle to the left of the measured "temperature bulb" on the left-hand side of the screen.

The actual temperature is shown below the bulb.

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

With the heater idle, the menu will look something like below:



Heater On / Off

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

Starting the heater

Hold the **CENTRE** key for a second or so, the heater should start and this is reflected in the status message:



Shortly thereafter, the fan will begin spinning, and the glow plug will be gently raised in temperature as can be observed by the steady rise in power applied to it.

Note that the fan icon rotates to indicate the fan is running, along with the measured RPM!



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:



Afterburner user manual

The pump starts at a slow speed and steadily rises, along with fan speed to get the heater ignited. Ignition tends to be sensed once the body temperature of the heat exchanger has risen by about 5-10°C.

The heat exchanger temperature is reported in the right hand side "thermometer bulb", and the actual value is shown below.

Once the body temperature has risen above 65°C or so, the glow plug is shut down, all the while the fuel pump and fan speeds are progressively increasing until at full speed.

The heater runs in this state for quite a few minutes getting the internals nice and hot:



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:



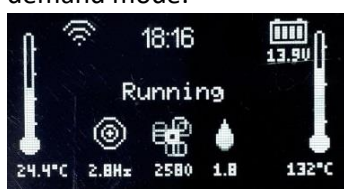
Thermostat or Fixed demand modes

Hold the **DOWN** key 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 indicator also disappears from the left-hand thermometer when in fixed demand mode:



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 gently 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



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 if the power is out by an integrated lithium coin cell.

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.

Set Time

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



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

The day field will first be surrounded by a loop.



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.



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 be 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.

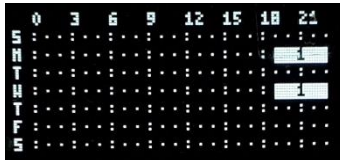
To set the time, you must press **CENTRE** when the SET field is selected.

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 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 Wednesday, starting around 19:00, stopping around 23:00.

The bar being solid indicates this timer repeats.

The numeric within the bar shows which timer definition applies.

There are 14 timers available.

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.



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

To start editing the timer, press the **UP** button.



- **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:



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:



When “per day mode” is enabled, the selection loop shrinks to only select one day.

- **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.

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.



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:



Returning to Root Menu Loop (Clock menu)

When the navigation line is selected, press the **CENTRE** key to return to the Main 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 Main Menu Loop, if any timers have been set, the next timer that is scheduled to run will appear at the top of the screen, the embedded numeric being the ID of the specific timer that will next operate:



If the timer is set to repeat, the icon will gain an arrow looping about the icon. The number within the icon indicates which timer is defined, not how many times it will repeat.

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.

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

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



If the selection reaches the top line, fuel pump priming becomes possible.



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

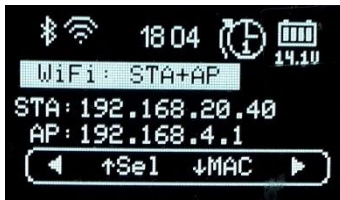
Enable the fuel pump by pressing **RIGHT**, the current pump speed is then reported.

Stop the fuel pump by pressing **UP**, **DOWN** or **LEFT**.

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.



Wi-Fi Mode menu



The Wi-Fi mode 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**, or **STA+AP** 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 to port 80 (default HTTP port):

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

Therefore, there are 4 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.

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 stored credentials exist 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** or **CFG STA+AP**

To switch to the selected Wi-Fi mode, press the **CENTRE** key.

The new Wi-Fi settings 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 uses the embedded ESP32 processor’s debug serial port.

For convenience, firmware can also be updated via the Wi-Fi Interface if the OTA feature is enabled.

By default, OTA mode is enabled, but in the interests of security it is preferable to only enable OTA mode when new firmware is to be uploaded.

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:



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 configure a fixed IP address on another network for the Afterburner controller.



Heater Settings menu



The heater settings menu is the gateway to being able to edit the heater's tuning settings. It also enables access to the User Settings menu loop.

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:



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

Only when "1688" is displayed, press the **CENTRE** key.

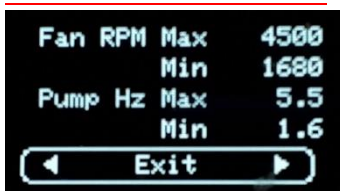
If any other value is shown, control returns to the Root Menu Loop's Heater Settings menu.

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

Heater Settings Menu Loop

The first menu is the Fuel Mixture settings menu:

Fuel Mixture edit menu



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:



Heater Settings edit menu

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



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**.

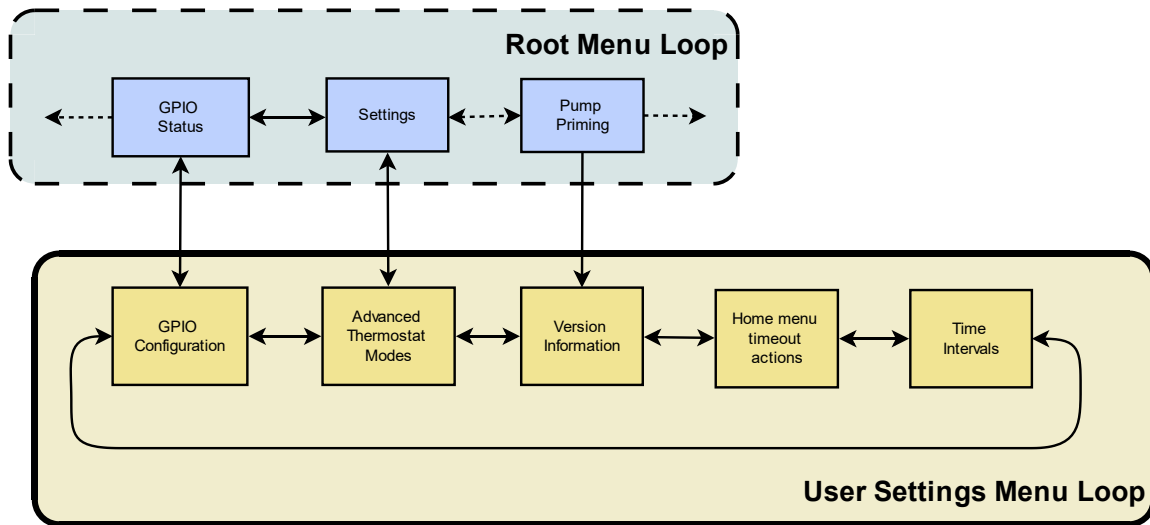
To set, save and apply the settings, the **CENTRE** key must be pressed whilst one of the adjustable settings is selected.

You must then confirm the changes by pressing the **UP** key.

STORING will then appear as the settings are saved to non-volatile memory:



User Settings Menu loop



The User Settings menu loop can be accessed via either the Heater Settings, GPIO status or Pump Priming menus of the Root Menu loop.

Pressing down upon any of those root menus will take you to the User Settings loop, either direct to the Advanced Thermostat modes, GPIO Configuration or Version Information menus.

Advanced Thermostat modes Menu (User Settings menu loop)

The standard thermostat employed by the heater uses the set point defined by the controller and sent over the communications protocol.

Standard Thermostat mode

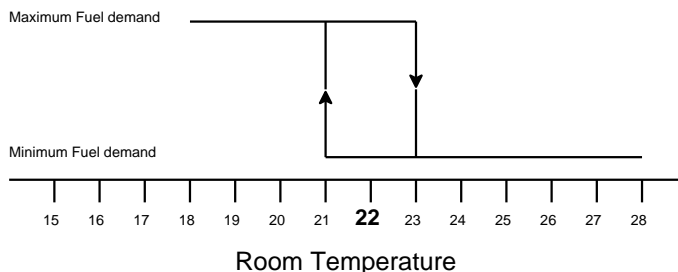
Starting from cold, the heater will run at the maximum fuel rate until the room temperature exceeds the setpoint by 1°C.

Once this threshold is crossed, the heater then runs at the minimum fuel rate.

When the temperature drops 1°C below the setpoint, the heater once again runs at the maximum fuel rate.

i.e. a 2 degree hysteresis.

Standard thermostat mode graphic:



The Afterburner can modify this behaviour in one of two ways:

1. Modify the actual temperature sent to the heater, controlling the width of the dead-band.
2. Operating the heater in a linear Fixed Hz mode.

Dead-band mode - modified hysteresis

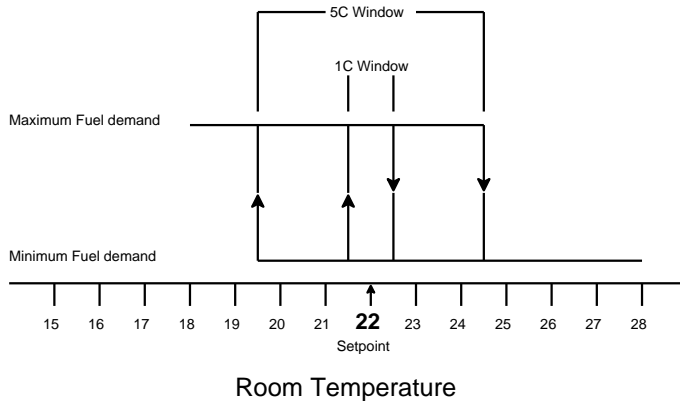
Dead-band mode still uses the heater's internal thermostat algorithm, but the Afterburner modifies the room temperature value that is communicated to the heater.

If the current room temperature lies within a user definable window about the setpoint, the setpoint temperature is sent to the heater.

If outside the window, the temperature sent will be at least 1 degree higher or lower than the setpoint.

Thereby the dead-band, or hysteresis of the thermostat is controlled.

Dead-band thermostat mode graphic:



Linear Hz

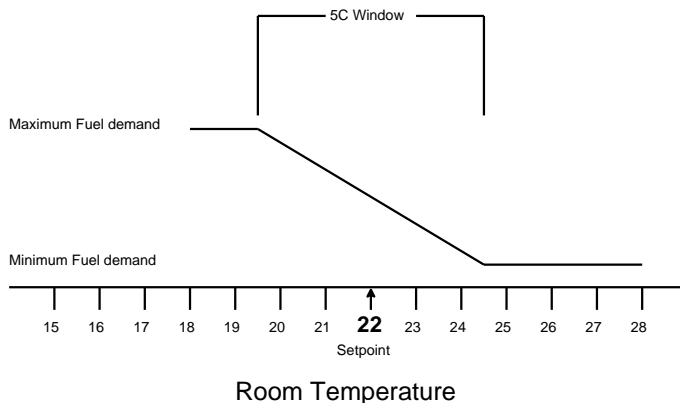
Linear Hz mode is a cunning method to remove the thermostat function entirely from the heater. The heater instead is told to run using Fixed Hz, or "heat demand" mode.

If the room temperature is below the limits of the user defined window about the setpoint, the maximum fuel rate is requested by the Afterburner.

If the room temperature is above the window's limit, the 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.

Linear Hz thermostat mode graphic:



Cyclic Thermostat (override)

The cyclic thermostat mode can be used to shut the heater down when a maximum temperature threshold is exceeded, and automatically turn the heater back on when the temperature falls below a minimum temperature threshold.

This will cause the glow plug to be cycled during both shutdown and restart, so consideration should be given to this higher electrical energy load when using a battery based system.

Thermostat settings adjustment

From the Root Loop Heater Settings menu, press the **DOWN** key.

This presents the Thermostat Mode menu:



Press **UP** to enter edit mode.

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

The available options are indicated against the graphic icons:

- **Spanner:** Thermostat algorithm selection (*Standard, Dead-band, Linear Hz*)
- **Up/Down arrows:** Thermostat window size (*0.2°C – 6.3°C range*)
- **Pause/Start:** Cyclic thermostat mode thresholds (*auto 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:



Version Information *(User Settings menu loop)*

The version information menu allows the current installed software version to be inspected.

From this menu you can also:

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



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

The third line shows the PCB being used.

Mk1 units will be shown as V1.0.

Mk2 units will be shown as V2.1 (*modifications are actually required to the native V2.0 PCB to enable the ADC input, a future V2.1 PCB will incorporate these changes*)

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 WiFi is connected to an Internet enabled network, when the version menu is opened a check is made for a possible firmware update being available from the Afterburner web server; <http://www.mrjones.id.au/afterburner> .

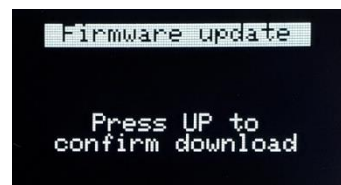
If there is a new update, a download animation from the world to the device is shown to the left of the software icon.

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:



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

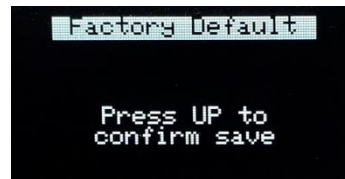
After 2 UP presses:



After 3rd UP press:



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



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



Home Menu Actions (User Settings menu loop)

The Home Menu option menu allows the user to customise which default Root home 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 if 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:



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.



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.

GPIO configuration (User Settings menu loop)

The Mk2 PCB specifically provides extra input and output conditioning circuitry to allow the ready addition of external switches or status indicators/relays.

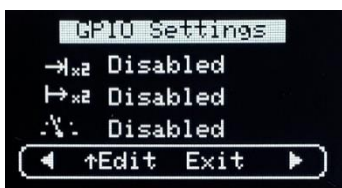
The Mk1 PCB can still provide these GPIO functions, but care needs to be taken to protect the fragile inputs of the ESP32 processor. (eg 3.3V logic, intolerant to errant voltages).

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.

This is the reason why 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:



The 1st line shows the mode of the 2 digital inputs.

The 2nd line shows the mode of the 2 digital outputs.

The 3rd line shows the analogue input mode.

Digital Inputs

The 2 digital inputs can be configured as:

<ul style="list-style-type: none"> • Disabled 	
<ul style="list-style-type: none"> • Input 1 ON, Input 2 OFF 	<p>A momentary close on input 1 will start the heater. A momentary close on input 2 will stop the heater</p>
<ul style="list-style-type: none"> • Input 1 ON (when active) Input 2 External Thermostat (to be implemented) 	<p>Holding Input 1 closed will run the heater, opening will stop the heater. Input 2 can be connected to a room thermostat. When closed the heater will be started. When opened the heater will be shutdown.</p>
<ul style="list-style-type: none"> • Input 1 momentary ON/OFF 	<p>If stopped, a momentary close on input 1 will start the heater, a subsequent momentary close on input 1 will then stop the heater.</p>

Digital Outputs

The 2 digital outputs can be configured as:

<ul style="list-style-type: none"> • Disabled 	
<ul style="list-style-type: none"> • LED Status Indicator 	<p>Standby – OFF Heater starting – “Breathe up” illumination Heater running – Steady illumination Heater starting – “Breathe down” illumination</p>
<ul style="list-style-type: none"> • User controlled 	<p>Can be connected to enable user defined circuitry</p>

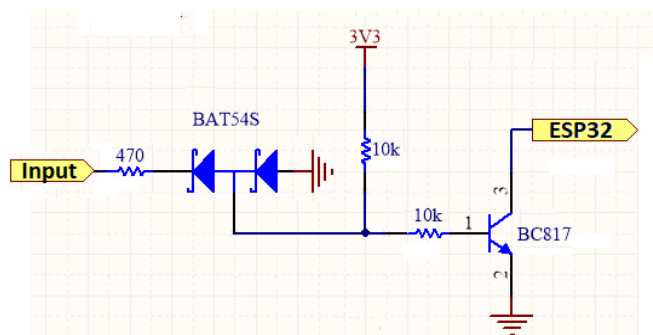
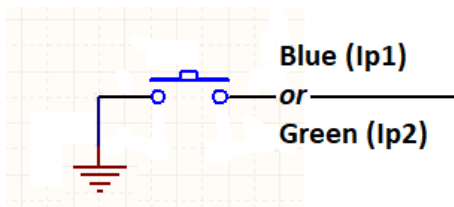
Analogue Input

The single analogue input can be configured as:

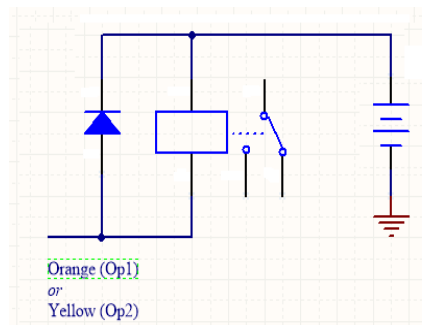
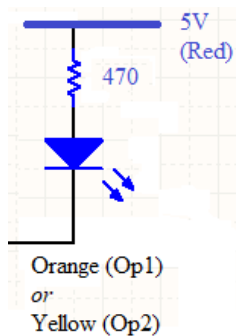
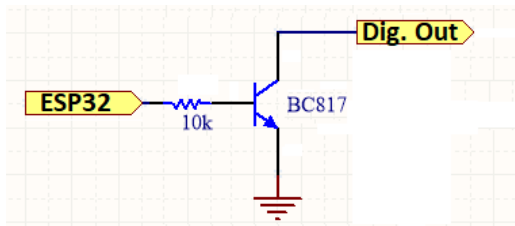
<ul style="list-style-type: none"> • Disabled 	
<ul style="list-style-type: none"> • Heater Demand (to be implemented) 	<p>An analogue potentiometer connected to the analogue input defines the desired temperature. This input will only be used if the last action was a press of digital input 1 and the heater is running.</p>

Example external expansion port circuitry

Digital inputs:

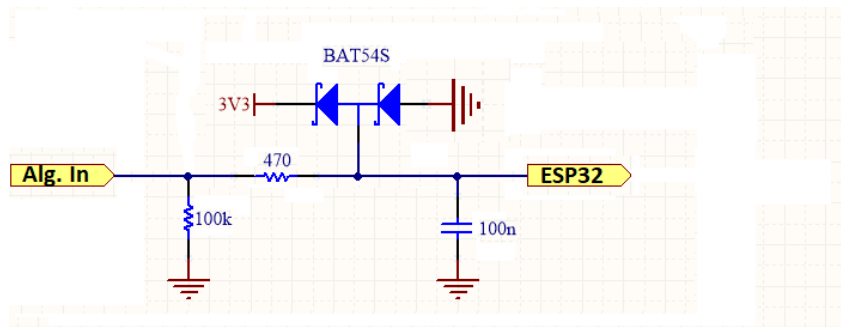
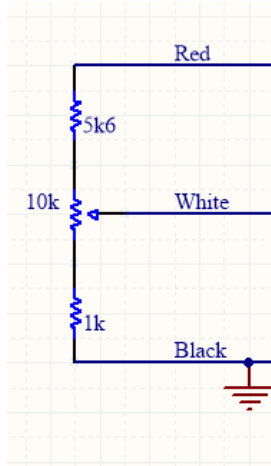


Digital outputs:



Afterburner user manual

Analogue input:



Bluetooth Connectivity

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.

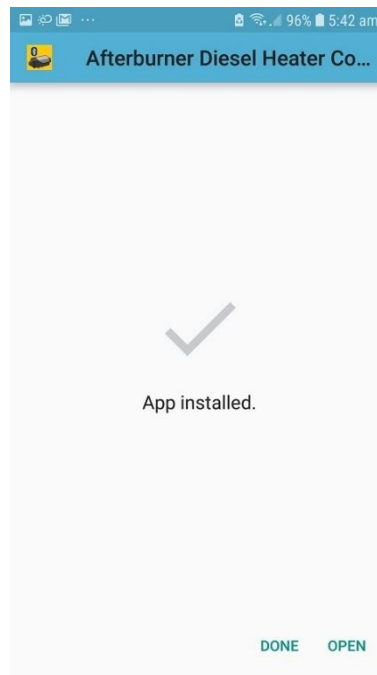
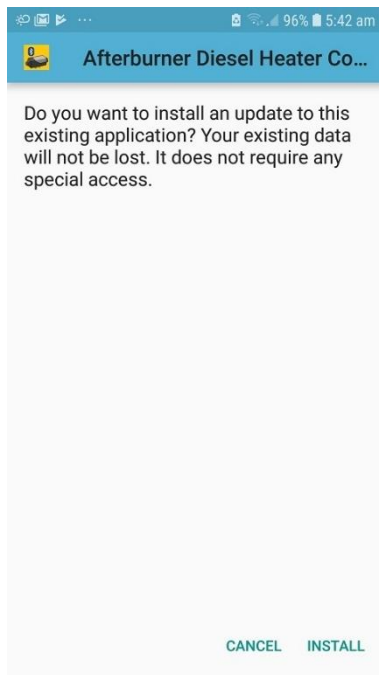
Download and save the **Android Bluetooth application** to your Android device from

<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.

Afterburner user manual



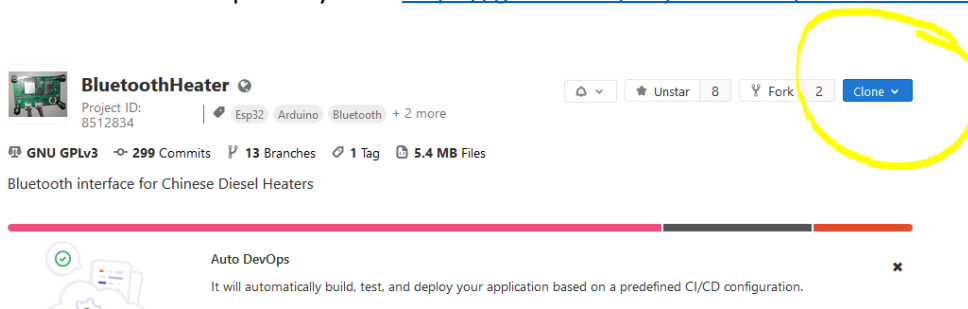
Modifying the Android application

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.

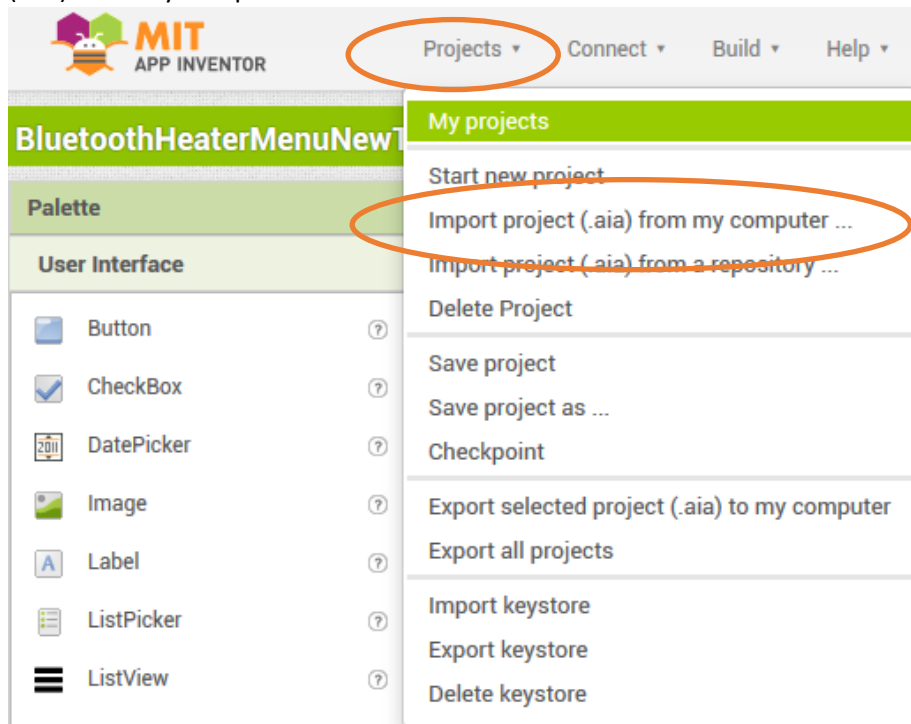
Rest assured though to just upload the design to your device the process is not all that difficult:

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

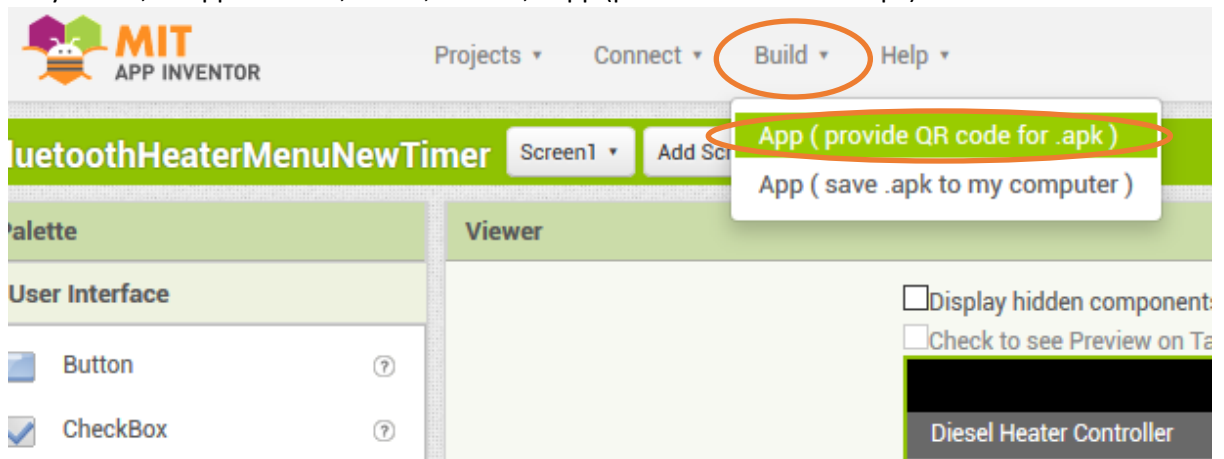


2. Unless you want to push back proposed design changes, it may be easiest to clone as a ZIP file.
3. In the downloaded repository, locate the "AppInventor" subdirectory which is under the repository's root directory.
4. The file for the Android design will have a ".aia" extension, presently it is BluetoothHeaterMenuNewTimer.aia
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!

- Once logged into App Inventor, select the “Projects” menu, then click upon “Import project (.aia) from my computer...”



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

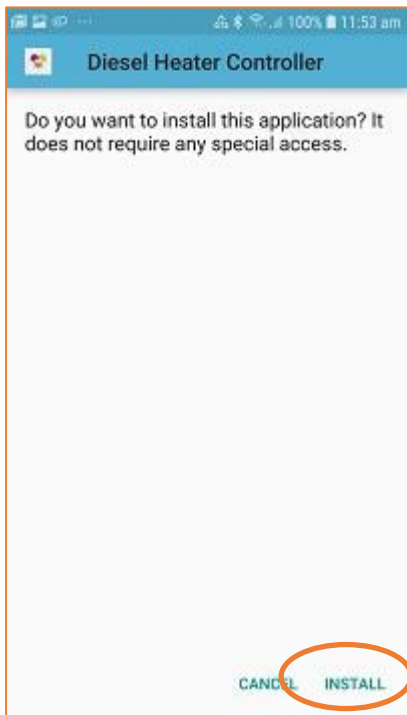


- The design will then be compiled, and eventually a unique QR code for this instance will then be presented on your PC's screen.

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



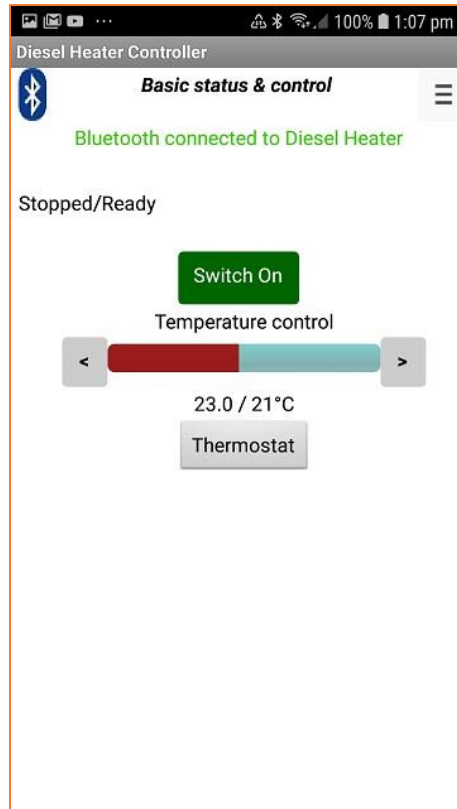
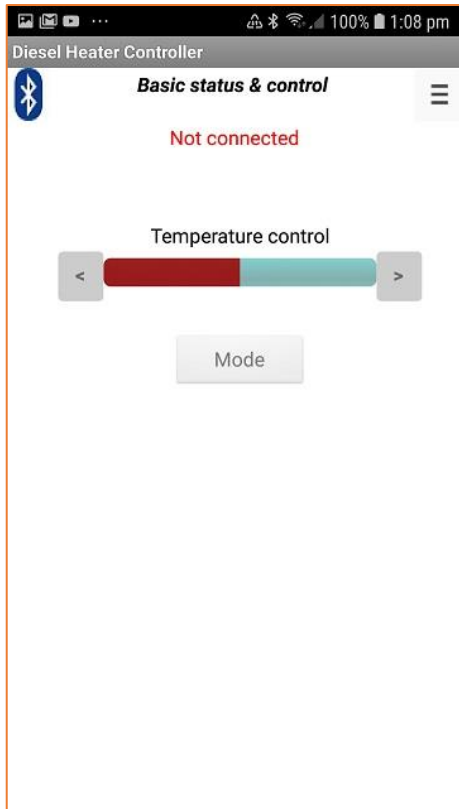
16. The application will then be uploaded to your Android device, but first you must authorise.



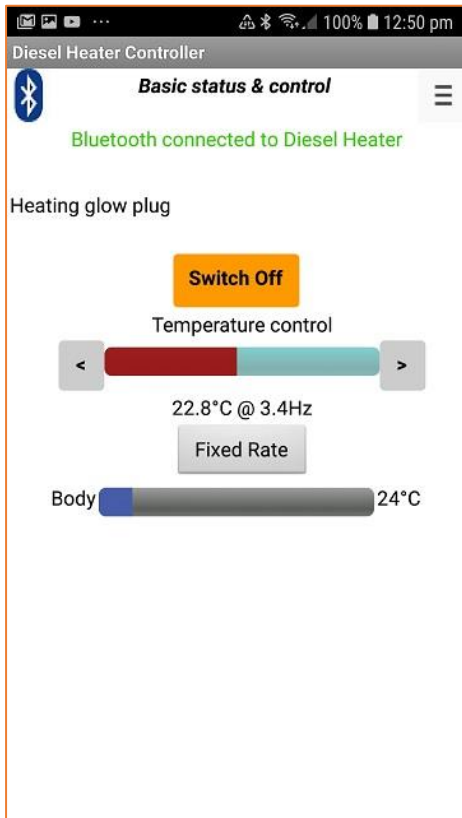
17. The application will now be installed to your Android device.

Using the Bluetooth Application

1. Once installed on your Android device, the 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.



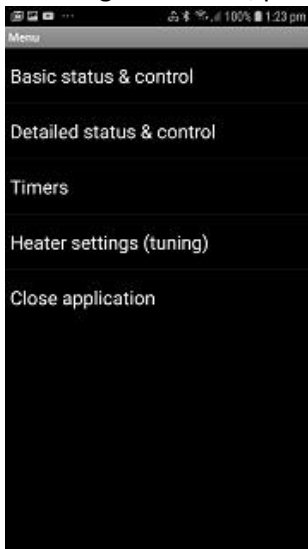
Go on, Press the big green button!



Menus

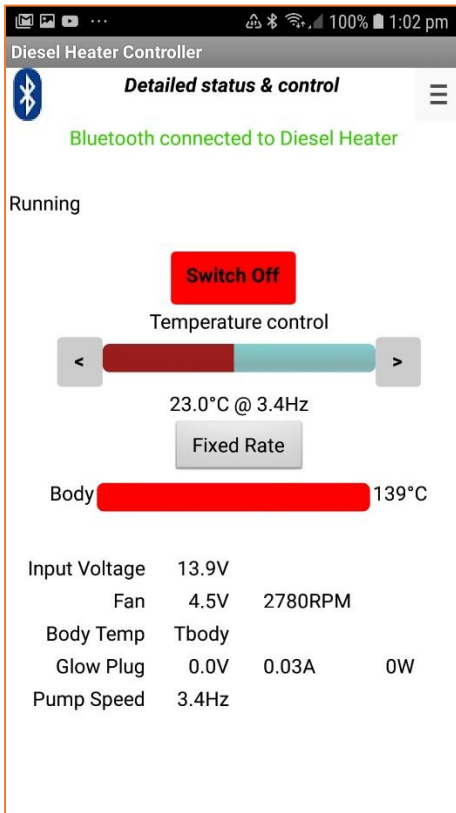
The application always starts in Basic Status & Control mode

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.



Timers mode

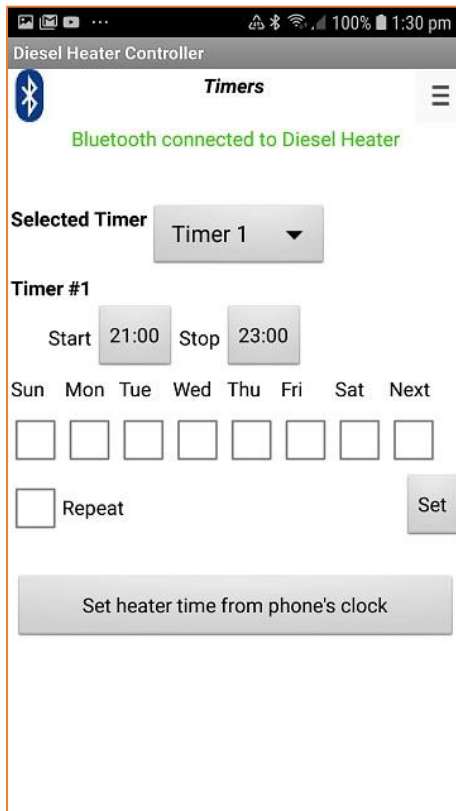
The timers provided by the Afterburner are extremely flexible.

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:



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.

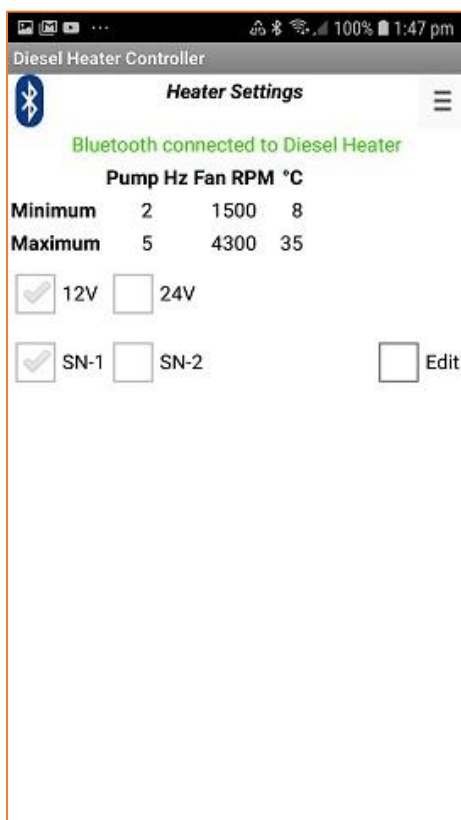
If OK the timer is set.

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.

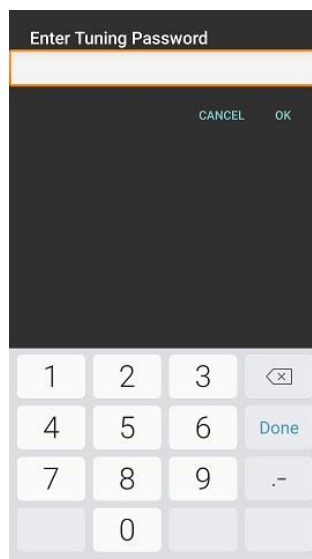
Heater Settings mode

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



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.



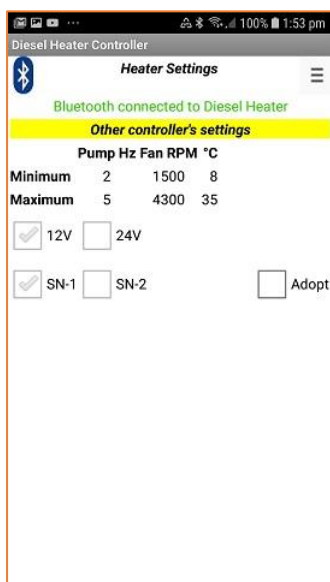
The password required is “1688”.

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.



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:

- Inbuilt Access Point - a Wi-Fi device can directly connect to the Afterburner.
- 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 BTCESP32-xxxx, where xxx is the last 2 fields of the AP's MAC address.

- Select the BTCESP32-xxxx 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 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.

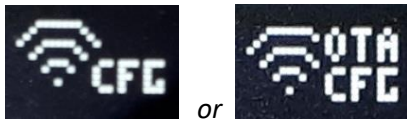
Enabling the Configuration Portal

Using the OLED [Wi-Fi Settings](#) menu, select **CFG AP only** to serve the Configuration Portal.

When the new mode is selected and accepted, the Afterburner will reboot into the new web server mode.

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.

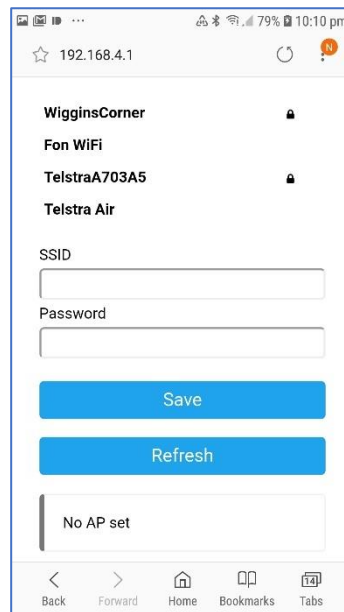
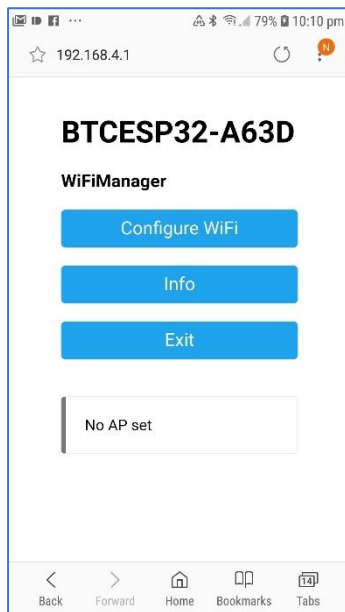


Connecting to the Configuration Portal

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.

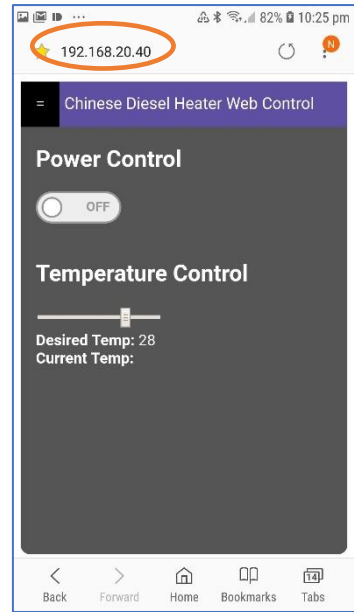
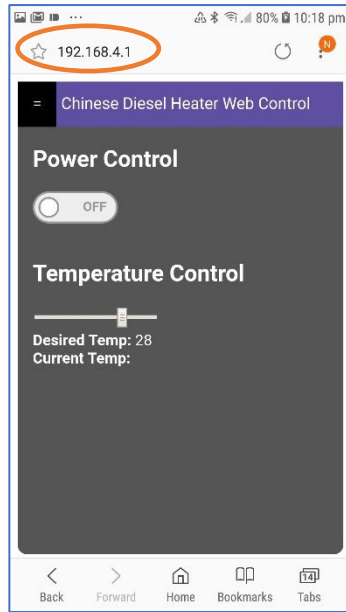
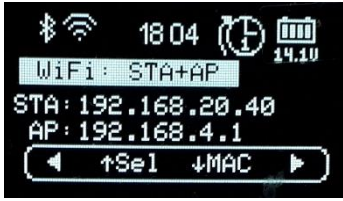
- The ESP32 will reboot.
- The web server will now serve the Diesel Heater Web Control page.

Web server connectivity is now available via:

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

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

Afterburner user manual

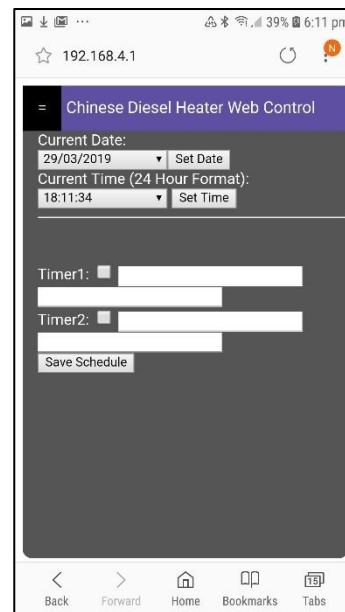
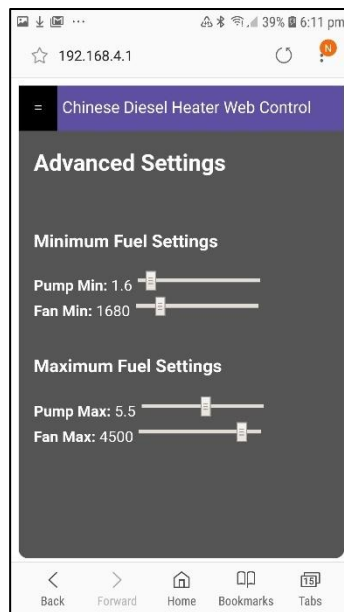
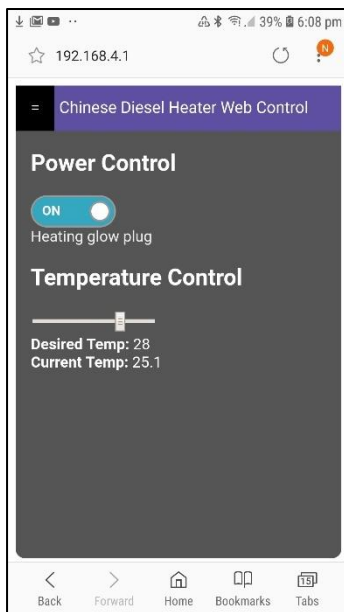


Heater Web Control

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:



The initial heater control web page allows basic heater control; on/off & temperature demand. To toggle to thermostat mode you currently need to use the OLED menus.

A couple of other views are available via the menu bar at the top left, being fuel settings and clock settings. The timer settings are presently ineffective – the design of the timers has since been changed, web pages are lagging behind.

The web pages still do need a bit of work, but they're a start!

If you're a web wizard, by all means improve upon them and offer the improved pages back – this is an open source project after all!

The web pages themselves are retrieved from the ESP32's SPIFFS partition. The files in the SPIFFS partition are uploaded from the repository's [Arduino/BTCDieselHeater/data](#) directory. Details on how to upload are in the [Firmware updates section](#).

Arduino Environment

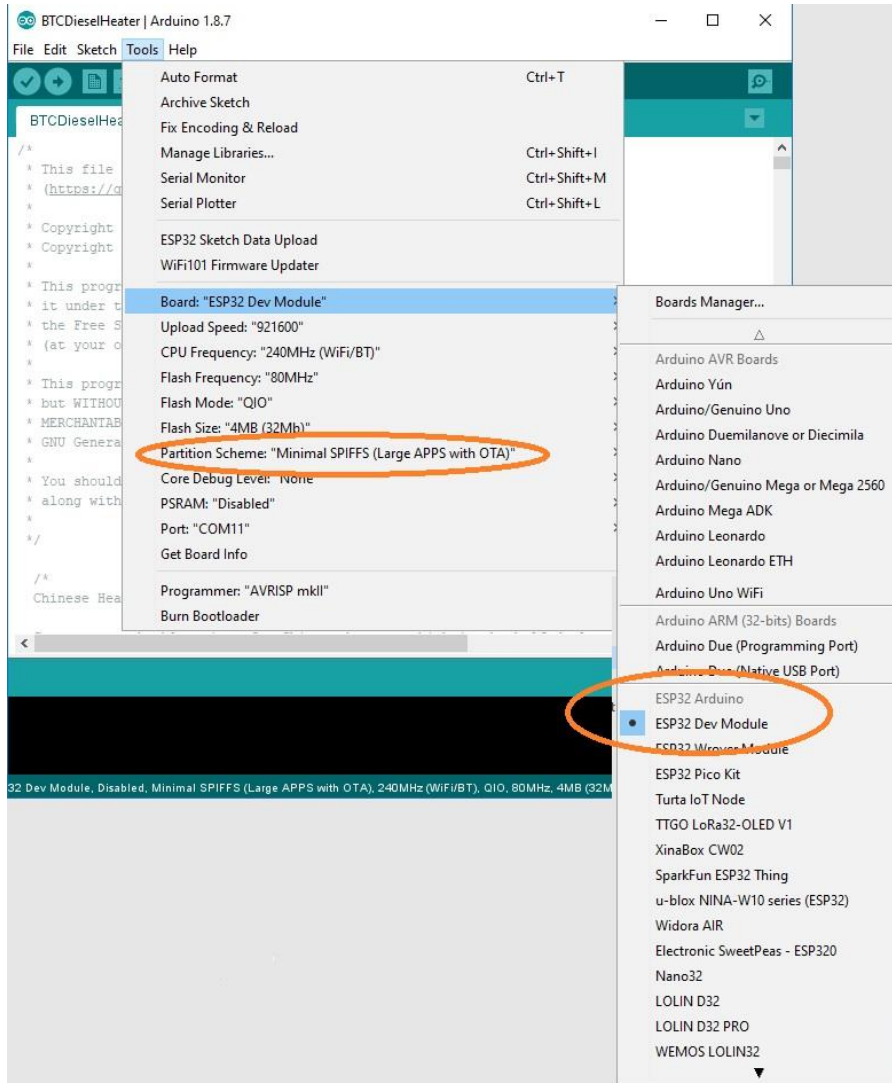
ESP32 build settings

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.



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

If the Afterburner is connected to a WiFi network, which itself is connected to the Internet, it is possible for the Afterburner to interrogate the Afterburner web site and determine if a newer 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.

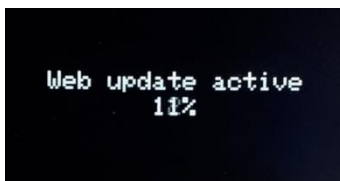
To authorise the upgrade, navigate to the Version Information menu, located in the User Settings menu loop.

Presuming an update is available, a larger animation takes place to the left side of the menu:



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

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



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.

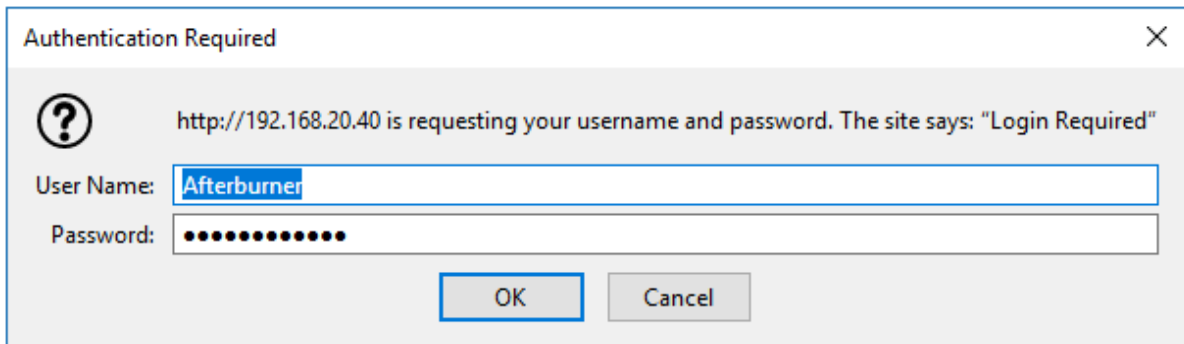
Browser update

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

Using a web browser, the Afterburner's browser update web page can be opened. 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 address, eg if using AP mode <http://192.168.4.1/update>

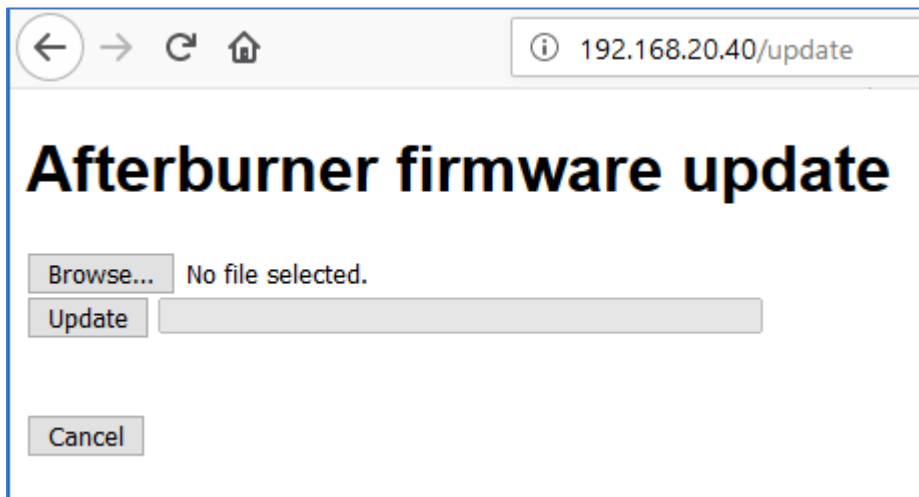
This will then present a password confirmation page:



The required credentials are:

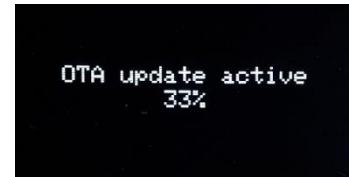
- **Username:** Afterburner
- **Password:** BurnBabyBurn

The following screen should then be presented:



Click upon **Browse** and locate the binary image previously downloaded from <http://www.mrjones.id.au/afterburner/firmware.html>.

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



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 HTML*) file to upload:



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:

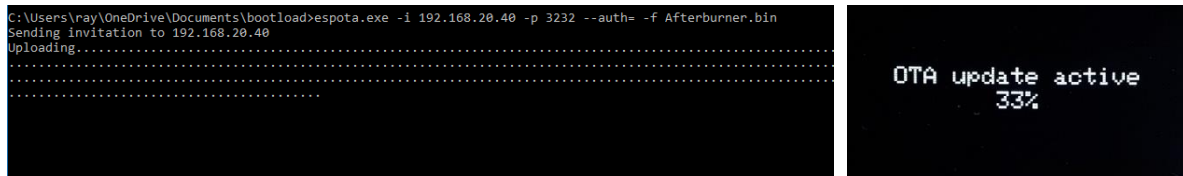
```
espota.exe -i 192.168.20.100 -p 3232 --auth= -f Afterburner.bin
```

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:



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:

Afterburner user manual

```
C:\Users\ray\OneDrive\Documents\bootload>esptool.exe --chip esp32 --port COM14 --baud 921600 --before hard_reset write_flash -z --flash_mode dio --flash_freq 80m --flash_size detect 0xe000 boot_apper_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. I.e. 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

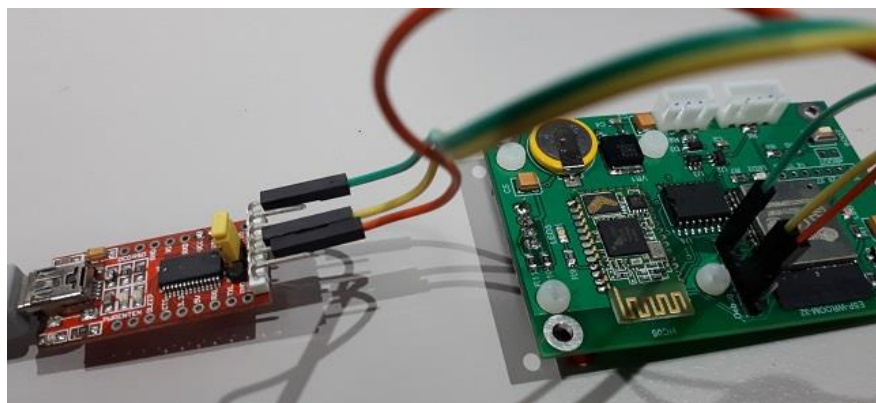
New firmware is uploaded to the ESP32 via its debug/programming serial port.

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:

5. Press and HOLD the **EN** button.
6. Press and HOLD the **BOOT** button.
7. Release the **EN** button.
8. 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 not accessible on the MK2 PCB when installed into the case's front panel. The provided header fields 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/SPIO_RE1_RE0_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:



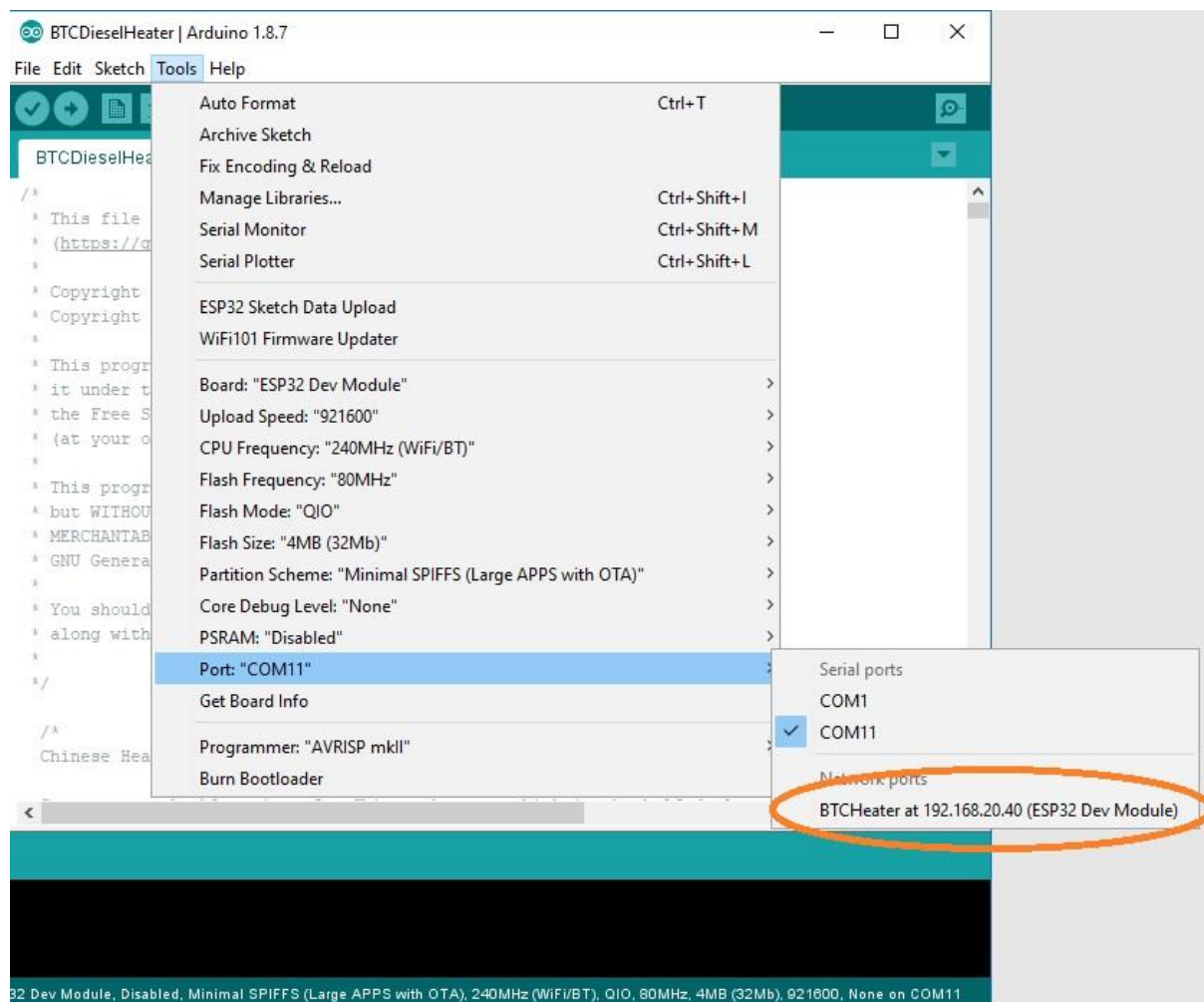
If not, you will also need to [use the Wi-Fi settings menu](#) to enable OTA.

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!



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.

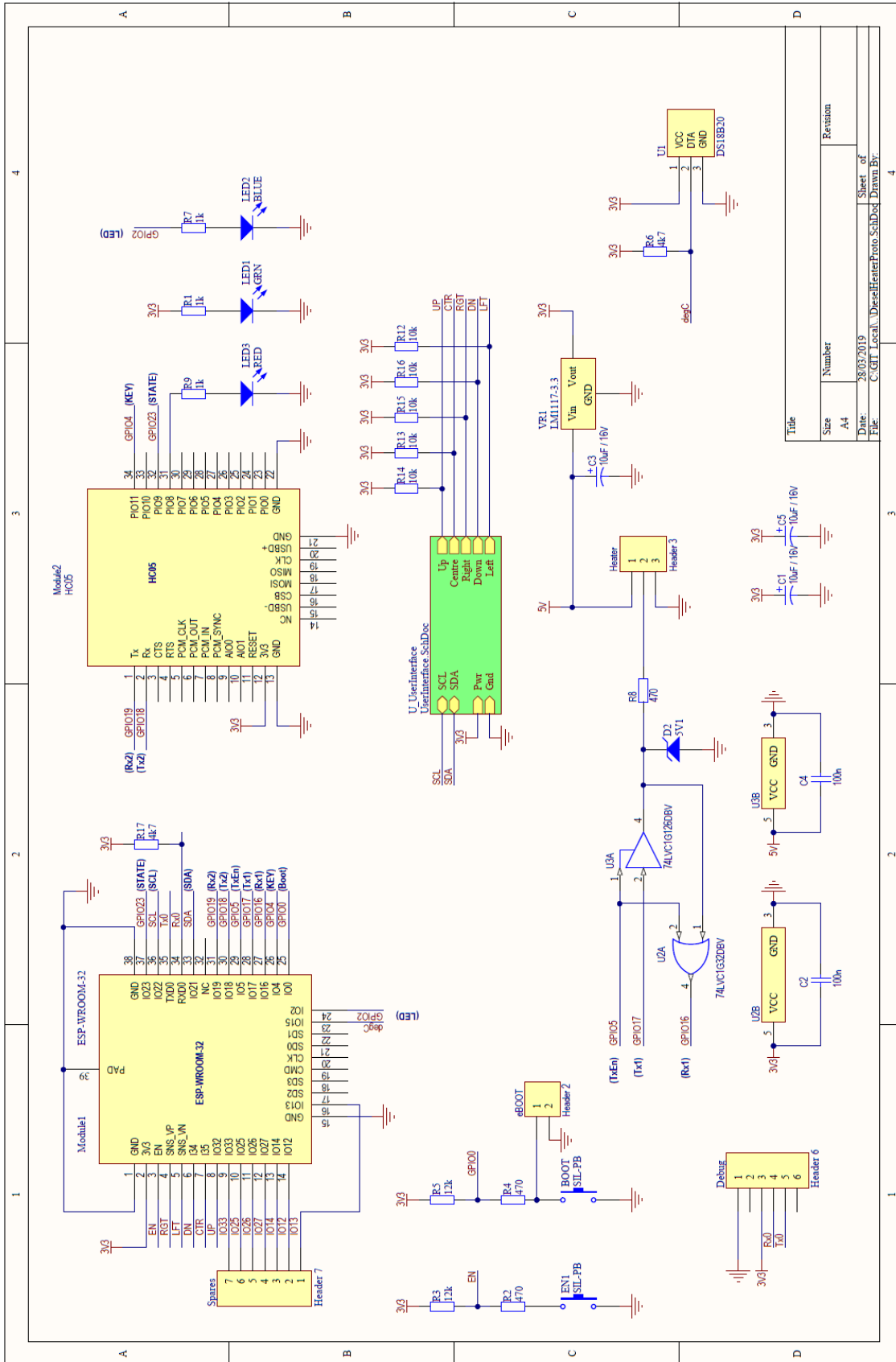
If using serial bootloading, ensure the USB to serial adapter is connected to the Afterburner's debug/programming port, and the ESP32 is ready for bootloading by performing the [EN/BOOT button sequence](#).

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

The contents of the **BTcdieselheater\data** directory will then be uploaded to the ESP32's SPIFFS partition.

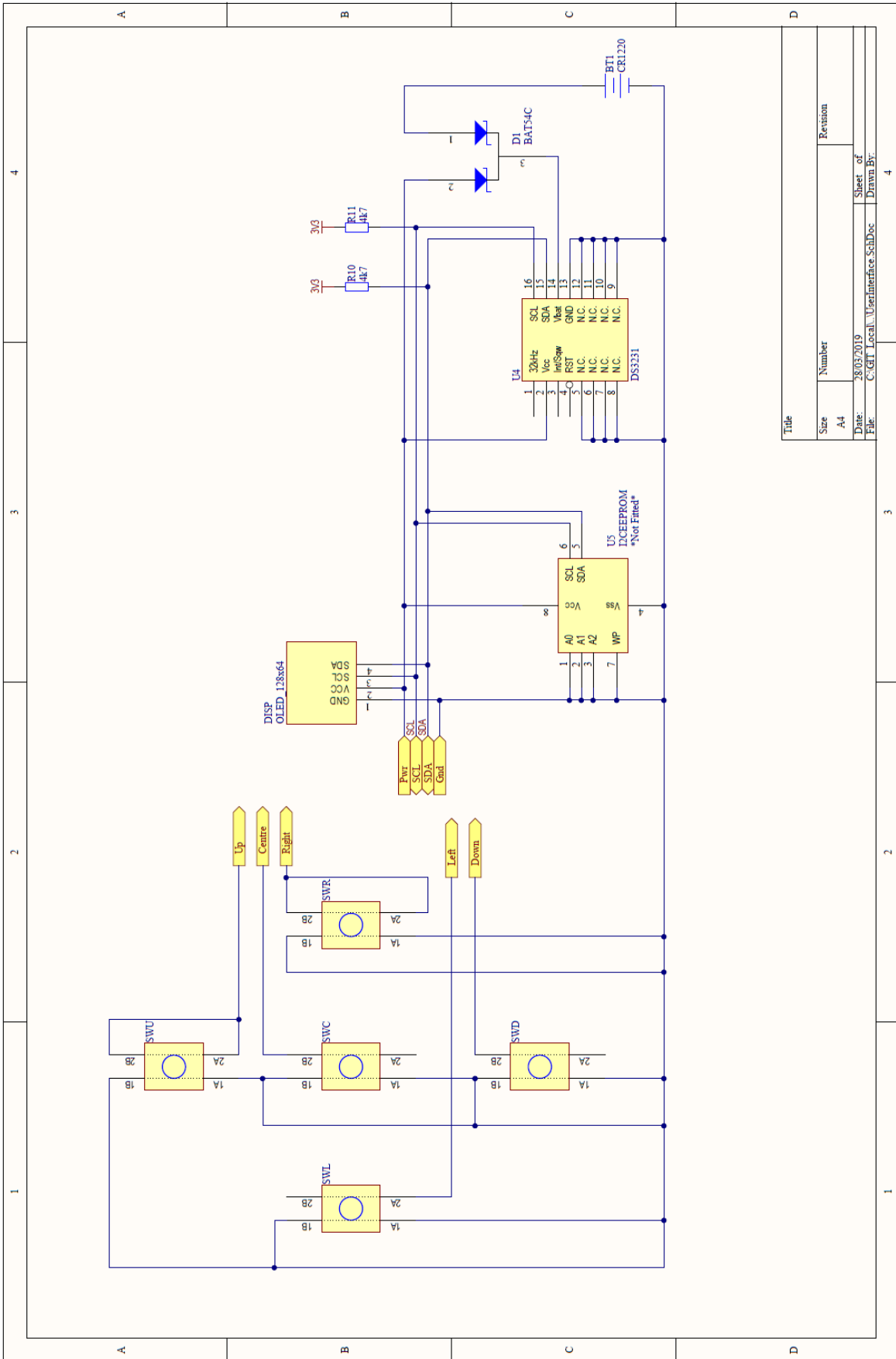
Schematics

Mk1 PCB



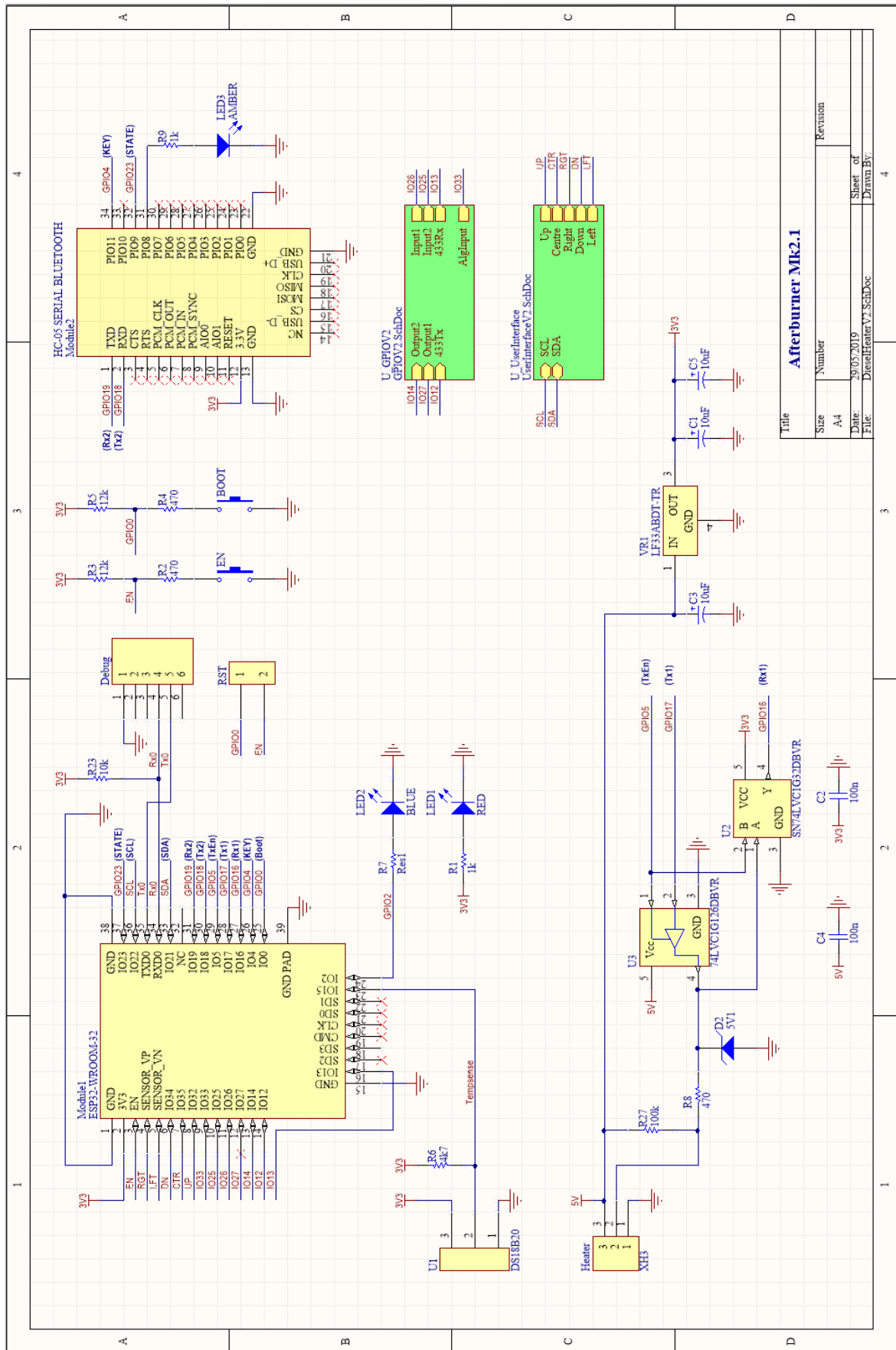
Title		Revision	
Size	A4	Number	
Date	28.03.2019	Sheet of	4
File	C:\GH_Local\Desktop\Afterburner\Proto_SchDoc	Drawn By	

Afterburner user manual



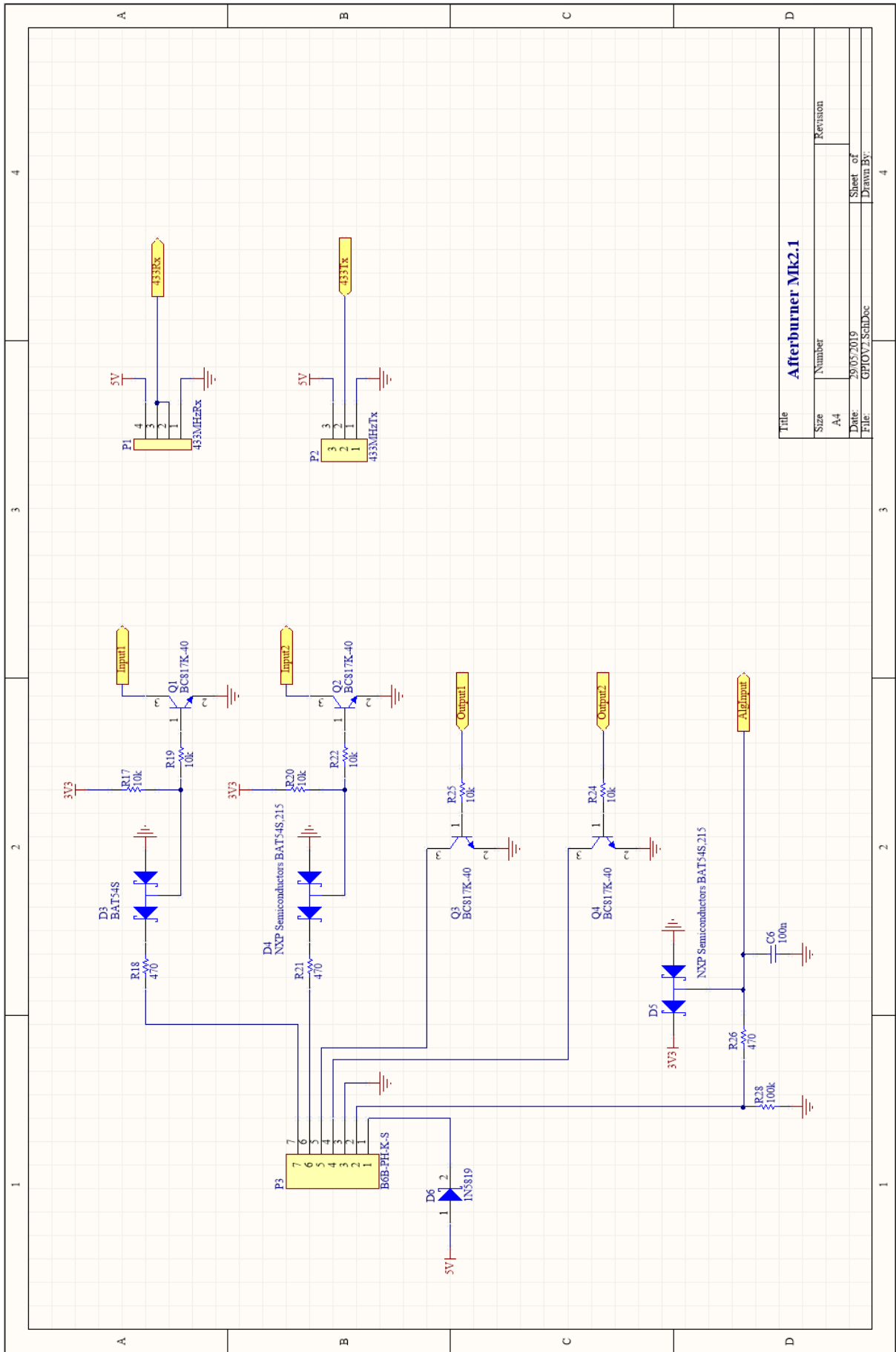
Title	
Size	Number
A4	Revision
Date:	Sheet of
File:	Drawn By:
C:\GIT Local\...UserInterface SchDoc	4

Mk2 PCB



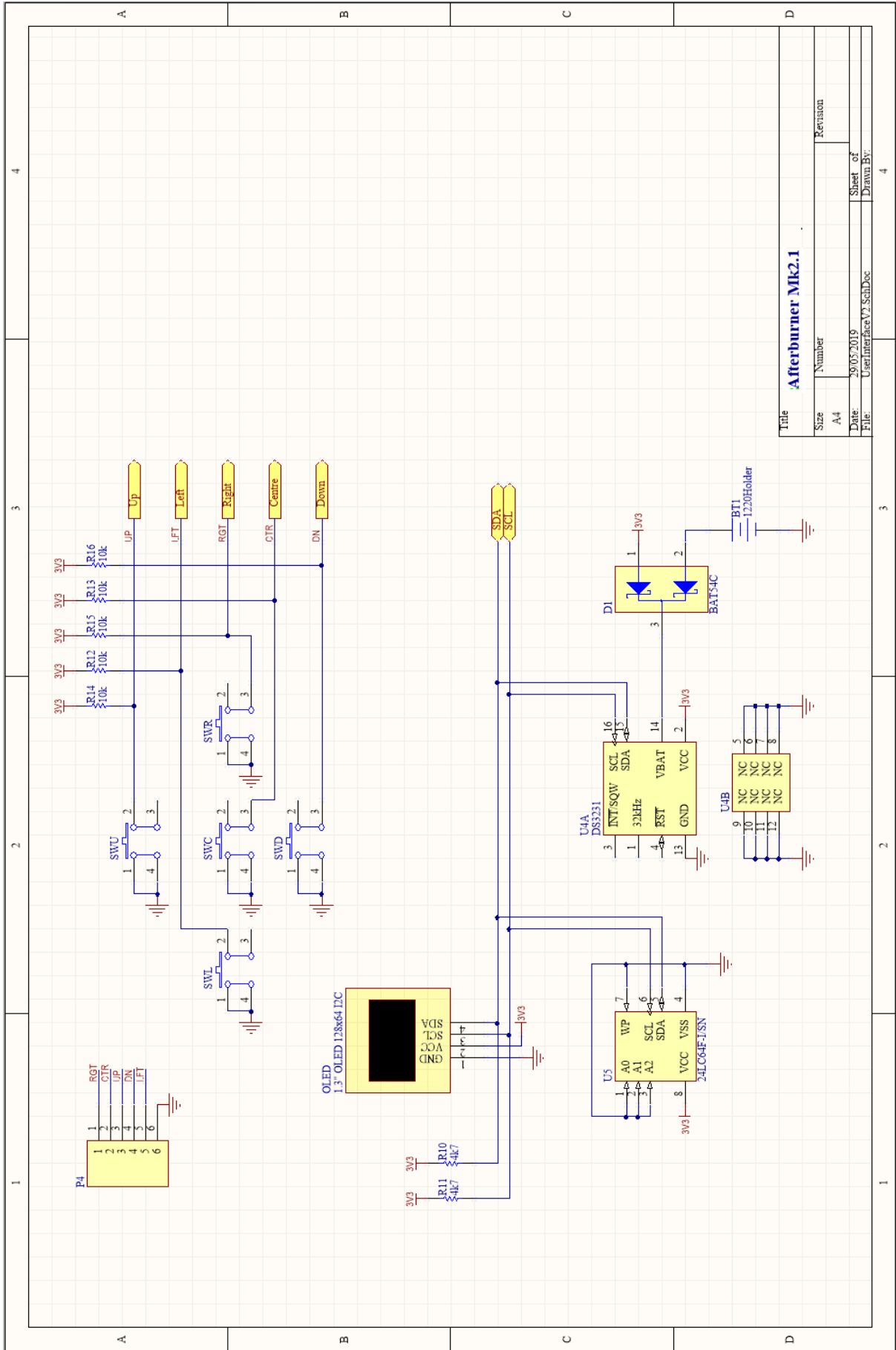
Title		Afterburner Mk2.1	
Size	Number	Revision	
A4			
Date	30/05/2019	Sheet of	4
File	DesellHaterV2.SchDoc	Drawn By:	

Afterburner user manual



Title		Afterburner Mk2.1	
Size	Number	Revision	
A4			
Date:	29/05/2019	Sheet of	
File:	GPIOV1.SchDoc	Drawn By:	

Afterburner user manual



Title		Afterburner Mk2.1	
Size	Number	Revision	
A4			
Date:	29/05/2019	Sheet of	
File:	UserInterfaceV2 SchDoc	Drawn By:	