GPS Unit Kit – Instructions

A series of articles has been published in previous issues of Archive describing the design and construction of a GPS receiver and speedometer:

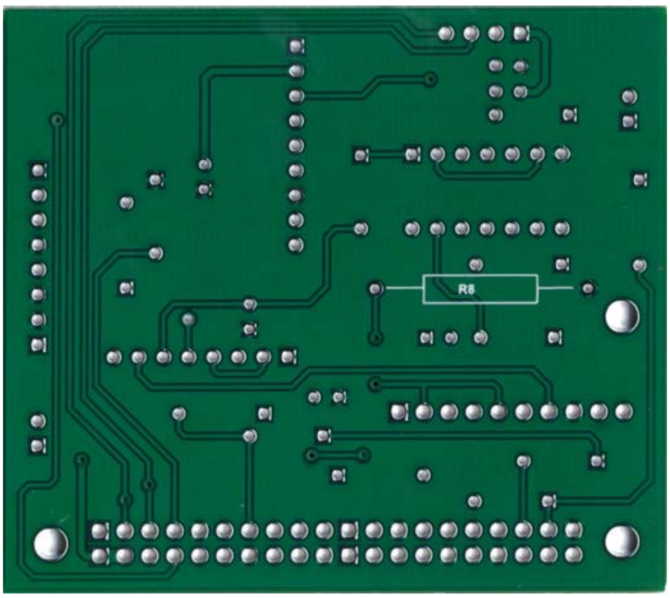
- In part 1 (Archive 24:2) I described a Raspberry Pi that could be carried around: instead of a mouse and monitor it had an Adafruit touchscreen that just plugged in to the HDMI and USB sockets. It had a GPS module sending position data to the mapping application (RiscOSM) by means of URI_Dispatch messages generated by the !Satnav application. It thus provided a rolling map of your location using Open Street Map data.
- Part 2 (Archive 24:3) described version 1.08 of Satnav, which could control a small text display or an OLED display and could talk to RiscOSM using Wimp messages.
- Part 3 (Archive 24:4) adds a Witty Pi for control of power consumption. Satnav is at version 1.40 and can control an 'electronic ink' display (Papirus).
- Part 4 (Archive 24:5, October 2019) adds an Adafruit power-boost board to extend battery life, an Adafruit ADS1015 ADC board to provide battery management, and an improved odometer function. Also added is power-switching under software control and GPX logging (rather than Anquet Export File (AEF) format).

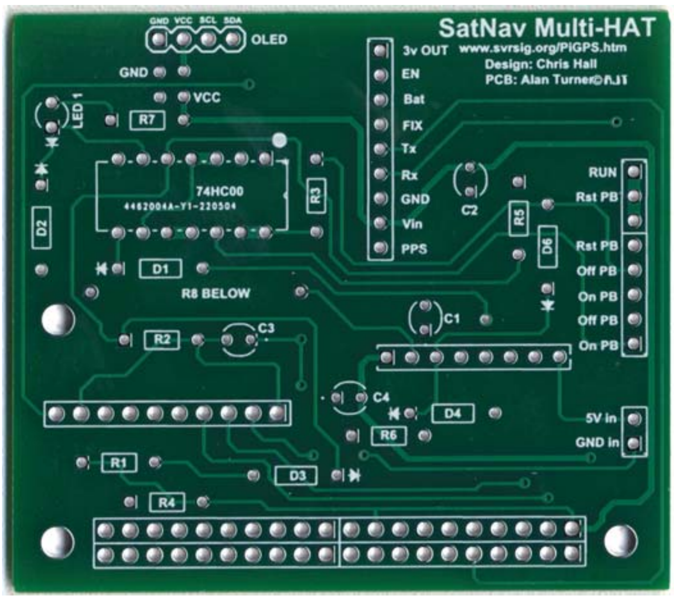
What's New?

Part 4 of this series was written in May 2017, updated in August 2018 and published in October 2019. The last version described had many flying wires to connect the various boards and push buttons together. It had a digitial voltmeter which, on the press of a button, would show the battery voltage, from which its remaining life could be inferred.

An article in Archive 24:6 described an improved version of the power control circuitry but focused on the construction of the case for 'a small portable'. It included the circuit diagram for power control and mentioned the use of an Adafruit ADS1015 4-channel analogue to digital conversion board.

This kit now uses a custom-built printed circuit board, illustrated below. The component list is also provided.





The custom PCB allows plenty of space for soldering the components. The large wire wound resistor sits below the PCB just above the micro USB sockets on the Pi Zero.

Construction - Parts List

1 x Printed circuit board
4 x resistors 15kΩ, R1; R4; R5; R6
2 x resistors 100kΩ, R2; R3
1 x resistor 2.2kΩ, R7
1 x resistor 0.1Ω, R8
5 x diodes 1N4148, D1-D4; D6
1 x LED, LED1
4 x tantalum capacitors 1μF, 16V, C1-C4
3 x push buttons
5 x header plugs (4, 8, 9, 10 and 40 pin)
1 x 74HC00 IC
1 x 2x2 header plug right angled
2 x links for above
----
1 x 1.3" OLED module
1 x Adafruit ultimate GPS module
1 x Adafruit ADC 1015 module
1 x Adafruit Power Boost 1000C
1 x Lithium Ion 4400mAh battery
1 x Raspberry Pi Zero WH
1 x SD card 16GB
1 x mains adapter
1 x USB pen drive 128MB
1 x USB adapter
1 x HDMI adapter
1 x GPS antenna cable
1 x GPS antenna
1 x CR1220 battery for the GPS unit
1 x case

Step 1

Taking the parts above the dotted line, solder the various components in place on the circuit board. The diodes need to be the correct way round, as shown the bar on the end of the diode indicates the end to which the current flows, i.e. is at lower voltage. The tanatalum capacitors need to be with the '+' indication towards the power rail and the other end to ground.

Note that there are two 'ground' levels connected by the 0.1Ω resistor.

Once the circuit board is connected, plug it in to the Raspberry Pi Zero and connect to an HDMI monitor. Power the unit from the micro USB socket on the Pi Zero, with a keyboard and mouse plugged in via a USB hub to the other micro USB socket. Plug in the OLED module but leave the other modules not connected. Make sure the links are fitted correctly so that the polarity for the OLED unit is correct (the power and ground pins are often swopped).

Download the Sat Nav unit SD card image - this will allow the unit to start up with no HDMI monitor connected and to start SatNav on power up with messages appearing on the OLED module.

The unit should start up and display a message on the OLED 'Sat Nav 2.50'.

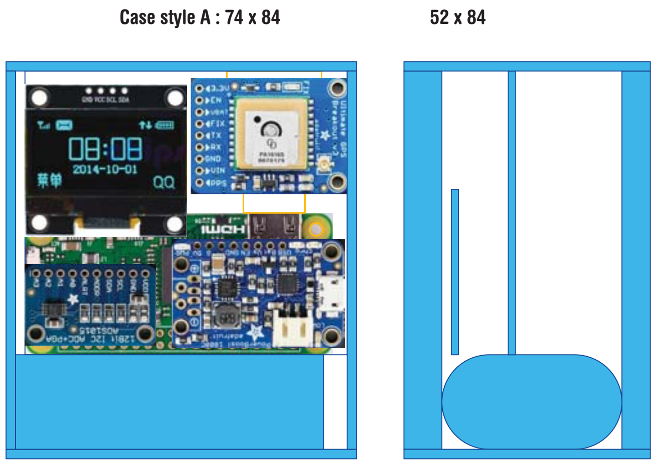
Step 2

Power the unit off and plug in the remaining three modules. Remove the power input to the Pi Zero and attach it to the micro USB socket on the power boost board. Plug in the Lithium Ion battery to the power boost board.

Pressing the 'on' button should now turn on the power. Pressing it again (once start up has been completed) should display battery voltages and remaining capacity. When the display dims it is wating for the 'on' button to be pressed, when the display shows in inverse mode, it is waiting for the 'on' button to be released.

Every other time you press the 'on' button for battery information it toggles a software switch to ignore or enact power off requests.

Pressing the 'off' button makes a shutdown request - SatNav will complete its logging, then initiate a ROM reset. The CMOS time and date is noted and as GPIO4 goes to high impedance, power is removed. This process ensures the SD card is not being written to when power is removed.



An example layout of the four modules in a case

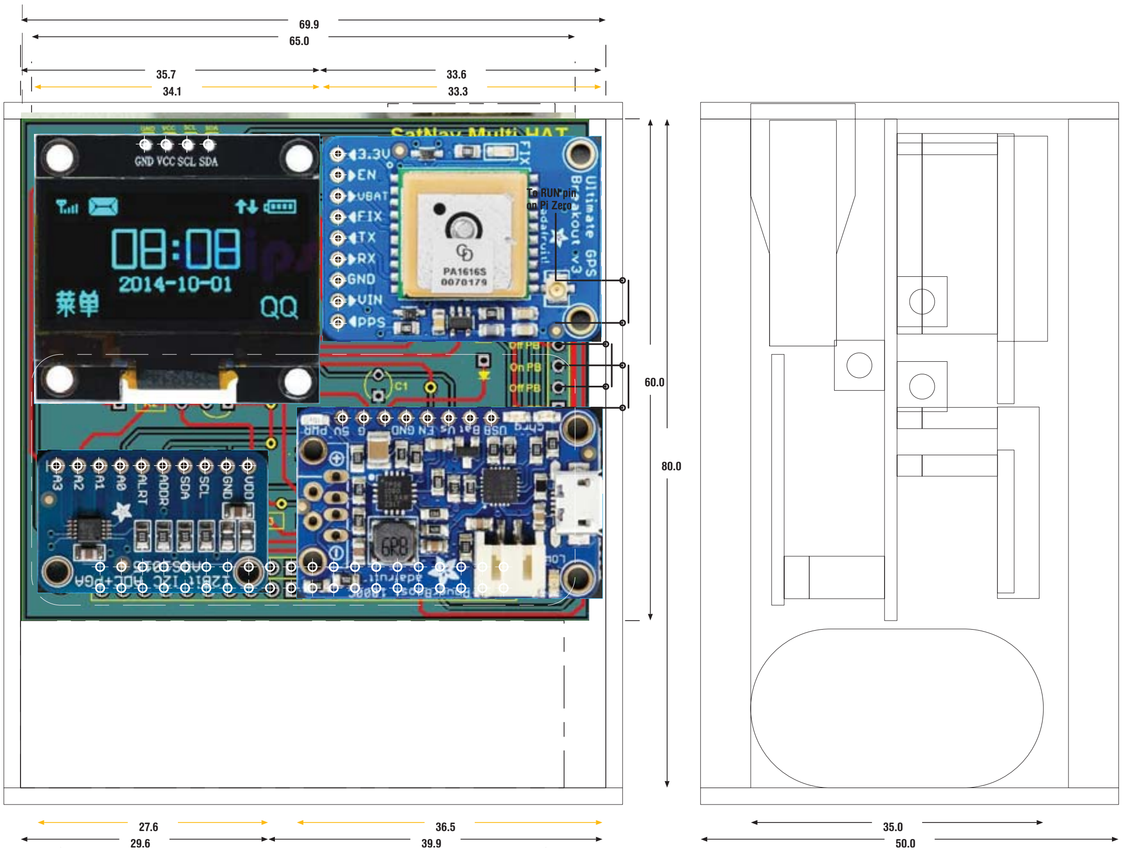
Testing

The power boost board features two DC to DC convertors - one to charge the battery from the external source and another (controlled by the 'enable' input EN) to provide 5.2V power which takes its input, via switching diodes, from either the external power source or the internal LiPo battery, whichever is the higher voltage. This voltage, about 40mV below battery voltage when there is no external power supplied, is available as an 'always on' power source from the internal battery.

The ADC board must be powered at between 2V and 5.5V and at a voltage at least equivalent to the highest voltage it is monitoring. Otherwise it will back power its supply rail via its protection diodes. The 74HC00 chip can be powered at between 2V and 5V and controls power supply switching. Both must therefore be powered continuously.

This difference of 40mV is within the ADC board's maximum rating of 300mV above supply voltage as one of its inputs monitors the battery voltage directly.

With the ADC board supplied from the 'always on' voltage and the GND pin of the power boost board, the next question was how to monitor the current consumption. Power to the unit, apart



A final layout check - the PCB is 68mm x 60mm and the internal dimension of the case is 70mm wide so the HDMI socket should just break through the back but the USB socket just be clear of it. The power boost micro SD socket should just touch the side so a cut out will allow power to be supplied to it.

from the 74HC00 and the ADC board, is supplied to the Pi Zero via the 5V pins on the 40 pin connector and the 3.3V rail is provided from the Pi. The current returns via the GND pins on the 40 pin connector and runs to the GND pin on the power boost board (and the ADC board) via a 0.1Ω resistor. This generates a small voltage on the GND plane of the PCB which is connected to 'A1', one of the monitoring inputs of the ADC board. A voltage of 256mV is set as 'full scale deflection' indicating current consumption of 2560mA with a resolution of ±1.25mA.

By monitoring current consumption and battery voltage, the remaining battery capacity can be shown as a percentage and as an endurance time in hours. The board also measures the voltage of the cell on the GPS module and of any external power source.

At each stage the hardware has become either more compact or more functional but each time has been based on a breadboarding style where a bespoke circuit is constructed. Some items such as a liquid-ink display or use of the Witty Pi to turn the unit on after a prescribed time have proved an unnecessary refinement.

Different breakout boards have been used - ProtoPAL, Zero Breakout as well as a GPS board with some breadboarding space. I have also used a Witty Pi board to allow a shutdown unit to be reawakened at a specific time and date.

The final version now uses four plug-in modules: a GPS module which deals with the satellite GPS signals; an OLED display module which provides a display in lieu of an HDMI monitor; an analogue to digital module which allows voltages to be measured and a power boost board which allows the unit to be run from a battery.

The 'Mulit-HAT' board can now be tested with no ancillary boards plugged in (supplying power directly to the micro USB socket on the Pi Zero). Each board could then be added in turn and tested individually.

Originally I used the 5V supply for the ADC board but then realised that when the unit was shut down the 5V would be back-powered from the protection diodes. The ADC board is therefore supplied continuously from the 'always on' supply.

The ADC board is thus supplied at a higher voltage than the 3.3V voltage used for the logic levels on the GPIO lines but these are not connected to it. The IIC lines are driven to low but go high with pull up resistors. Logic high is defined as at least 70% of the power voltage which means between 3.5 and 5V. I am hoping that 3.3V will prove OK as logic high (inactive). It did.

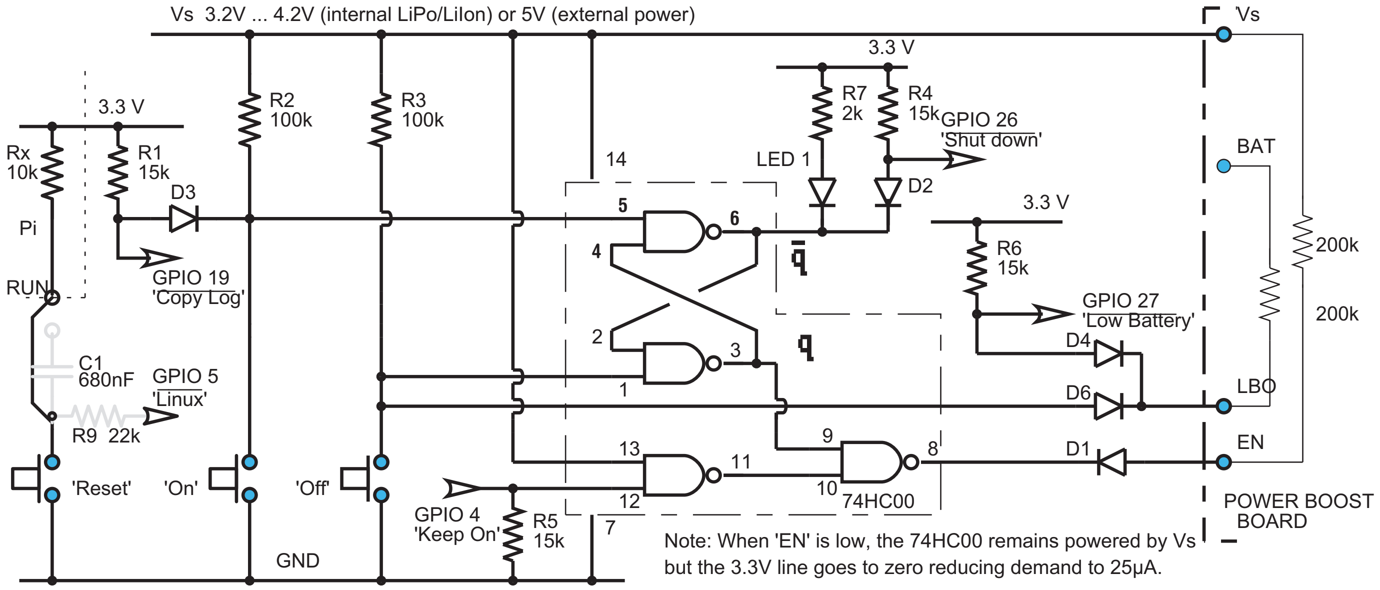
The ADC board measures voltages directly (the input impedance of inputs A0 to A3 is 10MΩ at a FSD of 6V and 100MΩ at a FSD of 256mV) rather than via a resitive divider circuit.

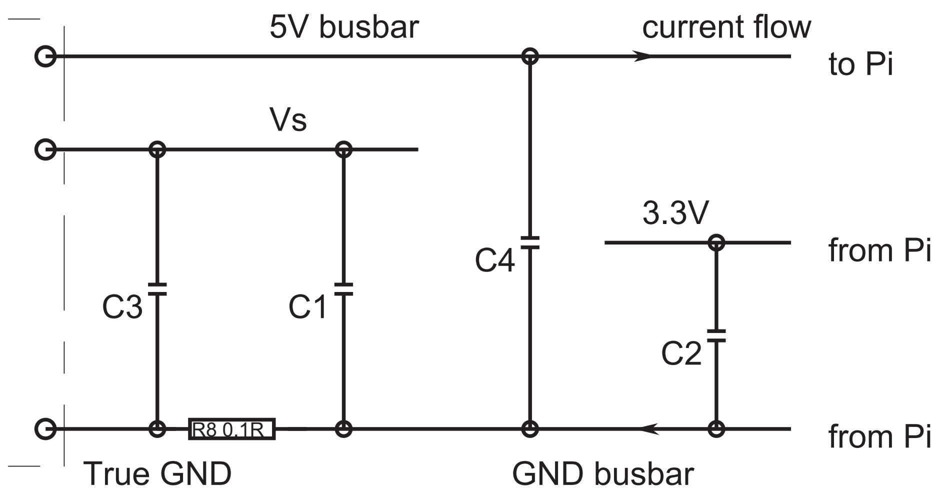
How It Works

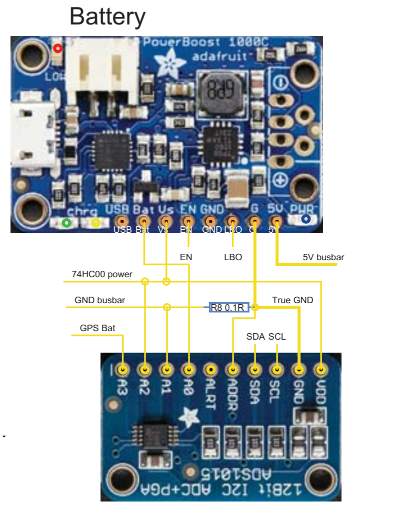
When the unit is turned on by pressing the 'on' button, the GPS module will flash at 1Hz indicating that a position fix has not yet been established. The GPS unit has an on-board battery backup so that so that satellite details (their trajectory in the sky) can be stored and the time and date maintained in a real time clock whilst the unit is shutdown.

This allows a 'warm start' when the unit is switched on, otherwise the details of each satellite it sees have to be slowly downloaded to establish its position in the sky. For reasons which I cannot work out, each time I switch the unit on, the GPS module initialises to 00:00 on 6 Jan xx80 (taken to be 2080) - this may be because I often plug in an external antenna.

RISC OS itself (in the absence of a







Top: *the circuit diagram for power switching*

Above left: *power supply smoothing & current flow*

Above right: *interconnection of Power Boost and ADC boards*

real time clock, a date of 2080 from the GPS module being rejected by SatNav as not valid) will initialise itself to the time and date at which it was last tidily shut down. In the initial stages of start up, therefore, RISC OS has two alternative dates, both of which it knows to be wrong.

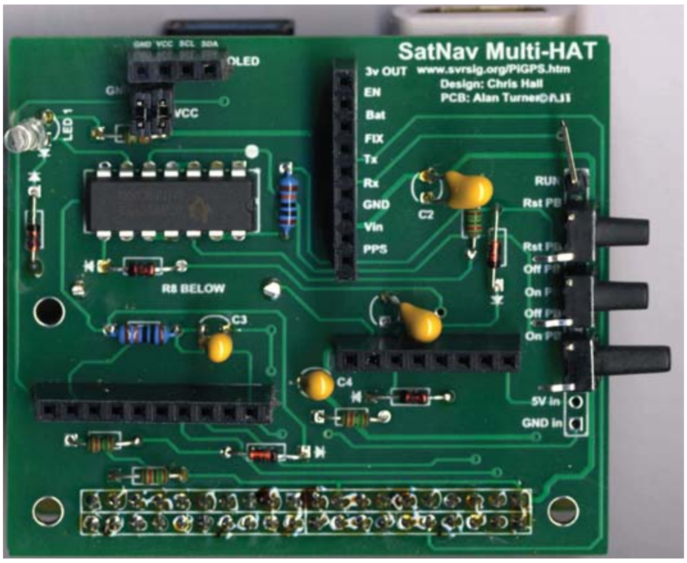
When the GPS module 'sees' a satellite it starts to download information from it and starts to include some of this in its output to the serial port. Each satellite maintains an extremely precise time and

Item	Source	Price
Pi Zero WH	PiHut	£13.50
PiZero Essentials	PiHut	£6.00
USB adapter	Amazon	£2.57*
PCB	JLCPCB	£5.24*
Resistors R1, R4-R6 15k Resistors R2-R3 100k Diodes D1-D5, D7 1N4148 Capacitors C1-C4 1μF 16V	Rapid Elx	£2.00*
74HC00	Rapid Elx	£0.50*
Header skts 4,8,9,10,20x2	PiHut	£5.00*
Push buttons	EBay	£0.50*
OLED display 1.3" IIC	EBay	£7.00
GPS module	PiHut	£28.80
ADC 1015 module	PiHut	£8.64
Power boost 1000C	PiHut	£18.90
LiPo battery 4400mAh	PiMoroni	£15.00
SD card 16GB	PiHut	£6.00
Polycarbonate case	Plastic People	£10.00*
	Total	£129.65
* - approx. price if bought in quantity		

date and so SatNav will start to show the correct time (in GMT), the number of satellites it can see and the number of those whose trajectory it knows.

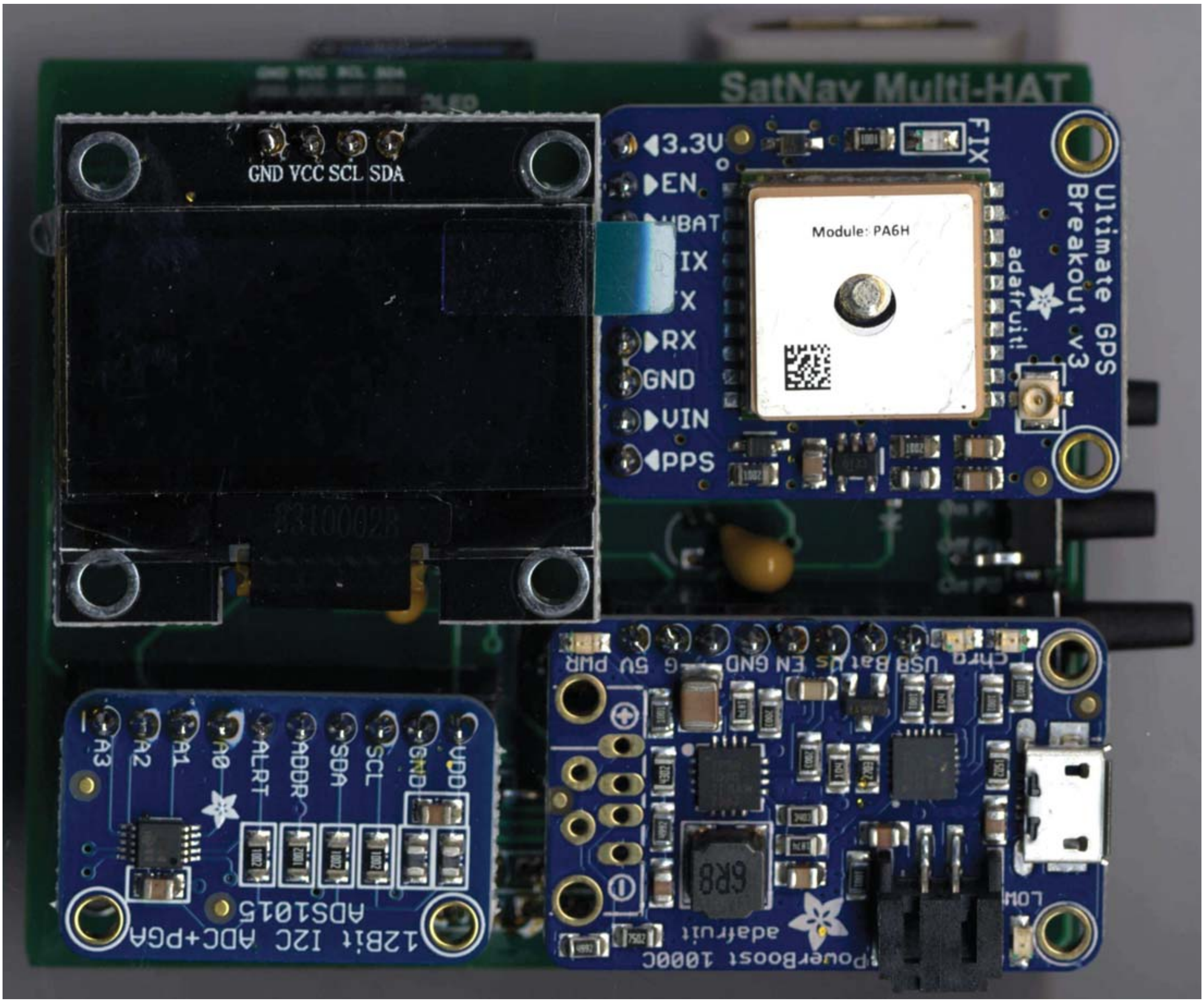
The power boost board is controlled from the output of the 74HC00 chip except that an 'always on' supply at just below battery voltage or at about 5V when an external power source is connected (i.e. between 2.7V and 5V) is provided for the few components that are powered continuously (the 74HC00 and the ADS1015). With no overhead from a DC to DC convertor, the unit consumes about 25µA (100µW at 4V) whilst shutdown.

After a few seconds the OLED display will show a time of '00:00' and '00/00 sats'



Above: *Components soldered and the board works!*

Below: *With the modules fitted, their exact position can be seen - almost as intended but with the GPS module slightly too far to the right.*



which will update to the correct time (in GMT) as soon as at least one satellite has been seen. At this point the RISC OS time will update to the correct time (also in GMT but daylight saving corrections for the territory will be applied in the desktop.

Pressing the ‘on’ button will attempt several things - it will try to download the logging file if a USB pen drive is plugged in and the display will change to show battery voltages, current consumption and remaining battery life in hours. The LiPo battery produces about 4.4V when fully charged, falling to about 3.4V when it is close to empty, after which it falls more rapidly, tirggering a ‘low battery’ condition at around 2.7V.

Pressing the ‘off’ button will make a shutdown request (via GPIO 26), expecting the SatNav software to complete the logging activity (which writes to the SD card) and then to enforce a shut down/ start up cycle. This will update the CMOS date and time and then reset the ROM, which will take GPIO 4 to a high impedance state, removing power. If SatNav is not running, then power will be removed on the next reset. This ensures the SD card is not being written to when power is removed.

If a low battery condition occurs, this is equivalent to a shutdown request with the additional indication (via GPIO 27) that the battery is low. The same sequence leads to power being removed.

There are several LEDs indicating status: a blue LED shows that power is being applied to to the Pi Zero. A green or yellow LED indicates that external power is being supplied and the LiPo is either fully charged (green) or charging (yellow). A steady red LED indicates a low battery.

A flashing red LED indicates that the GPS module is powered up - a flash every second means no position fix has been established whereas a flash every 15s means that the position has been determined.

The position fixing process is a bit complex. Each satellite broadcasts timing pulses and the difference in reception time between signals indicates the difference in flight time. The implied differential distance and the known position of each satellite in the sky means that three or four satellites are sufficient to establish latitude, longitude and height above sea level.

Once a position fix is established, the OLED will display the grid reference, milepost mileage (if close to the SVR) and speed in mph. Above 3mph just speed and time are displayed. Although the unit is normally used with no HDMI monitor connected, a full desktop is being ‘shown’ and messages will be sent to any RISC OS mapping software. RiscOSM will start up and display a rolling map of the current location.

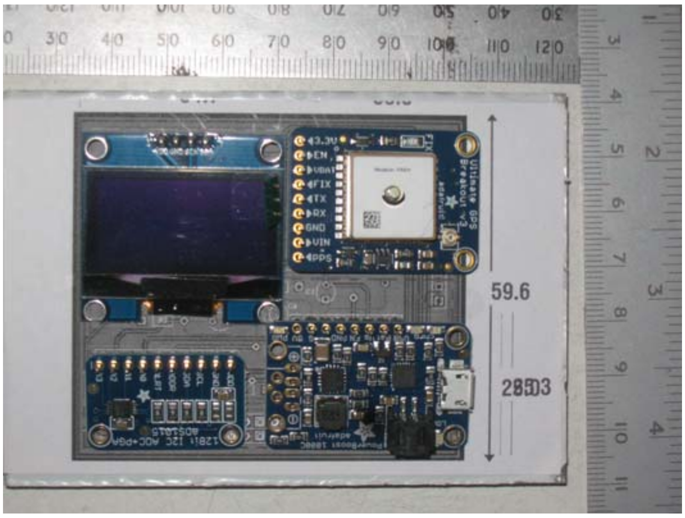
How much?

So how much did it cost? Ignoring the many trial versions and with some prices having actually come down, about £130 plus a bit extra if you want an external GPS antenna.

Conclusion

The GPS unit should now sit on the car dashboard or on a table in a railway carriage and show the speed: when stationary it will show the grid reference. The journey will be logged and can be downloaded onto a USB pen drive at the press of a button. The only trailing leads are the battery itself and the GPS antenna socket otherwise eveything is mounted on the circuit board.

An external power source - powerbank or mains adapter - can be plugged in (and unplugged) without affecting operation. This will supply the device and charge its internal battery at the same time.

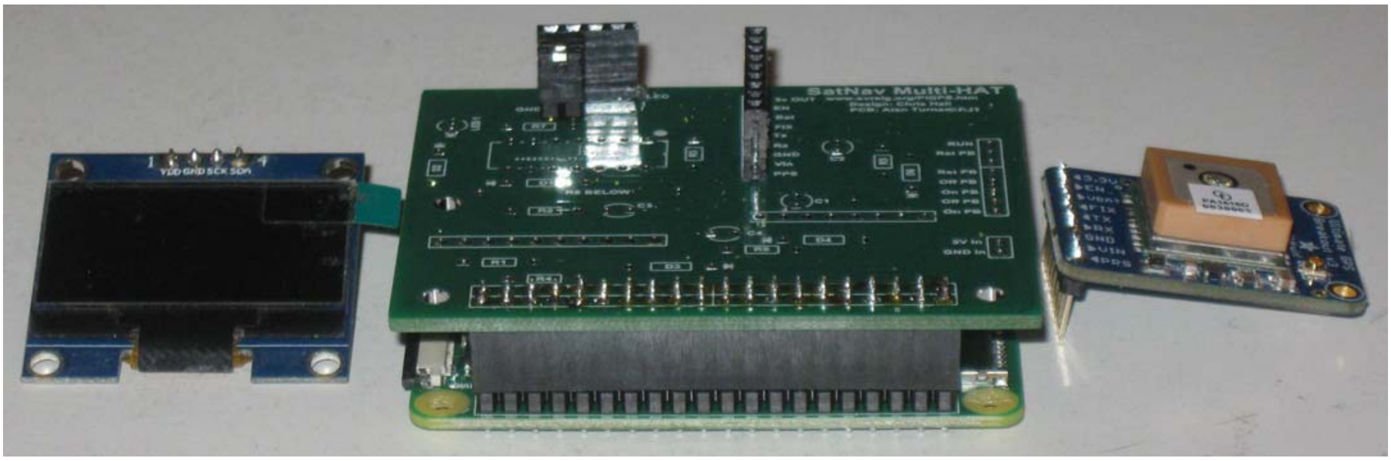


Above: *A mock up showing the board fully populated. The plan was that the right hand side would butt up to the micro SD card socket with the GPS module just clearing the side. As you can see, the GPS module protrudes slightly. The Pi Zero underneath will have an HDMI and a USB adapter which are flush with the case at the rear. Holes will therefore just accommodate the bare metal part of the plug with the adapters retained.*

What else can it do?

Another possibility is that just an OLED display and a GPS module be fitted. The circuit board can thus be partially populated and SatNav will cope with the missing modules (accessing the ADC board and power switching circuitry will automatically be disabled).

Chris Hall chris@svrsig.org



Above: *The circuit board has been populated with the socket for the OLED display, the links to select the polarity of the display and the socket for the GPS module. With power supplied directly to the micro USB socket on the Pi Zero, this unit will function as a GPS speedometer (and display a full RISC OS desktop on an HDMI monitor if one is connected, including a rolling map showing position) at minimum cost.*