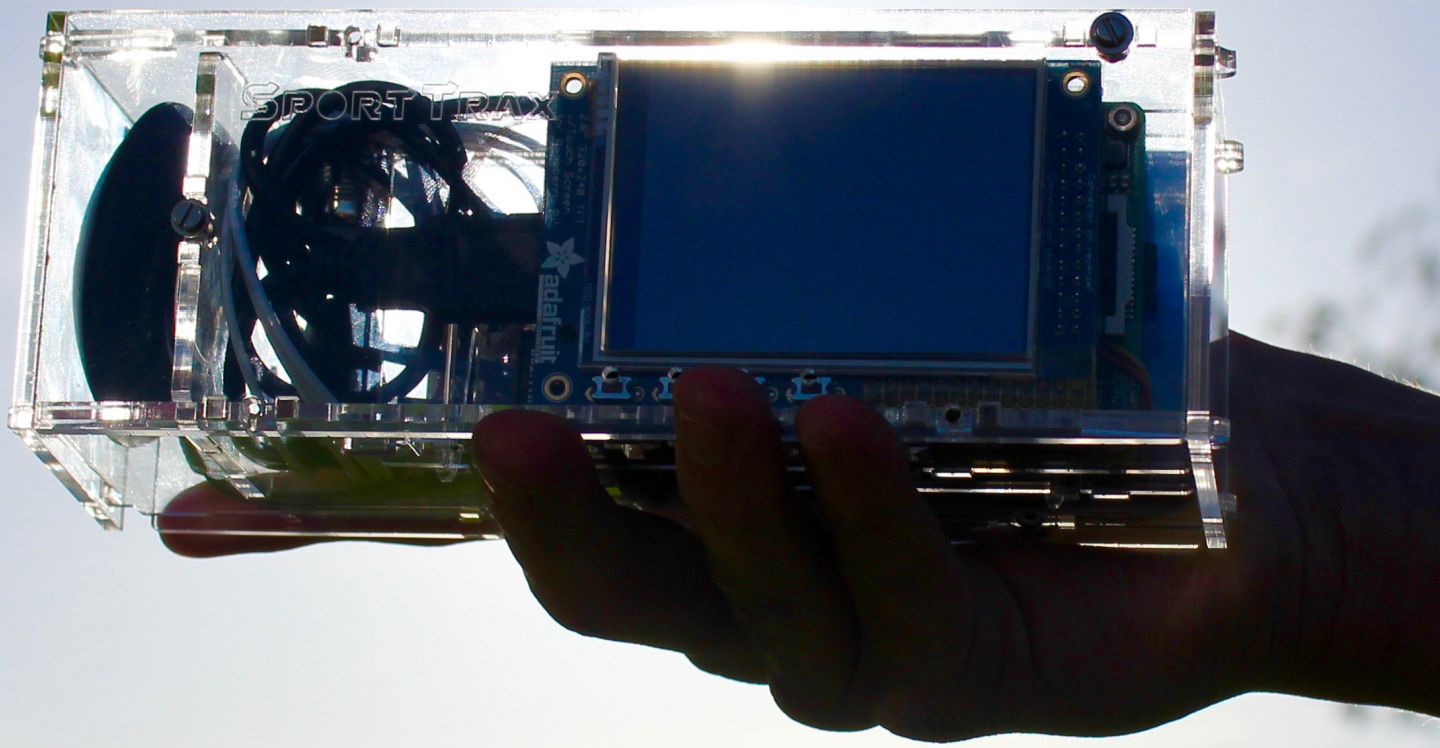
SportTrax

The Build Guide




INTRODUCING SPORTTRAX


TESTING THE PROTOTYPE


DEVELOPMENT AND CONSTRUCTION

OUR JOURNEY TO SPORTTRAX

WE WERE INSPIRED...

With our background in computing the SportTrax Team were able to identify an issue within society. We developed a product to provide a solution. During development of our revolutionary device our goal was to put technology into the hands of both professional and amateur sports enthusiasts. From our experience with friends and social networks we have had first hand involvement within this tightly-knit community helping them to engage with computing and electronics. We've taken the opportunity to teach these devotees and welcome them to the world of electronics. Of course, that is not to suggest similar products aren't available within a highly advertised finance driven industry. They are! We, however, have used our creativity to produce a product that brings sports technology and tracking to life in a new way.

DOWN TO EARTH

In terms of the product we needed a starting point. A place from which we could progress to developing a product which we hope to be a revolutionary leisure-based device. Our starting point came from an article within the MagPi that used a Pi Zero and a series 3 GPS module. From this we further developed a concept on having a standalone device that did not require any complex connections or wiring for the end user - unless of course they wished to take the product from the point of being a complete kit. From this mere concept, we produced 'SportTrax'.

Sammy Herring

TEAM LEADER

Sammy has been involved in a number of computing and electronics competitions since the age of 10, when he won the SEGFL National Computing Award. Today, he continues working in the world of electronics with the end goal of teaching within computer science and mathematics.

sammy.herring@social-systems.co.uk

@Sammy_Herring

#SPORTTRAX

In order to help publicise our campaign we utilised social media, in particular 'Twitter'. We achieved this through the use of '#SportTrax'. In doing so, we were even retweeted by the Raspberry Pi Foundation, while HackaDay even tweeted about our project (twice!). We received lots of support for our project - some 'tweeters' even asking for copies of our build guide.

'PROFESSIONAL GRADE TECHNOLOGY, IN THE HANDS OF ABSOLUTELY ANYONE' - SAMMY HERRING

INTRODUCING SPORTTRAX

In today's society, we found a problem. The problem is that sports enthusiasts fail to engage with technology. Regardless of the modern day use of mobile technologies within the sports industry, anyone who wishes to become a power user of technology is swiftly declined this opportunity due to the limitations of portable GPS/accelerometry systems. Therefore, we have come up with a new solution. SportTrax is a product we have designed to be a precise tracking device for almost any form of physical activity. The device offers data logging features such as altitude, direction, velocity and location. However, in actual fact SportTrax is so much more. We have designed SportTrax to be a modular device which allows an aficionado with a combination of these interests to not only use such an advanced device, but optionally build their own as a kit.

Equally, someone with little experience would have the ability to 'hack' our device and customize it to their own requirements – for example, placing a larger battery aboard the device. It should be mentioned that we have chosen the components in our product with care for the everyday athlete who wishes to have the ability to adapt their product for their own use. This is because those who work extreme scenarios may need to adapt their SportTrax for their own uses – which we have ensured is an easy process.

Of course, it is a requirement that the product itself is not only aesthetic but has a casing which allows it to be safely and easily transported, while ensuring it could be easily reproduced or manufactured. Therefore, we have opted to use Perspex as it allows the casing to both be easily altered or cut/engraved. On the casing itself we have also labelled all ports to ensure our product was user friendly.

'SPEED, LOCATION, ALTITUDE AND DIRECTION' - MARK LANE

A HANDHELD MODULAR DEVICE

James Hoyle

VIDEO PRODUCER

James has had a passion for computing from a young age. From fixing to building, computing has been one of his hobbies and passions, right from the get go. This highly influenced him on determining his life's ambition, to become a mechanical engineer. He is currently studying mathematics, physics and computer science.

JamesMcHoyle@gmail.com

@JamesMcHoyle

DISPLAY SETUP

2

ADAFRUIT 2.8" TOUCHSCREEN DISPLAY

You would probably think that installing the Adafruit 2.8" Touch Screen was one of the hardest parts of making the SportTrax in its entirety, but you would be wrong. Its far simpler then you would have imagined. You start by opening up the Raspberry Pi Console and typing in the following command:

```
curl –SLs https://apt.adafruit.com/add-pin | sudo bash
```

What this does it is adds the apt.adafruit.com to your repository list. This is so you can easily access code from the Adafruit servers. After you have entered this command you will next need to type in the following two:

```
sudo apt-get install raspberrypi-bootloader
    #Followed by:
sudo apt-get install adafruit-pitft-helper.
```

What these two commands do is actually download the programs required to run the touch screen. It's important to note that this will take a while to download as there is a large amount of software to replace for PiTFT support.

After this point in the setup, the kernel and the helper for the device have been installed onto the Pi itself. After this you need to run the helper as this will configure the kernel device tree overlays and also add a few configurations to give you access, and the ability to view, the console. Now you will need to input the following command into the Raspberry Pi Console:

```
sudo adafruit-pitft-helper –t 28r
```

By executing this command a helper installation program will appear in your console. In turn, it will install all of the dependancies and PiTFT interaction tools onto your device. You will also have the oppourtunity to setup the Display's GPIOs. On the first GPIO we opted to have a soft shutdown - although this is optional. Once completed, your display will be ready.

TEXTUAL INTERACTION

3

XSTROKE GESTURES

In terms of inputting textual information, we are limited to only a display - unless we wish to use a Bluetooth keyboard. Therefore, we have implemented a gesture input system, known as 'XStroke'. This facility allows the input of characters through the use of specific "gestures" which can be drawn on the screen. Here's how you install it.First you need to install a few dependencies. You can do this by simply opening up the Raspberry Pi Console and entering this command:

```
sudo apt-get -y install build-essential libxft-dev libxpm-dev libxtst-dev
```

Image Source: OzzMaker

Now that you have done this you can download, compile, and install xstroke by executing the following commands:

```
cd ~
wget http://mirror.egtvedt.no/avr32linux.org/twiki/pub/Main/XStroke/xstroke-0.6.tar.gz
tar xfv xstroke-0.6.tar.gz
cd xstroke-0.6
./configure
sed -i '/^X_LIBS = / s/$/ -lXrender -lX11 -lXext -ldl/' Makefile
make
sudo make install
```

TOP TIP

4

XSTROKE GESTURES

Having installed xstroke, you may wish to make it easy to access from the GUI of the Raspberry Pi. Therefore, you may want to add a couple of menu shortcuts to start xstroke:

```
wget https://github.com/adafruit/PiTFT_Extras
/raw/master/xstroke.desktop
wget https://github.com/adafruit/PiTFT_Extras
/raw/master/xstrokekill.desktop
sudo cp xstroke*.desktop /usr/share/applications/
```

MAKE YOUR OWN SPORTTRAX

1

GETTING STARTED WITH YOUR PI

In order to get started building your very own SportTrax you will need a Raspberry Pi. For this project you may use any model you wish. However, it's important to note that the casing has been designed for a Raspberry Pi 2 - with a few tweaks you could use any model you wish. For this project you will also need to install the latest version of Raspbian Jessie onto a formatted Micro SD Card for the Raspberry Pi. Simply plug your Micro SD Card into your computer and wait for the storage device to appear. To prepare the SD card for use within the Pi we used an Mac OS X ActionScript called 'Pi Filler' - http://ivanx.com/raspberrypi/. Once you have downloaded the script, open the application and follow the on-screen steps to flash the Raspbian image onto the SD card. It should be mentioned that you may also flash an SD Card using the Fedora ARM Installer for Windows computers.

Once you have finished preparing your Micro SD Card, you will want to place it in the SD Card tray of the Raspberry Pi. Then, plug your Raspberry Pi into an Ethernet Cable, HDMI Connection (optional) and plug in the Micro USB B power supply. You should see a solid red PWR LED (if using RPi2) and a flickering green ACT LED. This indicates your Pi is booting, you should also see yellow and green LEDs light up on the Ethernet port. If you have connected an HDMI with a monitor, you should initially see the Raspberry Pi logo. Then, once the console text has finished, scroll the screen down and you will see it prompts you to enter a username and password.

Once your Raspberry Pi's console has loaded you may login, by default the username and password are 'pi' and 'raspberry'. Upon your first boot a utility called Raspi-Config will open within the console giving you a menu-based interface. This configuration menu is very similar to that of a computer's BIOS settings. You will want to select 'expand_rootfs' from the menu to expand the file system of Raspiban to use the Micro SD Card completely. There are many other options that you may wish to alter, but for now that is all we need to do until later. Once you have gone through the 'expand_rootfs' procedure, the configuration menu will prompt you to reboot. Once rebooted, your Pi should boot direct to the Raspberry Pi's GUI thanks to the new default settings in Raspbian Jessie.

VIDEO INTRO

COMPONENTS

For this project you will require a number of components to get started:

- Raspberry Pi 2 (or Zero) - £25
- Micro SD Card (>4GB)
- Adafruit 2.8" TFT Touchscreen Display - £22
- Adafruit 1/4 Size Perma-Proto Breadboard - £7.50
- Adafruit PowerBoost 500c - £13.50
- 4400 mAh Lithium Ion Battery (interchangeable) - £13.50
- GlobalSat BU-353S4 - £30.00
- Adafruit Blue LED Metal Toggle Switch - £6

GETTING DOWN TO THE HARDWARE

INSTALLING YOUR GPS TRACKING MODULE

For this project we used a GlobalSat BU-353S4 as our GPS module of choice. It operates as a simultaneous receiver of 48 satellites (48 channels) which allow for precise geolocation. You may use any module you wish as long as it is a serial device using USB that broadcasts the NMEA data as $GPRMC sentence code - if not you will need to alter the parsing program. You will also need to know the BAUD rate (rate of frequency modulation) of your device. The model that we are using operates at 4800. In order to ensure that your GPS Tracker is emitting the correct data you may also plug the GPS Module into the Raspberry Pi and boot the device. Once started, login to the console and use the command:

```
tail -f /var/log/syslog
```

In turn the Raspberry Pi's console will display the system's log which should include (if the GPS module is working) a device connected which may appear as 'pl2303' listed under '/dev/ttyUSB0' (may differ on your own device). Now we may set the port's BAUD rate, by executing the command:

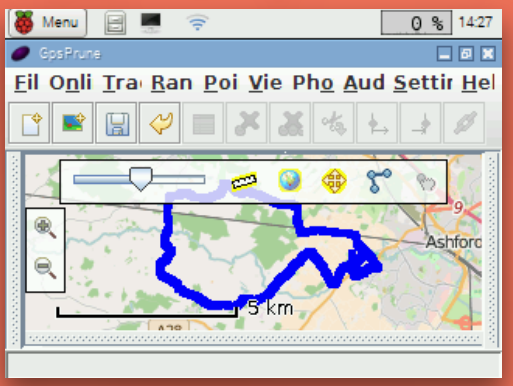
```
stty -F /dev/ttyUSB0 4800
```

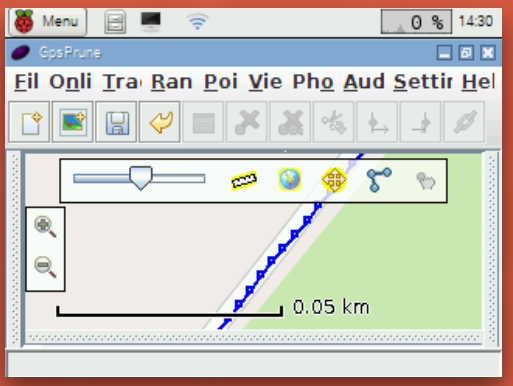
Should your device not be in the $GPRMC format, you may wish to check which format it emits the data in using the command:

```
cat /dev/ttyUSB0
```

If you do so, you will see the raw live stream of data coming from the module itself. Each of these lines of data are NMEA sentences which are a standardised format for communicating GPS data. Below you may see detailed information regarding what the array of $GPRMC NMEA data translates to if you wished to design your own program or alter its functionality, as discussed in the next section.

Sample Data	Description	Format
$GPMRC	Protocol Header	---
22220516	Time Stamp (UTC)	'hhmmss.sss'
A	Validity	'A' - ok, 'V' - void
5133.82	Current Latitude	'ddmm.mmmm'
N	North/South	'N' or 'S'
00046.34	Current Longitude	'dd.mm.mmmm'
W	East/West Hemisphere	'E' or 'W'
173.8	Speed (in knots)	'xxx.x'
231.8	True course	'xxx.x'
130694	UT Date Stamp	'ddmmyy'
004.2	Magnetic Variation	'xxx.x'
W	East/West Hemisphere	'E' or 'W'
*70	Checksum	'x*xx'





A screenshot of GPS Prune with our sample data, showing a high degree of accuracy.

Mark Lane

PRODUCT DEVELOPER

Mark is a student progressing within computer science at Homewood. During his free time, he operates as a freelance web developer and programmer whilst a part of the Apple Developer Program. He also has a passion for working with electronics and producing fun projects using the Raspberry Pi.

MarkLane825@gmail.com

@LaneMarklane825

GPS PRUNE

PARSING DATA

In order to log and parse our data for viewing through 'GPS Prune', we must run a custom python program. This may be found at our GitHub respitory or may be simply downloaded by using the following commands on the Raspberry Pi's console.

```
cd Desktop
wget https://github.com/SportTrax/RPi-GPS-Logger/archive/master.zip
unzip master.zip
```

Once you have extracted the program and log file structure, you will need to setup the Raspberry Pi to automatically log the data upon boot. To do so execute:

```
sudo nano /etc/rc.local
```

Then add the following line to the file before the 'exit 0' line.

```
python /home/pi/Desktop/gps.py &
```

From this point onwards, whenever your Pi boots all the data from the GPS will be logged and parsed.

VIEWING GPS DATA

In order to view the parsed logged data from the Raspberry Pi's GPS Module, you may use a library called 'GPS Prune'. In order to install the program execute the following command:

```
sudo apt-get install gpsprune
```

You may then open the utility from the Raspberry Pi's menu. Once opened, click File --> Open and select the simple (parsed) log file of your choice. Alternatively, if you wish to use another utility, you may use the raw data logs.

POWER DISTRIBUTION

To ensure the entire device is powered, we have used a 1/4 size Perma Proto Breadboard which acts as a distribution point, that has been powered by the Adafruit PowerBoost 500c. This particular model will provide enough power for the Raspberry Pi, however, if you modify the project and require more power you may wish to use the 1000c. On the board itself we soldered connections from the '5v' and 'G' of the PowerBoost to the positive and negative terminals of the Perma Proto breadboard. From these terminals we connect two female-to-female jumper wires to the GPIO of the display. Then, for the power switch we wire two arcade jumper wires to the common and normally-closed terminals and the 'EN' and 'GND' pins on the PowerBoost board. Finally, we use two additional arcade jumper wires to connect the LED of the switch to the positive and negative terminals to the PowerBoost. Optionally, you may wish to wire six additional jumper wires from the Display's GPIO Pins in a format of your choice for a Serial connection to the console of your Raspberry Pi during development. All of the pins are labelled on the casing for a FTDI Friend.

PROTOTYPING

Andrew and Will setting forth on their journey with the SportTrax on board.

When it came to creating our first prototype there was a lot of planning involved. We started by initially taking measurements of the Raspberry Pi itself so that we would know the general sizing and shape our case would have to take in order to house our creation. After knowing this, we used calipers to take the precise measurements of the ports on the Pi such as the HDMI and USB ports, including the tactile switches which were implemented via the 2.8” Adafruit touchscreen. Each of these tactile switches are connected to an individual GPIO pin which we programmed in Python (3.4), so the pi detects when the loop is closed upon pressing the momentary switch – see pages 3-4 for setup. We then used the precise measurements that we collected to produce our first card-based prototype case – which may be seen on the next page.

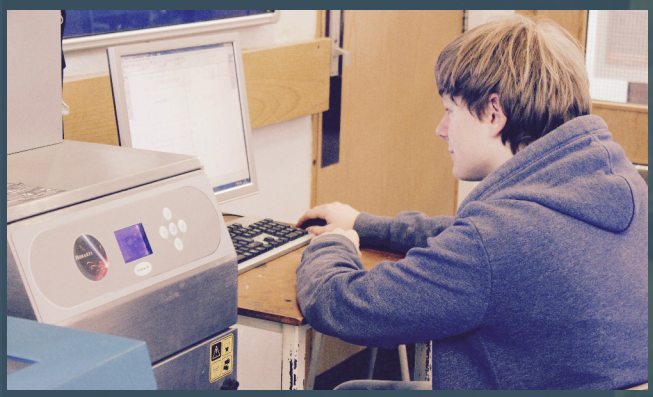
The process of using a card-based prototypes allowed us to see our designs come to life and find any potential flaws prior to producing the final acrylic design. This required a substantial amount of design time due to the complexity within the internals of the case. This is because the internals of the case were designed allowing wiring to pass through the different divisions of the case, which were required to ensure all hardware could be mounted properly, and securely, to ensure its safety during transit. In order to be able to map our design we used a program known as 2DDesign which allows us to plot the schematics of a case with a high-level of precision to control the laser. You can see a photograph next to this article of James processing a schematic prior to sending the data to the laser.

In terms of the physical material, we used clear Perspex as it allowed us to easily see all the components during development, although when cutting your own casing you may wish to use your own colour of Perspex which suits your taste. On the casing we have also engraved labels onto each of the ports of the device to ensure it is as user-friendly as possible. This includes a ‘Developer Section’ which consists of the ports that you may require if you wish to further develop the product yourself, including access to a serial connection point to the Pi’s GPIO to access the device’s console. Once you have cut out the device schematics you will nee to follow the top (‘T’) and bottom (‘B’) lettering on the end slides to clip the two ends of the case together. We have made this proces as easy as possible through the use of clips which are cut into the Perspex itself, streamlining the manufacturing process.

SportTrax

Cyclist Team

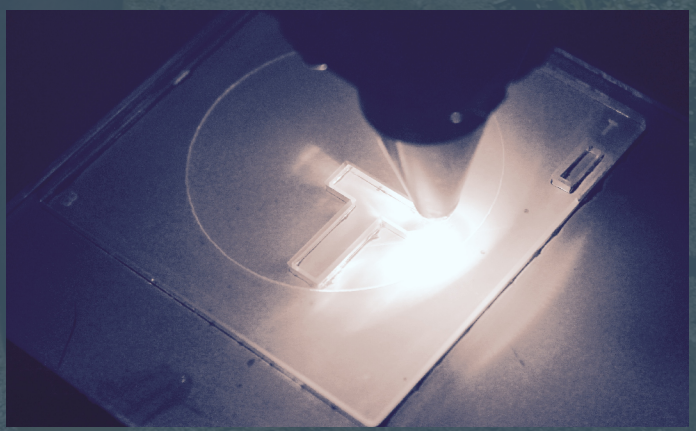
Andrew Bailey and William Shannon



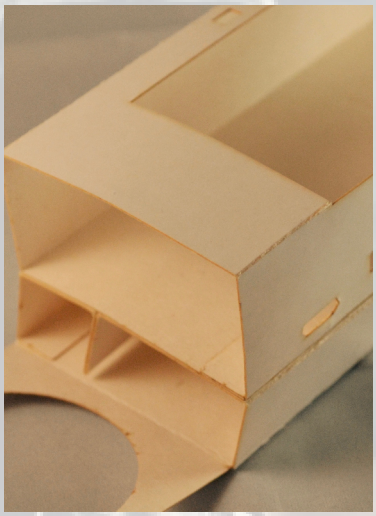
James assisting with prototype casing development. You may find the designs in our GitHub respitory.

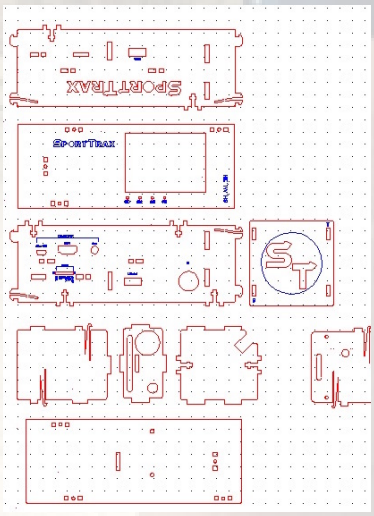
TESTING THE PRODUCT

To ensure the product was ready for production we asked two of our close friends and colleagues if they would be happy to test the SportTrax device. Thankfully, they accepted and took our device on a 25 km bicycle ride, thus, providing us with sample data for testing the mapping software (GPS Prune) to its full extent and capabilities. While testing we were also able to test the product's durability within its bespoke casing throughout its maiden journey. You can watch the cases assembly in our video introduction - http://sammyherring.co.uk/st-vid.



Here we may see the laser in action, cutting and engraving the side panel of the SportTrax casing.





STAGES OF PRODUCT DEVELOPMENT

During this project we have been through many stages of development. As you can see we initially produced a card-prototype in order to correctly size the Raspberry Pi within the final case. Once we had finished taking our measurements, we went on to produce a full-fledged net for the case, which was designed to work with 3mm Acrylic (Perspex). In the last photograph, you can see the completed product.

VIDEO INTRO

sammyherring.co.uk/st-vid

SPORTTRAX
THE BUILD GUIDE

sammyherring.co.uk/st-respo