Using your phone or a microcomputer to sense data streams as you walk.

The easiest and most obvious way to sense many data streams is to use your phone. Most smart phones have plenty of sensors, but they don't always allow you to see the data directly. For example, your phone changes its screen orientation as you move it, but it doesn't give you a minute by minute readout of its rotation vectors.

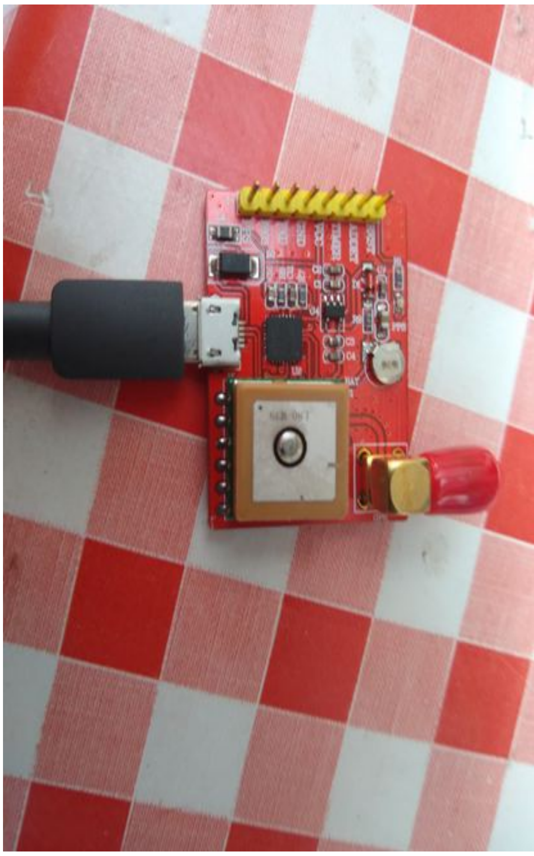
To get more access to this data, the OSCHook app can be downloaded free from GooglePlay for Android phones. It allows you to see the data coming from various sensors on your Android phone: these include the accelerometer, gravity, light, linear acceleration, rotation vector, GPS speed and position, proximity, compass orientation, and signal strength of the four closest Wifi networks. Each is displayed on the screen. However, you would need a good knowledge of Android programming to do anything other than look at them individually on the screen.

So if you want a system to analyse, store and display data about your walk, either in real time or afterwards, you need to go beyond the phone.

The best option I have found so far is to use a Raspberry Pi 3B microcomputer, which is currently available from Amazon for about £32. (Avoid earlier models, which come without built-in wifi.) If you have never used a Raspberry Pi before, they are a little complicated to set up, so go to a local class, or buy a good beginners' guide. The Pi does not come with a screen or a mouse or a keyboard, so you have to plug these in, using your TV as the screen, until you have the initial setup done.

Once you have set the Pi up, you can control it remotely using RealVNC software. There is a free version of this specially for the Pi – see: https://www.realvnc.com/en/raspberrypi/ This site also explains how to enable the VNC server on your Pi, and then to install a VNC viewer on your laptop or mobile. Once this is set up, you turn on the Pi (by connecting it to a power source), log in to it from your phone or laptop, and you can then see and work on the Pi screen.

Of course this requires a wifi link for your Pi. If you are a psychogeographer using your system away from civilisation and a wifi network, you need to set up your phone to act as a wifi 'hot spot'. On Android, go to 'settings', and then in 'network' or 'internet' choose hotspot and tethering, and turn on the hotspot. This is now a mini wifi station. Set up your Pi so that it will connect automatically to the hot spot network. (It doesn't matter if this network doesn't have internet access: you are only using it to let your phone talk to your Pi.) You only need to use this to turn the Pi on and set the sensor programme going; after that the Pi will look after itself. Later on you can link to the Pi again and download the data files is has saved.



Now for the sensors. It is possible to buy all sorts of sensors separately, and wire them in yourself, but this is too much like hard work. It also carries the risk of burning out the sensor, or the Pi itself, if you get the soldering wrong.

The Raspberry Pi Sense Hat is available from Amazon for about £30. This contains many of the same sensors as your phone. The 'Hat' format means that it simply plugs in to your Raspberry Pi, with no need for wires or soldering. There are very comprehensive instructions at https://pythonhosted.org/sense-hat/api/ - you can be using it within seconds of plugging it in.

The Sense Hat does not include a GPS sensor. You can buy these separately from Amazon. I bought a USB Port GPS module which cost about £30. (See details here: http://wiki.52pi.com/index.php/USB-Port-GPS_Module_SKU:EZ-0048). This has the great advantage that it connects to one of the USB sockets on the Pi, so again there is no need for soldering.

To power all this you will need a solid battery: I used an Anker PowerCore 20100. These are usually sold for charging mobile phones and tablets. It takes a few hours to charge up, but once charged will easily power everything for a couple of hours.

Programming all these to act together was quite complex. The Rasberry Pi comes with its own version of Linux, Raspbian, which seems a little 'geeky' to anyone used to graphical interfaces like Windows or Mac. Luckily Python and the IDLE IDE are built in, so they are easy to use, but integrating the GPS data requires using a little Raspbian to turn the board on and off and to read the incoming results. (This is explained in https://wiki.52pi.com/index.php/USB-Port-

GPS_Module_SKU:EZ-0048, and also in the 'man' pages for some of the commands.)

I wrote a Python programme to set everything going, and then to collect all the data from the GPS and Sense Hat sensors. It builds a line of data roughly every 20 seconds and writes this to a file for use later on. It also uses the 8X8 LED matrix on the Pi Sense Hat to show you an interpretation of the results 'in real time'.

If anyone wants a copy of this programme, with the usual provisos about not holding me responsible for anything, email me at cogible@gmail.com

The most challenging aspect of the project is not the technical side. As MI5 have discovered, in a different context, it's relatively easy to amass a lot of data. The question is what it means, and what to do with it.

I've developed a modular system of encoding different viewpoints. Each set is a chunk of software that can be swapped in or out of the main programme. The data points are used to set the positions, sizes, and colours of shapes on the LED screen, or to print out patterns on my laptop. But which data points do you use for what? And, a more subtle problem, how do you calibrate them, to take into account varying conditions?

For example a walk on the Equator might give a very different range of temperatures to a walk in Yorkshire. If you view all the possible temperatures, then your results over one day will show little variation during a walk. If you focus down on a narrow band of temperatures, your results may show much more interesting local temperature variations.

Designing and playing with these viewpoints is by far the most interesting part of the project so far. My next target is to find or to build more sensors for new data lines.

David Upton
4th World Congress of Psychogeography, 2019