



Geolocate yourself

For this activity you require access to a Raspberry Pi (plus peripherals) and a GPS dongle. We recommend [this one](#), which is not too pricey.

A calamity of epic proportions is underway, and you and your team are the only ones who can save us! Can you create something that will protect the world during the apocalypse? What about something to avert the apocalypse altogether? Time to get your survival kits out and your thinking hats on, as you #MakeYourIdeas and save us all!

We've just received this message:



rpf.io/pioneers

During the apocalypse, many terrestrial forms of communication have become unreliable or have broken down entirely. Hopefully, the satellites of the Global Positioning System (GPS) are still live, because you could use them to navigate and find your way to safety.



#MakeYourIdeas

For more inspiration and help, head to rpf.io/pioneers

To do this, you'll need a Raspberry Pi with an internet connection and a USB GPS Dongle.

During the apocalypse, many terrestrial forms of communication have become unreliable or have broken down entirely. Hopefully, the satellites of the Global Positioning System (GPS) are still live, because you could use them to navigate and find your way to safety. To do this, you'll need a Raspberry Pi with an internet connection and a USB GPS Dongle.

Most GPS receivers talk to your computer via **serial communication**. They use a specific port on your computer to do this, and you will need to find out which port your receiver is using.

With the receiver **not** connected, open a terminal window and type the following command:

```
ls /dev/ > dev_list_1.txt
```

When you press `Enter`, it will look like nothing has happened, but actually you have just created a text file called `dev_list_1.txt` containing the names of all the devices connected to your computer in the `/home/pi` folder.

Now **plug in** your GPS receiver, type the following line into a terminal window, and then press `Enter`. The command will display the names of any newly connected devices in your terminal window.

```
ls /dev/ | diff --suppress-common-lines -y - dev_list_1.txt
```

You should see a list similar to this:

```
serial
ttyACM0
```

The device you're interested in is the one with a name starting with `tty`, so in this example it is `ttyACM0`. Write the name of your device down here or somewhere else you can easily find it again.

GPS device name:



Raspberry Pi

#MakeYourIdeas

For more inspiration and help, head to rpf.io/pioneers

To make sure the device can be read, type the following into the terminal window, and press `Enter`.

```
cat /dev/ttyACM0
```

You should now see a continuous stream of data from the device, looking similar to this:

```
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,,,,,,,,N*30
$GPGGA,,,,,0,00,99.99,,,,,*48
$GPGSA,A,1,,,,,,,,,99.99,99.99,99.99*30
$GPGLL,,,,,V,N*64
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,,,,,,,,N*30
$GPGGA,,,,,0,00,99.99,,,,,*48
$GPGSA,A,1,,,,,,,,,99.99,99.99,99.99*30
$GPGLL,,,,,V,N*64
```

The data you see in your terminal window contains your longitude, latitude, altitude, and the current time. You can use Python to make a little more sense of this.

Firstly, open a **new Terminal window**, and download and install the `pigps` module this program needs to run by typing:

```
sudo pip3 install pigps
```

If you're asked to proceed, press `Y`. Wait for the program to install.

Now you can write a script that will gather the information from your GPS dongle. Type into the terminal:

```
nano gpstest.py
```



You've just created a new Python file called `gpstest.py`. You should now be looking at a black screen with the name of your file at the top in a grey band.

Type into the terminal:

```
from pigps import GPS
from time import sleep

gps = GPS()

while True:
    print(gps.lat)
    print(gps.lon)
    print(gps.time)
    print(gps.alt)
    print("\n" * 5)
    sleep(5)
```

Make sure all your code is correct, and press `Ctrl+O` and `Enter` to save your work.

Run the code in your terminal window by typing `python3 gpstest.py`. You will see text appear, and the text will change every five seconds. Once a few cycles of information have appeared, press `Ctrl+C` to kill your program so you can more easily read the text. Your latitude and longitude will be long numbers with 14 decimal places!

Your output will look a bit like this:

```
[datetime.time(##, ##, ##), (Latitude), (Longitude), (time),
(altitude)...]
##:##:##
[datetime.time(##, ##, ##), (Latitude), (Longitude), (time),
(altitude)...]
##:##:##
[datetime.time(##, ##, ##), (Latitude), (Longitude), (time),
(altitude)...]
##:##:##
(Latitude)
(Longitude)
(time)
(Altitude)
```



Raspberry Pi

MakeYourIdeas

For more inspiration and help, head to rpf.io/pioneers

Your latitude and longitude will be long numbers with 14 decimal places! These numbers describe your exact, unique location on the globe. Write them down here, or wherever you wrote down the name of your GPS device.

Lat:

Lon:

You have just learnt to **get your geolocation using a GPS device and a simple Python script.**

You've done it! This message has just come through:



Visit rpf.io/pioneers to find out more about how only **you** can save us!

Take it further

For some more awesome ideas on how to make things with the GPS dongle, have a look at these projects on the Raspberry Pi website:

[Poo Near you](#)

[Zombie Apocalypse map](#)

[Where's Zombie?](#)



Raspberry Pi

#MakeYourIdeas

For more inspiration and help, head to rpf.io/pioneers