

Raspberry PI GPS Recorder

V3.01.004

Jed Marti KI7NNP

January 1, 2025

Abstract

A Raspberry PI GPS hat is used to collect GPS data in the field. The program can output CSV, raw NMEA packets, and KML. This manual describes setup and results.

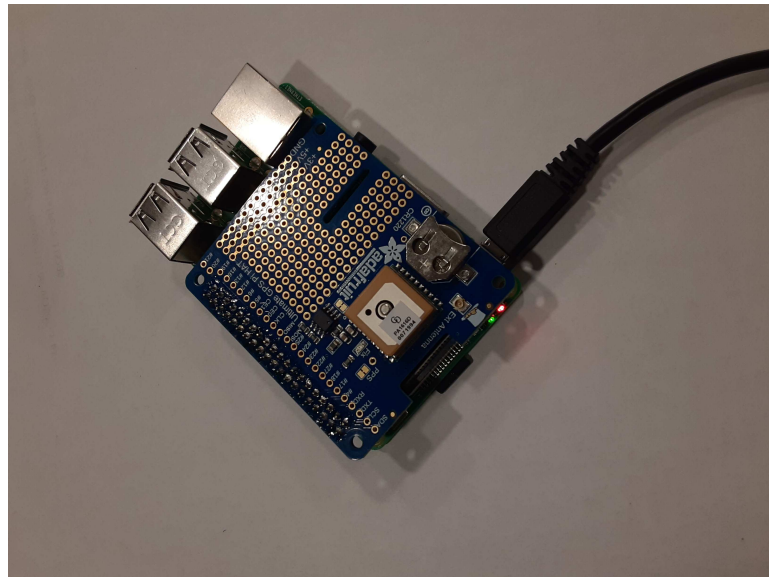
1 Introduction

Described is a program to use a Raspberry PI GPS hat to record GPS positions in the field in 3 different formats.

CSV Records th time, latitude, longitude, speed and true course.

NMEA Records all NMEA data from the GPS device in raw form.

KML Records the locations, time, speed and course in KML format for browser display.

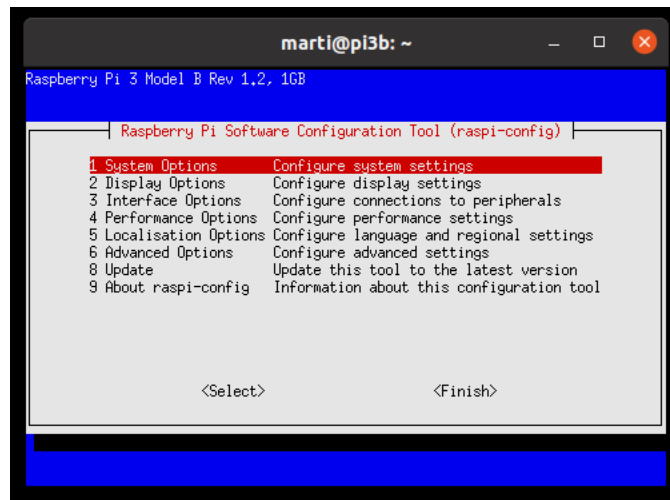


2 PI Serial Port

You must disconnect the OS from the serial port `/dev/ttyS0` or `/dev/serial0` - they both name the same device. At the same time, you might enable the Bluetooth interface if you want to communicate through it. Using either a WiFi connection (shown here) or through a window and keyboard.

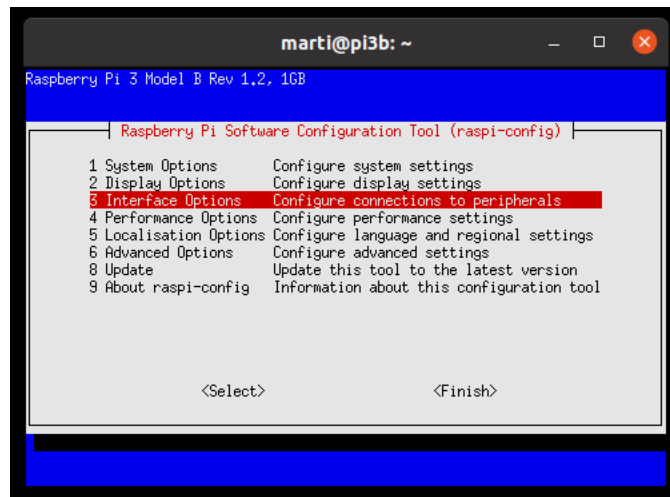
```
sudo raspi-config
```

You should see:



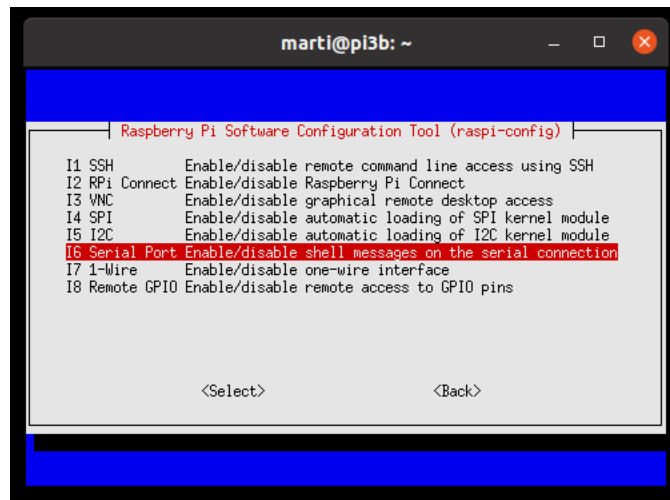
The configuration tool front page.

Use the arrow keys to move to number 3 **Interface Options**.



Configuration tool selecting Interface Options

Hit return, and you should see the following page. Use the arrow keys to highlight I6, **Serial Port** ... as shown.



Interface Options page, changing serial port.

The first page asks about the login shell. You don't want this on. Use Tab to switch to <NO> and hit return.



Make sure login shell is disabled.

You want the serial port hardware enabled. Use Tab to highlight <Yes> and hit return.



Serial port must be enabled.

Then you should see the following. Hit return.



The serial port will be set

Finally, use Tab to highlight <Finish> and hit return. You should then reboot.

3 Download Program

Download from www.cog9llc.com/gps_pi.tar. From a command window enter:

```
tar xvf gps_pi.tar
cd gps
sudo make install
```

This will copy the program to `/usr/local/bin`. Or you can just copy `gps/gps` to your preferred directory.

Next you need to setup the Pi's serial port to disconnect it from the operating system.

4 How to run

From the command line:

```
gps -h
gives you:
```

```
KI7NMP GPS server V3.01.004
gps V3.01.004
  -b -bad n          How many bad chars to tolerate (5)
  -brate n          Serial port baud rate (9600)
  -c -csv fname     Write CSV file
  -k -kml fname     Write KML file
  -n -nmea fname    Write NMEA raw file

  -serial dev       Serial port device name (/dev/serial0)
  -w -where double  Write location to console if more than value
                   change. <0.0 don't show (32.804)
```

The following options are available.

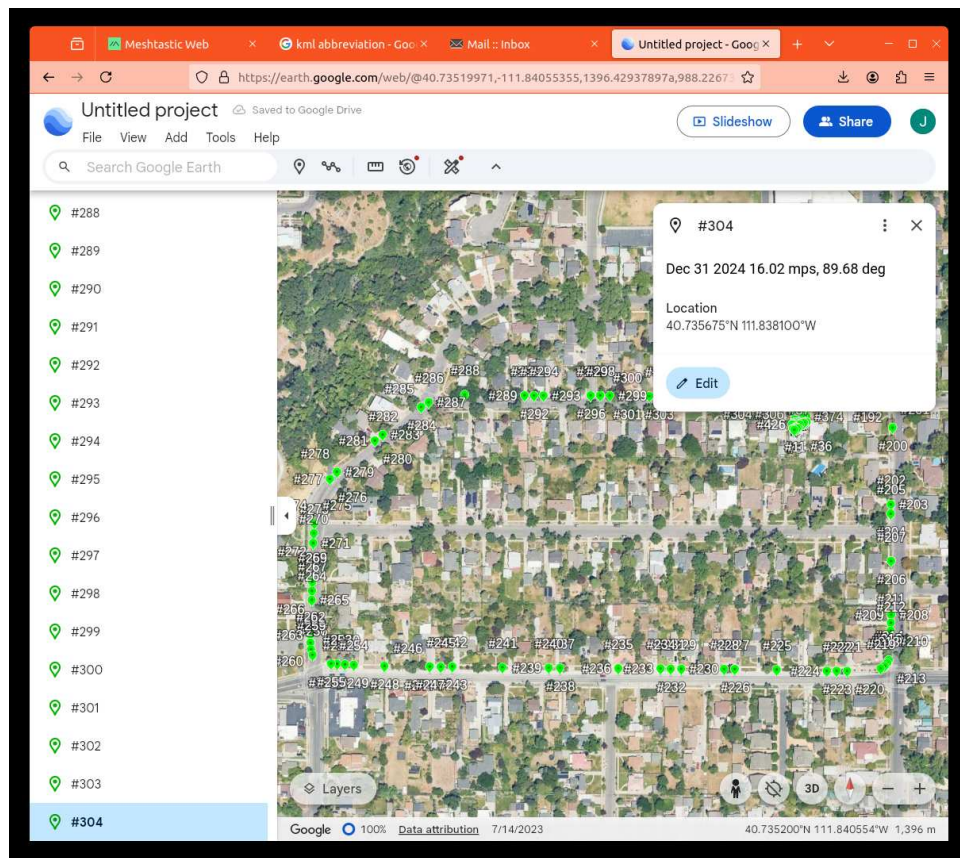
- bad or -b *number*** How many bad characters are allowed between NMEA messages. Exceeding this number is a sign that you haven't set up the PI serial port correctly or you have a noisy power supply. By setting this value high, the program will continue running but will be missing packets. The default is 5.
- brate *number*** Set the GPS baud rate. The accepted numbers are 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200. The system does not change the GPS device rate which defaults to 9600 baud.

-c or -csv file-name Generate a CSV (comma separated variable) file. This will have 6 columns.

Label	Description
TIME	Time in seconds since 1970.
LONGITUDE	The longitude in degrees from $-180.0 \rightarrow 180.0$.
LATITUDE	The latitude in degrees from $-90.0 \rightarrow 90.09$.
ELEVATION	The MSL elevation in meters.
SPEED	The current speed in meters/second.
COURSE	The current course in degrees true.

-k or -kml file-name Generate a **KML** file for each location point. Each point is numbered and its time, speed and course are included in the <description> field. A special small Placemark is referenced from the website.

If the program loses control from power loss, you may need to add the KML terminator before loading into Google Earth or other display program.



-
- n or -nmea *file-name*** Write the raw values read from the GPS device to the file name given.
 - serial *device*** Change the serial port device. Normally this is `/dev/serial0`.
 - w or -where *number*** If a positive number is given, write the longitude/latitude to the program window that started the program but only when the distance traveled from the last displayed point exceeds the number of feet given.

5 Raspberry PI Bluetooth Terminal

To communicate with your PI in the wilds you can establish a simple connection through bluetooth. You won't be able to use the serial port as the GPS is using those serial lines.

1. Edit `/etc/systemd/system/dbus-org.bluez.service`
2. Add `-C` and the next line

```
ExecStart=/usr/lib/bluetooth/bluetoothd -C
ExecStartPost=/usr/bin/sdptool add SP
```

3. Save and reboot.
4. Start terminal server (you may need to do this each time you reboot).

```
sudo rfcomm watch hci0 1 getty rfcomm0 115200 vt100 -a pi
```

There are a number of terminal programs that will then communicate through Bluetooth to a restricted console set. You can start the program and turn it off with this method though clumsy.

6 Bare Bones Recording

1. Connect PI to battery power in range of a WiFi connection.
2. Log into the PI through WiFi.
3. Start the program and wander off to where you will.
4. On return reconnect and $\uparrow C$, the program will terminate and the KML epilog will be written.

5. If the battery dies during recording you will miss the KML epilog. You will need to add the following two lines at the end of the file.

```
</Document>  
</kml>
```