

# GPS reader project

The GPS reader project reads (time, position) information from an external GPS device connected to a Raspberry PI.

## Getting Started

The GPS reader is program, coded in Python, used to read GPS information from a **Holux M 1200E GPS data logger**. The GPS data logger is connected to the Raspberry Pi through the USB port and the information is retrieved by reading the serial port.

For this project, you should be familiar with the basic use of [Raspberry Pi](#).

Before running the project you will need a Raspberry Pi ready to use, with ssh enabled, connected to the same network of your PC.

You also need to place the GPS data logger outdoor in order to retrieve satellite information from, at least, three satellites.

## Requirements

### 1. Hardware requirements

- **Single-board computer:** Raspberry PI 3 or Raspberry Pi Zero.  
In case of using a Raspberry PI 3, an Ethernet cable is needed to

connect the device to the PC for remote access.

- **GPS Data logger:** “Holux M 1200E GPS data logger” with the cable used to connect it to the Raspberry [Pi.In](#) case you use a Raspberry PI Zero you need a conveter USB <-> micro-USB to connect the data logger.

## 2. Software requirements

- **Operating System:** last version available at [Raspberry Pi website](#). The version used is “2018-11-13-raspbian-stretch.img” (updated and ready to use)
- **Programming language:** Python3.
- **Other tools:**

## 3. Raspberry Pi Configuration requirements

- **Remote access:** ssh enabled

## 4. Networking options

- **Raspberry PI 3:** Interface Eth0 connected to the PC.
- **Raspberry PI Zero:** Raspberry PI and PC in the same wireless network.

# Installation guide

## I. GPS data logger connection

Start by enabling the collecting of GPS information. For this, proceed as follows:

1. Place the GPS data logger outdoor.
2. Connect the GPS data logger to the Raspberry PI through an USB port, or micro-USB (in case of using a Raspberry Pi Zero).
3. Keep the GPS data logger turned off.

## II. Access the Raspberry PI

Now that, the GPS data logger is ready to work, it's time to boot the system and remotely access the Raspberry Pi. For this, proceed as follows:

1. Boot the Raspberry Pi by connecting it to a power supply.
2. Connect the Raspberry PI to ypur PC, as follows:
  - For Raspberry PI 3, connect them through Ethernet.
  - For Raspberry Pi Zero be sure that both PC and Raspberry Pi are part of the same wireless LAN.
3. Check if there is connectivity between the Raspberry PI and your PC. For this:

```
$ ping raspberrypi.local
```

4. Remotely access the Raspberry Pi by ssh . After digitising the command, upon request, digit "Yes" and enter the Raspberry Pi password.

```
$ ssh pi@raspberrypi.local
```

```
....upon request, digit "Yes"
```

```
.... upon request, enter the Raspberry Pi password.
```

## III. Installing the software

1. At your Raspberry Pi, clone the git repository **location/gps**.

```
$ cd /home/pi/  
$ git clone https://github.com/TeresaMVazao/VehicularNetworks/sensors/location/gps location/gps
```

2. Check if the Raspberry Pi contains the required of software:

```
---> /home/pi/code-examples/location/gps  
+-----gps-reader.py
```

# Deployment

## I. System status check

Now, the system is ready for deployment. Before, doing it, perform a preliminary check of the operational conditions.

1. Check if the GPS data logger has the battery light green (is powered up)
2. Check if the Raspberry PI is accessible by detecting if:
  - The boot is completed
  - There is an active remote access session between the Raspberry PI and your PC.

## II. Deployment and validation

Now, the system is ready for deployment. For that:

1. Turn on the GPS data logger and check the appearance of the other lights:

- **Blue light** - to signal the availability of a Bluetooth network (not used)
  - **Orange light** - to signal the search for visible satellites.
2. At the Raspberry Pi, run the program. For this, proceed as follows:

```
$ cd /code-examples/location/gps  
$ sudo python3 gps-reader.py
```

3. Check the output at the Raspberry PI terminal. In case it succeeds (time, latitude, longitude, speed,...), among others must be presented.

## Authors

- **Teresa Vazao** - *Initial work* - [SmartMob@Tecnico](#)

See also the list of [contributors](#) who participated in this project.