How to use Raspberry Pi to monitor Broadband speed

In this article, we will show you how to use the Raspberry Pi to monitor the broadband speed. This project involves a bit of a fair setup, including downloading and installing several packages, writing Python scripts, etc.

Once you have the basics set up, you can automate monitoring of your broadband connection, so it continues for as long as you like.

A CSV file will be generated (comma separated values) into Google Drive and updated once an hour. CSV is a very simple file format that can be opened in Microsoft Excel or imported into Google Sheets. Sounds like a handy thing? This article assumes you have Raspbian installed on your Pi.

Use Raspberry Pi to monitor Broadband speed

Step 1: Check for updates

As always, start with checking for the latest updates. Run the following commands in Terminal:

```
sudo apt-get update sudo apt-get upgrade
```

Step 2: Install speedtest-cli

There are different methods to measure a broadband connection speed. We'll use **speedtest-cli** , 'a command line interface for checking Internet bandwidth using speedtest.net'.

speedtest-cli is not available right from the Raspbian repository, but you can install it from the Python Package Index (PyPI). That's easily done using a tool called pip that comes pre-installed on Raspbian Jessie and Stretch. You can ensure that you have pips by running this command:

```
sudo apt-get install python-pip
```

If you get the message 'python-pip is the latest version', that means it's ready for use.

Next, use pip to install speedtest-cli:

```
sudo pip install speedtest-cli
```

With speedtest-cli now installed, you can measure your broadband speed easily with the following command:

```
speedtest-cli
```

However, for the purposes of this article, it is more convenient to use the Simple mode of speedtest-cli:

```
speedtest-cli --simple
```

You should see something like this:

```
Ping: 47.943 ms Download: 40.93 Mbit/s Upload: 2.33 Mbit/s
```

However, that output does not follow the CSV syntax. Hence, you will need to parse the data and get it right.

Step 3: Create Python script

Let's create a new Python file:

```
sudo nano speedtest.py
```

Here's what you should have inside the file (you can of course copy and paste these lines):

```
import os import re import subprocess import time response = subprocess.Popen('sp
```

(Assuming you are saving the script in **/ home / pi /** - otherwise just change the path here: **if os.stat ('/ home / pi / speedtest / speedtest.csv'). St_size = = 0:)** .

After you have the lines in the right place, you can save the file and exit the editor by pressing **Ctrl + X, Y** and **Enter** .

Script run speedtest-cli in Simple mode, parse the output and output it in CSV format. You can run the script with the following command:

```
python speedtest.py
```

And if you do, you should see a line like this:

```
10/26/17,10:18,47.943,40.93,2.33
```

Step 4: Create a directory

Let's create a directory for the CSV file:

```
mkdir speedtest
```

If you're wondering why you need a folder for a file, it's because you'll sync that folder with Google Drive. Once synced, everything inside the folder will correspond to the content of the Google Drive folder.

Now, if you run your Python script like this:

```
python speedtest.py >> speedtest/speedtest.csv
```

… You will have a CSV file with broadband speed data in the new folder.

If you check the contents of the file (e.g. **cat speedtest / speedtest.csv**), you can see lines like this:

```
Date,Time,Ping (ms),Download (Mbit/s),Upload (Mbit/s) 10/26/17,10:18,47.943,40.9
```

Step 5: Integrate the script with Google Drive

To integrate the script with Google Drive, you can use the Google Drive CLI Client of user GitHub Petter Rasmussen. Download its Raspberry Pi version by running the following command:

```
wget -O gdrive https://docs.google.com/uc?id=0B3X9GlR6EmbnVXNLanp4ZFRRbzg&export=
```

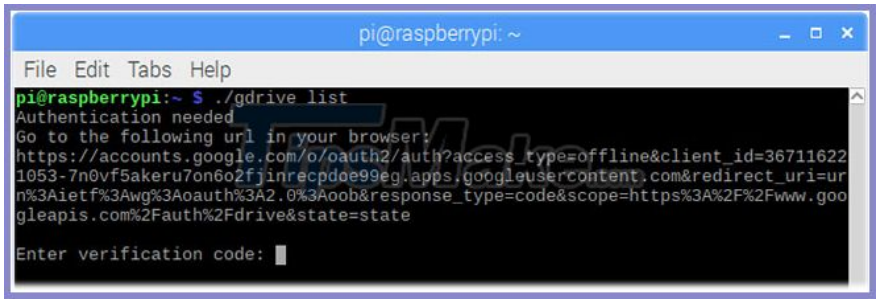
Next, let's specify file permissions:

```
chmod +x gdrive
```

You need to notify Google Drive to allow Google Drive CLI Client to connect to the account. It is possible to do so by running the Google Drive CLI Client with any parameters. For example, this command lists the contents of your Google Drive account:

```
./gdrive list
```

You should now see an authentication request like this:



Just follow the instructions: Visit the URL in your browser, log into your Google account and allow ' *GDrive (…) to view and manage the files in your Google Drive.* '(GDrive (…) view and manage your Google Drive files). Then you're ready to enter the verification code.

Finally, gdrive will list your Google Drive content.

Now that you have the speedtest folder on your Raspberry Pi, create a corresponding directory for Google Drive:

```
./gdrive mkdir speedtest
```

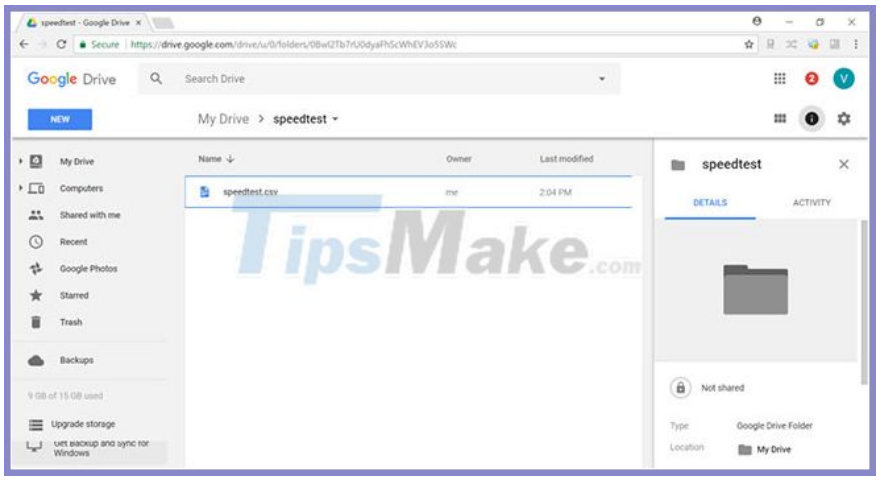
The program returns the ID of the new directory. Copy it, as you will need it later.

Next, let's sync the two speedtest folders:

```
./gdrive sync upload speedtest ID
```

Just replace the ID with the speedtest directory ID.

If everything goes as expected, you should now see a folder called speedtest in your Google Drive. In the directory, there is a previously created file (**speedtest.csv**):



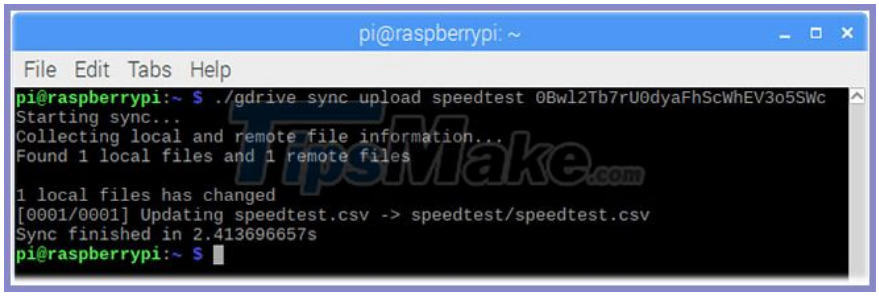
Now, if you run the Python script again, new broadband speed data will be added to the end of the file:

```
python speedtest.py >> speedtest/speedtest.csv
```

And if you run the sync command again, you can see your updated file in Google Drive:

```
./gdrive sync upload speedtest ID
```

(Again, remember to replace the ID with the ID of the speedtest directory).



You can see your updated file in Google Drive

Step 6: Automate everything

All that's left is to make things work automatically. To do this, use cron, which makes it possible to schedule commands to run at specific times, such as once an hour.

Let's create a short shell script containing commands to run once an hour:

```
sudo nano speedtest-cron.sh
```

Add the following familiar commands (Assuming you've done everything in the **/ home / pi / directory** - otherwise just change the path):

```
sudo python /home/pi/speedtest.py >> /home/pi/speedtest/speedtest.csv /home/pi/g
```

(And, again, remember to replace the ID with the ID of the speedtest directory.)

Then save and exit with **Ctrl + X, Y** and **Enter** .

Please specify the permissions to execute the script:

```
sudo chmod +x speedtest-cron.sh
```

Now you are ready to test the script:

```
./speedtest-cron.sh
```

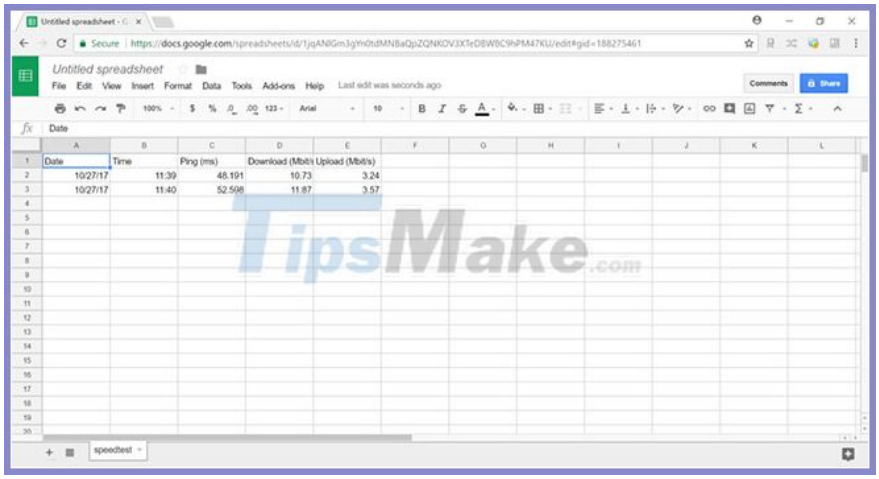
Okay, let's create the cron task:

```
crontab -e
```

Enter the following line in the editor, save and exit by pressing **Ctrl + X, Y, Enter** :

```
0 * * * * /home/pi/speedtest-cron.sh
```

Broadband speed monitoring is now available. The results of the speed test will be recorded once an hour. You can open CSV files in Google Drive using Microsoft Excel or Google Sheets. If you prefer the following option, you will have to import the file (**File> Import…**).



You can open CSV files in Google Drive using Microsoft Excel or Google Sheets

You finished reading the article "**How to use Raspberry Pi to monitor Broadband speed**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.