Sending SMS on a Raspberry Pi

19 May 2017 6 min read

blog raspberry pi terminal sms gammu

Sending SMS on a Raspberry Pi

Your Ingredients

To send SMS messages you will need: a Raspberry Pi, a USB 3G Modem and a SIM card.

As usual my Raspberry Pi came from the great ModMyPi , I have more Pis that I'm ever going to admit in public and they have all come from here. There have been a couple of problems, but they were fixed very quickly, not at my cost and in a very friendly manner. This is not a paid advert for them, I'm just a fan! Hi @ModMyPi!

I wrote some intro posts on Building a Pi on Inventability btw.

The USB 3G Modem I used was the Unlocked HUAWEI E173 from Amazon but I also used the Huawei 3G/21 Mbps Unlocked E3531 with some success as well. Unless you've already got a SIM card for a locked (meaning it will only work with one carrier) dongle, I definitely suggest getting an unlocked modem as it should work with SIM cards from any company. There are loads of other dongles out there that would do this job, but I know that these two will work.

The final piece of the puzzle was the SIM card, and I got that for free from GiffGaff and bought some pay you go minutes so that I could send SMS messages. I believe this is a UK only company, so YMMV depending on where you are in the world. I spent one whole English pound on a sim card at my local corner shop, and top it up with £5 which works perfectly well too.

The Process

Assembling Your Pieces and Parts

Take the cover off the back of your dongle, insert your SIM card, then go to your provider's website and register your new SIM card.

Finding Your Dongle

Now let's find out if our Raspberry Pi has recognised our USB modem or not. Open a new Terminal 1 either on your Raspberry Pi (if you're using a keyboard and monitor) or via SSH) from your local machine (if you're handsfree like I am).

Listing USB Devices

To find the modem we need to list all of the USB devices connected to our Raspberry Pi. Do this with the `lsusb` command. Run `lsusb` to make sure your modem is loaded. If you used a Huawei modem you will see a line like this:

```
Bus 001 Device 006: ID 12d1:1436 Huawei Technologies Co., Ltd. E173 3G Modem (modem-mode)
```

Where is our Dongle Mounted

We now need to know where on your Raspberry Pi the USB modem is mounted. So next run:

```
dmesg | grep ttyUSB
```

If you are successful you will see this in your Terminal:

```
[3.235831] usb 1-1.3.3: GSM modem (1-port) converter now attached to ttyUSB0
[3.236856] usb 1-1.3.3: GSM modem (1-port) converter now attached to ttyUSB1
[3.237626] usb 1-1.3.3: GSM modem (1-port) converter now attached to ttyUSB2
```

You will almost certainly be using the name `ttyUSB0` in the upcoming commands.

If You See Nothing

The first time I tried this process with the Huawei E3531 Dongle it didn't show up as a USB modem when I ran the `lsusb` command. So after much googling I managed to cobble together the following solution, but obviously YMMV because I have literally no idea why this worked only that it did for me!

Run `lsusb` command again and make a note of the eight numbers that appear after the word **ID**. They will be in the form `12d1:1f01` but they will not necessarily be those eight numbers, you will see that each device in that list will have an ID number using the same convention. Make a note of that number as you will need it later.

Install Packages

You will need the following packages, so run the command:

```
sudo apt-get install usb-modeswitch usb-modeswitch-data
```

Make Our Own usb_modeswitch Configuration File

Create a file called `"12d1:1f01"`, where those eight numbers are the ones you noted down from before with the following command:

```
sudo nano /etc/usb_modeswitch.d/12d1:1f01
```

The file should have the following contents: (Again you should use different numbers if your modem has different ID numbers.)

```
# Huawei E353 (3.se)

TargetVendor=  0x12d1
TargetProduct= 0x1f01

MessageContent="5553424312345678000000000000001106200000100000000000000000000"
NoDriverLoading=1
```

Now when you run the following command you should see the expected output:

```
dmesg | grep ttyUSB
```

I will confess that I have absolutely no idea why this works, but I've done it on three new installations of Raspbian on three brand-new Raspberry Pis with successful results. Obviously YMMV but if everything is working we can move onto the next step.

Installing the SMS Software

Next we need to install the software that's actually going to do the sending and receiving of the SMS messages. This software is called Gammu and we install it like this:

Installing Gammu

```
sudo apt-get install gammu
```

After installation comes the configuration of Gammu. Run this:

Configuration of Gammu

```
sudo gammu-config
```

A menu will appear. Use the arrow keys and the return key to navigate. When you've finished, your settings should look like this:

Recommended Gammu Settings

```
Port: /dev/ttyUSB0
Connection: at19200
Model: empty
Synchronize time: yes
Log file: leave empty
Log format: nothing
Use locking: leave empty
Gammu localisation: leave empty
```

Now use the arrow keys to navigate down to the Save option and press enter. We can tell if everything has worked by issuing the following command:

Identifying our Dongle

```
sudo gammu --identify
```

You should get a response back in your Terminal that looks something like this (obviously I've redacted some personal information, but yours will be fairly similar):

Gammu Identify Results

```
Device                : /dev/ttyUSB0
Manufacturer          : Huawei
Model                 : E173 (E173)
Firmware              : 11.126.85.00.00
IMEI                  : ***************
SIM IMSI              : ***************
```

Sending an SMS

Okay, let's send our first text message! Exciting stuff, kids: we're finally at the good bit. Use the following command to see if everything is working. Replace `**********` with your mobile number. You may also need to format it with the country code; I've never had to but you may need to.

Sending a test SMS

```
echo "test" | sudo gammu sendsms TEXT **********
```

If all goes well, after a few seconds you will receive a text message to your mobile phone with the word **test**.

Rock!

🤘

And that's it, you're finished! In a follow-up post I will show you how to get your Raspberry Pi to do things when it receives SMS messages containing particular words or phrases, so look out for that.

If you get stuck, just drop me a line and I'll be happy to help. I'm @escapologybb and @robotsandcakes On Twitter.

1. The terminal is located in the folder `/Applications/Utilities/Terminal.app` on macOS, and should be pretty easy to find on other distributions as well. ↩

By Stuart Turner

My name is Stuart, and I'm addicted to the Raspberry Pi.

Modeswitch does exactly what it says - it changes the mode of the modem to "modem" instead of "card reader" and then you can send messages and make calls and do whatever on your new cellular connection.

• Reply • Share ›

Thitisan Phuanglek • 6 months ago

Thank you so much . Your article help me to solved by issue about send and received sms on PC.
My info
Ubuntu 16.04 LTS
3G_USB card : E173

• Reply • Share ›

Stuart Mod → Thitisan Phuanglek • 4 months ago

You are very welcome, I'm glad it helped. :-)

15 • Reply • Share ›

Rafael Cota Rivas • a year ago

What about receiving SMS/calls? How di I find out the Caller ID so I know who's calling/sending SMS? @rafaelcotar

• Reply • Share ›

Stuart Mod → Rafael Cota Rivas • 4 months ago

Hello hello! Stupid Disqus wasn't telling me that I had comments on my blog. Apologies for the lateness! So if I understand you correctly you like to know who was sending you an SMS? Obviously this is a USB dongle that you can't receive calls on it, but the commands you need to see SMS messages sent to your dongle is:

sudo gammu getallsms

Hope this helps!

• Reply • Share ›

Andrian Bacalov → Rafael Cota Rivas • 7 months ago

Hi!
Good question...

• Reply • Share ›

Stuart Mod → Andrian Bacalov • 4 months ago

Hello, I think I answered your question above but if not let me know.

• Reply • Share ›

Jackiedk100 → Rafael Cota Rivas • 10 months ago

I look for the answer to this too

• Reply • Share ›

Stuart Mod → Jackiedk100 • 4 months ago

Hello, did you find the answer to the question? If not let me know :-)

• Reply • Share ›

Jackiedk100 → Stuart • 4 months ago

Yes, you just answered :)

Do you know if it's possible to make a call forwarder?

For instance, let's say someone call *THIS NUMBER* and after few seconds get forwarded to another number.

• Reply • Share ›

Jackiedk100 → Jackiedk100 • 4 months ago

Btw., great tutorial! I enjoyed it and learned lots of things from it :)

• Reply • Share ›