Raspberry Pi : How to access the Internet using GSM / GPRS Modem (SIM800)

Hardware Requirements

Raspberry Pi 4

GSM/GPRS Modem

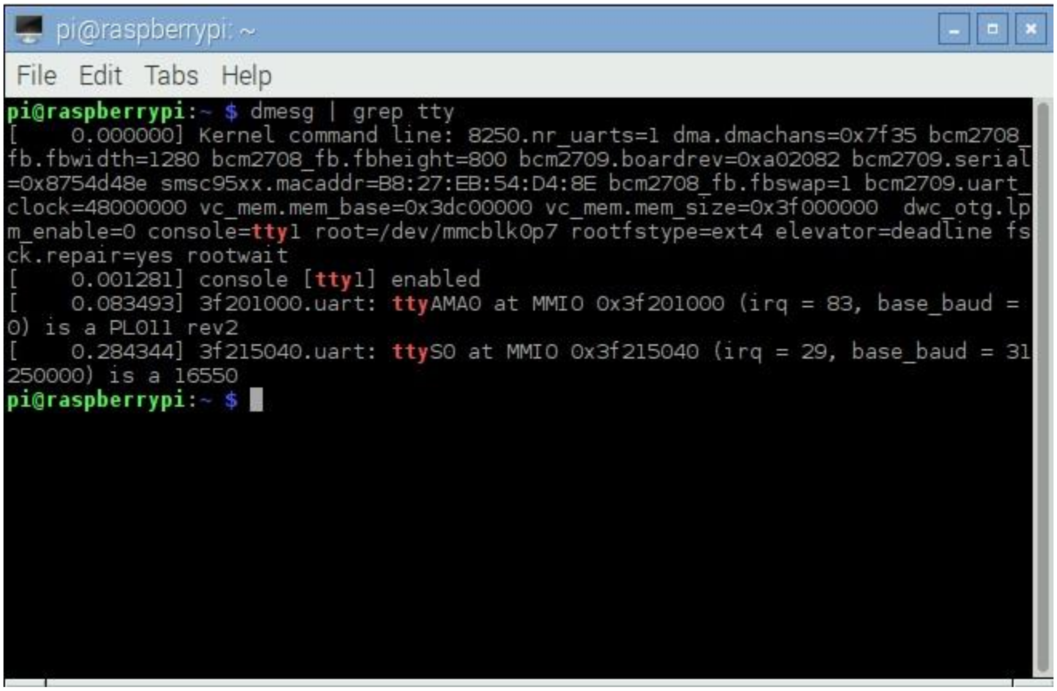
5v 3A power source (For GSM modem

Serial Communication in Raspberry Pi 4

Open the LX teminal and type:

```
dmesg | grep tty
```

This command will list the available UART modules, as shown in the figure:



```
pi@raspberrypi:~ $ dmesg | grep tty
[    0.000000] Kernel command line: 8250.nr_uarts=1 dma.dmachans=0x7f35 bcm2708_
fb.fbwidth=1280 bcm2708_fb.fbheight=800 bcm2709.boardrev=0xa02082 bcm2709.serial
=0x8754d48e smsc95xx.macaddr=B8:27:EB:54:D4:8E bcm2708_fb.fbswap=1 bcm2709.uart_
clock=48000000 vc_mem.mem_base=0x3dc00000 vc_mem.mem_size=0x3f000000  dwc_otg.lp
m_enable=0 console=tty1 root=/dev/mmcblk0p7 rootfstype=ext4 elevator=deadline fs
ck.repair=yes rootwait
[    0.001281] console [tty1] enabled
[    0.083493] 3f201000.uart: ttyAMA0 at MMIO 0x3f201000 (irq = 83, base_baud =
0) is a PL011 rev2
[    0.284344] 3f215040.uart: ttyS0 at MMIO 0x3f215040 (irq = 29, base_baud = 31
250000) is a 16550
pi@raspberrypi:~ $ 
```

The list will show, ttyS0 – UART corresponding to the GPIO header.

```
#GSM TESTING CODE
import serial
import os, time
  # Enable Serial Communication
port = serial.Serial("/dev/ttyS0", baudrate=9600, timeout=1)
  # Transmitting AT Commands to the Modem
# '\r\n' indicates the Enter key
  port.write('AT'+'\r\n')
rcv = port.read(10)
print( rcv)
```

Run the code and if the connections are all working good, then we will receive an OK acknowledgement in the python shell.

PPP Configuration

PPP or Point to Point Protocol establishes a Node to Node communication using serial interface. We make use of this, while accessing serial data connection on a PC. Here, using the serial connection, proper commands and PPP, we are about to access the internet on Pi.

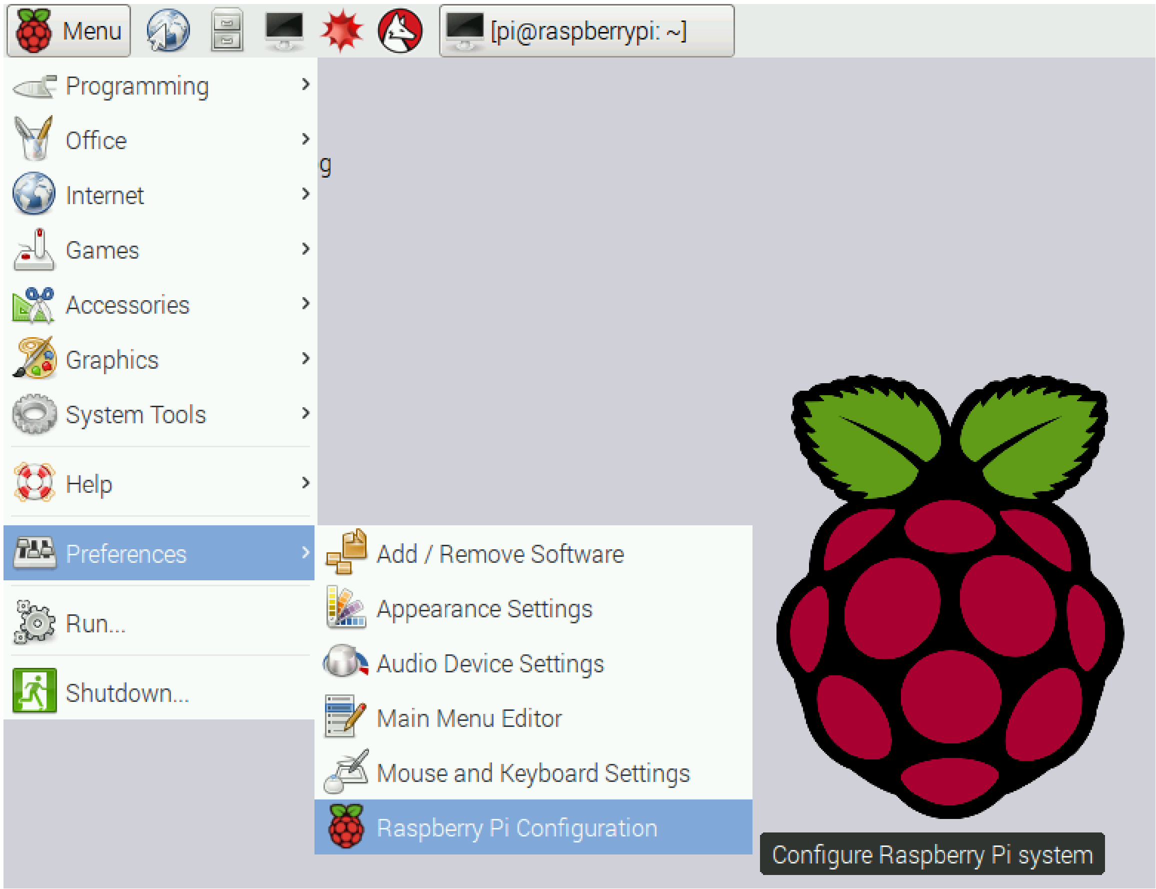
APN

An Access Point Name (APN) is the name of a gateway between a GSM/GPRS network and the internet. The APN of the cellular network that we are using, must be know to us. You can either ask it out on google or contact the service provider. In this example, we have used IDEA connection and for the APN we asked it on Google.

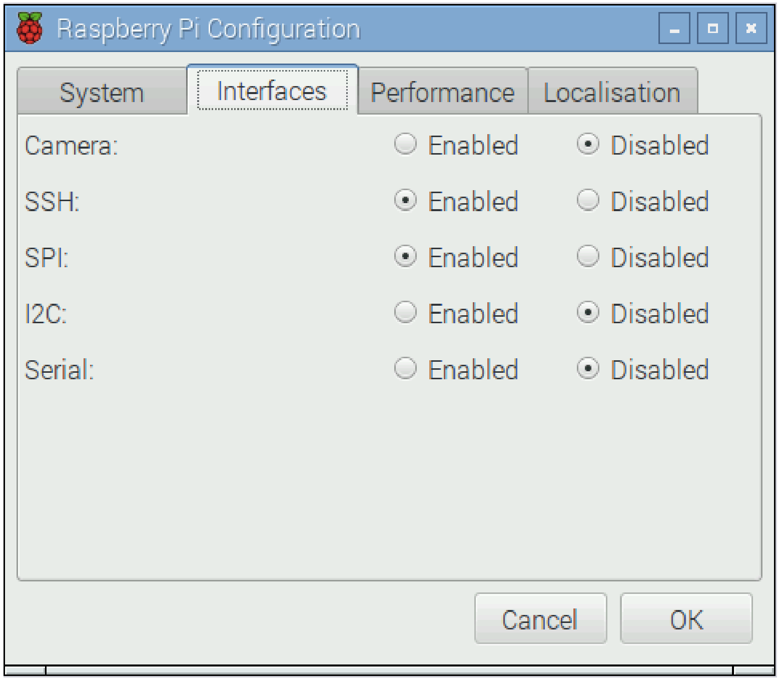
Installation

The first step is to disable the kernel's use of the hardware serial connection. By default, when the Raspberry Pi boots, it will use the serial connection to produce messages from the kernel and it will confuse the GSM modem. Follow the steps below

From Menu Select Preferences –> Raspberry Pi Configuration.



In this window, Select the interfaces tab and disable the serial option.



Camera: ○ Enabled ● Disabled

SSH: ● Enabled ○ Disabled

SPI: ● Enabled ○ Disabled

I2C: ○ Enabled ● Disabled

Serial: ○ Enabled ● Disabled

The next process is to install PPP Software. Make a suitable internet connection on the RPi by means of an Ethernet cable or WiFi. Open the LX Terminal and type:

```
sudo apt-get update
```

```
sudo apt-get install ppp screen elinks
```

After the installation we have to create a new PPP peer configuration. For this operation, we should login as root by entering `sudo -i` in the terminal and navigate to the peers directory,

```
cd /etc/ppp/peers/
```

Now open a new file rnet in a text editor by executing:

```
nano rnet
```

Copy the code shown below to the new file.

```
#code start
#imis/internet is the apn for idea connection
connect "/usr/sbin/chat -v -f /etc/chatscripts/gprs -T imis/internet"
  # For Raspberry Pi4 use /dev/ttyS0 as the communication port:
/dev/ttyS0
  # Baudrate
115200
  # Assumes that your IP address is allocated dynamically by the ISP.
noipdefault
  # Try to get the name server addresses from the ISP.
usepeerdns
  # Use this connection as the default route to the internet.
defaultroute
  # Makes PPPD "dial again" when the connection is lost.
persist
  # Do not ask the remote to authenticate.
noauth
  # No hardware flow control on the serial link with GSM Modem
nocrtscts
  # No modem control lines with GSM Modem
local
#code end
```

From the above file, two parameters must be changed which depends on the modules used. In the beginning command, connect, after the section -T, the APN of the service provider should be given. Here we have used the APN of Idea (imis/internet).

Then comes the communication port, as we have used Pi 4 in our exapmle , the port name is /dev/ttyS0. Save the configuration file by pressing ctrl+x.

In the above file, with in the line connect, a chat script is mentioned. To see this file, enter:

```
<strong>nano /etc/chatscripts/gprs</strong>
```

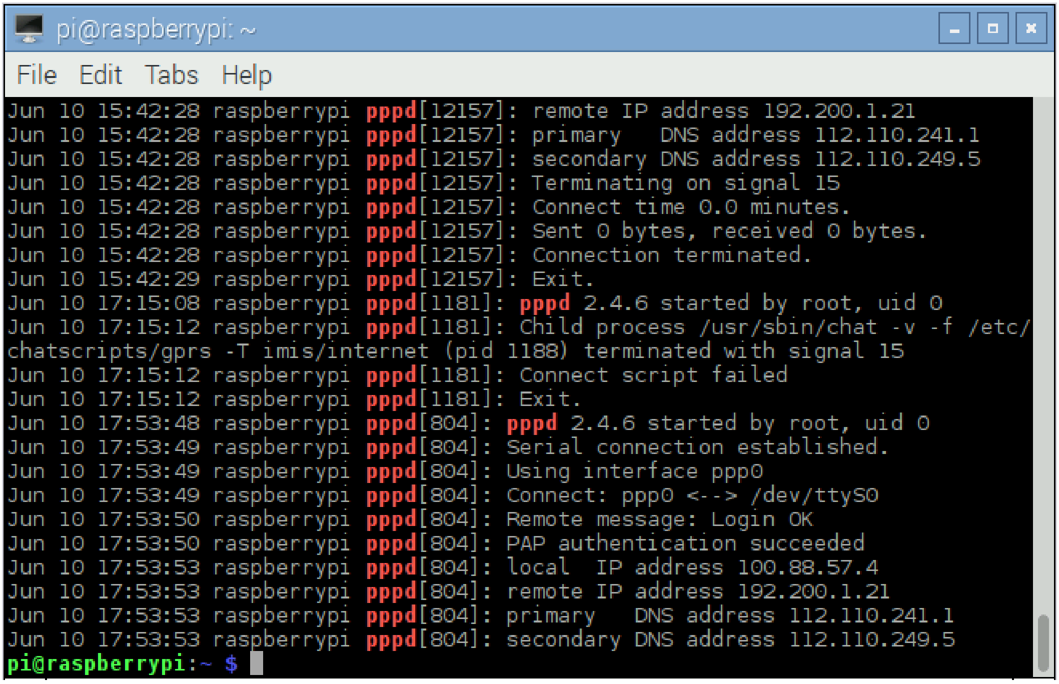
This will open a new file, which is not needed to be edited and this explains how a GPRS connection is created.   If the SIM card needs a PIN to unlock, un comment the line AT+CPIN=1234 and set the pin instead of 1234. Save the file and exit.

Next is to establish the connection. Before that make sure, all the steps described above are executed with no errors and proper wiring connections are also made.

Type, `sudo pon rnet` for creating the connection and it will exit with no response shown in the Lx terminal but the PPPD should be setting up the connection. To show the log for PPPD type,

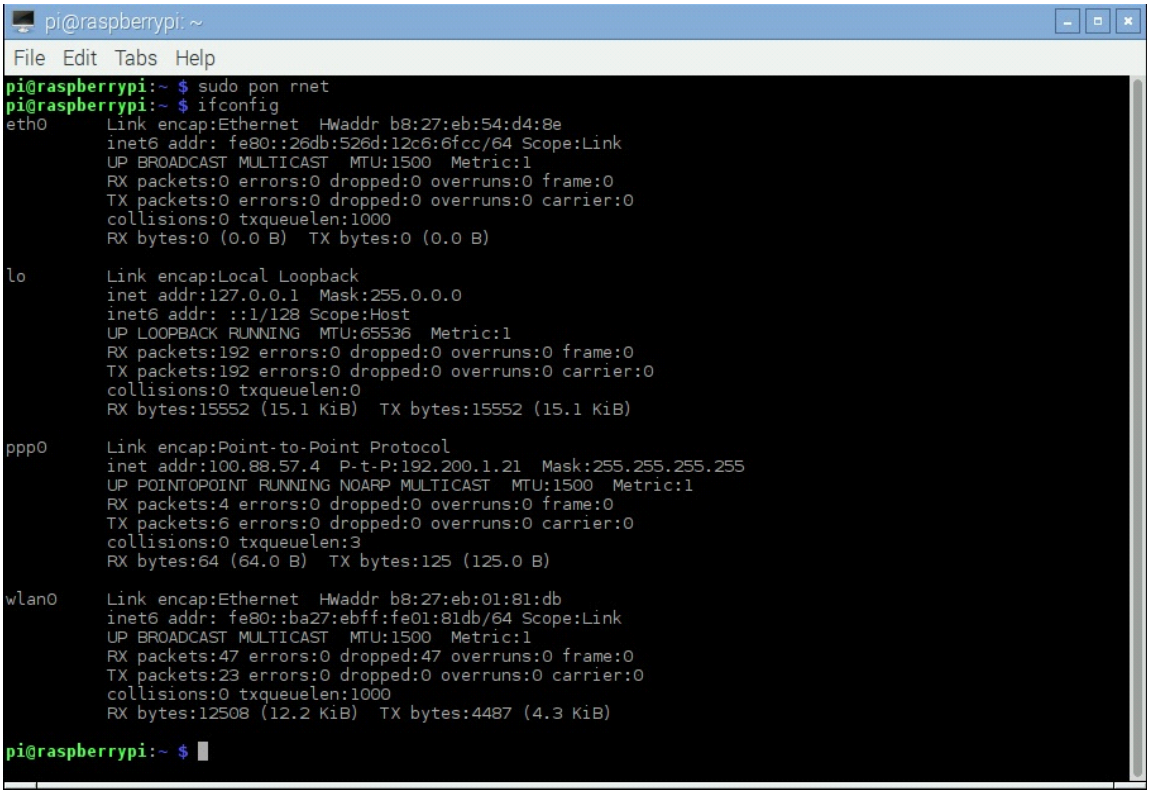
```
<strong>cat /var/log/syslog | grep pppd</strong>
```

It will shows more messages and if the connection was established successfully, we can see the messages at the end as shown in the figure:



```
Jun 10 15:42:28 raspberrypi pppd[12157]: remote IP address 192.200.1.21
Jun 10 15:42:28 raspberrypi pppd[12157]: primary   DNS address 112.110.241.1
Jun 10 15:42:28 raspberrypi pppd[12157]: secondary DNS address 112.110.249.5
Jun 10 15:42:28 raspberrypi pppd[12157]: Terminating on signal 15
Jun 10 15:42:28 raspberrypi pppd[12157]: Connect time 0.0 minutes.
Jun 10 15:42:28 raspberrypi pppd[12157]: Sent 0 bytes, received 0 bytes.
Jun 10 15:42:28 raspberrypi pppd[12157]: Connection terminated.
Jun 10 15:42:29 raspberrypi pppd[12157]: Exit.
Jun 10 17:15:08 raspberrypi pppd[1181]: pppd 2.4.6 started by root, uid 0
Jun 10 17:15:12 raspberrypi pppd[1181]: Child process /usr/sbin/chat -v -f /etc/
chatscripts/gprs -T imis/internet (pid 1188) terminated with signal 15
Jun 10 17:15:12 raspberrypi pppd[1181]: Connect script failed
Jun 10 17:15:12 raspberrypi pppd[1181]: Exit.
Jun 10 17:53:48 raspberrypi pppd[804]: pppd 2.4.6 started by root, uid 0
Jun 10 17:53:49 raspberrypi pppd[804]: Serial connection established.
Jun 10 17:53:49 raspberrypi pppd[804]: Using interface ppp0
Jun 10 17:53:49 raspberrypi pppd[804]: Connect: ppp0 <--> /dev/ttyS0
Jun 10 17:53:50 raspberrypi pppd[804]: Remote message: Login OK
Jun 10 17:53:50 raspberrypi pppd[804]: PAP authentication succeeded
Jun 10 17:53:53 raspberrypi pppd[804]: local  IP address 100.88.57.4
Jun 10 17:53:53 raspberrypi pppd[804]: remote IP address 192.200.1.21
Jun 10 17:53:53 raspberrypi pppd[804]: primary   DNS address 112.110.241.1
Jun 10 17:53:53 raspberrypi pppd[804]: secondary DNS address 112.110.249.5
pi@raspberrypi:~ $ 
```

In the terminal type ifconfig and enter.



```
pi@raspberrypi:~ $ sudo pon rnet
pi@raspberrypi:~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:54:d4:8e
          inet6 addr: fe80::26db:526d:12c6:6fcc/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:192 errors:0 dropped:0 overruns:0 frame:0
          TX packets:192 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:15552 (15.1 KiB)  TX bytes:15552 (15.1 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:100.88.57.4  P-t-P:192.200.1.21  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:64 (64.0 B)  TX bytes:125 (125.0 B)

wlan0     Link encap:Ethernet  HWaddr b8:27:eb:01:81:db
          inet6 addr: fe80::ba27:ebff:fe01:81db/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:47 errors:0 dropped:47 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12508 (12.2 KiB)  TX bytes:4487 (4.3 KiB)

pi@raspberrypi:~ $ 
```

A new section PPP0 can be seen with the IP 192.200.1.21. In the modem the blue led will blink fast continuously as the connection in ON. To close the connection, type: sudo poff rnet.