CONFIGURATION OF 4G LTE MODEM SIM 7600E

Step 1:

Connect 4G HAT to RPI via USB

we want to use the UART, we need to change the settings. Executing this command to enter the configuration page.

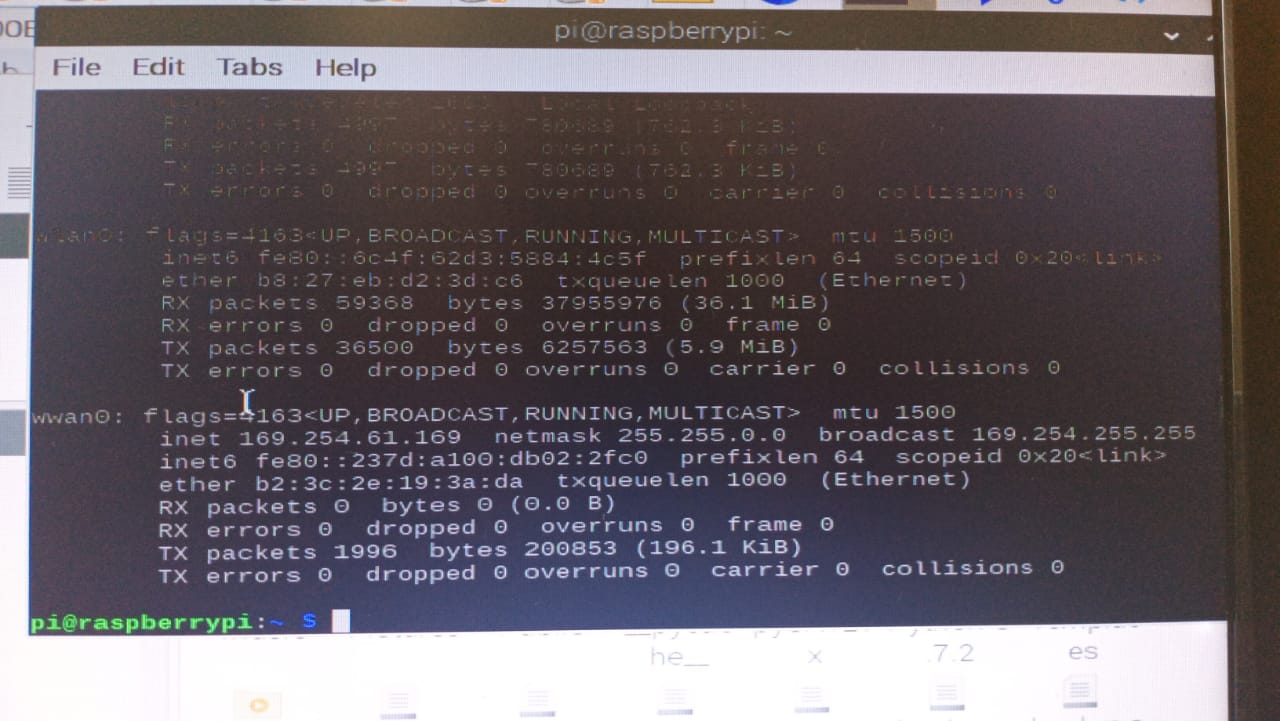
Sudo raspi-config

Go to interfacing options and disable the serial

Step 2:

Run the ifconfig command to see if a wwan0 card is identified

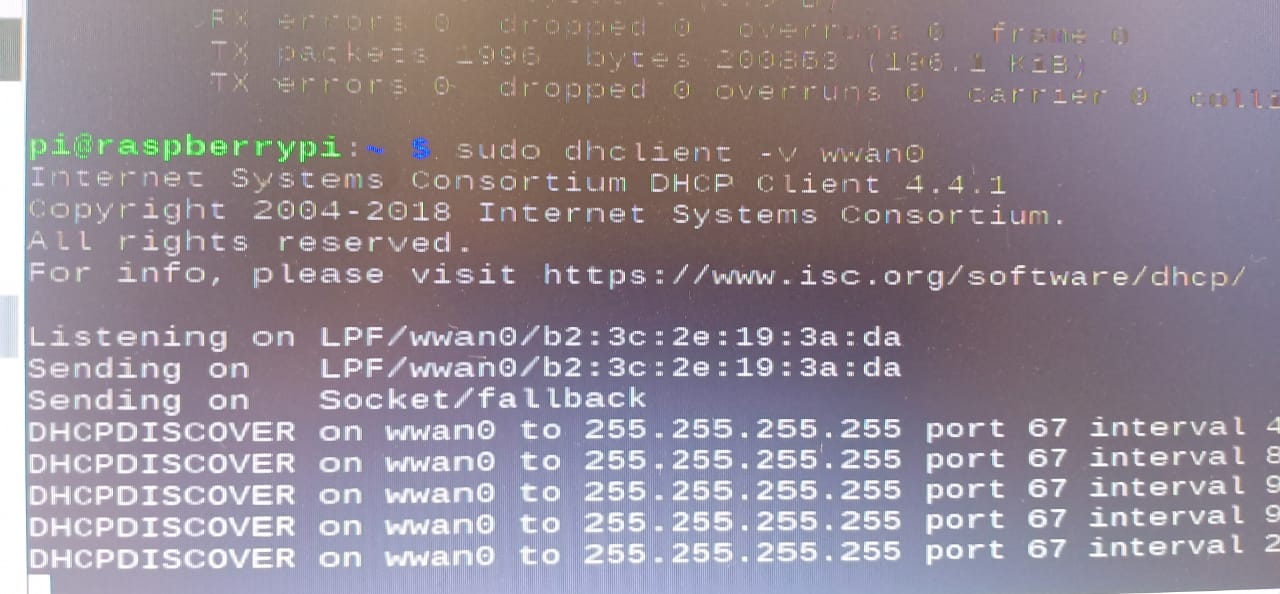
Ifconfig



Step 3:

Get IP address by typing the command

sudo dhclient –v wwan0



4.Install required software

sudo apt-get update && sudo apt-get install libqmi-utils udhcpc

5. when you reboot the pi, the SIM7600 module's cellular radio is OFF , use the newly installed qmi utils:

sudo qmicli -d /dev/cdc-wdm0 --dms-set-operating-mode='online'

qmicli -d /dev/cdc-wdm0 --dms-get-operating-mode

qmicli -d /dev/cdc-wdm0 --nas-get-signal-strength

qmicli -d /dev/cdc-wdm0 --nas-get-home-network

If the first command shows in its output 'Low Power' or anything other than 'online' it means your radio is off and needs to be turned on.  
The last of those commands should return the LTE network ID if your device successfully connected.  
  
6. Reconfigure the network interface for raw-ip protocol  
  
The qmi-wwan kernel driver creates the **wwan0** network interface for you when it detects the SIM7600 module connected to your Raspberry Pi. By default that interface is set to **802-3** protocol, however it seems the correct protocol should be **raw-ip**. The **qmi-network** script tries to set that up for you, but it will most likely fail. To make the change, do the following:

sudo qmicli -d /dev/cdc-wdm0 -w

sudo ip link set wwan0 down

echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw\_ip

sudo ip link set wwan0 up

7. Connect to the mobile network.  
  
After having done all the above, when trying to use **qmicli** to connect, I was getting the following error: **error: "couldn't start network: QMI protocol error (64): '(null)'.**.  
  
There have apparently been some changes to **qmi utils** in Stretch and most of the documentation I saw over the last days is almost correct but missing one vital detail - you need to add **ip-type=4** to other parameters for **qmicli -d /dev/cdc-wdm0 --wds-start-network=** as below.

qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-header' --wds-start network="apn='YOUR\_APN',username='YOUR\_USERNAME',password='YOUR\_PASSWORD',ip-type=4" --client-no-release-cid

[/dev/cdc-wdm0] Network started

Packet data handle: '2264328160'

[/dev/cdc-wdm0] Client ID not released:

Service: 'wds'

CID: '20'

8. Finally, configure the IP address and the default route with **udhcpc**:

sudo udhcpc -i wwan0

udhcpc (v1.22.1) started

No resolv.conf for interface wwan0.udhcpc

Sending discover...

Sending select for 10.65.52.178...

Lease of 10.65.52.178 obtained, lease time 7200

cat: /run/resolvconf/lock/pid: No such file or directory

/sbin/resolvconf: 733: kill: Illegal number:

clearing stale lock pid

Too few arguments.

Too few arguments.

As you can see, this spits out a number of errors, but they relate to configuration of DNS servers in your **resolv.conf** file, but otherwise this works and I can now get the connection via the mobile network:

ip a s wwan0

3: wwan0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UNKNOWN group default qlen 1000

link/none

inet 10.65.52.178/30 scope global wwan0

valid\_lft forever preferred\_lft forever

ip r s

default via 10.65.52.177 dev wwan0

10.65.52.176/30 dev wwan0 proto kernel scope link src 10.65.52.178

192.168.100.0/24 dev eth0 proto kernel scope link src 192.168.100.5 metric 202

ping -4 www.google.com

PING www.google.com (74.125.193.147) 56(84) bytes of data.

64 bytes from ig-in-f147.1e100.net (74.125.193.147): icmp\_seq=1 ttl=50 time=732 ms

64 bytes from ig-in-f147.1e100.net (74.125.193.147): icmp\_seq=2 ttl=50 time=79.4 ms

64 bytes from ig-in-f147.1e100.net (74.125.193.147): icmp\_seq=3 ttl=50 time=80.3 ms

64 bytes from ig-in-f147.1e100.net (74.125.193.147): icmp\_seq=4 ttl=50 time=79.4 ms

^C

---www.google.com ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3002ms

rtt min/avg/max/mdev = 79.447/242.945/732.530/282.662 ms