Connect to the Internet via USB with LTE EC25 Using HAT on Raspberry Pi

Hardware Prerequisites

- Raspberry Pi
- Raspberry PI HAT for IoT modules
- Quectel EC25 mini PCIe module
- LTE antenna
- Micro-USB cable

Hardware Setup

- Plug Quectel EC25 Mini PCIe module to the Raspberry PI HAT for IoT modules
- Connect required antennas to the Quectel EC25 module.
- Attach the Raspberry PI HAT on top of your Raspberry Pi
- Connect using micro-USB cable the Raspberry PI HAT

Software Setup

- First run update

```
sudo apt update && sudo apt upgrade
```

- Install some programs:
  sudo apt-get install libqmi-utils udhcpc

- Check connection:
  sudo qmicli -d /dev/cdc-wdm0 --dms-get-operating-mode
  sudo qmicli -d /dev/cdc-wdm0 --nas-get-signal-strength
  sudo qmicli -d /dev/cdc-wdm0 —nas-get-home-network

- Change qmi_wwan driver to use Raw-IP.
  Disable the network interfaces exposed by the cellular module:
  ip link set dev wwan0 down
  Trigger the Raw-IP support:
  echo Y > /sys/class/net/wwan0/qmi/raw_ip
  Enable the network interfaces again:
  ip link set dev wwan0 up

- Activate the data connection in the cellular module:
  qmicli --device=/dev/cdc-wdm0 --device-open-proxy --wds-start-network="ip-type=4,apn=<YOUR_APN>" --client-no-release-cid
- Once "Network started" is displayed, you can send a DHCP request on the network interface.
  udhcpc -q -f -n -i wwan0
- If the connection was successfully set up established, you now have data connectivity. A ping to a remote server using the cellular network interface can for example prove this:
  ping -I wwan0 8.8.8.8
- Disconnect the data bearer and data connection over QMI by command bellow and providing the network handle and CID returned at connection activation:
  qmicli --device=/dev/cdc-wdm0 --device-open-proxy --wds-stop-network=NETWORK_HANDLE —client-cid=CID

Additional useful commands:

Request module manufacturer:
qmicli --device=/dev/cdc-wdm0 --device-open-proxy --dms-get-manufacturer

Get module model:
qmicli --device=/dev/cdc-wdm0 --device-open-proxy --dms-get-model

Get firmware version:
qmicli --device=/dev/cdc-wdm0 --device-open-proxy --dms-get-revision

Get module IDs (IMEI etc.):
qmicli --device=/dev/cdc-wdm0 --device-open-proxy --dms-get-ids

Get SIM card status:
qmicli --device=/dev/cdc-wdm0 --device-open-proxy --uim-get-card-status

Recent cellular modules like Sierra Wireless EM7565 require at least libqmi V1.20. Check version with command:
qmicli --version

If the connection was successfully set up established, you now have data connectivity. A ping to a remote server using the cellular network interface can for example prove this:
ping -I wwan0 8.8.8.8

The ifconfig Linux tool can show the current details for the network interface:

ifconfig wwan0