HOWTO use
LTE modem on Tx/Cx Controller

it´s all about perfection

RESI

STRIVE IN PERFECTION
IN WHATEVER YOU
DO
TAKE THE BEST THAT
EXISTS AND MAKE IT
BETTER
WHEN IT DOES NOT
EXIST. DESIGN IT.

Sir Henry Royce

CONTENT

In this application note you will find ...

PREREQUISITES

Raspberry Pi is a trademark of the Raspberry Pi Foundation.
More information under www.raspberrypi.org

PREREQUISITES

We assume that the reader is familiar how to use WINDOWS® operating system, how to configure a LINUX® Ethernet interface, how to use a remote desktop program or SSH console to configure LINUX®. Also we assume that the user is able to consult the internet for technical hint, how to configure LINUX, especially Raspberry Pi devices.

RESI is in no way responsible, if you or your customer cannot use the given advice here, because of lack of education in your or their staff!

With the purchase of a IoT Controller from RESI, you have not purchased the right of free education or free consulting from RESI!

IMPORTANT SAFETY NOTES

Important hint:

Before you start with the installation and the initial setup of the device, you have to read this document and the attached installation guide and the actual manual for the device very carefully. You have to follow all the herein given information very accurate!

- Only authorized and qualified personnel are allowed to install and setup the device!
- The connection of the device must be done in de-energized state!
- Do not perform any electrical work while the device is connected to power!
- Disable and secure the system against any automatic restart or power on procedure!
- The device must be operated with the defined voltage level!
- Supply voltage jitters must not exceed the technical specifications and tolerances given in the technical manuals for the product. If you do not obey this issue, the proper performance of the device cannot be guaranteed. This can lead to fail functions of the device and in worst case to a complete breakdown of the device!
- You have to obey the current EMC regulations for wiring!
- All signal, control and supply voltage cables must be wired in a way, that no inductive or capacitive interference or any other severe electrical noise disturbance may interfere with the device. Wrong wiring can lead to a malfunction of the device!
- For signal or sensor cables you have to use shielded cables, to avoid damages through induction!
- You have to obey and to apply the current safety regulations given by the ÖVE, VDE, the countries, their control authorities, the TÜV or the local energy supply company!
- Obey country-specific laws and standards!
- The device must be used for the intended purpose of the manufacturer!
- No warranties or liabilities will be accepted for defects and damages resulting from improper or incorrect usage of the device!
- Subsequent damages, which results from faults of this device, are excluded from warranty and liability!
- Only the technical data, wiring diagrams and operation instructions, which are part to the product shipment are valid!
- The information on our homepage, in our datasheets, in our manuals, in our catalogues or published by our partners can deviate from the product documentation and is not necessarily always actual, due to constant improvement of our products for technical progress!
- In case of modification of our devices made by the user, all warranty and liability claims are lost!
- The installation has to fulfill the technical conditions and specifications (e.g. operating temperatures, power supply, ...) given in the devices documentation!
- Operating our device close to equipment, which do not comply with EMC directives, can influence the functionality of our device, leading to malfunction or in worst case to a breakdown of our device!
- Our devices must not be used for monitoring applications, which solely serve the purpose of protecting persons against hazards or injury, or as an emergency stop switch for systems or machinery, or for any other similar safety-relevant purposes!
- Dimensions of the enclosures or enclosures accessories may show slight tolerances on the specifications provided in these instructions!
- Modifications of this documentation is not allowed!
- In case of a complaint, only complete devices returned in original packing will be accepted!

Setup LTE modem on RESI-C4 controller

it´s all about perfection

Setup LTE modem on RESI-C4 controller

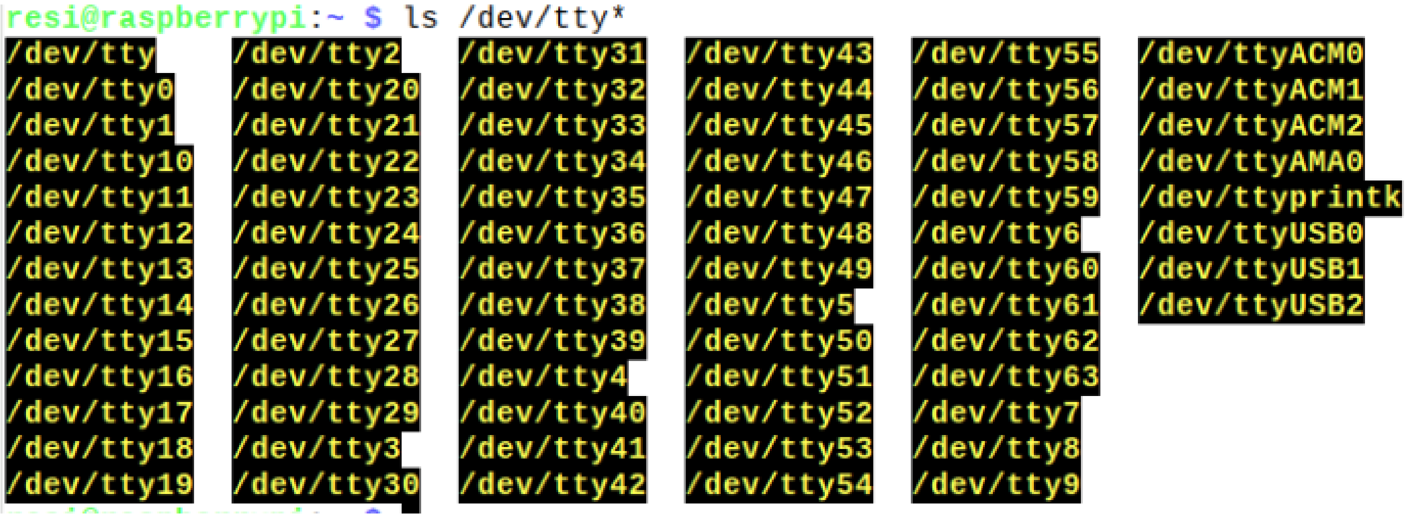
RESI uses the QUECTEC EC25 LTE modem inside of the controller. The modem is coupled with USB to the Raspberry Pi and it will offer some additional serial interfaces.

Open with VNCViewer the Raspberry Desktop or connect your monitor direct to HDMI and keyboard+mouse to the USB interface of our C4 controllers.

Open a shell and enter the command

```
ls /dev/tty*
```

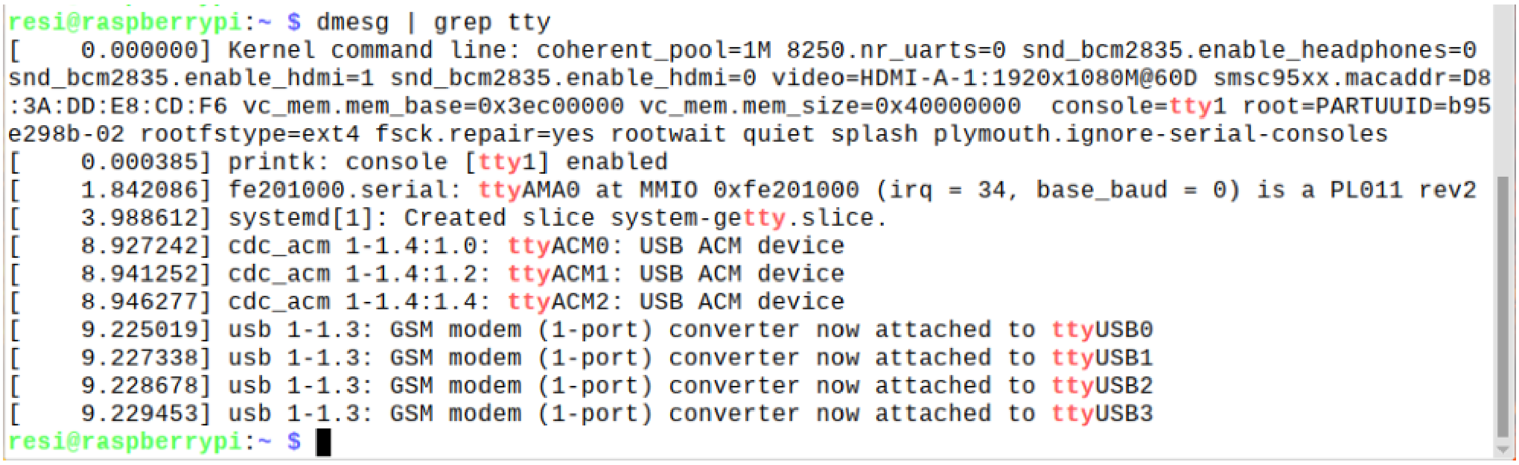
You will see a similar picture. You have to have the devices dev/ttyUSB0, dev/ttyUSB1, dev/ttyUSB2 and dev/ttyUSB3.



```
resi@raspberrypi:~ $ ls /dev/tty*
/dev/tty    /dev/tty2   /dev/tty31  /dev/tty43  /dev/tty55  /dev/ttyACM0
/dev/tty0   /dev/tty20  /dev/tty32  /dev/tty44  /dev/tty56  /dev/ttyACM1
/dev/tty1   /dev/tty21  /dev/tty33  /dev/tty45  /dev/tty57  /dev/ttyACM2
/dev/tty10  /dev/tty22  /dev/tty34  /dev/tty46  /dev/tty58  /dev/ttyAMA0
/dev/tty11  /dev/tty23  /dev/tty35  /dev/tty47  /dev/tty59  /dev/ttyprintk
/dev/tty12  /dev/tty24  /dev/tty36  /dev/tty48  /dev/tty6   /dev/ttyUSB0
/dev/tty13  /dev/tty25  /dev/tty37  /dev/tty49  /dev/tty60  /dev/ttyUSB1
/dev/tty14  /dev/tty26  /dev/tty38  /dev/tty5   /dev/tty61  /dev/ttyUSB2
/dev/tty15  /dev/tty27  /dev/tty39  /dev/tty50  /dev/tty62
/dev/tty16  /dev/tty28  /dev/tty4   /dev/tty51  /dev/tty63
/dev/tty17  /dev/tty29  /dev/tty40  /dev/tty52  /dev/tty7
/dev/tty18  /dev/tty3   /dev/tty41  /dev/tty53  /dev/tty8
/dev/tty19  /dev/tty30  /dev/tty42  /dev/tty54  /dev/tty9
```

You can also check with dmesg, which serial ports are used. Check the GSM modem entries:

```
dmesg | grep tty
```

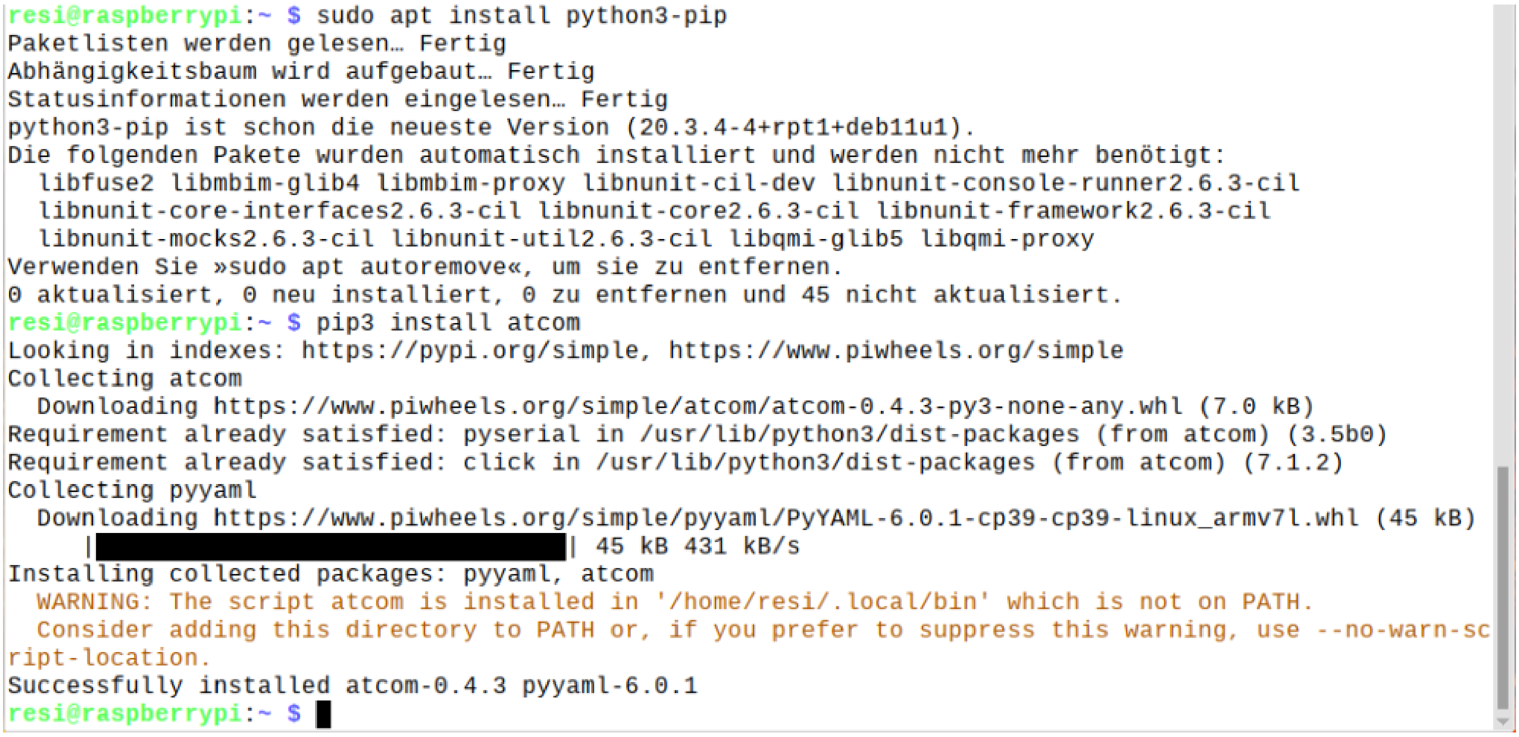


```
resi@raspberrypi:~ $ dmesg | grep tty
[    0.000000] Kernel command line: coherent_pool=1M 8250.nr_uarts=0 snd_bcm2835.enable_headphones=0
snd_bcm2835.enable_hdmi=1 snd_bcm2835.enable_hdmi=0 video=HDMI-A-1:1920x1080M@60D smsc95xx.macaddr=D8
:3A:DD:E8:CD:F6 vc_mem.mem_base=0x3ec00000 vc_mem.mem_size=0x40000000  console=tty1 root=PARTUUID=b95
e298b-02 rootfstype=ext4 fsck.repair=yes rootwait quiet splash plymouth.ignore-serial-consoles
[    0.000385] printk: console [tty1] enabled
[    1.842086] fe201000.serial: ttyAMA0 at MMIO 0xfe201000 (irq = 34, base_baud = 0) is a PL011 rev2
[    3.988612] systemd[1]: Created slice system-getty.slice.
[    8.927242] cdc_acm 1-1.4:1.0: ttyACM0: USB ACM device
[    8.941252] cdc_acm 1-1.4:1.2: ttyACM1: USB ACM device
[    8.946277] cdc_acm 1-1.4:1.4: ttyACM2: USB ACM device
[    9.225019] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB0
[    9.227338] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB1
[    9.228678] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB2
[    9.229453] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB3
resi@raspberrypi:~ $
```

Setup LTE modem on RESI-C4 controller

The next thing we need, is a tool to send AT commands to the LTE modem. Therefore we install ATcom on our image. We have preinstalled this tool on our image.

```
sudo apt install python3-pip

pip3 install atcom
```



```
resi@raspberrypi:~ $ sudo apt install python3-pip
Paketlisten werden gelesen… Fertig
Abhängigkeitsbaum wird aufgebaut… Fertig
Statusinformationen werden eingelesen… Fertig
python3-pip ist schon die neueste Version (20.3.4-4+rpt1+deb11u1).
Die folgenden Pakete wurden automatisch installiert und werden nicht mehr benötigt:
  libfuse2 libmbim-glib4 libmbim-proxy libnunit-cil-dev libnunit-console-runner2.6.3-cil
  libnunit-core-interfaces2.6.3-cil libnunit-core2.6.3-cil libnunit-framework2.6.3-cil
  libnunit-mocks2.6.3-cil libnunit-util2.6.3-cil libqmi-glib5 libqmi-proxy
Verwenden Sie »sudo apt autoremove«, um sie zu entfernen.
0 aktualisiert, 0 neu installiert, 0 zu entfernen und 45 nicht aktualisiert.
resi@raspberrypi:~ $ pip3 install atcom
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting atcom
  Downloading https://www.piwheels.org/simple/atcom/atcom-0.4.3-py3-none-any.whl (7.0 kB)
Requirement already satisfied: pyserial in /usr/lib/python3/dist-packages (from atcom) (3.5b0)
Requirement already satisfied: click in /usr/lib/python3/dist-packages (from atcom) (7.1.2)
Collecting pyyaml
  Downloading https://www.piwheels.org/simple/pyyaml/PyYAML-6.0.1-cp39-cp39-linux_armv7l.whl (45 kB)
     |████████████████████████████████| 45 kB 431 kB/s
Installing collected packages: pyyaml, atcom
  WARNING: The script atcom is installed in '/home/resi/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-sc
ript-location.
Successfully installed atcom-0.4.3 pyyaml-6.0.1
resi@raspberrypi:~ $ █
```

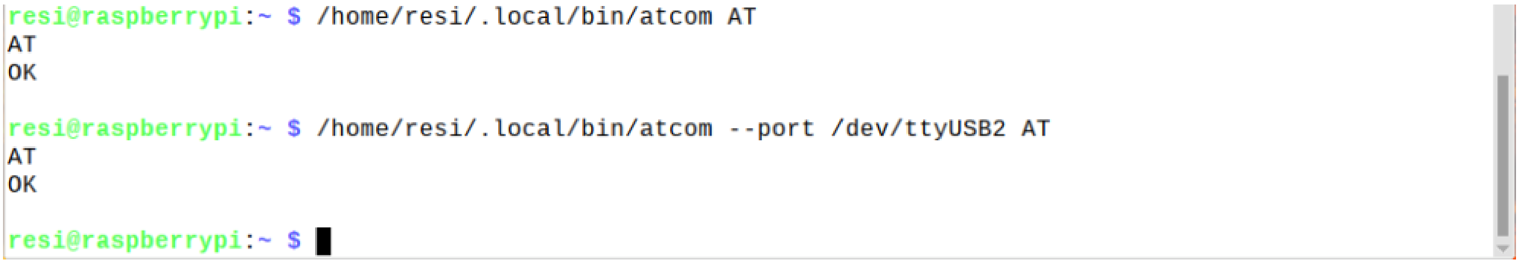
Since we are in our home directory , we have tu use the full path for atcom. But you can also specify the port with –port /dev/ttyUSBx

```
/home/resi/.local/bin/atcom AT
```

WARNING

If you need to send " or ' character, you should use escape character \ before it.

```
/home/resi/.local/bin/atcom AT+CGDCONT=1,\"IP\",\"internet\"
```



```
resi@raspberrypi:~ $ /home/resi/.local/bin/atcom AT
AT
OK

resi@raspberrypi:~ $ /home/resi/.local/bin/atcom --port /dev/ttyUSB2 AT
AT
OK

resi@raspberrypi:~ $ █
```

Setup LTE modem on RESI-C4 controller

In our ZIP file, you will find the QUECTEL AT COMMAND lists as PDF documents:

Quectel_EC2xEG9xEG2x-GEM05_Series_AT_Commands_Manual_V2.0-1.pdf
Quectel_EC2xEG2xEG9xEM05_Series_QCFG_AT_Commands_Manual_V1.0.pdf

Or search for QUECTEL EC25 AT Commands in the internet.

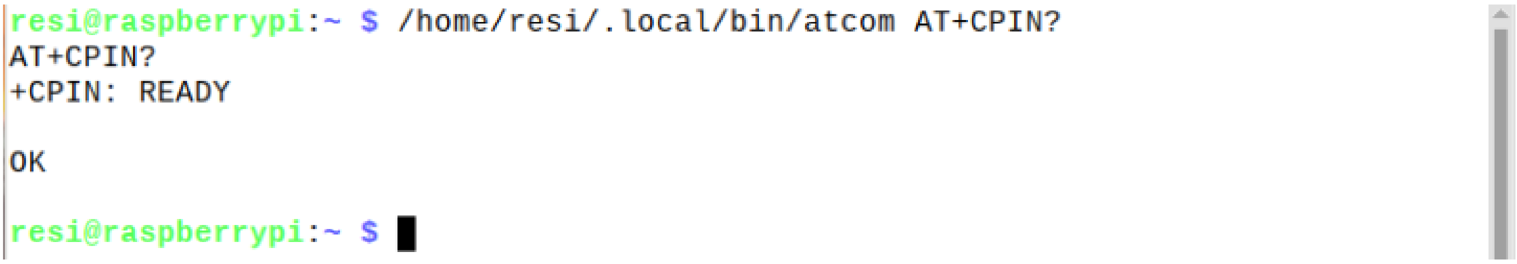
PLUG-IN YOUR SIM CARD AND CONNECT ANTENNA
Insert your SIM card very carefully in our SIM slot. If you miss it, it can be that your SIM lands within our housing. Then you have a real problem! Also connect your antenna to our SMA connector. Otherwise it could be, that the LTE modem cannot register in your network.

HINT: It will be good to deactivate the PIN code for first testing on your SIM card!

SIM CARD OK ?
Now let's check, if the PIN code is activated (Consult the QUECTEL manual, how the AT commands work and what information is encoded in the answer of the LTE modem)

```
/home/resi/.local/bin/atcom AT+CPIN?
```

The answer should be READY

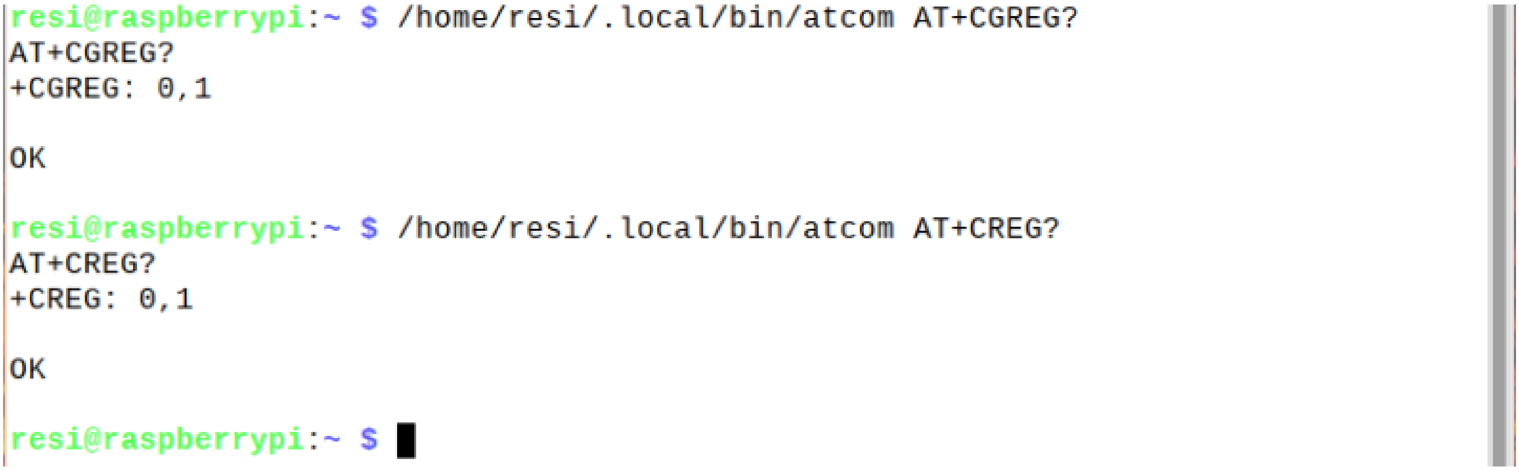


```
resi@raspberrypi:~ $ /home/resi/.local/bin/atcom AT+CPIN?
AT+CPIN?
+CPIN: READY

OK

resi@raspberrypi:~ $ 
```

LTE MODEM REGISTERED IN THE NETWORK ?
Then we check the registration status of the modem:

```
/home/resi/.local/bin/atcom AT+CGREG?
```
and
```
/home/resi/.local/bin/atcom AT+CREG?
```

The second answer parameter ,1 means, that the LTE modem is registered in the network.

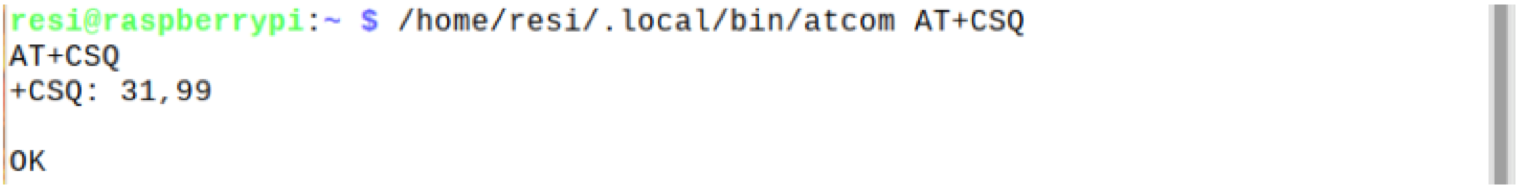


```
resi@raspberrypi:~ $ /home/resi/.local/bin/atcom AT+CGREG?
AT+CGREG?
+CGREG: 0,1

OK

resi@raspberrypi:~ $ /home/resi/.local/bin/atcom AT+CREG?
AT+CREG?
+CREG: 0,1

OK

resi@raspberrypi:~ $ 
```

Setup LTE modem on RESI-C4 controller

CHECK SIGNAL QUALITY ?
Next we check the signal quality:

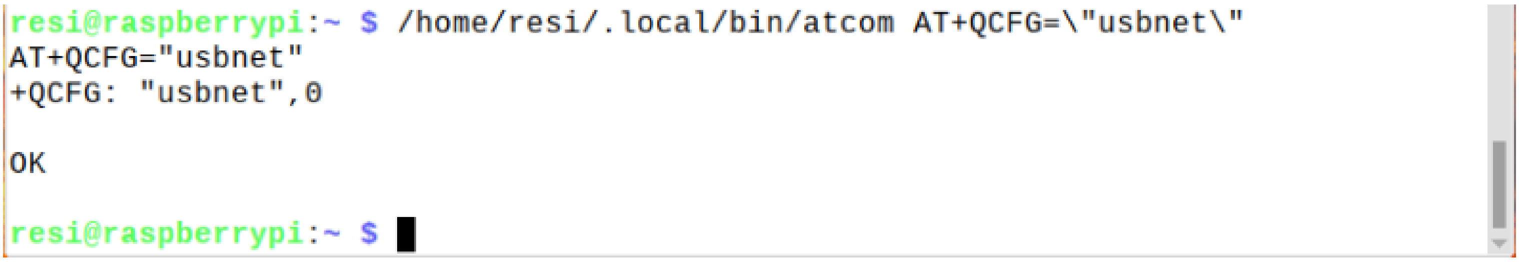
```
/home/resi/.local/bin/atcom AT+CSQ
```



```
resi@raspberrypi:~ $ /home/resi/.local/bin/atcom AT+CSQ
AT+CSQ
+CSQ: 31,99

OK
```

SETTING USBNET
Next we set USBNET mode

```
/home/resi/.local/bin/atcom AT+QCFG=\"usbnet\"
```



```
resi@raspberrypi:~ $ /home/resi/.local/bin/atcom AT+QCFG=\"usbnet\"
AT+QCFG="usbnet"
+QCFG: "usbnet",0

OK

resi@raspberrypi:~ $
```

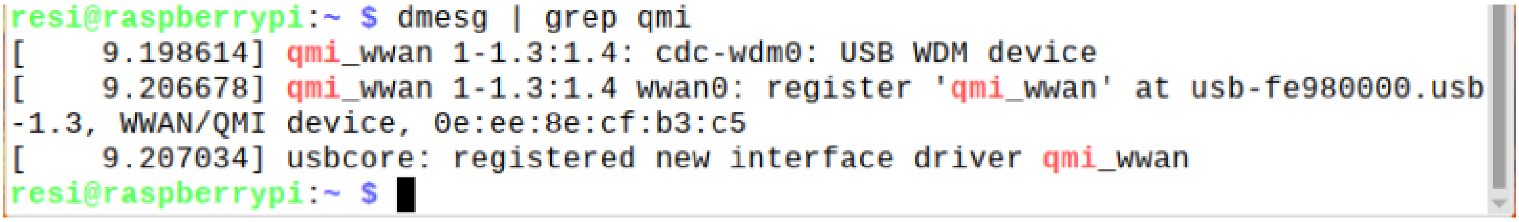
Setup LTE modem on RESI-C4 controller

CHECK FOR QMI DRIVER INSTALLED

We want to use the QMI wwan driver for Ethernet connection. First we check, if the driver is available:

dmesg | grep qmi

Your answer should look like this:

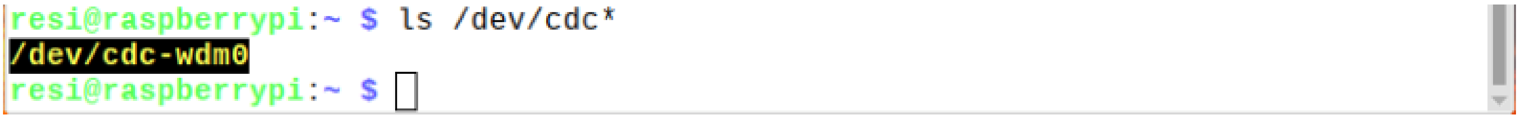


```
resi@raspberrypi:~ $ dmesg | grep qmi
[    9.198614] qmi_wwan 1-1.3:1.4: cdc-wdm0: USB WDM device
[    9.206678] qmi_wwan 1-1.3:1.4 wwan0: register 'qmi_wwan' at usb-fe980000.usb
-1.3, WWAN/QMI device, 0e:ee:8e:cf:b3:c5
[    9.207034] usbcore: registered new interface driver qmi_wwan
resi@raspberrypi:~ $ 
```

Then we check, if a device is listed in LINUX:

```
ls /dev/cdc*
```

We need this answer:



```
resi@raspberrypi:~ $ ls /dev/cdc*
/dev/cdc-wdm0
resi@raspberrypi:~ $ 
```

INSTALL QMI DRIVERS & UTILITIES

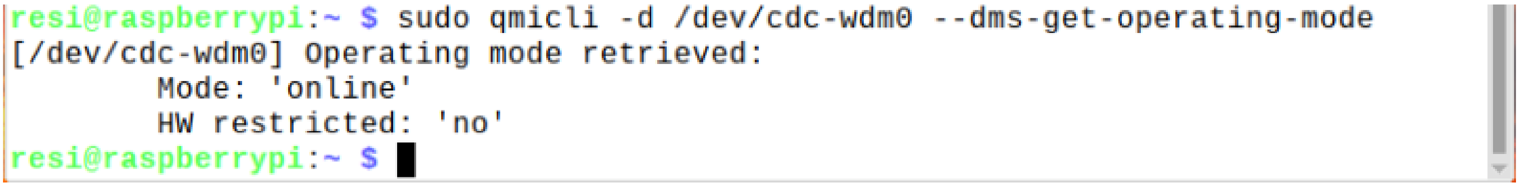
Next we install the following components

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

After finishing this installation we check with the command if the modem is online:

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

Answer should look like this:



```
resi@raspberrypi:~ $ sudo qmicli -d /dev/cdc-wdm0 --dms-get-operating-mode
[/dev/cdc-wdm0] Operating mode retrieved:
        Mode: 'online'
        HW restricted: 'no'
resi@raspberrypi:~ $ 
```

INFO: If this returns a resource temporarily unavailable error after a bit, you might need to first stop ModemManager (if it's installed and running) with

```
sudo systemctl stop ModemManager
```

then try again.

If it doesn't return online, run

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

to set it online.

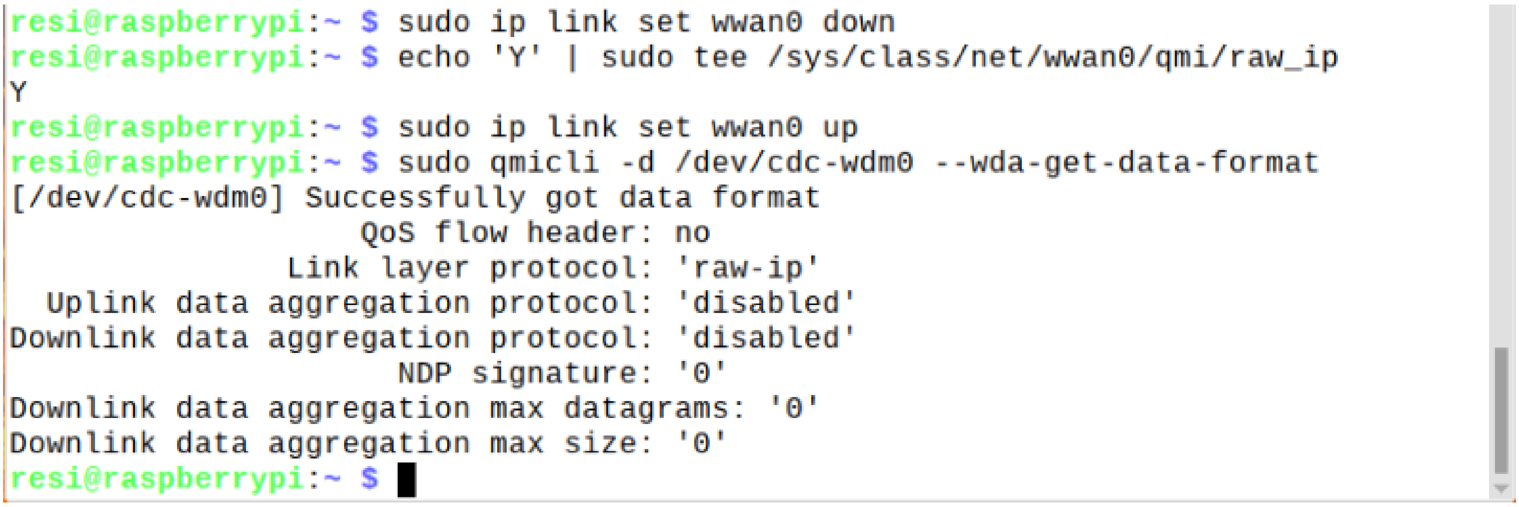
ACTIVATE RAW IP MODE

Now we activate the raw IP mode with this commands

```
sudo ip link set wwan0 down
echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
sudo ip link set wwan0 up
```

Check data format with

```
sudo qmicli -d /dev/cdc-wdm0 --wda-get-data-format
```



```
resi@raspberrypi:~ $ sudo ip link set wwan0 down
resi@raspberrypi:~ $ echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
Y
resi@raspberrypi:~ $ sudo ip link set wwan0 up
resi@raspberrypi:~ $ sudo qmicli -d /dev/cdc-wdm0 --wda-get-data-format
[/dev/cdc-wdm0] Successfully got data format
                   QoS flow header: no
               Link layer protocol: 'raw-ip'
  Uplink data aggregation protocol: 'disabled'
Downlink data aggregation protocol: 'disabled'
                     NDP signature: '0'
Downlink data aggregation max datagrams: '0'
Downlink data aggregation max size: '0'
resi@raspberrypi:~ $ 
```

Setup LTE modem on RESI-C4 controller

SETUP APN NETWORK INFO

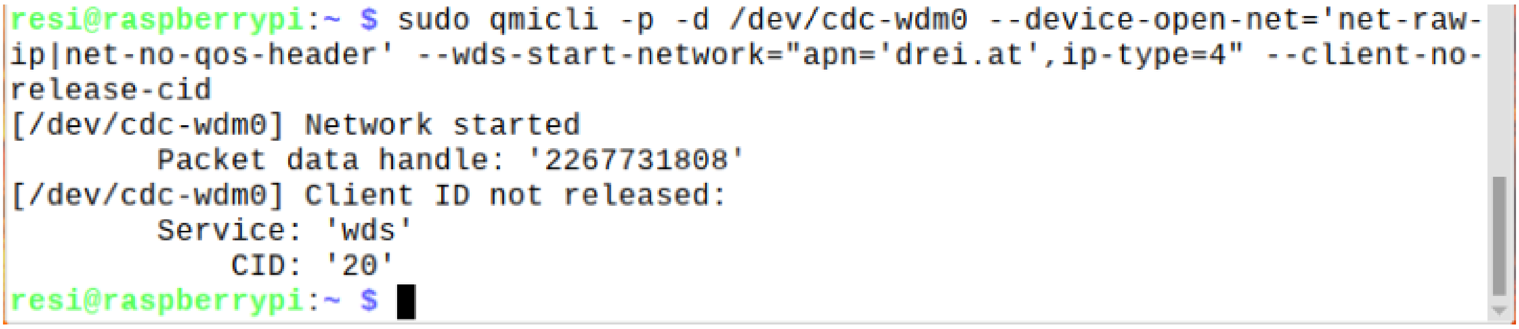
Now we have to setup the APN network info. In our case we use Austria network provider **drei.at**. It needs no username or password. If you need one add to the command:

```
apn='YOUR_APN',username='YOUR_USERNAME',password='YOUR_PASSWORD',ip-type=4
```

We use

```
sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-header' --wds-start-network="apn='drei.at',ip-type=4" -client-no-release-cid
```

Your answer should look like this:

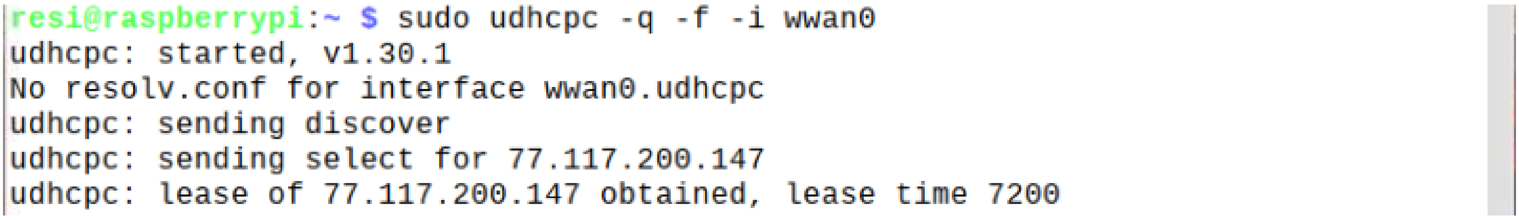


```
resi@raspberrypi:~ $ sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-header' --wds-start-network="apn='drei.at',ip-type=4" --client-no-release-cid
[/dev/cdc-wdm0] Network started
        Packet data handle: '2267731808'
[/dev/cdc-wdm0] Client ID not released:
        Service: 'wds'
            CID: '20'
resi@raspberrypi:~ $
```

CONFIGURE UDHCPC

Now we have to configure udhcpc to assign a default IP address and route:

```
sudo udhcpc -q -f -i wwan0
```

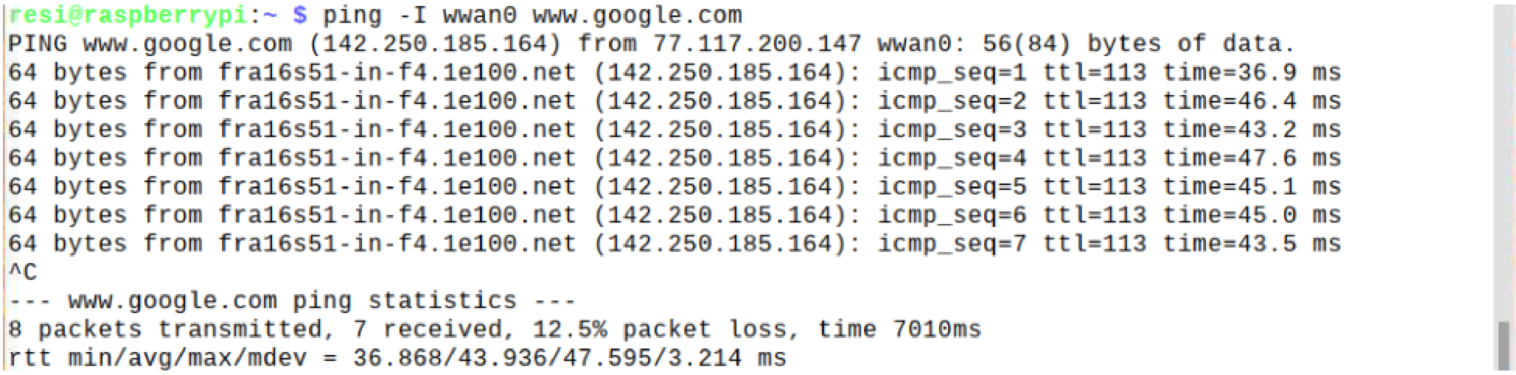


```
resi@raspberrypi:~ $ sudo udhcpc -q -f -i wwan0
udhcpc: started, v1.30.1
No resolv.conf for interface wwan0.udhcpc
udhcpc: sending discover
udhcpc: sending select for 77.117.200.147
udhcpc: lease of 77.117.200.147 obtained, lease time 7200
```

TEST LTE MODEM

Finally we can test the LTE modem. Use ping command

```
ping -I wwan0 www.google.com
```



```
resi@raspberrypi:~ $ ping -I wwan0 www.google.com
PING www.google.com (142.250.185.164) from 77.117.200.147 wwan0: 56(84) bytes of data.
64 bytes from fra16s51-in-f4.1e100.net (142.250.185.164): icmp_seq=1 ttl=113 time=36.9 ms
64 bytes from fra16s51-in-f4.1e100.net (142.250.185.164): icmp_seq=2 ttl=113 time=46.4 ms
64 bytes from fra16s51-in-f4.1e100.net (142.250.185.164): icmp_seq=3 ttl=113 time=43.2 ms
64 bytes from fra16s51-in-f4.1e100.net (142.250.185.164): icmp_seq=4 ttl=113 time=47.6 ms
64 bytes from fra16s51-in-f4.1e100.net (142.250.185.164): icmp_seq=5 ttl=113 time=45.1 ms
64 bytes from fra16s51-in-f4.1e100.net (142.250.185.164): icmp_seq=6 ttl=113 time=45.0 ms
64 bytes from fra16s51-in-f4.1e100.net (142.250.185.164): icmp_seq=7 ttl=113 time=43.5 ms
^C
--- www.google.com ping statistics ---
8 packets transmitted, 7 received, 12.5% packet loss, time 7010ms
rtt min/avg/max/mdev = 36.868/43.936/47.595/3.214 ms
```

Setup LTE modem on RESI-C4 controller

HOWTO SETUP FOR AUTOMATIC CONNECTION AFTER REBOOT

Now the LTE modem runs. But if you boot your machine new, the connection will be vanished. You have top configure it for permanent use.

So you have to rerun the following lines after reboot:

```
sudo ip link set wwan0 down
echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
sudo ip link set wwan0 up
sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-header' --wds-start-network="apn='drei.at',ip-type=4" -client-no-release-cid
sudo udhcpc -q -f -i wwan0
```

MANUALLY STARTING/STOPPING LTE MODEM

Next step, we create a file

```
sudo nano /etc/network/interfaces.d/wwan0
```

Then we put the following content into the new file:

```
iface wwan0 inet manual
     pre-up ifconfig wwan0 down
     pre-up echo Y > /sys/class/net/wwan0/qmi/raw_ip
     pre-up for _ in $(seq 1 10); do /usr/bin/test -c /dev/cdc-wdm0 && break; /bin/sleep 1; done
     pre-up for _ in $(seq 1 10); do /usr/bin/qmicli -d /dev/cdc-wdm0 --nas-get-signal-strength && break; /bin/sleep 1; done
# Don't forget to change to your APN name and user and passsword...
     pre-up sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-header' --wds-start-network="apn='drei.at',ip-type=4" --client-no-release-cid
     pre-up udhcpc -i wwan0
     post-down /usr/bin/qmi-network /dev/cdc-wdm0 stop
```

Now you can restart the system. Depower and repower the device to switch the modem also off and on!

Then you can switch the connection on with

```
sudo ifup wwan0
```

or you switch it off with

```
sudo ifdown wwan0
```

Check the wwan0 interface again with ping

```
ping -I wwan0 www.google.com
```

Setup LTE modem on RESI-C4 controller

AUTOMATICALLY START LTE MODEM AFTER REBOOT

We create or edit the file

```
sudo nano /etc/network/interfaces.d/wwan0
```

Then we put the following content into the file. auto wwan0 will bring the interface up after reboot.

```
auto wwan0
iface wwan0 inet manual
     pre-up sudo ifconfig wwan0 down
     pre-up echo Y > /sys/class/net/wwan0/qmi/raw_ip
     pre-up for _ in $(seq 1 10); do sudo /usr/bin/test -c /dev/cdc-wdm0 && break; /bin/sleep 1; done
     pre-up for _ in $(seq 1 10); do sudo /usr/bin/qmicli -d /dev/cdc-wdm0 --nas-get-signal-strength && break;
/bin/sleep 1; done
# Don't forget to change to your APN name and user and passsword...
     pre-up sudo qmicli -p -d /dev/cdc-wdm0 –device-open-net='net-raw-ip|net-no-qos-header' --wds-start-
network="apn='drei.at',ip-type=4" --client-no-release-cid
     pre-up sudo udhcpc -i wwan0
     post-down sudo /usr/bin/qmi-network /dev/cdc-wdm0 stop
```

For more debugging issues you can use this version too

```
auto wwan0
iface wwan0 inet manual
     pre-up echo "RESI:Starting LTE modem..." | sudo tee /dev/kmsg
     pre-up sudo ifconfig wwan0 down
     pre-up sudo echo Y > /sys/class/net/wwan0/qmi/raw_ip
     pre-up for _ in $(seq 1 10); do sudo /usr/bin/test -c /dev/cdc-wdm0 && break; /bin/sleep 1; done
     pre-up for _ in $(seq 1 10); do sudo /usr/bin/qmicli -d /dev/cdc-wdm0 --nas-get-signal-strength && break;
/bin/sleep 1; done
# Don't forget to change to your APN name and user and passsword...
     pre-up echo "RESI:Starting LTE modem 2..." | sudo tee /dev/kmsg
     pre-up sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-header' --wds-start-
network="apn='drei.at',ip-type=4" --client-no-release-cid
     pre-up echo "RESI:Starting LTE modem 3..." | sudo tee /dev/kmsg
     pre-up sudo udhcpc -i wwan0
     pre-up echo "RESI:Starting LTE modem is ok..." | sudo tee /dev/kmsg
     pre-up sudo qmi-network /dev/cdc-wdm0 status >>/dev/kmsg
     pre-up echo "RESI:Starting LTE modem FINISHED" | sudo tee /dev/kmsg
     post-down echo "RESI:Stopping LTE modem ..." | sudo tee /dev/kmsg
     post-down sudo /usr/bin/qmi-network /dev/cdc-wdm0 stop
     post_down sudo qmi-network /dev/cdc-wdm0 status | sudo tee /dev/kmsg
     post-down echo "RESI:Stopping LTE modem FINISHED" | sudo tee /dev/kmsg
```

check the results with

```
dmesg
```

Now you can restart the system. Depower and repower the device to switch the modem also off and on!
And the connection should establish automatically. Check the wwan0 interface again with ping

```
ping -I wwan0 www.google.com
```

QMI COMMANDS

Here are some useful QMI commands

```
sudo qmicli -d /dev/cdc-wdm0 --nas-get-signal-info
sudo qmicli -d /dev/cdc-wdm0 --nas-get-signal-strength
sudo qmicli -d /dev/cdc-wdm0 --nas-get-home-network
sudo qmicli -d /dev/cdc-wdm0 --nas-get-serving-system
sudo qmi-network /dev/cdc-wdm0 status
sudo qmicli -d /dev/cdc-wdm0  --wds-get-packet-service-status
```

RESI Informatik & Automation GmbH
Altenmarkt 29, A-8551 Wies, AUSTRIA
help@RESI.cc www.RESI.cc

RESI