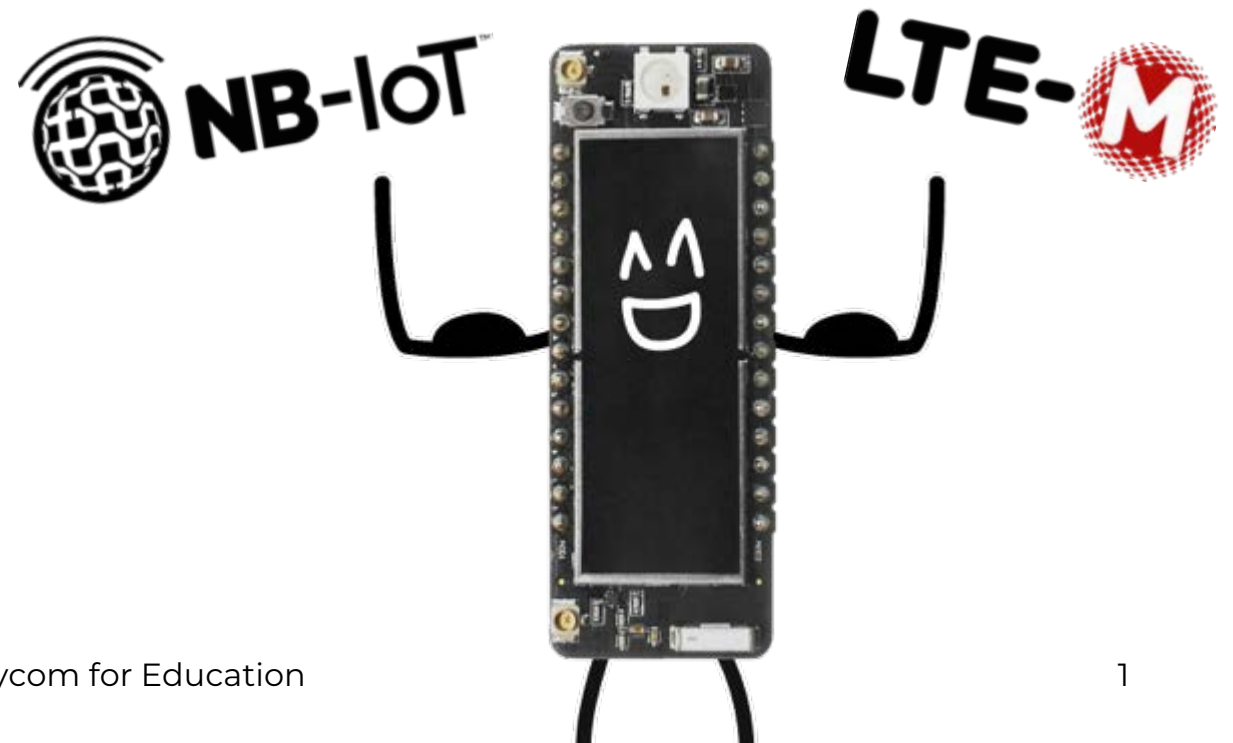


# LESSON 6: GETTING CONNECTED TO CELLULAR

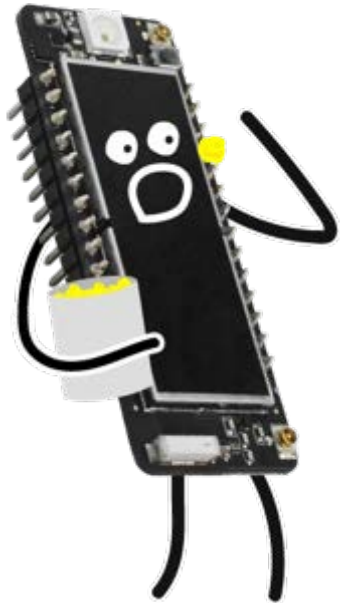
by Pycom



# LESSON OBJECTIVES

- Introduction to LTE
- Get connected with LTE
- Troubleshooting

# INTRODUCTION TO LTE



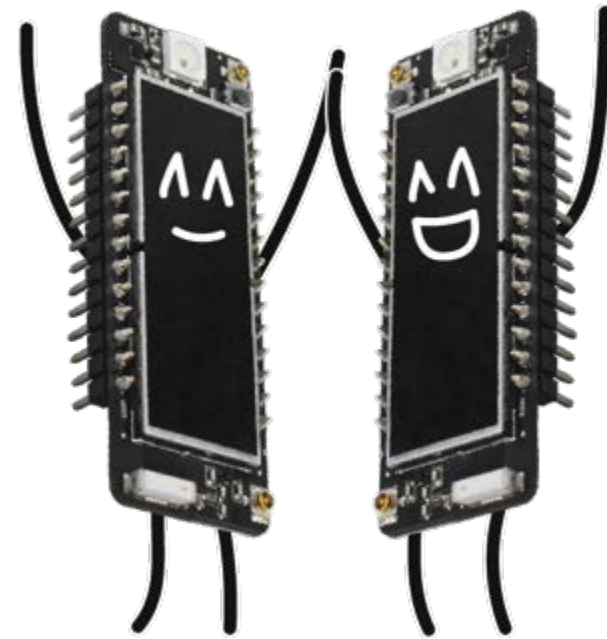
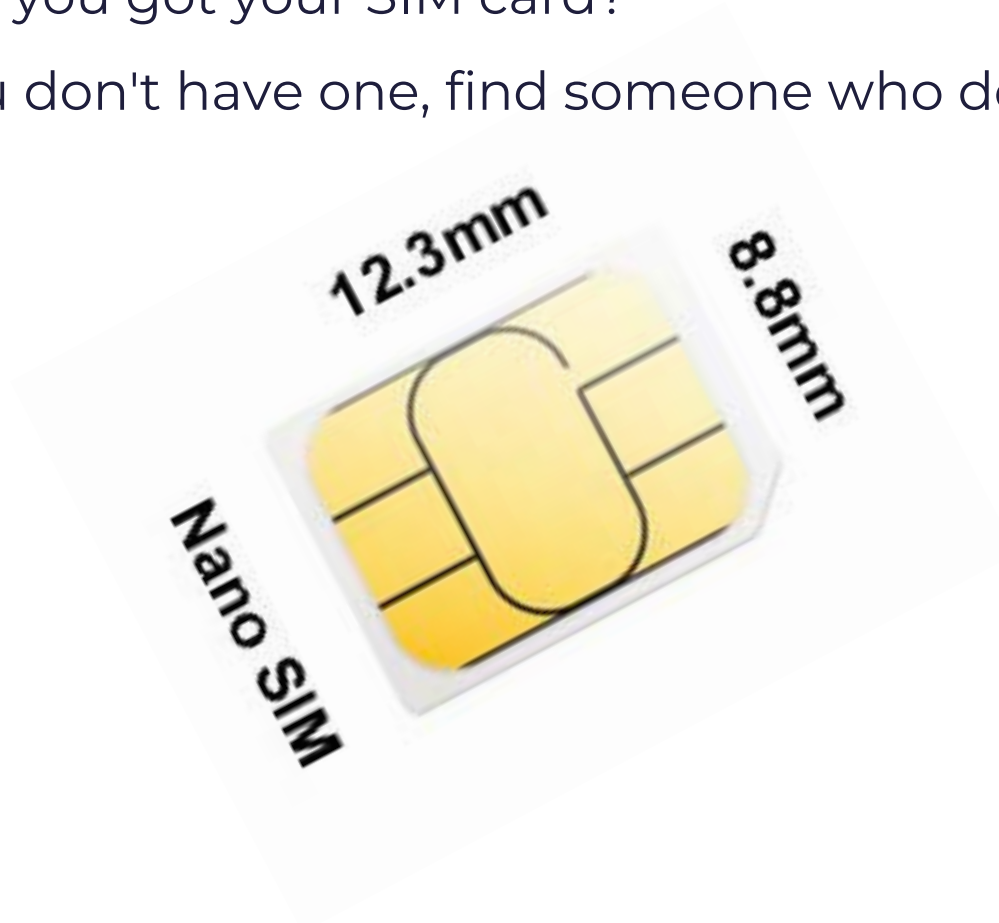
- Normal LTE, as used on your cell phone, is great for high speed and high bandwidths
- But it requires a lot of electric energy to operate!
- For IoT devices, which are more resource restricted, there are variants with a lower bandwidth and lower energy consumption:
  - Cat-M1
  - NB-IoT

<https://docs.pycom.io/tutorials/lte/cat-m1/>

# SIM CARD

Have you got your SIM card?

If you don't have one, find someone who does!



# SIM CARD

- Your SIM provider and contract determines whether you can use NB-IoT or CAT-M1 or both
- The Sequans Monarch chip on the Pycom board must run the matching NB-IoT or CAT-M1 firmware. (See appendix for how to re-flash your modem)
- Your SIM provider also gives you some connection details, like
  - Access Point Name
  - Band(s)
  - White listed IP Addresses ( NB-IoT)

# LTE CONNECTION

The code example shows how to get connected with LTE

To test the connection we download the start page from google.

<https://docs.pycom.io/tutorials/lte/>

```
import socket
import ssl
import time
from network import LTE

lte = LTE()          # instantiate the LTE object
lte.attach()        # attach the cellular modem to a base station
while not lte.isattached():
    time.sleep(0.25)
lte.connect()       # start a data session and obtain an IP address
while not lte.isconnected():
    time.sleep(0.25)

s = socket.socket()
s = ssl.wrap_socket(s)
s.connect(socket.getaddrinfo('www.google.com', 443)[0][-1])
s.send(b"GET / HTTP/1.0\r\n\r\n")
print(s.recv(4096))
s.close()

lte.disconnect()
lte.dettach()
```

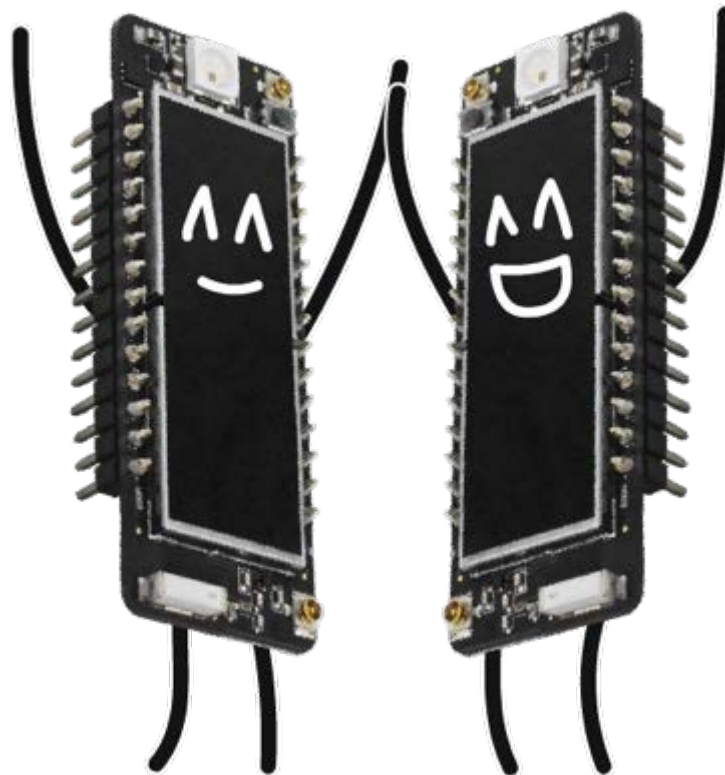
# LTE CONNECTION

To configure the connection parameters for your particular SIM card, add them in the attach() command

```
lte.attach(band=20, apn="nb.inetd.gdsp")
```

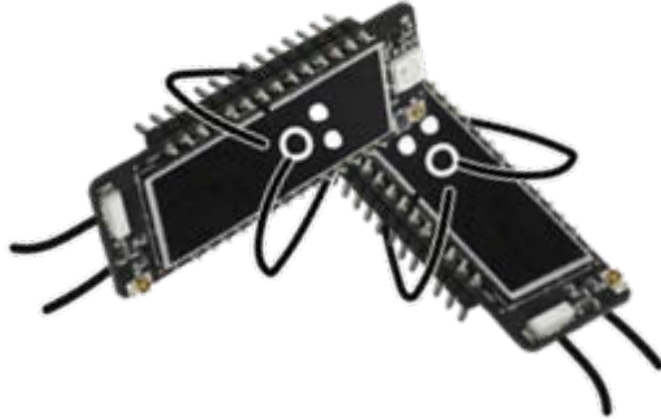
<https://docs.pycom.io/tutorials/lte/>

# YOU'RE UP AND RUNNING ON CELLULAR!





# TROUBLESHOOTING



# UPDATE PYCOM FIRMWARE

- In general you want to run the latest Pycom firmware

- Check the latest firmware release

<https://forum.pycom.io/category/1/announcements-news>

- Check the firmware version of your board by running:

```
>>> import os
>>> os.uname()
(sysname='GPy', nodename='GPy', release='1.17.0.b1', version='v1.8.6-849-d0dc708 on 2018-02-27', machine='GPy with ESP32')
```

# UPDATE SEQUANS FIRMWARE

- The Sequans modem inside the Pycom board has its own firmware
- Keep in mind that there is a different Sequans firmware for
  - NB-IoT
  - CAT-M1
- Match what your SIM card/provider supports!
- There could be a newer version than what's on your board

# UPDATE SEQUANS FIRMWARE

- Latest version can be found on [software.pycom.io/downloads/sequans2.html](https://software.pycom.io/downloads/sequans2.html)
- To download, log in with your Pycom forum credentials

# UPDATE SEQUANS FIRMWARE

Check the Sequans firmware on your Pycom board with

```
sqnsupgrade.info()
```

```
>>> import sqnsupgrade
```

```
>>> sqnsupgrade.info()
```

```
<<< Welcome to the SQN3330 firmware updater [1.2.6] >>>
```

```
>>> FiPy with firmware version 1.20.2.rc6
```

```
Your modem is in application mode. Here is the current version:
```

```
UE6.0.0.0
```

```
LR6.0.0.0-44556
```

# UPDATE SEQUANS FIRMWARE

- The easiest way to update the Sequans firmware is via an SD card
- Make sure the SD card is formatted with the FAT filesystem
- Unzip the package  
(e.g. **CATM1-39529.zip** or **NB1-37781.zip**)
- Place the files on the SD card  
(e.g. **CATM1-39529.dup** or **NB1-37781.dup**)

# UPDATE SEQUANS FIRMWARE

- Put the SD card into your expansion board
- Verify that you can see the .dup file with this code snippet

```
from machine import SD

sd = SD()
os.mkfs(sd)           # format SD card
os.mount(sd, '/sd')  # mount it
os.listdir('/sd')    # list its content
```

<https://docs.pycom.io/firmwareapi/pycom/network/lte/>

# UPDATE SEQUANS FIRMWARE

- Now you're ready to flash the firmware with the code snippet below
- Note:
  - The update may 'stall' (around 7-10% and again at 99%).
  - This is not an indication of failure, it just takes some time.
  - Do not interrupt the process as you will have to start again!

```
import sqnsupgrade
sqnsupgrade.run('/sd/CATM1-39529.dup', '/sd/updater.elf')
```

<https://docs.pycom.io/tutorials/lte/firmware/>



# UPDATE SEQUANS FIRMWARE

The full output from the update will look like this:



<https://docs.pycom.io/tutorials/lte/firmware/>

27/04/2020

Created by Pycom for I

```
<<< Welcome to the SQN3330 firmware updater >>>
Attempting AT wakeup...
Starting STP (DO NOT DISCONNECT POWER!!!)
Session opened: version 1, max transfer 8192 bytes
Sending 54854 bytes: [#####] 100%
Bootrom updated successfully, switching to upgrade mode
Attempting AT auto-negotiation...
Session opened: version 1, max transfer 2048 bytes
Sending 306076 bytes: [#####] 100%
Attempting AT wakeup...
Upgrader loaded successfully, modem is in upgrade mode
Attempting AT wakeup...
Starting STP ON_THE_FLY
Session opened: version 1, max transfer 8192 bytes
Sending 5996938 bytes: [#####] 100%
Code download done, returning to user mode
Resetting (DO NOT DISCONNECT POWER!!!).....
Upgrade completed!
Here's the current firmware version:

SYSTEM VERSION
=====

FIRMWARE VERSION
  Bootloader0 : 5.1.1.0 [33080]
  Bootloader1 : 5.1.1.0 [38638]
  Bootloader2* : 5.1.1.0 [38638]
  NV Info     : 1.1,0,0
  Software    : 5.1.1.0 [38638] by robot-soft at 2018-08-20 09:51:46
  UE          : 5.0.0.0d

COMPONENTS
  ZSP0       : 1.0.99-13604
  ZSP1       : 1.0.99-12341
```

# UPDATE SEQUANS FIRMWARE

- In case of a failure or interruption of the update process:
  - First do a hard reset, i.e disconnecting and reconnecting power  
Pressing the reset button is not enough!
  - Then you can repeat the steps from the previous slides
- Note: On the website there are more details and other ways of updating than via SD card  
[docs.pycom.io/tutorials/lte/firmware](https://docs.pycom.io/tutorials/lte/firmware)

<https://docs.pycom.io/tutorials/lte/firmware/>

# MODULE IMEI

Run the following code to get your module's IMEI number:



```
from network import LTE
lte = LTE()
lte.send_at_cmd('AT+CGSN=1')
```

You'll get a return string like this,

```
\r\n+CGSN: "354347xxxxxxxx"\r\n\r\nOK
```

<https://docs.pycom.io/tutorials/lte/imei/>