# LESSON 5: GETTING CONNECTED WITH SIGFOX

by Pycom

# LESSON OBJECTIVES

- Introduction to Sigfox
- Create a Sigfox account and register the device
- Send Sigfox messages
- Troubleshooting

https://docs.pycom.io/firmwareapi/pycom/network/sigfox/

# SIGFOX - INTRODUCTION

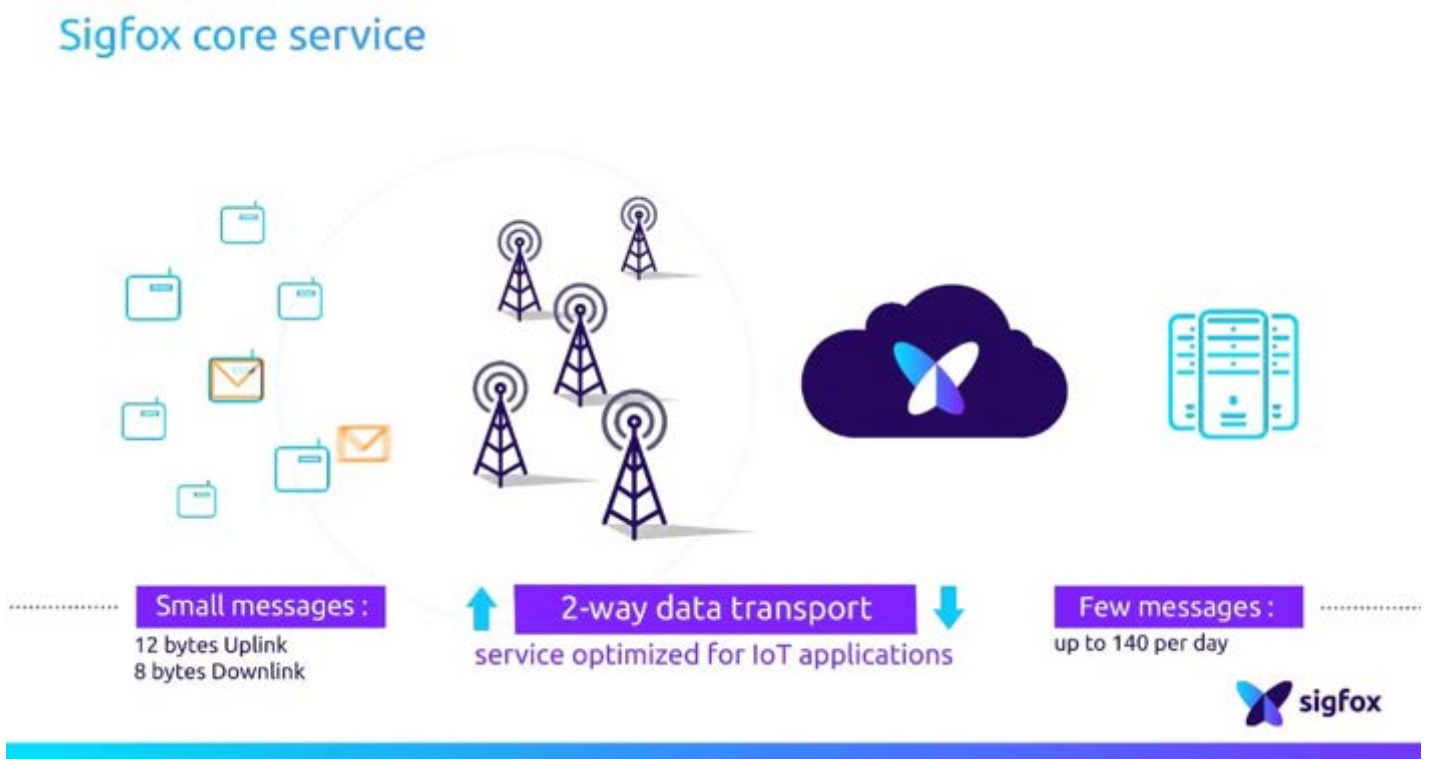
- Sigfox is a Low Power Wide Area Network that enables remote devices to connect using ultra-narrow band (UNB technology)
- It uses the Industrial, Scientific and Medical radio band
- It uses very narrow bits of spectrum, which minimises the effect of noise


(Image source :picto_sigfox-EN.jpg)

https://docs.pycom.io/firmwareapi/pycom/network/sigfox/

# SIGFOX - INTRODUCTION

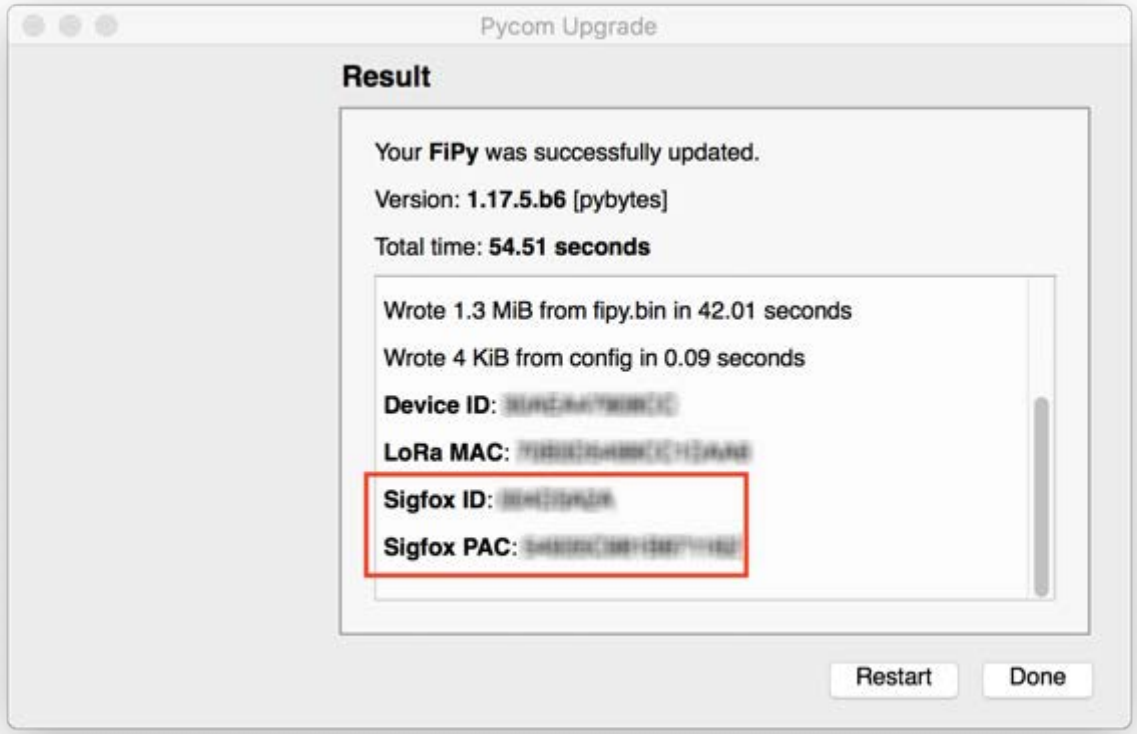
- A Sigfox message is limited to **12-bytes**!
- Also, the accounts are limited to 140 messages per **day**
- So think wisely about what you need to send!
- But rest assured it will be very power efficient




https://www.elektormagazine.com/news/review-wireless-modules-from-pycom/15521

# FIRMWARE UPDATE AND SIGFOX ID

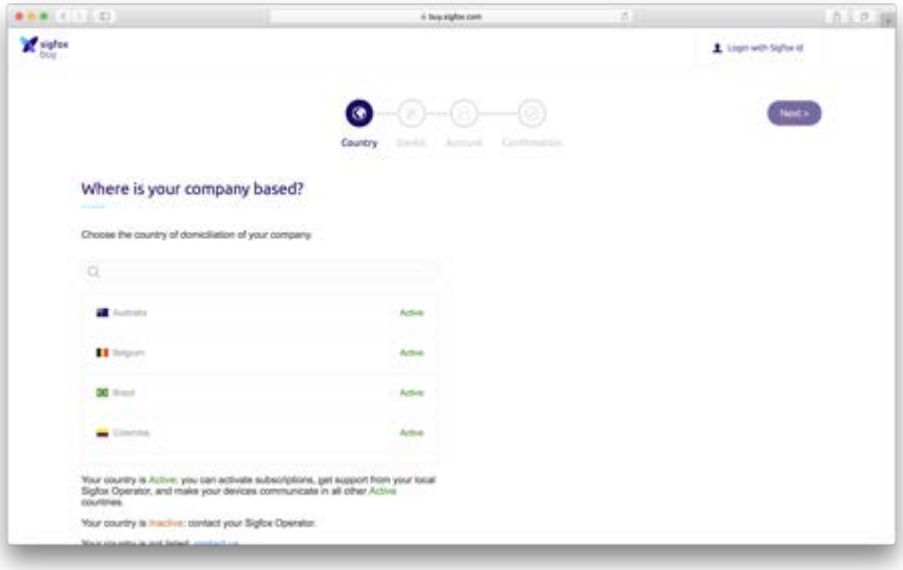
1. Use the Firmware Updater
2. Update your device to the latest firmware
3. At the end of the update
   - **a Sigfox ID** and
   - **a Sigfox PAC** are assigned to your device
4. Copy the **ID** and the **PAC** from the last screen of the Firmware Updater



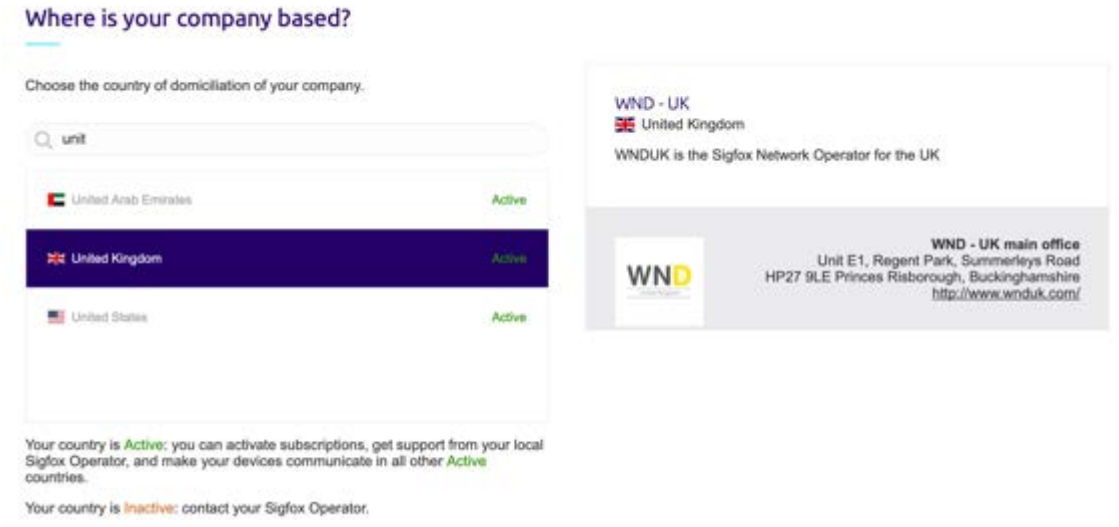

https://docs.pycom.io/gettingstarted/registration/sigfox/

# CREATE AN ACCOUNT ON SIGFOX BACKEND

1. Register to the Sigfox Backend - https://buy.sigfox.com/activate

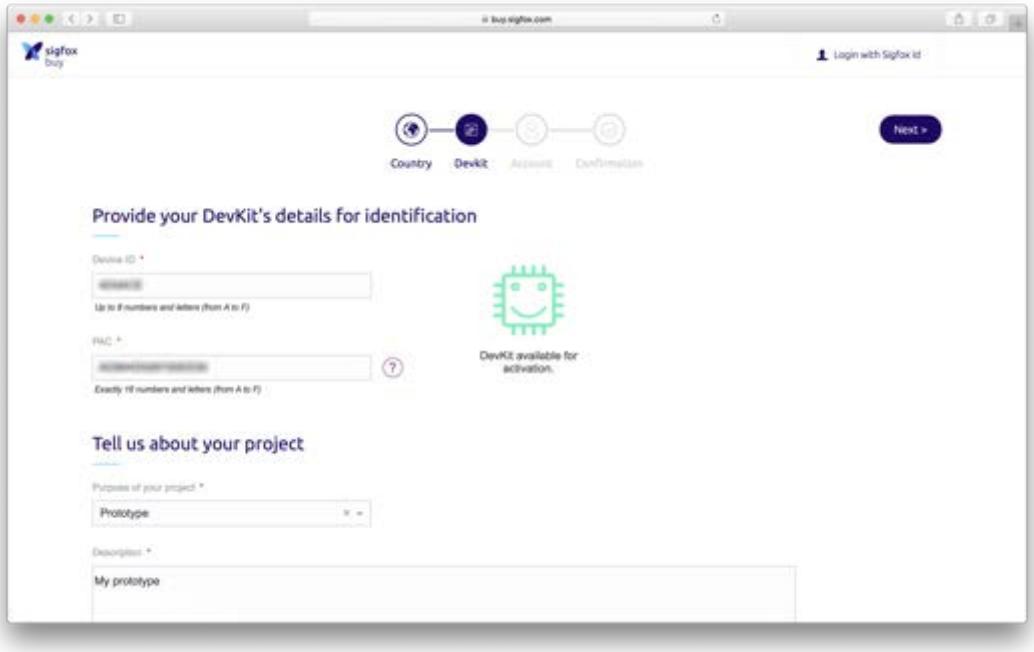



2. Enter your country

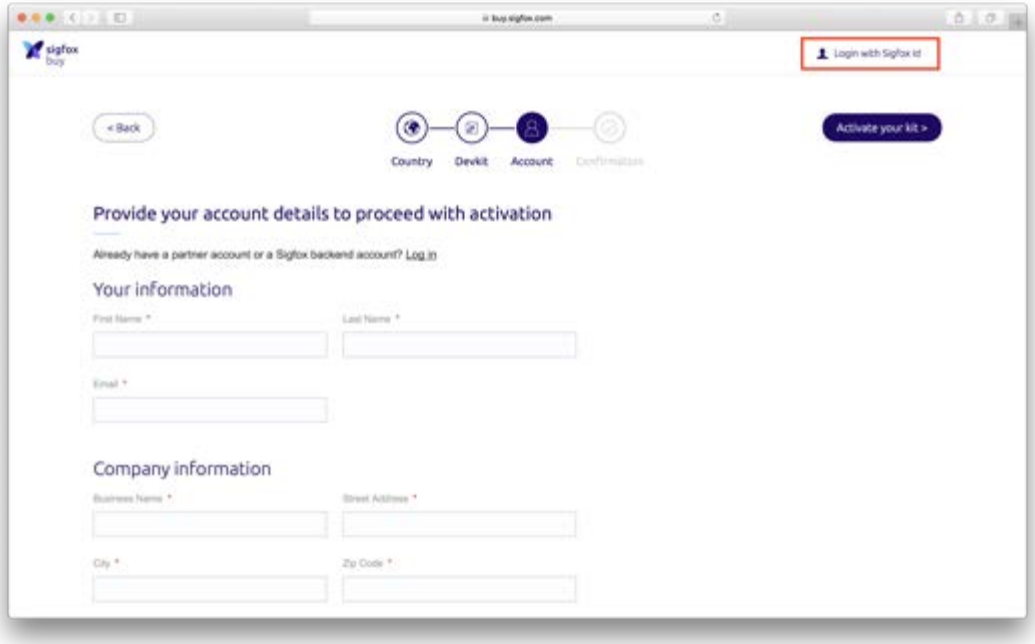


# CREATE AN ACCOUNT ON SIGFOX BACKEND

4. The correct **Sigfox ID** and **Sigfox PAC** should prompt a green microchip



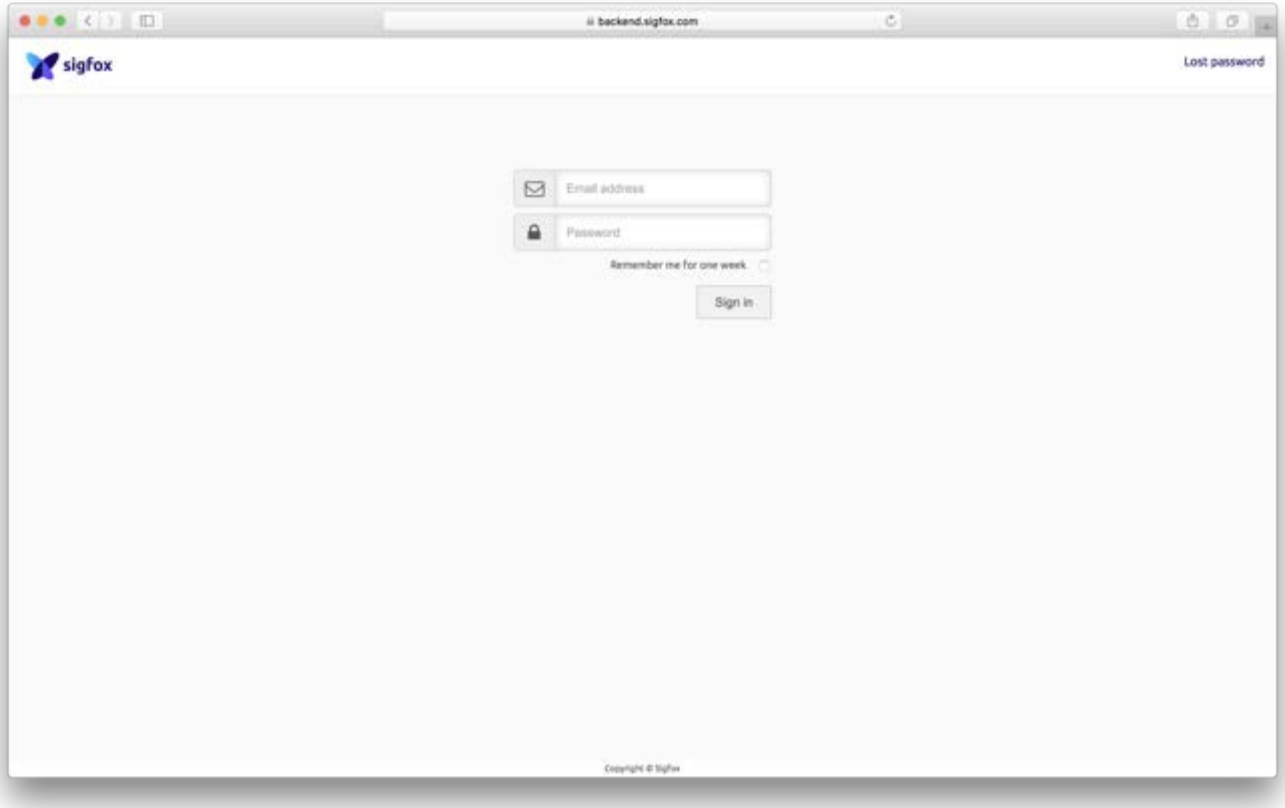

- 5. Provide the required information




https://docs.pycom.io/gettingstarted/registration/sigfox/

# CREATE AN ACCOUNT ON SIGFOX BACKEND

6. After registration, you will receive a confirmation email with a password to Sigfox's backend (https://backend.sigfox.com/auth/login)



# IF YOU ALREADY HAVE A SIGFOX ACCOUNT...

- Make sure that you use the same login details
- That way, all of your devices will be added to the same Sigfox Account

# SENDING A MESSAGE

- Use this code to send your first message

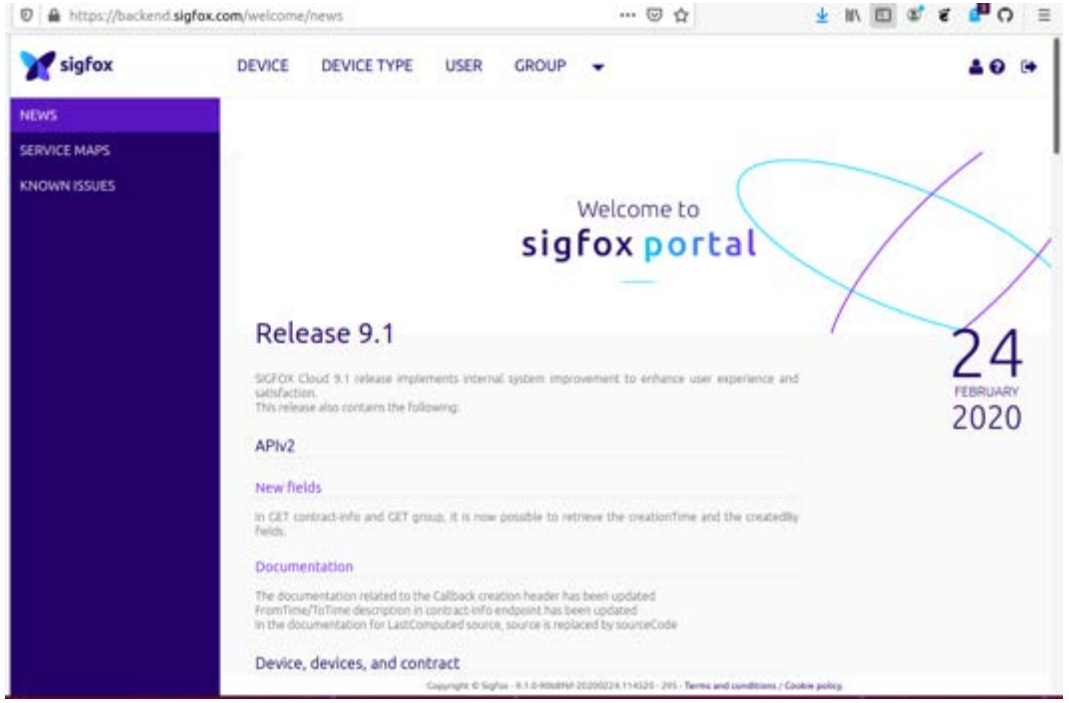
- set the "Region" depending on your geographic location:
  - RCZ1 for Europe
  - RCZ2 for USA, Mexico & Brazil
  - RCZ3 for  Japan
  - RCZ4 for Australia, New Zealand, Singapore, Taiwan, Hong Kong, Colombia & Argentina

```
 from network import Sigfox
import socket

# init Sigfox for RCZ1 (Europe)
sigfox = Sigfox(mode=Sigfox.SIGFOX, rcz=Sigfox.RCZ1)

# create a Sigfox socket
s = socket.socket(socket.AF_SIGFOX, socket.SOCK_RAW)

# make the socket blocking
s.setblocking(True)

# configure it as uplink only
s.setsockopt(socket.SOL_SIGFOX, socket.SO_RX, False)

# send some bytes
s.send(bytes([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]))
```

https://docs.pycom.io/firmwareapi/pycom/network/sigfox/

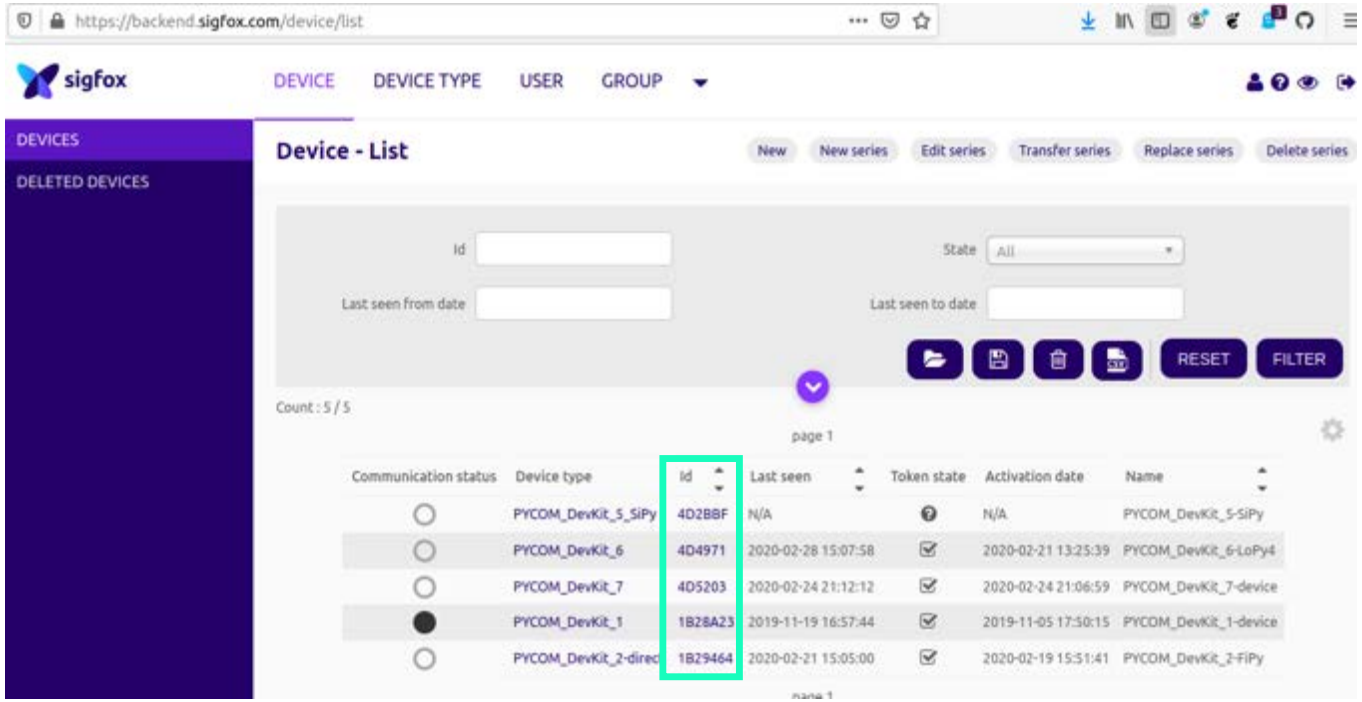
# VERIFYING THE SEND

1. Once you log in, you should see a page similar to the one below



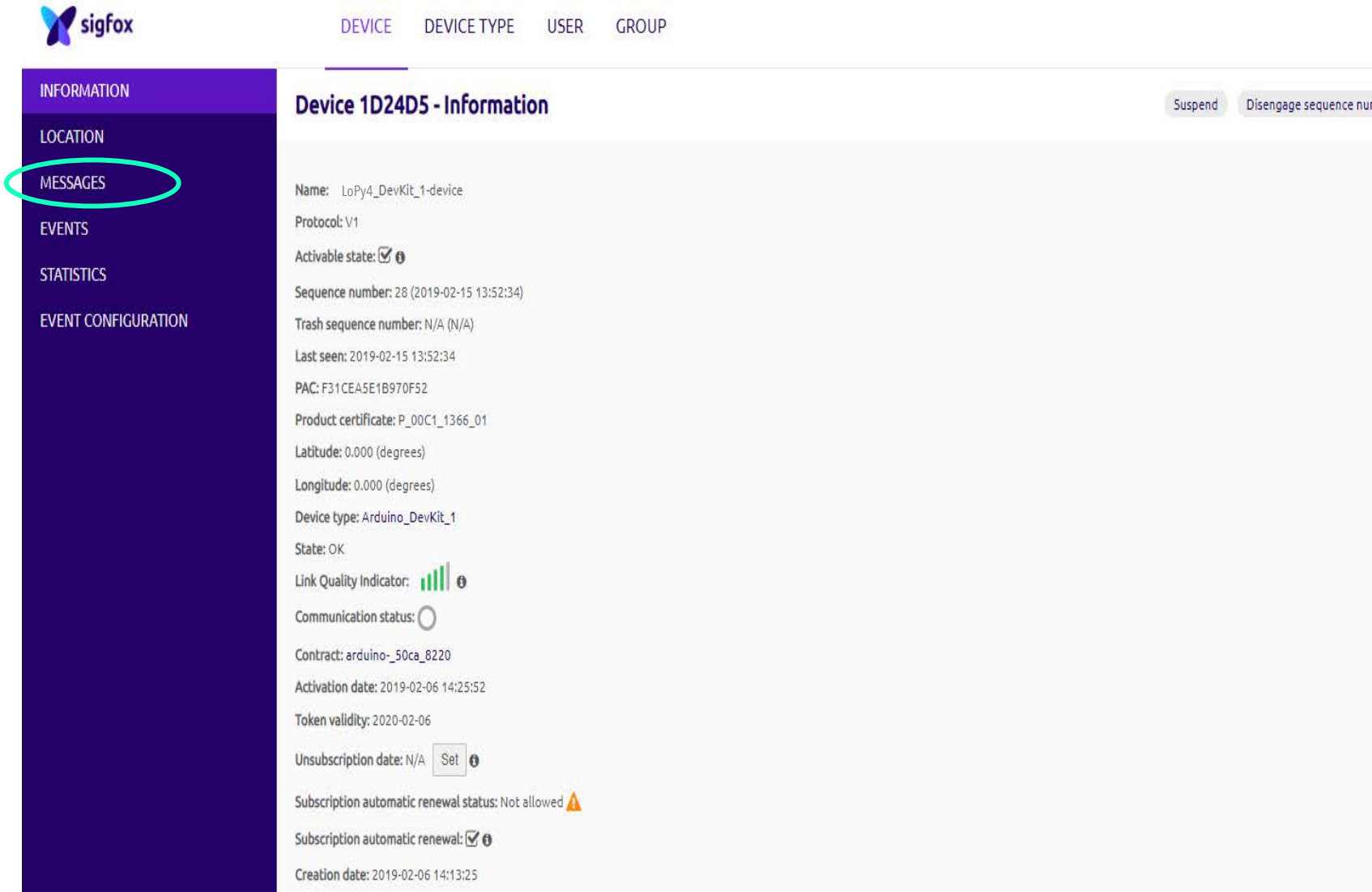
# VERIFYING THE SEND

2. Click on the ‘Device’ tab in the Sigfox backend and click on the hexadecimal device ID



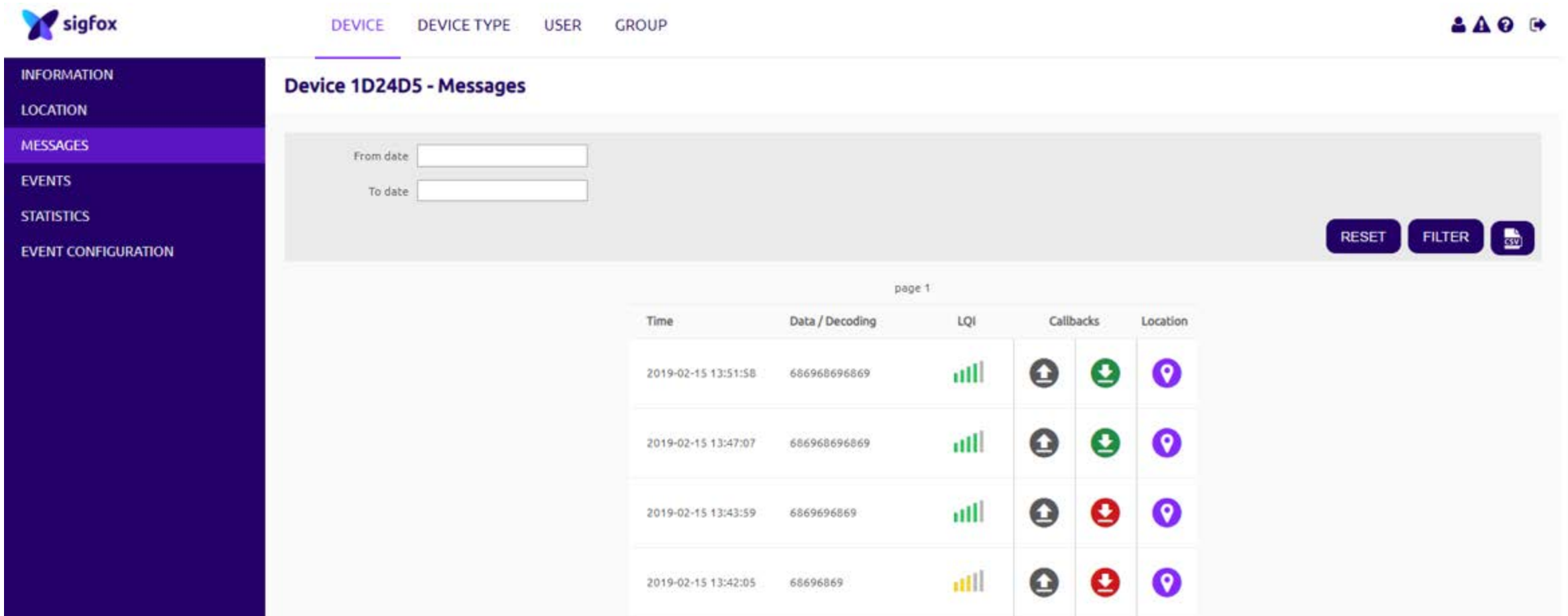
# VERIFYING THE SEND

3. Click on *Messages* in the left menu



# VERIFYING THE SEND

4.Now you should see your messages (hex numbers) together with timestamps and icons indicating the transmission status



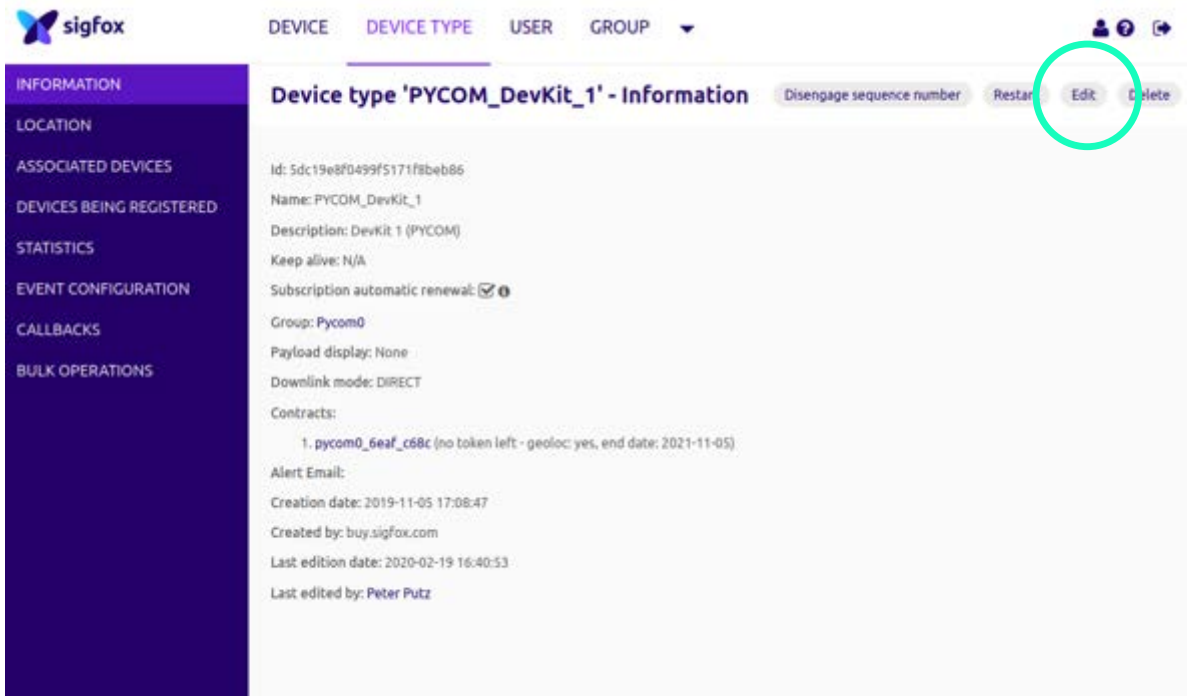
# RECEIVING A MESSAGE

- So far, we've **sent** a message from the Pycom device to Sigfox.
- Now, we will **receive** a message from Sigfox.
- We will demo the downlink, by configuring the Sigfox backend to always send one particular message in response

https://docs.pycom.io/firmwareapi/pycom/network/sigfox/

# CONFIGURE THE DOWNLINK

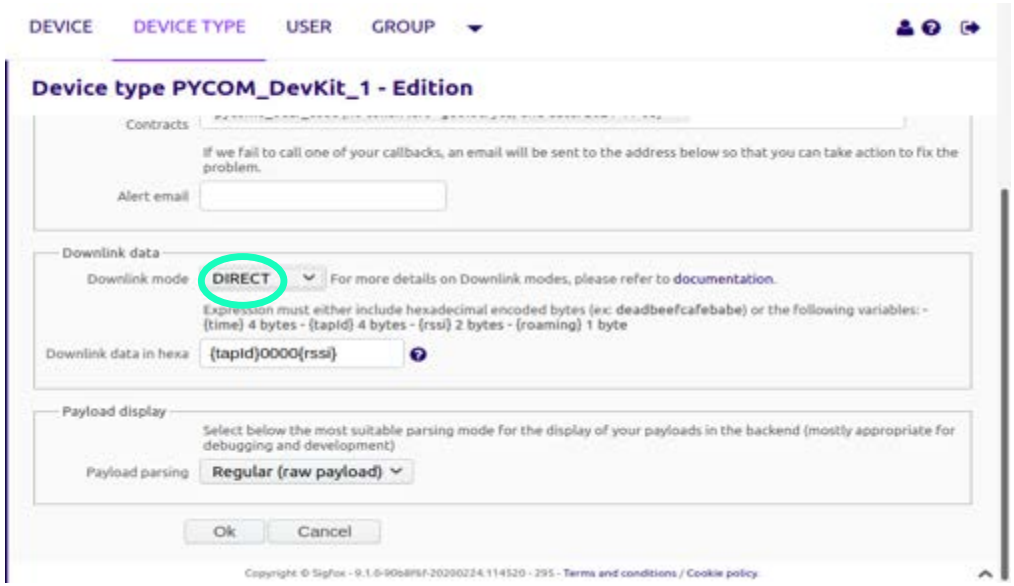
1.Go to your Device Type and click on Edit



# CONFIGURE THE DOWNLINK

2. Change the *Downlink mode* to *DIRECT*

3. Configure the *Downlink* data in hex, e.g. *{tapId}0000{rssi}*



# RECEIVING A MESSAGE

4. Send a Sigfox message again, but this time:

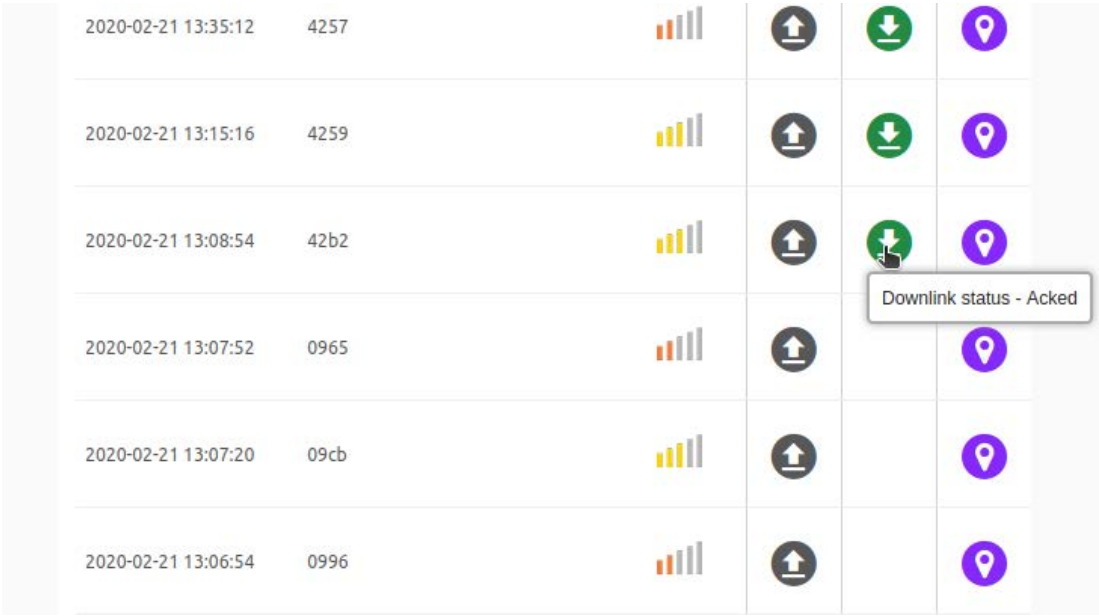
- Include a downlink request, by using True as the third setsockopt() parameter
- Add a s.recv()

```
 # init Sigfox for RCZ1 (Europe)
sigfox = Sigfox(mode=Sigfox.SIGFOX, rcz=Sigfox.RCZ1)

# create a Sigfox socket
s = socket.socket(socket.AF_SIGFOX, socket.SOCK_RAW)

# make the socket blocking
s.setblocking(True)

# configure it as DOWNLINK specified by 'True'
s.setsockopt(socket.SOL_SIGFOX, socket.SO_RX, True)

# send some bytes and request DOWNLINK
s.send(bytes([1, 2, 3]))

# await DOWNLINK message
s.recv(32)
```

https://docs.pycom.io/firmwareapi/pycom/network/sigfox/

# VERIFYING THE RECEPTION

- On the Sigfox backend you will see the downlink status as an icon and afterwards,
- You will receive the message on your Pycom device




**Congrats, you've sent uplink and downlink messages with Sigfox!**

https://docs.pycom.io/firmwareapi/pycom/network/sigfox/

# (A LITTLE NOTE BEFORE YOU GO...)

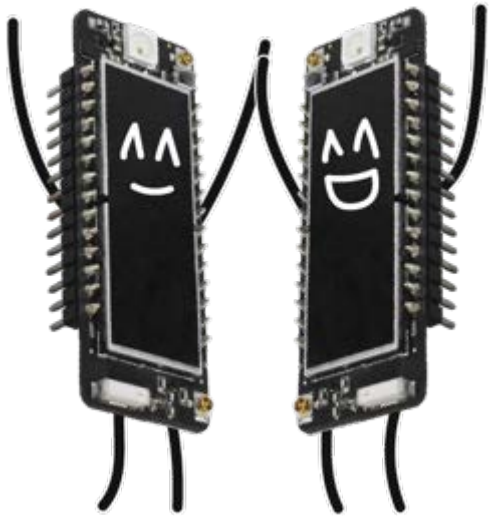
For the next lesson, you'll be connecting to a cellular network. You'll need to get a SIM card from a local provider.

Please note that this might differ from a standard SIM as our devices do not support standard LTE

We recommend contacting your local cellular providers to check out what their plans are surrounding LTE CAT-M1 and NB-IoT.

Click here to find a map of deployed networks and open labs here.

# HAVE A PLAY AROUND SEE WHAT HAPPENS!



Problems? Click here

# TROUBLESHOOTING

Problems? Click here

# DISENGAGE THE SEQUENCE NUMBER

- If you are experiencing connectivity issues, this could be due to the sequence number being out of sync
- If there is a large difference between the sequence number sent by the device and the one expected by the backend, the message is dropped by the network

https://docs.pycom.io/tutorials/sigfox/

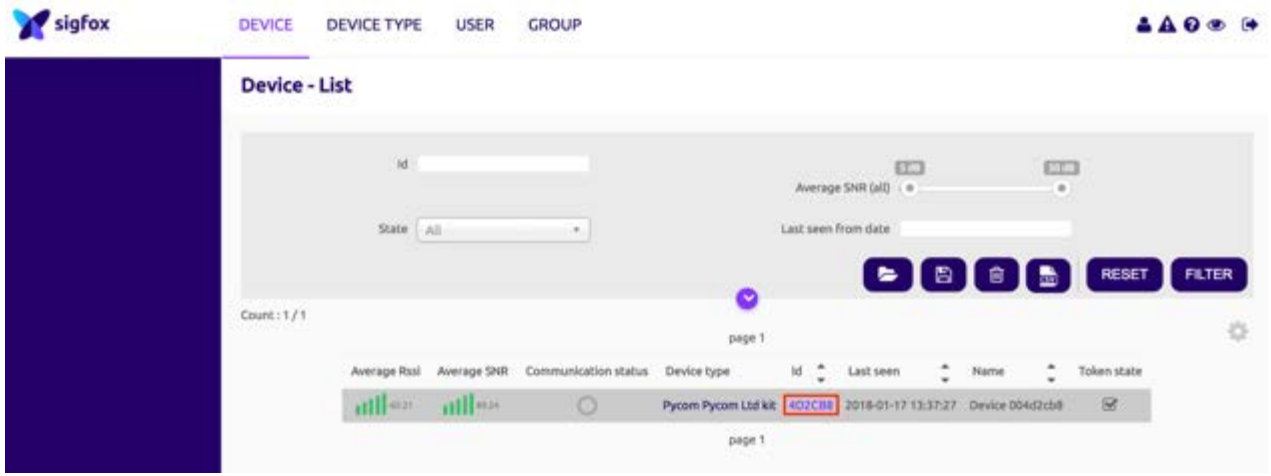
# DISENGAGE THE SEQUENCE NUMBER

- You can use the **Disengage Sequence Number** button on *Device Information*
- Alternatively, on the *Device Type* information page it is possible to reset the number expected by the backend

https://docs.pycom.io/tutorials/sigfox/

# DISENGAGE THE SEQUENCE NUMBER

1. Log into the Sigfox Backend
2. Navigate to **Device**
3. Click on the **Sigfox ID** of the affected device

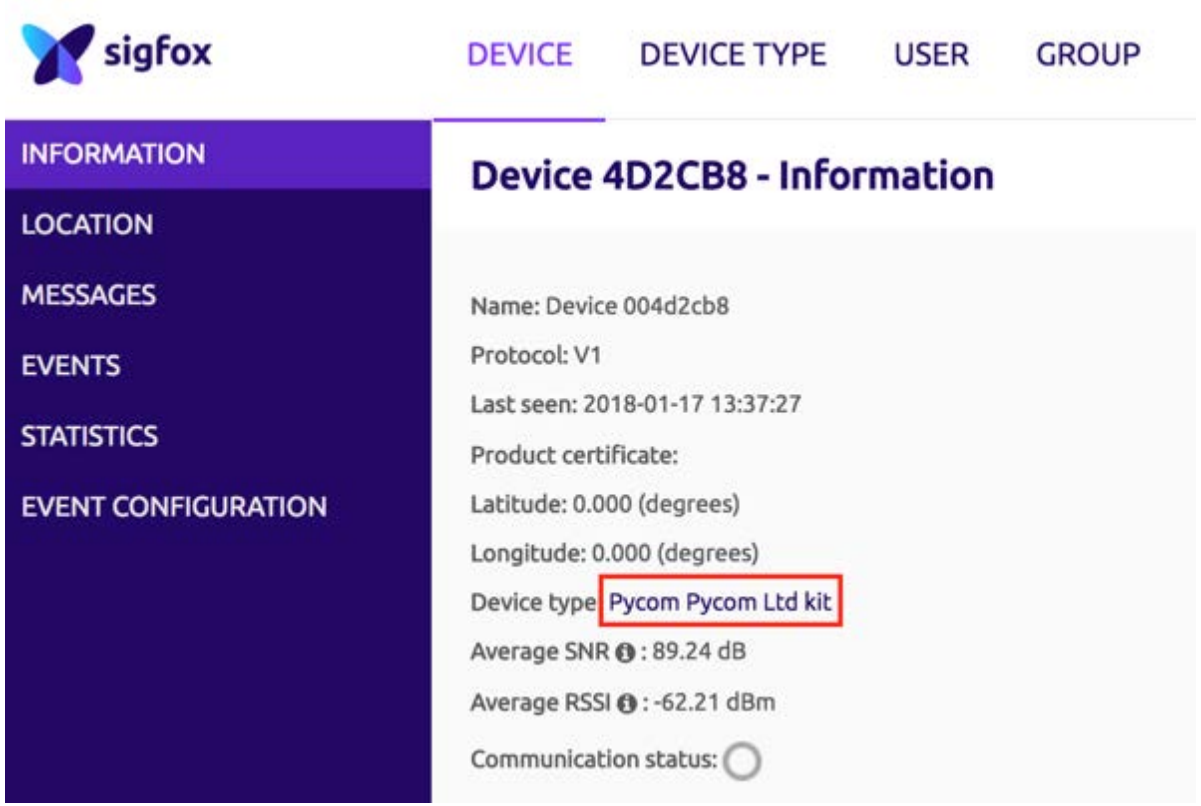



https://docs.pycom.io/tutorials/sigfox/

# DISENGAGE THE SEQUENCE NUMBER

4. The **Information** page should now be accessible

5. The 'Device Type' should be followed by a link – follow it




https://docs.pycom.io/tutorials/sigfox/

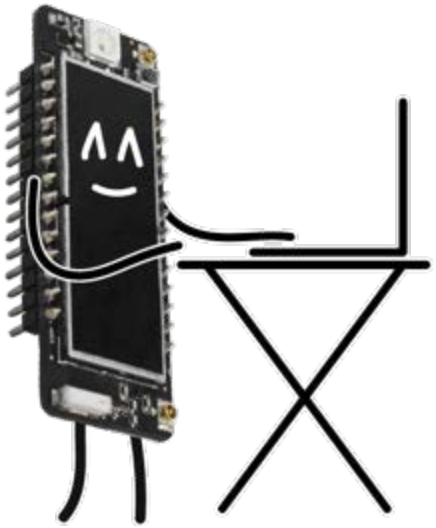
# DISENGAGE THE SEQUENCE NUMBER

6. On the page, click on **Disengage Sequence Number** (located in the upper right corner)


https://docs.pycom.io/tutorials/sigfox/

# ID AND PAC AGAIN

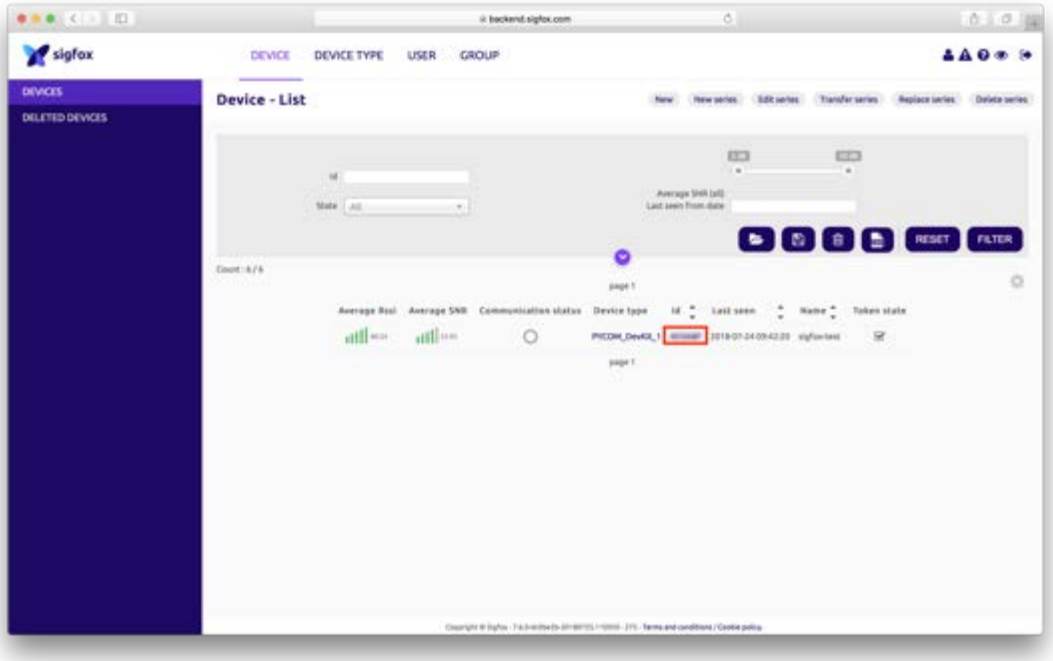
- If you want to get your **Sigfox ID** and **Sigfox PAC** after the firmware update, run these commands via the REPL



```
from network import Sigfox
import binascii

# initalise Sigfox for RCZ1 (You may need a different RCZ Region)
sigfox = Sigfox(mode=Sigfox.SIGFOX, rcz=Sigfox.RCZ1)

# print Sigfox Device ID
print(binascii.hexlify(sigfox.id()))

# print Sigfox PAC number
print(binascii.hexlify(sigfox.pac()))
```

# TRANSFERRING A DEVICE TO ANOTHER SIGFOX ACCOUNT

1. Navigate to your device on Sigfox backend
2. Click on **Devices**
3. Click on the **Sigfox ID** of the device you want to transfer

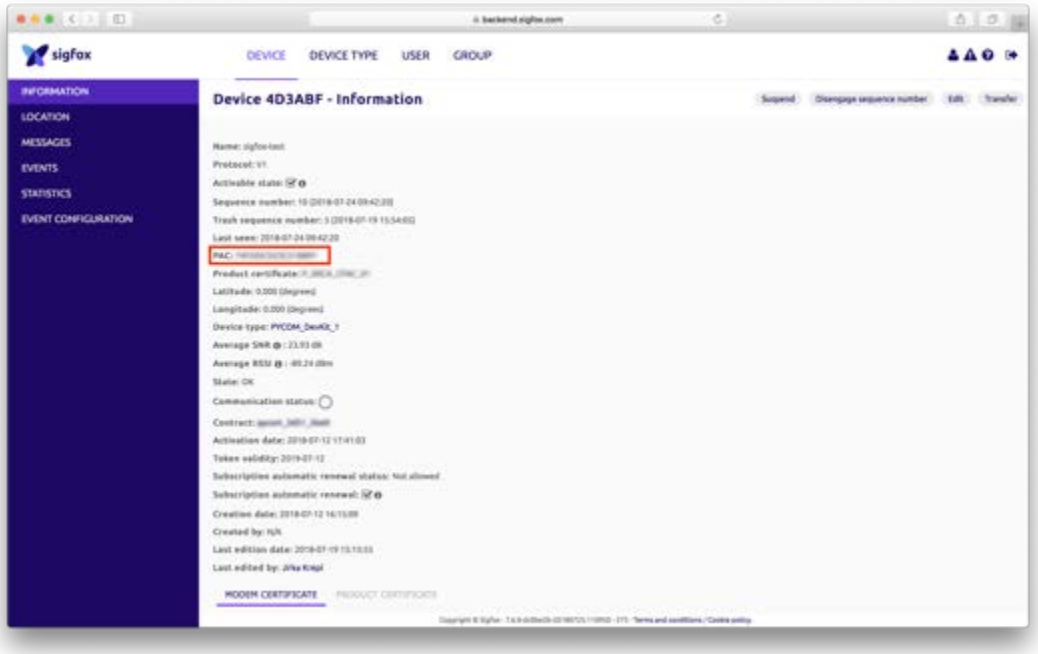



https://docs.pycom.io/gettingstarted/registration/sigfox/

# TRANSFERRING A DEVICE TO ANOTHER SIGFOX ACCOUNT

4. The new **Sigfox PAC** should be shown

5. Once you know your new ***Sigfox PAC*** go to https://buy.sigfox.com/activate and register with the other account




https://docs.pycom.io/gettingstarted/registration/sigfox/

# HAVE A PLAY AROUND AND SEE WHAT HAPPENS!