# nanogateway lopy4 add node APB

cmonaton

August 2019

## 1 Introduction

But : Connecter un device à The Things Network par la méthode Activation By Personalisation.

Prérequis : savoir utiliser The Thing Network, cf tuto *nanogateway_lopy4_lorawan*

Carte : pycom lopy 4 avec expansion board V3.0


Figure 1: pycom lopy 4

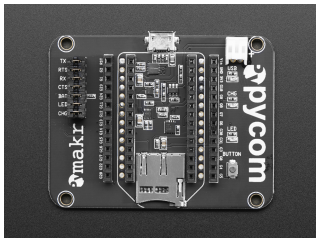



Figure 2: expansion board v3.0

Figure 3: antenne LoRa

## 2 Matériel

Branchez l'antenne LoRa avant d'alimenter la carte sinon la carte grille

## 3 Code pour Activation By Personalisation

**Le code se trouve à :** `https://github.com/GitClementtest/node_apb`

Informations complémentaires : https://docs.pycom.io/tutorials/lora/lorawan-nano-gateway/

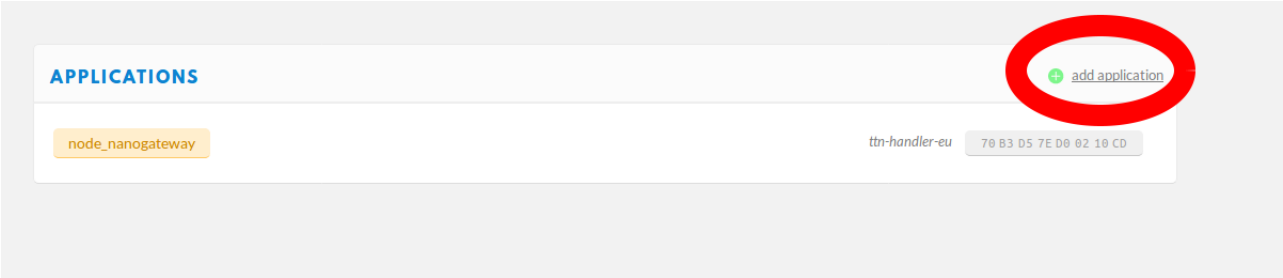
## 3.1 Activation By Personalisation

```
""" ABP Node example compatible with the LoPy Nano Gateway """

from network import LoRa
import socket
import ubinascii
import struct
import time

# Initialise LoRa in LORAWAN mode.
lora = LoRa(mode=LoRa.LORAWAN,region=LoRa.EU868)

# create an ABP authentication params
dev_addr = struct.unpack(">l", ubinascii.unhexlify('2601147D'))[0] # these settings can
#be found from TTN
nwk_swkey = ubinascii.unhexlify('3C74F4F40CAE2221303BC24284FCF3AF') # these settings can
#be found from TTN
app_swkey = ubinascii.unhexlify('0FFA7072CC6FF69A102A0F39BEB0880F') # these settings can
#be found from TTN

# join a network using ABP (Activation By Personalisation)
lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))

# remove all the non-default channels
for i in range(3, 16):
    lora.remove_channel(i)

# set the 3 default channels to the same frequency
lora.add_channel(0, frequency=868100000, dr_min=0, dr_max=5)
lora.add_channel(1, frequency=868100000, dr_min=0, dr_max=5)
lora.add_channel(2, frequency=868100000, dr_min=0, dr_max=5)

# create a LoRa socket
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)

# set the LoRaWAN data rate
s.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)

# make the socket non-blocking
s.setblocking(False)
```

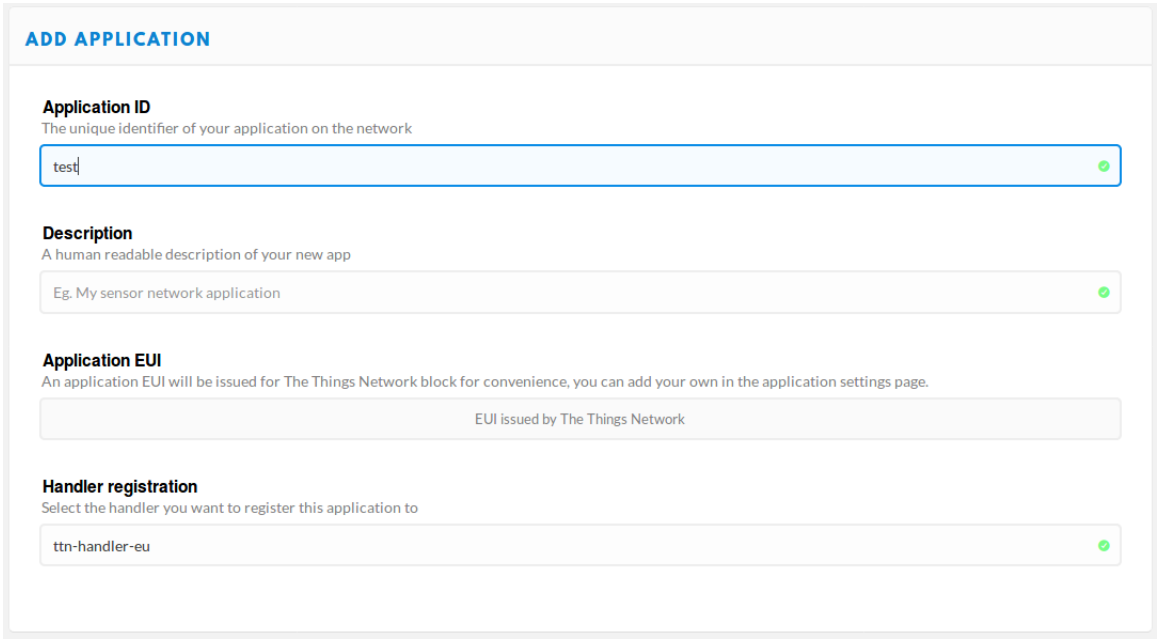
```
""" Your own code can be written below! """

for i in range (200):
    s.send(b'PKT #' + bytes([i]))
    time.sleep(4)
    rx = s.recv(256)
    if rx:
        print(rx)
    time.sleep(6)
```

### 3.2 Créer une application sur The Thing Network

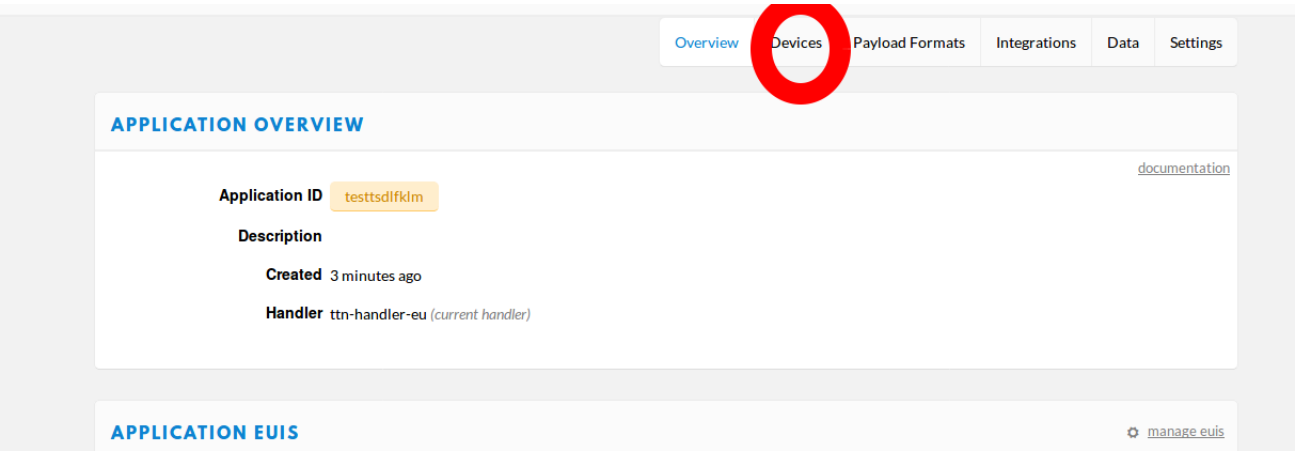
- Créez une application Depuis *Console* sélectionner application, puis ajoutez une application :



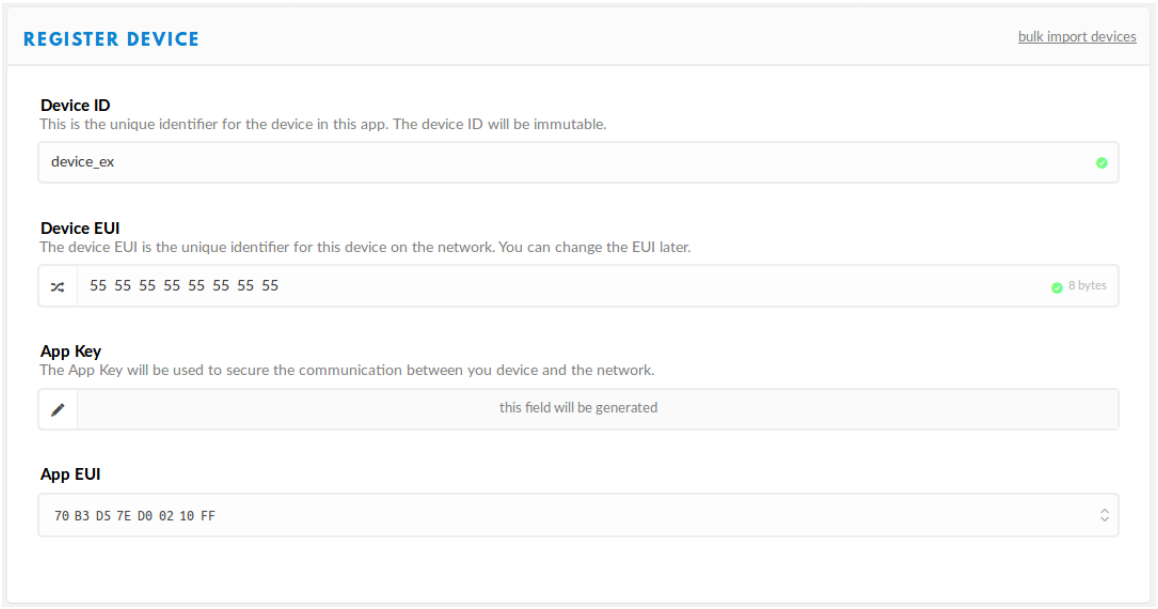

Donnez un nom pour l'application ID :




### 3.3 Ajouter un device dans l'application

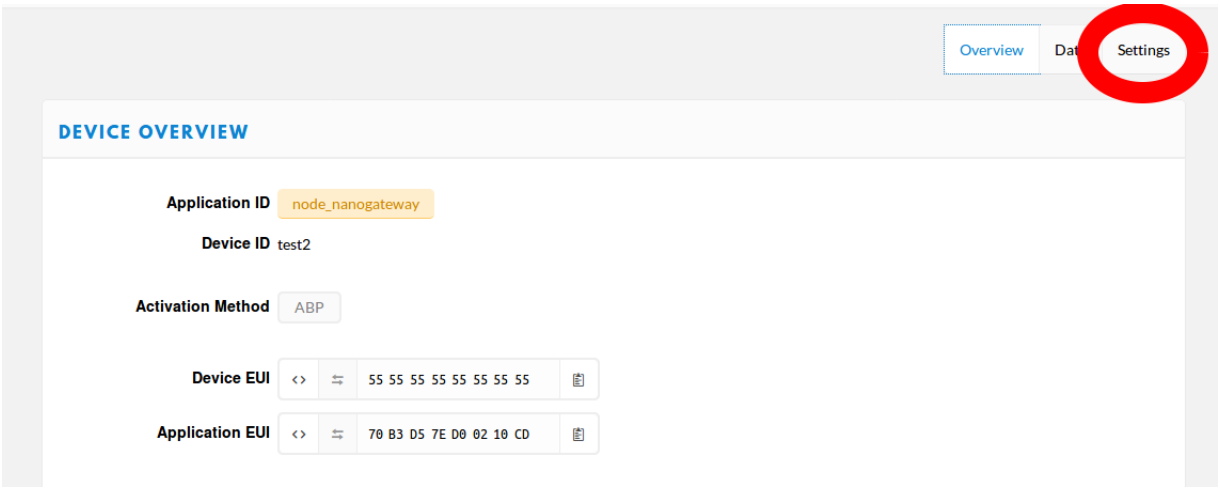



Puis *register device*
Choisir un nom pour le device ID et 16 caractères hexadecimaux pour l'EUI.

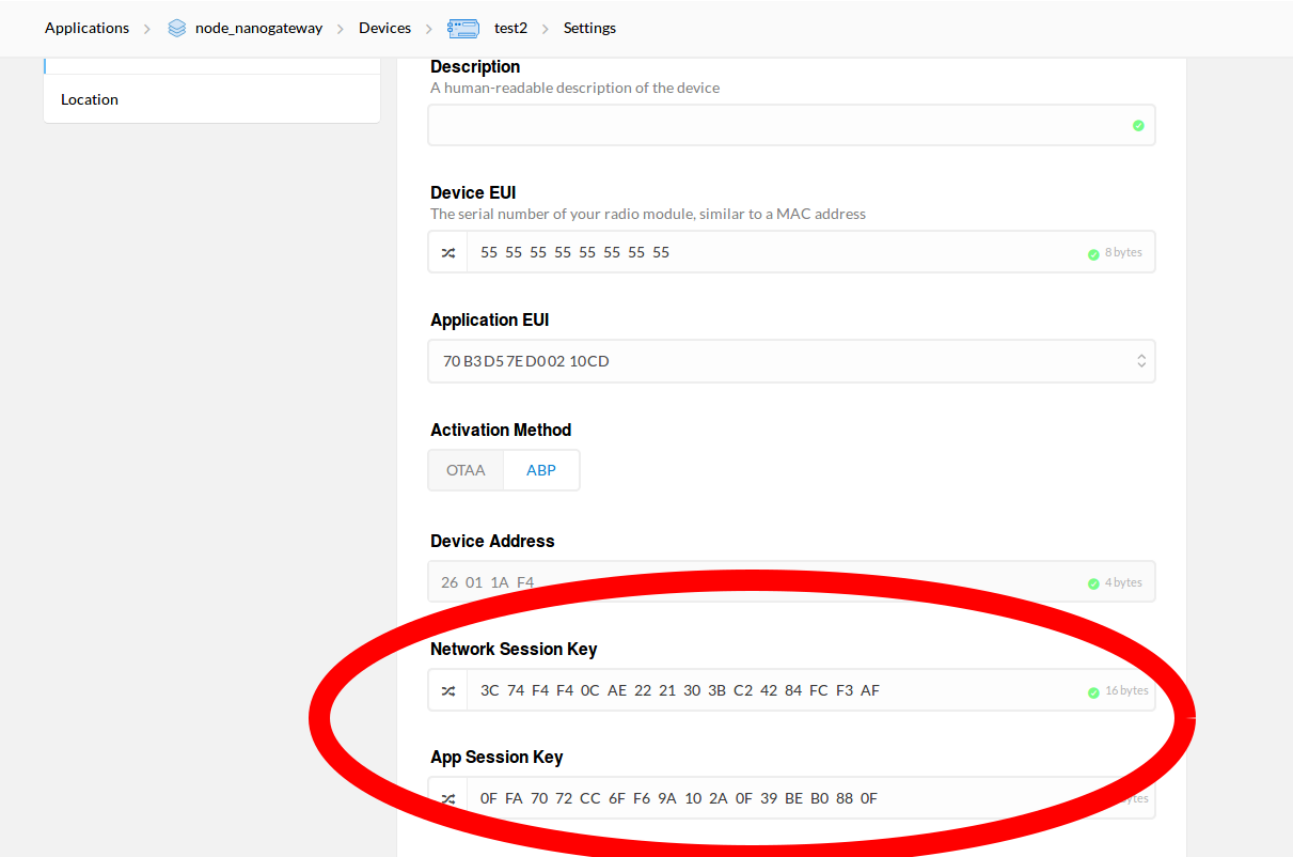



### 3.3.1 Choisir APB comme méthode d'activation

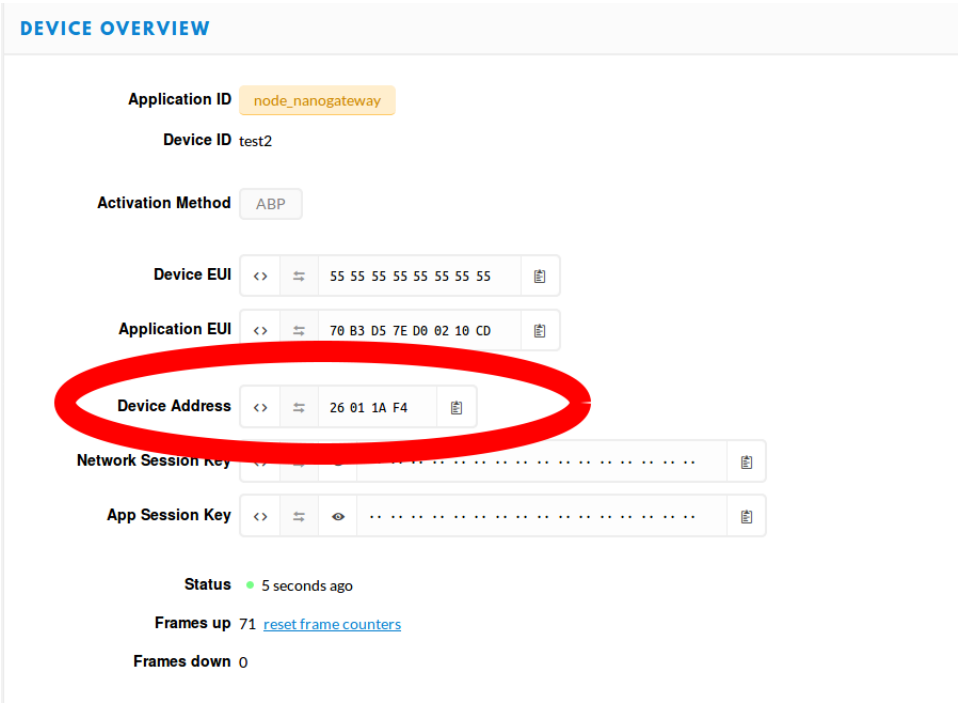
Dans *Device overview* allez dans *settings*




Puis entrez la network session key et l'application session key présents dans le code.



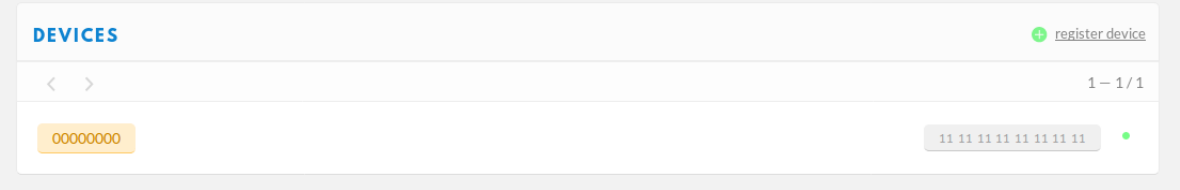

Bouton *Save* puis revenez dans l'onglet *Overview* et ajouter le device address dans le code.



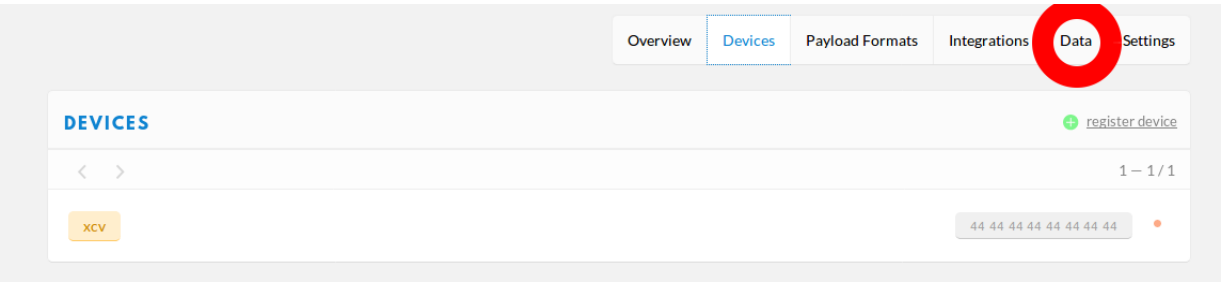

**Ajoutez Device address, network Session Key er App Session Key dans main.py**
Télécharger le code sur la carte.

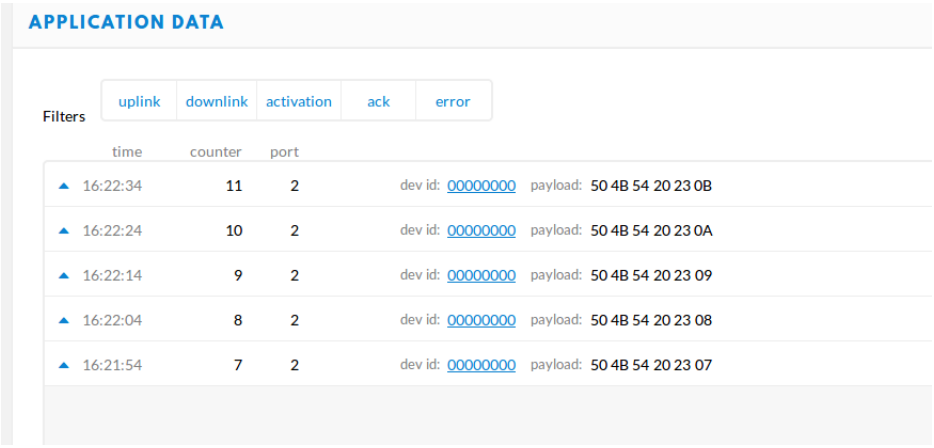
## 3.4 Lire la payload envoyée par le device

- Une fois le device connecté :




- Dans l'onglet data :


Overview
Devices
Payload Formats
Integrations
Data
Settings
DEVICES
register device
1 – 1 / 1
xcv
44 44 44 44 44 44 44 44


APPLICATION DATA
Filters
uplink
downlink
activation
ack
error
time
counter
port
16:22:34 11 2 dev id: 00000000 payload: 50 4B 54 20 23 0B
16:22:24 10 2 dev id: 00000000 payload: 50 4B 54 20 23 0A
16:22:14 9 2 dev id: 00000000 payload: 50 4B 54 20 23 09
16:22:04 8 2 dev id: 00000000 payload: 50 4B 54 20 23 08
16:21:54 7 2 dev id: 00000000 payload: 50 4B 54 20 23 07