

LoRa Mac avec Lopy4 ping pong

cmonaton

July 2019

1 Introduction

But : Envoyer des messages directement entre 2 Lopy4 en LoRa MAC c.à.d sans passer par le réseau LoRaWAN.



Figure 1: pycom lopy 4

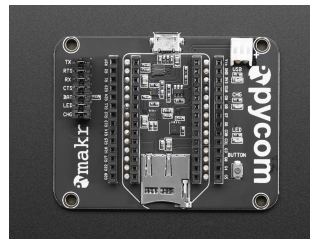


Figure 2: expansion board v3.0



Figure 3: antenne LoRa

2 Matériel

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

3 Code à télécharger sur les cartes

Téléchargez le code à : https://github.com/GitClementtest/pingpong_lobby4

3.1 Node A

```
from network import LoRa
import socket
import time

lora = LoRa(mode=LoRa.LORA, region=LoRa.EU868)
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
s.setblocking(False)

while True:
    if s.recv(64) == b'Ping':
        print('Pong sending')
        s.send('Pong')
        time.sleep(2)
```

3.2 Node B

```
from network import LoRa
import socket
import time

lora = LoRa(mode=LoRa.LORA, region=LoRa.EU868)
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
s.setblocking(False)
while True:
    data=s.recv(64)
    print(data)
    s.send('Ping')
    time.sleep(2)
```

4 Se connecter à la carte pour la programmer

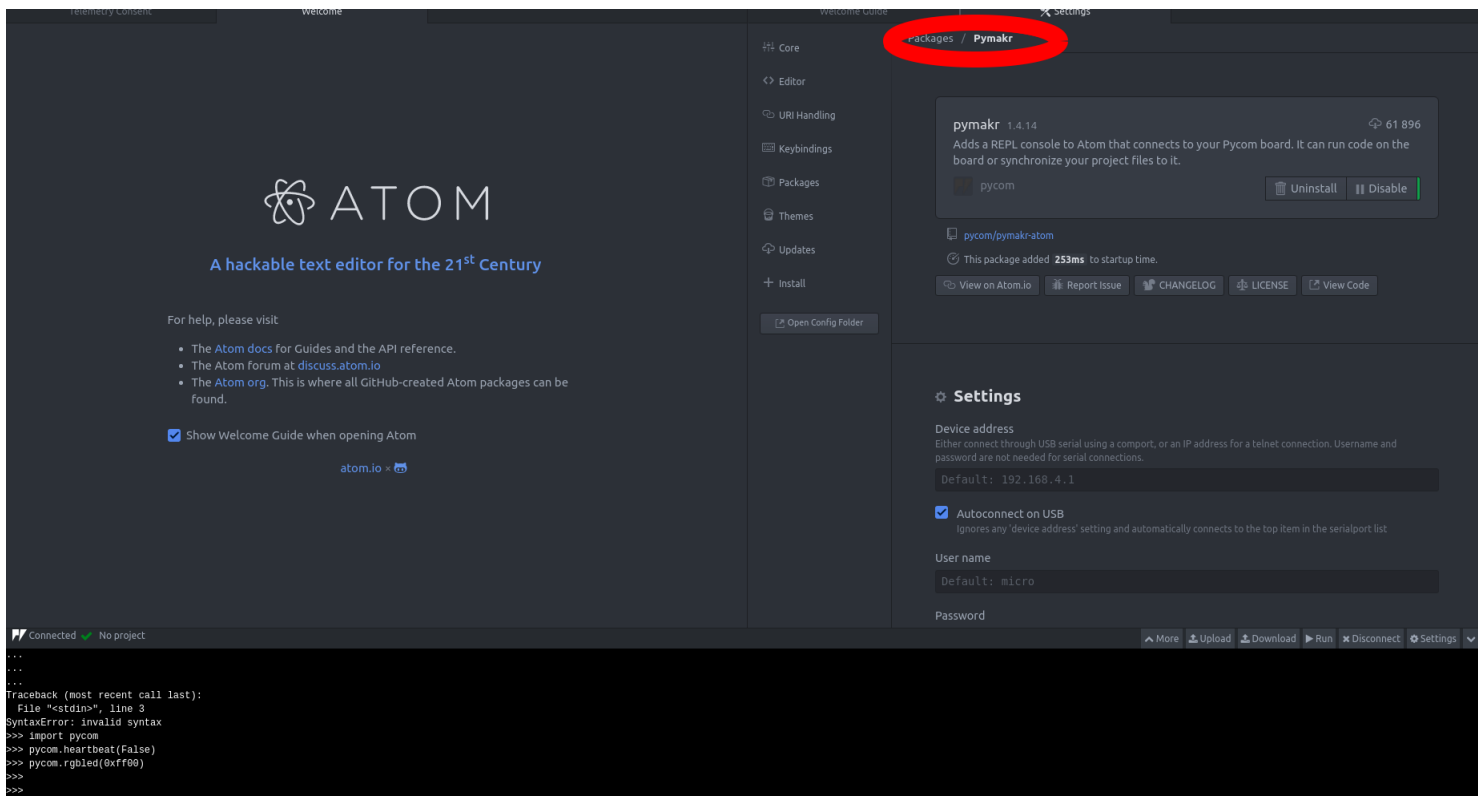
4.0.1 installer Atom text editor

Télécharger la version 1.39.0 sur <https://github.com/atom/atom/releases>
Téléchargez le .deb.

4.0.2 installer pymakr

Informations complémentaires : <https://docs.pycom.io/pymakr/installation/atom/>

Depuis atom selon l'image installer pymakr



Si il est impossible d'installer pymakr, essayez d'installer une version antérieure à : <https://github.com/atom/atom/releases>, Une alternative est aussi Visual Studio Code.

4.0.3 Connexion à la carte

Déterminer le port série sur lequel la carte est montée :
Après le branchement :

```
dmesg | grep tty
```

ttyACM0 dans mon cas. Appuyez sur reset si le programme ne se lance pas.

4.0.4 Pour ouvrir les ports ttyACM0 et ttyACM1

Solution temporaire

```
sudo chmod 666 /dev/ttyACM0
```

Il faut le entrez cette commande souvent.

Solution permanente

Créer un fichier dans son home

```
50-myusb.rules
```

l'éditer :

```
KERNEL=="ttyACM[0-9]*",MODE="0666"
```

Puis copiez ce fichier dans /etc/udev/rules.d/ et redémarrer votre PC.

```
sudo cp 50-myusb.rules /etc/udev/rules.d
```

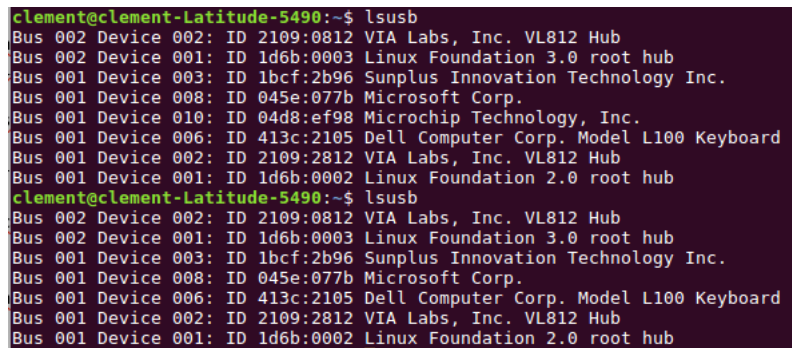
C'est suffisant pour ne plus avoir à réouvrir les ports manuellement. Cependant, n'importe quel dispositif usb connecté au PC a maintenant le droit d'écriture sur le PC.

Pour plus de sécurité ajouter ces lignes dans ce fichier :

```
ACTION=="add", KERNEL=="ttyACM[0-9]*", ATTRS{idVendor}=="xxxx",  
ATTRS{idProduct}=="yyyy", MODE="0666"
```

Pour déterminer idVendor et idProduct des cartes tapez lsusb avant et après avoir connecter la carte.

Dans mon cas avant et après avoir branché une carte lopy4 :

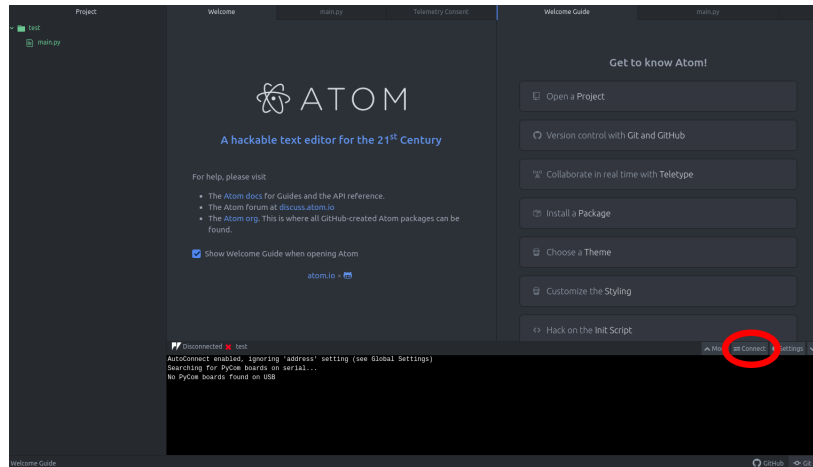


```
clement@clement-Latitude-5490:~$ lsusb  
Bus 002 Device 002: ID 2109:0812 VIA Labs, Inc. VL812 Hub  
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 001 Device 003: ID 1bcf:2b96 Sunplus Innovation Technology Inc.  
Bus 001 Device 008: ID 045e:077b Microsoft Corp.  
Bus 001 Device 010: ID 04d8:ef98 Microchip Technology, Inc.  
Bus 001 Device 006: ID 413c:2105 Dell Computer Corp. Model L100 Keyboard  
Bus 001 Device 002: ID 2109:2812 VIA Labs, Inc. VL812 Hub  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
clement@clement-Latitude-5490:~$ lsusb  
Bus 002 Device 002: ID 2109:0812 VIA Labs, Inc. VL812 Hub  
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 001 Device 003: ID 1bcf:2b96 Sunplus Innovation Technology Inc.  
Bus 001 Device 008: ID 045e:077b Microsoft Corp.  
Bus 001 Device 006: ID 413c:2105 Dell Computer Corp. Model L100 Keyboard  
Bus 001 Device 002: ID 2109:2812 VIA Labs, Inc. VL812 Hub  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

```
idProduct = ef98  
idVendor= 04d8
```

Pour ajouter d'autres appareils, copier coller ces lignes en changeant idProduct et idVendor.

Selon l'image utiliser la console d'Atom pour se connecter à la carte :



4.1 Télécharger un firmware sur la carte avec Atom Pymakr

Guide sur le site d'Atom : <https://docs.pycom.io/gettingstarted/programming/first-project/>

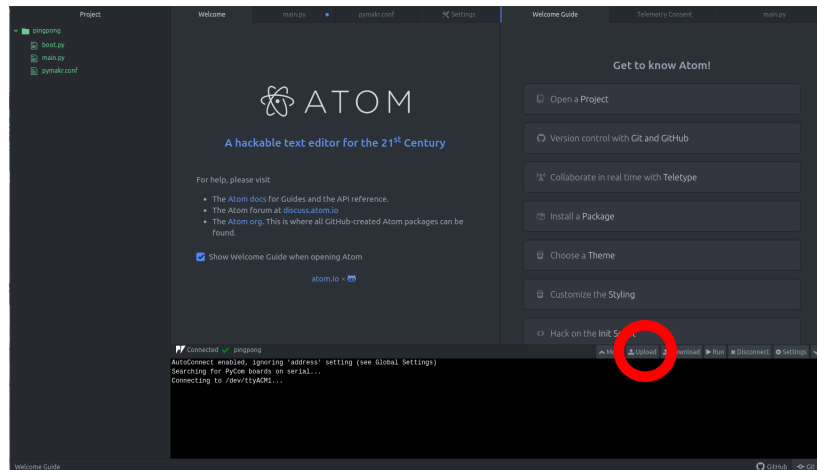
Créer 2 dossiers

Créer 1 main.py contenant le code décrit en première partie dans chaque dossier.

4.1.1 Connecter la carte avec Atom

Appuyer sur le bouton connect de la console pymakr. S'assurer que le port est ouvert est qu'aucun autre terminal série n'est ouvert sur ce port.

Dans Atom File, Open Folder pour choisir son projet puis cliquer sur Upload dans la console Pymakr pour télécharger le code sur la carte.



Si vous copiez-collez le code attention aux identations pour le firmware. Parfois avec des copier-coller les indentations se perdent. Il faut coller toutes les lignes à gauche du fichier (colonne 0), sauf lorsqu'on entre dans des conditions type while, if etc. Dans ces conditions il faut 1 tabulation au début de chaque ligne. Si ce n'est pas respecté le programme ne compile pas.

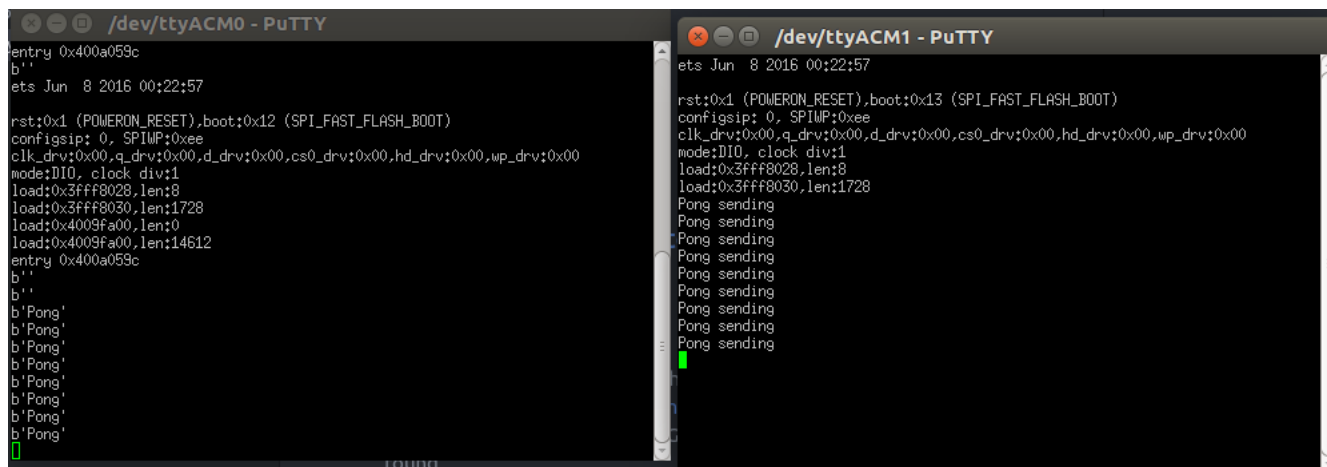
Après avoir téléchargé le firmware sur la carte, appuyer sur le bouton reset de la carte à côté des leds rgb pour lancer le programme.



4.2 Avec un terminal série type PuTTY

PuTTY : sélectionner liaison série, choisir le bon port `/dev/ttyACM0` dans mon cas, baudrate 115200. Les cartes sont branchées sur `ttyACM0` et `ttyACM1` dans mon cas.

5 Résultat



5.1 Avec minicom

Je recommande d'utiliser minicom car on a pas besoin de réouvrir le programme à chaque fois qu'on déconnecte un appareil.

5.1.1 Installer minicom

```
sudo apt-get install minicom
```

5.1.2 Pour quitter minicom

ctrl+a puis q

5.1.3 lancer minicom

```
sudo minicom
```

5.1.4 Configurer minicom

```
sudo minicom -s
```

6 Résultat

[illegible]