

riot idosens

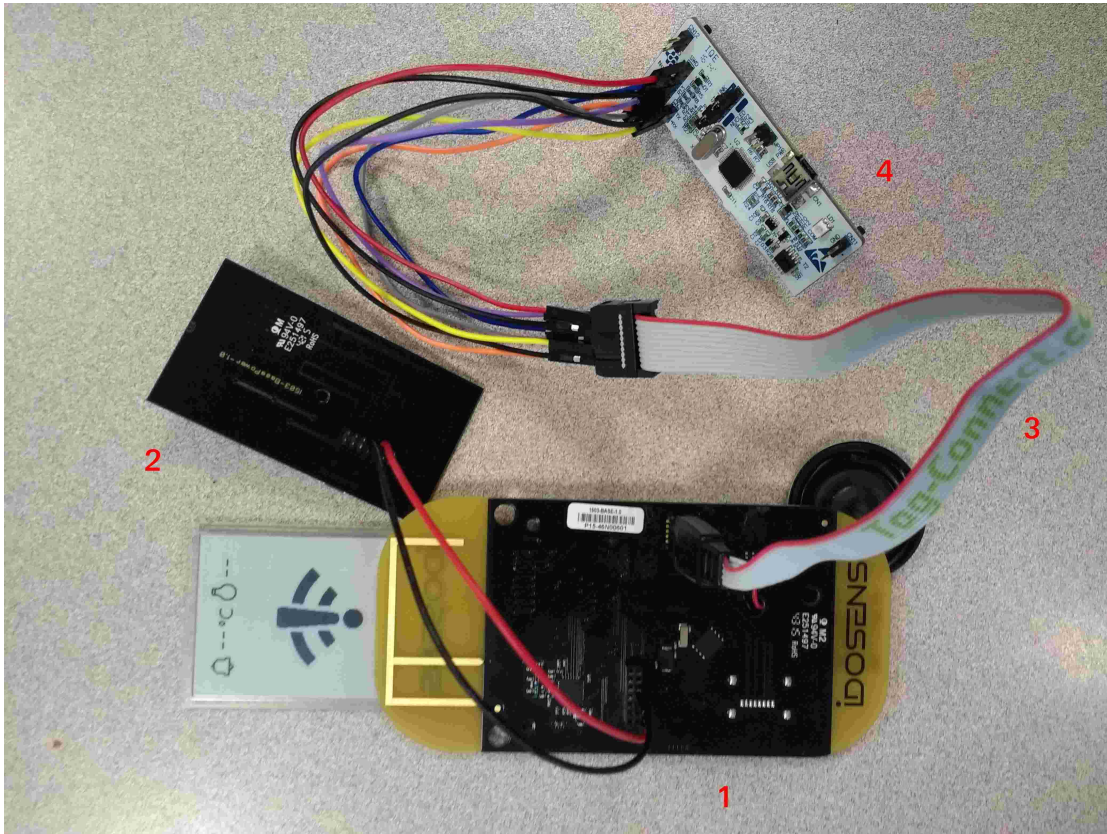
cmonaton

October 2019

## 1 Introduction

Installer l'os riot sur le détecteur d'ouverture de porte idosens. Cela permet entre autres d'utiliser la fonction radio du produit. Ensuite utiliser un exemple pour envoyer une trame LoRa d'une carte à une autre.

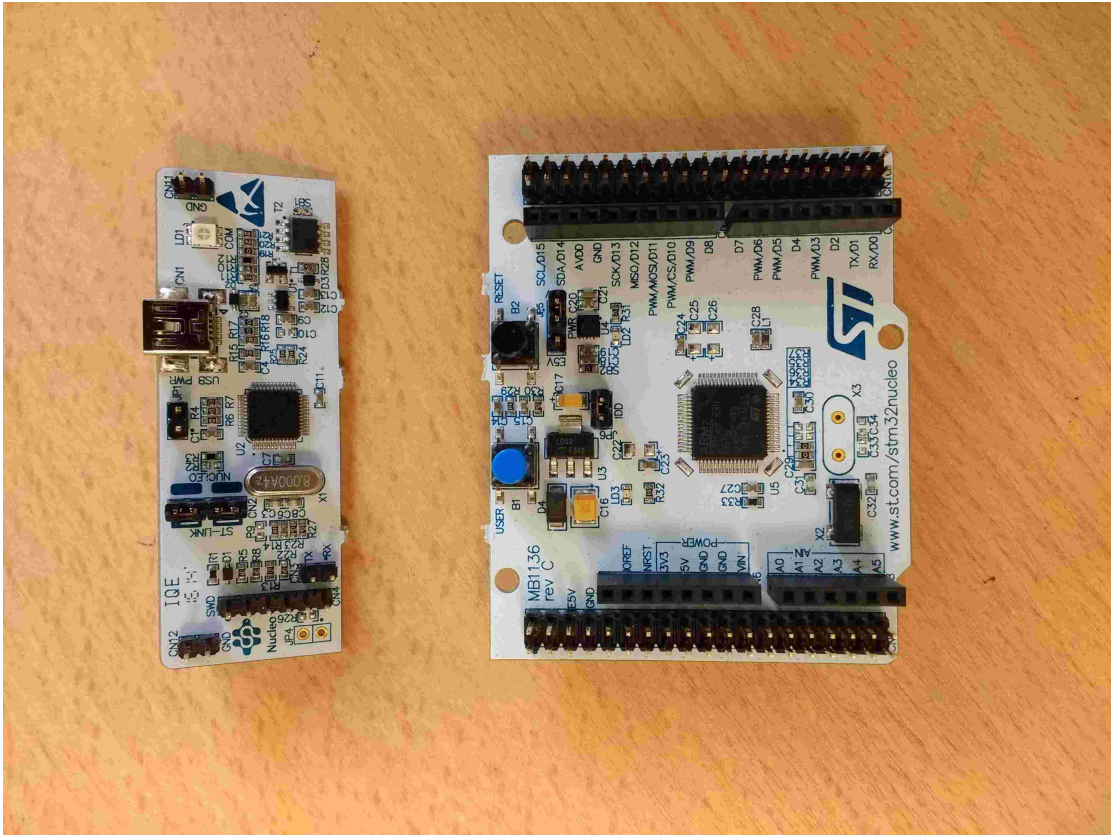
## 2 Matériel



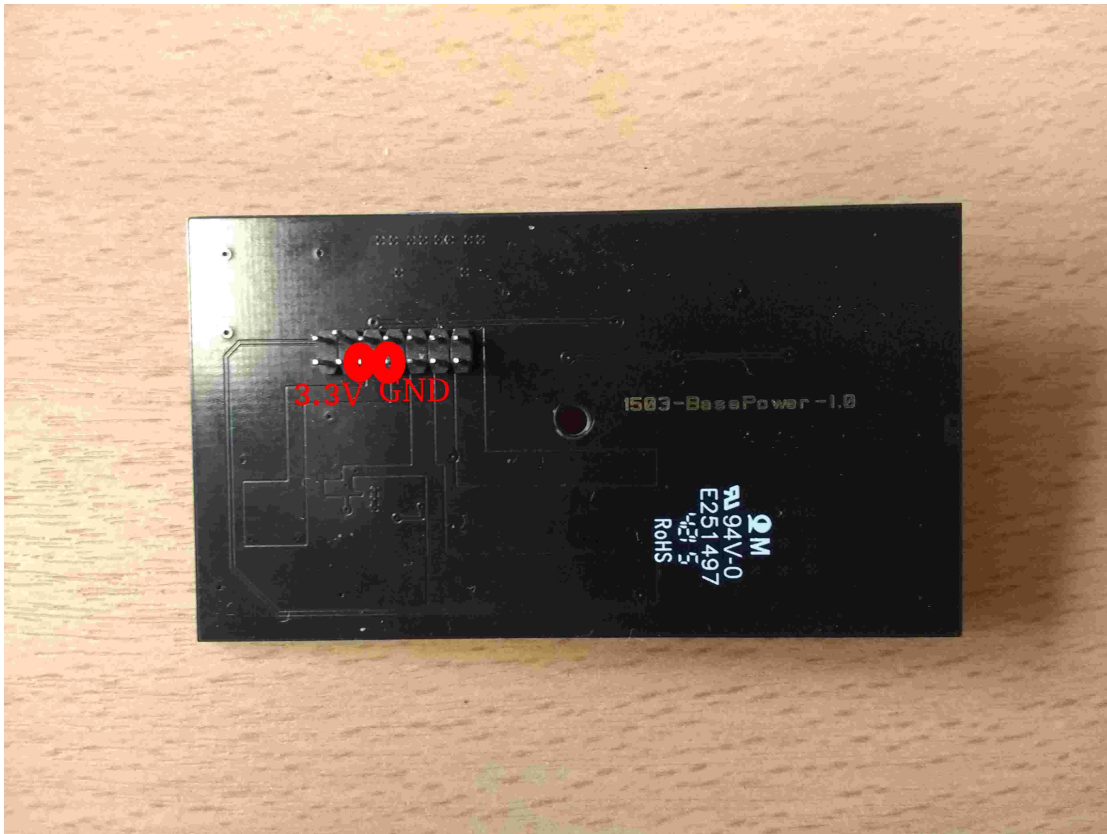
- 1 Base
- 2 base power
- 3 Tag-connect
- 4 ST-LINK

### 2.1 ST-LINK

Ce débogueur/flasheur ST-LINK a été prélevé sur une carte nucleo :



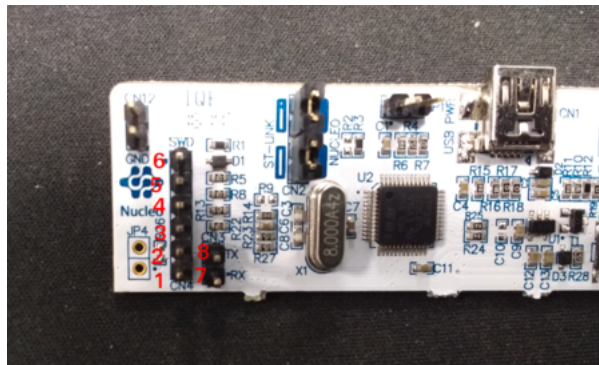
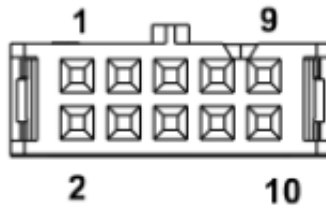
## 2.2 Connexion base power





### 2.3 Connexion ST-LINK-Tag-connect

Connectez le Tag-connect au nucleo selon ce tableau et selon les schémas des connecteurs :



Pin Nucleo	Pin Tag-Connect
1 Vcc	6 VDD TARGET
2 SWDIO	3 SWDIO
3 Gnd	4 GND
4 SWCLK	5 SWCLK
X	X
6 SWO	1 SWO
7 Tx	7 Rx
X	X
9 Rx	8 Tx
10 NRST	2 NRST

### 3 installation logiciels et configuration du PC

L'exemple est fait avec ubuntu 16.04

#### 3.1 Déverrouiller ports USB

ajoutez un fichier 50-myusb.rules dans



contenant  
avec idProduct et idVendor à renseigner en fonction du votre produit. On  
les trouve avec le programme  
Redémarrez le PC.

### 3.2 Installer pyserial pour communiquer avec la carte

### 3.3 Installer openocd pour flasher la carte

## 4 Compiler et utiliser riot

### 4.1 Installer le compilateur gcc-arm-embedded pour générer du code pour la carte

Téléchargez le code à : <https://github.com/CampusIoT/RIOT>

Dans un terminal, allez dans

Pour lancer la compilation et flasher la carte :

## 5 Ping-Pong idosens

Avec l'exemple flashé ci-dessus, on peut faire communiquer 2 idosens. Pour cela, le mieux est de connecter un idosens par PC. Une fois l'os flashé, entrez les commandes suivantes :

- idosens émetteur

```
setup 125 12 5
2019-10-22 11:42:41,897 # setup 125 12 5
2019-10-22 11:42:41,899 # setup: setting 125KHz bandwidth
2019-10-22 11:42:41,904 # [Info] setup: configuration set with success
> channel set 868000000
2019-10-22 11:43:02,937 # channel set 868000000
2019-10-22 11:43:02,939 # New channel set
> send 123
2019-10-22 11:43:16,496 # send 123
2019-10-22 11:43:16,499 # sending "123" payload (4 bytes)
> 2019-10-22 11:43:17,330 # Transmission completed
```

- idosens récepteur

```
2019-10-22 11:41:07,495 # setup 125 12 5
2019-10-22 11:41:07,500 # setup: setting 125KHz bandwidth
2019-10-22 11:41:07,508 # [Info] setup: configuration set with success
> channel set 868000000
2019-10-22 11:41:49,630 # channel set 868000000
2019-10-22 11:41:49,635 # New channel set
> listen
2019-10-22 11:41:58,820 # listen
2019-10-22 11:41:58,823 # Listen mode set
> 2019-10-22 11:43:17,216 # Data reception started
2019-10-22 11:43:17,387 # {Payload: "123" (4 bytes), RSSI: 128, SNR: -6, TOA: 99
2}
```