

Test-Range Application

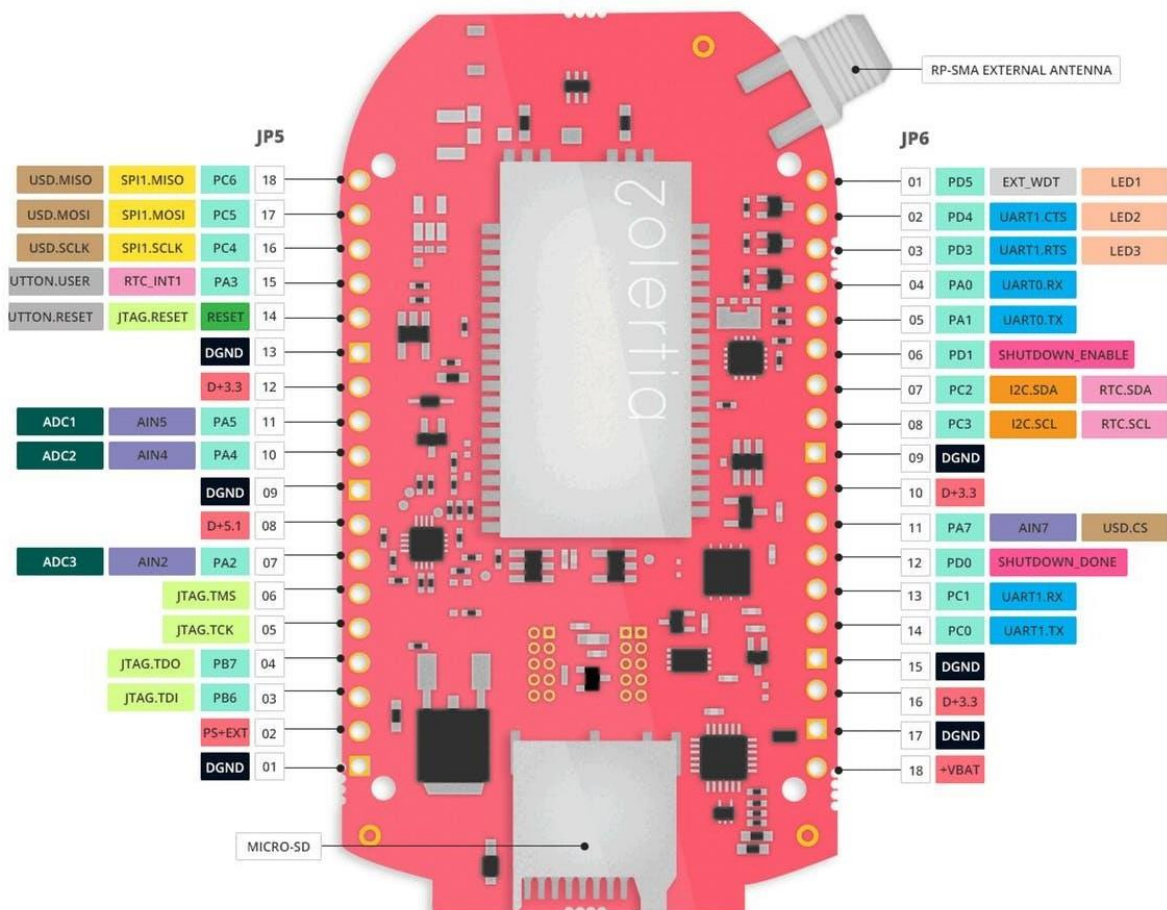
Objectif :

Essayer d'estimer la portée maximale entre deux appareils et de prévenir la perte de liaison due à la pluie, aux obstacles, à la décoloration, etc.

Circuit Physique

Afin d'appliquer ce test, vous devez avoir dans votre kit un émetteur et un récepteur ainsi qu'un ordinateur portable où stocker et sauvegarder les résultats. C'est là où l'importance d'un module Zolertia se manifeste. Vous pouvez alors réduire votre kit pour avoir à la fin deux modules Zolertia (un émetteur et un récepteur), un Grove LCD RGB Backlight et une batterie.

Voici ci-dessous une figure qui illustre le schéma d'un Zolertia Re-Mote :



La plateforme Re-Mote fonctionne sur les bandes 2,4GHz ou 863-950MHz.

Le LCD fonctionne sur I2C et est alimenté sur 5V (mais heureusement l'I2C a une logique de 3,3V).



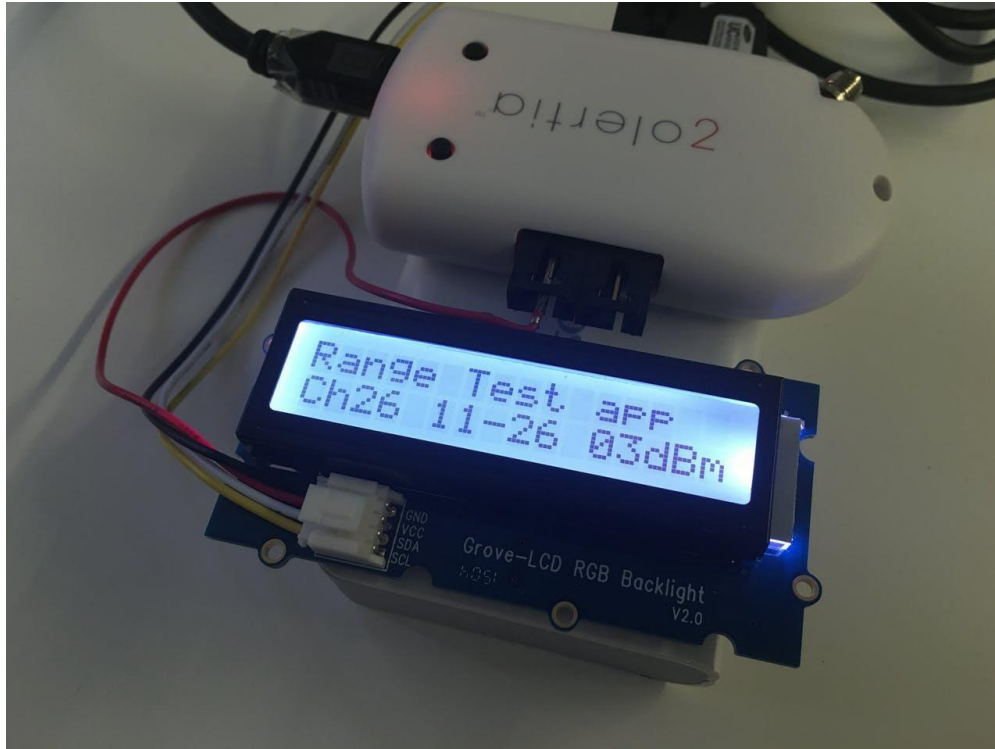
Comme le RE-Mote fonctionne avec une logique de 3,3V mais peut être alimenté par USB à 5V, j'ai alimenté l'écran LCD du RE-Mote sur la broche D+5,1 (voir ci-dessus). Les broches I2C de l'écran LCD (SDA et SCL) sont connectées au RE-Mote par les broches I2C.SDA et I2C.SCL comme prévu.



Le chargeur de batterie est une batterie USB d'une capacité de 6000mAh, ce qui donne beaucoup de jus pour les derniers jours de test. Le seul inconvénient est de devoir soit

appuyer en permanence sur le bouton de mise sous tension, soit désactiver le fonctionnement à faible puissance du RE-Mote, sinon la batterie "pense" qu'il n'y a pas d'appareil connecté car la consommation électrique est trop faible.

Résultat attendu :



Au démarrage de l'application, elle imprimera des informations sur la configuration de la radio. Le canal actuel est affiché sur la ligne inférieure à gauche, suivi des canaux disponibles et de la puissance d'émission (en dBm).

Code

Commencez par vous connecter sur votre machine virtuelle à base Ubuntu.

Vous devrez avoir déjà installé Contiki sur la machine. Sinon, rendez-vous sur le pdf « Contiki Implementation ».

Pour obtenir la version récente du code de l'application du test-range, tapez dans votre terminal la commande suivante :

```
git clone https://github.com/alignan/contiki
cd contiki && git checkout field-test-lcd
```

Entrez ensuite dans la directory `Contiki/examples/zolertia/zoul/range-test`

Si vous regardez le Makefile, il y a deux compilations à prendre en compte :

```

ifdef WITH_LCD
CFLAGS += -DLCD_ENABLED=$(WITH_LCD)
endif

ifdef WITH_SUBGHZ
CFLAGS += -DRF_SUBGHZ=$(WITH_LCD)
endif

```

Lors de la compilation et de la programmation du RE-Mote, vous pouvez choisir d'utiliser soit l'interface radio 2,4GHz (CC2538), soit la radio 868/915MHz (CC1200) (si vous ajoutez WITH_SUBGHZ=1). Si vous ajoutez l'argument WITH_LCD=1 à la compilation, elle inclura le support de l'écran LCD, sinon si vous n'avez pas d'écran LCD, vous pouvez toujours voir les résultats via USB si vous vous connectez à la RE-Mote, en utilisant simplement un câble micro-USB et du mastic ou similaire.

!! N.B : Vous ne pourrez pas visualiser l'affichage sur le LCD que lorsque vous tapez WITH_LCD=1 lors du Make. Sinon, vous obtiendrez un écran noir.

```

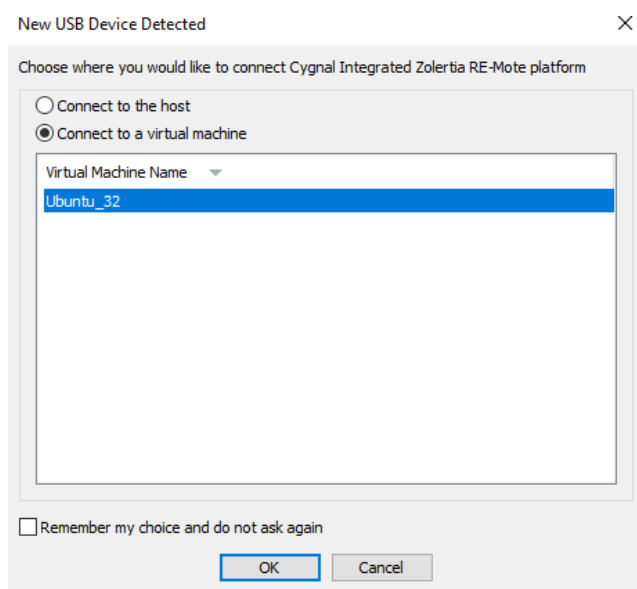
# Configure for 868MHz band
make range-test.upload WITH_SUBGHZ=1 WITH_LCD=1

or

# Configure for 2.4GHz band
make range-test.upload WITH_LCD=1

```

Connectez votre module Zolertia à votre pc ou à votre Raspberry Pi à l'aide d'un câble USB. N'oubliez pas de permettre l'utilisation de Zolertia dans votre machine.



Une fois dans le repertoire range-test, tapez :

```
$ sudo make TARGET=zoul range-test
```

Un affichage comme celui-là devrait apparaitre :

```
farah@farah-virtual-machine: ~/contiki/contiki/examples/zolertia/zoul/range-test
CC      ../../../../../../core/net/mac/framer-nullmac.c
CC      ../../../../../../core/net/mac/mac.c
CC      ../../../../../../core/net/mac/mac-sequence.c
CC      ../../../../../../core/net/mac/nordc.c
CC      ../../../../../../core/net/mac/nullmac.c
CC      ../../../../../../core/net/mac/nullrdc.c
CC      ../../../../../../core/net/mac/nullrdc-noframer.c
CC      ../../../../../../core/net/mac/phase.c
CC      ../../../../../../core/net/mac/contikimac/contikimac.c
CC      ../../../../../../core/net/mac/contikimac/contikimac-framer.c
CC      ../../../../../../core/net/llsec/anti-replay.c
CC      ../../../../../../core/net/llsec/ccm-star-packetbuf.c
CC      ../../../../../../core/net/llsec/nullsec.c
CC      ../../../../../../core/net/llsec/noncoresec/noncoresec.c
CC      ../../../../../../dev/cc1200/cc1200-802154g-863-870-fsk-50kbps.c
CC      ../../../../../../dev/cc1200/cc1200.c
CC      range-test.c
LD      range-test.elf
arm-none-eabi-objcopy -O ihex range-test.elf range-test.hex
arm-none-eabi-objcopy -O binary --gap-fill 0xff range-test.elf range-test.bin
cp range-test.elf range-test.zoul
rm obj_zoul/startup-gcc.o range-test.co
```

Vous sauvegardez la plateforme cible :

```
farah@farah-virtual-machine:~/contiki/contiki/examples/zolertia/zoul/range-test$
sudo make TARGET=zoul savetarget
saving Makefile.target
```

Puis vous flashez le code :

```
farah@farah-virtual-machine:~/contiki/contiki/examples/zolertia/zoul/range-test$
sudo make range-test.upload
```

Si vous obtenez une telle erreur :

```
ERROR: Could not find the cc2538-bsl script.
```

 Il suffit d'actualiser votre code,

donc :

```
$ sudo git submodule update
```

```
$ sudo git submodule
```

init Doit faire l'affaire.

Vous flashez ce code sur les 2 Zolertia de votre circuit. Vous devriez voir l'affichage ci-dessous :

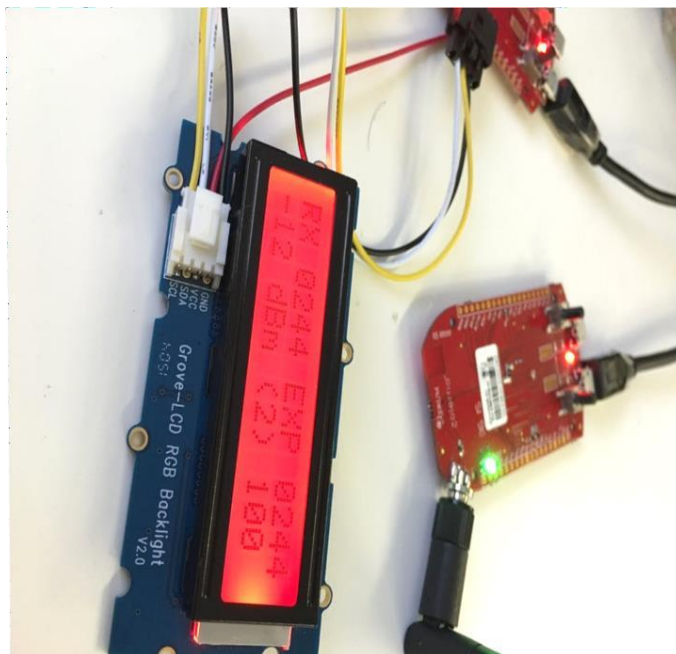


Il faut programmer un des Zolertia comme émetteur TX, et l'autre récepteur RX.

En principe, en appuyant longtemps sur le bouton de l'utilisateur (sans le relâcher), il bascule entre le mode émetteur et le mode récepteur. L'écran LCD sera rouge vif lorsqu'il est configuré comme récepteur, ou vert lorsque le mode émetteur est sélectionné.

Lorsque le bouton utilisateur est relâché, le mode de fonctionnement est activé (attention : si vous devez changer le mode de fonctionnement, appuyez sur le bouton de réinitialisation et répétez).

Une seule pression sur le bouton utilisateur lancera le test. Ci-dessous un Zolertia programmé en tant que récepteur.



Une fois le Zolertia allumé, la valeur du canal est par défaut 0 pour 868MHz et 26 pour 2.4GHZ. Ces deux canaux sont les meilleurs à utiliser.

Afin de changer la valeur du TX power, ou la puissance de transmission, 2 cas se posent :

- En 868MHz :

Le RE-Mote possède une interface embarquée de moins de 1 GHz basée sur l'émetteur-récepteur radio CC1120, configuré pour fonctionner dans les bandes 863-950 MHz. La puissance de transmission maximale autorisée dépend de la bande spécifique et des réglementations en vigueur, qui peuvent imposer également des limites au gain maximal de l'antenne, assurez-vous de vérifier les réglementations locales avant de modifier la puissance de sortie.

Table 4.1. CC2538 Transmission power recommended values (from SmartRF Studio⁶)

| TX Power (dBm) | Value |
|----------------|-------------|
| +7 | 0xFF |
| +5 | 0xED |
| +3 | 0xD5 |
| +1 | 0xC5 |
| 0 | 0xB6 |
| -1 | 0xB0 |
| -3 | 0xA1 |
| -5 | 0x91 |
| -7 | 0x88 |
| -9 | 0x72 |
| -11 | 0x62 |
| -13 | 0x58 |
| -15 | 0x42 |
| -24 | 0x00 |

Dans le fichier range-test-transmit.c

```
rd = NETSTACK_RADIO.set_value(RADIO_PARAM_CHANNEL, value);
```

- En 2.4GHz :

La plateforme RE-Mote utilise la radio CC2538 intégrée de 2,4 GHz. Par défaut, la puissance de transmission est réglée à 3 dBm (2 mW) dans l'en-tête `cpu/cc2538/dev/cc2538-rf.h`.

Modifiez le fichier `cpu/cc2538/dev/cc2538-rf.h`

CC2538_RF_TX_POWER_RECOMMENDED value

Table 4.3. CC1200 Transmission power recommended values (from SmartRF Studio¹⁰)

| TX Power (dBm) | Value |
|----------------|-------------|
| +14 | 0x7F |
| +13 | 0x7C |
| +12 | 0x7A |
| +11 | 0x78 |
| +8 | 0x71 |
| +6 | 0x6C |
| +4 | 0x68 |
| +3 | 0x66 |
| +2 | 0x63 |
| +1 | 0x61 |
| 0 | 0x5F |
| -3 | 0x58 |
| -40 | 0x41 |