





















## Nomenclature des éléments du système

- 1 Raspberry Pi B+ « RPi »
- 1 carte SD 8 Go avec l'OS Raspbian installé
- 2 câbles convertisseurs UART/USB
- 2 modules de fingerprint GT511c5
- 2 modules à LED « semaphore »
- 1 écran LCD/I2C 16x2 caractères
- 1 module à 2 relais 5V SPST
- 1 module à optocoupleur « intercom detector »

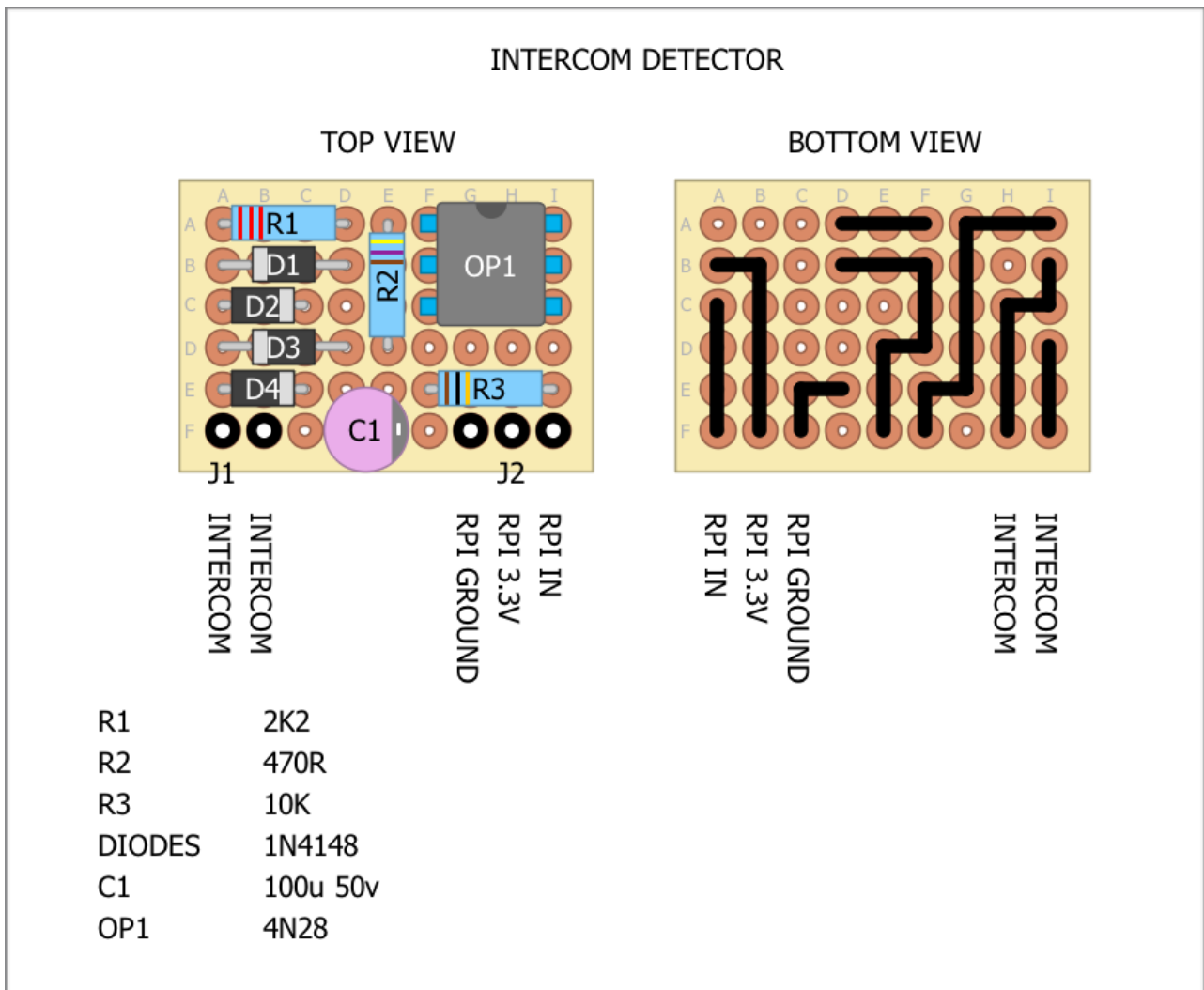
Brochage RPi :

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I2C)		DC Power 5v	04
05	GPIO03 (SCL1 , I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev. 1.1  
16/07/2014

<http://www.element14.com>

## Installation du détecteur Intercom



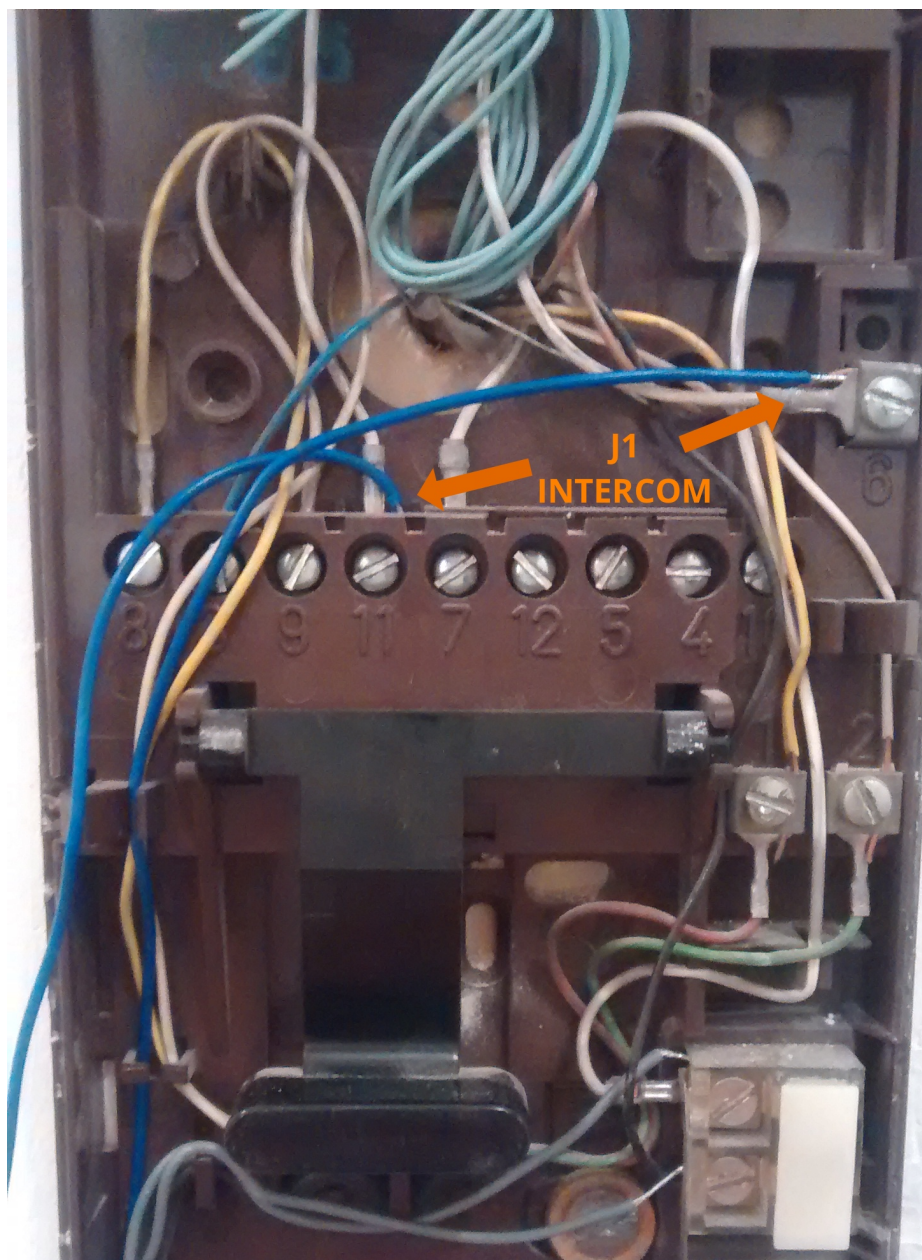
Brancher 3 fils sur le port J2 :

J2-1 ⇄ RPi Ground (n'importe laquelle)

J2-2 ⇄ RPi 3.3v (#1 ou #17)

J2-1 ⇄ RPi GPIO21 (#40)

Ouvrir l'interphone, (**AVEC DES GANTS, VOLTAGES ELEVES**) y placer le module à optocoupleur.  
Brancher 2 fils sur le port J1, les relier aux bornes de l'interphone comme ci-dessous (fils bleus) :

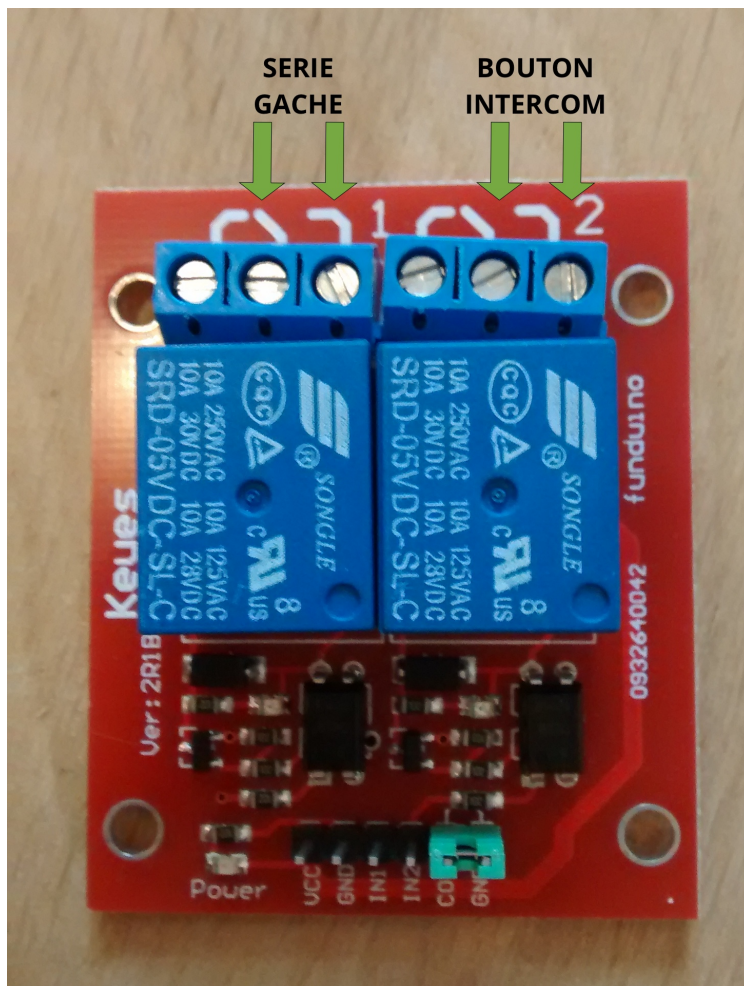


Brancher 2 fils aux bornes du bouton (ici en bas, en gris), ils seront connectés à la carte relais.

### Installation des relais de puissance :

#### Attention, tensions et ampérages dangereux !!!

Il faut connecter uniquement les sorties "OFF" par défaut des relais. L'autre sortie doit être ignorée :



Sur cette carte, connecter :

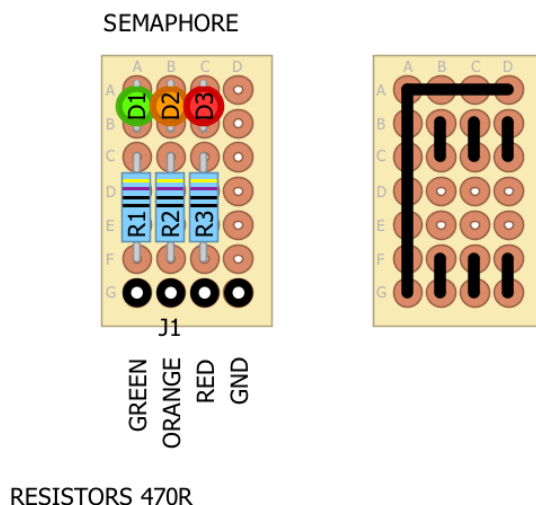
- VCC ⇔ RPi DC Power 5v (#02 ou #04)
- GND ⇔ RPi Ground (n'importe laquelle)
- IN1 ⇔ RPi GPIO3 (#15)
- IN2 ⇔ RPi GPIO23 (#16)

La sortie du 1er relais sert de coupure pour la gâche de la porte d'entrée : la connecter en série avec le circuit (comme un interrupteur).

La sortie du 2e relais doit être reliée directement par 2 fils aux contacts du bouton d'ouverture de l'interphone (fils gris sur la photo de l'interphone).

Refermer l'interphone.

## Installation des semaphores à LED :



### Module entrée :

- J1-1 (green) ⇔ RPi GPIO7 (#7)
- J1-2 (yellow) ⇔ RPi GPIO2 (#13)
- J1-3 (red) ⇔ RPi GPIO0 (#11)
- J1-4 (GND) ⇔ RPi Ground (n'importe laquelle)

### Module sortie :

- J1-1 (green) ⇔ RPi GPIO4 (#16)
- J1-2 (yellow) ⇔ RPi GPIO6 (#22)
- J1-3 (red) ⇔ RPi GPIO5 (#18)
- J1-4 (GND) ⇔ RPi Ground (n'importe laquelle)

### Installation de l'écran LCD de bienvenue (optionnel) :

- LCD VCC ⇔ RPi DC Power 5v (#02 ou #04)
- LCD GND ⇔ RPi Ground (n'importe laquelle)
- LCD SDA ⇔ RPi SDA1 (#03)
- LCD SCL ⇔ RPi SCL1 (#05)

### Installation des modules de fingerprint :

- GT511c5 fil rouge ⇔ UART/USB fil rouge (VCC)
- GT511c5 fil noir ⇔ UART/USB fil noir (GND)
- GT511c5 fil blanc ⇔ UART/USB fil blanc (TX/RX)
- GT511c5 fil vert ⇔ UART/USB fil vert (RX/TX)

Ne pas brancher les convertisseurs dans les ports USB tant que le RPi n'est pas démarré !

## Concept du logiciel

Le logiciel digilock est une sorte d'interpréteur de commandes, qui lance plusieurs threads en parallèle :

- un thread qui écoute les commandes entrées par l'utilisateur,
- un thread qui écoute les impulsions électriques envoyées par le détecteur Intercom et ouvre les portes d'entrées selon certaines conditions,
- deux threads qui attendent qu'un doigt soit pressé sur les capteurs d'empreintes, et accorde ou refuse l'ouverture de la gâche électrique.

Les données d'empreintes digitales (« fingerprint ») sont stockées sous forme de templates de 498 bytes (on ne peut pas remonter à une image d'empreinte depuis un template) sur les capteurs d'empreintes eux-mêmes. Ces données sont aussi copiées dans une base de données SQLite sur la carte SD du RPi, et également envoyées à un serveur distant (lorsqu'un nouveau template est créé) par HTTP.

Tous les évènements (accès accordé, refus, bouton pressé, etc.) sont stockées sur la base SQLite locale et envoyés au serveur distant par HTTP.

## Compilation et configuration du logiciel

Une fois l'installation hardware complète, démarrer le Raspberry Pi.  
login/pass : pi/pi

En ligne de commande, installer les librairies suivantes :

```
sudo apt-get install libsqlite3-dev  
sudo apt-get install libcurl4-openssl-dev  
sudo apt-get install wiringPi-dev
```

Récupérer les sources depuis le Github, compiler avec `./make`

Débrancher les ports USB vers les modules de fingerprint, puis brancher d'abord le module d'entrée, vérifier et noter son nom avec `"ls /dev/ttyU*"`.  
Faire pareil avec le module de sortie.

Ouvrir le fichier config.ini, mettre à jour les ports USB des modules d'entrée et de sortie, et changer les paramètres si besoin (le fichier est documenté). Attention, cette documentation prend en compte les ports I/O par défaut dans ce fichier.

Lancer le logiciel avec `./digilock`

### Liste des commandes acceptées :

#### **blink start / blink stop**

Démarre ou arrête le test d'envoi de commandes aux modules de fingerprint : ceux-ci doivent clignoter.



**quit**

Arrête les scans d'intercom et de fingerprint, et quitte le programme.

**enroll**

Arrête les scans de fingerprint et commence la procédure d'enrôlement d'un nouvel utilisateur.

Sont d'abord demandés un nom (16 caractères max) et un email (identifiant de l'utilisateur). Un utilisateur peut enrôler plusieurs de ses doigts, donc on peut entrer plusieurs fois le même email.

Ensuite, le personne doit centrer la pulpe de son doigt sur le module de fingerprint allumé (celui de sortie), et ce trois fois de suite afin de prendre une bonne image. Si tout est OK, le programme va alors synchroniser les bases de données des deux modules de fingerprint et relancer les scans.

**count**

Compte le nombre de doigts enrôlés dans chaque module de fingerprint. On doit obtenir le même nombre pour les 2, sinon il y a un problème de communication (vérifier les branchements) ou de synchronisation entre les modules.

**entry stop / exit stop / entry start / exit start**

Arrête ou démarre les scans de fingerprint d'entrée ou de sortie.

**intercom stop / intercom start**

Arrête ou démarre le scan de détection d'intercom.

**dump entry / dump exit**

Fait un dump de la base de données de templates de l'un des 2 modules de fingerprint (attention, ce peut être assez long).

**create user**

Equivalent à la commande **enroll**, mais sans demander d'empreintes. On pourra les rajouter plus tard avec **enroll**.

**delete all fingerprints**

Supprime les templates contenus dans la base de données des modules de fingerprint. La base de données SQLite sur le Raspberry Pi n'est pas impactée.

**delete fingerprint**

Demande un email, puis supprime tous les templates d'empreinte qui lui sont reliés. La base de données SQLite est impactée.

**delete user**

Idem que **delete fingerprint**, mais supprime aussi l'utilisateur. La base de données SQLite est impactée.

**sync fingerprints**

Efface tous les templates sur le module de fingerprint d'entrée, puis resynchronise depuis le module de sortie. La base de données SQLite n'est pas impactée.

**reload fingerprints**

Efface tous les templates sur les deux modules de fingerprint, puis resynchronise depuis la base de données SQLite.