 **Warning:** Trying to access array offset on value of type null in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/includes/option-types/typography-v2/class-fw-option-type-typography-v2.php** on line **148**

Warning: foreach() argument must be of type array|object, null given in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/includes/option-types/typography-v2/class-fw-option-type-typography-v2.php** on line **148**

Warning: Trying to access array offset on value of type null in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/helpers/general.php** on line **1275**

Warning: foreach() argument must be of type array|object, null given in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/helpers/general.php** on line **1275**



SMART BUD

Demo (<https://bud.eco-sensors.ch>) | Tutoriels (<https://eco-sensors.ch/tutoriels/>)



NOUS SOUTENIR

Shop (<https://eco-sensors.ch/shop/>) & Sponsoring (<https://eco-sensors.ch/product-category/dons/>)



SOCIAL

Facebook (<https://www.facebook.com/ecosensors>) - **Contact** (<https://eco-sensors.ch/contact/>)

Un raspberry pour lire vos cartes RFID / NFC

HOME ([HTTPS://ECO-SENSORS.CH/](https://eco-sensors.ch/))
/ SYSTÈMES EMBARQUÉS ([HTTPS://ECO-SENSORS.CH/CATEGORY/SYSTEMES-EMBARQUES/](https://eco-sensors.ch/category/systemes-embarques/))
/ UN RASPBERRY POUR LIRE VOS CARTES RFID / NFC

BY ECOSENSORS ([HTTPS://ECO-SENSORS.CH/AUTHOR/ECOSENSORS/](https://eco-sensors.ch/author/ecosensors/))

/

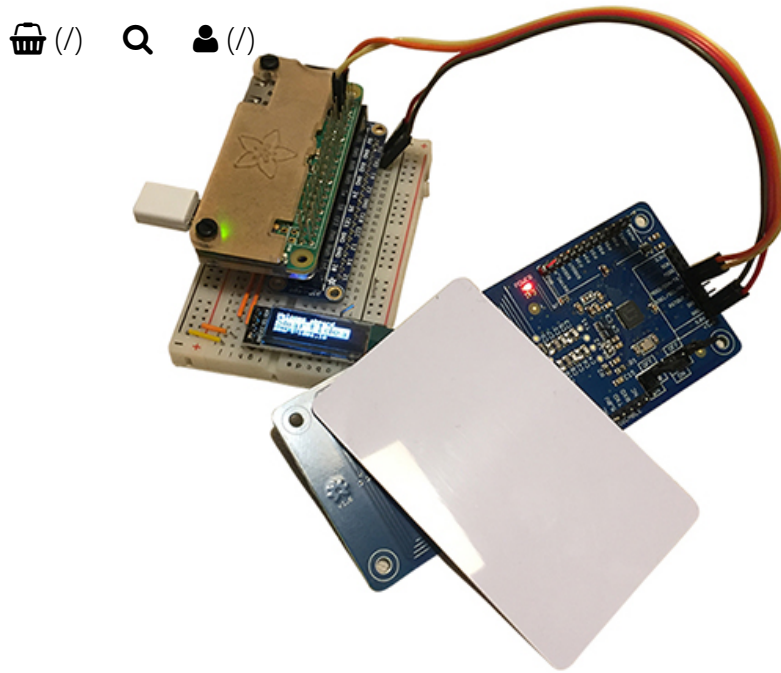
30 DÉCEMBRE 2017 ([HTTPS://ECO-SENSORS.CH/UN-RASPBERRY-POUR-LIRE-VOS-CARTES-RFID-NFC/](https://eco-sensors.ch/un-raspberry-pour-lire-vos-cartes-rfid-nfc/))

 0  4,325  0  

SYSTÈMES EMBARQUÉS ([HTTPS://ECO-SENSORS.CH/CATEGORY/SYSTEMES-EMBARQUES/](https://eco-sensors.ch/category/systemes-embarques/))

 French








Dans cet article, je vais vous résumer comment configurer un Raspberry Zero W pour lire une puce RFID (<https://fr.wikipedia.org/wiki/Radio-identification>) Vous aurez préalablement suivi la section *Configuration de votre Raspberry* (<https://eco-sensors.ch/2-faire-ses-sauvegardes-sans-connexion/>) de l'article **Faire ses sauvegardes avec raspberry.**

Matériel

- Raspberry Zero W (<https://www.raspberrypi.org/products/raspberry-pi-zero-w/>)
- NFC/RFID controller (<https://www.adafruit.com/product/364>)
- Pi Cobbler (<https://www.adafruit.com/product/2029>)
- GPIO Stacking Header (<https://www.adafruit.com/product/2223>)
- OLED I2C 128×32 (<https://www.adafruit.com/product/931>)

Connexion

OLED 128×32 I2C => Raspberry

3V => 3V
 (/)   (/)
SCL => SCL
SDA => SDA
GND => GND

LCD OLED 128×32 I2C

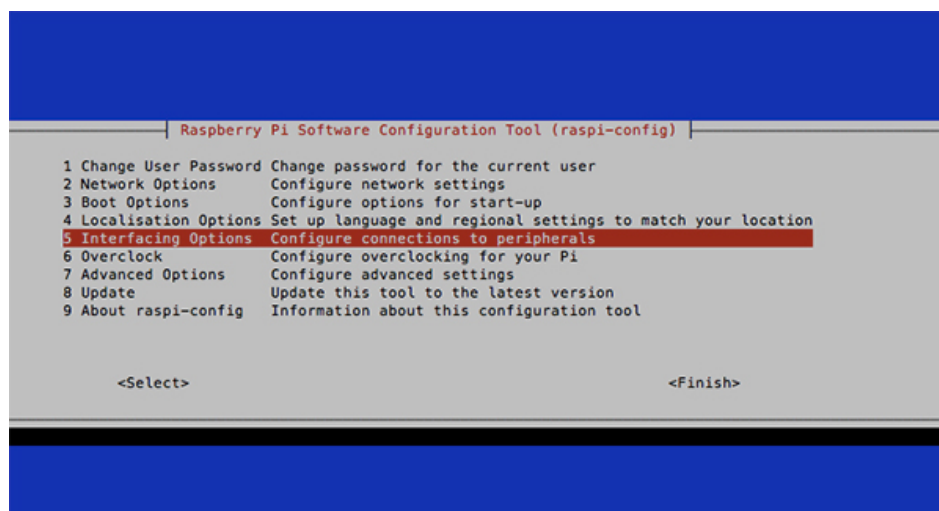
Références:

- <https://hallard.me/adafruit-oled-display-driver-for-pi/> (<http://hallard.me/adafruit-oled-display-driver-for-pi/>)
- <https://learn.adafruit.com/adafruit-nfc-rfid-on-raspberry-pi/building-libnfc> (<https://learn.adafruit.com/adafruit-nfc-rfid-on-raspberry-pi/building-libnfc>)

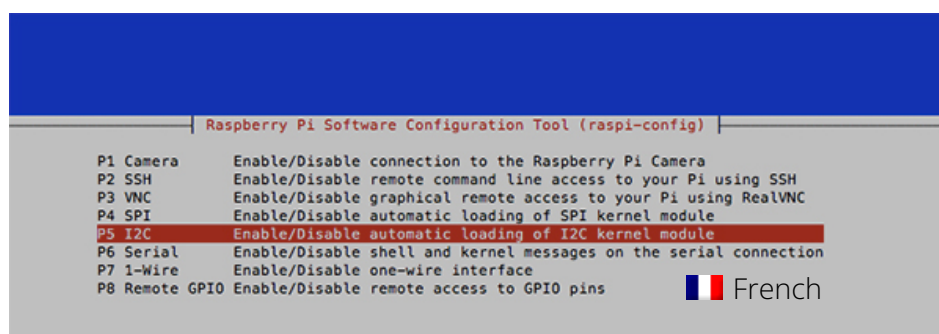
Nous allons d'abord vérifier que I2C est activé sur votre Raspberry. Ouvrez votre terminal et tapez la commande suivante

```
1 sudo raspi-config
```

Sélectionner **5 Interfacing Option**



Puis **P5 I2C**



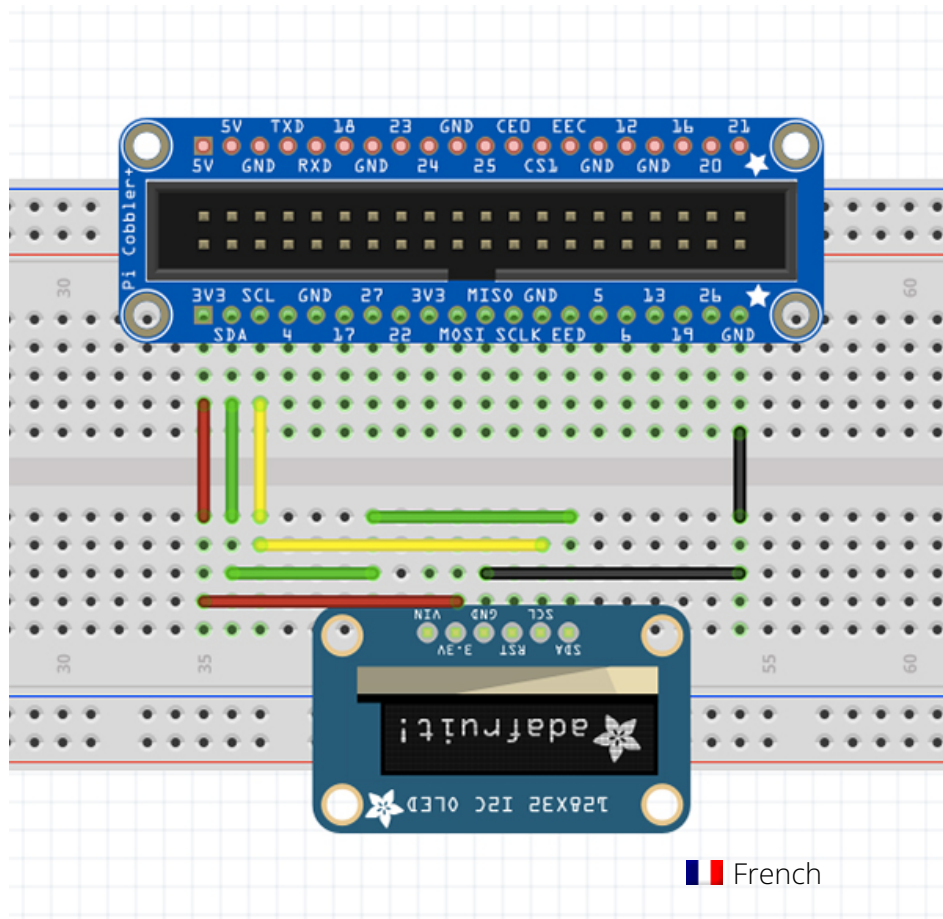


Puis, répondez YES pour l'activer. Puis cliquez sur <Finish>

Câblage

Référence: <https://learn.adafruit.com/monochrome-oled-breakouts/wiring-1-dot-3-128x64#using-with-i2c> (<https://learn.adafruit.com/monochrome-oled-breakouts/wiring-1-dot-3-128x64#using-with-i2c>)

Lisez et vérifiez bien ceci <https://learn.adafruit.com/monochrome-oled-breakouts/wiring-1-dot-3-128x64#using-with-i2c> (<https://learn.adafruit.com/monochrome-oled-breakouts/wiring-1-dot-3-128x64#using-with-i2c>) en fonction de votre carte OLED LCD



Installation

Installation des librairies (C) et de git

```
1 sudo apt-get install build-essential git-core
```

Vous pouvez déjà contrôler les périphériques I2C connectés à votre Raspberry

```
1 sudo i2cdetect -y 1
```


Installation des drivers ArduiPi_OLED

NB: ArduiPi_OLED est le successeur d'Adafruit_SSD1306

```
1 mkdir ~/Soft
2 cd ~/Soft
3 git clone https://github.com/hallard/ArduiPi_()
4 cd ArduiPi_OLED/
5 sudo make
```

Le nouveau script compilera la librairie (libArduPi_OLED *) et l'installera pour vous (c'est pourquoi nous devons utiliser sudo) dans

```
/usr/local/lib/
```

et en-têtes de bibliothèque dans

```
/usr/local/include/
```

Testons

Allez dans et compilez

```
1 cd ~/Soft/ArduPi_OLED/examples
2 sudo make
```

puis lancez le fichier de démo pour un écran 128×32
fonctionnant sur I2C

```
1 sudo ./oled_demo --verbose --oled 2
```

De mon côté, ça marche très bien.

Pour afficher votre texte au démarrage, il faut le faire démarrer



en tant que service. J'en parle ici (<https://eco-sensors.ch/un-raspberry-zero-un-lcd-et-lora-pour-recevoir-des-donnees-rfm95/>).

NFC / RFID

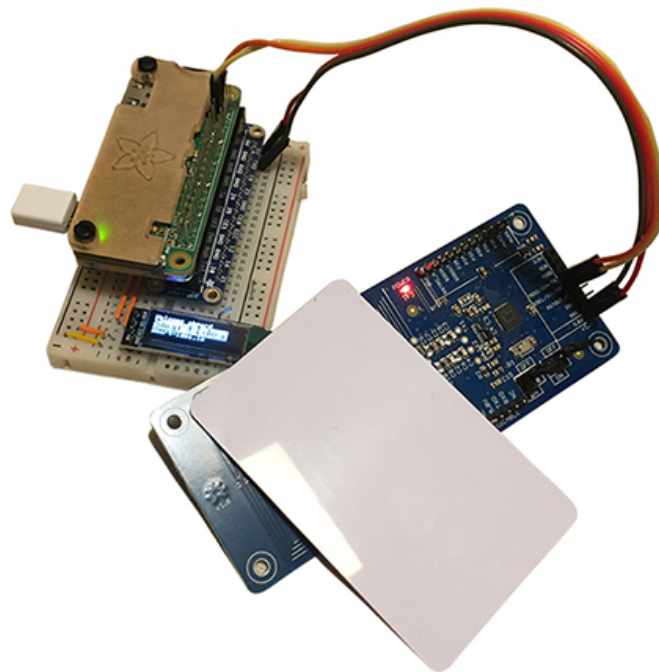
*N'hésitez pas à partager votre expérience
pour perfectionner cette partie*

Mode I2C

Références :

- <https://blog.stigok.com/post/setting-up-a-pn532-nfc-module-on-a-raspberry-pi-using-i2c> (<https://blog.stigok.com/post/setting-up-a-pn532-nfc-module-on-a-raspberry-pi-using-i2c>)
- <https://learn.adafruit.com/adafruit-nfc-rfid-on-raspberry-pi> (<https://learn.adafruit.com/adafruit-nfc-rfid-on-raspberry-pi>)

Connexion



Raspberry rfid nfc i2c

NFC Controller => Raspberry

5V => 5V

 (/)   (/)

SDA => SDA

SLC => SCL

GND => GND

Installation

```
1 sudo apt-get install i2c-tools
```

Vous pouvez maintenant scanner votre périphérique I2C

```
1 sudo i2cdetect -y 1
```

Vous devriez voir le 0x24 qui correspond à l'adresse de votre carte PN532

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
10:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20:	--	--	--	--	--	24	--	--	--	--	--	--	--	--	--	--
30:	--	--	--	--	--	--	--	--	--	--	--	--	--	3c	--	--
40:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
50:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Créez un dossier nfc dans /etc/

```
1 sudo mkdir /etc/nfc
```


Configuration

- <https://en.wikipedia.org/wiki/Autoconf> (<https://en.wikipedia.org/wiki/Autoconf>)
- https://en.wikipedia.org/wiki/GNU_Libtool (https://en.wikipedia.org/wiki/GNU_Libtool)

La prochaine étape est de configurer l'application.

Effectuez cette commande

```
1 ls /dev/i2c*
```

qui devrait afficher ceci:

```
/dev/i2c-1
```

Copiez libnfc.conf.sample dans /etc/nfc/

```
1 cd ~/Soft/libnfc
2 sudo cp libnfc.conf.sample /etc/nfc/libnfc.conf
```

   e (/com/mercy) la ligne, dans ce fichier

```
1 allow_intrusive_scan=true
```

Puis toujours dans le fichier libnfc.conf, décommentez ces lignes et corrigez-les ainsi: (/dev/i2c-1 doit correspondre au résultat de `ls /dev/i2c*`

```
1 device.name = "PN532 over I2C"
2 device.connstring = "pn532_i2c:/dev/i2c-1"
```

Installons maintenant les librairies NFC

```
1 sudo apt install libnfc5 libnfc-bin libnfc-exa
```

Installez les packages suivants:

```
1 cd ~/Soft/libnfc/  
2 sudo apt-get install autoconf  
3 sudo apt-get install libtool  
4 sudo apt-get install libpcsclite-dev libusb-dev  
5 autoreconf -vis  
6 ./configure --with-drivers=pn532_i2c --sysconf
```

L'exécution devrait se finir par ces lignes:





```

checking for log flag... yes
checking for conffiles flag... yes
checking for envvars flag... yes
checking for debug flag... no
checking which drivers to build... pn532_uart
checking for documentation request... no
checking for readline.h... not found
checking that generated files are newer than configure... c
configure: creating ./config.status
config.status: creating Doxyfile
config.status: creating Makefile
config.status: creating cmake/Makefile
config.status: creating cmake/modules/Makefile
config.status: creating contrib/Makefile
config.status: creating contrib/devd/Makefile
config.status: creating contrib/libnfc/Makefile
config.status: creating contrib/linux/Makefile
config.status: creating contrib/udev/Makefile
config.status: creating contrib/win32/Makefile
config.status: creating contrib/win32/sys/Makefile
config.status: creating contrib/win32/libnfc/Makefile
config.status: creating contrib/win32/libnfc/buses/Makefile
config.status: creating examples/Makefile
config.status: creating examples/pn53x-tamashell-scripts/Ma
config.status: creating include/Makefile
config.status: creating include/nfc/Makefile
config.status: creating libnfc.pc
config.status: creating libnfc/Makefile
config.status: creating libnfc/buses/Makefile
config.status: creating libnfc/chips/Makefile
config.status: creating libnfc/drivers/Makefile
config.status: creating test/Makefile
config.status: creating utils/Makefile
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands

Selected drivers:
acr122_pcsc..... no
acr122_usb..... no
acr122s..... no
arygon..... no
pn53x_usb..... no
pn532_uart..... no
pn532_spi..... no
pn532_i2c..... yes

```

Pour information, si vous ne spécifiez pas `-with-driver` en lançant cette commande ainsi:

1  (/)   (/) ./configure --sysconfdir=/etc --prefix=/usr

L'exécution devrait se terminer ainsi:

```
Selected drivers:
acr122_pcsc..... no
acr122_usb..... yes
acr122s..... yes
arygon..... yes
pn53x_usb..... yes
pn532_uart..... yes
pn532_spi..... yes
pn532_i2c..... yes
```

Installation de libnfc

```
1 cd ~/Soft/libnfc/
2 sudo make clean
3 sudo make install all
```

Testons!!

Référence: <http://nfc-tools.org/index.php/>

Libnfc:quick_start_example (<http://nfc-tools.org/index.php/>

Libnfc:quick_start_example)

D'abord n'oubliez pas de déplacer les deux jumper sur SEL0 sur ON et SEL1 sur OFF



Vous pouvez dès à présent poser une carte sur votre lecteur

Premier teste

Rendez-vous dans le dossier 'exemple' puis 'doc'

```
1 cd ~/Soft/libnfc/examples/doc
```

 (/)   (/)

puis lancer la commande

```
1 gcc -o quick_start_example1 quick_start_examp
```



1 sudo ./quick_start_example1



(/)



(/)

Vous devriez afficher ceci après avoir passé la carte RFID sur le lecteur

```
./quick_start_example1 uses libnfc 1.7.1
NFC reader: pn532_i2c:/dev/i2c-1 opened
The following (NFC) ISO14443A tag was found:
  ATQA (SENS_RES): 00 04
  UID (NFCID1): 67 fe a1 39
  SAK (SEL_RES): 08
```

Deuxième teste

Pour toutes ces commandes suivantes, vous trouverez plus d'information dans le manuel

- 1 `man nfc-list`
- 2 `man nfc-poll`
- 3 `man nfc-nfc-scan-device`

Vous pouvez aussi exécuter ces commandes, après avoir placé la carte sur le lecteur:

```
1 sudo nfc-list -v
```

Ce qui m'affiche

```

nfc-list uses libnfc 1.7.1
NFC device: pn532_i2c:/dev/i2c-1 opened
1 ISO14443A passive target(s) found:
ISO/IEC 14443A (106 kbps) target:
  ATQA (SENS_RES): 00 02
* UID size: single
* bit frame anticollision supported
  UID (NFCID1): ef 21 58 36
  SAK (SEL_RES): 38
* Compliant with ISO/IEC 14443-4
* Not compliant with ISO/IEC 18092
  ATS: 78 f7 b1 02 4a 43 4f 50 32 34 32 52 32
* Max Frame Size accepted by PICC: 256 bytes
* Bit Rate Capability:
  * Same bitrate in both directions mandatory
  * PICC to PCD, DS=2, bitrate 212 kbits/s supported
  * PICC to PCD, DS=4, bitrate 424 kbits/s supported
  * PICC to PCD, DS=8, bitrate 847 kbits/s supported
  * PCD to PICC, DR=2, bitrate 212 kbits/s supported
  * PCD to PICC, DR=4, bitrate 424 kbits/s supported
  * PCD to PICC, DR=8, bitrate 847 kbits/s supported
* Frame Waiting Time: 618.6 ms
* Start-up Frame Guard Time: 0.6041 ms
* Node Address not supported
* Card Identifier supported
* Historical bytes Tk: 4a 43 4f 50 32 34 32 52 32
  * Proprietary format

Fingerprinting based on MIFARE type Identification Procedure
* SmartMX with MIFARE 4K emulation
Other possible matches based on ATQA & SAK values:
* MFC 4K emulated by Nokia 6212 Classic

0 Felica (212 kbps) passive target(s) found.
0 Felica (424 kbps) passive target(s) found.
0 ISO14443B passive target(s) found.
0 ISO14443B' passive target(s) found.
0 ISO14443B-2 ST SRx passive target(s) found.
0 ISO14443B-2 ASK CTx passive target(s) found.
0 ISO14443A-3 Jewel passive target(s) found.
0 ISO14443A-2 NFC Barcode passive target(s) found.

```

ou encore

```
1 sudo nfc-list -t 1
```

qui doit afficher ceci:

```
nfc-list uses libnfc 1.7.1
NFC device: pn532_i2c:/dev/i2c-1 opened
1 ISO14443A passive target(s) found:
ISO/IEC 14443A (106 kbps) target:
ATQA (SENS_RES): 00 04
UID (NFCID1): 67 fe a1 39
SAK (SEL_RES): 08
```

Ce qui ressemble à ce que nous avons vu plus haut.

En gras sont les informations sur la carte.

Pour plus d'information sur la commande nfc-list, saisissez la commande.

Mais encore

```
1 sudo nfc-scan-device -v
```

affichera

```
nfc-scan-device uses libnfc 1.7.1
1 NFC device(s) found:
- pn532_i2c:/dev/i2c-1:
  pn532_i2c:/dev/i2c-1
  chip: PN532 v1.6
  initiator mode modulations: ISO/IEC 14443A (106 kbps), FeliCa
  target mode modulations: ISO/IEC 14443A (106 kbps), FeliCa
```

et finalement

```
1 sudo nfc-poll
```

qui semble attendre qu'une carte soit présentée et termine
lorsque la carte est retirée

```
nfc-poll uses libnfc 1.7.1
NFC reader: pn532_i2c:/dev/i2c-1 opened
NFC device will poll during 30000 ms (20 pollings of 300 ms)
ISO/IEC 14443A (106 kbps) target:
  ATQA (SENS_RES): 00 04
  UID (NFCID1): 67 fe a1 39
  SAK (SEL_RES): 08
nfc_initiator_target_is_present: Target Released
Waiting for card removing...done
```

Evolution

Malheureusement, je n'ai pas terminé en faisant de sorte que les résultats s'affichent sur l'écran OLED, ce que je ferai dans un avenir proche. Mais au moins vous pouvez tester la fonctionnalité d'un lecteur de cartes RFID.

Si vous avez la solution pour le faire, d'ici là que je le fasse, n'hésitez pas à partager!

Bravo!!!

Vous venez de configurer votre Raspberry pour qu'il affiche du texte. Mais vous venez surtout de configurer votre Raspberry pour qu'il puisse lire une carte RFID. J'apporterai encore deux trois détails tout prochainement.

Soutenez-nous

Si vous avez aimé cet article, publiez-le ou faites un don (<https://eco-sensors.ch/product/don/>) de quelques Euro pour financer ces projets et le matériel.

🔖 **ARDUIPI_OLED** ([HTTPS://ECO-SENSORS.CH/TAG/ARDUIPI_OLED/](https://eco-sensors.ch/tag/ARDUIPI_OLED/)) **GIT** ([HTTPS://ECO-SENSORS.CH/TAG/GIT/](https://eco-sensors.ch/tag/GIT/)) **GPIO** ([HTTPS://ECO-SENSORS.CH/TAG/GPIO/](https://eco-sensors.ch/tag/GPIO/)) **I2C** ([HTTPS://ECO-SENSORS.CH/TAG/I2C/](https://eco-sensors.ch/tag/I2C/)) **I2C-TOOLS** ([HTTPS://ECO-SENSORS.CH/TAG/I2C-TOOLS/](https://eco-sensors.ch/tag/I2C-TOOLS/)) **LIBI2C-DEV** ([HTTPS://ECO-SENSORS.CH/TAG/LIBI2C-DEV/](https://eco-sensors.ch/tag/LIBI2C-DEV/)) **NFC** ([HTTPS://ECO-SENSORS.CH/TAG/NFC/](https://eco-sensors.ch/tag/NFC/)) **OLED** ([HTTPS://ECO-SENSORS.CH/TAG/OLED/](https://eco-sensors.ch/tag/OLED/)) **RASPBERRY** ([HTTPS://ECO-SENSORS.CH/TAG/RASPBERRY/](https://eco-sensors.ch/tag/RASPBERRY/)) **RFID** ([HTTPS://ECO-SENSORS.CH/TAG/RFID/](https://eco-sensors.ch/tag/RFID/)) **SSD1306** ([HTTPS://ECO-SENSORS.CH/TAG/SSD1306/](https://eco-sensors.ch/tag/SSD1306/))

PREV POST

NEXT POST

(<https://eco-sensors.ch/un-raspberry-zero-un-lcd-et-lora-pour-recevoir-des-donnees-rfm95/>)

(<https://eco-sensors.ch/comment-orienter-une-camera-depuis-la-planete/>)

planète / French





Un raspberry Zero, un LCD et LoRa pour

25 DÉCEMBRE 2017

Leave **Comment:**

FULL NAME

EMAIL ADDRESS

PHONE NUMBER

YOUR COMMENT

© Copyright 2020 EcoSensors. - Tous droits réservés.