

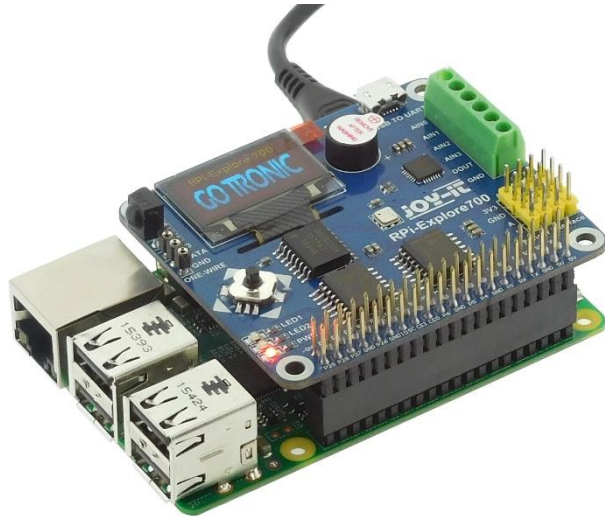
# Manuel d'utilisation du module RPI-Explorer700

## Introduction

Cher client,

Merci d'avoir acheté notre produit.

Veuillez observer les instructions ci-dessous avant la première utilisation.

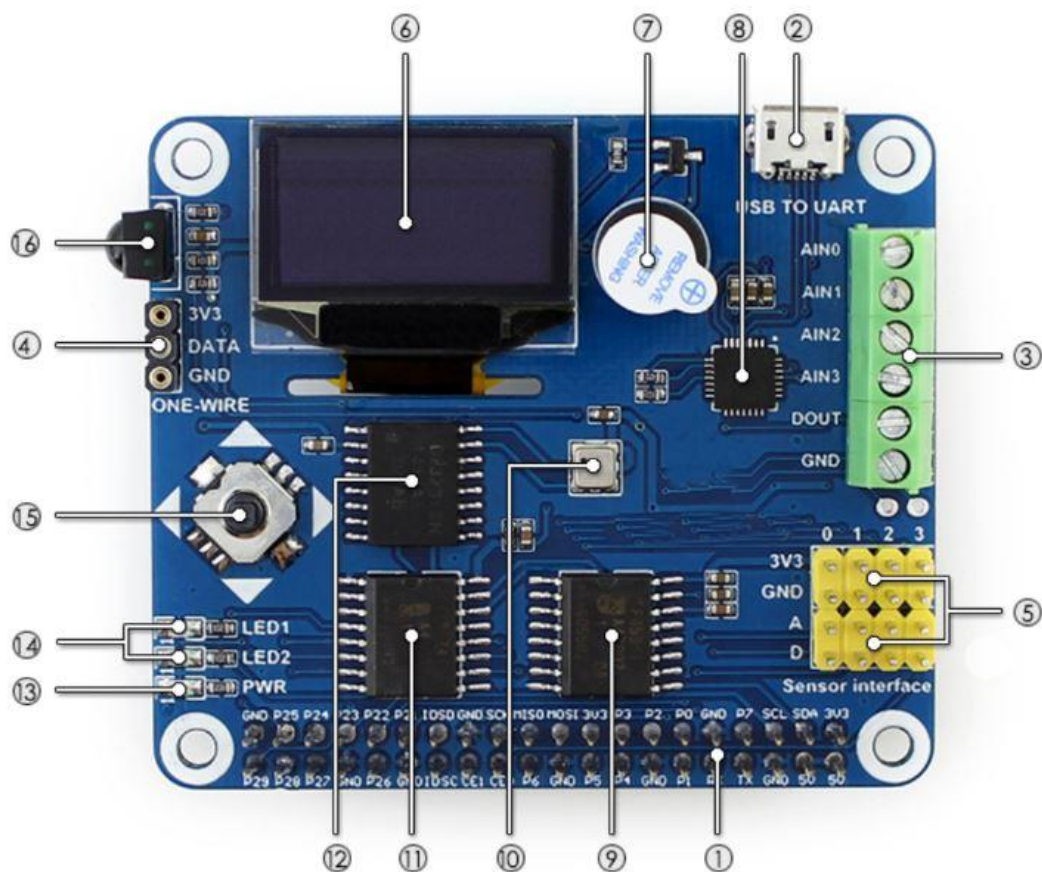


## Contenu

<b>Présentation .....</b>	<b>2</b>
<b>Connecter au Raspberry Pi.....</b>	<b>3</b>
Prérequis .....	3
Mise à jour et installation des bibliothèques.....	3
Activation des bus i2c et série .....	4
Téléchargement des exemples .....	5
<b>Exemples de programmes.....</b>	<b>6</b>
LED1 .....	6
Bouton du joystick .....	6
PCF8574 : LED2 et joystick.....	7
BMP180 : température et pression.....	8
DS3231 : horloge temps réel.....	8
DS18B20 : capteur de température.....	9
Capteur infrarouge.....	9
Ecran OLED.....	10

## Présentation

Le module RPI-Explorer700 regroupe une série de capteurs, afficheur graphique, entrées et sorties digitales ou analogiques sur une carte d'extension tout-en-un pour une utilisation didactique avec un Raspberry Pi 2 ou 3.



1. Interface GPIO du Raspberry Pi : à connecter au Raspberry Pi
2. Port série USB/UART : pour contrôler le Raspberry Pi depuis un terminal
3. Bornier d'entrées et sorties analogiques
4. Interface 1-WIRE : connexion de modules 1-Wire (p.ex. DS18B20)
5. Interface de capteurs : connexion de divers capteurs
6. Ecran OLED : driver SSD1306, 128x64 pixels, interface SPI
7. Buzzer
8. CS2101 : Convertisseur USB/UART
9. PCF8591 : Convertisseur analogue/digital 8 bits (i2c)
10. BMP180 : Capteur de pression et température (i2c)
11. PCF8574 : Circuit d'expansion des entrées et sorties (i2c)
12. DS3231 : Horloge temps réel (i2c), avec emplacement pour pile de sauvegarde au dos
13. Indicateur d'alimentation
14. Leds programmables LED1 et LED2
15. Joystick
16. Capteur infrarouge LFN0038K

## Connecter au Raspberry Pi

Le module s'enfiche simplement sur le connecteur GPIO du Raspberry Pi A+ / B+ / 2 ou 3 modèle B. Aucun autre branchement n'est nécessaire, mais il est conseillé de d'utiliser les entretoises livrées pour stabiliser la carte. Si vous voulez utiliser une pile de sauvegarde de l'horloge temps réel (non incluse), placez-la au préalable dans le support au dos de la carte.



Carte placée sur un Raspberry Pi B+, 2 ou 3 modèle B



Carte placée sur un Raspberry Pi A+

## Prérequis

Les explications et programmes qui suivent ont été créés et testés pour un Raspberry utilisant le système d'exploitation Raspbian.

Un guide d'installation est disponible ici :

<http://www.gotronic.fr/userfiles/www.gotronic.fr/files/Raspberry/GuideRPI.pdf>

Un guide de démarrage est disponible ici :

<http://www.gotronic.fr/userfiles/www.gotronic.fr/files/Raspberry/Raspberry%20Joyit.pdf>

## Mise à jour et installation des bibliothèques

Vérifiez la mise à jour du système d'exploitation de votre Raspberry et installez les bibliothèques principales en lançant les lignes de commandes ci-dessous dans LXTerminal :  
**(Attention : le Raspberry doit être connecté au réseau internet)**

Mises à jour du système :

```
sudo apt-get update  
sudo apt-get upgrade
```

Bibliothèque GPIO :

```
sudo apt-get install python-pip python-dev build-essential
sudo pip install RPi.GPIO
```

Bibliothèque smbus (i2c) :

```
sudo apt-get install python-smbus
```

Bibliothèque série :

```
sudo apt-get install python-serial
```

Bibliothèque imaging :

```
sudo apt-get install python-imaging
```

## Activation des bus i2c et série

Si ce n'est pas encore fait, il faut activer le bus I2C. Ouvrez les paramètres :

```
sudo raspi-config
```

Sélectionnez le menu des options avancées « Advanced Option »

```
Raspberry Pi Software Configuration Tool (raspi-config)
1 Expand Filesystem          Ensures that all of the SD card s
2 Change User Password      Change password for the default u
3 Boot Options              Configure options for start-up
4 Internationalisation Options Set up language and regional sett
5 Enable Camera            Enable this Pi to work with the R
6 Overclock                Configure overclocking for your P
7 Advanced Options         Configure advanced settings
8 About raspi-config       Information about this configurat

<Select>                    <Finish>
```

Activez les options SPI, I2C et Serial : « Enable/Disable automatic loading of SPI kernel module »

```
Raspberry Pi Software Configuration Tool (raspi-config)
A1 Overscan                You may need to configure oversca +
A2 Hostname                Set the visible name for this Pi
A3 Memory Split            Change the amount of memory made
A4 SSH                     Enable/Disable remote command lin
A5 VNC                     Enable/Disable graphical remote a
A6 SPI                     Enable/Disable automatic loading
A7 I2C                     Enable/Disable automatic loading
A8 Serial                  Enable/Disable shell and kernel m
A9 Audio                   Force audio out through HDMI or 3
AA 1-Wire                  Enable/Disable one-wire interface +

<Select>                    <Back>
```

Ensuite, le fichier *Modules* doit être modifié :

Ouvrez le fichier *Module* :

```
sudo nano /etc/modules
```

Copiez les deux lignes à la fin du fichier :

```
i2c-bcm2708  
i2c-dev
```

Sauvegardez le fichier (*Ctrl+O* et *Enter*) et quittez le fichier (*Ctrl+X*).

Redémarrez le Raspberry :

```
sudo reboot
```

Remarque : le port série n'est pas utilisable avec un Raspberry Pi 3 B car les broches 14 et 15 sont connectées au composant Bluetooth intégré.

### [Téléchargement des exemples](#)

Le dossier (.zip) contenant les programmes d'exemples utilisés dans les chapitres suivants est téléchargeable via le lien : <http://cloud.joy-it.net/index.php/s/zDgFZhhNiiWJvyc>

Ce dossier est à décompresser et placer dans le répertoire général du Raspberry (par défaut /home/pi)

## Exemples de programmes

Les programmes présentés ci-dessous sont inspirés et traduits du manuel de Joy-it et se concentrent sur les exemples en langage python. Le manuel Joy-it contient d'autres exemples de programmes dans d'autres langages, dont bcm2835 et wiringPi.

### LED1

Ouvrez *LXTerminal* et accédez au répertoire :

```
cd Explorer700/LED/python
```

Lancez le premier programme de test avec la commande suivante :

```
sudo python led.py
```

La LED1 devrait se mettre à clignoter.

Appuyez sur *CTRL+C* pour arrêter le programme.

Lancez le deuxième programme de test avec la commande suivante :

```
sudo python pwm.py
```

La LED1 devrait se mettre à clignoter graduellement.

Appuyez sur *CTRL+C* pour arrêter le programme.

### Bouton du joystick

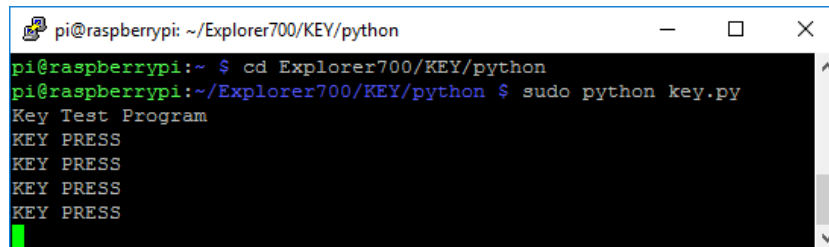
Ouvrez *LXTerminal* et accédez au répertoire :

```
cd Explorer700/KEY/python
```

Lancez le programme de test avec la commande suivante :

```
sudo python key.py
```

Appuyez verticalement sur le joystick. Le texte « KEY PRESS » devrait apparaître sur le terminal à chaque pression :



```
pi@raspberrypi: ~/Explorer700/KEY/python
pi@raspberrypi:~ $ cd Explorer700/KEY/python
pi@raspberrypi:~/Explorer700/KEY/python $ sudo python key.py
Key Test Program
KEY PRESS
KEY PRESS
KEY PRESS
KEY PRESS
```

Appuyez sur **CTRL+C** pour arrêter le programme.

## PCF8574 : LED2 et joystick

Ouvrez *LXTerminal* et accédez au répertoire :

```
cd Explorer700/PCF8574/python
```

Lancez le premier programme de test avec les commandes suivantes :

```
sudo python led.py
```

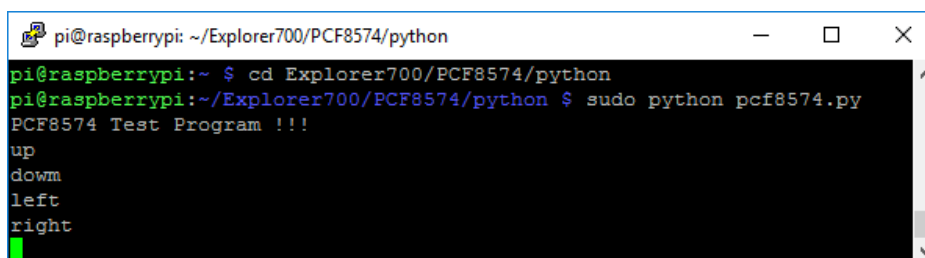
La LED2 devrait se mettre à clignoter.

Appuyez sur **CTRL+C** pour arrêter le programme.

Lancez le premier programme de test avec les commandes suivantes :

```
sudo python pcf8574.py
```

Faites bouger le joystick en avant, arrière, gauche et droite. La direction devrait apparaître sur le terminal :



```
pi@raspberrypi: ~/Explorer700/PCF8574/python
pi@raspberrypi:~ $ cd Explorer700/PCF8574/python
pi@raspberrypi:~/Explorer700/PCF8574/python $ sudo python pcf8574.py
PCF8574 Test Program !!!
up
down
left
right
```

Appuyez sur **CTRL+C** pour arrêter le programme.

## BMP180 : température et pression

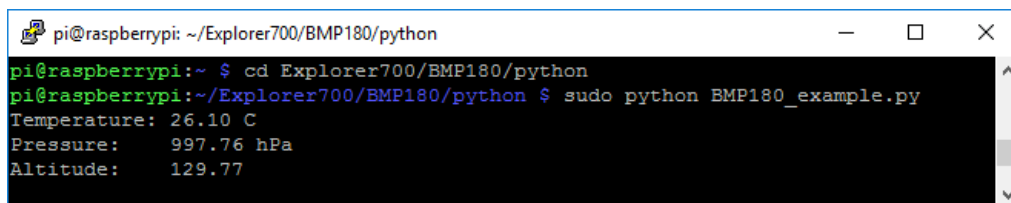
Ouvrez *LXTerminal* et accédez au répertoire :

```
cd Explorer700/BMP180/python
```

Lancez le programme de test avec la commande suivante :

```
sudo python BMP180_example.py
```

La température, la pression et l'altitude s'affichent sur le terminal :



```
pi@raspberrypi: ~/Explorer700/BMP180/python
pi@raspberrypi:~ $ cd Explorer700/BMP180/python
pi@raspberrypi:~/Explorer700/BMP180/python $ sudo python BMP180_example.py
Temperature: 26.10 C
Pressure:    997.76 hPa
Altitude:   129.77
```

Appuyez sur *CTRL+C* pour arrêter le programme.

## DS3231 : horloge temps réel

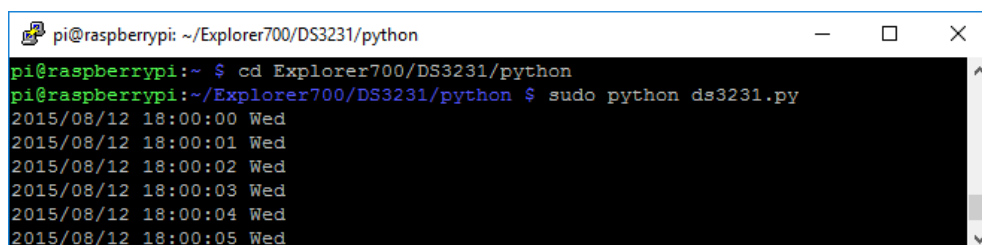
Ouvrez *LXTerminal* et accédez au répertoire :

```
cd Explorer700/DS3231/python
```

Lancez le programme de test avec la commande suivante :

```
sudo python ds3231.py
```

La date et l'heure de l'horloge temps réel s'affichent sur le terminal :



```
pi@raspberrypi: ~/Explorer700/DS3231/python
pi@raspberrypi:~ $ cd Explorer700/DS3231/python
pi@raspberrypi:~/Explorer700/DS3231/python $ sudo python ds3231.py
2015/08/12 18:00:00 Wed
2015/08/12 18:00:01 Wed
2015/08/12 18:00:02 Wed
2015/08/12 18:00:03 Wed
2015/08/12 18:00:04 Wed
2015/08/12 18:00:05 Wed
```

Appuyez sur *CTRL+C* pour arrêter le programme.



## DS18B20 : capteur de température

Cet exemple utilise la sonde de température DS18B20 livrée avec la carte.

Placez la sonde dans le bornier ONE-WIRE (côté plat vers l'intérieur de la carte).

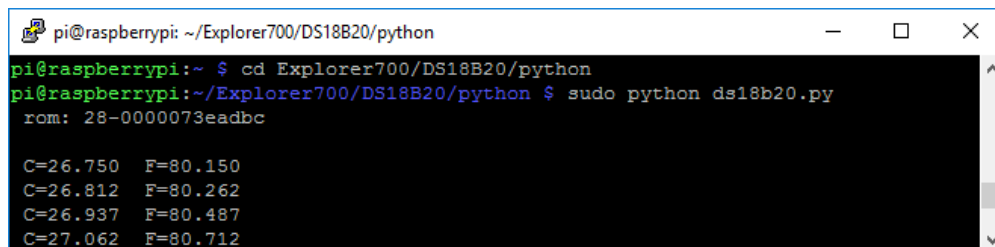
Ouvrez *LXTerminal* et accédez au répertoire :

```
cd Explorer700/DS18B20/python
```

Lancez le programme de test avec la commande suivante :

```
sudo python ds18b20.py
```

La température mesurée s'affiche sur le terminal :



```
pi@raspberrypi: ~/Explorer700/DS18B20/python
pi@raspberrypi:~$ cd Explorer700/DS18B20/python
pi@raspberrypi:~/Explorer700/DS18B20/python$ sudo python ds18b20.py
rom: 28-0000073eadbc

C=26.750 F=80.150
C=26.812 F=80.262
C=26.937 F=80.487
C=27.062 F=80.712
```

Appuyez sur *CTRL+C* pour arrêter le programme.

## Capteur infrarouge

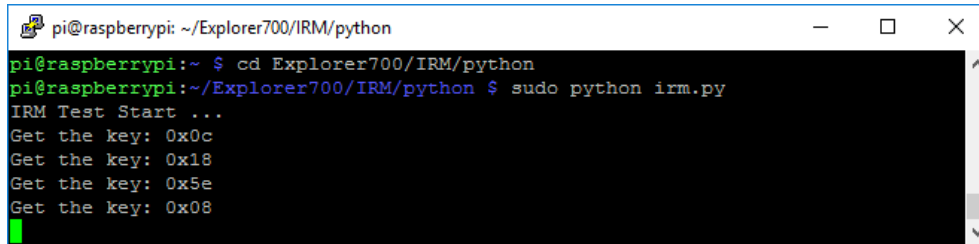
Ouvrez *LXTerminal* et accédez au répertoire :

```
cd Explorer700/IRM/python
```

Lancez le premier programme de test avec les commandes suivantes :

```
sudo python irm.py
```

Pointez une télécommande infrarouge vers le récepteur de la carte et appuyez sur des touches. Le code reçu par la carte s'affiche sur le terminal (hexadécimal).



```
pi@raspberrypi: ~/Explorer700/IRM/python
pi@raspberrypi:~ $ cd Explorer700/IRM/python
pi@raspberrypi:~/Explorer700/IRM/python $ sudo python irm.py
IRM Test Start ...
Get the key: 0x0c
Get the key: 0x18
Get the key: 0x5e
Get the key: 0x08
```

Appuyez sur **CTRL+C** pour arrêter le programme.

## Ecran OLED

Ouvrez *LXTerminal* et accédez au répertoire :

```
cd Explorer700/OLED/python
```

Lancez les commandes suivantes pour exécuter les différents exemples de programmes :

- Formes géométriques et texte :

```
sudo python oled.py
```

→ Cet exemple permet d'afficher une ellipse, un rectangle, un triangle, une croix et le texte « Hello World » sur l'écran.

- Texte

```
sudo python dispchar.py
```

→ Cet exemple permet d'afficher six lignes de texte à l'écran.

- Image

```
sudo python image.py
```

→ Cet exemple affiche une image (.ppm) à l'écran

- Animation

```
sudo python animate.py
```

→ Cet exemple fait défiler un texte à l'écran en forme de vague  
Appuyez sur **CTRL+C** pour arrêter le programme.

- Logo

```
sudo python waveshare.py
```

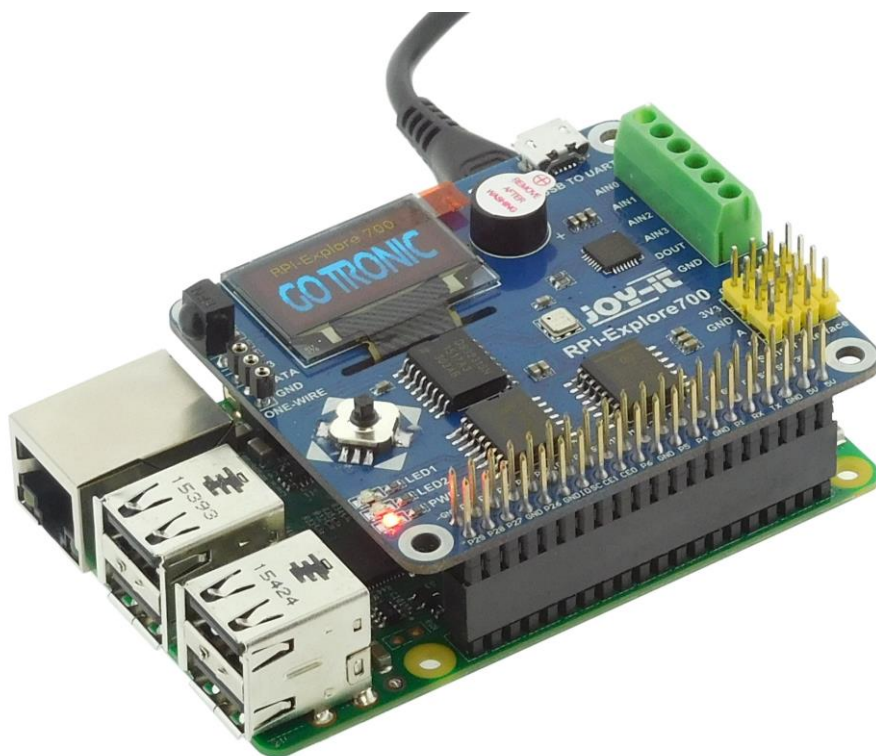
→ Cet exemple affiche un logo (.bmp) à l'écran

# GO TRONIC

ROBOTIQUE ET COMPOSANTS ÉLECTRONIQUES

Si vous rencontrez des problèmes, merci de nous contacter par courriel à :

[sav@gotronic.fr](mailto:sav@gotronic.fr)



Coordonnées du fabricant :

# **JOY-IT**<sup>®</sup>

service@joy-it.net

+49 (0)2845 9360 – 50