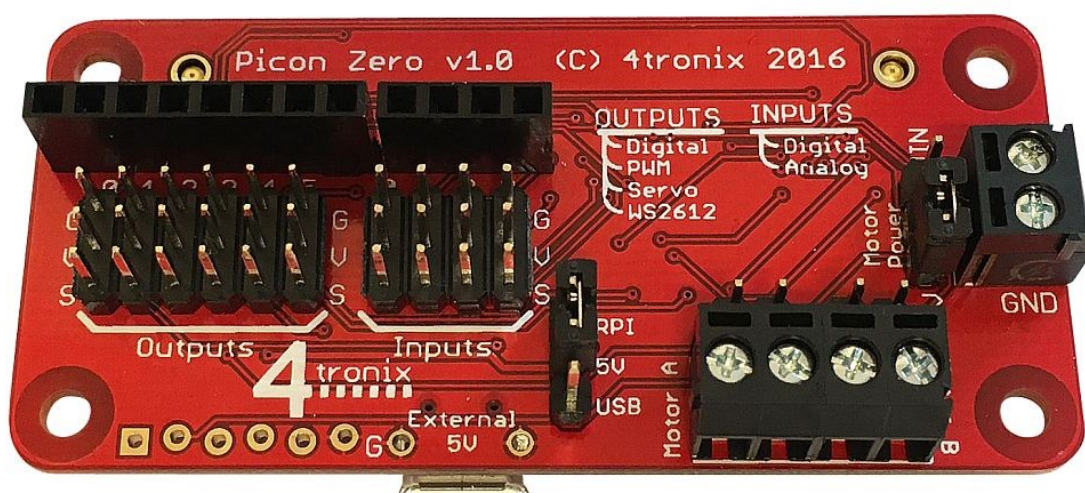


PiconZero

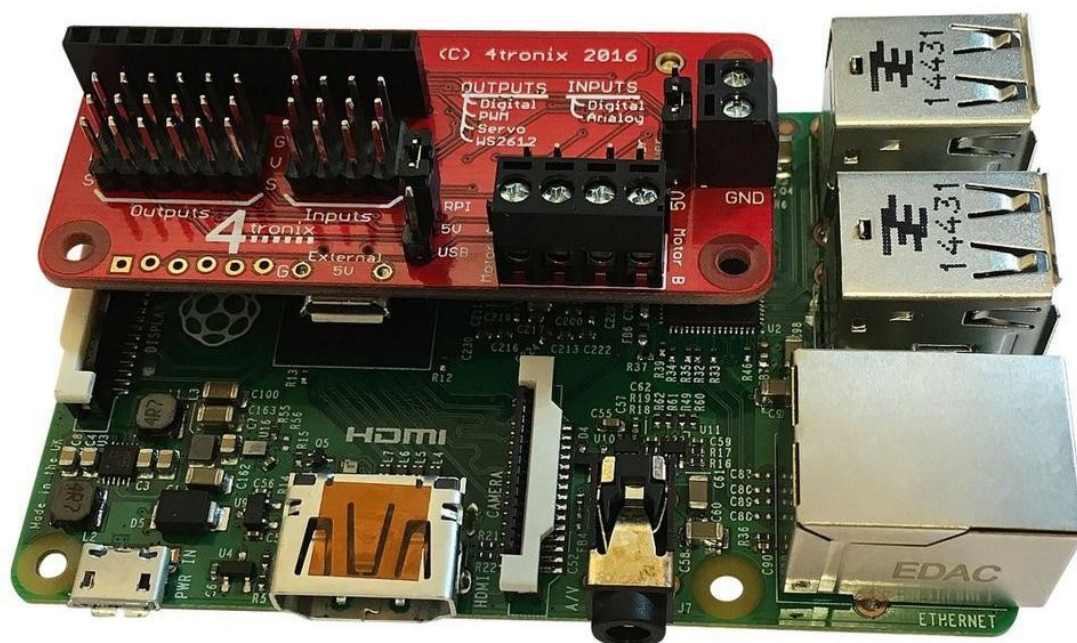
Qu'est-ce que le PiconZero



Installation

Matérielle

Emboîter le module PiconZero comme ceci :



Les pins du PiconZero s'emboîtent sur la Raspberry.

Attention : il ne tient pas très bien.

Logicielle

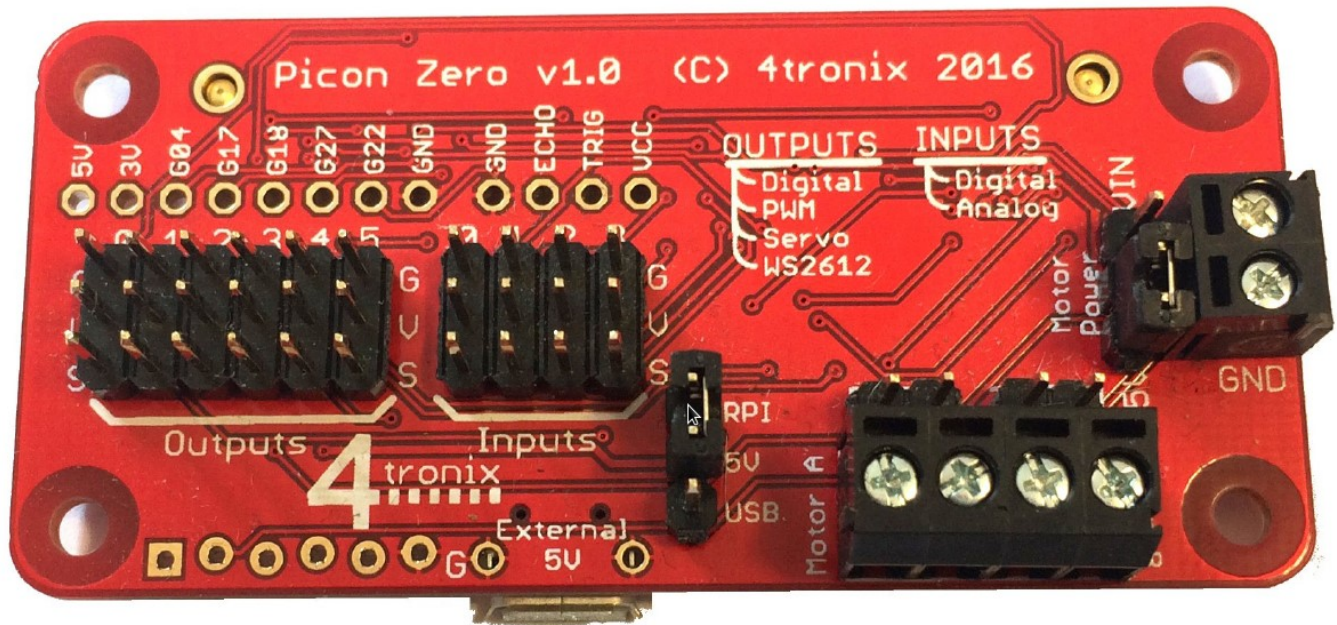
1. Assurez-vous d'avoir les bibliothèques Python (`apt-get install python-smbus python3-smbus python-dev python3-dev`)
2. Télécharger le fichier `piconz.sh` s'il n'est pas encodé disponible :
3. Soit en l'envoyant via SFTP (disponible dans les fichiers du Wiki)
4. Soit via Internet une fois connecté au Raspberry : `wget http://4tronix.co.uk/piconz.sh -O piconz.sh`
5. Assurez-vous d'avoir les paramètres `dtpram=i2c1` et `dtpram=i2c_arm` activés dans le fichier `/boot/config.txt` (`echo "dtpram=i2c1=on" >> /boot/config.txt; echo "dtpram=i2c_arm=on" >> /boot/config.txt`). Ensuite redémarrez.
6. Exécuter le script `piconz.sh` qui va créer un dossier `piconzero` dans votre `home`.

Configuration

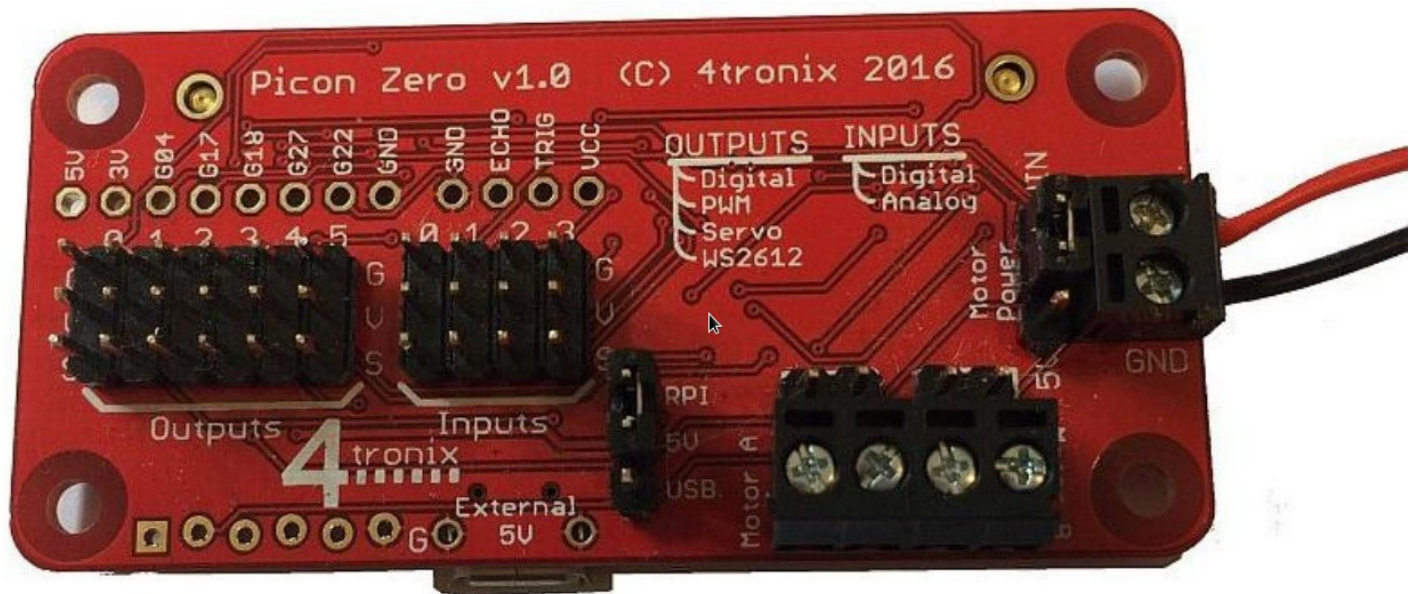
Alimentation

La configuration de l'alimentation va se faire sur base des *jumpeurs*. Choisissez comment vous voulez alimenter les moteurs et placer les *jumpeurs* de manière similaire à l'illustration correspondante.

Alimenter les moteurs avec la même source que le Raspberry



Alimenter les moteurs avec une source extérieure au Raspberry



Alimenter les moteurs via un port USB (5V)



- `piconzero.py` : bibliothèque pour le PiconZero.
- `hcsr04.py` : bibliothèque pour le capteur ultra-sonique possède également des fichiers de tests fonctionnels qui peuvent être utiles comme exemple d'utilisation :
- `motorTest.py` : montre comment on peut activer les moteurs afin de déplacer le robot sur base du clavier.
- `ioTest.py` : montre comment on peut lire des entrées analogiques sur base d'un potentiomètre en entrée 0 (la 1ère) et l'utiliser pour contrôler des périphériques variés.
- `sonarTest.py` : montre comment on peut mesurer la distance depuis un capteur HC-SR04.
- `tempTest.py` : montre comment on peut lire la température depuis un capteur DS18B20.
- `10linesTest.py` : simple programme de démonstration qui lit des entrées analogiques et qui produit des sorties pour des périphériques variés.
- `pixelTest.py` : flash tous les néo-pixels de blancs (on) à off et vice versa.

Utilisation

Moteur pas à pas

Servomoteurs

Autres

Référez-vous aux fiches officielles de notre fournisseur 4Tronix : <http://4tronix.co.uk/piconzero/WS0x.pdf> où x va de 1 à 8 inclus. Elles sont également disponibles dans les fichiers du Wiki.