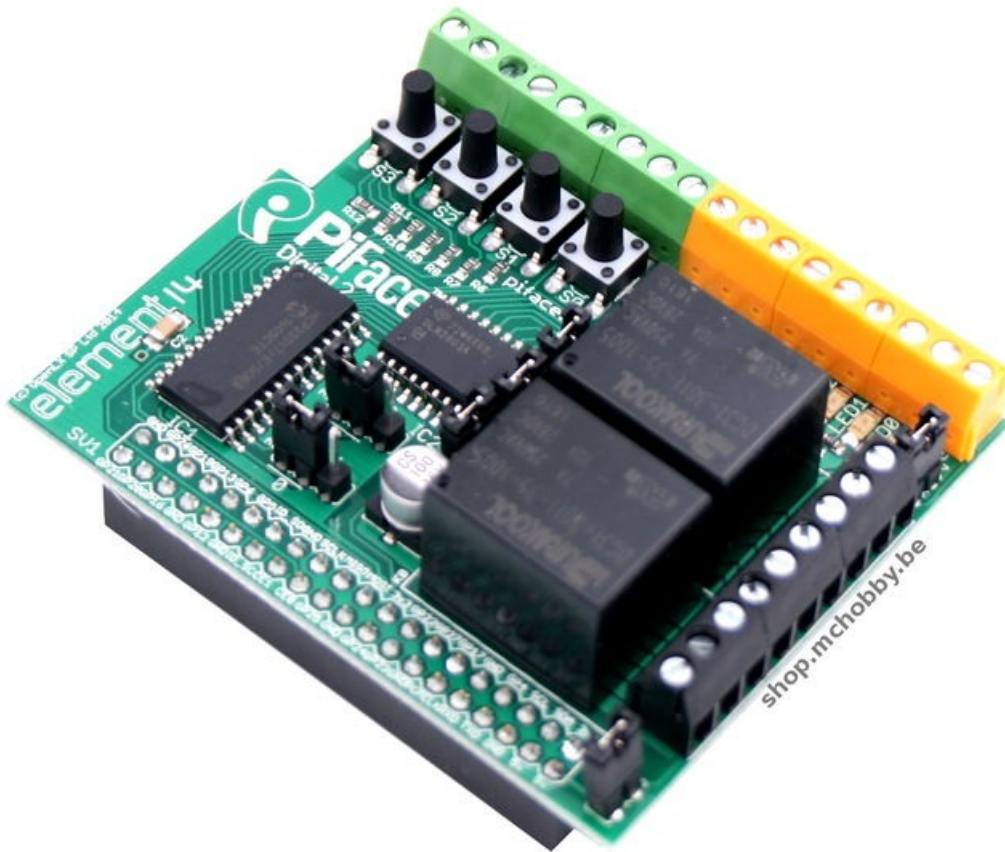


Guide utilisateur du PiFace Digital 2

Un guide pour démarrer rapidement et facilement avec le PiFace.
(version 0.1)

Traduit par MicroControleur Hobby (shop.mchobby.be)
Compilé depuis la traduction maintenue sur <https://wiki.mchobby.be/index.php?title=PiFace2-Manuel>
Les hyperliens sont disponibles sur la version en ligne du document.
Translated by MicroControleur Hobby (shop.mchobby.be)
Compiled from online translation available at <https://wiki.mchobby.be/index.php?title=PiFace2-Manuel>
Hyperlinks are available on the online version of this document.

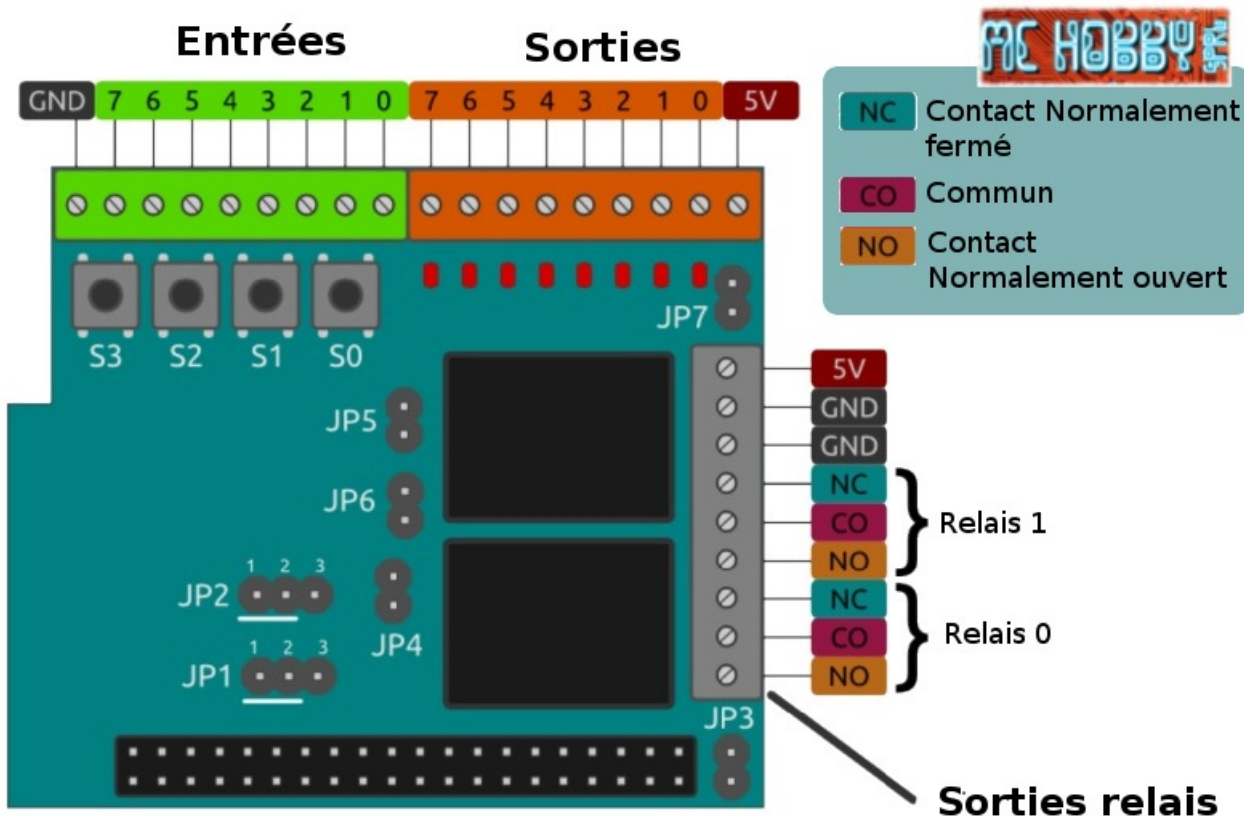
Introduction



PiFace Digital 2 se connecte sur Raspberry Pi. Conçu pour les écoles et débutants, il permet de "sentir" le monde extérieur avec ses entrées et d'agir sur son environnement par l'intermédiaire de ses sorties et relais.

PiFace Digital est conçu pour être connecté directement sur le GPIO de Raspberry Pi 3, 2, B+ et A+.

Evolution naturelle de la version 1 pour le GPIO 40 broches, PiFace digital 2 est livré assemblé. Il est donc prêt à l'emploi en quelques minutes en Python 2, Python 3, Scratch et C.



Avec PiFace Digital 2, vous pouvez détecter l'état d'un bouton poussoir (switch), d'un contact de porte, d'une pastille à pression et de tous les autres types de senseurs de type switch (contact fermé ou ouvert).

Une fois l'état détecté, vous pouvez écrire vos propres programmes pour Raspberry Pi qui déterminent comment votre Pi doit répondre à cette action du monde extérieur.

Vous pouvez piloter des moteurs de puissance, des actionneurs, LEDs ou tout ce que vous voulez d'autre.

Une seule source d'alimentation



Ne branchez pas de source d'alimentation externe sur le PiFace SI le PiFace est alimenté par l'intermédiaire du Raspberry-Pi via le cavalier JP3. Une seule alimentation 5V doit être présente sur le "Pi + PiFace" si JP3 est en place.

Recommandation

- La carte PiFace Digital 2 n'est pas conçue pour les systèmes critiques.
- Ne pas exposer à l'eau, buée ou températures extrêmes (sous -40°C ou au-dessus de 85°C).
- Manipuler avec soin pour éviter d'endommager électriquement ou mécaniquement la carte (le périphérique ou les connecteurs).
- Prenez les précautions nécessaires pour minimiser les risques de décharge électrostatique.
- La connexion de périphériques non approuvés peut affecter la conformité ou dégrader le matériel. Cela invalide la garantie.
- Les connexions sur le PiFace (incluant les circuits externes) et autres cartes doivent être réalisées lorsque le tout est hors tension.
- Assurez-vous que les circuits branchés sur le PiFace utilisent une source d'alimentation adéquate (qui sont conformes aux standards et règles en vigueur dans votre pays).
- Les informations de conformité sont disponibles en fin de ce document (<http://df.mchobby.be/RASP-PIFACE-DIGITAL2/Operating-Instruction.pdf>).

A Propos de MCHobby

Crédit de traduction

Sommaire

- 1 A propose de .: MC Hobby .:
- 2 Licence
- 3 Crédit de traduction
- 4 Limite de traduction
 - 4.1 Anglicisme
 - 4.2 Code source
- 5 Autorisations
- 6 Signaler une erreur

A propose de .: MC Hobby .:

MCHobby investit du temps et de l'argent dans la réalisation de traduction et/ou documentation. C'est un travail long et fastidieux réalisé dans l'esprit Open-Source... donc gratuit et librement accessible.

SI vous aimez nos traductions et documentations ALORS aidez nous à en produire plus en achetant vos produits chez MCHobby <https://shop.mchobby.be> .



MC Hobby SPRL, basé en Belgique, est le traducteur de ce manuel.

MC Hobby cherche à promouvoir, en français, la plateforme Open-Source Raspberry-Pi, ses extensions et exemples de programmation afin de la rendre accessible au plus grand nombre.

Licence

En commun accord avec XXXX, cette traduction est mise à disposition sous licence CC-BY-SA (*autorisation pending*)

Crédit de traduction

Toute référence, mention ou extrait de cette traduction doit être explicitement accompagné du texte de crédit de traduction.

Ce texte est repris en "pied de document" sur toutes les pages/documents, merci d'en prendre connaissance.

Limite de traduction

Anglicisme

Du fait que de nombreux anglicismes soient utilisés au jour le jour dans les milieux techniques, je me suis permis d'en préserver quelques-uns qui me semblaient avoir plus de sens dans la langue de Shakespeare que traduit dans la langue de Voltaire. Par ailleurs, ces anglicismes pourraient se révéler fort utiles lors de vos prochaines recherches sur Internet.

Ainsi, vous retrouverez les termes suivants :

- Pin : fait référence à une « broche » de raccordement d'un composant (Anglicisme fort répandu).
- Datasheet : fait référence à la « fiche technique » d'un composant.
- Pin Header : malheureusement intraduisible sans en perdre le sens mais vous aurez vite identifié les « Pinheaders ». La traduction courant est *connecteur*, ce terme restant beaucoup trop vague.

• LED : ce sont les diodes électroluminescentes (aussi appelées DEL). Cependant, le terme LED est si répandu qu'il a été préservé. • RGB : Fait référence aux 3 couleurs fondamentales Rouge Vert Bleu (Red Green Blue). Si le terme RVB est d'usage fréquent en Belgique et en France, l'acronyme RGB reste encore le plus répandu.

Code source

Nous avons traduit les commentaires dans les programmes afin de les rendre plus intelligibles.

Par contre, nous n'avons pas modifié les codes sources (nom des variables, ...) qui, eux, restent ceux fournis par le concepteur et le fabricant du kit.

Autorisations

Le présent manuel a été traduit avec l'autorisation de xxxxx (URL) *Autorisation pending*

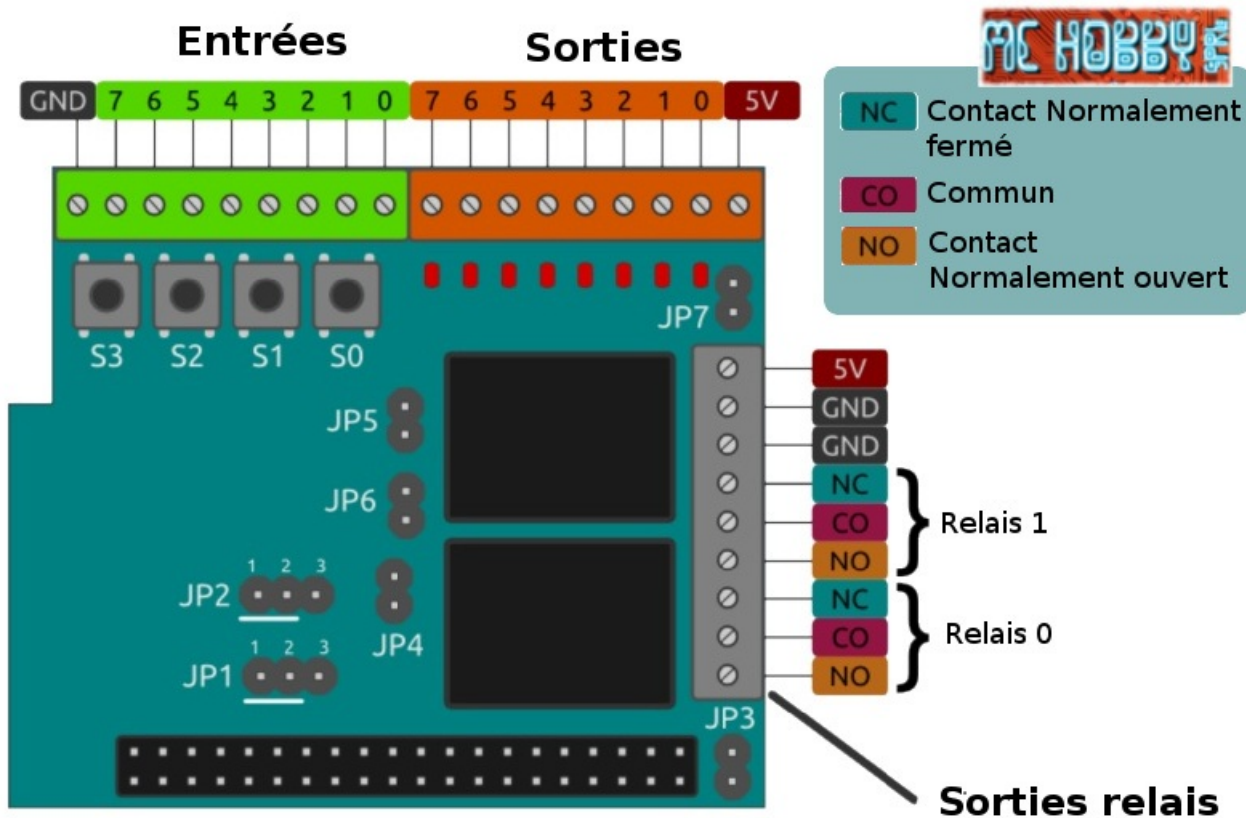
Signaler une erreur

Malgré tout le soin apporté à la réalisation de cette traduction, il n'est pas impossible qu'une erreur se soit malgré tout glissée dans ce document. N'hésitez pas à nous envoyer un e-mail si vous en constatiez une. Pour adresser vos remarques et commentaires relatifs à la traduction, ou pour demander une traduction complémentaire, contactez nous par e-mail à

- support (arobase) mchobby.be.

N'oubliez pas de mentionner le manuel/page/liens en mentionnant l'erreur ;-)

Cavaliers de configuration



- **JP1**: Configuration du bit d'adresse 0 (permet de modifier l'adresse de la carte sur le bus I2C, voir plus loin dans la documentation)
- **JP2**: Configuration du bit d'adresse 1
- **JP3**: Branche le +5V du PiFace sur le +5V du Raspberry-Pi
- **JP4**: Branche la diode de protection (dite en roue libre) sur le +5V (voir plus loin dans la documentation)
- **JP5**: Utilisation du relais 1 (si cavalier en place) sur la sortie 1
- **JP6**: Utilisation du relais 0 (si cavalier en place) sur la sortie 0
- **JP7**: Connecte l'alimentation 5V sur les LEDs et les bobines des relais



Ne branchez pas de source d'alimentation externe sur le PiFace SI le PiFace est alimenté par l'intermédiaire du Raspberry-Pi via le cavalier JP3. Une seule alimentation 5V doit être présente sur le "Pi + PiFace" si JP3 est en place.

Le cavalier JP3

JP3 permet de sélectionner si le PiFace Digital 2 partage la même source d'alimentation que le Raspberry Pi.

Cette alimentation peut:

- soit être fournie par le connecteur microUSB du Raspberry Pi
- soit depuis une alimentation externe fournissant l'alimentation via les bornes +5V et GND du PiFace

Lorsque le cavalier JP3 est en place, le Raspberry Pi et le PiFace partagent la même source d'alimentation.

Lorsque le cavalier JP3 est enlevé, le Raspberry Pi et le PiFace doivent, chacun, leur propre source d'alimentation.

Le cavalier JP7

Le cavalier JP7 connecte la source d'alimentation sur les sorties de la carte (Ex: les relais et leds).

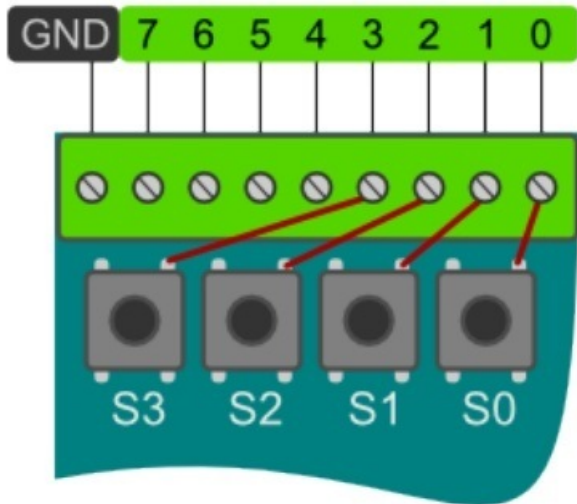
Enlever ce cavalier permet de désactiver les LEDs et les relais de la carte.

Les entrées

Les 8 borniers verts sont des entrées (dit "*inputs*" en anglais) et sont utilisées pour détecter si un switch ou un contact est ouvert ou fermé.

Une entrée est activée lorsqu'elle est connectée sur la masse (GND). Les entrées libellés sous la carte. Le bornier dispose d'une petite surface libre permettant également d'apposer une indication.

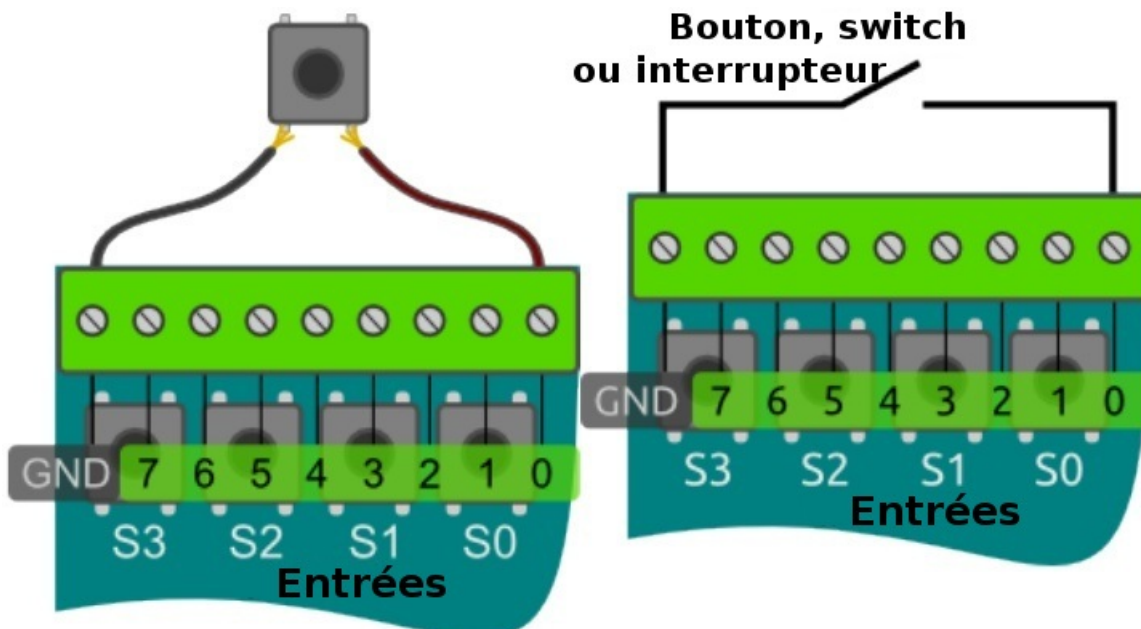
Les quatre bouton poussoirs (dit "*switch*" en anglais) portent les libellés S0 à S3. Ces boutons sont connectés en parallèles sur les quatres premières entrées (de 0 à 3).



Connexion de boutons

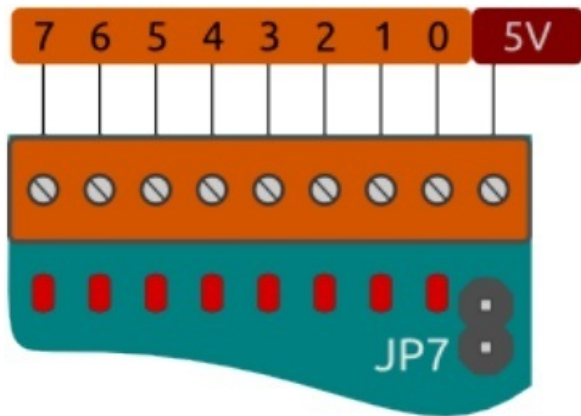
Pour raccorder un bouton extérieur sur le PiFace 2, connectez une entrée du PiFace sur un des côtés du bouton PUIS branchez l'autre côté du bouton sur la borne GND (abréviation de "*ground*" indiquant la masse)..

Contrôlez le fiche technique de votre bouton pour savoir quelles sont les bornes à utiliser.



Les sorties

Les 8 borniers oranges sont des sorties à collecteur-ouvert (voir ci-dessous pour plus de détails) utilisé pour contrôler des composants externes tels que des LEDs, des lumières ou des moteurs. Comme pour les entrées, les bornes portent le libellé sous la carte. Vous pouvez également utiliser la petite surface disponible sur le bornier pour y faire vos propres indications.



Indicateurs LEDs

La carte est équipée de LEDs qui peuvent être utilisées pour indiquer le statut de votre application pendant que celle-ci fonctionne sur votre Raspberry (sans avoir besoin de composant complémentaire).

Ces LEDs sont très utiles pour déboguer vos circuits et code en confirmant l'état attendu d'une sortie.

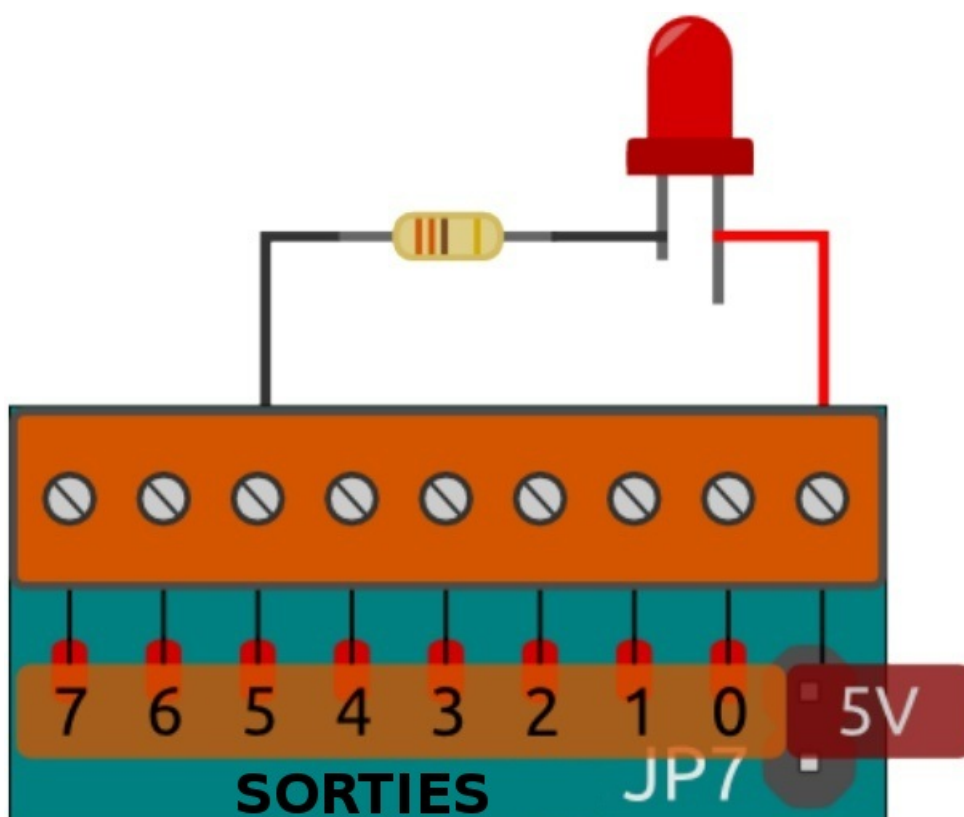
Connexion des sorties

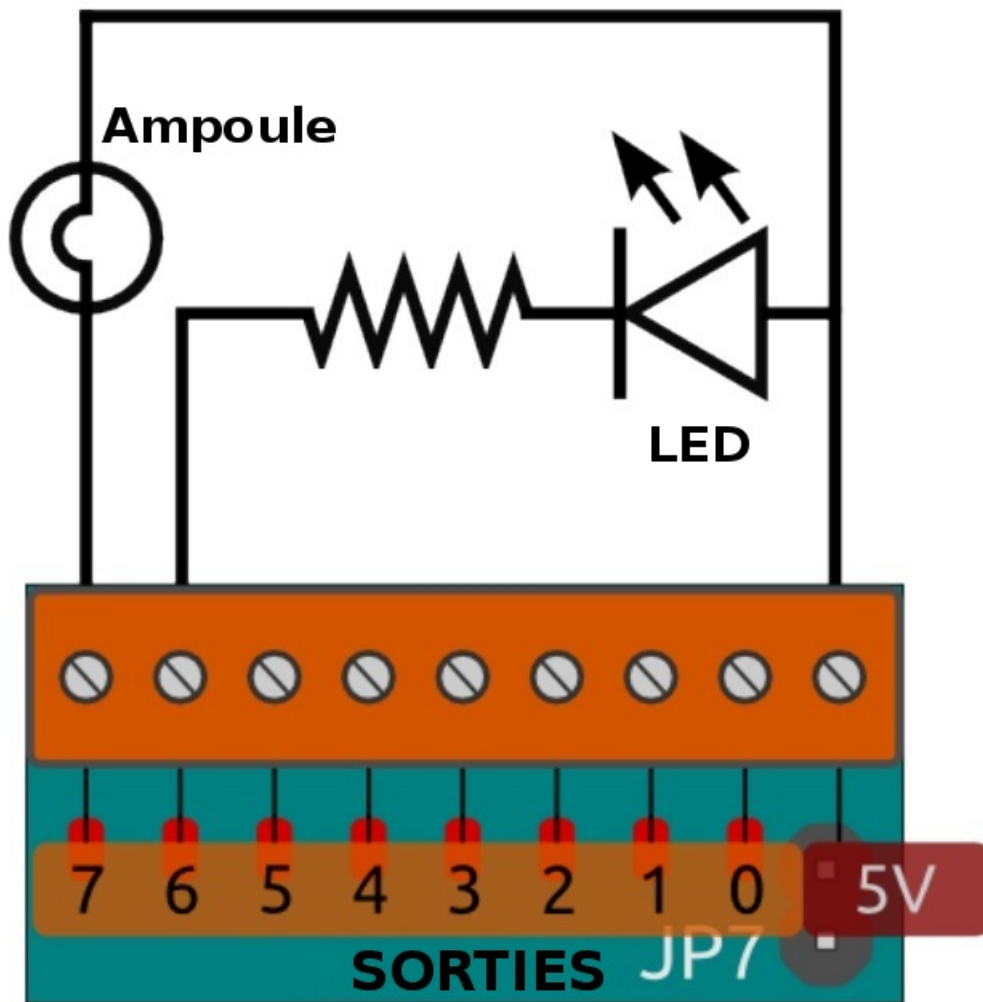
Pour connecter une LED externe, connectez l'anode (la broche "+" de la LED, la plus longue broche) sur la borne +5V.

Connectez ensuite la broche la moins longue ("-") de la LED sur une résistance (généralement 330 ohms pour la plupart des LED for most LEDs).

Finalement connectez l'autre côté de la résistance sur la borne de sortie du PiFace.

Lorsque la sortie du PiFace est activée, cela permet au courant de circuler du +5V vers la masse en passant à travers la LED (ce qui y provoque un effet lumineux).





Sorties à collecteur ouvert

Les 8 sorties du PiFace Digital 2 sont des sorties à collecteur ouvert. Elles peuvent être utilisées pour contrôler des périphériques tels que des ampoules, moteurs ou relais (qui peuvent parfois nécessiter des tensions d'alimentation différents).

Le collecteur-ouvert agit comme "un interrupteur" *qui autorise (ou pas) le retour du courant vers la masse/GND.*

Comme il s'agit de collecteur-ouvert, il n'y a pas de tension de sortie sur la borne, c'est à vous de fournir la tension nécessaire à votre circuit.

Cela offre une plus grande flexibilité étant donné qu'il est possible d'utiliser les sorties sous différentes tensions.



Note: si les sorties doivent être utilisées avec des tensions supérieures à 5V alors il FAUT MODIFIER LA CONFIGURATION DES CAVALIERS pour éviter d'endommager votre carte. Par exemple, déconnectez les cavaliers JP4: diode en roue livre, JP5 & JP6: les relais et JP7: alim 5v des relais.



Si vous utilisez différentes tensions, n'oubliez pas de mettre la masse en commun pour toutes les sources d'alimentation (la fameuse "masse commune").

Les relais

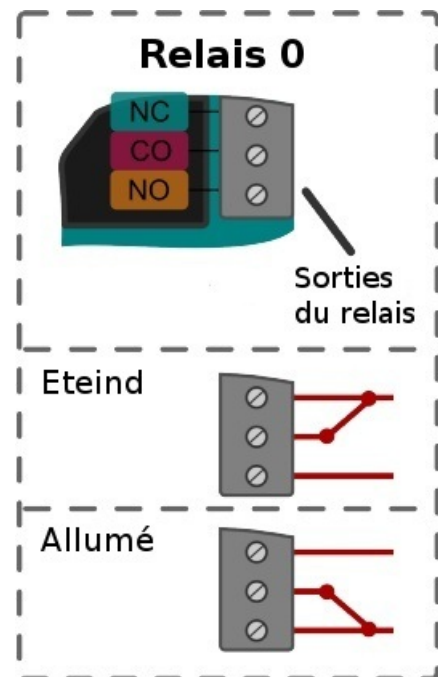
Le PiFace Digital 2 dispose de deux relais 2 directions branchés en parallèle sur les deux premières sorties. Chaque relais est connecté sur un bornier 3 vis. Les fonctions des broches du bornier sont décrites ci-dessous et identifiées par des libellés sous la carte.

- NC - Contact normalement fermé (NC provient de l'anglais **N**ormally **C**losed)
- CO - Est le point commun au contact NC et NO (Co provient de l'anglais **C**ommon)
- NO - Contact normalement ouvert (NO provient de l'anglais **N**ormally **O**pen)

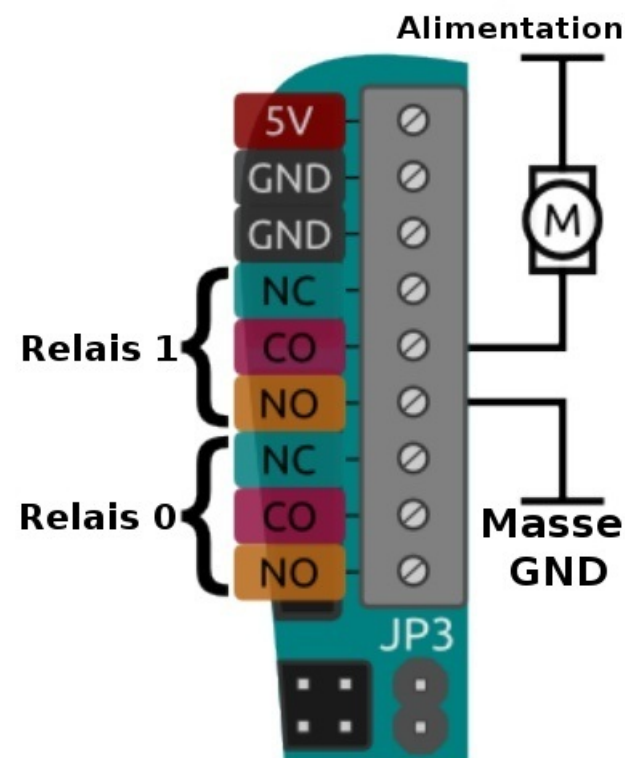
Un contact Normalement Ouvert (entre CO et NO) agit comme un "interrupteur ouvert" aussi longtemps que le relais reste inactif. Dans ce cas de figure, le courant ne peut pas voyager dans le circuit.

Une fois le relais activé, le contact entre CO et NO est fermé pour agir comme un "interrupteur fermé". Le courant peut alors circuler dans le circuit comme le permettrait un interrupteur.

Le contact entre les bornes NC et CO est fermé jusqu'à ce que le relais soit activé, moment où le contact du relais change de position pour connecter NO et CO.



Il est possible de désactiver les relais (avec les cavalier JP5 et JP6).



Cavaliers JP5 et JP6 des relais

Retirez les cavaliers JP5 et JP6 pour respectivement déconnecter les relais 0 et relais 1.

C'est pratique si vous désirez utiliser les sortie 0 et 1 en collecteur-ouvert (ou LED de sortie) sans entendre les cliquetis des relais lorsque les sorties passent de l'état 0 à l'état 1.

Alimentation

Les circuits intégrés du PiFace Digital 2 requièrent une tension d'alimentation de 3.3V (avec un courant tout à fait négligeable). Cette alimentation 3.3V est toujours ponctionnée sur le Raspberry-Pi.

Par contre les relais et les LEDs requièrent une tension de 5V pour fonctionner.

Sur le PiFace Digital 2, cette tension de 5V peut être fournie de différentes façons (voir les détails ci-dessous concernant la configuration des cavaliers):

- Soit avec une tension de 5V indépendante de celle du Raspberry-Pi (Via le bornier d'alimentation)
- Soit depuis l'alimentation 5V du Raspberry-Pi
- Soit via le bornier d'alimentation 5V -ET- alimenter le Pi en 5V depuis la carte PiFace



Ne jamais utiliser deux sources d'alimentation séparées en même temps (une sur le PiFace, l'autre sur le Pi) SI LE CAVALIER D'ALIMENTATION "JP3" EST EN PLACE.



Comme déjà précisé, les ports de sortie du PiFace ne fournissent pas de tension d'alimentation (car il s'agit de collecteur ouvert permettant un retour à la masse)



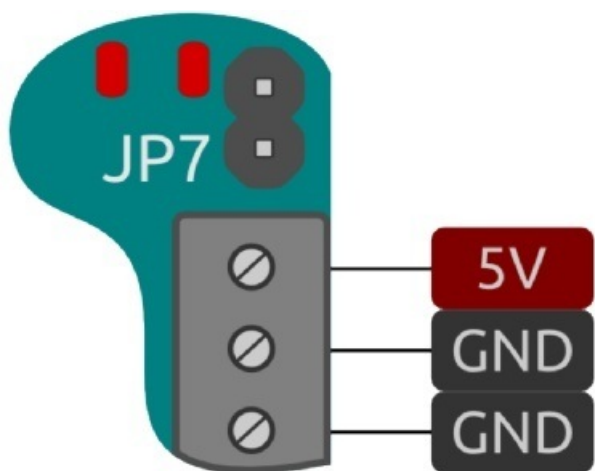
Le relais ne fournit pas de tension car il agit comme un "interrupteur".

L'ensemble des 3 bornes sur la droite de la carte peut être utilisé pour:

1. fournir une alimentation à la carte.
2. récupérer une source d'alimentation 5V pour votre projet (ex: composants).

La borne du haut propose du 5V et les deux bornes en dessous proposent des masses (GND).

Vous n'aurez pas besoin de faire de raccordement sur ces bornes dans la plupart de vos applications CAR le PiFace est alimenté par le Raspberry-Pi (configuration par défaut).



Les adresses

Ou comment utiliser plusieurs PiFace

Cette section concerne l'utilisation de plusieurs cartes PiFace digital 2.

Il est possible de profiter de plus d'entrées et plus de sortie en empilant plusieurs cartes PiFace Digital 2 en utilisant le connecteur approprié (ex: PiFace Rack ou PiFace Shim).

Pour faire la distinction entre les différentes interfaces (carte), il faut que chaque carte reçoive une adresse différente (Voyez la section concernant les cavaliers d'adresse).

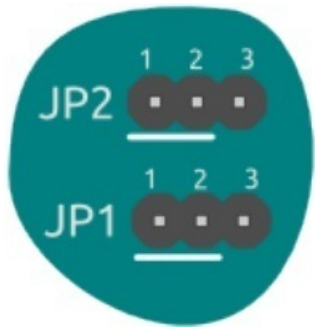
Cavaliers d'adresse

Dans la plupart des cas, les cavaliers peuvent rester en place.

Vous pouvez, cependant, changer l'adresse du PiFace en modifiant la position des cavaliers JP1 et JP2.

En modifiant l'adresse de la carte JP1 et JP2, vous pouvez utiliser plusieurs cartes PiFace en même temps.

L'adresse par défaut de la carte est l'adresse 0. Celle-ci est indiquée à l'aide de lignes blanches dessinées sous chaque cavalier.

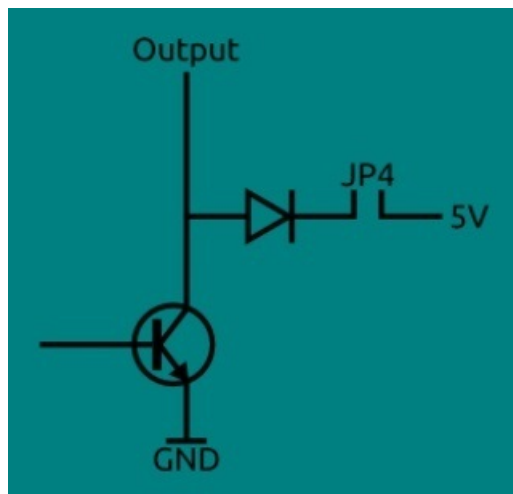


JP1 contrôle le bit moins significatif de l'adresse, et JP2 contrôle le bit le plus significatif.

| Adresse | Adresse binaire | Position JP2 | Position JP1 |
|---------|-----------------|--------------|--------------|
| 0 | 00 | 1-2 | 1-2 |
| 1 | 01 | 1-2 | 2-3 |
| 2 | 10 | 2-3 | 1-2 |
| 3 | 11 | 2-3 | 2-3 |

La diode en roue libre

Le cavalier JP4 branche la diode en roue libre de l'ULN2803A sur le 5V. La diode protège les transistors de l'ULN2803A des surtensions provoquées par la bobine du relais lorsque celle-ci est désactivée.



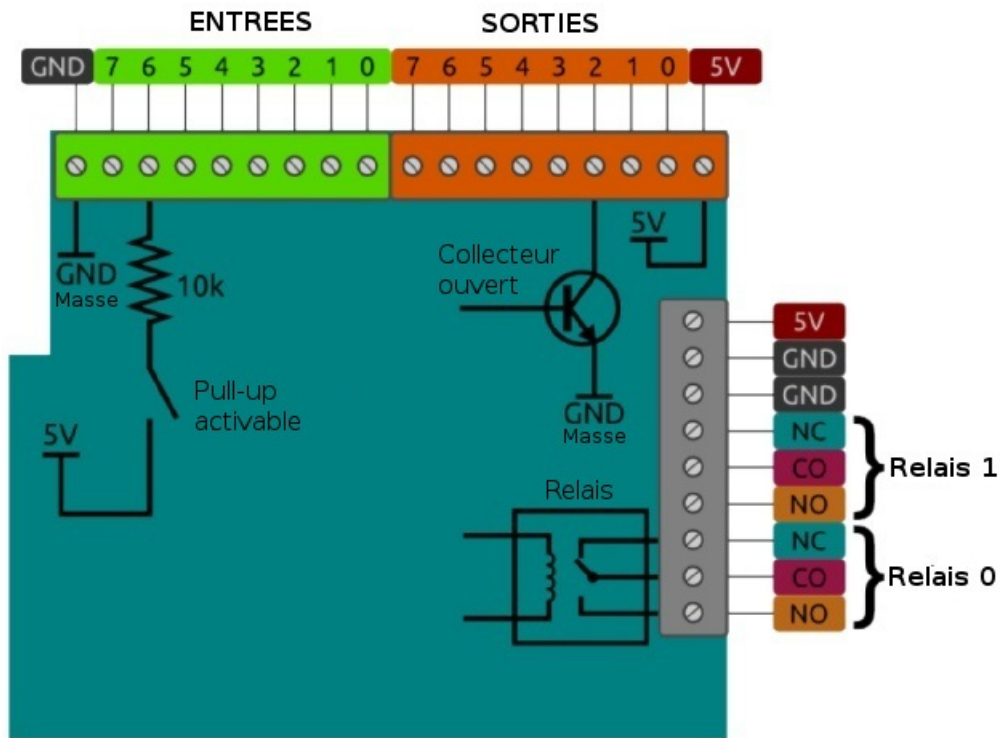
SI une source d'alimentation supérieure a 5 Volts est connectée sur le une sortie à collecteur-ouvert ALORS il faudra enlever le cavalier JP4... et vous ne pourrez plus utiliser les relais

Il est nécessaire d'enlever le cavalier sinon un courant circule entre la source d'alimentation (plus élevée) et le +5V.

Nous avons détaillé ce phénomène sur cette page documentaire réservée à l'ULN2803 <https://wiki.mchobby.be/index.php?title=ULN2803-Sch%C3%A9ma> .

Circuit équivalent

Le circuit équivalent pour les entrées et sorties du PiFace Digital 2 peuvent être vu comme le diagramme ci-dessous.



Installation logiciel

Activer le support SPI

PiFace Digital utilise le bus SPI du Raspberry-Pi pour communiquer avec la carte.

Vous pouvez utiliser l'utilitaire Raspi-Config (depuis une ligne de commande) pour activer le support SPI.

```
sudo raspi-config
```

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 | Choose whether to boot into a des |
| 4 Wait for Network at Boot | Choose whether to wait for networ |
| 5 Internationalisation Options | Set up language and regional sett |
| 6 Enable Camera | Enable this Pi to work with the R |
| 7 Add to Rastrack | Add this Pi to the online Raspber |
| 8 Overclock | Configure overclocking for your P |
| 9 Advanced Options | Configure advanced settings |
| 0 About raspi-config | Information about this configurat |

<Select>

<Finish>

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>

Would you like the SPI interface to be enabled?

<Oui>

<Non>

Redémarrez votre Pi pour activer le support du BUS.


```
sudo reboot
```

Installation logiciel



Si vous avez un distribution récente (avril 2017) alors le support PiFace est déjà installé. Pas besoin de l'installer

Pour installer la partie logicielle du PiFace Digital 2, ouvrez un terminal et exécutez la commande suivante:

```
sudo apt-get install python3-pifacedigitalio
```

Pour installer l' **Emulateur logiciel du PiFace Digital 2**, ouvrez un terminal et exécutez la commande suivante::

```
sudo apt-get install python3-pifacedigital-emulator
```

Si vous voulez utiliser **Scratch avec votre PiFace Digital 2**, il faut installer un logiciel gestionnaire (dit "*handler*") qui interceptera les messages destinés au PiFace et qui activera les fonctionnalités de la carte PiFace:

```
sudo apt-get install python3-pifacedigital-scratch-handler
```

Pour finir, il faudra redémarrer votre Raspberry-Pi. Saisissez la commande suivante dans un terminal:

```
sudo reboot
```

Tester

Sommaire

- 1 Démarrer l'émulateur
- 2 Sélectionner la version de la carte
- 3 Les sorties
- 4 Les entrées

Démarrer l'émulateur

Après avoir terminé l'installation logiciel et redémarré votre Pi, vous pourrez démarré l'émulateur PiFace depuis une terminal à l'aide de la commande suivante:

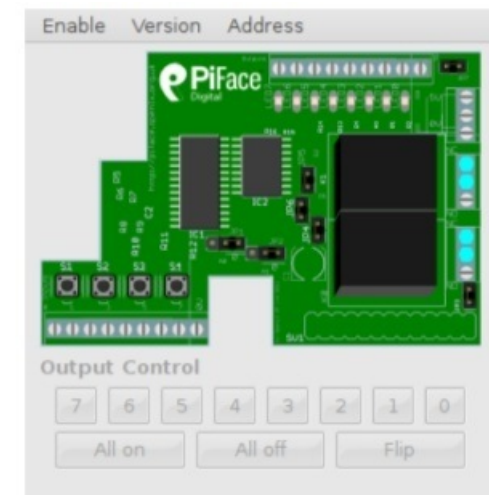
```
pi facedigital-emulator
```

Sélectionner la version de la carte

La première chose à faire est de sélectionner la version du PiFace utilisé par l'intermédiaire du menu "Version".



La carte PiFace existe sous deux versions: le format original et la version digital 2



Original PiFace™ Digital

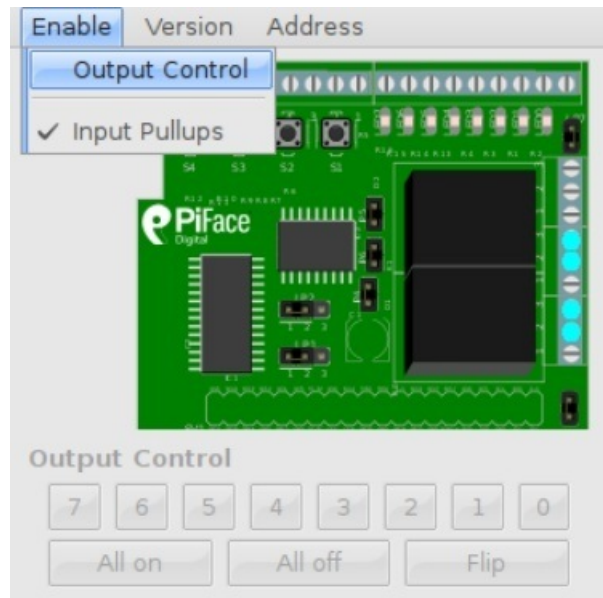


PiFace™ Digital 2

Les sorties

Pour pouvoir contrôler manuellement une sortie, il faut activer le contrôle des sortie .

Ouvrez le menu "Enable" puis activez l'option "Output Control" (l'entrée doit être précédé d'une coche).

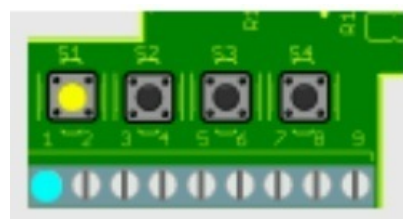


Changer l'état de la broche 0 en cliquant sur le bouton '0'. Le premier relais du PiFace cliqueter lorsqu'il s'active et la LED correspondante s'allumer (si les cavaliers restent dans leur configuration d'origine). Le graphique à l'écran est également rafraîchit pour afficher le nouvel état de la LED, le contact du relais et l'activation la première sortie.

Les LEDs sont en raccordées en parallèle avec le bornier de sortie, elles indiquent l'état des différentes sorties (vous savez donc lorsqu'elles sont activées).

Les entrées

Pressez sur un des bouton du PiFace. Voyez le changement de représentation à l'écran indiquant l'état de l'entrée (et la borne correspondante).



Python: Premiers pas

Sommaire

- 1 Premiers pas avec Python
- 2 Les pas suivants
 - 2.1 Sorties
 - 2.2 Entrées
 - 2.3 input_port et output_port
- 3 Exemples python
 - 3.1 Activer un relais
 - 3.2 Clignoter une LED
 - 3.3 Lire une entrée

Premiers pas avec Python

PiFace Digital 2 peut être contrôlé facilement en utilisant Python. Pour commencer, ouvrez un terminal (ligne de commande) et démarrez un interpréteur en saisissant la commande suivante:

```
python3
```



Vous pouvez également utiliser l'éditeur "Idle" disponible dans l'environnement graphique

Pour utiliser le PiFace Digital 2 depuis Python, il faut importer le module `pifacedigitalio` :

```
import pifacedigitalio
```

Avant d'utiliser la carte PiFace, il faut initialiser la bibliothèque en faisant l'appel suivant:

```
pifacedigitalio.init()
```

Si vous désirez savoir si un bouton est enfoncé ou non, il faut utiliser la fonction `digital_read` :

```
pifacedigitalio.digital_read( numero_de_broche )
```

La fonction retourne `1` si l'entrée indiquée dans `numero_de_broche` si le bouton est enfoncé ou l'entrée est connecté sur la masse (GND). Sinon la fonction retourne `0`.

Si vous désirez contrôler une sortie, il faut utiliser la fonction `digital_write` :

```
pifacedigitalio.digital_write( numero_de_broche, etat )
```

La fonction permet de fixer l'état de la broche `numero_de_broche` sur l'état `0` (inactif) ou `1` (actif). L'état `1` active le collecteur ouvert (qui laisse passer le courant) et la LED correspondante.

Les pas suivants

Une façon plus puissante et plus explicite de contrôler le PiFace est de passer par un objet `PiFaceDigital`.

En créant un objet de type `PiFaceDigital`, vous aurez accès à toutes les fonctionnalités du PiFace (incluant l'activation des pull-ups, l'input_port et output_port).

Démarrer un interpréteur python à l'aide de la commande:

```
python3
```

Instanciez ensuite un objet `PiFaceDigital` à l'aide de l'instruction suivante:

```
import pifacedigitalio
pfd = pifacedigitalio.PiFaceDigital()
```

Sorties

Pour fixer la première sortie (celle qui porte le numéro 0) sur l'état 1 (pour activer la sortie):

```
pfd.output_pins[0].value = 1
```

Notez le petit cliquetis du relais lorsqu'il s'active. La broche de sortie 0 contrôle également le premier relais.

Vous pouvez également désactiver la sortie numéro 0 (et le relais) en fixant la valeur de la sortie à 0:

```
pfd.output_pins[0].value = 0
```

Entrées

Vous pouvez lire la valeur d'une entrée à l'aide de l'instruction suivante:

```
pfd.input_pins[1].value
```

Essayez d'exécuter l'instruction ci-dessus en pressant le second bouton (qui porte le numéro 1) PUIS exécutée une seconde fois sans presser le bouton.

Vous devriez voir deux résultats différents.

input_port et output_port

Vous pouvez lire toutes les entrées en une seule opération en utilisant l'attribut `input_port` :

```
pfd.input_port.value
```

Input_port retourne une valeur numérique qui, une fois exprimée en valeur binaire, indique l'état de chaque entrée.

Il est également possible de contrôler toutes les sorties en une opération en manipulant l'attribut `output_port` :

```
pfd.output_port.value = 0xAA
```

Exemples python

Activer un relais

Les relais du PiFace Digital sont respectivement connecté sur les deux premières entrées 0 et 1. Vous pouvez donc contrôler ces deux relais à l'aide de la fonction `digital_write()`.

Start a new Python interpreter and type the following:

```
import piface.pfio as pfio
pfio.init()
# mettre la sortie 0 a l etat 1 (activer)
pfio.digital_write(0,1)
```

Clignoter une LED

Un programme qui fait clignoter une LED doit l'allumer et l'éteindre de façon répétitive (avec un délai entre chaque activation/désactivation).

```
from time import sleep
import piface.pfio as pfio
pfio.init()
while(True):
    # activer la sortie 0 = allumer la LED
    pfio.digital_write(0,1)
    # Attendre une seconde
    sleep(1)
    # désactiver la sortie 0 = éteindre la LED
    pfio.digital_write(0,0)
    sleep(1)
```

Lire une entrée

Il faut utiliser la fonction `pio.digital_read(pin)` pour lire l'état d'une entrée. La fonction retourne *1* si le bouton est pressé (ou entrée activée) sinon elle retourne *0*.

Démarrez un nouvel interpréteur Python et saisissez les instructions suivantes:

```
import piface.pfio as pfio
pfio.init()
# lecture de l'entree 0 (correspond au premier bouton)
pfio.digital_read(0)
```

Python affiche *0*.

Maintenez le premier switch enfoncé (celui portant le libellé *S0*) et saisissez une nouvelle fois la commande `pio.digital_read(0)`.

Cette fois Python defait afficher *1*.

Entrée tactile

Sommaire

- 1 Créer une entrée tactile
- 2 Désactiver les pull-ups
 - 2.1 En utilisant l'émulateur
 - 2.2 En utilisant python
- 3 Tester l'entrée "tactile"

Créer une entrée tactile

Vous avez envie d'aller au delà de l'utilisation des interrupteur et des fils? Vous pouvez transformer les entrées du PiFace Digital 2 en entrée tactile (qu'il suffit de toucher), ce qui permet de transformer n'importe quel matériaux conducteur en "interrupteur magique" :-)

Il est donc possible d'utiliser des bananes, des gens, etc pour réaliser des interrupteurs innovants comme le fait le Makey-Makey https://shop.mchobby.be/product.php?id_product=273 .

Pour utiliser les entrées en mode tactile, il faut configurer le PiFace pour qu'il puisse détecter la circulation de petites quantités de courant qui traverseront les objets.

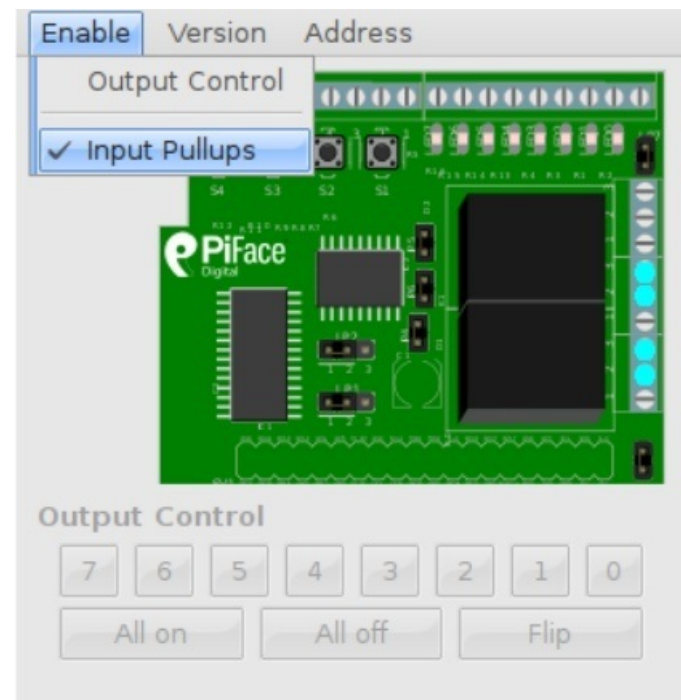
Sur chaque entrée, il faudra connecter une résistance de 25 à 40M ohm (Mega Ohms) entre la broche d'entrée et 5V.

Désactiver les pull-ups

Il faudra ensuite désactiver la résistance pull-up interne interne, ce qui peut être réalisé de deux façons différentes:

En utilisant l'émulateur

Si vous utilisez l'émulateur, sélectionner le menu "Enable" et désactiver l'option "Input Pullups" (elle ne doit plus être cochée).



En utilisant python

Si vous utilisez un script Python, vous pouvez désactiver les pullups de chaque broche avec la commande suivante:

```
digital_write_pullup( numero_de_broche, etat )
```

Où "etat" est fixé à 0 pour désactiver la résistance pull-up sur la broche d'entrée numero_de_broche.

Tester l'entrée "tactile"

Pour tester l'entrée:

- Connecter un fil sur l'entrée tactile, fil que l'on raccorde sur une banane (par exemple) ou que l'on laisse libre.
- Connecter un autre fil sur la masse (GND), fil dont vous garderez l'autre bout en main.

Maintenant, en gardant toujours le fils de masse dans votre première main, vous touchez le fil de l'entrée (ou la banane). Voilà, vous venez de fermer le circuit électrique et PiFace doit détecter l'activation de l'entrée.

Vous pouvez voir ce changement d'état en utilisant l'émulateur ou en effectuant des `digital_read` avec Python.

PiFace et Scratch

Il est possible d'utiliser PiFace avec Scratch, nous avons déjà écrit un tutoriel à ce sujet pour le PiFace 1... c'est donc l'endroit où vous pouvez poursuivre votre lecture

- Scratch: introduction
- Activer "Host Mesh"
- Démarrer Scratch et PiFace
- Entrée/sortie avec Scratch

Traduction basée sur le document Technical DataSheet http://www.farnell.com/datasheets/1881551.pdf?_ga=2.101471942.67776673.1494836467-728456141.1430074529 produit par Farnell

Toute référence, mention ou extrait de cette traduction doit être explicitement accompagné du texte suivant : « Traduction par MCHobby (www.MCHobby.be <http://www.MCHobby.be>) - Vente de kit et composants » avec un lien vers la source (donc cette page) et ce quelque soit le média utilisé.

L'utilisation commerciale de la traduction (texte) et/ou réalisation, même partielle, pourrait être soumise à redevance. Dans tous les cas de figures, vous devez également obtenir l'accord du(des) détenteur initial des droits. Celui de MC Hobby s'arrêtant au travail de traduction proprement dit.