



# Contrôlez votre carte Arduino avec un Raspberry Pi et Python

***La puissance du Raspberry et la simplicité de l'Arduino grâce à Python et une simple bibliothèque : Nanpy.***

## Introduction

Une carte Arduino peut communiquer avec le Raspberry Pi via une liaison par connexion USB. Une interface série virtuelle est créée et utilisée comme une interface normale, avec des lectures/écritures vers le fichier de périphérique série. Pour commencer, connectez votre carte Arduino et tapez :

```
$dmesg | tail
[..]usb 1-1.2: Manufacturer: Arduino[..]
[..]cdc_acm 1-1.2:1.0: ttyACM0: USB ACM
device[..]
```

Mon périphérique de carte Arduino Uno est /dev/tty/ACM0 et son pilote est cdc\_acm. Les anciennes cartes avec un circuit USB-série sont accessibles grâce à /dev/ttyUSB\* :

```
$ls -l /dev/ttyACM*
crw-rw---T 1 root dialout 166, 0 Nov 5
00:09 /dev/ttyACM0
```

Il faut maintenant ajouter l'utilisateur au groupe "dialout" pour lui donner les droits d'accès en lecture/écriture, puis fermer et rouvrir la session pour que la modification prenne effet :

```
$sudo usermod -a -G dialout UTILISATEUR
```

C'est important car Nanpy travaille par l'intermédiaire du fichier de périphérique. Nanpy est un projet open source distribué sous les termes de la licence MIT, et se compose d'une partie serveur (enregistrée sur l'Arduino pour attendre les commandes sur le port série) et d'une bibliothèque 100% Python. Celle-ci permet de communiquer avec l'Arduino branché en USB en utilisant des classes et des méthodes très similaires à celles du framework Arduino. En arrière-plan, quand vous créez/supprimez un objet ou appelez des méthodes avec Python, Nanpy communique via l'USB avec la partie serveur et lui demande de créer/supprimer l'objet correspondant ou d'appeler les méthodes sur l'Arduino : vous pouvez instancier autant d'objets que vous le souhaitez sans vous

préoccuper de désallocation et il est également possible de travailler dans un contexte multithreadé. Nanpy a pour objectif de simplifier la vie des développeurs en leur apportant des outils simples, clairs et rapides pour créer des prototypes et des scripts qui interagissent avec l'Arduino, économisant ainsi beaucoup de temps. Pour installer Nanpy, lisez le fichier README. Il faut installer Arduino sur votre portable ou votre Raspberry Pi afin de construire le micrologiciel :

```
$sudo apt-get install arduino
```

Nanpy est actuellement en plein développement et il n'a été testé que sur cartes Uno. Vous pouvez le récupérer depuis la page Pypi (<http://pypi.python.org/pypi/nanpy>) ou Github (<https://github.com/nanpy>).

Voyons maintenant Nanpy à l'œuvre et essayons d'allumer une DEL située sur la 13<sup>e</sup> broche de l'Arduino :

```
from nanpy import Arduino
Arduino.pinMode(13, Arduino.OUTPUT)
Arduino.digitalWrite(13, Arduino.HIGH)
```

Arduino fournit toutes les principales fonctions, temporisations, lecture et écriture analogique/numérique. Aucune fonction de configuration ou de boucle, uniquement des objets et des appels de méthodes. En réalité, Nanpy prend en charge toutes les principales méthodes Arduino - LCD, Tone, Stepper et autres bibliothèques. Voyons à présent comment utiliser notre écran ACL en mode texte 16x2 sur les broches 6, 7, 8, 9, 10, 11 en écrivant un script Bonjour tout le monde :

```
from nanpy import Lcd
lcd = Lcd([7, 8, 9, 10, 11, 12],[16, 2])
lcd.printString("Bonjour tout le monde
!")
```

**Juste un mot d'avertissement :** Le Raspberry Pi peut ne pas fournir assez de puissance pour alimenter votre Arduino, vous aurez donc peut-être besoin de le brancher sur une source d'alimentation externe.