

TP SCI N° 05:

Communication Arduino-Raspberry pi

(Partie théorique)

Réalisé par : Anfal Bourouina

Binôme : 05

Description du Nano-computer Raspberry.

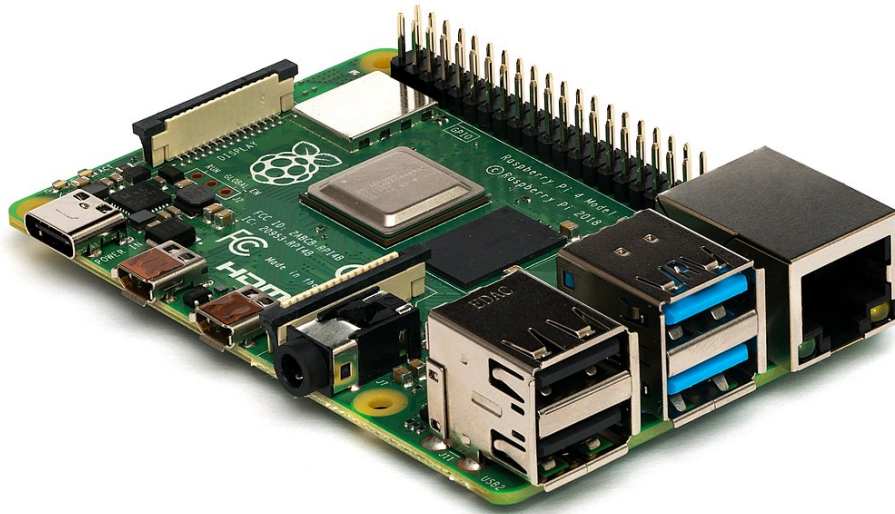


Figure 01: Carte Raspberry pi 4

Raspberry Pi est un petit ordinateur monocarte. En connectant des périphériques tels que le clavier, la souris, l'écran au Raspberry Pi, il agit comme un mini ordinateur personnel. Raspberry Pi est souvent utilisé pour le traitement d'images et de vidéos en temps réel, les applications IoT et les applications robotiques.

Quelques caractéristiques de Raspberry pi

- Raspberry Pi est plus lent qu'un ordinateur portable ou de bureau, mais reste un ordinateur capable de fournir toutes les fonctionnalités ou capacités attendues, avec une faible consommation d'énergie.
- La Fondation Raspberry Pi fournit officiellement le système d'exploitation Raspbian basé sur Debian. Elle fournit également le système d'exploitation NOOBS pour Raspberry Pi. Nous pouvons installer plusieurs versions tierces du système d'exploitation comme Ubuntu, Archlinux, RISC OS, Windows 10 IOT Core, etc.
- Raspbian OS est le système d'exploitation officiel disponible gratuitement. Ce système d'exploitation est efficacement optimisé pour être utilisé avec Raspberry Pi. Raspbian a une interface graphique qui comprend des outils pour la navigation, la programmation Python, la bureautique, les jeux, etc.
- Nous devrions utiliser une carte SD (minimum 8 GB recommandé) pour stocker le OS (système d'exploitation).
- Raspberry Pi est plus qu'un ordinateur car il permet d'accéder au matériel intégré, c'est-à-dire aux GPIO, pour développer une application. En accédant aux GPIO, nous pouvons connecter des appareils tels que des LED, des moteurs, des capteurs, etc. et les contrôler également.
- Il est doté d'un processeur Broadcom SoC basé sur ARM et d'un GPU (Graphics Processing Unit) intégré.
- La vitesse du processeur du Raspberry Pi varie de 700 MHz à 1,2 GHz. Il dispose également d'une SDRAM embarquée allant de 256 Mo à 1 Go.

- Raspberry Pi dispose également de modules SPI, I2C, I2S et UART intégrés.

Étude des GPIO et les possibilités de communication de Raspberry

La carte Raspberry Pi 4 possède un connecteur GPIO de 40 broches. Les GPIO nous permettent d'utiliser facilement les fonctions matérielles et la communication, directement depuis un ordinateur - le microprocesseur du Raspberry Pi. Cela rapproche le Raspberry Pi 4 des applications matérielles, le rendant parfait pour être intégré dans une application ou un produit matériel : un robot, une application de rétro-gaming, etc.

Voici un aperçu complet de tous les GPIO et de leur fonction principale.

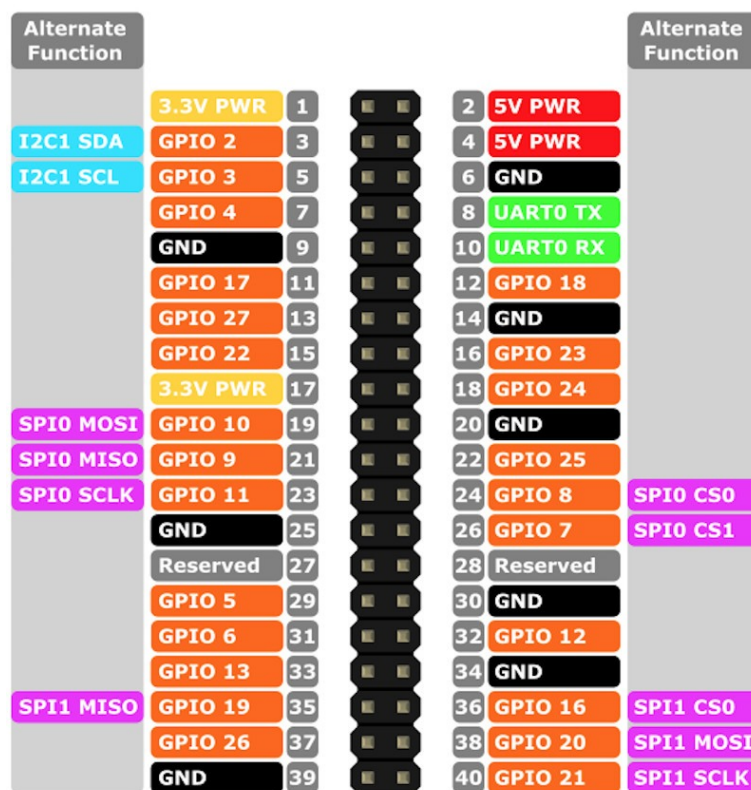


Figure 02: Détail des broches Raspberry pi

Avant de brancher quoi que ce soit sur une broche du Raspberry Pi 4, vous devez savoir que vous pouvez facilement endommager la carte si vous faites quelque chose de mal. Il n'existe pas de véritable sécurité matérielle en ce qui concerne les broches matérielles du Raspberry Pi.

Broches de masse (GND)

La broche de masse est très utile pour établir une référence commune entre tous les composants du circuit. 8 des 40 GPIOs sont connectés à la masse. On peut les trouver avec les 3 lettres GND.

Broches d'alimentation

On peut trouver 2 broches apportant 3.3V et 2 broches apportant 5V. Ces broches

peuvent être utilisées pour alimenter des composants tels que des capteurs ou de petits actionneurs. Notez qu'elles ne sont certainement pas assez puissantes pour actionner des moteurs tels que des servomoteurs ou des moteurs pas à pas. Pour cela, vous aurez besoin d'une source d'alimentation externe.

Les broches d'alimentation sont utilisées comme source d'alimentation des composants externes, et non pour alimenter le Raspberry Pi lui-même à partir d'une source externe. (Il existe un moyen d'alimenter le Raspberry Pi à partir de l'en-tête GPIO, mais vous avez une forte probabilité de le brûler, alors utilisez simplement le port micro-USB).

Broches réservées

Les broches 27 et 28 sont des broches réservées. Elles sont généralement utilisées pour la communication I2C avec une EEPROM.



Broches de communication

Jusqu'à maintenant on compte 14 broches réservées pour le GND, l'alimentation et les broches spéciales. Voyons maintenant comment les 26 autres GPIO sont utilisées pour la communication.

Les GPIO Raspberry 4

Les GPIO nous permettront de lire certains capteurs de base (ex : infrarouge), de contrôler certains actionneurs (ceux qui fonctionnent avec un mode ON/OFF), et de communiquer avec d'autres cartes matérielles, telles que Raspberry Pi, Arduino, Beaglebone, Jetson Nano, etc.

Les GPIO du Raspberry Pi 4 sont assez similaires aux "broches numériques" sur une carte Arduino. (on écrit soit HIGH soit LOW).

Tension des GPIOs

Tous les GPIOs fonctionnent à 3.3V. Il est important de le savoir, au cas où nous aurions besoin de brancher un composant ayant une tension différente.

Parfois, on trouve des capteurs alimentés en 5V, mais toutes les broches de communication fonctionnent en 3,3V. Dans ce cas, pas de problème : on peut utiliser la broche d'alimentation 5V du Raspberry Pi pour alimenter le composant, puis utiliser n'importe quelle GPIO 3,3V pour la communication. Si on ne mélange pas le signal 5V avec les signaux 3.3V, tout devrait bien se passer.

Si on a besoin de faire communiquer les GPIO de Raspberry Pi 4 avec des broches 5V directement (ex : Arduino Uno ou Mega), nous aurons besoin d'utiliser un décalage de niveau de 3,3V à 5V.

Utilisation des GPIO

Pour utiliser un GPIO, nous devons d'abord connaître son numéro.

Les numéros de broches et les numéros de GPIO sont différents. Les numéros de broches sont en gris, et les numéros de GPIO en orange. Selon la bibliothèque qu'on utilise pour manipuler les GPIO, on doit utiliser soit le numéro de la broche, soit le numéro du GPIO.

Il existe au moins 2 bibliothèques qui nous permettront d'utiliser facilement ces broches. Pour Python, nous pouvons utiliser *RPi.GPIO*, et pour Cpp nous pouvons utiliser *WiringPi*. Ces bibliothèques ont été développées pour que nous puissions utiliser les broches de

Raspberry Pi comme avec des broches Arduino.

Protocoles de communication

On peut utiliser certains protocoles de communication matériels directement avec les GPIO du Raspberry Pi 4. Ces protocoles de communication sont en fait les mêmes que ceux que vous pouvez utiliser nativement sur de nombreuses cartes Arduino.

Sur les schémas de brochage du Raspberry Pi 4, on peut voir une colonne pour les fonctions alternatives, qui inclut les protocoles de communication aussi.

UART

UART est un protocole de communication multi-maître. Ce protocole est assez facile à utiliser et très pratique pour communiquer entre plusieurs cartes : Raspberry Pi vers Raspberry Pi, ou Raspberry Pi vers Arduino, etc.



Pour utiliser UART nous avons besoin de 3 broches :

- GND qu'on connecte à la GND globale du circuit.
- RX pour Réception. On connecte cette broche à la broche TX de l'autre composant.
- TX pour Transmission. On connecte cette broche à la broche RX de l'autre composant.

Si le composant avec lequel on communique n'est pas déjà alimenté, on devra également utiliser une broche d'alimentation (3,3 V ou 5V) pour mettre ce composant sous tension.

En utilisant un convertisseur UART vers USB, on peut communiquer entre l'ordinateur portable et Raspberry Pi avec UART.

Maintenant, pour utiliser UART dans le code, on peut utiliser la bibliothèque *Serial* en Python, et *WiringPi* en Cpp.

I2C

I2C est un protocole de bus maître-esclave (il peut avoir plusieurs maîtres, mais on l'utilisera surtout avec un maître et plusieurs esclaves). L'utilisation la plus courante de l'I2C est de lire les données de capteurs et d'actionner certains composants.

Le maître est le Raspberry Pi, et les esclaves sont tous connectés au même bus. Chaque esclave possède un identifiant unique, de sorte que le Raspberry Pi sait à quel composant il doit s'adresser.



Pour utiliser I2C, nous aurions besoin de 3 broches :

- GND.
- SCL : horloge de l'I2C. On connecte tous les esclaves SCL au bus SCL.
- SDA : données échangées. On connecte tous les esclaves SDA au bus SDA.

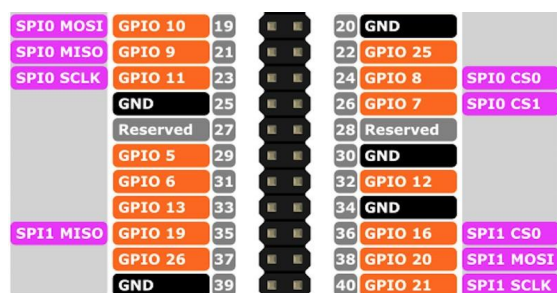
Les broches SDA et SCL du Raspberry Pi sont des fonctions alternatives aux GPIO 2 et 3. Lorsque on utilise une bibliothèque (Python,

Cpp, etc.) pour I2C, ces deux GPIO seront configurées pour pouvoir utiliser leur fonction alternative.

Certaines des meilleures bibliothèques faciles à utiliser pour I2C sont *SMBus* pour Python et *WiringPi* pour Cpp.

SPI

SPI est encore un autre protocole de communication matérielle. C'est un protocole de bus maître-esclave. Il nécessite plus de fils que l'I2C, mais peut être configuré pour fonctionner plus rapidement.



Il existe 2 SPIs par défaut : SPI0 et SPI1. Cela signifie qu'on peut utiliser le Raspberry Pi

comme maître SPI sur deux bus SPI différents en même temps.

Et, comme pour I2C, SPI utilise les fonctions alternatives des GPIOs.

Pour utiliser SPI, vous aurez besoin de 5 broches :

- GND .
- SCLK : horloge du SPI. On connecte toutes les broches SCLK ensemble.
- MOSI : signifie Master Out Slave In. C'est la broche pour envoyer des données du maître à un esclave.
- MISO : signifie Master In Slave Out. Il s'agit de la broche permettant de recevoir des données d'un esclave vers le maître.
- CS : signifie Chip Select. Nous aurons besoin d'un CS par esclave sur notre circuit. Par défaut, on a deux broches CS (CS0 - GPIO 8 et CS1 - GPIO 7). On peut configurer plus de broches CS à partir des autres GPIOs disponibles.

Dans le code, on peut utiliser la bibliothèque *spidev* pour Python, et *WiringPi* pour Cpp.

Description du BLE (Bluetooth Low Energy) pour Arduino

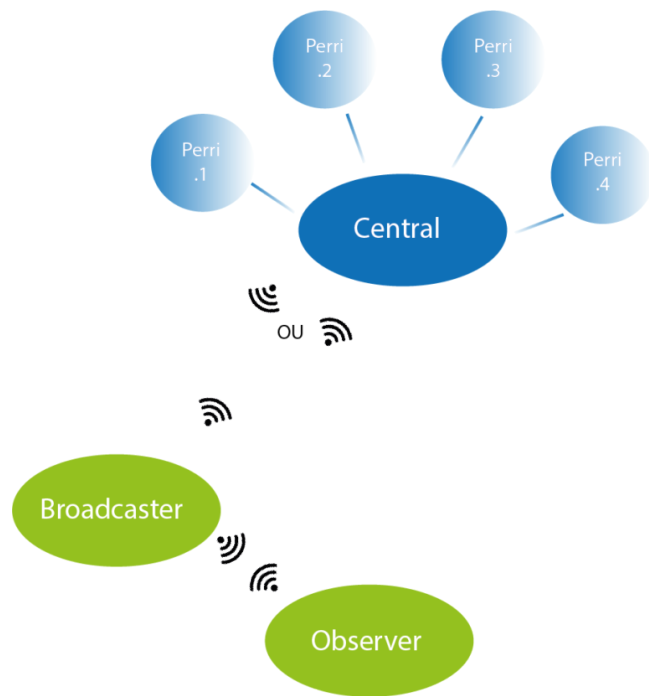
Bluetooth, un module de communication sans fil simple, très populaire depuis quelques décennies et facile à utiliser, est utilisé dans la plupart des appareils alimentés par batterie. Au fil des ans, la norme Bluetooth a fait l'objet de nombreuses mises à niveau pour répondre à la demande des clients et de la technologie en fonction des besoins du moment et de la situation. Tout a commencé avec la version 1.0 de Bluetooth et actuellement, la version 5.0 de Bluetooth est disponible sur le marché. Au cours de ces dernières années, de nombreuses choses ont changé, notamment le taux de transmission des données, la consommation d'énergie avec les appareils vestimentaires, les appareils IoT et les systèmes de sécurité.

BLE

A partir de la version 4.0 , Bluetooth 4.0 comprend à la fois le Bluetooth traditionnel,

désormais appelé *Bluetooth Classic*, et le Bluetooth Low Energy. Le Bluetooth Low Energy est optimisé pour une faible consommation d'énergie à de faibles débits de données, et a été conçu pour fonctionner avec de simples piles bouton au lithium.

Fonctionnement du BLE



Le BLE dispose de différents types de connexions. N'importe quel objet connecté en BLE peut avoir jusqu'à 4 fonctions différentes :

Broadcaster : il peut être utilisé comme serveur. Son objectif est de transmettre régulièrement des données à un autre appareil. Seul point négatif, il n'accepte aucune connexion entrante.

Observer : Ici, l'objet écoute et interprète les données envoyées par un « broadcaster ». L'objet ne peut pas envoyer de connexions vers le serveur.

Central : souvent un smartphone ou une tablette. Il peut réagir de deux façons différentes : soit en mode advertising, soit en mode connecté. Dans ce cas, il est le dirigeant et c'est de lui que part l'échange de données.

Peripheral : Va accepter toutes les connexions du central et lui envoyer des données de manière périodique. Ce système package les données de façon universelle via le protocole afin qu'elles soient comprises par les autres périphériques.

Caractéristiques du BLE

1. Faible consommation d'énergie:

Pour réduire la consommation d'énergie, un appareil BLE est maintenu en mode veille la plupart du temps. Lorsqu'un événement se produit, l'appareil se réveille et un court message est transféré à une passerelle, un PC ou un smartphone. La consommation électrique maximale/crête est inférieure à 15 mA et la consommation électrique moyenne est d'environ 1 μ A. La consommation d'énergie active est réduite à un dixième de la consommation d'énergie du Bluetooth classique. Dans les applications à faible cycle d'utilisation, une pile bouton pourrait assurer 5 à 10 ans de fonctionnement fiable.

2. Robustesse, sécurité et fiabilité:

La technologie BLE utilise la même technologie de saut de fréquence adaptatif (AFH) que la technologie Bluetooth classique. Cela permet à BLE de réaliser une transmission robuste dans les environnements RF bruyants que l'on trouve dans les applications domestiques, industrielles et médicales. Pour minimiser le coût et la consommation d'énergie de l'utilisation de la technologie AFH, la technologie BLE a réduit le nombre de canaux à 40 canaux de 2 MHz de large au lieu des 79 canaux de 1 MHz de large utilisés par la technologie Bluetooth classique.

3. Coexistence sans fil:

Les technologies Bluetooth, LAN sans fil, IEEE 802.15.4/ZigBee et plusieurs radios propriétaires utilisent la bande ISM (Industrial Scientific Medical) de 2,4 GHz sans licence.

Avec autant de technologies partageant le même espace radio, l'interférence peut diminuer les performances sans fil (c'est-à-dire augmenter la latence et diminuer le débit) en raison de la nécessité de corriger les erreurs et de retransmettre. Dans les applications exigeantes, il est possible de réduire les interférences en planifiant les fréquences et en concevant des antennes spéciales. Comme la technologie Bluetooth classique et la technologie BLE utilisent toutes deux l'AFH, qui minimise les interférences avec les autres technologies radio, la transmission Bluetooth est robuste et fiable.

4. Portée de la connexion:

La technologie BLE a une modulation légèrement différente de la technologie Bluetooth classique. Cette différenciation de modulation offre une portée allant jusqu'à 300 mètres avec un chipset radio de 10 dBm (maximum BLE).

BLE pour Arduino

Il existe plusieurs modules pour fournir les fonctionnalités du BLE pour Arduino, notamment le module HM-10. Le HM-10 est un petit module BLE Bluetooth 4.0 SMD 3.3v basé sur le SOC (System On Chip) Bluetooth. Il existe 2 versions du HM-10 : le HM-10C et le HM-10S.

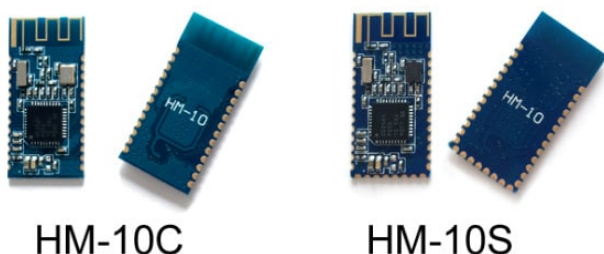


Figure 03: Module HM-10 pour BLE

Le HM-10 est également disponible sur une carte d'extension qui expose les connexions

Références

d'alimentation et UART à des broches compatibles avec les cartes d'essai (breadboard). La carte d'extension inclut un régulateur de puissance de 3.3v qui les rend compatibles 5V. Vous devez noter que la broche RX est toujours en 3.3v et si vous utilisez un Arduino 5v, vous devez convertir le TX 5v de l'Arduino en 3.3v pour le RX du HM-10.



1. **STATE** : broches de test d'état, connectées à la LED interne, généralement maintenues non connectées.
2. **RXD** : interface série, borne de réception.
3. **TXD** : interface série, terminal de transmission.
4. **GND** : Masse.
5. **VCC** : pôle positif de la source d'alimentation.
6. **EN/BRK** : break connect, cela signifie la rupture de la connexion Bluetooth, en général, il faut la garder non connectée.

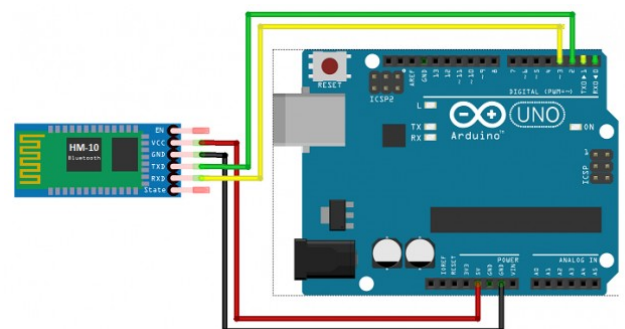


Figure 04: Branchement HM-10 avec Arduino

- [What is the Raspberry Pi 4? Everything you need to know about the tiny, low-cost computer | ZDNet](#)
- [Raspberry Pi 4 Pins - Complete Practical Guide - The Robotics Back-End \(roboticsbackend.com\)](#)
- [ArduinoBLE - Arduino Reference](#)
- [Using Arduino for Bluetooth Low Energy \(BLE\): A Step-by-Step - Arduino Project Hub](#)
- [Bluetooth Low Energy Tutorial with HM-10 BLE 4.0 & Arduino \(how2electronics.com\)](#)