



Module SCI

Ecole Nationale Supérieure d'Informatique d'Alger (Ex. ini)

Compte rendu TP5 (Partie théorique)

24 Avril 2022

Introduction

Pour booster les performances des modules embarqués tout en respectant le compromis énergie-performances, ARDUINO se dote d'une technologie BLE qui lui permet d'accomplir ses fonctionnalités sans se soucier de l'autonomie des batteries associées.

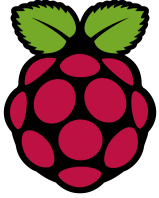
Actuellement, une autre carte électronique sous forme d'un nano-ordinateur envahi le marché, il s'agit de la raspberry Pi.

Objectifs

1. **Partie théorique :** Comprendre ce qu'est une carte raspberry et les différents pins exposés par le GPIO. Pour arduino, nous découvrons le mode de communication BLE (Bluetooth Low Energy)

Partie théorique

A. Résumer le nano-computer Raspberry



Le 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.

Le Raspberry Pi est souvent utilisé pour le traitement d'images et de vidéos en temps réel, les applications IoT et les applications robotiques.

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.

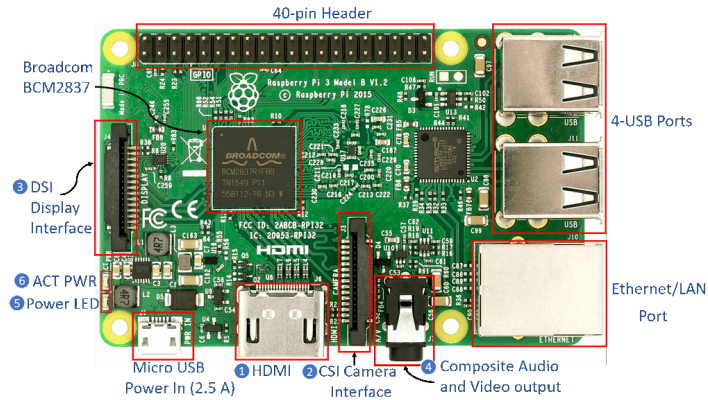
Raspberry Pi dispose également de modules SPI, I2C, I2S et UART intégrés.

Il existe différentes versions de Raspberry Pi, comme indiqué ci-dessous :

- Raspberry Pi 1 Modèle A
- Raspberry Pi 1 Modèle A+
- Raspberry Pi 1 Modèle B
- Raspberry Pi 1 Modèle B+
- Raspberry Pi 2 Modèle B

- Raspberry Pi 3 Modèle B
- Raspberry Pi 4 Modèle B
- Raspberry Pi Zero

B. Etude de la ligne GPIO de Raspberry Pi



GPIO signifie General Purpose

Input/Output (entrée/sortie à usage

général). En gros, il s'agit d'une broche que vous pouvez utiliser pour écrire des données vers des composants externes (sortie), ou lire des données à partir de composants externes (entrée).

Les GPIO du Raspberry Pi 4 sont assez similaires à ce que nous appelons des "broches numériques" sur une carte Arduino. Tous les GPIOs fonctionnent à 3.3V.

Protocoles de communication via les 4 broches du Raspberry Pi :

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.

Raspberry Pi 3 broches UART

Pour utiliser l'UART, nous avons besoin de 3 broches

Alternate Function					Alternate Function
	3.3V PWR	1		2	5V PWR
I2C1 SDA	GPIO 2	3		4	5V PWR
I2C1 SCL	GPIO 3	5		6	GND
	GPIO 4	7		8	UART0 TX
	GND	9		10	UART0 RX
	GPIO 17	11		12	GPIO 18
	GPIO 27	13		14	GND
	GPIO 22	15		16	GPIO 23
	3.3V PWR	17		18	GPIO 24
SPI0 MOSI	GPIO 10	19		20	GND
SPI0 MISO	GPIO 9	21		22	GPIO 25
SPI0 SCLK	GPIO 11	23		24	GPIO 8
	GND	25		26	GPIO 7
	Reserved	27		28	Reserved
	GPIO 5	29		30	GND
	GPIO 6	31		32	GPIO 12
	GPIO 13	33		34	GND
SPI1 MISO	GPIO 19	35		36	GPIO 16
	GPIO 26	37		38	GPIO 20
	GND	39		40	GPIO 21

- GND que vous connecterez à la GND globale de votre circuit.
- RX pour Réception. Vous connecterez cette broche à la broche TX de l'autre composant.
- TX pour Transmission. Vous connecterez cette broche à la broche RX de l'autre composant.
- 5V pour l'alimentation-

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



I2C

I2C est un protocole de bus maître-esclave, 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.

Broches I2C du Raspberry Pi 4:

- GND : Je suppose que vous commencez à vous y habituer !
- SCL : horloge de l'I2C. Connectez tous les esclaves SCL au bus SCL.
- SDA : données échangées. Connectez tous les esclaves SDA au bus SDA.



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.

Broches SPI du Raspberry Pi 4

Comme on peut le voir, nous avons 2 SPI par défaut : SPI0 et SPI1. Cela signifie qu'on pourra utiliser le Raspberry Pi comme maître SPI sur deux bus SPI différents en même temps.

- GND : quelle surprise ! Assurez-vous de connecter tous les GND de tous vos composants esclaves et du Raspberry Pi ensemble.
- SCLK : horloge du SPI. Connectez 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. Faites attention ici : vous aurez besoin d'un CS par esclave sur votre circuit. Par défaut, nous avons deux broches CS (CS0 - GPIO 8 et CS1 - GPIO 7). Vous pouvez configurer plus de broches CS à partir des autres GPIOs disponibles.

Dans notre code, vous pouvez utiliser la bibliothèque spidev pour Python, et WiringPi pour Cpp.

C. Arduino BLE

Contrairement à la communication Bluetooth® standard, qui repose essentiellement sur une connexion série asynchrone (UART), une radio Bluetooth® LE agit comme un tableau d'affichage communautaire. Les ordinateurs qui s'y connectent sont comme les membres de la communauté qui lisent le tableau d'affichage.

Lorsqu'une connexion Bluetooth® est établie, un dispositif, le périphérique, fait de la publicité ou diffuse des informations le concernant à tous les dispositifs proches. Dans le même temps, un autre dispositif, le dispositif central, effectue un balayage et écoute le ou les dispositifs qui diffusent des informations. Dès que le dispositif central capte les informations publicitaires du dispositif périphérique, il tente de connecter le dispositif périphérique. Une fois la connexion établie, le dispositif central interagit avec les informations disponibles sur le périphérique. Cet échange d'informations se fait à l'aide de ce que l'on appelle des services.

Structure de la bibliothèque

Comme la bibliothèque permet plusieurs types de fonctionnalités, il existe un certain nombre de classes différentes.

- **BLE** utilisée pour activer le module Bluetooth® Low Energy.
- **BLEDevice** utilisée pour obtenir des informations sur les périphériques connectés ou découverts lors du balayage.
- **BLEService** utilisé pour activer les services fournis par la carte ou interagir avec les services fournis par une carte distante.
- **BLECharacteristic** utilisé pour activer les caractéristiques que la carte offre dans un service ou interagir avec les caractéristiques qu'une carte distante fournit.
- **BLEDescriptor** utilisé pour décrire une caractéristique offerte par la carte.

