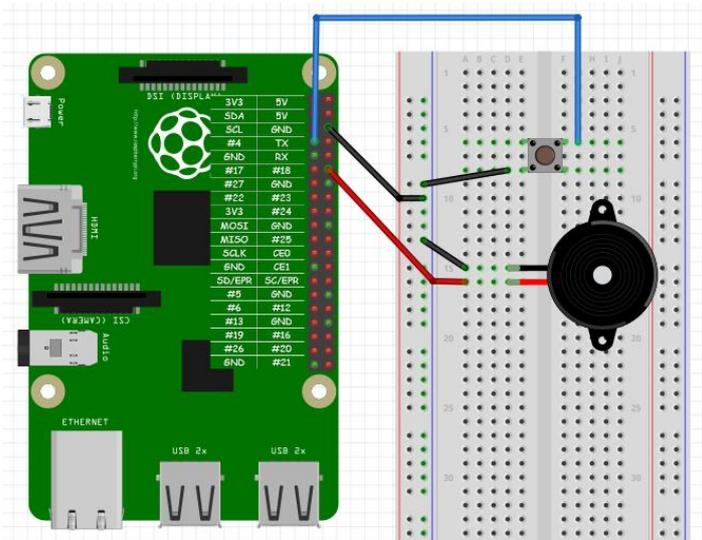
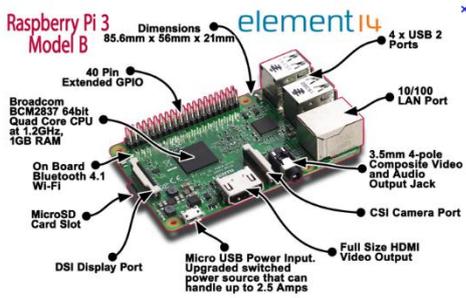
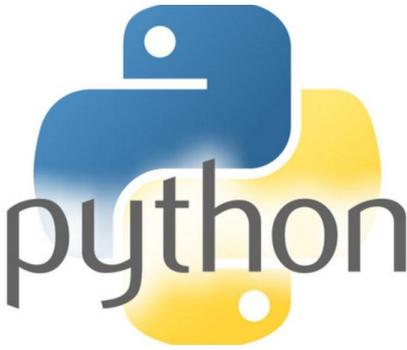


Langage Python appliqué aux systèmes





I. PRESENTATION DE PYTHON

1. Présentation du langage

Python est un langage non spécialisé, open source et orienté objet. Il fonctionne sur la plupart des plateformes, windows, linux et macOS.

Il se dit que :

- Python est simple, à apprendre, lisible
- Python est efficace, concis 3 à 5 fois moins de code que le C,
- Python est rapide, même si c'est un langage interprété
- Python est puissant grâce à ces nombreuses bibliothèques (gain de productivité)
- ...

2. Pourquoi Python dans notre section?

Ceci dit, dans nos domaines industriels, c'est le langage C qui prédomine, ça reste une connaissance indispensable à acquérir.

Ils restent imbattables sur la vitesse d'exécution ou pour les systèmes embarqués ou le contrôle de la mémoire (systèmes embarqués) sont importants. Python n'est pas adapté pour programmer bas niveau, proche du microprocesseur, registres, mémoire ...

Par exemple langage C/arduino permet d'accéder au registre DDR (Data Direction Register D).

La réponse est la Raspberry Pi (Pi pour Python) comme support d'enseignement : il y a eu les PIC, Arduino et maintenant c'est certainement la carte qui offre le plus de possibilités. Elle est très bon marché, puissante, robuste, sa connectique, carte réseau, ses protocoles disponibles (i2c, SPI, série) et grâce à ces GPIOs, des broches, permet de connecter des circuits électroniques pouvant être commandés.

Le python semble le plus approprié pour la programmation de cette carte, même si il est possible d'utiliser d'autres langages (C/C++, QT, java etc...) : il est intégré directement, d'accès facile et rapidement, ajouté à ses qualités intrinsèques (syntaxe, lisibilité, puissance ...) fait de ce langage une évidence.

3. Version du python

Il existe deux versions toujours présentes : python 2 (la dernière est la 2.7) et python 3. Malheureusement le code écrit pour la 2 ne fonctionne pas, sans modifications plus ou moins importantes, en python 3.

Il est dit que : « Python 2 est l'héritage et Python 3 le présent et le futur »

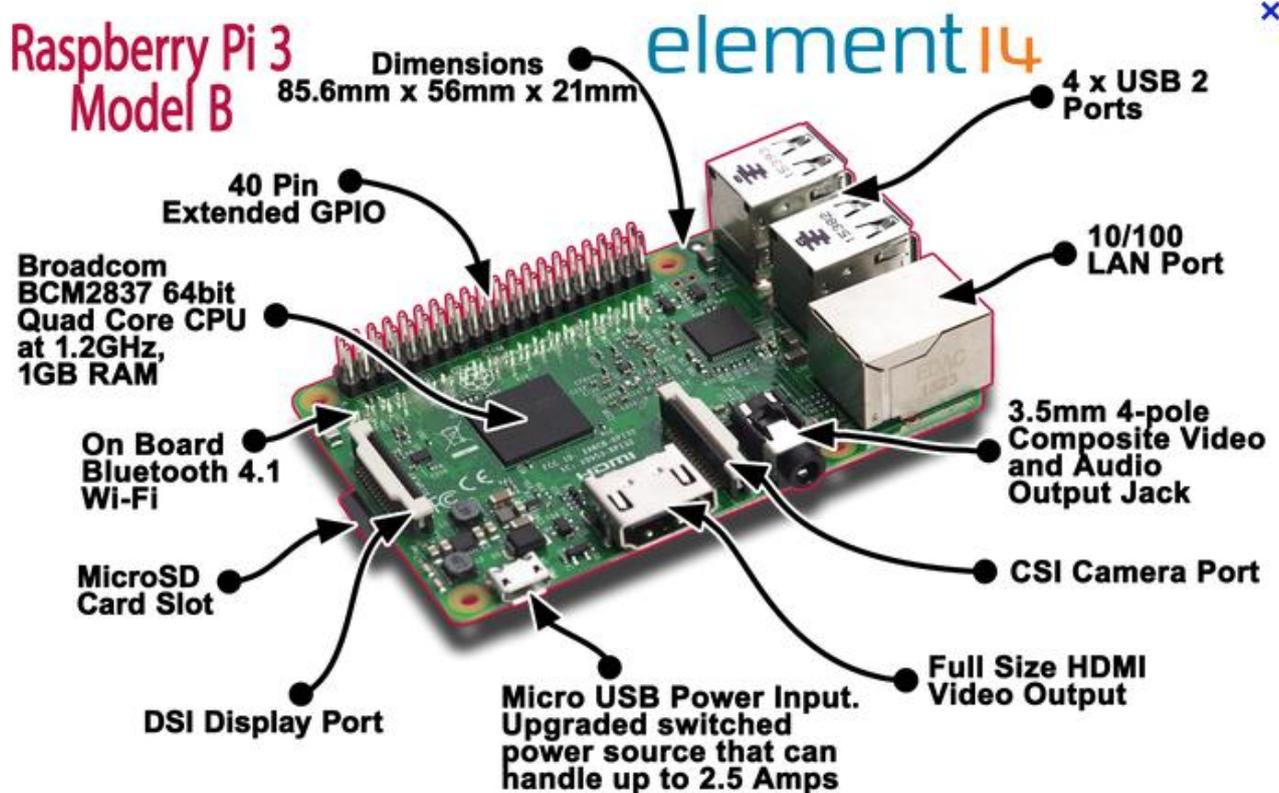
Il faut savoir jongler avec les deux versions, certains modules/bibliothèques ne fonctionnent qu'en python 2 tandis que d'autres en 3.



Le choix a été fait d'apprendre python avec la version la plus courante sur raspberry Pi, la 2.7.
A la fin des TP, vous allez passer à la version 3 et voir l'importance de cette nouvelle version.

II. PRESENTATION RASPBERRY PI

1. Présentation raspberry Pi



C'est donc un mini-ordinateur (ou nano-ordinateur) dont le cœur est un SoC (System On Circuit) qui inclus un CPU, GPU (Graphics Processing Unit), DSP (traitement numérique du signal) SDRAM ... Le GPU du SoC Broadcom libère le CPU de lourds calculs.



2. RPI ou Arduino ?



Pour faire court, le raspberry Pi est un véritable ordinateur, appelé nano-ordinateur, tournant sous Linux, et l'Arduino simplement un microcontrôleur (+ un IDE).

a. historique

Le RPI naquit en Angleterre : les fondateurs constatèrent que le niveau en programmation des élèves de 3^{ème} (A-Level) des années 2000 était bien inférieur à ceux de 90. En effet, dans ces années il n'était pas rare de trouver un micro-ordinateur de type Amiga, commodore 64, amstrad ... De nos jours, ils sont remplacés par des « cours d'informatique » word , excel, consoles, tablettes ou au mieux l'écriture de pages web.

Le coût devait être faible : leur but était que chacun jeunes anglais aient leurs propres ordinateurs. D'où vient le nom ? Initialement, il devait s'appeler 'ABC Micro' mais pour suivre une longue tradition qui consiste à choisir un nom de fruit, « Raspberry » fut choisi. La deuxième partie le « Pi » provient du langage "Python", qui a été choisi très tôt comme langage de programmation de la carte.

L'histoire de l'Arduino a commencé en 2005 dans une école située dans la ville Ivrea en Italie.

L'objectif des fondateurs est de permettre à des non-spécialistes d'informatique-d'électronique qui ne veulent pas forcément être des programmeurs, d'utiliser la programmation, le matériel pour arriver à leurs fins. l'expérience utilisateur est conçue pour minimiser le temps entre l'idée et la conception sur la board

D'où vient le nom Arduino ? Arduino a été le premier roi de la région (de 1002 à 1014 pour être plus précis) dans laquelle se situe Ivrea, mais c'est aussi (et surtout ?) le nom d'un bar où les étudiants et les professeurs d'Ivrea se rassemblaient.

Donc le but est identique : être des outils pour apprendre la programmation à des débutants.

b. Comparaison logiciel/matériel

	Arduino Uno	Raspberry PI 2 modèle B+
Prix	20 €	38 €
Taille	68,6 mm x 53,3 mm (Uno)	85,60 mm x 53,98 mm (Rpi 2)
Mémoire	2 ko	1 Go
Horloge	16 Mhz	900MHz
Microcontrôleur	ATmega328	ARM Cortex-A7 4 cœurs (SoC)



BTS SN - EC

Raspberry Pi

Présentation



Réseau	Non	Ethernet
Mémoire flash	32 Ko	De 16 à 32 Go
Nombre de broches	20 broches	40 broches d'extension GPIO
Tension des broches	5 volts	3.3 volts
USB	1 mais uniquement pour la communication	2 * 2 ports USB également pour connexion périphériques
Système d'exploitation	Aucun	Toutes distributions Linux + Windows 10 IOT
Outils de développement	IDE Arduino (environnement de développement + bibliothèques)	Scratch, IDLE (python), mais aussi toutes applications Linux : langage C, php, java ...
Facilité d'utilisation	Très facile	Demande un temps d'adaptation

Cette comparaison ne s'effectue qu'avec l'Arduino Uno, il en existe beaucoup qui sont plus puissantes (teensy, arduino zero) et qui pourraient rivaliser sur le plan performance avec le rpi.

c. IDE Arduino vs Linux du RPI

Généralement, on résume l'Arduino à une carte et un EDI (Environnement de développement intégré, IDE en anglais).

Je préfère y ajouter un troisième élément : les bibliothèques/fonctions. En effet, la complexité est encapsulée dans des fonctions comme par exemple pinMode, la librairie LCD, la classe permettant de gérer les cartes SD ...

Tout cet environnement fait que

Le raspberry quand à lui, s'appuie sur des distributions Linux :

- Communauté très présente sur Internet, il est très facile de trouver une manipulation même en français
- Stabilité de l'OS, ouvert, open source
- Continuité, par exemple la manipulation pour modifier le nom de l'ordinateur est la même depuis au moins 20 ans (modification des fichiers /etc/hosts, /etc/hostname et /etc/init.d/hostname.sh)

d. Type de projets

Voici une liste non exhaustive de projets :

Arduino	Raspberry Pi
Initier les élèves à la programmation, à l'utilisation de composants	Apprendre la programmation avec Python ou Scratch
Allumer des leds, utiliser des boutons poussoirs, calculer la distance d'un objet	Idem mais allumer la led à l'aide d'une page web, afficher la distance
Faire des montages électroniques simples, compliqué	Laisser Arduino faire ces tâches, et le piloter avec d'autres cartes Arduino et/ou d'autres composants



Mesurer la température, l'humidité ...	Effectuer ces mesures les injecter dans une base de données pour effectuer des statiques
Contrôler les roues d'un châssis d'une voiture/robot, piloter un drone	Oui mais
Contrôler les roues d'un châssis d'une voiture/robot	Piloter un voiture/robot en utilisant à l'aide d'une manette contenant un accéléromètre et un compat i2C et diffuser la vidéo sur dans un smartphone
	Utiliser comme ordinateur pour surfer, écouter de la musique, regarder un film
	En faire une station multimédia

J'utilise Arduino pour tester rapidement un nouveau composant et le raspberry pi pour des projets soient plus importants ou nécessite un site web, une base de données, une connexion réseau, des programmes plus complexes, une caméra, de l'audio

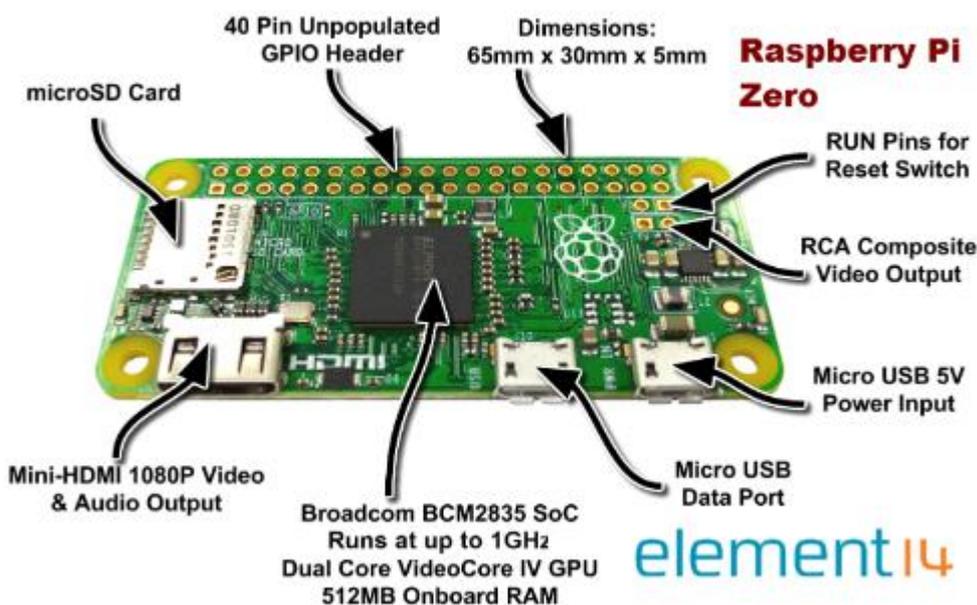
e. Plutôt Arduino ET raspberry Pi

Nous le ferons par la suite sur des exemples précis, mais souvent dans un projet Arduino est utilisé conjointement avec le Rpi pour plusieurs raisons :

- Pour bien séparer les différentes sous-parties du projet SI/STI2D/BTS soit pour une meilleure visibilité de la répartition des tâches, soit pour des questions de carte disponible
- Laisser à l'Arduino la partie commande (de moteurs, servo-moteurs etc...) il est adapté pour cela

3. Les différentes cartes

a. Pi Zero





b. Pi 1, modèle A+



Processeur Broadcom BCM2835 700 MHz

RAM : 512 Mo

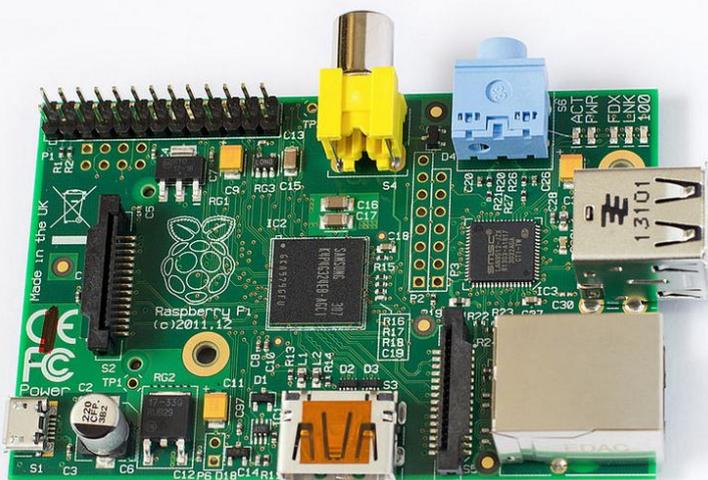
1 port USB 2.0

Extension GPIO
40 broches

Sorties vidéo HDMI et
RCA

Prix
17 € HT

c. Raspberry 1 Modèle B (plus disponible)



Processeur Broadcom BCM2835
700 MHz

RAM : 512 Mo

2 ports USB 2.0

26 broches GPIO

Sorties vidéo HDMI et RCA

d. Raspberry 1 Modèle B+



- Processeur Broadcom BCM2835 700 MHz
- RAM : 512 Mo
- 4 ports USB 2.0
- Extension GPIO 40 broches
- Sorties vidéo HDMI et RC

e. Raspberry Pi 2, modèle B



Processeur
 Broadcom
 BCM2836, quatre
 cœurs 900 MHz

RAM : 1 Go

4 ports USB 2.0

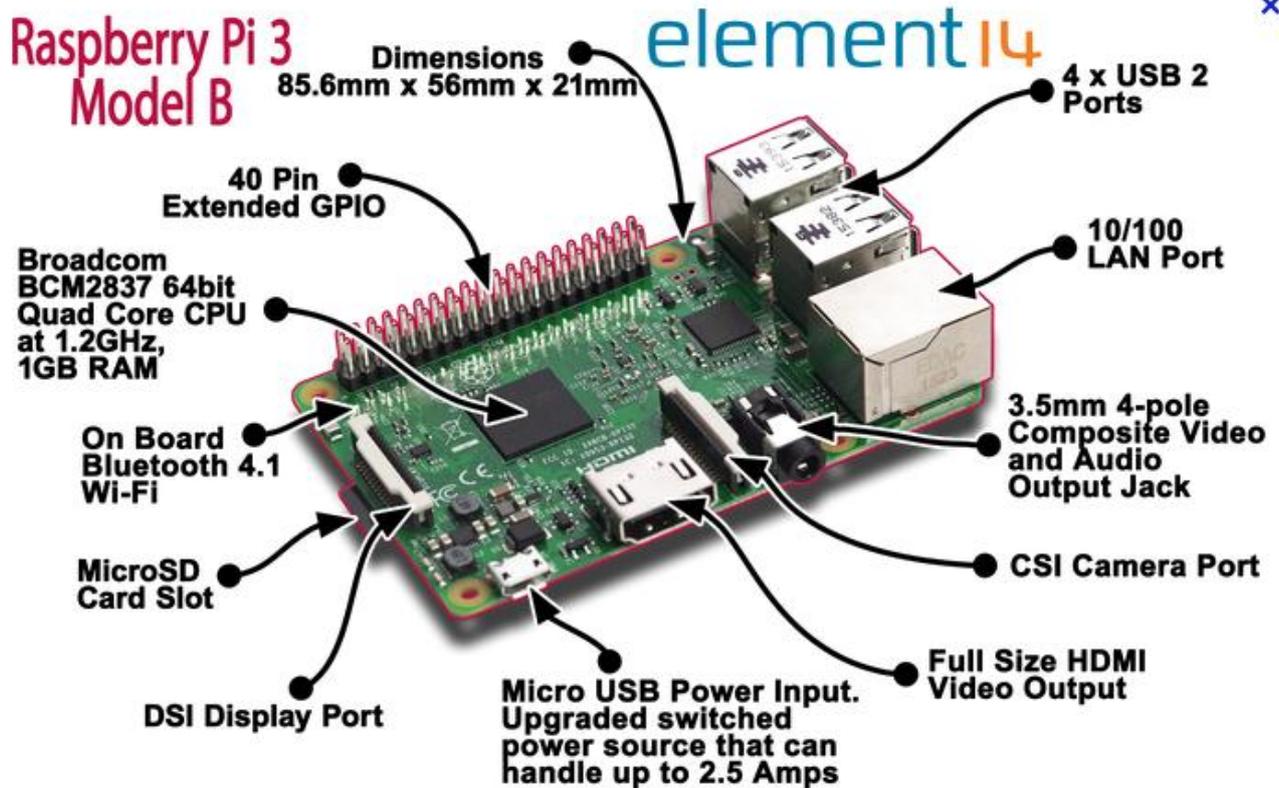
Extension GPIO
 40 broches

Sorties vidéo HDMI
 et RCA

Prix : 38 €



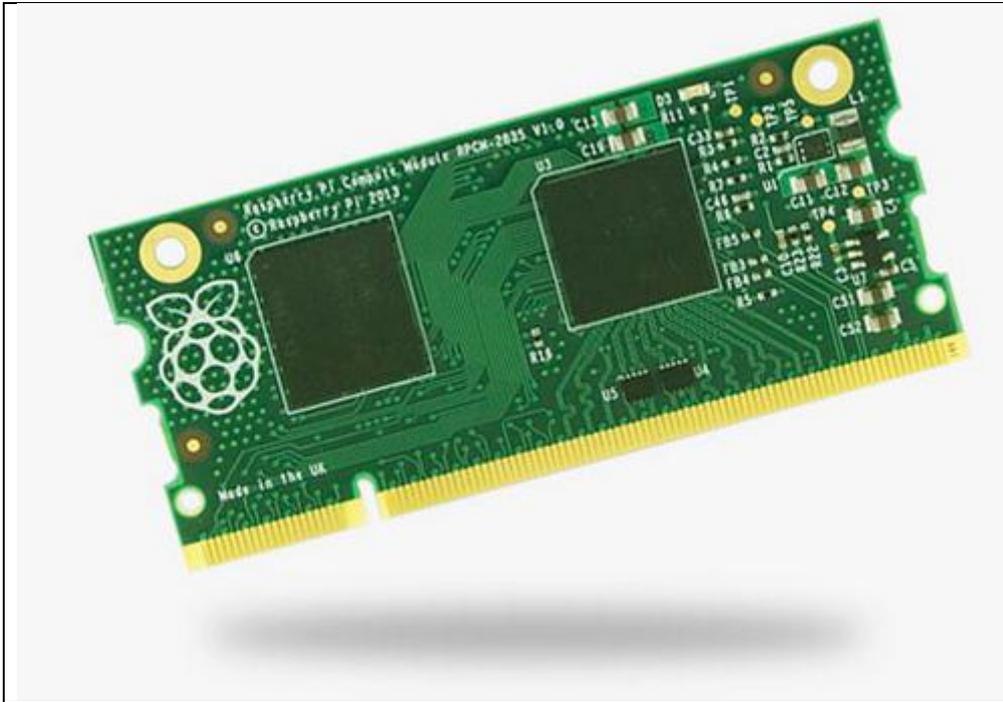
f. Raspberry Pi 3 - Modèle B



- Processeur Broadcom BCM2837 64 bits ARMv8 quatre cœurs 1,2 GHz
- Carte Wi-Fi BCM43438
- Carte Bluetooth Low Energy (BLE)
- RAM : 1 Go
- 4 ports USB 2
- Extension GPIO 40 broches
- Sorties vidéo HDMI et RCA
- Tarif : 46 €



g. Module de calcul Raspberry Pi



- Processeur
Broadcom
BCM2835
700 MHz
- RAM : 512 Mo
- Mémoire Flash
eMMC 4 Go
- À la taille d'une
carte SO-DIMM,
6 x 3,5 cm

4. Quelques caractéristiques

Pour le Pi Zero : branché à une télé HD 1080, un clavier un souris la consommation n'est que de 160 mA
 Le temps de démarrage, de boot :

- ▶ Boot jusque la ligne de commande 14.5s pour le rpi 2, 27 pour le Zero
- ▶ Boot jusque le desktop : RPI 2 18s, Zero 43s

5. Les HAT

Sous Arduino les cartes permettant d'ajouter de nouvelles fonctionnalités s'appellent des shields ou boucliers en français.

Pour le raspberry Pi, ces cartes se nomment des HAT pour Hardware Attached on Top (chapeau ?) et se positionnent, comme les shields, au dessus :



Et peuvent s'empiler :

