Électronique

Arduino vs Raspberry Pi

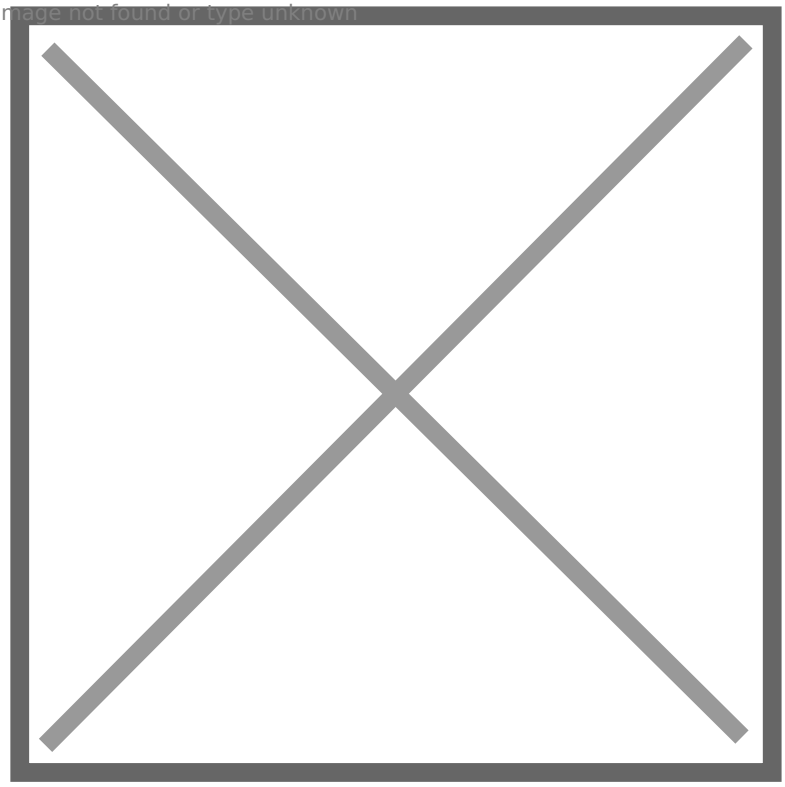
Quelles sont les différences entre Arduino et Raspberry Pi ?

Bien qu'elles se ressemblent et qu'elles peuvent être utilisées pour faire des choses similaires, ces deux cartes sont très différentes !

Arduino et Raspberry Pi sont des choix assez populaires dans les projets IoT (Internet of Things - l'internet des objets) ou de domotique. Les deux cartes vont pouvoir interagir avec leur environnement par le biais de capteurs (température, lumière, son...) et d'actionneurs (LED, moteurs, hauts-parleurs...).

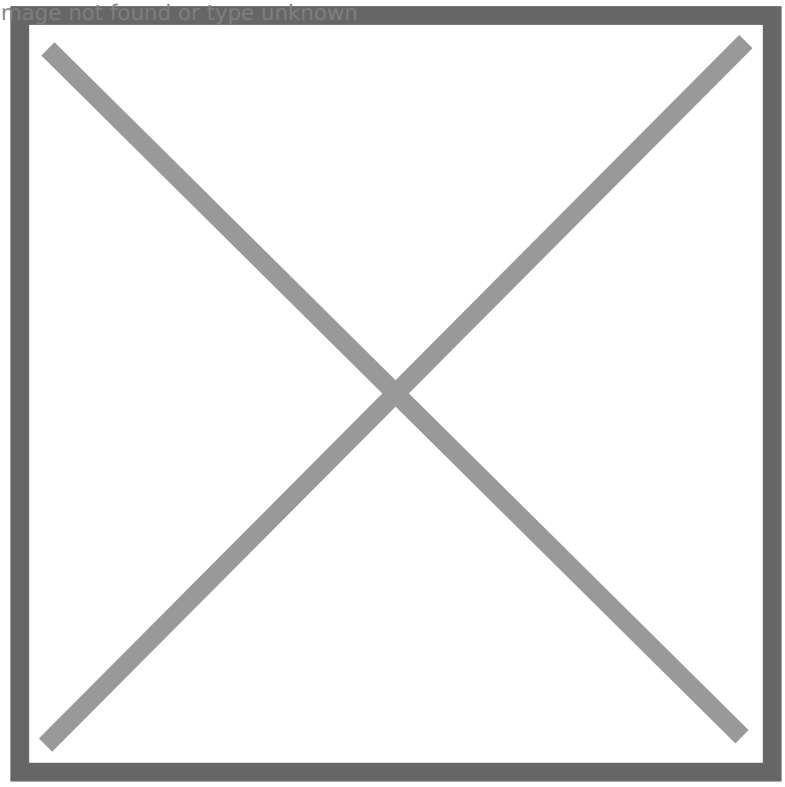
Le Raspberry Pi est un petit ordinateur monocarte pleinement fonctionnel. Il est doté d'un processeur, d'une mémoire et d'un système d'exploitation dédié (Linux). Ainsi il possède ses propres ports USB, une sortie audio et une sortie HDMI. Comme un ordinateur normal il peut exécuter plusieurs programmes en même temps et se connecter en WiFi et Bluetooth.

Image not found or type unknown



À l'inverse, l'Arduino est une carte dotée d'un microcontrôleur. Un microcontrôleur n'est pas aussi puissant qu'un ordinateur et il ne peut exécuter qu'un seul programme à la fois. L'avantage c'est qu'ils sont très rapides à configurer et permettent de contrôler facilement des capteurs et des actionneurs. En version de base et sans blindage ou shields (cartes additionnelles), un Arduino n'a aucune connectivité WiFi ou Bluetooth.

Image not found or type unknown



Il est à noter que chacune de ces cartes existent en plusieurs tailles et les plus petites versions ne prennent que très peu de place.

Dans quel cas va-t-on utiliser l'un plutôt que l'autre ? Cela va principalement dépendre de la puissance de calcul nécessaire. Pour un projet multimédia par exemple on utilisera un Raspberry Pi, tandis que pour un projet électronique avec des capteurs et des moteurs on optera plutôt pour un Arduino.

L'électronique avec Arduino

Arduino est une carte programmable créée par une équipe italienne. Elle facilite grandement l'utilisation des microcontrôleurs grâce à son interface de développement simple et disponible pour Windows, Linux et Mac. Elle permet d'ajouter une intelligence à vos montages électroniques et d'automatiser n'importe quelle tâche.

Où trouver de l'aide

Il existe une importante communauté autour de l'Arduino et vous trouverez beaucoup de réponses à vos problèmes avec une recherche Google.

Quelques liens :

- le site officiel Arduino (en anglais) et en particulier son forum ,
- le lien correspondant en français : Arduino (traduction partielle),
- le tutoriel et le forum dédié sur le site de la communauté francophone Open Classrooms (anciennement SiteDuZéro),
- le fil dédié sur Instructables,
- le subreddit dédié sur reddit

Tutoriels

- Arduino
- Lire la température de 3 sondes thermiques DS18B20 à partir d'un arduino
- Stocker les données envoyées par un arduino sur un fichier excel (uniquement sous Windsows)
- Les Quartz et Oscillateurs
- La mémoire flash (SPI)
- Utiliser un analyseur logique
- Utiliser l'émetteur/récepteur RF 433MHz
- Programmation d'un PIC à l'aide du PicKit3 et de MPLAB X IDE en C
- Les Servomoteurs
- Bases de l'électronique numérique (En cours)
- Fil résistif chauffant (En cours)

- Capteurs Grove (pour Arduino) En cours

L'électronique avec Raspberry Pi

Les Raspberry Pi sont des petits ordinateurs fonctionnant avec une distribution Linux et qui possèdent en plus des entrées et sorties programmables qui peuvent être utilisées dans des projets d'électronique. On peut également y brancher des périphériques externes en USB, une caméra, des périphériques I2C, etc...

On peut les utiliser pour faire un serveur, du traitement d'image, de l'IA... les possibilités sont nombreuses.

Les Raspberry Pi peuvent se programmer dans de nombreux langages de programmation mais le plus populaire reste Python.

- Série de tutoriels vidéo en anglais sur les bases de Raspberry Pi
- Tutoriel en français sur comment utiliser un bouton avec le Raspberry Pi
- Tutoriel Python complet

Composants et circuits

Ici vous trouverez des exemples de circuits pour des applications bien spécifiques.

> Documentation complète avec exemples de circuits et de code Arduino pour nos kits de capteurs : kit-capteurs.pdf

- Contrôleur de boost MAX1522 : pour convertir une alimentation de 3.7V en 12V par exemple.
- Amplificateur opérationnel : toutes les configurations possibles avec calculs des gains et des réponses en fréquence
- Microcontrôleur SAMD11C : comment uploader du code avec l'outil Atmel ICE
- Capteur de champ magnétique TLE493D : pour mesurer l'intensité et la position dans l'espace du champ magnétique

Librairies pour KiCAD et Eagle

- Base de données d'empreintes de composants (nécessite la création d'un compte gratuit)
- Base de données d'empreintes de composants

Les microcontrôleurs

Choisir un microcontrôleur

Nous avons plusieurs références de microcontrôleurs (MCU) au lab et pour mieux choisir voici un tableau comparatif de leurs caractéristiques :

	ATTiny44A	ATTiny45	ATMega328P	ATTiny412	ATTiny1614	ATTiny3216	AVR128DB32	ATSAMD11C14U	ATSAMD11D14U	ATSAMD20E17A	ATSAMD21E15	STM32F051K6T6	STM32F030F4P6	STM32F042F4P6
Tension d'alimentation	2.7-5.5V	2.7-5.5V	2.7-5.5V	1.8-5.5V	1.8-5.5V	1.8-5.5V	1.8-5.5V	1.6-3.6V	1.6-3.6V	1.6-3.6V	1.6-3.6V	2.4-3.6V	2.4-3.6V	2-3.6V
Architecture	AVR	AVR	AVR	AVR	AVR	AVR	AVR	ARM	ARM	ARM	ARM	ARM	ARM	ARM
Debug	ISP	ISP	ISP	UPDI	UPDI	UPDI	UPDI	SWD	SWD	SWD	SWD	SWD	SWD	SWD
Fréquence d'horloge	20MHz	20MHz	20MHz	20MHz	20MHz	20MHz	24MHz	48MHz	48MHz	48MHz	48MHz	48MHz	48MHz	48MHz
Flash	4K	4K	32K	4K	16K	32K	128K	16K	16K	128K	32K	32K	16K	16K
SRAM	256B	256B	2K	256B	2K	2K	16K	4K	4K	16K	4K	8K	4K	6K
GPIO	12	6	23	6	12	18	26	12	18	26	26	25	15	16
PWM	2	2	6	1	1	1	4	4	4	8	12	6	6	6
ADC	4	4	8	1	2	2	13	10	10	20	20	16	11	12
DAC	-	-	-	1	3	3	1	1	1	1	1	1	-	-
I2C	1	1	1	1	1	1	2	3	3	6	6	1	1	1
SPI	1	1	1	1	1	1	2					1	1	1
UART	1	1	1	1	1	1	3					2	1	2
Capacitive touch	-	-	-	-	6	12	-	12	18	26	26	13	-	14
USB	-	-	-	-	-	-	-	x	x	x	x	-	-	x
RTC	-	-	-	x	x	x	x	x	x	x	x	x	x	x
CAN	-	-	-	-	-	-	-	-	-	-	-	-	-	1
Compatibilité Arduino	+++	+++	+++	++	++	++	++	+	+	-	-	?	?	+

> Attention, tous les microcontrôleurs ne sont pas parfaitement pris en charge par l'environnement Arduino et les bibliothèques associées.

Si votre projet nécessite une connectivité WiFi ou Bluetooth il faudra opter pour un ESP32. Ils ne sont pas représentés dans le tableau car ils sont hors catégorie. Ils sont plus puissants et ont beaucoup plus de mémoire. On peut d'ailleurs les utiliser avec Micro Python.

Les microcontrôleurs sont une denrée rare et précieuse actuellement, donc choisir forcément celui qui est le plus puissant ou avec le plus d'E/S est une mauvaise idée. Il faut réfléchir en amont aux caractéristiques du projet et prendre celui qui est le plus adapté.

Les ATTiny44, 45 et les ATMega328P sont des technologies un peu datées (8 bits) mais suffisamment puissantes pour de nombreux projets. Ils ont l'avantage d'être parfaitement pris en charge par l'environnement Arduino. Le protocole utilisé pour uploader du code est toutefois moins pratique à mettre en oeuvre.

> Pour un novice, les MCU les plus polyvalents et les plus faciles à utiliser sont les ATTiny412, 1614 et 3216. Le protocole UPDI pour charger le code est facile à mettre en oeuvre, ils sont relativement puissants et sont compatibles avec la grande majorité des bibliothèques Arduino.

Les SAMD sont plus puissants (32 bits) et ont l'avantage de gérer la connexion USB en natif, mais leur prise en charge par Arduino n'est pas encore optimale.

Les STM32 sont des microcontrôleurs très puissants mais plus complexes à utiliser, car ils nécessitent d'utiliser l'environnement de développement de STM et ses bibliothèques. On a plus de possibilités mais au prix d'une plus grande complexité. Réservé aux utilisateurs aguerris.

Exemples de projets et documentation

- ATTiny44A (datasheet)
  - Blinky and button
  - Phototransistor
- ATTiny45 (datasheet)
- ATMega328P (datasheet)
- ATTiny412 (datasheet)
  - RGB LED
- ATTiny1614 (datasheet)
- ATTiny3216 (datasheet)
  - RGB LED
- AVR128DB32 (datasheet)
- ATSAMD11C14U (datasheet)
  - Blinky
  - Miniduino (development board)
- ATSAMD11D14U (datasheet)
  - Development board
- ATSAMD20E17A (datasheet)
- ATSAMD21E15 (datasheet)
- ESP32
  - Development board
- STM32F051K6T6 (datasheet / reference manual)

- STM32F030F4P6 (datasheet / reference manual)
- STM32F042F4P6 (datasheet / reference manual)

Démarrer avec les ATTiny (1-series)

Quelques ressources pour démarrer avec les ATTiny412, 1614 ou 3216 :

- Le programmeur / déboggueur
-

Moteur pas à pas (stepper motor)

Les moteurs pas à pas

...

Pour contrôler un moteur pas à pas bipolaire, le plus simple est d'utiliser un driver tel que le A4988. Ce type de driver offre une interface très simple pour connecter à Arduino.

Les meilleurs drivers et celui que nous allons utiliser ici est le SilentStepStick avec la puce TMC2209 :

https://learn.watterott.com/fr/silentstepstick/

https://www.airspayce.com/mikem/arduino/AccelStepper/index.html