



جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat

L'internet des objets Matériel & Logiciel

**Séance 6
Master spécialisé IPS**

Plan

- 1- Rappel : IoT - Vue d'ensemble
- 2- Rappel : Architecture de l'IoT
- 3- Protocoles IoT- Partie 2
- 4- Matériel et Logiciel
- 5- Raspberry Pi Platform

IoT : Vue d'ensemble



- Infrastructure (ex: 6LowPAN, IPv4/IPv6, RPL)
- Identification (ex: EPC, uCode, IPv6, URIs)
- Comms / Transport (ex: Wifi, Bluetooth, LPWAN)
- Data Protocols (ex: MQTT, CoAP, AMQP, Websocket)
- Semantic (ex: JSON-LD, Web Thing Model)

Architecture générale de bout en bout

Business

- Gérer l'ensemble des activités et services du système IoT
- Accompagnement des processus décisionnels

Application

- Fournir les services demandés par les clients
- Outils de visualisation

Middleware

- Donner accès aux services
- Traitement des données reçues et prise de décision

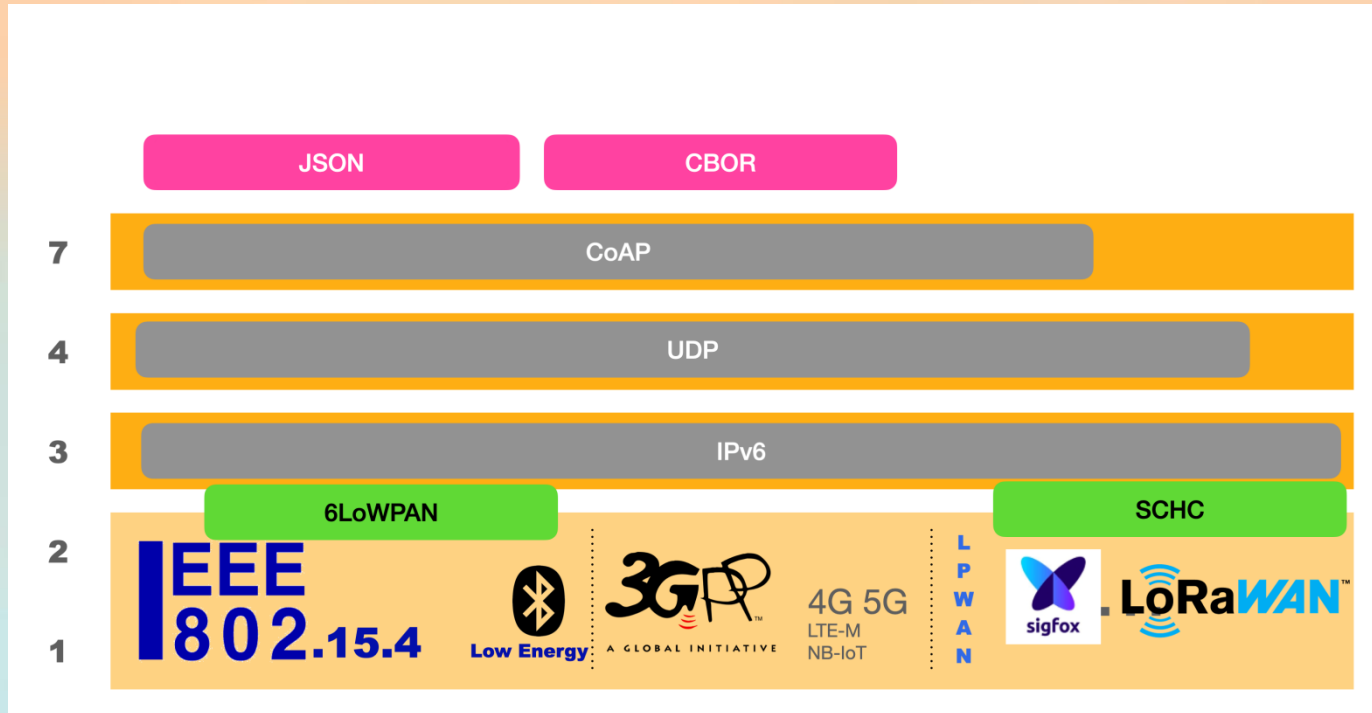
Réseau

- Transfert des données collectées de la couche Dispositifs à la couche Middleware

Dispositifs

- Objets physiques: capteurs et actionneurs
- Recueillir les événements détectés

Architecture de l'IoT : Rappel



Protocoles IoT- partie 2

Application Protocol		DDS	CoAP	AMQP	MQTT	MQTT-NS	XMPP	HTTP REST
Service Discovery		mDNS				DNS-SD		
Infrastructure Protocols	Routing Protocol	RPL						
	Network Layer	6LoWPAN				IPv4/IPv6		
	Link Layer	IEEE 802.15.4						
	Physical/ Device Layer	LTE-A	EPCglobal		IEEE 802.15.4		Z-Wave	
Influential Protocols		IEEE 1888.3, IPsec				IEEE 1905.1		

Protocoles de la couche application

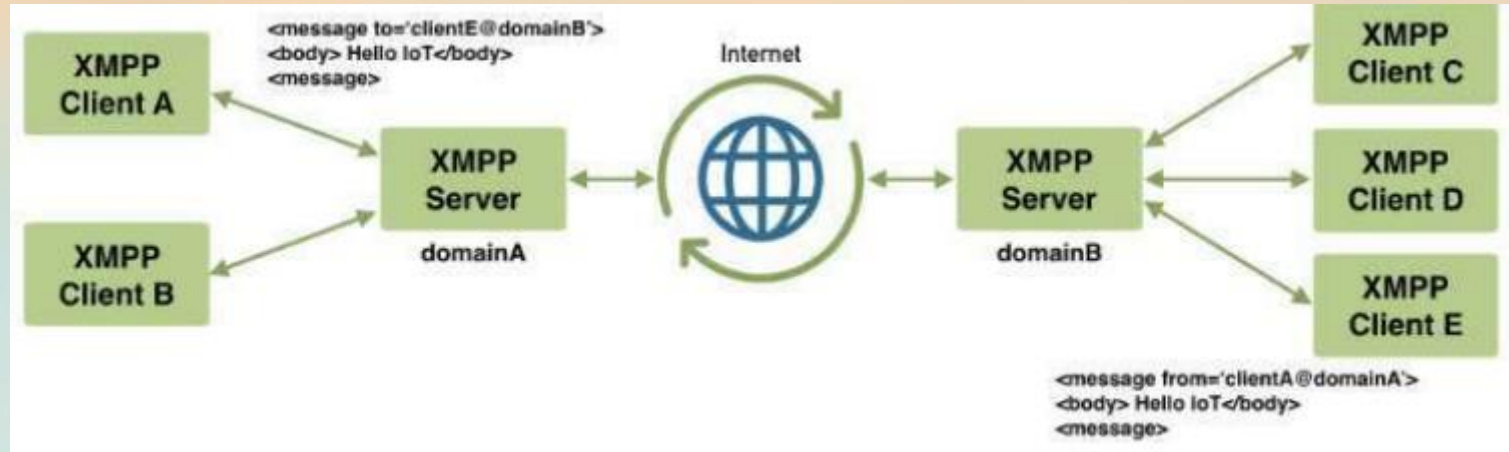
Les protocoles applicatifs qui utilisent un nombre limité de messages de petites tailles sont utilisés pour les applications IoT, et sont classés en 3 familles:

- Protocole de transfert web: Web REST, COAP
- Protocole de messagerie: MQTT, XMPP et AMQ.
- Protocole réseau: Websocket

XMPP

- XMPP (Extensible Messaging and Presence Protocol), est à l'origine un protocole de messagerie instantanée utilisé notamment dans les services Jabber et Google Talk.
- Grâce à son extensibilité, il est utilisé dans d'autres applications telle que la VoIP.
- Son fonctionnement est basé sur une architecture client/serveur où l'échange de données, au format XML.
- La communication entre deux clients est asynchrone et est réalisée au travers de serveurs XMPP.

Fonctionnement du protocole XMPP



XMPP

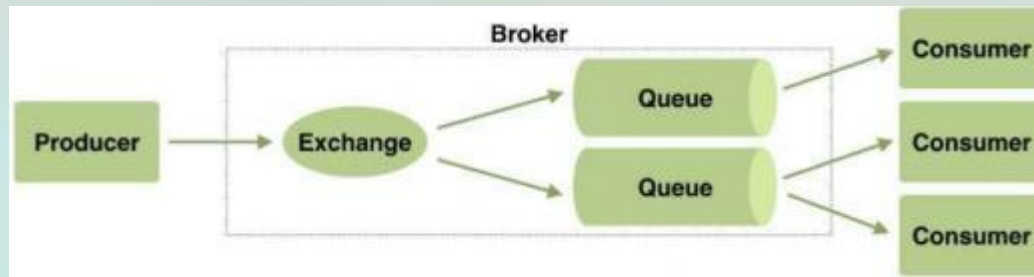
- Les principaux atouts de ce protocole sont son adressage avec identifiant unique, sa facilité de mise en place de la sécurité, son format de messages qui fournit des données structurées et son système de serveurs.
- Le protocole XMPP est plus adapté à l'IoT, contrairement au protocole MQTT (adaptés aux applications M2M):
 - il gère mieux l'intégration de nouveaux objets connectés, et
 - Il permet interopérabilité avec d'autres plateformes IoT et donc d'autres écosystèmes IoT.

AMQP

- Le fonctionnement du protocole AMQP est basé sur le même principe que celui de MQTT, toutefois la notion de publisher/subscriber est remplacée par celle de producer/consumer.
- En outre, grâce à un mécanisme interne noté « exchange », AMQP permet de router un message d'un producer vers **plusieurs** topics. Les critères de routage peuvent se faire de plusieurs façons ; inspection du contenu, de l'en-tête, clés de routage, etc. Ainsi, un même message peut être consommé par différents consumers via plusieurs topics.

AMQP

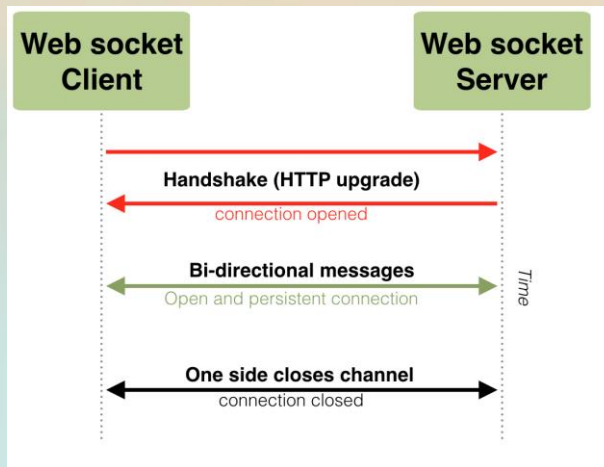
- Par conséquent, AMQP est plus adapté aux situations exigeant la fiabilité, des scénarios de messageries plus sophistiqués, l'interopérabilité entre implémentations du protocole et la sécurité. Ainsi, il est plus destiné aux objets connectés avec des contraintes de communication faibles et des exigences de sécurités importantes.



Protocole réseau (Websocket)

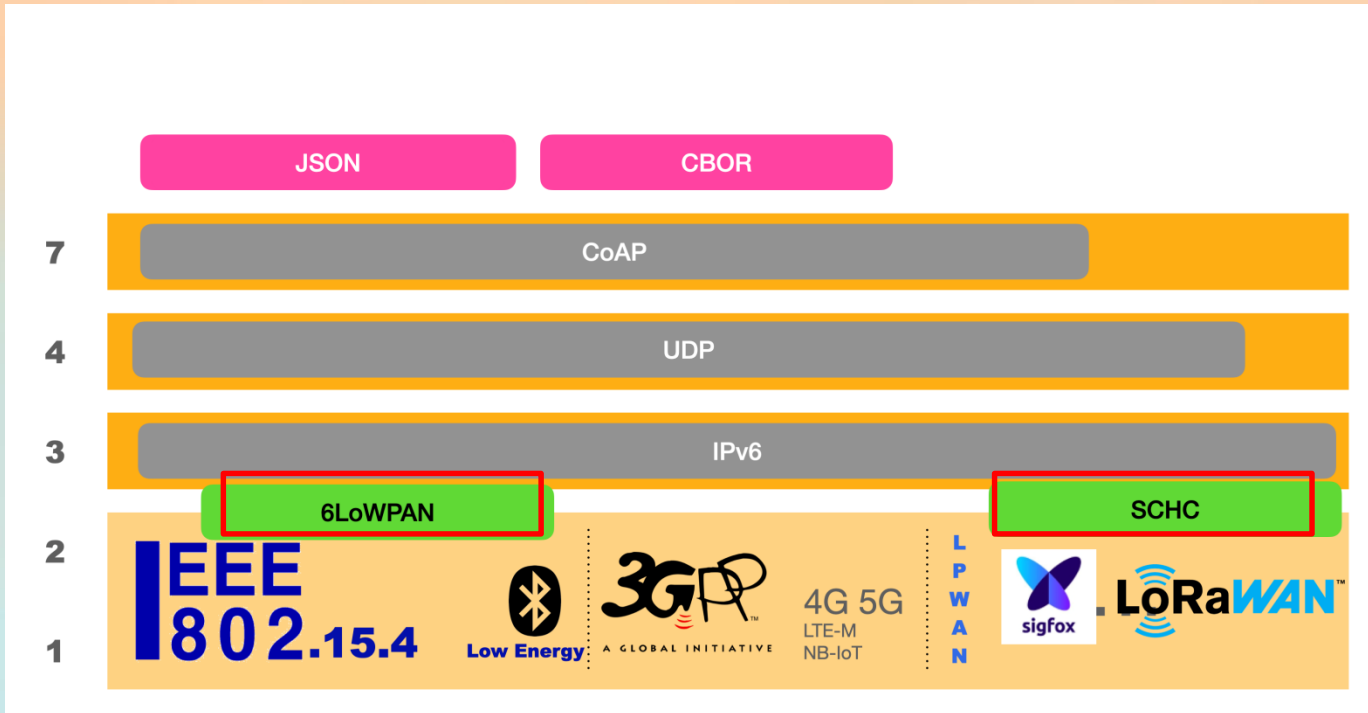
Le protocole Websocket permet l'établissement d'un canal de communication [full-duplex](#) en une seule connexion TCP entre un client et un serveur. L'image indique les trois principales phases de la vie du canal :

- la phase de connexion appelé « Handshake » initié par le client
- la phase d'échange bidirectionnel de messages
- la phase de clôture du canal initié par l'une des deux parties



Ce genre de communication est coûteux en termes de ressources matérielles (CPU, mémoires, source d'énergie) et de bande passante, il n'est donc adapté qu'à des capteurs avec des ressources suffisantes. Il est principalement adapté aux situations de surveillance et d'envoi d'informations en temps réel.

Couche d'adaptation



Couche d'adaptation

Longueur de l'entête IPv6 % à la donnée: écart.

Cet écart sera d'autant plus grand en regard des données transportées qui ne pèsent généralement que quelques octets (mesures d'une température...).

Deux actions sont mises en œuvre :

- Compression de la taille des en-têtes pour réduire leur impact.
- Fragmentation pour découper le paquet en petites trames si la première mesure ne suffit pas.
- Il existe deux grandes familles de couche d'adaptation :
6LoWPAN et SCHC

6LoWPAN

Standard IETF – RFC 4944 : 6LoWPAN (2007)

- 6LoWPAN (IPv6 Low Power Wireless Personal Area Network) est une combinaison de deux protocoles : Internet Protocol version (IPv6) et Low-Power Wireless Personal Network (LoWPAN).
- 6LoWPAN a été conçu pour permettre à IPv6 d'intégrer les appareils contraints et les réseaux 802.15.4 qui les interconnectent
- Les paquets IPv6 ont des en-têtes de taille fixe à 40 octet : taille inconvenable pour les réseaux IEEE 802.15.4. 6LoWPAN permet a des objets 802.15.4 de communiquer sur les réseaux IPv6 de sorte que la connexion de bout en bout est adressable et qu'un routeur peut être utilisé pour le routage des tâches.

6LoWPAN

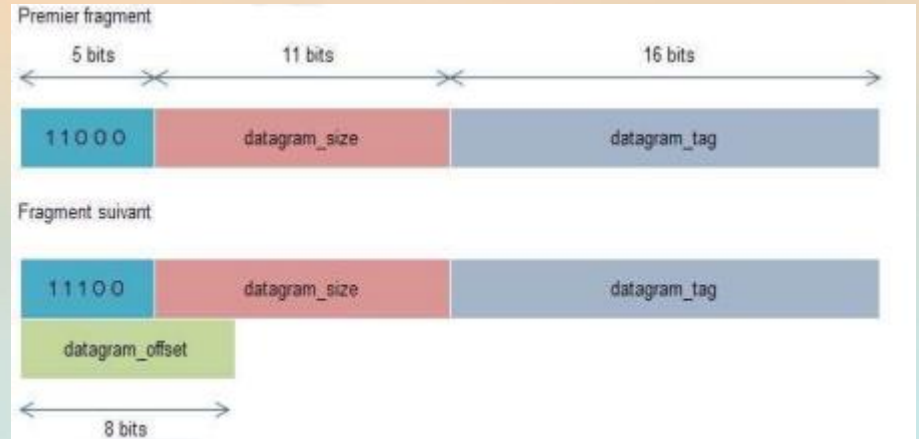
6LoWPAN est une couche d'adaptation qui réside entre la couche liaison de données et la couche réseau et réalise les fonctions suivantes :

- Fragmentation et rassemblements des paquets.
- Compression des en-têtes.
- Routage (connexions multi-sauts).
- Le standard 6LowPan ne prévoit pas de fonctions de sécurité en plus de celles potentiellement mises en œuvre au niveau du 802.15.4 et de IP v6

6LoWPAN

Fragmentation et Réassemblage

- La couche adaptation 6LoWPAN doit fragmenter les paquets IPv6 avant de les envoyer et les réassembler à la réception.
- Chaque fragment est précédé d'un en-tête de 4 ou 5 octets qui contient :
 - 5 bits : permet d'identifier qu'il s'agit d'un fragment.
 - 8 bits : position du fragment dans le paquet IP (uniquement présent dans les fragments suivant le premier).
 - 11 bits : taille du paquet IP avant fragmentation
 - 16 bits : identifiant commun à tous les fragments d'un même paquet IP



6LoWPAN

Routage

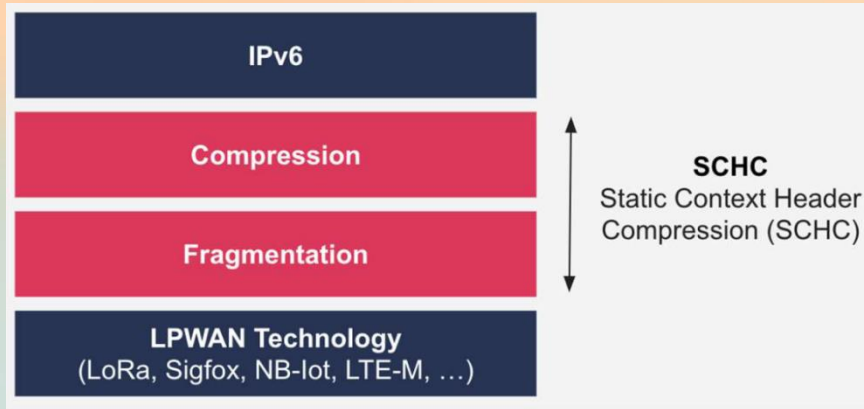
- La spécification RFC 49443 définit le mécanisme de compression des en-têtes IPv6 pour les réseaux LowPAN.
- L'utilisation de l'algorithme de compression LOWPAN_IPHC est recommandée par le groupe 6LoWPAN.
- L'en-tête IPv6s octets IPHC, résultante de la compression, intègre les informations :
 - de qualité de service,
 - des prochains en-têtes,
 - le nombre de sauts, et
 - les adresses source/destination compressées.

SCHC

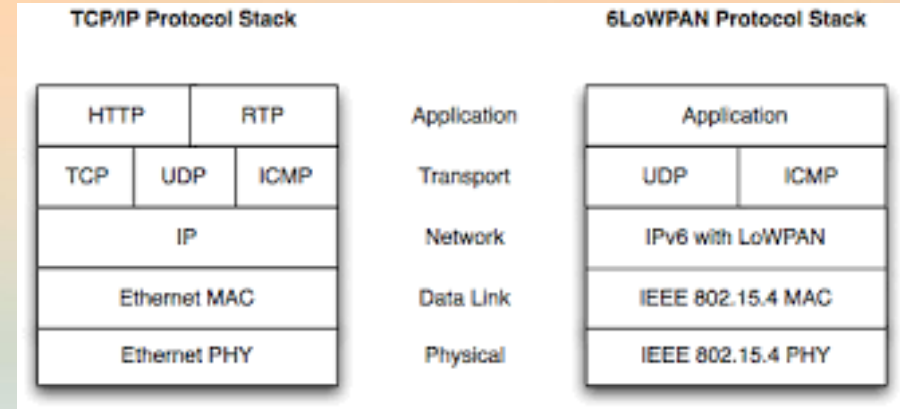
SCHC (prononcer chic) pour *Static Context Header Compression* [[RFC 8724](#)] va imposer des règles décrivant l'en-tête du message et va envoyer le numéro de la règle en remplacement de l'en-tête.

La compression est beaucoup plus importante et peut porter sur plusieurs couches protocolaires. Cependant, pour la mettre en œuvre, il faut avoir une idée des flux qui vont circuler sur le réseau. SCHC est spécifié pour les réseaux en étoile et plus particulièrement les LPWAN.

Résumé : Couche d'adaptation



La norme IETF "**SCHC** for CoAP" concilie interopérabilité, sécurité et efficacité énergétique
RFC 8724



6LoWPAN : IPv6 Low power Wireless Personal Area N
Défini les mécanismes d'encapsulation et de compression d'entêtes permettant aux paquets IPv6 d'être envoyés ou reçus via le protocole IEEE 802.15.4

Protocoles IoT- partie 2

Application Protocol		DDS	CoAP	AMQP	MQTT	MQTT-NS	XMPP	HTTP REST
Service Discovery		mDNS			DNS-SD			
Infrastructure Protocols	Routing Protocol	RPL						
	Network Layer	6LoWPAN				IPv4/IPv6		
	Link Layer	IEEE 802.15.4						
	Physical/ Device Layer	LTE-A	EPCglobal		IEEE 802.15.4		Z-Wave	
Influential Protocols		IEEE 1888.3, IPsec				IEEE 1905.1		

Protocole de routage RPL

Un des enjeux de l'IoT est le routage des paquets IP. Les objets ont des ressources électriques limitées et sont souvent connectés par des liens radio de qualité médiocre.

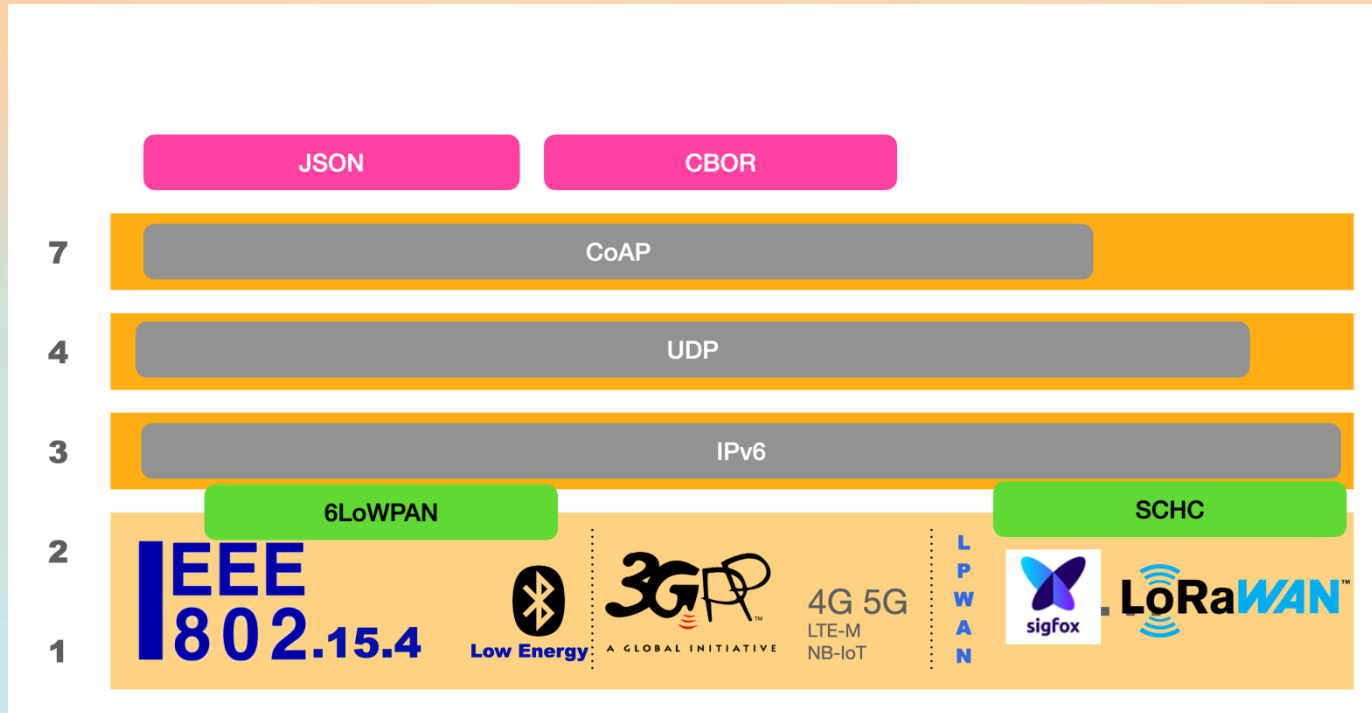
- Les protocoles de routage traditionnels ne sont pas très adaptés à cette situation.
- Le groupe de travail ROLL de l'IETF a produit un protocole «officiel», RPL (Routing Protocol for LLNs (où un LLN est un Low power and Lossy Network), un réseau où mêmes les routeurs ont peu de courant et où pas mal de paquets se perdent en route).

Protocole de routage RPL

RPL est un protocole de routage (routing), c'est-à-dire de construction de routes. Il utilise l'algorithme Trickle (basé sur la théorie des graphes pour distribuer l'information sur les routes et les routeurs.

- Pour optimiser les routes, RPL est paramétré avec une fonction nommée OF (pour Objective Function).
- Des réseaux différents peuvent utiliser des OF différents (une qui cherche le chemin le plus court, une qui cherche à ne pas utiliser comme routeur les machines n'ayant pas de connexion au courant électrique, etc), même s'ils utilisent tous RPL.

Architecture de l'IoT



Matériel et Logiciel

■ Matériel

Le matériel étant purement composé de composantes électroniques (ou électriques) on peut le séparer en 3 groupes:

- les capteurs et actionneurs
- **le microcontrôleur et le micro-ordinateur**
- l'émetteur et le récepteur (réseau de communication)

Matériel et Logiciel



Arduino + shield



Raspberry Pi



BeagleBone

Matériel et Logiciel

- **Logiciel**

La partie logiciel concerne principalement le système d'exploitation ...

- **Plateforme IOT**

Une plateforme IoT est un logiciel gérant la connexion et le pilotage *des objets connectés* dans un cloud pour collecter, stocker, corrélérer, analyser et exploiter leurs données, mais aussi pour assurer leur management.

The Raspberry Pi Platform

Objectifs de la section

1. Comprendre les composants de la carte et leurs fonctions
2. Installez la distribution Linux Raspbian
3. Configurer Raspbian pour la Raspberry Pi
4. Configurer les options de démarrage pour Raspbian

Prise en main de la RPi

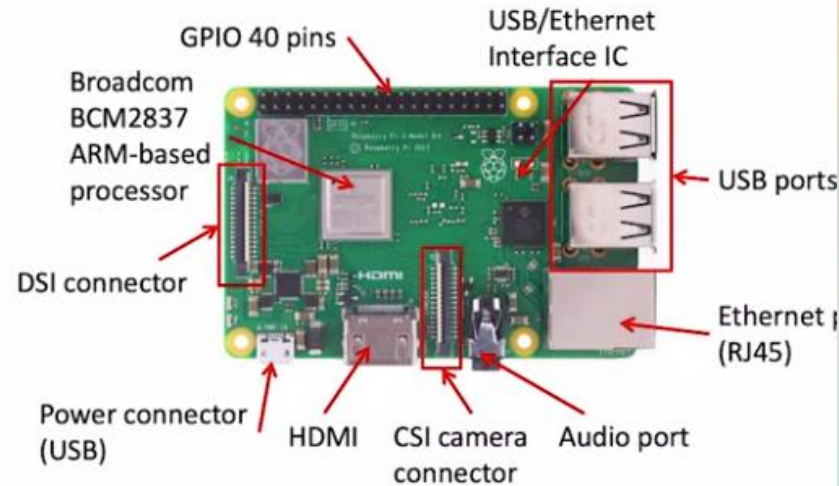
Une Raspberry PI est un véritable ordinateur « complet » :

Il a un processeur, de la mémoire et un espace de stockage. On peut l'utiliser comme ordinateur de bureau grâce à son connecteur HDMI qui lui permet d'afficher sur un écran et à ses ports USB sur lesquels on peut connecter clavier, souris et tout autre périphérique.

On peut aussi l'utiliser à distance comme un serveur de terminaux en affichant l'interface graphique sur un autre ordinateur (grâce au port HDMI) et on peut enfin l'utiliser comme un serveur sans écran et interagir en ligne de commande à distance.

Prise en main de la RPi

Raspberry Pi 3 B+ Board

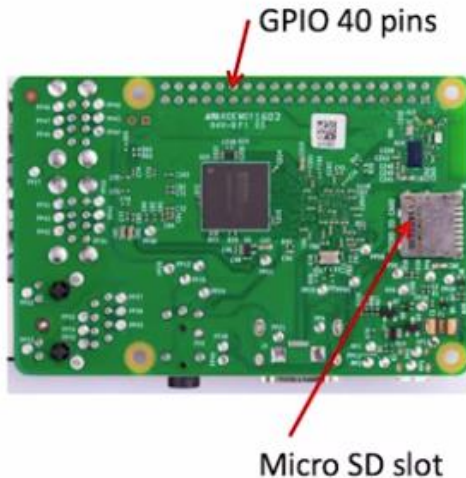


Raspberry Pi 3 B+ Hardware

- Broadcom BCM2837 SoC
 - 1.4GHz, 1Gb SRAM
- Quad Core
- 40 GPIO pins
- 4 USB ports
 - only 2 on the Raspberry Pi B
- Micro SD card slot
 - full-sized SD card slot on previous Raspberry Pis
 - Bluetooth 4.2 / BLE
 - Integrated Wi-Fi
 - Power over Ethernet (PoE)

Prise en main de la RPi

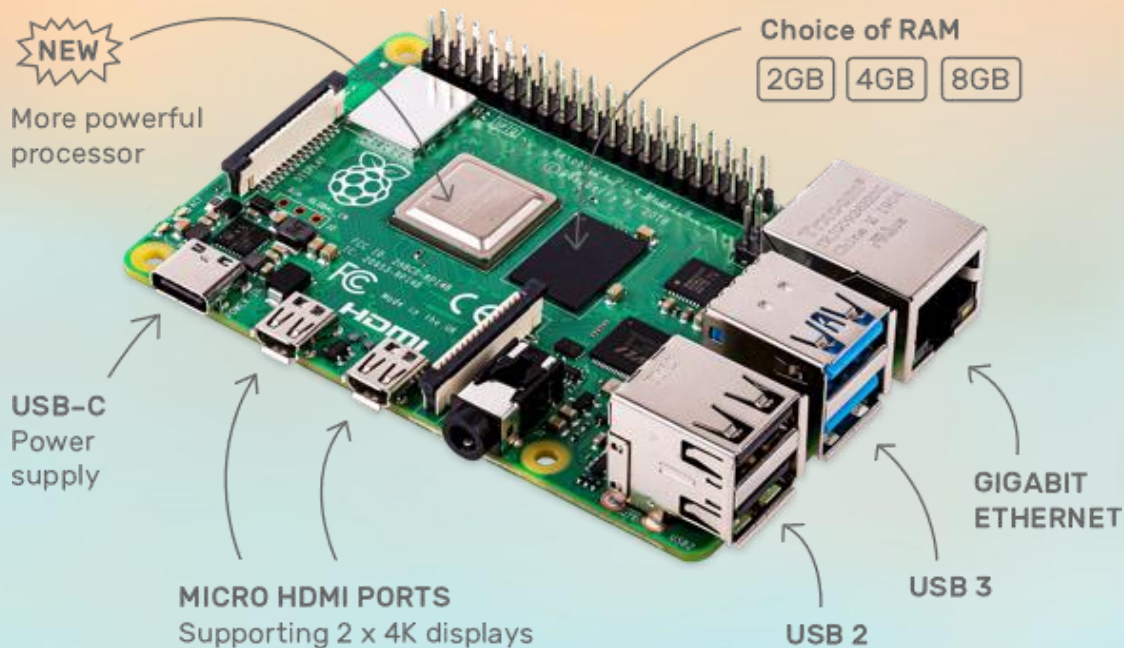
Raspberry Pi 3 B+, Backside



Raspberry Pi 3 B+ Hardware

- Broadcom BCM2837 SoC
 - 1.4GHz, 1Gb SRAM
- Quad Core
- 40 GPIO pins
- 4 USB ports
 - only 2 on the Raspberry Pi B
- Micro SD card slot
 - full-sized SD card slot on previous Raspberry Pis
- Bluetooth 4.2 / BLE
- Integrated Wi-Fi
- Power over Ethernet (PoE)

Prise en main de la RPi



Prise en main de la RPi

	Raspberry Pi 3 : modèle B	Raspberry Pi 2 : modèle B
CPU	Cortex-A53 - 1.2 GHz - quad-core	Cortex A7 - 900 MHz - quad-core
RAM	1 Go	1 Go
Vidéo	BCM2837	BCM2836
Sortie vidéo	HDMI (avec support du son 5.1) et vidéo composite	
USB	4 ports USB	
Entrées/Sorties	40 GPIO	40 GPIO
Ethernet	10/100 Mbit/s	
WiFi	802.11n	
Bluetooth	4.1	
Stockage	MicroSDHC	
Alimentation recommandée	Micro-USB ou GPIO (5V, 2.5 A)	Micro-USB ou GPIO (5V, 1.8 A)

Prise en main de la RPi

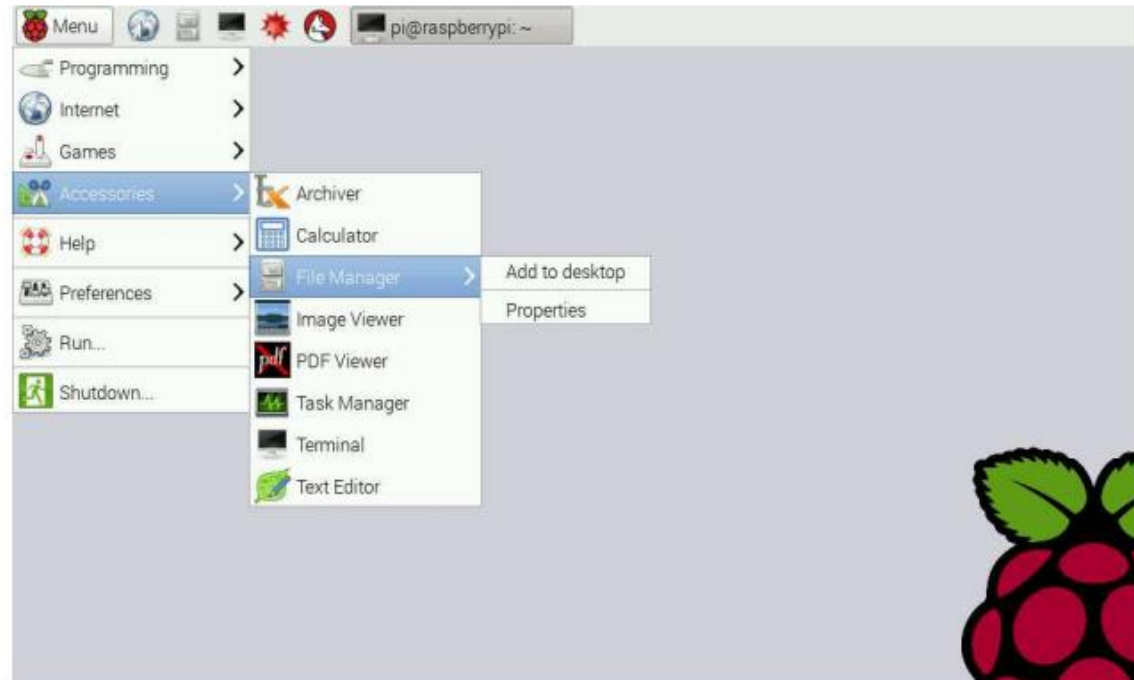
Raspberry Pi 4B vs 3B+



Features/Specs	Raspberry Pi 4 Model B	Raspberry Pi 3 Model B+
Release Date	24th June 2019	14th March 2018
SoC Type (Processor)	Broadcom BCM2711 (with metal cover)	Broadcom BCM2837B0 (with metal cover)
Core Type	Cortex-A72 64-bit (ARMv8)	Cortex-A53 64-bit (ARMv8)
No. of Cores	Quad-Core	
GPU	VideoCore VI	VideoCore IV
Multimedia	H.265 decode (4Kp60) H.264 decode (1080p60) H.264 encode (1080p30) OpenGL ES 1.1, 2.0, 3.0 Graphics	H.264, MPEG-4 decode (1080p30) H.264 encode (1080p30) OpenGL ES 1.1, 2.0 Graphics
CPU Clock	1.5 GHz	1.4 GHz
Memory/OS storage	microSD	
RAM	LPDDR4: 1GB, 2GB, 4GB and 8GB options	LPDDR2 1GB
Ethernet	True Gigabit Ethernet	Gigabit over USB 2.0 (Max 300Mbps)
USB Port	2 x USB 3.0 + 2 x USB 2.0	4 x USB 2.0
HDMI	2 x micro HDMI support Dual Display	1 x full size HDMI
WiFi	802.11 b/g/n/ac (2.4GHz+5GHz & Shielded)	
Bluetooth	5.0 + BLE (Shielded)	4.2 + BLE (Shielded)
Antenna	PCB Antenna (Similar to Rpi Zero W)	
GPIO	40 pins (Fully backwards-compatible with previous boards)	
Operating System	Raspbian (> 24 June 2019)	Raspbian (> March 2018)
Dimension	85mm x 56mm	
Power Input	5V via USB Type C (upto 3A) 5V via GPIO header (upto 3A) Power over Ethernet, requires PoE HAT	5V via USB Micro B (upto 2.5A) 5V via GPIO header (upto 3A) Power over Ethernet, requires PoE HAT

Prise en main de la RPi

Système d'exploitation : Raspbian

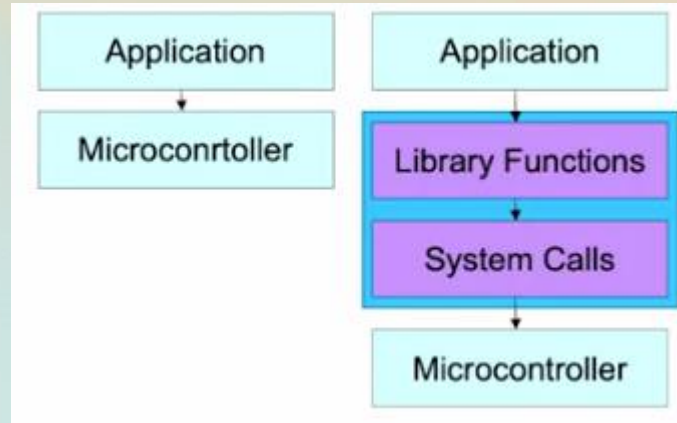


Raspberry Pi vs. Arduino

- Raspberry Pi processor is faster
 - 1.4GHz vs 16MHz
- 64-bit processor vs. 8-bit
 - Bigger address space, number representations
- Raspberry Pi has more memory
 - Arduino: 32K Flash, 2K SRAM, 1K EEPROM
 - Raspberry Pi: 1Gb SRAM, micro SD
- Raspberry Pi has lower I/O voltage levels
 - 3.3 V vs. 5 V

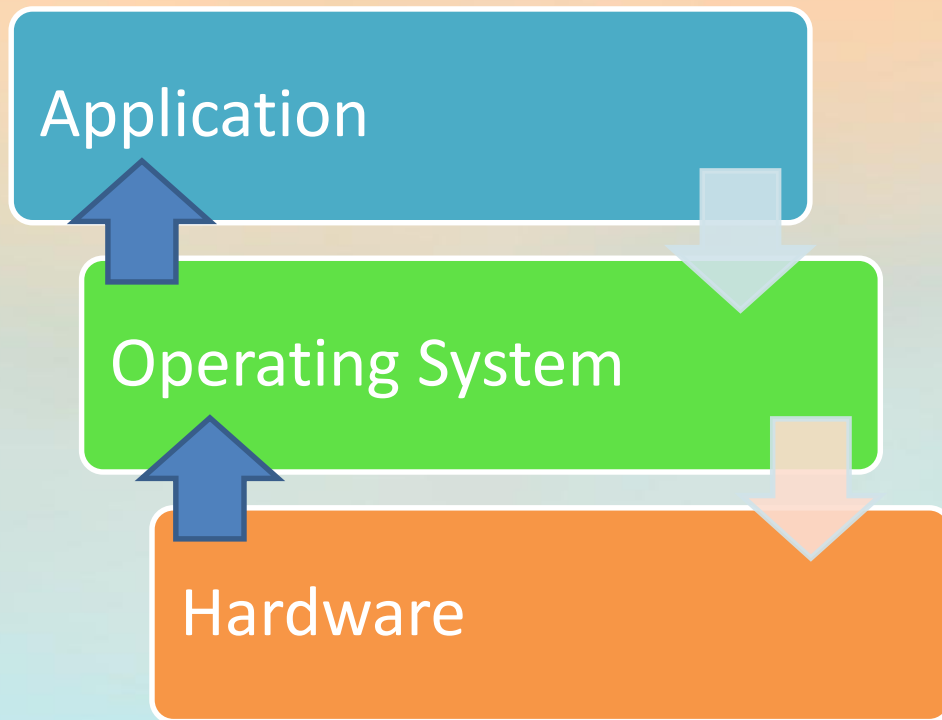
Utilisation de système d'exploitation

RPI peut supporter un système d'exploitation et donc permettre beaucoup de fonctionnalités.



Rappel sur les systèmes d'exploitation

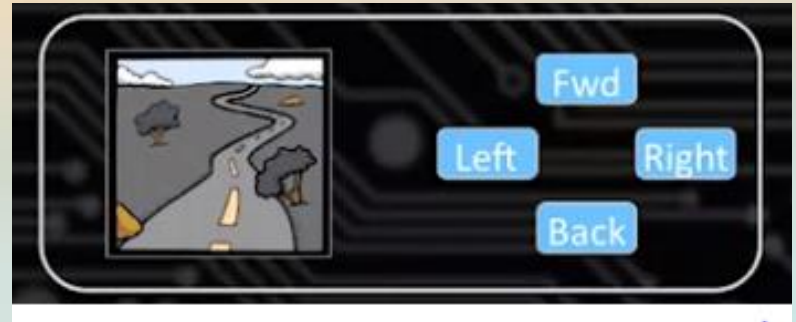
- Support pour les processus.
- Le process est une instantiation d'un programme.
- Gestion équitable des ressources (CPU/ Mémoire/ périphériques/ Horloges/ Réseau...)



Rappel sur les systèmes d'exploitation

Tâches multiples

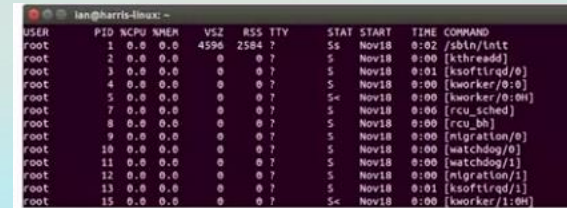
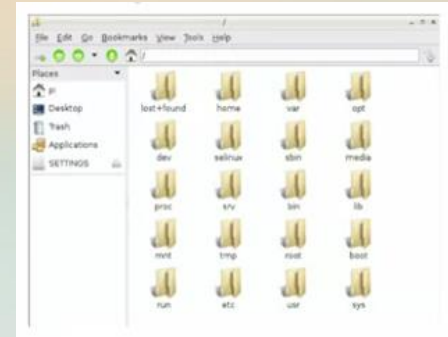
- Envoie de données vidéos.
- Service Mouvement selon les boutons touchés.
- Détection d'obstacle.
- Auto-freinage.



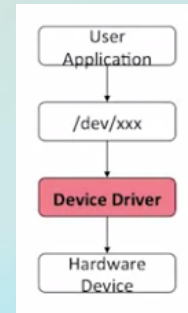
Utilisation de système d'exploitation

Quels sont les autres avantages d'un système d'exploitation?

- Interface graphique.
- Système de fichiers graphique.
- Multi-processes.
- Facilité d'usage de différents hardware (Drivers).



USER	PID	NCP	NMEN	VSZ	RSS	ITY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	4596	2584	7	Ss	Nov18 0:02	/sbin/init	
root	2	0.0	0.0	0	0	7	S	Nov18 0:00	[kthreadd]	
root	3	0.0	0.0	0	0	7	S	Nov18 0:01	[ksoftirqd/0]	
root	4	0.0	0.0	0	0	7	S	Nov18 0:00	[kworker/0:0]	
root	5	0.0	0.0	0	0	7	Ss	Nov18 0:00	[kworker/0:0H]	
root	7	0.0	0.0	0	0	7	S	Nov18 0:00	[rcu_sched]	
root	8	0.0	0.0	0	0	7	S	Nov18 0:00	[rcu_bh]	
root	9	0.0	0.0	0	0	7	S	Nov18 0:00	[migration/0]	
root	10	0.0	0.0	0	0	7	S	Nov18 0:00	[watchdog/0]	
root	11	0.0	0.0	0	0	7	S	Nov18 0:00	[watchdog/1]	
root	12	0.0	0.0	0	0	7	S	Nov18 0:00	[migration/1]	
root	13	0.0	0.0	0	0	7	S	Nov18 0:01	[ksoftirqd/1]	
root	15	0.0	0.0	0	0	7	Ss	Nov18 0:00	[kworker/1:0H]	



Est ce que la RPI est un dispositif IoT?

Peut être ça dépend de la manière dont il est utilisé

Similarities

- Network connectivity and computational intelligence
- Small and cheap (relative to a PC)
- Can interface directly with sensors/actuators via pins

Differences

- Interface can be exactly the same as a PC running Linux
 - Complexities of the system can be visible

Prise en main de la RPi

Quoi faire au début?

Step 1: Plug in a monitor (via HDMI) and a keyboard and mouse (via USB)

- Need an interface to the device

Step 2: Get an operating system

- Raspberry Pi needs an operating system
- Operating system image must be present on the micro SD card

Installation de l'OS

Use **New Out-Of-Box Software (NOOBS)**

- Comes preinstalled on micro SD bundled with Raspberry Pi boards
- Otherwise, download it for free from www.raspberrypi.org/downloads
- If NOOBS is not preinstalled on micro SD, you'll need to:
 1. Format the micro SD (need an SD reader)
 2. Extract the NOOBS download
 3. Put it on the micro SD



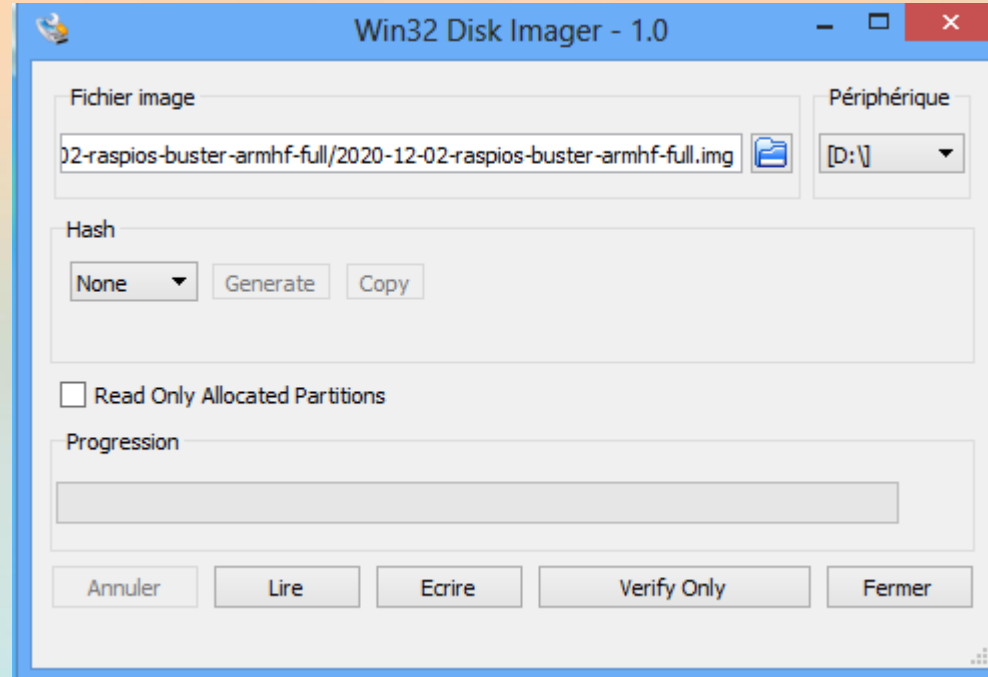
Préparation de la carte

- Votre carte SD est ce qui va servir de mémoire à votre Raspberry Pi : elle stockera absolument toutes les informations, à savoir : les fichiers textes, audios, photos, vidéos, etc., mais aussi le système d'exploitation.
- Raspbian est un système d'exploitation GNU/Linux spécialement conçu et optimisé pour la Raspberry Pi.
- Télécharger l'OS
- Formater la SD card

Set up de la Raspberry

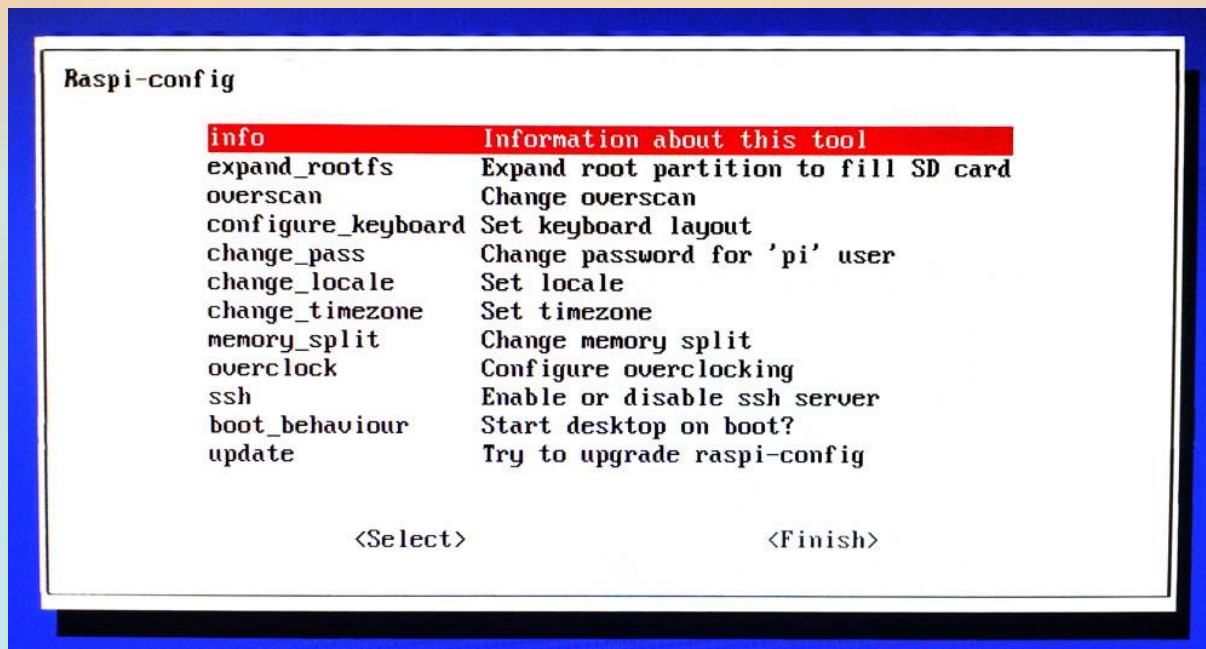
Installation de Raspbian :

- Télécharger l'image de Raspbian.
- Préparation de la carte :
 - Télécharger Win32DiskImager, installez-le, et lancez-le.
 - Sélectionner l'image de Raspbian et sélectionner la lettre du lecteur correspondant à votre carte SD
 - Cliquez sur Write.



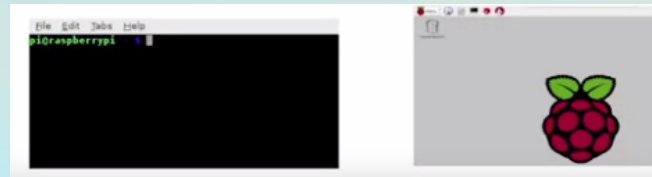
Set up de la Raspberry

On arrive sur ceci : (Pour y revenir plus tard on tape dans " sudo raspi-config ")



Set up de la Raspberry

- **expand_rootfs** : Va permettre d'utiliser tout l'espace de la carte SD
- **configure_keyboard** : Pour mettre le clavier en français on Sélectionne Generic 105-key (intl) PC puis Other puis French puis French puis The default for the keyboard layout puis No compose key puis No
- **change_pass** : modifier le mot de passe de l'utilisateur 'pi' (par défaut : raspberry)
- **change_locale** : changer la langue du système
- **change_timezone** : changer l'heure locale
- **ssh** : activer le SSH (Indispensable selon mon point de vue)
- **boot_behavior** : Pour choisir si on veut arriver directement sur le bureau au démarrage du système et non pas sur une console.



Set up de la Raspberry

Premier démarrage – connexion sans écran ni clavier

- Activation du mode SSH

Par défaut, et ce depuis l'attaque de masse en novembre 2016 sur les objets connectés, la fondation Raspberry a désactivé l'accès SSH. Cependant, il existe une astuce qui active le SSH lors du premier démarrage. Pour cela, avant de booter avec votre Raspberry, créez depuis un autre ordinateur un fichier se nommant ssh à la racine de votre carte SD (dans la partition Boot). Inutile de donner une extension au fichier, ni d'y ajouter du contenu. Un fichier vide se nommant tout simplement ssh suffit à activer le mode SSH lors du premier boot. Vous aurez donc accès à distance à votre installation toute fraîche.

Set up de la Raspberry

Premier démarrage – connexion sans écran ni clavier

- Configuration du WI-FI

Afin de configurer la connexion Wi-fi lors du démarrage de la Pi, nous allons créer un fichier `wpa_supplicant.conf`, situé à la racine de la partition boot de la carte.

Pour cela, nous allons utiliser l'application bloc-note. Le fichier devra contenir les lignes suivantes

```
country=fr
update_config=1
ctrl_interface=/var/run/wpa_supplicant
network={
scan_ssid=1
ssid="Nom de votre identifiant SSID : votre réseau wi-fi "
psk=" votre mot de passe »
}
```

Set up de la Raspberry

- Mettre en place une ip statique pour votre Raspberry Pi

Pour cette étape, vous devrez pouvoir accéder à la partition système de la carte, et non plus la partition boot. Par conséquent, vous aurez besoin d'un ordinateur faisant tourner Linux ou Mac OS X. Si vous n'avez qu'un ordinateur Windows, branchez simplement la Raspberry Pi, retrouvez son IP sur le réseau (par exemple depuis l'interface de gestion de votre box, et effectuez la suite des opérations directement sur la Raspberry Pi en SSH.

Set up de la Raspberry

- Mettre en place une ip statique pour votre Raspberry Pi

Dans le but de connaître l'ip que prendra la Raspberry Pi, nous allons lui donner une ip statique. Pour cela nous allons modifier **le fichier dhcpd.conf** situé dans le **dossier /etc/**.

Pour modifier le fichier, utilisez le même logiciel que pour configurer le Wi-Fi.

Une fois dans le fichier, allez à la dernière ligne et ajoutez le contenu suivant:

```
interface eth0
static ip_address=192.168.1.100/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1

interface wlan0
static ip_address=192.168.0.100/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Set up de la Raspberry

```
interface eth0
static ip_address=192.168.1.100/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1

interface wlan0
static ip_address=192.168.0.100/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

interface eth0 correspond à **une connexion de type filaire** et *interface wlan0* à **une connexion Wi-fi**. A vous donc de choisir celui qui correspond à votre setup.

Static ip_address permet d'indiquer l'ip que votre Raspberry Pi aura une fois démarrée. Généralement l'ip est de type *192.168.1.x*, remplacez le x par la valeur de votre choix, **attention à ne pas rentrer en conflit avec d'autres devices**.

Pour ce qui est de *static router* et *static domain_name_servers*, il s'agit généralement de l'ip de votre box, soit *192.168.1.1*.

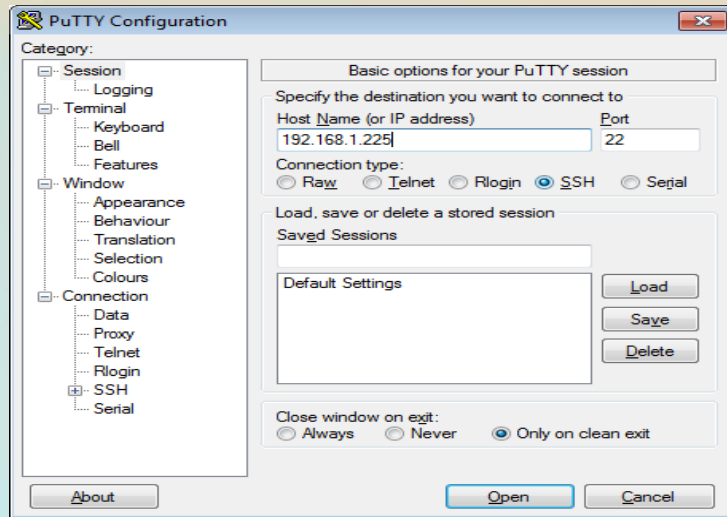
Vous n'avez plus qu'à enregistrer le fichier.

Set up de la Raspberry

La carte SD est maintenant prête à être insérée dans la Raspberry Pi et à la **démarrer sans écran et sans clavier**. Vous n'avez plus qu'à vous y connecter en SSH afin de finaliser la configuration de la Raspberry Pi (changement des mots de passe, étendre la carte SD, etc).

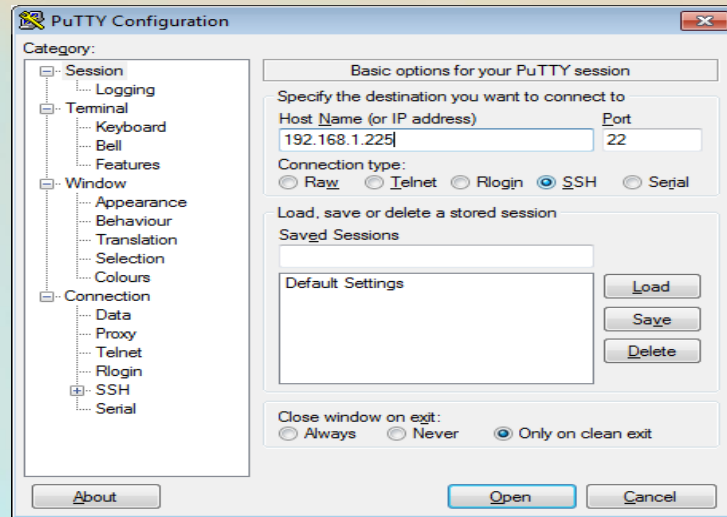
Set up de la Raspberry

- Connexion Putty (Korben pour Mac)
- On branche le câble d'alimentation et le câble Ethernet.
- Pour la suite on va se mettre sur un autre ordinateur, peu importe mais sur le même réseau , on télécharge putty, puis on le lance!



Set up de la Raspberry

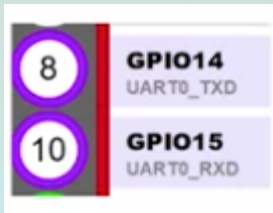
- Par défaut, l'utilisateur pi avec mot de passe raspberry est fourni pour chaque Raspberry Pi. Connectez-vous en utilisant ces données.
- login as: pi
- pi@192.168.1.225's password: raspberry





















General Purpose IO Pins

Pins dédiés pour
l'alimentation
3,3V (1,17), 5V (2,4)
Gnd (6,9,14,20,30,39)

UART Pins



Raspberry Pi2 GPIO Header					
Pin#	NAME		NAME	Pin#	
01	3.3v DC Power		DC Power 5v	02	
03	GPIO02 (SDA1, I2C)		DC Power 5v	04	
05	GPIO03 (SCL1, I2C)		Ground	06	
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08	
09	Ground		(RXD0) GPIO15	10	
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12	
13	GPIO27 (GPIO_GEN2)		Ground	14	
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16	
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18	
19	GPIO10 (SPI_MOSI)		Ground	20	
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22	
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24	
25	Ground		(SPI_CE1_N) GPIO07	26	
<hr/>					
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28	
29	GPIO05		Ground	30	
31	GPIO06		GPIO12	32	
33	GPIO13		Ground	34	
35	GPIO19		GPIO16	36	
37	GPIO26		GPIO20	38	
39	Ground		GPIO21	40	

Rev 1
26/01/2014

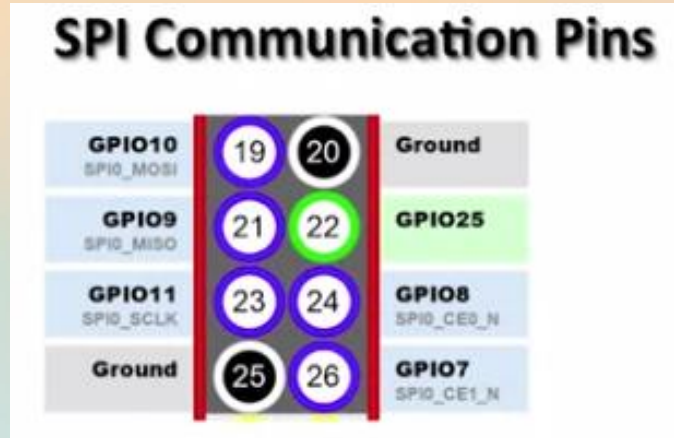
<http://www.element14.com>

General Purpose



Multi-Function
Extra label

General Purpose IO Pins



GPIO Access in Python

- Use the GPIO library
- `import Rpi.GPIO as GPIO`
- Execute your Python program (script) as root
 - Use “sudo” for this

Selecting Pin Numbering Mode

Pin Numbering Modes



Two ways to refer to the pins

1. The number of the pins in their order on the board
 - Shown in circles in the picture
2. The Broadcom SoC number
 - Shown in rectangles, after “GPIO”

Selecting Pin Numbering Mode

```
GPIO.setmode(GPIO.BOARD)
```

- Use board numbering

```
GPIO.setmode(GPIO.BCM)
```

- Use Broadcom SoC numbering
- Changes with different versions of Raspberry Pi

Pin Direction and Assignment

```
GPIO.setup(13, GPIO.OUT)
```

- Set the pin direction
- Like `pinMode(13, OUTPUT)` for Arduino

```
GPIO.output(13, True)
```

- Assign value to output pin
- Like `digitalWrite(11, HIGH)` for Arduino

Blink an LED

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(13, GPIO.OUT)
while True:
    GPIO.output(13, True)
    time.sleep(1)
    GPIO.output(13, False)
    time.sleep(1)
```

Reading Input Pins

```
GPIO.setup(13, GPIO.IN)
```

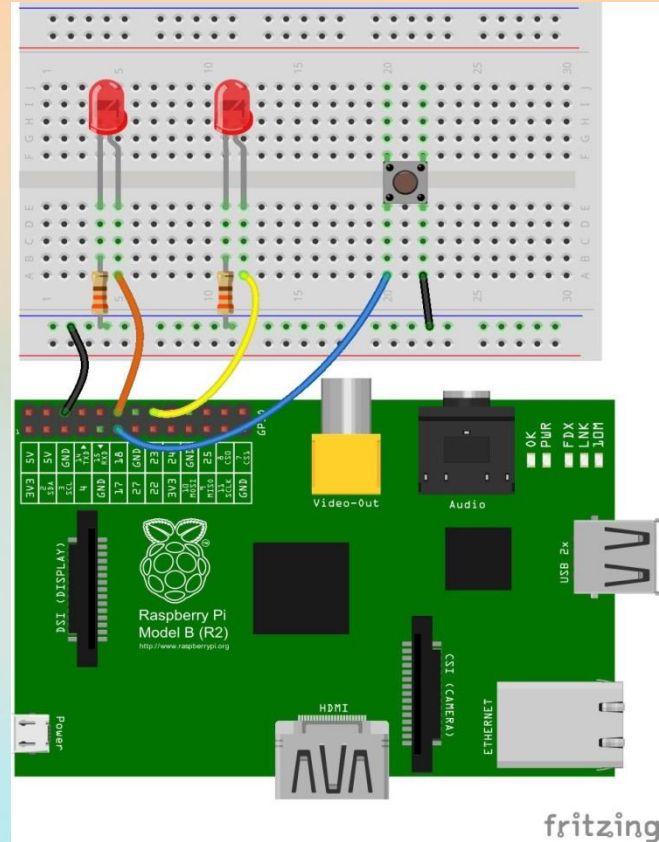
- Set the pin direction to an input

```
value = GPIO.input(13)
```

- Read value on input pin
- Only reads digital inputs
 - No analogRead equivalent
 - No analog-to-digital converter

Schéma

Sur <https://www.tinkercad.com/>



Liens utiles

Installing Raspbian with NOOBS

<https://projects.raspberrypi.org/en/projects/noobs-install>

Raspberry GPIO

<https://learn.sparkfun.com/tutorials/raspberry-gpio>

FIN