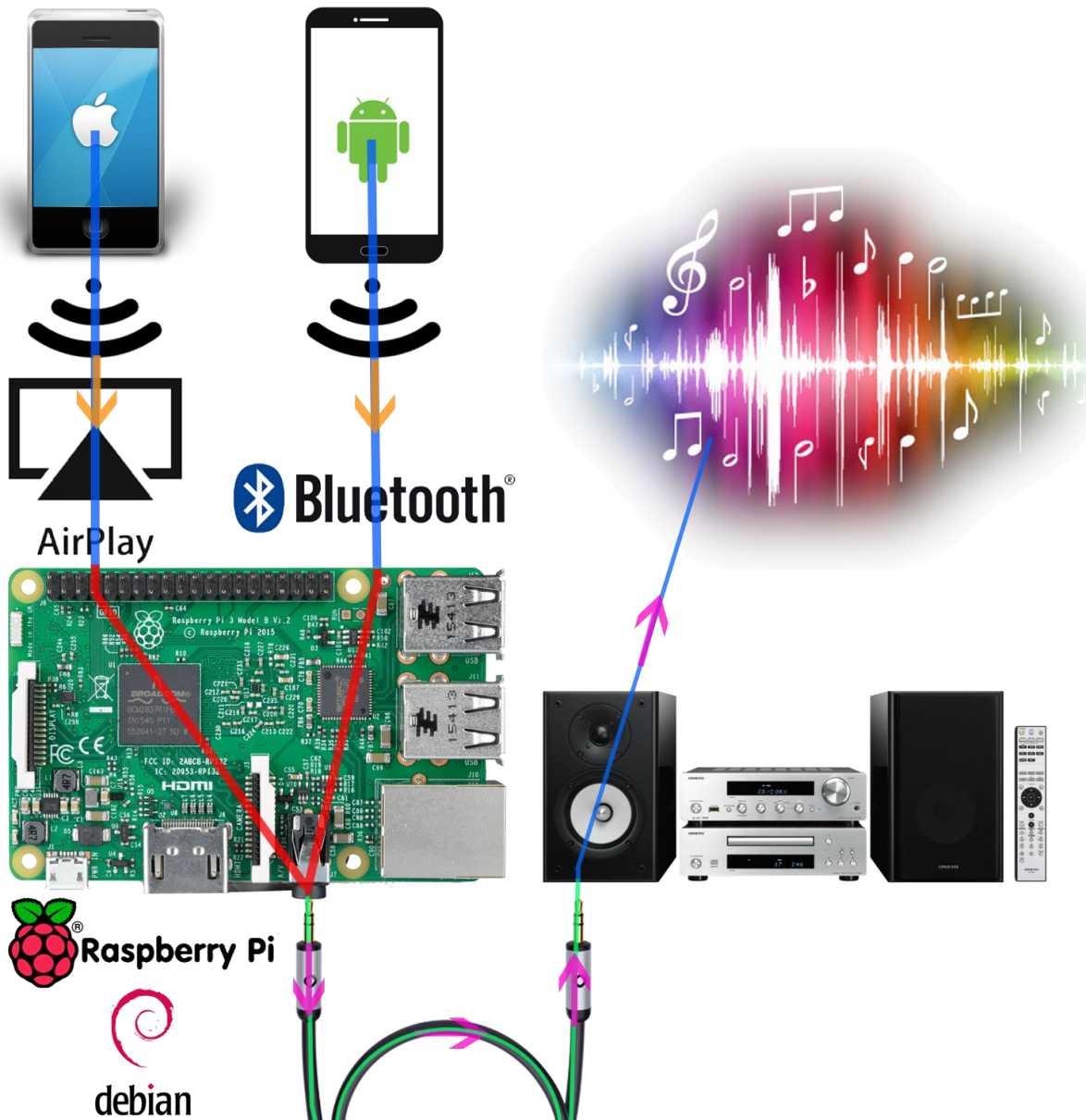


Projet Tutoré : Streaming Hi-Fi

ETUDE ET DEVELOPPEMENT D'UN APPAREIL D'ECOUTE DE MUSIQUE DE SALON



BONHOMME Benjamin

M4101 : PT, Projet de spécialité de dimension professionnelle

RICHER Benjamin

28/03/2017

DUT Réseaux & Télécommunications

Sommaire

Sommaire	1
Remerciements	4
1. Introduction et présentation du sujet.....	5
2. Présentation du matériel	10
2.1 Liste du matériel utilisé	10
2.2 Spécification du Raspberry	11
2.3 Tests comparatifs entre Pi B+ et Pi 3.....	13
3. Installation/configuration du Raspberry et des différents paquets	15
3.1 Choix de la distribution.....	15
3.2 Première installation et configuration de Raspbian.....	17
3.3 Les différents paquets utilisés.	18
4. Etude des protocoles Airplay (Shairport) et Bluetooth.....	20
4.1 Comparatifs entre Airplay et Bluetooth	20
5. Interface WEB (PHP).....	22
5.1 Installation paquets	22
5.1.1 Apache2.....	22
5.1.2 Php5.....	23
5.2 Les différents langages	24
5.2.1 HTML	24
5.2.2 CSS	24
5.2.2.1 Choix du Template	25
5.2.2.2 Compréhension et adaptation au Template final.....	26
5.2.3 PHP	27
5.2.3.1 Apprentissage	28
5.2.3.2 Phase de test.....	28
5.2.4 JavaScript, peu mais pourquoi	28
5.2.5 La CGI PHP/Bash.....	29
5.2.6 Mod_Python ?.....	30
5.3 Problèmes rencontrés	31
5.3.1 Droits d'accès et permissions entre root et www-data.....	31

5.3.2	Problème avec le reboot et PHP	31
5.4	Outil et méthodologie de développement.....	32
5.4.1	MobaXTerm, logiciel ssh et sftp	32
5.4.2	Notepad++ et nano	32
5.4.3	Inspecteur d'élément de Firefox	33
5.5	Sécurité.....	35
5.5.1	.htaccess	35
6.	Etude de la qualité sonore et de la consommation électrique.....	36
6.1	Qualité audio	36
6.2	Consommation électrique	39
7.	Etude marketing du produit dans l'optique d'un lancement dans le commerce	40
8.	Conclusion	42
	Annexes	43
	Annexe 1 : Présentation détaillée du matériel.	43
	Matériel fourni par l'IUT.....	43
	Matériel fourni par nos soins (acheté ou déjà en possession).....	46
	Annexe 2 : Test de Benchmarking effectués sur les 2 Raspberry (Pi B+ et Pi 3).....	49
	Script de Benchmarking utilisé.....	49
	Résultats bruts du test de Benchmarking	50
	Quelques caractéristiques.....	51
	Tests de connexion.....	52
	Test de processeur (CPU)	53
	Test de Multi-Threads	54
	Test de mémoire (RAM)	55
	Test sur la vitesse lecture/écriture de la carte MicroSD	56
	Annexe 3 : Manipulations première installation et configuration de Raspbian	57
	Mise en place matérielle	57
	Configuration de Raspbian	63
	Annexe 4 : Manipulations pour installer les différents paquets et applications.	73
	Installation et configuration de Airplay/Bluetooth.....	73
	Installation et configuration de hostapd et dnsmasq.....	79
	Annexe 5 : Etude détaillée des protocoles Airplay et Bluetooth	86

Airplay.....	86
Protocole RTSP	86
Protocole RTP	89
Bluetooth.....	91
Annexe 6 : Qualité audio dans les systèmes audio en réseau	95
Qualité audio	95
Systèmes audio en réseau.....	99
Annexe 7 : Interface Web - Partie programmation.....	103
Script.sh	103
Reboot.php.....	103
log.php.....	105
index.php.....	107
volume.php	109
Page Principal	112
Table des illustrations	113
Glossaire	118
Index.....	121
Bibliographie & Sitographie	123

Remerciements

Tout d'abord nous tenons particulièrement à remercier M. Viennet pour son soutien, ses conseils et sa disponibilité dans l'élaboration de ce projet.

Nous remercions également Mme Paulian pour ses conseils dans l'édition et la mise en forme du document.

Nous remercions aussi nos familles et proches de leurs soutiens et pour la relecture du rapport final.

Nous apprécions la mise à disposition de logiciels par l'université tel que Microsoft Office Visio qui a permis de réaliser l'ensemble des schémas de ce rapport.

1. Introduction et présentation du sujet

Pendant le semestre 3 nous avons été amenés à former des groupes pour effectuer un projet qui s'étendrait sur les semestres 3 et 4. Initialement notre groupe était formé de Benjamin BONHOMME, Brahim LAMINE et Benjamin RICHER mais depuis la fin du semestre 3 Brahim LAMINE est parti étudier en Norvège.

Nous nous sommes orientés vers le sujet n°4 : "Streaming Hi-Fi, étude et développement d'un appareil d'écoute de musique de salon" car il nous paraissait être le plus intéressant et le fait de pouvoir travailler sur la plateforme Raspberry Pi en environnement GNU/Linux nous a séduits. En effet travailler sur un Raspberry faisait partie de nos projets personnels (Benjamin BONHOMME avait l'intention de travailler sur un Raspberry pour en faire un serveur multimédia et Benjamin RICHER un serveur Web), de plus Benjamin BONHOMME a déjà travaillé sur les ondes sonores dans le cadre du TPE en classe de terminale scientifique.

Le projet consiste à développer et étudier un appareil qui va recevoir des données audio depuis des terminaux mobiles afin de les transmettre à un dispositif de diffusion musicale (chaîne Hi-Fi ou même des enceintes) via un câble jack. Ainsi nous allons moderniser les chaînes Hi-Fi qui peuvent avoir un peu d'ancienneté.

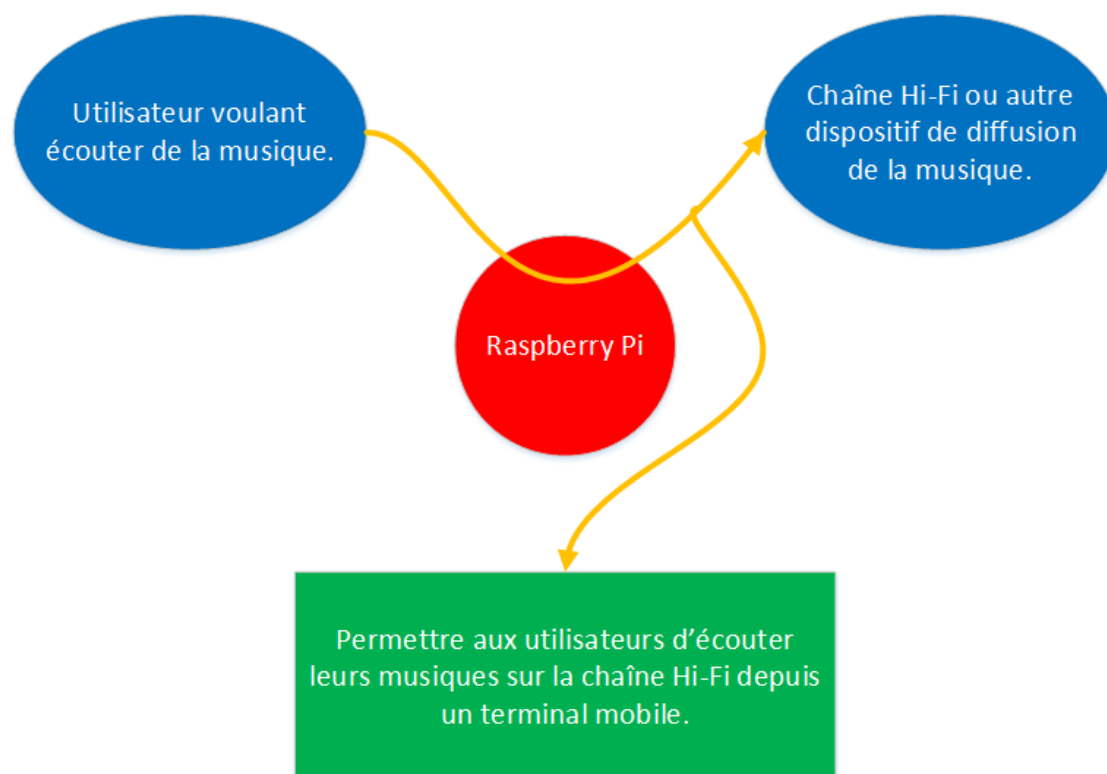


Figure 1 : Diagramme bête à cornes, permettant d'indiquer à quoi sert le produit.

Le besoin d'un tel dispositif provient du fait que de nombreux foyers sont équipés de chaînes Hi-Fi pour écouter de la musique provenant de sources analogique. Or aujourd'hui le plus souvent les utilisateurs se servent d'appareils mobiles pour stocker leurs musiques (tablettes, téléphone...). Des protocoles spécifiques comme Bluetooth, Airplay et DLNA ont été développés pour que ces appareils puissent communiquer sans connexion filaire et nous allons utiliser ces protocoles pour diffuser des flux audio à notre récepteur. On s'appuie sur la technologie sans-fil car aujourd'hui les derniers smartphone ne dispose plus d'interface Jack (iPhone 7) et il existe un certain confort d'utiliser son telephone dans son canapé sans avoir un cable jack à travers la pièce. On peut aussi préciser que de tels appareils existent déjà dans le commerce mais qu'il est très rare voir impossible de trouver un appareil qui cumule plusieurs protocoles de communication.

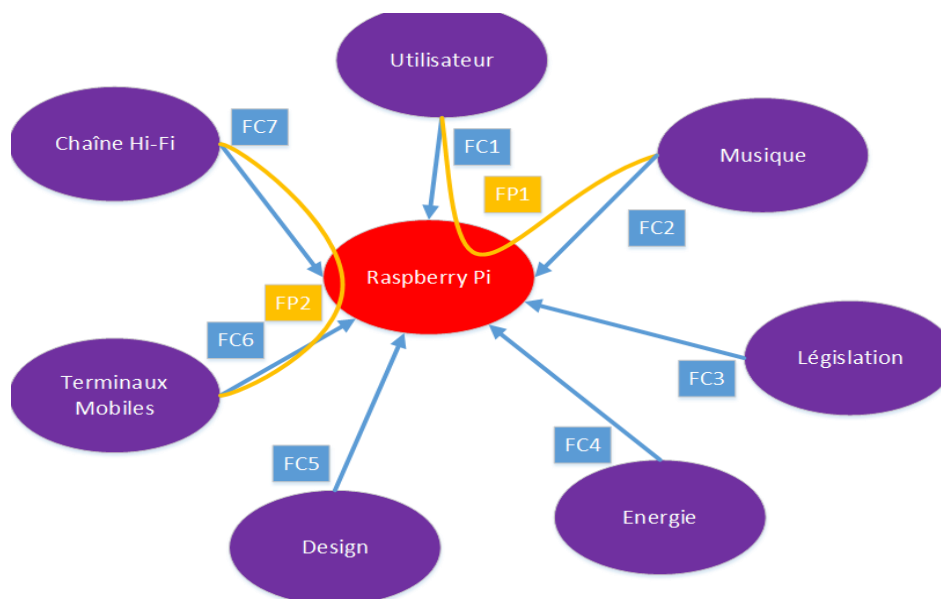


Figure 2 : Diagramme pieuvre, permet de relier le produit à son environnement extérieur (tout ce qui peut influencer sur son fonctionnement).

FP1 : L'utilisateur doit pouvoir écouter de la musique.

FP2 : Modernisation de la chaîne Hi-Fi grâce à l'utilisation des terminaux mobiles.

FC1 : Etre facile d'utilisation pour les utilisateurs.

FC2 : La musique doit être d'une bonne qualité.

FC3 : L'appareil et les logiciels utilisés doivent être libre.

FC4 : L'appareil ne doit pas trop consommer d'électricité.

FC5 : L'appareil doit être petit et discret.

FC6 : L'appareil devra utiliser des technologies sans-fil utilisées par les terminaux mobiles.

FC7 : La musique devra être transmise de l'appareil à une chaîne Hi-Fi ou un dispositif de diffusion de musique par un câble jack.

Nous avons donc un double objectif dans ce projet, tout d'abord il faut mettre en oeuvre le Raspberry de sorte à ce qu'il puisse recevoir des flux audio provenant d'une multitude d'appareils. On a donc pensés à intégrer les protocoles Bluetooth (pour les appareils Android), Airplay (pour les appareils Apple) et éventuellement DLNA pour les appareils compatibles. Bien entendu nous prendrons en compte des contraintes comme le budget (qui au minimum doit être le prix d'un Raspberry, c'est à dire d'une trentaine d'euros), l'appareil doit être facile à utiliser pour le grand public. Nous étudierons aussi des paramètres comme la qualité audio délivrée par le Raspberry Pi, plus précisément par les protocoles utilisés et la consommation électrique en utilisation et au repos pour avoir une idée du prix de sa consommation sur le long terme. Le deuxième objectif est de prévoir l'appareil en vue de sa commercialisation et de sa production à grande échelle, nous devons donc justifier son prix par le matériel utilisé, la main d'oeuvre fournie, les différentes fonctionnalités intégrées et essayer de le placer dans le marché par rapport aux concurrents.

Nous avons réfléchi à de nombreuses solutions mais qui n'ont pas pu toutes aboutir soit par le manque de temps, par la difficulté de mise en place ou le prix induit.

BONHOMME Benjamin
RICHER Benjamin

Sujet 4 - Idées d'amélioration

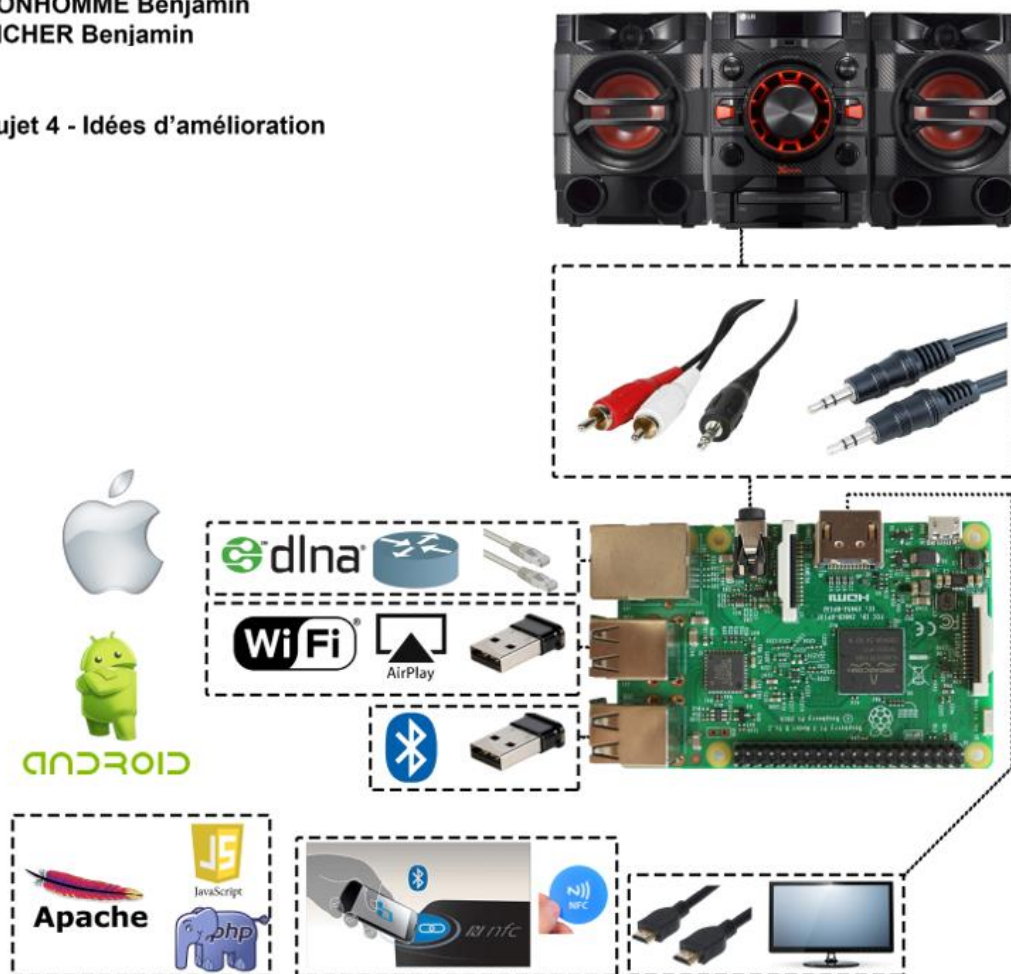


Figure 3 : Liste des idées d'améliorations que nous avons proposé.

- Ajout de la prise en charge DLNA (non réalisé car manque de temps).
- Mise en place d'une interface Web pour contrôler le son en sortie, pour avoir des informations sur différents services (Bluetooth, Shairport, DLNA, Wi-Fi). Cela nécessite les services Apache, JavaScript et Php pour pouvoir interfacier des scripts Bash au Php.
- Intégration d'un navigateur comme YoutubeTV pour écouter de la musique (non réalisé car manque de temps et peu de sources fiables pour la réalisation).
- Lire des fichiers depuis une clé USB (non réalisé car manque de temps).
- Accès au partage Windows (serveur Samba, non réalisé du au manque de temps).
- Tag NFC pour connexion Wi-Fi/Bluetooth rapide (non réalisé car cela induit des dépenses supplémentaires).
- Ajout d'un écran tactile au Raspberry (non réalisé car trop cher pour son utilité).
- Ajout d'une carte d'extension audio (non réalisé car trop cher pour son utilité et moins ergonomique).

De plus afin d'organiser ce projet nous avons réalisés des diagrammes de Gantt pour les semestres 3 et 4. Le diagramme de Gantt est utilisé pour ordonnancer et gérer les tâches avec une représentation visuelle, ainsi on peut voir l'état d'avancement du projet. Dans la partie gauche du diagramme nous avons la liste des tâches à réaliser. Il y a aussi l'échelle temporelle en haut qui s'étend de manière horizontale, comme un axe d'abscisse. Chacune des tâches est matérialisées par une barre horizontale dont la la position et la longueur représentent la date de début, la durée et la date de fin. Pour finir nous avons essayé de synthétiser au maximum ce rapport de projet, en effet tous les détails techniques et les manipulations effectuées ont été mis en annexes pour une meilleur lisibilité.

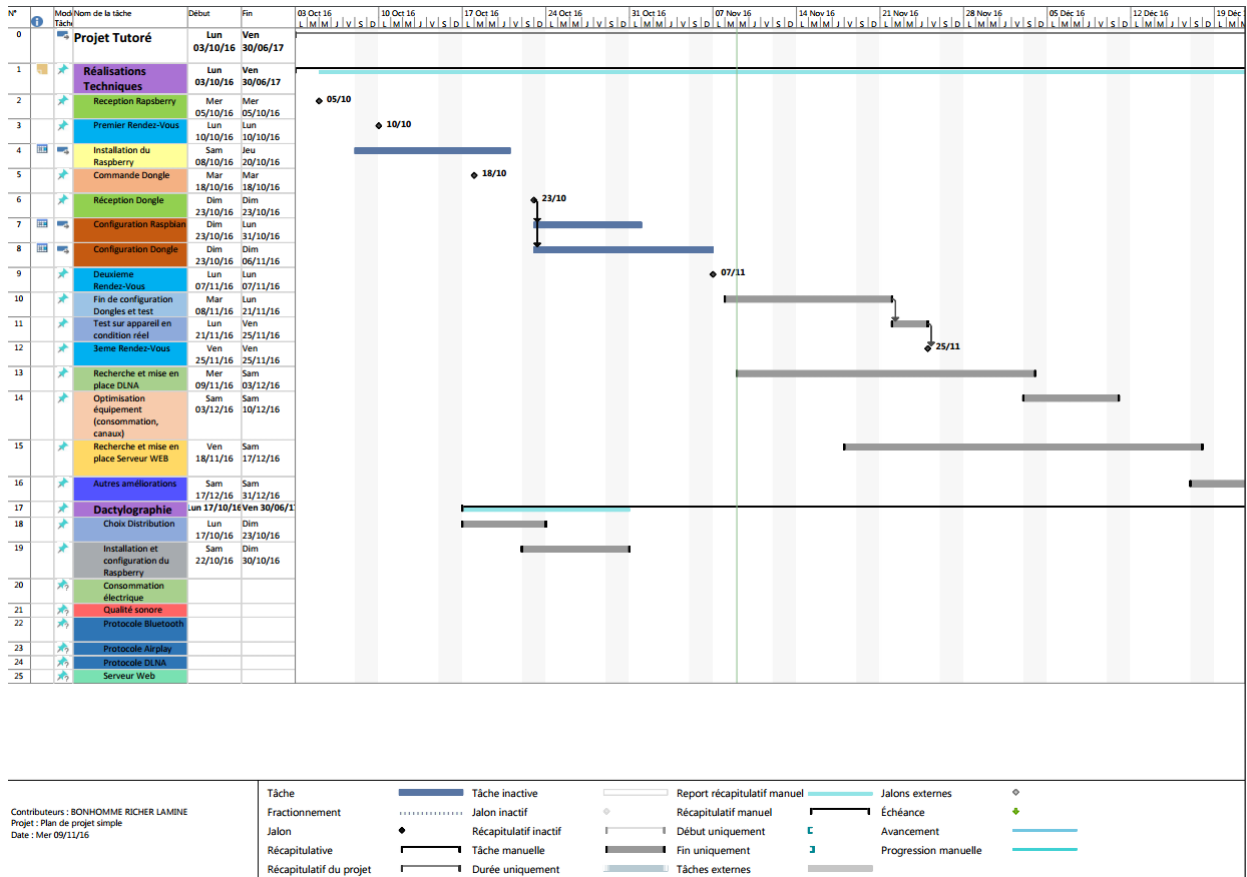


Figure 4 : Diagramme de Gantt du semestre 3.

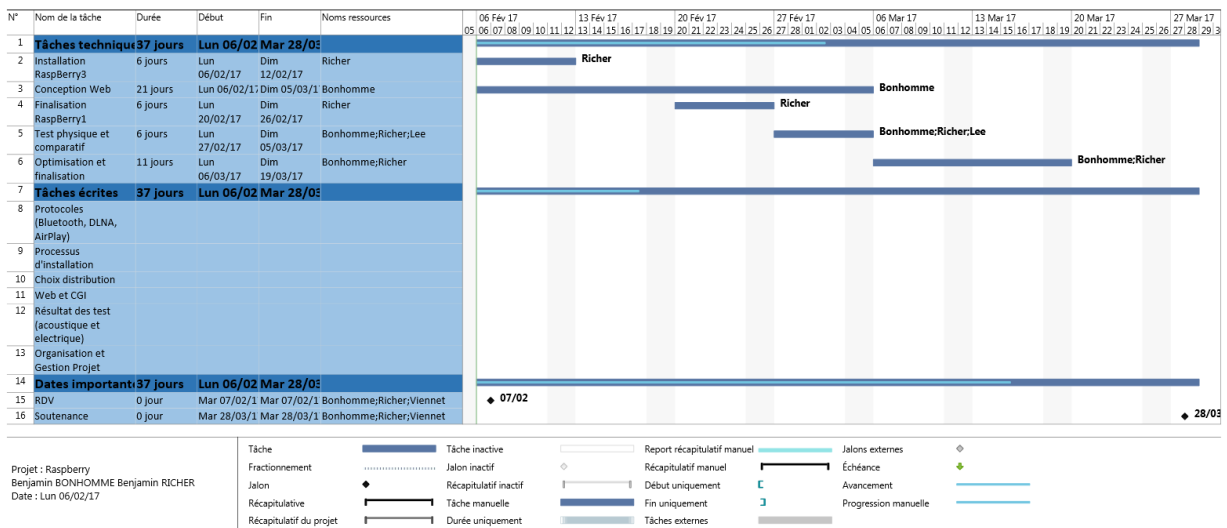


Figure 5 : Diagramme de Gantt du semestre 4.

2. Présentation du matériel

2.1 Liste du matériel utilisé

Au début de ce projet l'IUT nous a fourni du matériel que nous allons lister :

- Le Raspberry Pi Starter Kit qui contient :
 - o La carte Raspberry Pi B+
 - o Une carte SD de 8Go avec NOOBS préinstallé dessus
 - o Le boîtier pour la carte Raspberry Pi B+
 - o Le chargeur EU/US
 - o Un manuel d'utilisation
- Le Raspberry Pi 3 qui a été fourni courant décembre 2016 à notre demande. En effet le Pi 3 intègre nativement le Wi-Fi et le Bluetooth ce qui facilite les manipulations et apporte certains avantages (moins de drivers à installer car on n'utilise plus de dongles, on libère 2 ports USB ce qui va réduire un peu notre consommation électrique, les risques d'interférences entre les deux antennes sont mieux gérés).
- Un chargeur, dont la différence avec l'autre chargeur est l'ampérage qui est de 2,5A au lieu de 2A.

Nous avons aussi du matériel additionnel que nous avons déjà en notre possession ou que nous avons dû commander :

- Lecteur de carte MicroSD pour formater les cartes.
- Un dongle Wi-Fi pour le Raspberry Pi B+.
- Un dongle Bluetooth pour le Raspberry Pi B+.
- Un clavier, une souris, un écran qui sont utiles pour les premières utilisations du Raspberry Pi, c'est-à-dire l'installation du système d'exploitation et sa configuration.
- Le câblage nécessaire, c'est-à-dire un câble d'alimentation, un câble DVI (pour brancher le Raspberry à l'écran, avec un adaptateur HDMI), un câble Ethernet pour accéder au réseau et à internet afin de faire des mises à jour et installer des paquets.
- Un multimètre voltmètre ampère USB qui va nous permettre de simplement mesurer la consommation électrique.

La liste détaillée du matériel en images est donnée en annexe à la page 43, intitulé **Annexe 1 : Présentation détaillée du matériel.**

2.2 Spécification du Raspberry

Le Raspberry Pi est une série de nano-ordinateurs mono-carte (avec un processeur ARM) développé en Angleterre par David Braben (un créateur de jeux-vidéos) dans le cadre de la fondation « Raspberry Pi Foundation » afin de promouvoir l'apprentissage des sciences de l'informatique dans les écoles des pays en voie de développement. Cet ordinateur ayant la taille d'une carte de crédit permet l'utilisation de nombreux systèmes d'exploitation, en particulier GNU/Linux. Le premier modèle s'est répandu plus vite que ce qui était prévu avec des utilisations comme la robotique. Les périphériques comme les claviers, souris et boîtiers ne sont pas inclus avec le Raspberry Pi dans l'optique de réduire les coûts et réutiliser du matériel mais des accessoires ont été inclus dans des packs officiels ou non.

D'après la fondation Raspberry Pi, plus de 5 millions de Raspberry ont été vendus avant février 2015 ; à partir de septembre 2016 il y avait 10 millions de ventes. Le prix initial prévu par la fondation était de 25\$ en 2011 puis au fur et à mesure que les versions progressaient le prix a augmenté pour atteindre la somme de 38€ pour le Raspberry Pi 3.

Du côté hardware, la plupart des modèles suivent la représentation suivante, à l'exception des modèles A, A+ et Zero qui ne possèdent pas de connexion Ethernet et de ports USB.

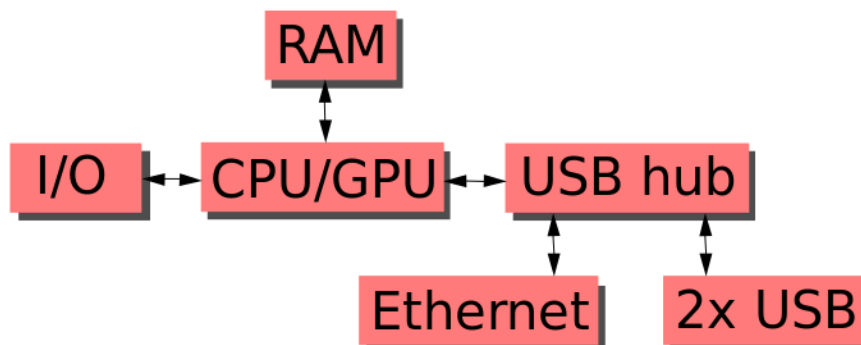


Figure 6 : Diagramme représentant les interactions entre les composants du Raspberry Pi.

Dans notre projet nous avons eu l'occasion de manipuler sur le Raspberry Pi B+ et le Raspberry Pi 3, nous allons donc comparer leurs spécifications :

Type	Raspberry Pi B+	Raspberry Pi 3
Génération	1 +	3
Date de sortie	Juillet 2014	Février 2016
Prix	25\$	35\$
Architecture	ARM v7-A (32 bits)	ARM v8-A (32/64 bits)
Puce (SoC)	BroadCom BCM2835	BroadCom BCM2837
Processeur (CPU)	ARM1176 mono-core 700MHz	ARM Cortex-A53 quad-core 1.2GHz
Graphique (GPU)	BroadCom VideoCore IV 250MHz	BroadCom VideoCore IV 300-400MHz
Mémoire (SDRAM)	512 MB	1 GB
Ports USB 2.0	4 via Hub USB 5 ports	4 via Hub USB 5 ports
Entrée Vidéo	15 pin pour camera	15 pin pour camera
Sortie Vidéo	HDMI, MIPI	HDMI, MIPI
Sortie Audio	Jack 3.5 mm ou HDMI	Jack 3.5 mm ou HDMI
Stockage	MicroSDHC	MicroSDHC ou USB en boot
Connexion Réseau	Ethernet (10/100 Mbits/s)	Ethernet (10/100), 802.11n, Bluetooth 4.1
Consommation	600 mA (3.0 W)	800 mA (4,0 W)
Source Electrique	5V via MircoUSB ou GPIO	5V via MircoUSB ou GPIO

Figure 7 : Tableau comparatif regroupant les caractéristiques du Raspberry Pi B+ et Raspberry Pi 3.

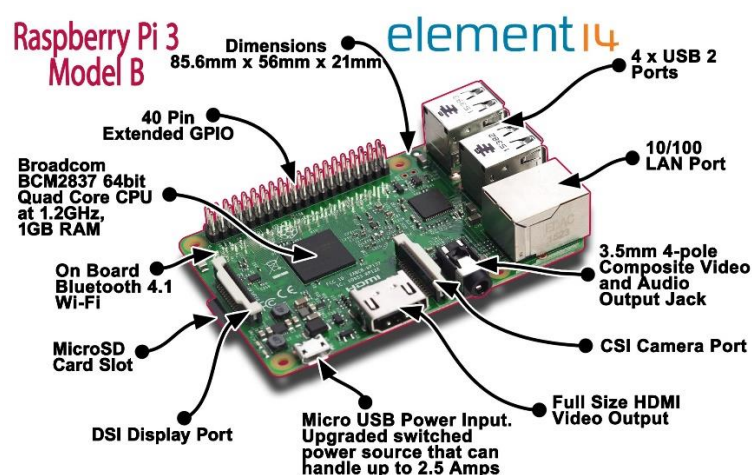


Figure 8 : Certaines des nouveautés incluses dans le Raspberry Pi 3.

2.3 Tests comparatifs entre Pi B+ et Pi 3

Dans cette partie nous allons mettre en pratique les spécifications des 2 versions du Raspberry en effectuant des tests de Benchmarking (gestion de la qualité). Pour cela nous avons utilisé un script bash rédigé par AikonCWD (GitHub).

Ce script va effectuer 7 tests de performance pour voir les limites des 2 versions du Raspberry :

- Un test de connexion : calculer le ping, le débit montant et descendant.
- Un test de processeur (CPU) : calculer les 5000 nombres premiers.
- Un autre test de processeur (CPU) : faire du Multithread avec 4000 yields et 5 locks.
- Un test de mémoire (RAM) : Accès séquentiel à 3Gb de mémoire.
- Un test sur la carte mémoire (HDParm) : calculer la vitesse maximale de lecture pour la carte mémoire.
- Un autre test sur la carte mémoire (écriture) : calculer la vitesse maximale d'écriture avec des fichiers de 512Mb.
- Un dernier test sur la carte mémoire (lecture) : calculer la vitesse maximale de lecture avec des fichiers de 512Mb.

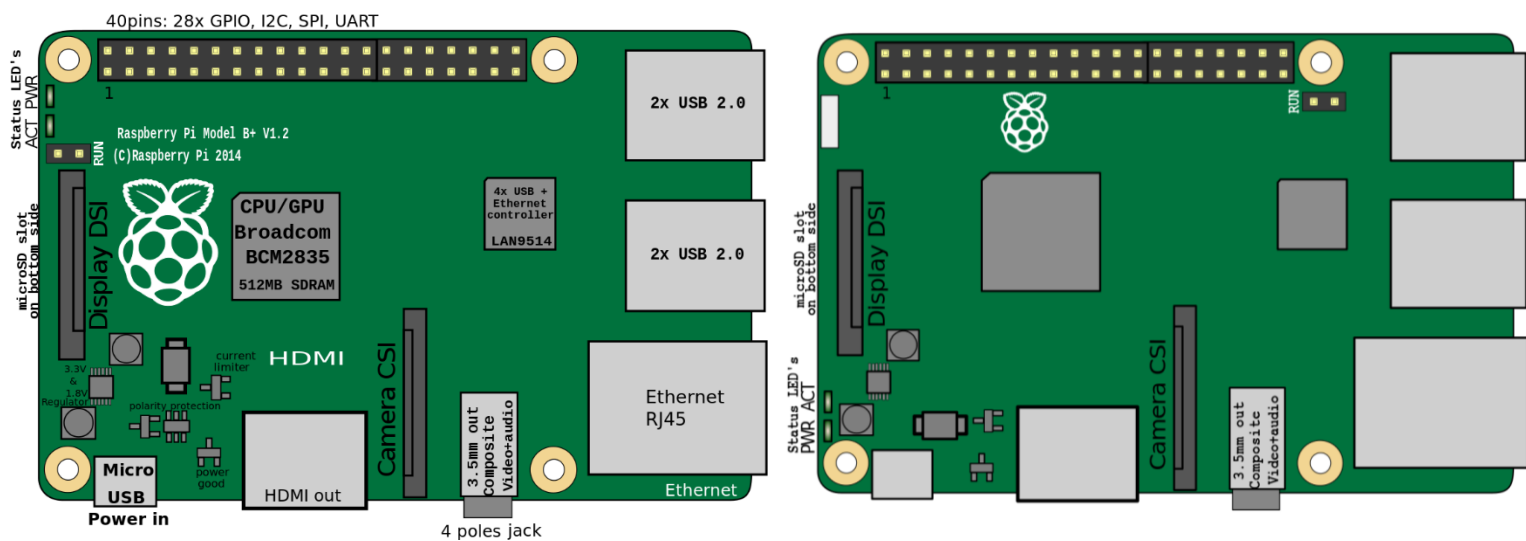


Figure 9 : Disposition du Raspberry Pi B+ à gauche et celle du Raspberry Pi 3 à droite

Le script utilisé, les résultats bruts du test de Benchmarking, ainsi que des graphiques de comparaison et leurs interprétations sont donnés en annexe à la page 49, intitulé **Annexe 2 : Test de Benchmarking effectuées sur les 2 Raspberry (Pi B+ et Pi 3)**.

Le Raspberry Pi 3 est vraiment beaucoup plus performant que le Raspberry Pi B+ sur presque tous les points (Réseaux, CPU, Mémoire et même légèrement pour l'écriture et la lecture). Le principal défaut du Pi 3 est bien entendu la température car nos tests coïncident bien avec différentes sources, le Pi 3 peut atteindre des températures de 100 °C en utilisation « extrême ».

Source : <http://www.minimachines.net/actu/raspberry-pi-3-atteint-presque-100c-38582>

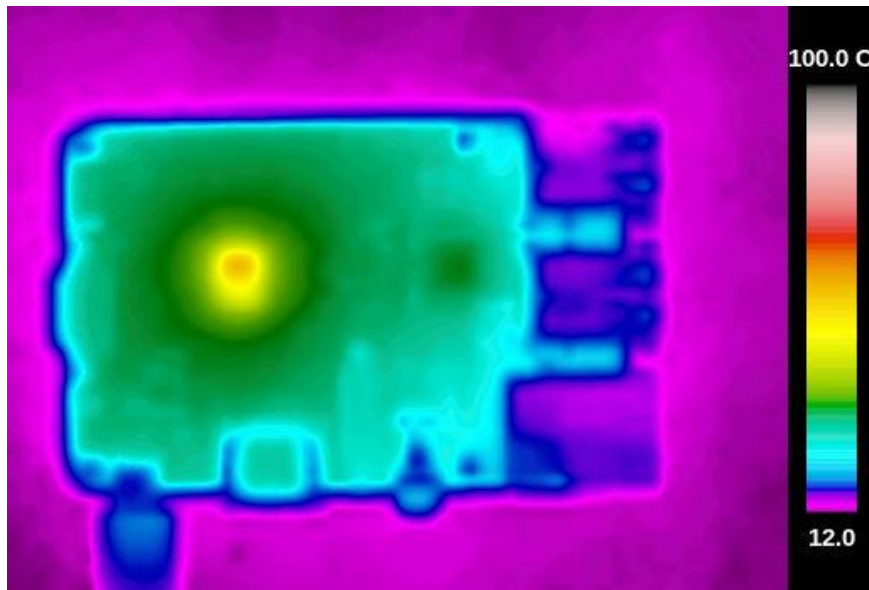


Figure 10 : Image thermique du Raspberry Pi B+. On peut voir que la température ne dépasse pas les 50°C.

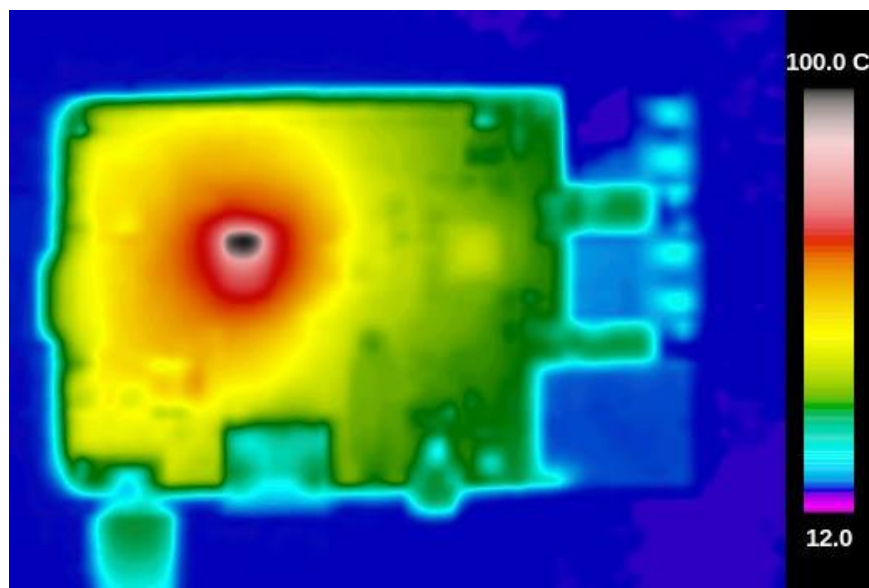


Figure 11 : Image thermique du Raspberry Pi 3. On peut observer que la température peut dépasser les 100 °C aux alentours du processeur.

3. Installation/configuration du Raspberry et des différents paquets

3.1 Choix de la distribution

Le Raspberry Pi possède de nombreux systèmes d'exploitation qui sont compatibles avec l'architecture ARM :

- Debian : qui a une version dédiée appelée Raspbian (avant avec le kernel Wheezy et puis maintenant avec Jessie), ce système est recommandé par la fondation Raspberry Pi.
- Ubuntu Mate et Snappy Ubuntu : basé sur Ubuntu, Snappy Ubuntu est plus réservé aux développeurs.
- Fedora : un autre système d'exploitation libre et une distribution GNU/Linux.
- Arch Linux : encore une distribution Linux qui met en avant la simplicité.
- Gentoo Linux, Slackware et Suse qui sont des distributions Linux.
- RISC OS : un système d'exploitation spécialement conçu pour les architecture ARM.
- NetBSD : un système d'exploitation libre de type Unix BSD.
- Kali Linux : anciennement Backtrack, ce système d'exploitation possède tous les outils nécessaires aux tests de sécurité d'un système d'information.
- Windows 10 IOT : Windows 10 spécialement conçu pour les objets connectés.
- OSMC : Media center gratuit et libre, basé sur du Linux.
- LibreElec : évolution de OpenElec, un autre media center.

Il existe aussi NOOBS (New Out Of the Box Software) qui est un assistant et va permettre l'installation d'un ou plusieurs système d'exploitation sur la carte MicroSD sans avoir à se soucier du partitionnement. NOOBS possède des avantages certains : tout d'abord il est simple à mettre en œuvre sur la plupart des systèmes d'exploitation et tous est simplifié au maximum grâce à une interface graphique épurée, on peut installer plusieurs systèmes d'exploitation sur une même carte SD, s'il y un problème on peut réinitialiser ou désinstaller le système d'exploitation source du problème.

Nous avons choisi de ne pas passer par NOOBS et d'installer manuellement Raspbian sur le Raspberry Pi. En effet les avantages proposés par NOOBS ne sont pas utiles dans notre situation car nous savons déjà installer Debian en mode expert sur une machine normale et nous comptons utiliser seulement Raspbian. De plus NOOBS prend sensiblement plus de place sur la carte mémoire et nous semble moins stable de façon général que Raspbian.

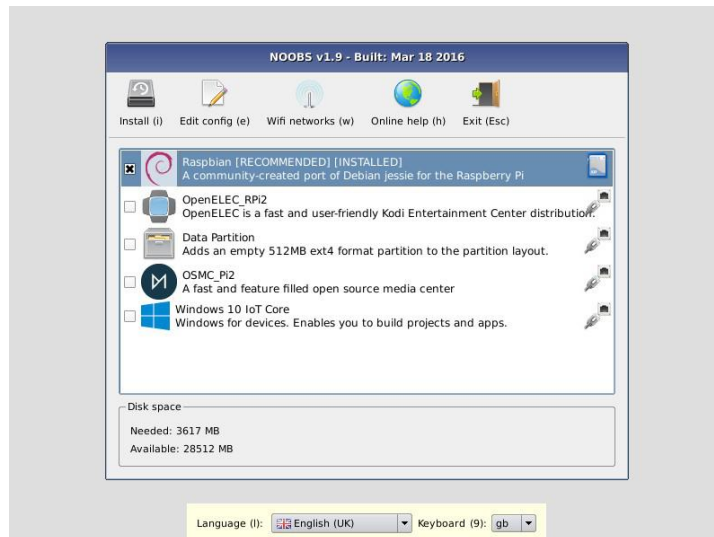


Figure 12 : Au démarrage du Raspberry Pi avec une carte MicroSD ayant NOOBS chargé dessus, c'est le menu que l'on obtient.

Pour faire court, si on veut comparer Raspbian et Debian il faut savoir que Raspbian est un portage non officiel de Debian Wheezy ARMhf (Arm Hard Float Port). Les architectures ARM, sont des architectures matérielles RISC (Reduced Instruction set computer, traduit littéralement par processeur à jeu d'instructions réduit). On ne peut donc pas installer les paquets de la même manière sur une architecture ARMhf et AMD64/i386 (Debian). Il va donc falloir utiliser ce qu'on appelle de la compilation croisée.

Une chaîne de compilation croisée est une chaîne compilée pour fonctionner sur l'architecture de processeurs de la machine hôte, mais qui va compiler des logiciels pour une architecture cible différente. La compilation croisée fait donc référence aux chaînes de compilation capables de traduire un code source en code objet dont l'architecture processeur diffère de celle où la compilation est effectuée. Ces chaînes sont principalement utilisées en informatique industrielle et dans les systèmes embarqués.

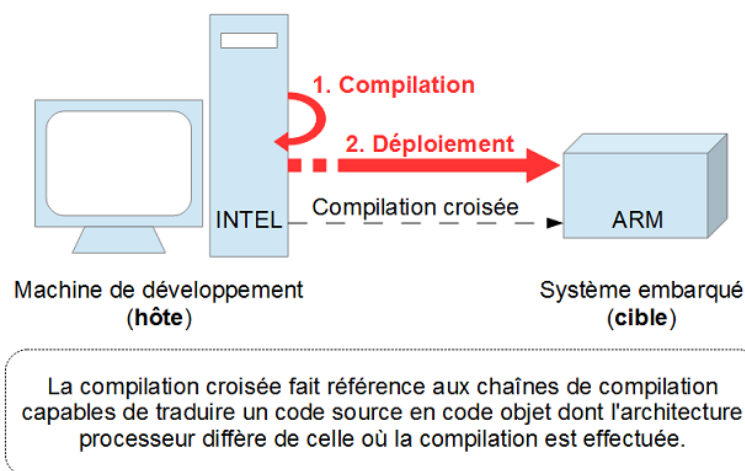


Figure 13 : Explication sous forme de schéma du fonctionnement de la compilation croisée.

3.2 Première installation et configuration de Raspbian

Nous allons énoncer les principales étapes pour installer et configurer Raspbian :

- Préparer la carte MicroSD, pour cela il faut télécharger l'image de Raspbian, puis utiliser un logiciel comme Win32DiskImager pour rendre la carte bootable avec le bon système d'exploitation dessus.
- Mettre la carte MicroSD dans le Raspberry et effectuer les branchements nécessaires à une première installation (souris, clavier, câble Ethernet, câble DVI avec un adaptateur DVI-HDMI) puis alimenter le Raspberry.
- Attendre que le système démarre et lorsque nous avons accès à l'interface graphique, lancer un terminal pour commencer la configuration de Raspbian avec la commande `sudo raspi-config`.
- Faire en sorte que Raspbian utilise bien toute la carte MicroSD avec Expand Filesystem.
- De base le clavier est en QWERTY, il faut donc le passer en AZERTY grâce au menu Change Keyboard Layout.
- Passer Raspbian en Français grâce au menu Change Locale.
- Modifier le mot de passe pour l'utilisateur par défaut pi et mettre un mot de passe pour root.
- Activer le protocole SSH afin de ne plus avoir besoin de tous les accessoires lors des prochaines utilisations.
- Mettre à jour les sources et faire un upgrade du système puis redémarrer.
- Optionnellement on peut installer VNCserver qui s'occupe d'exporter l'environnement graphique si nous avons besoin d'une interface graphique.

Toutes les étapes détaillées pour installer Raspbian et le configurer sont présentes en annexe à la page 57, intitulé **Annexe 3 : Manipulations première installation et configuration de Raspbian**.

3.3 Les différents paquets utilisés.

Nous allons lister les paquets, les applications et les bibliothèques qui nous ont été utiles dans le cadre de notre projet :

Nom	Droit d'usage	Fonction	Utilité dans notre projet
Wireshark	Libre	Analyse de paquets.	Régler un problème dû à un chargement trop long de nos pages web sur le téléphone.
Php5	Libre	Produire des pages web dynamiques.	Réalisation de l'interaction PHP-Bash pour avoir l'état des services sur une page web.
Apache2	Libre	Mettre en place un serveur web.	Le Raspberry sera notre serveur web.
Hamachi	Non Libre	Création de réseaux privés virtuels de façon simple.	Pouvoir travailler en groupe sur le Raspberry sans avoir à le transporter. Ne sera pas présent dans le produit final.
Hostapd	Libre	Permet la création d'un point d'accès Wi-Fi.	Airplay nécessitant une connexion Wi-Fi entre le récepteur/receveur on a voulu que Raspberry soit autonome et ne dépende pas d'un Wi-Fi domestique.
dnsmasq	Libre	Serveur léger pour fournir les services DNS, DHCP.	Permet de simplifier l'accès à l'utilisateur, le DHCP s'occupe de l'attribution des adresses.
PulseAudio	Libre	Serveur de son permettant de transférer le son, changer le format d'échantillonnage.	Nos protocoles Bluetooth et Airplay vont interagir avec ce serveur de son.
Bluez	Libre	Met en œuvre la technologie Bluetooth sur les OS Linux.	Bluetooth est une obligation pour les appareil Android.
Shairport-sync	Libre	Joue le rôle de lecteur audio Airplay.	Airplay est une obligation pour les appareils Apple.
Scrot	Libre	Permet de prendre des captures d'écran	Prendre les résultats de nos manipulations
tightvncserver	Libre	Permet d'avoir un bureau à distance.	Si nous avons vraiment besoin d'un interface graphique on peut utiliser VNC.

Figure 14 : Tableau qui recense l'ensemble des paquets utilisés.

Toutes les étapes détaillées pour installer les paquets et les configurer sont présentes en annexe à la page 73, intitulé **Annexe 4 : Manipulations pour installer les différents paquets et applications.**

4. Etude des protocoles Airplay (Shairport) et Bluetooth

4.1 Comparatifs entre Airplay et Bluetooth

Nous avons essayé de synthétiser au maximum les différences entre Airplay et Bluetooth.

	Bluetooth Audio	Airplay
Débit maximal	24Mbits/s	36,7 Mbits/s
Portée	10 – 15 m	28 – 70 m
Dépendant du Wi-Fi	Non	Oui
Formats audio compatibles	SBC, MPEG-1 2 et 4, MP3, AAC, ATAC, AptX	AAC, MP3, ALAC (Apple Lossless)
Droit de license	Non	Oui (mais Shairport Retro-ingénierie)
Transmission sans perte	Partiellement	Oui

Figure 15 : Tableau qui recense les caractéristiques des protocoles Bluetooth et Airplay.

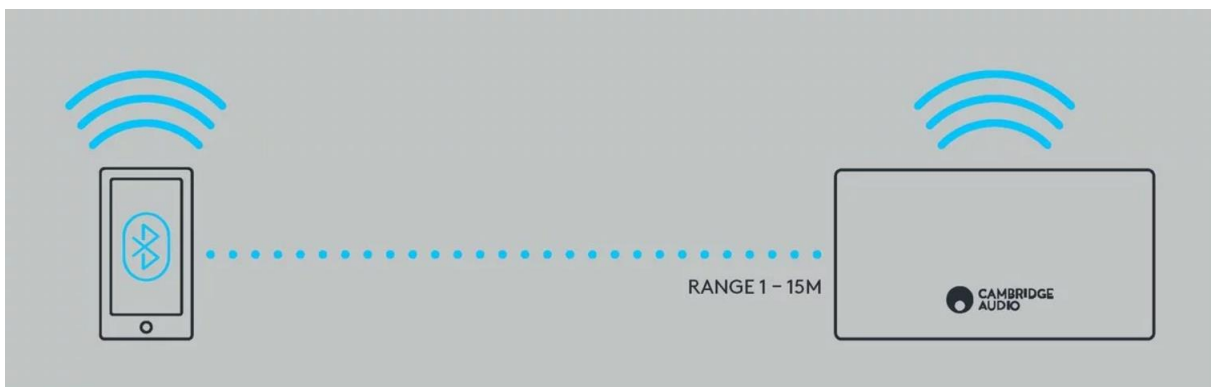


Figure 16 : Portée du Bluetooth.

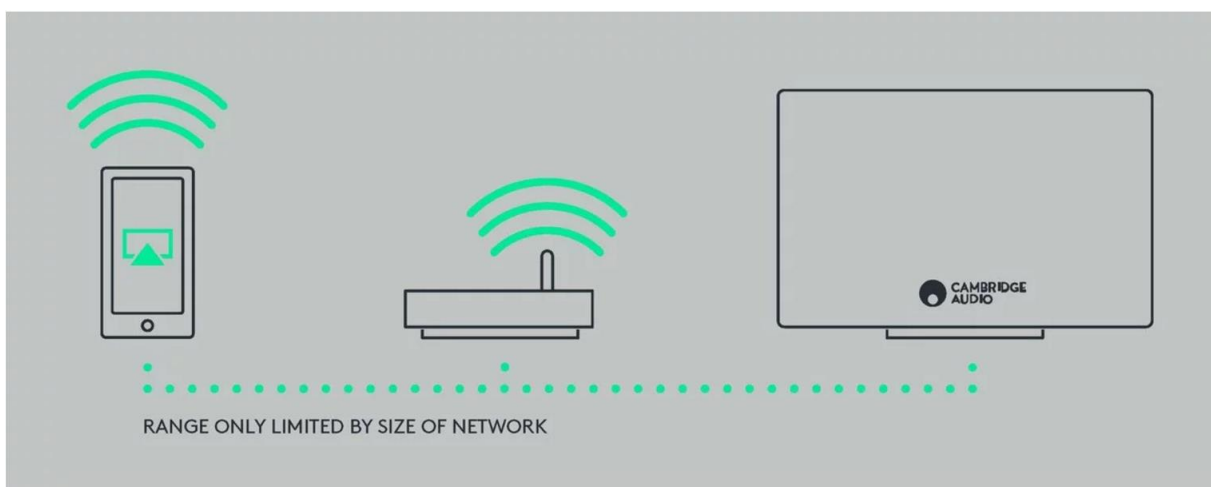


Figure 17 : Portée de Airplay.

	Avantages	Inconvénients
Airplay	<ul style="list-style-type: none">- Plus longue distance couverte pour la lecture de musiques.- Utilise de la compression sans perte.- Communication directe avec l'appareil iOS et l'appareil utilisant Airplay (contrôle de volume).	<ul style="list-style-type: none">- Doit passer par le réseau Wi-Fi ce qui amène une plus grande complexité dans les protocoles et l'architecture.- Airplay est en théorie disponible seulement sur Apple et quelques autres appareils rares.
Bluetooth	<ul style="list-style-type: none">- Protocole très large qui fonctionne avec un grand nombre d'appareil de système d'exploitation (Windows, Android, iOS).- On peut régler le volume avec l'appareil qui diffuse la musique.- Utilise une grande variété de formats dont certains de bonne qualité (AAC, aptX)	<ul style="list-style-type: none">- Distance limité comparé à Airplay (de 10 à 15 mètres grand maximum).- Bluetooth stream les flux audio avec une compression qui peut inclure des pertes, pas aussi bon que la compression de Airplay.

Figure 18 : Tableau recensant l'ensemble des avantages et inconvénients des 2 protocoles.

Tous les détails techniques des protocoles Airplay et Bluetooth sont présentés en annexe à la page 86, intitulé **Annexe 5 : Etude détaillée des protocoles Airplay et Bluetooth.**

5. Interface WEB (PHP)

Dans cette partie nous allons aborder tout ce qui est attrait à l'interface Web et les interactions entre l'utilisateur final et le produit.

D'abord définissons ce qu'est une interface web :

Une interface web correspond à une interface (dite « homme-machine ») qui permet le dialogue entre le système technique et son utilisateur. Une interface web est visualisable à partir de n'importe quel dispositif possédant un navigateur web (ordinateur, tablette ou smartphone, etc.).

A partir de là on peut donc adapter cette définition à notre projet car notre utilisateur final sera amené à utiliser le site web le plus souvent depuis son téléphone et sa tablette.

Cela pose la première contrainte qu'est la responsivité du site.

Ensuite on peut poser des contraintes de rapidité telles que le temps de réponse que doit retourner l'interface.

5.1 Installation paquets

Pour construire notre interface web nous avons besoin de plusieurs paquets essentiels. Un pour le serveur web HTTP et ceux liés au langage PHP qui nous permettra d'interfacer le serveur web et le Raspberry en lui-même.

5.1.1 Apache2



Figure 19 : Logo d'Apache (http server).

Le logiciel libre Apache HTTP Server est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.

Son installation est assez simple, il n'y a pas de configuration spécifique à apporter. On peut directement après son installation se rendre sur la page d'accueil (<http://localhost> ou <http://raspberrypi> selon l'emplacement) et visualiser sans aucun souci la page de test.

On peut ensuite se déplacer dans le répertoire de travail du serveur web : /var/www/html et y placer les dossiers ou fichiers que nous souhaitons.

On notera aussi que la résolution DNS au travers de notre VPN se passe bien et que nous pouvons au travers du tunnel afficher sans délai supplémentaire les pages du Raspberry.

5.1.2 Php5



Figure 20 : Logo de PHP5.

Une fois Apache installé on peut y ajouter le module php5.

PHP: Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

Ce langage va nous permettre d'interfacer un script local et afficher son résultat dans la page web.

Très utile donc par exemple pour régler le son du Raspberry.

De plus PHP se distingue par un aspect plus sécuritaire car le client ne reçoit pas le code PHP du serveur et donc ne peut pas lire les options du script local.

Le paquet en soit est un peu long à installer car l'invocation du module php5 au sein d'apache est un long processus. Nous avons pu le voir à maintes reprises quand on réinstalle PHP (suite à des erreurs de manipulations).

5.2 Les différents langages

Pour manipuler tous ces paquets nous devons écrire du code dans différents langages. Il existe de multiples combinaisons orienté web. Nous utiliserons une combinaison assez complexe mais extrêmement répandu, et donc assez bien documenté : HTML + CSS + PHP + JavaScript

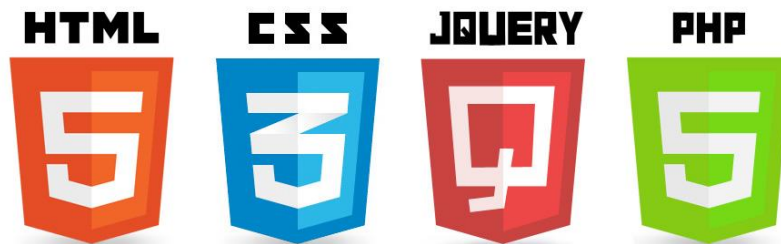


Figure 21 : Les différents langages utilisés dans ce projet.

Ces langages sont de nos jours devenus des standards dans le web dynamique.

HTML pour la partie web, CSS pour l'affichage, PHP pour l'interaction et JS pour la partie dynamique.

5.2.1 HTML

L'HyperText Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques.

Le choix de HTML est assez évident et peu de concurrent s'offrait à nous. De plus nous avons déjà des bases de HTML vu au premier semestre.

Le fonctionnement en balise et la grande documentation de ce langage nous a simplifié la tâche et nous avons au final eu très peu de problème d'un point de vue HTML.

5.2.2 CSS

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C).

Le CSS offrant un large panel de personnalisation nous avons décidé de ne pas nous lancer dans la conception de notre page CSS.

5.2.2.1 Choix du Template

Nous nous sommes donc dirigés vers des Template CSS. Cela nous permettait de partir sur une base et donc de pouvoir travailler sur la partie PHP plus vite.

Un Template (ou « thème », « layout ») désigne l'enveloppe graphique d'un site Internet, indépendamment de son contenu. Il s'agit par exemple de la disposition des colonnes, du choix des couleurs, de la structure des différents éléments, etc.

Beaucoup de créateurs de site utilisent un Template déjà fait, pour ne pas avoir besoin de s'occuper de la dimension graphique et se concentrer sur le contenu du site.

A l'instar d'un CMS tel que WordPress nous avons utilisé un Template CSS sobre fourni directement par la W3C. Nous avons juste besoin principalement d'une barre de menu responsive.

La barre de menu était notre principal critère, nous avons donc eu un panel de choix et nous avons choisi en premier abord ce Template qui semblait approprié à une page de gestion comme la nôtre :

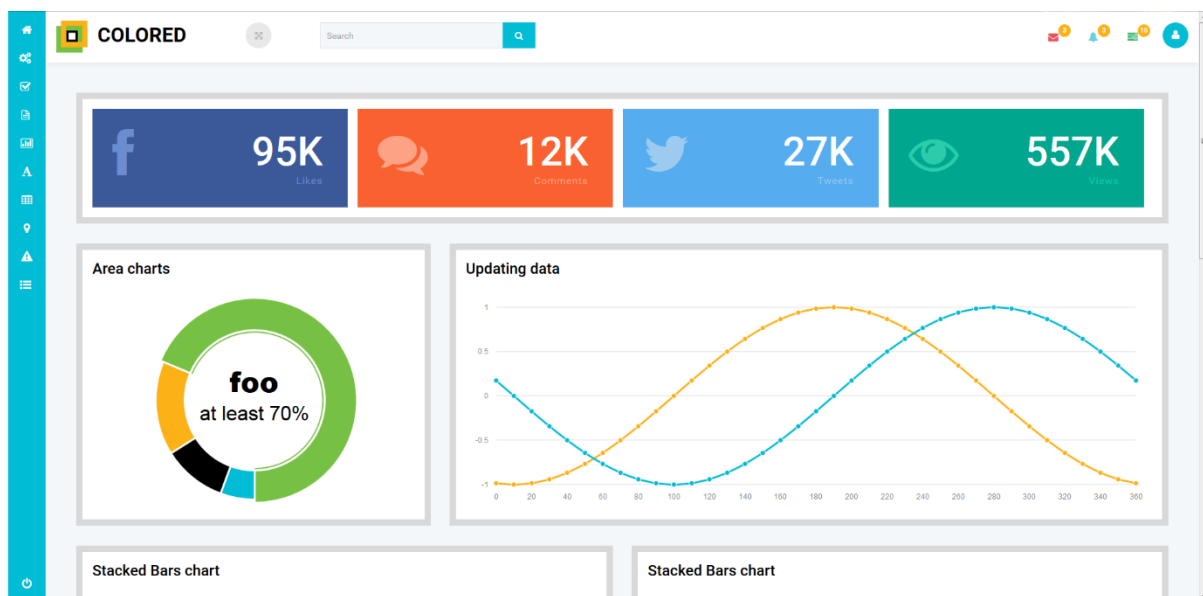


Figure 22 : Aperçu de la première interface utilisée.

Mais par la suite et vu la complexité de ce Template (trop d'interaction avec JavaScript) nous avons décidé de migrer vers ce Template beaucoup plus sobre.

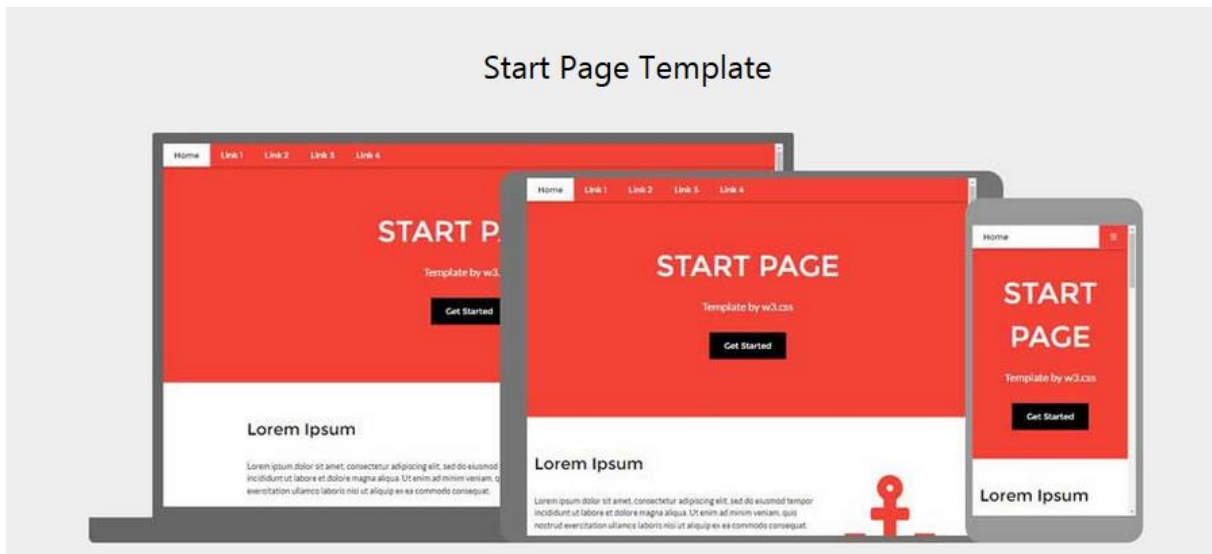


Figure 23 : Template pour l'interface finale utilisée dans notre projet.

5.2.2.2 Compréhension et adaptation au Template final

Une fois le choix du CSS fait, il fallait y appliquer les modifications nécessaires pour que celui-ci nous conviennent.

Il a d'abord fallu supprimer toutes les références dynamiques au page CSS externe pour une raison simple c'est que notre Raspberry et/ou l'appareil de l'utilisateur n'est pas censé être connecté à Internet pour fonctionner.

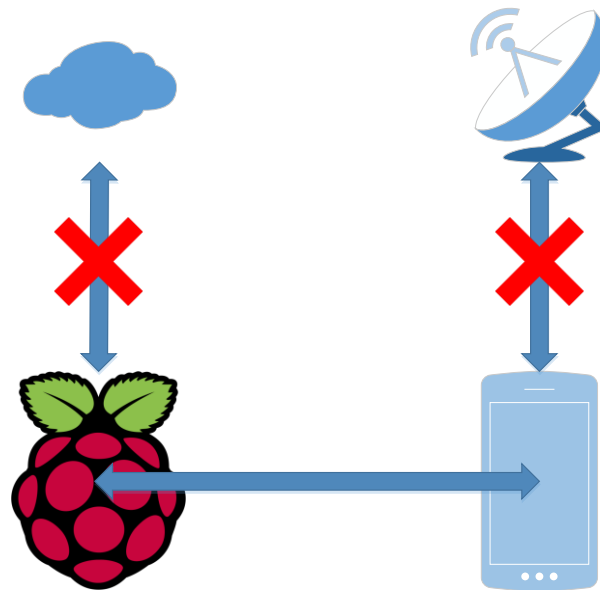


Figure 24 : Notre principal problème concernant l'utilisation des feuilles CSS sur le téléphone.

La résolution des feuilles CSS prenait donc un temps considérable (1 à 2 minutes) en fonction que le téléphone soit connecté ou non à Internet et/ou que le Raspberry le soit.

Nous avons donc dû télécharger manuellement toutes les feuilles CSS et leur dépendances (fichier .ttf de police, etc...).

Nous avons dû aussi faire des menues modifications de marge ou autre en fonction de l'affichage de certain composant.

Le fait de travailler sur un Template existant nous mâche le travail pour la conception mais dès lors que l'on souhaite modifier un peu le fichier, il faut se pencher sur sa conception globale ou utiliser l'inspecteur d'élément du navigateur qui nous dit directement les classes et feuille CSS qui génèrent l'élément mis en surbrillance. Cet outil nous a fortement aidés et on en reparlera après.

5.2.3 PHP

Le PHP est comme nous l'avons dit précédemment un langage de programmation orienté objet, nous en avons principalement besoin pour exécuter un script bash en local sur le serveur et au besoin inscrire le résultat sur la page Web.

5.2.3.1 Apprentissage

Il a donc fallu apprendre les bases du langage, la syntaxe variant d'un langage à un autre. Pour cela nous avons lu un tutoriel d'OpenClassRoom reprenant les bases du PHP.

Cela était nécessaire pour connaître la structure de base d'une condition ou d'une boucle par exemple.

De plus ce langage nous était à tous deux inconnus et nous ne l'avions pas abordé en cours. C'est un bon moyen d'enrichir notre culture linguistique.

5.2.3.2 Phase de test

Une fois les bases apprises, nous avons pu mettre en œuvre les différentes méthodes permettant d'exécuter un script, la plus simple étant *shell_exec*, on peut aussi citer *exec* ou *system*.

Ces 3 méthodes retournent un String que nous pouvions ensuite écrire dans le contenu de la page.

A noter que la page attend la fin de l'exécution du script pour afficher son contenu, nous avons pu en faire l'expérimentation en essaya de faire un ping sans spécifier de fin.

Notons aussi qu'il existe des méthodes spéciales pour protéger les caractères du shell, c'est-à-dire que si la commande *shell_exec* prend en paramètre une entrée utilisateur il ne faudrait pas que celui-ci puisse supprimer par exemple le contenu de la page Web.

5.2.4 JavaScript, peu mais pourquoi



Figure 25 : Logo JavaScript.

JavaScript nous permettait aussi à l'aide du fameux plug-in Node.js d'exécuter des commandes sur le système mais nous avons rencontré plusieurs problèmes lors de son installation du fait de l'architecture processeur du Raspberry.

De plus ce que PHP fait en une commande, il faut le faire en 10 lignes avec JavaScript car nous utilisons le principe de processus.

5.2.5 La CGI PHP/Bash



Figure 26 : Logo BASH, the bourne again shell.

Nous utilisons donc PHP pour appeler notre script bash à l'aide de la commande :

```
exec('/bin/bash /var/www/html/script.sh 1');
```

Cette commande spécifie donc l'interpréteur (car PHP ne comprend pas les shebang), l'emplacement du script à exécuter ainsi que l'argument. Ici nous utilisons un système d'argument chiffré qui sera ensuite récupéré dans le script bash sous la forme d'un switch case numérique.

Cela évite les accumulations de if else mais complique la tâche car il faut retenir le chiffre. Si le nombre d'argument venait à augmenter nous aurions utilisé un switch verbeux.

La valeur de retour du script bash sera récupéré par PHP dans une variable et on peut ensuite soit travailler sur la variable soit l'écrire directement dans le code.

Dans l'exemple du volume nous avons préféré utiliser des grep et des sed qui nous étaient plus familier.

Mais cela revient au même car c'est au Raspberry d'interpréter les commandes et non à l'utilisateur (comme le ferait AJAX).

5.2.6 Mod_Python ?



Figure 27 : Logo de mod-python

Mod_python est un module serveur Apache HTTP qui intègre le langage de programmation Python avec le serveur. Il est destiné à fournir une liaison de langage Python pour le serveur HTTP Apache.

Mod_Python était un concurrent de PHP et nous l'avions déjà vu en cours. Cela permet de faire des pages web dynamiques à l'aide d'une syntaxe Python.

Sauf que nous étions peu adroit avec ce langage et d'un point de vue morale celui-ci ne nous convenait pas, il était aussi peu mis à jour et la documentation se résumait au site officiel. Peu de forum informatique abordé ce langage python qui est à notre gout un peu simplet.

Toutes les lignes de code sont présentes en annexe à la page 103, intitulé **Annexe 7 : Interface Web : partie programmation.**

5.3 Problèmes rencontrés

5.3.1 Droits d'accès et permissions entre root et www-data

Le principal problème que nous avons rencontré dans cette partie sur l'interface web a été les droits d'accès. A savoir que les scripts PHP sont exécutés depuis l'utilisateur www-data qui est l'utilisateur système lié au serveur Apache. Nous avons donc dû manipuler les droits d'accès et les propriétaires (*chmod* et *chown*) pour pouvoir exécuter correctement notre script. De plus nous avons dû modifier les groupes secondaires de www-data pour lui ajouter les groupes audio, pulse et shairport.

5.3.2 Problème avec le reboot et PHP

Actuellement le reboot depuis l'interface web n'est pas possible du fait que la commande `sudo reboot` fait complètement planter le module PHP en entier et cela sans aucune trace dans les logs.

Nous avons essayé de placer l'utilisateur dans les fichiers *sudoers* en lui donnant en premier temps les droits nécessaires, puis par la suite tous les droits mais sans succès.

Nous avons aussi essayé de faire directement un *shell_exec* ou de passer par un processus *system*.

Nous obtenions le redémarrage mais impossible de faire marcher le module PHP au redémarrage (réinstallation nécessaire). Il était donc possible que ce soit le module qui crash au reboot, on a donc essayé de couper le service apache avant de redémarrer mais du coup on obtenait un time-out.

5.4 Outil et méthodologie de développement

5.4.1 MobaXTerm, logiciel ssh et sftp



Figure 28 : Logo de MobaXterm.

Comme nous l'avons précédemment dit, le Raspberry se trouve physiquement chez Benjamin RICHER, et de ce fait Benjamin BONHOMME devait donc se connecter au travers de ssh pour modifier des fichiers. Il a pour cela d'abord utilisé putty mais le programme était assez restreint niveau fonctionnalités. On a donc préféré passer sur un logiciel standalone un peu plus évolué : MobaXTerm. Ce logiciel permet de faire du ssh dans un environnement plus graphique et avec plus de fonctionnalités. Il a aussi une fonctionnalité de sftp très pratique pour faire du transfert de fichier.

5.4.2 Notepad++ et nano

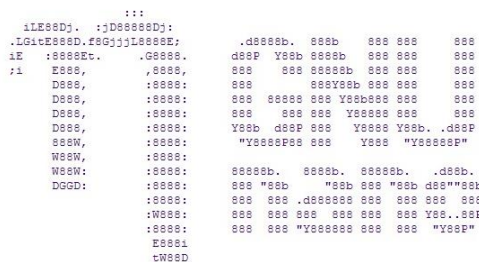


Figure 29 : Logo de GNU nano.



Figure 30 : Logo de Notepad++.

D'un point de vue rédactionnel nous utilisons en fonction du besoin l'un de ces deux éditeurs.

Notepad++ est très pratique sous l'environnement Windows mais nécessite donc de transférer le fichier une fois l'édition terminée.

Nano est un éditeur console et nous nous en servons principalement pour faire des petites modifications directement depuis le terminal ou la liaison SSH.

5.4.3 Inspecteur d'élément de Firefox



Figure 31 : Logo Mozilla Firefox.

L'inspecteur d'élément de Firefox nous permettait de retrouver et d'analyser rapidement les erreurs (souvent CSS) lié à notre page ou faire des analyses de latence et de temps de réponse au niveau applicatif.

A partir de cet outil on pouvait aussi simuler l'utilisation depuis un smartphone.

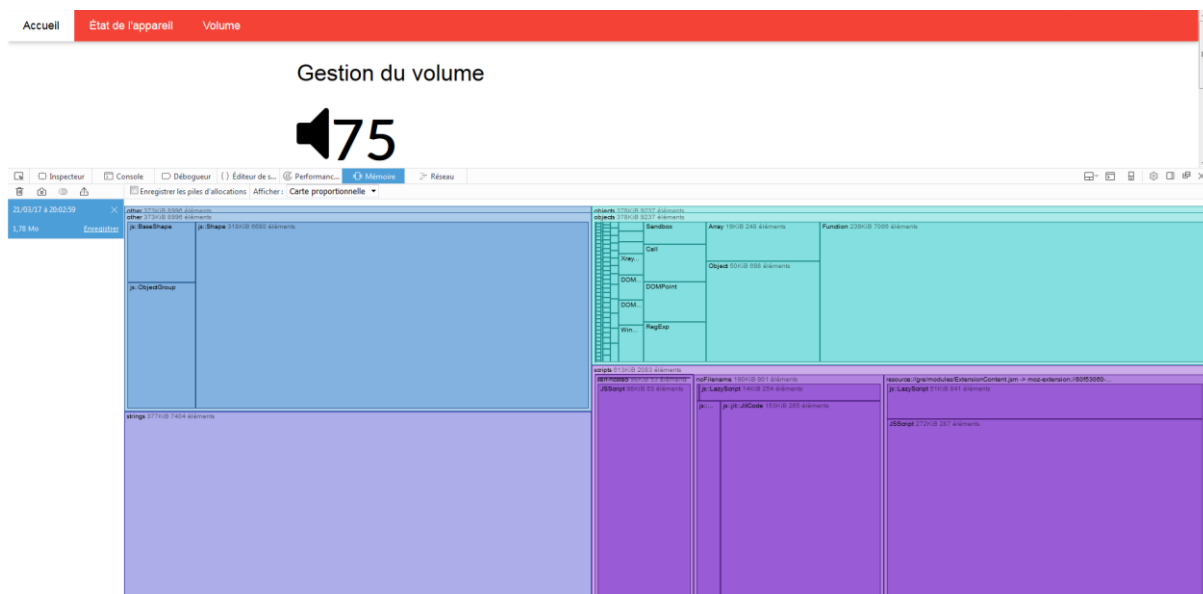


Figure 32 : Inspecteur d'élément sous Firefox.

Ici on peut voir l'utilisation détaillé de la mémoire en fonction des différents éléments de la page (script JS, objet PHP...)

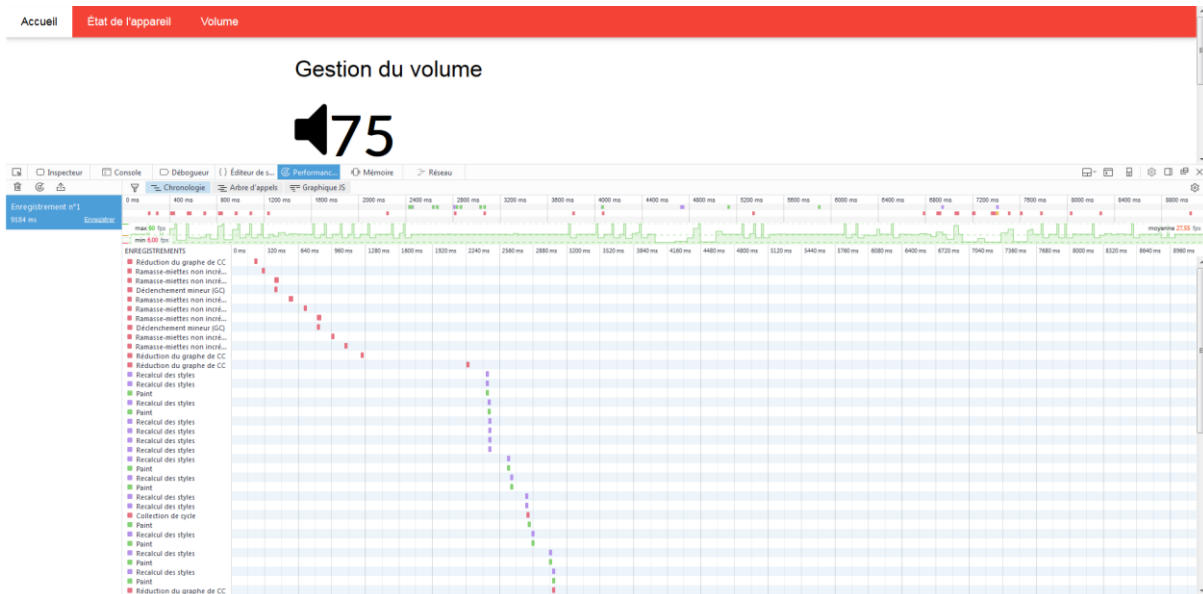


Figure 33 : Enregistrement nous montrant les performances lors d'un changement de volume.

Ici on a fait un enregistrement de performances lors d'un changement de volume (à 4sec et 7.2sec)

On peut voir toutes les interactions et les calculs du navigateur. A partir de ces résultats on peut déterminer qu'un changement de volume met 400ns à parvenir au Raspberry et entre 400ns et 1000ns à parvenir à l'utilisateur. Le dernier résultat a été obtenu en simulant une mauvaise connexion Wi-Fi.

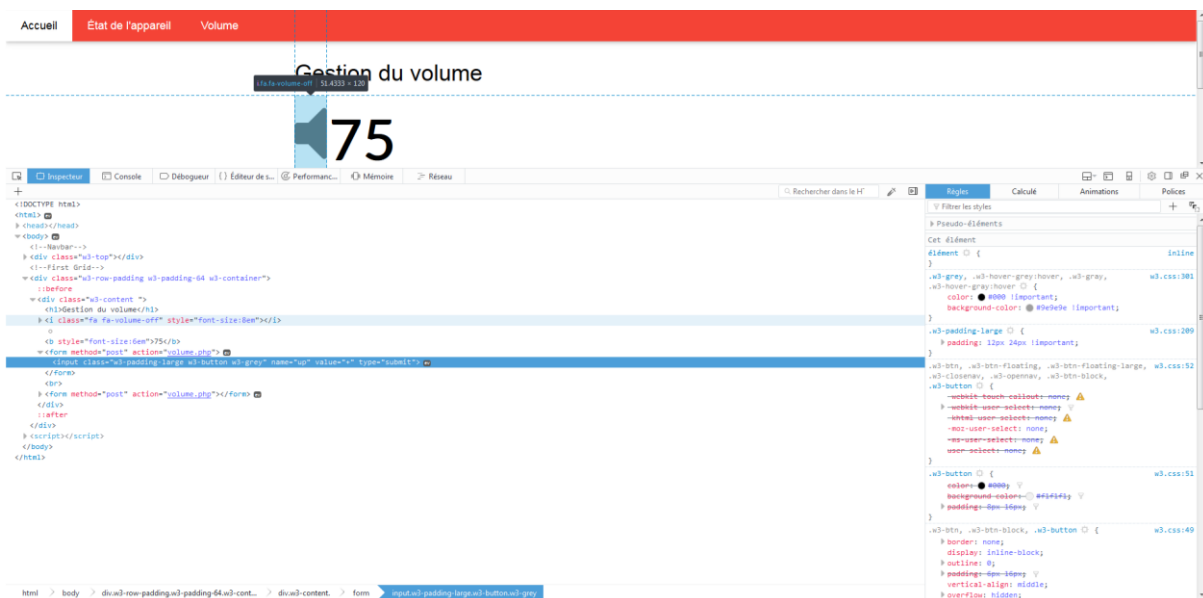


Figure 34 : Grâce à l'inspecteur d'élément nous pouvons facilement voir chaque élément de la page.

Ici la partie la plus utile, l'inspecteur en lui-même qui nous renseigne sur la ligne qui code l'élément pointé à l'écran et nous donne ces dimensions et les règles de style qui s'appliquent.

On a aussi des informations très détaillé sur les marges :

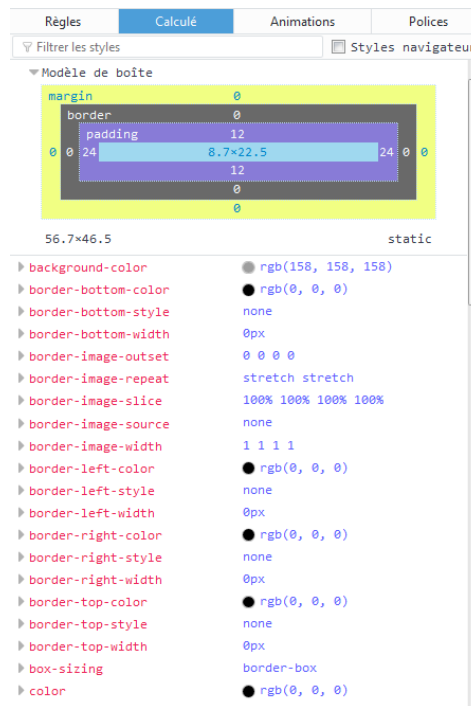


Figure 35 : Information concernant les marges.

5.5 Sécurité

Au niveau de la sécurité de la page web, l'utilisateur n'ayant pas d'entrée spécifique à fournir nous avons choisi un modèle sécuritaire très souple. Nous n'avons quasiment pas touché au fichier de configuration du serveur. La seule brèche que nous aurions pu créer aurait été le passage de www-data en sudoers pour la commande reboot.

5.5.1 .htaccess

Les fichiers `.htaccess` sont des fichiers de configuration d'Apache, permettant de définir des règles dans un répertoire et dans tous ses sous-répertoires (qui n'ont pas de tel fichier à l'intérieur). On peut les utiliser pour protéger un répertoire par mot de passe, ou pour changer le nom ou l'extension de la page index, ou encore pour interdire l'accès au répertoire.

Nous avons dans l'intention de modifier ce fichier pour mieux gérer la partie sécurité en ajoutant des règles comme :

- Interdiction accès si la source n'est pas du réseau local
- Interdiction d'accès aux terminaux n'ayant pas la résolution nécessaire

Et d'autres paramètres d'alias, mais le temps nous était compté et nous n'avons pas pu parcourir suffisamment ce fichier.

6. Etude de la qualité sonore et de la consommation électrique

6.1 Qualité audio

Pour évaluer la qualité audio d'un système audio en réseau, on peut évaluer les Performances du système en effectuant des mesures avec des appareils électroniques : analyseurs de niveau, oscilloscopes, analyseurs FFT, analyseurs de réponse impulsionnelle. Comme les critères de Performances d'un système sont clairement définis (normes), ces mesures peuvent être analysées et interprétées en se basant sur des définitions et des spécifications strictes.

Pour évaluer la qualité sonore d'un système audio, on peut mesurer la Réponse du système de façon électronique. Mais le résultat de mesures électroniques ne dit rien, par lui-même, de la qualité sonore : il faut le traduire en termes de sensations d'écoute (individuelles). On se rend bien compte de la complexité, puisqu'il n'existe aucune standardisation dans le domaine. Pour établir une matrice de transposition des caractéristiques physiques d'un son aux caractéristiques perceptives par un « public moyen », il faudrait mener des recherches cliniques, basées sur un grand nombre d'auditeurs. Aujourd'hui la qualité sonore de nombreux appareils audio s'exprime souvent sous forme de jugements personnels émis par des personnes faisant autorité parmi les professionnels de l'audio, dans la presse audio spécialisée, ou elle découle de la réputation de qualité et de l'image du fabricant. On note aussi qu'il y a de nombreuses séances d'écoute individuelles qui sont menées pour évaluer la qualité sonore d'appareils audio, mais il se présente ensuite de nombreuses complications pour transcrire ces sensations d'écoute en estimation de la qualité sonore de l'appareil.

Il n'est pas difficile de réaliser des mesures électroniques de systèmes analogiques ou numériques, puisque les appareils permettant de le faire sont très répandus. On mesure souvent un système partie par partie, directement au niveau des circuits internes de l'appareil. Les mesures électroniques possèdent l'avantage suivant: le système se mesure via un signal de test contrôlé, ce qui rend la mesure indépendante de la source sonore, et permet de reproduire la mesure à différents moments, en différents endroits, afin de la confirmer ou pour obtenir une meilleure signification statistique.

Le détail de la qualité audio dans les systèmes connectés ainsi que la différence entre qualité audio/sonore sont présents en annexe à la page 95, intitulé **Annexe 6 : Qualité audio dans les système audio en réseau.**

Nous avons essayé de tester la qualité audio délivré par le Raspberry grâce aux protocoles Airplay et Bluetooth. C'est PulseAudio qui fait l'interface entre les applications et les pilotes de cartes son et nous allons récupérer la sortie du pilote de la carte son avec GStreamer. La commande **pactl** permet d'envoyer des commandes au serveur PulseAudio.

Ici on lui demande les entrées et sorties de la carte son:

```
pactl list | grep -A2 "^Source"
```

Output: **alsa_output.0.analog-stereo.monitor**

Ce qui nous intéresse ici c'est la sortie (appelée output), on va la donner en paramètre à l'outil `gst-launch` qui commande GStreamer :

```
gst-launch-1.0 pulsesrc device=alsa_output.0.analog-stereo.monitor mute=false volume=1  
! audio/x-raw,channels=2 ! wavenc ! filesink location=test.wav
```

Les paramètres sont :

pulsesrc : source PulseAudio.

device=[...] : Sortie de la carte son trouvée plus haut.

mute=false : On s'assure que le son n'est pas coupé.

volume=1 : Le volume sonore par défaut, sans ce paramètre le son enregistré dépendra du volume général.

channels=2 : Stéréo (car par défaut c'est du mono).

wavenc : Conversion en fichier WAV.

filesink : Renvoi vers un fichier.

location =[...] : Emplacement du fichier WAV.

Pour arrêter l'enregistrement, il suffit de stopper le programme avec la combinaison de touches Ctrl+C.

Nous avons ensuite analysé l'enregistrement avec le logiciel Audacity pour obtenir les spectres :

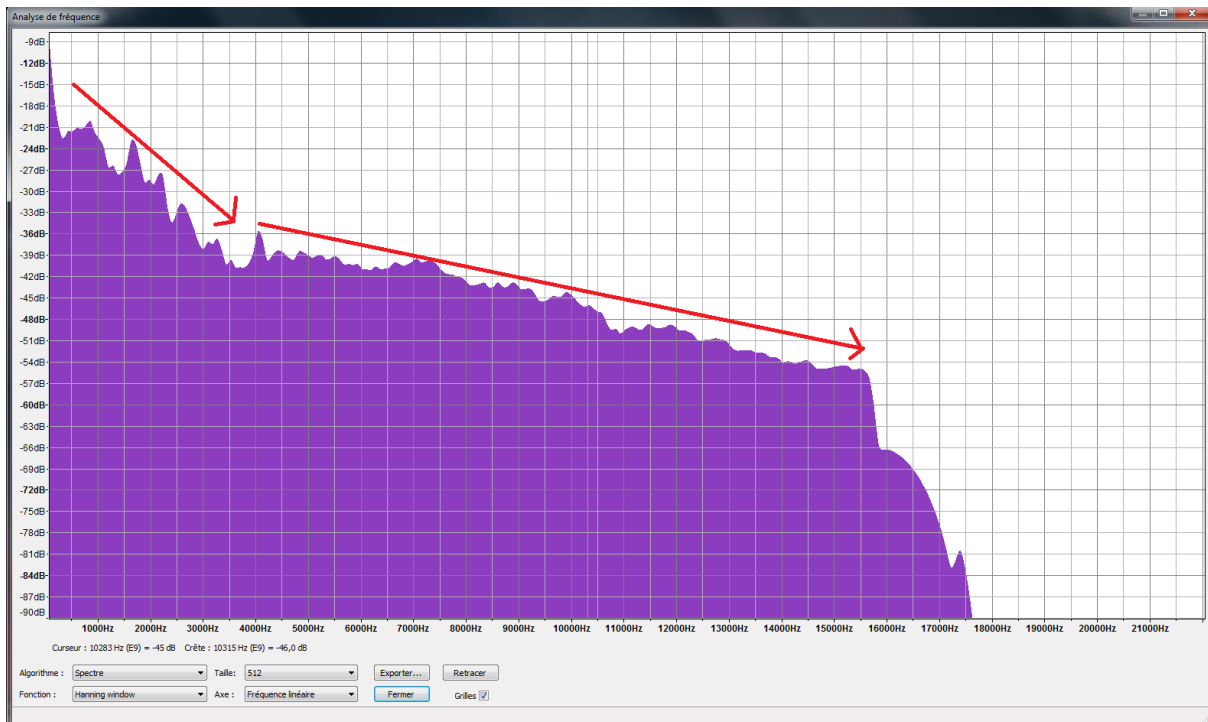


Figure 36 : Spectre Bluetooth.

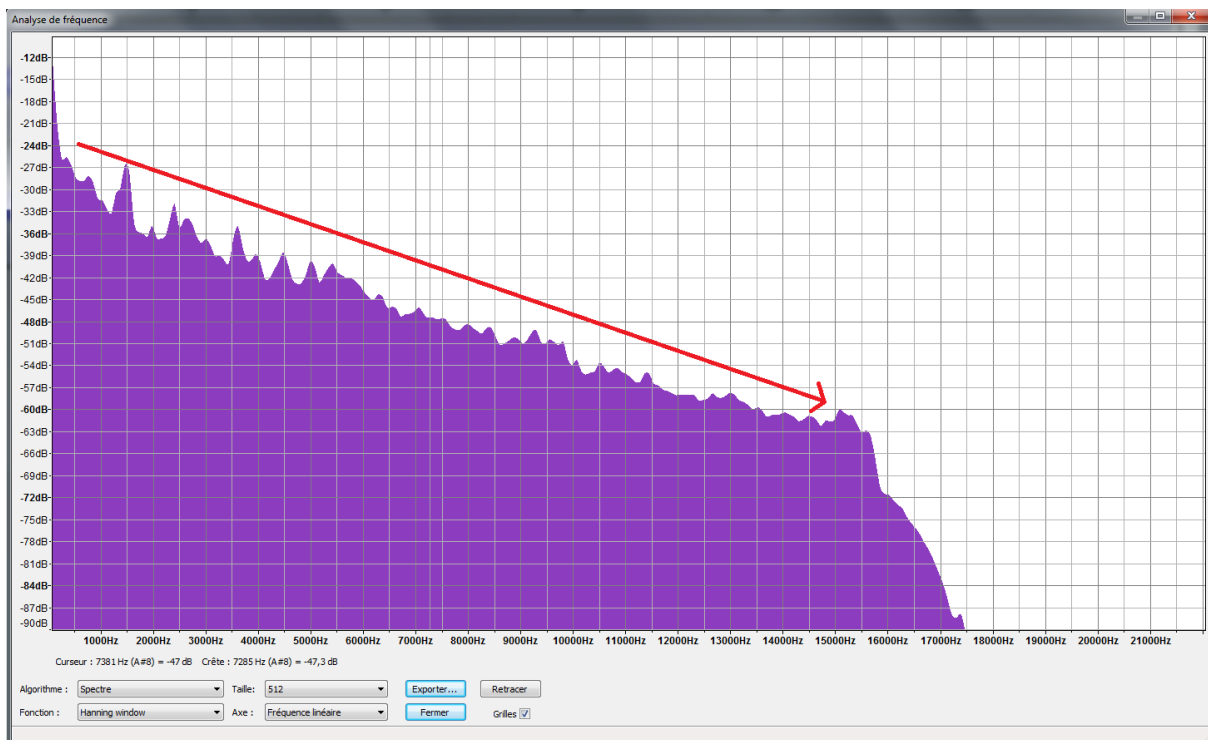


Figure 37 : Spectre Airplay.

6.2 Consommation électrique

Nous avons effectué une batterie de test concernant la consommation électrique du Raspberry.

Pour la manipulation nous avons effectué le petit montage suivant :



Figure 38 : Photo du montage pour observer la consommation électrique.

La prise courant est branché au mesureur de tension USB via câble micro-USB. Puis nous avons un câble USB to micro-USB qui part du mesureur pour aller alimenter le Raspberry.

Utilisation	Tension U (en V)	Courant I (en A)	Puissance P = U*I (en W)
Eteint mais sous tension	5,26	0,08	0,421
Démarrage sans matériel	5,21	0,53	2,761
Sans matériel, au repos	5,23	0,27	1,412
Avec matériel, au repos	5,20	0,49	2,548
Matériel + test CPU (nombres premiers)	5,18	0,67	3,471
Sans matériel + Accès serveur Web	5,22	0,30	1,566
Sans matériel + Shairport en marche	5,23	0,29	1,517
Sans matériel + Bluetooth	5,23	0,35	1,831
Sans matériel + Shairport en marche + Accès au serveur Web	5,20	0,37	1,924

Figure 39 : Tableau avec les valeurs de tension, courant et puissance.

Quand nous employons le mot matériel nous faisons référence à : Ecran + câble Ethernet + Souris + Clavier + 2 clés USB + casque audio.

7. Etude marketing du produit dans l'optique d'un lancement dans le commerce

Placement de notre produit sur une échelle de prix

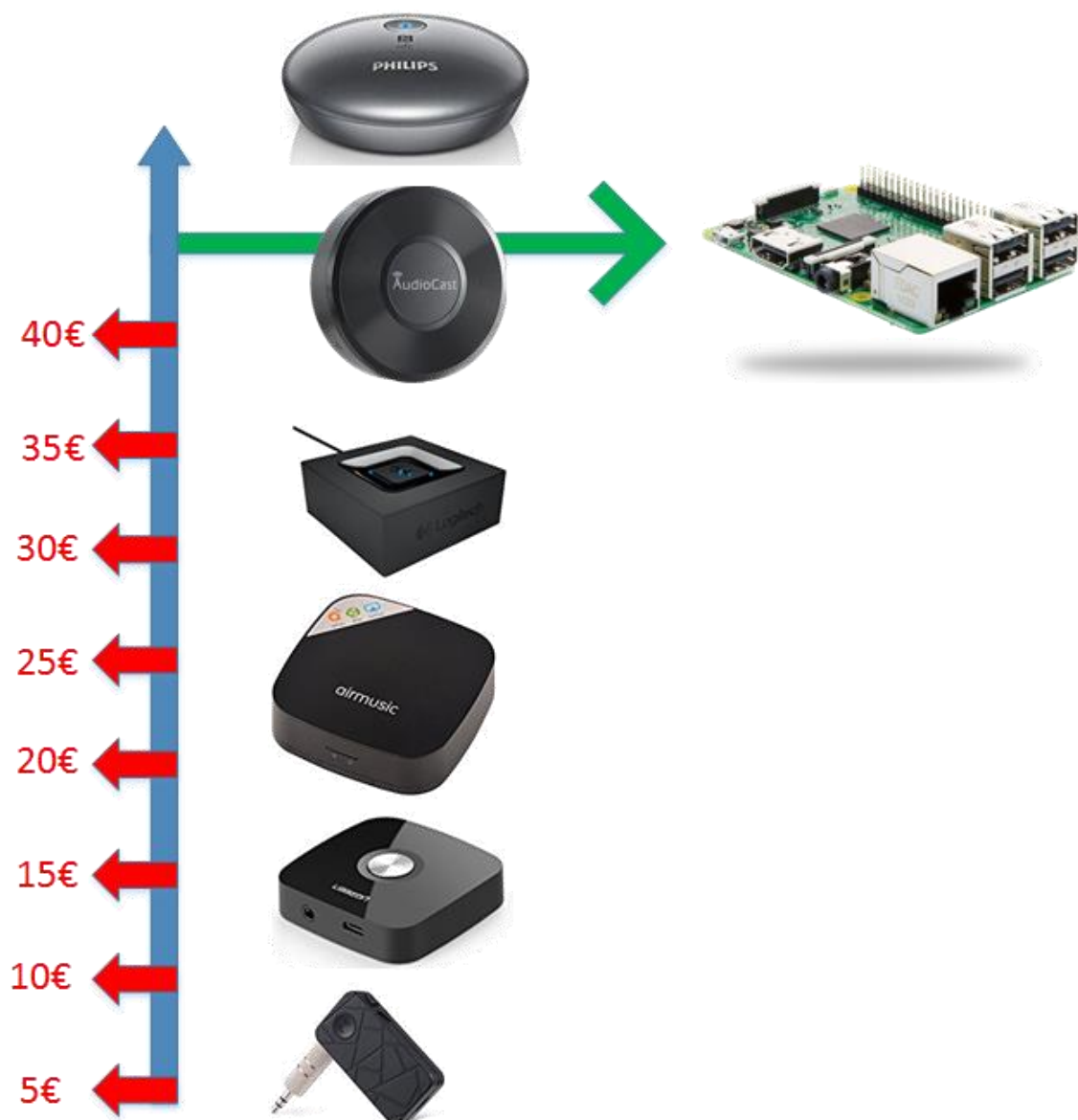


Figure 40 : Schéma montrant où se situe notre produit en terme de prix.

Placement de notre produit sur une échelle de fonctionnalités



Figure 41 : Schéma montrant où se situe notre produit en terme de fonctionnalités (nombre de protocoles implémentés).

8. Conclusion

Nous avons reçu du matériel de la part de l'université, notamment 2 versions de Raspberry ce qui nous a permis de comparer leurs performances mais nous avons aussi eu à acheter du matériel supplémentaire pour avancer dans le projet.

Nous nous sommes orienté vers la distribution GNU/Linux nommé Raspbian pour sa stabilité et la présence des paquets dont nous avons besoin, en même temps nous avons essayé de comprendre en quoi Raspbian diffère d'un Debian classique. Nous avons utilisé une multitude de paquets comme shairport-sync, pulseaudio, hostpad, apache, php et d'autres. Airplay étant un protocole propriétaire nous pensions avoir du mal à trouver un paquet assurant ses fonctions mais shairport-sync est une implémentation rétro-ingénérée de Airplay.

L'étude approfondie des 2 protocoles de diffusion de flux musical que nous avons utilisés, à savoir Airplay et Bluetooth nous ont permis de bien cerner leurs spécificités, points fort et faible.

L'apprentissage du langage PHP nous a permis de faire interagir des scripts Bash avec la page Web, ce qui a mené à l'élaboration d'une interface usager-serveur ou nous pouvons avoir un retour sur les services actifs du Raspberry et contrôler le volume. Des problèmes sont apparus au niveau de la gestion des droits notamment avec l'utilisateur système *www-data* et nous n'avons pas pu effectuer de reboot du Raspberry depuis l'interface Web. L'utilisation d'un Template a permis une économie de temps et de travailler en détail sur les feuilles de style CSS grâce à l'inspecteur d'élément.

Après avoir bien défini ce qu'était la qualité audio et plus particulièrement dans un réseau nous avons appris qu'il ne faut pas confondre avec la qualité sonore, nous avons aussi effectué une analyse spectrale grâce au logiciel Audacity qui nous a montré une ressemblance spectrale entre les 2 protocoles.

Au niveau de la consommation électrique le Raspberry varie entre 0,5W et 3,5W, ce qui nous amène à un cout annuel d'à peu près 5€ (<http://ludoviclemaire.fr/le-raspberry-pi-et-sa-consommation-electrique/>).

Notre produit que l'on estimerait à un prix de 45€ est placé parmi les plus cher mais peu de produits propose une multitude de protocoles (Airplay, Bluetooth et DLNA, que nous n'avons pas eu le temps de faire).

D'un point de vue personnel ce projet a été très enrichissant sur beaucoup de point : apprendre à gérer un planning en ayant des contraintes de temps, devoir se documenter sur des points assez techniques, produire un rapport synthétique qui en même temps montre l'ensemble des manipulations et se préparer pour l'oral qui devra être encore plus concis.

Annexes

Annexe 1 : Présentation détaillée du matériel.

Matériel fourni par l'IUT

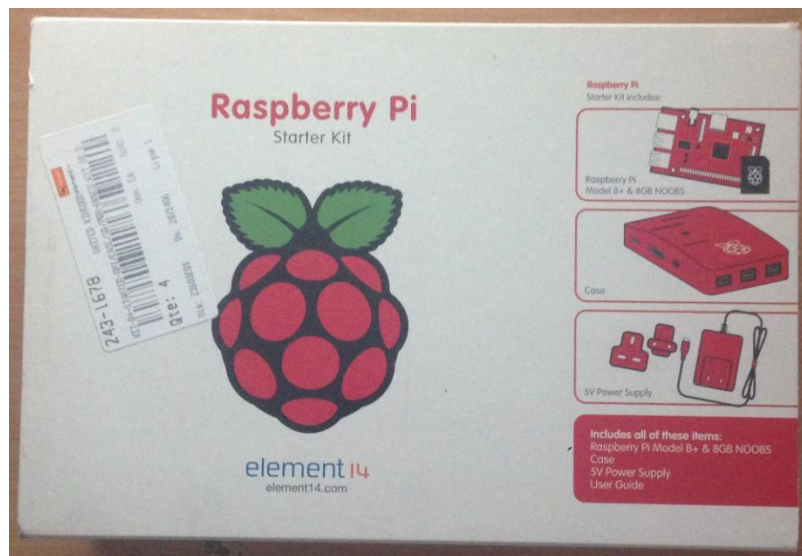


Figure 42 : Raspberry Pi Starter Kit, contient la carte Raspberry Pi B+, une carte SD de 8Gb avec NOOBS préinstallé dessus, le boîtier pour la carte, le chargeur EU/US, un manuel d'utilisation.



Figure 43 : La carte Raspberry Pi B+ dans son boîtier.



Figure 44 : La carte SD d'une capacité de 8Gb avec NOOBS préinstallé dessus.



Figure 45 : Le chargeur pour le Raspberry Pi B+ avec les adaptateurs EU/US.

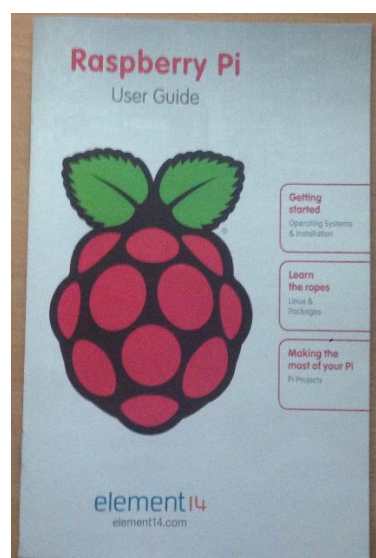


Figure 46 : Le manuel d'utilisation pour le Raspberry Pi.



Figure 47 : Le Raspberry Pi B+ et le Pi 3, la différence cruciale pour notre projet est que le Pi 3 intègre nativement le Wi-Fi et le Bluetooth.

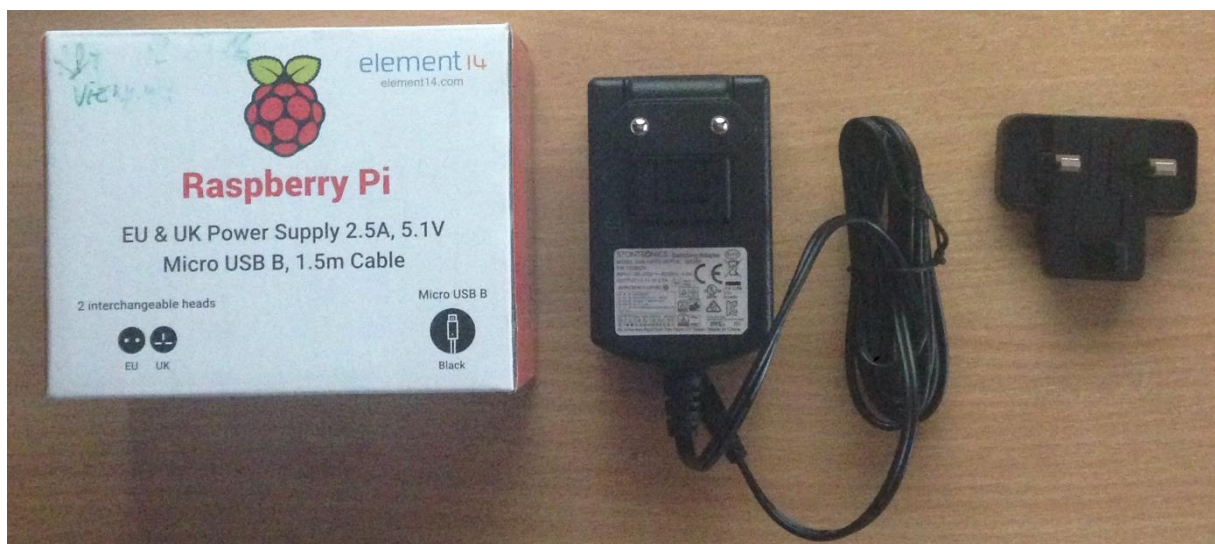


Figure 48 : Le chargeur délivré avec le Raspberry Pi 3, comparé avec l'autre chargeur la seule différence est l'ampérage qui est de 2,5A pour le Pi 3 et de 2A pour le Pi B+.

Matériel fourni par nos soins (acheté ou déjà en possession)



Figure 49 : Lecteur de carte micro-SD, Samsung Carte Mémoire EVO Micro-SD Classe 10, 16 Go, valeur estimée = 15€ (pour le Pi B+) ; Carte Mémoire Sandisk Micro SDHC, 16 Go (pour le Pi 3), valeur estimée = 10€.



Figure 50 : Le Raspberry Pi B+ étant dépourvu d'antennes Wi-Fi/Bluetooth, il a fallu rajouter un dongle Wi-Fi, TP-Link TL-WN725N Nano Adaptateur USB Wi-Fi N 150 Mbps, valeur estimée = 8€.



Figure 51 : Dongle Bluetooth pour Raspberry Pi B+, TRIXES Dongle 2.0 de Bluetooth pour Micro ou Raspberry Pi, valeur estimée = 5€.



Figure 52 : Un clavier et une souris qui ne sont utiles que pour la première utilisation c'est à dire l'installation et la configuration du Raspberry.



Figure 53 : Un écran qui est aussi utile seulement pour les premières utilisations du Raspberry.



Figure 54 : Un câble d'alimentation pour l'écran, un câble DVI pour brancher le Raspberry à l'écran et avec l'adaptateur HDMI, un câble Ethernet pour accéder à Internet et faire les mises à jour.



Figure 55 : iFun4U USB multimètre voltmètre ampère, cela va nous permettre de simplement mesurer la consommation électrique du Raspberry Pi, valeur estimée = 9€.

Annexe 2 : Test de Benchmarking effectuées sur les 2 Raspberry (Pi B+ et Pi 3).

Script de Benchmarking utilisé.

```
#!/bin/bash

[ "$(whoami)" == "root" ] || { echo "Must be run as sudo!"; exit 1; }

# Install dependencies
if [ ! `which hdparm` ]; then
    apt-get install -y hdparm
fi
if [ ! `which sysbench` ]; then
    apt-get install -y sysbench
fi
if [ ! `which speedtest-cli` ]; then
    apt-get install -y speedtest-cli
fi

# Script start!
clear
sync
echo -e "\e[96mRaspberry Pi Benchmark Test"
echo -e "Author: AikonCWD"
echo -e "Version: 3.0\n\e[97m"

# Show current hardware
vcgencmd measure_temp
vcgencmd get_config int | grep arm_freq
vcgencmd get_config int | grep core_freq
vcgencmd get_config int | grep sdram_freq
vcgencmd get_config int | grep gpu_freq
printf "sd_clock="
grep "actual_clock" /sys/kernel/debug/mmc0/ios 2>/dev/null | awk '{printf("%0.3f MHz", $3/1000000)}'
echo -e "\n\e[93m"

echo -e "Running InternetSpeed test...\e[94m"
speedtest-cli --simple
echo -e "\e[93m"

echo -e "Running CPU test...\e[94m"
sysbench --num-threads=4 --validate=on --test=cpu --cpu-max-prime=5000 run | grep 'total time:\|min:\|avg:\|max:' | tr -s
[:space:]
vcgencmd measure_temp
echo -e "\e[93m"

echo -e "Running THREADS test...\e[94m"
sysbench --num-threads=4 --validate=on --test=threads --thread-yields=4000 --thread-locks=6 run | grep 'total time:\|min:\|avg:\|
max:' | tr -s [:space:]
vcgencmd measure_temp
echo -e "\e[93m"

echo -e "Running MEMORY test...\e[94m"
sysbench --num-threads=4 --validate=on --test=memory --memory-block-size=1K --memory-total-size=3G --memory-access-mode=seq run
| grep 'Operations\|transferred\|total time:\|min:\|avg:\|max:' | tr -s [:space:]
vcgencmd measure_temp
echo -e "\e[93m"

echo -e "Running HDPARM test...\e[94m"
hdparm -t /dev/mmcblk0 | grep Timing
vcgencmd measure_temp
echo -e "\e[93m"

echo -e "Running DD WRITE test...\e[94m"
rm -f ~/test.tmp && sync && dd if=/dev/zero of=~/test.tmp bs=1M count=512 conv=fsync 2>&1 | grep -v records
vcgencmd measure_temp
echo -e "\e[93m"

echo -e "Running DD READ test...\e[94m"
echo -e 3 > /proc/sys/vm/drop_caches && sync && dd if=~/test.tmp of=/dev/null bs=1M 2>&1 | grep -v records
vcgencmd measure_temp
rm -f ~/test.tmp
echo -e "\e[90m"
```

Figure 56 : Script que nous avons utilisé pour réaliser les tests de performances.

Résultats bruts du test de Benchmarking

Raspberry Pi B+	Raspberry Pi 3
temp=46.5'C sd_clock=50.000 MHz	temp=51.5'C arm_freq=1200 core_freq=400 sdram_freq=400 gpu_freq=300 sd_clock=50.000 MHz
Running InternetSpeed test... Ping: 17.421 ms Download: 38.92 Mbits/s Upload: 19.20 Mbits/s	Running InternetSpeed test... Ping: 3.609 ms Download: 91.75 Mbits/s Upload: 89.97 Mbits/s
Running CPU test... total time: 221.6008s min: 46.98ms avg: 88.59ms max: 172.99ms temp=47.1'C	Running CPU test... total time: 17.5071s min: 6.96ms avg: 7.00ms max: 11.51ms temp=67.7'C
Running THREADS test... total time: 315.9794s min: 104.21ms avg: 126.38ms max: 190.87ms temp=47.6'C	Running THREADS test... total time: 11.7929s min: 4.58ms avg: 4.72ms max: 14.95ms temp=70.9'C
Running MEMORY test... Operations performed: 3145728 (75930.56 ops/sec) 3072.00 MB transferred (74.15 MB/sec) total time: 41.4290s min: 0.01ms avg: 0.04ms max: 70.82ms temp=48.2'C	Running MEMORY test... Operations performed: 3145728 (1260264.79 ops/sec) 3072.00 MB transferred (1230.73 MB/sec) total time: 2.4961s min: 0.00ms avg: 0.00ms max: 10.08ms temp=72.5'C
Running HDPARM test... Timing buffered disk reads: 60 MB in 3.09 seconds = 19.39 MB/sec temp=47.6'C	Running HDPARM test... Timing buffered disk reads: 66 MB in 3.06 seconds = 21.54 MB/sec temp=62.3'C
Running DD WRITE test... 512+0 enregistrements lus 512+0 enregistrements écrits 536870912 octets (537 MB) copiés, 54,5738 s, 9,8 MB/s temp=47.1'C	Running DD WRITE test... 512+0 enregistrements lus 512+0 enregistrements écrits 536870912 octets (537 MB) copiés, 49,4954 s, 10,8 MB/s temp=52.6'C
Running DD READ test... 512+0 enregistrements lus 512+0 enregistrements écrits 536870912 octets (537 MB) copiés, 25,6334 s, 20,9 MB/s temp=48.2'C	Running DD READ test... 512+0 enregistrements lus 512+0 enregistrements écrits 536870912 octets (537 MB) copiés, 24,0783 s, 22,3 MB/s temp=52.6'C

Figure 57 : Résultats bruts des tests effectués sur le Pi B+ et le Pi 3.

Quelques caractéristiques

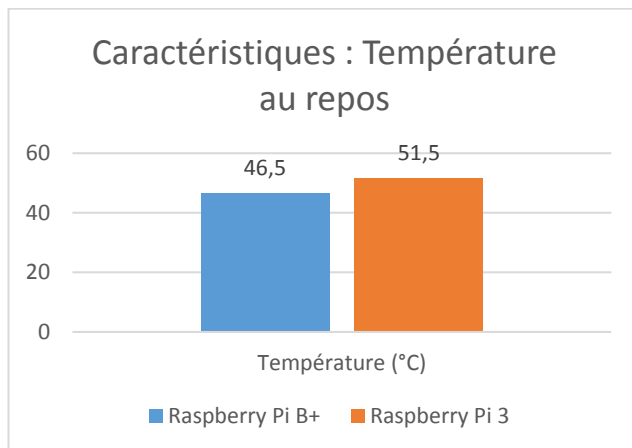


Figure 58 : Test de température au repos

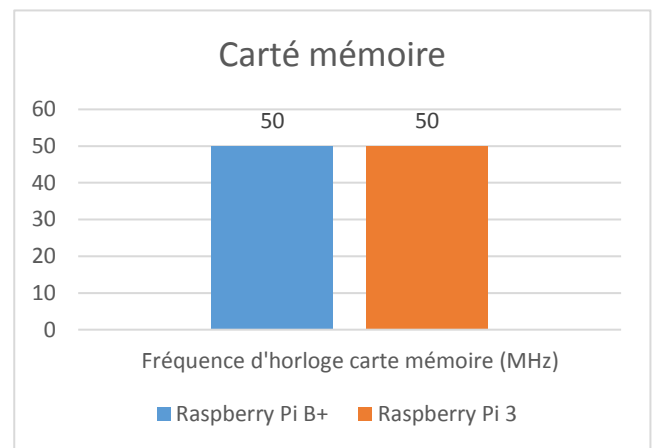


Figure 59 : Test sur les cartes mémoires (fréquence d'horloge en MHz).

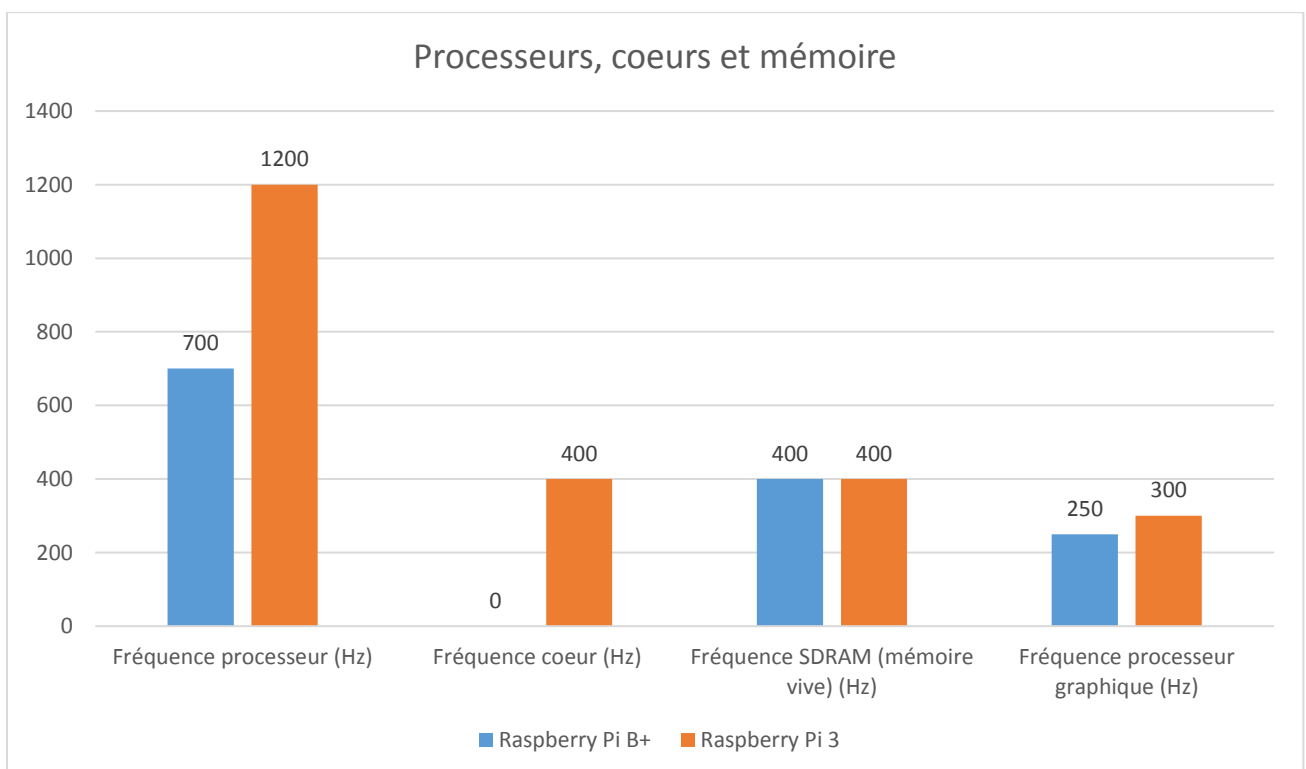


Figure 60 : Différences de processeurs, coeurs et de mémoire. Si la fréquence coeur du Pi B+ est à 0 c'est car celui-ci ne possède qu'un seul coeur.

Tests de connexion

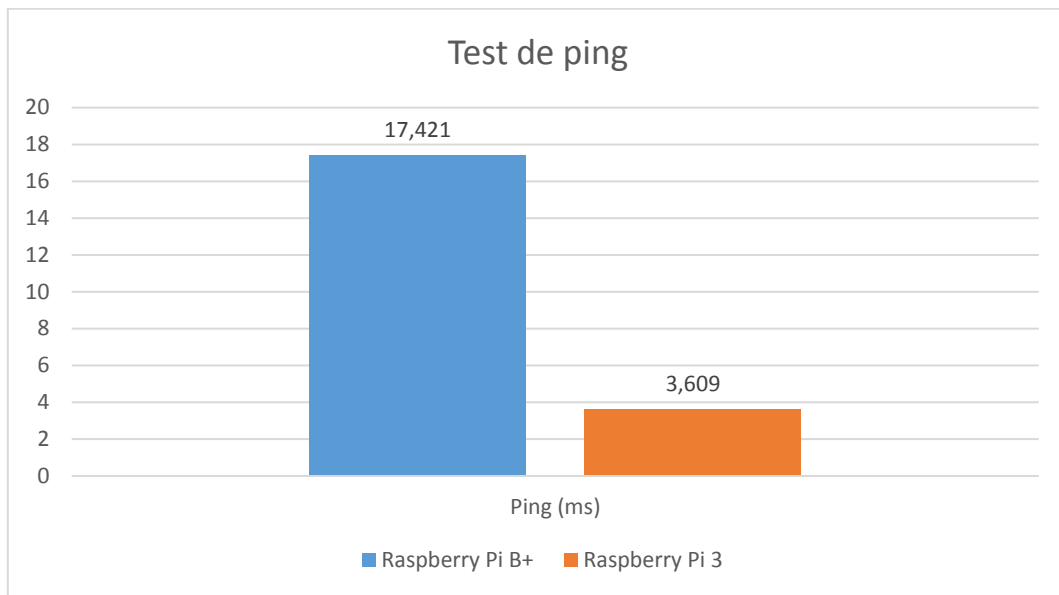


Figure 61 : Test de ping entre les 2 versions du Raspberry Pi. On peut voir que le Pi 3 a un ping beaucoup moins élevé.

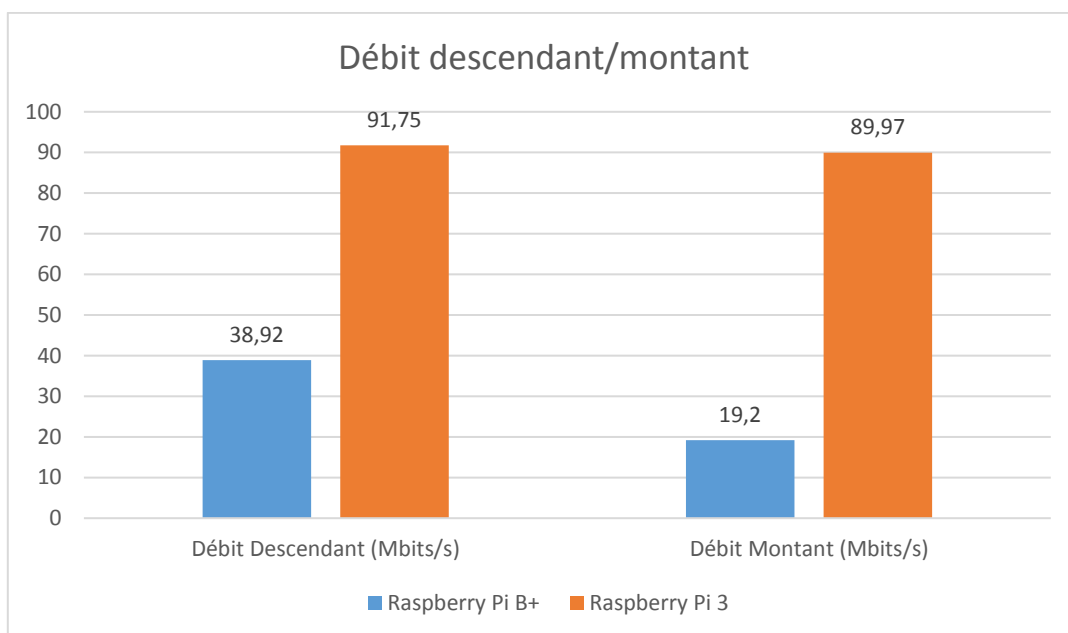


Figure 62 : Test comparatif sur les débits descendant/montant. On peut observer que le Pi 3 a en moyenne un débit 3 fois plus élevé que le Pi B+.

Test de processeur (CPU)

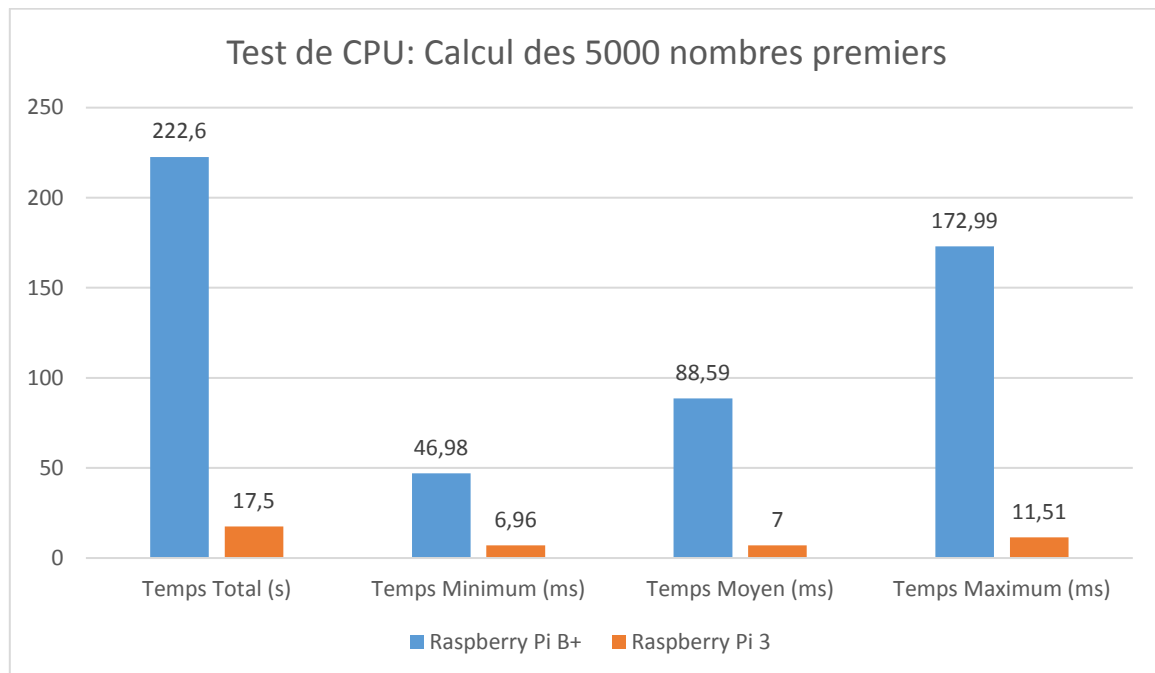


Figure 63 : Le premier test de CPU montre que le Pi 3 est largement devant. Ces résultats ne sont pas étonnants car il était prévu que le Pi 3 soit 10 fois plus performant, d'après : <https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/>

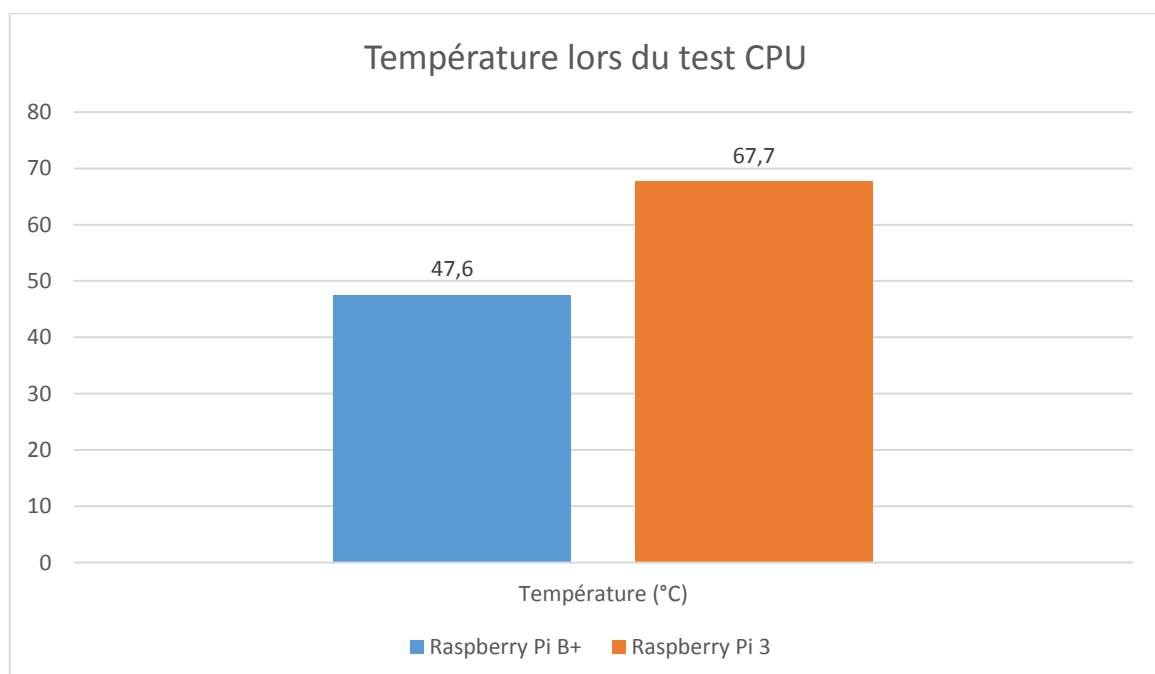


Figure 64 : En contrepartie, le Pi B+ chauffe beaucoup moins que le Pi 3 en période d'activité.

Test de Multi-Threads

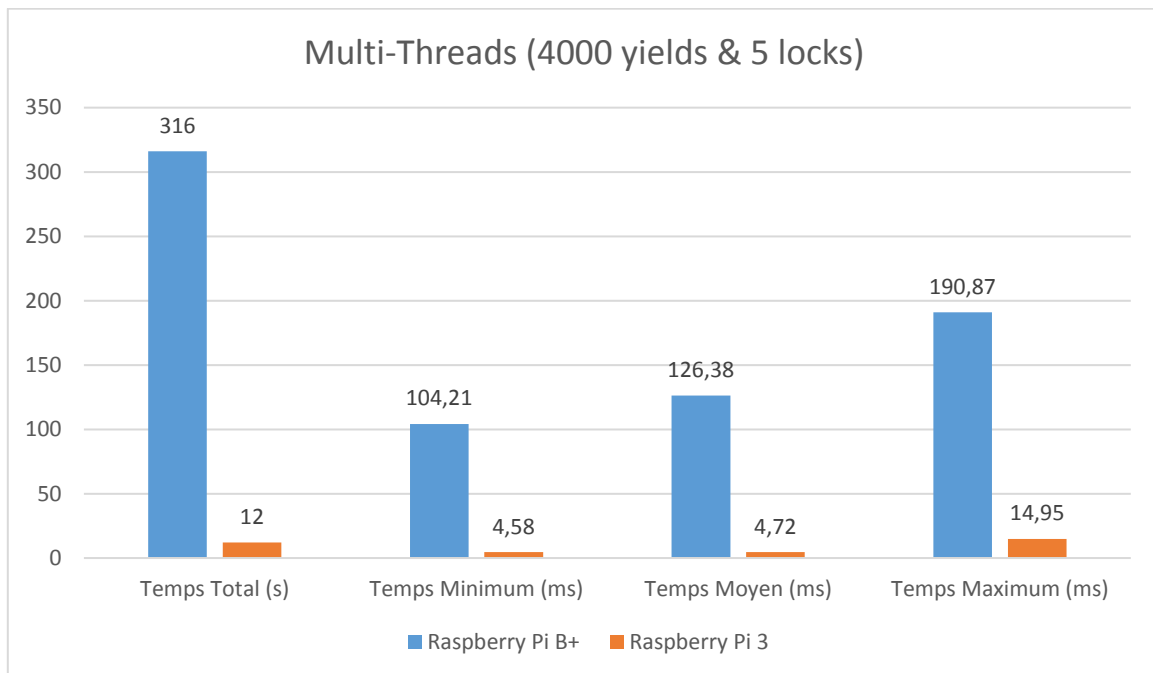


Figure 65 : Encore une fois le Raspberry Pi 3 montre que ses performances CPU sont bien plus élevées que le Pi B+.

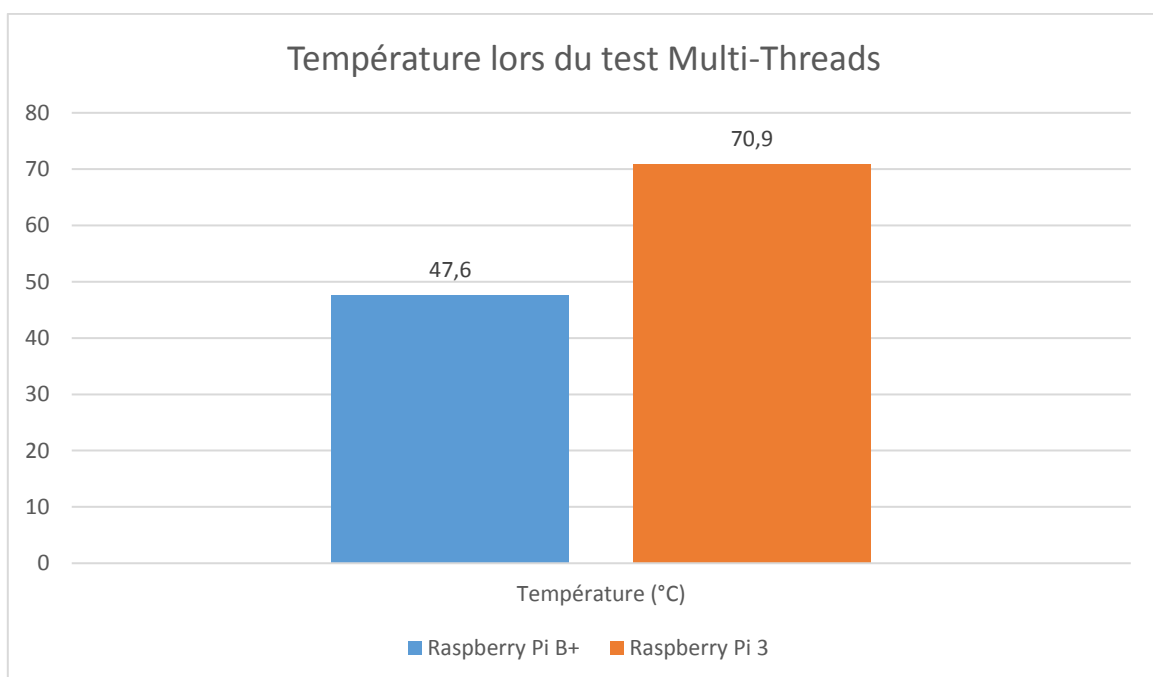


Figure 66 : La température du Pi B+ reste constante alors que le Pi 3 atteint plus de 70°C.

Test de mémoire (RAM)

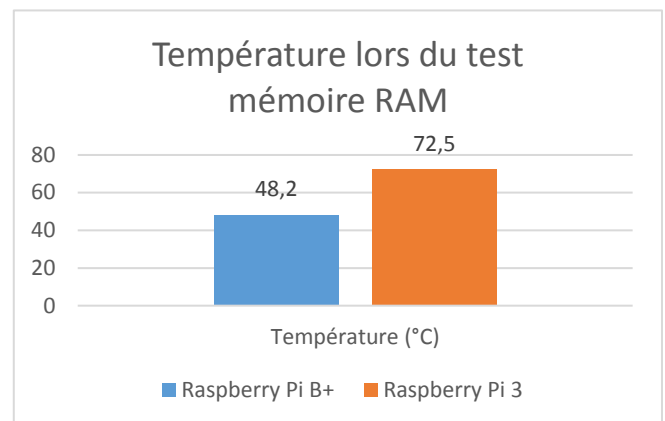
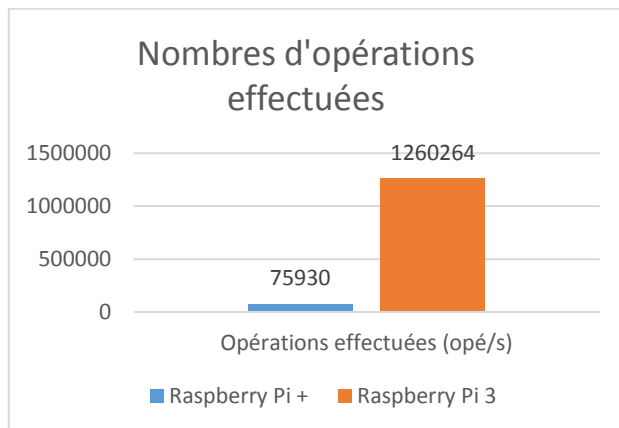


Figure 67 : La RAM allant de pair avec le processeur, il n'est pas étonnant que le Pi 3 soit largement devant.

Figure 68 : Depuis le début de ces tests la température du Pi B+ n'a connu aucune réelle augmentation comparé au Pi 3.

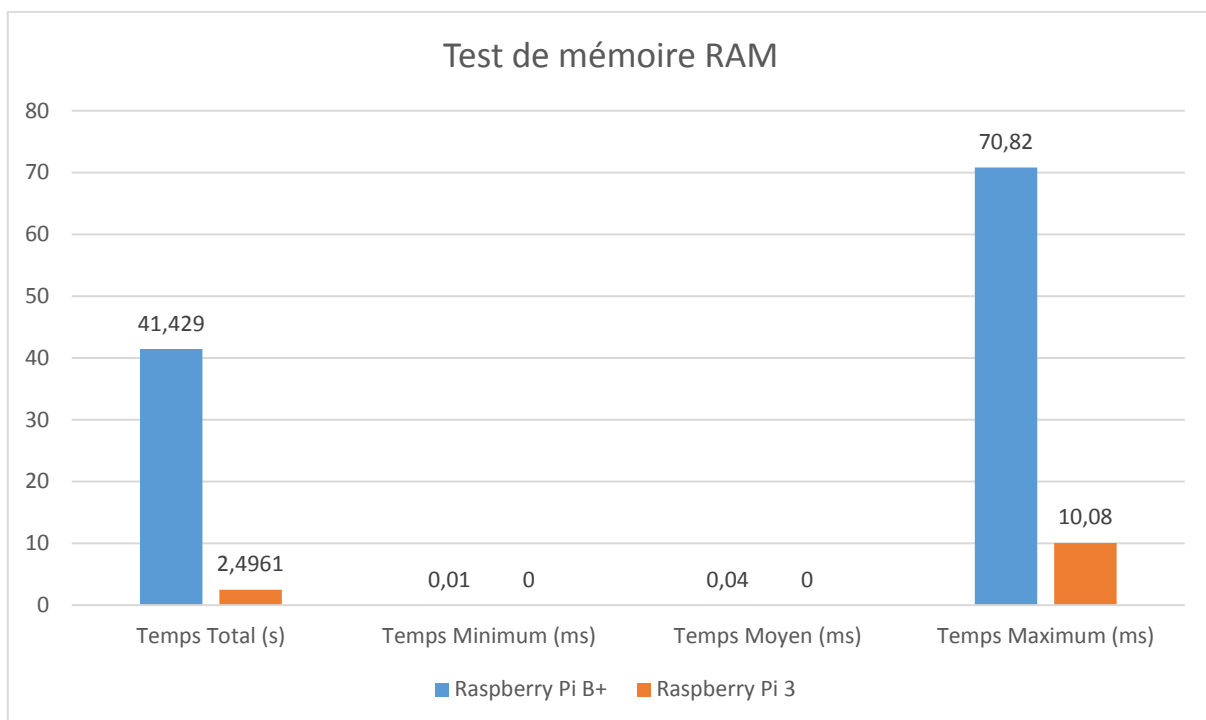


Figure 69 : Test de la mémoire RAM avec les temps total, minimum, maximum et moyen en ms.

Test sur la vitesse lecture/écriture de la carte MicroSD

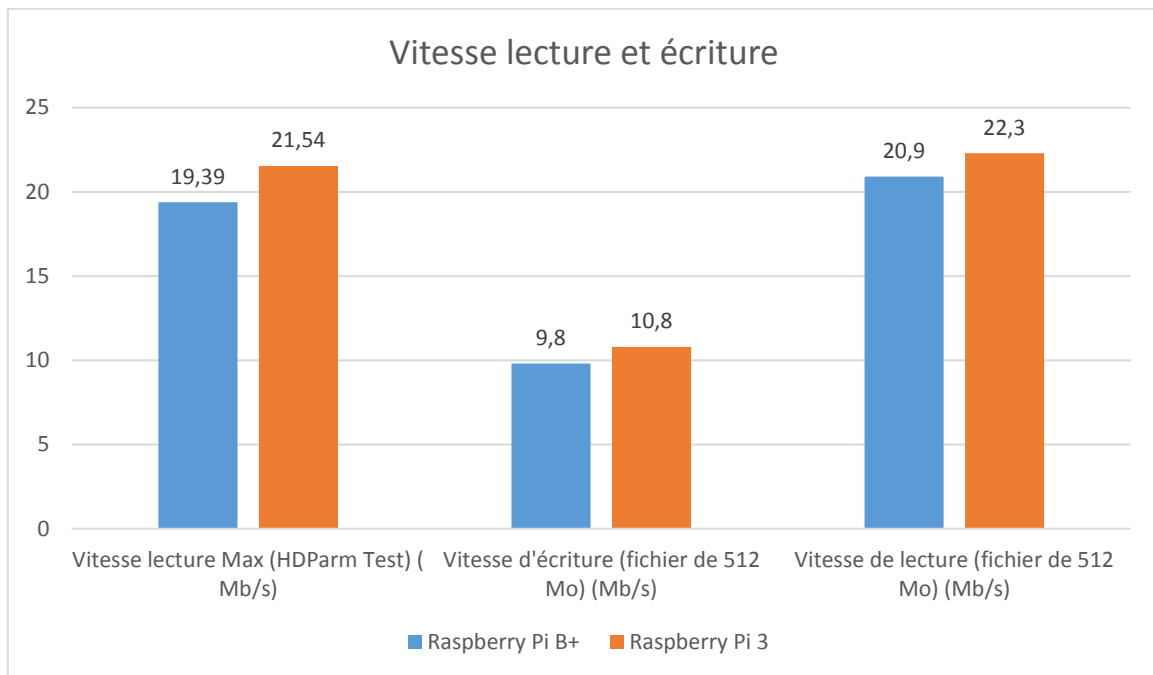


Figure 70 : Les différents tests sur la carte MicroSD. Les deux cartes étant sensiblement de même qualité on peut observer des résultats presque similaires sur les 2 versions.

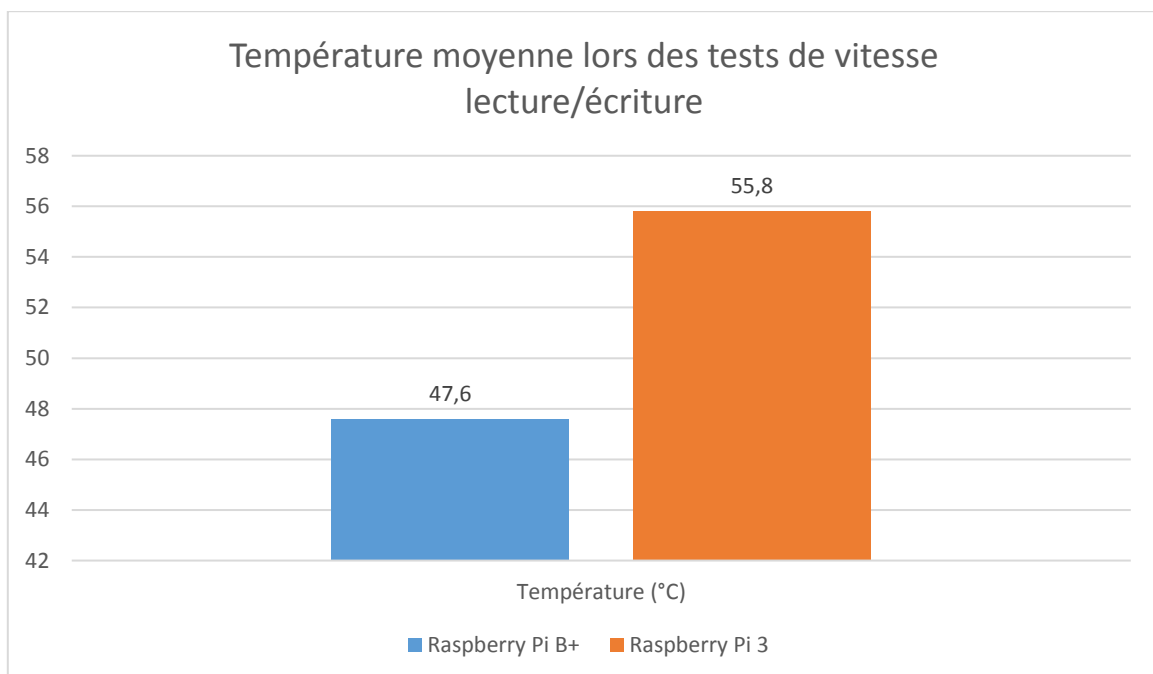


Figure 71 : Il y a une légère augmentation de température sur Pi 3 lors de ces différents tests.

Annexe 3 : Manipulations première installation et configuration de Raspbian

Mise en place matérielle

Tout d'abord nous allons préparer notre carte SD depuis Windows afin que celle-ci intègre Raspbian et non NOOBS, nous utiliserons le logiciel Win32DiskImager pour ce faire. Les performances du Raspberry Pi sont fortement influencées par la qualité de la carte SD on a donc choisi une carte SD de 16Go classe 10 (avec une vitesse d'écriture de 48Mo/s).

On a téléchargé la dernière version de Raspbian (23/09/2016) disponible sur le site officiel de Raspberry Pi : <https://www.raspberrypi.org/downloads/raspbian/>

Raspbian is the Foundation's official supported operating system. You can install it with [NOOBS](#) or download the image below and follow our [installation guide](#).

Raspbian comes pre-installed with plenty of software for education, programming and general use. It has Python, Scratch, Sonic Pi, Java, Mathematica and more.

Image Name	Version	Release Date	Kernel Version	Release Notes
RASPBIAN JESSIE WITH PIXEL	September 2016	2016-09-23	4.4	Link
RASPBIAN JESSIE LITE	September 2016	2016-09-23	4.4	Link

SHA-1: e0eeb96e2fa10b3bd4b57454317b06f5d3d09d46

SHA-1: 3a34e7b05e1e6e9042294b29065144748625bea8

Note: Raspbian and NOOBS contain Java SE Platform Products, licensed to you under the Oracle Binary Code Licence Agreement available [here](#).

Mathematica and the Wolfram Language are included in this release under license and with permission of Wolfram Research, Inc. and may be used for non-commercial purposes only. By using this software you agree to be bound by the Wolfram Raspberry Pi Bundle Licensee Agreement available [here](#).

Figure 72 : Téléchargement de Raspbian, daté du 23/09/2016 pour commencer l'installation sur le Raspberry Pi.

On a aussi téléchargé le logiciel Win32DiskImager à l'adresse suivante :

<https://sourceforge.net/projects/win32diskimager/files/latest/download>

Pour connecter la carte SD à l'ordinateur, nous utilisons un petit lecteur de carte comme montré ci-dessous :



Figure 73 : Lecteur de carte MicroSD et carte SD de 16Go pour copier l'image Raspbian sur la carte.

Après avoir installé et lancer Win32DiskImager, on décompresse le fichier .zip de l'image de Raspbian pour obtenir un fichier .img. On insère la carte SD dans le lecteur que l'on connecte à l'ordinateur, celle-ci doit bien être reconnue.

- 1 – On indique où se trouve l'image que l'on veut écrire sur la carte SD (ici 2016-09-23-Raspbian-jessie.img).
- 2 – On choisit le bon périphérique, ici notre carte SD qui est le périphérique [J::].
- 3 – On appuie sur Write.

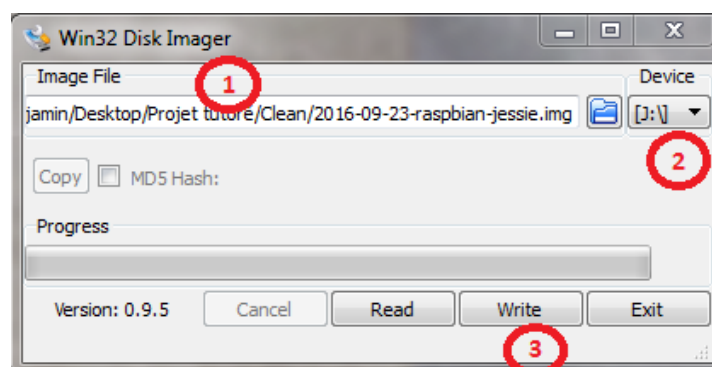


Figure 74 : Etapes pour décompresser l'image sur la carte SD.

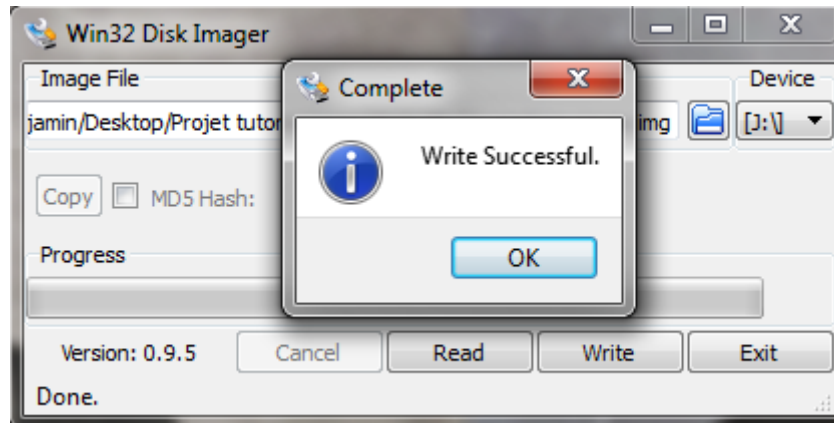


Figure 75 : Attendre la fin de l'écriture sur la carte SD.

Voici ce que contient maintenant notre carte SD :

Icon	File Name	Date	Type	Size
Folder	overlays	23/09/2016 12:26	Dossier de fichiers	
File	bcm2708-rpi-b.dtb	22/09/2016 18:07	Fichier DTB	14 Ko
File	bcm2708-rpi-b-plus.dtb	22/09/2016 18:07	Fichier DTB	14 Ko
File	bcm2708-rpi-cm.dtb	22/09/2016 18:07	Fichier DTB	14 Ko
File	bcm2709-rpi-2-b.dtb	22/09/2016 18:07	Fichier DTB	15 Ko
File	bcm2710-rpi-3-b.dtb	22/09/2016 18:07	Fichier DTB	16 Ko
File	bcm2710-rpi-cm3.dtb	22/09/2016 18:07	Fichier DTB	15 Ko
File	bootcode	22/06/2016 17:06	PowerISO File	18 Ko
File	cmdline	23/09/2016 13:49	Document texte	1 Ko
File	config	23/09/2016 12:26	Document texte	2 Ko
File	COPYING.linux	22/08/2015 02:04	Fichier LINUX	19 Ko
File	fixup.dat	22/09/2016 18:07	Fichier DAT	7 Ko
File	fixup_cd.dat	22/09/2016 18:07	Fichier DAT	3 Ko
File	fixup_db.dat	22/09/2016 18:07	Fichier DAT	10 Ko
File	fixup_x.dat	22/09/2016 18:07	Fichier DAT	10 Ko
File	issue	23/09/2016 14:02	Document texte	1 Ko
File	kernel	22/09/2016 18:07	PowerISO File	4 031 Ko
File	kernel7	22/09/2016 18:07	PowerISO File	4 130 Ko
File	LICENCE.broadcom	19/11/2015 01:01	Fichier BROADCOM	2 Ko
File	LICENCE.oracle	23/09/2016 14:02	Fichier ORACLE	19 Ko
File	start.elf	22/09/2016 18:07	Fichier ELF	2 700 Ko
File	start_cd.elf	22/09/2016 18:07	Fichier ELF	619 Ko
File	start_db.elf	22/09/2016 18:07	Fichier ELF	4 829 Ko
File	start_x.elf	22/09/2016 18:07	Fichier ELF	3 805 Ko

Contenu

Figure 76 : Notre carte SD est maintenant "bootable" et contient tout ce qu'il faut pour installer Raspbian.

Nous allons maintenant insérer la carte SD et effectuer les branchements nécessaires pour démarrer le Raspberry Pi comme montré ci-dessous :



Figure 77 : Insertion de la carte SD dans le Raspberry Pi. Sur le Pi B+ la carte se clips alors que sur le Pi 3 elle se glisse.

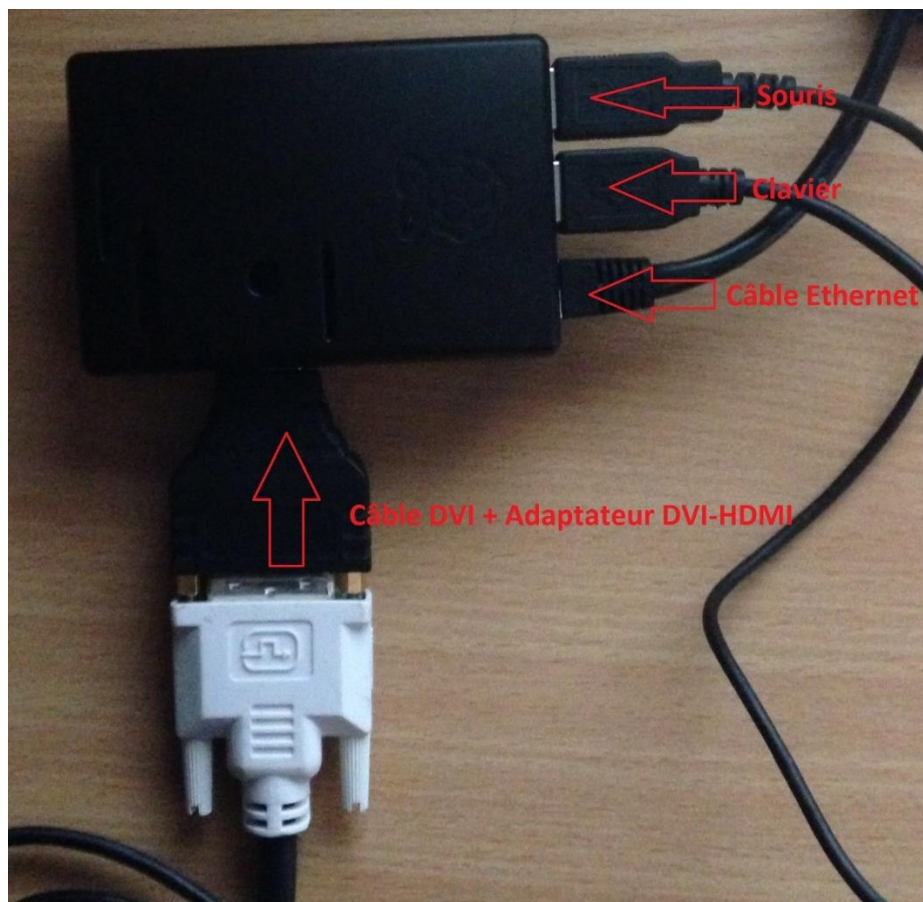


Figure 78 : Branchements des composants nécessaires à une première installation.

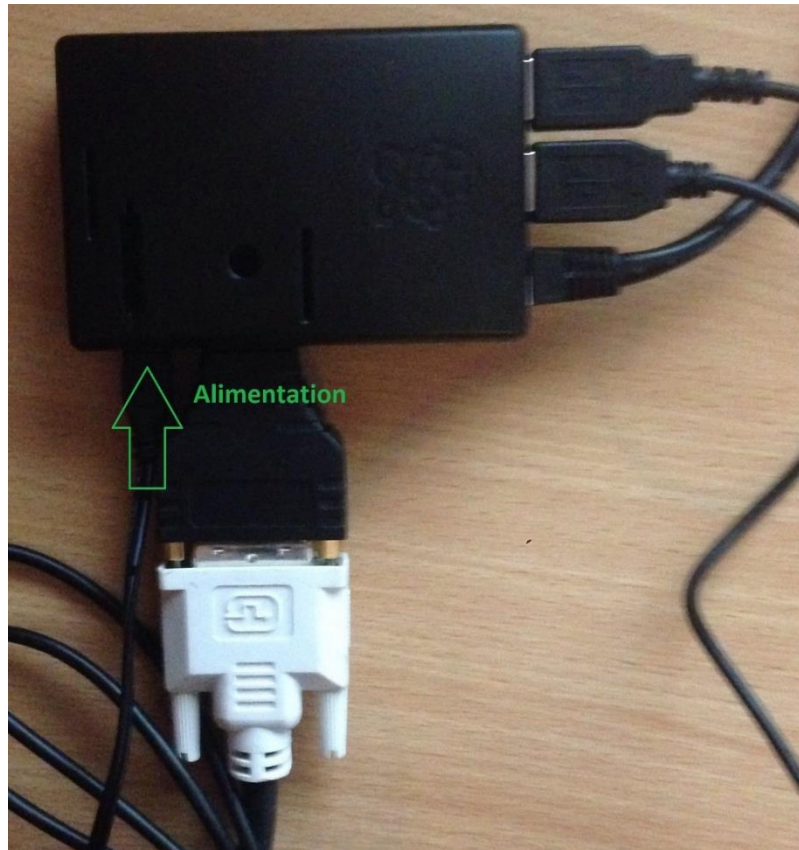


Figure 79 : Mettre sous-alimentation le Raspberry, après bien avoir mis la carte SD puis brancher les différents composants.

Au démarrage la LED PWR est rouge, ce qui signifie que tout est en ordre au niveau de l'alimentation et la LED ACT clignote vert ce qui signifie que le Raspberry a bien boot sur la carte SD. Nous avons eu un soucis avec l'ancienne carte SD qui était fourni avec le Raspberry : celle-ci ne voulait pas booter et il n'y avait aucun moyen d'y accéder physiquement depuis n'importe-quel OS et avec n'importe quel adaptateur. Nous pensons que cette carte est défectueuse.



Figure 80 : Les LED permettent de nous informer que tout se déroule bien comme prévu. A noter que sur le Pi 3 les LED ne sont plus présentes et ont été remplacées par les antennes Wi-Fi/Bluetooth.

Le Raspberry s'occupe lui-même de l'installation de Raspbian :

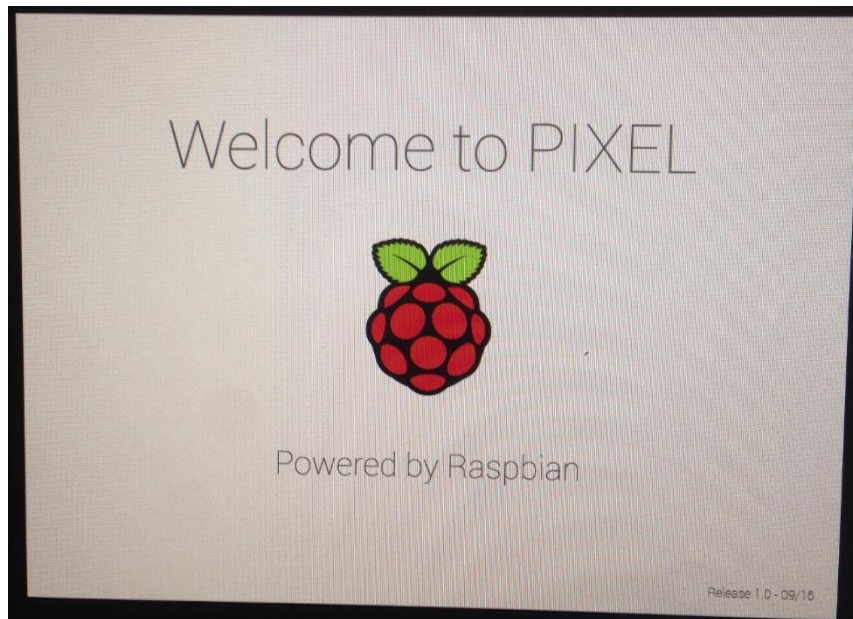


Figure 81 : Ecran d'accueil validant que Raspbian s'est bien installé.

Puis on tombe sur une belle Interface Graphique avec déjà pas mal de fonctionnalités installées : Python 2 et 3, Java IDE, Geany, Wolfram, la suite LibreOffice, Chromium et d'autres moins utiles.

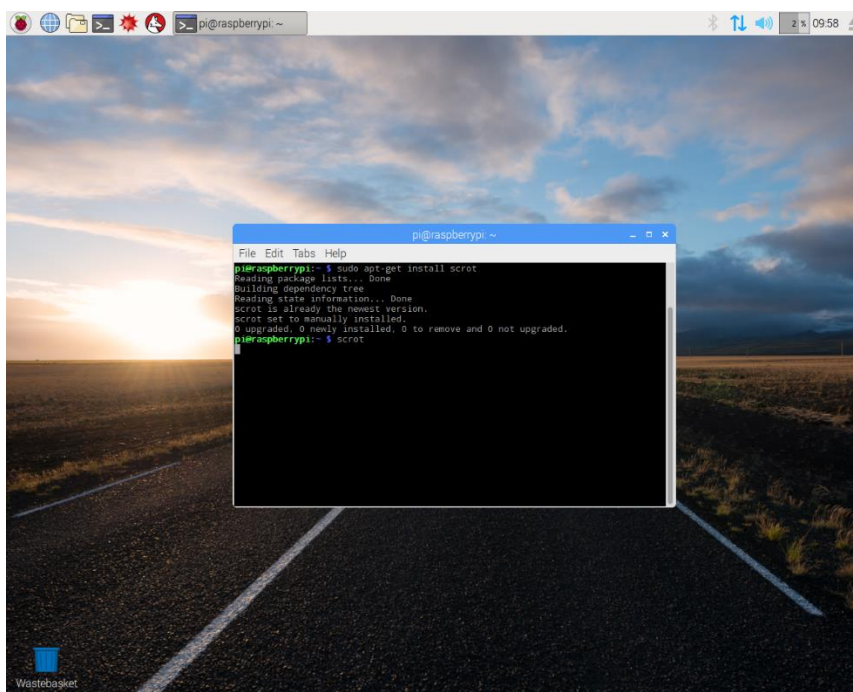


Figure 82 : Bureau du Raspberry Pi B+. Les screenshots ont été prises avec scrot.

Configuration de Raspbian

Le LOGIN par défaut est « pi » et le PASSWD est « raspberry ». Notons que le clavier est par défaut en QWERTY et non en AZERTY, il va donc falloir faire quelques réglages avant de pouvoir commencer à travailler sur le Raspberry Pi.

Pour commencer à configurer le Raspberry afin d'être dans des conditions optimales d'utilisation lors du redémarrage on ouvre un terminal et on tape la commande :

sudo raspi-config

- Tout d'abord on va vérifier que Raspbian utilise bien toute la carte SD :

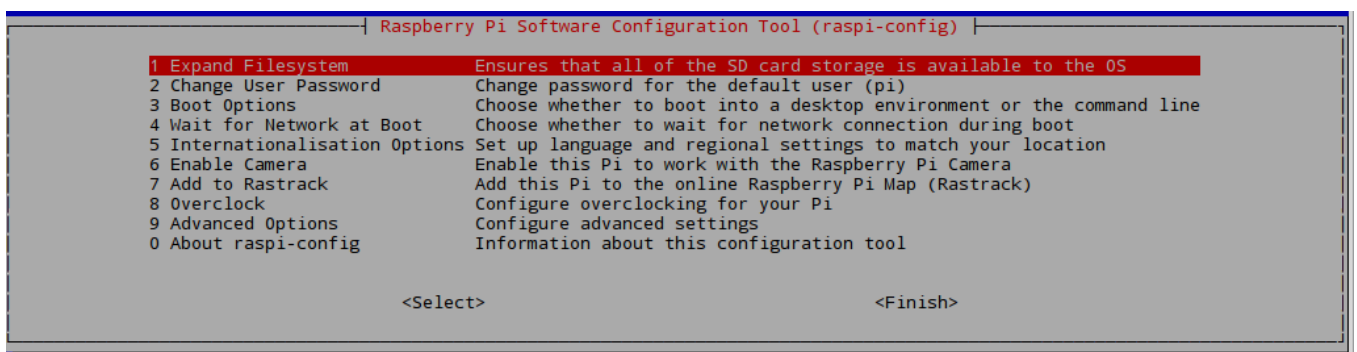


Figure 83 : Menu de configuration de Raspbian.

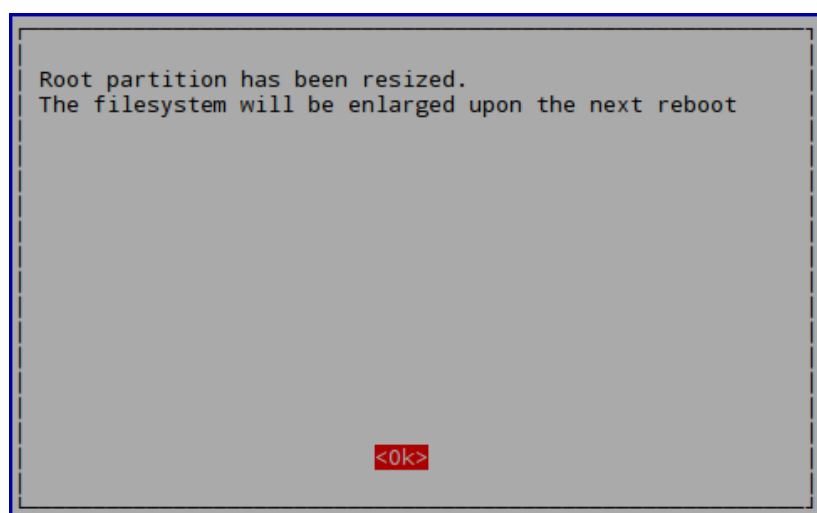


Figure 84 : On a fait "Expand Filesystem" pour que Raspbian utilise bien tout l'espace disponible.

- Ensuite nous allons passer notre clavier en AZERTY ce qui nous simplifiera les manipulations par la suite :

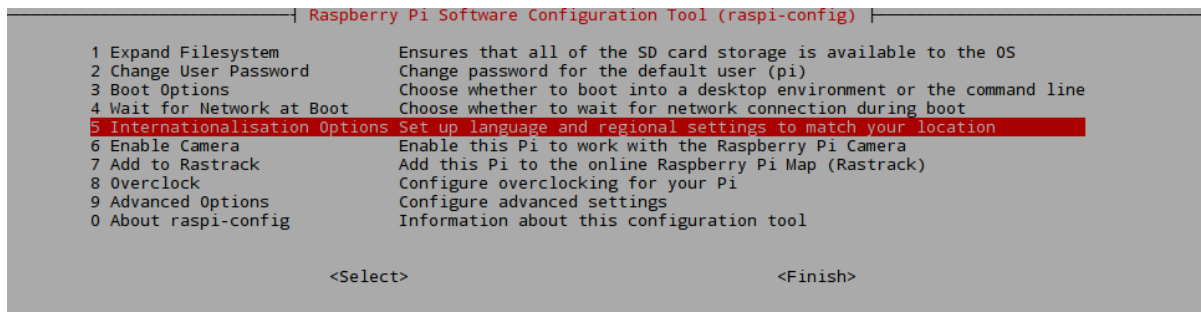


Figure 85 : Nous allons passer notre clavier en AZERTY dans le menu "Internationalisation Options".

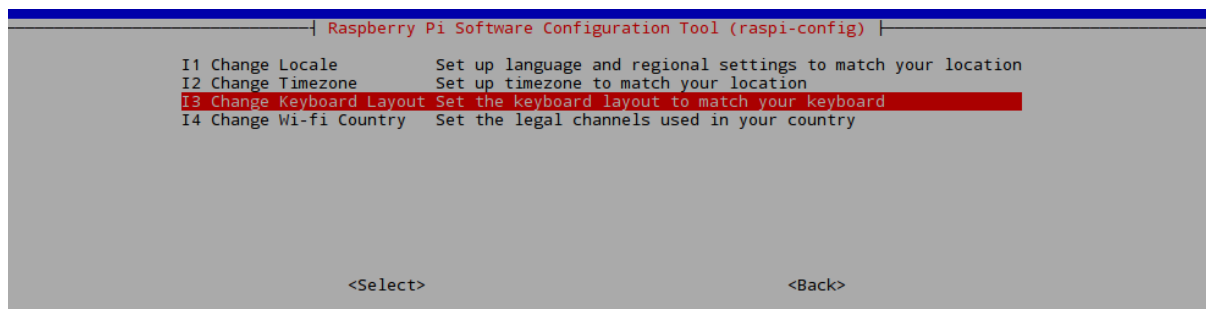


Figure 86 : On change l'agencement des touches du clavier.

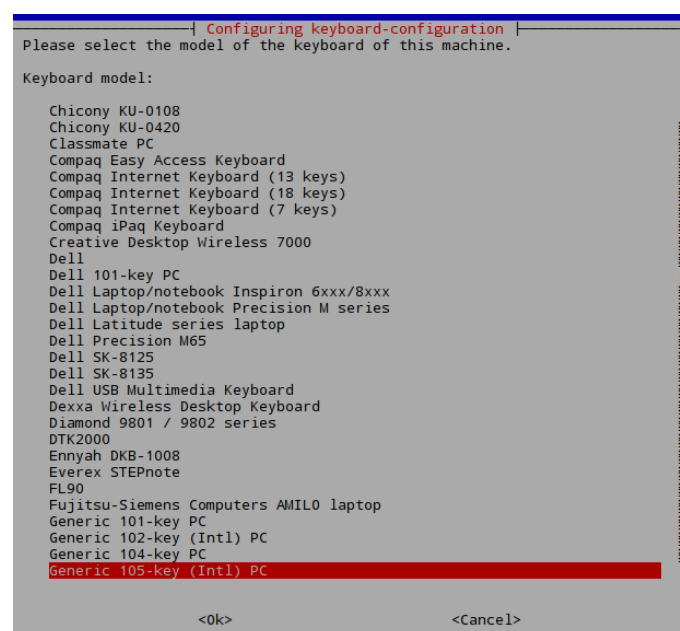


Figure 87 : On choisit "Generic 105-key PC", qui est le plus général.

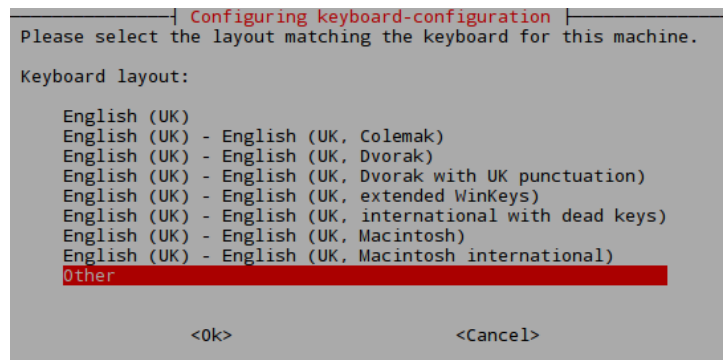


Figure 88 : Nous voulons choisir une autre langue que anglais pour le clavier.

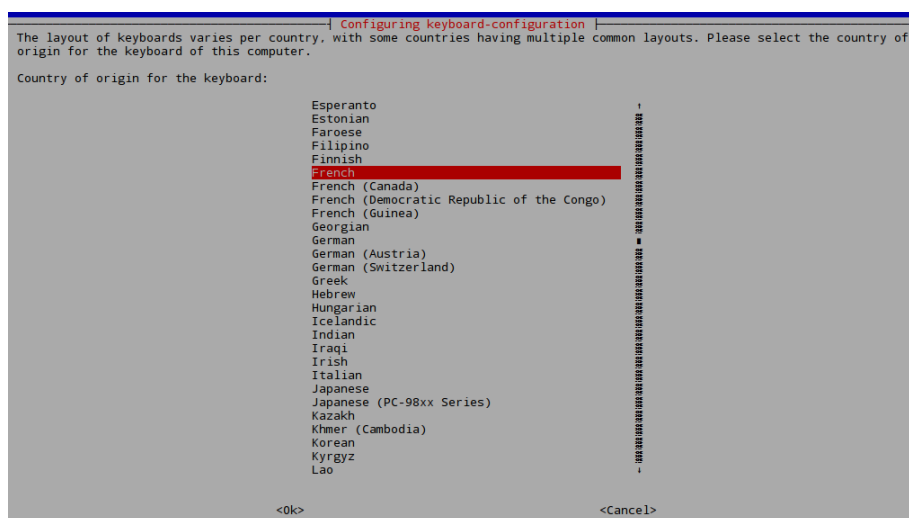


Figure 89 : On sélectionne la langue Français pour les l'agencement du clavier.

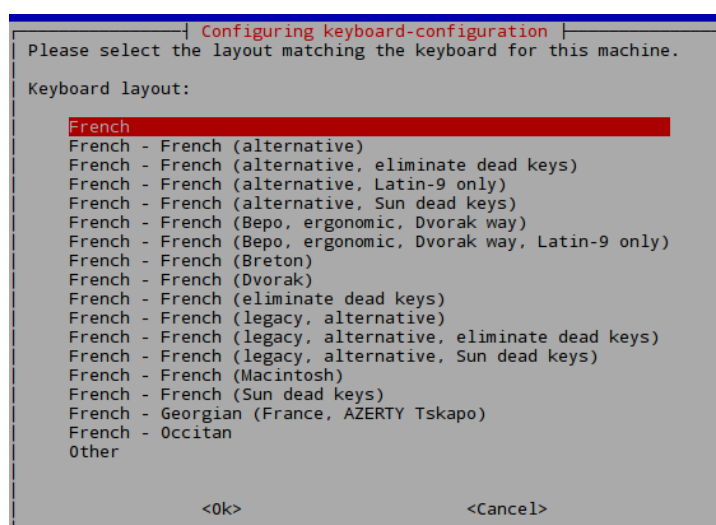


Figure 90 : On prend le Français de base.

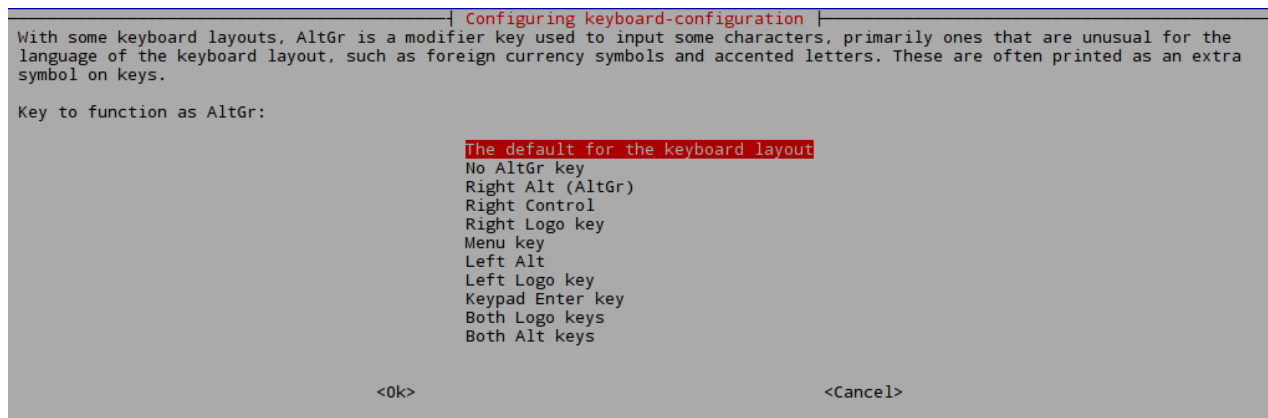


Figure 91 : On sélectionne la configuration par défaut.

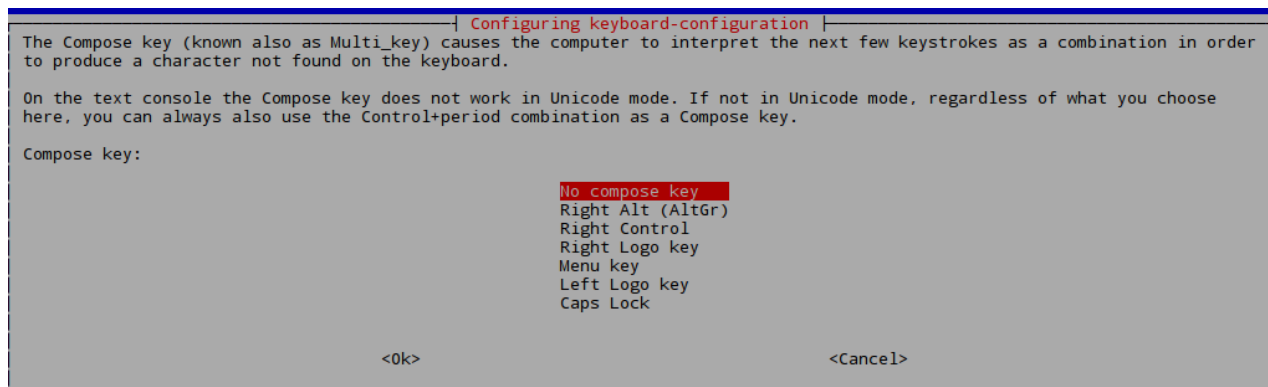


Figure 92 : On ne prend pas les touches composées.

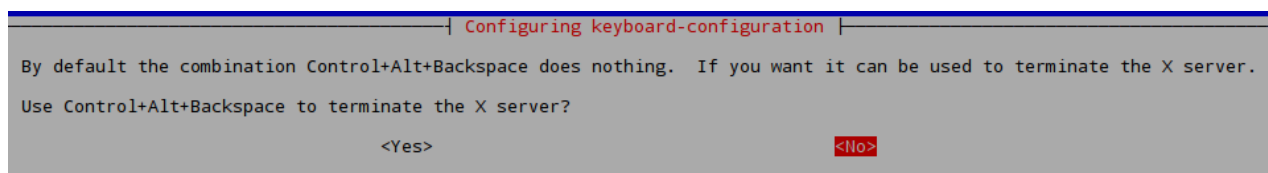


Figure 93 : On ne veut pas que Control+Alt+Backspace ferme le serveur X.

- Nous allons maintenant passer Raspbian en Français :

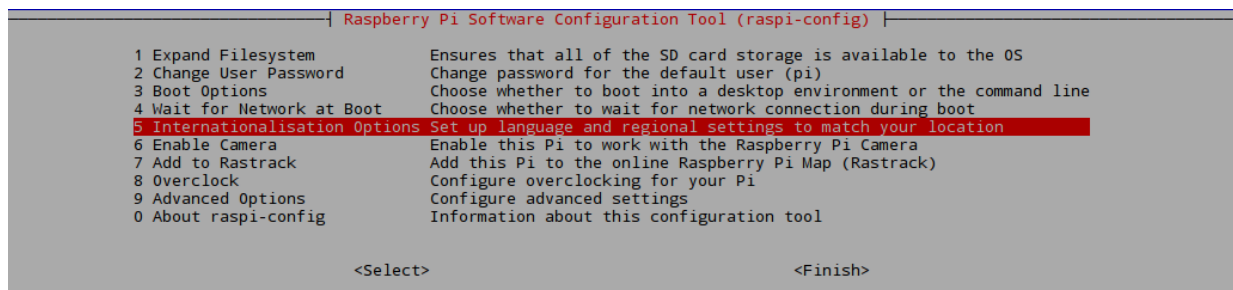


Figure 94 : On revient dans le menu "Internationalisation Options", cette fois-ci pour changer la langue du système.

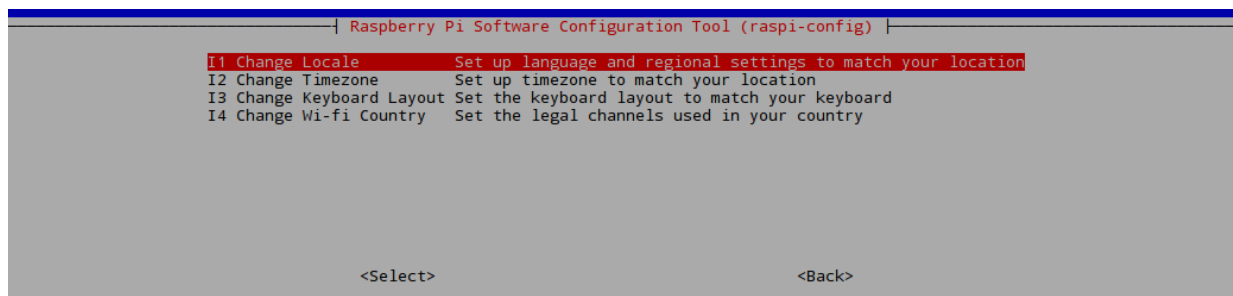


Figure 95 : On sélectionne "Change Locale" pour définir la langue par défaut.

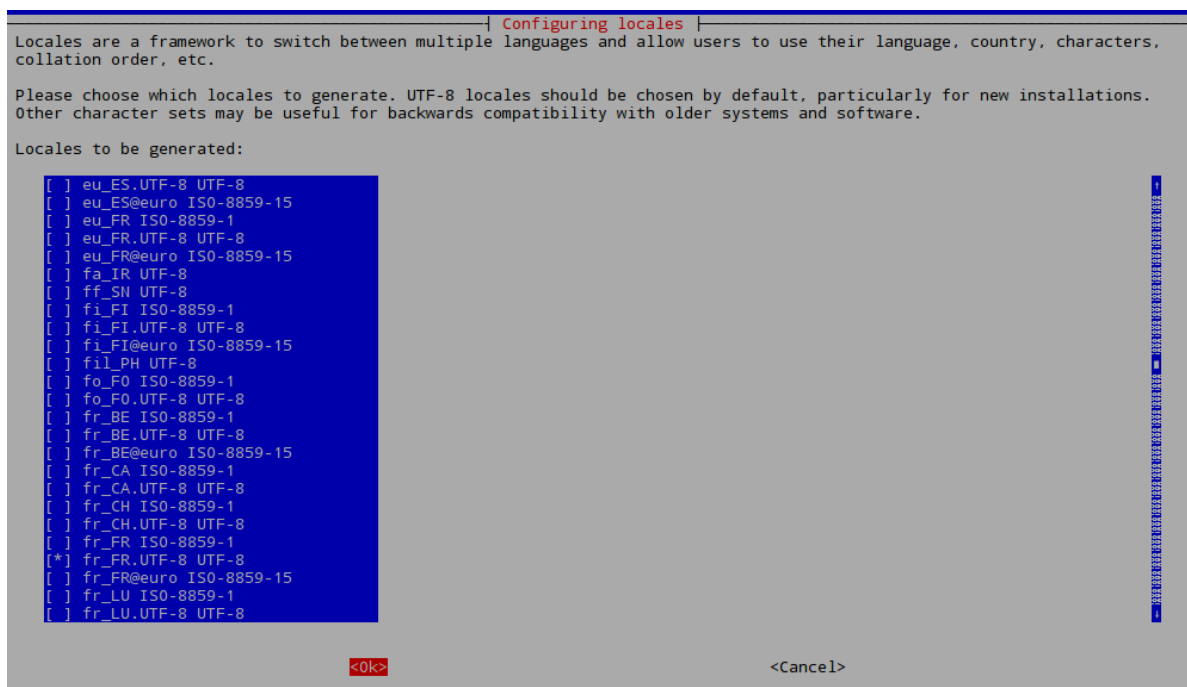


Figure 96 : On va sélectionner « fr_FR.UTF-8 UTF-8 » et « en_GB.UTF-8 UTF-8 » pour avoir les langues anglaise et française.

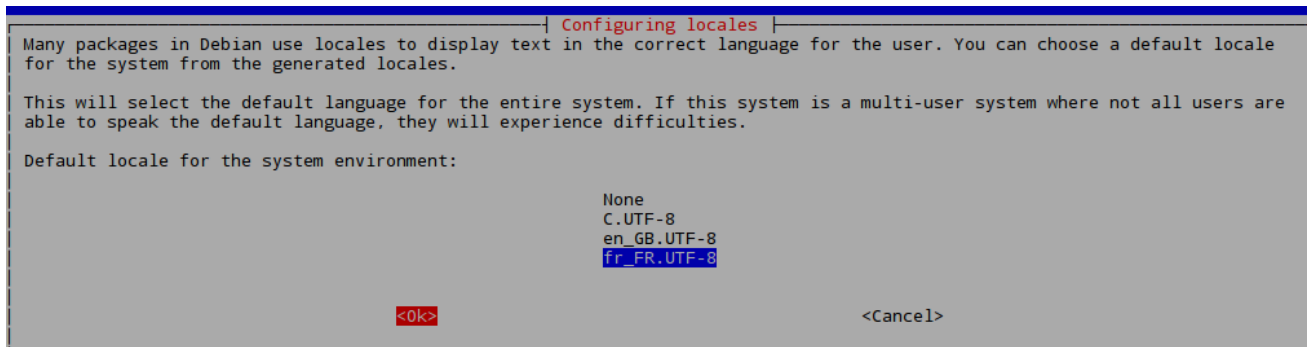


Figure 97 : On spécifie que l'on veut en français par défaut.

- Nous allons modifier le mot de passe par défaut du Raspberry pour des raisons de sécurité, vu que le Raspberry intègrera le Wi-Fi et le Bluetooth il ne faut pas qu'il soit une proie facile : nous choisissons le mot de passe « **azerty123** ».

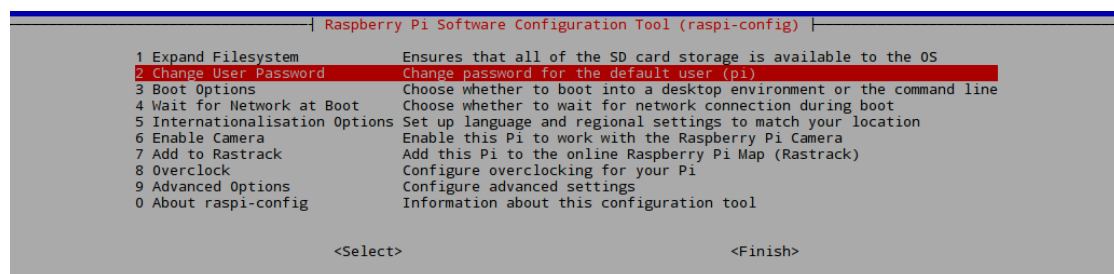


Figure 98 : Pour changer le mot de passe de l'utilisateur par défaut, à savoir "pi", il faut aller dans le menu "Change User Password".

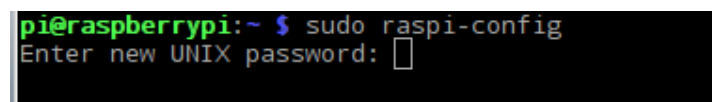


Figure 99 : On tape le mot de passe voulu.



Figure 100 : Le mot de passe a bien été changé.

- Par la même occasion on a décidé de définir un mot de passe pour root ce qui pourrait s'avérer utile par la suite. Nous choisissons le mot de passe « 012345 »

```
pi@raspberrypi:~ $ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Figure 101 : Mise en place du mot de passe root.

- Devoir utiliser un écran, un clavier, une souris rien que pour le Raspberry nous semble un peu trop donc on va configurer le SSH afin d'y accéder à distance depuis nos postes personnels :

```
Raspberry Pi Software Configuration Tool (raspi-config)
1 Expand Filesystem      Ensures that all of the SD card storage is available to the OS
2 Change User Password   Change password for the default user (pi)
3 Boot Options           Choose whether to boot into a desktop environment or the command line
4 Wait for Network at Boot Choose whether to wait for network connection during boot
5 Internationalisation Options Set up language and regional settings to match your location
6 Enable Camera          Enable this Pi to work with the Raspberry Pi Camera
7 Add to Rastrack        Add this Pi to the online Raspberry Pi Map (Rastrack)
8 Overclock              Configure overclocking for your Pi
9 Advanced Options       Configure advanced settings
0 About raspi-config     Information about this configuration tool

<Select>                                <Finish>
```

Figure 102 : Il faut aller dans "Advanced Options" pour activer le SSH sur Raspbian.

```
Raspberry Pi Software Configuration Tool (raspi-config)
A1 Overscan              You may need to configure overscan if black bars are present on display
A2 Hostname              Set the visible name for this Pi on a network
A3 Memory Split          Change the amount of memory made available to the GPU
A4 SSH                   Enable/Disable remote command line access to your Pi using SSH
A5 VNC                   Enable/Disable graphical remote access to your Pi using RealVNC
A6 SPI                   Enable/Disable automatic loading of SPI kernel module (needed for e.g. PiFace)
A7 I2C                   Enable/Disable automatic loading of I2C kernel module
A8 Serial                Enable/Disable shell and kernel messages on the serial connection
A9 Audio                 Force audio out through HDMI or 3.5mm jack
AA 1-Wire                Enable/Disable one-wire interface

<Select>                                <Back>
```

Figure 103 : On sélectionne SSH.

```
Would you like the SSH server to be enabled?

<Yes>                                <No>
```

Figure 104 : On active le SSH.

Finalement nous allons exécuter la commande suivante :

sudo aptitude update -y && sudo aptitude upgrade -y && sudo reboot

Cette commande va chercher des mises à jour, les appliquer et enfin redémarrer le Raspberry (l'argument `-y` permet de valider ces mises à jour). Les `&&` permettent d'effectuer une commande juste après une autre commande.

Maintenant que nous sommes affranchis de l'écran, du clavier et de la souris sur le Raspberry Pi, cela va réduire un peu sa consommation, nous avons juste besoin d'un câble Ethernet pour le mettre sur le réseau.

Le Protocole SSH permet de prendre la main dessus pour travailler ; mais si on avait envie d'avoir accès à l'interface graphique ce qui n'est pas possible avec SSH donc on utilise un outil très simple : VNCserver (l'interface graphique reste optionnelle).

On l'installe sur le Raspberry avec la commande : **sudo apt-get install tightvncserver**

On le lance avec la commande : **tightvncserver**

Il faut définir un mot de passe : « **012345** » puis on dit que l'on ne veut pas juste « view-only ».

Le processus est maintenant lancé et on peut y accéder depuis le réseau, par exemple sur Windows avec Google Chrome qui a une application VNC Viewer.

Exemple pour accéder au Raspberry :

```
benji@station:~$ ssh pi@10.5.1.26
pi@10.5.1.26's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct 28 16:05:26 2016
pi@raspberrypi:~$
pi@raspberrypi:~$ tightvncserver

New 'X' desktop is raspberrypi:1

Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:1.log

pi@raspberrypi:~$ █
```

Figure 105 : On est connecté en SSH sur le Raspberry Pi et on lance VNCserver.

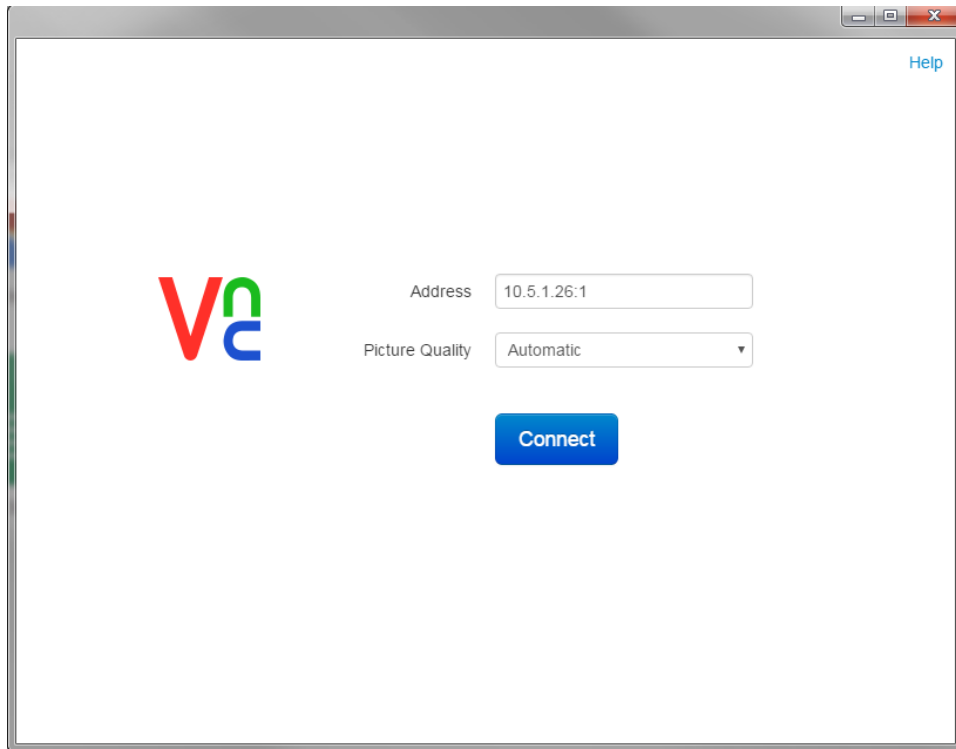


Figure 106 : Depuis chrome, on peut installer une application complémentaire VNCviewer.

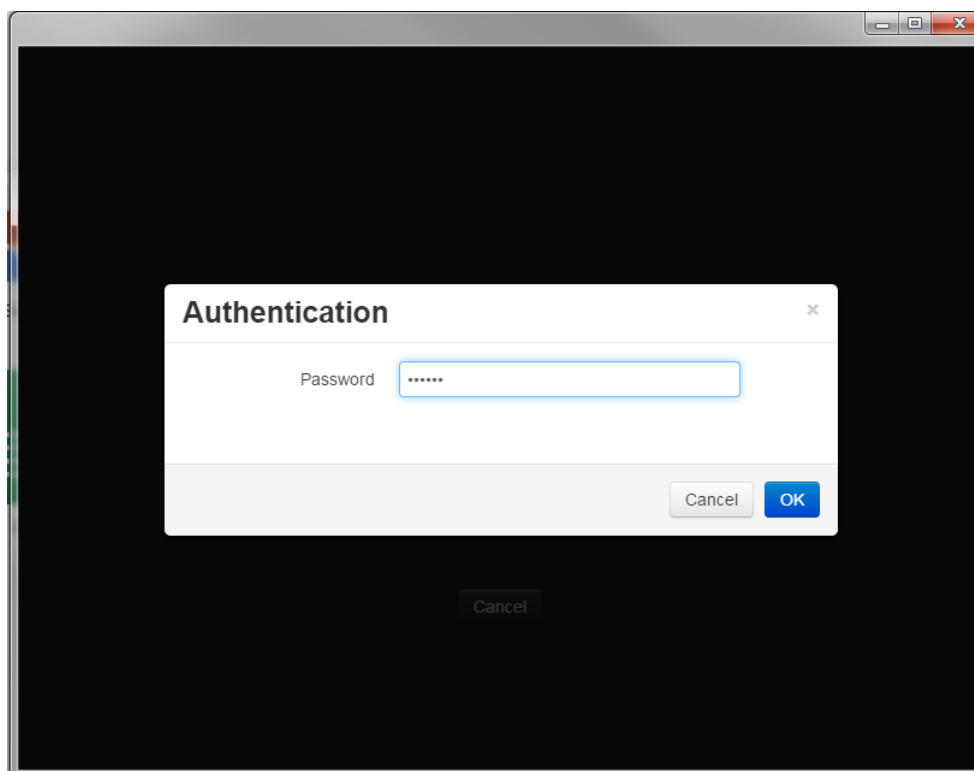


Figure 107 : Il faut s'authentifier avec le mot de passe défini auparavant.

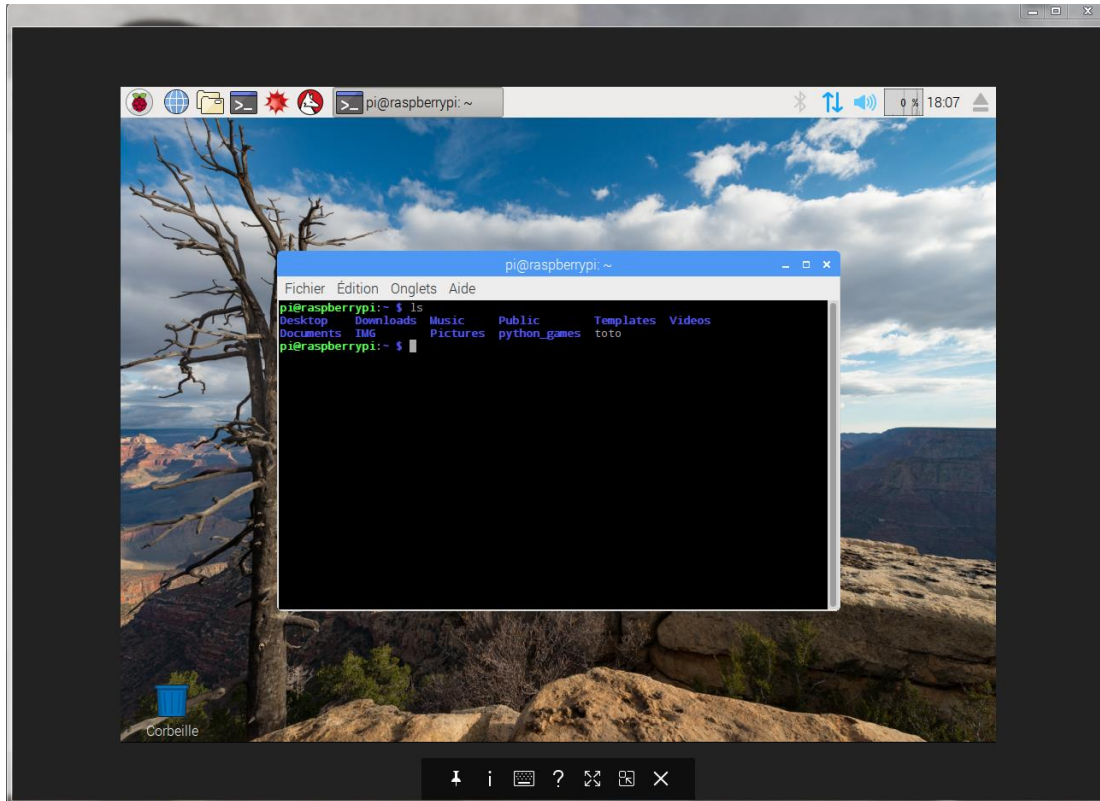


Figure 108 : On a bien accès à l'interface graphique, qui est un peu lente mais fonctionnelle.

Le Raspberry Pi est maintenant correctement installé/configuré avec Raspbian, on peut désormais commencer à travailler dessus.

Annexe 4 : Manipulations pour installer les différents paquets et applications.

Installation et configuration de Airplay/Bluetooth.

Mettre à jour la liste des fichiers disponibles.

```
sudo apt-get update
```

Mettre à jour tous les paquets installés en installant de nouveaux paquets nécessaires.

```
sudo apt-get dist-upgrade  
sudo reboot
```

Installer PulseAudio et BlueZ.

```
sudo apt-get install bluez pulseaudio pulseaudio-module-bluetooth
```

Installer dbus pour Python.

```
sudo apt-get install python-dbus
```

Installer espeak

```
sudo apt-get install -qy espeak
```

Créer les utilisateurs et les privileges necessaire pour permettre l'interaction entre Bluetooth et PusleAudio.

```
sudo addgroup --system pulse  
sudo adduser --system --ingroup pulse --home /var/run/pulse pulse  
sudo addgroup --system pulse-access  
sudo adduser pulse audio  
sudo adduser root pulse-access  
sudo adduser pulse lp  
sudo reboot
```

Ajout du nouveau fichier /etc/init.d/pulseaudio, le rendre exécutable et le lancer au démarrage

```
sudo cp init.d/pulseaudio /etc/init.d
sudo chmod +x /etc/init.d/pulseaudio
sudo update-rc.d pulseaudio defaults
```

Contenu du nouveau fichier /etc/init.d/pulseaudio

```
#!/bin/sh -e
```

```
### BEGIN INIT INFO
```

```
# Provides:          pulseaudio esound
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Should-Start:      udev network-manager
# Should-Stop:       udev network-manager
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start the PulseAudio sound server
# Description:       System mode startup script for
#                    the PulseAudio sound server.
### END INIT INFO
```

```
DAEMON=/usr/bin/pulseaudio
PIDDIR=/var/run/pulse
PIDFILE=${PIDDIR}/pid
DAEMONUSER=pulse
PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

```
test -x $DAEMON || exit 0
```

```
. /lib/lsb/init-functions
```

```
pulseaudio_start () {
    log_daemon_msg "Starting system PulseAudio Daemon"
    if [ ! -d $PIDDIR ]; then
        mkdir -p $PIDDIR
        chown $DAEMONUSER:$DAEMONUSER $PIDDIR
    fi
    start-stop-daemon -x $DAEMON -p $PIDFILE --start -- --system --disallow-exit --
disallow-module-loading=0 --daemonize --log-target=syslog --high-priority
    status=$?
    if [ -e /var/run/pulse/.esd_auth ]; then
        chown pulse:pulse-access /var/run/pulse/.esd_auth
        chmod 640 /var/run/pulse/.esd_auth
    fi
    if [ -e /var/run/pulse/.pulse-cookie ]; then
        chown pulse:pulse-access /var/run/pulse/.pulse-cookie
        chmod 640 /var/run/pulse/.pulse-cookie
    fi
    log_end_msg ${status}
}

pulseaudio_stop () {
    log_daemon_msg "Stopping system PulseAudio Daemon"
    start-stop-daemon -p $PIDFILE --stop --retry 5 || echo -n "...which is not running"
    log_end_msg $?
}
```

```
case "$1" in
    start|stop)
        pulseaudio_${1}
        ;;
    restart|reload|force-reload)
        if [ -s $PIDFILE ] && kill -0 $(cat $PIDFILE) >/dev/null 2>&1; then
            pulseaudio_stop
            pulseaudio_start
        fi
        ;;
    force-stop)
        pulseaudio_stop
        killall pulseaudio || true
        sleep 2
        killall -9 pulseaudio || true
        ;;
    status)
        status_of_proc -p $PIDFILE "$DAEMON" "system-wide PulseAudio" && exit 0 ||
exit $?
        ;;
*)
    echo "Usage: /etc/init.d/pulseaudio {start|stop|force-
stop|restart|reload|force-reload|status}"
    exit 1
    ;;
esac
exit 0
```

Ajout du nouveau fichier /etc/init.d/bluetooth, le rendre exécutable et le lancer au démarrage

```
sudo cp init.d/bluetooth /etc/init.d
sudo chmod +x /etc/init.d/bluetooth
sudo update-rc.d bluetooth defaults
```

Contenu du nouveau fichier /etc/init.d/bluetooth

```
#!/bin/sh -e

### BEGIN INIT INFO
# Provides:          bluetooth
# Required-Start:    $local_fs $syslog dbus
# Required-Stop:     $local_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Starts bluetooth daemons
### END INIT INFO

. /lib/lsb/init-functions

DESC=bluetoothd
DAEMON=/usr/libexec/bluetooth/bluetoothd
SSD_OPTIONS="--oknodo --quiet --exec $DAEMON --plugin=a2dp"
#SSD_OPTIONS="--oknodo --quiet --exec $DAEMON" #Change to this if you want media control
using DBus at the expense of volume control
HCI=hci0

case "${1}" in
```

```
start)
    log_daemon_msg "Starting Bluetooth daemon bluetoothd..."
    start-stop-daemon --start --background $SSD_OPTIONS
    log_progress_msg "${DAEMON}"

    hciconfig $HCI up > /dev/null 2>&1
    log_end_msg 0
    ;;

stop)
    log_daemon_msg "Stopping Bluetooth daemon bluetoothd..."
    start-stop-daemon --stop $SSD_OPTIONS
    log_progress_msg "${DAEMON}"
    log_end_msg 0
    ;;

restart)
    ${0} stop
    sleep 1
    ${0} start
    ;;

status)
    status_of_proc "$DAEMON" "$DESC" && exit 0 || exit $?
    ;;

*)
    echo "Usage: ${0} {start|stop|restart|status}"
    exit 1
    ;;
esac

exit 0
```

Ajout du nouveau fichier /etc/init.d/bluetooth-agent, le rendre exécutable et le lancer au démarrage

```
sudo cp init.d/bluetooth-agent /etc/init.d
sudo chmod +x /etc/init.d/bluetooth-agent
sudo update-rc.d bluetooth-agent defaults
```

Contenu du nouveau fichier /etc/init.d/bluetooth-agent

```
#!/bin/sh -e

### BEGIN INIT INFO
# Provides: bluetooth-agent
# Required-Start: $remote_fs $sysLog bluetooth pulseaudio
# Required-Stop: $remote_fs $sysLog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Makes Bluetooth discoverable and connectable to 0000
# Description: Start Bluetooth-Agent at boot time.
### END INIT INFO
#!/bin/sh
# /etc/init.d/bluetooth-agent
USER=root
HOME=/root
export USER HOME
```

```
case "$1" in
  start)
    echo "setting bluetooth discoverable"
    sleep 1
    sudo hciconfig hci0 piscan
    start-stop-daemon -S -x /usr/local/bin/simple-agent.autotrust &
    echo "bluetooth-agent started"
    ;;
  stop)
    echo "Stopping bluetooth-agent"
    start-stop-daemon -K -x /usr/local/bin/simple-agent.autotrust
    ;;
  *)
    echo "Usage: /etc/init.d/bluetooth-agent {start|stop}"
    exit 1
    ;;
esac
exit 0
```

Ajout du nouveau fichier /usr/local/bin/bluez-udev et lui donner les bonnes propriétés.

```
sudo cp usr_local_bin/bluez-udev /usr/local/bin
sudo chmod 755 /usr/local/bin/bluez-udev
```

Contenu du nouveau fichier /usr/local/bin/bluez-udev

```
#!/bin/bash
audio_sink=0
name=$(sed 's/\\/"/g' <<< $NAME)
#exit if not a BT address
if [[ ! $name =~ ^([0-9A-F]{2}[:-]){5}([0-9A-F]{2})$ ]]; then exit 0; fi

bt_name=`grep Name /var/lib/bluetooth/*/${name}/info | awk -F=' ' '{print $2}`

audio_source=bluez_source.${sed 's/:/_/g' <<< $name)

action=$(expr "$ACTION" : "\([a-zA-Z]\+\).*")
logger "Action: $action"
if [ "$action" = "add" ]; then
  logger "[$(basename $0)] Bluetooth device is being added [$name] - $bt_name"
  logger "[$(basename $0)] Patching $audio_source into ALSA sink #$audio_sink"
  #hciconfig hci0 noscan
  bluetoothctl << EOT
discoverable off
EOT
  amixer cset numid=3 1
  amixer cset numid=3 90%
  #espeak -s 160 -k 1 "Device, $bt_name Connected"
  /usr/local/bin/say.sh "Device, $bt_name Connected"
  amixer cset numid=3 100%
  sleep 1
  pactl set-sink-volume 0 65537
  pactl set-source-volume $audio_source 64500

  # Loop back this source to the default sink
  handle=$(pactl load-module module-loopback source=$audio_source sink=$audio_sink)
  logger "[$(basename $0)] PulseAudio module-loopback returned handle [$handle]"
  logger "$bt_name"
fi
```

```
if [ "$action" = "remove" ]; then
    logger ["$(basename $0)] Bluetooth device is being removed [$name] - $bt_name"
    #hciconfig hci0 pscan

    bluetoothctl << EOT
discoverable on
EOT

    # remove any loopback modules assigned to this source
    # only required for USB sound cards, which PulseAudio will not automatically remove
    for handle in $(pactl list short modules | grep module-loopback | grep
source=$audio_source | cut -f 1); do
        logger ["$(basename $0)] Unloading module-loopback with handle [$handle]"
        pactl unload-module $handle
    done
    sleep 5
    amixer cset numid=3 90%
    #espeak -s 160 -k 1 "Device, $bt_name Disconnected"
    /usr/local/bin/say.sh "Device, $bt_name Disconnected"
    amixer cset numid=3 100%
fi
```

Ajout du fichier /usr/local/bin/simple-agent.autotrust qui va permettre de faire confiance à n'importe quel appareil Bluetooth qui va vouloir se connecter.

```
sudo cp usr_local_bin/simple-agent.autotrust /usr/local/bin
sudo chmod 755 /usr/local/bin/simple-agent.autotrust
```

Installation de Shairport-sync

```
sudo apt-get -y install build-essential git autoconf automake
libtool libdaemon-dev libasound2-dev libpopt-dev libconfig-dev
avahi-daemon libavahi-client-dev libssl-dev
mkdir airplay
cd airplay
git clone https://github.com/mikebrady/shairport-sync.git
cd shairport-sync
autoreconf -i -f
./configure --with-alsa --with-avahi --with-ssl=openssl --with-
metadata --with-systemd --sysconfdir=/etc
make
getent group shairport-sync &>/dev/null || sudo groupadd -r
shairport-sync >/dev/null
getent passwd shairport-sync &> /dev/null || sudo useradd -r -M -g
shairport-sync -s /usr/bin/nologin -G audio shairport-sync
>/dev/null
sudo make install
sudo systemctl enable shairport-sync
```



```
sudo useradd -g audio shairport-sync  
sudo systemctl daemon-reload  
sudo service shairport-sync start
```

Installation et configuration de hostapd et dnsmasq.

Dans le cadre de notre projet on veut que notre Raspberry soit autonome, c'est-à-dire que nous n'avons pas besoin d'un Wi-Fi domestique pour faire fonctionner une application comme Airplay. On va donc configurer le Raspberry Pi en Wi-Fi access point.

Tout d'abord on va installer le driver nécessaire pour utilisation poussée du protocole 802.11 :

wget <https://dl.dropboxusercontent.com/u/80256631/install-wifi.tar.gz>

tar xzf install-wifi.tar.gz

./install-wifi

On a rapidement regardé le script install-wifi, il permet d'installer la bonne version en fonction de notre kernel et de notre dongle.

```
root@raspberrypi:/home/pi# ./install-wifi  
Your current kernel revision = 4.4.26+  
Your current kernel build    = #915  
  
Checking for a wifi module to determine the driver to install.  
  
Your wifi module is Bus 001 Device 004: ID 0bda:8179 Realtek Semiconductor Corp.  
  
And it uses the 8188eu driver.  
  
Checking for a new 8188eu wifi driver module for your current kernel.  
There is a driver module available for this kernel revision.  
Downloading the 8188eu driver.  
Installing the 8188eu driver.  
  
A version of the driver 8188eu.ko is already loaded and running. You will need to reboot to load the new driver.
```

Figure 109 : Installation du driver nécessaire au fonctionnement du Wi-Fi.

On effectue un reboot.

Au niveau logiciel nous avons besoin de plusieurs paquets :

- **hostapd** : va nous permettre de mettre en place un réseau Wi-Fi avec toutes les sécurité nécessaires
- **dnsmasq** : va permettre au Raspberry Pi de jouer le rôle de DHCP, pour attribuer des adresses IP dynamiquement aux périphériques qui se connecteront, c'est pour simplifier au maximum l'utilisation pour le client.

On va commencer par mettre à jour notre distribution avec :

sudo apt-get update

Puis on va installer les paquets cités ci-dessus :

sudo apt-install hostapd dnsmasq

On va configurer notre interface wlan0 pour que celle-ci soit statique

nano /etc/dhcpd.conf

On ajoute la ligne ***denyinterfaces wlan0***

```
GNU nano 2.2.6 Fichier : /etc/dhcpd.conf
# Safe to enable by default because it requires the equivalent option set
# on the server to actually work.
option rapid_commit

# A list of options to request from the DHCP server.
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
# Most distributions have NTP support.
option ntp_servers
# Respect the network MTU.
# Some interface drivers reset when changing the MTU so disabled by default.
#option interface_mtu

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

# A hook script is provided to lookup the hostname if not set by the DHCP
# server, but it should not be run by default.
nohook lookup-hostname

denyinterfaces wlan0
█
```

Figure 110 : Modification du fichier /etc/dhcpd.conf.

On va ensuite modifier le fichier **/etc/network/interfaces** et éditer ce qui concerne wlan0 de la manière suivante (on définit statiquement) :

```
GNU nano 2.2.6 Fichier : /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
allow-hotplug eth0
iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
#wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

auto wlan1
allow-hotplug wlan1
iface wlan1 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Figure 111 : Définition statique du réseau sur le fichier /etc/network/interfaces.

On redémarre notre service dhcp, on redémarre notre interface wlan0 et on vérifie que notre configuration a bien été mise en place :

/etc/init.d/dhcpdcd restart

ifdown wlan0

ifup wlan0

ifconfig

```
wlan0    link encap:Ethernet  HWaddr 18:a6:f7:1a:2c:53
         inet adr:192.168.1.1  Bcast:192.168.1.255  Masque:255.255.255.0
         UP BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:3571 errors:0 dropped:1949 overruns:0 frame:0
         TX packets:173 errors:0 dropped:1 overruns:0 carrier:0
         collisions:0 lg file transmission:1000
         RX bytes:480001 (468.7 KiB)  TX bytes:23718 (23.1 KiB)
```

Figure 112 : Nos modifications ont bien eu un impact car nous avons l'adresse qui avait été définie.

On peut maintenant passer à la configuration de notre hostapd :

On commence par créer un fichier **/etc/hostapd/hostapd.conf**

```
GNU nano 2.2.6 Fichier : /etc/hostapd/hostapd.conf
#nom de l'interface Wi-Fi que l'on configure
interface=wlan0

#Utilisation du driver nl80211 avec brcmfmac
driver=nl80211

#Nom du réseau
ssid=Raspberry-AP-HiFi

#Utilisation de la bande 2,4Ghz
hw_mode=g

#Utilisation du canal
channel=5

#Autorisation du 802.11n (norme par codage OFDM)
ieee80211n=1

#Autorisation du WMM (Wi-Fi Multimedia) basé sur 802.11e, cela fournit une QoS
wmm_enabled=1

#Autoriser les canaux de 40MHz avec 20ns d'intervalle de garde
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]

#Accepter toutes les adresses MAC
macaddr_acl=0

#Utilise l'authentification WPA
auth_algs=1

#Les clients ont besoin de savoir le nom du réseau
ignore_broadcast_ssid=0

#Utilise le WPA2
wpa=2

#Utilise une PSK (Pre-shared Key)
wpa_key_mgmt=WPA-PSK

#La clé PSK
wpa_passphrase=azerty123

#Utilisation de AES au lieu de TKIP
rsn_pairwise=CCMP
```

Figure 113 : Configuration de notre réseau Wi-Fi dans le fichier /etc/hostapd/hostapd.conf. Tout est commenté pour comprendre chaque ligne ajoutée.

Pour tester si cela fonctionne même si on n'a pas configuré de DHCP pour l'instant on fait la commande :

/usr/sbin/hostapd /etc/hostapd/hostapd.conf

```
root@raspberrypi:/home/pi# /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
Using interface wlan0 with hwaddr 18:a6:f7:1a:2c:53 and ssid "Raspberry-AP-HiFi"
random: Only 15/20 bytes of strong random data available from /dev/random
random: Not enough entropy pool available for secure operations
WPA: Not enough entropy in random pool for secure operations - update keys later when the first station connects
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA 98:fe:94:30:5a:43 IEEE 802.11: associated
wlan0: AP-STA-CONNECTED 98:fe:94:30:5a:43
wlan0: STA 98:fe:94:30:5a:43 RADIUS: starting accounting session 58304E4C-00000000
wlan0: STA 98:fe:94:30:5a:43 WPA: pairwise key handshake completed (RSN)
wlan0: STA 98:fe:94:30:5a:43 IEEE 802.11: disassociated
wlan0: AP-STA-DISCONNECTED 98:fe:94:30:5a:43
```

Figure 114 : Test pour montrer que le réseau est bien accessible.

Avec notre iPhone on essaye de se connecter au Wi-Fi celui-ci apparaît bien, on peut se connecter dessus mais on n'a pas d'adresse IP il faut donc configurer le DHCP maintenant :

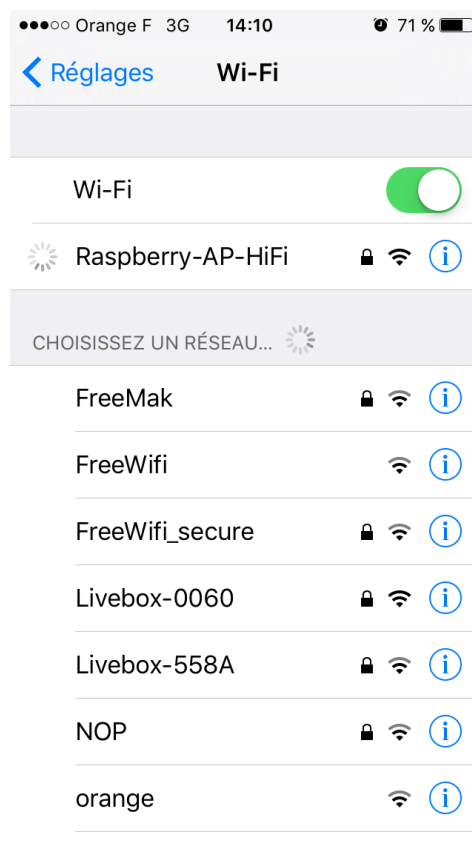


Figure 115 : Le réseau Raspberry-AP-HiFi apparaît bien dans la liste des réseaux Wi-Fi.

Tout d'abord on va indiquer à notre hostapd ou regarder pour les fichiers de configurations quand il démarre, pour cela on va ouvrir le fichier **/etc/default/hostapd** et trouver la ligne

#DAEMON_CONF= '' et la remplacer par :

DAEMON_CONF= '/etc/hostapd/hostapd.conf '

```
GNU nano 2.2.6 Fichier : /etc/default/hostapd
# Defaults for hostapd initscript
#
# See /usr/share/doc/hostapd/README.Debian for information about alternative
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
#
#DAEMON_CONF="/etc/hostapd/hostapd.conf"
#
# Additional daemon options to be appended to hostapd command:-
#   -d show more debug messages (-dd for even more)
#   -K include key data in debug messages
#   -t include timestamps in some debug messages
#
# Note that -B (daemon mode) and -P (pidfile) options are automatically
# configured by the init.d script and must not be added to DAEMON_OPTS.
#
#DAEMON_OPTS=""
```

Figure 116 : Modification du fichier /etc/default/hostapd pour montrer où se trouvent les fichiers de configurations.

On va créer un nouveau fichier **/etc/dnsmasq.conf** avec tout ce que l'on a besoin :

```
GNU nano 2.2.6 Fichier : /etc/dnsmasq.conf
#On indique l'interface que l'on va utiliser
interface=wlan0
#On spécifie l'adresse sur laquelle on doit écouter (l'adresse de l'AP)
listen-address=192.168.1.1
#On lie les interfaces pour s'assurer que l'on envoie pas ailleurs
bind-interfaces
#Ne pas rediriger les adresses dans un endroit où les adresses ne sont pas routables
bogus-priv
#Plage d'adresse (on considère que au maximum 3 utilisateurs pourront se connecter avec un bail de connexion d'une heure)
dhcp-range=192.168.1.10,192.168.1.13,1h
```

Figure 117 : Mise en place du fichier /etc/dnsmasq.conf pour attribuer des adresses dynamiquement.

On fait un reboot puis on va créer un script qui va démarrer nos services nécessaires lors du boot :

nano /etc/init.d/airplay-wifiAP

```
# !/bin/sh

# coding : utf-8

service start hostapd

service start dnsmasq

/usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

On fait la commande pour ajouter ce script au démarrage :

update-rc.d airplay-wifiAP defaults

On reboot, sans connexion SSH, sans câble Ethernet et le Wi-Fi est bien actif, notre Raspberry est devenu autonome.

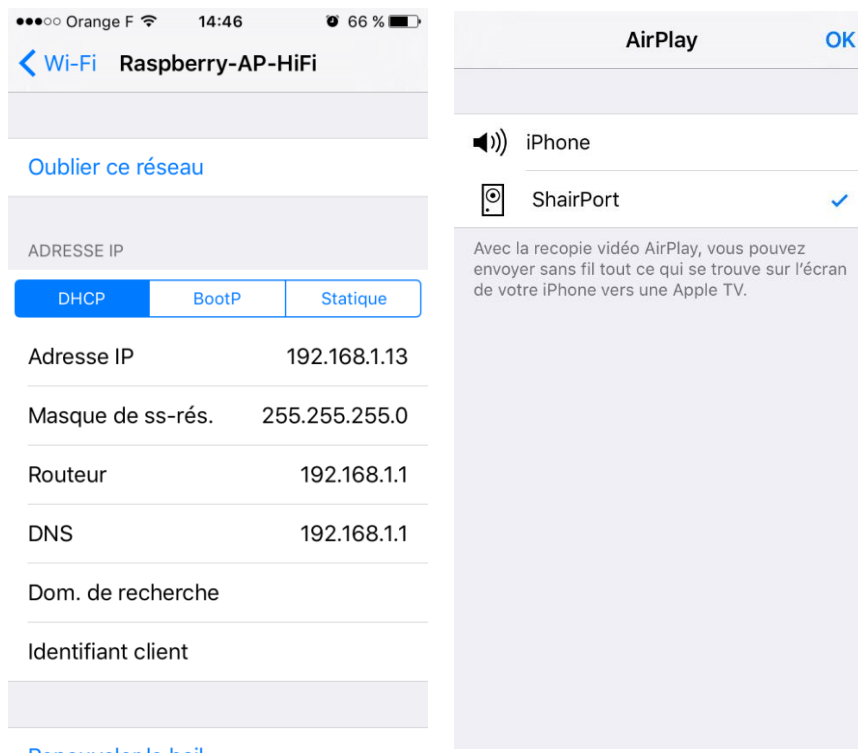


Figure 118 : Le DHCP a bien donné une adresse à notre iPhone et on peut se connecter avec l'application Airplay.

Annexe 5 : Etude détaillée des protocoles Airplay et Bluetooth

Airplay

Airplay est une famille de protocole implémenté par Apple pour diffuser différents types de contenu multimédia sur une Apple TV depuis n'importe quel appareil iOS (iPhone, iPod, iPad). Ainsi on peut montrer des photos, écouter de la musique et visionner de la vidéo depuis un iOS. Nous allons essayer de décrire ce protocole qui est basé sur des protocoles bien connu dans le réseau comme DNS, HTTP, RTSP, RTP ou NTP. Les informations données ont été récoltées par des techniques de reverse engineering il est donc possible qu'elles soient incomplètes ou inexactes.

Protocole RTSP

Le streaming audio est rendu possible grâce au protocole RTSP (Real Time Streaming Protocol, RFC 2326). Nous allons voir plus en détail les requêtes RTSP.

La requête « OPTIONS » demande au serveur RTSP les méthodes supportées, on peut compter 11 méthodes qui sont : « ANNOUNCE », « SETUP », « PAUSE », « FLUSH », « TEARDOWN », « OPTIONS », « GET_PARAMETER », « SET_PARAMETER », « POST », « GET ».

```
CLIENT → SERVER
OPTIONS * RTSP/1.0
CSeq: 3
User-Agent: iTunes/10.6 (Macintosh; Intel Mac OS X 10.7.3) AppleWebKit/535.18.5
Client-Instance: 56B29BB6CB904862
DACP-ID: 56B29BB6CB904862
Active-Remote: 1986535575
```

Figure 119 : Demande des méthodes disponibles pour RTSP.

```
SERVER → CLIENT
RTSP/1.0 200 OK
Public: ANNOUNCE, SETUP, RECORD, PAUSE, FLUSH, TEARDOWN, OPTIONS,
GET_PARAMETER, SET_PARAMETER, POST, GET
Server: AirTunes/130.14
CSeq: 3
```

Figure 120 : Ensemble des méthodes disponibles pour RTSP.

La requête « ANNOUNCE » permet d'interroger le serveur RTSP sur les propriétés de streaming en utilisant SDP (Session Description Protocol, RFC 4566), notamment les codecs et les clés de cryptage.

```

CLIENT → SERVER
ANNOUNCE rtsp://192.168.1.45/2699324803567405959 RTSP/1.0
X-Apple-Device-ID: 0xa4d1d2800b68
CSeq: 16
DACP-ID: 14413BE4996FEA4D
Active-Remote: 2543110914
Content-Type: application/sdp
Content-Length: 331

v=0
o=AirTunes 2699324803567405959 0 IN IP4 192.168.1.5
s=AirTunes
c=IN IP4 192.168.1.5
t=0 0
m=audio 0 RTP/AVP 96
a=rtpmap:96 mpeg4-generic/44100/2
a=fmtp:96
a=fpaesKey:R1BMWQECaQAAAAA8AAAAA0G6c4aMdLKXAX+1b7p/EhgAAAAQeX5uqGyYk8mJX+gd5ANER+am18urqFmVc
a=esiv:VZTaHn4wSJ84Jjz1b94m0Q==
a=min-latency:11025

SERVER → CLIENT
RTSP/1.0 200 OK
Server: AirTunes/130.14
CSeq: 16
    
```

Figure 121 : Interaction pour la méthode Announce.

La requête « SETUP » permet d'initialiser une session d'enregistrement pour le streaming, cette requête envoie toutes les informations nécessaires, de plus trois canaux UDP sont mis en place.

CHANNEL	DESCRIPTION
server	audio data
control	sync and retransmit requests
timing	master clock sync

Figure 122 : Liste des canaux avec leurs descriptions.

```
CLIENT → SERVER
SETUP rtsp://fe80::217:f2ff:fe0f:e0f6/3413821438 RTSP/1.0
CSeq: 4
Transport: RTP/AVP/UDP;unicast;interleaved=0-1;mode=record;control_port=6001;timing_port=6002
User-Agent: iTunes/10.6 (Macintosh; Intel Mac OS X 10.7.3) AppleWebKit/535.18.5
Client-Instance: 56B29BB6CB904862
DACP-ID: 56B29BB6CB904862
Active-Remote: 1986535575

SERVER → CLIENT
RTSP/1.0 200 OK
Transport: RTP/AVP/UDP;unicast;mode=record;server_port=53561;control_port=63379;timing_port=50607
Session: 1
Audio-Jack-Status: connected
Server: AirTunes/130.14
CSeq: 4
```

Figure 123 : Interactions pour la méthode Setup.

La requête « FLUSH » permet d'arrêter le streaming audio (pour mettre en pause par exemple).

```
CLIENT → SERVER
FLUSH rtsp://fe80::217:f2ff:fe0f:e0f6/3413821438 RTSP/1.0
CSeq: 31
Session: 1
RTP-Info: seq=25009;rtptime=1148010660
User-Agent: iTunes/10.6 (Macintosh; Intel Mac OS X 10.7.3) AppleWebKit/535.18.5
Client-Instance: 56B29BB6CB904862
DACP-ID: 56B29BB6CB904862
Active-Remote: 1986535575

SERVER → CLIENT
RTSP/1.0 200 OK
RTP-Info: rtptime=1147914212
Server: AirTunes/130.14
CSeq: 31
```

Figure 124 : Interactions pour la méthode Flush.

La requête « TEARDOWN » permet de fermer la session RTSP.

```
CLIENT → SERVER
TEARDOWN rtsp://fe80::217:f2ff:fe0f:e0f6/3413821438 RTSP/1.0
CSeq: 32
Session: 1
User-Agent: iTunes/10.6 (Macintosh; Intel Mac OS X 10.7.3) AppleWebKit/535.18.5
Client-Instance: 56B29BB6CB904862
DACP-ID: 56B29BB6CB904862
Active-Remote: 1986535575

SERVER → CLIENT
RTSP/1.0 200 OK
Server: AirTunes/130.14
CSeq: 32
```

Figure 125 : Interactions pour la méthode Teardown.

Protocole RTP

Le protocole RTSP ne transporte pas les données elles-mêmes et doit être associé à un protocole de transport comme RTP pour cette tâche, c'est pour cette raison que nous allons nous intéresser au protocole RTP (Real-time Transport Protocol).

Il existe différents types de contenu :

PAYLOAD TYPE	PORT	DESCRIPTION
82	timing_port	timing request
83	timing_port	timing reply
84	control_port	time sync
85	control_port	retransmit request
86	control_port	retransmit reply
96	server_port	audio data

Figure 126 : Liste des différents types de contenus avec leurs descriptions.

Les données audio sont envoyées en utilisant le Payload 96 (DynamicRTP-Type-96), le drapeau Marker est mis sur le premier paquet envoyé après une requête « RECORD » ou « FLUSH ». Les données peuvent être cryptées.

```
CLIENT → SERVER
0000  80 e0 b1 91 f7 79 16 c2 e8 bb 6b 2c bb 5c 8e 51
0010  aa 7c d2 96 00 c3 fd 60 eb ae 6e 41 31 38 fe ae
....
03e0  cb 1c 73 bf e7 05 93 30 fa 85 7f 32 77 8d a8 97
03f0  a0 c7 c8 78 7b e5 81 a1 4f b4 3e a3 43 db 7c

Real-Time Transport Protocol
 10.. .... = Version: RFC 1889 Version (2)
 ..0. .... = Padding: False
 ...0 .... = Extension: False
 .... 0000 = Contributing source identifiers count: 0
 1... .... = Marker: True
 Payload type: DynamicRTP-Type-96 (96)
Sequence number: 45457
Timestamp: 4151908034
Synchronization Source identifier: 0xe8bb6b2c (3904596780)
Payload: bb5c8e51aa7cd29600c3fd60ebae6e413138feae909b44f1...
```

Figure 127 : Payload 96, pour les données audio.

Les paquets Sync sont envoyés à une fréquence de 1 par seconde au port de contrôle. Ils sont utilisés pour synchroniser l'horloge RTP à celle du protocole NTP (Network Time Protocol). Pour cela le Payload 84 est utilisé. Le SSRC (Synchronisation source identifier) identifie de façon unique la source du stream.

BYTES	DESCRIPTION
8	RTP header without SSRC
8	current NTP time
4	RTP timestamp for the next audio packet

Figure 128 : Liste des octets pour la synchronisation avec leurs descriptions.

```

CLIENT → SERVER
0000  80 d4 00 04 c7 cd 11 a8 83 ab 1c 49 2f e4 22 e2
0010  c7 ce 3f 1f

Real-Time Transport Protocol
 10.. .... = Version: RFC 1889 Version (2)
 ..0. .... = Padding: False
 ...0 .... = Extension: False
 ... 0000 = Contributing source identifiers count: 0
 1... .... = Marker: True
 Payload type: Unassigned (84)
Sequence number: 4
Timestamp: 3352105384
Synchronization Source identifier: 0x83ab1c49 (2209029193)
Payload: 2fe422e2c7ce3f1f
    
```

Figure 129 : Payload 84, synchronisation temporelle.

Le volume audio peut être changé grâce à la méthode « SET_PARAMETER ». Le volume est une valeur float qui représente l'atténuation en dB, par exemple une valeur de -144 signifie qu'il n'y aucune sortie sonore.

```

CLIENT → SERVER
SET PARAMETER rtsp://fe80::217:f2ff:fe0f:e0f6/3413821438 RTSP/1.0
CSeq: 6
Session: 1
Content-Type: text/parameters
Content-Length: 20
User-Agent: iTunes/10.6 (Macintosh; Intel Mac OS X 10.7.3) AppleWebKit/535.18.5
Client-Instance: 56B29BB6CB904862
DACP-ID: 56B29BB6CB904862
Active-Remote: 1986535575
volume: -11.123877

SERVER → CLIENT
RTSP/1.0 200 OK
Server: AirTunes/130.14
CSeq: 6
    
```

Figure 130 : Interactions permettant de changer le volume grâce à la méthode set_parameter.

Bluetooth

Les caractéristiques techniques du Bluetooth sont supervisées par le Bluetooth Special Interest Group (SIG). Les deux implémentations les plus courantes sont : Bluetooth BR/EDR (Basic Rate / Enhanced Data Rate) qui a été adopté dans les versions 2.0/2.1 puis le Bluetooth LE (Low Energy) qui a été adopté dans les versions 4.0/4.1/4.2/5.0. Il existe un Dual-mode qui permet d'utiliser les deux technologies énoncées.

Bluetooth BR/EDR	Etablit une connexion sans-fil, continue et relativement de courte portée ce qui est idéal pour le streaming audio.
Bluetooth LE	Permet des bursts de données pour une connexion radio longue portée ce qui est idéal pour les IoT (Internet of Things) qui n'utilise pas de connexion continue et qui ont besoin d'une autonomie de batterie longue.
Dual-mode	Les smartphones et tablettes d'aujourd'hui ont souvent besoin des deux implémentations. Par exemple BR/EDR pour connecter un casque audio ou LE pour relier à des antennes.

Figure 131 : Tableau illustrant les différents types de Bluetooth et leurs utilisations.

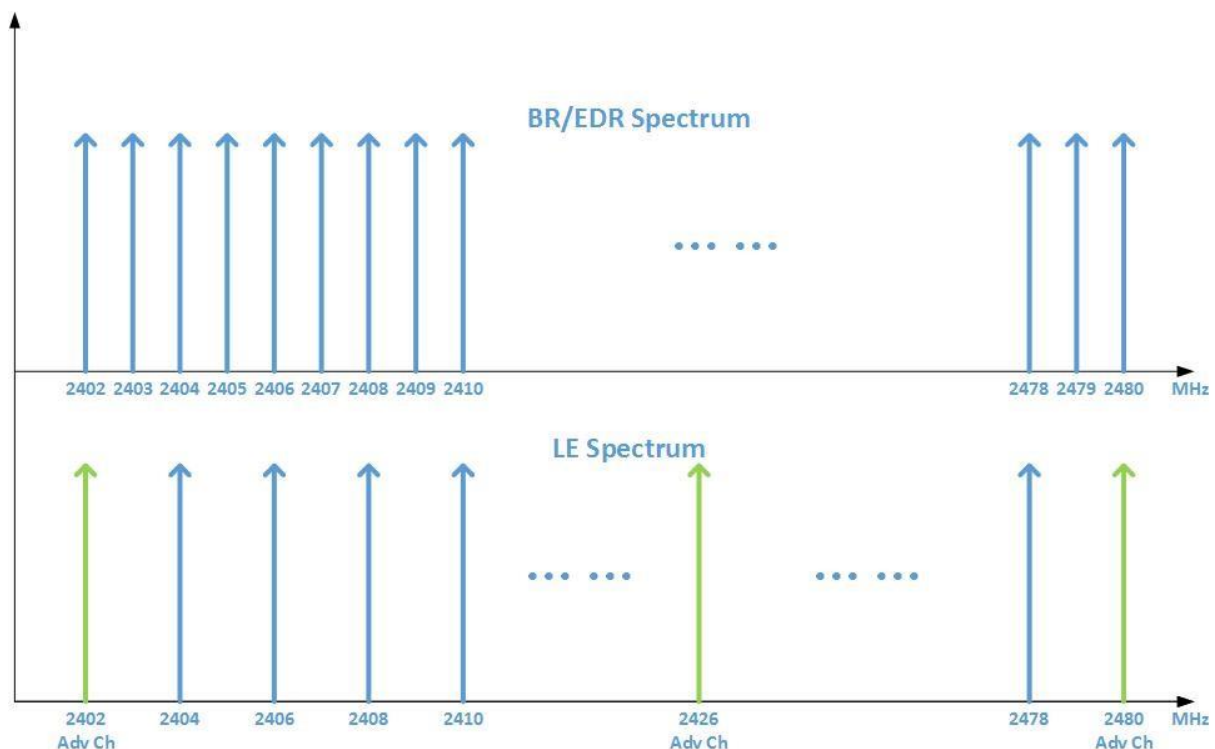


Figure 132 : Différence de canaux spectral entre les 2 types de bluetooth.

Pour assurer l'interopérabilité entre les systèmes, il a fallu définir un protocole d'échange de messages entre des couches de niveau équivalents.

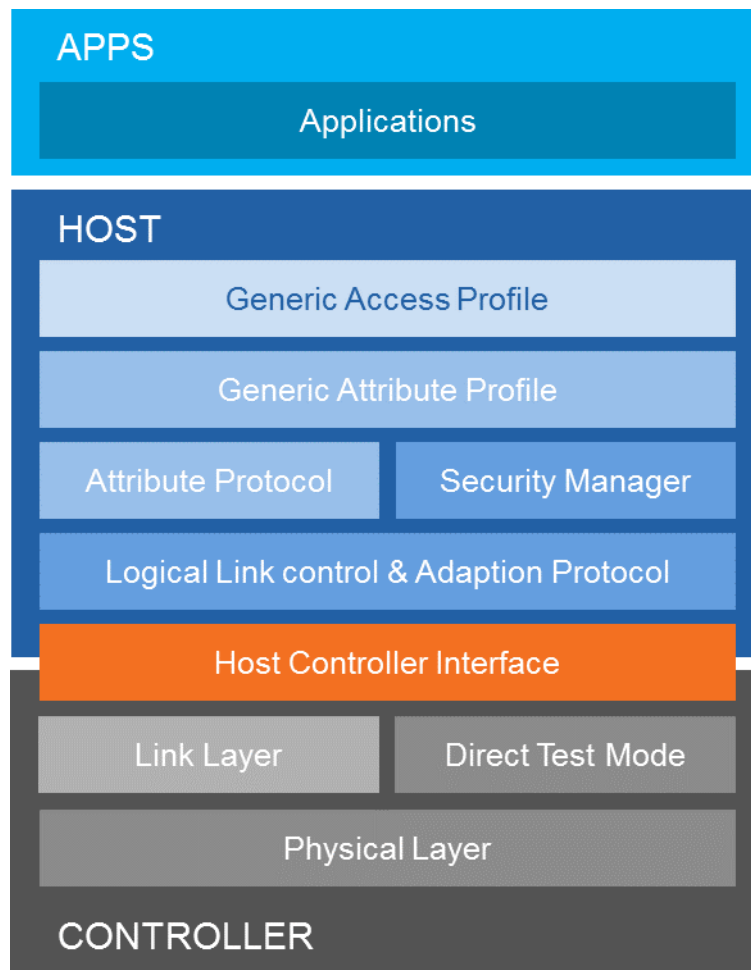


Figure 133 : Couches protocolaire montrant comment fonctionne le protocole Bluetooth.

Physical Layer	Permet de contrôler la transmission et la réception des ondes 2,4GHz avec les canaux Bluetooth. Le Bluetooth BR/EDR fournit plus de canaux étroits alors que LE fournit moins de canaux plus larges.
Link Layer	Définit les paquets concernant la trame, les canaux, la procédure de découverte et de connexion, l'envoi et la réception de données.
Direct Test Mode	Permet de tester la couche physique en transmettant ou recevant une séquence de paquets.

Host to Controller Interface (HCI)	Standard optionnel entre le contrôleur Bluetooth et l'hôte Bluetooth.
Couche L2CAP	Protocole basé sur paquet qui transmet les paquets au HCI. Ce protocole possède la capacité de fragmenter, assembler les paquets et de renseigner sur la qualité de service aux couches supérieures.
Attribute Protocol (ATT)	Définit un protocole client/serveur pour l'échange de données seulement quand une connexion a été établie. Ce protocole est surtout utilisé dans le Bluetooth LE.
Security Manager	Un autre protocole qui gère l'intégrité de l'appairage, l'authentification et le cryptage entre les appareils Bluetooth. Cette couche fournit aussi des outils de sécurité utiles pour des applications de plus haut niveaux.
Generic Attribute Profile (GATT)	Ce service utilise le protocole ATT et encapsule le comportement d'un appareil. Uniquement utilisé dans le Bluetooth LE.
Generic Access Profile (GAP)	Travaille de pair avec GATT dans le Bluetooth LE pour définir les procédures et les rôles dans la découverte d'appareil Bluetooth et le partage d'informations.

Figure 134 : Tableau avec la liste des différentes couches et leurs utilités.

L'opération de la couche liaison peut être décrit en terme d'état de la machine avec 5 états différents : Attente, Annonce, Scan, Initiation, Connexion. Un seul état ne peut être actif à la fois.

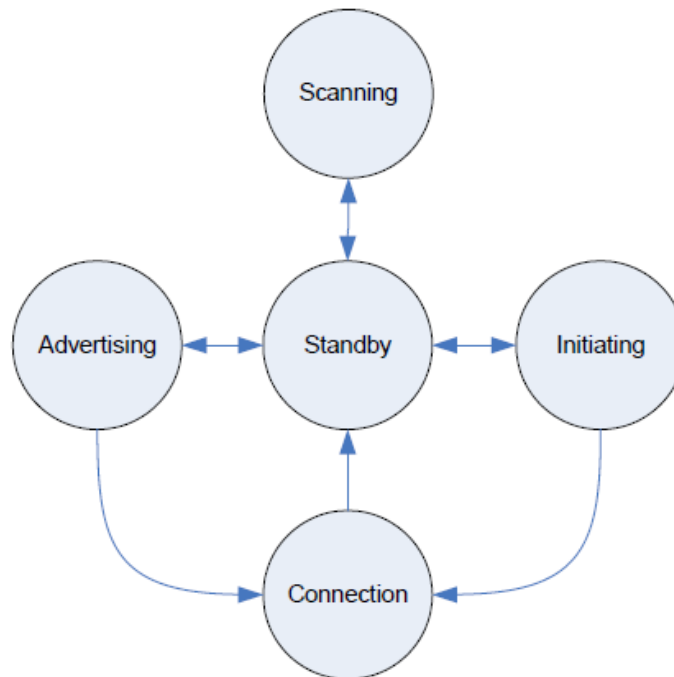


Figure 135 : Les 5 états qu'une machine peut prendre pendant l'utilisation du Bluetooth.

Standby State	La couche liaison ne transmet ou ne reçoit aucun paquet. C'est le seul état qui est atteignable par tous les autres.
Advertising State	Dans cet état la couche liaison va transmettre des annonces de canaux.
Scanning State	La couche liaison va écouter les annonces de canaux par les appareils qui sont en mode advertising.
Initiating State	Écoute d'une annonce de canal spécifique d'un appareil et va répondre pour initier la connexion avec cet appareil.
Connection State	État de connexion avec 2 rôles distincts : Master et Slave (ces rôles ont été définis pour définir des timings de transmission).

Figure 136 : Liste décrivant les 5 états.

Annexe 6 : Qualité audio dans les systèmes audio en réseau

Qualité audio

Depuis que les premiers systèmes audio numérique ont été lancés dans les années 1980, la transition de l'analogique au numérique dans l'industrie audio est presque terminée. La majorité de ces systèmes utilisent des composants numériques pour mixer, traiter et distribuer le son. De la même manière, dans les 10 prochaines années nous verrons une transition des systèmes d'informations fermés, basés sur la topologie de distribution « point à point » vers des systèmes « ouvert » s'appuyant sur une architecture réseau. La transition analogique-numérique a posé de nouvelles contraintes concernant la qualité audio et sonore, tout comme la transition vers des systèmes connectés pose des problèmes de niveaux et de cohérence temporelle.

Dans le domaine de la neurologie, le système permettant à l'Homme d'entendre est appelé le système auditif. C'est une structure biomécanique qui convertit les ondes acoustiques en des influx nerveux pour être traités par le cerveau, ces ondes atteignent les oreilles à travers différents milieux comme l'air et les os de l'oreille. Par ce procédé l'Homme peut avoir une sensation d'ouïe, cette sensation est provoqué par l'entente de signaux audio mais peut être fortement influencée par toutes les autres sensations (mémoire, vision, odeur et le toucher).

Nous allons définir quelques termes qui vont nous permettre de ne pas confondre les différentes notions :

- Signal audio : Portion de tout signal variant dans le temps qui tombe dans la plage audible du système auditif humain provoquant une sensation d'ouïe. Le signal peut être acoustique, électronique ou numérique.
- Processus audio : Concerne la production, le transport, la modification et le stockage d'un signal audio.
- Système audio : C'est un système qui produit un signal audio.
- Caractéristiques audio : Caractéristique physique d'un signal audio, par exemple le niveau, la fréquence et le temps.
- Caractéristiques d'un système audio : Caractéristique physique d'un système audio, par exemple la gamme dynamique (rapport de la plus grande valeur à la plus petite valeur d'une grandeur, décibels par exemple), plage de fréquence ou de temps.
- Source sonore : Cela peut être une voix humaine, la musique d'un instrument ou n'importe quel phénomène qui produit un signal audio.

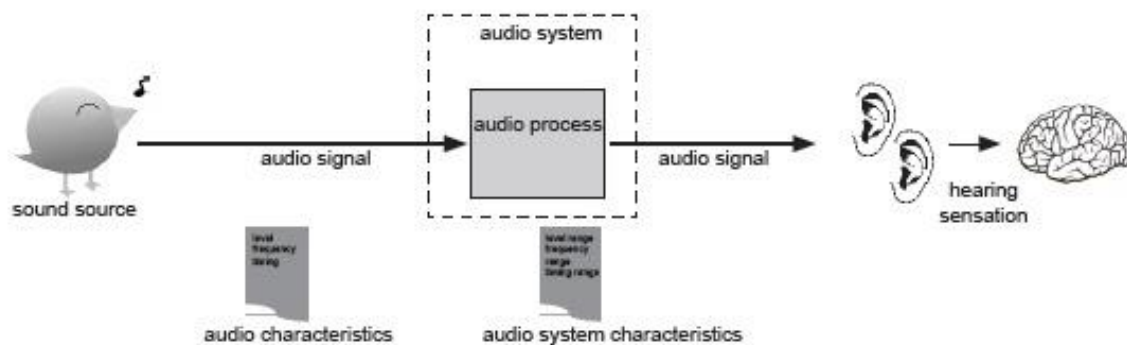


Figure 137 : Schéma montrant un système audio.

Comme nous l'avons énoncé précédemment l'entente d'un signal audio peut être fortement influencé par d'autres facteurs et comme chaque personne a des sensations différentes le même signal audio va provoquer une sensations d'écoute différente chez les sujets, un son différent.

Le son décrit la sensation d'écoute subjective produit par la stimulation du système auditif de la personne en train d'écouter un signal audio. Par ailleurs une source sonore désigne le point d'origine du signal audio.

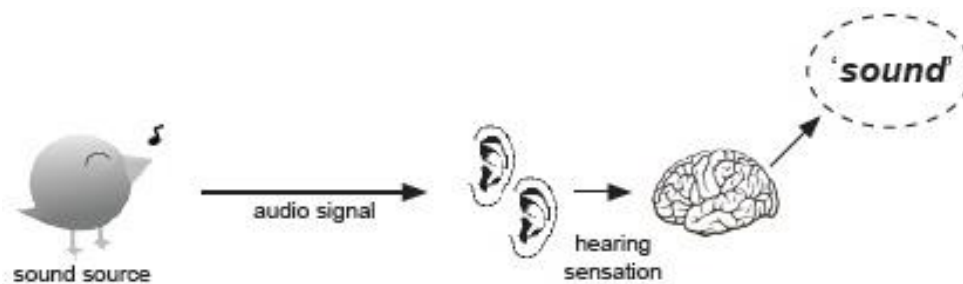


Figure 138 : Schéma montrant comment nous percevons les sons.

Un système audio modifie les caractéristiques d'un signal audio en appliquant son processus. On peut décomposer le processus audio en 3 parties distinctes :

- Limitation : Les limites physiques d'un système pour représenter des signaux en niveau, en fréquence et en temps. Par exemple il est très fortement déconseillé de dépasser les 120dB sous peine d'avoir des lésions permanentes de l'oreille. La plage de fréquence que l'Homme peut entendre s'étend de 20Hz à 20KHz (en général).
- Modification accidentelle : Modification d'un signal audio par un processus non voulu dans le système audio. On peut compter l'égalisation, la distorsion, la compression comme des processus non voulus. La plupart du temps ces modifications sont considérées comme impactant négativement la qualité mais dans certains cas si sa perception est jugée bonne par un assez grand nombre de personne alors le constructeur peut décider de ne pas corriger cette modification ; on passe donc d'une modification accidentelle à une modification prévue.

- Modification prévu : La modification est cette fois-ci voulue par le processus dans le système audio. Généralement utilisé pour améliorer la qualité du son auprès d'une attente spécifique.

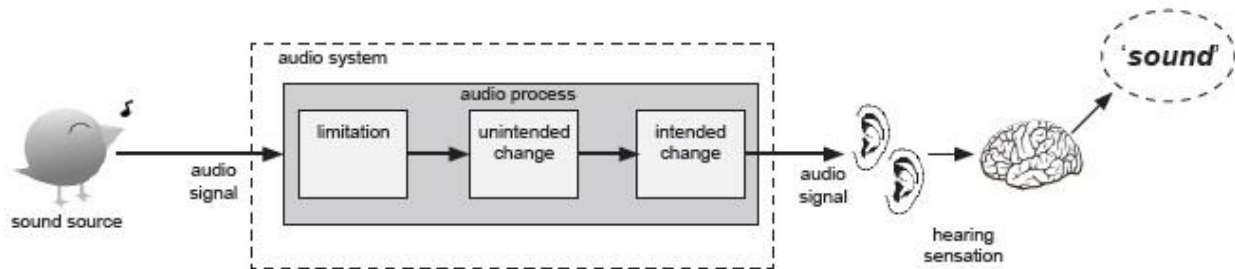


Figure 139 : Les différents types de changements provoqués par le processus audio.

La qualité exprime la notion de conformité aux besoins, selon Philip Bayard Crosby la qualité est toujours liée aux besoins établis pour la sortie du processus. Il faut différencier le terme de « qualité audio » qui fait référence aux caractéristiques physique d'un signal audio et le terme « qualité sonore » qui mentionne les caractéristiques perceptuelles provoquées par la sensation d'écoute. Par exemple la qualité audio peut être établit grâce à des spécificités électriques fondées sur des standards (ISO, AES, IEC). Qualifier la qualité sonore est plus complexe car elle est individuelle à chaque personne. On utilise souvent des mots comme « chaud » ou « transparent » pour qualifier ces caractéristiques.

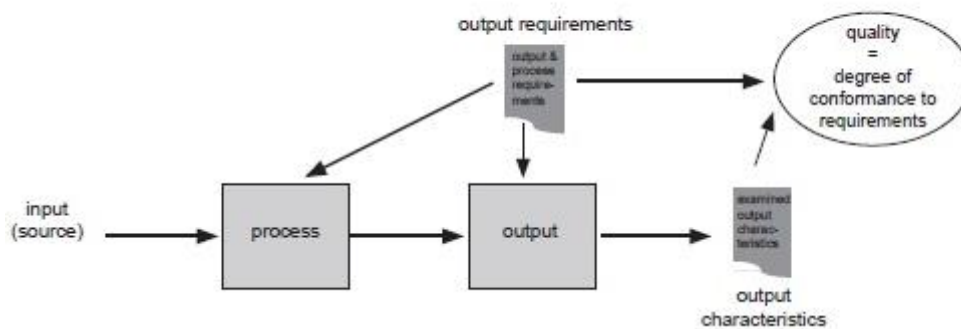


Figure 140 : Schéma montrant la gestion de la qualité.

Un signal audio examiné devrait représenter avec conformité et précision le signal audio d'origine, peu importe les modifications voulues d'un système audio. S'il n'y a pas de système audio entre la source et l'oreille alors le signal est exactement celui généré par la source. On pourrait essayer de définir la qualité audio comme le degré de précision avec lequel le signal audio examiné ressemble au signal audio d'origine sans prendre en compte les changements volontaires d'un système audio. De la même manière la qualité d'un système audio est décrite par la capacité à transporter et transformer un signal audio avec précision.

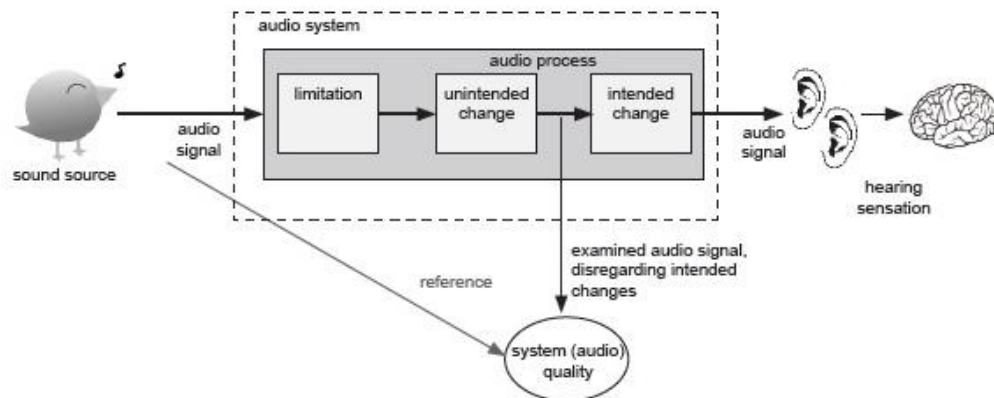


Figure 141 : Schéma de la qualité d'un système audio.

Pour la qualité sonore nous allons proposer plusieurs affirmations :

- La qualité sonore est le degré de satisfaction de l'attente ou de la sensation d'écoute préférée d'un individu résultant de l'entente d'un signal audio.
- La qualité sonore d'une source est le degré de satisfaction de l'attente ou de la sensation d'écoute préférée d'un individu résultant de l'entente d'un signal audio en ne prenant pas compte les limitations ou les changements d'un système audio.
- La qualité sonore d'un système est le degré de satisfaction de l'attente ou de la sensation d'écoute préférée d'un individu résultant d'un changement voulu sur un signal audio, par un système audio sur une source sonore.

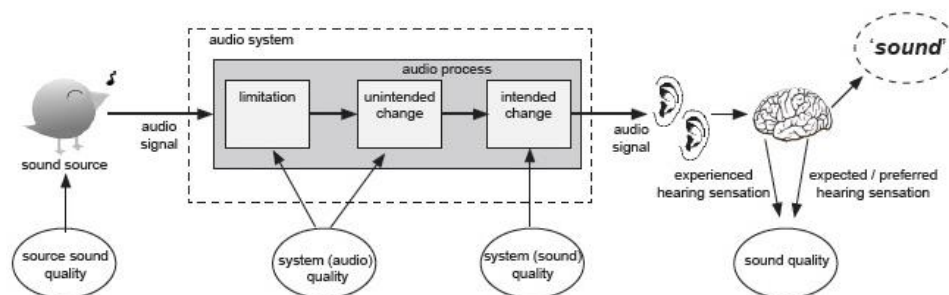


Figure 142 : Schéma de la qualité sonore.

Pour éclaircir les notions si les qualités audio et sonore sont un pourcentage Q, avec 0% et 100% respectivement comme les qualités minimale et maximale alors la qualité sonore d'un signal audio expérimenté par un individu est le produit de la qualité de la sonore de la source, de la qualité audio du système et de la qualité sonore du système. On a donc la formule suivante :

$$Q_{attendue(son)} = Q_{source(son)} * Q_{système(audio)} * Q_{système(son)}$$

Systèmes audio en réseau

Un système audio est une Collection de composants connectés ensemble pour traiter des signaux audio, afin d'améliorer la qualité sonore d'un système.

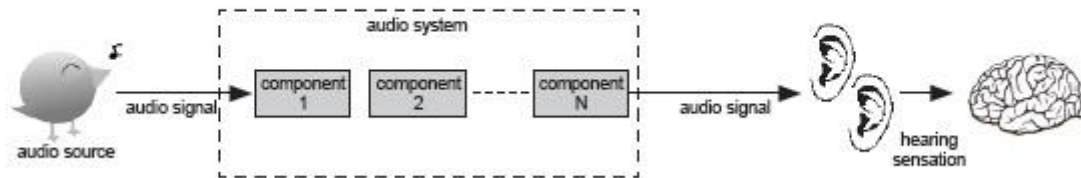


Figure 143 : Schéma des composants d'un système audio.

Fonction	Description
Conversion	Conversion de format des signaux audio
Transport	Transport des signaux audio, dans les câbles par exemple
Stockage	Stockage pour édition , transport et lecture en utilisant des supports audio tels que des disques dur, CD...
Mixage	Mélange de plusieurs entrées sur plusieurs sorties
Modifications	Egalisation, compression, amplification...

Figure 144 : Tableau montrant les types de traitements audio.

Format	Description
Acoustique	Signaux audio sous forme d'ondes de pression dans l'air
Analogique	Signaux audio sous forme de tension sans discontinuité
Numérique	Signaux audio sous forme de données numériques (exemple : résolution 16/24 bits, fréquence d'échantillonnage 44,1/48/88,2/96 KHz)
Données issues d'un réseau	Signaux audio transportées en streaming ou sous forme de paquets commutés (Ethernet)

Figure 145 : Tableau montrant les différents format audio.

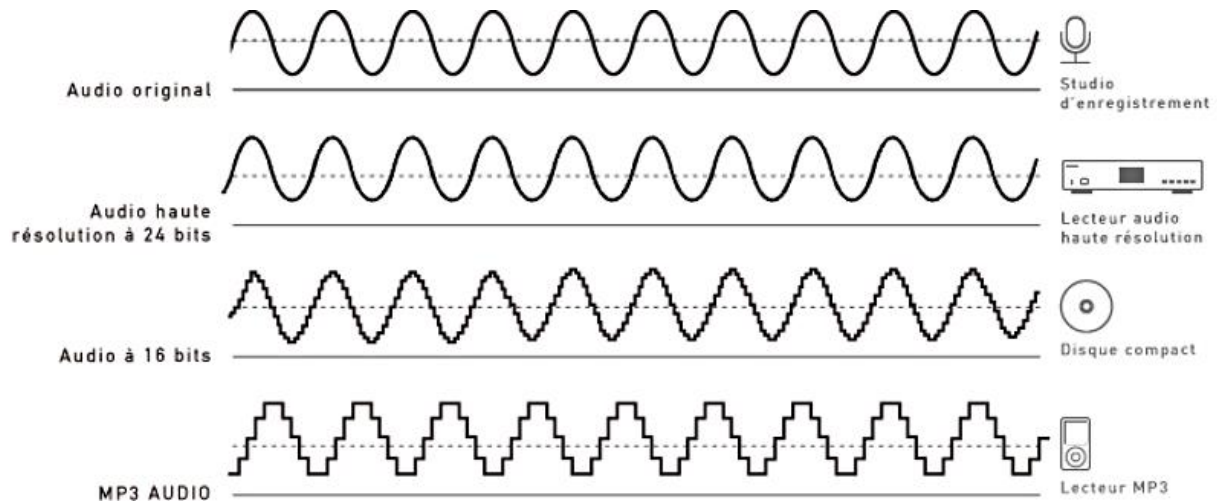


Figure 146 : Figure montrant la dégradation du signal à travers différents formats.

Format de source	Format de destination	Convertisseur source/destination
Acoustique	Analogique	Microphone
Analogique	Numérique/Réseau	CAN (convertisseur Analogique-Numérique)
Numérique/Réseau	Analogique	CNA (convertisseur Numérique-Analogique)
Analogique	Acoustique	Haut-parleur

Figure 147 : Liste des composants effectuant les conversions.

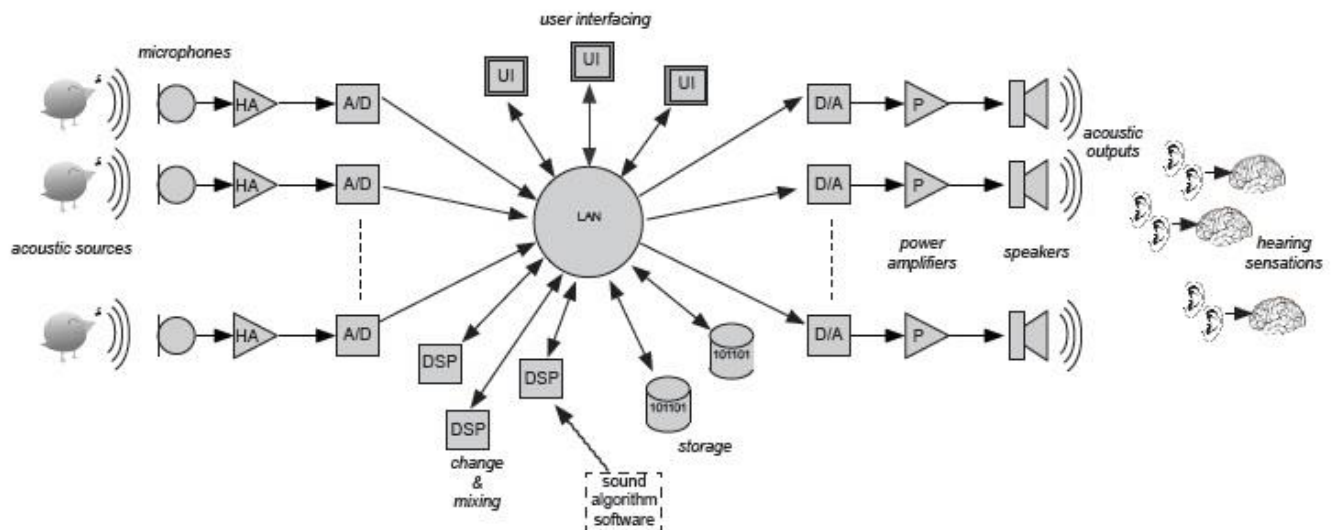












Figure 148 : Exemple d'un système audio en réseau.

Nom	Icone	Description
Source acoustique		Génère des vibrations et le communique à l'air ambiant. Tout ce qui peut produire un son (instrument, chant, vent, moteur...)
Système auditif humain		Composé de deux oreilles et du cerveau. C'est le cerveau qui crée la sensation auditive. Fréquence comprise entre 20Hz et 20KHz et pour un niveau maximale de 120dB
Microphone		Convertit les signaux acoustiques en signaux électriques dans le domaine analogique. Bobine/aimant pour générer le signal ou capteurs piézo-électriques.
Préampli du micro		Tension nominale efficace de référence = 0,775V. Mais la sortie d'un micro = 0,3mV, il faut donc amplifier ce signal pour l'amener vers les 0,775V. HA = Head Amps.
CAN		Convertir signaux analogiques en données numériques pour traitements ultérieurs dans des systèmes audio numériques (appelé l'échantillonnage). Résolution courant = 24 bits.
Réseau de distribution		Collection de composants servant à transférer les données depuis et vers tous les emplacements physiques dans le système audio. Ce réseau peut être point à point. Basé sur Ethernet (Dante, EtherSound, CobraNet).
Modification et mixage (DSP)		Modifier en temps réel et mixer les signaux audio. On peut les combiner avec des FPGA pour obtenir des puissances de calcul supérieur et s'assurer de ne pas avoir des erreurs de calculs dépassant le bruit de fond.
Stockage (enregistrement, lecture, édition)		Grâce au stockage un processus audio peut se répartir sur plusieurs système audio pour des tâches différentes (exemple données enregistrées sur un disque dur puis éditées sur un autre disque dur).
CNA		Convertir les données audio numériques en signaux électriques analogiques pour envoi aux amplificateurs de puissance.
Amplificateur de puissance		Augmente la tension du signal d'entrée qu'il reçoit, son signal de sortie est de basse impédance afin de transférer dans les meilleures conditions possible l'énergie jusqu'au haut-parleurs.



Enceinte acoustique		Convertit le signal électrique basse impédance qu'elles reçoivent en signaux acoustiques. Souvent les enceintes utilisent plusieurs haut-parleurs chacun générant son énergie acoustique pour une meilleure qualité.
Interface utilisateur		Permettre aux ingénieurs d'utiliser les systèmes audio. La plupart du temps cette interface est matérielle (potentiomètre, fader) mais aujourd'hui on se dirige vers des interfaces graphiques permettant le contrôle à distance.

Figure 149 : Tableau précisant les différents composants d'un système audio en réseau.

Annexe 7 : Interface Web - Partie programmation

Script.sh

```
#!/bin/bash

if [ $# -ne 1 ]
then
    echo "Erreur argument"
fi

case "$1" in
    1)
        ifconfig eth0
        ;;
    2)
        ifconfig wlan0
        ;;
    3)
        ping -c1 8.8.8.8
        ;;
    4)
        /etc/init.d/bluetooth status
        ;;
    5)
        shairport-sync --statistics
        ;;
    6)
        /etc/init.d/dhcpd status
        ;;
    7)
        pactl list sinks | grep '^[:space:]Volume:' | \
head -n $(( $SINK + 1 )) | tail -n 1 | sed -e 's,.* \([0-9][0-9]*\)%.*,\1,'
        ;;
    8)
        amixer -D pulse sset Master 5%+
        ;;
    9)
        amixer -D pulse sset Master 5%-
        ;;
    *)
        echo "erreur"
        ;;
esac
```

Reboot.php

```
<!DOCTYPE html>
<html>
```

```
<title>RaspSound</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/3/w3.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato">
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Montserrat">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<link rel="icon" type="image/png" href="fav.png">
<style>
body, h1, h2, h3, h4, h5, h6 {font-family: "Lato", sans-serif}
.w3-bar, h1, button {font-family: "Montserrat", sans-serif}
.fa-anchor, .fa-coffee {font-size:200px}
</style>
<body>

<script language="JavaScript">
function t() {
var compteur=document.getElementById('compteur');
s=duree;
m=0;
if(s<0) {
document.location.href="index.php"
} else {
if(s>59)
{
m=Math.floor(s/60);
s=s-m*60
}
if(s<10)
{
s="0"+s
}
if(m<10)
{
m="0"+m
}
compteur.innerHTML="<br>Votre RaspSound redémarre, redirection
dans : " + m+": "+s
}
duree=duree-1;
window.setTimeout("t();", 999);
}
</script>

<!-- Header -->
<header class="w3-container w3-blue-grey w3-center w3-padding-128">
<?php
//shell_exec('sudo /sbin/reboot');
?>
<h1 class="w3-margin w3-jumbo">Redémarrage de votre RaspSound en
cours</h1>
<i class="fa fa-spinner fa-spin fa-3x fa-fw"></i>
<div id="compteur" class="w3-xlarge"><div>
<script language="JavaScript">
duree="120";
```

```
        t();  
    </script>  
</header>  
  
</body>  
</html>
```

log.php

```
<!DOCTYPE html>  
<html>  
<title>RaspSound</title>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<link rel="stylesheet" href="css/w3.css">  
<link rel="stylesheet" href="css/lato.css">  
<link rel="stylesheet" href="css/montserrat.css">  
<link rel="stylesheet" href="css/font-awesome.min.css">  
<link rel="icon" type="image/png" href="fav.png">  
<style>  
body, h1, h2, h3, h4, h5, h6 {font-family: "Lato", sans-serif}  
.w3-bar, h1, button {font-family: "Montserrat", sans-serif}  
.fa-anchor, .fa-coffee {font-size:200px}  
</style>  
<body>  
  
<!-- Navbar -->  
<div class="w3-top">  
  <div class="w3-bar w3-red w3-card-2 w3-left-align w3-large">  
    <a class="w3-bar-item w3-button w3-hide-medium w3-hide-large w3-right  
w3-padding-large w3-hover-white w3-large w3-red" href="javascript:void(0);"  
onclick="myFunction()" title="Toggle Navigation Menu"><i class="fa fa-  
bars"></i></a>  
    <a href="index.php" class="w3-bar-item w3-button w3-padding-large w3-  
white">Accueil</a>  
    <a href="log.php" class="w3-bar-item w3-button w3-hide-small w3-  
padding-large w3-hover-white">&Eacute;tat de l'appareil</a>  
    <a href="volume.php" class="w3-bar-item w3-button w3-hide-small w3-  
padding-large w3-hover-white">Volume</a>  
  </div>  
  
  <!-- Navbar on small screens -->  
  <div id="navDemo" class="w3-bar-block w3-white w3-hide w3-hide-large w3-  
hide-medium w3-large">  
    <a href="log.php" class="w3-bar-item w3-button w3-padding-  
large">&Eacute;tat de l'appareil</a>  
    <a href="volume.php" class="w3-bar-item w3-button w3-padding-  
large">Volume</a>  
  </div>  
</div>  
  
<!-- First Grid -->  
<div class="w3-row-padding w3-padding-64 w3-container">  
  <div class="w3-content">  
    <div class="w3-twothird">  
      <h1>&Eacute;tat du réseau</h1>  
      <p class="w3-text-grey">Nom symbolique de la machine : </p>  
<?php
```

```
$output = shell_exec('hostname');
echo "<pre>$output</pre>";
?>
<p class="w3-text-grey"><i class="fa fa-link"></i>
&nbsp;&Eacute;tat du port Ethernet : </p>
<?php
exec('/bin/bash ./script.sh 1', $res);
foreach ($res as $item) {
    echo "<pre>$item</pre>";
}
?>
<p class="w3-text-grey"></br><i class="fa fa-wifi"></i>
&nbsp;&Eacute;tat de l'interface Wi-Fi :
</p>
<?php
unset($res);
exec('/bin/bash ./script.sh 2', $res);
foreach ($res as $item) {
    echo "<pre>$item</pre>";
}
?>
<p class="w3-text-grey"></br><i class="fa fa-google"></i>
&nbsp;&Eacute;tat de la connectivité internet (test sur serveur Google)
:
</p>
<?php
unset($res);
exec('/bin/bash ./script.sh 3', $res);
foreach ($res as $item) {
    echo "<pre>$item</pre>";
}
?>
</div>
</div>
</div>
<!-- Second Grid -->
<div class="w3-row-padding w3-padding-64 w3-container">
<div class="w3-content">
<div class="w3-twothird">
<h1>&Eacute;tat des services</h1>
<p class="w3-text-grey"><i class="fa fa-bluetooth-b"></i>
&nbsp;&Eacute;tat du service Bluetooth :
</p>
<?php
unset($res);
exec('/bin/bash ./script.sh 4', $res);
foreach ($res as $item) {
    echo "<pre>$item</pre>";
}
?>
<p class="w3-text-grey"></br><i class="fa fa-apple"></i>
&nbsp;&Eacute;tat du service Airport :
</p>
<?php
unset($res);
exec('/bin/bash ./script.sh 5', $res);
foreach ($res as $item) {
```

```
        echo "<pre>$item</pre>";
    }
    ?>
<p class="w3-text-grey"></br><i class="fa fa-link"></i>
    &nbsp;&Eacute;tat du service DHCP :
</p>
<?php
unset($res);
exec('/bin/bash ./script.sh 6', $res);
foreach ($res as $item) {
    echo "<pre>$item</pre>";
}
?>
</div>
</div>
</div>

<div class="w3-container w3-center">
<form method="POST" action="reboot.php">
    <input type="submit" value="Redémarrer le RaspSound" name="reboot"
class="w3-button w3-black w3-padding-large w3-margin-top w3-margin-bottom "
/>
</form>

<script>
// Used to toggle the menu on small screens when clicking on the menu
button
function myFunction() {
    var x = document.getElementById("navDemo");
    if (x.className.indexOf("w3-show") == -1) {
        x.className += " w3-show";
    } else {
        x.className = x.className.replace(" w3-show", "");
    }
}
</script>

</body>
</html>
```

index.php

```
<!DOCTYPE html>
<html>
<title>RaspSound</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/w3.css">
<link rel="stylesheet" href="css/lato.css">
<link rel="stylesheet" href="css/montserrat.css">
<link rel="stylesheet" href="css/font-awesome.min.css">
<link rel="icon" type="image/png" href="fav.png">
<style>
body, h1, h2, h3, h4, h5, h6 {font-family: "Lato", sans-serif}
.w3-bar, h1, button {font-family: "Montserrat", sans-serif}
.fa-anchor, .fa-coffee {font-size:200px}
</style>
<body>
```

```
<!-- Navbar -->
<div class="w3-top">
  <div class="w3-bar w3-red w3-card-2 w3-left-align w3-large">
    <a class="w3-bar-item w3-button w3-hide-medium w3-hide-large w3-right
w3-padding-large w3-hover-white w3-large w3-red" href="javascript:void(0);"
onclick="myFunction()" title="Toggle Navigation Menu"><i class="fa fa-
bars"></i></a>
    <a href="index.php" class="w3-bar-item w3-button w3-padding-large w3-
white">Accueil</a>
    <a href="log.php" class="w3-bar-item w3-button w3-hide-small w3-
padding-large w3-hover-white">&Eacute;tat de l'appareil</a>
    <a href="volume.php" class="w3-bar-item w3-button w3-hide-small w3-
padding-large w3-hover-white">Volume</a>
  </div>

  <!-- Navbar on small screens -->
  <div id="navDemo" class="w3-bar-block w3-white w3-hide w3-hide-large w3-
hide-medium w3-large">
    <a href="log.php" class="w3-bar-item w3-button w3-padding-
large">&Eacute;tat de l'appareil</a>
    <a href="volume.php" class="w3-bar-item w3-button w3-padding-
large">Volume</a>
  </div>
</div>

<!-- Header -->
<header class="w3-container w3-blue-grey w3-center w3-padding-128">
  <h1 class="w3-margin w3-xxlarge">Bienvenue sur votre RaspSound</h1>
  <p class="w3-xlarge">Retrouver les fonctionnalités essentielles ci
dessous</p>
</header>

<!-- First Grid -->
<div class="w3-row-padding w3-light-grey w3-padding-64 w3-container">
  <div class="w3-content">

    <div class="w3-twothird">
      <h1>Que puis-je faire</h1>
      <h5 class="w3-padding-32">&Agrave; partir de cette interface vous
pouvez ajuster le volume de l'appareil et consulter l'état des principaux
services actifs sur l'appareil</h5>

      <p class="w3-text-grey">En cliquant sur état de l'appareil vous aurez
accès aux informations réseaux et aux services audio et réseaux s'exécutant
sur le RaspSound.
      Depuis la page de volume, vous pouvez ajuster le volume par cran de
5%. La conception du site permet une réaction quasi-instantanée de la
plateforme.</p>
    </div>
  </div>
</div>

<!-- Second Grid -->
<div class="w3-row-padding w3-padding-64 w3-container">
  <div class="w3-content">
    <div class="w3-twothird">
```

```
<h1>Informations complémentaires</h1>
<h5 class="w3-padding-32">Plus d'informations sur notre projet ?</h5>

<p class="w3-text-grey">Si notre projet vous para&icirc;t pertinent
n'hésitez pas à lire notre rapport en cliquant sur ce lien.</p>
</div>

</div>
</div>

<div class="w3-container w3-black w3-center w3-opacity w3-padding-64">
  <h1 class="w3-margin w3-xlarge">Un projet de Benjamin BONHOMME et
Benjamin RICHER<br>Licence disponible dans le rapport.</h1>
</div>

<!-- Footer -->
<footer class="w3-container w3-padding-64 w3-center w3-opacity">
  <div class="w3-xlarge w3-padding-32">
    <p class="w3-large">Développé à l'aide de : </p>
    <i class="fa fa-html5"></i>
    <i class="fa fa-firefox"></i>
    <i class="fa fa-linux"></i>
    <i class="fa fa-stack-overflow"></i>
    <i class="fa fa-bluetooth"></i>
    <i class="fa fa-skype"></i>
    <i class="fa fa-wifi"></i>
    <p class="w3-large">Et beaucoup de : </p>
    <i class="fa fa-coffee "></i>
  </div>
  <p>Propulsé par <a href="https://www.w3schools.com/w3css/default.asp"
target="_blank">w3.css</a> et <a href="http://fontawesome.io"
target="_blank">FA</a></p>
</footer>

<script>
// Used to toggle the menu on small screens when clicking on the menu
button
function myFunction() {
  var x = document.getElementById("navDemo");
  if (x.className.indexOf("w3-show") == -1) {
    x.className += " w3-show";
  } else {
    x.className = x.className.replace(" w3-show", "");
  }
}
</script>

</body>
</html>
```

volume.php

```
<!DOCTYPE html>
<html>
<title>RaspSound</title>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/w3.css">
<link rel="stylesheet" href="css/lato.css">
<link rel="stylesheet" href="css/montserrat.css">
<link rel="stylesheet" href="css/font-awesome.min.css">
<link rel="icon" type="image/png" href="fav.png">
<style>
body,h1,h2,h3,h4,h5,h6 {font-family: "Lato", sans-serif}
.w3-bar,h1,button {font-family: "Montserrat", sans-serif}
.fa-anchor,.fa-coffee {font-size:200px}
</style>
<body>

<!-- Navbar -->
<div class="w3-top">
  <div class="w3-bar w3-red w3-card-2 w3-left-align w3-large">
    <a class="w3-bar-item w3-button w3-hide-medium w3-hide-large w3-right
w3-padding-large w3-hover-white w3-large w3-red" href="javascript:void(0);"
onclick="myFunction()" title="Toggle Navigation Menu"><i class="fa fa-
bars"></i></a>
    <a href="index.php" class="w3-bar-item w3-button w3-padding-large w3-
white">Accueil</a>
    <a href="log.php" class="w3-bar-item w3-button w3-hide-small w3-
padding-large w3-hover-white">&Eacute;tat de l'appareil</a>
    <a href="volume.php" class="w3-bar-item w3-button w3-hide-small w3-
padding-large w3-hover-white">Volume</a>
  </div>

  <!-- Navbar on small screens -->
  <div id="navDemo" class="w3-bar-block w3-white w3-hide w3-hide-large w3-
hide-medium w3-large">
    <a href="log.php" class="w3-bar-item w3-button w3-padding-
large">&Eacute;tat de l'appareil</a>
    <a href="volume.php" class="w3-bar-item w3-button w3-padding-
large">Volume</a>
  </div>
</div>

<!-- First Grid -->

<div class="w3-row-padding w3-padding-64 w3-container">
<?php
if(isset($_POST['down'])) {
    exec('/bin/bash ./script.sh 9');
}
if(isset($_POST['up'])) {
    exec('/bin/bash ./script.sh 8');
}
?>

  <div class="w3-content ">
    <h1>Gestion du volume</h1>
    <i class="fa fa-volume-off" style="font-size:8em"></i>
    <?php
    exec('/bin/bash ./script.sh 7', $res);
    echo "<b style=\"font-size:6em\"> $res[0]</b>";
    ?>
  </div>
</div>
```



```
    <form method="post" action="volume.php">
    <input type="submit" name="up" value="&#43;" class="w3-padding-
large w3-button w3-grey"/>
    </form>
    <br>
    <form method="post" action="volume.php">
    <input type="submit" name="down" value="&minus;" class="w3-padding-
large w3-button w3-grey"/>
    </form>
  </div>
</div>

<script>
// Used to toggle the menu on small screens when clicking on the menu
button
function myFunction() {
  var x = document.getElementById("navDemo");
  if (x.className.indexOf("w3-show") == -1) {
    x.className += " w3-show";
  } else {
    x.className = x.className.replace(" w3-show", "");
  }
}
</script>

</body>
</html>
```

Page Principal

The screenshot shows a web interface with a navigation bar at the top containing three items: 'Accueil', 'État de l'appareil', and 'Volume'. The 'État de l'appareil' item is highlighted with a red background. Below the navigation bar is a large blue-grey box with the text 'Bienvenue sur votre RaspSound' and 'Retrouver les fonctionnalités essentielles ci dessous'. Underneath this is a section titled 'Que puis-je faire' with a paragraph explaining that users can adjust volume and check service status. A smaller paragraph below describes network and audio service access. The next section is 'Informations complémentaires' with a link for more information. A dark grey box at the bottom contains the text 'Un projet de Benjamin BONHOMME et Benjamin RICHER' and 'Licence disponible dans le rapport.'. Below this are icons for development tools and a laptop icon, with text indicating the project was developed with these tools and proposed by w3css and FA.

Accueil État de l'appareil Volume

Bienvenue sur votre RaspSound

Retrouver les fonctionnalités essentielles ci dessous

Que puis-je faire

À partir de cette interface vous pouvez ajuster le volume de l'appareil et consulter l'état des principaux services actifs sur l'appareil

En cliquant sur état de l'appareil vous aurez accès aux informations réseaux et aux services audio et réseaux s'exécutant sur le RaspSound. Depuis la page de volume, vous pouvez ajuster le volume par cran de 5%. La conception du site permet une réaction quasi-instantanée de la plateforme.

Informations complémentaires

Plus d'informations sur notre projet ?

Si notre projet vous paraît pertinent n'hésitez pas à lire notre rapport en cliquant sur ce lien.

Un projet de Benjamin BONHOMME et Benjamin RICHER
Licence disponible dans le rapport.

Développé à l'aide de :

Et beaucoup de :

Proposé par w3css et FA

Figure 150 : Page d'accueil de notre interface Web.

Table des illustrations

Figure 1 : Diagramme bête à cornes, permettant d'indiquer à quoi sert le produit.	5
Figure 2 : Diagramme pieuvre, permet de relier le produit à son environnement extérieur (tout ce qui peut influencer sur son fonctionnement).	6
Figure 3 : Liste des idées d'améliorations que nous avons proposé.....	7
Figure 4 : Diagramme de Gantt du semestre 3.	9
Figure 5 : Diagramme de Gantt du semestre 4.	9
Figure 6 : Diagramme représentant les interactions entre les composants du Raspberry Pi.	11
Figure 7 : Tableau comparatif regroupant les caractéristiques du Raspberry Pi B+ et Raspberry Pi 3.	12
Figure 8 : Certaines des nouveautés incluses dans le Raspberry Pi 3.....	12
Figure 9 : Disposition du Raspberry Pi B+ à gauche et celle du Raspberry Pi 3 à droite.....	13
Figure 10 : Image thermique du Raspberry Pi B+. On peut voir que la température ne dépasse pas les 50°C.....	14
Figure 11 : Image thermique du Raspberry Pi 3. On peut observer que la température peut dépasser les 100 °C aux alentours du processeur.....	14
Figure 12 : Au démarrage du Raspberry Pi avec une carte MicroSD ayant NOOBS chargé dessus, c'est le menu que l'on obtient.....	16
Figure 13 : Explication sous forme de schéma du fonctionnement de la compilation croisée.	16
Figure 14 : Tableau qui recense l'ensemble des paquets utilisés.	18
Figure 15 : Tableau qui recense les caractéristiques des protocoles Bluetooth et Airplay.	20
Figure 16 : Portée du Bluetooth.....	20
Figure 17 : Portée de Airplay.....	20
Figure 18 : Tableau recensant l'ensemble des avantages et inconvénients des 2 protocoles.....	21
Figure 19 : Logo d'Apache (http server).....	22
Figure 20 : Logo de PHP5.....	23
Figure 21 : Les différents langages utilisés dans ce projet.....	24
Figure 22 : Aperçu de la première interface utilisée.....	25
Figure 23 : Template pour l'interface finale utilisée dans notre projet.....	26
Figure 24 : Notre principal problème concernant l'utilisation des feuilles CSS sur le téléphone.	27
Figure 25 : Logo JavaScript.....	28
Figure 26 : Logo BASH, the bourne again shell.	29
Figure 27 : Logo de mod-python	30
Figure 28 : Logo de MobaXterm.....	32
Figure 29 : Logo de GNU nano.	32
Figure 30 : Logo de Notepad++.....	32
Figure 31 : Logo Mozilla Firefox.	33
Figure 32 : Inspecteur d'élément sous Firefox.....	33

Figure 33 : Enregistrement nous montrant les performances lors d'un changement de volume.	34
Figure 34 : Grâce à l'inspecteur d'élément nous pouvons facilement voir chaque élément de la page.	34
Figure 35 : Information concernant les marges.	35
Figure 36 : Spectre Bluetooth.	38
Figure 37 : Spectre Airplay.	38
Figure 38 : Photo du montage pour observer la consommation électrique.....	39
Figure 39 : Tableau avec les valeurs de tension, courant et puissance.	39
Figure 40 : Schéma montrant où se situe notre produit en terme de prix.....	40
Figure 41 : Schéma montrant où se situe notre produit en terme de fonctionnalités (nombre de protocoles implémentés).	41
Figure 42 : Raspberry Pi Starter Kit, contient la carte Raspberry Pi B+, une carte SD de 8Gb avec NOOBS préinstallé dessus, le boîtier pour la carte, le chargeur EU/US, un manuel d'utilisation.	43
Figure 43 : La carte Raspberry Pi B+ dans son boîtier.	43
Figure 44 : La carte SD d'une capacité de 8Gb avec NOOBS préinstallé dessus.	44
Figure 45 : Le chargeur pour le Raspberry Pi B+ avec les adaptateurs EU/US.	44
Figure 46 : Le manuel d'utilisation pour le Raspberry Pi.	44
Figure 47 : Le Raspberry Pi B+ et le Pi 3, la différence cruciale pour notre projet est que le Pi 3 intègre nativement le Wi-Fi et le Bluetooth.	45
Figure 48 : Le chargeur délivré avec le Raspberry Pi 3, comparé avec l'autre chargeur la seule différence est l'ampérage qui est de 2,5A pour le Pi 3 et de 2A pour le Pi B+.	45
Figure 49 : Lecteur de carte micro-SD, Samsung Carte Mémoire EVO Micro-SD Classe 10, 16 Go, valeur estimée = 15€ (pour le Pi B+) ; Carte Mémoire Sandisk Micro SDHC, 16 Go (pour le Pi 3), valeur estimée = 10€.	46
Figure 50 : Le Raspberry Pi B+ étant dépourvu d'antennes Wi-Fi/Bluetooth, il a fallu rajouter un dongle Wi-Fi, TP-Link TL-WN725N Nano Adaptateur USB Wi-Fi N 150 Mbps, valeur estimée = 8€.	46
Figure 51 : Dongle Bluetooth pour Raspberry Pi B+, TRIXES Dongle 2.0 de Bluetooth pour Micro ou Raspberry Pi, valeur estimée = 5€.	46
Figure 52 : Un clavier et une souris qui ne sont utiles que pour la première utilisation c'est à dire l'installation et la configuration du Raspberry.	47
Figure 53 : Un écran qui est aussi utile seulement pour les premières utilisations du Raspberry.	47
Figure 54 : Un câble d'alimentation pour l'écran, un câble DVI pour brancher le Raspberry à l'écran et avec l'adaptateur HDMI, un câble Ethernet pour accéder à Internet et faire les mises à jour.....	47
Figure 55 : iFun4U USB multimètre voltmètre ampère, cela va nous permettre de simplement mesurer la consommation électrique du Raspberry Pi, valeur estimée = 9€.	48
Figure 56 : Script que nous avons utilisé pour réaliser les tests de performances.	49
Figure 57 : Résultats bruts des tests effectués sur le Pi B+ et le Pi 3.	50

Figure 58 : Test de température au repos	51
Figure 59 : Test sur les cartes mémoires (fréquence d'horloge en MHz).....	51
Figure 60 : Différences de processeurs, cœurs et de mémoire. Si la fréquence cœur du Pi B+ est à 0 c'est car celui-ci ne possède qu'un seul cœur.	51
Figure 61 : Test de ping entre les 2 versions du Raspberry Pi. On peut voir que le Pi 3 a un ping beaucoup moins élevé.	52
Figure 62 : Test comparatif sur les débits descendant/montant. On peut observer que le Pi 3 a en moyenne un débit 3 fois plus élevé que le Pi B+.	52
Figure 63 : Le premier test de CPU montre que le Pi 3 est largement devant. Ces résultats ne sont pas étonnants car il était prévu que le Pi 3 soit 10 fois plus performant, d'après : https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/	53
Figure 64 : En contrepartie, le Pi B+ chauffe beaucoup moins que le Pi 3 en période d'activité.	53
Figure 65 : Encore une fois le Raspberry Pi 3 montre que ses performances CPU sont bien plus élevées que le Pi B+.....	54
Figure 66 : La température du Pi B+ reste constante alors que le Pi 3 atteint plus de 70°C...	54
Figure 67 : La RAM allant de pair avec le processeur, il n'est pas étonnant que le Pi 3 soit largement devant.....	55
Figure 68 : Depuis le début de ces tests la température du Pi B+ n'a connu aucune réelle augmentation comparé au Pi 3.....	55
Figure 69 : Test de la mémoire RAM avec les temps total, minimum, maximum et moyen en ms.....	55
Figure 70 : Les différents test sur la carte MicroSD. Les deux cartes étant sensiblement de même qualité on peut observer des résultats presque similaires sur les 2 versions.....	56
Figure 71 : Il y a une légère augmentation de température sur Pi 3 lors de ces différents tests.	56
Figure 72 : Téléchargement de Raspbian, daté du 23/09/2016 pour commencer l'installation sur le Raspberry Pi.....	57
Figure 73 : Lecteur de carte MicroSD et carte SD de 16Go pour copier l'image Raspbian sur la carte.....	58
Figure 74 : Etapes pour décompresser l'image sur la carte SD.....	58
Figure 75 : Attendre la fin de l'écriture sur la carte SD.....	59
Figure 76 : Notre carte SD est maintenant "bootable" et contient tout ce qu'il faut pour installer Raspbian.	59
Figure 77 : Insertion de la carte SD dans le Raspberry Pi. Sur le Pi B+ la carte se clips alors que sur le Pi 3 elle se glisse.	60
Figure 78 : Branchements des composants nécessaires à une première installation.	60
Figure 79 : Mettre sous-alimentation le Raspberry, après bien avoir mis la carte SD puis brancher les différents composants.....	61
Figure 80 : Les LED permettent de nous informer que tout se déroule bien comme prévu. A noter que sur le Pi 3 les LED ne sont plus présentes et ont été remplacées par les antennes Wi-Fi/Bluetooth.	61

Figure 81 : Ecran d'accueil validant que Raspbian s'est bien installé.	62
Figure 82 : Bureau du Raspberry Pi B+. Les screenshots ont été prises avec scrot.	62
Figure 83 : Menu de configuration de Raspbian.	63
Figure 84 : On a fait "Expand Filesystem" pour que Raspbian utilise bien tout l'espace disponible.	63
Figure 85 : Nous allons passer notre clavier en AZERTY dans le menu "Internationalisation Options".....	64
Figure 86 : On change l'agencement des touches du clavier.....	64
Figure 87 : On choisit "Generic 105-key PC", qui est le plus général.....	64
Figure 88 : Nous voulons choisir une autre langue que anglais pour le clavier.....	65
Figure 89 : On sélectionne la langue Français pour les l'agencement du clavier.	65
Figure 90 : On prend le Français de base.	65
Figure 91 : On sélectionne la configuration par défaut.	66
Figure 92 : On ne prend pas les touches composées.....	66
Figure 93 : On ne veut pas que Control+Alt+Backspace ferme le serveur X.....	66
Figure 94 : On revient dans le menu "Internationalisation Options", cette fois-ci pour changer la langue du système.....	67
Figure 95 : On sélectionne "Change Locale" pour définir la langue par défaut.....	67
Figure 96 : On va sélectionner « fr_FR.UTF-8 UTF-8 » et « en_GB.UTF-8 UTF-8 » pour avoir les langues anglaise et française.	67
Figure 97 : On spécifie que l'on veut en français par défaut.	68
Figure 98 : Pour changer le mot de passe de l'utilisateur par défaut, à savoir "pi", il faut aller dans le menu "Change User Password".	68
Figure 99 : On tape le mot de passe voulu.....	68
Figure 100 : Le mot de passe a bien été changé.	68
Figure 101 : Mise en place du mot de passe root.	69
Figure 102 : Il faut aller dans "Advanced Options" pour activer le SSH sur Raspbian.	69
Figure 103 : On sélectionne SSH.	69
Figure 104 : On active le SSH.....	69
Figure 105 : On est connecté en SSH sur le Raspberry Pi et on lance VNCserver.	70
Figure 106 : Depuis chrome, on peut installer une application complémentaire VNCviewer. 71	
Figure 107 : Il faut s'authentifier avec le mot de passe défini auparavant.	71
Figure 108 : On a bien accès à l'interface graphique, qui est un peu lente mais fonctionnelle.	72
Figure 109 : Installation du driver nécessaire au fonctionnement du Wi-Fi.....	79
Figure 110 : Modification du fichier /etc/dhcpd.conf.....	80
Figure 111 : Définition statique du réseau sur le fichier /etc/network/interfaces.	81
Figure 112 : Nos modifications ont bien eu un impact car nous avons l'adresse qui avait été définie.....	81
Figure 113 : Configuration de notre réseau Wi-Fi dans le fichier /etc/hostapd/hostapd.conf. Tout est commenté pour comprendre chaque ligne ajoutée. ...	82
Figure 114 : Test pour montrer que le réseau est bien accessible.	83

Figure 115 : Le réseau Raspberry-AP-HiFi apparait bien dans la liste des réseaux Wi-Fi.	83
Figure 116 : Modification du fichier /etc/default/hostapd pour montrer où se trouvent les fichiers de configurations.	84
Figure 117 : Mise en place du fichier /etc/dnsmasq.conf pour attribuer des adresses dynamiquement.	84
Figure 118 : Le DHCP a bien donné une adresse à notre iPhone et on peut se connecter avec l'application Airplay.	85
Figure 119 : Demande des méthodes disponibles pour RTSP.	86
Figure 120 : Ensemble des méthodes disponibles pour RTSP.	86
Figure 121 : Interaction pour la méthode Announce.	87
Figure 122 : Liste des canaux avec leurs descriptions.	87
Figure 123 : Interactions pour la méthode Setup.	88
Figure 124 : Interactions pour la méthode Flush.	88
Figure 125 : Interactions pour la méthode Teardown.	88
Figure 126 : Liste des différents types de contenus avec leurs descriptions.	89
Figure 127 : Payload 96, pour les données audio.	89
Figure 128 : Liste des octets pour la synchronisation avec leurs descriptions.	90
Figure 129 : Payload 84, synchronisation temporelle.	90
Figure 130 : Interactions permettant de changer le volume grâce à la méthode set_parameter.	90
Figure 131 : Tableau illustrant les différents types de Bluetooth et leurs utilisations.	91
Figure 132 : Différence de canaux spectral entre les 2 types de bluetooth.	91
Figure 133 : Couches protocolaire montrant comment fonctionne le protocole Bluetooth. .	92
Figure 134 : Tableau avec la liste des différentes couches et leurs utilités.	93
Figure 135 : Les 5 états qu'une machine peut prendre pendant l'utilisation du Bluetooth.	94
Figure 136 : Liste décrivant les 5 états.	94
Figure 137 : Schéma montrant un système audio.	96
Figure 138 : Schéma montrant comment nous percevons les sons.	96
Figure 139 : Les différents types de changements provoqués par le processus audio.	97
Figure 140 : Schéma montrant la gestion de la qualité.	97
Figure 141 : Schéma de la qualité d'un système audio.	98
Figure 142 : Schéma de la qualité sonore.	98
Figure 143 : Schéma des composants d'un système audio.	99
Figure 144 : Tableau montrant les types de traitements audio.	99
Figure 145 : Tableau montrant les différents format audio.	99
Figure 146 : Figure montrant la dégradation du signal à travers différents formats.	100
Figure 147 : Liste des composants effectuant les conversions.	100
Figure 148 : Exemple d'un système audio en réseau.	100
Figure 149 : Tableau précisant les différents composants d'un système audio en réseau.	102
Figure 150 : Page d'accueil de notre interface Web.	112

Glossaire

Système embarqué : Un système embarqué est défini comme un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche bien précise. Le terme désigne aussi bien le matériel informatique que le logiciel utilisé. Ses ressources sont généralement limitées. Cette limitation est généralement d'ordre spatial (encombrement réduit) et énergétique (consommation restreinte).

ARM : ARM est une société britannique spécialisée dans le développement d'architectures 32 bits de type RISC. Fondée sous la forme d'une coentreprise par Acorn Computers, Apple Computer (maintenant Apple Inc.) et VLSI Technology, ARM développe également un grand nombre de blocs de propriété intellectuelle (IP). Elle est basée sur un modèle économique particulier de la microélectronique : la conception de propriétés intellectuelles (Intellectual Properties). Ainsi il n'est pas possible d'acheter un processeur ARM comme c'est le cas pour Intel. Les cœurs ARM sont intégrés au sein de systèmes sur puces (SoC) complets. Les cœurs de processeurs ARM sont très présents dans les systèmes embarqués (téléphone mobile, console portable, tablette électronique). Les architectures ARM sont des architectures matérielles RISC 32 bits (ARMv1 à ARMv7) et 64 bits (ARMv8)¹ développées par ARM Ltd depuis 1990 et introduites à partir de 1983 par Acorn Computers. Dotés d'une architecture relativement plus simple que d'autres familles de processeurs, et bénéficiant d'une faible consommation.

Raspbian : Raspbian est un système d'exploitation libre et gratuit fondé sur GNU/Linux/Debian et optimisé pour fonctionner sur un Raspberry Pi. Étant donné les ressources limitées de cet ordinateur, Raspbian utilise des logiciels réputés pour être légers tels que le gestionnaire de fenêtres LXDE et le navigateur Web Midori.

Chaîne de compilation croisée : Une chaîne de compilation (toolchain) désigne l'ensemble des paquets utilisés dans le processus de compilation d'un programme, pour un processeur donné. Le compilateur n'est qu'un élément de cette chaîne, laquelle varie selon l'architecture matérielle cible. Une chaîne de compilation croisée est une chaîne compilée pour fonctionner sur l'architecture de processeurs de la machine hôte, mais qui va compiler des logiciels pour une architecture cible différente. La compilation croisée fait donc référence aux chaînes de compilation capables de traduire un code source en code objet dont l'architecture processeur diffère de celle où la compilation est effectuée. Ces chaînes sont principalement utilisées en informatique industrielle et dans les systèmes embarqués.

SSH : SSH (Secure Shell) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Le protocole SSH a été conçu avec l'objectif de remplacer les différents programmes rlogin, telnet, rcp, ftp et rsh.

Système de fichiers : Un système de fichiers (File System) est une façon de stocker les informations et de les organiser dans des fichiers sur des mémoires de masse comme un disque dur. Une telle gestion des fichiers permet de traiter, de conserver des quantités importantes de données ainsi que de les partager entre plusieurs programmes informatiques. Il offre à l'utilisateur une vue abstraite sur ses données et permet de les localiser à partir d'un chemin d'accès. Un système de fichiers s'installe sur une partition (cf. fdisk). On crée un système de fichiers par une opération de formatage.

HTTP : L'HyperText Transfer Protocol, plus connu sous l'abréviation HTTP — littéralement « protocole de transfert hypertexte » — est un protocole de communication client-serveur développé pour le World Wide Web. HTTPS (avec S pour secured, soit « sécurisé ») est la variante du HTTP sécurisée par l'usage des protocoles SSL ou TLS.

HTML : L'HyperText Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques.

PHP : PHP: Hypertext Preprocessor, plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

VPN : En informatique, un réseau privé virtuel, VPN (Virtual Private Network) ailleurs, est un système permettant de créer un lien direct entre des ordinateurs distants. On utilise notamment ce terme dans le travail à distance, ainsi que pour l'accès à des structures de type cloud computing (dans notre projet nous utilisons Hamachi).

DNS : Le Domain Name System (ou DNS, système de noms de domaine) est un service permettant de traduire un nom de domaine en informations de plusieurs types qui y sont associées, notamment en adresses IP de la machine portant ce nom. À la demande de la DARPA (Defense Advanced Research Projects Agency).

CSS : Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

CMS : WordPress est un système de gestion de contenu gratuit (SGC ou content management system (CMS) en anglais) et open-source. Ce logiciel libre écrit en PHP, repose sur une base de données MySQL, et est distribué par l'entreprise américaine Automattic. Les fonctionnalités de WordPress lui permettent de créer et gérer différents types de sites Internet : blog, site e-commerce, site vitrine ou encore portfolio. Il est distribué selon les termes de la licence GNU GPL version 2. Le logiciel est aussi à l'origine du service WordPress.com.

W3C : Le World Wide Web Consortium, abrégé par le sigle W3C, est un organisme de standardisation à but non lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML5, HTML, XHTML, XML, RDF, SPARQL, CSS, XSL, PNG, SVG et SOAP.

JavaScript : JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.JS. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe.

CGI : La Common Gateway Interface (littéralement « Interface de passerelle commune »), généralement abrégée CGI, est une interface utilisée par les serveurs HTTP. Elle a été normalisée par la RFC 3875.

Index

Airplay.....	1, 2, 3, 6, 7, 18, 20, 21, 37, 38, 42, 73, 79, 85, 86, 113, 114, 117
analogiques	36, 101
Apache.....	8, 22, 23, 30, 31, 35, 113
Benchmarking.....	2, 13, 49, 50
Bluetooth.....	1, 2, 3, 6, 7, 8, 10, 12, 18, 20, 21, 37, 38, 39, 42, 45, 46, 61, 68, 73, 76, 77, 78, 86, 91, 92, 93, 94, 106, 113, 114, 115, 117
CAN.....	100, 101
CNA.....	100, 101
Courant.....	39
CPU	2, 12, 13, 14, 39, 53, 54, 115
CSS.....	1, 24, 25, 26, 27, 33, 42, 113
Debian	15
DLNA.....	6, 7, 8, 42
dnsmasq	2, 18, 79, 80, 84, 85, 117
Firefox.....	2, 33, 113
GStreamer	37
Hamachi.....	18
hostapd.....	2, 79, 80, 82, 83, 84, 85, 116, 117
HTML	1, 24
ISO	97
LED.....	61, 115
Linux	5, 11, 15, 18, 42, 118
MobaXTerm.....	2, 32
Mod_Python.....	1, 30
Nano	33, 46, 114
NOOBS.....	10, 15, 16, 43, 44, 57, 113, 114
Notepad++.....	2, 32, 113
Numérique.....	99, 100
PHP	1, 2, 18, 22, 23, 24, 25, 27, 28, 29, 30, 31, 33, 42
Php5.....	1, 18, 23
Puissance.....	39
Raspberry.....	1, 2, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 22, 23, 26, 27, 28, 29, 32, 34, 37, 39, 42, 43, 44, 45, 46, 47, 48, 49, 52, 54, 57, 60, 61, 62, 63, 68, 69, 70, 72, 79, 85, 113, 114, 115, 116, 118
RTP.....	3, 86, 89
RTSP.....	3, 86, 87, 88, 89, 117
shairport	31, 42, 78, 103
Shairport.....	1, 8, 18, 20, 39, 78
shell_exec	28, 31, 104, 106
SSH.....	17, 33, 69, 70, 85, 116, 119
sudoers.....	31, 35

Template.....	1, 25, 26, 27, 42, 113
Tension	39, 101
tightvncserver.....	18, 70
W3C	24, 25
Web	3, 5, 8, 22, 23, 24, 27, 28, 30, 39, 42, 103, 112, 117, 118
Wi-Fi	8, 10, 18, 20, 21, 34, 45, 46, 61, 68, 79, 82, 83, 85, 106, 114, 115, 116
Win32DiskImager	17, 57, 58
Windows.....	8, 15, 21, 32, 57, 70
Wireshark	18
www-data.....	1, 31, 35, 42
XML.....	24

Bibliographie & Sitographie

Airplay VS Bluetooth APT-X VS DLNA : Quelle Solution Propose La Meilleure Qualité Audio:

<http://blog.cobrason.com/2016/09/airplay-vs-bluetooth-apt-x-vs-dlna-quelle-solution-propose-la-meilleure-qualite-audio/>

Unofficial AirPlay Protocol Specification:

<https://nto.github.io/AirPlay.html#audio>

Audio Measurements using JAAA:

<http://kokkinizita.linuxaudio.org/papers/jaaa-pres.pdf>

Raspbian VS Debian: differences, analogies:

<https://blog.giuseppeurso.net/raspbian-vs-debian/>

Raspbian FAQ

http://www.raspbian.org/RaspbianFAQ#How_much_of_Debian_Wheezy_armhf_does_Raspbian_support.3F

Cross compilation avec un raspberry pi :

<http://oversimple.fr/cross-compilation-avec-un-raspberry-pi/>

ARMHF versus ARMEL

<http://www.xappsoftware.com/wordpress/2013/01/31/armhf-versus-armel/>

TP-Link TL-WN725N Nano Adaptateur USB Wi-Fi N 150 Mbps :

https://www.amazon.fr/TP-Link-TL-WN725N-Nano-Adaptateur-WiFi/dp/B008IFXQFU/ref=sr_1_3?ie=UTF8&qid=1488038472&sr=83&keywords=dongle+wifi+raspberry+pi

Samsung Carte Mémoire EVO Micro SD Classe 10 16 Go Avec Adaptateur SD :

https://www.amazon.fr/Samsung-Carte-M%C3%A9moire-ClasseAdaptateur/dp/B00J2BU7WO/ref=sr_1_2?ie=UTF8&qid=1488038459&sr=82&keywords=carte+sd+16+go+samsung

Raspberry PI Audio Receiver Install:

<https://bitbucket.org/adenbeckitt/raspberry-pi-audio-receiver-install/src/93032f81ec2a338a28473d1c067448fbb35318aa?at=master>

Raspberry Pi Bluetooth+Airplay Audio Receiver Combo:

<http://www.instructables.com/id/Raspberry-Pi-BluetoothAirplay-Audio-Receiver-combo/?ALLSTEPS>

A Comprehensive Raspberry Pi 3 Benchmark:

<https://www.element14.com/community/community/raspberry-pi/blog/2016/02/29/the-most-comprehensive-raspberry-pi-comparison-benchmark-ever>

Script to run some benchmark test for Raspberry Pi:

<https://github.com/aikoncwd/rpi-benchmark>

Shairport Sync

<https://github.com/mikebrady/shairport-sync>

iFun4U USB multimètre voltmètre ampère, tester ampère et voltage:

https://www.amazon.fr/iFun4U-multim%C3%A8tre-voltm%C3%A8tre-capacit%C3%A9-Chargeurs/dp/B01ISNFJB0/ref=sr_1_1?ie=UTF8&qid=1487614174&sr=81&keywords=usb%2Bpower%2Btester&th=1

Audio quality in networked systems

http://www.yamahaproaudio.com/europe/en_gb/training_support/selftraining/audio_quality/

Scopes & Realtime Visualizers:

<http://linux-sound.org/scopes.html>

Lancer un nouveau produit, élaborer le Plan Marketing (Partie 1) :

<http://www.conseilsmarketing.com/promotion-des-ventes/lancer-un-nouveau-produit-elaborer-le-plan-marketing-partie-1-2>

Compilation croisée pour Raspberry Pi

<http://tvaira.free.fr/projets/cdc-2016/activites/activite-raspberry-pi-cross-compilation.html>

Le Raspberry Pi 3 Chauffe et atteint presque les 100°C :

<http://www.minimachines.net/actu/raspberry-pi-3-atteint-presque-100c-38582>

Raspberry Pi 3 on sale now at \$35:

<https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/>

Raspberry Pi :

https://en.wikipedia.org/wiki/Raspberry_Pi

Debian vs Raspbian (Raspberry Pi 2):

<https://debian-facile.org/viewtopic.php?id=11559>

Qu'est-ce qu'un diagramme de Gantt:

<http://www.gantt.com/fr/>

Technical Considerations for Bluetooth:

<https://www.bluetooth.com/specifications/bluetooth-core-specification/technical-considerations>

Bluetooth Core Specification:

<https://www.bluetooth.com/specifications/bluetooth-core-specification>

Carte Mémoire Sandisk Micro SDHC, 16 GO + Adaptateur SD:

<http://www.fnac.com/Carte-Memoire-Sandisk-Micro-SDHC-16-GO-Adaptateur-SD/a7875037/w-4?omnsearchpos=1>

Analyse fonctionnelle (conception):

[https://fr.wikipedia.org/wiki/Analyse_fonctionnelle_\(conception\)](https://fr.wikipedia.org/wiki/Analyse_fonctionnelle_(conception))

Rédiger correctement un Cahier des Charges:

<https://openclassrooms.com/courses/rediger-correctement-un-cahier-des-charges>

Bluetooth vs AirPlay for Streaming Music:

<http://www.audiogurus.com/learn/speakers/bluetooth-vs-airplay/446>

Bluetooth vs AirPlay: What's The Difference:

<https://www.cambridgeaudio.com/fr/blog/difference-between-airplay-bluetooth>

Audio en BlueTooth avec le Raspberry Pi 3:

<http://www.framboise314.fr/audio-en-bluetooth-avec-le-raspberry-pi-3/>

Créez une carte SD Raspbian pour votre Raspberry Pi depuis Windows:

<https://raspbian-france.fr/creez-carte-sd-raspbian-raspberry-pi-windows/>

CSL - Adaptateur Wi-Fi 300 Mbps USB avec 2 antennes hautes performances (Dual Band):

<https://www.amazon.fr/CSL-Adaptateur-performances-Technologie-Articulation/dp/B00QT9VYX8>

Installer Raspbian, premier démarrage et configuration:

<https://raspbian-france.fr/installer-raspbian-premier-demarrage-configuration/>

Prenez la main à distance sur votre Raspberry Pi avec VNC:

<http://www.framboise314.fr/prenez-la-main-a-distance-sur-votre-raspberry-pi-avec-vnc/>

Transformer sa Raspberry Pi en borne Airplay:

<http://www.raspberrypi-france.fr/rasperry-pi-borne-airplay/>

Utiliser votre Raspberry Pi comme borne Airplay:

<http://blog.nicolargo.com/2013/03/utiliser-votre-raspberry-pi-comme-borne-airplay.html>

Enregistrer tout ce qui sort de la carte son:

<http://linuxmao.org/Capturer+les+sorties+audio>

Install Hamachi on a Raspberry Pi (Raspbian):

<https://www.igomez.net/install-hamachi-on-a-raspberry-pi-raspbian/>