



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique

MÉMOIRE DE MASTER

Sciences et Technologies
Electronique
Systèmes Emparqués

Réf. : Entrez la référence du document

Présenté et soutenu par :
Reffas Dounia

Le : mardi 2 juillet 2019

Système domotique avec Raspberry pi

Jury :

Pr.	Ben akcha abdelhamid	Pr	Université de Biskra	Président
Dr.	Fidias maraime	MAA	Université de Biskra	Examineur
Dr.	Bekhouche khaled	MCA	Université de Biskra	Rapporteur

Année universitaire : 2018 - 2019

RESUMES (Français et Arabe)

Notre projet vise à réaliser un Système Domotique relié à Internet ou à ce que l'on appelle Internet d'objet (IOT). Avec petite application Android contrôle les différentes parties de cette maison intelligente et carte magique Rosebery Pi . c'est le coeur de ce système . elle fonctionne comme un élément central responsable de l'intelligence et la prise de décision pour contrôler les périphériques de la maison connectée.

ملخص

هدف مشروعنا إلى تحقيق نظام أتمتة المنزل المتصل بالإنترنت أو ما يسمى بالإنترنت الأشياء (IOT). مع تطبيق أندرويد صغير يتحكم في الأجزاء المختلفة من هذا المنزل الذكي والبطاقة السحرية Raseberry Pi. هذا هو قلب هذا النظام. يعمل كعنصر مركزي مسؤول عن اتخاذ القرارات للسيطرة على أجهزة المنزل المتصل.

Dédicace

Ma mère

Mon père

Ma famille

Mes amis

Dounia Reffas

Remerciements

On remercie **ALLAH**, le tout puissant, qui nous a donné la force, la volonté et surtout le courage pour accomplir ce modeste mémoire.

Au terme de ce travail, on tient à remercier particulièrement *Dr. Bekhouche Khaled*, notre encadreur notre enseignant Maître de conférence à l'université Mohamed –khaidher- de Biskra pour son encadrement, Sa disponibilité, ses critiques et ses remarques pertinentes.

Il nous a conseillé et guidé du début du projet à sa fin.

J'exprime tous mes respects et remerciements à l'ensemble des membres de mon jury *Dr .BENAKCHA ABDELHAMID* et *DR. Fidias maraïme* et à Tous les remerciements et l'amour à tous mes ami(e) et à tous les personnes qui m'ont aidée dans ce travail .Ces paroles simples ne leur expriment certainement pas ma gratitude et ma reconnaissance.

Enfin, les mots les plus simples étant les plus forts, j'adresse toute mon affection à ma famille, et en particulier à ma maman et mon père. Merci pour avoir fait de moi ce que je suis aujourd'hui et de m'avoir fait croire à mes capacités. . J'adresse toute ma gratitude à tous mes ami(e)

Table des matières

Dédicace	
Remerciements	
Table des matières	
Liste des figures	3
Liste des tableaux	
Liste des abréviations	
Introduction générale	

1. Raspberry Pi

1.1. Introduction	14
1.2. L'histoire de raspberry pi	14
1.3. Raspberry Pi	14
1.4. Les modèles de raspberry	15
1.5. La comparaison entre les modèles de raspberry pi	17
1.6. Le raspberry pi modèle B 3	18
1.6.1. Les composants de Raspberry Pi	18
1.6.2. Les ports GPIO de la raspberry pi 3	19
1.7. La configuration de raspberry pi	20
1.7.1. Matériel nécessaire	20
1.7.2. L'installation de système d'exploitation sur carte SD	20
1.7.3. L'interface de configuration raspberry pi	21
1.8. Ajouter d'autres fonctions	22
1.8.1. La connexion de raspberry pi avec pc	22
1.9. Langage de programmation	24
1.9.1. Présentation du python	24
1.10. Conclusion	25

2. La domotique et l'internet des objets (IOT)

2.1. introduction	27
2.2. l'internet des objets (IOT)	27

2.2.1.	L'objectif de l'internet d'objet	28
2.2.2.	Fonctionnement d'internet d'objet	28
	• Les capteur	
	• Connectivité	
	• T raitement de l' information	
	• L' interface utilisateur	
2.2.3.	Les applications d ' internet d'objet (iot)	31
2.3.	La domotique	
2.3.1.	Diffintion	35
2.3.2.	Principe de fonctionnement de maisson connectée	35
2.3.3.	Les fonctions domotique de la maison intelegent	36
2.4.	Future de l'IoT	38
2.5.	Conclusion	39
3.	Realisation du système domotique	
3.1.	Introduction	41
3.2.	Representation de système réalisé	41
3.3.	Partie matrielle de projet	42
3.3.1.	La configuration de composant avec raspberry pi	43
3.4.	Partie logiciel de projet	44
3.4.1.	Conception de application android	45
3.4.1.1.	Android Studio	45
3.4.1.2.	Démarrer Android Studio	45
3.4.1.3.	Contenu du projet	50
3.4.1.4.	L_ AVD Android	51
3.4.2.	La création de mon application	52
3.4.3.	Appliation java du client et du serveur	53
3.5.	Conclusion	58

Liste de figure

Fig 1.1 : représentation d' un Raspberry pi stadard.....	14
Fig 1.2 : Modèle A	15
Fig 1.3 : Modèle A+.....	15
Fig 1.4 : Modèle B1.....	16
Fig 1.5 : Modèle B1+.....	16
Fig 1.6 : Modèle B2.....	16
Fig 1.7 : Modèle B3.....	16
<i>Fig 1.8 : Modèle 0.....</i>	<i>17</i>
Fig 1.9 : représente raspberry pi 3 modèle B.	18
Fig 1.10 : <i>représente GPIO raspberry pi 3 modèle</i>	<i>19</i>
<i>Fig 1.11 : système d'exploitation Raspbian.....</i>	<i>20</i>
Fig 1.12 : l' interface de Raspberry Pi 3	21
Fig 1.13 : configuration de Raspberry Pi 3	22
<i>Fig 1.14 :connection enrtre raspberry et pc</i>	<i>22</i>
<i>Fig 1.15 :configuration Putty.....</i>	<i>24</i>
<i>Fig 1.16 : Python Shell et son editeur de texte.....</i>	<i>25</i>
Fig 2.1 : internet d objet	28
Fig 2.2 : representation la fonctionnement IoT	29
Fig 2.3 : maison connecté	34
Fig 2.4 : ensemble des systèmes d'un maison <i>intelligent</i>	35
Fig 2.5 : principe de fonctionnement de maison intelligent	36

<i>Fig 2.6</i> : différente fonction d'maison intelligent	38
<i>Fig 3.1</i> : Schéma général du système.....	41
<i>Fig 3.2</i> : Organigramme de la communication entre l'application utilisateur Raspberry.....	42
<i>Fig 3.3</i> : Programme python d'une LED avec raspberry pi 3.....	43
<i>Fig 3.4</i> : Programme python d'un servomotoravec raspberry pi 3.....	44
<i>Fig 3.5</i> : Programme python d'un dh11 avec raspberry pi 3.....	44
<i>Fig 3.6</i> : Icone de l'Android studio.....	45
<i>Fig 3.7</i> : 1 ^{er} Démarrage de l'Android Studio.....	45
<i>Fig 3.8</i> :Nouvel projet.....	46
<i>Fig 3.9</i> :Selection de la version de l'Android.....	47
<i>Fig 3.10</i> :Liste des API.....	47
<i>Fig 3.11</i> : Empty Activity.....	48
<i>Fig 3.12</i> : Le nom de Empty Activity.....	48
<i>Fig 3.13</i> : Ecran global d'un nouveau projet MainActivity.java.....	49
<i>Fig 3.14</i> : Ecran global d'un nouveau projet Activity_main.xml.....	49
<i>Fig 3.15</i> : Structure du projet dans Android Studio.....	49
<i>Fig 3.16</i> : Android Virtual Device.....	50
<i>Fig 3.17</i> : application réalisé.....	52
<i>Fig 3.18</i> :Programme socket server python.....	47

Liste des tableaux

Tableaux 1.2 : COMPARISON ENTRE LES MODELES RASPBERRY PI

Tableaux 3.1 : Représentation des sous –dossier de ressources .

Liste des abréviations

GPIO	General purpose Input/Output
WIFI	wireless Fidelity
GPU	Graphics Processing Unit
CPU	Central Processing Unit
HDMI	High Definition Multimedia Interface
LCD	Liquid Crystal Display
I2C	Inter Integrated Circuit Bus
SPI	<u>Serial Peripheral Interface</u>
PWM	Pulse Width Modulation
SSH	Serveur Secure Shell
IOT	Internet of Things
IdO	Internet of Things
RFID	Radio Frequency Identification
LPWAN	Low-Power Wide-Area Network
LED	Light-Emitting Diode
USB	Universal Serial Bus
IP	Internet Protocol

Introduction générale

Dans le cadre de notre projet de fin d'études, nous avons opté pour la domotique, et cela en réalisant une maison intelligente avec raspberry pi .

Tout comme notre vie en général, nos maisons se retrouvent de plus en plus technologisées. Les habitats du futur répondent à une probable insatisfaction innée de l'homme qui croit augmenter son pouvoir sur les choses par la technique. On voit donc que sa maison «s'adapte» à lui et à ses besoins. Au même titre que nous gérons notre budget, organisons notre temps et nos activités, que nous profitons des équipements perfectionnés de notre voiture (fermeture centralisée, Climatisation...), l'évolution de la technologie et du mode de vie nous permet aujourd'hui de prévoir des espaces de travail et de logement mieux adaptés. De même, La majorité des individus, et plus particulièrement les personnes âgées, passent beaucoup de leur temps à domicile, d'où l'influence considérable de l'habitat sur la qualité de vie. L'amélioration du sentiment de sécurité et de confort dans l'habitat apparaît donc comme une tâche d'une grande importance sociale.

Le concept de domotique fait l'objet de nombreuses définitions. De la maison intelligente à la maison communicante, toutes les gradations sont possibles. De fait, la domotique concerne l'application à l'habitat de toutes les technologies dites nouvelles pouvant s'intégrer à ce contexte. Des auteurs tels que Brun (1988 : 3) ou Jarrosson (1996 : 3) nous présentent la définition la plus générale, qui s'exprime sous la forme : « Ensemble des services offerts aux occupants d'un logement fondés sur l'échange d'informations et permettant d'accéder à un nouvel « art de vivre.»

Ce mémoire est organisé en trois chapitres. D'abord, au premier chapitre, on va commencer avec une présentation de la carte raspberry pi vu ses performances techniques intéressantes. Le second chapitre on présente d'une façon générale la domotique et l'internet d'objet. Finalement, nous détaillerons les phases de réalisation du système domotique que ce soit dans sa partie matérielle ou logicielle.

CHAPITRE 01 :

RASPBERRY PI

1.1.Introduction :

Dans ce chapitre, on va présenter la carte électronique nano-ordinateur la Raspberry Pi comme un module de notre projet de domotique, On va voir c'est quoi la carte Raspberry Pi, ses composants, ses fonctionnalités aussi on parle du langage python son importance et sa simplicité pour contrôler les GPIO du Raspberry . .

1.2.L'histoire du raspberry pi :

Un petit retour sur l'histoire du Raspberry Pi (sur nommé RasPi) et sur sa cible du marché initiale permet d'en comprendre les contraintes et les limitations. Le concept de RasPi a été dévoilé autour de 2006 par Eben Upton et ses collègues de laboratoire d'informatique de l'université de Cambridge, en Angleterre. Ils étaient préoccupés par la baisse de niveau de connaissances et de compétences des nouveaux étudiants en informatique en comparaison de leurs aînés. Eben Upton a alors décidé de créer un ordinateur à faible prix, car il supposait que les parents craignaient de laisser leurs enfants jouer avec des PC modernes relativement coûteux. C'est de cette idée qu'est parti le développement du RasPi très bon marché. Cet ordinateur devait donner aux jeunes l'opportunité d'apprendre et de pratiquer la programmation, sans que leurs parents ne s'inquiètent des dégâts éventuels sur la machine [1].

1.3.Raspberry Pi :

Le Raspberry Pi est un ordinateur à mono-carte avec un processeur ARM conçu par le créateur de jeux vidéo David Braben . dans le cadre de sa fondation Raspberry Pi [2]. Cet ordinateur, qui a la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique ; il permet l'exécution de plusieurs variantes du système d'exploitation libre **GNU/Linux** et des logiciels compatibles [3].

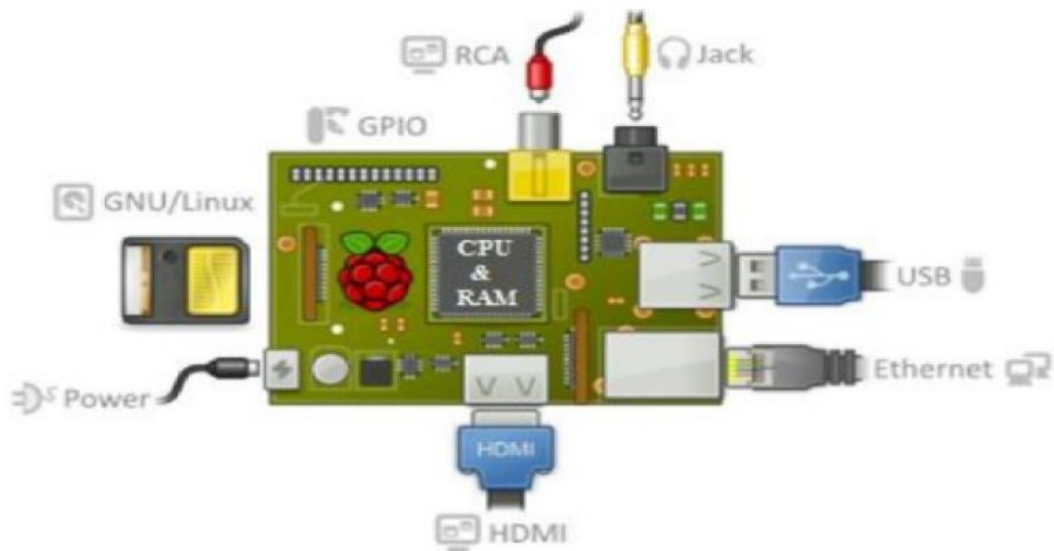


Fig 1.1 : représentation d' un Raspberry pi standard .

1.4. Les modèles de Raspberry Pi :

Plusieurs générations de Raspberry Pi : le 0, le 1 , le 2, le 3 et le 3B

- **Modèles A** : il y a deux types A et A+.

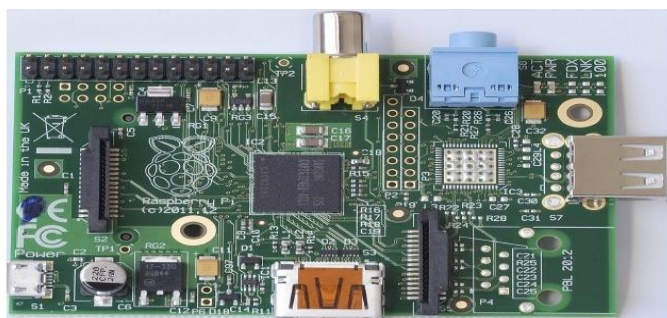


Fig 1.2 : Modèle A .



Fig 1.3 : Modèle A+ .

Modèles B :

il y a 4 types B1 et B1+, B2, B3 :



Fig 1. 4 : Modèle B1.



Fig 1.5 : Modèle B1+.



Fig1. 6 : Modèle B 2.



Fig 1.7 : Modèle B3.

● **Modèle 0 :**

La figure 8 représente le modèle 0.

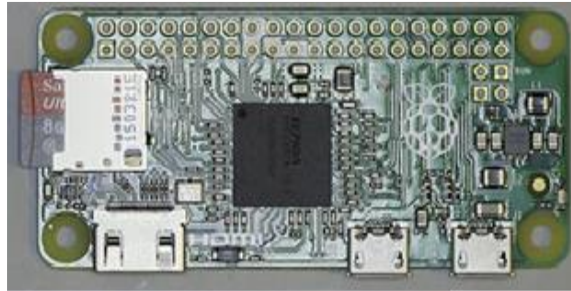


Fig 1.8 : Modèle 0.

1.5. Comparaison entre les modèles de raspberry pi :

Le tableau 1.1 présente les trois modèles principaux de Raspebery ainsi que leur caractéristiques et performances.

	<u>Raspberry Pi 3 Model B</u>	<u>Raspberry Pi 2 Model B</u>	<u>Raspberry Pi Model A</u>
Processor Chipset	Broadcom BCM2837 64Bit Quad Core ARM Cortex A53 at 1.2GHz	Broadcom BCM2836 32Bit Quad Core ARMv7 at 900MHz	Broadcom BCM2835 32Bit ARMv6k at 700MHz
GPU	Videocore IV @ 400MHz	Videocore IV @ 250MHz	Videocore IV @ 250MHz
Processor Speed	QUAD Core @1.2 GHz	QUAD Core @900 MHz	Single Core @700 MHz
RAM	1GB SDRAM @ 400 MHz	1GB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz
Storage	MicroSD	MicroSD	MicroSD
USB 2.0	4x USB Ports	4x USB Ports	4x USB Ports
Max Power Draw/voltage	2.5A @ 5V	1.8A @ 5V	1.8A @ 5V
GPIO	40 pin	40 pin	40 pin
Ethernet Port	Yes	Yes	Yes
WiFi	Built in	No	No
Bluetooth LE	Built in	No	No
Video Output	HDMI/Composite via RCA Jack	HDMI/Composite via RCA Jack	HDMI/Composite via RCA Jack
Audio Output	3.5mm Jack	3.5mm Jack	3.5mm Jack

Tableaux 1.2 : la comparaison entre les modèles de raspberry pi.

1.6. Le Raspberry Pi Modèle B 3 :

Pour notre projet, nous avons opté pour *Raspberry Pi B3*. Comparé au Pi 2, il dispose d'un processeur Broadcom BCM2837 64 bit à quatre cœurs ARM Cortex-A53 à 1,2 GHz, d'une puce Wifi 802.11n et Bluetooth 4.1 intégrée [4].

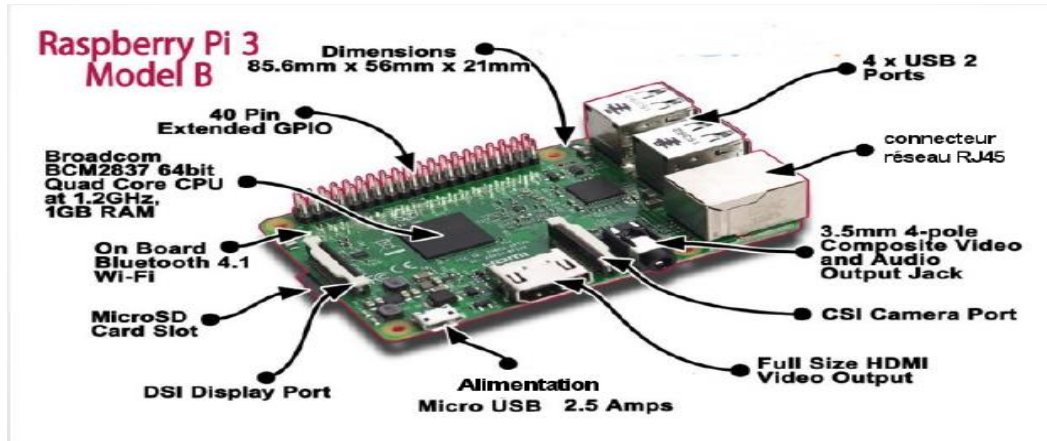


Fig1.9 : représente raspberry pi 3 modèle B.

1.6.1. Les composent Raspberry Pi 3 Type B :

- Carte mère **Raspberry Pi 3 Type B**
- Processeur intégré Quad-core ARM Cortex-A53 1.2 GHz (Broadcom BCM2837)
- RAM : 1024 Mo
- GPU Dual Core VideoCore IV Multimédia Co-Processor
- Lecteur de cartes Micro SD
- Ports : HDMI, 4x USB, RJ45, jack 3.5 mm, connecteurs pour APN et écran tactile
- Wi-Fi b/g/n et Bluetooth 4.1
- Support des distributions dédiées basées sur Linux et Windows 10 .

1.6.2. Le port GPIO Raspberry Pi 3 Model B :

Le Raspberry Pi possède, en plus des connectiques classiques USB, HDMI, etc... un connecteur **GPIO**. **GPIO** signifie en anglais « **General Purpose Input Output** » et pourrait être

traduit en français par entrées/sorties numériques. Ces entrées/sorties permettent d'étendre les fonctionnalités du Raspberry Pi en lui donnant la possibilité d'agir sur des LEDs ou des afficheurs LCD par exemple, lire l'état d'un interrupteur, d'un capteur, etc...

Ce connecteur **GPIO** dispose de différents types de connexion:

- Des broches utilisables en entrée ou sortie numérique tout ou rien.
- Des broches pour une interface I2C.
- Une interface SPI (le module NRF24L01 est connecté à cette interface)
- Les broches Rx et Tx pour la communication UART avec les périphériques séries.
- Des broches pouvant être utilisé en PWM ("Pulse Width Modulation") permettant le contrôle de puissance ou PPM ("Pulse Position Modulation") permettant de contrôler des servo moteurs par exemple [5]. .

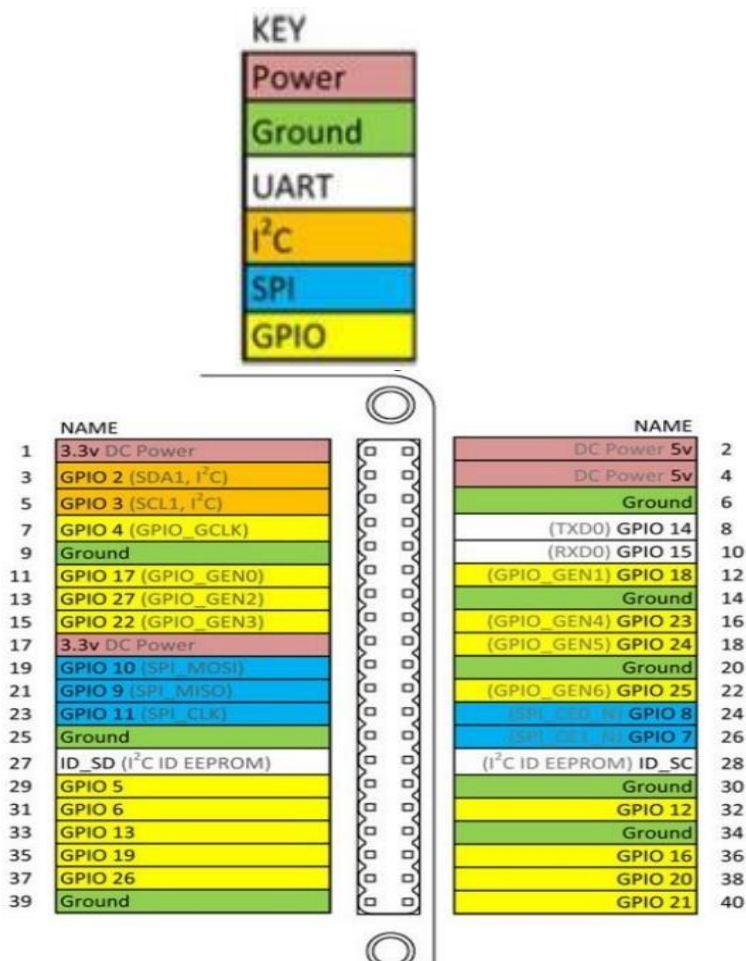


Fig 1.10 : GPIO raspberry pi 3 modèle B

1.7. La configuration de raspberry pi :

1.7.1. Matériel nécessaire

- Un Raspberry Pi 3 avec son boîtier et son alimentation
- Une carte micro SD (16 Go minimum) et un adaptateur micro-SD / USB ou bien micro-SD / SD Card pour la brancher sur son ordinateur
- Un clavier USB et une souris USB
- Un câble HDMI et un écran (avec entrée HDMI)
- Une connexion (WiFi ou filaire RJ45)

Un PC (sous Linux de préférence ; tout à fait faisable sous Windows, cependant) .

1.7.2. Installation du système d'exploitation sur la carte SD :

Les Raspberry Pi sont livrés par défaut **vierges de tout système d'exploitation** et de tout dispositif de stockage.

C'est à l'utilisateur de choisir et d'installer le système d'exploitation qu'il veut (la plus grande majorité sont des systèmes Linux) et de choisir la carte de stockage de type micro-SD avec la capacité qu'il veut (16 Go minimum conseillés) .

Il existe de nombreux systèmes d'exploitation dédiés au Raspberry Pi. Dans ce petit guide, on se contentera d'utiliser un système d'exploitation « *standard* », c'est à dire sans fonctionnalités orientées pour une utilisation spécifique. Dans cette optique, **Raspbian** est particulièrement indiqué. Il s'agit d'un système d'exploitation **GNU/Linux**, spécialement conçu et optimisé pour les Raspberry Pi. Sa version actuelle est basée sur la dernière version de la distribution Debian [6].

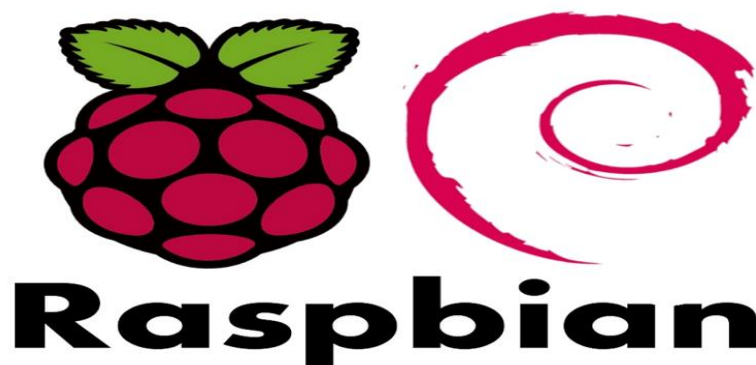


Fig 1.11 : système d'exploitation Raspbian.

1.7.3. **l'interface de configuration du Raspberry Pi :**

Au premier démarrage, vous allez avoir droit au lancement automatique de l'utilitaire de configuration Raspbi-config qu' il est possible de rappeler par la suite en saisissant la commande suivante dans un terminal : **sudo raspi-config** Raspbi-config se présente dans la figure suivante:

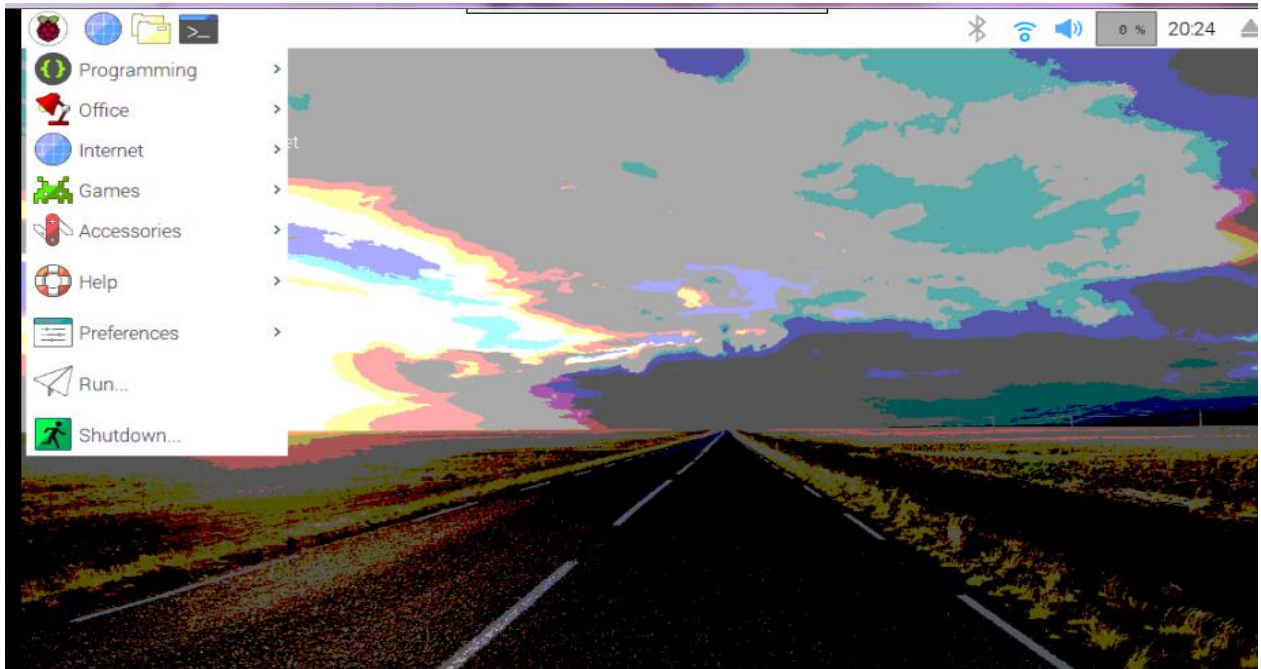


Fig 1.12 : l' interface de Raspberry Pi 3

- Les différentes fonctions permettent de :
- **expand_rootfs** : étendre la partition principale à la taille maximale de la carte SD (par exemple si vous avez une carte SD de 16 alors vous resterez avec une partition principale de 2 GO
- **overscan** : force l'affichage de marge pour s'adapter à des écrans 16/9 ou supérieur
- **configurer** : le clavier (de l'utilisateur par défaut (pi) (et l temps(change_timezone)
- **overclocking**: comme Raspberry est une société intelligente, elle autorise l'overclocking de son matériel sans perte de la garanti matériel qui risque de diminuer). Fonction à activer si vous voulez utiliser le Raspberry en mode graphique (comme client léger) ou tout autre besoin consommateur de CPU.
- **Memory-split** : pour définir la quantité de mémoire vive dédiée au GPU (64 Mo par défaut) .

- **ssh** : permet de définir si le serveur SSH doit être lancé par défaut .
- **boot_behavior** : choisir si on veut démarrer le boîtier en mode texte (utilisation serveur) ou graphique (client léger avec l'environnement OpenBox + LXDE) .
- **Update** : qui est en fait un raccourci vers la commande **apt-get update && apt-get upgrade** et ainsi mettre à jour votre système (qui doit être connecté à Internet) [7].

```

Raspi-config
info          Information about this tool
expand_rootfs  Expand root partition to fill SD card
overscan      Change overscan
configure keyboard Set keyboard layout
change_pass    Change password for 'pi' user
change_locale  Set locale
change_timezone Set timezone
memory_split   Change memory split
overclock     Configure overclocking
ssh           Enable or disable ssh server
boot_behaviour Start desktop on boot?
update        Try to upgrade raspi-config

                <Select>                <Finish>

```

Fig 1.13 : configuration Raspberry Pi 3 .

1.8.Ajouter d' autres fonctions :

1.8.1. La connection de raspberry pi avec pc :

L'utilisation de Raspberry Pi en SSH sans réseau Wifi et possible. On va donc simplement utiliser un câble Ethernet RJ45 entre le PC et le Raspberry Pi.

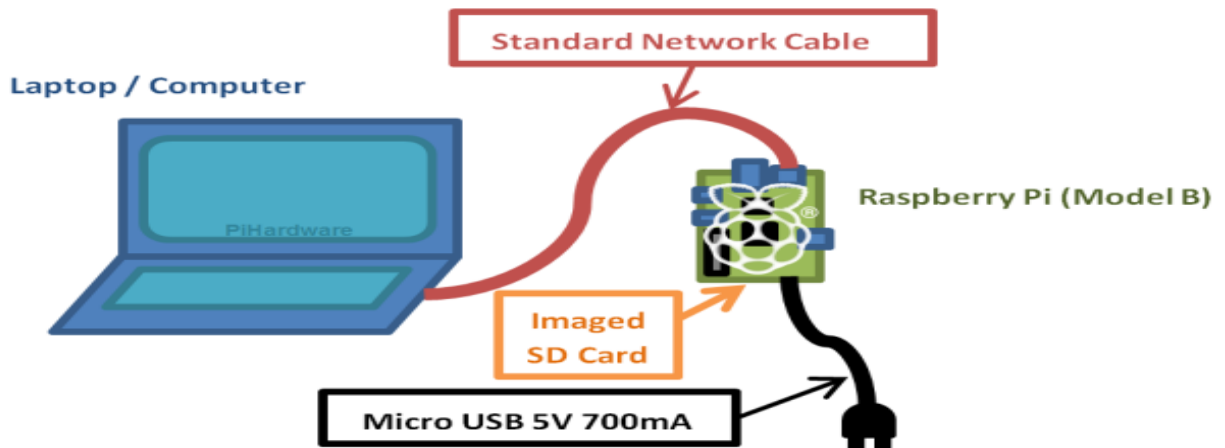


Fig 1.14 : La connection de raspberry pi avec pc .

Pour se connecter au Raspberry en Ethernet depuis son ordinateur, il faut d'abord attribuer une adresse IP fixe au Raspberry. Pour ce faire, récupérez la carte SD du Raspberry et lisez-la avec l'explorateur de fichiers Windows ou MAC. (La carte est normalement appelée 'boot'.) Cherchez le fichier "*cmdline.txt*", c'est lui que l'on va modifier. Vous pouvez faire une sauvegarde du fichier avant de le modifier pour plus de sécurité. Ouvrez donc le fichier avec votre éditeur de texte favori et ajoutez à la fin de la ligne (*pas de saut de ligne*) le texte suivant :

ip=169.254.x.x (x à choisir entre 169.254.0.0 et 169.254.255.255) Exemple :169.254.25.25

Compléter les 'x' avec des chiffres de votre choix pour avoir votre IP. Notez l'IP choisie quelque part. ;Une fois terminé, enregistrez le fichier (*ctrl S*) et retirez la carte. Ré-insérez-la dans le Raspberry Pi.

- Une fois la carte configurée, on peut brancher le Raspberry au PC/Mac à l'aide des ports Ethernet de chacun et alimenter le Raspberry au 5V. On attend 1-2 minutes le temps que le Raspberry s'initialise. Une fois le Raspberry initialisé, on peut vérifier que tout marche en ouvrant l'invite de commande Windows ou le Terminal sur mac en tapant :

ping 169.254.25.25 (en suivant l'exemple utilisé plus haut)

S'il n'y a pas d'erreur, le Raspberry est prêt à être connecté en SSH par Ethernet. A noter qu'il est toujours possible d'accéder à Internet sur l'ordinateur par le Wifi.

- Une fois le tout branché, on peut se connecter en SSH au Raspberry. Ouvrez votre logiciel Une fois coché, on retourne dans *Session*, on tape l'adresse IP de notre Raspberry que vous avez dû noter dans *Host Name*

Il ne reste plus qu'à se connecter (Par défaut : login = *pi* et password = *raspberry* et on est connecté au Raspberry, lui-même connecté en Ethernet au PC ou au Mac, en SSH. SSH, allez dans *Connection-/- SSH-/-X11* et cochez *Enable X11 forwarding*[8].

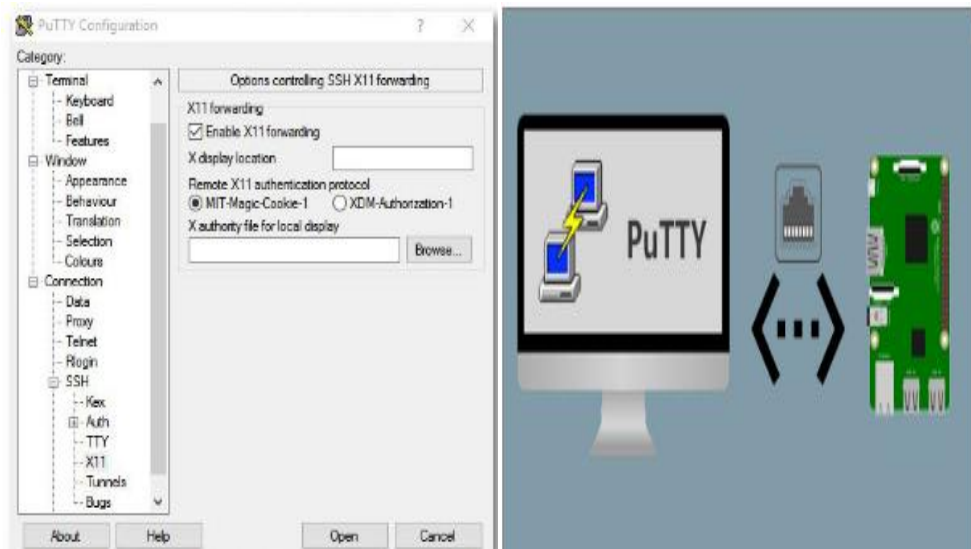


Fig 1.15 : configuration Putty .

1.9. Langage de programmation :

L'avantage d'un système d'exploitation comme GNU/Linux sur une carte embarquée Raspberry Pi est de disposer de **beaucoup de logiciels compatibles** à utiliser dès l'installation.

Et grâce à l'espace disque important (plusieurs Go), on peut encore ajouter des possibilités de programmation avec **tous les langages disponibles sur un ordinateur classique** en pouvant facilement passer de l'un à l'autre selon le projet[9], mais le Python est le langage le plus facile et la plus évolue .

1.9.1. Présentation du python:

Python est un langage de script de haut niveau, structuré et open source. Il est [multi-paradigme](#) et multi-usage.

Développé à l'origine par [Guido van Rossum](#) en 1989, il est, comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde [10].

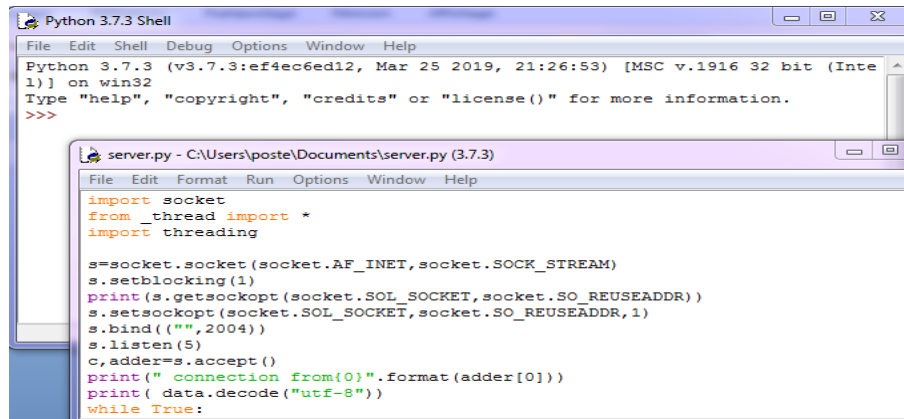


Fig 1.16 :Python Shell et son editeur de texte.

Parmi les autres raisons, citons la rapidité de développement (qualité propre aux langages interprétés), la grande quantité de modules fournis dans la distribution de base ainsi que le nombre d'interfaces disponibles avec des bibliothèques écrites en **C**, **C++** ou **Fortran**. Il est également apprécié pour la clarté de sa syntaxe.

1.10. Conclusion :

Dans ce chapitre dédié à la présentation du Raspberry Pi, nous avons présenté les composants de cette carte, son utilité spécialement pour notre projet sur la domotique, ensuite on a vu comment configurer cette carte et comment utiliser les broches pour contrôler les composants électroniques. Le langage de programmation Python est brièvement présenté dans ce chapitre.

CHAPITRE02 :

LA DOMOTIQUE ET
L'INTERNET D'OBJET

2. La domotique et L internet d'Objet (IOT) :

2.1. Introduction :

Bien que les objets connectés soient à la mode, pour beaucoup de gens cela reste encore une idée très vague, tout comme la domotique, qui véhicule avec elle depuis très longtemps la mauvaise réputation d'être chère et compliquée à mettre en place [11]. L'internet des objets consiste de manière simplifiée à connecter des objets. En quelque sorte il s'agit de l'extension de l'Internet au monde réel des objets qui nous entoure [12].

L'Internet des objets (IoT) se traduit à l'heure actuelle par l'accroissement d nombre d'objets connectés, c'est-à-dire d'appareils possédant une identité propre et des capacités de calcul et de communication de plus en plus sophistiquées : téléphones, montres, appareils ménagers, etc. Ces objets embarquent un nombre grandissant de capteurs et d'actionneurs leur permettant de mesurer l'environnement et d'agir, faisant ainsi le lien entre le monde physique et le monde virtuel [13].

2.2. L internet des objets (IoT) :

Le terme Internet of Things désigne généralement des scénarios où le réseau la connectivité et la capacité de calcul s'étendent aux objets [14], je dirai qu'un objet connecté est un objet électronique relié à Internet et capable de communiquer des informations, apportant ainsi un service ou une valeur ajoutée[15].

Notre terme, L'internet multidimensionnel est un prolongement de l'**IdO** qui englobe le système plus complexe constitué de la communication de machine à machine, de personnes et de processus. Autrement dit, l'Internet multidimensionnel est un réseau composé de personnes, de données, de processus et d'appareils [16].

L'Internet des Objets est un réseau de réseaux qui permet, via des systèmes d'identification électronique normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant[17]



Fig 2.1 : internet d objet.

2.2.1. L'objectif de l' internet d'objet :

L'objectif ambitieux derrière l'Internet des Objets :

- les différentes définitions présentées et la littérature que nous avons étudiée
- de stocker et de rendre accessible l'identité des objets, les informations produites par ces derniers et toutes les connaissances nécessaires pour que les objets gagnent en autonomie, ce au travers de représentations, de structures et de formats de données manipulables par les machines ;
- offrir les abstractions nécessaires aux êtres humains pour interagir avec ces machines, et par extension avec le monde physique, aussi simplement qu'avec le monde virtuel que nous connaissons aujourd'hui [18].

2.2.2. Fonctionnement d' internet d' objet :

Vous trouverez une explication simple de comment fonctionne un objet connecté [19].

Le système complet d'un objet connecté intègre quatre composants distincts :

- Des capteurs
- Une connectivité
- Un traitement de l'information
- Une interface utilisateur

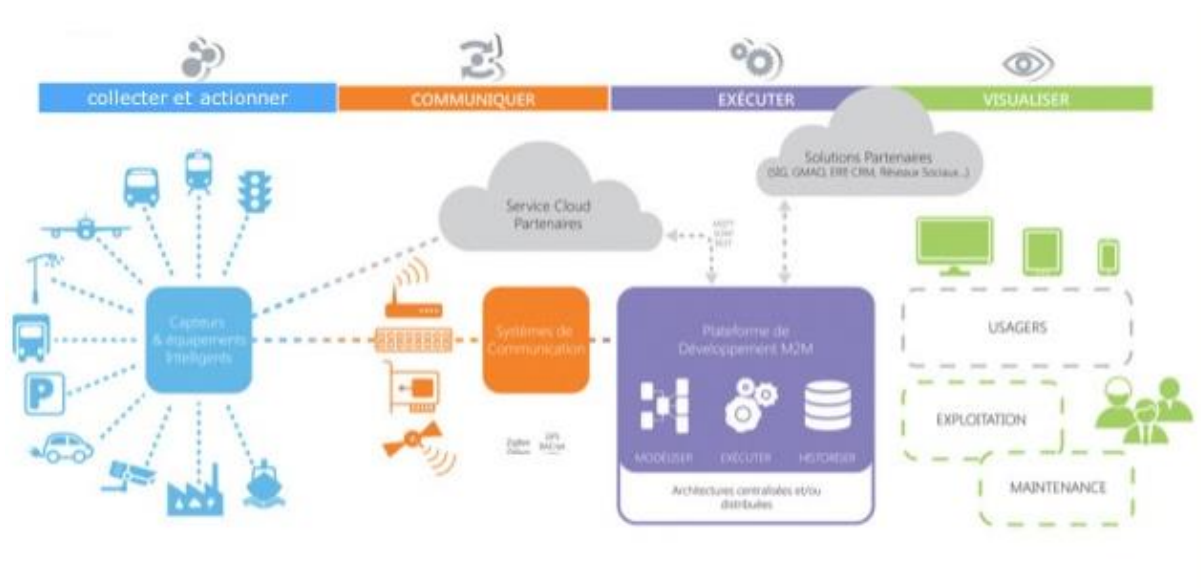


Fig 2.2 : représentation du fonctionnement ITO.

On va décrire brièvement ci-dessous les composants et leurs interactions.

- **Les capteurs**

Tout d'abord, **les capteurs récoltent les données de leur environnement**. Cela peut être une simple lecture de niveau de bruit comme une lecture complète d'un flux photo ou vidéo.

On utilise des capteurs, car ils peuvent souvent être regroupés ou faire partie d'un device qui fait beaucoup plus que juste sentir son environnement. Par exemple, votre téléphone est un appareil qui possède plusieurs capteurs : caméra, accéléromètre, GPS, etc...), mais votre téléphone n'est pas juste un capteur. Dans tous les cas, qu'il s'agisse d'un capteur autonome ou un appareil, la première étape est toujours la même : *quelque chose* capte les informations de son environnement.

- **Connectivité**

L'information est envoyée dans le cloud, mais elle a besoin d'un moyen d'y parvenir.

Les capteurs/appareils peuvent être connectés au cloud au travers d'une variété de méthodes incluant : réseau cellulaire, satellites, WIFI, Bluetooth, les réseaux LPWAN (Low-Power Wide-Area Network) ou une connectivité directe à internet via Ethernet.

Chaque option possède son lot de contrainte que ce soit en consommation d'énergie ou en bande passante. Choisir la meilleure connectivité dépend des fonctionnalités et de l'application de l'objet connecté. Au final, toutes les options offrent le même résultat : **envoyer les informations dans le cloud**.

- *Traitement de l'information*

Une fois l'information dans le cloud, des **logiciels réalisent un traitement** sur ces dernières.

Le traitement peut être très simple, comme vérifier si la température lu est comprise dans une fourchette acceptable définie précédemment. Il se peut que le traitement soit également très complexe, comme identifier un objet particulier au travers d'une vidéo lue par un ordinateur. Cette application est utilisée pour identifier des intrus dans une maison.

- *Interface utilisateur*

L'information est tout d'abord rendue utile et lisible pour l'utilisateur final. Elle peut prendre vie sous forme d'alerte à l'utilisateur (email, sms, notification, etc...). Un SMS est envoyé à l'utilisateur lorsqu'un intrus est détecté dans la maison, par exemple.

Aussi, les utilisateurs disposent souvent d'une interface qui leur permet de vérifier de manière proactive dans le système. Par exemple, notre utilisateur va pouvoir voir un flux vidéo de l'intérieur de sa maison sur une application mobile ou une page web.

En fonction du champ d'application de l'objet connecté, le flux d'information n'est pas toujours un envoi à sens unique. L'utilisateur peut aussi être en mesure de déclencher une action sur le système. Par exemple, l'utilisateur peut ajuster la température de la chambre froide lorsqu'il décèle une vague de chaleur grâce aux capteurs, directement depuis son téléphone.

D'autres actions sont déclenchées automatiquement. Ainsi, l'utilisateur peut sur certains systèmes **créer des scripts** qui se déclencheront grâce à des règles prédéfinies. Ainsi, au lieu de vous alerter directement, le système de l'objet connecté pourra alerter directement les autorités compétentes.

Un système d'objet connecté est composé de capteurs/d'appareils qui “*dialogue*” avec le cloud au travers d'une certaine connectivité. Une fois l'information relayée sur le cloud, des logiciels la transforment et la traitent et peuvent éventuellement décider d'agir en conséquence du résultat obtenu.

Si une interaction utilisateur est requise ou s'il veut simplement se connecter au système, une interface utilisateur lui permet d'agir en fonction. Tous les ajustements ou actions qu'un utilisateur fait sur son interface sont ensuite envoyés dans la direction opposée via le système : depuis l'interface utilisateur, vers le cloud, et finalement vers les capteurs/appareils pour effectuer un changement.

2.2.3. L'application d'internet d'objet :

On n'en entendait à peine parler il y a quelques années, et ils sont maintenant partout. Les objets connectés ont envahi notre quotidien sans même que nous y prêtions attention.

De la télé intelligente à la voiture connectée, nos loisirs, nos déplacements sont facilités par ces nouveaux outils qui augmentent grandement notre confort [20].

2.2.3.1. L'IoT dans le domaine de la santé :

Machines à rayons X et imagerie, moniteurs connectés, compteurs d'énergie... 60 % des hôpitaux mondiaux utilisent déjà l'Internet des Objets pour augmenter leur productivité et améliorer les soins apportés aux patients. L'étude d'Arubanetworks montre que d'ici 2019, c'est presque 90 % des services de santé qui auront intégré les objets connectés dans leur matériel médical. Les objets connectés sont utilisés au quotidien pour :

- La surveillance au sein des établissements médicaux et la maintenance
- Les opérations chirurgicales et le contrôle à distance
- Les services de géolocalisation

La normalisation de l'Internet des Objets dans le domaine de la Santé va permettre de créer de nouveaux modèles de fonctionnement qui augmenteront la productivité des employés, mais aussi la collaboration entre soignants ainsi que la communication avec les patients.

2.2.3.2. La révolution numérique en réponse aux impératifs

énergétiques : est une véritable plus-value dans le domaine énergétique, qui pourrait dans les années à venir représenter un investissement écologique décisif dans l'avenir de notre société. Les enjeux sont également économiques, et les entreprises l'ont bien compris.

L'IOT, dans le cadre de l'énergie, répond à des problématiques majeures :

- appauvrissement des ressources naturelles
- accroissement des besoins énergétiques à l'échelle mondiale
- instabilité des prix du marché
- manque de main d'œuvre humaine

La révolution numérique est entrée dans le débat des secteurs énergétiques par la gestion des ressources : compteurs énergétiques, réseau intelligent mais aussi l'Internet des Objets à domicile, avec la maison intelligente.

2.2.3.3. L'industrie connectée :

L'industrie n'est pas en reste sur l'usage de l'Internet des Objets et des bénéfices que celui-ci lui apporte. Dans le cadre des problématiques rencontrées dans le domaine industriel, l'usage des objets connectés est très spécifique et répond à des besoins :

- d'optimisation (chaîne logistique)
- de transformation des processus d'entreprise
- d'amélioration de l'efficacité et de la productivité
- de traçabilité et de sécurité

La révolution digitale est aussi l'opportunité pour certains types d'industrie de se renouveler et d'apporter une plus-value sur un terrain en perte de popularité. C'est par exemple le cas avec [SNCF Fret](#), qui retrouve doucement ses lettres de noblesse en lançant sa locomotive connectée, qui permet une meilleure traçabilité de ses wagons, et une plus grande sécurité pour le client.

2.2.3.4. L'Internet des objets dans l'agriculture :

La croissance rapide de la population mondiale, les changements d'habitudes alimentaires, les perturbations climatiques sont trois grands facteurs, parmi d'autres, qui font de l'agriculture moderne un défi au quotidien. D'ici 2050, la productivité agricole devra avoir augmenté de 70 % pour répondre à la demande mondiale. Plus qu'un défi technologique, il s'agit d'un enjeu humanitaire. Les céréaliers et maraîchers ont d'ors et déjà mis à profit les drones afin de récolter en temps réel des informations essentielles à la gestion de l'exploitation :

- humidité de la terre
- état des plantations
- climat, etc.

Les données récoltées sont transférées aux tracteurs connectés (parfois autonomes). Cela permet de doser finement le niveau d'engrais et d'arrosage sur telle ou telle parcelle et de réduire les coûts, tant financiers qu'énergétiques.

2.2.3.5. Les objets connectés dans les élevages :

Traceurs **GPS** pour le bétail, recueillement des habitudes alimentaires des bovins, les objets connectés ne sont pas seulement utiles aux agriculteurs, mais également aux éleveurs qui peuvent surveiller plus finement l'état de santé de leurs bêtes. Avez-vous déjà entendu parler des vaches connectées ? Fait amusant, il s'agit de l'animal le plus connecté au monde ! Son collier doté de nombreux capteurs permet une meilleure traçabilité mais aussi d'avoir des informations en temps réel sur son état de santé et son comportement.

2.2.3.6. Des villes intelligentes et connectées:

Appelées « smart cities » ou encore « villes connectées », les villes ont entamé leur transition digitale pour répondre aux enjeux de la société moderne. Urbanisme, économie et développement durable forment des enjeux importants pour les villes de demain, qui doivent répondre aux besoins d'une population toujours plus dense avec des ressources de plus en plus limitées. En effet, la pollution de même que la surpopulation sont deux problématiques majeures pour les agglomérations, qui doivent répondre aux enjeux économiques et sociaux d'une population croissante sans dégrader la qualité environnementale dans laquelle elle évolue.

Là encore, l'Internet des Objets est utile :

- foyers domotiques
- supports numériques
- capteurs et compteurs intelligents
- bornes de rechargement pour véhicules électriques
- éclairage citoyen intelligent, etc.

2.2.3.7. La domotique ou maison connectée :

Appelée également **domotique**, la maison intelligente est en train de se normaliser. Une étude du cabinet Juniper Research prévoit d'ailleurs un accroissement de 200 % du nombre d'objets connectés à l'intérieur des habitations d'ici fin 2021 [21].

Outre les objets de divertissement comme les télévisions intelligentes ou les enceintes connectées, la domotique a pensé également la sécurité et l'économie d'énergie au sein de l'habitat :

- centrale domotique : contrôle et programmation de différentes interventions à l'intérieur du foyer
- capteurs d'informations (système d'alarme, variations de température, etc.)
- actionneurs, qui permettent la programmation et le contrôle des différents appareils électroniques du foyer, même à distance



Fig 2.3 :maison connecté.

2.3. La domotique :

2.3.1. Définition :

La **domotique** est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans les bâtiments, plus ou moins « interopérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prises électriques, etc.)[22]. On peut résumer par dire que la domotique est le domaine technologique qui traite de l'automatisation du domicile, d'où l'étymologie du nom qui correspond à la contraction des termes "maison" (en latin "domus") et "automatique". Elle consiste à mettre en place des réseaux reliant différents type d'équipements (électroménager, hifi, équipement domotique, etc.) dans la maison. Ainsi, elle regroupe tout un ensemble de services permettant l'intégration des technologies modernes dans la maison[23].



Fig 2.4 : Ensemble des systèmes d'une maison intelligente.

2.3.2. Principe de fonctionnement d'une maison intelligente (domotique) :

Grâce au WIFI, aux ondes radio, à un réseau électrique ou à un système de câblage, vos équipements électriques communiquent entre eux, et sont commandés par un ordinateur central que vous aurez préalablement programmé. Les informations sont véhiculées sur le réseau électrique de la maison. Des logiciels pouvant s'installer sur tout ordinateur pourvu de systèmes d'exploitation permettent de gérer l'ensemble des équipements techniques de votre maison (éclairage, volets, chauffages, systèmes de sécurité). Par exemple, si votre cafetière n'est pas équipée d'un programmeur, en

branchant un module entre la prise murale et le cordon de votre machine, il est possible de **programmer de faire couler votre café au moment de votre réveil**. Avec du matériel plus évolué, c'est toute une série de situations en chaîne, appelées scénarios, qui est possible [24].

La domotique permet un confort absolu, mais peut être aussi d'une aide précieuse pour les personnes dépendantes et handicapées. Pour Augustin Rosenstiehl, architecte à l'atelier Soa : "Actuellement, nous avons un client qui est demandeur de domotique parce que sa femme est en chaise roulante. Ce n'est donc pas vraiment un choix pour la domotique mais un luxe nécessaire." Le contrôle de l'environnement et la gestion technique du bâtiment apportent accessibilité, autonomie, communication, circulation dans l'espace et confort des résidents [25].

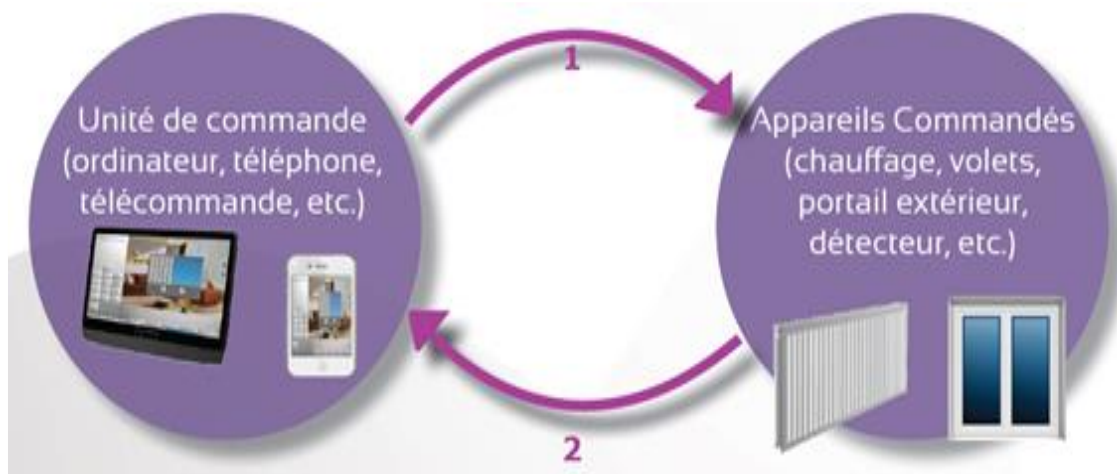


Fig 2.5 : principe de fonctionnement de maison intelligente.

2.3.3. Les fonction domotique de la maison intelligente :

Une habitation est constituée des plusieurs systèmes autonomes (chauffage, ventilation, éclairage, alarme, incendie, accès, ...) qui peuvent être intégrés grâce à la domotique. Il existe un grand nombre d'applications possibles qui diffèrent suivant les besoins des habitants et les fonctionnalités du système installé [26].

➤ **La gestion du chauffage et de l'énergie :**

La gestion pièce par pièce du chauffage ou de la climatisation optimisée par capteurs (mode, présence, fenêtre ouverte,...)

La gestion de la ventilation mécanique contrôlée (VMC) optimisée par capteurs (mode, présence, fenêtre ouverte,...)

La gestion des stores et volets roulants intégrée dans le concept d'économie d'énergie en protégeant des rayonnements du soleil en été et en laissant entrer le soleil en hiver.

La gestion de l'éclairage en fonction de la luminosité extérieure.

Mesurer les consommations d'électricité, d'eau et de gaz...

➤ **La sécurité :**

Alarme anti-intrusion

Commande automatique de l'éclairage intérieur et extérieur en cas d'alarme

Simulation de présence en cas d'absence

Contrôle d'accès

Alarmes techniques : inondation, incendie, gaz, CO2, CO

Coupure des arrivées d'eau, de gaz,...

Commande de la porte du garage et de l'éclairage extérieur lorsque que entrez et sortez et qu'il fait noir.

Caméras de vidéosurveillance accessibles sur les écrans de la maison et enregistrement automatique en cas d'alarme.

➤ **Le confort :**

Réglage de l'intensité lumineuse suivant l'activité.

Lorsque vous rentrez le soir, d'un simple geste le portail s'ouvre et l'éclairage extérieur s'allume. Enclenchement automatique de certains luminaires en fonction des détecteurs de présence.

Scénarios d'ambiance modifiables : lumières, chauffages, ouvrants, etc.

Diffusion multimédia (TV, Téléphone, Informatique) via prises de communication universelles RJ45.

➤ **La Commande de l'installation :**

Depuis les interrupteurs et écrans tactiles muraux

Depuis les tablettes portables de commande

Depuis les porte-clés radios

Depuis les notes-books et PC de bureau

Depuis le téléphone portable par SMS

Accès distant via Internet



Fig 2.6 : Différentes fonctions d'une maison intelligente .

2.4. Future de l'IoT :

Les possibilités de l'Internet of Everything sont infinies. Lorsque tout est connecté, peut communiquer et apprendre, le chemin de l'innovation est ouvert.

Mais des obstacles sont présents : actuellement, 99% de tous les objets du monde physique ne sont pas reliés à internet. À l'aide de capteurs, certains 'bêtes' appareils peuvent être

rapidement connectés à internet. En outre, il reste la question de la **sécurité**. Car dès que des objets sont reliés à internet, ils peuvent devenir des cibles potentielles des pirates.

Même si l'Internet of Everything peut rapidement devenir une réalité, le chemin à parcourir est encore long pour notre société contemporaine [27].

Avec l'apparition de plus en plus d'appareils IOT sur le marché, certaines mesures de cyber sécurité et de protection de votre vie privée sont de rigueur :

- Changez les noms d'utilisateur par défaut et utilisez des mots de passe forts pour votre réseau Wi-Fi et vos appareils intelligents. N'utilisez aucune information qui pourrait être associée à votre nom, votre adresse ou votre numéro de téléphone.
- Entrez un mot de passe pour verrouiller l'écran de vos appareils.
- Utilisez des logiciels de sécurité mis à jour pour votre ordinateur à la maison et pour vos appareils connectés. Faites les mises à jour du système d'exploitation de tous vos appareils.
- Isolez vos appareils d'IOT de votre réseau principal afin de limiter les dommages potentiels en cas de cyberattaque. Demandez l'aide de votre fournisseur de services pour créer un réseau séparé pour vos appareils IOT.
- Avant d'acheter un appareil d'IOT ou de télécharger une application, comprenez bien quelles sont les informations qui seront recueillies et pour quelles raisons.
- Désactivez la fonction de géolocalisation lorsque vous n'en avez pas besoin. Si une application peut voir votre emplacement, un cybercriminel le pourrait aussi.
- Désactivez la caméra et le microphone lorsque vous ne les utilisez pas.
- Assurez-vous de maintenir de bonnes habitudes de cyber sécurité. Par exemple, n'ouvrez pas de pièces jointes qui proviennent de sources inconnues et ne permettez pas la connexion automatique [28].

2.5. Conclusion :

Ce chapitre nous a permis d'avoir un aperçu sur l'internet des objets , ses fonctionnalités et ses domaines d'application, surtout dans la domotique qui fait l'objet de ce projet de fin d'études.

Chapitre03 :

Réalisation du

Systeme domotique

3. Réalisation du système domotique :

3.1. Introduction :

Ce chapitre est consacré à la réalisation de notre prototype d'une maison intelligente équipée de 3 LED qui jouent le rôle des lampes, d'un capteur de température, d'un capteur de gaz, d'une caméra de surveillance, d'une porte et d'un ventilateur.

Dans ce chapitre, nous détaillons les tâches que nous avons pu réaliser étape par étape ainsi que les différents tests de validation.

3.2. Représentation de système réalisé:

Dans ce travail, on a réalisé un système **domotique** à base de **Raspberry Pi 3**,

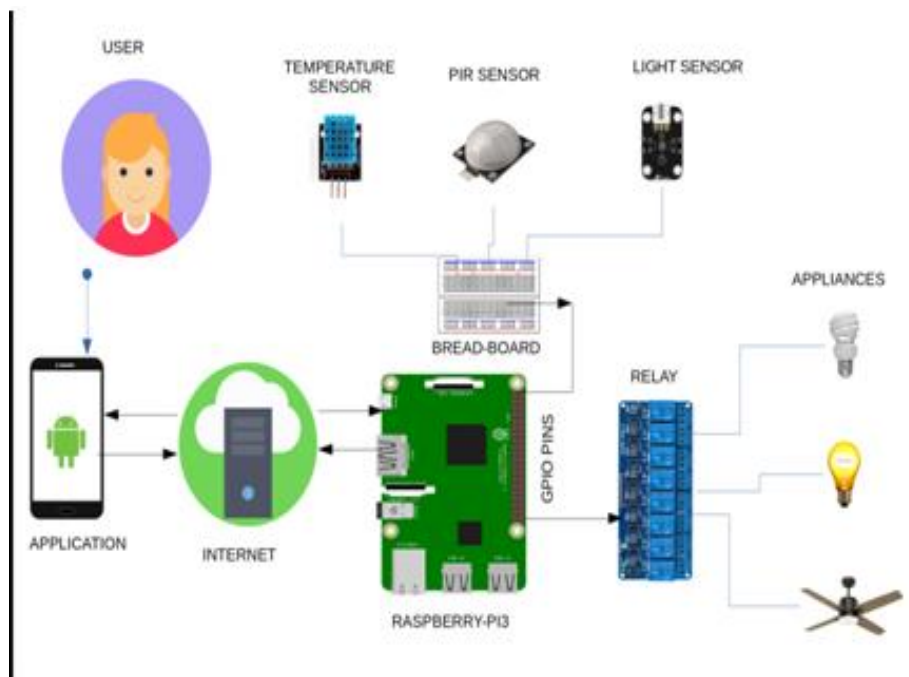


Fig 3.1 : Schéma général du système.

Nous avons utilisé l'application Android pour le contrôle et l'acquisition des informations de Raspberry Pi via Internet. Raspberry, qui à son tour reçoit des commandes de l'application et met en œuvre ou envoie les capteurs connexes puis envoie les informations et reçoit les commandes de Raspberry.

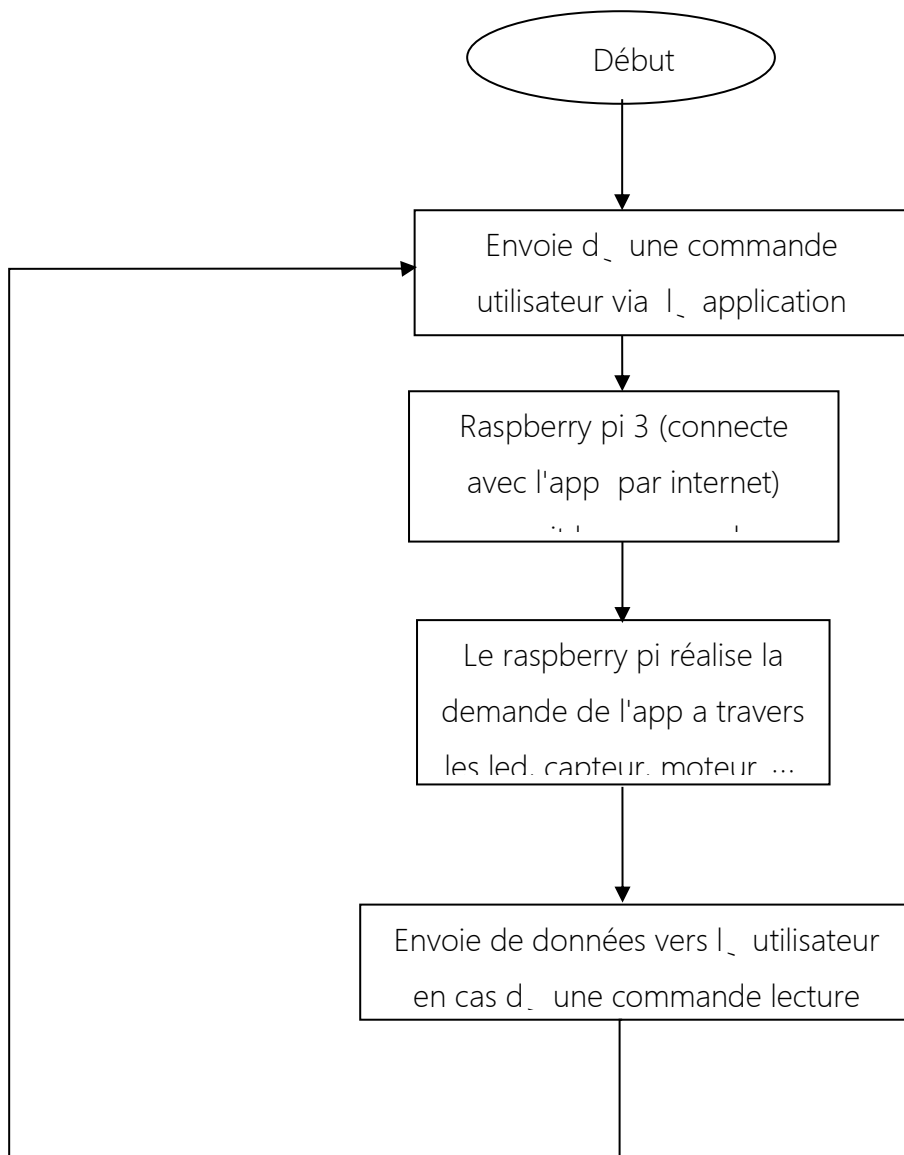


Fig 3.2 : Organigramme de la communication entre l’application utilisateur et Raspberry.

3.3. Partie matérielle du projet :

Dans ce projet on a utilisé les composants suivants pour la réalisation d’un système domotique (maison intelligente):

- Raspberry pi 3
- 3 LED vert (les lampes pour les chambres).
- 2 Servo moteurs pour la porte et la ventilateur.
- DH11 capteur température.
- Camera USB pour la surveillance.

3.3.1. La configuration de composant avec raspberry pi :

- **Les 3 Led vert utilisé :**

Led (*light-emitting diode*), désigne un composant optoélectronique qui permet l'émission de lumière monochromatique[29].

On la branche cette fois-ci avec la pin 11. Pour le programme python, il reste le même pour la LED2, LED3 vert à l'exception du numéro de broche qui doit changer.

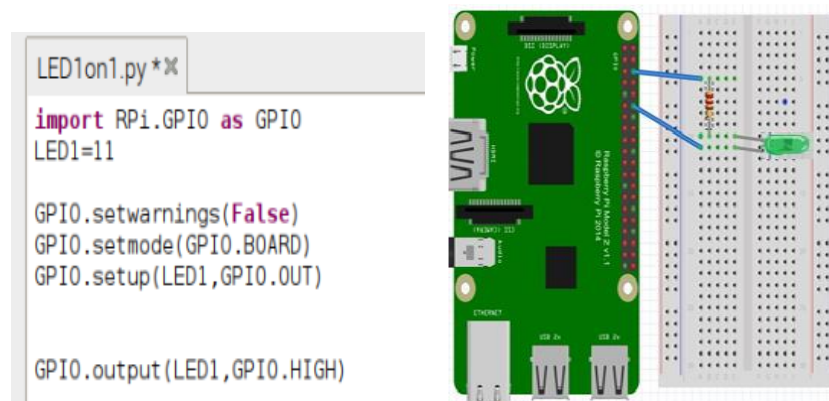


Fig 3.3 : Programme python d'une LED avec raspberry pi 3

- **Servo moteur :**

La plupart des servo moteurs ne sont pas *continus* ; c'est-à-dire qu'ils ne peuvent pas pivoter complètement, mais plutôt sur un angle d'environ 180 degrés.

La position du servomoteur est définie par la longueur d'une impulsion. Le servo s'attend à recevoir une impulsion au moins toutes les 20 millisecondes. Si cette impulsion est haute pendant 1 milliseconde, l'angle d'asservissement sera nul; si c'est 1,5 milliseconde, il sera à sa position centrale; et si c'est 2 millisecondes, ce sera à 180 degrés [30].



Fig 3.4 : Programme python d'un servomotoravec raspberry pi 3.

- **DH11 :**

Le capteur DHT11 est lui capable de mesurer des températures de 0 à +50°C avec une précision de +/- 2°C et des taux d'humidité relative de 20 à 80% avec une précision de +/- 5%. Une mesure peut être réalisée toutes les secondes[31].

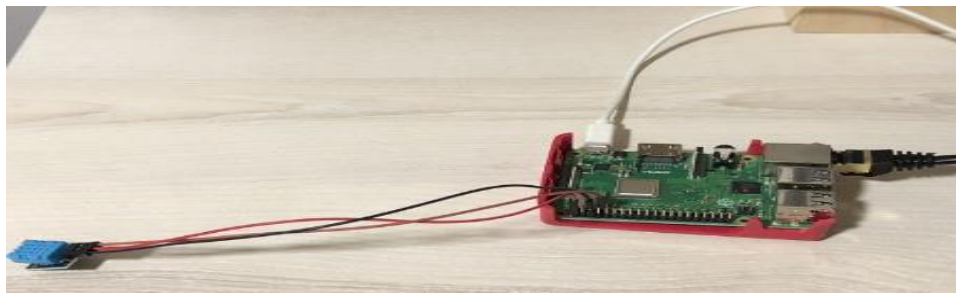
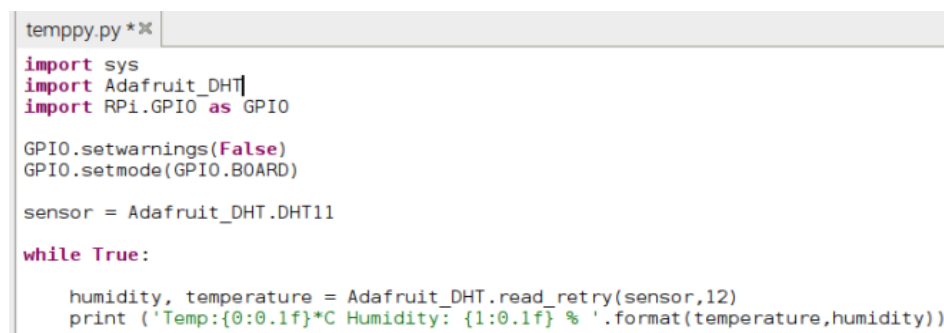


Fig 3.5 : Programme python d'un dh11 avec raspberry pi 3.

3.4. Partie logiciel de ce projet

Dans ce travail on a utilisé le langage Python pour la programmation côté Raspberry (serveur) et l'Android Studio pour la programmation côté smartphone (client):

- **IDE PYTHON** pour raspberry pi: commande/lecture de: led, capteur, motor
- **IDE ANDROID STUDIOS (JAVA)** pour la conception de l'application Android.

3.4.1. La conception d'application Android :

3.4.1.1. Android Studio:

Android Studio est un environnement de développement (IDE - Integrated Développement Environnement) pour développer des applications sous la plateforme Android. Il est basé sur IntelliJ IDEA de JetBrains et permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément. En 2014, Android Studio passe de version bêta à une version stable 1.0. L'environnement devient alors conseillé par Google et l'environnement Eclipse se voit délaissé[32].



Fig 3.6 : Icône de l'Android studio.

3.4.1.2. Démarrer Android Studio :

Après avoir installé l'IDE pour Android Studio, double-cliquez sur l'icône de l'application Android Studio pour la démarrer.

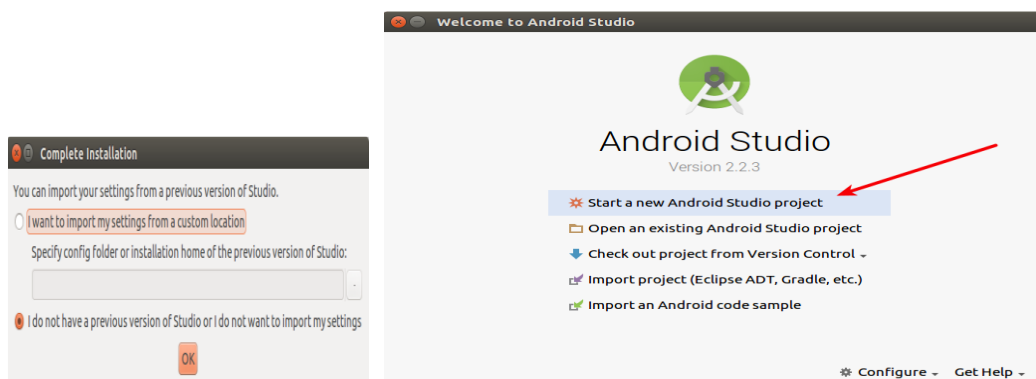


Fig 3.7 : 1^{er} Démarrage de l'Android Studio.

➤ Créer un nouveau projet (New project) :

Démarrer un nouveau projet Android Studio dans la fenêtre de bienvenue et nommez le projet avec le même nom. Pour l'application. Le nom de l'application précédé de votre nom ou du nom de domaine de votre société doit fournir un nom de package suffisamment unique. Si vous ne prévoyez pas de publier l'application, vous pouvez accepter le domaine d'exemple par défaut.

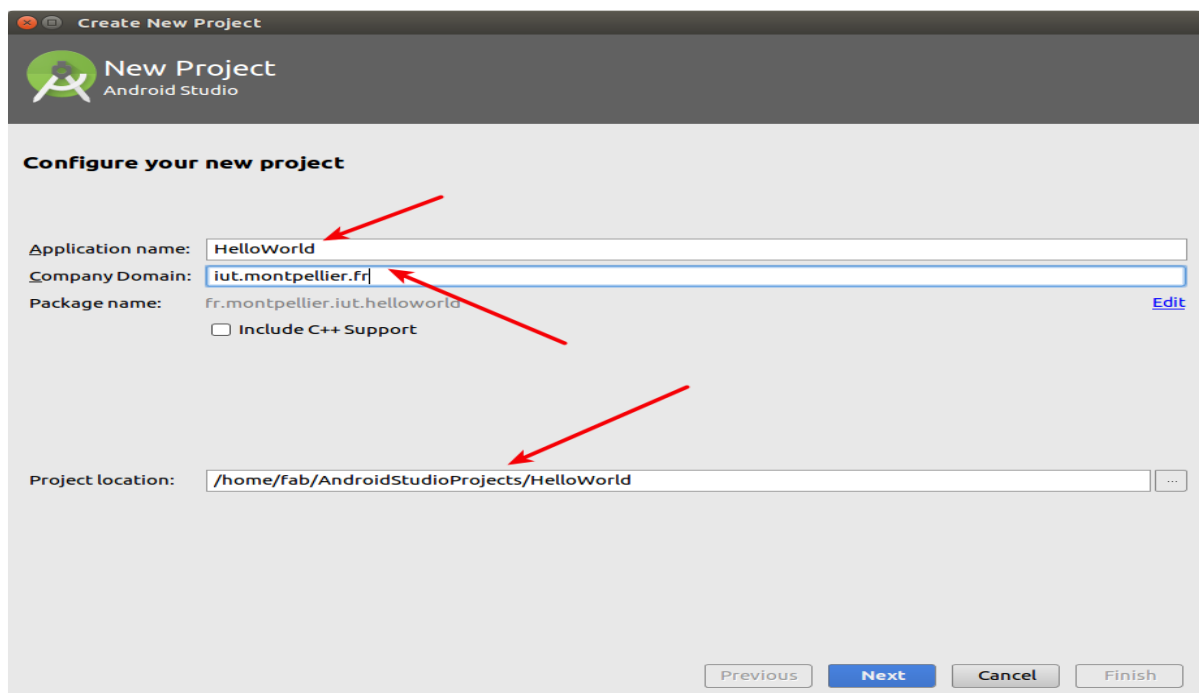


Fig 3.8 : Nouvel projet.

➤ Sélection de l'API Minimum :

Différents appareils exécutent différentes versions du système Android, comme Android 4.0.3 ou Android 4.4. Chaque version successive ajoute souvent de nouvelles API non disponibles dans la version précédente. Pour indiquer quel ensemble d'API est disponible, chaque version spécifie un niveau d'API. Par exemple, Android 1.0 est le niveau d'API 1 et Android 4.0.3 est le niveau d'API 15.

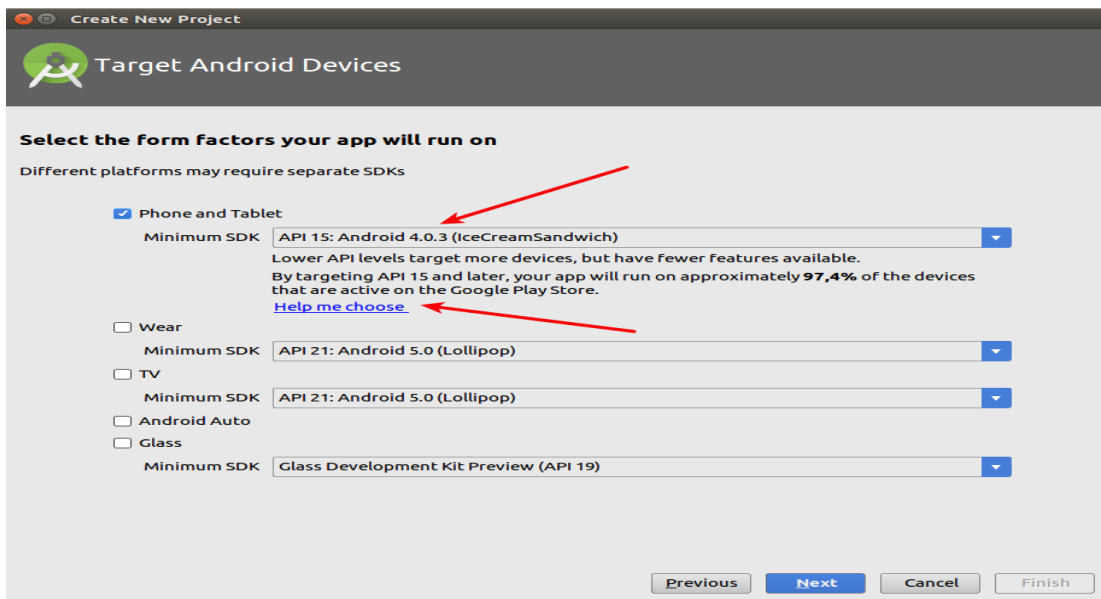


Fig 3.9 :Selection de la version de l'Android.

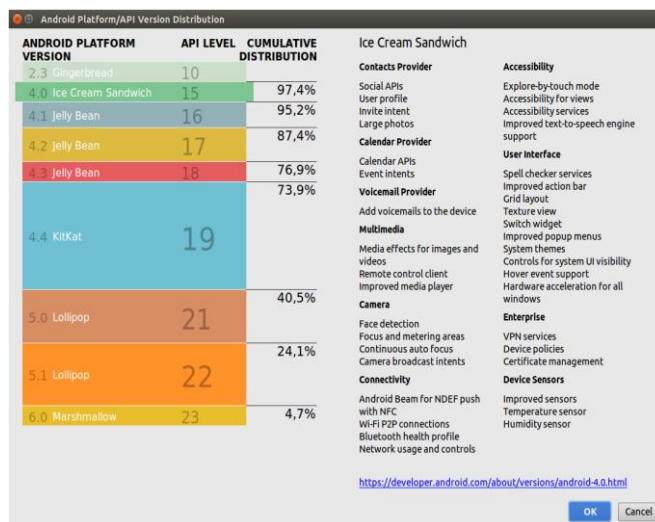


Fig 3.10 :Liste des API.

➤ Choix de l'activité héritée : Empty Activity

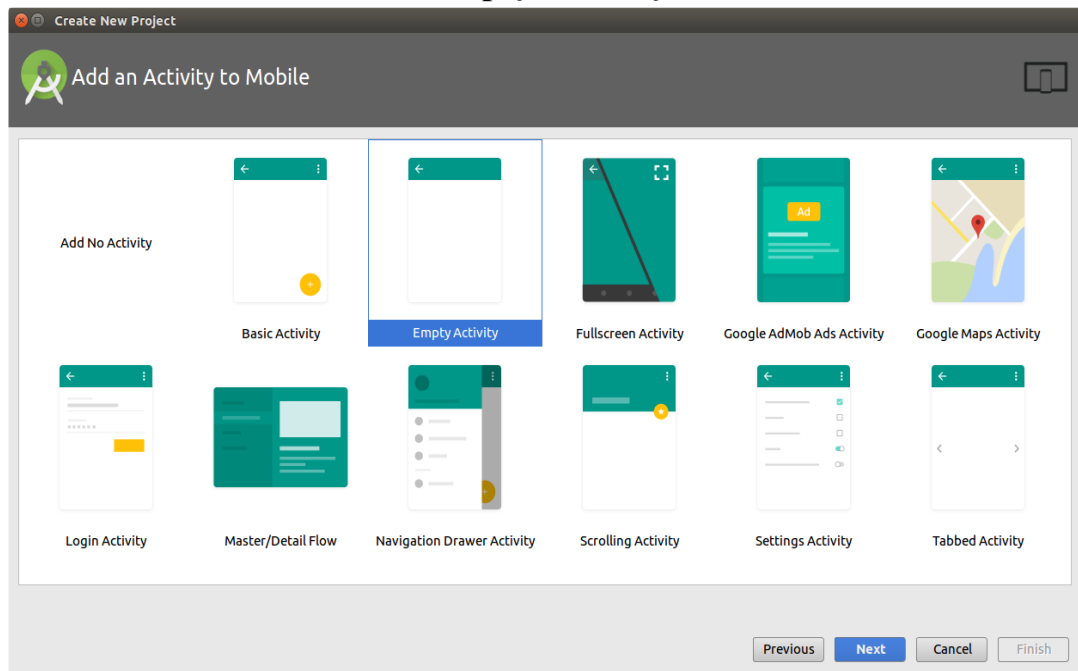


Fig 3.11 : Empty Activity.

➤ Paramétrage des noms de fichiers :

Dans cette fenêtre on a donné un nom à l'activité Empty.

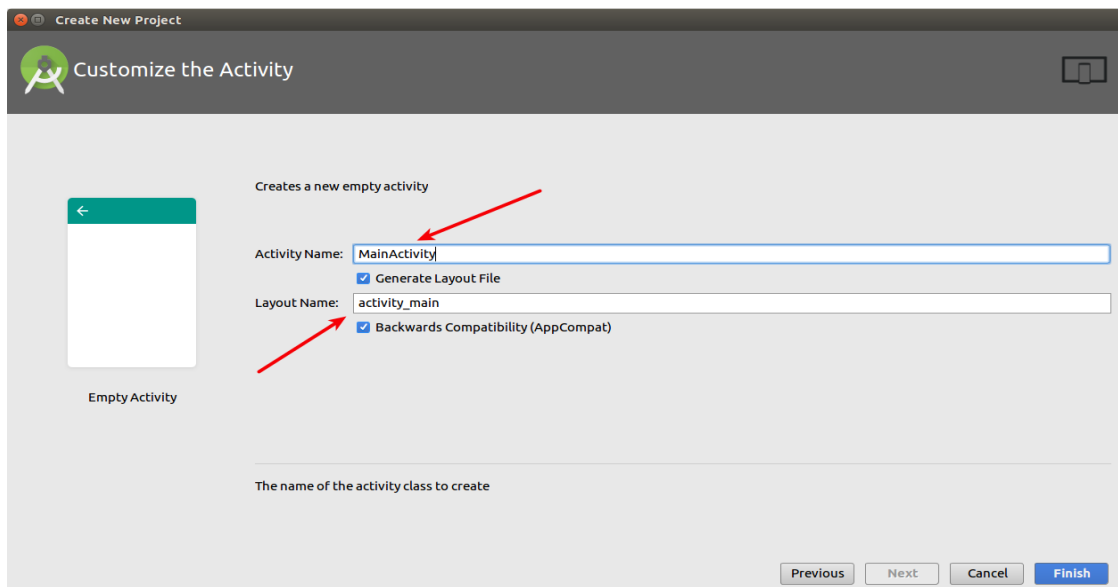


Fig 3.12 : Le nom de Empty Activity.

- **Vue globale** : Après avoir terminé les étapes précédentes la forme finale de la création d'un nouveau projet est montré sur la figure 3.11.

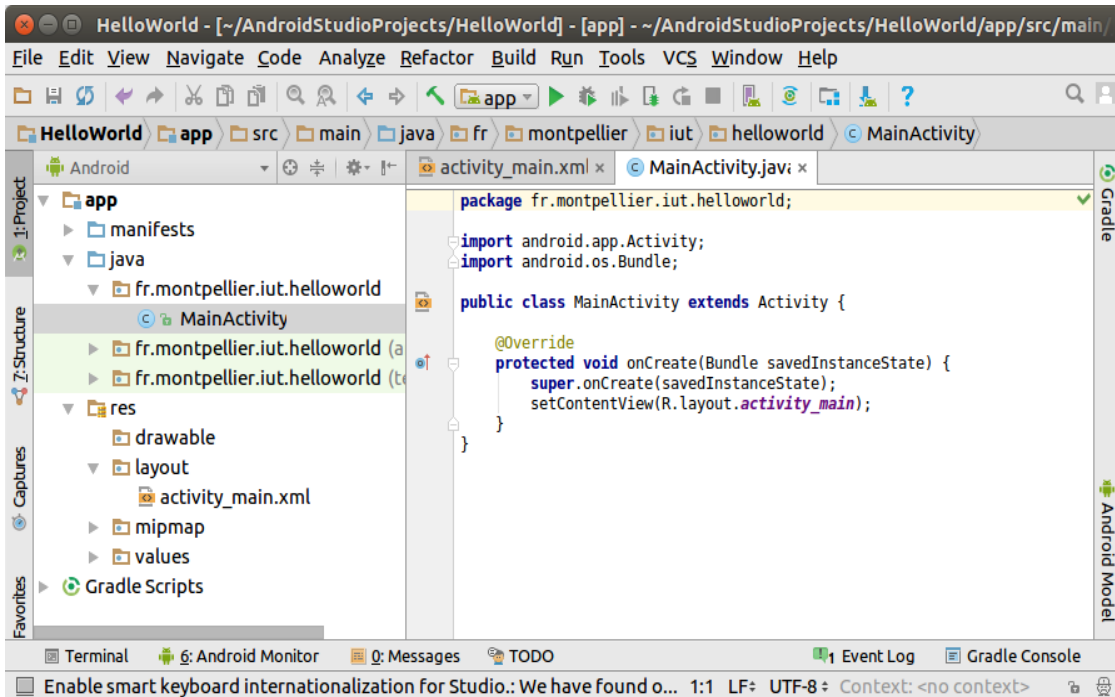


Fig 3.13 : Ecran global d'un nouveau projet mainActivity.java.



Fig 3.14 : Ecran global d'un nouveau projet Activity_main.xml.

3.4.1.3. Contenu du projet :

Tout projet Android doit respecter une hiérarchie bien précise qui permettra au compilateur de retrouver les différents éléments et ressources lors de la génération de l'application. Cette hiérarchie favorise la modularité des applications Android.

A la création du projet, Android Studio crée automatiquement des dossiers contenant les fichiers de code Java, les fichiers XML, et les fichiers multimédias.

L'explorateur de projet vous permettra de naviguer dans ces dossiers.

Les dossiers les plus utilisés sont **java** et **res**. Le premier contient le code Java qui définit le comportement de l'application (situé dans le répertoire de votre projet sous **app\src\main**) et la seconde comporte des sous-dossiers (dans **app\src\main\res**) où sont stockées les ressources qui définissent l'interface de l'application (l'apparence)[33].

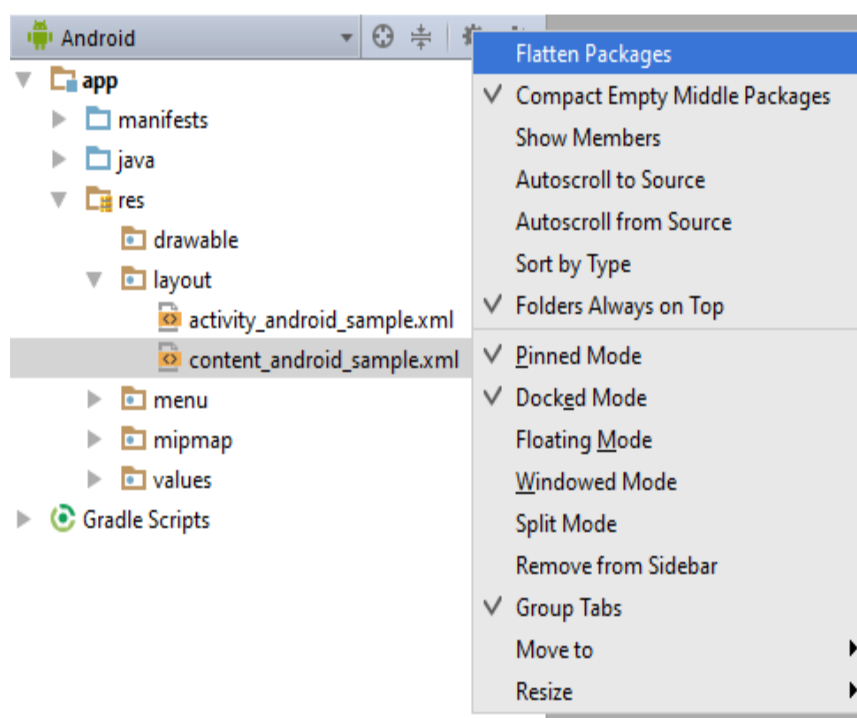


Fig 3.15 : Structure du projet dans Android Studio.

Le tableau suivant donne un aperçu des ressources prises en charge ainsi que le sous-dossier standard.

Ressource	Répertoire	Description
Dessins (drawables)	/res/drawables	Images (fichiers PNG ou JPEG par exemple) ou fichiers XML qui décrivent un dessin (Drawable).
Valeurs simples	/res/values	Permet de définir des chaînes de caractères, des couleurs, des dimensions, des styles et des tableaux statiques de chaînes ou d'entiers via des fichiers XML. Par convention chaque type est stocké dans un fichier séparé, par exemple, les chaînes sont définies dans le fichier /res/values/strings.xml
Mises en page (layouts)	/res/layout	Fichiers XML avec les mises en page utilisées pour définir l'interface utilisateur des activités et fragments.
Styles et Thèmes	/res/values	Fichiers qui définissent l'apparence de votre application Android.
Animations	/res/animator	Définitions XML pour l'API d'animation qui permet de modifier des propriétés arbitraires d'objets au cours du temps.
Données brutes (raw data)	/res/raw	Des fichiers arbitraires enregistrés dans leur forme brute. Vous y accédez via un objet InputStream.
Menus	/res/menu	Définit les propriétés des entrées d'un menu.

Tableaux 3.1 :Représentation des sous –dossier de ressources .

3.4.1.4. L'AVD Android:

Android Virtual Device est un dispositif mobile virtuel qui s'exécute sur l'ordinateur et permet de développer et de tester des applications Android sans l'aide d'un dispositif physique.

L'émulateur Android imite toutes les fonctionnalités matérielles et logicielles d'un dispositif mobile typique, tel que la lecture des fichiers audio et vidéo, stockage des données, sauf qu'il ne peut pas passer des appels réels[34].

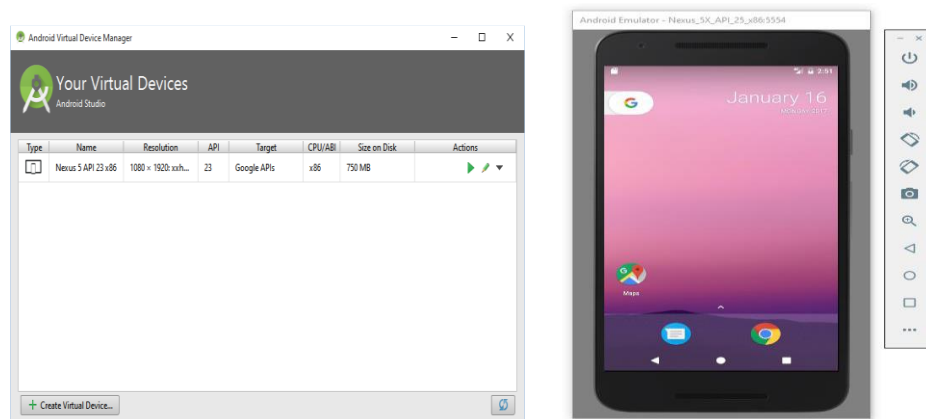


Fig 3.16 : Android Virtual Device.

AVD permet de tester les applications Android sur les différentes versions et configurations Android sans avoir accès au matériel réel.

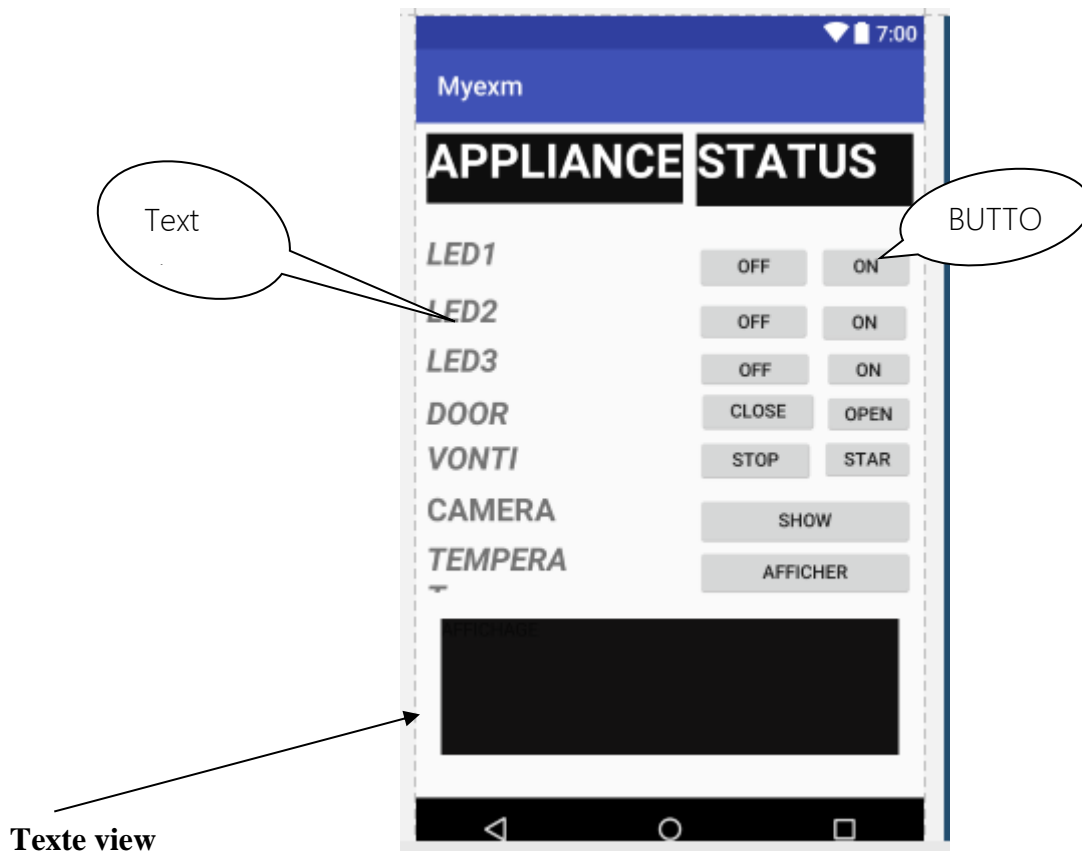
Lors de la création d'un AVD, on définit la configuration de l'appareil virtuel. Cela comprend, par exemple, la résolution, la version Android de l'API et la densité de votre écran[3.1].

3.4.2. La création de mon application

Dans ans se travail on a utilisé deux attributs XML:

Un bouton : c'est un composant cliquable affichant un texte. Utiliser, par exemple, inciter l'utilisateur a effectuer une action ou indiquer à l'utilisateur l'élément pour soumettre un formulaire [35].

Un Texte view : est utilisé pour rentrer afficher sur une page une information a un utilisateur. Cela vous permet de mettre de plus un titre a des images ou des descriptions [36].



Interface Pour afficher

Fig 3.17 : application réalisé

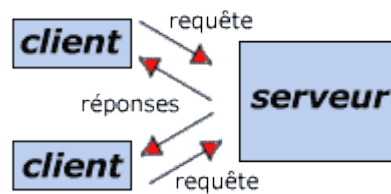
Température

- Pour contrôler la porte, on clique sur le bouton **open /close** .
- Même chose pour contrôler les LEDs, on clique sur le bouton **ON /OFF** et pour contrôler le ventilateur, on clique sur le bouton **STAR /STOP**.
- Lorsqu'on clique sur le bouton **AFFICHER** ça nous redirige vers l'interface qui affiche le degré de température.
- Pour accéder au vidéo streaming, on clique sur le bouton **Show**.

3.4.3. Application java du client et du serveur:

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des **machines clientes** (des machines faisant partie du réseau) contactent un **serveur**, une machine généralement très puissante en terme de capacités d'entrée-sortie, qui leur fournit des **services**. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc[37].

Un système client/serveur fonctionne selon le schéma suivant :



- Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port

3.4.3.1. Programmation réseau – Les sockets :

En anglais un **socket** est un "trou" qui laisse passer des choses, comme une prise électrique, l'emplacement du processeur, ou une bouche -on va s'arrêter la pour les exemples -

Le socket est donc dans notre cas **une association au niveau de l'OS entre un programme qui tourne en boucle et le port de la machine** qui lui a été dédié. On dit d'ailleurs que le programme écoute le port qui lui a été réservé. Il écoute le port et répond aux demandes faites par ce port[38].

Les sockets constituent un moyen très puissant pour deux processus de communiquer entre eux : ils peuvent maintenant se parler *via* un réseau, donc parfois à des kilomètres de distance. La communication par socket est souvent comparée aux communications humaines. On distingue ainsi deux modes de communication:

- Le mode connecté (comparable à une communication téléphonique), utilisant le protocole TCP. Dans ce mode de communication, une connexion durable est établie entre les deux processus, de telle façon que l'adresse de destination n'est pas nécessaire à chaque envoi de données.
- Le mode non connecté (analogue à une communication par courrier), utilisant le protocole UDP. Ce mode nécessite l'adresse de destination à chaque envoi, et aucun accusé de réception n'est donné [39].

Notre projet on a utiliser protocole socket TCP et choisi la carte raspberry pi comme serveur et l'application Android comme client . Le serveur, c'est une machine qui va traiter les requêtes du client.

nous allons créer deux classe : classe serveur pour raspberry pi et classe client application Android . Le serveur *écoute* donc en attendant des connexions et le client se connectent au serveur.

a) Classe socket côté client (application) :

Pour que le client se connecte au serveur, il nous faut deux informations :

- Le **nom d'hôte** (*host name* en anglais), qui identifie une machine sur Internet ou sur un réseau local.
- Un numéro de port, qui est souvent propre au type d'information que l'on va échanger. Si on demande une connexion web, le navigateur va en général interroger le port 80 si c'est en http ou le port 443 si c'est en connexion sécurisée (https). Le numéro de port est compris entre 0 et 65535 (il y en a donc un certain nombre !) et les numéros entre 0 et 1023 sont réservés par le système. On peut les utiliser, mais ce n'est pas une très bonne idée[40].

```
package com.example.poste.myexm;

import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;

public class MainActivity extends AppCompatActivity {
    private static String ip = "192.168.43.230";
    private static int port = 2004;
    private static Socket s;
    private static ServerSocket ss;
    private static InputStreamReader isr;
    private static BufferedReader br;
    String message = "";
    private static PrintWriter pp;
    Handler handler;
}
```

Les bibliothèques

Déclaration

Numéro de port et adresse IP de raspberry pi

```
class myTask extends AsyncTask<Void ,Void ,Void > {
    @Override
    protected Void doInBackground(Void... params) {
        try {
            s = new Socket(ip, port);
            pp = new PrintWriter(s.getOutputStream());
            isr = new InputStreamReader(s.getInputStream());
            br = new BufferedReader(isr);
            pp.write(message);
            pp.flush();
            pp.close();
            isr.close();
            br.close();
            s.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

Class socket client

b) Classe socket côté serveur (Raspberry pi) :

Voici dans l'ordre les étapes du serveur. Les étapes sont très simplifiées suivant les fonction important d' un socket serveur :

Commençons donc, dans la joie et la bonne humeur, par importer notre module socket **import socket.**

Choisi un objet socket s = connexion avec serveur

- *Construire notre socket*

Nous allons pour cela faire appel au constructeur socket. Dans le cas d'une connexion TCP, il prend les deux paramètres suivants, dans l'ordre :

socket.AF_INET : la famille d'adresses, ici ce sont des adresses Internet ;

socket.SOCK_STREAM : le type du socket, SOCK_STREAM pour le protocole TCP.

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

- *Connecter le socket*

Ensuite, nous connectons notre socket. Pour une connexion serveur, qui va attendre des connexions de clients, on utilise la méthode bind. Elle prend un paramètre : le tuple (nom_hote, port).

```
>>> s.bind(('', 2004))
```

- **Faire écouter notre socket**

Bien. Notre socket est prêt à écouter sur le port 2004 mais il n'écoute pas encore. On va avant tout lui préciser le nombre maximum de connexions qu'il peut recevoir sur ce port sans les accepter. On utilise pour cela la méthode listen. On lui passe généralement 5 en paramètre.

Cela veut dire que si 5 clients se connectent et que le serveur n'accepte aucune de ces connexions, aucun autre client ne pourra se connecter. Mais généralement, très peu de temps

après que le client ait demandé la connexion, le serveur l'accepte. Vous pouvez donc avoir bien plus de clients connectés, ne vous en faites pas.

```
>>> s.listen(5)
```

- *Accepter une connexion venant du client*

Enfin, dernière étape, on va accepter une connexion. Aucune connexion ne s'est encore présentée mais la méthode **accept** que nous allons utiliser va bloquer le programme tant qu'aucun client ne s'est connecté.

Il est important de noter que la méthode **accept** renvoie deux informations :

- le socket connecté qui vient de se créer, celui qui va nous permettre de dialoguer avec notre client tout juste connecté ;
- un tuple représentant l'adresse IP et le port de connexion du client.

```
>>> connexion_avec_client, infos_connexion = s.accept()
```

- **Faire communiquer nos sockets**

Bon, maintenant, comment faire communiquer nos sockets ? Eh bien, en utilisant les méthodes **send** pour envoyer et **recv** pour recevoir.

Donc côté serveur ,La méthode **send** vous renvoie le nombre de caractères envoyés.

```
>>> connexion_avec_client.send(b"Je viens d'accepter la connexion")
```

Maintenant, côté client, on va réceptionner le message que l'on vient d'envoyer. La méthode **recv** prend en paramètre le nombre de caractères à lire. Généralement, on lui passe la valeur 1024. Si le message est plus grand que 1024 caractères, on récupérera le reste après.

```
>>> msg_recu = connexion_avec_serveur.recv(1024)
```

- **Fermer la connexion**

Pour fermer la connexion, il faut appeler la méthode **close** de notre socket.

Côté serveur :

```
>>> connexion_avec_client.close()
```

Et côté client :

```
>>> connexion_avec_serveur.close ()
```

```
serv11.py *  
  
import socket  
import sys  
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)  
print(s.getsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR))  
s.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR,1)  
s.bind(("",2004))  
s.listen(5)  
  
while True :  
    c,address=s.accept()  
    print("connection from {}".format(address[0]))  
    data=c.recv(1024).decode("utf-8")  
    if not data:  
        break;  
    else:  
        print (data)  
        c.send(data)  
c.close()  
s.close()
```

Fig 3.18 : Programme socket server python.

3.5. Conclusion :

Dans Ce chapitre nous avons présenté notre travail qui consiste en une maison intelligente avec un serveur (Raspberry pi). Ce dernier communique avec un client (application Android de l'utilisateur). Les éléments de la maison sont des capteurs (capteur de température, capteur d'humidité, ...etc.) et des actionneurs (lampes, servo moteur, ...etc).

Conclusion général

Au bout de notre cursus en master 2 "Electronique des Systèmes Embarqués ", nous avons pu réaliser une maison intelligente.. Effectivement, dans le cadre du thème de la domotique avec raspberry pi, nous avons conçu une maison automatisée.

Notre travail est basé sur une carte Raspberry pi comme un serveur et le développement d'une application Android sur les technologies mobiles (Smartphone) pour la réalisation et la conception d'une maison intelligente. Elle est capable de gérer l'éclairage des chambres et les portes, la température et la surveillance de la maison à partir d'une caméra.

Ceci nous a amené à découvrir une nouvelle plateforme de développement et à enrichir nos connaissances théoriques ainsi que pratiques et notre expérience dans le domaine de l'IoT. Ce dernier qui est devenu un grand domaine de recherche et un grand marché de travail au monde. Ce travail nous a également permis de nous initier dans le domaine de la programmation mobile, en particulier, la programmation sous le système Android ainsi sur les systèmes embarqués.

Le seul point « négatif », serait sûrement le manque de temps pour pouvoir encore approfondir ce travail, car ce dernier ne s'arrête pas ici il a encore plusieurs taches qi peuvent être améliorées.

En effet, beaucoup de possibilités s'offrent aux passionnés de domotique, tant sur le matériel disponible que sur les actions à réaliser. Cependant rien ne nous empêche de continuer sur cette voie de notre propre côté. Avec l'aide des outils, les plateformes, et les bibliothèques Open Source, et les communautés qu'elles supportent, le développement de nouveaux produits libres n'a pas été aussi facile.

En perspective de présent travail nous envisageons d'ajouter des modules plus sophistiquées au Raspberry pour la mise en œuvre de tâches et fonctions complexes et avancés.

Bibliographie

- [1] robert.cireddu.free.fr/SIN/Raspberry%20Pi.pdf
- [2] chimiphyk.free.fr/cours/cours/speISN/1-Architecture/ArchiRaspb.pdf
- [3] <https://ismvsectioninfo.wordpress.com/2018/10/01/raspberry-pi/>
- [4] https://fr.wikipedia.org/wiki/Raspberry_Pi
- [5] <https://domopi.eu/utiliser-le-port-gpio-de-votre-raspberry-pi/>
- [6] <https://alain-michel.canoprof.fr/.../raspberrypi/.../premiers-pas-raspberrypi...>
- [7] <https://nicolargo.developpez.com/tutoriels/raspberry-pi/deballage-installation/>
- [8] <http://espace-raspberry-francais.fr/Configuration/Connecter-Raspberry-Pi-au-PC-par-cable-Ethernet-RJ45/>
- [9] <https://www.pobot.org/-Logiciels-et-langages-de-.html>
- [10] https://fr.wikibooks.org/wiki/Programmation_Python/Introduction
- [11] https://stopleftynonmerci.org/.../059-scl-leguidedelamaisonetdesobjetsconnecte_s.pdf
- [12] Introduction à l'internet de l'objet et réalisation D'un système domotique
- [13] [https://www.google.com/search?q=L'Internet+des+objets+\(IoT\)+se+traduit+à+l'heure+actuelle+par+l'accroissement+d+nombre+d'objets+connectés.pdf](https://www.google.com/search?q=L'Internet+des+objets+(IoT)+se+traduit+à+l'heure+actuelle+par+l'accroissement+d+nombre+d'objets+connectés.pdf)
- [14] dspace.univ-biskra.dz:8080/jspui/.../1/Réalisation_d'un_système_de_contrôle...pd
- [15] [Le guide de la maison et des objets connectés .pdf.](#)
- [16] https://www.pensezcybersecurite.gc.ca/cnt/rsks/ntrnt.../_fls/ntrnt-thngs-bsnss-fr.pdf
- [17] <http://blog-french-iot.laposte.fr/liot-quest-ce-que-cest/>
- [18] Mr: BOUHARAOUA Abderrahim et Mr: BOUKLI HACENE Mohammed Imad

<<Automatisation d'une maison intelligent via une application Android >> Mémoire de Master - de l'Université Tlemcen .Abou Beker Belkraid ...

dspace.univ-tlemcen.dz/bitstream/.../Ms.Tel.Bouharaoua%2BBoukli%20hacene.PDF

[19] <https://medium.com/iot-makers/les-4-fondamentaux-de-linternet-des-objets-3da39a18ae0>

[20] <https://blog.lesjeudis.com/10-applications-de-l-internet-des-objets-qui-revolutionnent-la-societe>

[21] <https://www.energir.com/blogue/ma-maison/web-3-0-domotique-maison-intelligente/>

[22] <https://fr.wikipedia.org/wiki/Domotique>

[23] https://sti.discip.ac-caen.fr/IMG/pdf/Presentation_Domotique2.pdf

[24] <https://www.linternaute.fr/bricolage/guide-maison-et-jardin/1335625-la-domotique-principe-et-fonctionnement/>

[25] <http://www.caseo-professionnels.com/articles/la-domotique-principe-et-fonctionnement>

[26] <https://knx-automation.com/domotique/les-fonctions-domotiques-de-la-maison-intelligente/>

[27] <https://onesmartcontrol.com/fr/qu-est-ce-que-domotique/linternet-des-objets-et-levolution-de-la-domotique/>

[28] <https://www.pensezcybersecurite.gc.ca/cnt/rsks/ntrnt-thngs/hm-fr.aspx>

[29] <https://www.futura-sciences.com/maison/definitions/maison-led-6968/>

[30] <https://www.instructables.com/id/Servo-Motor-Control-With-Raspberry-Pi/>

[31] <https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/>

[32] https://fr.wikipedia.org/wiki/Android_Studio

[33] <https://hal.archives-ouvertes.fr/cel-01241650v2/document>

[34] Mémoire de fin d'études Conception et Développement d'une application Java sous Android (Donner du sang) pdf

[35] <https://vogella.developpez.com/tutoriels/android/developpement-androidstudio-eclipse/#L2-6>

[36] msathias-seguy.developpez.com/tutoriels/android/utiliser-elements-graphiques-fondamentaux/#LIII-B

[37] <https://www.supinfo.com/articles/single/2810-creer-textview-android>

[38] <https://python-django.dev/page-reseaux-sockets-python-port>

[39] <http://www-etud.iro.umontreal.ca/~gottif/bdeb/inf32/sockets.htm>

[40] <https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/234698-gerez-les-reseaux>