



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Filière : Automatique
Option : Automatique et Informatique Industriel

Réf. :

Présenté et soutenu par :
OSMANE LAMRI

Le : **Lundi 25 juin 2018**

COMMANDE DES CHARGES 220V PAR LA RECONNAISSANCE VOCALE SOUS SYSTEME ANDROID ET CARTE ARDUINO UNO

Jury :

Mme. ABDOU Latifa	MCA	Université Biskra	Président
Mr. ZITOUNI Athmane	MCA	Université Biskra	Encadreur
Mr. REDHOUANE Ahmed	Ingenieur.Ensg	Institut National Biskra	Co.Encadreur
Mr. CHELIHI Abed Elghani	MCA	Université Biskra	Examineur

Remerciement

Nous tenons à remercier tout premièrement Dieu le tout puissant pour la volonté, la santé et la patience, qu'il nous a donné durant toutes ces longues années.

« الحمد لله الذي هدانا لهذا وما كنا لنهتدي لولا أن هدانا الله »

Aussi, nous tenons également à exprimer nos vifs remerciements à notre encadreur Mr ZITOUNI.A pour ses conseils, la confiance qu'ils nous a accordés.et nos remerciements à Mr REDOUANE AHMED. Nos remerciements à tous les membres du jury qui ont accepté d'examiner notre travail. Nous adressons nos remerciements à tous les membres de jury qui nous ont fait le grand honneur en acceptant de juger ce travail, espérons qu'il soit digne de leurs intérêts.

Nous tenons à remercier vivement toutes les personnes qui nous ont aidés à élaborer et réaliser ce mémoire, ainsi à tous ceux qui nous ont aidés de près ou de loin à accomplir ce travail.

Dédicace

*À la plus belle créature que Dieu a crée sur terre ,,,
À cet source de tendresse, de patience et de générosité,,,*

À ma mère

À le personne qui à toujours me donné la volonté

À mon père

À tous mes frères qui ont été toujours avec moi

À tous mes amis et collègues

À tous les étudiants de la promotion 2017/2018

Département Automatique

A tous ceux qui, par un mot, m'ont donné la force de continuer ...

ملخص :

يسمح لنا المشروع بالغوص في عالم التواصل من اجل الوصول الى إنشاء تطبيق يربط بين الهاتف الذكي و بطاقة الاردوينو و الاجهزة الكهربائية .

الهدف الاولي من هذا المشروع هو اتقان لغة البرمجة من أجل الوصول الى انشاء تطبيق اندرويد قادر على إرسال أوامر من طرف مستخدم ، هذا التطبيق المثبت على هاتف الذكي الى أجهزة كهربائية . فهم مختلف بروتوكولات الاتصال و إستعمال المنفذ التسلسلي كمنفذ لإرسال البرنامج الى بطاقة الأردوينو . في هذا التطبيق قمنا بتحقيق جهاز إتصال عن بعد عن طريق تقنية البلوتوث من أجل تحقيق التحكم في المعدات المنازل الذكية ، عن طريق تطبيق أندرويد المثبت على الهاتف الذكي .

. **الكلمات المفتاحية :** تطبيق أندرويد ، بطاقة اردوينو ،المنفذات ،لغة البرمجة،بروتوكولات الاتصال ، بلوتوث .

Résumé :

Cette application nous permet de prolonger dans le monde de communication afin d'atteindre la création d'une application qui connecte le téléphone intelligent, la carte arduino et les actionneurs .

Le but de ce projet c'est de connaître la langue de programmation pour l'accès a crée une application Android, comprendre les protocoles de communication.

Dans cette application on a réalisé une appareil de communication via bluetooth afin de réaliser le contrôle de l'équipement programme d'installation Android sur votre smart phone .

Mots clés : application Androïde , la carte arduino. Les actionneurs .la langue de programmation .les protocoles du communication., Bluetooth

REFERENCES BIBLIOGRAPHIE

Bibliographie

[01]: site web « <http://blewando.dlinkddns.com/elv/Promo2018/th14/pag6.html> »

[02]: GUENAOUA L.« Commande en position du a MCC par Arduino ». , Mémoire de Master, UNIVERSITE BADJI MOKHTAR ANNABA 2017 .

[03]: BELKACEM IZALA M.A, BENEDDINE Kh, «Commande d'un bras ou un servomoteur à l'aide d'un Smartphone par électromyographie (EMG) »
Mémoire de Master, université de Tlemcen, 2016

[04]: M. TOURE Mohamed Lamine «COURS DE PORTEUS Professional » ,cours .

[05]: site web « <http://www.labcenter.com/> »

[06] : site web « sig.fgranotier.info/IMG/pdf/debuter_app_inventor»

[07]:siteweb«https://wapiti.telecomlille.fr/commun/ens/peda/options/st/rio/pub/exposes/expos_esrio2008/Garot-Ombaka/A2_archi.html»

[08]:Cerille Hebert, apprenez à programmer en java, livre de zéro 1er Edition .

[09]:[Daniel Pers](#) « Développer des applications Android avec App Inventor », cours ,Publié le [23 janvier 2013](#) .

Les Logiciels Utilisés :



Pour programmer Arduino .



Pour la simulation virtuelle de notre carte .



Pour l' Application APK .

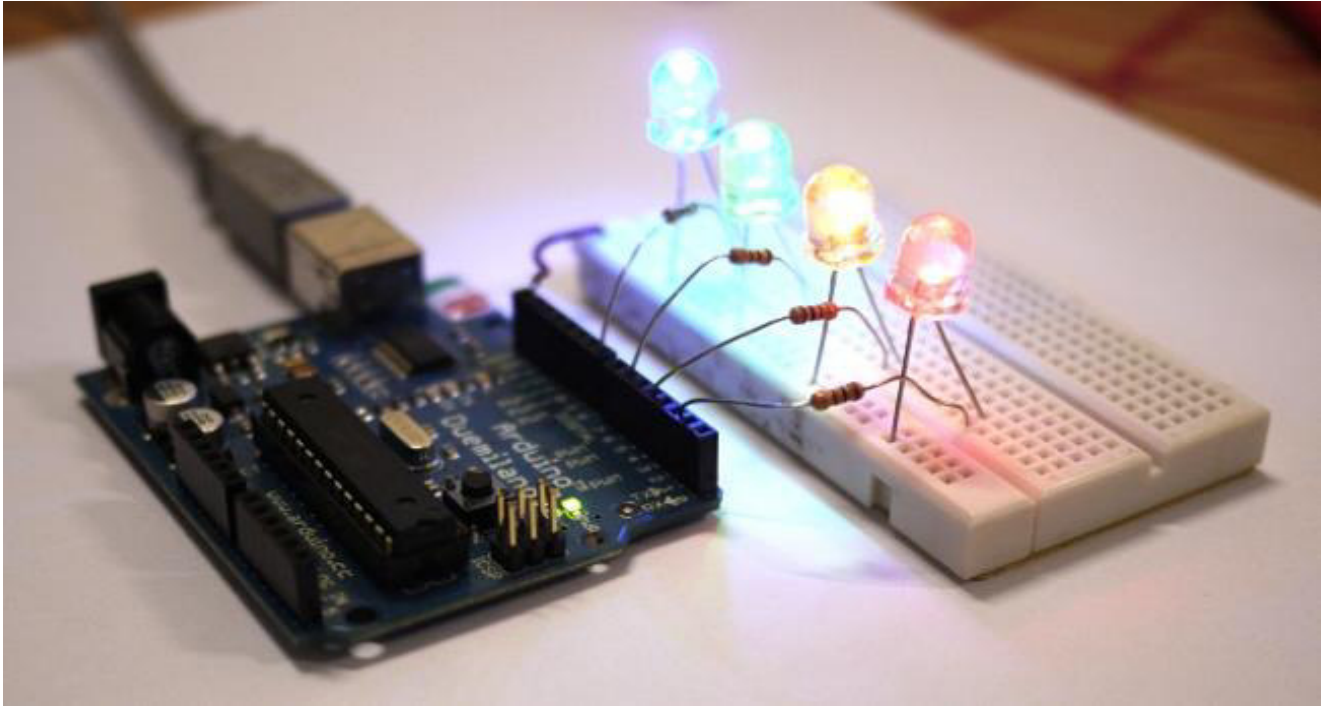


Photo de la réalisation pratique .



Annexe

Conclusion générale

1. Conclusion générale:

Mon projet consiste à réaliser un système de commande qui assure le contrôle de puissance a distance sans fil. Ce projet m'a été une source de découverte de plusieurs domaines rattachés à ma formation de base et une occasion précieuse pour approfondir nos connaissances. De plus, il m'a donné une idée sur la complémentarité entre la théorie et la pratique. En fait, ces résultats nécessité :

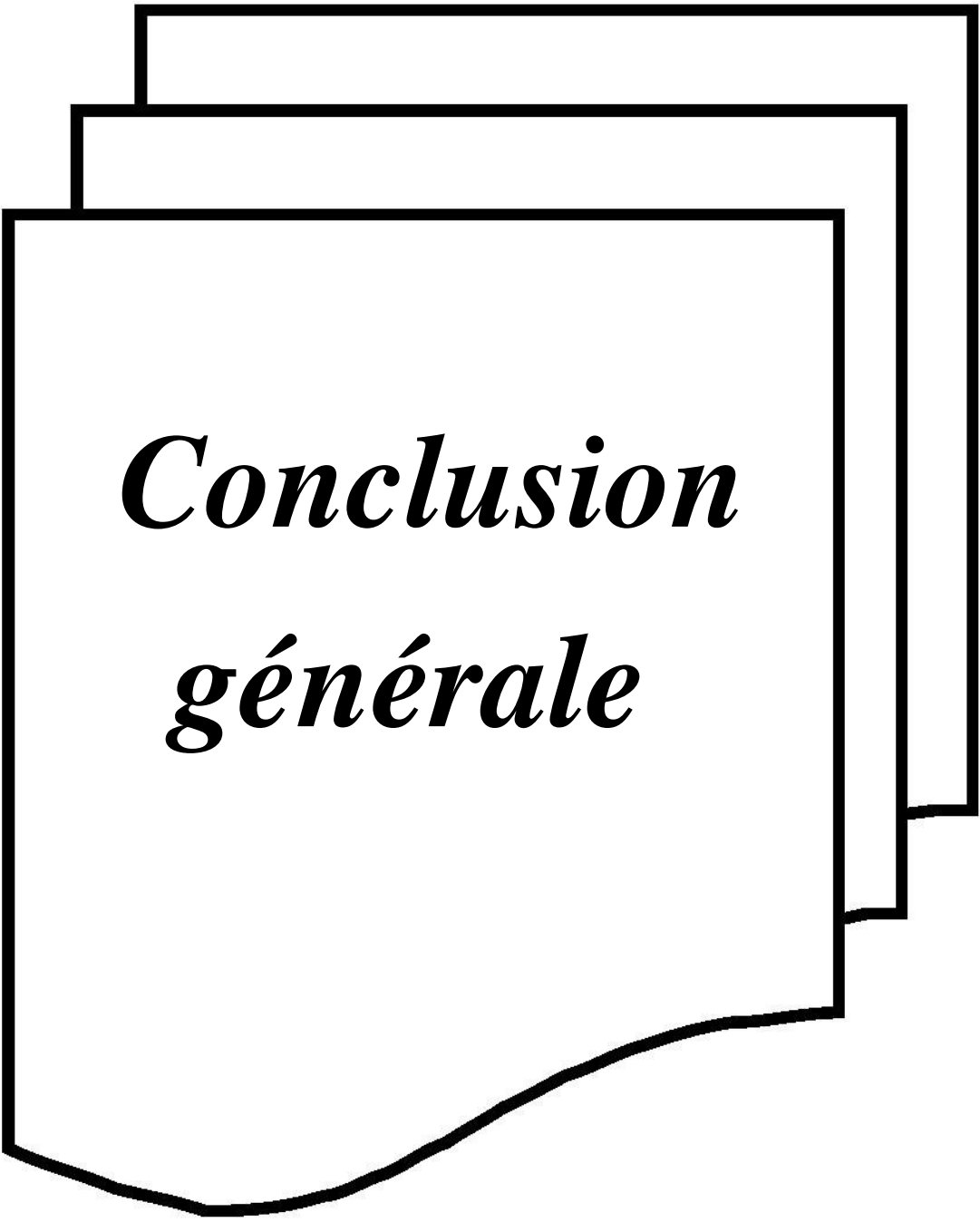
- en premier lieu la spécification des différentes tâches de mon sujet.
- En second lieu la conception de la partie matérielle et logicielle ainsi la recherche des solutions algorithmiques .
- en troisième lieu la réalisation de la carte finissant par la programmation du système.

Ce projet a été intéressant et récapitulatif de connaissances acquis et m'a permis de:

- Connaître plus des cartes d'interfaces électronique tel que : ARDUINO; BLEUTOOTH, SHIELD RELAY, ainsi la programmation des microcontrôleurs à travers IDE de CCS aussi bien L'APP INVENTOR pour la plateforme d'androïde.

- Accueillir de nouvelles idées concernant les autres projets de mes collègues.

Malgré les difficultés que j'ai rencontrées tout au long de ce travail, j'espère que j'ai préparé les champs nécessaires et l'intégration de mon système.



*Conclusion
générale*

III.1. Introduction :

Les systèmes d'exploitations sont nombreux et parmi eux on trouve l'arduino. Ce dernier consiste à développer les exigences du smart phone Dans les années passées on traite les informations par les ordinateurs on contre partie les téléphones intelligents font le même travail informatique.

La téléphonie mobile a connu une explosion dans les années 2000 mais aucune révolution n'a semblé arriver depuis que les appareils se ressemblent. Les innovations n'avaient plus vraiment de saveur ; les applications étaient difficiles d'accès de par leur mode de distribution et souvent peu performantes à cause des faibles capacités des appareils.

Depuis quelques années, les smart phones sont dotés d'une puissance plus importante et d'espaces de stockages conséquents. Les téléphones tendent à devenir des objets artistiques, presque de reconnaissance sociale, et possèdent des fonctionnalités qu'aucun téléphone ne pouvait espérer auparavant: connexion haut débit, localisation GPS, boussole, accéléromètre, écran tactile souvent multipoint, marché d'applications en ligne. Autant de qualités permettant de créer des applications innovantes et de les distribuer en toute simplicité.

La plate-forme Androïde apporte tout cela au consommateur, mais surtout, elle affranchit le développeur de nombreuses contraintes. Par son ouverture; elle permet à n'importe quel développeur de créer ses applications avec un ticket d'entrée quasi nul.

Le Framework et le système d'exploitation et outils associés ont un code source ouvert, leur accès est gratuit et illimité. Plus besoin de négocier avec le constructeur du téléphone pour qu'il vous laisse développer sur sa plate-forme. Tous les développeurs sont ainsi sur un même pied d'égalité, tous peuvent ajouter de la mobilité à des applications existantes.

Cette partie de notre étude ne nous donnera pas de bons résultats si on néglige certains paramètres ; donc le bon fonctionnement de notre système se base essentiellement sur une bonne démarche et une bonne réflexion de notre programme.

III.2. Les types de programmation :

Du premier chapitre et du second, on peut combiner que notre réalisation software a besoin de deux étapes : la première consiste à un programme qui va s'injecter aux microcontrôleurs de la carte Arduino après avoir été convertie par le IDE en code HEX et la deuxième est une plateforme qui sous App Inventor qui devra s'installé sous smart phone.

III.2.1. Présentation de l'organigramme IDE:

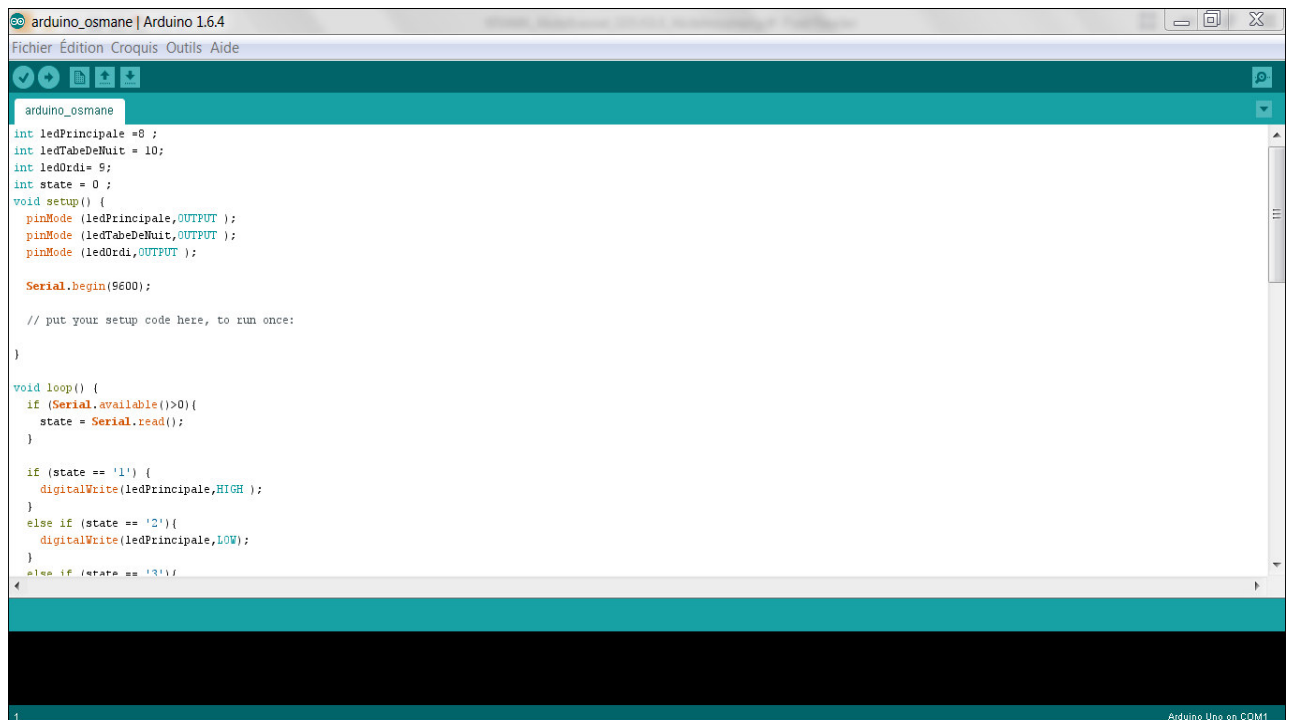
L'organigramme est une représentation schématique des liens fonctionnels, organisationnels et hiérarchiques d'un organigramme, « un programme »,

III.2.1.1. l'environnement de la programmation:

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois ,le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carteàtravers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino .

III.2.1.2. Présentation du programme IDE:

Nous avons préféré d'illustré l'interface du IDE de l'environnement Arduino dans lequel nous avons simulé notre programme.



```
arduino_osmane | Arduino 1.6.4
Fichier Édition Croquis Outils Aide

arduino_osmane

int ledPrincipale = 8 ;
int ledTabDeNuit = 10;
int led0rdi = 9;
int state = 0 ;
void setup() {
  pinMode (ledPrincipale, OUTPUT );
  pinMode (ledTabDeNuit, OUTPUT );
  pinMode (led0rdi, OUTPUT );

  Serial.begin(9600);

  // put your setup code here, to run once:
}

void loop() {
  if (Serial.available()>0){
    state = Serial.read();
  }

  if (state == '1') {
    digitalWrite(ledPrincipale,HIGH );
  }
  else if (state == '2'){
    digitalWrite(ledPrincipale,LOW);
  }
  else if (state == '3'){

```

Figure III. 1: Présentation graphique du programme

III.3. Présentation Application Android :

Android est l'OS (système d'exploitation) mobile de Google. L'OS mobile de Google a rencontré un tel succès qu'il se vend aujourd'hui beaucoup plus de Smartphones Android que d'IPhones. De nombreux constructeurs produisent des terminaux Android. Un succès qui doit beaucoup aux nombres d'applications disponibles et à la forte intégration des mobiles avec les services Google .L'interface des Smartphones Android est très épurée, claire, et facile à utiliser .

III.3.1. Les Versions d'Android :

Android a développé une douzaine de version depuis sa création en 2008

le tableau (III.2) récapitule les versions d' Andriod .

Version	Nom	API	Date de sortie
1.0	Apple Pie		11/11/2007
1.1	Bananas Split		22/10/2008
1.5	Cupcake		30/04/2009
1.6	Donut		15/09/2009
2.0	Eclaire	7	26/10/2009
2.2	Foryo	8	20/05/2010
2.3.3-2.3.7	Gingerbread	10	06/12/2010
4.0.3-4.0.4	Ice cream sandwich	15	19/10/2011
4.1.X	Jelly Bean	16	09/07/2012
4.2.X	Jelly Bean	17	13/11/2012
4.3	Jelly Bean	18	24/07/2013
4.4	KitKat	19	31/10/2013
5.0	Lollipop	21	03/11/2014
5.1	Lollipop	22	09/03/2015
6.0	Marshmallow	23	28/11/2015

Tableau III .1 : Les versions Andriod

III..3.2. Le système d'exploitation Android :

Androïde est un système d'exploitation développé initialement pour les Smart phones. Il utilise un noyau Linux qui est un système d'exploitation libre pour PC et intègre tous les utilitaires et les périphériques nécessaires à un smart phone. Il est optimisé pour les outils Gmail. Aussi, l'androïde est libre et gratuit et a été ainsi rapidement adopté par des fabricants.

La société Androïde a été rachetée en 2007 par Google. Mais aujourd'hui, l'Androïde est utilisé dans de nombreux appareils mobiles (smart phones). Les applications sont exécutées par un processeur de type ARM à travers un interpréteur JAVA. En plus de cela, l'androïde concurrence l'opérateur système d'Apple qu'il tend à dépasser en nombre d'utilisateurs. Androïde évolue pour mieux gérer l'hétérogénéité des appareils qu'il utilise.[06].

Android est en fait un système de la famille des Linux. Cet OS s'appuie sur :

- un noyau Linux (et ses drivers)
- une couche d'abstraction pour l'accès aux capteurs (HAL)
- une machine virtuelle : Dalvik Virtual Machine (avant Lollipop)

- un compilateur de bytecode vers le natif Android Runtime (pour Lollipop)
- des applications (navigateur, gestion des contacts, application de téléphonie...)

III.3.3. L'architecture de la plateforme d'Android :

L'architecture de la plateforme Android se décline selon une démarche bottom up en quatre principaux niveaux que sont le noyau Linux, les bibliothèques et la plateforme d'exécution, le module de développement d'applicatifs et enfin les différentes applications .[07]

Chacun de ces niveaux est décrit plus en détail ci-dessous.

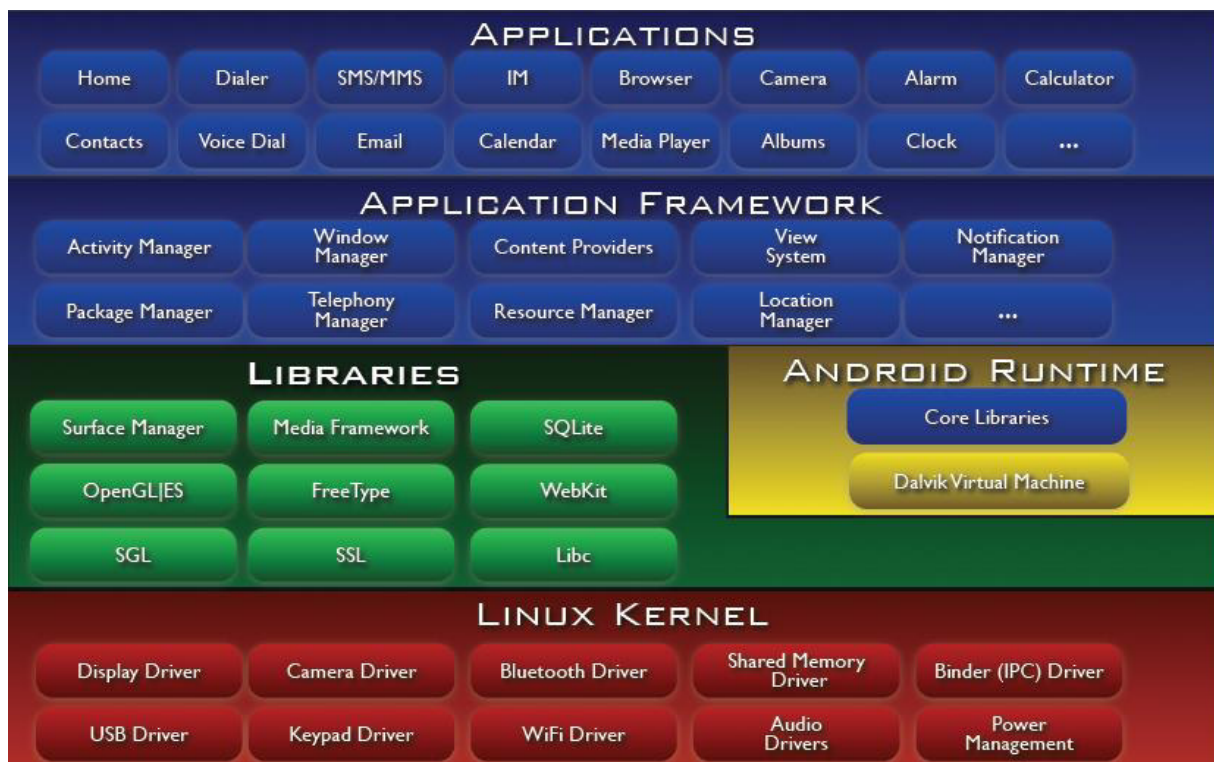


Figure III.2 :L'architecture de la plateforme d'Android

III.4. Les avantages d'Android :

- **Open source :**
Le contrat de licence pour Android respecte les principes de l'open source, c'est -à-dire qu'on peut à tout moment Télécharger les sources et les modifier.
- **Gratuit (ou presque):**
Android est gratuit, autant pour l'utilisateur que pour les constructeurs.
Les développeurs sous Android peuvent poster leurs applications développées sur le Play store (apk sur Android) Et cela coutera 25\$, ces 25 \$ permettent de publier autant d'applications souhaitées, à vie.

- **Facile à développer :**

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale.

- **Facile à vendre :**

Le Play Store (anciennement Android Market) est une plateforme immense et très visitée ; c'est donc une mine d'opportunités pour quiconque possède une idée originale ou utile.

- **Flexible :**

Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. Les Smartphones, les tablettes, la présence ou l'absence de clavier ou de trackball, différents processeurs... On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android ! Non seulement c'est une immense chance d'avoir autant d'opportunités, mais en plus Android est construit de manière à faciliter le développement et la distribution en fonction des composants en présence dans le terminal.

III.5. Explication du programme :

Le but de ce programme est de réaliser des actions « caractères » codées par des lettres ou par des chiffres émis de l'Arduino vers le smart phone .

On commence par la programmation du microcontrôleur situé sur l'Arduino ; au début, on déclare toutes les variables utilisées après avoir initialisé les ports de l'Arduino comme des entrées/sorties « OUTPUTS », ensuite ; on appelle le sous programme de temporisation déjà situé sous le compilateur IDE.

III.6. App Inventor pour Android :

Est une application développée par Google. Elle est actuellement entretenue par le Massachusetts Institute of Technology (MIT). [08]

Elle simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont pas familiers avec les langages de programmation. Elle est basée sur une interface entièrement graphique en absence totale de ligne de code, elle est particulièrement adaptée à l'initiation à la programmation sous Android.

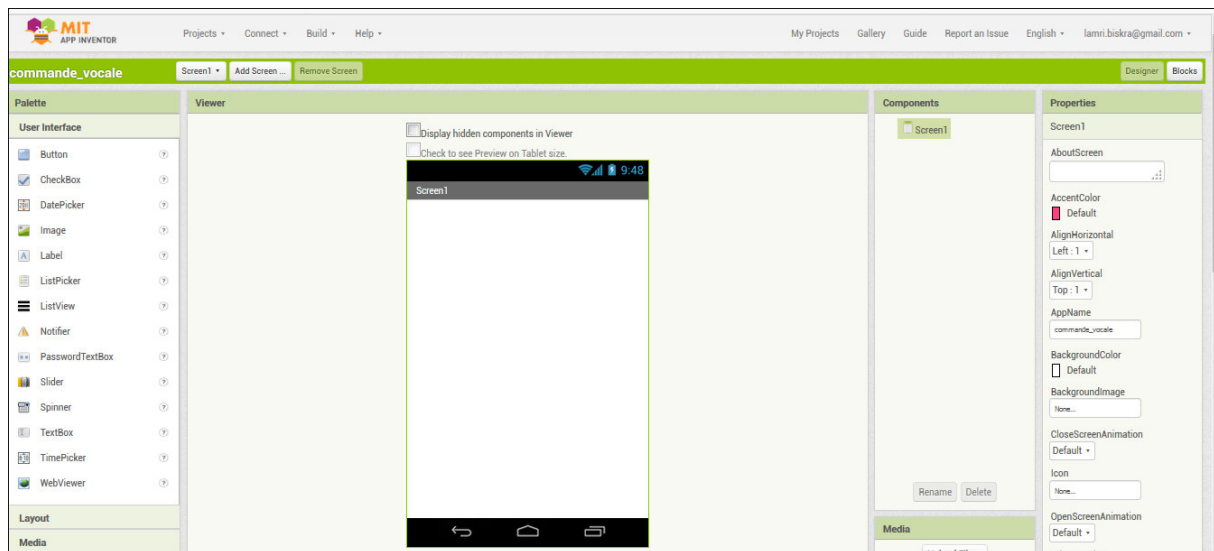


Figure III .3: l'espace du travail sur l'application

III.6.1. L'outil App Inventor :

App Inventor est un outil de développement des applications en ligne pour les smart phones sous androïde et permet à chacun de créer son application personnelle pour le système d'exploitation Androïde qui est développée par Google.

La plateforme de développement est offerte à tous les utilisateurs possédant un compte Gmail. Elle rappelle certains langages de programmation simplifiés basés sur une interface graphique similaire à Scratch. Les informations des applications sont stockées sur des serveurs distants.[06] .

Elles sont actuellement entretenues par le Massachusetts Institute of Technology (MIT).

L'environnement de App Inventor contient trois fenêtres qui sont proposées pendant le développement :

- Une pour la création de l'interface homme machine : permet de créer l'allure de l'application (App Inventor Designer) .
- Une pour la programmation par elle-même : elle permet, par l'assemblage des blocs de créer le comportement de l'application (App Inventor Block Editor) .
- Une pour l'émulateur : qui permet de remplacer un terminal réel pour vérifier le bon fonctionnement du programme.

La connexion d'un terminal réel sous Androïde permettra ensuite d'y télécharger le programme pour un test réel. Ce terminal pourra aussi bien être un téléphone qu'une tablette ; le comportement du programme sera identique. [06]

N.B : Scratch est un environnement graphique permettant aux débutants de s'initier à la programmation. On y programme le comportement de lutins animés à l'aide de briques visuelles qui s'emboîtent comme des Legos par Drag and Drop.

III.7. Historique de logiciel App Inventor :

2009 : Début du développement du logiciel App Inventor par Google à partir de recherches dans l'enseignement de l'informatique faites par le MIT, Boston près de New - York. L'Objectif de l'enseignement permet à des étudiants qui débutent en informatique d'apprendre à programmer sans se noyer sous le code Java.

- 2011: Google rend App Inventor open source. Le MIT poursuit le développement.
- 2012: Version bêta d' App Inventor diffusé par le MIT encore en version bêta aujourd'hui . [09]

III.8. Langage JAVA :

JAVA est un langage de programmation orienté objet, développé par Sun Microsystems, sorti en 1995. Sun Microsystems est racheté en 2009 par Oracle, une application écrite en JAVA et facilement portable sur plusieurs systèmes d'exploitation.

Une application exécutable sous Androïde (interprétable par une interface en JAVA) est un fichier avec l'extension « APK ».

III.9. Un commencement avec App Inventor :

Google fournit gratuitement un kit de développement (SDK) prévu pour s'intégrer (sous la forme d'un Plugin) à l'environnement de développement Eclipse (libre). Il permet de développer des applications codées en langage Java pour les différentes versions d'Androïde.

1. Se connecter à Internet.
2. Ouvrir notre navigateur et se connecter au compte Google.

3. Se connecter au site Internet d'App Inventor du MIT : Cela peut-être assez long parfois car App Inventor est lancée depuis le Web en mode Cloud .

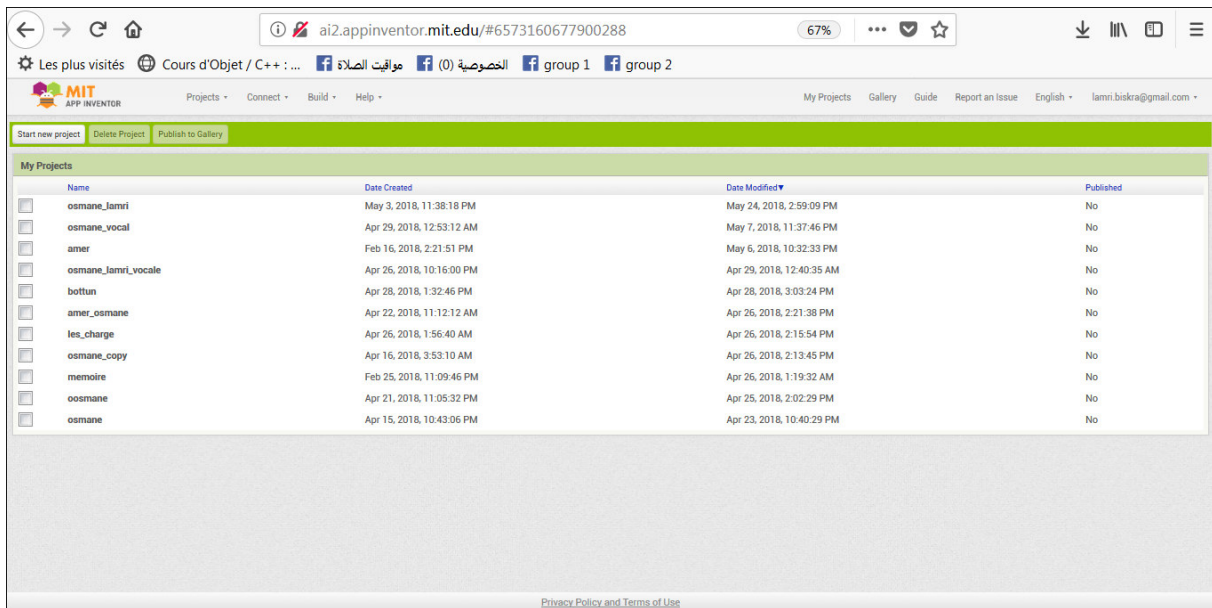


Figure III.4 : interface App Inventor

4. Créer un nouveau projet :

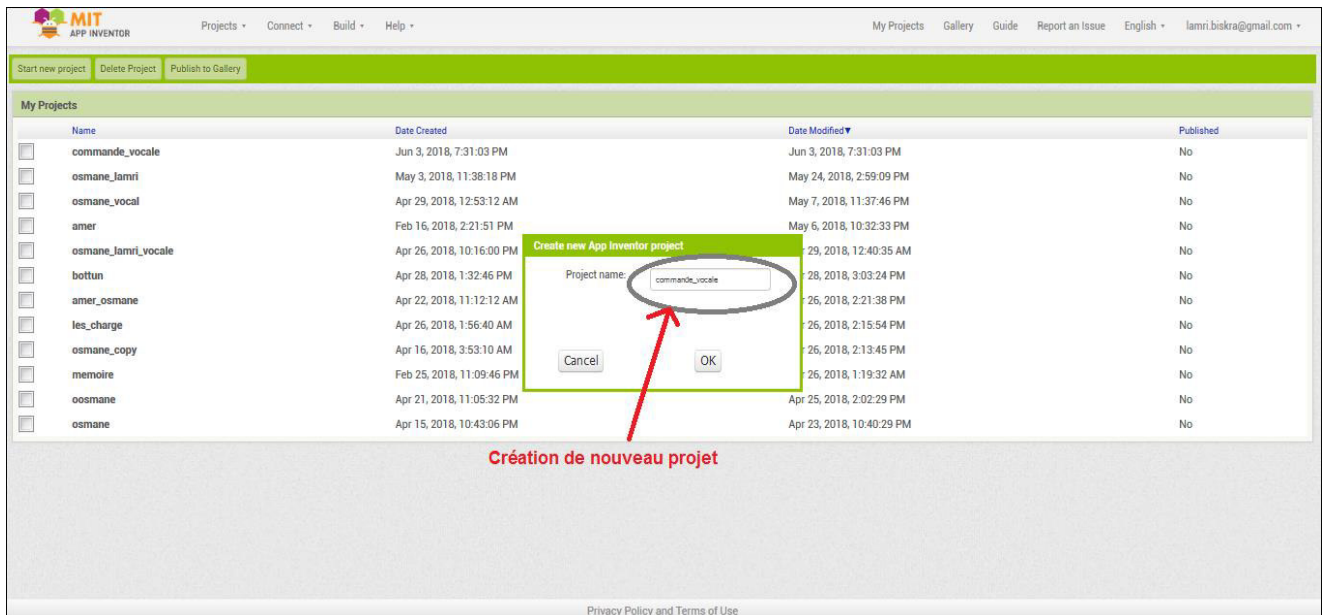


Figure III. 5 : Création de nouveau projet sur App Inventor

- ✓ Donc le projet est sauvegardé automatiquement sur notre compte Google. (Dans le Cloud).

III.10. Structure d'une application App Inventor :

Une application développée sous App Inventor est constituée de deux parties distinctes mais étroitement liées.

III.10.1. L'interface graphique :

Pour créer l'application sous App Inventor l'interface graphique contient nos propriétés (taille, couleurs, position, textes Ets).

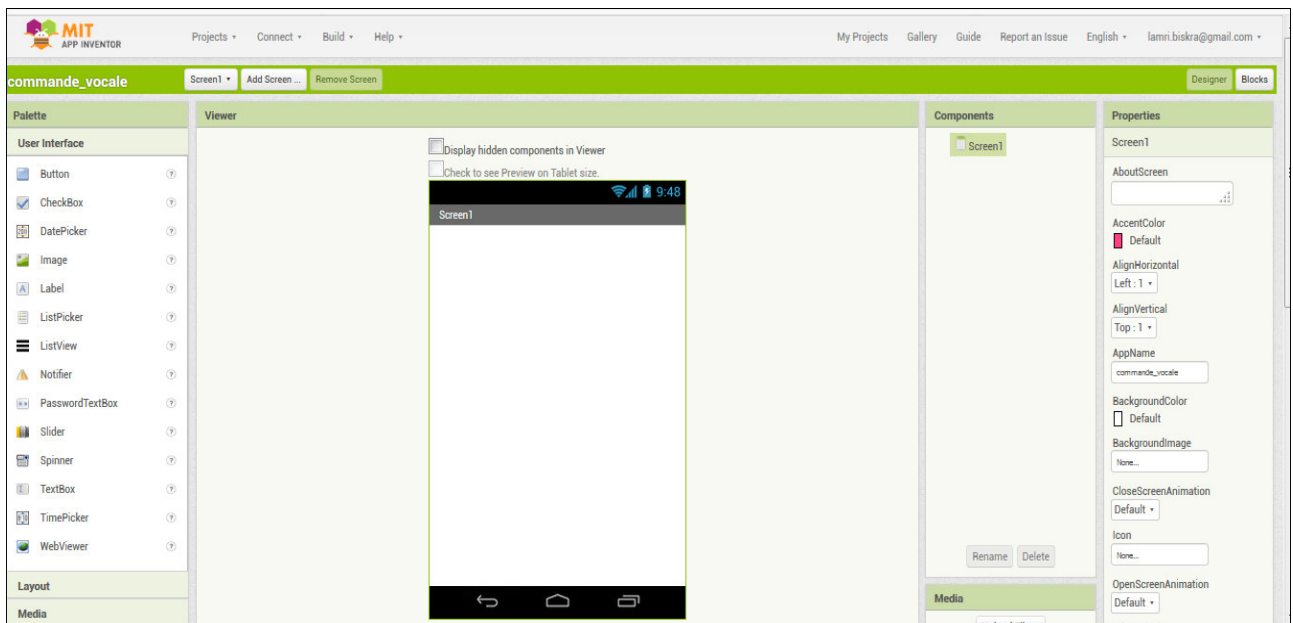


Figure III.6 : Première interface de la création App Inventor

Cette partie contient des composants (visibles ou invisibles). Cette interface graphique contient quatre parties :

- **Partie 1 :**

Une palette sous App Inventor contenant tous les éléments qui peuvent être positionnés sur l'écran du téléphone.

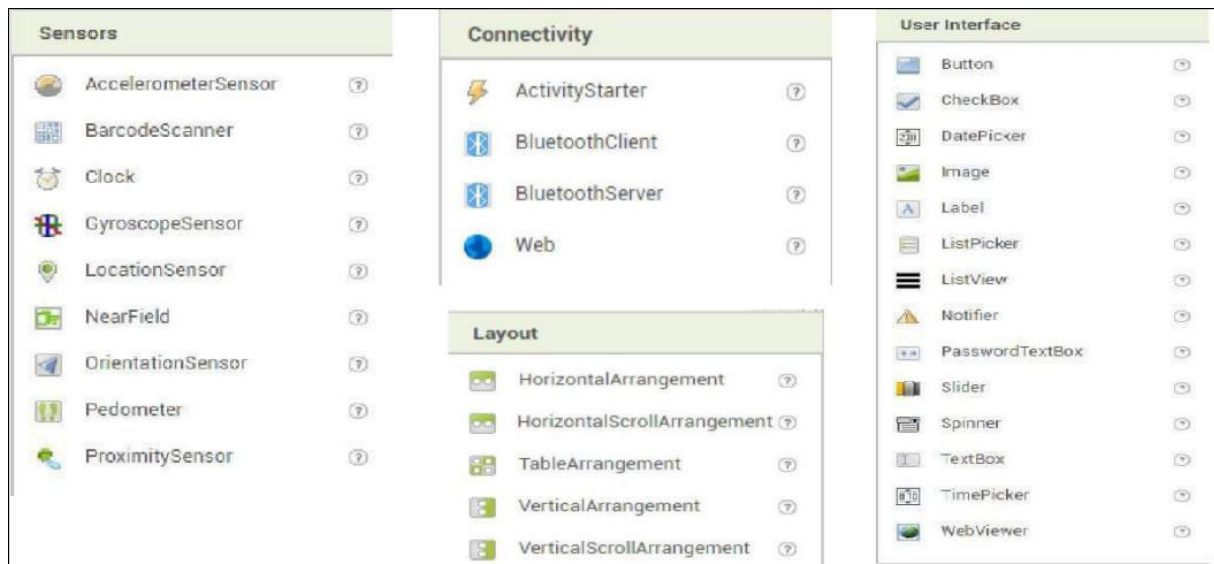


Figure III.7 : Les composants graphiques .

- **Partie 2 :**

C'est la surface du téléphone ajusté automatiquement par App inventor ou manuellement par nous même en utilisant le composant « Screen arrangement ».

- **Partie 3 :**

La liste des éléments et des médias utilisés sur l'écran.

- **Partie 4 :**

Les propriétés des différents éléments utilisés par exemple la couleur et la taille du bouton ou texture.

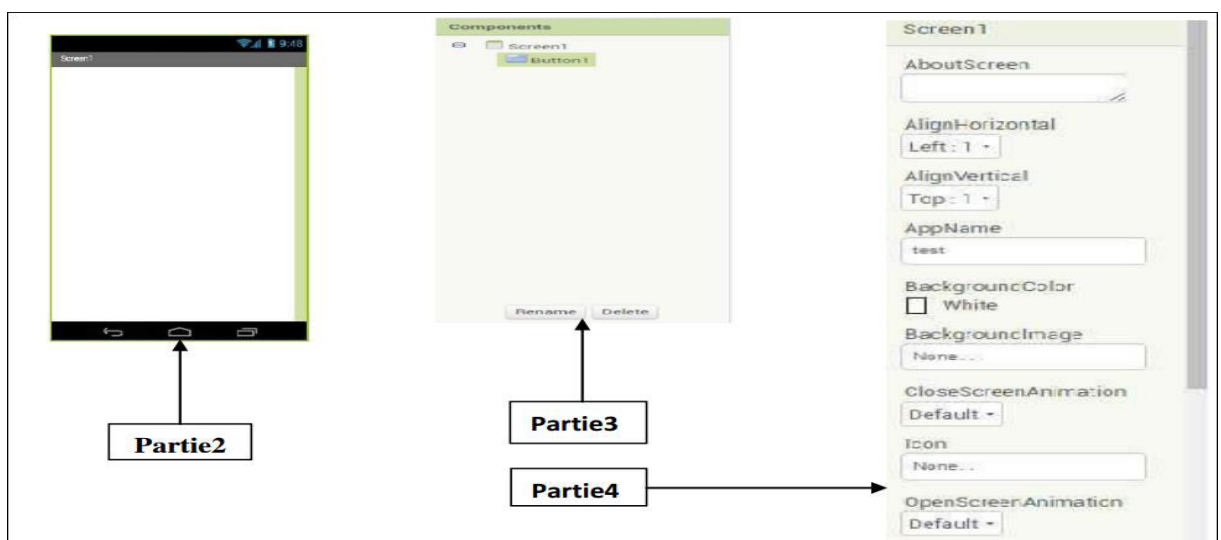


Figure III.8 : Les propriétés et les classes sous App Inventor .

Pour ajouter un bouton sur l'écran d'émulateur virtuel, on clique sans lâcher sur le mot « Bouton » dans la palette en déplaçant la souris sur Viewer et relâcher le bouton et là un nouveau bouton va apparaître sur le Viewer.

Pour ajuster la place du bouton, on utilise la propriété Width de notre exemple; on utilise le paramètre Fill parent.

Pour utiliser le bouton sur la largeur de l'écran, on sélectionne les nombres de pixels del'écran 50 pixels. De ces opérations, on obtient l'écran suivant :

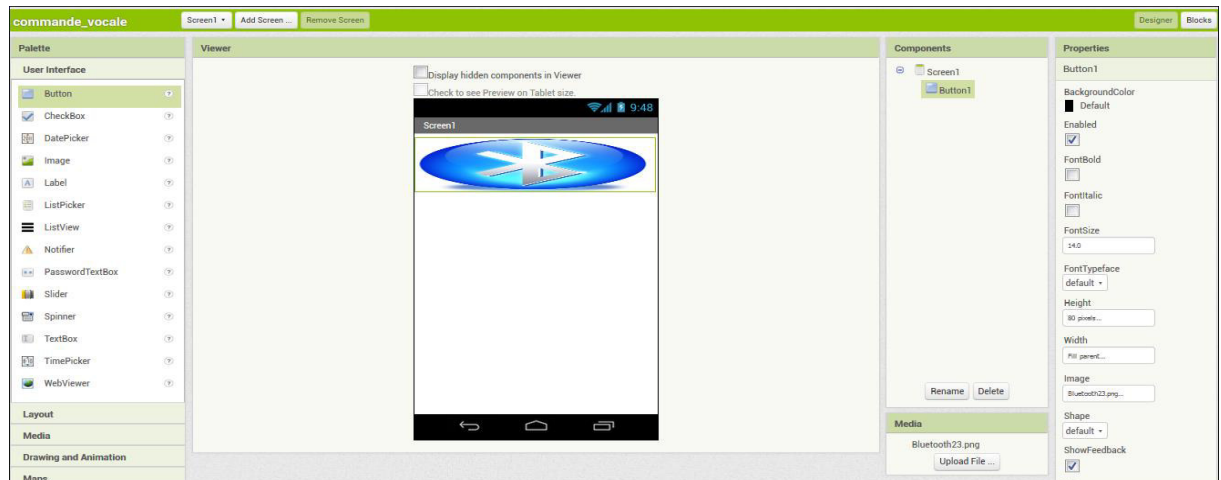


Figure III.9 : Ajustement d'un bouton sous App Inventor .

III.11. Editeur de blocs (Fenêtre Scratch):

Une fois les composants de l'écran de téléphone mis en place et désigné, nous passons à la deuxième phase de développement d'une application via App Inventor : l'interface Scratch, pour cela, il faut cliquer sur «Open the Blocks Editor» en haut à droite de la page.

L'interface Scratch permet d'imbriquer des éléments graphiques entre eux pour effectuer la partie programmation de l'application à développer. De cette partie, on peut assembler les différents blocs de l'application et indiquer comment les composants doivent se comporter et qui s'affichent dans l'émulateur virtuel (par exemple ; ce qui se produit quand un utilisateur clique un bouton) déterminant le fonctionnement même de l'application, en réaction à des événements (internes ou externes) ou à des réponses.[06]

Pour ouvrir ce Block, on a cliqué en haut à gauche sur « Open the Blocks Editor ».

App Inventor, lancé le Blocks Editor depuis le serveur du MIT et proposé l'exécution du fichier App Inventor For Androïde Code blocks, « d'un format *.Jnlp ». Après téléchargement, JAVA ouvre la fenêtre du Block Editor sur votre PC (cela peut être assez long).

N.B Jnlp :est un format de fichier associé à la technologie JAVA web start. Il s'agit de pouvoir déployer facilement des applications Java à partir d'un simple navigateur web,

Comme indiqué sur la figure (III.10) App Inventor, éditeur de blocks, cette interface est très simple et épurée. En effet, en haut ; on retrouve des éléments classiques «Save», «Undo» et «Redo» ainsi qu'un bouton de test pour lancer l'application sur le mobile ou sur l'émulateur. Cette interface contient des parties Built-in et blocks.



Figure III.10 : En-tête d'éditeur de blocks App Inventor .

La fenêtre à gauche contient les blocs qui assemblent la partie droite de la fenêtre pour décrire le comportement de notre application. Les blocs peuvent être standards (dans l'onglet« Built-in ») ou définis spécifiquement pour l'application (dans l'onglet « My Blocks »).

Aussi, la corbeille est utilisée pour jeter des morceaux de pseudo-code et la loupe sert à changer la taille l'écran de l'éditeur.

Chaque élément contient des blocks : (ces pages décrivent les blocs que nous pouvons utiliser lorsque nous construisons nos applications App Inventor.

Dans l'onglet « Built-in », nous retrouvons toujours les mêmes éléments :

- **Définition** : morceaux permettant de définir des procédures (avec/sans résultats/attributs).
- **Texte** : morceaux permettant de traiter le texte. Assimilables au type char et à la classe String en JAVA.
- **Lists** : morceaux permettant de traiter des listes. Assimilables aux sous-classes de List en JAVA.
- **Math** : morceaux permettant de traiter des nombres. Assimilables au type int et à la classe Integer en JAVA.

- **Logic** : morceaux permettant de traiter des booléens. Assimilables au type boolean et à la classe Boolean en JAVA.
- **control** : outils permettant d'effectuer de la programmation conditionnelle par exemple : dans l'élément logic .

Lorsqu' on clique sur «My Blocks» en haut et à droite de la page : nous obtenons la figure suivante :

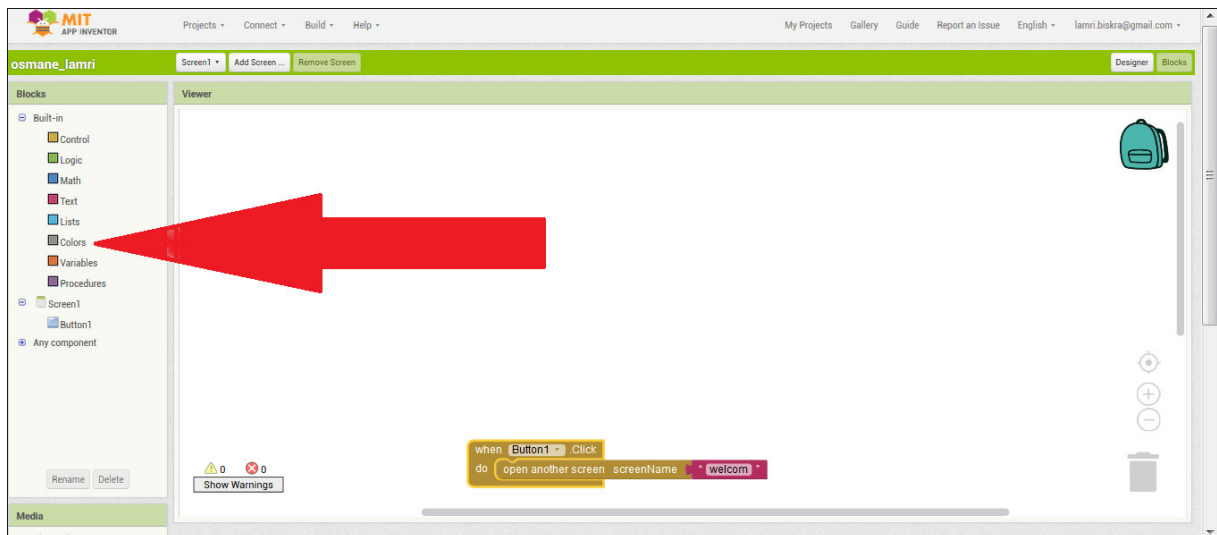


Figure III.11 : Editeurs de blocks App Inventor .

La palette des variables et fonctions est à gauche, l'onglet My blocks propose les fonctions associées aux éléments déposés sur notre écran au préalable .

Dans l'onglet «My Blocks», on retrouvera les éléments et leurs accesseurs et fonctions :

- **My Definitions** : variables et procédures globales.
- **Button1** : variables et procédures spécifiques au bouton.
- **Label1** : variables et procédures spécifiques au label.
- **Orientation Sensor1** : variables et procédures spécifiques au capteur.
- **Screen1** : variables et procédures spécifiques à l'écran (assez restreint).

Lorsque on Clique sur l'élément « Button1 » on obtient la figure suivante est cela comme échantillon pour n'importe quel composant associé dans notre application.

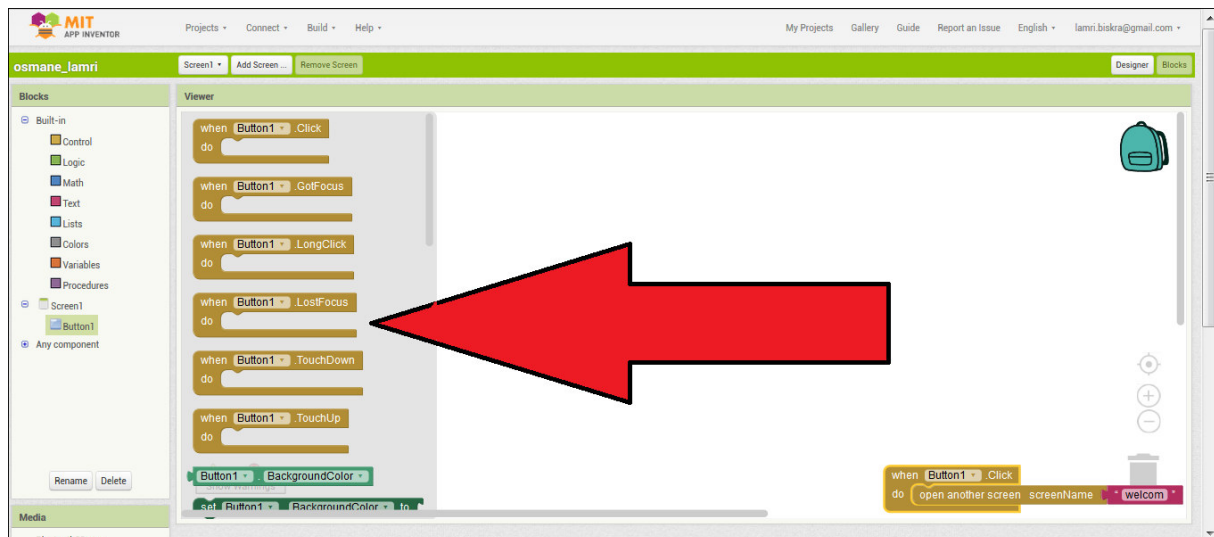


Figure III.12 : Echantillon d'un composant sous App Inventor .

A partir de l'onglet «Package for phone» ce test assure trois solutions accessibles :

- 1) L'émulateur : un écran s'affichera sur l'ordinateur.
- 2) la connexion se fera directement sur le smart phone en wifi.
- 3) USB : la connexion se fera sur le smart phone via un câble USB.

Pour installer notre application sur un appareil Androïde, on exécute une des trois solutions à partir du bouton "Package for Phone" (en haut à droite de l'interface web).

On résume notre application pour cette phase « Scratch » par la figure qui suit.

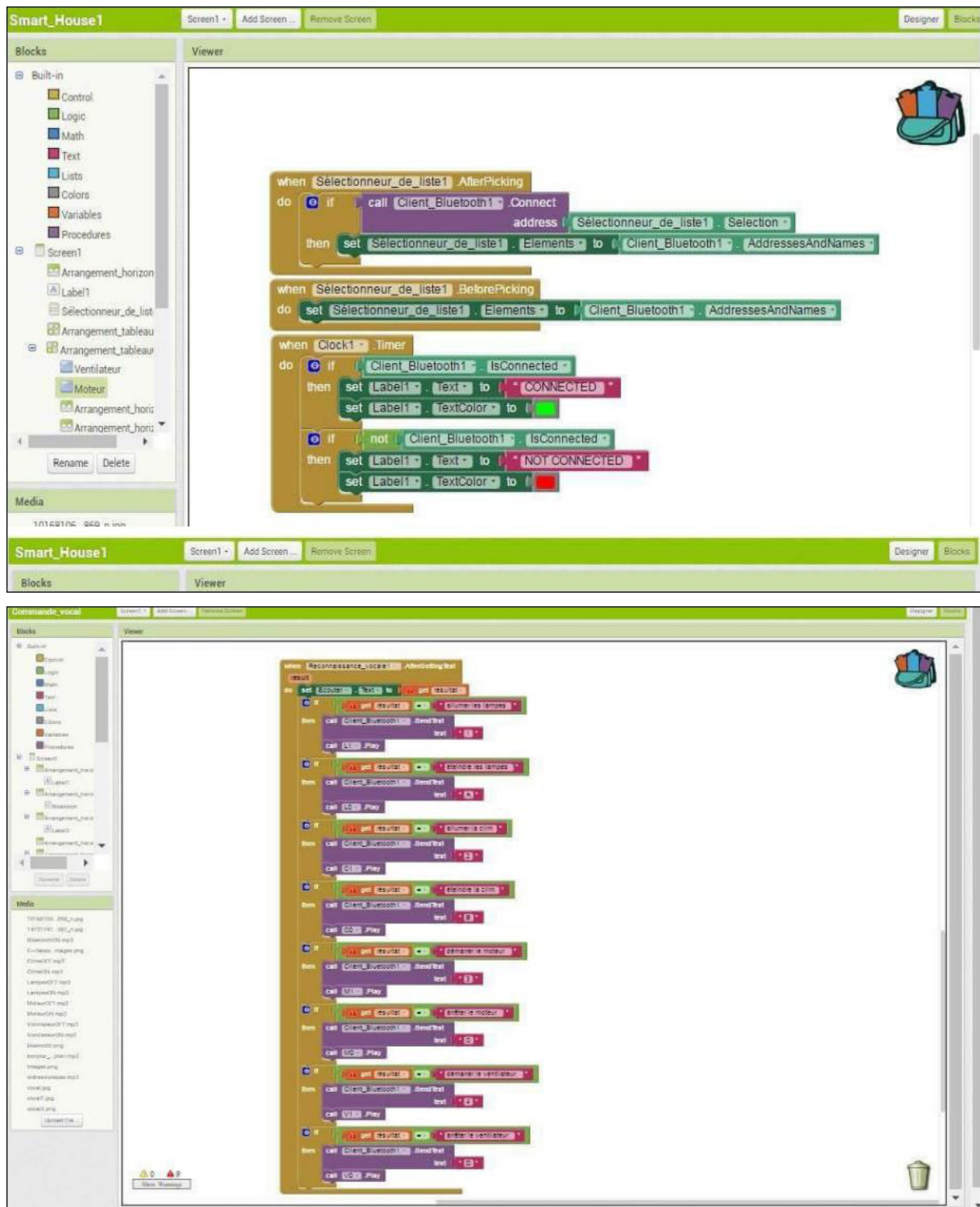


Figure III.13 : Schéma global du Scratch .

III.12. Explication et démarche d' application :

Nous allons présenter les principales interfaces graphiques de notre application

III.12.1. L'interface Commande vocale :

Notre interface contient une touche d'activation et désactivation du Bluetooth il faut d'abord s'assurer que le Bluetooth est connecté .

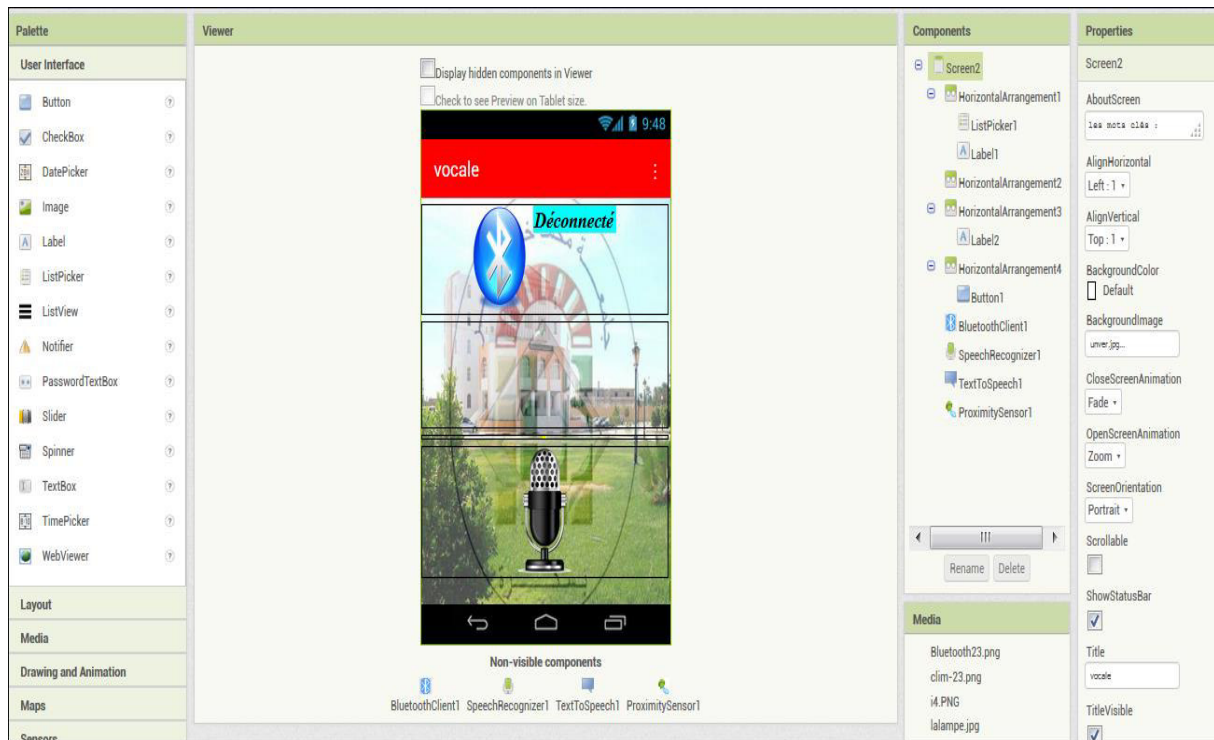


Figure III.14 : Interface de l'application vocale.

III.12.2. L'interface Commande Tactile des charges:

Notre interface contient une touche d'activation et désactivation du Bluetooth donc avant d'envoyer n'importe quel caractère à travers les boutons « ON », il faut d'abord s'assurer que le Bluetooth est connecté ; un mot de passe pour raison de protection a été programmé afin d'exécuter un ordre quelconque. Cette figure montre que notre application nommée « Smart House » peut être installée sur smart phone, sa taille est de deux mégas comme première version.

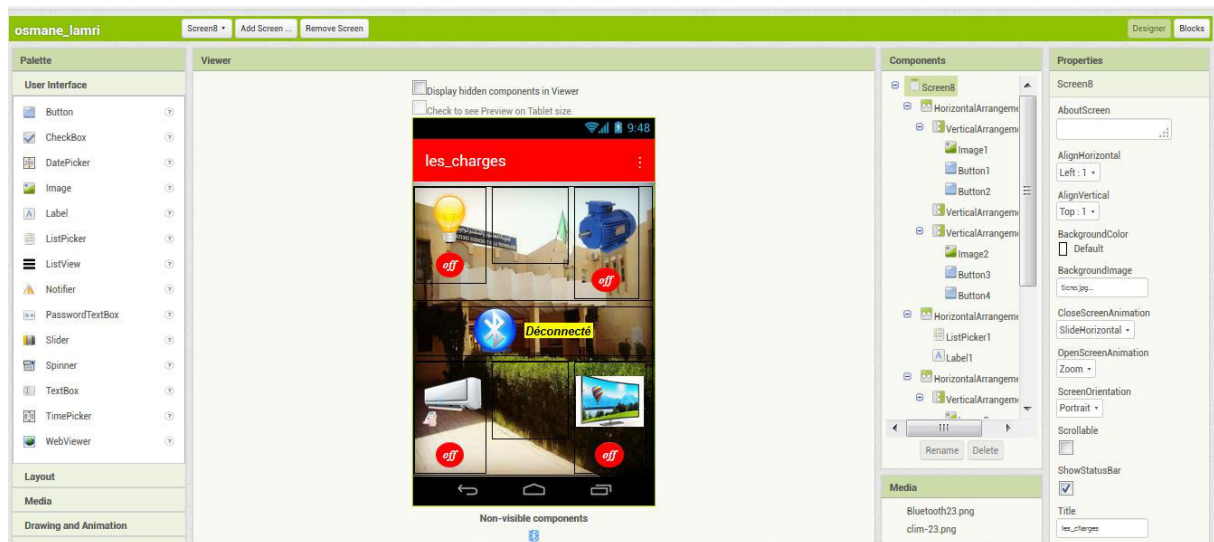


Figure III.15 : Interface de l'application Tactile .

III.12.3. L'interface Application :

Dans l'interface d'accueil, il s'affiche le bouton :

«commande des charges 220v par la reconnaissance vocale sous système ».

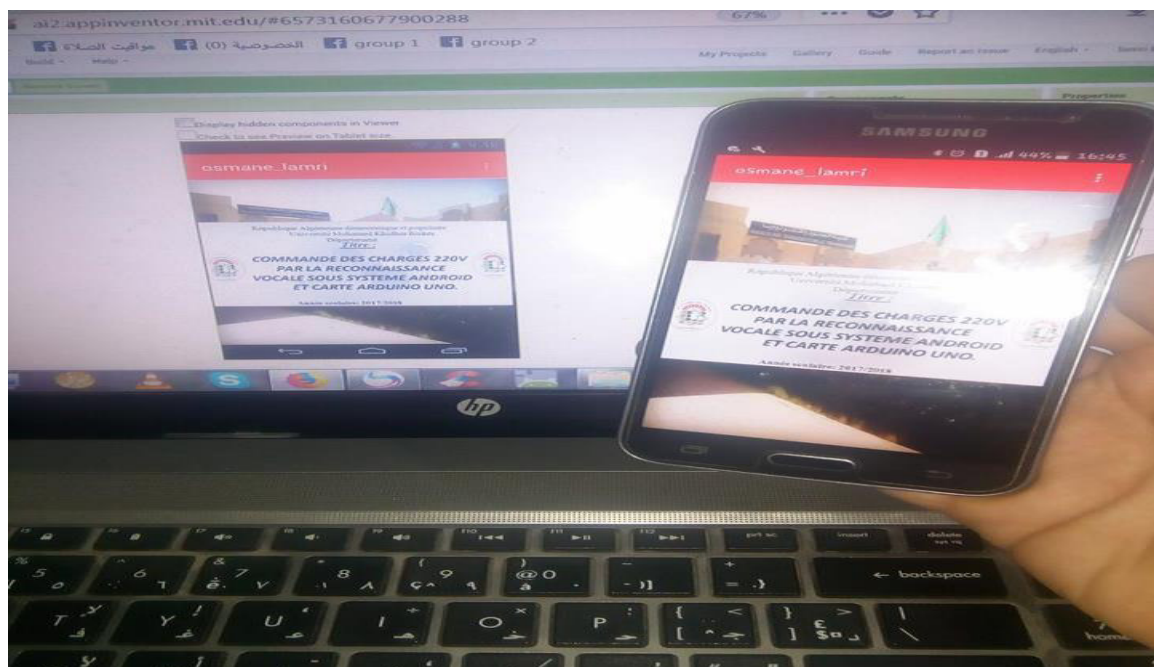
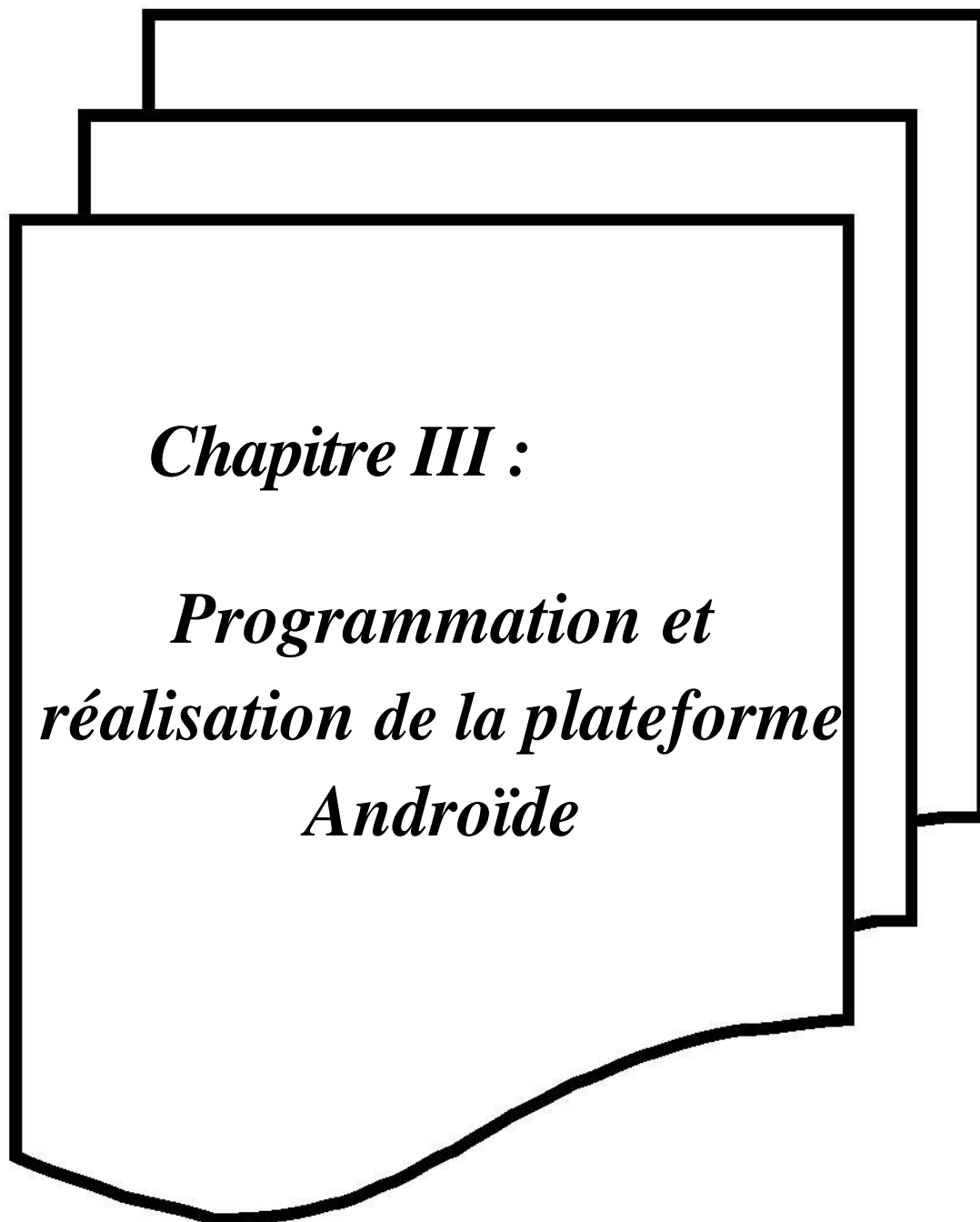


Figure III.16 : L'application Andriod du smart house .

III.13. Conclusion :

Sur le plan pratique, une manipulation adéquate du logiciel « Arduino » nous permet alors d'utiliser un compilateur ; il s'agit de « IDE » ce dernier possède une capacité de créer un code HEX, qui peut être injecté sur son microcontrôleur d'une part ; et d'autre part une simple élaboration de l'environnement App Inventor qui donne naissance à une application «apk » sous smart phone capable de lier une carte Arduino et un smart phone afin d'exécuter des ordres bien définis.

On peut conclure que les résultats obtenus sont satisfaisants compte tenu des limitations du matériel et des moyens dont nous disposons.



Chapitre III :

*Programmation et
réalisation de la plateforme
Androïde*



Chapitre II :

*Programmation et
simulation du projet*

I.1.Introduction :

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. On parle aussi de système embarquée ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes en ayant l'envie : ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation. On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

I.2. Définition du module Arduino :

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino .[02]

I.3. La carte Arduino:

La carte Arduino Uno est une carte microcontrôleur basée sur un Atmega328. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie

PWM) ,6 entrées analogiques, un cristal a 16 MHz, une connexion USB une prise jack d'alimentation, un en-tête ICSP et un bouton reset. Plusieurs cartes Arduino existent et qui se différencient par la puissance du microcontrôleur ou par la taille et la consommation de la carte. Le choix du type de carte Arduino s'effectue en fonction des besoins de votre projet. La carte Arduino UNO est la carte la plus couramment utilisée .

I.4.Les différents type d'Arduino :

I.4.1. La carte Arduino UNO :

C'est la carte idéale pour découvrir l'environnement ARDUINO. Elle permet à tout débutant de se lancer dans tous ses premiers petits projets. Comme c'est la carte la plus utilisée, il est très facile de se référer aux tutoriels très nombreux sur le net et ainsi de ne pas rester seul dans son exploration.

Sa simplicité devient par contre un handicap lorsqu'il s'agit de multiplier les périphériques, de manipuler des algorithmes lourds ou d'interagir avec les OS Android pour lesquels d'autres cartes arduino sont plus adaptées.[1]

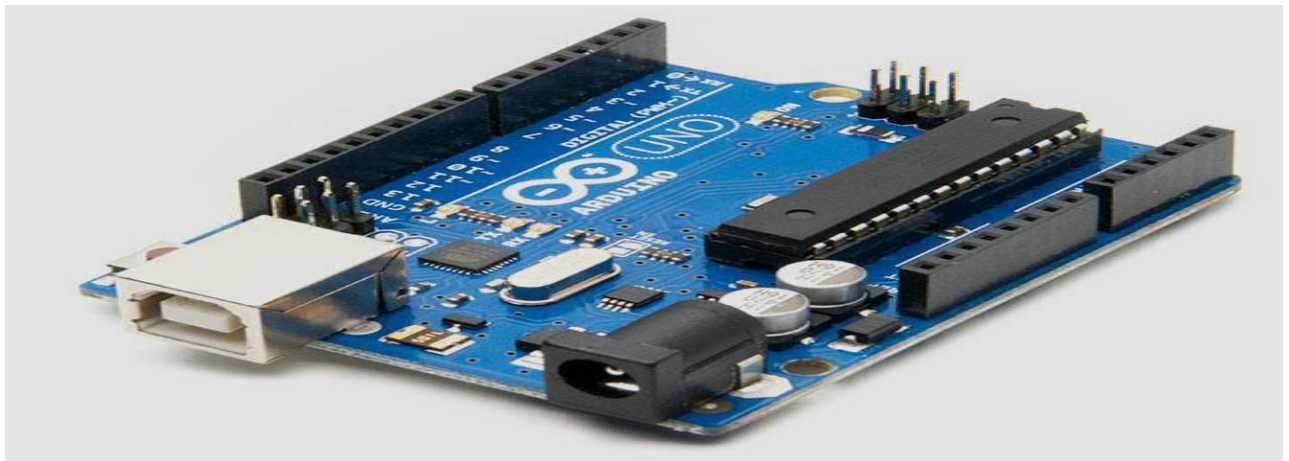


Figure I.1 : La carte Arduino Uno.

I.4.2. La carte Arduino Leonardo :

C'est la carte qui est prévue pour succéder à la carte Arduino Uno en présentant des caractéristiques équivalentes mais une ergonomie revue et une stabilité plus éprouvée. Sa diffusion moins importante limite le support utilisateur disponible sur le net . [1]

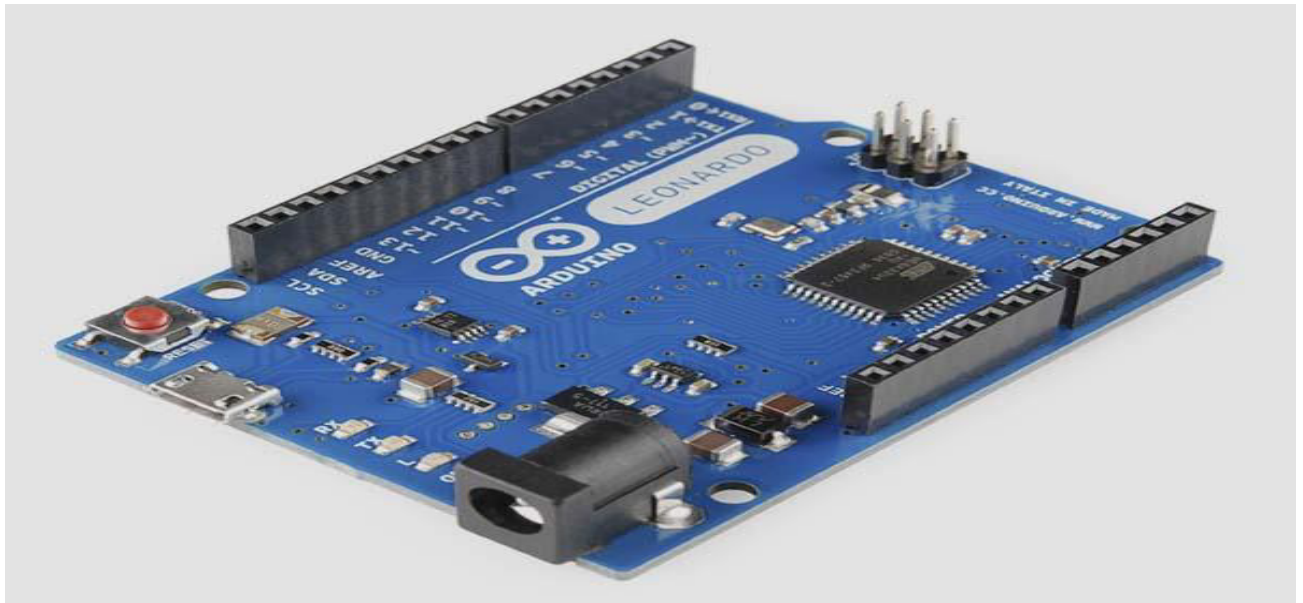


Figure I.2 : La carte Arduino Lenardo.

I .4.3. La carte Arduino Mega

La carte Arduino Mega est la carte la plus diffusée après la carte Arduino Uno. Elle offre un nombre d'entrées/sorties beaucoup plus important (54 contre 14), un processeur plus puissant doté d'une mémoire plus vaste qui permet d'exploiter des algorithmes plus complexes. [1]

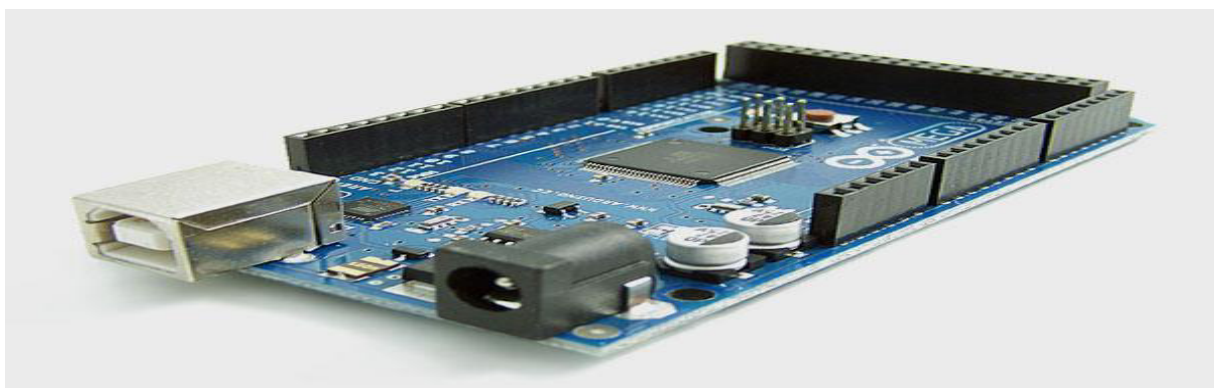


Figure I.3 : La carte Arduino Mega.

I.4.4. La carte Arduino Mega ADK :

La carte Arduino mega ADK offre les mêmes caractéristiques techniques que la carte Arduino mega mais son port USB permet de la connecter avec un environnement Android ouvrant de nouvelles perspectives d'interaction avec le monde des smartphones et des capteurs dont ils sont dotés. Sa mise en oeuvre nécessite par contre de solides connaissances en Java et la capacité à développer ses propres applications. [1]

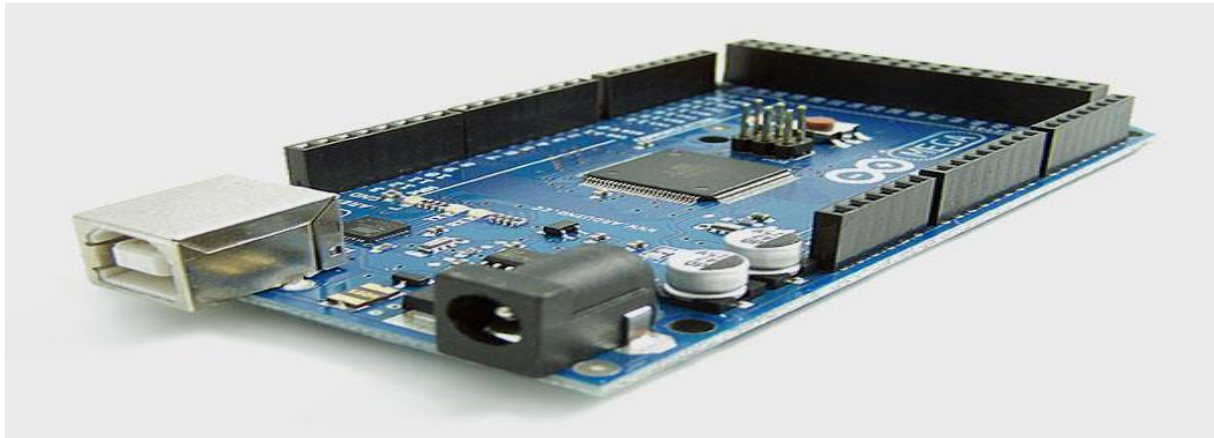


Figure I.4 : La carte Arduino Mega ADK.

I.4.5. La carte Arduino Due :

La carte Arduino Due est une évolution de la carte Arduino Mega et offre des performances réputées 3 fois supérieures. Elle permet de manipuler rapidement des algorithmes lourds particulièrement utiles dans le monde de la robotique par exemple. [1]

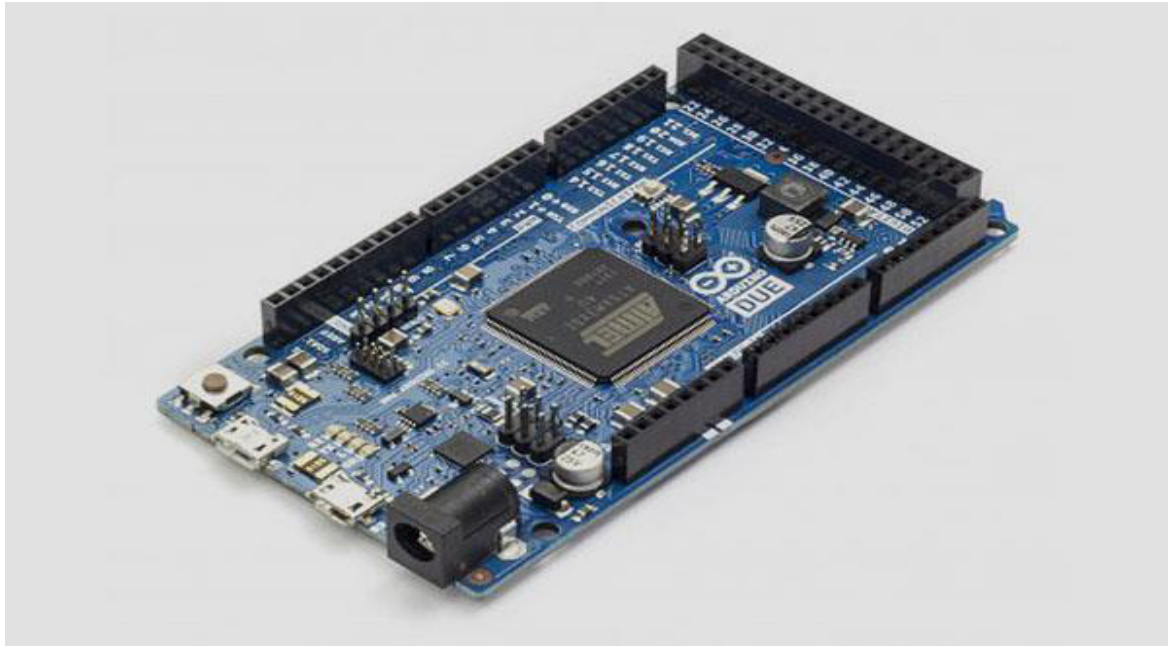


Figure I.5 : La carte Arduino DUE.

I.4.6. La carte Arduino Nano :

La carte Arduino nano n'est ni plus ni moins qu'une carte Arduino uno miniaturisée. Sa taille et son poids réduits la destinent à une utilisation dans des espaces réduits (en textile par exemple) ou dans des applications de robotique ou de modélisme pour lesquels le poids et la taille sont des facteurs déterminant (hélicoptères, drones...) [1]

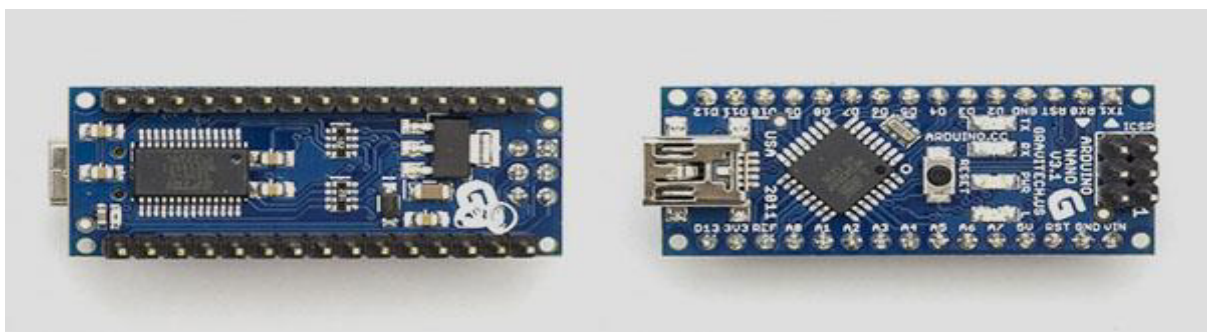


Figure I.6 : La carte Arduino NANO.

I.4.7. La carte Arduino Mini Pro :

La carte arduino Mini Pro est une carte Arduino Uno simplifiée à l'extrême permettant néanmoins de piloter de petits projets ou certains éléments d'un projet. Attention, cette carte n'intègre pas de port USB ce qui rends sa connectivité délicate. [1]

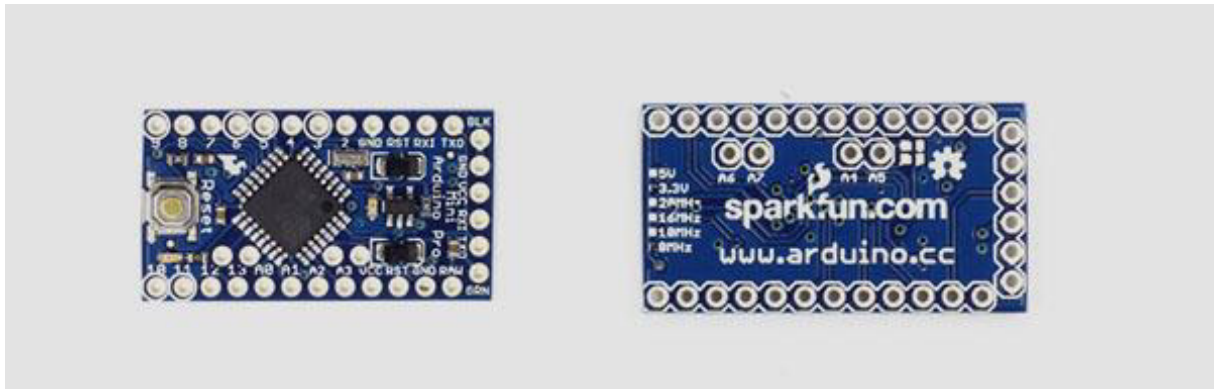


Figure I.7 : La carte Arduino Mini Pro.

I.4.8. La carte Arduino Yun :

La carte Arduino Yun, récemment proposée par Arduino, est conçue pour contrer les avantages de la carte Raspberry. Elle est un dérivé de la carte Leonardo et a pour objectif de combiner la puissance de Linux avec la facilité d'utilisation d'une carte Arduino. Elle est également la première carte Arduino à être dotée nativement d'un wifi intégré. [1]



Figure I.8 : La carte Arduino YUN.

I.5. Spécifications techniques de la carte Arduino UNO :**I.5.1. Caractéristiques de la carte arduino :**

- Microcontrôleur A Tmega328
- Voltage d'utilisation 5 V
- Voltage en entrée (recommandé) 7-12 V
- Voltage en entrée (limites) 6-20 V
- 14 connecteurs digitaux d'entrées/sortie (6pouvant être utilisées en tant que sorties PWM)
- 6 connecteurs analogiques en entrée
- courant direct par connecteur a 3.3 V : 50 mA
- mémoire flash 32 Ko dont 0.5 Ko est utilisé par le boot loader
- SRAM 2Ko
- EEPROM 1 Ko
- Vitesse d'horloge 16 MHz
- d'un bouton de réinitialisation (reset)
- D'un connecteur d'alimentation jack
- D'un connecteur ICSP (programmation " in-circuit").

I.5.2. Alimentation :

La carte Arduino Uno peut-être alimentée soit via la connexion USB (qui fournit 5v jusqu'a 500 mA) ou a laide d une alimentation externe. La source d alimentation est sélectionnée automatiquement par la carte. L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V a 12V sous 500 mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecte en branchant peut une prise 2.1 mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles

ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

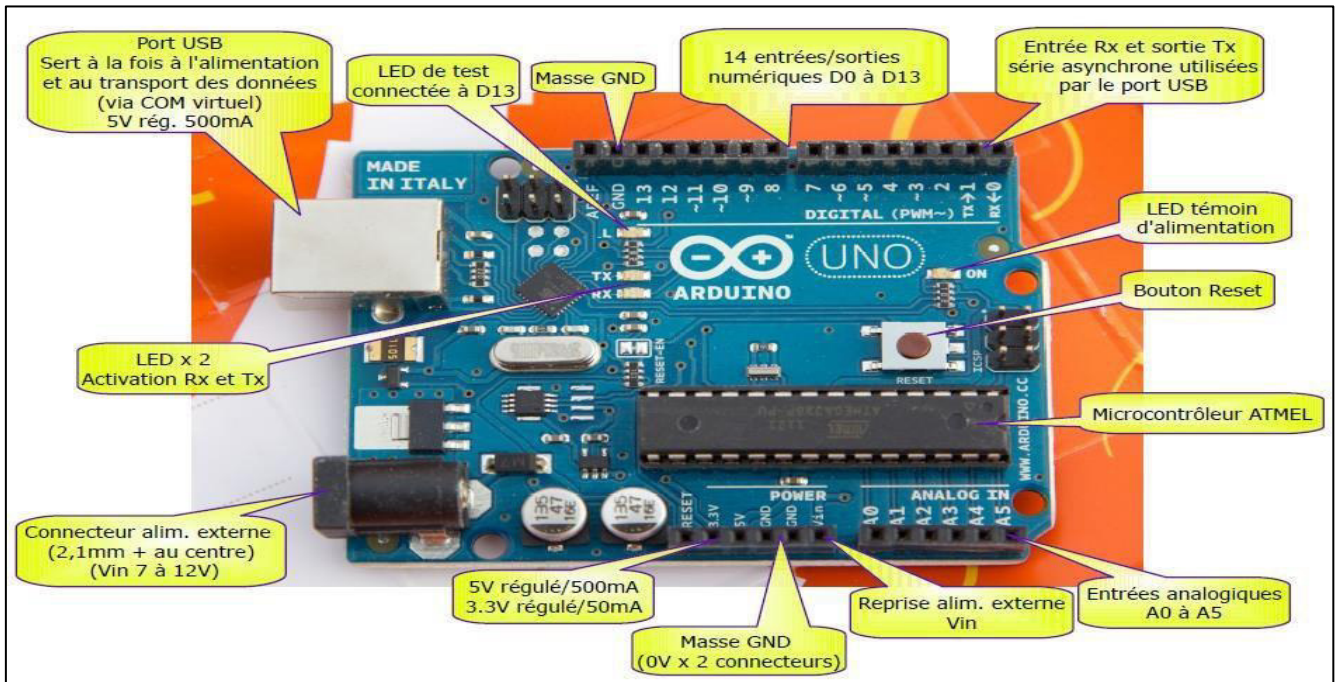


Figure I.9 : Constitution de la carte Arduino UNO

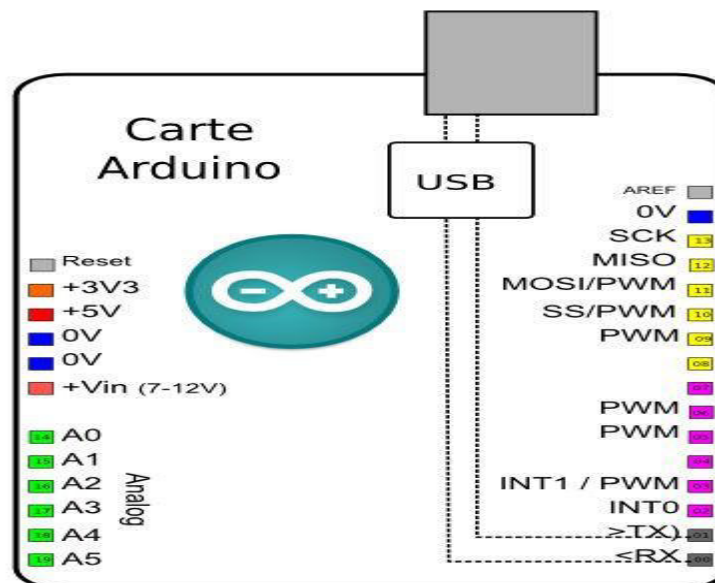


Figure I.10 : Brochage de la carte Arduino Uno

I.5.3. Mémoire :

L'ATmega 328 a 32 Ko de mémoire FLASH pour stocker le programme (dont 0.5Ko également utilisés par le boot loader). L'ATmega 328 a également 2ko de mémoire SRAM (volatile) et 1Ko de EPROM (non volatile –mémoire qui peut lue à l'aide de la librairie EEPROM).

Le boot loader est un programme préprogrammé une fois pour toute dans l'ATmega et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

I.5.4. Entrées et sorties numériques :

Chacune des 14 broches numériques de la carte UNO (numérotées de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pin Mode, digitalWrite du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne (pull-up) déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite(broche, HIGH).

- **Remarque :**

Les broches analogiques peuvent être utilisées en tant que broches numériques : Elles sont numérotées en tant que broches numériques de 14 à 19.

I.6. Avantages de la carte Arduino :

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit:

- **Prix (réduits) :** les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes.

- **Multi plateforme** : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- **Environnement de programmation clair et simple** : l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **Logiciel Open Source et extensible** : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).
- **Matériel extensible** : les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût.

I.7. Logiciel :

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateformes, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module) .Il est également possible de se passer de l'interface Arduino, et de compiler les programmes via l'interface en ligne de commande.

Le langage de programmation utilisé est le C++, compile avec `avg -g++`, et lie à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aussi le développement de programmes sur les plates-formes Arduino, à toute personne maîtrisant le C ou le C++ .

I.7.1. Communication :

La carte Arduino Uno dispose de toute une série de facilité pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega 328 dispose d'une UART (universel asynchrones Receveur transmetteur ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX) . Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparait comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega328utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire. Cependant, sous Windows, un fichier, inf. est requis.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simple depuis et vers la carte Arduino. Les LED RX et TX sur la carte clignotent lorsque les donnée sont transmises via le circuit intégré USB-vers-série et la connexion USB vers l'ordinateur (mais communication série sur les broches 0 et 1).

Une librairie Série Logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO.

L'ATmega 328 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface " 2 fils ")et SPI :

- Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C.
- Pour utiliser la communication SPI (Interface Série périphérique), la librairie pour communication SPI est disponible.

I.7.2. Programmer Arduino :

On présente ici le langage Arduino, son vocabulaire ainsi que la structuration d'un programme écrit en **#c**.

I.7.3. La structure d'un programme :

Un programme Arduino comporte trois parties :

- 1- La partie déclaration des variables (optionnelle).
- 2- La partie initialisation et configuration des entrées / sorties : la fonction setup.
- 3- La partie principale qui s'exécute en boucle : la fonction loop.

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino.

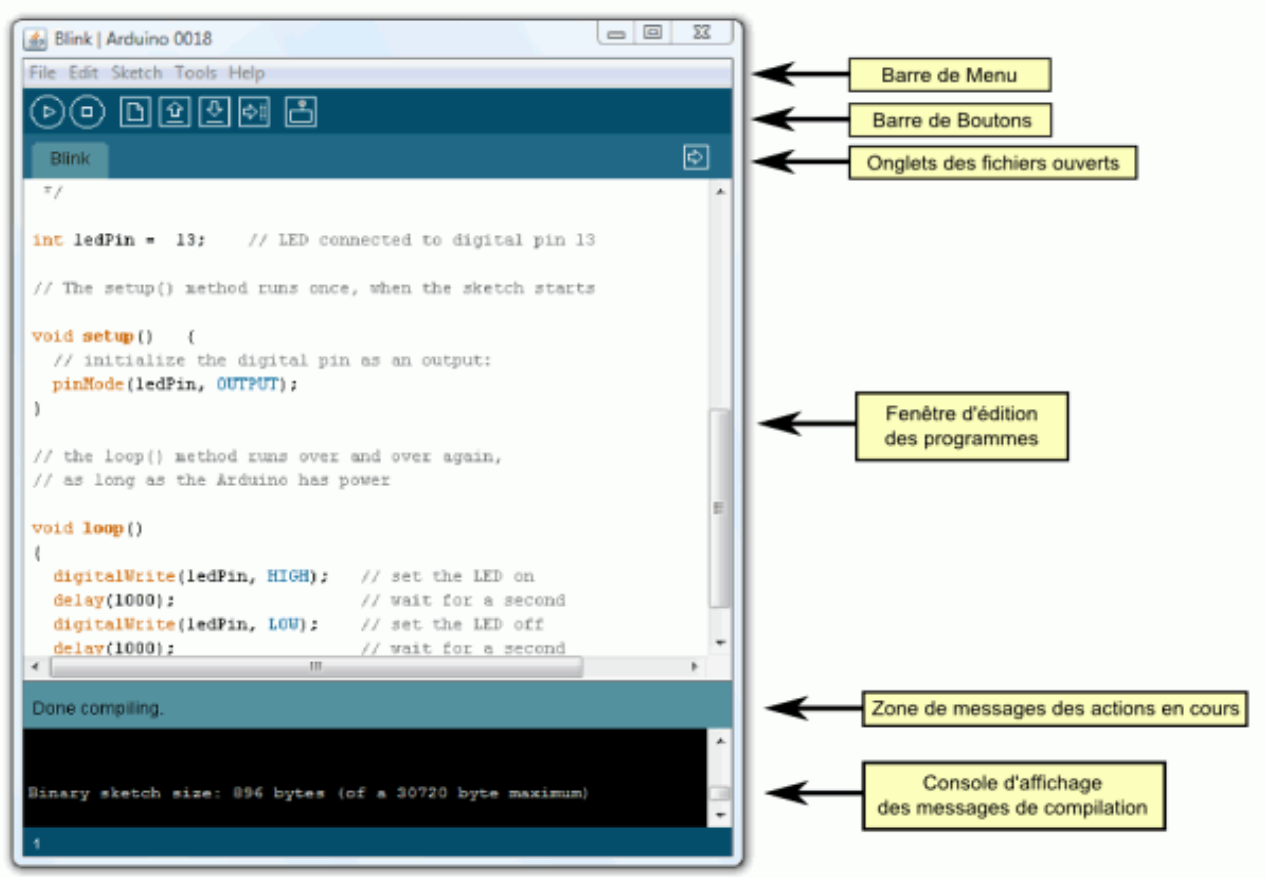


Figure I.11 : Interface IDE Arduino

I.8. Présentation du module Bluetooth (HC-06) :

Le module Bluetooth HC-06 est un accessoire indispensable si l'on souhaite communiquer sans fil (avec par exemple un Smartphone doté d'un Bluetooth) avec une carte Arduino. Le module HC-06 est différent du module HC-05. C'est un module « esclave » contrairement au HC-05 qui est un module « maître ». On ne pourra donc pas faire communiquer deux Arduino en utilisant deux modules HC-06 mais par contre, on peut très bien établir une liaison Bluetooth entre un Smartphone et un Arduino équipé de ce module HC-06.

Par défaut, l'identifiant du module Bluetooth est « HC-06 » et le code pin « 1234 ».

I.9. Module Bluetooth HC-06 :

Bluetooth est une technologie de réseau personnel sans fils (WPAN pour Wireless Personal Area Network), c'est-à-dire une technologie de réseaux sans fils d'une faible portée permettant de relier des appareils entre eux sans liaison filaire .L'objectif de Bluetooth est de permettre de transmettre des données ou de la voix entre des équipements possédant un circuit radio de faible coût, sur un rayon de l'ordre d'une dizaine de mètres à un peu moins d'une centaine de mètres et avec une faible consommation électrique.

Le nom « Bluetooth » (littéralement « dent bleue ») se rapporte au nom du roi danois Harald II (910-986), surnommé Harald II Blåtand (« à la dent bleue »), à qui on attribue l'unification de la Suède et de la Norvège ainsi que l'introduction du christianisme dans les pays scandinaves [03]

I.10. Définition Module Bluetooth HC-06 :

Pour le système de transmission des données vers le PC serveurs, notre choix est porté sur le Bluetooth HC 06, le module Bluetooth permet l'envoi de trames séries en Bluetooth à un ordinateur équipé d'un récepteur Bluetooth, pour se rassurer de la communication entre l'Arduino et le PC, nous avons utilisé un terminal série appelé terra terme. Ce logiciel nous a permis d'afficher les données reçues par l'ordinateur.

Ce terminal permet également d'envoyer des trames séries à la carte Arduino, permettant ainsi la configuration à distance du système.

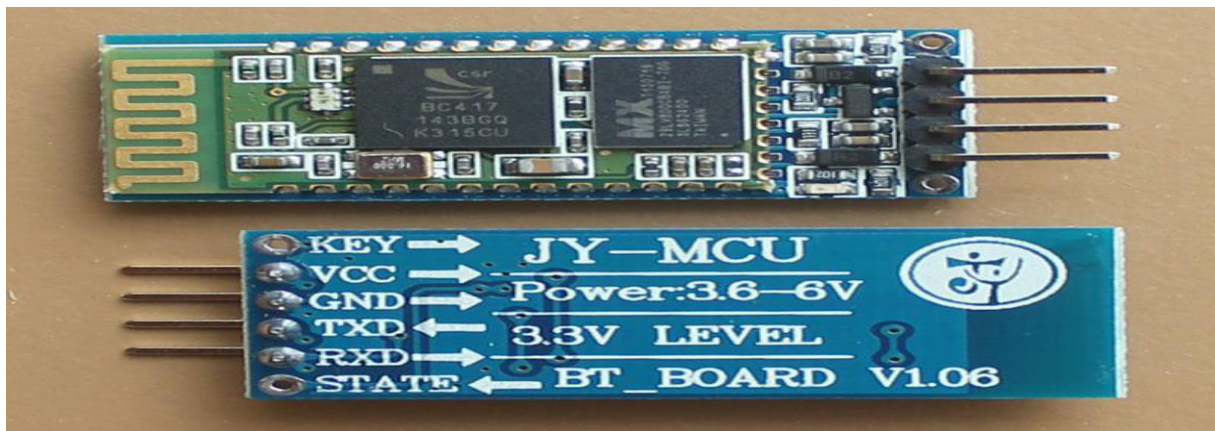


Figure I.12 : Module Bluetooth HC-06

Bluetooth est un protocole de haut niveau permettant la communication de petites radio, à consommation réduite, basée sur la norme IEEE 802, 15,4 pour les réseaux à dimension personnelle (Wireless personal area networks : WPAN).

- * Le baud rate est paramétrable de 4800 à 1382400 (par commande AT, uniquement si le module n'est pas associé, et maxi 115200 pour pouvoir l'utiliser avec une carte Arduino).
- * Alimentation de 3.3 à 5V DC
- * LED indicatrice : statu de connexion.
- * fonctionnement Bluetooth sur la bande 2,4 GHz, modulation GFSK.
- * Le module est apparié avec un mot de passe (1234) modifiable.

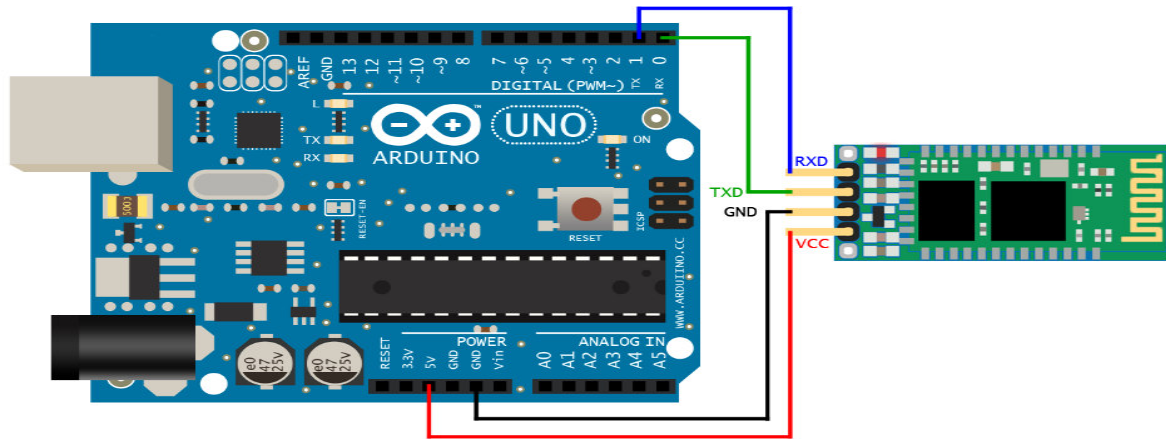


Figure I.13 : Branchement Bluetooth avec Arduino

I.11. Module relais :

Un relais est un commutateur électrique qui permet de commander un second circuit utilisant généralement une tension et un courant bien supérieur à ce que l'Arduino pourrait accepter (par exemple, allumer/éteindre une ampoule 220V). Il n'y a pas de connexion entre le circuit basse tension de l'Arduino/TinkerKit et le circuit haute tension.

Le relais est un mécanisme simple d'interrupteur on/off : il se ferme lorsque l'entrée est à 5V et s'ouvre lorsque l'entrée est à 0V. On le contrôlera donc à partir de la fonction `digitalWrite()` dans l'environnement de programmation Arduino.

Le module fournit trois connections COM, NC et NO. NC signifie "NORMALEMENT FERME". Cela veut dire que lorsque le relais n'a pas de signal d'entrée (valeur LOW dans `digitalWrite()`), le circuit haute tension connecté sera actif. Si par contre, nous appliquons une tension de 5V au relais, le circuit secondaire sera coupé. NO signifie "NORMALEMENT OUVERT". Cela veut dire qu'à contrario, une valeur de 5V appliqué au relais (valeur HIGH dans `digitalWrite()`). Coupera le circuit haute tension et inversement. Pour remplacer un interrupteur classique par le module, la procédure est simple : on retire l'interrupteur et on connecte les fils aux entrées COM et NO. De cette façon lorsque le relais est activé via `digitalWrite`, le courant parcourt l'appareil que nous souhaitons contrôler.

I.11.1. Description du module relais :

- Relais 250V 10A .
- transistor .
- LED qui signale que le module est correctement alimenté.

Ce module est un ACTIONNEUR. Le connecteur est une entrée(INPUT) qui doit être connecté à une des sorties (OUTPUT) de la carte "Sensor"

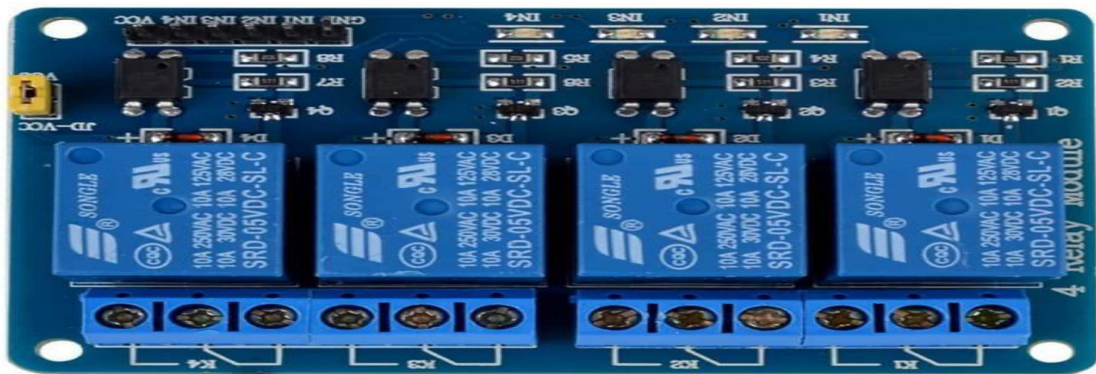


Figure I.14 : Module à 4 relais

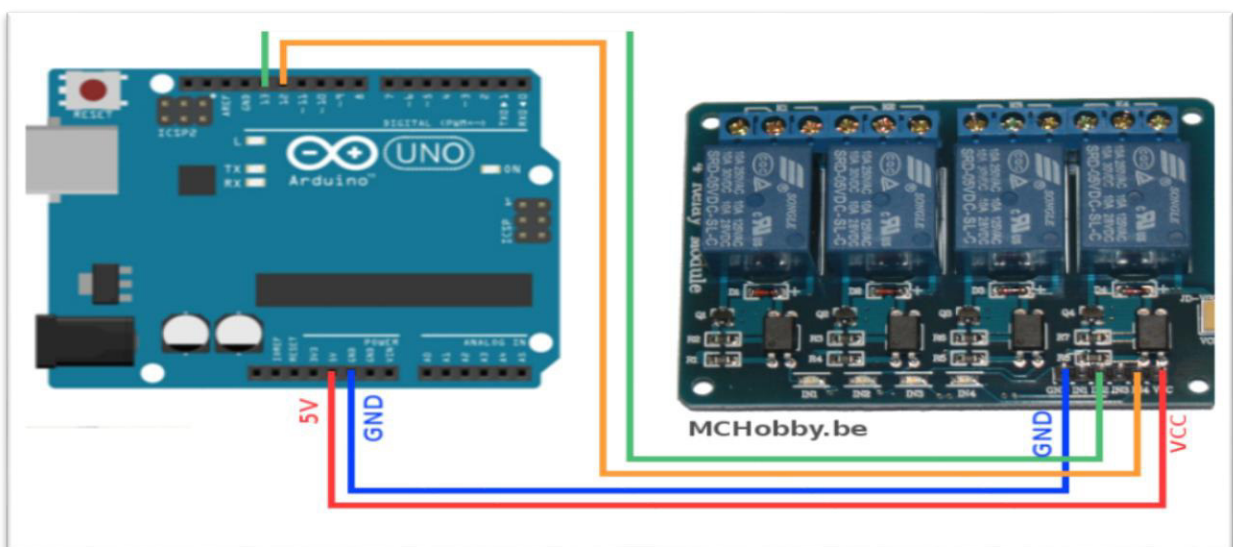


Figure I.15 : Branchement Module 4 relais avec Arduino

I.12. Smartphone (Samsung j3(2016)) :

I.12.1 Description :

Le Samsung J3 (2016) reprend les mêmes caractéristiques techniques haut de gamme du Samsung J .

- **Caractéristiques :**
 - Système d'exploitation (OS) : Android 5.1 Lollipop .
 - Résolution : 1280 x 720 pixels .
 - Processeur : Spreadtrum SC8830 - 1.2 GHz .
 - Bluetooth
 - WIFI



Figure I.16 : Smartphone Samsung j3(2016)

I.13. Conclusion:

Dans ce chapitre nous avons présenté brièvement les bases techniques et technologiques que nous souhaitons pour atteindre les objectifs visé.

Nous avons également présenté les différents composants et modules nécessaires à notre réalisation.



Chapitre I :

*Les composants
constitutifs du projet*

Introduction générale

1.Introduction générale :

La forte augmentation des ventes de smart phone et de tablettes électronique se fait en même temps qu'une adoption rapide par le grand public des technologies de la domotique ainsi que l'autopilotage. Au fond, le smart phone, avec sa connectivité Bluetooth intégrée, devient une télécommande universelle pour toute la maison et les équipements électriques. Les utilisateurs pourront à terme contrôler à distance un très grand nombre de fonctions sans avoir à tenir compte de la marque ou de l'origine du produit qu'ils pilotent. Pour répondre à cette évolution majeure, nous avons créé une carte qui permet de contrôler n'importe quel appareil, véhicule, machine à travers un smart phone ou à une tablette. Cette carte électronique réalisée rend la commande et le contrôle facile et souple lors du pilotage à distance .

2. Position du problème :

Dans la vie moderne, on utilise assez d'outils et d'accessoires de commande à distance afin de simplifier notre contrôle, donc nous cherchons toujours à se concentrer sur la souplesse de la commande et de contrôler sur une zone bien définie le plus grand nombre possible des charges .

Le smart phone occupe la première place d'objets qui ne nous quittent pas donc notre travail se concentre sur l'utilisation de ce dernier avec bien sûr sa liaison avec un système ou une carte de commande (carte d'interface) telle que l'Arduino .

Introduction générale

3.Objectif du projet :

Dans ce projet trois objectifs ont été visés :

- Le premier est de regrouper suffisamment d'informations sur une grande catégorie de cartes d'interfaçage (Arduino) :
 - son langage de programmation .
 - sa construction .
 - son principe de fonctionnement .

- Le deuxième consiste à avoir une carte électrique capable d'exécuter une action entre un Smartphone et une carte d'interfaçage (Arduino) en expliquant les différents blocs de sa construction.

- Le troisième est de réaliser une application sous smart phone et de le programmer sous l'environnement APPINVENTOR afin de simplifier la commande et de montrer son intérêt.



*Introduction
générale*

Sommaire

Introduction générale

1.Introduction générale	1
2.Position du problème.....	1
3.Objectif du projet	2

CHAPITRE I :

Les composants constitutifs du projet

I.1.Introduction :	4
I.2. Definition du module Arduino :	4
I.3. La carte Arduino:.....	4
I.4.Les différents type d'Arduino :	5
I.4.1. La carte Arduino UNO :	5
I.4.2. La carte Arduino Lenardo :	5
I .4.3. La carte Arduino Mega.....	6
I .4.4. La carte Arduino Mega ADK :	7
I .4.5. La carte Arduino Due :	7
I .4.6. La carte Arduino Nano :	8
I .4.7. La carte Arduino Mini Pro :	8
I .4.8. La carte Arduino Yun :	9
I .5. Spécifications techniques de la carte Arduino UNO :	10
I .5.1. Caractéristiques de la carte arduino :	10

Sommaire

I.5.2. Alimentation :	10
I.5.3. Mémoire :	12
I.5.4. Entrées et sorties numériques :	12
I.6. Avantages de la carte Arduino :	12
I.7. Logiciel :	13
I.7.1. Communication :	14
I.7.2. Programmer Arduino :	15
I.7.3. La structure d'un programme :	15
I.8. Présentation du module Bluetooth (HC-06) :	16
I.9. Module Bluetooth HC-06 :	16
I.10. Définition Module Bluetooth HC-06 :	16
I.11. Module relais :	18
I.11.1. Description du module relais : :	19
I.12. Smartphone (Samsung j3(2016)) :	20
I.12.1 Description :	20
I.13. Conclusion :	21

CHAPITRE II : ***Programmation et simulation du projet***

II.1. Introduction :	23
II.2. réalisé un schéma synthétique du système :	23
II.3. Les différentes étapes de la réalisation :	24
II.4. Schéma synoptique général :	25
II.5. Explication des Blocs :	26
II.5.1 Bloc d'alimentation :	26

Sommaire

II.5.2. Bloc de traitement et contrôle :.....	26
II.5.3. bloc de commande :	27
II.5.4. Bloc de puissance :	27
II.5.4.1. Etage de relais:	27
II.6. Le logiciel proteus:.....	28
II.6.1. ISIS :	29
II.6.2. La réalisation virtuelle sous ISIS « PORTEUS » :	29
II.7. fonctionnement de commande Tactile:	31
II.8. Conclusion :	32

CHAPITRE III :

Programmation et réalisation de la plateforme Androïde

III.1. Introduction :	34
III.2. Les types de programmation :.....	35
III.2.1. Présentation de l’organigramme IDE:	35
III.2.1.1. l’environnement de la programmation:	35
III.2.1.2. Présentation du programme IDE:.....	35
III.3. Présentation Application Android :	36
III.3.1. Les Versions d'Android :	36
III.3.2. Le système d'exploitation Android :.....	37
III.3.3. L’architecture de la plateforme d’Android :	38
III.4. Les avantages d'Android :	38
III.5. Explication du programme :	39
III.6. App Inventor pour Android :	39

Sommaire

III.6.1. L’outil App Inventor :.....	40
III.7. Historique de logiciel App Inventor :.....	41
III.8. Langage JAVA :	41
III.9. Un commencement avec App Inventor :.....	41
III.10. Structure d’une application App Inventor :	43
III.10.1. L’interface graphique :.....	43
III.11. Editeur de blocs (Fenêtre Scratch):.....	45
III.12. Explication et demarche d' application :	50
III.12.1. L’interface Commande vocale :.....	50
III.12.2. L’interface Commande Tactile des charges:	50
III.12.3. L’interface Application :	51
III.13. Conclusion :	52

Conclusion générale

1. Conclusion générale :.....	54
-------------------------------	----

Annexes

Annexes : A.....	56
Annexes : B.....	57

Bibliographie

Liste des Figures

Liste des Figures

CHAPITRE I :

Figure I.1 : La carte Arduino Uno.....	5
Figure I.2 : La carte Arduino Lenardo.	6
Figure I.3 : : La carte Arduino Mega.	7
Figure I.4 : La carte Arduino Mega ADK.	7
Figure I.5 : La carte Arduino DUE.	8
Figure I.6 : La carte Arduino NANO.	8
Figure I.7 : La carte Arduino Mini Pro.	9
Figure I.8 : La carte Arduino YUN.....	9
Figure I.9 : Constitution de la carte Arduino UNO.....	11
Figure I.10 : Brochage de la carte Arduino Uno.....	11
Figure I.11 : Interface IDE Arduino.....	15
Figure I.12 : Module Bluetooth HC-06.....	17
Figure I.13 : Branchement Bluetooth avec Arduino	17
Figure I.14 : Module à 4 relais.....	19
Figure I.15 : Branchement Module 4 relais avec Arduino	19
Figure I.16 : Smartphone Samsung j3(2016).....	20

CHAPITRE II

Figure II.1. Figure II.1.Schéma résume synthétique "Arduino ,Bluetooth, Relais" de communication.	24
Figure II.2: Schéma synoptique du dispositif	26
Figure II.3 : schéma global du projet (commande vocale).	28
Figure II.4: La carte réalisée sous ISIS –PORTEUS	30

Liste des Figures

Figure II.5 : Organigramme du système de commande Tactile.....30

CHAPITRE III

Figure III. 1: Présentation graphique du programme 36

Figure III .2 :L'architecture de la plateforme d'Android 38

Figure III .3: l'espace du travail sur l'application..... 40

Figure III.4 : interface App Inventor..... 42

Figure III. 5 : Création de nouveau projet sur App Inventor 42

Figure III.6: Première interface de la création App Inventor 43

Figure III.7 : Les composants graphiques 44

Figure III.8 : Les propriétés et les classes sous App Inventor 44

Figure III.9 : Ajustement d'un bouton sous App Inventor 45

Figure III.10 : En-tête d'éditeur de blocks App Inventor 46

Figure III.11 : Editeurs de blocks App Inventor 47

Figure III.12 : Echantillon d'un composant sous App Inventor 48

Figure III.13 : Schéma global du Scratch 49

Figure III.14 : L'image final de l'application vocal..... 50

Figure III.15 : L'image final de l'application Tactile 51

Figure III.16 : Photo réal de notre application smart house 51

Liste des abréviations

CAO :Conception Assistée par Ordinateur .

IDE : Integrated Development Environment.

CAN : Convertisseur Analogique /Numérique.

RX :Sigle de réception pour arduino.

TX :Sigle d'émission pour arduino.

Liste des Tableaux

Tableau III .1 : Les versions Andriod.....37

ملخص :

يسمح لنا المشروع بالغوص في عالم التواصل من اجل الوصول الى إنشاء تطبيق يربط بين الهاتف الذكي و بطاقة الاردوينو و الاجهزة الكهربائية .

الهدف الاولي من هذا المشروع هو اتقان لغة البرمجة من أجل الوصول الى انشاء تطبيق اندرويد قادر على إرسال أوامر من طرف مستخدم ، هذا التطبيق المثبت على هاتف الذكي الى أجهزة كهربائية . فهم مختلف بروتوكولات الاتصال و إستعمال المنفذ التسلسلي كمنفذ لإرسال البرنامج الى بطاقة الأردوينو . في هذا التطبيق قمنا بتحقيق جهاز إتصال عن بعد عن طريق تقنية البلوتوث من أجل تحقيق التحكم في المعدات المنازل الذكية ، عن طريق تطبيق أندرويد المثبت على الهاتف الذكي .

. **الكلمات المفتاحية :** تطبيق أندرويد ، بطاقة اردوينو ،المنفذات ،لغة البرمجة،بروتوكولات الاتصال ، بلوتوث .

Résumé :

Cette application nous permet de prolonger dans le monde de communication afin d'atteindre la création d'une application qui connecte le téléphone intelligent, la carte arduino et les actionneurs .

Le but de ce projet c'est de connaître la langue de programmation pour l'accès a crée une application Android, comprendre les protocoles de communication.

Dans cette application on a réalisé une appareil de communication via bluetooth afin de réaliser le contrôle de l'équipement programme d'installation Android sur votre smart phone .

Mots clés : application Androïde , la carte arduino. Les actionneurs .la langue de programmation .les protocoles du communication., Bluetooth

II.1.Introduction :

Dans ce chapitre, on présentera de manière sommaire une vue d'ensemble du dispositif expérimental «commande des charges 220v par la reconnaissance vocale sous système ».

Ce travail à base d'une carte Arduino UNO permet de commander une maison intelligente via système Bluetooth.

Après avoir entamé dans le chapitre précédent une description théorique sur le module Arduino et son environnement de développement, on va procéder à l'application expérimentale ; pour cette raison, plusieurs blocs ont été nécessaires afin de réaliser une telle combinaison.

II.2. réalisé un schéma synthétique du système :

L'ordinateur est relié au microcontrôleur par un port USB qui sert à transmettre les informations ainsi qu'à alimenter la carte Arduino.

Le module HC-06 communiquer avec la carte Arduino via la liaison série (broches numériques N°0 et N°1)

Branchement relais avec Arduino :

- **VCC** => V d'Arduino
- **GND** => GND Arduino
- **EN2** => Pin 13 Arduino (servira à commander le relais 2).
- **EN4** => Pin 12 Arduino (servira à commander le relais 4).

Le modèle fonctionne avec la tension 5v (HIGH) contacts de puissance à l'arrêt, et bien-sûr à 0V (LOW) le relais est en fonction.

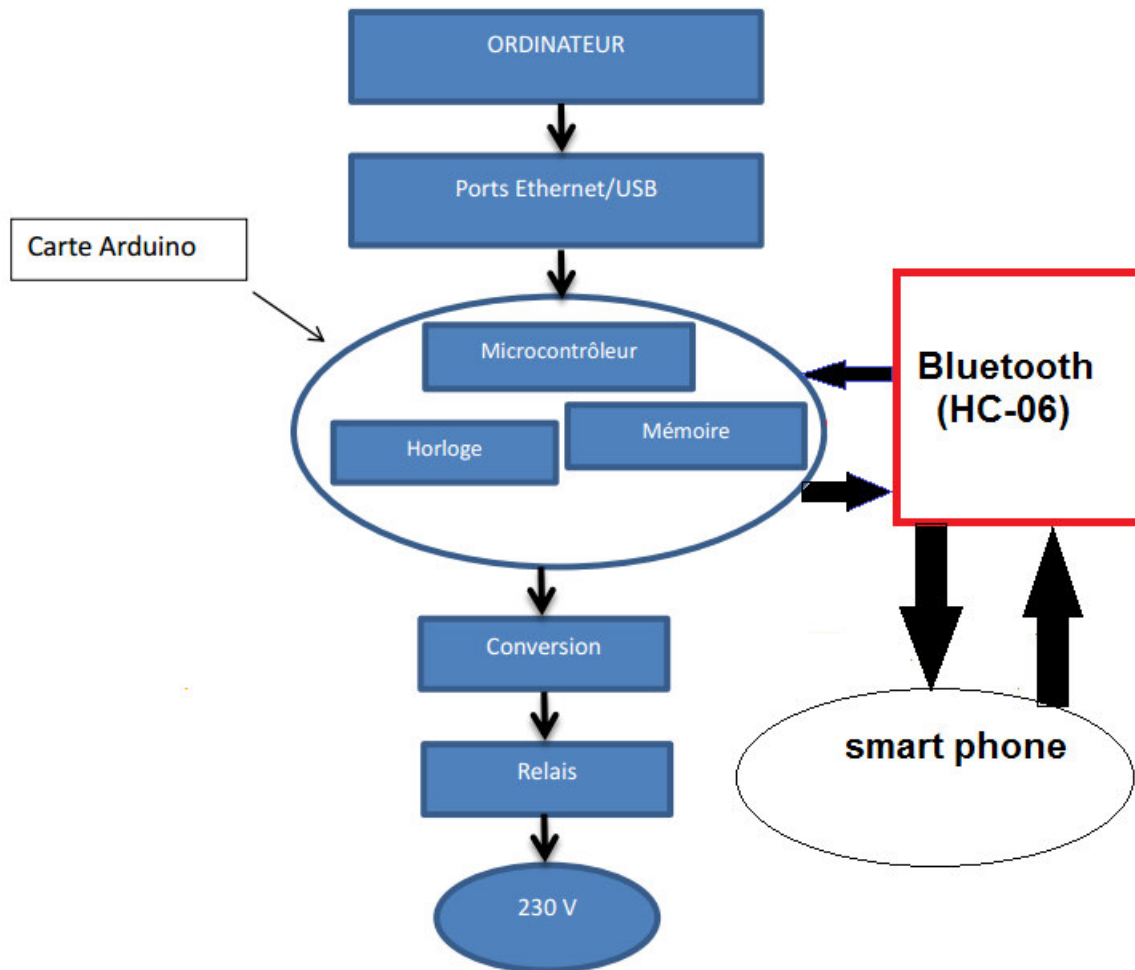


Figure II.1.Schéma résume synthétique "Arduino ,Bluetooth, Relais" de communication.

II.3. Les différentes étapes de la réalisation :

Notre réalisation pratique a été faite en trois parties:

- La première partie est la conception de tout le système électronique.
- La deuxième partie est la réalisation pratique de la carte.
- La troisième partie est la création d'une application androïde. (voir chapitre III)

La première partie de notre projet est très importante, on est passé par plusieurs étapes:

1. Chercher les différentes structures des blocs constituant notre maquette et qui vont avec les objectifs fixés et le moyens disponibles.

2. Présenter les différents éléments ou composants constituant chacun des blocs en choisissant des composants aux caractéristiques voulus, à défaut, on choisira ceux disponibles sur le marché.

1. On assemble ensuite les composants suivant notre schéma électronique de notre dispositif.

Dans la troisième partie « la création d'une application android », on passe par les deux étapes suivantes :

1. Création de l'interface de notre application android.

2. Programmation orientée objet de notre application (programmation par bloc).

II.4. Schéma synoptique général :

Le schéma synoptique général de notre dispositif est indiqué par la figure ci-dessous. Pour faciliter cette étude, on a décomposé le schéma en quatre blocs :

- Bloc d'alimentation .
- Bloc de traitement et contrôle .
- Bloc de commande .
- Bloc de puissance .

En ce qui concerne l'élément principal de ce dispositif, notre choix était le dispositif Arduino UNO.

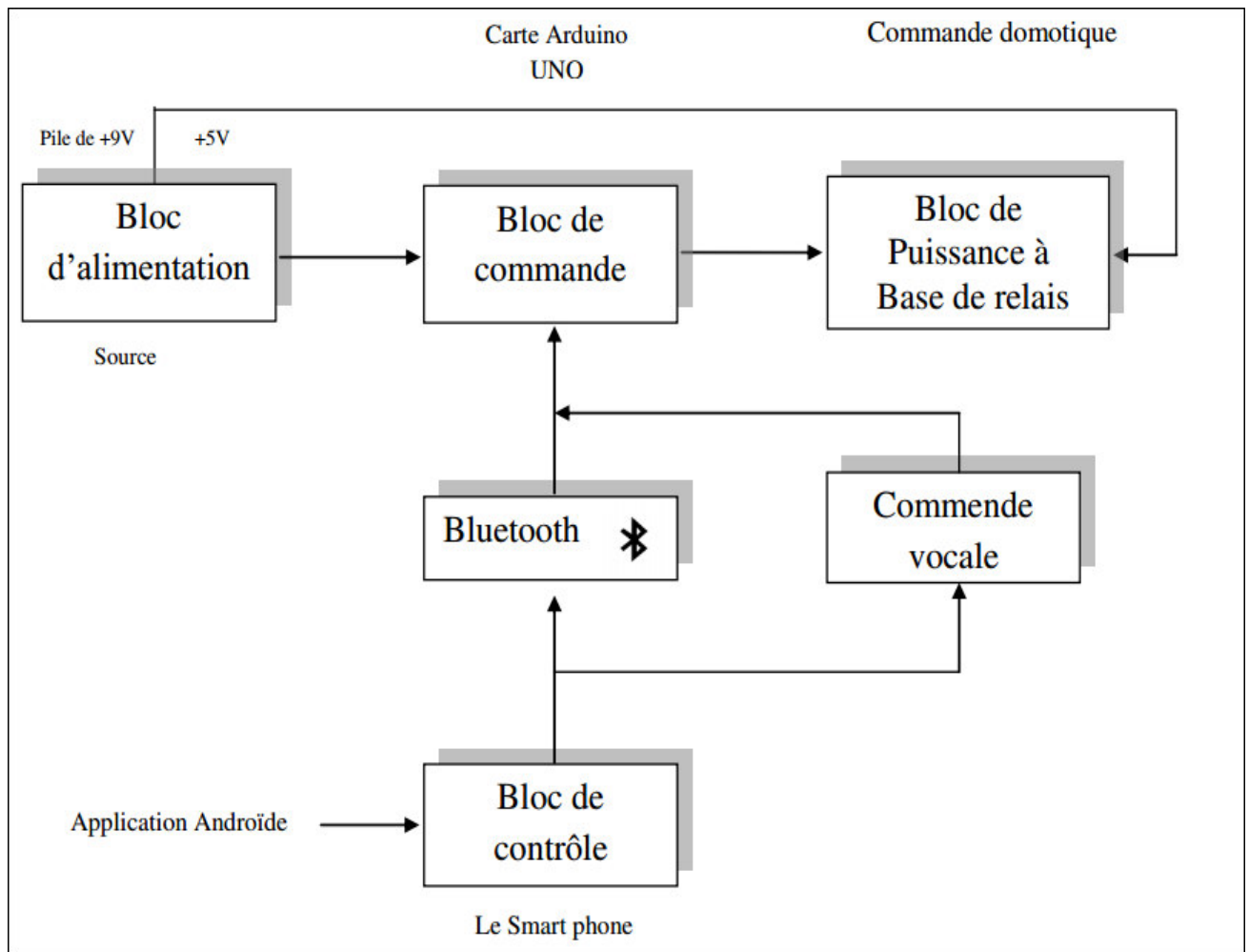


Figure II.2: Schéma synoptique du dispositif

II.5. Explication des Blocs :

II.5.1 Bloc d'alimentation :

L'ensemble des dispositifs « pile de +9 volts et source externe, relais, Arduino UNO » exige dans la plupart des cas, des alimentations stabilisées de (+5V),

II.5.2. Bloc de traitement et contrôle :

Le bloc concerné est le Smart phone, nous allons réaliser dans le chapitre qui suit une application à deux Screens dont : la première est capable de gérer une commande directe vers 4 relais (la commande domotique) et ce à travers le Bluetooth.

II.5.3. bloc de commande :

Notre bloque de commande, on le résume tout simplement par l'utilisation du module Arduino UNO qui est détaillé précédemment.

Dans cette partie, on peut classer l'accessoire Bluetooth de l'Arduino comme une suite de bloque de commande puisque il prend la relève de la validation des signaux émis par le smart phone vers l'Arduino ; donc, il a besoin d'une configuration lors de la programmation. Nous avons utilisé un modèle nommé HC-06.

II.5.4. Bloc de puissance :

Notre bloc de puissance se compose du module SRD-05VDC-SL-C :

II.5.4.1. Etage de relais:

Ce dernier se compose de quatre relais électriques donc c'est un organe électrique permettant de dissocier la partie puissance de la partie commande. Chaque relais est excité dans sa bobine par un signal généré par l'Arduino à travers un composant d'isolation (un opto-coupleur) afin d'assurer une protection de la carte Arduino.

Ce bloc contient quatre charges qui s'allument et s'éteint à savoir le signal émis par le smart phone.

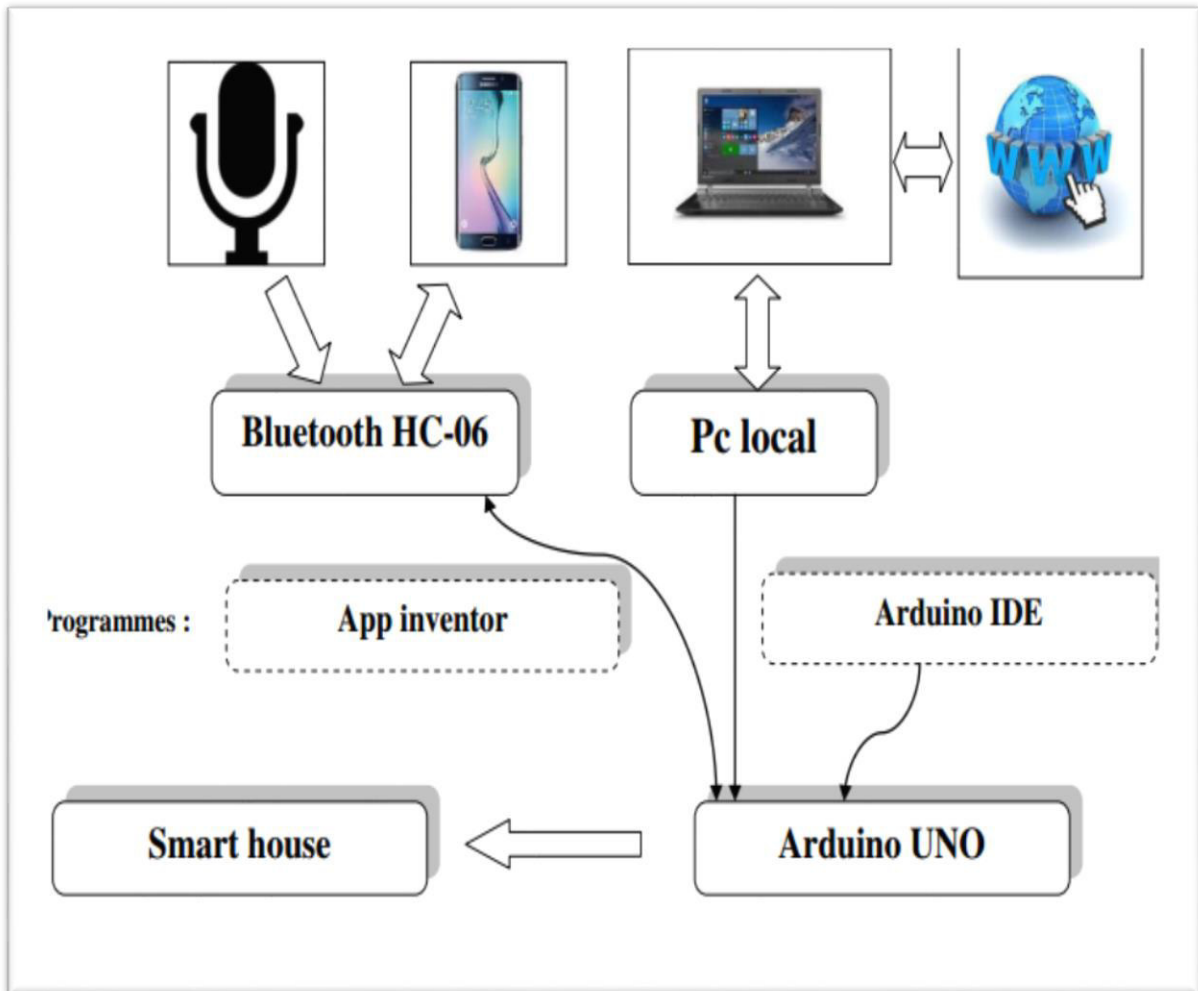


Figure II.3 : schéma global du projet (commande vocale).

II.6. Le logiciel proteus:

Proteus est un logiciel destinée à la simulation des circuits électroniques analogiques et numériques, développé par la société Labcenter Electronics. Les logiciels inclus dans Proteus permettent la CAO (Conception Assistée par Ordinateur) dans le domaine électronique, on parle de ISIS et ARES.

Outre la popularité de l'outil, Proteus possède d'autres avantages :

- Pack contenant des logiciels facile et rapide à comprendre et à utiliser.
- Le support technique est performant.
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet.

II.6.1. ISIS :

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.[04]

II.6.2. La réalisation virtuelle sous ISIS « PORTEUS » :

Avant de passer à la réalisation pratique, nous avons utilisé un CAO: il s'agit d'ISIS-PORTEUS, c'est un CAO électronique perfectionné conçu par l'Abcenter Electronics qui permet de dessiner des schémas électroniques, de les simuler et de réaliser le circuit imprimé correspondant. Le CAO électronique « PROTEUS » est disponible et téléchargeable sur le web, il se compose de nombreux outils regroupés en modules au sein d'une interface unique. [05]

Ce dernier nous permet de schématiser notre carte électrique et la simuler virtuellement comme le montre la figure suivante :

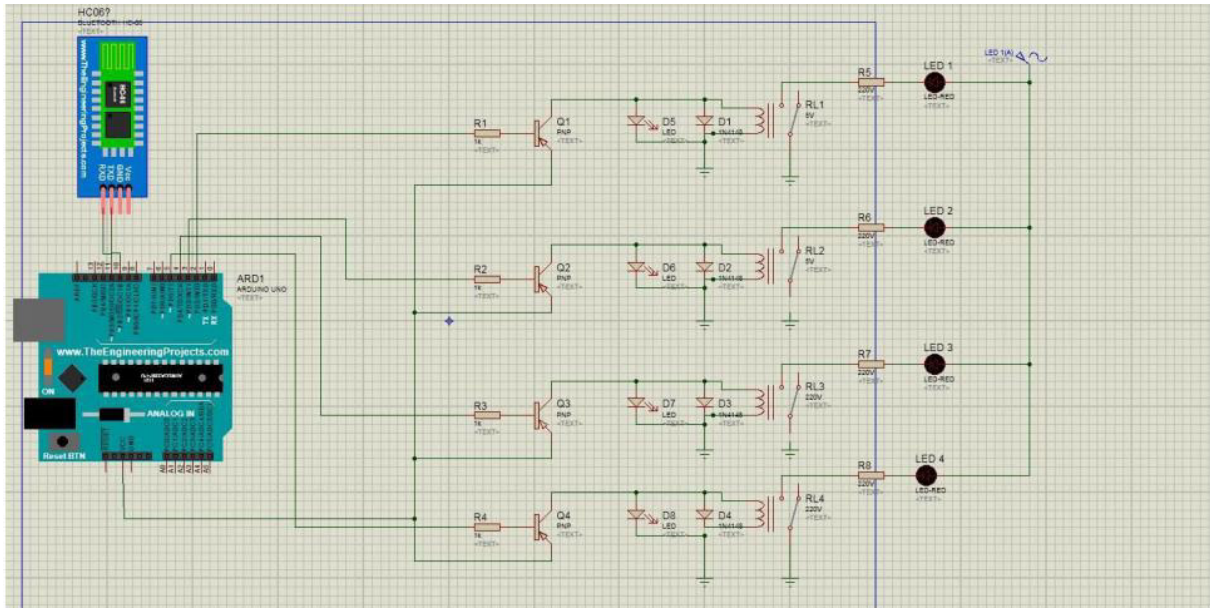


Figure II.4: La carte réalisée sous ISIS –PORTEUS

Ce CAO a la possibilité d'emporter même des codes hexadécimaux pour les réalisations qui contiennent des composants programmables ou des cartes programmables « Arduino » comme dans notre réalisation.

II.7. fonctionnement de commande Tactile :

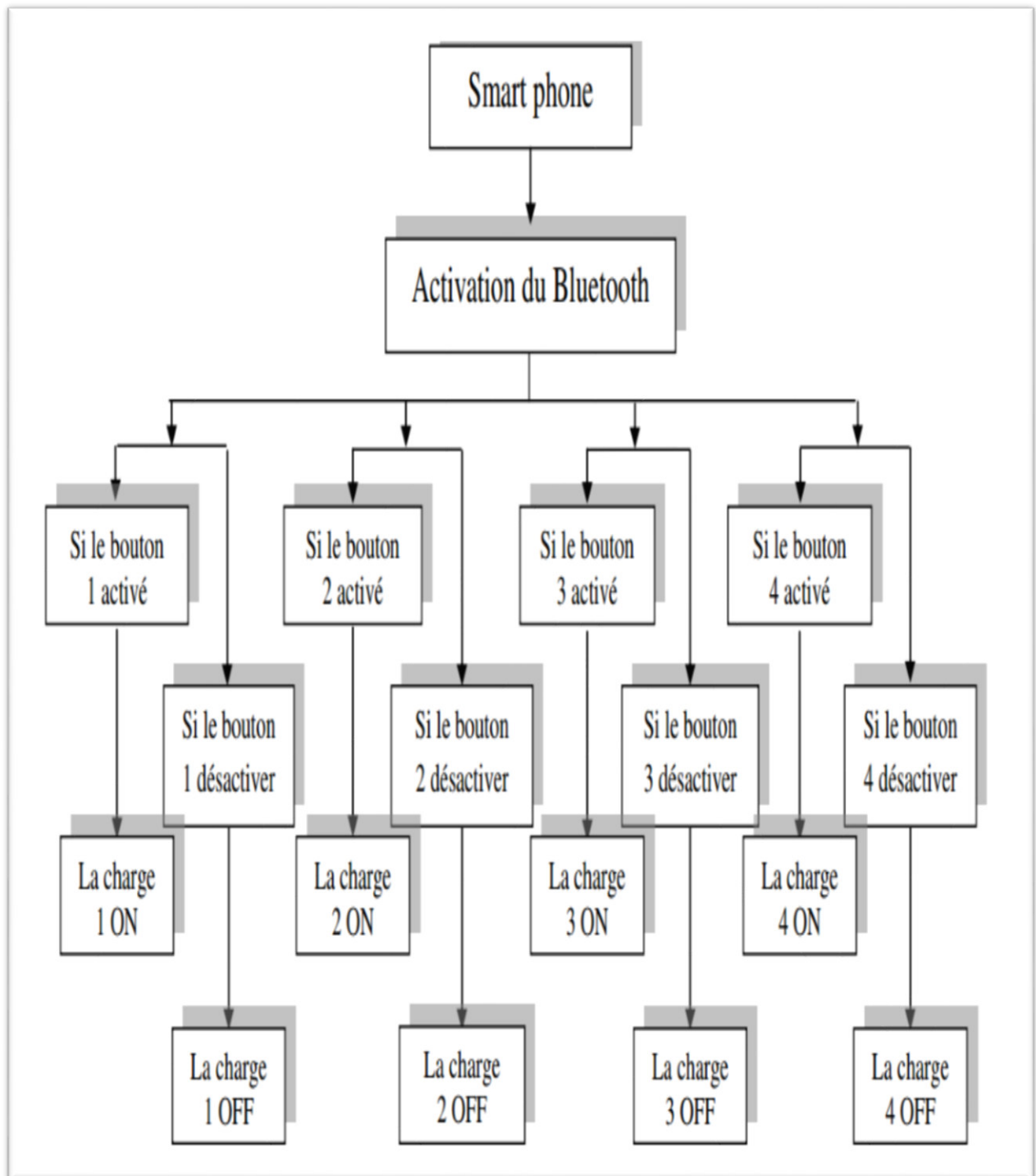


Figure II.5 : Organigramme du système de commande Tactile

II.8. Conclusion :

Cette partie a donné lieu à la réalisation pratique d'une carte électronique pour commander Une maison intelligente.

Ces commandes sont gérées à partir d'un smart phone (application Androïde) et une carte Arduino UNO. L'ensemble des travaux décrits dans ce chapitre est :

- La conception du système électronique réalisé à l'aide logiciel (ISIS- PROTEUS).
- Le choix des composants convenables pour chaque bloc.
- L'identification des paramètres des circuits d'alimentation, circuit de commande et circuit de puissance.

Cette carte électronique est destinée directement aux fabricants de produits électriques et/ou électroniques afin qu'ils rendent de manière très simple leur produit piloté à distance.

L'organigramme, le déroulement des différentes étapes du programme et la réalisation de l'application androïde seront présentés dans le chapitre suivant.