



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

MÉMOIRE DE MASTER

Sciences et Technologies
Filière : Télécommunications
Option : Réseaux et Télécommunications

Réf. :

Présenté et soutenu par :
ZERROUKI AHLAM

Le : samedi 23 juin 2018

Contrôle via une application androïde d'un système domotique basé Bluetooth

Jury :

Dr	TERGHINI WARDA	MCB	Université Mohamed Khider de Biskra	Président
Dr	BEKHOUCHE KHALED	MCA	Université Mohamed Khider de Biskra	Encadreur
Dr	RAHMANI NACER EDDINE	MAA	Université Mohamed Khider de Biskra	Examineur

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Télécommunications
Option : Réseaux et Télécommunications

**Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme:**

MASTER

Thème

**Contrôle via une application androïde
d'un système domotique basé Bluetooth**

Présenté par :
ZERROUKI AHLAM

Avis favorable de l'encadreur :
Dr BEKHOUCHE KHALED

Avis favorable du Président du Jury
M TERGHINI WARDA

Cachet et signature



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Télécommunications
Option : Réseaux et Télécommunications

Thème :

Contrôle via une application androïde d'un système domotique basé Bluetooth

Proposé par : Dr BEKHOUCHE KHALED

Dirigé par : ZERROUKI AHLAM

RESUMES (Français et Arabe)

Résumé : Notre rapport est rédigé dans le cadre du projet de fin d'études pour l'obtention du diplôme master 2 en communications et réseaux, qui nous permet de plonger dans le monde d'interfaçage afin d'arriver à réaliser un système domotique intelligent, surveillé et contrôlé par le smartphone et la carte Arduino en utilisant la technologie Bluetooth . L'objectif préliminaire est de manipuler un langage de programmation afin d'arriver à réaliser une application Androïde capable de transmettre des ordres émis par l'utilisateur vers des appareils électriques d'une part et de recevoir des données pour surveiller l'état du système d'autre part. Dans ce projet, nous avons établi un système de communication à distance via la technologie Bluetooth pour la gestion de l'éclairage de trois chambre et un ventilateur et recevoir des données pour surveiller la température et l'état du gaz et l'entrée de la maison via l'application d'Androïde installée sur le Smartphone.

ملخص: هذا التقرير تمت صياغته في إطار مشروع التخرج للحصول على شهادة ماستر 2 إتصالات وشبكات, حيث يسمح لنا بالغوص في عالم الاتصال من أجل الوصول إلى إنشاء نظام بيت ذكي تتم مراقبته والتحكم فيه عن طريق الهاتف الذكي وبطاقة الأردوينو باستخدام تقنية البلوتوث . الهدف الأولي من هذا المشروع هو إتقان لغة برمجة من أجل الوصول إلى إنشاء تطبيق أندرو يد قادر على إرسال أوامر من طرف مستخدم هذا التطبيق المثبت على الهاتف الذكي إلى أجهزة كهربائية من جهة واستقبال بيانات على نفس التطبيق لمراقبة حالة المنزل من جهة أخرى. حيث قمنا بانجاز نظام اتصال عن بعد عن طريق تقنية البلوتوث تتم فيه إرسال أوامر لتحكم في إضاءة ثلاثة غرف ومروحة واستقبال معطيات لمراقبة درجة حرارة وحالة الغاز و مدخل المنزل وذلك عن طريق تطبيق أندرو يد المثبت على الهاتف الذكي.

Remerciements

Nous tenons à remercier avant tout, louange au bon dieu qui nous a aidés à terminer ce travail

Nous adressons nos vifs remerciements à notre promoteur Mr : Bekhouche khâled ,pour les conseils qu'il nous à donné ,et pour ses remarque constructives

Nous tenons à remercier les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de juger notre travail.

Nos remerciements s'adressent également à tous les professeurs du département génie électrique pour leur contribution la quelle nous permis de recenser les informations nécessaire durant toute notre formation

Nous n'oublions pas de remercier

Nos parents pour leur aides tout au long notre étude

Enfin, exprimons notre remerciement à tous ceux qui nous ont aidés de près ou de loin.

Dédicace

*Je dédié le présent travail tout d'abord à tout
ma famille, plus particulièrement à ma parents
qui trouvent dans ce mémoire*

*L'aboutissement de longues années d'effort,
d'encouragement, et de soutien*

A vous : mes chers frères, mes chers sœurs

*Sans oublier mes tentes, mes oncles, mes
cousins et mes cousines*

A tout mes amies que j'ai connues dans ma vie

*A mes collègues à la promotion de
télécommunication et surtout Kenza*

A mes enseignants tout le long de mes études

A tout personne qui me connue

AHLAM

Résumé

Notre rapport est rédigé dans le cadre du projet de fin d'études pour l'obtention du diplôme master 2 en communications et réseaux, qui nous permet de plonger dans le monde d'interfaçage afin d'arriver à réaliser un système domotique intelligent, surveillé et contrôlé par le smartphone et la carte Arduino en utilisant la technologie Bluetooth . L'objectif préliminaire est de manipuler un langage de programmation afin d'arriver à réaliser une application Androïde capable de transmettre des ordres émis par l'utilisateur vers des appareils électriques d'une part et de recevoir des données pour surveiller l'état du système d'autre part.

Dans ce projet, nous avons établi un système de communication à distance via la technologie Bluetooth pour la gestion de l'éclairage de trois chambre et un ventilateur et recevoir des données pour surveiller la température et l'état du gaz et l'entrée de la maison via l'application d'androïde installée sur le Smartphone.

Mots clés : Domotique, maison intelligent, Bluetooth, androïde, Arduino ,contrôle .

Abstract

Our report is written as part of the final project for a communications and network engineer diploma, which allows us to dive into the world of interfacing to achieve a smart home automation system, monitored and controlled by the smartphone and the Arduino board using Bluetooth technology.

the preliminary goal is to manipulate a programming language in order to achieve an Androïde application capable of transmitting orders issued by the user to electrical devices d on the one hand and receive data to monitor the state of the system on the other hand.

In this project, we established a remote communication system via Bluetooth technology for the management of three-chamber lighting and a fan and receive data to monitor the temperature and condition of the gas and the entrance of the home via the Androïde application installed on the Smartphone.

Keywords : Home automation, Smart home, Bluetooth, androïde, Arduino , Control .

ملخص

هذا التقرير تمّت صياغته في إطار مشروع التخرج للحصول على شهادة ماستر 2 إتصالات وشبكات، حيث يسمح لنا بالغوص في عالم الاتصال من أجل الوصول إلى إنشاء نظام بيت ذكي تتم مراقبته والتحكم فيه عن طريق الهاتف الذكي وبطاقة الأردوينو باستخدام تقنية البلوتوث .

الهدف الأولي من هذا المشروع هو إتقان لغة برمجة من أجل الوصول إلى إنشاء تطبيق أندرويد قادر على إرسال أوامر من طرف مستخدم هذا التطبيق المثبت على الهاتف الذكي إلى أجهزة كهربائية من جهة واستقبال بيانات على نفس التطبيق لمراقبة حالة المنزل من جهة أخرى. حيث قمنا بانجاز نظام اتصال عن بعد عن طريق تقنية البلوتوث تتم فيه إرسال أوامر لتحكم في إضاءة ثلاثة غرف ومروحة واستقبال معطيات لمراقبة درجة حرارة وحالة الغاز و مدخل المنزل وذلك عن طريق تطبيق أندرويد المثبت على الهاتف الذكي.

الكلمات المفاتيح : بيت ذكي، بلوتوث، أندرويد، أردوينو، التحكم .

Liste d'abréviation

A

- ARQ Automatic repeat resquest
- ACL asynchrones, connectionless
- ADSL Asymeric digital subscriber

C

- CPL Courant porteur
- CAC Chanel Access code
- CRC Cyclic redundacy check

D

- DAC Device Access code
- DIAC Dedic inquiry Access code

F

- FEC Forward erro correction
- FTP Foiled Twisted Pair

G

- GSM Groupe spécial mobile
- GIAC Generic inquiry Access code
- GPL Gaz de pétrole liquéfié

H

- HEC Header error control

I

- IR Infrarouge
- IDE Integrated development environment
- IAC Inquiry Access code

Liste d'abréviation

P

-PDA Personal Digital Assistant

R

-RF Radio frequency

-RJ 45 Registered Jack 45

S

-STP Shielded Twisted Pair

-SCO synchrones connection-oriented

U

-USB universel Serial Bus

-UTP Unshielded Twisted Pair

W

-WIFI Wireless fidélité

Remerciements	
Dédicace	
Résumé	
Liste des abréviations	
Table de matière	
Listes des figures	
Introduction générale.....	1

Chpître I : La domotique

Introduction.....	4
I. 1-Définition.....	5
I.1.1. Domaines de la domotique.....	6
I.1.1.1. Protection des personnes et des biens	6
I.1.1.2. Confort de la vie quotidienne	6
I.1.1.3. Les économies d'énergie	7
I.1. 2.La maison communicante.....	7
I.1. 2.1.Domotique sans fil	8
I.1. 2.2.Domotique à courant porteur CPL.....	8
I.1. 2.3.Domotique câblée.....	9
I.2. Présentation de domotique proposés.....	10
I.2.1. Bluetooth.....	10
I.2.1. 1.Historique.....	10
I.2.1. 2.Définition.....	10
I.2.1. 3.caractéristiques.....	11

I.2.1. 4.Usage.....	11
I.2.1. 4.a. Périphériques informatiques sans fil.....	11
I.2.1.4.b. Téléphonie mobile.....	11
I.2.1. 4.c. Synchronisation de périphérique.....	12
I.2.1. 4.d. Domotique.....	12
I.2.1. 5.formt du paquet bluetooth.....	12
I.2.1. 5.a.structure.....	12
I.2.1. 5.b.Controle d'erreur.....	14
I.2.2. fonctions de domotique proposes.....	15
Conclusion.....	16

Chapitre II : Le dispositif programmable Arduino

Introduction	18
II.1 Définition du module Arduino	18
II.2 Les gammes de la carte Arduino	19
II.3 Pourquoi Arduino UNO	20
II. 4 La constitution de la carte Arduino UNO	21
II.4.1 Partie matérielle	22
II.4.1.1 Le MicrocontrôleurATMega328	22
II.4.1.2 Les sources de l'Alimentation de la carte	23
II.4.1.3 Les entrées & sorties.....	24
II.4.1.4 Les ports de communications	25
II.4.2 Partie programme.....	26
II.4.2.1 l'environnement de la programmation.....	26
II.4.2.2 Structure générale du programme (IDE Arduino).....	26
II.4.2.3 Injection du programme.....	27
II.4.2.4 Description du programme.....	27
II.4.2.5 Les étapes de téléchargement du programme.....	29

II.5- Les Accessoires de la carte Arduino.....	29
II.5.1 Communication	29
II.5.1.1 Le module Arduino Bluetooth	30
II.5.1.2 Le module shield Arduino Wifi	30
II.5.1.3 Le Module XBee	30
II.5.2 Les capteurs	31
II.5.2.1- capteur de GAZ	31
II.5.2.2- capteur de température.....	31
II.5.2.3- capteur de distance.....	32
II.5.3 Les drivers.....	32
II.5.3.1- <i>Des moteurs électriques</i>	32
II.5.3.2- <i>Les afficheurs LCD</i>	32
II.5.3.3- Module de Relais.....	33
Conclusion.....	34

Chapitre III: Développement d'une application Androïde

Introduction.....	36
III.1.le système Androïde.....	37
III.2.L'outil App Inventor.....	37
III.2.1.Historique de logiciel App Inventor.....	38
III.2.2. Langage JAVA.....	38
III.2.3.Un commencement avec App Inventor.....	38
III.2.4.Structure d'une application App Inventor.....	39
III.2.4.a.L'interface graphique.....	39
III.2.4.b.Editeur de blocs (Fenêtre Scratch).....	42
Conclusion.....	46

Chapitre IV : Réalisation de système domotique

Introduction.....	48
IV.1.Les différentes étapes de la réalisation.....	49
IV.2.Schéma general de système.....	49

Table de matière

IV.2.1.Arduino UNO.....	50
IV.2.1.1.Partie hardware.....	51
IV.2.1.2.Partie software	52
IV.2.2.Bloc qui envoyé des commandes.....	55
IV.2.3.Bloc qui recevoir des données.....	56
IV.2.3.1.Fonction d'acquisition de la temperature.....	56
IV.2.3.2.Fonction de détection de gaz.....	56
IV.2.3.3.Fonction de sécurité de l'habitant.....	57
IV.2.4.Bloc de controle et commandes.....	58
IV.2.4.1. <i>Editeur de blocs</i>	58
IV.2.4.2.Explication et demarche.....	60
Conclusion.....	61
Conclusion générale.....	62
Annexe A.....	65
Annexe B.....	66
Annexe C.....	67
Bibliographies	

Liste des figures

Figure	Nom de figure	Numéro de pages
Figure I 1	Logo de CPL	8
Figure I 2	Logo de Bus de terrain KNX	9
Figure I 3	Logo de Bluetooth	10
Figure I 4	Bluetooth au Périphériques informatiques sans fil	11
Figure I 5	technologie Bluetooth au téléphonie mobile	12
Figure I 6	Synchronisation de périphérique	12
Figure I 7	format de paquet Bluetooth	12
Figure I 8	l'entête header	12
Figure II 1	Exemples des cartes Arduino	18
Figure II 2	Carte Arduino UNO	20
Figure II 3	Microcontrôleur ATmega328	22
Figure II 4	Les tension de reference de carte arduino	23
Figure II 5	les entrées et les sortie numérique	24
Figure II 6	les entrées et les sortie analogique	25
Figure II 7	Constitution de la carte Arduino UNO	26
Figure II 8	Interface IDE Arduino	27
Figure II 9	Paramétrage de la carte	27
Figure II10	Les étapes de téléchargement du code	29
Figure II11	Type de modules Bluetooth	30
Figure II12	Module shield wifi	30
Figure II13	Module XBee	30
Figure II14	Capteur de gaz/fumée	31
Figure II15	Capteur de température	31
Figure II16	capteur de distance	32
Figure II17	Moteurs électriques	32
Figure II18	Afficheurs LCD	33
Figure II19	Module de relais	33
Figure II20	Principe de fonctionnement du relais.	33
Figure III.1	Création de nouveau projet sur App Inventor	39

Liste des figures

Figure III 2	Première interface de la création App Inventor	40
Figure III 3	Les composants graphiques	40
Figure III 4	Les propriétés et les classes sous App Inventor	41
Figure III 5	Ajustement d'un bouton sous App Inventor	42
Figure III 6	Schéma global du Screene	42
Figure III 7	En-tête d'éditeur de blocks App Inventor	43
Figure III 8	Editeurs de blocks App Inventor	44
Figure III 9	Echantillon d'un composant sous App Inventor	45
Figure IV 1	Schéma globale du système domotique à concevoir	50
Figure IV 2	brochage de la carte arduino UNO	51
Figure IV 3	Organigramme de la fonction d'acquisition de temperature	52
Figure IV 4	Organigramme de la fonction sécurité del'habitant	52
Figure IV.5	Organigramme de la fonction detection de CO	53
Figure IV 6	Présentation graphique du programme	54
Figure IV 7	brochage carte arduino avec module Bluetooth HC06	55
Figure IVI 8	Test le bloc par les LED	55
<i>Figure IV 9</i>	brochage de carte arduino avec capteur de temperature LM35	56
Figure IV 10	brochage de carte arduino avec capteur de GAZ MQ-09	57
<i>Figure IV 11</i>	brochage de carte arduino avec capteur de distance HC-SH04	57
Figure IV 9	Schéma global du Screene	58
Figure IV10	Schéma global du Scratch	59
Figure IV11	Photo réelle de notre application bluetooth3	60

Introduction générale

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.

La domotique ou encore la maison intelligente est définie comme une résidence équipée de technologies d'informatique, d'automatisme et d'électronique, ambiante qui vise à assister l'habitant dans les situations diverses de la vie domestique en améliorant le confort et simplifient un certain nombre de tâches.

Elle assure différentes fonctions:

-fonction de confort, gestion de l'éclairage pour adapter l'intensité de l'éclairage au besoin du moment, et ainsi en programmant les équipements électroménagers et multimédia.

-fonction d'économie d'énergie, en mettant en veille les dispositifs de chauffage quand les habitants sont absents ou adapter automatiquement l'utilisation des ressources électriques en fonction des besoins des résidents afin de diminuer les gaspillages de ressources énergétiques suivi des consommations et optimisation des tarifs.

- fonction de sécurité ; en outre, un autre but essentiel de l'application des technologies d'information aux maisons est la protection des individus. Cela est rendu possible par des systèmes capables d'anticiper des situations potentiellement dangereuses ou de réagir aux événements mettant en danger l'intégrité des personnes.

Notre mission porte sur la réalisation d'un contrôleur intelligent permettant de piloter la domotique à partir d'une application à distance . Les objectifs de notre projet peuvent être divisés essentiellement selon les axes suivants qu'on va les aborder sous des chapitres.

D'abord, on va commencer avec une présentation générale de la domotique ainsi ses secteurs d'application et le technologies bluetooth utilisées.

Introduction générale

Le second chapitre on présenter la carte Arduino UNO vu ses performances techniques intéressantes.

Le troisième chapitre est consacré à la description de l'application android.

Finalement, nous détaillerons les phases de réalisation du système domotique.

Chapitre I

La domotique

Introduction

Qu'est-ce que la domotique ? C'est la première question que nous nous sommes posés. La domotique rassemble différents domaines tels que l'électronique, l'informatique, la physique des bâtiments, la télécommunication... Elle a vu le jour dans les années 80 suite à la miniaturisation des systèmes électroniques. L'évolution des composants électroniques a donc permis d'améliorer les performances, de réduire la consommation en énergie des équipements utilisés et de les installer chez les particuliers.

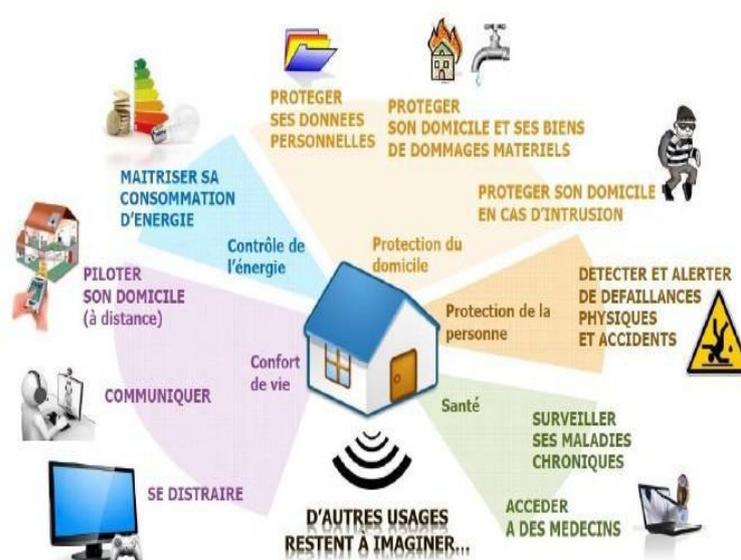
Actuellement, la domotique est utilisée pour rendre une maison "intelligente". En effet celle-ci consiste en l'automatisation et la centralisation de différents appareils électriques de la maison. Ainsi la domotique est utilisée pour le contrôle de la maison. Ce contrôle ne s'exerce pas n'importe comment. En effet le contrôle d'un appareil se fait suite à l'analyse de certaine donnée par exemple la force du vent extérieure. En effet un capteur est placé à l'extérieur et si le vent souffle trop fort les stores peuvent s'ouvrir. Ce principe est adapté à différents outils électriques de la maison. De plus, toutes ces informations sont centralisées dans un boîtier permettant à l'utilisateur de vérifier les différentes données (température de la maison, luminosité, force du vent...), d'entrer ses propres paramètres (par exemple, entrer une température idéale pour la maison, et si la température relevée est plus faible que celle voulue par l'utilisateur, les radiateurs se déclenchent).

I. 1-Définition

Avant de s’immerger dans notre projet, on a intéressé d’abord à définir clairement le mot domotique ; le mot domotique a été introduit dans le dictionnaire « **le petit Larousse** » en **1988**.

Ce mot a été construit à partir de « **Domus** », le domicile en latin, associé au suffixe « **tique** », couramment employé pour évoquer le terme des technologies d’automatique, électronique, électrique, informatique et des communications au service de la maison. Son champ d’application vise à assurer des fonctions de sécurité, de confort, de gestion d’énergie et de communication qu’on peut retrouver dans un espace domestique.

Câblée ou fonctionnant par ondes radio, la domotique investit notre univers quotidien pour nous faciliter la vie. Souvent on la pratique sans y penser. Dans ses applications les plus évoluées, la domotique met en réseau et coordonne le fonctionnement de différents types d’équipements ménagers, de travail et de loisir. Elle peut se charger des tâches les plus complexes ou contraignantes et, en même temps, assurer l’intendance de la maison. A l’opposé, elle peut accomplir des actions très basiques, comme allumer une lumière dans une pièce. Les applications possibles de la domotique concernent aussi bien la programmation, la surveillance, que le contrôle à distance. La domotique offre une simplification qui peut alléger le poids des actions quotidiennes pour les personnes âgées ou handicapées, ou tout simplement apporter un confort majeur.



I.1.1. Domaines de la domotique

Les services offerts par la domotique couvrent 3 domaines principaux :

- ❖ Assurer la protection des personnes et des biens en domotique de sécurité.
- ❖ Veiller au confort de vie quotidien des personnes âgées, entre autres, en installant une domotique pour les personnes à mobilité réduite.
- ❖ Faciliter les économies d'énergie grâce à la réactivité maîtrisée d'une maison intelligente.

I.1.1.1. Protection des personnes et des biens

La domotique permet le suivi des personnes âgées ou handicapées. En matière de sécurité domestique, rien n'est laissé au hasard. Alarmes, détecteurs de mouvement ou d'intrusion, interphones et portiers vidéo, téléphones, simulateurs de présence, etc. se combinent pour détourner les visiteurs indésirables et arbitrer toutes les fonctions.

D'autres systèmes de détection sont prévus pour surveiller les enfants, prévenir les risques d'accident (incendie, fuite de gaz, etc.) et signaler des pannes (inondation, coupure de courant électrique, etc.).

La domotique de sécurité passe également par la centralisation de la surveillance et du contrôle de toutes les zones de la maison. Des capteurs de mouvements, de bris de glace, d'ouverture, etc., des poignées biométriques, l'automatisme des volets... sont installés sur les ouvertures et préviennent de toute intrusion, car l'ensemble est couplé à des alarmes silencieuses sans fil ou des sirènes. Pour l'intérieur des pièces, des micros ultrasensibles, des caméras invisibles, des champs magnétiques, des détecteurs de fumées assurent aussi une grande sécurité s'ils sont judicieusement positionnés.

I.1.1.2. Confort de la vie quotidienne

Avec une installation domotique, on pourra aujourd'hui avoir une maison vivante et économe. Le fait de rendre la maison intelligente assurera un résultat basse-consommation évident. L'habitat offre aussi un bien-être sur-mesure, avec un confort en permanence.

Manipuler ses volets roulants ou battants en pressant un bouton est devenu chose courante de nos jours. De même qu'ouvrir le portail ou la porte du garage depuis sa voiture. Plus globalement, tout ce qui se fait avec un interrupteur ou une poignée peut être automatisé

et piloté à partir d'un poste fixe, ou à distance via une télécommande, un ordinateur ou un Smart phone.

I.1.1.3. Les économies d'énergie

La domotique permet de diminuer jusqu'à 10 % des factures d'énergie. Grâce aux automatismes et à des capteurs, les équipements électriques inter-reliés pilotent au plus juste la consommation énergétique (chauffage, éclairage, eau, ventilation, etc.), tout en gardant sous contrôle le confort des zones occupées.

Le but principal de la domotique est d'éviter le gaspillage en supprimant les dépenses inutiles. Les systèmes de régulation permettent de maîtriser la consommation d'électricité, de gérer le chauffage et la production d'eau chaude sanitaire, avec un niveau de confort optimal. Un détecteur de présence placé dans chaque pièce, par exemple, commande instantanément l'allumage ou l'extinction des éclairages, la mise en route ou l'arrêt du chauffage, etc.

Au jardin par exemple, l'arrosage s'automatise et le détecteur crépusculaire se charge d'allumer les lumières dès la tombée de la nuit et ainsi de lancer l'irrigation des plantes. La maison intelligente utilise la programmation domotique via des scénarios qu'on peut déterminer en fonction des besoins spécifiques, évitant les pertes thermiques inutiles et palliant les risques d'oubli ou de sécurité.[1]

I.1. 2.La maison communicante

L'homme avait imaginé qu'il est impossible de se communiquer avec son habitat, le contrôler à distance. Maintenant, la communication tient une place de plus en plus importante dans le logement. Une installation domotique adaptée, avec les appareils de la maison montés en réseau, satisfait aux besoins et aux loisirs de chaque personne du foyer.

La centralisation des commandes est le corps du système domotique. Les appareils mis en réseau se reconnaissent et dialoguent entre eux, se déclenchant par simple appui sur une touche. Par le biais d'un interrupteur centralisé, les éclairages et volets motorisés peuvent être actionnés individuellement, par groupes de pièces ou simultanément.

En communiquant avec l'habitat, il est possible de régler le chauffage par zones, de simuler à distance une présence, etc. En couplant l'installation avec une télécommande universelle ou avec un simple appui sur une touche sur son Smart phone, le pilotage s'effectue de n'importe où dans la maison, en fonction des besoins.

Afin d'adapter la domotique à chaque logement et utilisation, plusieurs configurations sont à disposition :

- En domotique sans fil (Wifi, ondes radio,...) ;
- Par domotique CPL ou à courant porteur (appelé X10) ;
- Avec un câblage domotique bien pensé.

I.1. 2.1.Domotique sans fil

La domotique sans fil utilise plusieurs supports technologiques : les ondes radio ou RF (sur des fréquences en MHz) et l'infrarouge ou IR, qui a pour inconvénient de ne pas traverser les murs. Il est conseillé, pour une meilleure stabilité du système, de ne pas mixer le sans fil avec un autre type de technologie, le CPL par exemple. Cela peut nuire à l'installation et à la qualité de la communication entre les équipements.

Les ondes radio sont employées par de multiples protocoles comme le X10 RF, le HomeEasy, le X2D, le Zigbee, le Zwave, ou encore le Bluetooth.

Les principales fréquences utilisées dans la domotique sont le 433 MHz et le 868MHz.

I.1. 2.2.Domotique à courant porteur CPL

L'utilisation de la domotique à courant porteur revient à transformer son habitat en maison communicante par le biais d'une installation domotique; c'est-à-dire que l'on utilise le réseau électrique déjà existant.

La domotique CPL(*Courant porteur en ligne*) est aussi connue sous l'acronyme de X10, qui est un protocole de communication et de contrôle de plusieurs appareils domotiques.



Figure I 1: Logo de CPL

Les CPL, c'est la possibilité de faire passer de l'information numérique (voix, donnée, image) sur le réseau électrique ordinaire. Ils s'avèrent très utiles en cas de rénovation.

L'intérêt de cette technologie porte sur l'utilisation d'un réseau filaire structuré déjà existant et parfaitement distribué dans toute la maison ou le bâtiment : le réseau électrique et ses points d'accès constitués par les prises électriques.

Toutefois, la fiabilité de la domotique CPL est contestable. Cette technologie peut parasiter le réseau et perturber les autres transmissions. De plus, cet équipement est

encore coûteux. À performances équivalentes, il est en effet plus cher que le sans fil. Enfin, ce système est aussi moins rapide, et il n'a pas de mobilité par construction.

I.1. 2.3. Domotique câblée

Certains professionnels ne sont pas favorables, au sein d'une installation domotique, aux approches sans fil ou CPL. Ils leur préfèrent une domotique par câbles. Le pré-câblage doit être souple et évolutif, car la technologie ne cesse d'évoluer.

Il faut ainsi prévoir un local technique, le «local de répartition», qui centralise les points d'arrivée de toutes les liaisons externes (électricité, téléphone, Internet, télévision, fibre optique ...).

Dans les logements, le Bus de terrain KNX est une excellente solution domotique. Ce Bus est constitué d'un câble fait de conducteurs torsadés par paires (deux au minimum) alimenté en très basse tension (courant faible).



Figure I 2: Logo de Bus de terrain KNX

Le réseau a pour but d'empêcher les interférences électriques reprochées au CPL. Cependant, tout repose sur la qualité des câbles choisis.

Trois types de câbles sont fréquemment rencontrés, le câble UTP (*Unshielded Twisted Pair*), le câble STP et le câble FTP (*Foiled Twisted Pair*). Les meilleurs câbles sont blindés ou écrantés, de type STP (*Shielded Twisted Pair*) ou FTP.

Il est conseillé de choisir un réseau électrique, car c'est le plus simple à installer (et le mieux connu par les artisans et les architectes). Il doit respecter toute fois la norme NFC15-100. Il est aussi préférable d'installer un panneau de brassage équipé de prises RJ45. Ensuite, il faut prévoir un onduleur pour les équipements du réseau (modem ADSL (*Asymmetric digital subscriber*), routeur, switches) et les équipements de la domotique de sécurité.[1]

I.2. Présentation de domotique proposés

Pour notre système nous avons choisis la domotique sans fil par technologie Bluetooth

I.2.1. Bluetooth

I.2.1. 1.Historique

La technologie Bluetooth a été à l'origine mise au point par Ericsson en 1994. En 1998, un groupe d'intérêt baptisé Bluetooth SIG (Bluetooth Special Interest Group) a été fondé par Ericsson, IBM, Intel, Toshiba et Nokia. Aujourd'hui, plus de 2500 entreprises ont rejoint le groupe.

L'origine de l'appellation Bluetooth fait référence à un roi danois Harald « Dent bleue» (dû à son goût immodéré pour les mûres) qui aurait unifié les différents royaumes nordiques à la fin du Moyen Âge, de même que Bluetooth SIG s'est créé autour d'intérêts communs. Le but principal du Bluetooth SIG est de développer des produits interopérables. C'est ainsi qu'a été créée une spécification sans licence pour ses membres afin de développer des produits et logiciels utilisant Bluetooth (standard IEE 802.15).[2]

I.2.1. 2.Définition

La technologie Bluetooth est beaucoup répandue dans le monde des télécommunications et dans les appareils sans fil. Depuis quelques années, cette technologie a subi plusieurs modifications et améliorations afin de percer le marché du monde industriel.



Figure I 3: Logo de Bluetooth

Cette technologie retient l'attention, car elle possède une excellente portée, une bonne vitesse de transmission et plusieurs autres avantages.

La recherche n'a pas été très complexe, car l'information était facilement accessible ainsi que le support technique des exemples de programmation et des documents possédant les commandes de programmation entre autres sont facilement accessibles. La simplicité des commandes et le support (documentation et accès à des professionnels) sont de très gros avantages.

La technologie Bluetooth possède d'autres atouts, puisqu'il fallait établir la communication à partir d'un ordinateur portable, un Smart phone ou une Tablette et que toutes les nouvelles technologies sont presque tous munis de cette technologie.[1]

I.2.1. 3.caractéristiques

Bluetooth a pour principal objectif de remplacer les câbles. En effet, les fils qui permettent de relier des périphériques à des ordinateurs par exemple sont assez souvent contraignants, et ne permettent pas une grande liberté de mouvement en plus d'être encombrants. C'est pour cela que cette technologie supporte les caractéristiques suivantes:

- ✓ Faible coût ;
- ✓ Faible puissance d'émission, d'où :
 - o Courte distance d'émission (quelques dizaines de mètres)
 - o Faible consommation d'énergie (donc adapté aux produits portables)
- ✓ Performances modestes (1Mbps) ;
- ✓ Topologie ad hoc ;
- ✓ Configurable dynamiquement ;
- ✓ Support des transferts voix et données ;
- ✓ Destiné à un usage personnel (PAN : Personal Area Network) ;
- ✓ Certification Bluetooth pour assurer la compatibilité des produits entre eux.[2]

I.2.1. 4.Usage

Bluetooth est aujourd'hui utilisé dans de nombreux secteurs dont voici quelques exemples

I.2.1. 4.a. Périphériques informatiques sans fil

On peut désormais utiliser clavier, souris et casque audio en toute liberté



Figure I 4: Bluetooth au Périphériques informatiques sans fil

I.2.1. 4.b. Téléphonie mobile

Il est possible de se connecter à partir de son ordinateur portable ou PDA vers un téléphone GSM avec fonction de modem et obtenir une connexion Internet. Une autre fonction plus répandue est l'apparition des kits main libre Bluetooth (oreillette).



Figure I 5: technologie Bluetooth au téléphonie mobile

I.2.1. 4.c. Synchronisation de périphérique

La synchronisation des contacts et calendrier des PDA via Bluetooth est également possible.



Figure I 6: Synchronisation de périphérique

I.2.1. 4.d. Domotique

gestion du chauffage électrique, de l'éclairage, des appareils électroménagers, de l'arrosage automatique [2]

I.2.1. 5. format du paquet bluetooth

I.2.1. 5.a. structure

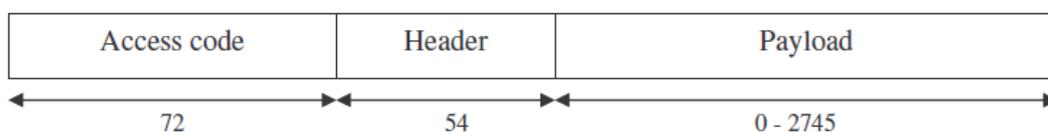


Figure I 7 format de paquet Bluetooth

Le paquet Bluetooth est composé de 3 différentes parties :

☒ « **Access code** » (68/72 bits) : Il identifie le piconet maître et il est utilisé pour la synchronisation, le « paging » et la recherche. Il y a trois types de code d'accès :

- *CAC (Channel Access Code)* : Ce code de canal est identifié un unique piconet.
- *DAC (Device Access Code)* : Le code de composant est utilisé durant pagination (« page » et « page scan ») et ses réponses. Ce code dérive de la BDA..
- *IAC (Inquiry Access Code)* : Ce code est utilisé dans les procédures de recherche de composant. Il y a deux type de IAC :

_ *GIAC (Generic IAC)* : Utilisé par défaut pour rechercher tous les périphériques Bluetooth dans la zone. (0x9E8B33)

_ *DIAC (Dedicated IAC)* : Utilisé pour rechercher un type spécifique de composant.

☒ « **Header** » (54 bits) : L'entête est codé sur 54 bits. Il s'agit en fait d'une séquence de 18 bits répétés trois fois. Cette séquence est composée de six champs :

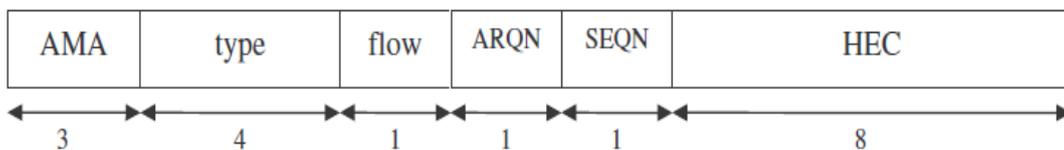


Figure I 8 l'entête header

o *AMA (3 bits)* : L'adresse active de l'esclave. 0 pour le broadcast et 1 à 6 pour le périphérique.

- *Type (4 bits)* : SCO, ACL, NULL, POLL / type de FEC / durée du payload
- *Flow (1 bit)* : Contrôle de flow pour signaler que la mémoire tampon est pleine
- *ARQN (1 bit)* : Indication de l'acquittement (ACK)
- *SEQN (1 bit)* : Numéro de séquence
- *HEC (8 bits)* : « Header Error Control »

☒ « **Payload** » : Données binaires utiles avec une zone de contrôle d'erreur de 16 bits.

I.2.1. 5.b. Contrôle d'erreur

☒ Fec

Les données peuvent être protégées par le code de correcteur d'erreur FEC (Forward Error Correction) :

- Le code FEC 2/3 nécessite 3 bits pour en protéger 2. (ex : 160b utiles pour 240b)
- Le code FEC 1/3 nécessite 3 bits pour en protéger 1. (ex. 80b utiles pour 240b) Il s'agit d'un codage avec répétition de bits. Chaque en-tête d'un est toujours protégé par un FEC car elle est composée d'une séquence de 18 bits répétés 3 fois. Cette protection réduit donc le débit utile mais, en contre partie, il permet la correction des paquets en erreur sur la liaison.

☒ CRC

Pour les données, il s'ajoute au FEC une détection d'erreur CRC (Cyclic Redundancy Check) qui permet au récepteur de détecter les paquets en erreurs. Si le paquet est corrompu, le CRC peut demander un ARQ (Automatic Repeat Request) pour lui redemander ce paquet.[2]

☒ Automatic Repeat

Request Dans le protocole ARQ, les paquets sont retransmis jusqu'à ce qu'un acquittement soit reçu (ou le dépassement du délai de temps). Bluetooth emploie une confirmation rapide et non numérotée dans laquelle il emploie des confirmations positives et négatives en plaçant les valeurs appropriées dans l'en-tête. Si le délai de temps est dépassé, le paquet est perdu et la transmission est poursuivie avec la suite des paquets. Dans la pratique, - ARQN =1 si paquet précédent bien reçu et SEQN=SEQN+1 (SEQN = NOT SEQN) - ARQN=0 si paquet précédent pas bien reçu. Cela arrive dans les cas suivants :

- o Le maître ne détecte pas le code d'accès. (paquet perdu)
 - o HEC échoue
 - o CRC échoue
 - o SEQN = SEQN donc retransmission et ignore les paquets SEQN+1
- Le broadcast n'a pas de ARQ.

☒ Contrôle de Flux

Le protocole de bande de base recommande l'emploi de files d'attente de type FIFO dans les liens ACL et SCO pour la transmission et la réception. Le gestionnaire de lien remplit ces files d'attente et le contrôleur de lien les vident automatiquement.

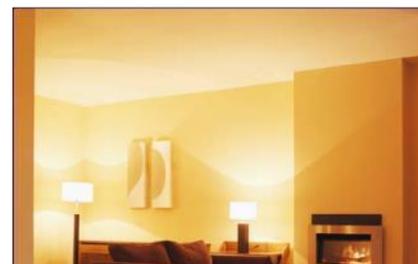
Pour éviter que la file d'attente de réception soit pleine, ce qui provoquerait des pertes de paquet et de la congestion, on utilise un contrôle de flux. Une indication d'arrêt est transmise lorsque la queue est pleine. Elle est insérée par le contrôleur de lien du récepteur dans l'en-tête du paquet de retour. Lorsque l'émetteur reçoit l'indication d'arrêt, il bloque ses files d'attentes. Lorsque le récepteur est à nouveau prêt il envoie un paquet pour continuer la transmission.[2]

I.2.2. fonctions de domotique proposes

comme fonctions à établir :

❖ Fonction de gestion d'éclairage et ventilation

Cette fonction permette aux utilisateurs de faire la gestion de l'éclairage de 3 pièces et un ventilateur dont le but d'économiser l'énergie électrique.



❖ Fonction d'acquisition de la température

L'acquisition de la température se fait via un capteur de température LM35 pour contrôler le climat à l'intérieur de l'habitat avec de la ventilation.



❖ Fonction de détection de gaz

Cette fonction permet de détecter les fuites de gaz via un capteur de gaz MQ-9. Il est apte à détecter le monoxyde de carbone, le methane et le Gaz de Pétrole, GPL.



❖ Fonction de sécurité de l'habitant

Cette fonction vous permet de détecter tout mouvement près de la maison via au capteur ultrasonique HC-SR04



Conclusion

Dans ce thème qu'est la domotique "Le contrôle de la maison à distance" nous avons voulu essayer de reproduire ce phénomène à une échelle plus réduite .Ainsi nous avons opté pour la problématique suivante : pouvons nous créer un système de contrôle à distance de la maison avec des moyens réduits. Pour communiquer avec la maison en question nous avons choisi d'utiliser le téléphone portable. En effet, celui-ci nous accompagne partout, nous l'avons toujours sur nous, il sera ainsi très facile pour n'importe qui de prendre son téléphone et d'envoyer un message de contrôle à "sa maison".

Chapitre II

Le dispositif

programmable

Arduino

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.1-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. [3]



Figure II 1 : Exemples des cartes Arduino

I.2-Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique:

- ❖ Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- ❖ L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- ❖ L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- ❖ L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- ❖ Le LilyPad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- ❖ Le NG d'Arduino plus, avec une interface d' USB pour programmer et usage d'un ATmega168.
- ❖ L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- ❖ L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- ❖ L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- ❖ L'Arduino Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- ❖ L'Arduino UNO, utilisations microcontrôleur ATmega328.
- ❖ L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- ❖ L'Arduino Leonardo, avec un morceau ATmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.

- ❖ L'Arduino Esplora : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération. [4]

Parmi ces types, nous avons choisi une carte Arduino UNO (carte Basique). L'intérêt principal de cette carte est de faciliter la mise en oeuvre d'une telle commande qui sera détaillée par la suite.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega 328 et des composants complémentaires. La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal a 16 MHz, une connexion USB et Possède un bouton de remise à zéro et une prise jack d'alimentation.

La carte est illustrée dans la figure si dessous. [5]



Figure II 2 : carte Arduino UNO

I.3-Pourquoi Arduino UNO

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:

- ✓ **Le prix (réduits)** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino pré-assemblées coûtent moins de 2500 Dinars).
- ✓ **Multi plateforme** : le logiciel Arduino, écrit en JAVA, tourne sous les systems d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- ✓ **Un 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 Open source et 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.[6]

I. 4 La constitution de la carte Arduino UNO

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

I.4.1-Partie matérielle

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

I.4.1.1-Le Microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique.

Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure I.2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.[5]



Le composant CMS



Le composant classique

Figure I.3 Microcontrôleur ATmega328

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

✓ **La mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).

✓ **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.

- ✓ **EEPROM** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [7]

I.4.1.2-Les sources de l'alimentation de la carte

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

- **VIN**. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V**. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- **3V3**. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit integer faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA. [4]



Figure II 4 : Les tension de reference de carte arduino

I.4.1.3-Les entrées & sorties

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` 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 de "rappel au plus" (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)`.



Figure II 5 : les entrées et les sortie numérique

En plus, certaines broches ont des fonctions spécialisées :

- ☒ **Interruptions Externes:** Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. -Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- ☒ **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- ☒ **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou `TWI - Two-Wire interface - interface "2 fils"`).
- ☒ **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

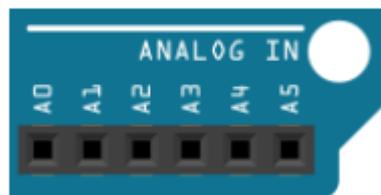


Figure II 6 : les entrées et les sortie analogique

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité).

Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé. [3]

I.4.1.4-Les ports de communications

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL).

Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil:

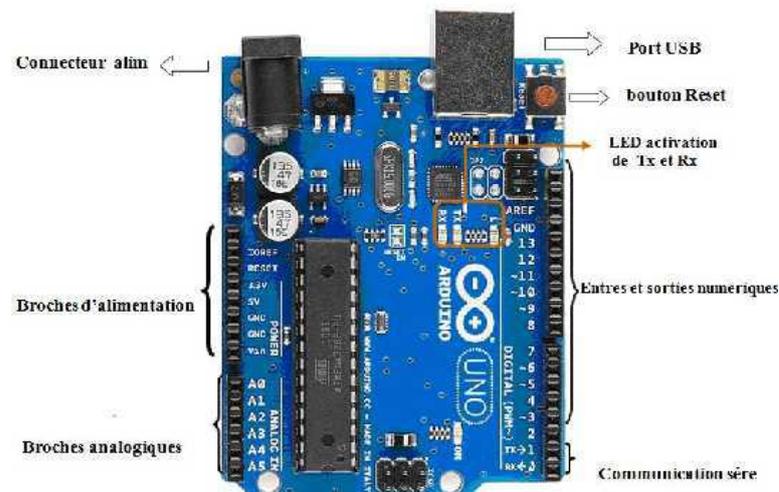


Figure II.7 Constitution de la carte Arduino UNO

I.4.2-Partie programme

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte.

L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

I.4.2.1-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 carte à travers 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. [8]

I.4.2.2-Structure générale du programme (IDE Arduino)

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

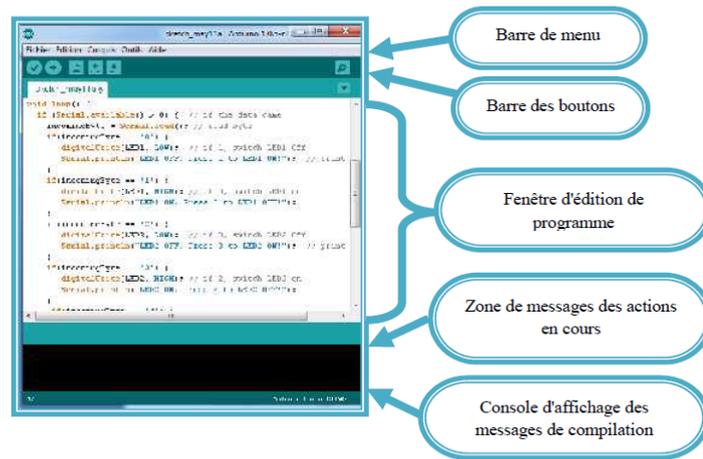


Figure II.8 Interface IDE Arduino

I.4.2.3-Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM 3) comme à titre d'exemple cette figure suivante.

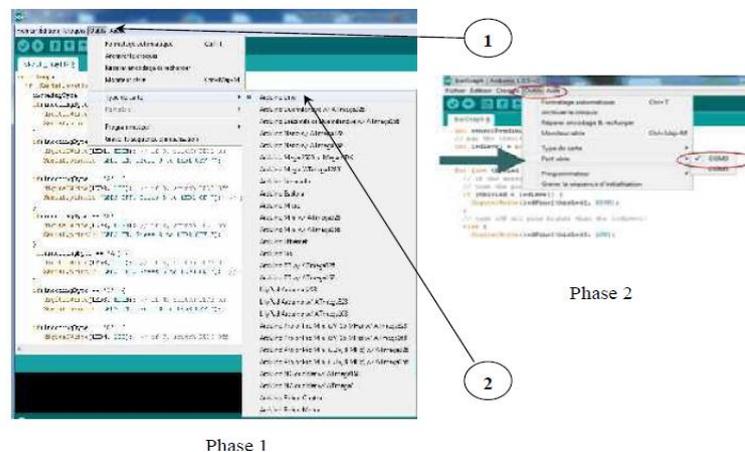


Figure II.9 Paramétrage de la carte

I.4.2.4-Description du programme

Un programme arduino est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

Commentaires

Les commentaires sont, en programmation informatique, des portions du code source

ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

```
1 /* programme de command 4 LED avec Smartphone via Bluetooth-----
2 *et fait également clignoter la diode de test de la carte-----
3 */-----
```

Définition des variables

Pour notre montage, on va utiliser une sortie numerique de la carte qui est par exemple la 3 éme sortie numerique ; cette variable doit être définie et nommée ici LED1 4 ; la syntaxe est pour désigner un nombre entier est **int**.

```
int LED1 = pin 4; // mettre la LED au pin 4-----
```

Configuration des entres et des sorties void setup ()

les broches numeriques de l'arduino peuvent aussi bien être configurées en entrées numeriques ou en sorties numeriques; ici on va configurer pin4 en sortie ; pin mode (nom,état) est une des quatre fonctions relatives aux entrées – sorties numériques.

```
5 void setup() {-----
6 // mettre le pin 4 comme sortie:-----
7 pinMode(4, OUTPUT); // lorsque le pin 4 est activé la LED1 étient-----
8 }-----
```

Programmation des interactions void loop :

Dans cette boucle ,on définit les opérations à effectuer dans l'ordre **digital write** (nom, état) est une autre des quatre fonctions relatives aux entrées – sorties numériques.

- **delay** (temps en mili-seconde) est la commande d'attente entre deux instructions.
- chaque ligne d'instruction est terminée par un point virgule.
- ne pas oublier les accolades qu' encadrent la boucle.

```
9 void loop() {-----
10 digital write ( LED1 1,HIGH); -----
11 delay (3000)-----
12 digital Write(LED1, LOW);-----
13 delay (1000)-----
14 }-----
```

I.4.2.5- Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.



Figure II.10 Les étapes de téléchargement du code

I.5- Les Accessoires de la carte Arduino

La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations.

I.5.1 Communication

Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de communications ; on peut éclaircir actuellement quelques types.

I.5.1.1 Le module Arduino Bluetooth

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion série Bluetooth à la place d'une connexion USB, très faible consommation d'énergie, très faible portée (sur un rayon de l'ordre d'une dizaine de mètres),(faible débit, très bon marché et peu encombrant.



Figure II.11 Type de modules Bluetooth

I.5.1.2 Le module shield Arduino Wifi

Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.



Figure II.12 Module shield wifi

I.5.1.3 Le Module XBee

Ce module permet de faire de la transmission sans fil, faible distance /consommation /débit/ prix. [9]



Figure II.13 Module XBee

I.5.2 Les capteurs

Un capteur est une interface entre un processus physique et une information manipulable. Il ne mesure rien, mais fournit une information en fonction de la sollicitation à laquelle il est soumis. Il fournit cette information grâce à une électronique à laquelle il est associé. [9]

I.5.2.1- Capteur de gaz/fumée

Le capteur de gaz inflammable et de fumée MQ9 détecte la concentration des gaz combustibles dans l'air et renvoie sa lecture comme tension analogique. La sonde peut mesurer des concentrations du gaz inflammable de 300 à 10.000 ppm (*parts per million*). Le capteur peut fonctionner à des températures de -20 à 50°C et consomme moins de 150 mA à 5 V.[1]

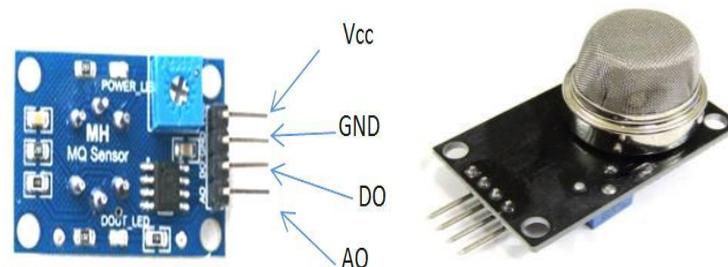


Figure II-14 : Capteur de gaz/fumée

I.5.2.2- capteur de température

Le capteur analogique de température, elle est capable de mesurer des températures allant de -55°C à +150°C, de quoi mesurer n'importe quelle température.

La sortie analogique du capteur est proportionnelle à la température. Il suffit de mesurer la tension en sortie du capteur pour en déduire la température. Chaque degré Celsius correspond à une tension de +10mV.[10]



Figure II-15 : Capteur de température

I.5.2.3- capteur de distance

Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet.

Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter. [11]

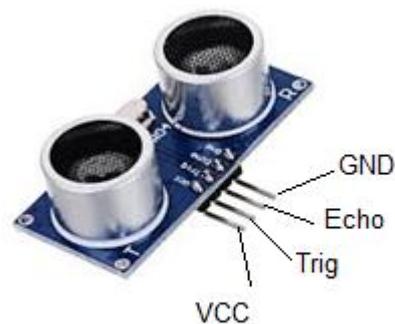


Figure II.16 capteur de distance

I.5.3 Les drivers

Il existe plusieurs drivers comme des cartes auxiliaires qui peuvent être attachées avec l'Arduino afin de faciliter la commande ; on peut citer quelques types.

I.5.3.1- Des moteurs électriques

Un servomoteur est un actionneur rotatif ou linéaire qui permet un contrôle précis de la position angulaire ou linéaire, de la vitesse et de l'accélération. Il se compose d'un moteur approprié couplé à un capteur pour le retour de position.[17]



Figure II.17 Moteurs électriques

I.5.3.2- Les afficheurs LCD

Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage des paramètres de fonctionnement.

Ces Afficheurs permettent d'afficher des lettres, des chiffres et quelques caractères spéciaux. Les caractères sont prédéfinis. [9]



Figure II.18 Afficheurs LCD

I.5.3.3- Module de Relais

Il s'agit d'une carte d'interface de relais, qui peuvent être contrôlé directement par un large éventail de microcontrôleurs comme Arduino, AVR, PIC, ARM, API, etc.

Ce module de relais est bas actif 5V. Il est également capable de contrôler appareils divers et autres équipements avec le grand courant. Cette interface standard peut être connectée directement avec les microcontrôleurs. Le voyant rouge qui indique l'état de travail est propice à l'utilisation de sécurité. Le module de relais est largement utilisé pour tout contrôle MCU, le secteur industriel, contrôle PLC, contrôle de la maison intelligente.

Dans notre cas, nous avons choisi un module de relais à 8 canaux car nous avons besoin de contrôler de l'éclairage de 4 pièces ainsi de la ventilation.[1]



Figure II.19 : Module de relais

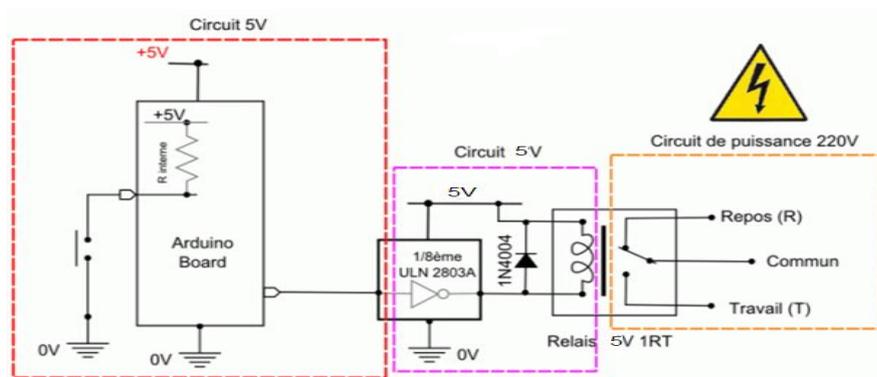


Figure II.20 : Principe de fonctionnement du relais.

Conclusion

Dans ce chapitre, nous avons projeté la lumière sur une carte arduino donnant ainsi les raisons pour lesquelles on l'a choisie, puis nous avons cité des différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; (la partie matérielle et la partie de programmation) plus précisément. Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques.

Le chapitre suivant sera consacré à l'étude Le d'un application android..

Chapitre III

Développement

d'une application

androïde

Introduction

L'androïde est parmi les derniers systèmes d'exploitation qui développent les exigences des téléphones intelligents. La plateforme androïde de smart phone devient de plus en plus importante pour les réalisateurs de logiciel, en raison de ses puissantes possibilités et open source.

Lors des années précédentes, le traitement des données informatiques se fait par des ordinateurs ; en revanche le smart phone a des avantages qui ont les mêmes fonctions que l'outil informatique ; ce dernier porte l'intérêt de l'ordinateur grâce à l'androïde. La téléphonie mobile a connu une explosion dans les années 2000 mais aucune révolution n'a semble 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 mois, 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, marche d'applications en ligne. Autant de qualités permettant de créer des applications innovantes et de les distribuer en toute simplicité. La plate-forme Androide 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.

III.1.le système Androïde

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. [12]

III.2.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. [12]

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. [12]

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.[13]

III.2.1.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 beta d' App Inventor diffusé par le MIT encore en version beta aujourd'hui.

III.2.2. 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 » [14]

III.2.3.Un commencement avec App Inventor

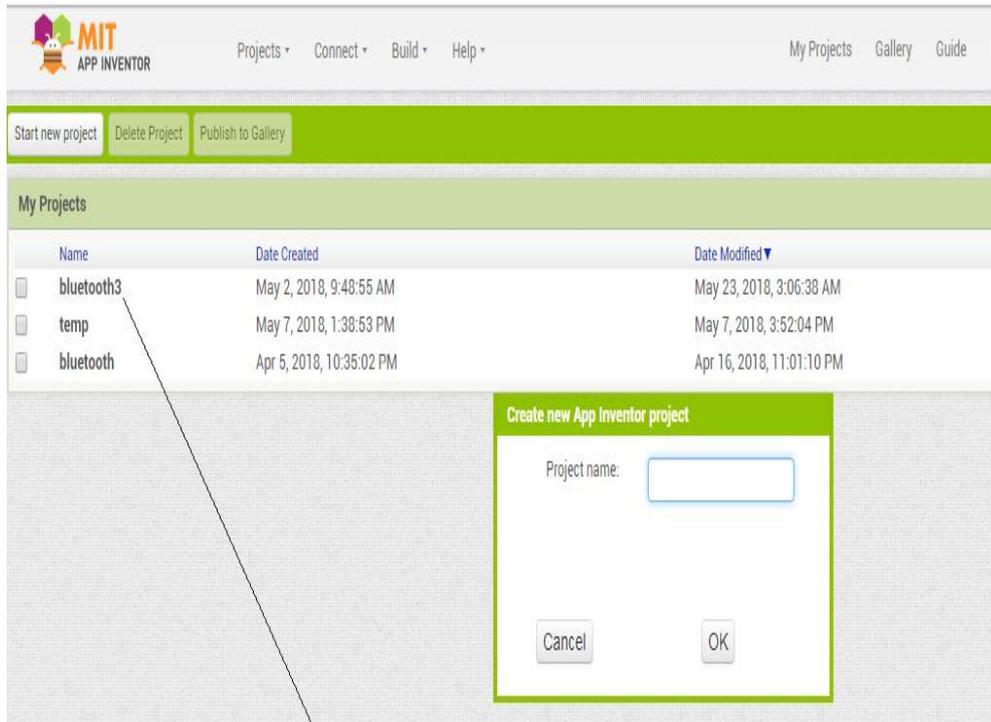
Google fournit gratuitement un kit de développement (SDK) prévu pour s'intégrer (sous la forme d'un Plugin) à l'environnement de développement Eclipse (libre). Il permet de développer des applications codées en langage Java pour les différentes versions d'Androïde.

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 : [15].

Cela peut-être assez long parfois car App Inventor est lancée depuis le Web en mode Cloud.

4. Créer un nouveau projet :

Cliquer sur My Project (en haut à gauche) / New / Project Name (sans espace) / OK.



Notre application

Figure III.1 Création de nouveau projet sur App Inventor

Donc le projet est sauvegardé automatiquement sur notre compte Google.

III.2.4. Structure d'une application App Inventor

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

III.2.4.a. L'interface graphique :

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

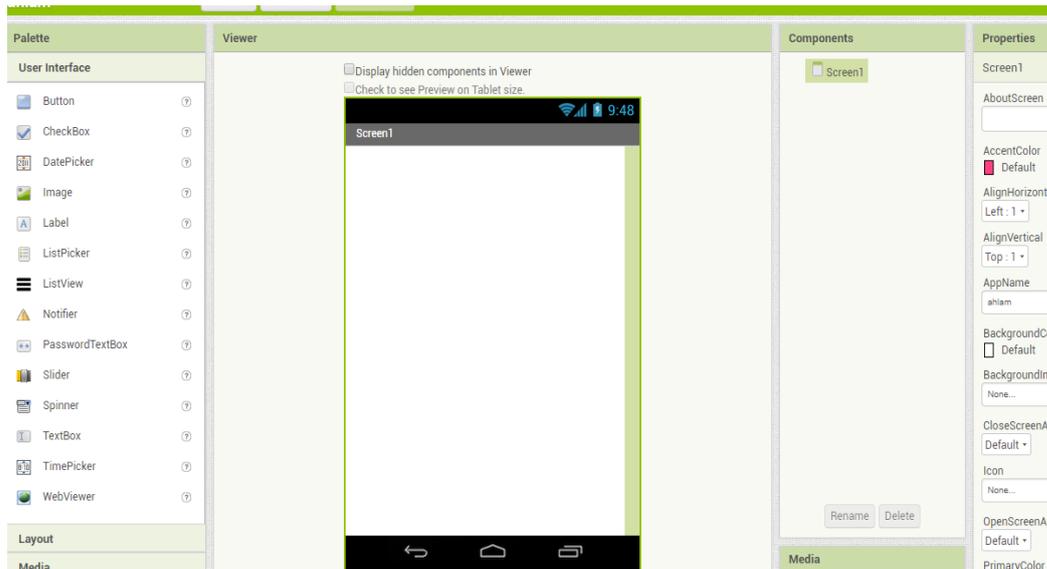


Figure III.2 Première interface de la création App Inventor

Cette partie contient des composants (visibles ou invisibles) [12].

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.

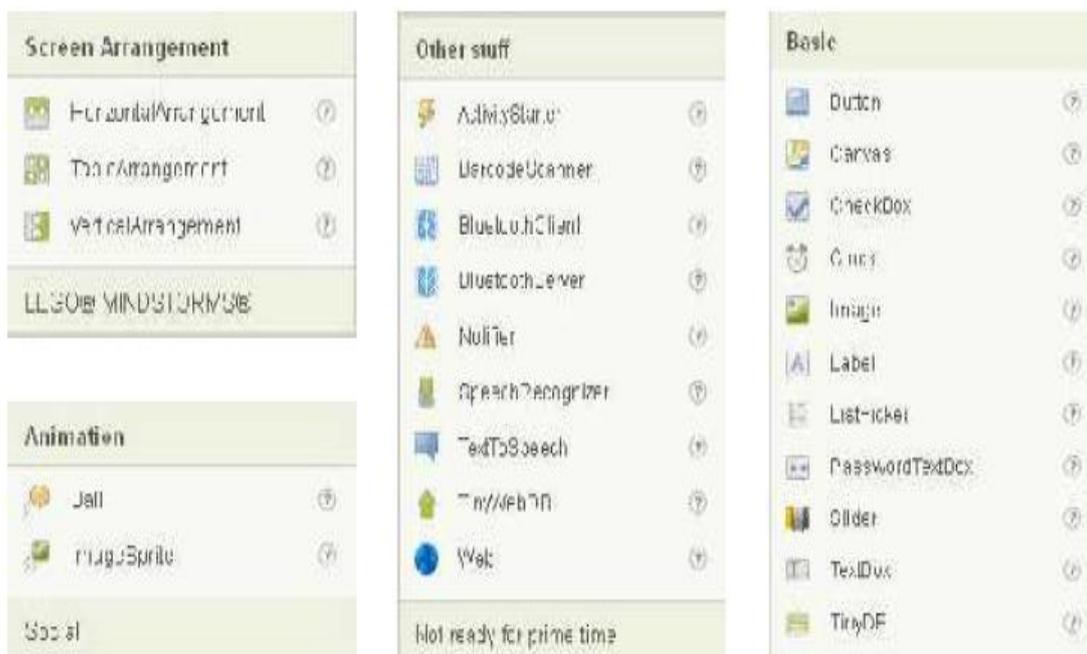


Figure III.3 Les composants graphiques

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 elements et des medias utilises sur l'ecran.

Partie 4 :

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

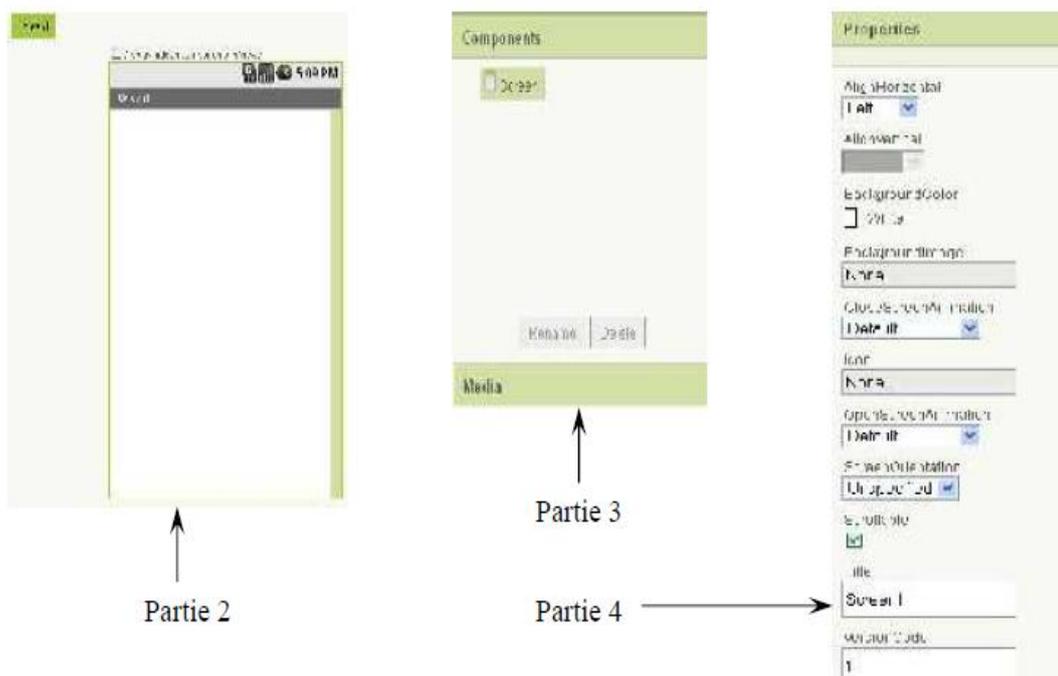


Figure III.4 Les propriétés et les classes sous App Inventor

Pour ajouter un bouton sur l'écran d'emulateur 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**.

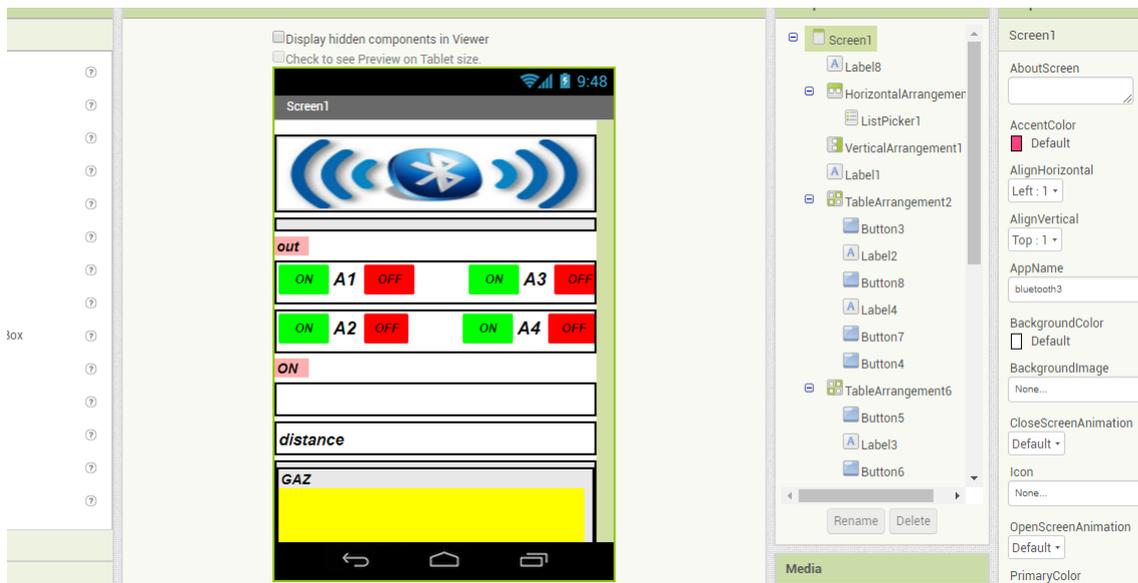
Pour ajuster la place du bouton, on utilise la propriété **Width** de notre exemple ; on utilise le paramètre **Fill parent**.

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



Figure III.5 Ajustement d'un bouton sous App Inventor

Après l'assemblage des différents composants qui constituent notre application, on peut résumer cette phase « *Scratch* » par la figure qui suit et cela pour le Screen1 .



III.2.4.b. 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**. [12]

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

App Inventor, lance le Blocks Editor depuis le serveur du MIT et propose 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, [16]

Comme indiqué sur la figure (III.11) 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.7 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 :

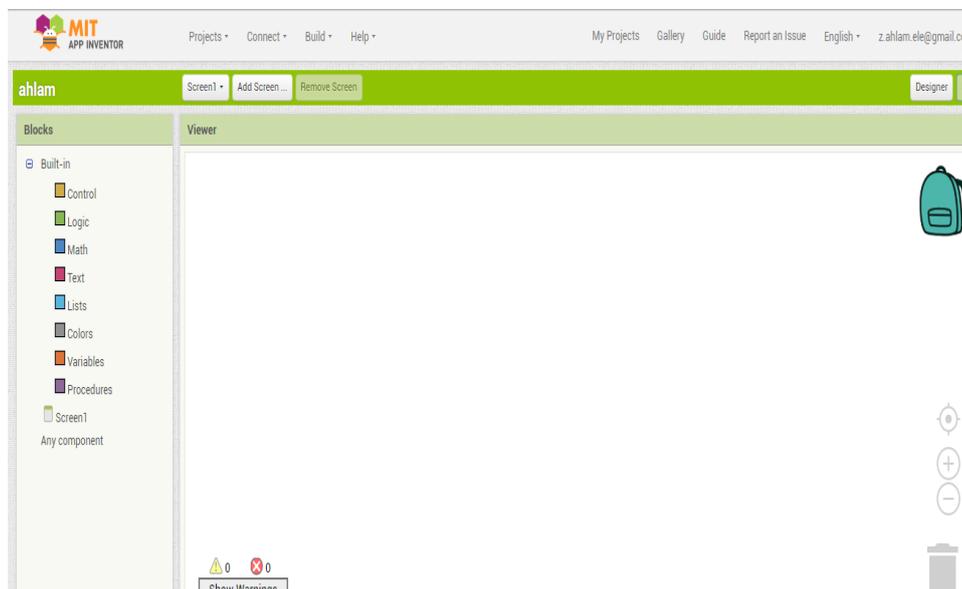


Figure III. 8 Editeurs de blocks App Inventor

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 .

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

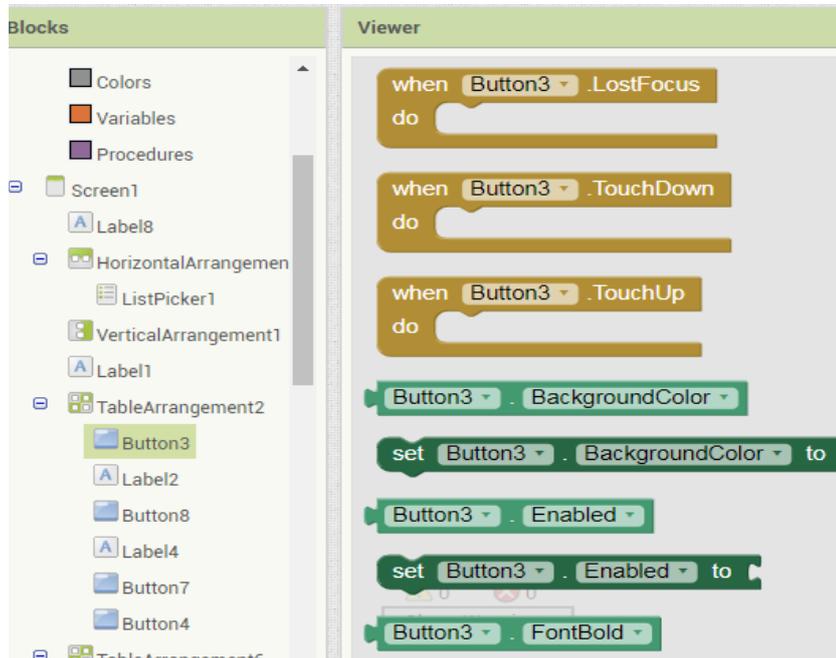


Figure III.9 Echantillon d'un composant sous App Inventor

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

1. L'emulateur : un ecran 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).

conclusion

App Inventor est une opportunité pour initier des projets en phase avec la génération des digital natives, et donner envie aux élèves de poursuivre leurs études. Dès le départ, les initiateurs du projet l'ont constaté, App Inventor transforme les élèves de consommateurs en créateurs : ils adorent créer des applications pour leur téléphone

Chapitre VI

Réalisation

De système

domotique

Introduction

Dans ce chapitre, on présentera de manière sommaire une vue d'ensemble du dispositif expérimental " réalisation d'un système domotique basé sur Bluetooth ".

Ce travail d'initiation a base d'une carte Arduino UNO et module bluetooth permet de commander une éclairage de maison et des appareils électroménagers et Surveiller la température et le l'état de gaz, l'entrée de la maison aussi par technologie bluetooth via le système Androïde.

Après avoir donné dans le chapitre précédent une description théorique sur l'application android on va procéder à la réalisation d'un système et les blocs utilisées.

IV.1.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 android.

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

1. Chercher les différentes structures constituants notre maquette et qui vont avec les objectifs fixés et les moyens disponibles.
2. Présenter les différents éléments en choisissant des composants aux caractéristiques voulus, à défaut, on choisira ceux disponibles sur le marché.

Dans la deuxième partie « réalisation pratique », on passe par les deux étapes suivantes:

1. Présenter les différentes étapes de la réalisation pratique de la système.
2. On assemble ensuite les composants suivants notre montage sur la carte arduino : module bluetooth, module relais, LM35, MQ09 et ultrasonic 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).

IV.2.Schéma general de système

Le schéma général de notre dispositif est indiqué par la figure (VI.1). Pour faciliter cette étude, on a décomposé le schéma synoptique en quatre blocs :

- ☒ Arduino UNO;
- ☒ Bloc qui envoyé des commandes ;
- ☒ Bloc qui recevoir des données;
- ☒ Bloc de controle et commandes;

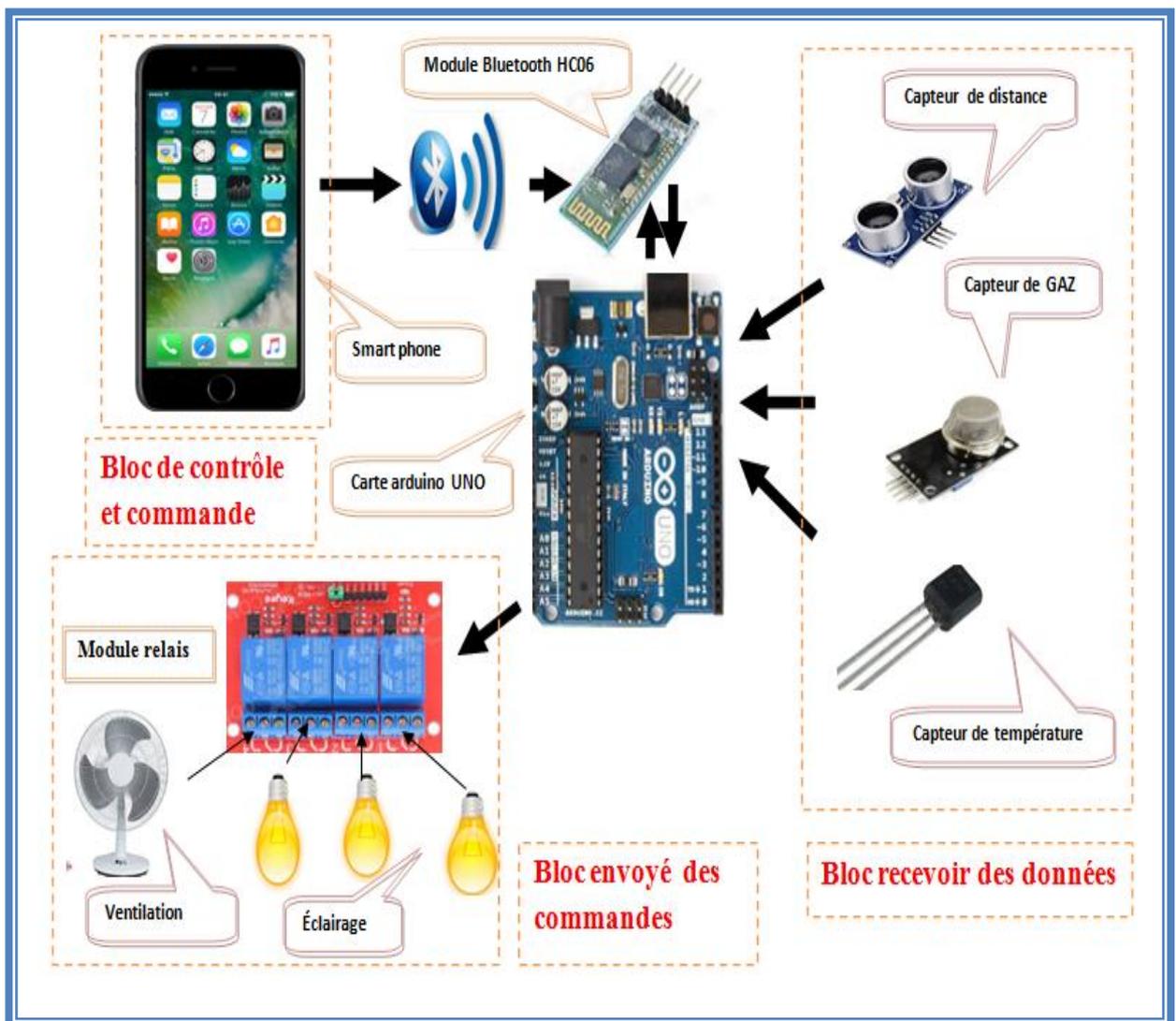


Figure IV 1 Schéma globale du système domotique à concevoir

IV.2.1.Arduino UNO

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

IV.2.1.1. Partie hardware :

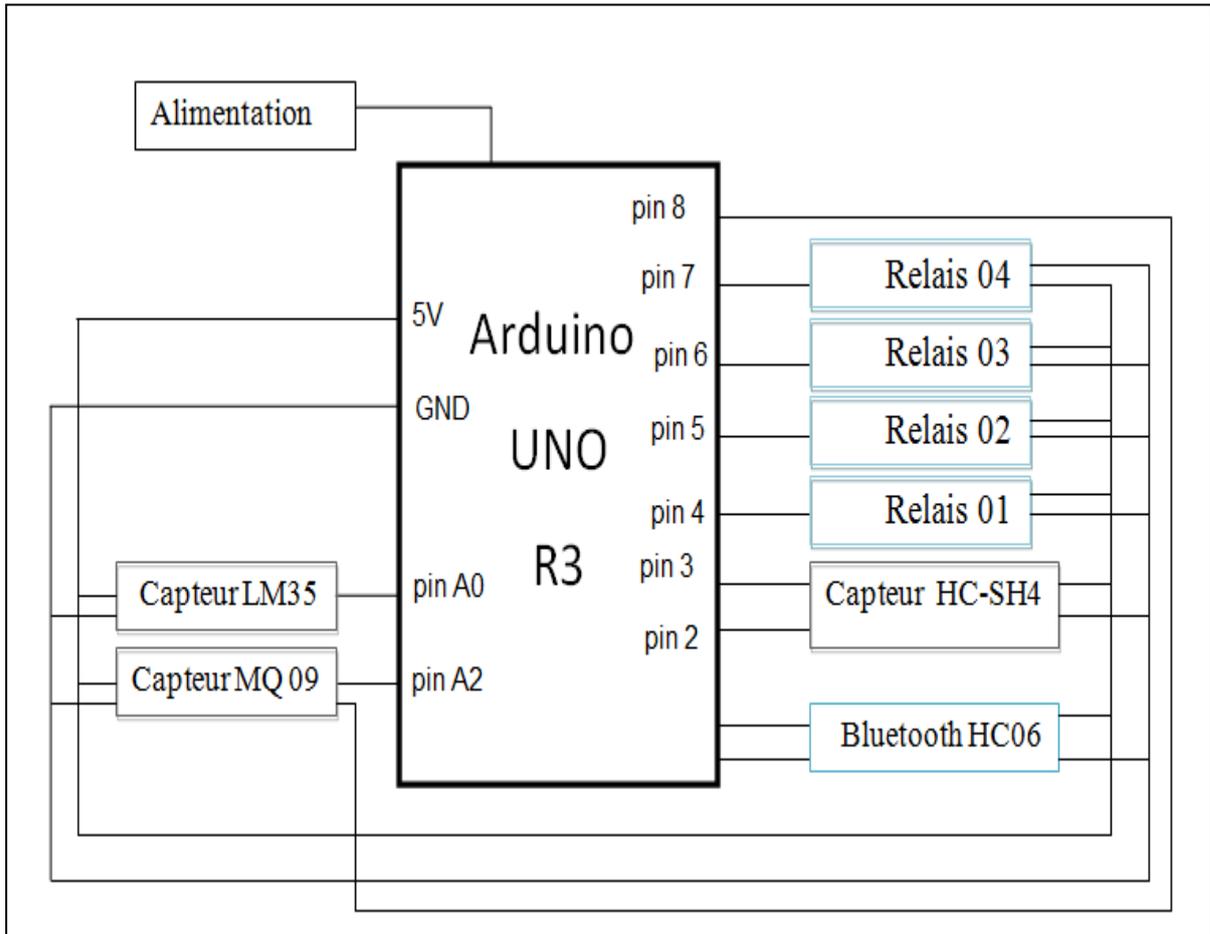


Figure IV 2 brochage de la carte arduino UNO

IV.2.1.2. Partie software :

✓ Organigrammes des fonctions

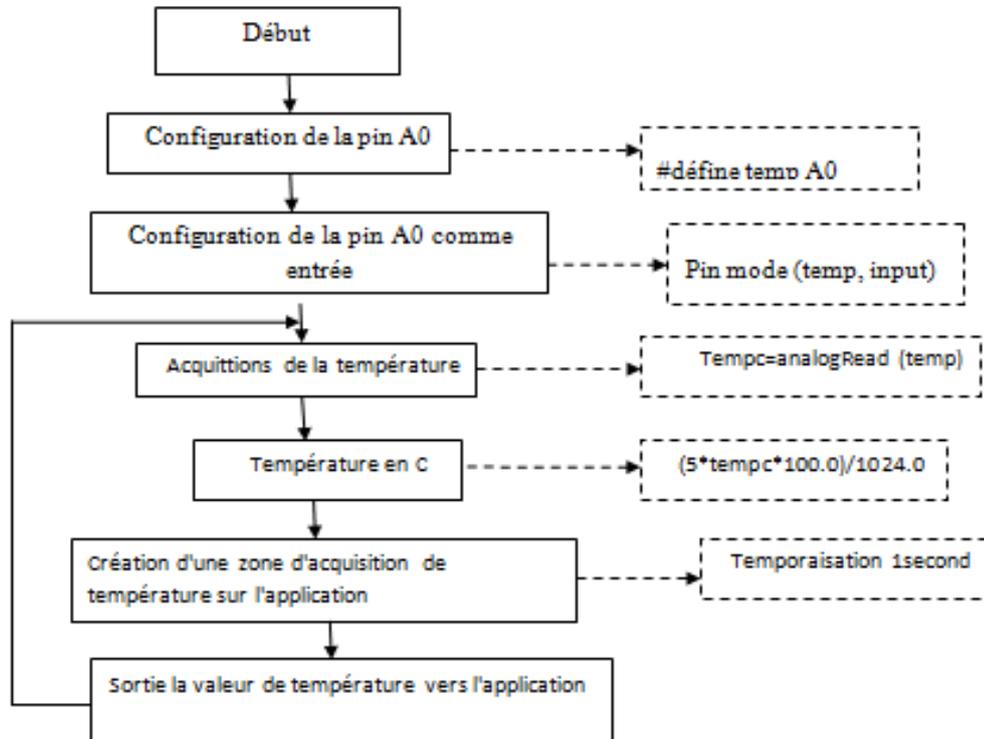


Figure IV 3 Organigramme de la fonction d'acquisition de temperature

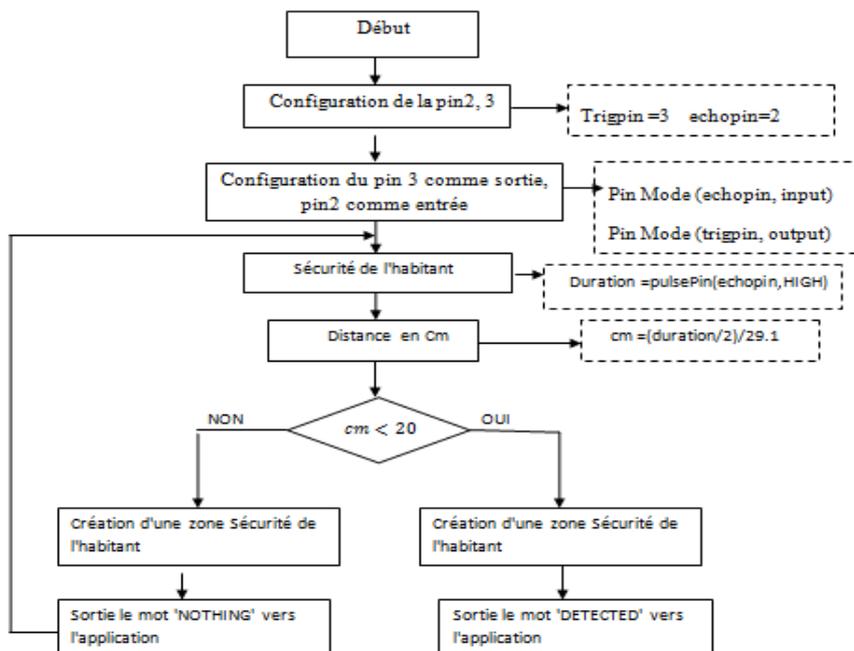


Figure IV 4 Organigramme de la fonction sécurité del'habitant

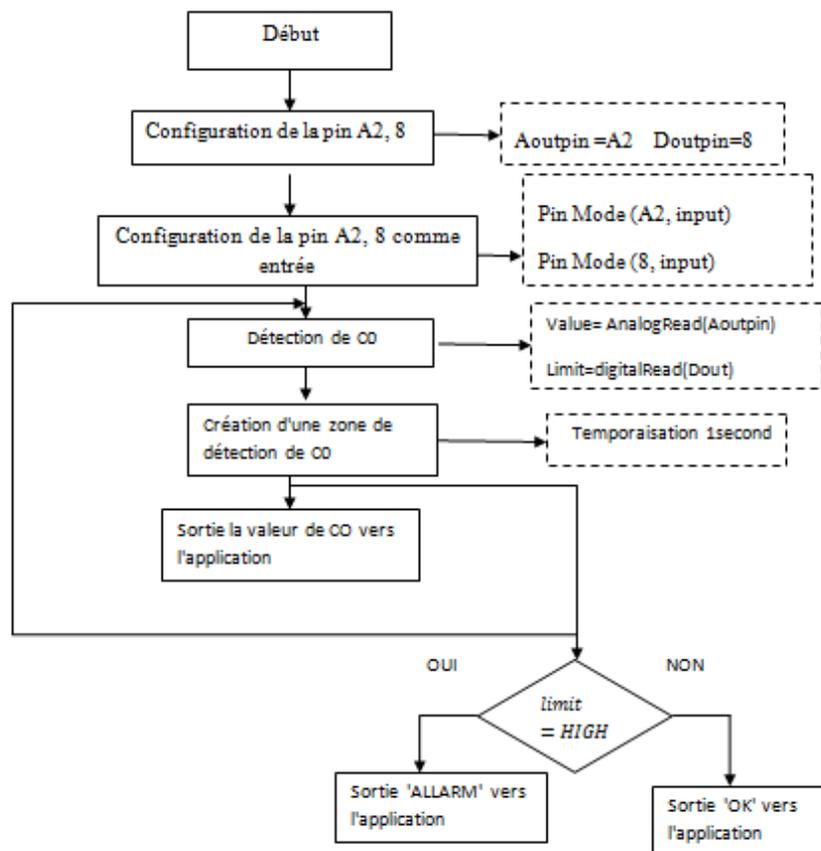


Figure IV 5 Organigramme de la fonction detection de CO

✓ Présentation du programme IDE

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

```

sketch_may23a
int data;

float tempc;
int tempx;
int tempPin = 0;
const int trigPin = 3; const int echoPin = 2; long duration, cm;
const int AOUTpin=A2;
const int DOUTpin=8;
const int ledPin=13;
int limit;
int value;

void setup()
{
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  //initialisation serial communication
  //at 9600 bits per second:
  Serial.begin(9600);
}
  
```

```

sketch_may23a
//at 9600 bits per second:
Serial.begin(9600);
}

void loop()
{
  if(Serial.available())
  {
    // read the data
    int data = Serial.read();
    delay(100);
    if(data=='1')
    digitalWrite(4, HIGH);
    if(data=='2')
    digitalWrite(4, LOW);
    if(data=='3')
    digitalWrite(5, HIGH);
    if(data=='4')
    digitalWrite(5, LOW);
    if(data=='5')
    digitalWrite(6, HIGH);
    if(data=='6')
    digitalWrite(6, LOW);
    if(data=='7')
    digitalWrite(7, HIGH);
    if(data=='8')
    digitalWrite(7, LOW);
    delay(100);
  }
  {
    tempc = analogRead(tempPin);
    tempc = (5.0 * tempc * 100.0) /1024.0;
    tempx = (int)tempc;
    Serial.print("temp:");
    Serial.print(tempx);
    Serial.print(" c*");
    delay(1000);
  }

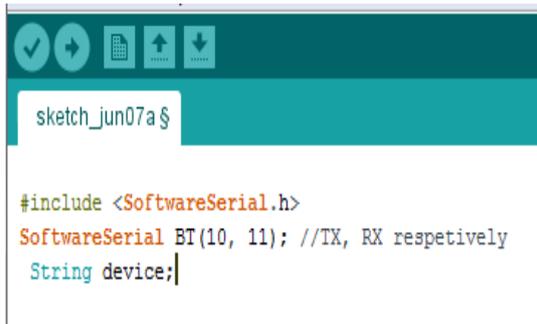
  {
    digitalWrite (trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite (trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite (trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    cm = (duration/2)/29.1;
    if(cm < 20)
    {
      Serial.print("d"); delay(400);
    }
    else
    {
      Serial.print("n"); delay(400);
    }
  }
  {
    value= analogRead(AOUTpin);
    limit= digitalRead(DOUTpin);
    Serial.print("CO value: ");
    Serial.println(value);
    Serial.print("limit: ");
    Serial.println(limit);
    delay(100);
    if (limit == HIGH){
      Serial.println("OK");
    }
  }
}

```

Figure VI 6 Présentation graphique du programme

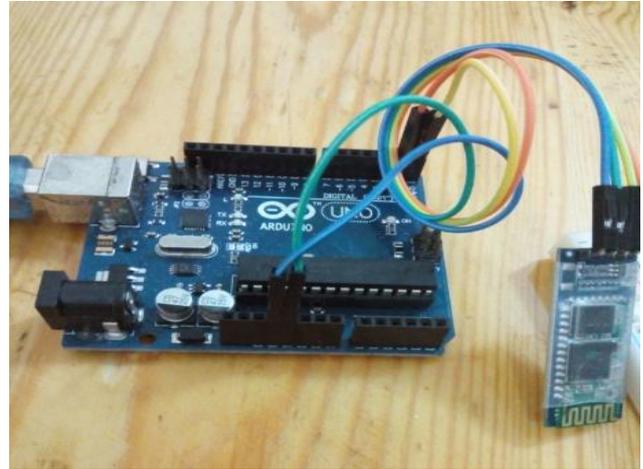
Dans cette partie, on peut classer l'accessoire Bluetooth de l'Arduino comme une suite de bloque de commande puisque il prend la relève de la validation des signaux emis par le smart phone vers l'arduino.

Nous avons utilisé un modèle nommé **HC-06**.



```
sketch_jun07a $
#include <SoftwareSerial.h>
SoftwareSerial BT(10, 11); //TX, RX respectively
String device;
```

Partie IDE



Partie matérielle

Figure IV.7 brochage carte arduino avec module Bluetooth HC06

IV.2.2.Bloc qui envoyé des commandes

Cette partie est représentée dans la Fonction de gestion d'éclairage et ventilation. Elle se compose de quatre relais électromécaniques .

Chaque relais est excité dans sa bobine par un signal généré par l'arduino.

Cette partie a été testée avec quatre LED qui s'allume et s'éteint à savoir le signal émis par le smart phone.

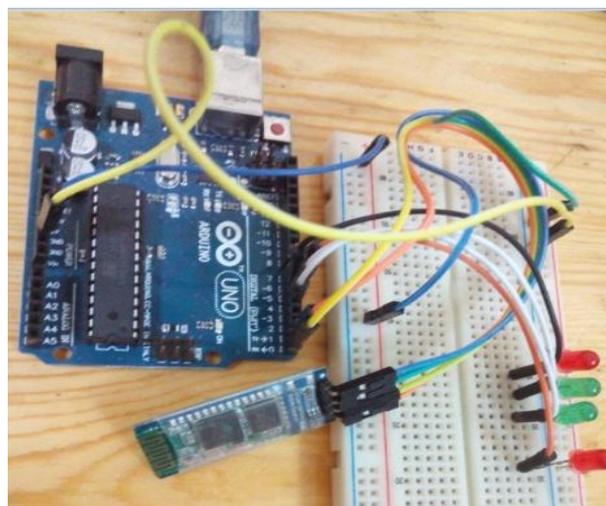


Figure IV.8 Test le bloc par les LED

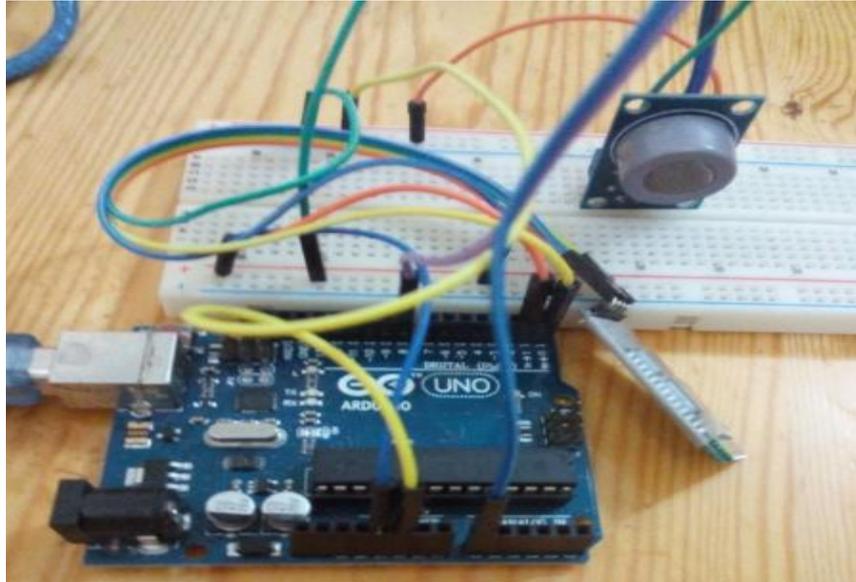


Figure IV.10 brochage de carte arduino avec capteur de GAZ MQ-09

IV.2.3.3.Fonction de sécurité de l'habitant

Cette fonction vous permet de détecter tout mouvement près de la maison via au capteur ultrasonique HC-SR04. ce module ultrason permettant l'évaluation d'une distance de 2cm à 400cm. La mesure est réalisée "sans contact" et dispose d'une précision de 3mm dans les conditions optimales.

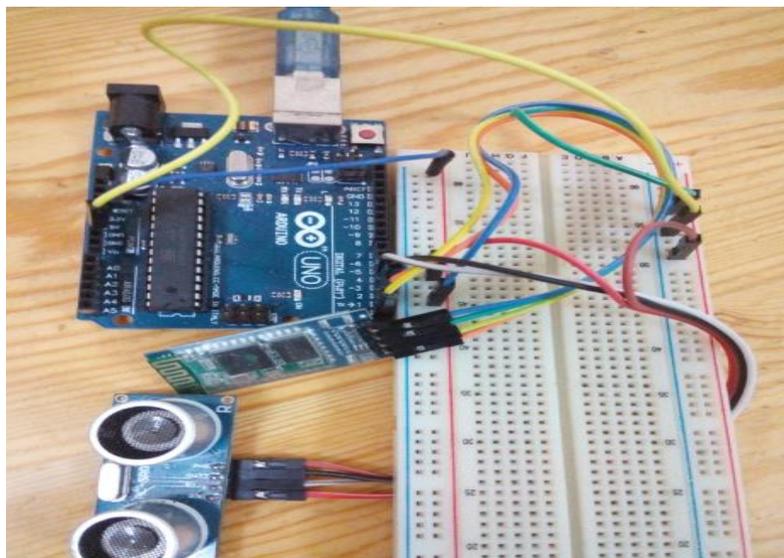


Figure IV.11 brochage de carte arduino avec capteur de distance HC-SR04

IV.2.4.Bloc de controle et commandes

Le bloc concerné est le Smart phone, nous avons crée un application contient deux partie : la première est capable de gérer une commande directe vers quatre relais (la commande domotique) et la deuxième recevoir des donnés des trios grandeurs (tempérrteur ,gaz et distance) et ce à travers le Bluetooth.

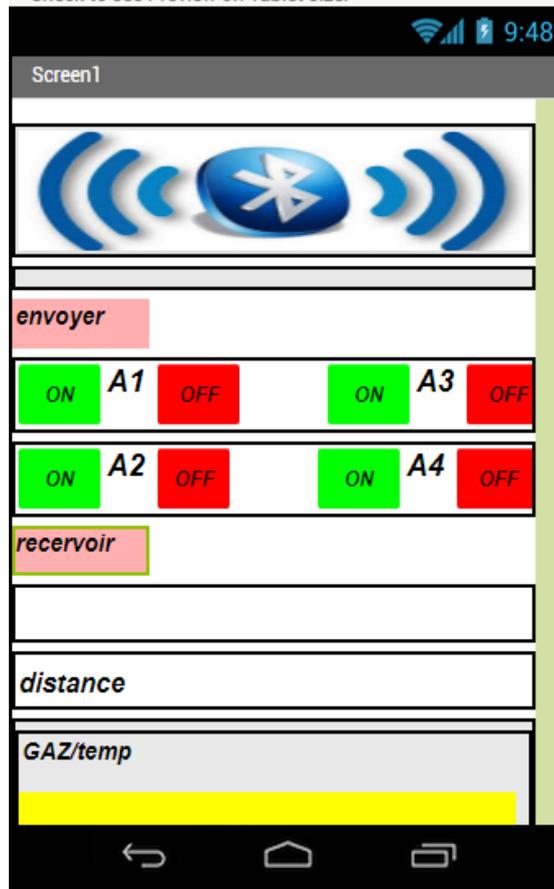


Figure IV.12 Schéma global du Screene

IV.2.4.1.Editeur de blocs

```

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do if call BluetoothClient1 .Connect
      address ListPicker1 . Selection
then set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
    
```

```

when Clock1 . Timer
do
  if BluetoothClient1 . IsConnected
  then
    set Label8 . Text to "connected"
    set Label8 . TextColor to green
    set Label14 . Text to call BluetoothClient1 . ReceiveText
    numberOfBytes call BluetoothClient1 . BytesAvailableToReceive
    if call BluetoothClient1 . BytesAvailableToReceive > 0
    then
      set Label7 . Text to call BluetoothClient1 . ReceiveSignedBytes
      numberOfBytes call BluetoothClient1 . BytesAvailableToReceive
      if call BluetoothClient1 . ReceiveText == "d"
      then
        set Label11 . Text to " "
        set Label11 . BackgroundColor to gray
        set Label10 . Text to "DETECTED"
        set Label10 . BackgroundColor to green
  
```

```

set Label11 . Text to "NOTHING"
set Label11 . BackgroundColor to red
set Label10 . Text to " "
set Label10 . BackgroundColor to gray

if not BluetoothClient1 . IsConnected
then
  set Label8 . Text to "not connected"
  set Label8 . TextColor to red
  
```

<pre> when Button3 . Click do call BluetoothClient1 . SendText text "1" </pre>	<pre> when Button7 . Click do call BluetoothClient1 . SendText text "5" </pre>
<pre> when Button4 . Click do call BluetoothClient1 . SendText text "2" </pre>	<pre> when Button8 . Click do call BluetoothClient1 . SendText text "6" </pre>
<pre> when Button5 . Click do call BluetoothClient1 . SendText text "3" </pre>	<pre> when Button9 . Click do call BluetoothClient1 . SendText text "7" </pre>
<pre> when Button6 . Click do call BluetoothClient1 . SendText text "4" </pre>	<pre> when Button10 . Click do call BluetoothClient1 . SendText text "8" </pre>

Figure IV 13 Schéma global du Scratch

IV.2.4.2. Explication et démarche

Notre interface contient une touche d'activation et désactivation du Bluetooth donc avant d'envoyer n'importe quel caractère à travers les boutons « bluetooth », il faut d'abord s'assurer que le bluetooth est allumé ; Ensuite, nous pouvons envoyer des commandes et recevoir des données.

Cette figure montre que notre application nommée « bluetooth3 » peut être installée sur smart phone, sa taille est 14.84 mégas comme première version.

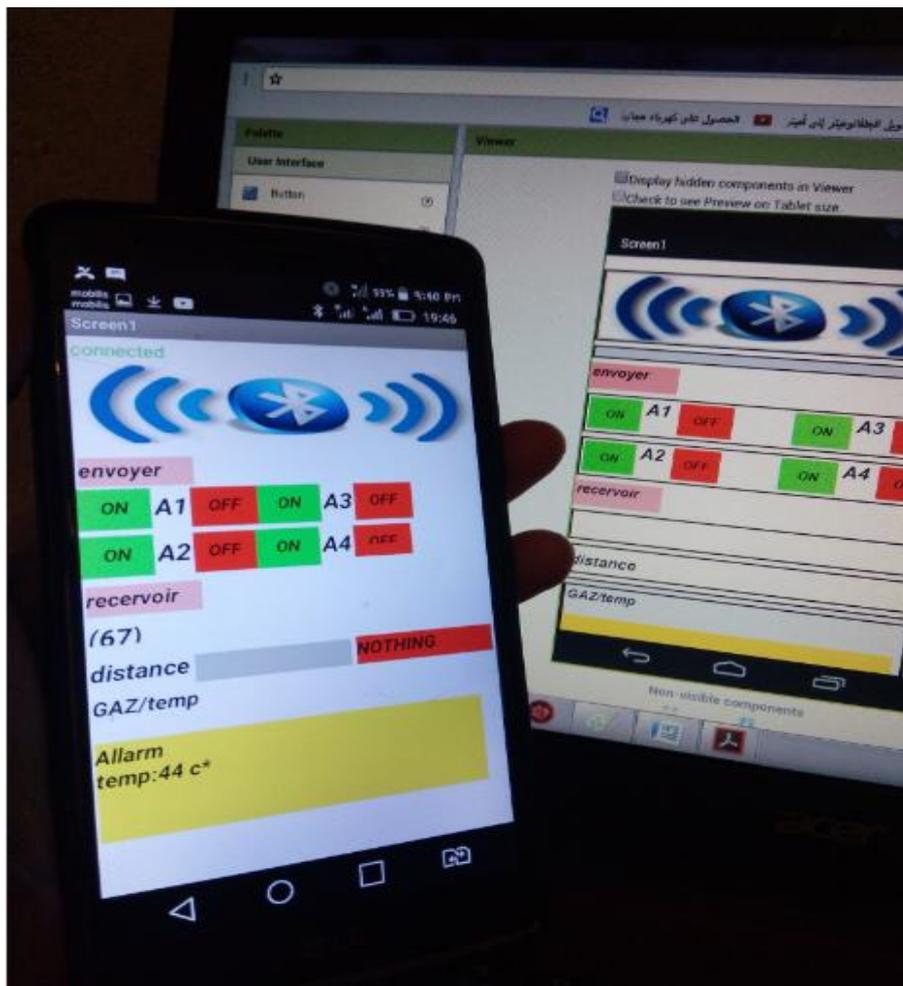


Figure IV.14 Photo réelle de notre application bluetooth3

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 ; d'une autre part une simple élaboration de l'environnement App Inventor qui nous pousse à réaliser une application « *.apk » sous smart phone capable de lier une carte Arduino et un smart phone afin d'exécuter des ordres bien définis.

Conclusion générale

La réalisation des maisons intelligentes a été au coeur de plusieurs projets de recherche lors de ces dernières années. D'énormes progrès ont été accomplis grâce aux avancées en intelligence artificielle, à la miniaturisation de dispositifs électroniques pour la domotique et au développement des réseaux de communication, malgré tout, il reste encore plusieurs défis à surmonter pour rendre possible l'implémentation des maisons intelligentes en situation réelle.

Dans notre projet, on a essayé de mettre en oeuvre un système qui répond à des fonctions principales de la domotique à savoir la gestion d'éclairage, la sécurité de l'habitant, l'acquisition de la température à l'intérieur de l'habitat dont le but de contrôler la ventilation.

Ce qui nous a permis de faire une connexion à distance entre les différents organes du système et l'application androïde via la technologie bluetooth.

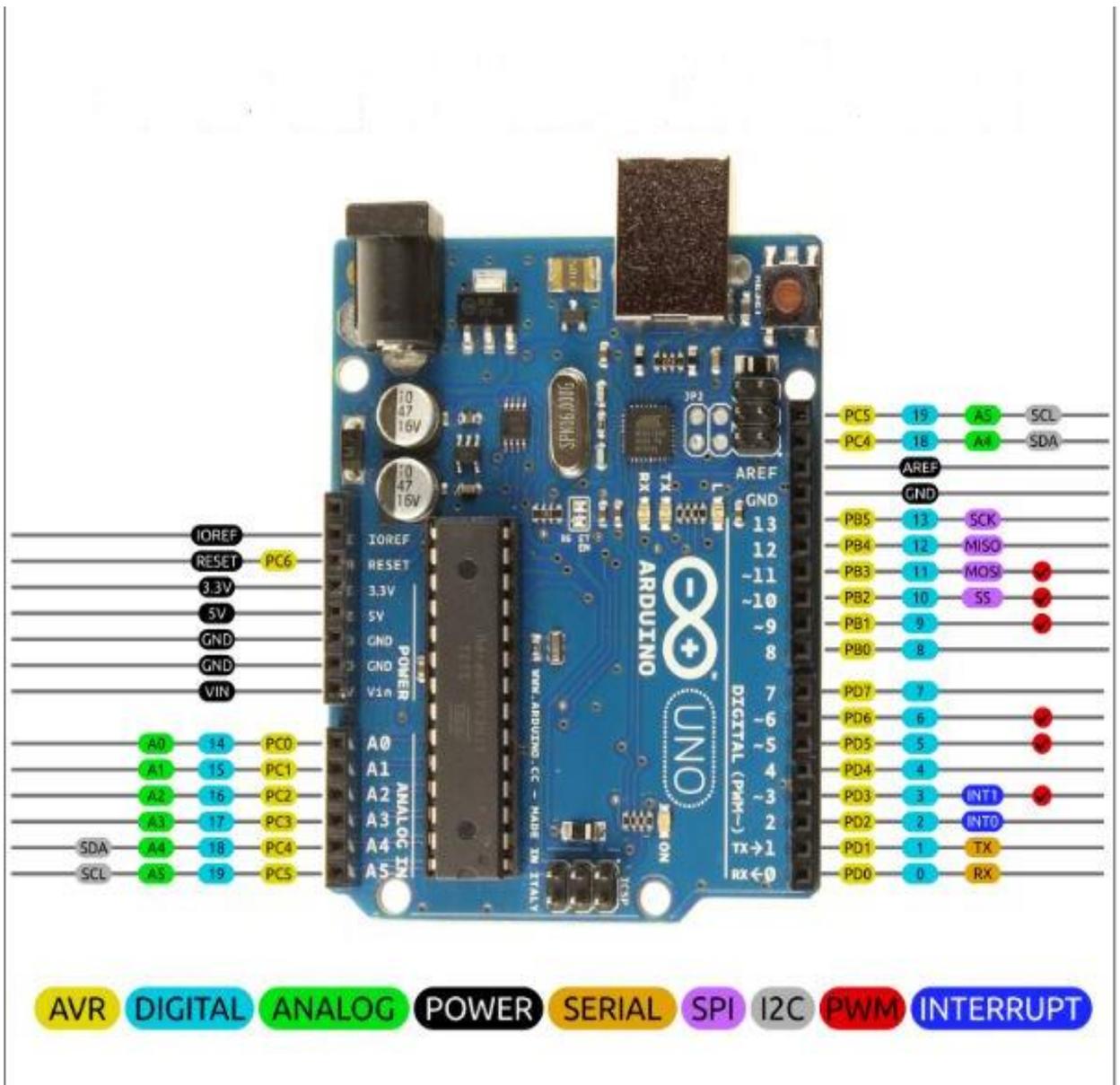
Pour ce qui est des perspectives de continuation plusieurs voies peuvent être envisagées. À titre indicatif, nous conseillons d'examiner et de développer les points suivants :

- ✓ Développer une application Androïde au lieu de celle réalisée dans ce projet de telle façon à enrichir le système de contrôle avec d'autres fonctions domotiques.
- ✓ Piloter la domotique à partir des données reçues d'un réseau domotique et d'un système d'analyse sonore ayant la capacité de reconnaître les ordres vocaux. Les actions exécutées par le contrôleur se produisent en réponse aux commandes vocales prononcées par l'habitant ou de manière proactive pour assurer sa sécurité et son confort.
- ✓ Dans le cas de la reconnaissance de la situation de l'habitant, connaître la position de la personne et sa localisation précise dans une pièce.
- ✓ Réaliser le contrôle domotique en utilisant un module GSM qui permet d'augmenter la possibilité de surveiller l'habitat tout en communiquant via des commandes envoyés par SMS.

Conclusion générale

- ✓ Ajouter une fonction de détection d'intrusion tout en utilisant une caméra de surveillance et avec un traitement d'images pour la détection facial des personnes.

Annexes



Brochage de la carte Arduino UNO

Les Logiciels Utilisés



Pour programmer Arduino



Pour l'application APK



Photos de la phase de réalisation de maison intelligente

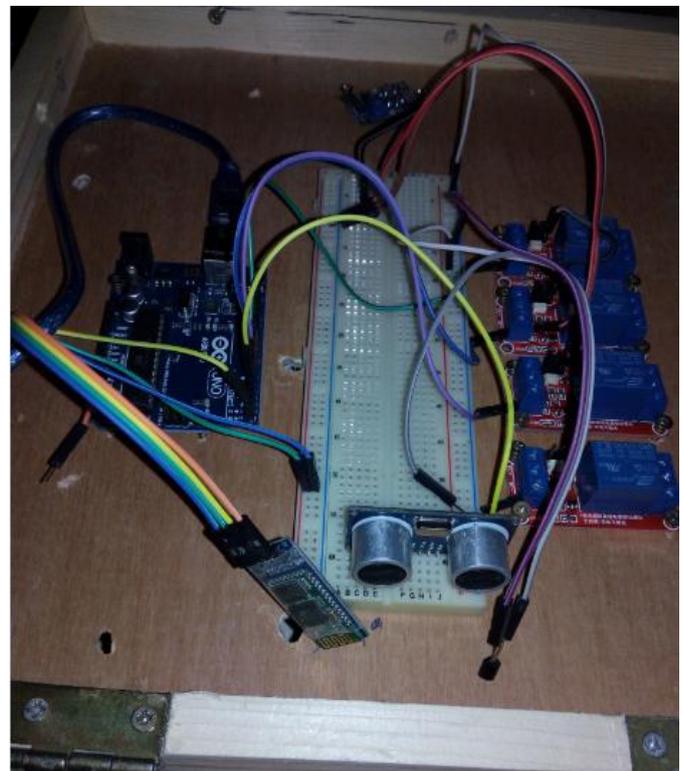
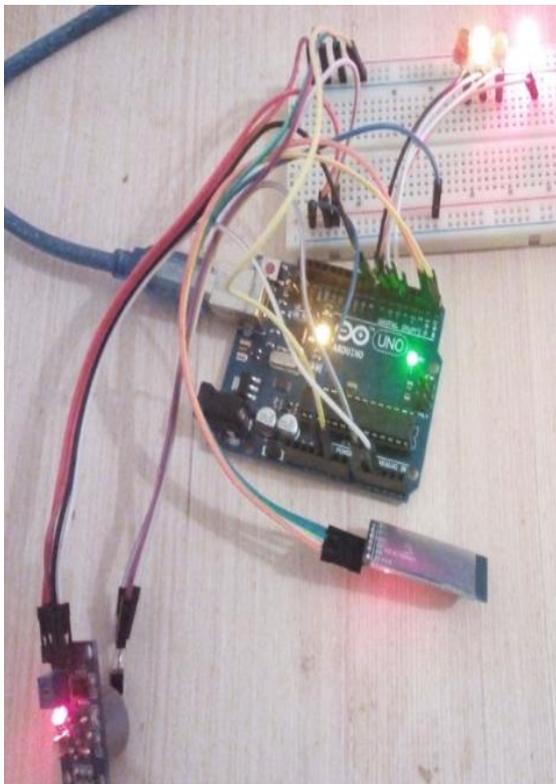


Photo de la réalisation pratique

Bibliographes

[1] HAMID HAMOUCHE , " Conception et réalisation d'une centrale embarquée de la domotique " , Universities Mohammed V de Rabat ,2015

[2] <http://lewebpedagogique.com/isneiffel/files/2017/05/Bluetooth.pdf>

[03] <http://www.generationrobots.com/fr/152-arduino>.

[04] S.V.D.Reyvanth, G.Shirish, « PID controller using Arduino ».
Arduino» .

[05] C. Tavernier, « Arduino applications avancées ». Version Dunod.

[06] X.HINAULT. www.mon-club-elec.fr.

[07] <http://www.acm.uiuc.edu/sigbot/tutorials/2009-11-17-arduino-basics>.

[08] Jean- Noël, « livret Arduino en français » , centre de ressources art sensitif .

[09] Eskimon, Olyte « Arduino pour bien commencer en électronique et en programmation ».

[10] <https://www.carnetdumaker.net/articles/mesurer-une-temperature-avec-un-capteur-lm35-et-une-carte-arduino-genuino>.

[11] Module de détection aux ultrasons - Lucien Bachelard - Le 28 novembre 2015

[12] sig.fgranotier.info/IMG/pdf/debuter_app_inventor.

[13] <http://scratch.mit.edu/>.

[14] <http://blogpeda.ac-poitiers.fr> .

[15] KRAMA Abdelbasset GOUGUI Abdelmoumen , " Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde " Universities de Wargla , 2014/2015

[16] <http://coursstimartinique.fr/>.

[17] <https://en.wikipedia.org/wiki/Servomotor>