

UNIVERSITE KASDI MERBAH OUARGLA

Faculté des Sciences Appliquées  
Département de Génie Electrique



Mémoire

MASTER ACADEMIQUE

Domaine : Sciences et technologies

Filière : Electrotechnique

Spécialité : Electrotechnique Industrielle

Présenté par :

**LAOUAR Bachir**

**AYACHI AMOR Hamza**

Thème :

# **Etude et Réalisation d'une Commande Domotique par ARDUINO Via Infrarouge**

Soutenu publiquement

Le : 04/06/2017

Devant le jury :

M<sup>lle</sup> **ZEHANI Soraya**

MC (A) Président

UKM Ouargla

M<sup>r</sup> **REZOUG Mohamed Redha**

MA (A) Encadreur/rapporteur

UKM Ouargla

M<sup>r</sup> **BENMAKHOULF Abdesslam**

MA (A) Examineur

UKM Ouargla

Année universitaire 2016/2017

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# Remerciements

En préambule à ce mémoire

Nous remerciant ALLAH qui nous aide nous donne la patience et le courage et nous donne le tout puissant, de nous avoir donné la santé et la volonté pour compléter ce modeste travail.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nous tenant à remercier sincèrement Mr **REZOUG MOHAMED REDHA**, en tant que Encadreur, qui a toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, nous avoir fait profiter de ses qualités aussi bien sur le plan scientifique que sur le plan professionnel. Un grand merci pour son sérieux, sa patience et son aide.

Nos remerciements s'adressent également aux membres de jury Mlle **ZEHANI Soraya** (président) et à Monsieur **BENMAKHOLOUF Abdesslam** (examinateur) d'avoir accepté de lire et d'évaluer notre mémoire.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.

Merci à tous.

# Dédicace

Ma Mère, Mon Père

Affable, honorable, aimable : vous représentez pour moi

Le symbole de la bonté par excellence, la source de tendresse

et l'exemple du dévouement qui n'a pas cessé

de m'encourager et de prier pour moi.

Soyez sûrs que je continuerai mon chemin.

Je vous dédie ce travail en témoignage de mon profond amour.

Puisse Dieu, le tout puissant, vous préserver et vous accorder santé,

longue vie et bonheur.

A mes sœurs

A mes frères

En témoignage de l'attachement, de l'amour et de l'affection que je porte pour vous.

A tous les membres de ma famille, petits et grands

A tous mes amis de proches

A tous les amis (es) d'études.

Aux habitants de la ville OUED SOUF en général.

Veillez trouver dans ce modeste travail l'expression de mon

Affection.

**BACHIR**

**HAMZA**

<b>Nomenclateurs</b>	<b>IV</b>
<b>Liste des figures</b>	<b>V</b>
<b>Liste des tableaux</b>	<b>VII</b>
<b>Introduction générale</b>	
1 Généralités	01
2 Position du problème	01
3 Objectif du projet	01
4 Présentation du mémoire	02
<b>CHAPITRE I</b>	
<b>Les protocoles de communication et la logique ATMEL</b>	
I.1 Introduction	03
I.2 La domotique	03
I.2.1 Les domaines de la domotique	03
I.2.2 Technique de la domotique	04
I.2.2.1 La centrale de commande	04
I.2.2.2 L'écran de contrôle	05
I.2.2.3 Le pilotage à distance	05
I.3 Evaluation du protocole de communication	06
I.3.1 Communications série	06
I.3.2 Communication parallèle	07
I.3.3 Communication Bluetooth	08
I.3.4 Communication infrarouge	08
I.4 Choix de la communication	09
I.4.1 Historique de l'infrarouge	09
I.4.2 Fonctionnement	09
I.4.3 Spécification infrarouge IrDA	09
I.5 Explication du protocole infrarouge	10
I.5.1 Le récepteur de type TSOP1736	10
I.5.2 Simulation IR TSOP1736 sous ISIS	11
I.6 Arduino	11
I.6.1 Historique	12
I.6.2 Exemples d'usages	12

I.6.3 Pourquoi Arduino Mega	13
I.6.4 Les caractéristiques de la carte Arduino Mega 2560	14
I.6.5 Le Microcontrôleur ATmega2560	15
I.7 Description générale du Arduino Mega	17
I.7.1 Le microcontrôleur	17
I.7.2 Les sources d'alimentation de la carte	17
I.7.3 Les ports de la carte	18
I.7.4 Ports de communication	19
I.7.5 Les fonctions IDE de l'environnement Arduino	20
I.7.6 Fonctionnement et utilisation	22
I.8 Conclusion	22

## **CHAPITRE II**

### **Réalisation Du Dispositif Expérimental**

II.1 Introduction	23
II.2 La domotique sur notre plan pratique	23
II.3 Liaison à distance par infrarouge (IR)	23
II.3.1 Le choix de la télécommande infrarouge	23
II.3.2 Système de codage infrarouge	26
II.3.3 La norme RC5	26
II.3.3.1 Constitution d'une trame en code RC5	27
II.3.3.2 Détail d'un bit de la trame RC5	27
II.3.3.3 La durée de la trame RC5	28
II.3.3.4 Codage des trames d'émetteur	29
II.3.3.5 Logique câblée du IR	29
II.4 Liaison manuelle par afficheur tactile (TFT)	32
II.4.1 choix d'afficheur	32
II.4.2 Spécifications	33
II.4.3 Câblage TFT Arduino	34
II.5 La connexion puissance commande	35
II.5.1 la démarche entre commande et puissance	36
II.5.2 Câblage équipement driver	36
II.6 La réalisation virtuelle	38

II.7 Conclusion	40
-----------------	----

### **CHAPITRE III**

#### **Description et analyse du programme**

III. Introduction	41
III.2 Présentation de l'organigramme	41
III.3 La librairie IRremote sous Arduino	43
III.4 Explication du programme	44
III.5 Présentation du programme IDE	45
III.6 Conclusion	49

#### **Conclusion générale**

1 Généralités	50
2 Problèmes rencontrés	50
3 Perspectives du projet	50

#### **Annexes**

Annexe A. Les résultats obtenus expérimentalement	51
Annexe B. Les logiciels Utilisés	52
Annexe C. Le code hexadécimal de notre programme	53
Annexe D. Schéma synoptique du dispositif	54

#### **Bibliographie**

**ANALOG** : Analogique.

**AREF** : Abréviation pour Analog REFérence, référence analogique.

**DIGITAL** : Numérique.

**GND** : Abréviation pour GrouND, la terre. C'est la masse, 0 Volt.

**ICSP** : Abréviation pour In Circuit Serial Programmions, programmation série sur circuit.

**PIN** : Broche.

**POWER** : Puissance, alimentation.

**PWM** : Abréviation de (Pulse Width Modulation), soit Modulation de l'Largeur d'Impulsion.

**RX** : Abréviation pour Reçoive, réception.

**TX** : Abréviation pour Transmit, transmission.

**MISO (Master In Slave Out)** : La ligne de Slave pour envoyer des données au maître.

**MOSI (Master Out Slave In)** : La ligne principale pour l'envoi de données vers les périphériques.

**SCK (horloge de série)** : Les impulsions d'horloge qui synchronisent la transmission de données généré par le maître.

**SS (Slave Select)** : la broche sur chaque périphérique que le maître peut utiliser pour activer et désactiver des dispositifs spécifiques.

**SDA (Serial Data Line)** : ligne de données bidirectionnelle.

**SCL (Serial Clock Line)** : ligne d'horloge de synchronisation bidirectionnelle.

**I2C** : Inter-Integrated Circuit.

**UART (Universal Asynchronous Receiver Transmitter)** : est un émetteur-récepteur asynchrone universel.

**ADC : Analog-to-Digital Converter** : un convertisseur analogique-numérique.

**IGE** : Integrated Development Environment.

**AVR** : Advanced Virtual RISC.

**RISC** : Reduced Instructions Set Computer.

## Liste des figures

<b>Figure</b>	<b>Nom de figure</b>	<b>Numéro de page</b>
Figure (I.1)	Liaison série	07
Figure (I.2)	Liaison parallèle	07
Figure (I.3)	Le boîtier TSOP1736	11
Figure (I.4)	Montage du IR sous ISIS PORTUES	11
Figure (I.5)	Architecture de Harvard pour les Atmel	15
Figure (I.6)	Microcontrôleur Atmega2560	16
Figure (I.7)	La carte Arduino Mega2560	17
Figure (II.1)	la télécommande « Car MP3 »	24
Figure (II.2)	schéma descriptif d'une télécommande infrarouge à 21 canaux	25
Figure (II.3)	le bit du code RC5	28
Figure (II.4)	Détail d'une trame RC5	28
Figure (II.5)	Enchaînement des trames	28
Figure (II.6)	Schéma de base récepteur IR	29
Figure (II.7)	Le support IR réalisé	30
Figure (II.8)	extraction des codes RC5 par Arduino	31
Figure (II.9)	Les codes RC5 sur le COM 5 de l'IDE	32
Figure (II.10)	TFT 2.4	33
Figure (II.11)	Les différentes réponses sur TFT	35
Figure (II.12)	Le driver à huit Relais	36
Figure (II.13)	La liaison équipement driver	37

Figure (II.14)	Les huit équipements domotiques	38
Figure (II.15)	La carte réalisée sous ISIS-PORTEUS	39
Figure (II.16)	Le dispositif domotique réalisé	39
Figure (III.1)	Organigramme principal	42
Figure (III.2)	Organigramme esclave	43
Figure (III.3)	Fenêtre du programme esclave	44
Figure (III.4)	Fenêtre initialisation du TFT	45
Figure (III.5)	Fenêtre initialisation de la librairie « IRremote »	46
Figure (III.6)	Fenêtre du menu du TFT	46
Figure (III.7)	Fenêtre des boucles de la temporisation	47
Figure (III.8)	Fenêtre d'activation et désactivation des équipements domotiques	47

**Liste des Tableau**

<b>Tableau</b>	<b>Nom de Tableau</b>	<b>Numéro de page</b>
Tableau (I.1)	Caractéristique de la carte Arduino Mega	15
Tableau (I.2)	Les broches PCINT	21
Tableau (II.1)	les codes des touches le la télécommande utilisée	30
Tableau (II.2)	Caractéristiques de notre TFT	33
Tableau (II.3)	connexions TFT et Arduino	34

# Introduction Générale

## **1 Généralités**

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

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).

## **2 Position du problème**

Dans la vie moderne, on utilise pas mal d'outils et d'accessoires de commande à distance afin de simplifier notre contrôle, donc nous chercherons toujours à se concentrer sur la souplesse de la commande et de contrôler sur une zone bien définie (notre périphérie) le plus grand nombre possible d'accessoires.

La télécommande occupe la première place d'objets que nous utilisons comme un intermédiaire (homme machine) donc notre travail se concentre sur l'utilisation de cette dernière : aussi bien sur sa liaison avec un système ou une carte de commande (carte d'interface) tel que l'Arduino.

Pour essayer de soulever ce problème, nous avons engagé à ce projet expérimental destiné à réaliser un système de commande et de contrôle à distance par infrarouge afin de simplifier la commande domotique.

## **3 Objectifs du projet**

Dans ce projet, trois objectifs ont été visés :

- Le premier est de regrouper suffisamment d'informations sur les protocoles de communication ainsi que les accessoires de la connectivité: construction interne, principe de fonctionnement ...
- Le deuxième consiste à réaliser un dispositif basé sur une carte nommée Arduino capable d'exécuter des actions de commande et de contrôle et de connecter sans fil avec différentes charges domotiques (lampes, actionnaire, pompe, moteur ...) afin de les activer ou désactiver à distance.
- Le troisième est de réaliser et programmer sous l'environnement du C de Arduino et familiariser avec des accessoires « shield » qui font la logique des systèmes embarqués afin de valider une conception pratique d'une carte de commande domotique.
- Avec Arduino Mega, nous allons commencer par apprendre à programmer puis à utiliser des composants électroniques. En fin de compte, nous saurons créer des systèmes électroniques plus ou moins complexes.

#### **4 Présentation du mémoire**

Pour le développement de ce projet, nous avons réparti le travail en trois chapitres.

Le premier présente une étude descriptive sur les protocoles de communication et les modalités de fonctionnement d'une communication entre deux accessoires, ainsi que des notions de base et des généralités sur les systèmes embarqués. Ce chapitre terminera avec le module qui est (Arduino Mega) ; sa construction, son environnement de programmation et son principe de fonctionnement afin de simplifier son utilisation.

Le deuxième chapitre sera consacré à l'étude et la réalisation du dispositif « virtuel et expérimental », ainsi que la description de chaque bloc du dispositif de commande.

Dans le troisième chapitre, nous allons procéder à la phase logarithmique en expliquant le fonctionnement de notre programme, son déclenchement après le chargement du code simulé dans la carte Arduino et de réaliser une application capable de transférer des codes de commande vers des différentes charges domotiques.

Enfin, on terminera avec une conclusion générale qui résumera l'intérêt de notre étude : les différents résultats obtenus expérimentalement seront donnés sur annexe (A) sous forme de photos donnant ainsi un aperçu sur les performances des cartes d'interfaçage et une idée sur les problèmes à résoudre ultérieurement.

# Chapitre I

**Les protocoles de communication  
et la logique ATMEL**

## I.1 Introduction

Communiquer consiste à transmettre des informations, mais tant que les communicants ne se sont pas attribués à un sens, il ne s'agit que des données et pas d'informations. Les communicants doivent donc non seulement utiliser un langage commun mais aussi maîtriser des règles minimales d'émission et de réception des données. C'est le rôle d'un protocole de s'assurer de tout cela.

Dans ce chapitre, on va faire une synthèse générale sur les différents protocoles de communication puis on mettra la lumière sur un module embarqué (Arduino Mega) et donner une description détaillée (construction, principe de fonctionnement, logique des ports ...).

## I.2 La domotique

Est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans les bâtiments plus ou moins « inter opérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prises électriques, etc.). La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage, de sécurité (alarme), de communications commandes à distance, signaux visuels ou sonores, etc.) que l'on peut retrouver dans les maisons, les hôtels, les lieux publics... [1]

La domotique peut par défaut ou sur commande conditionner l'éclairage et le chauffage d'une pièce ou la mise en route de certains appareils à la présence ou non d'un occupant.

Avec le temps, la domotique tend à sortir de la maison. Elle met par exemple en relation des unités d'habitation entre elles et avec un immeuble (c'est l'immotique) et avec la ville (on entre alors dans « l'urbatique » et ou avec un gestionnaire propriétaire et ou d'autres entités fournissant par exemple des services (eau, énergie, livraison de nourriture, soins à domicile ou à distance, lavage de vêtements, ...). Si ces services visent prioritairement à moins dégrader l'environnement, on parle parfois d'« éco-domotique urbaine ».

### I.2.1 Les domaines de la domotique

Les principaux domaines dans lesquels s'appliquent les techniques de la domotique sont :

- Le pilotage des appareils électrodomestiques, électroménagers par programmation d'horaires ou de macro (suites d'actions programmées réalisées par les appareils électroménagers). Le déclenchement des appareils peut être aussi lié à des événements (détecteurs de mouvement, télécommandes...). [2]
- La gestion de l'énergie du chauffage par exemple est de gérer les apports naturels (calories, frigories, vent, lumière, eau...) en fonction de l'enveloppe thermique du bâtiment, de la climatisation, de la ventilation, de l'éclairage, de l'ouverture et de la fermeture des volets en fonction de l'ensoleillement ou de l'heure de la journée, par exemple : le remplissage d'eau de la baignoire peut s'arrêter automatiquement grâce à un capteur, les robinets de lavabos peuvent ouvrir l'eau à l'approche des mains, etc... Il est également possible de recharger certains appareils électriques (ordinateurs, véhicules électriques...) en fonction du tarif horaire (voir Smart grid). Un compteur communicant peut être intégré dans un smart-grid ou raccordé à un système de télégestion. La régulation programmation du chauffage permet d'importantes économies. [2]
- La sécurité des biens et des personnes (alarmes, détecteur de mouvement, interphone, digicode) ;
- La communication entre appareil et utilisateur par le biais de la « sonification » (émission de signaux sous forme sonore) ;
- Le « confort acoustique ». Il peut provenir de l'installation d'un ensemble de haut-parleurs permettant de répartir le son et de réguler l'intensité sonore ;
- la compensation des situations de l'handicap et de la dépendance. [2]

## **I.2.2 Technique de la domotique**

La domotique est basée sur la mise en réseau par une « centrale de commande » des différents appareils électriques de la maison.

### **I.2.2.1 La centrale de commande**

Programmable et contenant des modules embarqués (passerelles domestiques) ou une interface micro-informatique (écran tactile, serveur, ...): elle joue le rôle d'une « intelligence » centralisée et d'interface (homme machine) centralisée pour l'utilisateur ou des services distants de contrôle. Elle tend à devenir plus réactive aux changements du contexte. Pour cela, elle réunit ou remplace divers appareils (programmeur régulateur de chauffage, centrale d'alarme, système de centralisation des persiennes électriques, contacteur jour/nuit du

cumulus, délesteur, programmeurs horaires, systèmes d'arrosage automatiques, etc.), qui peuvent fonctionner et interagir de manière asynchrone. [2]

### **I.2.2.2 L'écran de contrôle**

Il est fixé dans le domicile où il peut être émulé à distance via le réseau ADSL de la maison puis l'Internet), permettant le pilotage de la maison à distance pour tout ou une partie des fonctions domotiques. L'interface distante peut être par exemple un ordinateur de poche, un téléphone portable ou smart phone, une tablette tactile, une télécommande (universelle ou non), une interface sur télévision connectée, un écran plus une souris... [2]

### **I.2.2.3 Le pilotage à distance**

Il permet de faire face quasiment en temps réel à des situations particulières par exemple :

- Un des enfants de la maison a oublié ses clés, l'un des parents peut depuis un smart phone ouvrir le portail, déverrouiller la porte d'entrée et désactiver l'alarme.
- Une livraison est attendue dans la journée, le propriétaire peut laisser le livreur accéder au garage afin que le colis y soit déposé et communiquer avec lui.

Par ailleurs, l'utilisateur peut programmer certaines fonctions de la maison domestique grâce à cette interface qui est reliée aux appareils connectés. Il peut par exemple :

- Enclencher l'arrosage du jardin à une certaine heure de la journée ;
- Maintenir une température donnée dans la maison ;
- Ouvrir les volets à une heure donnée.

De plus, il est généralement possible par des réglages avancés, d'adapter le système à son propre rythme de vie, (en programmant des « scénarios ») : exemple de scénarios :

- En partant au travail, un simple clic sur un interrupteur installé dans l'entrée enclenche le scénario « départ au travail », l'éclairage s'éteint, le garage s'ouvre, le chauffage se met en veille au bout de 15 minutes, les volets et le garage se ferment après 30 minutes.
- En quittant le travail pour rentrer chez soi, on actionne le scénario de retour à l'aide du téléphone WAP ou depuis l'ordinateur du bureau : les volets s'ouvrent et le chauffage passe en mode confort.

- Quand on est fatigué, on agit sur la télécommande de la maison afin d'enclencher le scénario « relaxation », les lumières se tamisent, un fond sonore apaisant se propage dans la pièce.

Enfin, dans les systèmes les plus complets qui intègrent des capteurs, les scénarios peuvent également se mettre en œuvre automatiquement :

- Le vent se lève et souffle puissamment. Le store de terrasse se relève afin d'éviter que celui-ci soit arraché.
- En été, un détecteur d'ensoleillement informe le système domotique de baisser les stores vénitiens ou les brise-soleil orientables (BSO) afin de maintenir la maison au frais. En hiver, ce même détecteur peut à contrario ouvrir les fermetures de la maison afin de faire entrer la chaleur naturelle et économiser ainsi de l'énergie ;
- La nuit, un détecteur de fumée indique une anomalie chez une personne âgée la centrale domotique alerte un proche afin qu'il puisse prévenir les secours et les volets s'ouvrent afin de faciliter l'évacuation des personnes (dans certains pays, le système peut directement appeler les secours).

Avant d'être réellement installé, un système domotique peut théoriquement être virtuellement modélisé, prototypé et testé afin de vérifier son efficacité et son adaptabilité à des changements de contexte. [2]

### **I.3 Evaluation du protocole de communication**

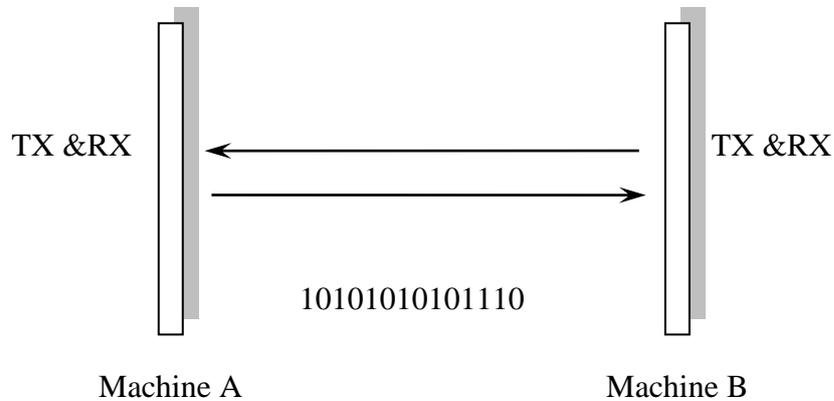
Les ports d'entrée-sortie sont des éléments matériels des systèmes embarqués. l'ordinateur permet au système de communiquer avec des éléments extérieurs, c'est-à-dire d'échanger des données, d'où l'appellation d'interface d'entrées-sorties (notées parfois interface d'E/S).

#### **I.3.1 Communications série**

RS-232 (parfois appelée EIA RS-232, EIA 232 ou TIA 232) est une norme standardisant un bus de communication de type série sur trois fils minimum (électrique, mécanique et protocole) disponible sur presque tous les PC depuis 1962 jusqu'au milieu des années 2000 ; il est communément appelé le « port série ». [3]

Sur les systèmes d'exploitation MS-DOS et Windows, les ports RS-232 sont désignés par les noms COM1, COM2, etc. cela leur a valu le surnom de « ports COM », encore utilisé de nos jours. Le standard RS-232 recouvre plusieurs autres standards : les recommandations UIT-T V.24 (définition des circuits) et V.28 (caractéristiques électriques), ainsi que la norme ISO 2110 pour la connectique. [3]

Les liaisons RS-232 sont fréquemment utilisées dans l'industrie pour connecter différents appareils électroniques (automate, appareil de mesure ...)

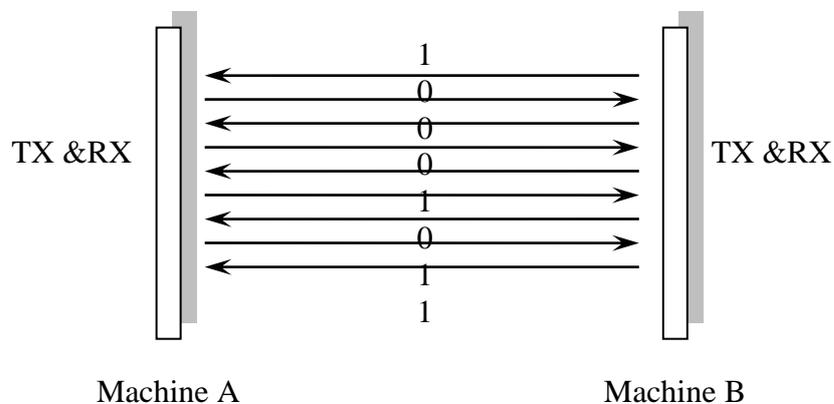


*Figure I.1 Liaison série*

On peut dire que la communication série consiste à transmettre des informations après les avoir préalablement découpées en plusieurs morceaux de taille fixe (cette taille est le nombre de lignes de communications disponibles). Elle s'oppose à la communication parallèle où les informations à transmettre n'ont pas à être découpées avant d'être envoyées (car il y a au moins autant de lignes de communication disponibles que de bits pour transmettre l'information)

### I.3.2 Communication parallèle

La transmission de données en parallèle consiste à envoyer des données simultanément sur plusieurs canaux (fils). Les ports parallèles présents sur les ordinateurs personnels permettent d'envoyer simultanément 8 bits (un octet) par l'intermédiaire de 8 fils. [4]



*Figure I.2 Liaison parallèle*

Le port parallèle est comme le port série intégré à la carte mère. Les connecteurs DB25 permettent de connecter un élément extérieur (une imprimante par exemple). [4]

Un certain nombre de problèmes apparaissent avec la transmission d'un grand nombre de bits en parallèle, comme des problèmes de diaphonie ou de désynchronisation, ce qui peut conduire à préférer la communication série.

À fréquence égale, la communication parallèle a un débit plus élevé. La communication série compense en général cette faiblesse par une fréquence plus élevée.

### **I.3.3 Communication Bluetooth**

Qu'est-ce qu'un Bluetooth ? C'est une technologie de réseau personnel sans fil et est un standard de communication permettant l'échange bidirectionnel de données à très courte distance et utilisant des ondes radio UHF. Son objet est de simplifier les connexions entre les appareils électroniques en supprimant des liaisons filaires. Elle peut remplacer par exemple les câbles entre ordinateurs.

Cette technologie supporte les caractéristiques suivantes :

- Faible coût ;
- Faible puissance d'émission ;
- Courte distance d'émission (quelques dizaines de mètres) ;
- Faible consommation d'énergie (donc adaptée aux produits portables) ;
- Performances modestes (1Mbps) ;
- Topologie ad hoc ;
- Configurable dynamiquement ;
- Support des transferts voix et données ;
- Destinée à un usage personnel ;
- Certification Bluetooth pour assurer la compatibilité des produits entre eux. [5]

### **I.3.4 Communication infrarouge**

Le nom signifie « en dessous du rouge » (du latin infra : « plus bas »), car l'infrarouge est une onde électromagnétique de fréquence inférieure à celle de la lumière visible : le rouge. La longueur d'onde des infrarouges est comprise entre le domaine visible ( $\approx 0,7 \mu\text{m}$ ) et le domaine des micro-ondes ( $\approx 1 \text{ mm}$ ). [6]

L'infrarouge est associé à la chaleur, car à température ambiante ordinaire, les objets émettent spontanément des radiations dans le domaine infrarouge ; la relation est modélisée par la loi du rayonnement du corps noir dite aussi loi de Planck. [6]

La longueur d'onde du maximum d'émission d'un corps noir porté à une température absolue T (en kelvin). Actuellement, il existe plus que ces protocoles déjà cités dont nous avons centrés que les principaux.

#### **I.4 Choix de la communication**

Dans notre travail, on a choisi la communication par port série sous IDE de l'Arduino et la communication infrarouge « IR » parce que c'est une technique simple à réaliser avec l'utilisation de niveaux de tension plus élevés et moins sensibles aux perturbations, ainsi elles peuvent être classifiées comme communication sans fil.

##### **I.4.1 Historique de l'infrarouge**

Cette technologie était utilisée dans la fin des années 1990 et le début des années 2000 notamment pour faire des transferts de fichiers entre des ordinateurs portables, des téléphones mobiles ou des assistants personnels.

Le rayonnement infrarouge (IR) est un rayonnement électromagnétique d'une longueur d'onde supérieure à celle de la lumière visible mais plus courte que celle des micro-ondes.

##### **I.4.2 Fonctionnement**

Pour émettre un signal infrarouge, on sert d'une LED infrarouge, (et pour les capter, on peut se servir d'un phototransistor ou d'une photodiode). Pour transférer les données, on utilise des protocoles de communication binaires comme le SIRCS, inventé par Sony, ou le RC5 de Phillips (le plus utilisé).

##### **I.4.3 Spécification infrarouge IrDA**

Association de données infrarouges est un protocole de communication infrarouge particulier, basé sur le modèle OSI. Il utilise ses propres composants et son propre protocole. Bien qu'il fût énormément utilisé dans les années 1990 et début des années 2000, il a été oublié principalement à l'insu du Bluetooth. Il permettait la communication entre PC, téléphones et PDA...

Cependant, celui-ci fait son retour depuis quelques années, intégré dans les smart phones, afin de transformer ces derniers en télécommande universelle, pour commander des télévisions, des appareils photos à l'aide d'applications.

## I.5 Explication du protocole infrarouge

Etant un standard de transmission des données infrarouge développé par Philips et très répandu en Europe est connu sous le nom de « code RC5 ». Ce code peut transmettre 2048 commandes diverses divisées en 32 groupes adressables de 64 commandes chacun. Avec ce système, chaque appareil se voit attribuer une adresse individuelle, de sorte que le réglage du son de la télé ne risque pas par exemple d'influencer le volume de la chaîne stéréo. Le code envoyé se compose d'un mot de données de 14 bits et est construit comme indiqué ci après :

- 2 bits de départ pour ajuster le niveau de la commande automatique du gain AGC (Auto Gain Control) dans le circuit intégré de réception.
- 1 bit de basculement indique une nouvelle transmission de données.
- 5 bits d'adressage du système.
- 6 bits d'instruction.

La valeur du bit de basculement change à chaque nouvelle activation d'une touche pour distinguer entre une pression nouvelle ou continue sur la même touche. Les 5 bits d'adressage qui suivent indiquent quel appareil doit réagir à la commande. La commande proprement dite est transmise en dernier. Les commandes du code RC5 sont codées en biphases, c'est-à-dire qu'un bit est composé de 2 demi-bits alternés. La combinaison bas/haut caractérise un bit positionné à 1 et la combinaison haut/bas un bit remis à 0. Chaque bit a une longueur de 1,778 ms, donc un protocole dure en tout 24,889 ms. Le code RC5 fait partie des protocoles les mieux documentés. Les adresses 7 et 13 entre autre sont intéressantes pour les appareils de construction « maison » car elles ont été conçues dans des buts expérimentaux.

### I.5.1 Le récepteur de type TSOP1736

Le récepteur infrarouge doit être en mesure de faire la différence entre la lumière infrarouge qui est omniprésente dans l'environnement et celle émise par l'émetteur. C'est un circuit électronique capable de faire une telle distinction. La lumière infrarouge qu'il détecte est particulière : elle doit être pulsée à une fréquence compatible à celle du récepteur. Il existe sur le marché une large gamme de récepteurs infrarouge. Chaque variante est caractérisée par sa fréquence de réception. La plupart des télécommandes émettent une fréquence de porteuse entre 30 et 40 KHz. L'un des récepteurs infrarouges que l'on trouve sur le marché est le récepteur de type TSOP1736. [7]



Figure I.3 Le boîtier TSOP1736

### I.5.2 Simulation IR TSOP1736 sous ISIS

Les ondes lumineuses infrarouges sont en général modulées à une fréquence située entre 30 et 40 KHz lors des transmissions de trames et/ou bits d'information.

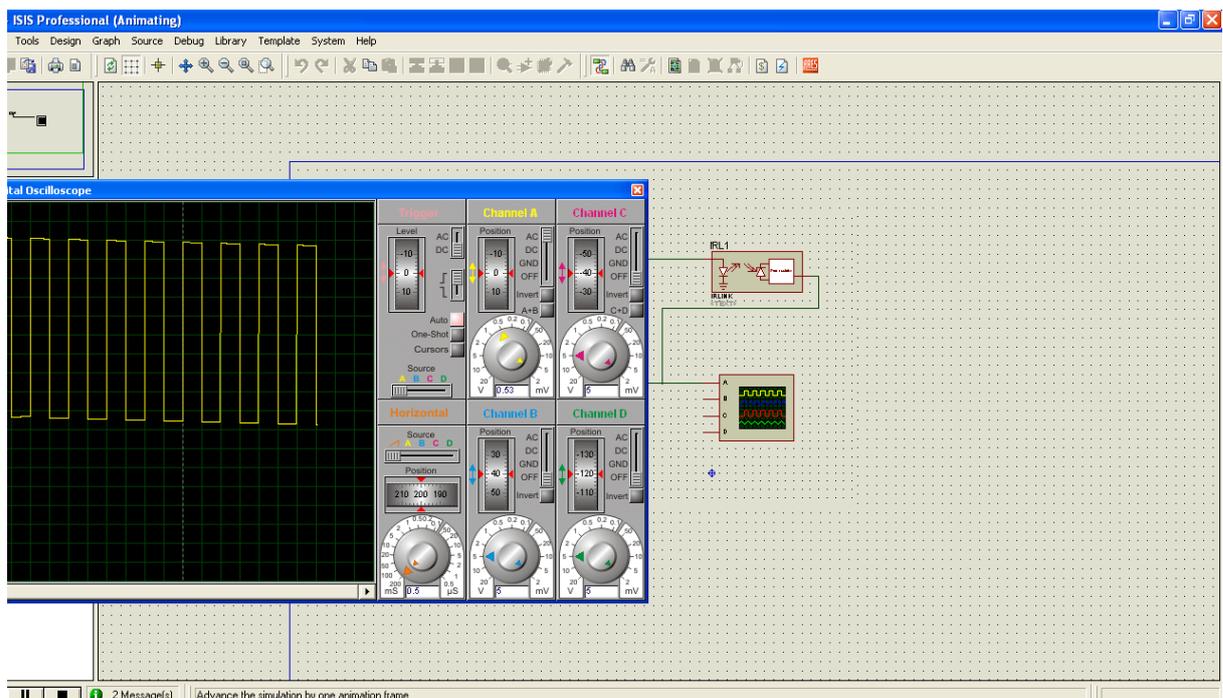


Figure I.4 Montage du IR sous ISIS PORTUES

### I.6 Arduino

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. Sans tout connaître, ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources

disponibles en ligne pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités de relations humain/machine ou environnement machine.

Arduino est un projet en source ouverte : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions.

La carte Arduino repose sur un circuit intégré (un mini ordinateur appelé également microcontrôleur) associé à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes. Il existe un grand nombre de variantes.

### **I.6.1 Historique**

Le projet Arduino est né en hiver 2005. Massimo Banzi enseigne dans une école de Design à Ivrea en Italie, et souvent ses étudiants se plaignent de ne pas avoir accès à des solutions bas prix pour accomplir leurs projets de robotique. Banzi en discute avec David Cuartielles, un ingénieur Espagnol spécialisé sur les microcontrôleurs... Ils décident de créer leur propre carte en embarquant dans leur histoire un des étudiants de Banzi, David Mellis qui sera chargé de créer le langage de programmation allant avec la carte. En deux jours, David écrira le code! Trois jours de plus et la carte était créée... Ils décidèrent de l'appeler Arduino (un bar fréquenté par les élèves à proximité de l'école) : cela devient un hit, tout de suite auprès des étudiants. [8]

Tout le monde arrive à en faire quelque chose très rapidement sans même avoir de connaissances particulières ni en électronique ni en informatique : réponse à des capteurs, faire clignoter des LEDs, contrôler des moteurs... Ils publient les schémas, investissent 3000 euros pour créer le premier lots de cartes : 200.

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

### **I.6.2 Exemples d'usages**

Plate-forme logicielle et matérielle de création d'objets numériques, Arduino permet de programmer des circuits électroniques qui interagissent avec le milieu qui les entoure. Connectés notamment à des capteurs sonores, thermiques, de mouvements, ces circuits électroniques peu coûteux, dénommés microcontrôleurs, peuvent en retour générer des images, actionner un bras articulé, envoyer des messages sur Internet, etc. Des dizaines de milliers d'artistes, de designers, d'ingénieurs, de chercheurs, d'enseignants et même d'entreprises l'utilisent pour réaliser des projets incroyables dans de multiples domaines :

- Prototypage rapide de projets innovants utilisant l'électronique, Arduino facilitant l'expérimentation en amont de la phase d'industrialisation ;
- Production artisanale d'objets numériques et de machines-outils à faible coût dans la perspective d'une culture d'appropriation technologique favorisant le bricolage et la débrouille ;
- Captation et analyse de données scientifiques (environnement, énergie, etc.) à des fins éducatives, de recherche ou d'appropriation citoyenne ;
- Spectacle vivant grâce aux nombreuses fonctions d'interaction offertes par Arduino, il est possible de créer des performances de VJing, utiliser le mouvement des danseurs pour générer en temps réel des effets sonores et visuels dans un spectacle ;
- Installations d'arts numériques, Arduino permettant de réaliser des oeuvres d'art interagissant de manière autonome avec le public ;
- Mode et design textile, plusieurs stylistes et designers investissant ce domaine créatif en exploitant les possibilités offertes par l'intégration de l'électronique notamment dans des vêtements (textile) ;
- Projets pédagogiques à destination d'étudiants, de professionnels ou du grand public selon les porteurs de ces initiatives : écoles supérieures, centres de formation spécialisée ou des Media Labs. [9]

### I.6.3 Pourquoi Arduino Mega

Il existe plusieurs modèles fabriqués par : ATMEL. Le choix dépend de plusieurs critères de sélection dont le développeur doit tenir compte du :

- Type du microcontrôleur ;
- Nombre d'entrées/sorties ;
- Liaison d'entrées/sorties ;
- Conversion analogique numérique et numérique analogique ;
- Mémoire RAM, ROM, EPROM interne ou externe, sa taille ;
- Vitesse d'horloge, temps d'exécution d'une multiplication, d'une division ;
- Bus de données 8bits /16bits ;
- Les logiciels de programmation (assembleur, c, micro,...) ;
- Les émulateurs pour la mise au point des applications ;
- Les évolutions prévisibles du composant, son prix, les sources.

Une carte électronique est un support plan, flexible ou rigide, généralement composé d'époxy ou de fibres de verre. Elle possède des pistes électriques disposées sur une, deux ou plusieurs couches (en surface et/ou en interne) qui permettent la mise en relation électrique des composants électroniques. Chaque piste relie tel composant à tel autre, de façon à créer un système électronique qui fonctionne et qui réalise les opérations demandées.

- Un prix dérisoire étant donné l'étendue des applications possibles. On comptera 20 euros pour la carte ;
- Une compatibilité sous toutes les plateformes, à savoir : Windows, Linux et Mac OS. Une communauté ultra développée ;
- Des milliers de forums d'entraide, de présentations de projets, de propositions de programmes et de bibliothèques, ... ;
- Un site en anglais [arduino.cc](http://arduino.cc) et un autre en français [arduino.cc](http://arduino.cc) où vous trouverez tout de la référence Arduino, le matériel, des exemples d'utilisations, de l'aide pour débiter, des explications sur le logiciel et le matériel, etc ;
- Une liberté quasi absolue. Elle constitue en elle-même deux choses :
  1. Le logiciel : gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir.
  2. Le matériel : cartes électroniques dont les schémas sont en libre circulation sur internet.

#### I.6.4 Les caractéristiques de la carte Arduino Mega 2560

Microcontrôleur	ATmega 2560
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12 V
Tension d'entrée (limites)	6-20 V
Broches E/S numériques	54 (dont 14 alimentent la sortie PWM)
Broches d'entrées analogiques	16
Intensité maxi disponible par broche E/S (5V)	40 mA
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul

Mémoire programme Flash	256 KB dont 8 KB sont utilisés par le bootloader
SRAM	8 KB
EEPROM	4 KB
Vitesse de l'horloge	16 MHz

Tableau I.1. Caractéristique de la carte Arduino Mega [10]

### I.6.5 Le Microcontrôleur ATmega2560

Le Atmel ATMEGA2560-16AU est un microcontrôleur 8 bits CMOS basse puissance basée sur architecture RISC améliorée des AVR. En exécutant des instructions puissantes en un seul cycle d'horloge, le ATMEGA2560-16AU atteint des débits approchant les 1MIPS par MHz permettant aux concepteurs de système d'optimiser la consommation d'énergie par rapport à la vitesse de traitement. [11]

Le microcontrôleur de cette carte est un objet capable de traiter, de stocker et de restituer de l'information. Il est en particulier constitué d'un microprocesseur. C'est la partie centrale qui permet le traitement de l'information : il comprend des milliers voir des millions de transistors. Ils réalisent les fonctions logiques suivantes :

- Calculs arithmétiques et logiques ;
- Mémorisation ;
- Interface de communication.

Les Atmel sont des composants dits RISC (**R**educed **I**nstructions **S**et **C**omputer), ou encore (composants à jeux d'instructions réduit). Ces microcontrôleurs utilisent l'architecture de Harvard : c'est-à-dire le programme et les données sont stockées dans des mémoires physiquement séparées. (Deux bus de données. Un bus est utilisé pour les données et un autre pour les instructions).

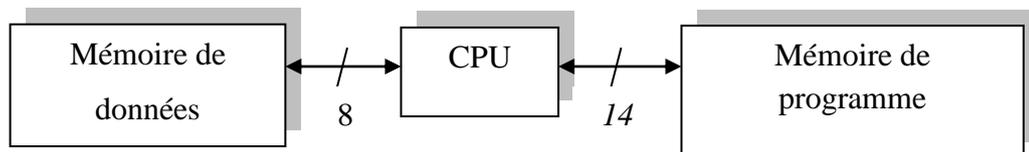


Figure I.5 Architecture de Harvard pour les Atmel

La figure (I.6) montre un microcontrôleur ATmega2560, qu'on trouve sur la carte Arduino.



*Figure I.6 Microcontrôleur Atmega2560*

- Architecture RISC avancée ;
- 135 Instructions puissantes, la plupart, avec une exécution en un seul cycle d'horloge ;
- Registres  $32 \times 8$  à usage général ;
- Fonctionnement entièrement statique ;
- Jusqu'à 16 MIPS à 16 MHz ;
- Multiplicateur en 2 cycles sur la puce ;
- Segments de mémoire non-volatile à haute endurance ;
- 256 Ko de Flash dans le système auto programmable ;
- EEPROM 4Ko ;
- SRAM 8Ko Interne ;
- Flash EEPROM 10 000 cycles de Lecture/Ecriture ;
- Conservation des données : 20 ans à 85°C / 100 ans à 25°C ;
- Section de code de démarrage en option avec bits de verrouillage indépendants ;
- Programmation IN SITU par le programme de démarrage sur puce ;
- Opération de lecture écriture réelle ;
- Verrou de programmation pour la sécurité du logiciel ;
- Endurance : Jusqu'à 64 Koctets d'espace de mémoire externe en option ;
- Supporte la librairie Atmel® QTouch® ;
- Boutons tactiles capacitifs, curseurs et molettes ;
- Acquisition QTouch et QMatrix®. [11]

## I.7 Description générale du Arduino Mega

La photo présente la carte Arduino Mega utilisée.

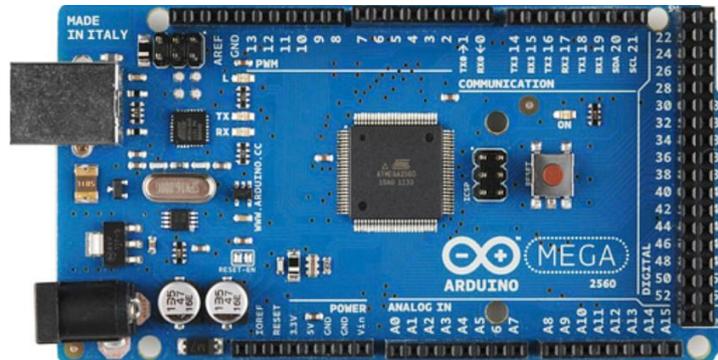


Figure I.7 La carte Arduino Mega2560

### I.7.1 Le microcontrôleur

Le microcontrôleur ATmega2560 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en effet 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 256 KB (dont bootloader de 8 KB) ;
- **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
- **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.
- Le **registre** : c'est un type de mémoire utilisé par le processeur ;
- La mémoire **cache** : c'est une mémoire qui fait la liaison entre les registres et la RAM.

[9]

### I.7.2 Les sources d'alimentation de la carte

On peut distinguer trois 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 régu-

lée). 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.

- **USB.** Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par ce le port
- **Fiche JACK :** Alimentation externe qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte. [9]

### I.7.3 Les ports de la carte

- **GND :** La carte arduino Mega2560 se compose de cinq ports de **GND** ;
- **POWER (Puissance) :** la carte Arduino Mega2560 se compose de quatre portes de **POWER**, et sont comme suit :
  - **IOREF.** Cette broche sur la carte fournit la référence de tension avec laquelle le microcontrôleur fonctionne. Un écran correctement configuré peut lire la tension de la broche **IOREF** et sélectionner la source d'alimentation appropriée ou activer des traducteurs de tension sur les sorties pour travailler avec le **5V** ou **3,3V** ;
  - **3V3.** Une alimentation de **3,3 volts** générés par le régulateur à bord, la consommation de courant maximal est de **50 mA** ;
  - **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 ;
  - **VIN.** La définition est déjà citée. [10]
- **RESET (Réinitialiser) :** Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte. [10]

- **Broches E/S numériques** : Chacune des 54 broches numériques (numérotée de 0 à 53) sur le Mega2560 peut être utilisée comme une entrée ou une sortie, en utilisant `pinMode()`, `digitalWrite()`, et `digitalRead()` fonctions. Ils fonctionnent à **5 volts**. Chaque broche peut fournir ou recevoir 20 mA en état de fonctionnement recommandée et a une résistance pull-up interne (déconnectée par défaut) de **20-50 k ohm**. Un maximum de **40mA** est la valeur qui ne doit pas être dépassée pour éviter des dommages permanents au microcontrôleur. [10]

En outre, certaines broches ont des fonctions spécialisées.

#### I.7.4 Ports de communication

- **Communication série** : Port Série Serial : 0 (RX) et 1 (TX) ; Port Série Serial 1: 19 (RX) et 18 (TX) ; Port Série Serial 2: 17 (RX) et 16 (TX) ; Port Série Serial 3: 15 (RX) et 14 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Les broches 0 (RX) et 1 (TX) sont connectées aux broches correspondantes du circuit intégré ATmega8U2 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. [9]

Ces broches peuvent être connectées avec des accessoires d'exploitation d'un réseau sans fil et ne sont pas visibles comme : wifi, GPS, Bluetooth...

- **Serial Périphérie Interface (SPI)** : est un protocole de données série synchrone utilisées par les microcontrôleurs pour communiquer avec un ou plusieurs périphériques rapidement sur de courtes distances. Il peut également être utilisé pour la communication entre les deux microcontrôleurs.

Avec une connexion **SPI**, il y a toujours un dispositif maître (généralement un microcontrôleur) qui contrôle les périphériques. En règle générale, il y a trois lignes communes à tous les appareils :

- **MISO** (port 50) - La ligne esclave pour envoyer des données au maître.
- **MOSI** (port 51) - La ligne principale pour l'envoi de données vers les périphériques.

Et une ligne spécifique pour chaque appareil :

- **SS** (port 53) - la broche sur chaque périphérique que le maître peut utiliser pour activer et désactiver des dispositifs spécifiques.
- **SCK** (port 52) - Les impulsions d'horloge qui synchronise la transmission de données générées par le maître. [12]

- **I2C / TWI broches :**
  - (**SDA**). Est sur la broche numérique 20.
  - (**SCL**). Est sur la broche numérique 21. [13]

Ces broches peuvent être connectées avec des accessoires d'exploitation d'un réseau sans fil FM.

### I.7.5 Les fonctions IDE de l'environnement Arduino

- **Fonctions d'initialisation**
  - begin() : initialise communication avec Arduino "maître"
  - begin(adresse) : initialise communication avec Arduino "esclave".
- **Fonctions mode maître**
  - requestFrom(adresse, quantite) : demande de données à un esclave
  - beginTransmission(adresse) : débute communication avec un esclave (ouvre stockage données à envoyer avec write)
  - endTransmission() : envoie des données vers esclave
  - write() : écrit les données à envoyer vers esclave
  - available() : teste ces données disponibles en provenance esclave (cf requestFrom)
  - read() : lit les données en provenance de l'esclave.
- **Fonctions mode esclave :**
  - write() : envoie les données vers le maître après requête
  - available() : test ces données disponibles en provenance du maître (cf onReceive)
  - read() : lit données en provenance maître
  - onReceive(fonction) : définit la fonction à appeler sur réception de données en provenance du maître
  - onRequest(fonction) : définit la fonction à appeler sur requête du maître.
- **Fonctions obsolètes**
  - send() : obsolète
  - receive() : obsolète. [13]
- **PWM : 2 à 13 et 44 à 46.** Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite(). [10]
- **PCIN (PinChangeInterrupt)**

Le tableau suivant présente les broches PCINT utilisables sur l'Arduino Mega2560 :

<b>Broches PCINT</b>	<b>PINs Arduino</b>
PCINT0	19
PCINT1	20
PCINT2	21
PCINT3	22
PCINT4	10
PCINT5	11
PCINT6	12
PCINT7	13
PCINT8	6
PCINT9	15
PCINT10	14
PCINT16	A8
PCINT17	A9
PCINT18	A10
PCINT19	A11
PCINT20	A12
PCINT21	A13
PCINT22	A14
PCINT23	A15

*Tableau I.2. Les broches PCINT.*

**Broches d'entrée analogiques** : La carte Mega2560 dispose de 16 entrées analogiques (numérotées de 0 à 15), chacune pouvant fournir une mesure d'une résolution de 10 bits (c.-à-d. 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. [9]

**ADC.** Un signal analogique est celui qui peut prendre un certain nombre de valeurs, contrairement à un signal numérique qui n'a que deux valeurs : HAUT et BAS. Pour mesurer la valeur des signaux analogiques, l'Arduino a intégré analogique-numérique (ADC). L'ADC transforme la tension analogique en une valeur numérique. La fonction qu'on peut utiliser pour obtenir la valeur d'un signal analogique est `analogRead (broche)`. Cette fonction

convertit la valeur de la tension sur une broche d'entrée analogique et renvoie une valeur numérique de 0 à 1023, par rapport à la valeur de référence. La référence est 15V sur Arduino Mega2560. Il a un paramètre qui est le nombre de broches Il y a quelques autres broches analogiques de la carte : AREF. Tension de référence pour les entrées analogiques (si différent du 5V). Utilisé avec l'instruction `analogReference()`. [9]

### **I.7.6 Fonctionnement et utilisation**

L'utilisation des cartes Arduino est très simple : connecter la carte à l'ordinateur (via le câble USB, en général), lancer « Arduino IDE » (c'est le logiciel qui permet de programmer la carte), mettre en place les composants, les relier à la carte, coder le programme, le charger sur la carte grâce au bouton téléviser et le programme s'exécutera ensuite en boucle.

### **I.8 Conclusion**

Les cartes Arduino donnent vraiment un potentiel de création quasi infini pourvu que l'on dispose d'un matériel approprié. Il est possible de réaliser des dispositifs, de gérer des caméras, de commander des moteurs, d'alimenter automatiquement un plan au bout d'un laps de temps. L'autre intérêt est de faire de cet investissement un grand avantage dans l'application de contrôle et de commandes moderne en domotique.

Il est clair que l'Arduino permet d'assurer les principaux protocoles d'interfaçage : communication bidirectionnel, acquisition (conversion) et traitement des différents types de signaux avec une bonne précision et rapidité avec une mémoire de programme de données importantes.

# Chapitre II

## **Réalisation Du Dispositif Expérimental**

## II.1 Introduction

Dans ce chapitre, on présentera de manière sommaire une vue d'ensemble du dispositif expérimental réalisé «Une carte de contrôle et de commande domotique par Arduino Mega destinée à piloter huit accessoires à distance par infrarouge».

Ce système disposé d'un affichage tactile permet lui aussi de commander manuellement les huit accessoires et afficher l'action et les états de ces derniers.

Après avoir donné dans le chapitre précédent une description théorique sur les protocoles de communication et la logique programmable d'un système embarqué très simple tel que la carte Arduino, on va procéder à l'application expérimentale : pour cette raison plusieurs blocs ont été nécessaires pour la réalisation d'un tel dispositif.

## II.2 la domotique sur notre plan pratique

Grâce au WIFI, aux ondes radio, infrarouge, Bluetooth, à un réseau électrique ou à un système de câblage, les équipements électriques communiquent entre eux.

Parmi ces module de connexion nous avons choisi la liaison infrarouge qui est précédemment présentée au chapitre I et en à ajouter une autre commande supplémentaire tel que la commande par tactile afin d'éviter des défaillances de communication vis infrarouge.

Donc les informations de contrôle et de commande sont véhiculées sur notre réseau électrique de la maison (l'ensemble des huit équipements) par :

1. Liaison à distance par infrarouge (IR) ;
2. Liaison manuelle par tactile (TFT).

## II.3 Liaison à distance par infrarouge (IR)

La télécommande est un dispositif généralement de taille réduite servant à manipuler un autre à distance. Cette dernière sert à interagir avec des jouets, des appareils audiovisuels comme une télévision ou une chaîne Hi-fi, un moteur de porte de garage ou de portail, un éclairage, l'ouverture des portières d'une voiture, des appareils de topographie...

### II.3.1 Le choix de la télécommande infrarouge

Une télécommande infrarouge se reconnaît à deux indices :

1. On voit sur le bout dirigé vers l'appareil une ampoule minime qui dépasse la coque ou est placée derrière, une plaque plus ou moins transparente selon les modèles.

2. Il faut viser précisément l'appareil que l'on veut commander.

Si une télécommande a une coque complètement fermée et opaque, il n'est pas nécessaire de viser l'appareil, elle fonctionne sans doute plutôt avec des ondes radio.

Notre télécommande choisie est celle d'un lecteur module MP3 pour voiture étant un choix très logique dû à la disponibilité dont coup est réduit sur le marché ainsi qu'à la simplicité de sa construction.

La figure suivante nous montre notre choix.



*Figure II.1 la télécommande « Car MP3 »*

Cet émetteur IR compact permet d'activer ou désactiver à distance jusqu'à 21 équipements au moyen d'un récepteur adéquat (décrit dans ce même numéro). Les codes de cet émetteur sont transformés en impulsions infrarouges et permettent d'intervenir soit sur un seul canal soit pour une séquence prédéfinie en une seule commande.

Nous allons examiner le schéma électrique de la figure II.1 puis passer au fonctionnement de l'émetteur de cette télécommande infrarouge à 21 canaux.

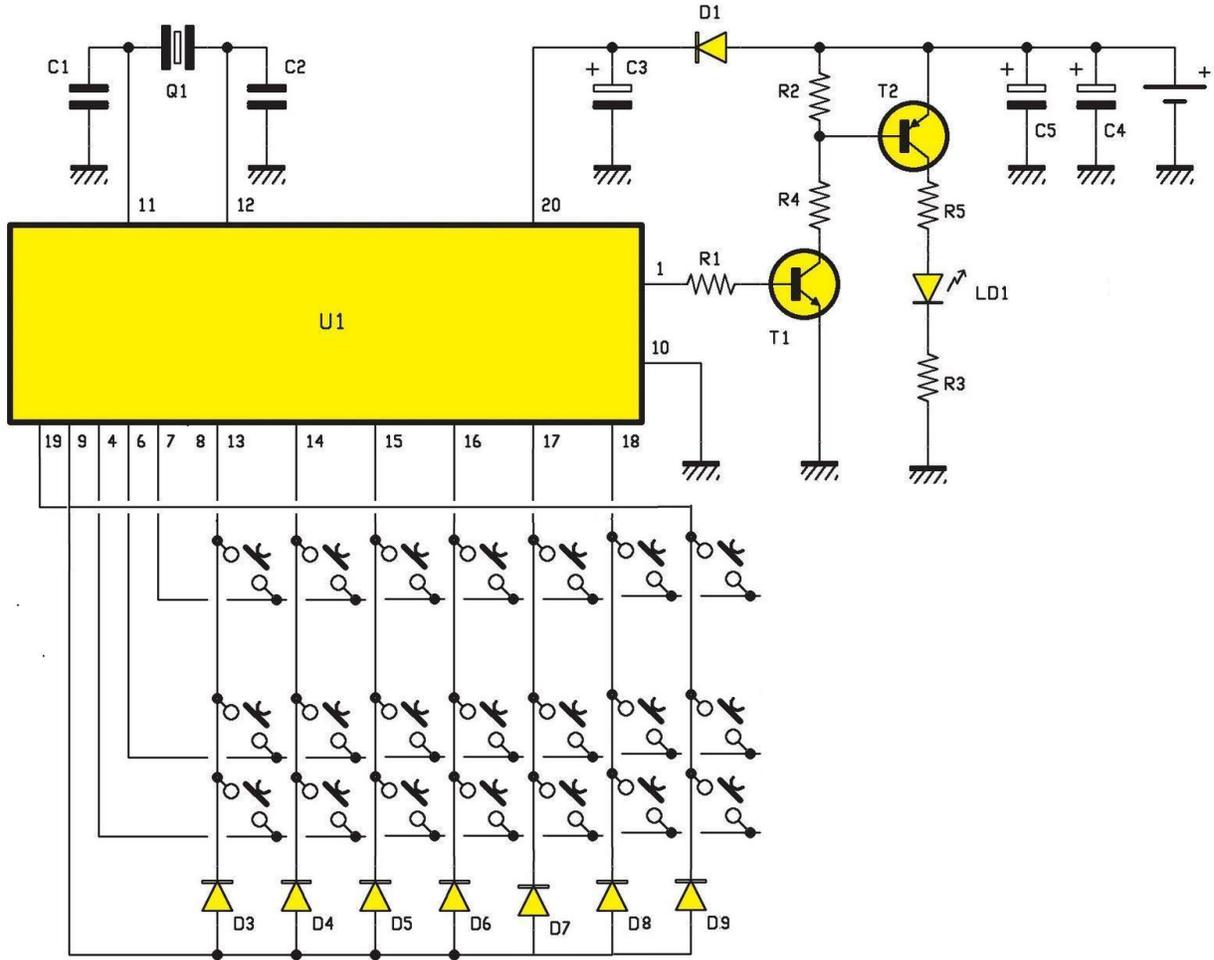


Figure II.2 schéma descriptif d'une télécommande infrarouge à 21 canaux

Le TX IR consiste essentiellement en un codeur, deux transistors et une LED infrarouge : le codeur est relié à un clavier à matrice dont les touches le contraignent à produire des codes que le RX interprétera comme les différentes commandes. La sortie du codeur pilote, au moyen de l'étage amplificateur formé des transistors, la LED infrarouge qui émet dans l'espace des impulsions de lumière modulée par des codes. Voyons cela en détail sur la référence [14], le codeur SAA3004 de Philips [15] est effectivement capable de passer jusqu'à 448 commandes, mais notre émetteur n'en utilise que 21 (21 canaux) plus deux (pour les fonctions spéciales) : chaque commande équivaut à une séquence d'impulsions ou à un code. Pour chaque combinaison, un signal numérique est produit par le circuit intégré entre la broche 1 et la masse. L'interface utilisateur est un clavier à membrane relié aux trois lignes correspondant aux broches 2, 3, 4 et les sept colonnes aux broches 13, 14, 15, 16, 17, 18, 19. Cette structure permet théoriquement  $3 \times 7 = 21$  combinaisons mais grâce à une astuce de connexion, le constructeur

arrive à en passer 64 ! Comme nous nous contentons de 21, le schéma électrique est simplifié et nous n'utilisons qu'une partie du clavier : les poussoirs sont reliés à des lignes et colonnes du circuit intégré SAA3004.

Le fonctionnement du codeur est cadencé par l'horloge constituée par un résonateur céramique de 455 kHz. Les broches 11 et 12 sont reliées au résonateur (et à C1 et C2 qui le complètent). La liaison se fait par diode afin que la broche 9 ne soit activée que quand la ligne DRV6 l'est aussi.

La sortie du circuit intégré, broche 1, émet un flux de données chaque fois qu'un des poussoirs est pressé : les impulsions correspondantes pilotent la base du NPN T1 dont le collecteur les restitue inversées pour polariser la base du PNP T2. Ce dernier alimente la LED infrarouge pour envoyer des impulsions de lumière infrarouge.

LD1 « la LED infrarouge » en lumière visible, sert de moniteur d'émission : elle clignote chaque fois qu'une touche impliquant une émission est pressée (les 21 des canaux. L'émetteur dans son ensemble est alimenté par une pile de 3 V (consommation 2 mA en émission) reliée aux bornes E1. D1 protège le circuit intégré contre toute inversion accidentelle de polarité. Voyons maintenant comment utiliser l'émetteur comportant 21 commandes pour autant de canaux de sortie du récepteur.

### **II.3.2 Système de codage infrarouge**

La plupart des télécommandes universelles possèdent une liste des appareils pilotables classés par marques et par modèles. Il suffit alors de trouver chaque appareil pilotable dans la base de données. Si aucun ne correspond, il est presque toujours possible d'engager un mode de recherche consistant à parcourir tous les appareils de la base et d'envoyer l'ordre d'extinction. Si l'appareil à piloter ne s'éteint pas, la télécommande passe au suivant. S'il s'éteint, il faut vérifier les autres touches, et si cela correspond, l'appareil est trouvé. Ce mode de recherche peut fonctionner parce que beaucoup d'appareils de même type existent sous des marques et des modèles différents selon les pays.

### **II.3.3 La norme RC5**

Le code RC5 est une salve de 14 bits. Les bits sont codés par l'intermédiaire de fronts et non pas de niveaux logiques ou encore de durées. A un front montant correspond un "1" logique, à un front descendant correspond un "0" logique. Ce type de codage serait aussi connu son le

nom de code biphase. Le code ne contient pas de "Header". En effet d'autres codes comme Technique, Yamaha et beaucoup d'autres ont un header qui se compose en général d'un "1" logique puis d'un "0" de durée en général plus grande que ceux des bits. Sur le code RC5 c'est en fait le premier bit qui a ce rôle. Le but de ce header est de permettre au récepteur de s'adapter au niveau du signal reçu. Pour la plupart des récepteurs IR intégrés il y a un CAG (Contrôle Automatique du Gain) qui permet de recevoir des signaux aussi bien de près que de loin. Mais il faut un temps d'adaptation pour que l'asservissement agisse. Ainsi, on risque de perdre le premier bit ; ce qui n'est pas très intéressant ! Le bit de synchronisation est un "1" logique.

### II.3.3.1 Constitution d'une trame en code RC5

Elle se compose d'une suite de 14 bits et sa construction est la suivante :

- 2 bits de départ ;
- 1 bit de basculement ;
- 5 bits d'adressage du système ;
- 6 bits d'instruction

Les 2 bits de départ sont utiles pour ajuster le niveau de la commande automatique du gain AGC dans le circuit intégré de réception. Le bit de basculement indique une nouvelle transmission de données.

Sa valeur change à chaque nouvelle activation d'une touche afin de distinguer une nouvelle pression d'une pression continue sur la même touche. Les 5 bits suivants déterminent l'adresse du dispositif devant réagir à la commande.

Nous avons donc  $2^5 = 32$  groupes d'adressage. L'instruction destinée à l'appareil est codée dans les 6 derniers bits. Ce qui donne  $2^6 = 64$  instructions. [16]

### II.3.3.2 Détail d'un bit de la trame RC5

Les bits du code RC5 sont codés en biphase, c'est à dire qu'un bit est composé de 2 demi bits alternés. La combinaison bas/haut caractérise un bit positionné à 1 et la combinaison haut/bas un bit remis à 0. [16], comme le montre la figure suivante :

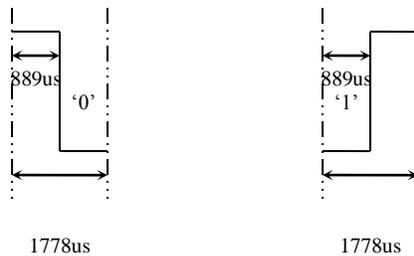


Figure II.3 le bit du code RC5

### II.3.3.3 La durée de la trame RC5

Chaque bit ayant une longueur de 1,778 ms, une trame rc5 dure alors  $14 \times 1,778 = 24,892$  ms (24,889 ms en réalité sur les documentations voir les explications un peu plus loin sur cette page). [16]

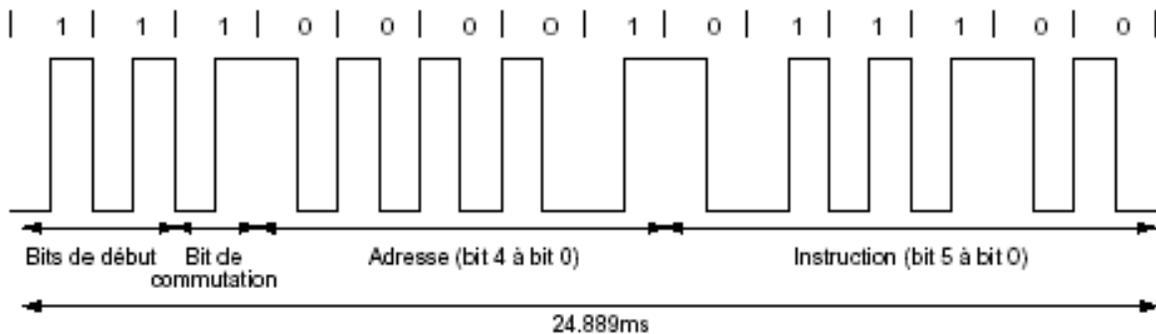


Figure II.4 Détail d'une trame RC5

La périodicité des messages (ou trame) a été choisie comme telle : c'est un multiple de la durée d'un bit soit,  $64 \times 1,778 = 113,792$  ms (113,78 ms exactement). [16]

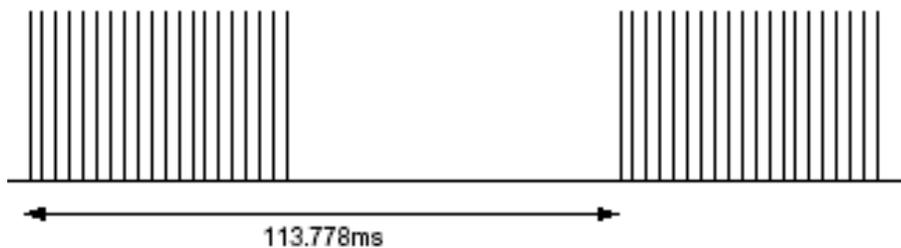


Figure II.5 Enchaînement des trames

En choisissant une fréquence de modulation infrarouge de 36 KHz, nous pouvons retrouver toutes les durées de bit, trame et période du code RC5. Les concepteurs et les développeurs travaillent avec des puissances de 2 pour simplifier la gestion des horloges. En partant du principe que l'on utilise que des diviseurs par 2, 4, 8, ... 64 ... 4096 ... nous disposerons de fréquences multiples ou sous-multiples de 36 KHz. En divisant 36 KHz par 64 nous obtenons une

fréquence de 562,5 Hz d'où une période de  $1 / 562,5 = 1,777777\dots$  soit 1,778 ms arrondi par excès : c'est exactement la durée d'un bit. En divisant  $14 / 562,5 = 24,888888\dots$  soit 24,889 ms arrondi par excès : cela correspond à la durée d'une trame de 14 bits. En divisant 36 KHz par 4096 nous obtenons une fréquence de 87,890625 Hz d'où une période de  $1 / 87,890625 = 113,777777\dots$  soit 113,78 ms arrondi par excès : c'est la durée de la périodicité des trames. [16]

### II.3.3.4 Codage des trames d'émetteur

Sur le plan pratique et selon le bouton sur lequel on appuie, l'émetteur de la télécommande envoie des messages différents au récepteur. Ces messages sont des séries de clignotements très rapides et très précis de l'émetteur infrarouge. Chaque bouton de la télécommande correspond à une série bien particulière de signaux.

### II.3.3.5 Logique câblée du IR

Pour tester l'utilisation et la connectivité et lier la trame émise de ce dernier, il faut utiliser un récepteur infrarouge 38 KHz, une résistance de 100 Ohms et une LED émettrice infrarouge. Le câblage sera le suivant à l'aide d'une Arduino :

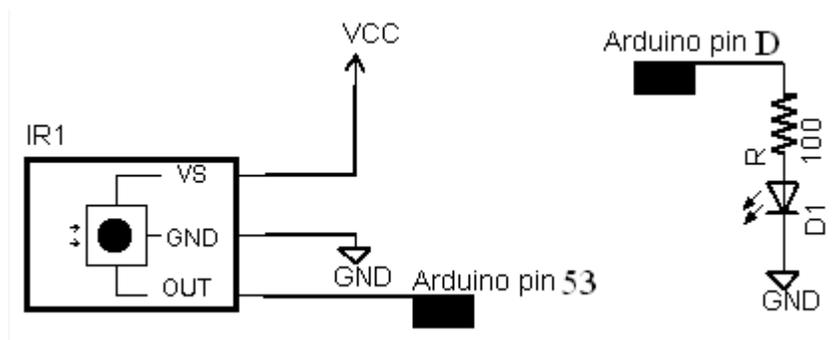


Figure II.6 Schéma de base récepteur IR

Arduino pin D : sortie digitale ;

Pour faciliter l'utilisation du récepteur IR nous avons réalisé un adaptateur qu'on doit coupler avec les PINs de l'Arduino comme le montre la figure suivante.

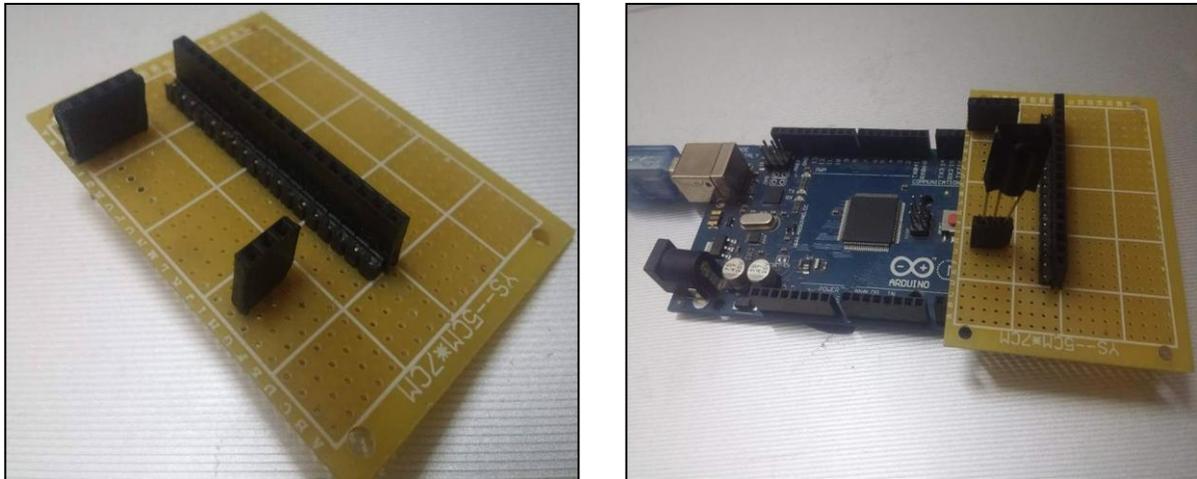


Figure II.7 Le support IR réalisé

On peut résumer la connectivité IR de notre télécommande via Arduino et afficher le codage déchiffré par un programme injecté comme programme esclave au noyau de l'Arduino sous forme d'un tableau. L'explication du programme esclave sera donnée au chapitre qui suit :

Touche	Code
CH-	16753245
CH	16736925
CH+	16769565
<<	16720605
>>	16712445
>	16761405
(-)	16761405
(+)	16754775
EQ	16748655
FOL-	16750695
FOL+	16756815
0	16738455
1	16724175
2	16718055
3	16743045
4	16716015
5	16726215
6	16734885
7	16728765
8	16730805
9	16732845

Tableau II.1 les codes des touches de la télécommande utilisée

Pour contrôler notre télécommande, l'Arduino devra alimenter une LED infrarouge de façon à ce qu'elle émette exactement le même signal. La figure suivante nous montre le montage réalisé afin de décoder les différentes touches de la télécommande.

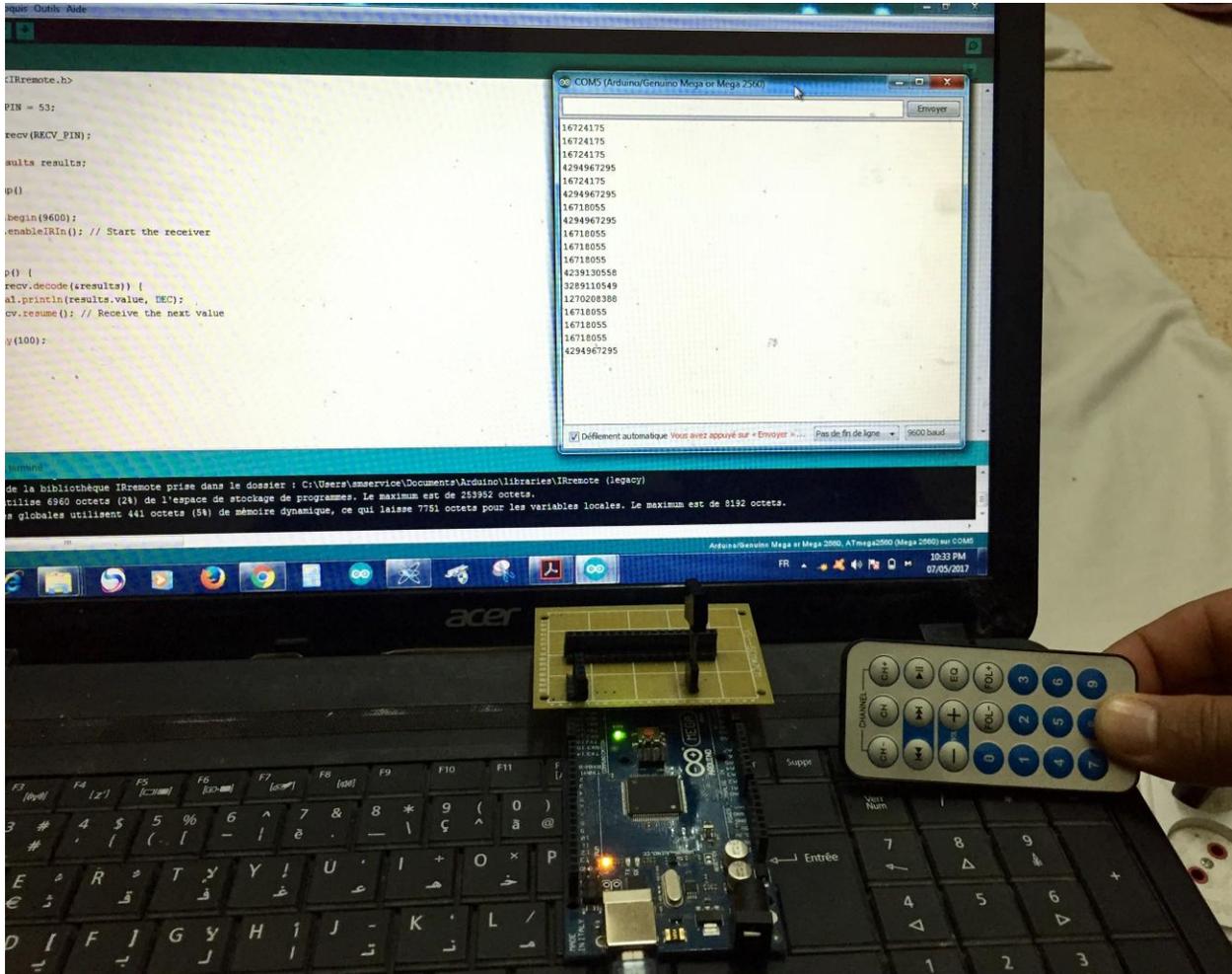


Figure II.8 extraction des codes RC5 par Arduino

Après exécution de ce code esclave (le programme numéro I), l'Arduino déchiffre à chaque réception d'une commande IR, et écrit le code reçu. La figure qui suit nous donne la réponse de ces actions sur les touches de la télécommande et le moniteur série de l'IDE au port 5 expose cette réponse.

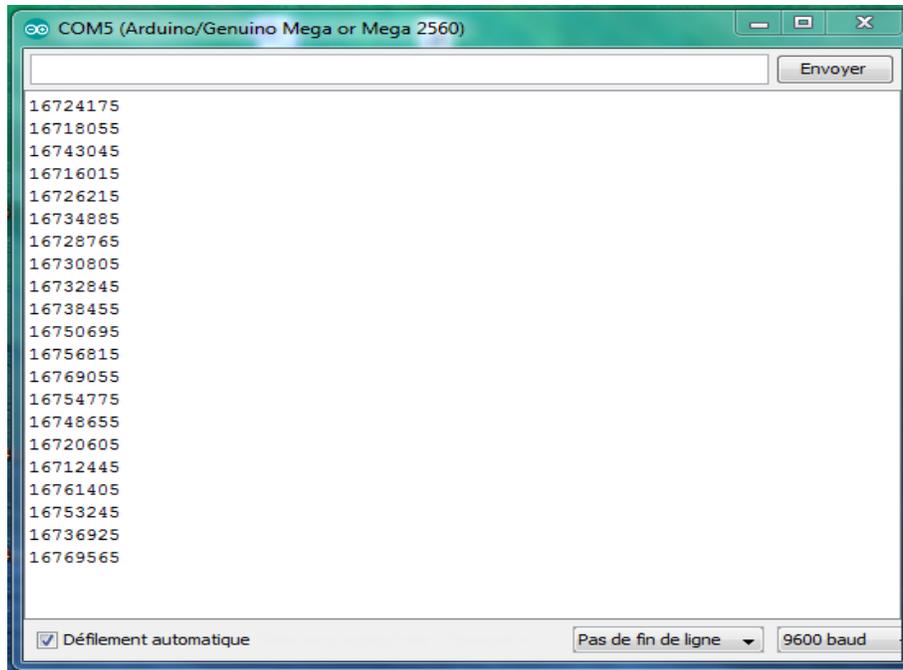


Figure II.9 Les codes RC5 sur le COM 5 de l'IDE

## II.4 Liaison manuelle par afficheur tactile (TFT).

Les afficheurs sont des écrans à cristaux liquides (liquid crystal display, LCD, en anglais) qui utilisent un mode d'affichage numérique sur un écran plat. On a tendance à confondre dans le sens commun LCD et TFT. Cependant les écrans dits TFT (acronyme de Thin Film Transistor) ne sont qu'un des types d'écrans à technologie LCD existants. Les écrans LCD-TFT sont aujourd'hui les plus courants.

### II.4.1 choix d'afficheur

Plusieurs afficheurs sont disponibles sur le marché et sont différents les uns des autres, non seulement par leurs dimensions, « lignes, caractères », mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage.

Les deux grands critères qui entrent en ligne de compte pour différencier les écrans LCD ou TFT de leurs prédécesseurs les écrans CRT sont : la dalle et l'éclairage.

- La dalle est le composant essentiel de la technologie LCD, cet élément rassemble toute l'électronique et les cristaux liquides.
- Le mode d'éclairage va permettre d'offrir des fonctionnalités de réglages avancées. Ainsi, nous aurons accès à des champs plus ou moins larges dans certains domaines de réglages. Il existe 3 grands modes de réglages.

Leur principal avantage du TFT est leur coût de production relativement bas. Leur inconvénient est une consommation excessive d'énergie, et donc une durée de vie réduite de la batterie pour les dalles de grande taille.

Notre choix est un écran couleur TFT 2.4 compatible avec Arduino Mega et est un écran tactile doté d'une carte SD. Le pilotage du TFT est basé sur un SSD1289 avec une interface de 8 bits pour les données et 4 bits pour les commandes. La figure si dessous présente une photo de ce dernier.



*Figure II.10 TFT 2.4*

Ecran TFT 2.4 tactile est un module (Touch Screen Shield) donc, il peut détecter la position de notre action sur sa surface externe.

#### **II.4.2 Spécifications**

Le tableau suivant résume les principales caractéristiques de notre afficheur ;

Screen taille	2.4 inch
Resolution	240/320
LCD Color	65 k
LCD Driver	ST7781
Interface	8080 8 data bit with 4 control bits
Touchscreen	4 Wire Resistive

*Tableau II.2 Caractéristiques de notre TFT [17]*

### II.4.3 Câblage TFT Arduino

Puisque notre TFT est Shield donc ces PINs sont par défaut adressés à des zones bien définies au Arduino. Le tableau qui suit illustre sa connexion.

PINs Arduino	TFT SHIELD PINs	Utilisation
3.3 V	3.3 V	Puissance
5 V	5 V	Puissance
GND	GND	Puissance
A0	LCD_RD	Contrôle LCD
A1	LCD_WR TOUCH_YP	Contrôle LCD / Données tactiles
A2	LCD_RS TOUCH_XM	Contrôle LCD / Données tactiles
A3	LCD_CS	Contrôle LCD
A4	LCD_RST	Réinitialisation LCD
D2	LCD_D2	Données LCD
D3	LCD_D3	Données LCD
D4	LCD_D4	Données LCD
D5	LCD_D5	Données LCD
D6	LCD_D6/TOUCH XP	Données LCD / données tactiles
D7	LCD_D7/TOUCH YM	Données LCD / données tactiles
D8	LCD_D0	Données LCD
D9	LCD_D1	Données LCD
D10	SD_CS	SD Select
D11	SD_DI	Données SD
D12	SD_DO	Données SD
D13	SD_SCK	Horloge SD

*Tableaux II.3 connexions TFT et Arduino [17]*

On constate que le Shield ne laisse que 3 broches libres : A5, D0 et D1. Comme D0 et D1 ne sont pas utilisées, on peut faire communiquer l'Arduino avec un autre système via RS232 d'un micro-ordinateur.

Sur le plan pratique, nous avons créé un menu qui présente la commande par tactile, ce menu contient huit boutons. Chacun, correspond à un des huit équipements.

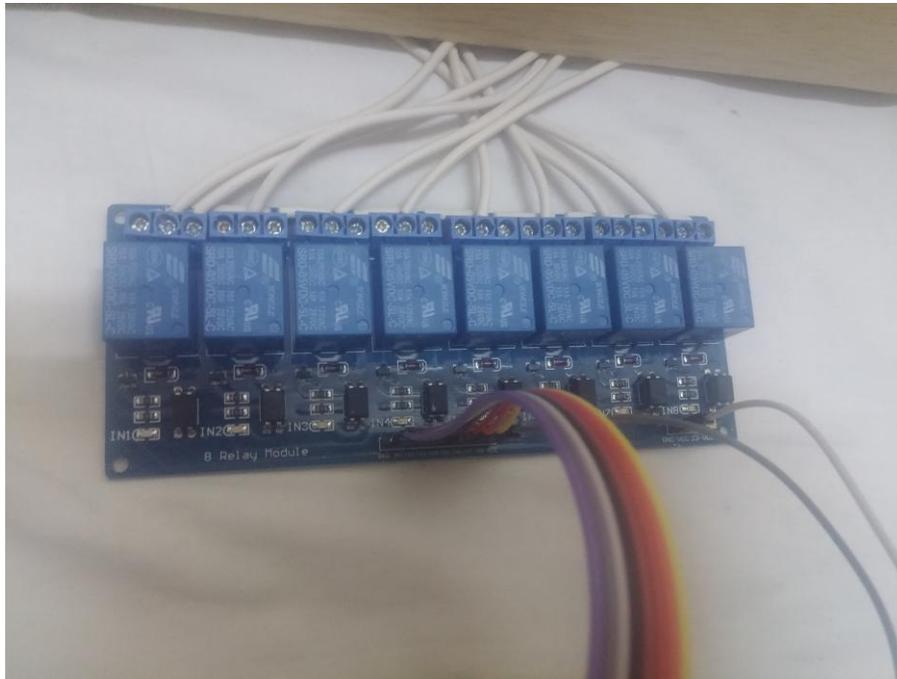
Les états des ces équipements sont connus si la couleur des boutons change. Les figures suivantes nous montrent le menu programmé sous IDE de l'Arduino et un exemple de test des boutons ainsi que leur état lors d'une action manuelle sur le tactile.



*Figure II.11 Les différentes réponses sur TFT*

## II.5 La connexion puissance commande

La majorité des équipements domotiques exige une source d'alimentation élevée dont la présence d'un étage de puissance est obligatoire pour cela, nous avons utilisé un driver qui joue la liaison entre notre carte de commande et les équipements. La photo qui suit montre le driver utilisé.



*Figure II.12 Le driver à huit Relais*

### **II.5.1 la démarche entre commande et puissance**

Un relais contient une bobine électrique qui génère un champ magnétique. Quand le champ magnétique est en place, cela déplace une pièce métallique à l'intérieur du relais, de sorte que celle-ci ouvre ou ferme un circuit électrique.

En pratique, cela implique que l'électronique de commande n'est jamais en contact réellement avec le courant de commande et quand on utilise un relais nu, il faut impérativement utiliser une diode anti-retour, sinon, quand la bobine n'est plus alimentée, le champ magnétique qui s'effondre induit un courant de retour qui pourrait bien détruire notre carte de commande.

Heureusement pour nous, ici, nous avons un circuit complet et autosuffisant « notre driver à huit relais ». La diode anti-retour est déjà présente, et nous avons même une petite LED indiquant que le courant passe.

### **II.5.2 Câblage équipement driver**

Nous pouvons alors distinguer 4 broches :

- GND, qui sera connectée à la masse du circuit de commande (la masse du Arduino) ;
- VCC, que l'on connectera au +5V du Arduino ;
- NC, qu'on ne connecte à rien ;

- SIG, que l'on connectera à l'une des sorties numériques de l'Arduino.

La dernière broche, SIG, sera la broche qui commandera l'allumage ou l'extinction du circuit commandé. En pratique, quand SIG est à l'état logique haut (5V), le courant pourra traverser le relais. Quand SIG est à l'état logique bas (0V), le relais ne laissera pas passer le courant.

En pratique, le connecteur bleu sur le côté gauche sera connecté sur l'un des fils de la phase de votre dispositif secteur. Cela donnera le montage suivant.

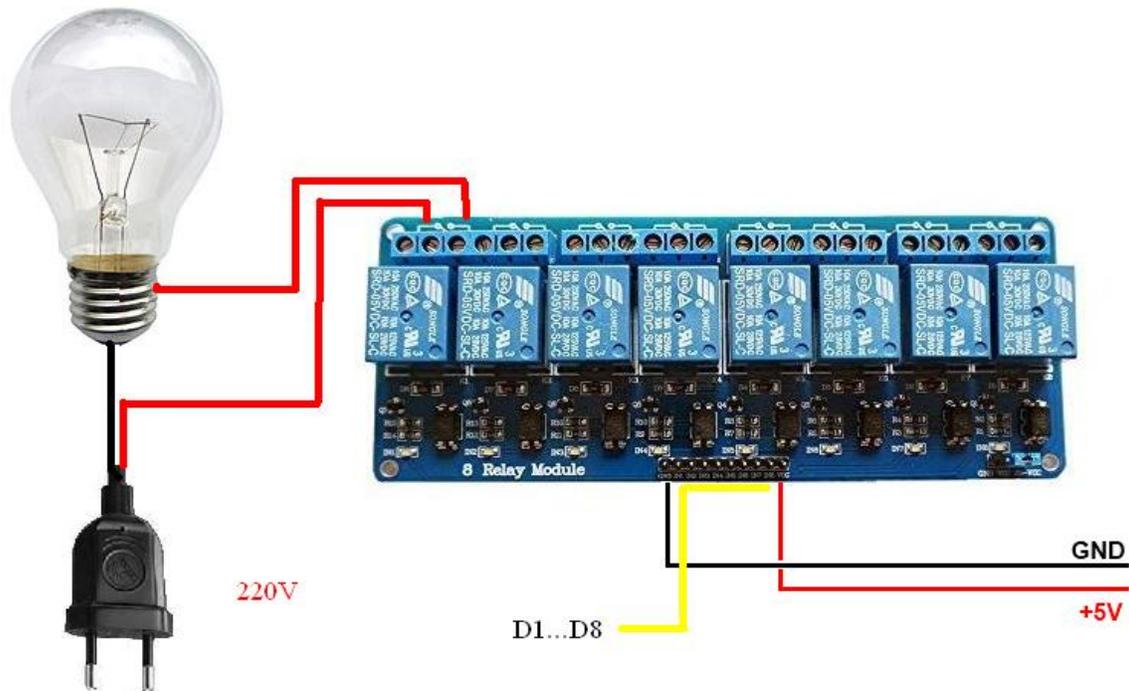


Figure II.13 La liaison équipement driver

Les fils rouge foncés sont la phase 220V, le noir est connecté à la masse de l'Arduino, le rouge au +5V, et le jaune à n'importe quelle broche numérique de l'Arduino (dans notre cas on a choisi les broches « 17, 18, 19, 20,...25 »).

Les différents équipements domotiques sont présentés sous forme de photos.

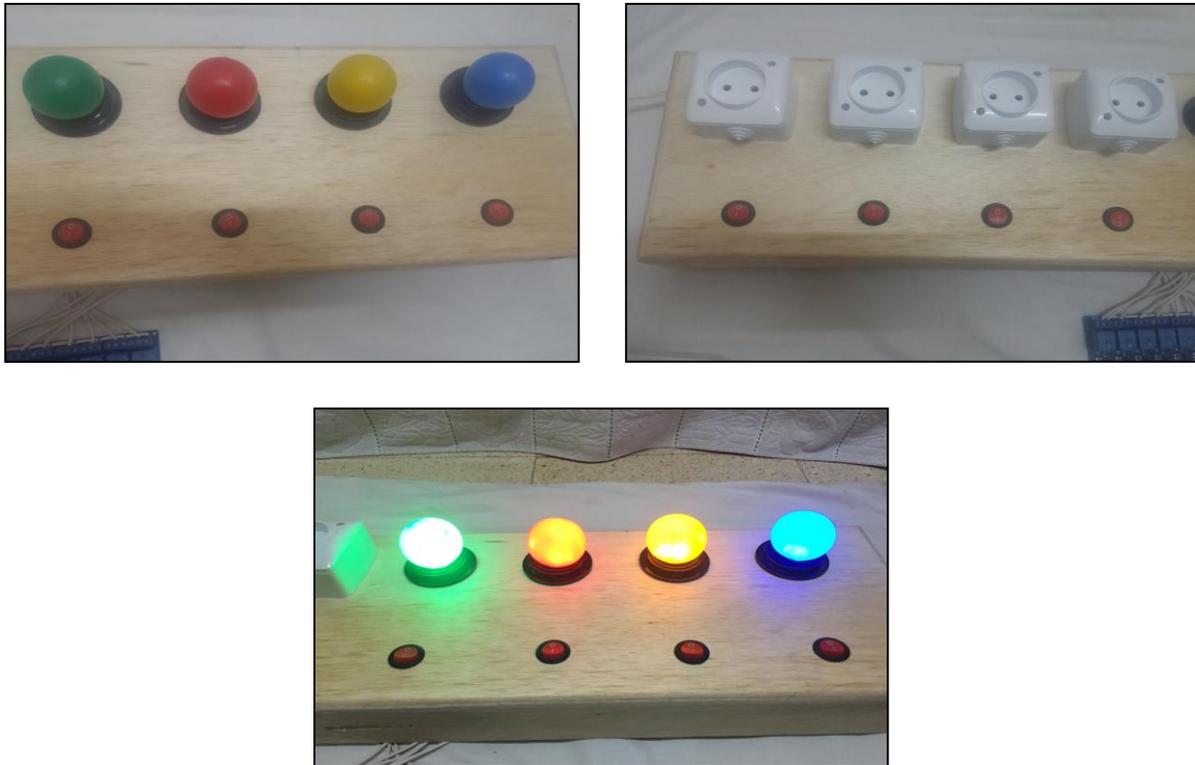


Figure II.14 Les huit équipements domotiques

NB : nos équipements domotiques sont quatre lampes plus quatre prises de 220V.

## II.6 La réalisation virtuelle

Avant de passer à la réalisation pratique et sa présentation, nous avons utilisé un CAO très important : il s'agit de ISIS-PORTEUS, c'est un CAO électronique perfectionné conçu par Labcenter Electronics qui permet de dessiner des schémas électroniques, de les simuler et de réaliser le circuit imprimé correspondant. Le CAO électronique « PROTEUS » se compose de nombreux outils regroupés en modules au sein d'une interface unique [18].

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

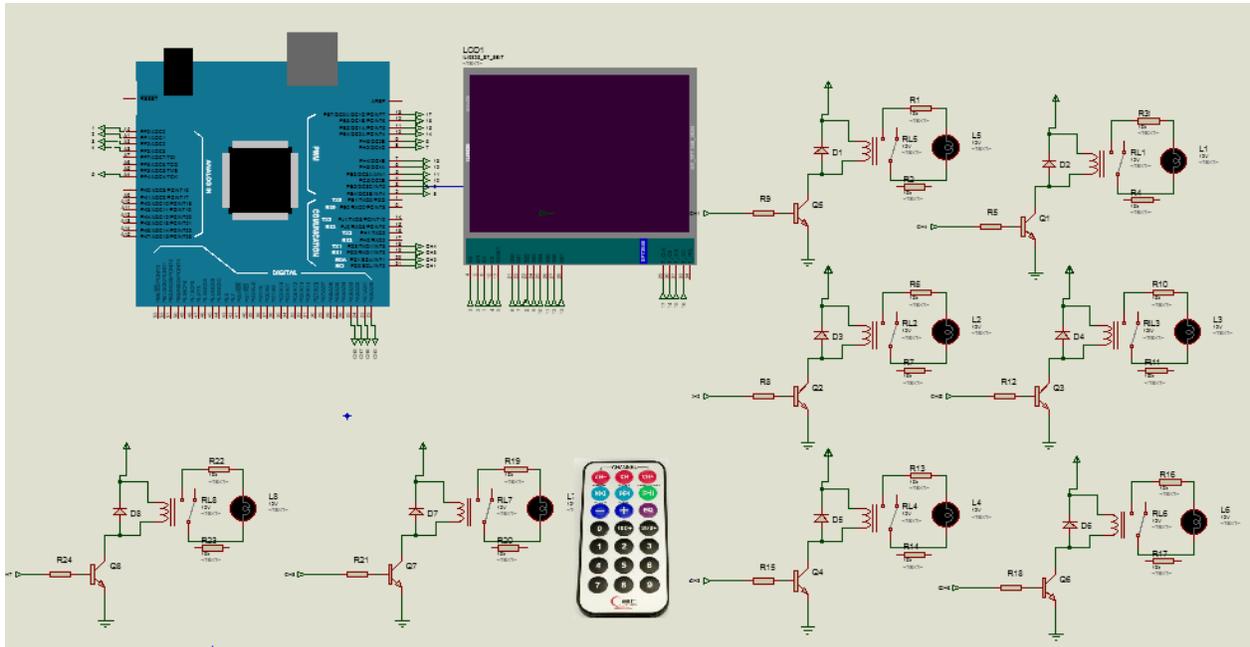


Figure II.15 La carte réalisée sous ISIS-ORTEUS

Ce CAO à la possibilité d'emporter le code hexadécimal généré par l'IDE vers l'Arduino, comme dans notre réalisation.

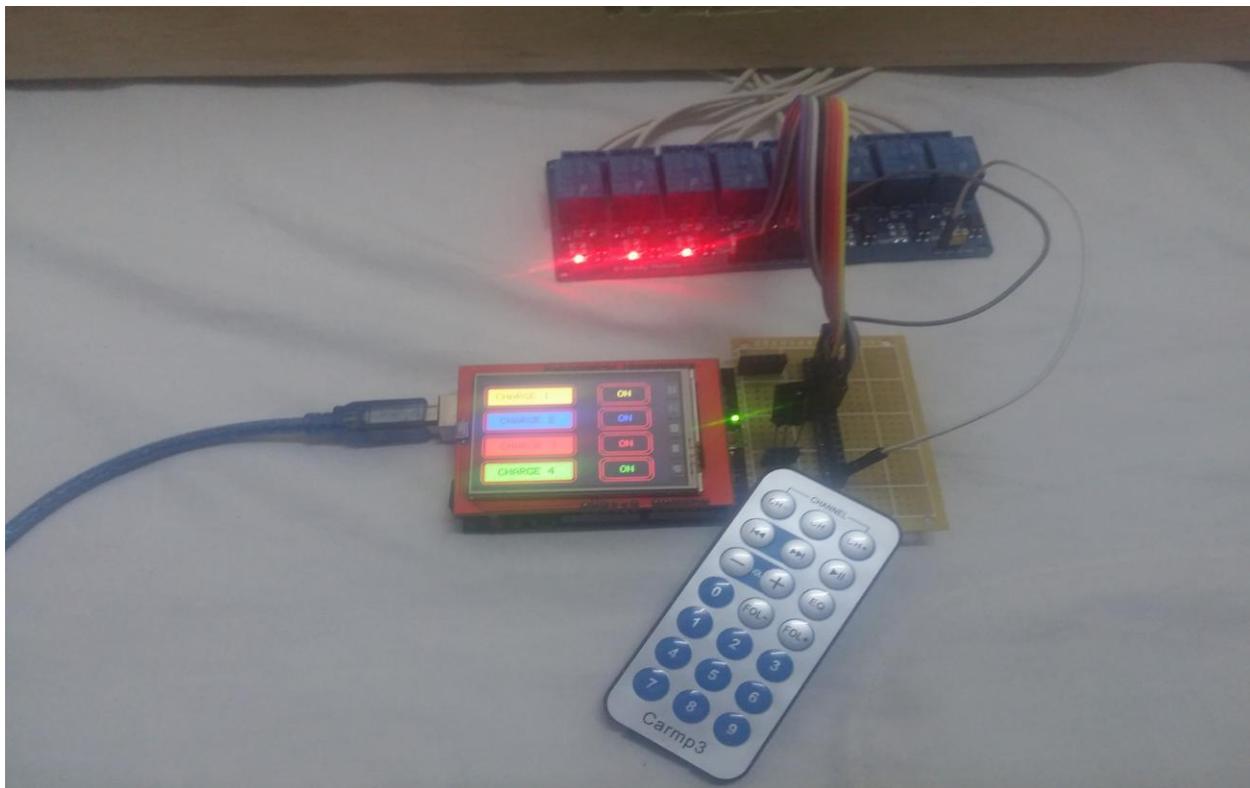


Figure II.16 Le dispositif domotique réalisé

## II. 7 Conclusion

Cette partie a donné lieu à la réalisation pratique d'un dispositif domotique dans le but de commander et contrôler huit équipements domotiques par infrarouge et tactile via un système embarqué tel que l'Arduino. L'ensemble des travaux décrits dans ce chapitre est :

- Le choix des composants convenables pour chaque bloc (commande, affichage et puissance) ;
- Exploitation de la construction interne des différentes cartes modulaires comme avantage.

Nous avons déduit un certain nombre de règles à suivre pour améliorer la conception de la carte. L'utilisation de la carte Arduino Mega, facilite le traitement des interruptions reçues par le IR et le tactile par les codes traduisant ces actions.

L'organigramme, le déroulement des différentes étapes du programme et les différents tests seront présentés dans le chapitre suivant.

# Chapitre III

**Description et analyse du  
programme**

### III.1 Introduction

Après avoir décrit la partie matérielle de notre dispositif domotique ainsi que la partie électronique et la conception de la carte de commande et contrôle du système, nous sommes arrivés à la partie la plus importante dans notre mémoire qui est la programmation de la carte Arduino. Dans ce chapitre, nous allons aborder cette programmation en passant par les étapes suivantes :

- Description de l'algorithme ;
- Choix du langage (Assembleur) qui est IDE (MPASM) ;
- Choix du logiciel (environnement C) ;

Le bon fonctionnement de notre système se base essentiellement sur une bonne démarche et une bonne réflexion de notre programme.

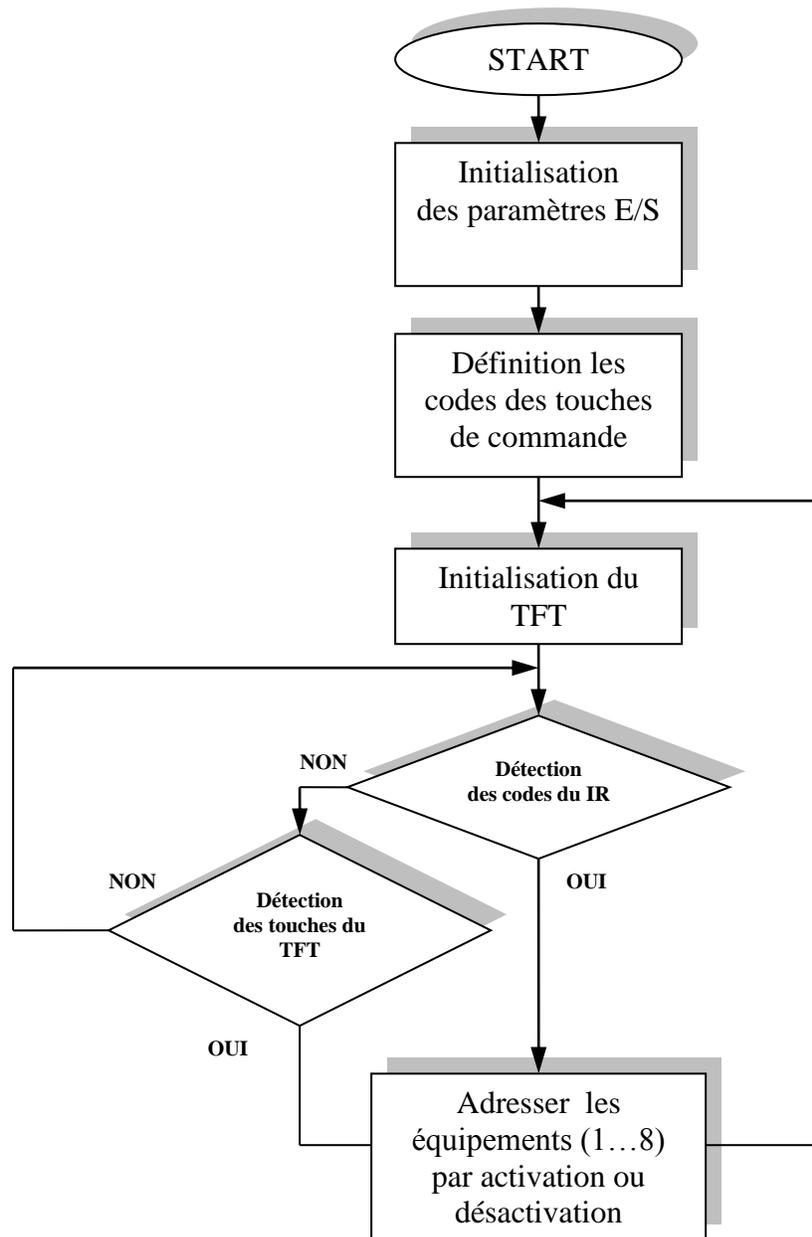
### III.2 Présentation de l'organigramme

Un organigramme de programmation « parfois appelé ALGORIGRAMME, LOGIGRAMME ou plus rarement ORDINOGRAMME » est une représentation graphique normalisée de l'enchaînement des opérations et des décisions effectuées par un programme d'ordinateur « dans notre cas c'est un programme assembleur ».

Avant de passer à la programmation, nous devons réaliser un organigramme qui explique le déroulement des différentes séquences tant intérieures qu'extérieures : il comportera plusieurs boucles dont la fin d'exécution succède toujours à son commencement.

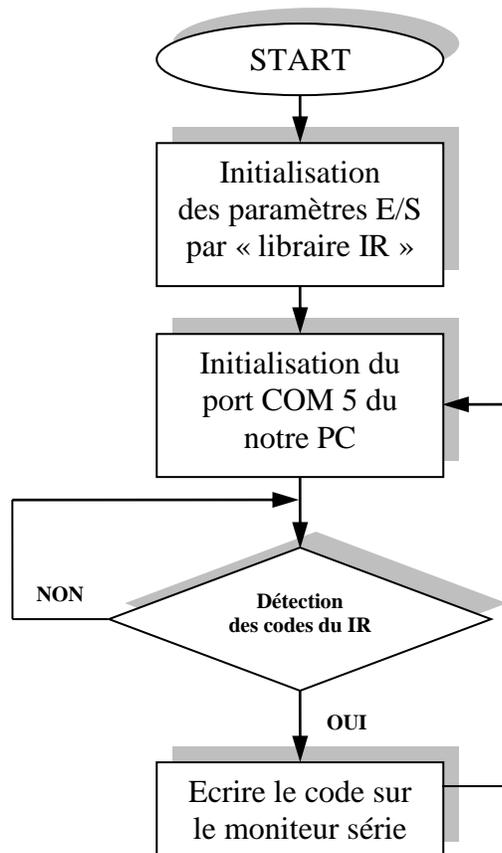
Chaque action est symbolisée par un rectangle et chaque choix est symbolisé par un losange.

*Le programme maître ;*



*Figure III.1 Organigramme principal*

*Le programme esclave ;*



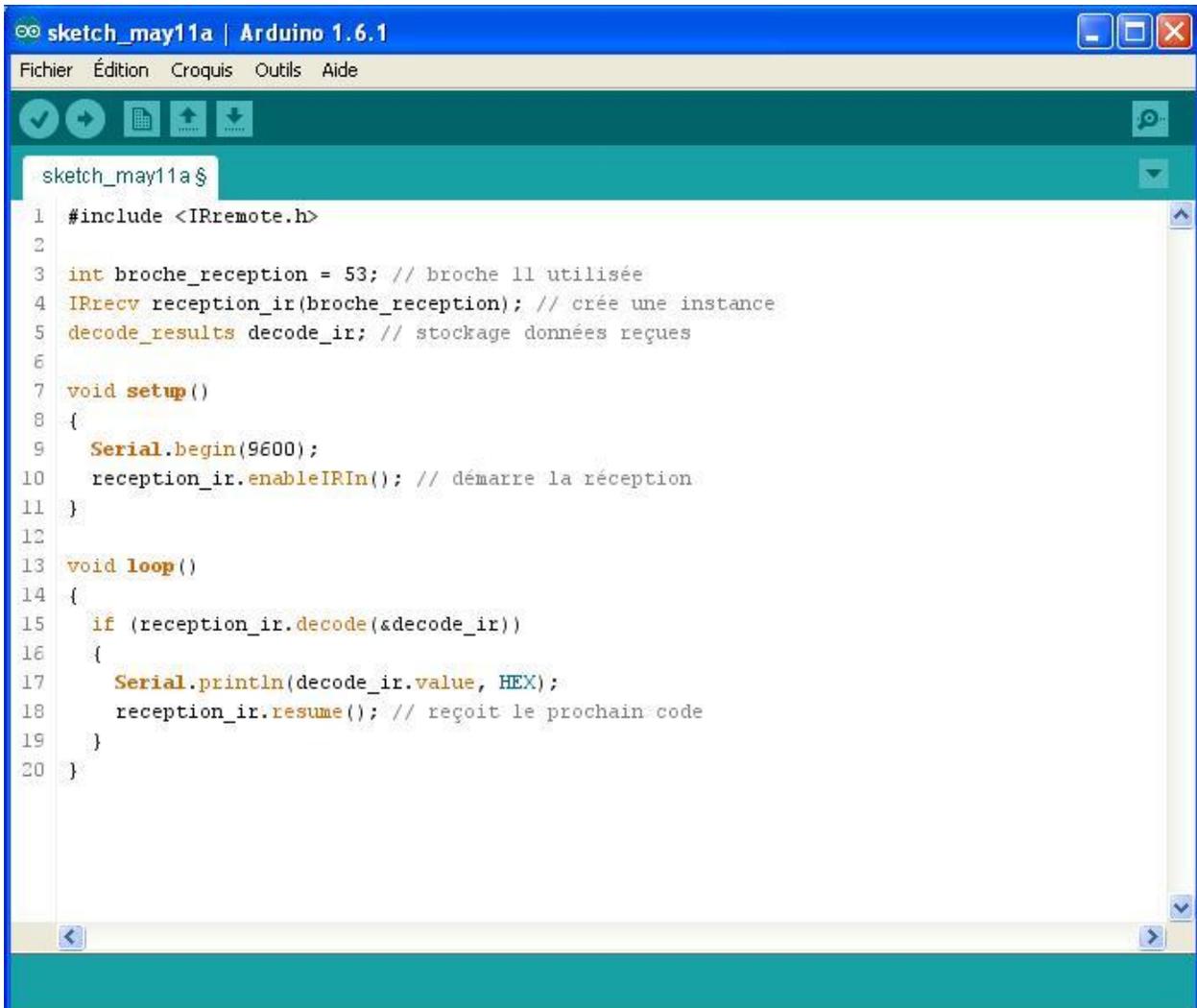
*Figure III.2 Organigramme esclave*

### III.3 La librairie IRremote sous Arduino

Ce programme utilise un capteur à infrarouge et une télécommande voir chapitre II pour identifier les codes transmis par la télécommande au capteur infrarouge et affiche le code transmis dans le moniteur série de l'éditeur Arduino. Ce qui permet de programmer l'action de chacun des boutons de la télécommande.

Depuis l'interface l'IDE Arduino, un Gestionnaire de bibliothèques est disponible par la commande Croquis/Inclure une bibliothèque/Gérer les bibliothèques... Il permet de retrouver une liste détaillée des bibliothèques présentes, de les mettre à jour lorsqu'une évolution a été détectée chez son créateur, et d'en installer de nouvelles propositions. Cette liste peut être triée par thème, ou par état (installées, à mettre à jour...).

La librairie « IRremote » permet d'émettre et de recevoir via l'infrarouge ; elle permet de gérer les protocoles NEC, Sony, Philips RC5/RC6 et le traitement de données brutes. Nous allons voir comment la mettre en œuvre.



```

sketch_may11a $
1  #include <IRremote.h>
2
3  int broche_reception = 53; // broche 11 utilisée
4  IRrecv reception_ir(broche_reception); // crée une instance
5  decode_results decode_ir; // stockage données reçues
6
7  void setup()
8  {
9      Serial.begin(9600);
10     reception_ir.enableIRIn(); // démarre la réception
11 }
12
13 void loop()
14 {
15     if (reception_ir.decode(&decode_ir))
16     {
17         Serial.println(decode_ir.value, HEX);
18         reception_ir.resume(); // reçoit le prochain code
19     }
20 }

```

*Figure III.3 Fenêtre du programme esclave*

Pour décoder un signal émis par la télécommande, il faut lire chaque temps où le signal est pulsé et chaque temps où il ne l'est pas. L'avantage d'un capteur de type RC5, c'est qu'il filtre les signaux à 38Khz (si l'on possède le bon capteur) et du coup nous pouvons tenter de capter un signal haut (et non une succession de signaux haut et bas). C'est toute la différence avec une photodiode qui capterait tous les signaux émis.

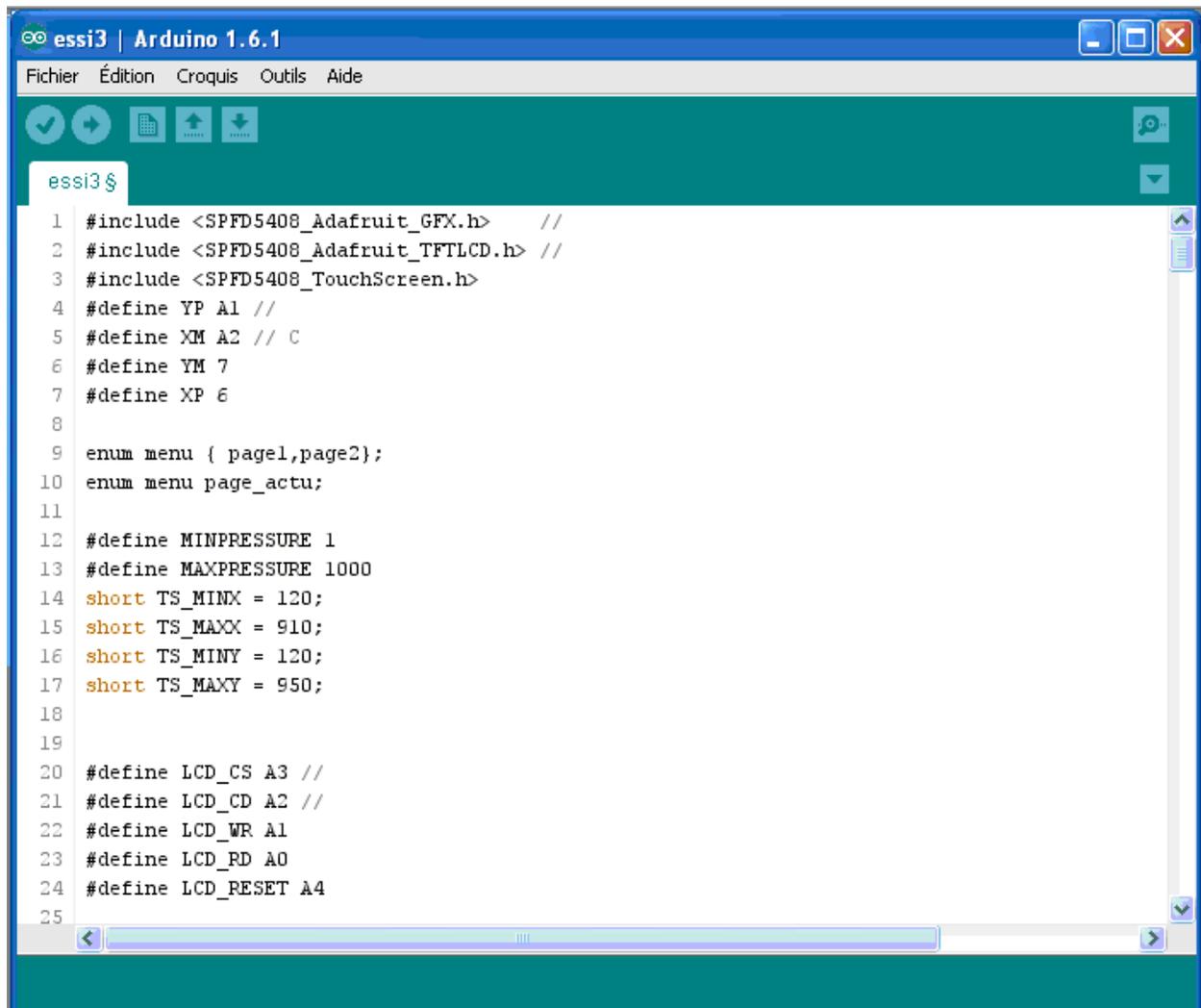
### III.4 Explication du programme

Notre programme maître nous permet de lire des trames des données et des codes envoyés via la liaison IR, ces trames de commande sont déjà implantées à notre Arduino (mémoire) ; ce dernier rappelle les caractères qui signifient l'action exacte sur le croisement lignes colonnes

« action matricielle 7×3 de la télécommande ». D'où l'interruption sera identifiée et active ou désactive un des ces huit équipements domotiques.

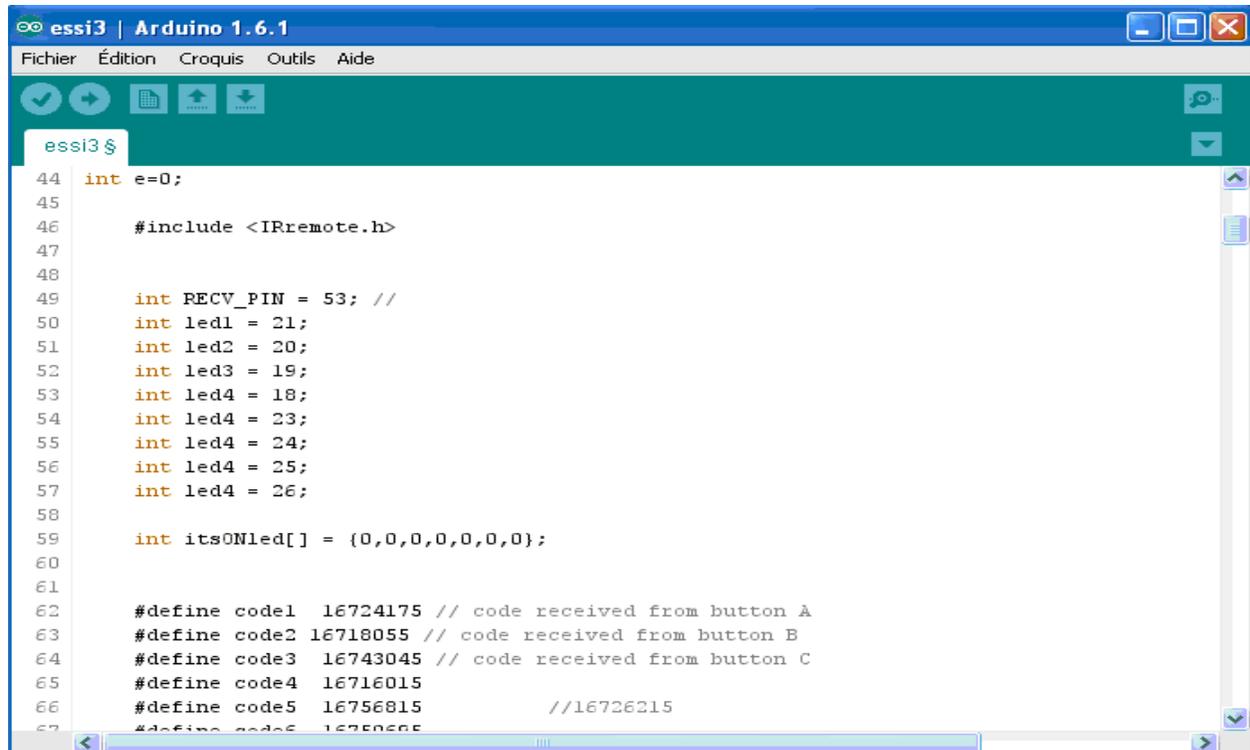
### III.5 Présentation du programme IDE

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



```
1 #include <SPFD5408_Adafruit_GFX.h> //
2 #include <SPFD5408_Adafruit_TFTLCD.h> //
3 #include <SPFD5408_TouchScreen.h>
4 #define YP A1 //
5 #define XM A2 // C
6 #define YM 7
7 #define XP 6
8
9 enum menu { page1,page2};
10 enum menu page_actu;
11
12 #define MINPRESSURE 1
13 #define MAXPRESSURE 1000
14 short TS_MINX = 120;
15 short TS_MAXX = 910;
16 short TS_MINY = 120;
17 short TS_MAXY = 950;
18
19
20 #define LCD_CS A3 //
21 #define LCD_CD A2 //
22 #define LCD_WR A1
23 #define LCD_RD A0
24 #define LCD_RESET A4
25
```

Figure III.4 Fenêtre initialisation du TFT

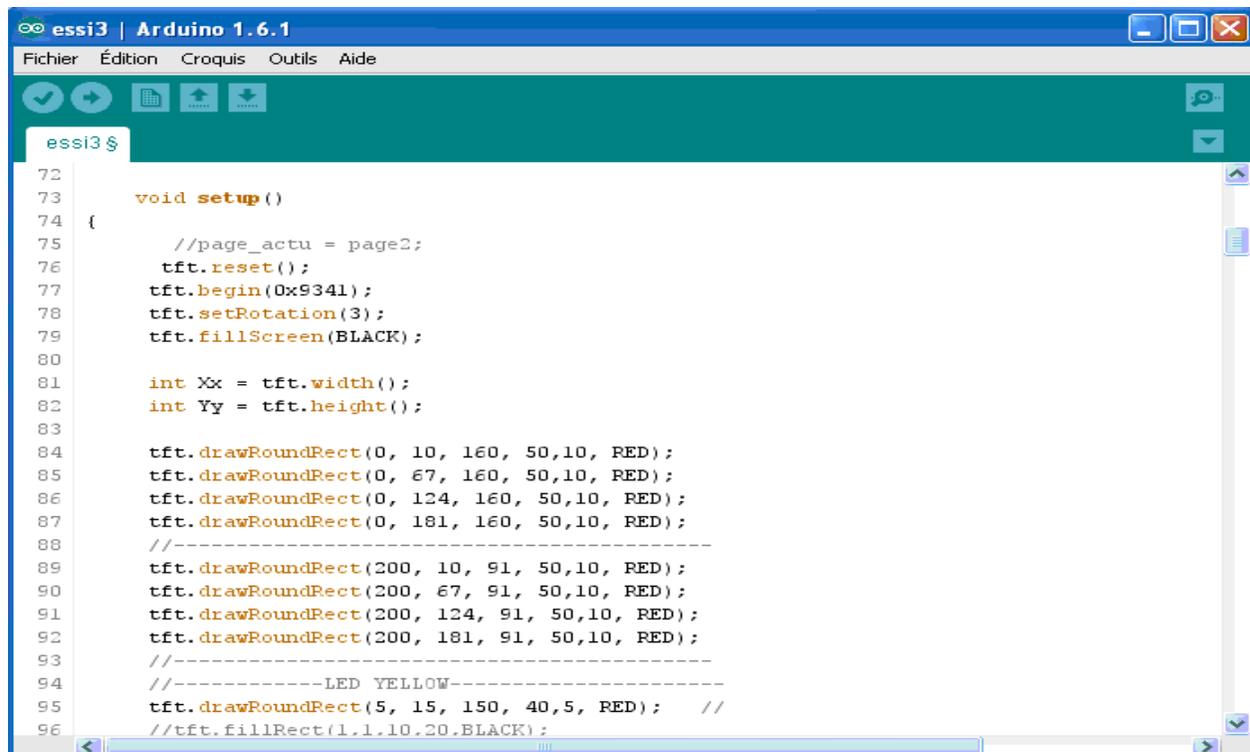


```

ess3 | Arduino 1.6.1
Fichier  Édition  Croquis  Outils  Aide
ess3 $
44 int e=0;
45
46 #include <IRremote.h>
47
48
49 int RECV_PIN = 53; //
50 int led1 = 21;
51 int led2 = 20;
52 int led3 = 19;
53 int led4 = 18;
54 int led4 = 23;
55 int led4 = 24;
56 int led4 = 25;
57 int led4 = 26;
58
59 int itsONled[] = {0,0,0,0,0,0,0};
60
61
62 #define code1 16724175 // code received from button A
63 #define code2 16718055 // code received from button B
64 #define code3 16743045 // code received from button C
65 #define code4 16716015
66 #define code5 16756815 //16726215
67 #define code6 16756815

```

Figure III.5 Fenêtre initialisation de la librairie « IRremote »



```

ess3 | Arduino 1.6.1
Fichier  Édition  Croquis  Outils  Aide
ess3 $
72
73 void setup()
74 {
75     //page_actu = page2;
76     tft.reset();
77     tft.begin(0x9341);
78     tft.setRotation(3);
79     tft.fillScreen(BLACK);
80
81     int Xx = tft.width();
82     int Yy = tft.height();
83
84     tft.drawRoundRect(0, 10, 160, 50,10, RED);
85     tft.drawRoundRect(0, 67, 160, 50,10, RED);
86     tft.drawRoundRect(0, 124, 160, 50,10, RED);
87     tft.drawRoundRect(0, 181, 160, 50,10, RED);
88     //-----
89     tft.drawRoundRect(200, 10, 91, 50,10, RED);
90     tft.drawRoundRect(200, 67, 91, 50,10, RED);
91     tft.drawRoundRect(200, 124, 91, 50,10, RED);
92     tft.drawRoundRect(200, 181, 91, 50,10, RED);
93     //-----
94     //-----LED YELLOW-----
95     tft.drawRoundRect(5, 15, 150, 40,5, RED); //
96     //tft.fillRect(1,1,10,20,BLACK);

```

Figure III.6 Fenêtre du menu du TFT

```

156 void loop() {
157     if (irrecv.decode(&results)) {
158         unsigned int value = results.value;
159         switch(value) {
160             case code1:
161                 if(!itsONled[1]) { // if first led is on then
162                     digitalWrite(led1, LOW); // turn it off when button is pressed
163                     tft.fillRoundRect(7, 18, 146, 36,2, YELLOW);
164                     tft.setCursor(20,28);
165                     tft.setTextColor(BLACK);
166                     tft.setTextSize(2);
167                     tft.print("CHARGE 1");
168                     tft.fillRoundRect(206, 16, 78, 38,10, BLACK);
169                     tft.setCursor(235,27);
170                     tft.setTextColor(YELLOW);
171                     tft.setTextSize(2);
172                     tft.print("ON");
173                     delay(150);
174                     itsONled[1] = 0; // and set its state as off
175                 }
176             else { // else if first led is off
177                 digitalWrite(led1, HIGH); // turn it on when the button is pressed
178                 tft.fillRoundRect(7, 18, 146, 36,2, BLACK);
179                 tft.setCursor(20,28);
180                 tft.setTextColor(YELLOW);

```

Figure III.7 Fenêtre des boucles de la temporisation

```

865     tft.setTextColor(GREEN);
866     tft.setTextSize(2);
867     tft.print("ON");
868     e = 1;
869     delay(150);
870 }
871 else if (e == 1) // Si la variable flag esta en 1
872 {
873     digitalWrite(led8, HIGH);
874     tft.fillRoundRect(7, 189, 146, 36,2, BLACK);
875     tft.setCursor(29,199);
876     tft.setTextColor(GREEN);
877     tft.setTextSize(2);
878     tft.print("CHARGE 8");
879     tft.fillRoundRect(206, 187, 78, 38,10, BLACK);
880     tft.setCursor(230,199);
881     tft.setTextColor(RED);
882     tft.setTextSize(2);
883     tft.print("OFF");
884     e = 0;
885 }
886 }
887
888 //-----
889

```

Figure III.8 Fenêtre d'activation et désactivation des équipements domotiques

Dans un premier temps, il est souhaitable pour la compréhension du programme et éventuellement désirant suivre ou modifier le programme et de définir des équivalences qui permettent d'associer une valeur à un nom, il est plus facile quelquefois de travailler avec des noms plutôt qu'avec des nombres. Pour définir ces équivalences et créer un programme source on tape la syntaxe suivante :

Port RP17 ; Le port RP17 se trouve à l'adresse 33

Port RB18 ; le port RP18 se trouve à l'adresse 34

Port RB19 ; le port RP19 se trouve à l'adresse 35

.....

Port RB25 ; le port RP25 se trouve à l'adresse 41

A chaque fois dans notre programme que le compilateur rencontrera l'équivalence définie, il la remplacera par sa valeur numérique. Il est plus facile de comprendre dans un programme le mot Inter plutôt que la valeur.

On peut structurer notre programme comme suit :

```
//Titre : réalisation d'une commande domotique par IR et tactile via Arduino
```

```
// Date : .././2017
```

```
// Auteur : Mr .....
```

```
; ----- appeler les bibliothèques TFT + IR -----
```

```
.....
```

```
; ----- Définition des constantes -----
```

```
.....
```

```
; ----- Définition des variables -----
```

```
.....
```

```
; ----- Initialisation du E/S -----
```

```
.....
```

```
; -----initialisation des temporisateurs-----
```

```
.....
```

```
; ----- Réalisation d'un menu de commande -----
```

```
.....
```

```
; ----- Programme principal -----
```

```
.....
```

```
; ----- Programme d'interruption -----
```

```
.....
```

```
; -----Exécution des données-----
```

```
.....
```

```
; ----- Fin -----
```

### III.6 Conclusion

Sur le plan pratique, une manipulation du logiciel « IDE de l'Arduino » nous permet de programmer facilement de :

- Commander à distance via IR « en utilisant une télécommande celle du MP3 pour voiture » huit équipements « quatre lampes + quatre prise de 220V ».
- Commander et contrôler via un TFT tactile manuellement ces derniers équipements.
- Connaître l'action de l'utilisateur sur le clavier de la télécommande 3×7 pour connaître les éléments de croisement lignes colonnes et celui-ci se résume en codes d'affichage sur le TFT. voir référence [19].
- Visualiser les actions effectuées sur écran TFT.

# Conclusion Générale

## 1. Généralités

Nos contacts avec les matrices sont des actions habituelles « téléphones, télécommandes, claviers d'un ordinateur, serrures codées, visionneuses, etc. » mais peu de nous ne connaissent pas ce qui se passe à l'intérieur.

Dos nos jours, les technologies de l'infrarouge parmi les nombreux systèmes utilisent dans la commande à distance cette technologie basée sur les émissions des rayonnements électromagnétiques invisibles.

La commande domotique de nos jours est quasi totalement influée par les techniques IR. Dans ce travail, nous avons présenté un système de contrôle et de commande à distance très pratique simple et souple à réaliser par Arduino en collaboration avec une télécommande infrarouge et un TFT tactile. Toutefois, nous ne nous attarderons que sur la commande domotique par infrarouge qui fait l'objet de notre étude.

- Le but de notre projet est de projeter la lumière sur les systèmes embarqués spécialement les cartes Arduino, ainsi que la simplicité de la commande domotique.
- La programmation d'Arduino exige une connaissance en langage assembleur « C, IDE » comme soft, afin d'arriver à l'objectif souhaité.

## 2. Problèmes rencontrés

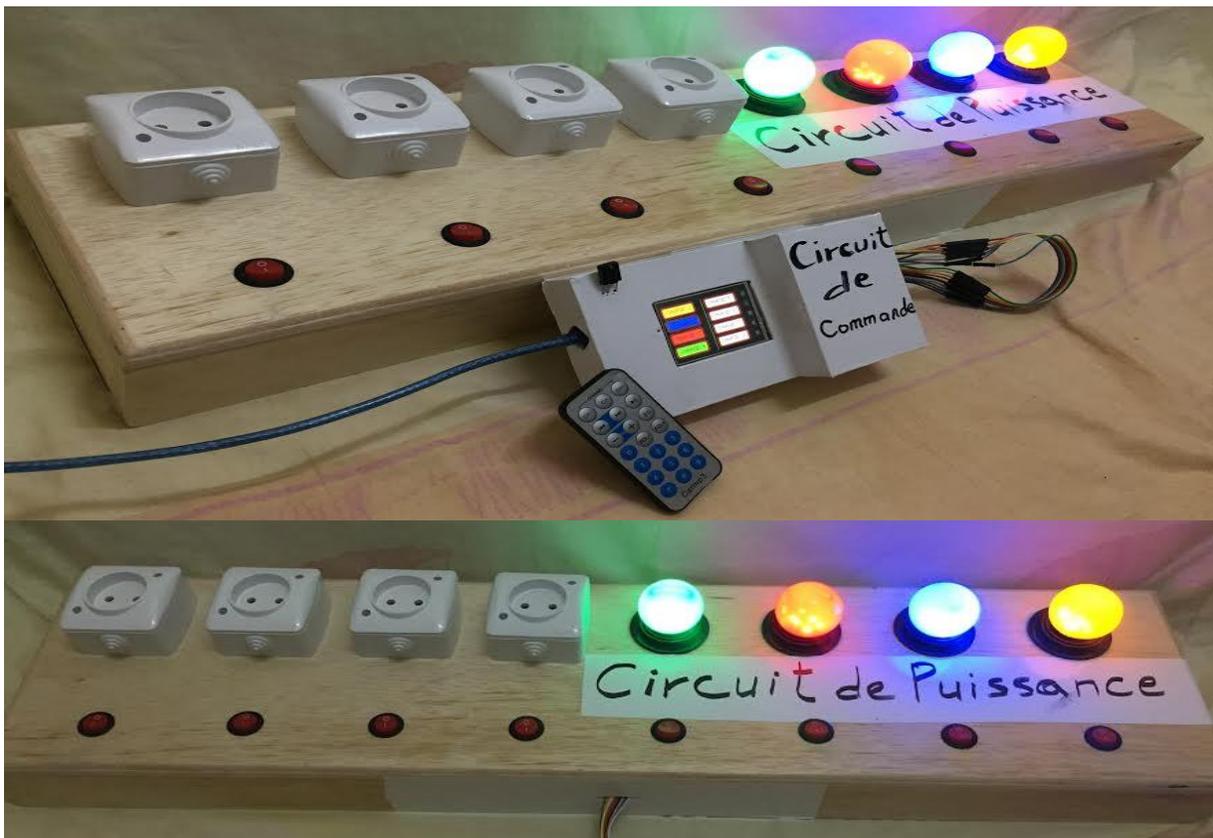
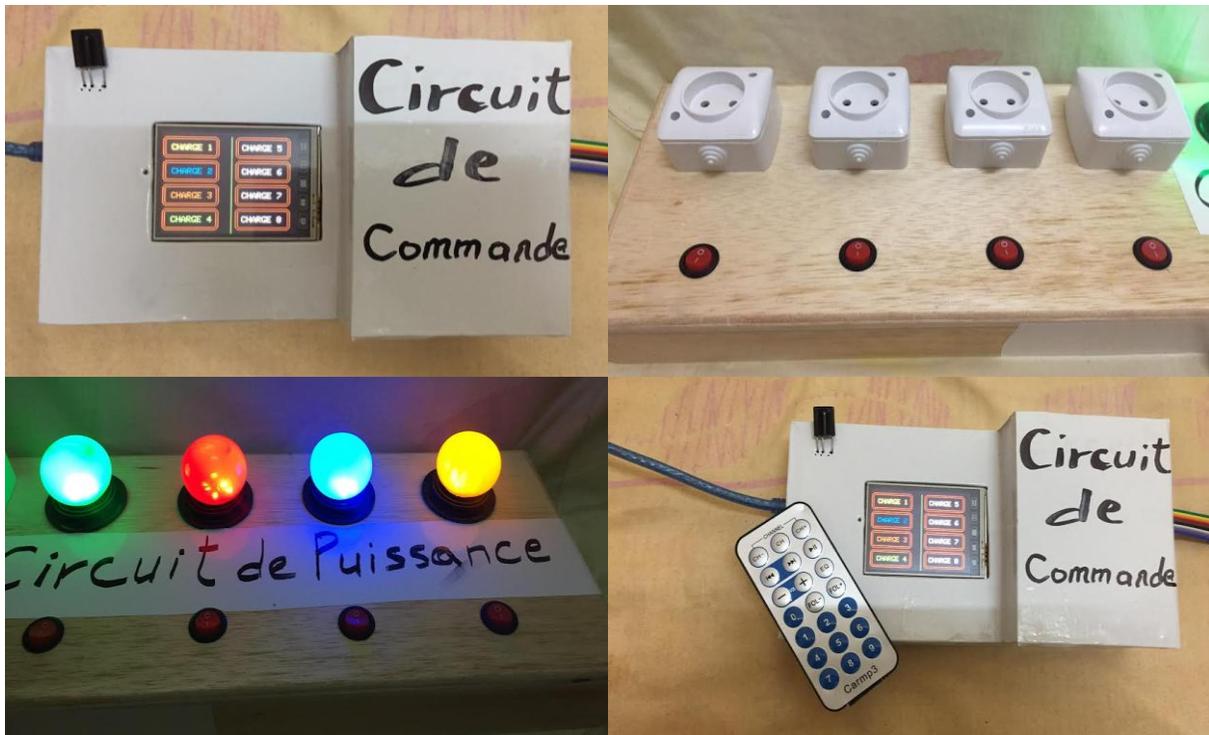
Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes confrontés à plusieurs problèmes (insuffisance du temps, manque collaboration étudiants, indisponibilité matériel adéquat...).

Cependant, on peut dire que malgré ces difficultés, les résultats obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études.

## 3. Perspectives du projet

Nous souhaitons vivement que ce projet puisse servir comme élément de base pour d'autres études plus approfondies pour le faire intégrer sous des systèmes plus complexes.

# Annexes



Les résultats obtenus expérimentalement

## Les Logiciels Utilisés



Pour programmer arduino



Pour la simulation virtuelle de notre carte

:1085100086957105610508940895E894BB27662740  
:108520007727CB0197F908950BD0C4CFB5DF28F09A  
:10853000BADF18F0952309F0A6CFABCF1124EECF08  
:10854000CADFA0F3959FD1F3950F50E0551F629FAE  
:10855000F001729FBB27F00DB11D639FAA27F00D9C  
:10856000B11DAA1F649F6627B00DA11D661F829FC3  
:108570002227B00DA11D621F739FB00DA11D621FA8  
:10858000839FA00D611D221F749F3327A00D611DC5  
:10859000231F849F600D211D822F762F6A2F1124A7  
:1085A0009F5750408AF0E1F088234AF0EE0FFF1FFA  
:1085B000BB1F661F771F881F91505040A9F79E3F31  
:1085C000510570F060CFAACF5F3FECF3983EDCF32B  
:1085D000869577956795B795F795E7959F5FC1F76E  
:1085E000FE2B880F911D9695879597F90895DB01CD  
:1085F0008F939F936AD0BF91AF91A29F800D911DE1  
:10860000A39F900DB29F900D11240895A1E21A2E00  
:10861000AA1BBB1BFD010DC0AA1FBB1FEE1FFF1F26  
:10862000A217B307E407F50720F0A21BB30BE40B76  
:10863000F50B661F771F881F991F1A9469F76095BD  
:108640007095809590959B01AC01BD01CF01089577  
:10865000052E97FB16F400940FD057FD05D0D6DFFA  
:1086600007FC02D046F408C05095409530952195FE  
:108670003F4F4F4F5F4F089590958095709561954E  
:108680007F4F8F4F9F4F0895EE0FFF1F0590F4917E  
:10869000E02D199405D0332312F48A1B9B0B10C0D4  
:1086A000A29FB001B39FC001A39F700D811D112433  
:1086B000911DB29F700D811D1124911D0895F0DF51  
:1086C000B7FF0895821B930B0895EADFA59F900DD5  
:1086D000B49F900DA49F800D911D112408950790C3  
:1086E000F691E02D19940F93083090F0982F872F72  
:1086F000762F652F542F432F322F22270850F4CF87  
:10870000220F331F441F551F661F771F881F991F95  
:108710000A95B2F70F91089597FB10F81694000888  
:108720000F93083098F00850232F342F452F562FE1  
:10873000672F782F892F902DF4CF059497958795E3  
:108740007795679557954795379527950A95AAF791  
:108750000F91089581E090E0F89400C0F894FFCF65  
:108760000100000017003500B60378008E03780082  
:108770000143484152474520310000000009E0C53  
:10878000E13FD1401C0722083908D3074F0884086D  
:10879000A20860C0A028100010181060C0A000003F  
:1087A0003F013F01EFEF00000000009E0CE13F742D  
:1087B000201C07051E9E1DD3076C1EF81EA2080074  
:1087C000000000723EE13FC73DE03DD23D513E0D0D  
:1087D0000A006E616E00696E66006F7666002E009C  
:00000001FF

Le code hexadécimal de notre programme

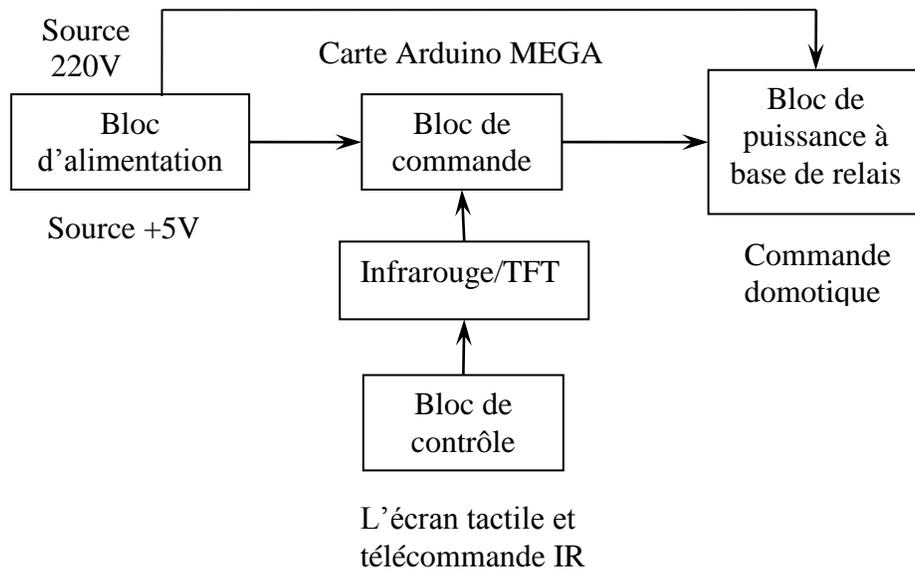


Schéma synoptique du dispositif

- [1] « Les automatismes du bâtiment La domotique Le maintien à domicile ». Décembre 2011. Liver.
- [2] MM. Alleguede et Eckert « Le guide de la domotique ». 2012 . Liver.
- [3] <http://www.commentcamarche.net/forum/affich-5013634-liaisons-rs232-et-rs485> mai 2017
- [4] <http://www.commentcamarche.net/contents/770-port-serie-et-port-parallele> mai 2017
- [5] [http://www.mon-club elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.Librairie](http://www.mon-club elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.Librairie) Wire mai 2017
- [6] <https://www.inspection-eb.com/thermographie> mai 2017
- [7] Armand Hadni. « L'infrarouge ». avril 2011. Liver.
- [8] <http://blog.ac-versailles.fr/technopeguy/public/Programmation/Arduino.pdf> mai 2017
- [9] [https://www.flossmanualsfr.net/\\_booki/arduino/arduino.pdf](https://www.flossmanualsfr.net/_booki/arduino/arduino.pdf) mai 2017
- [10] ERIK Bartmann. « Le grand livre d'Arduino ». 2<sup>eme</sup> Edition. 2015. Liver.
- [11] <http://fr.farnell.com/atmel/atmega2560-16au/micro-8-bits-256k-flash-5v-cms/dp/1288330#techDocsHook> mai 2017
- [12] [http://www.mon-club elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.LibrairieSPI](http://www.mon-club elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.LibrairieSPI). mai 2017
- [13] <http://igm.univ-mlv.fr/~duris/NTREZO/20042005/Nguyen-Vongvilay-WolowiecBluetooth.pdf> mai 2017
- [14] <http://www.schema-electronique.net/2010/04/un-emetteur-de-telecommande-infrarouge.html> mai 2017
- [15] <http://electro8051.free.fr/Datasheet/SAA3004.pdf> mai 2017
- [16] D. MENESPLIER. « Code RC5 ». 2002. Liver.
- [17] [http://ouilogique.com/2-4-in\\_TFT\\_Touch\\_screen/](http://ouilogique.com/2-4-in_TFT_Touch_screen/) mai 2017
- [18] A. KRAMA, A. GOUGUI. « Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde ». Mémoire master académique. Université Ouargla 2015.
- [19] « LCD Module Product Spécification ». 19 Décembre 2012. These.

## Résumé :

La démarche de la réalisation de notre projet nous permet de naviguer dans le monde de communication entre les appareils électroniques et les accessoires de commande afin de créer un environnement de commande et contrôle souple et simple.

Un tel système embarqué « ARDUINO » nous permet de réaliser une commande à distance à huit équipements pour une maison intelligente « la commande domotique » via le protocole infrarouge et la collaboration d'un écran tactile qui prend la tâche d'afficher les actions et l'état des équipements qui peut même assurer manuellement la commande lors d'une défaillance de contrôle à distance.

**Mots-clés :** carte Arduino, écran tactile, la technique de rayons infrarouges.

## Abstract:

The object of our project is to allow us to navigate in the world of communication between electronic devices and control accessories to create a flexible and simple controlling of our environment.

Such an embedded system « ARDUINO » allows us to remote controlling of eight equipments for a smart home « the home automation control » via the infrared protocol and the collaboration of a touch screen that takes the task of displaying actions and The state of the equipment ; which can even use the manual control when remote control failure.

**Keywords:** Arduino card, touch screen, infrared ray technique.

## ملخص :

الهدف من مشروعنا هو السماح لنا بالغوص في عالم الاتصالات بين الأجهزة الإلكترونية والتحكم في ملحقاتها، وهذا لخلق بيئة تحكم وسيطرة مرنة وبسيطة على محيطنا.

هذا نظام توافقي « أردوينو » حيث يسمح لنا بالسيطرة عن بعد على ثمانية معدات خاصة بالمنزل الذكي « التحكم الآلي للمنزل » عبر تقنية الأشعة تحت الحمراء بالإضافة إلى الاستعانة بشاشة تعمل باللمس التي تأخذ مهمة عرض الإجراءات وحالة المعدات. والتي يمكن استخدامها في التحكم اليدوي عندما يفشل نظام التحكم عن بعد.  
**الكلمات المفتاحية:** بطاقة أردوينو، شاشة تعمل باللمس، تقنية أشعة تحت الحمراء.