

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Djilali Bounaama - Khemis Miliana



Faculté des Sciences et de la Technologie
Département de la Technologie

Mémoire du Projet de Fin d'Etudes
Pour l'obtention de diplôme

Master

En

« Automatique »

Option :

« Automatique et informatique industrielle »

Thème :

**Etude et réalisation d'une carte de contrôle par
Arduino via le système Android**

Réalisé par :

BOUCHERIT Ahmed
MERAIMI Mouloud

Encadré par :

Mr. MAHDAB Salim

Année Universitaire 2017/2018

Dédicace

Je tiens à la fin de ce travail à remercier ALLAH le tout puissant de m'avoir donné la foi et de m'avoir permis d'en arriver là.

Je remercie infiniment mon encadreur Mr. (MAHDAB). Et je remercie aussi tous les profs de la Faculté des Sciences et de la Technologie de Université de Djilali Bounaama Khemis Miliana.

Mes remerciements vont également à ma mère pour tous les sacrifices pour me permettre de suivre mes études dans les meilleures conditions possibles et n'avoir jamais cessé de m'encourager tout au long de mes années d'étude.

Enfin, je remercie particulièrement mes collègues et mes amis, qui ont rendu cette formation très sympathique.

Boucherit Ahmed

Du fond de cœur je dédie ce travail, à mon père qui m'a encouragé, à ma mère qui dieu la protège, à mes frères et mes sœurs, mes tantes et mes oncles.

A toute la famille MERAIMI, a tous mes collègues, a tous mes amis et mon binôme

Boucherit Ahmed, à mon promoteur Salim MAHDAB.

Tous et celles dont les noms n'ont pu être cités, et a tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci d'être toujours là pour moi.

Remerciements

En préambule à ce mémoire nous remerciant ALLAH qui nous aide et nous donne la patience et le courage durant ces longues années d'étude. Nous souhaitant 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.

Ces remerciements vont tout d'abord au corps professoral et administratif de la Faculté des Sciences et technologie, pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

Nous tenant à remercier sincèrement Monsieur, Mahdab, qui, en tant qu'Encadreur de mémoire, il est toujours montrés à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'il est bien voulu nous consacrer et sans qui ce mémoire n'aurait jamais vu le jour. On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience. Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragées au cours de la réalisation de ce mémoire. Merci à tous et à toutes.

Résumé

Ce projet nous permet de contrôler une maison intelligente par un smart phone qui utilise une application Android créée par nous par donner des commandes a une carte Arduino via un module Bluetooth.

L'objectif est la réalisation d'un circuit imprimé qui émule une maison intelligente avec des moteurs DC et quelque LEDs contrôlée par un smart phone Android.

Mots-clés :

Système Android - carte Arduino - langage de programmation - Moteurs à courants continus-Circuit imprimé - port série.

Abstract

This project allows us to control a smart home by a smart phone that uses an Android application that we created by giving commands to an Arduino board via a Bluetooth module.

The objective is the creation of a PCB (Printed Circuit Board) that emulates a smart home with DC motors and some LEDs controlled by an Android smart phone.

Keywords:

Android system- Arduino board - Programming language - DC Motor- Printed circuit board – Serial port.

ملخص

هذا المشروع يسمح لنا بالتحكم بمنزل ذكي بواسطة هاتف ذكي يستعمل تطبيق خاص بالأندرويد و ذلك بإعطاء الأوامر لبطاقة الأردوينو عن طريق البلوتوث.

الهدف هو انشاء بطاقة الكترونية تحاكي منزل ذكي يحتوي على محركات تيار مستمر وبعض المصابيح يتحكم بها عن طريق هاتف ذكي بنظام الأندرويد.

كلمات مفتاحية :

نظام الأندرويد – بطاقة الأردوينو – لغة برمجة – محرك تيار مستمر – لوحة دارات مطبوعة – منفذ تسلسلي.

SOMMAIRE

INTRODUCTION GENERALE	1
-----------------------------	---

CHAPITRE I : GENERALITES SUR LA DOMOTIQUE

I.1 Introduction	3
I.2 Définition.....	3
I.3 Les applications de domotique	3
I.3.1 La santé	3
I.3.2 Les économies d'énergie	4
I.3.3 Le confort et les loisirs.....	4
I.3.4 La sécurité.....	4
I.4 Technique de la domotique.....	5
I.4.1 La centrale de commande	5
I.4.2 L'écran de contrôle.....	5
I.4.3 Le pilotage à distance.....	5
I.4.4 Fonctionnement des interfaces de contrôle.....	7
I.5 Conclusion	7

CHAPITRE II : ARDUINO ET DIFFERENTS MODULES

II.1 Introduction	8
II.2 Généralité sur la carte ARDUINO UNO.....	8
II.3 Le but et l'utilité.....	9
II.4 Les bonnes raisons de choisir Arduino	9
II.4.1 Le prix.....	9
II.4.2 La liberté.....	9
II.4.2.1 Le logiciel	9
II.4.2.2 Le matériel	9
II.4.3 La compatibilité.....	9

II.4.4 La communauté	10
II.5 Présentation de la carte.....	10
II.6 Constitution de la carte.....	10
II.6.1 Le microcontrôleur	10
II.6.2 Alimentation	11
II.6.3 Visualisation	11
II.6.4 La connectique.....	11
II.7 Installation.....	12
II.7.1 Sous Windows	12
II.8 Partie programme	13
II.8.1 L'environnement de la programmation	13
II.8.2 Structure générale du programme (IDE Arduino).....	13
II.8.3 Injection du programme.....	14
II.8.4 Description du programme	15
II.8.5 Commentaires	15
II.8.6 Définition des variables	15
II.8.7 Programmation des interactions void loop	16
II.9 Les accessoires de la carte Arduino	17
II.9.1 Communication	17
II.9.1.1 Les modules Bluetooth HC-05 et HC-06.....	17
II.9.1.2 Le module shield Arduino wifi	18
II.9.2 Les drives.....	18
II.9.2.1 Module de commande puissance L298HN	18
II.9.3 Les relais.....	20
II.10 Conclusion.....	20

CHAPITRE III : PRINCIPE DE FONCTIONNEMENT ET LA REALISATION VIRTUELLE

III.1 Introduction.....	21
II.2 Les différentes étapes de la réalisation.....	21
III.3 Schéma synoptique général.....	22
III.3.1 Bloc d'alimentation	23

III.3.2 Bloc de contrôle	24
III.3.3 Bloc de commande	24
III.3.4 Bloc de puissance	26
III.3.4.1 Etage 1	26
III.3.4.2 Etage 2	26
III.4 La réalisation virtuelle (Simulation) :	28
III.5 Conclusion	29

CHAPITRE IV : DEVELOPPEMENT DE L'APPLICATION ANDROID

IV.1 Introduction.....	30
IV.2 L'écosystème d'Android.....	31
IV.3 L'Operating System	31
IV.3.1 Dalvik et ART	32
IV.4 L'environnement Android Studio :	32
IV.4.1 Avantages.....	33
IV.4.2 Désavantages.....	33
IV.5 Les éléments d'une application	33
IV.6 Langage java	34
IV.7 Développement de l'application :	34
IV.7.1 Partie graphique	34
IV.7.1.1 Design de l'application.....	36
IV.7.2 Partie programmation.....	38
IV.8 Présentation du programme IDE.....	40
IV.8.1 Fichier JAVA	40
IV.8.3 Explication de programme	41
IV.9 Conclusion	42

CHAPITRE V : REALISATION DU DISPOSITIF EXPERIMENTAL

V.1 Introduction	43
V.2 Présentation de l'organigramme IDE.....	43

V.2.1 Organigramme Arduino UNO.....	43
V.3 Présentation du programme IDE.....	45
V.3.1 Explication de programme	48
V.4 La réalisation du circuit imprimé	48
V.5 Composants utilisés.....	50
V.6 Conclusion	50
CONCLUSION GENERALE	51

Liste des figures

Figure II.1 La carte Arduino UNO.....	8
Figure II.2 Presentation de la carte Arduino	10
Figure II.3 Une carte Arduino etendue avec un Ethernet shield	12
Figure II.4 Carte connectée et alimentée.....	12
Figure II.5 Interface IDE Arduino.....	13
Figure II.6 Paramétrage de la carte étape 1	14
Figure II.7 Paramétrage de la carte étape 2	14
Figure II.8 Les étapes de téléchargement du code	17
Figure II.9 Type de modules Bluetooth	18
Figure II.10 module shield wifi.....	18
Figure II.11 Les détails techniques du L298N	19
Figure II.12 relais 12V	20
Figure III.1 Schéma synoptique du dispositif	22
Figure III.2 Schéma synoptique de l'alimentation.....	23
Figure III.3 Schéma électrique de l'alimentation.....	23
Figure III.4 Schéma simplificateur de la configuration BT	24
Figure III.5 Module L298HN (H=Modèle Horizontal).....	26
Figure III.6 brochage et structure interne du circuit L298	26
Figure III.7 Schéma de brochage L298	28
Figure III.8 La carte réalisée sous ISIS-PORTEUS	29
Figure IV.1 Anatomie d'un déploiement.....	32
Figure IV.2 premier fenêtre pour créer une application sur Android Studio	35
Figure IV.3 Fenêtre pour sélectionner la version API de notre application.....	35
Figure IV.4 Fenêtre pour choisir l'Activité	36
Figure IV.5 Vue principale d'Android Studio	36
Figure IV.6 L'interface graphique de l'exemple 'Hello World'	37
Figure IV.7 Le code XML de l'exemple 'Hello World'	37
Figure IV.8 Les outils à utiliser dans l'interface graphique.....	38
Figure IV.9 L'interface finale de notre application	42
Figure V.1 Organigramme de notre programme	44
Figure V.2 Typon de circuit imprimée avec composants.....	49

Liste des tableaux

Tableau III.1 Tableau de commande Bluetooth.....	25
--------------------------------------------------	----

INTRODUCTION GENERALE

La forte augmentation des ventes de smart phone et de tablettes électronique se fait en même temps qu'une adoption rapide par le grand public des technologies de la domotique ainsi que l'autopilotage. Au fond, le smart phone, avec sa connectivité Bluetooth intégrée, devient une télécommande universelle pour toute la maison et les équipements électriques.

Dans la vie moderne, on n'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 contour) le plus grand nombre possible d'accessoires.

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

Dans ce projet trois objectifs ont été visés :

- Le premier est de regrouper suffisamment d'informations sur une grande catégorie de cartes d'interfaçage (Arduino) : son langage de programmation, sa construction, son principe de fonctionnement.
- Le deuxième est de réaliser une application sous smart phone et de le programmer sous l'environnement JAVA afin de simplifier la commande et de montrer l'intérêt de la programmation orientée objet.
- Le troisième consiste à réaliser une carte électrique capable d'exécuter une action entre un smart phone et une carte d'interfaçage (Arduino) en expliquant les différents blocs de sa construction.

Pour cela notre mémoire sera structurée en cinq chapitres :

Le premier chapitre est consacré sur la domotique son objectif, les avantages et ces applications dans la vie quotidienne.

Le deuxième chapitre sera consacré à une étude approfondie sur les cartes d'interface telles que l'Arduino, puis, on mettra la lumière sur un modèle de base qui est (Arduino UNO) sa construction son environnement de programmation et son principe de fonctionnement.

Le troisième chapitre sera consacré à l'étude et la réalisation virtuelle du dispositif expérimental ainsi que la description de chaque bloc du circuit.

Le quatrième chapitre sera consacré au présentation de système Android et l'IDE Android Studio et au développement de notre application Android et le principe de fonctionnement de cette application.

Le cinquième chapitre sera consacré à la réalisation du circuit imprimé avec la programmation d'Arduino.

CHAPITRE I

GENERALITES SUR LA DOMOTIQUE

I.1 Introduction

Pour donner du concret aux définitions, nous allons dans cette partie la définition de la domotique, les applications et les techniques que l'on peut avoir dans la domotique. Le marché de la domotique sera également évoqué dans cette partie.

I.2 Définition

La domotique est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans les bâtiments, plus ou moins « interopérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prises électriques, etc.). 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) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut retrouver dans les maisons, les hôtels, les lieux publics, etc.... [1]

I.3 Les applications de domotique

I.3.1 La santé

La première utilisation concerne la santé. Cette application de la domotique est essentiellement prévue pour le suivi des personnes fragiles (Personnes âgées, Handicapé lourd, Malentendants ou sourds...). On peut imaginer qu'un équipement installé sur une personne ou dans son domicile contrôle un certain nombre de paramètres comme : son rythme cardiaque, sa température corporelle, son taux de glycémie (pour le cas d'un diabétique), la qualité de l'air... On peut également imaginer un capteur sensoriel qui permet de détecter la perte de connaissance. En fonction des résultats mesurés par ces détecteurs, il va être possible de remonter des alertes vers des organismes compétents : services d'urgence, ... ou les proches.

I.3.2 Les économies d'énergie

Dans l'air du temps, les économies d'énergies sont aussi considérées en domotique. La solution pour les économies d'énergie est de minimiser le gaspillage. Il s'agit en fait de supprimer une énergie dépensée alors qu'elle n'est pas utilisée. Pour y arriver, disposer de l'information sur la consommation : on peut se trouver dans une pièce et voir les lampes allumées dans les autres sur un simple LED. De ce fait, on pourra piloter l'énergie pour qu'elle soit attribuée au besoin (à la demande). Mais certaines actions peuvent être gérées automatiquement par le système en place, par exemple, on peut :

- Ne chauffer que les pièces où les personnes se trouvent
- Gérer efficacement l'éclairages en mettant éteignant les lumières non utilisées
- Utiliser des lampes économiques

I.3.3 Le confort et les loisirs

Pour finir sur les besoins, les confort et les loisirs. Il existe de nombreuses applications pour les loisirs et le confort. Certaines actions peuvent être programmées. On peut par exemple programmer la préparation du café à la même heure tous les matins. On peut également citer tous les équipements électroménagers : tondeuse à gazon automatique, aspirateur qui détecte la saleté, ouverture et fermeture de portes de garages... Dans ce domaine, on profite des contrôles à distance pour qu'on avait déjà pour la télé, pour des fonctions plus avancées pour :

- Commander l'ouverture/fermeture des volets.
- Gérer un média center.

I.3.4 La sécurité

Aussi important que la Santé, la sécurité est aussi un point recherché. Quand on quitte son domicile on souhaite qu'elle reste surveillée. On cherche à éviter les intrusions, les tentatives de violations, intempéries (même légères). On a également d'autres types problèmes : incendie, inondation, fuite de gaz... La domotique peut proposer un panel très large de détecteurs et de capteurs

- Des mouvements
- Des inondations
- Le bris de verre
- Les vibrations
- L'ouverture de porte et fenêtre

- La fumée / feu
- L'humidité / inondation
- La pluie, le vent, la tempête

Une autre forme de sécurité de la maison est le fait d'être capable d'agir sur sa maison à distance. Il va ainsi être possible de bloquer ou libérer les serrures à distance. Ici encore, on peut être alerté en cas de détection de problèmes. [2]

I.4 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.4.1 La centrale de commande

Programmable et contenant des modules embarqués (passerelles domestiques) ou une interface micro-informatique (écran tactile, serveur, etc.) 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.

I.4.2 L'écran de contrôle

Il est fixe 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 partie des fonctions domotiques. L'interface distante peut être par exemple un ordinateur de poche, un téléphone portable ou Smartphone, une tablette tactile, une télécommande (universelle ou non), une interface sur télévision connectée, un écran + souris, etc...

I.4.3 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 Smartphone, 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'usager peut programmer certaines fonctions de la maison domotisée 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 »).

Exemples 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 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 - *a 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. Il existe un logiciel pédagogique de maison virtuelle à domotiser

I.4.4 Fonctionnement des interfaces de contrôle

Pour contrôler le serveur ces interfaces utilisent :

- Des surfaces de contrôle avec boutons, télécommandes.
- Des systèmes écran et souris, avec clavier avec ou sans fils existent.
- Des interfaces tactiles : écrans tactiles (PDA ou UMPC) associés à un logiciel ou une interface web.
- Des microphones permettent une activation par commande vocale (directement ou via téléphones GSM ou VoIP) associés à des logiciels de reconnaissance vocale.
- Des logiciels de reconnaissance des gestes, d'empreinte digitale, du visage.... [3]

I.5 Conclusion

L'évolution de la technologie et du mode de vie nous permet aujourd'hui de prévoir des espaces de travail et de logement mieux adaptés, tant en nouvelle construction qu'en rénovation. Nous devons ces nouvelles possibilités principalement aux progrès réalisés en électronique et à la nouvelle conception des réseaux de communication tant à l'intérieur qu'à l'extérieur des habitations. La domotique ouvre non seulement de nouvelles possibilités dans le domaine de l'automatisation de l'habitation, mais constitue aussi et surtout un moyen offert à l'individu de contrôler et de gérer son environne

CHAPITRE II

ARDUINO ET DIFFERENTS MODULES

II.1 Introduction

Dans ce chapitre, nous allons parler spécialement à la carte Arduino-UNO mentionnant les Caractéristiques et le principe de fonctionnement et nous donnons le schéma de principe de cette carte.

II.2 Généralité sur la carte ARDUINO UNO

La carte est composée de deux parties principales la partie matérielle représente la carte électronique est basée sur une simple composant à microcontrôleur (mini-ordinateur) ATMEGA328 (de la famille AVR) et la partie logicielle représente l'environnement de programmation Arduino C, cet environnement logiciel et matériel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

Donc l'Arduino est un projet en source ouverte (open source) peut profiter des ressources disponibles de trouver les réponses à ses questions.

Arduino est utilisé dans beaucoup d'applications comme l'électronique industrielle, Nous donnons la photo de la carte Arduino UNO. [4]

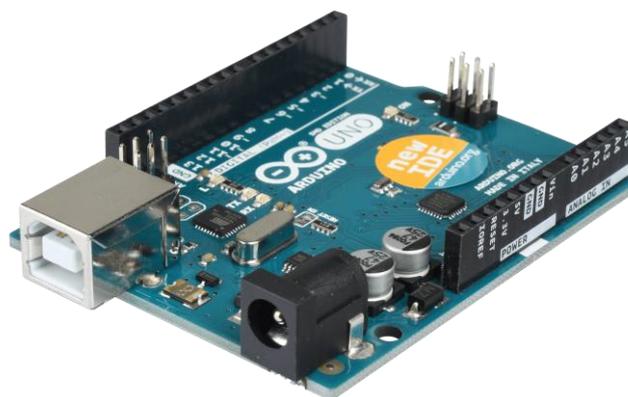


Figure II.1 La carte Arduino UNO

II.3 Le but et l'utilité

Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, nous allons programmer des systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique.

II.4 Les bonnes raisons de choisir Arduino

Il existe pourtant dans le commerce, une multitude de plateformes qui permettent de faire la même chose. Nous allons voir pourquoi choisir l'Arduino.

II.4.1 Le prix

En vue des performances qu'elles offrent, les cartes Arduino sont relativement peu coûteuses, ce qui est un critère majeur pour le débutant. Celle que nous utiliserons pour la suite du cours a un prix qui tourne aux environs de 4000 DA ce qui est un bon rapport qualité/prix.

II.4.2 La liberté

Elle constitue en elle-même deux choses :

II.4.2.1 Le logiciel

Gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris.

II.4.2.2 Le matériel

Cartes électroniques dont les schémas sont en libre circulation sur internet, cette liberté a une condition : le nom « Arduino » ne doit être employé que pour les cartes « officielles ».

Les cartes non officielles, on peut les trouver et les acheter sur Internet et sont pour la quasi-totalité compatibles avec les cartes officielles Arduino.

II.4.3 La compatibilité

Le logiciel, tout comme la carte, est compatible sous les plateformes les plus courantes (Windows, Linux et Mac), contrairement aux autres outils de programmation du commerce qui ne sont, en général, compatibles qu'avec Windows.

II.4.4 La communauté

La communauté Arduino est impressionnante et le nombre de ressources à son sujet est en constante évolution sur internet. De plus, on trouve les références du langage Arduino ainsi qu'une page complète de tutoriels sur le site arduino.cc (en anglais) et arduino.cc (en français).[5]

II.5 Présentation de la carte

Pour commencer notre découverte de la carte Arduino. Nous allons voir comment s'en servir et avec quoi.

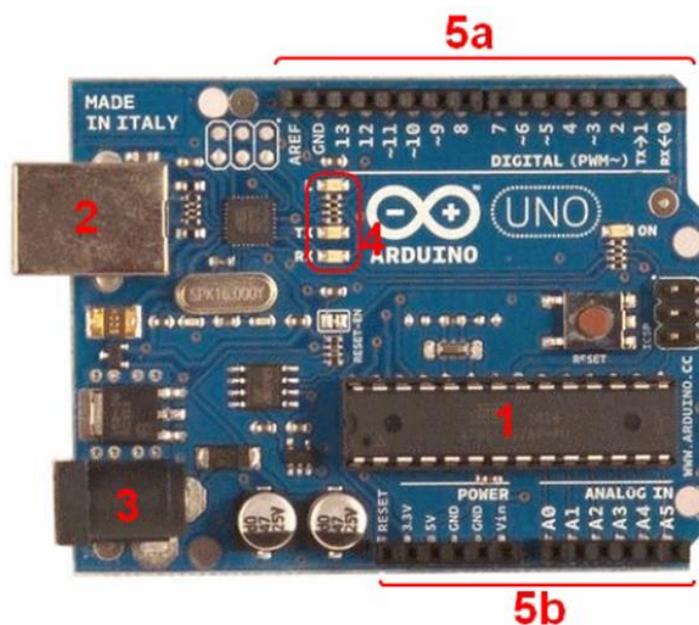


Figure II.2 Presentation de la carte Arduino

II.6 Constitution de la carte

Voyons quels sont ces points importants et à quoi ils servent.

II.6.1 Le microcontrôleur

Le cerveau de notre carte (en 1). C'est lui qui va recevoir le programme que vous aurez créé et qui va le stocker dans sa mémoire puis l'exécuter. Grâce à ce programme, il va savoir

faire des choses, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur, ...

II.6.2 Alimentation

Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB (en 2) ou bien par une alimentation externe (en 3) 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.

II.6.3 Visualisation

Les trois "points blancs" entourés en rouge (4) sont en fait des LED dont la taille est de l'ordre du millimètre. Ces LED servent à deux choses :

Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur et va servir pour tester le matériel.

Nota : Quand on branche la carte au PC, elle clignote quelques secondes.

Les deux LED du bas du cadre : servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

II.6.4 La connectique

La carte Arduino ne possédant pas de composants qui peuvent être utilisés pour un programme, mis à part la LED connectée à la broche 13 du microcontrôleur, il est nécessaire de les rajouter. Mais pour ce faire, il faut les connecter à la carte. C'est là qu'intervient la connectique de la carte (en 5a et 5b).

Par exemple, on veut connecter une LED sur une sortie du microcontrôleur. Il suffit juste de la connecter, avec une résistance en série, à la carte, sur les fiches de connections de la carte.

Cette connectique est importante et a un brochage qu'il faudra respecter. Nous le verrons quand nous apprendrons à faire notre premier programme. C'est avec cette connectique que la carte est "extensible", car l'on peut y brancher tous types de montages et modules ! Par exemple, la carte Arduino Uno peut être étendue avec des shields, comme le « Shield Ethernet » qui permet de connecter cette dernière à internet. [6]



Figure II.3 Une carte Arduino étendue avec un Ethernet shield

II.7 Installation

Afin d'utiliser la carte, il faut l'installer. Normalement, les drivers sont déjà installés sous GNU/Linux. Sous mac, il suffit de double cliquer sur le fichier. mkpg inclus dans le téléchargement de l'application Arduino et l'installation des drivers s'exécute de façon automatique.

II.7.1 Sous Windows

Lorsque on connecte la carte à l'ordinateur sur le port USB, un petit message en bas de l'écran apparaît, théoriquement, la carte doit s'installer toute seule.

Après l'installation et une suite de clignotement sur les micro-LED de la carte, celle-ci devrait être fonctionnelle; une petite LED verte témoigne de la bonne alimentation de la carte :



Figure II.4 Carte connectée et alimentée

II.8 Partie programme

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

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

II.8.1 L'environnement de la programmation

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino. [7]

II.8.2 Structure générale du programme (IDE Arduino)

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

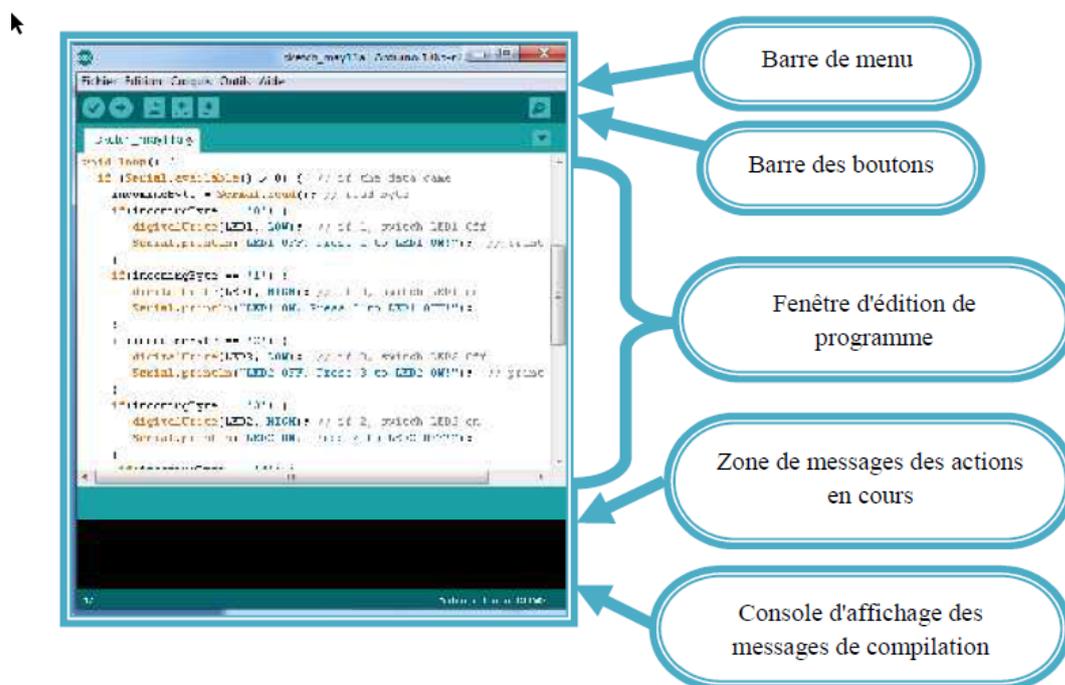


Figure II.5 Interface IDE Arduino

II.8.3 Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM 8) comme à titre d'exemple cette figure suivante.

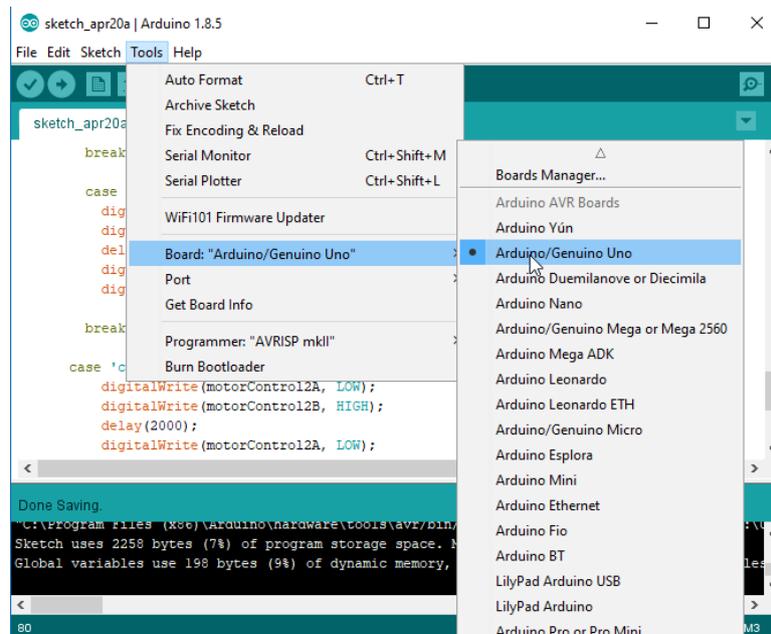


Figure II.6 Paramétrage de la carte étape 1

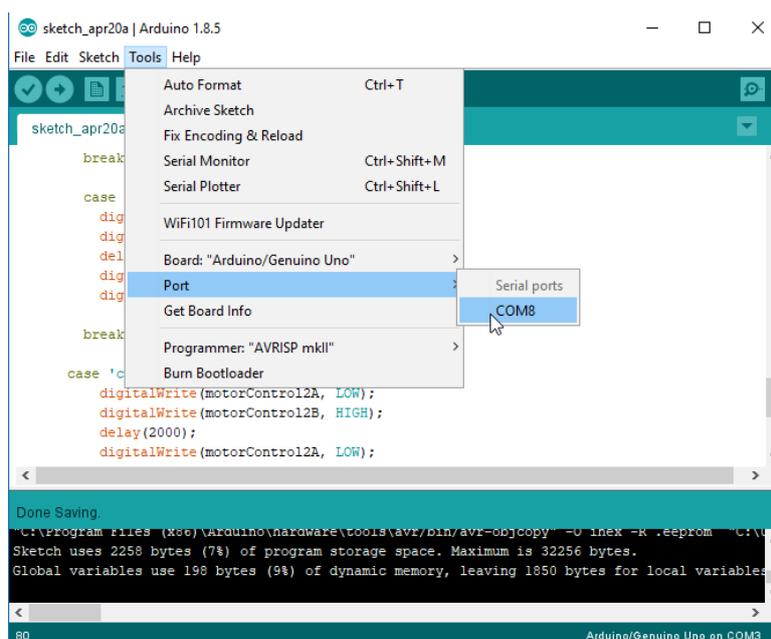


Figure II.7 Paramétrage de la carte étape 2

II.8.4 Description du programme

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

II.8.5 Commentaires

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

Par ex :

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

II.8.6 Définition des variables

Pour notre montage, on va utiliser une sortie numérique de la carte qui est par exemple la 3ème sortie numérique ; cette variable doit être définie et nommée ici moteur pin 3 ; la syntaxe pour désigner un nombre entier est int.

```
4 int moteur 1 = 3; // mettre le moteur au pin 3-----
```

Configuration des entrées et des sorties void setup () les broches numériques de l'arduino peuvent aussi bien être configurées en entrées

numériques ou en sorties numériques; ici on va configurer moteur pin en sortie ; pin mode (nom, état) est une des quatre fonctions relatives aux entrées – sorties numériques.

```
5 void setup() {-----
6 // mettre le moteur 1 comme sortie:-----
7 pinMode(motor 1, OUTPUT); // lorsque le pin 3 est activé le moteur tourne----- 8
}-----
```

II.8.7 Programmation des interactions void loop

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

-delay (temps en mili-seconde) est la commande d'attente entre deux instructions.

-chaque ligne d'instruction est terminée par un point virgule.

-ne pas oublier les accolades qu' encadrent la boucle.

```

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

```

II.8.8 Les étapes de téléchargement du programme

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

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

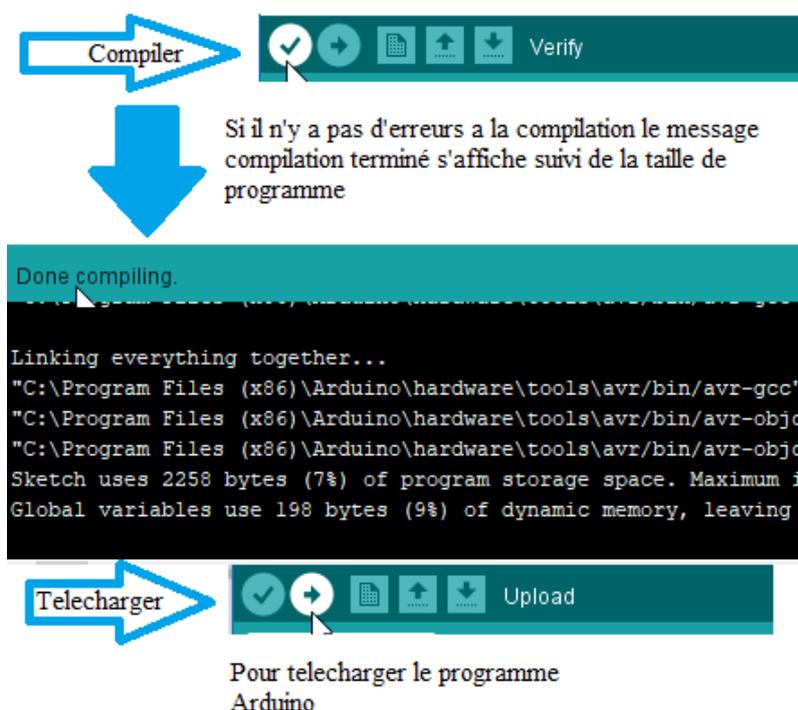


Figure II.8 Les étapes de téléchargement du code

II.9 Les accessoires de la carte Arduino

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

II.9.1 Communication

II.9.1.1 Les modules Bluetooth HC-05 et HC-06

Il existe deux sortes de module Bluetooth, tous deux compatibles Arduino et utilisables sur un breadboard (plaque d'essai en français). On les distingue par le nombre de pattes d'entrées / sorties :

- HC-05 : 6 sorties. Ce module peut être « maître » (il peut proposer à un autre élément Bluetooth de s'appairer avec lui) ou « esclave » (il ne peut que recevoir des demandes d'appairage). Ce module fait l'objet d'un autre article car il y a quelques différences pour le régler.
- HC-06 : 4 sorties. Ce module ne peut être qu'esclave.

Les deux modules peuvent être utilisés en mode COMMANDE, pour les programmer avec des « commandes AT », ou en mode DATA, pour échanger des données. [9]

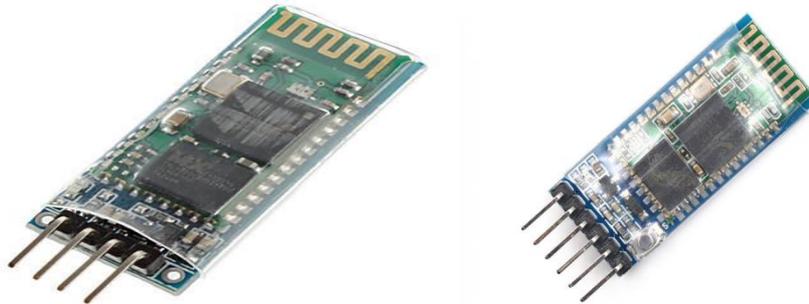


Figure II.9 Type de modules Bluetooth

II.9.1.2 Le module shield Arduino wifi

Le bouclier Arduino Wifi permet à une carte Arduino de se connecter à Internet en utilisant la bibliothèque Wifi et de lire et d'écrire une carte SD en utilisant la bibliothèque SD. [10]

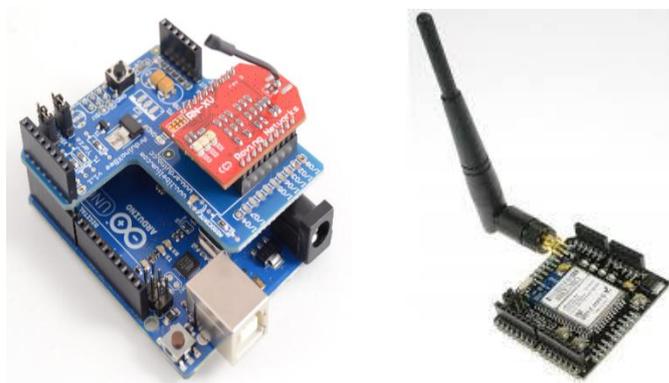


Figure II.10 module shield wifi

II.9.2 Les drives

II.9.2.1 Module de commande puissance L298HN

L'interface de puissance L298 est un module comportant un double pont en H capable de piloter deux moteurs à courant continu ou un moteur pas à pas sous 36 V et consommant 2 A, des diodes de roue libre pour protéger le L298, un régulateur de tension 5V, des condensateurs de filtrage, des borniers à vis pour y brancher les moteurs et des broches pour y

relier la carte Arduino. Le L298 possède trois entrées de commande par pont en H pour piloter le moteur.

Pour faire varier la vitesse du moteur, nous appliquerons un niveau bas sur une des deux entrées IN1 ou IN2 (cela permet de configurer le sens de rotation) et un signal MLI sur l'autre entrée IN1 ou IN2.

Les moteurs sont reliés aux borniers entourés en VERT.

Les alimentations (5 V et VCC) sont reliées au bornier entourés en JAUNE. Les masses de l'alimentation externe (VCC) et celle de la carte Arduino sont communes.

Le cavalier entouré en VERT doit être laissé en place pour activer le régulateur 5 V du module. Ce régulateur fournit le 5 V à la partie logique du L298. Si nous retirons ce cavalier, il faut alimenter le module en 5 V.

Les broches entourées en BLEU sont connectées aux broches numériques de la carte Arduino UNO. [11]

Dans l'univers Arduino, le module interface de puissance L298 est principalement utilisé dans des applications robotiques.

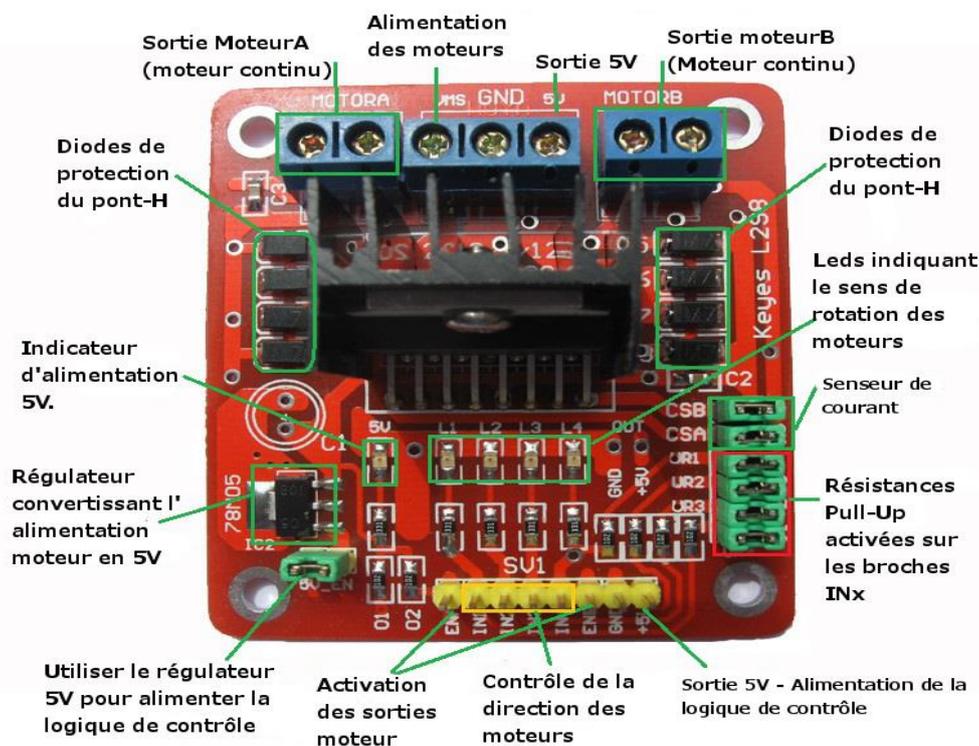


Figure II.11 Les détails techniques du L298N

II.9.3 Les relais

C'est un composant qui possède une bobine (électro-aimant) qui est parcourue par un courant électrique agissant sur un ou plusieurs contacts. Le relais est une solution à la commande en puissance. Il assure en outre une isolation galvanique en mettant en oeuvre un mouvement mécanique. [12]



Figure II.12 relais 12V

II.10 Conclusion

Dans ce chapitre, nous avons donné quelques concepts théoriques, et nous avons expliqué quelques composants principaux du notre montage.

Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; (la partie matérielle et la partie de programmation) plus précisément. Nous avons expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques.

CHAPITRE III

PRINCIPE DE FONCTIONNEMENT ET LA REALISATION VIRTUELLE

III.1 Introduction

Dans ce chapitre, on présentera les différents schémas bloc de notre système, et on va expliquer le principe de fonctionnement de notre système.

Ce travail d'initiation à base d'une carte Arduino UNO permet de commander une maison intelligente (la domotique) via système Bluetooth.

Pour ce faire, nous devons donner le schéma synoptique du dispositif et réaliser le montage virtuellement en utilisant un CAO (Conception assistée par ordinateur) comme ISIS-PROTEUS.

II.2 Les différentes étapes de la réalisation

Notre réalisation pratique a été faite en trois parties:

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

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

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

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

1. Présenter les différentes étapes de la réalisation pratique de la carte.

2. On assemble ensuite les composants suivants notre montage sur le circuit imprimé, commençant par l'alimentation générale de notre dispositif.

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

1. Création de l'interface de notre application Android.
2. Programmation orientée objet de notre application (JAVA).

III.3 Schéma synoptique général

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

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

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

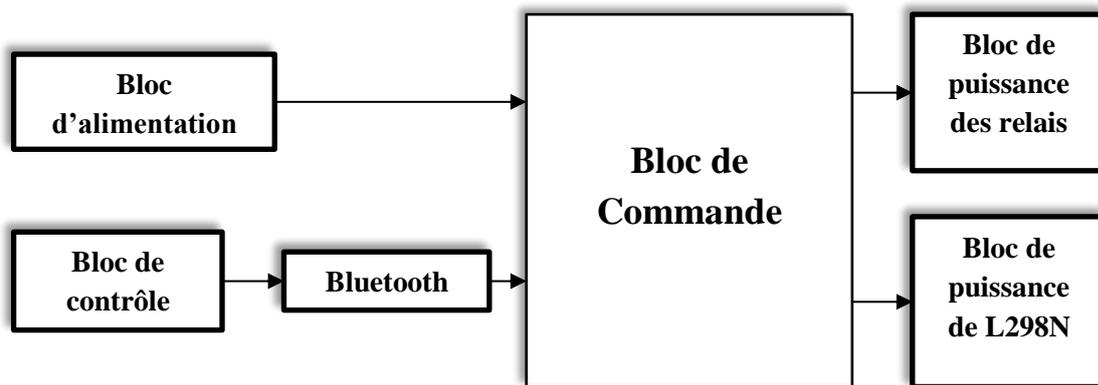


Figure III.1 Schéma synoptique du dispositif

III.3.1 Bloc d'alimentation

La tension issue du secteur transite par le filtre (L, C) qui élimine tout parasite provenant de la ligne 220v alternatif.

Cette tension sera abaissée par le transformateur (220/12v) pour être redresser par le pont de diode qui permet de délivrer une tension ondulée de même signe.

Cette dernière passe à travers le condensateur C2 et C5 éliminant tout parasite résiduels.

La tension contenue obtenue par C1 n'est pas stable de ce fait nous avons inséré un régulateur (LM7812) qui va fournir une tension +12v stable et précise. Cette tension 12v passe à travers un deuxième régulateur (LM7805) qui délivre à sa sortie une tension régulée de +5v.

Ainsi nous obtenons de quoi alimenter la carte Arduino avec +5v et le reste des autres composants (relais, charge, ...) avec un ampérage suffisant.

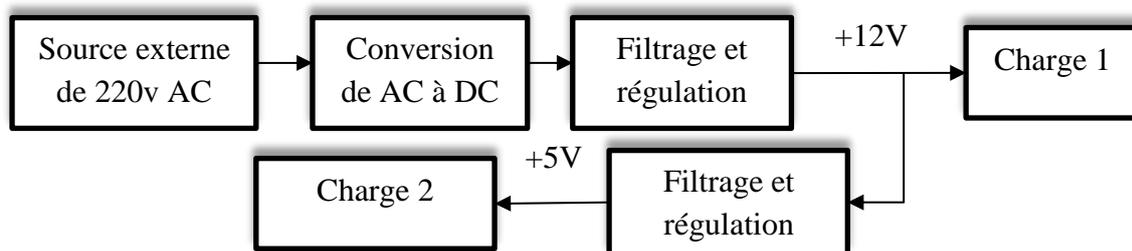


Figure III.2 Schéma synoptique de l'alimentation

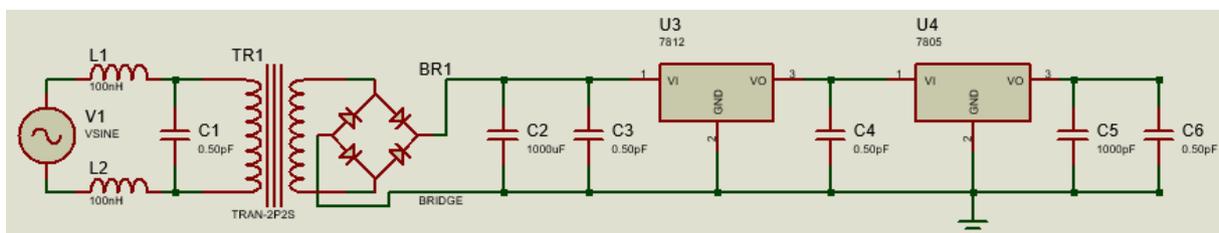


Figure III.3 Schéma électrique de l'alimentation

III.3.2 Bloc de contrôle

Ce bloc est constitué par le smartphone qui contient notre application qui permet de gérer l'ensemble de notre système qui comporte des relais et les deux moteurs.

III.3.3 Bloc de commande

Cette partie commande est assurée entièrement par la carte Arduino.

Dans cette partie l'accessoire Bluetooth est classé comme un périphérique puisqu'il prend le relevé de la validation des signaux émis par le smartphone vers l'Arduino, donc il a besoin d'être configuré lors de la partie programmation.

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

Avant de passer à la configuration du Bluetooth, il faut réaliser quelques étapes et ce comme suit :

- Enlever toutes les communications ;
- Placer la position de fonctionnement **DAT** de commutateur ;
- Relier le port d'USB à l'ordinateur ;
- Télécharger le programme ou ouvrir une terminale série ;
- Employer ce mode pour vérifier ou changer la configuration de Bluetooth : le mot de passe PIN, la vitesse baud, le mode de maître / esclave ;

La configuration d'un Bluetooth module de HC-05, peut être réalisée par l'intermédiaire de la terminale série aux commandes. (On peut utiliser IDE Arduino moniteur de série, sous menu "Tools / Serial Monitor", ou juste, cliquer sur CTRL+M du clavier du PC).

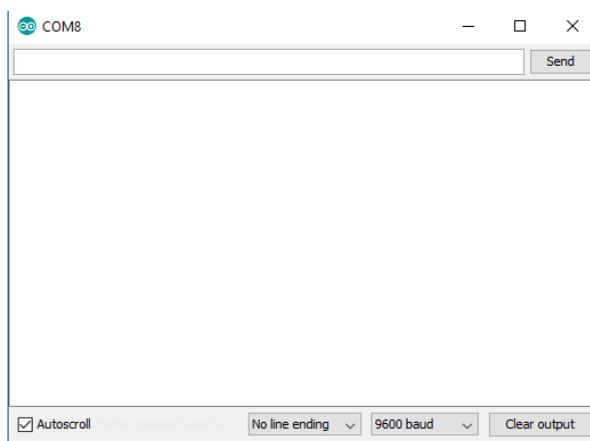


Figure III.4 Schéma simplificateur de la configuration BT

Dans le tableau qui suit, nous allons citer les plus importantes commandes que nous avons utilisé pour configurer le module de Bluetooth HC-06.

Commande	Fonction
AT	Ne fait rien mais la réponse "OK" signifie bon essai pour connaître le mode de CMD.
AT+VERSION	Donne la version de software de module.
AT+NAMExxxxxx	Pour donner le nom du dispositif Bluetooth.
AT+PIN1234	Pour modifier le mot de passe de Bluetooth a 1234.
AT+BAUDx	Placer la vitesse de baud x= 1---- 1200 2---- 2400 3---- 4800 4---- 9600 5---- 19200 6---- 38400 7---- 57600 8---- 115200 9---- 230400 A---- 460800 B---- 921600 C---- 1382400 Défaut: 9600

Tableau III.1 Tableau de commande Bluetooth

III.3.4 Bloc de puissance

Notre bloc de puissance se compose de deux étages :

- 1- Etage des relais
- 2- Etage de driver L298

III.3.4.1 Etage 1

Cet étage se compose de 04 relais électromécaniques qui permettant d'associer la partie puissance et la partie commande chaque relais est excité dans sa bobine par un signal généré par la carte Arduino à travers un Optocoupleur qui assure une parfaite isolation afin de protéger la carte Arduino.

III.3.4.2 Etage 2

Dans cette étage nous avons utilisé un composant robuste à savoir le L298N.

Sa fiche technique est représentée dans (Annexe A).



Figure III.5 Module L298HN (H=Modèle Horizontal)

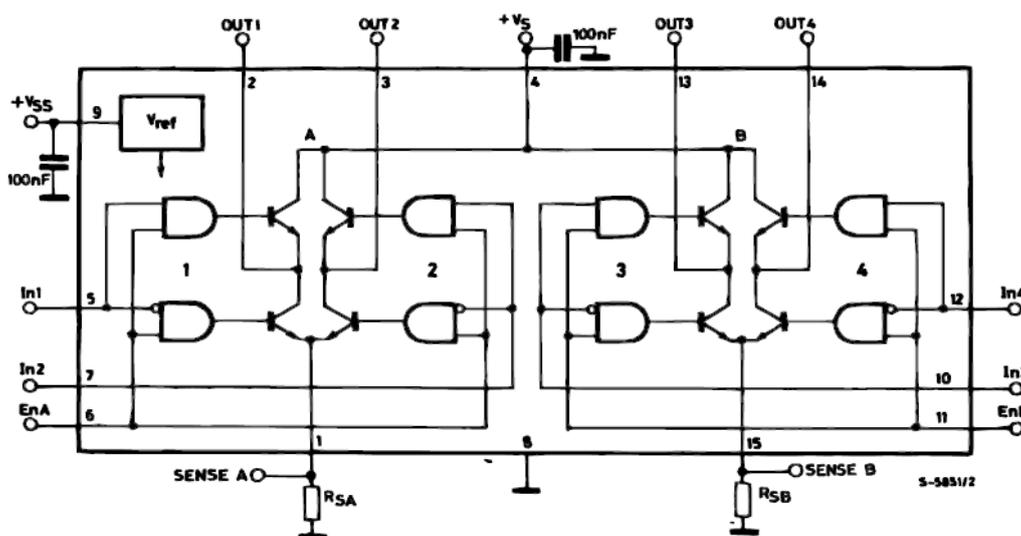


Figure III.6 brochage et structure interne du circuit L298

Les principes de montages du L298N sont rigoureusement identiques à ceux du L293E, ce qui facilite grandement son utilisation seul le brochage diffère, parmi les avantages qui nous ont laissés choisir ce composant :

- L298 supporte une tension de puissance jusqu'à 46 volts maximum ;
- Un courant de service de 2A. supporte les pics occasionnels à 3A et pics répétitifs à 2.5A ;
- Compatible TTL (peut donc être commandé directement avec Arduino) ;
- Dispose d'un dispositif de mesure du courant (sensor/ pins). A raccorder à la masse si on ne l'utilise pas ;
- Chute de tension **V_{ce_sat}** totale de 1.8 volts (typique), 3.2v à IL=1A, 4.9v à IL=2A. Si cela semble beaucoup, c'est aussi un avantage car cela permet d'utiliser un accu de 7.2v directement avec un moteur +5V. La chute de tension **V_{ce_Sat}** du Pont-H sera suffisante pour adapter la tension au moteur (généralement +/- 5V).

Ce composant dispose de 15 broches ; on peut éclaircir son brochage sous forme de tirets comme suit :

- **VSS** - Alimentation de la logique de commande (+5V). A raccorder à la borne +5V d'Arduino (donc sur le régulateur d'Arduino) ;
- **VS** - Alimentation de puissance des moteurs ;
- **GND** - Doit être raccordée à la masse (GND) de la source d'alimentation de puissance VS

(Donc le négatif de l'accumulateur) et à la masse de la source d'alimentation de VSS (donc GND Arduino) ;

- **OUTPUT1, OUTPUT2** - Broches à raccorder à la charge (le moteur). Via ces broches que le L298 commande le sens de rotation du moteur.
- **INPUT1, INPUT2** - Broches de commande du Pont-H. Se raccorde a Arduino.
- **ENABLE A** - (Chip Activer) permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 & OUTPUT2. **ENABLE. A** commande l'activation du premier

Pont-H. Si **ENABLE A = GND**, le pont-H est déconnecté et le moteur ne fonctionne pas. [13]

Si **ENABLE A** = **VSS**, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquées sur **INPUT1** & **INPUT2**.

- **CURRENT SENSING A & CURRENT SENSING B** - permet de faire une mesure du courant dans le circuit de puissance. A placer impérativement sur **GND** si cette fonctionnalité n'est pas utilisée.

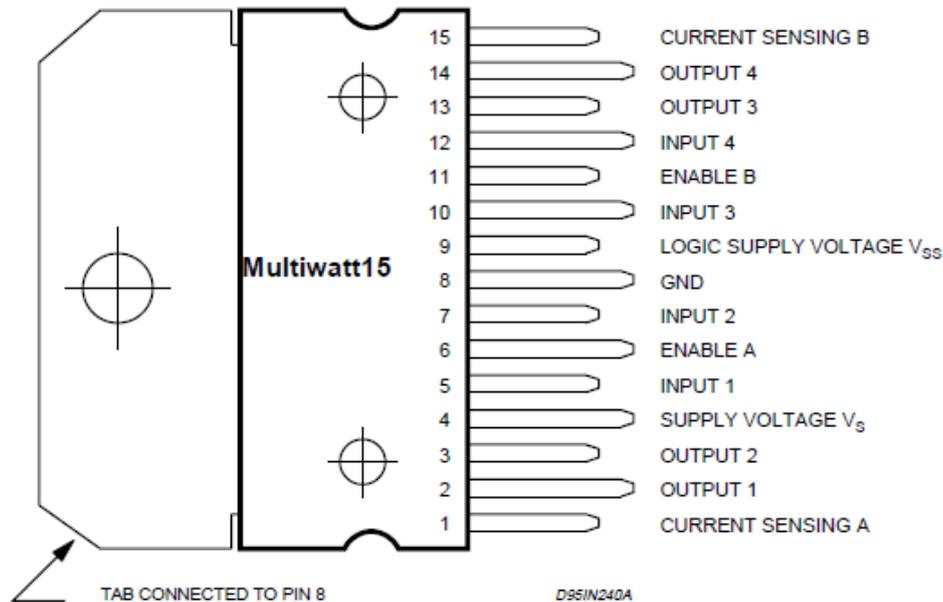


Figure III.7 Schéma de brochage L298

III.4 La réalisation virtuelle (Simulation) :

Avant de passer à la réalisation pratique, nous avons utilisé un CAO: il s'agit de ISISPORTEUS, c'est un CAO (Conception assistée par ordinateur) électronique 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 » est disponible et téléchargeable sur ce lien [14], se compose de nombreux outils regroupés en modules au sein d'une interface unique.

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

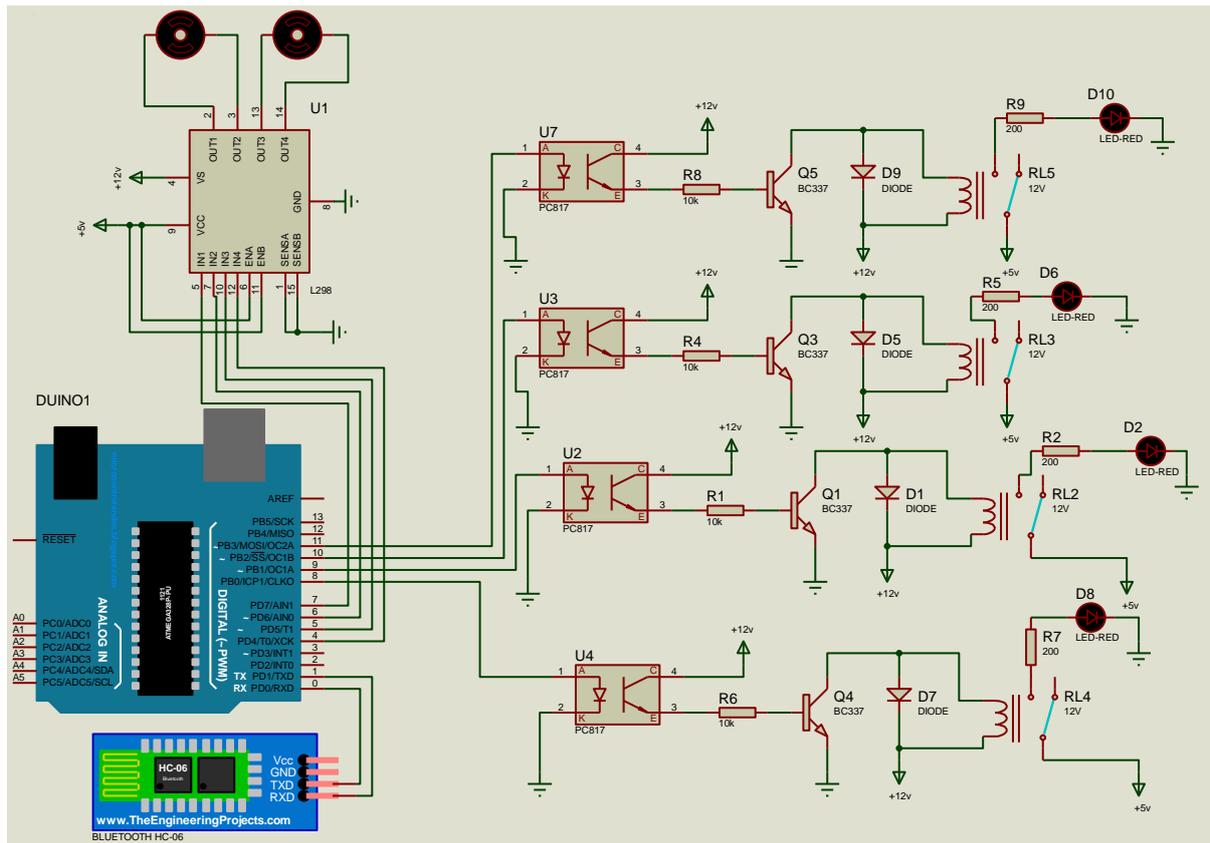


Figure III.8 La carte réalisée sous ISIS-PROTEUS

III.5 Conclusion

Dans ce chapitre, nous avons jeté les bases de notre réalisation par :

- Donner le schéma synoptique général.
- Identifier les paramètres de circuit d'alimentation, circuit de commande et circuit de puissance.
- Faire la réalisation virtuelle de notre carte à l'aide de PROTEUS.

Ce qui nous reste est de faire préparer notre application Android et appliquer tous qu'on a établie dans les chapitres précédents pour fabriquer le circuit imprimé.

CHAPITRE IV

DEVELOPPEMENT DE L'APPLICATION ANDROID

IV.1 Introduction

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

Lors des années précédentes, le traitement des données informatiques se fait par des ordinateurs, en revanche le smart phone a des avantages qui ont les mêmes fonctions que l'outil informatique, ce dernier porte l'intérêt de l'ordinateur grâce à l'Android.

Les innovations n'avaient plus vraiment de saveur, les applications étaient difficiles d'accès de par leur mode de distribution et souvent peu performantes à cause des faibles capacités des appareils.

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

La plate-forme Android apporte tout cela au consommateur, mais surtout, elle affranchit le développeur de nombreuses contraintes. Par son ouverture; elle permet à n'importe quel développeur de créer ses applications avec un ticket d'entrée quasi nul. Le Framework et le système d'exploitation et outils associés ont un code source ouvert, leur accès est gratuit et illimité. Plus besoin de négocier avec le constructeur du téléphone pour qu'il vous laisse développer sur sa plate-forme. Tous les développeurs sont ainsi sur un même pied d'égalité, tous peuvent ajouter de la mobilité à des applications existantes. [15]

Un système d'exploitation orienté dispositif mobiles il s'agit donc d'un ensemble de logiciels qui sert d'interface entre le matériel (les composants du téléphone, d'une tablette...) et les logiciels applicatifs (ceux que vous allez développer).

- Un système d'exploitation open source, disponibilité du code, importante communauté d'utilisateurs.
- Un système d'exploitation basé sur le noyau Linux.
- Environnement de développement gratuit.
- Programmation en Java ou en langage C.
- Kit de développement (SDK Android) disponible au lien [16]

IV.2 L'écosystème d'Android

S'appuie sur deux piliers

- le langage Java
- le SDK qui permet d'avoir un environnement de développement facilitant la tâche du développeur

Le kit de développement donne accès à des exemples, de la documentation mais surtout à l'API de programmation du système et à un émulateur pour tester ses applications.

Stratégiquement, Google utilise la licence Apache pour Android ce qui permet la redistribution du code sous forme libre ou non et d'en faire un usage commercial.

Le SDK était:

- anciennement manipulé par un plugin d'Eclipse (obsolète)
- maintenant intégré à Android Studio (IntelliJ)

IV.3 L'Operating System

Android est en fait un système de la famille des Linux, pour une fois sans les outils GNU. L'OS s'appuie sur:

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

- Un compilateur de bytecode vers le natif Android Runtime (pour Lollipop)
- Des applications (navigateur, gestion des contacts, application de téléphonie...)
- Des bibliothèques (SSL, SQLite, OpenGL ES, etc...)
- Des API d'accès aux services Google

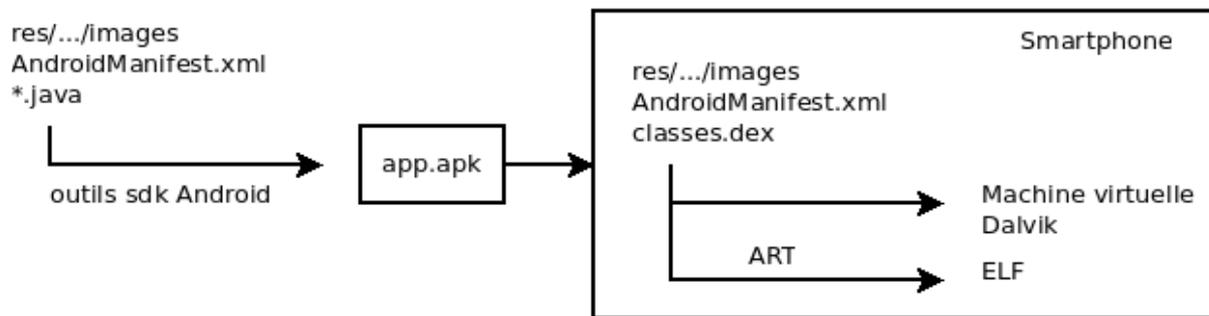


Figure IV.1 Anatomie d'un déploiement

IV.3.1 Dalvik et ART

[Dalvik] est le nom de la machine virtuelle open-source utilisée sur les systèmes Android. Cette machine virtuelle exécute des fichiers.dex, plus ramassés que les .class classiques. Ce format évite par exemple la duplication des String constantes. La machine virtuelle utilise elle-même moins d'espace mémoire et l'adressage des constantes se fait par un pointeur de 32 bits.

[Dalvik] n'est pas compatible avec une JVM du type Java SE ou même Java ME. La librairie d'accès est donc redéfinie entièrement par Google.

A partir de Lollipop, Android dispose d'ART qui compile l'application au moment du déploiement (Ahead-of-time compilation).

IV.4 L'environnement Android Studio :

Depuis mi-2015, il faut utiliser Android Studio (IDE IntelliJ ayant subi l'intégration de fonctionnalités de développement Android).

IV.4.1 Avantages

- meilleure intégration du SDK dans Android Studio
- puissance de l'IDE IntelliJ.

IV.4.2 Désavantages

- lourdeur de l'IDE IntelliJ.
- moins d'outils standalone (gestion des émulateurs, du SDK).
- nouvelle architecture des répertoires.
- meilleure gestion des dépendances avec gradle.

L'architecture d'un projet Android Studio

- app: le code de votre application
- build: le code compilé
- lib: les bibliothèques natives
- src: les sources de vos applications
- main/java: vos classes
- main/res: vos ressources (XML, images, ...)
- test: les tests unitaires
- Des fichiers de configuration:
- build.gradle (2 instances): règles de dépendance et de compilation
- settings.gradle: liste de toutes les applications à compiler (si plusieurs)

IV.5 Les éléments d'une application

Une application Android peut être composée des éléments suivants:

- des activités (android.app.Activity): il s'agit d'une partie de l'application présentant une vue à l'utilisateur
- des services (android.app.Service): il s'agit d'une activité tâche de fond sans vue associée

- des fournisseurs de contenus (`android.content.ContentProvider`): permet le partage d'informations au sein ou entre applications
- des widgets (`android.appwidget.`): une vue accrochée au Bureau d'Android
- des Intents (`android.content.Intent`): permet d'envoyer un message pour un composant externe sans le nommer explicitement
- des récepteurs d'Intents (`android.content.BroadcastReceiver`): permet de déclarer être capable de répondre à des Intents
- des notifications (`android.app.Notifications`): permet de notifier l'utilisateur de la survenue d'événements.[17]

IV.6 Langage java

Java est un langage de programmation très populaire développé par Sun Microsystems (maintenant détenu par Oracle). Développé longtemps après C et C ++, Java incorpore de nombreuses fonctionnalités puissantes de ces langages puissants tout en corrigeant certains de leurs inconvénients. Pourtant, les langages de programmation sont seulement aussi puissants que leurs bibliothèques. Ces bibliothèques existent pour aider les développeurs à créer des applications.

IV.7 Développement de l'application :

IV.7.1 Partie graphique

- Explication des paramètres du projet :
- Application name : Le nom de votre application.
- Company domain : Domaine de votre entreprise.
- Project location : Location de projet dans votre disque dur.
- Package name : Le nom da package Android.

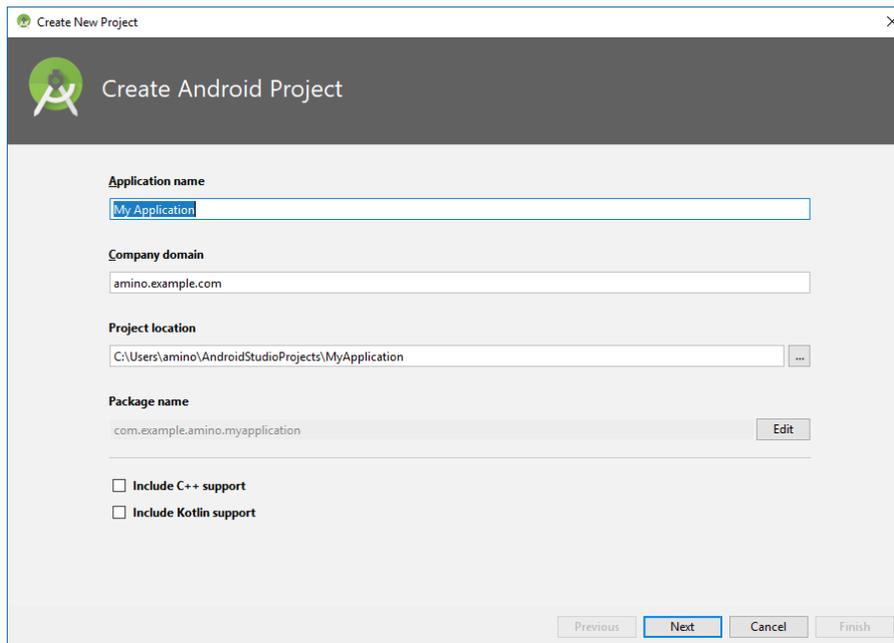


Figure IV.2 premier fenêtre pour créer une application sur Android Studio

Ensuite on vas sélectionner la version API qu'on va l'utiliser :

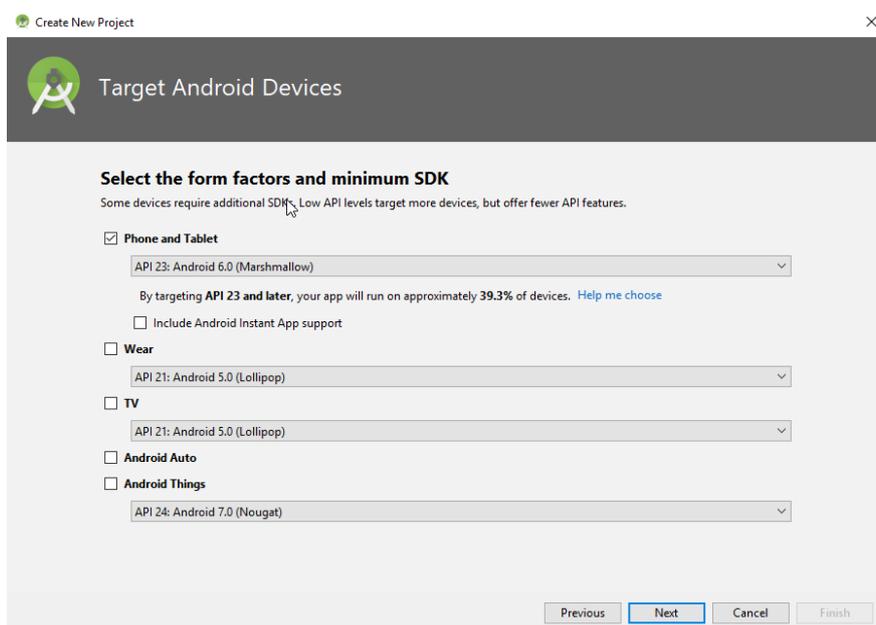


Figure IV.3 Fenêtre pour sélectionner la version API de notre application

Après on vas choisir l'Activité désirable :

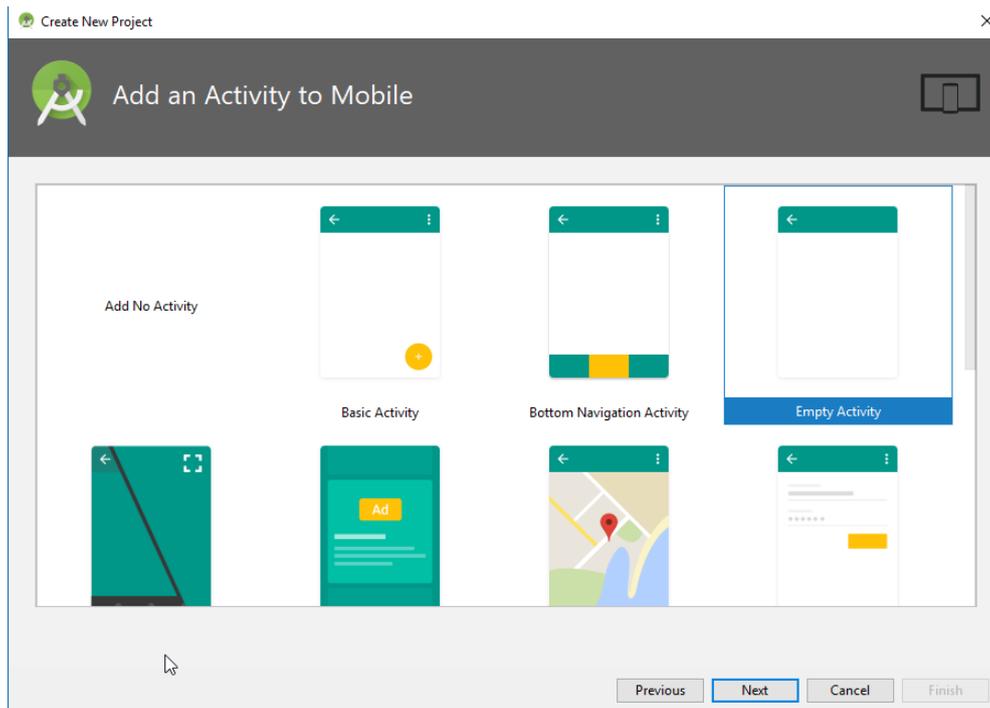


Figure IV.4 Fenêtre pour choisir l'Activité

Après on choisit le nom de l'Activité on est présentée par la fenêtre principale de notre programme :

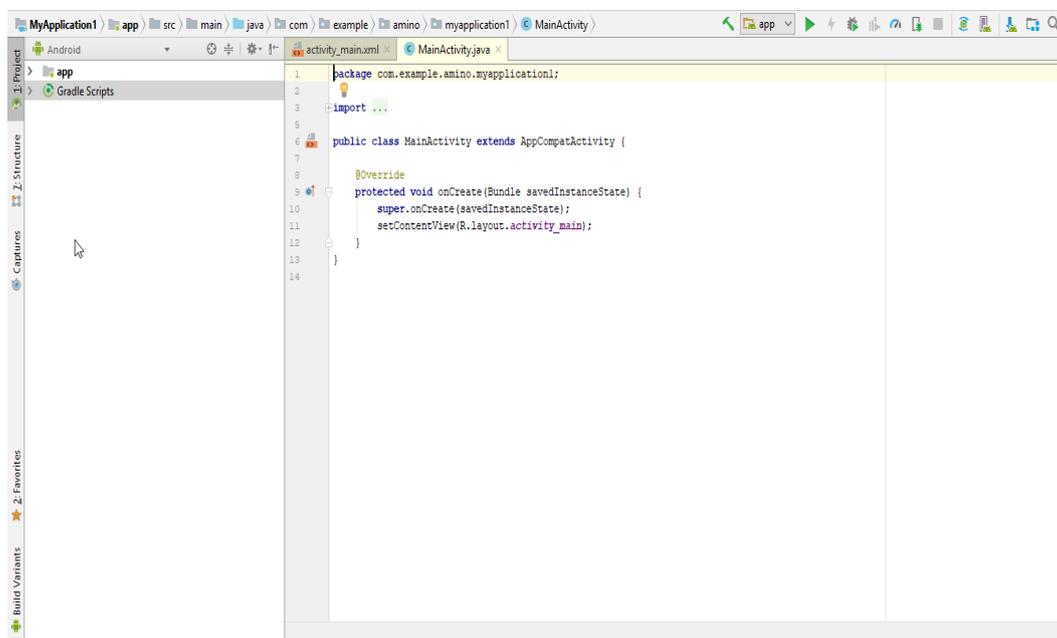


Figure IV.5 Vue principale d'Android Studio

IV.7.1.1 Design de l'application

Pour Designer notre application on a 2 options, par l'utilisation de l'interface graphique ou avec code XML.

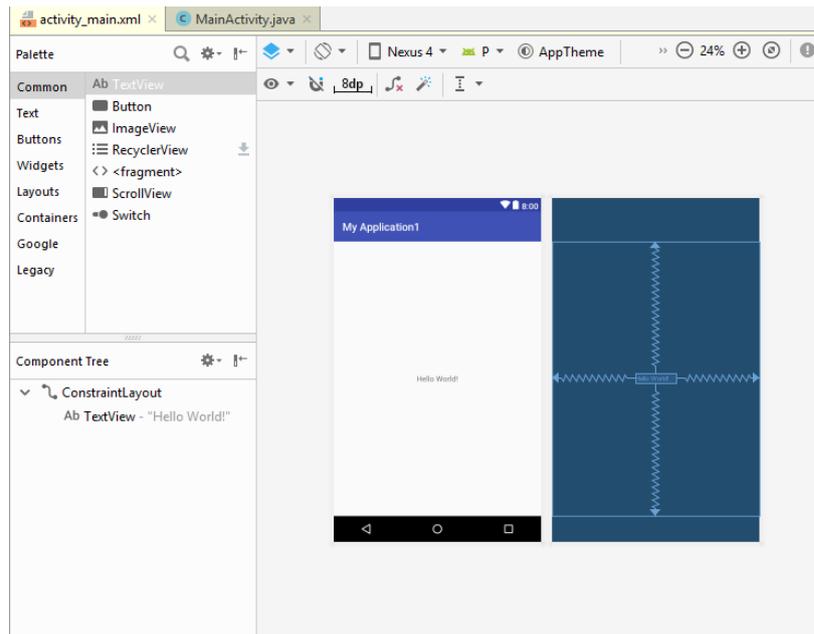


Figure IV.6 L'interface graphique de l'exemple 'Hello World'

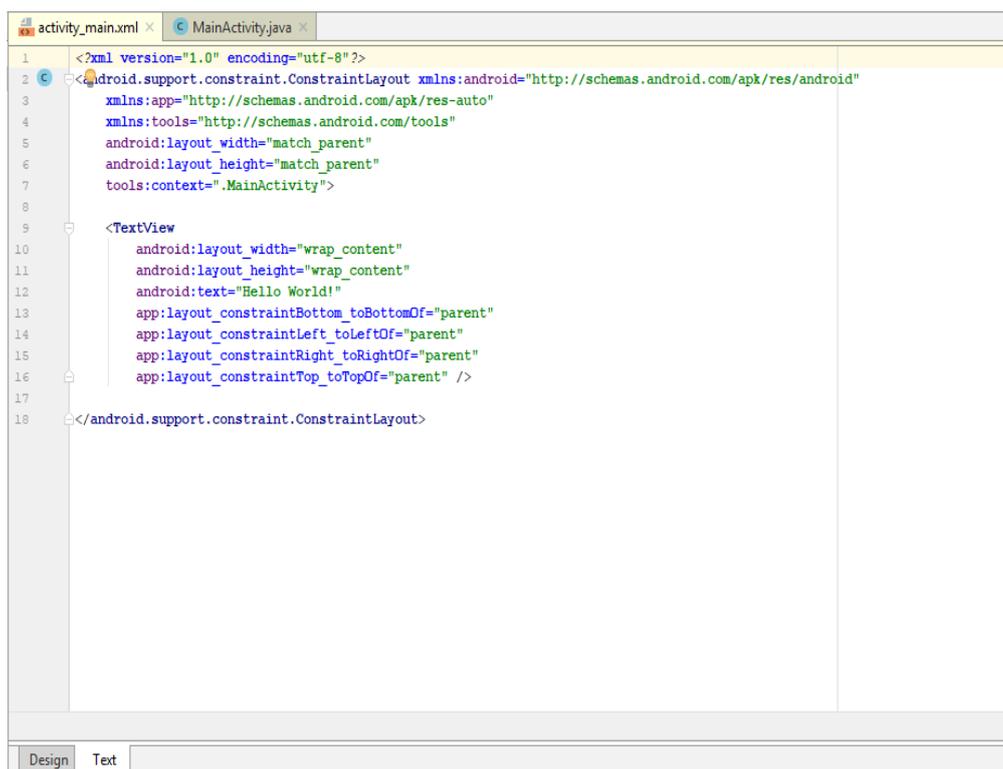


Figure IV.7 Le code XML de l'exemple 'Hello World'

Pour la personnalisation de notre application il y a plusieurs outils et composants qui facilitent le travail pour nous.

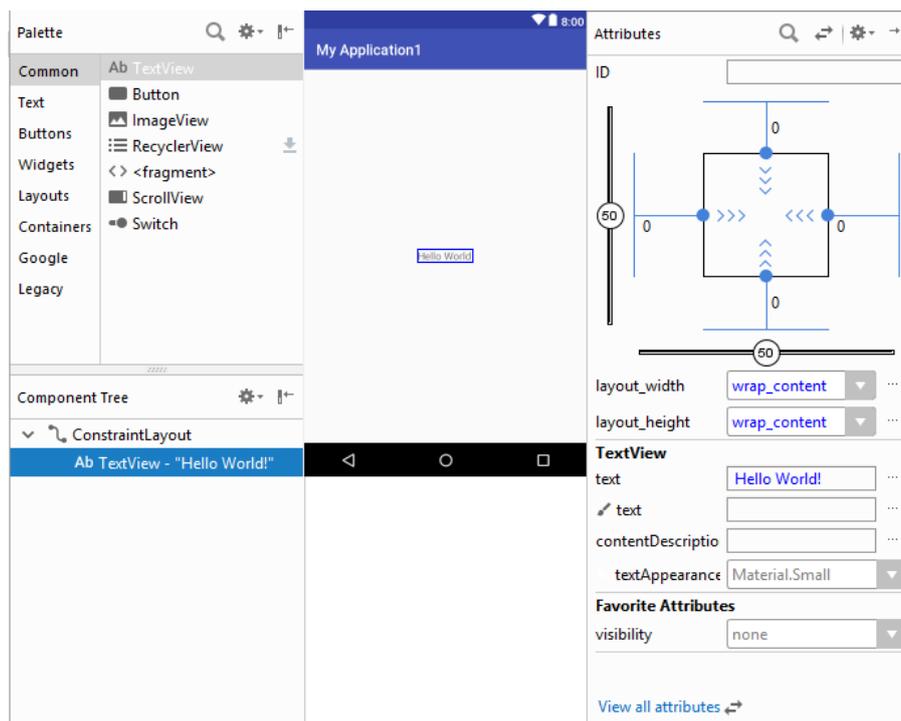


Figure IV.8 Les outils à utiliser dans l'interface graphique

Après l'assemblage des différents composants qui constituent notre application, on peut commencer la partie programmation de notre application.

IV.7.2 Partie programmation

Une fois les composants de l'écran de téléphone mis en place et désigné, nous passons à la deuxième phase de développement d'une application : la programmation JAVA.

Avant de commencer on va donner les permissions Bluetooth à notre application pour utiliser le module Bluetooth du Smartphone, pour faire ça il faut ajouter ces 2 lignes au fichier `AndroidManifest.XML` :

```
<uses-feature android:name="android.hardware.bluetooth" />
<uses-permission android:name="android.permission.BLUETOOTH" />
```

Après on va commencer à programmer notre application, pour communiquer avec le module Bluetooth de l'Arduino on va utiliser « Bluetooth chat sample » du site web Android Developers [18], cet échantillon est utilisé pour une application de chat entre deux smartphones mais on peut l'utiliser pour notre application pour envoyer des commandes à l'Arduino, les méthodes qu'on va l'utiliser avec notre application sont :

- `getDefaultAdapter()` : Cette méthode vérifie si le smartphone a de Bluetooth.

- `bondedDevices.isEmpty()`: Cette méthode vérifie si le smartphone est associé avec le module Bluetooth.
- `socket.connect()` : Utiliser pour connecter avec le module Bluetooth.
- `outputStream.write(command.getBytes())`: La méthode utiliser pour envoyer les commandes à l'Arduino.

Maintenant il faut associer les boutons avec leurs fonctionnalités dans le programme, pour envoyer les commandes on a utilisé des boucles « If » est la méthode : `outputStream.write(command.getBytes())` pour envoyer les commandes via le Bluetooth par exemple :

```
//OnCheckedChangeListener code pour le bouton LED n1
LED1_btn.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            command = "3";
            try {
                outputStream.write(command.getBytes()); //transmis la valeur de au module
bluetooth.
            } catch (IOException e) {
                e.printStackTrace();
            }
        } else {
            command = "4";
            try {
                outputStream.write(command.getBytes()); //transmis la valeur de au module
bluetooth.
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
});
```

IV.8 Présentation du programme IDE

IV.8.1 Fichier JAVA

```
//Declaration des boutons
Button gauche_btn, droite_btn, bluetooth_btn, ouvrir_btn, fermer_btn;
ToggleButton LED1_btn, LED2_btn, LED3_btn, LED4_btn;
String command; //variable string ou on va stoker les caractères qu'on va envoyer via le
Bluetooth a l'Arduino
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

```
    outputStream = new ByteArrayOutputStream(1024) ;
```

```
//L'attachement des boutons avec leurs boutons dans l'interface graphique
```

```
gauche_btn = findViewById(R.id.gauche_btn);
droite_btn = findViewById(R.id.droite_btn);
bluetooth_btn = findViewById(R.id.bluetooth_btn);
LED1_btn= findViewById(R.id.LED1_btn);
ouvrir_btn = findViewById(R.id.ouvrir_btn);
fermer_btn = findViewById(R.id.fermer_btn);
LED2_btn= findViewById(R.id.LED2_btn);
LED3_btn= findViewById(R.id.LED3_btn);
LED4_btn= findViewById(R.id.LED4_btn);
```

```
//OnClickListener pour le bouton ouvrir (button click)
```

```
gauche_btn.setOnClickListener(new View.OnClickListener() {
    @Override
```

```

public void onClick(View v) {
    command = "1";
    try {
        outputStream.write(command.getBytes()); //transmits the value of command to the
bluetooth module
    } catch (IOException e) {
        e.printStackTrace();
    }
}
});

//OnClickListener code pour le bouton BT connect (button click)
bluetooth_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(BTinit())
        {
            BTconnect();
        }
    }
});
}

```

IV.8.3 Explication de programme

Notre interface est utilisée pour contrôler une maison et contient 3 cartes pour séparer le garage et les deux chambres, il y a un bouton pour connecter au Bluetooth e,4 boutons (2 pour le rideau de garage et 2 pour les rideaux de fenêtre de chambre 1), et 4 bouton pour les différentes lampes dans la maison.

Cette figure montre que notre application nommée « Contrôle Maison » peut être installée sur smart phone.



Figure IV.9 L'interface finale de notre application

IV.9 Conclusion

Une simple élaboration de l'IDE Android Studio nous montre que c'est un programme un peu avancé et coûte de temps à l'apprendre, mais c'est un IDE très puissant qui permet trop de personnalisation et de contrôle sur l'application, on peut conclure que les résultats sont satisfaisants.

CHAPITRE V

REALISATION DU DISPOSITIF EXPERIMENTAL

V.1 Introduction

Dans ce chapitre, on présentera le programme de l'Arduino qui pour contrôler notre carte, une vue d'ensemble du dispositif expérimental et la réalisation d'une carte électronique contrôlée par un Smart phone et commandée par l'Arduino.

V.2 Présentation de l'organigramme IDE

L'organigramme est une représentation schématique des liens fonctionnels, organisationnels et hiérarchiques d'un organisme, d'un programme, etc...

Les organigrammes sont des schémas qui représentent un processus, un système ou un algorithme informatique. Ils sont largement utilisés dans de multiples domaines pour documenter, étudier, planifier, améliorer et faire partager des processus souvent complexes ainsi transposés dans des schémas clairs et faciles à comprendre. Les organigrammes utilisent des rectangles, des ellipses, des losanges et potentiellement de nombreuses autres formes pour définir le type d'étape, ainsi que des flèches de connexion pour définir le flux et la séquence. Ils peuvent aller de simples graphiques dessinés à la main à des schémas complets dessinés par ordinateur, qui représentent plusieurs étapes et parcours. Si l'on considère la diversité des formes d'organigramme, il s'agit de l'un des types de schémas les plus répandus sur la planète. Ils sont utilisés dans de nombreux domaines par des personnes techniques ou non techniques. [19]

V.2.1 Organigramme Arduino UNO

Avant de passer à la programmation, on va réaliser un organigramme qui explique les différentes séquences de notre programme, il comportera plusieurs boucles dont la fin d'exécution succède toujours à son commencement.

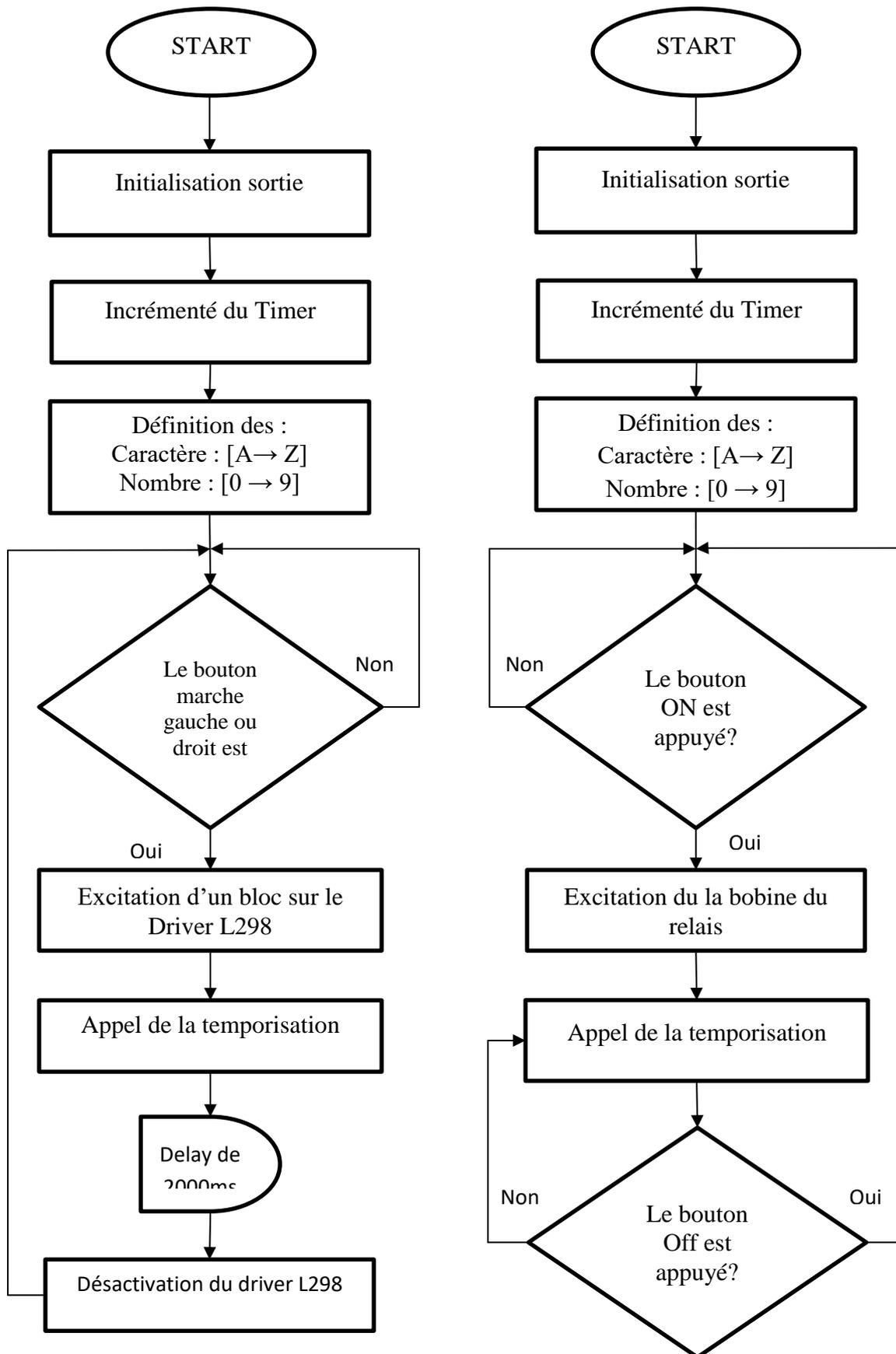


Figure V.1 Organigramme de notre programme

V.3 Présentation du programme IDE

```
int motorControlA = 6;
int motorControlB = 7;
int motorControl2A = 5;
int motorControl2B = 4;
int Allume_LED1 = 8;
int Allume_LED2 = 9;
int Allume_LED3 = 10;
int Allume_LED4 = 11;

void setup() {
  Serial.begin(9600);
  Serial.println(">> START<<");

  pinMode(motorControlA, OUTPUT);
  pinMode(motorControlB, OUTPUT);
  pinMode(motorControl2A, OUTPUT);
  pinMode(motorControl2B, OUTPUT);
  pinMode(Allume_LED1, OUTPUT);
  pinMode(Allume_LED2, OUTPUT);
  pinMode(Allume_LED3, OUTPUT);
  pinMode(Allume_LED4, OUTPUT);
}

void loop() {
  if(Serial.available() > 0)
  {
    char data;
    data = Serial.read();
    Serial.write(Serial.read());

    switch (data)
```

```
{  
  
case '1': //Droite  
    digitalWrite(motorControlA, HIGH);  
    digitalWrite(motorControlB, LOW);  
    delay(2000);  
    digitalWrite(motorControlA, LOW);  
    digitalWrite(motorControlB, LOW);  
  
    break;  
  
case '2': //Gauche  
    digitalWrite(motorControlA, LOW);  
    digitalWrite(motorControlB, HIGH);  
    delay(2000);  
    digitalWrite(motorControlA, LOW);  
    digitalWrite(motorControlB, LOW);  
  
    break;  
  
case '3': //LED  
    digitalWrite(Allume_LED1, HIGH);  
  
    break;  
  
case '4': //LED  
    digitalWrite(Allume_LED1, LOW);  
  
    break;  
  
case '5': //LED  
    digitalWrite(Allume_LED2, HIGH);  
  
    break;
```

```
case '6': //LED
    digitalWrite(Allume_LED2, LOW);

break;

case '7': //LED
    digitalWrite(Allume_LED3, HIGH);

break;

case '8': //LED
    digitalWrite(Allume_LED3, LOW);

break;
case '9': //LED
    digitalWrite(Allume_LED4, HIGH);

break;

case 'a': //LED
    digitalWrite(Allume_LED4, LOW);

break;

case 'b': //Droite
    digitalWrite(motorControl2A, HIGH);
    digitalWrite(motorControl2B, LOW);
    delay(2000);
    digitalWrite(motorControl2A, LOW);
    digitalWrite(motorControl2B, LOW);

break;
```

```
case 'c': //Gauche
    digitalWrite(motorControl2A, LOW);
    digitalWrite(motorControl2B, HIGH);
    delay(2000);
    digitalWrite(motorControl2A, LOW);
    digitalWrite(motorControl2B, LOW);

    break;

default: //If bluetooth module receives any value not listed above, the motor turns off
    digitalWrite(motorControlA, LOW);
    digitalWrite(motorControlB, LOW);
    digitalWrite(Allume_LED1, LOW);

}
}
}
```

V.3.1 Explication de programme

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

Chaque caractère active ou désactive une bobine des relais sur la carte réalisée ou faire tourner un de deux moteurs gauche ou droite à l'aide du driver L298N.

Chaque caractère est codé par des déclarations décimales sur les lignes d'entrée du microcontrôleur.

On commence par la déclaration toutes les variables utilisées après on initialise les ports du l'Arduino comme des sorties « OUTPUTS », ensuite ; on déclare une boucles Switch pour exécuter l'activation ou la désactivation des commandes.

V.4 La réalisation du circuit imprimé

Il s'agit de créer les liaisons entre les différents composants utilisés « résistances, relais, Arduino, transistors, photo-coupleurs etc.... », avec des pistes en cuivre sur le circuit imprimé. De ce fait, on utilise le logiciel nommé « EAGLE » disponible et téléchargeable sur ce lien.

[20]

Pour réaliser la carte électronique, nous allons passer par plusieurs étapes.

EAGLE nous permet de dessiner des schémas de haute qualité avec la possibilité de contrôler parfaitement l'apparence du dessin tout au long du processus d'édition: largeurs de lignes, styles de remplissage, couleurs et polices, etc.

Cette photo nous montre le circuit imprimé :

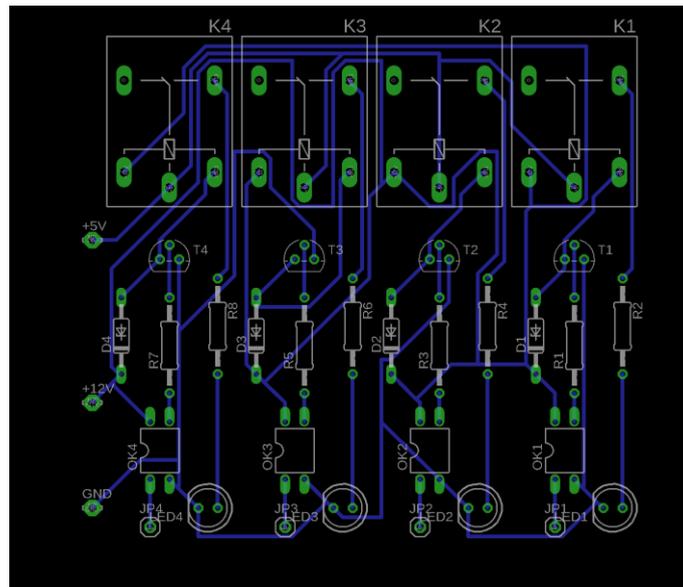


Figure V.2 Typon de circuit imprimé avec composants

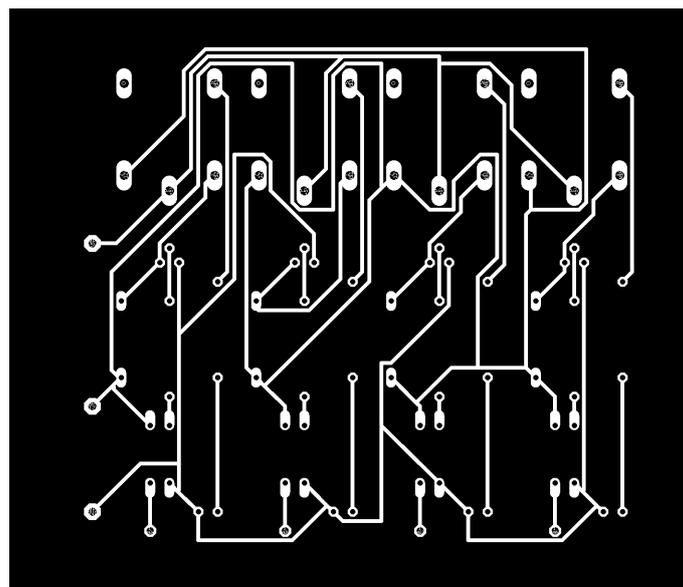


Figure V.3 Typon de circuit imprimé négatif

V.5 Composants utilisés

Pour la réalisation, nous avons assemblé ces différents composants:

- Une carte Arduino UNO.
- Un module Bluetooth HC-06.
- Un drive moteur L289N.
- 4 photo-coupleurs pc817.
- 8 résistances (10Kohms, 200ohms).
- 4 transistors BC337.
- 4 diodes pour filtrage.
- 4 relais 12V.
- 2 moteurs DC

La photo globale de notre réalisation est mentionnée sur annexe (B).

V.6 Conclusion

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

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

- La programmation de la carte Arduino UNO.
- La conception des composants électroniques qui compose la carte réalisée à l'aide logiciel (ISIS- PROTUSE), puis dessine le typon de notre carte à l'aide logiciel EAGLE.
- Le choix des composants convenables pour chaque bloc.

CONCLUSION GENERALE

Dans le cadre de ce projet, nous nous sommes intéressées à la réalisation d'un système électronique dédié au contrôle intelligent d'un domicile. Au cours de la période d'élaboration du projet, nous avons essayé d'atteindre l'objectif attendu :

La réalisation d'une interface Android sous forme d'application sur smartphone qui a constitué la première partie de notre travail.

Le second volet de notre mémoire a porté sur le côté traitement de tous les fonctions nécessaires qui permettent le contrôle et l'activation de l'ensemble des organes actionneurs par l'intermédiaire de la carte Arduino.

Le système Arduino est un outil pour fabriquer des dispositifs qui peuvent capter et contrôler davantage de choses du monde matériel que votre ordinateur.

C'est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

La programmation du microcontrôleur AVR exige une connaissance en langage « C » comme soft et la maîtrise des outils « IDE », afin d'arriver à l'objectif souhaité.

Dans ce travail, un assemblage de plusieurs logiciels est fait afin de réaliser un tel résultat voir annexe (C).

Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes confrontés à plusieurs problèmes surtout dans la partie réalisation de la carte.

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.

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.

BIBLIOGRAPHIE

- [1] <http://www.sirlan.com/livreblanc.pdf>
- [2] http://igm.univ-mlv.fr/~dr/XPOSE2007/aessaidi-ndiop_LA-DOMOTIQUE/intro.htm
- [3] <https://thomasdebenedetti.wordpress.com/2014/05/07/les-techniques-utilisees/>
- [4] Mecheri Salah-eddine. « Réalisation d'un onduleur de tension monophasé contrôlé par une carte Arduino » mémoire master, Université Constantine I
- [5] Boulares Ahmed. « Étude de la commande et simulation des circuits d'un pendule inversé » mémoire de master, UNIVERSITE M'HAMED BOUGARA-BOUMERDES
- [6] <https://eskimon.fr/tuto-arduino-104-le-mat%C3%A9riel>
- [7] Jean- Noël, « livret Arduino en français », centre de ressources art sensitif
- [8] krama abdelbasset, gougui abdelmoumen. « Étude et réalisation d'une carte de contrôle par arduino via le système Android » mémoire de master, université kasdi merbah ouargla, 2015
- [9] <https://knowledge.parcours-performance.com/arduino-bluetooth-hc-05-hc-06/>
- [10] <https://www.arduino.cc/en/Guide/ArduinoWiFiShield>
- [11] <http://pecquery.wixsite.com/arduino-passion/copie-de-le-detecteur-a-ultrasons-h>
- [12] A. Grimault, J. Querard « Article Procédé et dispositif de commutation d'un relais électromagnétique ». EP2312598 A1.
- [13] krama abdelbasset, gougui abdelmoumen. « étude et réalisation d'une carte de contrôle par arduino via le système Android » mémoire de master, université kasdi merbah ouargla, 2015
- [14] <https://www.labcenter.com/>
- [15] https://www.eyrolles.com/Chapitres/9782212125870/AP_Guignard.pdf

[16] <http://developer.android.com/sdk/index.html>

[17] Jean-Francois Lalande « Développement Android », November 2017 - Version 2.6

[18] <https://github.com/googlesamples/android-BluetoothChat>

[19] <https://www.lucidchart.com/pages/fr/organigrammes-diagrammes>

[20] <https://www.autodesk.com/products/eagle/overview>

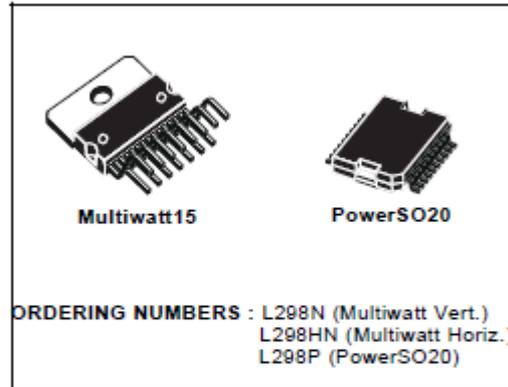
ANNEXE A

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

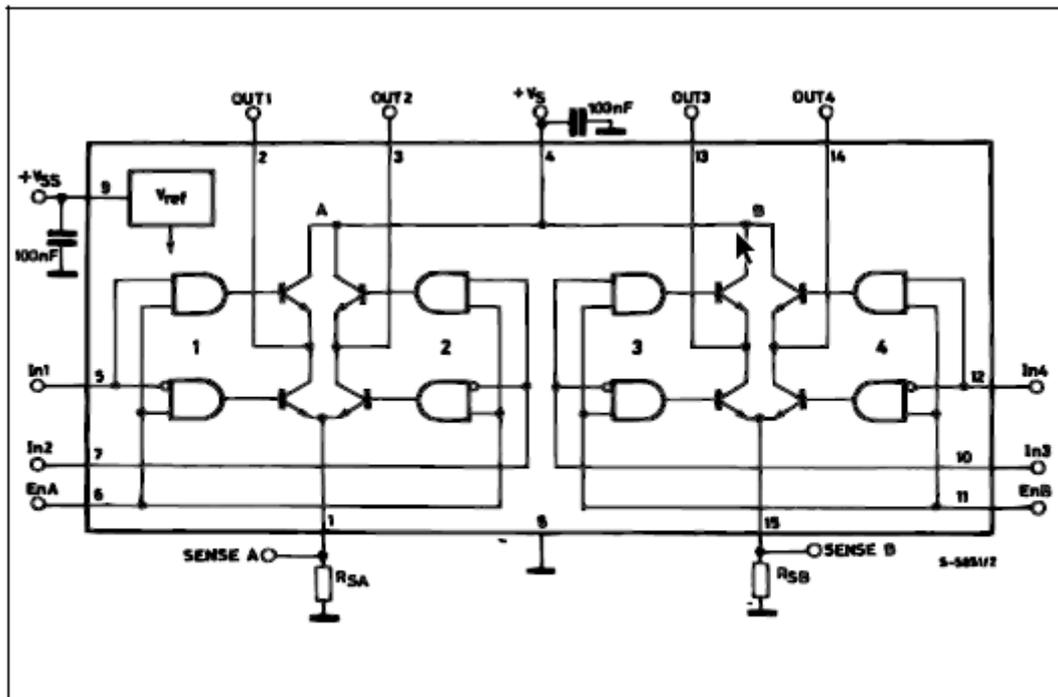
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM



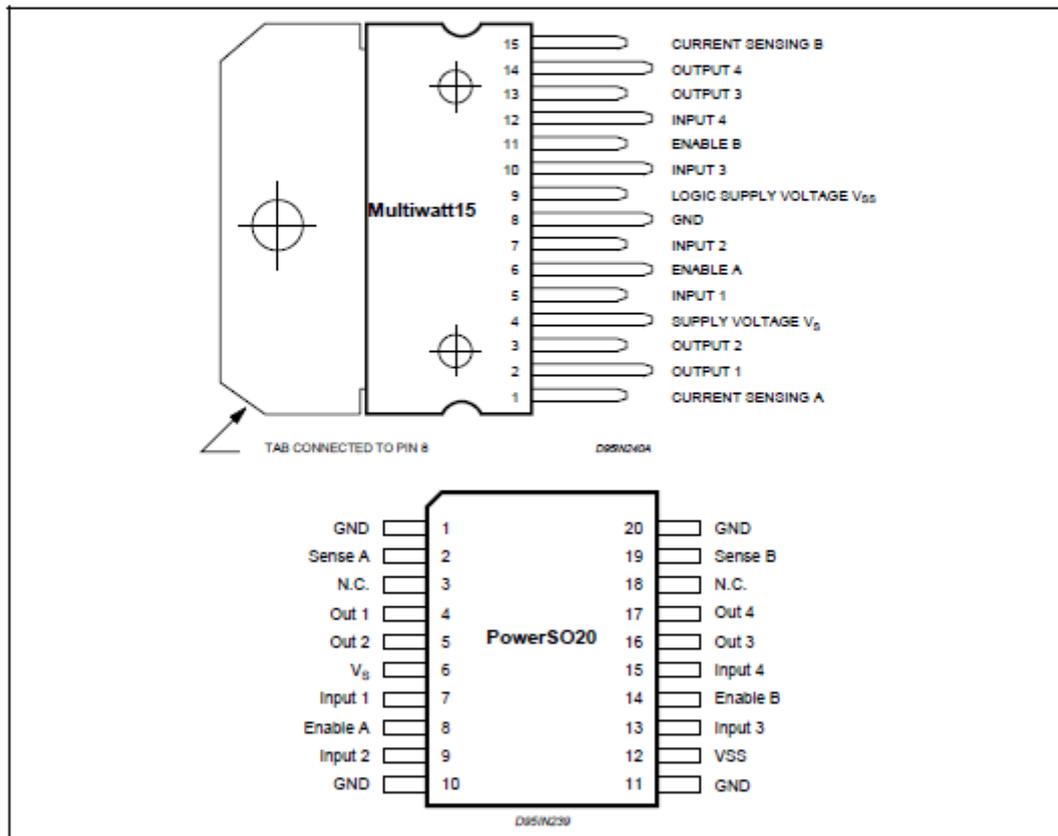
DATASHEET de L298(1)

L298

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_J	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



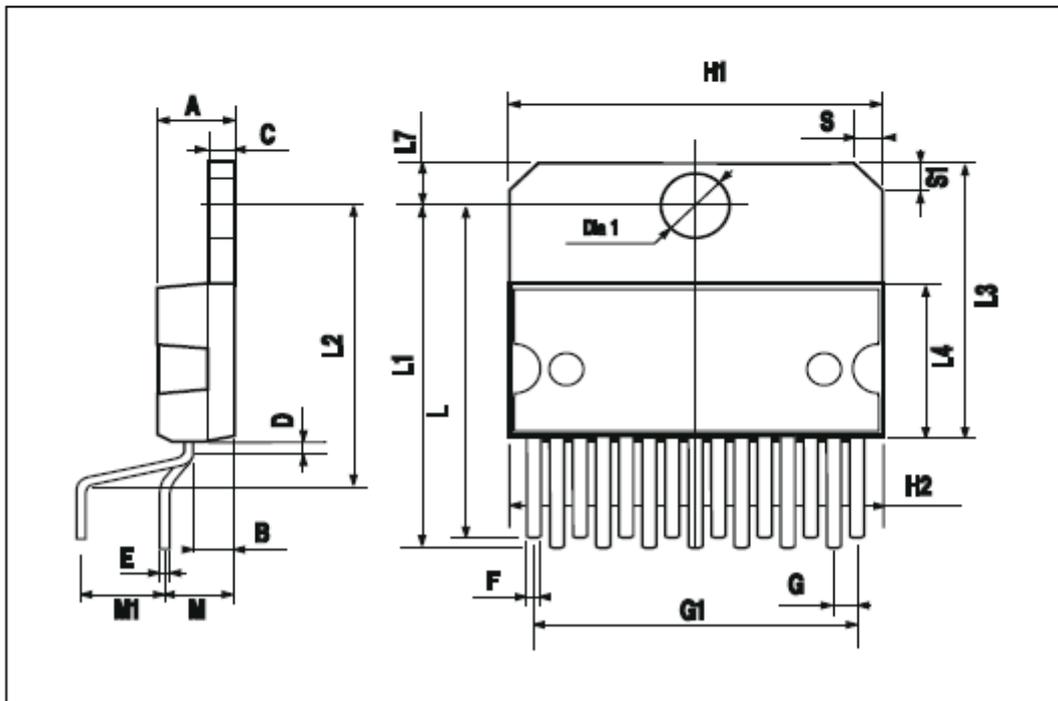
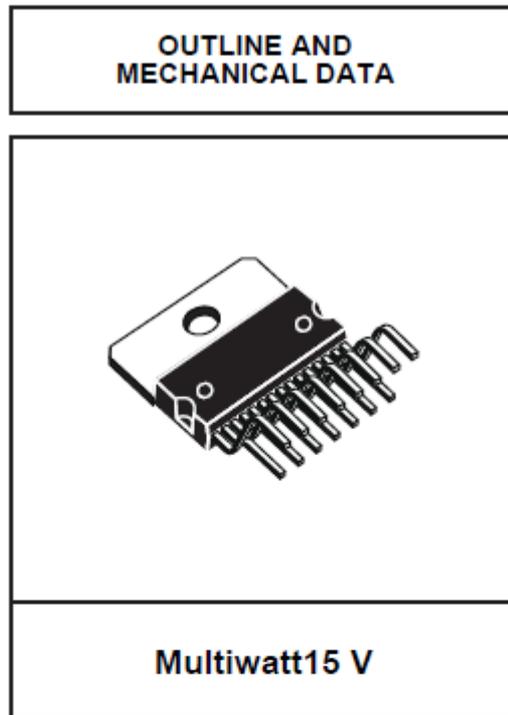
THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th(j-case)}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th(j-amb)}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate

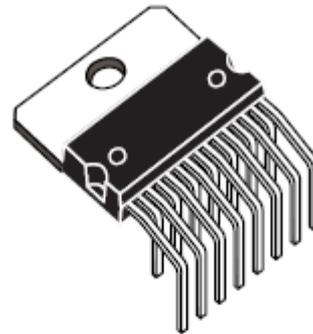
L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

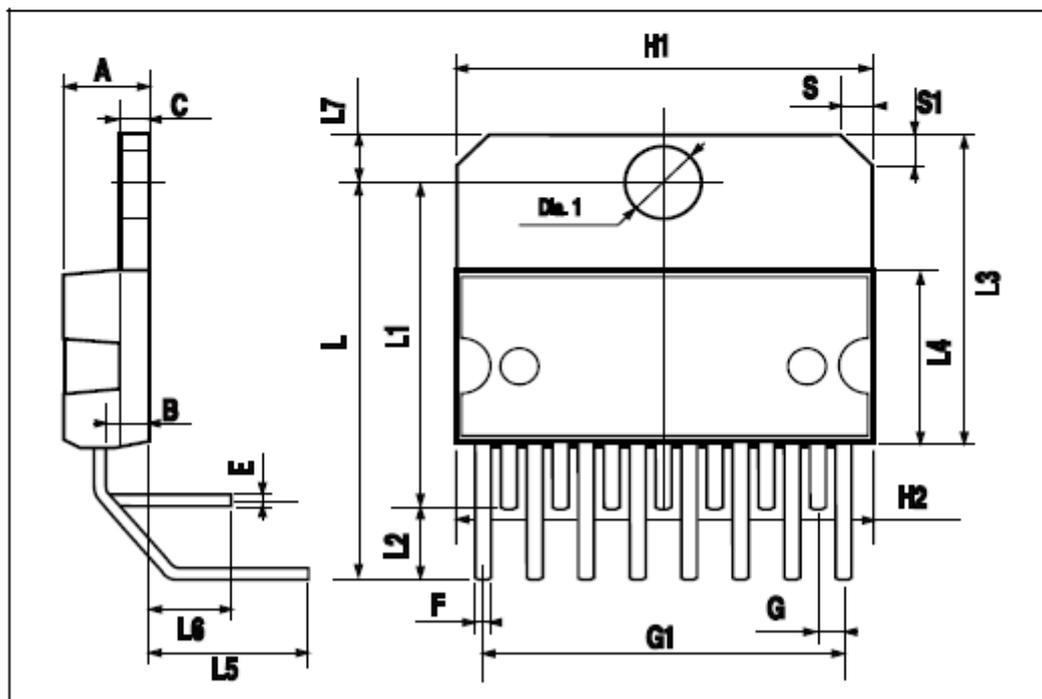


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.85			0.104
C			1.8			0.063
E	0.49		0.55	0.019		0.022
F	0.86		0.75	0.028		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.892	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.85		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



Multiwatt15 H



ANNEXE B

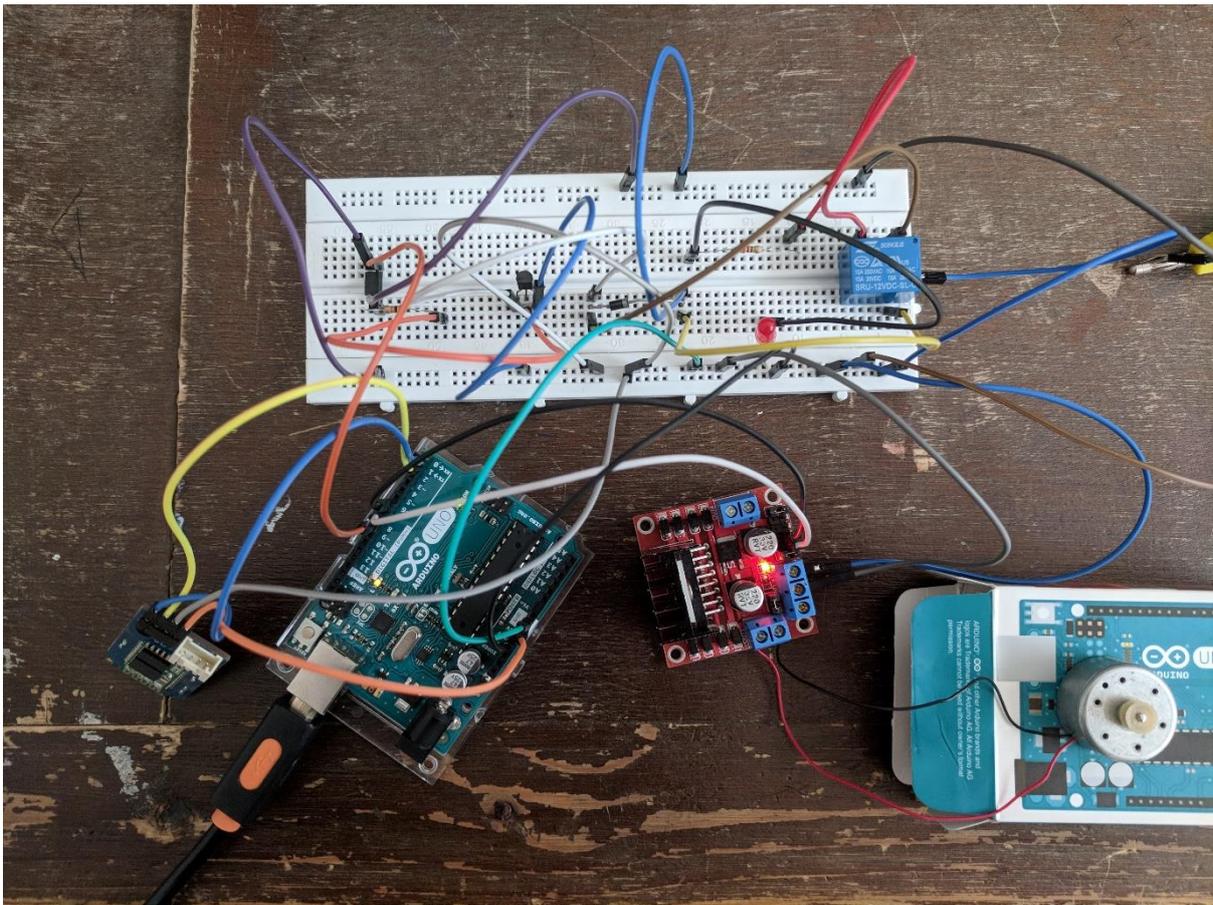
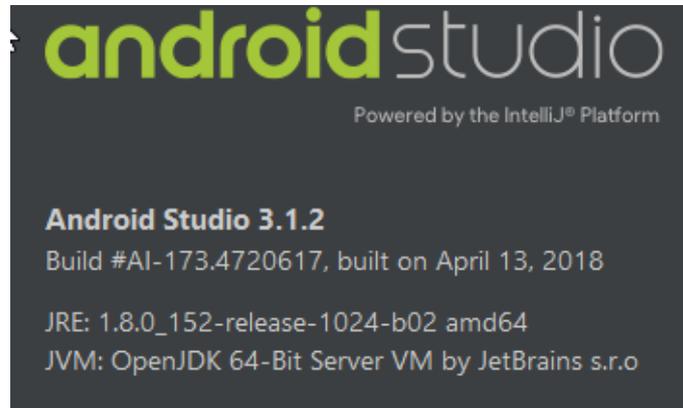


Photo de la réalisation pratique

ANNEXE C

Les Logiciels Utilisés



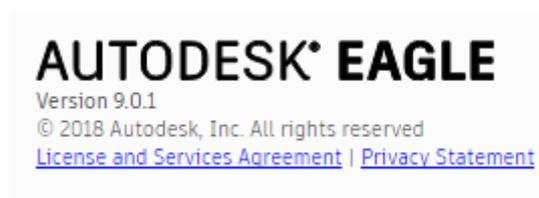
Pour programmer l'application Android



Pour programmer l'Arduino

© Labcenter Electronics 1989-2017.
Release 8.6 SP2 (Build 23525) with Advanced Simulation
<http://www.labcenter.co.uk>

Pour la simulation virtuelle de la carte



Pour le dessin de typon