

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université TAHRI Mohammed Béchar
Faculté de Technologies
Département de Génie Électrique



وزارة التعليم العالي
العلمي والبحث
جامعة طاهري محمد بشار
كلية التكنولوجيا
قسم الهندسة الكهربائية

Mémoire de Fin d'Etudes

Pour l'obtention du diplôme de licence

Domaine : Sciences et Techniques

Filière : Électrotechnique

Spécialité: Automatique

Thème

Télé-pilotage d'un robot mobile avec un système
de télé-T°

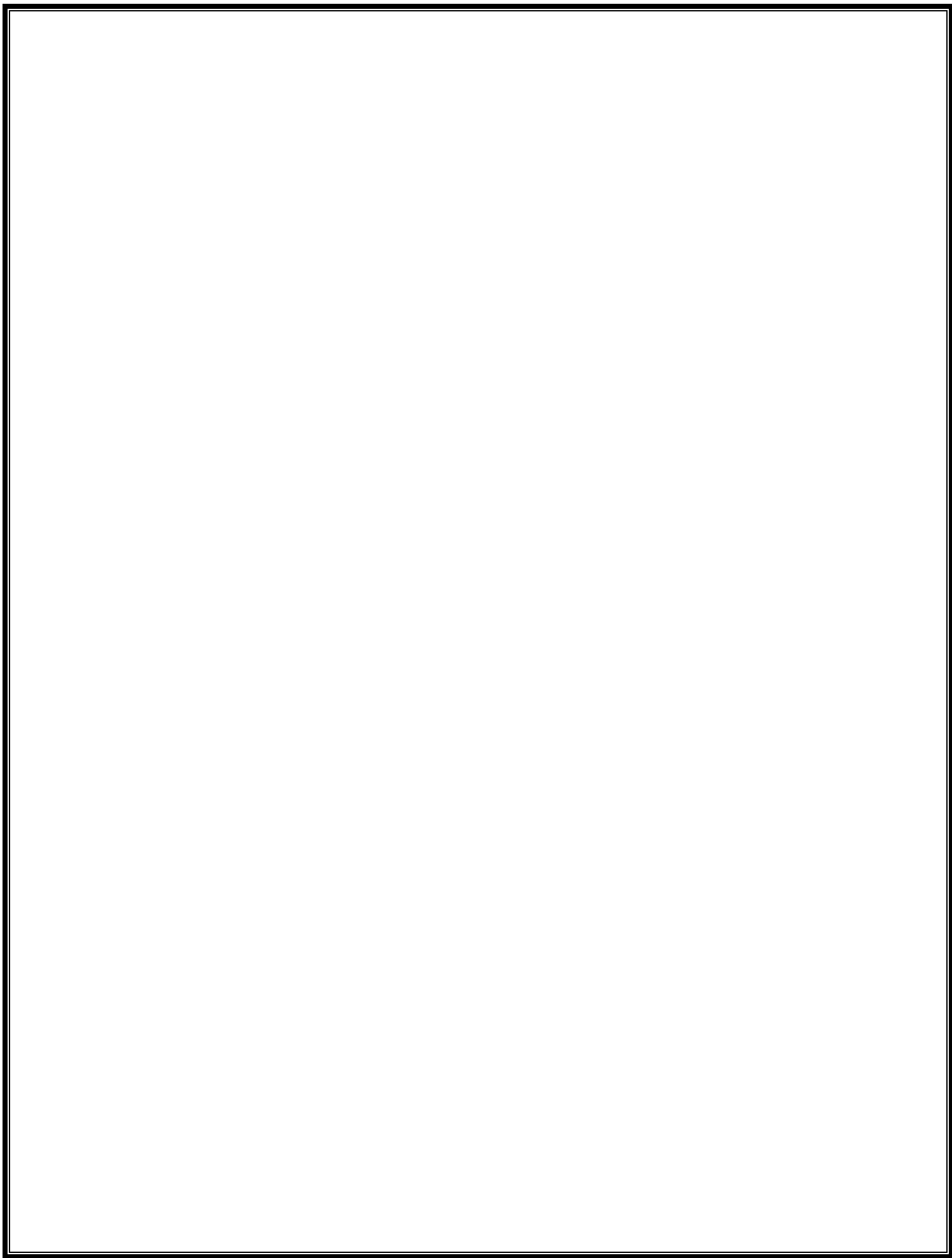
Présenté par: - Halouane Yacine
-Triki Karim

Président :

Examineur :

Encadreur : Dr. HAIDAS MOHAMMED

Année universitaire 2021/2022



Dédicaces

- A mon modèle dans la vie, ma mère **Alia** et mon père **Ahmed**, que Dieu la protège et prenne soin d'eux.
- A mes deux premières supportrices depuis que je suis jeune, mes sœurs **Nabila** et **Rokaia**.
- Au cœur tendre et au sourire chaleureux que ma grand-mère **Mahasen** ne m'a jamais oublié, que Dieu la protège.
- A celui qui m'a élevé et pris soin de moi dans mon enfance et a quitté la vie, l'homme persévérant et déterminé sur le succès et l'excellence qui restera toujours dans ma mémoire mon grand-père **Abdou**, que Dieu ait pitié de lui.
- À tous les professeurs qui m'ont enseigné du primaire au secondaire.
- À tous les professeurs qui ont étudié à l'université avec sincérité et nous ont fait exceller dans notre domaine
- Aux compagnons du chemin et à la camaraderie qui ont fait de 3 ans la chose la plus merveilleuse **Ayoub**, **Abderahmane**, **Chams Dine**.
- A mon collègue **Yacine**, sans qui je ne peux pas atteindre cette créativité.

Merci à tous

Triki Karim

Remerciements

Avant tout, je remercie le bon dieu, qui m'a donnée le courage, la volonté, la patience et la santé nécessaire pour affronter les différentes épreuves de la vie durant mes années d'études.

Au terme de mon projet de fin d'études, j'exprime ma profonde gratitude à Monsieur le Doyen de la Faculté de Génie Électrique, mes enseignants et tout le cadre administratif pour leurs efforts considérables, spécialement ceux du département Automatique.

Mon vif remerciement à Mr **Haidas Mohammed**, mon promoteur, sans qui, ce mémoire n'aura pas eu lieu. Je le remercie aussi pour son aide très consistant ainsi que pour l'intérêt avec lequel il a suivi la progression de mon travail, pour ses conseils efficaces, ses judicieuses directives et pour les moyens qu'il a mis à ma disposition matérielles et morales pour la réussite de ce travail tout au long de ma période de projet. Son soutien scientifique et hors scientifique et la confiance qu'il m'a accordée depuis que je suis au département d'automatique. Je me souviendrai toujours de lui comme un homme patient, judicieux directive avec les étudiants. En fin, rien de ce que j'écrirai ici ne sera suffisant pour lui exprimer ma gratitude.

Je remercie, également les membres de jury d'avoir accepté d'évaluer mon travail.

Vers la fin, il m'est très agréable d'exprimer toutes ma reconnaissance pour ceux qui m'ont entouré de près ou de loin pendant mes années études pour leur soutien, leur aide et, surtout, pour leur sympathie. Qu'ils trouvent ici l'expression de ma profonde reconnaissance et mon Profond respect.

Liste des figures

<i>Figure</i>	<i>Nom de figure</i>	<i>Numéro de page</i>
<i>Figure I.1</i>	<i>Premier modèle de la carte Arduino</i>	04
<i>Figure I.2</i>	<i>La carte Arduino UNO</i>	06
<i>Figure I.3</i>	<i>Microcontrôleur ATmega328</i>	09
<i>Figure I.4</i>	<i>Constitution de la carte Arduino UNO</i>	12
<i>Figure I.5</i>	<i>Interface IDE Arduino</i>	13
<i>Figure I.6</i>	<i>Paramétrage de la carte</i>	14
<i>Figure II.7</i>	<i>Vue du Module Bluetooth HC-05</i>	18
<i>Figure II.8</i>	<i>Vue du Capteur LM35</i>	18
<i>Figure II.9</i>	<i>Vue d'un servomoteur</i>	21
<i>Figure II.10</i>	<i>l'angle de l'axe du servomoteur en fonction de la largeur de l'impulsion</i>	22
<i>Figure II.11</i>	<i>DC moteur</i>	23
<i>Figure III.12</i>	<i>Arduino contrôle 2 DC moteurs via Bluetooth</i>	26
<i>Figure III.13</i>	<i>Arduino connecter avec un capteur de T°</i>	27
<i>Figure III.14</i>	<i>Programme d'Arduino contrôle 2 DC moteur</i>	31
<i>Figure III.15</i>	<i>Programme d'Arduino de T°</i>	32
<i>Figure III.16</i>	<i>Programme Bluetooth de DC moteur</i>	36
<i>Figure III.17</i>	<i>Programme Bluetooth de T°</i>	37

Liste des figures	I
Sommaire	II/III/IV

Introduction Général

1/- Généralités	01
2/- Position du problème	01
3/- Objectif du projet	02
4/- Présentation du mémoire	02

CHAPITRE I

Le dispositif programmable Arduino

I.1 Introduction	03
I.2 Historique de l'Arduino	03
I.3 Définition du module Arduino	04
I.4 Les différentes cartes Arduino	05
I.5 La carte Arduino UNO	07
I.6 La constitution de la carte Arduino UNO	08
I.6.1.1 Le Microcontrôleur ATmega328	08
I.6.1.2 Les sources de l'alimentation de la carte	10
I.6.1.3 Les entrées & sorties	10
I.6.1.4 Les ports de communications	12
I.6.2 Partie programme	13
I.6.2.1 l'environnement de la programmation	13
I.6.2.2 Structure générale du programme (IDE Arduino)	13
I.6.2.3 Injection du programme	14
I.6.2.4 Description du programme	14
I.6.2.5 Les étapes de téléchargement du programme	15
II. Conclusion	16

CHAPITRE II

Les équipements utilisés dans cette projet

II.1 Introduction	17
II.2 Description des composants utilisés	17
II.1.1 Bluetooth HC-05	17
II.1.2 Capteur de température LM 35	18
II.1.3 Servomoteur	20
II.1.4 DC moteur	23

CHAPITRE III

Réalisation d'un Robot automobile avec un détecteur de T° Et développement d'une application Androide

II.1 Introduction	24
III.2 Les types de programmation	25
III.2.1 Présentation de l'organigramme IDE	25
III.2.1.1 Organigramme Arduino UNO	25
Le programme utilisé pour représenter le schéma	26
III.3 Schéma de câblage de l'Arduino avec les DC moteurs	26
III.4 Schéma de câblage de l'Arduino avec le capteur de T°	27
III.5 Présentation du programme IDE	28
III.5.1 programme IDE de l'Arduino control 2 DC moteurs	28
III.5.2 programme IDE de capteur de T°	32
III.5.3 le système Androide	33
III.5.4 L'outil App Inventor	33

Sommaire

III.6 Explication de nos programmes	35
1er programme (DC moteur contrôlé par application Bluetooth)	35
2eme programme (Capteur de T°)	37
III.7 Conclusion	38

Conclusion Générale

1.Généralités	39
2.Problèmes rencontrés	39
3.Perspectives du projet	39

Introduction Générale

1. Généralités :

La forte augmentation des ventes de smart phone et de tablettes électronique se fait en même temps qu'une adoption rapide par le grand public des technologies de la domotique ainsi que l'autopilotage. Au fond, le smart phone, avec sa connectivité Bluetooth intégrée, devient une télécommande universelle pour toute la maison et les équipements électriques. Les utilisateurs pourront à terme contrôler à distance un très grand nombre de fonctions sans avoir à tenir compte de la marque ou de l'origine du produit qu'ils pilotent. Pour répondre à cette évolution majeure, nous avons créé une carte qui permet de contrôler n'importe quel appareil, véhicule, machine à travers un smart phone ou à une tablette. [06]

2. Position du problème :

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.

Afin d'effectuer certaines tâches qu'une personne ne peut pas effectuer et de simplifier des problèmes complexes à moins complexes grâce à cette technologie.

3.Objectif du projet :

Dans ce projet trois objectifs ont été visés :

- Le premier est de regrouper suffisamment d'informations sur une grande catégorie de cartes d'interfaçage (Arduino) : son langage de programmation, sa construction, son principe de fonctionnement.
- Le deuxième consiste à mentionner les équipements utilisés dans ce projet et les définitions
- Le troisième est de réaliser une application sous smart phone concerne les programmes de Arduino en expliquant les différents blocs de sa construction.

4.Présentation du mémoire :

Le premier chapitre sera consacré à une étude approfondie sur la carte d'interface tel 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 afin de simplifier son utilisation.

Le deuxième chapitre sera consacré à Les équipements utilisés dans ce projet ainsi que la description de chaque bloc du circuit.

Dans le troisième chapitre, on présentera Les programmes que on a utilisé ; on expliquera le fonctionnement de notre programme, son déclenchement à l'action à distance et cela sur deux modes de commande code simulé dans la carte Arduino et de réaliser une application capable de gérer une telle commande sous smarte phone.

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

Chapitre I

Le dispositif programmable

Arduino

I.1 Introduction :

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

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

I.2 Historique de l'Arduino :

L'Arduino est à l'origine un projet d'étudiants de l'école de Design d'Interaction d'Ivrea en Italie. Au début des années 2000, les outils de conception de projets dans le domaine du design d'interaction étaient onéreux, proches d'une centaine d'euros. Ces outils étaient pour la plupart conçus pour le domaine de l'ingénierie et de la robotique. Maîtriser et utiliser ces composants demandait beaucoup de temps et d'apprentissage et ralentissait fortement le processus de création pour ces jeunes étudiants.

Il leur vient alors à l'idée de créer une plateforme plus abordable et plus simple à utiliser, reposant sur l'environnement de développement Processing mis au point en 2001 par des étudiants du MIT. C'est donc en 2003 que, pour un projet de fin d'études, fut conçue la carte Wiring, ancêtre de l'Arduino. Visant à rendre la plateforme toujours moins chère et plus accessible, une équipe d'étudiants et de professeurs finirent par

concevoir la toute première Arduino en 2005. Entièrement open source, l'Arduino présentait l'avantage d'être multiplateforme et d'être en perpétuelle optimisation par la communauté d'utilisateurs.

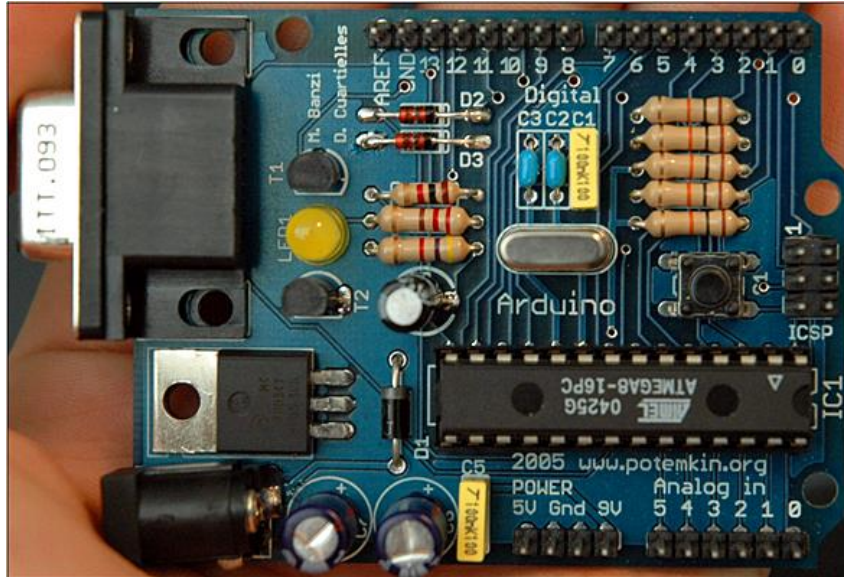


Figure I.1 : Premier modèle de la carte Arduino

I.3 Définition du module Arduino :

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

certaines modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino. [3]

I.4 Les différentes cartes Arduino :

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

- **Le NG d'Arduino**, avec une interface d'USB pour programmer et usage d'un ATmega8.
- **L'extrémité d'Arduino**, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- **L'Arduino Mini**, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- **L'Arduino Nano**, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- **Le Lily Pad Arduino**, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- **Le NG d'Arduino plus**, avec une interface d'USB pour programmer et usage d'un ATmega168.
- **L'Arduino Bluetooth**, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- **L'Arduino Diecimila**, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- **L'Arduino Duemilanove** ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- **L'Arduino Méga**, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire. [4]

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

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

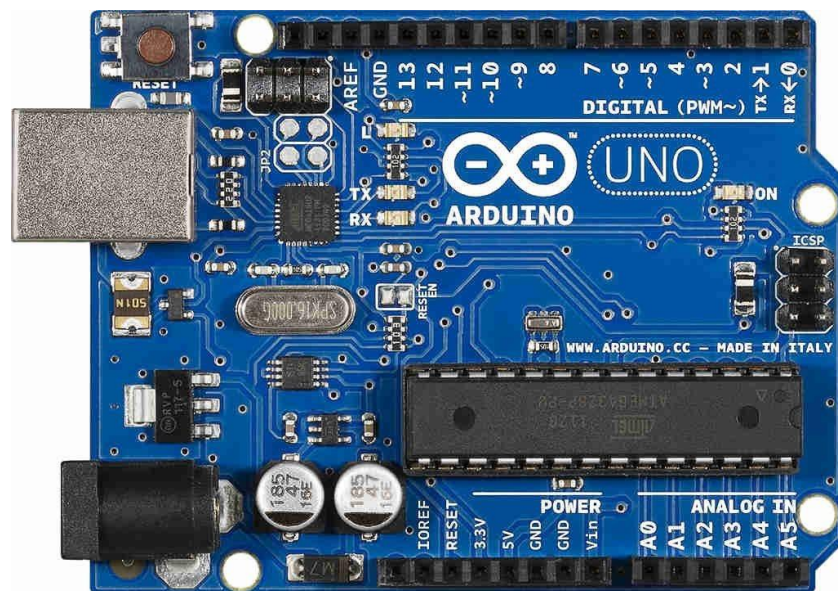


Figure I.2 : La carte Arduino UNO

I.5 La carte Arduino UNO :

I.5.1 Pourquoi Arduino UNO :

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

- **Le prix (réduits):** les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino préassemblées coûtent moins de 2500 Dinars).
- **Multi plateforme:** le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- **Un environnement de programmation clair et simple:** l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **Logiciel Open Source et extensible :** Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui Peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).
- **Matériel Open source et extensible:** les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168,

ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût.

I.6 La constitution de la carte Arduino UNO :

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

I.6.1.1 Le Microcontrôleur ATMega328 :

Un microcontrôleur ATMega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. La figure I.3 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino [1]



Le composant CMS



Le composant classique

Figure I.3: Microcontrôleur ATMega328

Le microcontrôleur ATMega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- **La mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont boot loader de 0.5 ko).
- **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **EEPROM :** C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [2]

I.6.1.2 Les sources de l'alimentation de la carte :

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

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

I.6.1.3 Les entrées & sorties :

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode ()`, `digitalWrite ()` et `digitalRead ()` du langage Arduino. Ces broches fonctionnent en 5V.

Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digital Write (broche, HIGH).

En plus, certaines broches ont des fonctions spécialisées :

- **Interruptions Externes:** Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. - Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analog Write ().
- **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface- Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface- interface "2 fils").
- **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

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

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte

Fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé. [3]

I.6.1.4 Les ports de communications :

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

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

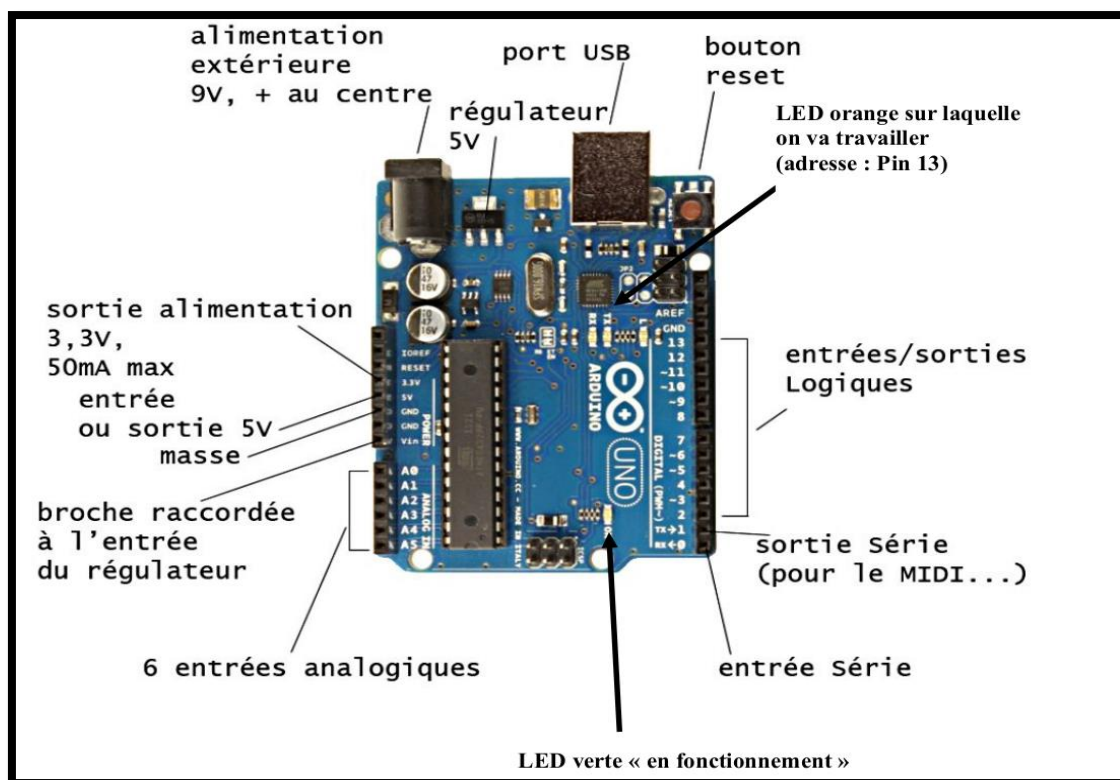


Figure I.4: Constitution de la carte Arduino UNO

I.6.2 Partie programme :

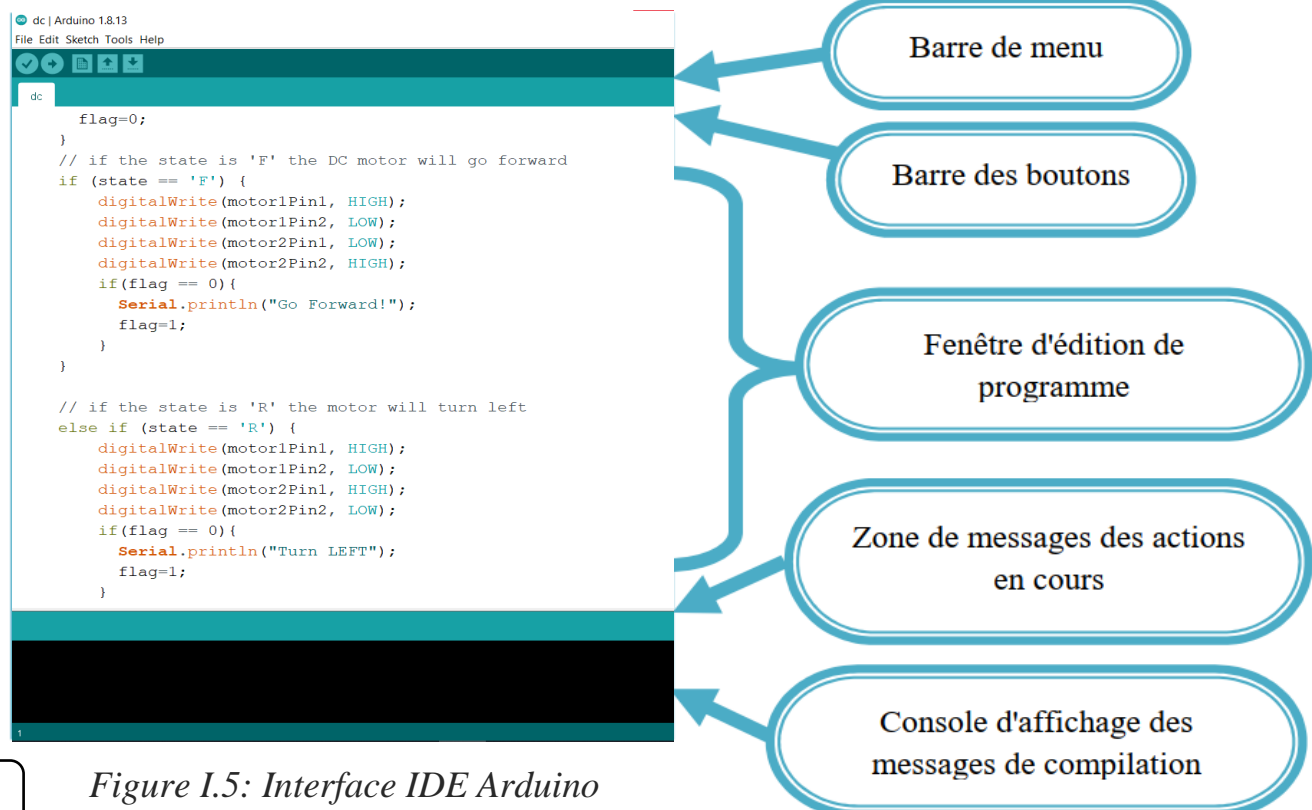
Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

I.6.2.1 l'environnement de la programmation :

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

I.6.2.2 Structure générale du programme (IDE Arduino) :

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



I.6.2.3 Injection du programme :

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

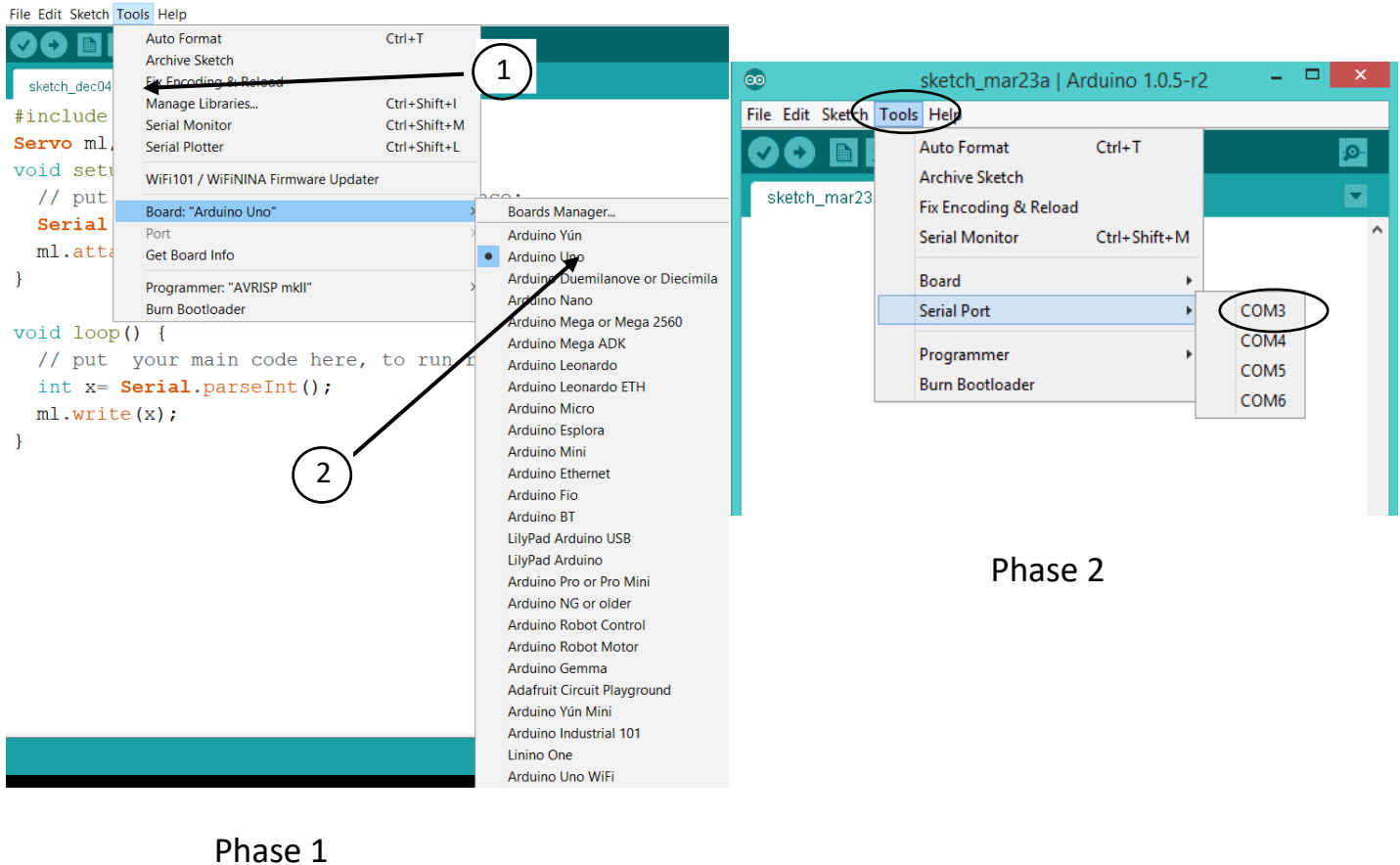


Figure I.6: Paramétrage de la carte

I.6.2.4 Description du programme :

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

Commentaire

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.

Définition des variables

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

Configuration des entres 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.

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 milliseconde) 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.

I.6.2.5 Les étapes de téléchargement du programme :

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

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

II. Conclusion :

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

Chapitre II

Les équipements utilisés dans

Ce projet

II.1 Introduction :

Dans ce chapitre, on présentera de manière sommaire une vue d'ensemble du dispositif expérimental « télé pilotage d'un robot mobile avec un système de télé-T° ».

Ce travail d'initiation à base d'une carte Arduino UNO permet de commander autoguidage d'un véhicule à distance (contrôle deux moteurs dans les deux sens de rotation) via système Bluetooth.

Après avoir donné dans le chapitre précédent une description théorique sur le module Arduino et son environnement de développement, nous passerons aux équipements utilisés dans cette projet expérimentale.

II.2 Description des composants utilisés :

II.1.1 Bluetooth HC-05 :

Le module HC-05 n'est pas plus gros qu'un pouce. Il est en fait un montage d'un module Bluetooth sur un petit P.C.B, cela permet de s'affranchir de certaines contraintes comme la soudure du module (qui est très délicate), la conversion 5V et 3.3V, la régulation de l'alimentation (3.3V de nouveau) ou encore l'ajout de LED de signal, tout cela est déjà intégré.

La norme « Bluetooth » fut mise au point en 1994 par le fabricant Suédois Ericsson à Lund. Le terme signifie littéralement en français : « dent bleue ».

Le Bluetooth est un protocole de communication sans fil, permettant l'échange bidirectionnel de données à une distance bien précise (15 à 20 mètres), en utilisant des ondes radio UHF sur une bande de fréquence de 2.4 GHz. Son objectif est de simplifier les connexions entre les appareils électroniques en supprimant les liaisons filaires.

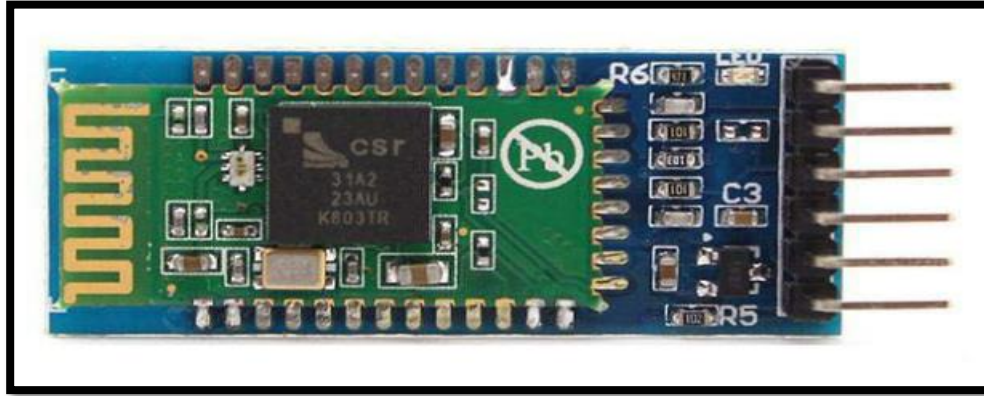


Figure II.7: Vue du Module Bluetooth HC-05

II.1.2 Capteur de température LM 35 :

Le capteur de température LM35 est un capteur de température analogique fabriqué par Texas Instruments, il est extrêmement populaire en électronique, car précis, peu coûteux, très simple d'utilisation et d'une fiabilité à toute épreuve, Il peut fonctionner dans une plage de -55°C à $+150^{\circ}\text{C}$.

Il dispose de trois pattes, la première pour la tension d'entrée (l'alimentation), la deuxième c'est la tension de sortie V_{out} la troisième pour la masse. Dont la tension de sortie V_{out} est linéairement proportionnelle à la température exprimée en degré Celsius, plus exactement, la tension de sortie V_{out} augmente de 10 mV chaque fois que la température augmente de 1°C, selon les données fournies par le constructeur.

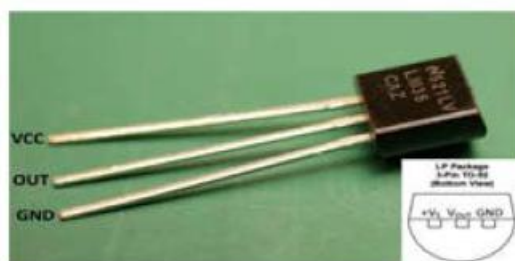


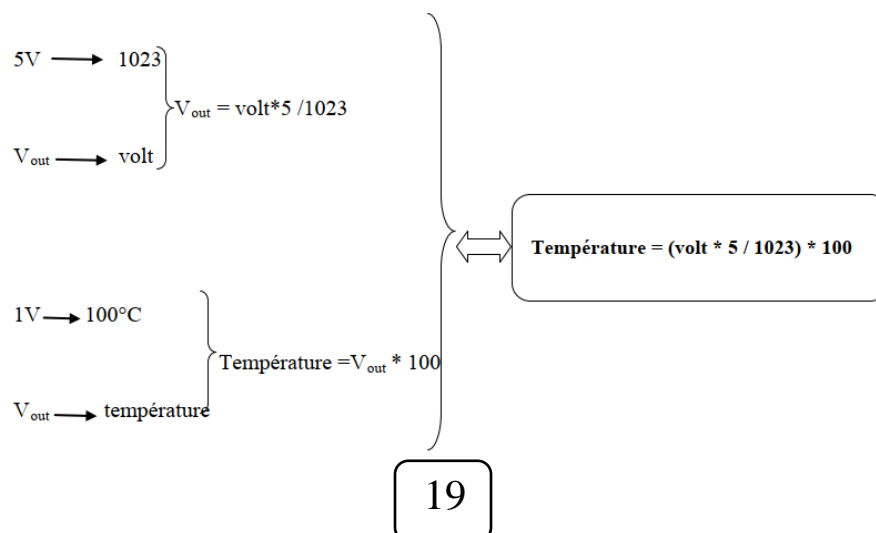
Figure II.8: Vue du Capteur LM35

<i>Nombre de broches</i>	<i>3 broches</i>
<i>Dimension</i>	4.30 mm x 4.30 mm
<i>Tension d'alimentation min</i>	4V
<i>Tension d'alimentation max</i>	20V
<i>Température Min</i>	-55 °C
<i>Température Max</i>	+155°C
<i>Consommation</i>	60μA
<i>Gain de la sonde (sortie)</i>	10mV /°C
<i>Précision</i>	±0.75°C

Le LM35 est un capteur, qui converti la variation de la température en une différence de potentielle, d'une façon proportionnelle. Ce capteur est très simple à utiliser, il suffit de l'alimenter avec les deux pattes VCC et GND, la sortie V_{out} sera récupérer par une entrée analogique d'un microcontrôleur (ATmega328). La conversion est aussi simple puisqu'un volt correspond à 100 degrés Celsius, la lecture analogique du signal de 0 à 5V étant codée de 0 à 1023, en utilise la formule suivante :

$$\text{Température} = (\text{volt} * 5 / 1023) * 100$$

Volt : est la valeur analogique(V_{out}) converti par le CAN en une valeur numérique.



II.1.3 Servomoteur :

Un servomoteur est un actionneur (moteur) capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continue et corrigée en fonction de la mesure. Le servomoteur est composé de plusieurs éléments visible:

- Les files.
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal.
- Le boîtier qui le protège.

Aussi de plusieurs éléments que l'on ne voit pas qui se trouve à l'intérieur du boîtier.

- Un moteur à courant continu.
- Des engrenages pour former un réducteur.
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent).
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.

La position est définie avec une limite de débattement d'angle de 180 degrés. Souvent abrégé en « servo » provenant du latin servus qui signifie « esclave ».

Il existe plusieurs servomoteurs, dans notre application en a utilisé un servomoteur de faible puissance (9 Gramme) SG90.

Il contient trois fils, un pour l'alimentation, l'autre pour la masse(GND) et la dernière pour le signal d'entrée (la commande).

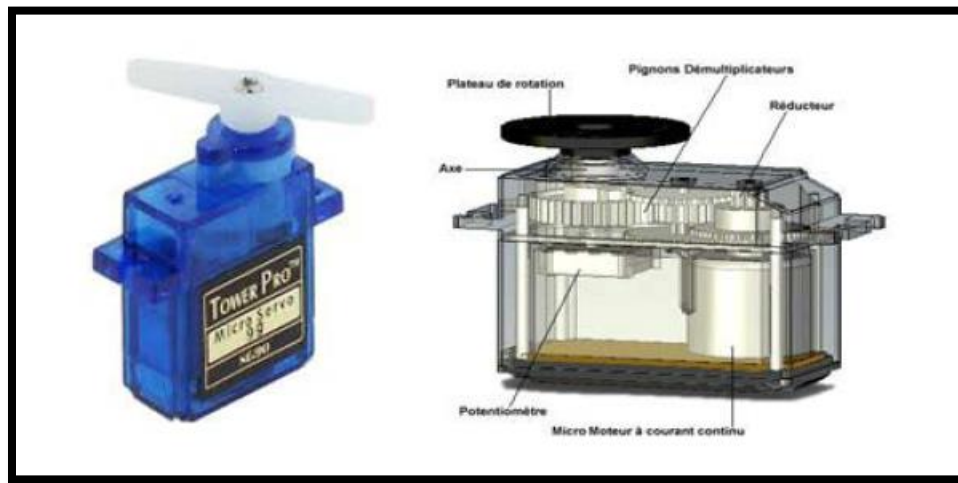


Figure II.9: Vue d'un servomoteur

Nombres de broches	3 fils	
	Marron	Masse
	Rouge	Vcc
	Orange	Commande
Dimension	22mm x 11.5mm x 27mm	
Poids	9 grammes	
Tension d'alimentation Min	4.8V	
Tension d'alimentation Max	6V	
Vitesse	0.12 s/60° sous 4.8V	
Couple	1.2 kg/cm sous 4.8V	
Consommation	125	

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique qui lui transmette des consignes de position sous forme d'un signal codé en largeur d'impulsion communément appelé PWM. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position de bras de commande du servomoteur. Comme la montre la figure ci-dessous :

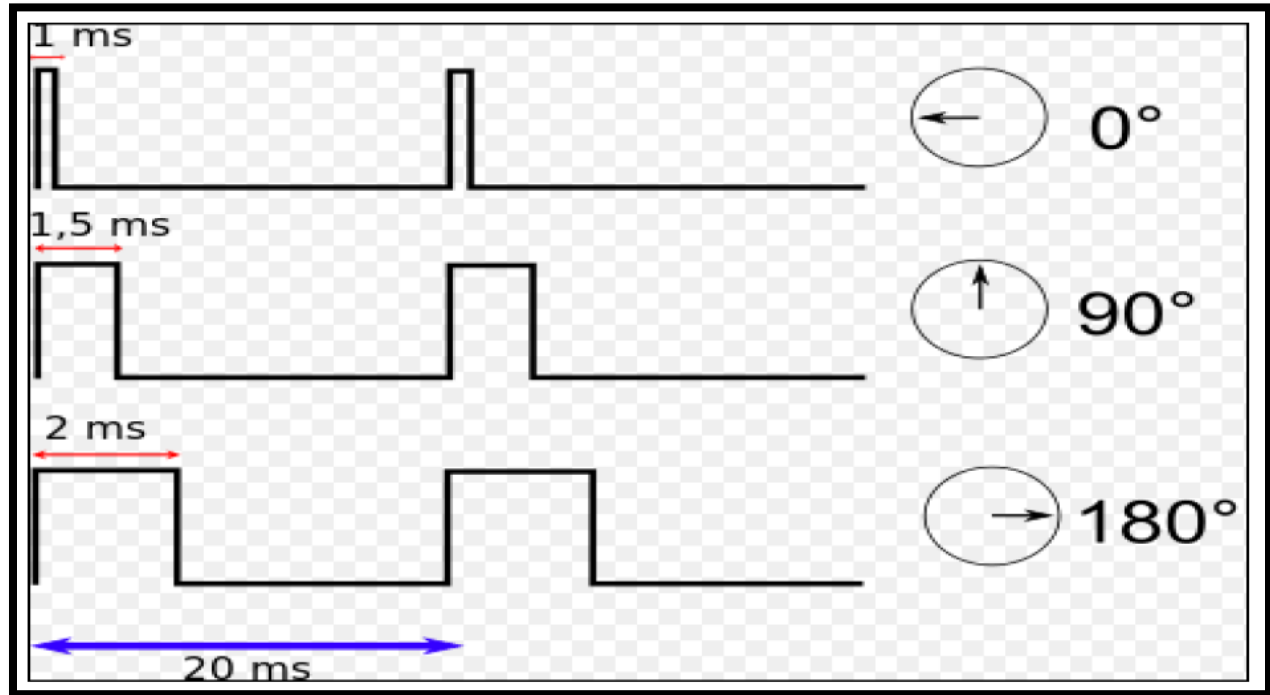


Figure II.10: l'angle de l'axe du servomoteur en fonction de la largeur de l'impulsion

Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

Lorsque le moteur tourne, l'axe du servomoteur change de position, ce qui modifie la résistance du potentiomètre. Le rôle de l'électronique est de commander le moteur pour que la position de l'axe de sortie soit conforme à la consigne reçue.

II.1.4 DC moteur :

Un moteur à courant continu fait partie d'une classe de moteurs électriques rotatifs qui convertit l'énergie électrique à courant continu en énergie mécanique. Les types les plus courants reposent sur les forces produites par les champs magnétiques. Presque tous les types de moteurs à courant continu ont un mécanisme interne, électromécanique ou électronique, pour changer périodiquement la direction du courant dans une partie du moteur.



Figure II.11: DC moteur

Chapitre III

Réalisation d'un Robot automobile
avec un détecteur de T°

Et développement d'une
application Androïde

III.1 Introduction :

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

Lors des années précédentes, le traitement des données informatiques se fait par des ordinateurs ; en revanche le smart phone a des avantages qui ont les mêmes fonctions que l'outil informatique ; ce dernier porte l'intérêt de l'ordinateur grâce à l'androïde. La téléphonie mobile a connu une explosion dans les années 2000 mais aucune révolution n'a semblé arriver depuis que les appareils se ressemblent. Les innovations n'avaient plus vraiment de saveur ; les applications étaient difficiles d'accès de par leur mode de distribution et souvent peu performantes à cause des faibles capacités des appareils. Depuis quelques 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, marché d'applications en ligne.

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

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

III.2 Les types de programmation :

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

III.2.1 Présentation de l'organigramme IDE :

L'organigramme est une représentation schématique des liens fonctionnels, organisationnels et hiérarchiques d'un organisme, d'un programme, etc. Il se doit de posséder une référence documentaire.

III.2.1.1 Organigramme Arduino UNO :

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.

Le programme utilisé pour représenter les schémas :

Fritzing ;

Fritzing est un logiciel open-source multiplateforme permettant de construire des schémas et des circuits électroniques que nous utilisons avec Arduino. Plusieurs vues sont disponibles : platine d'essai, schémas électriques et circuit imprimé. [12]

III.3 Schéma de câblage de l'Arduino avec les DC moteurs :

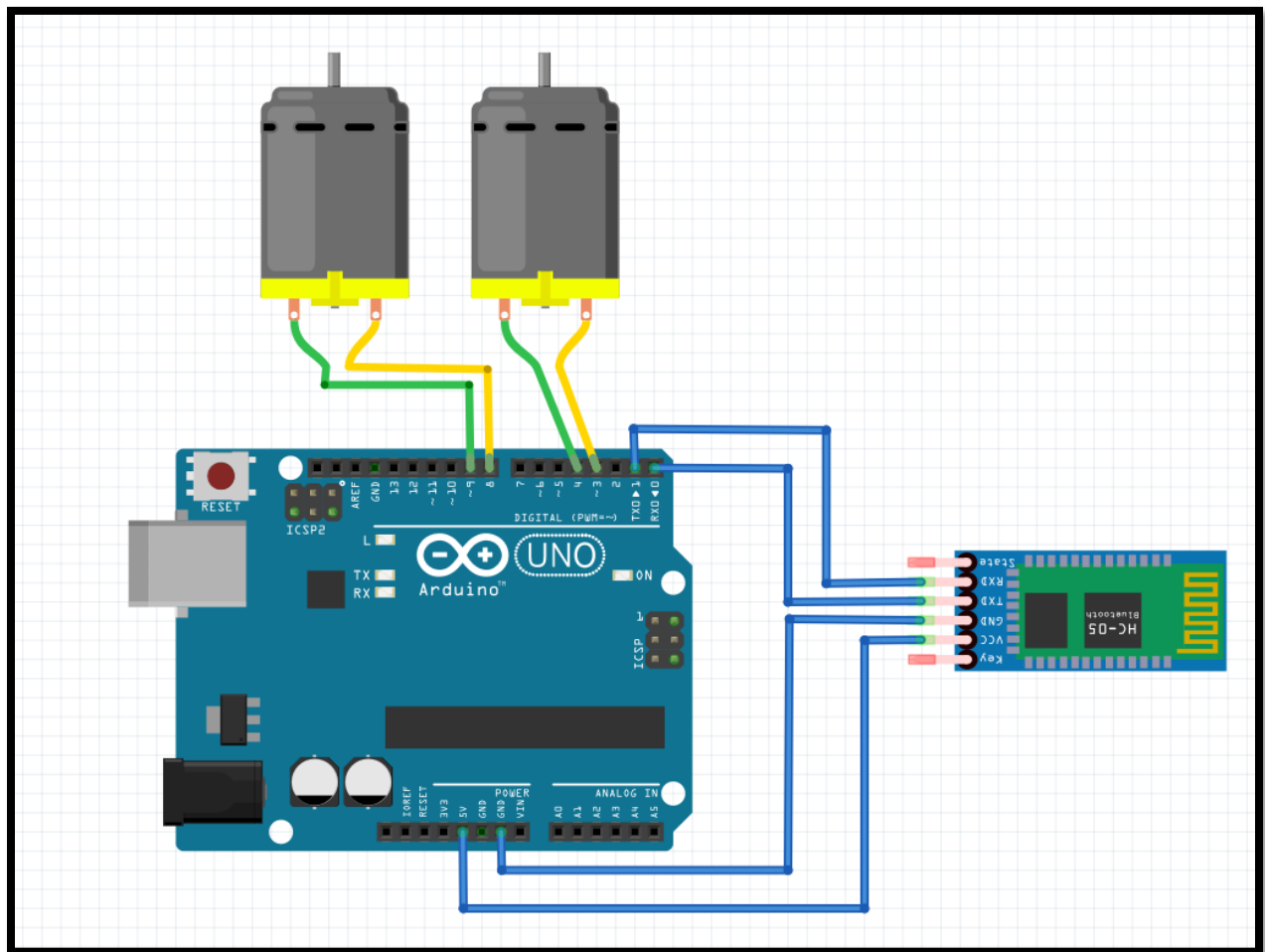


Figure III.12: Arduino controle 2 DC moteurs via Bluetooth

III.4 Schéma de câblage de l'Arduino avec le capteur de T° :

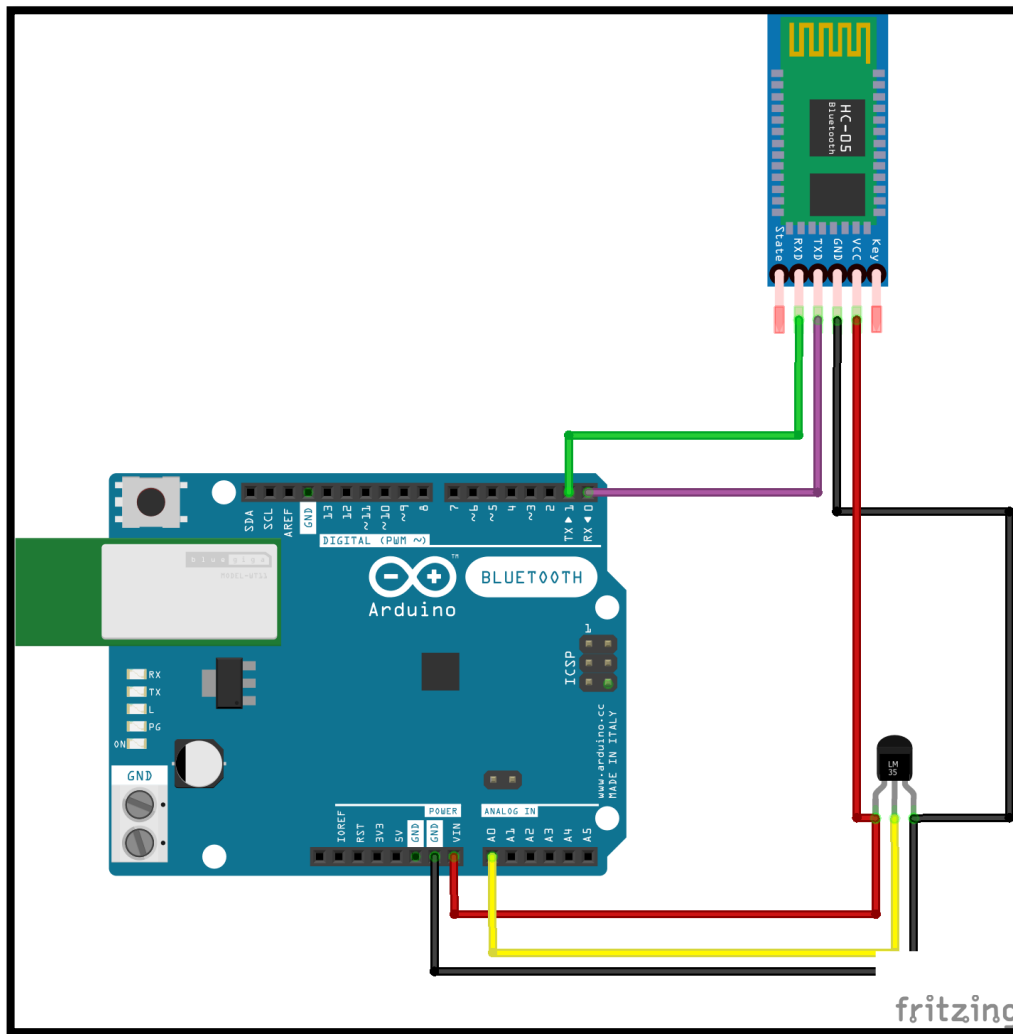


Figure III.13: Arduino connecter avec un capteur de T°

Matériel utiliser :

- 2 Arduino UNO
- 2 DC Moteur
- 2 Bluetooth
- Capteur de Température (LM35)

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.

III.5.1 programme IDE de l'Arduino control 2 DC moteurs :

```
FPLUTSHIMWBBWQ5$  
  
int motor1Pin1 = 4;  
int motor1Pin2 = 3;  
int enable1Pin = 6;  
int motor2Pin1 = 9;  
int motor2Pin2 = 8;  
int enable2Pin = 11;  
int state;  
int flag=0;  
int stateStop=0;  
void setup() {  
    pinMode(motor1Pin1, OUTPUT);  
    pinMode(motor1Pin2, OUTPUT);  
    pinMode(enable1Pin, OUTPUT);  
    pinMode(motor2Pin1, OUTPUT);  
    pinMode(motor2Pin2, OUTPUT);  
    pinMode(enable2Pin, OUTPUT);  
    digitalWrite(enable1Pin, HIGH);  
    digitalWrite(enable2Pin, HIGH);  
    Serial.begin(9600);  
}  
  
void loop() {  
    if(Serial.available() > 0){  
        state = Serial.read();
```


FPLUTSHIMWBBWQ5\$

```
    flag=0;
}
// if the state is 'F' the DC motor will go forward
if (state == 'F') {
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, HIGH);
    if(flag == 0){
        Serial.println("Go Forward!");
        flag=1;
    }
}

// if the state is 'R' the motor will turn left
else if (state == 'R') {
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, HIGH);
    digitalWrite(motor2Pin2, LOW);
    if(flag == 0){
        Serial.println("Turn LEFT");
        flag=1;
    }
}
```

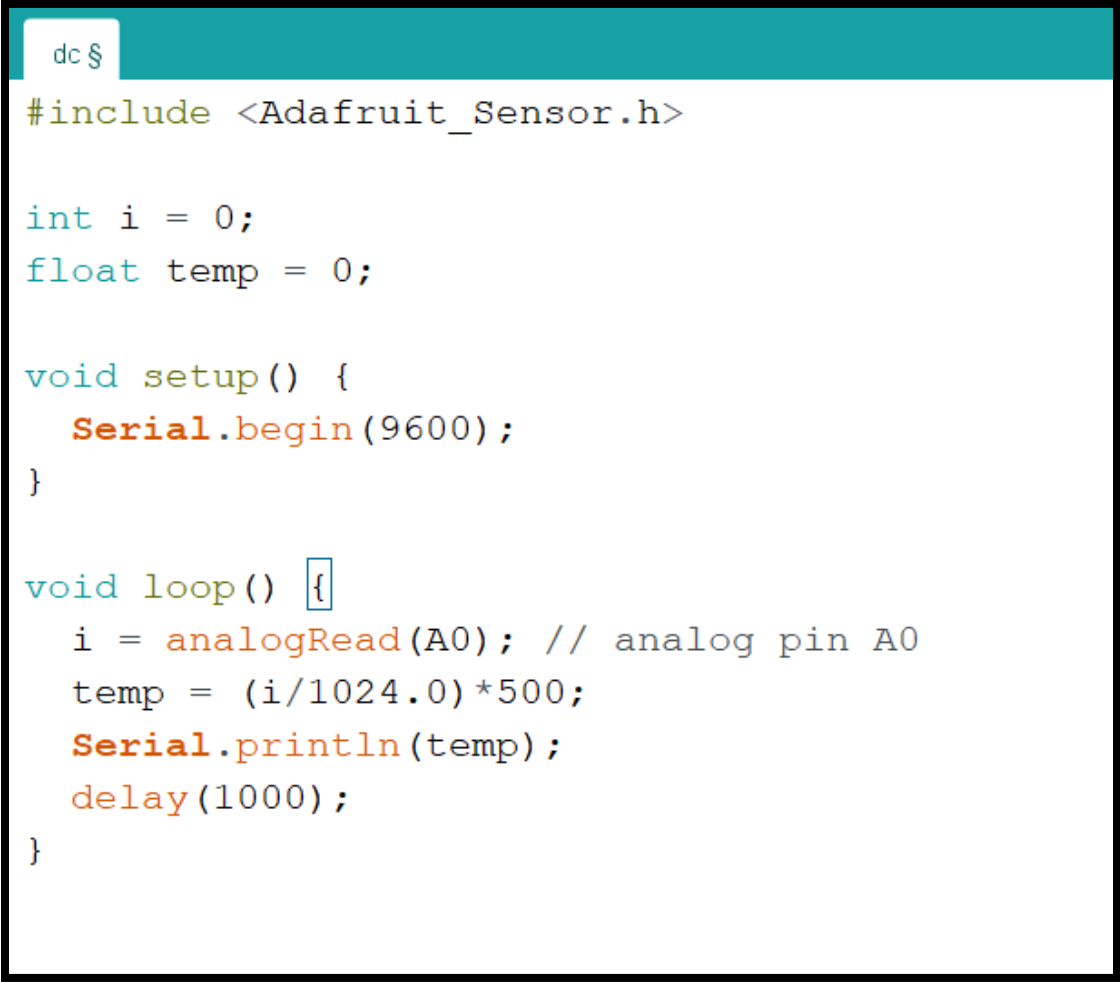
FPLUTSHIMWBBWQ5§

```
        delay(1500);
        state=3;
        stateStop=1;
    }
    // if the state is 'S' the motor will Stop
    else if (state == 'S' || stateStop == 1) {
        digitalWrite(motor1Pin1, LOW);
        digitalWrite(motor1Pin2, LOW);
        digitalWrite(motor2Pin1, LOW);
        digitalWrite(motor2Pin2, LOW);
        if(flag == 0){
            Serial.println("STOP!");
            flag=1;
        }
        stateStop=0;
    }
    // if the state is 'L' the motor will turn right
    else if (state == 'L') {
        digitalWrite(motor1Pin1, LOW);
        digitalWrite(motor1Pin2, HIGH);
        digitalWrite(motor2Pin1, LOW);
        digitalWrite(motor2Pin2, HIGH);
        if(flag == 0){
            Serial.println("Turn RIGHT");
```

```
        flag=1;
    }
    delay(1500);
    state=3;
    stateStop=1;|
}
// if the state is 'B' the motor will Reverse
else if (state == 'B') {
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, HIGH);
    digitalWrite(motor2Pin1, HIGH);
    digitalWrite(motor2Pin2, LOW);
    if(flag == 0){
        Serial.println("Reverse!");
        flag=1;
    }
}
}
```

Figure III.14: Programme d'Arduino contrôle 2 DC moteurs

III.5.2 programme IDE de capteur de T° :

A screenshot of an Arduino IDE code editor. The editor has a teal header bar with the text "dc §". The code is written in a monospaced font with syntax highlighting. The code includes a header file, declares two variables, and defines two functions: setup() and loop(). The loop() function reads an analog value from pin A0, converts it to a temperature, prints it, and delays for 1000ms.

```
dc §  
  
#include <Adafruit_Sensor.h>  
  
int i = 0;  
float temp = 0;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  i = analogRead(A0); // analog pin A0  
  temp = (i/1024.0)*500;  
  Serial.println(temp);  
  delay(1000);  
}
```

Figure III.15: Programme d'Arduino de T°

Le but de ces programmes sont d'effectuer des actions d'opération codées avec des lettres ou des chiffres envoyés du téléphone (Androïde) à l'Arduino et vice versa ; Chaque caractère envoie des instructions cryptées qui sont converties en commandes Arduino Qui consiste à déplacer ou à faire pivoter l'un des moteurs gauche ou droit. Chaque caractère est encodé dans un fichier simple et court ; Inclut des instructions décimales sur les lignes d'entrée du microcontrôleur. On commence par programmer le microcontrôleur sur l'Arduino ; Au début nous déclare toutes les variables utilisées après avoir initialisé les ports Arduino en tant que

sorties "OUTPUTS", puis ; Nous appelons la routine de synchronisation déjà ci-dessous compilateur IDE. Le programme incrémente automatiquement le code du caractère pour exécuter ou non les commandes.

III.5.3 le système Androïde :

Androïde est un système d'exploitation développé initialement pour les Smart phones. Il utilise un noyau Linux qui est un système d'exploitation libre pour PC et intègre tous les utilitaires et les périphériques nécessaires à un smart phone. Il est optimisé pour les outils Gmail. Aussi, l'androïde est libre et gratuit et a été ainsi rapidement adopté par des fabricants. La société Androïde a été rachetée en 2007 par Google. Mais aujourd'hui, l'Androïde est utilisé dans de nombreux appareils mobiles (smart phones). Les applications sont exécutées par un processeur de type ARM à travers un interpréteur JAVA. En plus de cela, l'androïde concurrence l'opérateur système d'Apple qu'il tend à dépasser en nombre d'utilisateurs. Androïde évolue pour mieux gérer l'hétérogénéité des appareils qu'il utilise. [13]

III.5.4 L'outil App Inventor :

App Inventor est un outil de développement des applications en ligne pour les smart phones sous androïde et permet à chacun de créer son application personnelle pour le système d'exploitation Androïde qui est développée par Google. La plateforme de développement est offerte à tous les utilisateurs possédant un compte Gmail. Elle rappelle certains langages de programmation simplifiés basés sur une interface graphique similaire à Scratch. Les informations des applications sont stockées sur des serveurs [13]

Distants. Elles sont actuellement entretenues par le Massachusetts Institute of Technology (MIT). L'environnement de App Inventor contient trois fenêtres qui sont proposées pendant le développement :

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

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

III.6 Explication de nos programmes :

1^{er} programme (DC moteur contrôlé par application Bluetooth) :

Notre interface dispose d'un bouton pour activer et désactiver Bluetooth, donc avant d'envoyer un caractère via des boutons, vous devez d'abord vous assurer que Bluetooth est activé ; Un mot de passe a été programmé pour des raisons de sécurité afin d'exécuter n'importe quelle commande. Cette figure montre que notre application appelée "BT control" peut être installée sur les smartphones, sa taille est de 2 mégaoctets en première version.

Après vous être assuré que Bluetooth est activé et que votre téléphone est connecté à l'appareil Bluetooth, vous pouvez maintenant utiliser librement l'application, qui est un ensemble de boutons qui déplacent notre robot où :

F : signifie que le robot avance.

R : représente la rotation du robot vers la droite.

L : signifie que le robot tourne vers la gauche.

B : signifie que le robot recule.

S : signifie que le robot s'est complètement arrêté.

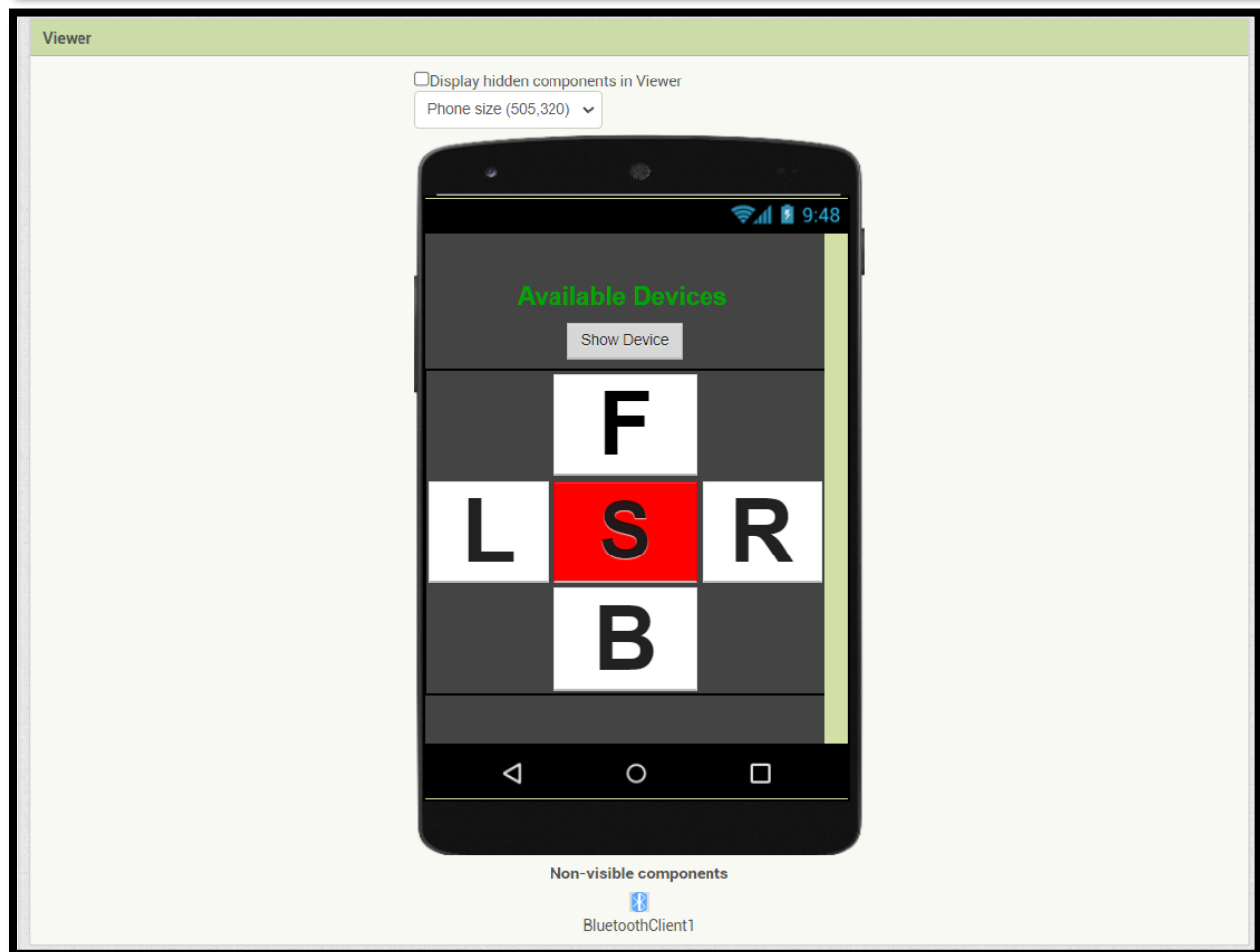
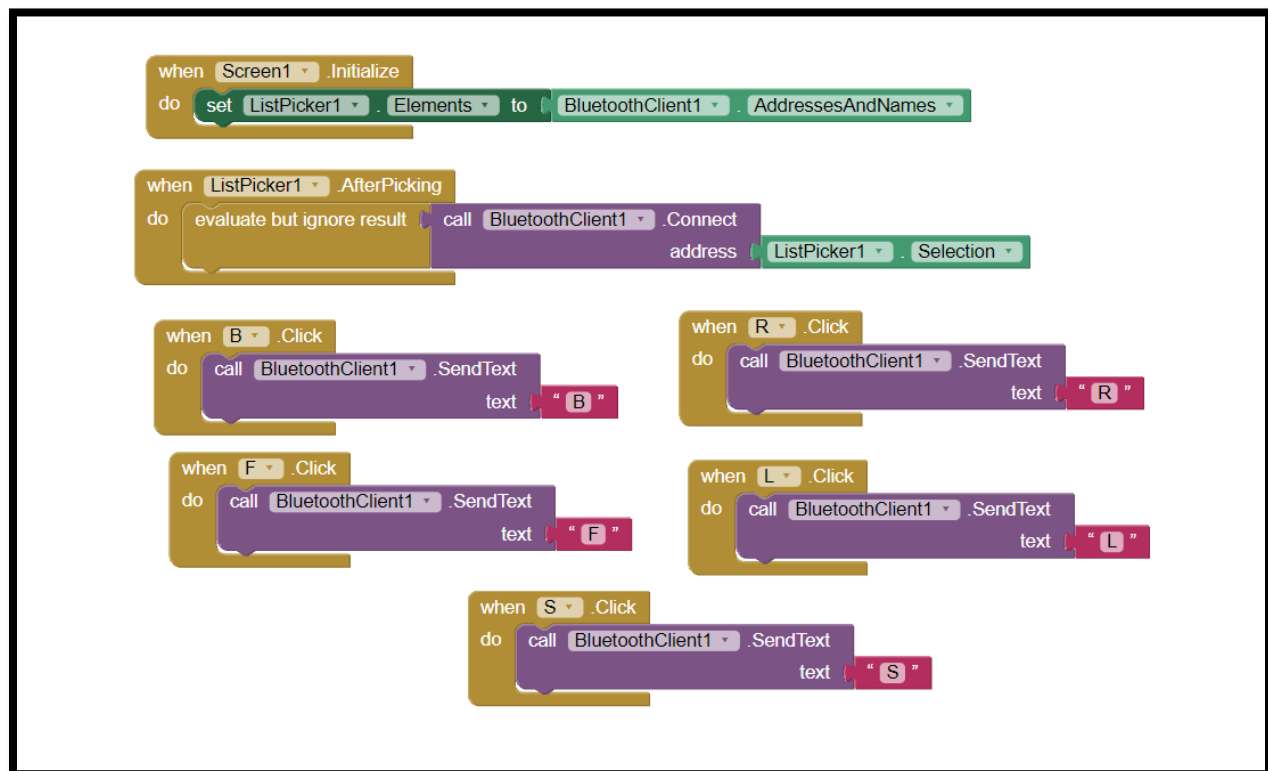


Figure III.16: L'application Bluetooth de DC moteur

2^{ème} programme (Capteur de T°) :

Cette application, elle contient un compteur ou une jauge de température comme le capteur de température connecté à l'Arduino capte la T° et l'envoi à l'application pour l'afficher sur l'appareil.

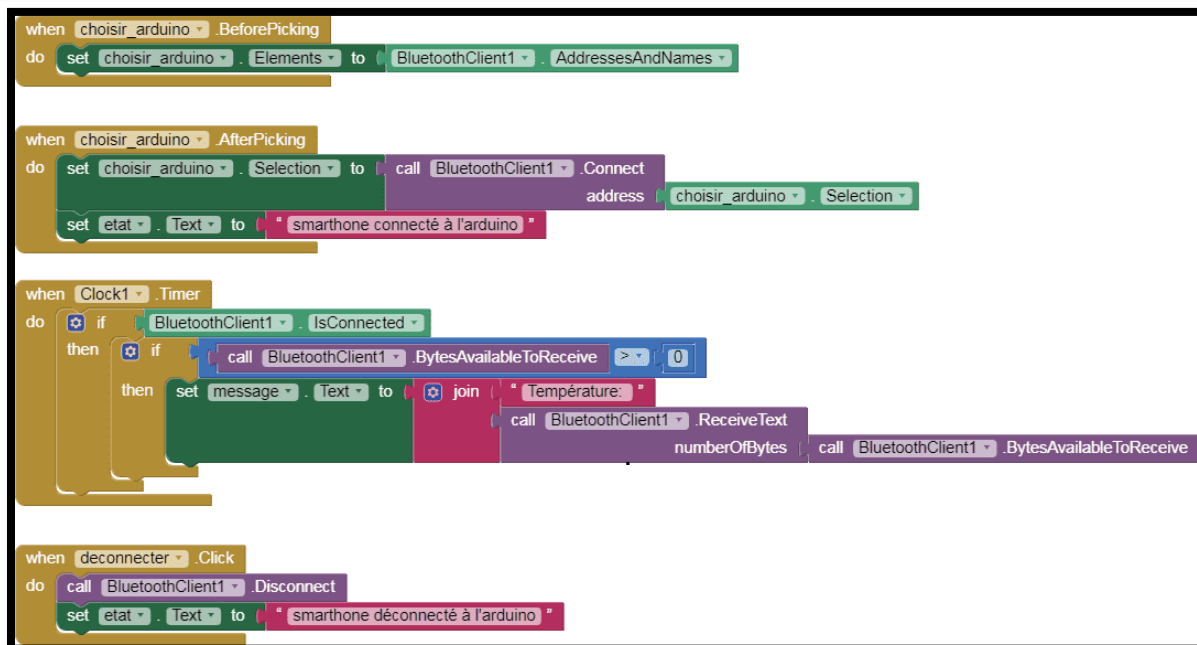
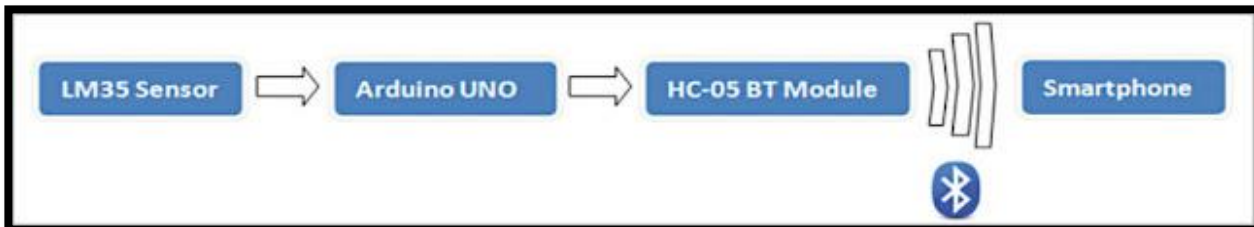
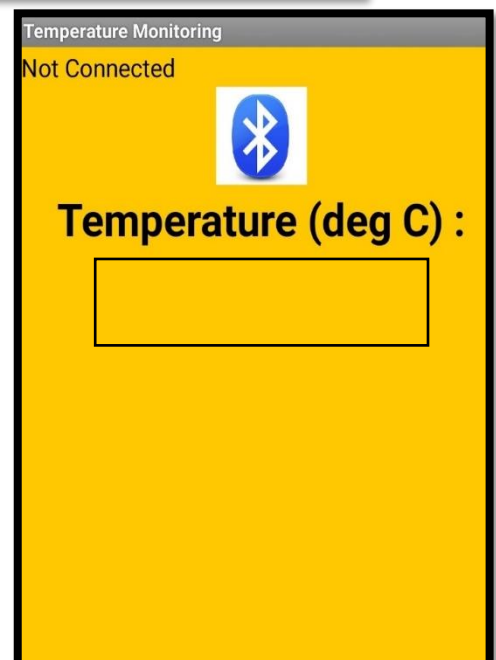


Figure III.17: L'application Bluetooth de Capteur T°



III.7 Conclusion :

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

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

Conclusion Générale

1.Généralités :

Afin de développer une application Androïde, vous devrez utiliser les composants mis à disposition par Google. 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.

2.Problèmes rencontrés :

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.

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.

Et développer la source d'énergie de ce projet pour qu'il s'intègre à tous égards et nous voulons également ajouter une eau pulvérisée pour éteindre les incendies.

- [1] C. Tavernier, « Arduino applications avancées ». Version Dunod.
- [2] <http://www.acm.uiuc.edu/sigbot/tutorials/>
- [3] <http://www.generationrobots.com/fr/152-arduino>
- [4] S.V.D.Reyvanth, G.Shirish, «PID controller using Arduino ».
- [5] A.SAI Kishore, J.Sandeep, A.Dadapeer, P.Sai Srinivas. « Three-phase inverter using Arduino».
- [6] Eskimon, Olyte « Arduino pour bien commencer en électronique et en programmation »
- [7] Jean- Noël, « livret Arduino en français », centre de ressources art sensitif
- [8] <http://arduino.cc/en/Main/ArduinoBoardUno>
- [9] <http://arduino.cc/fr/Main/Librairies>
- [10] ERIK Bartmann. « Le grand livre d'Arduino ». 2eme Edition,
- [11] <http://www.Datasheetcatalog.com>. + (datasheets for electronics components)
- [12] <https://fritzing.org/>
- [13] <http://ai2.appinventor.mit.edu/>
- [14] <http://blogpeda.ac-poitiers.fr>
- [15] <http://www.cadsoftusa.com/>
- [16] <http://www.labcenter.com/>

ملخص

يسمح لنا هذا المشروع بالغوص في عالم التواصل من أجل الوصول إلى إنشاء بطاقة إلكترونية تربط بين الهاتف الذكي وبطاقة الأردوينو والمنفذات.

الهدف الأولي من هذا المشروع هو إتقان لغة برمجة من أجل الوصول إلى إنشاء تطبيق أندرو يد قادر على إرسال أوامر من طرف مستخدم هذا التطبيق المثبت على الهاتف الذكي إلى محركات التيار المستمر و أجهزة كهربائية أخرى .

فهم مختلف برتوكولات الاتصال واستعمال المنفذ التسلسلي كمنفذ لإرسال البرنامج إلى بطاقة الأردوينو .

في هذا المشروع قمنا بتحقيق جهاز إتصال عن بعد عن طريق تقنية البلوتوث من أجل تحقيق نوعين من الأوامر، الأولى التحكم في معدات المنازل الذكية والثانية القيادة الذاتية لمركبة عن طريق تطبيق أندرويد مثبت على الهاتف الذكي .

الكلمات المفتاحية: تطبيق أندرويد ، بطاقة الأردوينو، المنفذات، لغة البرمجة، محركات التيار المستمر ، برتوكولات الإتصال.

Résumé

Ce projet nous permet de plonger dans le monde d'interfaçage afin d'arriver à réaliser une carte électronique qui communique entre (le système Androïde et la carte Arduino ainsi que les actionneurs).

L'objectif préliminaire est de manipuler un langage de programmation afin d'arriver à réaliser une application capable de transmettre des ordres émis par l'utilisateur vers des moteurs à courants continus et autre accessoires électriques via un smart phone.

Faire comprendre les différents protocoles de communication et extraire le port série comme étant un port de transfert du programme dans l'Arduino.

Mots Clés : système Androïde, carte Arduino, les actionneurs, langage de programmation, moteurs à courants continus, protocoles de communication.

Abstract

This project allows us to delve into the world of interfacing in order to arrive to create an electronic card that communicates between (Android system and the Arduino board and the actuators).

The preliminary goal is to manipulate a programming language in order to reach build an application able to transmit commands emitted by the user toward direct current motors and electrical equipment via a smart phone.

Understanding of the different communication protocols and extracts the serial port as a port for transferring program to Arduino board.

We performed a remote control device through a Bluetooth accessory to provide two types of control; the first is the home automation control and the second is self-guiding vehicles by intervention of an application in smart phone.

Key words: Android system, Adriano board, actuators, programming language, direct current motor, communication protocols.