République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université De Medea
Faculté de technologie
Département de Génie Mécanique

Mémoire de fin d'étude

En vue de l'obtention du diplôme de Master en génie mécanique

***Option* : mécatronique**

Thème

Commande et mesure de vitesse d'un moteur DC par Arduino avec application androïd

Présenté par:

Mr: Hourif mohamed mokhtar.

Dirigé par:

Mr: kermezli tayeb

Année universitaire 2021/2022

Remerciements

Mes remerciements vont tout d'abord à dieu tout puissant pour la volonté,la santé et la patience qu'il m'a donné durant toutes ces longues années.

Je tien à exprimer mes profondes gratitudes à mon promoteur **M^r^ kermezli tayeb**, pour avoir accepté de diriger ce travail. Je lui témoigne toute ma reconnaissance pour ses conseils, ses orientations et sa patience.

Mes sincères remerciements s'adressent aux membres de jury pour avoir accepté d'examiner ce travail.

Il m'est particulièrement agréable d'exprimer ici ma reconnaissance envers tous ceux qui ont rendu possible ce travail.

DÉDICACES

A la mémoire de ma très cher Sœur «……..».

A mes chers parents, pour leur patience, leur soutien, leur confiance, leurs encouragements et leurs sacrifices.

Je ne connais pas de termes assez forts pour remercier celui qui n'a pas cessé de m'encourager à réalisé ce travail «……».

Qu'il me soit permis d'adresser une dédicace spéciale à ma très chere Soeur «Sousou» et mes deux frères.

A notre adorable fille «Tinhinane».

Toute la famille

A mes amis qui m'ont soutenu pendant cette magnifique expérience et à toute la promotion génie mécanique 2022 ainsi que tous ceux qui ont contribuéde prés ou de loin à la réalisation de ce travail.

Hourif mohamed mokhtar**.**

الملخص:

تطور تكنولوجيا يؤدي إلى استخدام اآلالت التي تتطلب سرعات دقيقة ومتغيرة على سبيل المثال قطار فائق السرعة تكمن الميزة الرئيسية لآلات التيار المباشر في تكيفها البسيط مع وسائل ضبط أو تغيير سرعتها وعزم الدوران واتجا الدوران بالإضافة إلى اتصالها المباشر بمصدر للطاقة (بطاريات المركب ، البطاريات ، وما إلى ذلك) على الرغم من مشكلتها الرئيسية التي تنشأ على مستوى الاتصال بين الفرشاة أو "الكربون" والمجمع الدوار ، لا يزال من الممكن استخدام محرك التيار المباشر في العديد من التطبيقات.

في هذا المشروع ، أدركنا نظام التحكم في محرك DC وأيضا نظاما يكتشف عدد الدورات وسيتم تحويله إلى سرعة ثم إرساله إلى واجهة Android على هاتفنا الذكي ، ويتم عرض قيمة المحرك ويمكن تعديله من خلال العمل على شريط تمرير التطبيق.

أخيرا ، يتم الانتهاء من كل شيء من خلال الإدراك العملي للنموذج.

الكلمة الافتتاحية: دراسة كيفية التحكم عن بعد في محرك بالتيار المستمر. والكشف عن عدد الدورات ثم ارساله الى

الهاتف الذكي.

بالستعمال الأحهزة التالية: محرك بالتيار المستمر. لوحة الاردوينوا والكابتور والهاتف الذكي والبلوتوث

Sommaire:

Le développement de la technologie conduisant à l'utilisation de machines nécessitant des vitesses précises et variables, par exemple un train à grande vitesse, le principal avantage des machines à courant continu réside dans leur simple adaptation aux moyens de réglage ou de modification de leur vitesse, de leur couple et de leur sens de rotation. rotation ainsi que leur raccordement direct à une source d'alimentation (batteries de cuve, batteries, etc. it) Malgré son principal problème se situant au niveau du contact entre le balai ou "charbon" et le collecteur rotatif, un moteur à courant continu peut encore être utilisé dans de nombreuses applications. Dans ce projet, nous avons réalisé un système de contrôle de moteur à courant continu et également un système qui détecte le nombre de tours et le convertira en vitesse puis l'enverra à l'interface Android sur notre smartphone, la valeur du moteur est affichée et peut être modifiée en travaillant sur le curseur de l'application. Enfin, le tout est complété par la réalisation pratique du modèle. Keynote : Une étude sur la façon de contrôler à distance un moteur à courant continu. Et détecter le nombre de sessions puis l'envoyer à le smartphone. Utilisation des appareils suivants : Moteur à courant continu. Carte Arduino, capteur, smartphone, bluetooth

Summary:

The development of technology leading to the use of machines that require precise and variable speeds, for example a high-speed train, the main advantage of DC machines lies in their simple adaptation to the means of adjusting or changing their speed, torque and direction of rotation as well as their direct connection to a power source (vessel batteries, batteries, etc. it) Despite its main problem arising at the level of contact between the brush or "carbon" and the rotary collector, a DC motor can still be used in many applications. In this project, we realized a DC motor control system and also a system that detects the number of revolutions and will convert it to speed and then send it to the Android interface on our smartphone, the motor value is displayed and can be modified by working on the application slider. Finally, everything is completed by the practical realization of the model.
Keynote: A study of how to remotely control a DC motor. And detect the number of sessions and then send it to the smartphone. Using the following devices: DC motor. Arduino board,captor,smartphone,bluetooth

NTRODUCTION GENERALE

Ces dernières décennies l'électronique d'instrumentation a connues un développement vertigineux, dans le domaine de fabrication des composants, capteurs de précisions, appareils de mesure des laboratoires, ect….

Notre projet se place dans le cadre de la conception et de la fabrication d'un prototype de control et de mesure de vitesse d'un moteur à courant continu, Notre intérêt en tant que électronicien dans cette mémoire est de superviser le comportement du moteur de façon temporaire, c'est pour cette raison qu' on a utilisé une fourche optique qui va servir comme capteur de nombre de rotation c'est à dire à chaque passage il compte un tour, par la suite un microcontroleur est en service de faire tous le traitement nécéssaire en vue de comptabiliser le processus. On note ainsi que le capteur de fourche est de deux parties,une partie émettrice et une partie réceptrice. Leur principe de fonctionnement est simple, la première partie envoie un faisceau lumineux Infrarouge à destination de la seconde.

Lorsqu'il y a une coupure du faisseaux IR émanant de la première partie (cela peut être un objet,une tige, etc …) d'ou par la photodiode de la deuxième partie ne recoit rien ce qui montre la presence d'un objet. ensuite nous faisons appelle à la carte de devellopement ARDUINO. Sous ce nom Arduino se cachent non seulement du matériel mais aussi un logiciel. On parle alors de Physical Computing, qui désigne la construction de systèmes interactifs permettant de connecter le monde physique à celui des ordinateurs. Le monde dans lequel nous vivons est considéré comme un système analogique, alors que les ordinateurs agissent dans un environnement numérique ne connaissant que les états logiques 0 et 1.

C'est à nous, ensuite qu'il appartient d'établir une liaison entre ces deux mondes et de montrer par des actions et des faits de quoi nous sommes capables. C'est à partir de cette idée que projet a étè choisi. Notre mémoire est composé de quatre chapitres.

Le premier chapitre est consacré à la présentation de la théorie du moteur à courant continu.

Le second chapitre nous allons présenter une étude approfondie des différents composants et logiciels utilisés dans le projet.

Le troisième chapitre, dans ce chapitre nous allons présenter la partie pratique détaillée. C'est dans cette partie que nous allons voir le fonctionnement de la carte Arduino ainsi que la partie programmation.

Nous cloturons, notre travaille par une conclusion générale, dans la nous visons montré tous ce qu'on porra envisage perspective aux prochains étudiant

Chapitre I

Généralités sur les moteur à courant

I.1 Introduction

Jusqu'à un passé récent, les machine à courent continu ont été majoritairement utilisées dans les entrainements à vitesse variable. Ceci tient au fait que son fonctionnement est d'une grande simplicité, de même que sa modalisation, mais surtout à ses performances statiques est dynamique exceptionnelles.

Dans le cadre de ce chapitre, on présentera une brève description du moteur à courant continu, et des différents types existant dans la littérature. Une modélisation du moteur et ensuite illustrée en se basant sur des équations mathématique.

Une identification du modèle mathématique élaboré est ensuite faite pour le calcul des paramètres (électriques et mécaniques)

I.2 Construction d'un moteur à courant continu

Un moteur à courant continu est une machine électrique. Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique, d'où l'énergie électrique est transformée en énergie mécanique.

Un moteur à courant continu comprend quatre parties principales :

1) Le Stator composé d'un circuit magnétique comprenant une couronne annulaire massive ou feuilleté à laquelle sont fixés les pôles saillants, ces pôles sont de type, régulièrement alternés :

   - Les pôles inducteurs portent un ou plusieurs enroulements d'excitation alimentés en courant continu.
   - Les pôles auxiliaires ou commutation disposés entre les pôles inducteurs selon les axes interpolaires dits axes neutres, Ils servent à améliorer la commutation.

2) Le Rotor ou l'induit comporte un circuit magnétique feuilleté muni d'encoches régulièrement distribuées à la périphérie.
3) Le Collecteur est claveté sur l'arbre de l'induit, sur lequel frottent les balais. Il se compose de lame de cuivre de section trapézoïdale, portent chacune une ailette de connexion à l'enroulement de l'induit et isolées par du mica.

4) Les balais sont fixes, en charbon graphitique, appuient sur des lames diamétralement opposées du collecteur, ils sont immobiles, isolés électriquement du bâti de la machine.

1.2.1 Les différents types de moteurs à courant continu

On distingue deux types de moteurs à courant continu : 1.3.1 Les moteurs à inducteur à aimant permanent : Il n'y a pas de circuit inducteur, le flux inducteur est produit par un aimant permanent. Tous les moteurs à courant continu de faible puissance et les micromoteurs sont des moteurs à aimant permanent. Ils représentent maintenant la majorité des moteurs à courant continu. Ils sont très simples d'utilisation

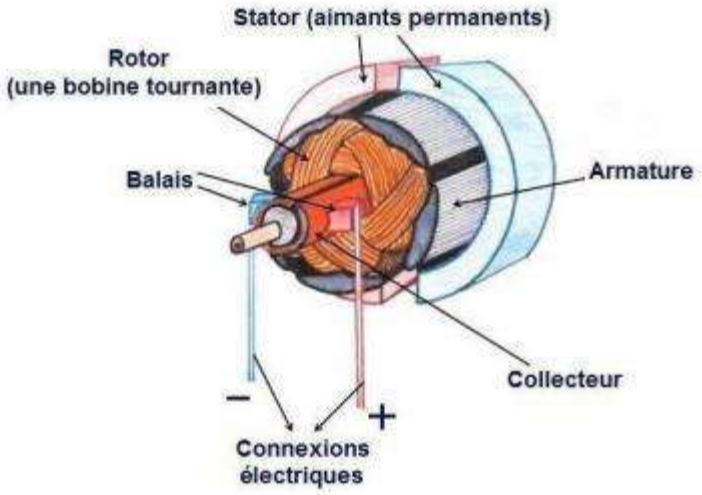


Figure 1.1 – Moteur à aimant permanent

1.2.2 Les moteurs à inducteur bobiné

Il existe 3 types différents de moteurs électriques qui sont classés en fonction du type d'excitation qui est employé, qui sont:

- Moteur à excitation série.
- Moteur à excitation shunt.
- Le moteur à excitation séparée.

1.2.3 Moteur série

Inducteur monté en série avec l'induit : L'induit et l'inducteur sont parcourus par le même courant[

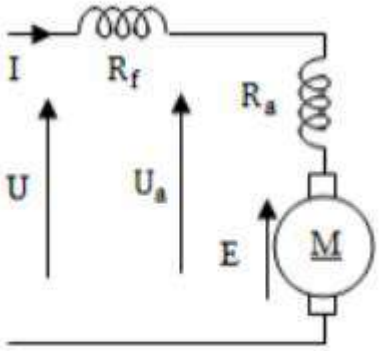


Figure 1..2 – Moteur à excitation série-

$$\begin{cases} U_a = (R_a + R_f)I + E \\ E = k\Omega\Phi \\ C_e = kI\Phi \end{cases} \quad avec\ \Phi = M_{af}I \Rightarrow \quad \begin{cases} E = k'\Omega I \\ C_e = k'I^2 \end{cases}$$

1.2.4 Moteur à excitation shunt (parallèle)

L'inducteur est monté en parallèle avec l'induit : les deux circuits sont soumis à la même tension[

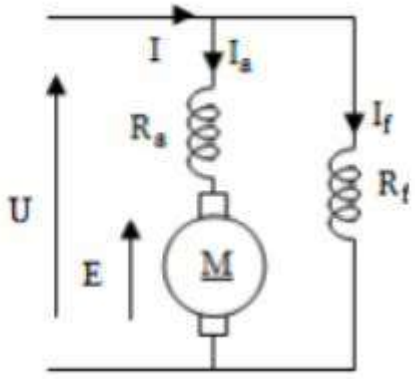


Figure 1.3 – moteur à excitation shunt-

$$\begin{cases} U = R_a I_a + E = R_f I_f \\ E = K\Omega\Phi = k' I_f \\ C_e = kI_a\Phi = k' I_a I_f \\ I = I_a + I_f \end{cases}$$

1.2.4Moteur à excitation séparée :

Le moteur étant alimenté, le bobinage induit sous tension Ua placé dans un champs magnétique inducteur B(If), est soumis à un couple donné par la loi de Laplace et par conséquent le moteur tourne à une vitesse Ω

$$\Omega = \frac{2\pi n}{60}$$

n : vitesse de rotation en (tr.min−1)

Ω : La vitesse de rotation en (rad.s−1)

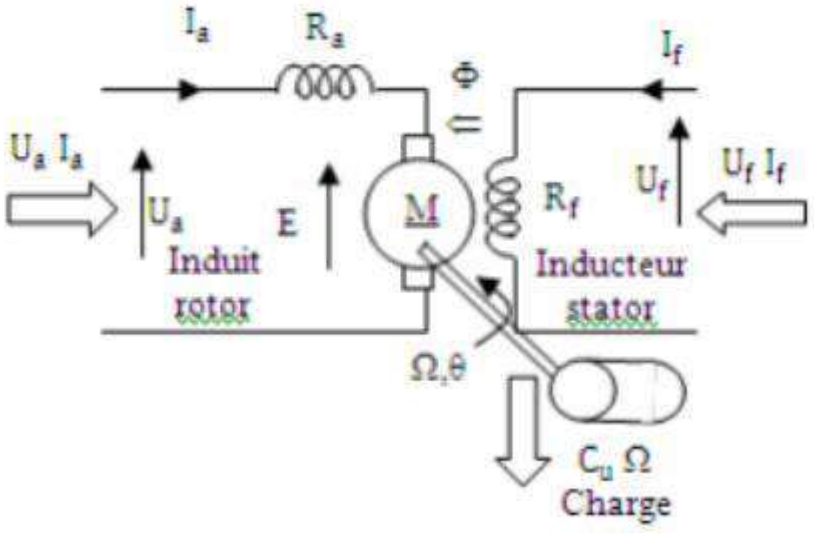


Figure 1..4– moteur à excitation independante

La relation de maille donne :

$$\begin{cases} Induit : & U_a = R_a I_a + E \\ Inducteur : & U_f = R_f I \end{cases}$$

$$I_a \gg I_f \qquad R_f \gg R_a$$

En pratique en charge :

Force électromotrice (fem) une bobine (induit) en mouvement placé dans un champ magnétique (inducteur) est le siège d'une force électromotrice motrice fem induite entre ces bornes donnée par la loi de Faraday :

$$E = k\Omega\Phi$$

K : constante qui dépend de la machine

Φ : flux maximum par pôle

le flux inducteur étant : Φ = **Maf If** Circuit magnétique non saturée

Le couple électromagnétique : Ce = KIaΦ

1.2.4 Caractéristiques de vitesse

D'après la fem :

$$E = k\Omega\phi = k'\Omega I_f = U_a - R_a I_a$$

La vitesse s''écrit :

$$\Omega = \frac{U_a - R_a I_a}{k' I_f} = \frac{U_a}{k' I_f} - k'' \frac{I_a}{I_f}$$

Caractéristique Ω(Ia) ,à Ua constant et If constant est une droite, si la charge augmente Ia augmente et Ω diminue (Ω varie peu avec la charge)

remarque si il Ya coupure du circuit inducteur If = 0 $\Omega \to \infty \Rightarrow$ emballement du moteur Ne jamais couper l'alimentation de l'inducteur d'un moteur à excitation indépendante si l'induit est sous tension

I.3 Identification des paramètres du moteur

I.3.1 Mesure des constantes électriques

I.3.1.1 La résistance de l'induit

On alimente le moteur avec une tension réduite, donnant un courant pas trop éloigné du courant nominal. On bloque soigneusement le rotor à l'aide d'une tige métallique. On relève les valeurs du courant d'induit qui correspondent aux valeurs des tensions d'alimentation. Puis, on calcule la moyenne des valeurs des résistances calculées par la relation suivante :

$$R = \frac{U}{I} \quad (I.9)$$

Tel que :

U : la tension d'induit.

I : le courant d'induit.

I.3.1.2 L'inductance de l'induit

Apres la mesure du courant du circuit rotorique (induit) , alimenté en petites tensions alternatives U, l'inductance de moteur est déterminée par une mesure de l'impédance du bobinage.

Cette dernière est exprimée par la formule suivante :

$$Z = \frac{U}{I} \quad (I.10)$$

$$\frac{1}{I} \quad \sqrt{\quad 2 \quad 2}$$

D'où $$L_a = \frac{\sqrt{Z_a - R_a}}{2\pi f} \quad (I.11)$$

Remarque : Cet essai est réalisé sous les conditions suivantes :

- La tension d'alimentation du circuit rotorique est réduite (environ 10% de la tension nominale) pour ne pas détériorer le moteur. Car il n'est pas prévu pour ce type de manipulation.
- Une fréquence élevée (1Khz), **[3]**.

I.3.1.3 La constante de temps électrique

Cette constante est calculée par la relation suivante :

$$\text{[illegible]} = \frac{\text{[illegible]}}{R} \quad (I.12)$$

I.3.2 Mesure des constantes mécaniques

I.3.2.1 La constante de vitesse K_e

On appel [illegible] le rapport liant la vitesse de rotation du moteur en fonction de la tension d'induit et pertes dans l'inducteur. Cette constante [illegible] est déterminée a partir d'un essai a vide effectué sur le moteur alimenté par la tension .

$$K_e = \frac{U}{\Omega} \quad (I.13)$$

Avec :

Ω: vitesse angulaire en [illegible] tel que : $\Omega = 2 * 3.14 * N/60$

N: vitesse du moteur en [illegible]

I.3.2.2 Le temps de ralentissement

On augmente la vitesse du moteur à une valeur constante, puis on la stabilise. A cet instant, on ouvre le circuit (on coupe toutes les alimentations en même temps) et on enregistre l'évolution de la vitesse par rapport au temps ave oscilloscope a mémoire).

I.3.2.3 Détermination du coefficient de frottement

On note par [illegible] le coefficient des frottements visqueux qui représentent les pertes à vide, et on note par C [illegible] le couple des frottements sec.

Afin de déterminer C [illegible] et [illegible] on fait deux essais a vitesses différentes : Pour Ω_1 on aura :

$$P_1 = C\Omega_1 + [\text{illegible}]^2$$

I.5.2.3.1 Le coefficient de frottement relatif a la charge

On utilise la génératrice comme une charge pour le moteur de coefficient de frottement f_1 relatif a la charge de la génératrice. Ce dernier est définit par l'expression suivante :

$$f_1 = \frac{P_{em}}{\Omega^2} = \frac{E_g I_g}{\Omega^2} \qquad (I.15)$$

Tel que :

P_{em} : Puissance électromagnétique de la génératrice.

E_g : force électromotrice de la génératrice.

I_g : courant délivré par la génératrice.

Ω: vitesse de rotation du groupe tournant.

$$E_g = R_g I_g + U_u$$

Avec :

$$U_u = R_u I_u$$

U_u et R_u tension de charge et sa résistance, respectivement.

R_g est la résistance de la génératrice mesurée a chaud.

Avec

$$E_{\text{}} = (R_{\text{}} + R_{\text{}}) \quad \text{(I. 16)}$$

On aura :

$$\text{}_{01} = \frac{E^{\blacksquare}}{\Omega^2 (R_{\text{}} +)} \quad \text{(I. 17)}$$

Enfin, les frottements totaux intervenant dans le moteur :

$$\text{}_{\text{}} = \text{}0 + \text{}_{01} \quad \text{(I. 18)}$$

I.3.2.4 Calcul du gain du capteur de vitesse k_v

Le gain du capteur de vitesse, $k_v$se mesure par le rapport de la tension $\text{}_{\text{}}$mesurée a la sortie du capteur de vitesse (génératrice tachymétrie) et de la vitesse réelle de rotation du moteur, **[5]**.

$$k_v = \frac{\text{}_{\text{}}}{\Omega} \quad \text{(I. 19)}$$

I.4 Simulation en boucle ouverte du MCC

Le moteur à courant continu choisi est un moteur de la société *Maxon* de référence *F2260/813* dont les caractéristiques techniques sont données par le tableau I.1 **[6].**

Tension nominale	24 *V*
Vitesse à la tension nominale	3820 *tr/min*
Tension maximale	32 *V*
Vitesse limite	5000 *tr/min*
Courant maximal à vide	22 *mA*
Courant maximal permanent	3.3 *A*
Courant de démarrage maximal	12.6 *A*
Couple permanant maximale	200 *mN.m*
Résistance aux bornes	1.91 *Ω (Ohm)*
Inductance	0.62 *mH*
Constante de vitesse	60.3 *mV/(tr/min)*
Constante du couple	60.3 $mN.m.A^{-1}$
Coefficient de frottements visqueux	$2.5\ 10^{-5}$
Inertie du moteur	$10^{-4}\ Kg.m^2$

Tableaux I.1 : *Caractéristiques techniques du MCC Maxon F2260/813.*

D’après la (figure I.8), on voit bien que la réponse en position est une rampe, ce qui signifie que le système dans un asservissement en position est instable, d’où il nécessite une régulation pour stabiliser la réponse en position et d’améliorer les performances en vitesse.

I.5 Conclusion

Pour un moteur électrique à courant continu, on a constaté que la réponse en position est instable, hors que la réponse en vitesse change considérablement avec la tension de l'entrée, d'où, on voit bien la nécessité d'une commande pour stabiliser le système MCC qui ne l'est pas, et d'a.

Chapitre II
Etude Matériels et logiciels

II.1 Introduction

L'objectif de notre travail consiste à l'étude et réalisation d'un système de contrôle d'accès à base de la technologie RFID et la saisie d'un code personnel d'identification pour la reconnaissance des individus. Dans ce chapitre, nous présentons l'ensemble des composants nécessaire pour notre projet :

- Unité de commande et de traitement : Arduino UNO.
- Capteur optique à fourche .
- Moteur DC.

II. 2 Materièls et composants du projet:

II. 2. 1 La carte Arduino UNO

C'est une petite carte électronique (5,33 × 6,85 cm) équipée d'un microcontrôleur, ce dernier permet, à partir des d'événements détectés par des capteurs, de programmer et de commander des actionneurs ; la carte Arduino est donc une interface programmable [23].



Figure II. 1 : La carte Arduino UNO

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATMega328. Le microcontrôleur ATMega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C [24].

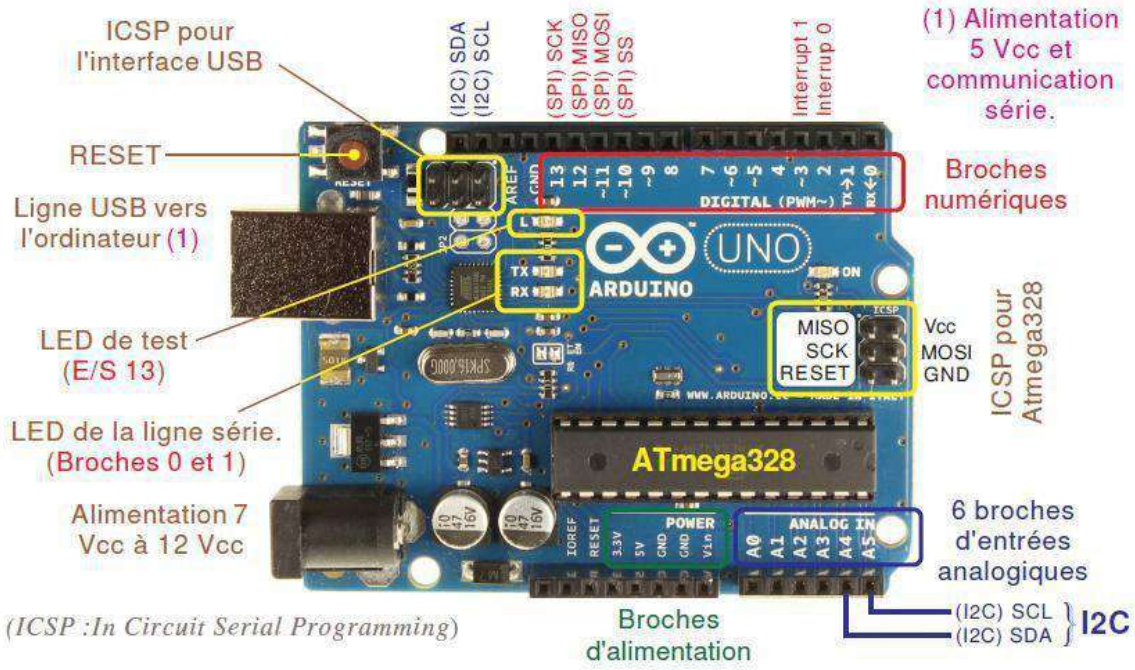


Figure II. 2 : Les E/S d'une carte Arduino UNO

II. 2. 1. 1. Caractéristiques techniques de la carte Arduino UNO [23]

- Microcontrôleur : ATmega328.
- Fréquence horloge : 16 MHz.
- Tension d'alimentation interne : 5Vcc.
- Tension d'alimentation externe recommandée : 7-12Vcc. (Limites : 6-20 Vcc)
- Courant max sur la sortie 3,3V généré par le régulateur interne : 50mA.
- Entrées/sorties binaires : 14 broches.
- Courant MAX par broches en sortie : 40 mA. (85 mA en court-circuit)
- Courant MAX cumulé par les broches en sorties : 200 mA. (Soit 14 mA en moyenne)
- Les E/S binaires 0 et 1 sont mobilisées par le dialogue sur la ligne série.
- S0 pour RX et S1 pour TX. Chaque broche est reliée à une LED via R = 1kΩ.
- Les E/S binaires 3, 5, 6, 9, 10, et 11 sont dédiées au mode PWM.
- L'E/S 13 est reliée sur la carte à la LED de test via une résistance de 1kΩ.
- Entrées analogiques : 6, le niveau logique maximal doit être de +5Vcc.
- Mémoire Flash 32 KB dont 0.5 KB utilisée par le Boot loader.
- Mémoire SRAM 2 KB, Mémoire EEPROM 1 KB.

- La carte s'interface au PC par l'intermédiaire de sa prise USB.
- La carte s'alimente par le jack d'alimentation**.**

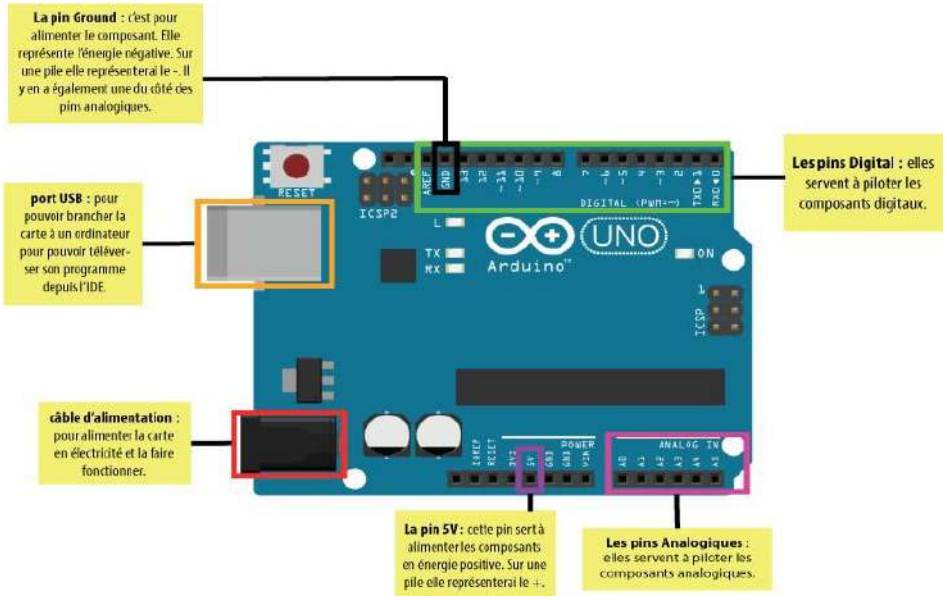


Figure II. 3 : Anatomie d'une carte Arduino UNO

II. 2. 1. 2. Schéma simplifié de la carte Arduino UNO

Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleur ATMega328, ce dernier est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit dont la programmation peut être réalisée en langage C [25]**.**

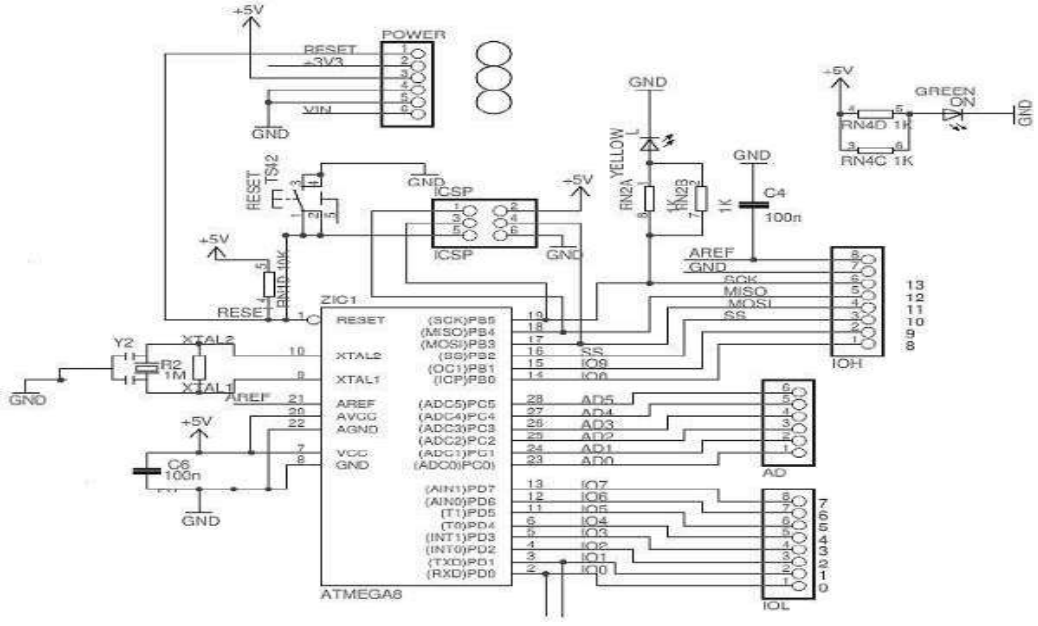


Figure II. 4 : Schéma simplifié de la carte Arduino UNO

Le ATMEL ATMega328 est un microcontrôleur ATMEL de la famille AVR 8bits [25]**,** il se caractérise par [25] :

- **FLASH :** mémoire programme de 32Ko
- **SRAM :** données (volatiles) de 2Ko
- **EEPROM :** données (non volatiles) de 1Ko
- **Digital I/O (E/S Tout Ou Rien) :**3 ports PortB, PortC, PortD (soit 23 broches I/O).
- **Timers/Counters :** Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits) Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB).
- **Plusieurs broches multi-fonctions :** certaines broches peuvent avoir plusieurs fonctions différentes choisies par programmation.
- **PWM :** 6 broches OC0A(PD6), OC0B(PD5), 0C1A(PB1), OC1B(PB3), OC2A(PB3), OC2B(PD3)
- **Analog to Digital Converter (résolution 10bits) :** 6 entrées multiplexées ADC0(PC0) à ADC5(PC5).
- **Gestion bus I2C (TWI TwoWire Interface) :** le bus est exploité via les broches **SDA(PC5)/SCL(PC4).**
- **Port série (USART) :** émission/réception série via les broches TXD(PD1)/RXD(PD0)
- **Comparateur Analogique :** broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption Watch dog Timer programmable.
- Gestion d'interruptions (24 sources possibles (interruptvectors)) : en résumé
    - Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3)
    - Interruptions sur changement d'état des broches PCINT0 à PCINT23
    - Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables)
    - Interruption liée au comparateur analogique
    - Interruption de fin de conversion ADC
    - Interruptions du port série USART Interruption du bus TWI (I2C).

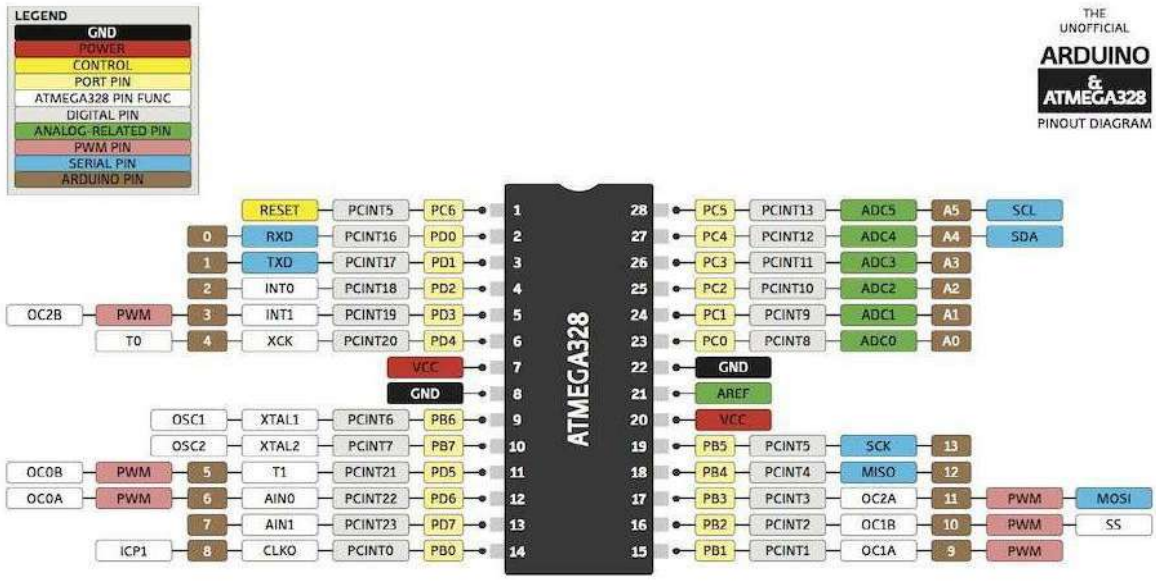


Figure II. 5 : Brochage de Microcontrôleur ATMega328

II. 2. 1. 3. Alimentation de la carte ARDUINO UNO

La carte Arduino UNO peut être alimentée via la connexion USB ou avec une alimentation externe, la source d'alimentation est automatiquement sélectionnée. Une alimentation externe peut provenir soit d'un adaptateur AC-DC ou d'une batterie. L'adaptateur peut être connecté en branchant une prise 2.1mm dans la prise d'alimentation de la carte ou à partir d'une batterie connectée dans le pin (ou broche) GND et Vin (alimentation externe) [26]**.**

- Le processeur peut fonctionner sur une alimentation externe de 6 à 20 V. Cependant, si la tension est inférieure à 7V, le pin 5V peut fournir moins de 5 V et le processeur peut devenir instable.
- Si la tension est supérieure à 12V, le régulateur de tension peut surchauffer et endommager la carte, la plage recommandée est de 7 à 12 V [27]**.**

II.2.2 Capteurs optique :

Un détecteur photoélectrique réalise la détection d'une cible, qui peut être un objet ou une personne, au moyen d'un faisceau lumineux. Ses deux constituants de base sont donc un

émetteur à diode électroluminescente et un récepteur à phototransistor de lumière. La détection est effective quand la cible pénètre dans le faisceau lumineux et modifie suffisamment la quantité de lumière reçue par le récepteur pour provoquer un changement d'état de la sortie. Elle est réalisée selon deux procédés :

- blocage du faisceau par la cible.

- renvoi du faisceau sur le récepteur par la cible. Le faisceau lumineux émis comporte deux zones :

- une zone de fonctionnement recommandée dans laquelle l'intensité du faisceau est suffisamment élevée pour assurer une détection normale. Selon le système utilisé, barrage, reflex ou proximité, le récepteur, le réflecteur ou la cible doivent être situés dans cette zone

- une zone dans laquelle l'intensité du faisceau n'est plus suffisante pour garantir une détection fiable.

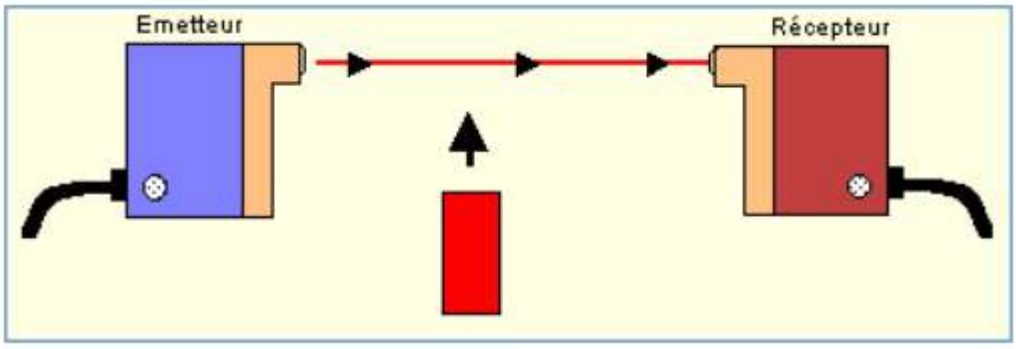
II.2.2.1 Les avantages :

• détection d'objets de toutes formes et de matériaux de toutes natures.

• détection à très grande distance.

II.2.2.2 Différents types de capteur optique :

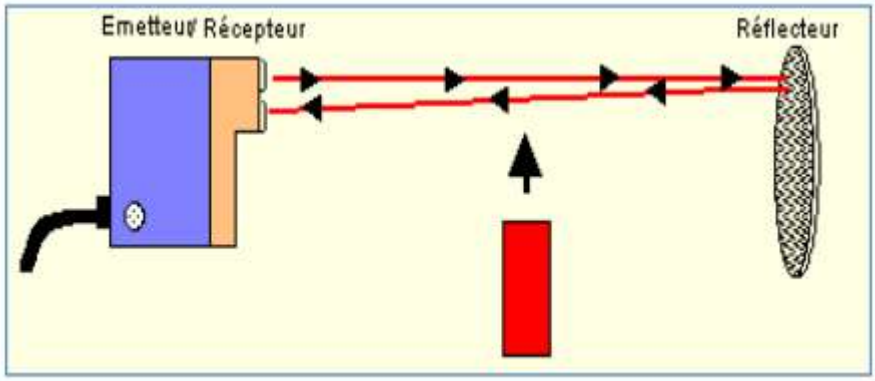
a. Système barrage :



FigureII.6 :Système barrage de capteur optique

L'émetteur et le récepteur sont dans deux boîtiers différents montés en vis-à-vis.L'objet passe entre les deux. Il est détecté dès qu'il vient couper le faisceau optique

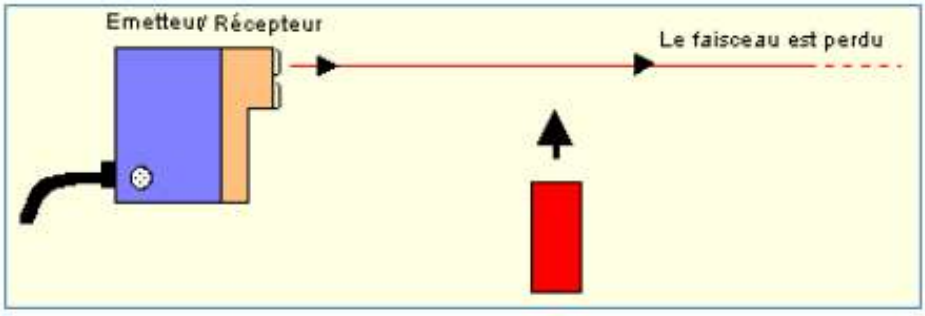
b. Système reflex :



FigureII.7 : Système reflex de capteur optique

L'émetteur et le récepteur sont dans le même boîtier. La lumière émise est renvoyée par un réflecteur. L'objet est détecté lorsqu'il coupe le faisceau optique.

c. Système proximité :



FigureII.8**:** Système proximité de capteur optique

Comme pour le système reflex, émetteur et récepteur sont regroupés dans unmême boîtier. Le faisceau lumineux, émis en infrarouge, est renvoyé vers le récepteur par tout objet suffisamment réfléchissant qui pénètre dans la zone de détection

d. Capteur infrarouge (TCRT5000):

Une fourche optique est un capteur constituée d'un dispositif d'émission de lumière infrarouge (LED) et de réception (phototransistor) placés en vis à vis.

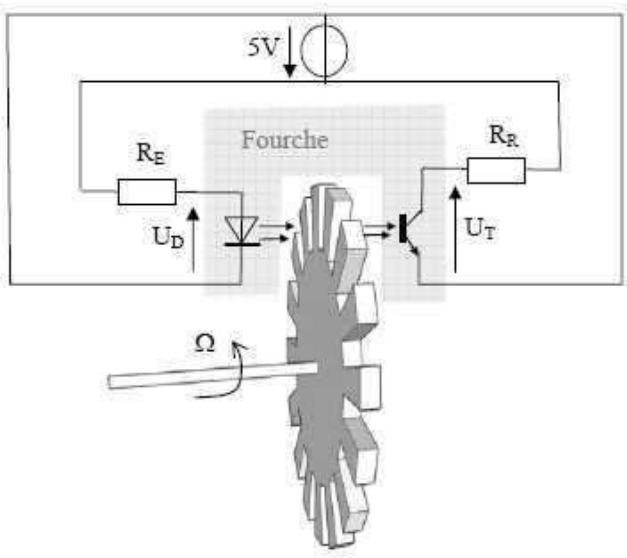


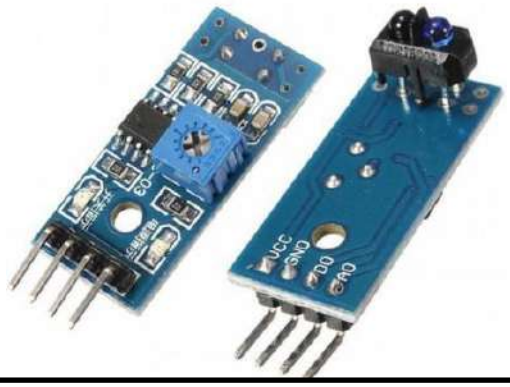
FIGURE II.9 principe du capteur à fourche

Ce capteur détecte le passage d'objets opaques à travers la fourche. Ces objets doivent donc être de petite taille, ou bien porter une extension fine. Il n'est cependant pas adapté aux mouvements forts lents, un peu plus complexe à insérer dans un mécanisme.

Il peut détecter des marques métalliques placées à des endroits critiques sur une pellicule cinéma. Il peut donner la localisation d'une roue dentée dans un mécanisme, ou d'une courroie, si elle est équipée d'un signe adéquat. Il permet de contrôler un mécanisme dans le temps en détectant des marqueurs de position.

Exemple de capteur IR (TCRT5000**) :**

Le module TCRT5000 de mois et plus que les simples capteurs infrarouges appelé détecteur d'approche ou détecteur de mouvement est constitué de deux petites pièces à l'intérieur d'une boîte plastique où la première pièce (la première partie), responsable de l'émission infrarouge, la photodiode (LED) Pour recevoir infrarouge et phototransistor.

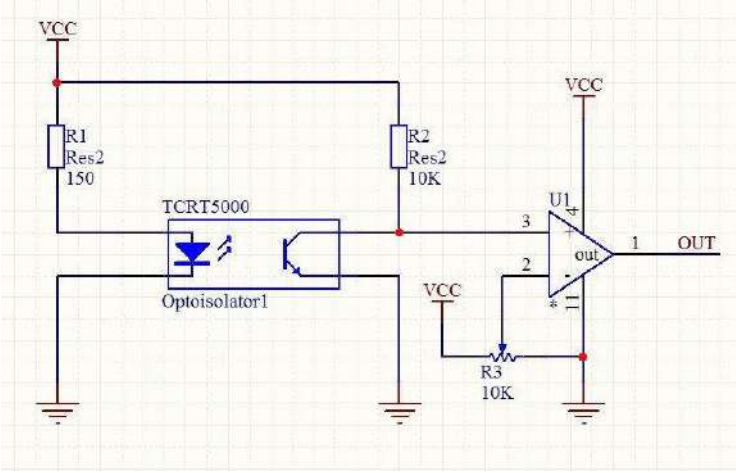


FigureII.10: Module infrarouge TCRT5000

-Principe de fonctionnement:

Le circuit de capteur est très simple où on roule la résistance de 100 ohms avec l'émetteur et la résistance de 20 kilo-ohms (Par exemple) à l'avenir de sorte que quand il y a une barrière à la sensibilité refléter l'infrarouge provenant de l'émetteur sur l'avenir et être

tension de sortie de 0 v soit lorsque l'absence d'une barrière, il y aura une réflexion de l'infrarouge et par conséquent Nous allons trouver une tension de 5 v sur la sortie en raison de la présence d'une grande résistance 20 k ohms(Par exemple).



FigureII.11: Circuit du module TCRT5000

- **Caractéristiques de module TCRT5000 :**

  - Tension de service : 3.3VDC à 5VDC
  - IC : LM393
  - Capteur : TCRT5000
  - Plage de fonctionnement : 1mm à 25mm
  - Dimensions : 31mm x 14mm / 1.22 po x 0.55 po

II.2.3 Module Bluetooth (HC-06):

II.2.3.1 Définition :

Le module Bluetooth HC-06 est un module Bluetooth SPP (Serial Port Protocol) facile à utiliser, conçu pour la configuration transparente de connexion en série sans fil. Sa communication se fait par série communication qui facilite l'interface avec le contrôleur ou le PC. HC-06 Bluetooth module fournit le mode de commutation entre le mode maître et esclave qui signifie qu'il peut utiliser ni recevoir ni transmettre de données.

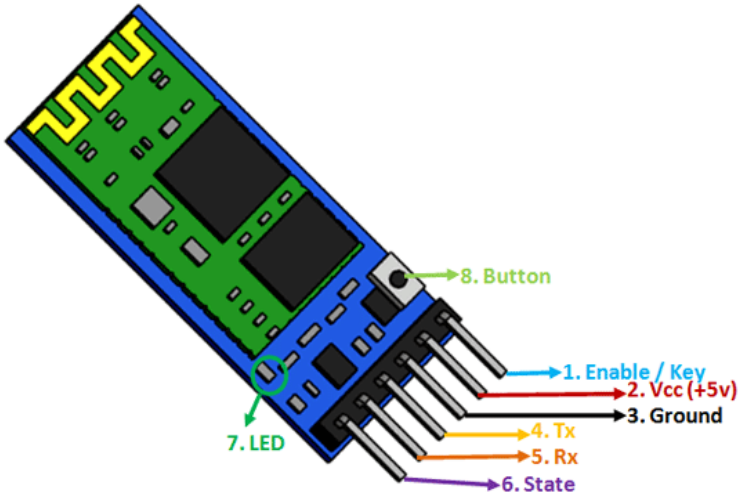
II.2.3.2 Fonctionnalités matérielles:

Sensibilité typique de -80dBm

- Jusqu'à +4dBm RF transmettre la puissance
- Fonctionnement de faible puissance 1,8 V, 1,8 à 3,6 V I/O

- Contrôle PIO
- Interface UART avec taux baud programmable
- Avec antenne intégrée
- Avec connecteur bord

II.2.4 Brochge:



FigureII.12: Brochage Bluetooth HC-05

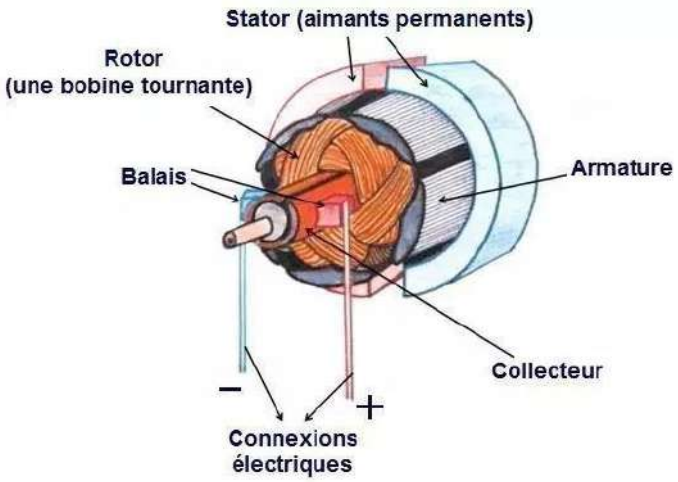
Pin	Description	Fonction
VCC	+5V	Connecté à +5V
GND	GROUND	Connecté à GROUND
TX	UART_TXD, Série Bluetooth envoi de signaux épingler	Transmet des données en série. Tout ce qui est reçu via Bluetooth sera donné par cette épinglette comme données de série
RX	UART_RXD, Série Bluetooth réception du signal épingler	Recevez des données de série. Toutes les données de série données à cette épinglette seront diffusées via Bluetooth
KEY	Commutateur de mode entrée	Cette Pin est utilisée pour basculer entre le mode données (réglez bas) et le mode de commande AT (réglez haut). Par défaut, il est en mode Données
STATE	L'état	La Pin d'état est connectée à la LED à bord, elle peut être utilisée comme rétroaction pour vérifier si Bluetooth fonctionne correctement.

Tableau II-2 : Configuration des Pins

II.2.5 Moteur à courant continu DC

Un moteur à courant continu est constitué de deux parties électriques : le stator et le rotor. Lorsqu'on alimente le moteur, il se crée une interaction magnétique qui met le moteur en mouvement. Lorsqu'on inverse le sens de la tension qui alimente le moteur, il tourne en sens inverse.

La figure suivante montre la constitution du moteur à courant continu.



FigureII.14 **moteur à courant continu**

II.2.5.1 Le stator

Le stator d'un moteur à courant continu est la partie fixe du moteur (statique = qui ne bouge pas). Le stator est aussi nommé l'inducteur, on fait passer un courant dans le bobinage du stator et c'est lui qui crée (induit) un champ magnétique. Le stator impose au rotor un champ magnétique. Le stator (inducteur) crée un champ magnétique B appelé champ inducteur, ou encore champ statorique. On le note Bs

Le stator peut être fait d'aimants permanents ou d'électroaimants qu'il faut alimenter. Dans ce cas, la relation électrique s'écrit :

$$Ue = Re.Ie$$

II.2.5.2 Le rotor

Le rotor est la partie en rotation du moteur. C'est lui qui tourne. Il est constitué du bobinage induit. Il faut alimenter cette bobine pour la transformer en électroaimant qui entrera en interaction avec le stator. Si on alimente pas le rotor, il ne serait l'objet d'aucune force et ne tournerait pas.

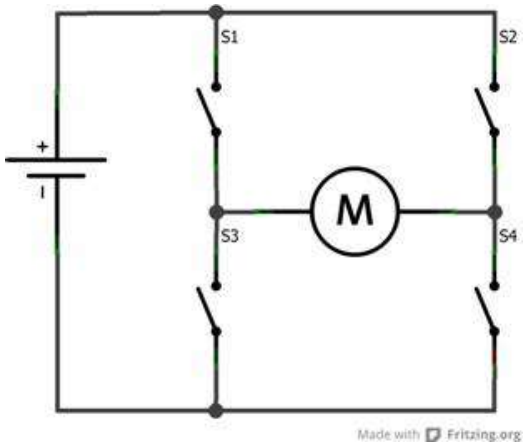
II.2.5.3 Commande du moteur dans les deux sens

Faire tourner un moteur c'est intéressant. Sans oublier de le faire tourner à la bonne vitesse.

II.2.5.4 Découverte du pont en H

Reprenons un dispositif de base avec un transistor que nous symboliserons ici par un interrupteur. Si ce dernier est activé le moteur tourne, si non le moteur est arrêté. Si on rajoute un deuxième transistor de l'autre côté du moteur rien ne va changer, mais il va falloir commander les deux transistors pour faire tourner le moteur. C'est aberrant, donc c'est pour cette raison que

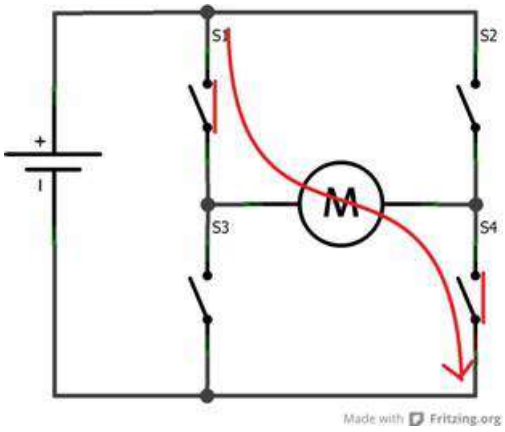
nous devons utiliser quatre transistors de manière à ce qu'on commande le moteur dans les deux sens comme le montre la figure suivant.



FigureII.15 **Le pont en H**

II.2.5.5 Pilotage dans le sens horaire

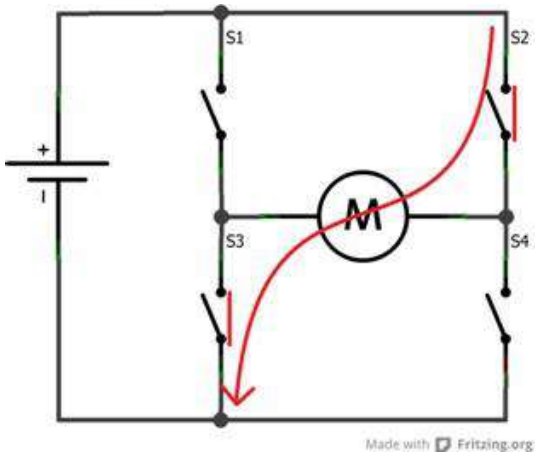
Le montage permet de piloter le moteur dans les deux sens de rotation. Le principe de fonctionnement est très simple, si on ferme les transistors 1 et 4 en laissant ouverts le 2 et le 3. Le courant passe de la gauche vers la droite.



FigureII.16 **Fonctionnement dans le sens horaire**

II.2.5.6 Pilotage dans le sens Anti-horaire

Si en revanche on fait le contraire (2 et 3 fermés et 1 et 4 ouverts), le courant ira dans l'autre sens ,ce qui va piloter le moteur dans l'autre sens.



FigureII.17 **Fonctionnement dans le sens anti-horaire**

II.2.5.7 Faire varier la vitesse du moteur à courant continu

Maintenant que nous avons les bases fondamentales pour faire tourner un moteur à courant continu, nous allons pouvoir régler la vitesse de rotation du moteur.

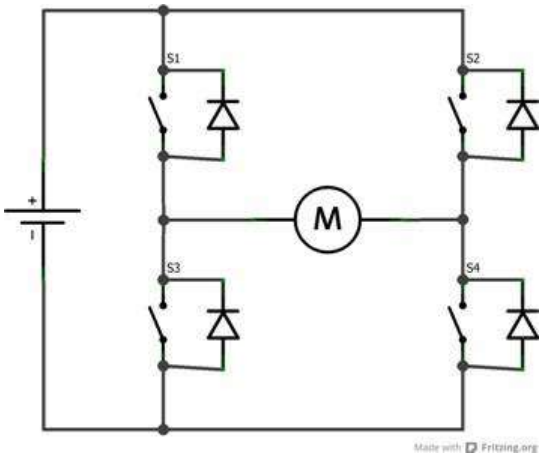
Un moteur à courant continu possède une relation directe entre sa tension d'alimentation et sa vitesse de rotation. En effet, plus la tension à ses bornes est élevée et plus son axe tournera rapidement.

Cependant si nous voulons contrôler le moteur par un microcontrôleur, ce dernier n'est capable de produire que des tensions de 0 ou 5V. En revanche, il peut faire sortir des tensions variables comprises entre 0 et 5V c'est ce qu'on appelle sorties PWM sur quelques broches du microcontrôleur. Nous sommes en mesure de produire un signal carré dont le rapport cyclique est variable. Et grâce à cela, nous obtenons une tension moyenne (comprise entre 0 et 5V) en sortie du microcontroleur.

En utilisant la PWM, on va générer une tension par impulsions plus ou moins grandes. Ce signal va commander les transistors du driver L293 qui va à son tour commander le moteur.

II.2.5.8 Les diodes de roue libre

En vue de protéger un transistor des parasites ou lors du freinage électronique du moteur, nous plaçons une diode ainsi appelé diode roue libre. Dans le cas présent, cette diode devra être en parallèle aux bornes du transistor comme nous voyons sur le schéma ci-dessous. Ici nous avons quatre transistors, nous utiliserons donc quatre diodes que nous placerons sur chaque transistor. Ainsi, le courant trouvera toujours un moyen de passer sans risquer de forcer le passage dans les transistors en les grillant.



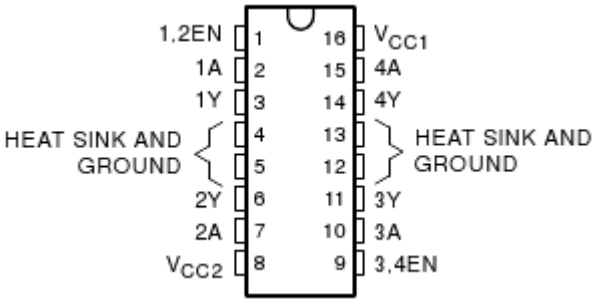
FigureII.18 **Pont en H avec ses diodes de protection**

Afin d'éviter de travailler avec les branchements des transistors et leur logique de contrôle, Nous allons voir comment peut-on s'en passer pour les remplacer par d'autre circuit de commande. C'est le cas par exemple d'utiliser le fameux driver L293.

II.2.6 Le L293(D)

Dans la gamme de circuit intégré "H Bridge Driver" disponible, le plus courant est le L293D. C'est un circuit intégré monolitique, à voltage important assurant un courant plus au moinspuissnat. Cela veut dire que ce circuit intégré peut être utilisé pour des moteurs DC et alimentation jusqu'a 36 Volts. le circuit peut fournir un maximum de 600mA par canal

II.2.6.1 Brochage



FigureII.19 Le circuit L293

II.2.6.2 Caractéristiques électriques :

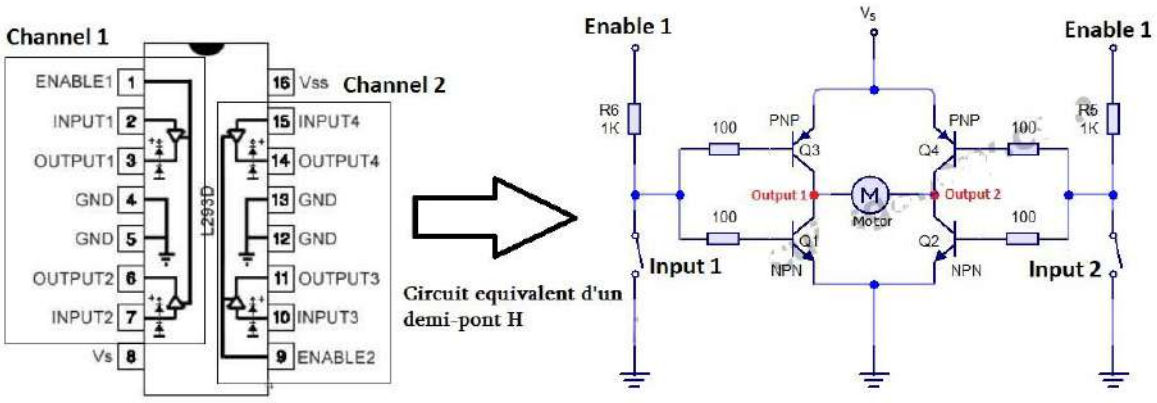
Total number of pins = 16
Ic Operating voltage = 5 – 7 v

Input voltage = 4.5 – 36 v

Output voltage = 4.5 – 36 v
Continuous Output Current = 600 mA or 0.6 A

II.2.6.3 Structure interne :

La structure interne du driver L293 est équivalente à deux canaux en pont H dont chacun contient quatre transistors comme le montre la figure suivante.



FigureII.20 - **construction interne du circuit L293D**

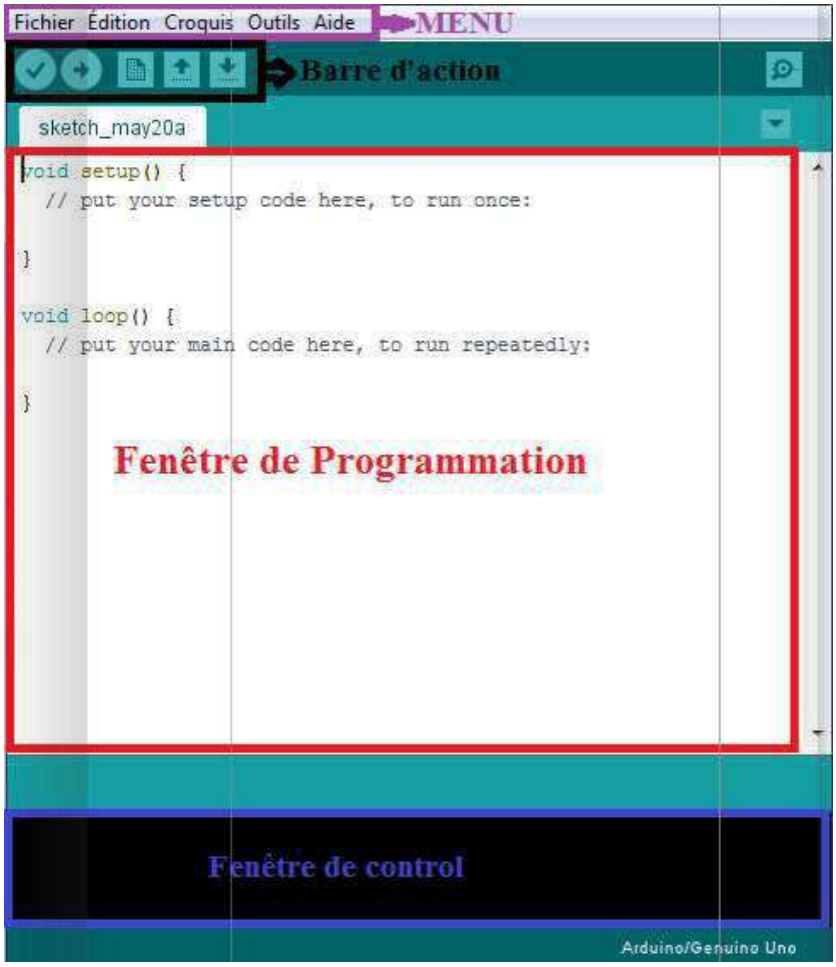
II.3 Etude des Logiciels

II.3.1 Logiciel (IDE Arduino)

L'environnement de programmation Arduino (IDE en anglais) est une application écrite en Java. L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte. Le logiciel va nous permettre de programmer la carte Arduino, il nous offre une multitude de fonctionnalités.

La programmation se fait dans le logiciel ARDUINO IDE, le logiciel a les fonctions suivantes :

- Ecrire le programme.
- Compiler pour tester le programme.
- Connecter et transférer le programme à la carte.
- Communiquer avec la carte.



FigureII.21 Logiciel ARDUINO IDE

La structure des programmes Arduino est un peu particulier, en apparence, des structures habituelles du langage C. La syntaxe est la même qu'en langage C.

Au début du programme, la déclaration des librairies utilisée par le programme est obligatoire.

1. **Menu :** Le menu comprend :
   - **Fichier** : pour créer, sauvegarder en spécifiant la destination, et d'appeler un programme.
   - **Edition** : Pour couper, copier, coller, supprimer, sélectionner,…etc. ;
   - **Croquis** : regroupe les fichiers réalisés.
   - **Outils** : pour spécifier le type de la carte, le port série, formater, recharger

et réparer l'encodage, graver la séquence d'initialisation, de la carte branchée sur l'ordinateur.

2. **Barre d'action :**



FigureII.22 Le Menu

- Bouton 1 : Compilation : vérifier le programme
- Bouton 2 : Transférer le programme à l'ARDUINO
- Bouton 3 : Créer un nouveau fichier
- Bouton 4 : Ouvrir un fichier existant
- Bouton 5 : Enregistrer un fichier

3. **Fenêtre de Programmation :** c'est l'éditeur de texte pour écrire le programme, chaque programme obéit à des instructions structurées pour savoir que le programme est bien écrit on va le compiler pour éviter les erreurs de syntaxes.

4. **fenêtre de control :** La barre des erreurs affiche les erreurs faites au cours du programme, comme l'oubli d'un point-virgule, le manque d'une accolade ou toute autre erreur dans les instructions. [5]

II.3.1.1 Structure du programme:

Le programme se déroule de la façon suivante :

1. Prise en compte des instructions de la partie déclarative
2. Exécution de la partie configuration (fonction setup( )),
3. Exécution de la boucle sans fin (fonction loop ( )): le code compris dans la boucle sans fin est execute indéfiniment.

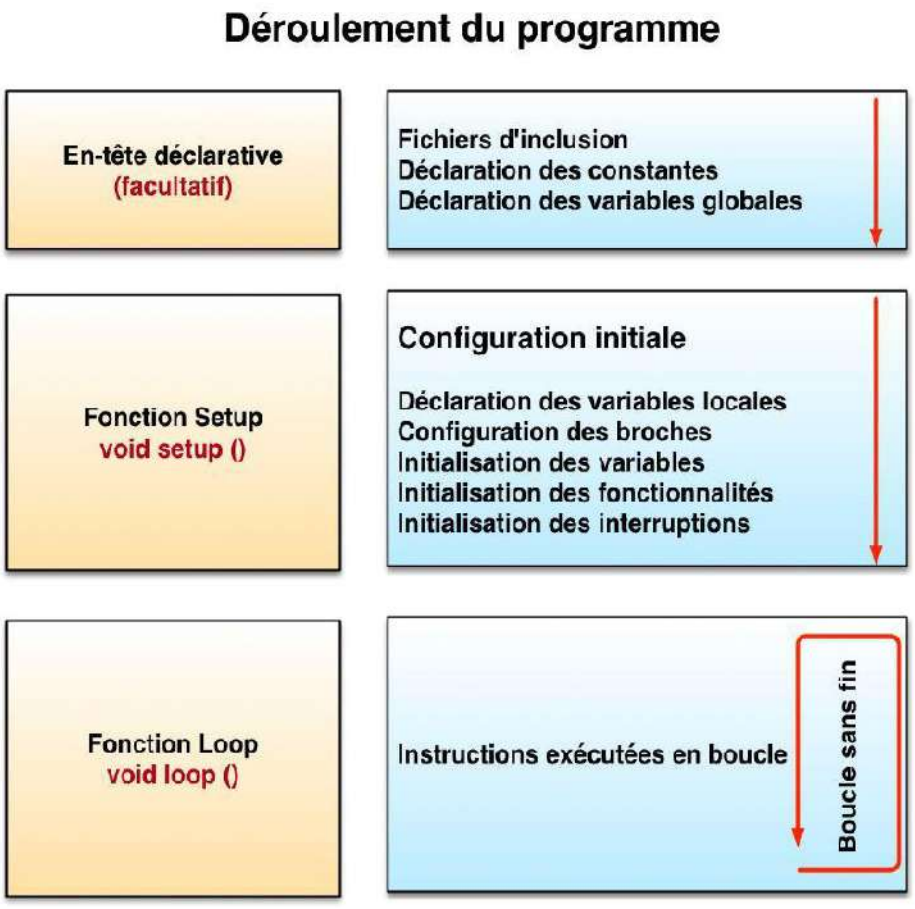


Figure II.23 Déroulement du programme [5]

II.3.1.2 Langage de programmation :

Le langage de programmation de l'IDE Arduino est en langage C/C++, il possède un jeu d'instruction très riche. Ces instructions décrivent :

- Les variables :

  peuvent être numériques (byte, int, float, word, ...),

  logiques (booleen, …),

  Sous forme de tableaux (array),

  Caractères ou chaines de caractères (char, string),

  Constantes particulières (True/False, HIGH/LOW) …etc.

- Les points virgules (;): Les points virgules terminent les instructions.

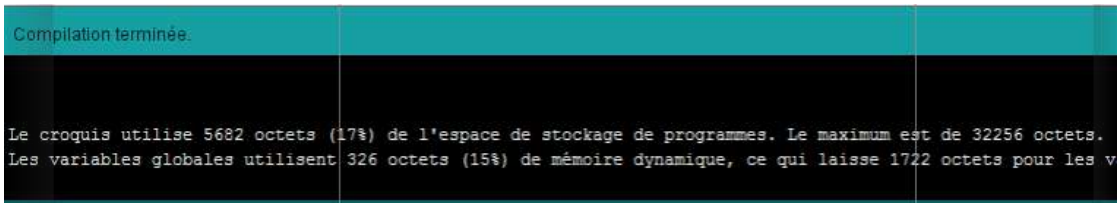
- Les accolades ({ }): Les accolades sont les "conteneurs" du code du programme. Elles sont propres aux fonctions, aux conditions et aux boucles.

- Les commentaires (//): Les commentaires sont des lignes de codes qui seront ignorées par le programme. Elles ne sont pas prises en considération lors de l'exécution du programme.

- Les fonctions arithmétiques et mathématiques : comme les fameuses quatre opérations arithmétiques (+, -, * et /) simples ou composées, les fonctions mathématiques (abs, min, max,..) et trigonométriques (cos, sin, …).
- Les opérateurs logiques (&&, = !,…) et les opérateurs de comparaisons (=, <,>, ..).
- Les structures de contrôle comme les boucles (for, while,..), les prises de
- décision (if-else,…), les sauts (breack, goto, continue,…).
- Gestions du temps (delay) et des entrées /sorties numérique (pinMode, digitalWrite/Read) ou analogique (analogRead/Write).
- Gestion du port série (Serial.begin/.end/.available/.read/.print,…etc).[6]

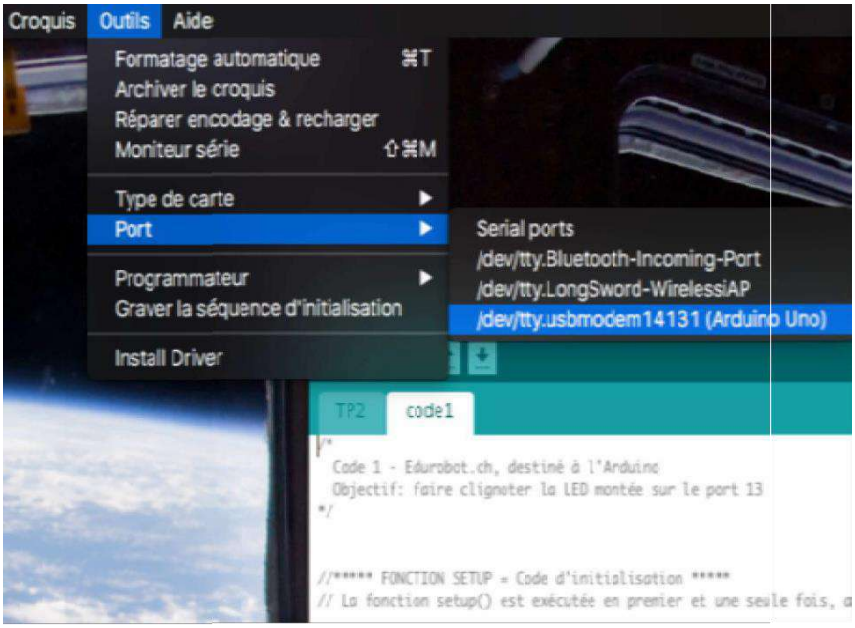
II.3.1.3 Compilation et Injection du programme [14] :

Une fois le code écrit (ou collé) dans la fenêtre de programmation terminé, on passe au test, en appuyant sur le bouton **Compiler**, une barre de progression s'affiche au-dessus de la barre des erreurs, s'il n y a pas d'erreur, on verra s'afficher le message «**Compilation terminée** », suivi de la taille du programme. Dans le cas contraire, un message d'erreur s'affiche.

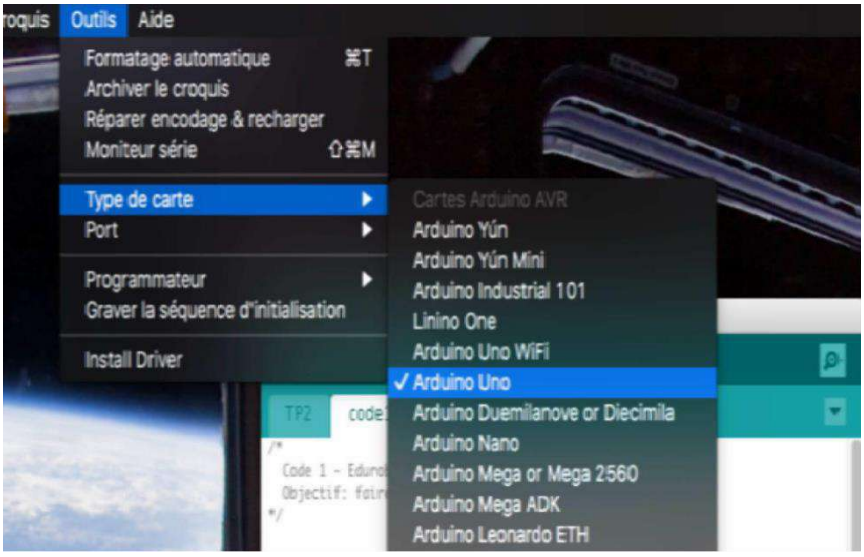


FigureII.24 Compilation

Quand la compilation terminée, le programme sera sauvegardé, donc il faut le transférer sur l'Arduino. Pour cela, après avoir connecté l'Arduino à l'ordinateur, il faut sélectionner le port (try_usbmodemXXXXX) et le type de
carte (ArduinoUno, dans notre cas). Ces deux réglages sont Outils. dans le menu Comme à titre d'exemple ces deux figures suivantes :



FigureII.25 Sélection du port



FigureII.26 Sélection le type de carte

Cliquer enfin sur le bouton « Télé verser » plus haut sur la barre d'action, le message «Compilation du croquis » s'affiche au-dessus de la barre des erreurs, ce message signifie que le programme est en train d'être compile, puis, on verra« Télé versement… », qui veut dire que le programme se charge dans la mémoire de la carte.

II.3.2 Logiciel MIT App-Inventor

II.3.2.1 Le système Android

Android 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 utilitaires et les périphériques nécessaires à un smart phone. Il est optimisé pour les outils *Gmail*. Aussi, l'Android est libre et gratuit et a été ainsi rapidement adopté par des fabricants.

La société Android a été rachetée en 2007 par *Google*. Mais aujourd'hui, l'Android 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'Android concurrence l'opérateur système d'Apple qu'il tend à dépasser en nombre d'utilisateurs. Android évolue pour mieux gérer l'hétérogénéité des appareils qu'il utilise.

II.3.2.2 Présentation de MIT APP INVENTOR :

APP INVENTOR est un logiciel en ligne développé par *Google Labs*. Abandonnée par *Google*, cette application a été reprise par le MIT (***M**assachusets **I**nstitute of **T**echnology*) et permet de développer des applications Android, basées sur l'assemblage de blocs (langage Scratch).

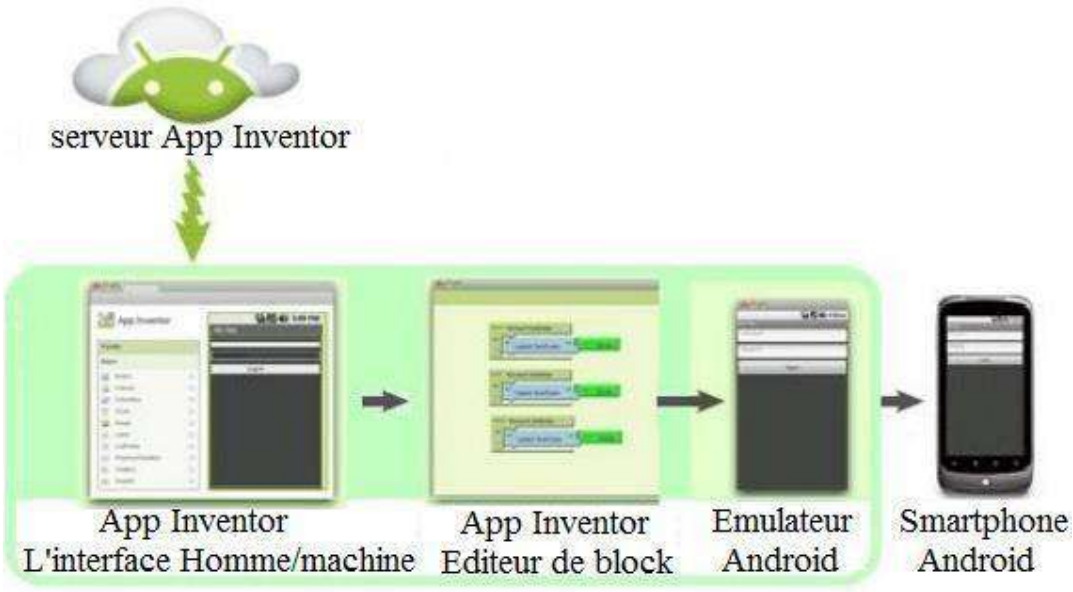
La programmation se réalise en ligne, à l'aide de son navigateur préféré, et sous l'environnement logiciel de son choix (Mac, Linux ou Windows). Seules contraintes : avoir un compte *Gmail* pour pouvoir y accéder, et un accès à internet évidemment. Les informations sont stockées sur des serveurs distants. Le site est en anglais, nous pouvons basculer sur une traduction française mais.



Figure II26 : Logo d'APP INVENTOR.

II.3.2.3 La conception d'APP INVENTOR

L'application est composée de trois fenêtres, une pour créer l'interface graphique, une autre pour assembler les blocs de programmation et l'émulateur :



FigureII.27: Création d'une application Android par APP INVENTOR.

- **L'interface graphique** :

  Dans cette partie on dispose de divers blocks graphiques comme des boutons, des champs de texte, des images, des sliders, mais aussi des layouts qui servent à délimiter les zones de notre application, des senseurs pour exploiter les capteurs du téléphone, et quelques autres fonctionnalités utiles de multimédia, de GPS et de communication Bluetooth.

- **Blocs de programmation :**

  Dans la partie blocs nous retrouvons toutes les fonctions mathématique, logique, ou évènementiel, ce rapportant aux objets utilisés dans la partie graphique, ce qui nous permet de tirer profit d'une simplicité de programmation orienté objet sous la forme de blocs près programmé, qu'ils ne restent plus qu'à organiser en fonction du programme souhaité.

- **L'émulateur :**

  Dans cette partie nous pouvons tester l'application. L'émulateur permet de remplacer un terminal réel pour vérifier le bon fonctionnement du programme.

Conclusion :

Dans ce chapitre, nous avons commencé par la partie matériels qui a englobé les différents composants qu'on a utilisé dans notre projet à savoir Arduino Uno et son principe de fonctionnement, servomoteur avec son principe et ses caractéristiques, le fameux module Bluetooth hc 06 sa définition et fonctionnement. Finalement le capteur optique à fourche TCRT5000 sa description et son principe de fonctionnement sans avoir oublié l'utilisation de l'application Android qui est explicitée et démontrée sous le logiciel de Google Labs qui fait partie de la société **MIT** (***M**assachusets **I**nstitute of **T**echnology*).

Chapitre III

Etude technique
Et pratique du projet

III.1 Introduction

Dans ce chapitre, nous allons développer les étapes nécessaires pour aboutir à notre objectif et faciliter la réalisation de notre projet ainsi que les Outils de programmation (Proteus ISIS, Arduino).

III.2 Les étapes de la réalisation

La réalisation de notre dispositif a connu trois étapes:

Première étape : C'est l'étape la plus difficile car elle consiste en la recherche des informations

bibliographiques et techniques à utiliser pour aboutir à l'objectif.

Deuxième étape : Etablir le circuit électrique dans l'environnement ISIS puis programmer l'Arduino Uno par son IDE arduino (logiciel Arduino).

Troisième étape : réalisation du montage complet sur une plaque d'essai puis sur maquette.

III.3 Etude technique

III.3.1 Principe et mode de fonctionnement

Le principe de fonctionnement de notre projet est la commande de vitesse d'un moteur à cournat continu DC, à l'aide d'un capteur optique IR et carte arduino uno via application android MIT App Inventor.

III.3.2 Outils de programmations

L'Arduino uno peut être programmé par divers langages et pour chaque langage il ya plusieurs compilateurs, nous avons choisi le langage Arduino avec son compilateur integé.

- **Logiciel IDE Arduino**

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.

- **Logiciel ISIS (Proteus)**

Isis est un éditeur de schéma électrique qui intègre un simulateur analogique, logique ou mixte. Toutes les opérations se passent dans cet environnement, aussi bien la configuration des différentes sources et le tracés des courbes.

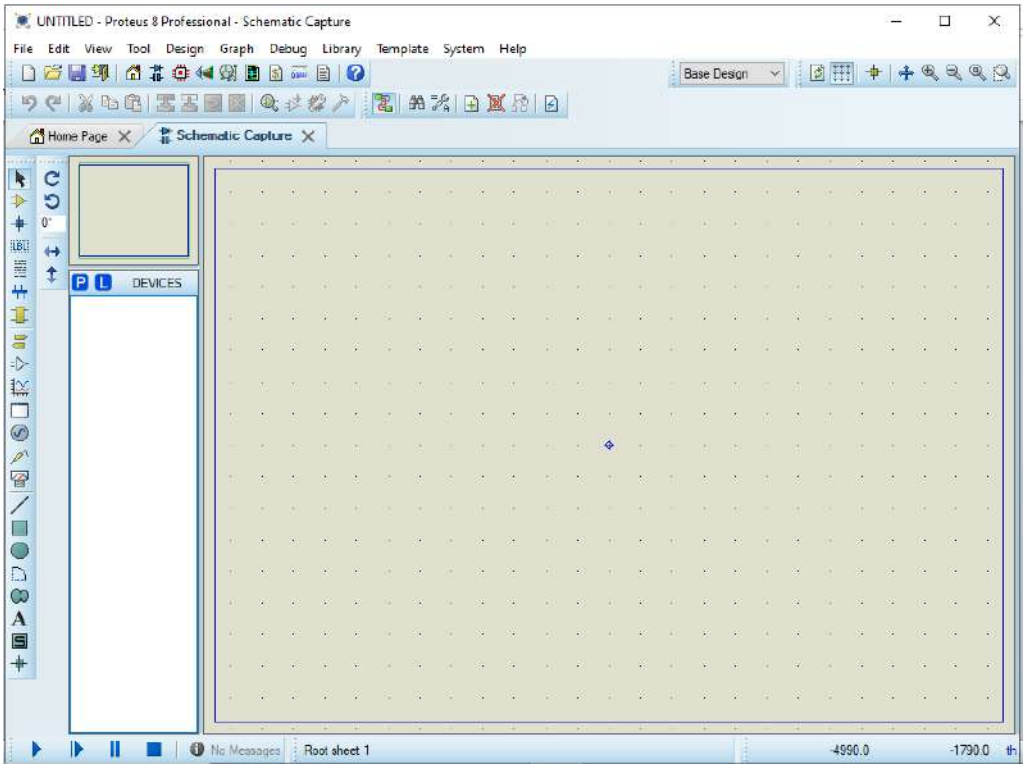
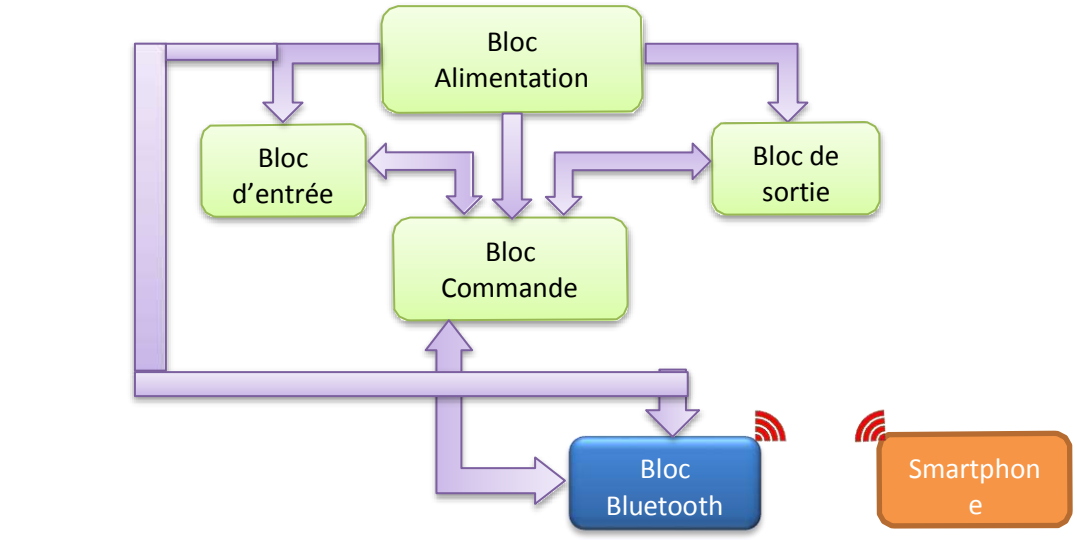


Figure IV.1 : ISIS (PORTEUS)

III.3.3 Schéma synoptique (Blocs) du montage



5v

Connexion sans fil

Fig Ⅲ-1 : Schéma synoptique

- **Bloc d'alimentation :**

L'alimentation labo nous délivre une tension de 5v, cette tension va assurer l'alimentation du reste des blocs en énergie électrique.

- **Bloc de commande :**

Le bloc microcontrloeur Arduino est responsable de tous traitement nécéssaires, comme par exemple l'acquisition de données produits par le capteur IR, ces données vont subir des calculs et paramétrages pour se finalisent par un résultat donnant la vitesse du moteur.

- **Bloc d'entrée :**

Le rôle fondamental de ce bloc est l'acquisition de données produites par le capteur IR, ces données vont subir des calculs et paramétrages pour se finalisent par un résultat donnant la vitesse du moteur.

- **Bloc de sortie :**

Dans ce bloc on trouve le circuit de puissance qui est responsable au pilotage du moteur en mode variation de vitesse et mode commande de sens de rotation du moteur.

- **Bloc Bluetooth :**

Comme son l'indique, ce bloc contient un module Bluetooth qui sert à connecter notre dispositif à l'application Androïde en vue de bien gérer les deux barrières.

III.3.4 Schéma électronique du montage

Avant de passer à la réalisation pratique, nous avons utilisé le logiciel PROTEUS, c'est un logiciel électronique perfectionné conçu par **Labcenter Electroniques** qui permet de dessiner des schémas électroniques, de les simuler et de réaliser le circuit imprimé correspondant. Le logiciel « PROTEUS » est disponible et téléchargeable gratuitement, il se compose de nombreux outils regroupés en modules au sein d'une interface unique (éditeur **ISIS** et éditeur **ARES**).

Ce dernier nous permet de schématiser notre carte électrique et faire la simulation.

La figure suivante montre notre schema établi sur l'interface d'éditeur ISIS.

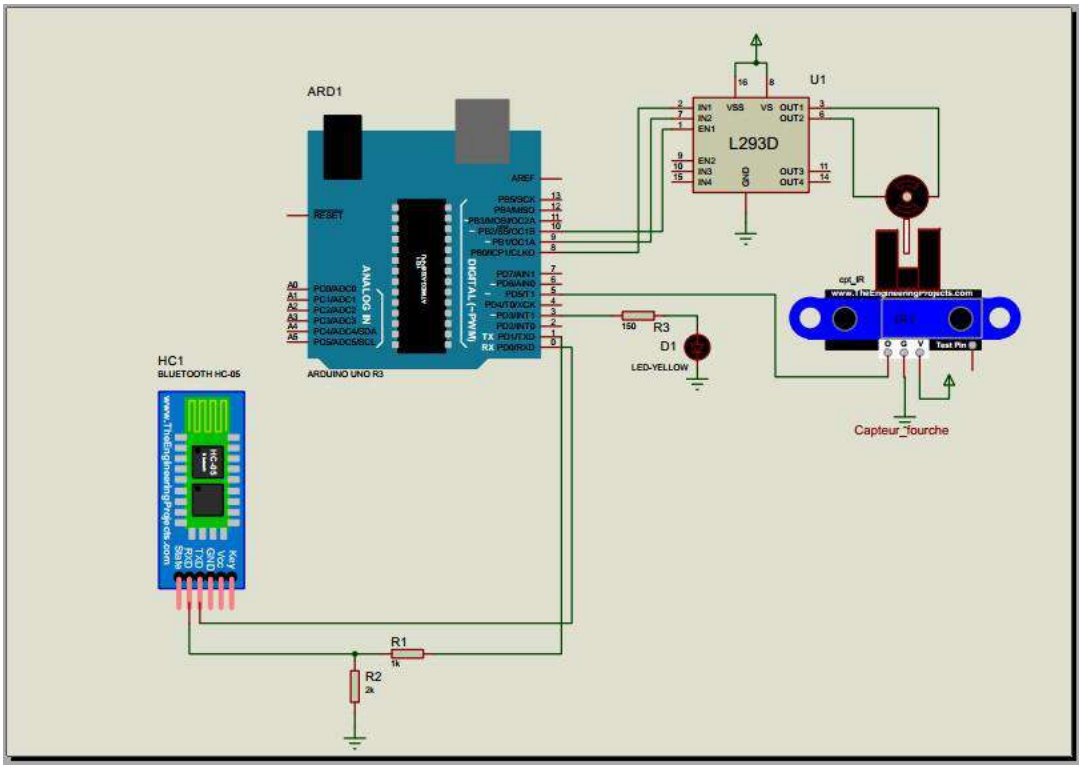


Figure IV.7 : Schéma électronique de notre projet

Le principe de ce montage est le suivant :

- ✓ Un Module Bluetooth de reference HC06 qui a pour role le controle à distance de la vitesse du moteur ainsi que la reception de données depuis l'application MIT App Inventor. Tel que la communication entre le module HC06 et Arduino se fait par un protocol nomé UART, l'échange de données est établi sur deux lignes appellées RX et TX.
  Les deux lignes RX et TX de HC06 sont connectées aux deux lignes TX et RX de la carte Arduino réspéctivement.

- ✓ Le capteur optique, qui a comme role le comptage du nombre de rotation du moteur, est connecté à la broche N° 5 de l'Arduino, cette broche est configure en entrée c'est à dire elle va recevoir des impulsions du capteur IR.
  Dans la mesure ou la vitesse est exprimée en nombre de rotation par minute on doit établir notre calcule d'impulsions en un interval de temps d' une minute.

- ✓ Comme est illustré sur le schema le moteur est commandé par le driver L293 à travers les broches 1,2 et 7. Le tableau suivant illustre la manière avec laquelle est commandé notre moteur,
  - Broche 1 est l'entrée de validation **EN1**
  - Broche 2 est l'entrée **IN1** responsible du sens de rotation
  - Broche 7 est l'entrée **IN2** responsible du sens de rotation

Enable 1	Input 1	Input 2	Fonction
High	Low	High	Tourne dans le sens horlogique
High	High	Low	tourne dans le sens anti-horlogique
High	Low	Low	Stop
High	High	High	Stop
Low	Non applicable	Non applicable	Stop

Ⅲ-4 Etude software :

III-4-1 Programmation de l'Application Android :

Etapes de développent de l'application :

Comme nous l'avons déjà souligné, notre dispositif sera équipé d'un module Bluetooth pour permettre un pilotage à distance. Pour cela nous avons utilisé le MIT App Inventor (en ligne) pour créer une application Android facilitant cette commande à distance.

Avant l'utilisation de cette application un compte Google Play personnel doit être crée. Les étapes du développement de l'application sont les suivantes :

- Commencer un nouveau projet.

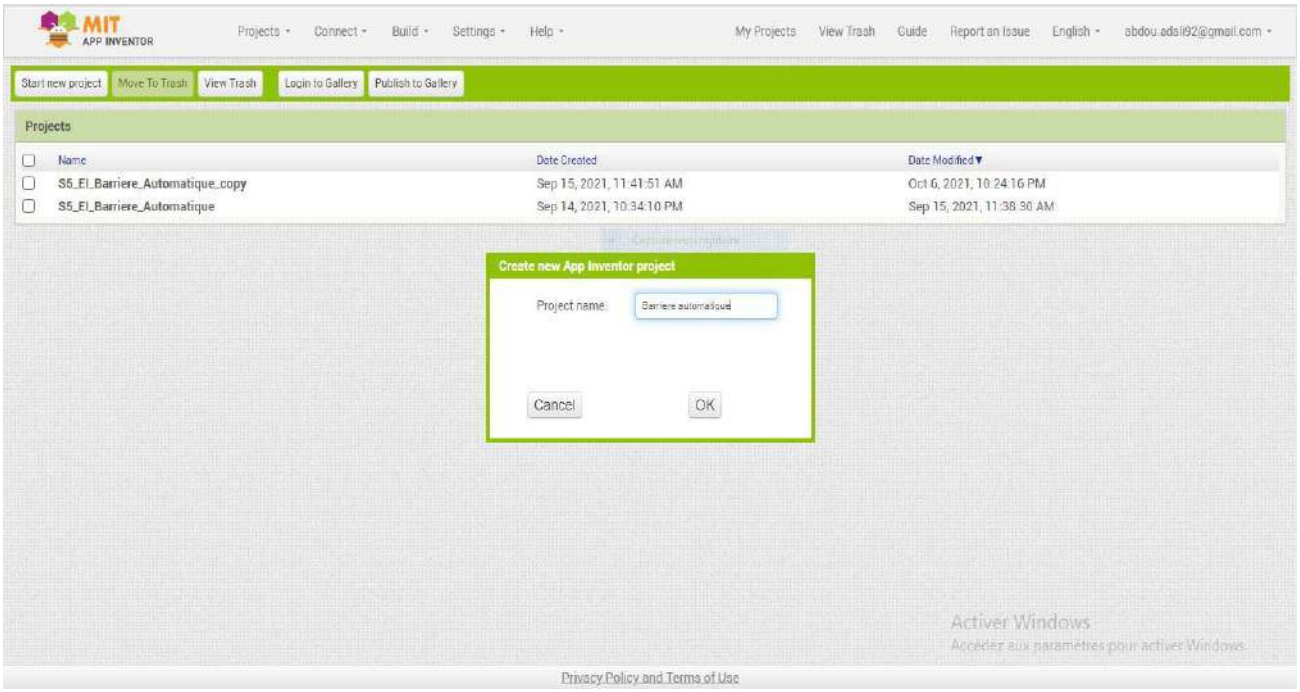


Fig Ⅲ-8 : L'interface d'un nouveau projet.

- Au début du projet et à partir de la section « Designer » du logiciel App Inventor on commence par la conception graphique désiré pour notre application. Les objets qu'on souhaite utiliser (les boutons, les tableaux, les ascenseurs, les zones de textes…etc.) doivent être glissés dans la« Screen1 ».

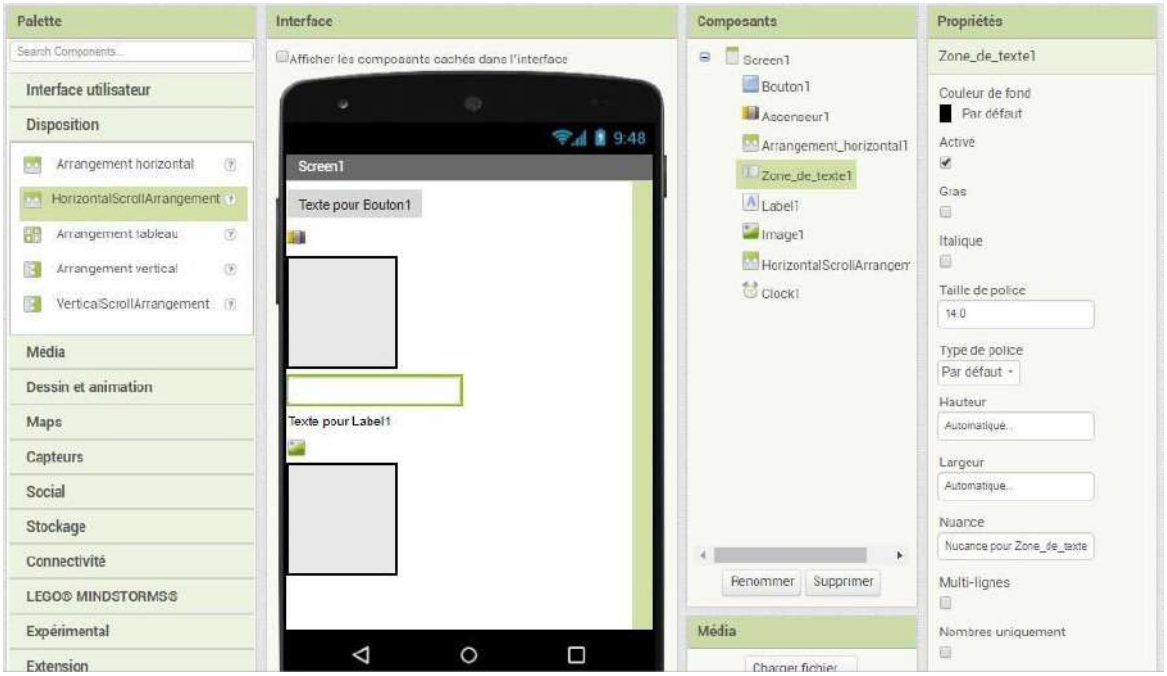


Fig Ⅲ-9 : Remplissage de « **Screen 1** ».

- L'étape suivante consiste à modifier les propriétés des différents objets utilisés (boutons, Label, Arrangement Horizontal, Arrangement Vertical, etc…) pour obtenir l'interface voulue.

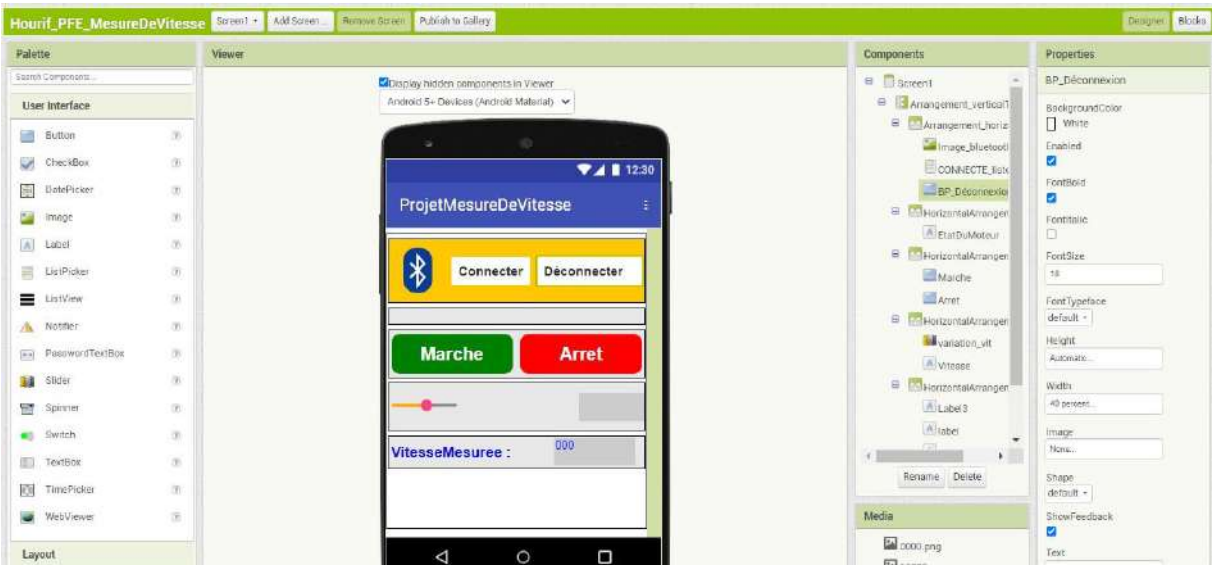


Fig Ⅲ-10 : L'interface de l'application sous App Inventor.

- Une fois qu'on à terminer l'interface graphique de l'application. On passe à la section « **Blocs** » où on programme tous les objets qui se trouvent dans le « **Screen 1** », chacun selon les tâches à exécuter.

- Le Block permettant d'afficher la liste des périphériques Bluetooth disponibles au Smartphone et d'établir la connexion :



Fig Ⅲ-12 : Block de la liste des périphériques disponibles et connexion

- Le Block qui montre l'état du Smartphone avec le module Bluetooth (connecté ou déconnecté) affiché avec une couleur différente :

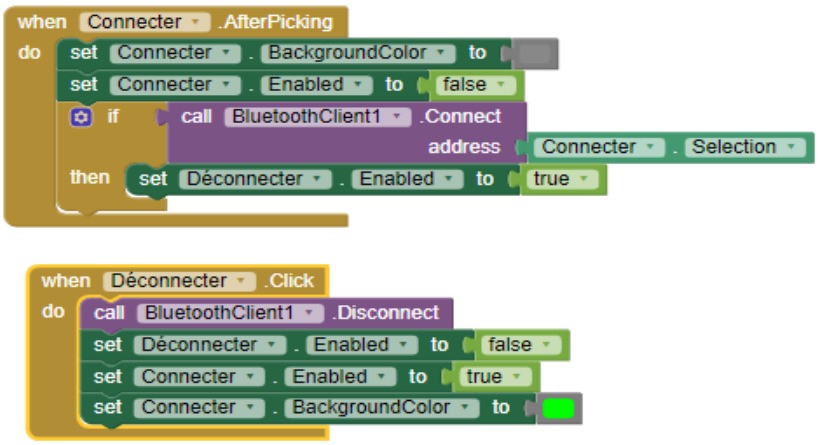


Fig Ⅲ-13 : Block de l'état du Bluetooth.

- Les Blocks de contrôle des modes de fonctionnement du systeme. Mode marche et mode Arret, chaque mode a son propre bloc de contrôle comme est montré sur la figure suivante :

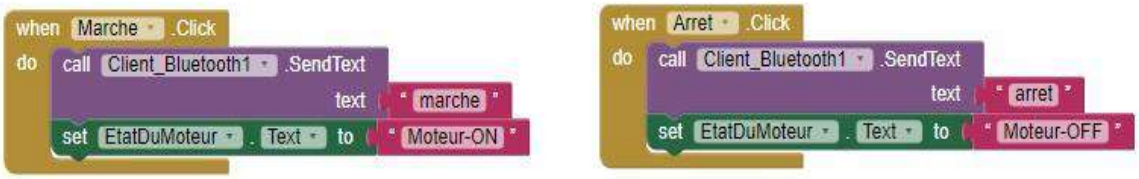


Fig Ⅲ-14 : Block de contrôle des modes

- Les Blocks de contrôle de vitesse du moteur est donné par la figure ci_dessous.

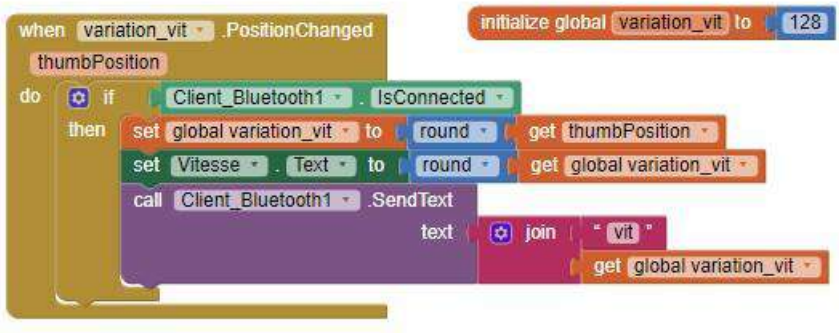


Fig Ⅲ-15 : Block de contrôle des barrières en mode manuel

- L'étape suivante consiste à terminer la création des blocs de l'application en cliquant sur "**construire**" pour créer un fichier avec une extension Android « .APK » pour que l'application soit compatible avec tous les téléphones Android.

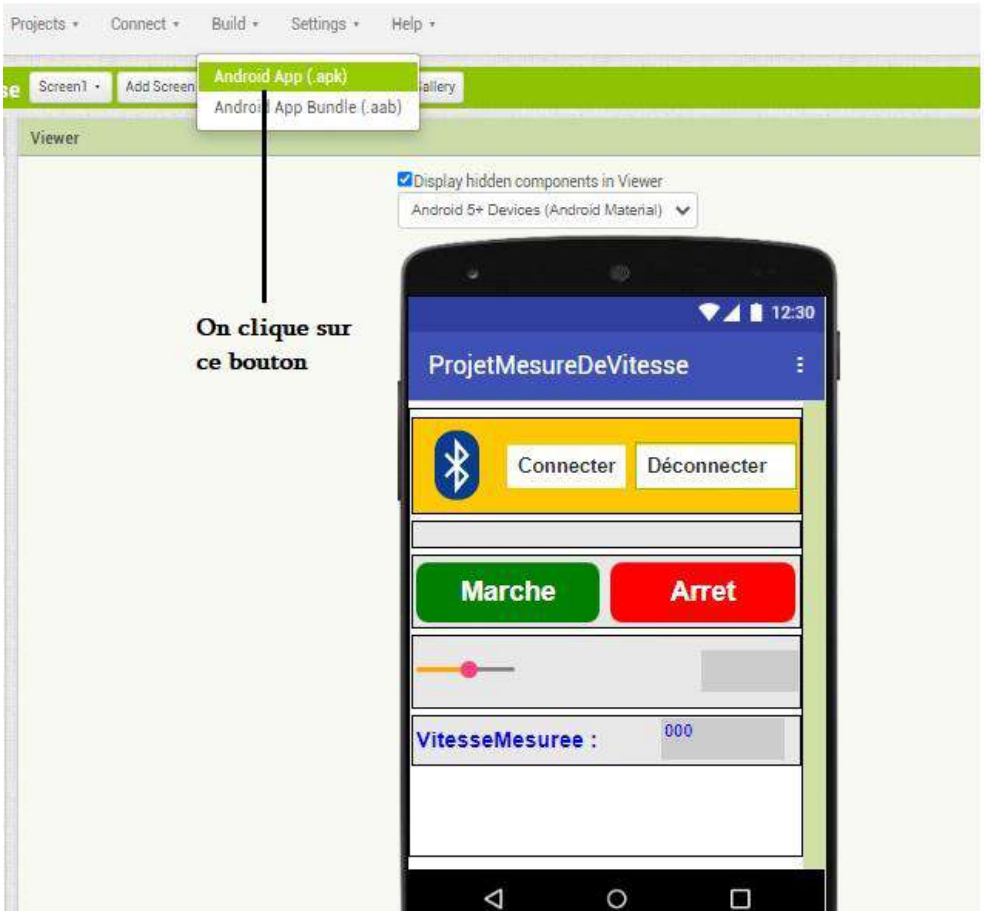


Fig Ⅲ-16 : Création du fichier avec l'extension Android.

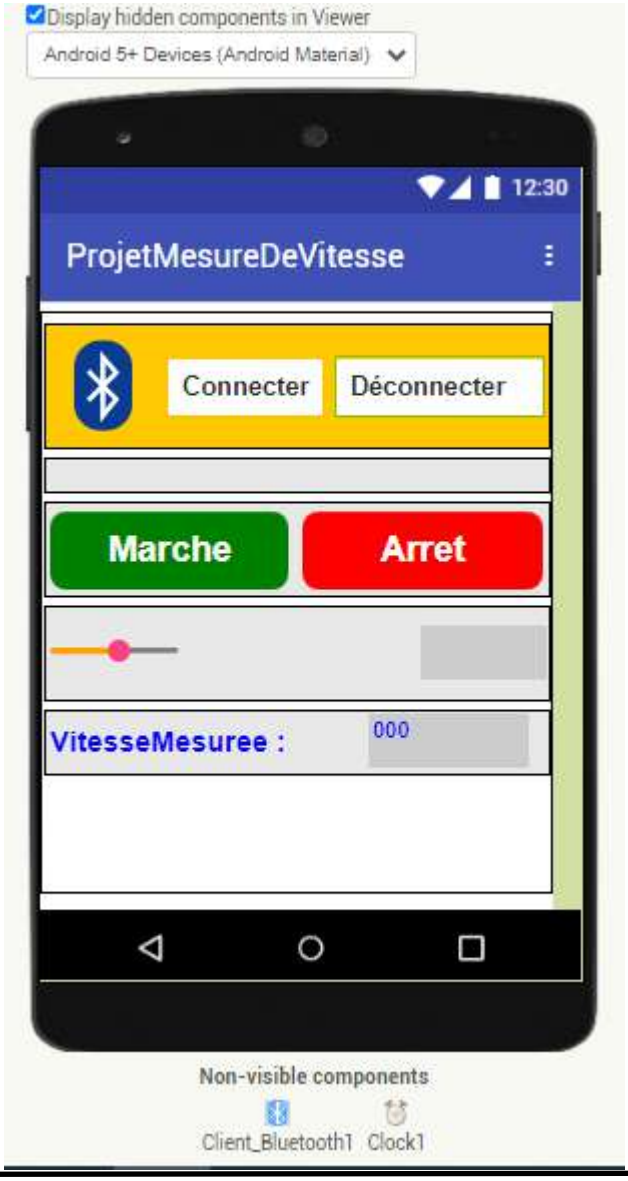


Fig Ⅲ-17 : L'interface de l'application sous Smartphone.

III.4.2 Code source du projet :

Partie déclaration de la librairie nécessaire dont on a besoin lors de la programmation au module Bluetooth c'est la liaison série pour connecter le module HC06.

```
/* Bluetooth_led est un programme qui pilote une led depuis une application */
#include <SoftwareSerial.h>
```

La deuxième partie est la création des instances (objets utilisés) et déclaration des variables et constantes.

```
int in1 = 8,in2=9,EN1=10,led=3,cpt_IR=5,RX=7,TX=6;
SoftwareSerial bluetooth(RX,TX);
String chaine_recue="";          // variable pour la reception
String etat_moteur="arret";
int i=0;
int valvitesse=0;
int lumiere;  // affiche si le faisceau lumineux est coupé ou non
int tour=0;   //variable qui compte le nombre de fois où la partie
float trmin;  // variable qui affichera la vitesse du disque en tou
unsigned long temps;  //sert à enregistrer le temps écoulé depuis
int lumiere_precedent=0;
int lumiere_actuelle=0;
```

La troisième partie du programme consiste en l'initialisation des variables et constantes ainsi que la configuration des différents paramètres et broches d'Arduino utilisés.

```
void setup(){
  Serial.begin(9600);             // initialisation liaison série à 9600 bauds
  bluetooth.begin(9600);          // initialiser la connexion  Bluetooth à 9600 bauds
  pinMode(led,OUTPUT) ;           // la broche led est en sortie
  pinMode(cpt_IR,INPUT) ;
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(EN1, OUTPUT);
  analogWrite(EN1,0);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
}
```

La quatrième partie consiste en la creation d'une function qui va permettre de faire le calculi de la vitesse du moteur en se basant sur les valeurs mesurées lors de l'acquisition du nombre de rotation.

```
void boucleDeCalcule_vitesse(){
  lumiere_precedent=lumiere_actuelle;
  lumiere_actuelle = digitalRead(cpt_IR);
  if (lumiere_precedent == LOW && lumiere_actuelle == HIGH) {
        // on ajoute 1 à la variable qui compte le nombre de tours
        tour=tour+1;
        Serial.print("tour: ");
        Serial.println(tour);
        // si il s'est passé plus d'une seconde depuis le début du calcul du nombre de tours

        if ((millis() - temps) >= 10000){
              // trmin= (nb de tours depuis le dernier calcul)/ (temps écoulé entre les deux
              trmin=tour*60000.0/(millis() - temps);
              Serial.print("vitesse en tour/min: ");
              Serial.println(trmin);
              //Réinitialisation du temps et du nombre de tours
              temps=millis();
              tour=0;
          }
  }
}
```

La cinquième partie est consacrée au programme principal de notre dispositif, à savoir qu'elle est débutée par l'attente d'éventuelle réception Bluetooth, si c'est le cas on aura alors les différents modes de.

```
void loop() {
    while (bluetooth.available()){
        chaine_recue=bluetooth.readString();
        if(chaine_recue=="marche" || chaine_recue=="arret"){
            Serial.print("chaine_recue: ");Serial.print(chaine_recue);
        }
        else if(chaine_recue.startsWith("v")){
          if(etat_moteur=="arret"){
            Serial.println("----Mettre d'abord le moteur en etat de marche----");
          }else if(etat_moteur=="marche"){

           }
        }
     }
    //delay(2000);
```

```
if (chaine_recue.startsWith("m")){
    etat_moteur="marche";
    Serial.println("  ------------->  VitesseMoteur MAX ");
    digitalWrite(led,HIGH);
    analogWrite(EN1,255);
    temps=millis();
    while(etat_moteur=="marche"){
       boucleDeCalcule_vitesse();
       while (bluetooth.available()){
          if(bluetooth.readString()=="arret"){
              etat_moteur="arret";
           }
       }
    }
    chaine_recue=" ";
    }

else if (chaine_recue.startsWith("a")){
     etat_moteur="arret";
     Serial.println("  ------------>  MoteurEnArret");
     bluetooth.print("   ");
     digitalWrite(led,LOW);
     analogWrite(EN1,0);
     chaine_recue=" ";
    }
 else if (chaine_recue.startsWith("v") && etat_moteur=="marche"){
     valvitesse = chaine_recue.substring(3,chaine_recue.length()).toInt();
     String valstr_vitesse=String(valvitesse);
     //Serial.print("valvitesse: ");
     Serial.print("  ------------->  VitesseDeRotation: ");
     Serial.println(valvitesse);
     bluetooth.print(valstr_vitesse);
     delay(300);
     analogWrite(EN1,valvitesse);
     chaine_recue=" ";
 }
```

Ⅲ-5 Quelques photos de la réalisation : Ⅲ-5-

1 Câblage et montage :

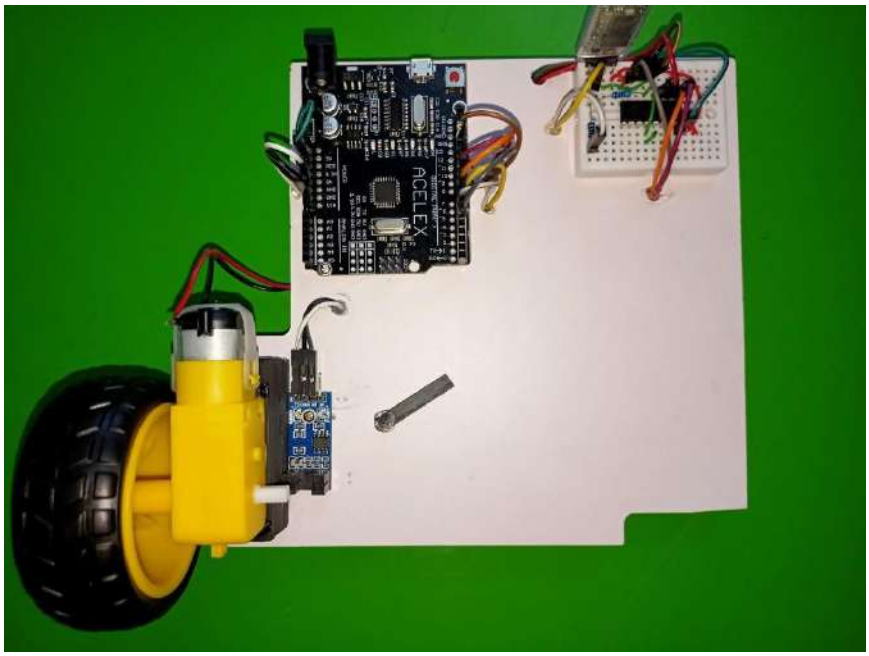


Fig Ⅲ-25 : Photo de la réalisation

III-5-2 Bloc Bluetooth et Driver du moteur:

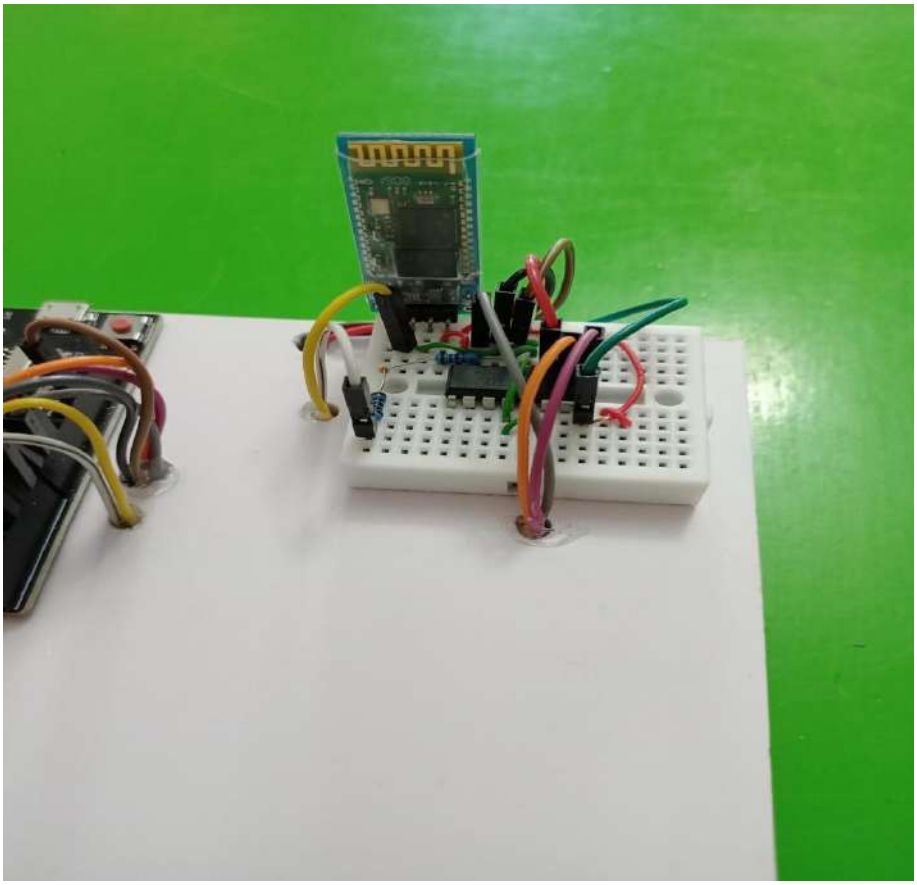


Fig Ⅲ-24 : Photo du bloc Bluetooth et driver L293

III-5-3 Bloc de commande et de traitement :

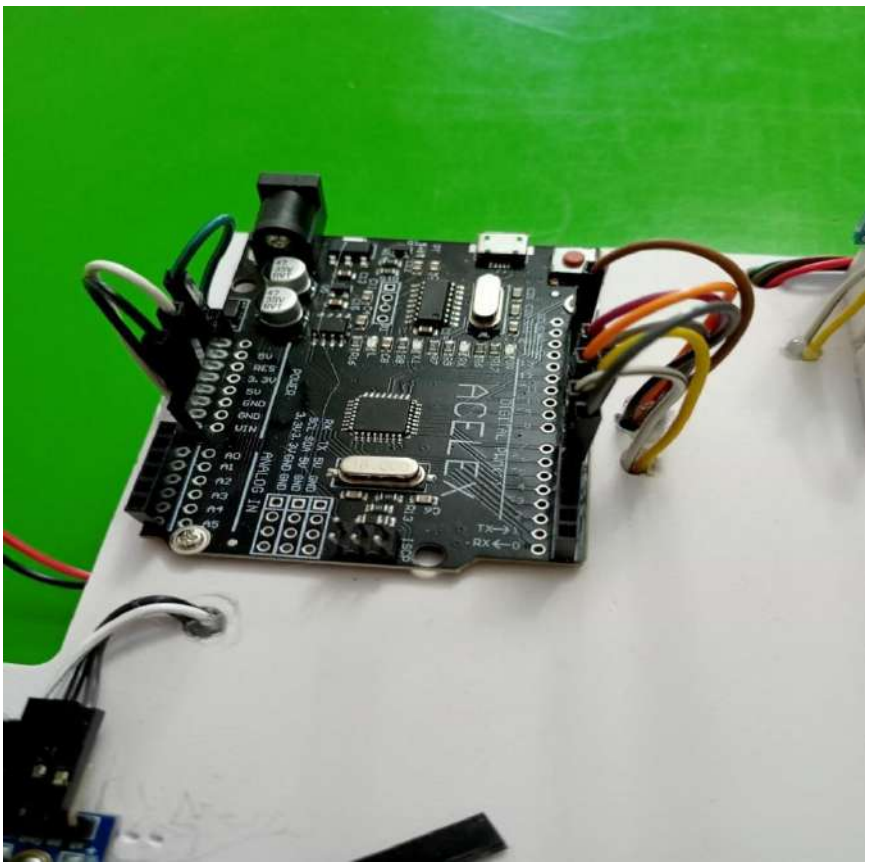


Fig Ⅲ-22 : Photo du bloc Arduino

III-5-4 Bloc du capteur optique IR et Moteur DC :

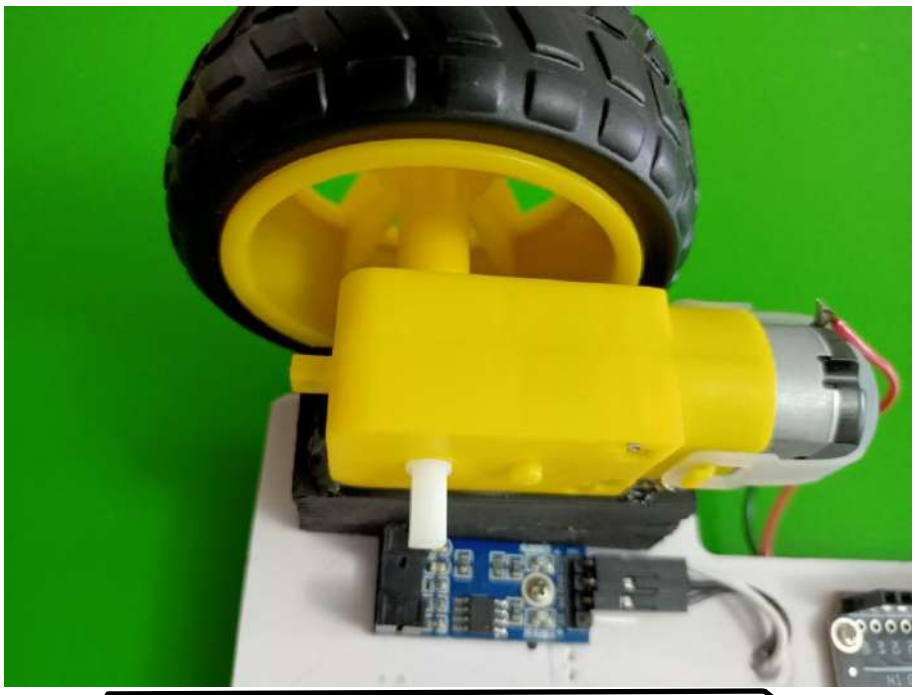


Fig Ⅲ-23 : Photo du bloc de capteur et moteur

III.6 Conclusion

Ce dernier chapitre a été cloture par une réalisation pratique en exploitant une méthodologie d'analyse hiérarchique.

Nous avons réalisé un système de commande de moteur DC et aussi un système qui détecte le nombre de rotation et qui sera convertie en vitesse puis envoyée à l'interface Android sur notre Smartphone, la valeur de celle est affichée et pourra etre modifiée en agissant sur le slider de l'application.

Enfin, le tout est finalisé par la réalisation pratique d'une maquette.

Conclusion Générale

Conclusion Générale :

Le travail présenté dans notre mémoire de fin d'étude (conception et réalisation d'une carte de contrôle de vitesse d'un Moteur DC via App Android) est consacré à la supervision et contrôle du moteur DC en paramètre de rotation

Le but de ce travail réside dans le développement d'une nouvelle idée, autrement dit un nouveau système de commande de moteur DC gérées par Androïde dont l'application est développée par le logiciel gratuit **MIT App Inventor**. A cet égard, nous avons opté pour deux modes de fonctionnement à savoir le mode lecture de vitesse , le mode variation de vitesse, tel que chacun de ces deux modes a été développé et détaillé de manière à ce que l'utilisateur ne sera confronté par aucune difficulté lors de son accès Smartphone.

La technique émetteur/récepteur infrarouge est choisie dans notre projet en vue de détecter le nombre de rotation du moteur et ce pour faciliter la mesure de vitesse.

Un module Bluetooth a été utilisé dans notre projet pour pouvoir aussurer la communication de la carte de commande et le Smartphone.

Le cœur du projet est articulé sur la carte de developpement qui tres repandue dans le monde d'éléctronique, c'est la carte Arduino-Uno. Ce dernier a été profondément détaillé dans ce mémoire.

En effet, plusieurs perspectives peuvent être exposées dans l'objectif d'améliorer notre projet à savoir :

- de mettre en place un système d'affichage des différents paramètres du projet tout en utilisant, à titre d'exemple, un afficheur LCD ou encore un afficheur Oled en vue de visionner le bon déroulement du processus.

- Envoyer et recevoir les données par WIFI , et ce pourra être géré par un microcontrôleur de la firme espressif (par exemple ESP8266, ou ESP32) .

- Il est également possible d'envisager de créer une page Web qui aura comme rôle transfert et réception des données.

- Mettre en perspective l'ajout d'un système de ventilation pour le refroidissement du moteur.

En conclusion, nous espérons que ce modeste travaille fait partie de tous ce que nous avons appris comme théories au sein de l'université Yahia Farès de Médèà, nous espérons également, que ce projet rend service et sert comme référence riche en information aux prochains étudiants.