وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA

جامعة باجي مختار- عنابة

Année : 2018

Faculté: Sciences de l'Ingéniorat
Département: Electronique

MEMOIRE
Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé :
Réalisation d'une Machine CNC à base d'Arduino contrôlé par un système Android

Domaine : Sciences et Technologie
Filière : AUTOMATIQUE
Spécialité : Automatique et Informatique Industrielle

Par :
Drici Radia

DEVANT Le JURY

Président : Abd-Elghani.REDJATI	**Grade**	**UBM Annaba**
Directeur de mémoire : Mohamed.N.SAADI	**Grade**	**UBM Annaba**
Examinateurs : Mohamed.FEZARI	**Grade**	**UBM Annaba**

Dédicaces

A la pensée de mon père, à ma chère maman pour tous leurs
Sacrifices, leur amour, leur tendresse, leur soutien et leurs
Prières tout au long de nos vies,

A mes chers frères, Hamza, Abd el-hamid, Mohamed, Hossain et Nouh pour leur appui et leur encouragement, et leur soutien moral ;
Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fuit de votre soutien infaillible,
Merci d'être toujours là pour moi.

A tous le nombre de la section de l'automatique 2018, et à mes amis.

À tous ceux qui me sont chers et proches,
À tous ceux qui ont semé en moi à tout point de vue,
À tous(te)les proches et les ami(e)s que le destin a arrachés à la vie.

Remerciements

Je tiens tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce Modeste travail.
En second lieu, je tiens à remercier mon encadreur Mr Saadi et son précieux conseil et son aide durant toute la période du travail.

Mes remerciements s'étendent également à Mr : Ben ouareth « Encadreur du Pratique » pour ses bonnes explications qui m'a éclairé le chemin de la recherche et sa collaboration avec moi dans l'accomplissement de ce modeste travail.

mon vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions. Nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail spécialement : Imed et mes collègues de la section d'automatique

Je tiens à exprimer mes sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

SOMMAIRE

SOMMAIRE

SOMMAIRE

LISTES DE FIGURES :

INTRODUCTION GENERALE :

Aujourd'hui, les machines commande numérique par calculateur (CNC) permettent une production économique et rentable. Le contrôle des coûts reste une préoccupation importante. La diminution des quantités des séries et le raccourcissement de la longévité des produits finaux demandent des déroulements de production de plus en plus flexibles.

Durant ces dernières années, la commande des machines électriques a subi des progrès significatifs. Ces progrès sont essentiellement dus à la révolution technologique en informatique industriel, ce qui a permis le développement de solutions numériques efficaces avec une possibilité d'implanter des algorithmes plus complexes. Ces commandes sont en majorité basées sur les microprocesseurs, les DSP (Digital Signal Processor) et les microcontrôleurs PIC. Ces processeurs sont équipés d'unité arithmétique et logique (UAL) dédiée à la réalisation des calculs arithmétiques. Ils intègrent également des périphériques tel que les convertisseurs analogiques/numériques et les « timers » bien adaptés aux besoins de commande de machines électriques.

Le moteur pas à pas est l'interface idéale entre l'électronique numérique et la mécanique, il permet de convertir directement un signal électrique en un positionnement angulaire à caractère incrémental. Pour cela on utilise le port USB d'un ordinateur de type PC qu'on commande avec une carte Arduino Méga et des circuits intégrés. Tous ca résument l'idée des systèmes embarqués, qui mobilisent de nombreuses compétences, l'architecture électronique pour le choix des composants et du micro-processeur, en tenant compte de contraintes comme la sensibilité électromagnétique. l'informatique pour décider de l'action en fonction des signaux des capteurs , et les réseaux, pour la communication de systèmes embarqués, dont la communication se fait à l'aide du technologie des smartphone qui facilite la connexion des objets en utilisant des applications dédier a cette communication « On qualifie de « système embarqué » un système électronique et informatique autonome dédié à une tâche précise, souvent en temps réel, possédant une taille limitée et ayant une consommation énergétique restreinte ». Des milliards d'objets et services dans le monde sont aujourd'hui « connectés ». Ils se développent sur divers marchés (industrie, santé, bien-être et domotique …), transforment et créent de nombreux métiers. Qu'ils soient utilisés à des fins professionnelles ou privées, ces objets et services sont « connectés » dans le sens où ils ont tous pour points communs :

- D'intégrer des programmes informatiques de recueil et de transmission des données fournies par les utilisateurs de ces objets, notamment les consommateurs.

- De pouvoir être reliés à des systèmes externes d'analyse et de conservation des informations résultant de leur utilisation.

- De pouvoir donner à des tiers prestataires de services, le droit de traiter et de stocker notamment dans le « nuage informatique » les données relatives aux utilisateurs de ces objets et les informations résultant de leur traitement.

Mots clés : CNC, ANDROID, ARDUINO

RESUME :

Aujourd'hui, les machines commande numérique par calculateur (CNC) permettent une production économique et rentable. Le contrôle des coûts reste une préoccupation importante. La diminution des quantités des séries et le raccourcissement de la longévité des produits finaux demandent des déroulements de production de plus en plus flexibles. Durant ces dernières années, la commande des machines électriques a subi des progrès significatifs. Ces progrès sont essentiellement dus à la révolution technologique en informatique industriel, ce qui a permis le développement de solutions numériques efficaces avec une possibilité d'implanter des algorithmes plus complexes. Ces commandes sont en majorité basées sur les microprocesseurs, les DSP (Digital Signal Processor) et les microcontrôleurs PIC. Ces processeurs sont équipés d'unité arithmétique et logique (UAL) dédiée à la réalisation des calculs arithmétiques. Ils intègrent également des périphériques tel que les convertisseurs analogiques/numériques et les « timers » bien adaptés aux besoins de commande de machines électriques. Le moteur pas à pas est l'interface idéale entre l'électronique numérique et la mécanique, il permet de convertir directement un signal électrique en un positionnement angulaire à caractère incrémental. Pour cela on utilise le port USB d'un ordinateur de type PC qu'on commande avec une carte Arduino Méga et des circuits intégrés de plus un Bluetooth et un téléphone ou bien un smartphone avec une application (App Inventor 2).

Dont dans notre projet on va commander cette machine CNC par une application Android en utilisant un téléphone et une connexion Bluetooth.

Et pour puisse compléter ce projet nous allons suivre la structure du travail :
Le premier chapitre, ferai l'objet de la présentation, la structure du carte Arduino Méga et son rôle important d'une autre manière : l'informatique embarque.

Le deuxième chapitre est spécifier à l'Android la partie d'application dans notre commande contient bien sûr la technologie, le principe de fonctionnement et les différents versions utilises dans ce domaine.

Le troisième chapitre est consacré à la théorie de la machine CNC et la description des composants électroniques en vue de la commande des moteurs pas à pas, basée sur de la carte Arduino Mega et les circuits intégrés des moteurs pas à pas.

Le quatrième chapitre est faite d'une façon qui permet d'avoir une idée sur la richesse et la diversité des solutions qui peuvent être ajoutées grâce aux différentes étapes suivre pour puisse finir notre travail de telle façon commander la CNC par une application Android.

CHAPITRE 1

ARDUINO

I. L'Arduino :

La carte Arduino est une carte électronique basée autour d'un microcontrôleur et de composants minimum pour réaliser des fonctions plus ou moins évoluées à bas coût. Elle possède une interface USB pour la programmer. C'est une plateforme open-source 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. Arduino est peut être utilisé pour développer des applications matérielles industrielles légères ou des objets interactifs, et il peut recevoir en entrées une très grande variété de capteurs.

Les projets Arduino peuvent être autonomes, ou communiquer avec des logiciels sur un ordinateur (Flash, Processing ou MaxMSP, Labview). Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré assemblées, le logiciel de développement open-source est téléchargé gratuitement.

La programmation de la carte Arduino présente les principales fonctionnalités de l'interface de l'application Arduino. L'application Arduino vous permet de créer et éditer un programme (appelé sketch) qui sera compilé puis télé versé sur la carte Arduino. Ainsi, lorsque vous apportez des changements sur le code, ces changements ne seront effectifs qu'une fois le programme télé versé sur la carte.

I. 1.L'Arduino Mega2560 :

'Arduino Mega2560 est une carte microcontrôleur basée sur l'ATmega2560 (data-sheet). Il a 54 entrée / sortie broches numériques (dont 14 peuvent être utilisées comme sorties PWM), 16 entrées analogiques, 4 UART (ports série matériels), un cristal 16 MHz oscillateur, une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation. Il contient tout ce qu'il faut pour soutenir Le microcontrôleur ;

Connectez-le simplement à un ordinateur avec un câble USB ou alimentez-le avec un adaptateur AC-DC ou une batterie pour Commencer. Le Mega est compatible avec la plupart des boucliers conçus pour l'Arduino Duemilanove ou Diecimila.

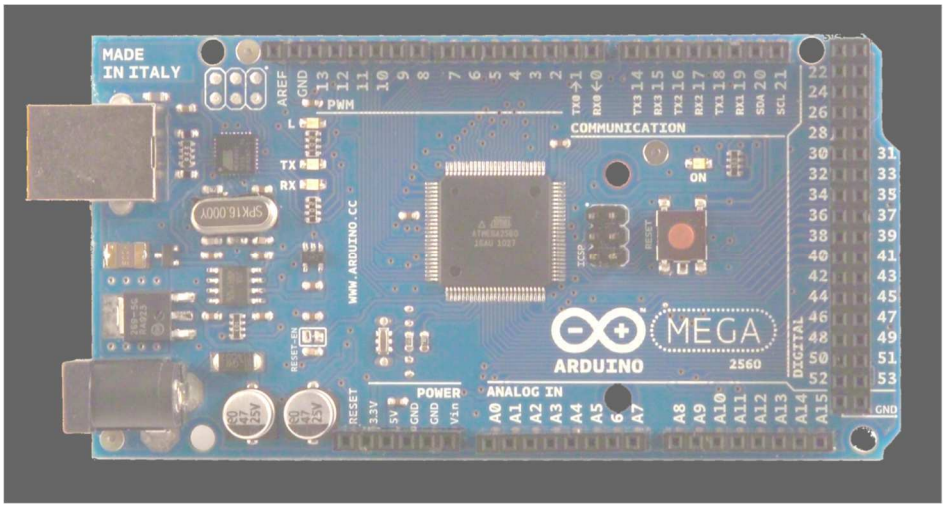


Figure1 : Arduino Mega2560

I.2 Les principaux points de fonctionnements (Résumé) :

Microcontrôleur ATmega2560

Tension de fonctionnement 5V

Tension d'entrée (recommandée) 7-12V

Tension d'entrée (limites) 6-20V

E / S numériques 54 (dont 14 fournissent une sortie PWM)

Pointes d'entrée analogiques 16

Courant CC par borne E / S 40 mA

Courant CC pour 3,3 V Pin 50 mA

Mémoire Flash 256 Ko dont 8 Ko utilisés par boot-loader

SRAM 8 Ko

EEPROM 4 Ko

Vitesse d'horloge 16 MHz

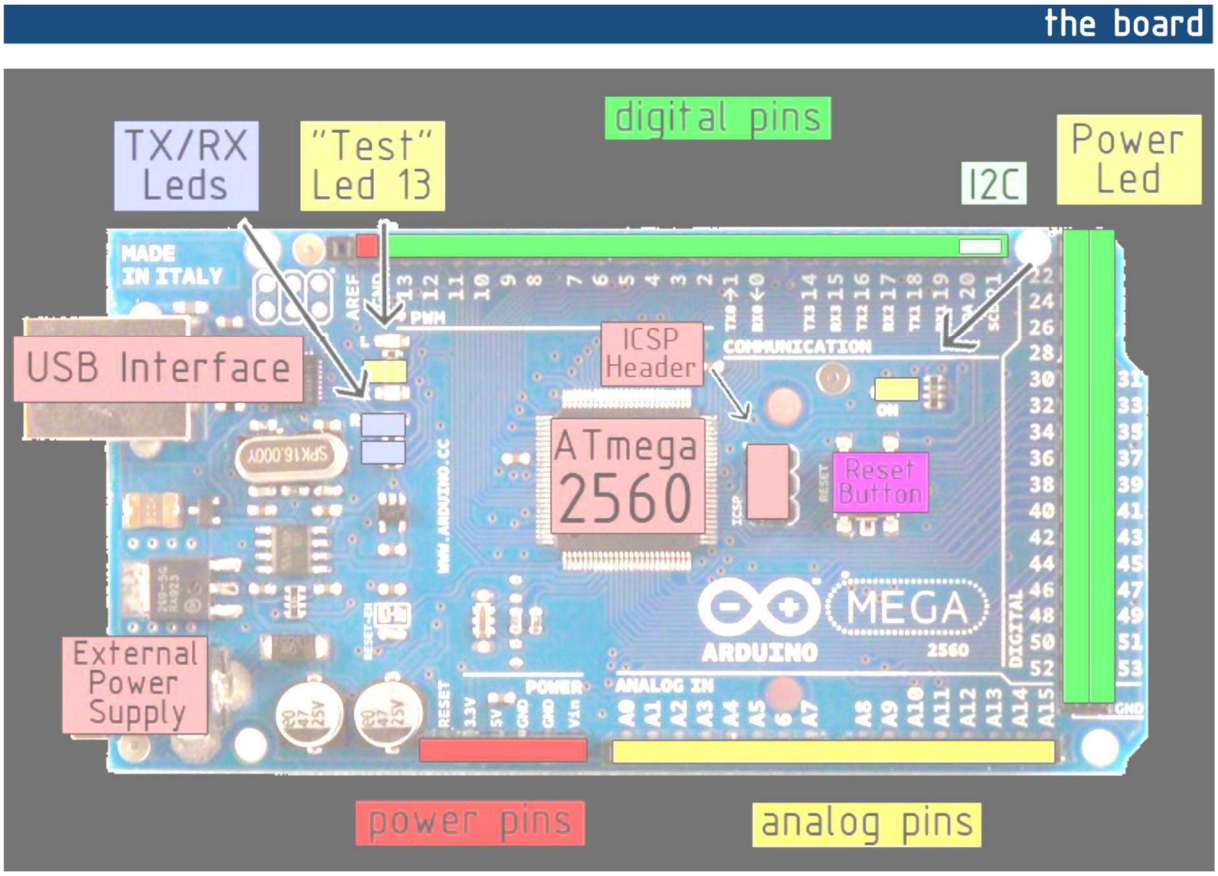


Figure 2 : conception schématique référence Arduino Mega2560

I.3 Puissance :

L'Arduino Mega peut être alimenté via la connexion USB ou avec une alimentation externe. La source d'alimentation est sélectionnée automatiquement.

L'alimentation externe (non USB) peut provenir d'un adaptateur AC-DC (wall-wart) ou d'une batterie. L'adaptateur peut être Connecté en branchant une fiche positive de centre de 2,1 mm dans la prise d'alimentation de la carte. Les fils d'une batterie peuvent être insérés dans les broches GND et Vin du connecteur POWER.

La carte peut fonctionner sur une alimentation externe de 6 à 20 volts. Si elle est fournie avec moins de 7V, cependant, la broche 5V peut fournir moins de cinq volts et le tableau peut être instable. Si vous utilisez plus de 12V, le régulateur de tension peut surchauffer et endommager Le tableau. La plage recommandée est de 7 à 12 volts.

Le Mega2560 diffère de toutes les cartes précédentes en ce qu'il n'utilise pas la puce de pilote FTDI USB vers série. Au lieu de cela, caractéristiques de l'Atmega8U2 programmé comme un convertisseur USB-série.

I.4 Les broches d'alimentation sont les suivantes :

- VIN. La tension d'entrée à la carte Arduino quand elle utilise une source d'alimentation externe (par opposition à 5 volts de la Connexion USB ou autre source d'alimentation régulée). Vous pouvez fournir une tension via cette broche ou, si vous fournissez une tension via la prise d'alimentation, accédez-y par cette broche.
- 5V. L'alimentation régulée utilisée pour alimenter le microcontrôleur et d'autres composants sur la carte. Cela peut venir soit à partir de VIN via un régulateur embarqué, soit être alimenté par USB ou par une autre alimentation 5V régulée.
- 3V3. Une alimentation de 3,3 volts générée par le régulateur embarqué. Le courant maximal consommé est de 50 mA.
- GND. Broches au sol.

I.5 Mémoire :

L'ATmega2560 dispose de 256 Ko de mémoire flash pour le stockage du code (dont 8 Ko pour le boot loader), 8 Ko de SRAM et 4 Ko d'EEPROM (qui peuvent être lu et écrit avec la librairie EEPROM).

I.6 Entrée et sortie :

Chacune des 54 broches numériques du Méga peut être utilisée comme entrée ou sortie, en utilisant pin Mode (), digital write (), et Fonctions digital Read (). Ils fonctionnent à 5 volts. Chaque broche peut fournir ou recevoir un maximum de 40 mA et dispose d'une résistance pull-up (déconnectée par défaut) de 20-50 kOhms. En outre, certaines broches ont des fonctions spécialisées :

- Série 0 : 0 (RX) et 1 (TX) ; Série 1: 19 (RX) et 18 (TX) ; Série 2: 17 (RX) et 16 (TX) ; Série 3: 15 (RX) et 14 (TX). Utilisé pour recevoir (RX) et transmettre (TX) des données sérient TTL. Les broches 0 et 1 sont également connectées à broches correspondantes de la puce série ATmega8U2 USB-to-TTL.

Interruptions externes : 2 (interruption 0), 3 (interruption 1), 18 (interruption 5), 19 (interruption 4), 20 (interruption3) et 21 (interruption 2). Ces broches peuvent être configurées pour

déclencher une interruption sur une valeur faible, un front montant ou descendant, ou un changement de valeur. Voir la fonction attach Interrupt () pour plus de détails.

- PWM : 0 à 13. Fournit une sortie PWM 8 bits avec la fonction analog Write ().
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI en utilisant la bibliothèque SPI.

Les broches SPI sont également réparties sur l'en-tête ICSP, physiquement compatible avec Uno, Duemilanove et Diecimila.

- LED : 13. Il y a une LED intégrée connectée à la broche 13 numérique. Lorsque la broche est à valeur HIGH, la LED est allumée, quand le La broche est BAS, c'est éteint.
- I2C : 20 (SDA) et 21 (SCL). Prise en charge de la communication I2C (TWI) à l'aide de la bibliothèque Wire (documentation sur le Site Web de câblage). Notez que ces broches ne sont pas au même endroit que les broches I2C sur le Duemilanove ou Diecimila.

Le Mega2560 possède 16 entrées analogiques, chacune fournissant 10 bits de résolution (c'est-à-dire 1024 valeurs différentes). Par défaut, il mesure de la terre à 5 volts, bien qu'il soit possible de changer l'extrémité supérieure de leur gamme en utilisant la broche AREF et Fonction analog Reference ().

Il y a quelques autres épingles sur le tableau :

1. AREF. Tension de référence pour les entrées analogiques. Utilisé avec analog Reference ().
2. Réinitialiser. Apportez cette ligne LOW pour réinitialiser le microcontrôleur. Généralement utilisé pour ajouter un bouton de réinitialisation aux boucliers qui bloquent celui au tableau.

I.7 La communication :

L'Arduino Mega2560 dispose d'un certain nombre d'équipements pour communiquer avec un ordinateur, un autre Arduino, ou autre microcontrôleurs. L'ATmega2560 fournit quatre UART matérielles pour la communication série TTL (5V). Un ATmega8U2 sur la carte canaux l'un de ces sur USB et fournit un port de com virtuel au logiciel sur l'ordinateur (Windows)

Les machines auront besoin d'un fichier .inf, mais les machines OSX et Linux reconnaîtront automatiquement la carte en tant que port COM. le logiciel Arduino comprend un moniteur série qui permet d'envoyer des données textuelles simples depuis et vers la carte.

Les voyants TX et RX sur la carte clignotent lorsque les données sont transmises via la puce ATmega8U2 et la connexion USB vers l'ordinateur (mais pas pour la communication série sur les broches 0 et 1).

Une bibliothèque Software Serial permet une communication série sur l'une des broches numériques du Mega2560.

L'ATmega2560 prend également en charge les communications I2C (TWI) et SPI. Le logiciel Arduino comprend une bibliothèque de fils à simplifier l'utilisation du bus I2C ; consultez la documentation sur le site Web de câblage pour plus de détails. Pour la communication SPI, utilisez le SPI bibliothèque.

I.8 La programmation :

L'Arduino Méga peut être programmé avec le logiciel Arduino (téléchargement). Pour plus de détails, voir la référence et les tutoriels.

L'ATmega2560 sur l'Arduino Méga est pré-brûlé avec un chargeur de démarrage qui permet de télécharger du nouveau code sans l'utilisation d'un programmeur matériel externe. Il communique en utilisant le protocole STK500 original (référence, C fichiers d'en-tête).

Vous pouvez également contourner le bootloader et programmer le microcontrôleur via l'ICSP (In-Circuit Serial Programming) entête ; Voir ces instructions pour plus de détails.

Le code source du micro-logiciel ATmega8U2 est disponible dans le référentiel Arduino. L'ATmega8U2 est chargé avec un DFU boot-loader, qui peut être activé en connectant le cavalier de soudure sur le dos de la carte (près de la carte de l'Italie) et puis réinitialiser le 8U2. Vous pouvez ensuite utiliser le logiciel FLIP d'Atmel (Windows) ou le programmeur DFU (Mac OS X et Linux) pour charger un nouveau firmware. Ou vous pouvez utiliser l'en-tête ISP avec un programmeur externe (en écrasant le DFU boot-loader). Consultez ce didacticiel écrit par l'utilisateur pour plus d'informations.

I.9 Réinitialisation automatique (logiciel) :

Plutôt que d'exiger une pression physique sur le bouton de réinitialisation avant un téléchargement, l'Arduino Mega2560 est conçu d'une manière cela lui permet d'être réinitialisé par un logiciel fonctionnant sur un ordinateur connecté. L'une des lignes de contrôle de flux matériel (DTR) de L'ATmega8U2 est connectée à la ligne de réinitialisation de l'ATmega2560 via un condensateur de 100 nanofarads. Quand cette ligne est affirmée (prise bas), la ligne de réinitialisation tombe suffisamment longtemps pour réinitialiser la puce. Le logiciel Arduino utilise cette capacité pour permettre de télécharger le code en appuyant simplement sur le bouton de téléchargement dans l'environnement Arduino. Cela signifie que le boot-loader peut avoir un délai d'attente plus court, car l'abaissement de DTR peut être bien coordonné avec le début du téléchargement.

Cette configuration a d'autres implications. Lorsque le Mega2560 est connecté à un ordinateur fonctionnant sous Mac OS X ou Linux, il réinitialise chaque fois qu'une connexion est établie à partir du logiciel (via USB). Pour la demi-seconde suivante, le boot-loader est fonctionnant sur le Mega2560. Alors qu'il est programmé pour ignorer les données malformées (c'est-à-dire tout autre chose qu'un téléchargement de code), il intercepte les premiers octets de données envoyés à la carte après l'ouverture d'une connexion. Si une esquisse s'exécutant sur le conseil reçoit une configuration unique ou d'autres données lors de son premier démarrage, assurez-vous que le logiciel avec lequel il communique attend une seconde après l'ouverture de la connexion et avant d'envoyer ces données.

Le Mega2560 contient une trace qui peut être coupée pour désactiver la réinitialisation automatique. Les coussinets de chaque côté de la trace peuvent être soudé ensemble pour le réactiver. Il est étiqueté "RESET-EN". Vous pouvez également désactiver la réinitialisation automatique en connectant un Résistance de 110 ohms de 5V à la ligne de réinitialisation ; voir ce fil de discussion pour plus de détails.

I.10 Protection à maximum de courant USB :

L'Arduino Mega2560 a une poly-fuse ré-initialisable qui protège les ports USB de l'ordinateur contre les court-circuités et les surintensités.

Bien que la plupart des ordinateurs fournissent leur propre protection interne, le fusible fournit une couche supplémentaire de protection. Si plus que 500 mA est appliqué au port USB, le fusible cassera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit enlevé.

I.11 Caractéristiques physiques et compatibilité avec le bouclier :

La longueur maximale et la largeur de la carte PCB Méga2560 sont de 4 et 2,1 pouces respectivement, avec le connecteur USB et prise d'alimentation s'étendant au-delà de la première dimension. Trois trous de vis permettent à la planche d'être attachée à une surface ou à un boîtier.

Notez que la distance entre les broches numériques 7 et 8 est de 160 mil (0,16 "), pas un multiple pair de l'espacement de 100 mil des autres broches.

Le Mega2560 est conçu pour être compatible avec la plupart des boucliers conçus pour l'Uno, le Diecimila ou le Duemilanove. Numérique les broches 0 à 13 (et les broches adjacentes AREF et GND), les entrées analogiques 0 à 5, l'en-tête de puissance et l'en-tête ICSP sont tous emplacements équivalents. En outre, l'UART principal (port série) se trouve sur la même broche (0 et 1), de même que les interruptions externes0 et 1 (broches 2 et 3 respectivement). SPI est disponible via l'en-tête ICSP sur le Mega2560 et le Duemilanove / Diecimila. S'il vous plaît noter que I2C ne se trouve pas sur les mêmes broches sur le Méga (20 et 21) que le Duemilanove / Diecimila (entrées analogiques 4 et 5).

II. La programmation :

Description logicielle :
Le logiciel de programmation des modules Arduino est une application java, libre et multiplateformes, serve d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme à travers la liaison série (RS232, Bluetooth, ou USB selon le module).
Il est également possible de se passer de l'interface Arduino, et de compiler les programmes en ligne de commande. Le langage de programmation utilisé est le C++, compilé avec AVR g++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programme sur les plates-formes Arduino, à toute personne qui maitrise le C ou le C++.
Afin de pouvoir programmer la carte Arduino, il faudra le logiciel Arduino pour charger le code en question.

Le logiciel Arduino a pour fonctions principales :

1. Pouvoir écrire et compiler des programmes pour la carte Arduino
2. Se connecter avec la carte Arduino pour y transférer les programmes
3. Communiquer avec la carte Arduino

Le logiciel Arduino intègre également un TERMINAL SERIE (fenêtre séparée) qui permet d'afficher des messages textes reçue de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facilite des programmes, permet d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes.

III. Logiciel d'édition et programmation Arduino :

Arduino, le logiciel de programmation

Figure 3 : Interface du logiciel Arduino.

Il est totalement gratuit et proposé en libre téléchargement sur le site Internet d'Arduino. Téléchargez et installez la dernière version que vous souhaitez sur votre ordinateur (Windows.

Linux ou Mac). Une fois décompressée comprend le logiciel, les drivers L'archive, exemples et librairies. Après l'indispensable installation des pilotes, lancez le logiciel à partir du dossier. La fenêtre de l'application Arduino comporte les éléments suivants.

III.1 Structure d'un programme Arduino :

Un programme Arduino doit impérativement être divisé en 2 parties : une fonction setup() et une fonction loop(). Ces deux fonctions sont de type void(), c'est à dire qu'elles ne peuvent pas prendre de valeurs dont la fonction setup est la fonction qui se lance au début du programme, Elle permet d'initialiser les variables et de définir les broches de la carte Arduino qui seront utilisées, par contre la fonction loop se lance après la fonction setup et comme son nom l'indique elle fait une boucle jusqu'à ce que la carte soit débranchée.

Les constantes du software pour le programme, de nombreuses constantes globales sont utilisées et définies en tout début de programme.

III.2 Arduino, structure d'un programme :

La structure se faite en disciple les étapes :

1) La définition des constantes et des variables

2) La configuration des entrées et des sorties

3) La programmation des interactions et comportements

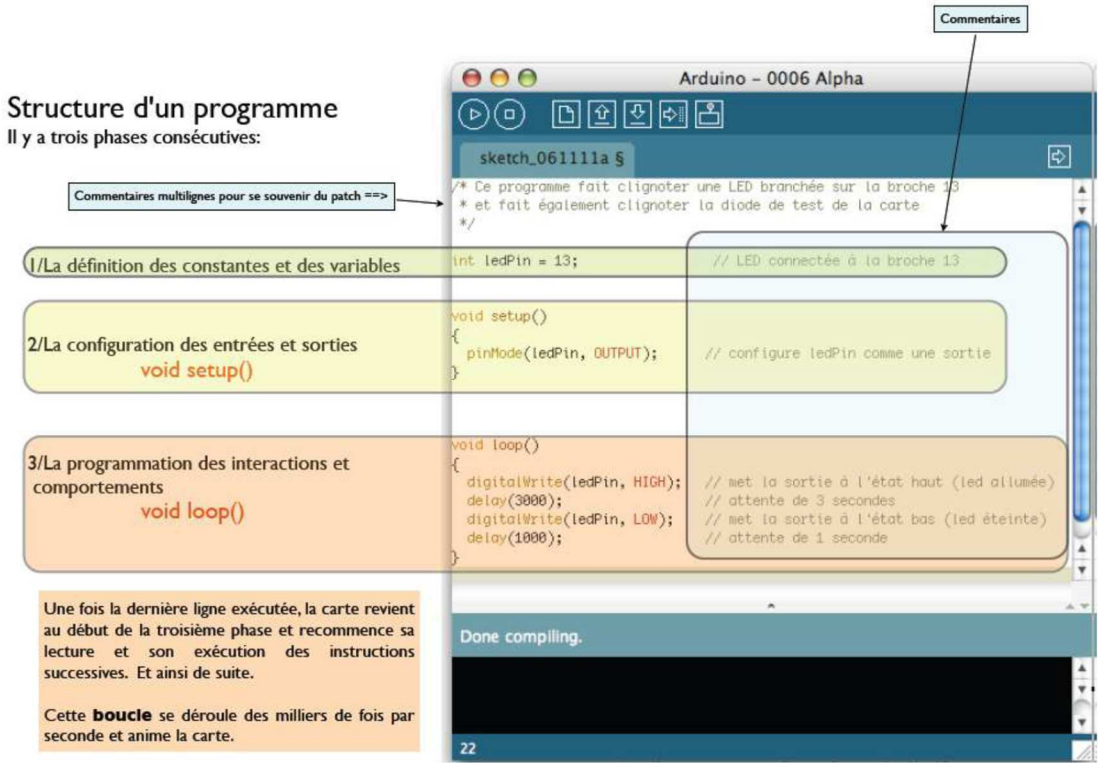


Figure 4 : structures d'un programme

IV. Arduino, le principe de fonctionnement :

1. Ouvrer un programme existant avec le logiciel Arduino.
2. Vérifier le programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, alors il faut modifier le programme.
4. Charger le programme sur la carte.
5. Câbler le montage électronique.
6. L'exécution de programme est automatique après quelques secondes.
7. Alimenter la carte soit par le port USB, soit par une source d'alimentation autonome (Pile 9 volts par exemple).
8. Vérifier le fonctionnement du montage.

V. Conclusion :

Ce chapitre est un portail pour la réalisation et la conception matérielle. Ce travail m'a permis de maîtriser les options de la carte Arduino Méga avec les circuits de commande et ses caractéristiques afin de les exploiter d'une manière correcte. J'ai présenté également des principes et des fonctions différentes de la programmation dédiée à la carte Arduino Méga, le plus puissant qu'Arduino Uno. Une brève description, ainsi qu'une figure montrant l'interconnexion des autres composants qui permettent de mener et apprendre plus traiter de multiples techniques et périphériques entrées numériques et analogiques. J'ai sélectionné la carte Arduino Méga pour son aspect économique, sa popularité et sa puissance de traitement. Ensuite j'ai présenté aussi une étude sur les circuits intégrés de commande des moteurs pas à pas utilisées dans mes cartes électroniques. Ce chapitre relève ainsi une utilité majeure pour ce qui suit puisqu'il détaille des notions exploitées au sein de la partie réalisation de mon projet.

CHAPITRE 2

ANDROID

I.ANDROID :

I.1. Plateforme Android :

Android est une couche logicielle qui inclut un système d'exploitation, un middleware et des applications clés. Google est le principal acteur, qui vient à l'esprit pour cette plateforme, mais d'autres acteurs comme les membres de l'Open Hand-set Alliance collaborent au développement. Le système d'exploitation est basé sur le noyau Linux.

La dernière version de la plateforme est maintenant Android 8.1 alias Oreo.

Android 8 introduit une variante pour les mobiles bon marché : Go Edition. Elle vise les dispositifs avec peu de RAM (1 Go ou moins), avec un accès internet limité et des processeurs moins performants. Android Go Edition est livré avec des versions adaptées des applications les plus populaires : Gmail Go, Google Go, Maps Go...

Il existait auparavant deux variantes de la plateforme. Une dédiée aux petits écrans principalement les téléphones mobiles (toutes les versions en dessous de 3.0), et une variante dédiée pour les tablettes : Honeycomb Android 3.0. (Certaines tablettes Android ne supportent pas cette version et utilisent une version Android 2.x).

Android 4 ou "Ice Cream Sandwich", est sorti en octobre 2011 elle fusionne les deux variantes pour avoir une plateforme plus versatile et uniforme. C'est la première version qui combine "Gingerbread" et "Honeycomb" pour une plateforme à la fois pour les tablettes et les téléphones.

Android 5.0 ou Lollipop cible encore plus d'appareils tel que les smart watches, les lecteurs pour la télévision, ou dans la voiture. Les appareils avec seulement 512 de mémoire peuvent supporter cette nouvelle version.

Note : Les chiffres des parts de marché (Déc. 2017) sont basés sur des statistiques qui utilise l'API level, donc les pourcentages regroupent plusieurs sous-versions de la plateforme Android.

I.2. L'apparence différente d'Android sur les smartphones :

Il faut prendre en considération que les constructeurs de smartphones utilisent la version Open Source d'Android à laquelle ils ajoutent les composants de Google (afin d'avoir le Play Store, entre autres).

La version la plus 'pure' d'Android est appelée Android Stock, ou encore parfois Android Vanilla. C'est celle qui est utilisé sur les appareils de Google, c'est à dire sur la gamme Nexus. Beaucoup de fans d'Android, considérés comme puristes par les autres, préfèrent l'interface d'Android Stock que celle des autres fabricants.

Certains fabricants utilisent une interface proche d'Android Stock, par exemple OnePlus, alors que d'autres s'en éloignent beaucoup, comme par exemple la marque Meizu avec son interface Flyme OS. Nous pouvons citer également l'interface TouchWiz de Samsung, Xperia UI de Sony ou encore EMUI (Emotion UI) de Huawei et Honor.

I.3 Le Play Store :

Le Play Store est l'espace virtuel d'Android dans lequel les utilisateurs peuvent télécharger des applications, des jeux, des livres, des films, de la musique etc. Il est extrêmement riche, vous y trouverez très probablement les applications que vous recherchez.

Ce que vous achetez sur le Play Store est associé à votre compte Google, nécessaire pour vous procurer des applications. Vous y aurez accès à partir de n'importe quel appareil connecté avec votre compte.

Figure 5 : symbole d'Android

II. Les principaux systèmes d'exploitation d'un Smartphone :

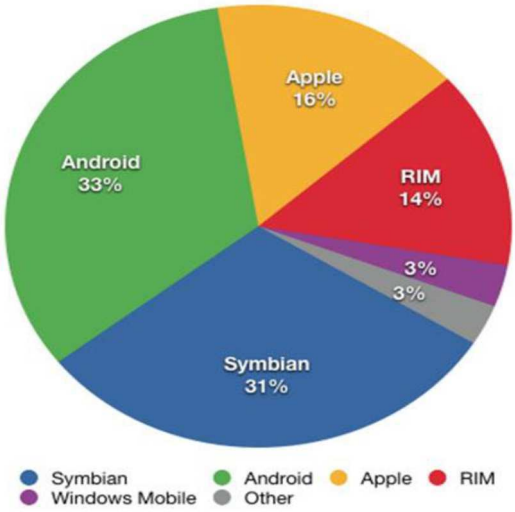


Figure 6 : Les principaux systèmes d'exploitation

II.1. L'Android :

Android est un système d'exploitation ouvert (Open Source) pour tablettes tactiles, terminaux mobiles et TV connectées. Il a été conçu en 2007, par la société Android, une start-up rachetée par Google. C'est un système d'exploitation fondé sur un noyau Linux Disponible grâce à une licence Apache, ce système inclut tous les utilitaires requis par un constructeur pour le mettre en œuvre dans un téléphone portable.il est proposé à tous les fabricants de téléphones mobiles, pour faciliter son adoption. Cependant, Android est modifiable par les constructeurs. Donc, les versions varient largement d'un constructeur à l'autre.

II.2. Les versions d'Android :

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.9%
4.1.x	Jelly Bean	16	3.5%
4.2.x		17	5.1%
4.3		18	1.5%
4.4	KitKat	19	20.0%
5.0	Lollipop	21	9.0%
5.1		22	23.0%
6.0	Marshmallow	23	31.2%
7.0	Nougat	24	4.5%
7.1		25	0.4%

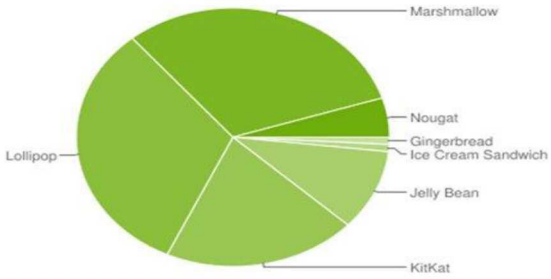


Figure 7 : Les versions d'Android

II.3. L'architecture de la plateforme d'Android :

L'architecture de la plateforme Android se décline selon une démarche bottom up en quatre principaux niveaux que sont le noyau linux, les librairies et la plateforme d'exécution, le module de développement d'applicatifs et enfin les différentes applications**.**
Chacun de ces niveaux est décrit plus en détail ci-dessous.

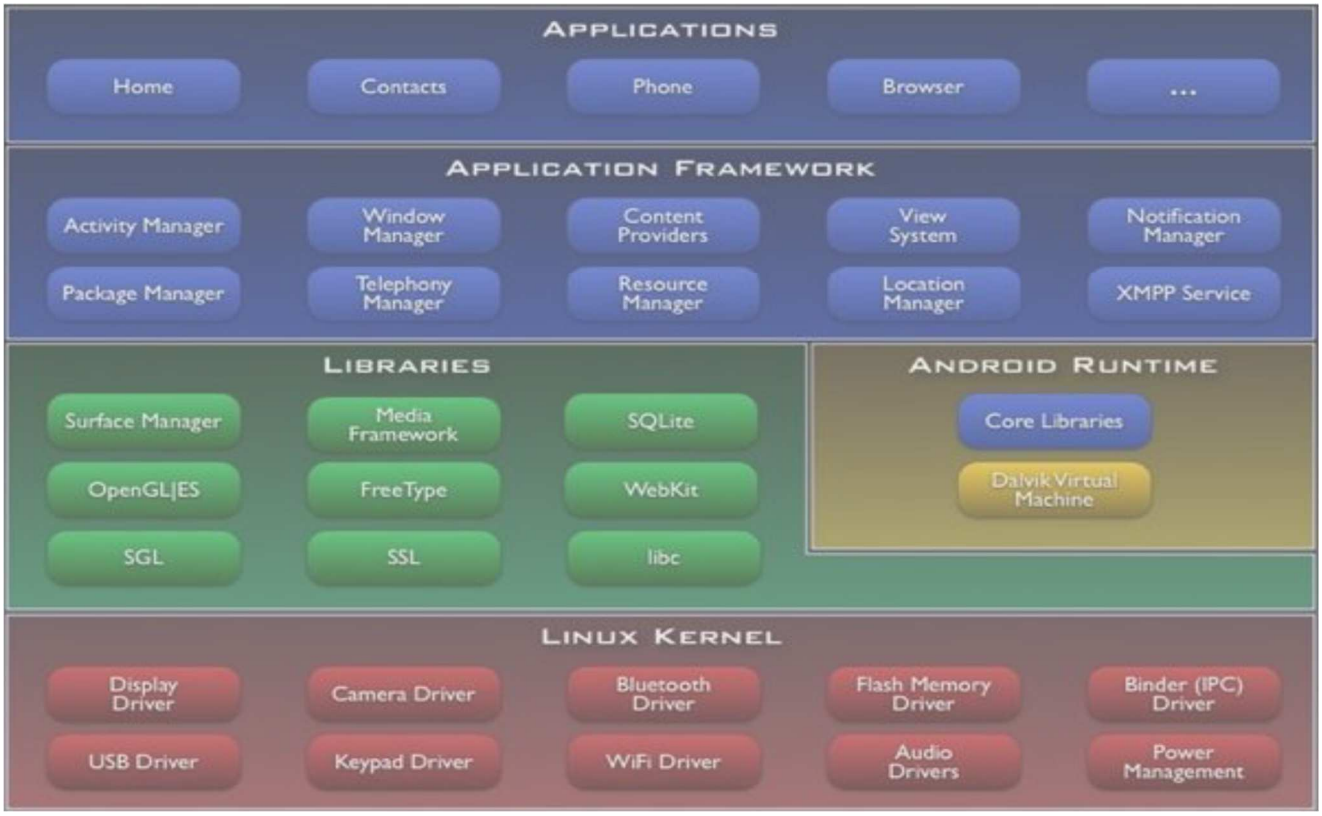


Figure 8 : L'architecture de la plateforme d'Android

II.3.1. Premier niveau : Le noyau Linux :

Android s'appuie sur un noyau Linux 2.6 qui agit également comme une couche d'abstraction entre le matériel et le reste de la pile logicielle sur laquelle vient s'agréger différents services tels que la sécurité, le gestionnaire de mémoire, le gestionnaire des processus et la pile réseau.

II.3.2. Deuxième niveau : Les librairies :

Les librairies natives sont écrites en langage C et C++.

- Le Surface Manager est chargé de la composition des items sur l'écran, de la gestion du dispositif d'affichage. Il permet de s'assurer que les pixels s'affichent bien à l'écran.
- Open GL/ES quant à lui gère le graphisme en 3D tandis que SGL gère l'affichage en 2D. Ainsi une même application peut combiner du 2D avec du 3D.
- Le Media Framework fourni par la société PacketVideo (membre de l'OHA) contient des codes audio et média (Mpeg 4, H.264, AAC, MP3…).
- Le Free type est une bibliothèque logicielle open source qui implémente un moteur de rendu de police de caractère.

- Le SQLite est une bibliothèque open source écrite en C permettant d'implémenter un moteur de base de données relationnelle.

Niveau adjacent : l'environnement d'exécution

- Le Dalvik Virtual Machine exécute des fichiers de type «.dex» qui est en fait le résultat en byte codes de la conversion de fichier «.class» et «.jar». Il permet un usage de la mémoire, un partage entre processus plus efficaces. C'est un interpréteur de byte code optimisé. Il est possible d'avoir plusieurs instances de DVM s'exécutant au même moment.
- Les «Core Librairies» écrit en java est un ensemble de collection, de classes, d'utilitaires d'entrée/ sortie.

II.3.3. Troisième niveau : le module de développement d'application :

Un Framework fournit un ensemble de fonctions facilitant la création de tout ou d'une partie d'un système logiciel, ainsi qu'un guide architectural en partitionnant le domaine visé en module.
L'Application Framework développée en java contient un certain nombre d'applications dédiées (application téléphonique, des applications écrites par Google ou par un tiers). Toutes les applications peuvent utiliser le même API.
« Activity Manager » permet de gérer le cycle de vie d'une application (application en tâche de fond par exemple).

- Le « Packet manager » garde une trace des applications installées dans l'équipement. Si l'on télécharge une nouvelle application par exemple, le packet manager informe sur la capacité du système.
- Le «Windows manager» s'occupe de gérer la fenêtre d'affichage.
- Le «Telephony manager» contient des API pour la construction d'une application téléphonique.
- Le «Content provider» permet le partage de données, l'interaction avec d'autres applications (répertoire, numéro de téléphone, dont on a besoin les autres applications).
- Le «Ressource Manager» quant à lui stocke les bitmaps locaux.

II.3.4. Quatrième niveau : les applications :

Niveau d'abstraction dans lequel on peut trouver toutes les applications spécifiques au fonctionnement d'un Smartphone (téléphone, répertoire, navigateur web…).

III.4 Les avantage d'Android :

III.4.1. Open source :

Le contrat de licence pour Android respecte les principes de l'*open source*, c'est-à-dire qu'on peut à tout moment Télécharger les sources et les modifier.

III.4.2. Gratuit (ou presque) :

Android est gratuit, autant pour l'utilisateur que pour les constructeurs.
Les développeurs sous Android peuvent poster leurs applications développées sur le Play store (APK sur Android) Et cela coutera 25$, ces 25 $ permettent de publier autant d'applications souhaitées, à vie.

III.4.3. Facile à développe :

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale.

III.4.4. Facile à vendre :

Le Play Store (anciennement Android Market) est une plateforme immense et très visitée ; c'est donc une mine d'opportunités pour quiconque possède une idée originale ou utile.

III.4.5. Flexible :

Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. Les Smartphones, les tablettes, la présence ou l'absence de clavier ou de *trackball*, différents processeurs... On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android !
Non seulement c'est une immense chance d'avoir autant d'opportunités, mais en plus Android est construit de manière à faciliter le développement et la distribution en fonction des composants en présence dans le terminal.

III.4.6. Ingénieux :

L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leur développement. Ces applications se trouvent essentiellement sur internet et leur principe est que vous pouvez combiner plusieurs composants totalement différents pour obtenir un résultat surpuissant. Par exemple, si on combine l'appareil photo avec le GPS, on peut poster les coordonnées GPS des photos prises.

IV. Développement sous Android :

IV.1 L'environnement de développement sous Android :

Afin de développer des applications sous Android, un ensemble d'outils est nécessaire. Vue que les procédures d'installation de ces outils sont assez longues et fastidieuses, alors les décrire, pas à pas, risquerai de prendre énormément de place et ainsi beaucoup de pages. Alors, on se contentera juste d'évoquer les outils et leur intérêt.

IV.2 Le JDK (Java Développement Kit) :

Les applications développées pour Android étant essentiellement écrites en langage java ; un langage de programmation orienté objet qui a la particularité d'être très portable. Cela signifie qu'un programme Java, fonctionnant sur Windows (par exemple), pourra facilement tourner sur Mac ou GNU/Linux.
Cette petite exploit vient du fait que Java s'appuie sur une machine virtuelle pour s'exécuter (appelée la JVM). Pour avoir une JVM sur votre ordinateur, il vous faut télécharger le JRE. Ce dernier contient, en plus de la JVM, des bibliothèques Java standards.
La JVM ne lit pas directement le code Java. Elle lit un code compilé (le byte code). Pour passer du code Java, que le développeur écrit, au code compilé, lu par la JVM, des outils spéciaux sont nécessaires. Ces outils sont inclus dans le JDK. De plus, le JDK contient le JRE (et donc la machine virtuelle), ce qui est bien pratique Pour résumer, on dira que :
-Pour un simple utilisateur de Java : il doit avoir le JRE ;
-Pour un développeur : il aura besoin des outils du JDK.

IV.3 Le SDK (Software Développent Kit) Android :

Un SDK, c'est-à-dire un kit de développement logiciel, est un ensemble d'outils que met à disposition un éditeur afin de permettre de développer des applications pour un environnement précis. Le SDK Android permet, donc, de développer des applications pour Android et uniquement pour Android.

Au premier lancement du SDK, un écran semblable à la Fig. (9) s'affichera :

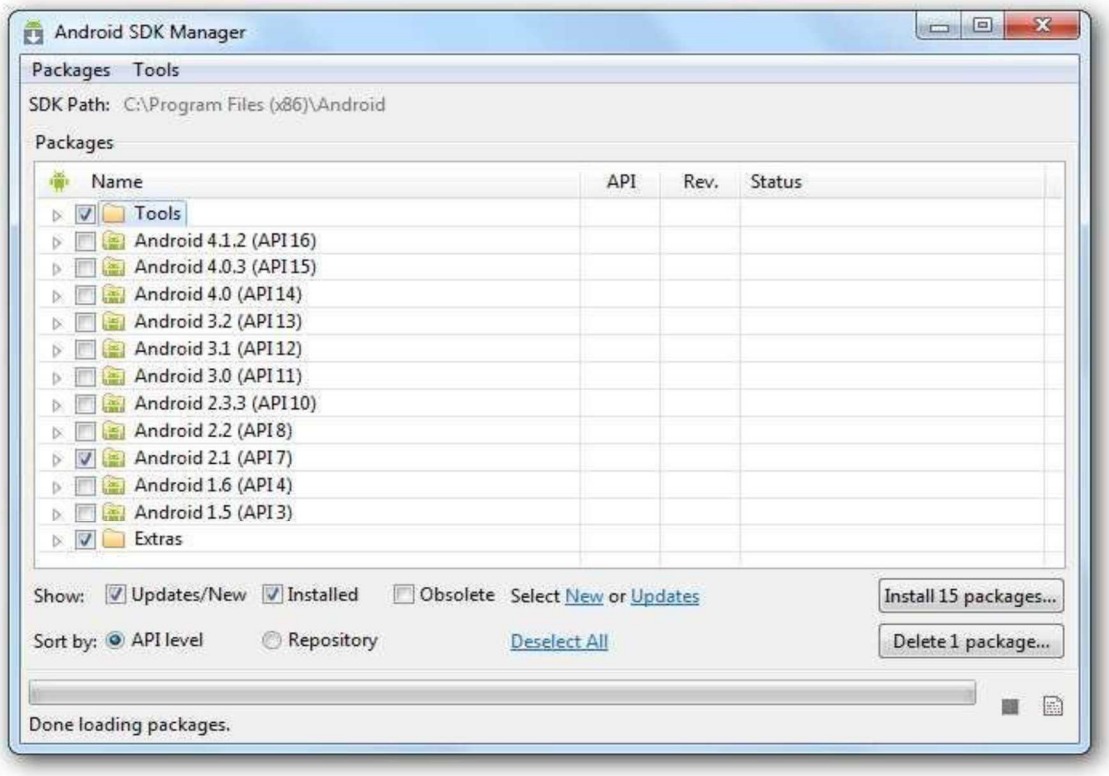


Figure 9 : Le SDK Android

En regardant bien le nom des paquets, vous remarquerez qu'ils suivent tous un même motif. Il est écrit à chaque fois : Android [un nombre] (API [un autre nombre]). La présence de ces nombres s'explique par le fait qu'il existe plusieurs versions de la plateforme en circulation. Le premier nombre correspond à la version d'Android et le second à la version de l'API Android associée. Quand on développe une application, il faut prendre en compte ces numéros, puisqu'une application développée pour une version précise d'Android ne fonctionnera pas pour les versions antérieures.

IV.4 L'IDE Eclipse :

Eclipse est un environnement de développement intégré. C'est un logiciel qui permet d'écrire un programme beaucoup plus facilement qu'avec le simple Bloc-notes. Outre la coloration du code, il permet d'apporter des outils très pratiques pour compiler vos programmes, les déboguer, etc. Il peut être utilisé pour programmer avec n'importe quel type de langage, mais nous l'utiliserons pour faire du Java.
De plus, Eclipse est conçu pour pouvoir être complété avec des plugins (extension). Ainsi, il existe un plugin pour développer des applications Android que nous verrons dans la partie suivante.

IV.5 Le plugin ADT pour Eclipse :

Google fournit un plugin pour Eclipse, nommé ADT (Android Développent Tools).
La fonction principale de ce plugin est de créer un pont entre Eclipse et le SDK Android.

IV.6 L'émulateur de téléphone : Android Virtual Device :

L'Android Virtual Device, aussi appelé AVD, est un émulateur de terminal sous Android, s'est-il- dire que c'est un logiciel qui fait croire à votre ordinateur qu'il est un appareil sous Android. C'est la raison pour laquelle vous n'avez pas besoin d'un périphérique sous Android pour développer et tester la plupart de vos applications En effet, une application qui affiche un calendrier par exemple peut très bien se tester dans un émulateur, mais une application qui exploite le GPS doit être éprouvée sur le terrain pour que l'on soit certain de son comportement.

Figure 10 : L'émulateur Android

V. Le développement d'applications sous Android studio :

Comme nous avons déjà entamé le langage de programmation « java », utilisé pour créer des applications Android, Android Studio est l'un des logiciels qui utilise lui aussi ce langage et qu'on utilise pour le développement des plusieurs applications.

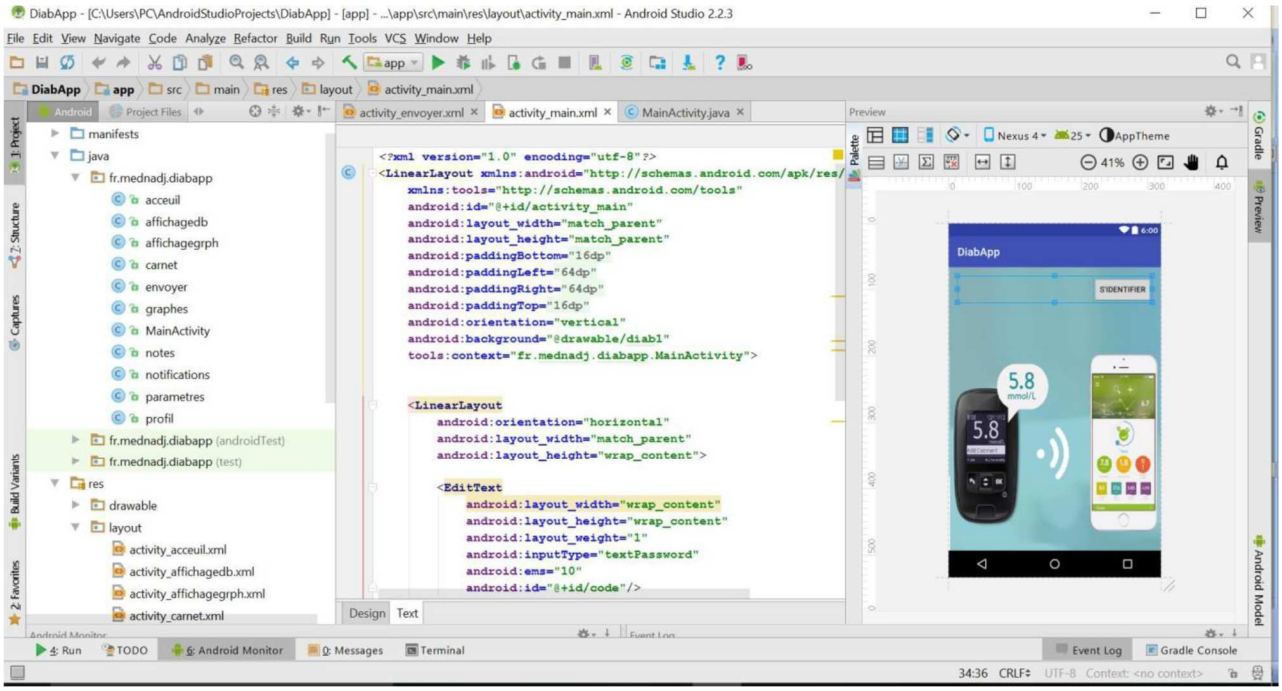


Figure 11 : L'aire de développement Android Studio

V.1 Architecture d'un projet Android studio :

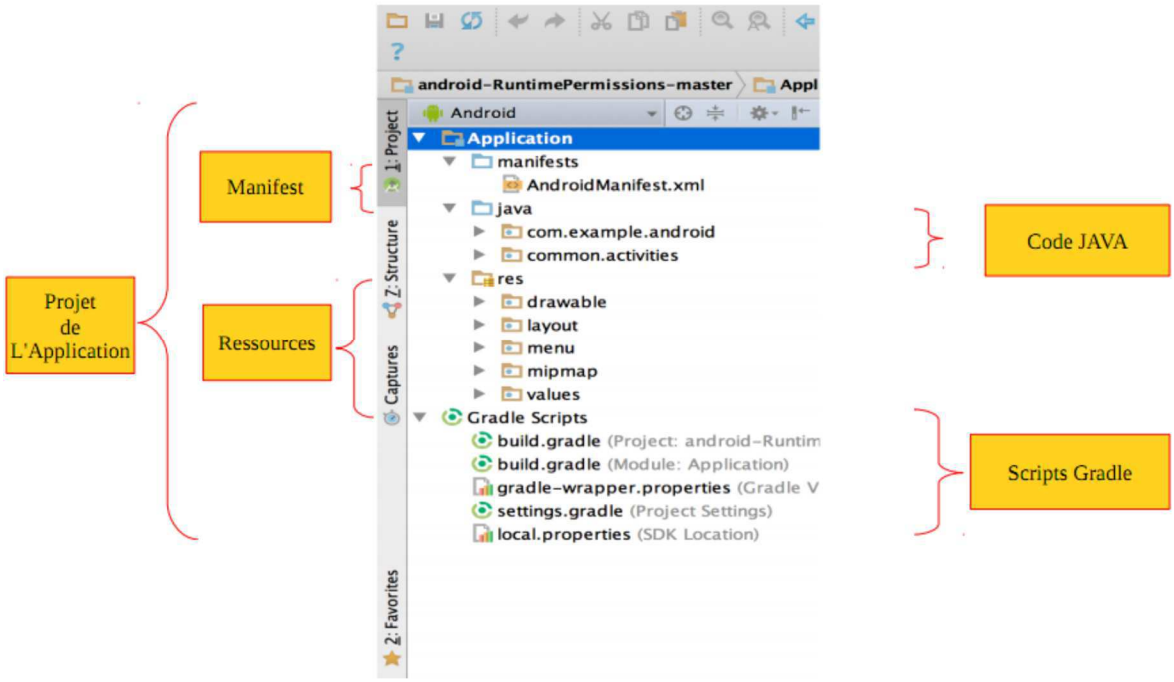


Figure 12 : Architecture de l'interface d'Android studio

Java : Contient le code source écrit en java classé par paquet. Dans notre projet, le paquet une activité se nommant « MainActivity ».

RES : Contient toutes les ressources de notre application c'est à dire les images, les couleurs, les textes dans toutes les langues, les thèmes… et surtout les interfaces graphiques.

Dans **RES** il y a les répertoires :

Drawable : Ils sont nombreux car il y en a un par type d'écran. On y met par exemple les images de votre application adaptée à chaque type d'appareil.

Layout : C'est le répertoire par défaut pour les interfaces graphiques.

Menu : Il permet de définir les menus qui sont proposés par les différentes activités.

Values : Ce répertoire contient par défaut les fichiers **dimens.xml** pour centraliser des informations de dimensionnement comme les marges et autres espacements entre les éléments de l'interface graphique, **strings.xml** qui contient les textes de votre application (texte des menus, texte des boutons…), et **styles.xml** qui permet de définir le style de votre application (Android propose des styles pour débuter facilement).

V.2 Application DiabApp :

Le Menu principal de DiabApp contient six Choix pour l'utilisateur, à savoir :

- Carnet glycémique
- L'alarme
- Les courbes « Graphes »
- Paramètres
- Les notes
- Le partage de données entre Médecin/patient

Cette application est conçue pour une télésurveillance locale et à distance.

V.2.1 Captures d'images de l'application « DiabApp » :

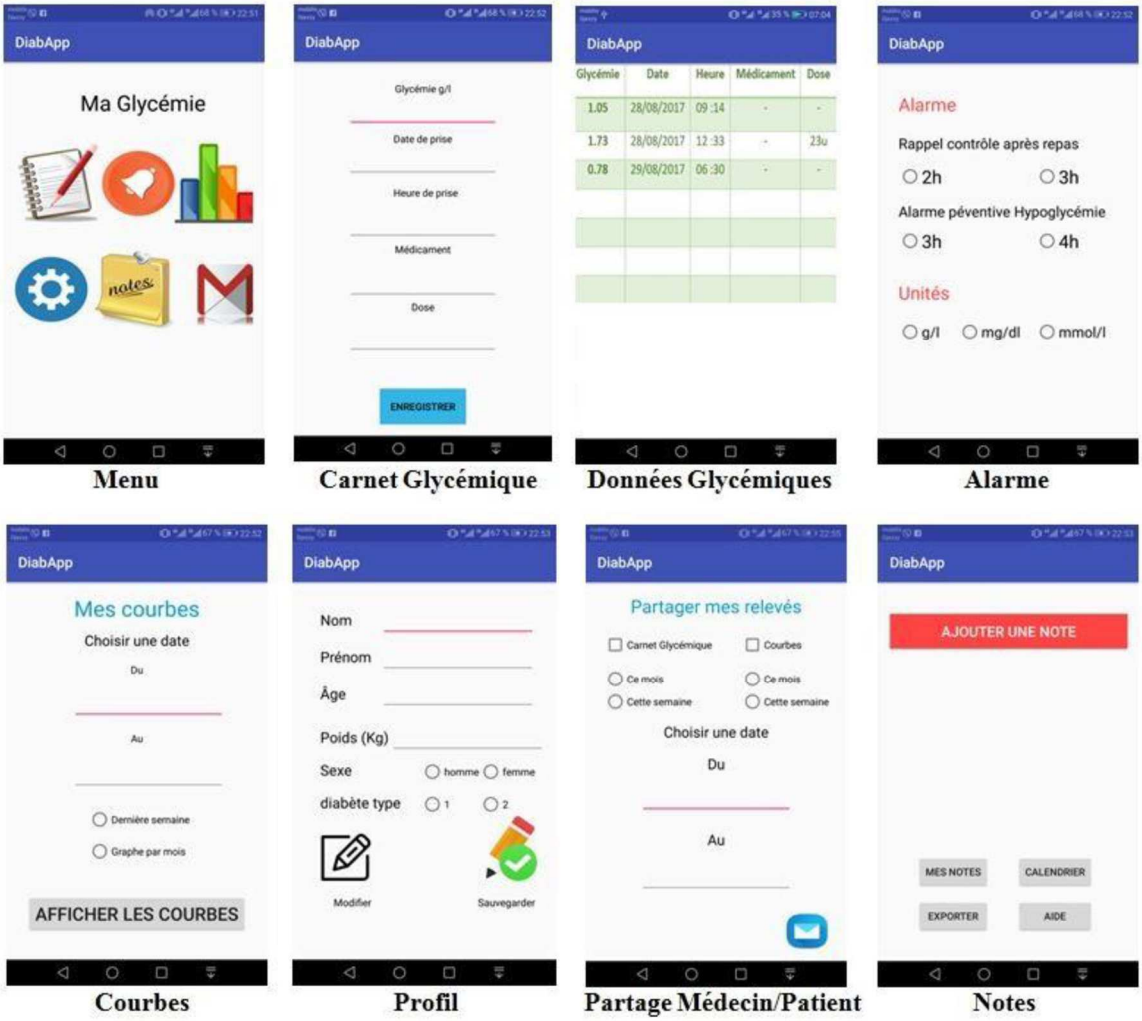


Figure 13 : Images représentatives de l'application « DiabApp »

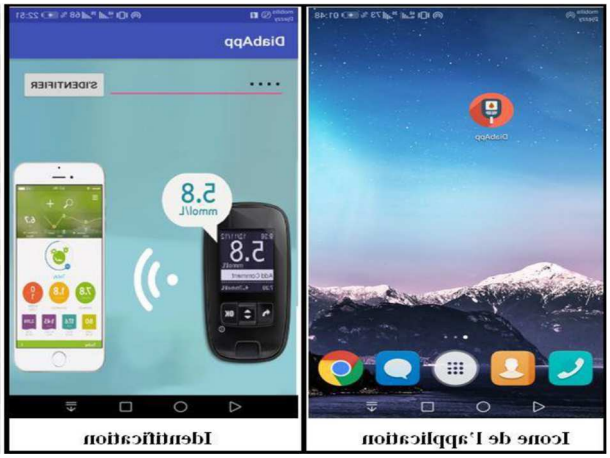


Figure 14 : Accès à l'application DiabApp

V.2.2 Principe de DiabApp :

Après l'obtention de la valeur du taux de glucose sur l'application DiabValues (Nous y reviendrons plus tard), la personne diabétique qui détient l'application, entrepose la valeur ainsi que les autres données (date, heure, médicament « dans le cas d'un diabète type 2 », dose d'insuline « dans le cas d'un diabète type 1 »). Ensuite, la sauvegarde se fait sur la base de données SQLite pour un éventuel partage avec le médecin et la réalisation des courbes pour un bon suivi de la maladie.

L'application dispose aussi, d'un gestionnaire de notes, de configurateur d'alarmes afin de bien gérer la maladie. Bien entendu, la personne enregistre son profil (Nom, prénom, âge, poids …) Fig. (), ces informations utiles pour le médecin traitant.

V.3 App Inventor pour Android :

Est une application développée par Google. Elle est actuellement entretenue par le Massachusetts Institute of Technology (MIT).

Elle simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont pas familiers avec les langages de programmation. Elle est basée sur une interface entièrement graphique en absence totale de ligne de code, elle est particulièrement adaptée à l'initiation à la programmation sous Android.

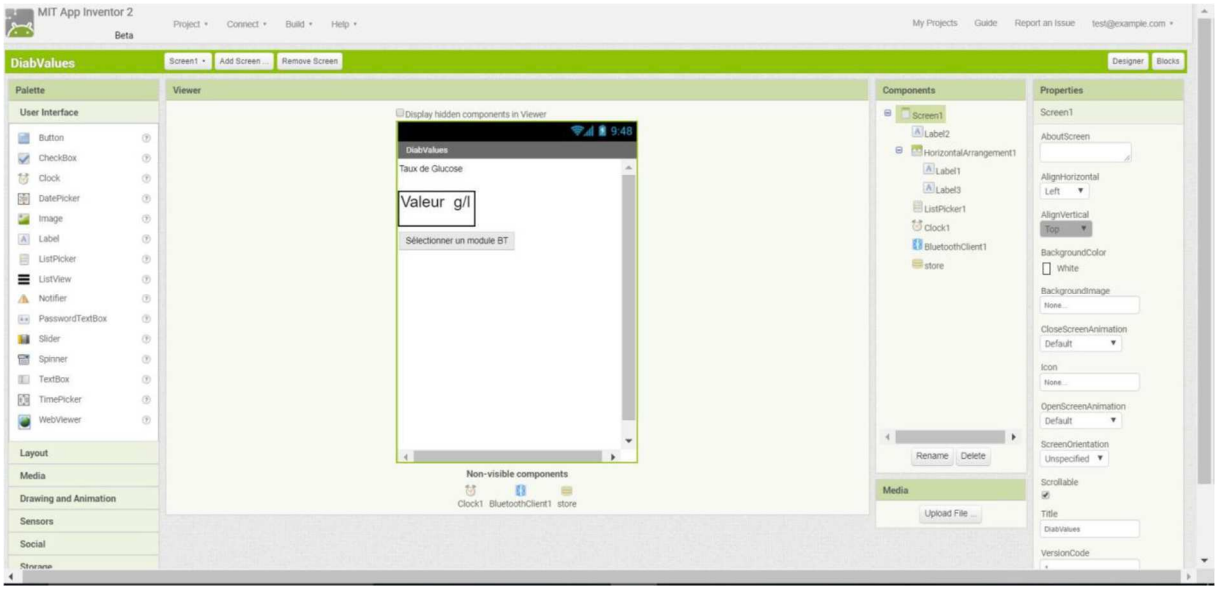


Figure 15 : L'aire d'App Inventor 2

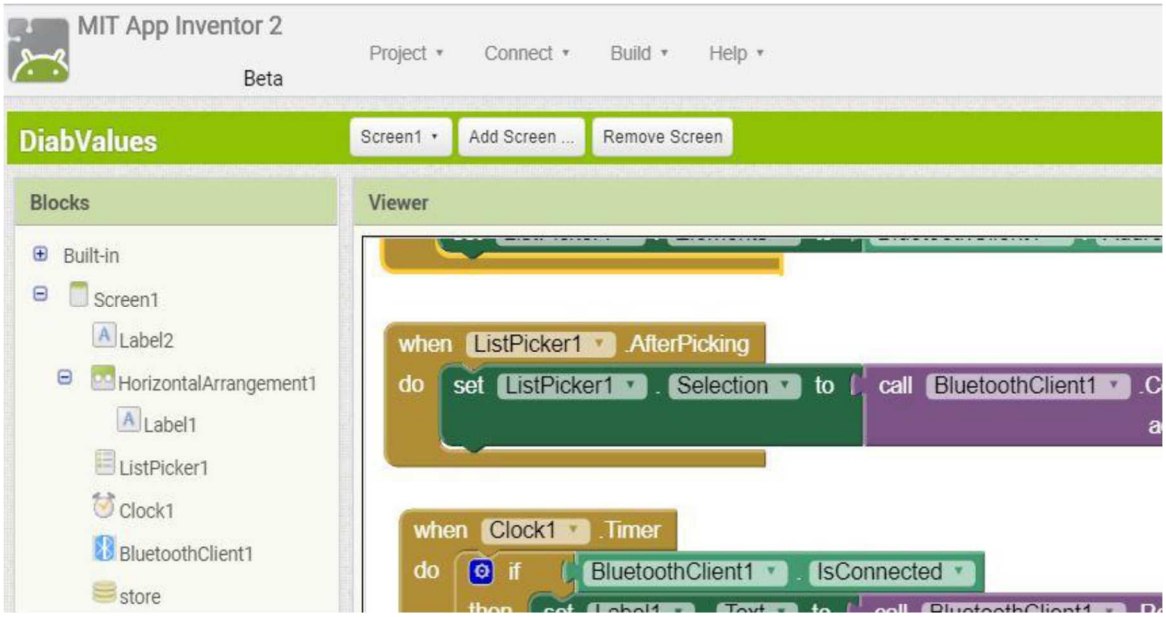


Figure 16 : Programmation sous App Inventor 2

V.4. Application DiabValues :

Cette application développée sous App Inventor 2 consiste à réceptionner la donnée émise de la carte Arduino via le module Bluetooth exemple HC-06.
Elle comporte

- Un bouton pour la connexion Bluetooth
- Un champ pour l'affichage du taux de glucose reçu.

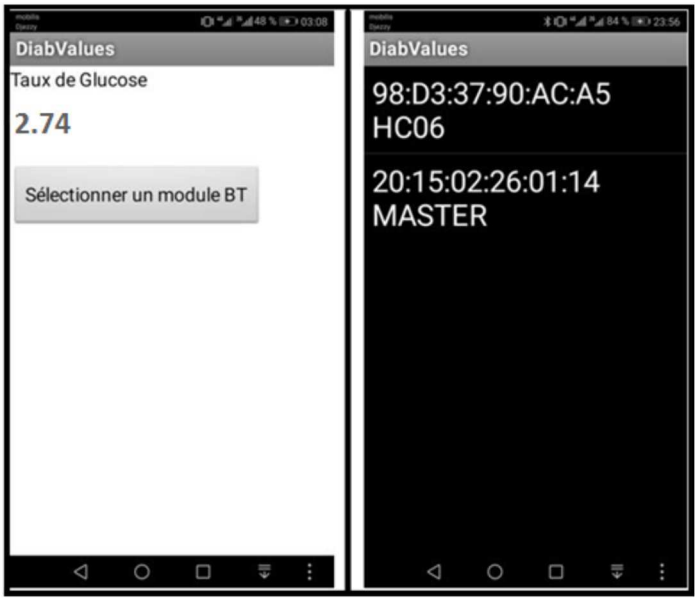


Figure 17 : L'application « DiabValues »

Une fois que le Bluetooth est associé au Smartphone « connexion établie », on aura l'affichage de la valeur du taux de glucose en temps réel, acquise depuis la carte Arduino.

VI. Conclusion :

Dans ce chapitre, j'ai présenté l'application qui se résume aux données de mes circuits, ces tâches sont effectuées par une application créée dans l'environnement App Inventor 2. La deuxième : l'application est destinée à la télécommande du procès(CNC) par Bluetooth.

CHAPITRE 3

MACHINE CNC

I. HISTORIQUE :

C'est en 1942 aux États-Unis que la C.N. a commencé à être exploitée, pour permettre l'usinage de pompes à injection pour moteurs d'avions. Il s'agissait en fait de l'usinage de cames, dont le profil complexe était irréalisable au moyen d'une machine traditionnelle

II. INTRODUCTION :

Les machines CNC (*Computer Numerical Control*) permettent d'effectuer automatiquement de nombreuses tâches dans divers domaines d'activité : la gravure et le perçage des circuits imprimés, la réalisation de petites pièces mécaniques, le traçage et la découpe des divers éléments constituants les modèles réduits, et bien d'autres choses encore.

Un nombre sans cesse croissant de modélistes, d'électroniciens et d'autres passionnés utilisent des machines-outils contrôlées numériquement. Ces petites machines, le plus souvent des fraiseuses, permettent d'obtenir un résultat de qualité quasi professionnelle.

Mais ces machines étant d'un prix d'achat souvent très élevé, beaucoup
D'utilisateurs les conçoivent et les réalisent eux-mêmes. La plupart des machines CNC possèdent trois axes, mais il existe des modèles à quatre, voire cinq axes. Les modèles à trois axes sont souvent suffisants dans la majorité des applications.
C'est la fabrication d'un modèle simple à trois axes que nous décrivons dans cette
Documentation. Toutes les étapes de la réalisation y sont détaillées, y compris l'électronique de commande et l'alimentation de puissance.

III. Description de la machine CNC (Exemple) :

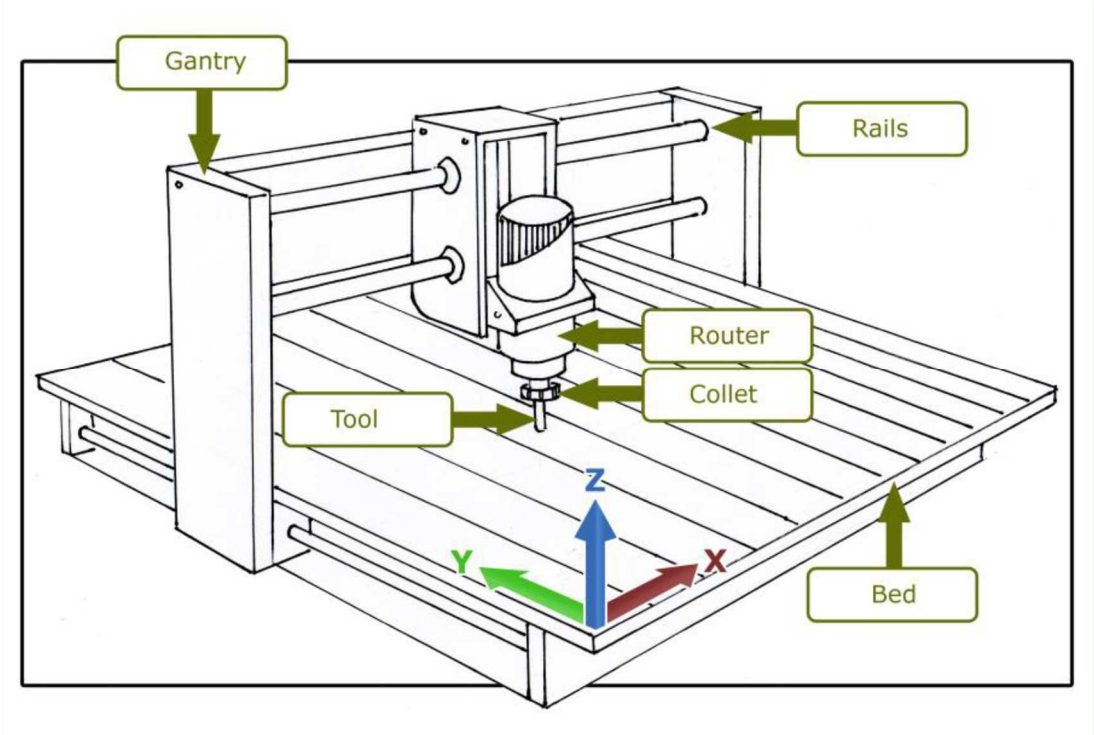


Figure 18 : Les principaux systèmes d'exploitation

Généralement parlant, un «CNC Router» par exemples est une machine contrôlée par ordinateur sur laquelle est montée une toupie ou une broche qui contient un outil de coupe (mèche). Il est typiquement mis en place avec 3 directions de mouvement appelées les axes X, Y et Z. La position du routeur est déterminée par un ordinateur indiquant aux moteurs montés sur chaque axe combien déplacer dans chaque direction.
En utilisant cette méthode de positionnement, n'importe quel emplacement dans la zone de travail des machines (enveloppe) peut être défini et le routeur peut être déplacé dans cet espace. Lorsque la machine est conduite par un ordinateur lui indiquant où se déplacer, l'opérateur utilise un logiciel pour dessiner les formes à couper et créer le chemin que la machine suivra.
Pour les besoins de ce document, je référerai simplement aux machines et à la technologie en tant que "CNC". Comme pour tout sujet technique, il existe de nombreuses variantes et nuances dans la terminologie, mais à ce stade, je vais essayer de le garder aussi simple que possible. L'image ci-dessous montre une disposition typique pour une commande numérique par ordinateur (il y a beaucoup de variations cependant) et un certain nombre de composants clés avec un indicateur des directions possibles de mouvement (X, Y et Z).
Contrôle numérique par ordinateur CNC est une forme spécialisée et polyvalente d'automatisation douce et ses applications couvrent de nombreux types.

IV. Comportement de CNC

IV. 1 Les différentes utilités de CNC :

Outils utilisables variés :

La CNC Machine pourra être utilisée :

- en simple « machine à dessiner » à l'aide d'un crayon, d'un marqueur ou même d'un pinceau.
- en découpeuse vinyle
- en machine de perçage automatisée
- en machine de gravure de PCB
- en machine de découpe 2D de matériau par fraisage,
- voire probablement à terme de fraisage 3D de petites dimensions

Figure 19 : outils utilisables varies au machine CNC

IV.2 Les outils, Modulaire et comprend :

- 1 bloc de translation pour l'axe X (1)
- 1 bloc de translation pour l'axe Y (2)
- 1 bloc de translation d'outil (3)

Propulsée par une électronique programmable standard et open source !

- simple carte Arduino MEGA
- carte d'interface CNC-shield
- étages moteurs

IV.3 Le Firmware :

- Simple G-Code : interpréteur de G-Code dédié (Simple G Code Sender) seulement compatible Arduino MEGA
- Compatible également GRBL

IV.4 Kit de base :

Le kit de l'Open Maker Machine comprend :

- toutes les **pièces mécaniques** utiles :
- les **moteurs et coupleurs d'axes**,
- l'**électronique de commande** : carte Arduino MEGA, le shield CNC, les drivers de moteur, les fins de course,
- l'**alimentation** et ses câbles,
- quelques **supports d'outils** pratiques (porte crayon, etc...),
- éventuels **accessoires** de montage utiles,
- etc.

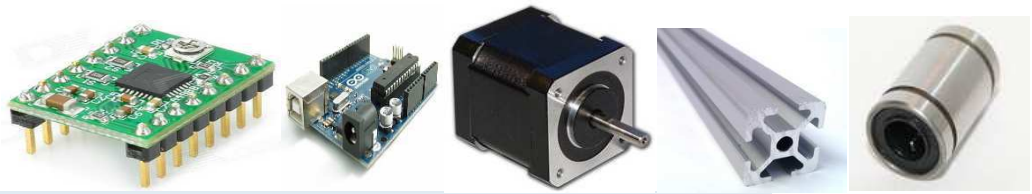


Figure 20 : pièces et composant utilisés à la machine CNC

Pièces mécaniques : composant le corps de la machine CNC y compris les axes.

Electronique :

- Carte Arduino MEGA
- Bluetooth shield
- Drivers moteurs
- Fins de courses + les câbles

Alimentation :

- Alimentation : 12V et 5V

IV.5 Simple G-Code Décoder :

Tenant dans une carte Arduino. Au final, à ce jour le Simple G-Code Décoder ne fait que 10Ko, bien qu'étant capable de décoder la plupart des instructions de G-Code usuelles aussi bien en mode absolu que relatif.

Les avantages d'un tel firmware simplifié sont multiples :

- il est facile de l'adapter à ses propres besoins sans passer des heures à comprendre la structure du code et de ses multiples onglets,
- on dispose de nombreuses broches E/S encore disponibles pour ajouter des dispositifs, etc... au besoin,
- il permet de facilement mettre en place une machine minimale de 1 à 3 axes, qui pourra être contrôlée par du vrai G-Code.

Le Simple G-Code Décoder a été testé et validé sur une machine CNC "grand format" avant d'être implémenté sur l'Open Maker Machine. Le brochage utilisé est par ailleurs compatible avec le CNC-SHIELD et le firmware GRBL (voir ci-dessous), permettant très simplement de faire évoluer sa machine si on le souhaite.

IV.6 Firmware GRBL :

Le firmware GRBL est un micro-logiciel existant, compatible Arduino UNO et faisant 27Ko. Il est plus évolué que le Simple G-Code Décoder et satisfera les plus exigent.

Ceci étant, il nécessite un câblage complet de tous les capteurs d'emblée, et fonctionnera de façon erratique sinon. Cette option est donc à la fois un avantage et un inconvénient :

- un avantage, car clé-en-main pour une machine complète
- mais un inconvénient, car hermétique dans son fonctionnement à moins d'y plonger sérieusement.
- un inconvénient car la plupart des broches de la carte Arduino sont utilisées par le firmware, rendant difficile des adaptations.

Le firmware GRBL, sous licence libre, dispose de son propre dépôt GitHub.

IV.7.G-Code :

G-code est un langage pour le contrôle numérique du mouvement de la machine. Il remonte aux années 1950. Alors qu'il y a un standard, il existe plusieurs variantes.
La CNC / GRBL utilise un sous-ensemble du langage standard, juste les essentiels communs.
G-code est un tas de lignes de texte. Chaque ligne (aussi appelée "bloc") est constituée d'un code, souvent avec paramètres, et peut-être quelques codes supplémentaires.

Précisément le fichier G-Code est un ensemble de commandes de mouvement et d'état dans un fichier texte. Il est coutume de donner au fichier une extension comme .nc, mais tout le Universal Gcode Sender se préoccupe vraiment, c'est qu'il s'agit d'un fichier texte brut. OS Préférerait que les fichiers texte aient l'extension .txt. Ça marche. (GRBL a également une limite de longueur de ligne de soit 50 ou 70 caractères, selon la façon dont il est configuré.) Il existe plusieurs façons d'obtenir un fichier G-Code

.

• en écrire un dans un éditeur de texte. C'est probablement un bon exercice pour écrire un fichier simple et essayer de l'exécuter.
Probablement écrire un fichier simple pour tester les taux d'alimentation avant de créer le fichier pièce.
• écrire un programme pour générer des fichiers G-Code. Cela peut être utile pour générer des coupes basées sur modèles mathématiques.
• utiliser un programme de FAO pour générer le code G basé sur une conception CAO ou une conception graphique. Pour Exemple,
Inkscape peut être utilisé pour générer un fichier .svg (graphique vectoriel évolutif) et le Web Programme MakerCam.com peut être utilisé pour générer le fichier G-Code.

IV.8. Les moteurs pas à pas :

Un **moteur pas à pas** permet de transformer une impulsion électrique en un mouvement angulaire. Ce type de moteur est très courant dans tous les dispositifs où l'on souhaite faire du contrôle de vitesse ou de position en boucle ouverte, typiquement dans les systèmes de positionnement.
L'usage le plus connu du grand public est dans les imprimantes reliées à un ordinateur (positionnement de la tête d'impression et rotation du rouleau porte-papier dans les imprimantes matricielles, à marguerite et à jet d'encre, et rotation du rouleau porte-papier seulement dans les imprimantes à xérographie à laser).
Deux moteurs pas à pas.
On trouve trois types de moteurs pas à pas :
• le moteur à réluctance variable ;
• le moteur à aimants permanents ;
• le moteur hybride, qui est une combinaison des deux technologies précédentes.

.

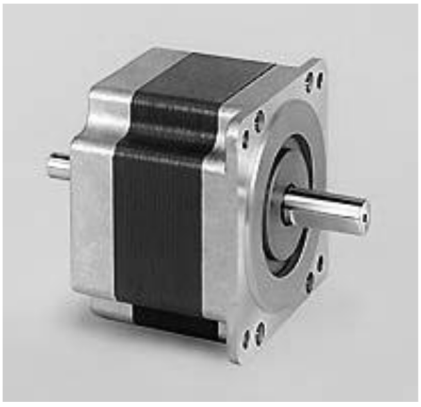
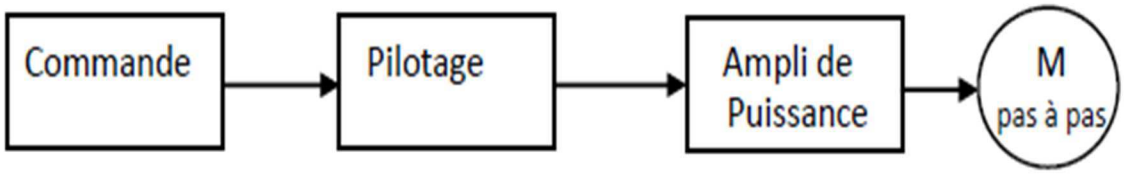


Figure 21 : moteur pas à pas

Et le principe de commande d'un moteur pas à pas :



CHAPITRE 4
La Conception

I. Introduction :

Ce chapitre est consacré à la fabrication et l'application d'une machine CNC et pour cette Raison, plusieurs étapes ont été envisagées :

- Simulation sur logiciel Isis Proteus : pour valider la faisabilité et le bon Fonctionnement des circuits déjà développées théoriquement. Implémentation de Circuit simulés sur la plaque à essais (plaque à trous) afin de s'assurer de son Fonctionnement.
- Une fois la simulation est bien vérifiée, on passe à l'étape de routage en utilisant L'outil Aigle. Réalisation de la carte électronique en imprimant les circuits et la soudure des composants sur des cartes principales (cartes de commandes) dont le rôle principal est de gérer les moteurs pas à pas.

Essayé d'employer des matériaux que l'on peut se procurer facilement et des Composants disponible pour la fabrication de la machine CNC.
Pour piloter les moteurs pas à pas des axes X et Y et Z, nous avons utilisé la carte Arduino Mega2560.

- Faire un test avec la machine CNC traceur.

Apres avoir compléter le travail sur les circuits imprimés : des cartes électroniques les 6 H-BRIDGES et les 3 SRQUENCEURS et le test des 3 moteurs pas à pas.

II.1.La commande dir/step du SEQUENCEUR :

La commande dir/step dépend sur un circuit qui obtenir une commande direct soit Mode à pas entier, une phase alimentée à la fois (One Phase ON, Full Step) ou Mode deux phases alimentées en même temps (Two Phase ON, Full Step) .ce circuit commande par deux proche :
Step : qui est l'horloge pour la vitesse. A chaque front descendant le moteur fera un pas.

Dir : (ClockWise ou Counter ClockWise) qui décidera du sens de rotation du moteur

Pour réaliser cette commande numérique on a réalisé les cartes électroniques : les 3 SRQUENCEURS et a fin de puisse faire piloter les moteurs on a besoin des H-BRIGE la partie de puissance

II.2.Fabrication de CNC machine 3D matériel nécessaire pour réaliser la Machine CNC :

Nous essayons d'employer des matériaux que l'on peut se procurer facilement en Magasin de bricolage et des composants possibles à trouver pour la fabrication de la Machine CNC. Et pour piloter les moteurs pas à pas des axes X, Y et Z, nous avons deux Possibilités. Soi utiliser un couples « un SEQUENCEUR et 2 H-BRIDGES »pour chaque axe (que nous avons réalisé précédemment), ou on utilise trois Carte Driver A4988 pour Arduino.

- Machine CNC déjà construit
- 1x Arduino Mega2560
- 1x Plaque d'essais
- 3x drivers A4988 ou bien 3SEQUENCEURS et 6 H-BRIDGES
- 3x fins de course
- Des fils de connexion

III. Les logiciels utilisés :

Tous les programmes utilisés proviennent du monde "Open Source". Ils existent pour Windows, Linux et OS-X. Sur un PC, Nous allons dans un premier temps les paramétrer et les installer.

III.1. Simulation électrique sous Proteus Professionnel :

Proteus est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels incluent dans Proteus permettent la CAO dans le domaine Électronique.

III.1.1Présentation générale :

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses Entreprises et organismes de formation (incluant lycée et université) utilisent cette suite Logiciel. Outre la popularité de l'outil, Proteus possède d'autres avantages.

- Pack contenant des logiciels facile et rapide à comprendre et utiliser
- Le support technique est performant
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de La conception d'un projet.

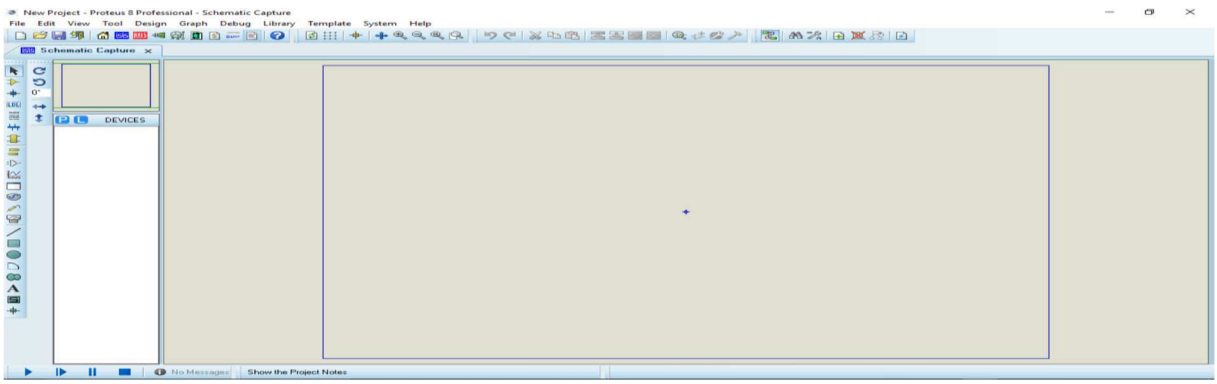


Figure 22 **:** Logiciel ISIS

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler Certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisé dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

III.1.2. Simulation et test du programme :

Pour saisir le schéma, il faut créer un nouveau projet puis placer les composants qui Doivent être sélectionné à partir de la bibliothèque des composants :

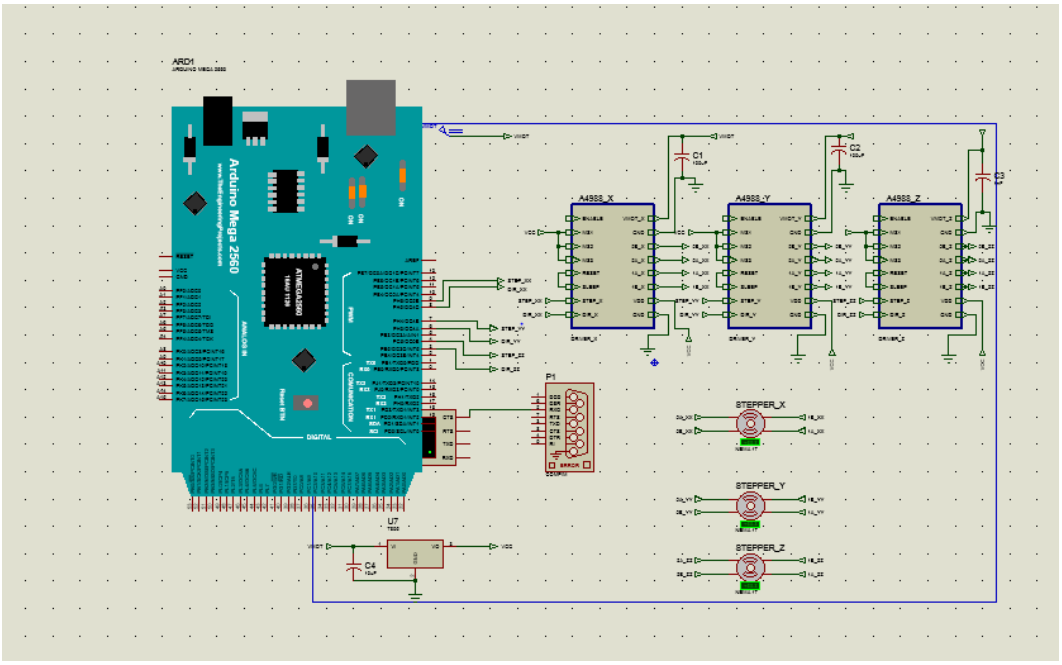


Figure 23 **:** Montage électronique dans Proteus

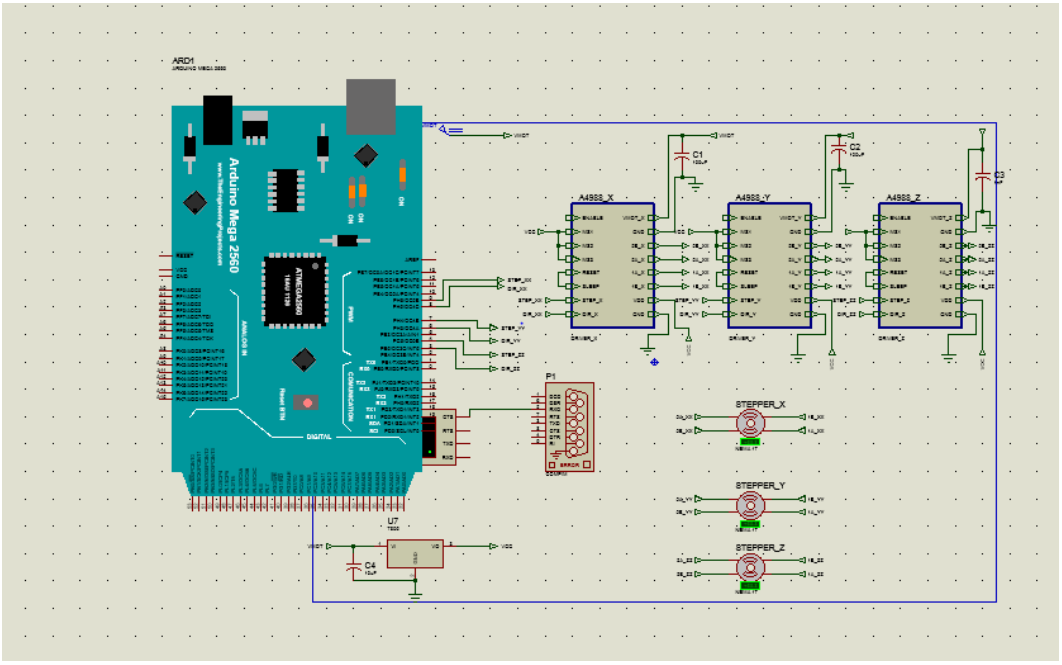


Figure 24 : simulation CNC machine

III.2.Logiciel de dessin : Inkscape :

Inkscape est un logiciel de dessin vectoriel que j'utilise pour ma part Windows7 64bits.

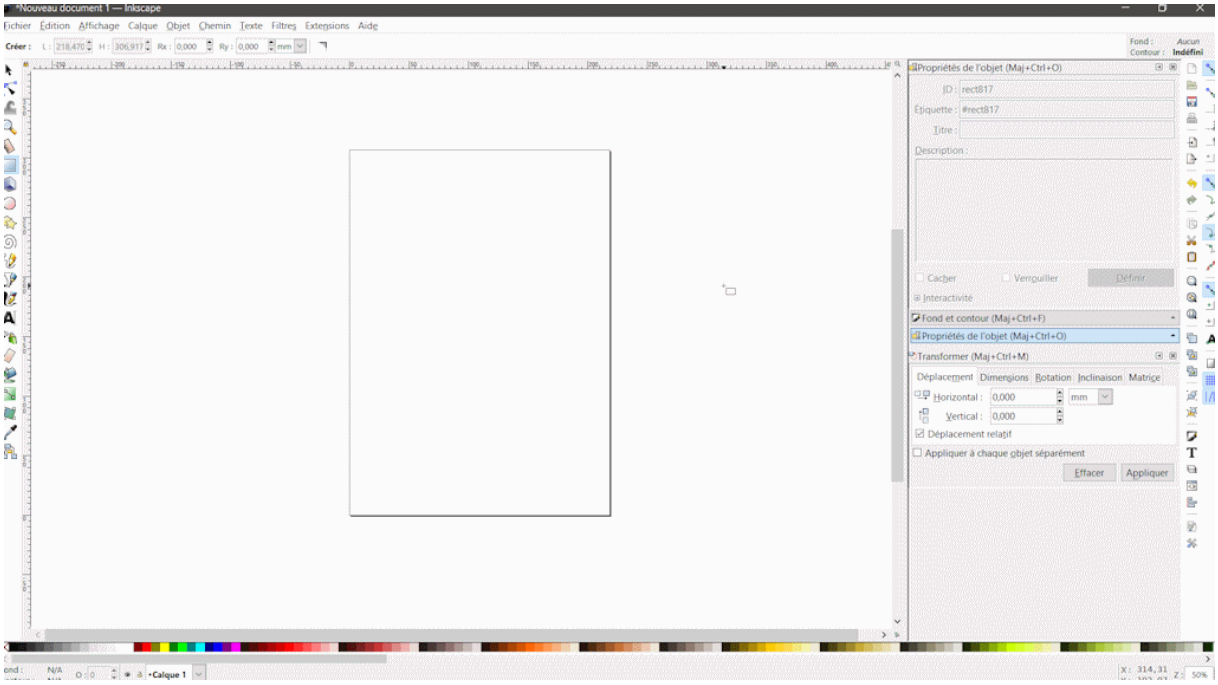


Figure 25 : Logiciel de dessin vectoriel Inkscape

III.3.UGS (Universal Gcode Sender) :

Universal Gcode Sender est une console permettant d'envoyer le Gcode sur la machine. C'est une application écrite en langage Java et elle ne nécessite pas d'installation spécifique.

Dans un premier temps télécharger la version spécifique, puis effectuer les actions suivantes :

- Créer un dossier "UGS" sur le disque dur et placer y les 4 fichiers inclus dans le ZIP.
- Connecter l'Arduino au PC via l'USB.
- Sous Windows, lancer le fichier "start-windows.bat".
- Dans UGS choisissez le port et sélectionner la vitesse de transmission, puis cliquez sur le bouton "Open".
- Ouvrez l'onglet "Machine Control", le panneau de commande est prêt !

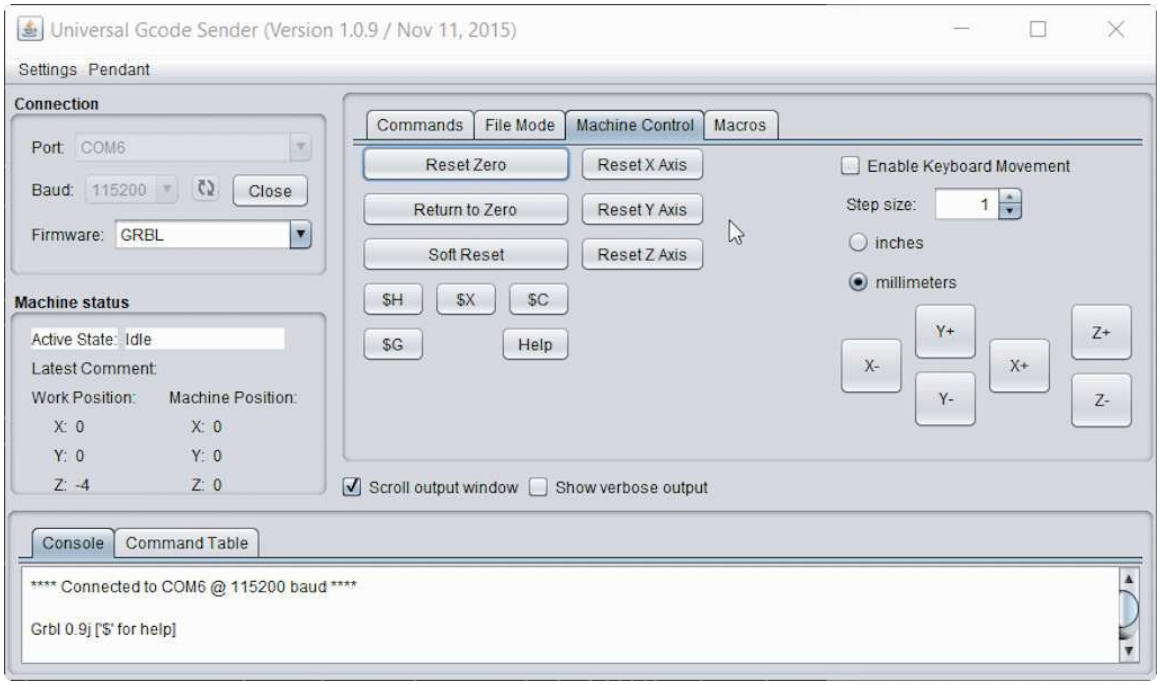


Figure 26 : Console CNC Universal Gcode Sender

III.4.Gestion machine :

III.4.1.GRBL :

GRBL est un firmware embarqué sur un Arduino. Il analyse le Gcode qu'on lui envoi et exécute des actions en envoyant des instructions haute fréquences aux moteurs pas-à-pas.

GRBL ne comporte pas d'interface utilisateur comme d'autres logiciels CNC. Nous allons donc exécuter UGS sur un PC qui interagira avec l'Arduino. Installer la version voulue de GRBL. Il est bien entendu écrit en langage Arduino qui est un langage proche du C et du C++.

Après avoir téléchargé GRBL, décompresser le dossier "grbl" contenu dans le fichier zip, dans le dossier "/libraries/" du programme Arduino.

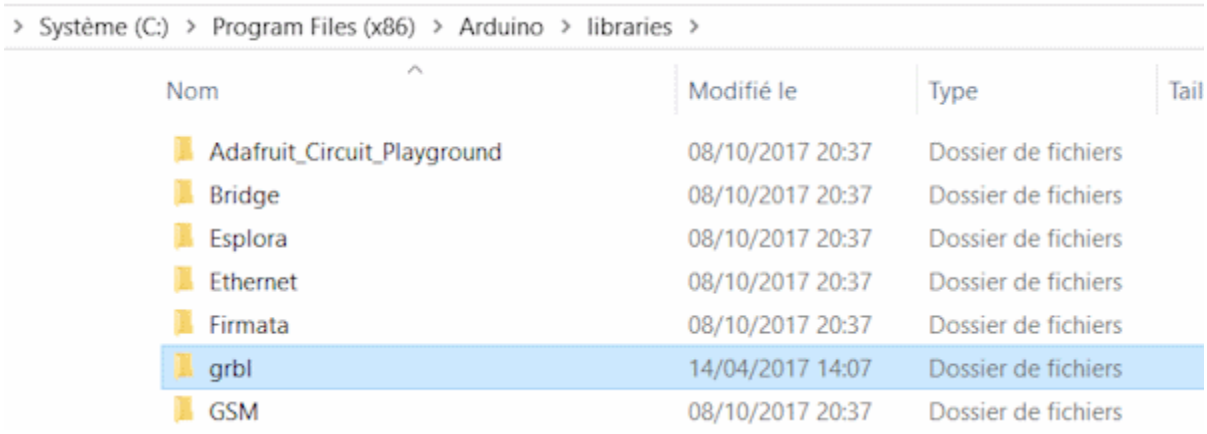


Figure 27 : Emplacement de GRBL sous Arduino

III.4.2.Compiler de GRBL :

Afin de puisse compiler le fichier GRBL il est recommander d'exécuter les étapes suivants :

- Connectez l'Arduino au PC via l'USB.
- Configurez le dans l'IDE Arduino (Type de carte et N° de port).
- Ouvrir "Fichiers/exemples/grbl/grblUpload".
- Cliquez sur la coche verte de vérification de code (En haut à gauche). Si tout se passe bien Il ne reste plus qu'à téléverser GRBL dans l'Arduino.

III.5.Paramétrage de GRBL :

Maintenant que GRBL est chargé dans l'Arduino avec la bonne configuration, il reste à faire le paramétrage, dont dans ce dernier toutes les dimensions sont en millimètres et si le cas comptez en pouces, la solution sera dans la conversion...

Le changement des valeurs des paramètres peut se faire des deux manières suivantes (Entre autres...).

Via UGS :

Si UGS est déjà installée on peut aussi utiliser son mode paramétrage. Connecter à l'Arduino, et cliquez sur "Paramètres/Paramètres Firmware/GRBL". Obtenez la fenêtre suivante :

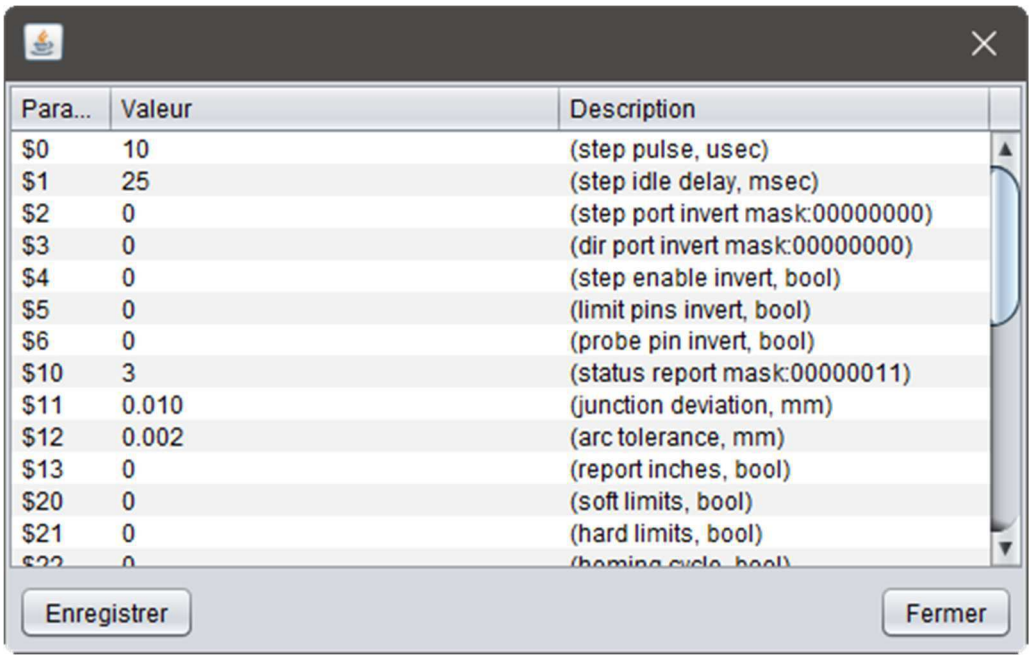


Figure 28 : paramètre Gcode Sender

Double cliquez sur la valeur à changer, modifier la et faite enregistrer. Une fois terminé, fermer cette fenêtre, GRBL est paramétré.

Via l'IDE Arduino :

La commande "$$" lancée depuis le terminal série donne ces données par défaut :

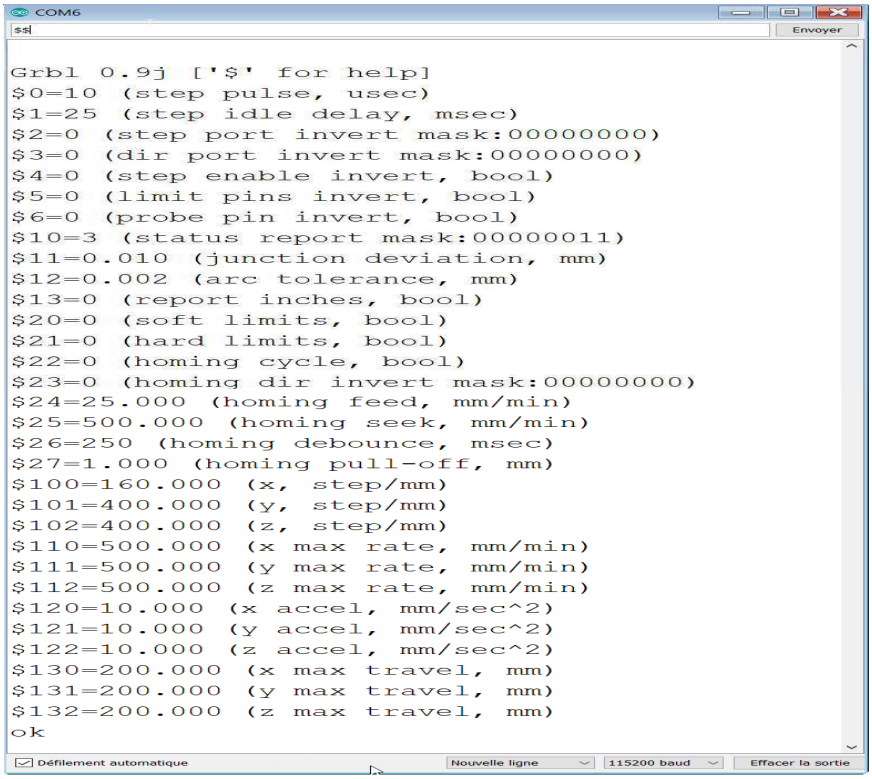


Figure 29 : Parametre depuis IDE

III. Les paramètres à modifier :

Dans le tableau ci-dessous, vous trouverez les raisons pour modifier ou non les lignes représentées ci-dessus, en fonction du matériel.

Paramètre	Définition
100	Nombre de pas moteur par mm sur l'axe X - Suivant le moteur X et son entraînement.
101	Nombre de pas moteur par mm sur l'axe Y - Suivant le moteur Y et son entraînement.
102	Nombre de pas moteur par mm sur l'axe Z - Suivant le moteur Z et son entraînement.
130	Distance maximum de travail sur l'axe X - Excursion X de la machine.
131	132Distance maximum de travail sur l'axe Y - Excursion Y de la machine.
110	Distance maximum de travail sur l'axe Z - Excursion Z de la machine.
111	Vitesse de déplacement maximum, pour le repositionnement, sur l'axe X.
112	Vitesse de déplacement maximum, pour le repositionnement, sur l'axe Y.

22	Vitesse de déplacement maximum pour le repositionnement, sur l'axe Z.
23	Validation du cycle de "Homing". Tout ou rien.
24	Vitesse lente de détection exacte des fins de courses pendant le "homing".
25	Vitesse de déplacement jusqu'à l'enclenchement des fins de courses, pendant le "Homing".
26	Temps de "rebond" des contacts électriques des fins de courses.
27	Distance de décalage de sécurité par rapport au déclenchement des fins de courses.

Tableau 1 : des paramètres à régler

V.1.Les composants utilisés pour la fabrication des cartes électroniques :

Chaque H-Bridge contient (on a réalisé 6 circuit du H-Bridge) :
4x IRF Z44N
4x DIODE CHOSTKY
2x PS (3 sorties)
3x CMD (3 sorties)
1x Jumper (2 pins)
3x Résistance 330Ω
2x Résistance 100Ω
2x Résistance 1K
2x Résistance 10K
2x 4N35
2x Led TLL G4400
Et chaque Séquenceur contient (on a réalisé 3 circuit du Séquenceurs)
2x PS (6 sorties)
1x 4030
1x 4013N
3x Résistance 330Ω
2x Résistance 10K
2x 4N35
2x Led TLL G4400
2x condensateur C 100 nf
1x condensateur C 100 mf
1x condensateur C 10 mf
1x 7805T IC3

V.2.Les photos des cartes électroniques :

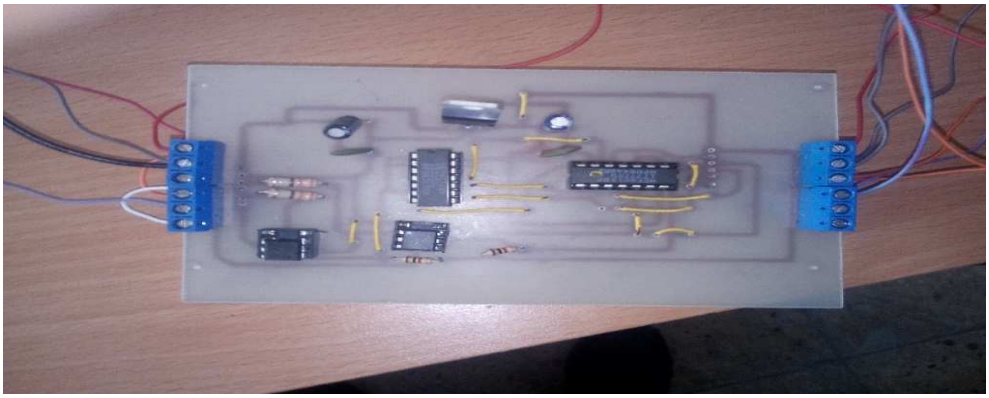


Figure 30 : photo du carte électronique «Séquenceur »

Figure 31 : photo du carte électronique «H-Bridge »

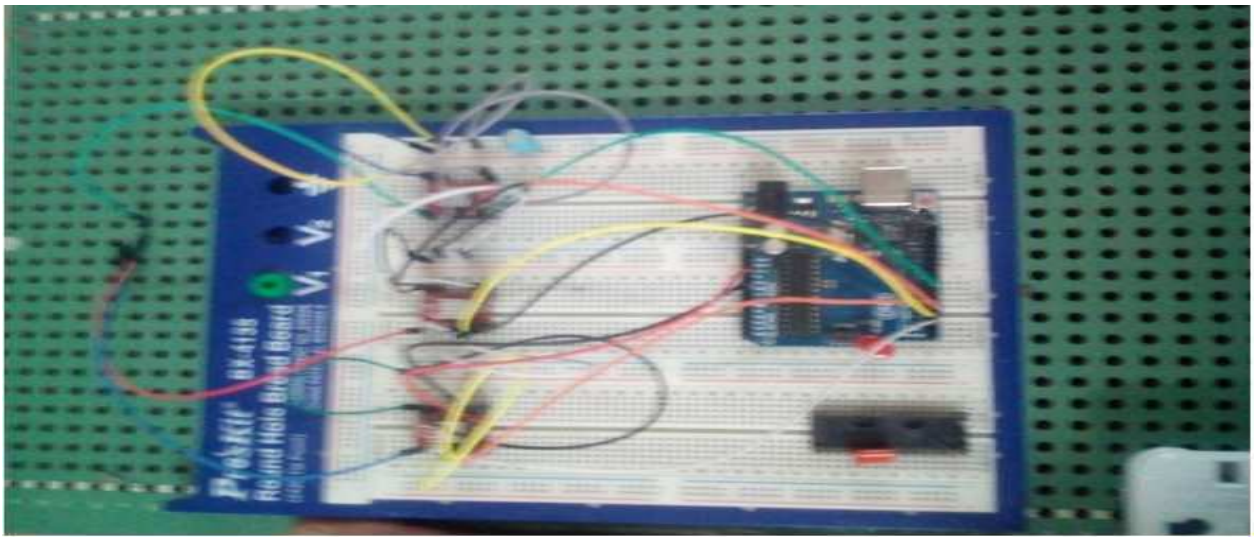


Figure 32: photo de l'Arduino avec les drivers

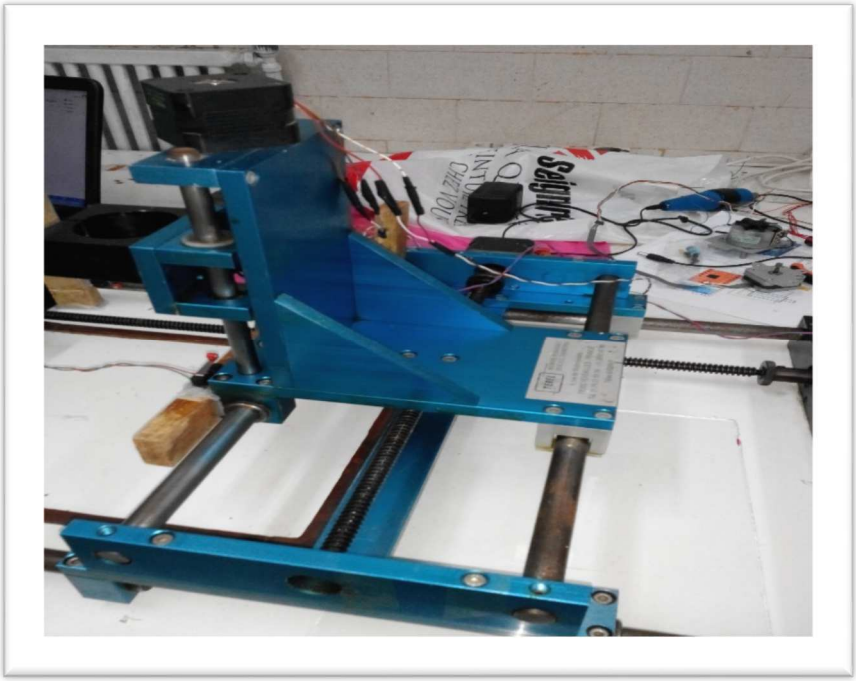


Figure 33: photo de la machine CNC

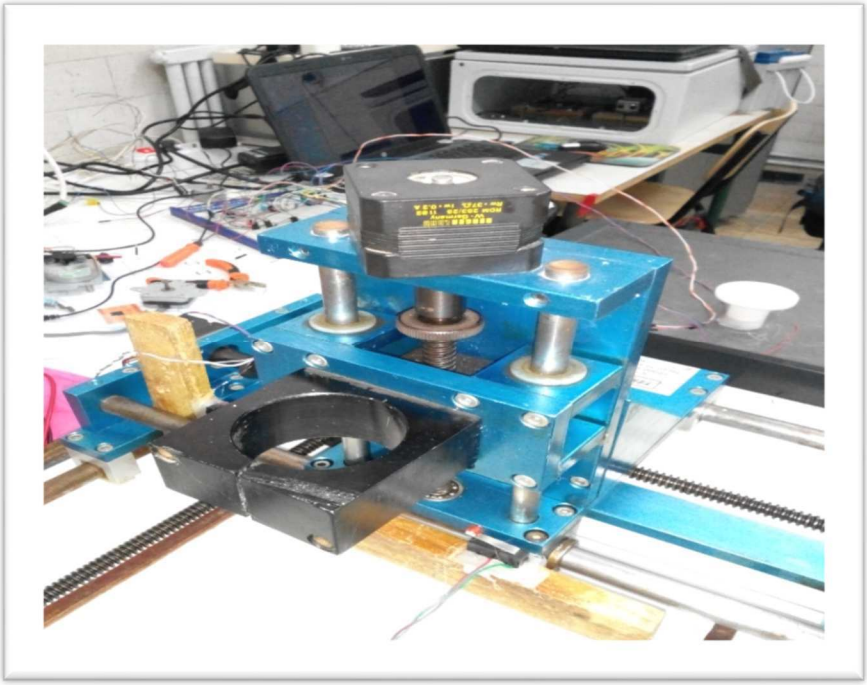


Figure 34: photo de la machine CNC/fin de course

Et enfin la partie de la commande Android via un Bluetooth qui comprend de la connexion du Bluetooth et le système de déplacement des moteurs des 3 axes en utilise l'application App Inventer2 qui est composées de différents blocs près programmé d'une façon de construire les éléments de l'application avec simple étapes.
Dont Le module Bluetooth HC-05 est un module Bluetooth SPP (Serial Port Protocol) facile à utiliser, conçu pour une configuration de connexion série sans fil transparente. Sa communication s'effectue via une communication série qui facilite l'interface avec le contrôleur ou le PC. Le module Bluetooth HC-05 fournit le mode de commutation entre le mode maître et le mode esclave, ce qui signifie qu'il ne peut ni recevoir ni transmettre de données.

Caractérisation :

1. Modèle : HC-05
2. Tension d'entrée : DC 5V
3. Méthode de communication : communication série
4. Le mode maître et esclave peut être commuté

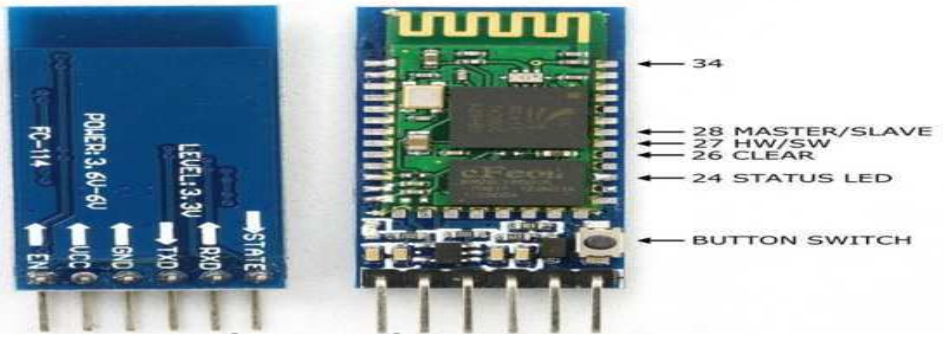


Figure 35 : Image du bluetooth HC-05

Conclusion :

Comme nous venons de le voir dans ce chapitre, beaucoup des travaux ont été réalisés. Concernant la machine CNC, j'ai présenté plusieurs types de logiciels utilisés pour la Simulation et la commande de machine CNC. Pour réaliser cette commande je dispose l'utilisation d'une Carte microcontrôleur "Arduino Mega2560" qui possède un espace de programmation qu'est très claire et simple et les circuits intègres les Driver. Ce dernier sert à contrôler les moteurs pas à pas bipolaire.
A l'aide du logiciel " Inkscape " j'ai stoker les codes G dans un fichier pour le copier en suite sur la carte Arduino Mega. Pour finir, je branche les moteurs pas à pas, et je teste la carte de commande des moteurs pas à pas. Les axes de la machine CNC fonctionne correctement et Répond aux différentes commandes et la dernière étape c'est l'application Android via le Bluetooth h-05.

Conclusion générale :

L'élaboration de ce travail dans le cadre du projet de la fin d'étude, m'a permis d'approfondir mes connaissances théoriques en électronique et d'acquérir une bonne expérience au niveau de la réalisation pratique.
Lors de cette manipulation, j'ai essayé de fournir l'automatisation de commande de trois moteurs pas à pas et applique à la machine CNC. Mon projet de fin d'étude consiste à détailler ainsi qu'à réaliser des cartes électroniques de commande de trois moteurs pas à pas.
Ce projet m'a donné une meilleure idée sur la complémentarité entre le volet théorique et le volet pratique.
J'ai validé la faisabilité et le bon fonctionnement des circuits déjà développées théoriquement.
Implémentation des circuits simulés sur la plaque à essais (plaque à Trous) afin de s'assurer de son fonctionnement.
Une fois la simulation est bien vérifiée, j'ai passé à l'étape de routage en utilisant l'outil AIGLE pour la réalisation des cartes électroniques en imprimant les circuits et la soudure des composants sur des cartes principales (cartes de commandes) dont le rôle principal est de gérer les moteurs pas à pas.
Finalement, mon travail m'a permis la réalisation pratique, et le test des différentes cartes assemblées, la commande d'un prototype d'une machine de dessin 2D. L'avantage de mon commande est qu'elle est à la norme de la commande numérique des machines à 3 axes.
J'ai rencontré plusieurs problèmes durant la conception de la commande et la réalisation pratique (périodes long pour l'assemblage des différents composants, protocole de communication, couple des moteurs et leurs vitesses…),

En fait, ce projet a été une source de découverte de plusieurs domaines d'étude telle que l'informatique pour la programmation embarquée et le design des schémas électroniques,
Sans oublier le savoir-faire dans le domaine électronique qui consiste à réaliser
Pratiquement les circuits électroniques.
En perspective, plusieurs points sont à approfondir et ouvrent la voie à de nouveaux axes de travaux :
-Contrôle des autres axes (imprimante 3D).
-Application de la commande avec modèle intelligente

BIBLIOGRAPHIE

BIBLIOGRAPHIE :

Arduino

1 Https://numerique.dunod.com/88890/Arduino.ebook
2 http://arduino.cc/en/Main/AruinoBoardMega2560
3 www.reality.be/elo/labos2/files/AtMega32DocFr.pdf

Android

6 androidpoly.pdf
7 google android open accessory development kit
8 Android Development Tools (ADT)
9 https://developer.android.com/guide/

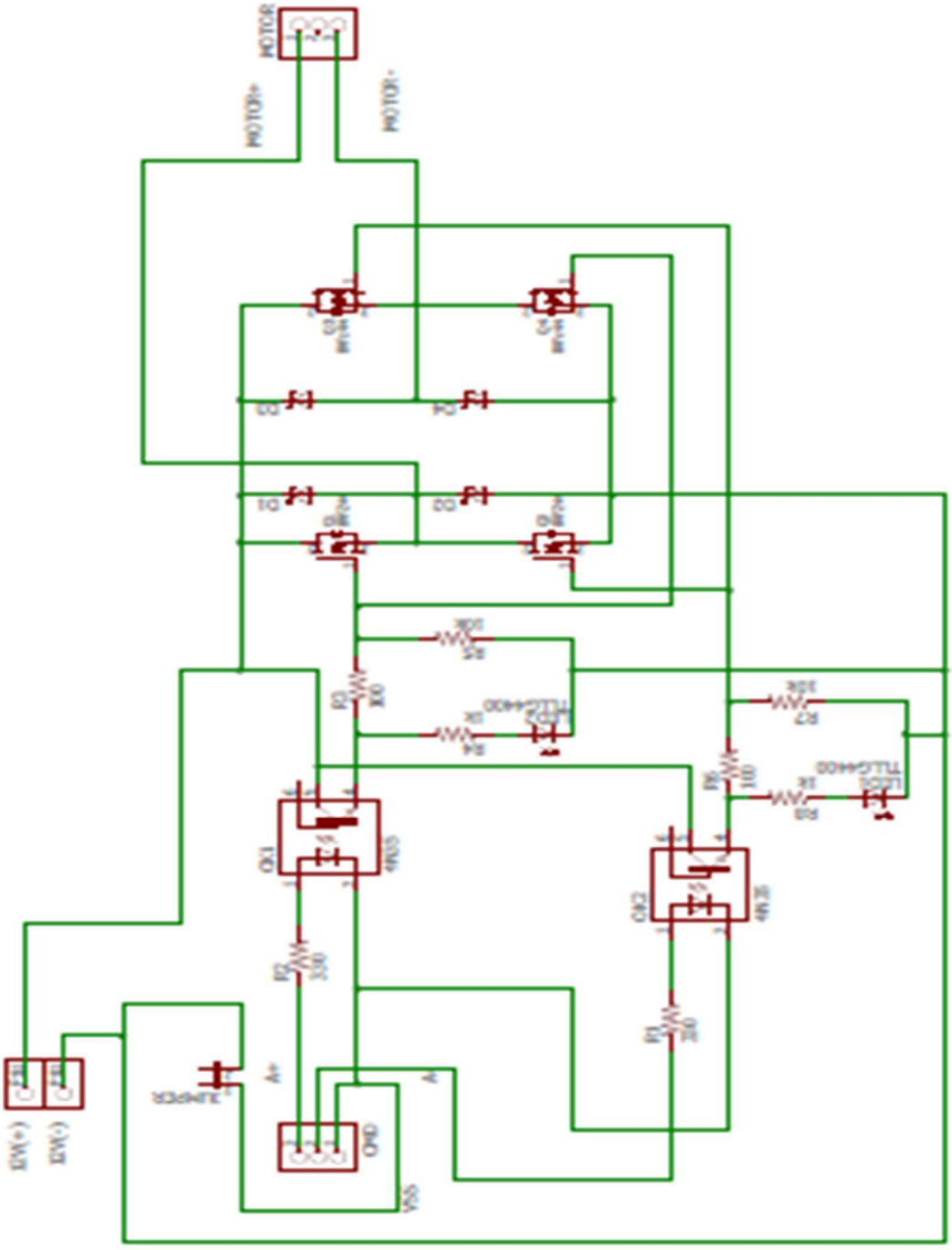
CNC

10-youtube//piloter-une-petite-CNC,
11 http://realisationcnc.canalblog.com/
12 http://demonter.net/comment-fabriquer-une-cnc-diy-3-axes-wip/
13 https://wiki.labomedia.org/index.php/Reglage_CNC
14 http://www.schmalzhaus.com/EasyDriver/
15 http://inkscape.fr/
16 https://github.com/grbl/grbl
17-https://github.gcode sender

Annexe

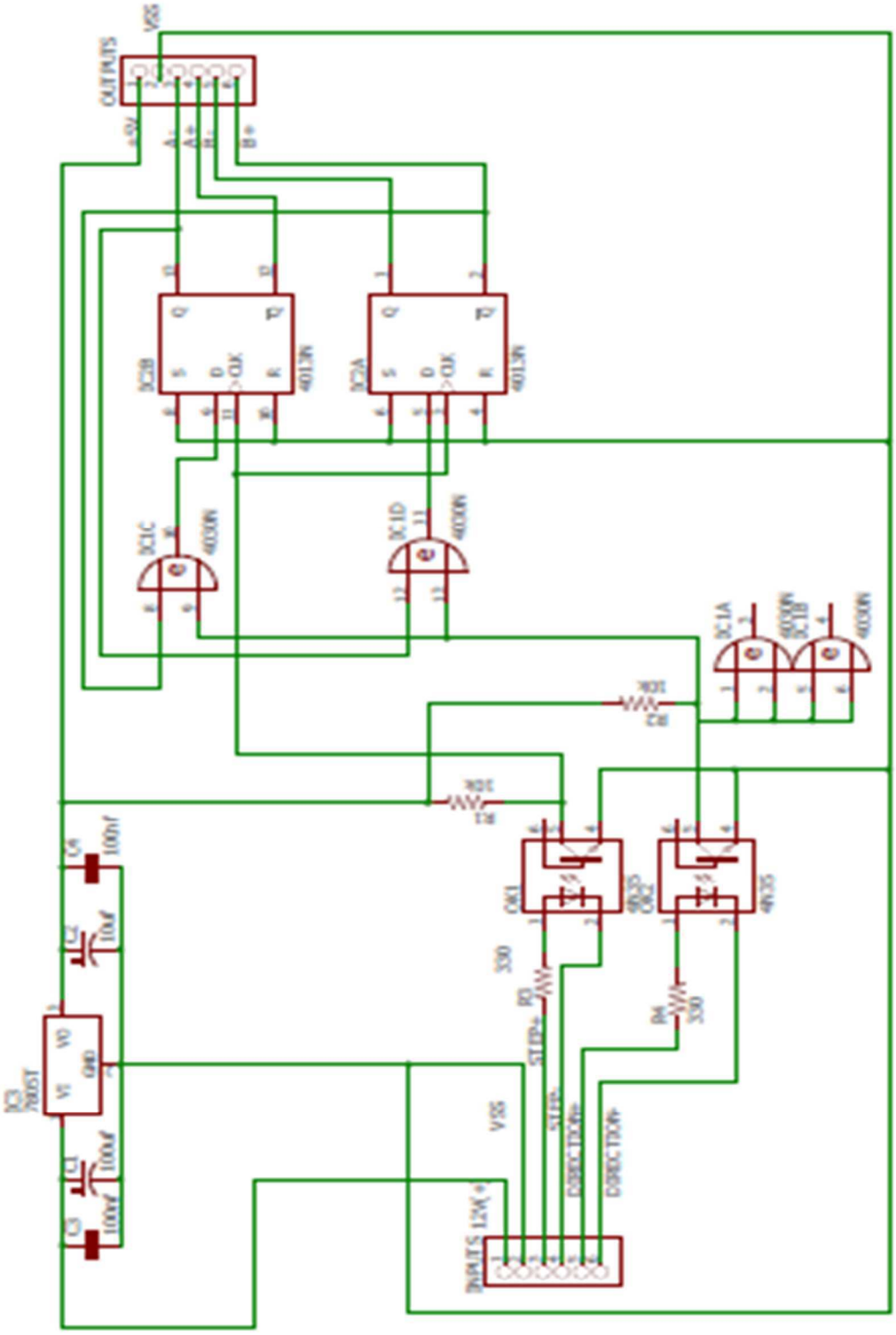
Les circuits électriques :

H-BRIDGE :



SEQUENSEUR :

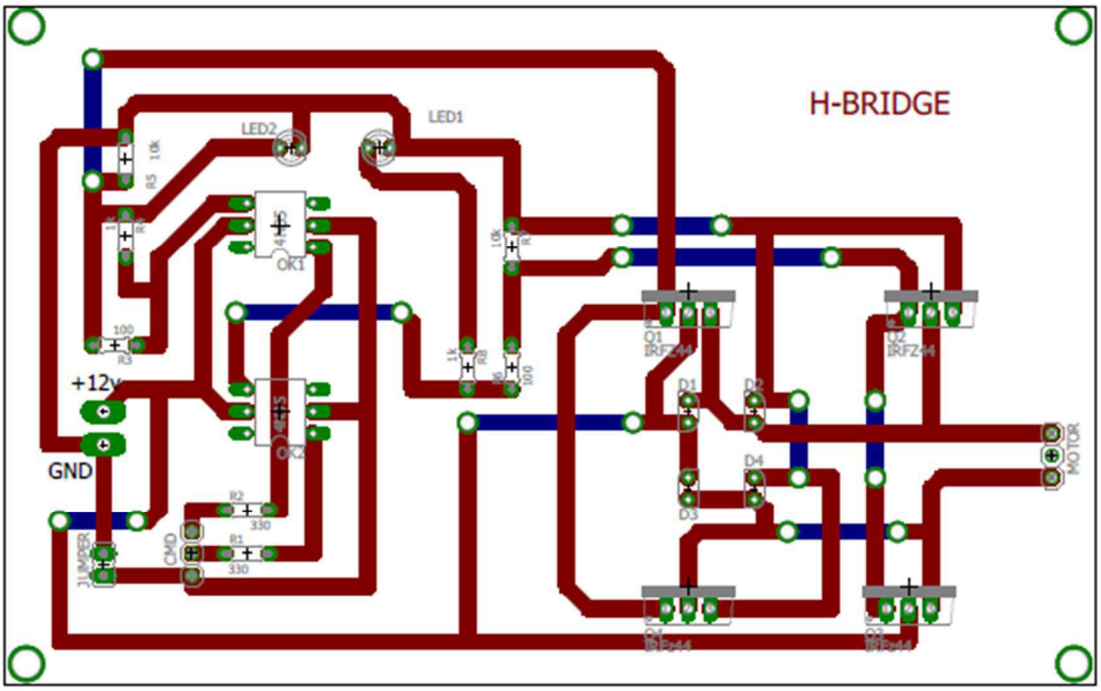
Annexe



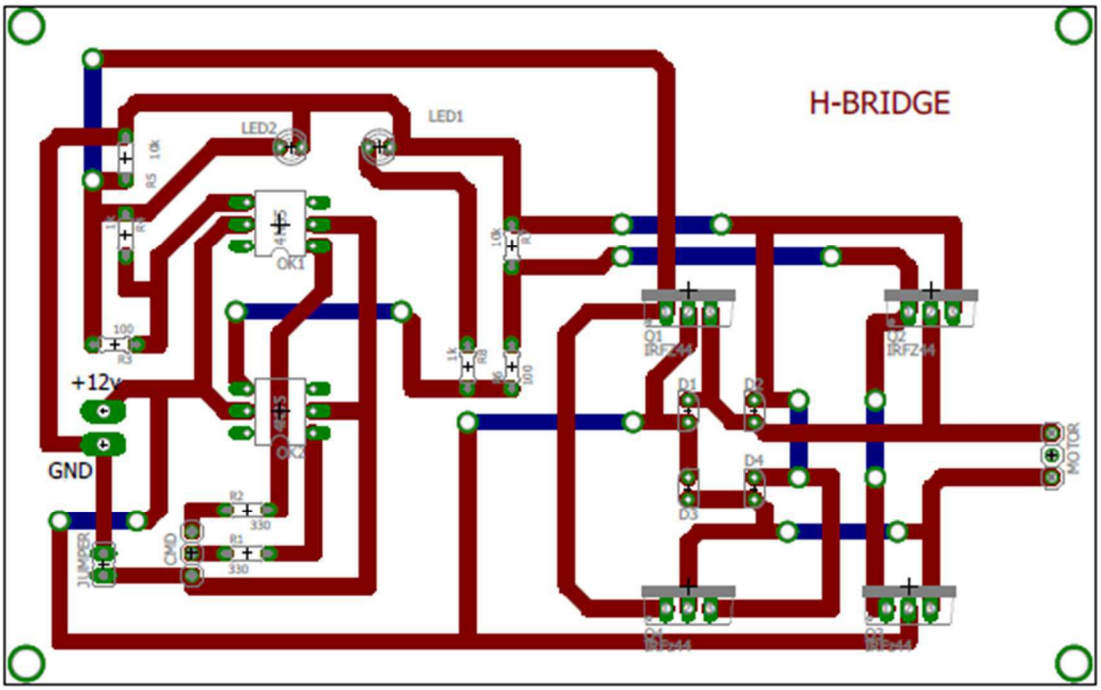
Annexe

Les circuits imprimés :

H-BRIDGE :



SEQUENSEUR :



Les data sheet des composants électriques :

SL4030B

Quad Exclusive-OR Gate

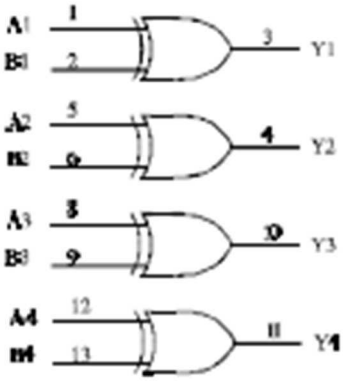
High-Voltage Silicon-Gate CMOS

The SL4030B types consist of four independent Exclusive-OR gates. The SL4030B provides the system designer with a means for direct implementation of the Exclusive-OR function.

- Operating Voltage Range: 3.0 to 18 V
- Maximum input current of 1 µA at 18 V over full package-temperature range; 100 nA at 18 V and 25°C
- Noise margin (over full package temperature range):
  1.0 V min @ 5.0 V supply
  2.0 V min @ 10.0 V supply
  2.5 V min @ 15.0 V supply

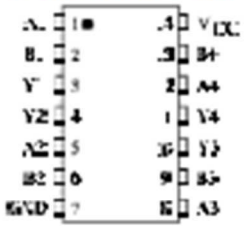
ORDERING INFORMATION
SL4030BN Plastic
SL4030BD SOIC
T_A = -55° to 125° C for all packages

LOGIC DIAGRAM



PIN 14 = V_{CC}
PIN 7 = GND

PIN ASSIGNMENT



FUNCTION TABLE

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

System Logic Semiconductor

MAXIMUM RATINGS*

Symbol	Parameter	Value	Unit
V_{CC}	DC Supply Voltage (Referenced to GND)	-0.5 to +20	V
V_{IN}	DC Input Voltage (Referenced to GND)	-0.5 to V_{CC} +0.5	V
V_{OUT}	DC Output Voltage (Referenced to GND)	-0.5 to V_{CC} +0.5	V
I_{IN}	DC Input Current, per Pin	±10	mA
P_D	Power Dissipation in Still Air, Plastic DIP+ SOIC Package+	750 500	mW
P_D	Power Dissipation per Output Transistor	100	mW
Tstg	Storage Temperature	-65 to +150	°C
T_L	Lead Temperature, 1 mm from Case for 10 Seconds (Plastic DIP or SOIC Package)	260	°C

*Maximum Ratings are those values beyond which damage to the device may occur.
Functional operation should be restricted to the Recommended Operating Conditions.
+Derating - Plastic DIP: - 10 mW/°C from 65° to 125°C
SOIC Package: : - 7 mW/°C from 65° to 125°C

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
V_{CC}	DC Supply Voltage (Referenced to GND)	3.0	18	V
V_{IN}, V_{OUT}	DC Input Voltage, Output Voltage (Referenced to GND)	0	V_{CC}	V
T_A	Operating Temperature, All Package Types	-55	+125	°C

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, V_{IN} and V_{OUT} should be constrained to the range GND≤(V_{IN} or V_{OUT})≤V_{CC}.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either GND or V_{CC}). Unused outputs must be left open.

SL4030B

DC ELECTRICAL CHARACTERISTICS (Voltages Referenced to GND)

Symbol	Parameter	Test Conditions	V_{CC} V	Guaranteed Limit ≥-55°C	25°C	≤125 °C	Unit
V_{IH}	Minimum High-Level Input Voltage	V_{OUT}=0.5V or V_{CC} - 0.5V V_{OUT}=1.0V or V_{CC} - 1.0V V_{OUT}=1.5V or V_{CC} - 1.5V	5.0 10 15	3.5 7 11	3.5 7 11	3.5 7 11	V
V_{IL}	Maximum Low -Level Input Voltage	V_{OUT}=0.5V or V_{CC} - 0.5V V_{OUT}=1.0V or V_{CC} - 1.0V V_{OUT}=1.5V or V_{CC} - 1.5V	5.0 10 15	1.5 3 4	1.5 3 4	1.5 3 4	V
V_{OH}	Minimum High-Level Output Voltage	V_{IN}=GND or V_{CC}	5.0 10 15	4.95 9.95 14.95	4.95 9.95 14.95	4.95 9.95 14.95	V
V_{OL}	Maximum Low-Level Output Voltage	V_{IN}=GND or V_{CC}	5.0 10 15	0.05 0.05 0.05	0.05 0.05 0.05	0.05 0.05 0.05	V
I_{IN}	Maximum Input Leakage Current	V_{IN}= GND or V_{CC}	18	±0.1	±0.1	±1.0	μA
I_{CC}	Maximum Quiescent Supply Current (per Package)	V_{IN}= GND or V_{CC}	5.0 10 15 20	0.25 0.5 1.0 5.0	0.25 0.5 1.0 5.0	7.5 15 30 150	μA
I_{OL}	Minimum Output Low (Sink) Current	V_{IN}= GND or V_{CC} U_{OL}=0.4 V U_{OL}=0.5 V U_{OL}=1.5 V	 5.0 10 15	 0.64 1.6 4.2	 0.51 1.3 3.4	 0.36 0.9 2.4	mA
I_{OH}	Minimum Output High (Source) Current	V_{IN}= GND or V_{CC} U_{OH}=2.5 V U_{OH}=4.6 V U_{OH}=9.5 V U_{OH}=13.5 V	 5.0 5.0 10 15	 -2.0 -0.64 -1.6 -4.2	 -1.6 -0.51 -1.3 -3.4	 -1.15 -0.36 -0.9 -2.4	mA

AC ELECTRICAL CHARACTERISTICS (C_L=50pF, R_L=200kΩ, Input t_r=t_f=20 ns)

Symbol	Parameter	V_{CC} V	Guaranteed Limit ≥-55°C	25°C	≤125°C	Unit
t_{PLH}, t_{PHL}	Maximum Propagation Delay, Input A or B to Output Y (Figure 1)	5.0 10 15	280 130 100	280 130 100	560 260 200	ns
t_{TLH}, t_{THL}	Maximum Output Transition Time, Any Output (Figure 1)	5.0 10 15	200 100 80	200 100 80	400 200 160	ns
C_{in}	Maximum Input Capacitance	-		7.5		pF

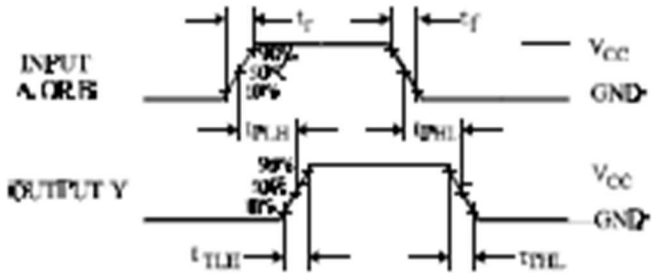
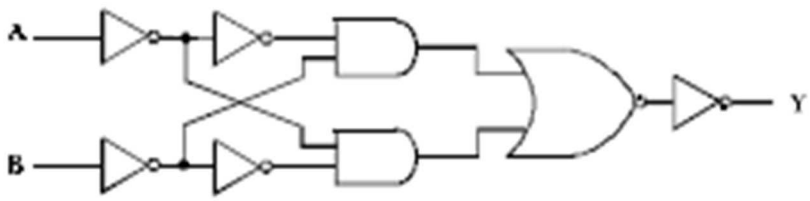


Figure 1. Switching Waveforms

EXPANDED LOGIC DIAGRAM
(1/4 of the Device)



New Jersey Semi-Conductor Products, Inc.

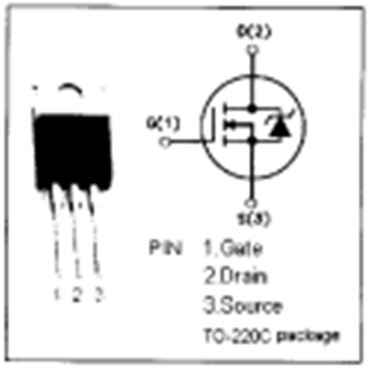
20 STERN AVE.
SPRINGFIELD, NEW JERSEY 07081
U.S.A.

TELEPHONE: (973) 376-2922
(212) 227-6005
FAX: (973) 376-8960

N-Channel MOSFET Transistor — IRFZ44N

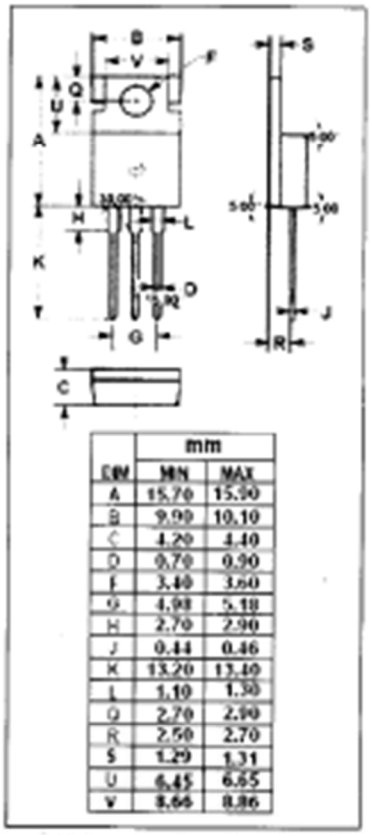
FEATURES

- Drain Current –I_D=49A@ T_C=25℃
- Drain Source Voltage-
  : V_{DSS}= 55V(Min)
- Static Drain-Source On-Resistance
  : $R_{DS(on)}$ = 0.032Ω (Max)
- Fast Switching



DESCRIPTION

- Designed for low voltage, high speed switching applications in power supplies, converters and power motor controls, these devices are particularly well suited for bridge circuits where diode speed and commutating safe operating areas are critical and offer additional safety margin against unexpected voltage transients.



DIM	mm MIN	MAX
A	15.70	15.90
B	9.90	10.10
C	4.20	4.40
D	0.70	0.90
F	3.40	3.60
G	4.98	5.18
H	2.70	2.90
J	0.44	0.46
K	13.20	13.40
L	1.10	1.30
Q	2.70	2.90
R	2.50	2.70
S	1.29	1.31
U	6.45	6.65
V	8.66	8.86

ABSOLUTE MAXIMUM RATINGS(T_a=25℃)

SYMBOL	PARAMETER	VALUE	UNIT
V_{DSS}	Drain-Source Voltage	55	V
V_{GS}	Gate-Source Voltage-Continuous	±20	V
I_D	Drain Current-Continuous	49	A
I_{DM}	Drain Current-Single Pluse (t_p≤10μs)	160	A
P_D	Total Dissipation @T_C=25℃	94	W
T_J	Max. Operating Junction Temperature	175	℃
T_{stg}	Storage Temperature	-55~175	℃

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	MAX	UNIT
$R_{th\,j\text{-}c}$	Thermal Resistance, Junction to Case	1.5	℃/W
$R_{th\,j\text{-}a}$	Thermal Resistance, Junction to Ambient	62	℃/W

N-Channel MOSFET Transistor — IRFZ44N

ELECTRICAL CHARACTERISTICS

T_C=25℃ unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
$V_{(BR)DSS}$	Drain-Source Breakdown Voltage	V_{GS}= 0; I_D= 0.25mA	55		V
$V_{GS(th)}$	Gate Threshold Voltage	V_{DS}= V_{DS}; I_D= 0.25mA	2	4	V
$R_{DS(on)}$	Drain-Source On-Resistance	V_{GS}= 10V; I_D= 25A		0.032	Ω
I_{GSS}	Gate-Body Leakage Current	V_{GS}= ±20V;V_{DS}= 0		±100	nA
I_{DSS}	Zero Gate Voltage Drain Current	V_{DS}= 55V; V_{GS}= 0 V_{DS}= 55V; V_{GS}= 0; T_J= 150℃		25 250	μA
V_{SD}	Forward On-Voltage	I_S= 25A; V_{GS}= 0		1.3	V

1N4001 THRU 1N4007

PLASTIC SILICON RECTIFIER

VOLTAGE - 50 to 1000 Volts CURRENT - 1.0 Ampere

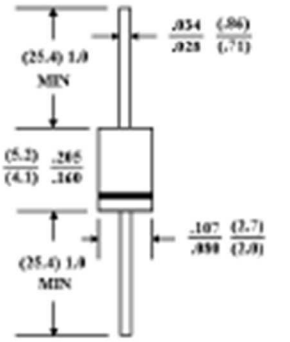
FEATURES

- Low forward voltage drop
- High current capability
- High reliability
- High surge current capability
- Exceeds environmental standards of MIL-S-19500/228

MECHANICAL DATA

Case: Molded plastic , DO-41
Epoxy: UL 94V-O rate flame retardant
Lead: Axial leads, solderable per MIL-STD-202, method 208 guaranteed
Polarity: Color band denotes cathode end
Mounting Position: Any
Weight: 0.012 ounce, 0.3 gram

DO-41



Dimensions in inches and (millimeters)

MAXIMUM RATINGS AND ELECTRICAL CHARACTERISTICS

Ratings at 25 •• ambient temperature unless otherwise specified.
Single phase, half wave, 60 Hz, resistive or inductive load.
For capacitive load, derate current by 20%.

	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	UNITS
Maximum Recurrent Peak Reverse Voltage	50	100	200	400	600	800	1000	V
Maximum RMS Voltage	35	75	140	280	420	560	700	V
Maximum DC Blocking Voltage	50	100	200	400	600	800	1000	V
Maximum Average Forward Rectified Current .375"(9.5mm) Lead Length at T_A=75 ••				1.0				A
Peak Forward Surge Current 8.3ms single half sine-wave superimposed on rated load (JEDEC method)				30				A
Maximum Forward Voltage at 1.0A DC and 25 ••				1.1				V
Maximum Full Load Reverse Current Full Cycle Average at 75 •• Ambient				30				•• A
Maximum Reverse Current at T_A=25 ••				5.0				•• A
At Rated DC Blocking Voltage T_A=100 ••				500				•• A
Typical Junction capacitance (Note 1)				15				pF
Typical Thermal Resistance (Note 2) R•• JA				50				••/W
Typical Thermal resistance (NOTE 2) R•• JL				25				••/W
Operating and Storage Temperature Range T_A T_{STG}				-55 to +150				••

NOTES:

1. Measured at 1 MHz and applied reverse voltage of 4.0 VDC.
2. Thermal Resistance Junction to Ambient and from junction to lead at 0.375"(9.5mm) lead length P.C.B mounted.

PANJIT

RATING AND CHARACTERISTIC CURVES

1N4001 THRU 1N4007

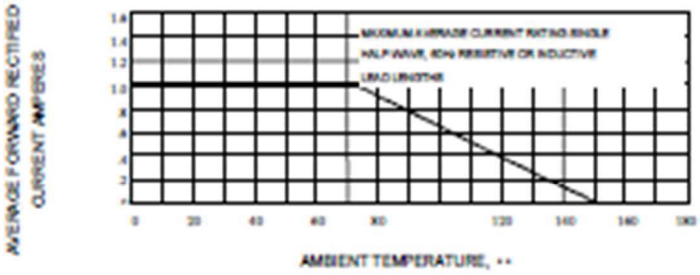


Fig. 1-TYPICAL FORWARD CURRENT DERATING CURVE

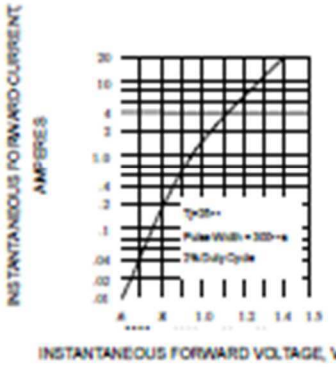


Fig. 2-TYPICAL FORWARD CHARACTERISTICS

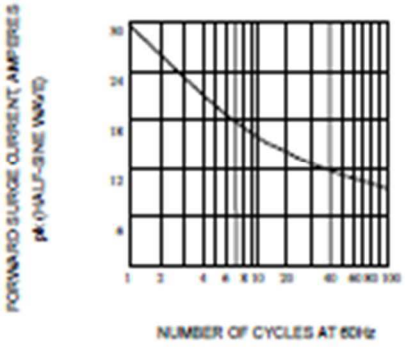


Fig. 3-MAXIMUM NON-REPETITIVE FORWARD SURGE CURRENT

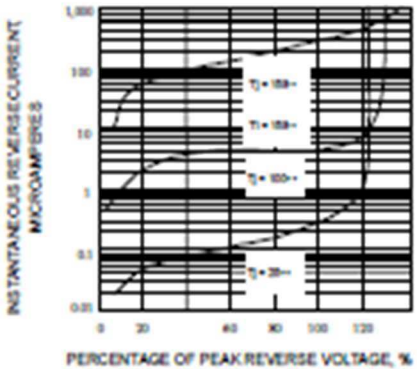


Fig. 4-TYPICAL REVERSE CHARACTERISTICS

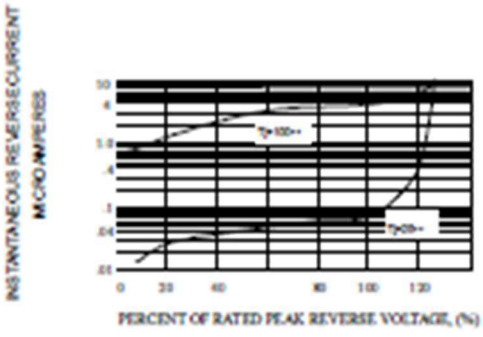


Fig. 5-TYPICAL REVERSE CHARACTERISTICS