

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

Université Mouloud Mammeri de Tizi-Ouzou



Faculté du génie électrique et d'Informatique

Mémoire de Fin d'Etudes
De MASTER PROFESSIONNELLE
Domaine : Sciences et Technologies
Filière: Génie Electrique
Spécialité : Electronique Industrielle

Présenté par

Mokrane YAHMI

Rachid SMAI

Thème

**Conception et Réalisation d'un simulateur du
cycle solaire à de Base Arduino**

Mémoire soutenu publiquement le 21/07/2016 devant le jury composé de :

Mr. LAZRI M., MCA, UMMTO,

President

Mr. OUALOUCHE F., MCB, UMMTO,

Encadreur

Mr. SEHAD M., MCB, UMMTO,

Examineur

Mr. HAMEG S. MAA, UMMTO,

Examineur

Remerciements

Je remercie le Bon DIEU de m'avoir donné la force et le courage d'accomplir ce travail.

Je remercie mon promoteur Mr. OUALOUCHE pour son aide, son orientation et ses conseils durant l'accomplissement du projet.

On en profite aussi pour remercier tous nos camarades de promotion pour leur support moral, qui nous ont aidés à dépasser les obstacles et à continuer vers l'avenir, et ce malgré des périodes de doute et de fatigue.

Je remercie également les membres du jury d'avoir accepté de juger ce modeste travail.

DEDICACES

Je dédie ce travail :

A ma famille.

A mes amis.

A tous ceux que me sont cher(e)s.

Mokrane

Ce travail modeste est dédié :

A ma chère mère et mon père.

A ma sœur et mes frères.

Et a tous mes chers amis.

A mes camarades de l'université Mouloud Mammeri de Tizi-Ouzou.

Smail

Sommaire

Introduction générale	7
Chapitre I : le cycle solaire	
1. Préambule :	8
<u>2.</u> Le Soleil :.....	8
2.1 Les Caractéristiques générales :.....	8
2.2 Le cycle solaire :.....	9
2.2.1. Définition le cycle solaire :	9
2.4.2. Les différents phases du cycle solaire chaque journée :.....	9
<u>3.</u> Présentation du projet :	11
3.1. Réalisation un simulateur du cycle solaire a la base d'Arduino :.....	11
<u>4.</u> Discussion :.....	12
Chapitre II : présentation de la carte Arduino	
1 préambule :	13
2 La carte d'acquisition :	13
2.1 Définition de la carte Arduino :.....	13
2.2 Historique.....	13
2.3 Les différentes cartes Arduino :.....	14
2.4 Caractéristique de la carte Arduino :	15
3 Description de la carte Arduino Mega :.....	16
3.1 Architecture de la carte arduino Mega 2560 :.....	17
3.2 Le microcontrôleur :.....	17
3.2.1 Caractéristiques du microcontrôleur ATMEGA 2560 :.....	17
3.2.2 Présentation physique du microcontrôleur ATMEGA2560 :.....	18
3.3 Le entrées-sorties de la carte :.....	23
3.4 Mémoire :	23
3.5 L'alimentation :	23
3.6 Visualisation :	25
3.7 La connectique :	25
3.8 Carte d'extension « SHIELD » :	25
3.8.1 Exemple d'usage de SHIELD pour connecter des moteurs à courant continu :	26
4 Architecture SOFTWARE de la carte :	26

4.1	Logiciel :	26
4.1.1	L'interface logicielle :	27
4.1.2	Structure de logiciel :	27
4.1.3	Structure de programme :	28
4.1.4	Programmation :	29
4.2	Développement d'un projet :	29
5.	Discussion :	31

Chapitre III : description de tout les composant

1.	Préambule	32
2.	Présentation des composants :	32
2.1	L'optotriac :	32
2.1.2	Fonctionnement :	32
2.1.3	Caractéristiques électriques :	33
2.2	Le triac :	34
2.2.1	Description :	34
2.2.2	Principe de Fonctionnement :	34
2.2.3	Commande d'un triac :	35
2.3	Transistors NPN BC547 :	36
2.3.1	Description :	36
2.3.2	Caractéristique du transistor NPN BC547 :	37
2.4	Les relais :	37
2.4.1	Principe :	37
2.4.2	Fonctionnement d'un relais :	38
2.4.3	Utilisation :	38
2.5	Potentiomètre.....	38
2.5.1	Présentation.....	38
2.5.2	<u>utélisation des potentiomètre.....</u>	38
2.5.3	Type de potentiomètres :	39
2.5.3.1	Potentiomètre linéaire :	39
2.5.3.2	potentiomètre logarithmique.....	40
2.5.3.3	potentiomètre à prise médiane.....	40
2.5.3.4	potentiomètre numérique	41
2.5.4	variation d'un potentiomètre.....	41

<u>3. Discussion</u>	<u>42</u>
Chapitre IV : réalisation pratique	
<u>1. Préambule</u>	<u>43</u>
<u>2. Les différentes étapes de la réalisation</u> :.....	<u>43</u>
<u>3. Schéma synoptique générale</u> :.....	<u>43</u>
<u>3.1 Bloc d'alimentation</u> :	<u>44</u>
<u>3.1.1 Source externe AC 220V</u> :	<u>44</u>
<u>3.1.2 Tension de +5V</u> :	<u>45</u>
<u>3.2 Bloc de commande</u> :.....	<u>45</u>
<u>3.2.1 Commande manuelle</u> :.....	<u>45</u>
<u>3.3 Bloc de puissance</u> :	<u>47</u>
<u>3.3.1 Bloc de puissance à base d'optotriac</u> :.....	<u>47</u>
3.3.1.1 Description.....	47
3.3.1.2 Branchement de bloc de puissance a base d'optotriac avec l'Arduino	48
<u>3.3.2 Zero grossing</u> :	<u>48</u>
3.3.2.1 Description.....	48
3.3.2.2 Branchement du zero crossing avec la carte Arduino	48
<u>3.3.3 Bloc de puissance a base de relais.</u>	<u>49</u>
3.3.3.1 Branchement du bloc avec la carte Arduino.....	49
<u>4 Réalisation PCF</u> :.....	<u>50</u>
<u>4.1 Composants utilisés</u> :.....	<u>51</u>
<u>5. Tests</u> :	<u>54</u>
<u>6. Discussion</u>	<u>58</u>
Conclusion générale.....	59
Bibliographie.....	60

Introduction

Introduction générale

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. Si bien que faire de l'électronique est devenu accessible à toute personne.

L'apparition des circuits intégrés programmables tel que les microprocesseurs, les microcontrôleurs, et les mémoires a révolutionné le monde de l'électronique. En effet, l'association de ces derniers avec les composant analogique (transistors, thyristors, triacs,...), nous a permis de crée des cartes électroniques programmables qui peuvent commandes tout les processus industriels robotique, domotique, électroménager [1].

L'une de ces cartes électroniques est l'ARDUINO ATmega2560, qui est équipée d'un microcontrôleur ATmegaz2560 (AVR) de type CMS. Ce dernier dispose des entres/sorties logique programmable, des entres analogiques et d'un protocole série (RS232).

Associer une réalisation électronique an cycle solaire peut remédier à résoudre plusieurs problèmes. En effet, notre réalisation consiste à commande un simulateur de cycle solaire qui est un gradateur de lumière, avec une carte programmable qui nomme ARDUINO ATmega2560.

Dans le cadre de projet de fin d'études, nous avons réalisé, un simulateur de cycle solaire à base d'arduino.

Nous avons structuré notre mémoire comme suite :

Dans le premier chapitre nous donnons la présentation de notre projet.

Le second chapitre nous allons décrire d'une façon détaillée la carte Arduino mega.

Le troisième chapitre est consacré à la description des différents composants utilisé dans notre projet.

Dans le dernier chapitre, nous présenterons les étapes de réalisation pratique de notre simulateur.

Nous terminons notre travail par une conclusion générale en donnant des perspectives.

Chapitre I

Présentation de projet

1. Préambule :

Le lever et le coucher du soleil ne sont pas des moments anonymes de la journée. Ils régulent également notre humeur et notre énergie. Le jour, la lumière du soleil nous vivifie ; la nuit.

La plus simple observation du ciel nous montre que le Soleil se lève chaque matin à l'Est; il s'élève progressivement, puis s'abaisse et va se coucher à l'Ouest.

En se tournant vers dans la direction opposée (soit vers le Nord ou le Sud respectivement), on observe encore ce mouvement des étoiles; mais certaines d'entre elles offrent cette particularité qu'elles ne descendent jamais au-dessous de l'horizon et décrivent un cercle entier autour du pôle céleste. Celui-ci restant immobile dans le ciel. Le mouvement, de ces étoiles, qui sont à l'instar des constellations qu'elles forment, dites circumpolaires semble dirigé d'Ouest en Est dans la partie inférieure de leur cercle.

2. Le Soleil :

Le soleil est l'origine principale de toute forme d'énergie sur terre. Ceci étant vrai autant pour les énergies conventionnelles d'origine fossile et l'étoile du Système solaire. Pendant toute la journée et jusqu'à la nuit, le phénomène majeur observable est la lumière du soleil.

Le Soleil est l'étoile autour de laquelle tourne la Terre, notre planète. C'est cette étoile qui nous donne la lumière et la chaleur, indispensables à la vie telle que nous la connaissons...

Le Soleil représente à lui seul 99,86 % de la masse du Système solaire ainsi constitué.

2.1 Les Caractéristiques générales :

La chromosphère et les protubérances sont les sources essentielles, bien plus chaudes que « la surface » (la photosphère).

Le Soleil est une étoile naine jaune qui se compose de 74 % d'hydrogène, de 24 % d'hélium et d'une fraction d'éléments plus lourds. Le Soleil est de type spectral G2–V. « G2 » signifie qu'il est plus chaud (5 770 kelvins en surface environ) et plus brillant que la moyenne, avec une couleur jaune tirant sur le blanc. Son spectre renferme des bandes de métaux ionisés et neutres, ainsi que de faibles bandes d'hydrogène. Le suffixe « V » (ou « classe de luminosité ») indique qu'il évolue actuellement, comme la majorité des étoiles, sur la séquence principale du diagramme de Hertzsprung-Russell : il tire son énergie de réactions de fusion nucléaire qui

transforment, dans son noyau, l'hydrogène en hélium, et se trouve dans un état d'équilibre hydrostatique, ne subissant ni contraction, ni dilatation continues.

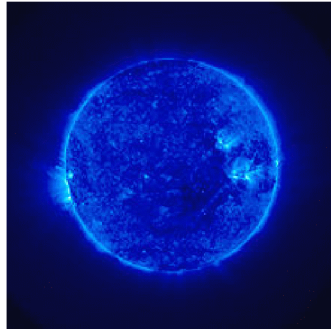


Figure (I.1) : Caractéristique de soleil

2.2 Le cycle solaire :

2.2.1. Définition le cycle solaire :

Un cycle solaire est une période pendant laquelle l'activité du Soleil varie en reproduisant les mêmes phénomènes que pendant la période de même durée précédente.

Vue de la Terre, l'influence du Soleil varie principalement selon une période journalière et annuelle. Dans l'absolu, l'activité solaire est réglée par un cycle d'une période moyenne de 11,2 ans d'un maximum au suivant mais la durée peut varier entre 8 et 15 ans. L'amplitude des maxima peut varier du simple au triple Les taches solaires

2.4.2. Les différents phases du cycle solaire chaque journée :

2.4.2.1. Le Jour :

Lumière du soleil sur la terre; cette lumière considérée suivant la position de la terre relativement au soleil. Le jour est la plus longue section du cycle.

Pendant le jour, on peut voir le soleil et le ciel est bleu clair. Pendant le jour, tous les blocs qui sont directement sous le ciel reçoivent la plus forte lumière possible.

2.4.2.2. Le Crépuscule :

Lumière faible et incertaine qui subsiste après le coucher du soleil avant que la nuit ne soit complètement tombée; moment correspondant de la journée.

Il s'agit d'une période entre le jour et la nuit, où le soleil se couche à l'ouest et où la lune se lève à l'est.

Tous les blocs qui sont directement sous le ciel perdent un niveau de luminosité toutes les 10 secondes.



Figure (I.2) : Exemple d'un coucher de soleil vu depuis un point élevé de la carte

Le cycle jour/nuit aura toujours la même durée et les mêmes effets même si le soleil est clairement plus petit ou plus grand.

Le soleil devient visiblement plus grand pendant le crépuscule.

2.4.2.3. La Nuit :

Durée comprise entre le coucher et le lever du Soleil et pendant laquelle ce dernier n'est pas visible ; obscurité plus ou moins grande qui accompagne cette durée.

2.4.2.4. L'Aube :

Est le moment de la journée où apparaissent à l'horizon est les premières lueurs du jour, avant le lever du soleil. Elle est également appelée le « petit matin » dans le langage courant ou le « point du jour » dans des textes plus anciens.

Il s'agit d'une période entre la nuit et le jour, où la lune se couche à l'ouest et où le soleil se lève à l'est. Tous les blocs directement sous le ciel gagnent un niveau de luminosité toutes les 10 secondes.

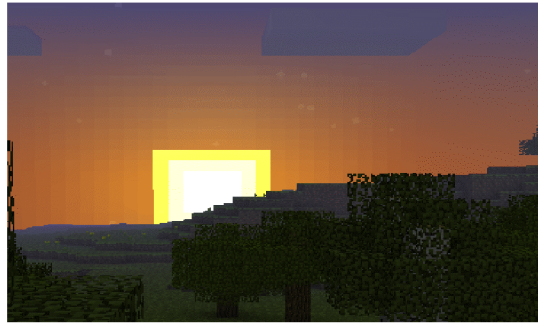
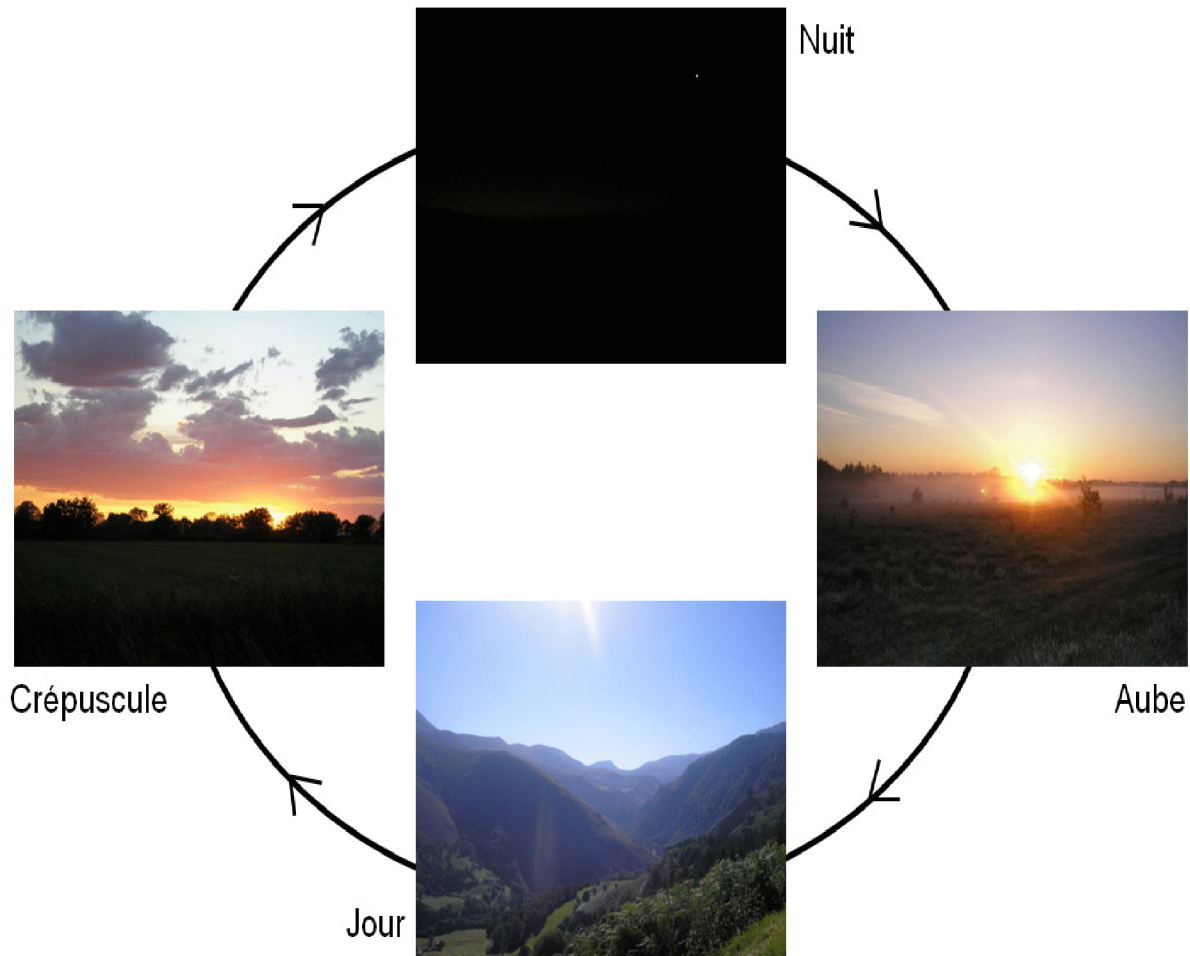


Figure (I.3) : Exemple d'un lever de soleil à l'horizon

3. Présentation du projet :

3.1. Réalisation un simulateur du cycle solaire a la base d'Arduino :

Ce montage a été conçu pour allumer très lentement des lampes à filament, de manière à simuler l'aube, le jour, le crépuscule et la nuit. Il s'agit d'un circuit qui allume très lentement une lampe pour simuler l'effet de l'aube et, quand cette lampe a atteint son maximum de luminosité, il la maintient un certain temps pour simuler l'effet du jour, après quoi, toujours lentement, il commence à l'éteindre pour simuler l'effet du crépuscule ; puis, quand la lampe est éteinte, il la maintient dans cet état un certain laps de temps pour simuler la nuit. Dans la mesure ne doit pas passer 24 heures, le circuit doit être doté d'une régulation manuelle, de manière à faire varier la durée de chaque cycle de quelques secondes à plusieurs minutes



Figure(I.4) : Schéma de cycle solaire avec ces différentes phases

4. Discussion :

L'évolution des méthodes d'observation répond aussi à une demande des pays industrialisés dans des domaines divers aussi bien à court terme qu'à long terme tels que la prévision des conditions climatiques qui est considérée à ce jour comme indispensable, mais surtout rentable.

De nouvelles méthodes de prévision des cycles solaires. Ces méthodes de prévision sont assez complexes et ne pourront être validées que par une observation systématique s'étendant sur plusieurs dizaines d'années.

Chapitre II

Présentation de la carte Arduino

1 préambule :

Pour la réalisation de notre système qui est un simulateur de cycle solaire, nous avons utilisé une carte d'acquisition de type Arduino afin d'assurer la communication entre les deux parties à savoir ; la partie commande (ordinateur) et les deux sites (maître et esclave) afin de pouvoir commander et piloter le système.

Dans ce travail, nous présentons la description de la carte Arduino qui sera utilisée pour notre application. En effet, les différentes parties constituant cette carte seront détaillées, à savoir un microcontrôleur, des ports d'entrée/sorties, une mémoire de type EPROM. Aussi, le logiciel IDE utilisé pour la programmation sera présenté dans ce chapitre

2 La carte d'acquisition :

2.1 Définition de la carte Arduino :

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

Arduino peut être utilisé pour développer des objets interactifs, pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles. Les projets Arduino peuvent être autonomes, ou bien ils peuvent communiquer avec des logiciels tournant sur l'ordinateur (tels que Flash, Processing ou MaxMSP, Matlab).

Le langage de programmation Arduino est une implémentation de Writing, une plate forme de développement similaire, qui est basée sur l'environnement multimédia de programmation Processing.

Ces cartes sont basées sur une interface entrée/sortie simple et sur un environnement de développement proche du langage C.

Elle contient tout le nécessaire pour piloter le microcontrôleur, il suffit simplement de la connecter à un ordinateur avec un câble USB pour l'utiliser simplement (fournit avec la carte).

2.2 Historique

D'après Wired Magazine : Le projet Arduino est né en hiver 2005. Massimo Banzi enseigne dans une école de Design à Ivrea en Italie, et souvent ses étudiants se plaignent de ne

pas avoir accès à des solutions bas prix pour accomplir leurs projets de robotique. Banzi en discute avec David Cuartielles, un ingénieur Espagnol spécialisé sur les microcontrôleurs...

Ils décident de créer leur propre carte en embarquant dans leur histoire un des étudiants de Banzi, David Mellis qui sera chargé de créer le langage de programmation allant avec la carte. En deux jours David écrira le code! Trois jours de plus et la carte était créée... Ils décidèrent de l'appeler Arduino (un bar fréquenté par les élèves à proximité de l'école)...

Il devient un hit tout de suite auprès des étudiants. Tout le monde arrive à en faire quelque chose très rapidement sans même avoir de connaissances particulière ni en électronique ni en informatique: réponse à des capteurs, faire clignoter des leds, contrôler des moteurs... Ils publient les schémas, investissent 3000 euros pour créer les premiers lots de cartes:

Les 50 premières partent directement à des élèves de l'école. En 2006 5 000 cartes vendues... En 2007 plus de 30 000, En 2011 plus de 120 000, sans compter les clones [2].

2.3 Les différentes cartes Arduino :

Il existe plusieurs est différentes carte arduino qui sont été fabrique par la société smart Projects.

Les plus utilisens sont :

- ✚ Arduino Duemilanove : utilise un Atmega168 et est alimenté en électricité par le connecteur USB ou une alimentation externe avec commutation automatique. La nouvelle version est équipée d'un ATmega328
- ✚ Carte arduino uno : utilise un ATmega328 comme les derniers modèles de Duemilanove, mais alors que le Duemilanove utilisait une puce FTDI pour la programmation via un connecteur USB, la Uno utilise une puce ATmega8U2 programmé comme un convertisseur série
- ✚ Carte arduino mega1280 : est équipé d'un ATmega1280 de type CMS (composant monté en surface) ou SMD (surface mounted device), pour avoir des Entrées/Sorties supplémentaires et de la mémoire.
- ✚ Arduino Diecimila : possède une interface USB et utilise un ATmega168 dans un boitier format DIL28.
- ✚ Arduino Ethernet : est une carte Arduino UNO intégrant un chip Wiznet W5100 pour rajouter la connectivité Ethernet intégré

- ✚ Carte arduino mega 2560 : est équipé d'un ATmega2560 de type CMS.
- ✚ Arduino DUE : est une évolution de l'Arduino Mega2560 avec un microcontrôleur 32 bits Atmel SAM3X [3].

2.4 Caractéristique de la carte Arduino :

Parmi les caractéristiques les plus importantes de la carte Arduino, nous avons retenu les suivantes :

- **pas cher** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes.
- **multiplateforme** : le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- **un environnement de programmation clair et simple** : l'environnement de programmation Arduino est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **logiciel Open Source et extensible** : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés.
- **matériel Open source et extensible** : les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, ATMEGA 2560 etc. Les schémas des modules sont publiés sous une licence Créative Commons, et les concepteurs de circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant [4].
- La carte Arduino est capable de stocker un programme et de le faire exécuter.
- La carte reçoit des informations analogiques ou numériques sur ses entrées et génère des informations analogiques ou numériques.

- Le microcontrôleur traitera ces informations et les transmettra vers les sorties numériques.
- Récupère les données des capteurs pour les transmettre à l'interface de commande (pc) et traduire les instructions pour faire fonctionner les actionneurs.

3 Description de la carte Arduino Mega :

Le système de contrôle est réalisé grâce à une carte arduino mega , nous avons choisi le contrôleur ATmega2560 pour sa puissance de calcul et sa facilité de développement d'un programme

La carte Arduino Mega 2560 est basée sur un ATmega2560 à 16 MHz. Elle dispose de 54 E/S dont 14 PWM 16 analogiques et 4 UARTs.

Elle est idéale pour des applications exigeant des caractéristiques plus complètes que la Uno.

Ce tableau nous donne les caractéristiques de la carte Arduino Mega :

Tension de fonctionnement	5v
Tension d'alimentation (recommandée)	7-12v
Tension d'alimentation (limites)	6-20v
broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5v)	40 mA
Intensité maxi disponible pour la sortie 3.3v	50 mA
Intensité maxi disponible pour la sortie 5v	Fonctionnement de l'alimentation utilisée- 500 mA max si port USB utilisé seul.
Mémoire programme flash	256 KB dont 8 KB sont utilisés par le boot leader.
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

Table II.1 : caractéristique de la carte Arduino Mega 2560

3.1 Architecture de la carte arduino Mega 2560 :

La figure suivante illustre le schéma synoptique de la carte Arduino.

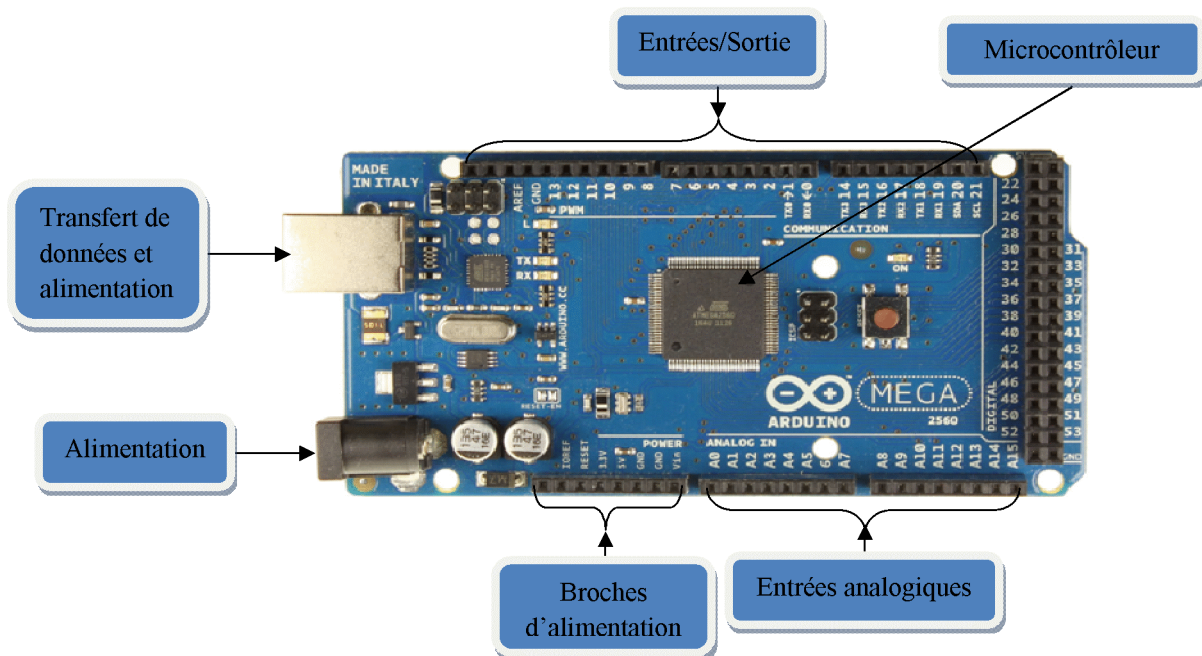


Figure (1):Architecture de la carte Arduino méga 2560

Les éléments de base qui composent la carte Arduino sont :

3.2 Le microcontrôleur :

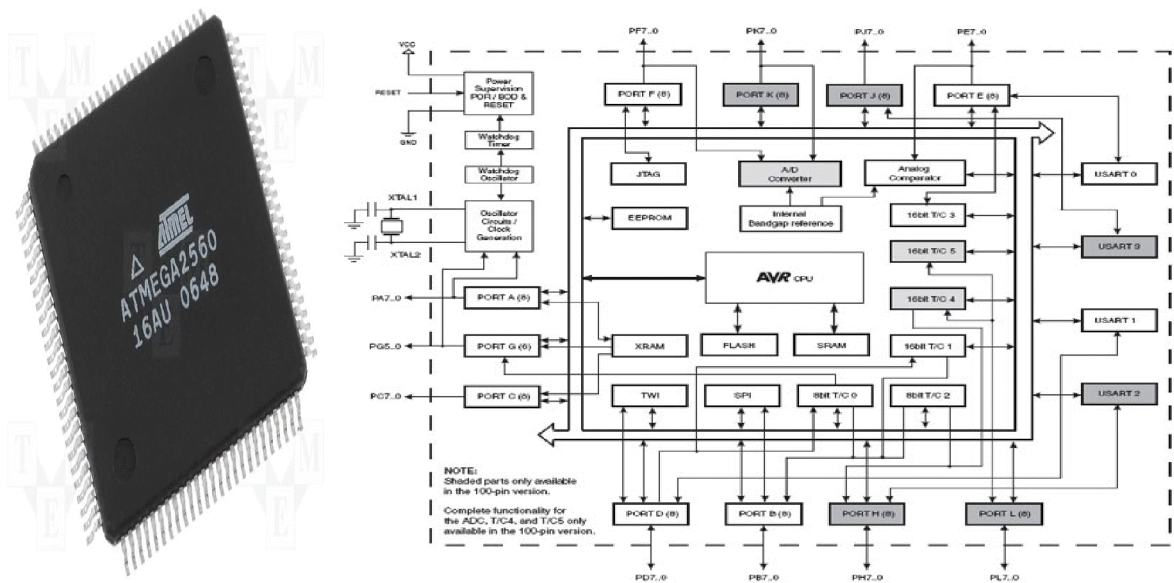
C'est le cerveau de la carte, permettant de recevoir le programme créé et stocké dans sa mémoire, puis l'exécuter. Grâce aux instructions de ce programme, il peut accomplir des tâches souhaitées, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, commander des moteurs comme dans notre système, envoyer des données à un ordinateur... etc.

3.2.1 Caractéristiques du microcontrôleur ATMEGA 2560 :

- ✓ 135 instructions puissantes, la plupart d'exécution simple de rythme.
- ✓ 32x8 registres d'usage universel de fonctionnement.
- ✓ Opération entièrement statique.
- ✓ Jusqu'à 16 MIPS de sorties à 16 Mhz.
- ✓ Multiplicateur de cycle de Sur-Morceau2.
- ✓ Mémoires non-volatiles de programme et de données.

- ✓ Bytes 64K/128K/256K de flash Individu-Programmable de Dans-Système.
- ✓ Resistance : 10.000 écrire /cycles d'effacement.
- ✓ 4K Bytes EEPROM.
- ✓ Serrure de programmation pour la sécurité de logiciel.
- ✓ Quatre canaux à 8 bits de PWM.
- ✓ Interface série à 2 fils orienté vers le byte.
- ✓ Puissance sur la remise et la détection programmable d'arrêt partiel.

La figure ci-dessous montre la structure interne du microcontrôleur ATMEGA 2560



Figure(2): Architecture du microcontrôleur ATMEGA 2560

3.2.2 Présentation physique du microcontrôleur ATMEGA2560 :

L'ATMEGA se présente sous la forme d'un circuit intégré à 100 broches pour le modèle ATMEGA2560, dans un type de boîtier MCU TQFP [5].

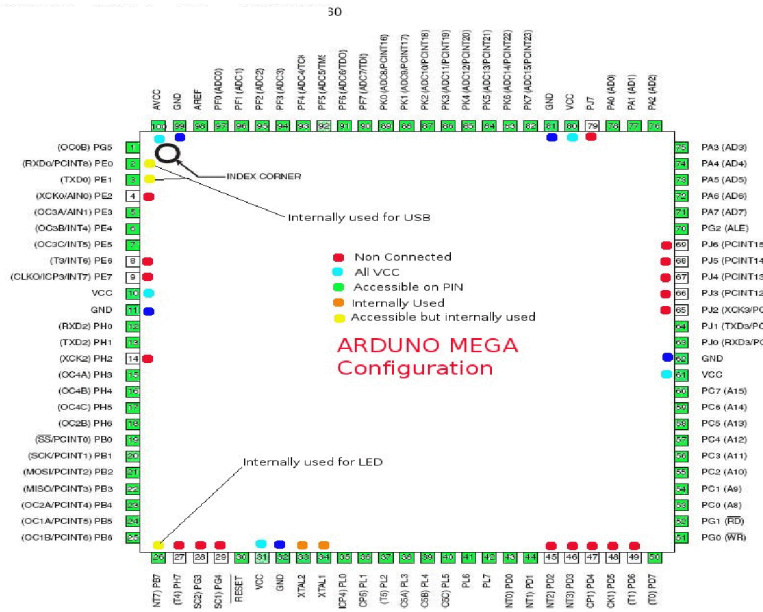


Figure (II.3) : microcontrôleur atmega2560

A. Description des branches :

- **Ports :** le microcontrôleur dispose de 11 ports (port A...port L) chaque port est un port d'entrée-sortie à 8 bits bidirectionnel sauf le port G qui est une sortie à 6 bits avec des résistances internes de tirage (pull-up) (choisi pour chaque bit).

Egalement utiliser pour des fonctions des différentes particularités.

- **RESET :** déclenché par un front descendant maintenue plus de 50 ns il produira le Reset du

Microcontrôleur, même si l'horloge ne court pas.

- **XTAL1 :** Entrée de l'oscillateur externe ou libre pour l'horloge interne.
- **XTAL2 :** Production de l'amplificateur d'oscillateur.
- **AREF :** est l'entrée de référence analogue pour le Convertisseur A/D avec une tension dans la gamme de 2 avec filtre passe bas.

B. L'horloge système :

Le choix du type d'horloge est déterminé lors de la programmation de la mémoire FLASH du Microcontrôleur.

L'oscillateur à quartz externe est connecté sur XTAL1 et XTAL2 pour cadence le microcontrôleur. La fréquence du quartz est de 16 MHz.

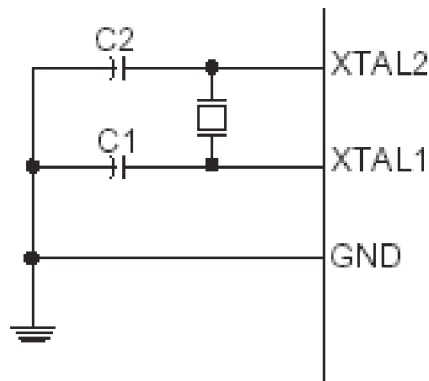


Figure (II.4) : montage du quartz

En employant l'oscillateur de type quartz externe, un amortisseur capacitif de l'ordre de 12 à 22 pF doit être connecté comme indiqué dans la figure 4.

C. Les registres :

➤ Registre système :

Les registres systèmes permettent de programmer le microcontrôleur selon le choix d'utilisation que le programmeur veut en faire. Ils sont au nombre de 4, mais seuls les deux premiers sont importants. Le fonctionnement du microcontrôleur est modifié lors de la programmation de la mémoire FLASH.

- Registre MCUCR (*MCU Control Register*) : C'est le registre de contrôle de l'unité centrale du microcontrôleur. Ce registre détermine le fonctionnement en mode sommeil et la gestion des interruptions externe 0 & 1.
- Registre MCUSR (*MCU Control and Status*) : Ce registre permet de connaître le statut du microcontrôleur.
- **Registre OSCCAL (*Oscillator Calibration Register*)** : Le registre de calibrage de l'oscillateur pour la programmation de la mémoire FLASH et l'EEPROM.
- Registre SPMCR (*Store Program Memory Control Register*) : Le registre de contrôle de sauvegarde de la mémoire de programme (SPM) contient les bits pour contrôler les opérations de chargement des programmes dans la mémoire FLASH en mode 'Boot' [6].

➤ Registre d'état et de pile :

Deux registres sont indispensables au système pour fonctionner, le registre d'état SREG et le registre de gestion de la pile pour les interruptions et la gestion des sous-programmes.

- Registre SREG : Le registre SREG ou registre d'état sert principalement avec les fonctions arithmétiques et logiques pour les opérations de branchements. Il indique et autorise aussi le fonctionnement des interruptions.
- Registre de pile : Le registre de pointeur de pile est utilisé par les instructions *PUSH* et *POP* de gestion de pile.

D. Interface série UART :

L'USART est l'abréviation de (Universal Synchronous and Asynchronous Receiver and Transmitter), l'application principale de ce périphérique est la communication entre le microcontrôleur et l'ordinateur.

L'ATmega2560 à quatre ports série (UART0, UART1, UART2, UART3), chaque port à une entrée RX est une sortie TX.

E. Interface série SPI :

L'interface SPI est l'abréviation (Serial Peripheral Interfacel Synchronous), soit Interface Série Synchrones. Contrairement à l'UART et comme son nom l'indique, ce type de périphérique génère les signaux d'horloge de synchronisation.

L'interface SPI permet le transfert de données ultrarapide synchrone entre l'ATMEGA et des périphériques ou entre plusieurs dispositifs.

F. Interface série I2C ou TWI :

L'interface à deux conducteurs périodique TWI est un dérivé de l'interface I2C de Philips, c'est une nouveauté du modèle ATMEGA, il n'existe pas dans les versions précédentes AT89 et AT90.

Le protocole TWI permet de connecter jusqu'à 111 systèmes différents employant seulement deux lignes de bus bidirectionnelles, un pour l'horloge SCL (Horloge de synchronisation du bus) et un pour les données SDA (Donnée transmise ou reçus du bus), Le seul matériel externe nécessaire pour la mise en œuvre du bus est un simple jeu de résistance pour chacune des lignes (R1, R2).

Tous les systèmes connectés au bus ont des adresses individuelles et les mécanismes de control sont inhérents au protocole I2C.

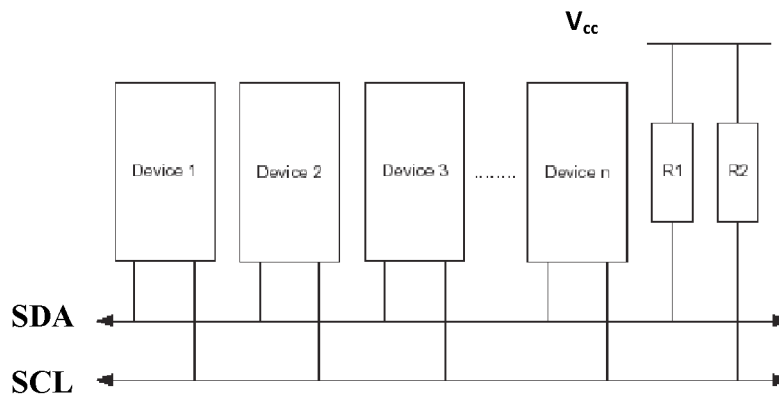


Figure (II.5) : Principe du bus I2C ou TWI.

Comme le montre la figure ci-dessus, les deux lignes du bus sont connectées à la tension positive par des résistances de tirage d'une valeur de 4,7 à 5 Kohm.

G. Le comparateur analogique (AC) :

Le comparateur analogique compare les valeurs d'entrée sur la broche positive **AIN0 (PB2)** et la broche négative **AIN1 (PB3)**. Quand la tension sur la broche **AIN0** est plus haute que la tension sur la broche **AIN1**, le comparateur analogique **ACO** est mis à 1. Le comparateur analogique peut être utilisé pour déclencher la fonction de capture d'entrée du Timer/Compteur1. De plus, le comparateur analogique peut déclencher une interruption séparée exclusive. L'utilisateur peut choisir le front montant ou descendant pour déclencher la sortie **ACO**.

H. Convertisseur analogique numérique (ADC) :

C'est un dispositif qui va convertir des grandeurs analogiques en grandeurs numériques. La valeur numérique obtenue sera proportionnelle à la valeur analogique fournie en entrée. Le convertisseur analogique/numérique **ADC** intégré dans l'ATMEGA à une résolution sur 10 bits, il renvoi des résultats de mesure sous forme d'entiers de 0 à 1023 [8][9].

II.3.3 Le entrées-sorties de la carte :

La carte Arduino possède 15 entrées analogiques, 54 entrées / sorties numériques dont les pins de (1 à 13) peuvent être utilisés comme des entrées/sorties analogiques.

- Les pins utilisés en numérique (PWM) ont une sortie binaire soit (0 logique qui correspond à l'état bas du pin ou 0V) ou (1 logique qui correspond à l'état haut du pin ou 5V) ce qui amène le fonctionnement des pins on mode tout ou rien.
- Les pins utilisés en analogique ont une sortie de valeur (0 à 255).

3.4 Mémoire :

Le ATmega2560 utilise trois types de mémoire, 128 Ko de mémoire flash pour stocker le code (dont 4 Ko est utilisée pour le boot loader), 8 Ko de SRAM et 4 Ko de mémoire EEPROM (qui peuvent être lues et écrites avec la bibliothèque EEPROM).

- **La mémoire programme :** La mémoire programme permet de stocker et de faire fonctionner le microcontrôleur, il contient 256 KB.
- **La mémoire de donnée :** La mémoire de donnée contient les 32 registres de travail, les 64 registres de commande et la mémoire SRAM pour les variables du programme.
- **La mémoire morte :** La mémoire morte est de type EEPROM d'accès plus complexe contiendra la configuration du programme et les données importantes qui seront sauvées pendant l'absence de courant électrique. Il contient 4 Ko de mémoire de donnée, est On peut écrire jusqu'à 100.000 fois dans l'EEPROM.

3.5 L'alimentation :

L'alimentation qui assure la distribution d'énergie aux différents composants de la carte Arduino, peut être effectuée via une connexion USB ou une alimentation externe. La source de courant est sélectionnée automatiquement.

L'alimentation externe (non USB) peut provenir d'un adaptateur CA/CC ou de piles, l'adaptateur peut être connecté par une fiche positive centrale de 2,1 mm dans l'alimentation de la carte. Les câbles de l'accu peuvent être insérés dans les prises de terre et de tension du connecteur d'alimentation.

La carte peut fonctionner sur une alimentation externe de 6 à 20 volts. Si moins de 7 volts sont fournis, la broche 5 V peut éventuellement fournir moins de 5 volts et la carte peut alors être instable. Si plus de 12 V sont utilisés, le régulateur de tension peut surchauffer et abîmer la carte. La tension recommandée varie entre 7 et 12 volts.

Les broches d'alimentation sont les suivantes :

- Tension d'entrée : la tension d'entrée de la carte Arduino quand une alimentation externe est utilisée (en opposition aux 5 volts de la connexion USB ou d'autres sources de courant régulées). On peut fournir une tension par cette prise ou, si l'alimentation est effectuée par un câble d'alimentation, y accéder via cette prise.

- 5 V : cette broche fournit une tension régulée de 5V. La carte peut être alimentée par le câble d'alimentation en courant continu, le connecteur USB (5V) ou la prise de terre sur la carte (7-12 V). Alimenter la tension par les broches 5 V ou 3,3 V contourne le régulateur et peut endommager la carte.

Une fois le programme chargé, on peut débrancher le cordon USB et connecter la carte soit à une pile, soit à un transformateur [7].

- **GND, Broche de masse (ou 0V).**

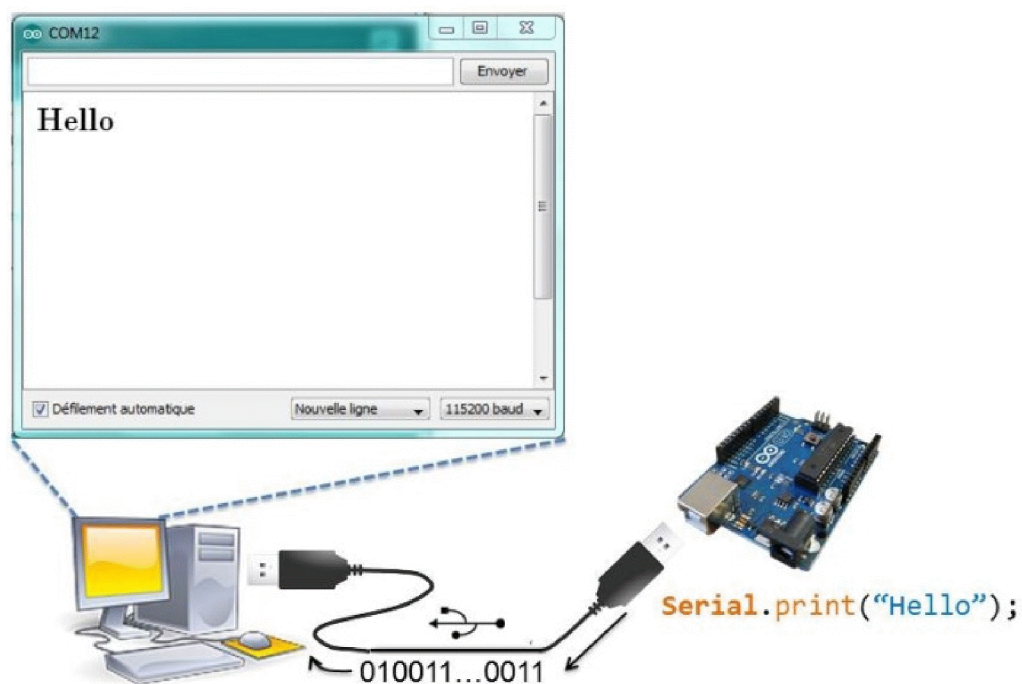


Figure (II.6) : alimentation de l'Arduino par un microordinateur

3.6 Visualisation :

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

- Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur qui peut servir pour tester le matériel. Quand on branche la carte au PC, elle clignote quelques secondes.
- Les deux LED du bas du cadre : servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

3.7 La connectique :

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

Sur les Arduino et sur beaucoup de cartes compatibles Arduino, les connecteurs se trouvent au même endroit. Cela permet de fixer des cartes d'extension, appelée " shield".

Cette connectique est importante et a un brochage qu'il faudra respecter. Nous le verrons quand nous apprendrons à faire notre premier programme. C'est avec cette connectique que la carte est "extensible", car l'on peut y brancher tous types de montages et modules.

Par exemple, la carte Arduino peut être étendue avec des shields, comme le « **Shield Ethernet** » qui permet de connecter cette dernière à internet.

3.8 Carte d'extension « SHIELD » :

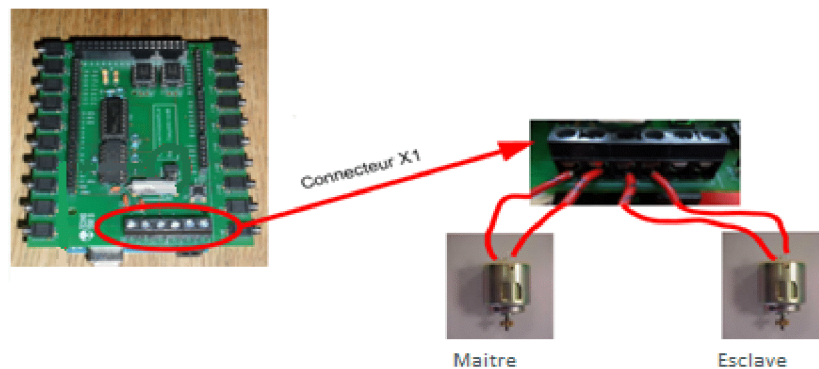
La carte EASYCON1 permet de connecter facilement 20 entrées-sorties à l'interface Arduino Méga. Les embases jacks stéréo 2,5mm servent à la connectique avec les cordons et les cartes I/O présentées. Extensible selon nos besoins, grâce à ces connecteurs stockables, et l'adjonction d'une carte EASYCON2 (18 entrées supplémentaires).

Deux kits optionnels de composants permettront d'étendre les capacités du shield en lui ajoutant :

- une commande de deux moteurs à courant continu (contrôle de la vitesse et de la direction).
- une ou deux "Banks" de mémoire EEPROM externe pour le stockage des projets autonomes.

3.8.1 Exemple d'usage de SHIELD pour connecter des moteurs à courant continu :

Comme on a évoquée précédemment l'existence des cartes shields pour Arduino. Voici un exemple qui permet de commander des moteurs en position et en vitesse en toute facilité, et offre aussi d'entrées/sorties supplémentaires [7].



Figure(II.7) : kit de commande de deux moteurs à

4 Architecture SOFTWARE de la carte :

Après la détermination des différentes entrées, sorties ainsi que la définition des conditions requises pour l'utilisation de la carte Arduino, nous allons décrire dans cette partie les éléments appropriés à l'architecture SOFTWARE de la carte, qui sont :

4.1 Logiciel :

L'environnement de programmation Arduino (IDE en anglais) est une application écrite en Java inspirée du langage Processing.

L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte.

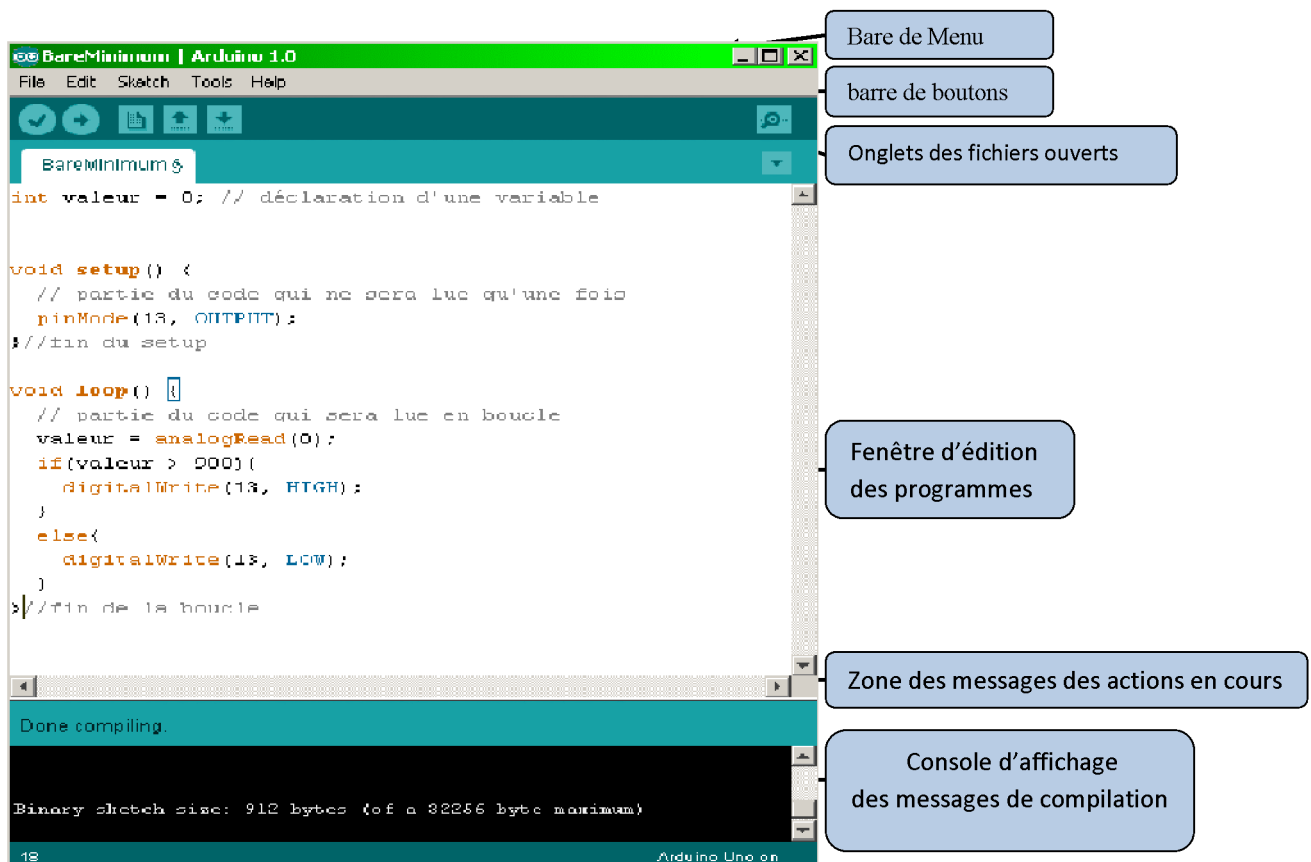
4.1.1 L'interface logicielle :

Sur un ordinateur, le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche de C); une fois le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers la liaison USB, le câble USB alimente à la fois en énergie la carte et transporte aussi l'information.

4.1.2 Structure de logiciel :

Le logiciel open source (Windows, Linux ou Mac) représenté par la figure ci-dessous fournit avec l'Arduino est un éditeur de texte qui permet :

- de programmer la carte en utilisant un langage simple proche du C.
- de communiquer avec la carte grâce au "terminal série" (faire apparaître des informations de la carte sur l'écran de l'ordinateur).



Figure(II.8): Structure d'un programme Arduino

4.1.3 Structure de programme :

Comme il est présenté sur l'interface du logiciel, le programme est lu par le microcontrôleur de haut vers le bas.

Une variable doit être déclarée avant qu'elle soit utilisée par une fonction.

La structure minimale est constituée :

- en tête : déclaration des variables, des constantes, indication de l'utilisation de bibliothèques etc.
- un setup (= initialisation) cette partie n'est lue qu'une seule fois, elle comprend les fonctions devant être réalisées au démarrage (utilisation des broches en entrées ou en sortie, mise en marche du midi, du port série de l'I2C, etc.)
- une loop (boucle) : cette partie est lue en boucle ! C'est ici que les fonctions sont réalisées.

En plus de cette structure minimale, on peut ajouter :

- des « sous-programmes » ou « routines » qui peuvent être appelées à tout moment dans la boucle, très pratique pour réaliser des morceaux de codes répétitifs.
- Des « callbacks », ce sont des fonctions qui sont rappelées automatiquement depuis une bibliothèque.

4.1.4 Programmation :

Arduino Méga peut être programmée avec le logiciel Arduino, l'ATMEGA 2560 sur Arduino Méga vient avec un boot loader qui permet de télécharger un nouveau code sans l'utilisation d'un programme externe.

La programmation de cette carte est composée de deux fonctions principales :

❖ Setup () :

Cette fonction est appelée dès l'exécution du programme. Elle a pour fonction d'initialiser les variables, indiquer les modes des broches, déclarer les bibliothèques, cette fonction s'exécute une seule fois, après chaque mise sous tension ou la réinitialisation de la carte Arduino.

❖ Loop () :

Cette fonction est appelée en Permanence et elle permet au programme de se répéter en forme de boucle principale, contient le contenu de programme.

4.2 Développement d'un projet :

Le développement sur Arduino est très simple :

- on code l'application : Le langage Arduino est basé sur les langages C/C++, avec des fonctions et des bibliothèques spécifiques à Arduino (gestions des e/s).
- on relie la carte Arduino au PC et on transfère le programme sur la carte.
- on peut utiliser le circuit.

Le logiciel de programmation des modules Arduino est une application Java multiplateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le firmware (et le programme) au travers de la liaison série (RS232, Bluetooth ou USB selon le module).

5. Discussion :

Dans ce chapitre, notre étude s'est focalisée sur la présentation de la carte (Arduino), et on a commencé par un bref historique et quelques détails concernant la création et l'utilité de cette carte. Puis on a parlé de son côté matérielle, enfin on a présenté le logiciel et le langage de programmation et Matlab qui est devenu un outil de commande et de programmation de cette dernière.

La compréhension de données exposées dans ce chapitre nous offre des bases de connaissances pour l'usage de la carte d'acquisition.

Chapitre III

Présentation des composants

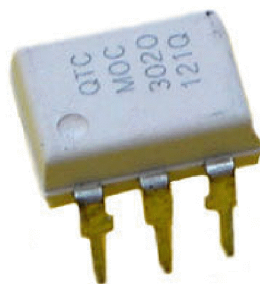
1. Préambule

Dans le cas de notre réalisation, nous avons utilisé l'optotriac, le triac et des potentiomètres dans ce chapitre, nous allons les caractéristiques essentielles de ces composants.

2. Présentation des composants :

2.1. L'optotriac :

C'est un composant électrique capable de transmettre un signal d'un circuit à un autre sans qu'il y de contact galvanique entre eux. Il est représenté par la figure (III.1).



Figure(III.1) : optotriac

2.1.2 Fonctionnement :

Un optotriac est un montage qui intègre un triac et une led. La mise en œuvre de ce dispositif est des plus simples. Puisqu'il suffit d'appliquer un niveau positif basse tension sur la led de l'optotriac, qui commande à son tour le triac. L'optotriac fait donc figure d'interface, en quelque sorte, entre le circuit de commande et le circuit commandé.

Ce composant assure à lui seul l'isolation électrique (galvanique) entre la partie commande et la partie de puissance.

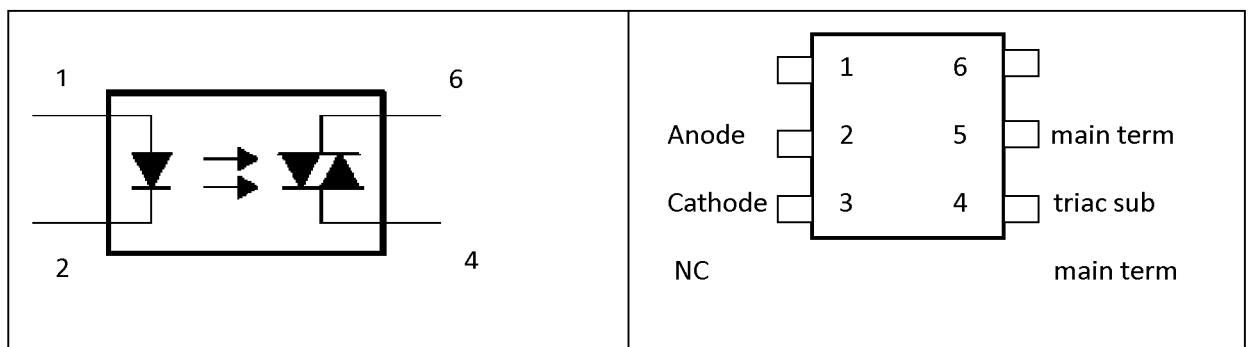


Figure (III.2) : schéma synoptique du MOC3020

Un optotriac est un montage qui intègre une LED en entrée (input) et un triac en sortie (output) :

A. L'entrée (input LED) :

L'entrée de l'optotriac est disponible sur la branche(1) et la branche(2), la branche (3) non utilisée (NC).

La tension admissible sur la branche (1) et (2) est la tension admissible aux bornes de la diode, soit 1,2 v typique et 1.5 v max, pour un courant de 50mA.

B. Sortie (output triac) :

La sortie est pilotée par l'entrée (led), lorsque qu'une tension est présente à l'entrée du montage, le triac s'amorce (courant dans la gâchette présent). En absence de tension en entrée, il faut attendre la détection du zéro de la sinusoïde (alternative) pour que le triac désamorce [10].

2.1.3 Caractéristiques électriques :

Les caractéristiques de l'optotriac sont présentées par le tableau suivant :

Paramètre	Condition d'essais	MIN	MAX	unité
IR courant inverse statique	$V_R = 3 \text{ v}$	0.05	100	
V _F tension directe statique	$I_F = 10 \text{ mA}$	1.2	1.5	
I(DRM) Répétitif courant à l'état, les deux sens	$V(\text{DRM}) = 400 \text{ V}$	10	100	nA
Dv/dt Taux critique de l'élévation de la tension à l'état bloqué	/	100		
IDF Taux critique de l'élévation de la tension de commutation	Moc3020	15	30	mA
	Moc3021	8	15	
	Moc3021	5	10	
	Moc3023	3	5	
courant de déclenchement d'entrée.	$I_{TM} = 100 \text{ mA}$			

Table III.1. Les caractéristiques de l'optotriac.

2.2 Le triac :

2.2.1 Description :

Le triac est un composant électronique au même titre qu'un thyristor ou un transistor.

Il est composé de cinq couches et par conséquent de quatre jonctions. Trois électrodes Permettent le lien du triac vers l'extérieur.

On peut visualiser un triac comme l'association tête-bêche de deux thyristors C'est en quelle que sorte un thyristor bidirectionnel.

Nous pouvons donc dire que le triac permet le contrôle d'un courant alternatif à partir de l'action sur une gâchette et ce quelle que soit le signe de la tension appliquée entre ses bornes Principales.

Le nom de triac provient de la contraction d'une expression anglo-saxonne : TRIode Alternative Current signifiant triode pour courant alternatif.

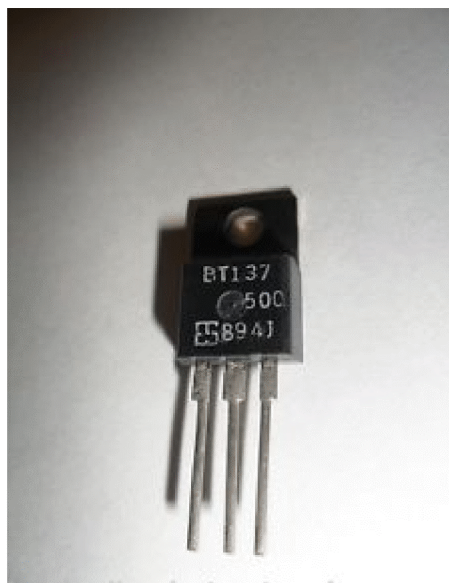


Figure (III.4) : triac BT137/500

2.2.2 Principe de Fonctionnement :

Un triac est un composant bidirectionnel, qui peut laisse passe le courant dans les deux sens, et comme le thyristor, ce compensant possède trois électrodes :

Deux électrode principales appelées A1 et A2 (pour Anode1 et Anode2) ou MT1 et MT2 (pour main terminal1 et main terminal2).

- ✚ Une électrode de commande G (pour Gâchette) qui permet d’amorcer ou de déclencher plus facilement la conduction du courant. Entre deux électrodes principales A1 et A2.

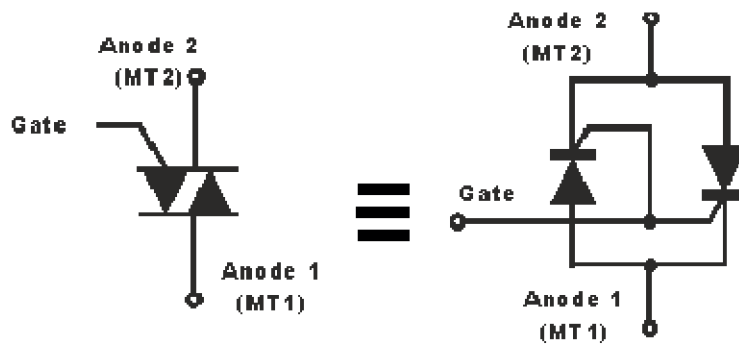


Figure (III.5) : Schéma synoptique du BT137/500

On peut dire que le triac est l'équivalent de deux thyristors montés tête-bêche. Mais comme le triac peut conduire dans les deux sens, son utilisation en alternatif est toute indiquée puisqu'on peut travailler avec les alternances positives et négatives [12].

2.2.3 Commande d'un triac :

Le triac peut être commandé dans quatre quadrants (1 à 4), ces quadrants correspondent simplement aux différentes polarités des courants et tensions auxquels peut être soumis le triac au travers ses trois électrodes.

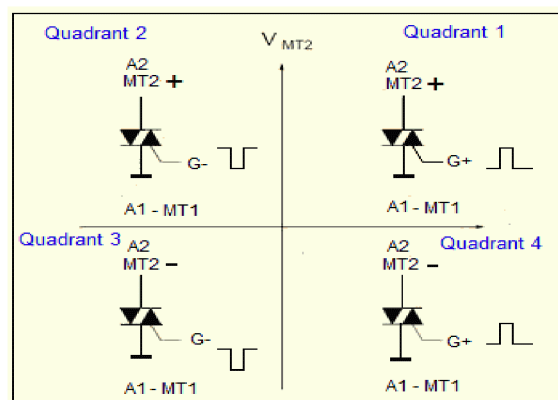


Figure (III.6) : les quadrants du triac

- Quadrant 1 : A2 est positif par rapport à A1, le triac est amorcé par une impulsion positive par rapport à A1.
- Quadrant 2 : A2 est positif par rapport à A1, le triac est amorcé par une impulsion négative par rapport à A1.
- Quadrant 3 : A2 est négatif par rapport à A1, le triac est amorcé par une impulsion négative par rapport à A1.
- Quadrant 4 : A2 est négatif par rapport à A1, le triac est amorcé par une impulsion positive par rapport à A1[13].

2.3 Transistors NPN BC547 :

2.3.1 Description :

C'est un transistor bipolaire et qui est un dispositif électronique à base de semi-conducteur de la famille des transistors. Son principe de fonctionnement est basé sur deux jonctions PN. L'une en direct et l'autre en inverse. La polarisation de la jonction PN inverse par un faible courant électrique (parfois appelé *effet transistor*) permet de « commander » un courant beaucoup plus important.

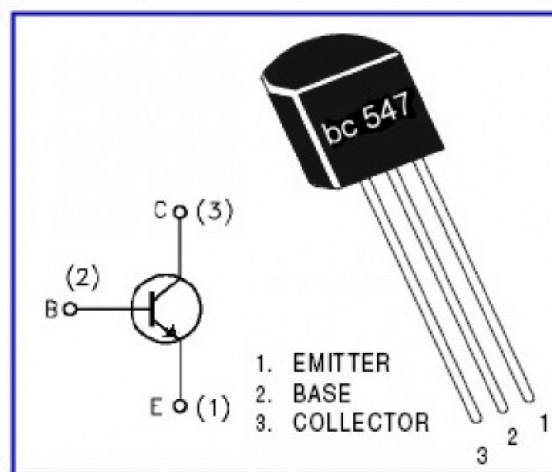


Figure (III.7) : transistor NPN BC547

2.3.2 Caractéristique du transistor NPN BC547 :

Le tableau suivant montre les principales caractéristiques du composant BC547 utiliser dans notre réalisation.

Symbole	Paramètre	VALEUR
BC547	Marquage du composant	/
U_{cb}	Tension collecteur-base (maximale)	50 v
U_{ce}	Tension maximale au-delà de laquelle le transistor risque d'être détruit	50v
U_{eb}	Tension émetteur (maximale)	6 v
I_c	Courant collecteur maximal	0.1 A
FT	Fréquence maximale de fonctionnement	300 MHZ
T_j	Température maximale de jonction	150 C
TO92	Type de boîtier	/

Table III.3. Les caractéristiques de transistor NPN BC547.

2.4 Les relais :

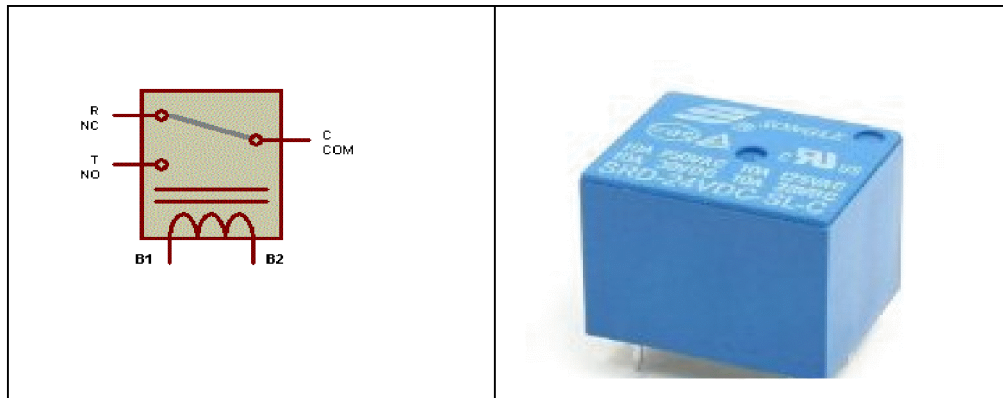
2.4.1 Principe :

Un relais est un composant qui permet de dissocier la partie puissance de la partie commande qui sert à l'ouverture/fermeture d'un circuit électrique par un second circuit complètement isolé (isolation galvanique).

En alimentant la bobine, le contact mobile est déplacé fermant ainsi le contact électrique. En l'absence de courant dans la bobine le ressort de rappel maintient le contact ouvert.

Il est constitué au moins de :

- ✚ d'un électroaimant (bobine+circuit ferromagnétique).
- ✚ d'une palette mobile supportant l contact mobile.
- ✚ ainsi qu'un contact fixe.
- ✚ d'un ressort de rappel du contact mobile.



Figure(III.8) : relai monophasé

2.4.2 Fonctionnement d'un relai :

Le relai fonctionne en monostable, les contacts commutent quand la bobine est alimentée et le retour à l'état initial se fait quand la bobine n'est plus alimentée.

2.4.3 Utilisation :

La fonction première des relais est le plus souvent de séparer les circuits de commande des circuits de puissance à des fins d'isolement, par exemple pour piloter une tension ou un courant élevé, à partir d'une commande plus faible, et dans certaines applications, assurer aussi la sécurité de l'opérateur.

2.5 Potentiomètre :

2.5.1 Présentation :

Un potentiomètre appelé aussi (résistance variable ou parfois Rhéostat).est un type de résistance à trois bornes, dont une est reliée à un curseur se déplaçant sur une piste résistante terminée par les deux autres bornes. Ce système permet de recueillir, entre la borne reliée au curseur et une des deux autres bornes, une tension qui dépend de la position du curseur et de la tension à laquelle est soumise la résistance.

2.5.2 Utilisation des potentiomètres :

Les potentiomètres sont couramment employés dans les circuits électroniques. Ils servent par exemple à contrôler le volume d'une radio. Les potentiomètres peuvent aussi être utilisés comme des transducteurs puisqu'ils convertissent une position en une tension. Ce type de

dispositif peut être rencontré dans des joysticks. Des potentiomètres de petite taille (les *trimmers* ou *trimpots*) se retrouvent fréquemment sur les circuits qui nécessitent des ajustements précis pour leur bon fonctionnement. Dans les appareils de la vie quotidienne, ces petits potentiomètres montés à même le circuit sont rarement destinés à être contrôlés par l'utilisateur final.

Un potentiomètre est un élément résistif possédant trois bornes:

- Deux correspondent aux extrémités du corps de la résistance,
- La dernière correspond au curseur qui peut se déplacer sur le corps de la résistance.

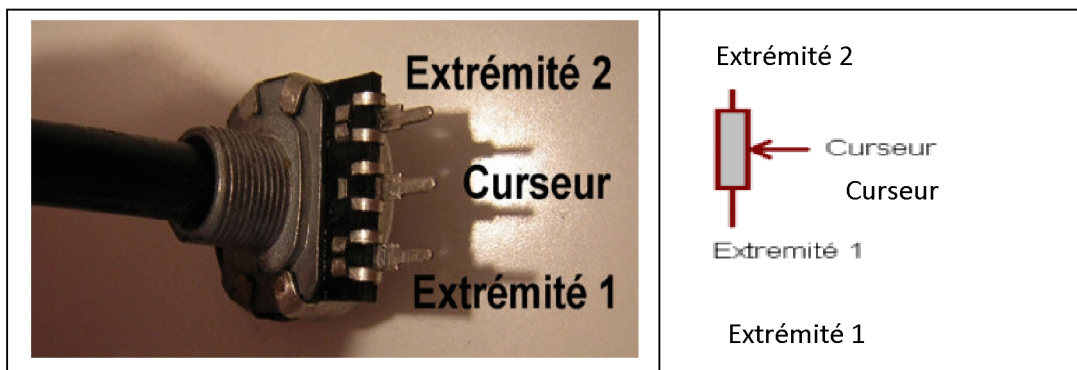


Figure (III.3) : Potentiomètre rotatif

2.5.3 Type de potentiomètres :

2.5.3.1 Potentiomètre linéaire :

Un potentiomètre linéaire est un potentiomètre dont la valeur de résistance varie proportionnellement à la distance entre ses bornes et le curseur. Il est utilisé, par exemple dans une source de tension variable.

La variation est progressive : quand le curseur se trouve au centre de la piste, la résistance ohmique que l'on peut mesurer entre le curseur et une extrémité est la même que celle que l'on peut mesurer entre le curseur et l'autre extrémité : $R_a = R_b$ (si le potentiomètre est un modèle 100 k Ω , $R_a = R_b = 50$ k Ω). Quand le curseur est à 80 % de sa course (plus vers l'extrémité haute), $R_a = 20$ % de la résistance totale et $R_b = 80$ % de la résistance totale. Il s'agit du type de potentiomètre qui est utilisé par défaut si rien n'est spécifié par l'auteur du

schéma électronique, sauf s'il s'agit d'un potentiomètre de volume (dans ce cas un modèle logarithmique est requis).

2.5.3.2 Potentiomètre logarithmique :

La résistance de ce type de potentiomètre varie de façon exponentielle, c'est-à-dire que la valeur de sa résistance augmente ou diminue de plus en plus rapidement lorsque l'on déplace le curseur.

La variation de la valeur de la résistance entre le curseur et une extrémité répond à une fonction logarithmique. Quand le curseur se trouve au centre de la piste, la résistance ohmique que l'on peut mesurer entre le curseur et une extrémité n'est pas la même que celle que l'on peut mesurer entre le curseur et l'autre extrémité.

Il est évident que l'on ne peut pas utiliser un potentiomètre de ce type dans une alimentation secteur pour ajuster finement la tension de sortie. En effet, la variation est lente quand le curseur se déplace vers une extrémité, et est très rapide quand le curseur arrive sur l'autre extrémité. Ce type de potentiomètre est principalement utilisé pour les réglages de volume sonore, pour « coller » à la caractéristique de l'oreille, qui possède justement une réponse logarithmique à la pression que l'air exerce sur les tympans. À cause de cette particularité, le respect du sens de branchement des deux extrémités de la piste résistive a bien plus d'importance que pour le potentiomètre linéaire.

Il est à noter que les limitations techniques rendent impossibles la fabrication de potentiomètres dont la variation de résistivité est véritablement continue ; en réalité, une mesure précise de la résistance selon la position du curseur donnera une fonction affine par morceaux approximant une fonction logarithmique, l'erreur dépendant de la qualité du potentiomètre.

2.5.3.3 Potentiomètre à prise médiane

Le potentiomètre à prise médiane, principalement utilisé dans les anciens amplificateurs, Ce type de potentiomètre continue d'être utilisé pour la lecture de position, et permet de faire une mesure différentielle entre le curseur et le point milieu.

Ce type de résistance variable est, électroniquement, presque équivalent à un potentiomètre ayant deux résistances fixes raccordées aux deux bornes qui ne sont pas le curseur.

2.5.3.4 Potentiomètre numérique :

Un potentiomètre numérique est un composant actif qui simule le comportement d'un potentiomètre analogique, mais à la différence de ce dernier, on ne fait pas varier sa résistance mécaniquement en le tournant. Il varie sa résistance en fonction d'une valeur numérique (souvent un octet) qu'il reçoit. Il ne peut donc prendre qu'un nombre fini de valeurs de résistances possibles. L'échelle de correspondance entre les n valeurs numériques et les différentes valeurs de résistances sont propres à chaque modèle de potentiomètre.

Ce type de potentiomètre est souvent limité à quelques dizaines de milliampères en entrée et à une tension maximale de 5 volts. Le changement de polarité à ses bornes peut poser problème, en faisant varier sa résistance au passage et donc en induisant des déformations non linéaires du signal [11].

2.5.4 variation d'un potentiomètre :

Il existe deux types de variation d'un potentiomètre plus répandues, la variation logarithmique est la variation linéaire.

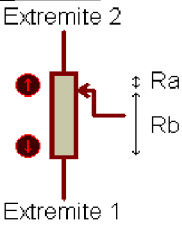
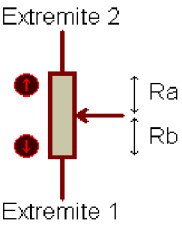
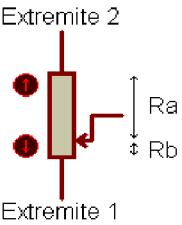
Position du curseur	Curseur à 80% de sa course totale	Curseur à 50% de sa course totale	Curseur à 20% de sa course totale
Schéma synoptique réalisé avec protus			
Pot-lin	Pot Lin : Ra = 20% Rb = 80%	Pot Lin : Ra = 50% Rb = 50%	Pot Lin : Ra = 80% Rb = 20%
Pot-log	Pot Log : Ra = 65% Rb = 35%	Pot Log : Ra = 85% Rb = 15%	Pot Log : Ra = 95% Rb = 5%

Table III.1. Les courbes de variation d'un potentiomètre.

3. Discussion

L'objectif de ce chapitre est de présenter tout les composants que nous avons utilisés dans notre réalisation, ainsi que leur liaison avec la carte d'acquisition.

Chapitre IV

Réalisation pratique

1. Préambule

Dans ce chapitre, on présentera notre réalisation qui est celle d'un simulateur du cycle solaire en utilisant une carte Arduino.

Le but essentiel de notre travail est d'utiliser l'Arduino pour commander le simulateur d'un cycle solaire, l'Arduino contient le programme qui nous permet de gérer la partie puissance.

Après avoir donné dans le chapitre précédent une description théorique sur le module Arduino et son environnement de développement, on va procéder à l'application pour cette raison, plusieurs blocs ont été nécessaires afin de réaliser un tel système.

2. Les différentes étapes de la réalisation :

Afin de réaliser notre système, nous l'avons devisé en trois parties.

A. Conception de tout le système électronique :

- chercher les différentes structures des blocs constituant notre maquette et qui vont avec les objectifs fixés et les moyens disponibles.
- Présenter les différents blocs de la réalisation.

B. Réalisation pratique de la carte :

- Présenter les différentes étapes de la réalisation pratique de la carte.
- On assemble ensuite les composants suivant notre montage sur la Breadboard (carte d'essai).

C. Création d'un programme via l'IDE Arduino :

- programmation de la carte d'acquisition qui est la carte Arduino mega2560.

3. Schéma synoptique générale :

Le schéma synoptique général de notre dispositif est indiqué par la figure (IV.1).

Pour faciliter cette étude, on a décomposé le schéma synoptique en plusieurs blocs :

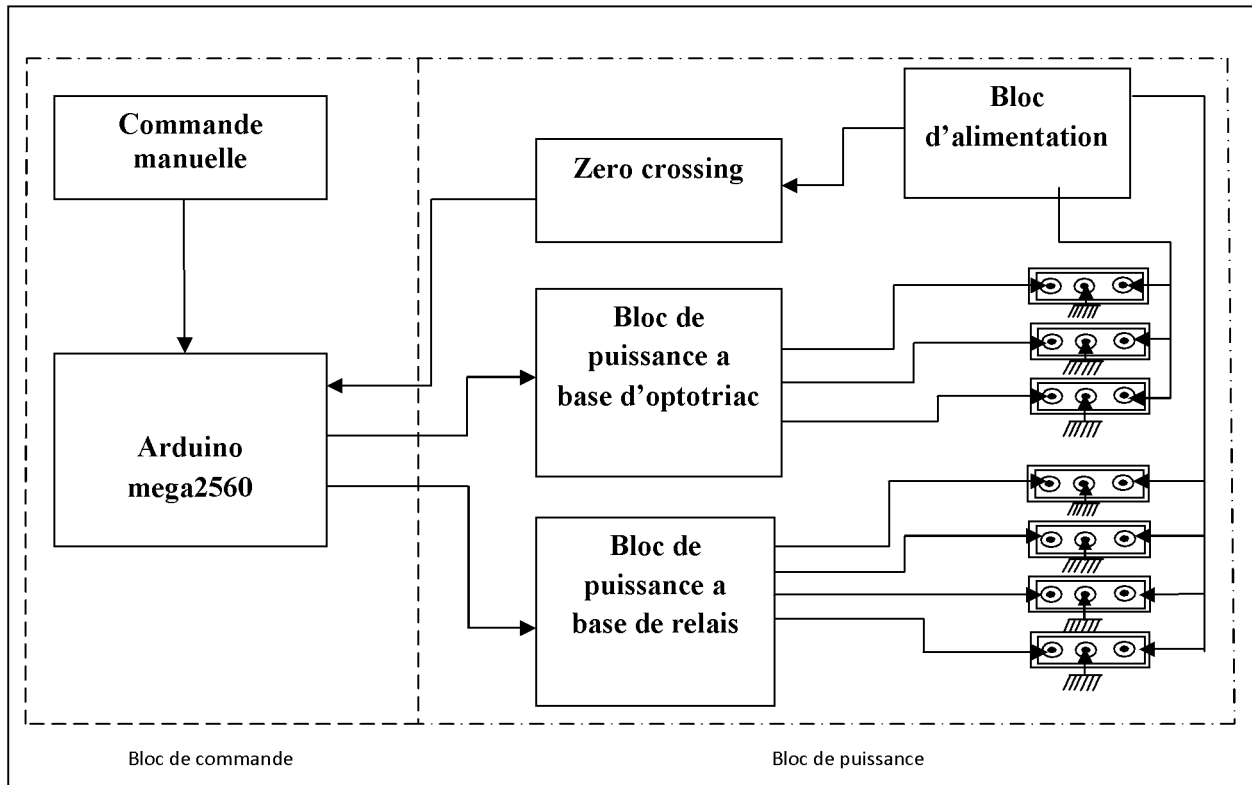


Figure (IV.1) : schéma synoptique générale

3.1. Bloc d'alimentation :

Le bloc d'alimentation contient, deux sources de tension (220V, tension d'alimentation de +5V).

3.1.1. Source externe AC 220V :

L'ensemble des blocs « bloc de puissance à base d'optotriac, et zero crossing » sont alimenté par une source de tension alternative (220V).

Nous avons prévu 3 sorties (1, 2 et 3) reliées à des TRIAC pour piloter en courant alternatif des lampes à filaments.

3.1.2. Tension de +5V :

Le fonctionnement de notre simulateur exige une tension d'alimentation de +5V, pour alimenté la partie commande.

Cette tension est produite par la carte Arduino.

3.2. Bloc de commande :

Notre bloc de commande, on le résume tout simplement par l'utilisation du module Arduino mega2560 qui est détaillé précédemment au chapitre II, Avec le bloc des commandes manuelles.

3.2.1. Commande manuelle :

C'est un sous bloc qui se compose de 4 potentiomètre variable, et un inverseur S1 (à trois position).

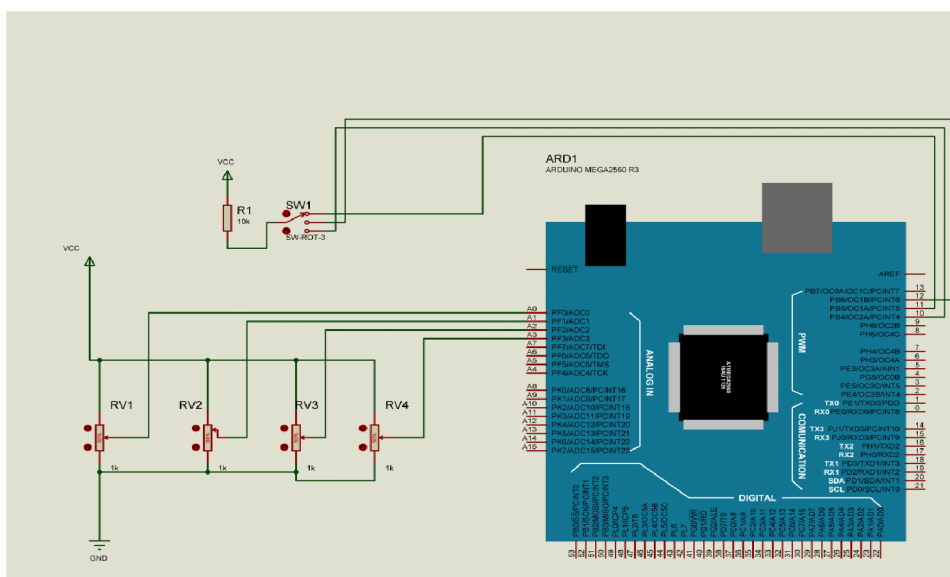
Les potentiomètres sont reliés à des entrées de la carte Arduino pour pouvoir commander la durée de luminosité des lampes à filament.

Branchement des potentiomètres avec la carte Arduino :

Nous avons relié les curseurs des potentiomètres avec les entrées Analogique de la carte Arduino

Le tableau ci-dessus montre le branchement de la carte Arduino avec les POT-HG.

Arduino mega	A0	A1	A2	A3	A8
Les potentiomètres	RV1	RV2	RV3	RV4	SW
Phase	Aube	Jour	Crépuscule	Nuit	/



Figure(V.1) : branchement du bloc commande manuelle avec l'Arduino sous

Chaque potentiomètre est utilisé pour varier la durée maximale d'une seule phase

Le potentiomètre RV5 est utilisé comme un inverseur de tension, il fonctionne comme suit :

A. RV5 en position haut (5V) :

- ✚ RV1 : Si nous tournons le curseur de potentiomètre RV1 complètement à droite, la tension à la sortie atteint sa valeur maximum en 40 minutes, si nous tournons vers la gauche la tension à la sortie attend sa valeur maximum en 3 minutes.
- ✚ RV2 : Si nous tournons le curseur de potentiomètre RV2 complètement à droite (5V) la tension à la sortie reste en maximum pendant 40 minutes, si nous tournons vers la gauche, la tension reste en maximum pendant 3 minutes.
- ✚ RV3 : Si nous tournons le curseur de potentiomètre RV3 complètement à droite la tension à la sortie atteint son minimum en 40 minutes, si nous tournons vers la gauche, la tension à la sortie attend son minimum en 3 minutes.
- ✚ RV4 : Si nous tournons le curseur de potentiomètre RV4 complètement à droite la tension à la sortie reste en minimum pendant 40 minutes, si nous tournons vers la gauche, la tension reste en minimum pendant 3 minutes.

B. RV5 en position centre (2.5V) :

- ✚ RV1 : Si nous tournons le curseur de potentiomètre RV1 complètement à droite, la tension à la sortie atteint sa valeur maximum en 6 minutes, si nous tournons vers la gauche la tension à la sortie attend sa valeur maximum en 30 secondes.
- ✚ RV2 : Si nous tournons le curseur de potentiomètre RV2 complètement à droite (5V) la tension à la sortie reste en maximum pendant 6 minutes, si nous tournons vers la gauche, la tension reste en maximum pendant 30 minutes.
- ✚ RV3 : Si nous tournons le curseur de potentiomètre RV3 complètement à droite la tension à la sortie atteint son minimum en 6 minutes, si nous tournons vers la gauche, la tension à la sortie attend son minimum en 30 secondes.
- ✚ RV4 : Si nous tournons le curseur de potentiomètre RV4 complètement à droite la tension à la sortie reste en minimum pendant 6 minutes, si nous tournons vers la gauche, la tension reste en minimum pendant 30 secondes.

C. RV5 en position bas (0V) :

- ✚ RV1 : Si nous tournons le curseur de potentiomètre RV1 complètement à droite, la tension à la sortie atteint sa valeur maximum en 1 minute, si nous tournons vers la gauche la tension à la sortie attend sa valeur maximum en 5 secondes.

- ✚ RV2 : Si nous tournons le curseur de potentiomètre RV2 complètement à droite (5V) la tension a la sortie reste en maximum pendant 1 minute, si nous tournons vers la gauche, la tension reste en maximum pendant 5 seconde.
- ✚ RV3 : Si nous tournons le curseur de potentiomètre RV3 complètement à droite la tension a la sortie atteint son minimum en 1 minute, si nous tournons vers la gauche, la tension a la sortie attend son minimum en 5 seconde.
- ✚ RV4 : Si nous tournons le curseur de potentiomètre RV4 complètement à droite la tension a la sortie reste en minimum pendant 1 minute, si nous tournons vers la gauche, la tension reste en minimum pendant 5 seconde.

3.3. Bloc de puissance :

Notre bloc de puissance se compose de deux sous blocs :

- Bloc de puissance à base d'optotriac.
- Bloc de puissance à base de relais.

3.3.1. Bloc de puissance à base d'optotriac :

Ce bloc se compose de trois triac chaque triac est relié à un optotriac, pour isoler la partie commande et la partie puissance (secteur).

3.3.1.1. Description :

Dans cette étape nous allons utiliser un gradateur, couramment appelé variateur, est un dispositif permettant de faire varier la puissance dans une charge raccordée au secteur. Cette charge est une ampoule dans notre application.

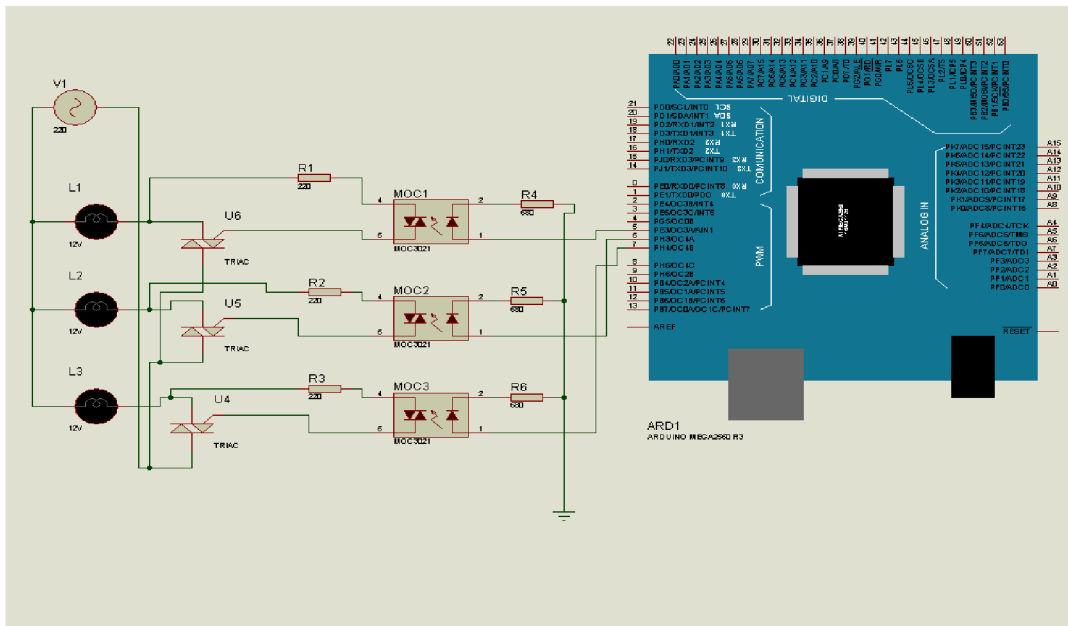
Le triac est assurément le composant tout désigné pour ce type d'application, le triac est ici associé à un composant galvanique.

Le gradateur que nous allons réaliser est destiné à une lampe d'éclairage. Il sera capable de faire varier l'intensité lumineuse de l'ampoule de (0 à 100%) d'une manière automatique, et autorisera un réglage très fin.

3.3.1.2. Branchement de bloc de puissance a base d'optotriac avec l'Arduino :

Le tableau ci-dessus montre le branchement de la carte Arduino avec les MOC3020 :

Arduino mega	5	6	7
Les MOC3020	MOC1 branche (1)	MOC2 branche (1)	MOC3 branche (1)



Figure(V.2) : branchement du bloc de puissance avec l'Arduino sous proteus

Lorsque la tension est à 5V, le courant dans la LED de MOC3020 permet la conduction de l'optotriac T1 (triac d'optotriac). Lorsque T1 est conducteur (il se comporte quasiment comme un fil), il pilote la gâchette du triac TRC1. C'est TRC1 qui pilote la charge secteur, comme un interrupteur. T1 n'est pas assez "puissant" pour piloter directement la charge secteur, il ne supporte que 100mA.

Lorsque la tension de commande repasse à 0V, l'optotriac reste passant jusqu'à l'annulation de son courant. L'annulation de ce dernier se fait grâce au système de zero crossing.

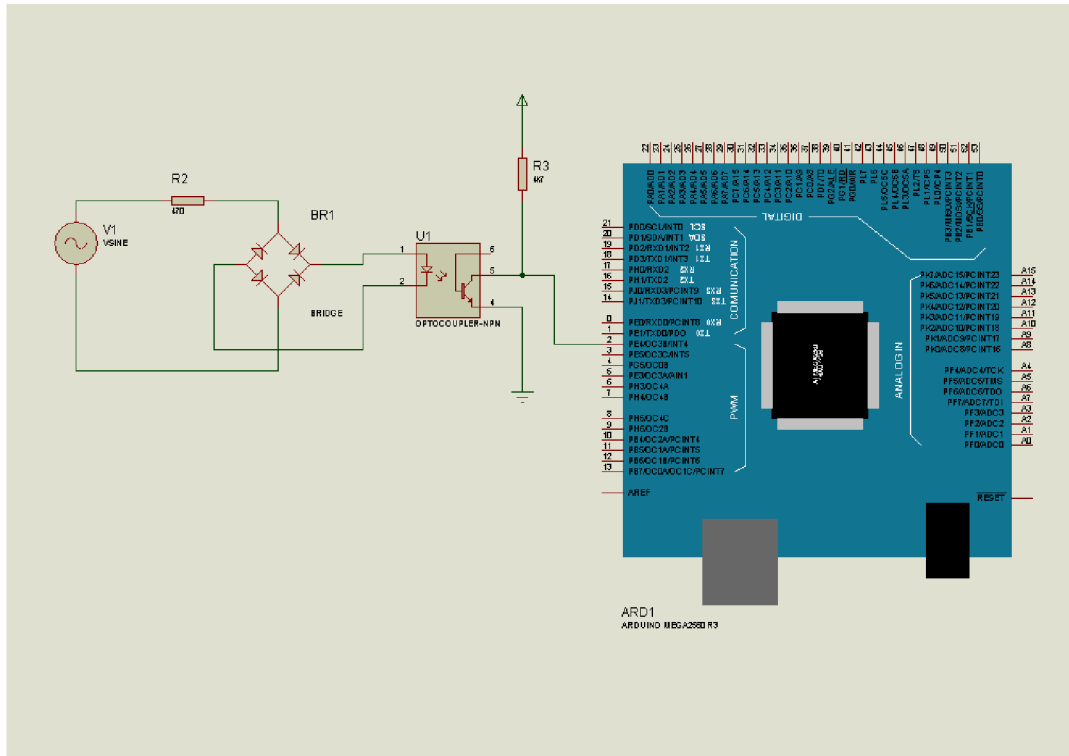
3.3.2. Zero grossing :

3.3.2.1. Description :

Un détecteur de passage par zéro est utilisé pour générer une impulsion de synchronisation en rapport avec l'angle de phase (courant alternatif) de tension souvent utilisé dans les circuits de commande de puissance.

Ce signal sert de synchronisme à la carte Arduino pour produire par les sorties que pilotent les optotriac, ces imputions.

3.3.2.2. Branchement du zero crossing avec la carte Arduino :



Figure(V.3) : branchement du zero crossing avec l'Arduino sous proteus.

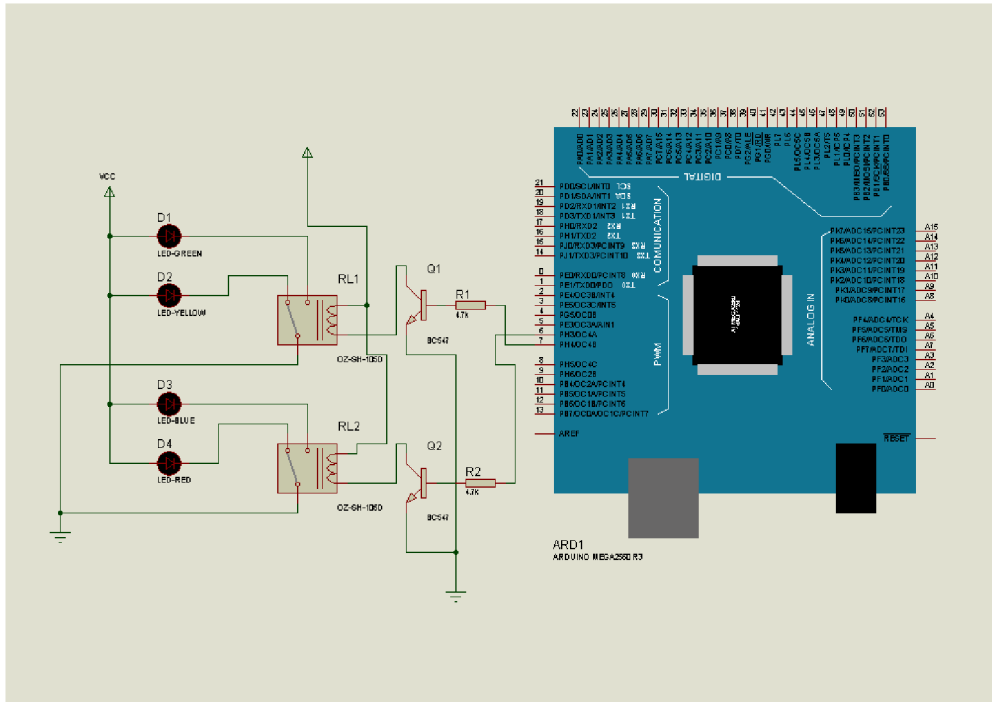
L'optocoupleur sert à isoler la détection du zéro secteur. En exploitant sa LED de commande qui est connecté à la branche 2(pwm)de la carte Arduino, on peut le rendre passant à chaque fois que le courant qui circule passe en-dessous d'une certaine valeur - en réalité bloquant, mais le signal est inversé en sortie à l'aide d'une résistance de *pull-up* (tension de tirage).

3.3.3. Bloc de puissance a base de relais :

Dans cette partie de puissance on a quatre sorties commandées par deux relais.

Les relais dévient la tension sur les 4 sorties et donc les sorties 4 (D1) et 6 (D2) sont sous tension quand le relais est relaxé et ne le sont pas quand il est excité. Au contraire, les sorties 5 (D3) et 7 (D4) sont sous tension seulement quand le relais est excité et ne le sont pas quand il est relaxé.

3.3.3.1. Branchement du bloc avec la carte Arduino :



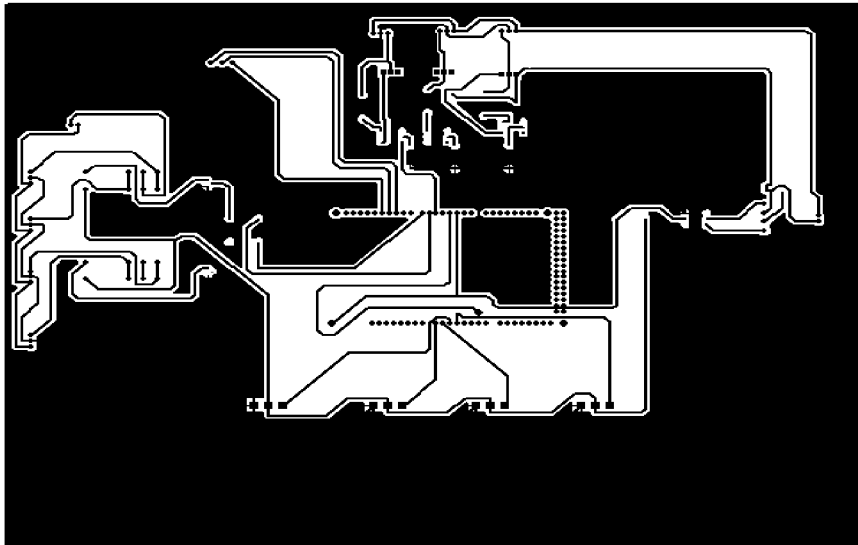
Figure(V.4) : branchement des relais avec la carte Arduino

4. Réalisation PCF :

Il s'agit de créer les liaisons entre les différents composants utilisés « résistances, relais, Arduino mega, optotriac transistors, photo-coupleurs etc », avec des pistes en cuivre sur le circuit imprimé. De ce fait, on utilise un CAO (ARES) [14].

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

Cette photo nous montre le circuit imprimé sur une feuille glacée.

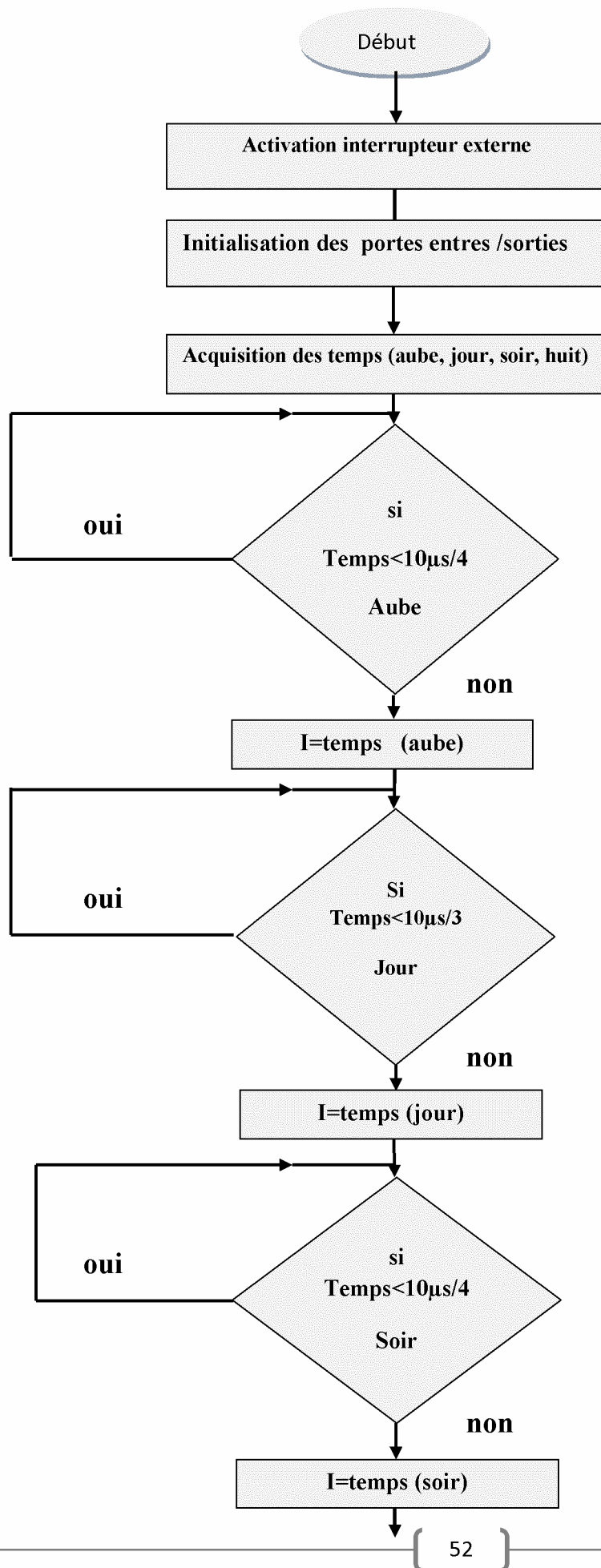


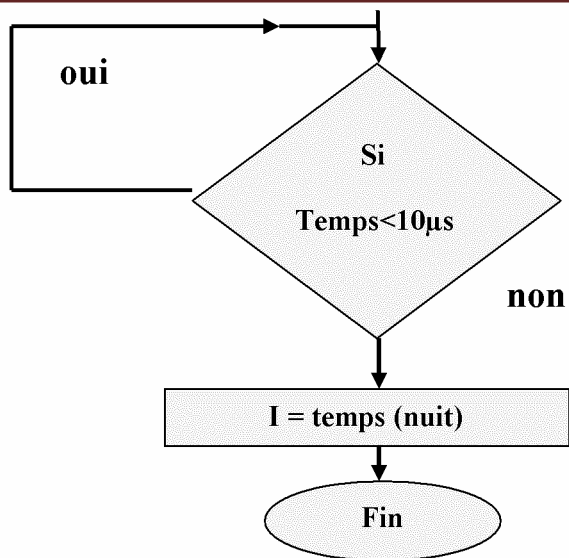
Figure(V.5) : circuit imprimé

4.1. Composants utilisés :

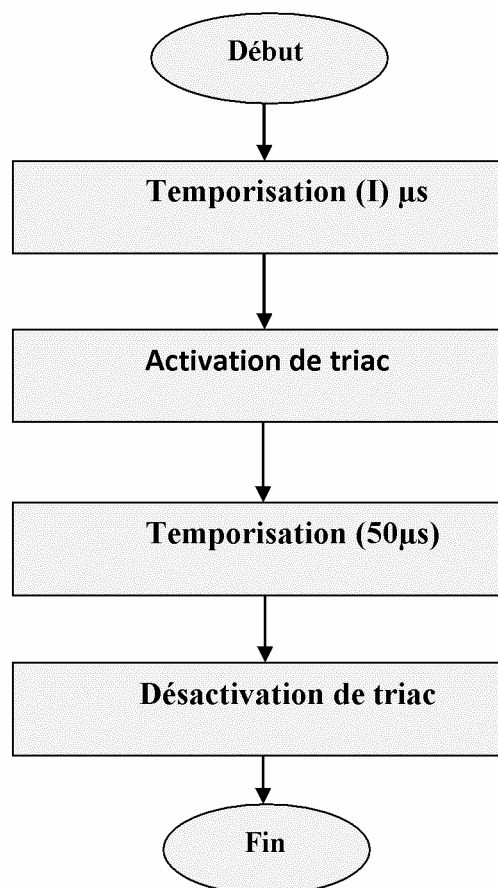
Pour une telle réalisation, nous avons assemblé les différents composants:

- Carte Arduino mega2560
- MOC3020
- MOC3020
- MOC3020
- Optocoupleur-NPN
- TR1 = NPN BC547
- TR2 = NPN BC547
- TRC1 = Triac BT137/500
- TRC2 = Triac BT137/500
- TRC3 = Triac BT137/500
- RL 1 = Relais 12 V min. ci
- RL 2 = Relais 12 V min. ci
- S1 = Inverseur 3 pos.
- RV1 = 10 k Ω pot.lin
- RV2 = 10 k Ω pot.lin
- RV3 = 10 k Ω pot.lin
- RV4 = 10 k Ω pot.lin
- R1, R2 = 1 k Ω
- R3, R4 = 470 Ω
- R5, R6, R7 = 680 Ω
- R8, R9, R10 = 220 Ω
- BR1=Bridge





Figure(V.6) : Organigramme global de système



Figure(V.7) : Fonction de la routine

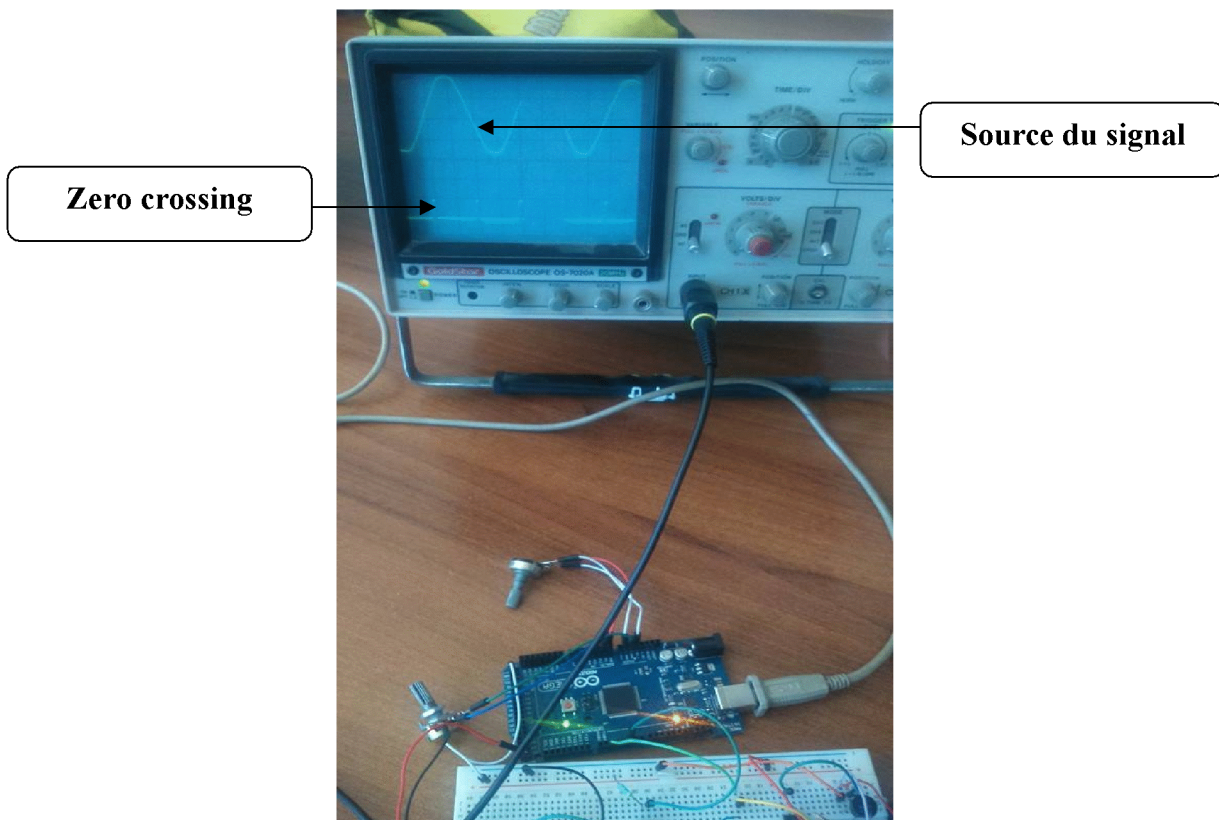
5. Tests :

Pour les tests :

On a réaliser la partie puissance (opto-transistor, optotriac, triac, et la charge) sur un lab d'essai puis on a reliaer la carte Arduino (impulsion de gâchette zéro crossing) ou lab d'essai.

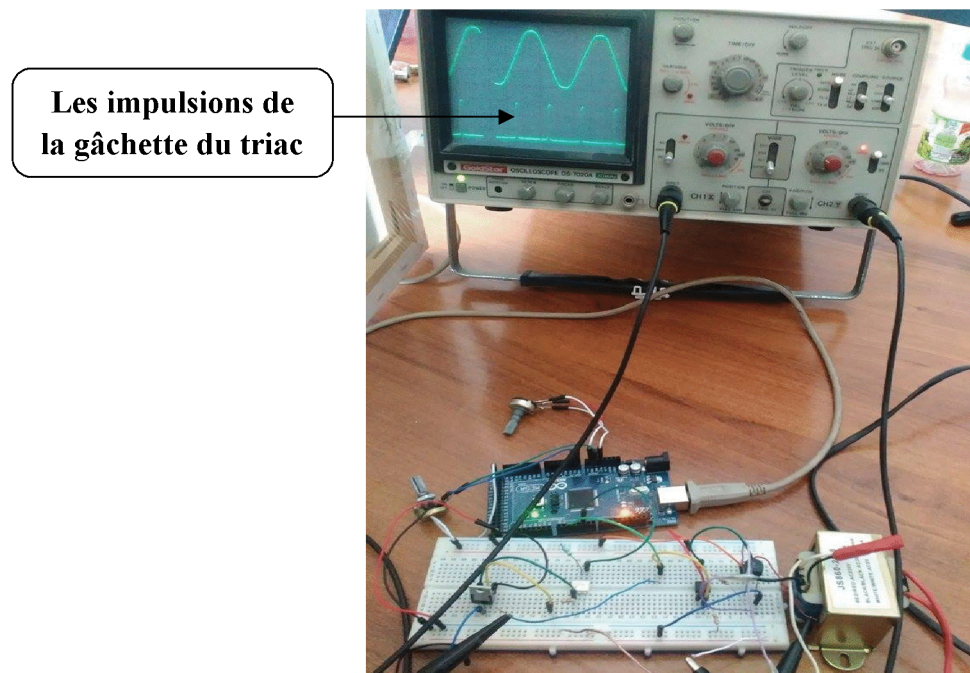
A l'aide d'un oscilloscope à deux canaux on visualise :

- 1- Les zero crossing par rapport à la tension secteur comme le montre la figure ci-dessous :



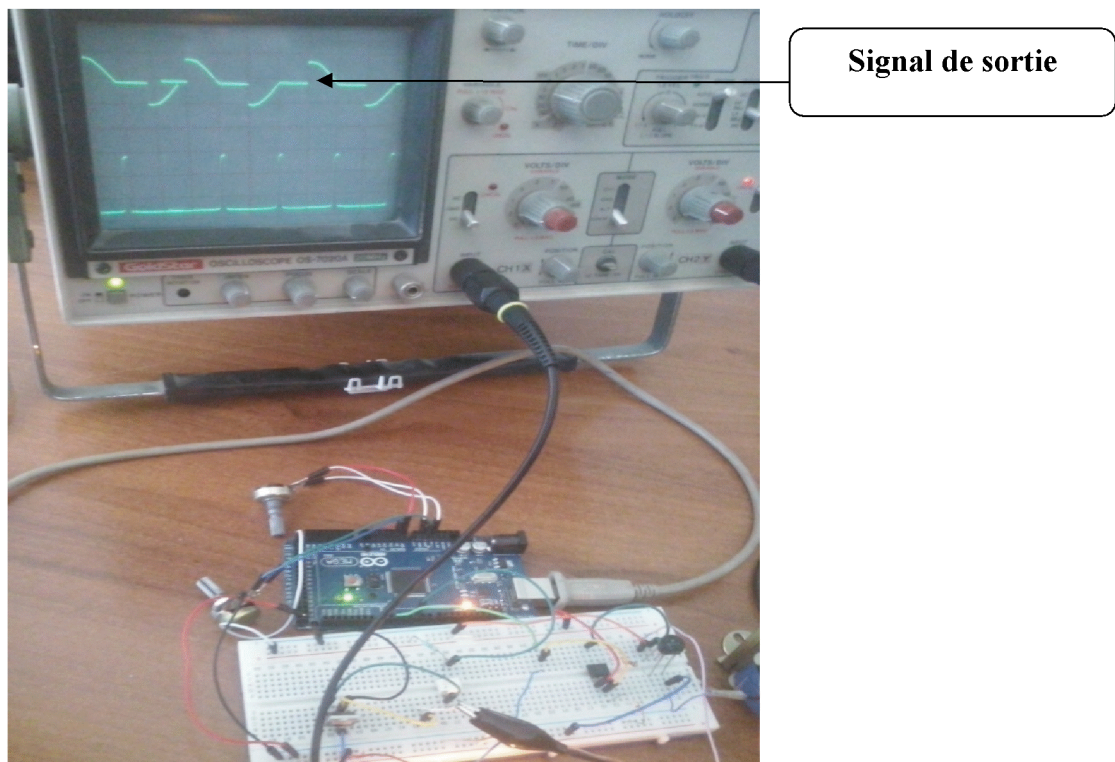
Figure(IV.7) : signal du zero crossing sous oscilloscope

- 2- La forme de signale qui alimente la charge en fonction de déphasage de l'impulsion gâchette fournir par la carte Arduino.

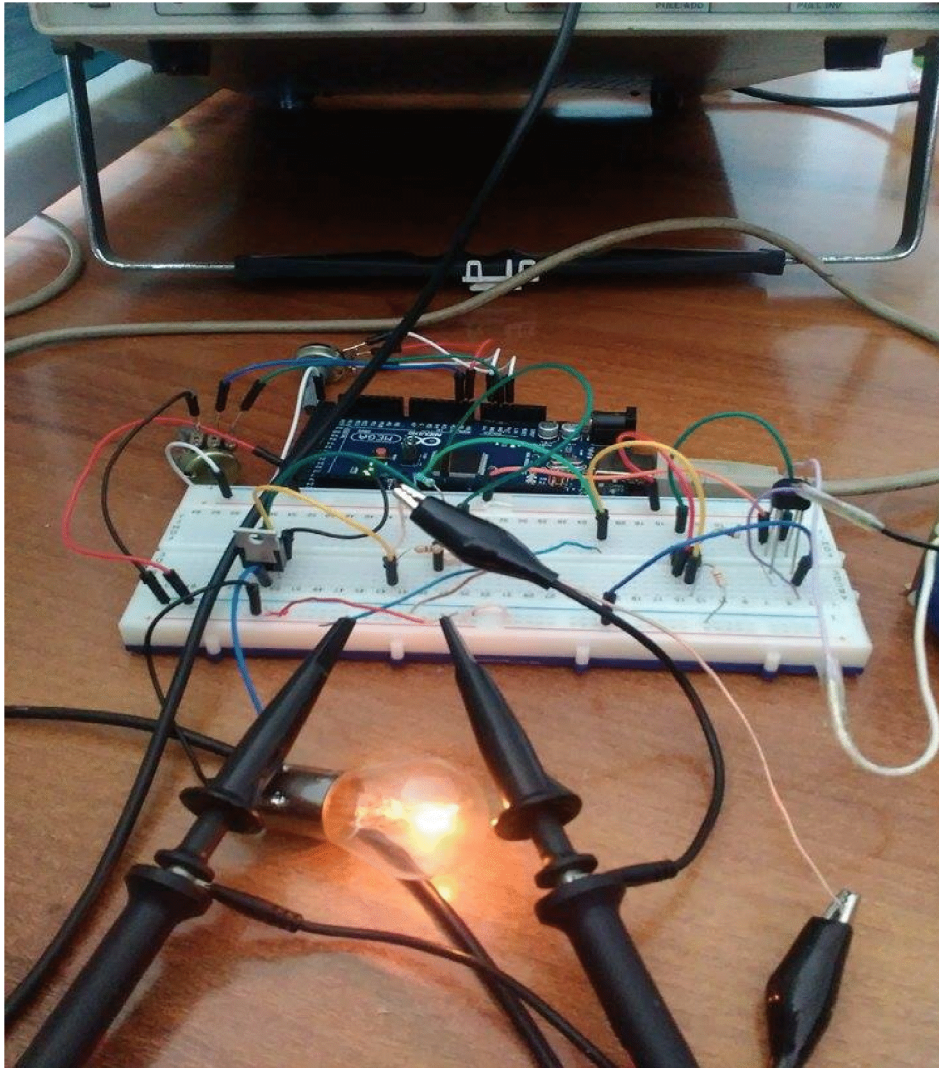


Figure(IV.7) : les impulsions de la gâchette sous oscilloscope

- 3- On fait le réglage des 4 potentiomètres en fonction de la simulation voulue on observe que la densité de la lumière répond à nos besoins.



Figure(IV.8) : les impulsions de la gâchette sous oscilloscope



Figure(IV.9) : photos de la réalisation pratique

Conclusion

Conclusion

Le travail réalisé dans ce présent mémoire a abouti à la mise en œuvre d'un simulateur du cycle solaire. Bien que cette application soit d'une grande utilité dans la vie quotidienne, il présente également l'avantage d'être très facile du point de vue utilisation.

Notons que pendant la réalisation des différents blocs de notre système nous sommes parvenus à assimiler et concrétiser les caractéristiques et fonctionnement de plusieurs dispositifs.

Pour réaliser notre simulateur du cycle solaire, nous avons utilisé une carte ARDUINO dont la fonction est de commandé la partie puissance du circuit qui est un gradateur (variateur) de lumière.

Après les tests, les lampes à filament s'allument très lentement, de manière à simuler l'aube, le jour, le crépuscule et la nuit. Qui allume très lentement une lampe pour simuler l'effet de l'aube et, quand cette lampe a atteint son maximum de luminosité, il la maintient un certain temps pour simuler l'effet du jour, après quoi, toujours lentement, il commence à l'éteindre pour simuler l'effet du crépuscule ; puis, quand la lampe est éteinte, il la maintient dans cet état un certain laps de temps pour simuler la nuit. Plus les fonctions que réalise notre système il peut aussi être utilisé comme simulateur de présence pour évite l'intrusion, ou économise de l'énergie électrique.

Comme perspectives, nous pouvons améliorer notre système en ajoutant des relais dans la partie puissance, et ainsi commande plusieurs systèmes, ou commande la vitesse des moteurs a courant alternatives.

Dans ce travail, un assemblage de plusieurs logiciels est fait afin de réaliser un tel résultat. Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes Confrontés à plusieurs problèmes surtout dans la partie réalisation de la carte. Cependant, on peut dire que malgré ces difficultés, les résultats obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études.

Références bibliographiques

Références bibliographiques

- [1]-<https://www.wikipedia.org/> consulte Mai 2016
- [2] Banzi, Massimo. Arduino : principe de base premiers montage par, Edition Dunod 2013.
- [3]-<http://www.arduino.cc/en/Main/ArduinoBoardMega2560> consulte Mai 2016
- [4] Bartmann, Erik. Le grand livre d'Arduino (texte imprime), EditionEyrolles, Del 2013.
- [5]- Microcontrôleur ATMEL ATMEGA www.reality.be/elo/labos2/files/Atmega2560.pdf
- [6]Etude et réalisation d'un enregistreur de communications téléphoniques, département d'électronique, Université Mouloud Mammeri Tizi-Ouzou promotion 99, Etudié par : Melle : TALEB.H et FERHAT.N.
- [7]-<http://www.siteduzero.com.pdf>
- [8]Etude et réalisation d'un répondeur téléphonique personnalisé, département d'électronique, Université Mouloud Mammeri Tizi-Ouzou promotion 2003. Etudié par : Mr MERDAOUI. K et Melle : DJENNADI.N et MEZIANI.F.
- [9]Etude et réalisation d'un répondeur téléphonique, Institut des techniciens supérieurs, département d'électronique (USTHB) promotion 2000, Etudié par: Mr: MAHANE Ali, et Mr : AFKIR Abdallah.
- [10]- https://www.sonelec-musique.com/electronique_theorie_triac.html
- [11] Pierre MAYE. Connaître les composants électroniques (Tome 3). Edition Fréquences 1993.
- [12] J.Chauveau, G.Chevalier, B.Chevalier.Mémotech électronique : Circuits et composants Edition Casteilla 1997.
- [13]- <http://letsmakerobots.com/node/4750/> consulte juin2016
- [14]- <https://www.labcenter.com/> consulte juin2016

Résumé

Dans le cadre de Project de fin d'étude, nous avons réalisé, un simulateur de cycle solaire. Pour réaliser notre simulateur du cycle solaire, nous avons utilisé une carte ARDUINO dont la fonction est de commandé la partie puissance du circuit qui est un gradateur (variateur) de lumière. Notre montage a été conçu pour allumer très lentement des lampes à filament, de manière à simuler l'aube, le jour, le crépuscule et la nuit,

Mots-clés : simulateur de cycle solaire, ARDUINO, ARDUINO sous proteus, gradateur de lumière, zerocrossing, isolation galvanique, triac.