UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DES **Génie Electrique et d'Informatique**
DEPARTEMENT **D'ELECTRONIQUE**

Mémoire de Fin d'Etude de MASTER PROFESSIONNNEL

Domaine : Sciences et technologies
Filière : Génie électrique
Spécialité : Electronique Industrielle

Thème

La gestion urbaine des feux de circulation de deux carrefours

Réalisé par:

HADRI FARIDA

BOURNNANE TINHINANE

Encadré par :

Mr R.ZIRMI

2eme Promotion
Année: 2017- 2018

Remerciements

Nous remerciant notre Dieux qui nous aide, nous donne la patience et le courage et nous donne le tout puissant, de nous avoir donné la santé et la volonté pour compléter ce modeste travail.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apport leur aide et qui ont contribué à l'élaboration de ce mémoire, ainsi qu'a la réussite de cette formidable année universitaire.

Nous tenant à remercier sincèrement Mr. Zirmi Rachid, en tant que encadreur, qui a toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration nous avoir fait profiter de ses qualités aussi bien sur le plan scientifique que sur le plan professionnel.

Un grand merci pour son sérieux, sa patience et son aide.

Nous remerciements s'adressent également aux membres de jury d'avoir accepter de lire, et d'évaluer notre mémoire.

En fin, nous adressons nos plis sincères remerciements à tous nos proches et amis qui nous ont toujours soutenue et encouragée, au cours de la réalisation de la mémoire.

Merci à tous.

Dédicaces

Je dédie ce modeste travail a :

A mes chers parents, pour tous leurs sacrifices, leur amour, leur soutien et leurs prières tout au long de mes études, grâce a eux que je suis arrivée jusque la aujourd'hui, j'espère qu'un jour je serai capable de leur donner au moins le minimum car quoi qu'il arrive on arrivera jamais à leur rendre tout.

A mes chers frères, samir et boualem et Houssam
A mes sœurs et leurs enfants

A mes chères cousines, nabila, yasmine et sabrina sans elles je n'aurai jamais tenu jusque la,

A mes amis (es), (Lydia, Dalila, Lamia, Katia et mohend ameziane) pour tous les bons moments passés et a venir inchallah.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

HADRI Farida.

Dédicaces

Je dédie ce modeste travail a :

A mes chers parents, pour tous leurs sacrifices, leur amour, leur soutien et leurs prières tout au long de mes études, grâce a eux que je suis arrivée jusque la aujourd'hui, j'espère qu'un jour je serai capable de leur donner au moins le minimum car quoi qu'il arrive on arrivera jamais à leur rendre tout.

A mes chers frères, yaghmourasen et youba ,
A mes chères sœurs,sabrina ,loundja et lamis .
A mes chers amis (bouabbache ouafia,seddour gaia et dahman koceila).

A mon mari djamel et mes chérs beaux parents ,
A mes beaux fréres (mourad,hakim,mouloud,samir,fares et sofiane),
A ma belle sœur Farida
pour tous les bons moments passés et a venir inchallah.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

BOURENANE TINHINANE.

GND : Abréviation pour GrouND, la terre c'est la masse, 0 volt.

POWER : Paissance, alimentation

PIN : Boche

TX : Abréviation pour reçoive, réception.

RX : Abréviation pour transmit, transmission.

MISO (Master in slave out): La ligne pour envoyer des donnée au maitre.

MOSI (Master out slave in) : La ligne principale pour l'envoi de données vers les périphériques.

SS (Slave Select) : la broche sur chaque périphérique que le maitre peut utiliser pour active et désactive des dispositifs spécifiques.

SDA (serial datation): ligne de données bidirectionnelle.

SCL (Serial Clock Line): ligne d'horloge de synchronisation bidirectionnelle.

12.C : Inter-Integrated Circuit.

ADC: Analog-To-Digital Converter : un convertisseur analogique numérique.

SCK (horloge de série): Les implusions d'horloge qui synchronisent transmission des données généré par le maitre.

PWM : Abréviation de (Pulse Width Modulation), Modulation de la largeur d'Implusion.

V (1, 3) : Feu vert de la voie 1 et 3.

O (1, 3) : Feu orange de la voie 1 et 3.

R (1, 3) : Feu rouge de la voie 1 et 3.

OTD (1, 3) : Fou orange de tourner a gauche de la voie 1 et 3.

RP (1, 3) : Feu rouge des piétons de la voie 1 et 3.

VP (1, 3): Feu vert des piétons de la voie 1 et 3.

OTG (1,3): Feu orange de tourner a gauche de la voie 1 et 3.

RTG (1,3): Feu rouge de tourner a gauche de la voie 1 et 3.

V (2,4) : Feu vert de la voie 2 et 4.

O (2, 4) : Feu orange de la voie 2 et 4.

R (2,4) : Feu rouge de la voie 2 et 4.

OTD (2,4) : Feu orange de tourner a gauche de la voie 2 et 4.

RP (2,4) : Feu rouge des piétons de la voie 2 et 4.

VP (2,4) : Feu vert des piétons de la voie 2 et 4

OTG (2, 4) : Feu orange de tourner a gauche de la voie 2 et 4.

RTG (2,4) : Feu rouge de tourner a gauche de la voie 2 et 4.

V (5, 7) : Feu vert de la voie 5 et 7.

O (5, 7) : Feu orange de la voie 5 et 7.

R (5, 7) : Feu rouge de la voie 5et 7.

OTD (5, 7) : Fou orange de tourner a gauche de la voie 5 et 7.

RP (5, 7) : Feu rouge des piétons de la voie 5 et 7.

VP (5, 7): Feu vert des piétons de la voie 5 et 7.

OTG (5, 7): Feu orange de tourner a gauche de la voie 5 et 7.

RTG (5, 7): Feu rouge de tourner a gauche de la voie 5 et 7.

V (6, 8) : Feu vert de la voie 6 et 8.

O (6, 8) : Feu orange de la voie 6 et 8.

R (6, 8) : Feu rouge de la voie 6 et 8.

OTD (6, 8) : Feu orange de tourner a gauche de la voie 6 et 8.

RP (6, 8) : Feu rouge des piétons de la voie 6 et 8.

VP (6, 8) : Feu vert des piétons de la voie 6 et 8

OTG (6, 8) : Feu orange de tourner a gauche de la voie 6 et 8.

RTG (6, 8) : Feu rouge de tourner a gauche de la voie 6 et 8.

Liste des figures

Puisque le débit est une variable continue qui nécessite une période de temps donnée pour être estimée, il y a toujours des écarts importants entre le débit estimé et le débit réel. Cela rend difficile l'exploitation du potentiel des infrastructures de la circulation à un niveau maximum.

Toutefois, et afin de mettre en œuvre cette méthode classique, nous devons faire face à deux grandes difficultés. Tout d'abord, comment échanger des informations entre les véhicules ou entre les véhicules et les infrastructures routières. Ensuite, comment trouver une séquence de passage des véhicules efficace de sorte à maximiser le débit de circulation aux intersections, tout en maintenant en même temps la sécurité des différents usagers.

Ce projet se décompose comme suit : dans l'immédiat, nous introduisons quelques généralités Sur les carrefours et les feux de signalisation ; dans la partie suivante, nous parlons sur la carte Arduino sous un aspect général, en faisant un rapide tour sur son principe de fonctionnement et d'utilisation ; dans la troisième partie qui est la conception et la réalisation du projet, nous présentons des outils permettant d'expliquer la gestion que nous avons utilisé.

Enfin, nous terminerons par une conclusion générale qui résumera notre technique de gestion pour pouvoir modéliser le trafic routier.

1. Introduction :

L'espace urbain est de plus en plus considéré comme un espace à partager entre les usagers de différents modes de transports : la voiture individuelle, les transports publics, les piétons…etc.

Actuellement, la forme des villes ainsi que le nombre d'habitants qu'elles abritent posent un problème de congestion qu'on rencontre quotidiennement dans les milieux urbains et surtout dans les carrefours.

La régulation du trafic urbain nécessite un système ou un moyen capable d'exécuter des actions entre les véhicules et les piétons en assurant leurs sécurités.

2. Les carrefours urbains :

Figure1: Carrefour à quatre branches avec des voies différentes**[1]**

Un carrefour est un nœud de communication entre les véhicules et les piétons, il est situé à la rencontre de plusieurs rues, déterminant des couloirs d'entré et de sortie. Un couloir est caractérisé par sa largeur et le nombre de ses voies ; certaines de ces voies (voies spéciales) peuvent être affectées par un flux particulier. Les courants des véhicules sont soit des courants directs, soit des courants de tourne à gauche, soit des courants de tourne à droite, figure2.1

3. Aspect réglementaire et normatif :[2]

L'espace urbain et la route en général sont très réglementés : signalisation, prise en compte des priorités, Code de la Route, …

Lors du franchissement d'une intersection, l'automobiliste doit :

- Ralentir en contrôlant son rétroviseur intérieur.
- S’assurer que la route est libre.
- Observer les autres automobilistes afin de déterminer leurs intentions.
- Respecter les règles de priorités de passage en fonction des panneaux, feux ou agents.

4. La régulation du carrefour:

La régulation du trafic routier est un outil essentiel permettant de mettre en œuvre une politique d’organisation de déplacements en milieu urbain en basant sur un système ou un moyen capable d’exécuter des actions entre les différents usagers tout en gardant la sécurité des véhicules et les piétons sur la voirie.

4.1. La régulation du carrefour par les feux de circulation :

Parmi les divers moyens de la régulation du trafic urbain, les feux de circulation jouent un rôle très important dans la gestion automatique des carrefours ; ils permettent d’abord d’assurer la sécurité en partageant dans le temps l’utilisation d’un même espace entre les flux conflictuels. Mais par le choix des durées de chaque couleur (vert, rouge) et par la synchronisation des feux entre eux.

4.2. Différentes catégories de signaux lumineux de circulation : [3]

Les feux de circulation sont verts, jaunes ou rouges, sauf ceux spécifiquement réservés auxvéhicules des services réguliers de transport en commun, qui sont blancs. Ils peuvent être groupésen signaux tricolores, bicolores ou unicolores. Ils sont généralement circulaires et, pour les feuxdestinés aux véhicules des services réguliers de transport en commun, peuvent comporter unpictogramme ou des signes spécifiques.

Les feux jaunes, rouges et le disque des feux pour véhicules des services réguliers de transport encommun peuvent être clignotants (c'est-à-dire alternativement allumés ou éteints chaque seconde,pendant des durées sensiblement égales). Les signaux lumineux d'intersection comprennent neuf grands types de signaux, R11 à R19.

R11 : Signal tricolore circulaire :

Il est normalement composé de trois feux circulaires vert, jaune, rouge (R11v) :voir figure 1. Exceptionnellement, et sous réserve d'une étude le justifiant, le vertpeut être remplacé par du jaune clignotant (R11j).

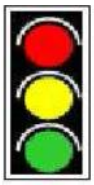


Figure 2 R11v

R12 : Signal piéton :

Il est constitué de deux feux vert et rouge, normalement disposés dans cet ordre de droite à gauche ; éventuellement ils peuvent être disposés l'un au-dessus de l'autre, le vert en bas.

Figure 3 R12

R13 : Signaux tricolores modaux :

Ils sont composés de trois feux vert, jaune, rouge, dans cet ordre de bas en haut, munis chacun d'un même pictogramme. Le feu vert peut être remplacé par un feu jaune clignotant, les signaux se dénommant alors respectivement : R13cj et R13bj.

En forme de cycle En mention Bus

Figure 4 R13c R13b

R14 : Signaux tricolores directionnels :

Ils sont destinés chacun à l'ensemble des véhicules qui ont pour destination la direction indiquée par la flèche, ou l'une des directions indiquées. En aucun cas le feu vert ne peut être remplacé par un feu jaune clignotant.

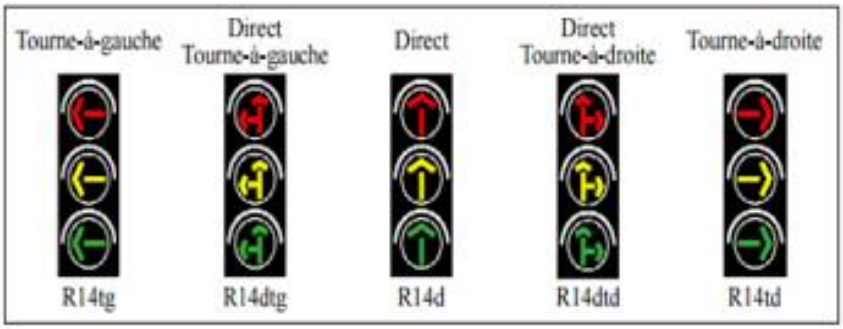


Figure 5

R15 : Signaux d'anticipation modaux :

En forme de cycle Avec mention BUS Ils sont composés d'un feu jaune clignotant et sont obligatoirement associés à un ensemble de feux tricolores circulaires du type R11v (vert sur le feu du bas). Ils sont munis d'un pictogramme.

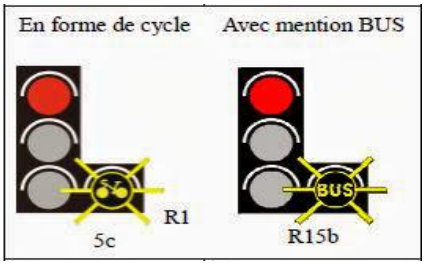


Figure 6

R16 : Signaux d'anticipation directionnels :

Ils sont composés d'un feu jaune clignotant et sont obligatoirement associés à un ensemble de feux tricolores circulaires R11v (vert sur le feu du bas). II est recommandé de les associer aux signaux R11 comme indiqué figure 6. Ils sont munis d'un pictogramme en forme d'une ou deux flèches :

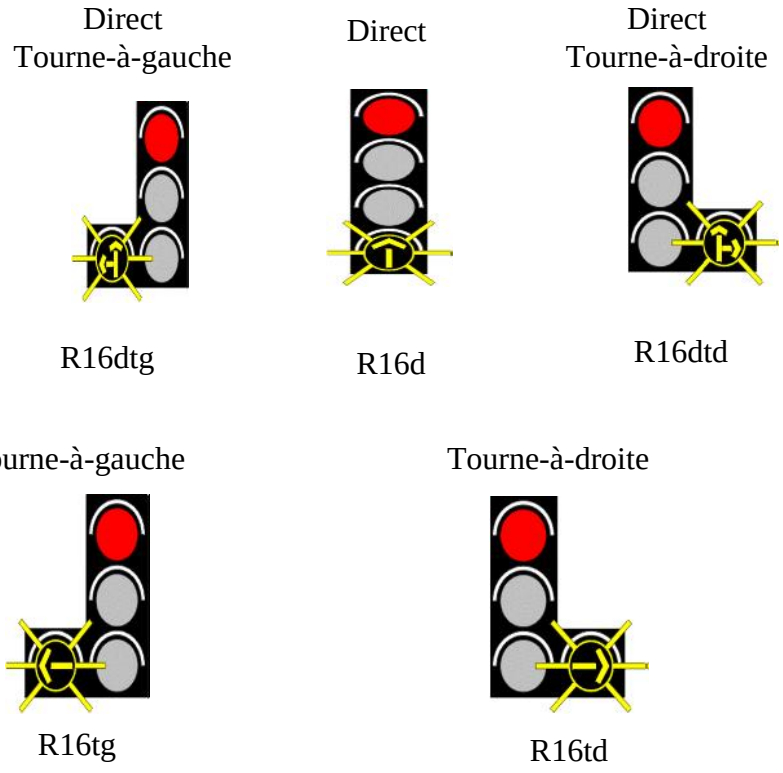


Figure 7

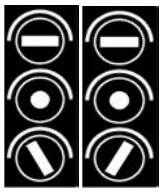
R17 : Signal pour véhicules des services réguliers de transport en commun :

Il est composé de trois feux blancs présentant. De bas en haut, une barre verticale, un disque et une barre horizontale, sur fond noir circulaire. Le feu central comportant le disque peut être clignotant.

Figure.7 R17

R18 : Signaux directionnels pour véhicules des services réguliers de transport en commun :

Ils sont composés comme le signal R17, à l'exception de la barre du feu inférieur qui est inclinée à gauche ou à droite. Ils s'adressent exclusivement aux véhicules des services réguliers de transport en commun qui ont pour destination la direction indiquée par la barre du feu inférieur.

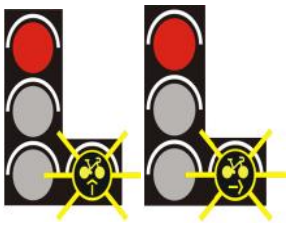


R18tg R18td

Figure.7b

R19 : Signaux d'autorisation conditionnelle de franchissement pour cycles :

Ils sont composés d'un feu jaune clignotant munis de deux pictogrammes et sont obligatoirement associés à un ensemble de feux tricolores circulaires dont le feu du bas est vert. Ils autorisent les cycles à ne pas marquer l'arrêt au feu pour s'engager dans la direction indiquée.



R19td R19d

Figure.7c

4.3. Paramètres nécessaires au calcul de la durée des feux : [4]

Ces paramètres sont très utiles pour la mise en œuvre des stratégies de commande. Leur évolution est considérée comme un facteur déterminant pour l'efficacité de telles stratégies (Tolba, 2004). Cependant, quelle que soit la méthode de commande appliquée à l'intersection, les indications des feux (vert, orange et rouge) doivent se succéder à l'intérieur d'un cycle défini comme étant la durée fixe ou variable, séparant deux passages successifs de l'ensemble des signaux par le même état (Cohen, 1990).

Définitions :

- *Vert effectif* : le temps de vert effectif est la somme du temps de vert réel et du temps de l'orange qui est souvent de 3 ou 5 secondes selon la vitesse d'approche dans le cas d'un carrefour ordinaire ;
- *Vert utile* : le temps de vert utile est la différence entre le temps de vert effectif et celui du temps perdu ;
- *Temps perdu* : la somme du temps perdu au début du vert et celui en fin de phase ;
- *Rouge utile* : obtenu en retranchant, de la durée du cycle, le temps du vert utile ;

- *Rouge intégral* : la durée pendant laquelle aucun véhicule n'est admis dans le carrefour, c'est-à-dire lorsque l'état est rouge sur chaque feu.

4.4. La condition d'installation des feux : [5]

L'installation d'un signal lumineux de circulation dans un carrefour doit être implanté et orienté pour être vu des usagers auxquels il est destiné et, dans la mesure du possible, ne pas être vu des usagers auxquels il n'est pas destiné. La face arrière des signaux doit être occultée pour ne laisser passer aucune lumière.

5. Conclusion

La régulation du trafic est un domaine vaste dans lequel plusieurs techniques et formes de signalisation sont utilisés pour faciliter la circulation des véhicules, augmenter la sécurité des usagers, réduire toute sorte de nuisances et rationaliser l'exploitation des infrastructures routières. Parmi les sujets les plus importants de la régulation du trafic, nous trouvons la gestion des carrefours à feux. Ces derniers constituent un terrain sensible pour les conflits de trafic et représentent un espace de convergence de différents acteurs du transport.

1. Introduction

La carte Arduino repose sur un circuit intégré (un mini-ordinateur appelé également microcontrôleur) associé à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes. Il existe un grand nombre de variantes**.**

II peut être utilisé pour effectuer des taches très diverses comme la charge de batterie, la domotique, le pilotage d'un moteur électrique…etc

L'Arduino est une carte électronique en Matériel Libre pour la création artistique interactive. Elle peut servir (7):

1/ pour des dispositifs interactifs *autonomes* simples.

2/ comme *interface* entre capteurs/actionneurs et ordinateur.

3/ comme programmateur de certains microcontrôleurs.

2. Les Caractéristiques techniques de la carte Arduino :

Les cartes arduino ont les mêmes caractéristiques c'est justes les valeurs qui changent, et ces caractéristiques sont :

1. Microcontrôleur (ATmelmega).
2. Fréquence d'horloge.
3. Tension de service.
4. Tension d'entrée (recommandés).
5. Tension d'entrée (limites).
6. Ports numériques.
7. Ports analogiques.
8. Courant maxi.par broche d'E/S.
9. Courant maxi.par broche.
10. Mémoires (flash, SRAM, EEROM).
11. Chargeur d'amorçage.
12. Interface(USB).
13. Dimension.
14. Prix.

3. Les gammes de la carte Arduino

De nos jours, il existe plus de 20 versions de module Arduino, nous citons quelques-uns fin de montrer l'évolution de ces cartes :

Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'unATmega8.

- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleurATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour Programmer en utilisant un Microcontrôleur ATmega168.
- L'ArduinoMega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'ArduinoUNO, utilisations Microcontrôleur ATmega328.
- L'Arduino Leonardo, avec un morceau ATmega3Ù4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- L'Arduino Mega2560, utilisations un Microcontrôleur urATmega2560, et possèdetoute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2.

4. Pourquoi Arduino Mega2560 :

Il existe plusieurs modèles fabriqués par : ATMEL. Le choix dépend de plusieurs critèresde sélection dont le développeur doit tenir compte du :

- Type du microcontrôleur ;
- Nombre d'entrées/sorties ;
- Liaison d'entrées/sorties ;
- Conversion analogique numérique et numérique analogique ;
- Mémoire RAM, ROM, EPROM interne ou externe, sa taille ;
- Vitesse d'horloge, temps d'exécution d'une multiplication, d'une division ;
- Bus de données 8bits /16bits ;
- Les logiciels de programmation (assembleur, c, micro,…) ;
- Les émulateurs pour la mise au point des applications.
- Il existe des schields spéciaux pour le prototypage qui en raison de leur surface supérieure, peuvent recevoir plus de composant.

4.1. Les caractéristiques de Mega 2560 :

Catégorie	Valeur
Microcontrôleur	ATMega2560
Fréquence d'horloge	16 MHZ
Tension de service	5V
Tension d'entrée (recommandée)	7_12V
Tension d'entrée (limités)	6_20V
Ports numériques	54 entrées et sorties (15commutables en MLI)
Ports analogiques	16 entrées analogiques
Courant maxi. par broche d'E/S(cc)	40mA
Courant maxi. par broche 3,3V	50mA
Mémoires	256Ko flash/ 8Ko SRAM/ 4Ko EEPROM
Chargeur d'amorçage	8Ko(en mémoire flash)
Interface	USB
Dimensions	10,16cm*5,3cm

Tableau 1 : Caractéristique de la carte Arduino Mega

4.2. Le Microcontrôleur ATMega2560 :

Le Atmel ATMEGA2560-16AU est un microcontrôleur 8 bits CMOS basse puissance basée sur architecture RISC améliorée des AVR. En exécutant des instructions puissantes en un seul cycle d'horloge, le ATMEGA2560-16AU atteint des débits approchant les 1MIPS par MHz permettant aux concepteurs de système d'optimiser la consommation d'énergie par rapport à la vitesse de traitement(7).

Le microcontrôleur de cette carte est un objet capable de traiter, de stocker et de restituer de l'information. Il est en particulier constitué d'un microprocesseur. C'est la partie centrale qui permet le traitement de l'information : il comprend des milliers voir des millions de transistors. Ils réalisent les fonctions logiques suivantes :

- Calcules arithmétiques et logiques.
- Mémorisation.
- Interface de communication.

Les Atmel sont des composants dits RISC (**R**educed**I**nstructions **S**et **C**omputer), ou encore (composants à jeux d'instructions réduit). Ces microcontrôleurs utilisent l'architecture de Harvard : c'est-à-dire le programme et les données sont stockées dans des mémoires physiquement séparées. (Deux bus de données. Un bus est utilisé pour les données et un autre pour les instructions.

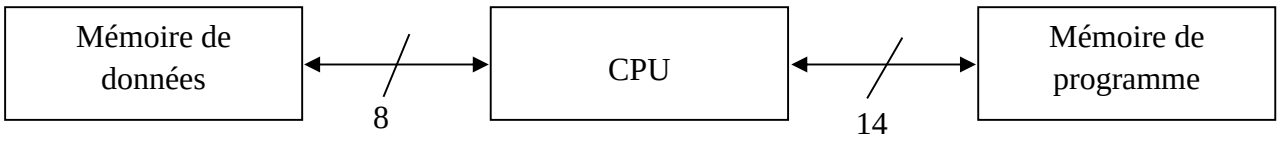


Figure 8 : Architecture e harvard pour les Atmel

La figure 2 montre un Microcontrôleur de ATmelMega 2560 qu'on trouve sur la carte Arduino :

Figure 9 : Microcontrôleur Atmel Mega2560

- Architecture RISC avancée ;
- 135 Instructions puissantes, la plupart, avec une exécution en un seul cycle d'horloge ;
- Registres 32 × 8 à usage général ;
- Fonctionnement entièrement statique ;
- Jusqu'à 16 MIPS à 16 MHz ;
- Multiplicateur en 2 cycles sur la puce ;
- Segments de mémoire non-volatile à haute endurance ;
- 256 Ko de Flash dans le système auto programmable ;
- EEPROM 4Ko ;
- SRAM 8Ko Interne ;
- Flash EEPROM 10 000 cycles de Lecture/Ecriture ;
- Conservation des données : 20 ans à 85°C / 100 ans à 25°C ;
- Section de code de démarrage en option avec bits de verrouillage indépendants ;
- Programmation IN SITU par le programme de démarrage sur puce ;

- Opération de lecture écriture réelle ;
- Verrou de programmation pour la sécurité du logiciel ;
- Endurance : Jusqu'à 64 Koctets d'espace de mémoire externe en option ;
- Supporte la librairie Atmel® QTouch® ;
- Boutons tactiles capacitifs, curseurs et molettes ;
- Acquisition QTouch et QMatrix®(7).

5. La description générale de l'Arduino Mega :

La photo présente la carte Arduino Mega utilisé :

.

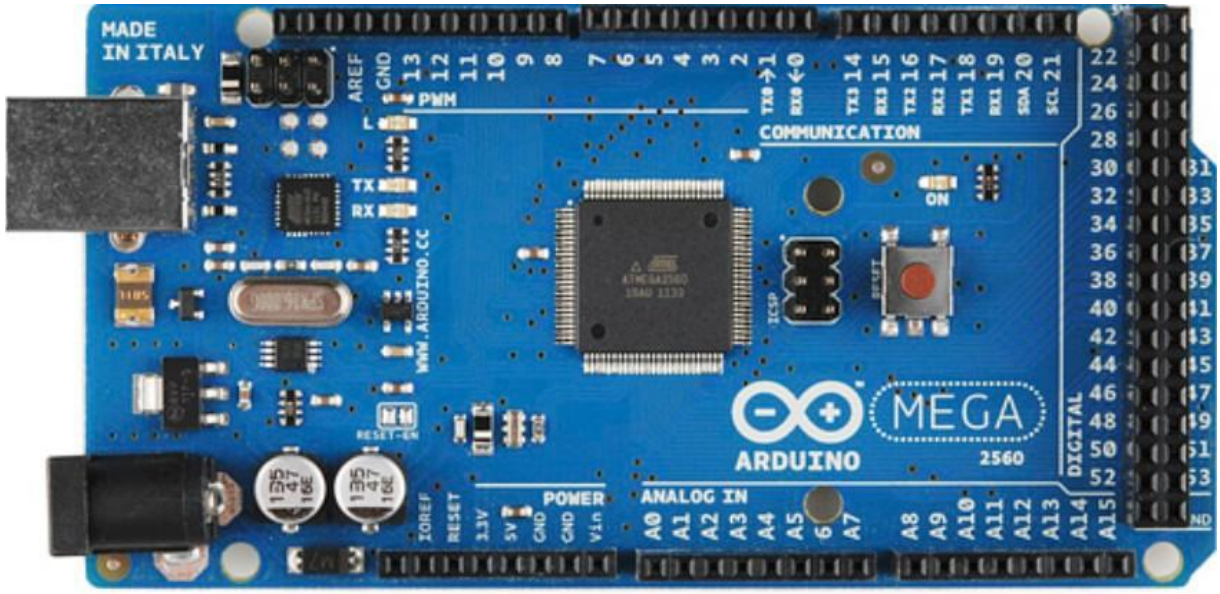


Figure 10 : La carte Arduino Mega 2560

5.1. Les mémoires de microcontrôleur ATMega 2560 :

Le microcontrôleur ATMega2560 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en effet constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

1. **La mémoire Flash** : C'est celle qui contiendra le programme à exécuter. Cette mémoireest effaçable et réinscriptible mémoire programme de 256 KB (dont boot loader de 8 KB) ;
2. **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elleest dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur

3. **EEPROM** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.
4. **Le registre** : c'est un type de mémoire utilisé par le processeur ;
5. **La mémoire Cache** : c'est une mémoire qui fait la liaison entre les registres et la RAM(8).

5.2. Les sources d'alimentation de la carte :

1. Alimentation :

La carte ArduinoMega 2560 peut être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adapteur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte.

La carte Arduino Mega2560 diffère de toutes les cartes précédentes car elle n'utilise par le circuit intégré FTDI usb-vers-série. A la place, elle utilise un Atmega8U2 programmé en convertisseur USB-vers-série.

2. Les broches d'alimentation sont les suivantes :

- **GND** : La carte arduino Mega2560 se compose de cinq ports de **GND** ;
- **POWER (Puissance)** : la carte Arduino Mega2560 se compose de quatre portes de **POWER**, et sont comme suit :
  - ✓ **IOREF :** Cette broche sur la carte fournit la référence de tension avec laquelle le microcontrôleur fonctionne. Un écran correctement configuré peut lire la tension de la

broche **IOREF** et sélectionner la source d'alimentation appropriée ou activer des traducteurs de tension sur les sorties pour travailler avec le **5V**ou **3,3V** ;

- **5V :** La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le **5V** régulé fourni par cette broche peut donc provenir soit dela tension d'alimentation **VIN** via le régulateur de la carte, ou bien de la connexion USB (qui fournit du **5V** régulé) ou de tout autre source d'alimentation régulée ;
- **3V3 :** Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA.
- **VIN :** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche (9).
- **RESET (Réinitialiser) :** Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte. (9)

3. Les broches d'E/S numériques :

Chacune des 54 broches numériques (numérotée de 0 à53) sur le Mega2560 peut être utilisé comme une entrée ou une sortie, en utilisant pinMode(), digitalWrite(), et digitalRead() fonctions. Ils fonctionnent à **5volts**. Chaque broche peut fournir ou recevoir 20 mA en état de fonctionnement recommandée et a une résistance pull-up interne (déconnectée par défaut) de **20-50 kohm**. Un maximum de **40mA** est la valeur qui ne doit pas être dépassée pour éviter des dommages permanents au microcontrôleur. (8)

En outre, certaines broches ont des fonctions spécialisées.

5.3. Ports de communication

- **Communication série** : Port Série Serial : 0 (RX) et 1 (TX) ; Port Série Serial 1: 19(RX) et 18 (TX) ; Port Série Serial 2: 17 (RX) et 16 (TX) ; Port Série Serial 3: 15(RX) et 14 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Les broches 0 (RX) et 1 (TX) sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur(8).

  Ces broches peuvent être connectées avec des accessoires d'exploitation d'un réseau sans

  Ces broches peuvent être connectées avec des accessoires d'exploitation d'un réseau sans filet ne sont pas visibles comme : wifi, GPS, Bluetooth…

- **Serial Périphérie Interface** (**SPI**) : est un protocole de données série synchrone utilisées par les microcontrôleurs pour communiquer avec un ou plusieurs périphériques rapidement sur de courtes distances. Il peut également être utilisé pour la communication entre les deux microcontrôleurs.

  Avec une connexion **SPI**, il y a toujours un dispositif maître (généralement un microcontrôleur) qui contrôle les périphériques. En règle générale, il y a trois lignes communes à tous les appareils :

  - ✓ **MISO** (port 50) - La ligne esclave pour envoyer des données au maître.
  - ✓ **MOSI** (port 51) - La ligne principale pour l'envoi de données vers les périphériques.

  Et une ligne spécifique pour chaque appareil :

  - ✓ **SS** (port 53) - la broche sur chaque périphérique que le maître peut utiliser pour active et désactiver des dispositifs spécifiques.
  - ✓ **SCK** (port 52) - Les impulsions d'horloge qui synchronise la transmission de donné générées par le maître.

- **12/C** / TWI broches :
  - ✓ (**SDA**). Est sur la broche numérique 20.
  - ✓ (**SCL**). Est sur la broche numérique 21.

Ces broches peuvent être connectées avec des accessoires d'exploitation d'un réseau sans fil FM.

5.4. Les fonctions IDE de l'environnement Arduino

- **Fonctions d'initialisation :**
  - ✓ Begin() : initialise communication avec Arduino "maître"
  - ✓ Begin(adresse) : initialise communication avec Arduino "esclave".

- **Fonctions mode maitre :**
  - ✓ RequestFrom(adresse, quantite) : demande de données à un esclave
  - ✓ BeginTransmission(adresse) : débute communication avec un esclave (ouvre stockage données à envoyer avec write)
  - ✓ EndTransmission() : envoie des données vers esclave
  - ✓ Write() : écrit les données à envoyer vers esclave
  - ✓ Available() : teste ces données disponibles en provenance esclave (cfrequestFrom)
  - ✓ Read() : lit les données en provenance de l'esclave.
- **Fonctions mode esclave :**
  - ✓ Write() : envoie les données vers le maître après requête
  - ✓ Available() : test ces données disponibles en provenance du maître (cfonReceive)
  - ✓ Read() : lit données en provenance maître
  - ✓ OnReceive(fonction) : définit la fonction à appeler sur réception de données en provenance du maître
  - ✓ OnRequest(fonction) : définit la fonction à appeler sur requête du maître.
- **Fonctions obsolétes :**
  - ✓ Send() : obsolète
  - ✓ Receive() : obsolète. (11)
- **PWM :** 2 à 13 et 44 à 46. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite(). (8)
- **PCIN (PinChangeInterrupt**)

Le tableau suivant présente les broches PCINT utilisables sur l'Arduino Mega2560 :

Broches PCINT	PINsAruino
PCINT10	19
PCINT1	20
PCINT2	21
PCINT3	22
PCINT4	10
PCINT5	11
PCINT6	12
PCINT7	13
PCINT8	6
PCINT9	15
PCINT10	14
PCINT16	A8
PCINT17	A9
PCINT18	A10
PCINT19	A11
PCINT20	A12
PCINT21	A13
PCINT22	A14
PCINT23	A15

Tableau 2 : Les broches PCINT

Broches d'entrée analogiques : La carte Mega2560 dispose de 16 entrées analogiques (numérotées de 0 à 15), chacune pouvant fournir une mesure d'une résolution de 10 bits (c.-à-d. Sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead() du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction analogReference() du langage Arduino. (7)

ADC : Un signal analogique est celui qui peut prendre un certain nombre de valeurs, contrairement à un signal numérique qui n'a que deux valeurs : HAUT et BAS. Pour mesurer la valeur des signaux analogiques, l'Arduino a intégré analogique-numérique (ADC). L'ADC transforme la tension analogique en une valeur numérique. La fonction qu'on peut utiliser pour obtenir la valeur d'un signal analogique est analogRead (broche). Cette fonction convertit la valeur

de la tension sur une broche d'entrée analogique et renvoie une valeur numérique de 0 à 1023, par rapport à la valeur de référence. La référence est 15V sur Arduino Mega2560. Il a un paramètre qui est le nombre de broches Il y a quelques autres broches analogiques de la carte : AREF. Tension de référence pour les entrées analogiques (si différent du 5V). Utilisé avec l'instruction analogReference().

6. Fonctionnement et utilisation

L'utilisation des cartes Arduino est très simple : connecter la carte à l'ordinateur (via le câble USB, en général), lancer « Arduino IDE » (c'est le logiciel qui permet de programmer la carte), mettre en place les composants, les relier à la carte, coder le programme, le charger sur la carte grâce au bouton téléviser et le programme s'exécutera ensuite en boucle.

6. Conclusion

Les cartes Arduino donnent vraiment un potentiel de création quasi infini pourvu que l'on dispose d'un matériel approprié. Il est possible de réaliser des dispositifs, de gérer des caméras, de commander des moteurs, d'alimenter automatiquement un plan au bout d'un lapsde temps. L'autre intérêt est de faire de cet investissement un grand avantage dans l'application de contrôle et de commandes moderne en domotique.

Il est clair que l'Arduino permet d'assurer les principaux protocoles d'interfaçage : communication bidirectionnel, acquisition (conversion) et traitement des différents types des signaux avec une bonne précision et rapidité avec une mémoire de programme de donnée importantes.

1. Fonctionnement du système

Afin de bien comprendre le principe de fonctionnement d'un carrefour, il faut faire une étude et une analyse générales sur la circulation des différents usagers dans les carrefours à développer, pour pouvoir faire une bonne gestion qui pourra résoudre les différents problèmes de congestion. Cette dernière se présente sous diverses méthodes différentes : graphe d'état (Rdp), grafcet, le phasage…etc.

Dans notre cas, nous avons fais une gestion avec la méthode du découpage en phase en utilisant les trois Catégories (R11, R12 et R14) de signaux tricolores dont chaque carrefour constitue :

- Une intersection de quatre routes principales, avec deux choix de direction : le premier est formé avec des mouvements d'aller tout droit et de tourner à droite, le deuxième est celui des mouvements de tourner à gauche.
- Quatre entrées du carrefour(voies E1,E2,E3,E4/E5,E6,E7,E8).
- Quatre sorties du carrefour (voies S1, S2, S3, S4/S5, S6, S7, S8).
- Et quatre passages piétons (p1, p2, p3, p4/p5, p6, p7, p8).

Nous avons réalisé un système qui permet de gérer deux carrefours voisins avec une carte électronique (ARDUINO).

Ce système fonctionne en deux modes ; le premier est le mode jour, le second est le mode nuit. Dans le premier mode les feux de signalisation fonctionnent automatiquement en quatre phases ordonnées (1,2,3 apres 4), quand la première phase est activée dans le premier carrefour,la deuxième sera activée dans le deuxième carrefour et ainsi de suite. Le deuxième mode consiste à fonctionner juste les feux oranges pour informer les usagers qu'il y a un carrefour et pour qu'ils fassent attention.

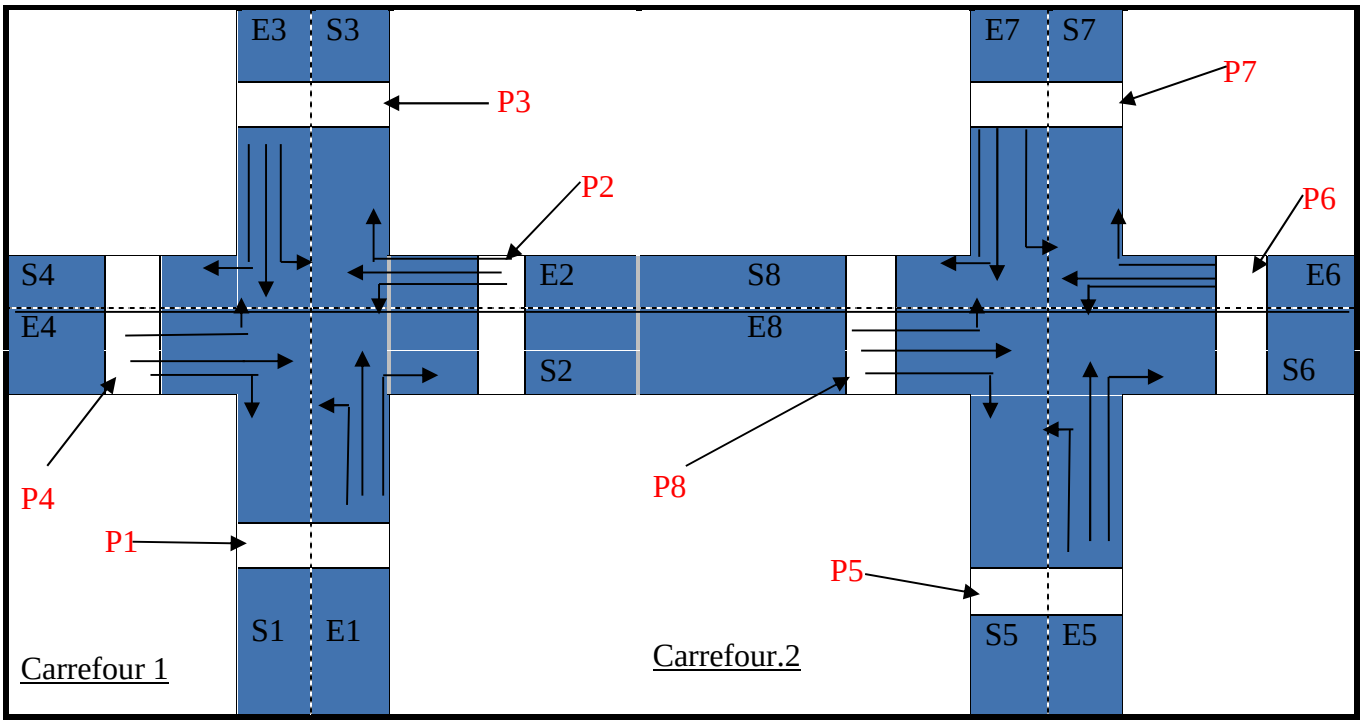


Figure 11 :Les carrefours étudiés

2. Le découpage en phases : (6)

Le découpage en phases, ou phasage est une méthode qui exprime toute une gestion dans un carrefour. Cette méthode consiste à se décomposer en plusieurs phases dont chaque phase exprime une étape de la gestion.

Le nombre de phases et leurs gestions se diffèrent d'un carrefour à un autre

2.1. le jour

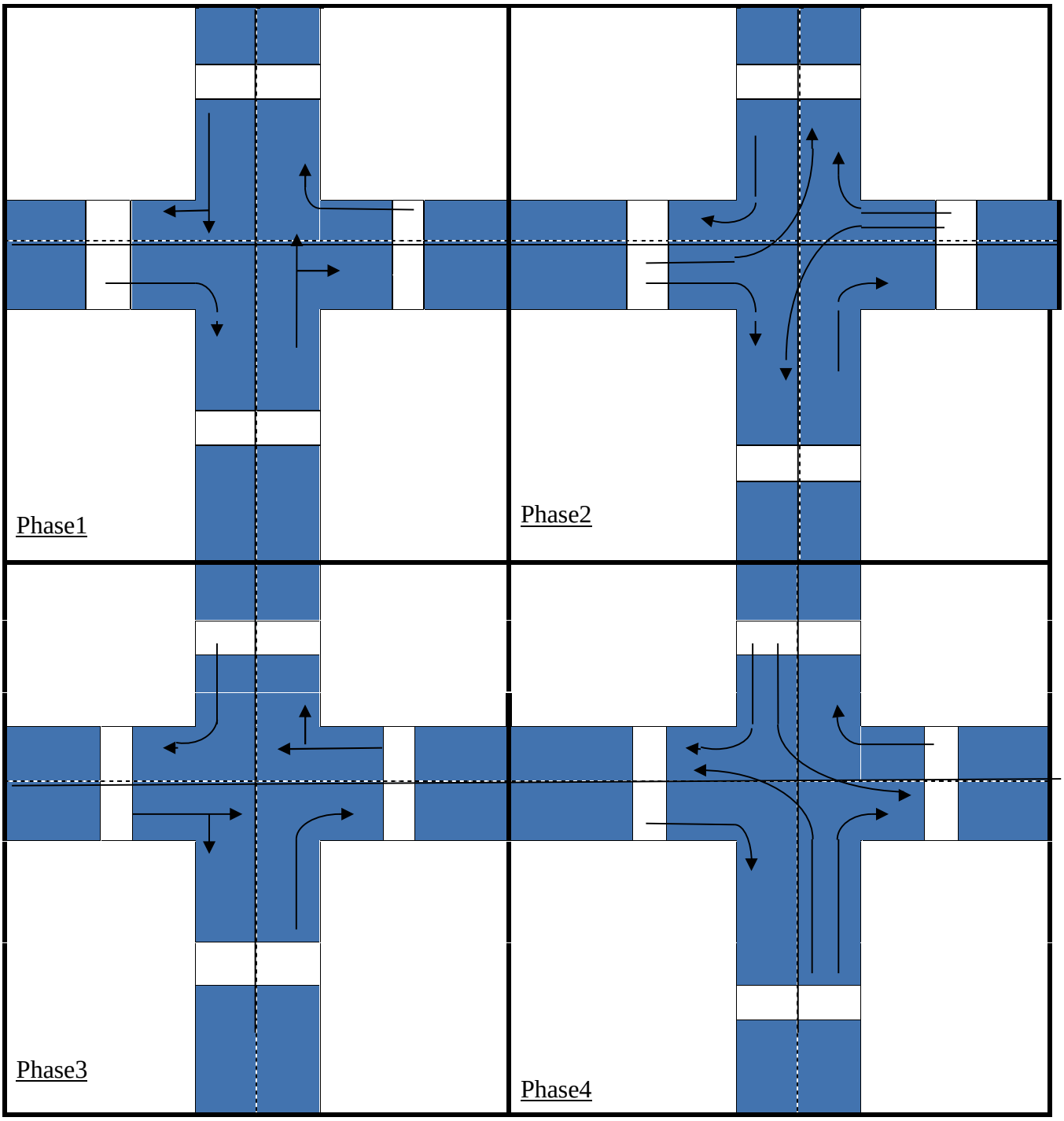


Figure 12:les quatre phases du mode jour

2.2.La nuit

Vu que la ville de Tizi-Ouzou n'est pas active en mode nuit, la gestion des carrefours n'est pas nécessaire, mais aussi il faut prendre en considération certains cas comme :

- Les cas urgents dont il faut clignoter les feux en orange pour indiquer qu'il y a un carrefour et pour que les usagers fassent attention.
- Le cas du mois de carême où il faut fonctionner les feux en mode jour.

Le passage du mode jour au mode nuit à partir d'une carte Arduino sera réaliser manuellement avec un interrupteur.

3. Les chronogrammes de fonctionnement :

3.1. En mode jour :

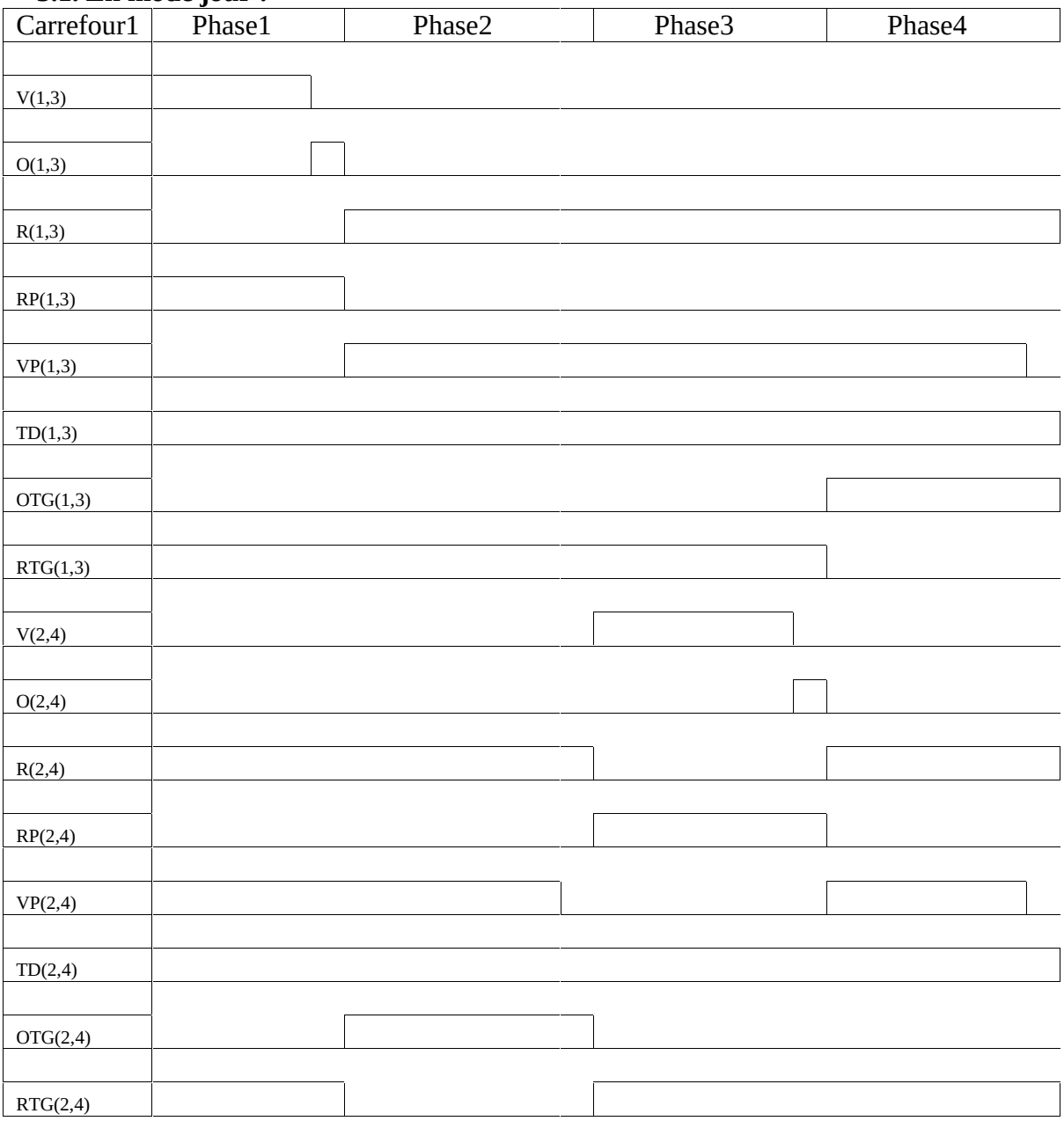


Figure 13: Fonctionnement du premier carrefour

3.1. En mode jour :

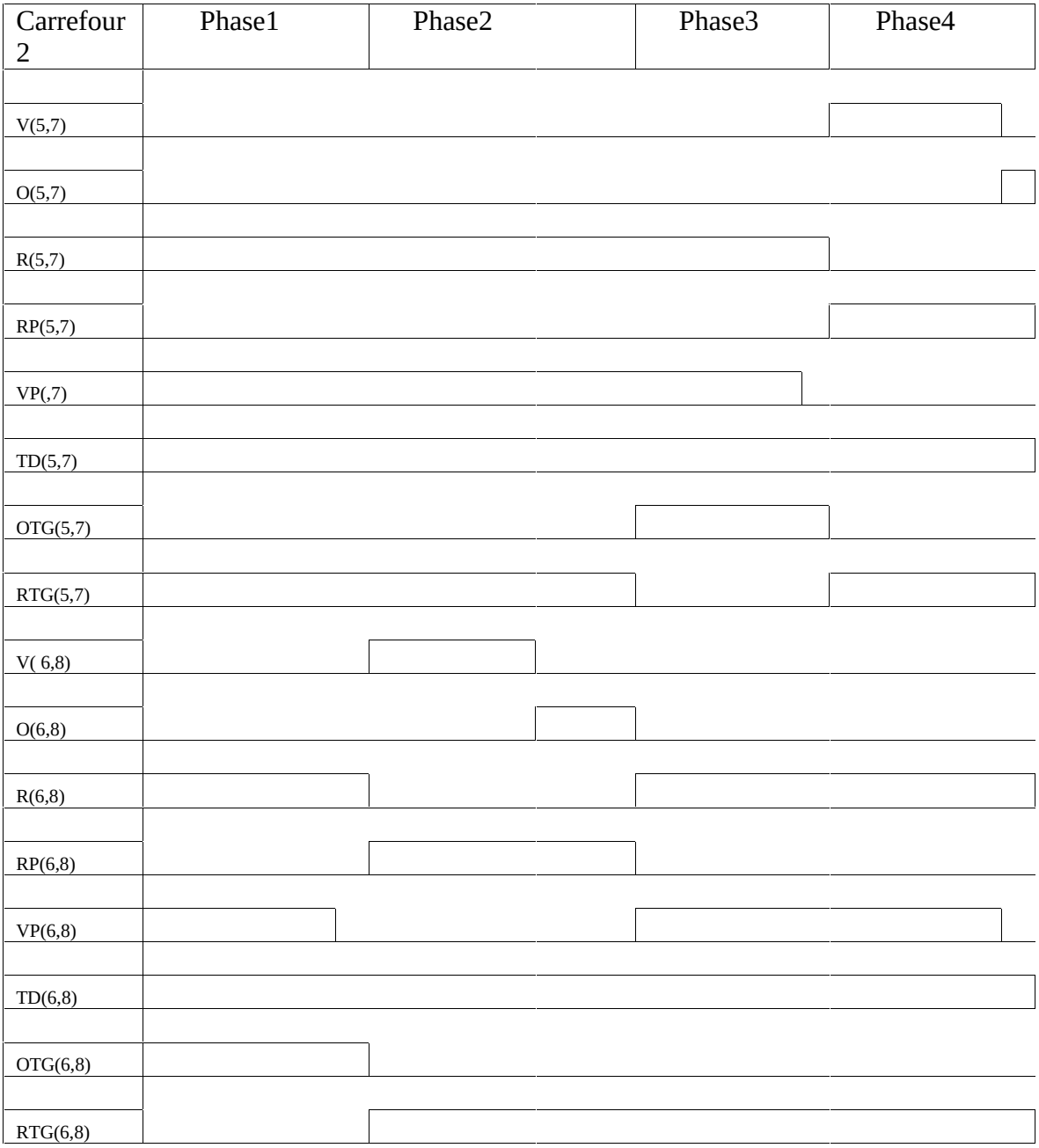


Figure14:Fonctionnement du deuxième carrefour

Ce chronogramme explique le fonctionnement des feux de signalisation avec deux niveaux, le premier est le niveau haut (« 1 » logique) qui veut dire que la led est allumée, le deuxième est le niveau bas (« 0 » logique) qui veut dire que la led est éteinte.

4. Explication de L'organigramme

Cet organigramme traduit tout le programme de fonctionnement de deux carrefours ; au début nous avons tous les feux qui sont désactivés et pour pouvoir fonctionner notre système nous avons placé une condition qui dit si l'interrupteur est fermé (=1) et que la led verte est allumée, le mode jour sera activé et si c'est non c'est le mode nuit qui sera activé (led rouge s'allumera et l'interrupteur s'ouvrera (=0)). Ensuite dans le mode jour nous avons les deux carrefours qui fonctionnent en parallèle avec des états différents, où quand l'ordre des états dans le premier carrefour est " état1, état2, état3,……état8", dans le deuxième sera état3, état4,……état8, état1, état1. le mode nuit fonctionnera après avoir ouvrir l'interrupteur comme nous l'avons dit précédemment.

Enfin, le cycle de fonctionnement ce répétera à chaque fois qu'il terminera.

7. Schéma

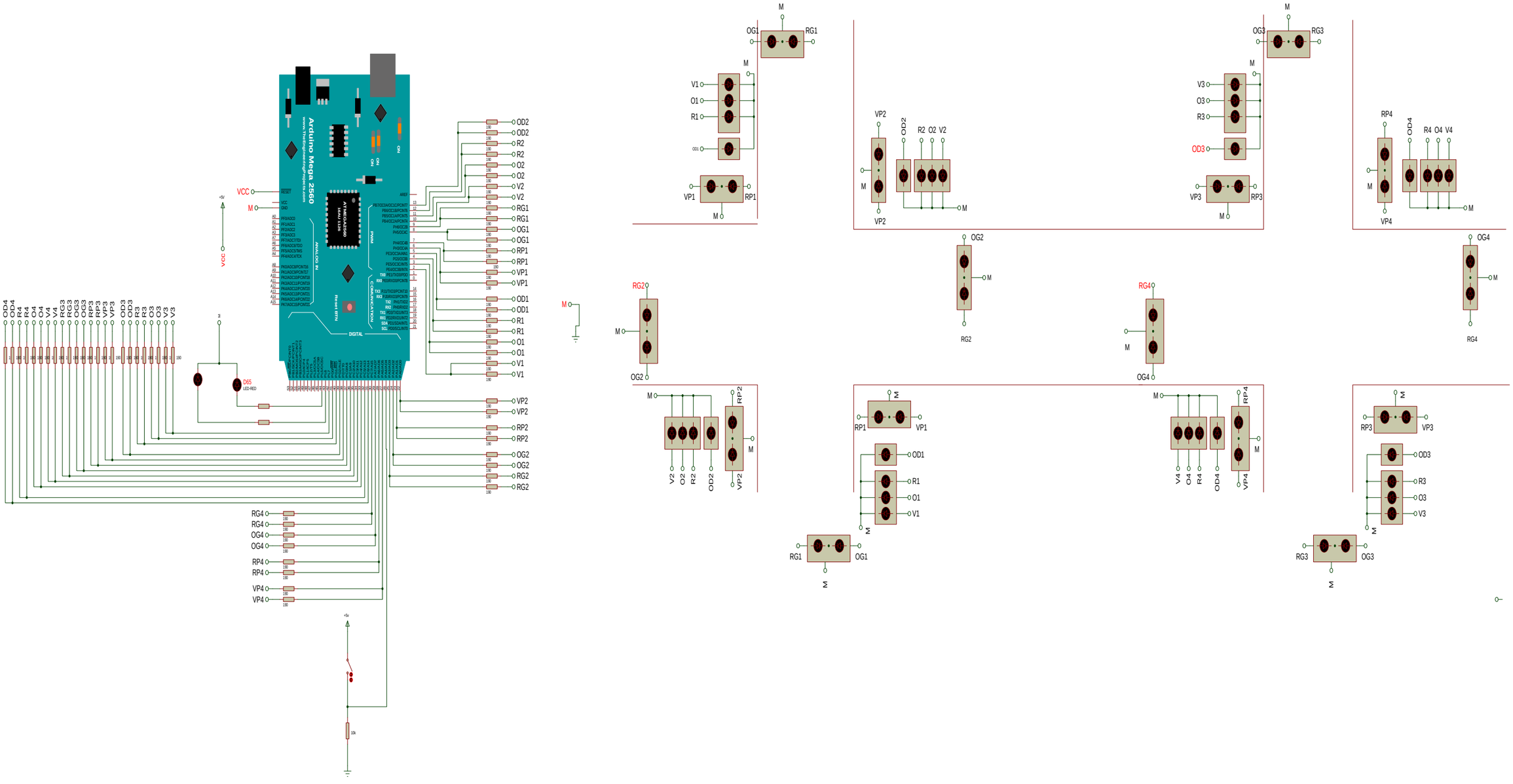
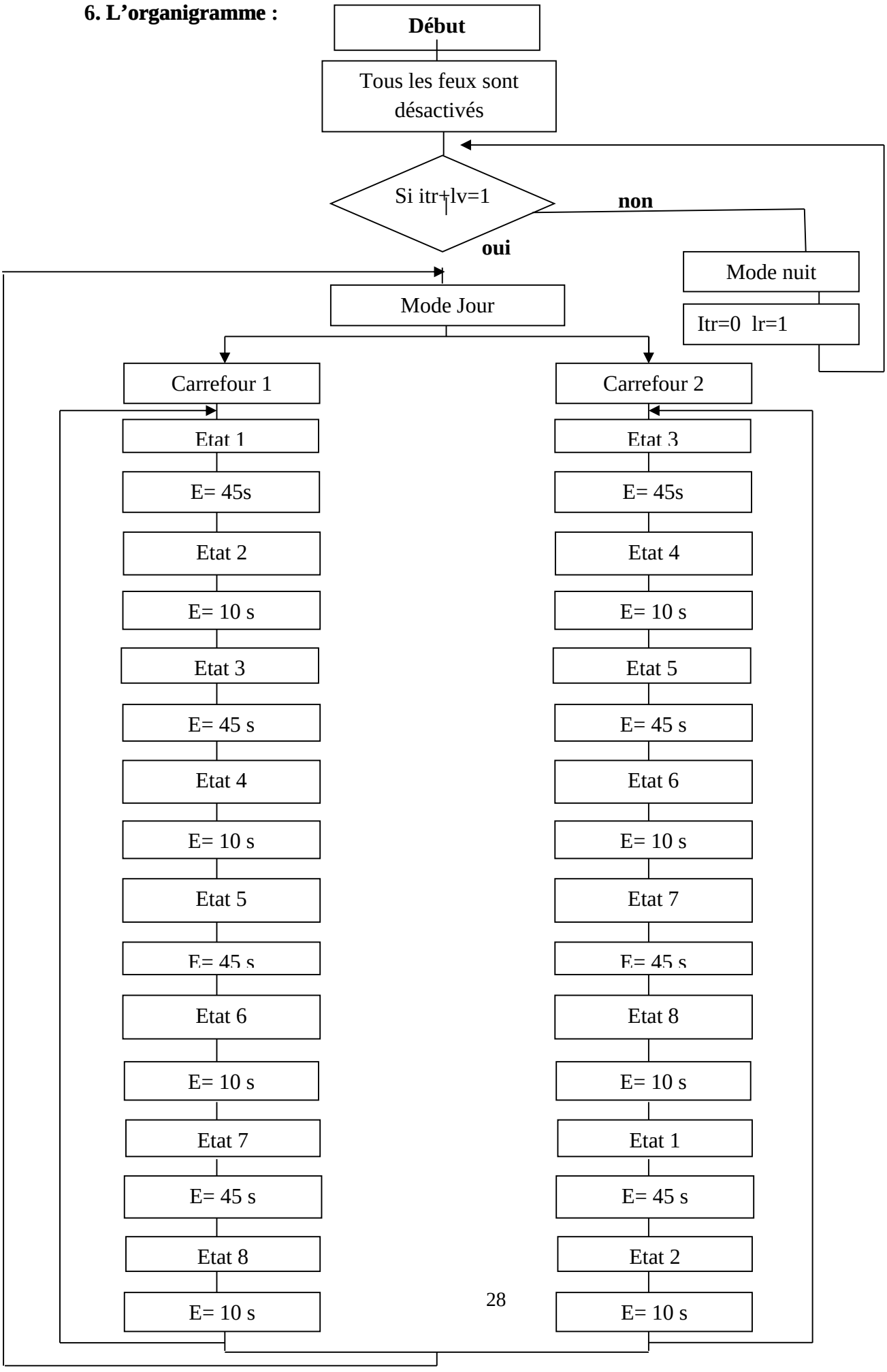


Figure 15 : Schéma réalisé sous ISIS proteus

6. L'organigramme :



7. Le programme :

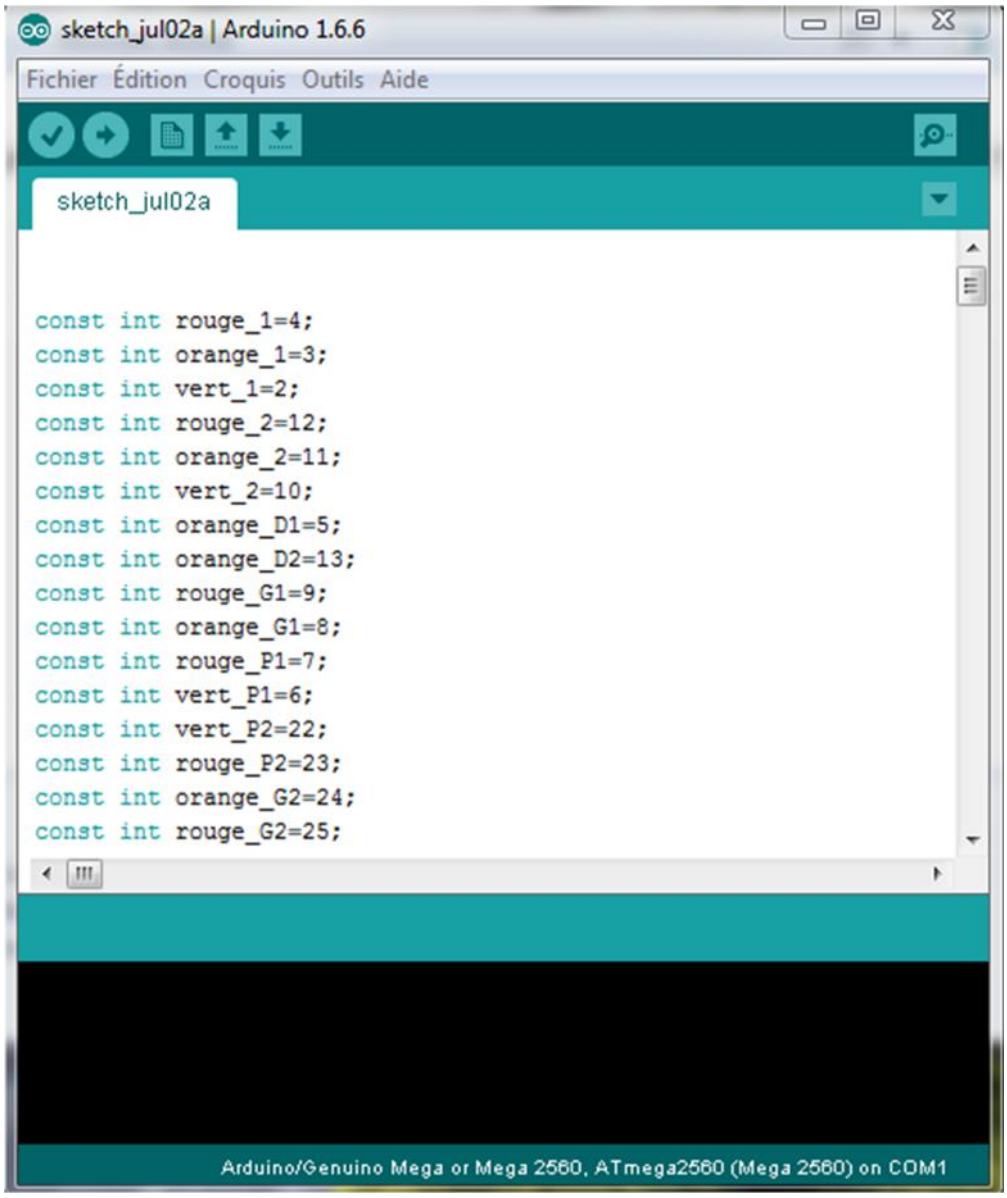


Figure 15 : Le programme sous Arduino

8. La parie réalisation

Apres avoir fait le programme, nous passons a la réalisation de notre carte électronique.

La figure ci-dessous nous montre le branchement des composants.

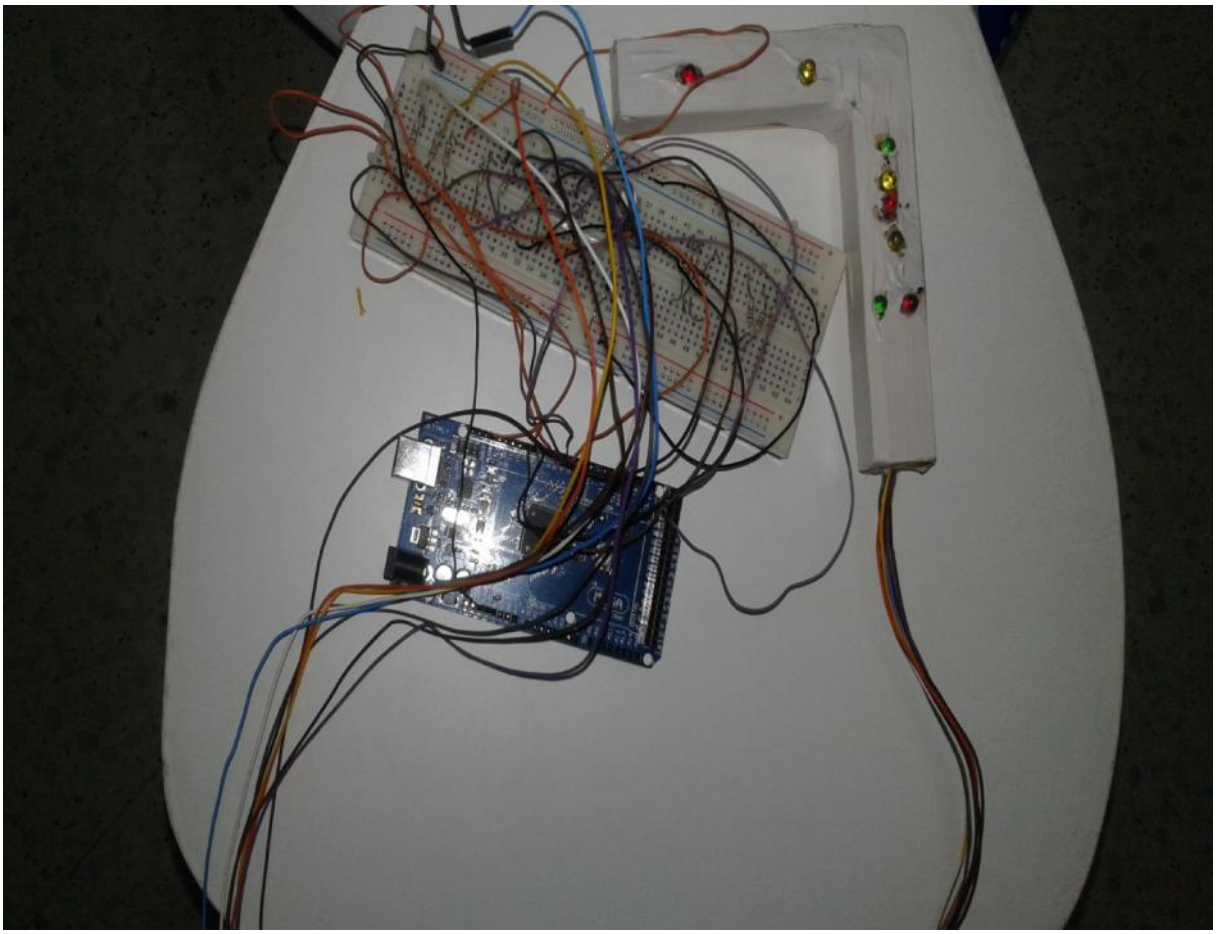


Figure 16 : Branchement des différents composants

_ La led vert remplace le feu vert qui indique la priorité de passage.

_ La led orange remplace le feu orange qui indique prudence ralentir.

_ La led rouge remplace le feu rouge qui indique un stop.

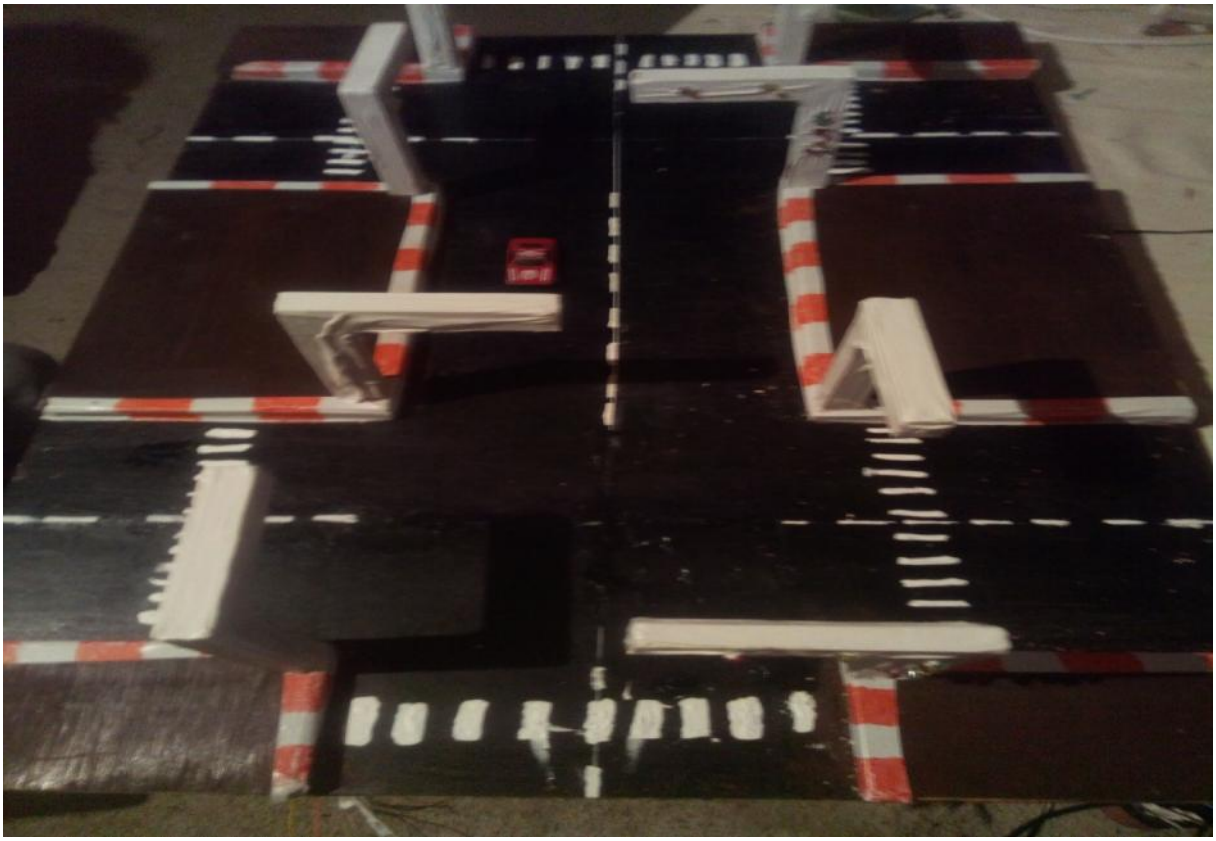


Figure 17 : Représentation de deux carrefour.

8.1. Les états du carrefour en mode jour :

Figure 18 : Etat 1 du carrefour

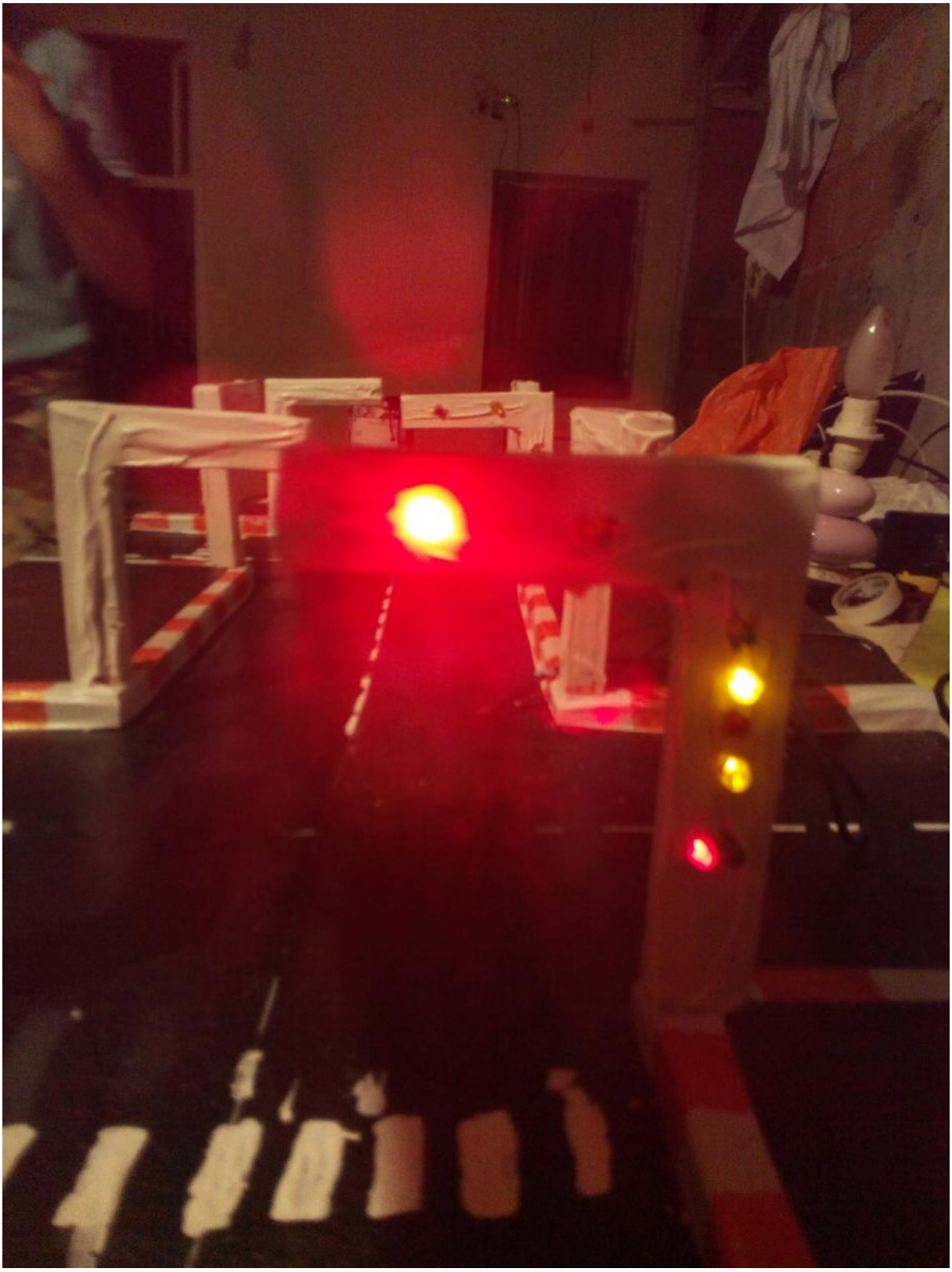


Figure 19 : Etat 2 du carrefour

L’objectif de notre mémoire est de montrer le système des carrefours, qui est un système très avancé pour gérer la circulation des véhicules et des piétons sur les carrefours, pour réaliser ce dernier nous avons fait une gestion urbaine des feux de circulation qui marche avec un temps raisonnable et réglé.

Dans ce travail nous avons présenté quelques généralités sur le trafic urbain et les feux de circulation.

Nous avons utilisé la carte Arduino pour traduire, programmer et réaliser notre gestion des feux de circulation sur les carrefours ; cette gestion est représenter par un organigramme qui organise le fonctionnement des feux en mode jour et nuit et cela est représenter sous forme de schéma avec un système de branchement respectif.

Ce travail est un projet très important dans la vie quotidienne, c’est un programme qui permet de régler les différents problèmes de circulation, de congestion mais aussi de minimiser les accidents de routes qui sont souvent lié a la mauvaise gestion des feux.

(1) : These Sammoud-Bassem-UTBM-pdf.

(2) : GE5S-2011-MEUNIER-memoire-pdf.

(3) : Article 109-3-df.

(4) : Tolba-2004-/Cohen 1990-pdf.

(5) : Caralg01-pdf.

(6) : These Sammoud-Bassem-UTBM-pdf.

(7) : Https//www.flossmanualsfr.net/-booti/arduino/arduino/arduino.pdfmai2017

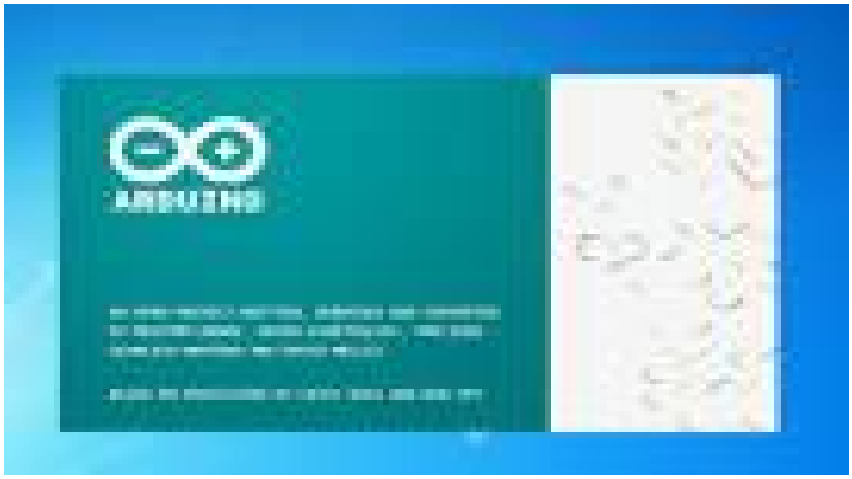
(8) : ERIK Bartamann. «le grand livre d'arduino».2eme Edition. Livre2015.

(9) : https://fr.farnell.com/atmelmega2560-16au/micro-8-bits-256K-flash-5v-cms/dp/1288350#tech Does hook mai2017.

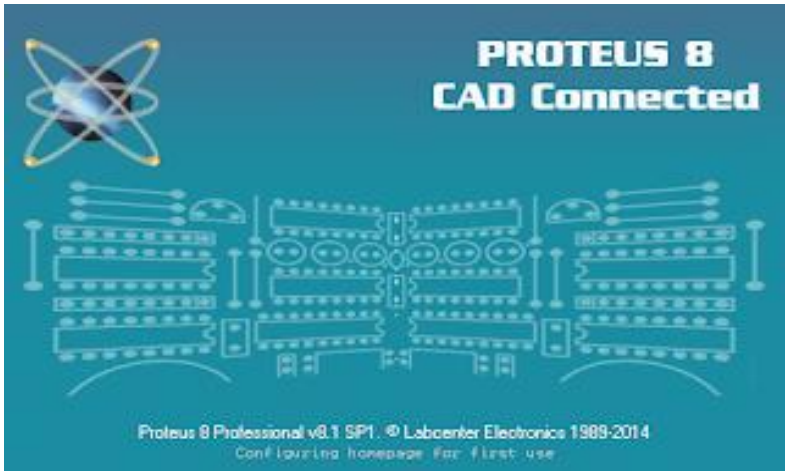
(10): https;//www.mon-club elrc.fr/pmwiki-reference-arduino/pmwiki-php? n=main. Librairie SPI mai 2017 mai

(11) : https://igm.univ-univ.mLv.fr/duris/NTROZO/2004/2005/nguyen-vogvilay-(12) Wolwise Bleutooth.pdf mai 2017.

1. **Les logiciels Utilisés :**



2. **Pour programme Arduino :**



3. Les états de le l'organigramme :

Carrefour 1

Phase 1 :

Etat 1 :

V(1,3)= 1

O(1,3)=0

R(1,3)=0

RP(1,3)=1

VP(1,3)=0

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=0

R(2,4)=1

RP(2,4)=0

VP(2,4)=1

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Etat 2 :

V(1,3)= 0

O(1,3)=1

R(1,3)=0

RP(1,3)=1

VP(1,3)=0

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=0

R(2,4)=1

RP(2,4)=0

VP(2,4)=1

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Phase 2 :

Etat 3 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=0

R(2,4)=1

RP(2,4)=0

VP(2,4)=1

OTD(2,4)=1

OTG(2,4)=1

RTG(2,4)=0

Etat 4 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=0

R(2,4)=1

RP(2,4)=0

VP(2,4)=1

OTD(2,4)=1

OTG(2,4)=1

RTG(2,4)=0

Phase 3 :

Etat 5 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=1

O(2,4)=0

R(2,4)=0

RP(2,4)=1

VP(2,4)=0

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Etat 6 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=1

R(2,4)=0

RP(2,4)=1

VP(2,4)=0

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Phase 4 :

Etat 7 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=1

R(2,4)=0

RP(2,4)=1

VP(2,4)=0

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Etat 8 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=1

RTG(1,3)=0

V(2,4)=0

O(2,4)=0

R(2,4)=1

RP(2,4)=0

VP(2,4)=1

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Carrefour 2 :

Phase 2 :

Etat 3 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=0

R(2,4)=1

RP(2,4)=0

VP(2,4)=1

OTD(2,4)=1

OTG(2,4)=1

RTG(2,4)=0

Etat 4 :

V(1,3)= 0

O(1,3)=1

R(1,3)=0

RP(1,3)=1

VP(1,3)=0

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=0

R(2,4)=0

RP(2,4)=0

VP(2,4)=1

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Phase 3 :

Etat 5 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=1

O(2,4)=0

R(2,4)=0

RP(2,4)=1

VP(2,4)=0

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Etat 6 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=1

R(2,4)=0

RP(2,4)=1

VP(2,4)=0

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Phase 4 :

Etat 7 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=1

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=1

R(2,4)=0

RP(2,4)=1

VP(2,4)=0

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Etat 8 :

V(1,3)= 0

O(1,3)=0

R(1,3)=1

RP(1,3)=0

VP(1,3)=0

OTD(1,3)=1

OTG(1,3)=1

RTG(1,3)=0

V(2,4)=0

O(2,4)=0

R(2,4)=1

RP(2,4)=0

VP(2,4)=0

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Phase 1 :

Etat 1:

V(1,3)= 1

O(1,3)=0

R(1,3)=0

RP(1,3)=1

VP(1,3)=0

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=0

R(2,4)=1

RP(2,4)=0

VP(2,4)=1

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1

Etat 2 :

V(1,3)= 0

O(1,3)=1

R(1,3)=0

RP(1,3)=1

VP(1,3)=0

OTD(1,3)=1

OTG(1,3)=0

RTG(1,3)=1

V(2,4)=0

O(2,4)=0

R(2,4)=0

RP(2,4)=0

VP(2,4)=1

OTD(2,4)=1

OTG(2,4)=0

RTG(2,4)=1