République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de L'**E**nseignement **S**upérieur et de la A **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DE **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Filière : Génie Electrique
Spécialité : Télécommunication et Réseaux

Présenté par
Samir TRIKI

Mémoire dirigé par **Mr Rachid ZIRMI**

Thème

Conception et réalisation d'un système de gestion d'éclairage d'une maison photovoltaïque

Dédicaces

Je dédie ce travail à :

A ma mère,

A mon père,

A mes frères et sœur,

A toute ma famille,

A tous mes enseignants et mes amis à l'université de Tizi-Ouzou.

A tous ceux que j'aime

Qu'ils trouvent ici l'expression de toute ma reconnaissance.

Remerciements

Ce modeste projet est l'occasion pour moi de remercier toutes les personnes qui ont contribué à ce travail ainsi que celles que j'ai pues rencontrer durant ces années d'études.

Tout d'abord, je tiens à exprimer toute ma profonde reconnaissance à Monsieur Rachid ZIRMI pour m'avoir proposé ce sujet passionnant. Je le remercie, aussi, de toujours m'avoir proposé de nouvelles idées de qualité pour mon travail. Je le remercie, également, de m'avoir témoigné de sa confiance et de son aide scientifique. Sans lui, la thèse n'aurait jamais vu le jour.

Je tiens aussi à remercier le président et les membres du jury qui m'ont fait l'honneur de participer à l'examen de ce modeste travail.

Mes remerciements du fond du cœur vont à ma famille qui a su me donner, sans cesse, son soutien, son amour et l'envie d'apprendre encore plus.

Enfin, merci à tous ceux qui m'ont aidé de près ou de loin durant ces années d'études, un grand merci à tous mes amis.

Sommaire

Introduction générale …… 1

Chapitre I : Généralités sur Arduino, LabVIEW et les systèmes PV Autonomes

I. Introduction ……3

II. Généralités sur Arduino…….3

II.1. Caractéristiques de la carte Arduino Mega …….3

II.2. Environnement de développement logiciel Arduino ……..7

II.3. Structure d'un programme Arduino…… 9

III. Généralités sur LabVIEW ……..11

III.1. Structure d'un programme labVIEW ……...11

III.2. Les outils de programmation sous LabVIEW ……..12

III.3. Exécution d'un programme LabVIEW ……15

IV. Les systèmes PV autonome ……...16

IV.1. Description ……...16

IV.2. Le générateur PV ……17

IV.2.a. Principe de Fonctionnement de la Cellule Photovoltaïque ……..17

IV.2.b. modélisation d'une cellule photovoltaïque ……..18

IV.2.c. Caractéristique courant-tension d'une cellule photovoltaïque …….19

IV.2.d. Effets des Variations Climatiques sur la caractéristique I(V) et P(V) d'une cellule PV ……20

IV.3. Régulateur MPPT ……21

Chapitre II : Conception et réalisation matérielle du système de gestion de l'éclairage

I. Introduction ……...23

II. Fonctionnement du système de gestion d'éclairage ……....23

III. Schéma bloc du système de gestion d'éclairage ……...25

III.1. Le bloc détecteur et capteur …….27

III.2. Le bloc d'éclairage ……...29

III.3. Le bloc de commande ……..30

III.4. Un système PV ……33
IV. Simulation sur Isis Proteus ……34
V. Réalisation Pratique ……38

Chapitre III : Développement logiciel

I. Introduction ...……39
II. Description du Programme Arduino ……39
III. Description de l'application LabVIEW ……..43
III.1. Le diagramme de l'application LabVIEW ……43
III.2. Face avant de l'application LabVIEW ……..46
III.3. Publication Web de l'interface Graphique LabVIEW …….....47

Conclusion générale ……...…….……....49

Bibliographie ……...……..50

INTRODUCTION GÉNÉRALE

Face aux défis du réchauffement climatique et de l'épuisement des ressources fossiles, l'efficacité énergétique est devenue un des sujets clefs dans tous les domaines d'activités, y compris le secteur résidentiel. Dans ce dernier, les besoins en énergie sont très variés et l'éclairage, après l'électroménager, représente une part non négligeable de la consommation électrique des familles.

Plusieurs études ont identifié que la part d'énergie électrique utilisée pour l'éclairage représente plus de 15 % de la consommation électrique totale d'un ménage. Pourtant, au contraire des installations de chauffage et de l'électroménager, les systèmes d'éclairage économes en énergie sont peu mis en avant et peu utilisés [1] [2].

Pour ces raisons, plusieurs gouvernements dans le monde incitent leurs citoyens à utiliser dans leurs habitations des équipements et techniques permettant la réduction de l'énergie électrique utilisée pour s'éclairer.

Parmi les solutions offertes par les professionnels du domaine permettant la réduction de l'énergie électrique consommée pour l'éclairage dans le bâtiment, on trouve deux types,
Le premier type consiste en un certain nombre de solutions à mettre en œuvre sans changer le circuit d'éclairage existant dans une habitation donnée, parmi ces solutions on peut trouver :

- Le changement des lampes classique à incandescence par d'autres types de lampes économiques comme les lampes basses consommation et les lampes LED
- installer des détecteur de présence permettant d'éteindre l'éclairage s'il n y a pas de présence de personne dans une zone donnée d'une habitation.

Le deuxième type consiste à installer un système de gestion d'éclairage permettant de gérer d'une manière intelligente et optimale l'éclairage dans les habitations. Ce système est basée principalement sur l'utilisation de capteurs (capteurs de présence de personnes, de présence de lumière naturelle,…) et d'un contrôleur qui ordonne automatiquement l'extinction ou l'allumage des lampes en fonction des informations qu'il reçoit des capteurs.

Ce présent mémoire de Master est consacré à la conception et réalisation d'un système d'éclairage d'une maison comportant quatre pièces (salon, chambre, cuisine et salle de bain) et dont l'alimentation en énergie électrique est assurée par un système photovoltaïque (PV) autonome.

L'objectif de ce système d'éclairage est assuré le confort lumineux nécessaire dans la maison tout en utilisant le minimum d'énergie électrique (photovoltaïques).

Pour réaliser cet objectif le principe mis en œuvre pour la gestion d'éclairage repose sur le fait que, pour assurer le confort lumineux nécessaire dans une pièce donnée de la maison, la lampe servant à éclairer cette pièce n'est allumée que lorsqu'une personne est présente dans cette pièce et que la lumière du jour, rentrant par les fenêtres, ne suffit pas à assurer ce confort lumineux.

Le mémoire est reparti en trois chapitres :

Le premier chapitre est consacré à la présentation d'un certain nombre de généralités et connaissances que nous avons utilisées pour la réalisation du système de gestion d'éclairage. En premier lieu, nous avons présenté les caractéristiques et la programmation de la carte Arduino Mega que nous avons utilisé pour contrôler notre système de gestion d'éclairage. En second lieu, nous avons présenté l'environnement de développement LabVIEW utilisé pour développer l'application de supervision de l'état du système de gestion d'éclairage. En dernier lieu, nous avons défini les composants d'un système photovoltaïque autonome.

Le deuxième chapitre expose en détail la partie hard de notre système réalisé. Ce chapitre comprend trois parties. La première partie est consacrée à la description du fonctionnement du système de gestion de l'éclairage. La deuxième partie est dédiée à la présentation du schéma bloc générale et à la description détaillée des blocs qui compose le schéma bloc. La dernière partie est consacrée à la réalisation pratique du système de gestion d'éclairage.

Le dernier chapitre présente la partie Soft de notre travail composée de trois applications logicielles. La première est le programme implémenté dans la carte Arduino traduisant le fonctionnement du système de gestion de l'éclairage. La deuxième partie décrit l'application que nous avons développée sous l'environnement LABVIEW. Le rôle de cette application est de lire les informations écrites sur le port série de la carte Arduino et d'afficher sur une interface graphique l'état des détecteurs de présence, des lampes et des volets. La dernière partie présente une publication web de l'interface graphique de l'application LabVIEW, permettant de télé-surveiller à distance l'état du système de gestion d'éclairage.

Ce mémoire est clôturé par une conclusion générale qui regroupe les contributions et les résultats obtenus ainsi que des perspectives pour des travaux futures.

CHAPITRE I

Généralités sur Arduino LabVIEW et les systèmes PV Autonomes

I. Introduction :

Avant d'entamer la description du système de gestion d'éclairage que nous avons réalisé, nous présentons tout d'abord un certain nombre de généralités (connaissances) que nous avons utilisées pour la réalisation de notre travail. En premier lieu ces généralités en concernées les caractéristiques et la programmation de la carte Arduino Mega. En second lieu, nous avons présenté l'environnement de développement LabVIEW. En dernier lieu, nous avons défini les composants d'un système PV autonome.

II. Généralités sur Arduino :

Arduino est une plate-forme open-source d'électronique programmée qui est basée sur une carte à microcontrôleur (de la famille AVR) et un environnement de développement intégré(IDE) pour écrire, compiler et transférer les programmes vers la carte à microcontrôleur [3].

Arduino peut être utilisé pour construire des objets interactifs indépendants (prototypage rapide), ou bien pour être connecté à un ordinateur afin communiquer avec des logiciels tournant sur un ordinateur (tels que Flash, Processing, LabView ou Matlab).

Les cartes Arduino sont basées sur une interface entrée /sortie simple pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou tout autre types d'actionneurs.

Le langage de programmation Arduino est une implémentation de Wiring, une plate-forme de développement similaire, qui est basée sur l'environnement multimédia de programmation Processing [4].

Arduino offre une grande variété de cartes électroniques (DUEMILANOVE, MEGA, LEONARDO, UNO, MINI ...) qui différent entre elles par des caractéristiques telle que la vitesse du microcontrôleur, type de connexion, nombre d'entrées/sorties, nombre des microcontrôleurs.

Pour la réalisation de notre système d'éclairage intelligent, notre choix c'est porté sur la carte Arduino Méga 2560.

II.1. Caractéristiques de la carte Arduino Mega :

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un microcontrôleur ATmega2560.Cette carte dispose :

a. Alimentation :

La carte Arduino Mega 2560 peut être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA), soit à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte [4].

b. Capacité Mémoire :

L'ATmega 2560 contient 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader). L'ATmega 2560 a également 8 ko de mémoire SRAM (mémoire volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM) [5].

Le bootloader est un programme préprogrammé une fois pour toute dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte [5].

c. broches d'Entrées/Sorties :

Ce sont les rangées de connecteurs de part et d'autre de la carte qui permettent sa connexion au monde extérieur. On distingue plusieurs types de broches d'entrées/sorties :

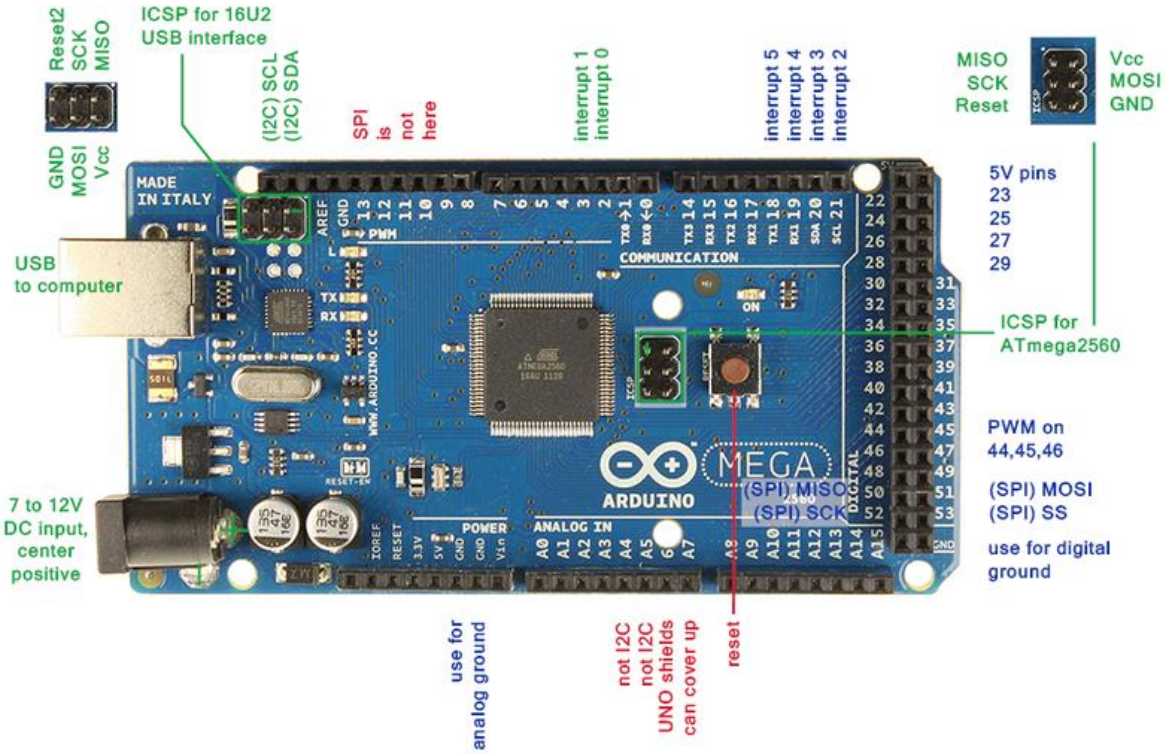


Figure. 1 : La connectique de l'Arduino Méga

- Broches d'entrées et sorties numériques :

La carte Arduino Mega dispose de 54 broches numériques d'entrées/sorties. Chacune de ces broches peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité.

De plus, certaines broches ont des fonctions spécialisées :

- **Communication Serie**: Port Serie Serial : 0 (RX) and 1 (TX); Port Serie Serial 1: 19 (RX) and 18 (TX); Port Serie Serial 2: 17 (RX) and 16 (TX); Port Serie Serial 3: 15 (RX) and 14 (TX) [5].

  Ces broches sont utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL.

- **Interruptions Externes**: Broches 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), et 21 (interrupt 2). Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur [5].

- **Impulsion PWM (largeur d'impulsion modulée)**: Broches 0 à 13. Fournissent des sorties PWM codées sur 8-bits.

- **SPI (Interface Série Périphérique)** : Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI (Interface Série Périphérique). Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Uno, Duemilanove et Diecimila.

- **I2C** : Broches 20 (SDA) et 21 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils").

- **LED** : Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

- **Broches d'entrées analogiques :**

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (c.à.d sur 1024 niveaux soit de 0 à 1023). 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.

Note : les broches analogiques peuvent être utilisées en tant que broches numériques.

- **Les broches d'alimentation :**

Les broches d'alimentation sont :

- 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)

- **5V**. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte.

- **3V3**. Une alimentation de 3.3V fournie par un circuit intégré permettant de faire l'adaptation du signal entre le port USB du PC et le port série de l'ATmega. Ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA.

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

- **Autres broches :**

Il y a deux autres broches disponibles sur la carte :

- **AREF** : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction analogReference() [6].
- **Reset** : 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.

II.2. Environnement de développement logiciel Arduino :

La carte Arduino présente le noyau de notre système, et pour qu'on puisse la programmer on doit de disposer du logiciel compatible avec cette carte, ce logiciel est **Arduino EDI** (Espace de développement intégrée) est une application écrite en Java inspiré du laquage Processing [4].

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

La fenêtre principale de cet EDI comporte six parties : (figure 2)

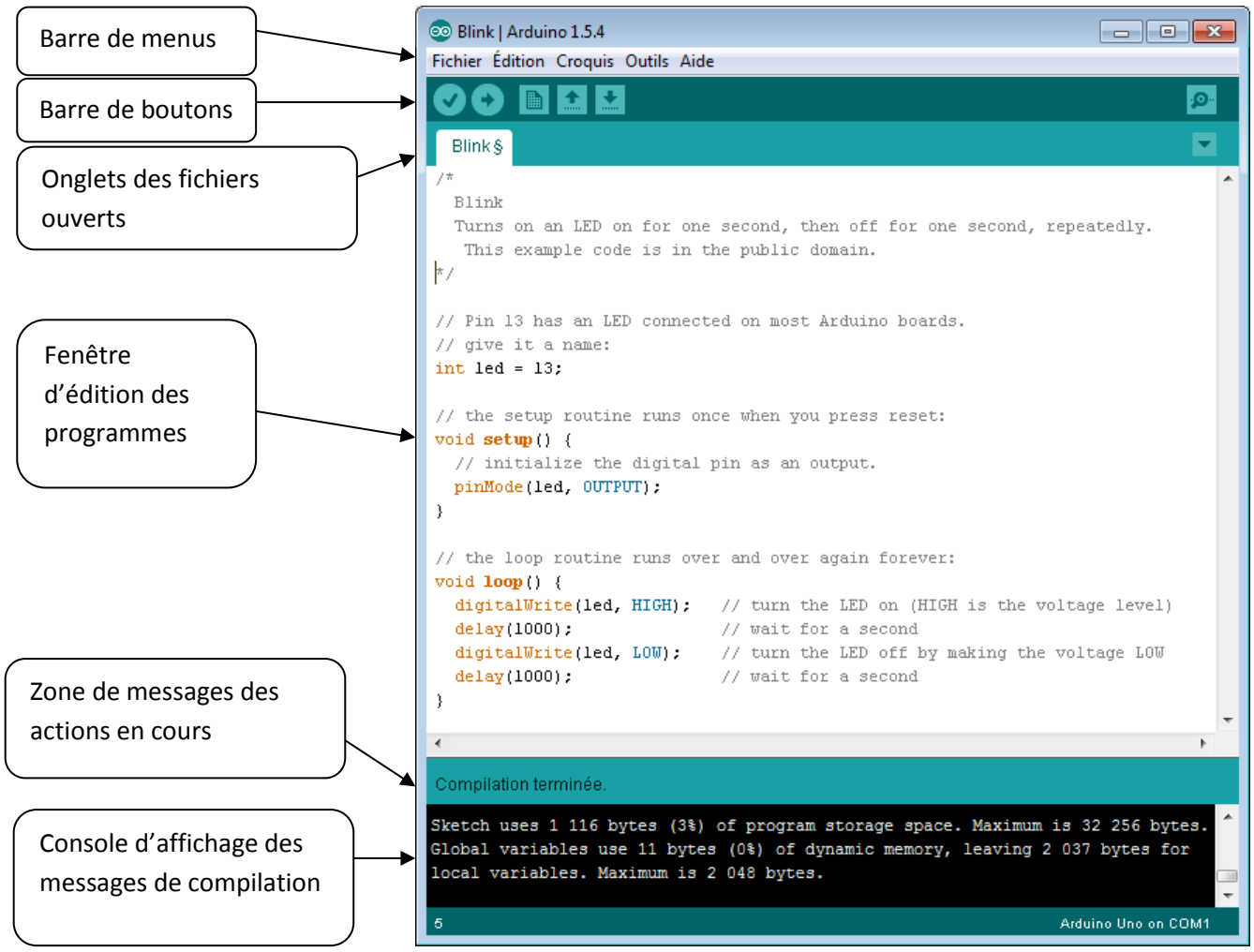
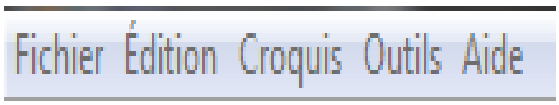


Figure.2 : Les différentes parties de la fenêtre principale du logiciel Arduino.

1. Une barre de menus :



Cette barre contient les icones suivantes :

- ✓ **File (Fichier)** : ce menu contient les différentes options de créations, d'ouverture' de sauvegarde, d'impression du programme, ou l'ouverture d'un exemple parmi les exemples qui accompagnent le logiciel Arduino [3].
- ✓ **Edite (Editer)** : ce menu contient les options de copier/coller, sélection, et les options de recherche.
- ✓ **Sketch (Programme ou séquence) :** ce menu contient les différentes fonctions de la barre des boutons, ainsi que les options d'ajouts de librairies ou de fichiers.
- ✓ **Tools (Outils) :** c'est dans ce menu qu'on sélectionne le type de carte à programmer, et le port série utilisée ainsi que la fonction du chargement du bootloader dans l'ATmega.
- ✓ **Help (Aide) :** ce menu est fait pour donner de l'aide concernant les différents problèmes rencontrés au niveau du logiciel Arduino. [3]

2. Une barre des boutons :

Vérifier/compiler : Vérifie le code à la recherche d'erreur.

Transférer vers la carte : Compile votre code et le transférer vers la carte Arduino.

Nouveau : Crée un nouveau code (ouvre une nouvelle fenêtre)

Ouvrir : Ouvre la liste de tous les programmes dans le « livre de programmes".

Enregistrer : Enregistre un programme.

Moniteur Série : Ouvre la fenêtre du moniteur (ou terminal) série.

3. un ou plusieurs onglets des fichiers correspondant aux sketchs écrits ou en cours d'écriture ;

4. une fenêtre d'édition des programmes C'est l'espace utilisé pour écrire le programme à réaliser, comme il dispose aussi des onglets de navigation [7].

5.Une zone de messages qui est utilisée par l'EDI pour communiqué à l'utilisateur des informations sur l'état du programme et des actions en cours.

6.Une console texte qui affiche les messages concernant le résultat de la compilation du programme (il nous indique si le programme comporte des erreurs) [3].

II.3. Structure d'un programme Arduino :

Le langage Arduino est basé sur les langages C/C++.

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code, comme lors d'une programmation classique. Cette structure se décompose en trois parties :

- **Partie 1 : Définition des constantes et variable :**

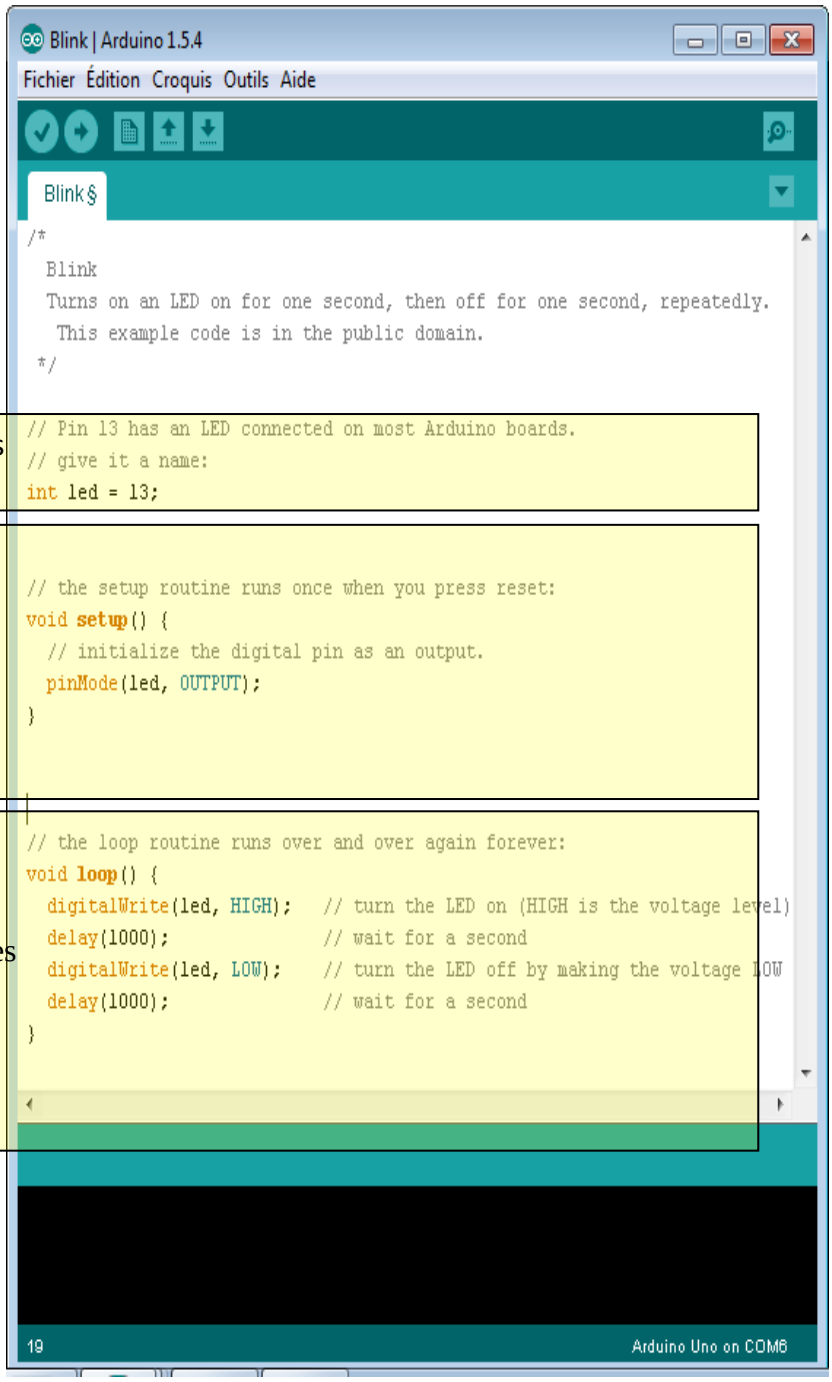
  Dans cette partie sont définit les constante et variable qui seront utilisés dans le programme,

- **Partie 2 : la boucle Void steup () :**

  C'est une fonction qui est exécuté une seule fois au démarrage du programme, dans cette partie est réalisé la configuration des pins de la carte Arduio en entrées ou en sortie,

- **Partie 3 : la boucle Void loop () :**

  C'est la partie qui comprend la programmation des interaction et comportement, cette partie s'exécute d'une manière continue.



Définition des constantes et variables globales

Fonction principale : *VOID SETUP()*
Initialisation des ressources de la carte
Configuration des entrées/sorties
Définition de la vitesse de fonctionnement du port série, etc.
Cette partie ne sera exécutée qu'une seule fois

Fonction boucle : *VOID LOOP()*
Description du fonctionnement général du programme
Gestion des interactions entre les entrées/sorties
Cette partie sera exécutée en boucle

Figure.3 : Les différentes composantes du programme Arduino

III. Généralités sur LabVIEW :

LabVIEW (Laboratory Virtual Instrument Engineering WorkBench) est un environnement de programmation graphique. Il utilise un langage de programmation essentiellement graphique dédié au contrôle, l'acquisition, l'analyse et la présentation de données. En LabVIEW, on n'écrit pas de lignes de programme dans un langage textuel comme en Pascal, C, Basic ou Fortran, mais on manipule des objets graphiques. Ces objets graphiques représentent à la fois les variables du programme et les fonctions qui vont réaliser des actions portant sur ces variables. La programmation en LabVIEW consiste simplement à concevoir le traitement de l'information, organiser et relier les variables avec les fonctions au moyen de fils.

LabVIEW offre des bibliothèques étendues de fonctions et de routines répondant à la plupart des besoins en programmation. LabVIEW contient également des bibliothèques de fonctions spécifiques à l'acquisition de données et au pilotage d'instruments GPIB, VXI ou encore d'instruments connectés sur une simple liaison série.

III.1. Structure d'un programme labVIEW :

Un programme LabVIEW est appelé instrument virtuel ou VI (pour Virtual Instrument). Il se compose principalement de deux éléments étroitement associés ; La face avant et le diagramme (figure 4).

- **La face avant** est l'interface utilisateur du VI. Elle représente le panneau de contrôle de l'instrument virtuel, elle permet de réceptionner les données acquises et d'afficher celles fournies en sortie par le programme. La face-avant est construite en utilisant des objets dénommés commandes et indicateurs. Les commandes sont des entrées qui servent à saisir des valeurs à l'écran (des boutons rotatifs, des boutons poussoirs, des cadrans,) et les indicateurs sont des sorties qui servent à afficher des variables ou des résultats de calculs (des graphes, des Leds, …) [8].
- **Le diagramme** est le programme principal de l'application. Il décrit le fonctionnement interne du VI. Le diagramme peut contenir des fonctions et des structures issues des bibliothèques de VIs intégrées à LabVIEW. Il peut aussi contenir des terminaux associés à des commandes et à des indicateurs créés dans la face-avant [9].

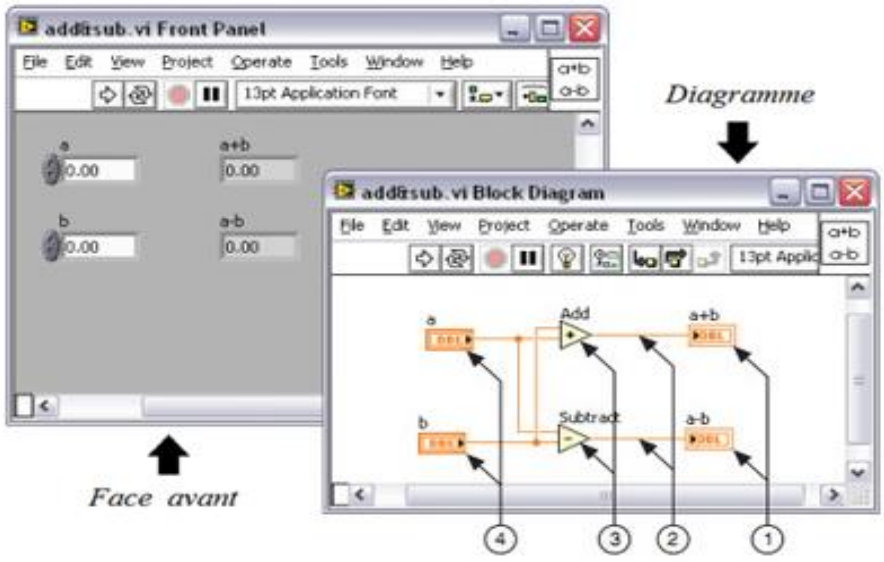


Figure. 4 : Diagramme et face avant d'un programme LabVIEW.

III.2. Les Outils de programmation sous LabVIEW :

Les outils de programmation sous LabVIEW sont répartis en trois palettes qui contiennent un certain nombre d'options qui servent à créer et éditer la face avant et le diagramme des VI. Ces palettes sont :

a. La palette d'outils :

La palette d'outils (figure 5) s'utilise aussi bien dans la face avant que dans le diagramme. Elle contient les outils nécessaires pour éditer et mettre au point les objets du diagramme et de la face avant [10].

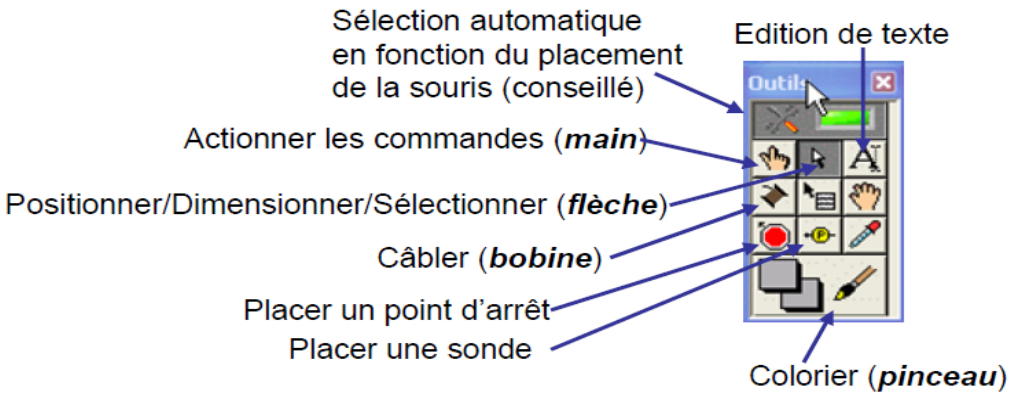


Figure. 5 : la palette d'outils

b. Les palettes de commandes :

Les palettes de commande (figure 6) sont utilisées uniquement dans la face avant. Elles contiennent les commandes et les indicateurs de la face avant qui sert à créer l'interface utilisateur.

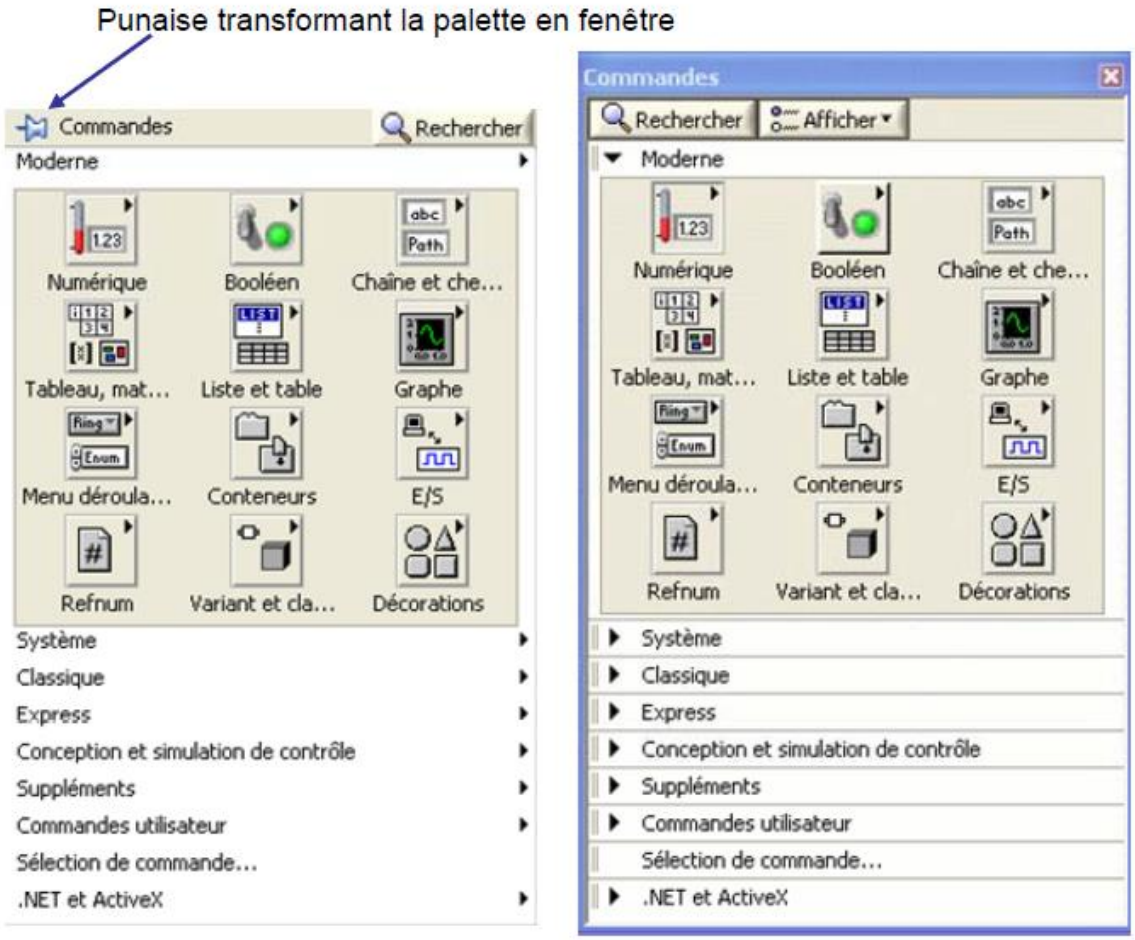


Figure. 6 : palettes de commandes : à gauche sous forme de menu (click droit sur la face-avant), à droite sous forme de fenêtre (via menu déroulant de la face avant, ou bien à l'aide de la punaise).

Plusieurs palettes de commande existent (« Moderne », « Système », « Classique », « Express », « Commandes utilisateur », « .NET et ActiveX », les autres palettes dépendant des modules supplémentaires installés) [8].

La palette « Moderne » est utilisée dans la plupart des VI. Dans cette palette les commandes sont organisées par catégorie (sous palette). Les plus fréquemment utilisé sont :

- « Numérique » : offre des commandes et indicateurs permettant de saisir ou afficher un **numérique**.

- « Booléen » : commandes et indicateurs booléens (correspondant à des variables qui peuvent prendre que la valeur vrai (1) ou la valeur faux (0)). Les **booléens** correspondent aux boutons à 2 états et aux indicateurs à 2 états (LED, etc.).
- « Chaîne et chemin » : commandes et indicateurs permettant de saisir ou d'afficher des **chaînes de caractères**. « Graphe » : propose différents indicateurs de graphes.

c. Les palettes des fonctions :

A l'instar des palettes de commandes utilisées pour la face-avant, Les **palette de fonctions** (Figure 7) contiennent les objets qui servent à programmer les VI, comme par exemple des opération arithmétique, d'E/S d'instrument, d'E/S sur fichiers et d'acquisition de données [11].

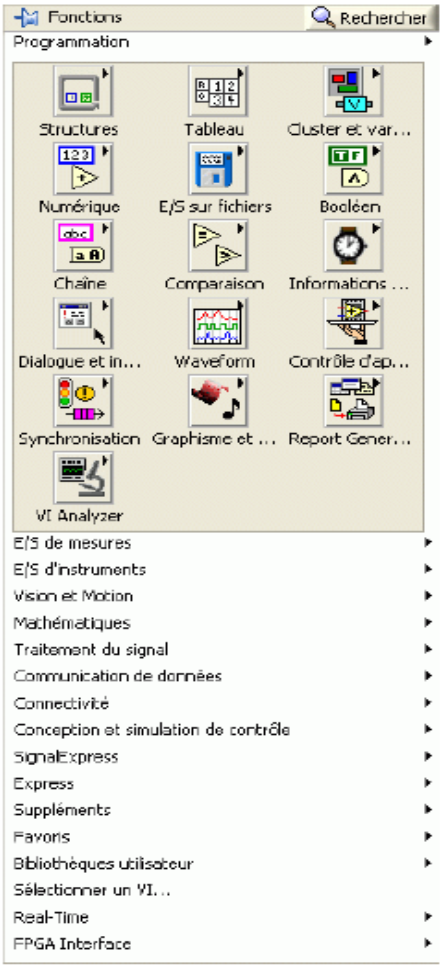


Figure.7 : Les palettes des fonctions

III.3. Exécution d'un programme LabVIEW :

Les VIs de LabVIEW suivent le concept de l'exécution de programme par flux de données. Le diagramme se compose de nœuds, tels que des structures ou des terminaux de la face avant. Ces nœuds sont reliés entre eux par des fils, qui définissent le flux de données à travers le programme [12].

L'exécution d'un nœud n'a lieu que lorsque toutes ses entrées sont disponibles. Dès qu'un nœud a fini de s'exécuter, il transfère toutes ses sorties au prochain nœud afin de s'exécuter, il transfère toutes ses sorties au prochain nœud dans le chemin du flux de données.

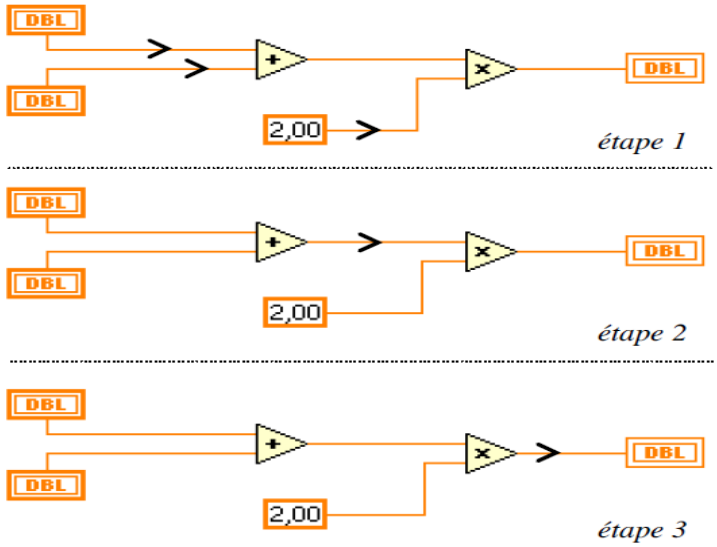


Figure.8 : Exécution d'un programme LabVIEW par flux de données.

IV. Les systèmes PV autonome :

IV.1. Description :

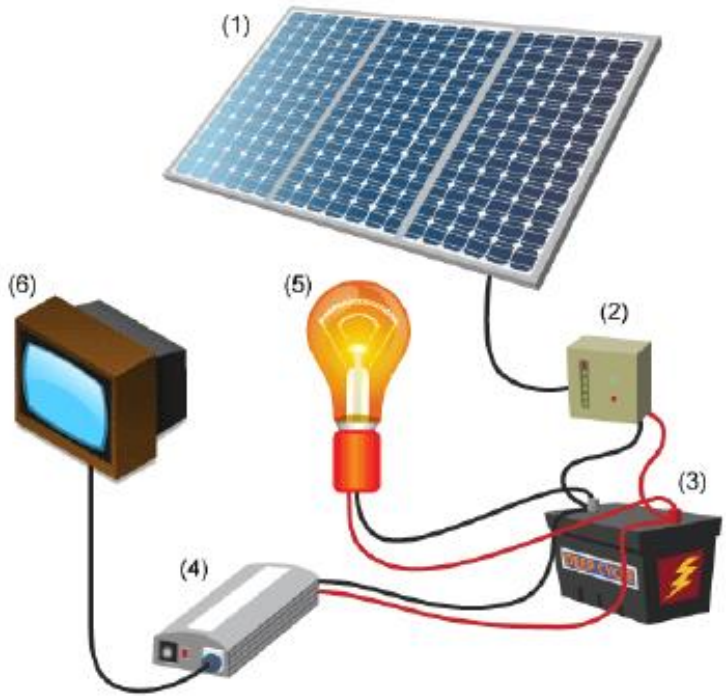


Figure.9 : Schéma typique d'un système PV photovoltaïque autonome.

Comme illustré sur la figure 9, un système PV autonome est constitué de :

- **un générateur PV** (1) permettant de générer l'énergie électrique nécessitant pour faire fonctionner les récepteurs,
- **des batteries** (3) qui sont utilisées pour stocker l'énergie électrique sous une forme chimique. en l'absence du Soleil cette énergie sera restituée et utilisée.
- **des charges** regroupant tous les récepteurs (appareils) électriques fonctionnant au courant continu (5) et au courant alternatif (6),
- **un onduleur** (4) permettant de convertir l'énergie électrique issue du générateur PV ou de batteries sous forme continue en courant et tension alternatifs nécessitants pour le fonctionnement des récepteurs électriques alternatifs,
- **un système de régulation**(2), Comprenant :
  - un régulateur de charge de batteries qui a pour fonction principale de protéger la batterie contre les surcharges et les décharges profondes. Il est un élément essentiel pour la durée de vie de la batterie,

- un convertisseur DC/DC dont le rôle est de faire l'adaptation entre la source PV et la charge pour un transfert de puissance maximal. Ceci est fait en maintenant le point de fonctionnement sur ou assez proche du MPP pour n'importe quelles conditions de fonctionnement (rayonnement, température, caractéristique de charge, etc.).
- un régulateur MPPT (Maximum Power Point Tracking) permet de faire fonctionner le générateur PV à sa puissance maximale.

IV.2. Le générateur PV :

Le générateur PV est la partie du système où se fait la conversion du rayonnement solaire en électricité. Cette conversion se fait au niveau des cellules. La puissance maximale que peut délivrer une cellule PV ne peut dépasser 2W. Par conséquent, pour pouvoir alimenter des équipements électriques, nécessitants souvent des puissances plus importantes, plusieurs cellules PV sont connectées en séries et/ou parallèles pour former un générateur, appelés panneaux PV ou modules PV, de puissances allons de quelques dizaines de Watts à quelques centaines de Watts. Pour des applications de grandes puissances ces modules sont assemblés en séries et/ou parallèles pour former des champs PV [13].

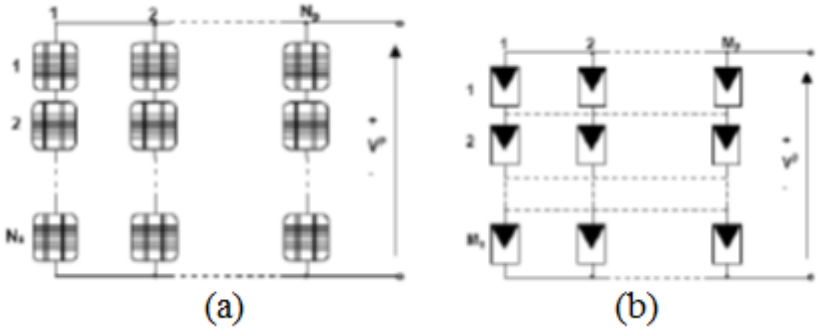


Figure.10 : (a) Module PV (plusieurs cellules), (b) Champs PV (plusieurs modules PV)

IV.2.a. Principe de Fonctionnement de la Cellule Photovoltaïque

Une cellule photovoltaïque (PV), aussi appelée photopile, est la juxtaposition de deux semi-conducteurs, l'un dopé P et l'autre dopé N. À la jonction des deux couches se forme un champ électrique. Ce champ électrique existe même si la cellule est dans l'obscurité. Sous un ensoleillement plus ou moins important, les photons ou grains de lumière, venant avec une énergie suffisante entrent en collision avec les atomes du crystal (figure 11.a). Ils parviennent

à faire passer les électrons de la bande de valence à la bande de conduction du matériau semi-conducteur, créant ainsi des paires d'électrons-trous. Ceux-ci, sous l'effet de la barrière de potentiel, vont s'accumuler sur chacune des faces extérieures des zones P et N [14].

Ainsi, une différence de potentiel entre les deux faces de la cellule est créée. Les grilles métalliques à l'avant et à l'arrière de la cellule PV collectent les électrons et les trous qui vont donc fournir à un circuit extérieur le courant électrique produit [13]. Si le photon est très énergétique, il ne peut tout de même extraire qu'un seul électron. L'énergie excédentaire est perdue en chaleur.

La zone N est couverte par une grille métallique qui sert de cathode, tandis qu'une plaque métallique (contact arrière) recouvre l'autre face du cristal et joue le rôle d'anode. L'épaisseur totale du cristal est de l'ordre du millimètre (figure2.4).

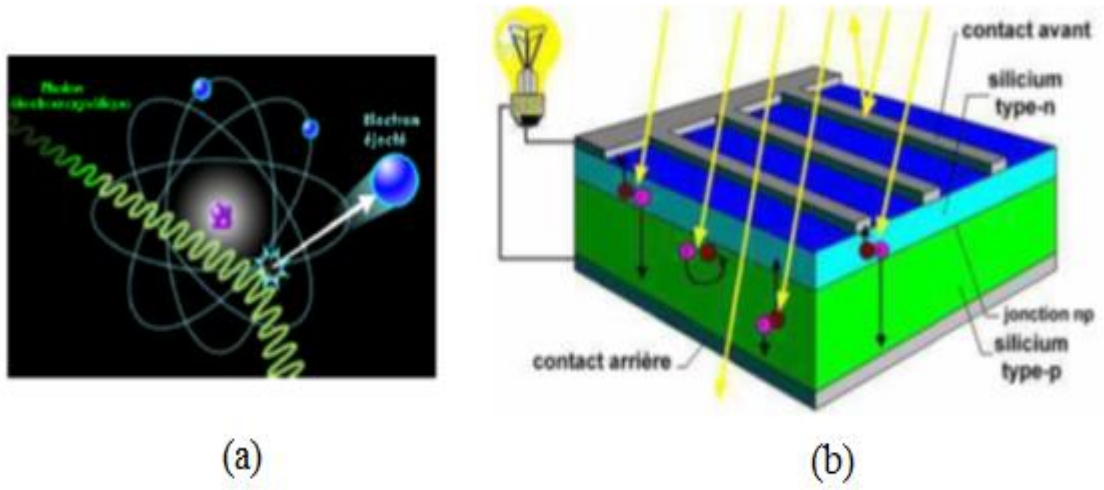


Figure.11 : (a) Collusion entre un photon et un atome, (b) Cellule photovoltaïque.

IV.2.b. modélisation d'une cellule photovoltaïque :

La figure (12) présente le schéma électrique équivalent d'une cellule photovoltaïque sous éclairement. Il correspond à un générateur de courant Iph monté en parallèle avec une diode. Deux résistances parasites sont introduites dans ce schéma pour représenter est la résistance shunt caractérisant les courants de fuite de la jonction(Rsh) et la résistance série représentant les diverses résistances de contacts et de connexions(Rs).

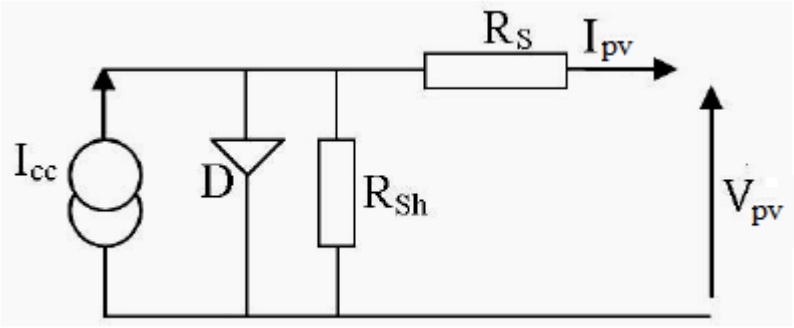


Figure.12 : Schéma équivalent électrique de la cellule PV

Le modèle mathématique pour la caractéristique courant-tension d'une cellule PV est donné par [14] :

$$I_{pv} = I_{ph} - I_{sat}\left[\exp\left(\frac{e\,(V_{PV}+(I_{pv}*R_s))}{nKT}\right) - 1\right] - \frac{V_{PV}+(I_{pv}*R_s))}{R_{sh}}$$

Où :

- Isat est le courant de saturation,
- K est la constante de Boltzmann (1,381 10-23 J/K),
- T est la température effective des cellules en Kelvin(K),
- e est la charge de l'électron (e=1,6 10-19 C),
- n est le facteur d'idéalité de la jonction (1< n<3),
- Ipv est le courant fourni par la cellule lorsqu'elle fonctionne en générateur,
- Vpv est la tension aux bornes de cette même cellule,
- Iph est le photo-courant de la cellule dépendant de l'éclairement et de la température,
- Rsh est la résistance shunt caractérisant les courants de fuite de la jonction,
- Rs est la résistance série représentant les diverses résistances de contacts et de connexions.

IV.2.c. Caractéristique courant-tension d'une cellule photovoltaïque :

Sous un éclairement donné, toute cellule photovoltaïque est caractérisée par une courbe courant-tension (I-V) représentant l'ensemble des configurations électriques que peut prendre la cellule. Trois grandeurs physiques définissent cette courbe :

- tension à vide (Vco): tension générée par une cellule éclairée non raccordée.
- courant court-circuit (Icc) : le courant généré par une cellule éclairée en court-circuit.
- point de puissance maximal (MPP) (en anglais : maximal power point) obtenu pour une tension et un courant optimaux : Vopt, Iopt (parfois appelés aussi Vmpp, Impp).

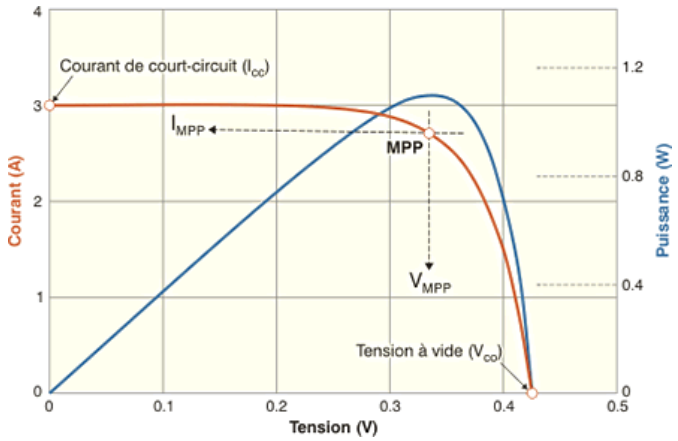


Figure.13 : Caractéristique I(V) d'une cellule PV.

Pour permettre une comparaison de l'efficacité de différentes cellules, on définit ces caractéristiques dans des conditions de test bien précises (STC = Standard Test Conditions). Ces conditions sont : émission lumineuse de 1 000 W/m², température de 25 °C, conditions spectrales Air Mass 1.5 (composition du spectre identique au spectre solaire lorsqu'il traverse une épaisseur et demie d'atmosphère, ce qui correspond à un angle d'incidence de 41.8° par rapport à l'horizontale) [15].

La caractéristique puissance–tension P(V) qu'on peut facilement calculer à partir de la I(V) peut être utiliser pour caractériser les performances d'une cellule PV.

IV.2.d. Effets des Variations Climatiques sur la caractéristique I(V) et P(V) d'une cellule PV :

Les figures suivantes présentent l'influence de la variation de l'éclairement et de la température sur la caractéristique électrique d'une cellule PV.

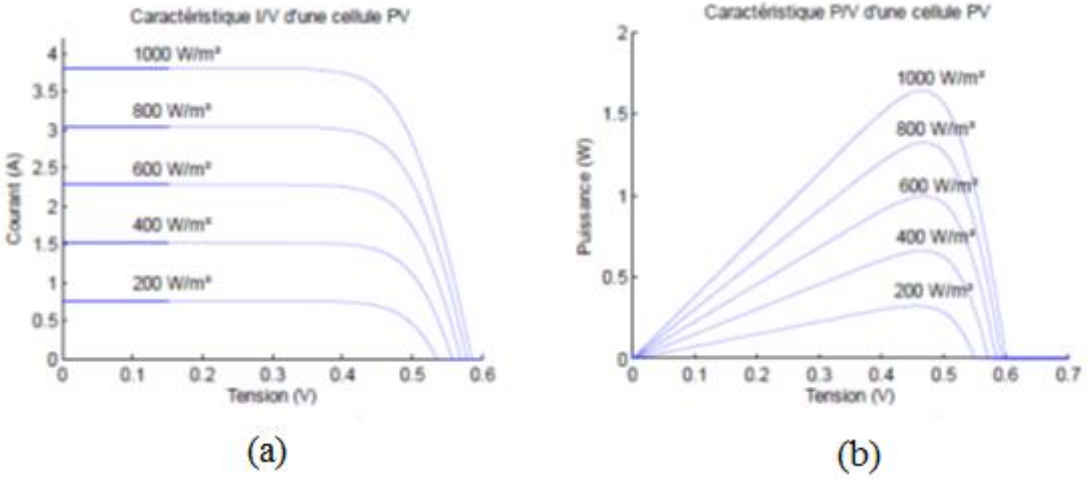


Figure.14 : (a) Caractéristique I(V) d'une cellule PV pour différents niveaux de rayonnement
(b) Caractéristique P(V) d'une cellule PV pour différents niveaux de rayonnement

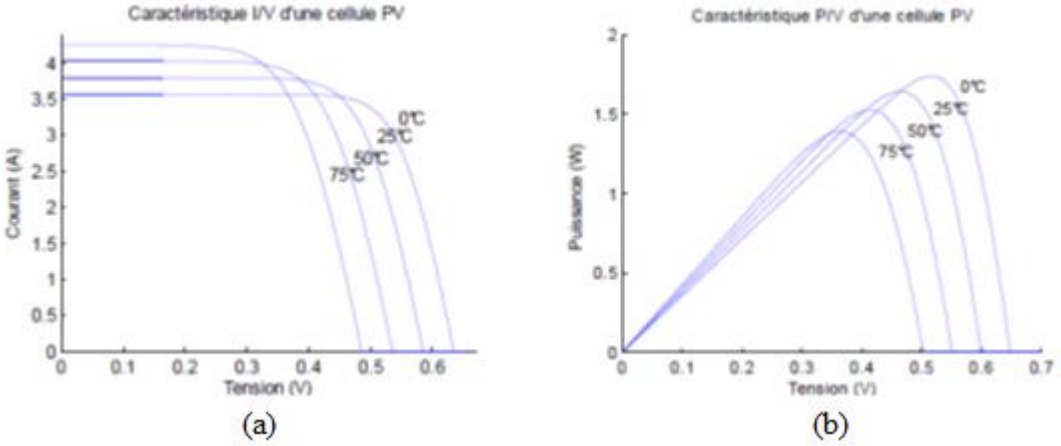


Figure.15 : (a) Caractéristique I(V) d'une cellule PV pour différentes températures,
(b) Caractéristique P(V) d'une cellule PV pour différentes températures.

A partir de ces figures on remarque que l'augmentation de l'éclairement implique une augmentation de l'énergie électrique générée et l'augmentation de la température implique la diminution de l'énergie générée.

IV.3. Régulateur MPPT :

Nous avons vue ci-dessus que pour chaque module PV donnée le point de sa puissance maximale (Vppm, Ippm) dépend de la valeur de l'éclairement solaire qu'il reçoit et de la température de ses cellules. Ainsi, pour assurer toujours le fonctionnement du module PV au

point de puissance maximale (i.e. rendement maximal) on utilise souvent dans les systèmes PV un régulateur MPPT qui fait varier en fonction des variations des conditions météorologiques (éclairement et température) le point de fonctionnement (Vref, Iref) pour le faire coïncider avec le point de puissance maximale (Vppm, Ippm) [16].

L'algorithme le plus utilisé par les régulateurs MPPT est l'algorithme P&O (perturbation and observation) dont l'organigramme est donné en figure suivante.

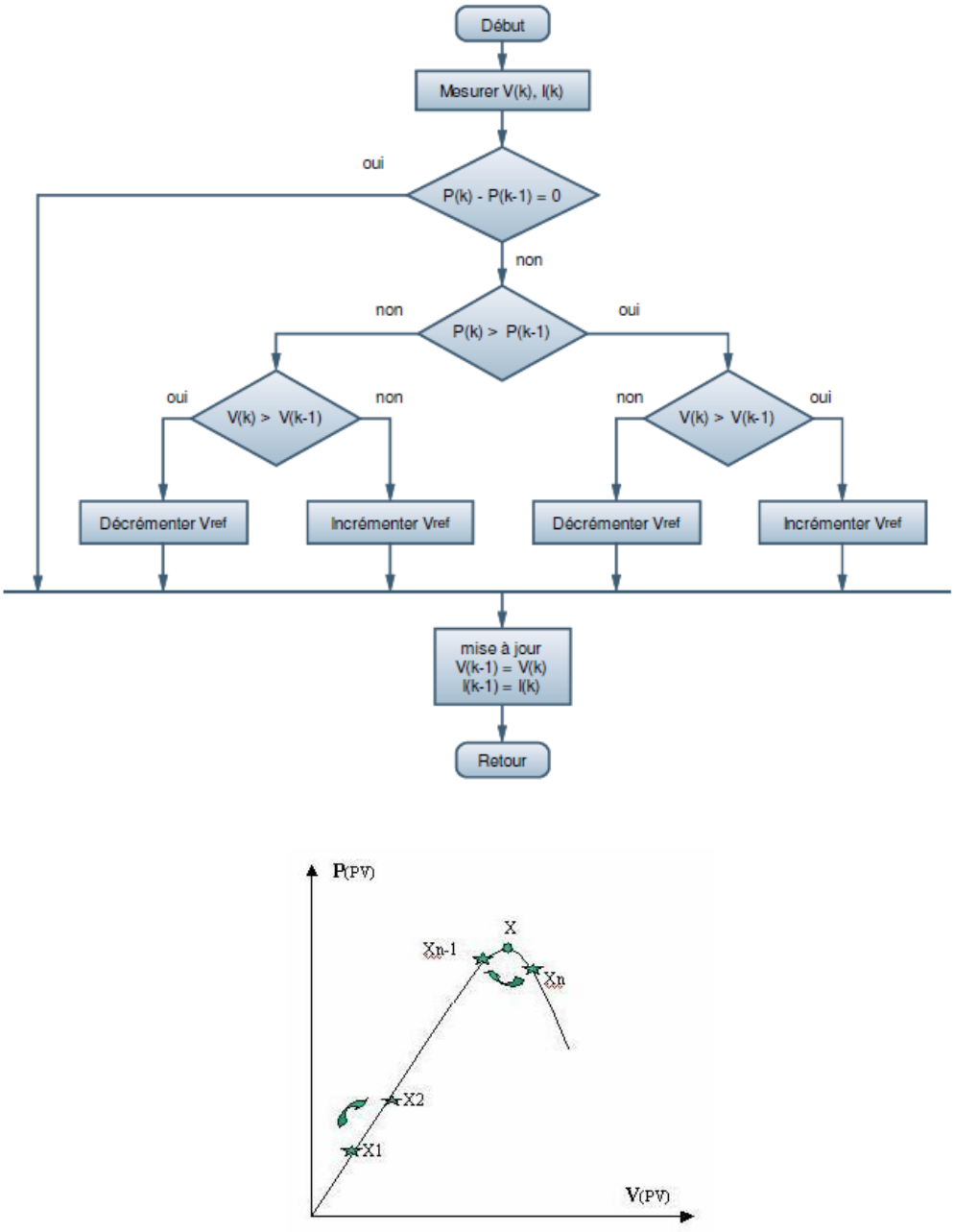


Figure.16 : Principe du contrôleur MPPT.

CHAPITRE II

Conception et réalisation matérielle du système de gestion de l'éclairage

I. Introduction :

Ce chapitre est dédié à la description de la partie matérielle du système de gestion d'éclairage que nous avons réalisé. Tout d'abord nous avons présenté son fonctionnement. Puis nous avons donné le schéma bloc et la descriptions différent blocs qui le composent. A la fin nous avons présenté la partie réalisation pratique.

II. Fonctionnement du système de gestion d'éclairage :

Comme illustré sur l'organigramme de la figure 1, le principe mis en œuvre pour la gestion d'éclairage repose sur le fait que, pour assurer le confort lumineux nécessaire dans une pièce donnée de la maison, la lampe servant à éclairer cette pièce n'est allumée que lorsqu'une personne est présente dans cette pièce et que la lumière du jour, rentrant par les fenêtres, ne suffit pas à assurer ce confort lumineux.

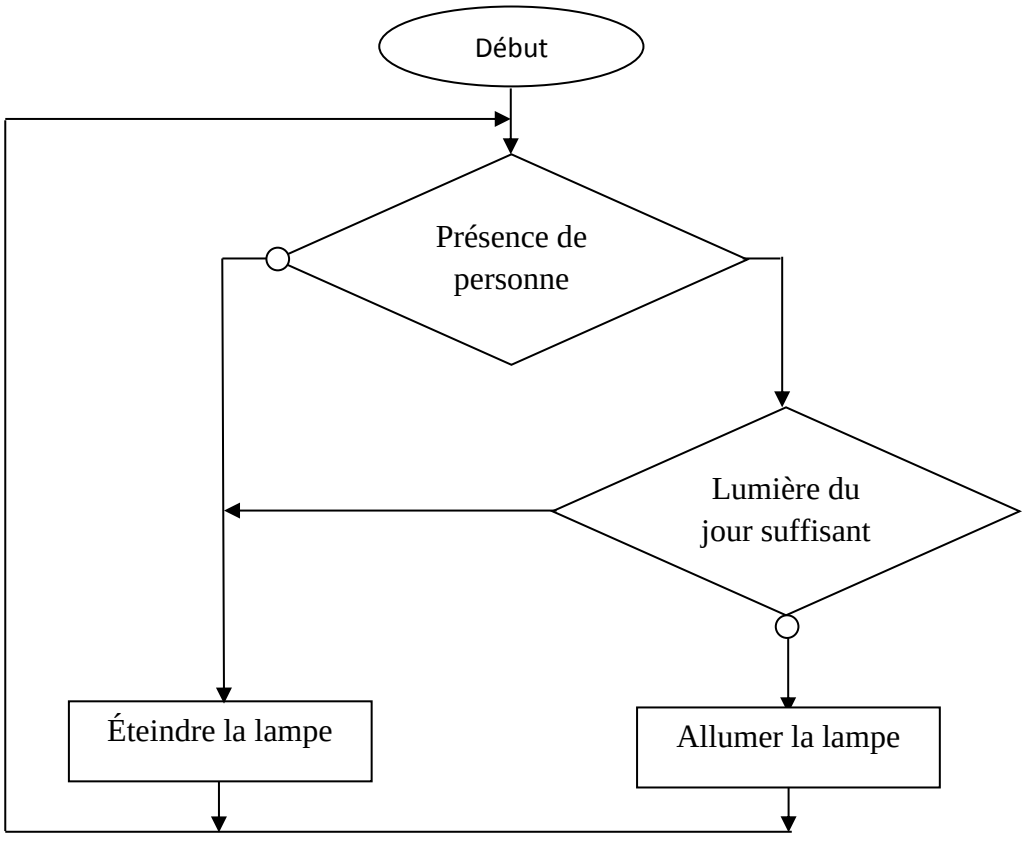


Figure.1 : principe de fonctionnement de la gestion d'éclairage.

Pour réaliser ce principe de fonctionnement le système de gestion d'éclairage utilise :

- Dans chacune des quatre pièces de la maison, un détecteur de présence qui se mat à l'état ON (fermer) s'il y a présence de personne et s'il n'y a pas de lumière suffisante dans la pièce et se met à l'état OFF (ouvert) dans le cas contraire,
- Un capteur de lumière qui mesure le niveau d'éclairement à l'extérieur de la maison pour savoir s'il fait jour (présence de lumière naturelle) ou il fait nuit (absence de lumière naturelle).
- Des volets motorisés pour les fenêtres du salon et les fenêtres de la chambre qu'on peut ouvrir ou fermer par l'envoie d'une commande électrique.

Le fonctionnement du système de gestion d'éclairage dans le salon et la chambre est décrit par l'organigramme de la figure 2 suivante.

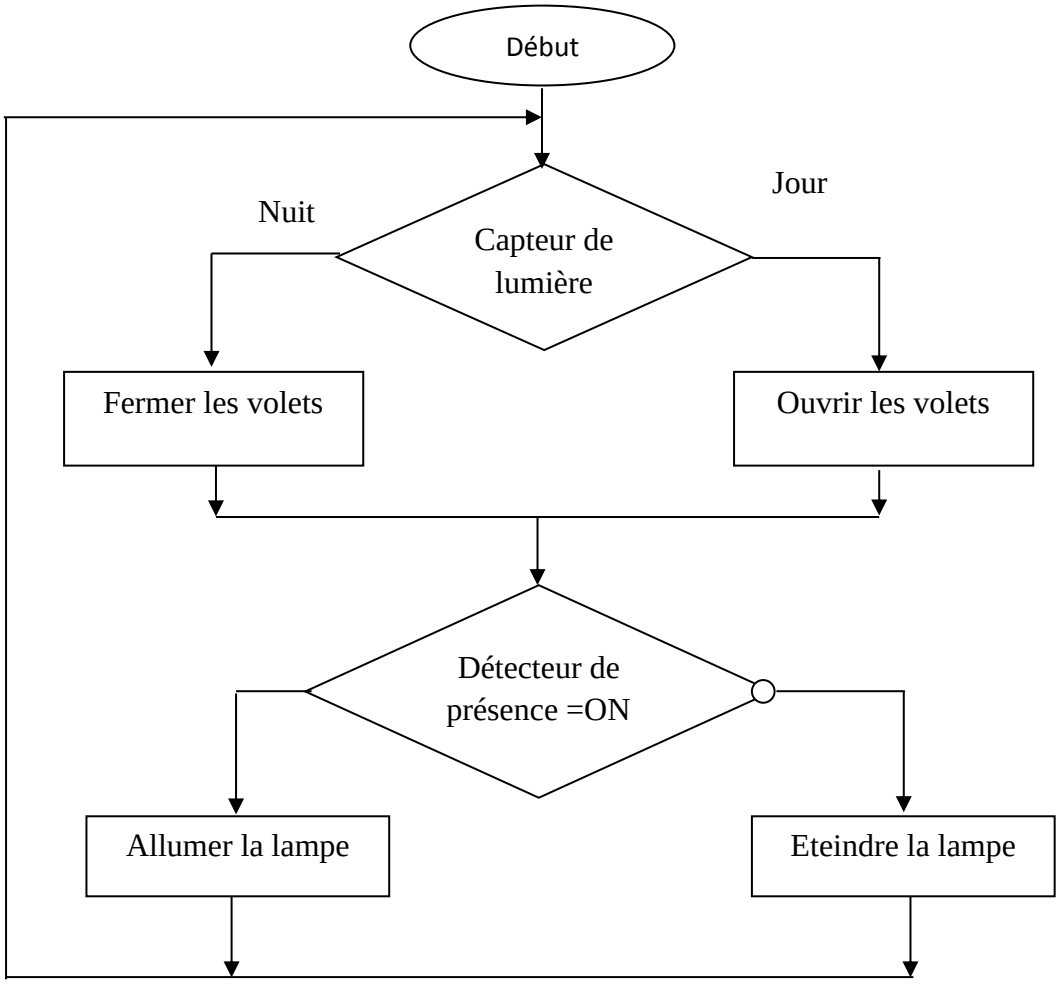


Figure.2 : Organigramme du fonctionnement du système de gestion d'éclairage dans le salon et la chambre.

Le fonctionnement du système de gestion d'éclairage dans la cuisine et salle de bain est décrit par l'organigramme de la figure suivante.

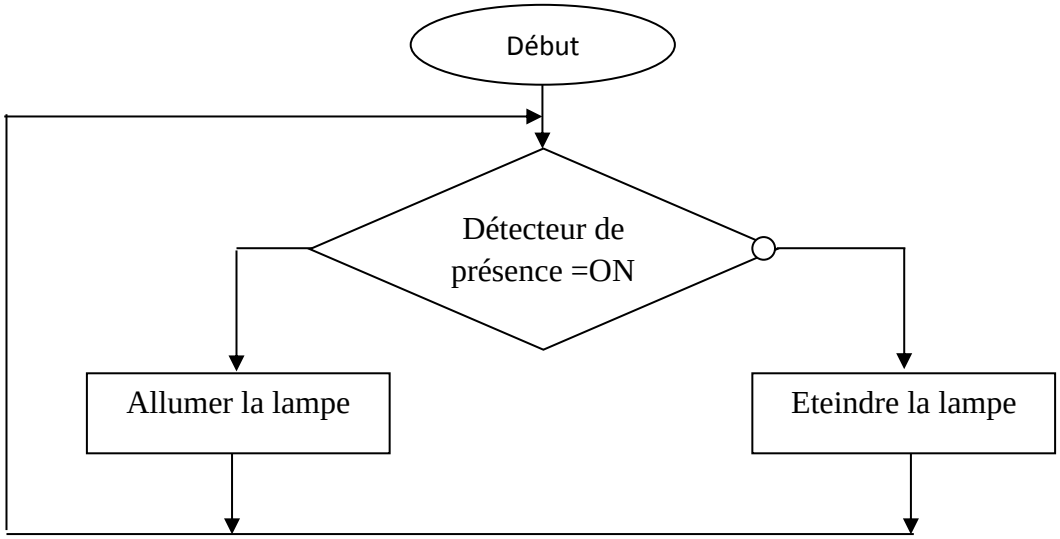


Figure.3 : Organigramme du fonctionnement du système de gestion d'éclairage dans la salle de bain et de la cuisine.

III. Schéma bloc du système de gestion d'éclairage :

Ce système est composé de cinq blocs principaux (figure.4),

a- Le bloc détecteurs et capteur, comprenant les détecteurs de présence et le capteur de lumière délivrant l'information de présence de personne et de lumière dans les pièces de la maison et le niveau d'éclairement à l'extérieur de la maison.

b- Le bloc de commande composé d'une carte à microcontrôleur de commande et des circuits interfaces de commande.

c- Le bloc d'éclairage réalisé par les circuits d'éclairage.

d- Bloc d'affichage et supervision comprenant une application logicielle installée sur un PC permettant d'afficher l'état du système de gestion d'éclairage.

e- Bloc du système PV qui génère l'énergie électrique nécessaire pour le fonctionnement du système de gestion d'éclairage.

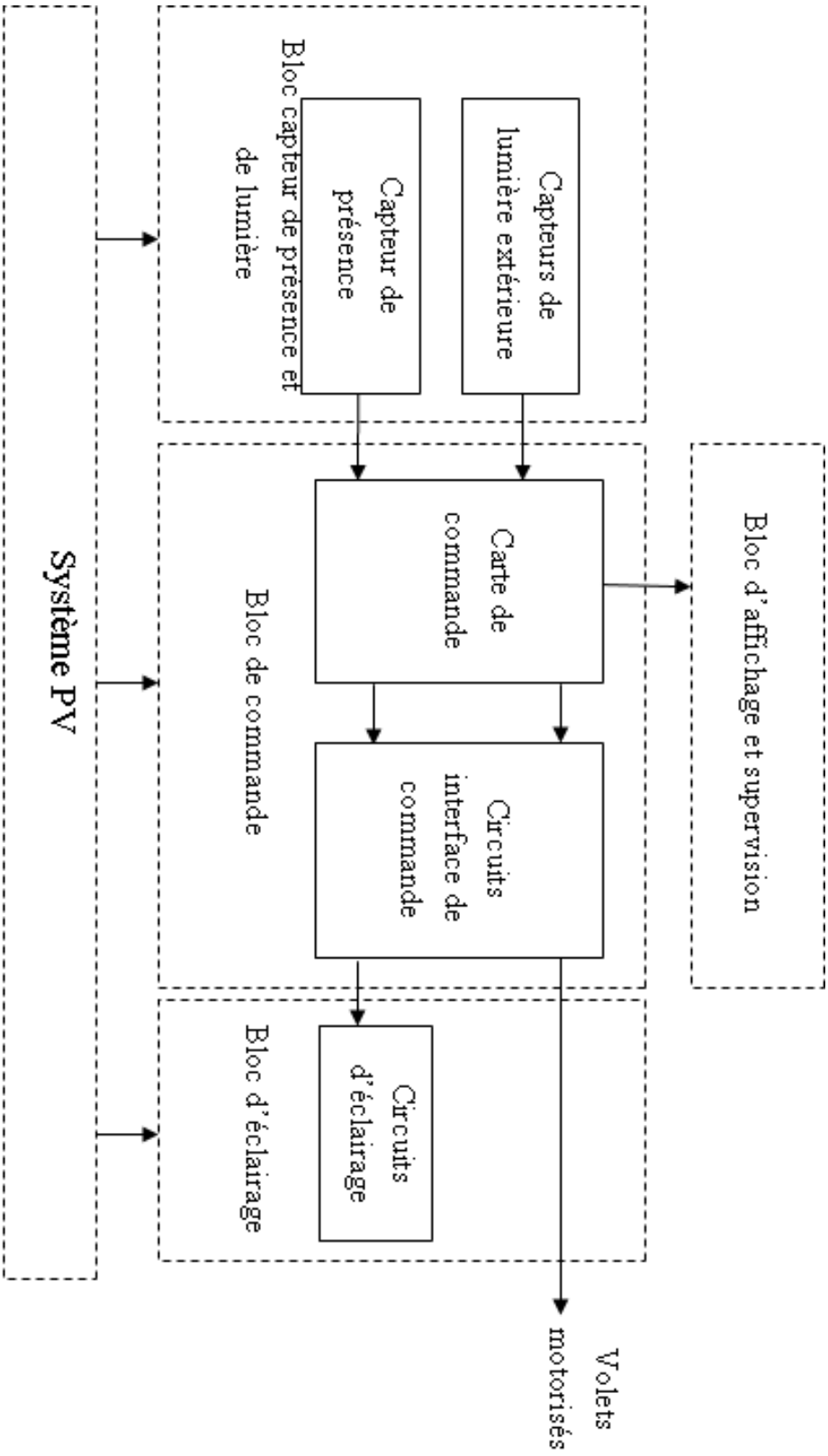


Figure.4 : Schéma bloc du système de gestion d'éclairage.

III.1. Le bloc détecteurs et capteur :

Ce bloc comprend :

- **Capteur de lumière externe :**

Ce Capteur de lumière externe (figure.5) est un circuit basé sur l'utilisation d'une photorésistance LDR (Light Dependent Resistor) qui mesure le niveau d'éclairement reçus par la LDR par la mesure de la résistance de la LDR. Le schéma de principe d'un tel circuit est donné en figure 5 suivante.

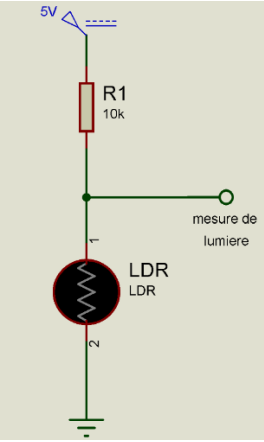


Figure. 5 : Circuit de principe du capteur de mesure de lumière

Dans ce schéma la tension Vs mesurée est définit par :

$$V_s = \frac{5 * R_{LDR}}{R_1 + R_{LDR}}$$

La valeur maximale de la résistance de la LDR utilisée qui correspond à la valeur mesurée à l'obscurité ne dépasse pas 100 Ohm

D'où : $R1+Rldr \approx R1$

D'où :

$$V_s = \frac{5 * R_{LDR}}{R_1}$$

D'où : la tension mesurée est proportionnelle à la résistance de la LDR qui est elle-même proportionnelle à la lumière qu'elle reçoit.

❖ **Les détecteurs de présence :**

Les détecteurs de présence utilisé pour notre système d'éclairage son des détecteurs de présence de système d'éclairage qui correspondent à des interrupteurs qui se mettent à l'état ON (fermé) en présence de personne et absence de lumière et se mettent à l'état OFF(ouvert) un certain temps d'attente (temps d'attente pour passage à l'état OFF) après une détection de présence de lumière et/ou absence de personne dans le périmètre de détection du détecteur.

La détection de présence ou d'absence de la lumière est réalisée par l'utilisation d'un circuit utilisant une LDR. Ce circuit permet de détecter si la lumière présente dans l'environnement du détecteur (i.e. reçues par la LDR) est supérieure ou inférieure à une valeur seuil prédéfinit (seuil de détection de lumière). Le schéma d'un tel circuit peut être donné comme sur la figure suivante.

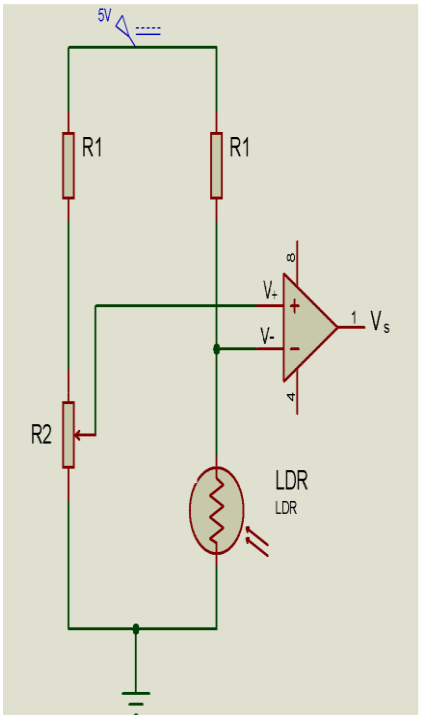


Figure. 6 : Schéma électrique d'un circuit de détection de présence de lumière

Dans ce circuit on a :

$$V_s = 5V \quad \text{Si} \quad V_+ > V_-$$

Et :

$$V_s = 0V \quad \text{Si} \quad V_+ < V_-$$

D'où :

$$V_{-} = \frac{5 * R_{LDR}}{R_1 + R_{LDR}} \approx \frac{5 * R_{LDR}}{R_1} \qquad R_{LDR} << R_1$$

$$V_{+} = \frac{5 * R_2}{R_1 + R_2} \approx \frac{5 * R_2}{R_1} \qquad R_2 << R_1$$

D'où : la sortie de ce circuit est de 5V si la lumière est inférieure à un seuil fixé par la valeur de la résistance variable R2 et elle est de 0V dans le cas contraire.

La détection de présence de personne est réalisée par un circuit de détection de mouvement basé sur un capteur infrarouge PIR qui détecte le mouvement de personnes présentes dans son périmètre de détection.

Le détecteur de présence utilisé est équipé de deux résistances ajustables pour régler le seuil de détection de lumière et le temps attente de passage à l'état OFF (figure. 7)

.

Figure. 7 : Détecteur de présence avec potentiomètres de réglage de temps d'attente de passage à l'état OFF et du seuil de présence (absence) de lumière).

III.2. Le bloc d'éclairage :

Ce bloc contient le circuit d'éclairage de la maison. Il est composé d'un circuit simple allumage pour chaque pièce de la maison (figure. 8).

Figure. 8 : Le schéma de ce circuit d'éclairage

III.3. Le bloc de commande :

Ce bloc est composé de :

a- Une carte Arduino Mega à laquelle sont connectées

- les sorties des détecteurs de présence et du capteur de lumière externe
- les entrées de commandes des lampes et d'ouverture/ fermeture des volets des fenêtres de la chambre et du salon
- le câble USB permettant de réaliser la connexion série entre la carte et le bloc d'affichage et supervision (PC + application logicielle)

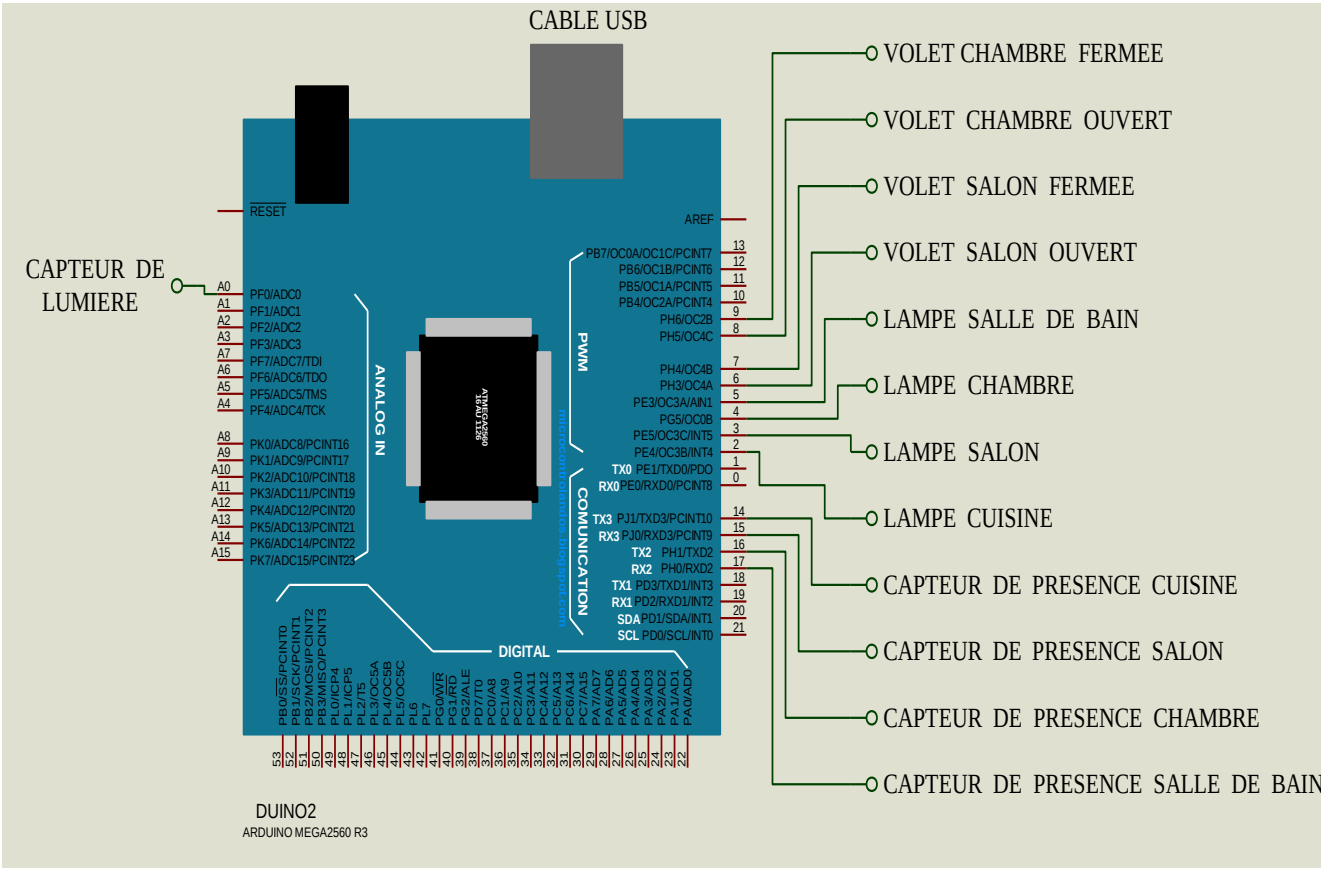


Figure. 9 : Connections de la carte Arduino.

Le rôle de cette carte est de générer les signaux de commande d'ouverture et de fermeture des volets motorisé des fenêtres et les signaux de commande ON/OFF des lampes des circuits d'éclairage conformément aux signaux quelle reçois des circuits de détection de présence et de lumière et au fonctionnement représenté par les organigrammes des figures 2 et 3.

b- Les circuits interface de puissance permettant d'adapter la puissance des signaux de commande issus de la carte Arduino aux entrées de commande des lampes et des volets motorisés des fenêtres. Ces circuits interfaces sont réalisés par une carte à relais commandés par le circuit intégré ULN2803 [17] (figure 10).

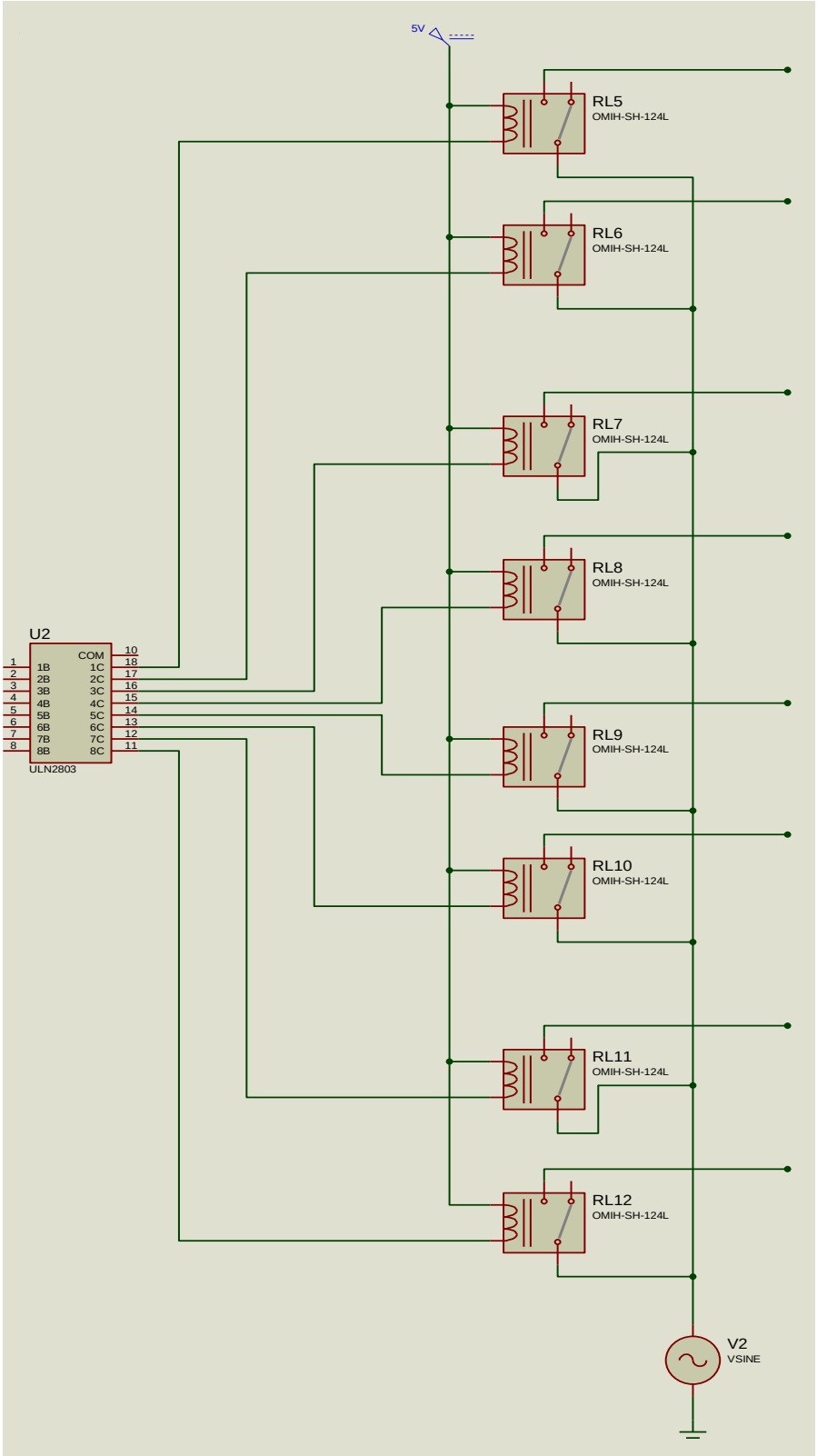


Figure.10 : Circuits interface de puissance.

III.4. Un système PV :

Il correspond à un système PV autonome permettant de générer l'énergie électrique nécessaire au fonctionnement du système de gestion d'éclairage. Ce système comprend

- Deux modules PV de 70W
- Un régulateur ayant les fonctions
  - Convertisseur de puissance DC/DC,
  - Régulateur de charge de la baterie
  - Régulateur MPPT.
- Une batterie de 12V, 100A/h
- Un onduleur 12VDC/220VAC

IV. Simulation sur Isis Proteus :

Après conception du circuit électrique du système de gestion d'éclairage et écriture du programme arduino de la carte de commande nous avons vérifié et validé par simulation sur Isis Proteus le bon fonctionnement de l'ensemble comportant le bloc capteurs et detecteurs, bloc de commande et bloc d'éclairage. A cet effet, le schéma électrique suivant à été réalisé sur Isis Proteus.

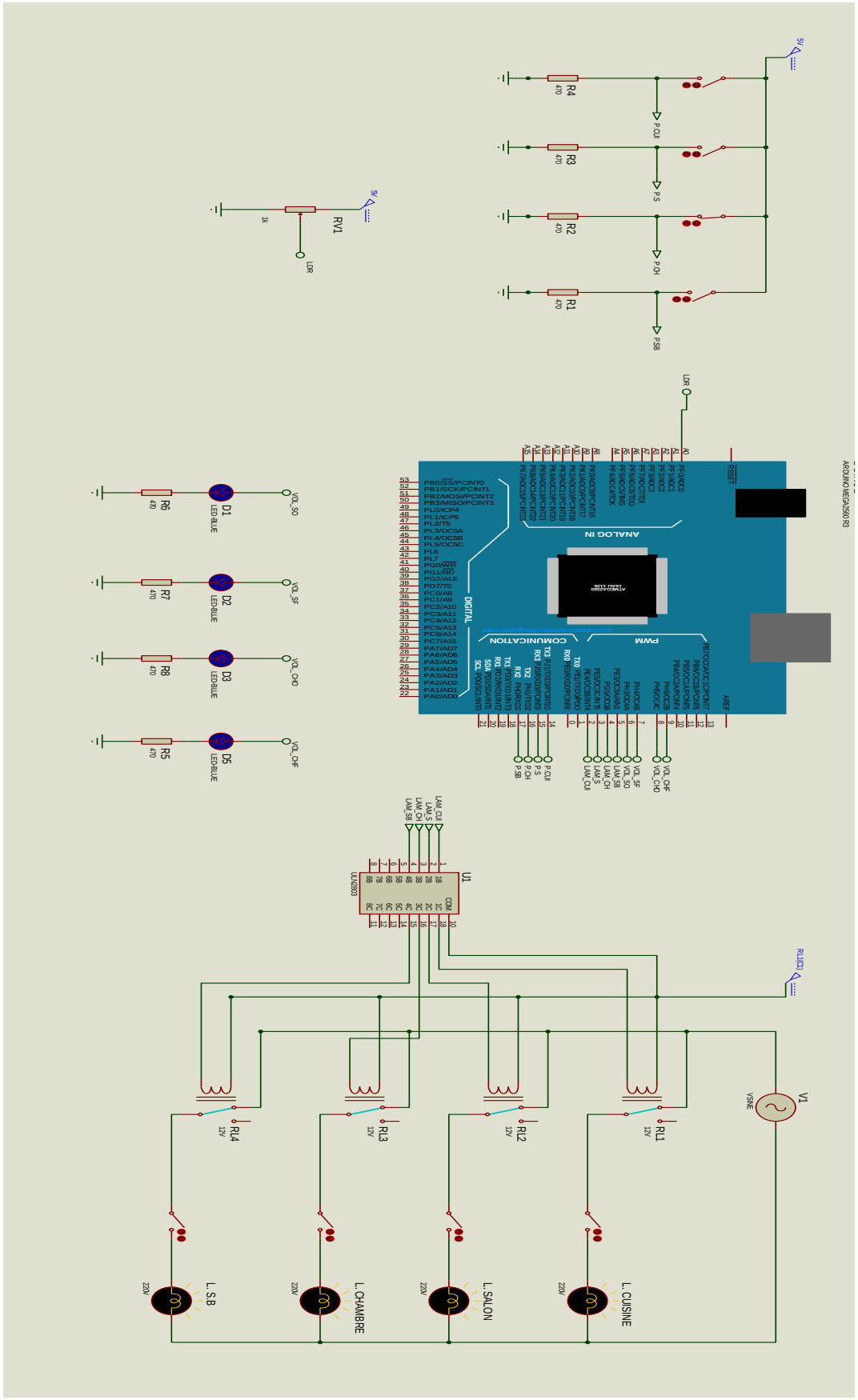
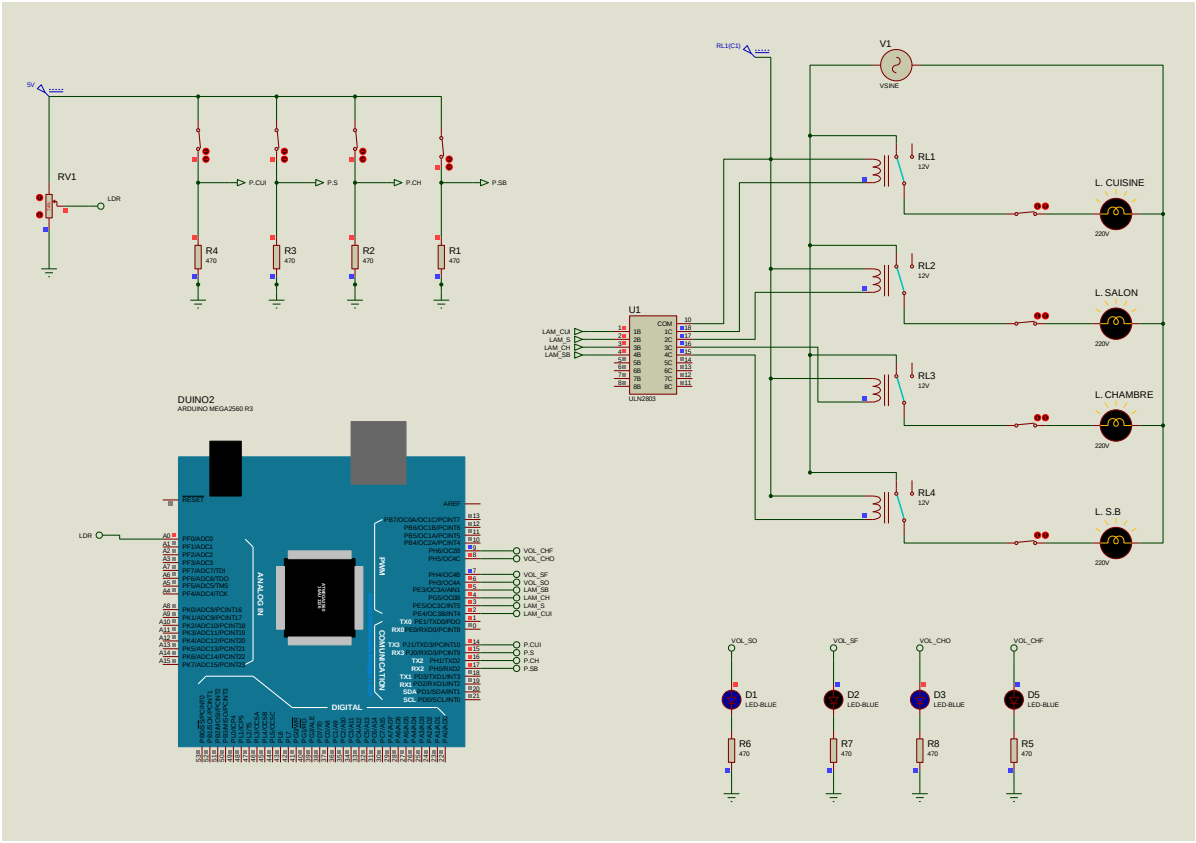


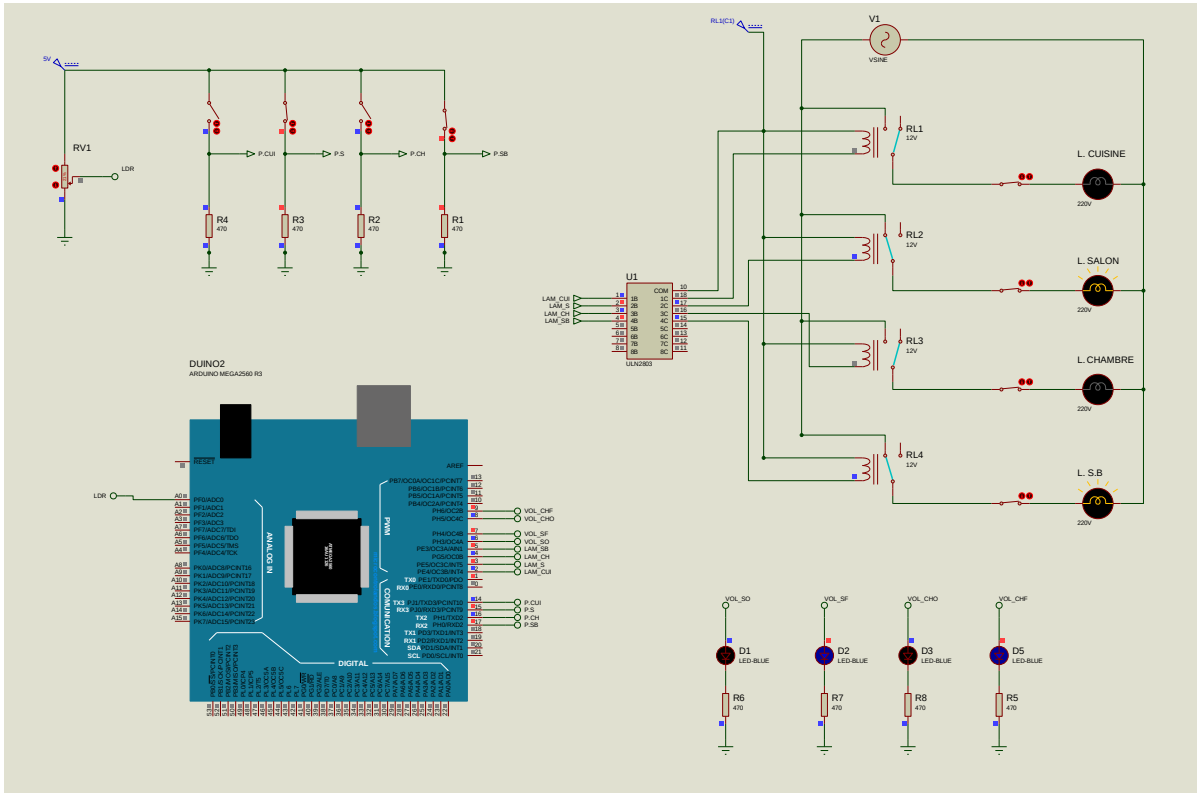
Figure 11 : Schéma de simulation réalisé sur Isis Proteus

Dans ce schéma,

- les sorties de commande des volets motorisés sont représentées par des Leds une led pour l'ouverture et une autre led pour la fermeture de chacun des deux volets motorisés.
- Le capteur de lumière externe est représenté par un potentiomètre.
- Les détecteurs de présence sont représentés par des interrupteurs à deux états.

Les schémas suivants obtenus après simulations de différentes situations démontrent le bon fonctionnement de circuit simulé.





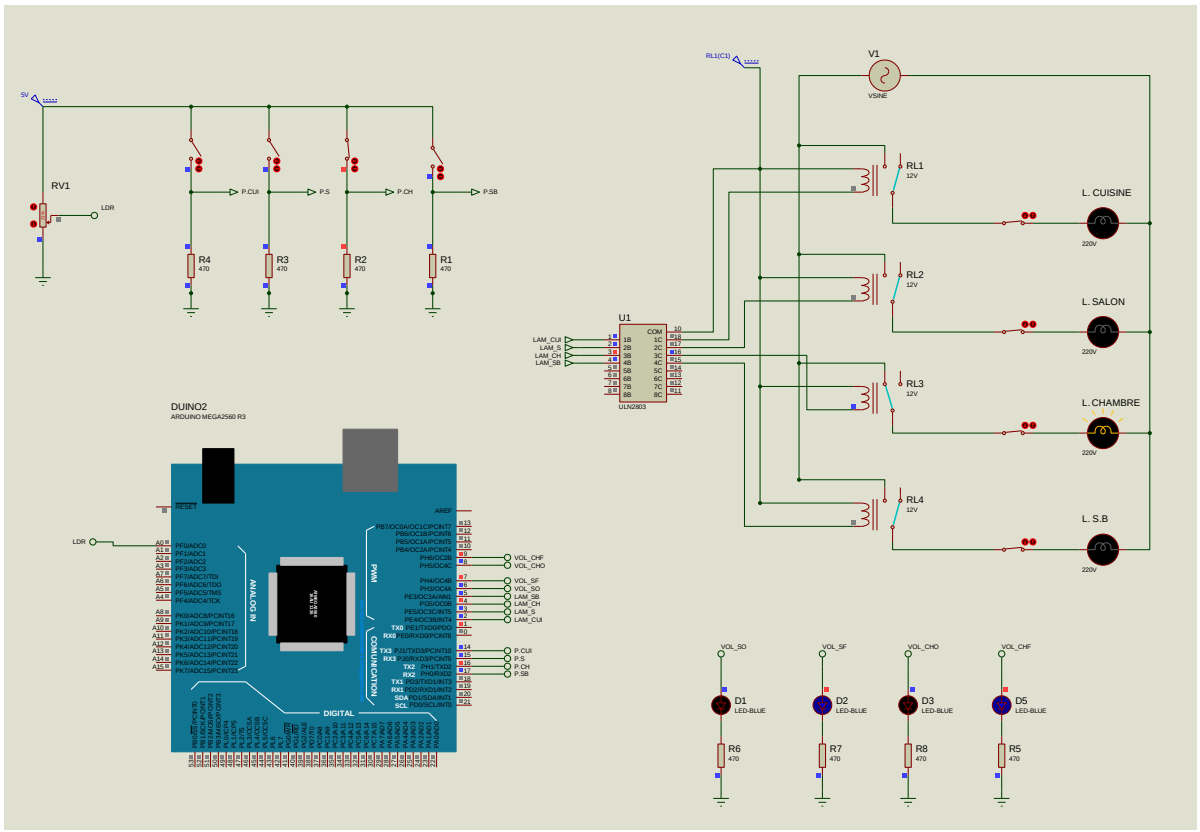


Figure. 12 : Résultats de simulation sous Isis Proteus

V. Réalisation Pratique :

A défaut d'installer pratiquement notre système de gestion d'éclairage dans une véritable maison nous l'avons réalisé sur un tableau représentant une maquette d'une maison comportant quatre pièces (chambre, cuisine, salon et salle de bain). D'un côté du tableau nous avons mis les Lampes LED, les interrupteurs, les détecteurs de présence, le capteur de lumière et quatre petites lampes pour représenter les états d'ouverture et fermeture des volets des fenêtres du salon et de la chambre. De l'autre côté du tableau nous avons installé la carte Arduino mega, la carte du circuit interface de commande, et le câblage (voir figure 13).

Figure. 13 : Circuit de gestion d'éclairage réalisé

Apres réalisation du système de gestion d'éclairage nous l'avons testé dans plusieurs situations possibles et les résultats ont été satisfaisants.

CHAPITRE III

Développement logiciel

I. Introduction :

Après la présentation de la partie conception et réalisation de la partie matérielle de notre système. Dans ce chapitre nous allons présenter la partie soft de notre système de gestion d'éclairage. En effet nous allons présenter dans ce qui suit nous allons présenter les trois applications logicielles que nous avons développée dans le de la réalisation du système de gestion d'éclairage. La première correspond au programme implémenté dans la carte Arduino traduisant le fonctionnement du système décrit précédemment et écrire sur le port série l'état des sorties des détecteurs de présence de capteur de lumière. La deuxième application est le programme LabVIEW permettant de lire les informations écrites sur le port série de la carte Arduino et d'afficher sur une interface graphique l'état des détecteurs de présence, des volets et des lampes de la maison. La dernière application est une publication web de l'interface graphique de l'application LabVIEW, permettant de télé-surveiller à distance l'état du système de gestion d'éclairage.

II. Description du Programme Arduino :

Le programme implémenté dans la carte Arduino sert, à lire les états des détecteurs et capteurs et commander les lampes et les volets de tel manière à réaliser le fonctionnement du système de gestion d'éclairage décrit précédemment d'une part, et à écrire sur le port série l'état des sorties des détecteurs de présence de capteur de lumière pour être lue par l'application LabVIEW d'une autre part. Ce programme est constitué de trois parties principales :

Partie 1 : déclaration des constantes et variables :

Dans cette partie est réalisé la déclaration des constantes et variables du programme. Les constantes sont utilisées pour définir les connexions des broches de la carte Arduino et les variables sont utilisées pour définir les états des quatre détecteurs de présence et du capteur de lumière externe.

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
constint LDR = A0;      // la LDR est reliée à la broche analogique A0
constint LAM_CUI = 2;   // la LAM_CUI est reliée à la broche digitale 2
constint LAM_S = 3;     // la LAM_S est reliée à la broche digitale 1
constint LAM_CH = 4;    // la LAM_CH est reliée à la broche digitale 0
constint LAM_SB = 5;    // la LAM_SB est reliée à la broche digitale 3
constint VOL_SO = 6;    // le VOL_SO est relié à la broche digitale 6
constint VOL_SF = 7;    // le VOL_SF est relié à la broche digitale 7
constint VOL_CHO = 8;   // le VOL_CHO est relié à la broche digitale 4
constint VOL_CHF = 9;   // le VOL_CHF est relié à la broche digitale 5
```

```
constint PCUI = 14;     // le capteur de présence dans la cuisine est relié a la broche digital 24
constint PS = 15;       // le capteur de présence dans le salon est relié a la broche digital 23
constint PCH = 16;      // le capteur de présence dans la chambre  relié a la broche digital 22
constint PSB = 17;    // le capteur de présence dans la salle de bain relié à la broche digital 25
int LUM,CCUI,CCH,CS,CSB,L;
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Partie 2 : la boucle void setup ():

Cette partie comprend les instructions qui sont exécutées une seule fois au démarrage. Dans cette partie on trouve :

- La configuration des pins de la carte Arduino en entrées ou en sorties (LAM_CUI, LAM_S, LAM_CH, LAM_SB, VOL_SF, VOL_CHO, VOL_CHF sont configurés en sortie et LDR, PCUI, PS, PCH, PSB sont configurés en entrés)
- Le choix de la vitesse de transmission du port série.

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void setup() {

Serial.begin (115200);

pinMode(LAM_CUI, OUTPUT); // configurer la broche en sortie
pinMode(LAM_CH, OUTPUT); // configurer la broche en sortie
pinMode(LAM_S, OUTPUT); // configurer la broche en sortie
pinMode(LAM_SB, OUTPUT); // configurer la broche en sortie
pinMode(VOL_CHO, OUTPUT); // configurer la broche en sortie
pinMode(VOL_CHF, OUTPUT); // configurer la broche en sortie
pinMode(VOL_SO, OUTPUT); // configurer la broche en sortie
pinMode(VOL_SF, OUTPUT); // configurer la broche en sortie
pinMode(PCUI, INPUT); // configurer la broche en entrée
pinMode(PCH, INPUT); // configurer la broche en entrée
pinMode(PS, INPUT); // configurer la broche en entrée
pinMode(PSB, INPUT); // configurer la brocheen entrée

 }
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Partie 3 : la boucle void loop () :

Cette partie comprend l'ensemble des instructions qui s'exécutent d'une manière continue. Ces instructions réalisent :

- La lecture de l'état de sortie du capteur de lumière et des détecteurs de présence.
- Ecriture sur le port série l'état de sortie du capteur de lumière et des détecteurs de présence.
- Comparer l'intensité de la lumière mesurée par la LDR à un seuil prédéfinit (seuil de présence de lumière externe) et conclure s'il y a lumière externe suffisante ou non.
- Allumer ou éteindre les lampes selon le résultat de sortie des capteurs de présence

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
voidloop() {

// Lire les valeurs des détecteurs de présence et du capteur de lumière
  LUM = analogRead(LDR);   // LUM prend la sortie de la LDR
  CCUI = digitalRead(PCUI);  //CCUI prend la sortie de PCUI
  CCH = digitalRead(PCH);    // CCH prend la sortie de PCH
  CS = digitalRead(PS);          // CS prend la sortie de PS
  CSB = digitalRead(PSB);      //CSB prend la sortie de PSB

// Ecrire les valeurs des détecteurs de présence et du capteur de lumière externe sur le port
série.
Serial.print ("L=");
Serial.print (L);
Serial.print ("CCUI=");
Serial.print (CCUI);
Serial.print ("CS=");
Serial.print (CS);
Serial.print ("CCH=");
Serial.print (CCH);
Serial.print ("CSB=");
Serial.println (CSB);

 // Définir s'il fait jour ou il fait nuit
if(LUM>500) {
    L=1;}
else{
    L=0;
   }
// ouvrir ou fermer les volets des fenêtres de la chambre et du salon
digitalWrite(VOL_CHO,L);
digitalWrite(VOL_CHF,!L);
digitalWrite(VOL_SO,L);
digitalWrite(VOL_SF,!L);

// Allumer ou éteindre les lampes de la cuisine, salon, chambre et salle de bain
digitalWrite(LAM_CUI,CCUI);
digitalWrite(LAM_S,CS);
digitalWrite(LAM_CH,CCH);
```

```
digitalWrite(LAM_SB,CSB);

delay(800);

}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Les trois parties du programme Arduino peuvent être résumée par l'organigramme suivant :

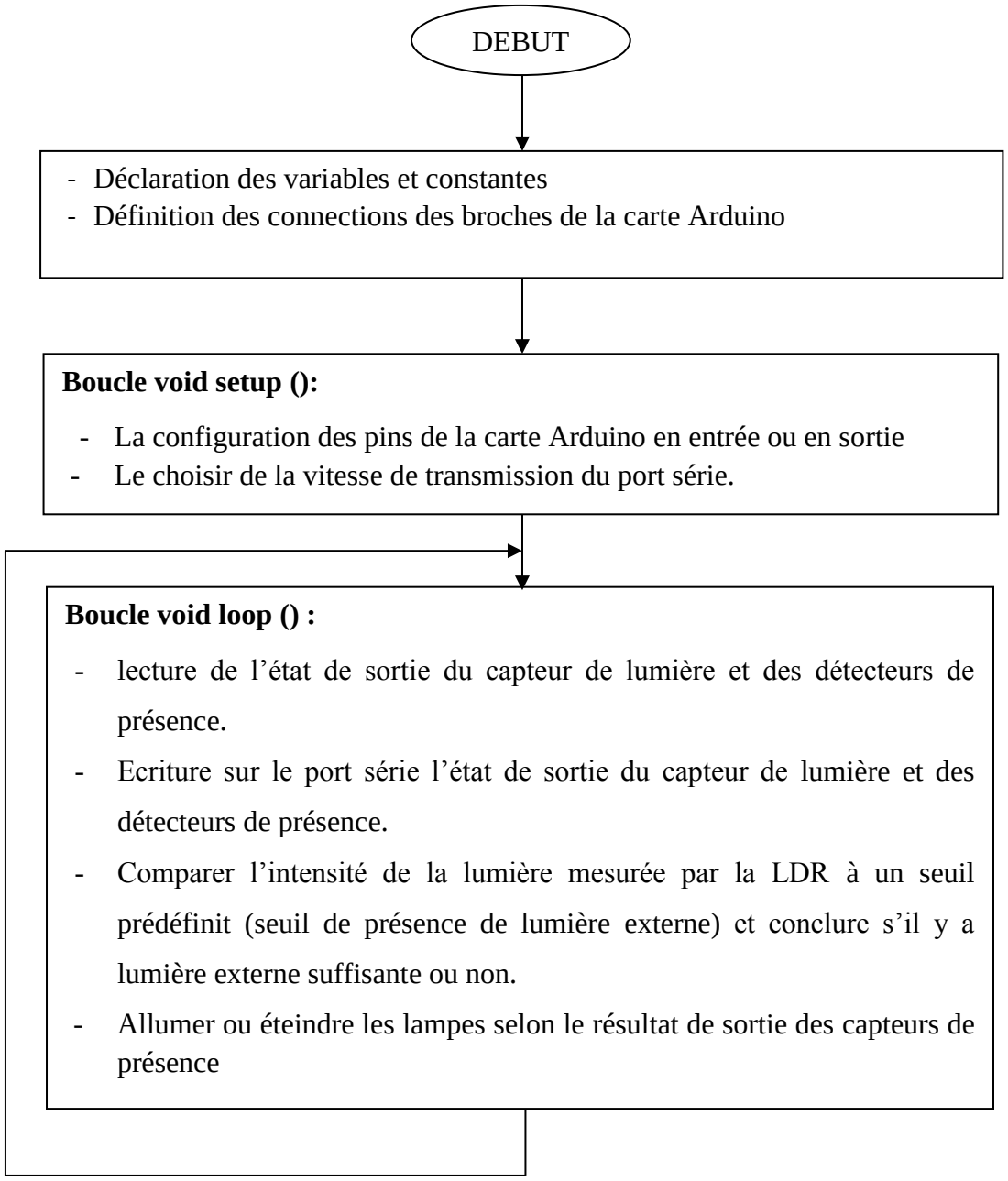


Figure.1 : Organigramme du programme de la carte Arduino.

III. Description de l'application LabVIEW :

Comme toute application LabVIEW, l'application que nous avons développé sous cet environnement est composée d'une face avant et d'un diagramme.

III.1. Le diagramme de l'application LabVIEW :

Comme illustré en figure (6) représentant le diagramme du VI de l'application réalisée, le programme de ce VI est réalisé principalement en quatre étapes :

- **Etape 1 : Configuration de la connexion série (VISA Config) :**

Configurer les paramètres de la connexion série permettant d'importer les données de la carte Arduino via le port USB. Cette configuration se fait grâce à une fonction LabVIEW nommée ' VISA Configure Serial Port'. Cette fonction est le cœur du système d'acquisition de notre interface sous LabVIEW [18].

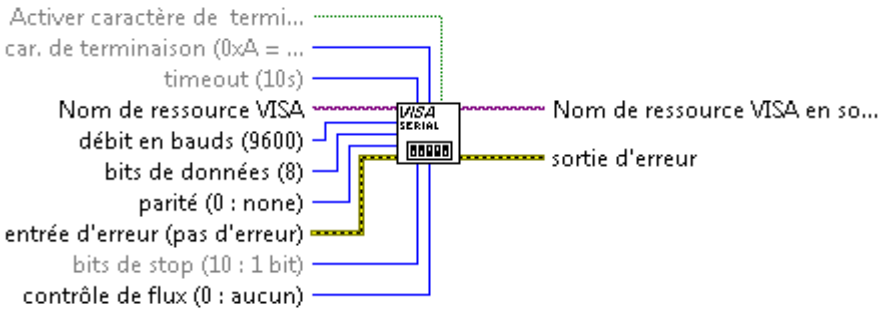


Figure.2 : La fonction 'VISA Configure Serial Port'.

Comme on peut le voir sur la figure 4 la configuration de cette fonction revient à renseigner les valeurs de plusieurs paramètres. Dans notre cas, pour le paramètre nom de ressource VISA nous avons donné l'emplacement du Port USB du PC auquel est connecté la carte Arduino (COM4). Pour le paramètre débit en bauds nous avons introduit la valeur de la vitesse de transmission de données que nous avons introduit même dans le programme Arduino. Pour les autres paramètres nous avons laissé les valeurs par défaut.

- **Etape 2 : Lire les données reçues de la carte Arduino (VISA Read) :**

Apres acquisition des données de la carte Arduino la fonction 'VISA Read Fonction' est utilisée pour lire ces données sous forme d'octets et de les transférer par la suite aux instruments de traitement. Cette fonction est représentée dans la figure suivante.

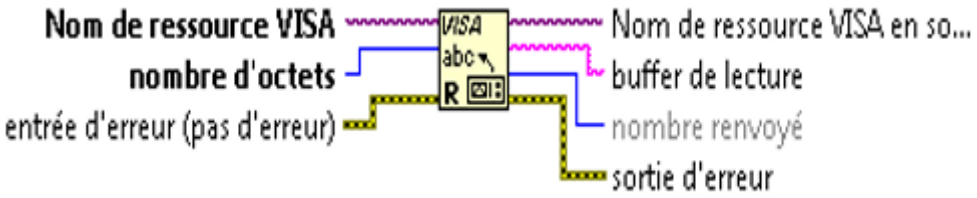


Figure. 3 : La fonction 'VISA Read Fonction'.

- **Etape 3 : découper la chaine reçue de l'Arduino :**

Après chaque acquisition le paquet de données reçues et lues comporte successivement les états des différents capteurs surveillés. Pour séparer ces valeurs et affecter à chaque état la valeur qui lui correspond, nous avons utilisé la fonction 'Rechercher une expression' [18].

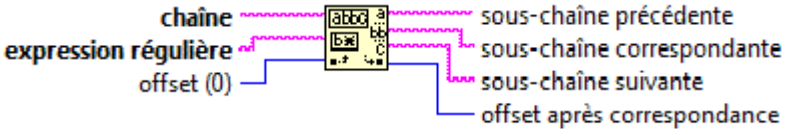


Figure. 4 : la fonction ' *Rechercher une expression'*.

- **Etape 4 : Affectation et représentation des donné reçue :**

Pour afficher les états des différent capteurs et indicateur utilisé pour la réalisation de la face avant de l'application développé les chaines de données reçues on était convertie en nombre par l'utilisation de la fonction 'Chaîne Fract/Exp en nombre '.

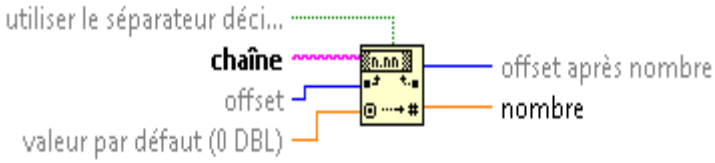


Figure. 5 : la fonction 'Chaîne Fract/Exp en nombre '.

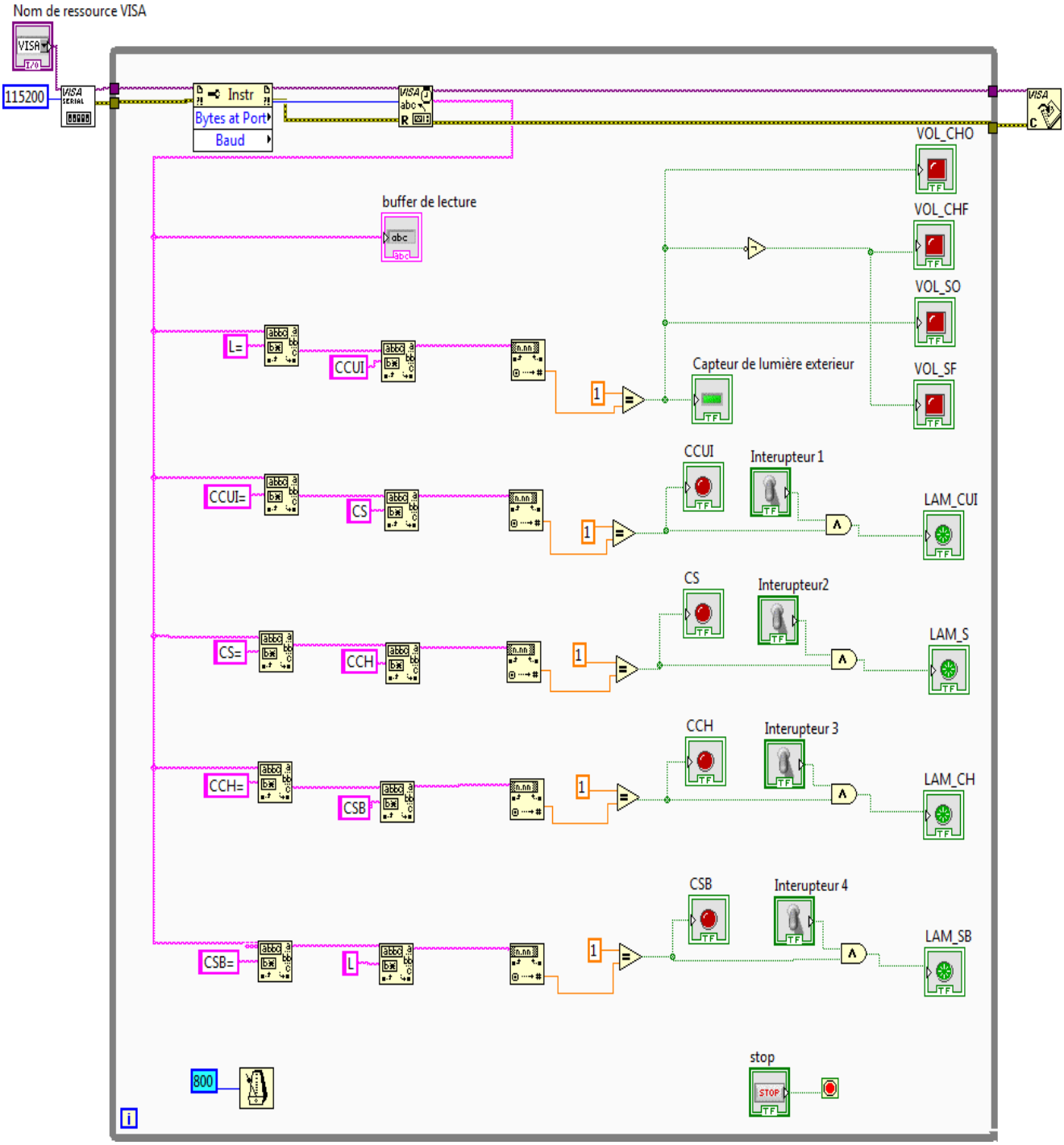


Figure.6 : Diagramme de l'application LabVIEW.

III.2. Face avant de l'application LabVIEW :

La face avant de l'application LabVIEW développée permet d'afficher l'état des variantes principales du système d'éclairage. Elle affiche l'état des lampes, l'état des volets et l'état des détecteurs de présence.

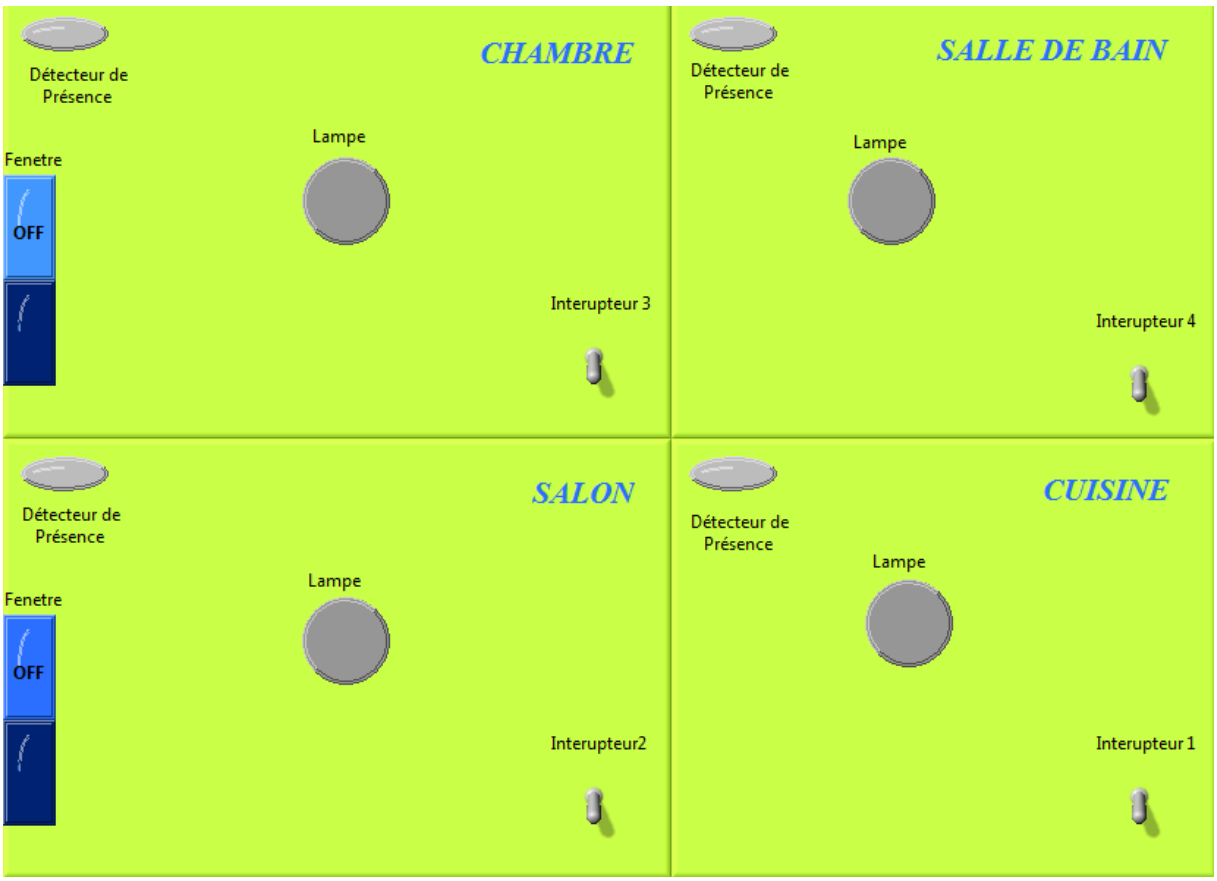


Figure.7 : Face avant de l'application LabVIEW

La face avant de notre application développé comporte :

- Quatre LED de forme ovale (figure.8) pour représenter les quatre détecteurs de présence (Led grise lorsque le détecteur à l'état OFF, Led rouge lorsque le détecteur est à l'état ON)

Figure.8 : (a) détecteur à l'état OFF, (b) détecteur à l'état ON.

- Quatre LED de forme circulaires (figure.9) pour représenter les quatre lampes (Led grise lorsque lampe éteinte, Led rouge lorsque lampe allumée)

Figure.9 : (a) Lampe allumée, (b) Lampe éteinte.

- Deux paires de LED rectangulaire (figure.10) pour représenter l'état des deux volets (ouvert ou fermé).

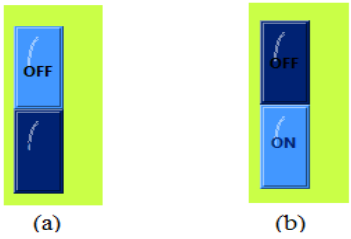


Figure. 10 : (a) volet ouvert, (b) volet fermé.

III.3. Publication Web de l'interface Graphique LabVIEW :

Afin de pouvoir visualiser l'état des détecteurs de présence, des lampes et des volets du système de gestion d'éclairage à partir de n'importe quel Pc connecté à internet, nous avons réalisé la publication web de l'interface graphique LabVIEW par l'utilisation de LabVIEW Server. L'apparence de cette publication Web est donnée sur la figure 11 suivante.

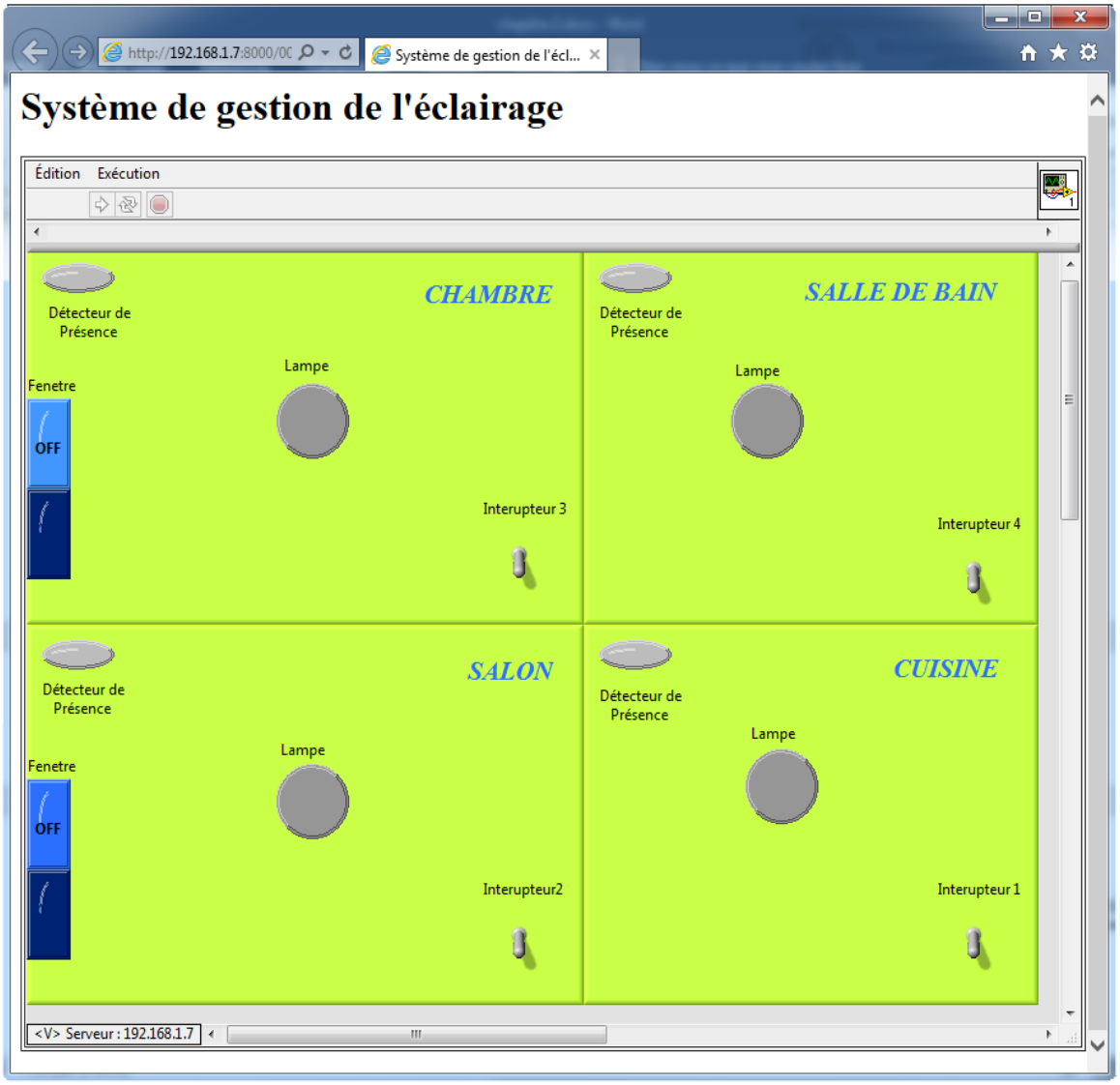


Figure.11 : Publication WEB de l'interface graphique LabVIEW.

CONCLUSION GÉNÉRALE

L'objectif de notre travail traité dans ce mémoire de master est de concevoir et réaliser un système d'éclairage d'une maison comportant quatre pièces (salon, chambre, cuisine et salle de bain) et dont l'alimentation en énergie électrique est assurée par un système photovoltaïque (PV) autonome.

Le principe mis en œuvre pour la gestion d'éclairage repose sur le fait que, pour assurer le confort lumineux nécessaire dans une pièce donnée de la maison, la lampe servant à éclairer cette pièce n'est allumée que lorsqu'une personne est présente dans cette pièce et que la lumière du jour, rentrant par les fenêtres, ne suffit pas à assurer ce confort lumineux.

La partie matérielle du système de gestion d'éclairage réalisé est composé de cinq blocs :

a- Le bloc détecteurs et capteur, comprenant les détecteurs de présence et le capteur de lumière délivrant l'information de présence de personne et de lumière dans les pièces de la maison et le niveau d'éclairement à l'extérieur de la maison.

b- Le bloc de commande composé d'une carte à microcontrôleur de commande et des circuits interfaces de commande.

c- Le bloc d'éclairage réalisé par les circuits d'éclairage.

d- Bloc d'affichage et supervision comprenant une application logicielle installée sur un PC permettant d'afficher l'état du système de gestion d'éclairage.

e- Bloc du système PV qui génère l'énergie électrique nécessaire pour le fonctionnement du système de gestion d'éclairage.

La partie Soft est composée de trois applications logicielles. La première est le programme implémenté dans la carte Arduino de commande traduisant le fonctionnement du système de gestion de l'éclairage. La deuxième application est l'application que nous avons développée sous l'environnement LABVIEW. Le rôle de cette application est de lire les informations écrites sur le port série de la carte Arduino et d'afficher sur une interface graphique l'état des détecteurs de présence, des lampes et des volets. La dernière est une publication web de l'interface graphique de l'application LabVIEW, permettant de télé-surveiller à distance l'état du système de gestion d'éclairage.

Bibliographie

[1] A. Deneyer, P. D'Herdt et B. Deroisy, « Guide pratique et technique de l'éclairage résidentiel », Université catholique de Louvain,2011.

[2] Tridonic, « Systèmes de gestion de l'éclairage », 2016.

[3] S. Landrault, H Weisslinger « Arduino : premiers pas en informatique embarquée », Juin 2014.

[4] J-N Montagné « Initiation à la mise en œuvre matérielle et logicielle de l'Arduino », centre de ressources art sensitif, novembre 2006, sous licence cc.

[5] S. Landrault « Arduino pour bien commencer en électronique et en programmation », Aout 2012.

[6]W. Durfee, « Arduino Microcontroller Guide » Université du Minnesota, Octobre 2011.

[7] Luis Reynier « c'est quoi Arduino ».

[8] E. Grolleau, « Introduction à LabVIEW Premiers pas vers l'expérience » Novembre 2007.

[9] F. Cottet, « LabVIEW Programmation et application », Collection EEA.

[10] National Instruments « LabVIEW Initiation à LabVIEW », juin 2013.

[11] D. Frey, P. Degryse « Le langage de programmation Labview », IUT1 Grenoble Département GEII1, 2007.

[12] V. Chollet « COURS LabVIEW », Université de Franche-Comté, *janvier* 2012.

[13] O. BENSEDDIK et F. DJALOUD, « Etude et optimistion du fonctionemment d'un système photovoltaique » Mémoire de Master, Uneversité Kasdi Merbah-Ouargla, 2012.

[14] W. BENSACI, « Modélisation et simulation d'un système photovoltaïque adapté par une commande MPPT » Diplôme de Master, Université Kasdi Merbah–Ouargla, 2012.

[15] M. CHAABENE, « Cours : gestion énergétique des panneaux photovoltaïques » École nationale d'ingénieurs de Sfax, 2008.

[16] A. BILBAO LEARRETA, « Réalisation de commande MPPT numériques », Rapport de stage, Université Rovira I Virgili, 2006.

[17] Texas Instruments « ULN2803A Darlington Transistor Arrays », janvier 2015.

[18] National Instruments Corporation, « LabVIEW Manuel de référence des VIs et des fonctions ».

Sites Web :

https://www.arduino.cc

http://www.ni.com

http://www.mon-club-elec.fr

http://www.alldatasheet.com

https://openclassrooms.com/dashboard