

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

Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et D'Informatique
DEPARTEMENT D'ELECTRONIQUE

**Mémoire de Fin d'Etude
de MASTER ACADEMIQUE
Spécialité : INSTRUMENTATION**

Présenté par
**Ounissa BOUHALI
Lila KAOUDJT**

Thème

**Conception et réalisation d'un système
de télésurveillance**

Mémoire soutenu publiquement le 26/09/2018 devant le jury composé de :

M Mourad LAZRI, Maitre de conférences A, UMMTO, Président

M Fethi OUALLOUCHE, Maitre de conférences B, UMMTO, Rapporteur

M Slimane HAMEG, Maitre Assistant A, UMMTO, Examineur

Remerciements

Nous remercions tout d'abord par excellence sa grandeur « LE BON DIEU » qui nous a donné le courage et la patience tout au long de notre vie.

Nous remercions notre promoteur Mr. OUALOUCHE pour ses précieuses orientations, son aide et ses conseils tout au long de ce projet.

Nous remercions également le président de jurys ainsi que les membres de jurys d'avoir accepté dévalué et juger notre travail.

Nos remerciements les plus chaleureux vont à nos chères FAMILLES pour leurs encouragements, leur patience et leur grand soutien durant toutes ces années d'études.

Enfin, un remerciement tout particulier à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet.



Dédicaces

Je dédie ce travail :

*En premier lieu à la mémoire de mon père que dieu le
recueillera dans son vaste paradis.*

*A ma très chère mère qui ma soutenue le long de ma
vie en témoignage de leur affection et sacrifices.*

*A mes très chers frères « Messaoud , Md Said, El
Houasse, Farhat et Yacine »*

A ma très chère sœur « Zahra »

*A ma binôme et meilleur amie «Lila» ainsi à toute sa
famille ;*

A tous ceux que j'aime ;

A tous ceux qui m'aiment ;

A tous mes ami(e)s et mes collègues

« Ounissa »



Dédicaces

Je dédie ce travail :

*A mes chers parents qui m'ont toujours soutenue tout
au long de mes études*

A mes très chers frères « Hachimi et Juba ;

*A mes très chers petits cousins « Massinissa et
Douaa » ;*

A toutes ma famille ;

*A ma binôme et meilleur amie « Nissa » ainsi à toute
sa familles ;*

A tous ceux que j'aime ;

A tous ceux qui m'aiment ;

A tous mes ami(e)s et mes collègues

« Lila »

AREF : Analog REference .

BNC : British Naval Connector.

CCD: Charge Coupled Device.

CMOS: Complementary Metal Oxide Semiconductor.

CS: chip select.

EEPROM: Electrically-Erasable Programmable Read-Only Memory.

FTDI: Future Technology Devices International.

GND: GrouND.

GSM: Global System for Mobile Communications.

HREF: Hypertext REference.

ICSP: In-Circuit Serial Programming.

IDE: Integrated Development Environment.

I2C: Inter-Integrated Circuit.

JPEG: Joint Photographic Experts Group.

LED: light-Emitting Diode.

MISO: Master Input, Slave Output.

MLI : Modulation de Largeur d'Impulsion.

MOSI: Master Output, Slave Input.

PCLK: Pixel Clock.

PIR: Passive InfraRed.

PWM: Pulse Width Modulation.

RAM: Random Access Memory.

RGB: Red, Green, Blue.

RJ: registered jack.

RVB: Rouge, Vert, Bleu.

SPI: Serial Peripheral Interface.

RST: Reset Input.

SCCB: Serial Camera Control Bus.

SCK — Serial Clock.

SCL: Serial Clock Line.

SD: Secure Digital.

SDA: Serial Data Line.

SS: Slave Select.

SCLK: Serial Clock, Horloge.

SRAM: Static Random Access Memory.

SVGA: Super Video Graphics Array.

SXGA: Super eXtended Graphics Array.

TOR: Tout Ou Rien.

TWI: Two Wire Interface.

UART: Universal Asynchronous Receiver Transmitter.

USB: Universal Serial Bus.

UXGA : Ultra Extended Graphics Array.

VSYNC : Vertical Synchronisation.

Introduction.....	01
--------------------------	-----------

Chapitre I: Généralités sur la carte Arduino

I.1. Préambule.....	02
I.2. Historique.....	02
I.3. Définition de la carte arduino	02
I.4. Les différentes versions de l'Arduino	02
I.4.1. Arduino Uno.....	02
I.4.2. Arduino Leonardo.....	03
I.4.3. Arduino Mega 2560	04
I.4.4. Arduino Nano.....	04
I.5. Architecture de la carte MEGA 2560.....	05
I.6 .Présentation matériel de la plateforme ARDUINO.....	06
I .6.1. Le microcontrôleur.....	06
I.6.1.1. Caractéristique du microcontrôleur ATMEGA 2560.....	07
I.6.1.2. Démarrage du microcontrôleur.....	07
I .6.1.3. A l'intérieur du microcontrôleur.....	08
I .6.2. Alimentation de la carte Arduino MEGA.....	08
I.6.3. broches d'alimentation de la Arduino MEGA.....	09
I .6.4. Mémoire.....	10
I .6.5.Entrées et sorties numérique.....	10
I .6.6. Broches analogiques.....	11
I .6.7. Autres broches.....	11
I .6.8. Communications.....	12
I .6.9. Protection du port USB contre la surcharge en intensité.....	12
I .7. Description de l'IDE (la partie logicielle).....	14
I .7.1.Description de la barre d'outils.....	14
I .7.2. Description de la barre des menus.....	16
I .7.3. L'envoi du programme.....	16
I .7.4. Réception du programme	17
I .8. Discussion.....	18

chapitre II: Les éléments de systèmes de télésurveillance

II.1. Préambule.....	19
II.2. Définition de la télésurveillance.....	19
II.2.1. Domaine d'applications.....	19
II.2.2. Avantages et inconvénients de la télésurveillance.....	20
II.2.3. Les éléments de télésurveillances.....	20
II.2.3.1. Capteur de position et d'état.....	21
II.2.3.2. Capteur d'images.....	21
II.2.3.3. Transmetteur.....	21
II.2.3.4. L'enregistreur.....	21
II.3. Les capteurs.....	21
II.3.1. Définition d'un capteur.....	21
II.3.2. Chaîne de mesure.....	22
II.3.3. Constitution d'un capteur.....	23
II.3.4. Rôle du capteur.....	24
II.3.5. Classification des capteurs.....	24
II.3.6. Les caractéristiques d'un capteur.....	25
II.3.7. Capteur infrarouge passif (PIR).....	26
II.3.8. Principe de fonctionnement.....	26
II.3.9. Caractéristiques électriques et fonctionnelles de capteur PIR.....	27
II.4. Les caméras.....	28
II.4.1. Les critères de choix d'une caméra.....	28
II.4.2. Différents types de caméras.....	30
II.4.3. La caméra ov2640.....	33
II.4.3.1. Définition des pins.....	34
II.5. ArduCAM shield V2.....	35
II.5.1. Caractéristiques.....	36
II.5.2. Définition des pins.....	36
II.6. Module WIFI ESP32.....	39
II.6.1. Caractéristiques.....	39
II.7. Discussion.....	39

Chapitre III :La réalisation pratique

III .1. Préambule.....	40
III.2. Réalisation matérielle.....	40
III .2.1. Schéma synoptique.....	40
III .2.2. Différents éléments utilisés dans le projet.....	41
III .2.2.1. Branchement de la caméra OV2640 sur l'Arducam.....	41
III .2.2.2. Branchement de l'Arducam shield v2 à la carte Arduino.....	41
III .2.2.3. Branchement du capteur PIR à la carte Arduino.....	42
III .2.2.4. Branchement d'Esp32 avec l'Arduino.....	43
III.2.3. Câblage du système.....	44
III .2.4. L'organigramme.....	46
III .2.5. Etapes de programmation de la carte Arduino	46
III .2.6.librairies.....	49
III .2.7. moniteur série.....	50
III .3. Tests de l'application.....	51
III .4. Serveur Web	53
III .4.1. Création de serveur Web.....	53
III .4.2. Accéder au serveur Web.....	54
III .5. Discussion.....	55
Conclusion	56

bibliographie

Chapitre I : Généralités sur la carte Arduino

Figure I.1. Carte Arduino Uno.....	04
Figure I.2. La carte Leonardo.....	04
Figure I.3. La Carte ARDUINO Mega2560.....	05
Figure I.4. Carte Arduino Nano.....	06
Figure I.5. Architecture de la carte MEGA 2560.....	07
Figure I.6. microcontrôleur ATMEGA2560.....	08
Figure I.7. les broches Tx et Rx de la carte Arduino.....	09
Figure I.8. Alimentation de la carte Arduino MEGA.....	09
Figure I.9. Les différentes parties de la fenêtre principale du logiciel Arduino.....	14
Figure I.10. le moniteur série.....	15
Figure I.11. la barre d'outils.....	16
Figure I.12. la barre de menus.....	17
Figure I.13. l'envoi de programme.....	18
Figure I.14. Réception de programme.....	18

Chapitre II : Les éléments de systèmes de télésurveillance

Figure II.1. la chaîne d'acquisition d'un capteur.....	22
Figure II.2. la chaîne de mesure d'un capteur.....	23
Figure II.3. Constitution d'un capteur.....	24
Figure II.4. capteur infrarouge passif PIR.....	26

Figure II.5. principe de fonctionnement d'un capteur.....	26
Figure II.6. capteur de mouvement.....	27
Figure II.8. réglages du PIR.....	28
Figure II.9. caméra CCD.....	30
Figure II.10. caméra analogique.....	30
Figure II.11. caméra infrarouge.....	31
Figure II.12. caméra CMOS.....	31
Figure II.13. caméra discrète.....	32
Figure II.14. caméra numérique	32
Figure II.15. Caméra ov2640.....	33
Figure II.16. ArduCAM shield V2.....	35
Figure II.17. Architecture d'ArduCAM shield V2.....	36
Figure III.18. Module WIFI ESP32.....	38

Chapitre III : La réalisation pratique

Figure III.1. Schéma synoptique.....	41
Figure III.2. Branchement de la caméra OV2640 sur l'Arducam.....	42
Figure III.3. Branchement de l'Arducam avec Arduino.....	43
Figure III.4. Branchement de capteur PIR.....	44
Figure III.5. Schéma de câblage d'Esp32 avec l'Arduino.....	44

Figure III.6. Câblage de système.....	45
Figure III.7. Montage de notre système.....	46
Figure III.8. L'organigramme.....	47
Figure III.9. L'écriture du programme Arduino.....	48
Figure III.10. Sélection de la cible.....	49
Figure III.11. sélection du port.....	49
Figure III.12. Transfert du programme.....	50
Figure III.13. Choix de librairie.....	51
Figure III.14. présentation du terminal série.....	51
Figure III.15. Ouverture de terminal série.....	52
Figure III.16. Ajustage du débit de la communication série.....	52
Figure III.17. Fenêtre de logiciel lors de détection de mouvement.....	53
Figure III.18. Les images enregistrées sur	53
Figure III.19. message affiché sur l'application.....	54
Figure III.22. l'affichage de l'adresse IP sur le moniteur série.....	55
Figure III.23. L'affichage de l'image sur le serveur web.....	55

Liste des tableaux

Tableau II.1: caractéristique de capteur PIR.....	28
Tableau II.2 : différentes pins de la caméra OV2640.....	35
Tableau II.3: Pins ArduCAM Shield V2.....	38
Tableau II.4 : pins de la broche P6 alternative.....	38
Tableau II.5 : pins de la broche d'interface de caméra.....	39

Introduction Générale

Introduction générale

De nos jours, la télésurveillance est l'une des solutions les plus répandues en matière de sécurité et de surveillance. En effet, on retrouve la télésurveillance dans de nombreux secteurs d'activités comme la sécurité routière, la surveillance de surfaces ou entrepôts industriels, la sécurité des lieux publics (métro, gares, aéroports...) et des lieux dits sensibles (banques). Ce système est aussi utilisé pour une télésurveillance privée (habitations) ou pour la surveillance à distance de publics vulnérables comme les enfants ou les personnes âgées.

Un système de sécurité est un ensemble d'éléments mis en place permettant de détecter une situation de danger potentiel et d'en prévenir les conséquences. De manière générale, il se compose d'une sirène, d'un détecteur de mouvement, d'un clavier, et d'une caméra. Les systèmes de surveillance traditionnels reposent sur des détecteurs capables de déclencher automatiquement l'alerte quand ils repèrent un phénomène suspect, mais encore faut-il pouvoir surveiller ce qui se passe sur les lieux quand on ne s'y trouve pas. C'est là qu'intervient la caméra de surveillance. Grâce à son objectif et à ses facultés de communication, on peut littéralement regarder ce qui se passe dans son champ de vision pour évaluer une menace éventuelle ou simplement contrôler l'absence d'anomalies.

L'objectif principal de notre projet de fin d'études présenté dans ce mémoire est de réaliser un système de télésurveillance commandé par une carte Arduino. Le but est de surveiller un lieu à distance en temps réel via une caméra. Celle-ci se déclenchera grâce à un détecteur de mouvement. Les images acquises seront enregistrées sur une carte mémoire et au même temps transmises vers un pc (un serveur web).

Afin de mener à bien notre projet, nous avons réparti ce mémoire en trois chapitres :

Dans le premier chapitre, nous allons étudier les généralités de la carte Arduino, et présenter la carte utilisée (Arduino Méga) et ses caractéristiques.

Le deuxième chapitre sera consacré à l'étude générale du domaine de la télésurveillance et les différents composants permettant de réaliser un tel système.

La présentation du projet réalisé fera l'objet du troisième chapitre. Nous présenterons les différentes étapes à suivre afin d'avoir un système fonctionnel. De plus, nous allons mettre en évidence les différents tests et les résultats obtenus.

Enfin, nous terminons notre travail par une conclusion et une bibliographie.

Chapitre /

Généralités sur la carte Arduino

I.1. Préambule

Le système Arduino donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, réduire le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique.

I.2. Historique

Le projet Arduino est issu d'une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea (Italie). Ils rencontraient un problème majeur à cette période (2003-2004) ; les outils nécessaires à la création de projet d'interactivité étaient complexes et onéreux (entre 80 et 100 euros). Les outils de prototypage étaient principalement dédiés à l'ingénierie, la robotique et aux domaines techniques. Leur préoccupation se concentre alors sur la réalisation d'un matériel moins cher et plus facile à utiliser.

En 2003, HeranadoBarragan, pour sa thèse de fin d'études, avait entrepris le développement d'une carte électronique dénommée « Wiring », accompagnée d'un environnement de programmation libre et ouvert. Cette carte a donc inspiré le projet Arduino (2005) et conçu par une équipe de professeurs et d'étudiants (David Mellis, Tom Igoe, Gianluca Martino, David Caurtielles, MassimiBanzi et Nicholas Zambetti).

I.3. Définition de la carte arduino

La carte Arduino est un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique, le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles ...etc. Chaque module Arduino possède un régulateur de tension +5V et un oscillateur à quartz. Pour programmer cette carte, on utilise le logiciel IDE Arduino.

I.4. Les différentes versions de l'Arduino**I.4.1. Arduino Uno**

Le modèle Arduino uno (Figure I.1) de la société arduino est une carte électronique basé sur un microcontrôleur ATmega 328.



Figure I.1. Carte Arduino uno

La carte Arduino uno possède un nombre suffisant de broches d'entrée – sorties pour les projets élémentaires et un vaste choix de Shields toutefois, cette carte présente l'inconvénient de ne pas disposer d'un nombre suffisant de broches d'entrée- sorties pour les projets ambitieux. De même, la mémoire disponible risque d'être un peu juste pour ces projets [1].

I.4.2. Arduino Leonardo

La carte Arduino Leonardo (Figure I.2), est équipée du microcontrôleur ATmega32U4 qui permet de reconnaître un clavier ou une souris d'ordinateurs.



Figure I.2. La carte Leonardo

L'Arduino Leonardo est une carte disposée de nombreuses entrées et sorties analogiques et numériques dont 6 entrées analogiques et 6ports numériques (sur les broches analogique A0 à A5) supplémentaires par rapport à la carte Arduino Uno, en plus d'un port série / COM virtuel et une broche MLI de plus que sur l'arduino Uno , mais ces entrées analogiques supplémentaires sont bloquées en cas d'utilisation des broches numérique, car elles sont

utilisées au même moment, et la même chose pour les entrées numériques supplémentaires en cas d'utilisation des broches analogiques[1].

I.4.3. Arduino Mega 2560

L'Arduino MEGA 2560 (Figure I.3): est une carte plus grande que UNO qui est disposée de plus d'espace d'entrée sortie et plus de puissance de calcul, basé sur un microcontrôleur ATmega2560.

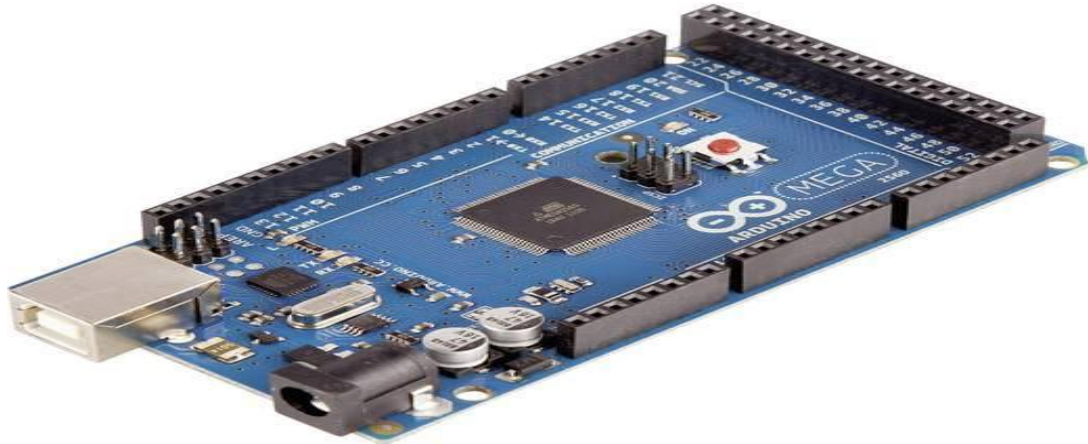


Figure I.3. *La Carte ARDUINO Mega2560*

La MEGA est une carte composée de nombreuses entrées et sorties pour raccorder des capteurs ou des actionneurs (16 analogiques et 54 digitales pouvant fournir jusqu'à 20mA). Comme elle dispose de 256KO de mémoire flash ainsi que 8KO de mémoire SRAM. De plus cette carte dispose de plusieurs broches UART (4 ports de communication), et de broches MLI (15 sorties numériques peuvent être utilisées comme MLI).

Cette carte présente deux inconvénients, par rapport à sa forme plus élevée, et son prix qui est deux fois plus cher que l'Arduino Uno[1]

I.4.4. Arduino Nano :

L'Arduino Nano (Figure I.4) est une petite carte complète et facile à utiliser basée sur l'ATmega328 elle fonctionne avec un câble USB au lieu d'un câble standard.

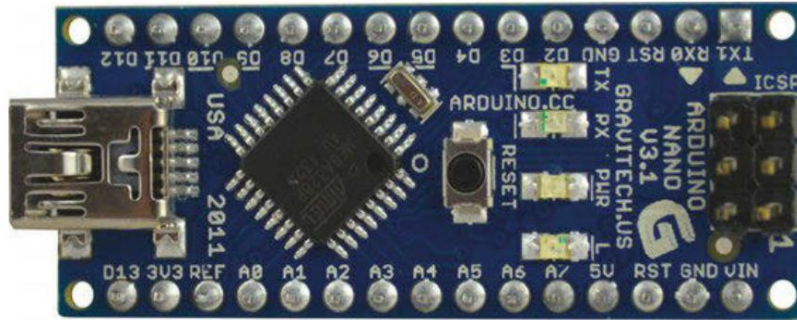


Figure I.4. Carte Arduino Nano

L'Arduino Nano est une carte plus petite, plus légère et plus puissante qu'une UNO, elle dispose de 14 pins digitaux et 8 pins analogiques ainsi 32ko de mémoire flash elle est parfaite pour des projets où l'espace est réduit ou un projet où la légèreté du système prime, comme elle peut être enfichée sur la plaque d'essais.

L'inconvénient de cette carte est qu'il n'est pas possible d'utiliser des shields [1].

➤ Parmi ces différents types on a utilisée la carte Arduino MEGA :

Car elle est adaptée pour les projets qui nécessitent plus de lignes d'E / S, plus de mémoire d'esquisse et plus de RAM. Avec 54 broches d'E / S numériques, 16 entrées analogiques et un plus grand espace, Cela donne à notre projets beaucoup d'espace et d'opportunités tout en maintenant la simplicité et l'efficacité de la plateforme

I.5. Architecture de la carte MEGA 2560

La carte Arduino MEGA 2560 est composée de plusieurs éléments essentiels qui sont présentés dans la figure (I.5).

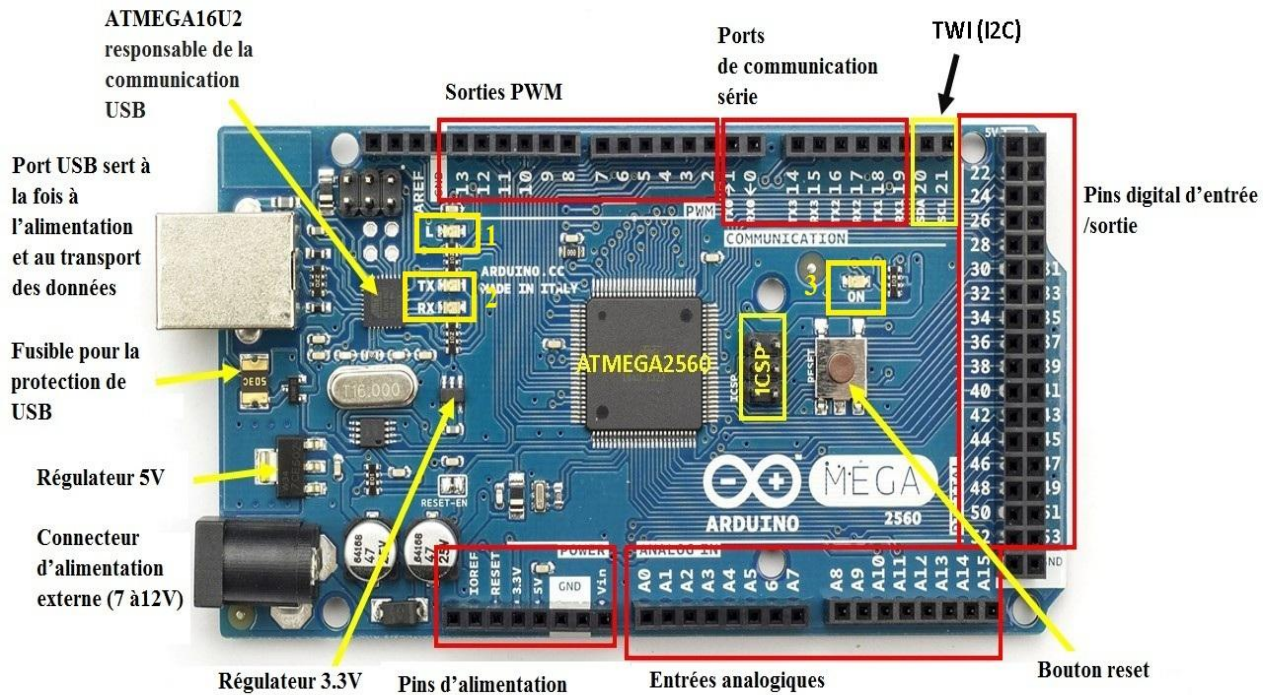


Figure I.5. Architecture de la carte MEGA 2560

Présentation des LED numérotés sur la figure :

- 1) LED de test connectée à D13.
- 2) LED d'activation Rx et Tx.
- 3) LED témoin d'alimentation.

I.6 .Présentation matérielle de la plateforme ARDUINO

L'Arduino est composée de deux parties indissociable : la carte qui est la partie hardware avec laquelle nous travaillons en construisant chaque projet et la plateforme IDE Arduino qui est la partie logicielle sur le PC, celle-ci permet de mettre au point et de transférer le programme qui sera par suite exécuté pas la carte.

I .6.1. Le microcontrôleur

Le microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur qui sont : processeur, mémoires, unités périphériques et interfaces d'entrées-sorties.

Le microcontrôleur de la carte Arduino Méga est un ATmega2560 présenté sur la Figure(I.6). C'est un microcontrôleur ATMEL de la famille AVR.



Figure I.6. *Microcontrôleur ATMEGA2560*

I.6.1.1. Caractéristique du microcontrôleur ATMEGA 2560

C'est un microcontrôleur ATMEL de la famille AVR 8 bits dont les caractéristiques sont :

- FLASH= mémoire programme de 256KO.
- SRAM= données (volatiles) 8KO.
- EEPROM= données (non volatiles) 4KO.
- 54 broches E/S numériques et 16 broches E/S analogiques.
- Vitesse de l'horloge 16MHz.

I.6.1.2. Démarrage du microcontrôleur

Lorsque le microcontrôleur démarre, il va commencer par lancer un bout de code particulier : le boot loader. C'est ce dernier qui va surveiller si un nouveau programme arrive sur la voie USB et s'il faut donc changer l'ancien en mémoire par le nouveau. Si rien n'arrive, il donne la main au programme, celui que nous avons créé. Ce dernier va alors défiler, instruction par instruction. Chaque fois qu'une nouvelle variable sera nécessaire, elle sera mise en RAM pour qu'on ai une mémoire de cette dernière (et supprimer lorsqu'elle n'est plus nécessaire)[2].

I.6.1.3. A l'intérieur du microcontrôleur

➤ L'emplacement du programme :

Le microcontrôleur reçoit le programme sous forme de signal électrique sur ses broches Tx et Rx, d'ailleurs disponible sur les broches de la carte comme le montre la figure (I.7) :

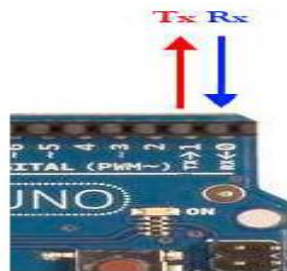


Figure I.7. les broches Tx et Rx de la carte Arduino

Une fois qu'il est reçu, il est intégralement stocké dans une mémoire de type Flash que l'on appellera "la mémoire de programme". Ensuite, lorsque la carte démarre, le cerveau va alors gérer les données et les répartir dans les différentes mémoires :

- La mémoire programme : est la mémoire dans laquelle est stockée le programme que nous avons composé.
- La mémoire de données, aussi appelé "RAM" va stocker les variables telles que le numéro de la broche sur laquelle est connectée une LED, ou bien une simple valeur comme un chiffre, un nombre, des caractères, etc [2].

I.6.2. Alimentation de la carte Arduino MEGA

La Figure (I.8) montre les prises d'alimentation de la carte Arduino :

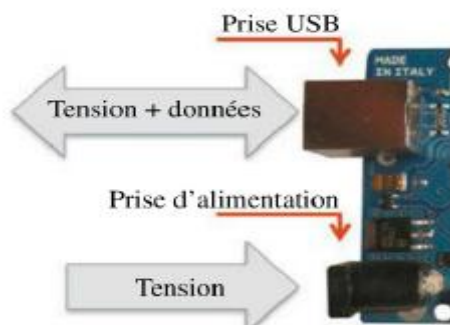


Figure I.8. Alimentation de la carte Arduino MEGA

La carte Arduino Mega 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 adaptateur 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 V_{IN} (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 nous utilisons 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 pas le circuit intégré FTDI (usb-vers-série) qui est un circuit intégré faisant l'adaptation du signal entre le port USB de l'ordinateur et le port série de l'ATmega. A la place, elle utilise un Atmega8U2 programmé en convertisseur USB-vers-série.

I.6.3. Les broches d'alimentation de Arduino MEGA

- V_{IN} . La tension d'entrée à la carte Arduino quand il s'agit d'utiliser une source d'alimentation externe (par opposition à 5 volts de la connexion USB ou autre source d'alimentation régulée). Nous pouvons alimenter la carte à travers de cette broche ou bien si l'alimentation est fournie par le jack (batterie ou un bloc secteur au connecteur) d'alimentation, accéder à la tension d'alimentation.
- 5V. L'alimentation régulée utilisée pour alimenter le microcontrôleur et d'autres composants de la carte. Cela peut provenir soit de V_{IN} via un régulateur de bord, ou être alimenté par USB (qui fournit de 5V régulée) ou d'une autre alimentation de 5V régulée.
- 3V3. alimentation de 3.3V fournie par le circuit intégré FTDI de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu de 5V). l'intensité maximale disponible sur cette broche est de 50mA.

- GND. les épingles de la terre [3].

I.6.4. La Mémoire

L'ATmega2560 à 256 Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisée par le bootlaoder). L'Atmega2560 a également 8Ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile-mémoire qui peut être lue à l'aide de la librairie EEPROM).

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 [3] .

I.6.5. Entrées et sorties numériques

Chacune des 54 broche numériques de la carte Mega peut être utilisée soit comme entrée numérique ou bien comme une sortie numérique avec l'utilisation des instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité.

Il y a entre ces broches celles qui ont des fonctionnalités en plus :

➤ Communication Série:

Il y a 4 ports séries pour la carte Arduino Méga qui est l'un de ces principaux avantages

- port série numéro 0 : broche 0(RX) et 1(TX).
- Port série numéro 1 : broche 19(RX) et 18(TX).
- Port série numéro 2 : broche 17(RX) et 16(TX).
- Port série numéro 3 : broche 15(RX) et 14(TX).

La communication série utilise pour recevoir la broche RX et pour la boche TX pour transmettre les données série de niveau TTL. les broches RX et TX sont connectées aux broche correspondante du circuit intégré ATmega8U2 programmée en convertisseur USB/ série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

La communication série est appelée grâce à l'instruction `Serial.print()` de langage Arduino .

- 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. Voir l'instruction `attachInterrupt()` pour plus de détails.
- Impulsion PWM (Pulse Width Modulation): Broches 0 à 13. Fournissent une impulsion PWM de 8bits à l'aide de l'instruction `analogWrite()`.
- SPI (Serial Peripheral Interface): Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI disponible avec la librairie pour communication SPI. 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"), disponible en utilisant la librairie `Wire/I2C` .
- 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 [3].

I.6.6. Broches analogiques

La carte Mega 2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la fonction `analogRead()` de la Arduino. 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 (`analogReference()`) .

I.6.7. 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()`.
- 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.

I.6.8. Communications

La carte Arduino Mega2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega2560 dispose de quatre UART (Universal Asynchronous Receiver Transmitter) ou émetteur-récepteur asynchrone universel en français pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun d'autre driver externe n'est nécessaire.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LED de RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré ATmega8U2 utilisé en convertisseur USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Une librairie Série Logicielle permet également la communication série sur n'importe quelle broche numérique de la carte.

L'ATmega2560 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface- Interface « 2 fille ») et SPI

L'ide Arduino inclut la Wire qui simplifie l'utilisation du bus I2C, de plus pour utilisé la communication SPI, la librairie `SPISettings` pour la communication SPI est disponible [3].

I.6.9. Protection du port USB contre la surcharge en intensité

La carte Arduino Mega2560 intègre un poly fusible réinitialisable qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppée.

I.7. Présentation logicielle de Arduino

L'IDE est un logiciel de programmation qui permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte. Il programme par code, contenant une cinquantaine de commandes différentes.

Le logiciel Arduino a pour fonctions principales :

- de pouvoir écrire et compiler des programmes pour la carte Arduino.
- de se connecter avec la carte Arduino pour y transférer les programmes.
- de communiquer avec la carte Arduino.

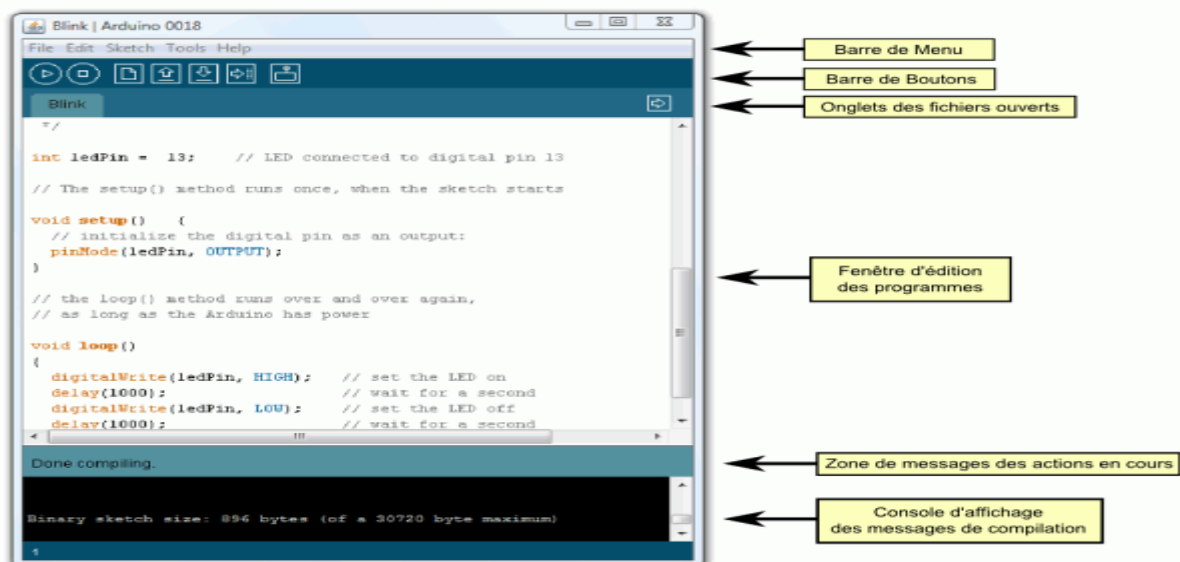


Figure I.9. Les différentes parties de la fenêtre principale du logiciel Arduino

La programmation des cartes Arduino comporte :

- une BARRE DE MENUS comme pour tout logiciel une interface graphique (GUI) ;

- une BARRE DE BOUTONS qui donne un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation ;
- un EDITEUR (à coloration syntaxique) pour écrire le code de vos programme, avec onglets de navigation ;
- une ZONE DE MESSAGES qui affiche indique l'état des actions en cours ;
- une CONSOLE TEXTE qui affiche les messages concernant le résultat de la compilation du programme ;
- un moniteur serie sur la Figure (I.10), permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino.

Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes [4].

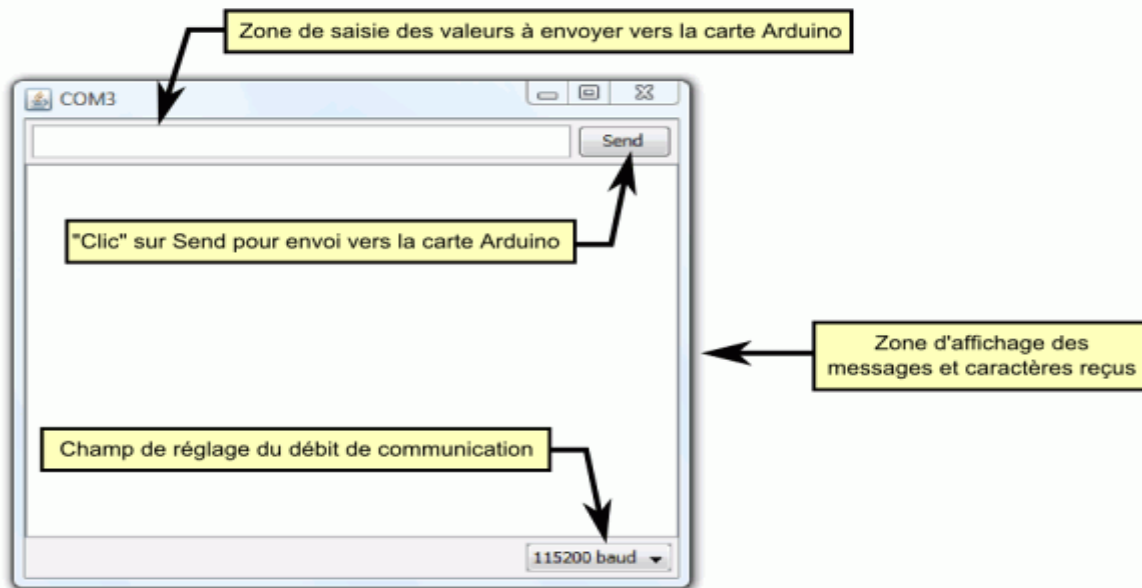


Figure I.10. Le moniteur série

I.7.1. Description de la barre d'outils

La barre d'outils qui est présenté sur la Figure I.11 contient plusieurs boutons qui sont communément utilisés lorsque vous écrivez des croquis pour Arduino [4].



Figure I.11. la barre d'outils



Vérifier/compiler: ce bouton nous permet de vérifier s'il y a des erreurs dans le programme en sélectionnant la ligne où se trouve l'erreur



Stop: ce bouton oblige le moniteur série ou les autres boutons activés de s'arrêter.



Nouveau: en appuyant sur cette icône, une fenêtre d'édition vide s'ouvre pour la saisie d'un nouveau programme.



Ouvrir: il suffit de cliquer sur ce bouton pour accéder à la liste de tous les programmes qui se trouvent dans le "livre de programmes". Cliquer sur l'un des programmes l'ouvre dans la fenêtre courante



Sauvegarder: avant de quitter le programme saisi, il faut l'enregistrer, et cela peut se faire en cliquant sur cette icône.



Transférer vers la carte: ce bouton permet de compiler votre code et de le transférer vers la carte Arduino



Moniteur Série: pour ouvrir la fenêtre du moniteur série, il suffit d'appuyer sur ce

I.7.2. Description de la barre des menus

Ce sont les options de configuration du logiciel qui nous permettent de créer de nouveaux programmes de les sauvegarder, et de gérer les paramètres de communication avec la carte Arduino [2].

La Figure (I.12) montre la barre de menus :

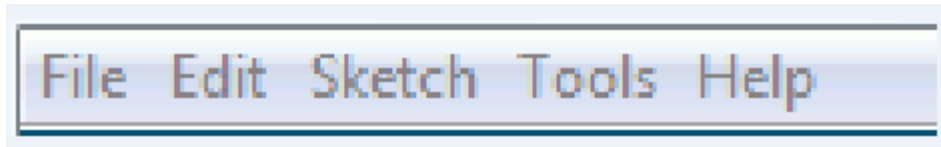


Figure I.12. *la barre de menus*

- **File (Fichier):** ce menu contient les différentes options de création, d'ouverture, de sauvegarde, d'impression d'un programme ou l'ouverture d'un exemple parmi les exemples qui accompagnent le logiciel Arduino.
- **Edit (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'ajoutée de bibliothèques ou de fichiers.
- **Tools (outils):** c'est dans ce menu qu'on sélectionne le type de carte à programmer, et le port série utilisé ainsi que la fonction de chargement du boot loader dans l'ATmega.
- **Help (Aide):** ce menu est fait pour donner de l'aide concernant les différents problèmes rencontrés au niveau du logiciel Arduino.

I.7.3. L'envoi du programme

Le programme est envoyé dans la carte lorsque nous cliquons sur le bouton upload (téléverser). Le logiciel Arduino va alors vérifier si le programme ne contient pas d'erreur et ensuite le compiler (le traduire) pour l'envoyer dans la carte [2]:

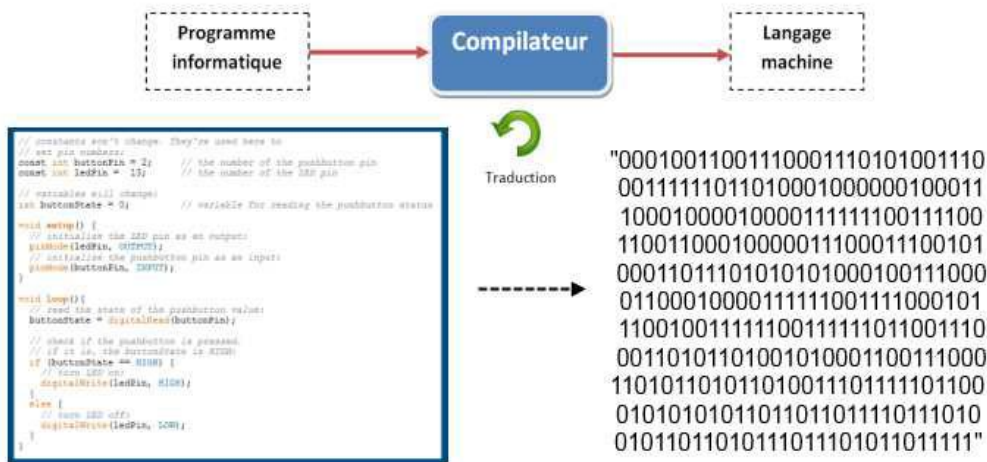


Figure I.13 L'envoi de programme

Au départ, le programme est sous forme de texte, puis il est transformé en un langage composé uniquement de 0 et de 1, dans le câble USB qui relie l'ordinateur à la carte et arrive dans la carte. Le reste se passe dans la carte elle-même.

I.7.4. Réception du programme

Le programme rentre donc dans la carte en passant en premier par le connecteur USB de celle-ci. Il va alors subir une petite transformation qui permet d'adapter le signal électrique correspondant au programme vers un signal plus approprié pour le microcontrôleur. On passe ainsi d'un signal code pour la norme USB à un signal code pour une simple voie série. Puis ce nouveau signal est alors intercepté par le microcontrôleur [2].

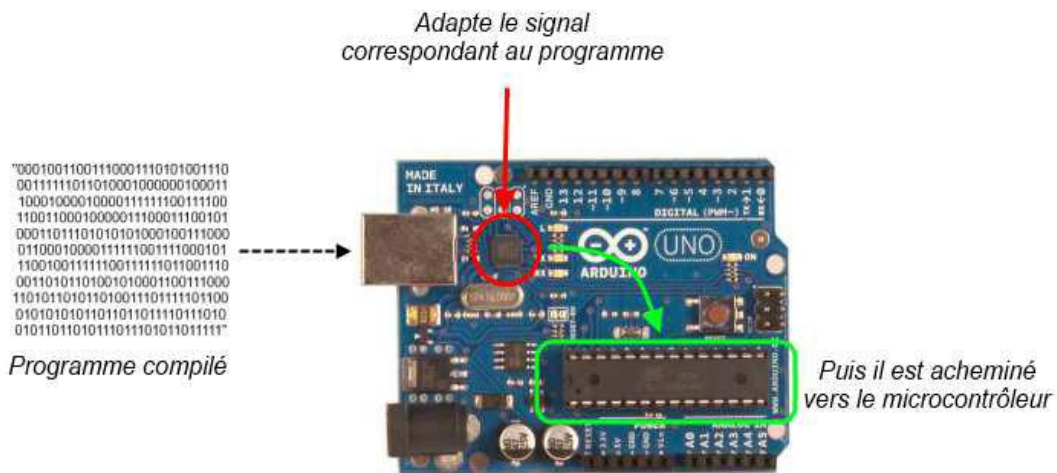


Figure I.14. Réception de programme

I .8. Discussion

Dans ce chapitre, nous avons projeté la lumière sur une carte d'acquisition qui est l'Arduino donnant ainsi les raisons pour lesquelles nous l'avons choisie, puis nous avons cité les différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; (la partie matérielle et la partie de programmation). Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses Caractéristiques.

D'après cette étude, il apparaît clairement que la carte Arduino est un outil économique parfaitement adapté pour commander des systèmes électroniques complexes.

CHPITRE II

Les éléments de systèmes de télésurveillance

II.1. Préambule

Le besoin de renforcer les niveaux de sécurité se fait de plus en plus ressentir dans différents domaines d'activités. Un des moyens utilisé est la télésurveillance.

La télésurveillance est omniprésente et on la retrouve dans de nombreux secteurs d'activité (banque, transport, industrie, etc.) ou lieux de vie (ville, immeubles de bureaux, ... etc.) dans le but de surveiller et de protéger des personnes et des biens. Dans ce chapitre nous allons étudier les composants essentiels de notre système de télésurveillance : l'Arducam shield V2, la caméra (ov2640) et le capteur infrarouge passif PIR et on termine par un module WIFI (ESP32).

II.2. Définition de la télésurveillance

La télésurveillance est la surveillance à distance d'un lieu, public ou privé, de machines ou d'individus. Elle est employée dans de nombreuses situations, généralement pour des raisons de sécurité. Son but est de surveiller des installations de systèmes de sécurité alarmes technique, alarmes vidéo, alarmes médicale (personnes âgées), alarmes agression et même de la vidéosurveillance [5].

II.2.1. Domaine d'applications

La télésurveillance est employée dans de nombreuses situations, généralement pour des raisons de sécurité :

- Dans le cadre de la sécurité routière, au moyen de caméras spécialisées ou des capteurs à proximité voire même noyés dans la chaussée permettent d'évaluer la densité du trafic, les ralentissements qui peuvent en découler, la présence de personnes sur les bandes d'arrêt d'urgence, etc.
- Pour la surveillance des machines : divers capteurs permettent d'évaluer l'état de la machine, ces informations peuvent alors être envoyées à un poste de surveillance. L'épuisement de consommables, une anomalie de fonctionnement ou même un acte de malveillance serait alors détecté à distance.

- Dans le cadre de la prévention de la délinquance (avec notamment la vidéosurveillance) .
- Pour la surveillance de lieux sensibles (banques, centrales nucléaires, etc.) et d'habitations, afin de prévenir les intrusions, les cambriolages et les actes de vandalisme .
- Dans le cadre de la télémédecine, et en particulier pour la surveillance des patients à distance.
- Pour la surveillance à distance des enfants et des personnes vulnérables [5].

II.2.2. Avantages et inconvénients de la télésurveillance

La télésurveillance présente les avantages suivants :

- la dissuasion des voleurs par l'affichage des caméras de télésurveillance sur écran de télésurveillance.
- l'analyse visuelle des intrusions et vols sur les enregistrements vidéo des caméras.
- les images capturées peuvent être visionnées en temps réel ou visionnées à distance à partir d'un ordinateur connecté au réseau (Internet, Intranet...).
- il peut être paramétré en fonction des besoins : enregistrement vidéo 24h/24 et 7j/7 ou bien enregistrement uniquement lorsque l'alarme se déclenche, prise de photos, alerte par e-mail ou SMS (sur PC ou PDA), association avec un système d'alarme ou de détection de mouvement, d'intrusion, de chaleur
- l'amélioration de la sécurité et la diminution des actes illicites dans les rues.

Toutefois la télésurveillance présente l'inconvénient de risque d'atteinte à la vie privée et aux libertés d'où la nécessité d'une démarche éthique [6].

II.2.3. Les éléments de télésurveillance

Les systèmes de télésurveillance sont très variés cependant la plupart d'entre eux sont composés des éléments suivants :

II.2.3.1. Capteur de position et d'état

Les capteurs de position et d'état ont pour tâche de donner et détecter tout mouvement dans les locaux sous surveillance.

II.2.3.2. Capteur d'images

Des caméras de surveillance permettent de visualiser et d'enregistrer les images du lieu à protéger en les transmettant par liaison vidéo, réseau IP ou sans fil à un centre de télésurveillance.

II.2.3.3. Transmetteur

Dans certains cas, le système est complété par un transmetteur téléphonique numérique ou par un transmetteur GSM qui alerte le centre de télésurveillance ou le propriétaire de l'habitation dès qu'une intrusion détectée.

II.2.3.4. L'enregistreur

L'enregistreur permet d'enregistrer et de sauvegarder des données collectées (images, sons, etc.) pour une utilisation ultérieure [6].

II.3. Les capteurs

Une mesure est une représentation quantifiée d'une grandeur physique (température, pression, champ magnétique ...). Nous définissant la terminologie suivante :

- Mesurande : grandeur physique soumise à un mesurage (pression, température, ...),
- Mesurage : toutes les opérations permettant l'obtention de la valeur d'une grandeur physique (mesurande),
- Mesure : valeur numérique représentant le mesurande (6 MPa, 20°C, 2 m.s-1 ...) [7].

II.3.1. Définition d'un capteur

Un capteur est un dispositif qui transforme une grandeur physique analogique d'entrée (température, pression, déplacement,...) en une autre grandeur de nature électrique. Les capteurs sont souvent intégrés à la chaîne d'acquisition permettant à la grandeur mesurée

d'être conditionnée afin que la mesure (ou signal de sortie) donne une estimation optimisée du mesurande.

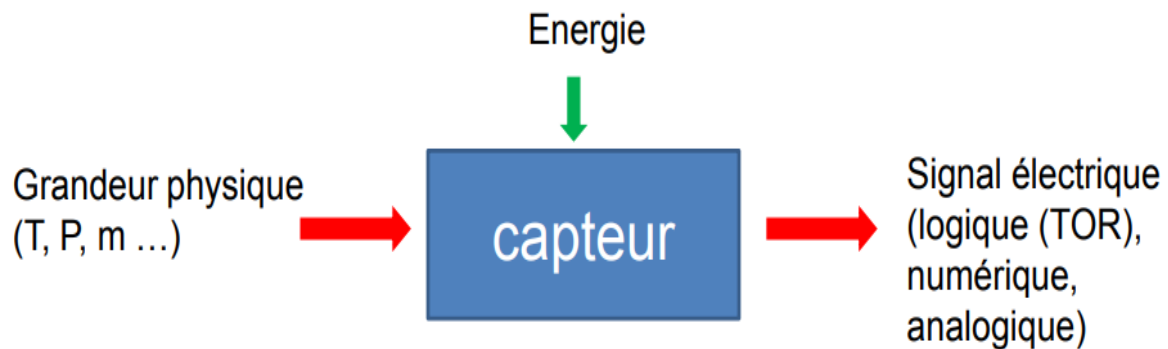


Figure II.1. *la chaîne d'acquisition d'un capteur.*

La grandeur de sortie du capteur peut varier :

- de manière binaire (information vraie ou fausse), c'est le capteur Tout Ou Rien (TOR);
- de façon progressive (variation continue), c'est le capteur analogique;
- d'échelon de tension ou de courant, c'est le capteur numérique [7].

II.3.2. Chaîne de mesure

Généralement, le signal de sortie de capteur n'est pas directement utilisable. On appelle chaîne de mesure l'ensemble des circuits ou appareils qui amplifient, adaptent, convertissent, linéarisent, digitalisent le signal avant sa lecture sur le support de sortie.

Pour obtenir une image d'une grandeur physique, la chaîne de mesure peut faire intervenir plusieurs phénomènes différents. Par exemple, la mesure d'un débit peut se faire en plusieurs étapes :

- ✓ Transformation du débit en une pression différentielle.
- ✓ Transformation de la pression différentielle en la déformation mécanique d'une membrane.
- ✓ Transformation de la déformation mécanique en une grandeur électrique (à l'aide d'un piézoélectrique) via un circuit électrique associé [7].

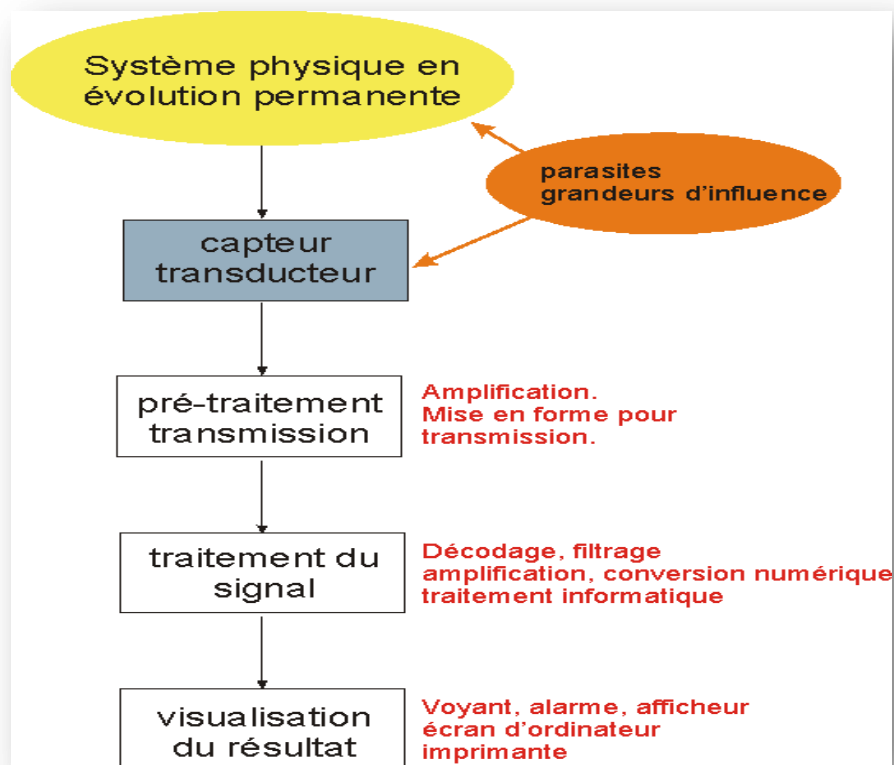


Figure II.2. La chaîne de mesure d'un capteur.

II.3.3. Constitution d'un capteur

Certains capteurs sont des capteurs dits « composites », c'est à dire composés de 2 parties ayant un rôle bien défini :

- Corps d'épreuve : C'est un élément qui réagit sélectivement aux variations de la grandeur à mesurer. Il a pour rôle de transformer cette grandeur en une autre grandeur physique dite mesurable.
- Élément de transduction : C'est un élément lié au corps d'épreuve qui traduit ses réactions en une grandeur physique exploitable.

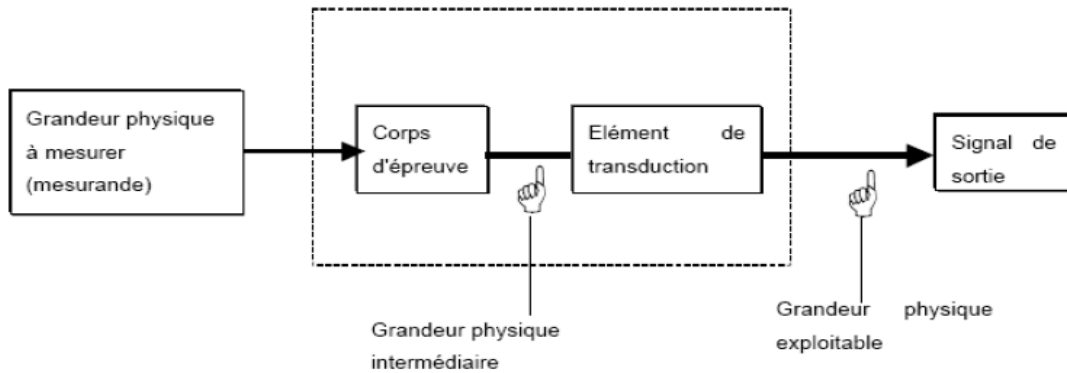


Figure II.3. Constitution d'un capteur

II.3.4. Rôle du capteur

Les capteurs jouent des rôles de plus en plus importants car ce sont eux qui permettent de mesurer les effets des phénomènes de toutes natures qui agissent sur l'environnement. Le rôle du capteur est de rendre exploitable ces différents phénomènes en vue de leur traitement ultérieur.

- Mesure de présence ;
- Mesure de position, de déplacement ou de niveau ;
- Mesure de vitesse ;
- Mesure d'accélération, de vibrations ou de chocs ;
- Mesure de débit, de force, de couples, de pressions ;
- Mesure de température, d'humidité.
-

II.3.5. Classification des capteurs

Si nous nous intéressons aux phénomènes physiques mis en jeu dans les capteurs, on peut classer ces derniers en deux catégories.

➤ Capteur actif

Fonctionnant en générateur, un capteur actif est généralement fondé dans son principe sur un effet physique qui assure la conversion en énergie électrique de la forme d'énergie propre à la grandeur physique à prélever, énergie thermique, mécanique ou de rayonnement.

Les effets physiques les plus classiques sont :

- *Effet thermoélectrique* : Un circuit formé de deux conducteurs de nature chimique différente, dont les jonctions sont à des températures T_1 et T_2 , est le siège d'une force électromotrice d'origine thermique (T_1 , T_2).
- *Effet piézo-électrique* : L'application d'une contrainte mécanique à certains matériaux dits piézo-électriques (le quartz par exemple) entraîne l'apparition d'une déformation et d'une même charge électrique de signe différent sur les faces opposées.
- *Effet d'induction électromagnétique* : La variation du flux d'induction magnétique dans un circuit électrique induit une tension électrique (détection de passage d'un objet métallique).
- *Effet photo-électrique* : La libération de charges électriques dans la matière sous l'influence d'un rayonnement lumineux ou plus généralement d'une onde électromagnétique.
- *Effet photovoltaïque* : Des électrons et des trous sont libérés au voisinage d'une jonction PN illuminée, leur déplacement modifie la tension à ses bornes.

- **Capteur passif**

Il s'agit généralement d'impédance dont l'un des paramètres déterminants est sensible à la grandeur mesurée. La variation d'impédance résulte :

- Soit d'une variation de dimension du capteur, c'est le principe de fonctionnement d'un grand nombre de capteur de position, potentiomètre, inductance à noyaux mobile, condensateur à armature mobile.
- Soit d'une déformation résultant de force ou de grandeur s'y ramenant, pression accélération (armature de condensateur soumise à une différence de pression, jauge d'extensomètre liée à une structure déformable).

II.3.6. Les caractéristiques d'un capteur

- Etendue de mesure : Valeurs extrêmes pouvant être mesurée par le capteur.
- Résolution : Plus petite variation de grandeur mesurable par le capteur.
- Sensibilité : Variation du signal de sortie par rapport à la variation du signal d'entrée.
- Précision : Aptitude du capteur à donner une mesure proche de la valeur vraie.
- Rapidité : Temps de réaction du capteur.
- Linéarité : représente l'écart de sensibilité sur l'étendue de mesure.

II.3.7. Capteur infrarouge Passif (PIR)

Dans le cas du système à réaliser, nous allons utiliser un PIR. Celui-ci est un capteur qui mesure les radiations infrarouge émises par les objets qui se trouvent dans son champ de vision. Il est généralement utilisé comme détecteur de mouvement dans divers systèmes de sécurité. Le capteur PIR peut détecter un mouvement jusqu'à une distance maximale de 6m [8].



Figure II.4. Capteur infrarouge passif PIR.

II.3.8. Principe de fonctionnement

Un détecteur infrarouge passif peut analyser le rayonnement thermique émis par tous corps présentant une température supérieure à celle du 0 absolu. L'homme présente une température externe avoisinant les 35°C. Il émet donc un rayonnement infrarouge. La fonction d'un détecteur infrarouge est de différencier le rayonnement émis par un être humain avec celui émis par des objets, de déterminer si la source d'émission du rayonnement est fixe ou en mouvement, d'émettre une information dans le cadre d'une détection (Voir figure II.5).

Le détecteur de mouvement à infrarouge passif (PIR = Passive InfraRed) associe un capteur pyroélectrique avec une électronique de conditionnement (adaptation, traitement) du signal et une optique à base de lentille de Fresnel [8].

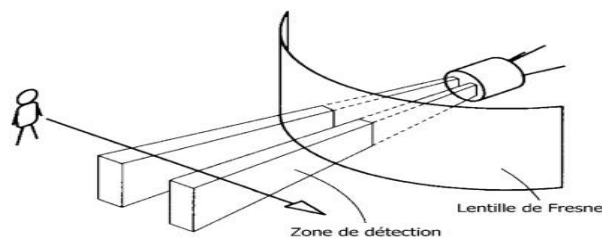


Figure II.5. Principe de fonctionnement d'un capteur.

✓ **La lentille de Fresnel**

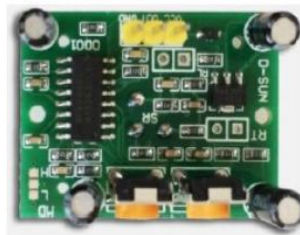
A pour fonction de focaliser sur le capteur pyroélectrique le rayonnement infrarouge d'une personne se déplaçant dans le champ de surveillance du détecteur. La forme et la taille de la lentille déterminent en grande partie la directivité et la sensibilité du détecteur.

✓ **Le capteur pyroélectrique**

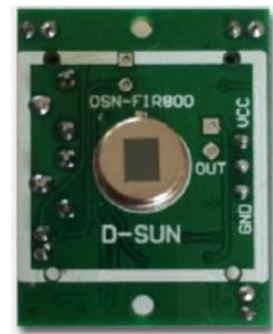
Détecte les variations de radiation infrarouge provoquées par le mouvement d'une personne (ou d'un objet) dont la température est différente de la température du milieu ambiant. Le capteur PIR n'est sensible qu'aux déplacements de chaleur se produisant dans la zone de détection (quelques mètres au maximum). Ainsi, les variations lentes de la température ambiante ne sont pas perçues.



Le capteur PIR



Vue côté composants



Une fois le capot enlevé

Figure II.6. Capteur de mouvement

II.3.9. Caractéristiques électriques et fonctionnelles de capteur PIR

Au moment de la mise sous tension, une durée de stabilisation d'environ 30s est nécessaire avant de pouvoir détecter un mouvement. Les caractéristiques essentielles du PIR sont données par le tableau suivant :

Grandeur	Mini	Typique	Maxi	Unité
Tension d'alimentation du module	3		5	V
Intensité du courant d'alimentation		50		μA

Température de fonctionnement	-20		+80	°C
Niveau de sortie au repos		0.4		V
Niveau de sortie lorsqu'un mouvement est détecté		4		V
Angle de détection		90		Degré
Distance de détection		6		M

Tableau II.1: caractéristiques de capteur PIR

Réglages du PIR

Afin de régler le PIR, on dispose de deux potentiomètres. L'un pour ajuster la sensibilité du capteur et l'autre pour régler le temps de réponse (Figure II.7).

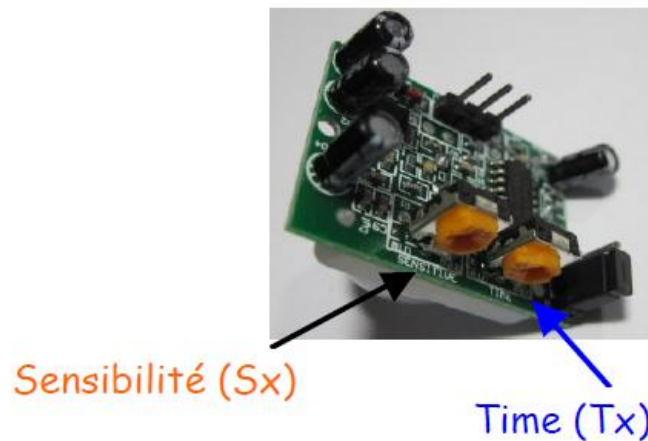


Figure II.7. Réglages du PIR

Sx: = Ajustement de la Sensibilité du capteur de 3-7m, visser pour augmenter

Tx : = Ajustement du délai (Time) pendant lequel la sortie reste verrouillée sur HIGH après une détection de mouvement (Tx). Visser pour augmenter la durée, jusqu'à 200 s.

II.4. Les caméras

C'est un système de prise d'images animées qui génère un signal vidéo noir et blanc ou couleur. La caméra est l'outil qui permet de capter les images en transformant la lumière en message électrique. Elle se compose parfois d'un capteur d'image et d'une électronique de traitement permettant de générer le signal vidéo. On peut trouver actuellement sur le marché plusieurs types de caméras qui peuvent répondre aux besoins d'un utilisateur selon ses objectifs.

II.4.1. Les critères de choix d'une caméra

Le choix d'une caméra de télésurveillance dépend en grande partie des besoins et des exigences.

Outre les différents types de caméras, celles-ci se distinguent par leurs performances concernant :

- ✓ La luminosité : elle s'exprime en « lux ». L'intérêt de certaines caméras est de pouvoir filmer aussi bien dans l'obscurité (0 lux) qu'avec beaucoup de luminosité.
- ✓ La qualité de l'image : plus l'image a de lignes, meilleure est sa qualité.
- ✓ La taille de l'objectif : plus l'objectif est large (de 2,5 mm à 100 mm) plus le champ sera large lui aussi ; plus l'objectif est petit, plus les zones distantes de la caméra seront zoomées.

II.4.2. Différents types de caméras

Nous pouvons trouver actuellement sur le marché pléthore de caméras.

L'objectif n'est pas ici de toutes les énumérer mais de présenter celles que l'on trouve le plus couramment dans le domaine de la vidéosurveillance [9].

➤ Caméra CCD (Charge Coupled Device)

Le capteur CCD ne fait pas de distinction entre les couleurs. L'analyse des couleurs se fait via des filtres qui permettent de récupérer les signaux RVB avant traitement.



Figure II.8. *Caméra CCD*

➤ **Caméra analogique**

Ces caméras sont facilement reconnaissables ; elles ont une sortie de type BNC, la liaison se fait via le câble coaxial.



Figure II.9. *Caméra analogique.*

➤ **Caméra infrarouge**

Si l'on se réfère à la représentation de la courbe photo pique concernant les longueurs d'onde, on s'aperçoit que l'infrarouge est une lumière invisible. Les infrarouges ne se mesurent pas en lux.

Quand utilise-t-on des caméras infrarouges ? Lorsqu'il n'y a pas de lumière.



Figure II.10. *Caméra infrarouge*

➤ **Caméra CMOS**

La fabrication des capteurs CCD requiert une fabrication spécifique engendrant des coûts importants. En revanche, la technologie CMOS est très utilisée dans la fabrication de composants électroniques, ce qui la rend économique pour la fabrication de caméras CMOS. Elles permettent une intégration à très grande échelle et consomment une quantité minime d'énergie par rapport aux CCD ; une caméra peut fonctionner avec une batterie au NiCd pendant une semaine tandis que la caméra CCD ne fonctionnerait que quelques heures.

La caméra CMOS génère un signal comportant du bruit nuisant à la fourniture d'une image de qualité. Cependant des progrès sensibles se font sentir et la caméra CMOS gagne inexorablement du terrain sur sa consœur la caméra CCD.



Figure II.11. *Caméra CMOS.*

➤ **Caméra discrète**

Les progrès des technologies font que de nos jours une caméra peut se loger dans n'importe quel accessoire. Les plus courants sont le détecteur d'intrusion, l'horloge et la tête de détection incendie.



Figure II.12. *Caméra discrète*

➤ **Caméra numérique (caméra IP)**

Une caméra numérique contrairement à sa consœur analogique, ne dispose pas de sortie coaxiale. En revanche elle dispose d'une liaison via une prise RJ 45 qui permet le raccordement au réseau informatiùque.



Figure II.13. *Caméra numérique*

Parmi ces différents types de on a utilisée la caméra ov2640

II.4.3.La caméra ov2640

L'ov2640 est un capteur d'image CMOS basse tension qui offre toutes les fonctionnalités d'une caméra UXGA mono-puce et d'un processeur d'images dans un boîtier compact.

L'ov2640 fournit des images 8 bits /10bits, sous-échantillonnées, mises à l'échelle ou des images binaires dans une large gamme de formats, contrôlées par l'interface du bus de contrôle de la caméra série (SCCB).

Ce produit dispose d'une matrice d'images pouvant fonctionner jusqu'à 15 images par seconde en résolution UXGA (1600*1200) pixel avec un contrôle complet de la qualité d'image, du formatage et du transfert des données de sortie. Toutes les fonctions de traitement d'image requises, y compris contrôle d'exposition, gamma, balance des blancs, saturation des couleurs, contrôle de la teinte, annulation des pixels blancs, suppression du bruit, l'ov2640 inclut également un moteur de compression pour une puissance de traitement accrue. En outre, la caméra omnivision utilise une technologie de capteur propriétaire pour améliorer la qualité de l'image en réduisant ou en éliminant les sources d'éclairage / électriques courantes, telles que le bruit de fond fixe, etc [10].

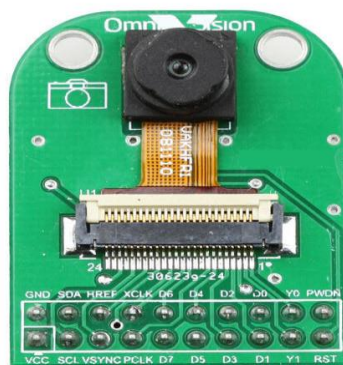


Figure II.14. La caméra ov2640

II.4.3.Identification des pins

Le tableau suivant montre les différents pins de la caméra OV2640

Broche	Nom de pin	Type	La description
1	VCC	PUISSANCE	Alimentation 3.3v

2	GND	Terre	Masse d'alimentation
3	SCL	Contribution	Horloge d'interface série à deux fils
4	SDATA	Bi-directionnel	E / S de données d'interface série à deux fils
5	VSYNC	Sortie	Actif haut: cadre valide; indique le cadre actif
6	HREF	Sortie	Actif haut: ligne / données valides; indique les pixels actifs
7	PCLK	Sortie	Pixel Clock sortie du capteur
8	XCLK	Contribution	Master Clock dans le capteur
9	DOUT9	Sortie	Sortie de données pixel 9 (MSB)
10	DOUT8	Sortie	Sortie de données pixel 8
11	DOUT7	Sortie	Sortie de données pixel 7
12	DOUT6	Sortie	Sortie de données pixel 6
13	DOUT5	Sortie	Sortie de données pixel 5
14	DOUT4	Sortie	Pixel Data Output 4
15	DOUT3	Sortie	Sortie de données pixel 3
16	DOUT2	Sortie	Pixel Data Output 2 (LSB)
17	DOUT1	Sortie	Pixel Data Output 1 (mode 10 bits)
18	DOUT0	Sortie	Pixel Data Output 0 (mode 10 bits)
19	RST	Contribution	Réinitialisation de la caméra, active basse
20	PWDN	Contribution	Appareil photo éteint, actif haut

Tableau II.2 : *déférentes pins de la caméra OV2640*

II.5. ArduCAM shield V2

Arducam série caméra shield est une carte de contrôle de caméra universelle pour Arduino. Il fournit une interface de contrôle de caméra facile à utiliser ainsi qu'une bibliothèque de codes source prêts à l'emploi. L'Arducam prend en charge divers modules de caméra de 0.3MP à 5MP avec des formats différents comme RAW, RGB, YUV, JPEG et peut être bien jumelé avec les cartes Arduino standard [11].

La figure suivante présente la carte ArduCAM Shield V2 :



Figure II.15. *ArduCAM shield V2*

ArduCAM Shield V2 est la deuxième génération de produits ArduCAM, ils offrent des performances améliorées et des fonctions améliorées que les révisions précédentes. Ils peuvent non seulement capturer des images JPEG 5MP décentes, mais aussi effectuer une capture d'image RAW de résolution 5MP complète, ce qui en fait la solution idéale pour les applications IoT et de traitement d'images . Encore plus, ils peuvent aussi enregistrer de courts clips.

II.5.1.Caractéristiques

- Supporte les modules caméra 0.3MP ~ 5MP,
- Construire dans une prise de carte SD / TF
- Soutenir le mode de compression de JPEG,
- Soutenez presque toute la plate-forme de microcontrôleur,
- Fournir une bibliothèque de code open source,
- Tous les ports IO d'Arducam sont tolérants 5V / 3.3V,
- Bien accouplé avec les cartes Arduino standard.
- Alimentation 5V/50mA.

II.5.2.Identification des pins

La figure II.16 montre le schéma de brochage du blindage ArduCAM V2. Il y a quatre connecteurs sur le bouclier ArduCAM. L'un est un connecteur ArduCAM standard qui peut bien s'accoupler avec la carte Arduino. Le second connecteur P6 est un port alternatif pour les broches Arduino, certains d'entre eux sont utilisés par le blindage ArduCAM, les

autres sont des broches libres. Le troisième connecteur est destiné au module de caméra. Le quatrième connecteur U5 est pour l'extension de module ArduCAM Nano ESP8266 [11].

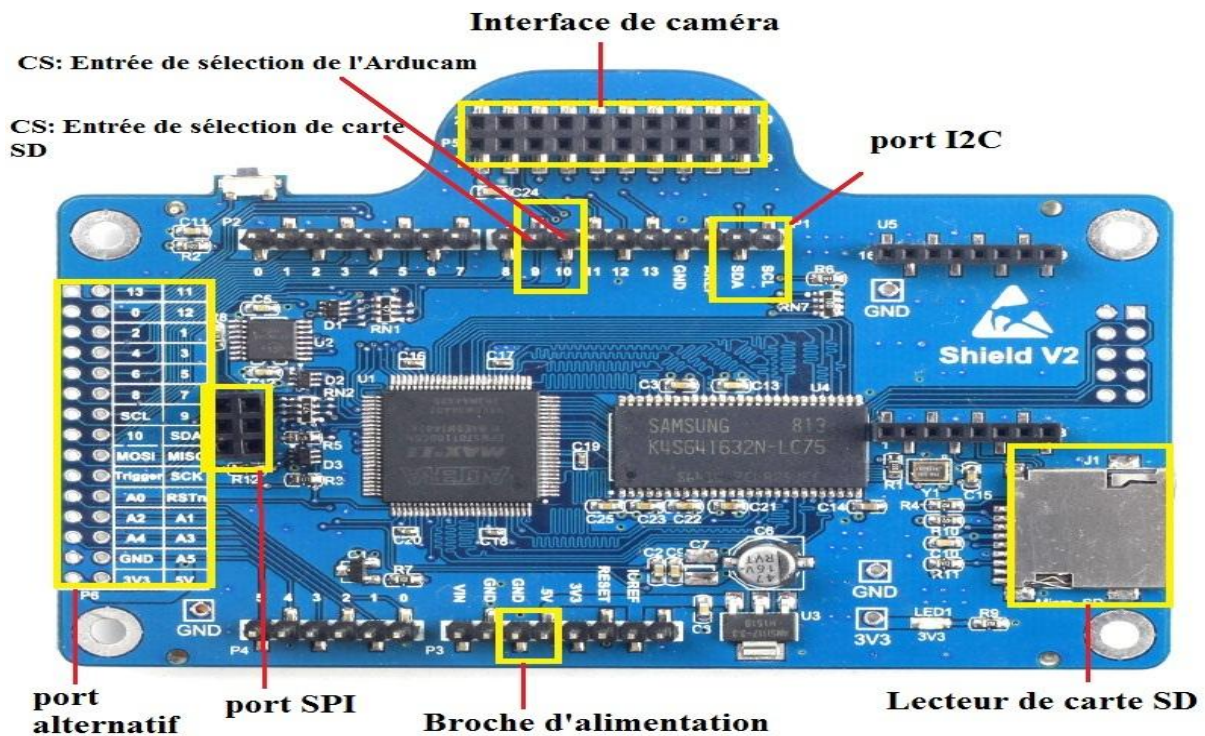


Figure II.16. Architecture d' ArduCAM shield V2

➤ Définition du Pins ArduCAM Shield V2

Le tableau suivant présente les différents pins d'Arducam :

Nom du pin	Type	Emplacement (Arduino standard)	Description
CS	Entrée	10	Entrée de la puce esclave SPI
MOSI	Entrée	ICSP-MOSI	Entrée esclave de sortie maître SPI
MISO	Sortie	ICSP-MISO	Sortie esclave d'entrée maître SPI
SCLK	Entrée	ICSP-SCLK	Horloge série SPI
SDA	Bidirectionnel	Arduino standard	Interface de données série à deux fils
SCL	Entrée	Arduino standard	Horloge d'interface série à deux fils
SD/TF CS	Entrée	9	Entrée de sélection de carte SD

Tableau II.3: Pins ArduCAM Shield V2

➤ **Définition de la broche P6 alternative**

Pin	Arduino standard	Pin	Arduino standard
1	13	2	11
3	0	4	12
5	2	6	1
7	4	8	3
9	6	10	5
11	8	12	7
13	SCL	14	9
15	10	16	SDA
17	MOSI	18	MISO
19	Trigger	20	SCK
21	A0	22	RSTn
23	A2	24	A1
25	A4	26	A3
27	GND	28	A5
29	3.3V	30	5V

Tableau II.4 : *pins de la broche P6 alternative*

➤ **Définition de la broche d'interface de caméra**

Pin	Nom	Pin	Nom
1	3.3V	2	GND
3	SCL	4	SDA
5	VSYNC	6	HREF
7	PCLK	8	XCLK
9	D7	10	D6
11	D5	12	D4
13	D3	14	D2
15	D1	16	D0
17	NC	18	NC

19	NC	20	NC
----	----	----	----

Tableau II.5 : pins de la broche d'interface de caméra

II.6. Module WiFi ESP32

Les module WiFi qui contient un micro contrôleur ESP32 sont programmables comme les cartes Arduino, et peuvent communiquer en wifi avec d'autres appareils (ordinateurs, Smartphones, etc.). l'ESP32 est une série de système à faible coût et faible consommation sur un microcontrôleur à puce avec WIFI et Bluetooth, cette puce est Dual Core (signifie qu'il dispose de 2 processeurs), Il offre deux unités de traitement 32 bits qui fonctionnent à une fréquence plus rapide, l'un est responsable de la connectivité Wifi et l'autre

est disponible pour exécuter notre programme [12].

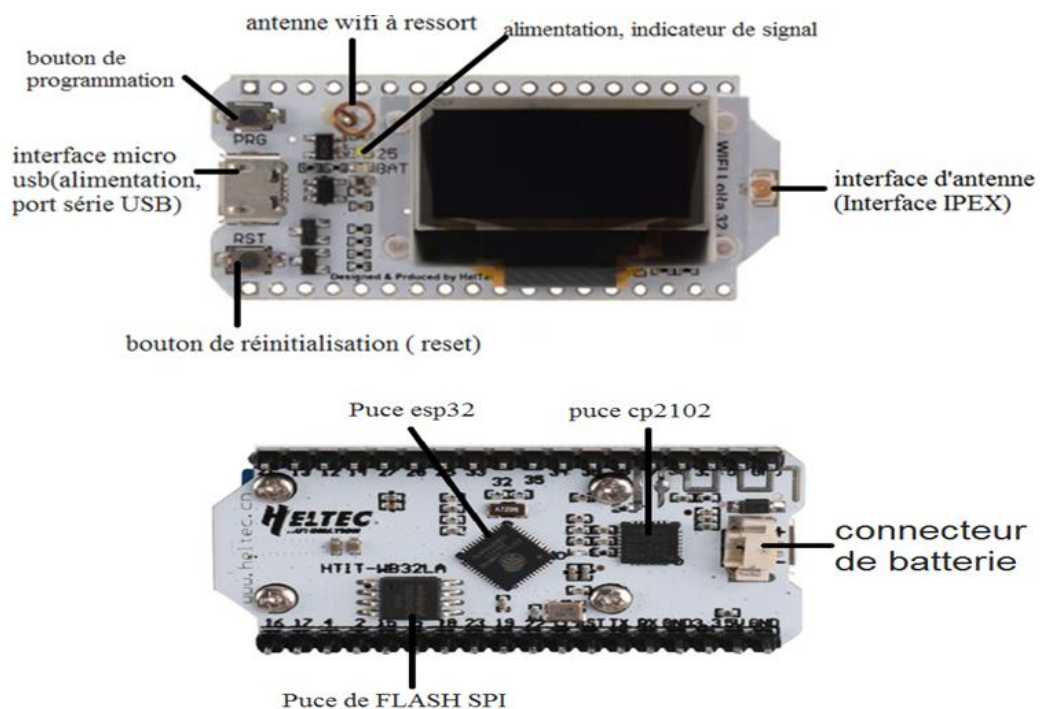


Figure II.16. Module WIFI ESP32

II.6.1. Caractéristiques:

- ESP32 est dual core, cela signifie qu'il dispose de 2 processeurs.
- Il a Wifi et Bluetooth intégré.

- Il exécute des programmes de 32 bits.
- La fréquence d'horloge peut aller jusqu'à 240 MHz
- Il offre 512 Ko de mémoire vive(RAM)
- Il a également une grande variété de périphériques disponibles, tels que: tactile capacitif, CAN, DAC, UART, SPI, I2C [12].

II.7. Discussion

Dans ce chapitre, nous sommes intéressées à l'étude et la présentation de la télésurveillance et ça commence par sa définition, son domaines d'applications, ces avantages et inconvénients ainsi les différents équipements nécessaires pour sa mise en place. Et à la fin on a présenté quelques types de cameras et capteurs utilisés dans la télésurveillance.

CHAPITRE III

La réalisation pratique

III .1. Préambule

Dans les chapitres précédents, nous avons défini les différents composants qui constituent notre système de télésurveillance ainsi que le logiciel Arduino qui va nous permettre de programmer notre carte Arduino Méga. Dans ce chapitre, nous allons présenter le système de surveillance réalisé. Dans un premier temps, nous présenterons le brochage de chaque composant avec la carte Arduino et nous testerons leur bon fonctionnement. Puis, en tenant compte des tests effectués, nous allons faire un assemblage de tous ces composants avec l'Arduino.

III.2. Réalisation matérielle

III .2.1. Schéma synoptique

Le but de ce projet est de réaliser un système de surveillance qui détecte des mouvements sur une distance de quelques mètres et prend des photos de ce qui se passe dans le champ de la caméra. Ces photos seront enregistrées sur la carte mémoire de type SD et au même temps transférées vers un serveur web via le wifi.

La figure suivante présente le schéma synoptique de notre système :

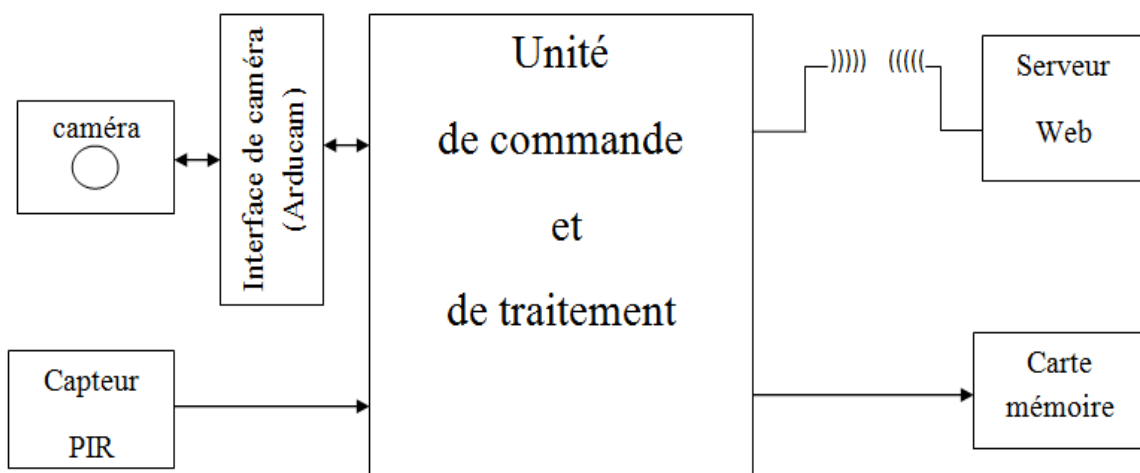


Figure III.1. Schéma synoptique

III .2.2. Différents éléments utilisés dans le projet

Pour réaliser notre système de télésurveillance on a utilisé les composants suivants : une carte Arduino méga, une caméra ov2640, l’Arducam shield V2 et le module wifi esp32.

III .2.2.1. Branchement de la caméra OV2640 sur l’Arducam

Ce module a fait l’objet d’une étude dans le chapitre 2. Dans ce qui suit nous présentons le brochage des différents pins d’ov2640 sur les pins d’interface de la caméra qui se trouve sur l’Arducam shield V2.

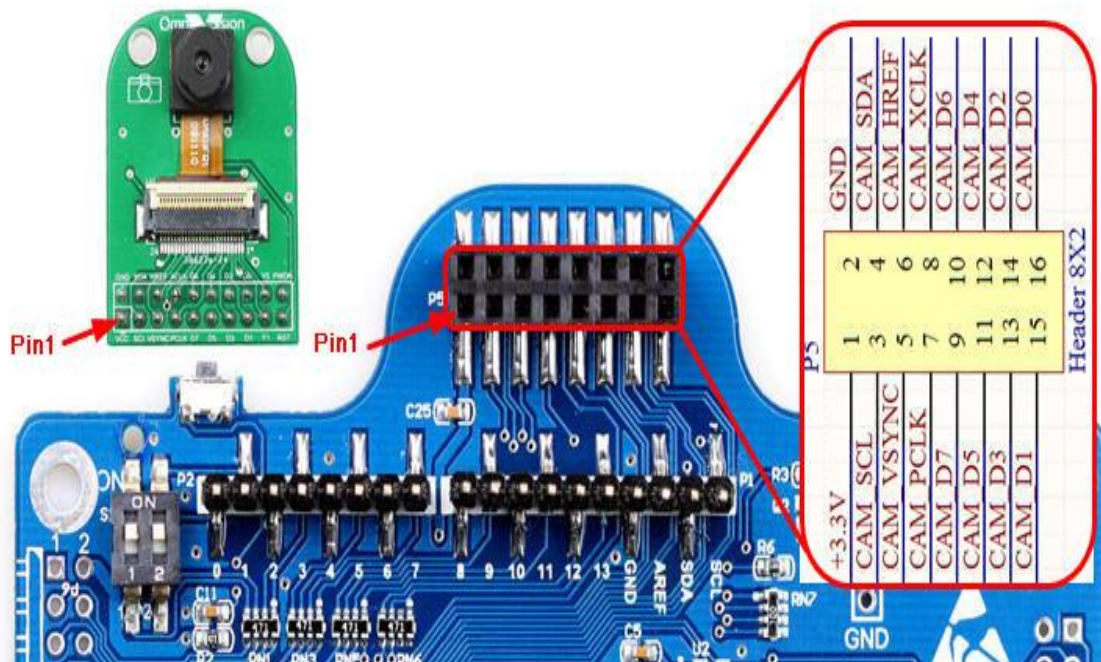


Figure III.2. Branchement de la caméra OV2640 sur l’Arducam

III .2.2.2. Branchement de l’Arducam shield v2 à la carte Arduino

La figure suivante présente le branchement de différents pins de l’Arducam avec l’Arduino :

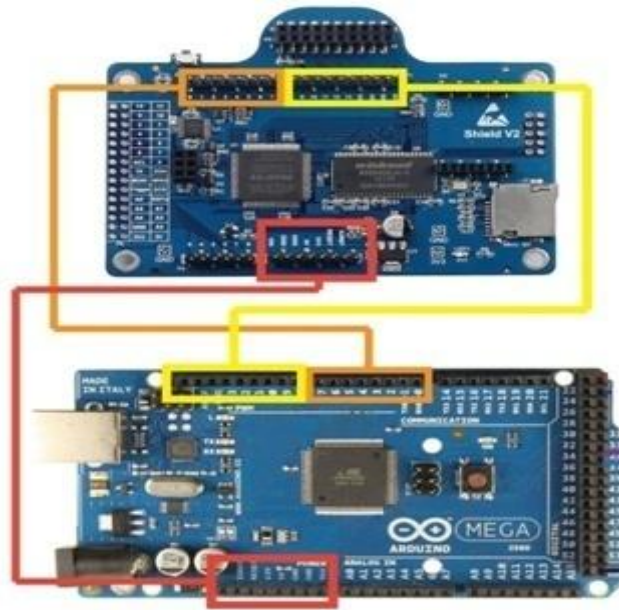


Figure III.3. Branchement de l'Arducam avec Arduino

III .2.2.3. Branchement du capteur PIR à la carte Arduino

Les capteurs PIR (capteur infrarouge passif) permettent de détecter la présence d'humains mobiles dans le champ du capteur. Ils sont utilisés dans divers systèmes de sécurité et détecteurs de mouvements. Ils sont miniatures, peu coûteux et ne consomment pas beaucoup d'énergie.

Le capteur PIR est un capteur numérique :

- Si un mouvement est détecté le signal en sortie du capteur est mis au niveau **haut (1)**
- Si aucun mouvement n'est détecté le signal en sortie du capteur est mis au niveau **bas (0)**

Le branchement du capteur PIR à la carte Arduino Méga est extrêmement simple, il suffit en effet de connecter la broche + (VCC) du capteur au 5V de l'Arduino , la broche GND (-) à la masse de l'Arduino (GND), et la broche out/signal (le signal de sortie) du capteur à l'entrée (digital) de la carte Arduino. Comme le montre la figure suivante :

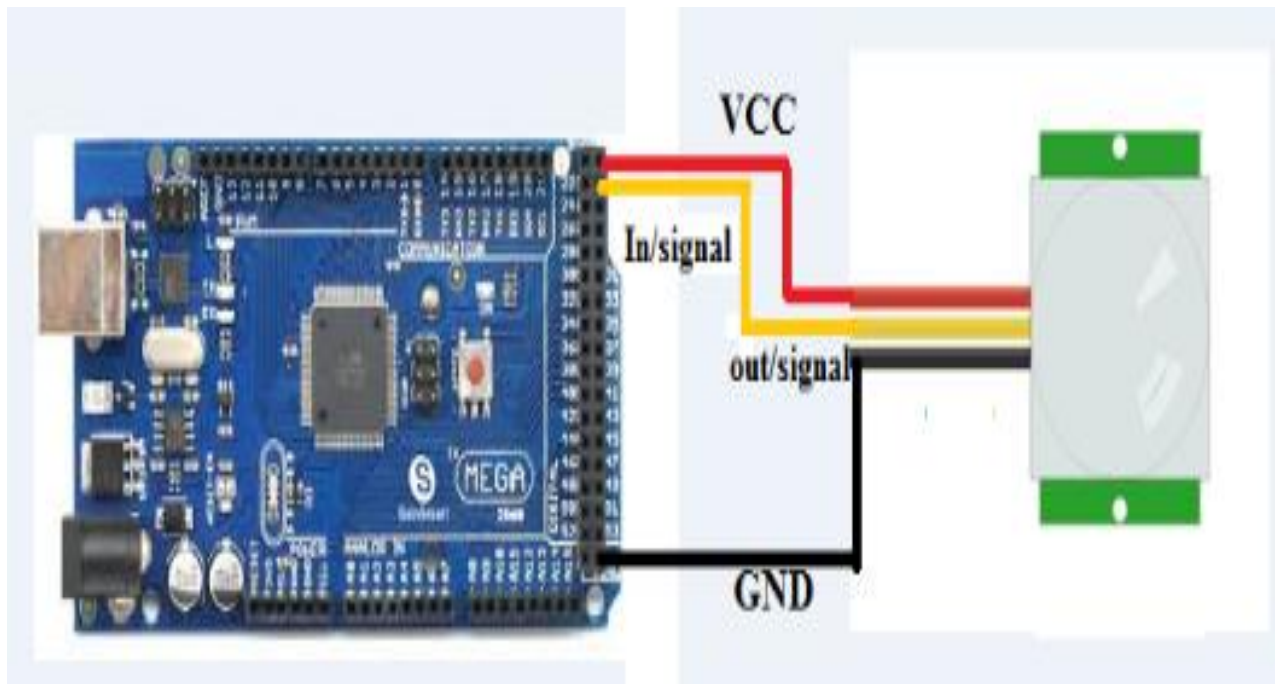


Figure III.4. Branchement de capteur PIR

III .2.2.4. Branchement d'Esp32 avec l'Arduino

La figure suivante présente le schéma de câblage d'Esp32 avec l'Arduino

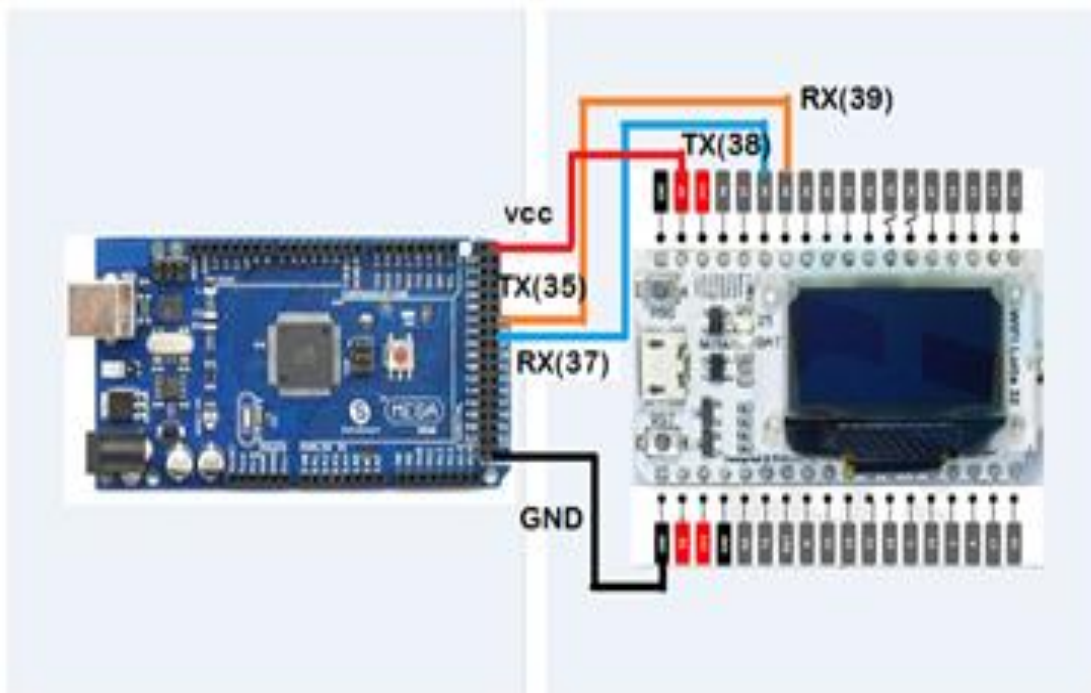


Figure III.5.Schéma de câblage d'Esp32 avec l'Arduino

III.2.3. Câblage du système

En tenant compte des tests de bon fonctionnement de chaque élément seul, le brachement général des différents composants vers la carte Arduino Méga a été effectué comme la montre la figure ci-dessous :

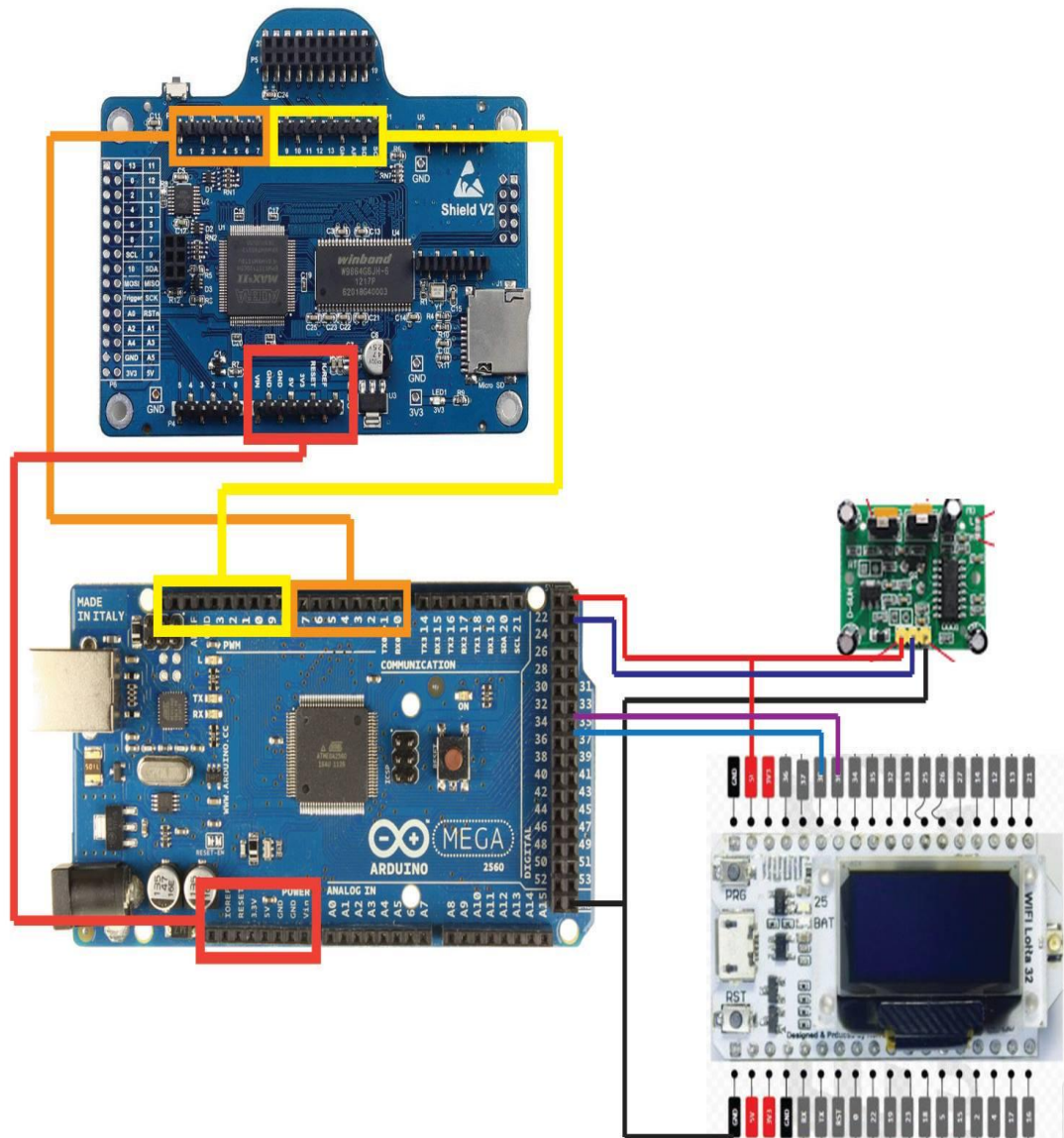
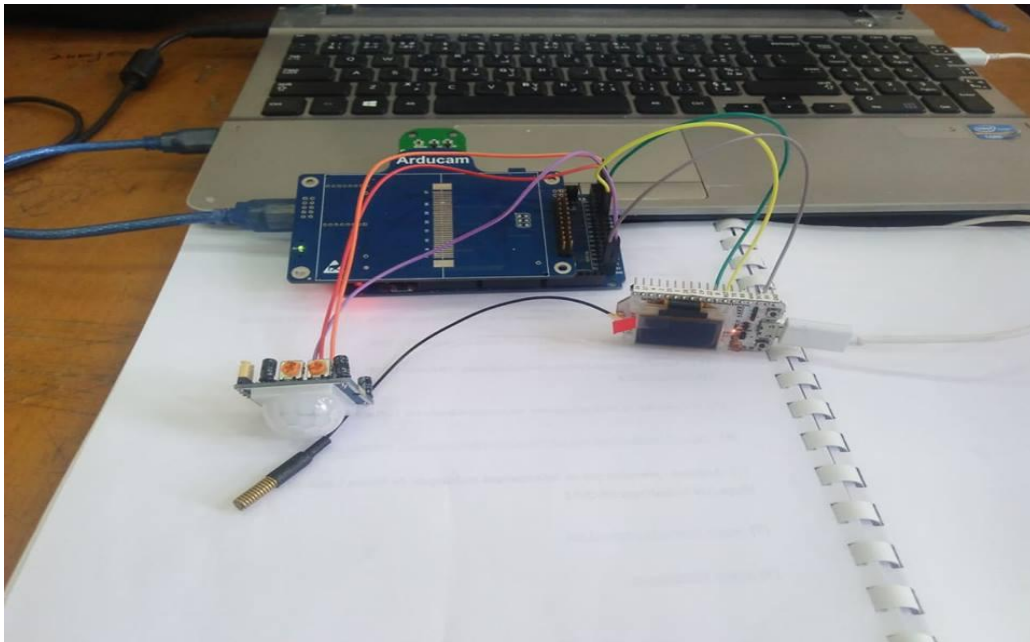


Figure III.6. Câblage de système

La figure III.7 représente le système de surveillance réalisé pour les tests.



La figure III.7. *Montage de notre système*

III .2.4. L'organigramme

Avant de passer à la programmation, nous devons réaliser un organigramme qui explique le déroulement des différentes séquences tant intérieures qu'extérieures : il comporte plusieurs boucles dont la fin d'exécution succède toujours à son commencement.

Chaque action est symbolisée par un rectangle et chaque choix est symbolisé par un losange.

L'organigramme de notre système est présenté par la figure suivante

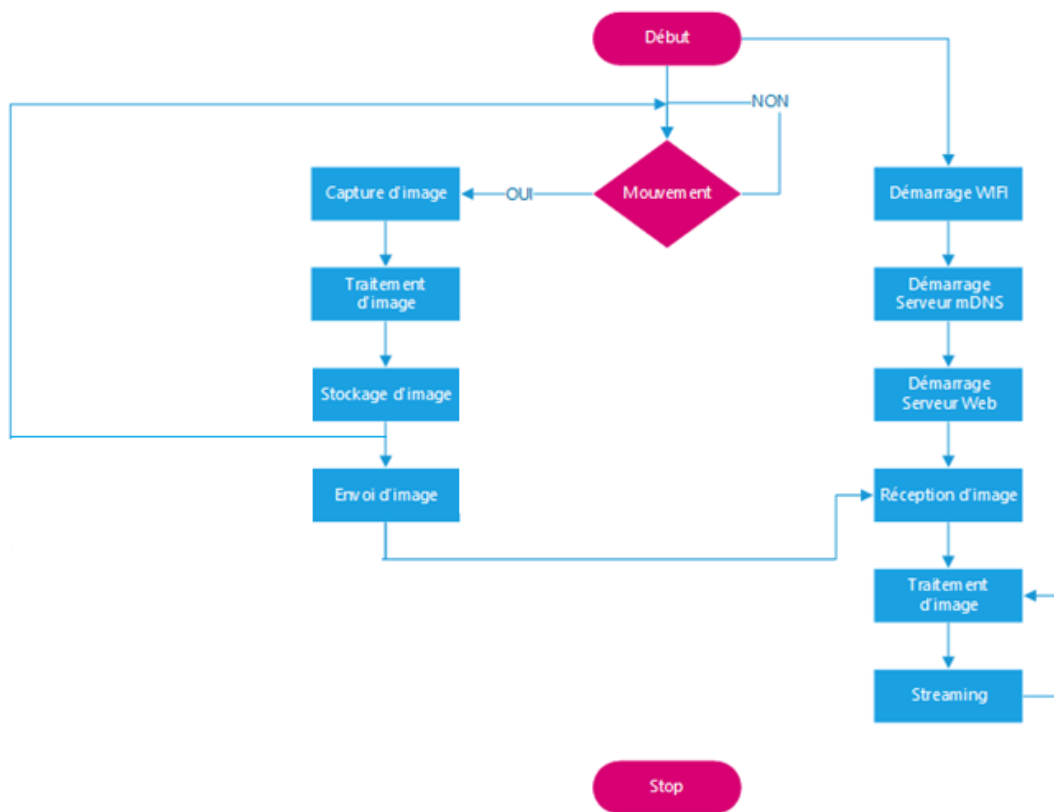


Figure III.8. L'organigramme du fonctionnement général.

III .2.5. Les étapes de programmation de la carte Arduino

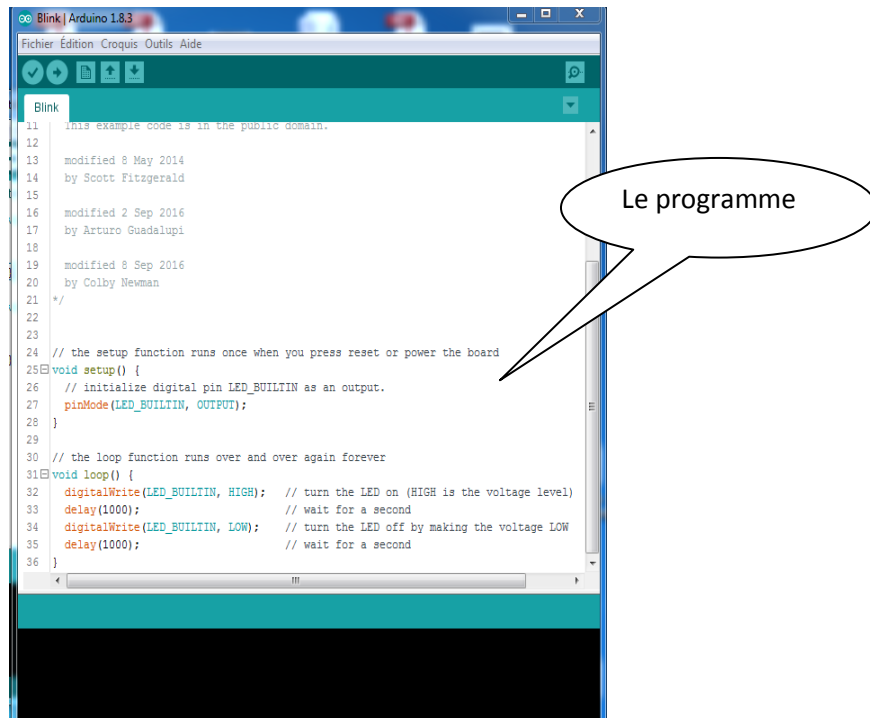
Pour programmer l'Arduino on doit d'abord installer l'environnement de développement Arduino (IDE) qui vas nous permettre de programmer la carte Arduino, et branché la carte à l'ordinateur via un câble USB.

- **Démarrage de logiciel Arduino et l'écriture de programme**

Le logiciel Arduino est un environnement de développement intégré, ou EDI, qui se présente sous la forme d'une interface graphique.

Après l'ouverture de l'interface IDE on va saisir le programme.

La figure suivante présente l'interface de l'IDE d'Arduino



```
11 // This example code is in the public domain.
12
13 // modified 8 May 2014
14 // by Scott Fitzgerald
15
16 // modified 2 Sep 2016
17 // by Arturo Guadalupi
18
19 // modified 8 Sep 2016
20 // by Colby Newman
21 */
22
23
24 // the setup function runs once when you press reset or power the board
25 void setup() {
26   // initialize digital pin LED_BUILTIN as an output.
27   pinMode(LED_BUILTIN, OUTPUT);
28 }
29
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
33   delay(1000); // wait for a second
34   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
35   delay(1000); // wait for a second
36 }
```

Figure III.9. L'écriture du programme Arduino

- **Compilation du programme**

Permet de vérifier que notre programme est bien conforme à la syntaxe (règles d'écritures), et que toutes les variables sont bien définies, ainsi que les appels aux bibliothèques ou autres fichiers. En cas d'anomalie dans le programme, le compilateur nous enseigne sur le type d'erreur et la ligne où elle se trouve.

- **Sélection de type de carte**

Avant de téléverser le code vers la carte Arduino, il faut confirmer et sélectionner le type de la carte qu'on a utilisé. Dans la barre des menus cliquez sur « Outils » puis sur « type de carte » et sélectionner la carte. (Dans notre cas on coche sur Mega 2560). comme le montre la figure suivante

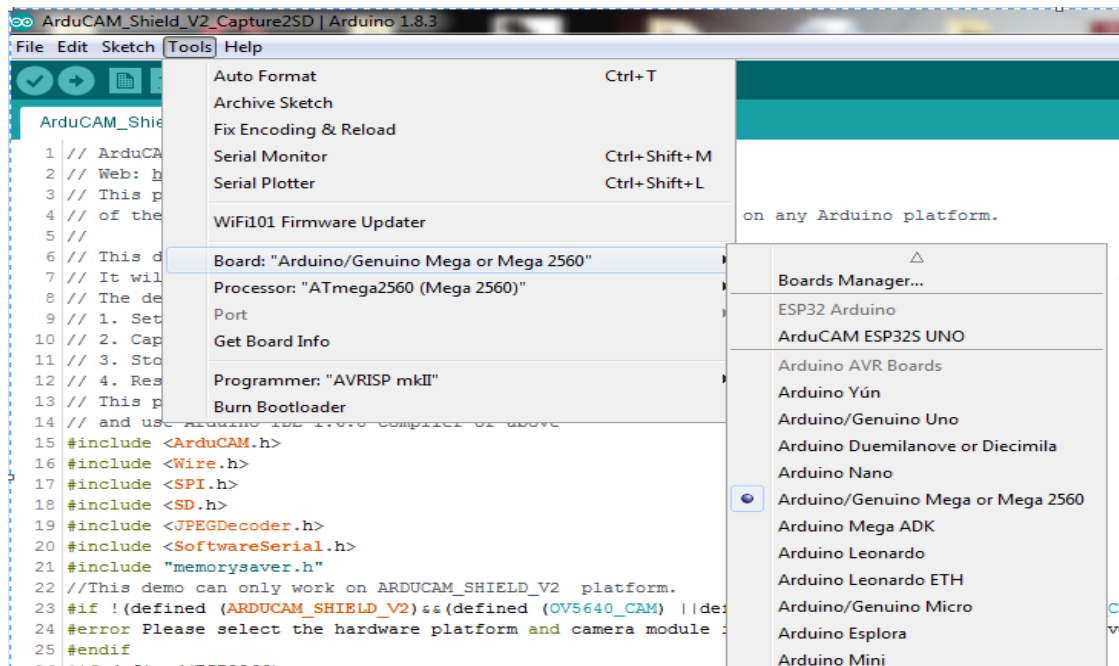


Figure III.10. Sélection de la cible

- **Sélection du port**

Le port : est la connexion qui permet à l'ordinateur et au matériel Arduino de communiquer.

Série : décrit la manière dont les données sont envoyées, bit (0 ou 1) après bit

Pour déterminer le port série, nous sélectionnons depuis le menu Outils->Port, comme le montre la figure suivante :

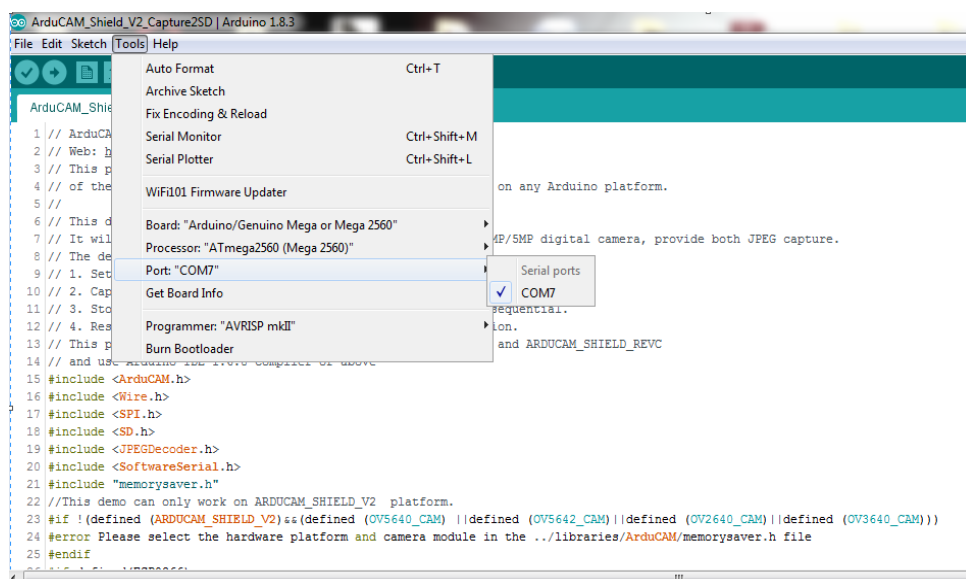


Figure III.11. Sélection du port

- Transfert du programme

Après les étapes passées le programme est envoyé dans la carte. Pour ce faire, il suffit de cliquer sur le bouton *Upluad* (ou « téléverser »)

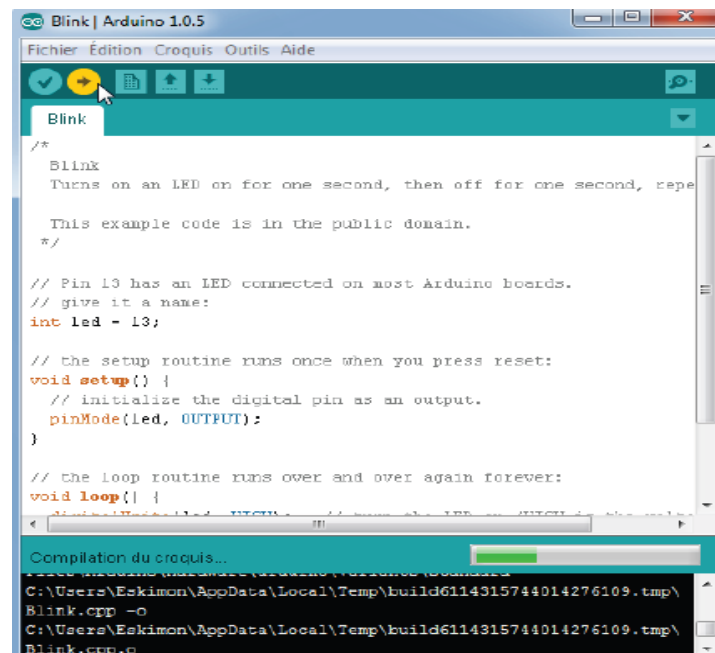


Figure III.12. Transfert du programme

III .2.6. Bibliothèques

Les bibliothèques fournissent des fonctions nouvelles qui peuvent être utilisées dans les programmes ;pour utiliser une bibliothèque il faut la sélectionner depuis le menu Sketch > Import library(Programme >Importer bibliothèque).cela insérera une ou plusieurs instructions **#include** au début du programme et compilera la bibliothèque avec le programme .

Puisque les bibliothèques sont transférées dans la carte avec le programme, elles augmentent la quantité de mémoire utilisée. Si un programme ne nécessite plus d'une bibliothèque, il faut effacer simplement l'instruction **#include** correspondante au début du code.

Il ya a toute une liste **de bibliothèque dans la référence du langage Arduino**. Certaines nombre de bibliothèque sont incluses dans le logiciel Arduino et d'autres peuvent être téléchargées depuis différentes source (voir notamment : [http : // www.github.com](http://www.github.com))

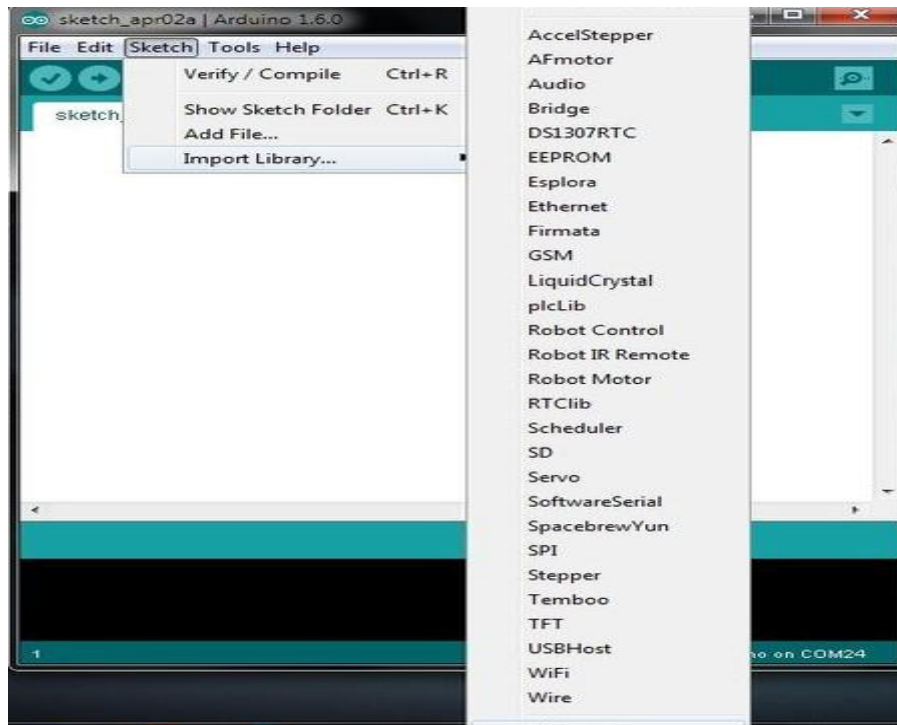


Figure III.13. Choix de librairie

III .2.7. Moniteur série

Un moniteur ou terminal est un écran/clavier relié à un microcontrôleur pour interagir avec un utilisateur. Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d’afficher sur l’ordinateur l’état de variable, de résultats de calculs ou de conversion analogique- numérique. C’est un élément essentiel pour améliorer, tester et corriger les programmes

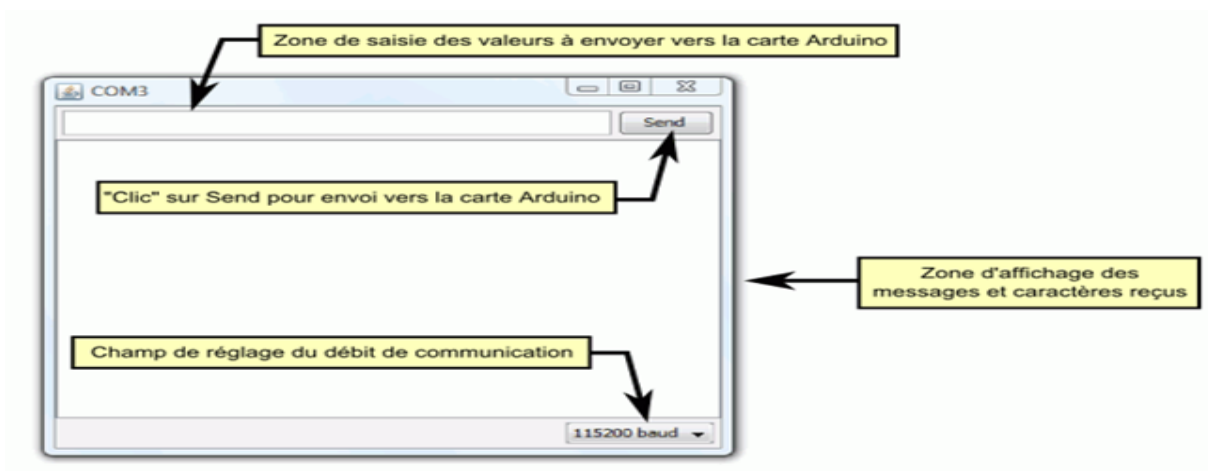


Figure III.14. Présentation du terminal série

Pour ouvrir le terminal série : il faut cliquer sur le serial monitor



Figure III.15. Ouverture de terminal série

Pour régler le débit de communication sur la même valeur que celle utilisée par le programme avec lequel programmer la carte Arduino

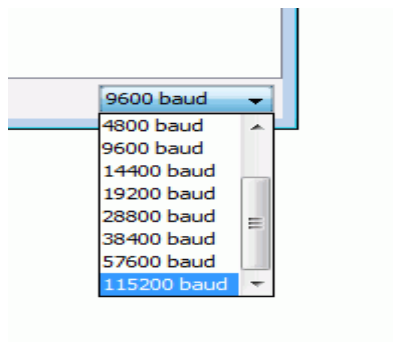


Figure III.16. Ajustage du débit de la communication série

III.3. Tests de l'application

Pour vérifier le fonctionnement de notre système nous avons réalisé des tests. Après avoir détecté un mouvement par le capteur infrarouge, le capteur d'images qui se trouve dans le module de camera reçoit des ordres pour prendre des photos par l'Arduino. Les images acquises seront enregistrées dans la carte SD.

Au même temps, ces images seront transférés via le bus UART de l'Arduino vers le module wifi esp32 pour les affichées sur un mini serveur Web.

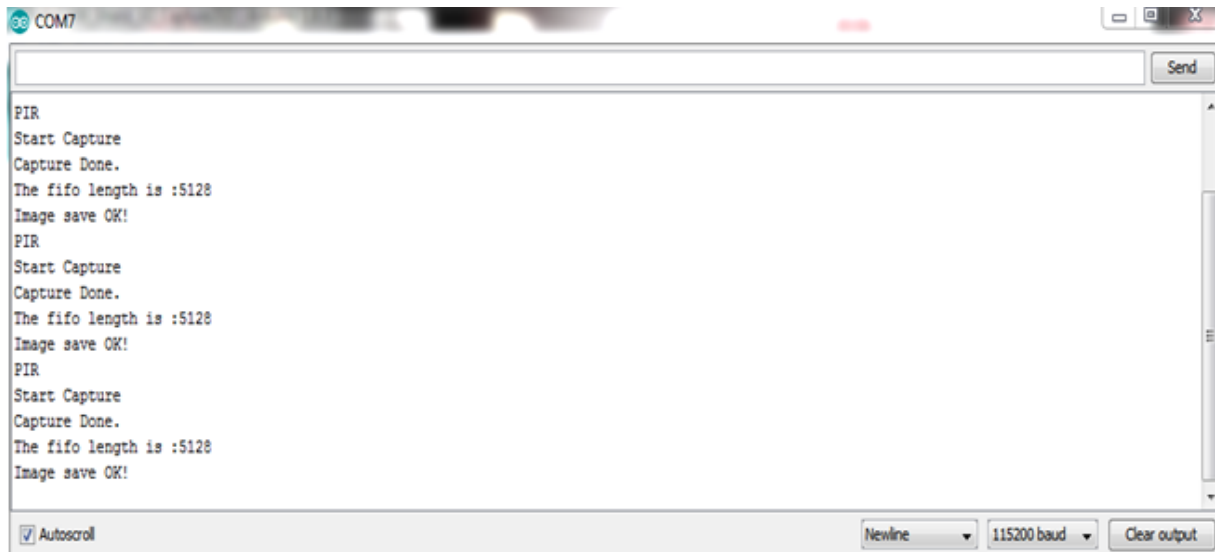


Figure III.17. Fenêtre de logiciel lors de détection de mouvement

Les images capturées par la caméra

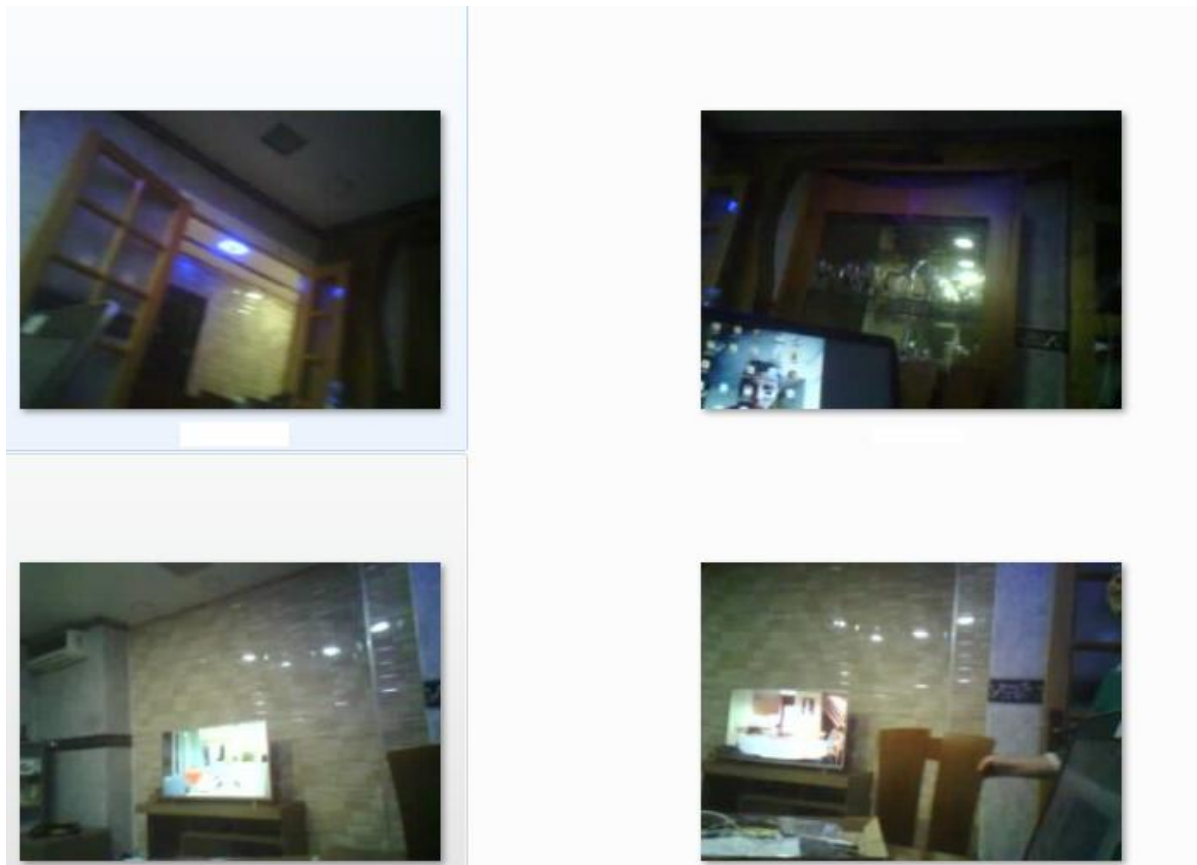


Figure III.18. Les images enregistrées sur la carte SD

L'application ArduCAM_Host_V2 est utilisée afin de visualiser et traiter les images en temps réels.

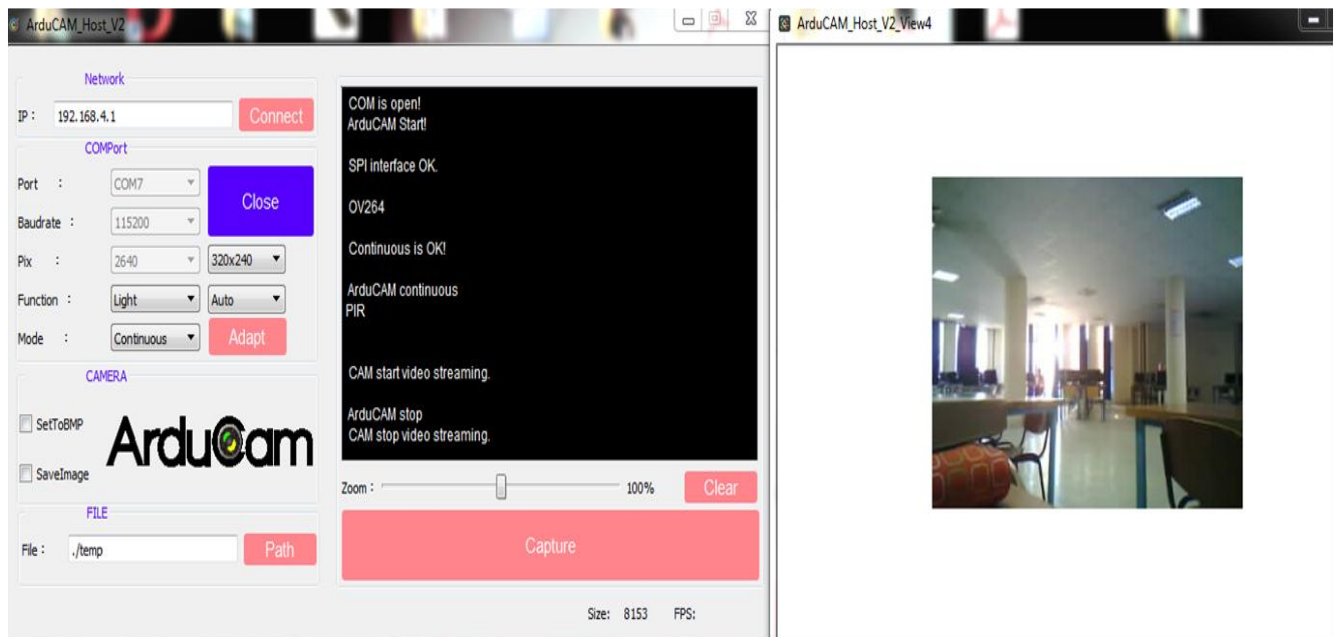


Figure III.19. Message affiché sur l'application

III .4. Serveur Web

Un serveur web a été réalisé afin d'accueillir les images acquises. Ce serveur permet à des clients d'accéder à des pages web, c'est-à-dire des fichiers au format HTML à partir d'un navigateur installé sur un ordinateur distant. Un serveur web est capable d'interpréter les requêtes HTTP arrivant sur le port associé au protocole HTTP (par défaut le port 80), et de fournir une réponse avec ce même protocole [13].

Afin de créer un serveur Web autonome avec un ESP32 nous avons suivis les étapes suivantes :

- Installation de la carte ESP32 dans Arduino IDE
- Ecriture du code sur IDE permettant de créer le serveur web
- Définition des informations d'identification du réseau : on doit définir le SSID et le mot de passe pour y accéder.
- Téléchargement ce code sur l'Esp32
- Ouverture du moniteur série à un débit de 115200.

La figure suivante montre l’affichage (sur le moniteur série) de l'adresse IP qu'on va utiliser pour accéder au serveur Web ESP32 :

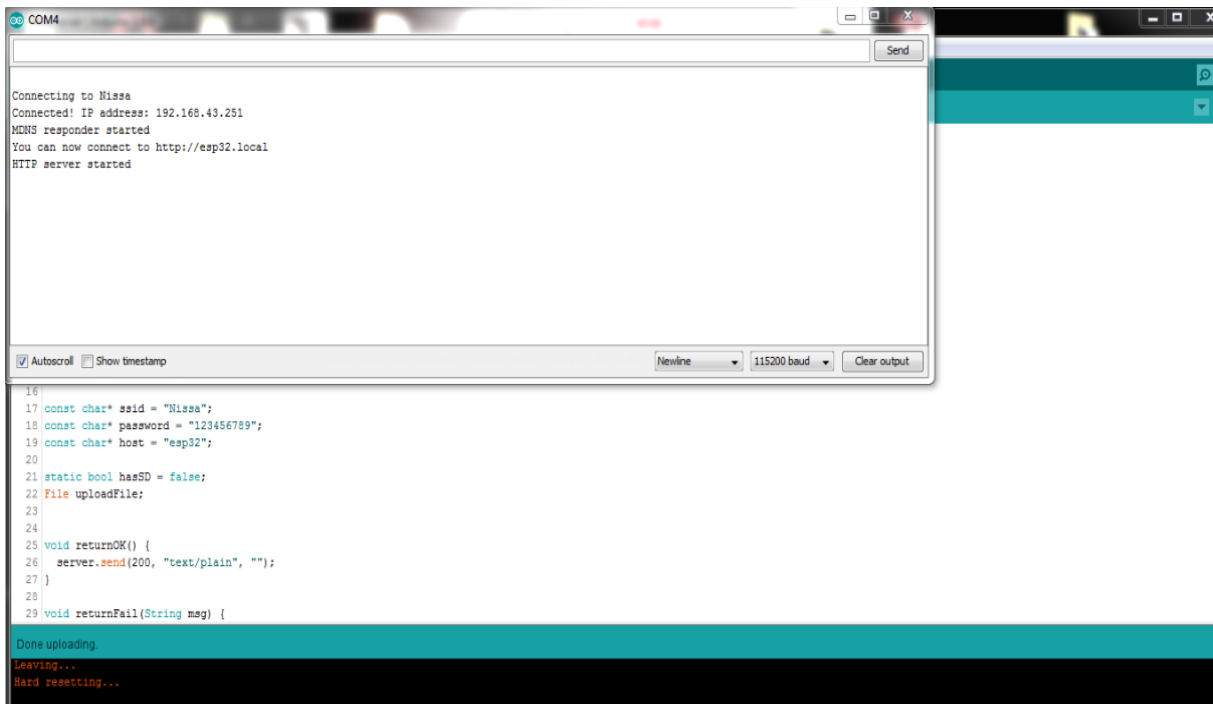
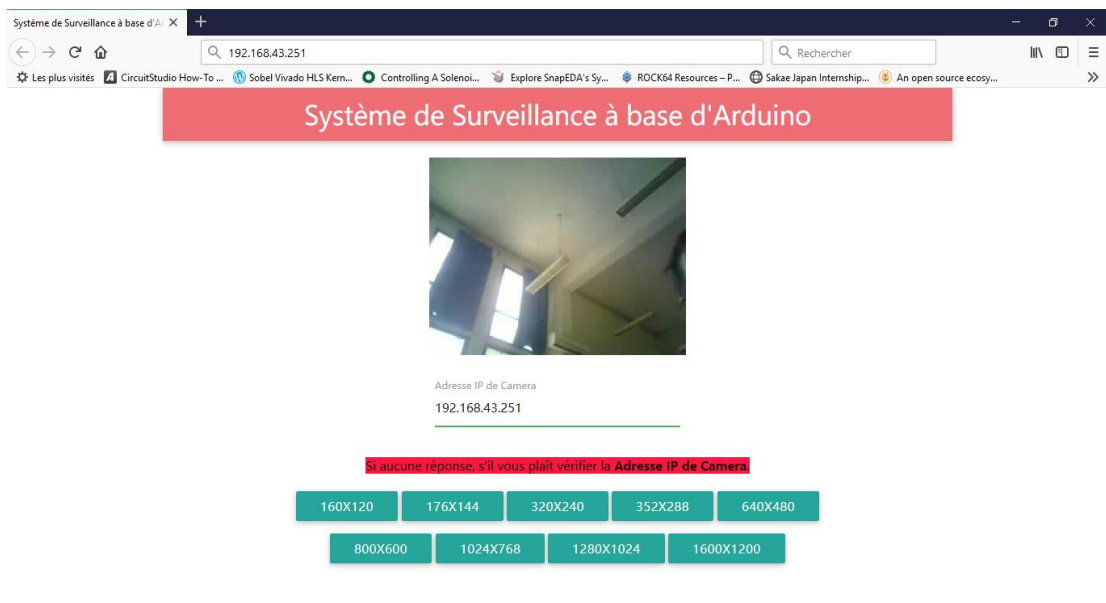


Figure III.20. L’affichage de l'adresse IP sur le moniteur série

Pour accéder au serveur Web, on ouvre le navigateur, puis dans la barre d’adresses nous saisissons l’adresse IP du serveur web réalisé : 192.168.43.251



III .5. Discussion

La carte Arducam est assez simple pour être utilisée avec un Arduino. En effet, Arducam prend en charge plusieurs modules de camera de 0.3Mp à 5Mp comme ov2640 qui nous permet de prendre des photos, ensuite les envoyer au serveur Web via le module wifi qui a présente l'avantage de la transmission à distance.

Les tests que nous avons effectués, montrent le bon fonctionnement de notre réalisation. Toutefois dans le cas du serveur Web, nous avons constaté quelques bugs. En effet, les images transmises vers ce serveur via le module WiFi ne s'affichent pas correctement à chaque fois. Ce problème est dû probablement au volume de données important envoyées avec le module WiFi.

Conclusion Générale

Conclusion

Conclusion

Dans ce mémoire nous avons présenté le système de surveillance réalisé. Ce système est basé sur l'utilisation de la carte Arduino qui a pour rôle la commande du système et le traitement.

Pour réaliser notre système de surveillance, nous avons d'abord testé le bon fonctionnement des différents éléments et la communication avec la carte centrale qui est Arduino. Puis nous avons assemblé tous ces éléments.

Les tests que nous avons effectués démontrent le bon fonctionnement de notre réalisation toutefois dans le cas du serveur Web, nous avons détecté des bugs. En effet, les images transmises vers ce serveur via le module WiFi ne s'affichent pas correctement à chaque fois. Ce problème est dû probablement au volume de données importantes envoyées avec le module WiFi.

Les résultats obtenus à travers cette réalisation sont satisfaisants, en capture et en transmission, ce qui rend notre système prêt à l'emploi. Toutefois, notre système est sujet à des améliorations.

Les perspectives de ce travail est l'utilisation d'un servomoteur permettant à la caméra de suivre la cible en mouvement. De plus, la réalisation de plusieurs modules de surveillances et la connexion vers le même serveur Web permettra de surveiller plusieurs endroits au même temps (les différents bureaux d'une entreprise). Et prévoir un système d'alarme pour avertir les personnes concernées.

Bibliographie

- [1] : Erik Bartmann, 2015. Le Grand livre d'Arduino, Edition Eyrolles.
- [2] : Simon Landraut et Hippolyte Weisslinger, 2014. Arduino : premier pas en informatique embarquée. Le blog d'Eskimon.
- [3] : http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki. Consulté le 15/06/2018.
- [4] : Christian Tavernier, 2014. Arduino : Maitrisez sa programmation et ses cartes d'interface (shields) . Edition Dunod.
- [5] : <http://dspace.univ-tlemcen.dz/bitstream/112/11010/1/Application-de-telesurveillance.pdf>
- [6] : <http://dictionnaire.sensagent.leparisien.fr/t%C3%A9l%C3%A9surveillance/fr-fr/>
- [7] : http://gte.univ-littoral.fr/sections/documents/Les_capteurs.pdf?nocache=1289041293.82
- [8] : http://ref11.r-e-f.org/pluginAppObj_501/PIR.pdf
- [9]: https://www.referentsurete.com/wa_files/guide_20video_20ip.pdf
- [10] : https://www.uctronics.com/download/cam_module/OV2640DS.pdf
- [11] : http://www.arducam.com/downloads/shields/ArduCAM_Camera_Shield_V2_DS.pdf
- [12] : www.esp32.net
- [13] : <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>

Résumé

Ce projet de fin d'étude porte sur la conception et la réalisation d'un système de télésurveillance basée sur l'utilisation de la carte Arduino combiné avec l'Arducam.

L'objectif de cette réalisation est de surveiller un lieu à distance en temps réel via une caméra. Celle-ci se déclenchera grâce à un détecteur de mouvement. Les images acquises seront enregistrées sur une carte mémoire et au même temps transmises vers un serveur web. Celui-ci peut être consulté par des clients autorisés.

Les tests réalisés montrent le bon fonctionnement du système réalisé. Toutefois des améliorations peuvent être apportées. En effet, l'utilisation d'un servomoteur permettant à la caméra de suivre la cible en mouvement. De plus, la réalisation de plusieurs modules de surveillances et la connexion vers le même serveur Web permettra de surveiller plusieurs endroits au même temps (les différents bureaux d'une entreprise). Et prévoir un système d'alarme pour avertir les personnes concernées.

Mots clés : Arduino, Arducam, capteur PIR, Télésurveillance.