REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculté *des Sciences et de la technologie*

Département d'Electronique

Rapport

Projet de Fin de Cycle (PFC)

MCIL 3

FILIERE : **Electronique**

Spécialité : Industries électroniques

Par

- **ZIDOUR Soumia**
- **KEBAILI Ibtissem**

Intitulé

Etude et réalisation d'un système de télésurveillance avec détection de présence, Basé sur une carte ESP32-CAM.

Présenté le : 26/06/2022

Devant le Jury composé de :

Nom & Prénom	*Grade*	*Qualité*	*Etablissement*
Mme. S. MEGUELLATI	*MAA*	*Président*	*Univ-BBA*
M. S.E. AZOUG	*MCB*	*Examinateur*	*Univ-BBA*
M. T. ABED	*MAA*	*Encadreur*	*Univ-BBA*

Année Universitaire 2021/2022

Remerciements

Au terme de ce modeste travail, nous tenons d'abord à remercier ALLAH le Miséricordieux, de nous avoir illuminé et donné force, courage, volonté et santé pour accomplir notre travail.

Nous tenons à exprimer toute notre gratitude à Mr ABED Tarek pour la confiance qu'il nous a faite en acceptant de diriger ce travail. Pour ses recommandations, sa disponibilité, ses encouragements continus, et son assistance ininterrompue et ses conseils judicieux qui nous ont aidés à mener à bien ce travail.

Nous exprimons notre reconnaissance à Mr BAHIH, Chef département d'électronique, ainsi qu'à l'ensemble des enseignants du département et de l''Université de Bordj Bou Arréridj.

A tous ceux qui nous ont aidé, ne serait-ce que par leur soutien, pour accomplir ce modeste travail, qu'ils trouvent ici, l'expression de notre profonde gratitude.

Dédicaces

Je dédie ce travail :

A la mémoire de mon père, puisse ALLAH lui accorder le repos éternel dans son vaste paradis.

A ma très chère maman.

A mes chers frères et sœurs.

A ma collègue et amie, Ibtissem ainsi qu'à toute sa famille.

A toute ma famille.

A tous mes ami(e)s.

Soumia

Dédicaces :

À mes chers parents pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chers frères et sœurs pour leurs encouragements permanents et leur soutien moral,

Aux petits bourgeons Hidaya, Yousra et Nahla

A ma collègue et meilleure amie « Soumia » ainsi à toute sa famille

A tous mes ami(e)s et mes collègues

Pour tous ceux que j'aime

Pour tous ceux qui m'aiment

Merci d'être toujours là pour moi

Ibtissem

Résumé :

Résumé :

Dans ce projet, il s’agit de proposer une solution de sécurité permettant de sécuriser une zone grâce à l’utilisation d’une caméra connectée. Ce projet consiste donc en l’étude et la réalisation d’un système de télésurveillance avec détection de présence, basé sur une carte ESP32-CAM.

Mots-clés : télésurveillance, ESP32-CAM, capteur de mouvement (PIR).

Abstract:

In this project, it is a question of proposing a security solution making it possible to secure an area thanks to the use of a connected camera. This project therefore consists of the study and the realization of a remote monitoring system with presence detection, based on an ESP32-CAM card.

Keywords: remote monitoring, ESP32-CAM, motion detector (PIR).

ملخص:

في هذا المشروع، يتعلق الأمر باقتراح حل أمني يجعل من الممكن تأمين منطقة بفضل استخدام كاميرا متصلة. لذلك يتكون هذا المشروع من دراسة وتحقيق نظام مراقبة عن بعد مع الكشف عن الوجود، بناءً على بطاقة ESP32-CAM

الكلمات الرئيسية: المراقبة عن بعد، مستشعر الحركة(PIR) ,ESP32-CAM

Sommaire :

Liste des figures

Chapitre 01 :

Chapitre 02 :

List des abréviations:

A

API **A**pplication **P**rogramming **I**nterface

B

BMP **B**it**M**a**P**

BLW **B**luetooth **L**ow **E**nergy

G

GPIO **G**eneral **P**urpose **I**nput / **O**utput

H

HTTP **H**yper **T**ext **T**ransfer **P**rotocol

HTTPS **H**yper **T**ext **T**ransfer **P**rotocol **S**ecure

I

IMAP **I**nternet **M**essage **A**ccess **P**rotocol

I2C (IIC) **I**nter **I**ntegrated **C**ircuit

J

JPEG **J**oint **P**hotographic **E**xperts **G**roup

P

PWM **P**ulse **W**idth **M**odulation

S

SoC **S**ystem **o**n **C**hip

SMTP **S**imple **M**ail **T**ransfer **P**rotocol

U

USB **U**niversal **S**erial **B**us

UART **U**niversal **A**synchronous **R**eceiver **T**ransmitter

W

Wi-Fi **W**ireless **F**idelity

WPA **W**i-Fi **P**rotected **A**ccess

WEP **W**ired **E**quivalent **P**rivacy

Introduction générale

Avec le développement d'internet et des télécommunications, plusieurs systèmes et applications ont pu voir le jour. Ces systèmes sont appelés systèmes intelligents, systèmes connectés, systèmes de domotique ou bien Internet des Objets. Ils permettent d'allier des techniques de l'électronique, de l'automatisme, de l'informatique et des télécommunications, afin d'offrir des solutions pour répondre aux besoins de confort, de gestion de l'énergie, de l'automatisation de l'éclairage et du chauffage, et surtout de sécurité.

La sécurité et sa mise en place font partie des objectifs que les individus cherchent à atteindre dans différents environnements, que ce soit à la maison ou même dans les institutions gouvernementales ou dans les entreprises privées. Parmi les méthodes de sécurité, nous trouvons les systèmes de télésurveillance. Ces systèmes permettent l'observation du mouvement et le comportement des personnes dans l'espace ou la zone surveillée. Actuellement, ces systèmes sont devenus une nécessité absolue pour la sécurisation de zones sensibles ou zones à risques. De nombreuses évolutions et améliorations ont été apportées aux systèmes de télésurveillance afin d'offrir des solutions performantes et d'un niveau très élevé en termes de sécurité.

C'est dans cette thématique que s'inscrit ce projet de fin d'études. En effet, il concerne l'étude et la réalisation d'un système de télésurveillance avec détection de présence et basé sur une carte de développement ESP32-CAM. Grâce à l'utilisation en particulier, d'un capteur de présence et d'une caméra, le système est capable de détecter un mouvement dans la zone sécurisée et de prendre instantanément une photo, de la sauvegarder et puis de la transmettre au propriétaire par courrier électronique.

Le mémoire est composé de deux grands chapitres qui comportent plusieurs parties, organisées de la façon suivante :
Le premier chapitre concerne l'étude théorique de tous les éléments nécessaires au projet. Il comporte une partie concernant les informations générales sur la conception de la carte de développement ESP32-CAM. Une autre partie, permet quant à elle, d'aborder les librairies de fonctions liées à ces cartes et qui permettent le développement de systèmes intelligents et connectés.

- Dans le deuxième chapitre, il s'agit de passer en revue toutes les étapes de réalisation, de programmation et de tests du projet. Toutes ces étapes sont décrites de manière claire et détaillée.

Chapitre 01 : Architecture et fonctions des cartes à base de microcontrôleur ESP32

1. Introduction :

Toute conception pour système embarqué nécessite une partie matérielle et une partie logicielle, un système embarqué est un système électronique et informatique piloté par un logiciel, qui est complétement intégré au système qu'il contrôle.

Le système Arduino donne la possibilité d'atteler les performances de la programmation à celles de l'électronique. Le plus grand 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 dans la phase de conception d'une carte électronique.

Dans ce chapitre, nous décrivons la carte ESP32 qui constitue le cœur de notre système de télésurveillance.

2. Le microcontrôleur ESP32 :

Créé et développé par Expressif Système, au dernier trimestre 2016, ESP32 est une série de microcontrôleurs SOC (System On Chip) à faible coût et à basse consommation avec connectivités intégrées Wi-Fi et Bluetooth bi-mode, c'est le successeur du microcontrôleur ESP8266.

2.1 Ressources et caractéristiques de l'ESP32 :

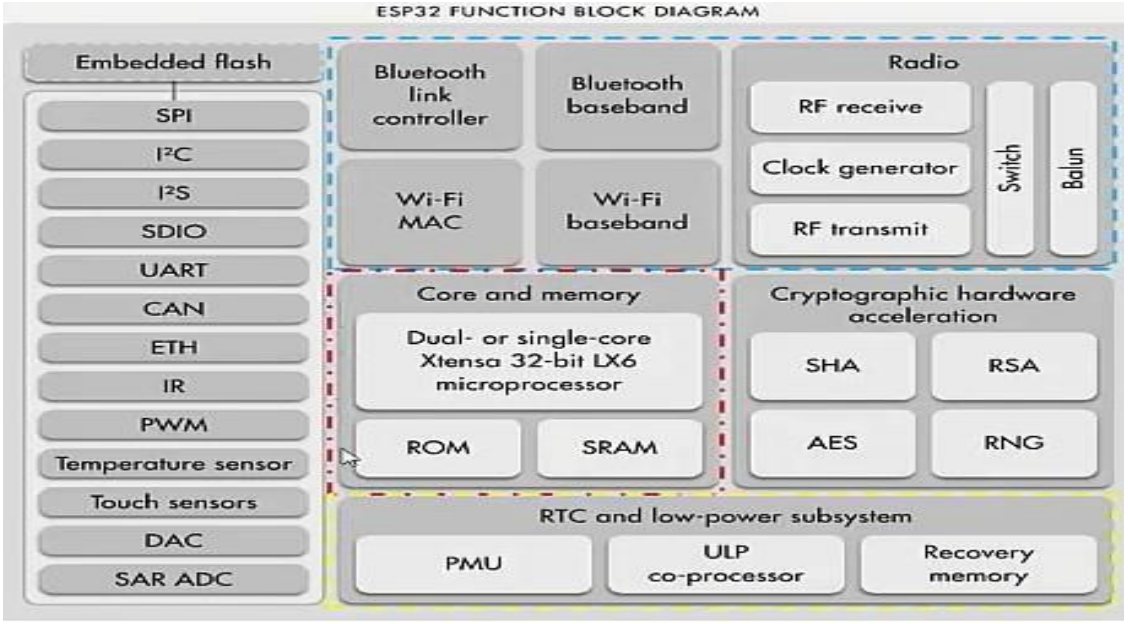

***Figure 1** : Ressources de l'ESP32*

➢ Processeurs :

- Processeur principal : Xtensa simple/double cœur 32 bits, jusqu'à 600 MIPS
- Coprocesseur à très faible consommation fonctionne en veille profonde

- Connectivité sans fil :

Wi-Fi et Bluetooth

- Mémoire :

- Mémoire interne :

- ROM : 448 Ko, pour le démarrage et les fonctions principales
- SRAM : 520 Ko, pour les données et les instructions
- 16 Ko SRAM en RTC : pour l'accès au coprocesseur en mode veille profonde

- Flash externe & SRAM :

ESP32 peut adresser jusqu'à 16 Mo de Flash externe et 8 Mo de SRAM externe.

- Interfaces périphériques :

- 10 × Capacitive touch sensors
- 18 × ADC (analog-to –digital converter)
- 2 × DAC (digital-to-analog converter)
- 2 × I²C (Inter-Integrated Circuit)
- 2 × I²S (Integrated Inter-IC Sound)
- 3 × UART (universal asynchronous receiver/ transmitter)
- 1 × CAN 2.0 (Controller Area Network)
- 4 × SPI (Serial Peripheral Interface)
- 16 × PWM (pulse width modulation)
- 6 × MCPWM (Motor Control Pulse Width Modulator)ssssss

2.2 Brochage du circuit ESP32 :

- La puce ESP32 dispose de 48 broches avec de multiples fonctions
- Toutes les broches ne sont pas disponibles dans toutes les cartes de développement ESP32, et certaines broches ne peuvent pas être utilisées.

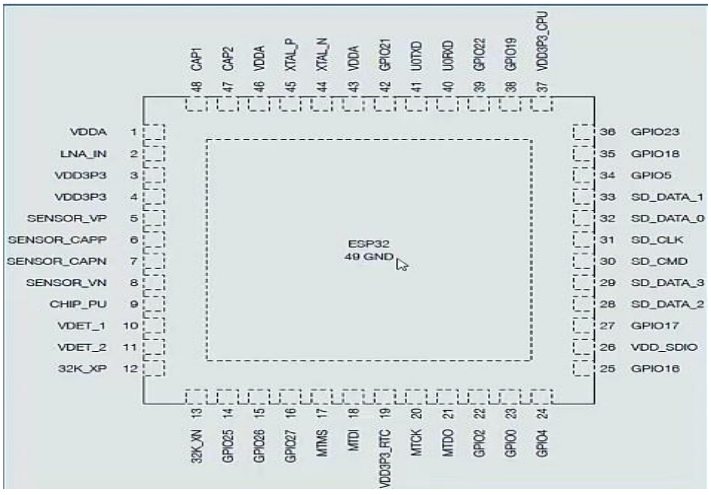

***Figure 2**: Brochage du microcontrôleur ESP32*

3. Présentation de la carte ESP32-CAM :

L'ESP32-CAM est une carte de développement avec une puce ESP32-S, une caméra OV2640, plusieurs GPIO pour connecter des périphériques et un emplacement pour carte micro SD qui peut être utile pour stocker des images prises avec l'appareil photo ou pour stocker des fichiers. L'ESP32-CAM est carte de développement ESP32 disposant de ses propres ressources additionnelles :

- Camera OV2640
- Interface USB - UART
- Bouton de réinitialisation (RST ou EN)
- 10 GPIO accessibles
- PSRAM 4 Mo
- Interface carte Micro-SD.

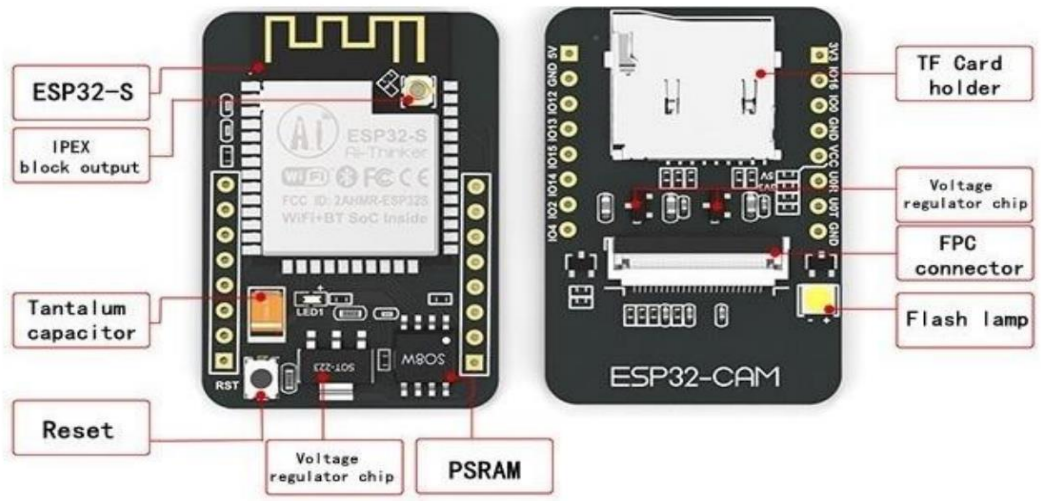

***Figure 3**: Carte ESP32-CAM*

3.1 Caractéristiques ESP32-CAM :

- Flash SPI : par défaut 32 Mbit
- RAM : 520 Ko SRAM + 4M PSRAM
- Bluetooth: Bluetooth 4.2 and Bluetooth Low Energy (BLE)
- Wi-Fi : 802.11 b/g/n
- Prise en charge de l'interface : UART, SPI, I2C, PWM
- Ports GPIO : 10
- Débit en bauds UART : par défaut 115200 bps
- Format de sortie d'image : JPEG (prise en charge OV2640 uniquement), BMP, GRAYSCALE

- Gamme de spectre : 2412 ~ 2484MH
- Antenne : Circuit imprimé intégré et connecteur IPEX
- Sécurité : WPA/WPA2/WPA2-Enterprise/WPS
- Gamme d'alimentation : 5V [2]

3.2 Brochage de la carte ESP32-CAM :

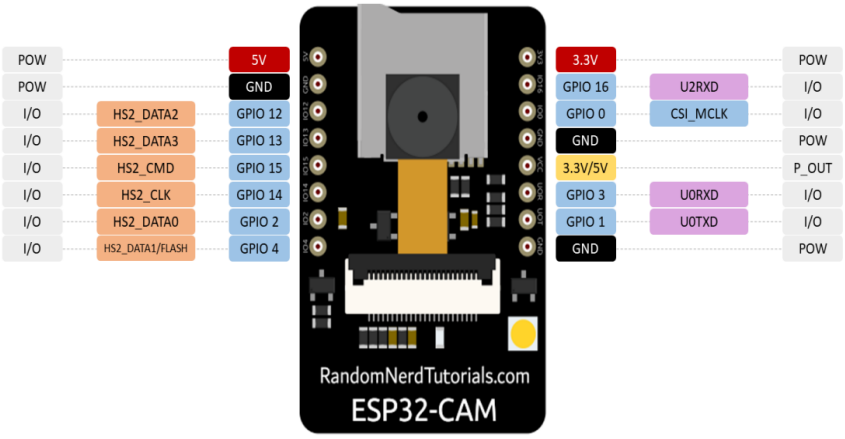

Figure 4 *: Brochure ESP32-CAM*

Il y a trois broches GND et deux broches pour l'alimentation : 3,3 V et 5 V.

GPIO 1 et GPIO 3 sont les broches série (TX et RX, respectivement).

GPIO 0 détermine si l'ESP32 est en mode programmation. Lorsque GPIO 0 est connecté à GND, l'ESP32 passe en mode programmation et il est donc possible de télécharger le code sur la carte. [4]

4. Capteur de mouvement PIR :

4.1 Définition :

Les capteurs PIR permettent de détecter les mouvements et sont souvent utilisés pour détecter si un humain s'est déplacé dans la zone de portée des capteurs. Ils sont petits, peu coûteux, de faible puissance, faciles à utiliser et ne s'usent pas, c'est pourquoi on les trouve couramment dans les appareils et gadgets utilisés dans les maisons ou les entreprises, car ils sont capables de détecter les variations des ondes infrarouges qui génèrent un courant électrique.

Figure 5*: Capteur de mouvement PIR*

4.2 Principe de fonctionnement du capteur de mouvement PIR :

Grâce aux rayons infrarouges, le capteur peut détecter les mouvements dans son champ de surveillance. Il est aussi appelé capteur thermoélectrique ou capteur PIR, il s'agit d'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)

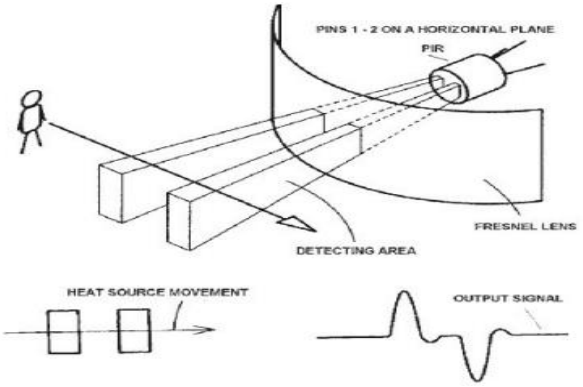

Figure 6:Schéma de fonctionnement du capteur PIR

4.3 Brochage :

Le module PIR comporte 3 broches :

- +5V : Alimentation électrique
- GND : Masse, borne négative de la source d'alimentation.
- OUT : C'est la sortie du capteur. Sa tension est nulle lorsqu'aucun mouvement n'est détecté, et passe à 3,3 V lorsqu'un mouvement est détecté.

4.4 Les réglages :

Le module PIR possède deux potentiomètres de réglage : un pour la sensibilité et un autre pour la durée :

- Potentiomètre de sensibilité : permet de modifier l'ampleur du mouvement qui sera nécessaire pour déclencher la sortie du capteur.
- Potentiomètre de temps : vous permet de contrôler la durée pendant laquelle le signal de sortie du capteur reste actif après la détection d'un mouvement. Si le signal est traité par un microcontrôleur, un signal très court suffit. Mais doit être assez long, dans le cas où le capteur contrôle directement un système d'éclairage.

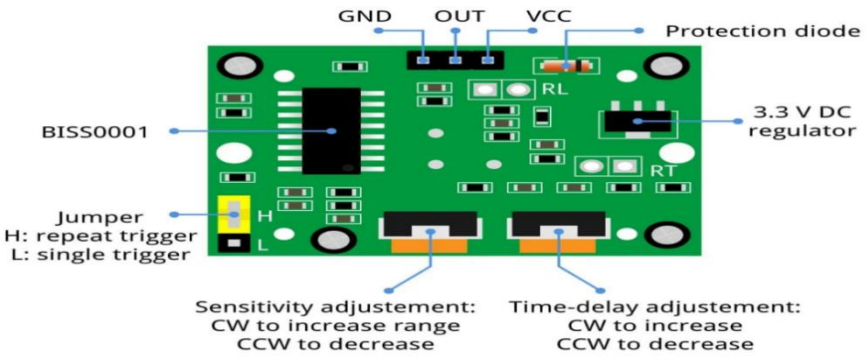

Figure 7: Réglage du HC-SR501

4.5 Cavalier de sélection de déclenchement :

Le cavalier (jumper) peut être utilisé pour sélectionner l'un des deux modes de déclenchement. Il peut être réglé sur L (déclenchement unique) ou H (déclenchement répété) :

- Déclenchement unique : La sortie passera à HAUT dès qu'un mouvement est détecté. Elle restera dans cet état pendant le temps réglé par le potentiomètre. Tout mouvement pendant cette période n'est pas traité et ne redémarre pas le chronomètre.
- Déclenchement répété : Chaque fois qu'un mouvement est détecté, le temporisateur est redémarré.[3]

5. Etude des fonctions de la librairie ESP32 (E/S, Wi-Fi)

5.1 Les fonctions entrée / sortie (GPIO):

L'un des périphériques les plus largement utilisés et les plus polyvalents pour les microcontrôleurs et couramment utilisé pour écrire et lire l'état des broches des circuits d'E/S autour du microcontrôleur. Il existe deux modes différents de configuration GPIO :

- **Mode d'entrée :**

Dans ce mode, le GPIO reçoit l'état numérique d'un appareil spécifique, cet appareil peut être un bouton ou un interrupteur.

- **Mode de sortie :**

Pour un mode de sortie qui modifie l'état numérique du GPIO vers un périphérique spécifique, il sert par exemple à piloter une LED ou tout autre circuit de sortie.

5.2 Les librairies des ports d'E/S :

- **pinMode**

Cette fonction est utilisée pour sélectionner le mode de fonctionnement GPIO pour une broche spécifique.

void pinMode (pin, mode) ; // pin définit le numéro de broche GPIO

Les modes suivants sont pris en charge pour *l'entrée* et la *sortie de* base :

- **INPUT** définit le GPIO comme entrée.
- **OUTPUT** définit le GPIO comme mode de sortie.
- **INPUT_PULLDOWN** définit le GPIO comme entrée avec le menu déroulant interne.
- **INPUT_PULLUP** définit le GPIO comme entrée avec l'extraction interne

- **digitalWrite**

La fonction digitalWrite définit l'état du GPIO sélectionné sur HIGH ou LOW. Cette fonction n'est utilisée que si le pinMode a été configuré comme OUTPUT.

void digitalWrite (pin, val) ; // val définit l'état numérique de la sortie sur HIGH ou LOW

- **digitalRead**

Cette fonction est utilisée pour lire l'état d'une broche spécifique qui est configurée comme INPUT

int digitalRead (pin) ;

- **attachInterrupt**

Cette fonction est utilisée pour attacher l'interruption à la broche spécifiée.

attachInterrupt (pin, voidFuncPtr handler, int mode) ; // mode définit le mode d'interruption

// handler définit la fonction du gestionnaire

Voici les modes d'interruption pris en charge :

- **DISABLED**
- **RISING**
- **FALLING**
- **CHANGE**

- **ONLOW**
- **ONHIGH**
- **ONLOW_WE**
- **ONHIGH_WE**

➢ **attachInterruptArg**

Cette fonction est utilisée pour attacher l'interruption au terminal spécifié en utilisant des arguments.

attachinterruptArg (pin, voidFuncPtrArg handler , arg, mode) ;

// mode définir le mode d'interruption

//arg pointeur vers les arguments d'interruption

➢ **detachInterrupt**

Cette fonction est utilisée pour séparer une interruption d'une broche spécifique.

detachInterrupt (pin) ;

5.3 API Wi-Fi

L'API Wi-Fi prend en charge le pilote de protocole 802.11b/g/n. Cette API comprend :

- Mode station (mode STA ou mode client Wi-Fi). ESP32 se connecte à un point d'accès
- Mode AP (alias mode Soft-AP ou mode Point d'accès). Les appareils se connectent à l'ESP32
- Modes de sécurité (WPA, WPA2, WEP, etc.)
- Recherche de points d'accès [5]

➢ Fonctionnement en point d'accès (AP) **:** Dans ce mode, l'ESP32 est configuré comme un point d'accès (AP) et il est capable de recevoir des connexions entrantes d'autres appareils (stations) en fournissant un réseau Wi-Fi Ce mode peut être utilisé pour servir comme serveur HTTP ou HTTPS.

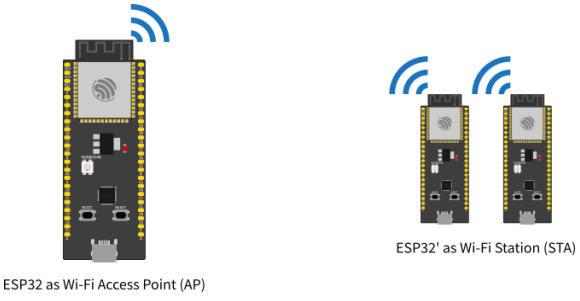

Figure 8 : Fonctionnement en point d'accès

- **Fonctionnement en station (STA) :**

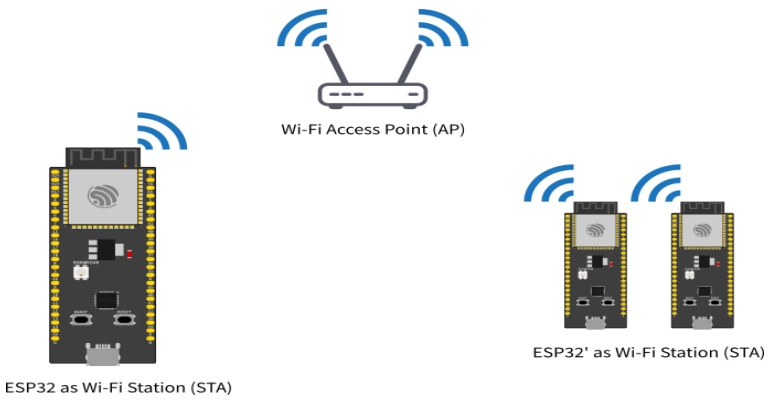

Figure 9 : Fonctionnement en station

Le mode STA est utilisé pour connecter l'ESP32 à un réseau Wi-Fi, fourni par un point d'accès. C'est le mode à utiliser pour connecter la carte ESP32 à Internet.

5.4 Exemples de fonctions Wi-Fi :

- Fonctions Point d'accès :

  - **WiFi.softAP(ssid, password);**

Permet de démarrer le wi-fi en point d'accès.

  - **bool softAP(const char* ssid, const char* passphrase = NULL, int channel = 1, int ssid_hidden = 0, int max_connection = 4, bool ftm_responder = false);**

C'est la fonction qui permet de configurer les caractéristiques du point d'accès, où :

. ssid : spécifie le Wi-Fi SSID.

. passphrase : spécifie le mot de passe Wi-Fi. Si le réseau est ouvert, mettre à NULL.

channel : configure le canal Wi-Fi

ssid_hidden : permet de spécifier si le réseau est caché

max_connection : configure le nombre maximum de connexions simultanées. 4 par défaut.

Renvoie true si la configuration a réussi.

- Fonctions STA :
  - **WiFi.begin(ssid, password);**

ssid et password sont les noms et mot de passe du réseau auquel on désire connecter l'ESP32.

  - **wl_status_t begin(const char* ssid, const char *passphrase = NULL, int32_t channel = 0, const uint8_t* bssid = NULL, bool connect = true);**

Où:

ssid : spécifie le SSID du réseau.

passphrase : configure le mot de passe. NULL un réseau ouvert.

channel : spécifie le canal Wi-Fi.

uint8_t* bssid spécifie le BSSID du point d'accès.

connect : est à true pour se connecter automatiquement au réseau configuré.

  - **bool reconnect();**

Fonction qui permet de reconnecter la carte ESP32 au réseau

  - **bool disconnect (bool wifioff = false, bool eraseap = false);**

Fonction qui permet de déconnecter la carte ESP32 du réseau

5.5 Librairie esp_camera :

C'est une bibliothèque utilisée pour contrôler la caméra OV2640 et elle est déjà incluse dans les bibliothèques ESP32 par défaut, pour l'utiliser elle doit être déclarée dans la parie « include » au début du programme.

```
# include  < esp_camera . h >
```

- **camera_config_t :**

L'instruction proposée par la bibliothèque pour configurer la caméra et est "esp_camera_init" qui prend la structure de type "camera_config_t" en paramètre.

```
camera_config_t  ConfigurationCamera_L ;
```

- **Format d'image :**

```
config . pixel_format = PIXFORMAT_TPEG

// Formts :  YUV422 / GRAYSCALE / RGB565 / JPEG
```

- **Taille du cadre :**

```
config . frame_size = FRAMESIZE_ UXGA

// Tailles de cadre : VGA / CIF / SVGA / XGA / SXGA / UXGA
```

- **Qualité :**

La qualité d'image varie de 0 à 63, un nombre inférieur signifie une qualité supérieure.

```
config . jpeg_quality  = 10 ;

config . fb_count  =  2
```

5.6 Libraires relative à la gestion de la carte SD :

Inclut les bibliothèques FS.h et SD_MMC.h pour la gestion de la carte microSD

```
#include  "FS.h" //  SD Carte ESP32

#include  "SD_MMC.h" //  SD Carte ESP32
```

- **initMicroSDCard ()**

La fonction initMicroSDCard () initialise la carte microSD, renvoie un message d'erreur si l'initialisation échoue.

```
Serial . println ("starting SD Card") ;
if ( !SD_MMC.begin ( )) {
Serial. println (" SD Card Mount Failed ");
return ;
```

- **takeSavePhoto() :**

Pour capture et enregistrer une photo, utilisez la fonction takeSavePhoto() qui accepte le chemin où vous souhaitez enregistrer la photo comme argument.

void takeSavePhoto (String path)

6. Conclusion :

Ce chapitre a permis de présenter le microcontrôleur ESP32 ainsi que la carte de développement ESP 32-CAM. On y décrit également le principe fonctionnement du capteur PIR et ainsi que les fonctions des libraires usuelles relatives au cartes ESP 32 (E/S, wifi, caméra, carte SD). D'après cette étude ; il apparait clairement que les cartes à base du microcontrôleur ESP32 offrent un outil économique parfaitement adapté pour commander des systèmes électroniques intelligents, complexes et connectés.

Chapitre 02 :

Réalisation Pratique

1. Introduction :

Dans ce chapitre, seront présentées toutes les étapes d'étude, de conception et de réalisation du projet. Une présentation du cahier de charge est faite au début, ensuite toutes les étapes suivies seront expliquées afin de présenter au mieux les techniques de réalisation et de programmation pour ce type de projets.

Cette partie décrit et représente chaque étape nous l'avons passé afin d'atteindre notre but de prendre une photo de chaque personne qu'a été détecté par le capteur du mouvement et l'envoie vers un émail spécifié après de le stocker.

2. Cahier des charges

Dans ce projet, il s'agit de concevoir un système de télésurveillance basé sur une carte ESP32-CAM et capable de détecter la présence et le mouvement dans la zone de surveillance. Il s'agira donc, dans le cas d'une présence humaine, de prendre une photo, de la sauvegarder sur un support local, et d'en envoyer une copie par e-mail à une adresse prédéfinie.

3. Schéma de principe

La carte ESP32-CAM est l'élément principal sur lequel doit être construit le projet, il constitue le cœur du système. En plus des capacités de traitement du microcontrôleur, le SOC offre également la connectivité par réseau Wi-Fi. De plus la carte ESP32-CAM intègre également la caméra ainsi qu'un module de stockage sur carte mémoire de type micro-SD. Il suffira uniquement de connecter un capteur de mouvement. Toutes les fonctionnalités du projet seront donc réalisées de manière programmée. Cet aspect de conception est très intéressant dans la mesure où il permet d'aboutir à un système compact dont la conception et la maintenance se fait en grande partie par software. Le schéma de la figure suivante permet de décrire les éléments qui constituent le système à développer.

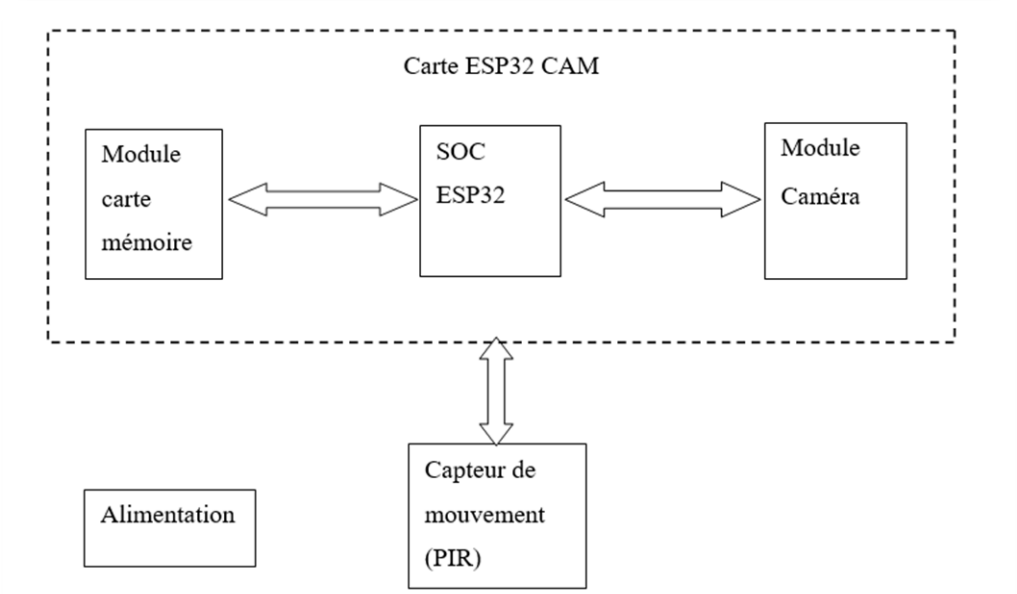

***Figure 10**:Schéma synoptique du système*

4. Schéma du montage électrique :

Le schéma de la figure suivante permet de montrer les connexions électriques préétablies de la caméra et du module carte mémoire. Ces connexions, particulièrement celles de la caméra, changent d'un modèle à un autre. Elles doivent être connues et spécifiées dans la partie d'initialisation du programme de gestion de la caméra. Pour plus de lisibilité, les détails de connexion sont présentés dans les tableaux qui suivent.

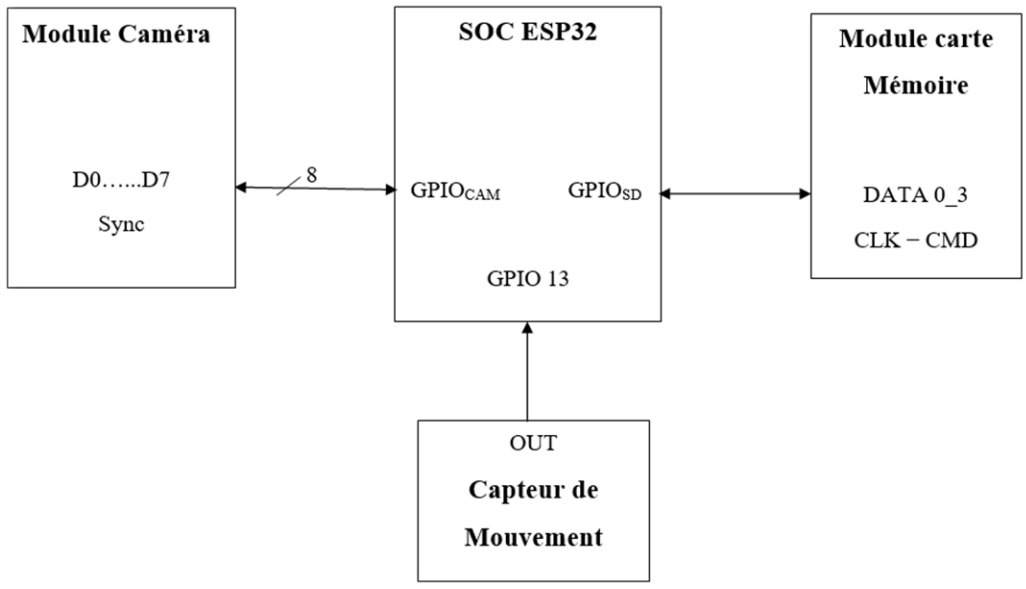

Figure 11:Schéma du montage électrique

4.1 Connexions de la caméra

Les connexions entre la caméra et le microcontrôleur sont celles du modèle AI-Thinker. Ce sont les liaisons $GPIO_{CAM}$ dans la figure 2. Elles sont illustrées dans le tableau suivant :

CAMÉRA OV2640	ESP32	CAMÉRA OV2640	ESP32
D0	GPIO 5	XCLK	GPIO 35
D1	GPIO 18	PCLK	GPIO 0
D2	GPIO 19	VSYNC	GPIO 22
D3	GPIO 21	HREF	GPIO 25
D4	GPIO 36	SDA	GPIO 23
D5	GPIO 39	SCL	GPIO 26
D6	GPIO 34	POWER PIN	GPIO 27
D7	GPIO 32		

4.2 Connexions du lecteur de carte mémoire :

Le brochage du tableau suivant est particulier au module de carte micro-SD dans le cas du modèle l'ESP32-CAM AI -Thinker. Ce sont les connexion $GPIO_{SD}$ dans la figure 2.

Carte SD	ESP32
CLK	GPIO 14
CMD	GPIO 15
DAIA0	GPIO 2
DATA1/Lampe de poche	GPIO 4
DATA2	GPIO 12
DATA3	GPIO 13

4.3 Connexions du capteur de mouvement

Le schéma suivant montre la manière avec laquelle est relié le capteur de mouvement à la carte ESP32-CAM. Pour cela, les connexions suivantes sont établies :

- PIR VCC à ESP32 5V
- PIR GND à ESP32 GDN
- PIR OUT vers ESP32 GPIO 13

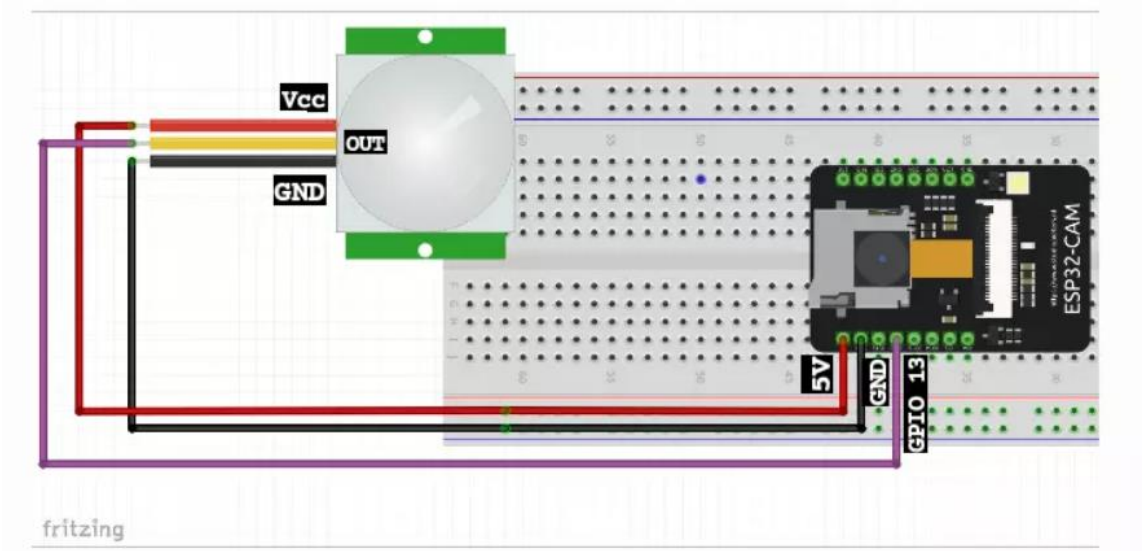

Figure 12 : Câblage du capteur PIR à la carte ESP32-Cam

4.4 Connexions pour la programmation de la carte :

Pour la programmation du module ESP32-CAM, ce dernier est relié à un convertisseur USB-série à travers les broches TxD et RxD en plus de l'alimentation. La broche GPIO0 est reliée à la masse pour permettre de démarrer l'ESP32-CAM en mode programmation (Transfer Mode).

5. Organigramme de fonctionnement :

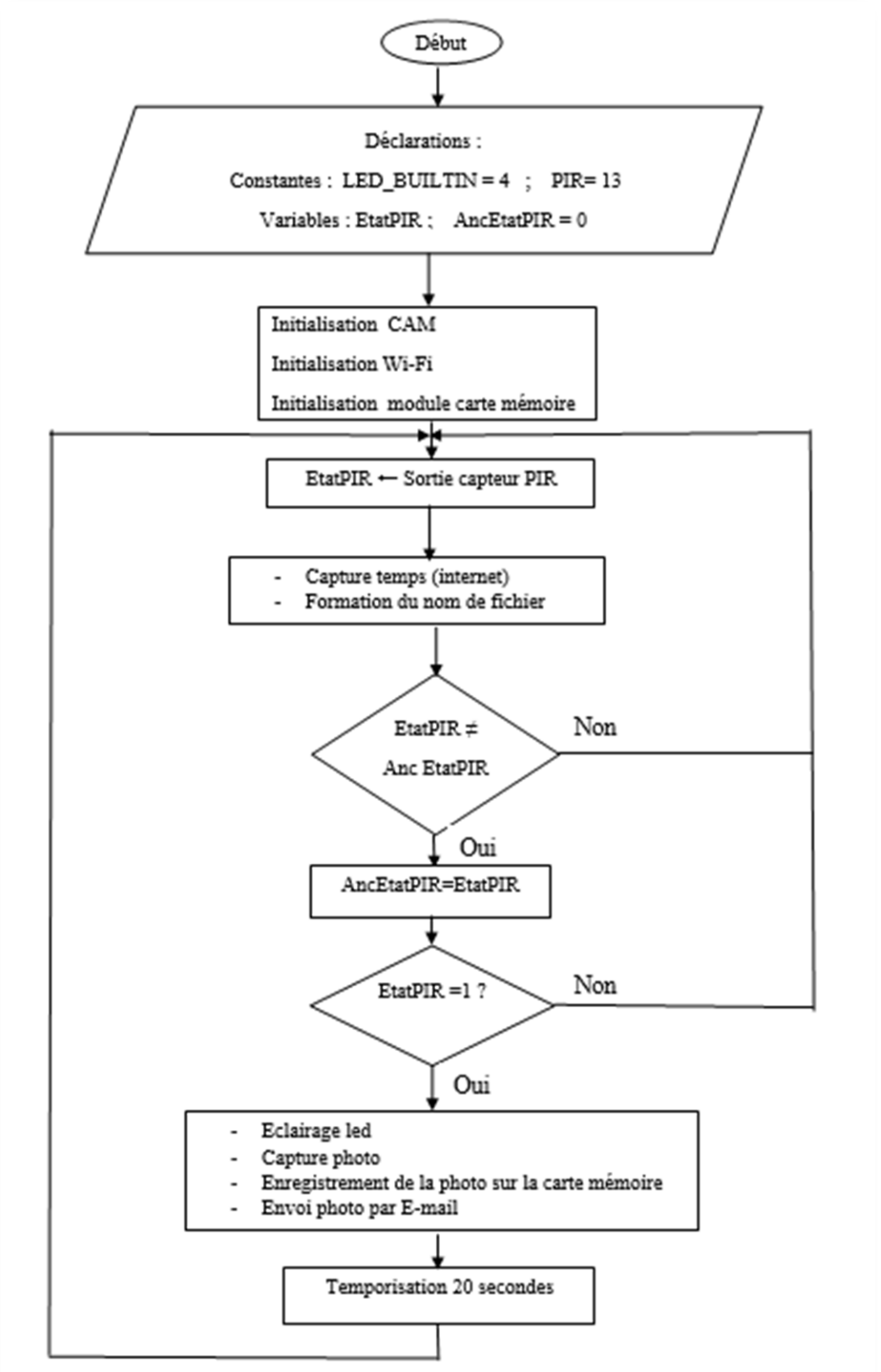

***Figure 13**:Organigramme du système*

6. Configuration de la carte ESP32-Cam dans le logiciel Arduino IDE :

6.1. Le système de développement Arduino IDE :

Le logiciel open source Arduino (IDE) facilite l'écriture de code et son téléchargement sur la carte. Ce logiciel peut être utilisé avec n'importe quelle carte Arduino et également avec les cartes ESP32. [6]

6.2 Installation du module ESP32 pour Arduino IDE :

Ouvrir la fenêtre Préférences dans l'IDE Arduino. Allez dans Fichier » Préférences" :

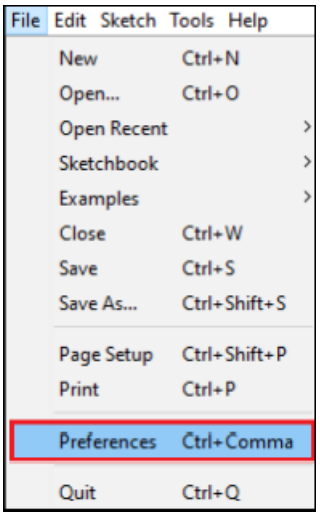

***Figure 14**:Installation du module ESP32 pour Arduino IDE (1)*

Entrer https://dl.espressif.com/dl/package_esp32_index.json dans le "AdditionalBoard Manager URLs " comme illustré dans la figure ci-dessous. Ensuite, cliquer sur le bouton "OK" :

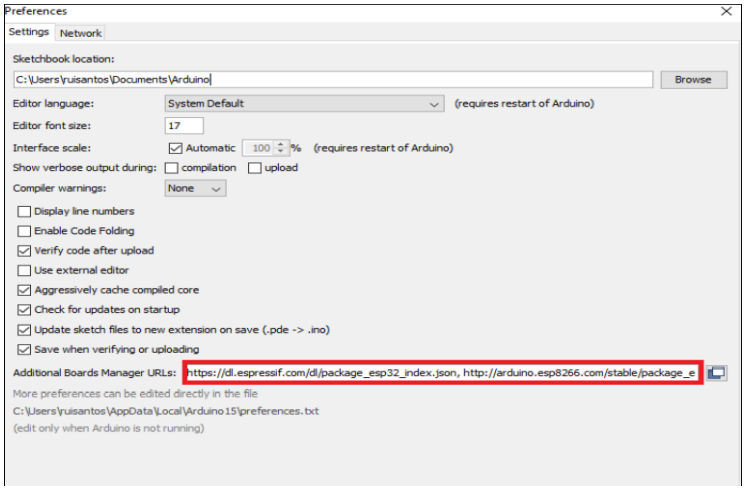

***Figure 15**:Installation du module ESP32 pour Arduino IDE (2)*

Ouvrir le Boards Manager "Atteindre Tools" →" Board" →" Boards Manager" :

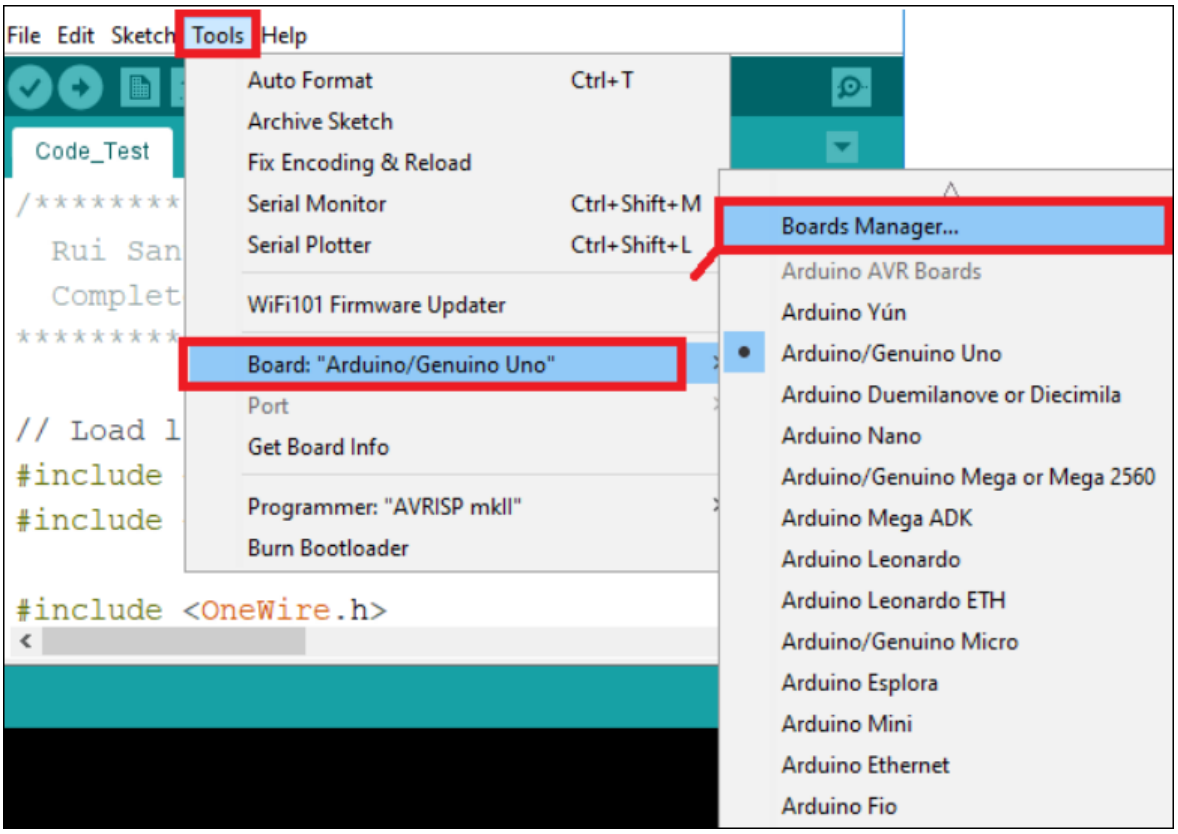

***Figure 16**:Installation du module ESP32 pour Arduino IDE (3)*

Rechercher ESP32 et appuyer sur le bouton d'installation pour le « ESP32 by Espressif Systems » :

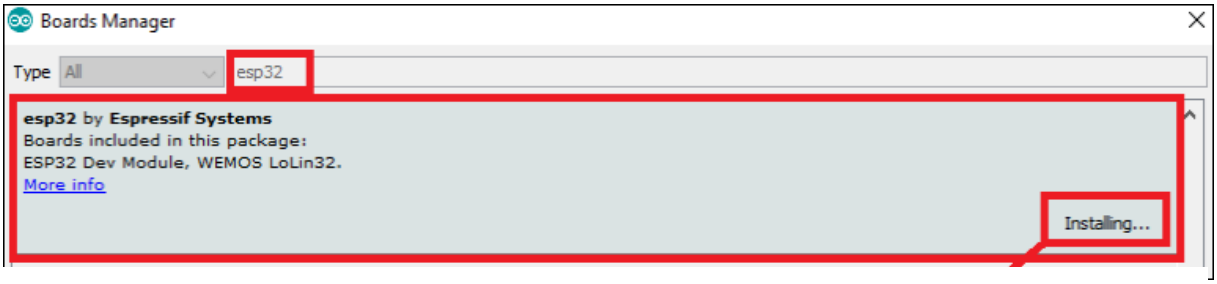

***Figure 17**:Installation du module ESP32 pour Arduino IDE (4)*

Enfin, redémarrer Arduino IDE. Aller ensuite dans " Tools – Board " il y a toutes les cartes ESP32 :

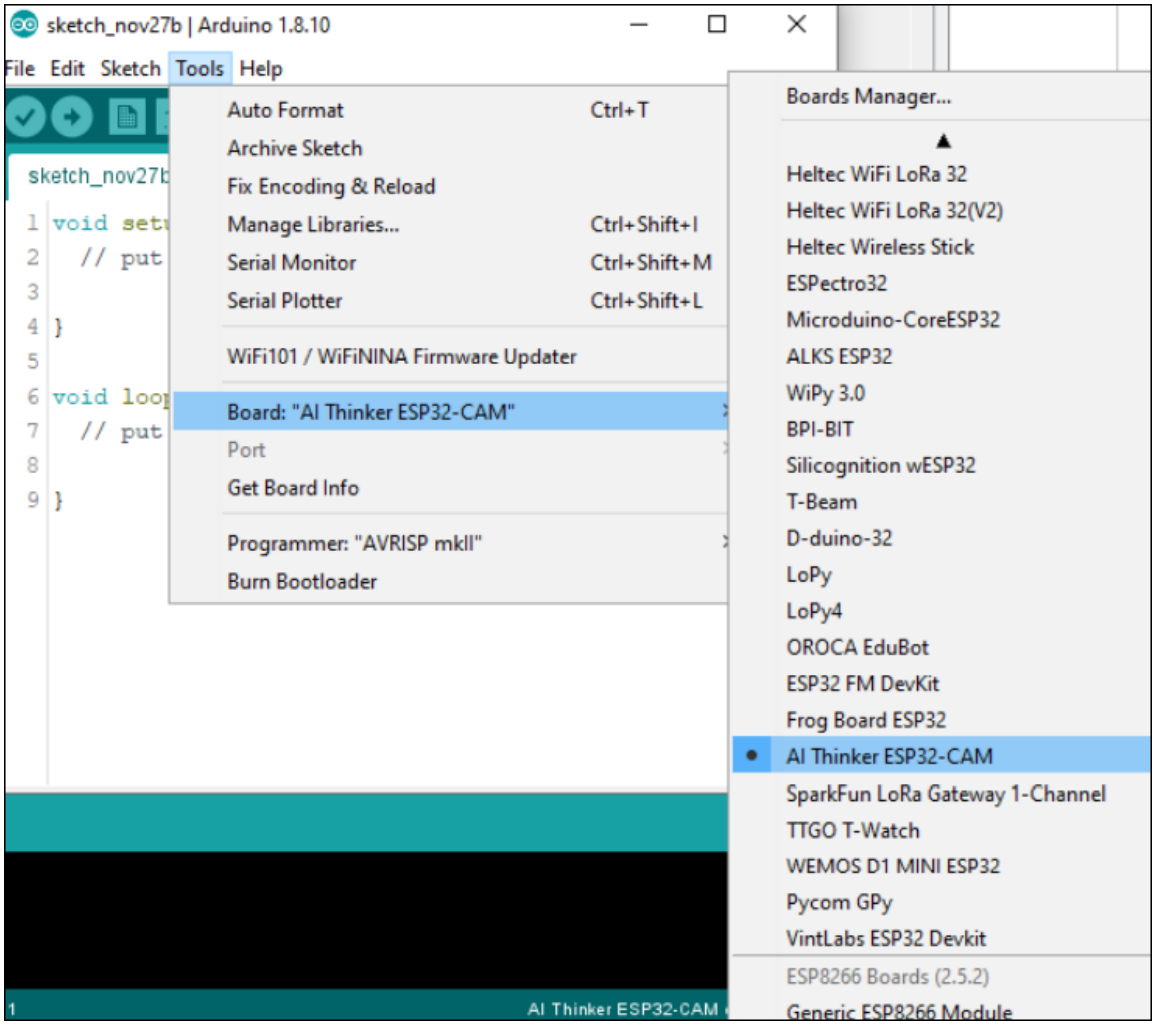

***Figure 18**:Installation du module ESP32 pour Arduino IDE (5)*

6.3.Bibliothèque ESP32 MailClient :

Pour envoyer des e-mails à l'aide d'ESP32, la bibliothèque de messagerie ESP32 est utilisée. Cette bibliothèque permet à ESP32 d'envoyer et de recevoir des e-mails avec ou sans pièces jointes via des serveurs SMTP et IMAP. Dans ce programme, nous utiliserons SMTP pour envoyer un e-mail avec une pièce jointe de l'image capturée à l'aide de l'ESP32-CAM.

6.4.Installation de la bibliothèque du ESP32 MailClient :

Installer le ESP32 Mail Client library. Cette bibliothèque peut être installée via le Arduino IDE Library Manager. Dans Arduino IDE, aller à *Sketch → Include Library → Manage Libraries*…Le gestionnaire de bibliothèque devrait s'ouvrir. Rechercher ESP32 *Mail Client* par « Mobizt » et installer la bibliothèque comme indiqué ci-dessous

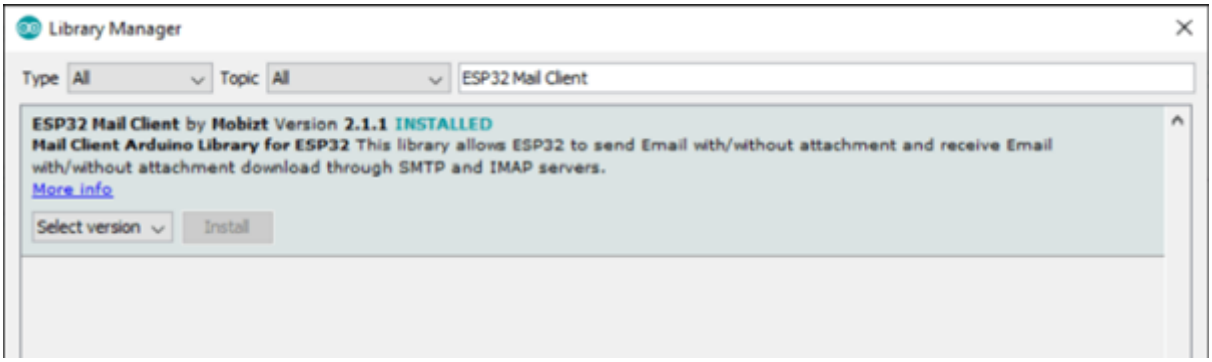

***Figure 19**:Installation de la bibliothèque du ESP32 Mail Client*

- Email de l'expéditeur (nouveau compte) :

Il est très recommandé de créer un nouveau compte de messagerie pour envoyer des emails à votre principale adresse e-mail personnelle. Il est préférable d'utiliser le compte personnel principal de messagerie pour envoyer des e-mails via ESP32. Si des erreurs se produisent ou si une suite d'e-mails par erreur, le serveur de courrier détectera un système automatisé d'envoi de e-mails et bloquera le compte.

- Autoriser les applications moins sécurisées :

Autoriser les applications moins sécurisées à accéder à ce nouveau compte Gmail, afin de pouvoir envoyer des e-mails. Ouvrir ce lien pour accéder au menu : « https://myaccount.google.com/lesssecureapps?pli=1 »

7. Fonctionnement du programme :

- **Connexion au réseau Wi-Fi :**

Dans la section de programme suivante est montrée la manière avec laquelle se fait la connexion au réseau wifi. Dans cette fonction, *ssid* est le nom du réseau et *password* le mot de passe d'accès au réseau.

```
WiFi.begin(ssid, password);
Serial.print("Connecting to WiFi...");
while (WiFi.status() != WL_CONNECTED) {
 delay(500);
 Serial.print(".");
}
Serial.println();
```

- **Programme principal :**

Void loop est la fonction du programme principal qui va être répétée en boucle. Dans cette fonction "*getLocalTime*" permet de capturer le temps à partir d'internet. "*path*" est une variable chaine de caractères qui est utilisée pour former le nom de fichier de sauvegarde de la photo. "*digitalWrite(LED_BUILTIN, HIGH)*" est une function d'E/S qui permet de contrôler la led disponible sur la carte, envoyer un 1 (*HIGH*) va donc allumer cette led.

```
void loop() {
struct tm timeinfo;
char now[20]; //testar
getLocalTime(&timeinfo);
strftime(now,20,"%Y-%m-%d_%H-%M-%S",&timeinfo);
String path = "/picture" + String(now) +".jpg";
 int PIRstate = digitalRead(PIR); // Lire l'état de la broche sélectionnée sur un capteur PIR
  if (PIRstate != lastPIRstate) {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000);
```

"*takeSavePhoto (path)*" est une fonction qui permet de capturer une photo à partir de la caméra et puis de l'enregistrer sur la carte mémoire sous le nom spécifié dans la variable *path*. "*capturePhotoSaveSpiffs()*" est également une fonction qui capture une photo et l'enregistre au format SPIFFS. "*sendPhoto()*" , quant à elle, est une fonction qui va envoyer l'image par e-mail.

```
    digitalWrite(LED_BUILTIN, LOW);
    lastPIRstate = PIRstate;
    Serial.print("Motion Detected :");
    Serial.println(PIRstate);
   }
    If (PIRstate==1){
    takeSavePhoto (path); // prenez et enregistrez la photo dans ce chemin
```

```
   capturePhotoSaveSpiffs();
    sendPhoto();
   Serial.printf("Picture file name: %s\n", path.c_str());
   delay(1000);
  }
  delay(1000);
}
```

8. Tests et résultats pratiques :

Après avoir téléchargé le code sur la carte ESP32-CAM, le système de télésurveillance est mis en service afin de vérifier qu'il fonctionne de manière correcte. D'après l'algorithme établi le microcontrôleur scrute en permanence la sortie du capteur de mouvement. Les procédures d'enregistrement et d'envoi par e-mail ne sont enclenchées que lorsqu'un mouvement est détecté. Un mouvement donc est simulé pour tester le bon fonctionnement du dispositif. Le moniteur série est utilisé pour vérifier, étape par étape, l'évolution de l'exécution d programme. La figure 11 ci-dessous, représente une capture d'écran de l'affichage du moniteur série.

```
COM5
Connecting to WiFi........
SPIFFS mounted successfully
IP Address: http://192.168.43.164
Initializing Date and Time from NTP Server:
Initializing the camera module...Ok!
Initializing the MicroSD card module... Starting SD Card
Saved file to path: /picture2022-05-25_21-10-37.jpg
Taking a photo...
Picture file name: /photo.jpg
The picture has been saved in /photo.jpg - Size: 0 bytes
Taking a photo...
Picture file name: /photo.jpg
The picture has been saved in /photo.jpg - Size: 0 bytes
Taking a photo...
Picture file name: /photo.jpg
The picture has been saved in /photo.jpg - Size: 90752 bytes
Sending email...
Connecting to SMTP server...
SMTP server connected, wait for response...
Identification...
Authentication...
Sign in...
Sending Email header...
Sending Email body...
Sending attachments...
/photo.jpg
Finalize...
Finished
Email sent successfully
Picture file name: /picture2022-05-25_21-10-37.jpg
Motion Detected :0
```

***Figure 20**: Affichage sur le moniteur série*

Nous simulons un mouvement par un déplacement de la main devant le capteur. Une image est donc capturée et enregistrée sur la carte mémoire, le nom de fichier contient la date et l'heure de capture. Ceci est illustré par la figure suivante de la photo prise et enregistrée sur la carte mémoire.

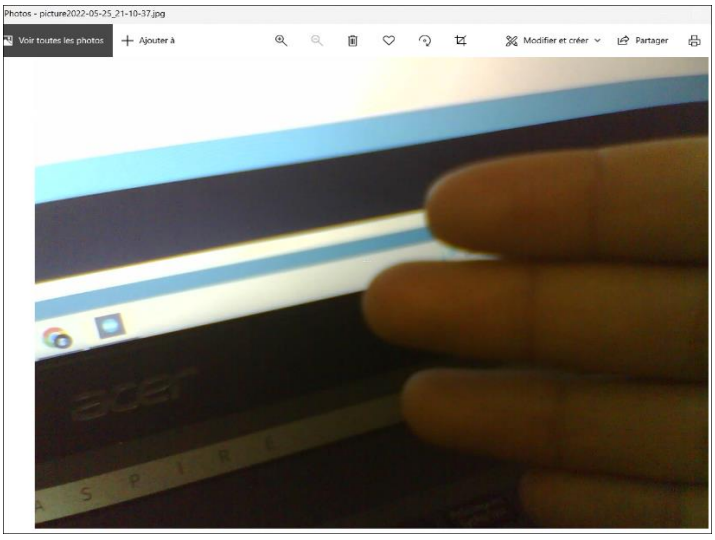

***Figure 21**: Photo prise par ESP32-CAM stockée dans la carte SD*

La figure suivante est celle d'une capture d'écran à partir du logiciel de gestion courrier électronique. On peut constater la réception d'un e-mail avec la photo enregistrée.

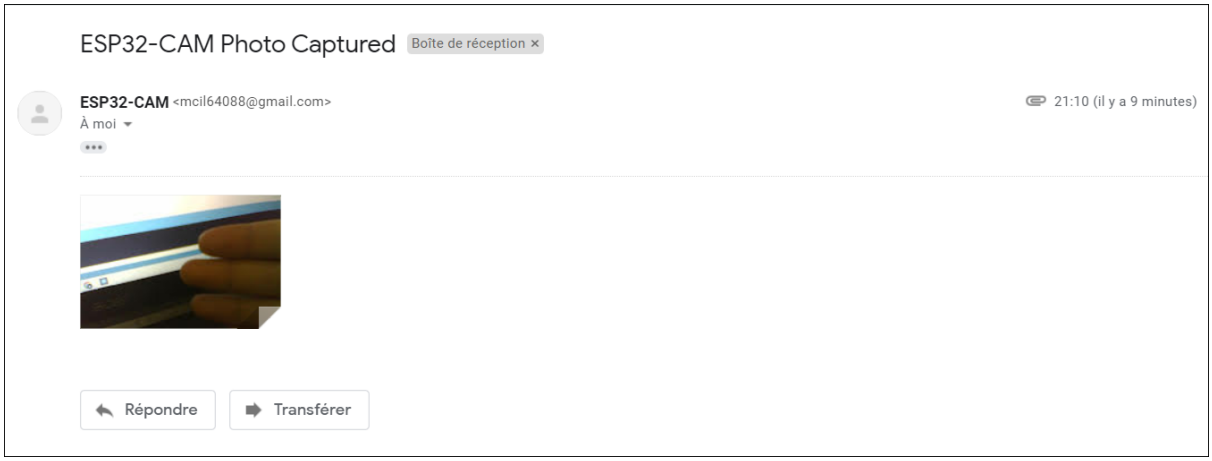

***Figure 22**: Capture d'écran de l'email reçu*

9. Conclusion :

Dans ce chapitre, les schémas de connexions électriques, l'algorithme de fonctionnement ainsi que les programmes ont été décrits afin de monter les étapes de développement du projet. Les figures présentées montrent le bon fonctionnement du système de télésurveillance. Au final, il ne restera que choisir le bon endroit pour l'installation du système afin de pouvoir protéger au mieux la zone à sécuriser.

Conclusion générale

La révolution technologique contemporaine que connaît l'humanité a amené la vie des gens à de nouveaux tournants, rendant difficile d'imaginer l'existence sur Terre en son absence, après qu'elle soit entrée dans tous les domaines de la vie.

Les caméras de surveillance sont l'une des inventions les plus importantes faites par l'homme au XXe siècle, dans sa quête pour améliorer la vie, réduire la criminalité, surveiller et protéger.

Nous avons eu l'opportunité d'aborder ce sujet comme projet de fin d'étude, dans lequel nous nous sommes intéressées à l'étude et la réalisation d'un système de télésurveillance avec détection de présence, basé sur une carte ESP32-CAM. Malgré la complexité et la difficulté de ce projet, nous avons pu atteindre les objectifs escomptés. Le système est tout à fait fonctionnel, et a donner pleine satisfaction lors des essais et des tests de mise en service.

La période de mise en œuvre a été une période d'apprentissage, au cours de laquelle nous avons abordés plusieurs domaines, de la conception électronique, jusqu'au développement de systèmes à base de microcontrôleurs et leur programmation.

La mise au point de ce projet nous a permis développer un système basique qui peut servir comme base à la conception d'un système de sécurité un peu plus élaboré, par l'adjonction d'autres types de capteurs pour aboutir à une protection plus précise et plus performante.

Au final, nous souhaitons que ce travail servira comme base au étudiants qui désirent concevoir des systèmes électroniques à base de cartes de développement ESP32 qui sont devenus de plus en plus utilisés pour la conception de systèmes personnels et même commerciaux.

Références Bibliographiques

[1] ESP32 Series Datasheet Version 3.0.

[2] Rui SANTOS and Sara SANTOS, ESP32-CAM Project Version 1.0.

[3] ESP32-CAM AI-Thinker Pinout Guide: GPIOs Usage Explained.

[4] lady ada, PIR Motion Sensor.

[5] www.espressif.com.

[6] Rui SANTOS and Sara SANTOS, ESP32 WEB SERVER WITH ARDUINO.IDE.