

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D' INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Électronique

Spécialité : **Électronique Industrielle**

Présenté par

Ghiles HESSAS

Amine IDDIR

Thème

Conception et réalisation d'un système de télésurveillance

Mémoire soutenu publiquement le 30/09/ 2024 devant le jury composé de :

Mme. SALEM.D

Présidente

Mme. BOUARABA.F

Examinatrice

Mr. IDJERI

Promoteur

Mr. ATTAF.Y

Co-promoteur

Année universitaire : 2023/2024



Remerciements

Nous tenons à présenter nos vifs remerciements à tous nos enseignants et toutes les personnes qui par leurs paroles, leurs écrits, leur soutien et leurs critiques constructives ont contribué à alimenter nos réflexions

Nous tenons aussi à exprimer toute notre profonde reconnaissance envers, Mr. IDJRI, pour sa patience, sa disponibilité et surtout pour ses judicieux conseils et prodigues orientations tout au long de la réalisation de ce mémoire.

Notre reconnaissance la plus sacré va aussi vers Mr. ATTAF, d'avoir accepté de continuer à nous encadrer en l'absence de Mr. IDJRI, et qui contribuer fortement à la finalisation de ce travail.

Nous remercions aussi les membres de jurys qui feront l'honneur de bien vouloir examiner ce travail.



Dédicace

Je dédie ce mémoire à :

A mes parents, pour leur amour, leur générosité, leur
patience, et pour avoir été de bon conseil pendant
toutes ses années

A toute mes Frères et mes sœurs, à tous amis

A toutes les personnes qui, de près ou de loin, ont
contribué à l'élaboration de ce travail

A tous les enseignants de l'Ummto, en particulier,
ceux du département l'électronique

Amine.





Dédicace

Je dédie ce mémoire à :

A mes parents, pour leur amour, leur générosité, leur
patience, et pour avoir été de bon conseil pendant
toutes ses années

A toute mes Frères et mes sœurs, à tous amis

A toutes les personnes qui, de près ou de loin, ont
contribué à l'élaboration de ce travail

A tous les enseignants de l'Ummto, en particulier,
ceux du département l'électronique

Ghilas.



Liste des tableaux

Tableau II.1 : Caractéristiques de capteur PIR	28
Tableau II.2 : Différentes pins de la caméra OV2640.....	36

Liste des Figures

Figure I.1. esp32	4
Figure I.2. La carte ESP32-WROVER.....	5
Figure I.3. ESP32-WROVER-B.....	5
Figure I.4. ESP32-S2	6
Figure I.5. ESP32-C3	6
Figure I.6. ESP32-S3	7
Figure I.7. ESP32-PICO-KIT.....	7
Figure I.8. Architecture de la carte esp32.....	8
Figure I.9. Microcontrôleur esp32	9
Figure I.10. Les broches Tx et Rx de l'ESP32-CAM	10
Figure I.11. Alimentation de la carte esp32	11
Figure I.12. Les différentes parties de la fenêtre principale du logiciel Arduino IDE	17
Figure I.13. Le moniteur série.....	18
Figure I.14. La barre d'outils.....	18
Figure II.1.la chaîne d'acquisition d'un capteur	23
Figure II.2 Chaîne de mesure.....	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	27
Figure II.6. Capteur de mouvement	28
Figure II.7. Réglages du PIR.....	29
Figure II.4. Module GSM Sim900.....	29
Figure II.8. Caméra CCD	31
Figure II.9. Caméra analogique	31
Figure II.10. Caméra infrarouge	32
Figure II.11. Caméra CMOS.....	33
Figure II.12. Caméra discrète	34

Figure II.13. Caméra numérique	34
Figure II.14. Caméra OV2640.....	35
Figure III.1. Schéma synoptique du system.....	38
Figure III.2. Branchement de la SIM900 avec ESP32-CAM	39
Figure III.3. Branchement de capteur PIR.....	40
Figure III.4. Câblage de système.....	41
Figure III.5. Montage de notre système	42
Figure III.6. L'organigramme du fonctionnement général.....	43
Figure III.7. L'écriture du programme Arduino	44
Figure III.8. Sélection de la cible	45
Figure III.9. Sélection du port.....	46
Figure III.10. Transfert du programme.....	47
Figure III.11. Choix de librairie.....	48
Figure III.12. Présentation du terminal série	48
Figure III.13. Ouverture de terminal série	49
Figure III.14. Ajustage du débit de la communication série	49
Figure III.15. Fenêtre de logiciel lors de détection de mouvement.....	50
Figure III.16. L'SMS reçu par le module GSM SIM900	50
Figure III.17. L'affichage de l'adresse IP sur le moniteur série	52

TABLE DES MATIERES

Introduction générale.....	1
----------------------------	---

Chapitre I : Généralités sur la carte ESP32

I.1.Introduction.....	3
I.2.Historique	3
I.3. Définition de la carte esp32.....	3
I.4. Les différentes versions de L'esp32.....	4
I.4.1. ESP32	4
I.4.2. ESP32-WROVER	5
I.4.3. ESP32-WROVER-B	5
I.4.4. ESP32-S2.....	6
I.4.5. ESP32-C3	6
I.4.6. ESP32-S3	7
I.4.7. ESP32-PICO-KIT.....	7
I.5. Architecture de la carte esp32	8
I.6. Présentation matérielle de la plateforme arduino	8
I.6.1. Le microcontrôleur	8
I.6.1.1. Caractéristique du microcontrôleur esp32.....	9
I.6.1.2. Démarrage du microcontrôleur.....	10
I.6.1.3. À l'intérieur du microcontrôleur ESP32-CAM	10
I.6.2. Alimentation de la carte esp 32.....	11
I.6.3. Les broches d'alimentation de carte esp 32.....	12
I.6.4. La Mémoire	13
I.6.5. Entrées et sorties numériques esp 32	14
I.6.6. Broches analogiques esp 32.....	15
I.6.7. Autres broches esp 32.....	15
I.6.8. Communications de la carte esp 32.....	15
I.6.9. Protection d'un port USB contre la surcharge en intensité ESP32	16
I.7. Présentation logicielle de Arduino IDE	16
I.7.1. Description de la barre d'outils	18

Chapitre II : Les éléments systèmes de télé surveillance

II.1 introduction	20
II.2. Définition de la télésurveillance	20
II.2.1. Domaine d'applications.	20
II.2.2. Avantages et inconvénients de la télésurveillance	21
II.2.3. Les éléments de télésurveillances	21
II.2.3.1. Capteur de position et d'état	21
II.2.3.2. Capteur d'images.	21
II.2.3.3. Un transmetteur	22
II.2.3.4. L'enregistreur	22
II.3. Les capteurs	22
II.3.1. Définition d'un capteur.....	22
II.3.2. Chaîne de mesure	23
II.3.3. Constitution d'un capteur.....	24
II.3.4. Rôle du capteur	24
II.3.5. Classification des capteurs	25
II.3.6. Les caractéristiques d'un capteur.....	26
II.3.7. Capteur infrarouge Passif (PIR).....	26
II.3.8. Principe de fonctionnement.....	27
II.3.9. Caractéristiques électriques et fonctionnelles de capteur PIR	28
II.4. Module gsm sim900.....	29
II.5. Les caméras.....	30
II.5.1. Les critères de choix d'une caméra	30
II.5.2. Différents types de caméras.....	30
II.5.3. La caméra OV2640	35
II.5.4. Identification des pins	36

Chapitre III : La réalisation pratique

III .1. Introduction	38
III.2. Réalisation matérielle	38
III .2.1. Schéma synoptique	38
III .2.2. Différents éléments utilisés dans le projet	39
III .2.2.1. Branchement du module SIM900 à la carte ESP32-CAM	39
III .2.2.2. Branchement du capteur PIR à la carte ESP32-CAM.....	40

III.2.3. Câblage du système	41
III.2.4. L'organigramme	42
III.2.5. Les étapes de programmation de la carte ESP32-CAM	44
III.2.6. Librairies	47
III.2.7. Moniteur série	48
III.3. Tests de l'application.....	49
III.4. Serveur Web.....	51
III.5. Discussion	53
Conclusion générale	54

Résumé

Introduction générale

Introduction générale

Avec l'évolution rapide d'Internet et des technologies de télécommunication, de nombreuses innovations ont vu le jour, notamment dans le domaine des systèmes intelligents et connectés. Parmi ces technologies émergentes, on trouve la domotique et l'Internet des Objets (IoT), qui combinent électronique, informatique, automatisation et télécommunications pour offrir des solutions adaptées à des besoins variés. Ces solutions visent à améliorer le confort, à optimiser la gestion énergétique, à automatiser certaines tâches (comme l'éclairage ou le chauffage) et, plus particulièrement, à renforcer la sécurité des espaces privés ou publics.

La sécurité reste en effet une priorité majeure, que ce soit pour la protection des habitations, des entreprises ou des institutions publiques. Les systèmes de télésurveillance jouent un rôle crucial dans ce domaine, permettant de surveiller en temps réel les mouvements et comportements dans une zone donnée. Ces dispositifs sont désormais indispensables pour assurer la sécurité des espaces sensibles ou à risque, et ont bénéficié de nombreuses évolutions technologiques qui les rendent de plus en plus performants et fiables.

C'est dans ce cadre que s'inscrit notre projet de fin d'études, qui porte sur la conception et la mise en place d'un système de télésurveillance intelligent avec détection de présence. Ce système repose sur une carte de développement ESP32-CAM et un module GSM SIM900. Grâce à l'intégration d'un capteur de mouvement et d'une caméra, il permet de détecter un mouvement dans la zone surveillée, de capturer une image en temps réel, de l'enregistrer et, en cas de détection, d'envoyer immédiatement une alerte par SMS au propriétaire via le module GSM SIM900. Ce dispositif vise à offrir une solution de sécurité efficace, réactive et accessible pour la protection des biens et des personnes.

CHAPITRE I

Généralités sur la carte ESP32

I.1.Introduction

Les systèmes ESP32 permettent de fusionner des capacités de programmation avancées avec des produits électroniques. Un des grands avantages des dispositifs programmables comme l'ESP32 est qu'ils simplifient largement la conception des circuits, ce qui contribue à réduire les coûts globaux. De plus, la charge de travail associée à la production est étroitement liée à celle de la conception des cartes électroniques, ce qui optimise le processus de développement et permet de gagner du temps et des ressources. En somme, l'ESP32 offre une solution efficace et économique pour le prototypage et la réalisation de projets électroniques connectés.

I.2.Historique

Le projet ESP32 a émergé d'une initiative collective d'enseignants et d'élèves d'une école spécialisée en design interactif. À l'époque, entre 2003 et 2004, ils faisaient face à un défi majeur : les outils nécessaires pour créer des projets électroniques connectés étaient à la fois complexes et coûteux, oscillant entre 80 et 100 euros. La plupart des solutions de prototypage se concentraient sur les domaines de l'ingénierie, de la robotique et de la technologie, ce qui restreignait leur accessibilité.

En 2003, Hernando Barragán a développé dans le cadre de sa thèse une carte électronique appelée "Câblage", qui offrait une programmation gratuite et ouverte. Cette innovation a ouvert la voie à des projets comme l'ESP32, qui a été conçu par une équipe de chercheurs et de créateurs désireux de rendre la technologie plus accessible. Leur objectif était de développer des appareils moins chers et plus faciles à utiliser, facilitant ainsi le prototypage et l'expérimentation pour tous les passionnés d'électronique.

I.3. Définition de la carte esp32

L'ESP32 est une carte de développement basée sur un microcontrôleur de type système sur une puce (SoC) System on Chip , conçu par Expressif System. Elle est principalement utilisée dans des applications d'Internet des objets (IoT) grâce à sa connectivité intégrée Wi-Fi et Bluetooth. Le microcontrôleur est basé sur l'architecture Xtensa LX6, offrant de bonnes performances grâce à son processeur à double cœur. L'ESP32 est également équipé d'un processeur de signal numérique (DSP), ce qui lui permet de gérer des tâches complexes, comme le traitement audio et le contrôle de capteurs. En termes de programmation, l'ESP32 peut être développé via plusieurs environnements, notamment l'ESP-IDF, Arduino IDE et Micro Python, rendant la plateforme accessible à différents niveaux de compétence.

Sa conception compacte et sa capacité à interagir avec divers périphériques grâce à ses nombreuses broches d'entrée/sortie (GPIO) General Purpose Input/Output , en font un choix prisé pour les développeurs et les makers. En raison de son coût abordable et de ses fonctionnalités avancées, l'ESP32 est devenu un outil incontournable pour créer des dispositifs connectés et des applications intelligentes. Pour programmer cette carte, on utilise le logiciel IDE.

I.4. Les différentes versions de L'esp32

I.4.1. ESP32 :

La version originale de l'ESP32 intègre un processeur à double cœur basé sur l'architecture Xtensa LX6. Elle propose une connectivité Wi-Fi et Bluetooth, ce qui en fait une plateforme idéale pour la plupart des applications d'Internet des objets (IoT). Grâce à ses performances élevées, l'ESP32 peut gérer des tâches complexes et traiter des données en temps réel.

Le modèle esp32 (Figure I.1)

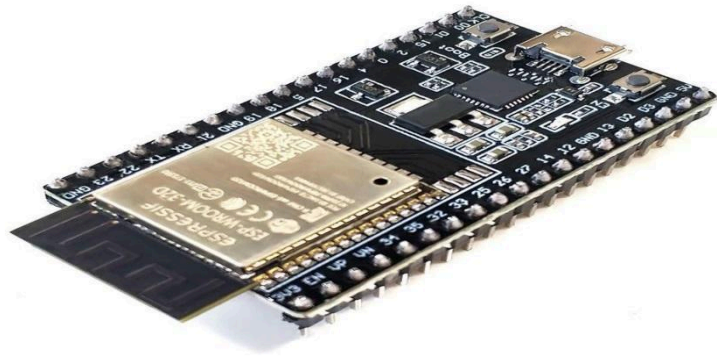


Figure I.1. esp32

I.4.2. ESP32-WROVER

Le module ESP32-WROVER est conçu pour des applications nécessitant plus de mémoire. En plus des fonctionnalités de base, il dispose d'une mémoire flash externe et d'une RAM supplémentaire. Cela le rend particulièrement adapté aux projets nécessitant un traitement intensif ou le stockage de données importantes, comme les systèmes de surveillance ou les applications multimédias.

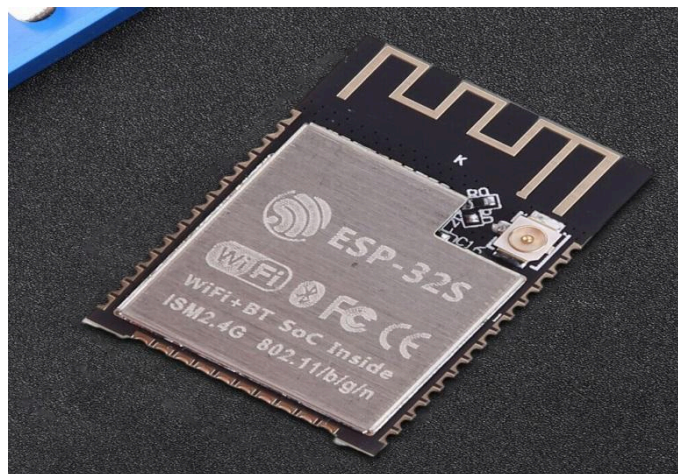


Figure I.2. La carte ESP32-WROVER

I.4.3. ESP32-WROVER-B

Amélioration du WROVER, l'ESP32-WROVER-B offre une meilleure gestion thermique et des

performances accrues. Ce module est idéal pour des applications critiques en termes de performance, comme le traitement d'images ou le contrôle de robots. Grâce à ses caractéristiques avancées, il convient parfaitement aux développeurs cherchant à créer des solutions robustes et efficaces.

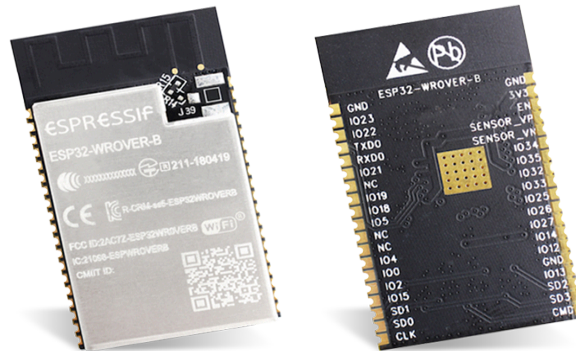


Figure I.3. ESP32-WROVER-B

1.4.4. ESP32-S2

L'ESP32-S2 est une version axée sur la sécurité, ne disposant pas de Bluetooth. Elle intègre des fonctionnalités de cryptage avancées et est optimisée pour les applications à faible consommation d'énergie. Ce modèle est particulièrement adapté pour des projets où la sécurité des données est primordiale, comme les systèmes de paiement ou les dispositifs de santé connectés.

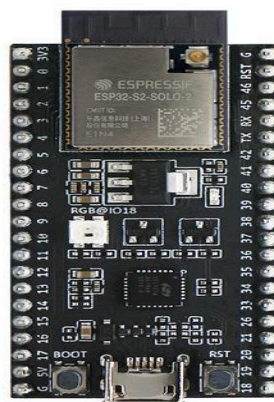


Figure I.4. ESP32-S2

1.4.5. ESP32-C3

L'ESP32-C3 se distingue par son architecture RISC-V, ce qui lui confère une grande efficacité énergétique tout en maintenant des performances élevées. Ce module est parfait pour les applications IoT à faible consommation, telles que les capteurs environnementaux ou les dispositifs portables. Sa conception permet également une intégration facile dans divers projets.

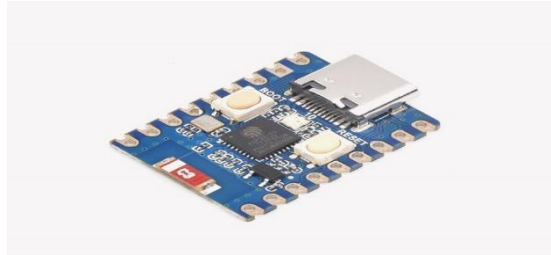


Figure 1.5. ESP32-C3

1.4.6. ESP32-S3

Le modèle ESP32-S3 apporte des capacités d'intelligence artificielle avec des accélérateurs dédiés. Cela permet l'exécution de modèles de machine learning directement sur le microcontrôleur. Idéal pour des applications comme la reconnaissance vocale ou visuelle, il s'adresse aux développeurs souhaitant intégrer des fonctionnalités avancées d'IA dans leurs projets.

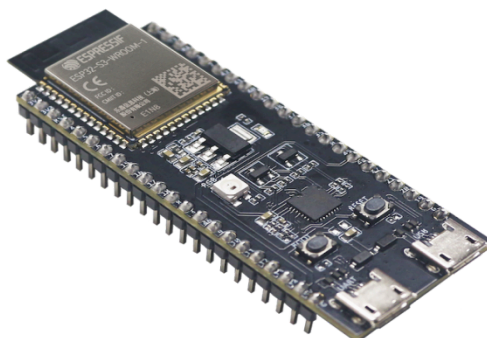


Figure 1.6. ESP32-S3

1.4.7. ESP32-PICO-KIT

L'ESP32-PICO-D4 est un module compact qui intègre toutes les fonctionnalités de l'ESP32 dans un format réduit. Sa taille le rend parfait pour les projets avec des contraintes d'espace, comme les appareils portables ou les capteurs miniaturisés. Ce module est idéal pour les développeurs cherchant à créer des solutions élégantes et peu encombrantes.

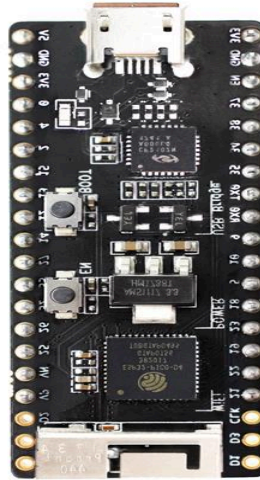


Figure I.7. ESP32-PICO-KIT

1.5. Architecture de la carte esp32

La carte esp32 est composée de plusieurs éléments essentiels qui sont présentés dans la figure (I.5)

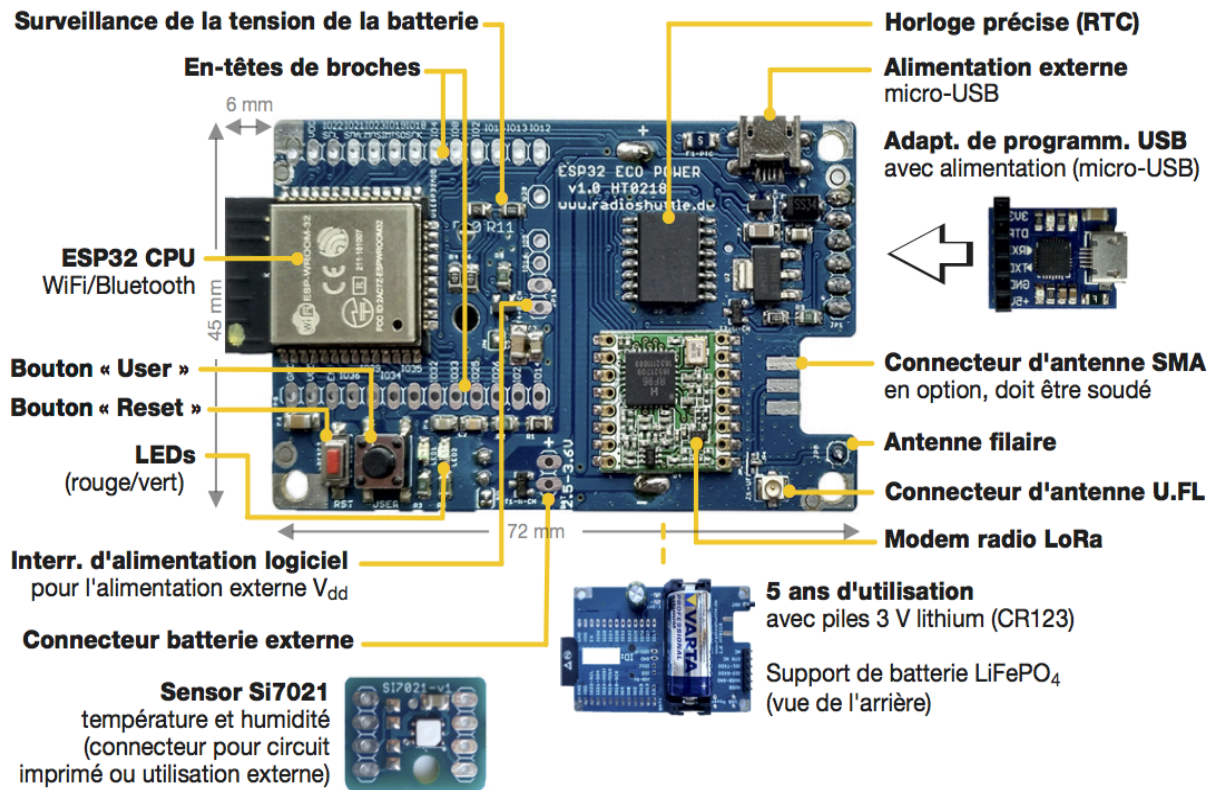


Figure I.8. Architecture de la carte esp32

1.6 .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.

1.6.1. Le microcontrôleur

Le microcontrôleur est un circuit intégré qui regroupe les composants essentiels d'un ordinateur, à savoir : le processeur, les mémoires, les unités périphériques, ainsi que les interfaces d'entrées-sorties.

Le microcontrôleur est un circuit intégré qui regroupe les composants essentiels d'un ordinateur, à savoir : le processeur, les mémoires, les unités périphériques, ainsi que les interfaces d'entrées-sorties.

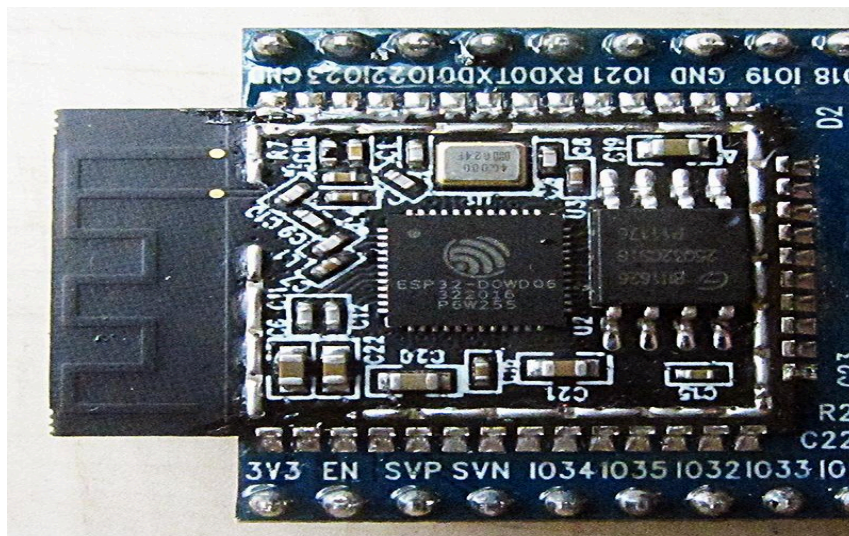


Figure I.9. Microcontrôleur esp32

1.6.1.1. Caractéristique du microcontrôleur esp32

Le microcontrôleur de la carte ESP32-CAM est un ESP32 de la famille Xtensa, un microcontrôleur 32 bits développé par **Espressif Systems**. Ses principales caractéristiques sont :

- **Mémoire Flash** : Jusqu'à **4 Mo**, utilisée pour stocker le programme.
- **SRAM** : Environ **520 Ko** pour le stockage des données volatiles.
- **EEPROM** : Simulée dans la mémoire flash, utilisée pour stocker des données non volatiles.
- **Digital I/O** : Jusqu'à **34 broches GPIO** (entrées/sorties numériques), dont certaines peuvent être utilisées pour la communication série, la gestion des capteurs ou d'autres périphériques.
- **WiFi** : Intégré pour la connectivité réseau sans fil (802.11 b/g/n).
- **Bluetooth** : Prend en charge à la fois **Bluetooth classique** et **Bluetooth Low Energy (BLE)**.
- **PWM** : Génération de signaux **PWM** (modulation de largeur d'impulsion) sur plusieurs broches GPIO, utile pour le contrôle des moteurs ou l'éclairage à intensité variable.
- **ADC** : 18 canaux ADC (analogique-numérique) avec une résolution de 12 bits.
- **DAC** : 2 canaux DAC (numérique-analogique) intégrés.
- **Timers/Counters** : Plusieurs timers matériels 16 bits disponibles, permettant une gestion précise du temps et la génération de signaux PWM.

1.6.1.2. 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 défilet, instruction par instruction. Chaque fois qu'une nouvelle variable sera nécessaire, elle sera mise en RAM pour qu'on ait une mémoire de cette dernière (et supprimer lorsqu'elle n'est plus nécessaire)

1.6.1.3. À l'intérieur du microcontrôleur ESP32-CAM

L'emplacement du programme :

Le microcontrôleur **ESP32-CAM** reçoit le programme sous forme de signal électrique via les broches **GPIO 1 (TX)** et **GPIO 3 (RX)**, ou en mode **UART0** par les broches **GPIO 15 (TX)** et **GPIO 16 (RX)**. Ces broches, disponibles sur la carte, permettent le transfert du programme depuis un ordinateur ou un autre périphérique. La figure (I.7) illustre les broches correspondantes sur la carte ESP32-CAM.

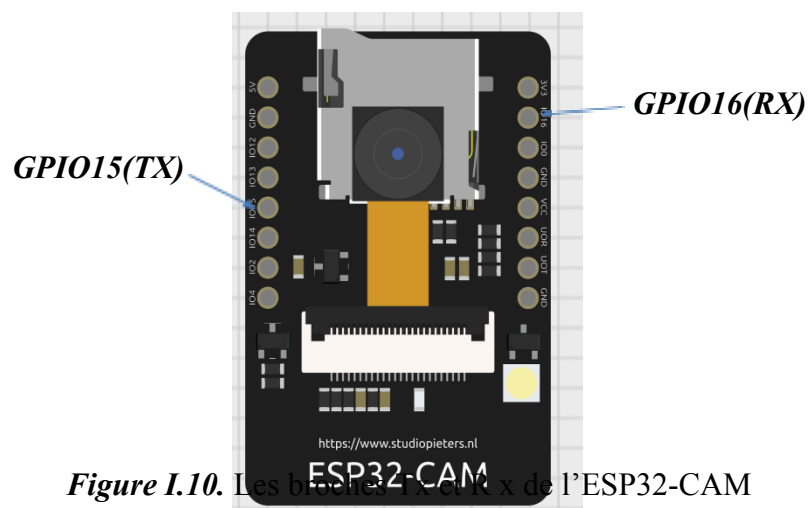


Figure I.10. Les broches TX et RX de l'ESP32-CAM

Une fois reçu, le programme est intégralement stocké dans une mémoire **Flash** non volatile. Cette mémoire, appelée "mémoire de programme", conserve le code même lorsque la carte est hors tension.

Lorsque la carte ESP32-CAM démarre, le processeur, que l'on peut considérer comme le "cerveau" du système, charge le programme depuis la mémoire Flash et commence à traiter les données, les répartissant dans différentes zones de mémoire, telles que :

Mémoire de programme (Flash) : Cette mémoire stocke le programme écrit et téléversé. Le code y est conservé même après un redémarrage ou une coupure d'alimentation.

Mémoire de données (RAM) : Aussi appelée **SRAM**, cette mémoire stocke temporairement les variables dynamiques comme les numéros de broches associées à des composants (LED, capteurs, etc.) ou des valeurs fluctuantes (chiffres, caractères, etc.). Les données dans la SRAM sont effacées à chaque redémarrage ou extinction de la carte.

Ce processus permet à l'ESP32-CAM d'exécuter le programme de manière optimale, tout en gérant efficacement les entrées-sorties et les ressources mémoire pour des applications complexes telles que la télésurveillance et la capture d'image.

1.6.2. Alimentation de la carte esp 32:

La Figure (I.11) montre les prises d'alimentation de la carte esp32 :

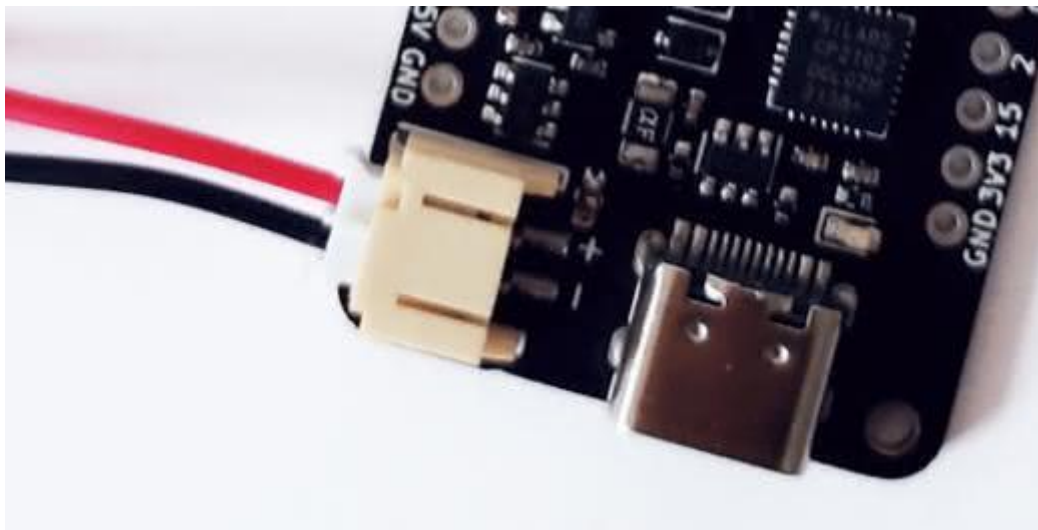


Figure I.11. Alimentation de la carte esp32

- **Alimentation de la carte esp32 par USB**

L'alimentation de la carte ESP32 par USB est une méthode simple et pratique pour le développement. Voici les points d'alimentation :

***Connecteur USB** : La plupart des cartes utilisent un connecteur Micro-USB ou USB-C.

***Tension d'Entrée** : L'ESP32 fonctionne à 3,3 V, avec une tension standard de 5 V fournie par USB, régulée par un circuit intégré.

***Courant de Sortie** : Le port USB fournit jusqu'à 500 mA, suffisant pour alimenter l'ESP32 et les périphériques connectés.

***Facilité de Programmation** : La connexion USB permet un téléversement facile du code et le débogage via des environnements de développement comme Arduino IDE.

1.6.3. Les broches d'alimentation de carte esp 32

L'ESP32 est un microcontrôleur puissant qui nécessite une gestion adéquate de l'alimentation pour fonctionner correctement. Voici un aperçu des principales broches d'alimentation présentes sur la carte :

1. Vin (V5 ou VUSB)

La broche Vin, également appelée V5 ou VUSB, est la connexion par laquelle la carte reçoit une tension d'entrée, généralement 5 V, lorsqu'elle est alimentée via un port USB. Cette tension est ensuite régulée par un circuit intégré pour fournir une alimentation stable au microcontrôleur.

2. V3.3 (VCC)

La broche V3.3, souvent désignée VCC, délivre une tension régulée de 3,3 V. C'est cette tension qui alimente l'ESP32 lui-même, ainsi que d'autres composants ou capteurs externes qui nécessitent une alimentation à 3,3 V. Cette broche est cruciale pour le bon fonctionnement des circuits.

3. GND (Masse)

Les broches GND sont les points de masse de la carte. Elles sont essentielles pour créer un circuit commun, permettant à tous les composants de partager une référence de tension. Une bonne connexion à la masse est vitale pour éviter des problèmes de fonctionnement et des interférences.

4. EN (Enable)

La broche EN, ou Enable, est utilisée pour activer ou désactiver le microcontrôleur. En la mettant à l'état bas, l'ESP32 se met en veille ou s'éteint complètement. Cela peut être utile pour gérer la consommation d'énergie dans les applications portables.

5. IO Pins (GPIO)

Bien que principalement utilisées pour la communication et le contrôle, certaines broches GPIO (General Purpose Input/Output) peuvent également être employées pour alimenter des capteurs et des modules externes. Cependant, il est important de respecter les limites de courant spécifiées pour éviter d'endommager le microcontrôleur.

1.6.4. La Mémoire

Mémoire du Microcontrôleur ESP32 est conçu avec plusieurs types de mémoire qui jouent un rôle essentiel dans son fonctionnement. Voici un aperçu des principales catégories de mémoire :

1. RAM (Random Access Memory)

- **Type** : SRAM (Static Random Access Memory)
- **Capacité** : Environ 520 Ko.
- **Fonction** : Utilisée pour stocker temporairement les données et les variables pendant l'exécution des programmes. Sa rapidité permet un accès instantané, ce qui est crucial pour le traitement en temps réel.

2. Mémoire Flash

- **Type** : Flash SPI (Serial Peripheral Interface)
- **Capacité** : Typiquement entre 4 Mo et 16 Mo, selon le modèle de la carte.
- **Fonction** : Stocke le firmware, les programmes, et les fichiers de données. Contrairement à la RAM, la mémoire flash conserve les données même lorsque l'alimentation est coupée, ce qui est essentiel pour le stockage permanent des applications.

3. Partitionnement de la Mémoire

L'ESP32 utilise un système de partitionnement de la mémoire flash qui permet de diviser l'espace en sections dédiées :

- **Firmware** : Contient le code principal de l'application.
- **SPIFFS (SPI Flash File System)** : Permet le stockage de fichiers, tels que des configurations ou des ressources.
- **NVS (Non-Volatile Storage)** : Utilisé pour stocker des paramètres de configuration et des données persistantes.

4. Mémoire RTC (Real-Time Clock)

- **Fonction** : Une petite quantité de mémoire RTC est incluse pour conserver des données critiques même lorsque le microcontrôleur est en mode veille. Cela est utile pour les applications qui nécessitent le stockage d'informations importantes qui doivent être disponibles après un réveil.

1.6.5. Entrées et sorties numériques esp 32

L'ESP32 dispose de jusqu'à 34 broches GPIO (General Purpose Input/Output) qui peuvent être configurées comme entrées ou sorties.

- **Entrées** : Utilisées pour lire l'état de capteurs et boutons.
- **Sorties** : Permettent de contrôler des dispositifs comme des LED et des relais.

Fonctions Avancées

- **Interruptions** : Certaines broches peuvent déclencher des interruptions pour une réponse rapide.
- **Pull-up/Pull-down** : Résistances internes pour stabiliser les entrées.

1.6.6. Broches analogiques esp 32 :

L'Esp32 dispose de 6 broches analogiques, numérotées GPIO34 à GPIO39, utilisées principalement pour lire des signaux analogiques à partir de capteurs ou de potentiomètres. Ces broches permettent des conversions analogiques-numériques (ADC) avec une résolution de 12 bits.

Les fonctions principales associées aux broches analogiques incluent 'analogRead(pin)', qui lit la valeur analogique d'une broche spécifiée, et 'analogReference(type)', qui définit la référence de tension utilisée pour les lectures analogiques. Les options de référence de tension incluent 'DEFAULT' (5V), 'INTERNAL' (1.1V), et 'EXTERNAL' (tension appliquée à la broche AREF).

1.6.7. Autres broches esp 32

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.

1.6.8. Communications de la carte esp 32

L'Esp32 propose plusieurs méthodes robustes de communication pour échanger des données avec divers périphériques externes. La communication série, utilisant les broches RX et TX, permet des échanges bidirectionnels de données avec des capteurs, des modules Bluetooth, et d'autres microcontrôleurs. Pour des configurations nécessitant la gestion de multiples périphériques sur un seul bus, le protocole I2C, exploité via les broches SDA et SCL, offre une solution efficace. En revanche, la communication SPI, implémentée sur les broches SS, MOSI, MISO et SCK, garantit des transferts rapides et synchrones adaptés aux écrans TFT, aux cartes SD, et aux modules WiFi. Enfin, le logiciel SoftwareSerial permet de créer des ports série supplémentaires sur n'importe quelles broches numériques, idéal lorsque les broches RX et TX sont déjà utilisées. Ces méthodes de communication diversifiées confèrent à l'esp32 une grande flexibilité pour une gamme étendue d'applications électroniques et robotiques.

1.6.9. Protection du port USB contre la surcharge en intensité esp 32

La carte Arduino uno 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.

1.7. Présentation logicielle de Arduino IDE

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 sur la carte esp32 .
- * de se connecter avec la carte esp32 pour y transférer les programmes.
- * de communiquer avec la carte esp32.

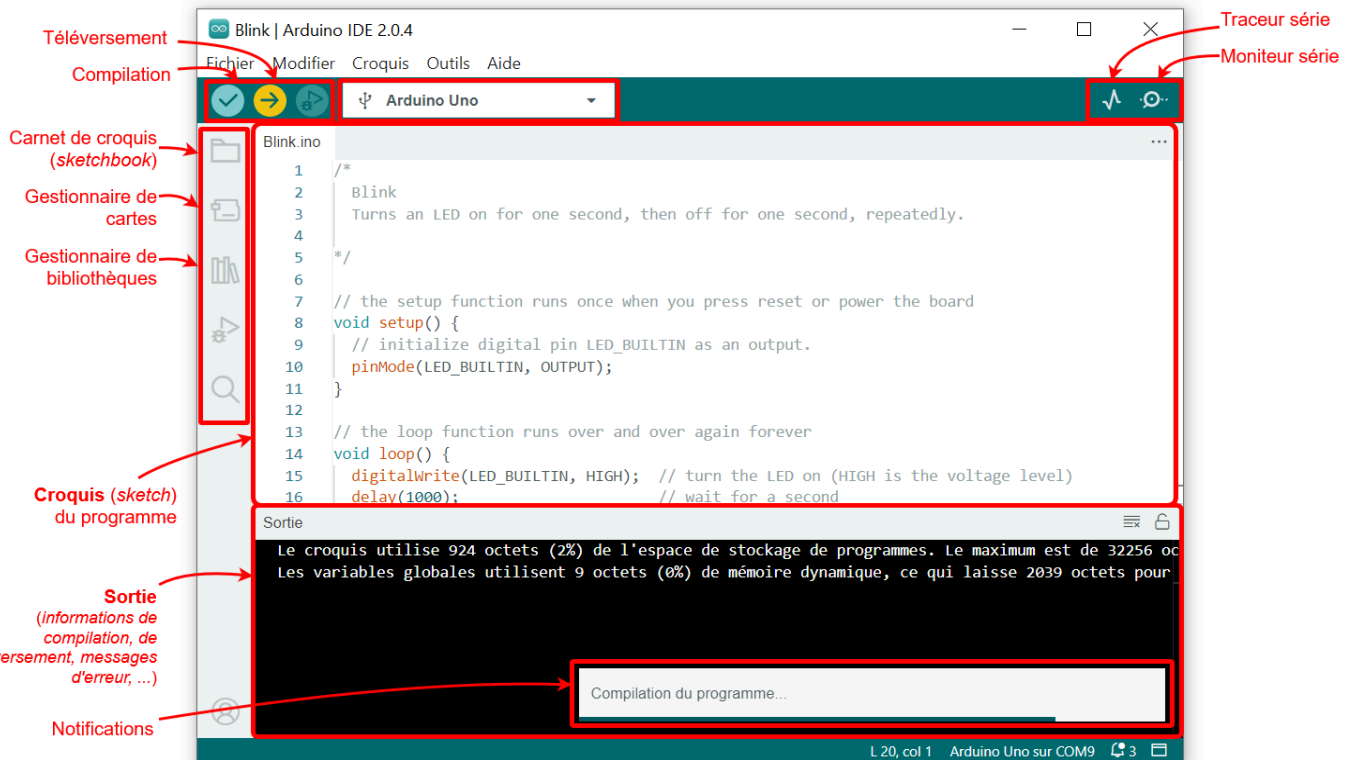


Figure I.12. Les différentes parties de la fenêtre principale du logiciel *Arduino IDE*

La programmation des cartes esp32 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 vous 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 série sur la Figure (I.10), permet d'afficher des messages textes reçus de la esp32 et d'envoyer des caractères vers la carte esp32.

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

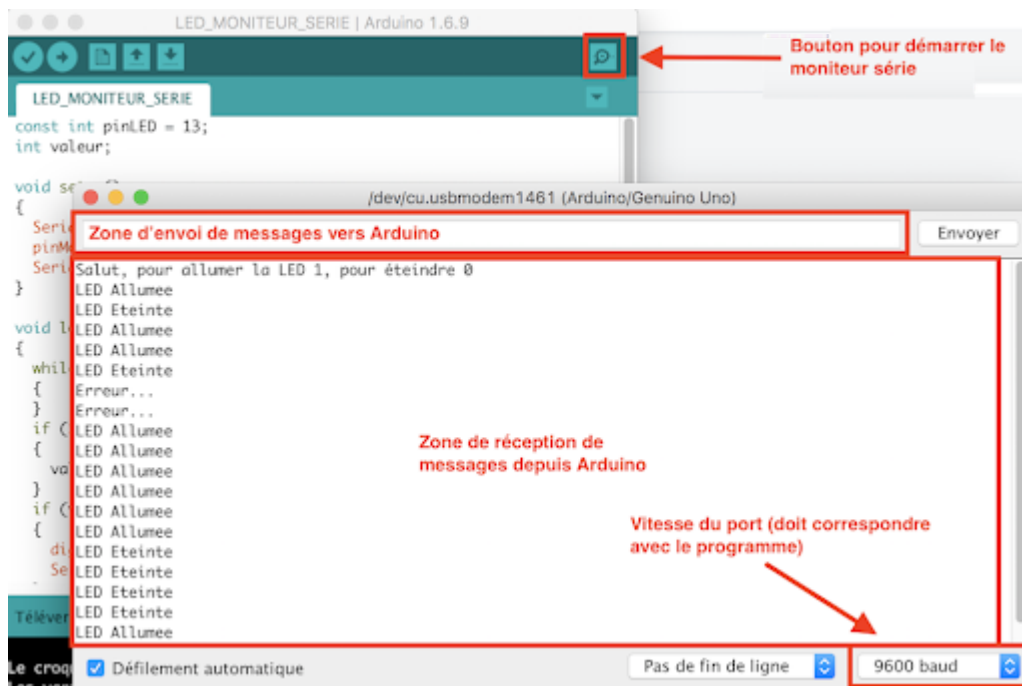


Figure I.13. Le moniteur série

I.7.1. Description de la barre d'outils

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

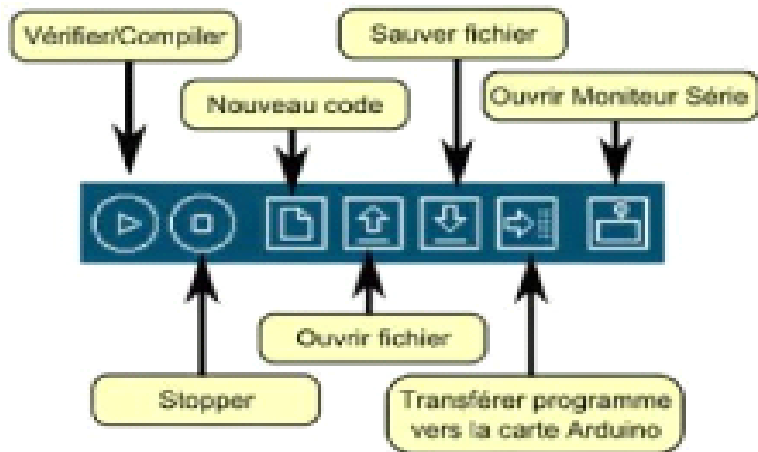


Figure I.14. la barre d'outils

CHAPITRE II

Les éléments de systèmes de télésurveillance

II.1 introduction

On ressent de plus en plus le besoin de renforcer les niveaux de sécurité dans divers secteurs d'activité. Un des moyens employés est la surveillance à distance. La télésurveillance est présente dans de nombreux domaines d'activité (banque, transport, industrie, etc.) ou lieux de vie (ville, immeubles de bureaux, etc.) afin de surveiller et de préserver des individus et des biens. Au cours de ce chapitre, nous examinerons les éléments clés de notre système de télésurveillance : l'Arducam shield V2, la caméra (OV2640) et le capteur infrarouge passif PIR. Enfin, nous concluons par un module WIFI (ESP32).

II.2. Définition de la télésurveillance

La télésurveillance est une technique de sécurité pratique et efficace pour surveiller des zones à distance et alerter en cas d'incident. Elle permet également de réduire les coûts de sécurité en évitant la présence physique permanente de gardiens ou de personnel de sécurité sur place. Cependant, elle peut également soulever des questions de respect de la vie privée et de protection des données personnelles, ainsi que des risques de piratage ou de vol de données

II.2.1. Domaine d'applications.

Dans de nombreuses situations, la télésurveillance est utilisée pour des raisons de sécurité :

Lorsqu'il s'agit de la sécurité routière, il est possible d'évaluer la densité du trafic en utilisant des caméras spécialisées ou des capteurs situés à proximité, voire même enfouis dans la chaussée, ainsi que la présence de personnes sur les bandes d'arrêt d'urgence, entre autres.

- En ce qui concerne la surveillance des machines, différents capteurs permettent d'évaluer l'état de la machine, ce qui permet d'envoyer ces informations à un poste de supervision. On pourrait ainsi repérer à distance l'épuisement des consommables, une anomalie de fonctionnement ou même un acte de malveillance.

- 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

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 visionner 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.

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 mission de repérer et repérer toute activité dans les zones surveillées.

II.2.3.2. Capteur d'images.

Il est possible de visualiser et d'enregistrer les images du lieu à protéger grâce à des caméras de surveillance qui les transmettent par vidéo, réseau IP ou sans fil à un centre de télésurveillance. Transmetteur

II.2.3.3. Un transmetteur

Un transmetteur téléphonique numérique ou un transmetteur GSM complètent parfois le système et signalent le centre de télésurveillance ou le propriétaire de l'habitation dès qu'une intrusion est détectée.

II.2.3.4. L'enregistreur

L'enregistreur offre la possibilité d'enregistrer et de conserver des informations recueillies (images, sons, etc.) afin de les utiliser ultérieurement.

II.3. Les capteurs

Les mesures sont des grandeurs physiques (température, pression, champ magnétique, etc.). Nous définissons les termes suivants :

- Mesuré : la grandeur physique mesurée (pression, température, etc.),
- Mesure : Toutes les opérations permettant d'obtenir des valeurs quantitatives physique (mesuré),
- Mesure : représente la valeur mesurée (6 MPa, 20°C, 2 m.s-1...)

II.3.1. Définition d'un capteur

Un capteur, de manière générale, est un dispositif conçu pour détecter et mesurer une grandeur physique ou une propriété spécifique de son environnement, puis convertir cette information en un signal généralement électrique ou analogique. Ce signal peut ensuite être traité, analysé ou utilisé pour effectuer une action spécifique, comme contrôler un système, afficher une mesure, ou transmettre des données à un dispositif externe.

Les capteurs sont des éléments essentiels dans une grande variété de domaines, de l'automobile à l'industrie, en passant par l'électronique grand public, la médecine, l'aérospatiale et bien d'autres. Ils permettent de surveiller, contrôler et automatiser un large éventail de processus et de systèmes, contribuant ainsi à améliorer l'efficacité, la sécurité et la qualité de vie

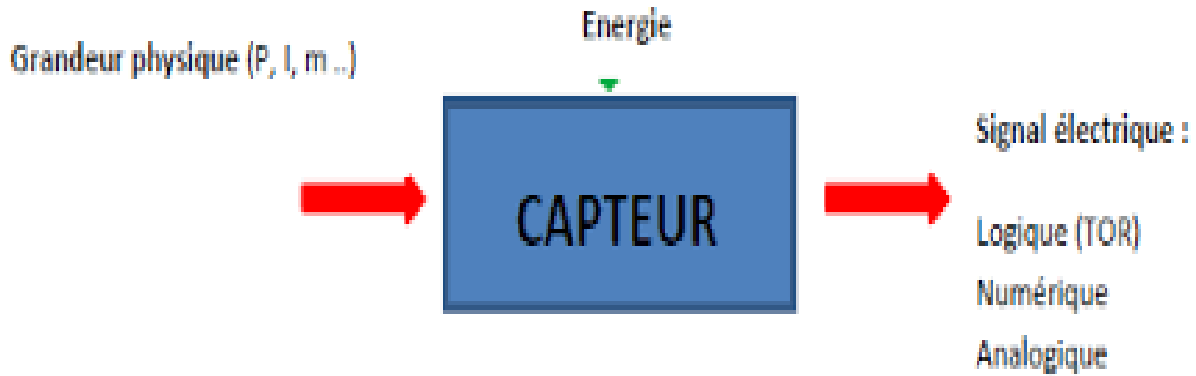


Figure II.1. la chaîne d'acquisition d'un capteur

La grandeur de sortie du capteur peut varier :

- En termes binaires (informations vraies ou fausses), il s'agit d'un capteur Tout Or Rien (TOR) ;
- Progressif (changement continu), est un capteur analogique ;
- Niveau de tension ou de courant, c'est un capteur numérique

II.3.2. Chaîne de mesure

Généralement, le signal de sortie du capteur ne peut pas être utilisé directement. nous La chaîne de mesure désigne l'ensemble des amplifications, adaptations, Le signal est converti, linéarisé et numérisé avant d'être lu sur le support de sortie. Pour obtenir une image d'une grandeur physique, une chaîne de mesure peut être utilisée Plusieurs phénomènes différents. Par exemple, la mesure du débit peut être effectuée dans plusieurs Trace de pas:

- ✓ Convertir le débit en pression différentielle.
- ✓ Convertir la différence de pression en déformation mécanique membrane.
- ✓ Convertir la déformation mécanique en électricité (en utilisant des éléments piézoélectriques) à travers des circuits associés.

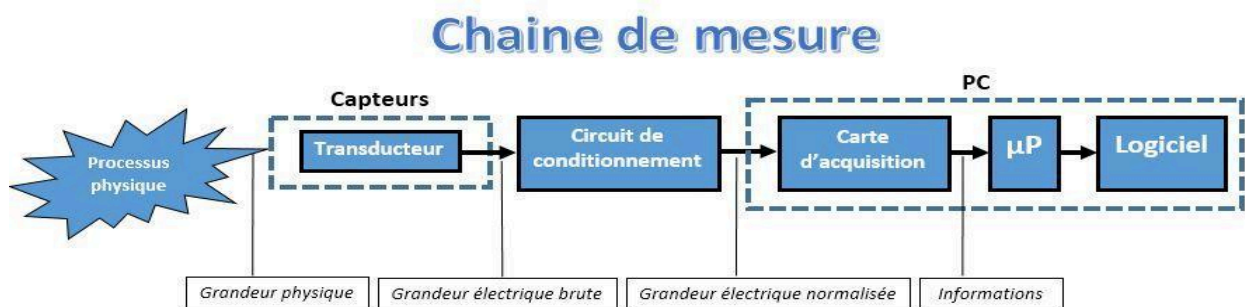


Figure II.2. Chaîne de mesure

II.3.3. Constitution d'un capteur

Certains capteurs sont des capteurs dits « composites », c'est à dire qu'ils sont constitués de 2 capteurs Des partis avec des rôles clairs :

- Corps d'épreuve : C'est un élément qui réagit sélectivement aux changements de l'environnement. La quantité à mesurer. Sa fonction est de transformer cette grandeur en une autre sorte de grandeur La quantité physique dite mesurable.
- Élément de transduction : C'est un élément connecté au corps d'épreuve et peut être traduit Sa réaction est convertie en grandeurs physiques utilisables

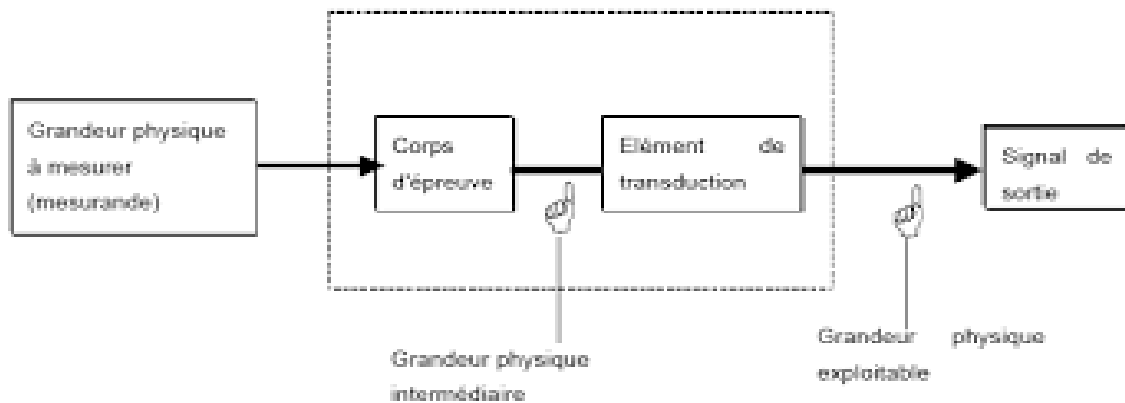


Figure II.3. Constitution d'un capteur

II.3.4. Rôle du capteur

Le rôle d'un capteur dépend de son type et de son utilisation spécifique, mais en général, un capteur est un dispositif qui détecte et mesure des phénomènes physiques, chimiques, biologiques ou autres, et convertit ces informations en signaux utilisables. Voici quelques-uns des rôles principaux des capteurs :

1. **Détection** : Les capteurs peuvent détecter la présence ou l'absence de certaines substances, objets, ou phénomènes tels que la lumière, le mouvement, la pression, la température, etc.
2. **Mesure** : Les capteurs mesurent des grandeurs physiques comme la température, la pression, l'humidité, la force, la vitesse, etc. Ces mesures peuvent être utilisées pour surveiller des systèmes, contrôler des processus ou collecter des données.
3. **Contrôle** : Dans certains cas, les capteurs sont utilisés pour fournir des informations en temps réel à des systèmes de contrôle. Par exemple, un capteur de température peut envoyer des données à un système de climatisation pour maintenir une température désirée.
4. **Surveillance** : Les capteurs sont souvent utilisés pour surveiller l'environnement, les conditions de travail ou la santé des personnes. Par exemple, des capteurs de pollution peuvent mesurer la qualité de l'air, tandis que des capteurs médicaux peuvent surveiller les signes vitaux d'un patient.

5. **Navigation** : Dans le domaine de la robotique et des véhicules autonomes, les capteurs sont utilisés pour percevoir l'environnement et prendre des décisions en conséquence. Les capteurs de distance, les caméras, les lidars et les radars sont souvent utilisés à cette fin.
6. **Sécurité** : Les capteurs sont utilisés pour détecter les intrusions, les incendies, les fuites de gaz, etc., contribuant ainsi à assurer la sécurité des personnes et des biens.

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

1. **Sensibilité** : Capacité à détecter de petites variations dans la grandeur mesurée.
2. **Plage de mesure** : Intervalle de valeurs mesurables par le capteur.
3. **Précision** : Proximité des mesures par rapport à la valeur réelle.
4. **Résolution** : Plus petite variation détectable par le capteur.
5. **Linéarité** : Proportionnalité entre la sortie du capteur et la grandeur mesurée.
6. **Hystérésis** : Différence de sortie pour la même entrée selon le chemin parcouru.
7. **Temps de réponse** : Vitesse à laquelle le capteur réagit aux changements.
8. **Durabilité et robustesse** : Résistance aux conditions difficiles et longévité.
9. **Bruitage et stabilité** : Niveau de bruit et constance des mesures.
10. **Consommation d'énergie** : Énergie requise pour fonctionner.
11. **Compatibilité et intégrabilité** : Facilité d'intégration dans des systèmes existants.
12. **Coût** : Coût initial et coûts de maintenance et de fonctionnement.

II.3.7. Capteur infrarouge Passif (PIR)

Pour le système que nous allons créer, nous utiliserons le PIR. c'est un capteur Il mesure le rayonnement infrarouge émis par les objets situés dans son champ de vision. Imaginer. Il est couramment utilisé comme détecteur de mouvement dans divers systèmes de sécurité.

Sécurité. Le capteur PIR détecte les mouvements jusqu'à 6 m

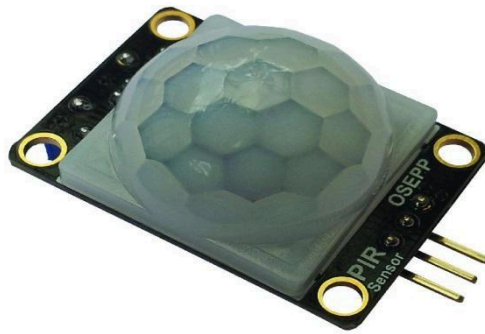


Figure II.4. Capteur infrarouge passif PIR.

II.3.8. Principe de fonctionnement

Le rayonnement thermique émis par tous les corps ayant une température supérieure à celle du 0 absolu peut être analysé par un détecteur infrarouge passif. La température externe de l'homme est d'environ 35°C. Il émet ainsi une lumière infrarouge. Un détecteur infrarouge a pour fonction de distinguer le rayonnement émis par un individu de celui émis par des objets, de déterminer si la source d'émission du rayonnement est fixe ou en mouvement, et d'envoyer une information dans le cadre d'une détection (Fig. II.5).

Le PIR, également connu sous le nom de détecteur de mouvement à infrarouge passif, allie un capteur pyroélectrique à une électronique de conditionnement (adaptation, traitement) du signal et une optique à base de lentille de Fresne.

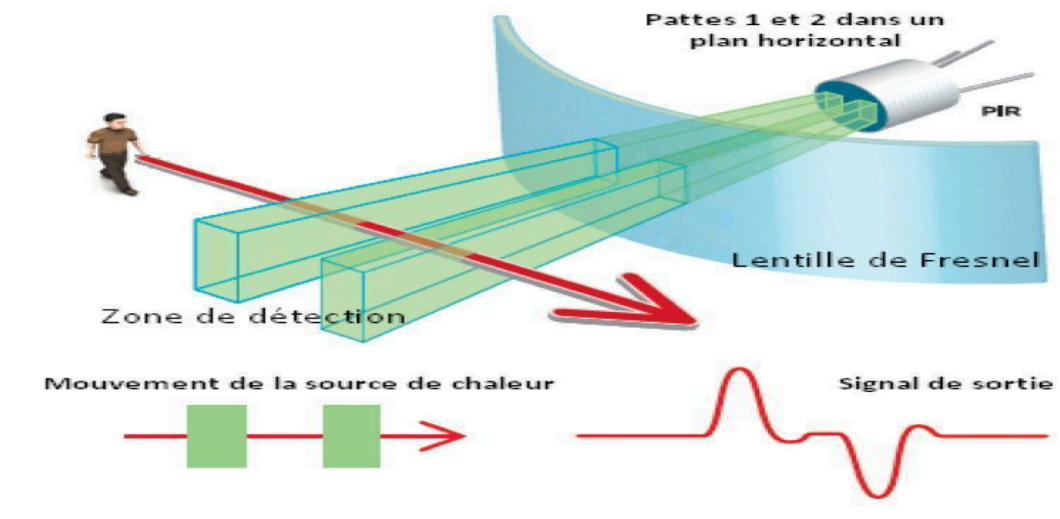


Figure II.5. Principe de fonctionnement d'un capteur

*** 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.



Figure II.6. Capteur de mouvement

II.3.9. Caractéristiques électriques et fonctionnelles de capteur PIR

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 :

Pour ajuster le PIR, deux potentiomètres sont disponibles. La sensibilité du capteur est réglée par l'un et le temps de réponse par l'autre (Figure II.7).

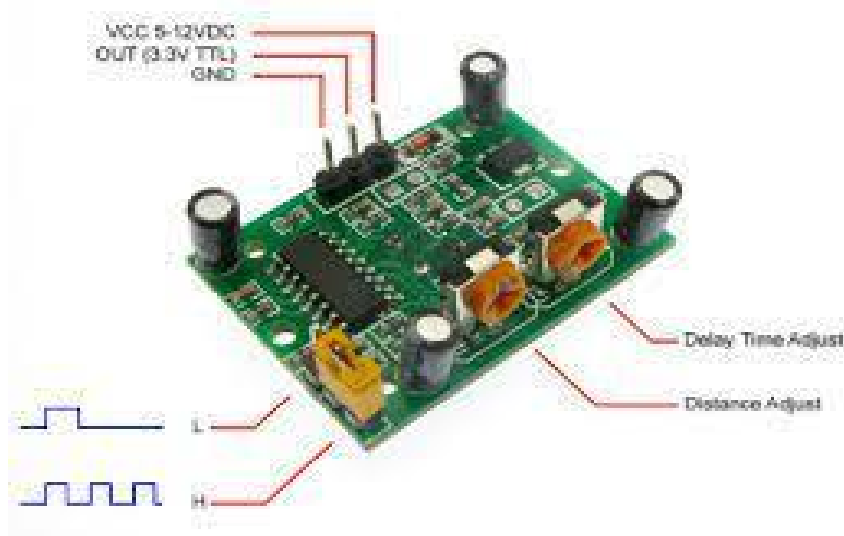
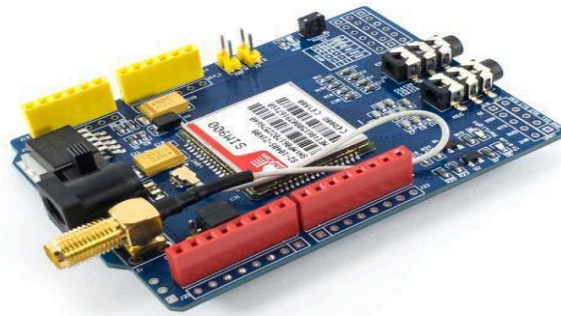


Figure II.7. Réglages du PIR

II.4. Module GSM sim900

Le module GSM SIM900 est un dispositif essentiel pour la communication mobile, permettant d'envoyer des SMS et de passer des appels via le réseau GSM. Il prend en charge les protocoles GSM/GPRS, ce qui facilite l'accès à Internet, et est compatible avec des microcontrôleurs comme Arduino et ESP32. Son utilisation s'étend à diverses applications, notamment les systèmes de sécurité et l'automatisation domestique. Le module fonctionne avec une alimentation de 3.4V à 4.4V et nécessite des connexions appropriées pour une intégration réussie dans des projets électroniques. Des bibliothèques comme `GSM.h` simplifient la programmation.

*Figure II.8.* Module GSM Sim900

II.5. Les caméras

C'est un système de capture d'images animées qui génère des signaux vidéo en noir et blanc ou en couleur. Un appareil photo est un outil qui capture des images en changeant la lumière en informations électriques. Il est parfois constitué d'un capteur d'image et d'une électronique traitée pour générer un signal vidéo. Actuellement disponible sur le marché, différents types de caméras sont disponibles pour répondre aux utilisateurs en fonction de leurs besoins cibles.

II.5.1. Les critères de choix d'une caméra

Le choix des caméras de surveillance à distance dépend en grande partie des besoins et Exiger. En plus des différents types de caméras, leurs performances diffèrent également. à propos :

- * 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.5.2. 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

***Caméra CCD (Charge Coupled Device)**

Les capteurs CCD ne font pas de différence entre les couleurs. L'analyse des couleurs est Réalisée avec un filtre qui permet de récupérer le signal RVB avant traitement

Figure II.9. Caméra



CCD

*** Caméra analogique**

Ces caméras sont faciles à identifier d'une sortie de type BNC, Connectez-vous via un câble coaxial

caméras identifier d'une

Figure II.10.
analogique

***Caméra**

réfère à la



Caméra

infrarouge

Si l'on se

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.11. 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.12. Caméra CMOS.

***Caméra discrète**

Les progrès technologiques signifient que les caméras peuvent désormais être montées sur Tous les accessoires. Les plus courants sont les détecteurs d'intrusion, les horloges et les têtes détection d'incendie



Figure II.13. 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 informatique

Parmi ces différents types de on a utilisé la caméra



Figure II.14. Caméra numérique

II.5.2. La caméra OV2640

La caméra OV2640 est un module de caméra compact et efficace, souvent utilisé dans des projets basés sur des microcontrôleurs comme l'ESP32. Avec une résolution maximale de 2 mégapixels (1600 x 1200 pixels), elle offre une qualité d'image convenable pour diverses applications, telles que la surveillance et l'Internet des objets (IoT). L'OV2640 utilise une interface de communication Serial Camera Control Bus (SCCB), similaire à I2C, pour la configuration et le contrôle. Ce module prend en charge plusieurs formats de sortie, notamment JPEG et RAW, permettant une compression efficace des images pour optimiser l'utilisation de la bande passante. Son intégration avec l'ESP32-CAM permet des fonctionnalités avancées telles que le streaming vidéo en temps réel et la capture de photos sur détection de mouvement, faisant de l'OV2640 un choix populaire pour les systèmes de télésurveillance et d'analyse d'image.



Figure II.15. Caméra OV2640

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.4.3. Identification des pins

Le tableau suivant montre les déférents pins de la caméra OV2640

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

CHAPITRE III

La réalisation pratique

III .1. Introduction

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

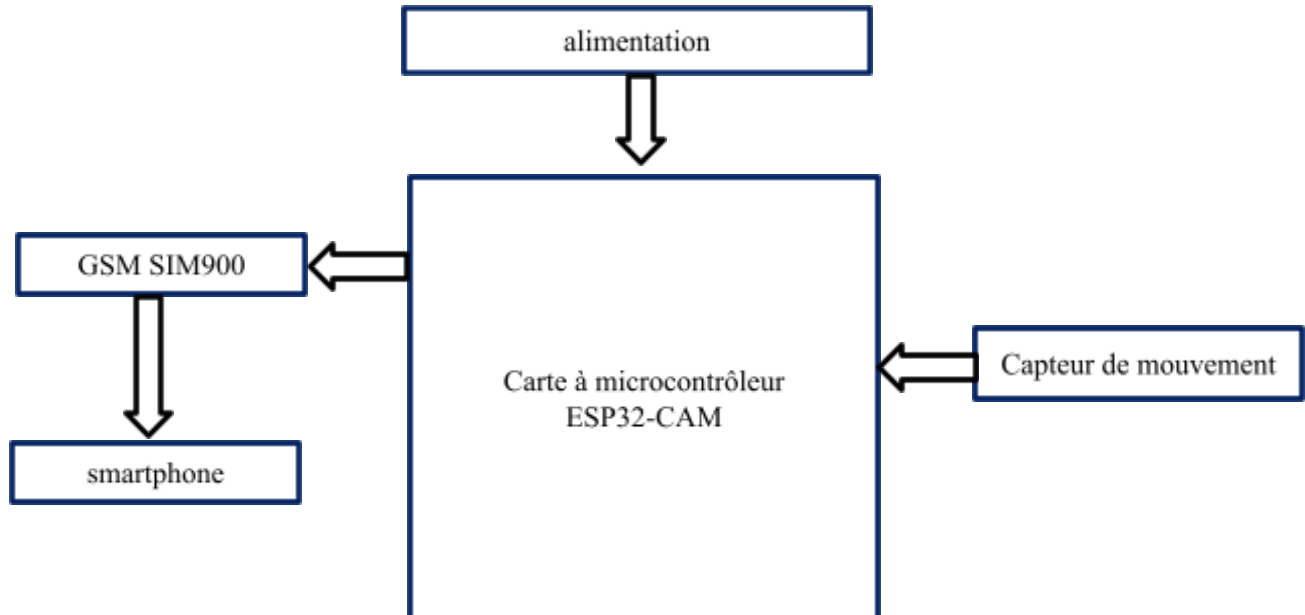


Figure III.1. schéma synoptique du system



III .2.2. Différents éléments utilisés dans le projet

Pour réaliser notre système de télésurveillance, nous avons utilisé les composants suivants : une carte ESP32-CAM, un module SIM900, un capteur PIR, et une carte mémoire SD pour l'enregistrement des photos.

III .2.2.1. Branchement du module SIM900 à la carte ESP32-CAM

La figure suivante illustre le branchement des différents pins du module SIM900 avec l'ESP32-CAM, en détaillant les connexions indispensables pour assurer la communication entre ces deux composants dans le cadre du système de télésurveillance.

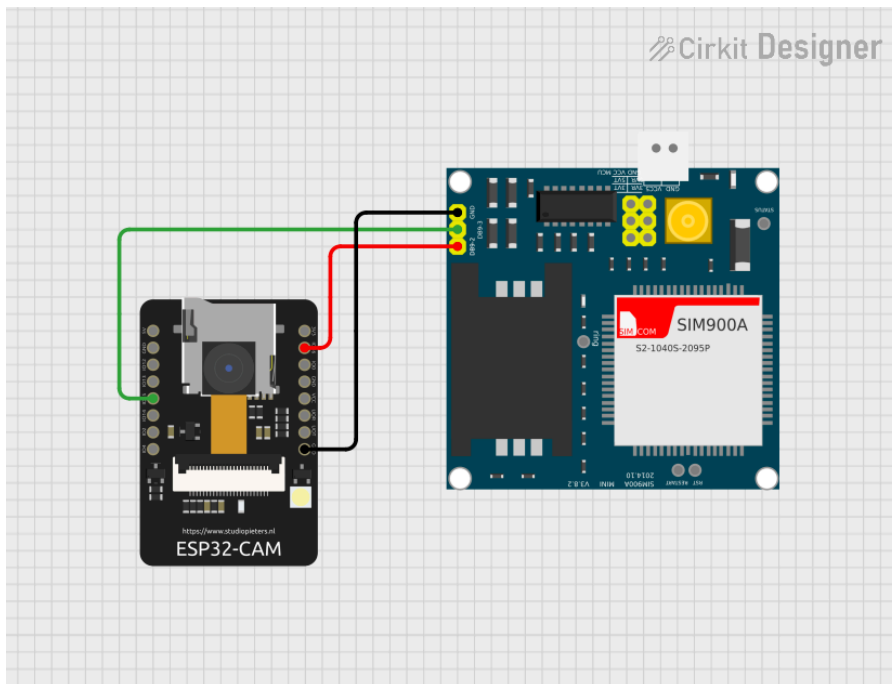


Figure III.2. Branchement de la SIM900 avec ESP32-CAM

III .2.2.3. Branchement du capteur PIR à la carte ESP32-CAM

Les capteurs PIR (infrarouges passifs) détectent la présence de personnes en mouvement dans leur champ de vision. Ils sont couramment employés dans divers systèmes de sécurité et détecteurs de mouvements. Compacts, abordables et à faible consommation d'énergie, ces

capteurs sont particulièrement prisés.

Fonctionnant comme des capteurs numériques, le comportement du capteur PIR est le suivant

- Lorsqu'un mouvement est détecté, le signal de sortie passe à un niveau haut (1).
- En l'absence de mouvement, le signal de sortie est à un niveau bas (0).

Pour connecter le capteur PIR à un système de télésurveillance, il suffit de brancher la broche d'alimentation positive (VCC) au 5V de l'alimentation, la broche de masse (GND) à la terre, et la broche de sortie (OUT) à une entrée numérique du système.

La figure suivante illustre ces connexions.

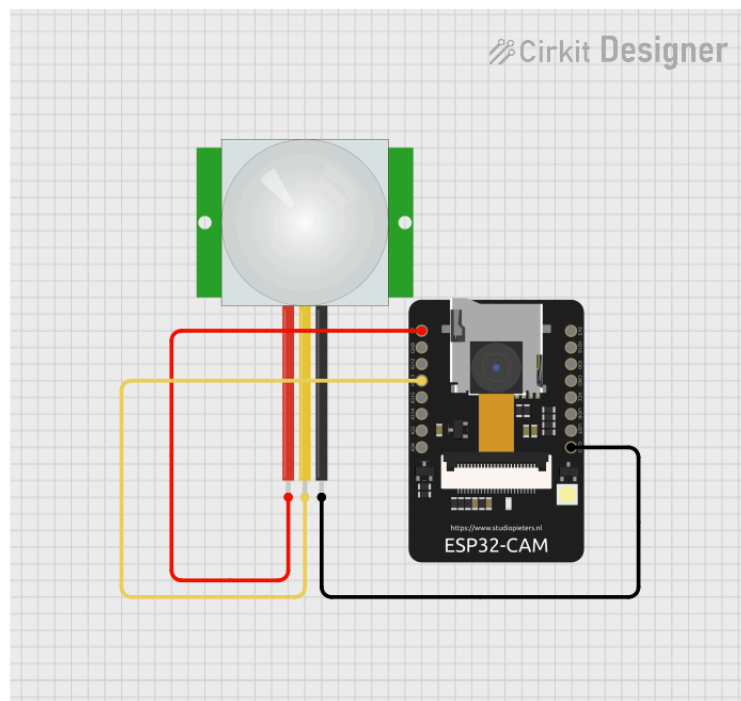


Figure III.3. Branchement de capteur PIR



III.2.3. Câblage du système

En tenant compte des tests de bon fonctionnement de chaque élément individuellement, le câblage général des différents composants vers l'ESP32-CAM a été réalisé, comme l'illustre la figure ci-dessous :

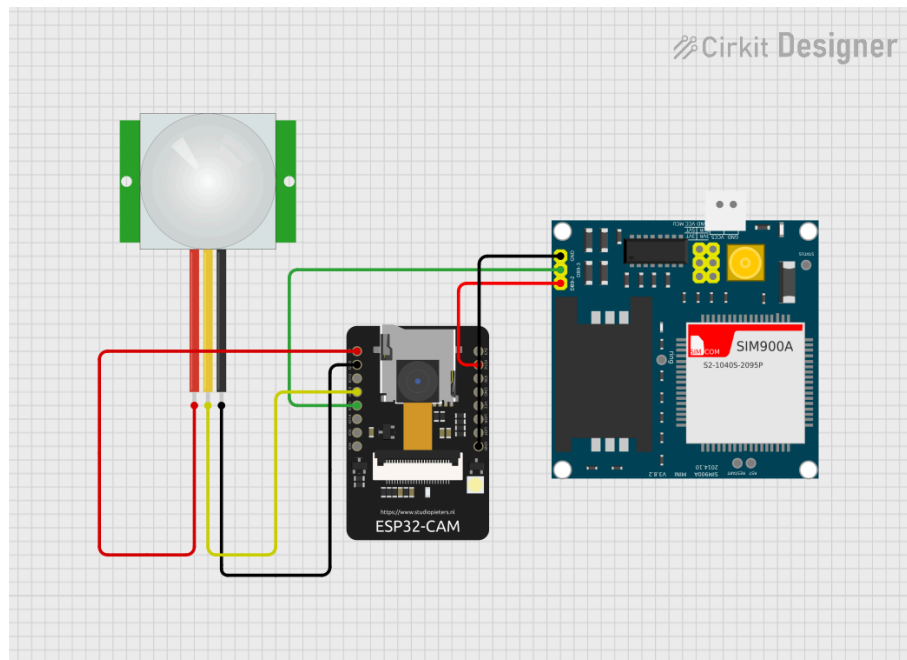
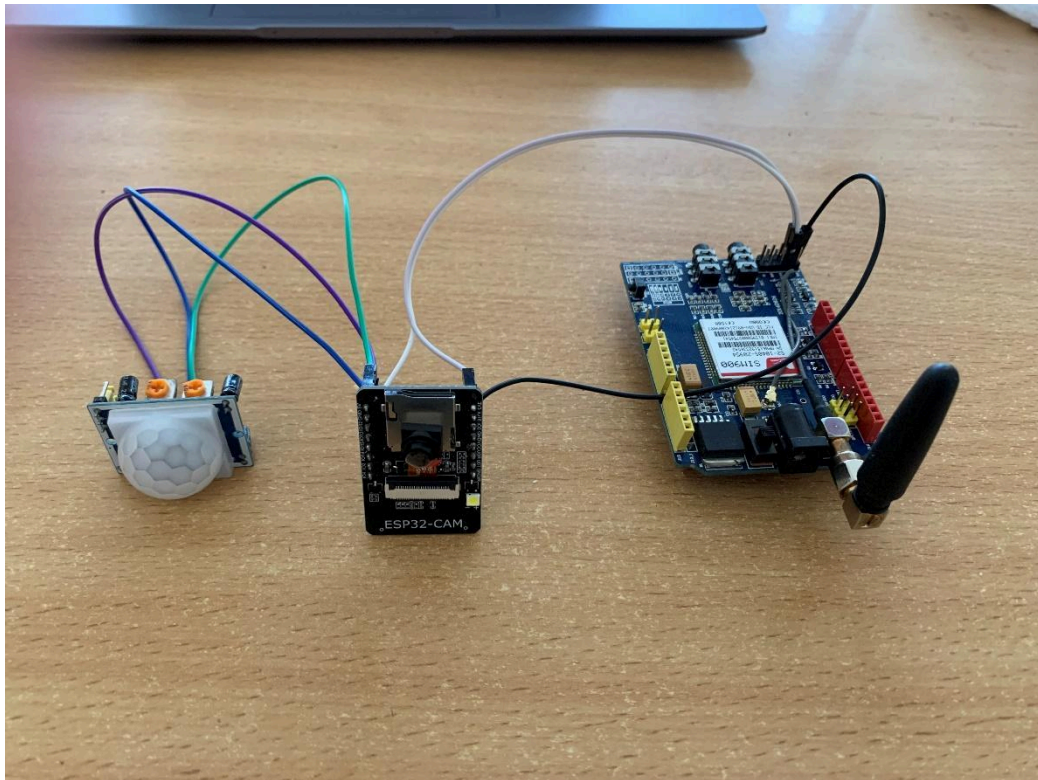


Figure III.4. Câblage de système



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



La figure III.5. *Montage de notre système*

III .2.4. L'organigramme

Avant de débiter la programmation, il est essentiel de concevoir un organigramme qui décrit le déroulement des différentes séquences, à la fois internes et externes. Cet organigramme inclut plusieurs boucles, chaque cycle se terminant toujours par son point de départ.



Les actions sont représentées par des rectangles, tandis que les choix sont indiqués par des losanges. La figure suivante présente l'organigramme de notre système.

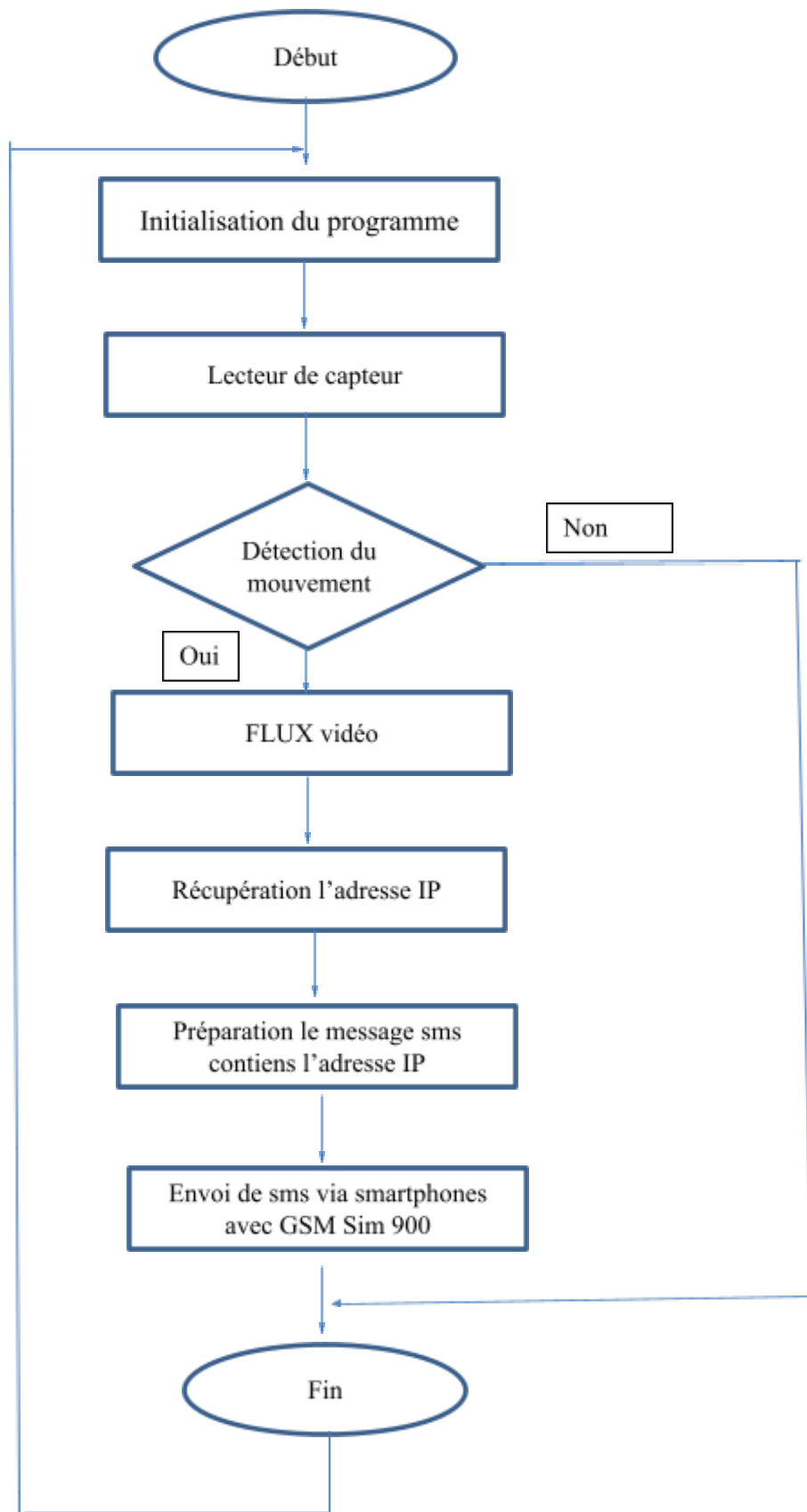


Figure III.6. *L'organigramme du fonctionnement général.*



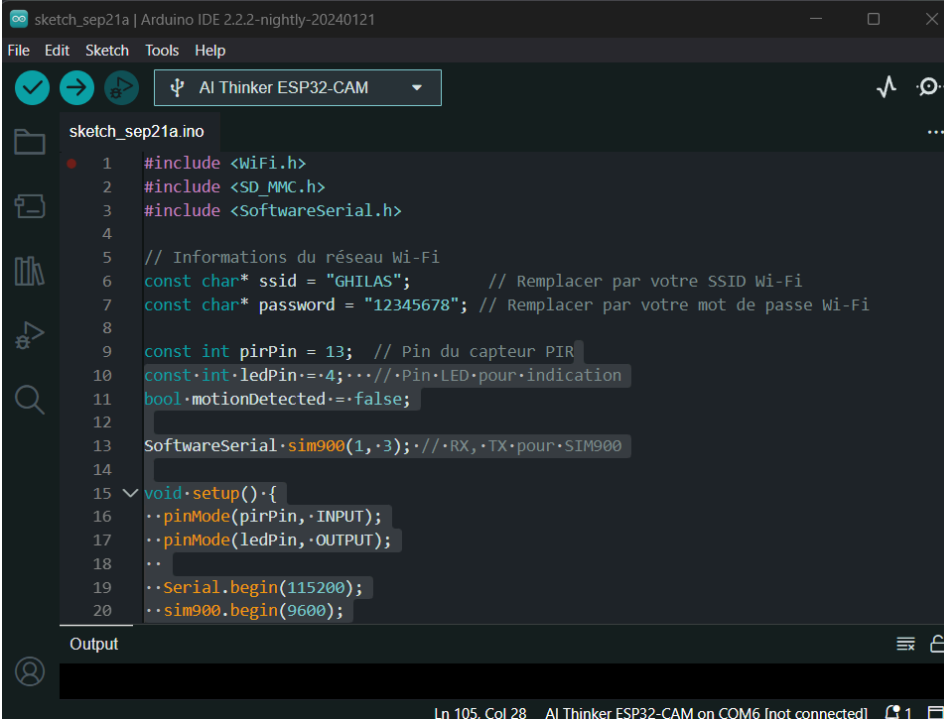
III .2.5. Les étapes de programmation de la carte ESP32-CAM

Pour programmer l'ESP32-CAM, il est d'abord nécessaire d'installer l'environnement de développement Arduino (IDE), qui permet de coder pour cette carte. Ensuite, il faut connecter la carte à l'ordinateur à l'aide d'un câble USB.

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

L'IDE Arduino est un environnement de développement intégré avec une interface graphique conviviale. Une fois l'IDE lancé, vous pouvez y saisir votre programme.

La figure suivante illustre l'interface de l'IDE Arduino.



```
sketch_sep21a | Arduino IDE 2.2.2-nightly-20240121
File Edit Sketch Tools Help
AI Thinker ESP32-CAM
sketch_sep21a.ino
1 #include <WiFi.h>
2 #include <SD_MMC.h>
3 #include <SoftwareSerial.h>
4
5 // Informations du réseau Wi-Fi
6 const char* ssid = "GHILAS"; // Remplacer par votre SSID Wi-Fi
7 const char* password = "12345678"; // Remplacer par votre mot de passe Wi-Fi
8
9 const int pirPin = 13; // Pin du capteur PIR
10 const int ledPin = 4; // Pin LED pour indication
11 bool motionDetected = false;
12
13 SoftwareSerial sim900(1, 3); // RX, TX pour SIM900
14
15 void setup() {
16   pinMode(pirPin, INPUT);
17   pinMode(ledPin, OUTPUT);
18   ..
19   Serial.begin(115200);
20   sim900.begin(9600);
Output
Ln 105, Col 28 AI Thinker ESP32-CAM on COM6 [not connected]
```

Figure III.7. L'écriture du programme Arduino

- **Compilation du programme**

La compilation permet de vérifier que votre programme respecte la syntaxe et que toutes les variables sont correctement définies, ainsi que que les bibliothèques ou fichiers nécessaires sont bien inclus. En cas d'erreurs, le compilateur indique le type d'erreur ainsi que la ligne concernée.

- **Sélection de type de carte**

Avant de téléverser le code vers la carte ESP32-CAM, 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 AI thinker ESP32-CAM) .comme le montre la figure suivante

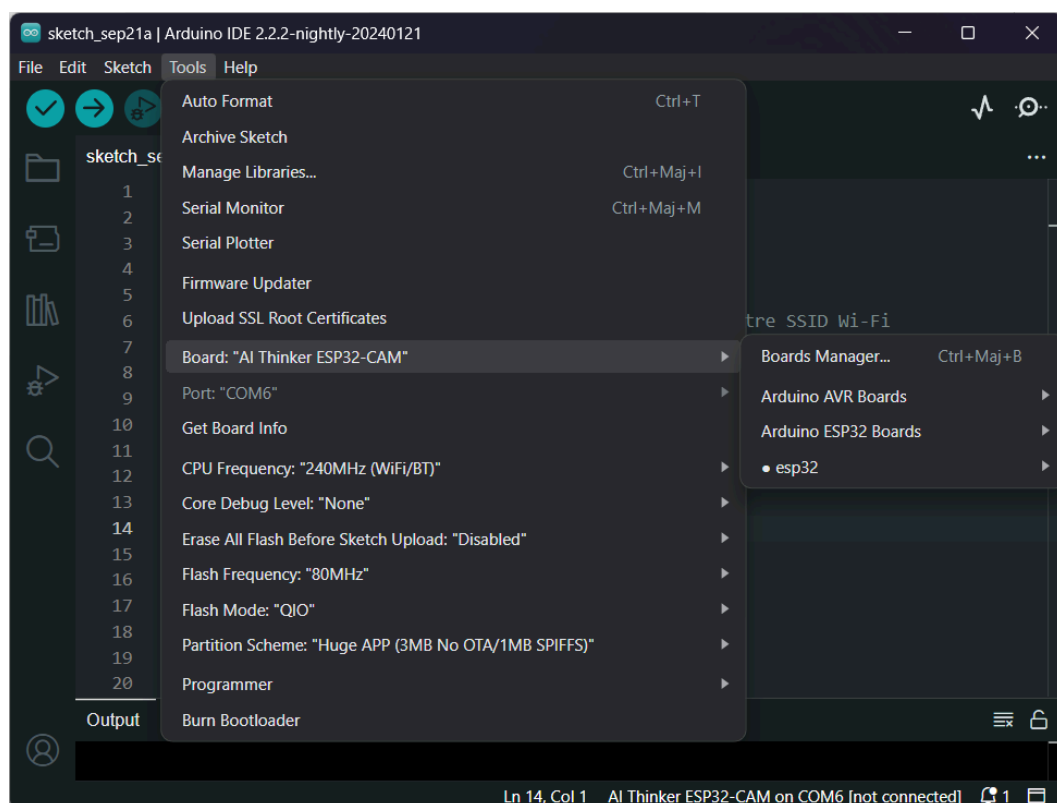


Figure III.8. Sélection de la cible



- **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 :

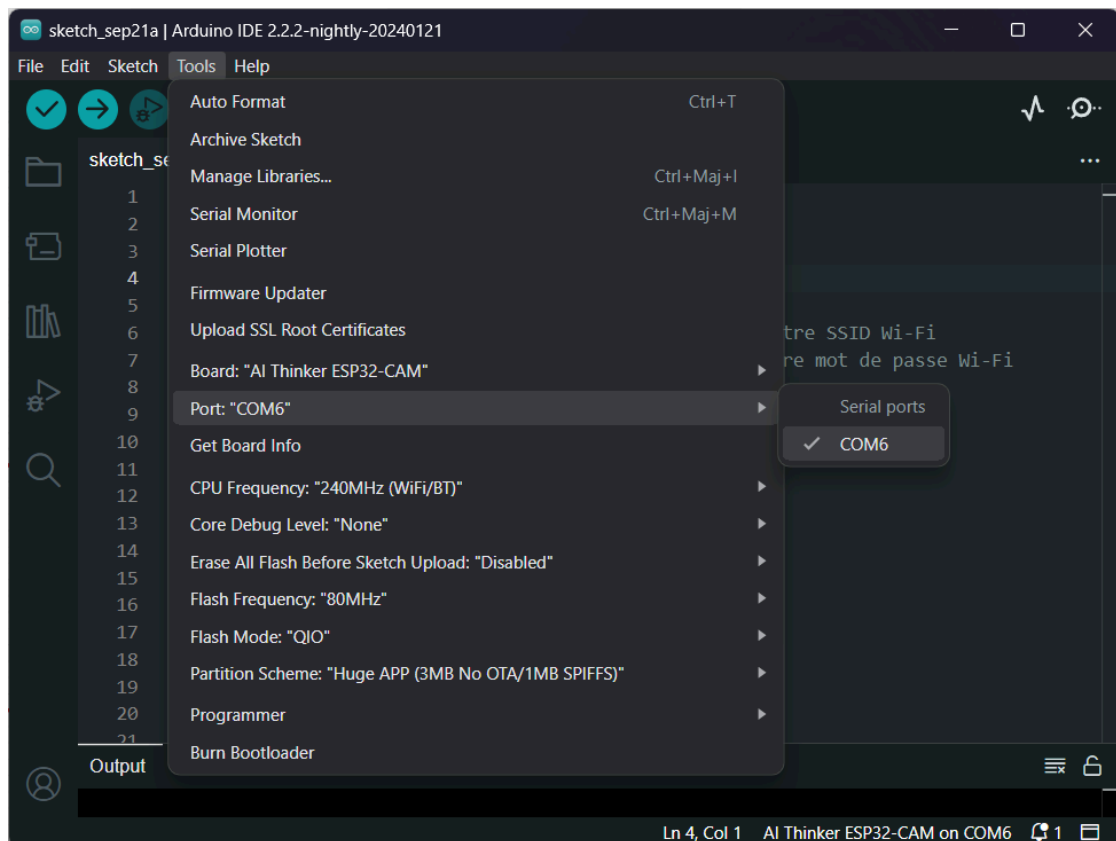


Figure III.9. Sélection du port

- **Transfert du programme**

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

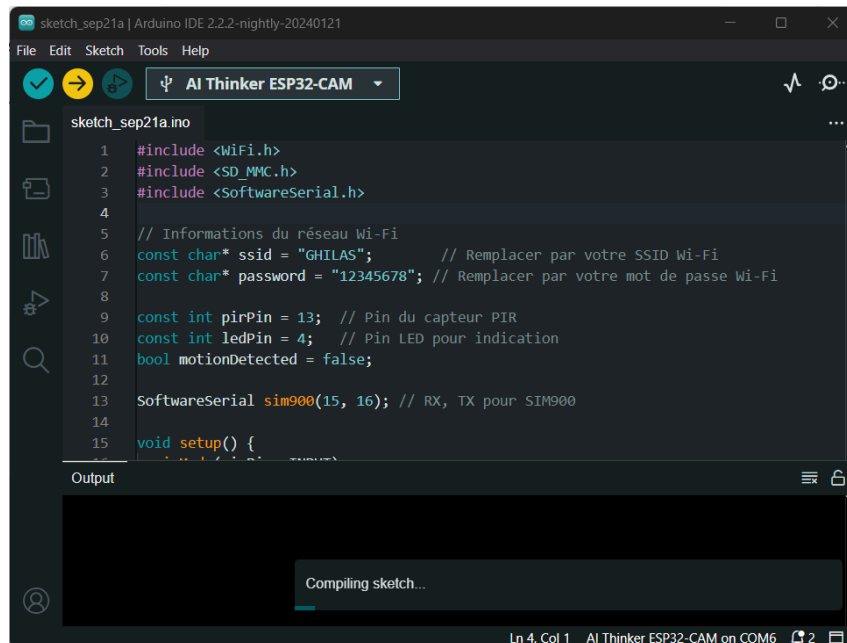


Figure III.10. *Transfert du programme*

III .2.6. Bibliothèques

Les bibliothèques ajoutent de nouvelles fonctions que l'on peut utiliser dans les programmes. Pour utiliser une bibliothèque, il suffit de la sélectionner dans le menu Sketch > Import Library (Programme > Importer une bibliothèque). Cela insérera une ou plusieurs instructions **#include** au début du programme et compilera la bibliothèque avec celui-ci.

Étant donné que les bibliothèques sont transférées sur la carte avec le programme, elles augmentent la quantité de mémoire utilisée. Si un programme n'a plus besoin d'une bibliothèque, il suffit de supprimer l'instruction **#include** correspondante au début du code. Une liste complète de bibliothèques est disponible dans la documentation du langage Arduino.

Certaines bibliothèques sont incluses dans le logiciel Arduino, tandis que d'autres peuvent être téléchargées à partir de différentes sources, comme par exemple : <http://www.github.com>.

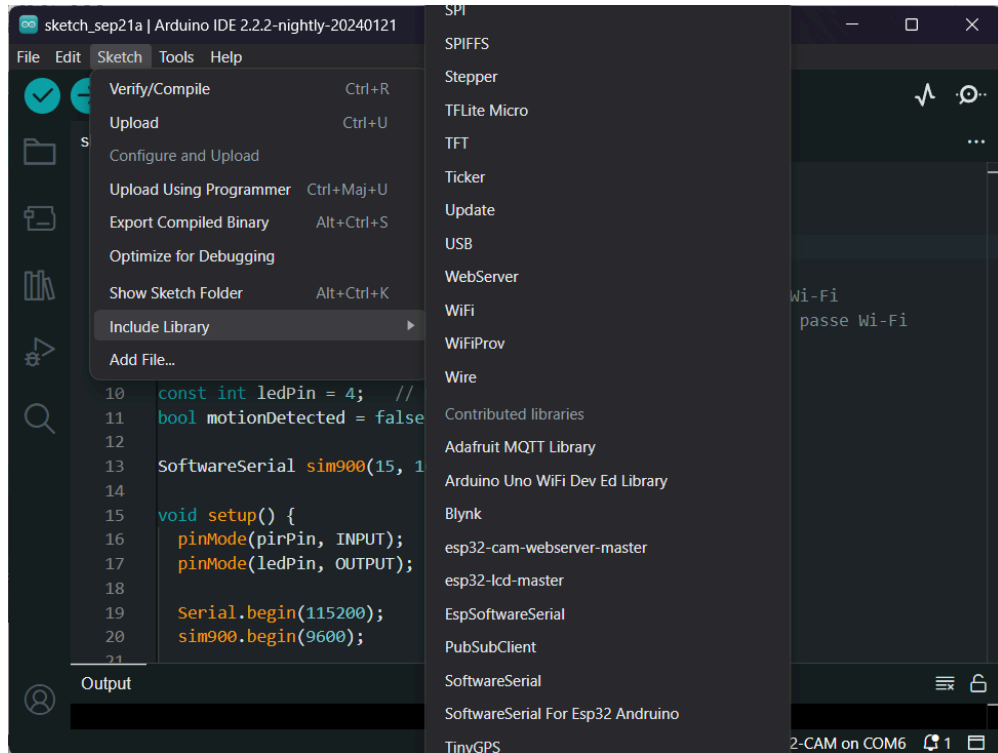


Figure III.11. Choix de librairie

III .2.7. Moniteur série

Un moniteur ou terminal est un écran et un clavier connectés à un microcontrôleur, permettant d'interagir avec l'utilisateur. Cette fonctionnalité facilite le débogage des programmes en affichant sur l'ordinateur l'état des variables, les résultats de calculs ou les conversions analogique-numérique. C'est un outil essentiel pour améliorer, tester et corriger les programmes.

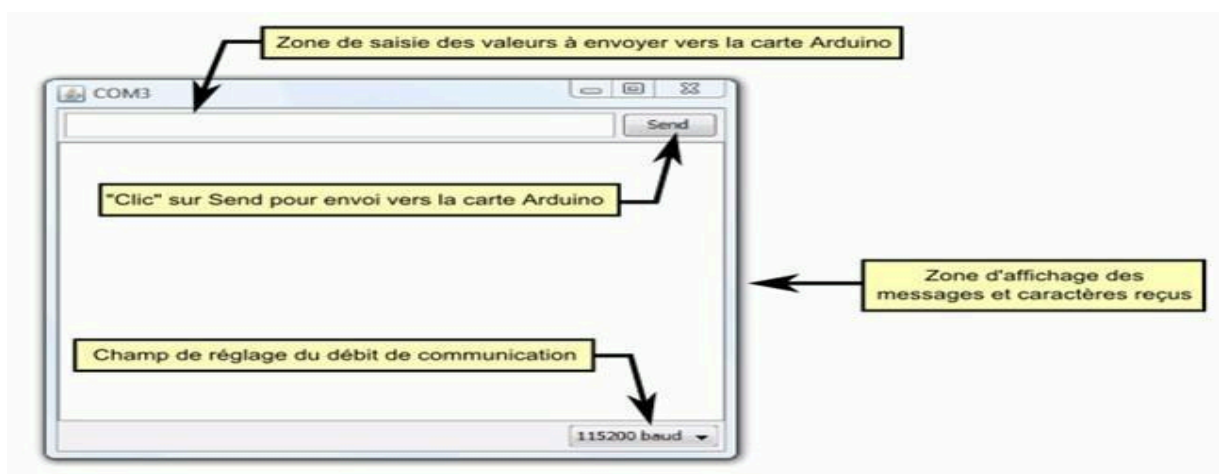


Figure III.12. Présentation du terminal série



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

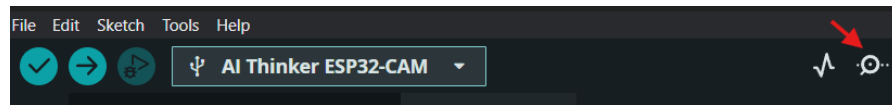


Figure III.13. Ouverture de terminal série

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

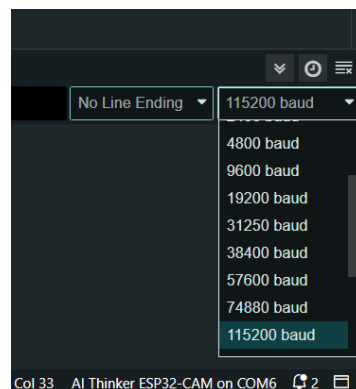


Figure III.14. Ajustage du débit de la communication série

III.3. Tests de l'application

Pour vérifier le bon fonctionnement de notre système, nous avons effectué plusieurs tests. Après la détection d'un mouvement par le capteur infrarouge, le module caméra de l'ESP32-CAM reçoit des instructions pour prendre des photos. Les images acquises sont enregistrées sur la carte SD. De plus, le module SIM900 envoie un SMS d'alerte contenant l'adresse IP du flux vidéo.



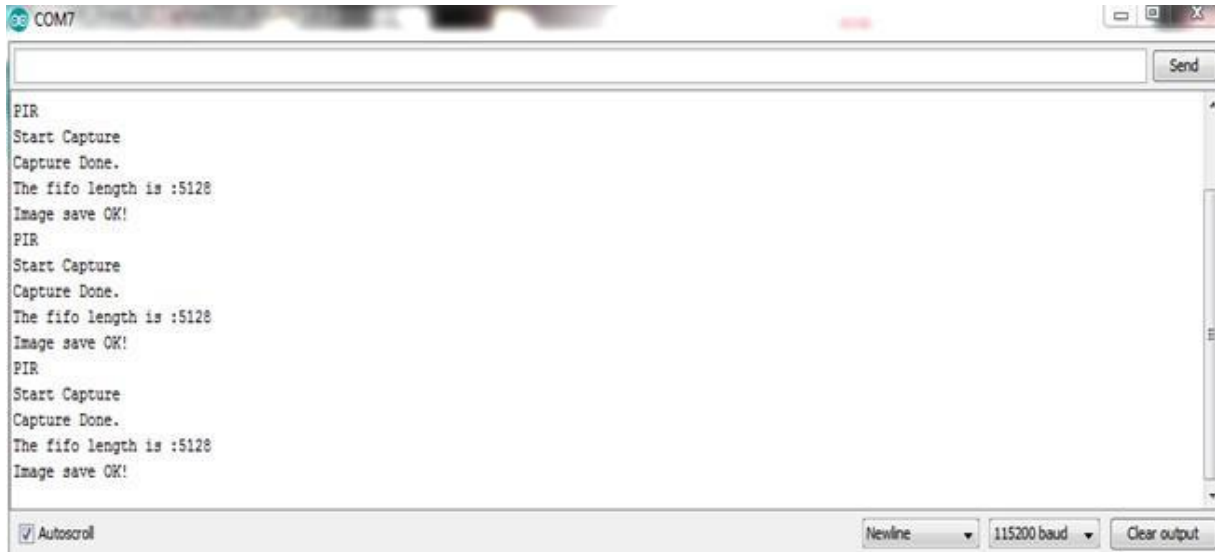


Figure III.15. Fenêtre de logiciel lors de détection de mouvement

L'sms reçu par le module GSM SIM900



Figure III.16. L'SMS reçu par le module GSM SIM900



III .4. Serveur Web

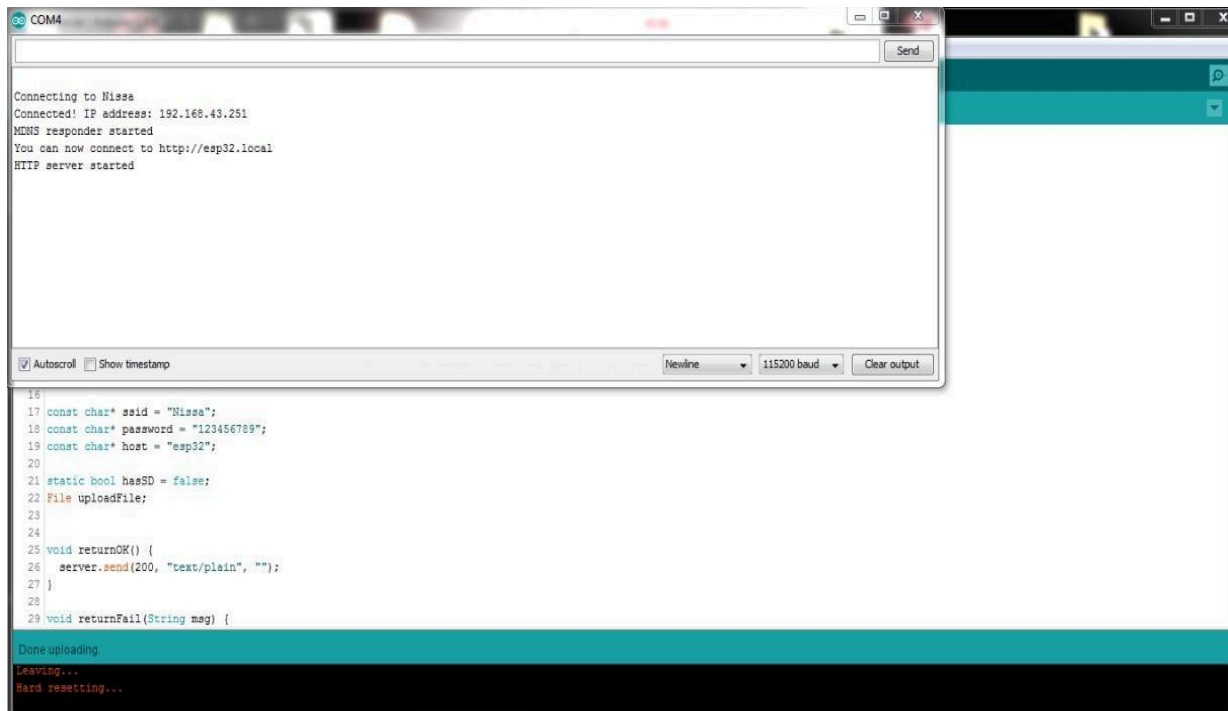
Un serveur web a été mis en place pour héberger les images acquises. Ce serveur permet aux clients d'accéder à des pages web, c'est-à-dire des fichiers au format HTML, à partir d'un navigateur installé sur un ordinateur distant. Un serveur web peut interpréter les requêtes HTTP reçues sur le port associé au protocole HTTP (par défaut, le port 80) et fournir une réponse en utilisant ce même protocole.

Pour créer un serveur web autonome avec un ESP32-CAM, nous avons suivi les étapes suivantes :

- Installation de la carte ESP32 dans l'Arduino IDE
- Rédaction du code dans l'IDE pour créer le serveur web.
- Définition des informations d'identification du réseau : il est nécessaire de spécifier le SSID et le mot de passe pour y accéder.
- Téléchargement du 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 :



```
COM4
Connecting to Nissa
Connected! IP address: 192.168.43.251
MDNS responder started
You can now connect to http://esp32.local
HTTP server started

Autoscroll Show timestamp Newline 115200 baud Clear output

16
17 const char* ssid = "Nissa";
18 const char* password = "123456789";
19 const char* host = "esp32";
20
21 static bool hasSD = false;
22 File uploadFile;
23
24
25 void returnOK() {
26   server.send(200, "text/plain", "");
27 }
28
29 void returnFail(String msg) {
```

Done uploading
Leaving...
Hard resetting...

Figure III.17. 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 : 172.20.10.8



III .5. Discussion

La carte ESP32-CAM est relativement simple à utiliser. Elle prend en charge plusieurs modules de caméra, allant de 0,3 Mp à 5 Mp, comme l'OV2640, ce qui nous permet de prendre des photos et de les envoyer au serveur web via le module WiFi, offrant ainsi l'avantage d'une transmission à distance.

Les tests que nous avons réalisés montrent que notre système fonctionne bien. Cependant, dans le cas du serveur web, nous avons rencontré quelques bugs.

En effet, les images transmises vers le serveur via le module WiFi ne s'affichent pas toujours correctement. Ce problème est probablement dû à la quantité importante de données envoyées par le module WiFi.



Conclusion Générale

Conclusion générale

Conclusion

Dans ce mémoire, nous avons présenté le système de surveillance développé. Ce système repose sur l'utilisation de la carte ESP32-CAM, qui commande le système et effectue le traitement des données.

Pour réaliser notre système de surveillance, nous avons d'abord testé le bon fonctionnement des différents éléments ainsi que leur communication avec la carte centrale, l'ESP32-CAM. Ensuite, nous avons assemblé tous ces composants.

Les tests effectués démontrent le bon fonctionnement de notre réalisation. Cependant, dans le cas du serveur web, nous avons identifié quelques bugs. En effet, les images transmises vers ce serveur via le module WiFi ne s'affichent pas toujours correctement, probablement en raison du volume important de données envoyées.

Les résultats obtenus en matière de capture et de transmission sont satisfaisants, rendant notre système prêt à l'emploi. Néanmoins, il reste sujet à des améliorations.

Les perspectives de ce travail incluent l'utilisation d'un servomoteur permettant à la caméra de suivre une cible en mouvement. De plus, la mise en place de plusieurs modules de surveillance connectés au même serveur web permettra de surveiller plusieurs lieux simultanément, comme les différents bureaux d'une entreprise. Enfin, l'intégration d'un système d'alarme pour avertir les personnes concernées serait également une amélioration à envisager.

Bibliographie

- [1] Mouser Electronics. (n.d.). ESP32-S2 Specifications. Consulté de <https://www.mouser.com/new/espressif/espressif-esp32-s2/>
- [2] Landraut, S., & Weisslinger, H. (2014). Arduino : premier pas en informatique embarquée. Le blog d'Eskimon.
- [3] - Mon Club Elec. (2018). Référence Arduino. Consulté de http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.
- [4] - Tavernier, C. (2014). Arduino : Maîtrisez sa programmation et ses cartes d'interface (shields). Éditions Dunod.
- [5] Université de Tlemcen. (n.d.). Application de télésurveillance. Consulté de <http://dspace.univ-tlemcen.dz/bitstream/112/11010/1/Application-de-telesurveillance.pdf>.
- [6] Sensagent. (n.d.). Télésurveillance. Consulté de <http://dictionnaire.sensagent.leparisien.fr/t%C3%A9l%C3%A9surveillance/fr-fr/>.
- [7] Université du Littoral. (n.d.). Les capteurs. Consulté de http://gte.univ-littoral.fr/sections/documents/Les_capteurs.pdf?nocache=1289041293.82.
- [8] REF11. (n.d.). PIR. Consulté de http://ref11.r-e-f.org/pluginAppObj_501/PIR.pdf.
- [9] Référent Sûreté. (n.d.). Guide vidéo IP. Consulté de https://www.referentsurete.com/wa_files/guide_20video_20ip.pdf.
- [10] UCTronics. (n.d.). OV2640DS. Consulté de https://www.uctronics.com/download/cam_module/OV2640DS.pdf.
- [11] ESP32.net. (n.d.). ESP32. Consulté de www.esp32.net.
- [12] Random Nerd Tutorials. (n.d.). ESP32 Web Server Arduino IDE. Consulté de <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 basé sur l'utilisation de la carte ESP32-CAM combinée avec le module SIM900. L'objectif de ce système est de surveiller un lieu à distance en temps réel via une caméra déclenchée par un détecteur de mouvement. Les images acquises sont enregistrées sur une carte mémoire et simultanément transmises à un serveur web, accessible par des utilisateurs autorisés, tandis que le module SIM900 envoie un SMS d'alerte contenant l'adresse IP du flux vidéo.

Les tests effectués démontrent le bon fonctionnement du système. Cependant, certaines améliorations peuvent être envisagées, notamment l'utilisation d'un servomoteur pour permettre à la caméra de suivre la cible en mouvement. La mise en place de plusieurs modules de surveillance connectés à un même serveur web permettrait de surveiller plusieurs lieux simultanément, comme différents bureaux dans une entreprise. Enfin, l'intégration d'un système d'alarme serait une amélioration utile pour alerter les personnes concernées.

Mots clés : ESP32-CAM, SIM900, capteur PIR, télésurveillance, serveur web..