

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

UNIVERSITÉ MOULOUD MAMMARI DE TIZI-OUZOU



FACULTÉ DU GÉNIE ÉLECTRIQUE ET D' INFORMATIQUE
DÉPARTEMENT D'ÉLECTRONIQUE

**Mémoire de Fin d'Études
de MASTER ACADEMIQUE**

Domaine : Sciences et Technologies

Filière : Électronique

Spécialité : Instrumentation

Présenté par

Nabil BOUZERNA

Fodil GHEZALI

Thème

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

Mémoire soutenu publiquement le 30 / 06 / 2024 devant le jury composé de :

Mr Hakim ACHOUR

Président

Mr Mourad LAGHROUCHE

Encadreur

Mme Hayat HAMOUCHE

Examineur

Promotion : 2023 / 2024

Remerciements

Tout d'abord nous remercions Allah le tout puissant de nous donner le courage et la volonté pour mener à bien ce modeste travail.

Nos remerciements s'adressent particulièrement à Mr Laghrouche Mourad pour avoir joué pleinement son rôle de promoteur et encadreur en étant à nos côtés tout au long de l'étude de notre projet, ses conseils et orientations nous ont guidés jusqu'à l'aboutissement de ce travail.

Nos vifs remerciements vont également aux membres du jury, pour l'intérêt qu'ils ont apporté à notre travail en acceptant de l'examiner et de l'enrichir par leurs propositions.

Nos sincères remerciements s'adressent à nos parents, nos frères, nos sœurs, nos amis, ainsi qu'à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail,

Dédicace

À mes chers parents, qui ont été ma source de soutien, d'amour et de persévérance tout au long de ce voyage académique.

À mes amis incroyables, dont le rire, l'amitié et le soutien inconditionnel ont éclairé ces années d'études.

À notre professeur, dont l'enseignement a façonné ma compréhension et a été une source d'inspiration pour ce mémoire.

Ce mémoire est le fruit de notre collaboration, de vos encouragements et de votre influence précieuse. Merci pour tout ce que vous avez apporté à ce travail.

Avec une profonde gratitude.

Bouzerna Nabil

Dédicace

Je dédie ce mémoire :

À mes très chers parents, qui ont œuvré pour ma réussite, de par leur soutien et sacrifice.

À mes frères et ma sœur, qui me pousse à être meilleur.

À mon binôme, avec qui j'ai passé des moments inoubliables.

À mes amis et camarade, qui illumine mes journées.

À notre professeur, sans qui ce travail n'aurait jamais existé.

Ghezali Fodil

Table des matières

Introduction.....	1
Chapitre 1 : Généralités sur la télésurveillance	4
1.1 Préambule.....	5
1.2 Définition de la télésurveillance :.....	5
1.3 Histoire de la télésurveillance :.....	5
1.4 Composantes d'un Système de Télésurveillance :.....	6
1.4.1 Caméras de Surveillance :	6
1.4.1.1 Caméras Analogiques :.....	7
1.4.1.2 Caméras IP (Internet Protocol)	7
1.4.1.3 Caméras PTZ (Pan-Tilt-Zoom) :.....	8
1.4.1.4 Caméras à Infrarouge (IR) :.....	8
1.4.1.5 Caméras Dômes :.....	9
1.4.1.6 Les critères de choix d'une caméra de Surveillance :.....	10
1.4.2 Capteurs et Détecteurs.....	11
1.4.2.1 Capteurs de Mouvement	11
1.4.2.2 Détecteurs de Fumée et de Chaleur :.....	12
1.4.2.3 Détecteurs de Gaz :.....	12
1.4.2.4 Capteurs d'Ouverture de Portes et Fenêtres :.....	13
1.4.2.5 Capteurs de Vibration et de Choc.....	13
1.4.2.6 Capteurs Environnementaux.....	13
1.4.3 Alarme de télésurveillance:	14
1.4.3.1 Sirènes	14
a) Sirènes intérieures :.....	14
b) Sirènes extérieures :	14
1.4.3.2 Avertisseurs Flash :.....	14
1.4.4 Les centrales d'alarme.....	15
1.4.4.1 Rôle des centrales d'alarme :.....	15
1.4.4.2 Centrale d'alarme filaire :.....	16
1.4.4.3 Centrale d'alarme sans fil :.....	16
1.4.5 Enregistreurs dans les Systèmes de Télésurveillance.....	16
1.4.5.1 Enregistreurs Vidéo Numériques (DVR).....	16
1.4.5.2 Enregistreurs Vidéo en Réseau (NVR).....	17
1.4.5.3 Comparaison et Choix.....	17

1.4.6 Transmetteurs dans les Systèmes de Télésurveillance.....	18
1.4.6.1 Transmetteurs IP.....	18
1.4.6.2 Transmetteurs GSM/GPRS.....	19
1.4.6.3 Transmetteurs Radio.....	19
1.4.6.4 Transmetteurs Satellite.....	19
1.4.7 Stations de surveillance :.....	20
1.4.7.1 Infrastructure Physique.....	20
1.4.7.2 Personnel et Formation.....	21
1.4.7.3 Logiciels de Gestion.....	21
1.4.7.4 Intégration avec d'Autres Services.....	21
1.4.7.5 Sécurité Physique et Logique.....	21
1.5 Application de la télésurveillance	22
1.5.1 Résidentielle.....	22
1.5.2 Commerciale	22
1.5.3 Industrielle.....	22
1.5.4 Urbaine.....	23
1.6 Avantages et Inconvénients de la télésurveillance :.....	23
1.6.1 Avantages :.....	23
1.6.2 Inconvénients :.....	23
1.7 Discussion	24
Chapitre 2 : Conception du système	25
2.1 Préambule :.....	26
2.2 Description du projet:	26
2.3 Description matérielle :.....	27
2.3.1 La carte ESP32-CAM :.....	27
2.3.1.1 Présentation de la carte ESP32-CAM :.....	27
2.3.1.2 Caractéristiques techniques de l'ESP32-CAM.....	28
2.3.1.3 Brochage de la carte ESP32-CAM.....	29
2.3.1.4 Le module MB pour l'ESP32-CAM:	30
2.3.2 La carte ESP32:.....	30
2.3.2.1 Présentation de la carte ESP32 :.....	30
2.3.2.2 Caractéristique de la carte ESP 32 :.....	31
2.3.2.3 brochage de la carte ESP32 :.....	31
2.3.3 Module RFID :.....	32

2.3.3.1 Badges RFID :	32
2.3.3.2 Le lecteur RFID RC522	33
2.3.3.3 Caractéristiques du lecteur RFID RC522 :	33
2.3.3.4 Le Brochage du lecteur RFID RC522:	33
2.3.4 Capteur de mouvement PIR HC-SR501 :	34
2.3.4.1 Caractéristique de PIR:	35
2.3.5 Capteur de gaz MQ-135 :	35
2.3.5.1 Caractéristique du Capteur MQ-135:	36
2.3.6 Le servomoteur sg90:	37
2.3.6.1 Caractéristiques du Servomoteur SG90 :	37
2.3.7 Buzzer :	38
2.3.8 Les Leds :	38
2.4 Description logicielle :	39
2.4.1 Logiciel Arduino IDE:	39
2.4.1.1 Description du logiciel Arduino IDE:	39
2.4.2 Présentation de l'application Telegram :	43
2.4.2.1 Caractéristiques Principales de Telegram :	43
2.4.2.2 Les bots telegram:	44
2.4.3 Le logiciel Fritzing :	44
2.5 Discussion :	45
Chapitre 3 : Réalisation pratique	46
3.1 Préambule:	47
3.2 Réalisation Matérielle :	47
3.2.1 Partie ESP32-CAM:	47
3.2.1.1 les connexions de l'ESP32-CAM:	48
3.2.2 Partie ESP32:	50
3.2.2.1 Contrôle d'Accès via RFID :	50
3.2.2.2 Détection de Gaz :	52
3.2.2.3 Détection de Mouvement :	52
3.2.2.4 Interaction et Commandes via Telegram :	53
3.3 Configuration logicielle :	55
3.3.1 Configuration des cartes ESP32 dans l'Arduino IDE :	55
3.3.2 Configuration de l'application Telegram :	58
3.3.2.1 Créer un compte Telegram :	58

3.3.2.2 Créer un bot Telegram:.....	59
3.3.2.3 Obtenez l'identifiant d'utilisateur Telegram:.....	61
3.4 Organigramme du système :.....	63
3.5 Teste du projet:.....	64
3.6 Discussion:.....	69
Conclusion.....	70
Bibliographie :.....	72
Annexe :.....	73
Résumé	89
Mots-clés.....	89

Liste des figures :

Figure 1: Caméras Analogiques.....	7
Figure 2: Caméras IP.....	8
Figure 3: Caméras PTZ	8
Figure 4: Caméras à Infrarouge	9
Figure 5: Caméras Dômes	10
Figure 6: capteurs détectent un mouvement.....	11
Figure 7: Détecteurs de Fumée	12
Figure 8: Capteurs d'Ouverture.....	13
Figure 9: Exemple d'une Centrale d'alarme	15
Figure 10: un DVR et un NVR.....	18
Figure 11: Stations de surveillance	20
Figure 12: schéma synoptique du système	27
Figure 13: Carte ESP32-CAM.....	28
Figure 14: Broches de la Carte ESP-32-CAM.....	29
Figure 15: Le module MB pour l'ESP32-CAM.....	30
Figure 16: carte ESP32.....	31
Figure 17: brochage de la carte ESP32	32
Figure 18: Badges RFID	32
Figure 19: Le Brochage du lecteur RFID RC522	34
Figure 20: Le module capteur HC-SR501 PIR.....	35
Figure 21: Capteur MQ-135	36
Figure 22: servomoteur sg90.....	37
Figure 23: buzzer	38
Figure 24: Diode électroluminescente et son symbole	39
Figure 25: Tableau représentant le voltage de chaque led par rapport à sa couleur.....	39
Figure 26: L'interface de l'IDE Arduino	40
Figure 27: barre d'icône de l'IDE	41
Figure 28: zone d'édition de l'IDE.....	42
Figure 29: zone de détection d'erreur de l'IDE	42
Figure 30: zone barre menu de l'IDE	43
Figure 31: logo de l'application Telegram	43
Figure 32: logiciel Fritzing.....	45
Figure 33: Branchement des Pins du capteur de mouvement.....	50

Figure 34: Connexion du contrôle d'Accès	51
Figure 35: Connexion pour détection de Gaz.....	52
Figure 36: Connexion pour détection de de Mouvement PIR.....	53
Figure 37: Connexion des leds et du buzzer à l'ESP32.....	54
Figure 38: schéma général du montage de notre système.	55
Figure 39: copie de l'URL dans le fichier préférence	56
Figure 40: installation de l'esp32 dans le gestionnaire de carte.....	56
Figure 41: choix du type de carte pour ESP32-CAM.....	57
Figure 42: configuration des paramètre de la carte ESP32-CAM	57
Figure 43: choix du type de carte pour esp32.....	58
Figure 44: interface de l'application telegram pour débiter l'ouverture d'un compte	59
Figure 45: recherche de @BotFather sur telegram	59
Figure 46: démarrage de la création du bot.....	60
Figure 47: nommer le bot.....	61
Figure 48: message de réception du jeton du bot	61
Figure 49: recherche de IDBot sur telegram	62
Figure 50: réception de l'identifiant	62
Figure 51: capture d'image du bot telegram	64
Figure 52: capture d'image du port serie	64
Figure 53: capture d'image du bot telegram	64
Figure 54: capture d'image du port serie	64
Figure 55: fichier ou sont sauvegarder les photos	65
Figure 56: photo du bon badge et du sg90 qui se tourne	65
Figure 57: capture d'image du port serie	65
Figure 58: photo de la led verte qui s'allume.....	65
Figure 59: photo du mauvais badge et du sg90 qui se tourne	66
Figure 60: capture d'image du port serie	66
Figure 61: photo de la led rouge qui s'allume.....	66
Figure 62: capture d'image du bot telegram	66
Figure 63: photo de simulation de fuite de gaz	67
Figure 64: capture d'image du bot telegram	67
Figure 65: capture d'image du port serie	67
Figure 66: capture d'image du port serie	67
Figure 67: capture d'image du bot telegram	67

Figure 68: capture d'image du bot telegram	68
Figure 69: photo de la led rouge allumer	68
Figure 70: photo de la led orange allumer.....	68
Figure 71: capture d'image du bot telegram	69

Liste des tableaux :

Tableau 1: Branchement des Pins de la Caméra	48
Tableau 2: Branchement des Pins de la Carte SD.....	49

Introduction

Introduction

La technologie occupe une place de plus en plus importante dans notre vie quotidienne, apportant confort, aisance et rapidité. Par exemple, une personne à mobilité réduite peut faire ses courses sur Internet. De l'automobile au smartphone, qui nous accompagne constamment, la technologie investit désormais nos maisons où nous passons une grande partie de notre temps. L'habitat du futur doit impérativement s'adapter aux besoins évolutifs de l'homme. Une maison intelligente se distingue par sa capacité à intégrer les technologies de l'électronique, de l'automatique, de l'informatique et des télécommunications pour améliorer le confort, la sécurité, la communication et la gestion de l'énergie.

L'être humain cherche toujours à mettre en place un système de sécurité et de surveillance fiable pour protéger ses biens immobiliers et locaux contre les intrusions et prévenir les vols. Avec l'avènement des systèmes embarqués, les outils de communication à distance sont devenus essentiels, en particulier les nouveaux instruments pour la sécurité humaine. Depuis longtemps, les serrures ont évolué pour assurer la sécurité, et aujourd'hui, des systèmes de verrouillage intelligents permettent de gérer l'accès aux endroits privés de manière pratique. Cette évolution technologique a permis le développement de systèmes de sécurité de plus en plus performants, grâce notamment à l'utilisation des applications de l'électronique moderne, tant du point de vue de la communication entre les périphériques de commande (Bluetooth, WiFi, infrarouge) que des composants (microcontrôleurs programmables, cartes comme l'Arduino).

Parmi les méthodes de sécurité, les systèmes de télésurveillance se distinguent par leur capacité à observer les mouvements et comportements des personnes dans une zone surveillée. Ces systèmes sont devenus indispensables pour sécuriser les zones sensibles ou à risque. De nombreuses améliorations ont été apportées pour offrir des solutions performantes et de haute sécurité. Ces dernières années, l'intégration de l'intelligence artificielle et de l'apprentissage automatique à révolutionner la télé surveillance. les systèmes modernes peuvent désormais détecter automatiquement les comportements suspects, identifier les objets et les personnes, et même prédire les incidents potentiels en utilisant des algorithmes avancés.

C'est dans cette thématique que s'inscrit notre projet. Il concerne l'étude et la réalisation d'un système de télésurveillance basé sur d'un microcontrôleurs qui se charge de mettre en œuvre diverses fonctionnalités telles que la détection de mouvement, la capture de photos, la surveillance de gaz, le contrôle d'accès par RFID, l'interaction à distance, ainsi que le stockage et l'accès aux données.

Introduction

Pour ce faire, nous avons réparti ce mémoire en trois chapitres :

1. **Chapitre 1** : Nous présenterons les généralités de la télésurveillance : définition, éléments de télésurveillance, application...
2. **Chapitre 2** : Nous décrivons la conception du système ainsi que l'ensemble des composants et logiciels utilisés pour développer notre projet.
3. **Chapitre 3** : Nous détaillerons les étapes de réalisation pratique de notre système, ainsi que la configuration de la partie logicielle.

Ce projet vise à démontrer la faisabilité et l'efficacité de l'utilisation d'un ESP32 comme centre de contrôle pour un système de sécurité domestique intelligent, en combinant plusieurs technologies pour offrir une solution innovante et efficace en matière de sécurité domestique.

Chapitre 1 :

Généralités sur la

télé-surveillance

Chapitre 1 : Généralités sur la télésurveillance

1.1 Préambule :

Le besoin de surveiller la vie humaine et les biens est primordial dans la vie quotidienne, pour cela, l'être humain a opté depuis plusieurs années à la création de système d'alarme lui aidant à protéger sa vie et ses biens. Un système d'alarme représente un moyen fiable pour protéger la vie des personnes et leurs biens. Le système d'alarme de base est composé généralement d'un panneau de commande et des capteurs placés tout autour pour détecter toute anomalie, une fois l'alarme déclenchée, le système active une sirène et envoi une alerte de sécurité vers un PC au centre de surveillance. Le simple fait de mettre une sirène sur la façade dissuade la plupart des tentatives d'intrusions. Dans les paragraphes qui vont suivre nous introduisons les différents types d'alarmes, leurs domaines d'applications ainsi que les technologies utilisées

1.2 Définition de la télésurveillance :

La télésurveillance est un système de surveillance à distance utilisant des technologies telles que la vidéo, les capteurs, les alarmes et les communications pour surveiller et contrôler des lieux, des biens ou des personnes. Ce système permet à des opérateurs de surveiller activement les événements en temps réel et de réagir en cas d'incident ou d'anomalie détecté. La télésurveillance trouve des applications variées telles que la sécurité résidentielle et commerciale, la surveillance industrielle, le suivi médical à distance et la protection de l'environnement. Son objectif principal est d'offrir une surveillance efficace et proactive, améliorant ainsi la sécurité, l'efficacité opérationnelle et la réactivité aux événements.

1.3 Histoire de la télésurveillance :

L'histoire de la télésurveillance est un voyage fascinant à travers les évolutions technologiques et les besoins changeants en matière de sécurité et de surveillance. Les premières formes de surveillance à distance remontent à l'Antiquité, où des systèmes mécaniques simples, comme les tours de guet, étaient utilisés pour surveiller les environs et détecter les menaces potentielles. Ces systèmes ont été progressivement améliorés au fil des siècles, bien que leur portée et leur efficacité étaient limitées.

Avec le développement des télécommunications au début du 20e siècle, l'homme a pu surveiller des lieux éloignés à l'aide de systèmes de télégraphie et de téléphonie, permettant la transmission d'informations sur de longues distances. Cela a ouvert la voie à une surveillance plus sophistiquée sur de plus grandes distances, bien que les premiers systèmes étaient souvent lents et peu fiables.

Chapitre 1 : Généralités sur la télésurveillance

Les avancées significatives dans la télésurveillance ont débuté avec l'invention des caméras de surveillance dans les années 1940 et 1950. Ces premières caméras ont permis de capturer des images en temps réel, facilitant ainsi une surveillance visuelle à distance. Initialement utilisées principalement dans les environnements industriels et militaires, ces caméras ont ensuite été adoptées par les entreprises et les organismes gouvernementaux pour la sécurité des installations et la surveillance publique.

Dans les années 1960 et 1970, les systèmes d'alarme ont été intégrés aux systèmes de surveillance à distance, permettant une détection automatique des intrusions et des incidents. Cela a considérablement amélioré la capacité des opérateurs de surveillance à réagir rapidement aux menaces détectées.

À partir des années 1980, la numérisation des systèmes de surveillance a commencé à remplacer les technologies analogiques, offrant une meilleure qualité d'image ainsi que des capacités de stockage et de traitement des données accrues. L'avènement d'Internet dans les années 1990 a ouvert la voie à la surveillance à distance via des réseaux informatiques, marquant le début de la surveillance vidéo sur IP.

Ces dernières années, l'intégration de l'intelligence artificielle et de l'apprentissage automatique a révolutionné la télésurveillance. Les systèmes modernes peuvent désormais détecter automatiquement les comportements suspects, identifier les objets et les personnes, voire prédire les incidents potentiels grâce à des algorithmes avancés.

1.4 Composantes d'un Système de Télésurveillance :

Un système de télésurveillance se compose de plusieurs éléments essentiels qui interagissent pour fournir une surveillance efficace et sécurisée à distance :

1.4.1 Caméras de Surveillance :

Les caméras de surveillance sont des dispositifs essentiels dans tout système de télésurveillance. Elles capturent des images et des vidéos des zones surveillées et les transmettent au moniteur ou à l'enregistreur. Elles jouent un rôle crucial dans la prévention et la détection des incidents.

Un système de télésurveillance peut intégrer différents types de caméras pour une efficacité optimale :

Chapitre 1 : Généralités sur la télésurveillance

1.4.1.1 Caméras Analogiques :

- ⑩ **Description** : Utilisent des signaux analogiques pour transmettre les images aux enregistreurs vidéo.
- ⑩ **Avantages** : Coût initial relativement bas, simplicité d'installation.
- ⑩ **Inconvénients** : Résolution inférieure et qualité d'image moindre par rapport aux caméras numériques, dépendance à des câbles coaxiaux pour la transmission des données.
- ⑩ **Applications** : Utilisées dans les systèmes de surveillance plus anciens ou lorsque le budget est limité.



Figure 1 Caméras Analogiques

1.4.1.2 Caméras IP (Internet Protocol) :

- ⑩ **Description** : Transmettent les vidéos via des réseaux IP, offrant une meilleure qualité d'image et des fonctionnalités avancées.
- ⑩ **Avantages** : Haute résolution, capacités de zoom numérique, facilité d'intégration avec d'autres systèmes de sécurité, accès à distance via Internet.
- ⑩ **Inconvénients** : Coût plus élevé, besoin de bande passante suffisante pour la transmission des vidéos en haute qualité.
- ⑩ **Applications** : Idéales pour les systèmes de surveillance modernes, y compris les entreprises, les résidences haut de gamme et les espaces publics.

Chapitre 1 : Généralités sur la télésurveillance



Figure 2: Caméras IP

1.4.1.3 Caméras PTZ (Pan-Tilt-Zoom) :

- ⑩ **Description** : Permettent de contrôler l'orientation (panoramique et inclinaison) ainsi que le zoom de la caméra à distance.
- ⑩ **Avantages** : Flexibilité de surveillance, couverture étendue d'une grande zone avec une seule caméra, zoom optique pour des détails précis.
- ⑩ **Inconvénients** : Coût élevé, nécessitent une gestion active pour tirer parti de toutes les fonctionnalités.
- ⑩ **Applications** : Utilisées dans les environnements nécessitant une surveillance dynamique comme les parkings, les stades et les grands espaces publics.



Figure 3: Caméras PTZ

1.4.1.4 Caméras à Infrarouge (IR) :

- ⑩ **Description** : Équipées de LED infrarouges pour capturer des images en conditions de faible luminosité ou de nuit.

Chapitre 1 : Généralités sur la télésurveillance

- ⑩ **Avantages** : Capacité de vision nocturne, fournissent des images claires dans l'obscurité totale.
- ⑩ **Inconvénients** : Portée limitée des LED infrarouges, images généralement en noir et blanc en mode infrarouge.
- ⑩ **Applications** : Utilisées pour la surveillance nocturne dans les résidences, les entrepôts et les périmètres extérieurs.



Figure 4: Caméras à Infrarouge

1.4.1.5 Caméras Dômes :

- ⑩ **Description** : Caméras enfermées dans un boîtier en forme de dôme, souvent discrètes et résistantes aux intempéries.
- ⑩ **Avantages** : Discrétion et dissuasion, difficile de déterminer la direction de la caméra, résistance aux actes de vandalisme.
- ⑩ **Inconvénients** : Installation spécifique selon les modèles (intérieur ou extérieur).
- ⑩ **Applications** : Utilisées dans les environnements commerciaux, les bureaux et les espaces publics intérieurs.

Chapitre 1 : Généralités sur la télésurveillance



Figure 5: Caméras Dômes

1.4.1.6 Les critères de choix d'une caméra de Surveillance

Le choix d'une caméra de télésurveillance dépend en grande partie des besoins et exigences, celles-ci se distinguent par leurs performances:

- ⑩ **Résolution** : Plus la résolution est élevée, plus les détails dans l'image sont clairs. Les caméras modernes offrent des résolutions allant de 720p à 4K.
- ⑩ **Angle de Vision** : Détermine la largeur de la zone que la caméra peut couvrir. Les caméras à large angle de vision peuvent surveiller de plus grandes zones.
- ⑩ **Luminosité** : Elle est exprimée en « lux ». Certaines caméras peuvent filmer aussi bien dans l'obscurité (0 lux) qu'avec beaucoup de luminosité.
- ⑩ **Capacités de Zoom** : Le zoom optique utilise des lentilles pour agrandir l'image sans perte de qualité, tandis que le zoom numérique agrandit l'image en utilisant des techniques logicielles, ce qui peut entraîner une perte de qualité.
- ⑩ **Résistance aux Intempéries** : Les caméras destinées à une utilisation en extérieur doivent être résistantes aux conditions météorologiques (pluie, neige, poussière). Les normes IP (Ingress Protection) indiquent le niveau de protection.
- ⑩ **Audio** : Certaines caméras sont équipées de microphones et de haut-parleurs pour capturer et diffuser l'audio, permettant une communication bidirectionnelle.
- ⑩ **Stockage** : Pour le stockage interne, on utilise des cartes SD intégrées. Pour le stockage externe, on utilise l'enregistrement sur DVR/NVR ou le stockage cloud pour un accès à distance et une sauvegarde sécurisée.

Chapitre 1 : Généralités sur la télésurveillance

- ⑩ **Alimentation** : PoE (Power over Ethernet) permet de transmettre l'alimentation et les données via un seul câble Ethernet. Les caméras sans fil utilisent des batteries rechargeables pour une installation flexible.

1.4.2 Capteurs et Détecteurs

Les capteurs et détecteurs sont des composants essentiels dans un système de télésurveillance. Ils jouent un rôle crucial dans la détection précoce des incidents et des situations anormales, permettant ainsi une réponse rapide et efficace. Voici un détail des principaux types de capteurs et détecteurs utilisés en télésurveillance :

1.4.2.1 Capteurs de Mouvement



Figure 6: capteurs détectent un mouvement

Les capteurs de mouvement détectent tout mouvement dans une zone surveillée. Ils sont utilisés pour déclencher des alertes et des enregistrements vidéo lorsque du mouvement est détecté.

- ⑩ **Technologie infrarouge** : Ces capteurs détectent les changements de chaleur infrarouge dans leur champ de vision, typiquement émis par les corps humains.
- ⑩ **Micro-ondes** : Utilisent des ondes radio pour détecter les mouvements en mesurant les variations dans le reflet des ondes.
- ⑩ **Ultrasons** : Utilisent des ondes sonores à haute fréquence pour détecter les mouvements par le changement des ondes réfléchies.

Chapitre 1 : Généralités sur la télésurveillance

1.4.2.2 Détecteurs de Fumée et de Chaleur :

Ces détecteurs sont essentiels pour la prévention des incendies en détectant les signes précoces de feu.

- ⑩ **Détecteurs de fumée optiques** : Utilisent un faisceau lumineux pour détecter la présence de particules de fumée dans l'air. Lorsque la fumée interrompt le faisceau, l'alarme est déclenchée.
- ⑩ **Détecteurs de fumée ioniques** : Utilisent une petite quantité de matériau radioactif pour ioniser l'air dans une chambre détectrice. La présence de fumée perturbe cette ionisation, déclenchant l'alarme.
- ⑩ **Détecteurs de chaleur** : Déclenchent une alarme lorsqu'une certaine température est atteinte ou lorsqu'il y a une montée rapide de la température, indiquant un possible incendie.



Figure 7: Détecteurs de Fumée

1.4.2.3 Détecteurs de Gaz :

Ces détecteurs sont conçus pour identifier la présence de gaz dangereux dans l'environnement, protégeant ainsi contre les intoxications et les explosions.

- ⑩ **Détecteurs de monoxyde de carbone (CO)** : Détectent la présence de monoxyde de carbone, un gaz incolore et inodore qui peut être mortel en concentrations élevées.
- ⑩ **Détecteurs de gaz naturel et de propane** : Utilisés principalement dans les environnements domestiques et industriels pour détecter les fuites de gaz combustibles.
- ⑩ **Détecteurs de gaz toxiques** : Capables de détecter une variété de gaz dangereux comme l'ammoniac, le chlore et les composés organiques volatils (COV).

Chapitre 1 : Généralités sur la télésurveillance

1.4.2.4 Capteurs d'Ouverture de Portes et Fenêtres :

Ces capteurs détectent lorsque des portes ou des fenêtres sont ouvertes, permettant ainsi de surveiller les accès non autorisés.

- ⑩ **Capteurs magnétiques** : Composés de deux parties, l'une fixée à la porte ou la fenêtre, et l'autre au cadre. Lorsque la porte ou la fenêtre est ouverte, les deux parties se séparent et déclenchent l'alarme.
- ⑩ **Capteurs à contact reed** : Utilisent un contact reed qui se ferme en présence d'un champ magnétique (quand la porte ou la fenêtre est fermée) et s'ouvre en l'absence de ce champ.



Figure 8: Capteurs d'Ouverture

1.4.2.5 Capteurs de Vibration et de Choc

Ces capteurs sont utilisés pour détecter les tentatives de forçage ou de bris de vitres.

- ⑩ **Capteurs de vibration** : Détectent les vibrations causées par des tentatives d'effraction.
- ⑩ **Capteurs de choc** : Conçus pour détecter les impacts violents ou les vibrations de haute intensité, comme celles causées par un bris de vitre.

1.4.2.6 Capteurs Environnementaux

Ces capteurs surveillent les conditions environnementales pour prévenir les dommages matériels et assurer la sécurité.

- ⑩ **Capteurs d'inondation** : Détectent la présence d'eau dans des zones sensibles, comme les sous-sols, pour prévenir les dégâts des eaux.
- ⑩ **Capteurs de température et d'humidité** : Surveillent les variations de température et d'humidité, importants pour la conservation de certains matériaux ou pour prévenir la moisissure.

Chapitre 1 : Généralités sur la télésurveillance

1.4.3 Alarme de télésurveillance :

Les systèmes d'alarme sont une composante clé des solutions de télésurveillance, conçus pour détecter et signaler rapidement les intrusions, les incendies, les fuites de gaz et autres situations d'urgence. Ils contribuent à la sécurité des biens et des personnes en alertant les occupants et les autorités compétentes en cas de danger.

1.4.3.1 Sirènes

Il existe deux types de sirènes : intérieures et extérieures.

a) Sirènes intérieures :

- ⑩ A pour fonction de surprendre et repousser l'intrus avec un signal sonore strident insupportable.
- ⑩ Doit être équipée d'une batterie interne maintenue en charge par l'alimentation de la centrale pour rendre la sirène autonome.
- ⑩ En cas de coupure des fils, la sirène doit se déclencher.
- ⑩ La puissance sonore doit être importante et le signal très strident.
- ⑩ Son emplacement doit être difficilement accessible tout en permettant une bonne diffusion du signal dans l'ensemble de l'habitation .

b) Sirènes extérieures :

- ⑩ Conçues pour avoir une portée très importante et avertir le voisinage.
- ⑩ Doivent être fixées en hauteur pour éviter l'accès facile.
- ⑩ Doivent être autonomes, équipées de boîtiers étanches et anti-effraction, protégées contre les mousses expansives, et temporisées pour chaque déclenchement à 3 minutes maximum.
- ⑩ Dotées d'une autoprotection contre l'ouverture du boîtier.
- ⑩ Doivent être audibles au-delà du voisinage.

1.4.3.2 Avertisseurs Flash :

- ⑩ Émettent un signal lumineux, placés à l'extérieur et en hauteur.
- ⑩ Assurent la dissuasion de l'intrus et l'alerte du voisinage avec un puissant flash clignotant permettant la localisation de l'alerte.

Chapitre 1 : Généralités sur la télésurveillance

1.4.4 Les centrales d'alarme

Les centrales d'alarme représentent le cœur vital des systèmes de sécurité, jouant un rôle crucial dans la surveillance et la protection des biens et des personnes. Elles agissent comme des émetteurs/récepteurs radio centralisant tous les éléments du système de sécurité. Dotées d'une alarme sonore puissante, elles sont conçues pour déclencher des alertes en cas d'intrusion ou d'incident détecté. Ces centrales sont généralement protégées par une coque en acier robuste et équipées de technologies avancées telles qu'un système radio à double fréquence pour éviter le brouillage et des dispositifs anti-arrachement.

Leur fonctionnement repose sur une commande via une télécommande dédiée ou un clavier mural installé de manière indépendante. Pour garantir leur efficacité, elles sont positionnées en hauteur et placées stratégiquement dans la maison, inaccessible directement pour prévenir toute tentative de neutralisation. Cette position centrale permet à la centrale d'alarme d'être à une distance équidistante des différents capteurs du système, tels que les détecteurs d'ouverture, de bris de surface vitrée et de mouvement, assurant ainsi une couverture optimale de la zone surveillée.



Figure 9: Exemple d'une Centrale d'alarme

1.4.4.1 Rôle des centrales d'alarme :

- ⑩ Traite les informations des détecteurs.
- ⑩ Alimente les détecteurs.
- ⑩ Surveille le bon fonctionnement des détecteurs et des différentes liaisons.
- ⑩ Donne l'ordre aux éléments de signalisation (sirènes, transmetteur téléphonique) de donner l'alerte.

Chapitre 1 : Généralités sur la télésurveillance

1.4.4.2 Centrale d'alarme filaire :

- ⑩ Le plus fiable des systèmes mais plus difficile à installer.
- ⑩ Reliée par des fils entre la centrale et les détecteurs.
- ⑩ Idéale pour les habitations à plusieurs étages et en construction.
- ⑩ Avantages : Fiabilité, moins chère à qualité égale, très efficace pour les grandes surfaces.
- ⑩ Inconvénients : Installation complexe, système définitif non déplaçable, coût d'installation élevé, inefficacité si mal installée.

1.4.4.3 Centrale d'alarme sans fil :

- ⑩ Communique par ondes radios avec les détecteurs.
- ⑩ Transmetteur GSM / 3G pour connexion permanente à un centre de télésurveillance et contrôle à distance via une application.
- ⑩ Avantages : Facilité d'installation, esthétique, discret, modulable, évolutif, réutilisable, déménagement facile.
- ⑩ Inconvénients : Moins fiable que filaire, possibilité d'interférence, plus vulnérable aux brouillages, plus chère, portée limitée des ondes radios.

1.4.5 Enregistreurs dans les Systèmes de Télésurveillance

Les enregistreurs sont des composants essentiels des systèmes de télésurveillance, responsables de la capture, du stockage et de la gestion des données vidéo et parfois audio provenant des caméras de surveillance. Ils jouent un rôle crucial dans la capacité des systèmes à fournir une surveillance continue et à permettre la consultation des enregistrements pour des besoins de sécurité, de vérification ou d'enquête ultérieure. Voici les détails sur les principaux types d'enregistreurs utilisés dans les systèmes de télésurveillance :

1.4.5.1 Enregistreurs Vidéo Numériques (DVR)

Les DVR sont conçus pour fonctionner avec des caméras de surveillance analogiques. Voici leurs caractéristiques principales :

- ⑩ **Conversion analogique-numérique:** Ils convertissent les signaux vidéo analogiques provenant des caméras en formats numériques lisibles par les ordinateurs.

Chapitre 1 : Généralités sur la télésurveillance

- ⑩ **Stockage local:** Les enregistreurs DVR sont équipés de disques durs internes où les enregistrements vidéo sont stockés localement. La capacité de stockage peut varier en fonction du modèle et des besoins spécifiques du système.
- ⑩ **Interfaces de communication:** Ils peuvent inclure des ports USB pour le transfert de données et la sauvegarde, ainsi que des ports réseau pour la connectivité LAN.
- ⑩ **Logiciel de gestion:** Les DVR sont souvent livrés avec un logiciel de gestion dédié permettant aux utilisateurs de visualiser, de rechercher et de gérer les enregistrements stockés.

1.4.5.2 Enregistreurs Vidéo en Réseau (NVR)

Les NVR sont spécialement conçus pour les caméras de surveillance IP, offrant des fonctionnalités avancées adaptées à ces systèmes modernes :

- ⑩ **Gestion des flux vidéo IP:** Les NVR sont capables de gérer et d'enregistrer simultanément plusieurs flux vidéo haute résolution provenant de multiples caméras IP.
- ⑩ **Stockage sur réseau:** Contrairement aux DVR, les NVR stockent les enregistrements sur des périphériques de stockage réseau (NAS) ou dans le cloud, offrant une plus grande flexibilité et une capacité de stockage étendue.
- ⑩ **Fonctions avancées:** Ils peuvent intégrer des fonctionnalités telles que la détection de mouvement avancée, l'analyse vidéo intelligente et la compression vidéo efficace pour optimiser l'utilisation de la bande passante et de l'espace de stockage.
- ⑩ **Connectivité et intégration:** Les NVR sont généralement équipés de ports réseau Gigabit Ethernet pour une connectivité rapide et stable avec les caméras IP et d'autres périphériques du réseau de sécurité.

1.4.5.3 Comparaison et Choix

Le choix entre un DVR et un NVR dépend largement de l'infrastructure existante, des besoins spécifiques en matière de sécurité et des préférences de gestion des données. Voici quelques points à considérer :

- ⑩ **Type de caméras:** Les DVR conviennent aux caméras analogiques traditionnelles, tandis que les NVR sont indispensables pour les systèmes basés sur des caméras IP.
- ⑩ **Capacité de stockage et évolutivité:** Les NVR offrent généralement une évolutivité plus facile grâce à leur capacité de stockage sur réseau extensible.

Chapitre 1 : Généralités sur la télésurveillance

- ⑩ **Fonctionnalités avancées:** Si des fonctionnalités avancées telles que l'analyse vidéo intelligente sont nécessaires, les NVR sont souvent le choix privilégié.



Figure 10: un DVR et un NVR

En résumé, que ce soit avec un DVR ou un NVR, les enregistreurs jouent un rôle crucial dans la gestion des données vidéo dans les systèmes de télésurveillance, offrant aux utilisateurs la possibilité de surveiller, de sauvegarder et d'analyser les activités capturées pour assurer la sécurité et la conformité.

1.4.6 Transmetteurs dans les Systèmes de Télésurveillance

Les transmetteurs jouent un rôle essentiel dans les systèmes de télésurveillance en permettant la transmission fiable et sécurisée des données collectées par les capteurs et les caméras vers les stations de surveillance ou les centres de contrôle. Voici une détaillisation des différents types de transmetteurs utilisés :

1.4.6.1 Transmetteurs IP

Les transmetteurs IP sont largement utilisés dans les systèmes modernes de télésurveillance en raison de leur capacité à transmettre des données via Internet. Voici quelques caractéristiques clés des transmetteurs IP :

- ⑩ **Connectivité Internet** : Ils utilisent des protocoles standard tels que TCP/IP pour la communication avec les serveurs de gestion.
- ⑩ **Haute résolution** : Ils supportent souvent la transmission de vidéos haute définition, permettant une surveillance détaillée.

Chapitre 1 : Généralités sur la télésurveillance

- ⑩ **Flexibilité** : Ils peuvent être intégrés dans des systèmes de télésurveillance existants et sont souvent compatibles avec divers logiciels de gestion et plates-formes cloud.

1.4.6.2 Transmetteurs GSM/GPRS

Les transmetteurs GSM/GPRS utilisent les réseaux mobiles pour transmettre les données de surveillance. Voici leurs principales caractéristiques :

- ⑩ **Communication sans fil** : Ils sont idéaux pour les sites où l'accès à Internet filaire est limité ou non disponible.
- ⑩ **Fiabilité** : Ils utilisent les infrastructures de téléphonie mobile existantes, assurant une couverture étendue dans de nombreuses régions.
- ⑩ **Simplicité d'installation** : Ils nécessitent souvent seulement une carte SIM et peuvent être configurés pour envoyer des alertes par SMS en cas d'incident.

1.4.6.3 Transmetteurs Radio

Les transmetteurs radio utilisent des ondes radio pour la transmission des données, offrant une alternative robuste et autonome dans certaines situations :

- ⑩ **Indépendance du réseau filaire** : Ils sont utiles dans les zones reculées ou isolées où d'autres formes de connectivité sont limitées.
- ⑩ **Faible latence** : Ils offrent généralement une réponse rapide en cas d'alarme ou d'incident.
- ⑩ **Sécurité renforcée** : Certains transmetteurs radio utilisent des protocoles de communication cryptés pour garantir la confidentialité des données.

1.4.6.4 Transmetteurs Satellite

Les transmetteurs satellite sont utilisés dans des environnements extrêmement éloignés ou où les autres formes de connectivité sont indisponibles. Voici leurs caractéristiques principales :

- ⑩ **Couverture globale** : Ils utilisent des satellites en orbite pour transmettre les données, offrant une couverture mondiale.
- ⑩ **Fiabilité dans des conditions extrêmes** : Ils sont utilisés dans des applications telles que la surveillance de zones reculées, les navires en mer, ou les installations dans des régions éloignées.

Chapitre 1 : Généralités sur la télésurveillance

- ⑩ **Coût élevé** : En raison de la technologie impliquée, les transmetteurs satellite peuvent être plus coûteux à installer et à maintenir par rapport à d'autres types de transmetteurs.

1.4.7 Stations de surveillance :

Les stations de surveillance jouent un rôle central dans les systèmes de télésurveillance en centralisant la réception, la gestion et la réponse aux données provenant des caméras, capteurs et autres dispositifs de sécurité.



Figure 11: Stations de surveillance

1.4.7.1 Infrastructure Physique

Les stations de surveillance sont généralement situées dans des installations sécurisées, souvent appelées centres de contrôle ou postes de sécurité. Elles sont équipées de plusieurs composants essentiels :

- ⑩ **Écrans de Surveillance** : Les opérateurs utilisent ces écrans pour visualiser en temps réel les flux vidéo provenant des caméras de surveillance installées dans divers lieux.
- ⑩ **Postes de Travail** : Chaque opérateur dispose d'un poste de travail équipé d'ordinateurs et de logiciels spécialisés pour la gestion des systèmes de télésurveillance, y compris la visualisation des caméras et l'analyse des données.

Chapitre 1 : Généralités sur la télésurveillance

- ⑩ **Systèmes de Communication** : Pour assurer une communication efficace avec les équipes sur le terrain, les autorités locales ou d'autres intervenants en cas d'incident.

1.4.7.2 Personnel et Formation

Le personnel des stations de surveillance est composé d'opérateurs formés pour surveiller attentivement les écrans, interpréter les événements capturés par les caméras et répondre aux alertes. Les aspects importants incluent :

- ⑩ **Formation en Surveillance** : Les opérateurs sont formés pour identifier les comportements suspects, suivre les protocoles de sécurité et réagir rapidement aux situations d'urgence.
- ⑩ **Rotation et Planification** : Pour assurer une surveillance 24/7, les stations de surveillance fonctionnent par roulement, avec des équipes travaillant par quarts pour garantir une présence continue.

1.4.7.3 Logiciels de Gestion

Les stations de surveillance utilisent des logiciels spécialisés pour gérer et analyser les données recueillies par les systèmes de télésurveillance. Ces logiciels offrent plusieurs fonctionnalités :

- ⑩ **Surveillance en Temps Réel** : Permet aux opérateurs de visualiser les caméras en direct et de suivre les événements en temps réel.
- ⑩ **Analyse Vidéo Avancée** : Utilisation d'algorithmes pour détecter automatiquement les comportements suspects ou les événements anormaux.
- ⑩ **Gestion des Alertes** : Centralisation des alertes provenant de différents capteurs et caméras, avec des procédures prédéfinies pour chaque type d'incident.

1.4.7.4 Intégration avec d'Autres Services

Les stations de surveillance peuvent être intégrées à d'autres services de sécurité et d'urgence, tels que la police, les services d'incendie ou les services médicaux d'urgence. Cette intégration permet une réponse coordonnée et rapide en cas d'incident.

1.4.7.5 Sécurité Physique et Logique

En raison de la sensibilité des données et de l'importance de maintenir l'intégrité opérationnelle, les stations de surveillance sont également sécurisées physiquement et logiquement. Les mesures de sécurité comprennent :

Chapitre 1 : Généralités sur la télésurveillance

- ⑩ **Contrôles d'Accès** : Pour limiter l'entrée aux personnes autorisées uniquement.
- ⑩ **Sécurité Informatique** : Protection contre les cyberattaques et la manipulation des données, assurant ainsi la confidentialité et l'intégrité des informations traitées.

En résumé, les stations de surveillance représentent le cœur opérationnel des systèmes de télésurveillance, fournissant une surveillance continue, une réponse rapide aux incidents et une gestion efficace des situations critiques pour assurer la sécurité des sites surveillés.

1.5 Application de la télésurveillance :

La télésurveillance trouve des applications variées dans différents secteurs, offrant une solution efficace pour la sécurité et la gestion des données à distance. Cette section explore comment la télésurveillance est utilisée dans divers contextes :

1.5.1 Résidentielle

La télésurveillance résidentielle est devenue courante pour assurer la sécurité des maisons et des propriétés privées. Elle inclut l'utilisation de caméras de sécurité, de capteurs d'intrusion, et de systèmes d'alarme connectés à des centres de surveillance. Les résidents peuvent surveiller leur domicile en temps réel via des applications mobiles et recevoir des alertes en cas d'activité suspecte. Cette technologie permet une réponse rapide aux incidents et dissuade les intrusions.

1.5.2 Commerciale

Dans le secteur commercial, la télésurveillance est cruciale pour la protection des biens et la sécurité des employés. Les commerces, bureaux et établissements commerciaux utilisent des systèmes de caméras IP haute résolution, couplés à des logiciels de gestion vidéo, pour surveiller les zones sensibles et détecter les comportements anormaux. Ces systèmes sont souvent intégrés à des systèmes de contrôle d'accès pour renforcer la sécurité globale des installations.

1.5.3 Industrielle

Les applications industrielles de la télésurveillance visent à protéger les sites industriels et les infrastructures critiques contre les intrusions, les accidents et les actes de vandalisme. Les usines, les entrepôts et les installations de production utilisent des systèmes de surveillance avancés pour surveiller les processus de fabrication, détecter les anomalies dans les équipements, et assurer la conformité aux normes de sécurité et environnementales. Ces systèmes contribuent également à optimiser l'efficacité opérationnelle en permettant une surveillance continue à distance.

Chapitre 1 : Généralités sur la télésurveillance

1.5.4 Urbaine

Dans les environnements urbains, la télésurveillance est utilisée pour surveiller les espaces publics, les infrastructures urbaines et les systèmes de transport. Les caméras de surveillance installées dans les rues, les parcs, les transports en commun et aux abords des bâtiments gouvernementaux permettent aux autorités de surveiller la sécurité publique, de prévenir la criminalité et de gérer les situations d'urgence. Ces systèmes jouent un rôle crucial dans la sécurisation des zones densément peuplées et dans la protection des infrastructures essentielles.

1.6 Avantages et Inconvénients de la télésurveillance :

La télésurveillance présente un ensemble d'avantages et d'inconvénients qu'il est essentiel d'analyser pour comprendre pleinement son impact et ses limitations.

1.6.1 Avantages :

La télésurveillance offre plusieurs avantages significatifs :

- ⑩ **Sécurité renforcée** : Permet une surveillance continue et en temps réel des environnements sensibles, réduisant ainsi les risques d'intrusion, de vol et d'autres incidents criminels.
- ⑩ **Réactivité accrue** : Notifications instantanées en cas d'incident permettant des réponses rapides et efficaces des autorités compétentes ou des équipes de sécurité.
- ⑩ **Réduction des coûts** : Comparée à la sécurité physique traditionnelle, la télésurveillance peut souvent être moins coûteuse à long terme, en réduisant les besoins en personnel de sécurité sur site.
- ⑩ **Surveillance à distance** : Permet la surveillance de sites distants ou difficiles d'accès sans nécessiter une présence physique constante.
- ⑩ **Intégration avec d'autres systèmes** : Capacité à intégrer des technologies complémentaires comme l'analyse vidéo avancée et les systèmes d'alerte automatisés pour une sécurité améliorée.

1.6.2 Inconvénients :

Cependant, la télésurveillance présente également certains inconvénients et défis :

- ⑩ **Coût initial élevé** : L'installation et la mise en place initiales peuvent être coûteuses, en particulier pour les systèmes complexes ou les grandes installations.

Chapitre 1 : Généralités sur la télésurveillance

- ⑩ **Dépendance à la technologie** : Vulnérabilité aux pannes de courant, aux défaillances des réseaux de communication et aux cyberattaques qui pourraient compromettre la sécurité.
- ⑩ **Questions de vie privée** : Préoccupations concernant la collecte et l'utilisation des données personnelles, nécessitant une gestion rigoureuse pour respecter les normes de protection des données.
- ⑩ **Maintenance et mise à jour** : Nécessité de maintenir et de mettre à jour régulièrement les systèmes pour assurer leur efficacité et leur sécurité continues.
- ⑩ **Limitations techniques** : Bien que les technologies évoluent rapidement, certaines limitations persistent, telles que la qualité de la vidéo dans des conditions de faible luminosité ou les fausses alarmes.

1.7 Discussion

Dans ce chapitre, nous nous sommes intéressés à l'étude et à la présentation de la télésurveillance. Nous avons commencé par définir ce concept, puis exploré les différents équipements nécessaires pour sa mise en place on présente quelques types de caméras et de capteurs couramment utilisés dans la télésurveillance, ainsi que ses domaines d'application, tout en examinant ses avantages et inconvénients.

Chapitre 2 :

Conception

du système

Chapitre 2 : Conception du système

2.1 Préambule :

La conception d'un système électronique est une étape cruciale qui détermine le succès global du projet. L'objectif de ce chapitre est de mettre l'accent sur la description de notre projet, ainsi que de ces besoins matériels et logiciels.

2.2 Description du projet:

Notre projet permet de développer un système de surveillance à distance basé sur des microcontrôleurs d'Espressif Systems. Ces cartes mettent en œuvre une variété de capteurs, d'actionneurs et de modules pour surveiller l'environnement et interagir avec une application de messagerie (Telegram) pour envoyer des notifications et recevoir des commandes. Notre projet a diverses fonctionnalités les voici:

- ⑩ Détection de Mouvement : Utilise des capteurs de mouvement pour détecter les intrusions et déclencher des alertes.
- ⑩ Capture de Photos : Prend des photos lorsque du mouvement est détecté et les envoie via Telegram.
- ⑩ Notifications en Temps Réel : Envoie des notifications instantanées sur Telegram pour informer l'utilisateur des activités suspectes.
- ⑩ Surveillance de Gaz : Utilise un capteur de gaz pour détecter la présence de gaz nocifs et alerter en cas de danger.
- ⑩ Contrôle d'Accès : Permet de gérer l'accès à des zones sécurisées via des badges, enregistrant chaque accès.
- ⑩ Interaction à Distance : L'utilisateur peut envoyer des commandes à distance via Telegram pour contrôler et interagir avec le système.
- ⑩ Stockage et Accès aux Données : Les données collectées, telles que les photos, sont stockées pour une consultation ultérieure.

Chapitre 2 : Conception du système

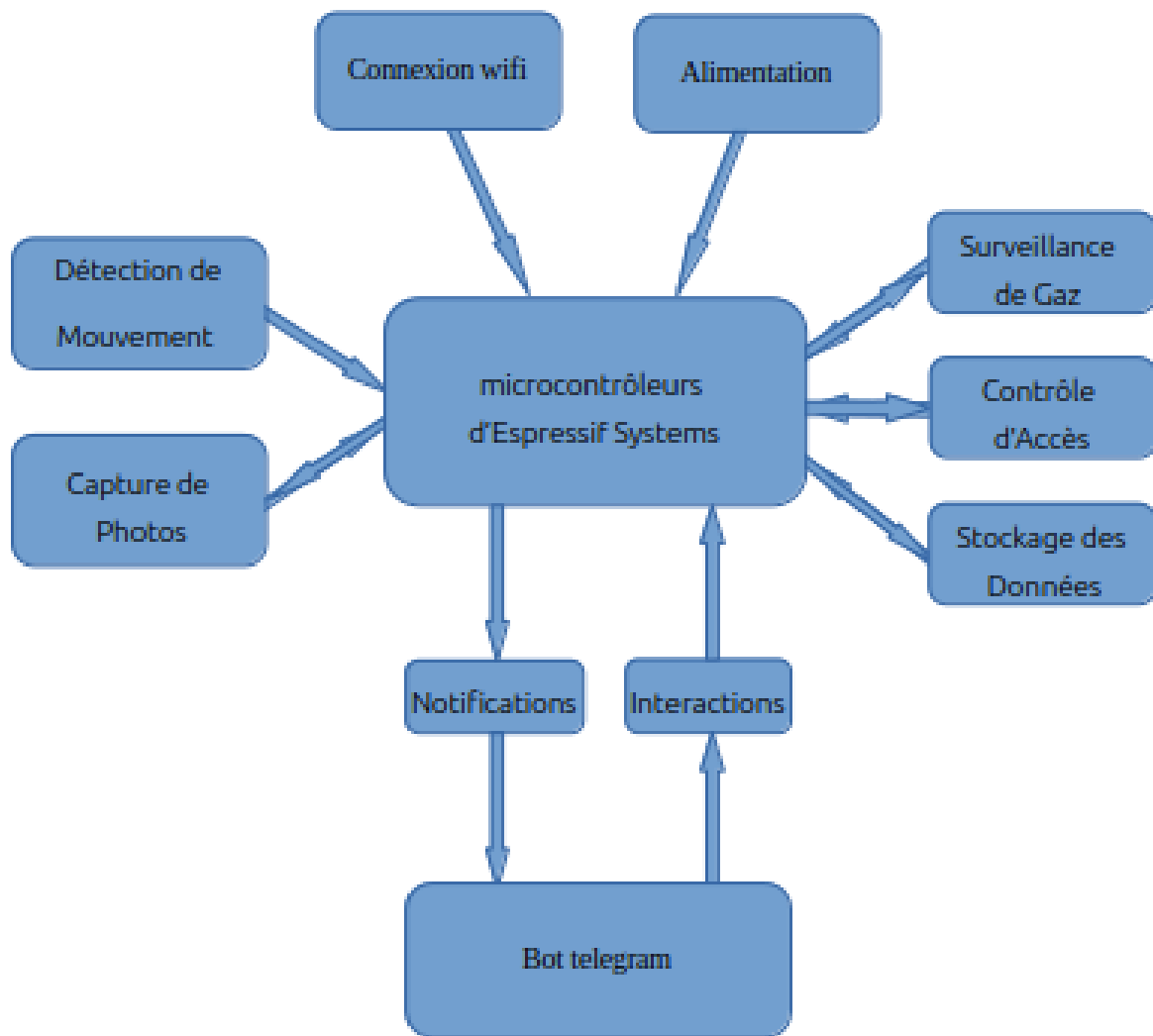


Figure 12: schéma synoptique du système

2.3 Description matérielle :

2.3.1 La carte ESP32-CAM :

2.3.1.1 Présentation de la carte ESP32-CAM :

L'ESP32-CAM est une carte de développement basée sur le microcontrôleur ESP32 d'Espressif Systems. En plus de sa connectivité Wi-Fi et Bluetooth, elle est équipée d'une caméra, de 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. Elle est particulièrement populaire pour les projets de surveillance, de reconnaissance d'images, et autres applications nécessitant la capture et le traitement de l'image.

Chapitre 2 : Conception du système

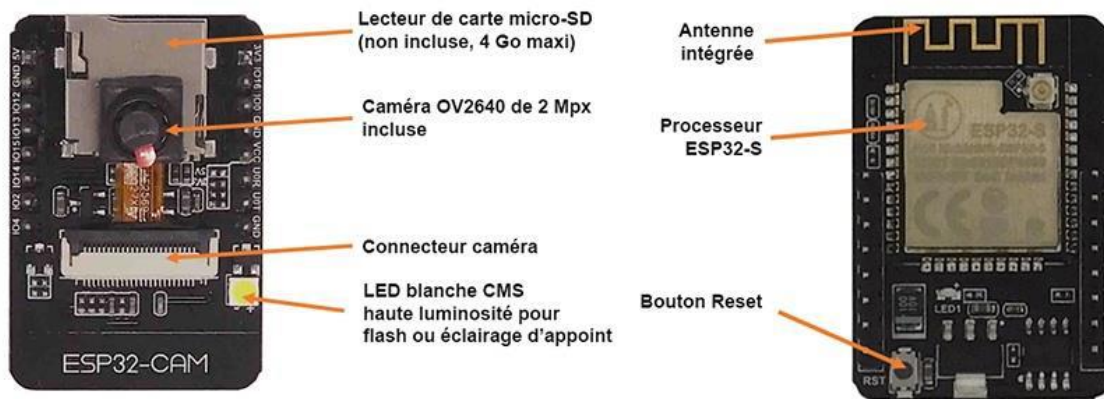


Figure 13: Carte ESP32-CAM

2.3.1.2 Caractéristiques techniques de l'ESP32-CAM

Voici les principales caractéristiques techniques de la carte ESP32-CAM :

- ⑩ **Flash SPI** : 32 Mbit par défaut
- ⑩ **RAM** : 520 Ko SRAM + 4 Mo PSRAM Externe
- ⑩ **Bluetooth** : Bluetooth 4.2 et Bluetooth Low Energy (BLE)
- ⑩ **Wi-Fi** : 802.11 b/g/n
- ⑩ **Ports GPIO** : Plusieurs broches d'Entrée/Sortie Général, bien que certaines soient utilisées par la caméra.
- ⑩ **Débit en bauds UART** : 115200 bps par défaut
- ⑩ **Gamme de fréquences Wi-Fi** : 2412 ~ 2484 MHz
- ⑩ **Antenne** : Circuit imprimé intégré et connecteur IPEX
- ⑩ **Sécurité** : WPA/WPA2/WPA2-Enterprise/WPS
- ⑩ **Caméra** : Supporte les caméras OV2640 avec un capteur de 2MP et une matrice UXGA de 1622x1200 pixels, sortie en format compressé 8 bits, capable de capturer entre 15 et 60 images par seconde.
- ⑩ **ESP32-S** : Un microcontrôleur double cœur Xtensa LX6 cadencé à 240 MHz.
- ⑩ **Stockage** : Support pour une carte microSD jusqu'à 4 Go.
- ⑩ **Interface** : Connectée via une interface CSI (Camera Serial Interface).
- ⑩ **UART, SPI, I2C, PWM, ADC, DAC** : Interfaces de communication et de contrôle.
- ⑩ **LED Flash** : Une LED blanche pour éclairage ou flash.

Chapitre 2 : Conception du système

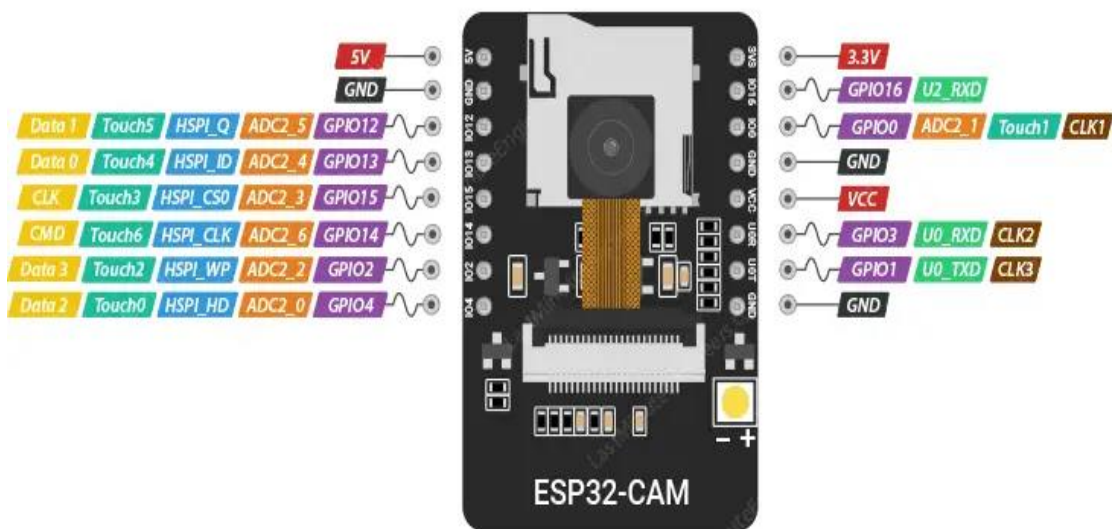
⑩ **Alimentation** : Via micro-USB ou directement par des broches 5V et GND.

⑩ **Dimensions** : Environ 27 mm x 40.5 mm.

2.3.1.3 Brochage de la carte ESP32-CAM

L'ESP32-CAM dispose de 16 broches au total, bien que seulement 10 soient disponibles en raison de l'utilisation interne pour la caméra et la PSRAM. Voici les détails des principales broches :

- ⑩ **Broches d'alimentation** : 5V et 3V3 (il est recommandé d'utiliser la broche 5V pour l'alimentation)
- ⑩ **GND** : Broche de masse
- ⑩ **GPIO 1 et GPIO 3** : Broches série (TX et RX)
- ⑩ **GPIO 0** : Utilisée pour passer en mode programmation lorsque connectée à GND
- ⑩ **GPIO 2, 4, 12, 13, 14, 15** : Utilisées pour interfacer la carte microSD, mais peuvent également servir d'entrées/sorties normales si la carte microSD n'est pas utilisée
- ⑩ **LED intégrée** : Connectée à GPIO 4, peut fonctionner comme un flash pour la caméra
- ⑩ **LED rouge** : Connectée à GPIO33, peut indiquer l'état du Wi-Fi ou d'autres événements



ESP32-CAM Pinout

Last Minute ENGINEERS.com

Figure 14: Broches de la Carte ESP-32-CAM

Chapitre 2 : Conception du système

2.3.1.4 Le module MB pour l'ESP32-CAM:

Le module MB pour l'ESP32-CAM est une carte adaptateur qui facilite la programmation et l'alimentation de l'ESP32-CAM. Ce module est souvent utilisé pour simplifier le développement et le prototypage avec l'ESP32-CAM en fournissant une interface USB pour le téléchargement de code et en offrant des broches supplémentaires pour une connexion facile

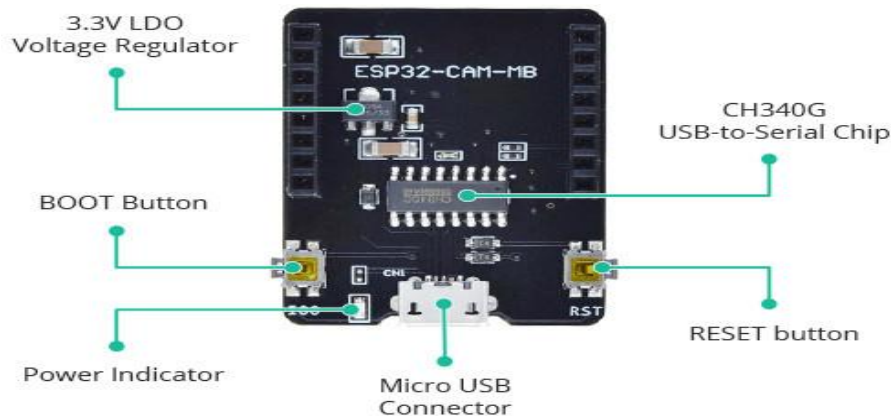


Figure 15: Le module MB pour l'ESP32-CAM

2.3.2 La carte ESP32:

2.3.2.1 Présentation de la carte ESP32 :

La carte ESP32, conçue par Espressif Systems, est une plateforme de développement électronique basée sur le microcontrôleur ESP32. Elle est particulièrement prisée pour les projets d'Internet des objets (IoT) grâce à sa connectivité sans fil robuste, sa puissance de traitement élevée et sa faible consommation d'énergie. La carte ESP32 est dotée de nombreuses interfaces d'entrées/sorties (GPIO, SPI, I2C, UART, etc.) permettant la connexion de divers périphériques, ainsi que d'une connectivité Wi-Fi et Bluetooth intégrée. Elle peut être programmée via divers environnements de développement, notamment Arduino IDE, MicroPython et ESP-IDF.

Disponible en plusieurs modèles, tels que l'ESP32-WROOM-32, l'ESP32-DevKitC et l'ESP32-PICO-KIT, la carte ESP32 répond à une vaste gamme de besoins et d'applications. Elle est largement utilisée dans divers projets électroniques, allant des dispositifs de surveillance à distance et des capteurs sans fil aux systèmes de domotique et aux robots. Cette polyvalence en fait un outil essentiel pour les développeurs et les ingénieurs travaillant sur des solutions IoT innovantes.

Chapitre 2 : Conception du système

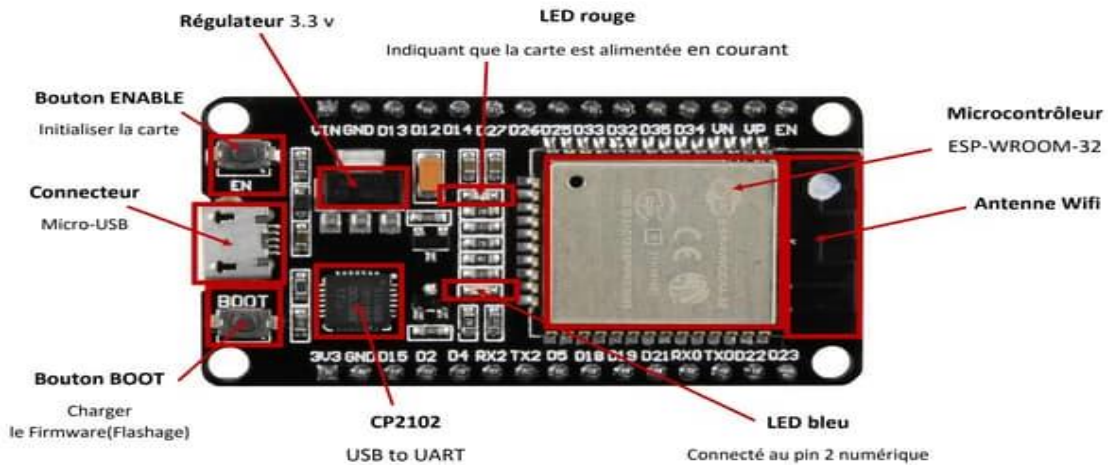


Figure 16: carte ESP32

2.3.2.2 Caractéristique de la carte ESP 32 :

Les caractéristiques de la carte ESP32 peuvent varier en fonction du modèle spécifique, mais voici quelques-unes des caractéristiques générales communes à la plupart des cartes ESP32 :

- ⑩ **Microcontrôleur** : ESP32 à double cœur basé sur l'architecture Xtensa LX6 de la société Cadence.
- ⑩ **Vitesse du processeur** : Jusqu'à 240 MHz.
- ⑩ **Mémoire** : RAM de 520 Ko à 4 Mo, mémoire flash de 4 Mo à 16 Mo.
- ⑩ **Connectivité sans fil** : Wi-Fi 802.11 b/g/n, Bluetooth 4.2 BR/EDR et BLE.
- ⑩ **Interfaces d'entrées/sorties** : GPIO, SPI, I2C, UART, etc.
- ⑩ **Autres périphériques** : Horloge temps réel (RTC), convertisseur analogique-numérique (CAN), convertisseur numérique-analogique (CNA), etc.
- ⑩ **Programmation** : Support pour divers environnements de développement tels que Arduino IDE, MicroPython, ESP-IDF, etc.
- ⑩ **Alimentation** : Peut être alimentée via un port USB ou une alimentation externe.
- ⑩ **Sécurité** : Fonctionnalités de cryptographie matérielle intégrée (SHA, RSA, AES) pour une sécurité renforcée.

2.3.2.3 brochages de la carte ESP32 :

L'ESP32 dispose d'un grand nombre de broches d'entrées/sorties générales (GPIO) et d'autres interfaces pour connecter divers périphériques. Voici une figure détaillant les broches et leurs fonctions :

Chapitre 2 : Conception du système

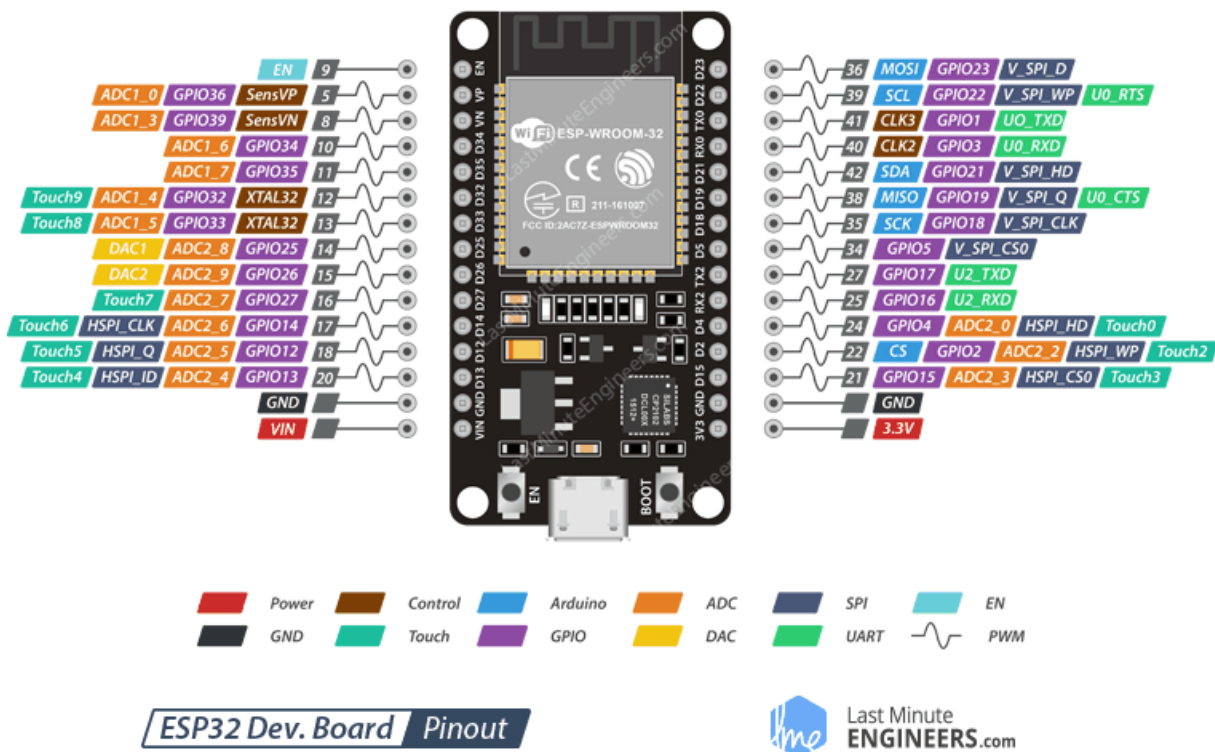


Figure 17: brochage de la carte ESP32

2.3.3 Module RFID :

RFID veut dire radio-frequency identification (identification par radiofréquence), c'est un module qui utilise des champs électromagnétiques pour transférer des données sur de courtes distances. La RFID est utile pour identifier les personnes, transactions, ou même pour ouvrir des portes. Et seule la personne ayant les bonnes informations sur sa carte est autorisée d'accès. Un système RFID a besoin de badge et d'un lecteur

2.3.3.1 Badges RFID :

Ces badges sont attachés à l'objet à identifier et permettent un contrôle d'accès fiable et sécuritaire. Chaque badge a sa propre identification unique (UID). Ils sont munis d'une antenne et d'une puce aussi, pour permettre la transmission de données avec un lecteur.



Figure 18 : Badge RFID

Chapitre 2 : Conception du système

2.3.3.2 Le lecteur RFID RC522

C'est un émetteur-récepteur radio bidirectionnel qui envoie un signal au badge RFID et lit sa réponse. Conçu pour créer un champ électromagnétique de 13,56 MHz qu'il utilise pour communiquer avec les Tags. Le lecteur est basé sur le circuit intégré MFRC522, et communique avec un microcontrôleur via une interface périphérique série (SPI) avec un débit de données maximal de 10 Mbps. Il prend également en charge la communication via les protocoles I2C et UART. Voir la figure ci-dessous.

2.3.3.3 Caractéristiques du lecteur RFID RC522 :

- ⑩ RFID RC522 utilise l'induction mutuelle pour activer les cartes et 13,56 MHz pour le transfert de données.
- Les cartes RFID sont utilisables des deux côtés du module à 5 cm maximum.
- Seulement 3,3 V sont nécessaires pour activer l'appareil.
- Son mode de veille automatique en fait un module de consommation d'énergie moindre.
- Le module dispose de trois types de communications (UART, SPI, I2C). Par conséquent, il est utilisable avec presque tous les microcontrôleurs ou appareils du marché.
- Les cartes RFID et le lecteur (RC522) peuvent transférer des données jusqu'à 10 Mb/s.

2.3.3.4 Le Brochage du lecteur RFID RC522:

- ⑩ **VCC** : utilisé pour alimenter le module, généralement on utilise 3,3 V
- ⑩ **RST**: broche de réinitialisation - utilisée pour réinitialiser ou éteindre le module
- ⑩ **Ground**: connectée à la masse du système.
- ⑩ **IRQ**: broche d'interruption - utilisée pour interrompre le module lorsqu'un périphérique entre dans la portée.
- ⑩ **MISO/SCL/Tx**: la broche MISO lorsqu'elle est utilisée pour la communication SPI, agit comme SCL pour I2C et Tx pour UART.
- ⑩ **MOSI**: sortie maître entrée esclave pour la communication SPI
- ⑩ **SCK**: broche d'horloge série - utilisée pour fournir une source d'horloge
- ⑩ **SS/SDA/Rx**: Agit comme entrée série (SS) pour la communication SPI, SDA pour I2C et Rx pour l'UART

Chapitre 2 : Conception du système

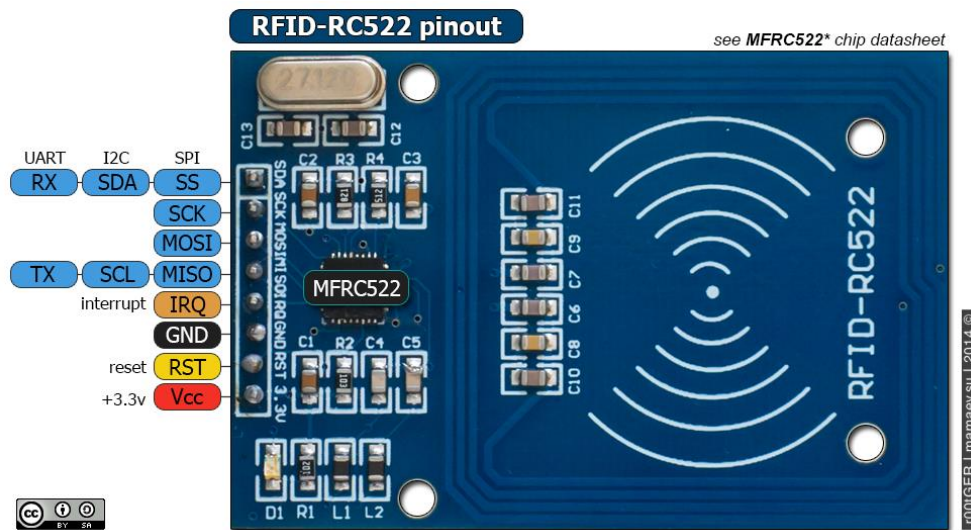


Figure 19: Le Brochage du lecteur RFID RC522

2.3.4 Capteur de mouvement PIR HC-SR501 :

Le capteur de mouvement PIR (Passive Infrared Sensor) est un dispositif électronique qui détecte la lumière infrarouge émise par les objets dans son champ de vision. Il est largement utilisé dans les systèmes d'alarme et de détection de présence en raison de son coût réduit et de son efficacité avérée.

Le module HC-SR501 PIR se compose de trois broches principales : une pour la masse, une pour l'alimentation VCC, et une broche de sortie qui indique un niveau logique élevé lorsque le capteur détecte un objet. De plus, il possède deux potentiomètres : l'un ajuste la sensibilité du capteur, et l'autre contrôle la durée pendant laquelle le signal de sortie reste haut après la détection d'un objet, réglable de 3 secondes à 5 minutes.

En outre, le module comporte trois autres broches avec un cavalier entre deux d'entre elles, utilisées pour sélectionner les modes de déclenchement. Le premier mode, appelé "déclenchement non répétitif", fait passer automatiquement la sortie du niveau haut au niveau bas lorsque le temps de retard est écoulé après la détection d'un objet. Le second mode, "déclenchement répétitif", maintient la sortie haut tant que l'objet détecté reste dans la portée du capteur.

Sur le plan technique, ce capteur repose sur un circuit intégré BISS001, conçu spécifiquement pour cette application. Il fonctionne avec une tension de 3 à 5V et consomme très peu de courant, le rendant idéal pour une utilisation sur batterie.

Chapitre 2 : Conception du système

Fondamentalement, le module HC-SR501 PIR utilise un capteur pyroélectrique qui génère de l'énergie en réponse à la chaleur détectée. Ainsi, lorsque des êtres humains ou des animaux pénètrent dans la portée du capteur, leurs émissions thermiques sous forme de rayonnement infrarouge sont détectées comme des mouvements.

En résumé, ce capteur détecte les variations de rayonnement infrarouge émises par les objets en mouvement dans son champ de vision, les convertissant en signaux électriques interprétables comme des détections de mouvement.

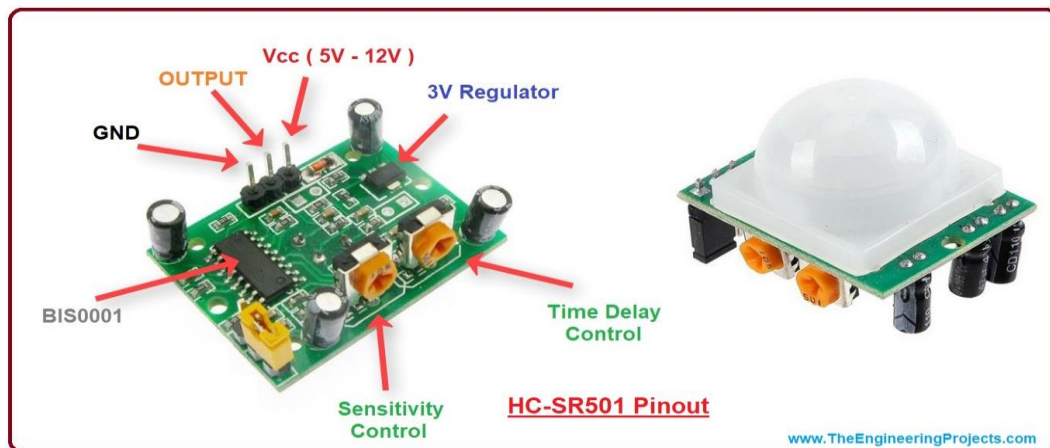


Figure 20: Le module capteur HC-SR501 PIR

2.3.4.1 Caractéristique de PIR:

- Tension d'alimentation : 5V – 20V
- Consommation : 65mA
- Sortie TTL : 3.3V, 0V
- Temps pendant lequel la sortie est à 1: Ajustable (3s à 5min)
- Temps de verrouillage T_i : ~0.2 sec à 3s selon fabricant
- Déclenchement : avec déclenchement , sans déclenchement
- Sensibilité : moins de 120°, jusque 7 mètres
- Température : – 15 à +70°C
- Dimension : 32*24 mm, distance entre trous de montage 28mm, vis M2Diamètre de la lentille : 23mm

2.3.5 Capteur de gaz MQ-135 :

Capteur Gaz MQ-135 est un capteur de contrôle de la qualité de l'air et de détection des gaz nocifs (CO₂, l'ammoniac, soufre, benzène, l'oxyde d'azote, les composants aromatiques, l'alcool, la fumée, etc) . Il est capable de repérer différents éléments chimiques présents dans l'air , En cas

Chapitre 2 : Conception du système

d'augmentation d'une concentration chimique, le capteur transforme cette concentration qui est dans l'air en une plage de tension adéquate, qui peut être traitée par un microcontrôleur (la Carte arduino, ESP32...).

Dans nos projets utilisant le MQ135 avec des microcontrôleurs comme l'ESP32, la sortie analogique du capteur peut être lue et traitée pour déterminer la concentration des gaz détectés, En calibrant et en interprétant les lectures des capteurs, on peut évaluer la qualité de l'air et même calculer un indice de qualité de l'air (IQA) pour une surveillance et une analyse complètes. Pour lire les données analogique du capteur MQ135 sur un ESP32, On devra utiliser une résistance de division pour adapter la tension de sortie analogique du capteur (5v) a la plage d'entrée de l'esp32 (3,3v). Les résistances couramment utilisées sont 62k et 39k

Ce capteur possède 4 broches : VCC, GND, DOUT, et AOUT :

La broche VCC est alimentée avec 5v.

Le pin GND est raccordé à la terre.

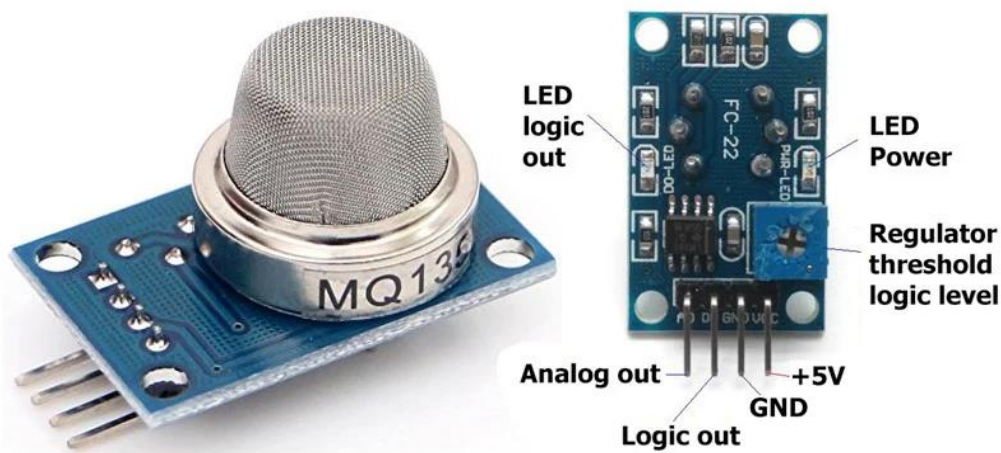


Figure 21: Capteur MQ-135

2.3.5.1 Caractéristique du Capteur MQ-135:

- Gaz détectables: ammoniac, benzène, alcool, fumée.
- Plage de détection: 10-300 ppm NH₃, 10-1000 ppm Benzène, 10-600 ppm d'alcool.
- Temps de réponse: $\leq 10s$ (réponse de 70%)
- Temps de récupération: $\leq 30s$ (70% de réponse)
- Température ambiante d'opération: -20 à + 55 C
- Humidité: $\leq 95\%$ HR
- Teneur environnementale en oxygène: 21%
- Température de stockage: -20 – +70

Chapitre 2 : Conception du système

- Humidité: $\leq 70\%$ RH
- Circuit d'entraînement simple

2.3.6 Le servomoteur sg90:

Le servomoteur (généralement abrégé en «Servo», du latin Servus, signifiant «esclave»)

Il s'agit d'un actionneur capable de maintenir une position statique, avec un débattement de 180 degrés. Le servomoteur est utilisé en modélisme afin de produire les mouvements nécessaires aux déplacements des oranges de direction. Il est commandé par l'intermédiaire d'un récepteur radio.

Il est constitué d'un potentiomètre, d'un moteur électrique à courant continu, d'un réducteur pour augmenter le couple et de palonnier qui se trouve en haut, c'est ce dernier qui effectue les rotations. Les servomoteurs servent en principe à actionner les parties mobiles d'un système. Ils sont prévus pour être commandés facilement en position ou en vitesse. En effet, ils sont équipés d'un système d'asservissement basé sur un potentiomètre rotatif qui sert à capter la rotation.

Doté de trois fils marron, rouge et jaune qui représentent respectivement la masse, l'alimentation et la sortie PWM

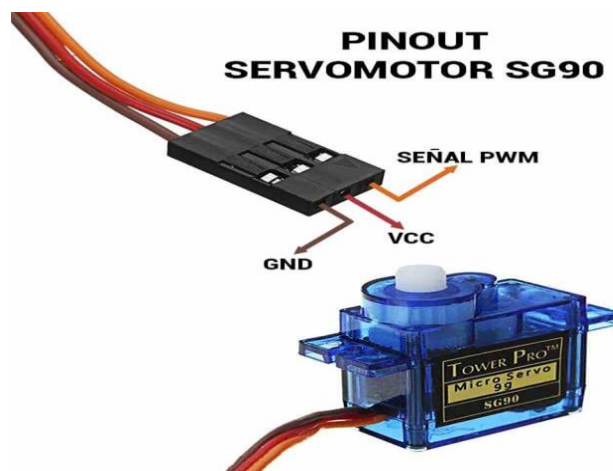


Figure 22: servomoteur sg90

2.3.6.1 Caractéristiques du Servomoteur SG90 :

- ⑩ **Type de Moteur** : Servomoteur analogique à rotation limité.
- ⑩ **Plage de Rotation** : Environ 180 degrés (90 degrés dans chaque direction).
- ⑩ **Couple** : Modéré, adapté aux petites charges.
- ⑩ **Vitesse de Rotation** : Variable selon la tension d'alimentation, généralement entre 0.1 et 0.2 secondes pour effectuer un mouvement de 60 degrés.
- ⑩ **Tension d'Alimentation** : Typiquement 4.8V à 6V DC.

Chapitre 2 : Conception du système

- ⑩ **Courant de Fonctionnement** : Environ 100mA en charge.
- ⑩ **Interface de Commande** : PWM (Pulse Width Modulation) pour contrôler la position.

2.3.7 Buzzer :

Les dispositifs audio de signalisation, comme les beeper ou les buzzers, peuvent être de nature électromécanique, piézoélectrique ou mécanique respectivement

Un buzzer, également connu sous le nom de bipeur, est un dispositif électromécanique ou piézoélectrique qui génère un son distinctif lorsqu'il est soumis à une tension : le bip. Il y a ceux qui requièrent une tension continue, tandis que d'autres requièrent une tension alternative. Son rôle principal consiste à transformer les signaux audio en son. En règle générale, on utilise la tension continue pour alimenter les appareils électriques, et elle est employée dans les situations nécessitant une tension alternative, comme les minuteries, les alarmes, les imprimantes, les alarmes et les ordinateurs. D'après son design, il a la capacité de générer divers sons tels que des alarmes, de la musique, des cloches et des sirènes.

Il est composé de deux broches, une positive et une négative. Le symbole '+' ou une borne plus longue représentent la borne positive de celui-ci. L'alimentation de cette borne est de 6 Volts, tandis que la borne négative est symbolisée par le symbole '-' ou borne courte et elle est reliée à la borne GND.



Chapitre 2 : Conception du système

Figure 23: buzzer

2.3.8 Les Leds :

On l'appelle également Diode électro lumineuse. Une diode électroluminescente, également connue sous le nom de DEL en français, ou LED, est un dispositif optoélectronique qui peut générer de la lumière lorsqu'il est parcouru par un courant électrique. Une diode électroluminescente ne permet que le courant électrique se déplace dans un seul sens et génère un rayonnement monochromatique ou polychromatique non cohérent lorsqu'un courant circule à travers elle.

Lorsque l'on souhaite émettre de la lumière, il est nécessaire de polariser directement une diode électroluminescente en sens direct .

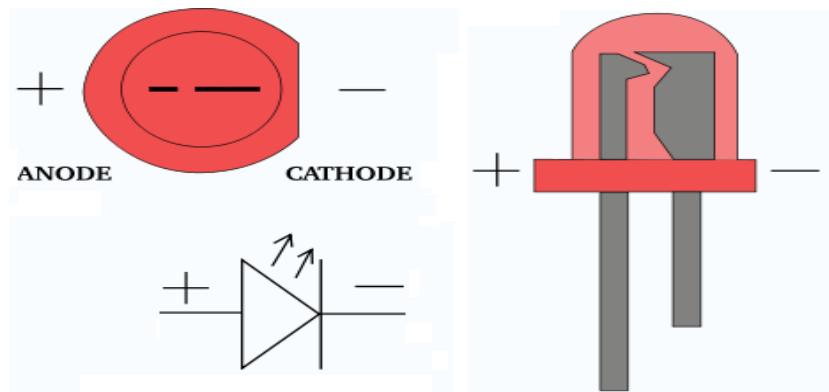


Figure 24: Diode électroluminescente et son symbole

Pour chaque couleur de led un voltage est précis comme suit :

Chapitre 2 : Conception du système

Figure 25: Tableau représentant le voltage de chaque led par rapport à sa couleur.

2.4 Description logicielle :

2.4.1 Logiciel Arduino IDE:

L'Arduino IDE (Integrated Development Environment), est un logiciel utilisé pour programmer les cartes électroniques Arduino mais en peut également programmer les cartes ESP. Il fournit un ensemble d'outils nécessaires pour écrire, compiler et téléverser du code sur les cartes. Il prend en charge la programmation dans le langage Arduino, qui est basé sur le langage de programmation C/C++.

2.4.1.1 Description du logiciel Arduino IDE:

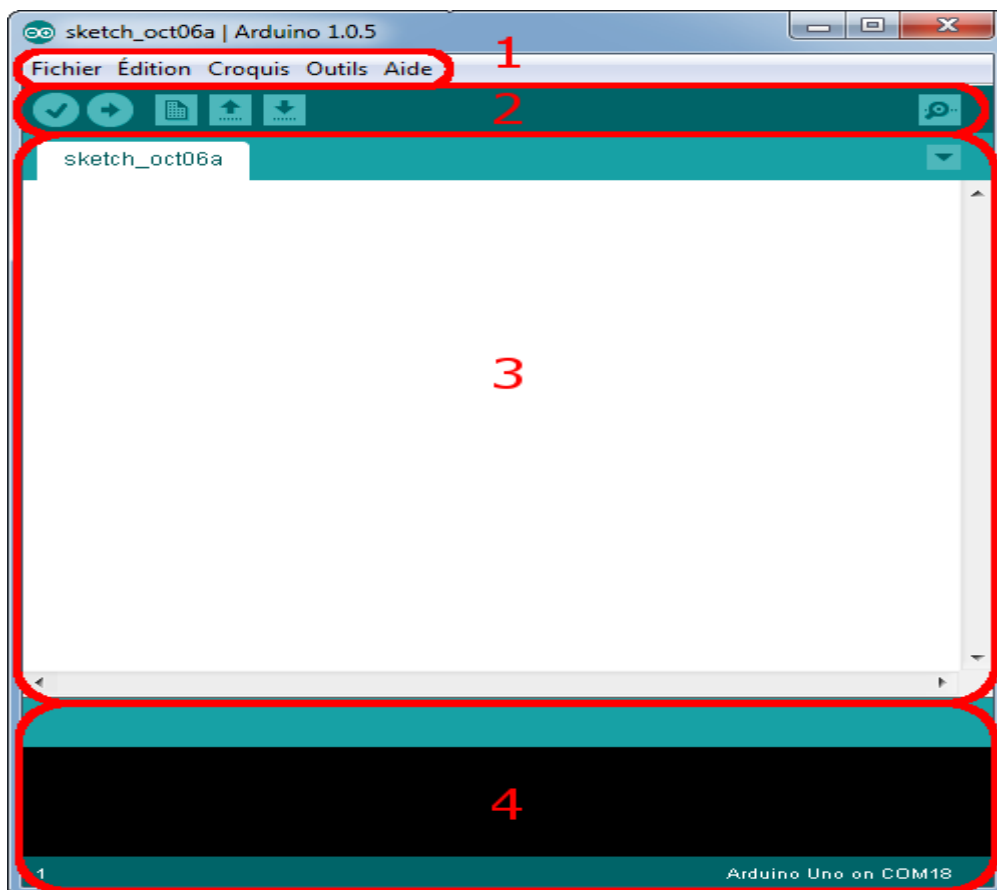


Figure 26: L'interface de l'IDE Arduino

L'interface de l'IDE Arduino est plutôt simple, elle offre plusieurs zones, chacune de ces zones à une fonctionnalité bien précise.

Chapitre 2 : Conception du système

- La zone de la barre d'icônes: permet de réaliser diverses actions (vérifier le code, téléverser le code, ouvrir un nouveau projet, ouvrir un projet de programme Arduino existant, sauvegarder le projet, ouvrir le moniteur série pour communiquer avec l'Arduino)

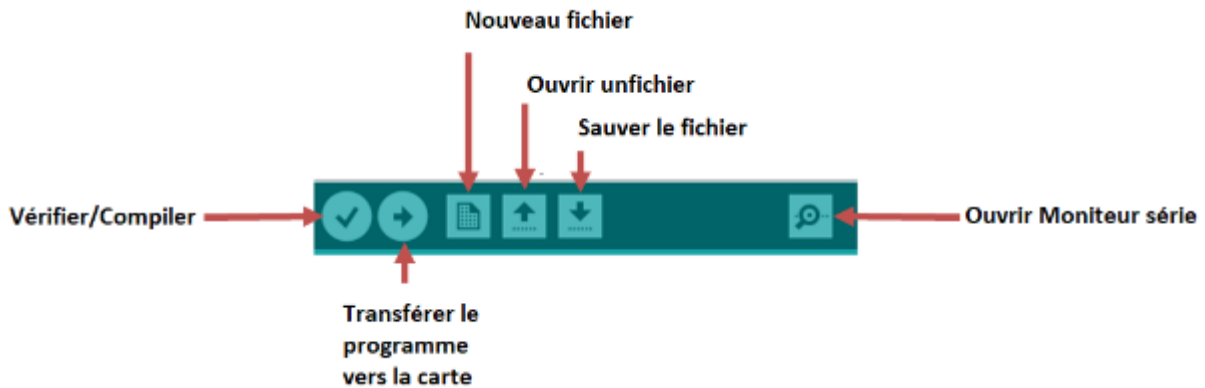


Figure 27: barre d'icône de l'IDE

- La zone d'édition: l'endroit où le code source doit être saisi, composé de 3 parties

. Partie en-tête: permet la déclaration et l'initialisation des variables et des constantes, elle permet aussi d'intégrer des bibliothèques utilisées dans le programme.

. Partie initialisation: permet d'initialisation des E/S, la fonction `setup()` est la première fonction qui va être exécutée au démarrage de l'Arduino et permet de paramétrer les broches en entrée ou sorties, elle s'exécute une seule fois après le démarrage.

. Partie d'exécution en boucle: La fonction `loop()` est la 2ème fonction à être lancée par Arduino après la fonction `setup()`. Cette fonction sera répétée indéfiniment, le code de cette dernière est exécuté en boucle, une fois terminé, il recommence

Chapitre 2 : Conception du système

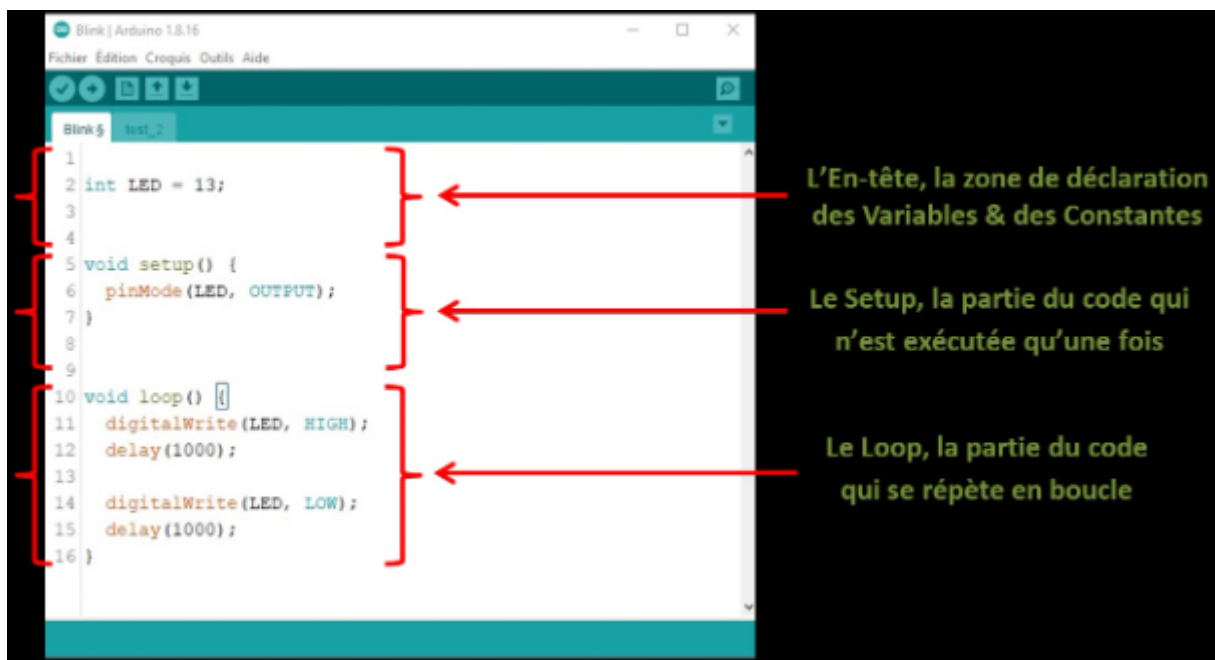


Figure 28: zone d'édition de l'IDE

- La zone d'information et de statut : cette zone affiche divers information sur le programme, les erreurs de compilation, le transfert de programme

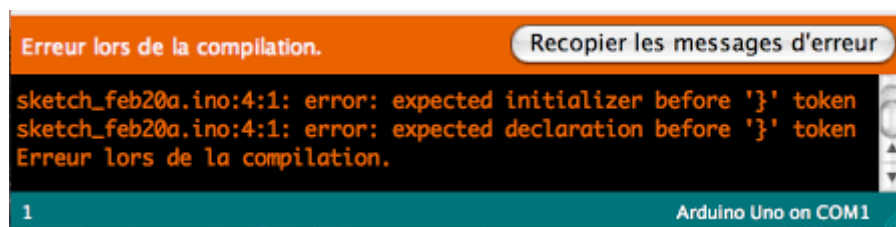


Figure 29: zone de détection d'erreur de l'IDE

- La zone de la barre menu: elle offre diverses options pour travailler avec les codes.
- . Fichier: permet de créer un nouveau fichier, ouvrir un fichier, enregistrer, imprimer, etc.
- . Édition: Comprend des fonctions d'édition de texte telles que Copier, Coller, etc.
- . Croquis: Offre des fonctionnalités pour créer de nouveaux croquis, ajouter des bibliothèques, vérifier le code pour les erreurs, téléverser le code vers les cartes, etc.
- . Outils: Permet de configurer la carte utilisée, choisir le port de communication, gérer les bibliothèques, accéder au moniteur série pour le débogage, etc.
- . Aide: Fournit un accès à la documentation de l'IDE ,et à d'autres ressources utiles.

Chapitre 2 : Conception du système

Fichier Édition Croquis Outils Aide

Figure 30: zone barre menu de l'IDE

2.4.2 Présentation de l'application Telegram :

Telegram est une application et un service de messagerie instantanée multiplateforme hébergé dans un cloud. L'application cliente est gratuite, libre et open source, disponible sur plusieurs plateformes, y compris Android, iOS, Windows, macOS et Linux. Elle est également accessible via un navigateur web grâce à Telegram Web. Les utilisateurs peuvent échanger messages, photos, vidéos et documents sans limite de taille.



Figure 31: logo de l'application Telegram

2.4.2.1 Caractéristiques Principales de Telegram :

- ⑩ **Cryptage** : Les messages sont chiffrés en utilisant une combinaison de cryptage AES symétrique à 256 bits, de cryptage RSA à 2048 bits et de l'échange de clés sécurisé Diffie-Hellman.
- ⑩ **Chats Secrets** : Les chats secrets utilisent un chiffrement de bout en bout et ne laissent aucune trace sur les serveurs de Telegram. Les messages peuvent être programmés pour s'autodétruire automatiquement après un certain temps.
- ⑩ **Confidentialité** : Les utilisateurs peuvent choisir qui peut voir leur numéro de téléphone, leur dernière connexion et leur photo de profil.

Chapitre 2 : Conception du système

- ⑩ **Messages et Médias** : Tous les messages, photos, vidéos et fichiers partagés sont stockés dans le cloud et accessibles depuis n'importe quel appareil connecté à Internet.
- ⑩ **Synchro Multiplateforme** : Les utilisateurs peuvent accéder à leurs messages et fichiers depuis différents appareils simultanément (smartphones, tablettes, ordinateurs).
- ⑩ **Groupes et Supergroupes** : Les utilisateurs peuvent créer des groupes pouvant accueillir jusqu'à 200 000 membres.
- ⑩ **Canaux** : Les canaux permettent aux utilisateurs de diffuser des messages à un nombre illimité d'abonnés.
- ⑩ **Bots** : Telegram supporte des bots qui peuvent automatiser des tâches, fournir des services, et interagir avec les utilisateurs via des commandes.
- ⑩ **Types de Fichiers** : Telegram permet le partage de pratiquement tous les types de fichiers, y compris des documents, des archives, des fichiers multimédias, etc.
- ⑩ **Limite de Taille** : Les fichiers individuels peuvent aller jusqu'à 2 Go.
- ⑩ **Appels Vocaux et Vidéo** : Telegram offre des appels vocaux et vidéo avec un chiffrement de bout en bout.
- ⑩ **Messages Vocaux et Vidéo** : Les utilisateurs peuvent envoyer des messages vocaux et vidéo instantanément.
- ⑩ **APIs et Bots** : Telegram propose une API puissante pour les développeurs, permettant l'intégration avec d'autres services et la création de bots sophistiqués.

2.4.2.2 Les bots telegram:

Les bots sont des programmes informatiques qui s'exécutent entièrement dans l'application Telegram. Les utilisateurs interagissent avec les bots via des interfaces flexibles pouvant prendre en charge tout type de tâche ou de service.

Les bots Telegram sont des comptes spéciaux qui ne nécessitent pas de numéro de téléphone pour être configurés. Les bots sont connectés au serveur de leur propriétaire, qui traite les entrées et les demandes des utilisateurs.

2.4.3 Le logiciel Fritzing :

Fritzing est un logiciel de conception de circuits électroniques qui permet aux utilisateurs de créer des schémas de circuits imprimés (PCB) et de les documenter de manière visuelle. Développé

Chapitre 2 : Conception du système

initialement par le Interaction Design Lab de l'Université des sciences appliquées de Potsdam, Fritzing est largement utilisé par les amateurs d'électronique, les enseignants, et les professionnels pour prototyper des projets et partager des conceptions.

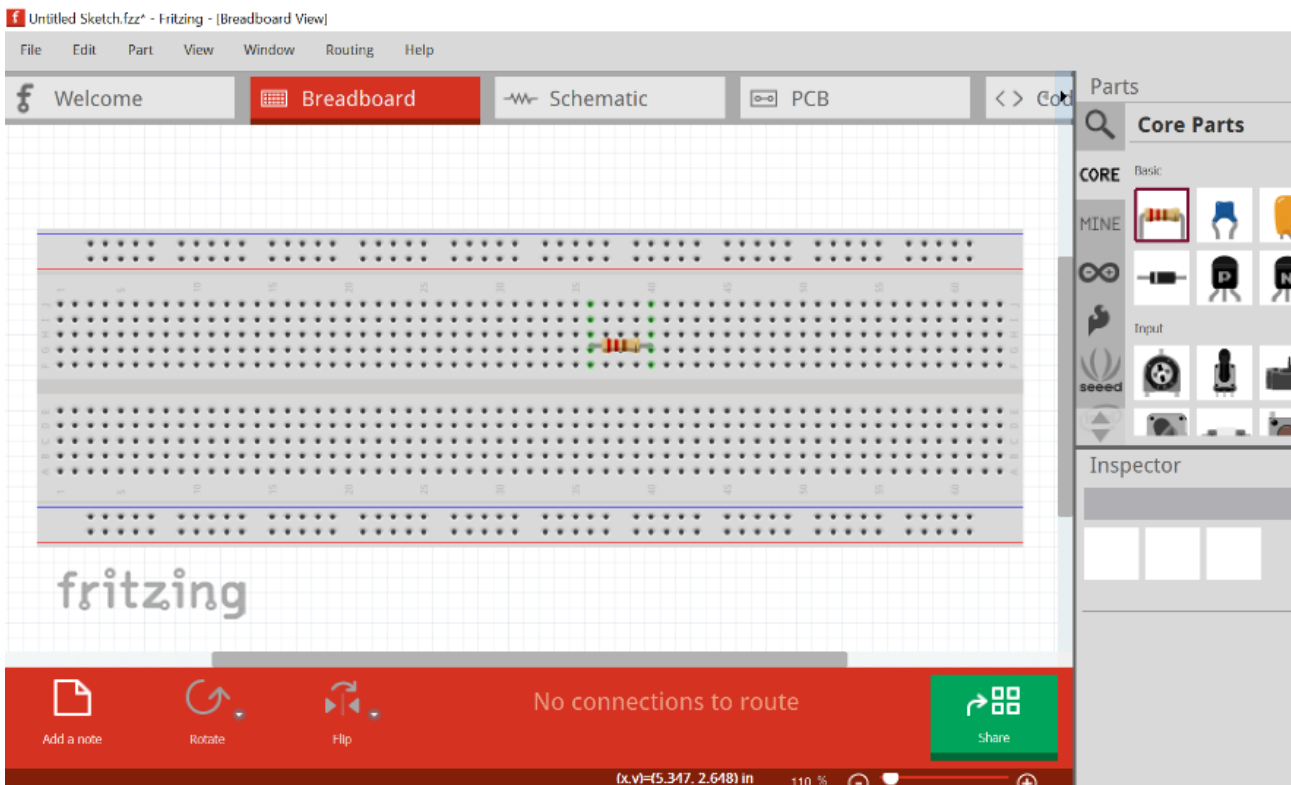


Figure 32: logiciel Fritzing

2.5 Discussion :

Dans ce chapitre l'utilisation des différents matériels et logiciels, nous a permis de concevoir notre système de télésurveillance qui est basé sur diverses cartes d'Espressif Systems.

La conception d'un système électronique est une étape indispensable pour assurer le bon fonctionnement du projet

Chapitre 3 :
Réalisation
pratique

Chapitre 3 : Réalisation pratique

3.1 Préambule:

Ce chapitre présente en détail la réalisation du système de télésurveillance. Dans les sections suivantes, nous décrirons en détail les fonctionnalités du système et les connexions matérielles . Nous présenterons également la configuration logicielle et les tests effectués pour garantir son bon fonctionnement. Enfin, nous concluons par une discussion.

3.2 Réalisation Matérielle :

La réalisation Matérielle de notre système est divisée en deux parties : une avec l'ESP32-CAM et l'autre avec l'ESP32. Chacun de ces modules joue un rôle crucial dans le fonctionnement global du système et sont tous deux connectés à Telegram pour faciliter la communication et le contrôle à distance.

3.2.1 Partie ESP32-CAM:

La première partie permet de créer un système de surveillance en utilisant une caméra ESP32-CAM et un capteur de mouvement PIR (Passive Infrared Sensor) ainsi qu'une carte sd. Voici ses fonctionnalités:

Détection de Mouvement :

- ⑩ Utilise un capteur PIR pour détecter toute activité ou mouvement dans la zone surveillée.
- ⑩ Déclenche des actions lorsque le mouvement est détecté.

Capture de Photos :

- ⑩ Prend automatiquement des photos à l'aide de la caméra ESP32-CAM lorsque le capteur PIR détecte un mouvement.
- ⑩ Les images capturées peuvent être stockées localement sur une carte SD.

Envoi de Notifications :

- ⑩ Envoie les photos capturées à un bot Telegram, permettant à l'utilisateur de recevoir des alertes et de visualiser les images en temps réel.
- ⑩ Permet aux utilisateurs autorisés de demander manuellement des photos via des commandes Telegram.

Interaction via Telegram :

Chapitre 3 : Réalisation pratique

- ⑩ Le bot Telegram peut recevoir des commandes de l'utilisateur pour prendre une photo sur demande.
- ⑩ Gère les messages et les commandes pour permettre une interaction en temps réel avec le système de surveillance

3.2.1.1 les connexions de l'ESP32-CAM:

- Branchement des Pins de la Caméra:

L'ESP32-CAM AI-Thinker est un module de caméra compact et polyvalent basé sur le microcontrôleur ESP32. Il est équipé d'une caméra OV2640 capable de capturer des images haute résolution et de les transmettre via WiFi.

Tableau 1: Branchement des Pins de la Caméra

Fonction	Pin ESP32-CAM AI-Thinker	Description
PWDN	GPIO 32	Power Down
RESET	-1 (non connecté)	Reset
XCLK	GPIO 0	External Clock
SIOD	GPIO 26	Serial Data
SIOC	GPIO 27	Serial Clock
Y9	GPIO 35	Data Y9
Y8	GPIO 34	Data Y8
Y7	GPIO 39	Data Y7
Y6	GPIO 36	Data Y6
Y5	GPIO 21	Data Y5
Y4	GPIO 19	Data Y4
Y3	GPIO 18	Data Y3
Y2	GPIO 5	Data Y2
VSYNC	GPIO 25	Vertical Sync
HREF	GPIO 23	Horizontal Reference
PCLK	GPIO 22	Pixel Clock

Chapitre 3 : Réalisation pratique

- Branchement des Pins de la Carte SD:

La carte SD est un support de stockage amovible utilisé pour sauvegarder des données de manière fiable. Dans le contexte de l'ESP32-CAM, la carte SD est utilisée pour stocker les images capturées par la caméra. Cela permet de conserver les données localement, offrant une méthode de sauvegarde et d'accès aux fichiers sans dépendre uniquement de la transmission sans fil.

Tableau 2: Branchement des Pins de la Carte SD

Fonction	Pin ESP32-CAM AI-Thinker	Description
SD_CMD	GPIO 15	Command Line
SD_CLK	GPIO 14	Clock Line
SD_DATA0	GPIO 2	Data Line 0
SD_DATA1	GPIO 4	Data Line 1
SD_DATA2	GPIO 12	Data Line 2
SD_DATA3	GPIO 13	Data Line 3

- Branchement des Pins du capteur de mouvement:

Le schéma ci-dessous illustre la connexion du capteur de mouvement à la carte ESP32-CAM. Les connexions sont établies comme suit :

- ⑩ Le VCC du capteur PIR est connecté au 5V de l'ESP32.
- ⑩ Le GND du capteur PIR est connecté à la masse (GND) de l'ESP32.
- ⑩ Le signal OUT du capteur PIR est relié au GPIO 12 de l'ESP32.

Chapitre 3 : Réalisation pratique

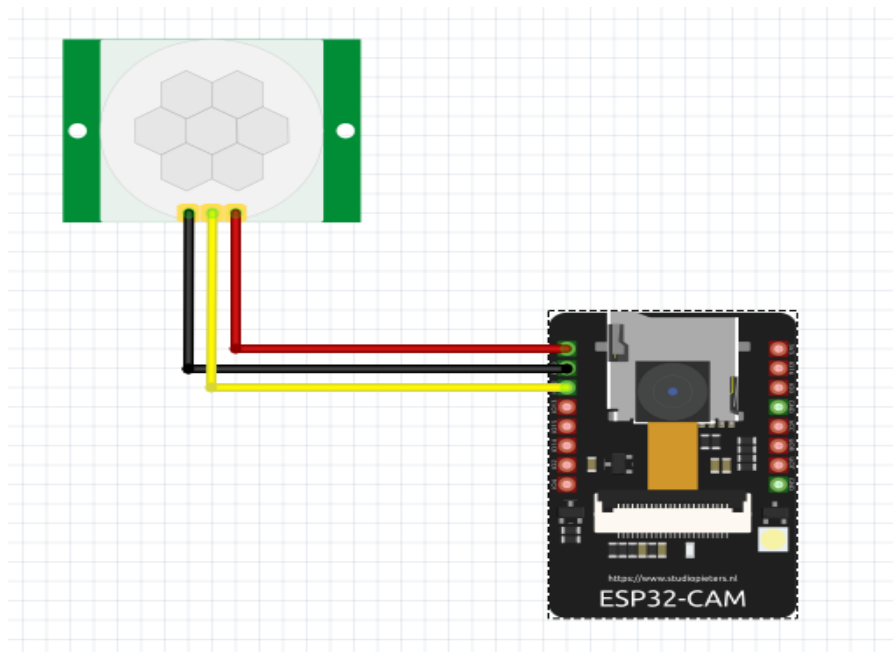


Figure 33: Branchement des Pins du capteur de mouvement

3.2.2 Partie ESP32:

La deuxième partie du système s'étend pour intégrer des fonctionnalités avancées de surveillance et de contrôle d'accès, en exploitant un ensemble diversifié de capteurs et d'actionneurs. Voici en détail les fonctionnalités principales :

3.2.2.1 Contrôle d'Accès via RFID :

- ⑩ Utilise un module RFID pour lire les cartes et les badges.
- ⑩ Compare l'UID des badges lus à une liste d'UID autorisés pour accorder ou refuser l'accès.
- ⑩ Contrôle un servo-moteur pour ouvrir ou fermer une porte en fonction de l'autorisation.
- ⑩ Une led verte s'allume pendant deux secondes lorsque l'accès est autorisé et une led rouge s'allume pendant une seconde lorsque l'accès n'est pas autorisé,
- ⑩ Les notifications sont envoyées via Telegram pour informer des événements d'accès autorisé ou non autorisé, assurant ainsi une sécurité renforcée.

- Connexion du contrôle d'Accès:

Connexion du module RFID MFRC522 à l'ESP32 :

- ⑩ La broche SS/SDA (slave select) est connectée à la pin 5.
- ⑩ La broche SCL/ SCK (Serial Clock) est connectée à la pin 18.
- ⑩ La broche MOSI (master output slave input) est connectée à la pin 23.

Chapitre 3 : Réalisation pratique

- ⑩ La broche MISO (master input slave output)) est connectée a la pin 19.
- ⑩ La broche IRQ n'est pas connectée.
- ⑩ La broche GND est connectée à GND.
- ⑩ La broche RST est connectée a la pin 27
- ⑩ La broche 3,3 est connectée à 3,3

Connexion du Servo-Moteur à l'ESP32 :

- ⑩ Signal (Contrôle) du servo à GPIO14 de l'ESP32.
- ⑩ Alimentation (VCC) du servo à 5V de l'ESP32.
- ⑩ Terre (GND) du servo à GND de l'ESP32.

Connexion des leds à l'ESP32 :

- ⑩ L'anode de la LED Rouge à GPIO13 de l'ESP32.
- ⑩ L'anode de la LED Verte à GPIO12 de l'ESP32.
- ⑩ La cathode des LEDs a GND de l'ESP32.

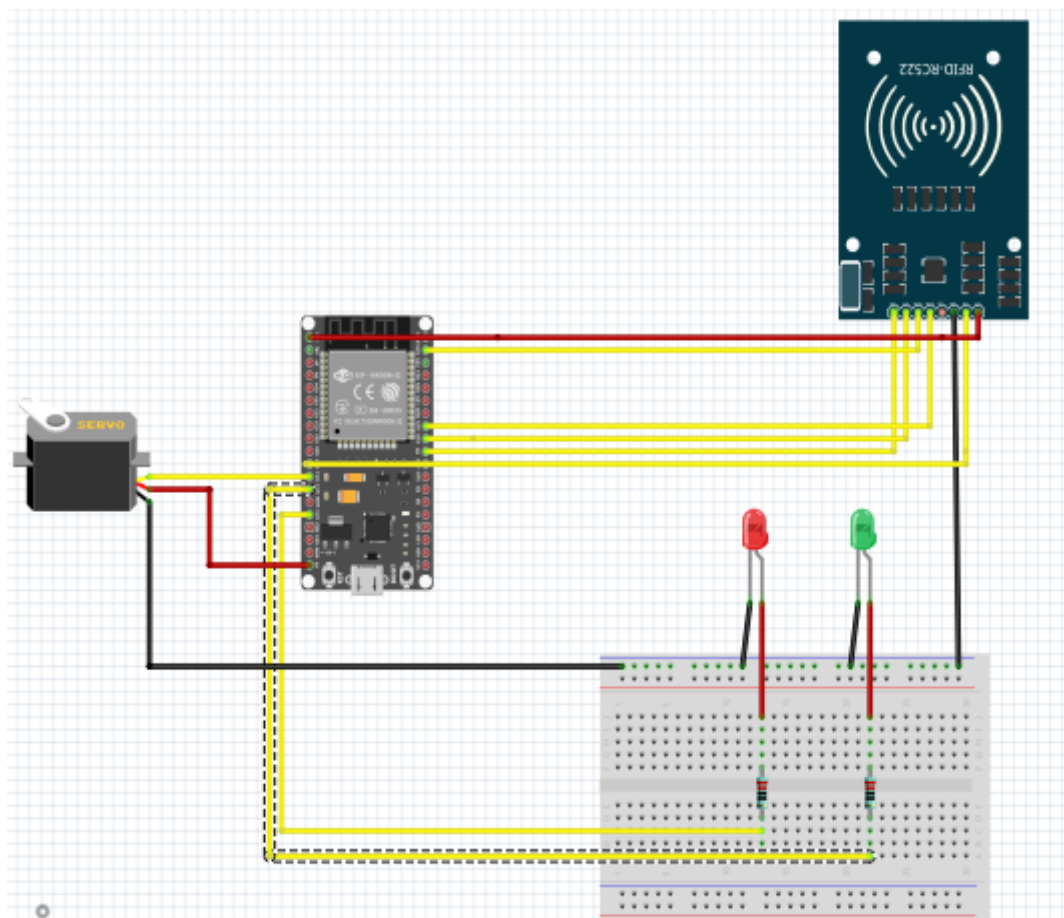


Figure 34: Connexion du contrôle d'Accès

Chapitre 3 : Réalisation pratique

3.2.2.2 Détection de Gaz :

- ⑩ Utilise un capteur de gaz pour surveiller la concentration de gaz dans l'environnement.
- ⑩ Envoie des alertes via Telegram si la concentration de gaz dépasse un seuil prédéfini.
- ⑩ Active un buzzer pour une période de 5 seconde si la concentration de gaz dépasse un seuil prédéfini.

- Connexion pour détection de Gaz :

Connexion du capteur MQ135 à l'ESP32 :

- ⑩ Broche analogique du capteur de gaz A0 à GPIO34 de l'ESP32.
 - ⑩ Alimentation (VCC) du capteur de gaz à 5V de l'ESP32.
 - ⑩ Terre (GND) du capteur de gaz à GND de l'ESP32.
- Connexion du buzzer à l'ESP32 :
- ⑩ GND du buzzer à GND de l'ESP32.
 - ⑩ Buzzer à GPIO26 de l'ESP32.

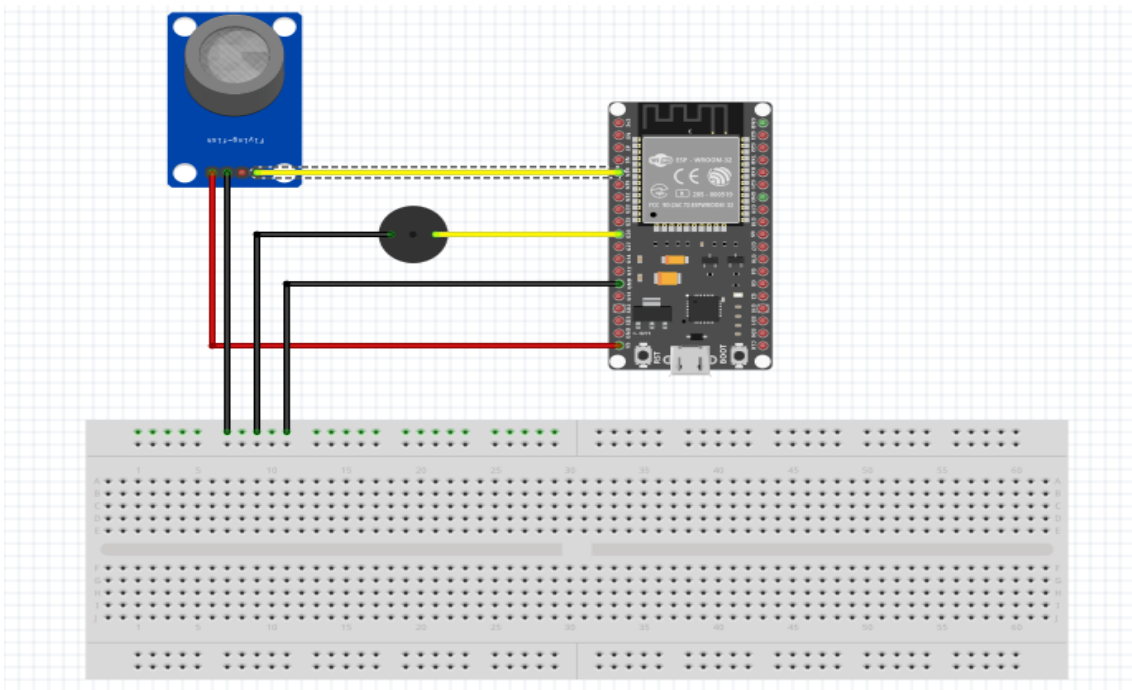


Figure 35: Connexion pour détection de Gaz

3.2.2.3 Détection de Mouvement :

- ⑩ Utilise des capteurs PIR pour détecter tout mouvement dans la zone surveillée.
- ⑩ Envoie des notifications via Telegram en cas de détection de mouvement.

Chapitre 3 : Réalisation pratique

- Connexion pour détection de de Mouvement PIR :

- ⑩ VCC des capteurs PIR à 5V de l'ESP32-CAM.
- ⑩ GND des capteurs PIR à GND de l'ESP32-CAM.
- ⑩ OUT du capteur PIR1 à GPIO32 de l'ESP32-CAM.
- ⑩ OUT du capteur PIR2 à GPIO33 de l'ESP32-CAM.
- ⑩ OUT du capteur PIR3 à GPIO25 de l'ESP32-CAM.

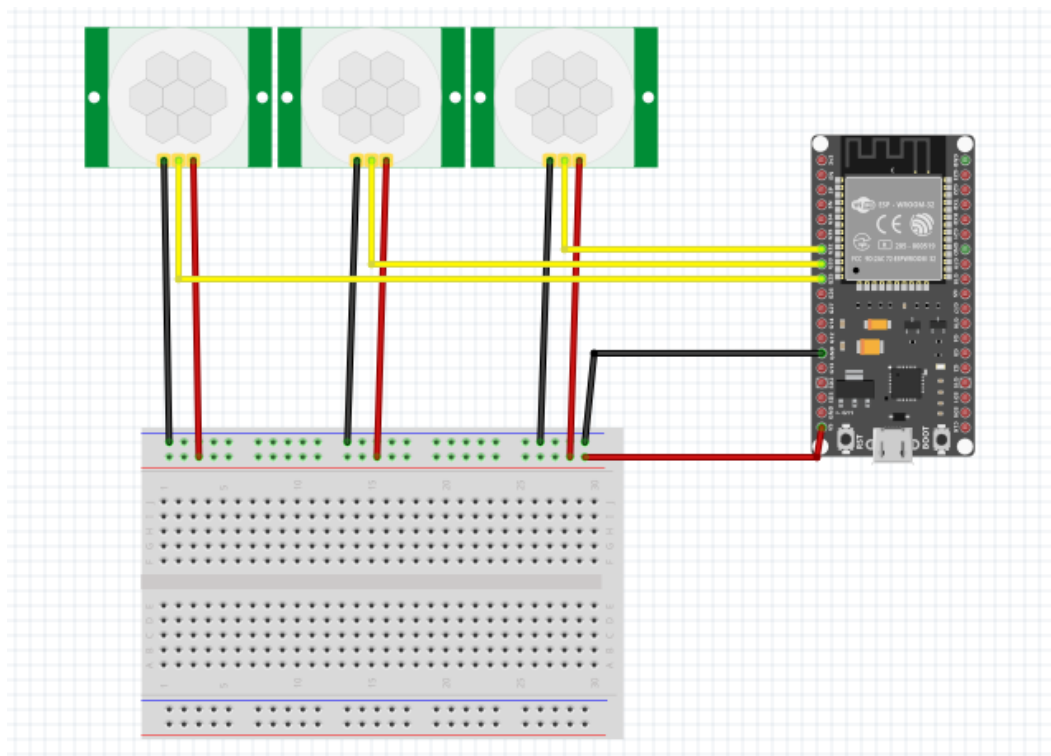


Figure 36: Connexion pour détection de de Mouvement PIR

3.2.2.4 Interaction et Commandes via Telegram :

Permet aux utilisateurs d'envoyer des commandes au bot Telegram pour contrôler divers aspects du système, tels que l'allumage ou l'extinction des LEDs et l'activation d'un buzzer. Lorsque l'utilisateur envoie le message "/go" sur Telegram, un message de bienvenue informe celui-ci des commandes disponibles pour contrôler différents éléments du système connecté à l'ESP32. Ces commandes comprennent :

- ⑩ /ledR : Permet d'allumer ou d'éteindre la LED rouge.
- ⑩ /ledV : Permet d'allumer ou d'éteindre la LED verte.
- ⑩ /ledO : Permet d'allumer ou d'éteindre la LED orange.

Chapitre 3 : Réalisation pratique

- ⑩ /ledA : Active ou désactive un mode de LED aléatoire.
- ⑩ /buzzer : Active ou désactive le buzzer.

En appuyant sur "/ledR" et/ou "/ledV" et/ou "/ledO" et/ou "/buzzer", les LEDs correspondantes et le buzzer s'activeront. En rappuyant sur ces commandes, les LEDs et le buzzer se désactiveront, pour la commande "/ledA" allume les leds de manière aléatoire toute les une seconde afin de simuler une présence,

- Connexion des leds et du buzzer à l'ESP32 :

- ⑩ GND du buzzer à GND de l'ESP32.
- ⑩ Buzzer à GPIO15 de l'ESP32.
- ⑩ L'anode de la LED Rouge à GPIO4 de l'ESP32.
- ⑩ L'anode de la LED Verte à GPIO2 de l'ESP32.
- ⑩ L'anode de la LED Orange à GPIO16 de l'ESP32.
- ⑩ La cathode des LEDs a GND de l'ESP32.

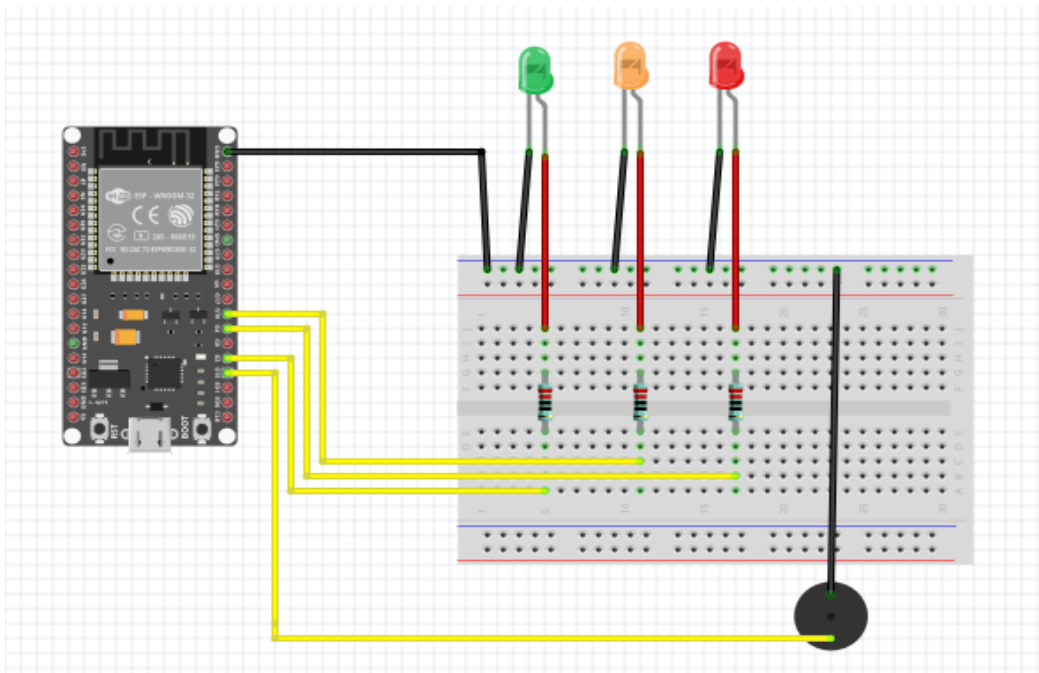


Figure 37: Connexion des leds et du buzzer à l'ESP32

Ces deux parties illustrent les capacités polyvalentes de l'ESP32 et l'ESP32-CAM pour créer des systèmes de surveillance et de sécurité robustes, intégrant plusieurs types de capteurs et actionneurs et permettant une interaction et un contrôle à distance via un bot Telegram. La figure ci-dessous représente le schéma général du montage de notre système.

Chapitre 3 : Réalisation pratique

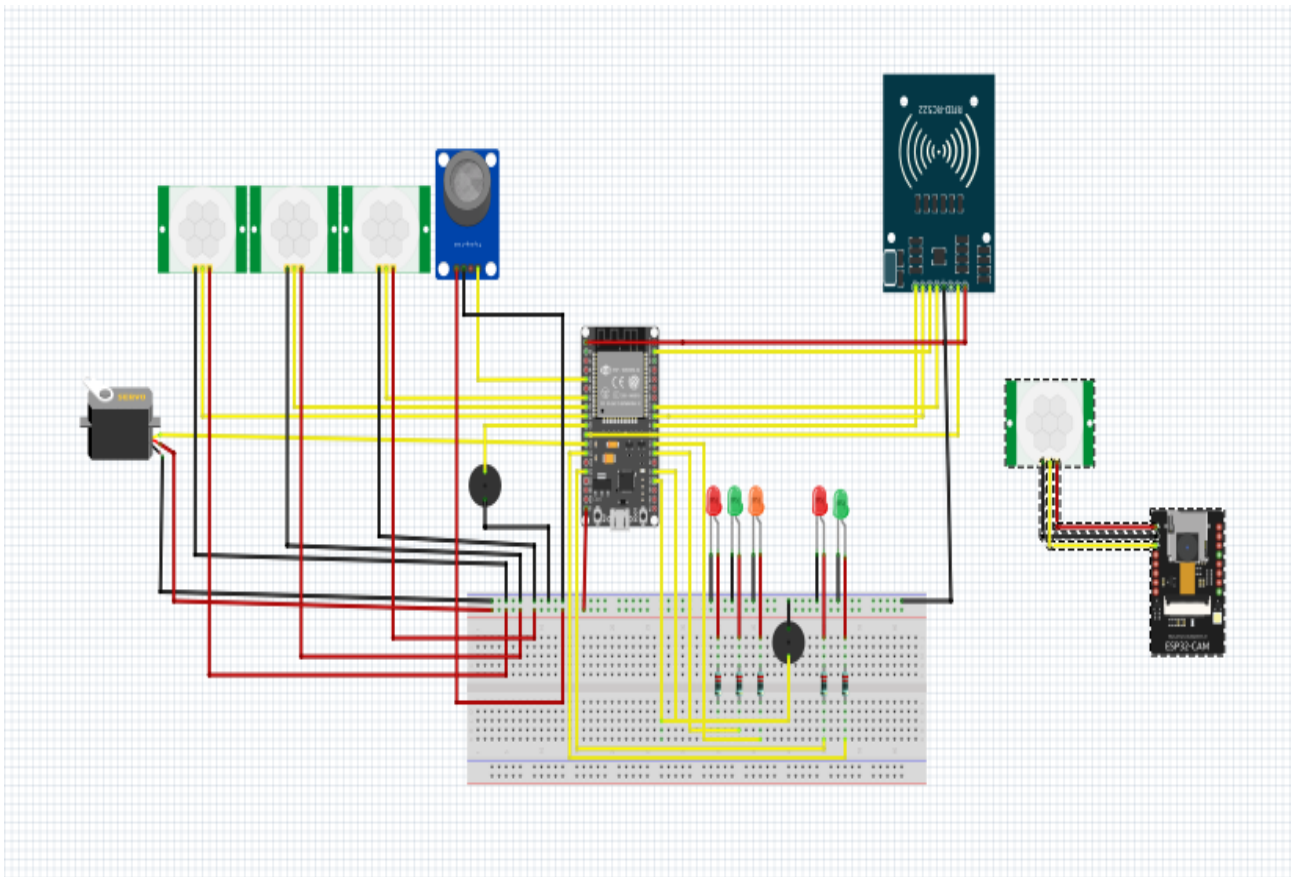


Figure 38: schéma général du montage de notre système.

3.3 Configuration logicielle :

3.3.1 Configuration des cartes ESP32 dans l'Arduino IDE :

Les cartes du fabricant "Espressif Systems" ne sont pas gérées nativement par l'environnement de développement Arduino. Il faut donc procéder à son installation. Pour cela, dans le menu "Fichier", choisir "Préférences", puis dans la zone de saisie "URL de gestionnaire de cartes supplémentaires" renseigner l'URL des cartes de la famille ESP32 dont l'ESP32-CAM fait partie

L'URL:

https://espressif.github.io/arduino-esp32/package_esp32_index.json

Chapitre 3 : Réalisation pratique

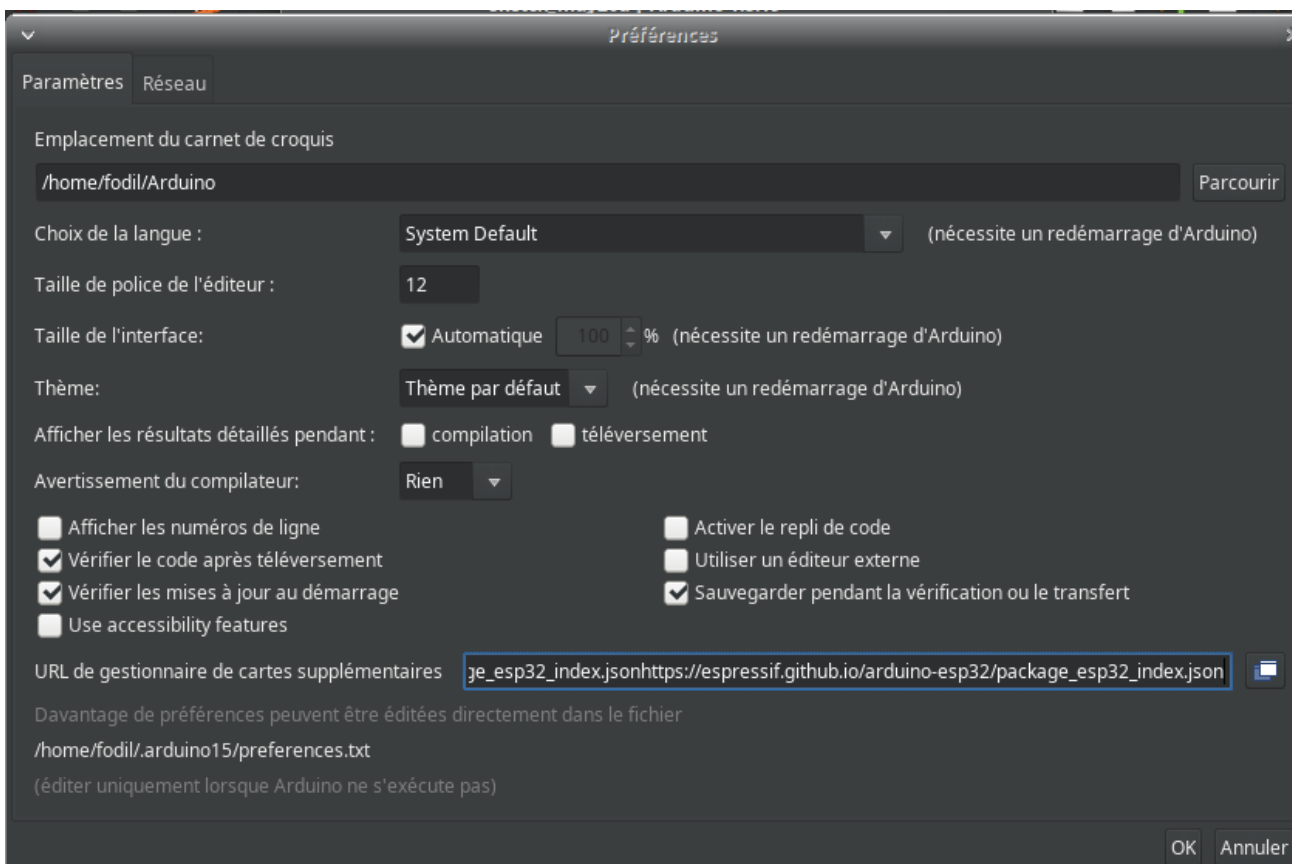


Figure 39: copie de l'URL dans le fichier préférence

Maintenant que l'URL est renseignée, il faut procéder à son installation. Dans la barre menu accéder a "Outils", choisir "type de carte" puis cliquer sur "Gestionnaire de carte".

La fenêtre des gestionnaires de cartes s'ouvre alors. En saisissant "esp32" dans la zone de recherche "esp32 by Espressif Systems" s'affiche, ensuite cliquer sur le bouton "Installer".



Figure 40 : installation de l'ESP32 dans le gestionnaire de carte

Chapitre 3 : Réalisation pratique

Enfin, redémarrer Arduino IDE. Aller ensuite dans ‘‘outils’’ puis ‘‘type de carte’’ il y a toutes les cartes ESP32 : pour l’ESP32-CAM nous allons choisir ‘‘esp32 wrover module’’, et pour l’ESP32 choisir ‘‘esp32 Dev Module’’il ne reste plus qu’a configurer les paramètre des cartes on changeons les paramètre ainsi que de choisir le port adapter et de laisser les autre paramètre a défaut: Upload speed:’’115200’’ et Partition scheme:’’Huge APP (3MB No OTA/1MB SPIFFS)’’ pour l’ESP32-CAM et Upload speed:’’115200’’ pour l’ESP32

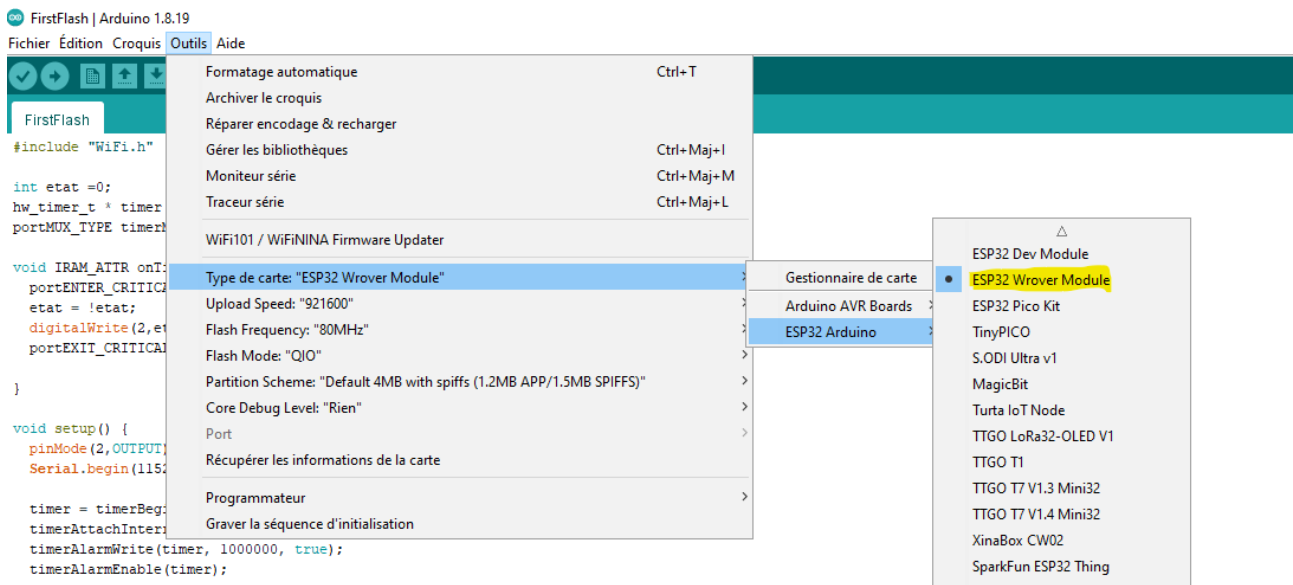


Figure 41 : Choix du type de carte pour ESP32-CAM

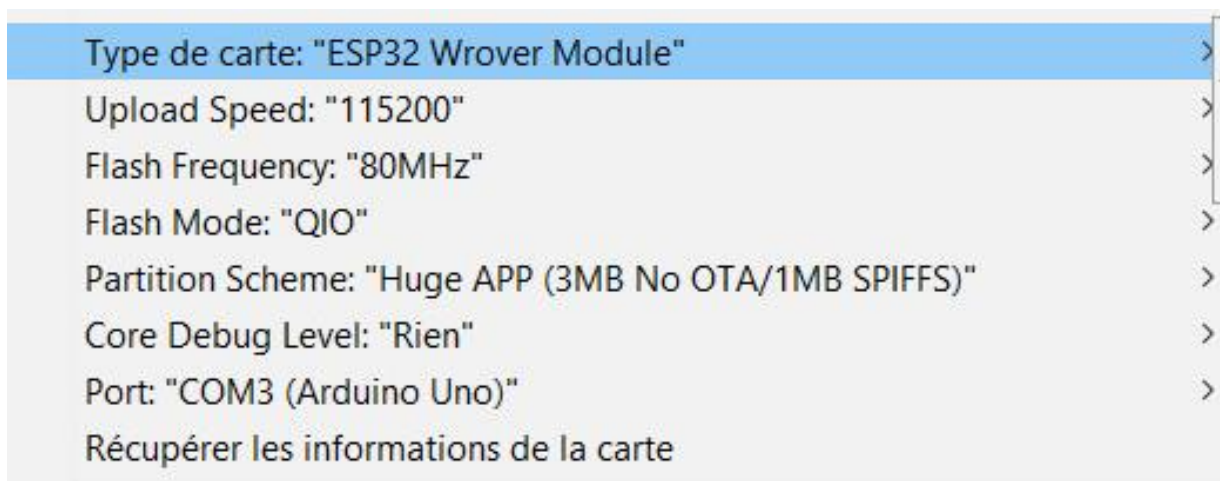


Figure 42 : Configuration des paramètre de la carte ESP32-CAM

Chapitre 3 : Réalisation pratique

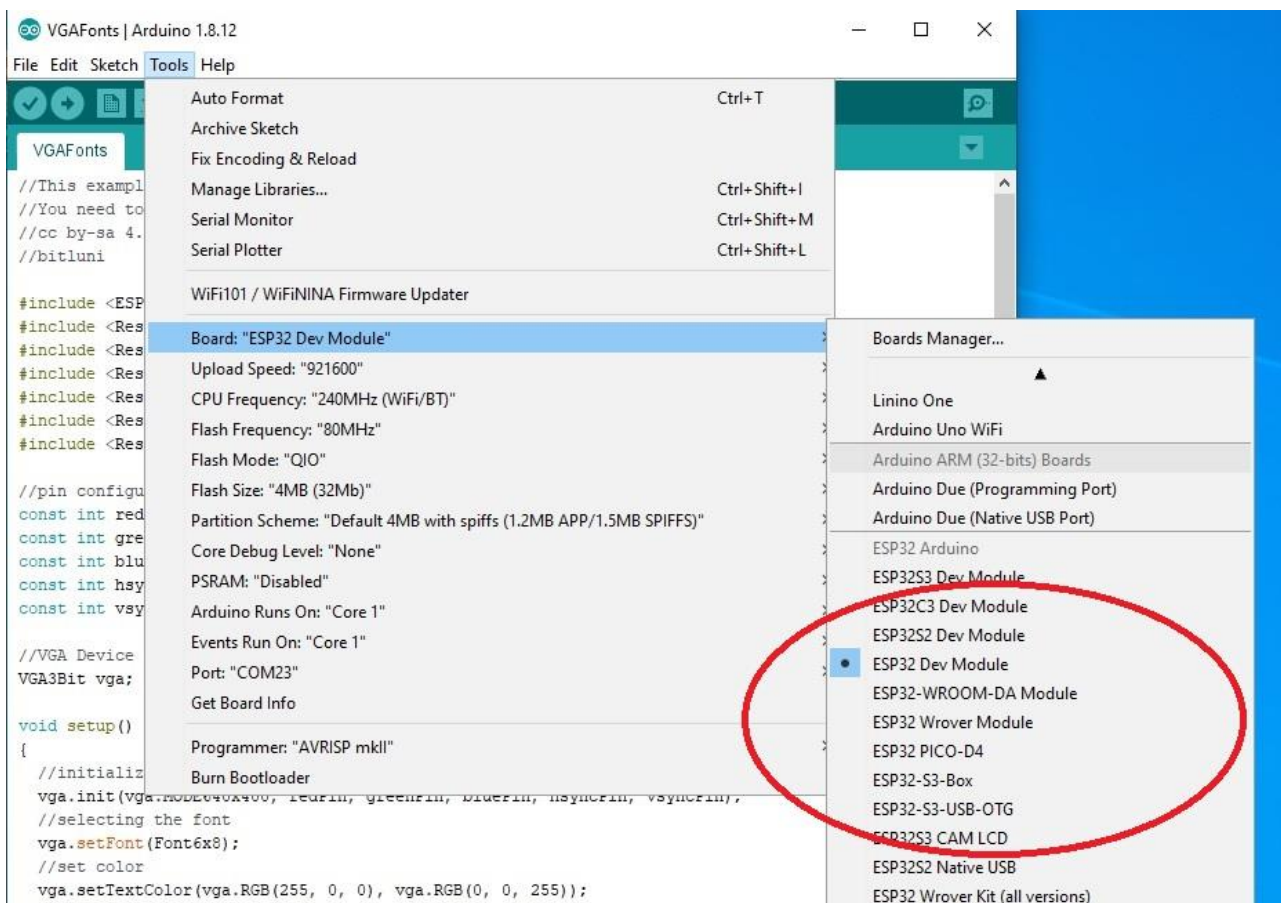


Figure 43: choix du type de carte pour esp32

3.3.2 Configuration de l'application Telegram :

3.3.2.1 Créer un compte Telegram :

- Choisir le support sur lequel utiliser Telegram (smartphone, ordinateur ou sur le web),
- Démarrez l'appli ou le logiciel, puis appuyez sur start- Saisissez le numéro de téléphone portable,
- Un code sera envoyé par SMS. Saisissez ce code à 5 chiffres dans l'application, puis validez,
- Entrez ensuite les informations pour créer le compte,

Chapitre 3 : Réalisation pratique

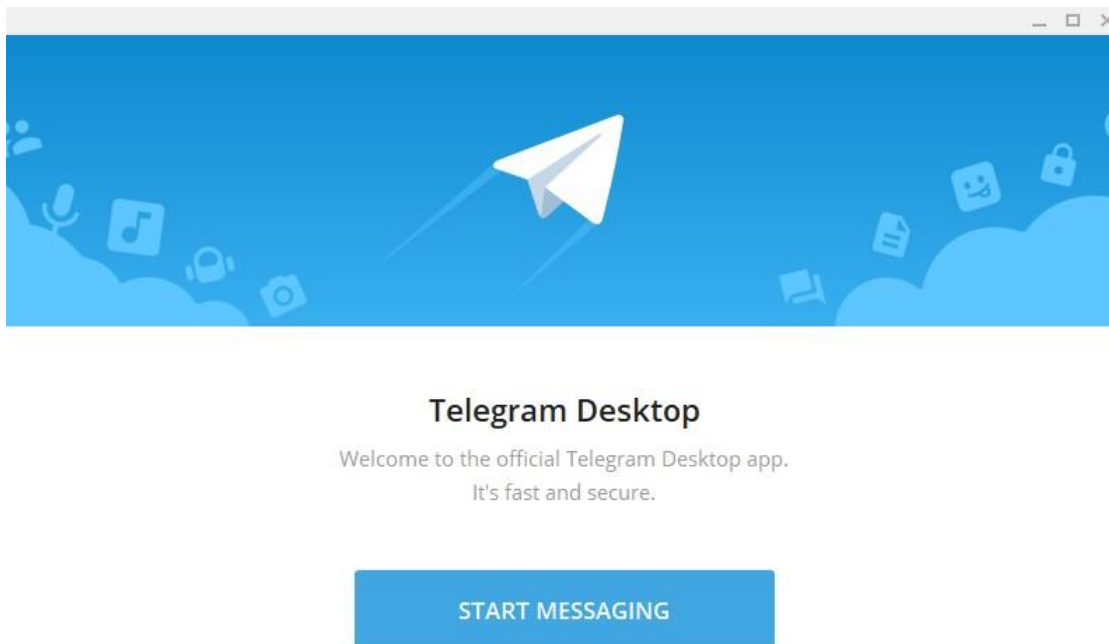


Figure 44 : Interface de l'application telegram pour débuter l'ouverture d'un compte

3.3.2.2 Créer un bot Telegram:

Lancer l'application telegram puis dans la zone "Rechercher", saisir @BotFather puis appuyez sur la touche Entrée.

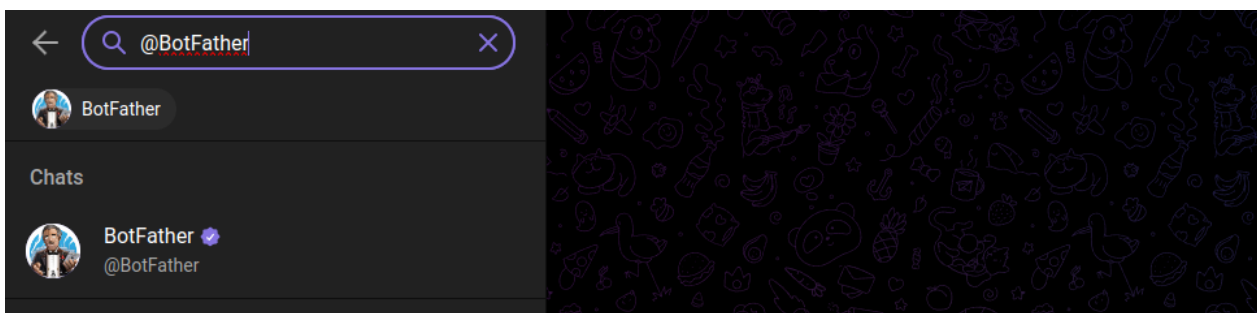


Figure 45: recherche de @BotFather sur telegram

Cliquer sur le bouton Démarrer, puis saisir /start,

Chapitre 3 : Réalisation pratique

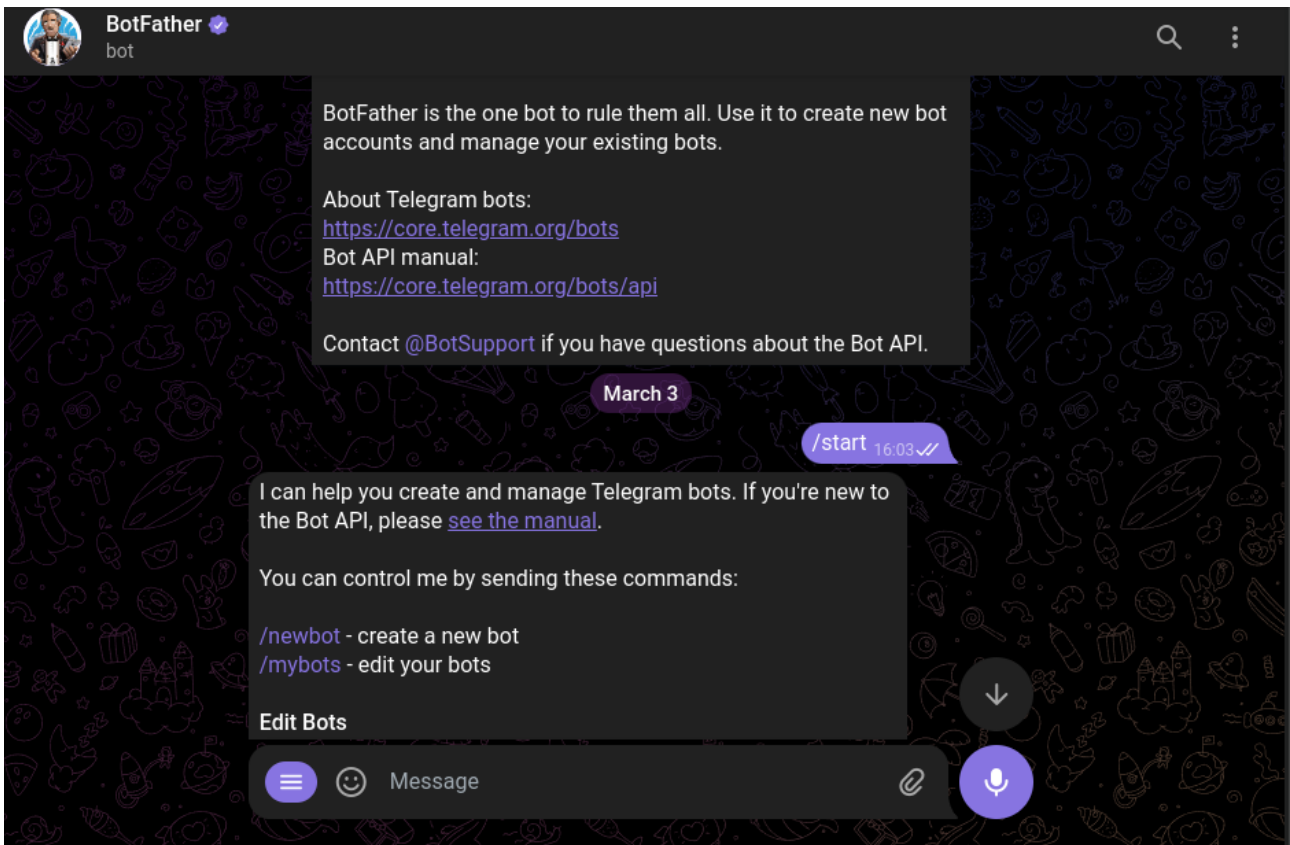


Figure 46: démarrage de la création du bot

Saisir /newbot pour demander au Bot de Telegram de créer un nouveau Bot, BotFather va demander de donner au bot un nom. Ce sera le nom affiché dans la discussion, exemple «ESP32CAMPIR» Ensuite, il faudra indiquer un nom "public", qui permettra de le retrouver dans l'annuaire. Il faut trouver un nom qui sera unique. Attention, il faut que le nom se termine par "bot" pour bien identifier qu'il s'agit d'un robot, exemple «ESP32CAMPIRprojet_bot»

Chapitre 3 : Réalisation pratique

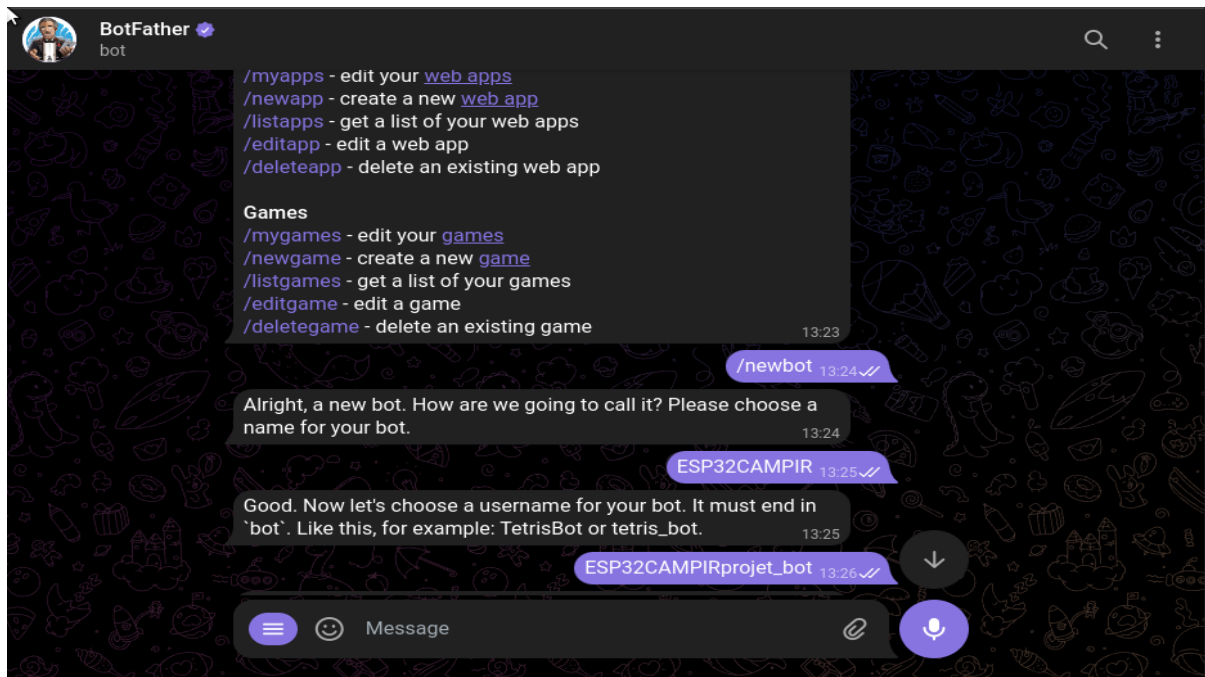


Figure 47: nommer le bot

Si le bot est créé avec succès, un message sera reçue avec un lien pour accéder au bot et au jeton du bot.

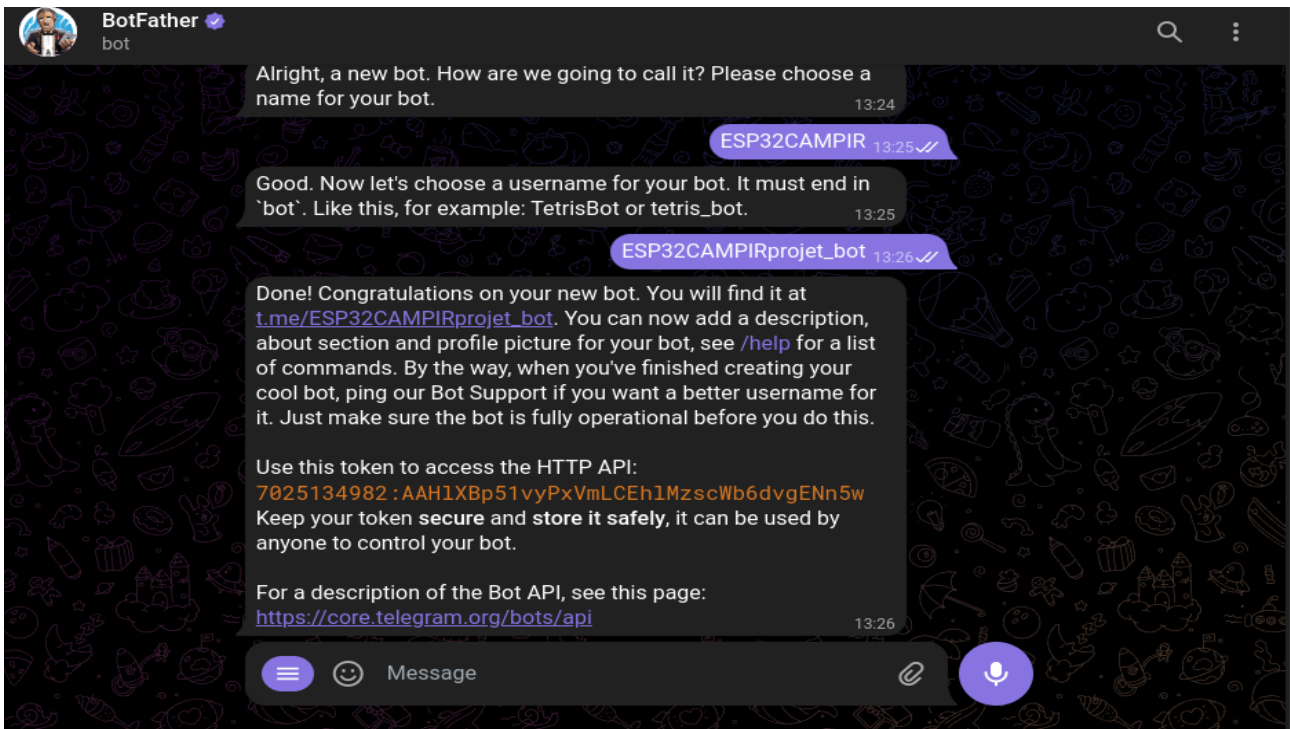


Figure 48: message de réception du jeton du bot

Chapitre 3 : Réalisation pratique

3.3.2.3 Obtenez l'identifiant d'utilisateur Telegram:

Toute personne connaissant le nom d'utilisateur du bot peut interagir avec lui. Pour ignorer les messages qui ne proviennent pas de notre compte Telegram (ou de tout utilisateur autorisé), il faut obtenir un identifiant d'utilisateur Telegram. Ensuite, lorsque le bot Telegram reçoit un message, l'ESP peut vérifier si l'identifiant de l'expéditeur correspond à votre l'identifiant utilisateur et traiter le message ou l'ignorer.

Dans l'application Telegram, saisir «IDBot» puis appuyez sur la touche Entrée.

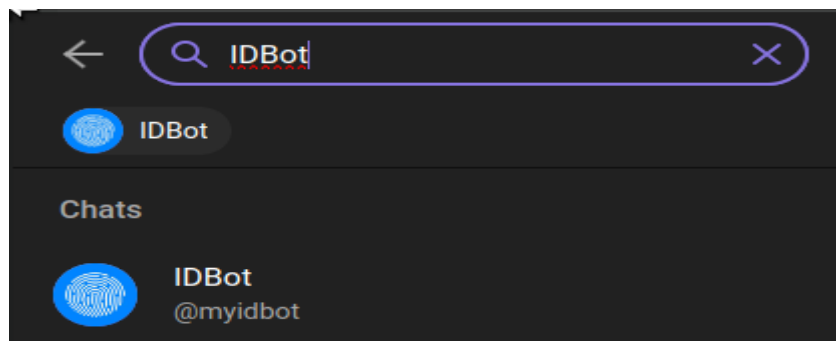


Figure 49: recherche de IDBot sur telegram

Cliquez sur Démarrez puis saisir /start puis /getid. Une réponse avec l'identifiant d'utilisateur sera envoyer.

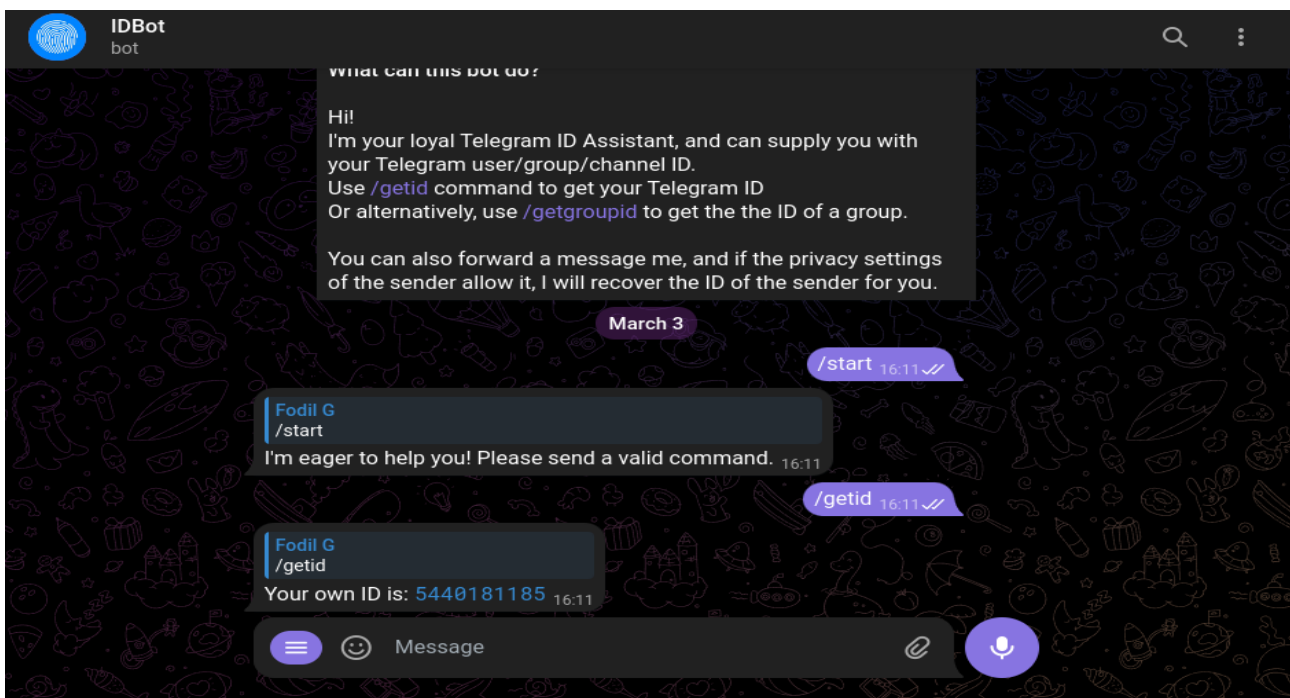
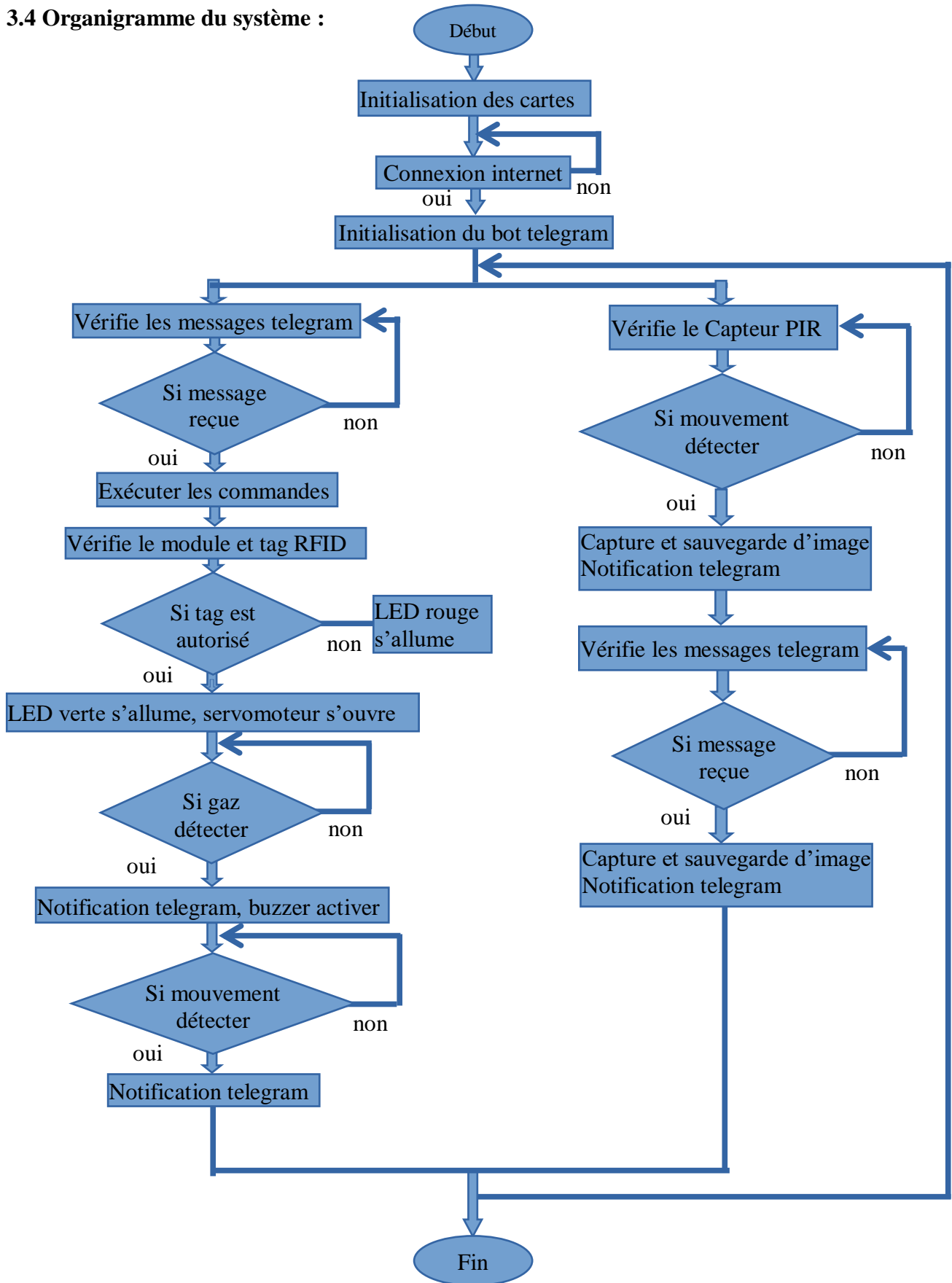


Figure 50: réception de l'identifiant

Chapitre 3 : Réalisation pratique

3.4 Organigramme du système :



Chapitre 3 : Réalisation pratique

3.5 Teste du projet:

- Test de l'ESP32-CAM:

nous avons saisi le message /start dans le bot telegram puis nous avons appuyez sur /photo pour faire une capture et une sauvegarde d'image

```
Réponse reçue
Gérer les nouveaux messages : 1
/start
Réponse reçue
Gérer les nouveaux messages : 1
/photo
Nouvelle demande de photo
Préparation de la photo
État de flash réglé sur HAUT
Photo sauvegardée: /photo1.jpg
Connexion à api.telegram.org
Connexion réussie
..{"ok":true,"result":{"message_id":
État de flash réglé sur BAS
Photo sauvegardée: /photo2.jpg
```

Figure 52: capture d'image du port serie



Figure 51: capture d'image du bot telegram

Les captures indique que l'essai est correcte

Si un mouvement est fait en face du capteur pir des photos nous serons envoyer et sauvegarder comme l'indique les figure si-dessous:

```
Mouvement détecté !
Envoi de la photo à Telegram
Connexion à api.telegram.org
Connexion réussie
...{"ok":true,"result":{"message_id":
Mouvement terminé !
Photo sauvegardée: /photo3.jpg
```

Figure 54: capture d'image du port serie

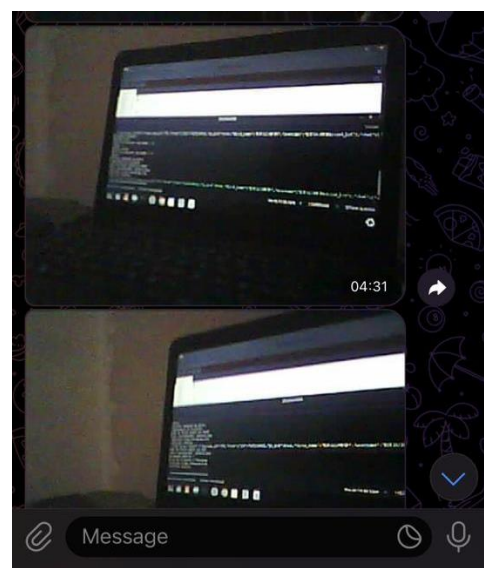


Figure 53: capture d'image du bot telegram

Chapitre 3 : Réalisation pratique

la figure ci-dessous montre le fichier ou sont sauvegarder les photos

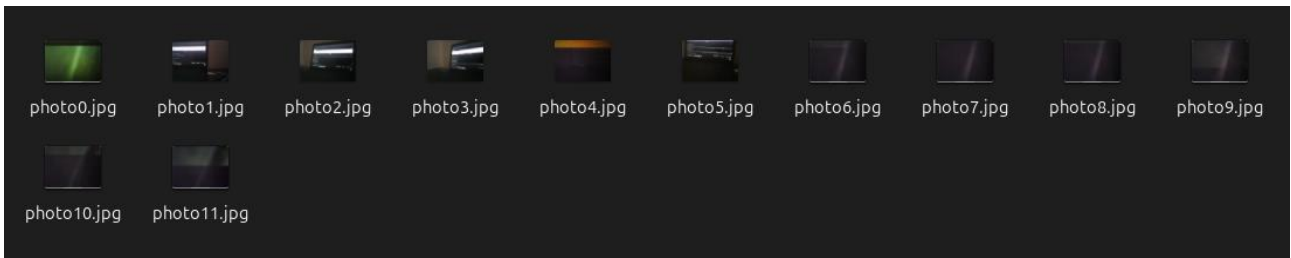


Figure 55: fichier ou sont sauvegarder les photos

- test contrôle d'accès:

lorsque nous présentons le bon badge, le servo-motors se tourne a 90° est la led verte s'allume

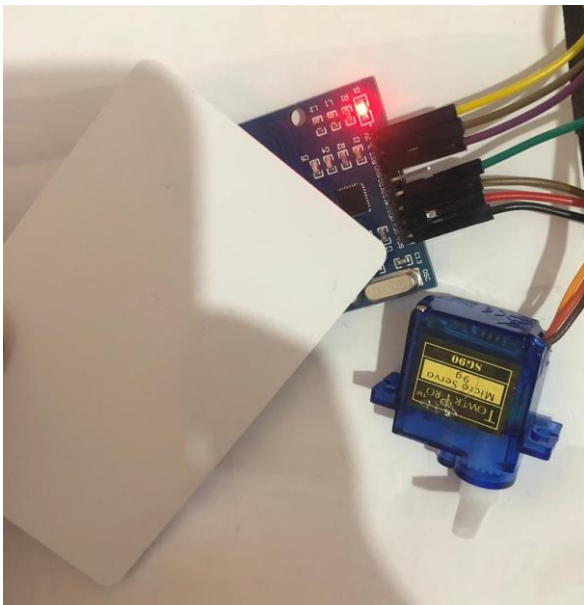


Figure 56: photo du bon badge et du sg90 qui se tourne

```
Acces autorise. Ouverture...  
Card UID: 73 C3 CC FD  
Card SAK: 08  
PICC type: MIFARE 1KB
```

Figure 57: capture d'image du port serie

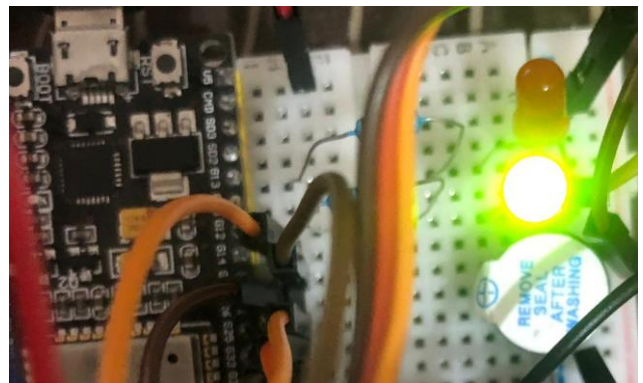


Figure 58: photo de la led verte qui s'allume

lorsque nous présentons le mauvais badge, le servo-motors se tourne a 0° est la led rouge s'allume

Chapitre 3 : Réalisation pratique

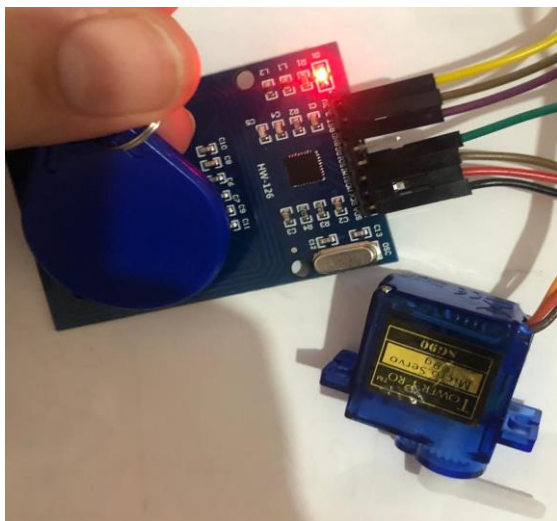


Figure 59: photo du mauvais badge et du sg90 qui se tourne

```
Accès refusé. Mauvais badge.  
Card UID: A3 A1 43 F5  
Card SAK: 08  
PICC type: MIFARE 1KB
```

Figure 60: capture d'image du port serie

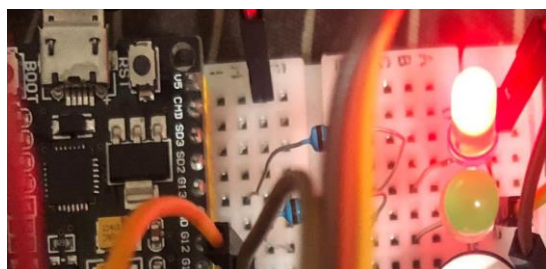


Figure 61: photo de la led rouge qui s'allume

des notifications nous parviennent du bot Telegram nous indiquons si l'accès est autoriser ou non, et si un bon ou mauvais badge est détecter

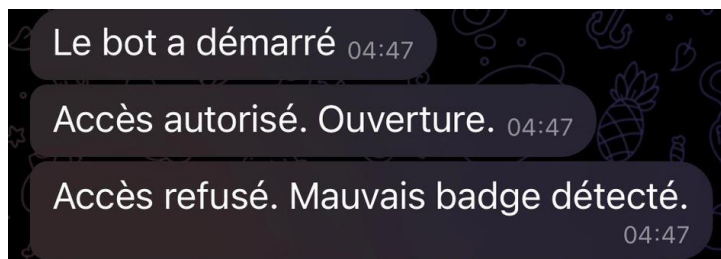


Figure 62: capture d'image du bot telegram

Chapitre 3 : Réalisation pratique

- Test de détection de gaz : Nous allons utiliser un briquet pour faire une simulation de fuite de gaz



Figure 63: photo de simulation de fuite de gaz

```
Valeur MQ135 : 281
Valeur MQ135 : 273
Valeur MQ135 : 266
Valeur MQ135 : 288
Valeur MQ135 : 272
Valeur MQ135 : 272
Valeur MQ135 : 270
Valeur MQ135 : 261
Valeur MQ135 : 259
Valeur MQ135 : 252
Valeur MQ135 : 259
Valeur MQ135 : 256
Valeur MQ135 : 248
Valeur MQ135 : 453
```

Figure 65: capture d'image du port serie



Figure 64: capture d'image du bot telegram

la simulation a marcher comme le montre les figures au-dessus

- test détection de mouvement:

Si un mouvement est fait en face d'un des capteurs pir un message nous sera envoyer

```
Mouvement détecté
Mouvement détecté2
Mouvement détecté3
```

Figure 66: capture d'image du port serie

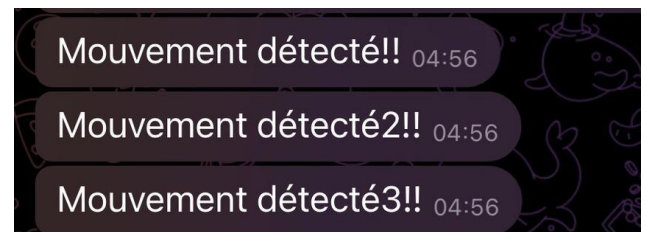


Figure 67: capture d'image du bot telegram

- test de commandes via Telegram :

nous allons saisir le message /go sur le bot telegram des commandes vont apparaître, puis nous allons appuyez sur les commandes /ledR et /ledO pour allumer les leds puis rappuyer pour les éteindre

Chapitre 3 : Réalisation pratique



Figure 68: capture d'image du bot telegram

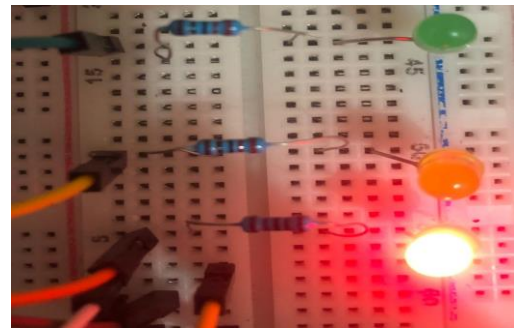


Figure 69: photo de la led rouge allumer

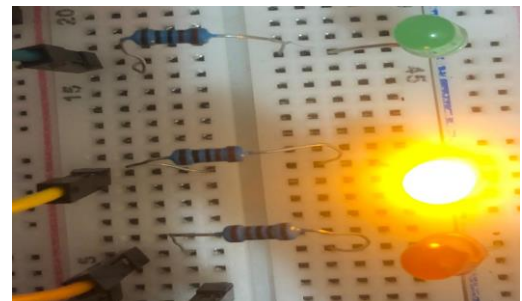


Figure 70: photo de la led orange allumer

nous allons appuyez maintenant sur la commande /ledA pour allumer les leds de manière aléatoire puis rappuyer pour les éteindre

Chapitre 3 : Réalisation pratique

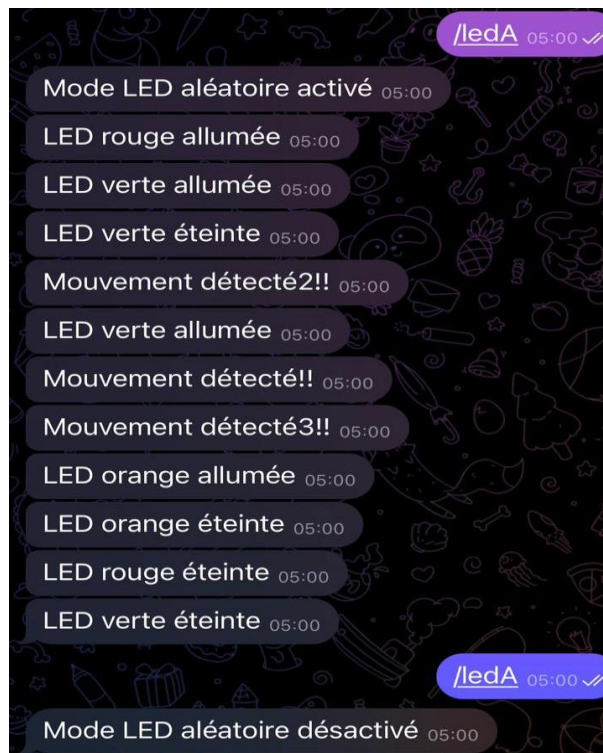


Figure 71: capture d'image du bot telegram

3.6 Discussion:

Dans ce chapitre nous avons décrit en détail notre système, en abordant ses fonctionnalités ainsi que ses connexions matérielles et configuration logicielle. Des tests ont été effectués avec succès garantissant son bon fonctionnement.

L'objectif principal de ce projet était de concevoir et réaliser un système de télésurveillance à l'aide des microcontrôleurs ESP32-CAM et ESP32, ce projet a démontré la faisabilité et l'efficacité de l'utilisation des ESP32 comme centre de contrôle pour un système de sécurité domestique intelligent. En combinant plusieurs capteurs et actionneurs et en exploitant les capacités de communication de l'ESP32, nous avons pu créer un système polyvalent et réactif capable de surveiller l'environnement et d'alerter les utilisateurs en temps réel via Telegram. Les défis rencontrés ont permis d'apprendre et d'appliquer des techniques avancées de gestion des ressources et d'intégration matérielle. Ce projet représente une base solide pour des applications de sécurité domestique intelligentes et peut servir de modèle pour des projets similaires à l'avenir.

Conclusion

Conclusion

La révolution technologique a profondément transformé notre quotidien, facilitant nos interactions et nous permettant de rester connectés en permanence. Les smartphones, les ordinateurs et l'internet jouent désormais un rôle essentiel dans nos vies sociale et professionnelle. De même, les systèmes de surveillance, en particulier les technologies de vidéosurveillance, ont évolué de manière significative, passant d'applications militaires à une utilisation domestique plus large.

Dans le cadre de notre projet de fin d'études, nous avons eu l'opportunité d'explorer ce domaine en nous intéressant à l'étude et à la réalisation d'un système de télésurveillance avec détection de mouvement, basé sur une carte ESP32-CAM et ESP32. Malgré la complexité et les défis inhérents à ce projet, nous avons réussi à atteindre nos objectifs. Le système est pleinement fonctionnel et a donné entière satisfaction lors des essais et des tests de mise en service.

Ce projet nous a permis de faire le lien entre l'étude théorique d'un montage électronique et sa réalisation pratique, validant ainsi nos connaissances théoriques par la pratique. À travers ce mémoire, nous avons réalisé notre objectif de créer un système de sécurité et d'alarme. En outre, nous avons acquis les compétences suivantes :

- ⑩ Compréhension de l'architecture interne du module ESP32-CAM et apprentissage de sa programmation. De même pour l'ESP32.
- ⑩ Utilisation du module RFID.
- ⑩ Maîtrise du programme Arduino IDE.

Le système que nous avons développé est composé d'un ensemble de capteurs permettant l'acquisition des données (PIR pour la détection de présence et un capteur de gaz MQ135), des actionneurs (servomoteur, buzzer et LED) pour un système d'accès sécurisé. Ce système inclut également un module RFID pour gérer servomoteur comme une porte d'entrée. De plus, nous avons intégré une carte ESP32-CAM qui reçoit les données des capteurs PIR détectant les mouvements dans des zones spécifiques. Chaque mouvement capturé est sauvegardé grâce à la caméra intégrée et la carte microSD de l'ESP32. Grâce à cette carte, nous pouvons recevoir des données à distance via une application Telegram, nous informant des mouvements dans notre domicile en notre absence.

Ce travail nous a permis de gagner en confiance et nous espérons que d'autres projets futurs utiliseront des techniques avancées de systèmes de sécurité pour améliorer la gestion des accès aux espaces privés.

Bibliographie :

- Mémoires :

[1] Rachedi K, 2022, Mémoire en Master, thème : Conception d'un système de sécurité avec système de surveillance caméra ESP32-CAM intégrée, UMMTO

[2] SIDI YKHLEF A et LASLA A, 2015, Mémoire en Master, thème : Mise au point d'une « Application de télésurveillance », UABB de Tlemcen

[3] ZIDOUR S et KEBAILI I,2022, Mémoire en Master, thème : Etude et réalisation d'un système de télésurveillance avec détection de présence Basé sur une carte ESP32-CAM, Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

[4] BOUHALI O et KAOUDJT L, 2018, Mémoire en Master, thème :Conception et réalisation d'un système de télésurveillance, UMMTO

- Sites Internet :

[5] <https://randomnerdtutorials.com/>

[6] <https://forum.arduino.cc/>

[7] <https://www.instructables.com/Home-Security-System-Using-ESP32-CAM-and-Telegram-/>

[8] <https://fr.wikipedia.org/wiki/T%C3%A9l%C3%A9surveillance>

[9] <https://www.forum.fritzing.org>

[10] <https://github.com/>

[11] <https://www.hackster.io/>

Annexe :

Programme de l'ESP32-CAM :

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <FS.h>
#include <SD_MMC.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
const char* ssid = "xxxx";
const char* password = "xxxx";
String BOTtoken = "xxxx:xxxx";
String CHAT_ID = "xxxx";
bool sendPhoto = false;
#define FLASH_LED_PIN 4
bool flashState = LOW;
WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);
// Vérifie les nouveaux messages toutes les 1 seconde.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;
const int PIRsensor = 12;
int PIRstate = LOW; // On commence en supposant qu'aucun mouvement n'a été détecté
int val = 0;
// Temps donné au capteur pour se calibrer (environ 10-60 secondes selon la fiche technique)
const int calibrationTime = 30; // 30 secondes
int numero_fichier = 0;
// Modèle de caméra AI_THINKER
#define PWDN_GPIO_NUM 32
```

```

#define RESET_GPIO_NUM  -1
#define XCLK_GPIO_NUM   0
#define SIOD_GPIO_NUM   26
#define SIOC_GPIO_NUM   27
#define Y9_GPIO_NUM     35
#define Y8_GPIO_NUM     34
#define Y7_GPIO_NUM     39
#define Y6_GPIO_NUM     36
#define Y5_GPIO_NUM     21
#define Y4_GPIO_NUM     19
#define Y3_GPIO_NUM     18
#define Y2_GPIO_NUM     5
#define VSYNC_GPIO_NUM  25
#define HREF_GPIO_NUM   23
#define PCLK_GPIO_NUM   22

void configInitCamera() {
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;

```

```

config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
// Initialisation avec des spécifications élevées pour pré-allouer des tampons plus grands
if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10; // 0-63, un nombre plus bas signifie une meilleure qualité
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12; // 0-63, un nombre plus bas signifie une meilleure qualité
    config.fb_count = 1; }
// Initialisation de la caméra
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Échec de l'initialisation de la caméra avec l'erreur 0x%x", err);
    delay(1000);
    ESP.restart();}
// Réduction de la taille de l'image pour un débit d'image initial plus élevé
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_CIF); //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA}
void sauvegarderPhoto(camera_fb_t *fb) {
    String nomFichier = "/photo" + String(numero_fichier) + ".jpg";
    numero_fichier++;
    File fichierPhoto = SD_MMC.open(nomFichier, FILE_WRITE);
    if (!fichierPhoto) {
        Serial.println("Erreur lors de l'ouverture du fichier pour sauvegarder la photo");
        return; }
    fichierPhoto.write(fb->buf, fb->len);

```

```

fichierPhoto.close();
Serial.println("Photo sauvegardée: " + nomFichier);}
void handleNewMessages(int numNewMessages) {
  Serial.print("Gérer les nouveaux messages : ");
  Serial.println(numNewMessages);
  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID) {
      bot.sendMessage(chat_id, "Utilisateur non autorisé", "");
      continue; }
    // Afficher le message reçu
    String text = bot.messages[i].text;
    Serial.println(text);
    String from_name = bot.messages[i].from_name;
    if (text == "/start") {
      String welcome = "Bienvenue, " + from_name + "\n";
      welcome += "Utilisez la commande suivante pour interagir avec l'ESP32-CAM \n";
      welcome += "/photo : prend une nouvelle photo\n";
      bot.sendMessage(CHAT_ID, welcome, ""); }
    if (text == "/photo") {
      sendPhoto = true;
      Serial.println("Nouvelle demande de photo"); } } }
String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String getBody = "";
  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if (!fb) {
    Serial.println("Échec de la capture par la caméra");
    delay(1000);
    ESP.restart();

```

```

return "Échec de la capture par la caméra"; }
Serial.println("Connexion à " + String(myDomain));
if (clientTCP.connect(myDomain, 443)) {
    Serial.println("Connexion réussie");
    String head = "--c010blind3ngineer\r\nContent-Disposition: form-data; name=\"chat_id\";
\r\n\r\n" + CHAT_ID + "\r\n--c010blind3ngineer\r\nContent-Disposition: form-data;
name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
    String tail = "\r\n--c010blind3ngineer--\r\n";
    uint16_t imageLen = fb->len;
    uint16_t extraLen = head.length() + tail.length();
    uint16_t totalLen = imageLen + extraLen;
    clientTCP.println("POST /bot" + BOTtoken + "/sendPhoto HTTP/1.1");
    clientTCP.println("Host: " + String(myDomain));
    clientTCP.println("Content-Length: " + String(totalLen));
    clientTCP.println("Content-Type: multipart/form-data; boundary=c010blind3ngineer");
    clientTCP.println();
    clientTCP.print(head);
    uint8_t *fbBuf = fb->buf;
    size_t fbLen = fb->len;
    for (size_t n = 0; n < fbLen; n = n + 1024) {
        if (n + 1024 < fbLen) {
            clientTCP.write(fbBuf, 1024);
            fbBuf += 1024; }
        else if (fbLen % 1024 > 0) {
            size_t remainder = fbLen % 1024;
            clientTCP.write(fbBuf, remainder); } }
    clientTCP.print(tail);
    esp_camera_fb_return(fb);
    int waitTime = 10000; // timeout de 10 secondes
    long startTimer = millis();
    boolean state = false;
    while ((startTimer + waitTime) > millis()) {

```

```

Serial.print(".");
delay(100);
while (clientTCP.available()) {
  char c = clientTCP.read();
  if (state == true) getBody += String(c);
  if (c == '\n') {
    if (getAll.length() == 0) state = true;
    getAll = ""; }
  else if (c != '\r')
    getAll += String(c);
  startTimer = millis(); }
  if (getBody.length() > 0) break; }
clientTCP.stop();
Serial.println(getBody);
} else {
  getBody = "Échec de la connexion à api.telegram.org.";
  Serial.println("Échec de la connexion à api.telegram.org."); }
return getBody; }

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  // Initialisation du Moniteur Série
  Serial.begin(115200);
  // Définir la LED Flash comme sortie
  pinMode(FLASH_LED_PIN, OUTPUT);
  digitalWrite(FLASH_LED_PIN, flashState);
  // Définir le capteur PIR comme entrée et la LED comme sortie
  pinMode(PIRsensor, INPUT);
  // Donner un certain temps au capteur PIR pour se stabiliser
  Serial.println("Attente que le capteur se stabilise au premier démarrage");
  delay(calibrationTime * 1000); // Temps converti en millisecondes
  // Se connecter au Wi-Fi
  WiFi.mode(WIFI_STA);

```

```

Serial.println();
Serial.print("Connexion à ");
Serial.println(ssid);
WiFi.begin(ssid, password);
clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Ajouter le certificat racine pour
api.telegram.org
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500); }
Serial.println();
Serial.print("Adresse IP de l'ESP32-CAM : ");
Serial.println(WiFi.localIP());
  bot.sendMessage(CHAT_ID, "Bot started up", "");
// Configurer et initialiser la caméra
configInitCamera();
if (!SD_MMC.begin()) {
  Serial.println("Erreur lors de l'initialisation de la carte SD");
  return; } }
void loop() {
  val = digitalRead(PIRsensor);
  if (val == HIGH) {
    if (PIRstate == LOW) {
      camera_fb_t *fb = esp_camera_fb_get();
      if (fb) {    sauvegarderPhoto(fb);
        esp_camera_fb_return(fb);    }
      // Nous venons de nous allumer car un mouvement a été détecté
      Serial.println("Mouvement détecté !");
      digitalWrite(FLASH_LED_PIN, HIGH);
      delay(500);
      Serial.println("Envoi de la photo à Telegram");
      sendPhotoTelegram();
      PIRstate = HIGH;

```

```

    digitalWrite(FLASH_LED_PIN, LOW); } }
else if (sendPhoto) {
Serial.println("Préparation de la photo");
digitalWrite(FLASH_LED_PIN, HIGH);
Serial.println("État de flash réglé sur HAUT");
delay(500);
// Capture de la photo
camera_fb_t *fb = esp_camera_fb_get();
if (fb) {
    // Sauvegarde de la photo sur la carte SD
sauvegarderPhoto(fb);
    // Envoi de la photo via Telegram
sendPhotoTelegram();
    // Libération de la mémoire allouée pour la photo
esp_camera_fb_return(fb); }
sendPhoto = false;
digitalWrite(FLASH_LED_PIN, LOW);
Serial.println("État de flash réglé sur BAS"); }
else if (millis() > lastTimeBotRan + botRequestDelay) {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
        Serial.println("Réponse reçue");
        handleNewMessages(numNewMessages);
        numNewMessages = bot.getUpdates(bot.last_message_received + 1); }
    lastTimeBotRan = millis(); } else {
    if (PIRstate == HIGH) {
        Serial.println("Mouvement terminé !");
PIRstate = LOW; } } }

```

Programme de l'ESP32 :

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <SPI.h>
#include <MFRC522.h>
#include <ESP32Servo.h>

// Informations sur le réseau Wi-Fi
const char* ssid = "xxxx";
const char* password = "xxxx";
String BOTtoken = "xxxx:xxxx";
String CHAT_ID = "xxxx";

// Définition des broches
const int SS_PIN = 5; // SDA du module RFID
const int RST_PIN = 27; // RST du module RFID
const int servoPin = 14; // Broche pour le servo
const int greenLED2 = 12; // Broche pour la LED verte
const int redLED2 = 13; // Broche pour la LED rouge
const int motionSensor = 32; // Broche pour le premier capteur de mouvement
const int motionSensor2 = 33; // Broche pour le deuxième capteur de mouvement
const int motionSensor3 = 25; // Broche pour le troisième capteur de mouvement
const int MQ135Pin = 34; // Broche pour le capteur de gaz
const int buzzerPin2 = 26; // Broche pour le buzzer
const int gasThreshold = 300; // Seuil de détection du gaz

#define LED_RED_PIN 4
#define LED_GREEN_PIN 2
#define LED_ORANGE_PIN 16
#define BUZZER_PIN 15 // Ajout du pin du buzzer

WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);
MFRC522 rfid(SS_PIN, RST_PIN);
```

```

Servo myServo;
int botRequestDelay = 1000;
unsigned long lastTimeBotRan; // Déclaration de lastTimeBotRan ici
bool randomLedMode = false; // Flag pour le mode aléatoire
// Variables pour la détection de mouvement
bool motionDetected = false;
bool motionDetected2 = false;
bool motionDetected3 = false;
bool gasDetected = false;
unsigned long previousMillisGas = 0;
const long intervalGas = 5000; // Intervalle pour la réinitialisation de la détection de gaz
// Fonctions d'interruption pour les capteurs de mouvement
void IRAM_ATTR detectsMovement() {
    motionDetected = true; }
void IRAM_ATTR detectsMovement2() {
    motionDetected2 = true; }
void IRAM_ATTR detectsMovement3() {
    motionDetected3 = true; }
void setup() {
    Serial.begin(115200);
    pinMode(LED_RED_PIN, OUTPUT);
    pinMode(LED_GREEN_PIN, OUTPUT);
    pinMode(LED_ORANGE_PIN, OUTPUT);
    pinMode(BUZZER_PIN, OUTPUT);
    // Éteindre toutes les LEDs et le buzzer au démarrage
    digitalWrite(LED_RED_PIN, LOW);
    digitalWrite(LED_GREEN_PIN, LOW);
    digitalWrite(LED_ORANGE_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);
    myServo.attach(servoPin);
    pinMode(greenLED2, OUTPUT);
    pinMode(redLED2, OUTPUT);

```

```

// Initialisation du bus SPI et du module RFID
SPI.begin();
rfid.PCD_Init();
// Afficher la version du firmware du module RFID
rfid.PCD_DumpVersionToSerial();
// Configurer les capteurs de mouvement
pinMode(motionSensor, INPUT_PULLUP);
pinMode(motionSensor2, INPUT_PULLUP);
pinMode(motionSensor3, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(motionSensor), detectsMovement, RISING);
attachInterrupt(digitalPinToInterrupt(motionSensor2), detectsMovement2, RISING);
attachInterrupt(digitalPinToInterrupt(motionSensor3), detectsMovement3, RISING);
// Configurer la broche du buzzer
pinMode(buzzerPin2, OUTPUT);
digitalWrite(buzzerPin2, LOW); // S'assurer que le buzzer est éteint au démarrage
// Connexion au Wi-Fi
Serial.print("Connexion au Wi-Fi");
WiFi.begin(ssid_wifi, password_wifi);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print("."); }
Serial.println();
Serial.println("Wi-Fi connecté");
Serial.print("Adresse IP : ");
Serial.println(WiFi.localIP());
// Configurer le client Telegram
clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Ajouter le certificat racine pour
api.telegram.org
  bot.sendMessage(CHAT_ID, "Le bot a démarré", ""); }
void handleNewMessages(int numNewMessages) {
  Serial.print("Gérer les nouveaux messages : ");
  Serial.println(numNewMessages);

```

```

for (int i = 0; i < numNewMessages; i++) {
  String chat_id = String(bot.messages[i].chat_id);
  if (chat_id != CHAT_ID) {
    bot.sendMessage(chat_id, "Utilisateur non autorisé", "");
    continue;  }
  // Afficher le message reçu
  String text = bot.messages[i].text;
  Serial.println(text);
  String from_name = bot.messages[i].from_name;
  if (text == "/go") {
    String welcome = "Bienvenue, " + from_name + "\n";
    welcome += "Utilisez les commandes suivantes pour interagir avec l'ESP32 \n";
    welcome += "/ledR : allumer/éteindre la led rouge \n";
    welcome += "/ledV : allumer/éteindre la led verte \n";
    welcome += "/ledO : allumer/éteindre la led orange \n";
    welcome += "/ledA : allumer/éteindre les leds aléatoirement \n";
    welcome += "/buzzer : allumer/éteindre le buzzer \n"; // Ajout de l'instruction pour le buzzer
    bot.sendMessage(CHAT_ID, welcome, "");  }
  if (text == "/ledR") {
    if (digitalRead(LED_RED_PIN) == LOW) {
      digitalWrite(LED_RED_PIN, HIGH);
      bot.sendMessage(CHAT_ID, "LED rouge allumée", "");
    } else {
      digitalWrite(LED_RED_PIN, LOW);
      bot.sendMessage(CHAT_ID, "LED rouge éteinte", "");  }  }
  if (text == "/ledV") {
    if (digitalRead(LED_GREEN_PIN) == LOW) {
      digitalWrite(LED_GREEN_PIN, HIGH);
      bot.sendMessage(CHAT_ID, "LED verte allumée", "");
    } else {
      digitalWrite(LED_GREEN_PIN, LOW);
      bot.sendMessage(CHAT_ID, "LED verte éteinte", "");  }  }
}

```

```

if (text == "/ledO") {
  if (digitalRead(LED_ORANGE_PIN) == LOW) {
    digitalWrite(LED_ORANGE_PIN, HIGH);
    bot.sendMessage(CHAT_ID, "LED orange allumée", "");
  } else {
    digitalWrite(LED_ORANGE_PIN, LOW);
    bot.sendMessage(CHAT_ID, "LED orange éteinte", "");  }  }
if (text == "/ledA") {
  randomLedMode = !randomLedMode;
  if (randomLedMode) {
    bot.sendMessage(CHAT_ID, "Mode LED aléatoire activé", "");
  } else {
    // Éteindre toutes les LEDs lorsque le mode est désactivé
    digitalWrite(LED_RED_PIN, LOW);
    digitalWrite(LED_GREEN_PIN, LOW);
    digitalWrite(LED_ORANGE_PIN, LOW);
    bot.sendMessage(CHAT_ID, "Mode LED aléatoire désactivé", "");  }  }
if (text == "/buzzer") { // Gestion de la commande pour le buzzer
  if (digitalRead(BUZZER_PIN) == LOW) {
    digitalWrite(BUZZER_PIN, HIGH);
    bot.sendMessage(CHAT_ID, "Buzzer allumé", "");
  } else {
    digitalWrite(BUZZER_PIN, LOW);
    bot.sendMessage(CHAT_ID, "Buzzer éteint", "");  }  }  }  }
void loop() {
  unsigned long currentMillis = millis();
  // Vérifier les nouveaux messages Telegram
  if (millis() > lastTimeBotRan + botRequestDelay) {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
      Serial.println("Réponse reçue");
      handleNewMessages(numNewMessages);
    }
  }
}

```

```

    numNewMessages = bot.getUpdates(bot.last_message_received + 1); }
lastTimeBotRan = millis(); }
// Section RFID
if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
    String UID = "";
    for (byte i = 0; i < rfid.uid.size; i++) {
        UID.concat(String(rfid.uid.uidByte[i] < 0x10 ? " 0" : " "));
        UID.concat(String(rfid.uid.uidByte[i], HEX)); }
    UID.toUpperCase();
    // Remplacer "73 C3 CC FD" par l'UID de votre carte autorisée
    if (UID.substring(1) == "73 C3 CC FD") {
        Serial.println("Accès autorisé. Ouverture...");
        digitalWrite(greenLED2, HIGH);
        myServo.write(90); // Position déverrouillée, ajuster si nécessaire
        bot.sendMessage(CHAT_ID, "Accès autorisé. Ouverture.", "");
        delay(2000);
        digitalWrite(greenLED2, LOW);
    } else {
        Serial.println("Accès refusé. Mauvais badge.");
        digitalWrite(redLED2, HIGH);
        myServo.write(0);
        bot.sendMessage(CHAT_ID, "Accès refusé. Mauvais badge détecté.", "");
        delay(1000);
        digitalWrite(redLED2, LOW); }
    rfid.PICC_DumpToSerial(&(rfid.uid)); }
// Section capteur de gaz
int sensorValue = analogRead(MQ135Pin);
Serial.print("Valeur MQ135 : ");
Serial.println(sensorValue);
if (sensorValue > gasThreshold && !gasDetected) {
    bot.sendMessage(CHAT_ID, "Gaz détecté!", "");
    Serial.println("Gaz détecté!");
}

```

```

digitalWrite(buzzerPin2, HIGH); // Activer le buzzer
previousMillisGas = currentMillis;
gasDetected = true; // Empêcher l'envoi de multiples messages
}
// Réinitialiser la détection de gaz après 5 secondes
if (gasDetected && currentMillis - previousMillisGas >= intervalGas) {
    digitalWrite(buzzerPin2, LOW); // Éteindre le buzzer
    gasDetected = false; }
// Section détection de mouvement
if (motionDetected) {
    bot.sendMessage(CHAT_ID, "Mouvement détecté!!", "");
    Serial.println("Mouvement détecté");
    motionDetected = false; }
if (motionDetected2) {
    bot.sendMessage(CHAT_ID, "Mouvement détecté2!!", "");
    Serial.println("Mouvement détecté2");
    motionDetected2 = false; }
if (motionDetected3) {
    bot.sendMessage(CHAT_ID, "Mouvement détecté3!!", "");
    Serial.println("Mouvement détecté3");
    motionDetected3 = false; }
// Gestion du mode LED aléatoire
if (randomLedMode) {
    int ledToToggle = random(0, 3);
    int ledPin;
    String ledColor;
    switch (ledToToggle) {
        case 0:
            ledPin = LED_RED_PIN;
            ledColor = "rouge";
            break;
        case 1:

```

```

    ledPin = LED_GREEN_PIN;
    ledColor = "verte";
    break;
case 2:
    ledPin = LED_ORANGE_PIN;
    ledColor = "orange";
    break; }
int ledState = digitalRead(ledPin);
digitalWrite(ledPin, !ledState);
bot.sendMessage(CHAT_ID, "LED " + ledColor + " " + String(ledState ? "éteinte" : "allumée"),
"");
    delay(1000); // Attendre une seconde avant de changer l'état des LEDs
}
delay(100); // Attendre 100 ms entre chaque boucle pour lire les capteurs
}

```

Résumé :

La sécurité des biens et des personnes est devenue une préoccupation majeure dans notre société moderne. Avec l'augmentation des taux de criminalité et la nécessité accrue de protéger les espaces privés et publics, les systèmes de télésurveillance sont devenus des outils indispensables. Ces systèmes permettent non seulement de dissuader les actes malveillants, mais également de détecter et de réagir rapidement aux incidents. C'est dans cette thématique que s'inscrit notre projet, visant à concevoir et réalisation un système de télésurveillance basé sur un microcontrôleurs qui se charge de mettre en œuvre diverses fonctionnalités telles que la détection de mouvement, la capture d'image, la surveillance de gaz, le contrôle d'accès, l'interaction à distance, ainsi que le stockage et l'accès aux données. Pour ce faire, nous avons réparti ce mémoire en trois chapitres : Généralités sur la télésurveillance, conception du système, et réalisation pratique.

Mots-clés :

Télésurveillance, détection de mouvement, alarme, sécurité, capteurs, actionneurs, microcontrôleurs, ESP32, module RFID, arduino IDE, application telegram, bot, capteur d'image