

الجـمـهـورـيـةـ الـجـزـائـرـيـةـ الـدـيمـقـرـاتـيـةـ الشـعـبـيـةـ
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارـةـ التـعـلـيـمـ العـالـيـ وـ الـبـحـثـ الـعـلـامـيـ
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جـامـعـةـ أـمـمـ مـدـبـوقـةـ بـوـمـرـدـاسـ
UNIVERSITE M'HAMED BOUGARA – BOUMERDES-
Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme de LICENCE**

En : (Electronique)

Par: (Ahmed Benyahia Nour El Islam Adlane)

Sujet

SYSTEME DE SURVEILLANCE A DOMICILE

Encadrant : Karima Kessaissia

ANNÉE UNIVERSITIES : 2022/2023

Sommaire

I. Introduction	9
II. CHAPITRE 1 : Généralité.....	11
1. Définition du système de surveillance à domicile.....	12
2. Types du système de surveillance à domicile.....	13
2.1. Caméras de sécurité.....	13
2.2. Capteurs de mouvement.....	13
2.3. Systèmes d'alarme.....	13
2.4. Contrôle d'accès.....	13
2.5. Domotique.....	13
2.6. Notifications et contrôle à distance.....	13
3. Application du système de surveillance à domicile.....	14
3.1. Surveillance de sécurité.....	14
3.2. Surveillance des enfants.....	14
3.3. Détection d'incendie.....	14
3.4. Contrôle de l'accès.....	14
3.5. Gestion de l'énergie.....	14
4. Durée de vie d'un système de surveillance à domicile.....	14
4.1. La qualité des composants.....	14
4.2. L'environnement.....	14
4.3. La fréquence d'utilisation.....	14
4.4. La maintenance.....	15
4.5. Les mises à jour technologiques.....	15
5. Conclusion.....	15
III. CHAPITRE 2 : outils de réalisation.....	16
1. Outils de réalisation.....	17
1.1. La carte arduino.....	17
1.2. Les types d'arduino	17
1.2.1. Arduino Uno.....	17
1.2.2. Arduino Mega.....	17
1.2.3. Arduino Nano.....	18
1.2.4. Arduino Due.....	18
1.2.5. Arduino Leonardo.....	18
1.2.6. Arduino Pro Mini.....	18
1.2.7. Arduino Zero.....	18
1.2.8. Arduino MKR1000.....	18
1.2.9. Arduino Yun.....	18
1.2.10.Arduino Robot.....	19

1.2.11. Arduino Nano Every.....	19
1.2.12. Arduino Portenta H7.....	19
1.2.13. Arduino MKR Vidor 4000.....	19
1.2.14. Arduino Nano RP2040 Connect.....	19
1.2.15. Arduino Tian.....	19
1.3. Arduino uno.....	20
1.3.1. Description.....	20
a. Microcontrôleur.....	20
b. Connecteur d'alimentation.....	21
c. Ports d'entrée/sortie numériques.....	21
d. Ports d'entrée/sortie analogiques.....	21
e. Communication série.....	21
f. IDE Arduino.....	21
g. Compatibilité.....	21
1.4. Les composants de la carte arduino uno.....	21
1.4.1. Microcontrôleur ATmega328Pt.....	21
1.4.2. Oscillateur.....	21
1.4.3. Port USB.....	21
1.4.4. Régulateur de tension.....	21
1.4.5. Broches d'E/S.....	22
1.4.6. LED d'alimentation.....	22
1.4.7. Bouton de reset.....	22
1.4.8. Connecteur d'alimentation.....	22
1.4.9. LED intégrée.....	22
1.4.10. Condensateurs.....	22
1.4.11. Résistances.....	22
1.4.12. Diodes.....	22
1.4.13. Connecteurs de broches.....	22
1.4.14. Fuseau horaire.....	22
1.4.15. Mémoire flash.....	22
1.4.16. Mémoire EEPROM.....	22
1.5. L'IDE Arduino.....	22
1.5.1. Interface utilisateur.....	23
1.5.2. Éditeur de code.....	23
1.5.3. Bibliothèques.....	23
1.5.4. Compilation et téléchargement.....	23
1.5.5. Débogage.....	23
1.6. Capteur de flamme.....	24
a. Fonctionnement.....	24

b.	Types de capteurs.....	24
c.	Applications.....	24
d.	Caractéristiques.....	24
e.	Limitations.....	24
1.7.	Capteur de lumière TEMT6000.....	25
a.	Fonctionnement.....	25
b.	Sensibilité.....	25
c.	Avantages.....	25
d.	Limitations.....	25
e.	Applications.....	25
1.8.	Capteur de son KY-038.....	26
a.	Fonctionnement.....	26
b.	Sensibilité.....	26
c.	Avantages.....	26
d.	Limitations.....	26
e.	Applications.....	26
1.9.	Capteur de gaz MQ-2.....	27
a.	Sensibilité.....	27
b.	Fonctionnement.....	27
c.	Avantages.....	27
d.	Limitations.....	27
e.	Applications.....	27
1.10.	GSM 800L.....	28
a.	Fréquences de bande.....	28
b.	Fonctionnalités.....	28
c.	Antenne.....	28
d.	Consommation d'énergie.....	28
e.	Commandes AT.....	28
1.11.	Camera.....	29
a.	Types de caméras.....	29
b.	Interface.....	29
c.	Alimentation.....	29
d.	Logiciel.....	29
e.	Applications.....	29
f.	Limitations.....	29
1.12.	Le module Bluetooth HC-05.....	30

a. Connexion	30
b. Bibliothèque	30
c. Configuration	30
d. Communication	30
e. Connexions multiples	30
f. Projets sans fil	31
 1.13. Buzzer.....	31
a. Types de buzzers.....	31
b. Connexion.....	31
c. Contrôle de la fréquence et de la durée.....	31
d. Utilisations avancées.....	31
 1.14. Capteur de mouvement SR602.....	32
a. Fonctionnement.....	32
b. Portée.....	32
c. Réglages.....	32
d. Alimentation.....	32
e. Facilité d'utilisation.....	32
f. Type de capteur.....	32
g. Utilisations.....	32
 1.15. App Inventor.....	33

IV. CHAPITRE 3 : Résultat de réalisation.....	35
1. Application Asentry.....	36
1.1. Définition.....	36
1.2. Block de Programmation du l'application.....	37
2. Résultat de réalisation.....	38
2.1. La simulation.....	38
2.2. Résultat final	40
V. CONCLUSION.....	41
VI. ANNEXE.....	43
a) Capteur de lumière TEMT6000.....	44
b) Capteur de flamme.....	45
c) Capteur de gaz MQ-2.....	46
d) GSM 800L.....	47
e) Capteur de mouvement SR602.....	48
f) Capteur de son KY-038.....	49

g) module Bluetooth HC-05.....	50
h) le projet finale.....	51
VII. BIBLIOGRAPHIE.....	55

Liste des figures

Figure 1 : système de surveillance à domicile.....	12
Figure 2 : Les types d'Arduino.....	19
Figure 3 : la carte Arduino Uno.....	20
Figure 4 : Les composants de la carte Arduino uno.....	22
Figure 5 : Arduino IDE.....	23
Figure 6 : Capteur de flamme.....	24
Figure 7 : capteur de lumière TEMT6000.....	25
Figure 8 : Capteur de son KY-038.....	26
Figure 9 : Capteur de Gaz MQ-2.....	27
Figure 10 : Le GSM 800L.....	28
Figure 11 : Camera Arduino.....	29
Figure 12 : Le module HC-05.....	31
Figure 13 : Le buzzer.....	32
Figure 14 : Capteur de mouvement SR602.....	33
Figure 15 : App Inventor.....	34
Figure 16 : captures de l'application Asentry.....	36
Figure 17 : Logo de l'application.....	36
Figure 18 : Block de programmation de l'application.....	37
Figure 19 : Circuit de Capteur de flamme.....	38
Figure 20 : Circuit de capteur TEMT6000.....	38
Figure 21 : Circuit de Capteur MQ-2.....	38
Figure 22 : Circuit de GSM 800L.....	38
Figure 23 : Circuit de Capteur SR602.....	38
Figure 24 : Circuit de Capteur KY-038.....	38
Figure 25 : Circuit de module Bluetooth HC-05.....	39
Figure 26 : Circuit de la caméra.....	39
Figure 27 : Circuit de projet.....	39
Figure 28 : la réalisation de projet.....	40

REMERCIEMENTS

C'est avec un immense plaisir que je présente ce mémoire, fruit d'un travail passionnant et enrichissant. Je tiens à remercier du fond du cœur ma directrice de mémoire, Madame Karima Kessaissia, pour son accompagnement, sa confiance et ses remarques pertinentes. Elle a su me guider et me stimuler tout au long de cette recherche.

Je remercie également l'ensemble du corps professoral et des intervenants professionnels qui ont contribué à ma formation et à mon épanouissement personnel. Leurs enseignements et leurs expériences ont été une source d'inspiration et de réflexion.

Je n'oublie pas mes collègues de promotion, qui ont partagé avec moi cette aventure humaine et intellectuelle. Leur soutien et leur bonne humeur ont été précieux pour surmonter les difficultés et les moments de doute.

Enfin, je dédie ce mémoire à ma famille, qui m'a toujours soutenu et encouragé dans mes études. Je leur témoigne toute ma gratitude et mon amour.

MERCI !

I. INTRODUCTION :

Le système de surveillance à domicile est devenu une solution de plus en plus populaire pour sécuriser les résidences. Avec l'augmentation des cambriolages et des incendies domestiques, les propriétaires cherchent des moyens plus efficaces de protéger leur famille et leur domicile. Les systèmes de surveillance à domicile sont des technologies avancées conçues pour offrir une surveillance i

Les avantages des systèmes de surveillance à domicile sont nombreux, notamment la dissuasion des cambrioleurs et des voleurs, la surveillance en temps réel, la détection précoce des incendies, la notification en cas de mouvements ou de bruits suspects, et la tranquillité d'esprit. Cependant, il est important de noter que la mise en place d'un système de surveillance à domicile doit respecter les lois et réglementations en vigueur dans votre pays, notamment en matière de protection de la vie privée.

Les systèmes de surveillance à domicile offrent une protection supplémentaire pour votre famille et votre domicile, en vous donnant la tranquillité d'esprit que vous méritez. Ils peuvent être adaptés à vos besoins spécifiques, être facilement installés et contrôlés, et fournir une surveillance en temps réel de votre domicile, que vous soyez à l'intérieur ou à l'extérieur.

II. CHAPITRE 1 :

Généralité

1. Définition du système de surveillance à domicile :

Le système de surveillance à domicile est un ensemble de dispositifs électroniques utilisés pour surveiller et protéger une maison, un appartement ou tout autre lieu de résidence. Ces dispositifs sont conçus pour offrir une surveillance constante et détecter les menaces potentielles en temps réel, tels que les cambriolages, les incendies, les fuites de gaz, les intrusions et autres dangers.

Les systèmes de surveillance à domicile peuvent être composés de divers dispositifs tels que des caméras de surveillance, des capteurs de mouvement, des détecteurs de fumée et de gaz, des serrures intelligentes, des interphones vidéo, ainsi que des alarmes sonores et visuelles. Ces dispositifs peuvent être installés à l'intérieur et/ou à l'extérieur de la maison, selon les besoins et les préférences du propriétaire.

L'un des avantages des systèmes de surveillance à domicile est qu'ils permettent aux propriétaires de surveiller leur domicile en temps réel, peu importe où ils se trouvent, grâce à des applications mobiles, des alertes par e-mail ou SMS et des services de surveillance en direct. Les systèmes de surveillance à domicile peuvent également être configurés pour alerter les occupants en cas de situation d'urgence, tels que les mouvements ou les bruits suspects, ou en cas de détection de fumée, de gaz ou d'incendie.

Les systèmes de surveillance à domicile sont de plus en plus populaires en raison de leur efficacité à dissuader les cambrioleurs et à offrir une tranquillité d'esprit aux occupants de la maison. Cependant, il est important de noter que la mise en place d'un système de surveillance à domicile doit respecter les lois et réglementations en vigueur dans votre pays, notamment en matière de protection de la vie privée.

En résumé, le système de surveillance à domicile est un ensemble de dispositifs électroniques conçus pour surveiller et protéger une résidence contre les menaces potentielles, et permettent aux propriétaires de surveiller leur domicile en temps réel et d'être alerté en cas d'urgence. Les systèmes de surveillance à domicile offrent une protection supplémentaire pour votre famille et votre domicile, en vous donnant la tranquillité d'esprit que vous méritez.



Figure 1 : système de surveillance à domicile

2. Types du système de surveillance à domicile :

Le système de surveillance à domicile est une application populaire de l'électronique et de la technologie qui vise à assurer la sécurité et la surveillance d'une résidence. quelques exemples d'applications du système de surveillance à domicile :

- 2.1. **Caméras de sécurité** : Les caméras de sécurité sont installées à des endroits stratégiques de la maison pour surveiller les entrées, les jardins, les garages, etc. Elles peuvent être connectées à un réseau local ou à Internet, permettant aux propriétaires de visualiser les flux vidéo en temps réel ou d'accéder aux enregistrements à distance.
- 2.2. **Capteurs de mouvement** : Les capteurs de mouvement sont utilisés pour détecter toute activité inhabituelle à l'intérieur ou à l'extérieur de la maison. Ils peuvent déclencher des alarmes, envoyer des notifications aux propriétaires ou activer les caméras de sécurité pour enregistrer les événements suspects.
- 2.3. **Systèmes d'alarme** : Les systèmes d'alarme sont conçus pour alerter les propriétaires en cas d'intrusion, de détection de fumée, de fuite d'eau, etc. Ils peuvent inclure des détecteurs de portes et de fenêtres, des détecteurs de fumée, des détecteurs de monoxyde de carbone, des détecteurs de gaz, etc.
- 2.4. **Contrôle d'accès** : Les systèmes de contrôle d'accès permettent de restreindre l'entrée dans la maison en utilisant des codes d'accès, des cartes RFID ou des empreintes digitales. Ils offrent une couche supplémentaire de sécurité en empêchant l'accès non autorisé.
- 2.5. **Domotique** : La domotique intègre des fonctionnalités de surveillance à domicile avec d'autres systèmes automatisés. Par exemple, il est possible de contrôler à distance les lumières, le chauffage, les appareils électroménagers, etc., et de les lier aux systèmes de sécurité pour simuler une présence lorsque les propriétaires sont absents.
- 2.6. **Notifications et contrôle à distance** : Les propriétaires peuvent recevoir des notifications sur leur smartphone ou leur ordinateur en cas d'événements de sécurité importants. Ils peuvent également contrôler et surveiller leur système de surveillance à domicile à distance, en utilisant des applications mobiles ou des interfaces web.

3. Application du système de surveillance à domicile :

Le système de surveillance à domicile trouve de nombreuses applications dans la sécurisation et la protection des résidences. quelques exemples d'applications courantes :

- 3.1. **Surveillance de sécurité** : Le système de surveillance à domicile peut être utilisé pour détecter les intrusions et les activités suspectes dans et autour de la maison. Cela peut inclure des capteurs de mouvement, des caméras de sécurité et des alarmes qui sont activées en cas de détection d'une intrusion.
- 3.2. **Surveillance des enfants** : Les parents peuvent utiliser des caméras de surveillance pour garder un œil sur leurs enfants lorsqu'ils ne sont pas à la maison. Cela permet de s'assurer de leur sécurité et de les superviser à distance.
- 3.3. **Détection d'incendie** : Les systèmes de surveillance à domicile peuvent également inclure des détecteurs de fumée et de monoxyde de carbone pour alerter les occupants en cas d'incendie ou de présence de gaz dangereux. Ces systèmes peuvent également être connectés à des services d'urgence pour une intervention rapide.
- 3.4. **Contrôle de l'accès** : Les systèmes de surveillance à domicile peuvent être intégrés à des systèmes de contrôle d'accès pour permettre l'ouverture des portes à distance. Cela peut être utile pour permettre l'entrée de visiteurs autorisés ou pour surveiller l'accès des personnes à la propriété.
- 3.5. **Gestion de l'énergie** : Certains systèmes de surveillance à domicile offrent des fonctionnalités de gestion de l'énergie, permettant aux utilisateurs de contrôler et de surveiller la consommation d'énergie de leurs appareils électriques. Cela peut aider à réduire les coûts énergétiques et à promouvoir l'efficacité énergétique.

4. Durée de vie d'un système de surveillance à domicile :

La durée de vie d'un système de surveillance à domicile dépend de plusieurs facteurs, notamment :

- 4.1. **La qualité des composants** : la durée de vie du système dépendra de la qualité des composants utilisés pour la fabrication des caméras, des enregistreurs vidéo et des autres équipements de surveillance. Il est important de choisir des équipements de haute qualité pour s'assurer qu'ils résistent à l'usure et à la dégradation au fil du temps.
- 4.2. **L'environnement** : le système de surveillance doit être capable de fonctionner dans des environnements difficiles tels que les températures extrêmes, l'humidité, la poussière et les intempéries. Si le système est exposé à ces conditions, sa durée de vie peut être réduite.
- 4.3. **La fréquence d'utilisation** : la durée de vie du système peut également être influencée par la fréquence d'utilisation. Les systèmes de surveillance qui sont

utilisés plus fréquemment peuvent s'user plus rapidement que ceux qui sont utilisés occasionnellement.

4.4. La maintenance : la durée de vie du système dépend également de la maintenance régulière. Les systèmes de surveillance doivent être nettoyés régulièrement pour éviter l'accumulation de poussière et de saleté, ce qui peut réduire la qualité de l'image et affecter les performances du système.

4.5. Les mises à jour technologiques : les technologies de surveillance évoluent rapidement et les systèmes plus anciens peuvent devenir obsolètes avec le temps. Il est important de mettre à jour régulièrement les systèmes de surveillance pour s'assurer qu'ils restent efficaces et à jour avec les dernières technologies.

En conclusion, la durée de vie d'un système de surveillance à domicile dépend de la qualité des composants, de l'environnement dans lequel il est utilisé, de la fréquence d'utilisation, de la maintenance régulière et des mises à jour technologiques. Pour maximiser la durée de vie du système, il est important de choisir des équipements de haute qualité, de les installer correctement et de les entretenir régulièrement.

5. Conclusion :

En conclusion de ce premier chapitre, nous avons vu que le système de surveillance à domicile est un ensemble de dispositifs électroniques qui permettent de surveiller et de sécuriser un domicile. Nous avons défini les différents types de systèmes de surveillance à domicile, notamment les systèmes de vidéosurveillance, les systèmes d'alarme, les systèmes de détection de mouvement, et les systèmes de contrôle d'accès. Nous avons également examiné les applications possibles des systèmes de surveillance à domicile, telles que la protection contre le vol, la surveillance des enfants, la surveillance des animaux domestiques, et la surveillance de personnes âgées ou malades.

En outre, nous avons abordé la question de la durée de vie d'un système de surveillance à domicile, qui peut varier en fonction de plusieurs facteurs tels que la qualité des composants, la fréquence d'utilisation, l'environnement dans lequel il est utilisé et la maintenance régulière. Pour maximiser la durée de vie du système, il est important de choisir des équipements de haute qualité, de les installer correctement et de les entretenir régulièrement.

Aussi dans ce chapitre, nous allons examiner plus en détail les différents types de systèmes de surveillance à domicile.

III. CHAPITRE 2 :

Outils de Réalisation

1. OUTILS DE REALISATION :

1.1.La carte arduino :

La carte Arduino est une plateforme de développement électronique open-source qui permet de créer des projets interactifs en utilisant des microcontrôleurs. Elle est composée d'une carte de circuit imprimé avec un microcontrôleur, des broches d'entrée/sortie, des connecteurs et des composants électroniques [1].

L'un des microcontrôleurs les plus couramment utilisés dans les cartes Arduino est l'ATmega328P, qui offre une puissance de traitement suffisante pour de nombreux projets. Cependant, il existe également d'autres modèles comme l'ATmega2560, l'ESP8266 et l'ESP32, qui offrent des fonctionnalités supplémentaires telles que le Wi-Fi et le Bluetooth, ainsi qu'une plus grande capacité de stockage et de mémoire [1].

La carte Arduino est programmable à l'aide du logiciel Arduino IDE (Integrated Development Environment), qui est spécialement conçu pour simplifier le processus de développement et de téléchargement du code sur la carte Arduino. L'IDE Arduino est basé sur le langage de programmation Wiring, qui est une version simplifiée de C++ [1].

Avec Arduino, vous pouvez écrire du code pour contrôler les broches d'entrée/sortie de la carte, interagir avec différents capteurs tels que des capteurs de lumière, de température, d'humidité, etc., et piloter des actionneurs tels que des moteurs, des LEDs, des servomoteurs, etc. Vous pouvez également communiquer avec d'autres périphériques via des interfaces telles que l'I2C, le SPI ou l'UART [1].

Arduino offre une grande flexibilité et une facilité d'utilisation, ce qui en fait une plateforme populaire pour les projets électroniques, qu'ils soient destinés aux débutants ou aux experts en électronique.

1.2.Les types d'arduino :

1.2.1. **Arduino Uno** : est l'une des cartes Arduino les plus populaires et couramment utilisées. Elle est équipée d'un microcontrôleur ATmega328P et offre 14 broches d'entrée/sortie numériques, 6 broches d'entrée analogique et fonctionne à une fréquence d'horloge de 16 MHz. Elle est compatible avec la plupart des boucliers Arduino [2].

1.2.2. **Arduino Mega** : est une carte similaire à l'Arduino Uno, mais dotée d'un microcontrôleur plus puissant (ATmega2560) et d'un plus grand nombre de broches d'entrée/sortie. Elle offre 54 broches d'entrée/sortie numériques, 16 broches d'entrée analogique et fonctionne à une fréquence d'horloge de 16

MHz. Cette carte est fréquemment utilisée pour des projets complexes nécessitant de nombreuses connexions [3].

- 1.2.3. **Arduino Nano** : est une version plus petite de l'Arduino Uno, idéale pour les projets nécessitant une carte compacte. Elle est équipée d'un microcontrôleur ATmega328P et offre 14 broches d'entrée/sortie numériques ainsi que 8 broches d'entrée analogique. De plus, elle est compatible avec la plupart des boucliers (shields) Arduino [4].
- 1.2.4. **Arduino Due** : cette carte utilise un microcontrôleur ARM Cortex-M3 plus puissant et dispose d'un grand nombre de broches d'entrée/sortie. Elle est souvent utilisée pour les projets nécessitant une grande puissance de calcul. Elle est équipée de 54 broches d'entrée/sortie numériques, 12 broches d'entrée analogique, une fréquence d'horloge de 84 MHz et est compatible avec les boucliers Arduino standard [5].
- 1.2.5. **Arduino Leonardo** : cette carte utilise un microcontrôleur ATmega32u4 et dispose de 20 broches d'entrée/sortie numériques et 12 broches d'entrée analogique. Elle est souvent utilisée pour les projets nécessitant des fonctions d'émulation de clavier ou de souris. Elle dispose d'une fonction de connexion USB nativement, ce qui la rend idéale pour des projets nécessitant une communication série ou HID [6].
- 1.2.6. **Arduino Pro Mini** : c'est une version ultra-compacte de l'Arduino Uno, idéale pour les projets nécessitant une carte très petite. Elle est équipée d'un microcontrôleur ATmega328P, dispose de 14 broches d'entrée/sortie numériques et 8 broches d'entrée analogique. Elle ne dispose pas d'un connecteur USB natif et doit être programmée à l'aide d'un programmeur externe [7].
- 1.2.7. **Arduino Zero** : cette carte utilise un microcontrôleur ARM Cortex-M0+ plus puissant que l'ATmega328P, avec une fréquence d'horloge de 48 MHz. Elle est équipée de 14 broches d'entrée/sortie numériques et 6 broches d'entrée analogique. Elle dispose d'une connexion USB native et est compatible avec la plupart des boucliers Arduino [8].
- 1.2.8. **Arduino MKR1000** : cette carte est une carte compatible WiFi basée sur le microcontrôleur ARM Cortex-M0+. Elle dispose de 14 broches d'entrée/sortie numériques et 7 broches d'entrée analogique. Elle est équipée d'un module WiFi intégré qui permet la connexion à un réseau sans fil. Elle est souvent utilisée pour les projets IoT (Internet of Things) [9].
- 1.2.9. **Arduino Yun** : cette carte utilise un microcontrôleur ATmega32u4 et dispose d'un processeur MIPS Atheros AR9331 pour une connectivité WiFi intégrée. Elle est équipée de 20 broches d'entrée/sortie numériques et 12 broches d'entrée analogique. Elle est souvent utilisée pour les projets nécessitant une connexion Internet et pour la communication avec des appareils externes via des protocoles tels que REST ou MQTT [10].

1.2.10. **Arduino Robot** : cette carte est conçue pour les projets robotiques et utilise un microcontrôleur ATmega32u4. Elle dispose de 5 capteurs de suivi de ligne, d'un capteur de distance ultrasonique, de 4 moteurs et d'un bouclier (shield) pour les connexions sans fil. Elle est souvent utilisée pour les projets robotiques éducatifs ou pour les projets nécessitant un contrôle précis des mouvements [11].

1.2.11. **Arduino Nano Every** : c'est une version mise à jour de l'Arduino Nano classique, équipée d'un microcontrôleur ATMega4809 plus récent. Elle dispose de 20 broches d'entrée/sortie numériques et 12 broches d'entrée analogique, ainsi que d'une fréquence d'horloge de 20 MHz [12].

1.2.12. **Arduino Portenta H7** : cette carte est basée sur le microcontrôleur STM32H747XI et dispose d'un processeur ARM Cortex-M7 double cœur, ainsi que d'un co-processeur Cortex-M4. Elle est équipée d'un grand nombre de broches d'entrée/sortie numériques et analogiques, ainsi que de connexions sans fil telles que le WiFi et le Bluetooth [13].

1.2.13. **Arduino MKR Vidor 4000** : cette carte est équipée d'un FPGA (Field Programmable Gate Array) intégré qui permet une programmation personnalisée et une accélération matérielle. Elle utilise un microcontrôleur SAMD21 pour les fonctions de base, et dispose de connexions sans fil telles que le WiFi et le Bluetooth [14].

1.2.14. **Arduino Nano RP2040 Connect** : cette carte utilise le microcontrôleur RP2040 de Raspberry Pi, qui offre des performances élevées et une faible consommation d'énergie. Elle dispose de connexions sans fil telles que le WiFi et le Bluetooth, ainsi que de plusieurs broches d'entrée/sortie numériques et analogiques [15].

1.2.15. **Arduino Tian** : cette carte utilise un microcontrôleur ARM Cortex-A8 et un processeur ATMega32u4 pour une connectivité WiFi et Ethernet intégrée. Elle dispose également d'une entrée HDMI et de 32 Go de stockage eMMC intégré[16].



Figure 2 : Les types d'Arduino.:

1.3. Arduino uno :

1.3.1. Description :

L'Arduino Uno est l'une des cartes de développement les plus populaires de la famille Arduino. Elle est basée sur le microcontrôleur ATmega328P d'Atmel et est équipée de 14 broches d'entrée/sortie numériques, de 6 broches d'entrée/sortie analogiques et d'un port USB pour la programmation et la communication série [2].

La carte Arduino Uno comprend également un connecteur d'alimentation, un oscillateur à cristal, un bouton de réinitialisation et une diode électroluminescente (LED) qui indique l'état de la carte [2].

La plupart des broches numériques de l'Arduino Uno peuvent être configurées comme des entrées ou des sorties, ce qui permet de les utiliser pour connecter différents types de capteurs, de LED et d'autres périphériques électroniques. Les broches analogiques quant à elles permettent de mesurer des signaux analogiques tels que des tensions ou des courants.

La programmation de la carte Arduino Uno se fait à l'aide de l'IDE Arduino, qui offre aux utilisateurs la possibilité de créer des programmes en langage C/C++ et de les télécharger sur la carte via le port USB. Ces programmes peuvent être développés pour contrôler divers types de périphériques électroniques, communiquer avec des ordinateurs et d'autres cartes Arduino, effectuer des calculs et des opérations logiques, ainsi que pour interagir avec le monde extérieur.

En raison de son accessibilité, de sa facilité d'utilisation et de sa large communauté de développeurs et de fabricants de périphériques électroniques, la carte Arduino Uno est largement utilisée dans le domaine de l'éducation, des projets électroniques de bricolage et des applications industrielles. De nombreux projets et tutoriels en ligne sont disponibles pour aider les débutants et les professionnels à utiliser la carte Arduino Uno pour leurs propres projets [3].

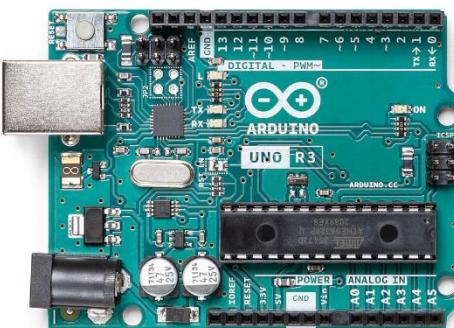


Figure 3 : la carte Arduino Uno.

- a. **Microcontrôleur** : Le microcontrôleur ATmega328P est un microcontrôleur 8 bits d'Atmel, fonctionnant à une fréquence de 16 MHz. Il est doté de 32 Ko de mémoire

flash, 2 Ko de RAM et 1 Ko d'EEPROM pour le stockage des programmes, des données et des paramètres [17].

- b. **Connecteur d'alimentation** : La carte Arduino Uno peut être alimentée soit par une source d'alimentation externe, soit par le port USB. Son connecteur d'alimentation accepte des tensions de 7 à 12 volts en courant continu [18].
- c. **Ports d'entrée/sortie numériques** : La carte Arduino Uno dispose de 14 broches d'entrée/sortie numériques numérotées de 0 à 13. Ces broches sont utilisées pour connecter des périphériques électroniques numériques tels que des LED, des boutons, des relais, des capteurs et des moteurs [19].
- d. **Ports d'entrée/sortie analogiques** : La carte Arduino Uno possède également 6 broches d'entrée/sortie analogiques numérotées de A0 à A5, qui permettent de mesurer des signaux analogiques tels que des tensions ou des courants [19].
- e. **Communication série** La carte Arduino Uno est équipée d'un port USB qui permet la programmation et la communication série avec un ordinateur. Elle peut également communiquer avec d'autres cartes Arduino et périphériques électroniques via les broches TX et RX pour les interfaces série [19].
- f. **IDE Arduino** : L'IDE Arduino est un logiciel open-source disponible en téléchargement gratuit. C'est un environnement de développement intégré qui permet aux utilisateurs de créer, modifier et télécharger des programmes sur la carte Arduino Uno [20].
- g. **Compatibilité** : La carte Arduino Uno est compatible avec une large gamme de périphériques électroniques et de capteurs, ce qui en fait une plateforme de développement polyvalente pour de nombreux projets électroniques tels que la domotique, la robotique, les objets connectés, les systèmes de surveillance et les systèmes de contrôle industriel [21].

1.4. Les composants de la carte Arduino uno :

La carte Arduino Uno est une carte de développement électronique open-source basée sur un microcontrôleur AVR ATmega328P. Elle est équipée de plusieurs composants électroniques qui lui permettent de fonctionner et de communiquer avec d'autres périphériques externes. Voici une liste des composants principaux de la carte Arduino Uno :

- 1.4.1. Microcontrôleur ATmega328P** : Il s'agit du cerveau de la carte, responsable de l'exécution du code et de la gestion des entrées/sorties [2].
- 1.4.2. Oscillateur** : Il fournit une horloge pour le microcontrôleur, permettant ainsi de synchroniser les opérations de la carte [2].
- 1.4.3. Port USB** : Il permet de connecter la carte à un ordinateur pour télécharger du code et communiquer avec d'autres périphériques externes [2].
- 1.4.4. Régulateur de tension** : Il régule la tension d'entrée de la carte pour la maintenir à 5 volts, alimentant ainsi le microcontrôleur et les autres composants électroniques [2].

- 1.4.5. Broches d'E/S** : Ce sont des broches numériques et analogiques qui permettent à la carte de communiquer avec des capteurs, des actionneurs et d'autres périphériques externes [2].
- 1.4.6. LED d'alimentation** : Elle indique que la carte est sous tension [2].
- 1.4.7. Bouton de reset** : Il permet de réinitialiser le microcontrôleur et de redémarrer le programme [2].
- 1.4.8. Connecteur d'alimentation** : Il permet d'alimenter la carte à partir d'une source externe telle qu'une pile ou un adaptateur secteur [2].
- 1.4.9. LED intégrée** : Elle est connectée à la broche 13 et peut être utilisée pour indiquer l'état d'une sortie numérique [2].
- 1.4.10. Condensateurs** : Ils sont utilisés pour stabiliser la tension et filtrer le bruit électrique [2].
- 1.4.11. Résistances** : Elles sont utilisées pour limiter le courant ou diviser la tension dans les circuits [2].
- 1.4.12. Diodes** : Elles permettent de contrôler la direction du courant électrique [1].
- 1.4.13. Connecteurs de broches** : Ils facilitent la connexion de fils et de composants externes à la carte [2].
- 1.4.14. Fuseau horaire** : Il permet de régler l'horloge interne du microcontrôleur pour les opérations liées au temps [2].
- 1.4.15. Mémoire flash** : Elle stocke le programme Arduino et les données temporaires [2].
- 1.4.16. Mémoire EEPROM** : Elle permet de stocker des données non volatiles qui restent même après la mise hors tension [2].

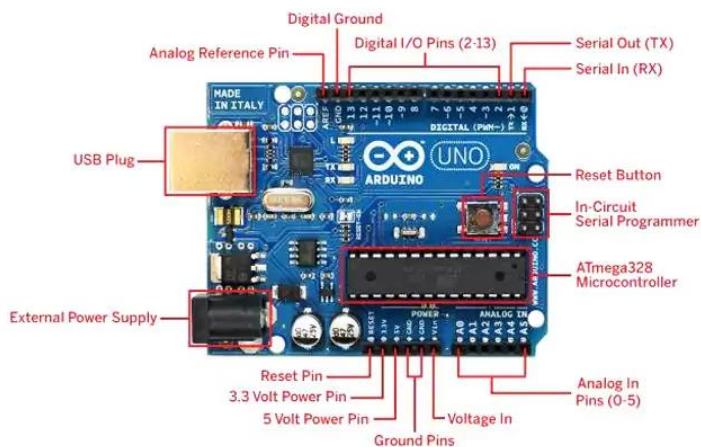


Figure 4 : Les composants de la carte Arduino uno.

1.5. L'IDE Arduino :

L'IDE Arduino (Integrated Development Environment) est un logiciel utilisé pour programmer les cartes Arduino et développer des projets électroniques. Il fournit un

environnement de développement convivial et intuitif, permettant aux utilisateurs de créer et de télécharger du code sur leur carte Arduino.

1.5.1. Interface utilisateur : L'IDE Arduino présente une interface utilisateur simple et claire, avec un éditeur de code intégré, une console de messages, une barre d'outils et des menus pour accéder aux différentes fonctionnalités [18].

1.5.2. Éditeur de code : L'éditeur de code de l'IDE Arduino offre des fonctionnalités telles que la coloration syntaxique, l'indentation automatique, la complétion automatique et la numérotation des lignes pour faciliter la programmation [19].

1.5.3. Bibliothèques : L'IDE Arduino est livré avec une bibliothèque standard comprenant de nombreuses fonctions et exemples de code prédéfinis, permettant aux utilisateurs de gagner du temps lors du développement de leurs projets. Il est également possible d'ajouter des bibliothèques tierces pour étendre les fonctionnalités de l'IDE [20].

1.5.4. Compilation et téléchargement : L'IDE Arduino utilise un compilateur pour transformer le code source en langage machine compréhensible par la carte Arduino. Il dispose également d'une fonction de téléchargement qui permet d'envoyer le code compilé sur la carte Arduino via un câble USB ou une interface de programmation spécifique [21].

1.5.5. Débogage : L'IDE Arduino ne propose pas de fonctionnalités de débogage avancées telles que des points d'arrêt ou une exécution pas à pas. Cependant, des messages de débogage peuvent être affichés dans la console de l'IDE pour faciliter le processus de débogage [22].

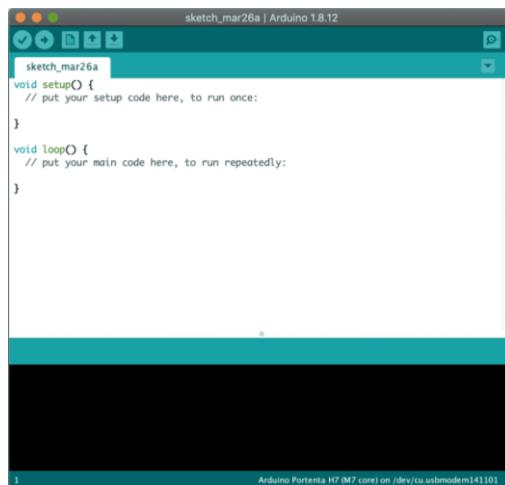


Figure 5 : Arduino IDE

1.6. Capteur de flamme :

Un capteur de flamme est un capteur optique qui détecte la présence de flammes dans son champ de vision. Il utilise généralement une photodiode à avalanche ou un phototransistor pour détecter la lumière émise par la flamme, qui est ensuite convertie en un signal électrique pour être traitée par le microcontrôleur [23].

- a. **Fonctionnement** : Les capteurs de flamme sont sensibles aux rayonnements infrarouges et ultraviolets émis par les flammes. Lorsqu'un rayonnement est détecté, le capteur génère un signal électrique qui est ensuite traité par le microcontrôleur [24].
- b. **Types de capteurs** : Il existe plusieurs types de capteurs de flamme, notamment les capteurs infrarouges, les capteurs ultraviolets et les capteurs combinés. Les capteurs infrarouges sont les plus courants et sont souvent utilisés dans les détecteurs de fumée et les systèmes de sécurité incendie [25].
- c. **Applications** : Les capteurs de flamme sont couramment utilisés dans les applications de sécurité incendie, de protection des équipements, de contrôle de processus et d'automatisation industrielle. Ils peuvent être utilisés pour détecter les flammes dans les chaudières, les incinérateurs, les cheminées, les cuisinières, les machines industrielles et autres équipements [26].
- d. **Caractéristiques** : Les capteurs de flamme sont généralement compacts, fiables et faciles à utiliser. Ils sont souvent équipés d'un seuil de détection réglable pour permettre une sensibilité plus fine ou plus grossière en fonction des besoins. Certains capteurs sont également équipés d'un système d'auto calibration pour améliorer leur précision [25].
- e. **Limitations** : Les capteurs de flamme ne sont pas adaptés à toutes les situations et présentent certaines limitations. Par exemple, ils peuvent avoir du mal à détecter les flammes dans des environnements très lumineux ou très sombres. Ils peuvent également être affectés par les interférences électromagnétiques, la saleté, la poussière et les obstructions [25].

En conclusion, les capteurs de flamme sont des capteurs optiques qui détectent la présence de flammes en convertissant la lumière émise par la flamme en un signal électrique. Ils sont largement utilisés dans les applications de sécurité incendie, de protection des équipements et d'automatisation industrielle pour détecter les flammes et activer des actions automatiques [24][25][26].



Figure 6 : Capteur de flamme.

1.7. Capteur de lumière TEMT6000 :

Le capteur de lumière TEMT6000 est un type de capteur photoélectrique qui mesure la quantité de lumière ambiante dans un environnement donné. Il est utilisé dans une variété d'applications, notamment pour réguler la luminosité de l'écran d'affichage d'un téléphone portable, pour régler l'éclairage dans une pièce en fonction de la luminosité ambiante, ou encore pour contrôler l'éclairage extérieur d'une maison en fonction de la lumière du jour [27]. Informations supplémentaires sur ce capteur :

- a. **Fonctionnement** : Le capteur de lumière TEMT6000 utilise une diode photoélectrique pour détecter la lumière ambiante, qui est ensuite convertie en une tension proportionnelle à l'intensité lumineuse. La plage de mesure typique est de 0 à 1000 lux [27].
- b. **Sensibilité** : Le capteur de lumière TEMT6000 est sensible à la lumière visible, c'est-à-dire aux longueurs d'onde comprises entre 400 et 700 nm. Sa sensibilité peut être ajustée à l'aide d'un potentiomètre présent sur le module [27].
- c. **Avantages** : Le capteur de lumière TEMT6000 est peu coûteux, facile à utiliser et peut être alimenté en tension continue entre 2,7V et 5,5V. Il peut être utilisé pour mesurer l'intensité lumineuse ambiante dans diverses applications électroniques [27].
- d. **Limitations** : Le capteur de lumière TEMT6000 peut être sensible aux variations de température et d'humidité, ce qui peut affecter sa précision. De plus, sa plage de mesure est limitée à 0 à 1000 lux, ce qui peut ne pas être suffisant pour certaines applications [27].
- e. **Applications** : Le capteur de lumière TEMT6000 est couramment utilisé dans des projets électroniques tels que la régulation de la luminosité de l'écran d'un téléphone portable en fonction de l'éclairage ambiant, la détection de la lumière dans des applications de domotique ou encore la mesure de la lumière dans des projets d'art interactif [27].

Le capteur de lumière TEMT6000 est un capteur de lumière ambiante qui peut être utilisé pour mesurer l'intensité lumineuse dans diverses applications électroniques. Bien qu'il présente certaines limitations, il reste un outil utile pour les projets électroniques impliquant la mesure de la lumière ambiante.



Figure 7 : capteur de lumière TEMT6000.

1.8. Capteur de son KY-038 :

Le capteur de son KY-038 est un petit module électronique qui permet de détecter les variations de pression acoustique dans l'environnement et de les convertir en signaux électriques. Ce capteur utilise un microphone électret qui est sensible aux ondes sonores dans une plage de fréquences allant de 50 Hz à 10 kHz [28].

- a. **Fonctionnement** : Le capteur KY-038 utilise un microphone pour détecter les sons et les vibrations, qui sont ensuite amplifiés et traités par un circuit intégré LM393. Lorsque le son ou la vibration atteint un certain niveau, le circuit intégré génère une sortie numérique qui peut être utilisée pour déclencher des actions [28].
- b. **Sensibilité** : Le capteur KY-038 est sensible aux sons et aux vibrations dans une plage de fréquences spécifique, généralement entre 20 Hz et 20 kHz. Sa sensibilité peut être ajustée à l'aide d'un potentiomètre présent sur le module [28].
- c. **Avantages** : Le capteur KY-038 est peu coûteux, facile à utiliser et peut être alimenté en tension continue entre 3,3V et 5V. Il peut être utilisé pour détecter les sons et les vibrations, ce qui en fait un capteur polyvalent pour les projets électroniques [28].
- d. **Limitations** : Comme tout microphone, le capteur KY-038 peut être sensible aux bruits environnants, ce qui peut entraîner des fausses détections. Il est également important de noter que le capteur ne peut pas être utilisé pour mesurer la pression acoustique (en décibels) [28].
- e. **Applications** : Le capteur KY-038 est couramment utilisé dans des projets électroniques tels que la détection de claquements de mains, la détection de bruits de moteurs, la détection de mouvements de porte et la détection de bruits d'animaux [28].

Le capteur KY-038 est un capteur de son et de vibrations qui peut être utilisé pour détecter les sons et les vibrations dans une plage de fréquences spécifique. Bien qu'il présente des limites et des précautions d'utilisation, il reste un outil utile pour les projets électroniques impliquant la détection de sons et de vibrations.

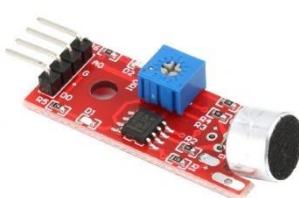


Figure 8 : Capteur de son KY-038.

1.9. Capteur de gaz MQ-2 :

Le capteur de gaz MQ-2 est un module électronique qui permet de détecter la présence de différents gaz dans l'air ambiant. Ce capteur utilise un élément sensible qui réagit aux gaz combustibles, tels que le méthane, le propane, le butane, le gaz naturel, ainsi qu'aux gaz nocifs comme le monoxyde de carbone (CO) et l'ammoniac (NH₃) [29].

- a. **Sensibilité** : Le capteur MQ-2 est sensible à différents types de gaz combustibles et toxiques, y compris le méthane, le propane, le butane, le gaz naturel, le monoxyde de carbone (CO) et l'ammoniac (NH₃). Sa sensibilité peut être ajustée en fonction de la résistance de la charge [29].
- b. **Fonctionnement** : Le capteur MQ-2 utilise un élément chauffant en combinaison avec une couche de matériau sensible au gaz, qui peut être un oxyde métallique. La présence de gaz provoque une réaction chimique qui modifie la résistance de la couche de matériau sensible. Cette variation de la résistance est ensuite convertie en une tension de sortie par le circuit de traitement du signal [30].
- c. **Avantages** : Le capteur MQ-2 est peu coûteux, facile à utiliser et peut être alimenté en tension continue entre 5V et 15V. Il peut être utilisé pour détecter différents types de gaz, ce qui en fait un capteur polyvalent [31].
- d. **Limitations** : Le capteur MQ-2 peut être sensible à d'autres gaz présents dans l'air, ce qui peut entraîner des fausses détections. Il est également important de noter que la durée de vie du capteur peut être limitée en raison de l'exposition à des gaz corrosifs [30].
- e. **Applications** : Le capteur MQ-2 est couramment utilisé dans des applications telles que la surveillance de la qualité de l'air, la détection de fuites de gaz dans les maisons et les bâtiments, la détection d'incendies et la surveillance de la pollution de l'air [29] [31].

Le capteur MQ-2 est un capteur de gaz polyvalent qui peut être utilisé pour détecter différents types de gaz combustibles et toxiques. Bien qu'il présente des limites et des précautions d'utilisation, il reste un outil utile pour la surveillance de la qualité de l'air et la détection de fuites de gaz.



Figure 9 : Capteur de Gaz MQ-2.

1.10. GSM 800L :

Le module GSM 800L est un module de communication sans fil compact et facile à intégrer, largement utilisé dans divers projets électroniques pour transmettre des données à distance via un réseau de téléphonie mobile [32]. Il offre plusieurs fonctionnalités et peut être connecté à un microcontrôleur ou à un ordinateur via une communication série (UART) ou une interface USB [33].

- a. **Fréquences de bande** : Le module GSM 800L prend en charge les fréquences de bande GSM 850/900/1800/1900 MHz, ce qui lui permet de fonctionner dans la plupart des réseaux de téléphonie mobile dans le monde entier [33].
- b. **Fonctionnalités** : Ce module offre plusieurs fonctionnalités, notamment l'envoi et la réception de SMS, l'accès à Internet, la transmission de données en temps réel et la géolocalisation [33]. Il peut également être configuré pour déclencher des alertes en cas de détection de certains événements, ce qui en fait un outil polyvalent pour les projets de communication sans fil.
- c. **Antenne** : L'utilisation d'une antenne externe est nécessaire pour assurer une bonne réception des signaux du réseau de téléphonie mobile. Il est important de sélectionner une antenne appropriée en fonction des fréquences de bande utilisées et de la zone de couverture souhaitée [33].
- d. **Consommation d'énergie** : En termes de consommation d'énergie, le module GSM 800L est relativement économique, ce qui en fait un choix populaire pour les projets alimentés par batterie. Il peut fonctionner avec une tension d'alimentation comprise entre 3,4 et 4,5 V [33].
- e. **Commandes AT** : La programmation du module GSM 800L se fait à l'aide de commandes AT standard. Ces commandes permettent de configurer les paramètres du module et d'effectuer des opérations telles que l'envoi et la réception de SMS [33].

Le module GSM 800L est une solution simple et pratique pour les projets électroniques nécessitant une communication sans fil à distance. Il offre plusieurs fonctionnalités utiles et est facile à intégrer dans une variété de projets électroniques grâce à son interface standard et à ses commandes [33].



Figure 10 : Le GSM 800L.

1.11. Camera :

Une "camera Arduino" est un dispositif qui permet de capturer des images ou des vidéos en utilisant une carte Arduino. Ce type de caméra est souvent utilisé dans des projets électroniques qui nécessitent une surveillance visuelle en temps réel ou une reconnaissance d'images automatisée [34].

- a. **Types de caméras** : Il existe différents types de caméras Arduino disponibles, allant de petits modules à basse résolution à des caméras haute définition plus sophistiquées. Les types les plus courants sont les caméras OV7670, OV2640 et le Raspberry Pi Camera [35].
- b. **Interface** : La caméra Arduino est généralement connectée à la carte Arduino via une interface de communication série telle que l'I2C ou le SPI. Cela permet à la carte Arduino de communiquer avec la caméra et de recevoir les données de l'image ou de la vidéo [35].
- c. **Alimentation** : La caméra Arduino peut être alimentée via la carte Arduino ou via une source d'alimentation externe. Il est important de vérifier les exigences de puissance spécifiques de chaque caméra pour s'assurer qu'elle est correctement alimentée [35].
- d. **Logiciel** : Pour utiliser une caméra Arduino, il est généralement nécessaire de télécharger et d'installer une bibliothèque ou un programme spécifique pour la caméra. Certains modèles peuvent également nécessiter un logiciel tiers pour la capture et le traitement des images [35].
- e. **Applications** : Les caméras Arduino sont souvent utilisées dans des projets de surveillance, de robotique, de vision par ordinateur, de reconnaissance de formes, de sécurité, de suivi d'objets, etc [35].
- f. **Limitations** : Les caméras Arduino ont souvent des limitations en termes de résolution, de vitesse de capture, de qualité d'image et de traitement des données. Il est important de sélectionner la caméra appropriée pour le projet spécifique et de comprendre ses capacités et limitations [35].

Une caméra Arduino est une solution pratique pour capturer des images et des vidéos dans des projets électroniques, offrant une flexibilité et une facilité d'intégration avec une carte Arduino. Cependant, il est important de sélectionner la caméra appropriée pour le projet spécifique et de comprendre ses capacités et limitations.



Figure 11 : Camera Arduino

1.12. Le module Bluetooth HC-05 :

Le module Bluetooth HC-05 est couramment utilisé en combinaison avec l'Arduino pour créer des projets de communication sans fil. Cette utilisation répandue est due aux capacités de communication sans fil offertes par le module HC-05, ainsi qu'à sa compatibilité avec l'Arduino, qui est une plateforme populaire pour les projets électroniques [36].

- a. **Connexion** : Pour connecter facilement le module HC-05 à l'Arduino, vous pouvez utiliser une connexion série (UART). Assurez-vous d'utiliser une broche d'alimentation appropriée sur l'Arduino, telle que la broche 3,3 V sur l'Arduino Uno, car le module HC-05 fonctionne avec une tension d'alimentation de 3,3 V [36].
- b. **Bibliothèque** : Pour utiliser le module HC-05 avec l'Arduino, il est effectivement recommandé de télécharger et d'installer la bibliothèque "SoftwareSerial" dans l'IDE Arduino [37]. Cette bibliothèque permet de créer une deuxième liaison série distincte de la liaison série principale utilisée pour la communication avec l'ordinateur, facilitant ainsi la communication avec le module HC-05. La bibliothèque "SoftwareSerial" offre également la possibilité de configurer les broches d'entrée/sortie de l'Arduino qui seront utilisées pour la communication avec le module HC-05. Vous pouvez définir ces broches en utilisant la fonction "SoftwareSerial(rxPin, txPin)", où "rxPin" représente la broche de réception et "txPin" représente la broche d'émission [37]. Cela permet d'assurer une connexion correcte entre l'Arduino et le module HC-05.
- c. **Configuration** : Une fois que le module HC-05 est correctement connecté à l'Arduino et que la bibliothèque "SoftwareSerial" est installée, vous pouvez communiquer avec le module en utilisant les commandes AT. Les commandes AT sont des instructions spécifiques permettant de configurer différents paramètres du module HC-05, tels que son nom, son code PIN, sa vitesse de transmission, et bien d'autres [37].
- d. **Communication** : Pour envoyer des données sans fil entre l'Arduino et un autre périphérique Bluetooth (par exemple, un téléphone portable), il est possible d'utiliser des commandes série simples pour envoyer et recevoir des données via la liaison Bluetooth. Les données peuvent être envoyées en utilisant la fonction "Serial.write()" sur l'Arduino et reçues en utilisant la fonction "Serial.read()" [37].
- e. **Connexions multiples** : Le module HC-05 peut également être utilisé pour créer des connexions sans fil entre plusieurs Arduinos. Dans ce cas, il est nécessaire d'avoir un module HC-05 configuré en mode maître pour initier la connexion et des modules configurés en mode esclave pour se connecter au module maître [37].

f. **Projets sans fil** : En utilisant le module HC-05 avec l'Arduino, il est possible de créer une large gamme de projets sans fil, tels que des robots télécommandés, des capteurs sans fil, des systèmes de contrôle à distance, des systèmes de surveillance et bien plus encore.



Figure 12 : Le module HC-05

1.13. Buzzer :

Un buzzer Arduino est un dispositif électronique utilisé pour produire des sons lorsqu'il est activé par une tension électrique. Il est couramment utilisé dans divers projets électroniques tels que les systèmes d'alarme, les avertisseurs, les indicateurs et les jeux [38].

- a. **Types de buzzers** : Il existe deux types de buzzers Arduino, à savoir les buzzers passifs et les buzzers actifs. Les buzzers passifs sont des composants électroniques simples qui nécessitent une tension alternative pour produire un son. Les buzzers actifs sont des modules plus complexes dotés d'un oscillateur et d'un amplificateur intégrés pour générer un son [39].
- b. **Connexion** : Pour utiliser un buzzer Arduino, vous devez généralement connecter une broche de sortie numérique de l'Arduino à l'une des broches du buzzer. En activant la broche de sortie avec une tension électrique, le buzzer émettra un son [40].
- c. **Contrôle de la fréquence et de la durée** : Vous pouvez contrôler la fréquence et la durée du son produit par le buzzer en ajustant la fréquence de la tension électrique envoyée à la broche de sortie. Cela peut être réalisé en utilisant des fonctions telles que "tone()" ou "noTone()" dans l'IDE Arduino [40].
- d. **Utilisations avancées** : Les buzzers Arduino peuvent être utilisés pour produire des sons simples tels que des bips et des clignotements, ainsi que pour reproduire des mélodies plus complexes en utilisant des séquences de fréquences spécifiques. Ils peuvent également être combinés avec d'autres composants électroniques pour créer des projets plus avancés, tels que des systèmes de surveillance sonore, des systèmes d'alarme, des indicateurs de niveau, etc. [39] [41].

En conclusion, les buzzers Arduino offrent une grande polyvalence pour produire différents types de signaux sonores dans vos projets électroniques.



Figure 13 : Le buzzer.

1.14. Capteur de mouvement SR602 :

Le capteur de mouvement SR602 est un capteur de mouvement infrarouge passif (PIR) utilisé pour détecter les mouvements dans une zone spécifique. Il est largement utilisé dans les systèmes de sécurité, les systèmes d'éclairage automatique et les systèmes de contrôle d'accès. Voici les principales caractéristiques du capteur de mouvement SR602 :

- a. **Fonctionnement** : Le capteur de mouvement SR602 utilise des capteurs PIR pour détecter les changements de chaleur dans une zone spécifique. Lorsqu'un mouvement est détecté, le capteur envoie un signal de sortie haute à l'Arduino ou à tout autre dispositif de contrôle connecté.
- b. **Portée** : La portée de détection du capteur de mouvement SR602 est d'environ 5 mètres, avec un angle de détection de 120 degrés.
- c. **Réglages** : Le capteur de mouvement SR602 dispose de deux potentiomètres pour ajuster la sensibilité et la durée de temporisation du capteur. La sensibilité peut être réglée pour détecter les mouvements de petite ou de grande amplitude, tandis que la durée de temporisation peut être réglée pour déterminer la durée pendant laquelle le signal de sortie reste haut après la détection d'un mouvement.
- d. **Alimentation** : Le capteur de mouvement SR602 fonctionne avec une tension d'alimentation de 5 volts et consomme environ 65 microampères en veille et jusqu'à 15 milliampères lorsqu'il détecte un mouvement.
- e. **Facilité d'utilisation** : Le capteur de mouvement SR602 est facile à utiliser et peut être connecté directement à l'Arduino ou à tout autre microcontrôleur compatible avec une entrée numérique. Il est également facile à installer grâce à sa petite taille et à son design compact.
- f. **Type de capteur** : Le capteur SR602 utilise un capteur infrarouge passif (PIR) à deux éléments disposés en quinconce pour fournir une zone de détection plus large.
- g. **Utilisations** : Le capteur de mouvement SR602 peut être utilisé dans une variété d'applications, telles que les systèmes d'éclairage automatique, les systèmes de sécurité à domicile, les systèmes de contrôle d'accès, etc. Il est

généralement utilisé pour détecter les mouvements et activer des actions automatiques, telles que l'allumage d'une lumière ou l'envoi d'une notification

En conclusion, le capteur de mouvement SR602 est un capteur PIR fiable et facile à utiliser, qui peut être utilisé dans une variété d'applications pour détecter les mouvements et activer des actions automatiques. Sa petite taille, sa sensibilité réglable et sa faible consommation d'énergie en font un choix populaire pour les projets de bricolage et les systèmes de sécurité à domicile.



Figure 14 : Capteur de mouvement SR602.

1.15. App Inventor :

App Inventor est un environnement de développement d'applications mobiles pour Android créé par Google et actuellement maintenu par le Massachusetts Institute of Technology (MIT) [42]. Il permet aux utilisateurs de créer des applications mobiles sans nécessiter une connaissance avancée en programmation.

En termes de taille dans App Inventor, cela peut se référer à deux choses différentes. D'une part, la taille de l'application créée avec App Inventor dépendra de la quantité de contenu et de fonctionnalités incluses. Plus l'application contient de fonctionnalités et de ressources multimédias, plus sa taille sera grande. Il est recommandé de garder les applications aussi légères que possible pour assurer une performance optimale et une expérience utilisateur fluide [43].

D'autre part, si vous parlez de la taille des écrans et des éléments de l'interface utilisateur dans l'éditeur de conception d'App Inventor, vous pouvez définir la taille des éléments en utilisant les outils disponibles dans l'éditeur, tels que les propriétés de dimensionnement et les dispositions. Vous pouvez ajuster la taille des composants pour créer une interface utilisateur adaptée à vos besoins et à vos préférences [44].

App Inventor a été initialement publié par Google le 15 décembre 2010, mais Google a mis fin à son activité le 31 décembre 2011. Depuis lors, le MIT a pris en charge le projet et fournit le support technique sous le nom "MIT App Inventor" [45].

En résumé, App Inventor est un outil convivial qui permet aux utilisateurs de créer des applications mobiles pour Android sans nécessiter de compétences avancées en programmation. La taille d'une application créée avec App Inventor dépend de son contenu

et de ses fonctionnalités, tandis que la taille des éléments de l'interface utilisateur peut être ajustée dans l'éditeur de conception.

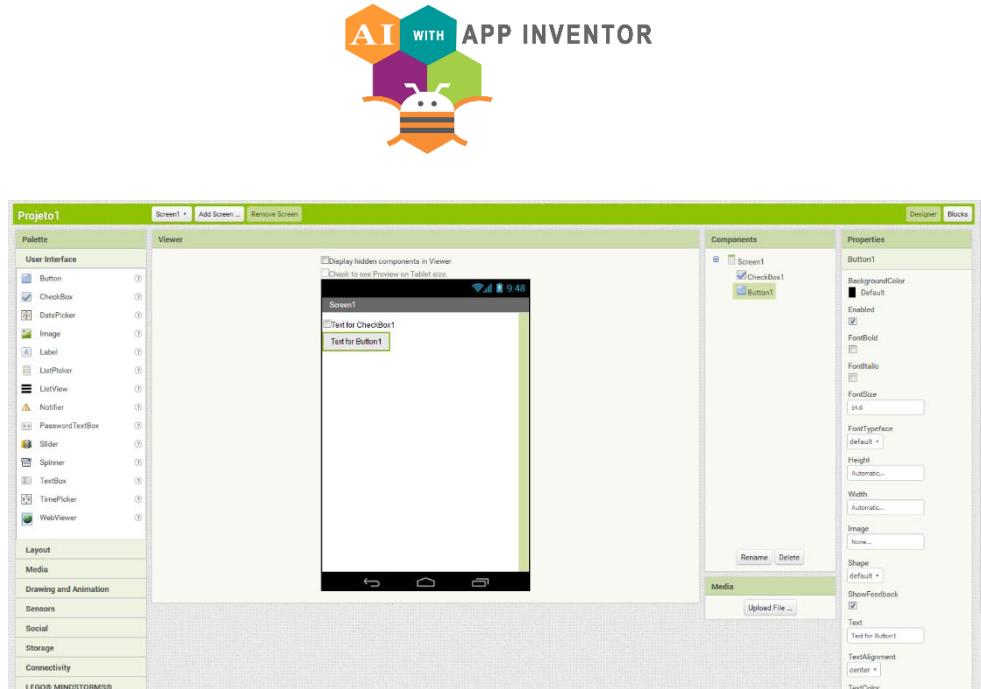


Figure 15 : App Inventor.

IV. CHAPITRE3 :

Résultat de réalisation

1. Application Asentry :

1.1. Définition :

ASENTRY est une application mobile innovante développée pour faciliter le contrôle et la gestion des capteurs d'un système de surveillance à domicile. En utilisant la connectivité Bluetooth du module HC-05, cette application permet aux utilisateurs de contrôler à distance les capteurs de leur projet de sécurité résidentielle, ainsi que d'accéder à la caméra en temps réel, le tout depuis leur téléphone portable.

L'objectif principal **d'ASENTRY** est d'offrir une solution pratique et conviviale pour renforcer la sécurité domestique. Grâce à cette application, les utilisateurs peuvent allumer et éteindre les capteurs de leur système de surveillance directement depuis leur téléphone, offrant ainsi un contrôle flexible et personnalisé. Que ce soit pour activer les capteurs lorsqu'ils quittent leur domicile ou pour les désactiver à leur retour, **ASENTRY** permet une gestion facile et pratique du système de surveillance.

La sécurité est une préoccupation primordiale dans tout système de surveillance, et **ASENTRY** répond à cette exigence en intégrant un code PIN personnalisé. Ce code PIN garantit que seules les personnes autorisées peuvent accéder à l'application et contrôler les capteurs. Ainsi, les utilisateurs bénéficient d'une expérience sécurisée et peuvent avoir la tranquillité d'esprit en sachant que leur système de surveillance est protégé contre les accès non autorisés.

En résumé, **ASENTRY** est une application mobile conçue pour faciliter le contrôle à distance des capteurs d'un système de surveillance à domicile. Grâce à la connectivité Bluetooth du module HC-05, les utilisateurs peuvent allumer et éteindre les capteurs via leur téléphone portable, ainsi que visualiser la caméra en temps réel. Avec sa sécurité renforcée par un code PIN personnalisé, **ASENTRY** offre aux utilisateurs un moyen pratique et sécurisé de gérer leur système de surveillance, améliorant ainsi la sécurité et la tranquillité d'esprit à domicile.

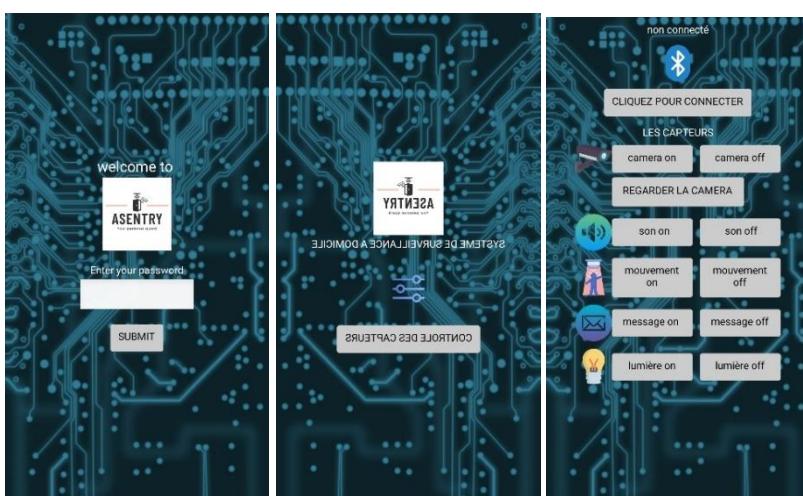


Figure 16 : captures de l'application Asentry.



Figure 17 : LOGO de l'application.

1.2. Block de Programmation du l'application :



Figure 18 : Block de Programmation du l'application

2. Résultat de réalisation

2.1. La simulation :

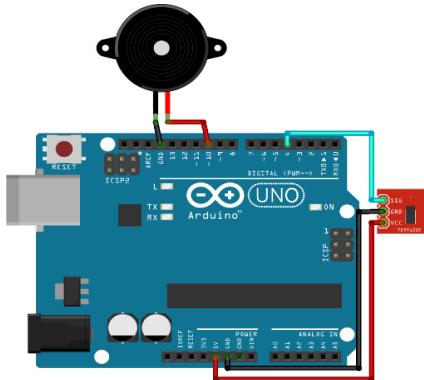


Figure 19 : Circuit de Capteur de flamme

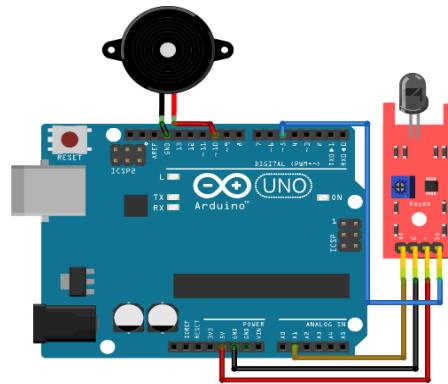


Figure 20 : Circuit de capteur TEMT6000

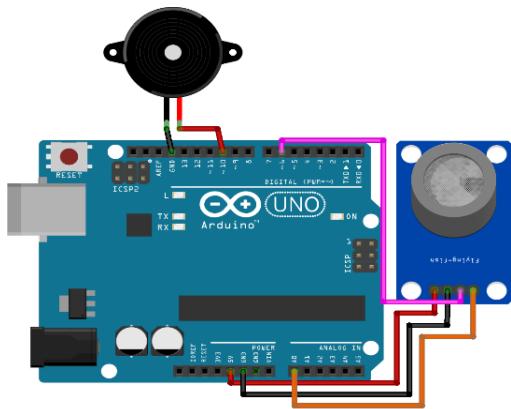


Figure 21 : Circuit de Capteur MQ-2

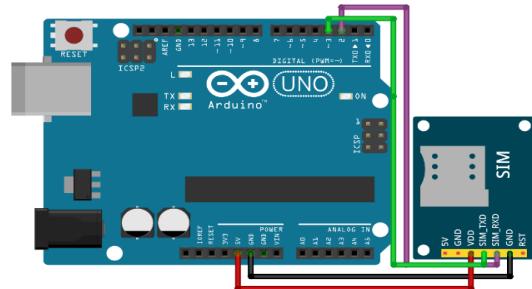


Figure 22 : Circuit de GSM 800L

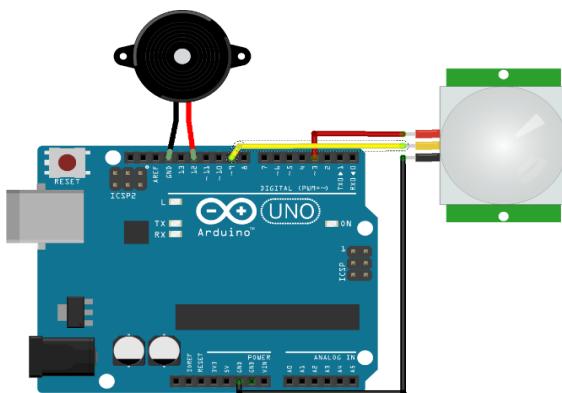


Figure 23 : Circuit de Capteur SR602

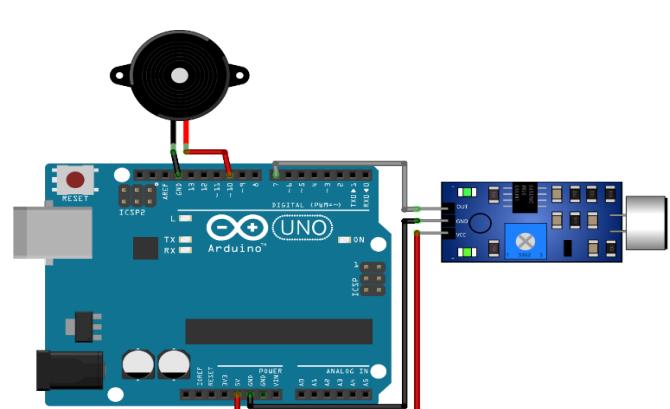


Figure 24 : Circuit de Capteur KY-038

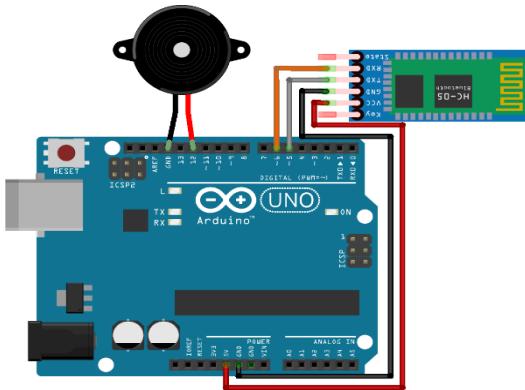


Figure 25 : Circuit de module Bluetooth HC-05

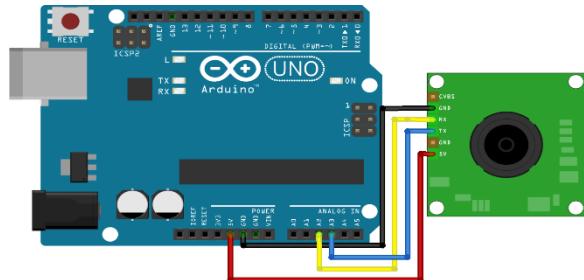


Figure 26 : Circuit de la camera

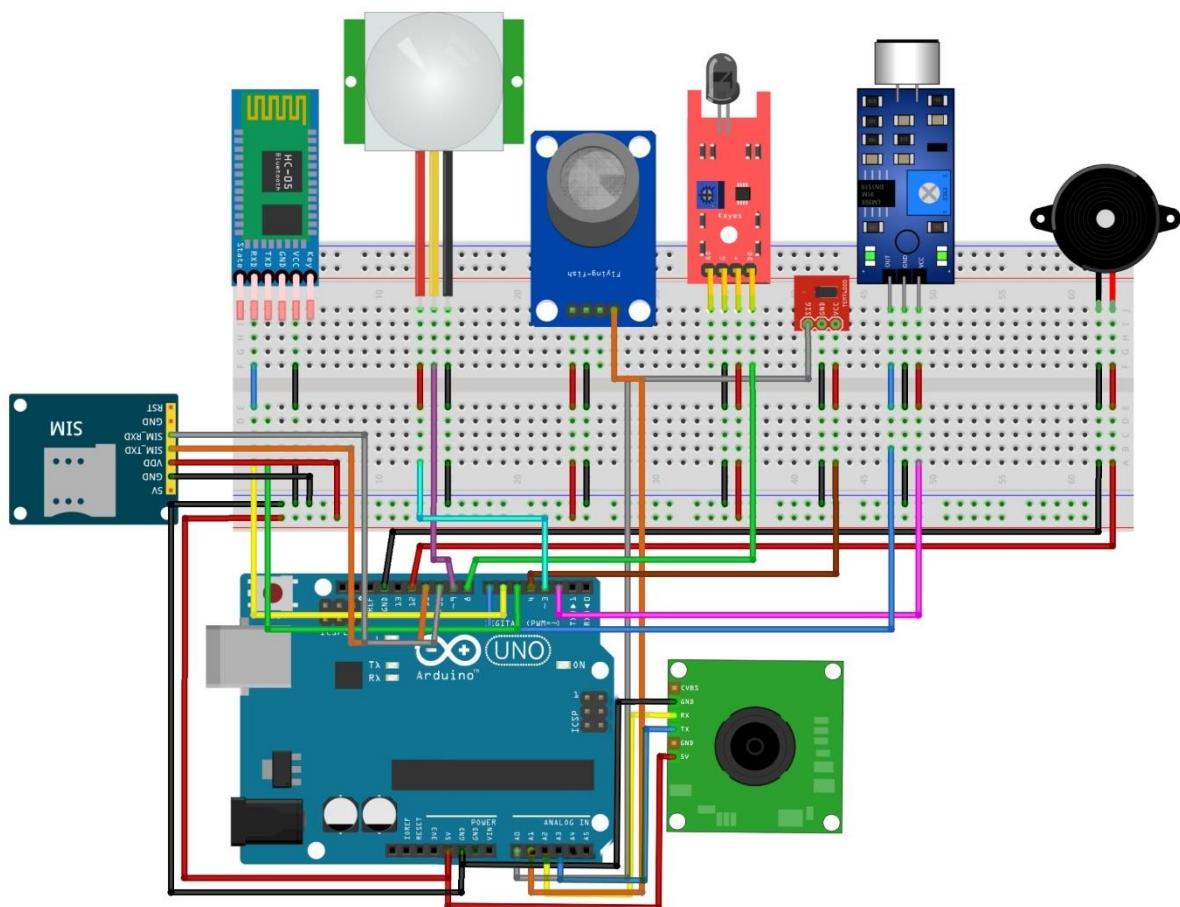


Figure 27 : Circuit de projet

2.2. Résultat final :

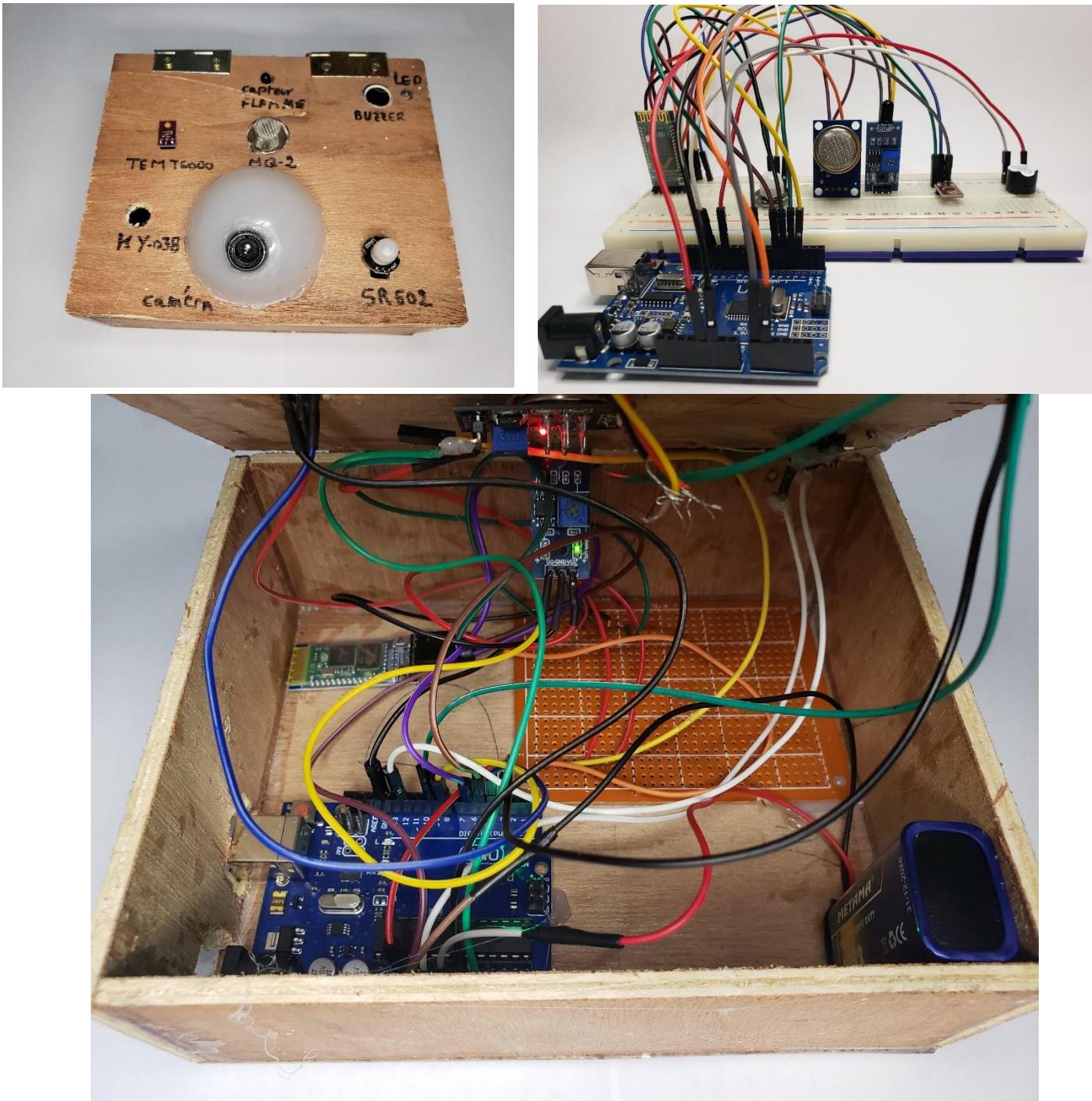


Figure 28 : la réalisation de projet

V. Conclusion :

Les systèmes de surveillance à domicile sont devenus de plus en plus populaires ces dernières années en raison de la montée de la technologie connectée et des préoccupations croissantes concernant la sécurité domestique. Ces systèmes permettent aux propriétaires de surveiller leur maison, leur famille et leurs biens à distance, en utilisant des dispositifs tels que des caméras de sécurité, des capteurs de mouvement, des détecteurs de fumée, des serrures intelligentes et des systèmes d'alarme.

Les caméras de sécurité sont l'un des éléments les plus courants d'un système de surveillance à domicile. Elles peuvent être installées à l'intérieur ou à l'extérieur de votre maison pour surveiller les activités suspectes et vous alerter en cas de mouvement ou d'activité inhabituelle. Les caméras peuvent également être équipées de fonctionnalités telles que la vision nocturne, la reconnaissance faciale, la détection de son et l'enregistrement en continu.

Les capteurs de mouvement sont un autre élément important d'un système de surveillance à domicile. Ils peuvent être installés à l'intérieur ou à l'extérieur de votre maison pour détecter les mouvements suspects et déclencher une alarme ou une alerte. Les détecteurs de fumée sont également essentiels pour détecter les incendies à un stade précoce et alerter les occupants de la maison.

Les systèmes d'alarme sont une autre composante importante d'un système de surveillance à domicile. Ils peuvent être activés manuellement ou automatiquement et déclencher une alarme sonore ou une alerte pour signaler une intrusion ou une activité suspecte. Les systèmes d'alarme peuvent également être connectés à des services de surveillance professionnels pour une sécurité accrue.

Enfin, les systèmes de surveillance à domicile peuvent être équipés de fonctionnalités avancées telles que la surveillance à distance et la connectivité en ligne. Cela permet aux propriétaires de surveiller leur maison à distance à partir de leur téléphone portable ou de leur ordinateur portable, de recevoir des alertes en temps réel et de contrôler les fonctions de sécurité à distance.

Un système de surveillance à domicile bien conçu peut offrir une tranquillité d'esprit et une sécurité accrue pour votre maison et votre famille. Il est important de choisir un système qui répond à vos besoins spécifiques et d'investir dans des produits de qualité pour assurer la fiabilité et la sécurité à long terme.

VI. ANNEXE

a) Capteur de lumière TEMT6000 :

La programmation :

```
// Définir les constantes pour les broches du capteur de luminosité TEMT6000 et du
// buzzer
const int temt6000Pin = A0;
const int buzzerPin = 9;
// Définir une variable pour le seuil de luminosité
const int threshold = 500;
void setup () {
    // Initialiser la communication série à 9600 bauds
Serial.begin (9600);
    // Définir la broche du buzzer comme sortie numérique
pinMode(buzzerPin, OUTPUT);
}
void loop () {
    // Lire la valeur analogique du capteur de luminosité (entre 0 et 1023)
int value = analogRead (temt6000Pin);
    // Afficher la valeur sur le moniteur série
Serial.println (value);
    // Si la valeur est supérieure au seuil, faire sonner le buzzer
if (value > threshold) {
    tone(buzzerPin, 1000); // Générer un son de 1000 Hz sur la broche du buzzer
    delay(100); // Attendre 100 millisecondes
    noTone(buzzerPin); // Arrêter le son sur la broche du buzzer
    delay(100); // Attendre 100 millisecondes
}
}
```

b) Capteur de flamme :

La programmation :

```
// Définir les constantes pour les broches du capteur de flamme et du buzzer
const int flamePin = 2;
const int buzzerPin = 12;
// Définir une variable pour stocker l'état du capteur de flamme
int flame;
void setup() {
    // Initialiser la communication série à 9600 bauds
    Serial.begin(9600);
    // Définir la broche du capteur de flamme comme entrée numérique
    pinMode(flamePin, INPUT);
    // Définir la broche du buzzer comme sortie numérique
    pinMode(buzzerPin, OUTPUT);
}
void loop() {
    // Lire l'état du capteur de flamme (HIGH ou LOW)
    flame = digitalRead(flamePin);
    // Afficher l'état du capteur sur le moniteur série
    Serial.print("Flame Sensor - ");
    Serial.println(flame);
    // Si l'état du capteur est LOW, cela signifie qu'une flamme est détectée
    if (flame == LOW) {
        tone(buzzerPin, 1000); // Générer un son de 1000 Hz sur la broche du buzzer
        delay(100); // Attendre 100 millisecondes
        noTone(buzzerPin); // Arrêter le son sur la broche du buzzer
        delay(100); // Attendre 100 millisecondes
    }
}
```

c) Capteur de gaz MQ-2 :

La programmation :

```
// Définir les constantes pour les broches du capteur de gaz MQ2 et du buzzer
const int PIN_MQ2 = A1;
const int buzzerPin = 13;
// Définir une variable pour stocker la valeur analogique du capteur de gaz (entre 0 et
1023)
int value;
// Définir une variable pour le seuil de détection du gaz
const int threshold = 200;
void setup() {
    // Initialiser la communication série à 9600 bauds
Serial.begin(9600);
    // Définir la broche du buzzer comme sortie numérique
pinMode(buzzerPin, OUTPUT);
    // Définir la broche du capteur de gaz comme entrée analogique
pinMode(PIN_MQ2, INPUT);
}
void loop() {
    // Lire la valeur analogique du capteur de gaz
value = analogRead(PIN_MQ2);
    // Afficher la valeur sur le moniteur série
Serial.println("VALUE - " + String(value));
Serial.println(" ");
    // Si la valeur est supérieure au seuil, cela signifie qu'un gaz est détecté
if (value > threshold) {
    tone(buzzerPin, 1000); // Générer un son de 1000 Hz sur la broche du buzzer
    delay(100); // Attendre 100 millisecondes
    noTone(buzzerPin); // Arrêter le son sur la broche du buzzer
    delay(100); // Attendre 100 millisecondes
}
}
```

d) GSM 800L :

La programmation :

```
// Inclure la bibliothèque SoftwareSerial pour communiquer avec le module GSM
#include <SoftwareSerial.h>
// Définir les constantes pour les broches du module GSM et du buzzer
const int GSM_TX = 3;
const int GSM_RX = 2;
const int BUZZER_PIN = 12;
// Créer un objet SoftwareSerial pour le module GSM
SoftwareSerial GSMSerial(GSM_TX, GSM_RX);
// Définir le numéro de téléphone et le message à envoyer en cas d'alerte
String phone_number = "+213xxxxxxxx";
String message = "Alerte ! Un signal a été détecté par les capteurs !";
void setup() {
    // Définir la broche du buzzer comme sortie numérique
    pinMode(BUZZER_PIN, OUTPUT);
    // Initialiser la communication série à 9600 bauds
    Serial.begin(9600);
    // Initialiser la communication avec le module GSM à 9600 bauds
    GSMSerial.begin(9600);
    // Attendre 5 secondes que le module GSM soit prêt
    delay(5000);
    // Envoyer la commande AT au module GSM pour vérifier sa disponibilité
    GSMSerial.println("AT");
    // Attendre 1 seconde la réponse du module GSM
    delay(1000);
    // Afficher la réponse du module GSM sur le moniteur série
    while (GSMSerial.available() > 0) {
        Serial.write(GSMSerial.read());
    }
}
void loop() {
    // Si la condition est vérifiée selon les capteurs utilisés, cela signifie qu'une alerte est déclenchée
    if /* condition à définir selon les capteurs utilisés */ {
        digitalWrite(BUZZER_PIN, HIGH); // Faire sonner le buzzer
```

```

sendSMS(phone_number, message); // Envoyer un SMS au numéro défini avec le
message défini
}
else {
digitalWrite(BUZZER_PIN, LOW); // Arrêter le buzzer
}
}

// Définir une fonction pour envoyer un SMS avec le module GSM
void sendSMS(String number, String text) {
GSMSerial.println("AT+CMGF=1"); // Passer en mode texte pour l'envoi de SMS

```

e) Capteur de mouvement SR602 :

La programmation :

```

// Définir les constantes pour les broches du capteur de mouvement et du buzzer
const int MOUVEMENT_PIN = 9;
const int BUZZER_PIN = 12;

void setup() {
// Définir la broche du capteur de mouvement comme entrée numérique
pinMode(MOUVEMENT_PIN, INPUT);
// Définir la broche du buzzer comme sortie numérique
pinMode(BUZZER_PIN, OUTPUT);
// Initialiser la communication série à 9600 bauds
Serial.begin(9600);
}

void loop() {
// Lire l'état du capteur de mouvement (HIGH ou LOW)
if (digitalRead(MOUVEMENT_PIN) == HIGH) {
// Si l'état est HIGH, cela signifie qu'un mouvement est détecté
tone(BUZZER_PIN, 1000); // Générer un son de 1000 Hz sur la broche du buzzer
delay(100); // Attendre 100 millisecondes
noTone(BUZZER_PIN); // Arrêter le son sur la broche du buzzer
delay(100); // Attendre 100 millisecondes
}
}

```

f) Capteur de son KY-038

Le montage :La programmation :

```
// Définir les constantes pour les broches du capteur de son et du buzzer
const int SON_PIN = 7;
const int BUZZER_PIN = 12;

void setup() {
    // Définir la broche du capteur de son comme entrée numérique
    pinMode(SON_PIN, INPUT);
    // Définir la broche du buzzer comme sortie numérique
    pinMode(BUZZER_PIN, OUTPUT);
    // Initialiser la communication série à 9600 bauds
    Serial.begin(9600);
}

void loop() {
    // Lire l'état du capteur de son (HIGH ou LOW)
    if (digitalRead(SON_PIN) == HIGH) {
        // Si l'état est HIGH, cela signifie qu'un son est détecté
        tone(BUZZER_PIN, 1000); // Générer un son de 1000 Hz sur la broche du buzzer
        delay(100); // Attendre 100 millisecondes
        noTone(BUZZER_PIN); // Arrêter le son sur la broche du buzzer
        delay(100); // Attendre 100 millisecondes
    }
}
```

g) module Bluetooth HC-05

La programmation :

```
// Inclure la bibliothèque SoftwareSerial pour communiquer avec le module Bluetooth
#include "SoftwareSerial.h"

// Définir les constantes pour les broches du module Bluetooth
const int BT_TX = 5;
const int BT_RX = 6;

// Créer un objet SoftwareSerial pour le module Bluetooth
SoftwareSerial mySerial(BT_TX, BT_RX);

// Définir une variable pour stocker la valeur reçue du module Bluetooth
int val;

void setup() {
    // Initialiser la communication avec le module Bluetooth à 9600 bauds
    mySerial.begin(9600);

    // Définir les broches comme sorties numériques
    pinMode(1, OUTPUT);
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(13, OUTPUT);
}

void loop() {
    if(mySerial.available()) {
        val = mySerial.read();

        // Selon la valeur reçue, allumer ou éteindre une LED
        if (val == 'a') {digitalWrite(2, HIGH);} // Allumer le capteur de son si la valeur est 'a'
        if (val == 'b') {digitalWrite(2, LOW);} // Éteindre le capteur de son si la valeur est 'b'
        if (val == 'c') {digitalWrite(3, HIGH);} // Allumer le PIR si la valeur est 'c'
        if (val == 'd') {digitalWrite(3, LOW);} // Éteindre le PIR si la valeur est 'd'
        if (val == 'e') {digitalWrite(4, HIGH);} // Allumer le capteur de lumière si la valeur est 'e'
        if (val == 'f') {digitalWrite(4, LOW);} // Éteindre le capteur de lumière si la valeur est 'f'
        if (val == 'g') {digitalWrite(13, HIGH);} // Allumer le GSM si la valeur est 'g'
        if (val == 'h') {digitalWrite(13, LOW);} // Éteindre le GSM si la valeur est 'h'
        if (val == 'i') {digitalWrite(1, HIGH);} // Allumer la caméra si la valeur est 'i'
        if (val == 'j') {digitalWrite(1, LOW);} // Éteindre la caméra si la valeur est 'j'
    }
}
```

h) le projet finale :

La programmation :

```
// Inclure la bibliothèque SoftwareSerial pour communiquer avec le module Bluetooth et
// le module GSM
#include "SoftwareSerial.h"
// Définir les constantes pour les broches du module Bluetooth
const int BT_TX = 5;
const int BT_RX = 6;
// Créer un objet SoftwareSerial pour le module Bluetooth
SoftwareSerial mySerial(BT_TX, BT_RX);
// Définir les constantes pour les broches du capteur de flamme, du capteur de lumière
// TEMT6000, du buzzer, du capteur de mouvement PIR, du capteur de gaz MQ2 et du
// capteur de son KY-038
const int flamePin = 8;
const int temt6000Pin = A0; // Pin analogique utilisée pour la lecture de la lumière
const int buzzerPin = 12;
const int PIR = 9;
const int PIN_MQ2 = A1;
const int SON_PIN = 7;
// Définir les variables pour stocker les valeurs des capteurs
int val;
int flame;
int pirVal;
int GAZ;
int valeurLumiere;
int valeurSon;
// Définir les variables pour le numéro de téléphone et le message à envoyer en cas
// d'alerte
String phone_number = "+213xxxxxxxx";
String message = "Alerte ! Un signal a été détecté par les capteurs !";
// Définir les variables pour le seuil de détection des capteurs
const int thresholdFlame = 0; // Le capteur de flamme renvoie 0 si une flamme est
// détectée
const int thresholdLumiere = 500; // Le capteur de lumière renvoie une valeur entre 0 et
// 1023
const int thresholdPIR = HIGH; // Le capteur PIR renvoie HIGH si un mouvement est
// détecté
const int thresholdGAZ = 200; // Le capteur de gaz renvoie une valeur entre 0 et 1023
```

```

const int thresholdSon = HIGH; // Le capteur de son renvoie HIGH si un son est détecté

void setup() {
    // Initialiser la communication avec le module Bluetooth à 9600 bauds
    mySerial.begin(9600);
    // Définir les broches comme sorties numériques
    pinMode(1, OUTPUT);
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(13, OUTPUT);
    // Définir les broches des capteurs comme entrées numériques ou analogiques selon le
    cas
    pinMode(PIN_MQ2, INPUT);
    pinMode(PIR,INPUT);
    pinMode(flamePin, INPUT);
    pinMode(buzzerPin, OUTPUT);
    pinMode(emt6000Pin, INPUT);
    pinMode(SON_PIN, INPUT);
    // Initialiser la communication série à 9600 bauds
    Serial.begin(9600);
    // Attendre 5 secondes que le module GSM soit prêt
    delay(5000);
    // Envoyer la commande AT au module GSM pour vérifier sa disponibilité
    mySerial.println("AT");
    // Attendre 1 seconde la réponse du module GSM
    delay(1000);
    // Afficher la réponse du module GSM sur le moniteur série
    while (mySerial.available() > 0) {
        Serial.write(mySerial.read());
    }
}

void loop() {
    // Lire les valeurs des capteurs
    flame = digitalRead(flamePin);
    pirVal = digitalRead(PIR);
    GAZ = analogRead(PIN_MQ2);
}

```

```

valeurLumiere = analogRead(temt6000Pin); // Lit la valeur de la lumière sur la broche A0
valeurSon = digitalRead(SON_PIN);

// Afficher les valeurs des capteurs sur le moniteur série
Serial.print("Flame Sensor - ");
Serial.println(flame);
Serial.print("GAZ - ");
Serial.println(GAZ);
Serial.print("Lecture de lumière :");
Serial.println(valeurLumiere);
Serial.print("PIR - ");
Serial.println(pirVal);
Serial.print("Son - ");
Serial.println(valeurSon);
Serial.println(" ");

if(mySerial.available()) {
    // Si le module Bluetooth envoie des données
    // Lire la valeur reçue du module Bluetooth
    val = mySerial.read();

    // Selon la valeur reçue, allumer ou éteindre
    if (val == 'a') {digitalWrite(2, HIGH);} // Allumer le capteur de son si la valeur est 'a'
    if (val == 'b') {digitalWrite(2, LOW);} // Éteindre le capteur de son si la valeur est 'b'
    if (val == 'c') {digitalWrite(3, HIGH);} // Allumer le PIR si la valeur est 'c'
    if (val == 'd') {digitalWrite(3, LOW);} // Éteindre le PIR si la valeur est 'd'
    if (val == 'e') {digitalWrite(4, HIGH);} // Allumer le capteur de lumière si la valeur est 'e'
    if (val == 'f') {digitalWrite(4, LOW);} // Éteindre le capteur de lumière si la valeur est 'f'
    if (val == 'g') {digitalWrite(13, HIGH);} // Allumer le GSM si la valeur est 'g'
    if (val == 'h') {digitalWrite(13, LOW);} // Éteindre le GSM si la valeur est 'h'
    if (val == 'i') {digitalWrite(1, HIGH);} // Allumer la caméra si la valeur est 'i'
    if (val == 'j') {digitalWrite(1, LOW);} // Éteindre la caméra si la valeur est 'j'
}

if (flame == thresholdFlame){
    // Si une flamme est détectée par le capteur de flamme
    tone(buzzerPin, 1000); // Générer un son de 1000 Hz sur la broche du buzzer
    delay(100); // Attendre 100 ms
    noTone(buzzerPin); // Arrêter le son sur la broche du buzzer
    delay(100); // Attendre 100 ms
    sendSMS(phone_number, message); // Envoyer un SMS au numéro défini avec le message défini
}

```

```

}

if (pirVal == thresholdPIR) {
    // Si un mouvement est détecté par le capteur PIR
    tone(buzzerPin, 1000); // Générer un son de1000 Hz sur la broche du buzzer
    delay(100); // Attendre 100 ms
    noTone(buzzerPin); // Arrêter le son sur la broche du buzzer
    delay(100); // Attendre 100 ms
    sendSMS(phone_number, message); // Envoyer un SMS au numéro défini avec le
    message défini
}

if (GAZ > thresholdGAZ) {
    // Si un gaz est détecté par le capteur MQ2
    tone(buzzerPin, 1000); // Générer un son de1000 Hz sur la broche du buzzer
    delay(100); // Attendre 100 ms
    noTone(buzzerPin); // Arrêter le son sur la broche du buzzer
    delay(100); // Attendre 100 ms
    sendSMS(phone_number, message); // Envoyer un SMS au numéro défini avec le
    message défini
}

if(valeurLumiere > thresholdLumiere) {
    // Si une lumière forte est détectée par le capteur TEMT6000
    tone(buzzerPin, 1000); // Générer un son de1000 Hz sur la broche du buzzer
    delay(100); // Attendre 100 ms
    noTone(buzzerPin); // Arrêter le son sur la broche du buzzer
    delay(100); // Attendre 100 ms
    sendSMS(phone_number, message); // Envoyer un SMS au numéro défini avec le
    message défini
}

if(valeurSon == thresholdSon) {
    // Si un son fort est détecté par le capteur KY-038
    tone(buzzerPin, 1000); // Générer un son de1000 Hz sur la broche du buzzer
    delay(100); // Attendre 100 ms
    noTone(buzzerPin); // Arrêter le son sur la broche du buzzer
    delay(100); // Attendre 100 ms
    sendSMS(phone_number, message); // Envoyer un SMS au numéro défini avec le
    message défini
}
}

```

VII. BIBLIOGRAPHIE

- [1] "Arduino - Introduction." Arduino. Consulté le 20 mai 2023. Disponible sur :
<https://www.arduino.cc/en/Guide/Introduction>
- [2] Documentation Arduino Uno : <https://www.arduino.cc/en/Guide/ArduinoUno>
- [3] Site officiel Arduino, <https://docs.arduino.cc/learn/startng-guide/getting-started-arduino>
- [4] Référence : Site officiel Arduino - <https://www.arduino.cc/en/Guide/ArduinoNano>
- [5] Arduino. (s.d.). Arduino Due. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-due>
- [6] Arduino. (s.d.). Arduino Leonardo. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-leonardo>
- [7] Arduino. (s.d.). Arduino Pro Mini. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-pro-mini>
- [8] Arduino. (s.d.). Arduino Zero. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-zero>
- [9] Arduino. (s.d.). Arduino MKR1000. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-mkr1000>
- [10] Arduino. (s.d.). Arduino Yún. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-yun>
- [11] Arduino. (s.d.). Arduino Robot. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-robot>
- [12] Arduino. (s.d.). Arduino Nano Every. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-nano-every>
- [13] Arduino. (s.d.). Arduino Portenta H7. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-portenta-h7>
- [14] Arduino. (s.d.). Arduino MKR Vidor 4000. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-mkr-vidor-4000>
- [15] Arduino. (s.d.). Arduino Nano RP2040 Connect. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-nano-rp2040-connect>
- [16] Arduino. (s.d.). Arduino Tian. Consulté le 20 mai 2023, à l'adresse
<https://store.arduino.cc/arduino-tian>
- [17] Microchip Technology, <https://www.alldatasheet.com/datasheet-pdf/pdf/241077/ATMEL/ATMEGA328P.html>

- [18] Site officiel d'Arduino : <https://www.arduino.cc/en/software>
- [19] Documentation Arduino - Éditeur de code :
<https://www.arduino.cc/en/Guide/Environment#editor>
- [20] Documentation Arduino - Bibliothèques : <https://www.arduino.cc/en/Guide/Libraries>
- [21] Documentation Arduino - Téléchargement du code :
<https://www.arduino.cc/en/Guide/Environment#uploading>
- [22] Forum Arduino - Débogage avec l'IDE Arduino : <https://forum.arduino.cc/t/how-can-i-debug-my-arduino-sketch/73049>
- [23] J. Chen et al., "Optical Flame Detection Based on Photodetection Technology," in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 828-831.
- [24] Zeng, W., Zhu, M., Zhao, L., & Han, J. (2014). Flame detection method based on optical sensors. International Journal of Distributed Sensor Networks, 2014.
- [25] Wang, C., & Li, H. (2017). An infrared flame detection algorithm based on improved clustering and self-adaptive threshold. Sensors, 17(6), 1332.
- [26] Neri, G., & Passero, A. (2016). Flame monitoring in industrial burners: A review. IEEE
- [27] Vishay Semiconductor. "TEMT6000 Ambient Light Sensor Datasheet." Consulté le 20 mai 2023. Disponible sur : <https://www.vishay.com/docs/81521/temt6000.pdf>.
- [28] Keyes. "KY-038 Microphone Sound Sensor Module." Consulté le 20 mai 2023. Disponible sur : <https://www.theengineeringprojects.com/wp-content/uploads/2017/09/KY-038-Microphone-Sound-Sensor-Module.pdf>.
- [29] Pardeep Kumar, Sandeep Kumar, and Harsimranjit Singh, "Gas Leak Detection and Localization Based on Wireless Sensor Networks," International Journal of Distributed Sensor Networks, vol. 11, no. 2, February 2015.
- [30] A. S. AlSalhi, T. S. Alrushoodi, and K. S. Hameed, "Design and Implementation of MQ-2 Gas Sensor for Monitoring and Detection of LPG and Methane Gas," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 3, March 2015.
- [31] N. A. Latif and N. F. Ariffin, "Design and Development of a Gas Leakage Detection and Monitoring System," 4th International Conference on Intelligent and Advanced Systems (ICIAS), 2012.
- [32] "Module GSM 800L - Overview and Applications." ElectroSchematics. Consulté le 20 mai 2023. Disponible sur : <https://www.electroschematics.com/gsm-module-sim800l/>.

[33] "SIM800L GSM/GPRS Module - Datasheet, Pinout, and Features." Components101. Consulté le 20 mai 2023. Disponible sur : <https://components101.com/modules/sim800l-gsm-module>.

[34] "Arduino Camera Tutorial: Interfacing OV7670 Camera Module with Arduino." Circuit Digest. Consulté le 20 mai 2023. Disponible sur : <https://circuitdigest.com/microcontroller-projects/arduino-camera-tutorial-interfacing-ov7670-camera-module-with-arduino>.

[35] "Guide to Arduino Camera Modules: OV7670, OV2640, Raspberry Pi Camera." Maker Pro. Consulté le 20 mai 2023. Disponible sur : <https://maker.pro/arduino/tutorial/arduino-camera-modules-guide-ov7670-ov2640-raspberry-pi-camera>.

[36] Document officiel du module Bluetooth HC-05

[37] Documentation officielle d'Arduino sur l'utilisation du module HC-05 :
<https://www.arduino.cc/en/Reference/SoftwareSerial>

[38] "Arduino Buzzer Tutorial and How to Use It with Arduino Board." Arduino Tutorial | How to Use the Buzzer with Arduino Board?, 18 Jan. 2021, www.tutorialspoint.com/arduino/arduino_buzzer.htm.

[39] "Arduino Buzzer." Arduino Project Hub,
www.create.arduino.cc/projecthub/projects/tags/buzzer.

[40] "How to Use a Buzzer (or Piezo Speaker) - Arduino Tutorial." Random Nerd Tutorials, 27 May 2019, www.randomnerdtutorials.com/guide-for-ws2812b-addressable-rgb-led-strip-with-arduino/.

[41] "Buzzer Module." Arduino, www.arduino.cc/en/Reference/Tone.

[42] "MIT App Inventor." App Inventor, www.appinventor.mit.edu/about.

[43] "Best Practices: Optimizing Your App." App Inventor,
www.appinventor.mit.edu/explore/ai2/best-practices.

[44] "Designer: User Interface Layout." App Inventor,
www.appinventor.mit.edu/explore/ai2/concept-summaries/designer-user-interface.

[45] "App Inventor History." App Inventor, www.appinventor.mit.edu/about/history.