

UNIVERSITE KASDI MERBAH OUARGLA
Faculté des Sciences Appliquées
Département de Génie Electrique



Mémoire
MASTER ACADEMIQUE
Domaine : Sciences et technologies
Filière : Génie électrique
Spécialité : Electrotechnique Industrielle
Présenté par :
Badjadi Mohammed Elamine Ghilani Yakoub

Thème :

Réalisation d'un système de contrôle pour la sécurité d'une maison intelligente

Soumis au jury composé de :

M ^r Taibi Djamel	MCB	Président	UKM Ouargla
M ^r Benmakhlouf Abdeslam	MCB	Encadreur/rapporteur	UKM Ouargla
M ^r Rezoug Redha	MCB	Examineur	UKM Ouargla

Année universitaire 2023/2024

Remerciements

Avant tout, on remercie ALLAH le Tout-puissant de nous avoir donné le courage, la volonté et la patience de mener à terme ce présent travail dans des meilleures conditions.

On tient tout d'abord à remercier notre encadreur Mr Djarah Djalal, pour avoir accepté de nous encadrer, et pour le thème qu'il nous a proposé. Ainsi que toutes ces remarques constructives qui nous ont permis d'approfondir les connaissances scientifiques.

Je remercie également les membres du jury pour l'honneur de juger notre travail.

Nous adressons également nos sincères remerciements à nos mères et à nos pères qui nous ont Donné chaque petite chose pour notre bonheur dans tous les domaines de la vie.

Nous remercions également tous les professeurs de génie électrique, en particulier les Professeurs de l'Université Kasdi Merbah, pour leurs efforts, ainsi que tous les étudiants qui

Nous ont aidés de près ou de loin et nous leur souhaitons également du succès dans leur vie Quotidienne.

Dédicaces

Bism allh alrahman alrahim

À ceux qui m'ont appris la patience et la persévérance, à mes chers parents qui ont toujours été ma source d'inspiration et de soutien, à ma chère famille qui n'a jamais été avare de me tendre un coup de main et de me soutenir, je vous dédie cet humble travail. .

À mes amis (Elkacene, Fakhri, Mostapha, Basset) partenaires de ce voyage scientifique, j'adresse mes plus hautes expressions de gratitude.

Alhamdalalah qui m'a aidé à réaliser ce travail.

Ghilani Yakoub Et Badjadi Mohammed Elamine

Résumé

Ce travail traite de la conception d'un système de surveillance de la sécurité des maisons intelligentes utilisant l'IoT. Le système comprend des appareils connectés au Web quotidiens et télécommandés, offrant confort et indépendance, en particulier pour les personnes âgées et les personnes handicapées. Le système s'appuie sur le panneau ESP32 pour traiter les données des différents capteurs, et effectue des tâches telles que l'activation des alarmes, l'éclairage et l'ouverture des portes, et envoie des alertes instantanées au propriétaire via l'application de surveillance et de contrôle de la maison Blynk.

Mots-clés = Internet des objets (IoT), web, Panneau de contrôle Esp32, Capteurs, Blynk.

الملخص

يتناول هذا العمل تصميم نظام مراقبة لأمن المنزل الذكي باستخدام تقنية إنترنت الأشياء (IoT). يتضمن النظام أجهزة يومية متصلة بالويب، يمكن التحكم فيها عن بُعد، مما يوفر راحة واستقلالية، خاصة لكبار السن وذوي الإعاقة. يعتمد النظام على لوحة ESP32 لمعالجة بيانات الحساسات المختلفة، ويقوم بمهام مثل تشغيل الإنذارات والإضاءة وفتح الأبواب، مع إرسال تنبيهات فورية إلى مالك المنزل عبر تطبيق Blynk لمراقبة المنزل والتحكم فيه عن بعد.

كلمات المفتاحية = إنترنت الأشياء (IoT)، الويب، لوحة التحكم esp32، الحساسات ، Blynk.

Abstract

This work deals with the design of a smart home security monitoring system-using IoT. The system includes every day, remote-controlled web-connected devices, providing comfort and independence, especially for older adults and people with disabilities. The system relies on the ESP32 panel to process data from different sensors, performs tasks such as activating alarms, lighting and opening doors, and sends instant alerts to the owner via the app monitoring and control of the Blynk house.

Keywords = Internet of Things (IoT), web, esp32 control panel, sensors, Blynk.

Sommaire

Introduction générale.....	2
Chapitre I : Méthodologie et matériels	
I.1. Introduction.....	5
I.2. Matériels utilisés :.....	5
I.2.1Capteur de gaz (MQ2).....	6
I.2.1.1. Les Caractéristiques du MQ2 sont :	6
I.2.1.2.Les broches du capteur MQ2.....	7
I.2.2. Capteur de température et d'humidité DHT11.....	8
I.2.2.1. Caractéristiques :.....	8
I.2.2.2. Les broches du capteur DHT11.....	8
I.2.2.3. Le protocole de communication.....	9
I.2.2.4. Les trois grandes parties d'une trame « type » sont :.....	9
I.2.3.Capteur de Ultrason.....	9
I.2.3.1.Principe de fonctionnement du capteur de ultrason.....	10
I.2.3.2. Les Caractéristiques de l'Ultrason sont :.....	11
I.2.3.3. Les broches du capteur Ultrason.....	11
I.2.4. Carte électronique ESP32.....	11
I.2.4.1. Les Caractéristiques d'ESP32-WROOM-32.....	13
I.2.4.2.ESP32 WROOM 32 GPIO(Pins).....	13
I.2.5. Servomoteur.....	14
I.2.5.1. Principe de fonctionnement du servo moteur.....	15
I.2.5.2. Les Caractéristiques du servomoteur SG90 :.....	15
I.2.6. Buzzer.....	16
I.2.6.1. Les Caractéristiques du buzzer :.....	16
I.2.7. Relais module.....	16
I.2.7.1. Les Caractéristiques du relais :.....	17
I.2.7.2. Principe de fonctionnement du relais.....	17
I.2.8. Ventilateur.....	18
I.2.9. Plaque d'essai	18
I.2.10. Les leds	18
I.2.11. Les Résistances	19
I.2.12. Fils de connexions.....	19
I.3.Partie logiciel	20
I.3.1. Définition de l'IDE	20
I.3.1.1. Description de l'IDE.....	20

I.3.1.2. Barre de menu.....	21
I.3.1.3. Fenêtre d'édition du programme	22
I.3.1.4. Zone de message des actions en cours.....	22
I.3.1.5. Définition des constantes et des variables	22
I.3.1.6. Configuration des entrées/sorties.....	23
I.3.1.7. Programmation des interactions et des comportements	23
I.3.1.8. Commentaires.....	23
I.3.1.9. Utilisation d'un ESP32 avec l'Arduino IDE	23
I.3.1.10. Configuration de l'IDE Arduino pour l'ESP32	23
I.3.1.11. Ajouter la carte ESP32 dans la base des cartes Arduino de l'IDE.....	24
I.3.1.12. Installer la carte ESP32 dans l'IDE Arduino.....	26
I.3.1.13. Sélectionner la carte ESP32	26
I.3.1.14. Étapes pour télécharger le programme	27
I.3.2. Plateforme Blynk	27
A. Blynk App.....	28
B. Blynk Server	29
C. Les bibliothèques Blynk	29
D. Fonctionnalités.....	29
E. Configuration de l'application Blynk pour ESP32.....	30
I.3.3. Fritzling.....	31
I.4.Conclsn.....	32
Chapitre II : Résultats et discussions	
II.1. Introduction.....	34
II. 2. Résultats :.....	34
II. 2.1.La lecture de différentes données telles que la température et l'humidité	34
a.Avant d'allumer le ventilateur	35
b.Quand le ventilateur tourne	36
II. 2.2.La surveillance de fuites de gaz et de fumée.....	37
1.Mesurer différentes concentrations de gaz et de fumée	38
2. Avant d'ouvrir le ventilateur et la fenêtre.....	38
3. Après avoir ouvert le ventilateur et la fenêtre.....	39
II. 2.3.Surveiller l'ouverture et la fermeture de la porte.....	40
II. 2.4. Schéma principal.....	41
II.3. Discussions :.....	42
II.3.1. La lecture des données de température et d'humidité.....	42
II.3.2. La lecture des données de fuite de gaz	43

II.3.3. La lecture des données de capteur ultrason.....	43
II.4.Conclusion.....	43
Conclusion générale.....	45
Bibliographie.....	48
Annexe.....	50

Liste des figures

Figure I. 1 : Schéma principal du projet.....	7
Figure I. 2 : Capteur de gaz MQ2.....	7
Figure I. 3 : Les broches du capteur gaz MQ2.....	8
Figure I. 4 : Capteur DHT11.....	9
Figure I. 5 : Les broches du capteur DHT11.....	9
Figure I. 6 : Les bases pour communiquer avec un DHT11.....	10
Figure I. 7 : Capteur d'Ultrason.....	11
Figure I. 8 : Début des travaux de capteur HC-SR04.....	11
Figure I. 9 : ESP32-WROOM-32.....	13
Figure I.10 : ESP32 WROOM 32 GPIO Pins.....	15
Figure I. 11 : Micro servo SG90.....	16
Figure I. 12 : Composants du servo moteur et les broches.....	16
Figure I. 13 : Buzzer.....	17
Figure I.14 : Relais module 5v.....	18
Figure I.15 : Principe de fonctionnement du relais.....	18
Figure I.16 : ventilateur 5v.....	19
Figure I.17 : Plaque d'essai (Breadboard).....	19
Figure I.18 : Les Leds.....	20
Figure I.19 : Résistance.....	20
Figure I.20 : Fils de connexions.....	21
Figure I. 21 : Interface IDE Arduino.....	22
Figure I. 22 : Barre d'action.....	22
Figure I. 23 : Zone de message des actions.....	23
Figure I. 24 : Configuration des entrées/sorties.....	24
Figure I. 25 : Configuration de l'IDE Arduino pour ESP 32 étape 1.....	25
Figure I. 26 : Configuration de l'IDE Arduino pour ESP 32 étape 2.....	26
Figure I.27 : Configuration de l'IDE Arduino pour ESP 32 étape 2.....	26
Figure I.28 : Configuration de l'IDE Arduino pour ESP 32 étape 3.....	27
Figure I. 29 : Paramétrage de la carte.....	28
Figure I. 30 : étapes de téléchargement du code.....	28
Figure I. 31 : Fonctionnement de Blynk.....	29
Figure I. 32 : Interface de Blynk.....	30
Figure I. 33 : Etapes de Configuration de l'application Blynk pour ESP32.....	31
Figure I. 34 : Fonctionnalités de l'application Blynk.....	32

Figure I. 35 : Interface de Fritzing.....	32
Figure II. 36 : Circuit réel de l'application du control de temperature et humidité.....	35
Figure II. 37 : Lecture du capteur de température et d'humidité Avant d'allumer le ventilateur.....	36
Figure II. 38 : Lecture du capteur de température et d'humidité quand le ventilateur tourne.	37
Figure II. 39 : Circuit La surveillance de fuites de gaz.....	38
Figure II. 40 : Lecture du capteur de gaz à différentes concentrations.....	39
Figure II. 41 : Lecture du capteur de gaz Avant d'ouvrir le ventilateur et la fenêtre.....	39
Figure II. 42 : Message d'avertissement du capteur de gaz dans l'application mobile Blynk..	40
Figure II.43 : Lecture du capteur de gaz Après avoir ouvert le ventilateur et la fenêtre.....	40
Figure II. 44 : Circuit du branchement du capteur à ultrason.....	41
Figure II. 45 : la lecture du capteur à ultrason.....	41
Figure II. 46 : Message d'alerte du capteur à ultrasons lorsque la porte est ouverte dans l'application mobile Blynk.....	42
Figure II. 47 : Schéma principal.....	43

Liste des tableaux

Tableau I.1 : Les broches du capteur MQ2.....	8
Tableau I.2 : Les broches du capteur DHT11.....	9
Tableau I.3 : Les broches du capteur Ultrason.....	12

Abréviations

• MCU	Microcontroller Unit
• IOT	internet of things
• SOC	A system on a chip
• IdO	internet des objets
• GPIO	A general-purpose input/output
• USB	Universal Serial Bus
• SD	Secure Digital
• UART	A universal asynchronous receiver-transmitter
• SPI	The Serial Peripheral Interface
• SDIO	Secure Digital Input Output
• LED	A light-emitting diode
• PWM	Pulse-width modulation
• I2S	Inter-IC Sound
• I2C	Inter-Integrated Circuit
• ADC	analog-to-digital converter
• DAC	digital-to-analog converter
• BLE	Bluetooth Low Energy
• GND	Ground
• FreeRTOS	Real-Time Operating System
• IDE	integrated development environment
• SDK	Software Development Kit
• OTA	Over-The-Air
• IOS	iPhone Operating system
• CPU	Central Processing Unit
• TLS	Transport Layer Security
• RTC	Real-Time Clock

INTRODUCTION GÉNÉRALE

Introduction générale

À l'ère contemporaine, l'internet des objets (IdO) s'est imposé comme une force de transformation, améliorant considérablement divers aspects de la vie quotidienne. La technologie de l'IdO connecte les appareils et les systèmes à l'internet, ce qui permet une communication et une interaction transparentes. Cette connectivité est cruciale dans de nombreux domaines tels que les soins de santé, la sécurité et la gestion de l'énergie, où les applications IoT rationalisent les opérations et améliorent l'efficacité. Parmi ces applications, la domotique intelligente est un exemple frappant de la manière dont l'IdO peut révolutionner la vie résidentielle en offrant un contrôle, une commodité et un confort accrus.

Avec la diversité des méthodes de communication humaine, les technologies de l'information et des télécommunications sont devenues une condition suffisante pour assurer une communication illimitée avec tous les habitants de la planète. La communication homme-machine ou machine-machine peut être considérée comme un nouveau type de dialogue possible. En effet, au cours de la dernière décennie, les appareils sont devenus de plus en plus intelligents, peuvent agir en fonction des profils des utilisateurs et prendre des décisions indépendantes. La maison intelligente fera partie de notre avenir. Qui ne voudrait pas d'une maison capable de gérer les tâches ménagères, d'assurer plus de confort, de sécurité et de bien-être à ses occupants, de leur faciliter la vie et de répondre à leurs besoins et désirs avant même qu'ils ne les expriment ? La domotique intelligente implique l'intégration d'appareils et de capteurs IoT pour créer un environnement connecté où diverses fonctions domestiques peuvent être surveillées et contrôlées à distance par le biais de smartphones ou d'autres appareils compatibles avec l'internet. Ce système permet aux utilisateurs de gérer les appareils électriques et électroniques tels que les lumières, les ventilateurs, les climatiseurs et les systèmes de sécurité, optimisant ainsi la consommation d'énergie et améliorant l'expérience de vie globale [1]. La mise en œuvre d'une maison intelligente repose sur les capacités de l'IdO à simuler et à faire fonctionner divers appareils domestiques. En utilisant un réseau de capteurs et d'appareils, un système domestique intelligent peut automatiser des tâches en fonction de conditions prédéfinies ou de données en temps réel. Par exemple, une webcam intégrée à des capteurs de mouvement peut renforcer la sécurité de la maison en détectant les intrus et en alertant les propriétaires, tandis qu'un thermostat intelligent peut ajuster la température en fonction de l'occupation et des préférences, ce qui permet d'économiser de l'énergie [2].

Grâce à la domotique, un système peut être contrôlé et commandé à distance. Les applications peuvent être utilisées dans divers domaines tels que le contrôle et la commande à

distance de machines, les systèmes d'alarme et de surveillance, le contrôle des portes et des fenêtres, ou l'allumage ou l'extinction de lumières ou de ventilateurs, entre autres. L'objectif principal de notre projet est de « Concevoir et mettre en œuvre une maison intelligente via un module Wi-Fi ». Cette maison comprend plusieurs systèmes qui rendent la vie agréable et optimisent l'énergie électrique. Elle peut être surveillée à distance via une application mobile ou un ordinateur, avec un focus particulier sur l'application « Blynk ». Un système domotique intelligent pratique comprend généralement plusieurs dispositifs interconnectés tels que des webcams, des humidificateurs, des ventilateurs de plafond et des climatiseurs. Ces appareils sont équipés de capteurs qui collectent des données sur divers paramètres tels que l'humidité, la température et les mouvements. Les données sont ensuite transmises à une passerelle domestique centrale, qui traite les informations et exécute des commandes pour contrôler les appareils en conséquence [3]. Par exemple, un humidificateur intelligent peut maintenir un niveau d'humidité optimal en s'allumant ou en s'éteignant en fonction des relevés d'humidité en temps réel d'un capteur. De même, un ventilateur de plafond ou un climatiseur peut être contrôlé en fonction des données de température afin d'assurer un climat intérieur confortable tout en minimisant la consommation d'énergie. L'intégration de ces dispositifs dans un système unifié améliore non seulement le confort et la commodité de l'environnement domestique, mais contribue également à d'importantes économies d'énergie [4].

Notre mémoire est structurée en deux chapitres essentiels :

Premièrement, nous entamons notre exploration avec une présentation exhaustive des composants matériels et logiciels que nous avons employés, en mettant l'accent sur leurs principes de fonctionnement. Cette section sert de fondement à notre étude, en fournissant une compréhension approfondie des outils technologiques qui ont permis la réalisation de notre système. Deuxièmement, nous présentons les résultats empiriques obtenus lors de nos expérimentations. Cette partie contient une analyse détaillée et rigoureuse des données recueillies, permettant une évaluation objective de la performance de notre système. Nous discutons ces résultats dans le contexte des objectifs initiaux de notre recherche, en soulignant les réussites et en identifiant les domaines d'amélioration potentiels. En conclusion, nous synthétisons les différentes phases de notre travail dans un résumé global. Nous mettons en lumière les bénéfices tangibles de notre système d'une maison intelligente, tout en soulignant les défis rencontrés et surmontés. Enfin, nous traçons les perspectives futures de notre travail, en envisageant des améliorations potentielles et de nouvelles directions de recherche. Cette conclusion sert à la fois de récapitulation de notre travail et de point de départ pour des recherches futures dans ce domaine passionnant.

CHAPITRE I

MÉTHODOLOGIE ET MATÉRIELS

Chapitre I : Méthodologie et matériels

I.1. Introduction

Le système de contrôle domestique représente une étape importante vers l'automatisation de la vie quotidienne de manière abordable et pratique. Ce système se compose d'un groupe d'appareils intelligents intégrés qui fonctionnent en harmonie pour gérer diverses fonctions de la maison.

Dans notre projet, nous avons des entrées qui se composent de capteurs, tels qu'un capteur de gaz MQ2 pour détecter les fuites de gaz, un capteur DHT11 qui mesure la température et l'humidité, et un capteur de distance Ultrason qui mesure la distance et détecte les objets environnants. Tous ces capteurs sont contrôlés par une unité de contrôle ESP32 qui traite les informations et les transmet via les sorties, qui comprennent une sonnette d'alarme, des lumières, un servomoteur c'est un moteur électromécanique précis et un relais utilisé pour contrôler des appareils tels qu'un ventilateur. Toutes ces appareils sont surveillés et contrôlés par l'application Blynk.

I.2. Matériels utilisés :

Pour la réalisation du projet nous avons utilisé le matériel électronique suivant :

- Un capteur de gaz (MQ2).
- Un capteur d'humidité et de température DHT11.
- Un capteur de Ultrason.
- Une carte électronique ESP32.
- 2 Servomoteurs.
- 2 Buzzer.
- 2 Relais.
- 2 Ventilateur (5v).
- 4 Leds.

Comme indiqué dans la figure 1.

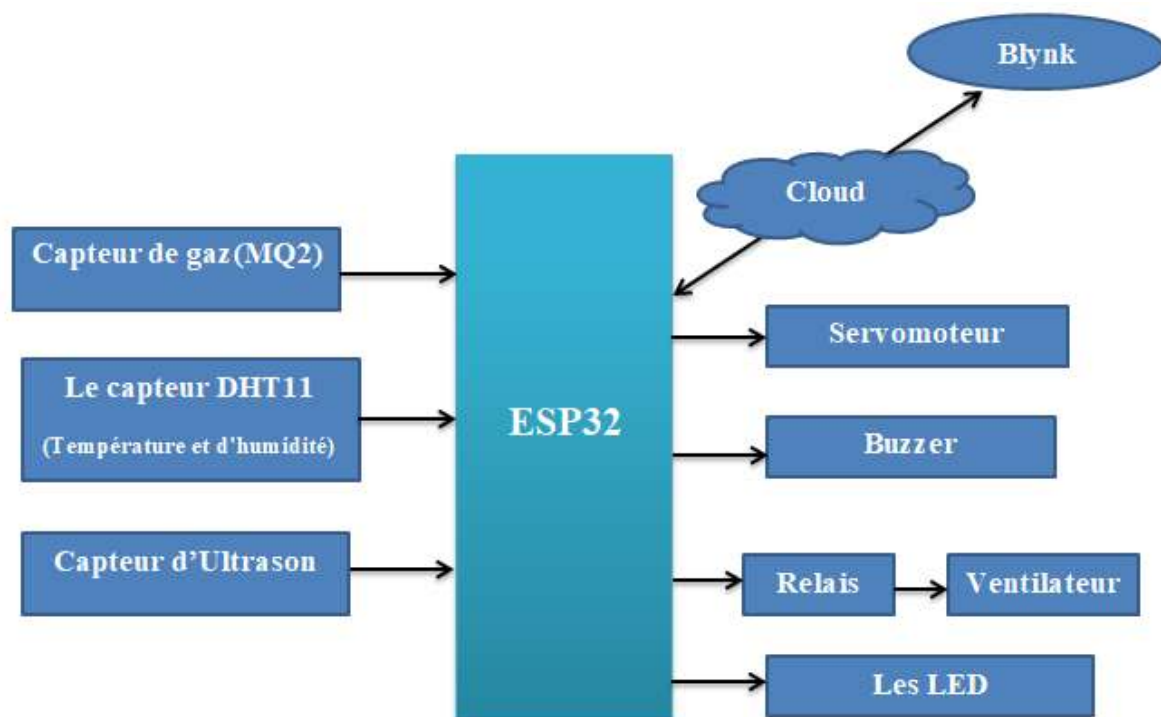


Figure I. 1 : Schéma principal du projet.

I.2.1 Capteur de gaz (MQ2)

Le capteur à semi-conducteur MQ-2 détecte la présence de gaz/fumée à des concentrations allant de 300 ppm à 10000 ppm. La concentration de gaz/fumée dans l'air est détectée par MQ-2 et le résultat est affiché sous forme de tension analogique. On peut utiliser une concentration de détection allant de 300 ppm à 10000 ppm pour détecter les fuites. La température du capteur peut varier de -10 à 50 °C et sa consommation est inférieure à 150 mA et 5 V. [5]



Figure I. 2 : Capteur de gaz MQ2.

I.2.1.1. Les Caractéristiques du MQ2 sont :

- Alimentation : 5V.

- Type d'Interface : Analogique et numérique.
- Connectique : 1- Sortie 2- GND 3- VCC.
- Large panel de détection.
- Réponse rapide et haute sensibilité.
- Système stable à longue durée de vie.
- Dimensions : 40x20mm.

I.2.1.2. Les broches du capteur MQ2

Voici la traduction des composants du capteur MQ-2 en français :

Tableau I.1 : Les broches du capteur MQ2.

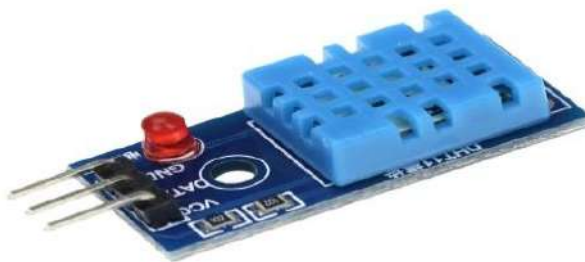
Mesh	Grille en acier, empêche l'entrée de particules non gazeuses dans le ca
Rs	Résistance de sensibilité, permet d'ajuster la sensibilité du capteur
VCC	Connecteur d'alimentation
GND	Masse
D0	Sortie numérique
A0	Sortie analogique



Figure I. 3 : Les broches du capteur gaz MQ2.

I.2.2. Capteur de température et d'humidité DHT11

Le DHT11 est un capteur numérique d'humidité et de température qui a la capacité de mesurer la température ambiante et l'humidité relative de l'air environnant. Il est fréquemment employé dans des domaines où il est essentiel de surveiller les conditions environnementales, comme les systèmes de contrôle de l'environnement dans les bâtiments, les serres, les incubateurs, les chambres de culture ou les projets de bricolage. Les projets de domotique utilisent également fréquemment le capteur DHT11 afin de surveiller la température et l'humidité de l'air et d'ajuster en conséquence les systèmes de climatisation ou de



chauffage.[6]

Figure I. 4 : Capteur DHT11.

I.2.2.1. Caractéristiques :

- Alimentation : 3.3 à 5 Vcc.
- Température : de 0 à 50°C, précision : +/- 2°C.
- Humidité : de 20 à 96% RH, précision +/- 5% RH.
- Consommation maxi : 2,5 mA.
- Dimensions : 16 x 12 x 7 mm.

I.2.2.2. Les broches du capteur DHT11

Le capteur DHT11 dispose de quatre broches, qui sont :

Tableau I.2 : Les broches du capteur DHT11.

VCC	Connecteur d'alimentation
GND	Masse
Data	Les données sont transmises en série sous forme de signaux numériques

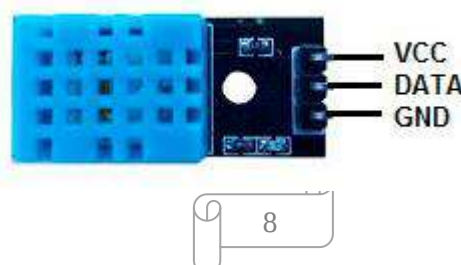


Figure I. 5 : Les broches du capteur DHT11.

I.2.2.3. Le protocole de communication

Le DHT11 se distingue par l'utilisation d'une seule broche pour communiquer avec son enseignant. Il est donc nécessaire de communiquer en utilisant un protocole propriétaire qui requiert des timings spécifiques. [6]

Explication des fondements de la communication avec un DHT11.

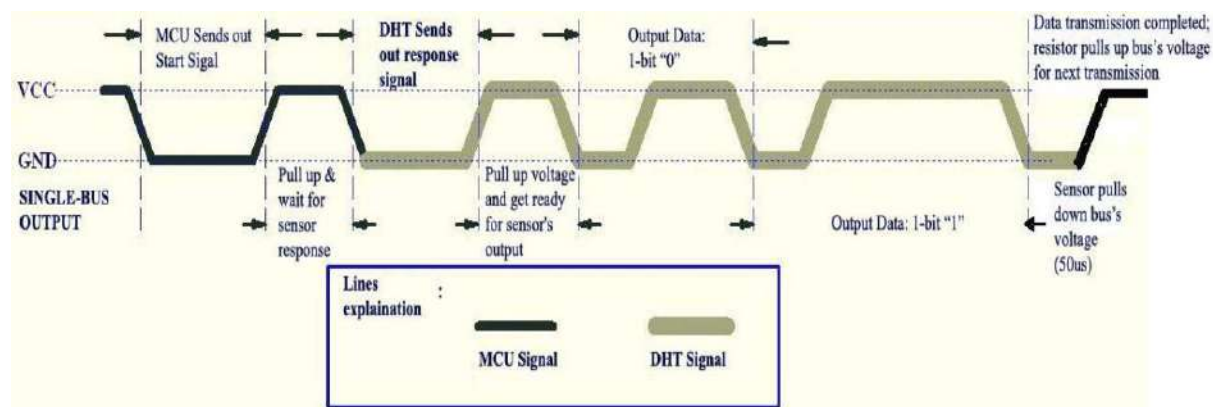


Figure I.6 : Les bases pour communiquer avec un DHT11.

I.2.2.4. Les trois grandes parties d'une trame « type » sont :

- Le maître réveille le capteur en utilisant un signal de « Start ».
- Le capteur se réveille et envoie une réponse pour signaler qu'il est en vie.
- Le capteur se réveille et transmet une réponse pour affirmer qu'il est en vie.
- Le capteur transmet 40 bits (5 octets) de données comprenant la température, le taux d'humidité et un chiffre de vérification.
- Il convient de souligner que la sortie du capteur est appelée "collecteur ouvert", ce qui implique que le capteur ne sort pas des niveaux logiques 0v/5v, mais utilise un transistor en sortie. Dans le cas où le capteur souhaite envoyer un « 0 », il limite le signal à 0v en entraînant le passage de son transistor (entre DATA et GND).
- En cas d'envoi d'un « 1 », le capteur laisse le signal tel quel en bloquant son transistor.

C'est pourquoi la résistance de tirage est cruciale, comme on peut le voir sur le schéma. Sans résistance, il n'y a pas de signal.

I.2.3. Capteur de Ultrason

Le HC-SR04 est un capteur ultrasonique qui permet de mesurer la distance d'un objet, quel que soit l'intensité de la lumière, la température ou le type de matière. Il évalue l'écart entre

les objets situés. Son champ de vision varie de 2cm à 400cm selon l'environnement, avec une précision de 3mm. [7]



Figure I. 7 : Capteur d'Ultrason.

I.2.3.1.Principe de fonctionnement du capteur de ultrason

Le déclenchement d'une mesure se fait par l'envoi d'une impulsion à l'état haut (5v) pendant 10 μ s sur l'entrée Trig. Le capteur émet une série de 8 impulsions ultrasoniques à 40 kHz, Puis il attend le signal réfléchi, lorsque celui-ci est détecté, une impulsion de durée proportionnelle à la distance mesurée est envoyée sur la sortie "Echo". Le calcul de la distance peut se faire grâce au temps mesuré entre le signal émis et le signal reçu :

$$D = (T * V_s) / 2$$

D = distance entre l'objet et le capteur.

T= temps mesuré entre le signal émis et le signal reçu.

Vs= vitesse du son dans l'air : 340 m/s.

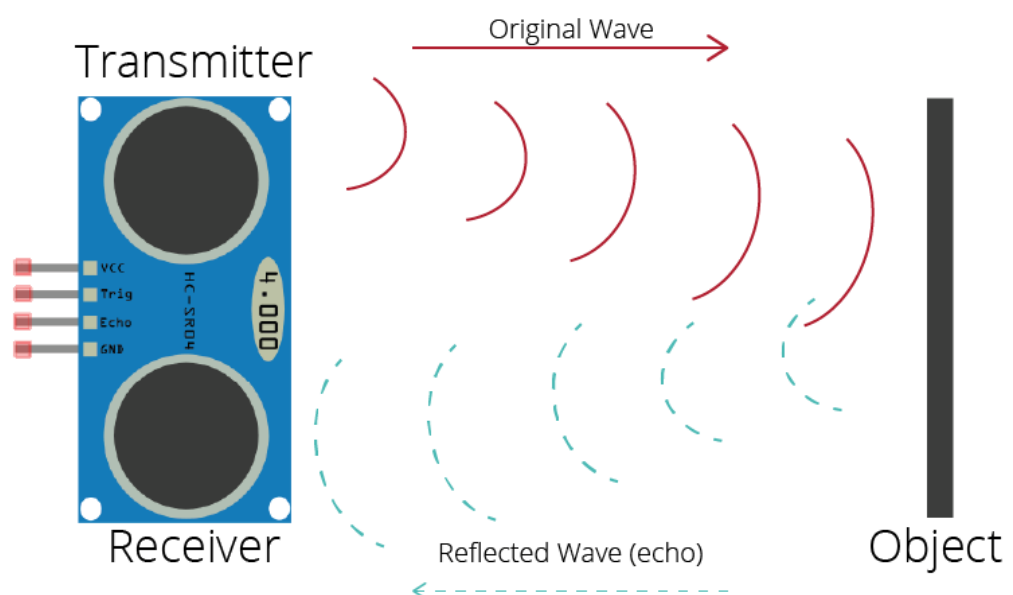


Figure I. 8 : Début des travaux de capteur HC-SR04.

I.2.3.2. Les Caractéristiques de l'Ultrason sont :

Voici une liste de certaines des caractéristiques et spécifications du capteur à ultrasons HC-SR04. Pour plus d'informations, vous devriez consulter la fiche technique du capteur :

- Alimentation : 5V DC.
- Courant d'alimentation : 15mA.
- Angle de détection : 15 degrés.
- Fréquence de travail : 40khz.
- Distance maximale de détection : 4m.
- Distance minimale de détection : 2cm.
- Signal d'entrée de l'émetteur : Impulsion à l'état haut de 10µs.
- Signal de sortie du récepteur Signal numérique à l'état haut et la distance proportionnellement.
- Dimensions : 45 mm x 20 mm x 15 mm.

I.2.3.3. Les broches du capteur Ultrason

Voici le brochage du capteur à ultrasons HC-SR04 :

Tableau I.3 : Les broches du capteur Ultrason.

VCC	Alimente le capteur (5V)
Trigonométrie	Broche d'entrée de déclenchement
Écho	Broche de sortie d'écho
GND	Masse

I.2.4. Carte électronique ESP32

ESP-WROOM-32 est un module MCU polyvalent Wi-Fi + BT + BLE offrant des fonctionnalités puissantes et une variété d'applications. Il est adapté aux réseaux de capteurs à faible puissance et aux tâches complexes, comme l'encodage vocal, le streaming audio et le décodage mp3.

La puce ESP32-D0WDQ6 est le cœur de ce module, offrant une flexibilité et une adaptabilité. On peut gérer les deux cœurs CPU de manière indépendante ou les allumer.

L'horloge peut être réglée entre 80 MHz et 240 MHz de fréquence. L'utilisateur a la possibilité de couper l'alimentation du processeur et d'utiliser un coprocesseur de faible puissance afin de surveiller en permanence les variations d'état des périphériques ou si certaines quantités analogiques dépassent le seuil. ESP32 intègre aussi de nombreux périphériques, tels que des capteurs tactiles capacitifs, des capteurs à effet Hall, des amplificateurs de capteurs à faible bruit, une interface de carte SD, une interface Ethernet, un SDIO/SPI à haute vitesse, UART, I2S et I2C.

Le module est polyvalent grâce à l'intégration du Bluetooth, du Bluetooth LE et du Wi-Fi, ce qui permet de cibler une large gamme d'applications. Le Wi-Fi offre une portée physique élevée et une connexion directe à Internet via un routeur Wi-Fi, tandis que le Bluetooth permet à l'utilisateur de se connecter aisément au téléphone ou de diffuser des balises à faible énergie pour une détection simplifiée. La puce ESP32 présente un courant de veille inférieur à 5 μ A, ce qui en fait une solution idéale pour les applications électroniques alimentées par batterie et portables.

Le module supporte un débit de données de 150 Mbps et une puissance de sortie de 20 dBm à l'antenne afin de garantir une portée physique optimale. Le module présente donc des caractéristiques de pointe dans le domaine et les performances les plus élevées en ce qui concerne l'intégration électronique, la portée, la consommation d'énergie et la connectivité.

L'ESP32 est équipé d'un système d'exploitation freeRTOS avec LwIP, avec TLS 1.2 intégré avec accélération matérielle. On offre également la possibilité de mettre à jour sécurisée (chiffrée) par voie aérienne (OTA), permettant ainsi aux utilisateurs de mettre à jour leurs produits même après leur sortie, à un coût et à un effort minimaux. [8]



Figure I. 9 : ESP32-WROOM-32.

I.2.4.1. Les Caractéristiques d'ESP32-WROOM-32

L'ESP32 WROOM 32 présente les spécifications suivantes :

- Microcontrôleur : Processeur dual-core Tensilica Xtensa LX6 (fréquence d'horloge de 240 MHz).
- Connectivité sans fil : Connectivité Wi-Fi intégrée 802.11 b/g/n. Connectivité Bluetooth intégrée (Bluetooth Low Energy, BLE).
- Mémoire : 520 Ko de SRAM. 448 Ko de ROM (mémoire de programme).
- Support en option pour une mémoire flash externe SPI.
- Périphériques : Périphériques Pins GPIO (Entrée/Sortie à usage général).
- UART, SPI, I2C, I2S, ADC, DAC.
- PWM (Modulation de Largeur d'Impulsion).
- Capteur à effet Hall.
- Sécurité : Fonctionnalités de sécurité basées sur le matériel, y compris le démarrage sécurisé et le chiffrement de la mémoire flash.
- Gestion de l'alimentation : Modes de faible consommation multiples pour une efficacité énergétique. Régulateur programmable pour l'alimentation du cœur.
- Tension de fonctionnement : 2.2V to 3.6V.
- Température de fonctionnement : "-40°C to +85°C".
- Facteur de forme : Module compact avec des bords à créneaux pour une soudure facile.
- Horloges et minuteries : Horloge temps réel (RTC) intégrée. Minuteries et watchdogs pour les applications sensibles au temps.
- Antenne intégrée : Antenne intégrée pour la communication Wi-Fi et Bluetooth.
- Certifications : Certifications FCC, CE, IC, et autres pour la conformité.
- Support de développement : Support de l'IDE Arduino pour la programmation. Support MicroPython. Kit de développement logiciel (SDK) fourni par Espressif Systems.
- Options d'emballage : Disponible dans différentes options d'emballage, y compris les populaires ESP32 WROOM 32D et ESP32 WROOM 32U.
- Dimensions : Les dimensions typiques du module sont d'environ 18 mm x 25,5 mm.[9]

I.2.4.2.ESP32 WROOM 32 GPIO(Pins)

Le module ESP32 WROOM-32 dispose d'un ensemble de broches GPIO (Entrée/Sortie à Usage Général) qui peuvent être utilisées à diverses fins, notamment en tant qu'entrées/sorties numériques, entrées analogiques, modulation de largeur d'impulsion (PWM), et interfaces de communication telles que UART, I2C et SPI. La correspondance exacte des broches peut dépendre du module ESP32 spécifique ou de la carte de développement utilisée, il est donc essentiel de se référer à la documentation spécifique à votre appareil. [9]

Grâce à ces avantages, l'ESP32 est un choix privilégié pour les développeurs et les amateurs IoT, fournissant une plate-forme robuste et fiable pour construire des projets diversifiés et innovants.

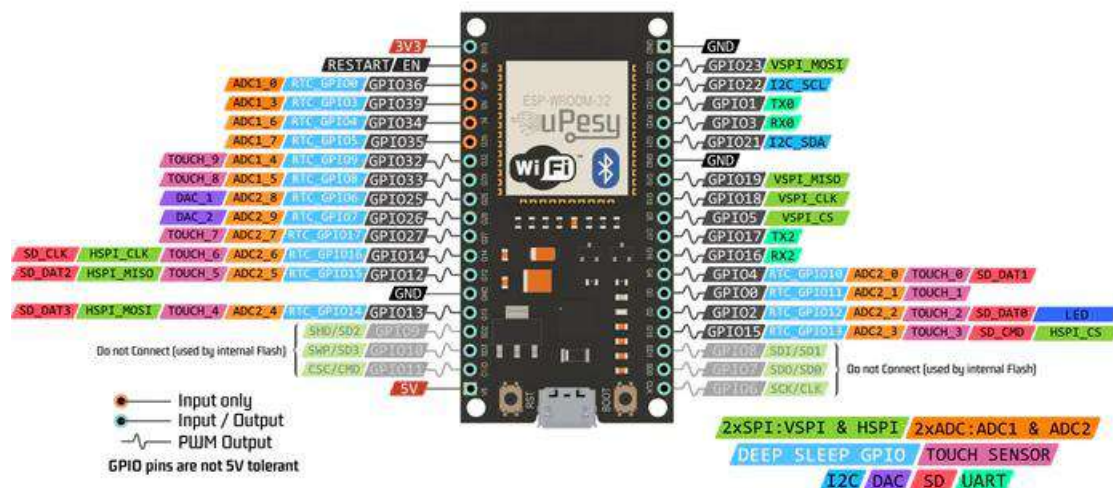


Figure I. 10: ESP32 WROOM 32 GPIO Pins.

I.2.5. Servomoteur

Le moteur servo SG90 est un type de moteur servo petit et économique couramment utilisé dans les projets de robotique et d'automatisation. Ce moteur offre un contrôle précis de la position angulaire de l'appareil. Le SG90 pèse environ 9 grammes, a un couple allant jusqu'à 1,8 kg/cm, et une rotation approximative jusqu'à 180 degrés.

Les applications principales des moteurs servo SG90 comprennent les projets de loisirs à petite échelle tels que les voitures télécommandées, les robots, les avions. Ces moteurs peuvent être utilisés pour contrôler la position de la roue de direction, des ailes, et d'autres parties nécessitant une position précise. De plus, le moteur servo SG90 est également couramment utilisé dans les projets DIY tels que les bras robotiques, les machines, et d'autres

dispositifs mécaniques nécessitant un contrôle précis de leur mouvement. [10]



Figure I. 11 : Micro servo SG90.

I.2.5.1. Principe de fonctionnement du servo moteur

Un servo moteur est généralement composé de quatre composants principaux : un moteur DC, une boîte de vitesses, une résistance variable, et un circuit de contrôle. Le moteur DC est un moteur à haute vitesse et à faible couple, mais la boîte de vitesses est utilisée pour réduire la vitesse à environ 60 tours par minute tout en augmentant le couple en même temps. La boîte de vitesses est un composant essentiel du servo car elle permet des mouvements précis par le servo. La résistance variable est utilisée pour fournir un retour au circuit de contrôle, lui permettant de régler précisément la position du servo. Ce retour est utilisé par le circuit de contrôle pour contrôler le mouvement du servo moteur. [10]

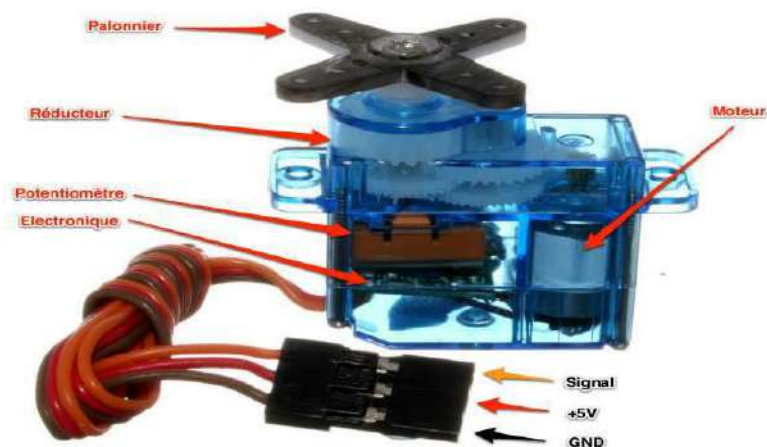


Figure I. 12 : Composants du servo moteur et les broches.

I.2.5.2. Les Caractéristiques du servomoteur SG90 :

- Couple de décrochage : 1.2kg·cm 4.8V, 1.6kg·cm 6V.

- Tension de fonctionnement : 3.5 – 6V.
- Courant sans charge : 100mA.
- Courant de décrochage : 650mA.
- Vitesse maximale : 60 degrés en 0,12 s.
- Poids : 9g.

I.2.6. Buzzer

Les buzzers sont également utilisés dans les systèmes d'alarme, les ordinateurs, les minuteries et pour confirmer l'entrée des utilisateurs. Ils sont des buzzers continus avec une sortie sonore minimale de 70 dB à 1 mètre pour leur taille. Sa puissance électrique nominale est de 3 V et 150 mA. [11]



Figure I. 13 : Buzzer.

I.2.6.1. Les Caractéristiques du buzzer :

- Dimensions : diamètre approximatif de 12 mm, hauteur 9mm.
- Type de sonnerie : continue.
- Tension d'alimentation : 2V - 4V.
- Couleur du corps : noir.
- 2 broches : positive et négative.

I.2.7. Relais module

Un relais 5V est un type de relais qui requiert une entrée DC de 5V afin de fonctionner. Le relais lui-même et le module de commande sont composés de deux parties distinctes. La bobine qui génère le champ magnétique, l'armature qui se déplace pour finir ou déconnecter un circuit, et les contacts qui s'ouvrent et se ferment pour faire fonctionner l'interrupteur de charge, se trouvent dans le relais. L'interface ou la partie du module de relais avec laquelle l'utilisateur interagit est le module de commande du relais.

Le module de commande comprend également des LED pour l'alimentation et l'état, ainsi que d'autres éléments tels que la diode de protection, le transistor, la résistance et d'autres

dispositifs à semi-conducteurs indispensables à son fonctionnement. [12]

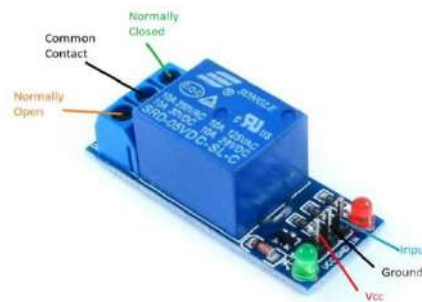


Figure I.14 : Relais module 5v.

I.2.7.1. Les Caractéristiques du relais :

- Tension normale : 5 V CC.
- Courant normal : 70mA.
- Courant de charge maximum : 10A/250V AC, 10A/30V DC.
- Tension de commutation maximale : 250 V CA, 30 V CC.
- Temps de fonctionnement : 10 ms.
- Temps de relâchement : ≤ 5 ms.

I.2.7.2. Principe de fonctionnement du relais

Le module de relais 5V requiert un signal de 5V provenant d'un microcontrôleur ou d'un capteur afin de mettre en marche l'interrupteur. Il est également extrêmement facile à utiliser. Le relais s'allume lorsque la broche d'entrée est HIGH, tandis que lorsque l'entrée est LOW, il s'éteint. Ci-dessous, vous découvrirez le fonctionnement du module de relais 5V. [12]

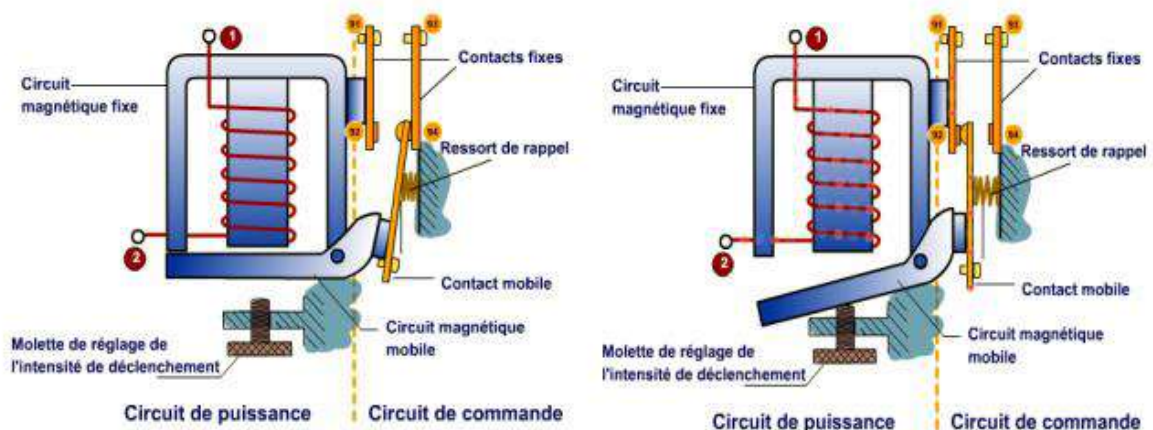


Figure I.15 : Principe de fonctionnement du relais.

I.2.8. Ventilateur

Un ventilateur ou un extracteur est utilisé pour extraire l'air chaud et humide d'une serre, dans le but de favoriser la circulation de l'air et de réguler certains paramètres essentiels au bon fonctionnement de la serre, tels que la température, l'humidité ou les concentrations de gaz, dans le voisinage des consignes choisies. [13]



Figure I.16 : ventilateur 5v.

I.2.9. Plaque d'essai

Une plaque d'essai (breadboard ou protoboard) est un tableau qui est constitué d'orifices électriquement reliés entre eux de manière interne. On peut placer les composants électroniques et les fils sur la plaque afin de réaliser le montage des circuits électroniques. [14]

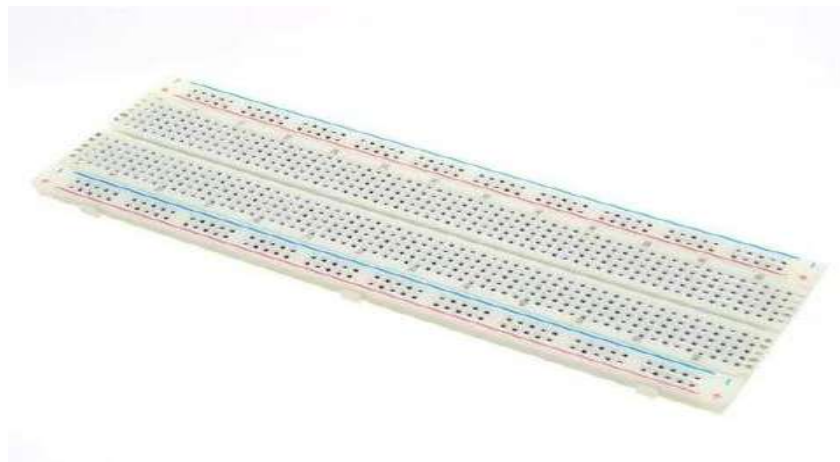


Figure I.17 : Plaque d'essai (Breadboard).

I.2.10. Les leds

Les LED sont des diodes à jonction PN qui sont fabriquées à partir d'un matériau semi-

conducteur fortement dopé avec une recombinaison radiative. La LED produit un flux de lumière incohérent issu de l'émission spontanée de photons. [15]

La LED possède 2 broches :

- Le long : positive.
- Le petit : négative.



Figure I.18 : Les Leds.

I.2.11. Les Résistances

En tant que composant électrique (un dipôle), il résiste au passage du courant électrique, comme son nom l'indique. Son objectif est donc de restreindre l'intensité du courant électrique dans un circuit. [16]

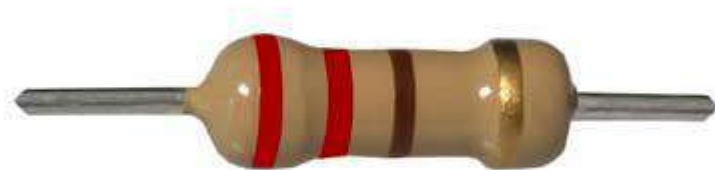


Figure I.19 : Résistance.

I.2.12. Fils de connexions

Les fils de connexion sont les câbles qui connectent divers appareils intelligents tels que les systèmes d'éclairage, de chauffage et de refroidissement, les caméras de sécurité et les

capteurs entre eux et au système de contrôle central, permettant la transmission de l'électricité et des signaux pour contrôler ces appareils et coordonner leur travail automatiquement. et efficacement.



Figure I.20 : Fils de connexions.

I.3.Partie logiciel

Le logiciel pour développer des modules arduino est un logiciel Java libre et multiplateforme qui offre la possibilité de créer du code et de le compiler. Il peut également transférer le programme via une liaison série (RS232, Bluetooth ou USB en fonction du module). L'interface arduino peut aussi être supprimée et les programmes compilés en ligne de commande. Le langage utilisé pour la programmation est le C++. [17]

I.3.1. Définition de l'IDE

L'environnement de programmation arduino (IDE) est une application écrite en Java et inspirée du langage processing. L'IDE vous permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles par la carte.

I.3.1.1. Description de l'IDE

L'IDE est un logiciel de programmation axé sur le code qui contient une cinquantaine de

commandes différentes. A l'ouverture, l'interface visuelle du logiciel contient le menu un menu, des boutons de commande en haut, une page blanche, une bande noire en bas .

Comme ceci (voir la figure 3.6) :



Figure I. 21 : Interface IDE Arduino.

I.3.1.2. Barre de menu

Les différents éléments du menu vous permettent de créer de nouvelles esquisses (programmes), de les sauvegarder et de gérer les préférences du logiciel et les paramètres de communication avec votre carte arduino. Le menu comprend : Fichier : pour créer, sauvegarder en spécifiant la destination, et appeler un programme ; Edition : pour couper, copier, coller, supprimer, sélectionner,...etc. ; Esquisse : pour regrouper les fichiers réalisés ; Outils : pour spécifier le type de carte, le port série, le format, recharger et réparer le codage, enregistrer la séquence d'initialisation, de la carte connectée à l'ordinateur.

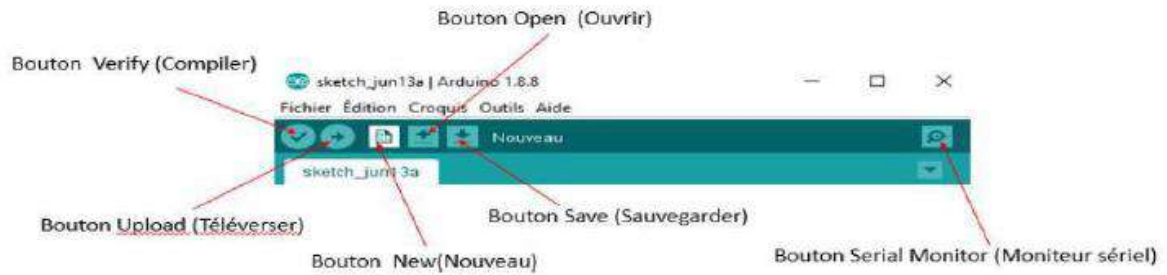


Figure I. 22 : Barre d'action.

I.3.1.3. Fenêtre d'édition du programme

C'est l'éditeur où le programme est écrit, chaque programme obéit à quelques notions pour pouvoir bien structurer le programme afin de le compiler et éviter les erreurs de syntaxe et autres.

I.3.1.4. Zone de message des actions en cours

La barre qui indique si le programme est correcte ou non montre les erreurs commises pendant le programme, comme l'oubli d'un point-virgule, l'absence d'une accolade ou toute autre erreur dans les instructions.



Figure I. 23 : Zone de message des actions.

I.3.1.5. Définition des constantes et des variables

Cette partie est facultative, vous définissez et déclarez chaque entrée et sortie, en lui donnant un nom arbitraire et en lui attribuant le numéro de l'entrée ou de la variable souhaitée, sans oublier de préciser le type de la variable.

I.3.1.6. Configuration des entrées/sorties

Les instructions viennent après la configuration du vide (?), après avoir ouvert un crochet, on peut manipuler les broches de la carte en les configurant comme entrées ou sorties, selon les besoins.

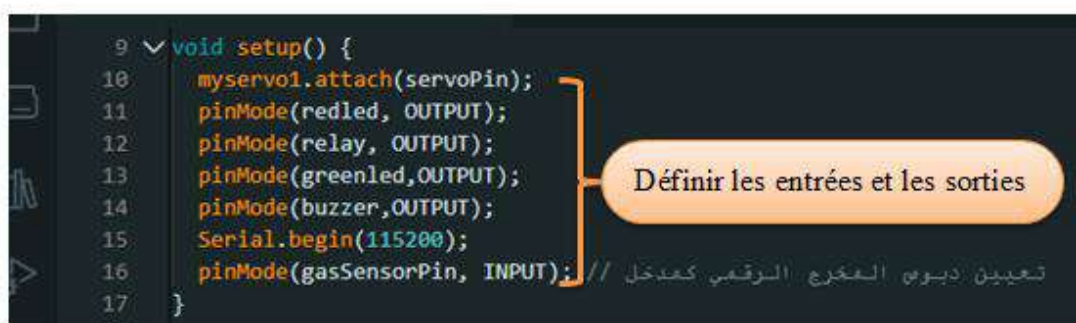


Figure I. 24 : Configuration des entrées/sorties.

I.3.1.7. Programmation des interactions et des comportements

Ceux-ci viennent après la boucle `void` (?), c'est la partie principalement, où nous écrivons les instructions et les opérations telles que la lecture des données, les boucles, les assignations,...etc. Chacun d'entre eux doit se terminer par un point-virgule.

I.3.1.8. Commentaires

Dans la configuration des E/S, les commentaires doivent être écrits après une barre oblique ou une étoile, ou les deux, alors que dans une ligne de code, ils sont écrits après deux barres obliques.

I.3.1.9. Utilisation d'un ESP32 avec l'Arduino IDE

Dans cette étude, nous examinerons la manière de programmer un ESP32 à partir de l'EDI Arduino en utilisant des langages C ou C++. Actuellement, la société « Espressive Systems » (fabricant de l'ESP32) n'a pas intégré cette carte de manière complète dans le gestionnaire de carte Arduino, ce qui oblige à un processus manuel.

I.3.1.10. Configuration de l'IDE Arduino pour l'ESP32

Rendons-nous dans le menu « Préférences » de votre interface IDE.

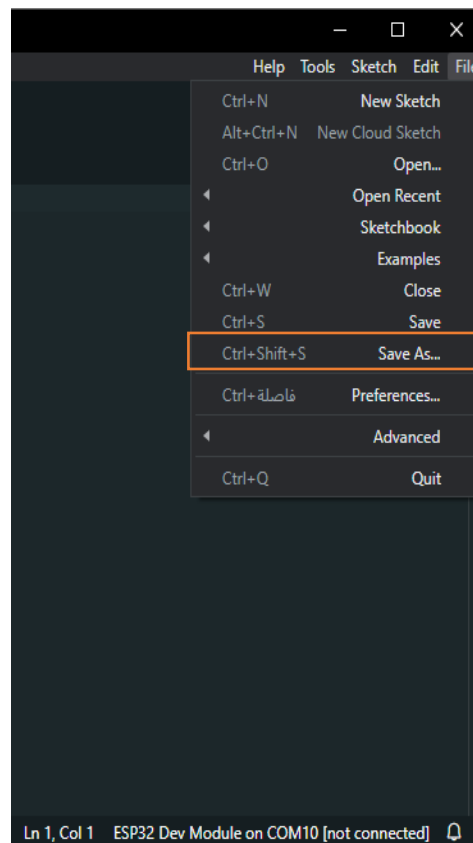


Figure I. 25 : Configuration de l'IDE Arduino pour ESP 32 étape 1.

I.3.1.11. Ajouter la carte ESP32 dans la base des cartes Arduino de l'IDE

Dans l'IDE Arduino :

- Ouvrir l'onglet Fichier > Préférences
- Entrer l'adresse suivante dans le champ

URL de gestionnaire de cartes supplémentaires :

https://dl.espressif.com/dl/package_esp32_index.json

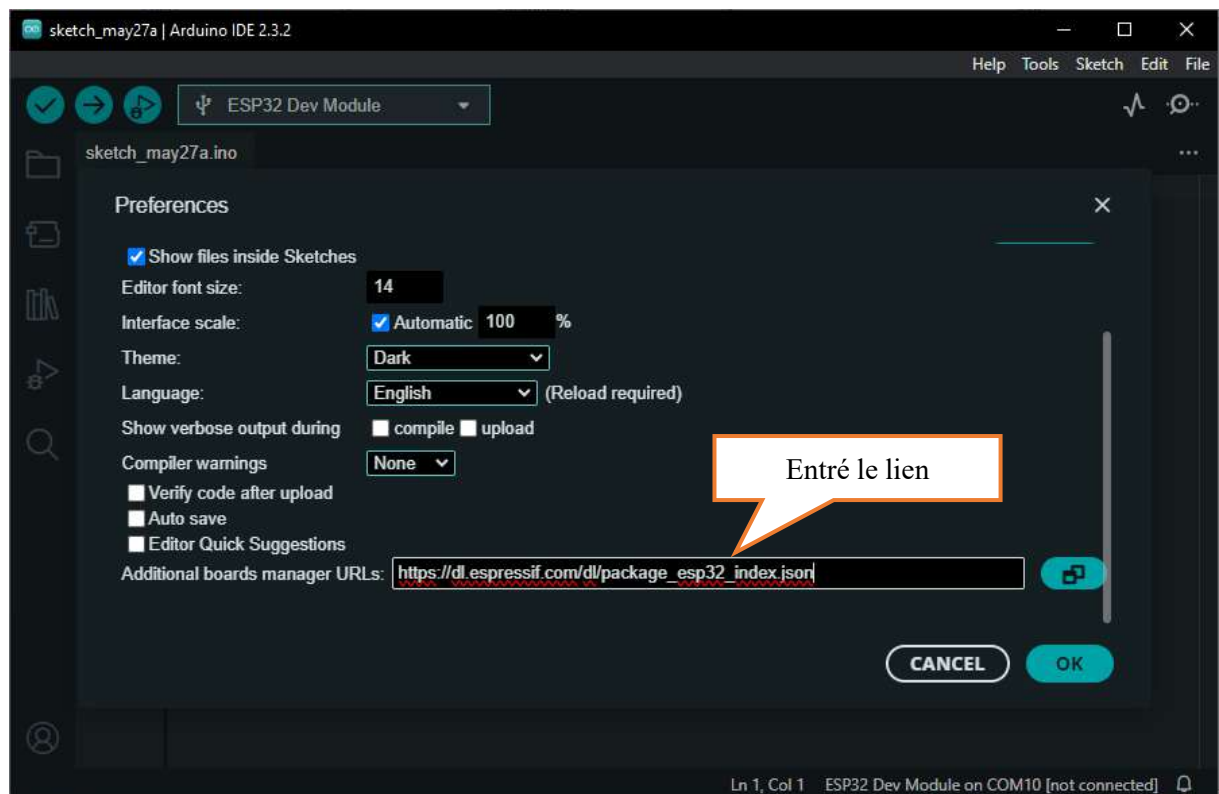


Figure I. 26 : Configuration de l'IDE Arduino pour ESP 32 étape 2.

Nous enregistrons par (OK) et quitter le menu « Préférences » Rendons-nous, maintenant, dans le menu « boards manager » par le biais du menu « Tools » et « boards ».

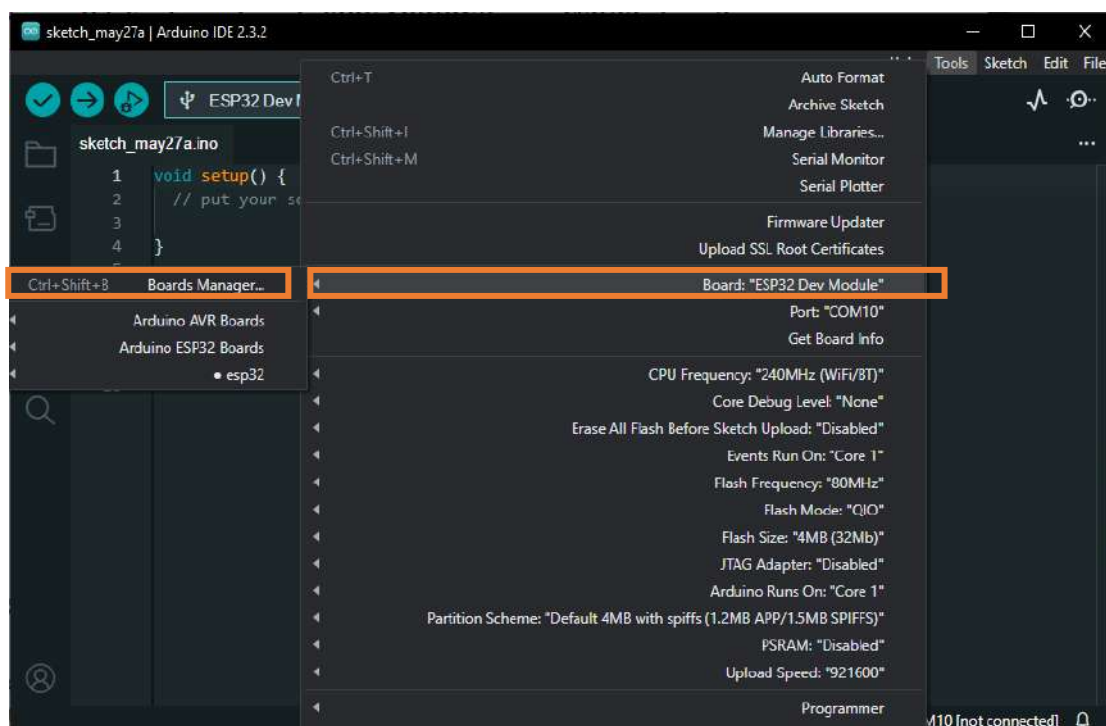


Figure I.27 : Configuration de l'IDE Arduino pour ESP 32 étape 2.

I.3.1.12. Installer la carte ESP32 dans l'IDE Arduino

Toujours dans l'IDE Arduino :

- Ouvrir l'onglet.
- «Tools» > «boards» > «boards manager».
- Rechercher le paquet ESP32.
- Installer le paquet : sélectionnez la dernière version disponible et cliquer sur Installer.

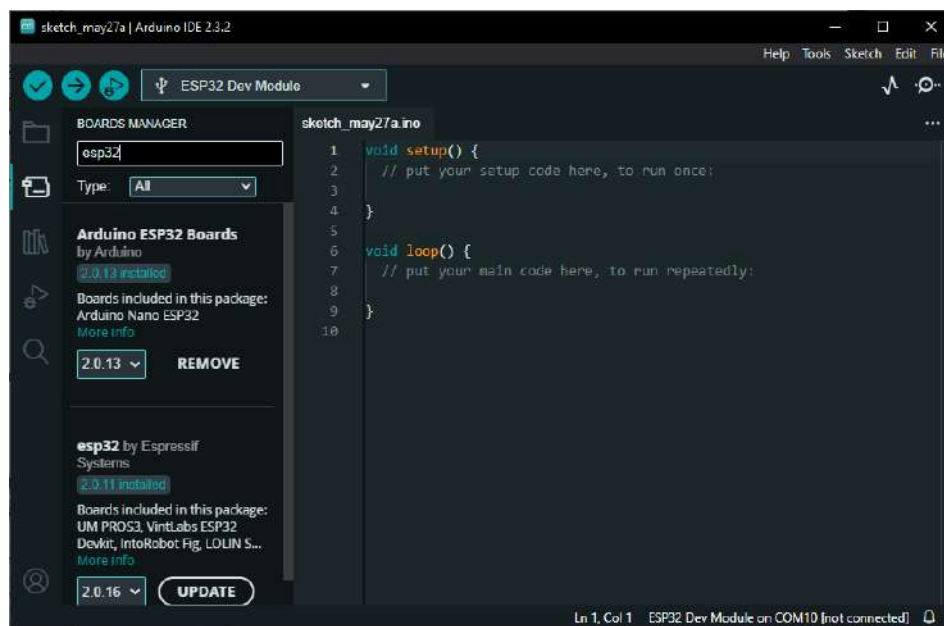


Figure I.28 : Configuration de l'IDE Arduino pour ESP 32 étape 3.

I.3.1.13. Sélectionner la

Rechercher le paquet ESP32

Avant d'envoyer un programme à la carte, il est nécessaire de sélectionner le type de carte (esp32 dev module) et le numéro du port USB (COM 10) comme indiqué dans la figure ci-dessous (voir la figure 3).

Installer le paquet

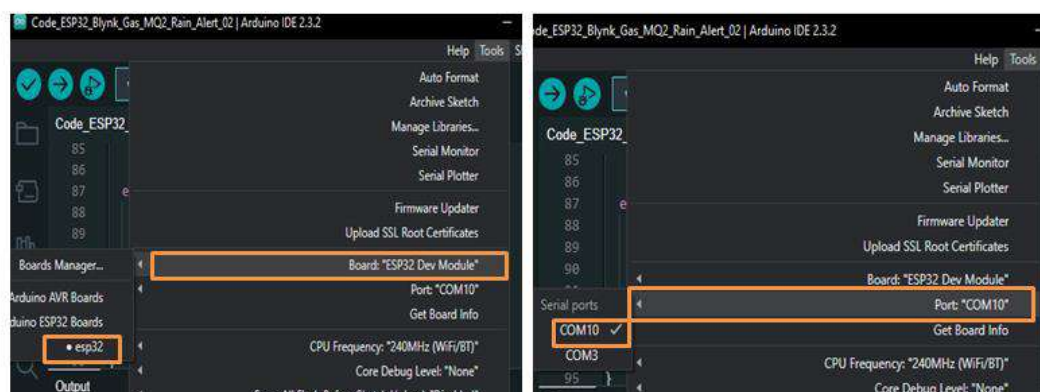


Figure I. 29 : Paramétrage de la carte.

I.3.1.14. Étapes pour télécharger le programme

Il faut suivre une manipulation simple pour injecter un code vers la carte esp32 via le port USB (voir la figure 3.9).



Figure I. 30 : étapes de téléchargement du code.

I.3.2. Plateforme Blynk

Blynk est une plateforme pour l'Internet des Objets (IoT). Elle permet notamment de concevoir une application mobile (Android et iOS) pour contrôler et visualiser les données d'un système embarqué via un serveur cloud public ou privé.

La conception de l'application mobile (Android et Ios) à base de widgets (éléments graphiques) est réalisée par simple glisser & déposer sans écrire une ligne de code. [18]

Blynk a été conçu pour l'Internet des objets. C'est une plate-forme avec des applications Ios et

Android pour contrôler Arduino, Raspberry Pi, ESP32 et autres sur Internet. Il peut également:

- Contrôler le matériel à distance
- Afficher les données des capteurs
- Stocker des données, les visualiser et faire beaucoup d'autres choses intéressantes.

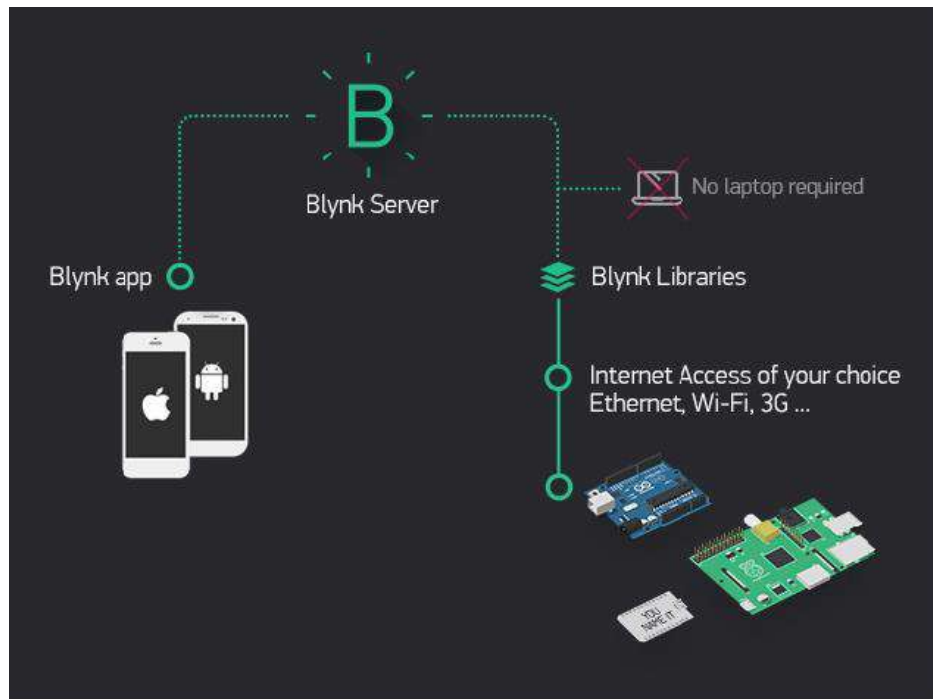


Figure I. 31 : Fonctionnement de Blynk.

La plate-forme comprend trois (03) composants principaux :

A. Blynk App

Nous permet de créer des interfaces incroyables pour notre projet en utilisant les divers widgets que nous fournissons.

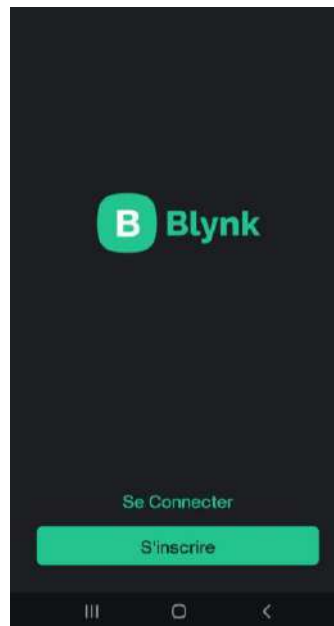


Figure I. 32 : Interface de Blynk.

B. Blynk Server

Responsable de toutes les communications entre le Smartphone et le matériel. Nous pouvons utiliser notre Blynk Cloud ou exécuter notre serveur Blynk privé localement. Il est open-source, pourrait facilement gérer des milliers d'appareils et peut même être lancé sur un Raspberry Pi.

C. Les bibliothèques Blynk

Pour toutes les plates-formes matérielles populaires permettent la communication avec le serveur et traitent toutes les commandes entrantes et sortantes.

D. Fonctionnalités

- API et UI similaire pour tous les hardwares et périphériques supportés.
- Connexion au cloud via : (Ethernet Wifi Bluetooth et BLE USB (Serial) o ...).
- Collection de widgets faciles à utiliser.
- Manipulation des broches directes sans code à écrire.
- Facilité d'intégrer et ajouter de nouvelles fonctionnalités en utilisant les broches virtuelles.
- Surveillance de l'historique des données via le widget **Historie Graph**.
- Communication Périphérique-à-Périphérique en utilisant le widget **Bridge**.
- Envoi d'emails, de tweets, de notifications push, etc.

E. Configuration de l'application Blynk pour ESP32

Après avoir téléchargé l'application Blynk sur un Smartphone, nous avons besoin de suivre les étapes suivantes :

1. Créer un compte où on peut utiliser notre compte Gmail actuel.
2. Après la création du compte, une fenêtre s'ouvrira et on clique sur **Nouveau projet**.
3. On donne un nom en fonction de notre choix et dans l'appareil, on choisit **ESP32** et dans le type de **connexion**, on choisit **wifi**, puis on clique sur Créer.
4. Une fenêtre apparaîtra qui montre notre **jeton d'authentification** qui sera sauvegardé par nos soins dès la réception de messagerie pour des éventuelles utilisations plus tard. Donc ; nous pouvons ouvrir notre e-mail pour vérifier la clé d'authentification et après on ajoute les widgets dont on a besoin. [19]

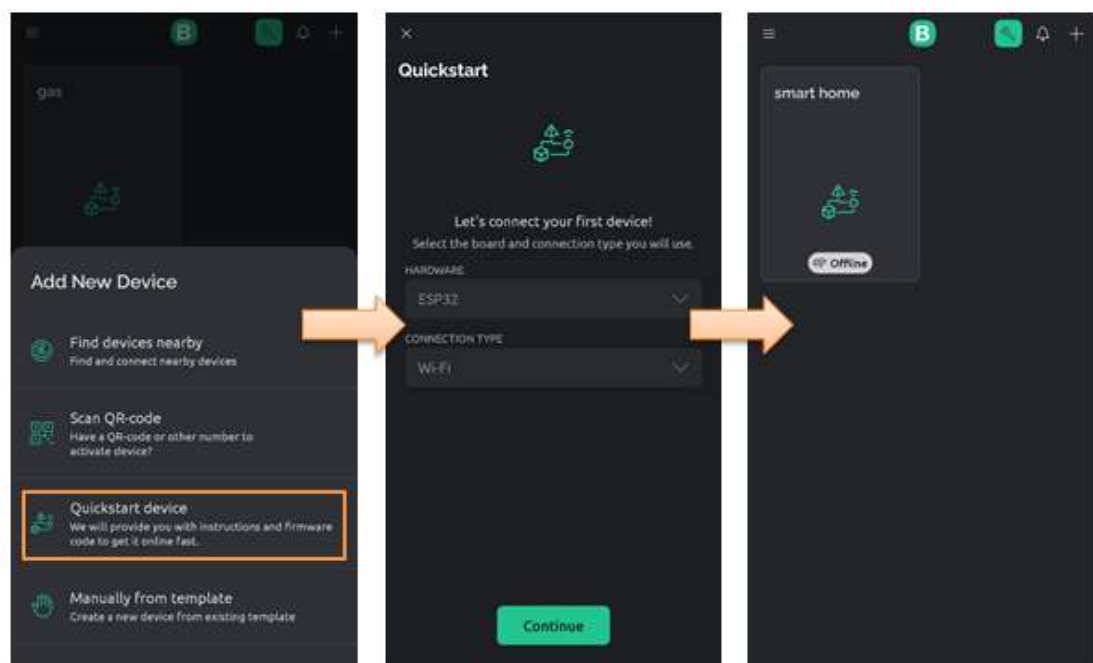


Figure I. 33 : Etapes de Configuration de l'application Blynk pour ESP32.

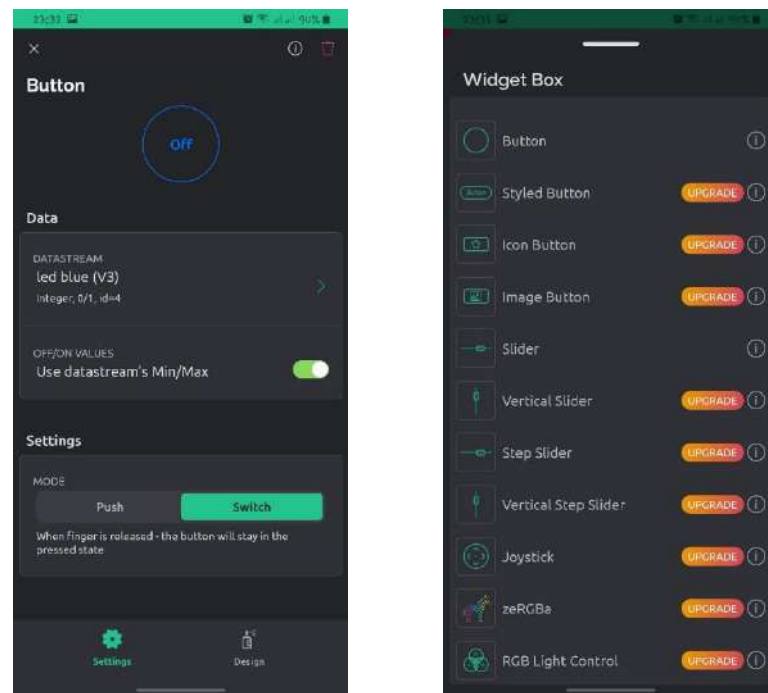
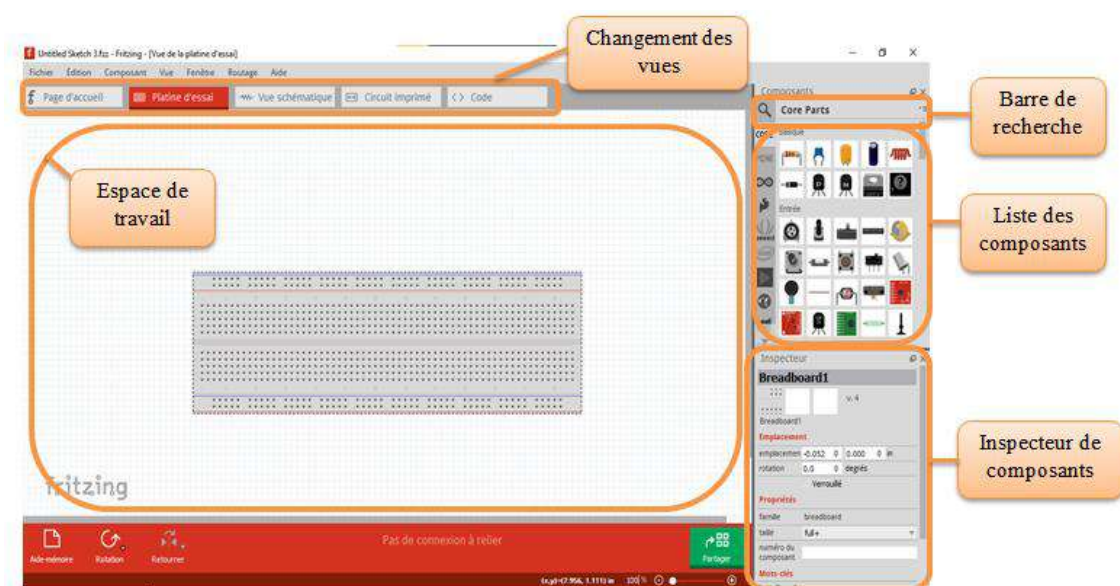


Figure I. 34 : Fonctionnalités de l'application Blynk.

I.3.3. Fritzing

Fritzing est un logiciel open-source de simulation multiplateforme qui offre diverses perspectives : La platine d'essai, les schémas électriques et le circuit imprimé sont utilisés pour créer des schémas des circuits que nous utilisons avec Arduino et les tester en utilisant le code développé dans l'IDE d'Arduino. Il permet également de sauvegarder les schémas de connexions afin que l'utilisateur puisse reprendre le montage fonctionnel avec son programme



ultérieurement. [20]

Figure I. 35 : Interface de Fritzing.

I.4.Conclusion

Les objectifs liés aux outils de développement ont été clairement exposés dans ce chapitre, où nous avons pu acquérir des connaissances fondamentales sur la fonctionnalité de la carte ESP32 et les divers modules requis pour mettre en place notre smart habitat, ainsi que sur les logiciels utilisés pour la programmation du système. La compréhension de ces concepts dans ce chapitre nous sera extrêmement bénéfique pour continuer le prochain chapitre qui se concentrera sur la partie pratique de notre travail.

CHAPITRE II

RÉSULTATS ET DISCUSSIONS

Chapitre II : Résultats et discussions

II.1. Introduction

La conception de cette maison intelligente consiste principalement à gérer à distance les appareils électroniques comme les ventilateurs, les lumières, la climatisation, etc. Utiliser un smartphone. Grâce à l'application Blynk, il est possible de gérer les appareils électroniques de la maison de n'importe où et d'analyser leurs résultats.

Les applications que nous avons réussi à les mettre en Suvre sont :

- La lecture de différentes données telles que la température et l'humidité.
- La surveillance de fuites de gaz et de fumée.
- Surveiller l'ouverture et la fermeture de la porte.
- schéma principal.

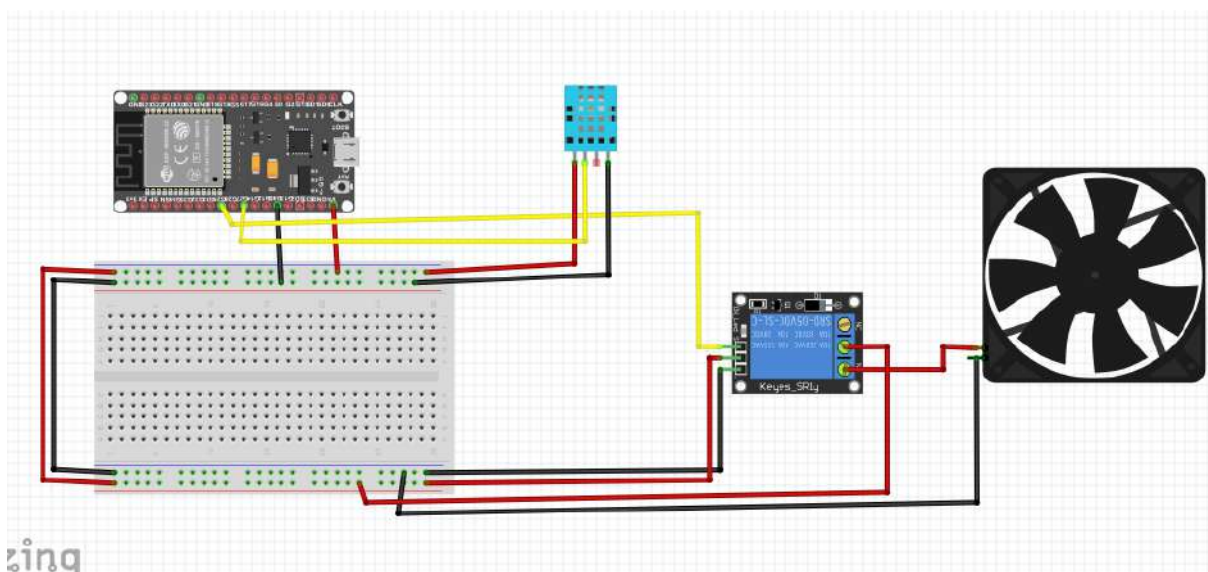
II. 2. Résultats :

II. 2.1.La lecture de différentes données telles que la température et l'humidité

Système de détection d'humidité et de température :

Lorsque le système détecte une augmentation de la température ou de l'humidité,

Le capteur envoie des données au microcontrôleur ESP32 Ils sont envoyés à la « Blynk app » du smartphone via la base de données, permettant de visualiser La température, l'humidité et l'application contrôlent le fonctionnement du système.



Ventilation (ventilateur) si nécessaire.

Figure II. 36 : Circuit réel de l'application du control de temperature et humidité.

En surveillant la température et l'humidité à l'intérieur de la pièce à l'aide de capteurs connectés à l'application Blynk, nous obtenons deux résultats :

- Avant d'allumer le ventilateur.
- Quand le ventilateur tourne.

a. Avant d'allumer le ventilateur

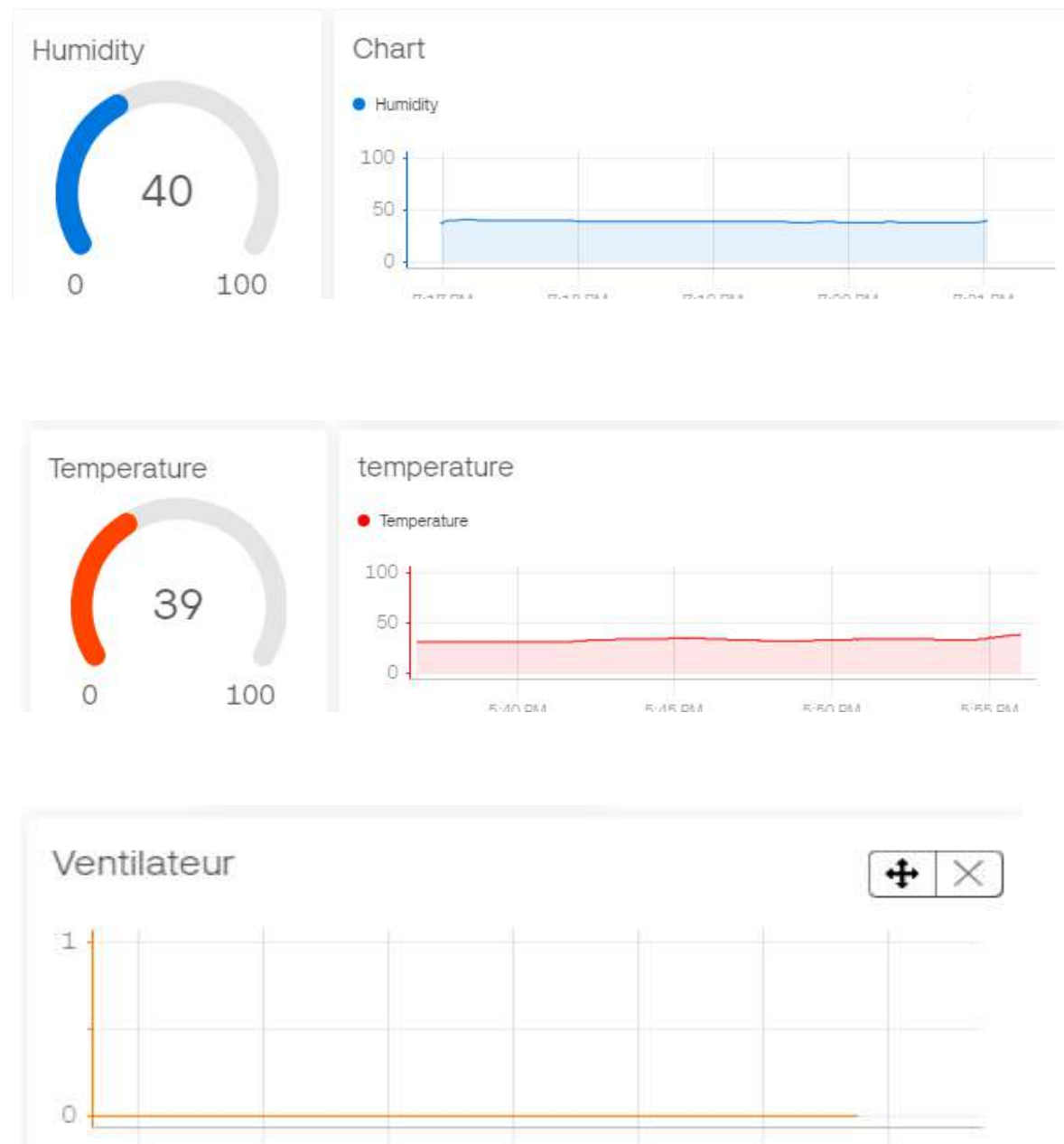


Figure II. 37 : Lecture du capteur de température et d'humidité Avant d'allumer le ventilateur.

On remarque que la température et l'humidité sont élevées et restent à une certaine valeur

avant d'allumer le ventilateur.

b.Quand le ventilateur tourne

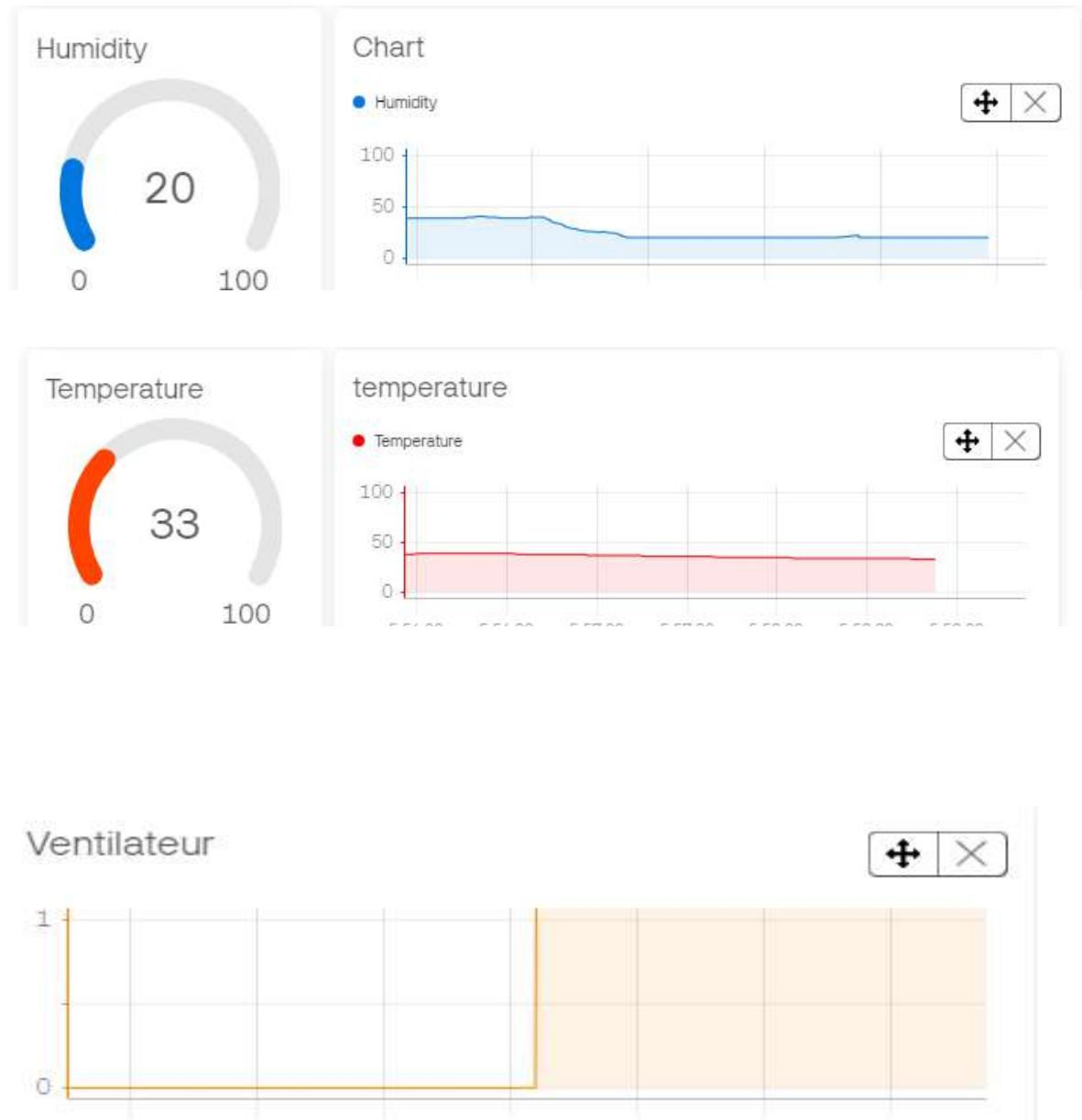


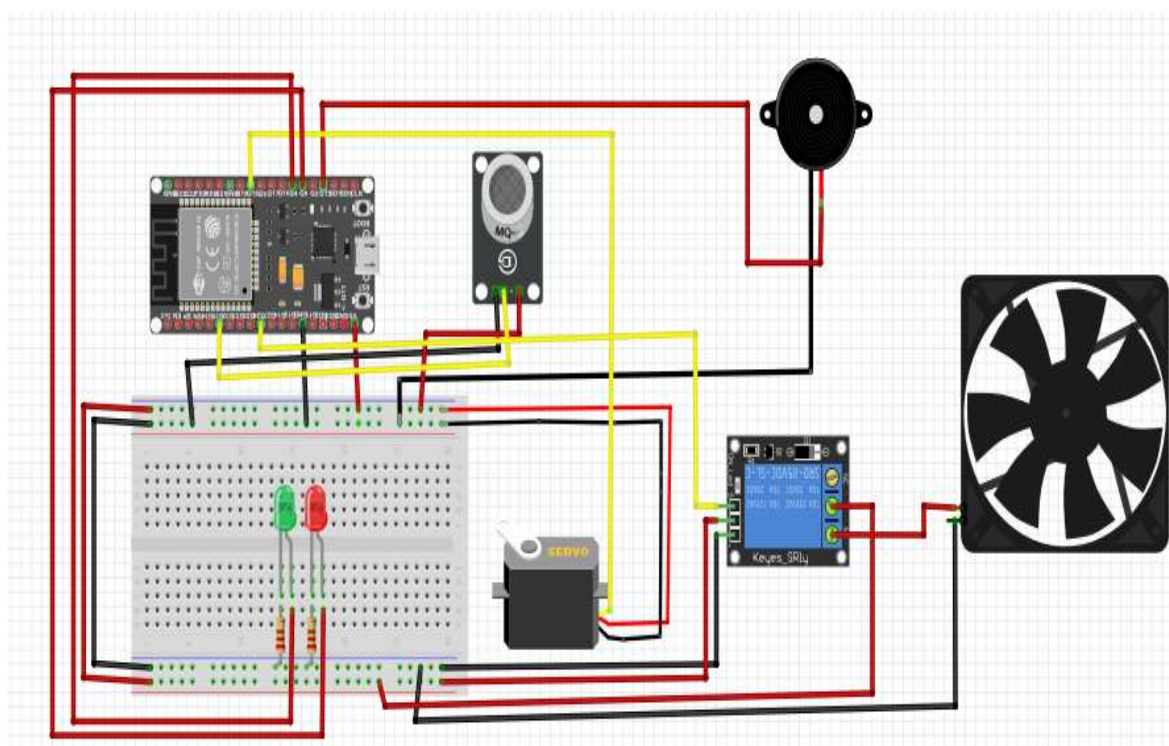
Figure II. 38 : Lecture du capteur de température et d'humidité quand le ventilateur tourne.

On remarque que la température et l'humidité diminuent et restent à une certaine valeur lorsque le ventilateur est allumé.

II. 2.2. La surveillance de fuites de gaz et de fumée

Le capteur de gaz MQ-2 a été testé pour mesurer différentes concentrations de gaz au niveau des ouvertures des fenêtres et des ventilateurs. Le capteur a été connecté à une carte esp32 et les valeurs numériques résultantes ont été lues. Plusieurs expériences ont été menées avec différentes concentrations de gaz, et les lectures ont été enregistrées et analysées.

Figure II. 39 : Circuit La surveillance de fuites de gaz.



Cette expérience vise à mesurer l'effet de l'ouverture de la fenêtre et du ventilateur sur différentes concentrations de gaz dans la

cuisine à l'aide d'un capteur de gaz MQ2. Les lectures seront comparées dans trois cas différents :

- Mesurer différentes concentrations de gaz et de fumée.
- Avant d'ouvrir le ventilateur et la fenêtre.
- Après avoir ouvert le ventilateur et la fenêtre.

1. Mesurer différentes concentrations de gaz et de fumée



Figure II. 40 : Lecture du capteur de gaz à différentes concentrations.

On remarque que les valeurs lues sur le capteur augmentent clairement avec l'augmentation de la concentration de gaz.

2. Avant d'ouvrir le ventilateur et la fenêtre



Figure II. 41 : Lecture du capteur de gaz Avant d'ouvrir le ventilateur et la fenêtre.

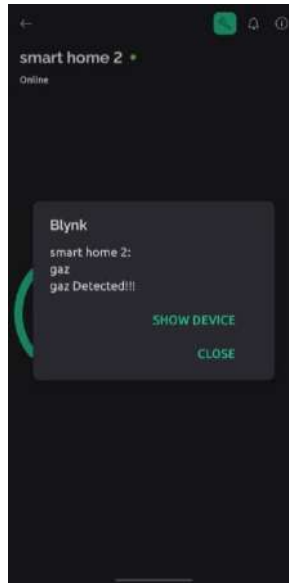


Figure II. 42 : Message d'avertissement du capteur de gaz dans l'application mobile Blynk.

On remarque que les valeurs lues sur le capteur sont élevées lorsqu'il est exposé au gaz avant d'ouvrir la fenêtre et le ventilateur, et la cloche d'alarme et led rouge s'allument et une alerte est envoyée à l'application Blynk.

3. Après avoir ouvert le ventilateur et la fenêtre

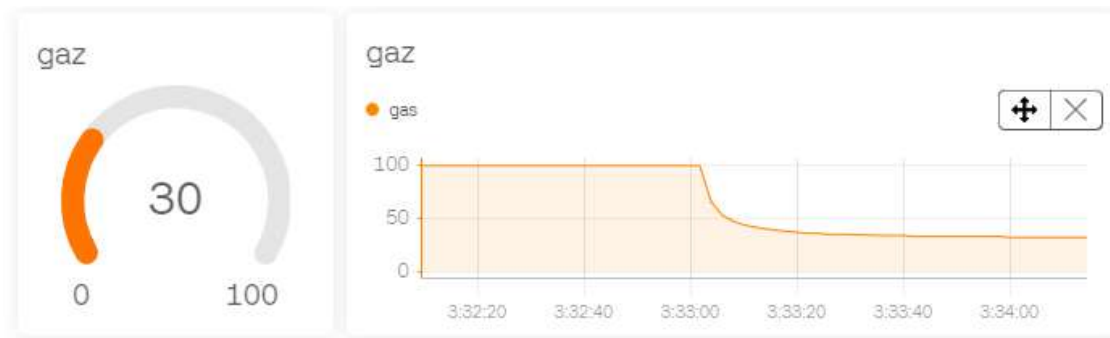


Figure II.43 : Lecture du capteur de gaz Après avoir ouvert le ventilateur et la fenêtre.

On remarque que les valeurs lues sur le capteur de gaz diminuent après l'ouverture de la fenêtre et du ventilateur et que led vert s'allume.

II. 2.3. Surveiller l'ouverture et la fermeture de la porte

Nous utilisons un capteur à ultrasons pour mesurer différentes distances entre celui-ci et la cible, et le connectons à un servomoteur pour ouvrir et fermer la porte. Une notification est envoyée par l'application Blynk lorsque la porte est ouverte.

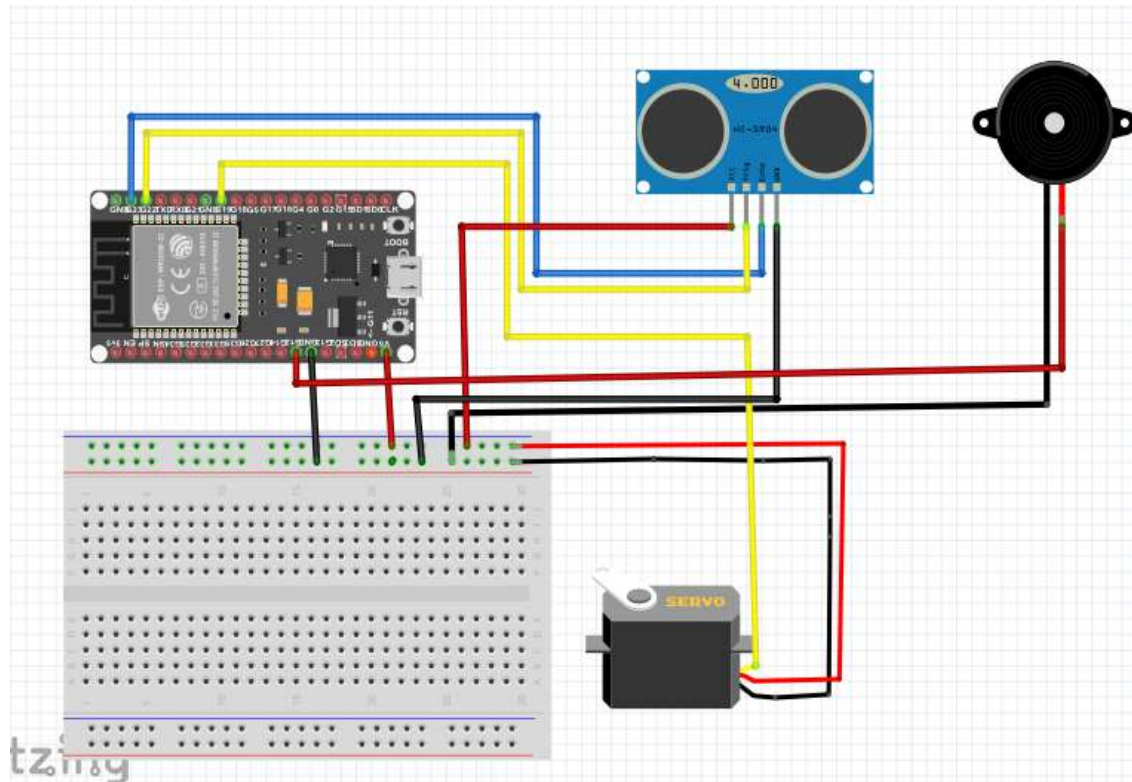


Figure II. 44 : Circuit du branchement du capteur à ultrason.



Figure II. 45 : la lecture du capteur à ultrason.

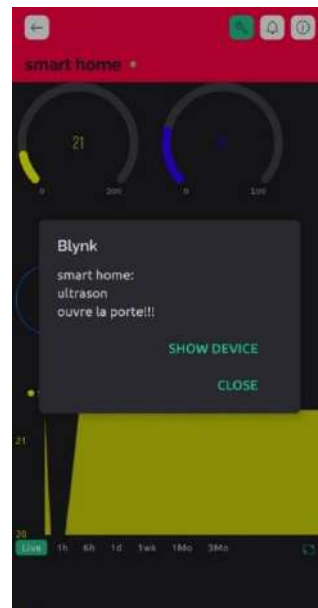


Figure II. 46 : Message d'alerte du capteur à ultrasons lorsque la porte est ouverte dans l'application mobile Blynk.

Nous notons que les résultats de la lecture du capteur à ultrason à longue distance sont instables et moins précis, tandis qu'à courte distance la lecture est plus précise, et lorsqu'une valeur inférieure à 10 cm est détectée, la porte s'ouvre et une alerte est envoyée au téléphone via l'application Blynk.

II. 2.4. Schéma principal

Le schéma électronique présentant la connexion électrique des différents composants et circuits est donné par la figure suivante, obtenue en utilisant le logiciel de prototypage Fritzing.

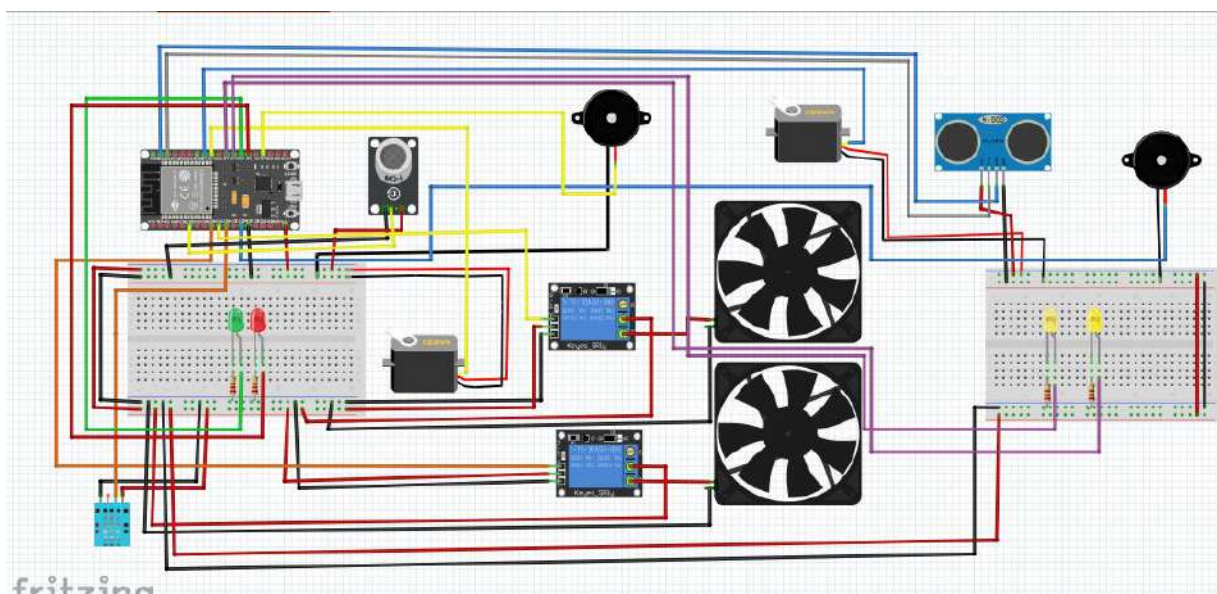


Figure II. 47 : Schéma principal.

II.3. Discussions :

II.3.1. La lecture des données de température et d'humidité

L'effet du fonctionnement du ventilateur peut s'expliquer par une meilleure circulation de l'air, qui permet de répartir plus efficacement la chaleur et l'humidité. Lorsque l'air d'une pièce est au repos, la chaleur et l'humidité ont tendance à s'accumuler dans certaines zones. Faire fonctionner un ventilateur aide à déplacer l'air, ce qui entraîne une répartition uniforme de la chaleur et de l'humidité et réduit l'accumulation.

Améliorer la circulation de l'air au sein de la pièce améliore le confort thermique. Une baisse de température, même légère, peut augmenter le confort, notamment dans les environnements chauds.

Faire fonctionner un ventilateur permet de rafraîchir l'air de la pièce, réduisant ainsi l'accumulation de polluants et d'odeurs. Cela peut contribuer à améliorer la qualité de l'air intérieur, ce qui a un impact positif sur la santé et la sécurité des occupants de la maison.

Sur la base des données collectées, nous pouvons conclure que le fonctionnement du ventilateur a un effet notable sur la température et l'humidité à l'intérieur de la pièce. Cet effet peut être bénéfique pour améliorer le confort thermique et la qualité de l'air intérieur. Par conséquent, les ventilateurs fonctionnels peuvent être adoptés comme l'une des solutions pour améliorer l'environnement interne des maisons intelligentes, améliorant ainsi l'efficacité des

systèmes de ventilation et de climatisation à l'intérieur de la maison.

II.3.2. La lecture des données de fuite de gaz

Le capteur MQ-2 s'est montré très efficace dans la détection des fuites de gaz dans un environnement de cuisine. Il peut être considéré comme un choix fiable pour la sécurité domestique, car il peut avertir rapidement les utilisateurs d'une fuite de gaz inflammable, réduisant ainsi le risque d'explosion ou d'incendie.

Les résultats ont montré qu'une ventilation efficace (ouvrir la fenêtre et allumer le ventilateur) peut réduire considérablement la concentration de gaz inflammables dans la cuisine. La ventilation contribue à réduire l'accumulation et la propagation des gaz dans l'environnement.

II.3.3. La lecture des données de capteur ultrason

Le capteur à ultrasons a montré une grande précision dans la détection de l'approche à des distances rapprochées inférieures à (30 cm), ce qui garantit que la porte n'est pas ouverte sans lorsque cela est nécessaire, car l'écho revient très rapidement et le signal est fort, et les lectures peuvent être interrompues. S'il y a de petits obstacles ou des changements rapides dans l'environnement.

Cependant, sur de longues distances, les lectures du capteur sont moins précises car le signal réfléchi devient plus faible, le temps nécessaire au retour de la rouille devient plus long et il existe un risque d'interférence avec d'autres signaux ou d'augmentation des réflexions multiples, ce qui peut affecter la précision du capteur, lectures.

II.4.Conclusion

Dans ce chapitre, les expériences ont démontré que la réponse du système était précise et rapide, ce qui a permis d'améliorer le confort de l'utilisateur. Les données extraites des capteurs ont également montré une amélioration notable des lectures et une réponse rapide dans l'envoi d'alertes instantanées via l'application Blynk de l'utilisateur. Cela confirme l'efficacité du système pour améliorer l'environnement domestique et le rendre plus sûr.

CONCLUSION GÉNÉRALE

Conclusion générale

En conclusion, la maison intelligente se présente comme une solution innovante et puissante pour relever les défis actuels de la gestion de l'habitat. En exploitant les avancées technologiques, notamment dans les domaines de l'Internet des objets (IoT), des capteurs sans fil, des microcontrôleurs tels que l'ESP32, des capteurs environnementaux comme le DHT11, des servomoteurs, des capteurs de gaz et de sonar, et de l'automatisation via des plateformes comme Blynk, elle permet une utilisation optimale des ressources énergétiques. Cette approche révolutionne les pratiques résidentielles traditionnelles en offrant un contrôle précis et adapté aux besoins spécifiques des occupants. Elle contribue ainsi à la préservation des ressources énergétiques, une question cruciale dans le contexte du changement climatique et de la croissance démographique mondiale. De plus, la maison intelligente ouvre la voie à un habitat plus durable et respectueux de l'environnement. En minimisant le gaspillage d'énergie, elle favorise la durabilité environnementale tout en améliorant le confort des occupants. Cependant, il est important de noter que l'utilisation de l'ESP32, du DHT11, des servomoteurs, des ventilateurs et de Blynk dans un système de maison intelligente présente certains défis et limitations :

ESP32 : Malgré sa puissance et sa connectivité Wi-Fi intégrée, l'ESP32 a une capacité de traitement limitée et une mémoire réduite, ce qui peut poser des problèmes pour des applications plus complexes. De plus, sa consommation d'énergie peut être relativement élevée par rapport à d'autres microcontrôleurs, ce qui peut être un problème pour les applications alimentées par batterie.

DHT11 : Le capteur DHT11, bien qu'abordable et facile à utiliser, a une précision relativement faible et une plage de fonctionnement limitée par rapport à d'autres capteurs d'humidité et de température. Il peut également être lent à répondre aux changements rapides de l'humidité ou de la température.

Servomoteurs : Les servomoteurs peuvent consommer beaucoup d'énergie et générer du bruit lorsqu'ils sont en fonctionnement. De plus, leur durée de vie peut être limitée, surtout s'ils sont utilisés fréquemment ou dans des conditions difficiles.

Ventilateurs : Les ventilateurs peuvent également consommer beaucoup d'énergie et générer

du bruit. De plus, ils peuvent nécessiter un entretien régulier pour éviter l'accumulation de poussière et d'autres particules qui peuvent réduire leur efficacité.

Blynk : Bien que Blynk soit une plateforme puissante pour la création d'applications IoT, elle nécessite une connexion Internet stable pour fonctionner correctement. De plus, la personnalisation de l'interface utilisateur peut être limitée, et il peut y avoir des frais associés à l'utilisation de certaines fonctionnalités ou à la gestion d'un grand nombre de dispositifs. Il est important de noter que ces défis et limitations ne sont pas insurmontables, et qu'il existe souvent des solutions ou des alternatives disponibles. Par exemple, l'utilisation de capteurs plus précis ou de microcontrôleurs plus puissants, l'optimisation de la consommation d'énergie, ou l'utilisation de plateformes IoT alternatives peuvent aider à surmonter certains de ces défis. De plus, une conception et une mise en œuvre soigneuses du système peuvent contribuer à minimiser ces problèmes et à maximiser les avantages de la maison intelligente.

En perspective, la maison intelligente, avec sa capacité d'adaptation et d'évolution en parallèle des avancées technologiques, promet de jouer un rôle déterminant dans la transformation de l'habitat du futur. Elle offre une opportunité non seulement d'améliorer la gestion de l'énergie dans l'habitat, mais aussi de contribuer à la sécurité et au bien-être des occupants face aux défis du changement climatique. L'intégration de capteurs supplémentaires, tels que des capteurs d'inondation, de monoxyde de carbone (CO), de radio-identification (RFID) pour l'ouverture des portes et des garages, et bien d'autres, pourrait enrichir la précision et l'efficacité du système de maison intelligente. Ces capteurs pourraient fournir des informations précieuses qui permettraient d'optimiser davantage les systèmes de la maison intelligente. L'introduction de l'intelligence artificielle pour la prédiction de la consommation énergétique pourrait ouvrir de nouvelles possibilités pour l'optimisation de l'énergie. Les modèles d'apprentissage automatique pourraient être formés pour prédire la consommation d'énergie en fonction des habitudes des occupants et des conditions environnementales, permettant ainsi une gestion de l'énergie encore plus précise et efficace. De plus, l'intégration de caméras de vision pourrait améliorer la sécurité et le confort des occupants. Ces caméras pourraient être utilisées pour surveiller l'environnement de la maison, détecter les mouvements suspects et même reconnaître les occupants pour personnaliser les paramètres de la maison en fonction de leurs préférences. Il est donc essentiel de poursuivre la recherche et le développement dans ce domaine, afin de maximiser les avantages de la maison intelligente et de surmonter les défis potentiels liés à son déploiement à grande échelle. Cela

nécessitera une collaboration étroite entre les chercheurs, les ingénieurs, les décideurs politiques et les autres parties prenantes pour garantir que les bénéfices de cette technologie soient pleinement réalisés. Cela souligne l'importance de l'innovation continue et de l'exploration de nouvelles technologies pour améliorer et enrichir les capacités des maisons intelligentes.

Bibliographie

- [1]: Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660.
- [2] : Alam, M. R., Reaz, M. B. I., & Ali, M. A. M. (2017). A review of smart homes—past, present, and future. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(3), 494-509.
- [3]: Perera, C., Zaslavsky, A., Christen P., & Georgakopoulos, D. (2014). Context-aware computing for the Internet of Things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414-454.
- [4]: Stojkoska, B. L. R., & Trivodaliev, K. V. (2017). A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140, 1454-1464.
- [5] : ZENNOUCHE Kahina Et HADJ ALI Zineb «Etude et réalisation d'un système de contrôle pour la sécurité, le confort et l'environnement d'une maison intelligente-Smart house», Mémoire De Master, Université De AKLI MOHAND OULHADJ- BOUIRA. 2019.
- [6] : HARITI Ayoub « CONCEPTION ET RÉALISATION D'UNE MAISON INTELLIGENTE », Mémoire De Master, Université de SAAD DAHLAB-BLIDA, 2023.
- [7] : Haouchine Imene Et Ait Mouhoub Sabrina « Etude de l'IoT et réalisation d'une maison intelligente », Mémoire De Master, Université de A.MIRA-BEJAIA, 2019.
- [8]: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.
- [9]: [ESP32 WROOM 32 GPIO |dzduino.](#)
- [10]: <https://www.arduino-circuit.com/sg90-servo-motor-with-arduino/>
- [11] : ADDOU Asmaa Et BAHOU Nawel «Réalisation d'une maison intelligente à base d'Arduino», Mémoire De Master, Centre Universitaire de Ain Témouchent - BELHADJ Bouchaib, 2020.
- [12] : <https://www.geya.net/fr/5v-relay-module-how-it-works-and-application/>
- [13] : OULARBI Chemseddine, SADOUNE Gaya « Etude et conception d'une serre agricole intelligente (Smart Green House) »Mémoire de fin d'étude, 2019, Université AKLI MOHAND OULHADJ-Bouira.
- [14] : MEGUEHOUT Walid Et MEGHZILI Yasser «Etude et réalisation d'une maison intelligente en utilisant un Arduino», Mémoire De Master, Université BOUSSOUF Abdelhafid - Mila, 2022.
- [15] : CHAIB Inas Et GAMRI Hiba «Conception et réalisation d'une maison intelligente à base de GSM», Mémoire De Master, Université Kasdi Merbah - Ouargla, 2022.
- [16] : Djehaiche Rania Et Benziouche Nihad «Etude Et Application D'un Système De Communication M2m », Mémoire De Master, Université De Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj, 2019.
- [17] : <https://blynk.io/>
- [18] : Baia-Rasso Abir«Conception d'un système de gestion d'une cité intelligente» Mémoire De Master, Université De BADJI MOKHTAR – ANNABA, 2021.

[19] : AIT BELKACEM Dihia Et ABDI Cylia « Conception et réalisation d'un robot télécommandé », Mémoire De Master, UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU, 2018.

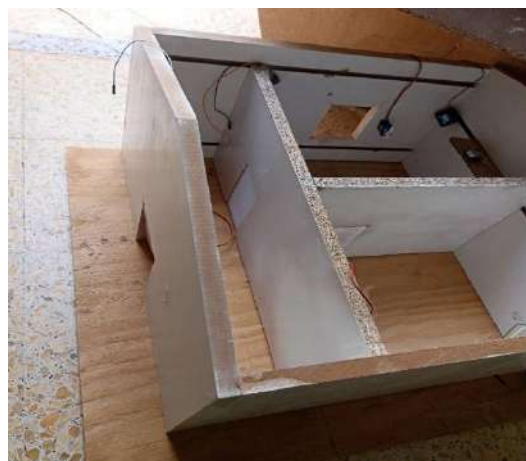
[20] : ZAZ Sofia Hanene Et BENKAHLA Lilia« Réalisation d'une maison intelligente utilisant un module WIFI», Mémoire De Master, Université Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj, 2021.

Annexe

Annexe A

La réalisation de la maison intelligente

1.Photos d'intérieur et d'extérieur de maquette de maison:



2.Photos La réalisation finale de la maison intelligente:



Annexe B

Programme globale de système

```
/* Fill-in your Template ID (only if using Blynk.Cloud) */
#define BLYNK_TEMPLATE_ID "TMPL2m-I0BMGJ"
#define BLYNK_TEMPLATE_NAME "gas"
#define BLYNK_AUTH_TOKEN "vipm2rmfoPkUA6DBzIMLcGqgLQG2aPX6"
// Your WiFi Credentials.
// Set password to "" for open networks.
char ssid[] = "Co ba";
char pass[] = "12233344441";

// define the GPIO connected with Sensors & LEDs
#define MQ2_SENSOR 35
#define GREEN_LED 4
#define RED_LED 2
#define blue 16
#define buzzer 15
#define buzzer2 12
#define relay 26
#define relay2 25
#define DHTPIN 27
#define DHTTYPE DHT11
#define echoPin 23 // CHANGE PIN NUMBER HERE IF YOU WANT TO USE A DIFFERENT
PIN
#define trigPin 22
long duration, distance;

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
BlynkTimer timer;
#include <ESP32Servo.h>
Servo myservo1;
Servo myservo2;
int servoPin=18;
int servoPin2=19;
int MQ2_SENSOR_Value = 0;

bool isconnected = false;
char auth[] = BLYNK_AUTH_TOKEN;
#define VPIN_BUTTON_1 V1
void checkBlynkStatus() { // called every 2 seconds by SimpleTimer
  isconnected = Blynk.connected();
  if (isconnected == true) {

    sendData();
    Serial.println("Blynk Connected");
  }
  else{
```

```
    Serial.println("Blynk Not Connected");
  }
}

void getSensorData()
{
  MQ2_SENSOR_Value = map(analogRead(MQ2_SENSOR), 0, 4095, 0, 100);
  if (MQ2_SENSOR_Value > 50 ){
    digitalWrite(GREEN_LED, LOW);
    digitalWrite(RED_LED, HIGH);
    digitalWrite(buzzer,HIGH);
    digitalWrite(relay, HIGH);
    myservo1.write(90);

  }
  else{
    digitalWrite(GREEN_LED, HIGH);
    digitalWrite(RED_LED, LOW);
    digitalWrite(buzzer,LOW);
    digitalWrite(relay , LOW);
    myservo1.write(0);
  }
  delay(1000);
}
#define DHTPIN 27
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  if (h>50)
  {
    digitalWrite(relay2, LOW);
    Serial.println(" FAN 2 ON");
  } else {
    digitalWrite(relay2, HIGH);
    Serial.println(" FAN 2 OFF");
    ///////////////doooooor///
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration * 0.034 / 2;
    String disp = String(distance);
    Serial.print("Distance: ");
    Serial.print(disp);
  }
}
```

```
Serial.println(" cm");
if (distance < 10) {
  digitalWrite(buzzer2,HIGH);
  Blynk.logEvent("ultrason", "open door!");
  myservo2.write(90); // تدوير محرك السيرفو إلى 90 درجة
  delay(1000);
  //delay(5000);
} else {
  myservo2.write(0);
  digitalWrite(buzzer2,LOW);
}
}
// You can send any value at any time.
// Please don't send more that 10 values per second.
Blynk.virtualWrite(V0, h);
Blynk.virtualWrite(V2, t);
Blynk.virtualWrite(V4, distance);
}
void sendData()
{
  Blynk.virtualWrite(VPIN_BUTTON_1, MQ2_SENSOR_Value);
  if (MQ2_SENSOR_Value > 50 )
  {
    Blynk.logEvent("gaz", "gaz détecté !!!");
  }
  Blynk.virtualWrite(V4, distance);
  if (distance < 10 )
  {
    Blynk.logEvent("ultrason", "ouvre la porte!!!");
  }
}
//////////leds//////////
BLYNK_WRITE(V3) {
  digitalWrite(16, param.asInt());
}
//////////
void setup()
{
  Serial.begin(115200);

  pinMode(MQ2_SENSOR, INPUT);
  myservo1.attach(servoPin);
  myservo2.attach(servoPin2);
  myservo2.write(0);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);
  pinMode(blue, OUTPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(buzzer2,OUTPUT);
  pinMode(relay,OUTPUT);
  pinMode(relay2,OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  //
```

```
WiFi.begin(ssid, pass);
timer.setInterval(2000L, checkBlynkStatus); // check if Blynk server is connected every 2 seconds
Blynk.config(auth);
delay(1000);
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
dht.begin();
timer.setInterval(1000L, sendSensor);
}

void loop()
{
  getSensorData();
  Blynk.run();
  timer.run();
}
```