

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE MINISTÈRE DE
L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDERRAHMANE MIRA - BEJAIA
FACULTÉ DE SCIENCES EXACTES
DÉPARTEMENT INFORMATIQUE



MÉMOIRE DE FIN DE CYCLE
EN VUE D'OBTENTION DU DIPLÔME DE MASTER
OPTION ADMINISTRATION ET SÉCURITÉ DES RÉSEAUX

Thème

Systeme de surveillance à distance d'une ruche d'abeilles basé sur l'IoT

Réalisé par :

Mlle. BOUKHEZZAR Imane
Mlle. HAOUCHINE Melyna

Soutenu le 01 Juillet 2024 devant le jury composé de :

M. BENNAI Yani Athmane	Encadrant
M. SADI Mustapha	Président
M. DJEBARI Nabil	Examineur

Promotion 2023 - 2024

Remerciements

Nous exprimons tout d'abord notre profonde gratitude à Dieu le Tout-Puissant pour nous avoir accordé la force, la volonté et la connaissance nécessaires à l'accomplissement de ce travail.

*Nous tenons à exprimer notre profonde gratitude envers notre encadrant, **M. BENNAI Yani Athmane**, pour son soutien indéfectible, sa disponibilité constante, ses conseils et ses encouragements qui ont grandement enrichi notre expérience tout au long de ce projet.*

*Nous exprimons également notre reconnaissance et nos sincères remerciements aux membres du jury, **M. SADI Mustapha** et **M. DJEBARI Nabil**, pour avoir accepté d'examiner ce travail.*

Nous tenons à adresser nos plus vifs remerciements au laboratoire de recherche LAMOS de l'université de Bejaia pour nous avoir fourni le matériel utilisé dans notre projet. Leur soutien technique a été indispensable pour la réussite de ce travail.

*Nous remercions chaleureusement tout le personnel de l'entreprise Api-ruche de Bejaia, en particulier **M. OURAMEDANE Salim**, pour son aide précieuse durant notre stage.*

Nous souhaitons également exprimer notre profonde reconnaissance à nos familles pour leur soutien inconditionnel et leurs encouragements constants. Leur amour et leur soutien ont été une source d'inspiration et de motivation tout au long de cette recherche.

Enfin, nous tenons à remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

BOUKHEZZAR Imane, HAOUCHINE Melyna

TABLE DES MATIÈRES

Remerciements	I
Table des matières	II
Table des figures	VI
Liste des tableaux	IX
Liste des abréviations	X
Introduction	1
1 Généralités et contexte	2
1.1 Introduction	2
1.2 Introduction aux ruches d'abeilles	2
1.2.1 Présentation d'une ruche moderne	2
1.2.2 L'importance des ruches d'abeilles	3
1.2.3 Les produits d'une ruche	3
1.3 La santé d'une ruche d'abeilles	6
1.3.1 Les principaux critères de bonne santé d'une ruche	6
1.3.2 Les causes de la disparition des abeilles	6
1.3.3 Les solutions traditionnelles de surveillance des ruches	7
1.4 Potentiel de l'Internet des objets dans la surveillance des ruches	8
1.4.1 La définition de l'IoT	8
1.4.2 Objets connectés	8
1.4.3 Les étapes clés du fonctionnement de l'Internet des objets (IoT)	9

1.4.4	Les technologies fondamentales sur lesquelles repose l'IoT	9
1.4.5	Les avantages de l'IoT	10
1.4.6	Les limites de l'IoT	10
1.4.7	Le système Arduino	11
1.4.7.1	Les cartes Arduino	11
1.4.7.2	L'IDE Arduino	15
1.4.7.3	Les avantages de l'Arduino	16
1.4.8	Comment l'Internet des objets peut améliorer la surveillance des ruches	17
1.5	Définition des objectifs de l'étude	17
1.6	Conclusion	18
2	Présentation de l'organisme d'accueil	19
2.1	Introduction	19
2.2	Présentation de l'entreprise Api-ruche	19
2.2.1	Superficie	20
2.2.2	Localisation	20
2.2.3	Capacité de production et effectifs	21
2.2.4	Organigramme de l'entreprise	21
2.2.5	Équipements	22
2.3	Domaine d'activité de l'entreprise	22
2.3.1	Les activités principales	23
2.3.1.1	Les fonctions spécifiques de l'entreprise	23
2.3.1.2	La transhumance des ruches	24
2.3.2	Les activités secondaires	24
2.4	Tâches effectuées pendant notre stage	25
2.5	Problématique	25
2.6	Conclusion	26
3	Solution proposée	27
3.1	Introduction	27
3.2	Motivation	27
3.3	Le module ESP8266	27
3.3.1	La carte Wemos D1 R2	28
3.3.2	Les composants de la carte WeMos	28
3.3.3	Pourquoi choisir la WeMos D1 R2 ?	29
3.4	Les capteurs	29

3.5	Les actionneurs	30
3.6	L'ensemble des composants de notre système	31
3.6.1	Les capteurs utilisés	31
3.6.1.1	Le capteur de température et l'humidité DHT22	31
3.6.1.2	Le capteur de poids "Jauges de contraintes et amplificateur HX711"	32
3.6.1.3	Le capteur de son (bruit) KY-038	34
3.6.1.4	Le capteur de distance à ultrasons HC-SR04	34
3.6.1.5	Le module capteur antivol SIM808 GSM/GPRS/GPS	35
3.6.1.6	Le capteur de flammes YG1006	36
3.6.2	Le matériel utilisé	37
3.6.2.1	Platine d'Expérimentation (breadboard)	37
3.6.2.2	Câbles de connexion (jumper wires)	37
3.6.2.3	Écran LCD	38
3.6.2.4	Batterie rechargeable LiPo 2600mAh 11.1V	39
3.7	La plateforme Blynk	40
3.7.1	Qu'est ce que Blynk ?	40
3.7.2	Composants de la plateforme Blynk	40
3.7.3	Comment fonctionne Blynk ?	40
3.7.4	Exemples d'utilisation de l'application Blynk	41
3.7.5	Avantages de Blynk	41
3.8	Le principe de fonctionnement de la solution	42
3.8.1	Les étapes à réaliser	44
3.8.2	Les contraintes du système	44
3.8.3	L'estimation budgétaire du système	44
3.8.4	Les avantages de ce système	45
3.9	Conclusion	46
4	Mise en œuvre et réalisation du système	47
4.1	Introduction	47
4.2	Branchement de l'afficheur LCD	47
4.3	Branchement et programmation du capteur de température et d'humidité DHT22	48
4.3.1	Le câblage	48
4.3.2	Le code utilisé avec le capteur DHT22	49
4.3.3	La description des fonctions utilisées	50

4.4	Branchement et programmation du capteur de distance ultrasonique HC-SR04	52
4.4.1	Le câblage	52
4.4.2	Le code utilisé avec le capteur HC-SR04	53
4.4.3	La description des fonctions utilisées	55
4.5	Branchement et programmation du capteur de flammes YG1006	56
4.5.1	Le câblage	56
4.5.2	Le code utilisé avec le capteur YG1006	57
4.5.3	La description des fonctions utilisées	57
4.6	Branchement et programmation du capteur de son KY-038	58
4.6.1	Le câblage	58
4.6.2	Le code utilisé avec le capteur KY-038	59
4.6.3	La description des fonctions utilisées	61
4.7	Branchement et programmation du capteur de poids	61
4.7.1	Le câblage	61
4.7.2	Le code utilisé avec le capteur de poids	63
4.7.3	La description des fonctions utilisées	64
4.8	Branchement et programmation du module SIM808 GSM/GPRS/GPS	65
4.8.1	Le câblage	65
4.8.2	Le code utilisé avec le module SIM808	67
4.8.3	La description des fonctions utilisées	69
4.9	Configuration de l'application Blynk	70
4.9.1	La création d'un compte sur l'application Blynk	70
4.9.2	La création d'un projet personnalisé sous le nom "Mon Projet"	70
4.9.3	La création d'un nouvel appareil ESP8266 dans l'application Blynk	70
4.9.4	La configuration des widgets dans le tableau de bord	71
4.9.5	La création des broches virtuelles	72
4.9.6	La configuration des événements et des notifications	72
4.10	Installation du système complet dans une ruche d'abeilles	73
4.11	Tests et résultats du système	74
4.12	Conclusion	76
	Conclusion	77

TABLE DES FIGURES

1.1	La présentation d'une ruche moderne	3
1.2	Le miel	4
1.3	La propolis	4
1.4	La gelée royale	5
1.5	Le pollen	5
1.6	La cire	5
1.7	Les étapes du fonctionnement de l'IoT	9
1.8	La carte arduino Uno	12
1.9	La carte arduino Mega 2560	13
1.10	La carte arduino Mega ADK	13
1.11	La carte arduino Due	14
1.12	La carte arduino Nano	14
1.13	La carte arduino Yun	14
1.14	Interface de l'IDE Arduino	15
2.1	L'entreprise Api-ruche	20
2.2	L'emplacement de l'entreprise Api-ruche sur Google Maps	21
2.3	L'organigramme représentant la structure de l'entreprise Api-ruche.	22
2.4	L'évolution de la production de miel de 2019 à 2023 dans l'entreprise Api-ruche	23
3.1	La carte WeMos D1 R2	28
3.2	Les principaux composants d'une carte WeMos D1 R2	28
3.3	Le fonctionnement d'un capteur	30
3.4	Le fonctionnement d'un actionneur	30

3.5	Le capteur DHT22	31
3.6	Une jauge de contrainte de 50 KG	32
3.7	La présentation des résistances au niveau des cellules de charge	33
3.8	L'amplificateur HX711	33
3.9	Le capteur de son KY-038	34
3.10	Le capteur de distance ultrasonique HC-SR04	35
3.11	Le module SIM808GSM/GPRS/GPS	36
3.12	Le module capteur de flammes YG1006	37
3.13	La plaque d'expérimentation de 1660 points	37
3.14	La différence entre les câbles mâle-mâle et les câbles mâle-femelle	38
3.15	L'écran LCD	38
3.16	L'adaptateur I2C	39
3.17	La batterie rechargeable LiPo 2600mAh 11.1V	39
3.18	Le fonctionnement de Blynk	41
3.19	Le schéma montrant l'architecture de la ruche connectée	43
3.20	Le schéma montrant l'architecture du fonctionnement global de notre système	43
4.1	Le montage de l'écran LCD avec son adaptateur I2C	47
4.2	Schéma de connexion du capteur DHT22 avec l'afficheur LCD I2C	48
4.3	Branchement du capteur de température et d'humidité DHT22	49
4.4	Schéma de connexion du capteur de distance ultrasonique HC-SR04	53
4.5	Branchement du capteur de distance ultrasonique HC-SR04	53
4.6	Schéma de connexion du capteur de flammes YG1006	56
4.7	Branchement du capteur de flammes YG1006	56
4.8	Schéma de connexion du capteur de son KY-038 avec l'afficheur LCD I2C	59
4.9	Branchement du capteur de son KY-038	59
4.10	Schéma de connexion du capteur de poids avec l'afficheur LCD I2C	62
4.11	Branchement du capteur de poids	62
4.12	Le support 3D utilisé avec le capteur de poids	63
4.13	Schéma de connexion du module SIM808	66
4.14	Branchement du module SIM808	66
4.15	Illustration de l'étape de création de "Mon projet"	70
4.16	Illustration de l'étape de création d'un nouvel appareil	71
4.17	Récupération des données d'authentification Blynk	71
4.18	Illustration des widgets créés	72

Table des figures

4.19	Illustration des broches virtuelles créées	72
4.20	Illustration des événements créés	73
4.21	Installation du système complet dans une ruche d'abeilles	73
4.22	Visualisation des résultats sur le tableau de bord de Blynk	74
4.23	Les différentes alertes détectées	75
4.24	Résultats récupérés par le capteur GPS	76

LISTE DES TABLEAUX

1.1	Les types d'Arduino les plus répandus	12
3.1	Description des principaux composants de la carte WeMos D1 R2 . . .	29
3.2	Liste des composants avec leurs quantités et prix	45
4.1	Description des fonctions principales utilisées avec le capteur de température et d'humidité	52
4.2	Description des fonctions principales utilisées avec le capteur de distance à ultrasons.	55
4.3	Description des fonctions principales utilisées avec le capteur de flammes	58
4.4	Description des fonctions principales utilisées avec le capteur de son. .	61
4.5	Description des fonctions principales utilisées avec le capteur de poids	65
4.6	Description des fonctions principales utilisées avec le module SIM808	69

LISTE DES ABBREVIATIONS

2G	Second Generation
3D	3-Dimensional
3G	Third Generation
5G	Fifth Generation
ADC	Analog-to-Digital Converter
AO	Analog Output
AREF	Analog Reference
AT	Attention
CO₂	Carbon Dioxide
COM	Communication
DHT22	Digital Humidity and Temperature Sensor 22
DO	Digital Output
DT	Data
E/S	Entrée Sortie
GM	Google Maps
GND	Ground
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Internet Protocol
IR	Infrared
iOS	iPhone Operating System
JS	JavaScript
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LiPo	Lithium Polymer
LTE	Long-Term Evolution
OUT	Output
PWM	Pulse Width Modulation
RST	Reset
RX	Receive
RXD	Receive Data

Liste des abréviations

SCK	S erial C lock
SCL	S erial C lock L ine
SDA	S erial D ata A ine
SIM	S ubscriber I dentify M odule
SMS	S hort M essage S ervice
TCP	T ransmission C ontrol P rotocol
TRIG	T rigger
TXD	T ransmit D ata
USB	U niversal S erial B us
VCC	V oltage C ommon C ollector
Wi-Fi	W ireless F idelity

Introduction générale

Dans un monde où la technologie façonne notre quotidien, l'émergence de l'Internet des Objets (IoT) représente une avancée majeure dans la manière dont nous interagissons avec notre environnement. Popularisé par Kevin Ashton en 1999, l'IoT offre un cadre pour connecter et interagir avec une multitude d'objets physiques via Internet, ouvrant ainsi de nouvelles possibilités dans divers domaines, y compris l'apiculture.

Une application spécifique de l'IoT dans le domaine apicole est la surveillance des ruches d'abeilles. Les abeilles jouent un rôle vital dans la pollinisation des cultures et la préservation de la biodiversité, mais elles font face à de nombreux défis, notamment les maladies, les parasites et les changements environnementaux. La surveillance à distance des ruches grâce à des capteurs IoT, offre aux apiculteurs la possibilité de surveiller en temps réel l'état de leurs colonies, de détecter rapidement les problèmes et de prendre des mesures préventives.

Dans le cadre de notre projet de mémoire de fin d'études, nous nous penchons sur le développement d'un système de surveillance à distance pour une ruche d'abeilles, basé sur la technologie IoT.

Notre objectif est de concevoir et de mettre en œuvre un prototype fonctionnel de ruche connectée, capable de collecter et de transmettre des données environnementales et opérationnelles à partir de capteurs intégrés dans la ruche.

Pour présenter ce travail, notre mémoire est organisé comme suit :

Le premier chapitre présente les ruches d'abeilles et l'IoT, offrant une compréhension de base des concepts théoriques clés.

Le deuxième chapitre introduit l'organisme d'accueil, l'entreprise Api-ruche, et la problématique qui a conduit à notre solution.

Le troisième chapitre explore en détail la solution proposée, mettant en avant les technologies sélectionnées et les motivations qui sous-tendent leur sélection.

Le quatrième chapitre détaille la mise en œuvre de notre système de surveillance, ainsi que les résultats obtenus. Enfin, une conclusion clôturera notre étude en abordant les perspectives futures dans ce domaine en évolution constante.

CHAPITRE 1

GÉNÉRALITÉS ET CONTEXTE

1.1 Introduction

Dans ce chapitre introductif, nous explorons l'importance des abeilles dans la pollinisation des plantes et la préservation de la biodiversité. Nous mettons en lumière les enjeux majeurs liés à leur protection et à la préservation de leurs habitats. De plus, nous abordons la notion fondamentale de la santé d'une ruche en identifiant les critères clés qui la déterminent, ainsi que les méthodes traditionnelles utilisées par les apiculteurs pour surveiller leurs colonies, pour terminer avec l'internet des objets et son apport potentiel.

1.2 Introduction aux ruches d'abeilles

1.2.1 Présentation d'une ruche moderne

La ruche moderne, représentée dans la figure 1.1, est conçue selon un modèle rationnel à cadres mobiles. Cette structure comprend un compartiment de base, appelé "corps", où les cadres sont disposés côte à côte. La colonie commence à construire ses rayons sur ces cadres, les utilisant à la fois pour le couvain et les réserves de nourriture. Au-dessus du corps, l'apiculteur place des structures similaires, appelées "hausses", destinées à stocker le surplus de miel. Seules les ouvrières ont accès aux hausses, car une "grille à reine" est installée entre le corps et les hausses pour empêcher la reine de pondre dans ces dernières. Ainsi, le miel récolté provient exclusivement des hausses.[1]

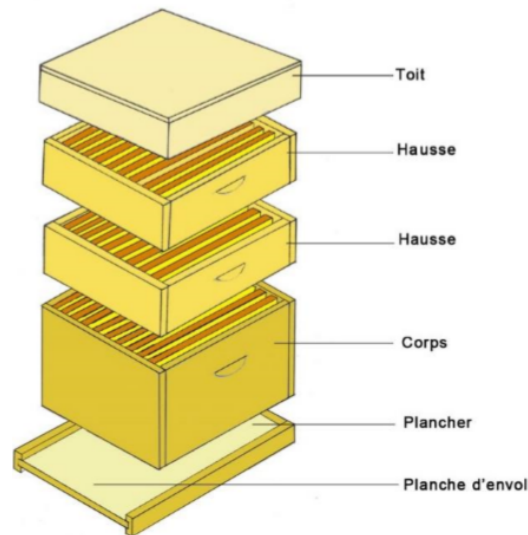


FIGURE 1.1 – La présentation d'une ruche moderne [2]

1.2.2 L'importance des ruches d'abeilles

"Si l'abeille venait à disparaître, l'humanité n'aurait plus que quatre ans à vivre."
- Albert Einstein -.

Les abeilles, apparues il y a plus de 100 millions d'années en même temps que les premières plantes à fleurs, ont évolué en parallèle avec la flore sauvage, contribuant ainsi à la biodiversité végétale. Elles jouent un rôle essentiel en tant qu'indicateur de la santé de notre planète, et leur déclin suscite des préoccupations parmi les scientifiques.[3]

Quatre-vingt pour cent des espèces végétales dépendent directement de la pollinisation par les insectes. L'abeille domestique, en tant qu'agent pollinisateur, occupe une place significative dans cet équilibre. En butinant diverses espèces de plantes à fleurs, les abeilles favorisent le développement en quantité et en qualité des fruits et des graines, essentiels pour la survie des espèces végétales, tout en nourrissant une variété d'insectes, d'oiseaux et de mammifères. Maintenir l'équilibre entre ces insectes pollinisateurs revêt donc une importance primordiale.[3]

En ce qui concerne l'apiculture, elle représente une option intéressante pour diversifier les revenus, surtout à la lumière de l'effondrement des prix du pétrole. Outre sa contribution à la diversification des revenus familiaux, elle nécessite peu de temps et d'investissement, en particulier à petite échelle. Les produits principaux de la ruche, tels que la cire et le miel, offrent des avantages économiques, tandis que d'autres produits comme le pollen, le couvain, la gelée royale ou la propolis présentent des propriétés nutritionnelles et thérapeutiques de premier ordre.[4]

1.2.3 Les produits d'une ruche

La ruche représente une véritable usine à trésors, produisant une variété de produits précieux, tels que :

- **Le miel** : Substance sucrée naturelle produite par les abeilles à partir du nectar des fleurs. Stocké comme aliment pour l'hiver, le miel contient des enzymes agissant comme catalyseurs dans la digestion humaine. Il est également utilisé comme remède pour favoriser la guérison rapide des blessures et des ulcères.[5]

Il est représenté dans la figure 1.2.



FIGURE 1.2 – Le miel
[6]

- **La propolis** : Collectée principalement auprès des conifères, la propolis est utilisée par les abeilles pour boucher les fissures de leur ruche, agissant comme un mastic naturel. Dotée de propriétés anti-infectieuses, elle contribue au maintien de la propreté à l'intérieur de la ruche.[5]

Elle est représentée dans la figure 1.3.



FIGURE 1.3 – La propolis
[7]

- **La gelée royale** : Utilisée pour nourrir les larves et la reine adulte, la gelée royale possède une haute valeur nutritive. Elle est utilisée comme fortifiant naturel, pour lutter contre les effets du vieillissement et renforcer le système immunitaire.[5]

Elle est représentée dans la figure 1.4.



FIGURE 1.4 – La gelée royale
[8]

- **Le pollen** : Source de protéines pour les abeilles, le pollen est spécifiquement utilisé pour nourrir les larves du couvain. Il est également riche en protéines, possède des propriétés antioxydantes et régulatrices du système immunitaire humain.[5]

Il est représenté dans la figure 1.5.



FIGURE 1.5 – Le pollen

- **La cire** : La cire, représentée dans la figure 1.6, est une substance grasse produite par les abeilles, utilisée pour la construction des rayons et l'opercule des alvéoles. Cette matière imperméable trouve également des applications dans les produits cosmétiques en raison de ses propriétés hydratantes et protectrices de la peau.[5]



FIGURE 1.6 – La cire

1.3 La santé d'une ruche d'abeilles

1.3.1 Les principaux critères de bonne santé d'une ruche

Le bien-être d'une ruche est essentiel en apiculture, car il affecte directement la productivité et la stabilité de la colonie d'abeilles. Voici quelques indicateurs simples de bonne santé d'une ruche :

- **Population d'abeilles saine** : Une population saine est cruciale, avec environ 15 000 à 30 000 abeilles en hiver et entre 30 000 et 60 000 en période d'activité. La présence d'ouvrières, de faux bourdons et d'une reine productive est un bon signe. De plus, un couvain bien développé montre que la colonie est productive.[9]
- **Température** : Pour que les larves survivent, la température doit être maintenue entre 33°C et 36°C. Si la température dépasse 30°C, cela signifie que la population augmente, mais si elle descend en dessous de 18°C en hiver, cela peut indiquer des problèmes sérieux.[10]
- **Humidité** : L'humidité, entre 50% et 60%, est importante pour l'élevage des larves. En cas de sécheresse, les abeilles maintiennent l'humidité en collectant de l'eau et en favorisant son évaporation.[11]
- **Poids** : Le poids de la ruche indique sa capacité à se développer et à produire du miel. Un poids stable signifie qu'il y a suffisamment de nourriture pour survivre aux moments difficiles ou à l'hiver. Il est influencé par divers facteurs tels que la région et la saison. Une ruche standard peut peser jusqu'à environ 40 kg [12], et peut parfois atteindre jusqu'à 150 kg avec l'ajout de hausses additionnelles.
- **Son et vibration** : Les sons et vibrations émis par la ruche peuvent indiquer l'activité des abeilles, la présence de la reine et d'autres aspects de la vie de la colonie. Des variations inhabituelles peuvent signaler des problèmes potentiels. Les abeilles émettent des sons dont l'amplitude et la fréquence varient, allant de moins de 100 Hz à 3600 Hz.[13]
- **Position** : L'emplacement de la ruche peut être un indicateur de sa santé, car il influe sur des facteurs tels que l'exposition au soleil, la protection contre les intempéries comme le vent, et la proximité des sources de nourriture.[14]
- **Comportement calme des abeilles** : Le comportement des abeilles à l'entrée de la ruche donne des indications sur leur santé générale. Des signes de stress, comme des regroupements désordonnés, peuvent signaler la présence de prédateurs, de maladies ou de perturbations environnementales. De plus, une ruche propre, sans débris ou moisissures, est un indicateur de la gestion hygiénique de la colonie et que les abeilles font leurs tâches quotidiennes sans problèmes liés à leurs comportements.[14]

1.3.2 Les causes de la disparition des abeilles

Le syndrome de l'effondrement des colonies est un phénomène complexe qui entraîne la disparition soudaine de colonies entières d'abeilles. Les causes de ce syndrome varient selon les régions géographiques et les pratiques agricoles locales. Voici les principales raisons identifiées :

- **Substances chimiques et pesticides** : Les abeilles sont exposées à de nombreux produits chimiques et pesticides dans leur environnement, surtout dans les cultures traitées chimiquement. Ces substances peuvent être toxiques, affaiblissant le système immunitaire des abeilles et les rendant plus sensibles aux maladies et aux parasites.[3][15]
- **Causes pathogènes** : Les colonies d'abeilles peuvent être affectées par différents parasites, virus et champignons qui compromettent leur santé, contribuant ainsi à leur effondrement. Parmi ceux-ci, le *Varroa destructor*, un acarien parasite, est particulièrement nuisible aux colonies d'abeilles.[16]
- **Manque de végétation et agriculture intensive** : L'agriculture intensive, caractérisée par la monoculture et l'utilisation excessive d'engrais et de pesticides, réduit la diversité végétale. Cela prive les abeilles de sources de nourriture variées, affaiblissant les colonies en réduisant leur accès à des nutriments essentiels.[17]
- **Causes génétiques** : Les abeilles sélectionnées pour leur productivité en miel peuvent présenter une faible diversité génétique, les rendant moins aptes à s'adapter à leur environnement et plus vulnérables aux maladies et au stress environnemental.[18]
- **Pollution électromagnétique** : Des études suggèrent que la pollution électromagnétique, provenant des champs électriques et magnétiques générés par les lignes électriques et les appareils électroniques, peut perturber le comportement des abeilles et altérer leur capacité à naviguer et à retrouver leur ruche.[19]
- **Pratiques des apiculteurs** : Les pratiques des apiculteurs, telles que le transport des ruches pour la pollinisation des cultures, peuvent également contribuer au stress des abeilles en les exposant à des conditions environnementales variables et en affaiblissant leurs colonies par des déplacements fréquents.[20]

1.3.3 Les solutions traditionnelles de surveillance des ruches

La surveillance des ruches revêt une importance essentielle pour les apiculteurs afin d'assurer la santé et la productivité des colonies d'abeilles. Les méthodes traditionnelles de surveillance impliquent généralement des inspections manuelles. Voici quelques-unes de ces solutions traditionnelles :

- **Inspection visuelle** : Les apiculteurs réalisent régulièrement des inspections visuelles des ruches pour évaluer la population d'abeilles, vérifier la présence de la reine, observer la construction des rayons de cire et détecter la présence de maladies ou de parasites.[21]
- **Pesée des ruches** : Pour peser une ruche, deux méthodes sont couramment utilisées : soulever la ruche et placer un pèse-personne en dessous, ou utiliser un peson mécanique. Bien que ces méthodes puissent indiquer la quantité de miel stocké et la population d'abeilles, elles ne permettent pas d'obtenir des informations détaillées sur l'état interne de la ruche.
- **Contrôle de la température** : Les variations de température peuvent signaler des problèmes potentiels. Certains apiculteurs utilisent des thermomètres pour surveiller la température à l'intérieur de la ruche.

- **Analyse des entrées/sorties d’abeilles** : Observer le comportement des abeilles à l’entrée de la ruche est essentiel. Cela peut fournir des indications sur l’état de la colonie. Par exemple, un faible trafic d’abeilles à l’entrée peut indiquer un problème, comme une diminution de la population d’abeilles ou un problème de santé de la colonie.
- **Écoute des bruits de la ruche** : Certains apiculteurs expérimentés utilisent leur ouïe pour détecter des problèmes à l’intérieur de la ruche, comme le bourdonnement particulier des abeilles ou l’absence de bruit, qui peut indiquer un problème.

1.4 Potentiel de l’Internet des objets dans la surveillance des ruches

1.4.1 La définition de l’IoT

L’Internet des objets (IoT, Internet of Things) englobe à la fois le processus de connectivité des objets physiques à Internet et l’infrastructure réseau qui permet cette interconnexion. Bien que l’IoT soit conçu pour opérer de manière autonome, il offre également la possibilité d’interaction entre les individus et les machines, que ce soit pour la configuration, le paramétrage ou simplement l’accès aux informations.

Le terme ‘objets’ ne se limite pas seulement aux dispositifs de la vie quotidienne tels que la domotique et les montres connectées, mais englobe également des équipements médicaux, des machines agricoles, des éléments de chaînes d’approvisionnement, des robots industriels, et même des feux de signalisation routière, c’est-à-dire que l’IoT concerne tout élément capable de recueillir, de transmettre et de partager des données au sein d’un réseau.[22]

1.4.2 Objets connectés

Un objet connecté est un objet physique équipé de capteurs ayant la capacité de se connecter à un réseau de communication tel que l’Internet des objets, via des moyens tels que le Wi-Fi, le Bluetooth et le réseau Internet mobile, notamment la 5G.

L’objet peut avoir sa propre fonctionnalité, qu’elle soit mécanique ou électrique. Il est conçu pour communiquer, souvent une simple information, comme la détection de la lumière ou une mesure de température, et peut, selon les cas, recevoir, stocker, traiter et transmettre des données. Ces objets peuvent être autonomes ou fonctionner avec un smartphone ou une tablette permettant de les contrôler ou de servir de relais pour échanger des données.[23]

1.4.3 Les étapes clés du fonctionnement de l'Internet des objets (IoT)

Le fonctionnement de l'Internet des objets repose sur plusieurs étapes clés, représentées dans la figure 1.7, permettant la capture et l'exploitation efficace des données :

- **Collecte des données** : Les capteurs intégrés aux terminaux IoT enregistrent une gamme variée de données, comme des mesures de température simples ou des flux vidéo en temps réel, en fonction de leur programmation.
- **Transmission des données** : Ces terminaux IoT transmettent ensuite les données via des connexions réseau vers un système cloud, un autre terminal, ou les stockent localement pour un traitement en périphérie.
- **Traitement des données** : À cette étape, le logiciel programmé analyse les données pour déclencher des actions spécifiques, telles que l'activation d'un dispositif ou l'envoi d'une notification.
- **Exploitation des données** : L'analyse des données issues de l'ensemble des terminaux d'un réseau IoT génère des insights significatifs, facilitant la prise de décisions éclairées et la mise en œuvre de mesures appropriées.[24]

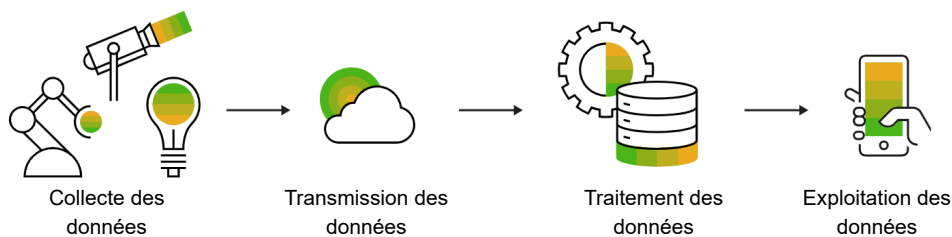


FIGURE 1.7 – Les étapes du fonctionnement de l'IoT

Ce processus permet ainsi la gestion à distance, l'automatisation des tâches, et une amélioration globale du processus décisionnel, offrant ainsi des avantages significatifs en termes d'efficacité opérationnelle et de prise de décision.[24]

1.4.4 Les technologies fondamentales sur lesquelles repose l'IoT

Les progrès récents dans diverses technologies ont facilité la concrétisation de l'Internet des objets, bien que le concept ne soit pas nouveau. Voici quelques-unes de ces avancées :

- **Réseaux de capteurs sans fil (WSN)** : Composés de dispositifs distribués équipés de capteurs, ils surveillent l'environnement et les conditions physiques, communiquant via Wi-Fi ou Bluetooth. Les réseaux comprennent des nœuds finaux, des routeurs et un coordinateur qui connecte le WSN à l'Internet, souvent en utilisant des protocoles comme IEEE802.15.4, ESP8266 et ZigBee.
- **Cloud computing** : Permet l'accès à la demande à des ressources informatiques telles que et serveurs réseau, stockage, applications et services. Il offre

une flexibilité une échelle accrue pour le développement et le déploiement d'applications IoT.

- **Big data analytics** : Traite et analyse de grandes quantités de données générées par les systèmes IoT en utilisant des outils avancés pour gérer le volume, la vitesse, la variété, la véracité et la valeur des données. Cela permet d'extraire des informations essentielles pour prendre des décisions éclairées.
- **Systèmes embarqués** : Des systèmes informatiques spécialisés comprenant des microprocesseurs ou des microcontrôleurs, de la mémoire, des unités de réseau, des unités d'entrée/sortie et des solutions de stockage, conçus pour effectuer des tâches spécifiques. Exemples de systèmes embarqués : montre numérique, appareil photo numérique, distributeurs automatiques.
- **Protocoles de communication** : Établissent les règles et les normes pour l'échange de données entre les dispositifs IoT. Ils assurent l'interopérabilité, la connectivité réseau et le couplage aux applications, définissant les formats de données, l'encodage, le routage et le contrôle des flux de données.[25]

1.4.5 Les avantages de l'IoT

Les bénéfices découlant de l'Internet des objets sont vastes et variés, notamment :

- **Automatisation et efficacité** : L'IoT permet l'automatisation de nombreuses tâches, améliorant ainsi l'efficacité opérationnelle. Dans l'industrie, par exemple, les capteurs IoT peuvent surveiller et contrôler les machines, réduisant les temps d'arrêt et augmentant la productivité.[26]
- **Prise de décision basée sur des données** : Les appareils IoT collectent des données en temps réel, permettant aux entreprises de prendre des décisions plus précises. Cela conduit à une meilleure planification des ressources et à une compréhension approfondie des besoins des clients.[27]
- **Amélioration de la qualité de vie** : Dans le domaine de la domotique, l'IoT permet le contrôle à distance d'appareils et de systèmes domestiques tels que le chauffage, l'éclairage et les appareils électroménagers, offrant ainsi plus de confort et de sécurité aux utilisateurs.[27]
- **Suivi et gestion de la santé** : Les dispositifs tels que les montres connectées et les capteurs de santé permettent aux individus de surveiller en permanence leur santé, y compris l'activité physique, le sommeil et d'autres paramètres vitaux, contribuant ainsi à la prévention des problèmes de santé.[27]
- **Optimisation des ressources** : Dans des secteurs tels que l'apiculture et la gestion de l'énergie, l'IoT facilite la surveillance et le contrôle plus efficaces des ressources. Par exemple, les capteurs IoT dans l'apiculture peuvent optimiser la gestion des ruches en surveillant des paramètres tels que la température, l'humidité et le poids des ruches, permettant des décisions éclairées et réduisant les risques pour la colonie d'abeilles.[27]

1.4.6 Les limites de l'IoT

Les défis touchant à l'Internet des objets sont multiples, touchant divers domaines :

- **Sécurité et confidentialité des données** : La sécurité des données et la protection de la vie privée sont des préoccupations majeures dans l’IoT. Les informations collectées par les appareils connectés peuvent être vulnérables sans des mesures de sécurité rigoureuses, suscitant des inquiétudes quant à une utilisation non autorisée ou à un partage indésirable.[27][28]
- **Interopérabilité** : Les appareils IoT, provenant de divers fabricants ou fonctionnant sur des protocoles de communication différents, peuvent rencontrer des difficultés d’interopérabilité, ce qui gêne la compatibilité entre les dispositifs.[27][28]
- **Fiabilité et disponibilité** : La fiabilité des appareils IoT peut être mise à l’épreuve en raison de pannes techniques, d’interférences de signal et d’autres facteurs externes. De plus, la disponibilité des services IoT dépend souvent de la qualité de la connexion Internet, limitant ainsi leur utilisation dans des régions ou des environnements à connectivité réduite.[27][29]
- **Coût** : Le déploiement et l’entretien d’un écosystème IoT peuvent représenter un investissement financier significatif, surtout lorsqu’il s’agit de déployer de nombreux appareils sur une vaste étendue ou dans des environnements difficiles d’accès.[27]

1.4.7 Le système Arduino

Une équipe de développeurs, comprenant Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti, a mis en œuvre un projet appelé Arduino. Celui-ci consiste en une carte électronique programmable associée à un logiciel de programmation adapté aux caractéristiques de la carte.

L’objectif de ce projet est de faciliter aux utilisateurs la création des systèmes électroniques tout en leur permettant d’acquérir des compétences en programmation informatique.[30]

1.4.7.1 Les cartes Arduino

Ce sont des cartes électroniques programmables, équipées d’un microcontrôleur et de mémoire, sur lesquelles il est possible de connecter divers capteurs tels que la température, l’humidité, la vibration ou la lumière, ainsi que des périphériques tels qu’une caméra, des boutons, des potentiomètres de réglage et des contacts électriques. De plus, elles disposent de connecteurs permettant le branchement de LED, de moteurs, d’afficheurs, d’un écran, et bien d’autres composants encore.[31]

Plusieurs cartes Arduino ont été produites et vendues dans le commerce à ce jour, les plus connues sont citées dans le tableau 1.1.

Version de carte	Année de sortie	Micro-contrôleur
Diecimila	2007	ATmega168V
LilyPad	2007	ATmega168V / ATmega328V
Nano	2008	ATmega328 / ATmega168
Mini	2008	ATmega168
Mini Pro	2008	ATmega328
Duemilanove	2008	ATmega168 / ATmega328
Mega	2009	ATmega1280
Fio	2010	ATmega328P
Mega 2560	2010	ATmega2560
Uno	2010	ATmega328P
Ethernet	2011	ATmega328
Mega ADK	2011	ATmega2560
Leonardo	2012	ATmega32U4
Esplora	2012	ATmega32U4
Micro	2012	ATmega32U4
Yún	2013	ATmega32U4 + Linino

TABLE 1.1 – Les types d’Arduino les plus répandus

Les cartes Arduino Uno, Mega 2560, Mega ADK, Due, Nano et Yún sont parmi les modèles les plus populaires et utilisés de la gamme Arduino.

La carte Arduino UNO

La carte Arduino UNO, illustrée à la figure 1.8, est idéale pour les débutants souhaitant découvrir l’environnement Arduino, car elle permet de réaliser des projets avec facilité. Cependant, elle peut s’avérer limitée pour la gestion de nombreux périphériques ou le traitement d’algorithmes complexes.[32]

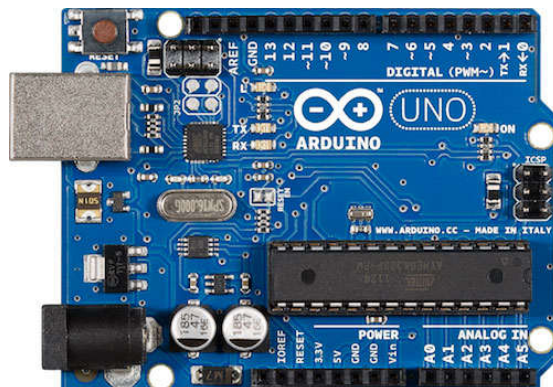


FIGURE 1.8 – La carte arduino Uno [33]

La carte Arduino Mega 2560

La carte Arduino Mega 2560, illustrée à la figure 1.9, est la plus populaire après l’Arduino Uno. Elle se distingue par son processeur plus puissant et sa mémoire étendue. Ces caractéristiques permettent le traitement d’algorithmes plus complexes et la gestion d’un plus grand nombre de périphériques.[34]

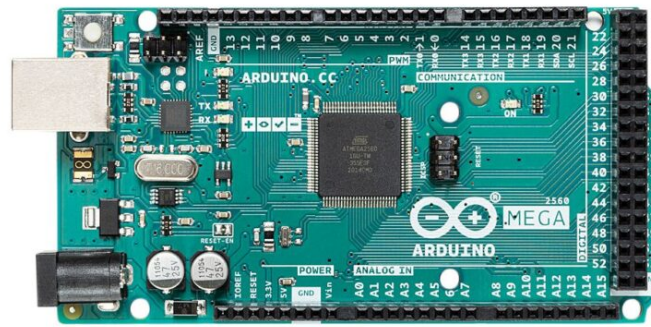


FIGURE 1.9 – La carte arduino Mega 2560
[34]

La carte Arduino Mega ADK

La carte Arduino Mega ADK, illustrée à la figure 1.10, est similaire à la carte Arduino Mega. Sa particularité réside dans un port USB spécifique permettant une connexion directe avec les appareils Android, ouvrant ainsi de nouvelles possibilités d'interaction avec les smartphones et leurs capteurs intégrés.[35]

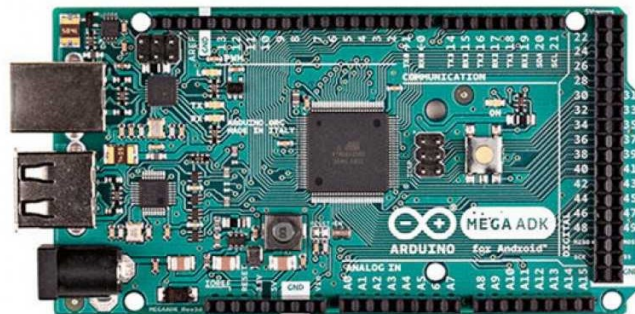


FIGURE 1.10 – La carte arduino Mega ADK
[35]

La carte Arduino Duemilanove

La carte Due offre des performances jusqu'à trois fois supérieures à celles des autres cartes. Cette puissance accrue la rend idéale pour la manipulation rapide d'algorithmes complexes, notamment dans le domaine de la robotique.[36] La figure 1.11 représente la carte Due.

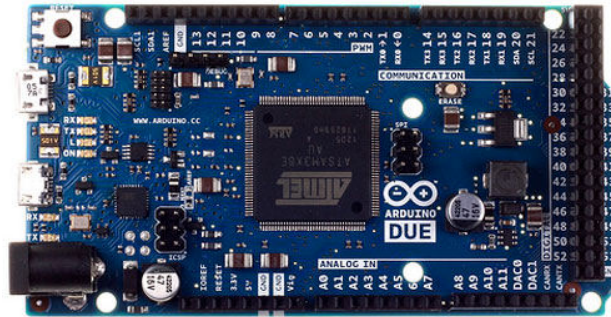


FIGURE 1.11 – La carte arduino Due
[37]

La carte Arduino Nano

La carte Arduino Nano 1.2 est une version compacte et légère de l'Arduino Duemilanove, offrant des fonctionnalités similaires. Grâce à sa petite taille, elle est largement utilisée dans des espaces restreints et divers domaines tels que la sécurité, l'environnement et la robotique.[38]

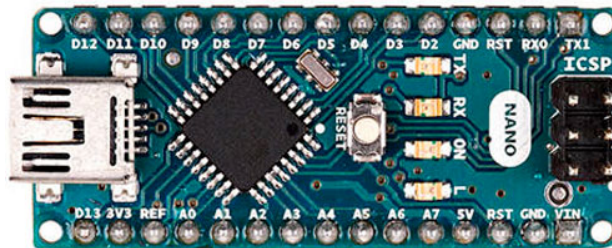


FIGURE 1.12 – La carte arduino Nano
[39]

La carte Arduino Yun

La carte Arduino Yun, illustrée à la figure 1.13, a été récemment introduite par Arduino. Elle est basée sur la conception de la carte Leonardo. La Yun combine la simplicité d'utilisation d'une carte Arduino avec la puissance de Linux, et elle est la première carte Arduino à intégrer nativement une connectivité Wi-Fi.[40]

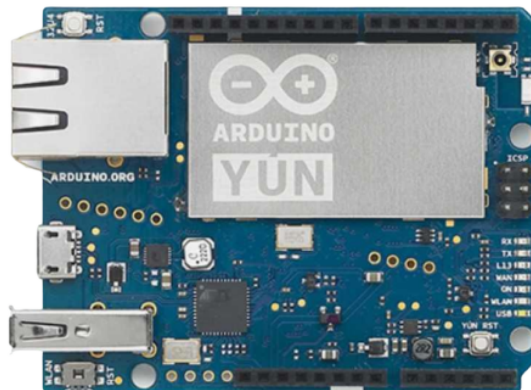


FIGURE 1.13 – La carte arduino Yun
[41]

1.4.7.2 L'IDE Arduino

L'IDE Arduino, ou Environnement de Développement Intégré Arduino, est une plateforme logicielle dédiée à la programmation des cartes Arduino. Cette interface, illustrée dans la figure 1.14, offre un ensemble d'outils facilitant la création, l'édition, le téléchargement et la gestion des codes sources pour les projets électroniques réalisés avec ces cartes. Elle est largement utilisée par les amateurs, les éducateurs et les professionnels pour prototyper rapidement leurs projets électroniques (les robots, les capteurs connectés, etc.) et créer des solutions personnalisées dans divers domaines.

L'IDE Arduino prend en charge plusieurs langages de programmation, principalement le langage de programmation Arduino, ainsi que le C et le C++.

L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation, comme présenté dans la figure 1.14.

Sa version actuelle est la 2.3.2, offrant une apparence pratiquement similaire sur toutes les plateformes, que ce soit Windows, Mac ou Linux.[42]

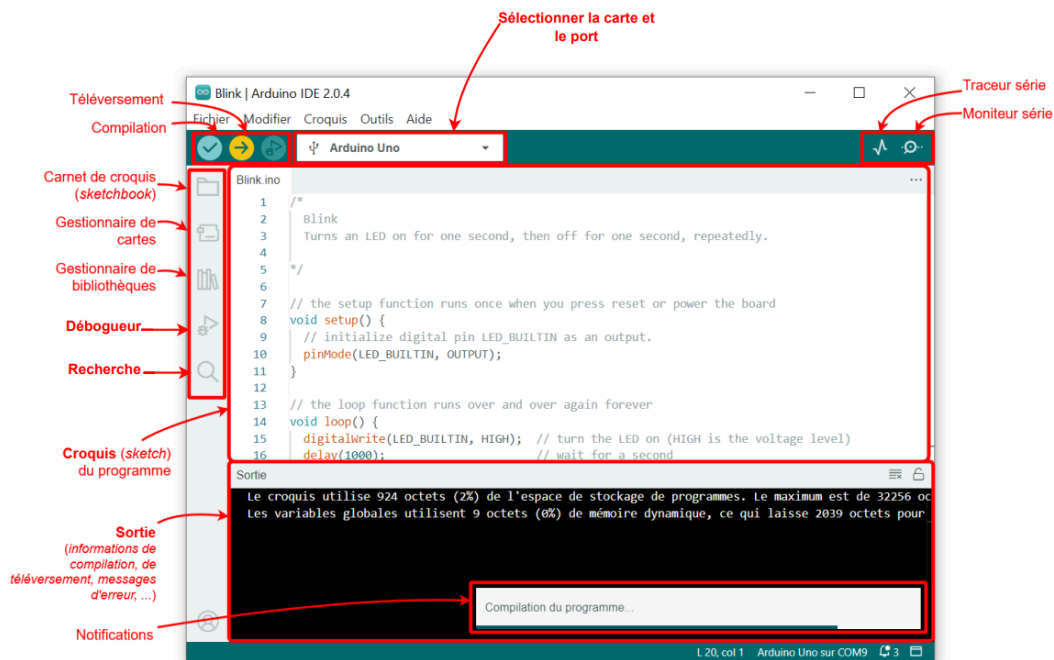


FIGURE 1.14 – Interface de l'IDE Arduino

[43]

Voici une brève explication sur les fonctionnalités de L'IDE montrées dans la figure 1.14 :

- **Compilation / Téléversement** : Compile et téléverse le code sur la carte Arduino connectée.
- **Débogueur** : Permet de tester et de déboguer des programmes en temps réel.
- **Recherche** : Rechercher des mots-clés dans le code.
- **Sélectionner la carte et le port** : Affiche automatiquement les cartes Arduino détectées et leur numéro de port.
- **Carnet de croquis** : Accède aux croquis stockés localement et permet la

- synchronisation avec le cloud Arduino.
- **Gestionnaire de cartes** : Permet de parcourir et d'installer des packages Arduino et tiers.
- **Gestionnaire de bibliothèques** : Donne accès à des milliers de bibliothèques Arduino créées par la communauté.
- **Le moniteur série** : Ouvre l'outil moniteur série dans un nouvel onglet de la console.
- **La fenêtre du croquis** : C'est l'espace principal pour écrire et éditer le code Arduino.
- **La fenêtre de sortie** : Affiche les messages de compilation et les résultats de l'exécution du code.
- **La barre de notification** : Fournit des informations sur l'état de l'IDE et du programme en cours d'exécution.[44]

Fonctions principales du langage Arduino :

- **La fonction setup()** est une fonction spéciale dans Arduino, appelée une seule fois au démarrage du programme. Elle est utilisée pour initialiser les variables, configurer les modes des broches, établir les communications série et d'autres paramètres nécessaires pour préparer le programme à exécuter sa boucle principale.
- **La fonction loop()** est la fonction principale dans un programme Arduino. Elle est appelée en boucle sans fin une fois que le programme démarre. C'est ici que le code principal du programme s'exécute de manière répétitive, permettant à l'Arduino de réagir en temps réel aux événements, de lire des capteurs, de contrôler des actionneurs, etc.[30]

1.4.7.3 Les avantages de l'Arduino

Le système Arduino offre plusieurs avantages significatifs :

- Les cartes Arduino sont une option moins chère et plus pratique comparées à d'autres microcontrôleurs.[31]
- Arduino convient à la fois aux débutants et aux experts grâce à la facilité d'utilisation de son environnement de développement intégré (IDE). De nombreux conseils, tutoriels et exemples en ligne sont disponibles pour faciliter la prise en main.[30][31]
- les cartes Arduino offrent une fiabilité reconnue, que ce soit pour la conception de projets électroniques simples ou complexes.[30]
- Le logiciel Arduino, développé en Java, est compatible avec les systèmes d'exploitation Windows, Macintosh et Linux.[30][31]
- Existence de " shield " (boucliers en français) qui sont des cartes supplémentaires qui se connectent sur le module Arduino pour augmenter les possibilités comme par exemple : afficheur graphique couleur, interface ethernet, GPS, etc.[30]

1.4.8 Comment l'Internet des objets peut améliorer la surveillance des ruches

- **Limiter les interventions humaines** : Le suivi à distance des ruches permet à l'apiculteur de réduire ses déplacements, surtout pour les ruches éloignées ou difficiles d'accès, comme celles en zone montagneuse. Même si des visites régulières restent nécessaires, l'utilisation d'outils connectés réduit le nombre d'interventions nécessaires. Cela signifie moins d'ouvertures de ruches pour évaluer leur production, leur santé et leur activité, et donc moins de risques de propagation de maladies entre les ruches.[1]
- **Anticiper l'avenir de la colonie** : L'utilisation de l'Internet des objets permet aux apiculteurs de faire des prévisions sur l'avenir de leurs colonies. Les données collectées fournissent des indications sur l'évolution des colonies tout au long de la saison active. Cela permet par exemple à l'apiculteur d'évaluer si une colonie est prête à survivre à l'hiver avec ses propres ressources. Par exemple, si une ruche montre des signes de faible poids, de réserves limitées, de population réduite d'abeilles ou de température non optimale en fin de saison, l'apiculteur peut envisager d'intervenir en ajoutant des réserves alimentaires et en renforçant la colonie pour l'hiver.[1]
- **Optimiser la santé des ruches** : Grâce à l'IoT, les apiculteurs peuvent surveiller de près la santé des ruches. Ils sont en mesure de détecter les anomalies ou les écarts par rapport aux conditions optimales, comme les variations brusques de température ou les modèles sonores inhabituels. Cette surveillance en temps réel alerte les apiculteurs sur les problèmes potentiels, leur permettant d'intervenir rapidement pour protéger la ruche et prévenir la propagation de maladies ou de parasites.[45]
- **Renforcer la sécurité des ruches** : L'IoT se révèle être une solution précieuse pour protéger les ruches contre les pertes d'exploitation causées par le vol. Grâce à cette technologie, l'apiculteur peut localiser instantanément toute ruche perdue ou volée, réduisant ainsi considérablement les risques et permettant une intervention rapide pour préserver la colonie.[46]

1.5 Définition des objectifs de l'étude

Après avoir étudié l'environnement des abeilles ainsi que l'impact potentiel de l'Internet des objets sur la surveillance des ruches, nos objectifs principaux sont :

- Concevoir un système de surveillance en temps réel basé sur l'IoT, chargé de collecter et de traiter des données sur les conditions environnementales et le comportement des abeilles.
- Fournir aux apiculteurs des informations précieuses pour une gestion proactive et efficace de leurs colonies sans nécessité de déplacement.
- Préserver la santé et la productivité des abeilles en utilisant les données collectées pour identifier rapidement les anomalies dans les ruches.

1.6 Conclusion

En conclusion, ce chapitre marque le début d'une exploration approfondie de la relation entre les abeilles et les technologies émergentes, notamment l'IoT. Face aux défis actuels tels que le changement climatique, la perte d'habitat et les maladies, il devient impératif d'adopter des approches novatrices pour comprendre et protéger ces pollinisateurs et améliorer la surveillance des ruches, ouvrant ainsi la voie à des interventions plus précises et ciblées visant à garantir la santé des colonies.

Le prochain chapitre sera consacré pour la présentation de l'organisme d'accueil dans lequel nous avons effectué notre stage, ainsi que notre expérience pratique en apiculture.

CHAPITRE 2

PRÉSENTATION DE L'ORGANISME D'ACCUEIL

2.1 Introduction

Avant d'engager tout processus d'amélioration ou de développement, il est essentiel d'avoir une compréhension approfondie de la situation actuelle de l'entreprise. Dans ce chapitre, nous allons présenter l'entreprise Api-ruche, où nous avons effectué notre stage dans le cadre de notre projet de fin d'études. Nous débuterons par la présentation de cette entreprise, suivie d'une analyse des problèmes rencontrés lors de notre expérience, qui seront les axes centraux de notre mémoire.

2.2 Présentation de l'entreprise Api-ruche

L'histoire d'Api-ruche commence en 1993, lorsque Monsieur Sadek Ouramedane, l'actuel directeur, ouvre son premier magasin et atelier à Tayma, dans la commune de Oued Ghir. Sa détermination et son engagement l'ont conduit à poursuivre son expansion. En 2000, en collaboration avec un partenaire, ils ouvrent un atelier et un magasin plus spacieux à Ibourassen, toujours dans la région de Oued Ghir. En 2004, une étape importante est franchie avec la création d'une coopérative apicole en partenariat avec l'État. Cette initiative permet à l'entreprise d'acquérir un matériel apicole plus performant et d'augmenter sa production.

En 2010, Monsieur Ouramedane décide de transformer Api-ruche en une entreprise familiale. Cette nouvelle orientation permet à l'entreprise de se concentrer sur ses valeurs fondamentales et de proposer des produits et services d'une qualité irréprochable.

Aujourd'hui, Api-ruche est une entreprise expérimentée dans le domaine de l'apiculture en Algérie. Forte de son savoir-faire, elle offre une large gamme de produits et services liés aux ruches d'abeilles. La figure 2.1 représente les site de l'entreprise.



FIGURE 2.1 – L'entreprise Api-ruche

2.2.1 Superficie

Le site de l'entreprise Api-ruche s'étend sur une superficie totale de 347 m², répartie comme suit :

- Magasin : 16 m²
- Atelier avec espace de stockage : 294 m²
- Chambre à désoperculer : 25 m²
- Chambre froide : 45 m³

Elle occupe également cinq champs d'une superficie totale de 1048 m², répartis comme suit :

- Champs Laazib à Tayma : 333 m², comprenant 30 ruches et 70 essaims.
- Ferme Pilot à Elkseur : 125 m², comprenant 28 ruches.
- Champs Sens Unique à Elkseur : 148 m², comprenant 30 ruches.
- Champs Amassin à Amizour : 280 m², comprenant 80 essaims.
- Champs Tayma : 162 m², comprenant 18 ruches et 25 essaims.

2.2.2 Localisation

L'entreprise Api-ruche est implantée à Iboulassen, dans la commune de Oued Ghir, située dans la wilaya de Béjaïa. Son emplacement est stratégique, à seulement 20 mètres de l'antenne communale d'Iboulassen et de la route nationale numéro 12,

comme représenté dans la figure 2.2, ce qui facilite l'accès pour ses clients.

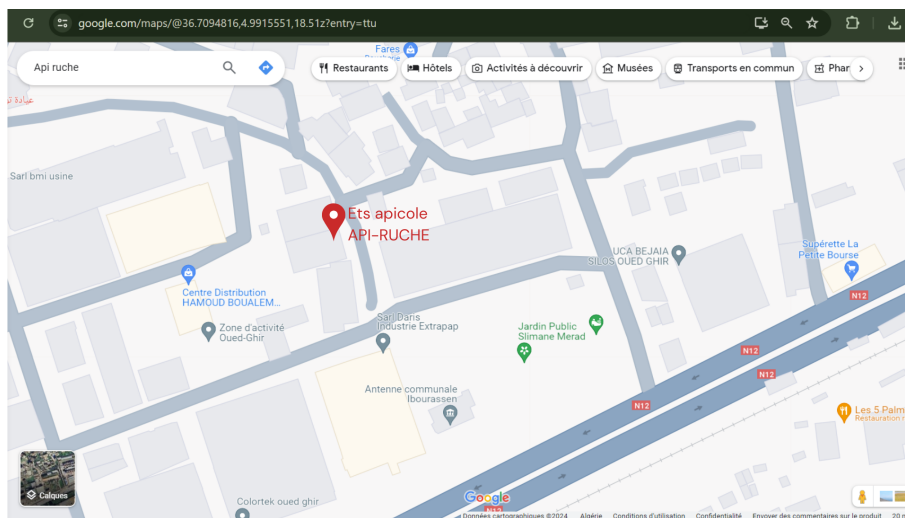


FIGURE 2.2 – L'emplacement de l'entreprise Api-ruche sur Google Maps

2.2.3 Capacité de production et effectifs

Api-ruche a une capacité de production moyenne de 1 000 ruches par an, ce qui équivaut à environ 10 000 cadres par an. Elle produit également en moyenne 7 quintaux de miel par an. Pour maintenir cette production, l'entreprise compte sur une équipe composée de :

- 15 agents dans l'atelier chargés de la transformation du bois en composants de ruches, ainsi que de la fabrication et de l'assemblage des cadres.
- 5 apiculteurs responsables de l'élevage des ruches, veillant à la santé et au bien-être des abeilles tout en supervisant la production de miel et de ses dérivés.
- Un représentant commercial gérant le magasin et assurant la responsabilité des ventes.
- Un comptable chargé de la gestion financière et comptable de l'entreprise.

2.2.4 Organigramme de l'entreprise

La figure 2.3 montre l'organigramme représentant la structure de l'entreprise.

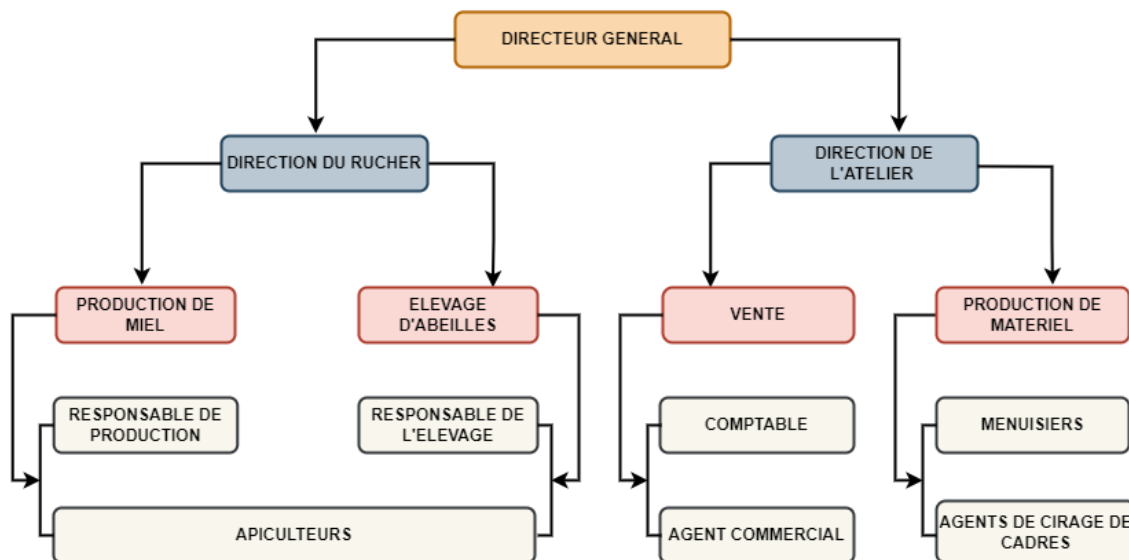


FIGURE 2.3 – L'organigramme représentant la structure de l'entreprise Api-ruche.

2.2.5 Équipements

Les principaux équipements de l'atelier comprennent :

- **Une dégauchisseuse-raboteuse** : Cet équipement est utilisé pour aplanir et surfer le bois, ce qui peut être utile dans la construction ou la réparation de ruches et d'autres structures apicoles.
- **Deux toupies-tenonneuses** : Les toupies-tenonneuses sont des machines polyvalentes utilisées pour façonner le bois, notamment pour la fabrication de cadres de ruches et d'autres éléments structurels.
- **Deux scies circulaires** : Les scies circulaires sont essentielles pour découper le bois avec précision, ce qui peut être nécessaire pour la construction et la réparation des équipements apicoles.
- **Une mortaiseuse** : Cet équipement est utilisé pour réaliser des mortaises, des encoches ou des cavités dans le bois, ce qui peut être nécessaire pour l'assemblage de certaines pièces des ruches ou d'autres équipements.
- Les équipements principaux au niveau de la chambre à désoperculer sont :
- **Trois extracteurs de miel de 4 cadres** : Ces extracteurs sont utilisés pour extraire le miel des cadres de ruche sans endommager les rayons de cire, ce qui permet de récolter le miel de manière efficace et propre.
- **Cinq maturateurs de 200 kilos** : Les maturateurs sont des réservoirs utilisés pour stocker et mûrir le miel après son extraction, aidant ainsi à améliorer sa saveur et sa texture avant la mise en bouteille, ils permettent également d'éliminer certaines impuretés.
- **Deux maturateurs de 100 kilos.**

2.3 Domaine d'activité de l'entreprise

Dans l'entreprise Api-ruche, on peut distinguer deux types d'activités : les activités principales et les activités secondaires.

2.3.1 Les activités principales

Les activités principales concernent essentiellement les opérations de production, la distribution et la commercialisation de tous types de produits apicoles. Le chiffre d'affaires d'Api-ruche est principalement constitué de la production vendue ainsi que de la vente de produits et de matériel apicoles. L'entreprise Api-ruche est spécialisée dans la production de matériel et de produits suivants :

- **Ruches** : Les éléments constitutifs des ruches, comprenant les hausses, les cadres, les couvre-cadres et les toits, représentent 36% de la production.
- **Essaims** : Des ruchettes contenant des colonies d'abeilles élevées, proposées à la vente par l'entreprise, représentent 24
- **Miel** : Une gamme variée de miels, incluant le miel multi-fleurs, le miel de jujubier, le miel de roquette, le miel d'eucalyptus, le miel d'abricot, etc., constitue 30%. La figure 2.4 représente un graphe illustrant l'évolution de la production de miel de 2019 à 2023.

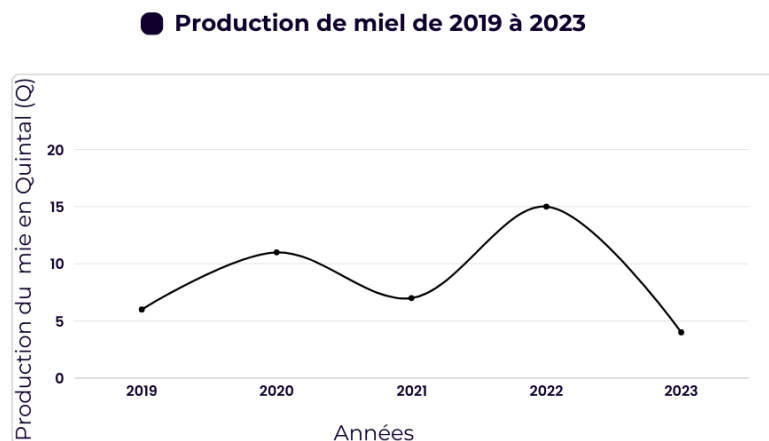


FIGURE 2.4 – L'évolution de la production de miel de 2019 à 2023 dans l'entreprise Api-ruche

L'entreprise se distingue également par la vente de divers produits et équipements apicoles, comprenant des extracteurs, des maturateurs, des enfumoirs, des grilles à reine, de la cire, des compléments alimentaires pour les abeilles, des combinaisons d'apiculture, etc. Ces articles représentent 10% de son activité.

2.3.1.1 Les fonctions spécifiques de l'entreprise

1. Gestion du rucher :

Élevage des abeilles :

- Suivi de la santé des colonies
- Sélection des reines
- Multiplication des colonies
- Production d'essaims

- Traitement des maladies
- Maintenance du rucher
- Production de miel :**
- Planification de la production
- Suivi des récoltes
- Extraction du miel

2. Gestion de production de matériel :

Production des ruches :

- Approvisionnement en matières premières
- Fabrication des éléments de la ruche (cadres, hausses, etc.)
- Réparation du matériel

Installation des cadres :

- Installation des cadres dans les ruches
- Cirage des cadres
- Gestion des stocks de cadres

3. Gestion de vente :

- Définition de la stratégie commerciale
- Prospection clients
- Négociation des contrats
- Suivi des commandes
- Vente directe aux apiculteurs

2.3.1.2 La transhumance des ruches

L'entreprise Api-ruche utilise la pratique de la transhumance des ruches pour produire une variété de miels provenant de différentes régions. La transhumance consiste à déplacer les ruches d'un endroit à un autre, souvent vers des zones spécifiques où les conditions environnementales sont propices à la production de miel de qualité. Dans ce cas, les ruches sont déplacées hors de la wilaya de Bejaia, généralement vers des régions comme Djelfa, Aghouat, etc.

Cette opération de transhumance peut prendre de 20 à 30 jours, selon la distance à parcourir et les conditions de déplacement.

2.3.2 Les activités secondaires

En plus de son activité principale, Api-ruche s'engage dans d'autres activités secondaires telles que :

- La production et la vente de pots de concassée de miel pur.
- L'organisation de formations pratiques en apiculture, ainsi qu'un suivi individualisé pour les apiculteurs.
- La participation aux événements et expositions apicoles.
- La production agricole et la vente des produits agricoles.

2.4 Tâches effectuées pendant notre stage

Pendant notre stage, nous avons eu l'opportunité de réaliser diverses tâches essentielles dans le domaine de l'apiculture :

1. Observation de la population d'une ruche :
 - Compréhension du fonctionnement interne.
 - Suivi de la taille de la colonie.
 - Détection d'éventuels problèmes (comme la disparition inattendue des reines et la présence de parasites tels que le varroa).
2. Marquage des reines :
 - Identification de la reine dans la colonie.
 - Pose d'une marque de couleur visible correspondant à son année de naissance pour faciliter son repérage.
3. Observation et compréhension du processus d'essaimage naturel pour la reproduction des colonies d'abeilles.
4. Apprentissage des techniques d'essaimage artificiel pour contrôler et gérer la croissance des colonies : Mise en pratique de l'essaimage artificiel sous la supervision d'un apiculteur expérimenté.
5. Préparation et distribution de sirop nourrissant pour l'alimentation des abeilles.
6. Apprentissage des techniques de cirage des cadres et fabrication de cadres de ruche cirés prêts à l'emploi.
7. Parallèlement à ces activités, nous avons observé de près les différents produits de la ruche tels que le miel, le pollen et la gelée royale, ainsi que le matériel utilisé dans les ruches, notamment les cadres, les hausses et les grilles à reine.

2.5 Problématique

Après avoir effectué notre période de stage au sein de l'entreprise Api-ruche, plusieurs problèmes et difficultés majeures liés à l'élevage des abeilles, à la gestion des colonies et à leur environnement ont été identifiés :

- L'entreprise subit quotidiennement des pertes de ruches, principalement en raison des changements climatiques, ainsi que des maladies et autres affections, souvent à cause d'une intervention tardive.
- L'ouverture régulière des ruches pour surveiller leur activité et leur santé, bien que nécessaire, comporte un risque de transmission de maladies entre les colonies, que ce soit par les abeilles contaminées ou par le matériel apicole déjà utilisé dans des ruches infectées.
- La carence en main-d'œuvre qualifiée représente un défi majeur pour l'entreprise, ajoutant une charge supplémentaire aux quelques apiculteurs déjà en poste. Cette situation est accentuée par la nécessité de déplacements réguliers vers les différents ruchers, complexifiant ainsi la gestion et la surveillance des colonies d'abeilles.
- L'absence de sécurité dans les ruchers expose les apiculteurs au risque de vol et de prise illégale de leurs ruches. Cette situation peut entraîner des pertes financières importantes.

- L'absence d'outils de surveillance avancés et fiables dans l'entreprise entraîne un manque de précision dans les mesures des paramètres de la ruche comme la température, l'humidité et le poids.

2.6 Conclusion

Dans ce deuxième chapitre portant sur la présentation de l'entreprise Api-ruche, nous avons détaillé les composants et le fonctionnement de cet organisme d'accueil, et effectué une analyse qui a abouti à la formulation de notre problématique.

Le prochain chapitre se concentrera sur la proposition d'une solution envisageable et efficace pour pallier une partie de ces problèmes.

CHAPITRE 3

SOLUTION PROPOSÉE

3.1 Introduction

Dans ce chapitre, nous présenterons la solution que nous proposons pour répondre aux problèmes identifiés dans le chapitre précédent. Nous détaillerons les composants et le fonctionnement de notre solution, ainsi que ses avantages potentiels pour l'entreprise et les apiculteurs.

3.2 Motivation

Après avoir identifié les problèmes et les besoins au sein de l'entreprise d'accueil de stage, nous avons constaté un manque d'outils et de solutions modernes pour surveiller et contrôler les ruches de manière efficace. C'est pourquoi nous avons choisi une approche innovante basée sur l'IoT pour y répondre.

Notre solution consiste à développer un système de surveillance à distance d'une ruche d'abeilles, équipé de capteurs pour surveiller en temps réel différents paramètres essentiels tels que la température, l'humidité, le poids, le bruit, etc. Les données de ces paramètres seront envoyées à une plateforme en ligne sécurisée accessible aux apiculteurs, qui pourront intervenir rapidement en cas de problème.

3.3 Le module ESP8266

L'ESP8266 est une famille populaire de microcontrôleurs spécialement conçus pour les applications de l'Internet des objets (IoT). Ces microcontrôleurs sont réputés pour leur prix abordable, leur faible consommation d'énergie et leur architecture permettant une connectivité Wi-Fi 802.11 b/g/n complète, incluant le protocole TCP/IP, le chiffrement et la gestion du réseau. Parmi les cartes ESP8266 populaires, on retrouve NodeMCU, WeMos D1 R2, LOLIN, Adafruit Feather HUZZAH

ESP8266, etc.[47]

3.3.1 La carte Wemos D1 R2

La carte "WeMos D1 R2", illustrée dans la figure 3.1, est un microcontrôleur basé sur l'ESP8266 qui offre une connectivité Wi-Fi intégrée et adopte le format de l'Arduino Uno. Elle est compatible matériellement avec les shields et les modules de cette carte, et elle est également prise en charge par l'IDE Arduino, facilitant ainsi son utilisation pour les projets IoT.[47]

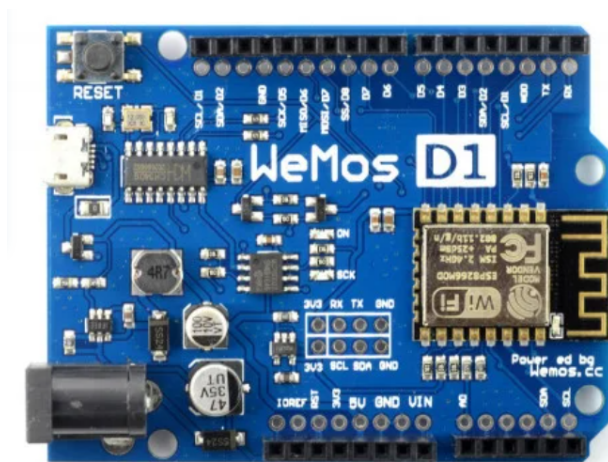


FIGURE 3.1 – La carte WeMos D1 R2

3.3.2 Les composants de la carte WeMos

Les principaux composants de la carte WeMos D1 R2 sont représentés dans la figure 3.2.

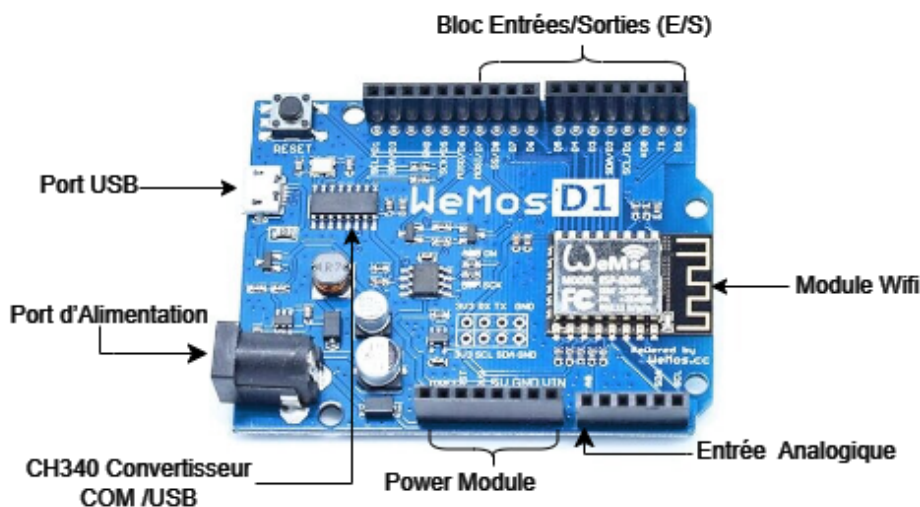


FIGURE 3.2 – Les principaux composants d'une carte WeMos D1 R2

Le tableau 3.1 présente la description des principaux composants de la carte WeMos D1 R2.[47]

Composant	Spécifications
Bloc d'entrées/sorties (E/S)	La carte dispose de 11 broches d'entrées/sorties numériques (D0 à D8, TX, RX). Ces broches permettent de connecter divers capteurs et actionneurs.
Entrées analogiques	La carte est équipée d'une unique entrée analogique (A0) destinée à la mesure de signaux analogiques.
Module Wi-Fi	Un module Wi-Fi ESP8266 qui permet de connecter le système à un réseau sans fil.
Port USB	Il permet d'alimenter la carte et de la programmer depuis un ordinateur.
Port d'alimentation	Pour alimenter des composants externes.
CH340 convertisseur COM/USB	CH340 est conçu pour convertir un signal USB en un signal série de 3,3 V ou 5 V. Son rôle est de permettre la communication entre un ordinateur et la carte WeMos.
Power module	Ce module gère l'alimentation de la carte. Elle peut être alimentée en 5V via le port micro-USB ou directement en 3,3V.

TABLE 3.1 – Description des principaux composants de la carte WeMos D1 R2

3.3.3 Pourquoi choisir la WeMos D1 R2 ?

- La carte WeMos D1 R2 est équipée d'un module Wi-Fi intégré, ce qui permet aux développeurs de créer des projets qui impliquent la connexion à Internet sans avoir à utiliser des modules Wi-Fi supplémentaires.
- Compatible avec l'IDE Arduino, cette carte peut être programmée et bénéficier de la vaste communauté Arduino.
- La disposition des broches dans la carte WeMos D1 R2 offre une bonne flexibilité facilitant ainsi le prototypage, l'expérimentation et la création de divers projets électroniques.
- la WeMos D1 R2 est l'une des cartes les moins chères du marché, son prix abordable en fait un choix populaire pour de nombreux projets IoT.[47]

3.4 Les capteurs

Un capteur est un dispositif électronique capable de mesurer et de collecter des données à partir de son environnement et de les convertir en affichage lisible, ou de les transmettre en temps réel via le réseau à une plateforme pour un traitement immédiat ou ultérieur. Il peut détecter des données physiques telles que la température, la pression, la luminosité, ou bien chimiques telles que le CO₂, l'acide et bien plus encore.[48][49]

La figure 3.3 présente le fonctionnement d'un capteur.



FIGURE 3.3 – Le fonctionnement d'un capteur

Voici quelques exemples de capteurs :

- Le capteur sismique, ou sismomètre, détecte, amplifie et enregistre les petits mouvements du sol, utile pour les tremblements de terre et les éruptions volcaniques.
- Le capteur de lumière est un dispositif qui convertit énergie lumineuse en un signal électrique. Principalement utilisé dans l'éclairage automatique et la domotique.
- Le capteur de pression mesure la force externe sur une surface et convertit cette pression en un signal électrique, principalement via une jauge de déformation. Il est utilisé pour surveiller la pression des liquides, de l'air et des gaz.[50]

3.5 Les actionneurs

Un actionneur est un dispositif qui convertit l'énergie qu'il reçoit en une action mécanique ou un changement d'état physique.[51]

La figure 3.4 présente le fonctionnement d'un capteur.



FIGURE 3.4 – Le fonctionnement d'un actionneur

Voici quelques exemples d'actionneurs :

- La LED est un voyant lumineux activé par un courant électrique, utilisé pour indiquer l'état de fonctionnement d'une machine, généralement par des couleurs telles que le vert ou le rouge.[51]
- Le buzzer est un actionneur qui transforme un courant électrique en un son audible. Il est couramment utilisé pour générer une alarme sonore, notamment dans les détecteurs de fumée.[51]
- La résistance chauffante est un dispositif qui génère de la chaleur à partir d'un courant électrique, pour des usages variés comme les radiateurs électriques et

les chauffe-eau.[52]

3.6 L'ensemble des composants de notre système

En complément de la carte WeMos D1 R2 mentionnée précédemment, pour la mise en œuvre de notre système de surveillance à distance des ruches d'abeilles, nous avons choisi les éléments matériels suivant :

3.6.1 Les capteurs utilisés

Nous avons sélectionné nos capteurs en fonction des différents paramètres à surveiller pour assurer la santé et le bien-être des ruches incluant la température, l'humidité, le poids, le bruit ; ainsi que pour répondre aux besoins spécifiques de l'entreprise, en se basant sur les problèmes déjà détectés, nous avons décidé d'intégrer des fonctionnalités telles qu'un système de géolocalisation (GPS), un capteur de distance à ultrasons et un capteur de flammes.

3.6.1.1 Le capteur de température et l'humidité DHT22

Description

Pour surveiller la température et l'humidité à l'intérieur des ruches, nous avons opté pour le capteur DHT22, illustré dans la figure 3.5. Ce capteur mesure l'humidité à l'aide d'un capteur capacitif qui détecte les variations de capacité électrique dues aux changements de l'humidité de l'air. Pour la température, il utilise une thermistance, un type de résistance dont la valeur change en fonction de la température.[53] Le modèle dispose de trois broches :

- GND : Broche de masse (0V).
- OUT : Broche de sortie de données.
- VCC : Broche d'alimentation positive, typiquement entre 3.3V et 5.5V.

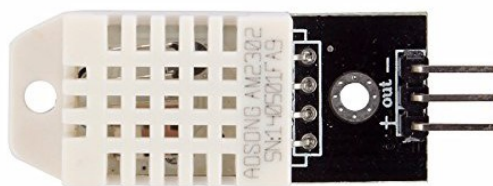


FIGURE 3.5 – Le capteur DHT22
[54]

Spécifications

- Tension d'alimentation : 3V à 5V.
- Courant de consommation : 2.5mA max.
- Plage de mesure :
 - température : -40 à +80 °C.
 - humidité : 0 à 100 % RH.

- Précision :
température : $\pm 0,5$ °C.
humidité : ± 2 à 5% RH.
- Période de mesure : une conversion toutes les 2 secondes.
- Dimensions : 25.1 x 15.1 x 7.7 mm.[55]

3.6.1.2 Le capteur de poids "Jauges de contraintes et amplificateur HX711"

Ce capteur est composé de deux composants complémentaires :

1. Jauges de contraintes (cellules de charge) :

Description

Les jauges de contrainte, montrées dans la figure 3.6, sont des résistances électriques qui mesurent des forces ou des poids en utilisant des éléments métalliques déformables, appelés éléments de flexion, sur lesquels elles sont collées. Sous l'effet d'une force, l'élément de flexion se déforme, entraînant une variation de la résistance des jauges. Ces variations sont converties en un signal électrique mesurable proportionnel à la force appliquée, puis est traité pour fournir une mesure précise. Dans notre cas, nous avons utilisé 4 cellules de charge, chacune capable de mesurer jusqu'à 50 kg, totalisant 200 kg pour le système.[56][57]



FIGURE 3.6 – Une jauge de contrainte de 50 KG
[58]

Spécifications

- Capacité : 50 KG.
- Sensibilité de sortie (mV/V) : $1.0 \pm 0.15\%$.
- Répétabilité (% de la pleine échelle) : 0.05%.
- Sortie de zéro (mV/V) : 0.3.
- Résistance d'entrée : 1000 ± 10 ohm.
- Résistance de sortie : 1000 ± 10 ohm.

- Méthode de connexion des fils : Rouge = Signal ; Blanc = Excitation ; Noir = Excitation.
 - Dimensions : 34mm x 34mm x 8mm.[59]
- Les résistances au niveau des cellules de charge sont visibles dans la figure 3.7.

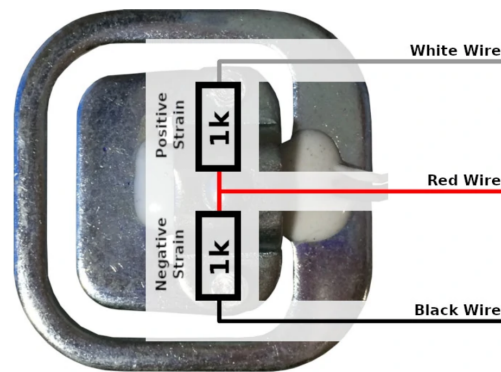


FIGURE 3.7 – La présentation des résistances au niveau des cellules de charge [60]

2. L'amplificateur HX711 (ADC) :

Description

En compagnie des 4 cellules de charge, nous avons utilisé le HX711, un convertisseur analogique-numérique 24 bits (ADC), illustré dans la figure 3.8. Il permet de convertir avec précision de petits signaux analogiques en données numériques. Il possède quatre broches principales :

- GND : Broche de masse (0V).
- DT : Broche de sortie de données.
- SCK : Broche d'horloge pour la communication avec le microcontrôleur.
- VCC : Broche d'alimentation positive, typiquement entre 2.6V et 5.5V.
- E+ (Excitation +) : Broche positive pour l'excitation de la cellule de charge.
- E- (Excitation -) : Broche négative pour l'excitation de la cellule de charge.
- A+ et A- : Entrées différentielles pour le canal A (principalement utilisé pour les mesures).
- B+ et B- : Entrées différentielles pour le canal B (secondaire).[61]

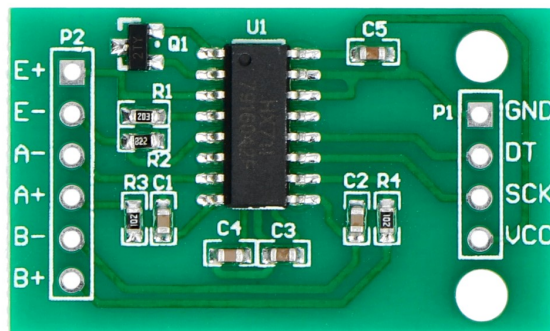


FIGURE 3.8 – L'amplificateur HX711

Spécifications

- Tension d'alimentation : 2.6V à 5.5V.
- Courant de consommation : inférieur à 1,5 mA.
- Canaux : Deux canaux d'entrée différenciés.
- Dimensions : 2,4 cm x 1,6 cm x 0,3 cm.[61]

3.6.1.3 Le capteur de son (bruit) KY-038

Description

En ce qui concerne la surveillance du bruit, nous avons opté pour le module KY-038, montré dans la figure 3.9. Il est équipé d'un microphone à condensateur électrique et utilise la puce principale LM393 (couramment utilisée dans les applications de détection et de comparaison de signaux). Ce module génère un signal de sortie à canal unique avec un niveau bas, idéal pour détecter les sons des abeilles.[62][63]

Le module est composé de quatre broches :

- GND : Broche de masse (0V).
- VCC : Broche d'alimentation positive, typiquement entre 3.3V et 5V.
- AO : Sortie de signal analogique.
- DO : Sortie de signal numérique.

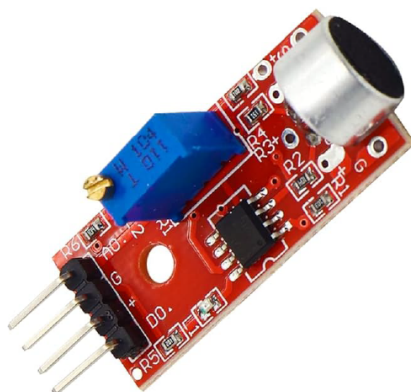


FIGURE 3.9 – Le capteur de son KY-038

Spécifications

- Tension d'alimentation : 3.3V à 5V.
- Courant de consommation : 5mA (faible consommation).
- Fréquence de réponse : 100Hz à 10KHz.
- Sensibilité : 46 ± 2.0 dB.
- Dimensions : 44mm x 15mm x 10 mm.[62]

3.6.1.4 Le capteur de distance à ultrasons HC-SR04

Description

Pour plus de sécurité, nous avons choisi d'utiliser un capteur de distance ultrasonique HC-SR04 (figure 3.10), afin de surveiller l'environnement des abeilles autour de la ruche et de détecter tout mouvement à proximité. Il est conçu pour mesu-

rer avec précision la distance entre le capteur et un objet en utilisant des ondes ultrasonores. Ce capteur dispose de quatre broches :

- VCC : Broche d'alimentation positive, typiquement 5V.
- GND : Broche de masse (0V).
- TRIG : Déclenche la mesure de distance par un signal bref.
- ECHO : Reçoit et mesure le temps écoulé depuis l'émission pour calculer la distance.[64]



FIGURE 3.10 – Le capteur de distance ultrasonique HC-SR04 [65]

Spécifications

- Tension d'alimentation : 5V.
- Courant de fonctionnement : 15 mA.
- Angle effectif : inférieur à 15 degrés.
- Angle de mesure : 30 degrés.
- Distance de portée : 2 cm à 400 cm.
- Résolution : 0,3 cm.
- Dimensions : 40 mm x 20 mm x 15 mm.[64]

3.6.1.5 Le module capteur antivol SIM808 GSM/GPRS/GPS

Description

Pour suivre la localisation géographique des ruches pendant la transhumance et connaître leurs emplacements en cas de vol, nous avons utilisé le module SIM808 multifonction qui combine les fonctions GSM, GPRS et GPS, représenté dans la figure 3.11.

Le package comprend un module SIM808, une antenne GSM et une antenne GPS. Ce capteur est basé sur le dernier module GSM/GPS SIM808 de SIMCOM, qui prend en charge les réseaux GSM/GPRS Quadri-Bande et intègre la technologie GPS pour la navigation par satellite.

Il est doté d'un circuit de charge pour les batteries lithium, offre une faible consommation d'énergie en mode veille, une grande sensibilité GPS, et peut être contrôlé via des commandes AT.[66][67]

Le module SIM808 comprend les broches suivantes :

- VCC : Alimentation positive (typiquement entre 3.4V et 4.4V).
- GND : Masse (0V).

- TXD : Broche de transmission de données (pour la communication série).
- RXD : Broche de réception de données (pour la communication série).
- RESET : Broche de réinitialisation.
- RST : Broche de réinitialisation matérielle.
- SIM CARD : Emplacement pour la carte SIM.

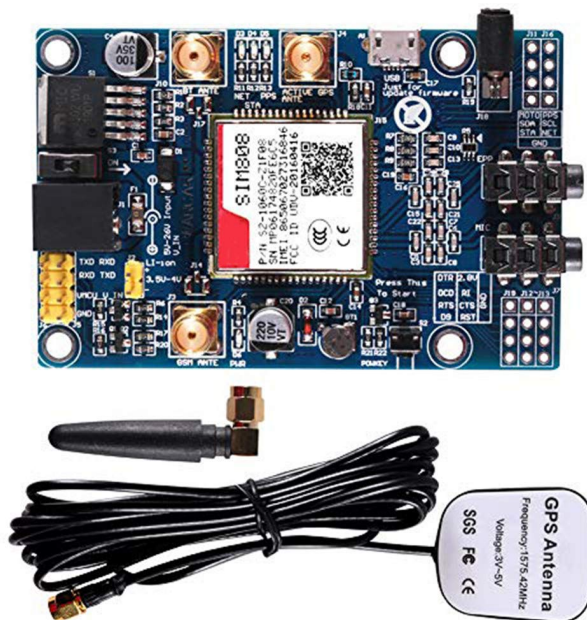


FIGURE 3.11 – Le module SIM808GSM/GPRS/GPS

Spécifications

- Tension d'alimentation : 3.4V à 4.4V.
- Courant de consommation : 10 mA.
- Quadri-bande : 850/900/1800/1900 MHz.
- Dimensions de la carte : 53 mm x 48 mm x 1,6 mm.[68]

3.6.1.6 Le capteur de flammes YG1006

Description

Pour détecter les feux à proximité des ruches, nous avons choisi d'utiliser le module capteur de flammes YG1006 à 3 broches (figure 3.12).

Il est composé d'un récepteur IR YG1006, d'une résistance, d'un condensateur, d'un potentiomètre et d'un comparateur (LM393) intégrés dans un circuit. Il détecte la lumière infrarouge dans une plage de longueurs d'onde de 760 nm à 1100 nm. La sonde de flamme infrarouge convertit la lumière détectée en variations de courant. Ces trois broches sont :

- GND : Masse (0V).
- OUT : Broche de sortie de signal numérique.
- VCC : Alimentation positive (typiquement 5V).[69]

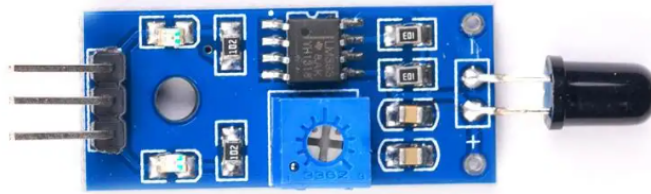


FIGURE 3.12 – Le module capteur de flammes YG1006 [70]

Spécifications

- Tension d'alimentation : 3.3V à 5V.
- Courant de fonctionnement : 15 mA.
- Angle de détection : De 0 à 60 degrés.
- Dimensions : 32mm x 14mm.[69]

3.6.2 Le matériel utilisé

3.6.2.1 Platine d'Expérimentation (breadboard)

Les breadboards sont des plaques d'essai qui facilitent la création rapide de circuits électroniques sans nécessiter de soudure.[30][71]

Dans ce projet, nous avons utilisé une breadboard de 1660 points (figure 3.13).

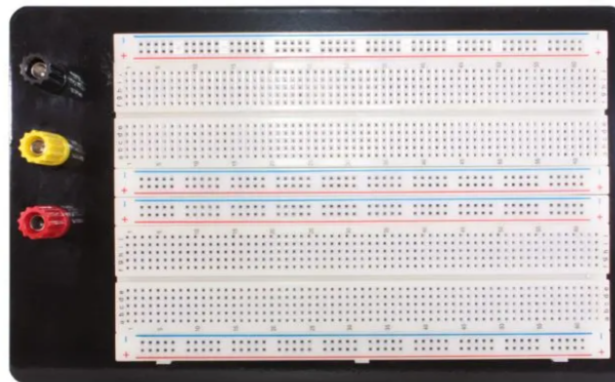


FIGURE 3.13 – La plaque d'expérimentation de 1660 points [72]

3.6.2.2 Câbles de connexion (jumper wires)

Un câble de connexion est un fil électrique muni d'un connecteur ou d'une broche à chaque extrémité. Il est utilisé pour relier les composants d'une breadboard ou d'un autre circuit de prototype.[30]

Dans notre cas, nous avons utilisé deux types de câbles : des câbles mâle-mâle et des câbles mâle-femelle, représentés dans la figure 3.14.



FIGURE 3.14 – La différence entre les câbles mâle-mâle et les câbles mâle-femelle

3.6.2.3 Écran LCD

Description

L'écran LCD, montré dans la figure 3.15, est un afficheur à cristaux liquides qui utilise la polarisation de la lumière pour afficher du texte. Nous intégrons cet écran dans notre système pour visualiser les données de température, d'humidité, de son et de poids. [30]

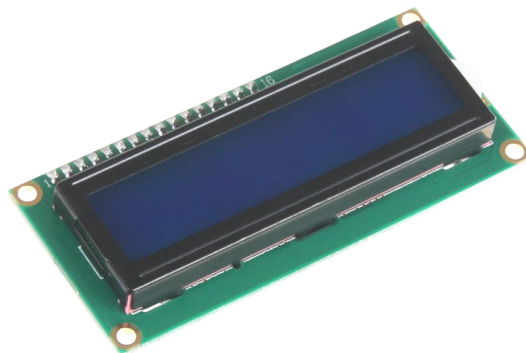


FIGURE 3.15 – L'écran LCD
[73]

Il peut être connecté à la carte WeMos D1 R2 via un adaptateur I2C, illustré dans la figure 3.16. L'adaptateur I2C possède 4 broches :

- GND : Masse (0V).
- VCC : Alimentation positive (typiquement 5V).
- SDA : Données sérielles .
- SCL : Horloge sérielle .

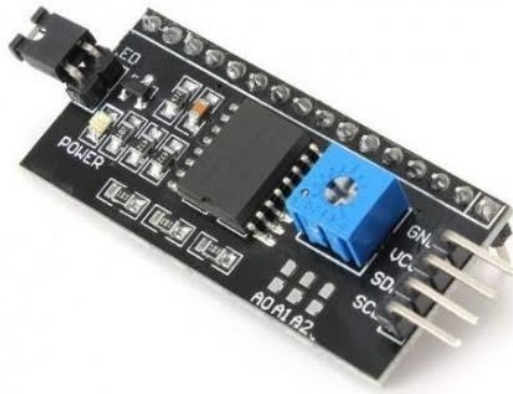


FIGURE 3.16 – L’adaptateur I2C
[74]

Spécifications

- Nombre de caractères : 2 lignes x 16 caractères.
- Tension : 5V .
- Taille du module : 80 mm x 35 mm x 11 mm.
- Taille de la zone d’affichage : 64,5 mm x 16 mm.[30]

3.6.2.4 Batterie rechargeable LiPo 2600mAh 11.1V

Pour alimenter le système, nous avons opté pour la batterie LiPo HJ POWER 2600mAh 25C 11.1V 3S (figure 3.17). Elle se distingue par sa haute densité énergétique et sa capacité de décharge rapide. Avec une tension nominale de 11.1V, elle est constituée de trois cellules en série, chacune fournissant 3.7V. Sa capacité de 2600mAh permet un courant constant de 2.6A pendant une heure, tandis que son taux de décharge de 25C lui permet de fournir un courant de pointe jusqu’à 65A, soit 25 fois sa capacité.



FIGURE 3.17 – La batterie rechargeable LiPo 2600mAh 11.1V

3.7 La plateforme Blynk

3.7.1 Qu'est ce que Blynk ?

Blynk se distingue comme une plateforme logicielle IoT complète et à faible code, fondée en 2014 par Pavel Yanenko, Dmitriy Dumanskiy et Alex Vasilyev. Elle permet de configurer une application mobile (Android et iOS) pour contrôler et visualiser les données d'un système embarqué via un serveur cloud public ou privé.

Elle est conçue pour répondre aux besoins des entrepreneurs, des ingénieurs et des grandes entreprises. Sa force réside dans sa simplicité d'utilisation et sa polyvalence, permettant à quiconque de créer des applications IoT, de gérer des appareils, des données et des clients dans un environnement cloud sécurisé.[75]

3.7.2 Composants de la plateforme Blynk

La plateforme Blynk se compose de trois éléments principaux :

1. **L'application Blynk** : Permet de créer des interfaces pour les projets à l'aide de différents widgets fournis.[76][77]

2. **Le serveur Blynk** : La gestion des communications entre le smartphone et le matériel est assurée par le serveur Blynk, qui peut être hébergé sur le Cloud Blynk ou configuré localement. Ce serveur est open-source et peut prendre en charge de nombreux périphériques.[76]

3. **La bibliothèque Blynk** : Les bibliothèques Blynk permettent aux dispositifs matériels (comme Arduino, Raspberry Pi, etc.) de se connecter au serveur Blynk pour envoyer et recevoir des commandes, simplifiant ainsi l'échange de données entre le matériel et la plateforme.[75][76]

3.7.3 Comment fonctionne Blynk ?

L'application Blynk communique avec un serveur Blynk, qui peut être public (blynk-cloud.com) ou privé. Ce serveur interagit ensuite avec le système embarqué via différentes technologies de communication telles que Ethernet, Wi-Fi, GSM, 2G, 3G, LTE, etc., en utilisant la bibliothèque Blynk fournie. Les technologies supportées pour cette communication incluent C, C++, JS, Python, et HTTP.[76]

La figure 3.18 montre le fonctionnement de Blynk.

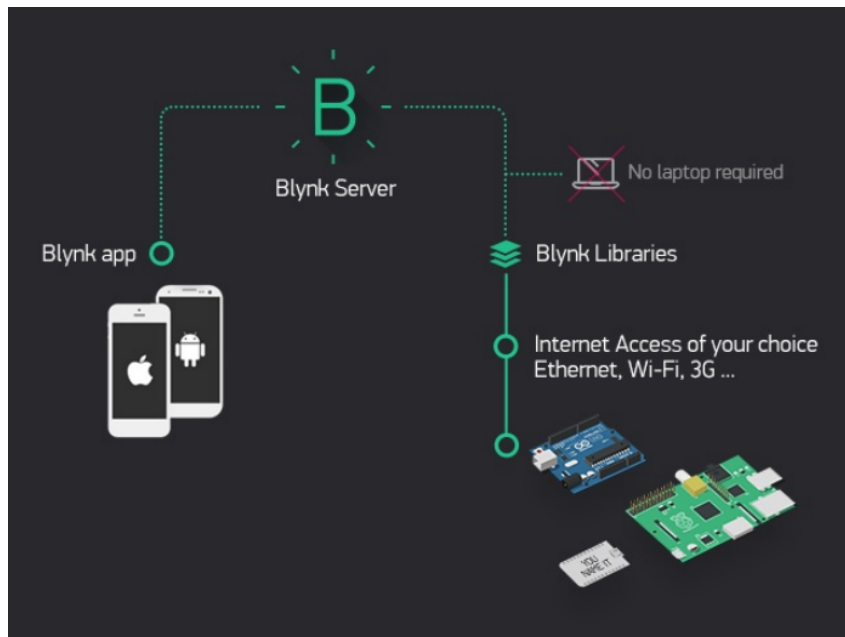


FIGURE 3.18 – Le fonctionnement de Blynk
[78]

3.7.4 Exemples d'utilisation de l'application Blynk

Voici quelques exemples d'utilisation de Blynk :

- Contrôler à distance les lumières, thermostats, serrures intelligentes et autres appareils domestiques depuis un smartphone.
- Envoyer des notifications push sur le smartphone de l'utilisateur en cas d'événements particuliers comme un seuil dépassé ou une anomalie détectée par les capteurs.
- Afficher en temps réel des données captées par des capteurs connectés, comme la température, l'humidité, le poids, etc.
- Programmer des actions à déclencher à certaines heures ou en réponse à des événements, comme l'arrosage automatique d'un jardin.[76]

3.7.5 Avantages de Blynk

Blynk offre plusieurs avantages par rapport à d'autres applications IoT :

- Blynk permet de créer des interfaces personnalisées sans avoir à coder, afin de contrôler des appareils électroniques, afficher des données en temps réel, automatiser des tâches et recevoir des notifications push.[76]
- La compatibilité étendue de Blynk avec une variété de matériels tels qu'Arduino, ESP8266, ESP32, Raspberry Pi et Particle offre une polyvalence pour connecter et gérer divers appareils.[76]
- La possibilité offerte par Blynk de configurer leur propre serveur privé assure aux utilisateurs un contrôle total sur la sécurité et la confidentialité des données.[75]

- La communauté active de Blynk partage des projets et propose diverses ressources en ligne, tutoriels et forums pour aider les utilisateurs à comprendre et à utiliser la plateforme de manière efficace.[77]

Malgré ses avantages, Blynk présente certains inconvénients tels que l'obligation de s'abonner à un forfait mensuel pour certaines fonctionnalités avancées ou pour avoir un grand nombre d'appareils connectés, ainsi que des problèmes de compatibilité avec les anciens programmes et matériels. De plus, la dépendance au serveur cloud peut entraîner des difficultés de disponibilité ou de latence dans certaines situations.[75]

3.8 Le principe de fonctionnement de la solution

Voici comment fonctionne notre solution de surveillance à distance des ruches :

1. **Collecte des données de capteurs :** Le système surveille en permanence la ruche à l'aide de divers capteurs. Ces capteurs, notamment de température, d'humidité, de poids, de bruit, de flammes, de positionnement géographique (GPS) et de distance ultrasonique, sont stratégiquement positionnés à l'intérieur et autour de la ruche. Ils sont connectés à une carte WeMos D1 R2, alimentée par une batterie rechargeable et programmée pour collecter les données en temps réel. La carte WeMos D1 R2 sert de centre de traitement et de communication pour le système, en collectant et en préparant les données pour la transmission.
2. **Transmission et traitement des données collectées :** Une fois collectées, les données des capteurs sont transmises instantanément vers un serveur cloud (Blynk cloud) via le réseau Wi-Fi pour traitement et stockage. En parallèle, le module SIM808 (GPS), qui fait partie intégrante du système, est prêt à fournir des données de localisation via le réseau cellulaire lorsque cela est nécessaire.
3. **Visualisation des données collectées :** Les données traitées, à l'exception de la localisation, sont accessibles aux apiculteurs via l'application Blynk, leur permettant de visualiser en temps réel les conditions de leurs ruches à travers des graphiques et des tableaux de bord. Cela offre aux apiculteurs une surveillance continue de l'état de leurs colonies.
4. **Accès aux données de localisation :** Pour obtenir la localisation de la ruche, l'apiculteur doit envoyer un SMS au module SIM808 qui répondra par un SMS contenant un lien Google Maps indiquant la position géographique de la ruche. Cela permet à l'apiculteur de vérifier la position de la ruche en temps réel en cas de besoin.
5. **Alertes et notifications :** Le système est conçu pour détecter automatiquement toute variation anormale des paramètres surveillés grâce à l'utilisation de seuils prédéfinis. En cas de détection d'une anomalie, l'apiculteur est immédiatement alerté par e-mail ou notification push via l'application Blynk, lui permettant ainsi d'intervenir rapidement pour prévenir tout problème potentiel. Les alertes peuvent inclure des variations de température, la présence de flammes, des changements de poids significatifs, etc.

Chapitre 3. Solution proposée

L'architecture de la ruche connectée est illustrée dans la figure 3.19, ainsi que l'architecture du fonctionnement global de notre système dans la figure 3.20.

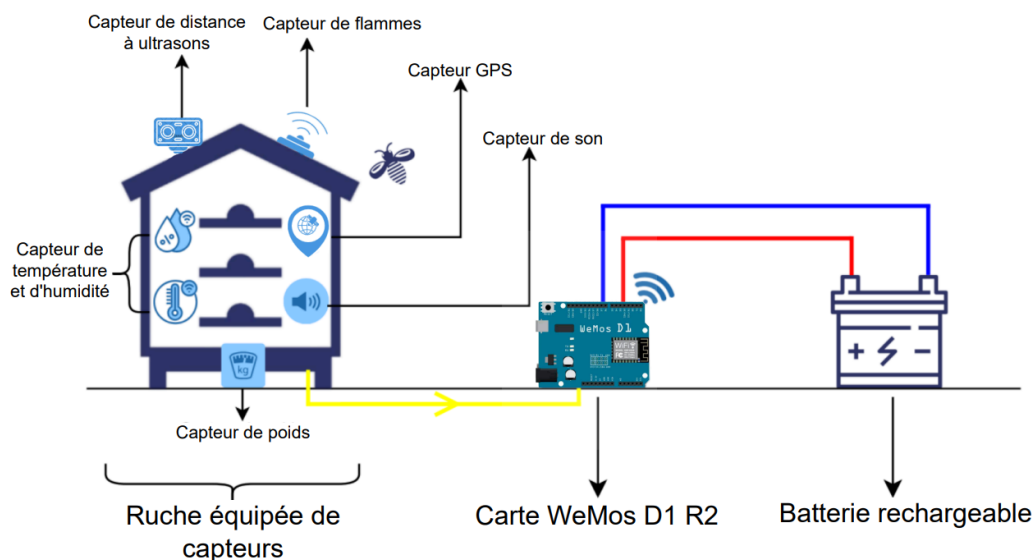


FIGURE 3.19 – Le schéma montrant l'architecture de la ruche connectée

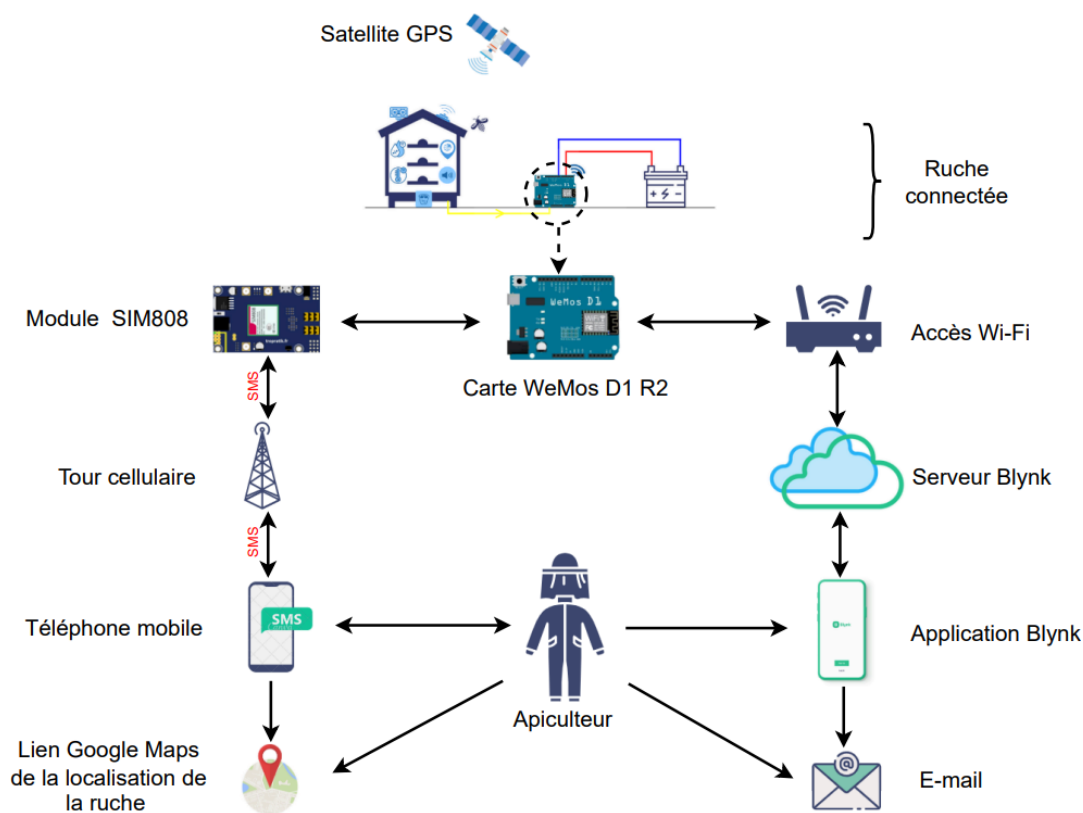


FIGURE 3.20 – Le schéma montrant l'architecture du fonctionnement global de notre système

3.8.1 Les étapes à réaliser

Pour assurer le bon fonctionnement de notre solution, il est essentiel de suivre attentivement les étapes suivantes :

1. Choix approprié des capteurs à utiliser selon deux critères importants : les contraintes relatives à la ruche et les contraintes énergétiques.
2. Étude de la disposition de ces capteurs pour que la meilleure acquisition soit privilégiée et qu'ils ne soient pas invasifs pour les abeilles.
3. Connexion de chaque capteur à la carte WeMos D1 R2 en utilisant les câbles de connexion appropriés.
4. Création d'un projet dans l'interface Blynk, y compris la configuration des broches virtuelles et des widgets (jauges, indicateurs numériques) pour afficher les données des capteurs, ainsi que le paramétrage des événements.
5. Création d'un nouvel appareil ESP8266 dans l'application Blynk et récupération des informations d'authentification nécessaires pour la communication avec le cloud.
6. Installation de l'IDE Arduino, des outils nécessaires pour programmer le système ainsi que le gestionnaire de cartes ESP8266.
7. Programmation du système avec l'IDE arduino pour l'acquisition et la collecte de ces différentes données.
8. Configuration de la carte WeMos D1 R2 : Sélectionner le type de carte WeMos D1 R2, configurer la vitesse de communication et le numéro de port COM associé à la carte.
9. Connexion de la carte WeMos D1 R2 à l'application Blynk via le réseau Wi-Fi. Cela permet à l'apiculteur de visualiser les données collectées.
10. Test de la solution : Une fois la carte WeMos D1 R2 programmée et l'application Blynk configurée, il est possible de tester la solution en installant les capteurs dans la ruche et en vérifiant que les données sont correctement collectées et affichées dans l'application Blynk.

3.8.2 Les contraintes du système

- **Contrainte énergétique** : Choisir une méthode optimale pour l'alimentation du système afin d'assurer une autonomie prolongée.
- **Contrainte environnementale** : Assurer que le système n'ait pas d'impact négatif sur l'environnement des abeilles ni sur leur comportement naturel.
- **Contrainte de disposition** : Choisir soigneusement le positionnement des capteurs au niveau de la ruche pour obtenir des données fiables et précises.
- **Contrainte de réseau** : Garantir qu'il y ait un réseau disponible à proximité de la ruche pour permettre la communication avec l'apiculteur.

3.8.3 L'estimation budgétaire du système

Après avoir sélectionné les matériaux, il est essentiel de connaître le prix de chaque composant pour obtenir une estimation précise du coût total du système.

Composant	Quantité	Prix unitaire (en DA)	Prix total (en DA)
Carte WeMos D1 R2	1	2 000	2 000
Capteur DHT22	1	1 500	1 500
Jauges de contrainte 50 KG	4	600	2 400
Amplificateur HX711	1	700	700
Capteur de son KY-038	1	350	350
Capteur de distance HC-SR04	1	500	500
Module SIM808 GSM/G-PRS/GPS	1	8 000	8 000
Capteur de flammes YG1006	1	350	350
Breadboard 1660 points	1	2 000	2 000
Les câbles mâle-femelle	1 kit (40 câbles)	400	400
Les câbles mâle-mâle	1 kit (40 câbles)	400	400
Écran LCD (16*2)	1	900	900
Adaptateur I2C	1	400	400
Batterie LiPo 2600mAh 11.1V	1	6 900	6 900
Total			26 800

TABLE 3.2 – Liste des composants avec leurs quantités et prix

Ces composants proviennent du laboratoire de recherche LAMOS de l’université de Béjaïa.

Le coût total du système est estimé à environ 26 800 DA, comme indiqué dans le tableau 3.2. Ce montant comprend les capteurs, la carte WeMos D1 R2 ainsi que les autres composants utilisés.

3.8.4 Les avantages de ce système

Notre solution de surveillance à distance des ruches offre plusieurs avantages aux apiculteurs, notamment :

- Réduire les déplacements vers les ruches, ce qui permet de gagner du temps, de l’énergie et de diminuer les risques de transmission d’affection d’une ruche à l’autre par l’intermédiaire de l’homme.
- Repérer automatiquement toute variation anormale des paramètres surveillés, permettant une alerte rapide de l’apiculteur.
- Détecter la présence de maladies et de stress dans la colonie.
- Localiser les ruches perdues ou les mouvements suspects à proximité.
- Suivre les ruches en déplacement lors de la transhumance.
- Identifier les moments optimaux pour l’extraction du miel (en suivant l’évolution du poids de la ruche).

3.9 Conclusion

Dans ce chapitre, nous avons détaillé les éléments constitutifs de notre solution de surveillance à distance d'une ruche d'abeilles. En combinant la carte WeMos D1 R2, les capteurs nécessaires ainsi que la plateforme Blynk, nous avons sélectionné le matériel adéquat pour répondre aux besoins spécifiques de notre entreprise de stage. En outre, nous avons décrit de manière exhaustive le principe de fonctionnement et les étapes nécessaires pour mettre en œuvre notre solution.

Le chapitre suivant abordera l'implémentation de notre solution et les étapes pour sa mise en place pratique.

CHAPITRE 4

MISE EN ŒUVRE ET RÉALISATION DU SYSTÈME

4.1 Introduction

Ce chapitre couvrira la mise en œuvre de notre système de surveillance à distance d'une ruche d'abeilles, nous commencerons par décrire l'assemblage des composants matériels et logiciels, notamment l'installation des capteurs dans la ruche et les différents codes utilisés pour chaque capteur. Ensuite, nous procéderons à des tests pour évaluer la précision des données collectées et la fiabilité des transmissions.

4.2 Branchement de l'afficheur LCD

Avant de procéder à la connexion, nous avons soudé l'adaptateur I2C à l'arrière de l'afficheur (figure 4.1). Ensuite nous avons connecté la broche GND de l'adaptateur I2C à la broche GND de la carte WeMos D1 R2, la broche Vcc à 5V, ainsi que les broches SCL et SDA du l'adaptateur I2C respectivement aux broches D1 et D2 de la carte WeMos.



FIGURE 4.1 – Le montage de l'écran LCD avec son adaptateur I2C

L'emplacement de l'écran LCD dans la ruche :

Nous avons positionné l'écran LCD à l'extérieur de la ruche, permettant aux apiculteurs de visualiser directement sur celui-ci les résultats des paramètres mesurés tels que la température, l'humidité, le son et le poids, sans avoir besoin de consulter l'application Blynk.

4.3 Branchement et programmation du capteur de température et d'humidité DHT22

4.3.1 Le câblage

Pour utiliser le capteur DHT22 avec la carte WeMos D1 R2, nous avons connecté la broche (+) à 5V, la broche (OUT) à la broche D4 et la broche (-) à la masse GND, comme montré dans le schéma de connexion représenté dans la figure 4.2.

Le branchement du capteur est illustré dans la figure 4.3.

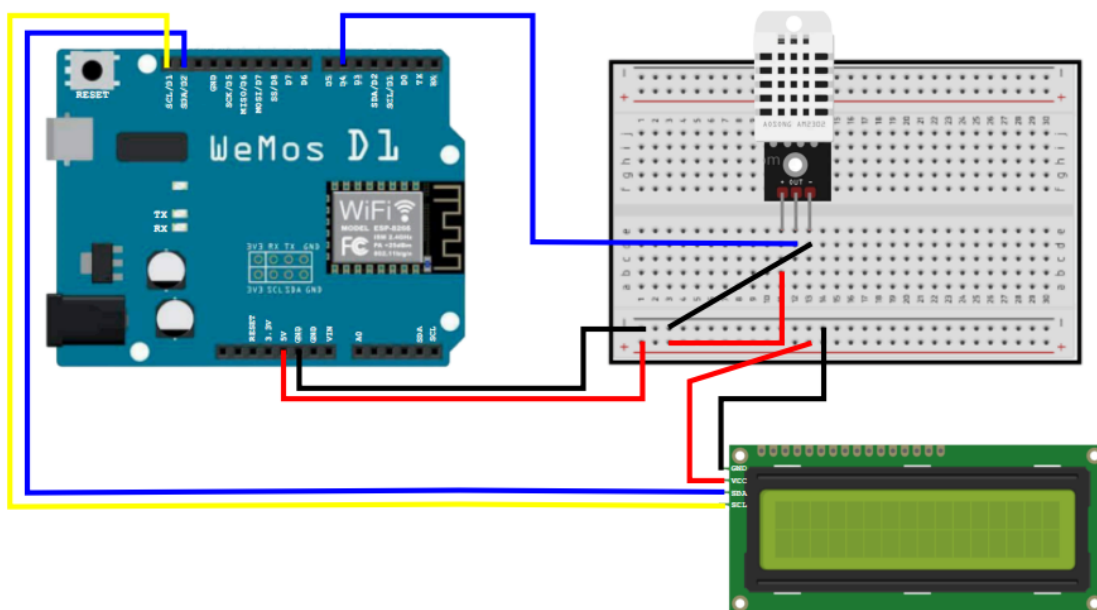


FIGURE 4.2 – Schéma de connexion du capteur DHT22 avec l'afficheur LCD I2C

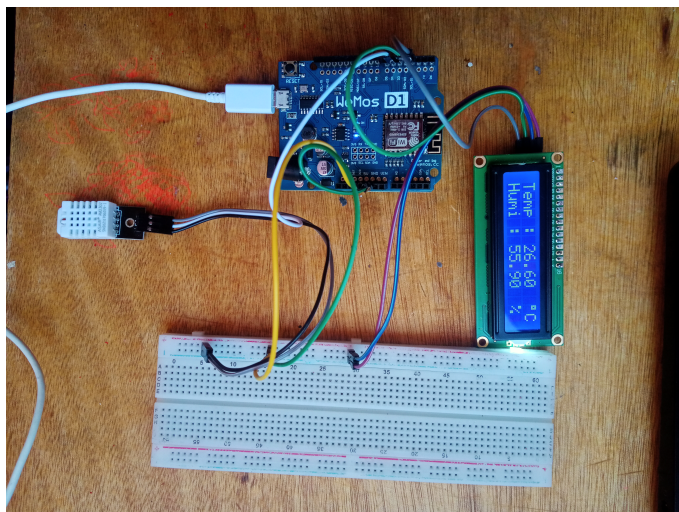


FIGURE 4.3 – Branchement du capteur de température et d’humidité DHT22

4.3.2 Le code utilisé avec le capteur DHT22

```
1 // Identifiant du modèle Blynk utilisé dans l'application
2 #define BLYNK_TEMPLATE_ID "TMPL2rWOwd6Ah"
3 // Nom du modèle Blynk utilisé dans l'application
4 #define BLYNK_TEMPLATE_NAME "Mon Projet"
5 // Jeton d'authentification Blynk
6 #define BLYNK_AUTH_TOKEN "Lw66giatr4cjBK1lqg6c0l8p64ZX0f8h"
7 #define BLYNK_PRINT Serial
8 // Inclusions des bibliothèques nécessaires
9 #include <ESP8266WiFi.h>
10 #include <BlynkSimpleEsp8266.h>
11 #include <Wire.h>
12 #include <LiquidCrystal_I2C.h>
13 #include <DHT.h>
14 //Initialisation de l'écran LCD avec l'adresse 0x27, 16 colonnes et 2
    lignes
15 LiquidCrystal_I2C lcd(0x27, 16, 2);
16 char auth[] = BLYNK_AUTH_TOKEN;
17 // Informations d'identification WiFi
18 char ssid[] = "D-Link_DAP-1360";
19 char pass[] = "";
20 #define DHTPIN 2
21 #define DHTTYPE DHT22
22 DHT dht(DHTPIN, DHTTYPE);
23 BlynkTimer timer;
24 void checkTempHumidSensorr() {
25     float h = dht.readHumidity();
26     float t = dht.readTemperature();
27     if (isnan(h) || isnan(t)) {
28         Serial.println("Failed to read from DHT sensor!");
29         return; }
30     Blynk.virtualWrite(V1, h);
31     Blynk.virtualWrite(V0, t);
32 // Affiche les valeurs sur le moniteur série
33     Serial.print("Temperature : ");
34     Serial.print(t);
```

```
35 Serial.println(" \xDF"C"); // Affiche le symbole degré Celsius sur
    le moniteur série
36 Serial.print("humidity : ");
37 Serial.print(h);
38 Serial.println(" %");
39 // Affiche les valeurs sur l'écran LCD
40 lcd.setCursor(0, 0);
41 lcd.print("Temp : ");
42 lcd.print(t);
43 lcd.print(" \xDF"C"); // Affiche le symbole degré Celsius sur l'é
    cran LCD
44 lcd.setCursor(0, 1);
45 lcd.print("Humi : ");
46 lcd.print(h);
47 lcd.print(" % ");
48 //Avertissements
49 if(t >= 38){
50   Blynk.logEvent("Temp et humid_alerte", "La température à l'intérieur
    de la ruche est élevée");
51 }if(t <= 28){
52   Blynk.logEvent("Temp et humid_alerte", "La température à l'intérieur
    de la ruche est basse");
53 }if(h >= 65){
54   Blynk.logEvent("Temp et humid_alerte", "L'humidité à l'intérieur de la
    ruche est élevée");
55 }if(h <= 40){
56   Blynk.logEvent("Temp et humid_alerte", "L'humidité à l'intérieur de la
    ruche est basse");
57 }void setup(){
58   Wire.begin(D2, D1); // Initialise la communication I2C avec les
    broches D2 (SDA) et D1 (SCL)
59   lcd.init(); // Initialise l'écran LCD
60   lcd.backlight(); // Active le rétroéclairage de l'écran LCD
61   Serial.begin(115200);
62   Blynk.begin(auth, ssid, pass);
63   dht.begin();
64   timer.setInterval(2500L, checkTempHumidSensor);
65 }void loop(){
66   Blynk.run();
67   timer.run(); }
```

4.3.3 La description des fonctions utilisées

Le tableau 4.1 présente les différentes fonctions utilisées dans le programme avec le capteur DHT22, ainsi que leur description.

Nom de la fonction	Description et exemples
<code>checkTempHumidSensor()</code>	Responsable de la lecture des données du capteur DHT22, qui mesure à la fois l'humidité (h) et la température (t).
<code>dht.readHumidity()</code>	Utilisée pour lire l'humidité, qui est stockée dans la variable h.
<code>dht.readTemperature()</code>	Utilisée pour lire la température, qui est stockée dans la variable t.
<code>isnan(Variable)</code>	Fonction prédéfinie utilisée pour vérifier si une valeur est un NaN (Not a Number).
<code>begin()</code>	Méthode utilisée pour initialiser un périphérique ou une communication. Serial.begin(115200) : Initialise la communication série à une vitesse de transmission de 115200 bauds. Blynk.begin(auth, ssid, pass) : Initialise la connexion Blynk avec les identifiants d'authentification, le nom du réseau WiFi et le mot de passe spécifiés. dht.begin() : Initialise le capteur DHT pour permettre la lecture des données à partir de celui-ci. Wire.begin(D2, D1) : Initialise la communication I2C en spécifiant les broches D2 D2 pour les données (SDA) et D1 pour l'horloge (SCL).
<code>virtualWrite(broche, valeur)</code>	Utilisée pour envoyer des données vers les widgets associés à des broches spécifiques dans l'application Blynk. Blynk.virtualWrite(V0, t) : Pour envoyer la valeur de la température (t) au widget associé à la broche virtuelle V0 dans l'application Blynk.
<code>print()</code> et <code>println()</code>	Affiche les valeurs sur le moniteur série de l'IDE Arduino. Serial.print(t) : Affiche la température.
<code>lcd.init()</code>	Initialise l'écran LCD, configurant l'interface I2C et préparant l'écran à afficher des données.
<code>lcd.setCursor(C, L)</code>	Positionne le curseur à la colonne C et à la ligne L de l'écran LCD. lcd.setCursor(0, 0) : Positionne le curseur à la colonne 0 et à la ligne 0 de l'écran LCD.
<code>lcd.print(valeur)</code>	Imprime la valeur de la variable spécifiée à la position actuelle du curseur sur l'écran LCD. lcd.print(t) : Imprime la valeur de la variable t à la position actuelle du curseur sur l'écran LCD.

Nom de la fonction	Description et exemples
<code>logEvent("Nom de l'évènement", "Description de l'évènement")</code>	Fonction de la bibliothèque Blynk qui permet d'enregistrer des événements dans le journal de l'application Blynk. Blynk.logEvent("Temp et humid_alerte", "Description") : Utilisée pour enregistrer les événements de température et d'humidité.
<code>setInterval(intervalle en millisecondes, Nom de fonction)</code>	Utilisée pour exécuter une fonction spécifique à des intervalles de temps réguliers, définis en millisecondes. timer.setInterval(2500L, checkTempHumidSensor) : Configure la fonction <code>checkTempHumidSensor</code> pour qu'elle soit appelée toutes les 2,5 secondes.
<code>run()</code>	est utilisée pour exécuter et maintenir des processus en cours ainsi que pour gérer des tâches qui se répètent à intervalles réguliers. Blynk.run() : Maintient la connexion Blynk. Timer.run() : Exécute les fonctions planifiées.

TABLE 4.1 – Description des fonctions principales utilisées avec le capteur de température et d'humidité

L'emplacement du capteur DHT22 dans la ruche :

Nous avons inséré le capteur DHT22 à l'intérieur du corps de la ruche, pour garantir des conditions de mesure optimales.

4.4 Branchement et programmation du capteur de distance ultrasonique HC-SR04

4.4.1 Le câblage

Pour connecter le capteur HC-SR04 à la carte WeMos, nous avons relié la broche VCC à la broche 5V, la broche GND (masse) à l'une des broches de masse (GND), la broche Trig à la broche D5 et la broche Echo à la broche D6, comme montré dans le schéma de connexion représenté dans la figure 4.4.

Le branchement est illustré dans la figure 4.5.

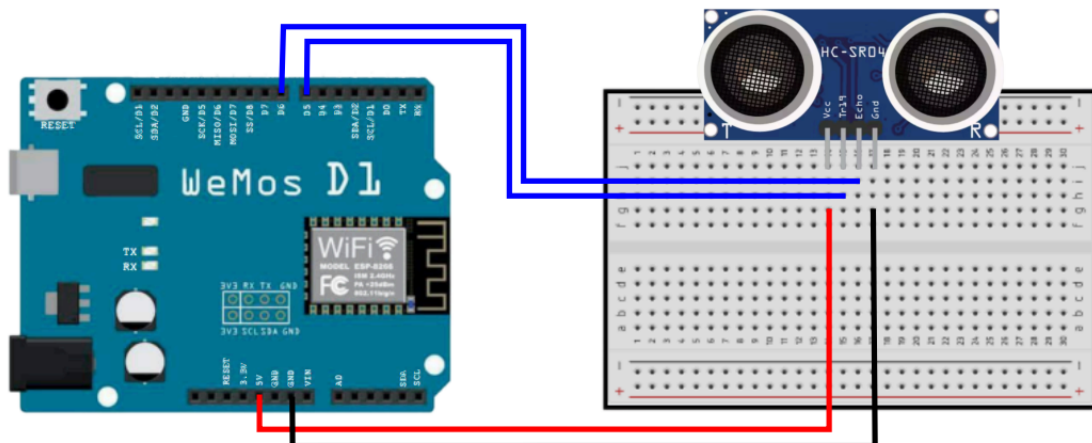


FIGURE 4.4 – Schéma de connexion du capteur de distance ultrasonique HC-SR04

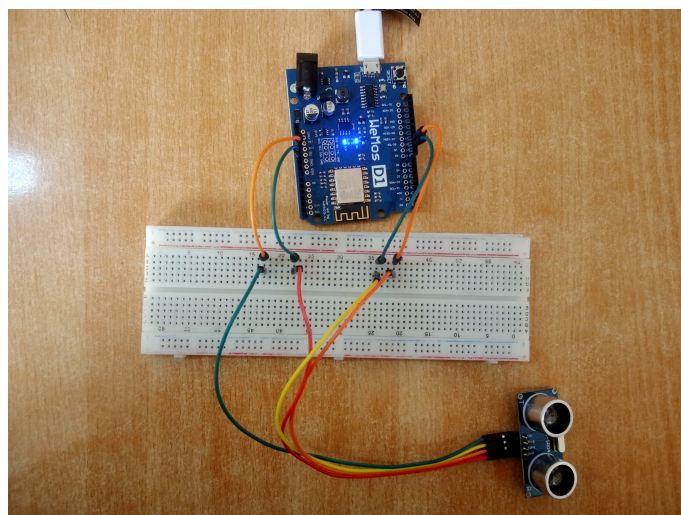


FIGURE 4.5 – Branchement du capteur de distance ultrasonique HC-SR04

4.4.2 Le code utilisé avec le capteur HC-SR04

```
1 /*Définition des identifiants du modèle Blynk pour l'application*/
2 #define BLYNK_TEMPLATE_ID "TMPL2rWOwd6Ah"
3 #define BLYNK_TEMPLATE_NAME "Mon Projet"
4 #define BLYNK_AUTH_TOKEN "Lw66giatr4cjBK1lqg6c0l8p64ZX0f8h"
5 /* Définition de la sortie série pour le débogage Inclusion des
   bibliothèques nécessaire*/
6 #define BLYNK_PRINT Serial
7 #include <ESP8266WiFi.h>
8 #include <BlynkSimpleEsp8266.h>
9 /*Définition des broches de déclenchement et d'écho pour le capteur
   ultrasonique et Déclaration des variables pour la durée et la
   distance*/
```



```
10 #define trig D5
11 #define echo D6
12 long duration;
13 int distance;
14 const int DISTANCE_THRESHOLD = 100;
15 // Authentification Blynk
16 char auth[] = BLYNK_AUTH_TOKEN;
17 // Informations d'identification WiFi
18 char ssid[] = "D-Link_DAP-1360";
19 char pass[] = "";
20 BlynkTimer timer;
21 void setup()
22 {
23 // Configuration des broches pour le capteur ultrasonique
24   pinMode(trig, OUTPUT);
25   pinMode(echo, INPUT);
26 // Démarrage de la communication série
27   Serial.begin(9600);
28 // Connexion au réseau WiFi et au serveur Blynk
29   Blynk.begin(auth, ssid, pass);
30 // Configuration de la fonction à appeler toutes les secondes
31   timer.setInterval(1000L, checkDistanceSensor);
32 }
33 void loop()
34 {
35 // Exécution de la boucle Blynk
36   Blynk.run();
37 // Exécution de la boucle du timer
38   timer.run();
39 }
40 void checkDistanceSensor()
41 {
42 // Envoi d'un signal ultrasonique
43   digitalWrite(trig, LOW);
44   delayMicroseconds(2);
45   digitalWrite(trig, HIGH);
46   delayMicroseconds(10);
47   digitalWrite(trig, LOW);
48 // Mesure de la durée pour obtenir la distance
49   duration = pulseIn(echo, HIGH);
50   distance = duration * 0.034 / 2;
51 // Affichage de la distance sur le moniteur série
52   Serial.print("Distance = ");
53   Serial.print(distance);
54   Serial.println(" cm");
55 // Vérification de la distance par rapport au seuil
56   if (distance < DISTANCE_THRESHOLD) {
57     Serial.println("Proximité détectée, envoi d'une alerte Blynk");
58     Blynk.virtualWrite(V3, "Quelqu'un s'approche de la ruche");
59     Blynk.logEvent("alerte-dist", "Quelqu'un s'approche de la ruche");
60   } else {
61     Blynk.virtualWrite(V3, "Aucune intrusion détectée"); }}
```


4.4.3 La description des fonctions utilisées

Le tableau 4.2 présente les différentes fonctions utilisées dans le programme avec le capteur de distance à ultrasons HC-SR04, ainsi que leur description.

Nom de la fonction	Description et exemples
<code>checkDistanceSensor()</code>	Chargée de mesurer la distance jusqu'à un objet en utilisant un capteur ultrasonique.
<code>pinMode(broche, mode)</code>	Utilisée pour configurer une broche spécifique du microcontrôleur en tant qu'entrée ou sortie. pinMode(trig,OUTPUT) : Configure la broche trig comme sortie (OUTPUT) pour envoyer des impulsions ultrasoniques. pinMode(echo,INPUT) : Configure la broche echo comme entrée (INPUT) pour lire les signaux des échos ultrasoniques réfléchis.
<code>digitalWrite(broche, état)</code>	Utilisée pour définir l'état d'une broche spécifiée du microcontrôleur en signal numérique (haut ou bas). digitalWrite(trig, LOW) : Met la broche trig à l'état bas (LOW) pour préparer le capteur ultrasonique à envoyer une impulsion. digitalWrite(trig, HIGH) : Met la broche trig en état haut (HIGH) pour envoyer un signal d'impulsion.
<code>delayMicroseconds(temps)</code>	Utilisée pour créer une pause ou un délai très court dans l'exécution du programme, mesuré en microsecondes (μ s).
<code>pulseIn(broche, état)</code>	Mesure la durée d'un signal d'impulsion sur une broche spécifiée. pulseIn(echo, HIGH) : Mesure le temps pendant lequel un signal reste à l'état haut (HIGH) sur la broche echo.
<code>Blynk.logEvent("alerte-dist", "Description")</code>	Utilisée pour enregistrer les événements de proximité avec la ruche.
<code>Blynk.virtualWrite(V3, "Quelqu'un s'approche de la ruche")</code>	Utilisée pour envoyer un message indiquant que quelqu'un s'approche de la ruche au widget associé à la broche virtuelle V3 dans l'application Blynk.
<code>timer.setInterval(1000L, checkDistanceSensor)</code>	Utilisée pour planifier l'exécution régulière de la fonction <code>checkDistanceSensor()</code> à intervalles de temps réguliers de 1000 millisecondes.

TABLE 4.2 – Description des fonctions principales utilisées avec le capteur de distance à ultrasons.

L'emplacement du capteur HC-SR04 dans la ruche :

Pour détecter la présence humaine ou animale à proximité de la ruche, nous avons

positionné le capteur ultrasonique HC-SR04 sur le côté, à une hauteur optimale pour capter les mouvements.

4.5 Branchement et programmation du capteur de flammes YG1006

4.5.1 Le câblage

Pour connecter le capteur YG1006 à la carte WeMos D1 R2 nous avons branché la broche VCC à la broche 5V, la broche GND à l'une des broches de masse (GND), la broche DO à la broche D3, comme le montre la figure 4.6. Le branchement est illustré dans la figure 4.7.

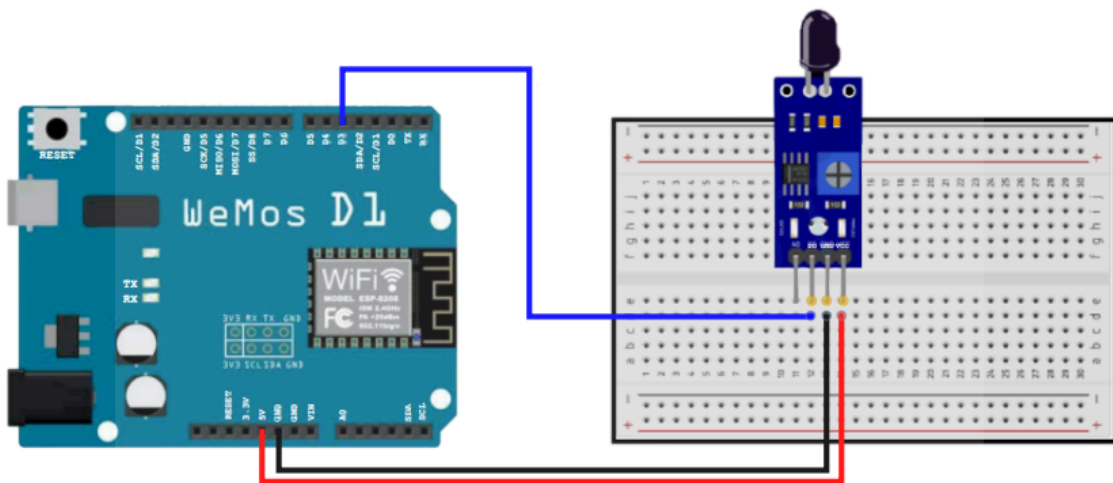


FIGURE 4.6 – Schéma de connexion du capteur de flammes YG1006

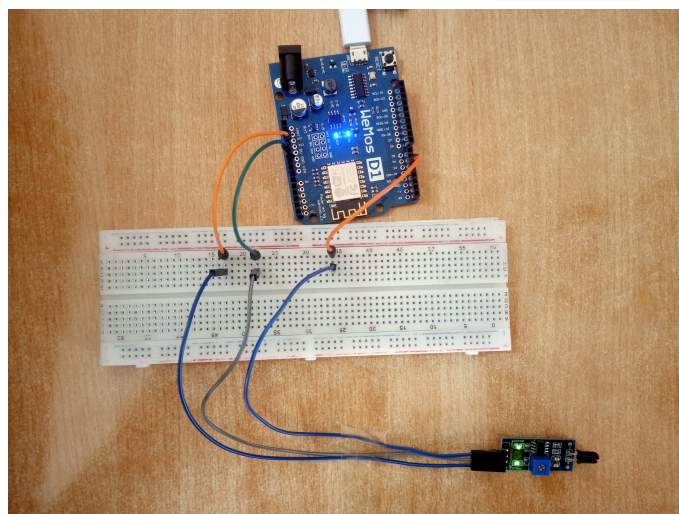


FIGURE 4.7 – Branchement du capteur de flammes YG1006

4.5.2 Le code utilisé avec le capteur YG1006

```
1 // Définition des identifiants du modèle Blynk
2 #define BLYNK_TEMPLATE_ID "TMPL2rWOwd6Ah"
3 #define BLYNK_TEMPLATE_NAME "Mon Projet"
4 #define BLYNK_AUTH_TOKEN "Lw66giatr4cjBK1lqg6c0l8p64ZX0f8h"
5 // Inclure les bibliothèques nécessaires
6 #define BLYNK_PRINT Serial
7 #include <ESP8266WiFi.h>
8 #include <BlynkSimpleEsp8266.h>
9 char auth [] = BLYNK_AUTH_TOKEN;
10 // Informations d'identification WiFi
11 char ssid [] = "D-Link_DAP-1360";
12 char pass [] = "";
13 // Définition de la broche connectée au capteur de flammes
14 #define FLAME_SENSOR 3
15 // Initialisation du timer Blynk
16 BlynkTimer timer;
17 // Variable pour indiquer l'état du capteur de flammes
18 int flag=0;
19 // Fonction pour vérifier l'état du capteur de flammes
20 void checkFlameSensor() {
21     int isButtonPressed = digitalRead(FLAME_SENSOR);
22     if (isButtonPressed == 0 && flag == 0) {
23         Serial.println("Une source de flamme détectée près de la ruche");
24         // Alerte
25         Blynk.logEvent("fire_alerte", "Une source de flamme détectée près
de la ruche");
26         flag = 1;
27     } else if (isButtonPressed == 1) {flag = 0;
28     }}
29 void setup() {
30     pinMode(FLAME_SENSOR, INPUT);
31     Serial.begin(115200);
32     Blynk.begin(auth, ssid, pass);
33     timer.setInterval(5000L, checkFlameSensor);
34 }void loop() {
35     Blynk.run();
36     timer.run();}
```

4.5.3 La description des fonctions utilisées

Le tableau 4.3 présente les différentes fonctions utilisées dans le programme avec le capteur de flammes YG1006, ainsi que leur description.

Nom de la fonction	Description et exemples
<code>checkFlameSensor()</code>	Utilisée pour vérifier si le capteur de flammes détecte une source de flammes.
<code>digitalRead(broche)</code>	Fonction utilisée pour lire l'état d'une broche numérique. digitalRead(FLAME_SENSOR) : Renvoie la valeur de l'état de la broche FLAME_SENSOR, qui peut être soit HIGH (élevé) si aucun signal n'est détecté, soit LOW (bas) si un signal est détecté.
<code>pinMode(FLAME_SENSOR, INPUT)</code>	Permet de Configurer la broche FLAME_SENSOR en tant qu'entrée pour le capteur de flammes.
<code>Blynk.logEvent("fire_alerte", "Description")</code>	Sert à enregistrer un événement dans l'application Blynk lorsqu'une source de flammes est détectée.
<code>setInterval(5000L, checkFlameSensor)</code>	Permet d'appeler la fonction <code>checkFlameSensor()</code> toutes les 5 secondes.

TABLE 4.3 – Description des fonctions principales utilisées avec le capteur de flammes

L'emplacement du capteur YG1006 dans la ruche :

Nous avons positionné le capteur de flammes de manière stratégique sur la ruche pour détecter rapidement toute source de feu. En l'installant sur le dessus, près du toit, nous assurons une détection efficace.

4.6 Branchement et programmation du capteur de son KY-038

4.6.1 Le câblage

Il est simple de connecter le capteur à la carte WeMos D1 R2. Nous avons relié la broche VCC à 5V, la sortie AO à l'entrée A0 et la broche GND du capteur à la masse (GND) de la carte pour établir une connexion complète. La figure 4.8 montre le schéma de connexion du capteur. Le branchement est illustré dans la figure 4.9.

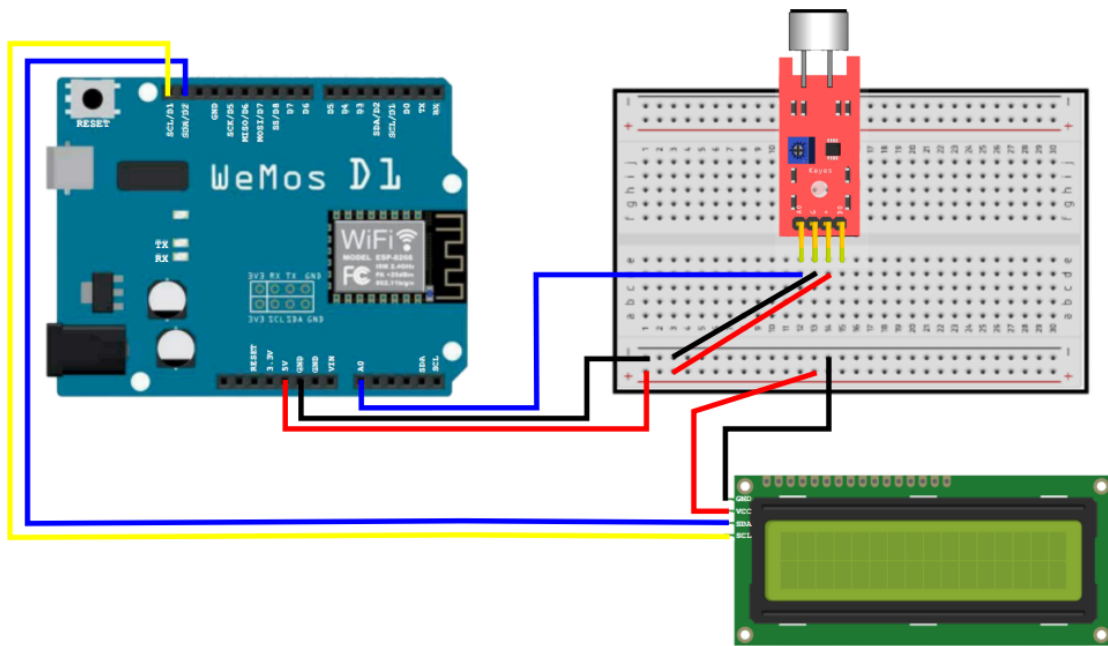


FIGURE 4.8 – Schéma de connexion du capteur de son KY-038 avec l'afficheur LCD I2C

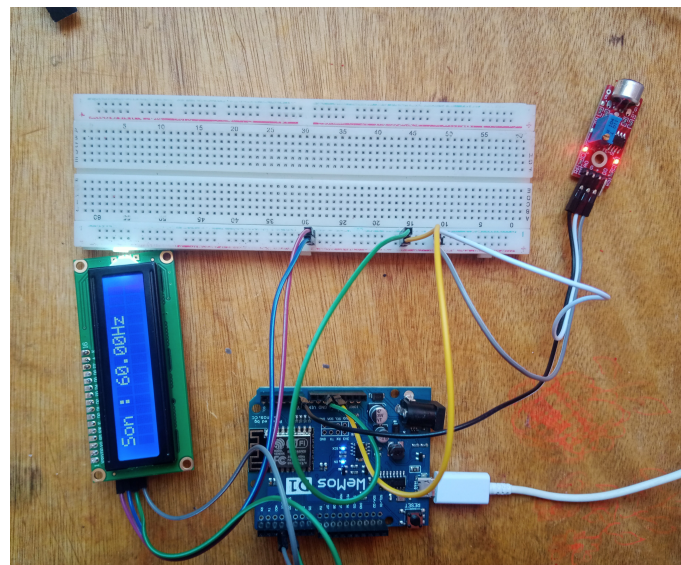


FIGURE 4.9 – Branchement du capteur de son KY-038

4.6.2 Le code utilisé avec le capteur KY-038

```
1 // Auth Token envoyé par Blynk
2 #define BLYNK_TEMPLATE_ID "TMPL2rWOwd6Ah"
3 #define BLYNK_TEMPLATE_NAME "Mon Projet"
4 #define BLYNK_AUTH_TOKEN "Lw66giatr4cjBK1lqg6c0l8p64ZX0f8h"
5 #define BLYNK_PRINT Serial
6 // Inclure les bibliothèques nécessaires
7 #include <ESP8266WiFi.h>
```

```
8 #include <BlynkSimpleEsp8266.h>
9 #include <Wire.h>
10 #include <LiquidCrystal_I2C.h>
11 // Initialisation de l'écran LCD avec l'adresse 0x27, 16 colonnes et 2
    lignes
12 LiquidCrystal_I2C lcd(0x27, 16, 2);
13 char auth[] = BLYNK_AUTH_TOKEN;
14 // Informations d'identification WiFi
15 char ssid[] = "D-Link_DAP-1360";
16 char pass[] = "";
17 const int pinMicrophone = A0; // Broche du capteur de son (KY-038)
18 const int sampleWindow = 50; // Taille de la fenêtre d'échantillonnage
    (ms)
19 unsigned int sample;
20 void setup() {
21     Serial.begin(9600);
22     Blynk.begin(auth, ssid, pass);
23 // Afficher un message de connexion
24     Serial.println("Connecting to Blynk...");
25     Wire.begin(D2, D1);
26     lcd.init();
27     lcd.backlight(); }
28 void loop() {
29     Blynk.run();
30     unsigned long startMillis = millis();
31     unsigned int peakToPeak = 0;
32     unsigned int signalMax = 0;
33     unsigned int signalMin = 1024;
34 // Capture des données du capteur de son pendant la fenêtre d'é
    chantillonnage
35     while (millis() - startMillis < sampleWindow) {
36         sample = analogRead(pinMicrophone);
37         if (sample < 1024) {
38             if (sample > signalMax) {
39                 signalMax = sample;
40             } else if (sample < signalMin) {
41                 signalMin = sample;
42             }
43         }
44         peakToPeak = signalMax - signalMin;
45 // Calcul de la fréquence en Hertz
46         float sonhertz = (1000.0 / sampleWindow) * peakToPeak;
47 // Affichage de la fréquence dans le moniteur série
48         Serial.print("Fréquence : ");
49         Serial.print(sonhertz);
50         Serial.println(" Hz");
51 // Affiche la valeur sur l'écran LCD
52         lcd.setCursor(0, 0);
53         lcd.print("Son : ");
54         lcd.print(sonhertz);
55         lcd.print("Hz");
56 //Alerte
57         if(sonhertz >= 1500){
58             Blynk.logEvent("Son_alerte", "Le niveau sonore à l'intérieur de la
                ruche est élevé");}
59 // Envoyer les données à Blynk
60         Blynk.virtualWrite(V2, sonhertz);delay(1000);}
```

4.6.3 La description des fonctions utilisées

Le tableau 4.4 présente les différentes fonctions utilisées dans le programme avec le capteur de son KY-038, ainsi que leur description.

Nom de la fonction	Description et exemples
<code>millis()</code>	Renvoie les millisecondes écoulées depuis le démarrage du microcontrôleur, utilisée pour mesurer le temps pendant la capture des données du capteur de son.
<code>lcd.print(sonhertz)</code>	Imprime la valeur de la variable <code>sonhertz</code> à la position actuelle du curseur sur l'écran LCD.
<code>Blynk.logEvent("Son_alerte", "Description")</code>	Utilisée pour enregistrer les événements d'alerte sonore.
<code>Blynk.virtualWrite(V2, sonhertz)</code>	Envoie la valeur <code>sonhertz</code> au widget associé à la broche virtuelle V2 sur l'interface Blynk.
<code>delay(durée_en_millisecondes)</code>	Cette fonction suspend l'exécution du programme pendant une durée spécifiée. delay(1000) : Pour mettre en pause l'exécution du programme pendant 1000 millisecondes.

TABLE 4.4 – Description des fonctions principales utilisées avec le capteur de son.

L'emplacement du capteur KY-038 dans la ruche :

Nous avons opté pour placer le capteur de son KY-038 à l'intérieur de la ruche, là où les abeilles sont les plus actives. Cette position stratégique nous permet de capter les sons émis par les abeilles et de surveiller leur activité de manière efficace.

4.7 Branchement et programmation du capteur de poids

4.7.1 Le câblage

La configuration implique la formation d'une boucle, en reliant les fils extérieurs de même couleur (blanc et noir) des quatre cellules de charge. La validation du câblage se fait en mesurant la résistance des deux diagonales (fils rouges), qui doivent être d'environ 2k ohms. Le câble du milieu (rouge) d'une des diagonales est alors connecté aux sorties E+ et E- du module HX711 pour l'alimentation des cellules. L'autre diagonale est branchée aux entrées A+ et A- du module HX711 pour la mesure.^[60]

Le GND du HX711 est relié à la masse de la carte WeMos et le VCC à la broche 5V. Enfin, le DT et le SCK du HX711 sont connectés, respectivement, aux broches

D7 et D8 de la carte WeMos. Le schéma de connexion du capteur est représenté dans la figure 4.10. La figure 4.11 représente le branchement du capteur de poids.

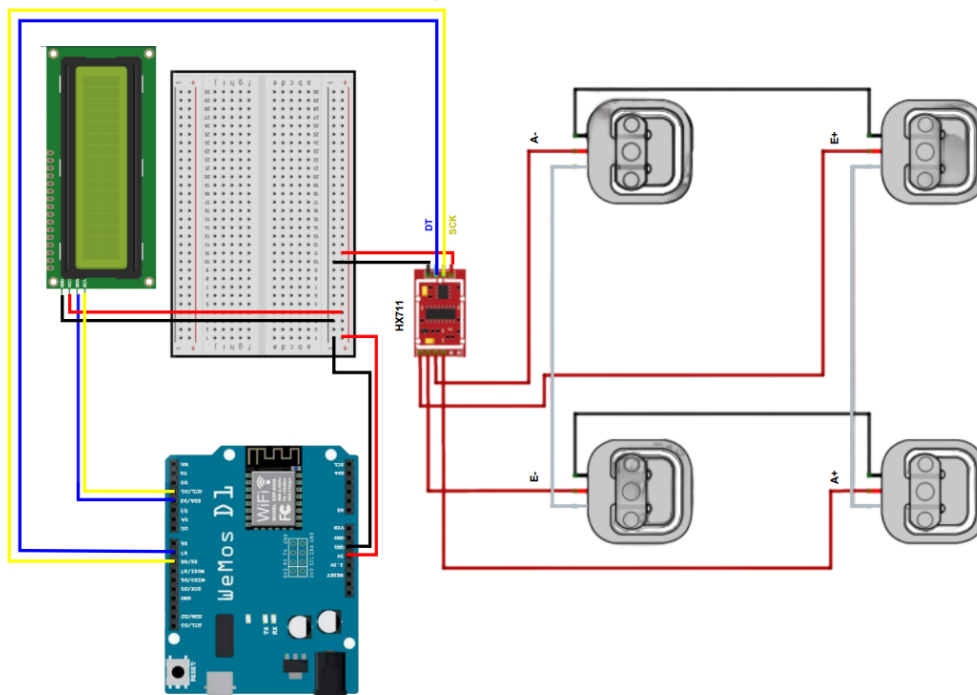


FIGURE 4.10 – Schéma de connexion du capteur de poids avec l’afficheur LCD I2C

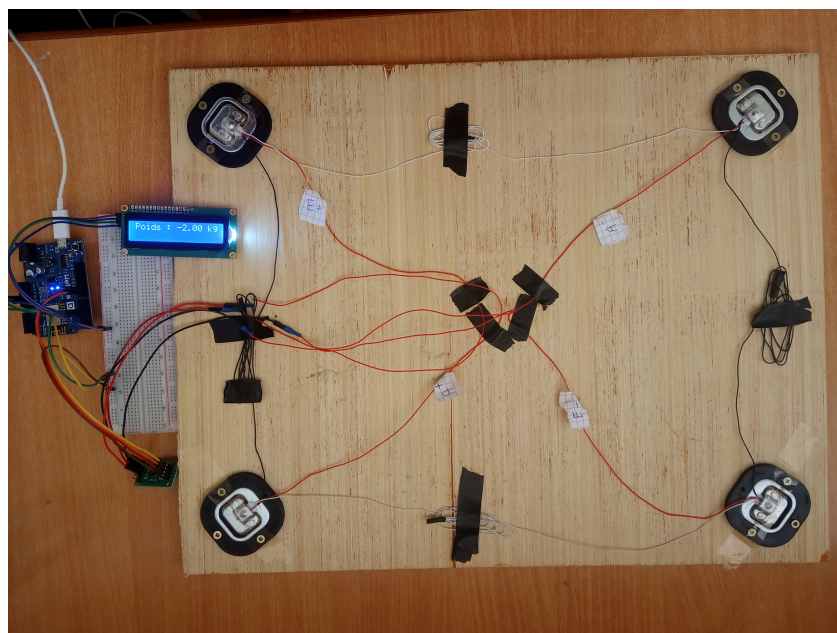


FIGURE 4.11 – Branchement du capteur de poids

Pour assurer la stabilité des mesures, nous avons utilisé un système similaire à celui des balances usuelles. Nous avons préconisé l’utilisation d’une planche en bois sur mesure pour la ruche, d’une épaisseur de 1 cm et de dimensions 47 cm x 37 cm, ainsi que de petits supports 3D, représentés dans la figure 4.12. Ces supports sont

fabriqués sur mesure pour une bonne fixation des cellules de charge. Cette étape est importante pour assurer une stabilité maximale et garantir des mesures précises.

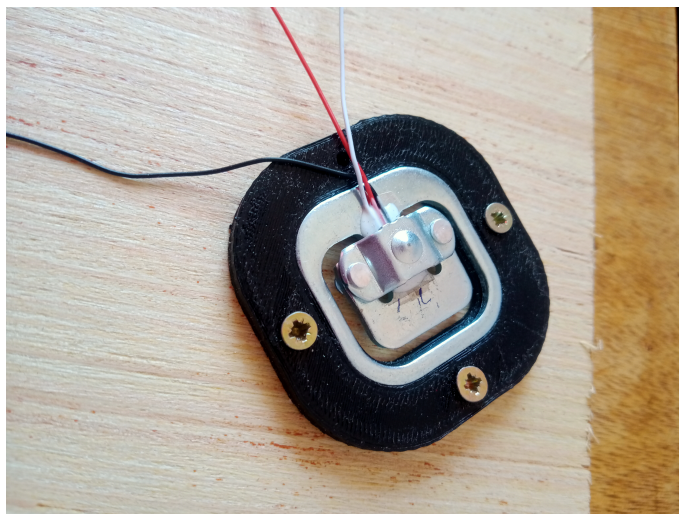


FIGURE 4.12 – Le support 3D utilisé avec le capteur de poids

4.7.2 Le code utilisé avec le capteur de poids

```
1 // Définition des identifiants du modèle Blynk
2 #define BLYNK_TEMPLATE_ID "TMPL2rWOwd6Ah"
3 #define BLYNK_TEMPLATE_NAME "Mon Projet"
4 #define BLYNK_AUTH_TOKEN "Lw66giatr4cjBK11qg6c0l8p64ZX0f8h"
5 // Inclure les bibliothèques nécessaires
6 #include <HX711.h>
7 #include <BlynkSimpleEsp8266.h>
8 #include <ESP8266WiFi.h>
9 #include <Wire.h>
10 #include <LiquidCrystal_I2C.h>
11
12 LiquidCrystal_I2C lcd(0x27, 16, 2); // Adresse I2C 0x27, 16 colonnes et
13 // Configuration des broches pour le HX711
14 const int LOADCELL_DOUT_PIN = D7;
15 const int LOADCELL_SCK_PIN = D8;
16 HX711 scale; // Objet pour la lecture de la balance
17 BlynkTimer timer;
18 // Facteur de calibration et poids de tare
19 float calibration_factor = 21000; // Ajuste cette valeur en fonction de
20 // calibration
21 float tare_weight = 2.0; // Poids de tare à soustraire des lectures
22 float weight_threshold = 60.0; // Poids seuil pour l'alerte
23 char auth[] = BLYNK_AUTH_TOKEN;
24 // Informations d'identification WiFi
25 char ssid[] = "D-Link_DAP-1360";
26 char pass[] = "";
27 void setup() {
28   Serial.begin(115200);
29   // Initialisation de la balance
30   scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
```

```
30  scale.set_scale(calibration_factor); // Ajustement de l'échelle
31  scale.tare(); // Réinitialisation de la balance à zéro
32  // Connexion au WiFi
33  WiFi.begin(ssid, pass);
34  while (WiFi.status() != WL_CONNECTED) {
35      delay(500);
36      Serial.print(".");
37  }
38  Serial.println("WiFi connecté");
39  Blynk.begin(auth, ssid, pass);
40  // Initialize I2C
41  Wire.begin(D2, D1);
42  lcd.init();
43  lcd.backlight();
44  timer.setInterval(1000L, sendWeightSensor);
45  }
46  void sendWeightSensor() {
47      float weight = scale.get_units() - tare_weight; // Lecture du poids
48      // Affiche les valeurs sur le moniteur série
49      Serial.print("Poids : ");
50      Serial.print(weight);
51      Serial.println("      KG");
52      // Affiche la valeur sur l'écran LCD
53      lcd.setCursor(0, 0);
54      lcd.print("Poids : ");
55      lcd.print(weight);
56      lcd.print("kg");
57      Blynk.virtualWrite(V4, weight);
58      if (weight >= weight_threshold) {
59          // ALerte
60          Blynk.logEvent("Poids_alerte", "Le poids de la ruche atteint 60 KG!");
61      }
62  }
63  void loop() {
64      Blynk.run();
65      timer.run();}
```

4.7.3 La description des fonctions utilisées

Le tableau 4.5 représente les différentes fonctions utilisées dans le programme avec le capteur de poids, ainsi que leur description.

Nom de la fonction	Description et exemples
<code>sendWeightSensor()</code>	Fonction responsable de la lecture du poids à partir du capteur de poids HX711 et de l'envoi des données à Blynk.
<code>scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN)</code>	Initialise l'amplificateur HX711 avec les broches définies.
<code>set_scale()</code>	Utilisée pour définir le facteur de calibration de la balance. scale.set_scale(calibration_factor) : Définit le facteur de calibration pour le capteur de poids.
<code>scale.tare()</code>	Met la balance à zéro en supprimant le poids de tare.
<code>WiFi.status()</code>	Vérifie l'état de la connexion WiFi et renvoie une valeur indiquant l'état actuel de la connexion WiFi.
<code>timer.setInterval(1000L, sendWeightSensor)</code>	Permet d'appeler la fonction sendWeightSensor toutes les secondes.
<code>scale.get_units()</code>	Lit les données brutes de la cellule de charge et applique un facteur de calibration pour convertir ces données en unités de poids et retourne la valeur calculée.
<code>lcd.print(weight)</code>	Imprime la valeur de la variable <code>weight</code> à la position actuelle du curseur sur l'écran LCD.
<code>Blynk.virtualWrite(V4, weight)</code>	Envoie la valeur <code>weight</code> au widget associé à la broche virtuelle V4 sur l'interface Blynk.
<code>Blynk.logEvent("Poids_alerte", Description)</code>	Utilisée pour enregistrer les événements d'alerte de poids.

TABLE 4.5 – Description des fonctions principales utilisées avec le capteur de poids

L'emplacement du capteur de poids dans la ruche :

Le capteur de poids est naturellement positionné sous la ruche ; celle-ci est placée sur la balance reliée au module HX711 pour mesurer les variations de poids de manière précise.

4.8 Branchement et programmation du module SIM808 GSM/GPRS/GPS

4.8.1 Le câblage

En premier lieu, nous avons inséré une carte SIM de l'opérateur téléphonique "Djezzy" dont les fréquences sont compatibles avec le module SIM808. Nous avons installées les antennes GPS et GSM dans leurs emplacements désignés sur la carte

SIM808, qui sont des connecteurs coaxiaux SMA. Ensuite, pour connecter le module à la carte WeMos, nous avons relié la broche RXD de la SIM808 à la broche TX de la WeMos, ainsi que la broche TXD à la broche RX de la WeMos. Nous avons également connecter les deux masses (GND) des deux cartes, comme illustré dans le schéma de connexion représenté dans la figure 4.13. Le branchement est montré dans la figure 4.14.

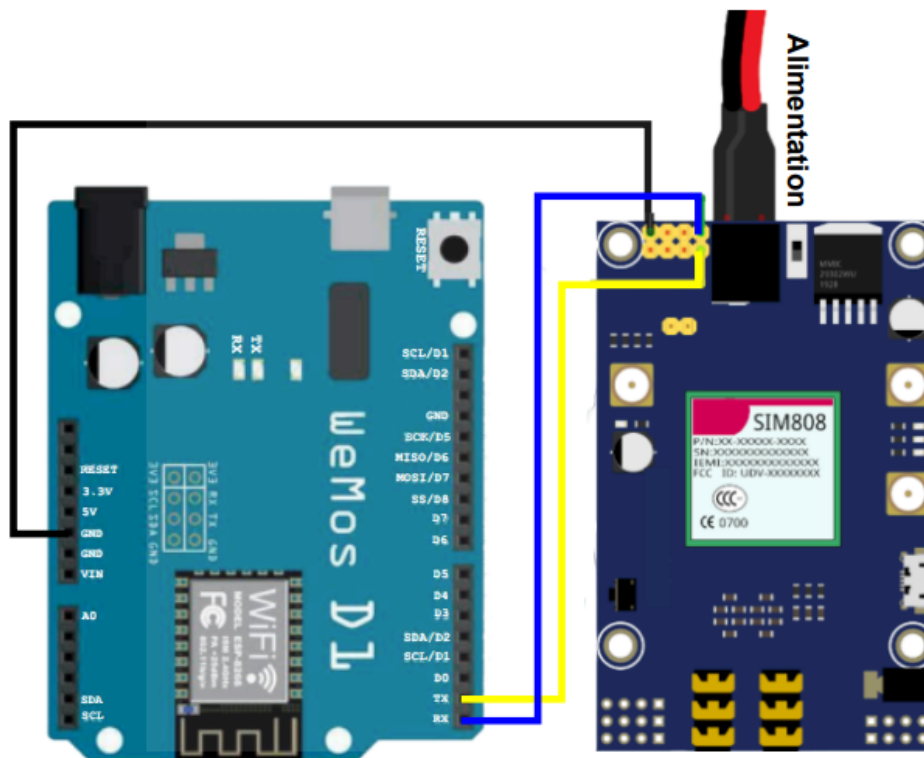


FIGURE 4.13 – Schéma de connexion du module SIM808

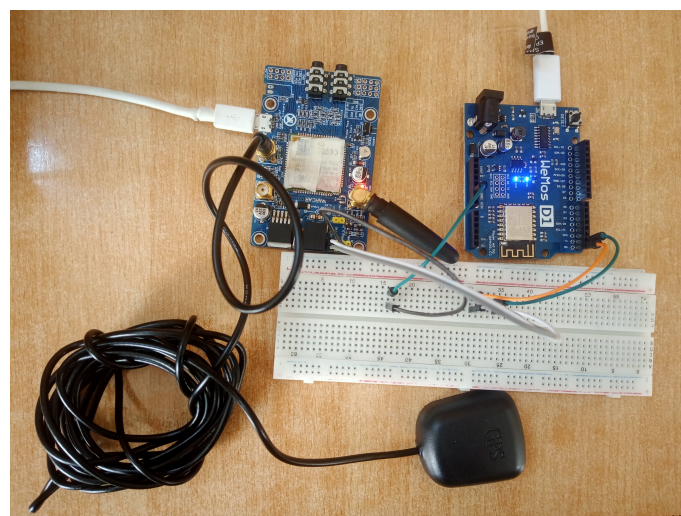


FIGURE 4.14 – Branchement du module SIM808

4.8.2 Le code utilisé avec le module SIM808

```

1 #include "sms.h"
2 #include <AltSoftSerial.h>
3 #include <Sim808_EVB.h>
4 // Dimensionnement des variables
5 #define TAILLE_MAX_MESSAGE 100
6 #define TAILLE_NUMERO 15
7 // Définitions des types
8 typedef struct {
9     char telephone[TAILLE_NUMERO + 1];
10    char message[TAILLE_MAX_MESSAGE + 1];
11 } sms_t;
12 // Prototypes de fonction
13 void initialiser_sim808(void);
14 void envoyer_localisation(const char* numero);
15 void recuperer_et_envoyer_localisation(sms_t* sms);
16 bool lire_premier_sms_non_lu(sms_t* sms_p);
17 void envoyer_sms(const char* numero, const char* message);
18 void demarrer_gps(void);
19 void arreter_gps(void);
20 void donner_coordonees_en_texte(char* position_texte, sexagesimale_t
    latitude, sexagesimale_t longitude);
21 float sexagesimale_vers_degrees_decimaux(sexagesimale_t coordonnee);
22 void flottant_vers_texte(char* texte, float valeur);
23 void donner_lien_google_map(char* lien, sexagesimale_t latitude,
    sexagesimale_t longitude);
24 // Variables globales
25 SMSGSM* gestionnaire_sms;
26 AltSoftSerial altSoftSerial_G;
27 sim808 sim808EVB_G;
28 // Fonction de démarrage, s'exécute une seule fois
29 void setup() {
30     Serial.begin(9600);
31     while (!Serial);
32     pinMode(LED_BUILTIN, OUTPUT);
33     initialiser_sim808();}
34 // Fonction de boucle principale, s'exécute en continu
35 void loop() {
36     sms_t sms_recu;
37     if (lire_premier_sms_non_lu(&sms_recu)) {
38         Serial.print(F("Message reçu de ")); Serial.print(sms_recu.telephone);
39         Serial.print(F(" : ")); Serial.println(sms_recu.message);
40         if (strcmp(sms_recu.message, "localisation") == 0) {
41             recuperer_et_envoyer_localisation(&sms_recu);} else {Serial.println(F("
                Message non reconnu. Aucun traitement effectué."));}}}
42 // Initialise la communication avec la carte SIM808
43 void initialiser_sim808(void) {
44     Serial.println(F("Connexion avec la carte SIM808."));
45     altSoftSerial_G.begin(38400);
46     while (!sim808EVB_G.begin(altSoftSerial_G)) {
47         Serial.println(F("Echec de communication avec la carte SIM808.
                Nouvelle tentative...")); delay(1000);}
48 // Extrait les coordonnées GPS et envoie un lien par SMS
49 void envoyer_localisation(const char* numero) {
50     float latitude, longitude;
51     char message[TAILLE_MAX_MESSAGE + 1];

```

```
52 // Récupérer la latitude et la longitude depuis le module SIM808
53 sim808EVB_G.getLatitude(latitude);
54 sim808EVB_G.getLongitude(longitude);
55 // Formater les coordonnées en format sexagésimal
56 char sexa_latitude[30], sexa_longitude[30];
57 sprintf(sexa_latitude, "%.6f N", latitude);
58 sprintf(sexa_longitude, "%.6f E", longitude);
59 // Construire le message avec les coordonnées et le lien Google
  Maps
60 sprintf(message, "Coordonnées géographiques: %s, %s\n",
  sexa_latitude, sexa_longitude);
61 // Ajouter les coordonnées décimales au lien Google Maps
62 char lien_google[TAILLE_MAX_MESSAGE + 1] = "Lien google Map: https
  ://www.google.com/maps/@?api=1&map_action=map&center=";
63 char dec_latitude[15], dec_longitude[15];
64 dtostrf(latitude, 8, 6, dec_latitude);
65 dtostrf(longitude, 8, 6, dec_longitude);
66 strcat(lien_google, dec_latitude);
67 strcat(lien_google, ",");
68 strcat(lien_google, dec_longitude);
69 strcat(lien_google, "&zoom=18");
70 // Concaténer le lien Google Maps au message
71 strcat(message, lien_google);
72 // Envoyer le message complet par SMS
73 envoyer_sms(numero, message);}
74 // Lit les SMS et envoie la localisation
75 void recuperer_et_envoyer_localisation(sms_t* sms) {
76 char telephone[TAILLE_NUMERO + 1];
77 strcpy(telephone, sms->telephone);
78 demarrer_gps();
79 envoyer_localisation(telephone);
80 arreter_gps();}
81 // Lit le premier SMS non lu
82 bool lire_premier_sms_non_lu(sms_t* sms_p) {
83 bool resultat = false;
84 char telephone[TAILLE_NUMERO + 1];
85 char message[TAILLE_MAX_MESSAGE + 1];
86 char index;
87 // Lire le SMS
88 return resultat;}
89 // Envoie un SMS
90 void envoyer_sms(const char* numero, const char* message) {
91 // Permet d'envoyer un SMS à un numéro de téléphone spécifié}
92 // Démarre le GPS
93 void demarrer_gps(void) {
94 // Active l'alimentation du GPS, initialise et vérifie l'état du
  GPS en attente de la réception de signaux satellites.}
95 // Arrête le GPS
96 void arreter_gps(void) {
97 // Coupe l'alimentation du GPS pour l'arrêter.}
98 // Donne les coordonnées en texte
99 void donner_coordonees_en_texte(char* position_texte, sexagesimale_t
  latitude, sexagesimale_t longitude) {
100 char texte_latitude[15], texte_longitude[15];
101 flottant_vers_texte(texte_latitude,
  sexagesimale_vers_degrees_decimaux(latitude));
102 flottant_vers_texte(texte_longitude,
  sexagesimale_vers_degrees_decimaux(longitude));
```



```

103     sprintf(position_texte, "Latitude : %s, Longitude : %s",
        texte_latitude, texte_longitude);}
104 // Convertit les coordonnées sexagésimales en degrés décimaux
105 float sexagesimale_vers_degrees_decimaux(sexagesimale_t coordonnee) {
106     // Convertit des coordonnées géographiques exprimées en format
        sexagésimal (degrés, minutes, secondes) en degrés décimaux.}
107 // Donne le lien Google Maps
108 void donner_lien_google_map(char* lien, sexagesimale_t latitude,
        sexagesimale_t longitude) {
109     char texte_latitude[15], texte_longitude[15];
110     flottant_vers_texte(texte_latitude,
        sexagesimale_vers_degrees_decimaux(latitude));
111     flottant_vers_texte(texte_longitude,
        sexagesimale_vers_degrees_decimaux(longitude));
112     sprintf(lien, "Lien google Map :https://www.google.com/maps/@?api
        =1&map_action=map&center=%s,%s&zoom=18", texte_latitude,
        texte_longitude);}

```

4.8.3 La description des fonctions utilisées

Le tableau 4.6 représente les différentes fonctions utilisées avec le module SIM808, ainsi que leur description.

Nom de la fonction	Description et exemples
initialiser_sim808()	Initialise la communication avec le module SIM808.
envoyer_localisation(const char *numero)	Récupère et formate les coordonnées GPS du module SIM808 en un message texte avec latitude et longitude, crée un lien Google Maps, puis envoie ces informations par SMS.
recuperer_et_envoyer_localisation(sms_t *sms)	Appelée lors de la réception du message "localisation", la fonction démarre le GPS, récupère les coordonnées GPS du SIM808, les envoie au numéro de téléphone via envoyer_localisation(), puis arrête le GPS.
lire_premier_sms_non_lu(sms_t *sms_p)	Vérifie s'il y a un SMS non lu.
donner_coordonees_en_texte(char* position_texte, sexagesimale_t latitude, sexagesimale_t longitude)	Convertit les coordonnées GPS depuis le format sexagésimal en degrés décimaux, puis les formate dans une chaîne de texte lisible et les stocke dans position_texte.
donner_lien_google_map(char* lien, sexagesimale_t latitude, sexagesimale_t longitude)	Génère un lien Google Maps avec les coordonnées GPS fournies, formaté en degrés décimaux dans la chaîne lien.

TABLE 4.6 – Description des fonctions principales utilisées avec le module SIM808

L'emplacement du module SIM808 dans la ruche :

Pour le capteur antivol GPS, l'emplacement doit être choisi avec soin pour assurer sa sécurité, sa discrétion et sa fonctionnalité optimale. Pendant les tests de fonctionnement, nous l'avons fixé temporairement à l'extérieur de la ruche, positionné de manière à avoir une vue dégagée du ciel pour un bon signal GPS.

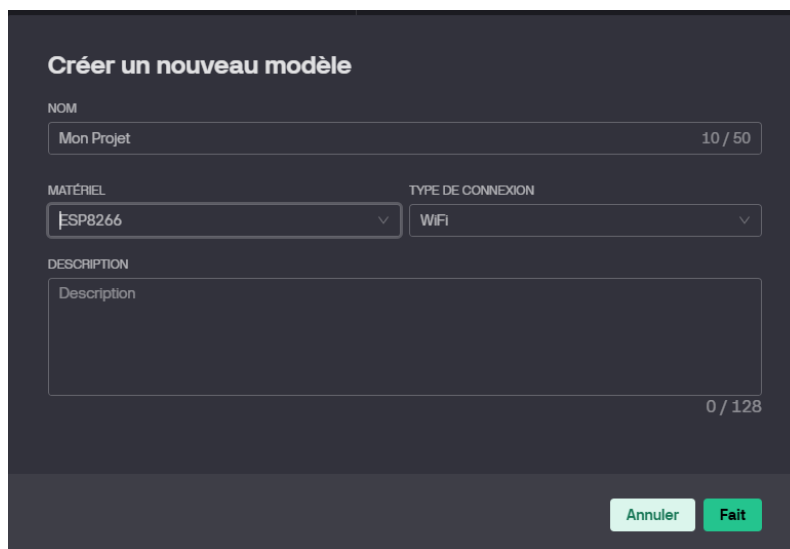
4.9 Configuration de l'application Blynk

4.9.1 La création d'un compte sur l'application Blynk

Afin d'exploiter les fonctionnalités IoT en mode développeur offertes par l'application Blynk, nous avons créé un compte sur l'application. Cette démarche implique le téléchargement et l'installation de l'application, suivie de l'inscription en fournissant une adresse e-mail et un mot de passe valides.

4.9.2 La création d'un projet personnalisé sous le nom "Mon Projet"

Dans cette étape, illustrée dans la figure 4.15, nous avons créé et configuré un projet dans l'application Blynk. Tout d'abord, nous avons attribué le nom "Mon Projet" à notre projet. Ensuite, nous avons sélectionné le matériel "ESP8266" et défini le type de connexion "Wi-Fi".



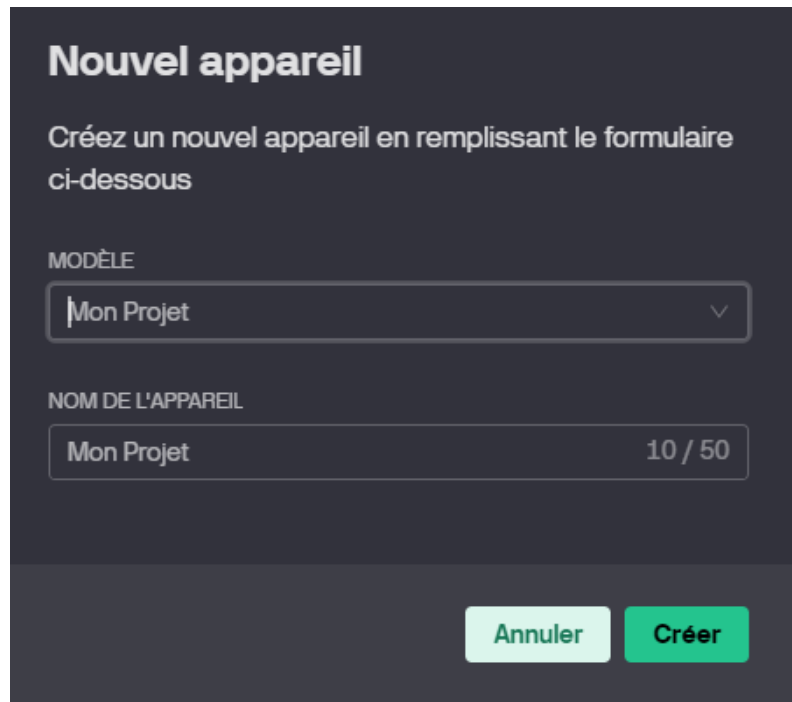
The screenshot shows the 'Créer un nouveau modèle' (Create a new model) screen in the Blynk application. It features a dark theme with white text. At the top, the title 'Créer un nouveau modèle' is displayed. Below it, there are three main sections: 1. 'NOM' (Name): A text input field containing 'Mon Projet' with a character count '10 / 50' on the right. 2. 'MATÉRIEL' (Hardware): A dropdown menu showing 'ESP8266'. 3. 'TYPE DE CONNEXION' (Connection Type): A dropdown menu showing 'WIFI'. Below these is a 'DESCRIPTION' section with a large text area containing the placeholder 'Description' and a character count '0 / 128' at the bottom right. At the bottom of the screen, there are two buttons: 'Annuler' (Cancel) in white on a dark background and 'Fait' (Done) in green on a dark background.

FIGURE 4.15 – Illustration de l'étape de création de "Mon projet"

4.9.3 La création d'un nouvel appareil ESP8266 dans l'application Blynk

Pour cette étape, nous avons ajouté un nouvel appareil ESP8266 dans l'application Blynk (figure 4.16), puis nous avons récupéré les données d'authentification

nécessaires (figure 4.17). Cette étape est essentielle pour connecter notre carte de développement à la plateforme Blynk et permettre la communication avec le cloud.



The screenshot shows a dark-themed web form titled "Nouvel appareil". Below the title, it says "Créez un nouvel appareil en remplissant le formulaire ci-dessous". There are two input fields: "MODÈLE" with a dropdown menu showing "Mon Projet" and "NOM DE L'APPAREIL" with a text input field containing "Mon Projet" and a character count "10 / 50". At the bottom right, there are two buttons: "Annuler" (light green) and "Créer" (dark green).

FIGURE 4.16 – Illustration de l'étape de création d'un nouvel appareil



The screenshot shows a dark-themed notification box titled "Nouveau Dispositif Créé !". It contains a code block with the following text:

```
#define BLYNK_TEMPLATE_ID "TMPL2rW0wd6Ah"  
#define BLYNK_TEMPLATE_NAME "Mon Projet"  
#define BLYNK_AUTH_TOKEN  
"Lw66giatr4cjBK1lqg6c0l8p64ZX0f8h"
```

Below the code, it says "L'ID du Modèle, le Nom du Modèle et le Jeton d'Authentification doivent être déclarés en haut du code du firmware." At the bottom, there are two buttons: "Documentation" (light green) and "Copier dans le presse-papiers" (dark green).

FIGURE 4.17 – Récupération des données d'authentification Blynk

4.9.4 La configuration des widgets dans le tableau de bord

Dans cette étape, représentée dans la figure 4.18, nous avons configuré les widgets du tableau de bord avec des jauges de mesure pour la température, l'humidité, le son et le poids, ainsi qu'un champ "display value" pour afficher un message de proximité avec la ruche et un graphe qui montre les variations de poids.

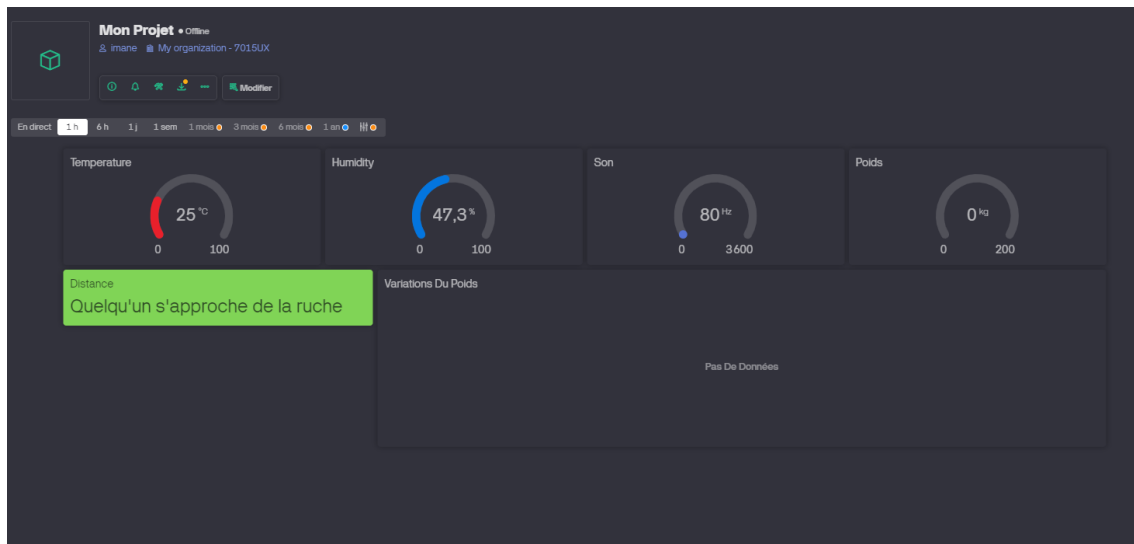


FIGURE 4.18 – Illustration des widgets créés

4.9.5 La création des broches virtuelles

Nous avons créé 5 broches virtuelles (figure 4.19), définissant pour chacune le nom approprié, le type et l'unité de mesure. Ainsi, V0 est dédiée à la température, V1 à l'humidité, V2 au son, V3 au message de proximité et V4 au poids. Chaque broche virtuelle est associée à un widget spécifique dans l'interface de l'application Blynk.

The screenshot shows the 'Flux de données' configuration page in Blynk. It displays a table with 5 rows, each representing a virtual pin configuration. The table has columns for Id, Nom, Alias, Couleur, Épinglé, Type de..., Unités, Brut, Min., Max., Décimales, and Actions.

Id	Nom	Alias	Couleur	Épinglé	Type de...	Unités	Brut	Min.	Max.	Décimales	Actions
1	Temperature	Temperature	■	V0	Double	°C	false	0	100	###	
2	Humidity	Humidity	■	V1	Double	%	false	0	100	###	
3	Son	Son	■	V2	Double	Hz	false	0	3600	###	
4	Distance	Distance	■	V3	Chaîne		false			-	
5	Poids	Poids	■	V4	Double	kg	false	0	200	###	+

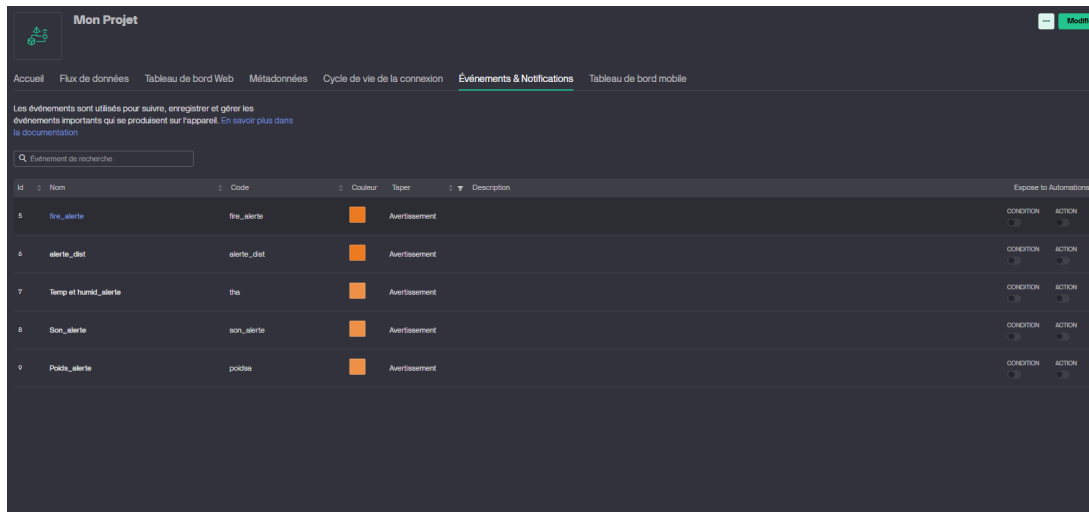
FIGURE 4.19 – Illustration des broches virtuelles créées

4.9.6 La configuration des événements et des notifications

Pour recevoir des alertes et des notifications, nous avons configuré les événements et les notifications dans l'application Blynk, en définissant le nom des événements (figure 4.20) et leur type, soit "Avertissement" ou "Information", et nous avons activé

les options nécessaires pour recevoir les notifications Blynk, même via l'email. Voici les différentes alertes créées :

- "Temp et humid_alerte" pour les alertes de température et d'humidité.
- "alerte_dist" pour l'alerte de proximité.
- "fire_alerte" pour l'alerte de détection d'incendie.
- "Poids_alerte" pour l'alerte de poids.
- "Son_alerte" pour l'alerte de son.



The screenshot shows the Blynk web interface for a project named "Mon Projet". The "Événements & Notifications" tab is active, displaying a table of events. The table has columns for Id, Nom, Code, Couleur, Type, and Description. There are five rows of events, all of type "Avertissement".

Id	Nom	Code	Couleur	Type	Description
5	fire_alerte	fire_alerte	Orange	Avertissement	
6	alerte_dist	alerte_dist	Orange	Avertissement	
7	Temp et humid_alerte	tra	Orange	Avertissement	
8	Son_alerte	son_alerte	Orange	Avertissement	
9	Poids_alerte	poids	Orange	Avertissement	

FIGURE 4.20 – Illustration des événements créés

4.10 Installation du système complet dans une ruche d'abeilles

Nous avons installé notre prototype dans une ruche d'abeilles afin d'évaluer et de tester les performances de notre système, comme illustré dans la figure 4.21.

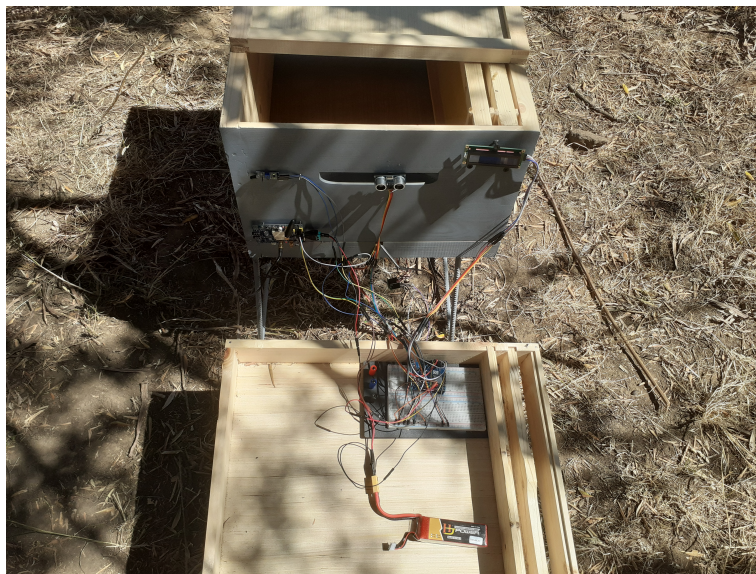


FIGURE 4.21 – Installation du système complet dans une ruche d'abeilles

4.11 Tests et résultats du système

Pour évaluer les performances de notre système, il est essentiel de générer des variations dans les différents paramètres mesurés par notre réseau de capteurs. Il est également important de vérifier le bon transfert de ces données au tableau de bord de l'application Blynk.

Nous l'avons mis en service pendant 5 heures, en effectuant des vérifications régulières. Aucune anomalie n'a été signalée, et le système a répondu de manière stable à nos attentes pendant plusieurs heures.

1. Visualisation des résultats sur le tableau de bord de l'application Blynk :

La figure 4.22 présente les données des capteurs, affichées sur le tableau de bord de l'application Blynk, offrant une image complète de l'environnement de la ruche. La température est observée à 29,1°C, avec un taux d'humidité de 49,2%. La fréquence sonore est de 80 Hz et le poids est d'environ 34,99 kg. Le graphique du poids montre un état stable mais peut aussi révéler des variations au fil du temps. De plus, le message "Quelqu'un s'approche de la ruche" indique qu'une personne se trouve à moins d'un mètre de la ruche.

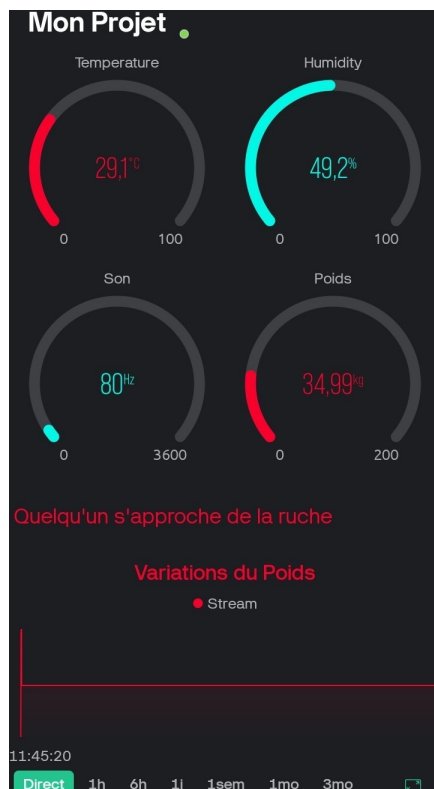
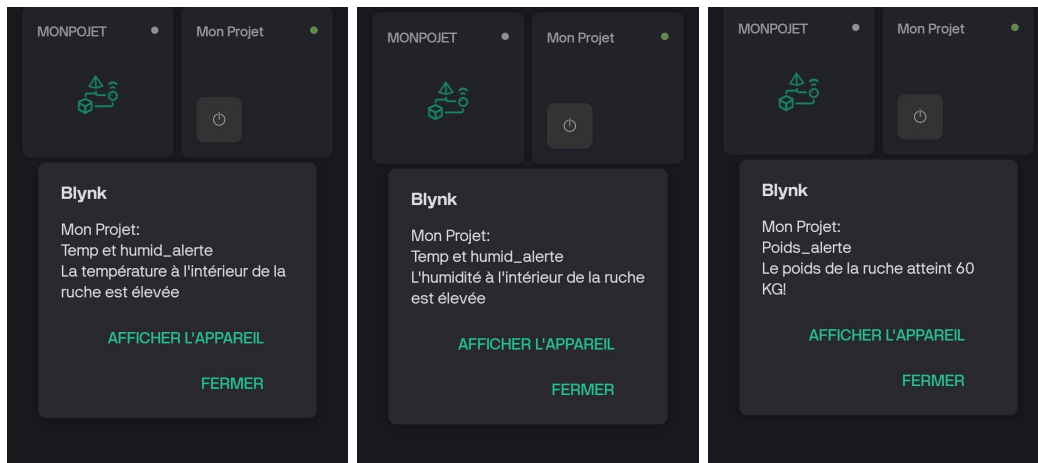


FIGURE 4.22 – Visualisation des résultats sur le tableau de bord de Blynk

2. Visualisation des résultats des événements :

La figure 4.23 montre quelques alertes détectées lors des tests effectués.

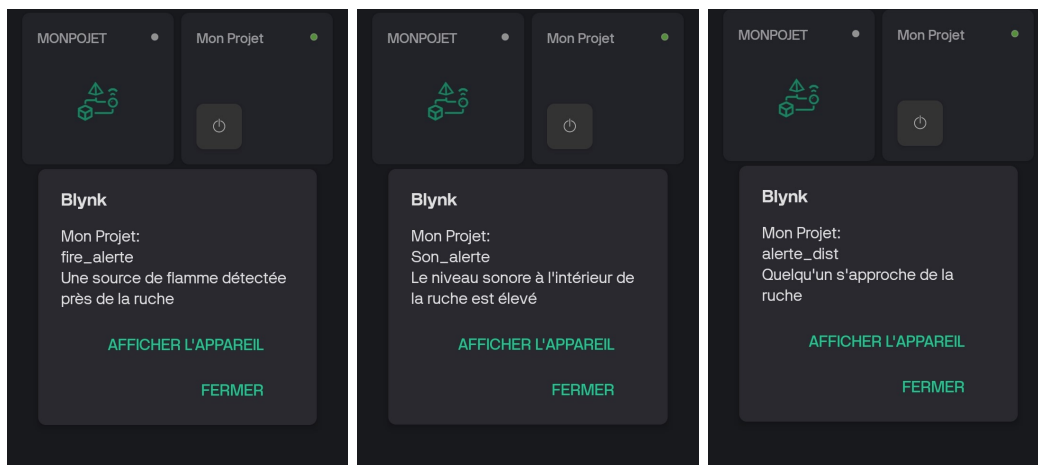
Les sous-figures 4.23a et 4.23b illustrent respectivement des alertes de température élevée et d'humidité élevée. La sous-figure 4.23c montre un poids élevé, suggérant qu'il est peut-être temps de récolter le miel. La sous-figure 4.23d alerte sur la présence de flammes à proximité. La sous-figure 4.23e indique que les fréquences sonores à l'intérieur de la ruche sont élevées. Enfin, la sous-figure 4.23f représente une alerte de présence animale ou humaine près de la ruche.



(a) Alerte de température

(b) Alerte d'humidité

(c) Alerte de poids



(d) Alerte de flammes

(e) Alerte de son

(f) Alerte de proximité

FIGURE 4.23 – Les différentes alertes détectées

3. Visualisation des résultats du capteur GPS :

La figure 4.24 présente les résultats obtenus à l'aide du capteur GPS. La sous-figure 4.24a illustre la récupération des coordonnées géographiques et du lien Google Maps par SMS, tandis que la sous-figure 4.24b montre précisément la localisation sur Google Maps.

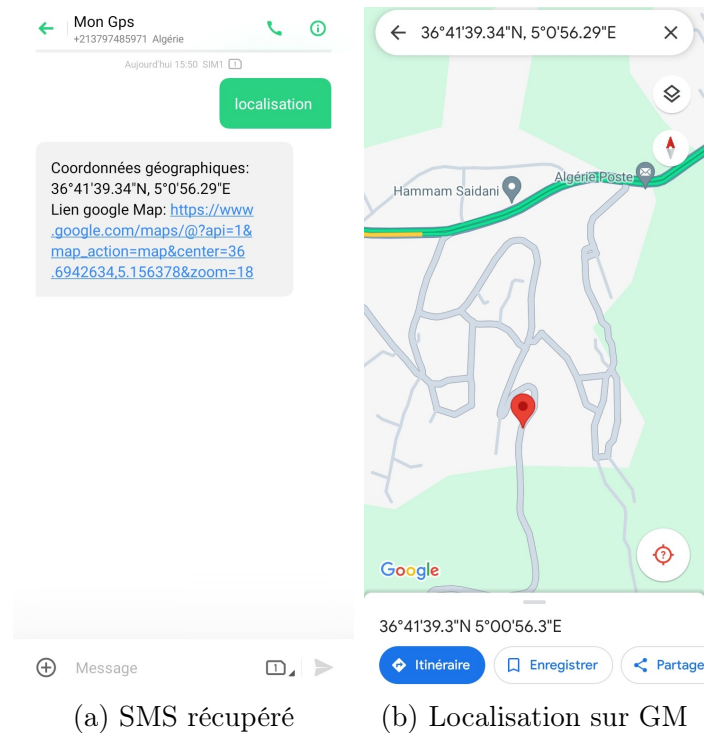


FIGURE 4.24 – Résultats récupérés par le capteur GPS

4.12 Conclusion

Ce chapitre a couvert la mise en œuvre complète de notre système de surveillance des ruches, détaillant l'assemblage des composants matériels et logiciels ainsi que l'installation des capteurs. Nous avons décrit toutes les étapes de conception et de réalisation de notre ruche connectée, y compris la présentation du prototype et le programme utilisé pour permettre aux capteurs de collecter des données et de les envoyer au serveur Blynk. Ce système de surveillance à distance basé sur l'IoT contribuera à améliorer la surveillance des ruches en offrant aux apiculteurs un outil efficace pour protéger leurs colonies.

Conclusion générale et perspectives

Dans ce projet, nous avons créé un système de surveillance à distance d'une ruche d'abeilles basé sur l'IoT, intégrant divers capteurs pour fournir des données en temps réel sur la température, l'humidité, le poids et le bruit. Appelée "ruche connectée", cette solution permet aux apiculteurs de surveiller leurs ruches à distance via Internet grâce à l'application Blynk. En plus de suivre l'évolution des paramètres de la ruche, le système détecte la présence humaine ou animale ainsi que les incendies, et envoie des alertes d'urgence par notifications push et e-mails. Il peut également transmettre l'emplacement exact des ruches par SMS lors de transhumances ou de tentatives de vol.

Ce dispositif représente un avantage considérable pour l'apiculture future, un domaine qui s'appuyait jusqu'à présent sur des méthodes traditionnelles et qui peut être maintenant révolutionné grâce à la technologie. Il vise à optimiser les efforts de l'apiculteur et à assurer un suivi précis de la colonie sans nécessiter de déplacements sur le terrain. De plus, il contribue à la protection et à la préservation des abeilles, essentielles à l'équilibre des écosystèmes.

Les résultats des tests effectués se sont révélés satisfaisants, bien que des améliorations potentielles aient été identifiées pour l'avenir. Comme perspectives, nous citons :

- Ajouter des capteurs supplémentaires pour évaluer l'orientation du vent, la qualité de l'air et la pression atmosphérique, afin d'assurer une surveillance plus complète de l'environnement des ruches.
- Intégrer un système de comptage des abeilles sortantes et entrantes pour mieux comprendre l'activité de la colonie.
- Développer une application dédiée à la surveillance à distance des ruches, offrant des fonctionnalités avancées telles que la gestion complète du rucher, la surveillance simultanée de plusieurs ruches et une interface plus conviviale pour les apiculteurs.
- Protéger les composants électroniques en les plaçant dans un boîtier adapté, afin de les préserver des conditions environnementales et d'assurer la sécurité des abeilles.
- Explorer l'utilisation de modules photovoltaïques pour alimenter le système, avec une configuration permettant la recharge de batteries lithium polymère pour une autonomie prolongée.
- Utiliser des technologies de communication à faible consommation d'énergie, comme LoRaWAN, pour permettre une surveillance à plus grande échelle tout en conservant une autonomie élevée.
- Intégrer un système pour remplir automatiquement le réservoir de sirop pour les abeilles lorsque le niveau est bas.

Ces améliorations futures visent à renforcer les fonctionnalités et la robustesse de notre système de surveillance, offrant ainsi aux apiculteurs des outils plus avancés pour la gestion de leurs colonies.

BIBLIOGRAPHIE

- [1] M. MAUGER, « Les ruches connectées : description des outils connectés commercialisés en France et présentation de leurs intérêts, » thèse de doct., École nationale vétérinaire d'Alfort et Université Paris-Est Créteil Val-de-Marne (France), 2022.
- [2] A.-C. DEBLOCK, *Les Types de Ruches*, Site Web. Disponible à l'adresse : <https://www.aubonmiel.com/les-ruches/>. Consulté le : 2024-07-09, 2017.
- [3] P. MOCKLER et C. TARDIF, « L'importance de la santé des abeilles pour une production alimentaire durable au Canada, » Comité sénatorial permanent de l'agriculture et des forêts, Rapport, 2015.
- [4] A. MALIK et A. R. AZAD, « Economic Importance and Utilization of Apiculture, » *Just Agriculture e-Magazine*, 2020. adresse : <https://justagriculture.in/files/magazine/nov/>.
- [5] FAO, *Produits de la ruche*, Fiche d'informations, 2020. adresse : <https://openknowledge.fao.org/handle/20.500.14283/ca8049fr>.
- [6] NITR, *Jar of honey with honeycomb*, Site Web. Disponible à l'adresse : <https://stock.adobe.com/>. Consulté le : 2024-07-09.
- [7] KOSOLOVSKYY, *Bee propolis*, Site Web. Disponible à l'adresse : <https://www.shutterstock.com/>. Consulté le : 2024-07-09, 2016.
- [8] JPC-PROD, *raw organic royal bee jelly*, Site Web. Disponible à l'adresse : <https://stock.adobe.com/>. Consulté le : 2024-07-09.
- [9] S. IMBODEN, *Abeilles*, Site Web. Disponible à l'adresse : <https://www.2imanagement.ch/fr/divers/liens/wwwapisavoirch/-des-chiffres-etonnants->, Consulté le : 2024-03-04.
- [10] H. ABOU-SHAARA, A. OWAYSS, Y. IBRAHIM et N. BASUNY, « A review of impacts of temperature and relative humidity on various activities of honey bees, » *Insectes Sociaux*, t. 64, p. 455-463, 2017. DOI : [10.1007/s00040-017-0573-8](https://doi.org/10.1007/s00040-017-0573-8).

Bibliographie

- [11] B. ONONYE et C. AKUNNE, « Microclimate, Colonization, Colony Strength and Absconding of *Apis mellifera adansonii* Reared in Hives with Modified Entrances, » *Journal of Apiculture*, t. 34, p. 117-123, 2019. DOI : [10.17519/apiculture.2019.06.34.2.117](https://doi.org/10.17519/apiculture.2019.06.34.2.117).
- [12] O. HONDERMARCK, *Combien pèse une ruche ?* Site Web. Disponible à l'adresse : <https://www.mellifere.com/pages/page-combien-pese-ruche.php>, Consulté le : 2024-06-21.
- [13] A. QANDOUR, I. AHMAD, D. HABIBI et M. LEPPARD, « Remote Beehive Monitoring using Acoustic Signals, » *Acoustics Australia / Australian Acoustical Society*, t. 42, p. 204-209, 2014.
- [14] SNGTV, *Guide pour les apiculteurs débutants*. Ministère de l'Agriculture et de l'Alimentation française, 2022, p. 43. adresse : <https://agriculture.gouv.fr/>.
- [15] F. SÁNCHEZ-BAYO et K. GOKA, « Impacts of Pesticides on Honey Bees, » in *Beekeeping and Bee Conservation*. Rijeka : IntechOpen, 2016, chap. 4, p. 77-97. DOI : [10.5772/62487](https://doi.org/10.5772/62487).
- [16] BAYER BEE CARE, *Santé des Abeilles : Comprendre les Enjeux, Fournir des Solutions*, Rapport d'information. Disponible à : https://www.crop.bayer.com.au/-/media/bcs-inter/ws_australia/use-our-products/product-resources/confidor-200/honey-bee-health.pdf, 2012.
- [17] BAYER BEE CARE ET EDUVERSUM VERLAG, *What affects Honey Bee Health ? What do Honey Bees need ?* Matériel éducatif. Disponible à : https://www.bayer.com/sites/default/files/Educational-Material_Pollinator-Diversity-and-Agriculture_BEE-HEALTH_Work-Sheets-and-Solutions.pdf, 2019.
- [18] J. RANGEL et A. FISHER, « Factors affecting the reproductive health of honey bee (*Apis mellifera*) drones-a review, » *Apidologie*, t. 50, p. 759-778, 2019. DOI : [10.1007/s13592-019-00684-x](https://doi.org/10.1007/s13592-019-00684-x).
- [19] M. BOZORGMANESH et F. KOWKABI, « The Effect of Mobile Phone Electromagnetic Waves on Honey Bee, » *Disponible à SSRN 4439490*, p. 5, 2023.
- [20] K. AHN, X. XIE, J. RIDDLE, J. PETTIS et Z. Y. HUANG, « Effects of Long Distance Transportation on Honey Bee Physiology, » *Psyche : A Journal of Entomology*, t. 2012, p. 1-9, 2012. DOI : [10.1155/2012/193029](https://doi.org/10.1155/2012/193029).
- [21] FAO, *Good beekeeping practices : Practical manual on how to identify and control the main diseases of the honeybee (Apis mellifera)* (TECA – Technologies and practices for small agricultural producers). Rome : Food et Agriculture Organization of the United Nations (FAO), 2020, t. 1, p. 84. DOI : [10.4060/ca9182en](https://doi.org/10.4060/ca9182en).
- [22] OVH CLOUD, *Qu'est-ce que l'Internet des objets (IoT) ?* Site Web. Disponible à l'adresse : <https://www.ovhcloud.com/fr/learn/what-is-iot/>, Consulté le : 2024-01-06, 2024.
- [23] ORANGE BUSINESS SERVICES, *Qu'est-ce qu'un objet connecté ?* Site Web. Disponible à l'adresse : <https://iotjourney.orange.com/fr-FR/explorer/les-solutions-iot/objet-connecte>, Consulté le : 2024-01-06, 2024.

Bibliographie

- [24] SAP SE, *Qu'est-ce que l'Internet des Objets (IoT) ?* Site Web. Disponible à l'adresse : <https://www.sap.com/france/products/artificial-intelligence/what-is-iot.html>, Consulté le : 2024-02-10.
- [25] I. KHAN et A. RASHID, « Internet of Things (IoT) Its Applications and Enabling Technologies in Healthcare : A Review, » *Educational Administration : Theory and Practice*, t. 30, p. 9699-9714, 2024. DOI : [10.53555/kuey.v30i4.5031](https://doi.org/10.53555/kuey.v30i4.5031).
- [26] V. U. IHEKORONYE, C. I. NWAKANMA, G. O. ANYANWU, D.-S. KIM et J.-M. LEE, « Benefits, Challenges and Practical Concerns of IoT for Smart Manufacturing, » in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, 2021, p. 827-830. DOI : [10.1109/ICTC52510.2021.9620771](https://doi.org/10.1109/ICTC52510.2021.9620771).
- [27] L. M. JIMÉNEZ, *IoT : Technologies, Cas d'usages, avantages et limites - Guide complet*, Site Web. Disponible à l'adresse : <https://blog.ubisolutions.net/iot-technologies-cas-dusages-avantages-et-limites-guide-complet>, Consulté le : 2024-02-16, 2023.
- [28] D. B. PATEL, R. TRIPATHI et N. GOEL, « IoT an Overview : Advantage, Disadvantage and Applications, » *International Journal of Computer Applications Technology and Research*, t. 10, p. 119-122, 2021. DOI : [10.7753/IJCATR1005.1003](https://doi.org/10.7753/IJCATR1005.1003).
- [29] M. SARIKA, V. KOTAK et A. DURAFE, « A Review Paper on Internet of Things(IoT) and its Applications, » *International Research Journal of Engineering and Technology (IRJET)*, t. 6, p. 1623, 2019.
- [30] ESKIMON et OLYTE, *Arduino : premiers pas en informatique embarquée*. 2023, p. 642. adresse : <https://eskimon.fr/extra/ebooks/arduino-premiers-pas-en-informatique-embarquee.pdf>.
- [31] A. S. ISMAILOV, Z. B. JO'RAYEV et al., « Study of Arduino Microcontroller Board, » *Science and Education*, t. 3, p. 172-179, 2022.
- [32] ARDUINO DOCS, *Arduino UNO R3*, Fiche technique, 2024. adresse : <https://docs.arduino.cc/hardware/uno-rev3/>.
- [33] INFOFORALL, *Arduino*, Site Web. Disponible à l'adresse : <https://infoforall.fr/arduino/arduino-accueil.html>. Consulté le : 2024-07-09.
- [34] ARDUINO DOCS, *Arduino Mega 2560 Rev3*, Fiche technique, 2024. adresse : <https://docs.arduino.cc/hardware/mega-2560/>.
- [35] ARDUINO DOCS, *ARDUINO MEGA2560 ADK*, Fiche technique, 2015. adresse : <https://docs.arduino.cc/retired/boards/arduino-mega-adk-rev3/>.
- [36] ARDUINO DOCS, *Arduino Due*, Fiche technique, 2024. adresse : <https://docs.arduino.cc/hardware/due/>.
- [37] HARDWARE LIBRE, *Arduino Due*, Site Web. Disponible à l'adresse : <https://www.hwlibre.com/fr/arduino-en-raison/>. Consulté le : 2024-07-09.
- [38] ARDUINO DOCS, *Arduino Nano*, Fiche technique, 2024. adresse : <https://docs.arduino.cc/hardware/nano/>.

Bibliographie

- [39] DISTRELEC, *Arduino Nano Board*, *Arduino*, Site Web. Disponible à l'adresse : <https://www.distrelec.de/en/arduino-nano-board-arduino-a000005/p/11096733>. Consulté le : 2024-07-09.
- [40] A. ISLAM, N. KAUR, F. AHMAD et S. PICHANDI, « Intelligent Wireless Fire Extinguishing Robot, » *International Journal of Current Engineering and Technology*, t. 6, p. 520-526, 2016.
- [41] ARDUINO DOCS, *Arduino Yún*, Site Web. Disponible à l'adresse : <https://docs.arduino.cc/retired/boards/arduino-yun/>. Consulté le : 2024-07-09.
- [42] M. FEZARI et A. AL DAHOUD, « Integrated development environment "IDE" for Arduino, » *WSN applications*, t. 11, p. 1-12, 2018.
- [43] ARDUINO BLAISEPASCAL, *L'IDE Arduino*, Site Web. Disponible à l'adresse : <https://arduino.blaisepascal.fr/presentation/logiciel/>. Consulté le : 2024-07-09.
- [44] K. SÖDERBY et J. HYLÉN, *Getting Started with Arduino IDE 2*, Site Web. Disponible à l'adresse : <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2/>, Consulté le : 2024-02-4.
- [45] SENSMONITOR, *Use Cases : Mobile Beehives*, Site Web. Disponible à l'adresse : <https://sensmonitor.com/use-cases/mobile-beehives/>, Consulté le : 2024-01-20, 2023.
- [46] G. SOPHIE, *Les ruches connectées BeeGuard au service des apiculteurs*, Site Web. Disponible à l'adresse : <https://laplateformedumiel.fr/ruches-connectees-beeguard/>, Consulté le : 2024-01-20, 2024.
- [47] V. K. UGAH et C. J. NNONYELU, « A Wemos-D1-R2-Based Remote-Switching Module for Home Internet of Things Applications, » in *LGT-UNN 1st International Multidisciplinary Conference on Technology*, University of Nigeria Nsukka, t. 1, 2019, p. 1-6.
- [48] P. SUBRAMANIAN et A. DAVID, « Sensors -An Overview, » in *International Conference on Multidisciplinary Research and Innovative Practices*, J.S. University, Shikohabad, Firozabad (U.P), 2021, p. 696-702.
- [49] A. DE MARCELLIS et G. FERRI, « Physical and Chemical Sensors, » *Analog Circuits and Systems for Voltage-Mode and Current-Mode Sensor Interfacing Applications*, p. 1-35, 2011. DOI : [10.1007/978-90-481-9828-3_1](https://doi.org/10.1007/978-90-481-9828-3_1).
- [50] B. C. PATEL, G. SINHA et N. GOEL, *Introduction to sensors*. IOP Publishing Bristol, UK, 2020, p. 1-21, ISBN : 978-0-7503-2707-7. DOI : [10.1088/978-0-7503-2707-7ch1](https://doi.org/10.1088/978-0-7503-2707-7ch1).
- [51] MYMAXICOURS, *Capteurs et actionneurs*, Site Web, adresse : <https://www.maxicours.com/se/cours/capteurs-et-actionneurs/>, Consulté le : 2024-04-09.
- [52] TECHNOLOGIE AU COLLÈGE, *Exemples d'actionneurs*, Site Web. Disponible à l'adresse : http://technologieaucollege.free.fr/ressources_web/ressources2.techno.free.fr/mecanique/systemes/actionneur.htm, Consulté le : 2024-04-09.

Bibliographie

- [53] B. MIHAI, « How to Use the DHT22 Sensor for Measuring Temperature and Humidity with the Arduino Board, » *ACTA Universitatis Cibiniensis*, t. 68, 2016. DOI : [10.1515/aucts-2016-0005](https://doi.org/10.1515/aucts-2016-0005).
- [54] SAINSMART, *DHT22/AM2302 Digital Temperature Humidity Measurement sensor*, Site Web. Disponible à l'adresse : <https://sainsmart.com/products/dht22-am2302-digital-temperature-humidity-measurement-sensor>. Consulté le : 2024-07-09.
- [55] A. PUTRI, « FOOD WARMER SYSTEM BASED ON DHT-22, » *JAREE (Journal on Advanced Research in Electrical Engineering)*, t. 4, 2020. DOI : [10.12962/j25796216.v4.i1.118](https://doi.org/10.12962/j25796216.v4.i1.118).
- [56] S. DAS, A. KARMAKAR, P. DAS et B. KOLEY, « Manufacture of electronic weighing machine using load cell, » *IOSR Journal of Electrical and Electronics Engineering*, t. 14, p. 32-37, 2019. DOI : [10.9790/1676-1404013237](https://doi.org/10.9790/1676-1404013237).
- [57] A.-M. SARAH et W. ALEX, *Load cell amplifier HX711 breakout hookup guide*, Tutoriel. Disponible à : <https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookupguide/all>, 2016.
- [58] MHTRONIC, *Capteur de pression à jauge contraire – 50kg*, Site Web. Disponible à l'adresse : <https://mhtronic.com/produit/capteur-de-pression-a-jauge-contre-50kg/>. Consulté le : 2024-07-09.
- [59] ZEMIC EUROPE, *Type 2N-50kg Full-bridge Load Cell*, Fiche technique, 2019. adresse : https://www.zemiceurope.com/media/Documentation/2N_50kg_Full_Bridge.pdf.
- [60] CIRCUIT JOURNAL, *50kg Load Cells with HX711*, Site Web. Disponible à l'adresse : <https://circuitjournal.com/50kg-load-cells-with-HX711>. Consulté le : 2024-05-08.
- [61] A. SEMICONDUCTOR, « 24-bit analog-to-digital converter (adc) for weigh scales, » *Hx711*, t. 9530, p. 1-9, 2016.
- [62] ARDUINO FRANCE, *Branchement capteur de son Arduino (KY-038)*, Site Web. Disponible à l'adresse : <https://arduino-france.site/capteur-son/>, Consulté le : 2024-04-09, 2021.
- [63] JOY-IT, *KY-038 Module capteur de son avec microphone*, Fiche technique, 2017.
- [64] S. SAFDAR, A. ZEB, A. KHAN et Z. KALEEM, « Android based vehicle tracking system, » *EAI Endorsed Transactions on Energy Web*, t. 5, p. 154-447, 2018. DOI : [10.4108/eai.10-4-2018.154447](https://doi.org/10.4108/eai.10-4-2018.154447).
- [65] ISAAC, *HC-SR04 : tout sur le capteur à ultrasons*, Site Web. Disponible à l'adresse : <http://hc-sr04.com>. Consulté le : 2024-07-09.
- [66] S. SAFDAR, A. ZEB, A. KHAN et Z. KALEEM, « Android based vehicle tracking system, » *EAI Endorsed Transactions on Energy Web*, t. 5, p. 154-447, 2018. DOI : [10.4108/eai.10-4-2018.154447](https://doi.org/10.4108/eai.10-4-2018.154447).
- [67] SIMCOM, *SIM808 Hardware Design V1.03*, Document technique, 2016. adresse : http://www.simcom.com/documents/SIM808_Hardware_Design_V1.03.pdf.

Bibliographie

- [68] SIMCOM, *SIM808 SPEC_V1507*, Fiche technique, 2015. adresse : <https://simcom.ee/documents/?dir=SIM808>.
- [69] ADIY, *Flame Sensor Module*, Site Web. Disponible à l'adresse : <https://adiy.in/shop/flame-sensor-module/>. Consulté le : 2024-04-09, 2024.
- [70] KUONGSHUN ELECTRONIC, *IR Infrared Flame Detection Sensor*, Site Web. Disponible à l'adresse : <https://www.kuongshun-ks.com/uno/uno-sensor/hot-sell-ir-infrared-flame-detection-sensor.html>. Consulté le : 2024-07-09.
- [71] R. MUSTAFA, B. AHMEDI et K. MUSTAFA, « Digitalization of the Parking Lot at the Public University “Kadri Zeka” in Gjilan, » *International Journal of Recent Contributions from Engineering, Science IT (iJES)*, t. 9, p. 54, 2021. DOI : [10.3991/ijes.v9i3.24219](https://doi.org/10.3991/ijes.v9i3.24219).
- [72] SURPLUSTEK, *PureVolt Breadboard Solderless 1660 Points*, Site Web. Disponible à l'adresse : <https://surplustek.ca/en/purevolt-breadboard-solderless-1660-points>, Consulté le : 2024-07-09.
- [73] JOY-IT, *16x2 LCD MODULE*, Site Web. Disponible à l'adresse : <https://www.joy-it.net/en/products/SBC-LCD16x2>. Consulté le : 2024-07-09.
- [74] PICCLICK, *Iic / I2C / Twi /Spi Serial Interface Board Module Port pour Arduino 1602LCD*, Site Web. Disponible à l'adresse : <https://picclick.fr/Iic-I2C-Twi-Spi-Serial-Interface-235371791749.html>. Consulté le : 2024-07-09.
- [75] BLYNK.DOCUMENTATION, *Introduction*, Site Web. Disponible à l'adresse : <https://docs.blynk.io/en>, Consulté le : 2024-04-09.
- [76] M. LAKSHMI, C. SAMEEN, D. MANEESHA et al., « Smart Home using Blynk App Based On IOT, » *International Journal of Creative Research Thoughts (IJCRT)*, t. 10, p. d341-345, 2022.
- [77] K. K. S. LIYAKAT, « Blynk IoT-Powered Water Pump-Based Smart Farming, » *Recent Trends in Semiconductor and Sensor Technology*, t. 1, p. 8-14, 2024.
- [78] THIERRY VAIRA, *Blynk*, Site Web. Disponible à l'adresse : <http://tvaira.free.fr/dev/tutoriel/blynk.html>. Consulté le : 2024-07-09.

Résumé

L'Internet des objets s'intègre de plus en plus à notre quotidien, touchant de nombreux secteurs, y compris l'apiculture. Face aux nombreuses contraintes rencontrées par les apiculteurs, notre projet de ruche connectée vise à alléger leur charge de travail et à faciliter la gestion des ruches.

Ce mémoire présente un système de surveillance en temps réel d'une ruche d'abeilles, utilisant des capteurs de température, d'humidité, de poids et de son pour surveiller les paramètres de santé de la ruche. Pour renforcer la sécurité, nous incluons également un capteur de distance ultrasonique, un capteur de flammes et un module GSM/GPRS/GPS. Les données collectées sont traitées par un microcontrôleur (WeMos D1 R2) et envoyées via Wi-Fi à un serveur Blynk. Ce dernier transforme les données en tableaux et graphiques, accessibles via l'application Blynk, qui génère des alertes en cas de changements inattendus. Un prototype basé sur ce système est également construit pour améliorer la gestion des ruches.

Mots clés : Internet des objets (IdO), Ruche connectée, Arduino, Serveur Blynk, Capteur.

Abstract

The Internet of Things is increasingly integrating into our daily lives, impacting many sectors, including beekeeping. Faced with the numerous challenges experienced by beekeepers, our connected hive project aims to lighten their workload and facilitate hive management.

This thesis presents a real-time hive monitoring system, utilizing temperature, humidity, weight, and sound sensors to monitor the hive's health parameters. To enhance security, we also include an ultrasonic distance sensor, a flame sensor, and a GSM/GPRS/GPS module. The collected data is processed by a microcontroller (WeMos D1 R2) and sent via Wi-Fi to a Blynk server. This server converts the data into tables and graphs, accessible through the Blynk application, which generates alerts in case of unexpected changes. A prototype based on this system is also built to enhance hive management.

Keywords : Internet of Things (IoT), Connected Hive, Arduino, Blynk Server, Sensor
