PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
IBN KHALDOUN UNIVERSITY OF TIARET
FACULTY OF APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL ENGINEERING

GRADUATION MEMOIR

To obtain

Master's Degree

In Embedded Systems

Theme

Air-Conditioned Beehive with Central Heating

Submitted by

RADJEL Samia
ZAHI Ayoub Abderrahmene

Presented on 26/ 06 / 2024 in front of the jury composed of

KOUADRIA Mohamed	MCB	University of Tiaret	Chairman
MAAMAR Noureddine	MAB	University of Tiaret	Examiner
BEZZEROUK Mohamed Amine	MCA	University of Tiaret	Supervisor
SAFA Ahmed	MCA	University of Tiaret	Co-Supervisor

Academic Year 2023 / 2024

Acknowledgements

First and foremost, we would like to express our thanks and gratitude to Allah who granted us health, patience and support we need to finish this work.

We would like to thank, Dr. Bezzerouk Amine and Dr. Safa Ahmed for accepting to supervise us and for their outstanding support, guidance and encouragement through all our senior period of work, our heartfelt thanks also go to the members of the jury who agreed to read and evaluate this work.

We would like also to thank everyone who helped us from close and far, and a special greeting to every professor in the electrical engineering department, the LTI laboratory members for sharing their space of work with us and for all the mental support.

Dedications

First things first I would like to express my thankfulness and gratefulness to Allah for guiding and lighting my way and giving me enough power and patience along these years. I could not be more delighted to dedicate this work to my beautiful parents, Mamitous and Daddy, to my sisters and brothers, my nieces and my nephews in particular my little mignonette Hanou, to all my friends, particularly Yassine El Guerri, and family for their enduring love and unwavering encouragement

For two of my most special and precious persons, Mr Belhadji Youcef, who did not hesitate for once to show me all sorts of guidance and being a source of positivity even in the darkest of times, and for Mr Otmani Radouane, the one and only who has believed in me since day one, words are not enough to express how much I was lucky to have him as a source of support and inspiration throughout the past 3 years. This achievement would not have been possible without each of you.

To all the Electrical engineering department, my colleagues whom I passed with them this beautiful journey, my teachers with their insightful advice… a huge Thank You!

RADJEL Samia

Dédicaces

Je dédie ce modeste travail :

A la mémoire de mon grand-père, je ne sais pas comment exprimer ma tristesse face à ton absence. J'aurais aimé que tu sois à mes côtés et que tu sois fière de ton petit fils.

A ma précieuse offre du dieu, l'épaule solide, l'œil attentif compréhensif, et mon soutien moral et la source de ma joie « mon père ».

A la femme qui a souffert sans me laisser souffrir, le soleil de mes jours et la lune de mes nuits, la personne qui m'a encouragé dans chaque pas « ma mère ».

A mes sœurs : -Ouezna, Manel, Hadjer vous êtes un précieuses dans ma vie, vous m'avez toujours donné le courage pour continuer. Merci

A mes grandes mères et mon grand-père, l'amour éternel que dieu vous gardes inchaallah

ZAHI Ayoub Abderrahmene

Contents

List of Figures

List of Acronyms

MQTT: Message Queuing Telemetry Transport

ISO: International Organization for Standardisation

IDE: Integrated Development Environment

OS: Operating System

ML: Machine Learning

UI: User Interface

ABCH: Air conditioner Beehive with Central Heating

LAN: Local Area Network

CPU: Central Processing Unit

AI: Artificial Intelligence

IoT: Internet of Things

WiFi: Wireless Fidelity

General Introduction

Beekeeping or apiculture entails the rearing or keeping of bees with the aim of exploiting its products (such as honey, pollen grain, propolis, and brood), honey products generate multiple market opportunities, and are also nutritious foods. In addition, the production process is not in competition with another form of agriculture and it can be integrated positively.

The same to that, honey production is also considered as a natural resource conserving and environmentally friendly activity through its plant pollination services. Thus, it should be one of the most important intervention areas for sustainable development among countries such as our country Algeria. But however, Honey creation faces a dilemma.

Honey bee wellbeing is declining because of dangers like parasites, infections, and absence of blooming plants from territory misfortune. Pesticides utilised in horticulture can additionally hurt them. These difficulties debilitate honey bee provinces, prompting less honey and even state breakdown. While beekeeping rehearses themselves can likewise be distressing for honey bees, the greatest buzzkill is the decrease in honey bee wellbeing welcomed on by natural changes. Beside these circumstances, another major reason why the honey production is keeping decreasing , Temperature , this important condition for bees' well-being must be guaranteed if we are interested about such projects , Honey bees are actually incredibly resourceful at maintaining their hive temperature themselves, but the problematic here we are now trying to solve in our graduation memoir is how to make an automatic system that helps the bees to keep the temperature inside the hive in the right level so can the bees keep working on honey's production instead of wasting tens of bees in trying to keep that work .

Air conditioner beehive with central heating is the unique and modern idea which is the solution we are going to explain and to propose in order to solve this problem. The project incorporates sensors such as humidity, temperature, and actuators like lamps, heaters, and water pumps. and the Python programming language is used for implementation, while MQTT protocol is employed for communication. The Raspberry Pi Pico acts as the broker for data exchange. The project aims to provide real-time monitoring of environmental parameters and enable remote control of actuators The utilisation of MQTT convention guarantees productive and dependable correspondence between gadgets. MQTT's lightweight nature makes it ideal for most of modern applications, as it limits transfer speed and power utilisation.

The Raspberry Pi goes about as the focal specialist, working with consistent information trade between sensors, actuators, and the UI. This Master thesis is organised in three chapters, the first chapter formed to introduce us to the ACBCH by presenting generalities on that last, these generalities talk about how the life in a beehive is organised, job of each part of it “queen, workers and drones», also we’re going to give a detailed explanation about beehives and the perfect conditions for them so the bees find the proper environment to work and the second chapter a hardware description used in this project. Finally, in the third chapter, system implementation.

CHAPTER I

Generalities on the Air Conditioner Beehive with Central Heating

I.1. Introduction

In this chapter, we delve into the critical aspects of maintaining optimal conditions within beehives, focusing on temperature control as a cornerstone of colony health. Building upon the foundation laid out in the preceding discussion, we explore the innovative concept of the Air Conditioner Beehive with Central Heating (ABCH). This chapter examines the practical implications and advantages of such a system for enhancing bee welfare. Furthermore, we evaluate different types of beehives, including the Dadant and Langstroth models, to illuminate key considerations in adopting advanced hive management technologies

I.2. Generalities

Bees have an impact on pollinating crops and maintaining harmony. Yet the prosperity and health of bee communities are intricately linked to elements, temperature. Maintaining conditions within beehives is crucial, for supporting colony well-being and efficiency. These fascinating creatures boast complex social structures and live in intricately designed homes known as beehives. These beehives were made with random materials, particularly from materials that were available locally, like hollowed-out logs, woven grass, etc.

Today, most beehives are built from wood and designed in a manner that it can be easily managed and controlled by beekeepers, and if we have to specify a category, we have to pick the “Dadant hive” and the “Langstroth” one, who are now the example of the modern hives that are legalised by the ISO [1].

The initial chapter is focused on introducing the idea of the Air Conditioner Beehive with Central Heating (ABCH) delving into information about beehives addressing concerns regarding temperature control and outlining a thorough description of the suggested system while highlighting its advantages for bees. Furthermore, this chapter will touch on types of beehives to determine factors, for selecting the ABCH system.

I.3. Exploring bees’ community

As known, inside a beehive there are three (3) main types of bees:

I.3.1. Queen

There is only one queen in a colony. It is considerably larger than the members of other castes. Her wings are much shorter in proportion to her body. Because of her long tapering abdomen, it appears more wasp-like than other inmates of the colony. The queen is the only individual which lay eggs in a colony and is the mother of all bees. It lays up to *2000* eggs per day in *Apis mellifera* [2] . Five to ten days after emergence, she mates with drones in one or more nuptial flights. When her sperm theca is filled with sperms, she will start laying eggs and will not make any more. She lives for 3 years [1]. The secretion from mandibular gland of the queen is called queen's substance. The queen substance if present in sufficient quantity, prevent swarming and absconding of colonies, prevent development of ovary in workers, and maintains colony cohesion [3]. The queen can lay either fertilized or sterile eggs depending on the requirement.

The differentiation in worker and queen is due to the quantity and quality of food fed to the larva. The larva which becomes the queen is fed the *royal jelly*, a secretion from hypo pharyngeal glands of the worker bees. The queen is reared in large finger-shaped cells in the lower portion of the combs. Only one queen can remain in a colony, but during unfavourable season two queens are also observed. The old queen is killed as soon as the new queen is fertilised [3].

Generally, queens are reared only during swarming season, but if the queen dies accidentally the bees can rear a new queen. The phenomenon of raising queen in off-season is called supersedure [4]. There is a good family planning in the colony. The number of eggs and egg laying depend on the availability of pollen and nectar in nature. If the food is scarce, workers do not permit the queen to lay eggs. The queen is carefully, looked after by young workers, known as-attendants, which feed her and keep her clean and well combed. The queen never leaves the, hive except with a swarm [2].

I.3.2. Workers

The workers are the smallest inhabitants of the beehive. They form the bulk of the population. The number of workers in a colony varies from 1,500 to 50,000[1]. They are imperfect females, incapable of laying eggs. On certain occasions when the colony is in need of a queen, some of the workers start laying eggs from which only drones are produced. These workers, called *laying workers*, are killed as soon as a new queen is introduced or produced in the colony [4].

The life-span of a worker is about 4 weeks during active season and 8 to 10 weeks during less active season. Their range of flight varies from 1,000 to 1,500 m [1]. The division of work within a colony among the worker bees is based on the age of the individual and on the needs of the colony [1]. Normally, the young bees, immediately after their emergence, do the work of cleaning cells and feeding older larvae. When they are grown and their hypo pharyngeal glands have developed, they secrete the royal jelly with which they feed the younger larvae. These bees are called *nurse-bees [2]*. For the first 2 to 18 days of their life, the bees perform indoor duty inside the hive, including comb construction when some young bees start secreting wax. Later on they become foragers, collect water, pollen, nectar and propolis (beeblue). Pollen is a nitrogenous food and is essential for brood - rearing and young bees. Bees wax, of which the comb is made, is a secretion of the wax glands located in the abdomen of the worker bees. For producing 1 kg of wax the bees consume 10 kg of honey [1].

Thus, the lifespan of workers can be divided into two phases as first three weeks for house hold duty and rest of the life for outdoor duty. [2]

A. Household duties [2]

- Build comb with wax secretion from wax glands.
- Feed the young larvae with royal jelly secreted from hypo pharyngeal gland.
- Feed older larvae with bee-bread, a mixture of pollen and honey
- Feeding and attending queen.
- Feeding drones.
- Cleaning, ventilating and cooling the hive.
- Guarding the hive.
- Evaporating nectar and storing honey.

B. Outdoor duties [2]

- Collecting nectar, pollen, propolis and water.
- Ripening honey in honey stomach.

I.3.3. Drones

The drones are the male bees. They are much larger and stouter than either the queen or the workers although their body is not quite as long as that of the queen [1]. They have no sting; a suitable proboscis for gathering nectar is also absent. They are, therefore, physically incapable for the ordinary work of the hive. Their only function is to impregnate the young queen a task which they are unable to perform until they are about 10 days of age. They also help in maintenance of hive temperature. They go out of the hive only at the mid-day when the weather is warm.

Figure 1. *The three types of bees.*

The number of drones in a colony often is very large amounting to hundreds and sometimes to thousands. The drones are reared and tolerated during the breeding season. They are driven out of the hive to die of starvation before the monsoon and the winter.

The drones are produced by unfertilized eggs of the queen, or by those workers which take up the reproductive function due to the absence of a queen in a colony. The normal life-span of a drone is 57 days. Mating takes place in the open when the queen is in flight. The drone dies in the act or immediately afterwards. Its abdomen has to burst open to allow the genital organ to function

I.4. Honeybees:

Here are some types of honeybees we find regionally

I.4.1. Giant Honeybee *(Apis dorsata)*

Found in South and Southeast Asia, *Apis dorsata* is known for its large size and distinctive open hanging nests. These bees build their nests on high tree branches, and their colonies can contain thousands of individuals. They are good pollinators and the wax obtained from these bees are called *Ghedda* wax. Figure (I.2)

Figure 2. *Giant honeybee.*

I.4.2. Little Honeybee *(Apis florea)*

Also known as the dwarf honeybee, *Apis florea* is the smallest of the honeybee species. These bees are found in parts of Asia and the Middle East. They build small, exposed nests, often in shrubs or trees. They are efficient pollinators and honey producers (Fig. 1.3).

Figure 3. *Little honeybee.*

I.5. Generalities about beehives

A detailed description summarised to get a deep knowledge about beehives

I.5.1. History of beehives

While bees exist in diverse species, most construct hives. These structures, marvels of natural engineering, provide shelter, regulate temperature, and offer protection from predators. Throughout history people have tamed bees to harvest honey and assist in pollination. Beekeeping practices trace back, to times with beehives crafted from clay or straw to provide shelter for these creatures. In a beehive you'll usually find components, like the main body, frames and a roof. These parts create a space for bees to construct their combs and stash away their honey and pollen.

I.5.2. Beyond the Honeycomb

The hive interior is a masterpiece of organization. Hexagonal honeycomb cells, built from beeswax, serve as nurseries for developing bees and storage units for food. Worker bees maintain an intricate ventilation system with their wings, ensuring optimal temperature and humidity.

I.5.3. Types of beehives

While bees exist in diverse species, most construct hives. These structures, marvels of natural engineering, provide shelter, regulate temperature, and offer protection from predators. Let's peek at some Common types:

a) Langstroth Hive:

The most popular design, featuring stacked boxes that allow easy access for inspection and honey harvesting.

b) Top Bar Hive:

Simpler and closer to bees' natural cavities, with removable bars for honey collection.

c) Warre Hive:

Mimics natural cavities with square boxes stacked on top of each other, promoting natural beekeeping practices.

d) Horizontal Hive (Layens Hive):

Long, rectangular boxes laid out horizontally, offering bees more space and mimicking fallen logs they might use in nature.

e) WBC Hive (Wye Valley Beekeeping Club Hive):

Similar to the Langstroth but with smaller boxes, promoting natural swarm control.

f) Dadant hive:

the Dadant hive is valued for its capacity to support strong colonies and maximize honey production, making it a preferred choice for beekeepers aiming for high productivity in their apiaries.

I.5.4. Major used beehives in the world

here are the most used beehives in the world

A. The Langstroth:[1]

The Langstroth beehive is the most widely used beehive design in the world, named after its inventor, Reverend Lorenzo Langstroth. It's a vertical, modular hive consisting of several stacked boxes that house removable frames where bees build their honeycomb. It's known for its bee-friendly design, ease of management, and adaptability, making it a popular choice for both hobbyist and commercial beekeepers. Therefore, our project is based on that type of beehives in order to complete the best experience. Here are the key features of a Langstroth beehive[28]

a) Bee space:

This crucial gap, about 8mm wide, prevents bees from attaching comb to hive walls, allowing beekeepers to easily inspect and manage the colony without harming the bees.

b) Removable frames:

Each box contains several frames where bees build comb. These frames can be easily removed for inspection, honey harvesting, and pest control.

c) Modular design:

Boxes can be added or removed as needed to accommodate the growing colony.

d) Queen excluder:

This optional insert placed between brood and honey boxes allows honey collection without the queen laying eggs in honey frames.

Figure 4. *Langstroth hive.*

B. Benefits of using Langstroth

Here are some of the benefits of using a Langstroth beehive:[1]

- **Easy to inspect and manage:** The removable frames allow beekeepers to easily monitor the health of the colony and address any problems quickly.
- **Efficient honey harvesting:** Honey can be harvested without disturbing the brood or damaging the comb.
- **Adaptable to different climates and beekeeping styles:** The modular design allows beekeepers to customize the hive to their specific needs.
- **Widely available:** Langstroth hives and equipment are readily available from beekeeping suppliers. However, there are also some potential drawbacks to be considered:[1]
- **Initial cost:** Langstroth hives can be more expensive than some other types of hives.
- **Weight**: Larger hives can be heavy and difficult to move.
- **Learning curve:** Beekeeping requires some knowledge and experience, and managing a Langstroth hive is no exception. Overall, the Langstroth beehive is a versatile and well-designed hive that is a great choice for beekeepers of all levels.

A- 1 Dadant beehive

The Dadant beehive, named after its inventor Charles Dadant, is a highly popular model of movable-frame hive used in beekeeping. It’s simple and efficient design promotes the development of bee colonies and honey production.

C. Key features of the Dadant beehive:

here some key features we picked for the dadant hive

- **Movable frames:** Just like the Langstroth hive, the Dadant hive utilises removable frames that allow easy manipulation of honeycomb for inspection, harvesting, and colony management.
- **Large size:** Compared to the Langstroth, the Dadant hive offers more generous space for bees, favouring their development and honey production. This larger size, however, makes them heavier comparing to the langstroth hives. [28]
- **Supers:** Similar to the Langstroth, the Dadant hive uses additional boxes called supers stacked on top of the brood chamber. These supers provide extra space for honey storage as production increases. [28]
- **Materials:** Traditionally, Dadant hives are constructed from wood, providing natural insulation and good durability. However, other materials like polystyrene are also becoming increasingly popular. [28]

Figure 5. *Dadant beehive frame.*

D. Advantages of the Dadant beehive

The dadant hive advantages are

- **Ease of use:** The manipulation of frames is straightforward and beginner-friendly. [28]
- **Flexibility:** The modular design allows for adjustments based on colony needs and seasonal variations. [28]
- **High honey production:** The ample space and internal organization contribute to significant honey yields.[28]
- **Convenient harvesting:** Extracting honey is simplified thanks to the removable frames. [28]

E. Disadvantages of the Dadant beehive:

In what follows, we are going to site what are the major disadvantages of using the dadant hive.

- **Cost:** The Dadant hive can be more expensive than other models, particularly due to its larger size and material choices.
- **Weight:** Handling full honey supers can be challenging, sometimes requiring tools for assistance.
- **Bulkiness:** The imposing size of the Dadant hive might limit its placement in certain areas.[28]

F. Variations of the Dadant beehive

The Dadant hive has several variations concerning it

- **Dadant 10 frames:** The most common version, offering a good balance between capacity and manageability.
- **Dadant 8 frames:** More compact and lighter, suitable for smaller spaces and less populous colonies.
- **Dadant 12 frames:** Provides increased storage capacity for highly productive colonies. [1]

I.6. Bees’ most important conditions inside a beehive

inside the beehive, we make sure that these conditions are guaranteed for the bees’ wellbeing, in another perspective, how to ensure bees’ life

I.6.1. importance of temperature in a beehive

Maintaining the temperature, inside the hive is vital, for the growth of baby bees ensuring that the worker bees can do their job effectively and keeping the honey safe. Bees are creatures that control the temperature by airing out the hive and huddling together to stay warm. Temperature is crucial for a beehive for a couple of key reasons:

a) Healthy Brood Development:

Honeybees maintain a very specific temperature range, around 34.5°C to 35.5°C (94°F to 96°F), in the centre of the hive where the brood (developing bees) are located. This warmth is essential for the eggs to hatch and the larvae to develop properly. If the temperature strays too far from this ideal range, the brood development can be slowed or even stopped, leading to a weaker colony. [9]

b) Overall Bee Health:

Besides the brood, the proper temperature also helps regulate the activity level and health of the adult bees. They use their wings to fan and circulate air within the hive to maintain this temperature, especially during hot weather. Proper temperature also helps with honey production and other hive functions. [9]

c) Humidity:

Interior a bee sanctuary, the ideal stickiness level for guaranteeing bee well-being ordinarily ranges between 50% to 60%. [9] This extend is vital since:

- **Brood Improvement:** Keeping up satisfactory mugginess makes a difference within the legitimate advancement of bee brood (hatchlings and pupae). [9]
- **Nourishment Capacity:** Bees store honey and dust within the hive, and legitimate mugginess levels offer assistance in protecting these nourishment sources. [9]
- **Ventilation:** Bees direct stickiness through their behaviour and ventilation endeavours. Tall stickiness can lead to dampness build up, which can be inconvenient to the hive.
- **Wellbeing:** Overabundance stickiness can advance parasitic development and contribute to maladies inside the hive. Beekeepers regularly screen mugginess levels within the hive to guarantee they drop inside this ideal run, particularly amid periods of tall mugginess or when natural conditions may influence hive stickiness levels. [9]

I.6.2. Problematic of thermal regulation

Fluctuations in climatic conditions and extreme weather conditions may disturb the natural thermoregulation of beehives, threatening both colony health and productivity; thus, it is important to the necessity of the finding optimal techniques for helping bees to regulate the thermal conditions inside a hive. Integumentary, bees strive to keep the temperature inside the hive stable to ensure optimal conditions for larval development and adult activities, such as wax production or honey and pollen storage. Anything that can threaten the optimal temperature insides a hive can result in thermal stress, making the colony weak.

I.7. Limitations

As it has been established, traditional methods of beehive climate regulation, including natural ventilation and insulation materials, are not effective in terms of preserving stable thermal conditions, particularly considering adverse weather conditions. Therefore, an innovative solution is needed, and the following section presents the Air Conditioner Beehive with Central Heating. The ABCH is an integrated system that uses air conditioning and central heating to maintain the optimal temperature inside the hive. Therefore, the proposed solution will allow for precise control of temperature and ensure bees' thermal comfort throughout the year.

I.8. Conclusion

In this chapter we have deliberately chosen simple and varied definitions for the bees' colony, including all its parts from members to their homes to the conditions which the humans can offer for them in order to guarantee the well going in the apiculture domain

CHAPTER II

Generalities on the Air Conditioner Beehive with Central Heating

II.1. Introduction

In this chapter, we focus on the implementation of a smart monitoring and supervision system specifically designed for hive of bees, there for we explore the utilisation of hardware components, including the Raspberry Pi pico w, and a selection of sensors. These Components are crucial in enabling real-time monitoring and control capabilities within the beeshive.

Originally designed by Raspberry Pi, the Raspberry Pi Pico W is an economical yet flexible development platform for RP2040. It represents a wireless upgraded Raspberry Pi Pico model. It has a 2.4GHz wireless interface in addition to all of Pico's features.

- **18-Monk, S. (2022). *Raspberry Pi Cookbook*. " O'Reilly Media, Inc.".**

(MQTT)The machine-to-machine protocol is used for message queuing services. It must operate via a protocol for communication that provides ordered, lossless, bi-directional connections. It is intended for connections with distant sites that have devices with resource limitations or restricted network capacity, such as devices belonging to the Internet of Things (IoT).

- **10-Hillar, G. C. (2017). *MQTT Essentials-A lightweight IoT protocol*. Packt Publishing Ltd.**

II.2. Raspberry pi pico w

The Raspberry Pi Pico W expands upon the renowned Raspberry Pi Pico line of products by adding wireless connection. With it’s large on-chip memory, symmetric dual-core processor complexe, deterministic bus fabric, and rich peripheral set enhanced by our unique Programmable I/O (PIO) sub-system, the RP2040 silicon platform, the foundation of all Pico products, embodies our signature values of high performance, low cost, and ease of use in the microcontroller space. For professional users, this platform offers unparalleled power and flexibility. With a UF2 bootloader in ROM, a well-polished MicroPython port, and extensive documentation, it provides the lowest entry barrier for hobbyist and novice.

A modern 40nm process node is employed in the Manufacturing of the RP2040, providing excellent performance, minimal dynamic power consumption, minimal leakage, and a range of low-power modes to enable prolonged battery-powered operation

Raspberry Pi Pico W offers 2.4GHz 802.11 b/g/n wireless LAN support and Bluetooth 5.2, with an Onboard antenna, and modular compliance certification. It is able to operate in both station and access-point modes. Full access to network functionality is available to both C and MicroPython developers. Raspberry Pi Pico W pairs RP2040 with 2MB of flash memory, and a power supply chip supporting input voltages from 1.8–5.5V. It provides 26 GPIO pins, three of which can function as analogue inputs, on 0.1”-pitch through-hole pads with castellated edges. Raspberry Pi Pico W is available as an individual unit, or in 480-unit reels for automated assembly [8]

18-Monk, S. (2022). *Raspberry Pi Cookbook*. " O'Reilly Media, Inc.".

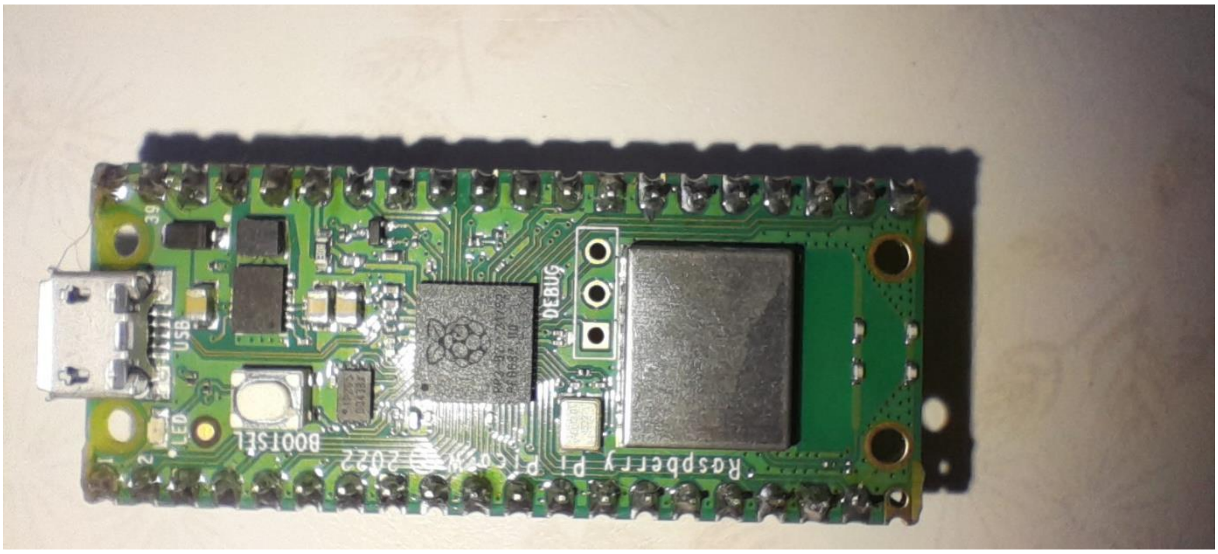

Figure 6. *Raspberry pico pi w.*

II.2.1. Features

- RP2040 microcontroller chip designed by Raspberry Pi in the United Kingdom
- Dual-core Arm Cortex M0+ processor, flexible clock running up to 133 MHz
- Onboard Infineon CYW43439 wireless chip, supports 2.4GHz Wi-Fi
- 264KB of SRAM, and 2MB of Onboard Flash memory
- Castellated module allows soldering directly to carrier Boards
- USB 1.1 with device and host support
- Low-power sleep and dormant modes
- Drag-and-drop programming using mass storage over USB
- 26 × multi-function GPIO pins
- 2 × SPI, 2 × I2C, 2 × UART, 3 × 12-bit ADC, 16 × controllable PWM channels
- Accurate clock and timer on-chip
- Temperature sensor
- Accelerated floating-point libraries on-chip
- 8 × Programmable I/O (PIO) state machines for custom peripheral support

18-Monk, S. (2022). *Raspberry Pi Cookbook*. " O'Reilly Media, Inc.".

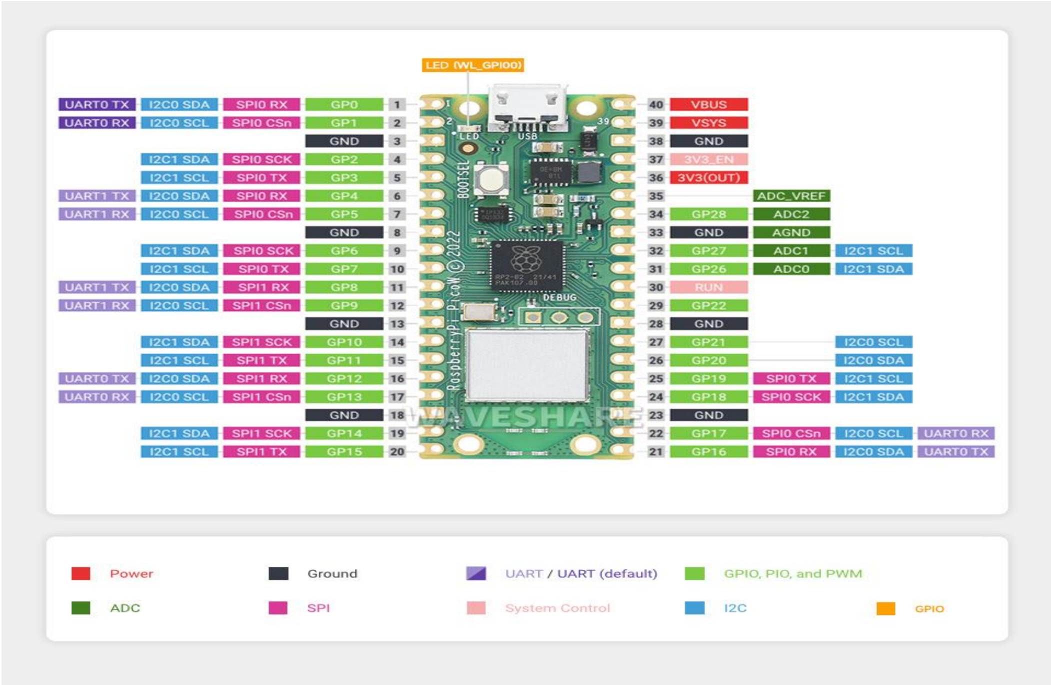

Figure 7. *Raspberry pi pico w pinout.*

https://learn.adafruit.com/getting-started-with-raspberry-pi-pico-circuitpython/pinouts

II.2.2. Specifications

- Form factor : 21 mm × 51 mm
- CPU : Dual-core Arm Cortex-M0+ @ 133MHz
- Memory: 264KB on-chip SRAM ; 2MB Onboard QSPI flash
- Interfacing: 26 GPIO pins, including 3 analogue inputs
- Peripherals
  - • 2 × UART.
  - • 2 × SPI controllers.
  - • 2 × I2C controllers.
  - • 16 × PWM channels.
  - • 1 × USB 1.1 controller and PHY, with host and device support.
  - • 8 × PIO state machines.
- Connectivity 2.4GHz IEEE 802.11b/g/n wireless LAN, Onboard antenna Bluetooth 5.2
  - • Support for Bluetooth LE Central and Peripheral roles
  - • Support for Bluetooth Classic
- Input power : 1.8–5.5V DC
- Operating temperature : -20°C to +70°C
- Production life time: Raspberry Pi Pico W will remain in production until at least January 2034

18-Monk, S. (2022). *Raspberry Pi Cookbook*. " O'Reilly Media, Inc.".

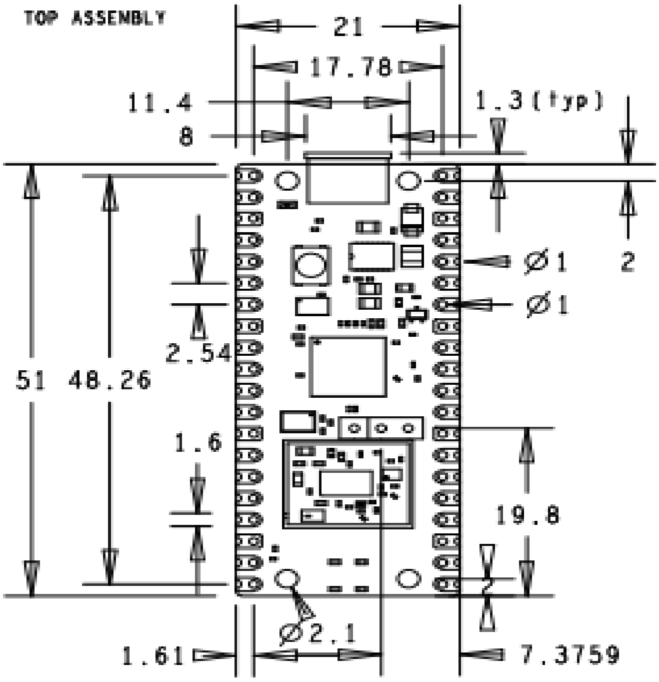

Figure 8. *Physical Specifications.*

Raspberry pi pico W DatasheetAn RP2040-based microcontroller with wireless

II.2.3. Setup and installation

To set ower raspberry pi pico w to work we start by Download the latest version of Raspberry Pi Pico W firmware then connect the small end of ower micro USB cable to the Raspberry while holding down the « BOOTSEL » button,

Connect the other end to ower desktop computer, laptop which the file manager should open up, with Raspberry Pi Pico being show as an externally connected drive as we drag and drop the firmware file we downloaded into the file manager which ower Raspberry Pi Pico should disconnect

We Install thonny, after that we open the Thonny editor it'll show the version of Python that is being used in the bottom right-hand corner of the editor If it does not say 'MicroPython (Raspberry Pi Pico)' there, we click on the text and select 'MicroPython (Raspberry Pi Pico)' from the options

Then we need to install the pico zero library as a Thonny package. In Thonny, choose Tools > Manage packages, complete the projects in the path [8]

18-Monk, S. (2022). *Raspberry Pi Cookbook*. " O'Reilly Media, Inc.".

II.3. MQTT

MQTT is a push-subscribe messaging protocol based on the TCP/IP protocol.

Clients and brokers are the two different kinds of systems in the MQTT architecture. The broker is the server that allows clients to communicate with one another. It takes in messages from clients and forwards them to other clients. Clients always communicate through the broker rather than directly with one another as a result, Every customer may be a subscriber, a publisher, or both.

With the objective to reduce the number of transmissions, MQTT is an event-driven protocol. Data is delivered neither continuously nor at predetermined intervals; clients publish only when they have information to share, and brokers only notify subscribers when they receive new data. [10]

10-Hillar, G. C. (2017). *MQTT Essentials-A lightweight IoT protocol*. Packt Publishing Ltd

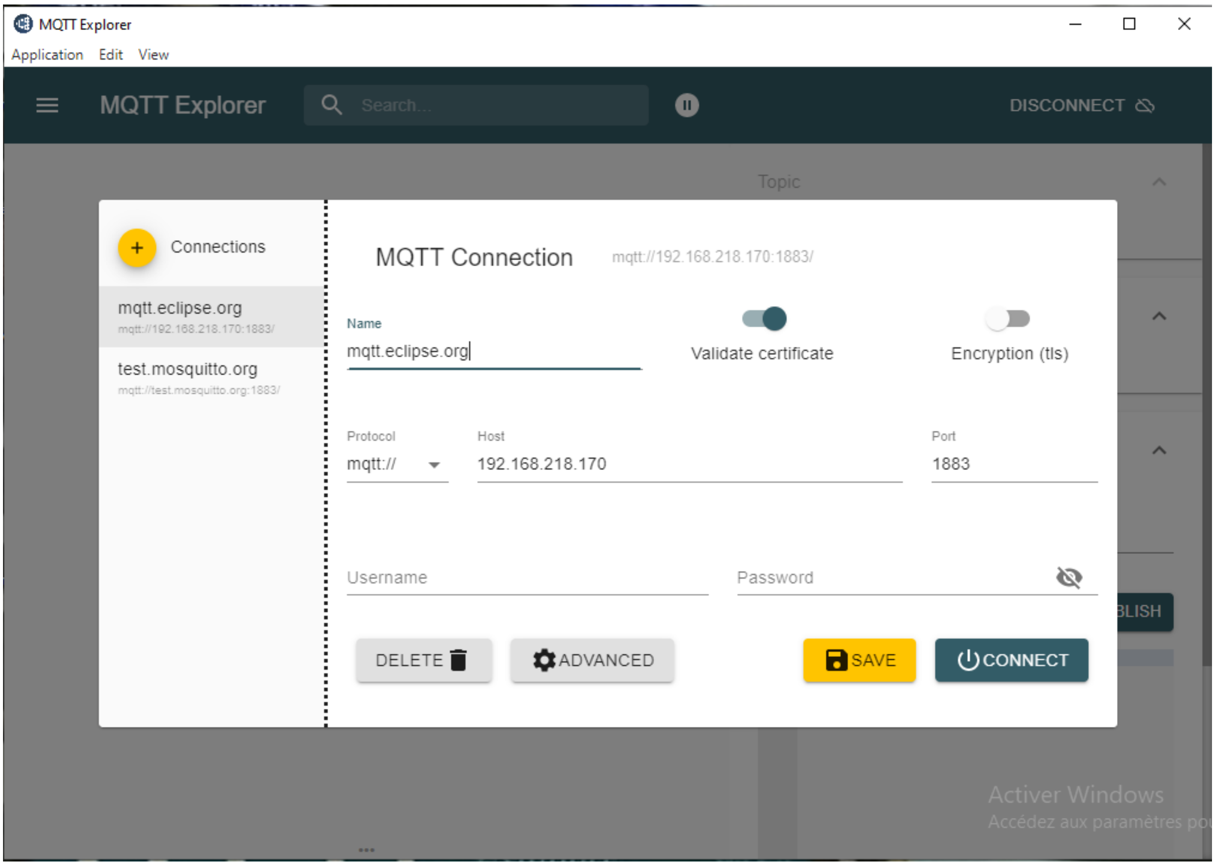

Figure 9. *MQTT Explorer.*

A. Importance

1 Among the IoT protocols now in use, it is one of the lightest. Any type of hardware or software can use this open standard. Since client libraries are available for every major programming languages, creating MQTT-based IoT applications is simple.

2 Because of MQTT's flexibility, various use cases and IoT project architectures can be supported. It should be noted that publishers and subscribers to the system don't even need to be aware of one another because the broker manages all connections.

3 Due to the protocol's ability to construct highly scalable projects, millions of IoT devices may be connected to a system. MQTT's bi-directional communication allows messages to be disseminated to huge groups of devices.

4 A variety of authentications and data security techniques, including TLS encryption, are supported by MQTT. [14]

14-Domínguez-Bolaño, T., Barral, V., Escudero, C. J., & García-Naya, J. A. (2024). An IoT system for a smart campus: Challenges and solutions illustrated over several real-world use cases. *Internet of Things*, *25*, 101099

B. use cases

When IBM first created MQTT, it was intended for use by the oil and gas industry, which Required a way to transmit and receive data from sensors on remote pipelines. After IBM eventually published a MQTT definition, other companies realized that MQTT was useful for a lot of their own IoT workloads outside of the oil and gas sector, and they started developing their own MQTT implementations.

Let's take a look at a handful of industries that are using MQTT :

- Smart homes and wearable devices.
- Manufacturing.
- Automotive.
- Retail.
- Logistics.
- Smart Cities.
- Agriculture.
- Mobile application development. [14] [15]

14-Domínguez-Bolaño, T., Barral, V., Escudero, C. J., & García-Naya, J. A. (2024). An IoT system for a smart campus: Challenges and solutions illustrated over several real-world use cases. *Internet of Things*, *25*, 101099.

15-Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., & Al-Hatmi, R. (2017, May). Internet of Things: Survey and open issues of MQTT protocol. In *2017 international conference on engineering & MIS (ICEMIS)* (pp. 1-6). Ieee.

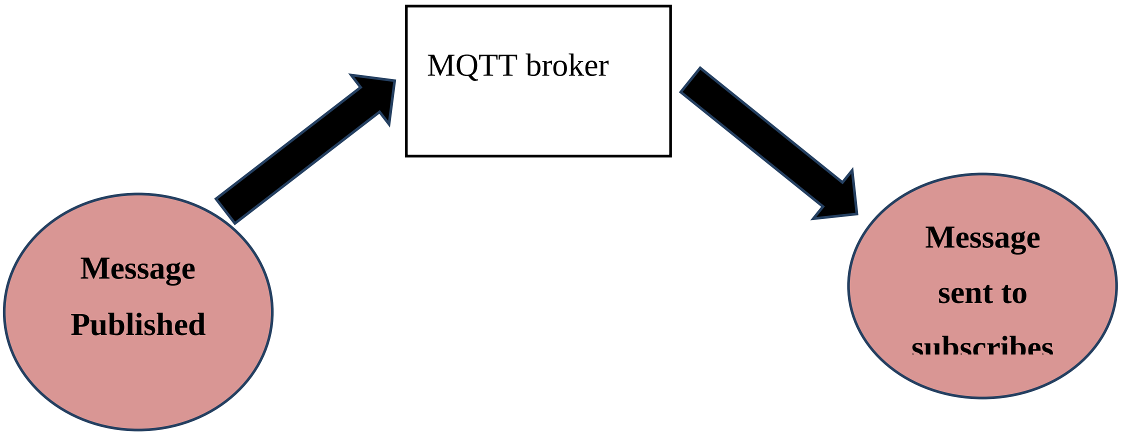

Figure 10. *Publish-subscribe Model.*

Made by me based from Staglianò, L., Longo, E., & Redondi, A. E. (2021, August). D-MQTT: design and implementation of a pub/sub broker for distributed environments. In 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS) (pp. 1-6). IEEE

II.3.1. MQTT Broker

The MQTT broker is a computer program that can be hosted by a third party or it can be self-developed and run on-premises or in the cloud. Implementations that are proprietary and open source are both accessible.

The broker serves in the capacity of a post office. MQTT clients use the "Topic" subject line instead of the intended recipient's direct connection address. Each subscriber gets a copy of every message related to that subject. A single client can register subscriptions to topics with several brokers, and numerous clients can subscribe to a subject from a single broker (one to many capabilities).

Because MQTT is a bi-directional communication protocol, each client can produce and receive data via publishing as well as subscribing. The devices can post sensor data while still receiving configuration data or control orders. This facilitates device management and control as well as data sharing. A client must publish numerous messages to the broker, each with a distinct topic specified. It is not possible for a client to broadcast the same data to multiple topics at once.

The client devices and server application become detached using the MQTT broker architecture. The clients are kept ignorant of each other's information in this way. If set up properly, MQTT can use TLS encryption over connections that are password, username, and certificate secured. Certification of the connection may be optional and take the form of a certificate file provided by the client that needs to match the server's copy.

Clients and broker software have the option to automatically switch to a redundant or automatic backup broker in the event of a failure. Backup brokers can also be configured to distribute client loads among several on-site servers, cloud servers, or a mix of these.

The broker can support both standard MQTT and MQTT for compliant specifications such as Sparkplug

This can be done with same server, at the same time and with the same levels of security. [16] [17]

- 16-www.catchpoint.com/network-admin-guide/mqtt-broker
- 17-Staglianò, L., Longo, E., & Redondi, A. E. (2021, August). D-MQTT: design and implementation of a pub/sub broker for distributed environments. In *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)* (pp. 1-6). IEEE.

II.3.2. Steps to set up an MQTT Broker

To install and Configure MQTT Broker on Raspberry Pi we start by connecting Raspberry Pi to WiFi Router, then we Open the terminal on Main Raspberry Pi and execute the following commands :

We type 'sudo nano /etc/hostssome text' which we change change rasberry pi to mainrasberry pi, we press 'CTRL+O' then 'ENTER' and then 'CTRL+X' follow up by statement 'sudo reboot'.

Once ower raspberry pi reboots we proceed by run the following commands :

- sudo apt install ufw
- sudo ufw enable
- sudo apt install mosquitto mosquitto-clients
- sudo systemctl enable mosquitto
- sudo ufw allow 1883

We are going to open two terminals once we finish, the first terminal will contain subscribe command **mosquitto_sub -t outTopic -v -h localhost**, Meanwhile in the second terrminal we excute the publish command **mosquitto_pub -d -t outTopic -m "Hello World" -h localhost**

It will show statement « hello world » in the first terminal (subscribe) [16]

16-www.catchpoint.com/network-admin-guide/mqtt-broker.

II.4. Implentation MQTT on Raspberry Pi Pico

Steps to implement MQTT on the Raspberry Pico

A. Required hardware

- Raspberry Pi Pico W : The main microcontroller with Wi-Fi capability.
- 2. USB Cable : To transmit data and power between ower PC and the Pico W.
- 3. For component connection and prototyping, use the Breadboard and Jumper Wires.
- Sensors/Actuators : we may need various sensors (e.g. temperature, humidity) and actuators (e.g. LEDs, relays).
- Power Supply : Typically, we power the Pico W via the USB cable, but for standalone applications, we use a battery pack. [8] [11] [7]

11-Khare, A. (2020). Design and implementation of enhanced message queuing telemetry transport protocol for internet of things (Doctoral dissertation, UPES, Dehradun).

17-Staglianò, L., Longo, E., & Redondi, A. E. (2021, August). D-MQTT: design and implementation of a pub/sub broker for distributed environments. In *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)* (pp. 1-6). IEEE.

18-Monk, S. (2022). *Raspberry Pi Cookbook*. " O'Reilly Media, Inc.".

B. Required software

- MicroPython: MicroPython : Python for microcontrollers Raspberry Pi Pico support officially added to MicroPython
- MQTT Library: For MicroPython, the `umqtt.simple` library is commonly used. You can install it via the `upip` package manager.
- MQTT Broker: This is the server where your Pico W will make conection, Options include Public brokers like Eclipse Mosquitto, HiveMQ

- Local broker on a separate Raspberry Pi or computer, using software like Mosquitto. [18] [11] [17]

11-Khare, A. (2020). Design and implementation of enhanced message queuing telemetry transport protocol for internet of things (Doctoral dissertation, UPES, Dehradun).

17-Staglianò, L., Longo, E., & Redondi, A. E. (2021, August). D-MQTT: design and implementation of a pub/sub broker for distributed environments. In *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)* (pp. 1-6). IEEE.

18-Monk, S. (2022). *Raspberry Pi Cookbook*. " O'Reilly Media, Inc.".

C. Connecting raspberry and MQTT

Steps to Set Up:

a) Set Up the Raspberry Pi Pico W:

We connect the Raspberry pi pico w to ower computer via USB, then downloading and flash the micro-Python firmware to it.

b) Install MQTT Library on MicroPython :

We upload the<umqtt.simple> library to pico w by using thonny IDE.

c) Write Your MQTT Code :

The subscribes/publishes to topics as needed by connecting to the MQTT broker after writing MicroPython script that connect to ower Wi-Fi network. [18] [13] [17]

13-Parikh, D. (2022). Raspberry Pi and MQTT Essentials: A complete guide to helping you build innovative full-scale prototype projects using Raspberry Pi and MQTT protocol. Packt Publishing Ltd).

17-Staglianò, L., Longo, E., & Redondi, A. E. (2021, August). D-MQTT: design and implementation of a pub/sub broker for distributed environments. In *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)* (pp. 1-6). IEEE.

18-Monk, S. (2022). *Raspberry Pi Cookbook*. " O'Reilly Media, Inc.".

D. Code for MQTT communication :

```
import network
import ubinascii
import machine
import time
from umqtt.simple import MQTTClient
# Wi-Fi credentials
SSID = "your_wifi_ssid"
PASSWORD = "your_wifi_password"
# MQTT broker details
MQTT_BROKER = "broker.hivemq.com"
CLIENT_ID = ubinascii.hexlify(machine.unique_id())
TOPIC = b"your/topic"
# Connect to Wi-Fi
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect(SSID, PASSWORD)
while not wlan.isconnected():
    pass
print("Connected to WiFi")
# Connect to MQTT broker
client = MQTTClient(CLIENT_ID, MQTT_BROKER)
```

Figure 11. *MQTT communication.*

```
client.connect()
def sub_cb(topic, msg):
    print((topic, msg))
client.set_callback(sub_cb)
client.subscribe(TOPIC)
While True:
    client.wait_msg()  # wait for a message
    time. Sleep(1)
```

Figure 12. *MQTT communication.*

II.4.2. Testing the set up

We verify ower MQTT setup on Raspberry pi pico W by testing the capability of publish and subscribe to MQTT topics correctly :

- **Publish a test message**

```python
import time
from umqtt.simple import MQTTClient

# MQTT broker configuration
MQTT_BROKER = "192.168.1.104"
MQTT_CLIENT_ID = "pico_client"
MQTT_TOPIC = "test"

# Callback function for when a message is received
def on_message(topic, msg):
    print("Received message on topic {}: {}".format(topic, msg))

# Create an MQTT client instance
client = MQTTClient(MQTT_CLIENT_ID, MQTT_BROKER)

# Set the callback function
client.set_callback(on_message)

# Connect to the MQTT broker
client.connect()
```

Figure 13. *Testing the MQTT set up on Thonny.*

```python
# Create an MQTT client instance
client = MQTTClient(MQTT_CLIENT_ID, MQTT_BROKER)

# Set the callback function
client.set_callback(on_message)

# Connect to the MQTT broker
client.connect()

# Subscribe to a topic (optional)
client.subscribe(MQTT_TOPIC)

while True:
    # Publish a message every 5 seconds
    client.publish(MQTT_TOPIC, "Hello from Pico!")
    time.sleep(5)

    # Check for incoming messages
    client.check_msg()
```

Figure 14. *MQTT set up.*

We replace BROKER_ADRESS, USERNAME and PASSWORD by ower MQTT broker's address and authentication credentials to get msg « hello-world »after running the programme.

- **Subscribe to the topic**

```python
from umqtt.simple import MQTTClient
import time
```

```python
# MQTT broker configuration
BROKER_ADDRESS = "mqtt.example.com"
CLIENT_ID = "pico-client"
TOPIC = "test/topic"
USERNAME = "your_username"
PASSWORD = "your_password"

# Callback function to handle incoming messages
def callback(topic, msg):
    print("Received message on topic {}: {}".format(topic, msg))

# Initialize MQTT client
client = MQTTClient(CLIENT_ID, BROKER_ADDRESS, user=USERNAME, password=PASSWORD)

# Set callback function
client.set_callback(callback)

# Connect to MQTT broker
client.connect()

# Subscribe to MQTT topic
client.subscribe(TOPIC)

# Wait for messages
while True:
    client.wait_msg()

# Disconnect from MQTT broker
client.disconnect()
```

Figure 15. *Subscribing to topic.*

We replace BROKER_ADRESS, USERNAME and PASSWORD by ower MQTT broker's address and authentication credentials, by running the programme on another devise whose connected in same MQTT-BROKER, it should subscribe to the specified MQTT topic and print out any messages received. [18] [13] [17]

13-Parikh, D. (2022). Raspberry Pi and MQTT Essentials: A complete guide to helping you build innovative full-scale prototype projects using Raspberry Pi and MQTT protocol. Packt Publishing Ltd).

17-Staglianò, L., Longo, E., & Redondi, A. E. (2021, August). D-MQTT: design and implementation of a pub/sub broker for distributed environments. In 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS) (pp. 1-6). IEEE.

18-Monk, S. (2022). Raspberry Pi Cookbook. " O'Reilly Media, Inc.".

- **Verify communication**

After running both scripts, we should see the message "Hello, world !" being published by the Pico script and received by the subscriber script but if ever we encounter any issues, we need double-check MQTT broker configuration, network connection, and authentication credentials[18] [11] [17]

II.4.3. Common issues

These are some issues we face usually while setting up

A. Network connectivity issues:

- Ensure that ower Raspberry Pi Pico is properly connected to the internet via Wi-Fi or Ethernet by Double-check Wi-Fi credentials and make sure they are entered correctly in ower code.
- Ensure that ower MQTT broker is accessible from the network and Pico is connected to Firewalls or network configurations could block the connection**.**

B. Authentication problems

- Ensure that ower MQTT broker is configured and able to connect to correct and specified username and password.
- If using TLS encryption, ensure that ower certificates are configured correctly

C. Incorrect MQTT broker address or port

- Double-check the address and port of ower MQTT broker in the code.
- Ensure that MQTT broker is running and accessible at the specified address and port.

D. Subscription issues

- Subscribing to the correct MQTT topic in ower code simultaneously ensuring that the topic exists and is being published to by other clients.
- Verify that the callback function for handling incoming messages is set up correctly

E. Message formatting errors

- Ensuring that the messages are published and subscribed to are coordinated correctly.
- Verify the problems with encoding or decoding message payloads

F. Resource constraints :

- Raspberry Pi Pico has limited resources compared to larger microcontrollers for that we ensure that ower code is optimized and does not exceed the memory or processing capabilities of the Pico

G. Debugging and logging :

- Implement logging and debugging mechanisms in the code to help identify and diagnose issues.
- Print debug messages to the console or log them to the file to track the flow of execution and any errors encountered

H. Broker-specific issues

- Familiarize with the documentation of ower chosen MQTT broker and follow any specific guidelines or recommendations cause different MQTT brokers have their own quirks and configuration requirements. [5] [4]

14-Domínguez-Bolaño, T., Barral, V., Escudero, C. J., & García-Naya, J. A. (2024). An IoT system for a smart campus: Challenges and solutions illustrated over several real-world use cases. *Internet of Things*, *25*, 101099.

15-Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., & Al-Hatmi, R. (2017, May). Internet of Things: Survey and open issues of MQTT protocol. In *2017 international conference on engineering & MIS (ICEMIS)* (pp. 1-6). Ieee.

II.4.4. Solutions to these issues

To solve the issues mentioned above we propose

A. Network connectivity issues

- Checking ower network settings and ensure the Raspberry Pi Pico is connected properly by double-check Wi-Fi credentials.
- Verify router settings and firewall configurations

B. Authentication problems

- Confirm the correctness of the username and password.
- Check MQTT broker settings to ensure it's configured to accept connections with the provided credentials.
- Ensure TLS certificates are correctly configured if using encryption.

C. Incorrect MQTT broker address or port

- Triple-check the broker address and port.
- Confirm that the broker is up and running

D. Subscription issues

- Verify subscribe to the correct existed topic whose being published.
- Check the callback function setup

E. Message formatting errors

- Ensure message payloads are correctly encoded and decoded.
- Validate message content against what's expected.

F. Resource constraints

- Optimise code to reduce memory usage and processing load by avoiding heavy operations and unnecessary loops

G. Debugging and logging

- Implement comprehensive logging and error handling.

- Print out debug messages or log them to a file.
- Utilise MicroPython's print () function for debugging

H. Broker-specific issues

- Consult the MQTT broker's documentation or online forums
- Check communities for user experiences and troubleshooting tips[15] [14]

14-Domínguez-Bolaño, T., Barral, V., Escudero, C. J., & García-Naya, J. A. (2024). An IoT system for a smart campus: Challenges and solutions illustrated over several real-world use cases. *Internet of Things*, *25*, 101099.

15-Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., & Al-Hatmi, R. (2017, May). Internet of Things: Survey and open issues of MQTT protocol. In *2017 international conference on engineering & MIS (ICEMIS)* (pp. 1-6). Ieee.

II.5. Real world application

MQTT (Message Queuing Telemetry Transport) a simple messaging protocol that is perfect for low-bandwidth, high-latency, or unstable networks to facilitate communication between devices. The Raspberry Pi Pico's low-cost, low-power microcontroller makes it a great platform for developing MQTT-based apps. That help us in ower project :

We designed a system that utilizes MQTT messages to remotely control temperature in beehives by heating water. We analysed and visualised this data in real time by publishing it over MQTT, which is provided by sensors attached to the Raspberry Pi Pico in status reports.

The Raspberry Pi Pico's ability to send it’s location and other relevant data to a central server via MQTT makes asset tracking how much energy is used, which helps us assure quality, optimize the process, and make it easy to maintain.

II.5.1. Sensors

A. DHT22

The DHT22 sensor, also known as the AM2302, is a popular choice for measuring temperature and humidity in various applications, including beehive monitoring. Its robust design and relatively high accuracy make it suitable for environments where precise climate control is essential for maintaining optimal conditions[5] . When integrating the DHT22 sensor with a Raspberry Pi for beekeeping purposes, the process typically involves connecting the sensor to the GPIO pins of the Raspberry Pi. Python scripts can then be written using libraries like `Adafruit_DHT` to read data from the sensor. This setup allows beekeepers to continuously monitor temperature and humidity levels inside the hive. By leveraging the capabilities of the Raspberry Pi and Python programming, beekeepers can analyze real-time sensor data to ensure that hive conditions remain within ideal ranges[. This proactive monitoring enables them to take timely actions, such as adjusting ventilation, activating heating or cooling devices, or triggering humidifiers or dehumidifiers, to optimize hive health and productivity. Overall, the integration of the DHT22 sensor with Python and Raspberry Pi exemplifies how technology can enhance beekeeping practices, providing valuable insights and control over environmental factors critical to bee colony well-being. As sensor technology continues to evolve, so too will the opportunities for innovation in sustainable beekeeping strategies [6].

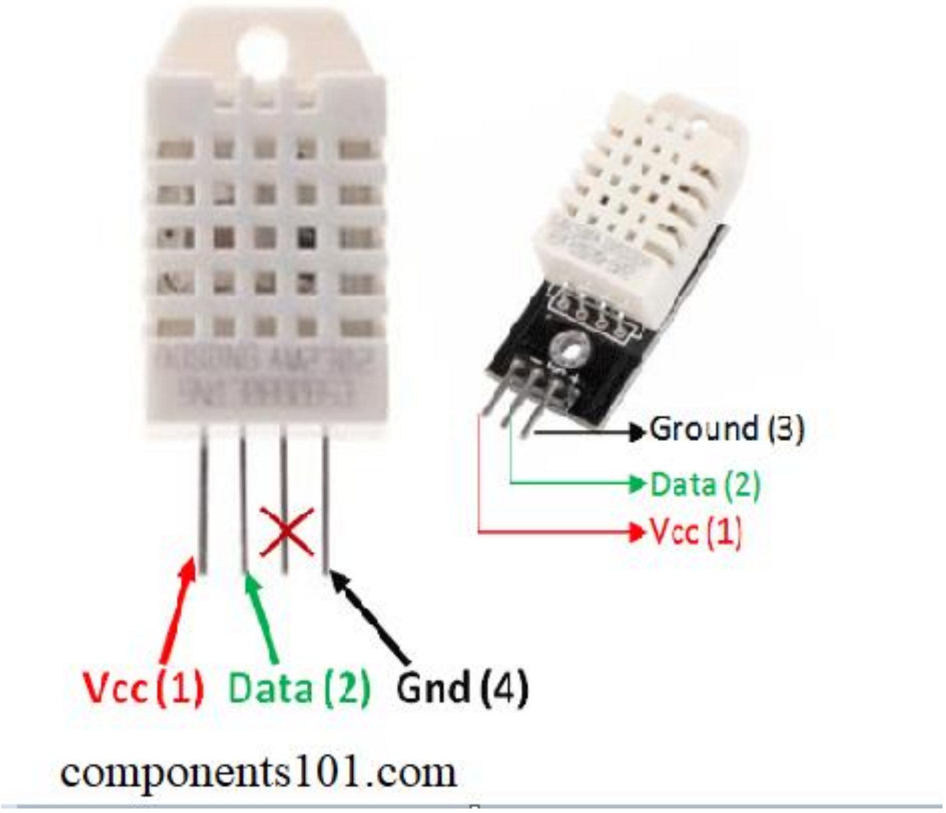

Figure 16. *DHT22 Diagram.*

B. DS18b20

The DS18B20 is a digital temperature sensor known for its accuracy, reliability, and ease of use in various applications, including beekeeping. It communicates over a 1-Wire bus, which allows multiple DS18B20 sensors to be connected to a single microcontroller, such as the Raspberry Pi, using only one data line.[7]

Key features of the DS18B20 include:

1 **Digital Output:** The DS18B20 provides digital temperature readings, which are easy to interface with microcontrollers like the Raspberry Pi. This simplifies the process of integrating the sensor into electronic projects and data acquisition systems.

2 **High Accuracy:** It has a resolution of up to 12 bits, allowing precise temperature measurements with a temperature accuracy of ±0.5°C in the range of -10°C to +85°C.

3 **Wide Operating Range:** The DS18B20 can measure temperatures ranging from -55°C to +125°C, making it suitable for both indoor and outdoor applications, including environments where temperature extremes are common, such as beekeeping.

4 **Unique 64-Bit Serial Code:** Each DS18B20 sensor has a unique 64-bit serial code burned into it during manufacturing. This allows multiple sensors to be distinguished on the same 1-Wire bus, simplifying the setup of sensor networks.

5 **Low Power Consumption:** It operates at a low power supply voltage (3.0V to 5.5V) and consumes very little power during temperature conversion (typically 750 µA).[7]

In beekeeping, the DS18B20 sensor is often used inside beehives to monitor internal temperature variations. By interfacing the DS18B20 with a Raspberry Pi and writing Python scripts to read and log temperature data, beekeepers can monitor hive conditions remotely. This capability helps in ensuring that the hive remains within the optimal temperature range for the well-being of the bees, particularly during critical periods such as winter or summer extremes.

Overall, the DS18B20's robust design, accuracy, and compatibility with microcontrollers make it a valuable tool for precise temperature monitoring in beekeeping and other environmental sensing applications. Its integration with platforms like the Raspberry Pi exemplifies how technology can support sustainable practices by providing real-time data for informed decision-making and proactive hive management.

Figure 17. *DS18b20 Diagram.*

II.5.2. Oled

The OLED (Organic Light Emitting Diode) display is a versatile and efficient technology used for visual feedback and data display in various electronic applications, including those related to beekeeping and environmental monitoring with devices like the Raspberry Pi.[8]

Key features and benefits of OLED displays include:

6 **High Contrast and Brightness:** OLED displays produce vibrant colors and high contrast ratios, making them easy to read in different lighting conditions, including outdoor settings typical of beekeeping environments.[8]

7 **Low Power Consumption:** OLEDs operate without a backlight, meaning they consume less power compared to traditional LCD displays. This is advantageous for battery-operated or energy-efficient applications. [8]

8 **Thin and Lightweight:** OLED displays are physically thinner and lighter than LCDs, making them suitable for compact and portable devices like handheld monitoring systems or integrated into hive monitoring equipment. [8]

9 **Fast Response Time:** OLEDs have faster response times than LCDs, resulting in smoother motion graphics and less motion blur. This is beneficial for displaying real-time data or graphs on temperature trends inside beehives. [8]

10 **Flexible and Organic Material:** OLED technology uses organic materials that can be deposited on flexible substrates, enabling the creation of curved or flexible display panels. While not always a necessity in beekeeping applications, this flexibility allows for creative integration in custom designs. [8]

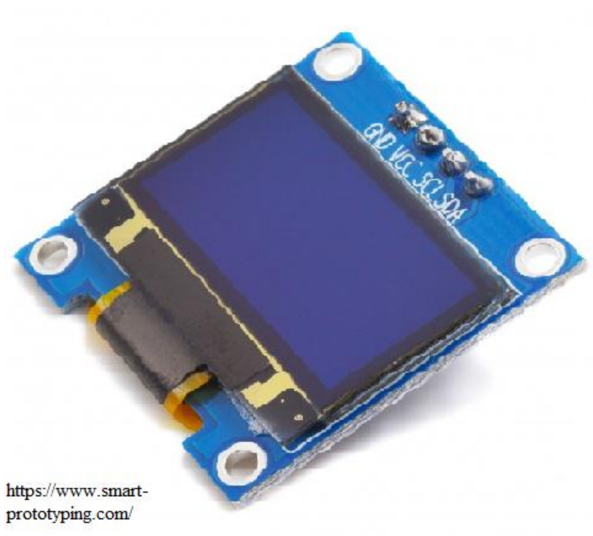

Figure 18. *OLED display Diagram.*

In beekeeping, OLED displays can be used in conjunction with sensors like the DS18B20 or DHT22 to provide real-time temperature and humidity readings directly on the device without needing a connected monitor. For instance, a Raspberry Pi with an OLED display could display current hive temperature and humidity levels, alert beekeepers to potential issues, and provide historical data trends for analysis.The integration of OLED technology with devices like the Raspberry Pi exemplifies how advanced display capabilities can enhance monitoring and management practices in beekeeping, contributing to more informed decision-making and improved hive health outcomes.

II.6. Conclusion

Throughout this chapter we were introduce to both raspberry pi pico w and MQTT and their mecanism and all advantages and cons which we discovered :

- The features and specifications of raspberry pi pico w and process of setup.
- The process to setup MQTT and use-cases.
- MQTT broker setup.
- The Required hardware and software.
- Problems encountered and thier solutions.
- Potential of MQTT with raspberry in real world.

CHAPTER III

System implimentation

III.1. Introduction

In the third chapter, we are going to present the implementation of our proposed system, detailed explanation about each step we have taken in order to give the final full image of the prototype.

III.2. System implementation

In this explanation, we implement the software and the hardware, we give detailed instructions of how we combined the programme with raspberry card and the sensors (DHT22, DS18B20) and displaying the results on the OLED, all of this work is going to be put in the beehive, both of the beehive and the electronic part are considered as the two main parts of our project.

Figure 19. *Full image of the system.*

In the picture above, the system is all combined, the central heater, the frames of the beehive, the water pump and also the raspberry card and the sensors who are not really shown in the image.

III.3. Steps of manufacturing

The following explanations present each single step we took to manufacture our prototype

A. Software Designing of the central heater

The first step we are going to take is preparing the central heater’ s design which follows the dimensions of our beehive, in order to that, we are going to use the ArtCam Pre software which is a software developed by Autodesk that specialised in 3D design and machining for woodworking, engraving, and manufacturing of complex shapes and artistic designs. The ArtCam allowed us as users to design intricate our 3D model, generate toolpaths for machining, simulate the machining process, and export files compatible with CNC machines.

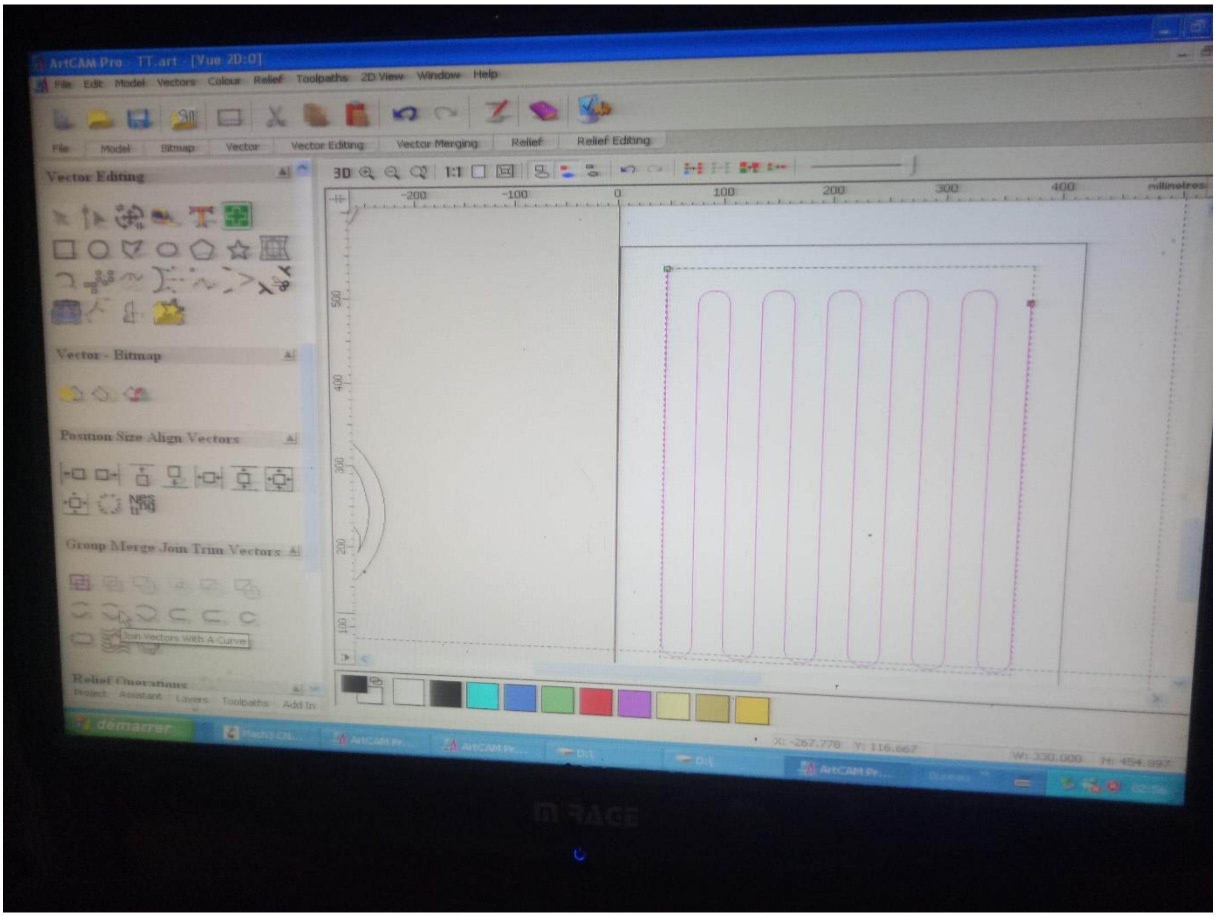

Figure 20. *Design of the central heater in the ArtCam Pre software.*

the figure III-2 appears the ultimate plan of the CH, we would begin with conceptualise the central heater's structure and components. This includes making point by point 3D models of the heater's body, warming components, ventilation frameworks, and any other vital portion ,By leveraging ArtCAM's capabilities, able to make a central radiator that's not as it were tastefully satisfying but moreover proficient and dependable in terms of warm yield and vitality utilisation. The software's capacity to handle perplexing 3D plans and create exact toolpaths makes it appropriate for creating complex warming frameworks custom-made to particular necessities and natural conditions.

B. Workshop Procedures

In the workshop, as in the figureIII-3 the chosen material for the central heater, such as what we have chosen stainless steel, undergoes several processes. Initially, the material is measured and cut according to precise dimensions using cutting tools or CNC machines. Next, components are shaped and formed through processes like bending, welding, or casting to achieve the desired structure and functionality. Surface treatments such as polishing or coating may be applied for durability and aesthetic purposes. Final assembly involves fitting all parts together according to design specifications. Quality checks ensure that the central heater meets safety standards and operational requirements before it is ready for installation or distribution.

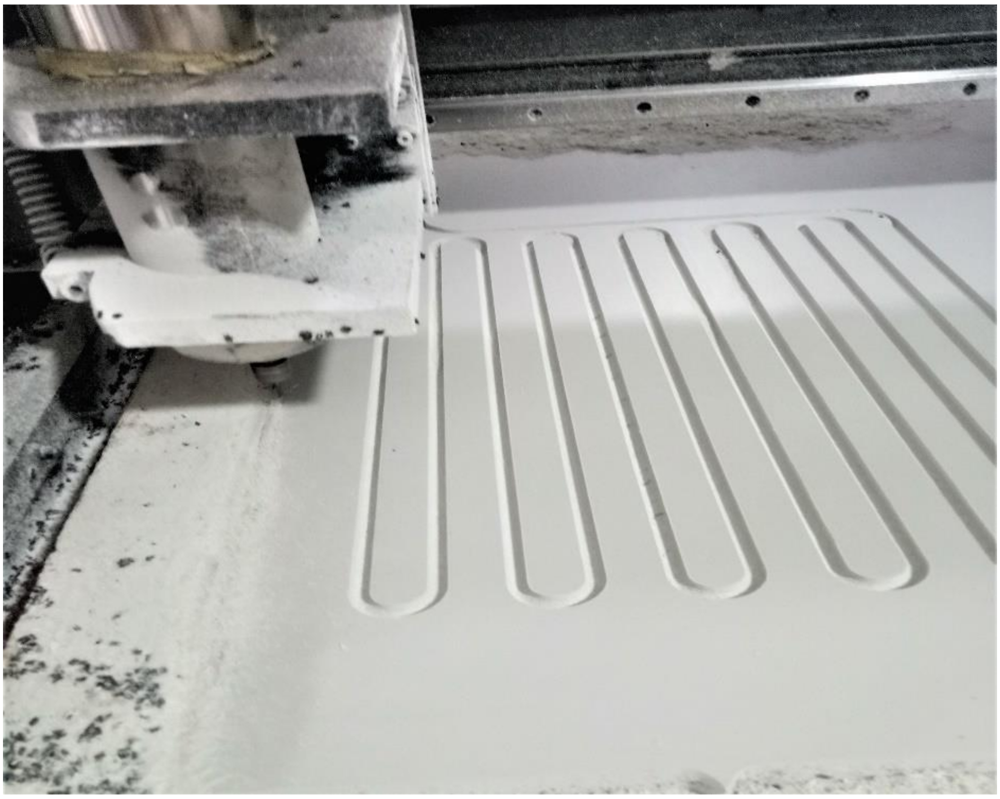

Figure 21. *Cutting procedure.*

C. Installation Process of Water Pipes in Central Heating Systems

When we install the water pipe in the central heating system, we follow several steps. Firstly, we measure and cut the pipes to the appropriate size based on the heating system design. Next, we prepare the ends of the pipes by stripping and ensuring cleanliness to ensure a tight connection. We then attach the pipes to the central heating fittings using seals and clamps to prevent leaks. Once we make the connections, we often conduct a pressure test to verify the system's integrity and ensure it will operate effectively once in use

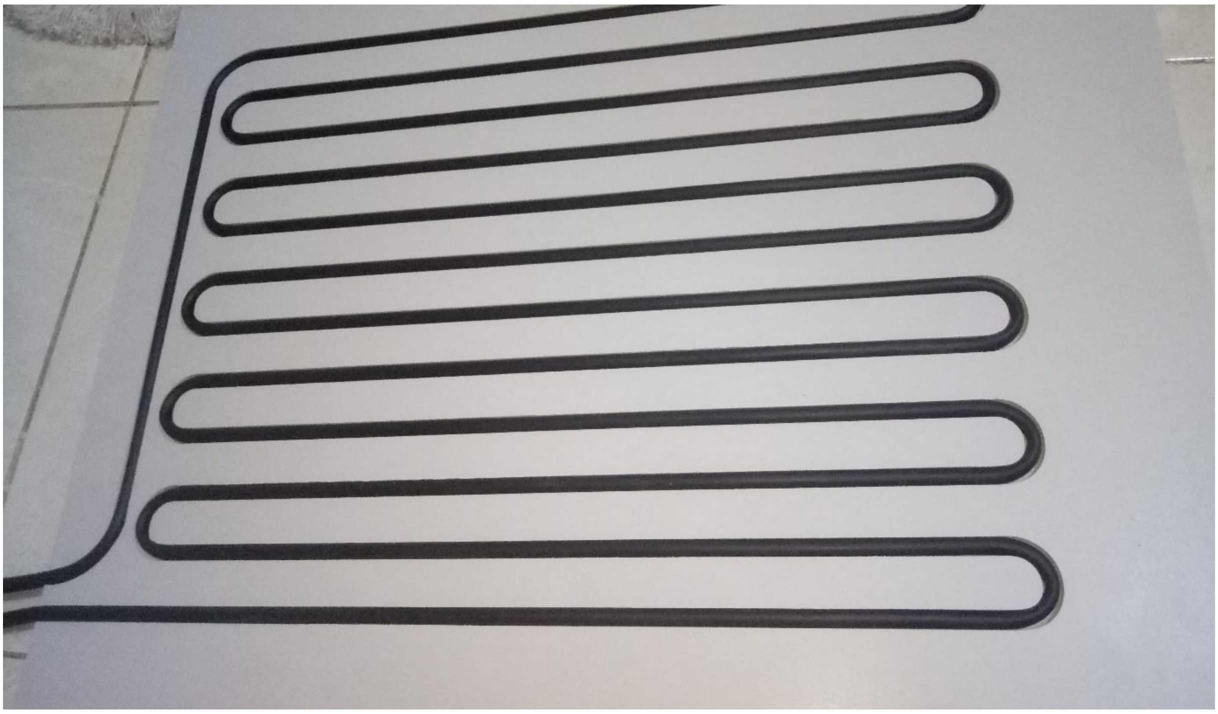

Figure 22. *Water pipe installation.*

In the figure III-4 we can see the well-installed water pipe after choosing the dimensions for the both of the pipe and inside the central heater, this led to a perfect installation for the pipe so it can never be moved or displaced during the water cycling.

D. Installation of Central Heater in the Beehive

When integrating a central heating system into a countertop, we need to carefully consider placement and functionality. Choosing an appropriate location is crucial to ensure the heating unit fits seamlessly without obstructing other elements. Whether recessed within the countertop or positioned visibly on top, our priority is to install it securely and efficiently. We must also manage electrical and plumbing connections meticulously to guarantee safety and optimal performance. Once installed, we'll test the system to ensure it operates effectively and make any necessary adjustments for seamless integration and functional it.

When integrating a central heating system into a countertop, we need to carefully consider placement and functionality. Choosing an appropriate location is crucial to ensure the heating unit fits seamlessly without obstructing other elements. Whether recessed within the countertop or positioned visibly on top, our priority is to install it securely and efficiently. We must also manage electrical and plumbing connections meticulously to guarantee safety and optimal performance. Once installed, we'll test the system to ensure it operates effectively and make any necessary adjustments for seamless integration and functionality.

Figure 23. *beehive after the Central Heater installation.*

The figure III-5 shows the final installation of the central heater in the bottom of the beehive after making sure that it is safely integrated.

E. Software

on the software side we are going to declare the code, the execution on the Thonny and the integration with the beehive

- **The code**

```
import time
import onewire
import ds18x20
import dht
from machine import Pin, I2C
from ssd1306 import SSD1306_I2C
```

```
# Initialize DS18B20 sensors
ow = onewire.OneWire(Pin(28))
ds1 = ds18x20.DS18X20(ow)
ds2 = ds18x20.DS18X20(ow)
devices = ds1.scan()

# Initialize DHT22 sensor
dSensor = dht.DHT22(Pin(13))

# Initialize LED
led = Pin(14, Pin.OUT)

# Pin numbers for I2C communication
sda_pin = 0
scl_pin = 1

# Display dimensions
WIDTH = 128
HEIGHT = 64

# Set up I2C communication
i2c = I2C(0, scl=scl_pin, sda=sda_pin, freq=200000)
time.sleep(0.1)

# Initialize SSD1306 display with I2C interface
oled = SSD1306_I2C(WIDTH, HEIGHT, i2c)

def read_sensors():
    try:
        # Read temperatures from DS18B20 sensors
        ds1.convert_temp()
        ds2.convert_temp()
        time.sleep_ms(750)
        temp_ds1 = ds1.read_temp(devices[0])  # First DS18B20 device
        temp_ds2 = ds2.read_temp(devices[1])  # Second DS18B20 device

        # Read temperature and humidity from DHT22
        dSensor.measure()
        temp_dht = dSensor.temperature()
        temp_f = (temp_dht * (9/5)) + 32.0
        hum_dht = dSensor.humidity()

        return temp_ds1, temp_ds2, temp_dht, temp_f, hum_dht
    except OSError as e:
        print('Failed to read data from sensors')
        return None, None, None, None, None

while True:
    temp_ds1, temp_ds2, temp_dht, temp_f, hum_dht = read_sensors()
```

```
    if temp_ds1 is not None and temp_ds2 is not None and temp_dht is not None and temp_f is not None and hum_dht is not None:
        # Clear the display
        oled.fill(0)

        # Display temperature from DS18B20 sensors
        oled.text("TF: {:.1f} C".format(temp_ds1), 0, 10)
        oled.text("Text Temp: {:.1f} C".format(temp_ds2), 0, 20)

        # Display temperature and humidity from DHT22
        oled.text("TempB: {:.1f} C".format(temp_dht, temp_f), 0, 30)
        oled.text("Hum: {:.1f}%".format(hum_dht), 0, 40)

        # Show the updated display
        oled.show()

        # Check if DHT22 temperature is greater than 35°C
        if temp_dht > 35:
            # Turn on the LED
            led.value(1)
        else:
            # Turn off the LED
            led.value(0)

    # Wait before taking the next reading
    time.sleep(2)
```

- **Code on Thonny**

The declaring of the code on thonny allows the users to see the results on the shell within if there are mistakes to be corrected or if the user want to develop and add instructions to his beehive to level up the security and the friability of the system .

```python
import time
import onewire
import ds18x20
import dht
from machine import Pin, I2C
from ssd1306 import SSD1306_I2C

# Initialize DS18B20 sensors
ow = onewire.OneWire(Pin(28))
ds1 = ds18x20.DS18X20(ow)
ds2 = ds18x20.DS18X20(ow)
devices = ds1.scan()

# Initialize DHT22 sensor
dSensor = dht.DHT22(Pin(13))

# Initialize LED
led = Pin(14, Pin.OUT)

# Pin numbers for I2C communication
sda_pin = 0
scl_pin = 1
```

Figure 24. *Initialisation part.*

```python
# Display dimensions
WIDTH = 128
HEIGHT = 64

# Set up I2C communication
i2c = I2C(0, scl=scl_pin, sda=sda_pin, freq=200000)
time.sleep(0.1)

# Initialize SSD1306 display with I2C interface
oled = SSD1306_I2C(WIDTH, HEIGHT, i2c)

def read_sensors():
    try:
        # Read temperatures from DS18B20 sensors
        ds1.convert_temp()
        ds2.convert_temp()
        time.sleep_ms(750)
        temp_ds1 = ds1.read_temp(devices[0])  # First DS18B20 device
        temp_ds2 = ds2.read_temp(devices[1])  # Second DS18B20 device

        # Read temperature and humidity from DHT22
        dSensor.measure()
        temp_dht = dSensor.temperature()
```

Figure 25. *Defining and reading temperature from DS18b20.*

Thonny - <untitled> @ 82 : 18

File Edit View Run Tools Help

<untitled> *

```python
def read_sensors():
    try:
        # Read temperatures from DS18B20 sensors
        ds1.convert_temp()
        ds2.convert_temp()
        time.sleep_ms(750)
        temp_ds1 = ds1.read_temp(devices[0])  # First DS18B20 device
        temp_ds2 = ds2.read_temp(devices[1])  # Second DS18B20 device

        # Read temperature and humidity from DHT22
        dSensor.measure()
        temp_dht = dSensor.temperature()
        temp_f = (temp_dht * (9/5)) + 32.0
        hum_dht = dSensor.humidity()

        return temp_ds1, temp_ds2, temp_dht, temp_f, hum_dht
    except OSError as e:
        print('Failed to read data from sensors')
        return None, None, None, None, None

while True:
    temp_ds1, temp_ds2, temp_dht, temp_f, hum_dht = read_sensors()

```

Figure 26. *Reading temperature and humidity from the DHT22.*

```python
while True:
    temp_ds1, temp_ds2, temp_dht, temp_f, hum_dht = read_sensors()

    if temp_ds1 is not None and temp_ds2 is not None and temp_dht is not None and temp_f is not None and hum_dht is not None:
        # Clear the display
        oled.fill(0)

        # Display temperature from DS18B20 sensors
        oled.text("TF: {:.1f} C".format(temp_ds1), 0, 10)
        oled.text("Text Temp: {:.1f} C".format(temp_ds2), 0, 20)

        # Display temperature and humidity from DHT22
        oled.text("TempB: {:.1f} C".format(temp_dht, temp_f), 0, 30)
        oled.text("Hum: {:.1f}%".format(hum_dht), 0, 40)

        # Show the updated display
        oled.show()

        # Check if DHT22 temperature is greater than 35°C
        if temp_dht > 35:
            # Turn on the LED
```

Figure 27. *Displaying the sensors' values on the Oled.*

```
        oled.fill(0)

        # Display temperature from DS18B20 sensors
        oled.text("TF: {:.1f} C".format(temp_ds1), 0, 10)
        oled.text("Text Temp: {:.1f} C".format(temp_ds2), 0, 20)

        # Display temperature and humidity from DHT22
        oled.text("TempB: {:.1f} C".format(temp_dht, temp_f), 0, 30)
        oled.text("Hum: {:.1f}%".format(hum_dht), 0, 40)

        # Show the updated display
        oled.show()

        # Check if DHT22 temperature is greater than 35°C
        if temp_dht > 35:
            # Turn on the LED
            led.value(1)
        else:
            # Turn off the LED
            led.value(0)

    # Wait before taking the next reading
    time.sleep(2)
```

Figure 28. *Adding a condition and illuminating a warning LED.*

When we add our code to Thonny, we start by writing our Python script in the editor and saving it. Next, we click the "Run" button to execute the code. If there are errors, Thonny highlights them and provides error messages indicating where they occur. We then analyse these messages together to understand and fix any issues, whether they're syntax errors or logical mistakes. Using Thonny's debugging tools, we inspect variable values and step through the code to pinpoint and correct errors efficiently. After making adjustments, we run the code again to ensure it runs smoothly without errors, collaborating to refine and finalise the script as needed.

F. Optimising Beehive Conditions: Integrating Python, Raspberry Pi, and Sensors for Humidity and Temperature Control

When we combine our Python code with a Raspberry Pi and a beehive to manage humidity and temperature, we begin by setting up the hardware components. This involves connecting humidity and temperature sensors (such as DHT series sensors) to the GPIO pins of the Raspberry Pi. These sensors are carefully placed inside the beehive to capture accurate readings of the internal environmental conditions.Next, in our software development phase using tools like Thonny, we write Python scripts that enable the Raspberry Pi to interface with these sensors. We use libraries such as `Adafruit_DHT` for DHT sensors to retrieve data in real-time.

The Python script continuously reads sensor values from the Raspberry Pi's GPIO pins. Once the data is acquired, our Python code processes it to ensure optimal hive conditions. This includes implementing algorithms to analyse the sensor readings. For instance, we might calculate average temperature and humidity levels over time, detect sudden fluctuations, or compare current values against predefined thresholds that indicate potential issues such as overheating or excessive moisture. To enhance functionality, we integrate control mechanisms within our code. For example, based on the analysed data, we can include logic to activate cooling or heating devices if temperature thresholds are exceeded, or trigger humidifiers or dehumidifiers to adjust humidity levels accordingly. This proactive approach helps maintain stable and favourable conditions inside the beehive, crucial for the health and productivity of the bee colony.

Throughout the integration process, testing is crucial. We use Thonny's debugging tools to verify that sensor readings are accurate and that our algorithms respond appropriately to varying environmental conditions. We also consider implementing logging and monitoring features within our code to track long-term trends and ensure ongoing optimization of hive management practices. By combining hardware components like sensors and the Raspberry Pi with our Python programming expertise, we create a comprehensive system that not only monitors but actively manages the microclimate inside the beehive. This integrated approach supports sustainable beekeeping practices by ensuring optimal conditions for bee health and productivity.

G. Data analysis

1. Variation of Humidity, internal, external temperature and Fluid temperature as function as Time

In order to control the behavior of our system, we have plotted the variation of Humidity, internal, external temperature and Fluid temperature as function as Time as shown in Figure 29.

The variation of the fluid's temperature as function as time is divided into three parts. the first one represents an increase in temperature from 40 to 50°C over a period of 12 minutes, then a second part of a stable temperature with a small variation (+/- 1°C) followed by a third part which represents the temperature drops from 50 to 40°C over a period of 20 minutes. By comparing the two durations, the heating duration is less than the duration of the temperature drop, which confirms that the system is profitable and retains the heat inside the Beehive.

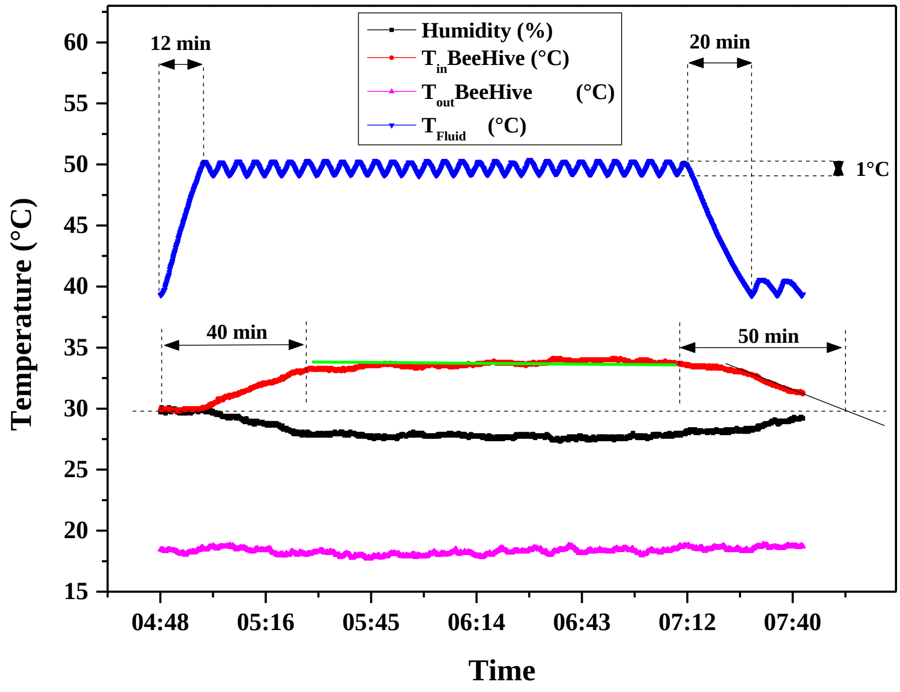

Figure 29. *Variation of Humidity, internal, external temperature and Fluid temperature as function as Time.*

The variation of the internal temperature of the hive follows the same behavior as the temperature of the fluid, it is also divided into three parts, the first part is devoted to heating the hive which takes a duration of 40 min to go from 30°C to 34°C then followed by a second part with a stable temperature and then a third part which takes 50 min for the temperature to drop from 34°C to 30°C. This result shows that our system is a good thermal insulator and reliable to be used in the field of beekeeping. We note that all these values of temperatures are recorded in external temperature between 17 and 18 °C.

2. Variation of Humidity as function as internal temperature

Figure 30. shows the variation of humidity as a function of internal temperature in BeeHive. It is clearly seen that humidity is inversely proportional to the internal temperature. In the middle of the day we notice a decrease in humidity due to the increase in air temperature, at the same time there is an increase in air speed, but at night the opposite phenomenon occurs. The variation follows a linear function with a slope of - 0.57. It is seen that the correlation coefficient (r^2) is very close to 1 (r^2 = 0.9673) which confirms the well-fitting of the experimental results.

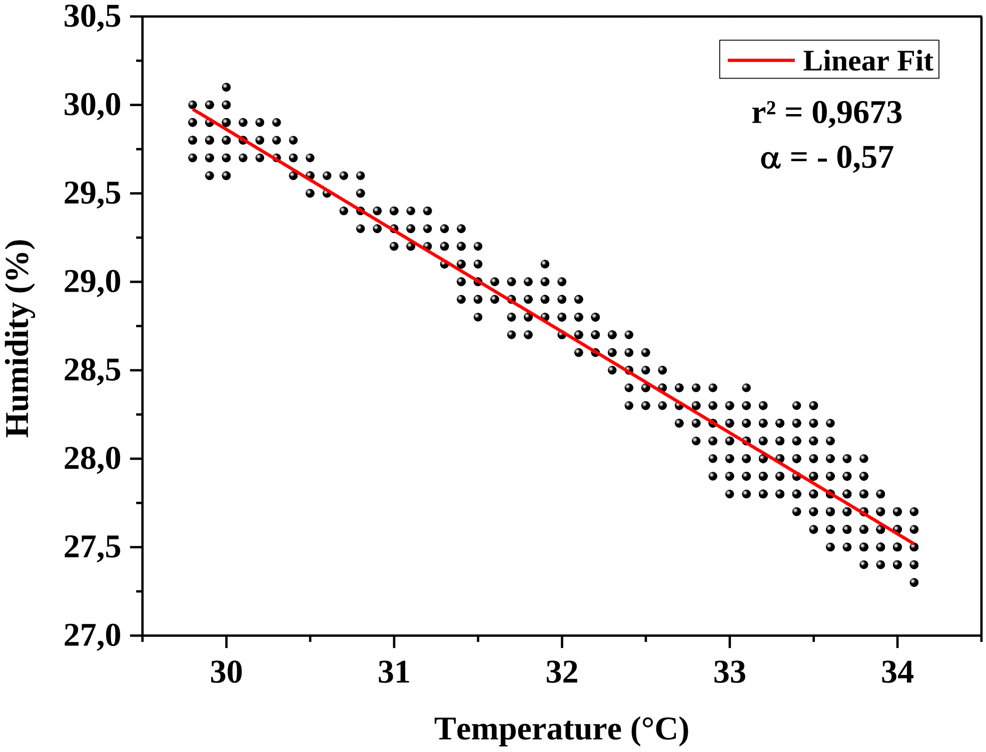

Figure 30. *Variation of Humidity as function as internal temperature.*

III.4. Conclusion

In conclusion, our exploration into integrating Python programming with a Raspberry Pi and various sensors for managing a beehive's humidity and temperature has unveiled a powerful approach to modern beekeeping. Through careful hardware setup, including the connection of sensors to the Raspberry Pi, and meticulous software development using tools like Thonny, we've established a robust framework. This framework allows us to not only monitor but also dynamically control the hive environment. By analysing real-time sensor data and implementing responsive actions, such as adjusting environmental controls based on preset thresholds, we enhance hive health and productivity. This integrated system exemplifies how technology can support sustainable beekeeping practices, ensuring optimal conditions for bee colonies. Moving forward, continued refinement and innovation in sensor technology and software development will further empower beekeepers to make informed decisions and foster thriving bee populations.

General Conclusion

Conclusion

In conclusion, the concept of an air-conditioned beehive with central heating represents a promising avenue for enhancing beekeeping practices. It holds potential to mitigate environmental stressors and support honeybee colonies in a changing climate. However, balancing technological benefits with ecological and economic considerations remains essential to ensure sustainable beekeeping practices and the long-term health of pollinator populations. Reflecting on the concept of an air-conditioned beehive with central heating presents both innovative opportunities and considerations for the well-being of honeybee colonies. The integration of modern climate control technologies into traditional beekeeping practices marks a significant evolution in apiary management, aimed at enhancing bee health and productivity. By regulating the hive's internal temperature and humidity levels, such advanced systems offer potential benefits such as climate stability, winter survival, heat stress mitigation, and disease prevention. Looking ahead, the addition of a cooling system for summer use could further enhance the resilience of beehives in extreme temperatures. This adaptation would address the challenges posed by increasingly hot summers, ensuring that bees remain productive and healthy even in the face of rising temperatures. However, while technological advancements promise these advantages, there are also considerations and potential challenges to address: natural adaptation, energy consumption, cost and accessibility, and finally, long-term effects. It is crucial to monitor and study the impacts of these innovations on bee behavior and health over extended periods. Additionally, ensuring that these technologies are economically viable and accessible to beekeepers of all scales will be key to their widespread adoption. Incorporating climate control systems into beekeeping practices represents a forward-thinking approach to supporting honeybee populations. By carefully balancing the benefits and challenges, we can work towards a more sustainable and resilient future for both bees and the ecosystems they support

References

[1] life of bees https://www.lib.uidaho.edu/digital/objects/guidedreading/guidedread087.pdf

[2] Abhishek. K Yadav and sanjay.H (2023)the fascinating life of bees article in ResearchGate

[3] Crane, E. (1999). *The world history of beekeeping and honey hunting*. Routledge. Frisch, K. V. (1993). *The dance language and orientation of bees*. Harvard University Press

[4] Klein, A. M., Vaissière, B. E., Cane, J. H., Steffan-Dewenter, I., Cunningham, S. A., Kremen, C., & Tscharntke, T. (2007). Importance of pollinators in changing landscapes for world crops. *Proceedings of the royal society B: biological sciences*, *274*(1608), 303-313

[5] manufacturer of the DHT22 sensor (AM2302) https://www.adafruit.com/product/385 (Acccessed June,2024)

[6] Raspberry Pi Documentation https://www.raspberrypi.com/documentation/ (Acccessed June,2024)

[7] ds18b20 datasheet https://www.alldatasheet.com/ (Accessed June,2024)

[8] Adafruit OLED Display Guide. https://learn.adafruit.com/monochrome-oled-breakouts/arduino-library-and-examples (accessed June,2024)

[9] Baily.J(1990) . the life cycle of bee

[10] Hillar, G. C. (2017). MQTT Essentials-A lightweight IoT protocol. Packt Publishing Ltd.

[11] Khare, A. (2020). Design and implementation of enhanced message queuing telemetry transport protocol for internet of things (Doctoral dissertation, UPES, Dehradun).

[13] Parikh, D. (2022). Raspberry Pi and MQTT Essentials: A complete guide to helping you build innovative full-scale prototype projects using Raspberry Pi and MQTT protocol. Packt Publishing Ltd.

[14] Domínguez-Bolaño, T., Barral, V., Escudero, C. J., & García-Naya, J. A. (2024). An IoT system for a smart campus: Challenges and solutions illustrated over several real-world use cases. Internet of Things, 25, 101099.

[15] Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., & Al-Hatmi, R. (2017, May). Internet of Things: Survey and open issues of MQTT protocol. In 2017 international conference on engineering & MIS (ICEMIS) (pp. 1-6). Ieee.

[16] www.catchpoint.com/network-admin-guide/mqtt-broker

[17] Staglianò, L., Longo, E., & Redondi, A. E. (2021, August). D-MQTT: design and implementation of a pub/sub broker for distributed environments. In 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS) (pp. 1-6). IEEE.

[18] Monk, S. (2022). Raspberry Pi Cookbook. " O'Reilly Media, Inc.".

[19] Lindauer, M. (1971). *Communication among social bees*. Harvard University Press.

[20] Meixner, M. D. (2010). A historical review of managed honey bee populations in Europe and the United States and the factors that may affect them. *Journal of invertebrate pathology*, *103,* S80-S95.

[21] Seeley, T. D. (1989). The honey bee colony as a superorganism. *American Scientist*, *77*(6), 546-553.

[22] Seeley, T. D. (2009). *The wisdom of the hive: the social physiology of honey bee colonies*. Harvard University Press.

[23] Seeley, T. D. (2014). *Honeybee ecology: a study of adaptation in social life* (Vol. 36). Princeton University Press.

Winston, M. L. (1987). *The biology of the honey bee*. harvard university press.

[24] alldatasheet.fr, "YF-S401 Datasheet(PDF) - List of Unclassifed Manufacturers." https://www.alldatasheet.fr/datasheet-pdf/pdf/1242513/ETC1/YF-S401.html (accessed Jul. 03, 2023).

[25] Honey Beekeeping magazine https://medpcelarskimagazin.mvbyte.com/ (accessed on June,2024)

[26] Top quality beehive https://littlelarueapiary.patternbyetsy.com/listing/530026543/2-deep-complete-bee-hive-kit-wframes 1 nov 2017 accessed on june 2024

[27] changed flybee trading https://fly-bee.com/www.fly-bee.com 2021 accessed on June 2024

[28] American Beekeeping Federation. (n.d.). American Beekeeping Federation. Retrieved June 20, 2024, from https://www.abfnet.org/

ملخص

يقدم هذا المشروع تقنية جديدة في عالم تربية النحل، حيث أن التحكم الذكي داخل بيوت النحل يسمح بمراقبة سلوكه عن طريق المستشعرات الموضوعة داخله ، من بين هاته المستشعرات نذكر أجهزة الاستشعار الخاصة بدرجة الحرارة و الرطوبة للإنذار في حالة حصول حركة غير اعتيادية الفعال مع وضع اجهزة للاتصال MQTT.

يستفيد هذا النظام من بروتوكول و نقل البيانات من أجهزة الاستشعار و دمجها مع نظام EXCEL لمراقبة فعالية أداء الاجهزة و دقة معطياتهم مع امكانية ملاحظة الخطأ في حال حدوثه و تصحيحه فوريا ، هذا المشروع يعرض أهمية دمج عالم التقنية و البرمجيات مع مختلف المجالات و كيف أنها تساهم بشكل كبير من ارتفاع المردودية و بالتالي النهوض بالقطاع الاقتصادي في البلاد

الكلمات المفتاحية: النحل -بيوت النحل-أجهزة الاستشعار -LED-بروتوكول -MQTTتدفئة مركزية-DADANT- LANGSTROTH- Raspberry pi pico – تحليل البيانات –حفظ البيانات .w

Résumé

Ce projet introduit une nouvelle technologie dans l'apiculture, où un système de climatisation à l'intérieur des ruches avec un contrôle intelligent qui permet de surveiller la température et l’humidité grâce à des capteurs placés à l'intérieur. Ce système utilise le protocole MQTT pour une communication efficace et une transmission de données depuis les capteurs, les intégrant avec l’Excel pour traiter les données et surveiller l'efficacité et la fiabilité de notre système, avec la capacité d'observer et de corriger immédiatement les erreurs. Ce projet démontre l'importance d'intégrer la technologie et les logiciels avec divers domaines et comment ils contribuent significativement à accroître la productivité et donc à faire progresser le secteur économique du pays.

Mots clés: Abeilles - Ruches - Capteurs - LED - Protocole MQTT - Chauffage Central - Dadant - Langstroth - Raspberry Pi Pico - Analyse de Données - Stockage de Données.

Abstract

This project introduces a new technology in beekeeping, where an air conditioning system inside the hives with intelligent control allows monitoring of temperature and humidity through sensors placed inside. This system uses MQTT protocol for efficient communication and data transmission from the sensors, integrating them with Excel for data processing and monitoring the efficiency and reliability of our system, with the ability to observe and immediately correct errors. This project demonstrates the importance of integrating technology and software with various domains and how they significantly contribute to increasing productivity and thus advancing the country's economic sector.

Keywords: Bees - Hives - Sensors - LED - MQTT Protocol - Central Heating - Dadant - Langstroth - Raspberry Pi Pico - Data Analysis - Data Storage.