Smart Beehive Monitoring System

Team 01

05.07.2023

Team members

- *Jaouaher Belgacem*
- *Jasmeet Singh*

Introduction

In this project, we are aiming to develop a Smart Beehive system to help beekeepers with monitoring the microclimate inside the beehive remotely using their mobile phones or web service. Thus, to check whether the beehive is healthy or not, we will focus on monitoring two main factors which are the humidity and temperature. It is important to mention that, for a healthy bee colony, the temperature should be within the range of 32 to 35 C and between 50-60% for humidity. As a result, our system will cool/heat the beehive in case the humidity and/or the temperature exceed or be less than the acceptable ranges.

Concept description

Wireless sensor networks are widely used in several applications, especially in agriculture to improve production efficiency and effectiveness. Thus, our WSN targets the agriculture area of agriculture and it consists of a distributed system that is encapsulated in the form of sensing devices to measure the microclimate inside the beehive. The used sensor is integrated for collecting data regarding temperature and humidity.

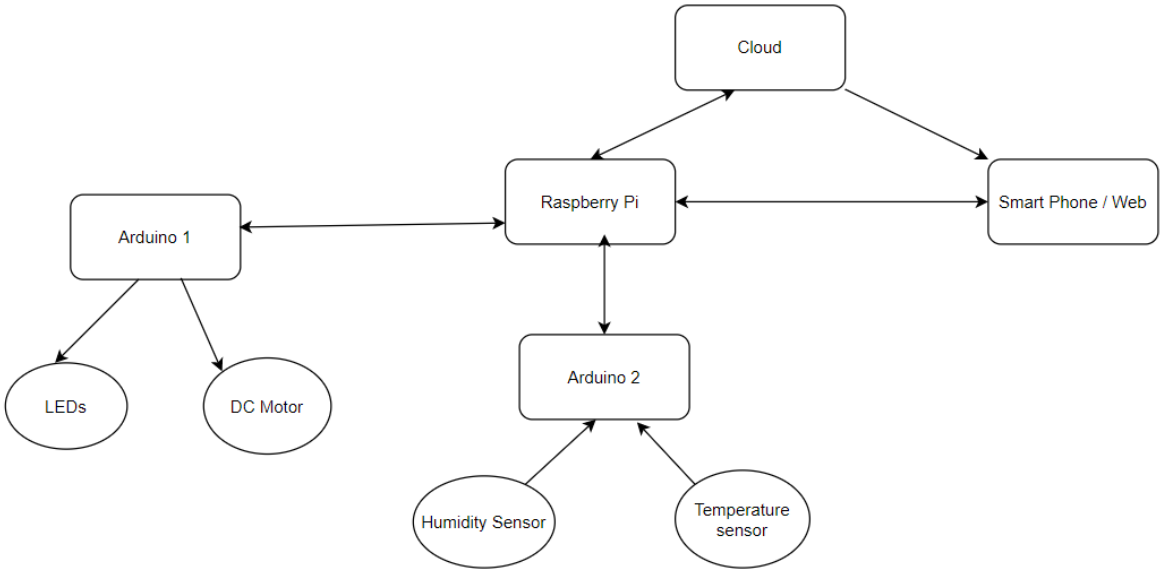


Fig.1 Block Diagram of the Smart Beehive

This figure illustrates the concept of our system. As shown in the block diagram our Wireless Sensor Network system uses the Raspberry Pi as a base communication node between the two subsystems. The sensor node represented by Arduino 1 will be placed inside the

beehive. The type of this Arduino is Wifi R2 and it is responsible for collecting data from the sensors which are the temperature and humidity sensors (we may use one sensor that encapsulates both functions). The second subsystem (Arduino 2), which has the same type as the first part of the system will be attached outside the beehive near its entrance. The second part is composed of two actuators: a DC motor that will act as a cooling/heating system and RGB LED that will be used as indicators for the monitoring systems' functioning status. Thus, in case the monitoring system is working correctly and communicating data the RGB LED will be on. However, in case of a communication problem, it will be off.

The MQTT communication is also a big part of this system, where both Arduino's will exchange data with the Raspberry Pi to send data from one part and react upon it in the other part. Besides, the Raspberry Pi will share the exchange with the smartphone/ web and share it with the cloud too.

The Main parts of our proposed system are:

1. 1 x RGB LED Actuator
2. 1 X Humidity/Temperature Sensor
3. 1 x DC Motor
4. 1 X Raspberry Pi
5. 2 X Microcontrollers: Arduino Uno Wifi R2
6. 1 X SmartPhone
7. 1 X Web page
8. Cloud (optional)
9. Documentation

Project/Team management

Project management

As a team, we chose to work with the scrum project management model. As it is easier to break down the life cycle of the project into several iterations. The tasks are gradually assigned to reach the ultimate task of the project. This leads to periodic meetings to check the progress status of the tasks of each team member.
To manage this project we used several tools:

- Github: to upload files
- WhatsApp group: to discuss some points briefly and send meetings information
- Google Meet: to meet and discuss the progress, tasks, and problems we have faced.

Tasks and Roles

The project is divided into two main parts:

1. Arduino + Sensor part
2. Raspberry Pi part (MQTT)
3. Cloud Connections
4. Documentation

For our team, we both did all parts. Hence, we divided the project into :

- Jasmeet: Earlier stage Block diagram + MQTT Server Setup + cloud, 1 subscriber, 1 Publisher: Topic Temperature and actuator DC motor and documentation.

- Jaouaher: developed the overall concept + Block diagram + MQTT Server Setup, 1 Subscriber 1 Publisher: Topic Humidity and actuator RGB LED. Documented the different parts.

Technologies

To realize our smart Beehive, we developed 4 parts:

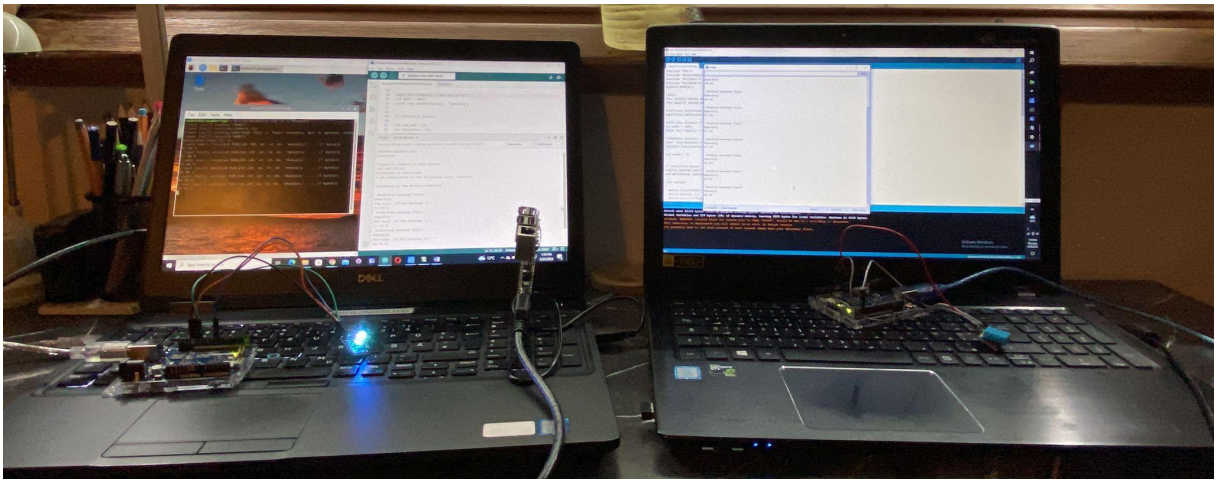
- MQTT Communication over WiFi
- Humidity/Temperature Sensor
- Arduino C

Implementation

For the implementation, we had two parts:

Arduino

One Arduino is publishing the Humidity/temperature measurements, and another Arduino is reading this data through the MQQT protocol and acting accordingly by managing the actuators.



Raspberry Pi

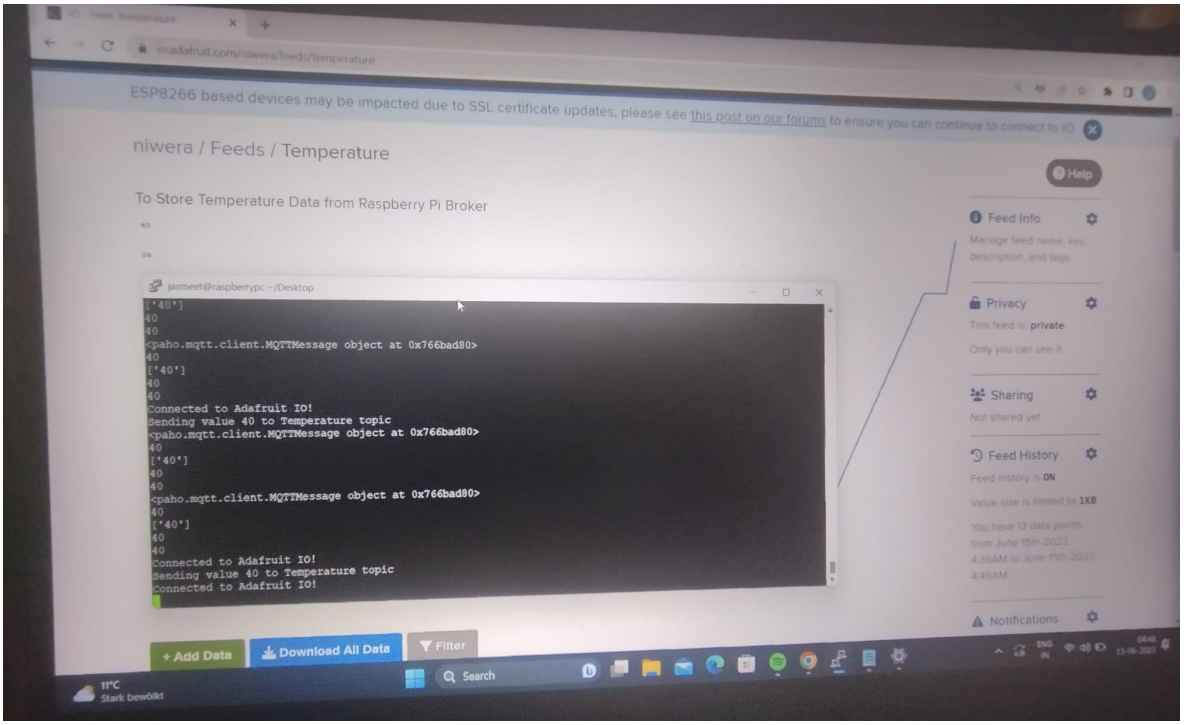
For the Raspberry Pi, after we set up its operating system, we use the VNC viewer software/ Putty SSH to access it remotely. The Raspberry Pi is our broker which is the intermediate between the publisher and the subscriber. Once, it is connected to the WiFi, it read the sent data by the publisher and displays it on the CMD window. This data will be accessed through the subscriber to read it.
The different devices of the overall system have to be connected to the same network.

Adafruit IO

Connecting the Raspberry Pi Broker to the Cloud
Method to connect the broker to the cloud

1) Create an Adafruit IO account.

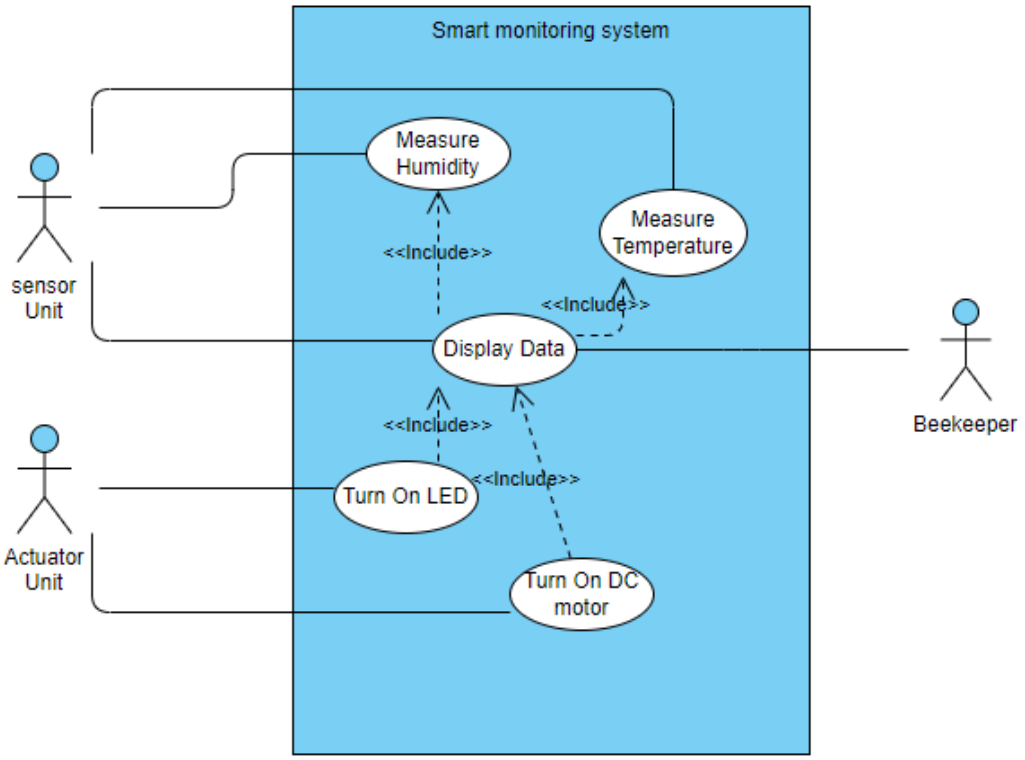
2) Create Feed and Dashboard (Remember that the Feed is key sensitive when connecting to the broker.)
3) Get the BridgeBroker code from the professor (Dr. Ali Hayek) - {Connecting Raspberry Pi to the Cloud}.
4) Since that code was developed for another project, debug it, and after debugging it, run it on a Raspberry Pi.
5) Things to debug :
    a) Update the code syntax with respect to available version of Paho library
    b) Update the main function
    c) update the transmission function
    d) update the key values
    e) If using username and password for MQTT connection, provide username and password in main function to the Paho library to connect.



Use Case

The main Use Case of our system is to have a smart Beehive monitoring system. Our system should offer:

- Remote monitoring ability for the Beekeeper
- Autonomous control of the temperature and humidity
- Accurate and live measurement of temperature and humidity



Sources/References

https://github.com/JaouaherBelgacem/Hw_and_AES-projects/tree/main/AES%20Project

Credentials used in the project

Raspberry Pi

username - Jasmeet
password - root

MQTT Protocol

username - jas1
password - jasmeet

Adafruit IO

It was created using temp mail; therefore, it can become unavailable soon. The credentials of temp mail are:

username - niwera
password - xzNyY%A-H5g4X8&