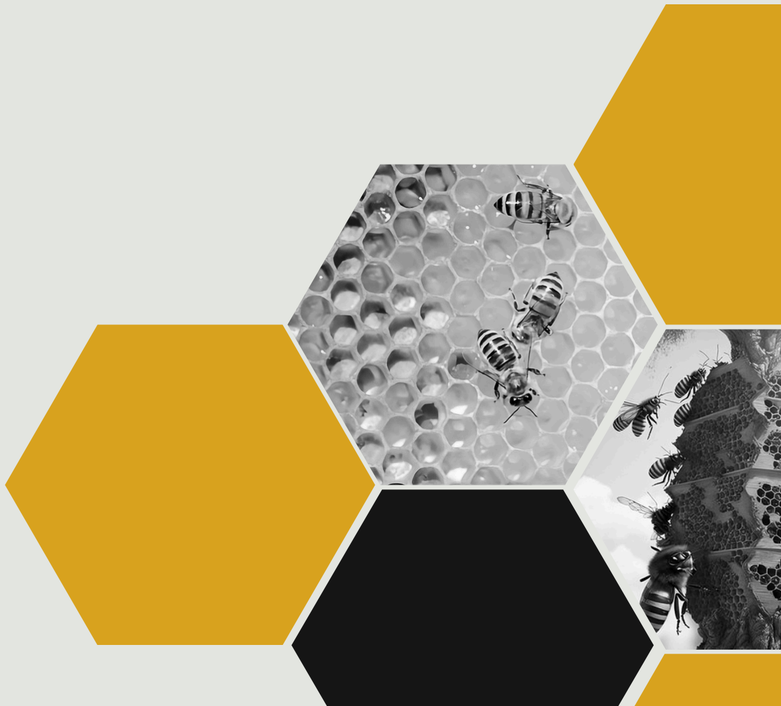
HIVE SENSE

Centrul Județean de EXCELENȚĂ Vaslui

MEREUȚĂ ROBERT
LICEUL TEORETIC “EMIL RACOVIȚĂ”
CHIRIȚESCU DENIS
LICEUL TEORETIC “MIHAIL KOGĂLINCEANU”



ROMANIAN BEEKEEPING

In Romania, beekeeping has a long-lasting tradition. Our country being a runner-up in Europe regarding honey production. Thanks to natural resources being so abundent and the floral diversity

COLONY COLLAPSE DISORDER

Colony Collapse Disorder- The Bee Families depopulation sindrome is an affection in which most of the working bees from a colony suddenly and inexplicably disappear, only leaving behind little bees and larvae.
The exact causes aren’t fully discovered yet, but it is believed that factors such as pesticides, parasites and stress can be a contributing factor of this problem.
CCD has serious consequences for the bee industry and can affect plant pollination and global food production.

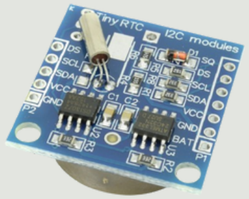
GENERAL OVERVIEW

Colony Collapse Disorder was first encountered in the United States of America in 2006, this affection being signaled in many countries worldwide, including Europe, such as: Germany, France, Spain, Italy. Romania hasn’t got any report of this case, but prevention is the best cure! So this is an autonomous diagnosis system which can be installed right near to a bee colony and can measure certain parameters of the colony in their active period (March to November). The system measures the total weight of the colony in the whole period just as it does for the termperature and humidity from the inside of the hive. Moreover it measures the frequency of the flight of the bees so it can detect if the queen is making the tooting sound, it means that the hive is swarming. The data is then sent periodically to an MQTT server from where they are extracted and processed to evaluate and, eventually to emit alerts. The independent system is composed of a scale and a probe which takes the said measures. For an energy source, it is completely independent and sustainable, being equipped with a solar panel which ensures charging the battery and the greater part of the system consumption, one of the main goal being to save energy (to consume the least amount possible). There is only a short period of time in which communication can be made, so it can consume less energy doing so, each module being powered only in the moment of use.

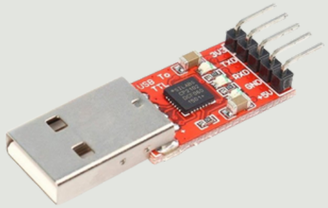
OBIECTIVES

1. Generation of alerts in case of a quick depopulation
2. Prevention or combat CCD
3. Measuring health and diagnosing the hive
4. Data analysis regarding climate change

Components

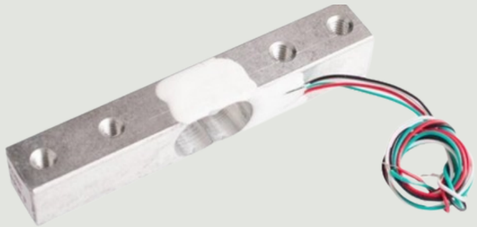


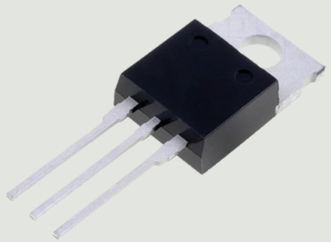
-SIM800L(GSM module);
-AHT10(temperature and humidity sensor);
-STM32F1C8(STM 32 processor);
-HX711(loadcell amplificator module);
-DS1307(RTC-Real Time Clock module);



-Sound Sensor;
-Arduino Mega2560;
-SD Card Module.

-Voltage step-down module;
-USB to UART convertor;
-4 tranzistors MOSFET P channel;
-1000µF Condensator;
-LoadCell;
-2.7 kΩ / 10kΩ Rezistor;
-Support;
-Raspberry Pi 3;





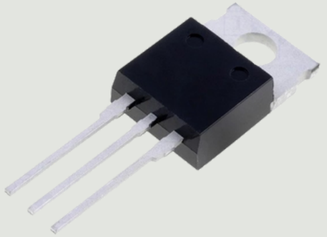
ELECTRICAL SIDE

Power

The power module contains only 2 3.7V Li-ion batteries connected in parralel and a voltage boost module which allows battery charge using a solar panel.

Using the voltage step down module the voltage is set to 4.2V to be able to power the SIM800l module which only works between 3.9V and 4.2V.

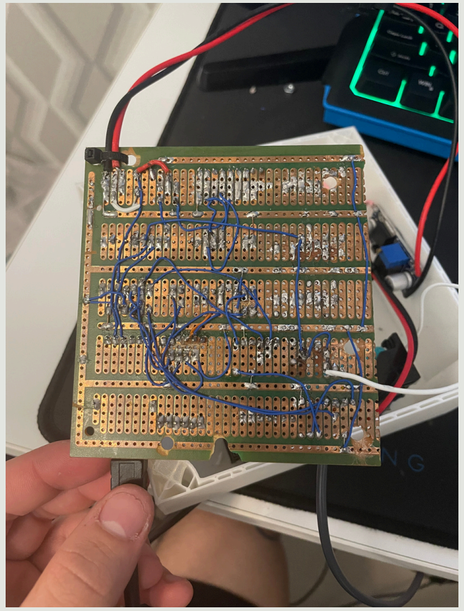
Given that the GSM SIM800L module draws a high amplitude current (1A) upon startup, it's crucial to ensure minimal voltage fluctuations or losses.

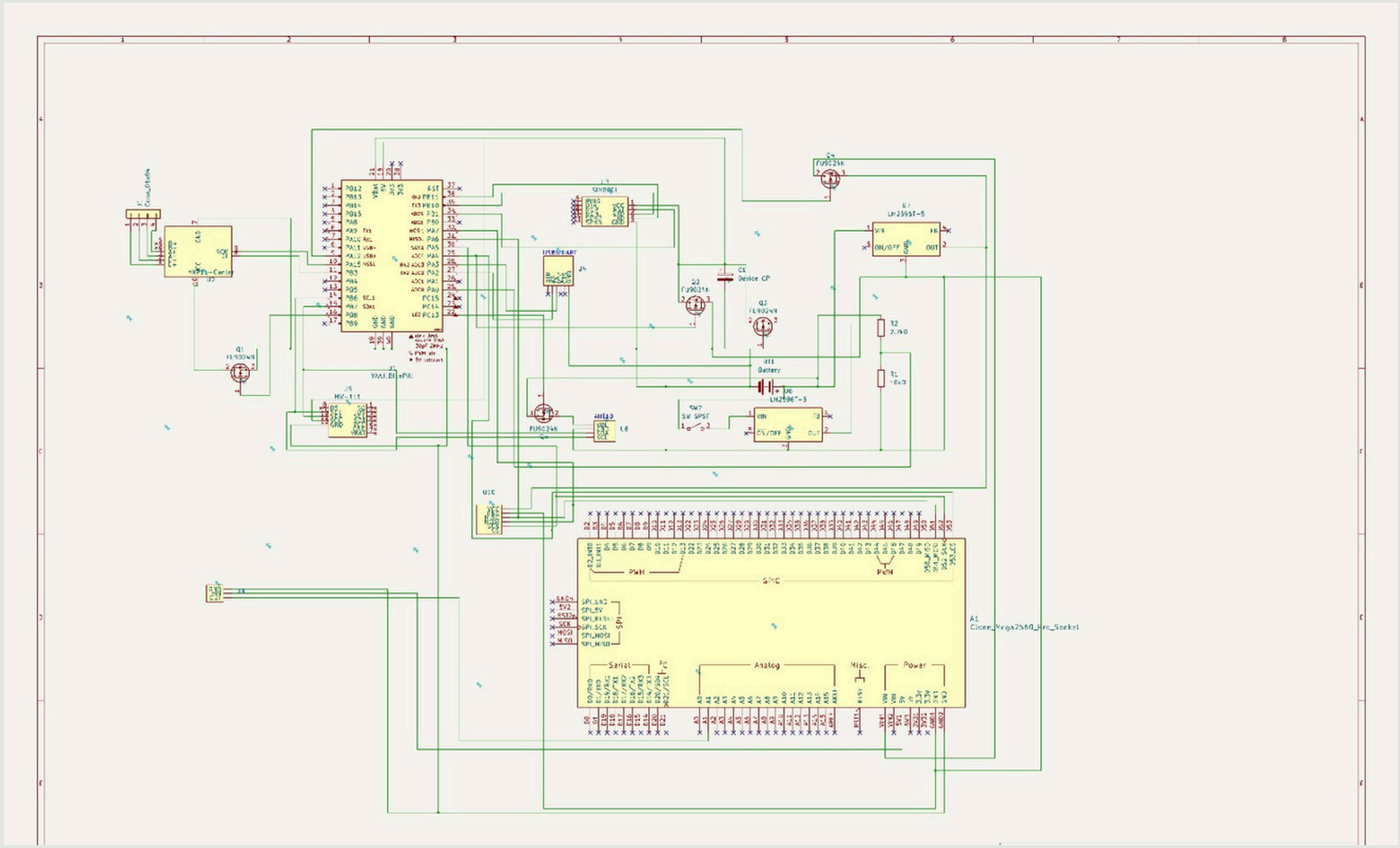


Reducing Voltage drops

To reduce voltage drops or fluctuations, we used MOSFET transistors instead of diodes for protection.

"The diodes can cause voltage drops of up to 0.6V, whereas the transistors have negligible losses (less than 0.1V)."





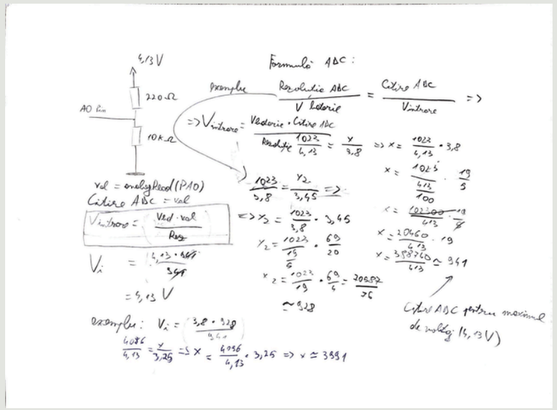


fig.2

ELECTRICAL PART

- **Energy Consumption**

Usually, the scale uses around 0.3A but ,while connecting, it can use up to 2A. However, while in the sleep mode the consumption is reduced to 7mA.

$$ReLU$$

$$f(x) = max\,(0, x)$$

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

- **AI**

Artificial intelligence, namely Deep Learning, was used to detect the sudden changes in the graph that displays the detection of the sound frequency in the hive. Two types of neural networks (CNN and DNN) were used. The convolutional neural network (CNN) has two-dimensional layers and detects the graph as a picture. The deep neural network (DNN) detects the graph values. For the layers activation ReLU function was used and for the output layer the sigmoid function was used.

CNN

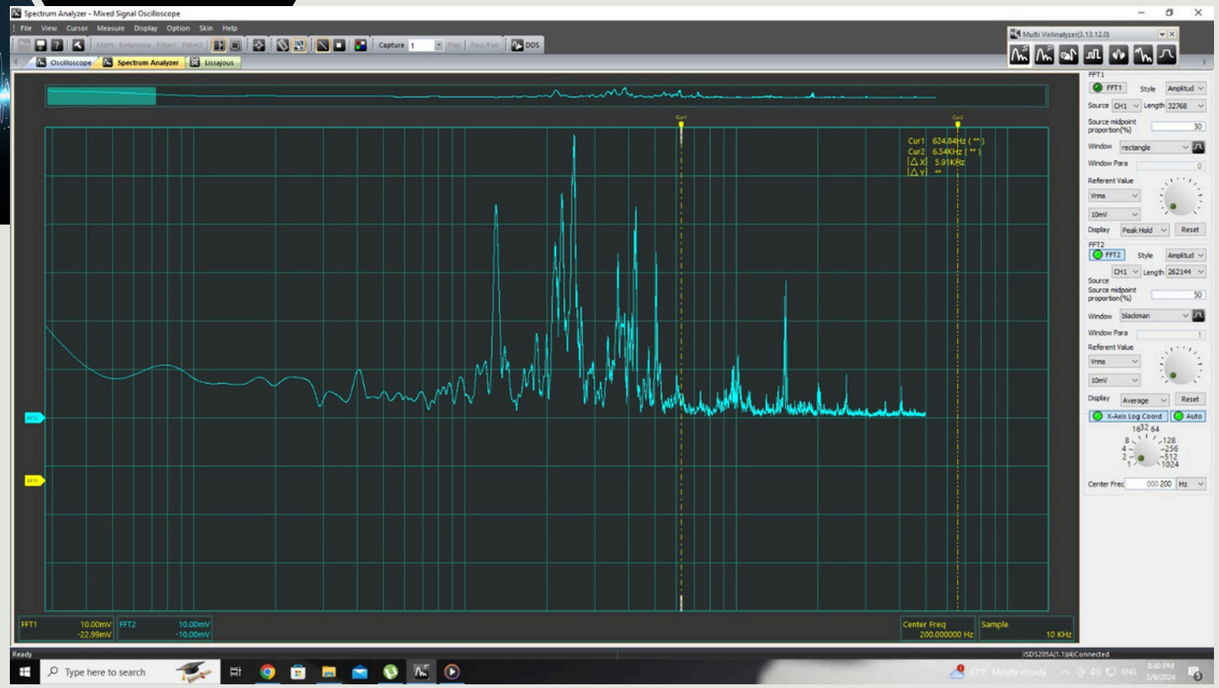
```
model = keras.Sequential([
    keras.layers.Conv2D(32, (3, 1), activation='relu',
padding='same', input_shape=(15, 1, 1)),
    keras.layers.MaxPooling2D((2, 1)),
    keras.layers.Conv2D(64, (3, 1), activation='relu',
padding='same'),
    keras.layers.MaxPooling2D((2, 1)),
    keras.layers.Conv2D(128, (3, 1), activation='relu',
padding='same'),
    keras.layers.MaxPooling2D((2, 1)),
    keras.layers.Flatten(),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(1, activation='sigmoid')
])
```

DNN

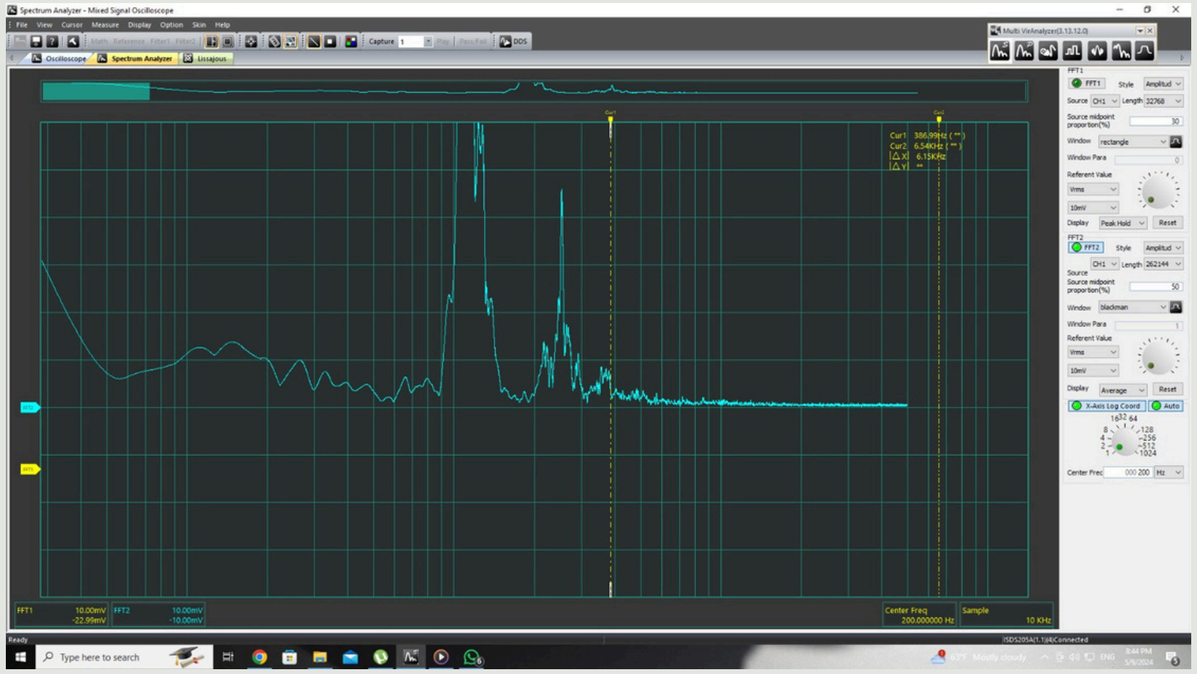
```
model = keras.Sequential([
    keras.layers.Input(shape=(15,)),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(1, activation='sigmoid')
])
```

Sound Processing

An Arduino Mega2560 processor is used to process the sound data, it takes the analog data from the sensor pin and transforms it into frequency using FFT (Fourier Transform). It receives 15 values in the frequency series, after which it adds them to the "DATA.txt" file on the SD card.



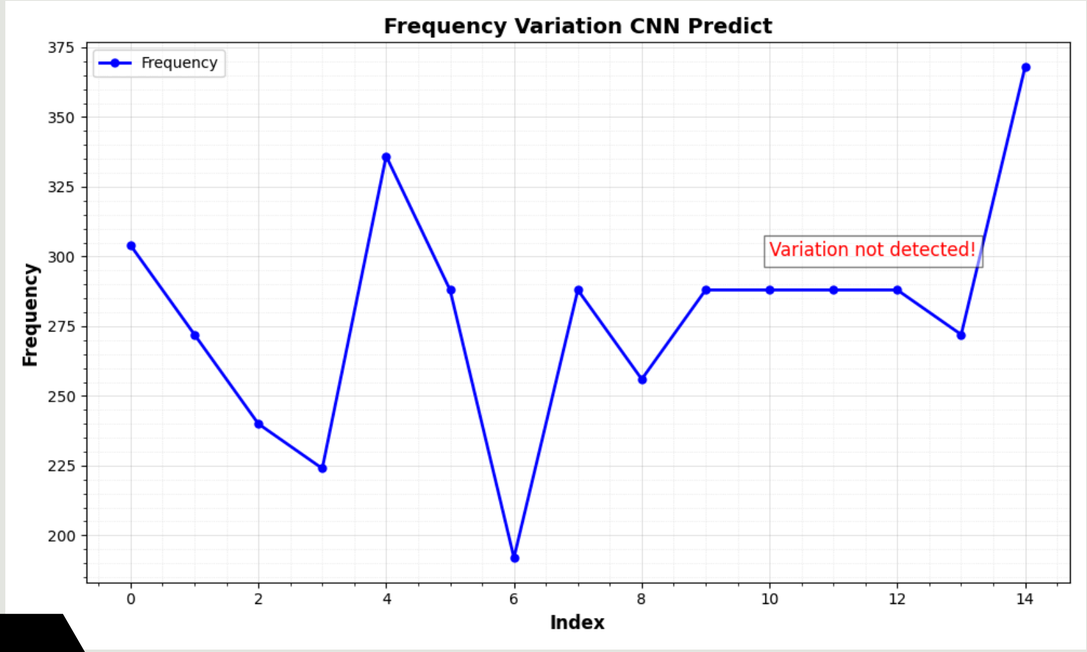
Oscilloscope data on swarm



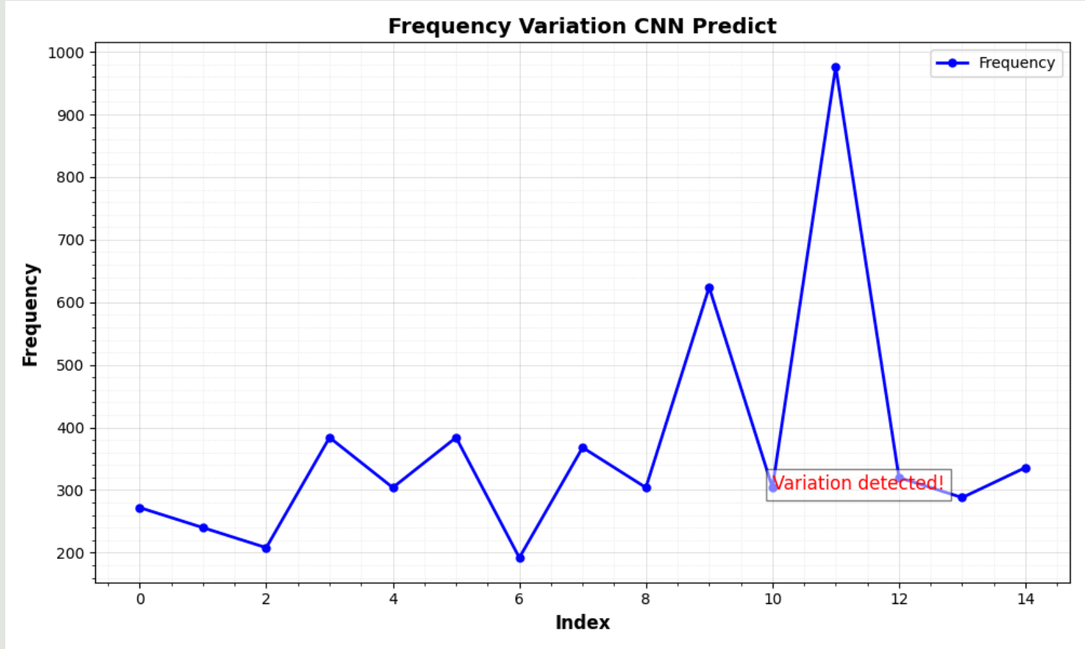
Oscilloscope data normally

Data Collecting

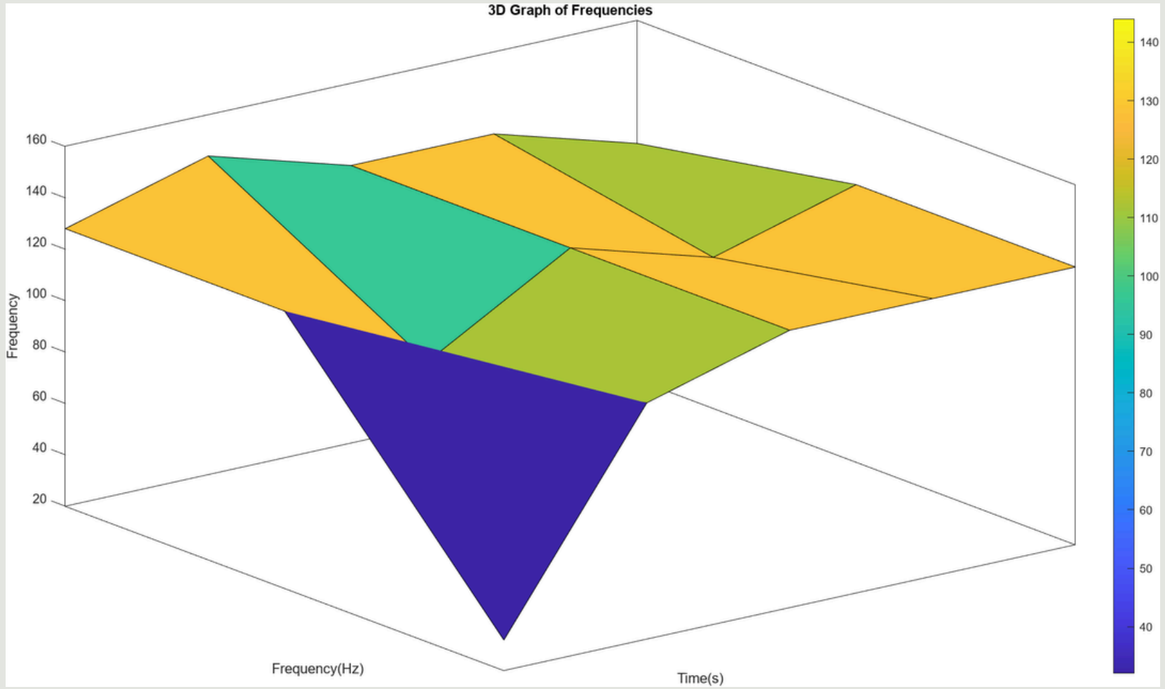
The data is real and have been taken from a hive so that we can see the real data on the hives provided by the hive in normal harvest conditions, but also in swarming conditions.



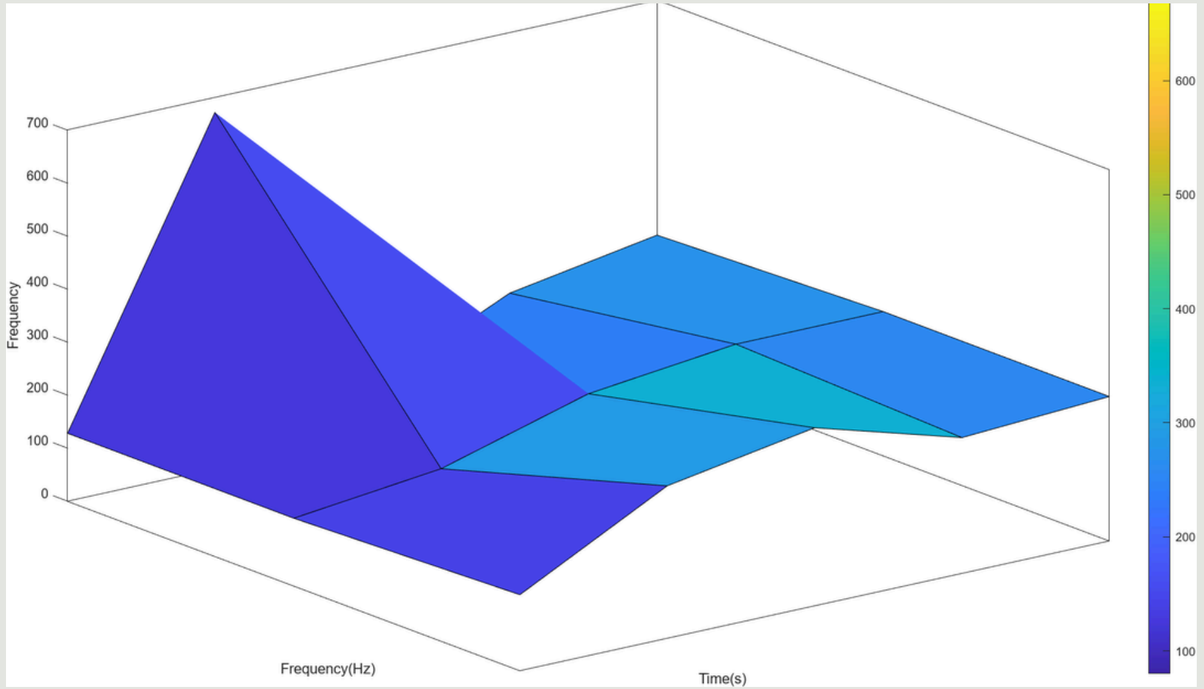
Normal graph without swarming



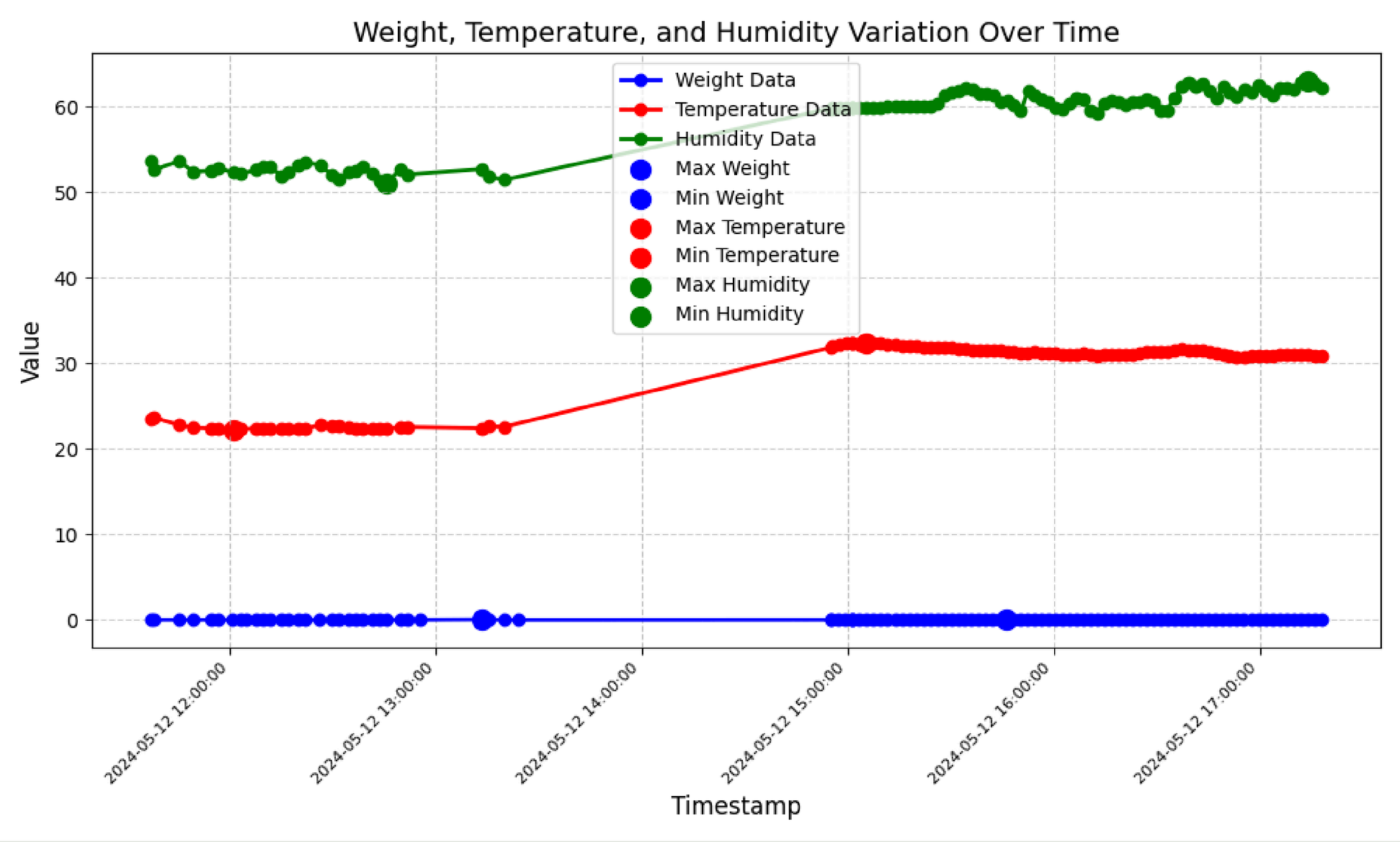
Swarm graph



Graphics without swarming



Swarm graph



Weight, temperature and humidity data

STATE MACHINE

State Machine operation is divided into 5 windows:
Status 0: The robot checks if the GSM module is functional.
State 1: The signal level of the GSM module is checked.
State 2: The scale is ready and records weight, temperature and humidity.
State 3: The scale enters sleep and resets the weight.
State 4: The GSM module connects to mobile data.
State 5: The connection to the MQTT server is established, the data is published and the sound data is processed.

Other events during states:

- In window 4, an SMS is sent to announce the opening of the communication window.
- In window 5 we can send orders.

DATA STORING

To keep all the data sent by the diagnostic system, a Raspberry Pi 3 board will stay on constantly. It listens to the topics where the messages are sent, when it receives a message it stores it in a different database for each topic.

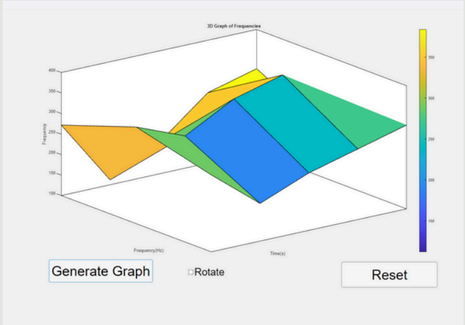
SOFTWARE

Arduino

The program behind the automatic bee diagnostic system is organized into a main loop that manages the various states of the device. In each state, certain actions are performed depending on the set conditions. For example, state 0 is checking GSM modem connection and signal, and state 2 is measuring weight and temperature/humidity and sending the data to an MQTT broker. Additionally, several helper functions are defined to manage MQTT hardware and connections.

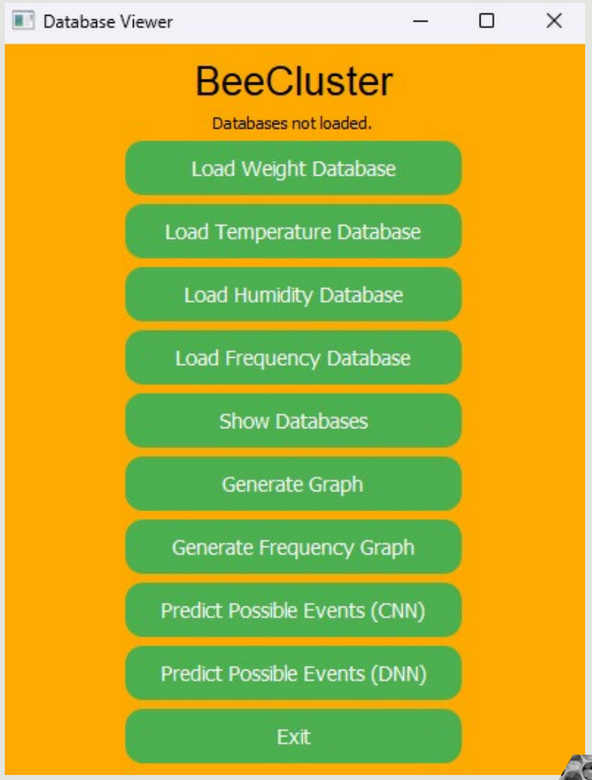
MATLAB

The program for viewing 3D graphs was made using the MATLAB software, it takes the last string of numbers from a text file where all the frequency data from the databases are stored and makes a 3D graph with them.



Python

The program behind the database visualization and data graphing application was written in Python using the PyCharm IDE software. The "PyQt5" library was used to create the graphic image of the code. To work with the databases, the "sqlite3" library was used, which uses the SQL language, in addition to Python. To store the data received by the scale, the Raspberry Pi 3 board is constantly on, when it receives a message on the mass and temperature topics it stores them in a different database. The "paho.mqtt" library was used to connect to the MQTT server, and the "sqlite3" library was used to write the databases, which uses both the Python language and the SQL language.



BIBLIOGRAPHY &SITOGRAPHY

https://ro.wikipedia.org/wiki/Apicultura_%C3%AEn_Rom%C3%A2nia#:~:text=Apicultura%20%C3%AEn%20Rom%C3%A2nia%20sau%20stup%C4%83ritul,cear%C4%83%20%C8%99i%20alte%20produse%20apicole.

https://www.itemis.com/en/products/itemis-create/documentation/user-guide/overview_what_are_state_machines

https://en.wikipedia.org/wiki/Colony_collapse_disorder
https://www.pnas.org/doi/abs/10.1073/pnas.1422089112
“Cum să menținem albinele sănătoase” de Wolfgang Ritter

https://cleste.ro/
https://ardushop.ro/
https://www.aliexpress.com/

https://www.techtarget.com/whatis/definition/analog-to-digital-conversion-ADC https://www.kicad.org/

https://www.omega.com/en-us/resources/load-cells
https://mqtt.org/
https://aws.amazon.com/what-is/mqtt/#:~:text=The%20MQTT%20broker%20is%20the,and%20sending%20them%20the%20messages.

https://www.autodesk.com/
https://www.raspberrypi.com/

https://www.arduino.cc/
https://www.st.com/content/st_com/en.html
https://www.st.com/en/development-tools/stm32cubeprog.html
https://www.python.org/
https://www.jetbrains.com/pycharm/?var=1

THANK YOU