# Smart Pet Feeder

*Escola Superior de Tecnologia de Setúbal*
*Licenciatura em Engenharia Eletrotécnica e de Computadores, Projeto em Internet das Coisas*

| | |
|---|---|
| Eduardo Silva [Database] | 202001449@estudantes.ips.pt |
| Guilherme Correia [Mobile App] | 202002058@estudantes.ips.pt |
| João Reis [Node-RED & 3D Model] | 200500970@estudantes.ips.pt |
| Manuel Lagarto [Firmware] | 202002147@estudantes.ips.pt |
| Rui Colaço [Hardware] | 199100317@estudantes.ips.pt |

## I. INTRODUCTION

Most of the time, pet owners spend their time away from home and away from their pet friends, whether during vacation or work, leaving them a certain quantity of food and water which may get spoiled or be spilled away accidentally by the animal.

To solve these problems was created the Smart Pet Feeder, which provides pet owners with an innovative way to keep their furry friends well-fed, from monitoring their feeding habits to scheduling portion sizes based on veterinary advice.

This solution consists of the main device, the Pet Feeder that pairs with the Central Hub through a wireless connection. With this method, pairing more than one Pet Feeder to a single Central Hub is possible.

The Pet Feeder can feed both food and water, where the food is stored in an internal dispenser with an integrated stepper motor for precision portion control, and the water is regulated through a valve connected to the house's water line. A set of sensors implemented in the Pet Feeder constantly monitors and transmits to the Central Hub the food and water levels in the bowls and dispenser, as well as the animal presence.

The Central Hub is implemented with a 7-inch display that displays each Pet Feeder's dashboard. These dashboards show the levels of each Pet Feeder, as well as the important notifications and scheduled portions. All the data needed by the Central Hub and the rest of the systems are stored in the integrated database.

A mobile app for Android users is also available for the Smart Pet Feeder. With this app, pet owners can schedule portions based on veterinary advice, monitor feeding habits, control the connected devices, receive push notifications, and many more features that make the Smart Pet Feeder more interactive.

To summarize, the Smart Pet Feeder has different features that increase the quality of life of pets and allow pet owners more ease in feeding and monitoring the animal, for example:

- Scheduled portions of food and water.
- Pet Feeder's state monitoring.
- Feeding habits monitoring.
- Mobile app notifications and warnings.
- Minimalist and compact design.
- Be integrated into the pet owner's house or in an institution.

## II. STATE OF THE ART

The industry of Smart Pet Feeders is increasing daily with many models appearing on the market with scheduling and monitoring capabilities, being one of the biggest takes of the Internet of Things revolution.

While most Smart Pet Feeders can only control the dispensed food based only a schedule defined by the pet owner, there are pet feeders that can also monitor the feeding habits, do self-tests, and integrate their data with the main Smart Home app. An example is the *XIAOMI® Smart Pet Feeder*.

On the other hand, one feature that all the Smart Pet Feeders available on the market share is providing pet owners the convenience and ease of feeding their furry friends, ensuring their pets are well-fed and hydrated even when their owners are away from home during vacation or work.

## III. SYSTEM DESCRIPTION

The Smart Pet Feeder is based on two main components, the ESP32, and the Raspberry Pi microcontrollers, used in the Pet Feeder and Central Hub devices (Figure 1), respectively.

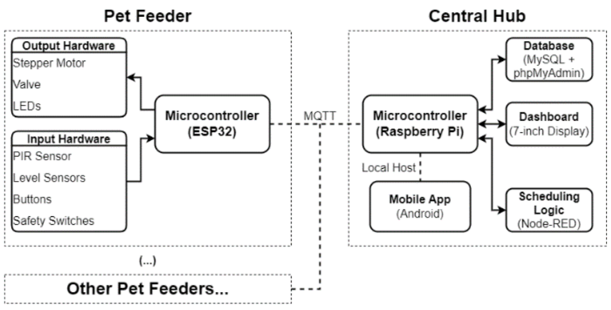


Figure 1 - Smart Pet Feeder Block Diagram.

The Pet Feeder is where the hardware to interface with the pet feeding process is implemented and has the following components: an ESP32, a Stepper Motor, a Valve, a PIR Sensor, two IR Level Sensors, an Ultrasonic Level Sensor, three LEDs, two Push Buttons and three Safety Switches.

The Central Hub is used to pair the installed Pet Feeders via the MQTT communication protocol and is where the scheduling control logic, Pet Feeder dashboard, and interface with the database are implemented. This device is composed of the Raspberry Pi and a 7-inch display.

### Hardware

The Pet Feeder hardware schematic (Figure 2) was designed using *KiCad*, an open-source EDA software for hardware and PCB design.

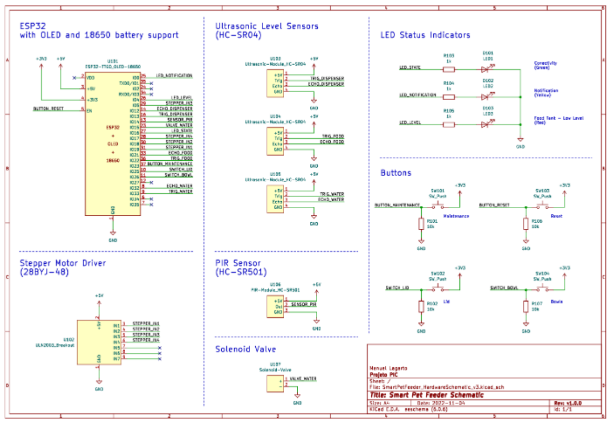

ESP32 with OLED and 18650 battery support
Ultrasonic Level Sensors (HC-SR04)
LED Status Indicators
Buttons
Stepper Motor Driver (28BYJ-48)
PIR Sensor (HC-SR501)
Solenoid Valve
Title: Smart Pet Feeder Schematic

Figure 2 - Pet Feeder electrical hardware schematic.

The main control board integrates the **ESP32 Board** socket and all the screw-type terminal connectors for the sensors and actuators signals, allowing seamless maintenance and integration.

To rotate the paddle wheel to therefore control the amount of food purged through the nozzle, the **28BYJ-48 Stepper Motor**, which pairs with the **ULN2003 Driver**, was selected. This Stepper Motor has four coil windings and delivers 64 steps per turn, corresponding to 5,625 degrees of precision. As the motor consumes about 217mA of current, the ESP32 board can't supply the driver so an external power supply of 5V DC is required.

To regulate the water feeding was selected a generic one-way **Solenoid Water Valve**. An external power supply of 12V DC is required to command this valve.

As it is necessary three types of external power supply – main control board (5V), Stepper Motor (5V), and Solenoid Valve (12V) – a **three-stage power supply** was designed (Figure 3). The selected ESP32 board already has a 3.3V regulator.

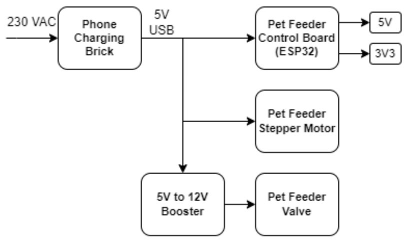


Figure 3 - Three-stage power supply.

For animal presence detection, the **HC-SR501 PIR Sensor** was selected, which uses an infrared LED. This sensor is placed at the front of the device and has 110 degrees of detection angle, supplied with 5V DC from the main control board.

Inside the Pet Feeder are placed three level sensors. Two of them are the **SHARP GP2Y0A21YK0F**, mounted above the food and water bowls to measure its levels with infrared technology. These sensors offer better precision for harsh environments but are more expensive, so an **HC-SR04 Ultrasonic Sensor** was mounted inside the dispenser to measure its level, which is a closed and more controlled environment.

A set of **LEDs** and **Buttons** is mounted on the side of the Pet Feeder to give a user interface to the pet owner. A green LED indicates the state of the connectivity, a yellow LED indicates an unread notification, and a red LED indicates the low level of food in the dispenser. One of the buttons does a manual purge of the food and the other resets the system.

Three **Safety Switches** are mounted to stop the system in case one of the lids or the bowls are removed from the Pet Feeder.

## Firmware

The firmware for the Pet Feeder is based on the dual-core ESP32 microcontroller and developed using the *C++* programming language and the *FreeRTOS* operating system [*www.freertos.org*], natively supported by the Arduino framework.

All the firmware was developed with the *Visual Studio Code* code editor, with the extension *PlatformIO* to give it IDE capabilities to upload and debug the program in the ESP32.

The *FreeRTOS* is a real-time operating system used in embedded systems that require multi-tasking operation, and this project is no exception.

For this firmware, a set of tasks were created where the tasks related to the input/output hardware are pinned to the ESP32 Core0 (Figure 4 & Figure 5), and the tasks related to the MQTT communication are pinned to the ESP32 Core1 (Figure 6). With this method, it's possible to guarantee that the MQTT connectivity with the Central Hub is never interrupted.

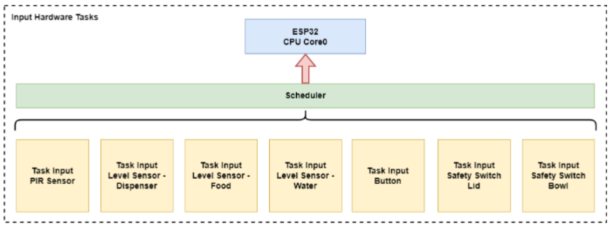


Figure 4 - Input hardware FreeRTOS tasks.

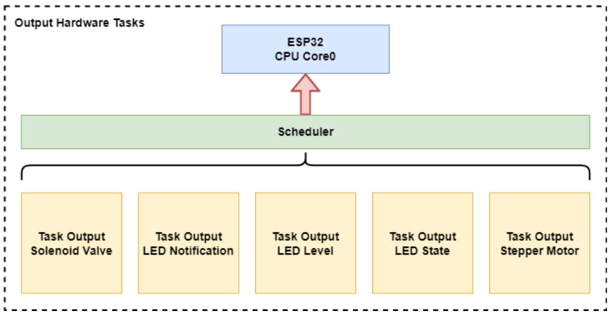


Figure 5 - Output hardware FreeRTOS tasks.

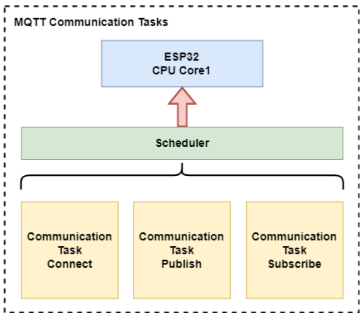


Figure 6 - MQTT communication FreeRTOS tasks.

To pass data between different tasks, a Queue Management API is available in *FreeRTOS*. With this method, it's possible to safely pass information between the MQTT communication tasks and the I/O hardware tasks, by implementing the *subscribe_queue* and the *publish_queue* (Figure 7). The *xQueueSend()* function sends data to the queue, and the *xQueueReceive()* function waits for data to arrive in the queue.

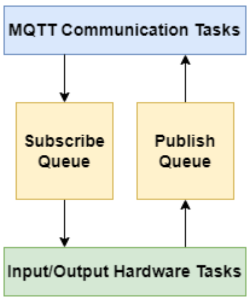


Figure 7 - Intertask communication diagram.

The MQTT protocol is largely used in IoT devices by using the server's IP address, in this case, the Raspberry Pi installed on the Central Hub, to communicate over a wireless network connection. Each packet of communication data is composed of a topic and a message, i.e., the sensor id and the sensor data, respectively.

As seen in Figure 8, the Firmware flowchart is illustrated, representing the control algorithms and the interactions between each task. The complete code developed for the Firmware is attached at the end of this article.

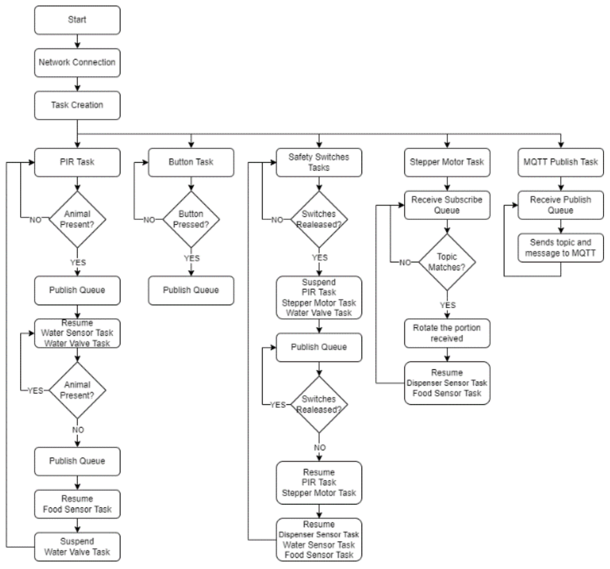


Figure 8 - Firmware's flowchart.

The interaction between each hardware task is done by the Task Control API available in *FreeRTOS*, which includes *vTaskSuspend()* and *vTaskResume()* functions. A task resumes its operation from the instant it was previously suspended.

## Database

The Database for the Smart Pet Feeder stores all the data needed for the schedules of portions, pet owner login, feeding habits, pet information, and other important data for the system operation.

The relational model (Figure 9) was developed with the ERDPlus online modeling tool (https://erdplus.com/). This model represents the tables (entities) used in the database to store the data, the data types of each attribute, and the relationships between each entity.

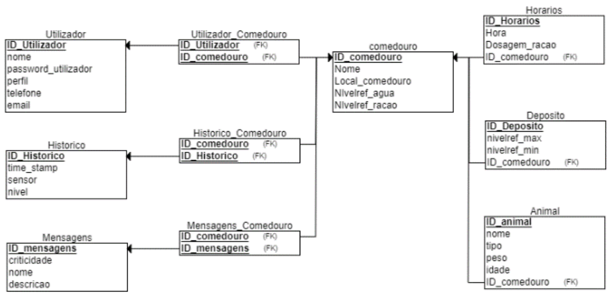


Figure 9 - Relation model planned for the Database.

Next, the Database was created with the MySQL Workbench software according to the previous model, using the DDL and DML languages.

To host and manage the created Database it was used the web application phpMyAdmin configured in the Raspberry Pi of the Central Hub, which makes use of *.php* scripts to a system interact with the Database. This application provides a web interface to manage and debug the Database which greatly eases the operation, creation, and maintenance of the Database.

## Mobile App

With the Mobile App pet owners can have a personal user login to access the paired devices to the Central Hub, see the Pet Feeder status, schedule portions of meals based on the veterinary advice or animal's metabolism, receive important notifications, and update the user information.

The app was developed with the *Android Studio IDE* using *Java*, *JavaScript*, *HTML*, and *CSS* programming languages. For the app was developed a flowchart (Figure 10) for the navigation between the different menus. After launching the app, the user will be prompted with the login screen that leads to the main page corresponding to the devices list. From this page, the user can access the side menu screen to access the other menus, such as schedule, feeder status, notifications, and devices list.

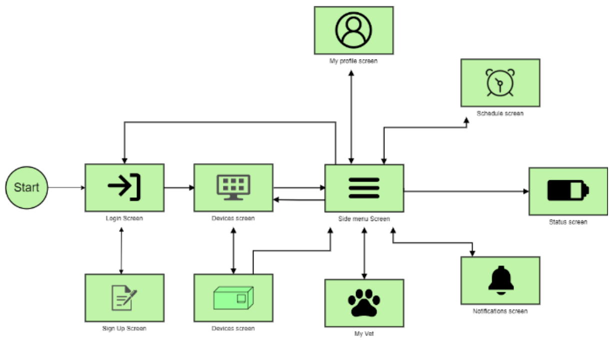


Figure 10 - Mobile App menu navigation flowchart.

To use this app, it's necessary to always be connected to the same Smart Pet Feeder network because the app is constantly communicating with the database through the PHP functions, to realize queries and updates in the Database.

Figure 11 represents the user registration and login page menu in the app where the pet owner must insert his personal information.

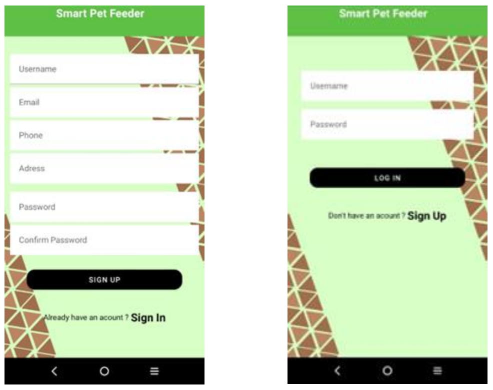


Figure 11 - Mobile App user registration and login page.

Figure 12 represents the Mobile App side panel where the user can navigate through all the menus available in the app.

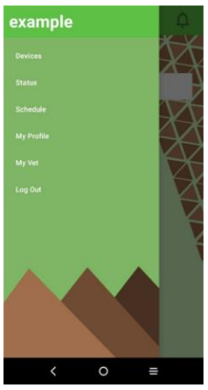


Figure 12 - Mobile App side panel.

Figure 13 represents the pairing process of a Pet Feeder to the Central Hub, by scanning its QR Code, and the menu of the paired device list.

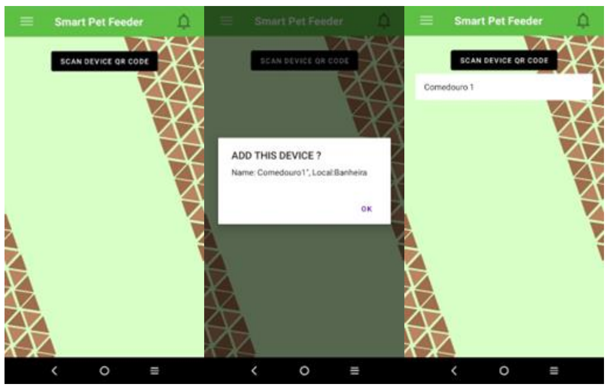


Figure 13 - Mobile App pairing process of a Pet Feeder.

Figure 14 represents the Pet Feeder's status page where is shown the assigned name, bowls, and dispenser levels and its home location.

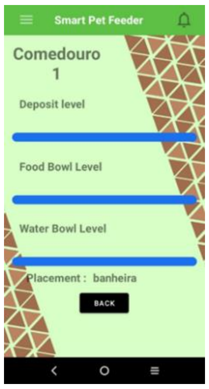


Figure 14 - Mobile App Pet Feeder's status page.

Figure 15 represents the animal's scheduled meals where a new one can be added by inserting its time and portions.

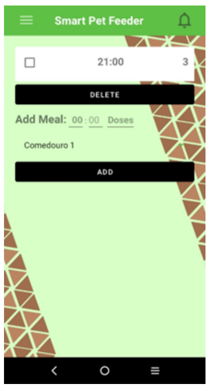


Figure 15 - Mobile App scheduled meals page.

Figure 16 represents the page to associate each animal to each Pet Feeder, by inserting their information and the pretended Pet Feeder id. On this page is also possible to view a list of the already associated animals to the app.

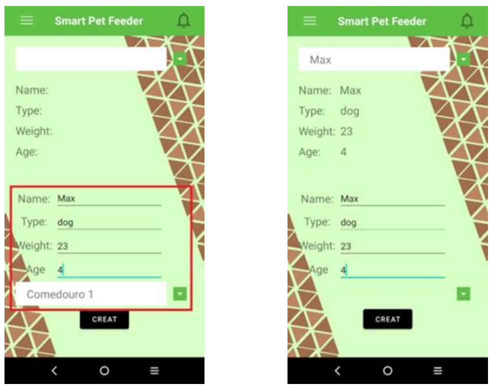


Figure 16 - Mobile App animal association page.

[Show the menu flow diagram. **Explain each app menu**]

**Software (Node-RED)**

Node-RED was used to program the control logic for the scheduling portions and other features of the Smart Pet Feeder. This tool is a web-based programming tool to connect different hardware devices to different APIs and online services.

For the control logic was developed a flowchart (Figure 17 & Figure 18) in which is represented the subscribe and publish functions from the MQTT, and the query/update process on the Database.

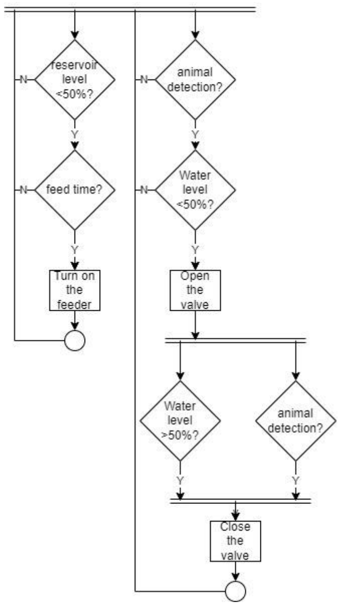


Figure 17 - Node-RED control logic flowchart, part I.

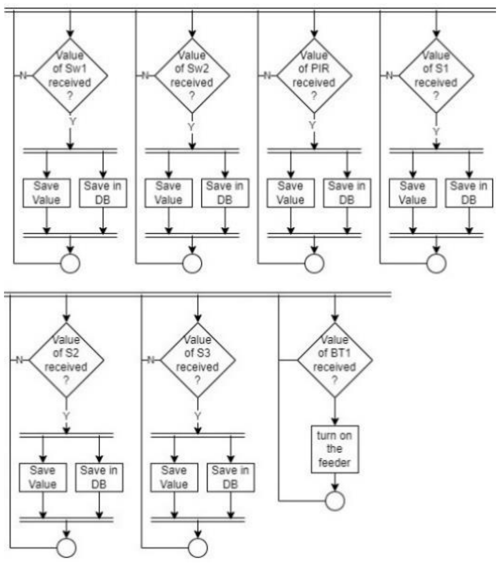


Figure 18 - Node-RED control logic flowchart, part II.

The control logic (called flows in the Node-RED) was programmed using various types of nodes available from the palette in the development environment of the programming tool. For this project, the two most important types of nodes were the "*in/out mqtt*" node and the "*mysql*" node, used to communicate with the Pet Feeder and the Database, respectively.

## 3D Model

The 3D Model of the Pet Feeder was designed with the *Autodesk® Inventor* CAD software and manufactured using a 3D printer with PLA filament.

In Figure 19 is represented the 3D front view of the Pet Feeder where the food bowl is located on the left and the water bowl is located on the right.

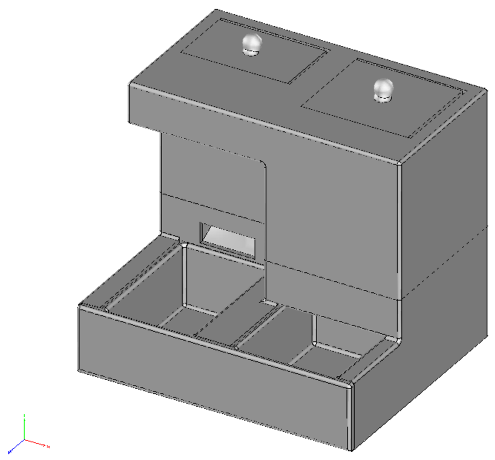


Figure 19 - Front view of the Pet Feeder's 3D model.

In Figure 20 the same view is represented but with transparent faces to show the location of the dispenser and the feeding mechanism inside the Pet Feeder.

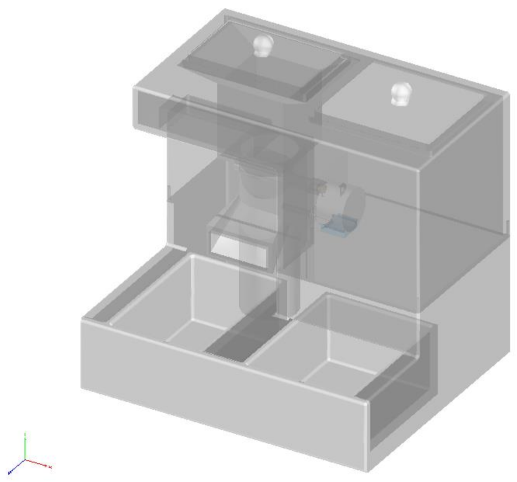


Figure 20 – Transparent view of the Pet Feeder's 3D model.

In Figure 21 is represented a cut section of the side view of the Pet Feeder to represent the feeding mechanism under the dispenser.

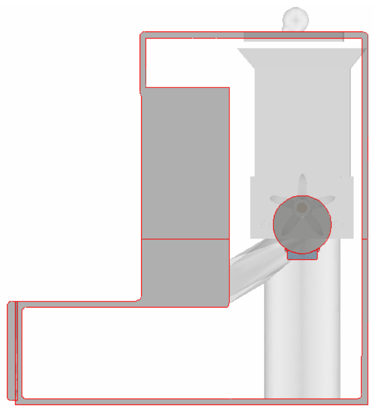


Figure 21 - Section cut of the Pet Feeder's 3D Model.

This mechanism is directly driven by the Stepper Motor and consists of a paddle wheel with five equally spaced blades that represent each portion of the animal's food.

After collecting the food from the dispenser with the paddle wheel, the food is directed to a channel that leads to the front of the Pet Feeder, falling into the food bowl right under the exit.

Both the food and the water bowls are removable and integrated into the same part which eases the cleaning process.

## IV. FINAL RESULTS

After 3D printing all the parts needed for the Smart Pet Feeder, the Pet Feeder's model was assembled (Figure 22) with all the electrical hardware wired together and mounted inside (Figure 23).

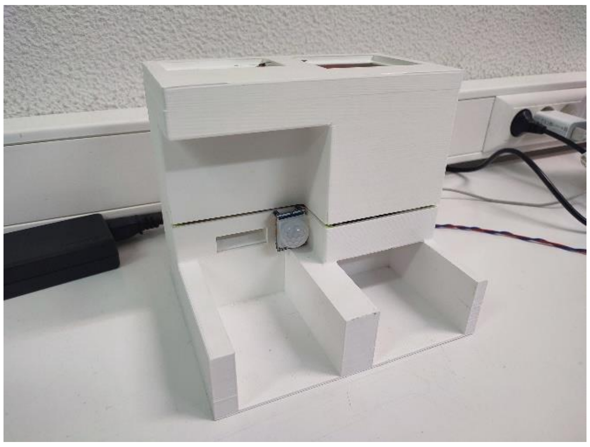


Figure 22 - Pet Feeder assembled.

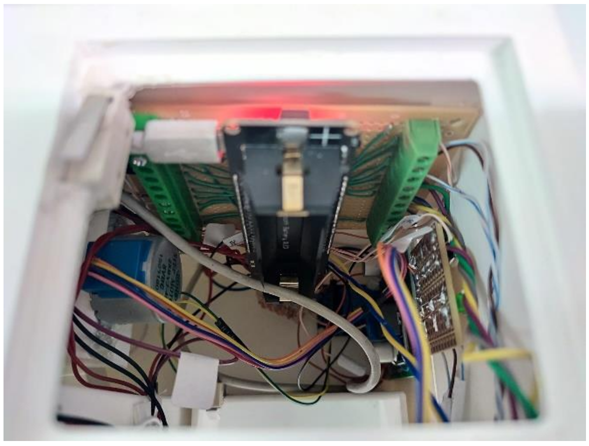


Figure 23 - Electrical hardware wired inside the Pet Feeder.

Figure 24 shows the LEDs and Buttons assembled on the side of the Pet Feeder, indicating the connected state (Green LED), low food level (Red LED), unread notifications (Yellow LED), manual feed (top button), and reset (bottom button).

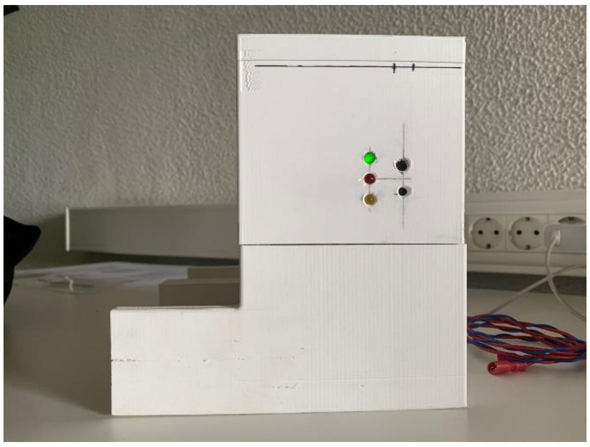


Figure 24 - LEDs and Buttons mounted on the side.

## V. CONCLUSIONS & FUTURE UPDATES

With the development of this project, it was possible to acquire all the skills needed to develop an IoT product from the ground up.

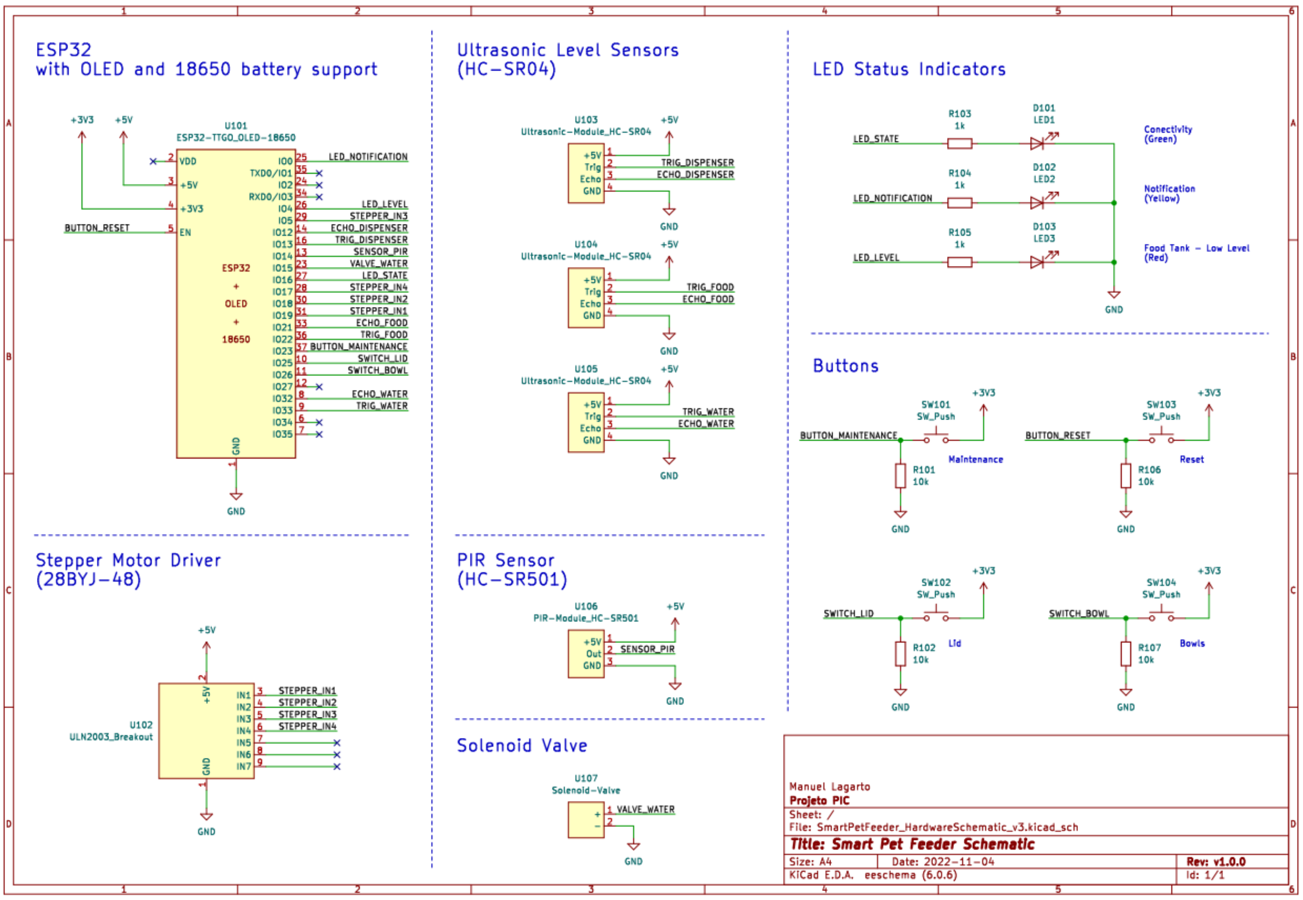
This project addressed different topics that would not be possible to approach in this course. Hardware design where a schematic was designed with all the selected hardware for the project needs. Firmware development where the group was introduced to the *FreeRTOS* real-time operating system for the first time, which is used for an embedded system with multi-tasking operation. Database design by modeling an entire database according to the data needed for the project. Android app development where the group used Android Studio with different programming languages to develop the Mobile App for the pet owner to interact with the Smart Pet Feeder. Introduction of the Node-RED programming tool which was useful to integrate the different hardware devices with different APIs (Database for example). 3D Model development to 3D print the Pet Feeder concept model.

For the next iteration of the project, the Hardware needs to be integrated into a PCB board to achieve a more compact and eliminate noise in the wiring connections. The stepper motor also needs to be upgraded to a more powerful one, since it was detected a lack of force in the rotation of the paddle wheel during the tests.

## VI. REFERENCES

1. KiCad EDA [*Link*]
2. Visual Studio Code [*Link*]
3. *FreeRTOS* documentation: [*Link*]
4. ERDPlus database modeling tool [*Link*]
5. Andriod Studio IDE [*Link*]
6. Node-RED programming tool [*Link*]
7. 28BYJ-48 Stepper Motor datasheet [*Link*]
8. ULN2003 Driver datasheet [*Link*]
9. SHARP GP2Y0A21YK0F datasheet [*Link*]
10. HC-SR04 Ultrasonic Sensor datasheet [*Link*]
11. HC-SR501 PIR Sensor datasheet [*Link*]

## VII. ATTACHMENTS


ESP32
with OLED and 18650 battery support
+3V3
+5V
U101
ESP32-TTGO_OLED-18650
VDD
+5V
+3V3
EN
BUTTON_RESET
ESP32 + OLED + 18650
IO0 LED_NOTIFICATION
TXD0/IO1
IO2
RXD0/IO3
IO4 LED_LEVEL
IO5 STEPPER_IN3
IO12 ECHO_DISPENSER
IO13 TRIG_DISPENSER
IO14 SENSOR_PIR
IO15 VALVE_WATER
IO16 LED_STATE
IO17 STEPPER_IN4
IO18 STEPPER_IN2
IO19 STEPPER_IN1
IO21 ECHO_FOOD
IO22 TRIG_FOOD
IO23 BUTTON_MAINTENANCE
IO25 SWITCH_LID
IO26 SWITCH_BOWL
IO27
IO32 ECHO_WATER
IO33 TRIG_WATER
IO34
IO35
GND
Ultrasonic Level Sensors
(HC-SR04)
U103
Ultrasonic-Module_HC-SR04
+5V Trig Echo GND
TRIG_DISPENSER
ECHO_DISPENSER
U104
Ultrasonic-Module_HC-SR04
TRIG_FOOD
ECHO_FOOD
U105
Ultrasonic-Module_HC-SR04
TRIG_WATER
ECHO_WATER
LED Status Indicators
LED_STATE R103 1k D101 LED1 Conectivity (Green)
LED_NOTIFICATION R104 1k D102 LED2 Notification (Yellow)
LED_LEVEL R105 1k D103 LED3 Food Tank - Low Level (Red)
Buttons
BUTTON_MAINTENANCE SW101 SW_Push Maintenance R101 10k
BUTTON_RESET SW103 SW_Push Reset R106 10k
SWITCH_LID SW102 SW_Push Lid R102 10k
SWITCH_BOWL SW104 SW_Push Bowls R107 10k
Stepper Motor Driver
(28BYJ-48)
U102
ULN2003_Breakout
IN1 STEPPER_IN1
IN2 STEPPER_IN2
IN3 STEPPER_IN3
IN4 STEPPER_IN4
IN5
IN6
IN7
PIR Sensor
(HC-SR501)
U106
PIR-Module_HC-SR501
+5V Out GND
SENSOR_PIR
Solenoid Valve
U107
Solenoid-Valve
VALVE_WATER
Manuel Lagarto
Projeto PIC
Sheet: /
File: SmartPetFeeder_HardwareSchematic_v3.kicad_sch
Title: Smart Pet Feeder Schematic
Size: A4
Date: 2022-11-04
Rev: v1.0.0
KiCad E.D.A. eeschema (6.0.6)
Id: 1/1