PawPal SmartFeeder - Group 15

34365 IoT Prototyping E24

Jorge Guixà González
MSc. Electrical Engineering
s243218@dtu.dk

Tomás Ruiz Aybar
MSc. Comm. Technologies and System Design
s242887@dtu.dk

Niyaz Mahmud Sayem
MSc. Comm. Technologies and System Design
s240041@dtu.dk

Zuhaib Zulfiqar
MSc. Electrical Engineering
s223296@dtu.dk

Marius Dornonville de la Cour Schiller
MSc. Design and Innovation
s175037@dtu.dk

Aysha Rahman Ratry
MSc. Comm. Technologies and System Design
s232194@dtu.dk

***Abstract*—The PawPal SmartFeeder is an innovative IoT-based pet feeding system that combines automation, real-time monitoring, and energy efficiency to enhance pet care. The system employs a modular design, integrating sensors, actuators, and two microcontrollers (Arduino Uno and Pro Mini) for precise food and water dispensing. Connectivity is ensured via LoRaWAN, facilitating long-range, low-power communication and data visualization through Azure IoT Hub. The physical prototype leverages laser-cut MDF and selective 3D printing to achieve cost-effectiveness and scalability. This paper discusses the system architecture, CAD modeling, physical construction, software implementation, and market analysis, positioning PawPal as a sustainable and inclusive solution for modern pet owners. Features like advanced health monitoring, remote management, and inclusive design make it a compelling choice in the evolving IoT pet care market.**

I. Introduction

Given the need for more complete and detailed control over pet's health, it was decided to design these smart bowls to feed them in a simple and precise way. This eliminates the possibility of forgetting to refill both water and food on routine days, as well as being able to manage them remotely when owners are unable to physically do so.

The setup will mainly consist of two bowls, one for water and one for food, each with an exclusive sensor to monitor both contents, a water level sensor (HW-038) and a LDR sensor respectively. These sensors will therefore work together to warn and order the bowls to be refilled through the mechanical systems of each tank, the water one working through a valve (servo motor) and the food bowl being activated by a stepper motor. All this information will be available to the user through the Azure interface via the LoraWAN TTN network and also a possible physical display in the setup itself as a UX.

II. System Setup and Diagrams

A. Circuit and Power Consumption, Jorge - s243218

The PawPal Smart Feeder employs a modular electronic design, integrating a combination of sensors, actuators, and communication modules to automate the feeding process while providing real-time telemetry data. The electronics architecture is structured around two microcontrollers: an **Arduino Uno** and an **Arduino Pro Mini**, which work collaboratively to process data and control the hardware components.

1) Electronic Components:

- **Microcontrollers**:
  - The **Arduino Uno** serves as the main controller, managing sensor inputs and actuator outputs.
  - The **Arduino Pro Mini** is dedicated to communication tasks, interfacing with the LoRaWAN module to transmit telemetry data to the cloud through The Things Network (TTN) and Azure IoT Hub.
- **Sensors**:
  - **Light Dependent Resitor (LDR)**: Monitors if food is left on the bowl.
  - **Water Level Sensor**: Tracks water availability in the water bowl.
  - **Ultrasonic Sensor**: Measures the amount of food left in the reservoir.
- **Actuators**:
  - A **stepper motor** and it's controller operates the food dispensing system with precision.
  - A **servo motor** controls the water flow to the bowl.
- **Communication Module**:
  - The **LoRaWAN module** ensures low-power, long-range wireless connectivity, enabling remote monitoring and data logging.
- **Display**:
  - A **16x2 LCD screen** provides a local user interface to display system status, including food and water levels.

2) Wiring: The electronic components are interconnected as illustrated in Fig. 1, ensuring efficient data flow and power management. The system operates on a 5V DC power supply, with shared ground connections prevent electrical noise and maintain system stability.

This system is designed to be placed indoors and connected to standard household electrical outlets. However, the implementation of an AC/DC rectifier, step-down converter and

Identify applicable funding agency here. If none, delete this.

filtering to adapt the mains power to the system's requirements was not covered under the scope of this project.

The modular design allows flexibility for future enhancements, such as integrating additional sensors or adapting the system for alternative power sources.

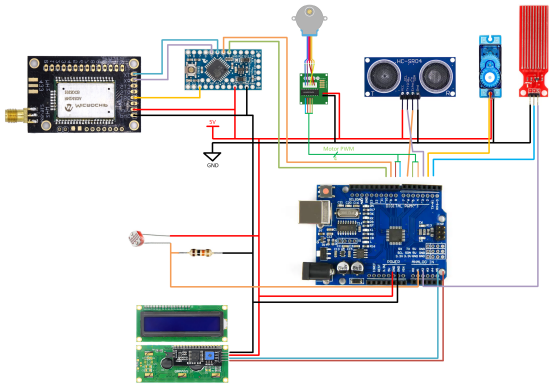


Fig. 1. Electronics Diagram.

3) Power Consumption and Management: The PawPal Smart Feeder is designed to operate with energy efficiency, minimizing power usage while ensuring reliable performance. Key contributors to power consumption include the motor, sensors, and communication module. As the system is not battery-powered and relies on household electrical outlets, power consumption does not impose significant restrictions on the functionalities, allowing all components to operate optimally.

Estimated Consumption:

- **Arduino Uno:** Implements a sleep mode during idle states, drastically reducing power consumption to as low as 5 mA.
- **Arduino Pro Mini:** Operates at 10 mA in active mode and remains always awake to ensure it is ready to read LoRaWAN information and facilitate communication when the Arduino Uno wakes up from sleep mode. This ensures seamless operation and communication between the modules.
- **LoRaWAN Module:** Consumes approximately 50 mA during transmission bursts, while remaining idle in low-power mode.
- **Sensors (combined):** Require 15 mA during operation.
- **Actuators:** The stepper and servo motors exhibit peak current draws of 100-200 mA during activation.

Energy Optimization:

- Duty cycling of sensors and actuators minimizes power usage by activating them only when necessary.
- LoRaWAN's low-power protocol enhances efficiency for IoT data transmission, ensuring prolonged operation without excessive energy demands.

This system is designed to be placed indoors and connected to standard household electrical outlets. However, the implementation of an AC/DC rectifier and step-down converter to adapt the mains power to the system's requirements was not covered under the scope of this project. With an estimated total power consumption of approximately 500-1000 mAh based on the Arduino Uno waking up and refilling both bowls twice per hour, sensors and actuators operating briefly during those periods, and the Arduino Pro Mini remaining continuously active—the system ensures energy efficiency while leveraging mains power. The modular and low-power design aligns with eco-friendly goals and allows for future integration of alternative power sources.

B. CAD Modelling of the physical prototype, Marius s175037

The physical prototype were made in four parts:

- The foodbox that contain and dispenses food, and houses the Ultrasonic Sensor
- The food bowl, that catches and presents the food, and houses the LDR
- The waterbox that contain and dispenses water
- The water bowl that catches and presents the water, and houses the Water Level Sensor

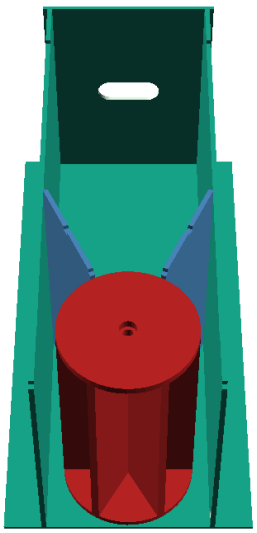
The two bowls were made out of paper, and as such didn't need to be modelled. Everything else were modelled parametrically, such that each dimension is determined by either an easily changable variable, or a series of geometric formulas. This way, it was possible to only decide on the final sizes and ratios at the end.

a) Foodbox: The foodbox controls the flow of food using a wheel (pictured below in red) that would be rotated by the stepper motor. The wheel consist of 6 compartments. When generating the wheel, the CAD model would calculate the volume of each compartment, and in the case of the final prototype, they could hold 95.3483 mL. This design allows the dispensing mechanism to dispense a consistent amount of food with every 60 degree rotation. The wheel would be held up by the motor in one end, and a bolt in the other, and holes were made both in the plates and the wheel to accommodate for these.

In order to guide the food into the compartments of the wheel, a funnel was added (pictured in blue). For the funnel, the angle of it were parameterized, and the actual dimensions were calculated by the model, to make sure they fit.

In the top plate, a slot were made to hold the Ultrasonic Sensor.

The plates would be joined using taps and slots that were cut where relevant.

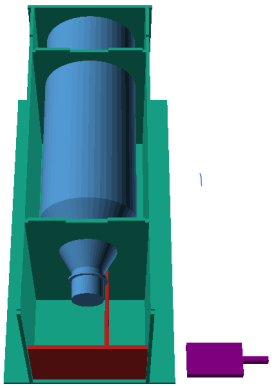


b) Waterbox: In order to make the waterbox simpler to construct, it were designed to hold an upside down bottle, that would act as a watertight compartment for water storage (pictured below in blue). In order to control the flow, a proper valve would've been preferred, but due to budgetary limitations, it were done with a paddle (pictured next to the box in purple) controlled by the servo motor.

The servo were held in place with some small plates (pictured in red).

In order to make sure the paddle would interface properly with the servo motor, an attachment gear were found on Thingiverse [7], which the rest of the paddle were built around.

Similarly to the foodbox, the plates were joined using slots and taps where relevant.



C. Physical prototype - Zuhaib 80 %, Niyaz 20 %

The physical prototype of the PawPal SmartFeeder was constructed using laser-cut sheets of Medium-Density Fibreboard (MDF). While plywood was initially chosen for its durability and aesthetics, its high cost during the iterative prototyping phase led to the decision to use MDF due to its affordability. MDF provided a cost-effective solution while maintaining the structural integrity required for the prototype. However, the laser-cutting process for MDF generated a significant amount of soot. To address this, all cut surfaces were meticulously cleaned, and the edges were treated with varnish. This varnishing process not only locked the burnt edges in place but also prevented staining of hands and clothes during assembly and usage.

a) Marking and Detailing: To enhance usability and aesthetics, the product's name and the food and water sections were engraved directly onto the MDF using the laser cutter. These engravings served as clear identifiers, ensuring that the prototype was intuitive and user-friendly while maintaining a polished and professional appearance.

b) Construction and Material Use: Laser cutting was chosen for its precision and ability to create intricate designs with minimal material wastage. The cut MDF pieces were carefully assembled and glued to form the main body of the feeder. This structure housed the electronic components, including the sensors, motors, and microcontrollers. Laser cutting enabled accurate alignment and fitting, which was critical for the seamless integration of these components.

c) 3D Printing for Complex Components: While the majority of the prototype relied on laser-cut MDF, the food dispensing wheel was 3D printed due to its complex geometry, which could not be feasibly produced using a laser cutter. The 3D printing process allowed for the creation of intricate designs that ensured the functionality of the dispensing mechanism. However, the use of 3D printing was deliberately minimized due to its slower production speed and higher material costs compared to MDF. This selective use of 3D printing balanced functionality with efficiency and cost-effectiveness.

d) Final Assembly: After cutting, engraving, cleaning, and varnishing the MDF pieces, they were glued together to form a stable and durable structure. The modular design facilitated the integration of electronic components and enabled easy adjustments during testing. The use of MDF, combined with selective 3D printing, resulted in a prototype that was both practical and visually appealing, showcasing the feasibility and potential of the PawPal SmartFeeder.

This methodical approach to prototyping ensured that the design aligned with the project's goals of sustainability, cost-efficiency, and functionality while providing a robust platform for future development and testing.

Fig. 2. Top view of the food box

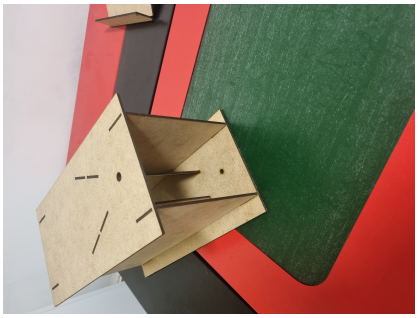


Fig. 3. Side view of the food box

Fig. 4. Semi assembled water box

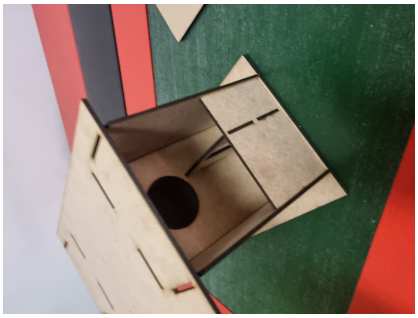


Fig. 5. fully assembled water box

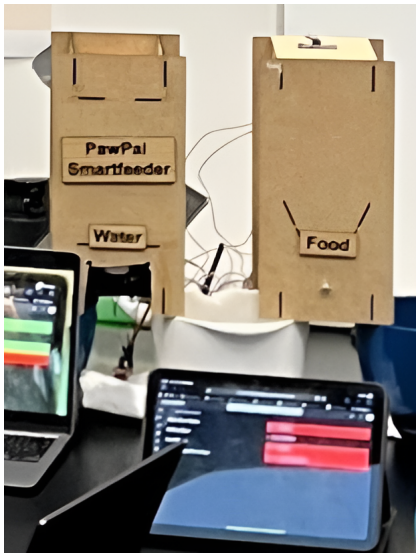


Fig. 6. Fully assembled prototype

e) 3D Printing: The 3D printed actuator for the smart pet feeder was developed to provide the required rotational movement of the food supply system. In PrusaSlicer, the model was prepped for printing by fine-tuning settings that affect printing precision, and subsequently save time. Printing was performed at DTU Skylab, where a range of fabrication equipment is available. This yields a satisfactory actuator suitable for feeder automation making it cheaper and more efficient in integration with other entire systems. The material of the actuator is also Prusament PLA which is printed from DTU Skylab in Fig. 7.

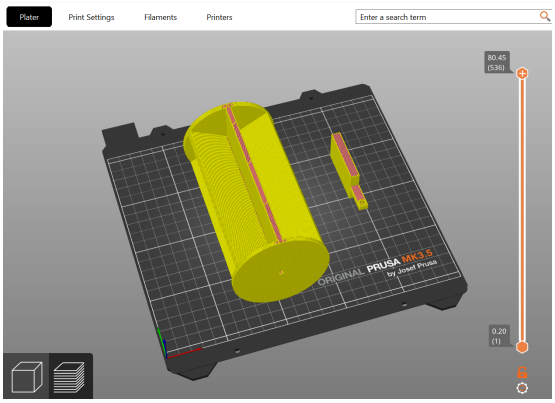


Fig. 7. 3D Printed model in Prusa Slicer

III. Code Highlights, Tomás - s242887

Two main codes have been programmed in order to make this prototype work. One code is for the Arduino Uno and the other is for the Arduino Pro Mini, as it has already been explained, the code for the Arduino Uno will mainly control the wide variety of sensors and actuators that the product has, while the code which is ran in the Pro Mini will deal with

the different communication protocols and routines that need to work simultaneously.

Both codes can be found in the Appendix, they are called `ArdUno_sensors.ino` and `ArdMini.ino` for the Arduino Uno and Pro Mini, respectively. The general workflow between both codes can be seen in Figure 8.

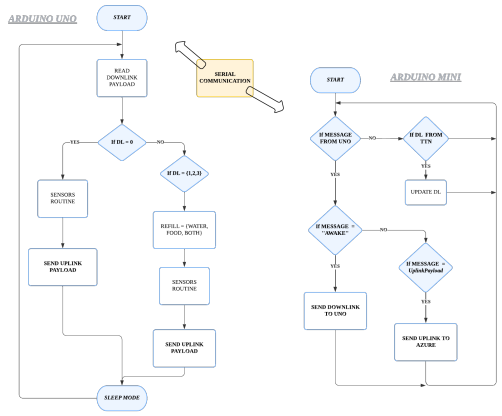


Fig. 8. Code's Flowcharts.

From these flowcharts, we can easily distinguish how some clear routines are run in parallel and are interconnected with each other through serial communication between the boards. The following is a more detailed explanation of each of the two parts of the code, paying special attention to the most important details that have made it possible to achieve the initial objectives of the prototype.

A. Arduino Uno Code

Firstly, it is salient to not the libraries that have been used among this code:

- **Wire.h**: Handles I2C communication for the LCD display.
- **LiquidCrystal_I2C.h**: Manages the I2C LCD display for output.
- **Stepper.h**: Controls a stepper motor used in the system.
- **Servo.h**: Used to control a servo motor.
- **SoftwareSerial.h**: Enables serial communication on additional pins, used for serial data transfer with the Arduino Pro Mini since the usual serial pins were already used by some sensor/actuator.

Furthermore, some of the sensors have needed special code calibration to perform the way we wanted for the project's purpose. For instance, both motors (stepper and servo) need the degrees we want them to move as input; the servo motor will just tilt itself some degrees down and up to allow the water coming down from the water tank, while the stepper motor continuously rotates on itself in 60 degree steps, as the design of the food dispenser has 6 slits that continuously rotate it ($\frac{360}{6} = 60$).

The ultrasonic sensor will radiate radar waves through its `trigPin` and `echoPin` to determine the time delay between those signals and thus it will be able to compute the distance between the origin and the object. Once this distance is known, as we have designed the food tank with a fixed height for the food, we can calculate the percentage of food remaining in it by

$$\text{Food}_{tank}[\%] = \frac{\text{height} - \text{distance}}{\text{height}} \cdot 100$$

The light sensor just need an appropriate threshold for being able to measure whether there is food or not in the food bowl. Regarding the water sensor it must be noted that, without some special considerations, the resultant measurements are not very accurate. This is mainly due to the way this sensor is designed, it has several variable resistors whose measurements vary as a function of the intrinsics water's properties, that means that the sensor might be reprogrammed every time we use water from different sources. In addition to this first consideration, this sensor must be digitally turned on and off every time we want it to measure, otherwise, if it is kept on for a long period of time, the measurements will no be accurate enough.

Once we know how the different sensors and actuators work, we can describe the general process and situations that the Arduino Uno will face according to the user's needs. Firstly, the Arduino Uno will read the downlink payload received via serial communication from Arduino Pro Mini, this downlink has four possible values $\in 0, 1, 2, 3$, and depending on the value inside this payload the board will perform 4 different possible routines, which are very similar but not the same. They are listed below:

1) **DL = 0**: It will only proceed to the sensors routine, where it will collect sensor data for water levels, food availability, and the rest of the parameters.
2) **DL = 1**: It will refill the water bowl, and after that action, it will update the setup information by performing the sensor routines (measurements).
3) **DL = 2**: This case will perform the same steps as in the previous one, with the difference that here the food tank will be refilled instead of the water one.
4) **DL = 3**: This downlink payload will be performed when the user desires to refill both bowls simultaneously.

Finally, the Arduino Uno will create an uplink message composed of 4 bytes which will contain all the information gathered through the different sensors. In addition, the fourth byte informs the user which of the actuators was the last to perform. After sending this uplink payload to the Arduino Pro Mini, the Uno board will enter into sleep mode for power consumption considerations, this sleep mode can be modified according to the user needs; however, as a normal pet eats between 3-6 times per day, this sleep mode's timeout can be set significantly high, for example, every one hour.

B. Arduino Pro Mini Code

The Arduino Pro Mini will be the main controller for keeping all the communications alive between the prototype's different subsystems. The hardest task to code was related to the two serial communications running simultaneously, between the Arduino Uno and Pro Mini and between the Arduino

Pro Mini and the LoraWAN module. Finally, to accomplish a solid performance, a technique had to be implemented that allowed one of the communications to be switched on while the one that was not needed at the time was switched off. For instance, if the Arduino Pro mini wanted to read the uplink payload from the Arduino Uno, the serial communication with the LoraWAN modulo must be off, and vice versa.

Once this consideration is fulfilled, the Arduino Pro Mini routine was constantly hearing for new and updated information coming from two possible paths, the Arduino Pro Mini (uplink message) or the LoraWAN/TTN Network (downlink message). Depending on the message received, the Arduino Pro Mini will perform the following actions:

- If the message is sent by the Arduino Uno, it will automatically check whether the message is equal to `Awake` or not. If it is an awake message, that would mean that the Arduino Uno has just started running its routines and therefore it is looking for the Downlink data. Nevertheless, if this message received from Uno is not awake, the Arduino Pro Mini has received the Uplink message that has to be uploaded to the LoraWAN network so the user can visualize this data via Azure.
- The second option is that the message received by the Pro Mini comes from the LoraWAN/TTN network. If this is the case, the Arduino Pro Mini would just have to update the Downlink variable so that the message is ready in case this information has to be sent to the Arduino Uno and it is the Arduino Uno that executes the actions desired by the user.

IV. Communication and Data Handling System-Niyaz-s240041

A. Software Serial

We established communication between an Arduino Uno and Pro Mini through the SoftwareSerial Library to transmit data. We define pins serving as RX and TX for 10 and 11, which was useful in order to reserve the primary hardware serial connection for debugging only. I made sure that both boards had same baud rate for serial communication, while concerning voltage compatibility as Uno works at 5V originally and Pro Mini can work at both 5V and 3.3V. As for data management, I introduced a low-level signaling technique, consisting of start/stop signals, in order to properly convey data through the connection.

B. LoRaWAN Communication

The LoRaWAN communication is achieved by the module RN2483. Before the LoRa communication, we need to initialize the LoRa antenna with a few settings, which depend on the communication scenario we are facing. We set the frequency as 868GHz. Since there might be other users using the same frequency as ours, there would be some other practical problem occurs like spectrum occupancy or collision of the packets from/to other users. We can reduce the occurrence of this situation by sending some error intervals every 30s, so that it can release the frequency which representing the spreading factor from sf7 to sf12. The spreading factor is an important parameter in LoRa modulation, used to adjust the performance and range of data transmission. In LoRa modulation, the spreading factor determines the multiplication factor by which the original data bits are spread across more signal symbols. A higher spreading factor means more signal symbols are used to transmit the same number of data bits, resulting in a lower data transmission rate but increased signal robustness and transmission range. In practical applications, selecting the appropriate spreading factor based on specific communication needs and environmental conditions can optimize the performance and power consumption of LoRa systems.

To control our smart pet feeder sensor I implemented the up and downlink communication using LoRaWAN through The Things Network (TTN). Some sensor on the smart feeder had to monitor the parameters such as food levels, water levels and the dispensing status. The feeder's status was then monitored remotely in near real time through uplink communication – these readings were transmitted to the TTN cloud. An overview of network architecture is given below in Fig. 9.

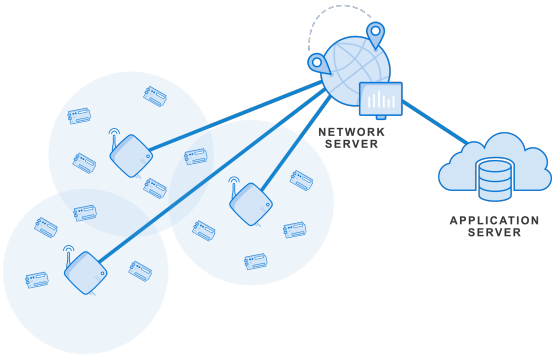


Fig. 9. Overview of Network Architecture

Source:https://www.thethingsnetwork.org/docs/network/
The overview of network architecture describes how information is passed from Arduino to a LoRaWAN gateway, a LoRaWAN network server, and ThingSpeak. Data from for instance, a sensor (uplink data) is transmitted through the Arduino to the LoRaWAN gateway and then to the LoRaWAN network server. ThingSpeak is the fourth tier through which the server comes into interaction for data visualization and analytics. Thus in the downlink, actions or replies by ThingSpeak are transmitted through the network server as well as the gateway to the Arduino thus creating a two-way communication. This setup proves perfect for IoT applications that entail distant information supervision and manipulation. The pet feeder configuration in TTN was to have messages downlink scheduled to the feeder. Commanding, for example, which food to dispense how much or otherwise adjusting the feeding time according to predefined routines or on the fly ad-hoc needs, were the types of downlink messages that were delivered to these uplink messages which is recieved in azure IoT and user can see update in real time in our UI about the feeder system as the his/her pet ate the food

or not, even the pet ate more than usual he can control the food supply system through downlink. Our Azure setup overview is being attached here for visualization in Fig. 10. The user can also see how much food is in the container as he/she need to buy more food for the pet or not and it will give the user update everyday. Overall, the user can make a dietary routine for his/her pet. To accommodate downlink

Fig. 10. User Interface of Azure IoT

packets controlled by duty cycle regulations and timing requirements from LoRaWAN downlinks, precise timings were set for the downlink packets during the scheduling process. That added expense, however, helped the system stay efficient and compliant with LoRaWAN standards. Since the device was intended to operate in low power mode most of the time, a key challenge was to have the uplink and downlink communication cycles be synchronized. I also used techniques such as configuring the device to start listening for downlink messages as soon as you send an uplink, or take advantage of the RX1 or TX2 receive times of the LoRaWAN protocol. I also added error handling so that commands were sent that were received by the feeder and carried out. Thus, LoRaWAN is firmly positioned to solve exactly those remote control use cases requiring long-range as well as low power consumption as depicted in this project. Data routing and scheduling flexibility coupled with scalability were provided by integration of TTN, enabling the smart pet feeder to be easily managed remotely. If the system receives a downlink message of 01 it will dispense Food, 02 will dispense water and 03 will dispense both which can be seen from Fig. 11.

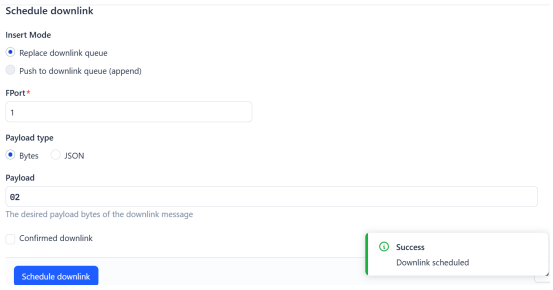


Fig. 11. Schedule Downlink

There are some challenges we faced for two different Lora module. The continuous connection to TTN failed but it can do both uplink and downlink. On the other hand it can send uplink continuously with no disturbance but downlink is not possible with it. This happens due to duty cycle limits, or firmware mismatches.

V. Market analysis - Zuhaib 60 %, Aysha 40 %

The global smart pet feeder market has been growing at an unprecedented pace, fueled by advancements in technology, shifting demographics, and evolving consumer expectations. In 2023, the market was valued at $1.66 billion and is projected to grow at a compound annual growth rate (CAGR) of 25.1%, reaching $12.51 billion by 2032 [1]. This growth reflects a broader trend of integrating smart technologies into pet care as part of the larger Internet of Things (IoT) ecosystem. The rising adoption of pets as family members, known as pet humanization, is a significant factor driving this market expansion, alongside increasing disposable incomes and the growing need for convenience in pet management.

Evolution of Pet Care Technologies

Traditionally, pet care products focused on manual operation, requiring significant owner intervention. Feeding, cleaning, and health monitoring were time-intensive tasks, often prone to human error. The introduction of automatic feeders marked a significant evolution, enabling users to schedule feeding times and ensure consistent food portions. However, these early solutions were largely mechanical, offering limited customization or adaptability to individual pet needs.

The advent of smart technologies, particularly IoT-enabled devices, revolutionized the market. Modern feeders incorporate connectivity, data analytics, and automation to provide enhanced control, convenience, and precision. Features like mobile app integration, remote control, and voice commands have become standard in many products. Yet, significant gaps remain in addressing broader challenges, including inclusivity, connectivity in underserved areas, and advanced health monitoring. PawPal aims to bridge these gaps by integrating innovative features like LoRaWAN connectivity, enhanced energy efficiency, and inclusive design.

Addressing Connectivity Challenges

Most existing smart feeders rely on Wi-Fi or Bluetooth for connectivity. While effective in urban and suburban areas with stable internet access, these methods falter in rural or remote locations where connectivity is unreliable. This limitation excludes a considerable segment of potential users, including pet owners in less connected regions and large-scale operations like boarding facilities or shelters.

PawPal differentiates itself by incorporating LoRaWAN technology, a long-range, low-power wireless communication protocol. Unlike Wi-Fi and Bluetooth, LoRaWAN operates over larger distances and requires minimal power, making it ideal for environments with limited or intermittent internet access [2]. This feature not only ensures seamless operation in remote areas but also supports offline functionality, a critical requirement for regions where internet infrastructure is underdeveloped.

Sustainability and Energy Efficiency

As sustainability becomes a priority for consumers, energy efficiency is increasingly viewed as a key feature in smart devices. Many current feeders offer basic power-saving modes, which marginally reduce energy consumption. However, PawPal introduces an advanced sleep mode that significantly

lowers power usage without compromising functionality. This not only reduces operational costs for users but also aligns with global sustainability goals and eco-conscious consumer behavior [3].

The environmental impact of IoT devices has been a growing concern, with research highlighting the need for energy-efficient designs to minimize carbon footprints. PawPal's design reflects this ethos, ensuring that its advanced features, such as health tracking and connectivity, are delivered without excessive energy demands.

Advanced Health Monitoring

Health and wellness are increasingly important to pet owners, particularly as pets age or develop chronic conditions requiring closer observation. While most existing smart feeders automate feeding schedules, they often fail to provide insights into a pet's eating behavior—a critical indicator of health. Sudden changes in appetite, overeating, or underfeeding can signal underlying health issues.

PawPal addresses this gap by integrating sensors and algorithms to monitor and analyze eating patterns. These systems detect irregularities such as decreased consumption or excessive feeding, triggering alerts for the owner. By combining these insights with Azure IoT Hub's data analytics capabilities, PawPal not only provides real-time monitoring but also generates detailed reports to aid in veterinary consultations [3]. This proactive approach to pet care ensures early intervention and supports better long-term health outcomes for pets.

Inclusivity in Design

Inclusivity is a cornerstone of PawPal's design philosophy, recognizing the diverse needs of its user base. Existing smart feeders often cater to young, tech-savvy urban consumers, neglecting other significant demographics. PawPal is designed to address the needs of:

- **Visually Impaired Pet Owners:** According to the World Health Organization, over 2.2 billion people globally live with vision impairments, many of whom rely on service animals for companionship and assistance [4]. PawPal incorporates tactile buttons, voice commands, and audible alarms, empowering visually impaired users to manage their pets' feeding independently.
- **Busy Professionals:** Urbanization and the prevalence of dual-income households have created a demand for time-efficient solutions. Automated feeding schedules, remote management capabilities, and real-time monitoring make PawPal ideal for busy professionals. With over 60% of urban households being dual-income, these features address a critical market need [5].
- **Pet Boarding Services:** The growing reliance on boarding facilities during vacations and work trips has created a need for scalable solutions. PawPal's centralized control system enables boarding facilities to manage multiple feeders efficiently, ensuring precision and ease of operation [6].

Global Trends and Market Opportunities

The global pet care industry is characterized by two dominant trends: pet humanization and IoT adoption. In the United States alone, 67% of households now own at least one pet, reflecting a growing cultural shift toward treating pets as family members [5]. This trend has led to increased spending on premium pet care products and services, creating significant opportunities for innovative solutions like PawPal.

The IoT pet care market, valued at $5.14 billion in 2020, is projected to reach $10.92 billion by 2027, growing at a CAGR of 11.6% [6]. PawPal's integration of IoT technologies, including Azure IoT Hub and LoRaWAN, positions it as a leading solution in this expanding market. These technologies enable seamless connectivity, real-time monitoring, and advanced analytics, ensuring PawPal meets the evolving expectations of a tech-savvy consumer base.

Competitive Positioning

PawPal's unique combination of advanced automation, energy efficiency, health monitoring, and inclusivity sets it apart from competitors. By addressing underserved segments and aligning with global trends, PawPal not only meets market demands but exceeds expectations. Its innovative design and user-centric approach ensure it is well-positioned to lead the smart pet feeder market in the coming decade.

VI. Challenges, Comments, and Conclusion, Jorge - s243218

A. Challenges

Developing the PawPal SmartFeeder presented several technical and operational challenges that required innovative solutions and iterative testing to address effectively.

a) Communication Between the Microcontrollers:: Establishing reliable communication between the Arduino Uno and Arduino Pro Mini was one of the most complex aspects of the project. While the `SoftwareSerial` library enabled serial communication over non-standard pins, managing two simultaneous communication channels posed difficulties. The need to toggle communication lines between the LoRaWAN module and the Uno introduced latency and complexity, necessitating careful timing and error handling. Additionally, ensuring voltage compatibility between the 5V Uno and the Pro Mini, which operates at either 3.3V or 5V, required precise adjustments to maintain signal integrity.

b) Setting Up Downlink Communication with LoRaWAN:: Implementing downlink communication using LoRaWAN via The Things Network (TTN) posed another significant challenge. While uplink data transmission was stable, achieving synchronized downlink packets required precise configuration.

c) Calibrating Sensors and Actuators:: The sensors and actuators demanded extensive calibration to ensure accuracy and consistency. The ultrasonic sensor, used for measuring food levels, required adjustments to eliminate noise caused by environmental factors. Similarly, the water level sensor's sensitivity to water source variations necessitated recalibration

for each testing setup. For the actuators, achieving precise control over the stepper motor and servo motor involved meticulous tuning of parameters like rotation angles and activation thresholds.

B. Comments

The PawPal SmartFeeder demonstrates the potential of IoT technology to transform pet care through automation, connectivity, and advanced monitoring capabilities. However, the challenges encountered emphasize the importance of system reliability, particularly in communication protocols and sensor accuracy. Iterative prototyping and testing were crucial to overcoming these challenges, highlighting the value of modular design and scalability. The inclusion of LoRaWAN proved to be a differentiating feature, enabling broader applicability in areas with limited connectivity.

From a design perspective, the decision to use laser-cut MDF and selective 3D printing balanced cost-efficiency and functionality, but the material constraints of MDF, such as fragility and soot generation during cutting, highlighted the need for alternative materials in future iterations. Additionally, the inclusivity features, such as tactile buttons and audible alarms, could be further refined to meet the needs of diverse user groups.

C. Conclusion

The PawPal SmartFeeder successfully addresses key gaps in the smart pet feeder market, offering an innovative solution that combines automation, energy efficiency, and advanced connectivity. By leveraging IoT technology, the system provides pet owners with real-time monitoring and control, enabling better management of their pets' health and feeding routines. Despite the challenges encountered, the project achieved its core objectives, demonstrating the feasibility of integrating modular electronics, LoRaWAN communication, and Azure IoT Hub into a cohesive system.

Moving forward, future iterations could focus on enhancing sensor robustness, improving downlink reliability, and exploring alternative materials for the physical prototype. The addition of machine learning algorithms for predictive feeding patterns and integration with wearable pet health trackers could further expand the system's functionality. As the smart pet care market continues to grow, the PawPal SmartFeeder is well-positioned to lead the industry with its innovative, sustainable, and inclusive design.

VII. Appendix

The market analysis slides and the code used in the project has been submitted in a zip file. A series of test videos demonstrating how the project works can be viewed from this link: IoT Project Media.

A. Note about contribution percentages

Note: The contributions in report are written in the heading with respesctive percentages. However for the actual tasks the percentage contributions are mentioned in the table below.

TABLE I
Project Responsibilities Distribution

Responsibility	**s240041**	**s223296**	**s175037**	**s242887**	**s243218**	**s232194**
Physical Prototype	10%	80%	10%			
Circuit Design				20%	80%	
Code				80%	20%	
Communication	80%			10%		10%
CAD Modelling			100%			
Market Analysis		30%				70%

References

[1] Business Research Insights, "Automatic and smart pet feeder market size, share, and forecast 2023-2032." Retrieved from *https://www.businessresearchinsights.com/market-reports/automatic-and-smart-pet-feeder-market-107575*.

[2] TagoIO, "Benefits and applications of LoRaWAN technology." Retrieved from *https://tago.io/blog/lorawan-benefits-applications*.

[3] Fact.MR, "Automatic pet feeder market insights and trends." Retrieved from *https://www.factmr.com/report/automatic-pet-feeder-market*.

[4] World Health Organization, "World report on vision," 2021. Retrieved from *https://www.who.int/publications-detail/world-report-on-vision*.

[5] Statista, "Global pet care market trends." Retrieved from *https://www.statista.com/*.

[6] Allied Market Research, "IoT in pet care market analysis and trends." Retrieved from *https://www.alliedmarketresearch.com/iot-in-pet-care-market*.

[7] Thingiverse "Micro-Servo (SG90) Attachment gear" Retrieved from *https://www.thingiverse.com/thing:3627706*