Scalio: An Automated Weighing Solution for Orthonological Research

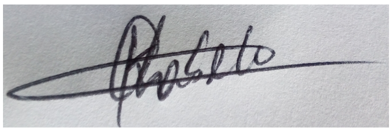
Prepared by:
Revashan Soobiah
Bonga Njamela
Kananelo Chabeli
Ankush Chohan

Prepared for:
EEE4113F
Department of Electrical Engineering
University of Cape Town

May 11, 2024

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

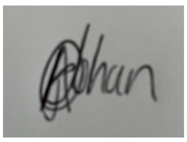


May 11, 2024

Kananelo Chabeli — Date

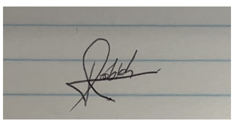
May 11, 2024

Bonga Njamela — Date



May 11, 2024

Ankush Chohan — Date



May 11, 2024

Revashan Soobiah — Date

Contents

Chapter 1

Introduction

1.1 Background

This project focuses of research being done on avian species in the Kalahari by Ornithologists working under the Fitzpatrick institute. One of these Ornithologists, Ben Murphy, studies the behaviour and characteristics of Fork-Tailed Drongos for conservation purposes. One of Ben's objectives is to understand why this species in particular is surviving well, while others are dying due to climate change. He uses weight measurement techniques to do this. His current data gathering procedure involves placing a scale with a makeshift perch attached to it on the ground, and then observing the scale's output through binoculars while standing some distance away.Ben's current solution for collecting weight data, therefore, proves to be very inefficient, inaccurate, and labour intensive, forcing him to be out in the harsh conditions of the Kalahari continuously to record data for his invaluable research. This project details a prototype to help Ben conduct his research more effectively. The project is therefore based on the following problem statement.

1.2 Problem Statement

Accurate weight data is an essential piece of the puzzle when studying bird species for conservation, and therefore, Ben Murphy who is a PHD Student that is doing research in the Kruger needs a way to record weight data with ID of Fork-Tailed Drongos to understand the species better.

1.3 Scope & Limitations

The scope of this project is as follows:

- The goal of this project is to automate the process in weighing Fork-tailed Drongos.
- The avian species are being researched in the Kalahari having harsh outdoor conditions, whilst being remote.

The limitations of this project is as follows:

- The harsh conditions of the Kalahari pose a risk of components repeatedly overheating and failing.
- The final prototype must be completed within 6 weeks.

- Cost constraint heavily influenced the decision component quality.
- The design will take place in Cape Town, where there is no access to the conditions in the Kalahari for testing purposes.

1.4 Report Outline

The report is structured as follows:

Chapter 1 provides an overview of the project, discussing the identified problem and project constraints.

Chapter 2 conducts a comprehensive literature review on sensor technologies and traditional methods in wildlife monitoring. It explores techniques such as camera traps, PIT tags, radar, and thermal detection, emphasizing the importance of bird weight measurement. Additionally, it discusses data storage, transmission solutions, traditional data processing, and the role of machine learning in wildlife data analysis.

Chapter 3 details the design of the scale and housing for weighing Fork-Tailed Drongos, explaining design choices for load cell circuitry and housing components. It discusses testing procedures, results, and acceptance tests, highlighting areas for further refinement.

Chapter 4 focuses on the design of the data acquisition and retrieval subsystem, discussing requirements, design choices, and testing procedures related to RFID detection, ADC calibration, Kalman filtering, and wireless data transmission.

Chapter 5 explores the design of the power subsystem, addressing battery selection, solar energy utilization, power regulation, and protection mechanisms. It outlines an acceptance test procedure for validating the power module's functionality and reliability.

Chapter 6 addresses the design of the user interface and wireless communication system, detailing user requirements, design choices, and acceptance test results for high fidelity prototypes, codebase specifications, and feature unit tests.

Chapter 7 concludes the report with findings and recommendations for future work.

Chapter 2

Literature Review

2.1 Introduction

The development of electronic devices that can effectively operate in the wild is an important consideration when dealing with the problem at hand which is that Ben Murphy who is a PHD Student that does research in the Kalahari needs a way to record weight data, with ID of Fork-Tailed Drongos to understand the species better. To achieve such a task, electronic devices will be deployed into the Kruger to gather data on the Fork-tailed Drongos. The systems that Ben uses need to operate accurately regardless of the environmental conditions. This literature review aims to examine existing technologies that have been used in the field of wildlife conservation, specifically focusing on their application in wildlife monitoring, with a particular emphasis on recording weight data and identifying Fork-tailed Drongos.

2.2 Sensor technologies for wildlife monitoring

The following section addresses existing wildlife monitoring sensor technologies that have been/ are currently being used in the field.

2.2.1 Camera Traps

The utilization of electronic devices in challenging environmental conditions, particularly in the domain of wildlife monitoring, is an area of growing importance as the need for efficient, and effective conservation techniques increases with the decline in environmental welfare, due to global warming. This is exemplified in a study conducted on Barro Colorado Island in Panama that employs motion-sensitive camera traps to monitor the movement and distribution of terrestrial bird and mammal populations [1]. The study underscores the significance of employing sensor technologies in addressing environmental challenges such as invasive species, infectious diseases, and climate change. Camera traps offer a 'non-invasive' and efficient means of recording animal activity by, 'capturing photographs with invisible infrared flashes' [1] to minimize disturbance to wildlife. The article highlights the advantages of camera traps, including their ease of deployment, robust data collection capabilities, and the potential to record not only species presence but also behavior. Moreover, the study emphasizes the importance of randomizing camera deployment locations and avoiding the use of baits to reduce bias and ensure accurate data collection. By integrating sensor networks with wildlife monitoring efforts, researchers can gain valuable insights into species diversity, abundance, and ecological interactions, ultimately contributing to informed conservation decisions and management strategies. This study [1] serves as a compelling example of the potential of electronic devices and sensor technologies to advance our understanding of wildlife dynamics in harsh environments.

2.2.2 Passive Intergrated Transponders (PIT)

A method for identifying an individual bird when gathering data on them (i.e to determine if the bird named Bonga, or Rev is on the scale at a particular point in time) was found to be via the use of passive integrated transponders (PIT) tags. The tags are 'either implanted or attached to a leg ring. Birds then can be identified automatically each time they approach an antenna at a feeder, a balance, or a nest box' [2]. They do not require a power source, they are small and very light [3]. This shows that PIT tags are an effective method of identification for relatively close proximity uses. The tags work off RFID technology that transmits a 10-character code when near a scanner [3]. This opens up the potential for 'automated recognition of birds' [2].

2.2.3 Other Tracking Methods

The utilization of sensor technologies for wildlife monitoring, particularly focusing on birds, has significantly advanced our understanding of avian behavior and migration patterns. Researchers have been able to attach lightweight tracking devices to birds, allowing for the study of their movement in unprecedented detail [2]. Biotelemetry techniques enable the tracking of individual birds during migration flights [2], providing insights into flight duration, route selection, and energy expenditure. Satellite tracking has further expanded the scope of monitoring, allowing for the tracking of birds over vast distances, revealing remarkable long-distance migrations such as the record-breaking flight of an Eastern Bar-tailed Godwit across the Pacific Ocean [2]. Geolocator loggers have also contributed to tracking migrating songbirds by recording data on dawn and dusk times to calculate geographical coordinates. These technologies, while facing limitations such as device size and retrieval challenges, continue to evolve, promising even greater insights into avian ecology and migration behavior[2].

2.2.4 Complete System Of Weight and Identification

Boisvert and Sherry [3] present a comprehensive system for the automated recording of feeding behavior and body weight in avian species, focusing particularly on black-capped chickadees. Utilizing passive integrated transponder (PIT) tags, the authors describe a method that uniquely identifies individual birds and continuously records their visits to a food source along with their body weight [3].

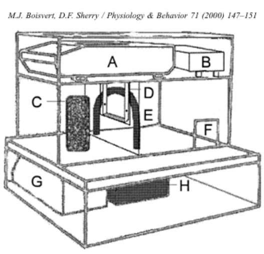


Figure 2.1: Boisvert and Sherry's Avian Monitoring System [3]
A-Balance, B-Interface, C-Food Source, D-Photobeam, E-Scanner Loop,F-Entrance G-Reader, H-Scanner

This system, comprising a PIT tag scanner, electronic balance, and perch with an infrared photobeam, and a food source enclosed in a waterproof housing, operates autonomously, providing detailed data on feeding behavior and body weight dynamics. Notably, the accuracy of body weight measurements is highlighted, with birds remaining on the perch while taking seeds, ensuring precise weight recordings. By overcoming the limitations of traditional repeated capture techniques, this innovative approach offers a non-invasive and continuous monitoring solution for studying avian feeding behavior and body weight regulation [3]. The system is shown in figure 2.1 above.

2.3 Traditional Techniques for Bird Detection and Recognition

The ability of birds to change spatial location at a rapid rate poses challenges that make it difficult to detect, image, and associate biometric data with a specific bird. Additionally, body size and plumage can affect a viewer's ability to recognize a bird from a distance, or in low-light conditions. The following section assimilates the traditional methods for detecting, imaging and acquiring avian biometric data. Furthermore, the following discusses adaptations of traditional methods in developing avian biometric collection, and identification, as well as locating wild birds as they fly around their natural habitat. Specifically, techniques that rely on the naked-eye and naked-ear, including the use of colour to mark birds, binoculars, as well as standard sound, and image-capturing devices, were considered to be traditional methods. Advancements in these methods that are discussed include radar imaging, which uses reflected radiowaves to construct an image of a large region, as well as thermal imaging, which uses infrared to produce surface temperature mappings of objects in the surroundings. Typically, systems offer singular modalities of bird detection and bird data collection that do not offer the advantages of other modalities. For example, devices that measure weight data typically do not offer imaging capabilities. This chapter reviews methods that combine modalities of detection, imaging, as well as bird recognition.

2.3.1 Using Dye to Mark, Detect and Identify Birds

Bird detection and recognition allow ornothological researchers to associate biometrics, such as weight data, with a specific bird. This is significantly difficult to achieve when the bird has a small body size and dark plumage. Patterson elaborates that the fact that colour plays a significant roll in the how birds interact with one another and the environment, it is not always suitable to use coloured dye markers to identify the birds [4]. Careful consideration has to be taken in selecting dye as the wrong chemical composition can deter birds from interacting with a marked bird [4]. Crucially, dyes for marking birds only last until the molt, i.e. when a bird typically loses plumage feathers. The chemical composition of the dye also has an effect on the health of the bird's plumage and possibly, the overall health of the bird. A suitable dye that was implemented by Ellis and Ellis [5], uses *picric* acid, which is a pale yellow substance typically found in low-cost human hair dyes [4]. In their experiment, molted Golden Eagle feathers were dotted with mixtures of commercial human hair dyes to produce dark and light colours [5]. The durability of the dye was measured by leaving the marked feathers unprotected from the weather and sunlight for two months[5]. The best durability was achieved for dark colors, which were prominent at the end of the two month weathering period [5].

After conducting a durability test, white Golden Eagle nestlings were spotted with a black aniline dye [5]. Remarkably, the dye was still noticeable and the bird was identifiable after of 21 months from fledging, showing that the dye was an effective method for bird recognition after 1 annual molt [5]. Therefore, one of the advantages of this traditional method used in detecting and recognizing birds using the naked-eye is that the dye can last for a sufficient amount of time to analyse changes in the weight of a bird. Another advantage, is that dye is easy to apply and remove, where Ellis and Ellis state that excess dyeing solution was removed using a sharp tool to scratch it off and water to rinse the feathers [5]. Finally, the dye-marking technique offered the advantage of causing minimal to no disturbance in the behaviour of the birds [5], which is critical for ensuring their survival.

2.3.2 Augmenting Capabilities of the Naked-eye for Bird Detection and Recognition

Although the dye technique offers low cost, durability, easy application, as well as being non-disruptive to bird behaviour, a critical shortcoming becomes apparent when the bird is not in range of visibility, i.e. when the bird is too far from the human to distinguish the markings on its feathers. Typically, ornothological researchers and wildlife conversationists work in sparse natural habitats. This increases the chances of having to retrieve information from a long distance from the target. A traditional method for augmenting the capabilities of the human naked-eye employs binoculars to bring birds that are out of the range of visibility into a closer and more discernible field of view. Glowinski [6] posits that the affordability of binoculars has made *birdwatching* (or *birding*) - which is defined as 'the active observation, identification, and/or photography of birds for recreational purposes' - more accessible to the general public [6]. However, due to the need for human cross-validation, binoculars are typically used by researchers for visual surveys. For instance, Duberstein et al. [7] performed a study of automated image processing for detection and classification of bird species that inhabit nearshore marine environments in which the primary data acquisition method was video capture [7]. Although the primary method was video capture, an observer was required to perform concurrent visualizing using image-stabilized binoculars. This is indicative of the importance of binoculars in the conservation of birds in that they remain a fundamental device in ensuring that the correct data is retrieved from a bird-sighting. However, traditional binoculars do not offer good visibility in low-light conditions and the ones that do are typically expensive and apply non-traditional techniques for augmenting human vision.

2.3.3 Traditional Methods for Detecting and Identifying Birds in Low-light Conditions

To overcome the challenge of detecting and identifying birds in low-light conditions, a different form of energy transfer may be required, other than the transfer of electromagnetic energy in the form of visible light that occurs when a bird is observed using dye and binoculars. Ornithologists, and indeed the lay observer, typically use the musical call of a bird, called a `song`, to detect the birds using the naked-ear or a sound capturing device. In this case, the mechanical energy is transferred as the sound from the bird propagates as longitudinal waves through the air, with frequencies that are unique to that species of bird. Since this uses the propagation of sound through air, the brightness of the surroundings as well as the colour of the plumage do not have an effect on the detection and identification of a bird.

Although sound detection offers a solution for detecting birds in low-light conditions, errors due to noise in the environment can lead to false detections of birds. Another significant shortcoming of using sound detection is that birds, such as the fork-tailed Drongo, mimic the calls of other bird species [8]. A study by Samira et al. (2014) on quantifying vocal mimicry in the greater racket-tailed drongo suggests that human cross-validation is still neccesary despite advancements in technology [9]. Furthermore, Samira et al. suggests that automated and human cross-validation techniques for vocal mimicry analysis have similar accuracy. This indicates that sound detection is not a suitable for implementation as the only technique for detecting and recognizing birds.

2.3.4 Application of Cameras in Traditional Bird Detection and Identification Techniques

A traditional method for overcoming many of the shortcomings of sound detection and relying on the naked-eye, involves employing cameras to detect and identify birds, as introduced in in section 2.2.1. This can be extended to annotated camera videography, as well as other uses of permanent cameras. For example, Vertraeten et al. (2010) suggest that webcams are low-cost and can be easily enhanced to withstand extreme weather conditions. The experiment investigated the performance limits of a webcam's detection and tracking capabilities in relation to a bird's velocity, contrast and size using a pendulum to model a flying object [10]. Stereo image recording was used, which requires the set up of two webcams that are positioned on a level plane and point in the same direction as shown in figure 2.2. Using this apparatus, the pendulum experiment by Vertraetren et al. resulted in larger objects, of about 1.6 cm, being more visible at higher speeds than objects that were smaller, about 0.6 cm in size [10]. The study also found that objects with a darker contrast (black) displayed longer visibility than objects with a lighter contrast (white)[10]. The study highlights the importance of selecting the

Figure 2.2: Showing a stereo image recording setup with two webcams mounted on a level plane and pointing in the same direction. [10]

correct algorithm for processing camera recordings. These algorithms are required to compensate for the issues that arise from high speed motion detection and the fact that webcams have low cost plastic lenses which can give an obscured representation of reality. Using motion detection by background subtraction, stereo vision and lens distortion, Vertraeten et al. (2010) demonstrated that webcams can sufficiently indicate a moving object's minimum size, velocity and required contrast - provided that a suitable modular processing scheme is selected. The following section investigates adaptations of traditional methods that use different frequencies of electromagnetic waves and rely on computational interpretations, only using human naked-eyes and ears to validate computational results.

2.4 Advanced Techniques for Avian Detection and Recognition

In recent years, radar and infrared sensing has seen a rise in popularity due to the ability to overcome the shortcomings of the traditional methods. For example, infrared detection works well in low-light conditions and is therefore useful in ornithological analyses of nocturnal birds. The following subsections investigate the implementation of advanced techniques that have been implemented in bird detection and imaging.

2.4.1 Radar and Thermal Detection and Imaging

Methods for detecting and identifying birds for biometrics extraction such as stereo imaging can lead to errors if multiple birds moving at the same velocity, altitude and also have the same size in the visibility field of the camera. Sidney et al. proposed a more accurate technique to overcome this potential difficulty where an upward pointing static radar beam, together with an upward pointing thermal imaging camera were used to accurately tally birds moving passed a stationary circular sampling region and to accurately measure the altitude of the bird [11]. This technique was applied in the study of migratory birds by analyzing video footage recorded between the year 2000 and 2003. In a more recent study by Horton et al. [12], a combination of weather surveillance radar, thermal infrared cameras and an acoustic recorder were implemented to quantify the patterns of nocturnal traffic estimates of birds and bats between 2011 and 2012. Horton et al. [12] suggests that deploying 'weather surveillance radar in ornithological research provides unique means of monitoring avian movements' because radar is suitable for flying animals located in a sparse environment [12]. Sidney et al. suggest that an application of these three modalities, i.e. radar and thermal imaging, as well as acoustic detection, is sufficient for providing satisfactory estimates of bird biometrics such as density, mass and size [13]. In their study, Horton et al. collected data during morning and evening twilight, and downloaded square kilometer resolution reflectivity data from the National Mosaic [12]. The downloaded data was used to determine the average reflectivity of radiowaves across all the radar scans, so as to measure the relative density of an animal through the night [12]. Together with thermal imaging and acoustic detectors, the experiment was able to detect 16573 animals during spring [12].

2.5 Measuring Weight of Birds

As previously highlighted by Ben, the assessment of weight for fork-tailed Drongos has emerged as a pivotal factor in understanding why this avian species is not affected by extinction factors such as global warming. Weight serves as a key indicator for researchers to evaluate a bird's physiological condition, with fluctuations in weight often reflecting variations in energy expenditure associated with activities such as foraging [14]. Notably, the impact of global warming has been observed to correlate with a decline in body weight among passerine birds[15], a weight category encompassing the Fork-tailed Drongo.

2.5.1 Techniques Used to Weigh Small Animals

Various methodologies have been employed to ascertain the weights of small animals, each tailored to suit specific research requirements and environmental constraints.

Traditional approaches to weighing small animals, commonly entail the use of scales. Specially designed scales, compact and portable, offer convenience and usability in challenging environmental settings. These scales utilize gravity and the object's mass to determine its weight [16].

Zoologists have devised strategies to weigh small birds without inducing stress, thereby enhancing measurement accuracy. As noted by the Smithsonian National Zoo, enticing birds onto scales using a baited log, typically with live worms, proves effective, as birds naturally investigate the bait, affording researchers sufficient time for measurements [17]. However, it's important to note that this method is applicable primarily within controlled zoo environments and not wildlife habitat.

Alan Poole and Jon Shoukima observed challenges associated with obtaining repeated weight measurements of wild adult birds due to factors such as birds' transient behavior and environmental variability [14]. Their innovative approach, depicted in Figure 2.3, involved utilizing perch-like poles to simulate natural perching spots, minimizing disturbance to birds while enabling weight measurements.

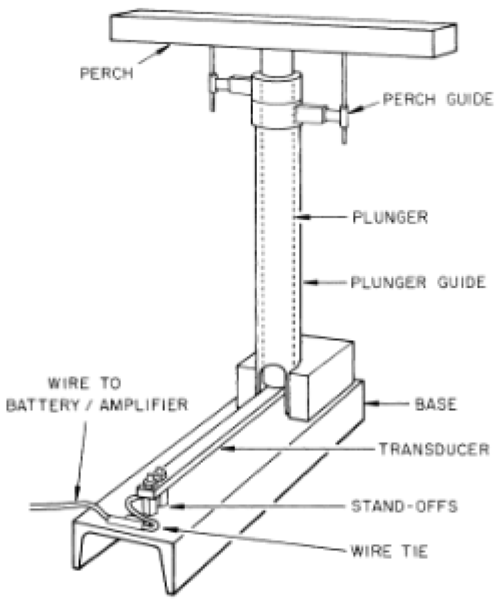


Figure 2.3: Showing a bird scale used in wildlife research taken from source[14]

2.5.2 Effect of Unpredictable Bird Movement on Accuracy Readings

The unpredictable movements of birds pose significant challenges when attempting to weigh them, particularly in field research settings. Birds' agility and erratic behavior make securing them for precise measurements challenging.

The operation of scales involves transducer beams detecting changes in electrical resistance upon object placement, with readings relayed to controllers via analog-to-digital converters [18]. The prolonged measurement process of scales can be influenced by birds' unpredictable movements. Nevertheless, innovative methodologies have been devised to accommodate and mitigate such challenges, as discussed in the previous subsection.

2.5.3 Challenges and Limitations

Power Limitations

In the realm of wildlife research, employing technology for data collection and storage presents notable challenges, particularly concerning power constraints in harsh environments [19]. Electrical components exhibit temperature thresholds and susceptibility to damage under excessive force. The temperature thresholds has been accounted for by using heat sinks,etc[20].

However, the primary challenge arises from the need for prolonged device operation, often spanning hours or days, in remote locations devoid of electricity. While batteries offer temporary solutions, harsh environmental conditions, including sunlight exposure, can compromise their efficacy. As a remedy, scales must prioritize reduced power consumption, often necessitating alternative power sources like solar panels [21].

2.6 Conclusion

In conclusion, this literature review highlights the critical role of advanced sensor technologies, traditional methods, and machine learning approaches in wildlife monitoring and conservation. Camera traps and passive integrated transponders (PIT) tags offer non-invasive monitoring and automated recognition, while tracking technologies provide valuable insights into avian behavior and migration patterns. Traditional techniques like dye markers and binoculars are augmented with modern technologies to overcome limitations. Challenges in measuring bird weight and data storage/transmission are addressed through innovative approaches such as cloud storage, wireless transmission, and data compression. Machine learning algorithms play a crucial role in managing and interpreting vast amounts of data efficiently. By integrating these tools and techniques, researchers can gather accurate data, gain insights, and inform evidence-based conservation decisions effectively, contributing to the protection of biodiversity and ecological integrity.

2.7 Data Storage and Transmission

Technological progress boosts ecological research but faces data storage challenges [22, 23]. This section examines solutions to this critical issue.

2.7.1 Cloud Storage

Significant improvements in wireless access networks or the internet at large is the motivation behind development of internet-based storage and computing. Cloud storage basically refers to infrastructure that allows users to access data on the internet at anytime, anywhere in the world [24]. The word 'cloud' here refers to a cluster of high-performance computers and servers that remotely store and do computations on data [25]. Over the years, this storage paradigm has been deployed by a number of enterprises because of its countless advantages. With cloud storage, one has cost effective [25], convenient, safe and flexible storage [24]. Furthermore, cloud storage has unlimited storage space as opposed to a traditional personal desktops [24].

Within the field of wildlife conservation and monitoring, cloud storage can be used as a central storage for large amounts of data acquired by sensor networks to be used for research. This has an advantage of allowing researchers to collaborate on a single storage platform. Internet storage has recently been deployed by a number of wildlife conservation organisations. For instance, eBird is an internet-based program that was developed by Cornell Lab of Ornithology and National Society to keep bird data at a central location [23]. It engages individuals from all over the world, often called 'citizen-scientists' to make observations on birds [22] [23].

In a nutshell, internet-based data storage is a convenient storage approach that could speed-up analysis and access to data acquired by sensor networks used in wildlife monitoring systems. This storage guarantees high data integrity, confidentiality, and accessibility.

2.7.2 Wireless Data Transmission

Wild animals are often not friendly with electronics around their habitats. Thus, wireless data transmission is an inherently important option to consider when designing a wireless sensor monitoring system for wildlife. There are a number of wireless technologies that have been proposed in literature, and some have been deployed in wildlife monitoring systems. However, the choice of a wireless technology is significantly influenced by its power consumption [26].

Bluetooth is a wireless technology that has been around for about 20 years. It was one of the first wireless technologies to be implemented [26], and has gained increasing relevance recently. Although it would provide a cheap alternative for ecologists to interface with wireless sensors, it is inherently disadvantaged by its power consumption rates, and the high complexities of its protocol stack [26]. Long range wireless technologies could be used to mitigate these problems. However, the wild habitats where the wireless sensor systems will be used are normally out of cellular coverage [27]. Fortunately, developments and active research in wireless technologies have resulted in more power efficient technologies that could be deployed in resource constrained wildlife monitoring systems. For instance, the development of Bluetooth Low Energy [26] or BLE-Mesh [27] promises to resolve the disadvantages of classical Bluetooth. Furthermore, Ayele et al argue that the emergence of recent low power wide area (LPWA) wireless technologies such as LoRa and Sigfox also promise to provide large coverage and lower power consumption [27].

2.7.3 Data Compression

In resource-constrained wildlife monitoring systems, the implementation of data compression techniques is crucial for reducing both storage requirements and transmission bandwidths for data collected by sensor nodes. Data compression, the process of encoding information to use fewer storage and network resources [28] [29], is categorized into *lossy* and *lossless* types. *Lossy* compression techniques allow for data reduction at the cost of losing some information [29], whereas *lossless* compression ensures data is preserved without loss during decompression [28]. Various compression algorithms have been explored in the literature, with some already being implemented in wireless sensor networks for wildlife monitoring. The selection of a suitable compression algorithm largely depends on its efficiency in terms of computational resources, speed, and power consumption, with preference given to those that are cost-effective yet high in computational resource efficiency, catering specifically to the constraints of wireless sensor networks in wildlife monitoring environments.

2.8 Data Processing

Wildlife research struggles with data processing and analysis, despite technological potential [30]. This section reviews traditional methods, identifies challenges, and explores machine learning's capacity to expedite data processing.

2.8.1 Traditional Data Processing

Traditionally, wildlife data gathered by monitoring systems is analysed and processed by researchers manually. For instance, the eBird program that has been mentioned before [23] consists of citizen-scientists manually counting birds and observing their behaviour. This approach to data gathering and analysis has proven to be time-consuming, labour intensive and expensive [30]. Advancement in sensor technologies has significantly improved data acquisition techniques [30], with the challenge of large volumes of data gathered in small amount of time. For instance, Snapshot Serengeti gathered 3.2 million images [22], and processing these images manually could be time-consuming. The demand for autonomous data processing arises from these challenges.

2.8.2 Machine Learning Approaches

The advent of sensor technologies in wildlife research has led to an unprecedented surge in data collection, notably through camera traps, posing challenges in data processing due to the sheer volume of information [30] [22]. Machine learning, a subset of computer science focused on pattern recognition and predictive analysis, emerges as a promising solution to manage and interpret these data effectively [30]. It encompasses both supervised algorithms, which learn from labeled data to predict outcomes for new inputs, and unsupervised algorithms, which infer patterns from unlabeled data [30]. These technologies have been applied in various wildlife monitoring efforts, notably integrating machine learning with unmanned aerial vehicles for the classification of high-resolution thermal imagery. Among the machine learning algorithms deployed, support vector machine (SVM) classifiers and convolutional neural networks (CNN) have shown significant promise in enhancing the efficiency and effectiveness of wildlife conservation efforts [31] [22].

Chapter 3

Scale and Housing Design

This section was completed by Ankush Chohan [CHHANK001]

3.1 Introduction

This sub-module details the process used to design the perch, scale housing, and weighing mechanism to obtain the raw weight data of the Fork-Tailed Drongos. The perch design was based off the current solution that Ben Murphy employs, as it successfully allows the Drongos to land on the scale. The weighing system was designed to attain accurate weight readings, in terms of a voltage which is then processed by the micro-controller sub-system. The entire scale unit is designed to be placed on the floor, with the perch sticking up, allowing the Drongos to land, and feed from a 'food bowl' while their weight data is recorded. This sub-section is divided into two parts, namely, the load cell circuitry, and external housing. The process for the design and implementation of this sub-module is detailed in the following sections.

3.2 Requirements

3.2.1 User Requirements

After an interview with Ben Murphy, the following user requirements relevant to this section were identified.

- The scale must weigh Fork-tailed Drongos with an accuracy of 0.1g.
- The system must identify the bird which is being weighed.
- The system must be deployed in the Kalahari Reserve.

3.2.2 Requirements Analysis

- **Accurate Weighing of Fork-Tailed Drongos:** This requirement specifies that the scale must weigh the birds with an accuracy of 0.1g. This implies that the system must be capable of precise measurements tailored to the weight range of the birds.
- **Identification of Birds:** This suggests the need for a bird identification mechanism embedded in the solution.

- **Deployment in the Kalahari Reserve:** This implies considerations such as ruggedness, and weather resistance of the design.

3.2.3 Design Specifications

- **Accurate Weighing of Fork-Tailed Drongos:** A load cell and corresponding circuitry must be employed to detect weight changes with a sensitivity of 0.1g, and output this value as an analogue voltage.
- **Identification of Birds:** The hardware must be designed such that an RFID scanner can be placed at the correct distance to a bird on the perch.
- **Deployment in the Kalahari Reserve:** The solution must be waterproof and dustproof (IP54) [1], as well as protect the internal components from outside temperatures of up to 50°C.

3.3 Design Choices

3.3.1 Load Cell

For weight measurement, the use of an electronic load cell was found to be the most commonly used mechanism and was employed for this sub-module. Full bridge load cells consist of four strain gauges placed on a metal bar in a wheat-stone bridge configuration. When a mass is placed on the load cell, the metal bar bends slightly, thus changing the resistance of the strain gauges, which then output a voltage depending on the degree of deflection on the metal bar. The voltage is in turn proportional to the weight of the mass applied.

Full bridge load cells offer several benefits, they provide higher accuracy compared to other types of load cells [2] due to the redundancy and balancing of multiple strain gauges within the bridge configuration. The bridge configuration enhances sensitivity to small changes in force or weight, making them suitable for applications where precise measurements, such as the 0.1g accuracy specification, are necessary. One of the main reasons why a full bridge load cell was chosen, was due to the thermal compensation of the configuration. The operation of the load cell depends on very small changes in the resistance of the strain guages. Changes in their resistance are caused by either deflection of the bar, or changes in temperature, where resistance decreases with an increase in temperature. However, since the temperature of all the strain guages will increase/decrease by the same amount, and the output voltage is gained via a voltage divider principle, temperature changes have a negligible effect on the output readings. This is an important principle due to the location that the scale is designed to be deployed in, where daytime temperatures can reach up to 50°C.

For this application, an HKD Electronic load cell of 1Kg capacity was bought from Communica. While the Drongos weigh between 42 and 62g, the load cell has to withstand the weight of the perch, as well as the force of the birds landing and taking off, thus leading to the selection of a capacity of 1Kg.

An entire unit was purchased for this application, to ensure the highest possible accuracy. The

[1] IP54 means protection from limited dust ingress and protected from water spray in any direction. Find more information here: https://www.iec.ch/ip-ratings

[2] E.g Half bridge, and capacitive load cells

alternative was to purchase 4 strain gauges and mount them onto a flexible bar, however, due to the difficulty of the mounting process, as well as procurement of bar that would bend consistently, resulted in the decision to acquire a complete unit despite the increased cost.

3.3.2 Load Cell Circuitry

Amplification, and protection, circuitry is required to convert the output signal of the load cell to a level which the micro controller can read in accurately. Load cells output very small voltage readings, typically in the millivolt range, which is too small to be accurately sampled by the 12 bit ADC on the ESP32 which is the micro controller that was selected for this design. To explain further, a 12 bit ADC has 4096 levels. The maximum input voltage of the ADC is 3.3V, that means that each level corresponds to 0.8mV, therefore, to achieve the required accuracy, the signal into the ADC has to change by a value greater than 0.8mV for every 0.1g. The voltage from the load cell will therefore be propagated through the following systems:

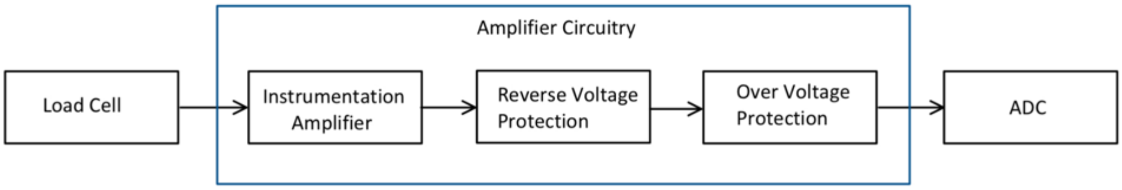


Figure 3.1: Amplifier Circuitry

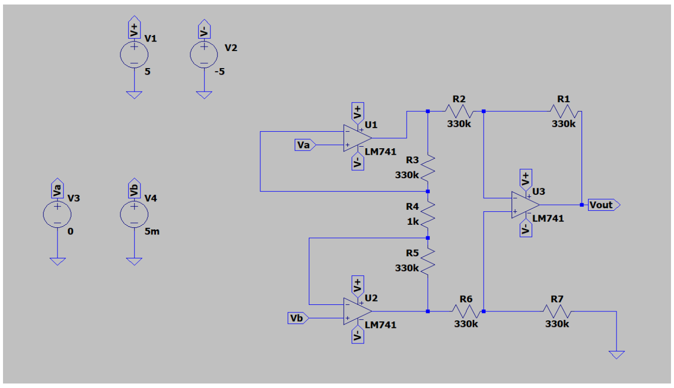
Instrumentation Amplifier

An instrumentation amplifier configuration was selected for this design. Instrumentation amplifiers are used to provide large amounts of gain for very low level signals [32]. This is a useful property for the function of amplifying the small voltage outputted by the load cell to a value that the ADC can use. Instrumentation amplifiers are in fact often used to to directly amplify signals from passive sensors such as strain guages [32]. Additionally, instrumentation amplifiers have a high common-mode rejection ratio (CMRR), therefore, the amplifier will be less affected by electro-magnetic interference and noise that could arise from the other modules of the entire system, especially the RFID scanner. Instrumentation amplifiers are designed to maintain stability over a wide temperature range [3] which is desirable since the scale is designed to be used outdoors, in the Kalahari.

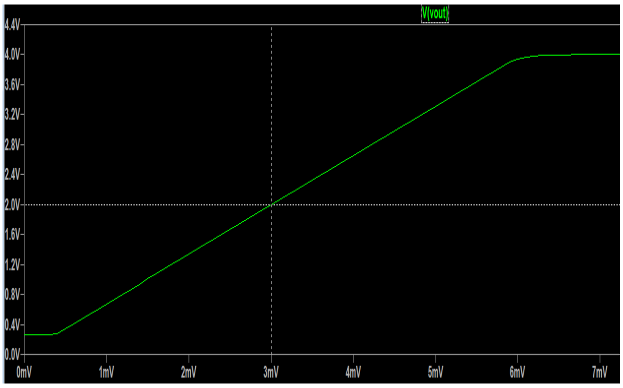
The design of the instrumentation amplifier began with the circuit shown in Fig 3.2.

A gain of approximately 660 was chosen to amplify signals from the mV range to the V range, i.e. 5mV is amplified to the maximum ADC input voltage of 3.3V. Fig 3.2b shows that the response is linear, but saturates at 4V, which turns out to be a negligible characteristic as voltages above 4V will have to be rejected anyway due to the ADC's capacity. The circuit was built on a breadboard, however, it did not function as expected. After some research it was found that an instrumentation amplifier with the precision required for this design cannot be made from basic lab components, as it requires

[3]The temperature range for specified performance of the AD620A is -40 to +85°C



(a) Amplifier Circuit



(b) Amplifier Response

Figure 3.2: Amplifier Simulation

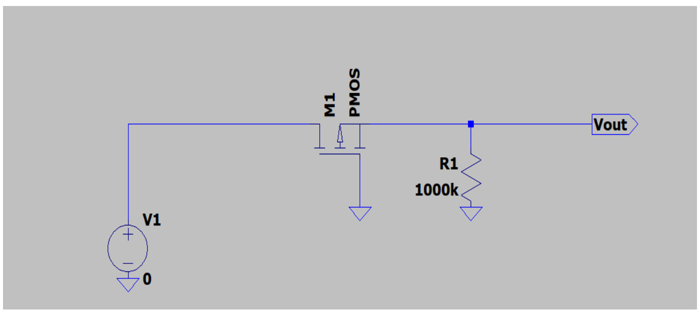
precision resistors to achieve accurate and stable performance. Therefore, a complete unit AD620 instrumentation amplifier was purchased from micro-robotics.

The AD620 was chosen for the following qualities:

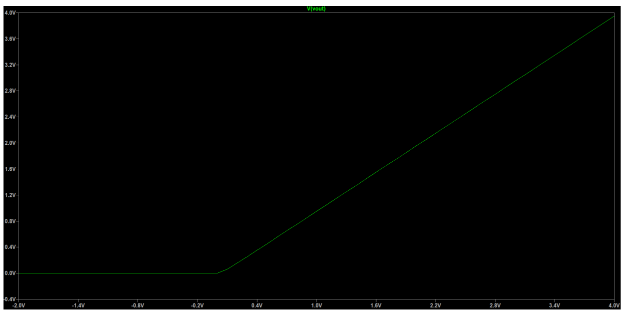
- It is a low cost, high accuracy amplifier
- It requires only one external resistor to set the gain between 1 and 10 000.
- It offers low power consumption making it a good fit for this battery powered application.
- It can be supplied with ±5V which is within the power sub-system's capability.

Reverse Voltage Protection

If the load cell bends downwards i.e. in the expected direction, then a positive potential difference is present at the output. If however, it is to bend upwards, which could happen during the takeoff or landing of a bird on the scale, a negative voltage will be present at the output. This negative voltage could damage other electrical components present in the design. Thus, a reverse voltage protection circuit is implemented at the output of the instrumentation amplifier. The chosen circuit is shown as follows in simulation:



(a) Reverse Voltage Protection Circuit



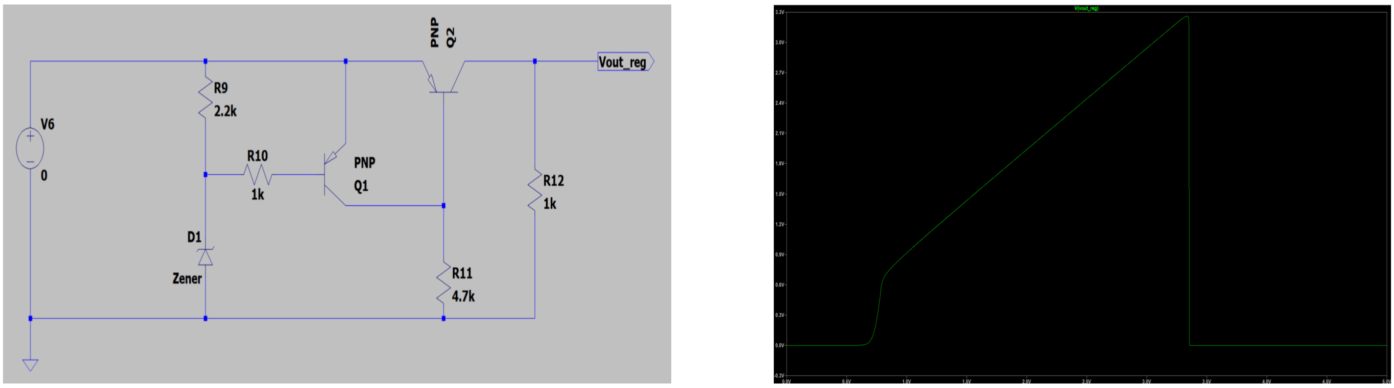
(b) Reverse Voltage Protection Circuit Response

Figure 3.3: Reverse Voltage Protection Simulation

As can be seen from Fig 3.3, the circuit rejects negative input voltages, but allows positive voltages through. For this application an, IRF9540N P-Channel MOSFET was selected, and purchased from Micro Robotics. The main considerations for choosing this particular MOSFET as as follows: The IRF9540N has a relatively low on-state resistance $R_{DS(on)}$ of 0.2 Ω which will result in minimal losses, allowing the battery consumption to be optimised. Additionally, the IRF9540N has an operating temperature of 175°C and is equipped with a heat-sink, allowing for optimal performance in the field.

Over Voltage Protection

The maximum voltage that the ADC pin of the ESP32 micro-controller can handle is 3.3V. Depending on the mass placed on the scale, the output of the amplifier circuit can reach voltages above 3.3V. Therefore, to avoid damage, an over voltage protection circuit needs to be implemented. The following circuit shown in simulation implements the required over-voltage protection functionality:



(a) Over Voltage Protection Circuit (b) Over Voltage Protection Response

Figure 3.4: Over Voltage Protection Simulation

A zener diode with a breakdown voltage of 3.3V is used to allow the output to cut off at values larger than 3.3V. A PN2907A transistor was used to build the circuit. This transistor was selected due to it on hand availability, low leakage current [4] which helps in minimizing power consumption which is essential for this battery operated device, and low saturation voltage [5] which means that it can switch on an off rapidly with minimal power loss.

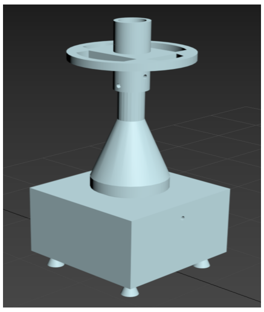
3.3.3 Design Choices for Housing

The design of the housing of the scale is based on the current set up that Ben uses in the field.

The scale components were modeled using 3ds Max, and subsequently printed using a Prusa Mini 3D printer. The system was designed in 10 pieces, and then put together using glue for some parts, and M4 nuts and bolts for others.

[4] 50nA at 25°C
[5] 0.3V to 0.5V depending on collector and base current

(a) Ben's Setup



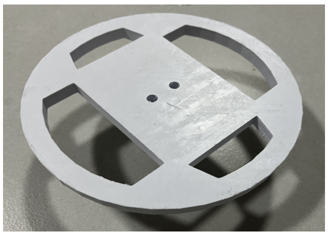
(b) CAD Model of Scale

(c) 3D Printed Scale

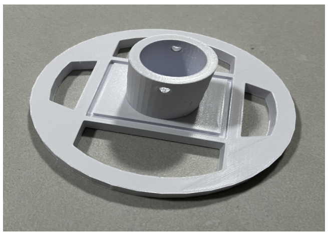
Figure 3.5: Complete Setups

Perch and Pole

The bowl placed at the very top of the scale is designed to hold the food that will attract the Drongos to land on the perch which is the circular section with gaps. The width of the circular part of the perch is 1cm to allow the Drongo's to hold onto it with their small feet.



(a) Top Of The Perch

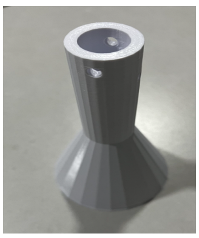


(b) Underneath Of Perch

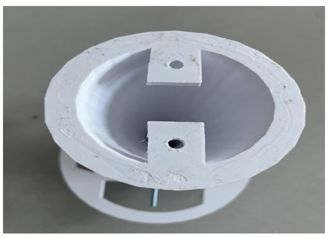
(c) Image Showing Cable Hole

Figure 3.6: Perch

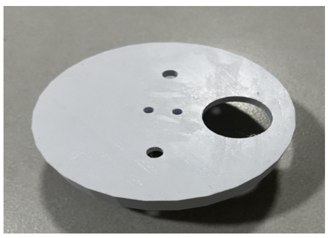
Fig 3.6a shows the M4 holes to connect the food bowl to the middle of the perch. The food bowl has corresponding holes on its base. Fig 3.6b shows the underneath of the perch where the RFID scanner will be placed. The rectangular groove has dimensions of 50x62x3mm to house the RFID antenna, protecting it as much as possible. The image also shows the holes used to fix the perch to the pole. Fig 3.6c shows the perch connected to the pole, as well as a hole to feed the cable of the RFID antenna through and into the base of the scale. The circular section was designed to have a maximum distance of 4cm from the RFID coil as per specification supplied by the micro-controller subsection.



(a) Pole



(b) Underneath Of Pole

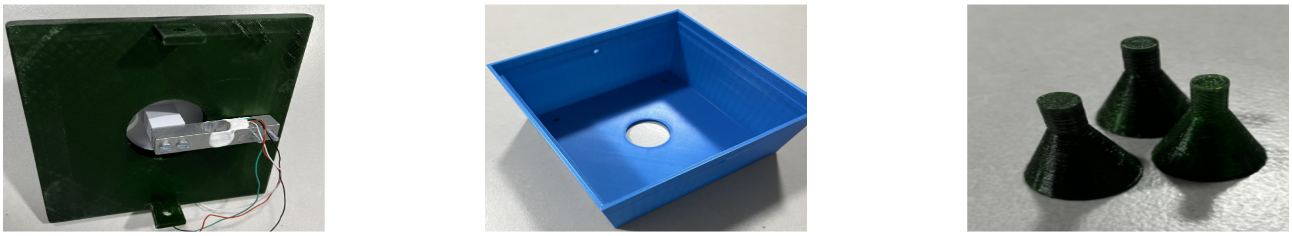


(c) Top of Plate

Figure 3.7: Pole and Plate

As shown in Fig 3.7a, the pole has M4 holes at the top to fix onto the perch. The base has a conical design to avoid the buildup of debris on the scale which would affect the weight measurements. Fig 3.7b shows the underneath of the pole, which has been hollowed out to allow the cable from the RFID antenna through, as well as allowing the part to be printed faster, and with less material. The fixtures that are be bolted to the plate can also be seen in the image. The plate shown in Fig 3.7c connects to the pole on its top section, and to the load cell on its underneath. The large hole is to allow the cable from the RFID scanner through to the base of the scale.

Load Cell Fixture and Base



(a) Load Cell Fixture (b) Scale Base (c) Legs

Figure 3.8: Load Cell Fixture, Base and Legs

As seen in Fig 3.8a, the load cell is bolted to the underneath of the green cover on one end, and bolted to the underneath of the plate on the other end. This keeps the load cell perfectly horizontal, and fixes the perch, pole and plate firmly to the base of the scale. The green cover was designed to be 1cm thick rather than 0.5cm like many of the other components. The added thickness is to ensure that the cover does not bend when a mass is placed on the perch, as one end of the load cell has to be securely fixed to a stiff support (i.e the cover) to gain accurate readings. The cover has two fixtures with M4 hols to bolt to the box. Fig 3.8b shows the base of the scale where all the electronic components of the design are to be stored. It has and inset around the top perimeter to allow the cover to slot in. This provides the cover with maximal support as well as seals the box. There are four small holes in the base of the box to secure the legs. The large hole in the base was included to allow for easy access to components during the testing process and will not be present in further iterations of the design. The legs have a conical shape to prevent them from sinking into the ground when the scale is placed in the field. The weight of the scale will be distributed on the large flat surface of the legs, providing stability to the structure, as well as raising the scale base off the ground to protect it from moisture, and heat from the ground.

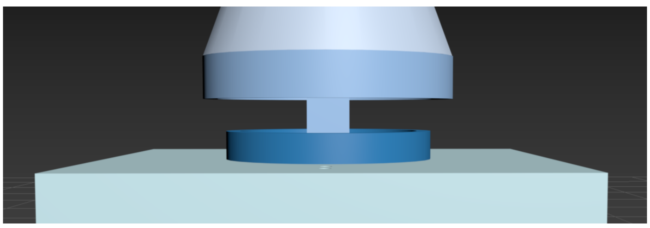


Figure 3.9: Overlap

As can be seen from the above image, there is a lip on the top of the cover that fits into, and overlaps with a corresponding lip on the plate. The figure is taken from the CAD model as it is difficult to see on the 3D printed model. This overlap is designed to protect the components in the box from water and dust.

3.4 Preliminary Testing

Load Cell

The load cell was connected to a power supply of ±5V, and the voltage between the two output leads was recorded after adding a masses of known weight to the perch which the load cell was mounted on.

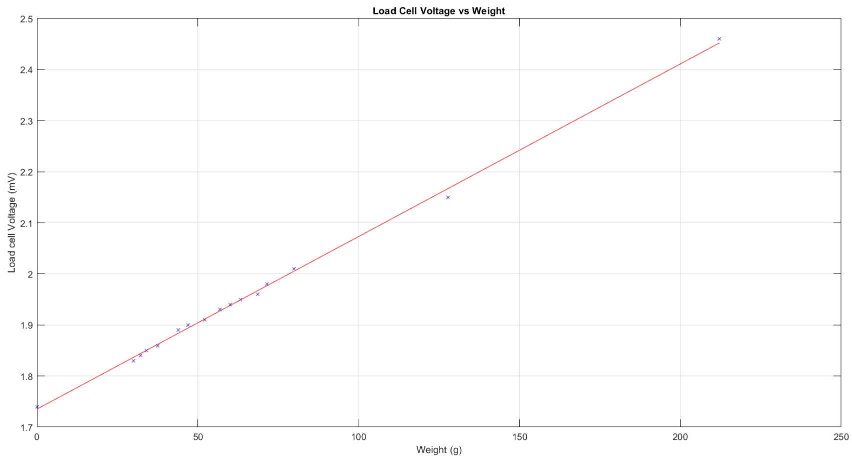


Figure 3.10: Load Cell Response

As can be seen from the line of best fit plotted on the graph, the relationship between mass and output voltage is linear. Ideally with no weight on the perch, the load cell would output 0V, since the bridge configuration is supplied with ±5V, however, due to the weight of the perch itself, the load cell outputs 1.74mV with no added weight on the scale. A description of how this is accounted for is detailed later in the report.

Reverse Voltage Protection

The reverse voltage protection circuit was built on veroboard and tested with a range of input voltages. The results are shown in the following plot:

As can be seen from the graph, the circuit effectively rejects negative voltages, and allows positive voltages through, with minimal losses. At low voltages there is non-linear behaviour, therefore, the overall system is designed to operate at voltages above 0.5V thus nullifying the initial non-linear behaviour.

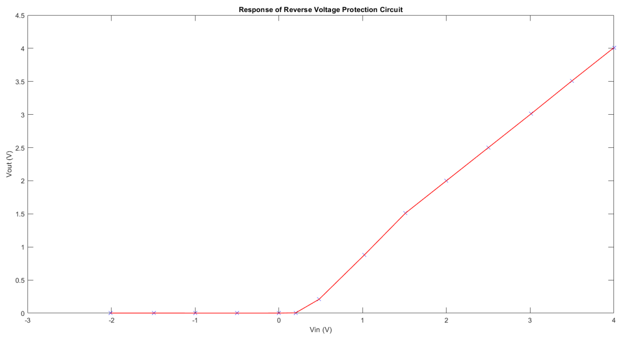


Figure 3.11: Reverse Voltage Protection Response

Over Voltage Protection

The over voltage protection circuit was built on veroboard and tested with a range of input voltages. The results are shown in the following plot:

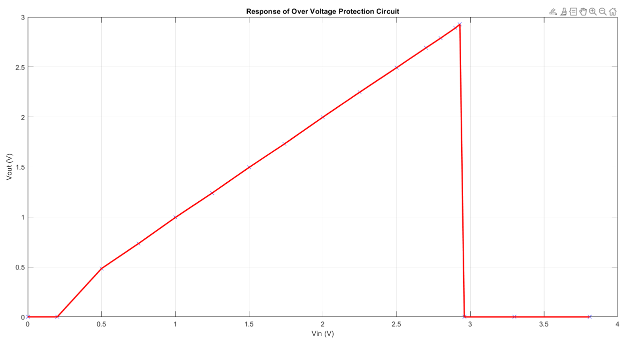


Figure 3.12: Over Voltage Protection Response

As can be seen from the graph, the circuit effectively cuts off the output when the input goes over 2.93V. The circuit was designed to cut off at 3.3V, however, due to the non-ideal behaviour of practical components, it cuts off at a slightly lower voltage. For the required application of protecting the micro controller, the circuit still performs effectively. At low voltages, the circuit has non-linear behaviour. This was seen in simulation, as well as in practice. As was the case with the reverse voltage protection circuit, the overall system is designed to operate at voltages above 0.5V thus nullifying the initial non-linear behaviour.

3.5 Subsequent Design Choices

To account for the 1.74mV output of the load cell when no weight was placed on the perch, a differential amplifier configuration was employed. This effectively 'zeros' the scale to account for the weight of the perch itself, allowing the full voltage range (0.5V-2.9V) to be used for weighting the birds.

The updated circuitry of the system is shown in the following block diagram:

The voltage from the load cell can not go to the differential amplifier directly, since the differential amplifier can not accurately process such small voltages, therefore, the output of the instrumentation amplifier is fed into the difference amplifier. The theoretical gain of the instrumentation amplifier was set to x2745, as this gain resulted in the output being slightly less than 3.3V when there was no load

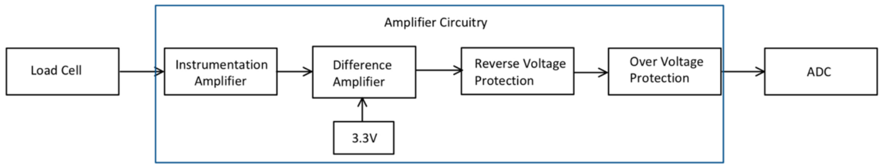


Figure 3.13: Amplifier Circuitry Updated

on the scale. The scale was then calibrated by adding small weights until the output was 3.3V. The output from the instrumentation amplifier, as well as a constant 3.3V from the micro controller are inputted into the terminals of the difference amplifier to zero the scale.

Differential Amplifier

The Differential Amplifier circuit is as follows:

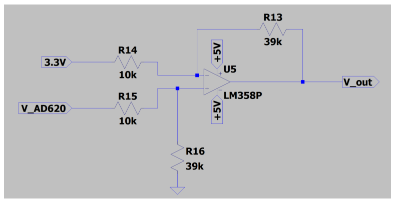


Figure 3.14: Differential Amplifier

The LM358P was chosen due to on hand availability, as well as the fact that it can be powered off the same ±5V supply as the rest of the circuitry. The gain of the differential amplifier is set to a value of x4 to further amplify the signal.

3.6 Final results

Housing Testing

A paper towel was placed in the box, and water was splashed on the entire structure. The paper towel was then inspected and showed no signs of moisture.

Circuitry Testing

The entire system for this subsection was connected up, and tested with various values of known weight being placed on the perch. Fig 3.15 shows the results:

Weights around the average mass (44g) of a Fork-Tailed Drongo were used to test the system. As can be seen from the graph, the system outputs voltages in the range of 0V-2.9V corresponding to masses of 0g-64g. If a weight larger than 64g is placed on the scale, the voltage from the difference amplifier exceeds 2.9V, and the over voltage protection circuit kicks in, and cuts off the signal. This is shown by the data points at 65g and 67g. All of the expected weights output voltage readings above 0.5V, thus validating the fact that the design operates in the linear region of the over voltage and reverse voltage protection circuits.
Since the 12 bit ADC has 4096 levels, and takes in voltages up to 3.3V, each level corresponds to

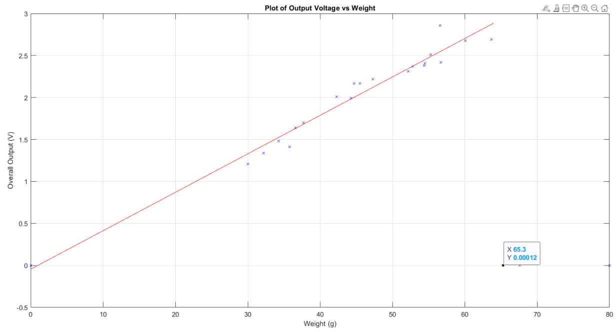


Figure 3.15: Overall Scale Results

0.8mV. Since the output range of the scale is 0V-2.9V, the output voltage can be represented by 3625 levels [6]. Those levels correspond to 0g-64g. therefore, each level corresponds to 0.017g [7]. Therefore, the scale has an accuracy of 0.018g. It is important to note, however, that this calculation is based solely on the output of the load cell circuitry. The actual accuracy of the system will be determined in the Microcontroller subsection, after the signal from the load cell circuitry has been processed, and mapped to weight.

3.7 Acceptance Test Procedures

The following acceptance test procedures were considered for this sub-section:

User Acceptance Testing (UAT)

- **Accuracy of 0.1g:** This is tested by investigating the resolution of the final signal that is outputted. The expected result is that one ADC level corresponds to less than 0.1g
- **The system is weatherproof:** This is tested by subjecting the system to weather conditions or simulations thereof. The expected result is that no water or dust enters the system.

Unit Tests

- **Reverse voltage protection:** This is tested by subjecting the reverse voltage protection circuitry to reverse voltages and observing the output. The expected result is that the circuit rejects reverse voltages and allows forward voltages through.
- **Over voltage protection:** This is tested by subjecting the over voltage protection circuitry to voltages above 3.3V, and observing the output. The expected result is that voltages below 3.3V are allowed to pass, while voltages above 3.3V are cut off.
- **Water proof to IP54 standard:**[8] This is tested by splashing water on the housing and observing if any moisture enters it. The expected result is that there will be no trace of water in the housing.

[6] 2.9/0.8mV

[7] 64/3625

[8] IP54 means protection from limited dust ingress and protected from water spray in any direction. Find more information here: https://www.iec.ch/ip-ratings

- **Dust proof to IP54 standard:** This is tested by deploying the system in the field and observing if any dust enters the housing. The expected result is that no dust will enter the housing.
- **Protection from temperatures up to 50°C** This is tested by subjecting the system to high temperatures in the field. The expected result is that the system will continue to operate optimally.
- **Scale subsystem interacts with microcontroller subsystem:** This is tested by connecting the output of the load cell circuitry to the ADC pin of the microcontroller. The expected result is that the microcontroller reads in the voltages.

3.8 Results of Acceptance test Procedures

ATP	Result	Comment
Accuracy of 0.1g	Pass	Load Cell circuitry has an accuracy of 0.018g
Reverse voltage protection	Pass	
Over voltage protecton	Pass	
Water proof to IP54 standard	Pass	Partially satisfies UAT 2
Dust proof to IP54	N/A	Needs to be tested in the field
Protection from temperatures up to 50°C	N/A	Needs to be tested in the field
Interaction with microcontroller	Pass	Further details are in the UC subsection.

3.9 Discussion and Conclusions

This subsection successfully met its design requirements and specifications as stipulated in section 3.2.3 by being a sensitive, and weatherproof solution that provides a readable output. All of the testable ATPs of the subsystem were passed. The protection circuitry fulfilled its requirement to protect subsequent electronics, especially the micro controller, from reverse, and over voltages. The output of the entire system was a linear response as shown in fig 3.15, and was successfully fed into the ADC pin of the micro controller. The 3D printed housing passed the test for IP54 waterproofing, however, the tests for dust proofing, and temperature resistance could not be done as these would have to be tested either in the field, or with specialised equipment, which was not acquirable. A load cell is often used with an HX711 24-Bit ADC to process the small load cell voltage, however, for this application, processing circuitry was designed from scratch to make the system more sensitive to the expected weights. This design choice proved to be beneficial, as shown by the high sensitivity of the system between 0-64g. The design of the housing was largely based on its shape in terms of optimizing the amount of material used as well as its functionality. The perch specifically was designed to have the Drongos be able to land on the scale, and be close enough to ping the RFID scanner while they feed from the food bowl. The housing was 3D printed using a material that is not well suited to long periods of outdoor exposure, and consideration was not given to other types of material that could be used to build the housing. This is an aspect of the design that needs to be considered in further iterations. Recommendations can be found in Chapter 8.2. Overall, Scalio's Scale and Housing subsection performs well, successfully meeting the design requirements.

Chapter 4

Data System Design

This section was completed by Kananelo Chabeli [CHBKAN001]

4.1 Introduction

This chapter details the design process of data acquisition and retrieval subsystem. This subsystem is responsible for detecting drongos birds when on the scale and estimate its weight saving the results to file in memory for later retrieval. To address the problem effectively, the system was further divided into two sub-subsystems: **data acquisition**, and **data storage and retrieval**. the former detects the bird when it lands on the scale and samples noisy analog signal from scale system. This signal is then filtered to get a better approximation of the bird's weight. Data storage and retrieval, is tasked with storing data into the system memory and retrieving it when needed. This subsystem also integrates hardware wireless communication interface that communicates with the mobile application. The chapter begins with detailed analysis of requirements, design choices made to meet these requirements and system design procedures followed to implement these choices. Moreover, tests results made on the system and acceptance test procedures are presented and conclusions are drawn based on these results.

4.2 Requirements

The user requirements of the system were gathered and refined through continuous engagement with the end-user (Ben). The requirement are analysed in section 4.3, when considering different design options available to meet each user requirement. From these analysis, the user requirements were effectively translated into function requirements and design specifications. Table 4.1 below summarises the outcome of the process.

Table 4.1: Summary of system requirements and design specifications

User Requirements	**Functional Requirements**	**Design Specifications**
R01: System must be power efficient	Must be driven to deep sleep mode when IDLE	Must draw 10 times less current when in sleep mode
R02: The system must be cost effective	Must be designed to minimize operational and maintenance costs	Total cost must not exceed R500.00
R03:System must transmit data wirelessly	System must support Bluetooth Low Energy and WiFi	Should have maximum latency of 2 seconds
		Must maintain minimum data transmission success rate of 99%
R04:Must estimate weight with 0.1g uncertainty	Must deploy optimization digital filter	Will deploy Kalman filtering algorithm
		Should have ADC with minimum of 10-bit resolution
		Calibrate gain and offset error using 2-point calibration
R05: System should operate in Kalahari conditions	Should be small enough to fit within the scale	Must be soldered on verboard
		Must operate in temperature above 45 C

4.3 Design Choices

4.3.1 Development Boards and Microprocessor Selection

Three micro-controllers were evaluated for the project based on their ability to meet specified user requirements: the Xtensa LX6 on the ESP32 DevKitC board, the ATMEGA328P on the Arduino Uno R3 board, and the Xtensa LX7 on the ESP32-S2 DevKit development board. These controllers were selected primarily for their wireless communication capabilities (R03). The key user requirements influencing the choice were R01-04 in table 4.1. Requirement R01 specifies low power usage, which is achievable through a processor with a high clock speed, an Ultra-Low Power (ULP) co-processor, and minimal current consumption in sleep mode.

Additionally, the system must be cost-effective and capable of shutting down power to ancillary subsystems like the scale system when not actively acquiring data or transmitting it. To satisfy R04, the system requires a high-resolution analog-to-digital converter (ADC) for accurate sampling analog signal from the scale and dynamic digital filtering, suggesting a need for more RAM and higher processor speeds. The microprocessor selected also needs to support I2C for communication with RTC module, and have ample SRAM and ROM. A comparison of these micro-controllers is provided in Table 4.2.

Table 4.2: Microprocessor comparison

Features	ESP32 Dev(Xtensa LX6)	ESP32-S2 Dev(Xtensa LX7)	Uno R3(ATMEGA328P)
cost	R205	R199.94	R120
Sleep current	10 μA	$25\mu A$	$0.24mA$
CPU Speed	240MHz	240MHz	16MHz
ADC Res	12-bit	12-bit	10-bit
ROM	448KB	128KB	512B
RAM	520KB SRAM	320KB SRAM	2KB SRAM
Cores	Dual Core	Single Core	Single Core

The table reveals that the ATMEGA328P microprocessor fails to satisfy most of the requirements except for cost-effectiveness. However, the modest cost difference is not justified by its limited capabilities, leading to its exclusion in the initial selection phase. In contrast, both the Xtensa LX6 and Xtensa LX7 microprocessors advanced to the next round of evaluation due to their shared features, such as 240MHz clock speeds and 12-bit ADCs, fulfilling the requirements for high-speed processing and high-resolution ADC. Additionally, both processors are reasonably priced.

Among these, the Xtensa LX6 (ESP32) stands out with more memory and a dual-core architecture, enhancing its ability to handle multiple tasks simultaneously through multi-threading with reduced time-slicing compared to single-core processors. This capability ensures smoother system operations. Moreover, the Xtensa LX7's higher current consumption in sleep mode makes it less appealing for fulfilling the low power requirement (R01). Stakeholder feedback emphasized the importance of power efficiency, measurement accuracy, and ease of data retrieval. Consequently, the Xtensa LX6 on the ESP32 DevKitC development board was chosen as the most suitable processor for this system.

4.3.2 Bird Identification

For the system to be power efficient, effectively meeting requirement R01, it was necessary to drive ESP32 into deep sleep mode. Thus, a mechanism to automatically detect when the bird is on the scale and start taking samples was needed. Two possible options were considered: **Image Bird Detection** and **RFID Tag Detection**. The former translated to implementing object detection algorithm which obviously needed tons of data to achieve higher generalisation. Furthermore, it meant keeping the camera on almost always and continuously monitoring objects in the field of view of the camera to detect presence of Drongos. However, it was found to be much cheaper, thus is would meet requirement R02.

Contrary to that, using RFID reader to detect birds turned out to be extremely power efficient. For instance, the reader and micro-controller could both be driven to sleep mode when no acquisition process is under-going, and reader interrupts the controller when bird's RFID tag is detected in range of the reading antenna. The drawback of this approach was that the RFID Reader module are costly. Thus, a trade-off needed to be made between cost and power efficiency of the system. Accessing cost and power from all other systems, and considering the user weights (preferences) to costs and power effectiveness, the method of detecting birds was chosen to be through the use of **RFID Reader module**, and costly side of it were traded-off for power efficiency.

4.3.3 Data Storage, Retrieval and Transmission

Data Storage and Retrieval For data storage, external non-volatile memory options such as EEPROM or SDCard were initially considered. However, these options increased the overall system costs and the complexity of the circuitry. Therefore, internal memory was chosen for file storage. The ESP32 utilizes the LittleFS file system, which is efficient and user-friendly. It allows for file operations similar to those in a standard operating system, enabling data to be stored in a readable and useful manner.

Data Transmission Because the system is required to work in harsh Kalahari Desert conditions, traditional data retrieval methods such as plugging device via USB port present significant challenges. First, desert dust could fill-in the USB port and damage its internal circuitry. Secondly, continued tempering with the scale system could result is malfunctioning of the system, needing repeated reboots. Lastly, it would not be feasible to wait under hot Kalahari sun and load data manually into a device.

Alternative approach is to use wireless data transmission techniques. The ESP32 board supports Bluetooth Classic, Bluetooth Low Energy (BLE) and WiFi. Classic Bluetooth consumes more power per each byte transmitted compared to BLE. Also, Wifi is also power consuming technology but offers more data rates and security. To achieve power effectiveness, BLE was selected to used when transmitting less payload data such as calibrating the scale (zeroing instructions from the battery), and sending of battery life. For data integrity and security, BLE has flaws as it is less secure than WiFi. Furthermore, setting up ESP32 as both Server and WiFi Access Point(AP) turns out to be the most convenient to manage data transmission. As such, WiFi is selected to be used when the user want to download the data-file to the mobile phone.For this transmission, ESP32 is set as the HTTP Web Server and WiFi AP, while the mobile will be acting as the client, and a WiFi station. In this way, the user can set custom WiFi password and SSID.

4.4 System Design

4.4.1 Hardware Interfaces

This section outlines how the system interacts with other subsystems through various interfaces. The first interface, HW-I01, consists of two power lines (5V and GND) that supply power to the ESP32, RTC module, and RDM6300 RFID reader module. The second interface, HW-I02, involves a single line from the scale that transmits an analog voltage proportional to the bird's mass. This voltage is processed during data acquisition to estimate the bird's weight.The third interface, HW-I03, includes connections between the ESP32 and the RFID reader. It has a UART Transmit line (Tx) for reading the identification number from bird tags and an Interrupt line to wake the ESP32 from sleep mode when an RFID tag is detected.Moreover, HW-I04 is the interface between the ESP32 and the Real-Time-Clock (RTC) module, which includes a serial data line and a serial clock line for accurate timekeeping.Lastly, HW-I05 features a push-button used to interrupt the ESP32's sleep mode for data download or system configuration. These interfaces are detailed in the accompanying table 4.3.

Table 4.3: System Hardware Interfaces

Interface ID	**Description**	**Pin Definitions**	
HW-I01	Power Interface	**ESP32 PIN**	**Description**
		ESP32 Vin	5V from Power System
			Common ground with Power System
HW-I02	Voltage from Scale System	Connects to GPIO39 of ES32 Board	
HW-I03	RFID Reader Interface	**ESP32 PIN**	**RDM6300 PIN**
		ESP32 GPIO5	UART Tx Pin
		ESP32 GPIO21	Interrupt Pin (Labelled LED on RDM6300)
HW-I04	RTC D3132 Interface	**ESP32 PIN**	**D3132 PIN**
		ESP32 GPIO22	I2C CLK
		ESP32 GPIO21	I2C SDA
HW-I05	User Button	ESP32 GPO34	

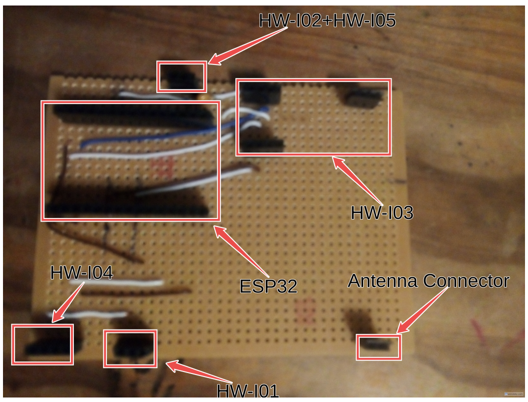


Figure 4.1: Hardware interfaces

4.4.2 Software Interface

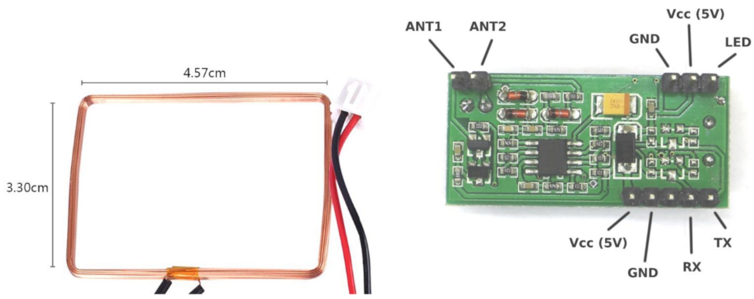
The software interface was designed for interacting with the mobile app. This consists of application programming interface that the mobile app interfaces with to retrieve data from the ESP32 Web Server. Each acquired data entry saved in the encapsulated into the C-structure:

```
typedef struct data_entry_t{
    String tag_id;
    String time;
    String data;
    float weight;
};
```

During data retrieval, the function $int\ get\_data(File *\ root, data\_entry_t *\ data)$; is invoked. For each invocation, this function retries next data entry from the data file store in LittleFS partition, and encapsulates it into structure mentioned above. This is then translated to JSON format for wireless transmission to the Mobile.

4.4.3 Data Acquisition

As mentioned before, the Data System is divided into Data Acquisition and Data Retrieval sub-modules. This section presents detailed design of data acquisition component. Data acquisition process is tasked with detecting when bird is on the scale, sampling analog voltage from scale and approximating weight of the bird. The system utilizes RFID technology to identify birds via tagged IDs when they are on the scale, with each measurement uniquely linked to the bird's Tag ID. Initially, a 134.2kHz RFID reader module was chosen to match the frequency used by the birds, but due to stock unavailability, a 125kHz module was selected as a temporary alternative for testing. The setup, including the reading antenna, is depicted in fig.4.2.



(a) 125kHz Antenna

(b) RDM6300 125kHz RFID Reader module

Figure 4.2: RFID Reader and Reading Antenna

To conserve energy, the system operates in deep sleep mode unless actively acquiring data. The LED pin, shown in fig.4.9b, facilitates system interruption when an RFID tag comes within 5cm of the antenna, triggering a response at the falling edge. Other connections include pins ATN1 and ATN2 for the antenna (without polarity concerns) and a Tx UART pin for reading the tag's ID, as outlined in table 4.3. Power is supplied through VCC 5V and GND pins connected to the power interface, **HW-I01**, while remaining pins are unused.

Analog-to-Digital Converter When the bird lands on the scale, the scale will output analog signal that needs to be sampled and be processed. ESP32's 12-bit ADC is used in this project to obtain digital signal. The ADC input voltage range is 0V- 3.3V, and is supported by 16 channels on ESP32.Unfortunately, this ADC suffers extremely from DC non-ideal effects such as gain error, offset error, and differential non-linearity. To achieve specified precision of weight measurement as in table 4.1, these effects needed to be calibrated out. The method used to calibrate ADC's gain and offset errors is two-point calibration method.

This method involves taking two inputs to the ADC at ($V_{in,A}$) 10% and ($V_{in,B}$) 90% of full scale and record the corresponding ADC codes. The slope of non-ideal transfer function is then calculated as follows:

$$Slope_{non-ideal} = \frac{ADC\ Code_B - ADC\ Code_A}{V_{in,B} - V_{in,A}} \tag{4.1}$$

Using arbitrarily point ($V_{in,A}, ADC\ Code_A$), the non-ideal transfer function is given by

$$ADC\ Code = Slope_{non-ideal}(V_{in} - V_{in,A}) + Code_A \tag{4.2}$$

Substituting $V_{in} = 0$ in equation 4.2, gives the ADC offset error. The linear model of an ideal 12-bit ADC transfer function is given by

$$ADC\ CODE_{ideal} = 2^{12}\left(\frac{V_{in}}{V_{ref}}\right) \tag{4.3}$$

Thus one can easily use equation 4.3 to find the ideal ADC code for the calibration input voltages $V_{in,A}$ and $V_{in,B}$, and the slope of ideal transfer function can then be defined by

$$Slope_{ideal} = \frac{Code_{B,ideal} - Code_{A,ideal}}{V_{in,B} - V_{in,A}} \tag{4.4}$$

and ADC gain error is given by:

$$Gain\ Error = \frac{Slope_{non-ideal} - Slope_{ideal}}{Slope_{ideal}} \tag{4.5}$$

With this offset and gain error, the calibrated ADC code is given by:

$$Code_{calibrated} = (Code_{raw} - offset) * \frac{Slope_{ideal}}{Slope_{non\_ideal}} \tag{4.6}$$

Kalman Filtering With ADC calibrated, we needed to design a filter to mitigate the noisy analog signal from the scale as result of dynamic bird on the scale. The filtering used is one-dimensional Kalman filter. This filter analyses the system variable(input voltage from scale) over time, and attempts to provide optimal estimate of the system variable. It is performed in the following steps:

- **Step 1 Initialization**: Provide initial estimate of the system state $X_{0,0}$ and variance $\rho_{0,0}$.
- **Step 2: Measurement**: This step is performed at every iteration in the filter. In provides current state measurement Z_n and its variance r_n.

- **Step 3: System Update** In this step, the filter uses current measurement, and previous state estimate to obtain current state estimate. This is done if the following steps:
  1. Compute Kalman gain by $K_n = \frac{\rho_{n,n-1}}{r_n + \rho_{n,n-1}}$, where $\rho_{n,n-1}$ is state variance estimated in the previous iteration, and r_n is measurement variance.
  2. Update state using state update equation

$$X_{n,n} = X_{n,n-1} + K_n(Z_n - X_{n,n-1}) \tag{4.7}$$

     where $X_{n,n-1}$ is state estimate obtained in the previous state.
  3. Compute current state variance $\rho_{n,n} = (1 - K_n)$
  4. Extrapolate the state variance and state estimate using dynamic model of the system: $X_{n+1,n} = X_{n,n}$, and $\rho_{n+1,n} = \rho_{n,n} + q_n$, where q_n is the process noise variance.

The block diagram of kalman filter is shown in figure 4.3.

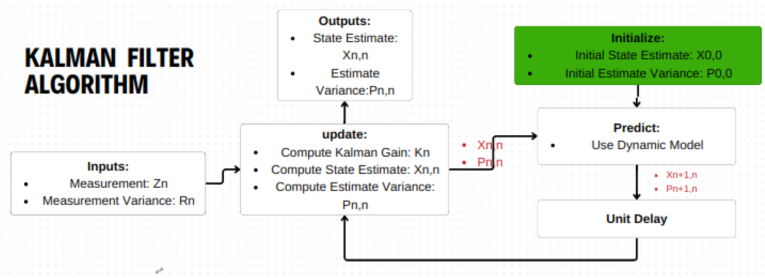


Figure 4.3: Kalman Filter Block Diagram

Figure 4.4 shows the flow diagram of data acquisition process.

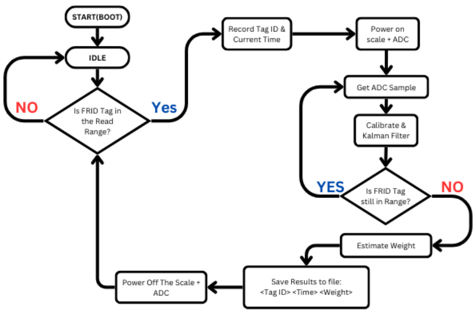


Figure 4.4: Flow chart of data acquisition process

4.4.4 Data Storage and Retrieval

The second sub-module of the Data System manages data storage and retrieval in the ESP32's LittleFS file system. Data entries, formatted as <Tag ID> <Time> <Weight> are logged for each data acquisition event. Tag ID identifies each bird uniquely, Time (in HH:MM:SS) is provided by the precise D3132 Real-Time-Clock module, and Weight is the measured value. Figure 4.5 illustrates the data storage and retrieval process. The system also accepts configuration commands via a mobile app, such as zeroing the scale and setting battery parameters.

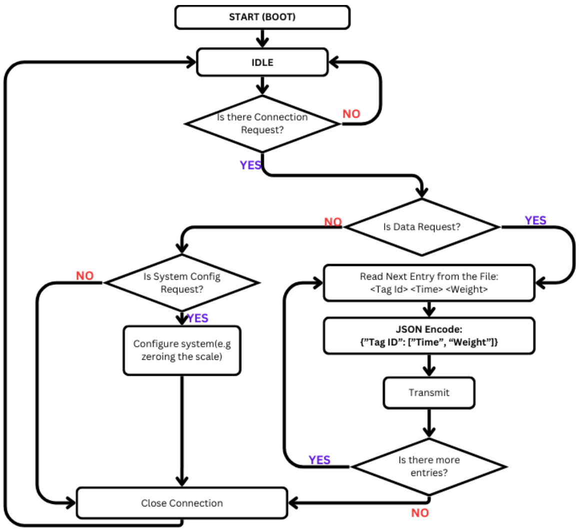


Figure 4.5: Data Storage and Retrieval Flow chart

4.4.5 Overall System Integration

The overall system state machine is shown in figure 4.6 and consists of three states . The IDLE state is the default state upon system boot, indicating the ESP32 is in sleep mode. This state can be exited through two interrupts: one from a bird landing on the scale, detected by the RDM6300 RFID reader module, and the other initiated by a user pressing a button. DATA ACQUISITION, and DATA TRANSMISSION operate as explained in this section.

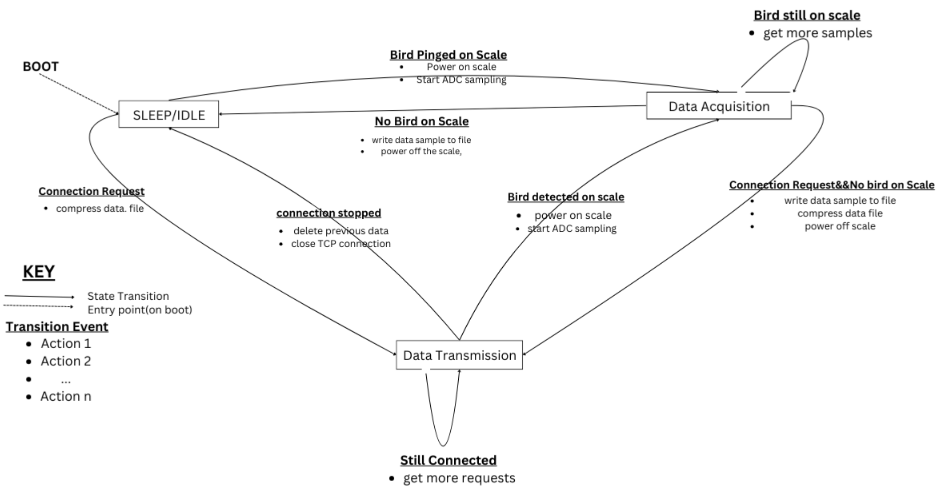
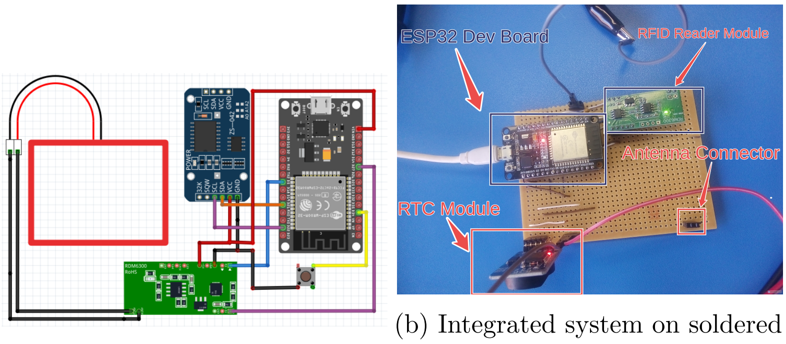


Figure 4.6: Data System State Machine



(a) Integrated system connection (b) Integrated system on soldered board.

Figure 4.7: Overall Data System Integration

4.5 Testing and Results

4.5.1 Acceptance Tests

Table 4.4: System Acceptance Criteria

Label	Description	Pass/Fail
AT01	125kHz RFID tag Detection	Pass
AT02	Calibrate ADC gain and offset Errors	Fail
AT03	Kalman filter convergence & accuracy	Pass
AT04	Timestamp data	Pass
AT05	Weight estimation accuracy test	Fail
AT06	Transmit data Wirelessly	Pass

4.5.2 Acceptance Test Procedures

This section details the procedures to carry the acceptance tests shown in table 4.4.

ATP01 - RFID tag Detection Connect RDM6300 RFID reader module to ESP32 by following table 4.3, and upload $RFIDTest$ sketch. Press reset button on the ESP32, and open serial monitor. Text; *Bring tag closer...* should be displayed. Bring the tag over the reading antenna, the tag's ID should be displayed on the monitor immediately when the tag is within 5cm from the antenna.

ATP02 - Calibrate ADC gain and Offset Errors Connect GPIO39 of ESP32 to power supply as set the voltage to 0V. Upload sketch $ADC - Calibrate$ and press the reset button. ADC codes should start being displayed on the monitor every second. Now set the voltage to 0.3V and press reset button. Note the ADC Code displayed after about 5 seconds. Next set the voltage to 2.97V and also note ADC output displayed after about 5 seconds. Use these information with calibration equations in section 4.4 to obtain calibration parameters of the ADC. Now set to supply voltage to 2.0V use the ADC code displayed with equation 4.6 to obtain the calibrated Code. The expectation is that the calibrated code should ideally map to input voltage (using equation 4.3) with standard deviation of $\pm 0,05V$ from 2.0V.

ATP03 - Kalman Filter convergence . Connect ESP32 to supply power as explained in previous ATP and upload sketch $KalmanTest$, then press reset button. Set the voltage to 2.0V, and shake the wire at input of the ADC to make some significant variations in the input. Observe the filter outputs as they get displayed on the serial monitor. After about 30 seconds, the filter should be converging to 2.0V voltage.

ATP04 - Timestamp data Connect DS3132 RTC module to ESP32 by following table 4.3. The LED on the DS3132 should be turned on if connection is correct. Upload sketch *TimeTest*, and press reset button. Observe the time as it is displayed on the serial monitor every second in the format: $HH : MM : SS$. Confirm that this time corresponds to actual real-time.

ATP05 - Weight estimation accuracy This test procedure is done with scale. connect scale to the power supply as described in chapter 3, and connect GPIO39 of ESP32 to the output of the scale. Upload sketch *ScaleTest* and press reset button on the ESP32 and immediately drop a known mass on the scale (similar to how the bird would land). Additionally, shake the mass on the scale to mimic the dynamic movement of the bird on the scale. After abut 60 seconds, remove the mass and observed the estimated weight on the serial monitor. The test is passed if the estimation has deviation of $\pm 0.1g$ from the known mass.

ATP06- Data transmission This procedure must done with mobile phone having application designed in chapter 6 installed. Upload the sketch *DataTransmit* and press reset button on ESP32. Open the WiFi and the mobile app on the phone. navigate the app as described in chapter 6, to download the file. Verify that the contents of the file are the same as the content printed on the serial monitor.

4.6 Results and Discussion

4.6.1 Obtaining Voltage-Weight Relationship

The load cell used by Scale Subsystem has no explicit transfer function to map output voltage to weight. As a result, several known masses were placed on the scale and corresponding voltages measured. These data are plotted in figure 3.15. A linear model was fitted to this dataset to obtain a regression relationship between voltage output and weight. This relationship was found to be

$$V = 0.04579165 \times W \tag{4.8}$$

where V is output voltage of the scale (input to the ADC), and W is the weight on the scale.

4.6.2 ADC Calibration

The analog output of the scale needed to be sampled in-order to by processed by a digital processor. This sampling would lead to loss of information due to quantization noise and non-ideal effects of the ADC. Acceptance test *AT02* dealt with calibrating gain and offset errors of the ADC. The following are results obtained from carrying this test:

- $V_{ref} = 3.3V$
- $V_{in,A} = 0.1 \times V_{ref} = 0.33V$

- $V_{in,B} = 0.9 \times V_{ref} = 2.97$

The ADC codes obtained are $Code_A = 278$ and $Code_B = 3870$ and were used to calculate the slope of the non-ideal transfer function using equation 4.1 resulting in $Slope_{non-ideal} = 1327$. Ideal ADC outputs, calculated using the ideal transfer function in equation 4.3, were found to be $Code_{A,ideal} = 409$ and $Code_{B,ideal} = 3686$, giving an ideal slope of $Slope_{ideal} = 1241$.

From these measurements, the ADC offset error was found to be $Offset\text{-}Error = -159$. The ideal and non-ideal slopes were then used in equation 4.6 to obtain calibrated ADC codes with gain and offset errors. The quantify this calibration effect, the input to the ADC was set to **2.0V**, and the recorded code was **2384** (averaged over 10 codes obtained). Using the calibration above, this estimates the input to the ADC as **1.916V**, resulting in a **0.084V** uncertainty, thus failing *AT01*. The plot of ideal and non-ideal transfer functions of the ADC is shown in Figure 4.8 below.

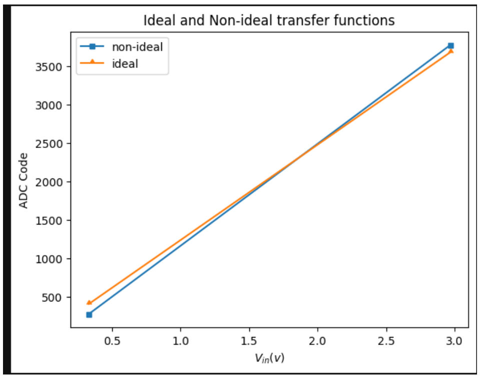


Figure 4.8: Ideal and Non-Ideal Transfer Function of ESP32's ADC1

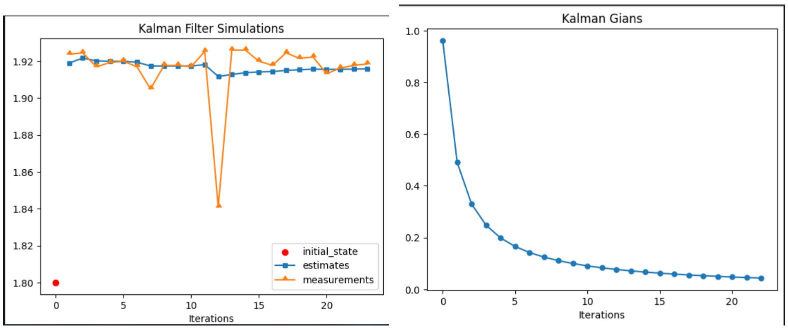
4.6.3 Kalman Filtering

With ADC non-ideal effects mitigated by the aforementioned calibration procedure, it was also necessary to consider the fact that birds will not be static on the scale, hence the need for a dynamic filtering algorithm. The algorithm used is Kalman filtering because of its robustness and convergence speed. This is tested specifically in test procedure *ATP-03*, and further used to test acceptance test *AT05*. The filtering results carried by acceptance test *ATP03* are shown in figure 4.9.

In addition to above results, acceptance test procedure *ATP05* is about testing the accuracy of the system with Scale and know mass which was moved over the scale to mimic bird movement. The resulting plot of the filter output is shown in figure 4.10. As shown in the figure, the Kalman converges to value of about 1.514V. Using equation 4.8, the estimated mass would be 33.1g. However the actual mass of the object used for *ATP-05* was 33.5g. Thus the estimation error of system is $0.4g > 0.1g$, thus failing to meet *ATP05*, and user requirement *R04*.

4.6.4 Discussion on Acceptance Tests

The acceptance tests in Table 4.4 assess whether the system meets specific requirements from Table 4.1. *AT01* checks the system's ability to detect RFID tags and read their IDs, critical for interrupting the system's deep sleep mode (addressing *R01*) when a bird lands on the scale.



(a) Kalman filter output plot vs iterations (b) Kalman gain as function of iterations

Figure 4.9: Results of Kalman filtering from ATP-03

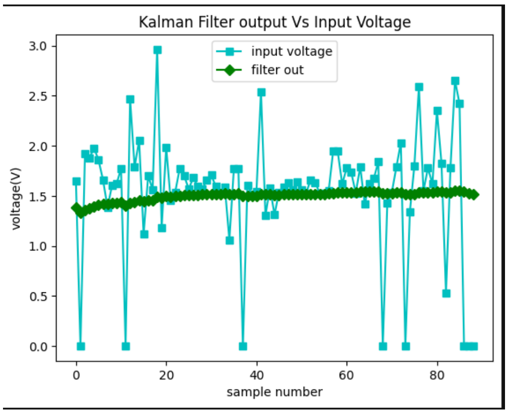


Figure 4.10: Kalman filter outputs as input dynamically varies

AT02 and *AT03* focus on improving weight estimation accuracy (*R04*). *AT02* evaluates how the system compensates for non-ideal ADC effects. Despite efforts to mitigate gain and offset errors, ESP32's ADC still struggles with differential and integral non-linearity.*ATP03* tests the Kalman filter's performance, which ideally converges to the optimal voltage despite variations, essential for tracking the dynamic voltage as the bird moves on the scale. However, the convergence rate of the filter significantly depends on the accuracy of the initial estimate. Finally, *ATP06* examines the system's capability to connect to WiFi and retrieve data, with details provided in Chapter 6.

4.7 Conclusion

The Data system was designed primarily to meet user requirements including a mass accuracy of 0.1g, power effectiveness, and wireless data transmission. Testing results showed that the system failed to meet the mass accuracy requirement due to several factors, including the limited datasets used to predict the mass-voltage regression. Furthermore, the 12-bit resolution used was not the optimal choice. With adjustments in budget, a 24-bit resolution external ADC would be beneficial for meeting this requirement. Additionally, the convergence of the Kalman filtering algorithm used was found to be highly dependent on the initial estimate, and an optimal initial point would be 70% of the voltage produced by the bird when it lands. Since this voltage is unknown, it poses a challenge for the system. However, the power effectiveness requirement was met by driving the system into deep sleep mode and selecting a low-power micro-controller.

Chapter 5

Power Design

5.1 Introduction

The power sub-system plays a crucial role in the design process, as it is the backbone of modern civilisation.Ben Murphy uses a scale to collect data on the weights of the fork-tailed Drongo birds, this data helps to identify the effect that the climate has on birds, therefore the continuous operation of the device requires a reliable power subsystem that is able to withstand harsh conditions.

This chapter details the design process of the power module. It is responsible for supplying power to the scale sub-system and the data acquisition and retrieval sub-system. The current system runs on batteries that are manually replaced. The new design mitigates this with a combination of batteries, solar panels and power management systems.

5.2 Requirements

5.2.1 User Requirements:

An interview with Ben Murphy was held towards the beginning of March. From that interview, the following requirements were provided:

Label	**User Requirement**
UR01	It must be cost effective
UR02	The power system must have a long lifespan
UR03	It must be able to show battery percentage
UR04	It must be able to withstand the harsh conditions

Table 5.1: User Requirements

5.2.2 System Functional Requirements

From the user requirements, further requirements can be made in terms of the physical power subsystem.

5.3 Specifications

From the above User Requirements and System Functional Requirements, specifications are able to be drawn.

Label	System Functional Requirement
SFR01	In order to meet UR01 cheap components must be selected within reasonable ranges
SFR02	In order to meet UR02 solar panels and rechargeable batteries must be implemented
SFR03	In order to meet UR04, SFR01 must be taken into consideration when choosing components for harsh environments

Table 5.2: Functional System Requirements

Label	Specifications
SP01	Must supply a regulated 5V to the ESP32
SP02	Must supply a +-5V to Instrumentation amplifier.
SP03	Batteries must use solar energy to recharge and store energy.
SP04	UVLO cuts voltage off below 6.4V
SP05	Voltage range of 6V - 16V solar power regulated down to 7.2V for battery charging
SP06	Over-voltage protection circuit cuts off when voltage after the linear voltage regulator goes higher than 8.2V

Table 5.3: Specifications

5.4 Traceability Matrix

5.5 System Overview

The above figure shows the initial power supply design. The Input Solar power is a varying voltage determined by the intensity of the sun. There are two different supply voltages, one being the regulated solar input, the other being the rechargeable batteries. When the batteries voltage goes below the threshold, the UVLO will turn on and the relay then switches to the regulated solar power. The relay will then switch to the batteries when it has charged up and the solar power has decreased due to various reasons. The output of the relay then goes into Protection circuitry such as a Over Voltage protection,Reverse Polarity and Short-Circuit Protection. The output then goes into a boost converter that steps the voltage up to 10V. Using a voltage divider circuit, +-5V is created that goes into the Scale sub-module. The +5V will go to the Micro-Controller, which has an on-board Voltage Regulator, and RFID reader.

5.6 Initial Design Considerations and Choices

The following section will dive into the different component and configuration considerations and choices used to determine the different protection circuitry and power requirements needed for the different sub-modules.

5.6.1 Power Source Requirements

Ben works in the Kalahari Desert, therefore the power system needs to withstand wide temperature ranges and harsh conditions. Since the Kalahari experiences direct sunlight for long hours, solar energy is deemed to be the most efficient. The current power system runs on replaceable batteries. Using

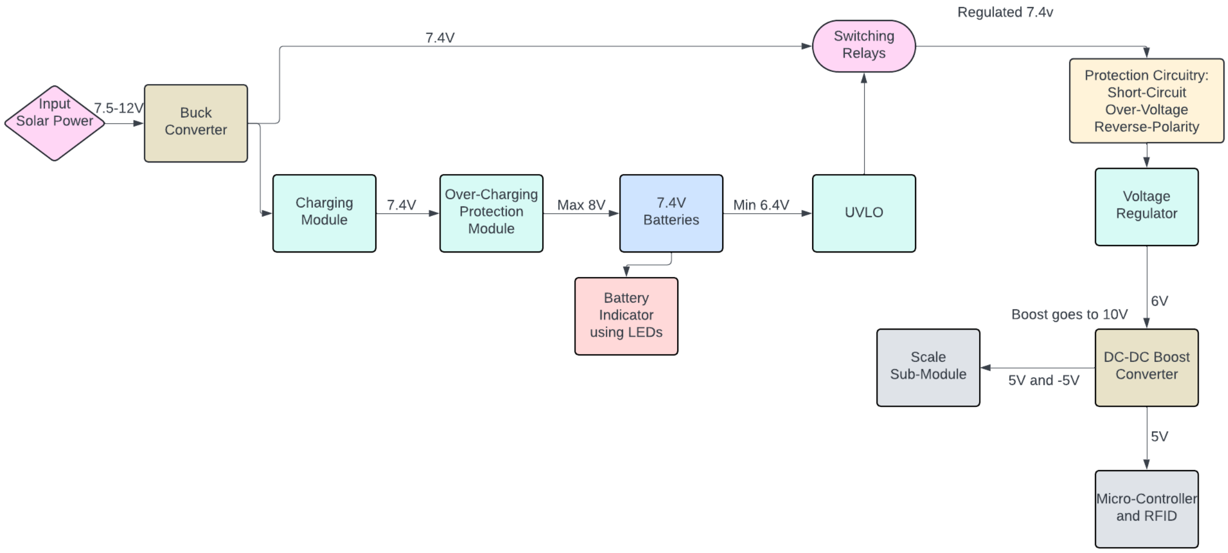


Figure 5.1: Sub-system Overview

Solar energy and rechargeable batteries mitigates the process of replacing the batteries.

ESP32 Micro-Controller

The ESP32 Micro-Controller has 3 different options in order to power on:

Options:	Reasoning:
3.3V	Initially, the 3.3V pin was chosen, however since the RFID reader uses 5V, it was deemed inefficient to use this 3.3V pin. Hence why it was not chosen.
Micro-USB	This option was not viable due to the power module being placed within the scale, it would be hard to reach the point of insertion. Hence why it was not chosen.
5V	Since the RFID reader uses 5V, it was deemed efficient to use this pin as there would not be added components to step-up the voltage. Hence this pin was chosen.

Table 5.4: Options to power on the ESP32

Scale Sub-module

The scale sub-module uses an instrumentation amplifier to amplify small voltages in the presence of high noise levels. This is what will be used to determine different weights at different voltages. As such, it needs +5V and -5V in order to operate.

Since voltage is determine between a reference point and ground, a secondary ground is created. 10V will be supplied, using a Dc to Dc Boost converter. A voltage divider circuit will be used to create +5V and ground, the +5V will now be the new 'ground' and ground now becomes -5V.

	Alkaline Batteries	Li-ion batteries
Advantage	Easily Acquired	Stable power supply
Disadvantage	Does not provide a constant voltage	Has short Life span

Table 5.5: Caption

Sub-Module	Mode	Min Current	Typical Current Drawn	Max Current
ESP32 Dev Board	Active	95mA	120mA	240mA
	Sleep	150 µA	0.1mA	0.15mA
Instrumentation Amplifier	N/A	0.8mA	1mA	1.3mA
RFID reader	N/A	13mA	19mA	26mA

Table 5.6: Current Consumption of the different sub-modules

5.6.2 Battery and Solar Panel

The battery is the first step in navigating the design process. In order to determine the optimal type of battery, the power requirements for the different subsystems were analysed. The battery would need to supply a constant power supply to the ESP32 Development board and instrumentation amplifier, as this allows it to operate with maximum efficiency. Various batteries were taken into consideration, these being:

The ESP32 and instrumentation amplifier needs a constant voltage supply in order to operate efficiently.From Table 5.5, it can be seen that the Lithium-Ion battery is better suited to the power requirements of the project. In order to mitigate the drawback of a short lifespan, a solar panel will be used to supply the circuitry when the battery has been discharged.

From Table 5.5, it can be seen that two Lithium-Ion 18650 3.7V batteries in series is the economical and energy efficient choice with dimensions of 65x18x18mm. The battery is rated at 8800mAh from the analysis below:

From Table 5.6, a power consumption can be calculated to determine the Capacity expected from the batteries. The capacity of the batteries will have to be at least:

The total power required from the battery is 4.583W. The energy required is therefore:

$$\text{The watt-hours is thus Total Power x Operating time.}$$
$$\text{Energy} = 3.583\text{W x 12 hours} = 55 \text{ Watt-hours(Wh)}$$

With the usage of the 7.4V chosen, the Capacity required is then calculated as follows:

$$\text{Capacity(mAh)} = \frac{55}{7.4} = 7432\text{mAh}$$

The solar panels will need to be compact for ease of transportation and rated for the batteries voltages. However, there is a voltage drop when a load is connected, therefore the solar panel with a higher voltage rating is chosen.A 9V,4.2W solar panel is thus the efficient choice with a cost of R124. The solar panel operates in a temperature range up to 85°C with dimensions of 165mm x 165 mm x 3mm

	Linear Voltage Regulator	Buck Regulator
Regulation	Suited for small voltage range	Suited for wide range of voltage
Cost	Low Cost	Higher cost
Complexity	Simple	Complex at higher voltage ranges
Output Power	Lower power due to heat dissipation	Efficient in output power

Table 5.7: Caption

Implementation	Advantage	Disadvantage
Forward-biased Diode	cheap and easy to implement	Constant Voltage Drop and shortens equipment operation time
MOSFET	Minimal Voltage drop	Not as cheap as a simple diode

Table 5.8: Diode vs MOSFET Implementation

5.6.3 Voltage Regulator

Voltage regulators regulate voltages when there is a variation in the input voltage. Thus allowing stable voltages to be supplied to various sub-systems. However,there is a drawback of power loss from different regulators.

5.6.4 Power supply switching

To maximise solar energy, a relay switch will be implemented to allow for a regulated solar power to supply the circuitry. This will happen when the battery voltage has discharged, allowing for it to charge to full capacity before delivering power to the circuit again.This will lengthen the battery life whilst taking advantage of renewable energy. Since the combined nominal voltage of the batteries will be rated at 7.4V, a 6.5V relay switch will be implemented.

5.6.5 Reverse Voltage Protection

Reverse Voltage Protection circuitry protection protects the circuit from negative voltages when the reverse voltage has been applied to the input or output terminals. Two different designs are used in implementing this. These are shown in Table 5.8 below.

Table 5.8 shows that a MOSFET implementation is ideal due to its minimal voltage drop.

5.6.6 Charging and Discharging

The HKD 2S LITH BATT CHarge Charging module was chosen

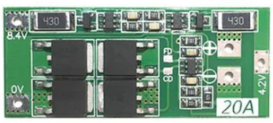


Figure 5.2: Enter Caption

Component	Equation	Value
Inductor	$L =$	100μH
Capacitor		150μF

Table 5.9: Caption

5.6.7 Boost Converter

The boost converter uses inductors,diodes,capacitors and a switching mechanisms in order to step voltages up. In order to implement a boost converter, a PWM signal is needed to provide the switching mechanism its on and off time. A 555 timer IC chip, in Astable mode, is used to generate the the oscillating signal. The output voltage depends on the duty cycle, The output voltage needed is 10V whilst the input voltage is 6V. Since the input and output voltage is known, we are able to make the appropriate calculations for the 555 timer and Boost converter:

$$D = 1 - \frac{6}{10} = 0.4 \text{From the Boost equation}$$
$$D = \frac{R1+R2}{R1+2*R2} \text{ with Toff} = 0.7\text{*R2*C from the 555timer equations}$$
$$\text{This leads to } R1 = 1k\Omega\text{, R2} = 30\text{k } \Omega \text{ and C=470pF}$$

Measuring the output of the 555 timer, the square waveform with the Duty cycle of 0.4 is seen. The switching mechanism used is a N-Channel MOSFET. For the above application,the IRFZ44N N-channel MOSFET was selected, due to its low-on state resistance[1] allowing for minimal power losses and operating temperature range[2]. The Boost converter uses an Inductor,Capacitor and Diode in order to scale the voltage higher. The Inductor and Capacitor values were calculated as seen in Table 5.9.

5.7 Final Design Choices and Considerations

Buck did not work Include Battery Level

First iteration included linear voltage regulator

5.7.1 Problems Occurring

Buck Converter never work with feedback did not work, Charging and discharging did not work

5.8 Final Design

This includes the buck regulator Include veroboard - If I have time Include if anything went wrong -Why I chose this and why I change to PCB, etc. Include boost IC

5.9 Subsequent Results

Include the pictures here

[1] Rds(on) = 0.175Ω
[2] Max Operating temperature of 175°C

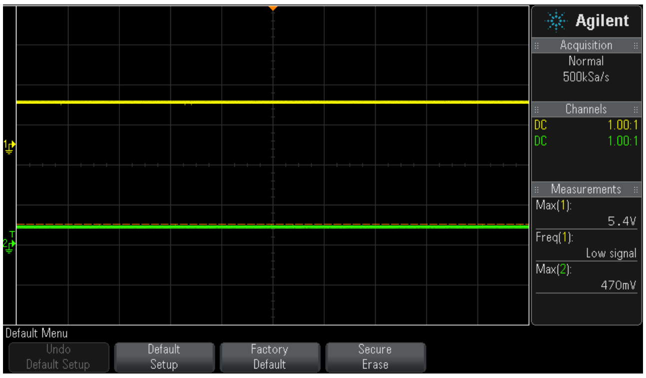


Figure 5.3: Under-Voltage Lockout

5.10 Acceptance Test Procedure

Unit Acceptance Test

- **ATP01- Under Voltage Lockout:** Upon connecting a 7.4V input to the circuit, measure the output by connecting a multi meter. Decrease the input voltage in increments such that the output voltage will decrease rapidly to 0V when the input voltage is below the threshold.
- **ATP02- Over-Voltage Protection:** Connect a 7.4V to the input of the circuit. Measure the output with multi meters as the input voltage is increased. Once the input voltage reaches a certain threshold of 8.2V, the output voltage is 0V.
- **ATP03- Reverse-Polarity:** Reverse-polarity is tested by swapping the battery terminals around, the positive terminal is now connected to the ground of the circuit and negative terminal to the positive. Measuring the output voltage, with multi meters, will give 0V.
- **ATP04- Boost Converter:** Connect the boost converter to an input voltage, the measured output voltage will be 10V needed for th e+5V and -5V.
- **ATP05- Voltage Regulation:** Upon connecting the battery, the output of the circuit will be a constant,regulated voltage.
- **ATP06- Charging and Discharging:** In order to test battery charging , a battery will be connected to the output of the charging circuit and the amount of voltage going to the battery will be measured.
- **ATP06- Overall Testing** Apply a varying voltage input to mimic a solar panel into the circuit. Measuring the output voltage with multi meters will give 10V from the output of the boost converter to ground, +5V from the boost to the middle of the voltage divider circuit and -5V from the voltage divider circuit to ground.
- **ATP07- Harsh Conditions** The circuit as a whole will be subjected to harsh conditions such as heat and wind. Using a hairdryer, heat is applied to the circuit. In testing wind, the circuit is shaken back and forth.

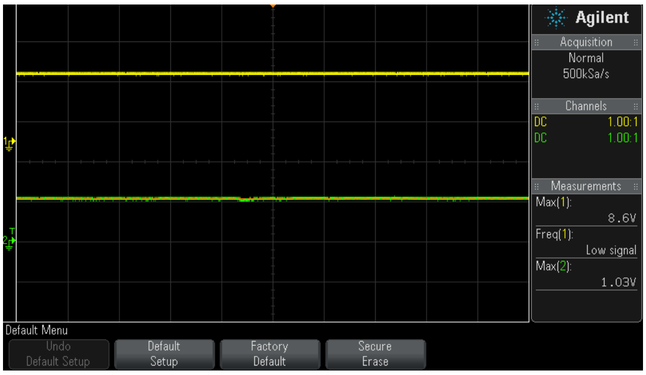


Figure 5.4: Over-Voltage

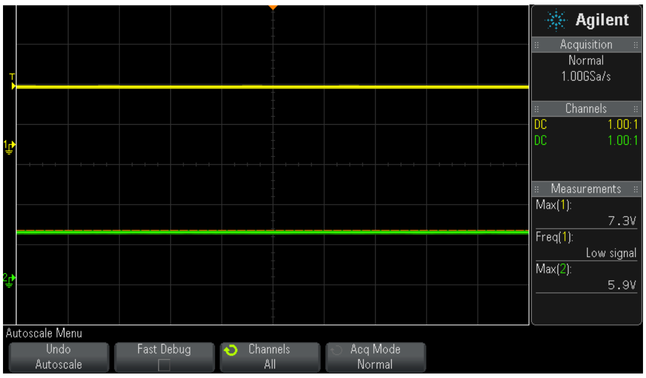


Figure 5.5: Linear Voltage Regulator

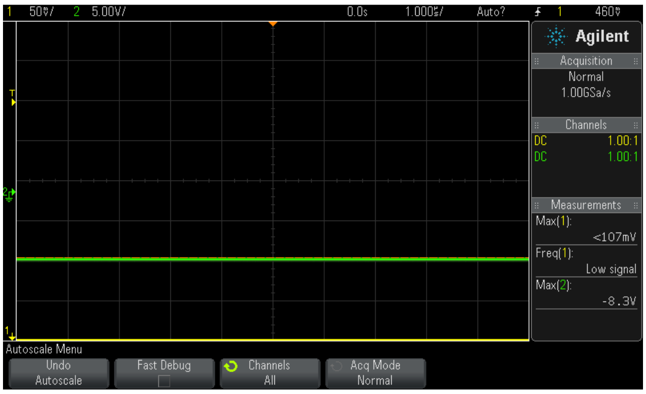


Figure 5.6: Reverse Polarity

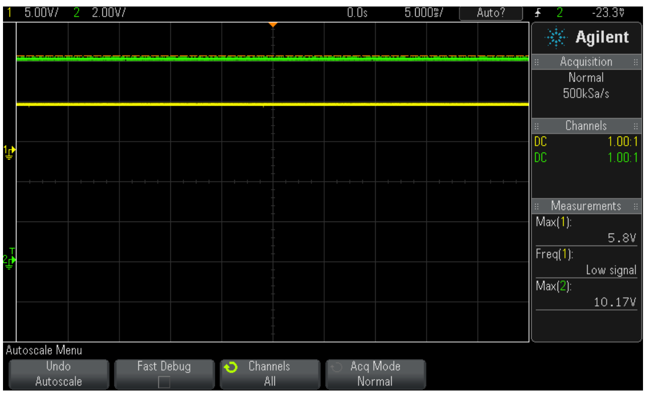


Figure 5.7: Boost Converter

5.11 ATP Result

5.11.1 Integrated Testing

Test everything together

5.12 Recommendations

PCB iteration

Chapter 6

User Interface

6.1 Introduction

An appraisal of the problem analysis indicates that a significant amount of time was spent by the user in reviewing data acquired in the field. Previously, the user would read mass values from the scale using binoculars from a distance so that the subject could land on the device. This method of recording and viewing mass was inefficient: requiring a significant amount of time to setup data capture and to analyse the data.

This chapter details the design of the user interface and wireless communication which addressed the shortcomings of the method described above. The chapter begins with a recount of the user requirements in the context of viewing the weight data. An analysis of the requirements follows, which is then used to support the formulation of design choices in the next section. The subsystem design section describes the mobile application and wireless communication it establishes with the microcontroller. The chapter ends with a description of tests of the mobile application and the wireless network. A conclusion is drawn on the results from the acceptance tests of the subsystem.

6.2 Sub-system Requirements

6.2.1 User Requirements

The user interface was required to be a mobile application to interact with the scale. The following user requirements were given in the initial stages of the design process:

- The user interface must run on a mobile device
- View the mass of the bird weighed by the scale
- Associate a mass with a bird identity

6.2.2 Functional Requirements

The following functional requirements are expanded from the above user requirements:

- The mobile application should be compatible with commonly used mobile devices, e.g. smartphones, tablets, laptops, and commonly used operating systems, eg. Android, Linux, Windows.
- The interface must be able to send and receive packets of data from the microcontroller subsystem.

- The interface must display weight measurements in real-time from downloaded data, allowing users to monitor ongoing data capturing events promptly.
- The application must provide features for associating measured mass values with specific bird identities, either through manual input or automated identification.
- The subsystem must be able to interact with the scale device by sending operational commands such as resetting the scale between weighing subjects.
- The application should be able to store collected data for future access and viewing while maintaining the state of the data.

6.2.3 Non-functional Requirements

The following requirements serve as a basis that can be used to benchmark the operation of the system:

- The user interface should be intuitive and easy to navigate and require no training for users to operate effectively.
- The application must ensure consistent performance and present data accurately under different conditions.
- Contents of data views must be streamlined and allow for searching and data manipulation processes to minimize the time and effort required for data analysis.
- The mobile application should be built at a low cost.

6.2.4 Requirements Analysis

The user interface subsystem must operate on a mobile device to provide flexibility for accessing the interface in remote areas where avian targets are located. It requires seamless integration with a wireless communication protocol to transmit data between the mobile application and the scale device. This protocol should facilitate the transmission of various types of data associated with weight and scale operation.

The mass data captured by the microcontroller subsystem needs to be transferred to the user interface for viewing through the wireless communication protocol. The interface must present data in a clear and easily interpretable manner, without overwhelming the user with unnecessary information. It should prominently display the mass of the weighed subject and label other relevant data, such as collection location and timestamp.

Each data capture event is associated with a specific subject, requiring reliable subject identification to facilitate pattern recognition and data analysis. Therefore, the user interface subsystem must transmit an ID value along with captured information for each event.

The user interface should also provide information about the status of the scale device, including wireless connection status, memory usage, and power status. Additionally, users should be able to access hardware features on the device, such as resetting the scale reading, remotely from the mobile application to avoid disturbing the subject during data capture events.

6.2.5 Design Choices

The following section discusses considerations for meeting the requirements, compares available options and elaborates on the selected solution. The focus of the considerations includes figures of merit related to cost, implementation, ease of testing, reliability and maintenance costs in the design process.

A. Mobile Application Prototyping Tool Selection:

The design cycle of the user interface includes a prototyping stage aimed at visualizing the app interface, user interactions, and user flows. Prototypes can vary in fidelity, ranging from low to high fidelity, and can be either manual or interactive. Mobile application prototyping tools facilitate the creation of high-fidelity prototypes efficiently.

Among mobile app prototyping tools, Figma stands out as a comprehensive solution offering numerous advantages over competitors such as InVision and Adobe XD. Overall, Figma's superior combination of features, collaboration capabilities, pricing plans, and flexibility makes it the preferred choice for prototyping the mobile application. Comparison table 6.1 highlights the design considerations that suggested that Figma was the suitable solution for prototyping.

Table 6.1: Prototyping Tool Feature Comparison Table

Features	**Figma**	**InVision**	**AdobeXD**
User Interface	Auto-layout features for multiple screen sizes	User-friendly with a wide range of templates	Options for creating layout grids
Collaboration	Real-time editing and feedback	Collaboration features with version control	Integrates seamlessly with other Adobe Creative Cloud applications
Compatibility	Multi-platform support	Limited compatibility with certain devices	Limited compatibility with Linux OS devices
Pricing	Free for individuals	Pricing upon request	Pay before usage
Prototyping Features	Advanced interactions and animations	Standard prototyping capabilities	Collection of triggers that facilitate more types of UI designs

B. Mobile Development Framework Selection:

Mobile application development frame works consist of embedded libraries and other tools and resources to facilitate the establishment of software that can be installed on mobile devices such as smartphones, tablets and laptops. Mobile app development frameworks can be classified into:

- *Native*: Provide tools for specific platforms using native languages
- *Cross-Platform*: Use a single codebase to synthesise the same application on multiple platforms
- *Hybrid*: Contain native and web capabilities.

By leveraging Flutter's cross-platform compatibility, native-like performance, customizable UI development capabilities, hot reload feature, and integration capabilities, the decision to use the Flutter framework and its software development kit (SDK) for developing the mobile application for measuring the weight of birds in the wilderness was made with confidence. Its ability to meet the project's requirements while enabling efficient communication with a microcontroller played a pivotal role in the selection process. Figure 6.5 illustrates the features that support the selection of the Flutter SDK for developing the mobile app.

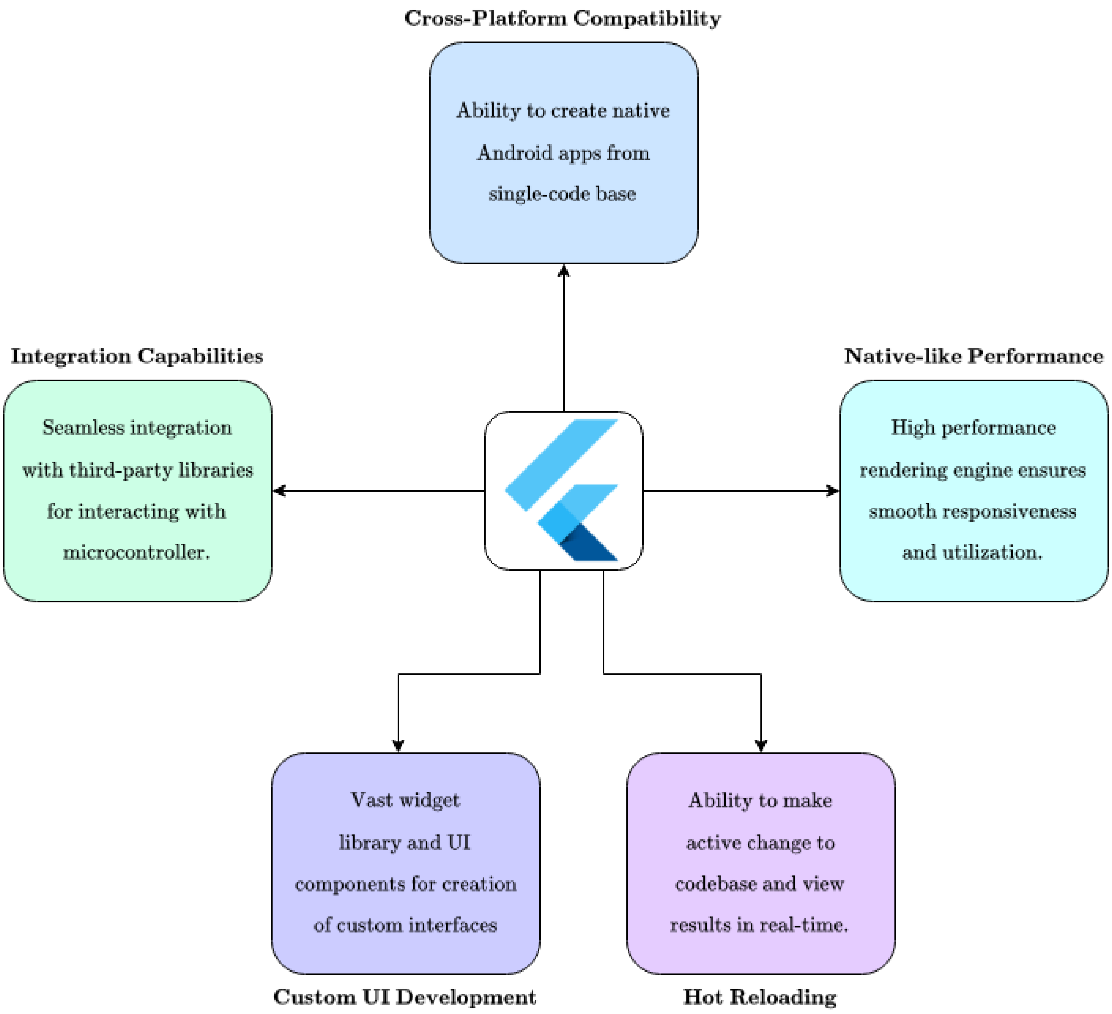


Figure 6.1: Showing features supporting selection of the Flutter SDK for developing the mobile app user interface.

C. Wireless Communication Protocol Design:

The wireless communication protocol for transmitting data between the data subsystem and the mobile application required real-time weight data transfer capabilities, the ability to transmit critical instructions such as zeroing the scale, and the ability to retrieve information about the status of the device such as battery life and the status of the RFID reader. The data subsystem used an ESP32 microcontroller which supports a variety of different wireless communication protocols. The advantages and disadvantages of each protocol were taken into consideration when selecting the most suitable solution from the available technologies as seen in table 6.2.

Table 6.2: Showing list of available communication protocols for ESP32 microcontrollers.

Technology	**Short description**
Bluetooth Low Energy (BLE)	Low-bandwidth capabilities suitable for short-distance transmission of small amounts of data. BLE is optimized for short burst data transmission.
Bluetooth Classic	Suitable for exchanging data over short distances and is highly efficient for continuous data streaming. Uses a significant amount of power compared to BLE.
ESP-NOW	Suitable for short-packet transmission between multiple devices.
Wi-Fi	a client-server communication protocols allowing data exchange between ESP32 boards and other devices using HTTP.
Message Queuing Telemetry Transport (MQTT)	Simple messaging protocol that can be used to exchange data through an MQTT broker that manages message transfers.
LoRa	Implements radio modulation technique that requires Semtech LoRa transceiver chips, allowing for low-bandwidth and long-range communication.
GSM/GPRS/LTE	Requires a modem to facilitate SMS transmissions, phone calls and connections to the internet using a SIM card.

The wireless communication protocol design for transmitting data between the data subsystem and the mobile application involved evaluating the advantage and disadvantages of the wireless communication protocols to ensure efficient, reliable, and real-time data transfer. Each protocol was carefully assessed based on its specific characteristics and suitability for the project's requirements.

BLE was chosen due to its low power consumption, making it ideal for transmitting short bursts of data, such as instructions for zeroing the scale while minimizing energy usage. Additionally, BLE's support for point-to-point communication and broadcast mode aligns well with the need for direct communication between the ESP32 microcontroller and the mobile application, ensuring timely transmission of critical instructions. The ESP32's bluetooth module is compliant with Bluetooth v4.2 BR/DR and Bluetooth LE specifications. A high-speed UART host controller interface layer is capable of transmitting 4 Mbps, which is a suitable for real-time display of information about the status of the device.

For the transmission of weight data in JSON format, Wi-Fi was chosen for its high data transfer rates, extensive range, and reliable performance. The ESP32 Wi-Fi's long-range communication capabilities enable remote monitoring and control of the weight measurement system, typically up to 300 m in open areas, facilitating access to real-time data from a distance without compromising on data integrity or transmission speed. ESP32 supports the 802.11b/g/n standards making it capable of reaching transmission speeds of up to 150 Mbps.

By leveraging BLE for instruction transmission and Wi-Fi for data transmission, the communication protocol design was expected to achieve a balance between energy efficiency, real-time data transfer, and system responsiveness. ESP32s also support a dual-mode that employs Wi-Fi and Bluetooth simultaneously to optimize data transmission for demanding tasks. This approach ensures a stable and user-friendly interaction between the data subsystem and the mobile application, enabling efficient monitoring, control, and analysis of weight data in real-time, even in remote wilderness environments.

Flutter uses the FlutterBluePlus (`flutter_blue_plus`) plugin to manage BLE. Version 1.32.5 of FlutterBluePlus supports a variety of features on all platforms using simple and robust code [33]. Using this package, the user interface can automatically connect to the ESP32 BLE server by selecting the Universal Unique Identifier (UUID) when scanning for devices. To request data from the Wi-Fi server hosted on the microcontroller, the design considered the use of the `http` package which includes a group of high-level functions and classes manage HTTP resources [34]. These packages can be added to the `pubspec.yaml` file as `dependencies`. Importing these packages in the code allows the use of BLE and HTTP handling functions such as the `read()` function for reading the descriptors of a connected device.

Overall, the selection of BLE and Wi-Fi as the primary communication protocols underscores the project's emphasis on energy efficiency, real-time data transfer, and seamless integration with existing network infrastructure to support the collection the weight of bird mass data in the remote areas.

D. Real-time Data Display Optimization:

The user interface aims to display real-time mass data collected by the scale and data subsystem, including status updates for power, scale, and data components, especially crucial in harsh environments. It needs to continuously update displayed information for field use.

Communication between the mobile app and ESP32 microcontroller relies on BLE and Wi-Fi protocols for real-time data display and instruction transmission. BLE optimizes power usage for status updates and instructions, while Wi-Fi accommodates higher-bandwidth mass data transmission. Wi-Fi is capable of transmitting data at a speed of 120 Mbps and a baud rate of 115200 which is suitable for real-time data display. The ESP32 acts as the server access point, providing data to the mobile app client for weight display.

E. Data Structures and Sorting Data:

In field settings, manual input of geographic coordinates supplements GPS for accurate location recording during data collection. Data handling prioritizes reliable storage and retrieval on mobile devices, with maps chosen as the primary data structure for their key-value pairs, aiding interpretation and ensuring data clarity.

Maps offer flexibility in organizing and retrieving information, allowing intuitive searches and tailored data arrangement. The user interface supports sorting and filtering based on user-defined criteria, enhancing usability and facilitating meaningful analysis in ornithological research.

6.3 User Interface Subsystem Design

The following section details the design and specifications of the user interface subsystem. Figma was used to design and prototype the mobile application with the aim of representing an intuitive platform for displaying weight data and interacting with the scale device. An intuitive platform is marked by adherence to design heuristics and principles, as well as navigation patterns that are easy to understand. To expedite the design process, the Figma mockup screens were executed using Flutter's Material3 standards in order to reduce the amount of time it takes to convert design frames into Dart code.

Flutters rendering service uses widgets as the primary layout mechanism. Widgets extend to layout models, images, icons, as well as rows, columns and grids that manage, arrange and constrain visible components. Layouts were created by composing widgets in a widget tree. A Flutter application is typically described within a Scaffold widget. A default banner, background colour and APIs for additional features are automatically provided for Material applications.

The design used Flutter version 3.19.6 which supports the deployment of applications for a variety of hardware architectures and operating systems. The Android SDK platform was of particular interest for the development of the user interface as mentioned in the requirements section above. Android SDK platforms are typically associated with x64, Arm32 and Arm64 hardware architectures. The Android SDK uses virtual machine acceleration to significantly enhance the execution of the platform on a mobile emulator during development.

Development of Flutter applications maintains support for the various integrated development environments (IDE). In this project, Visual Studio Code was used to employ Flutter tools and customize the graphical user interface.

6.3.1 Mobile App Design Specifications

A. High Fidelity Prototype and Navigation Pattern Specifications

The mobile application's overall navigation pattern is represented in the flow chart in figure 6.2. Each arrow represents a navigation pattern specification with labels from SDP-1 to SDP-8. Alerts and pop-ups are illustrated by green blocks. The acceptance test for each navigation pattern specification is that screens should change between states as shown by an arrow. For example, SDP-1 labels the `home-to-scale` navigation pattern specification. The acceptance test for SDP-1 is that a user should be able to navigate from the Home screen to the Data screen. For pop-ups and alerts, the acceptance test is that the pop-up should appear. Users should be able to dismiss pop-ups and alerts.

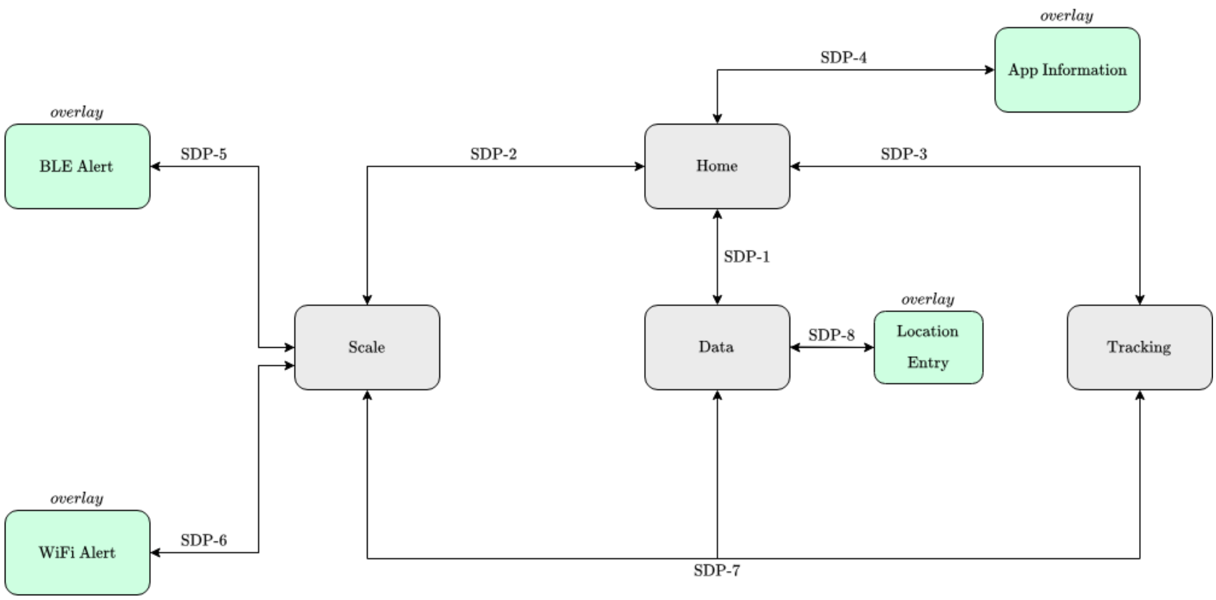


Figure 6.2: Mobile app navigation pattern specification showing each screen as a gray block and the direction of flow with an arrow.

Table 6.3: High Fidelity Prototype Specifications

Label	**Specification Description**	**Acceptance Criteria**
SNP-1	High fidelity prototype with 4 frames: Home, Scale, Data, and Tracking, each sized 393px x 830px. Viewable in a web browser mockup.	Prototype should display when a URL link is clicked, fitting properly in an Android device mockup viewport.
SNP-2	Home screen with app bar, bottom navigation bar, and grouped frames for scale actions, data display, and tracking. Navigable to scale, data, and tracking screens from navigation bar icons.	Clicking scale icon navigates to scale screen; same for data and tracking icons.
SNP-3	Scale screen with app bar, navigation bar, and buttons for power, pair, reset, download, battery, and ping. Pop-up alerts confirm scale actions, dismissible via close button.	Viewable in web browser; navigation to other screens via navigation bar icons; pop-up confirmation for scale actions.
SNP-4	Data screen with app bar, navigation bar, and list of rectangular cards displaying RFID, date, time, location, and mass of subjects.	Accessible on web page; displays subject details; navigation to other screens via bottom navigation bar.
SNP-5	Tracking screen with app bar, navigation bar, and list of map cards displaying location associated with RFID, date, and time.	Visible in Android mockup; navigation to other screens via bottom navigation bar buttons.
SNP-6	A Google Pixel 4 is used as the prototyping device for Android. Google Pixel 4 device has a 5.7-inch screen with a screen size (resolution) of 1080px $\times$ 2280px , a 393px $\times$ 830px viewport, and a CSS Pixel Ratio of 2.75. Screens are be scrollable in the vertical direction where components do not fit inside the viewport.	High fidelity prototype is viewable in a Google Pixel 4 device mockup. Buttons and icons are all placed within the viewport of the mobile phone. Screens should be scrollable in the vertical where components do not fit inside the viewport.

B. Codebase Specifications

Table 6.4 outlines key specifications for the codebase, including project structure, programming language, integration with external packages, modularization, architectural pattern, and version control practices. Each specification is accompanied by its description and acceptance criteria to ensure the codebase meets the desired standards.

C. Mobile App Feature Specifications

Table 6.5 outlines the key features of the mobile application, along with descriptions of each feature's functionality and acceptance criteria for validation.

Table 6.4: Codebase Specifications

Label	**Specification Description**	**Acceptance Criteria**
SCB-1	All code is written in Dart programming language, compatible with Flutter framework. Changes to the codebase are displayed in an Android emulator using Flutter's hot-reload feature.	No errors or warnings are reported during compilation, and all code follows Dart language syntax and best practices.
SCB-2	The codebase uses Flutter's `http` and `flutter_blue_plus` packages to manage HTTP requests, Bluetooth communication, and JSON parsing.	External packages are imported correctly in `pubspec.yaml` file and function as intended, enhancing the application's capabilities without compromising stability.
SCB-3	The codebase follows the MVC (Model-View-Controller) and MVVM (Model-View-ViewModel) architecture.	Each component of the architecture is clearly defined and serves its designated purpose, enhancing code readability and scalability.
SCB-4	Version control using Git is implemented, with regular commits and descriptive commit messages.	Git repository history reflects the progression of development, allowing for easy tracking of changes and collaboration among team members.

Table 6.5: Feature Specifications

Label	**Specification Description**	**Acceptance Criteria**
SFS-1	Weight Measurement	Users can view the weight of measured birds using the mobile application.
SFS-2	Zeroing Functionality	Users can zero the scale for accurate weight measurement.
SFS-3	Real-time Data Update	Weight measurements are updated in real-time on the mobile application interface.
SFS-4	Microcontroller Communication	Facilitates communication between the mobile app and the ESP32 microcontroller.
SFS-5	Data Logging	Records and stores weight measurement data for future reference and analysis.
SFS-6	Notification System	Sends alerts or notifications to users based on predefined thresholds or events.
SFS-7	Data Searching and Sorting	Allows for data searches and allows users to filter data views.
SFS-8	Settings Management	Allows users to customize app settings, such as units of measurement, notification preferences and app themes.
SFS-9	Download Functionality	Enables users to export weight data from the ESP32 microcontroller unit to the mobile app.
SFS-10	User generated information	Allows user to save notes on the mobile application and to add location data manually.

D. Wireless Communication Specifications

Table 6.6 outlines the specifications for wireless communication between the mobile application and the ESP32 microcontroller.

Table 6.6: Wireless Communication Specifications

Label	Specification Description	Acceptance Criteria
SWS-1	BLE is optimized for short bursts data at 4 Mbps to power off the scale, to check the battery status and reset the scale reading.	If the scale is ON, it should change to low power mode when power button is pressed. If the scale is in low power mode, then pressing the power button should turn the scale ON. Battery level is displayed as a percentage when the battery life button is pressed. Scale reading should be reset when the reset button is pressed.
SWS-2	The mobile app uses the `connect()` function to connect to the ESP32. The name of the BLE server is Scalio. The service UUID of the BLE server is `4fafc201-1fb5-459e-8fcc-c5c9c331914b`.	The mobile device should scan and connect to Scalio BLE server when the user clicks on the `Connect` button in a pop-up alert that shows when the `Pair` button is pressed.
SWS-3	The codebase uses the `read()` function to read battery percentage from the microcontroller. The battery level is transmitted in	The battery life is represented by an icon in the user interface when the `Battery` button is pressed in the mobile app.
SWS-4	Wi-Fi facilitates high-speed data transfer (54 Mbps) and a maximum range of 300 m in an open area. Wi-Fi connection has a baud rate of 115200.	A JSON file should be downloaded and displayed within 3 seconds after the `Download` button is pressed on the UI.
SWS-5	App implements error handling for communication failures to maintain data integrity.	App handles BLE disconnections or Wi-Fi interruptions to prevent data loss.
SWS-6	ESP32 supports BLE and Wi-Fi for seamless integration with the app.	ESP32 successfully establishes and maintains both BLE and Wi-Fi connections for effective communication.

6.3.2 Non-Functional Specifications

This following table 6.7 summarizes the non-functional specifications for the mobile application, outlining requirements related to performance, accessibility, security, scalability, compliance, automation, and network optimization. Each specification is accompanied by acceptance criteria to ensure that the application meets the desired quality standards.

6.3.3 Acceptance Test Results

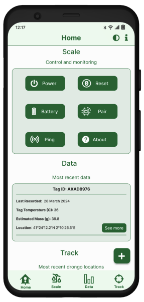
A. High Fidelity Prototypes and Navigation Pattern

Figma was successfully used to create a high fidelity prototype of the user interface. The prototype includes 4 layers, consisting of frames with a width of 393px and a height of 830px. The design goal of the prototype is to allow the user to interact with the interface and provide feedback on the improvements that can be made. The prototype also allows changes to be made to the appearance of the user interface before making changes in the codebase. The `Device Model` plugin was used to generate the Google Pixel 4 Android device mockup shown in figure 6.4.

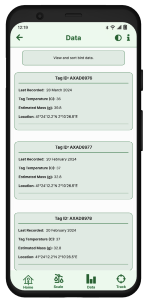
The high fidelity prototype was successfully accessed through the following link:

Table 6.7: Non-Functional Specifications

Label	**Specification Description**	**Acceptance Criteria**
NF-1	The mobile application must be responsive, providing smooth interaction and quick feedback to user inputs.	User interactions, such as button presses and screen transitions, should result in immediate visual feedback without lag or delay.
NF-2	The application should be accessible to users with disabilities, adhering to WCAG guidelines for accessibility.	The app should support screen readers, provide alternative text for images, and ensure sufficient color contrast for readability.
NF-3	The application should be optimized for various network conditions, including low bandwidth and intermittent connectivity.	The app should gracefully handle network disruptions, provide offline functionality where appropriate, and minimize data usage.

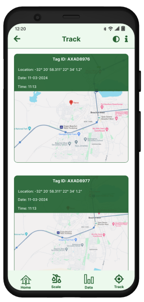


(a) Home screen prototype.



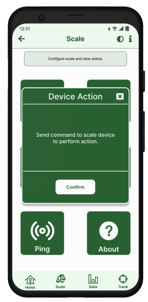
(b) Data screen prototype.

(c) Scale screen prototype.



(d) Tracking screen prototype.

(e) About app pop-up prototype.



(f) Scale action alert prototype.

Figure 6.4: High fidelity prototype illustrations showing the home, data, scale, and tracking screen as well as scale action alerts and pop-ups.

- https://www.figma.com/proto/XcTGHieEBOyBvg6OQvqSSE/drongo?type=design&node-id=101-3&viewport=-177%2C156%2C0.45&t=x1crWKKy4Ygr7c02-0&scaling=scale-down&page-id=0%3A1&starting-point-node-id=101%3A2

This satisfied the design goal for creating a visually interactive prototype, enabling users to provide feedback and iterate on the design before implementation of new features to the user interface. Results of tests based on the acceptance criteria in specifications table 6.3 are summarised in table A.1 in the Appendix. A test outcome of **P** indicates a passed test and **F** indicates a failed outcome.

The high fidelity prototype successfully modelled the navigation patterns illustrated in figure 6.2. When the current screen was the Home screen (a), as shown in figure 6.4, clicking on the data icon in the navigation bar changed the screen to the Scale screen. Similarly, all the navigation pattern specifications shown in figure 6.2 successfully met the acceptance criteria.

B. Codebase Test Outcomes

The results of the code tests are shown in table 6.8. The codebase satisfied the requirement for creating a mobile application for Android devices.

Table 6.8: Codebase Acceptance Tests

Label	**Acceptance Criteria**	**Test Outcome**
SCB-1	No errors or warnings are reported during compilation, and all code follows Dart language syntax and best practices.	**P**
SCB-2	External packages are imported correctly in `pubspec.yaml` file and function as intended, enhancing the application's capabilities without compromising stability.	**P**
SCB-3	Each component of the architecture is clearly defined and serves its designated purpose, enhancing code readability and scalability.	**P**
SCB-4	Git repository history reflects the progression of development, allowing for easy tracking of changes and collaboration among team members.	**P**

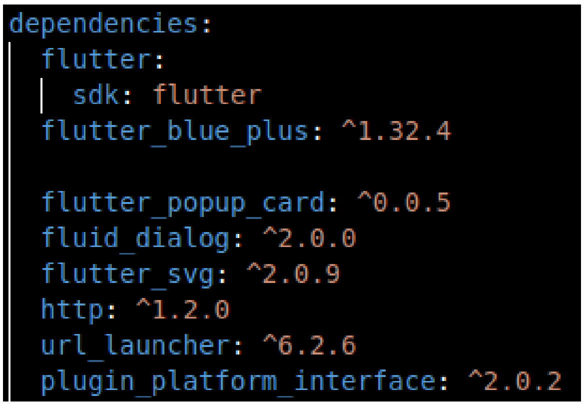
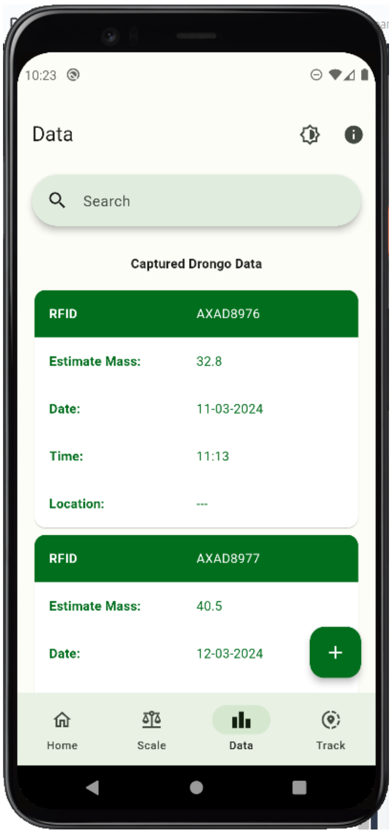


Figure 6.5: Screenshot of the Flutter `pubspec.yaml` dependencies for the mobile application.

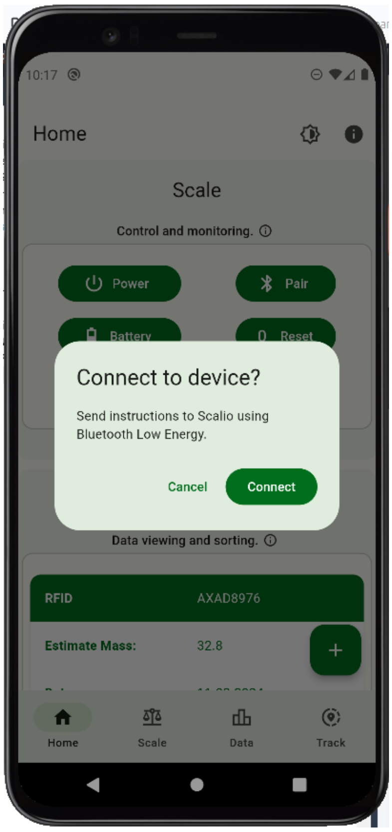
6.3.4 Features and Wireless Communication Test Outcomes

Each feature meets its acceptance criteria and passes the test successfully. These features include viewing bird weight, zeroing the scale, real-time weight updates, communication with the ESP32 microcontroller, data recording and storage, alerts and notifications, data search and filtering, app settings customization, data export from the microcontroller, and note-taking with manual location addition. The unit test outcomes are collected in table A.2 in the Appendix. The following images were taken during the testing of each feature.

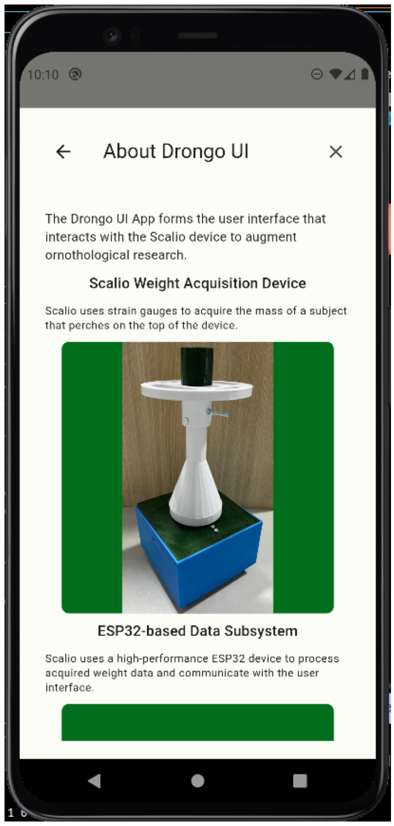
The mobile application successfully utiziled the `flutter_blue_plus` plugin to manage the BLE connection. When the scale was ON, pressing the **Pair** button successfully initialised the BLE connection to the ESP32 BLE server named Scalio.



(a) Data screen with search.



(b) BLE alert for user to connect.



(c) Data screen prototype.

The test of the Wi-Fi connection and the ability to download data from the Wi-Fi access point on the ESP32 showed satisfactory operation of the connection. The acceptance test showed that data was successfully transmitted using the `http` package in Flutter. The system was able to achieve seamless data transmission, capable of real-time data display. A 4.2 kB JSON file was transmitted from the ESP32 using Wi-Fi, without causing the mobile application to slow down or produce errors.

Chapter 7

Bill of Materials

Scale Subsystem	
Item	**Price**
AD620	R39.10
Load cell 1kg	R45.00
LM358	R2.88
IRF5904N	R6.9
Through hole resistors*8	R1.12
Zener diode	R1.04
3D printing filament	R90
PN2907A*2	R2.02
Total	188.06
Power Subsystem	
Item	**Price**
Item 1	Price 1
Item 2	Price 2
Item 3	Price 3
Item 4	Price 4
Item 5	Price 5
Item 6	Price 6
Item 7	Price 7
Item 8	Price 8
Item 9	Price 9
Item 10	Price 10
Microcontroller Subsystem	
Item	**Price**
ESP32	R205
D3132 RTC Module	R52,17
RDM6300 RFID Reader	R73,04
Total	R330.21
GUI Subsystem	
Item	**Price**
Item 1	Price 1
Item 2	Price 2
Item 3	Price 3
Item 4	Price 4
Item 5	Price 5

Table 7.1: Table with 2 columns and 10 rows

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

This project aimed to introduce a solution to provide an accurate and effective way to gather weight data of Fork-Tailed Drongos to aid in avian conservation efforts. The proposed solution discussed in this report is Scalio. Scalio is an automated weighing system specifically designed for Fork tailed Drongos, and to be deployed in the Kalahari.The solution can be deployed in the field for an entire day, and can be battery, or solar powered. Scalio is designed to withstand the harsh conditions of Kalahari, while providing accurate weight measurements, (accuracy of 0.4g) with ID via an RFID scanner. The system incorporates an informative, and easily accessible (via an app) user interface to provide the researcher with all the data.

8.2 Recommendations

Based on the problem statement and the results of this system, the following are the recommendations to improve its perfomance

8.2.1 Power System

The power section was designed around first principles basis.

8.2.2 Scale System

8.2.3 Data System

- Investigate other materials to build the housing out of, for example wood, CNC machining or 3D printing with alternative filaments.
- Consult Ben about the height of the perch off the ground, as well as its shape, and make adjustments if necessary.
- Test the scale with more weights to enhance the accuracy of the regression algorithm used in the micro controller section.

8.2.4 Data System

- To reduce power consumption even more, implement a mechanism to power off the Scale system when there's no acquisition process.
- Use a high resolution ADC (24-bit at least), to achieve desired accuracy.
- Use more data sets for modelling weight-voltage transfer function.

8.2.5 User Interface System

Bibliography

[1] R. Kays, B. Kranstauber, P. Jansen, C. Carbone, M. Rowcliffe, T. Fountain, and S. Tilak, "Camera traps as sensor networks for monitoring animal communities," in *2009 IEEE 34th Conference on Local Computer Networks*. IEEE, 2009, pp. 811–818.

[2] W. Fiedler, "New technologies for monitoring bird migration and behaviour," *Ringing & Migration*, vol. 24, no. 3, pp. 175–179, 2009. [Online]. Available: https://doi.org/10.1080/03078698.2009.9674389

[3] M. J. Boisvert and D. F. Sherry, "A system for the automated recording of feeding behavior and body weight," *Physiology & Behavior*, vol. 71, no. 1, pp. 147–151, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031938400003176

[4] I. J. Patterson, "Tags and other distant-recognition markers for birds," in *Animal Marking*. London: Macmillan Education UK, 1978, pp. 54–62.

[5] D. H. Ellis and C. H. Ellis, "Color marking golden eagles with human hair dyes," *The Journal of Wildlife Management*, vol. 39, no. 2, pp. 445–447, 1975.

[6] S. L. Glowinski, *A bird in the binoculars: Understanding birdwatchers' potential to contribute to sustainable development and biological conservation*. The University of Southern Mississippi, 2013.

[7] C. A. Duberstein, S. Matzner, V. I. Cullinan, D. J. Virden, J. R. Myers, and A. R. Maxwell, *Automated Thermal Image Processing for Detection and Classification of Birds and Bats - FY2012 Annual Report*. Pacific Northwest National Laboratory (U.S.), 2012.

[8] I. A. Adams, "Fork-tailed drongos (dicrurus adsimilis) use different types of mimicked alarm calls in response to different alarm threats," 2014.

[9] S. Agnihotri, P. V. Sundeep, C. S. Seelamantula, and R. Balakrishnan, "Quantifying vocal mimicry in the greater racket-tailed drongo: A comparison of automated methods and human assessment," *PloS one*, vol. 9, no. 3, pp. e89 540–e89 540, 2014.

[10] W. W. Verstraeten, B. Vermeulen, J. Stuckens, S. Lhermitte, D. van der Zande, M. van Ranst, and P. Coppin, "Webcams for bird detection and monitoring: A demonstration study," *Sensors (Basel, Switzerland)*, vol. 10, no. 4, pp. 3480–3503, 2010.

[11] S. A. Gauthreaux Jr and J. W. Livingston, "Monitoring bird migration with a fixed-beam radar and a thermal-imaging camera," *Journal of Field Ornithology*, vol. 77, no. 3, pp. 319–328, 2006.

[12] K. G. Horton, W. G. Shriver, and J. J. Buler, "A comparison of traffic estimates of nocturnal flying animals using radar, thermal imaging, and acoustic recording," *Ecological Applications*, vol. 25, no. 2, pp. 390–401, 2015.

[13] S. A. Gauthreaux Jr and C. G. Belser, "Radar ornithology and biological conservation," *The Auk*, vol. 120, no. 2, pp. 266–277, 2003.

[14] A. Poole and J. Shoukimas, "A scale for weighing birds at habitual perches," *Journal of Field Ornithology*, vol. 53, no. 4, pp. 409–414, 1982. [Online]. Available: http://www.jstor.org/stable/4512767

[15] Y. Yom-Tov, S. Yom-Tov, J. Wright, C. JR Thorne, and R. Du Feu, "Recent changes in body weight and wing length among some british passerine birds," *Oikos*, vol. 112, no. 1, pp. 91–101, 2006.

[16] Seotechwriter, "Different types of weighing scales," Sep 2023. [Online]. Available: https://www.wiltronics.com.au/wiltronics-knowledge-base/types-of-weighing-scales/

[17] "How do you weigh animals at the zoo." [Online]. Available: https://nationalzoo.si.edu/animals/news/how-do-you-weigh-animals-zoo

[18] [Online]. Available: https://hbitech.com/articles/how-digital-scales-work/#:~:text=When%20an%20item%20is%20placed,via%20the%20unitâĂŹs%20circuit%20board.

[19] F. Vézina, D. Charlebois, and D. Thomas, "An automated system for the measurement of mass and identification of birds at perches," *Journal of Field Ornithology*, vol. 72, pp. 211–220, 04 2001.

[20] D. Group, "Heat dissipation design guide for electronics: Colomn: Solutions/products/services: Dnp dai nippon printing." [Online]. Available: https://www.global.dnp/biz/column/detail/10162258_4117.html#:~:text=As%20their%20temperature%20increases%2C%20not,components%20or%20the%20board%20to

[21] D. F. Larios, C. Rodríguez, J. Barbancho, M. Baena, M. Á. Leal, J. Marín, C. León, and J. Bustamante, "An automatic weighting system for wild animals based in an artificial neural network: How to weigh wild animals without causing stress," *Sensors*, vol. 13, no. 3, pp. 2862–2883, 2013.

[22] H. Nguyen, S. J. Maclagan, T. D. Nguyen, T. Nguyen, P. Flemons, K. Andrews, E. G. Ritchie, and D. Phung, "Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring," in *2017 IEEE international conference on data science and advanced Analytics (DSAA)*. IEEE, 2017, pp. 40–49.

[23] B. L. Sullivan, C. L. Wood, M. J. Iliff, R. E. Bonney, D. Fink, and S. Kelling, "ebird: A citizen-based bird observation network in the biological sciences," *Biological conservation*, vol. 142, no. 10, pp. 2282–2292, 2009.

[24] P. Yang, N. Xiong, and J. Ren, "Data security and privacy protection for cloud storage: A survey," *Ieee Access*, vol. 8, pp. 131 723–131 740, 2020.

[25] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, "Cloud storage as the infrastructure of cloud computing," in *2010 International conference on intelligent computing and cognitive informatics*. IEEE, 2010, pp. 380–383.

[26] E. Mackensen, M. Lai, and T. M. Wendt, "Performance analysis of an bluetooth low energy sensor system," in *2012 IEEE 1st International Symposium on Wireless Systems (IDAACS-SWS)*, 2012, pp. 62–66.

[27] E. D. Ayele, N. Meratnia, and P. J. Havinga, "Towards a new opportunistic iot network architecture for wildlife monitoring system," in *2018 9th IFIP international conference on new technologies, mobility and security (NTMS)*. IEEE, 2018, pp. 1–5.

[28] A. Kaur, N. S. Sethi, and H. Singh, "A review on data compression techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 1, pp. 769–773, 2015.

[29] H. Devi Kotha, M. Tummanapally, and V. K. Upadhyay, "Review on lossless compression techniques," in *Journal of physics: conference series*, vol. 1228, no. 1. IOP Publishing, 2019, p. 012007.

[30] D. Tuia, B. Kellenberger, S. Beery, B. R. Costelloe, S. Zuffi, B. Risse, A. Mathis, M. W. Mathis, F. Van Langevelde, T. Burghardt *et al.*, "Perspectives in machine learning for wildlife conservation," *Nature communications*, vol. 13, no. 1, pp. 1–15, 2022.

[31] L. F. Gonzalez, G. A. Montes, E. Puig, S. Johnson, K. Mengersen, and K. J. Gaston, "Unmanned aerial vehicles (uavs) and artificial intelligence revolutionizing wildlife monitoring and conservation," *Sensors*, vol. 16, no. 1, p. 97, 2016.

[32] H. Austerlitz, "Chapter 3 - analog signal conditioning," in *Data Acquisition Techniques Using PCs (Second Edition)*, second edition ed., H. Austerlitz, Ed. San Diego: Academic Press, 2003, pp. 29–50. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780120683772500036

[33] C. Weinberger, "pub.dev," 2024, accessed on April 29, 2024. [Online]. Available: https://pub.dev/packages/flutter_blue_plus#usage

[34] B. Quinland and N. Bosch, "pub.dev," 2022, accessed on April 29, 2024. [Online]. Available: https://pub.dev/packages/http

Appendix A

Appendix

A.1 User Interface Acceptance Test Tables

Label	Acceptance Criteria	Test Outcome
SNP-1	Prototype should display when a URL link is clicked, fitting properly in an Android device mockup viewport.	**P**
SNP-2	Clicking scale icon navigates to scale screen; same for data and tracking icons.	**P**
SNP-3	Viewable in web browser; navigation to other screens via navigation bar icons; pop-up confirmation for scale actions.	**P**
SNP-4	Accessible on web page; displays subject details; navigation to other screens via bottom navigation bar.	**P**
SNP-5	Visible in Android mockup; navigation to other screens via bottom navigation bar buttons.	**P**

Table A.1: High Fidelity Acceptance Test Outcomes

Label	Acceptance Criteria	Test Outcome
SFS-1	Users can view the weight of measured birds using the mobile application.	**P**
SFS-2	Users can zero the scale for accurate weight measurement.	**P**
SFS-3	Weight measurements are updated in real-time on the mobile application interface.	**P**
SFS-4	Facilitates communication between the mobile app and the ESP32 microcontroller.	**P**
SFS-5	Records and stores weight measurement data for future reference and analysis.	**P**
SFS-6	Sends alerts or notifications to users based on predefined thresholds or events.	**P**
SFS-7	Allows for data searches and allows users to filter data views.	**P**
SFS-8	Allows users to customize app settings, such as units of measurement, notification preferences and app themes.	**P**
SFS-9	Enables users to export weight data from the ESP32 microcontroller unit to the mobile app.	**P**
SFS-10	Allows user to save notes on the mobile application and to add location data manually.	**P**

Table A.2: Feature Specifications