A Modernized Automated Penguin Monitoring System

A reliable networked wildlife monitoring device for use in harsh conditions

Prepared by:
Jehan Singh
SNGJEH001
Department of Electrical Engineering
University of Cape Town

Prepared for:
Mr. Justin Pead
Department of Electrical Engineering
University of Cape Town
July 2020

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Electrical and Computer Engineering.

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.

Signature:

Jehan Singh

Date: 2020–07–05

Acknowledgments

To my family, for supporting me through everything. To my good friends and teammates Dillon, Clara and Stefan, without whom my lab work sessions would be twice as productive but a quarter as interesting. To my aunt Shajila for providing a home away from home during my academic career. To my supervisor Justin Pead, for your confidence in trusting undergraduate students with interesting side projects like this and for providing me with all the help I needed when this side project became my honors thesis, and to Dr. Ludynia and Dr. McInnes from the SANCCOB, for deciding that I could work on part of a serious conservation project. I could never have been so confident in my abilities. To Victoria, whose friendship I will dearly miss and whose incredible engineering spirit has always encouraged me to bite of more than I can chew and figure out the details later. To everyone else who I haven't mentioned here: I'm sorry I forgot you. You know how it is.

Abstract

An accurate understanding of the factors affecting endangered wildlife populations is essential for effective conservation efforts. African penguin populations in South Africa are under serious threat, but their colonies are located in remote and inhospitable areas, making research difficult and expensive. Automated monitoring systems for tracking the movement, health and population of the colonies would be a valuable tool for conservation teams. This project details the development and testing of a modernized Automated Penguin Monitoring System to the specifications of researchers at the South African Foundation for the Conservation of Coastal Birds. The device combines the functionality of several prior designs, starting with the typical RFID tag tracking and weight monitoring features of past monitoring systems, as well as the novel features of integrating a cellular remote access connection and video capture of animals. The monitoring system is based around the Raspberry Pi 3B+ embedded Linux computer, and takes advantage of modern software management tools to provide a powerful but reliable and self-repairing embedded system suitable for extended deployments in remote locations. A prototype device was successfully produced, and the report documents preliminary lab tests on the stability and functionality of the device. These tests show that the device could offer access to previously uncollected data when monitoring the African penguin colonies, however field testing is strongly recommended before attempting to deploy a device such as this in the field.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The population of African penguins (pictured in figure 1.1) around South Africa, especially those on St. Croix island, have suffered a significant decrease in population in the past 30 years[1]. The Southern African Foundation for the Conservation of Coastal Birds (SANCCOB) is interested in measuring the movement of African penguins and their weights in order to monitor the general health of the colony, especially focussed on the availability of food. By keeping long-term weight information about individual birds in the colony, it will be possible to measure trends in food abundance and population over time. A sound understanding of the threats facing the birds as well as their environment will help institutes like the SANCCOB make effective decisions to sustain and protect the penguin population.

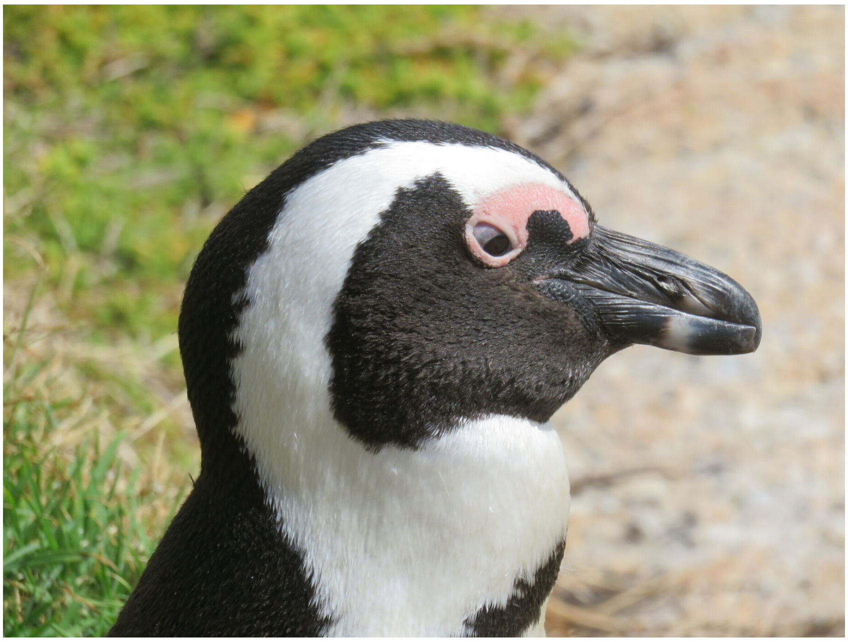


Figure 1.1: An African penguin from the Boulders Beach penguin colony

The sharp decline in African penguin populations is a serious concern for the health of their ecosystems and the biodiversity of marine life. While there are ongoing conservation efforts since 2003 to protect the remaining birds, their populations continue to fall. Serious research efforts into the cause of these declines are needed. Unfortunately, many African penguin populations are located in remote and inaccessible areas. Surveying these populations via traditional techniques is expensive, time-consuming and impractical. In order to effectively gather information on these populations, new methods must be investigated.

Automated monitoring systems are a valuable tool for researchers interested in long term ecological trends in a population. Historically the Automated Penguin Monitoring System (henceforth abbreviated to APMS) has been used to track populations, but due to its age it is affected by limitations that are no longer concerns in the modern day it produces relatively limited information. The aim of this project is to produce a more modern version of the Automated Penguin Monitoring System that incorporates full data capture and video capability in order to provide more information to researchers.

The African penguins concerned in this study are fitted with RFID tracking tags, allowing individuals to be reliably identified over time. By combining this information with a weight-sensing bridge, it will be possible to use individual penguins as sensors to monitor the health of the food supply in their environment.

1.1 Past Work

There have been multiple attempts to install automated monitoring systems at African penguin colonies around South Africa. A simple system that only tracked the movement of penguins was installed in 2018 at Stony Point, which was able to track general population movements, but members of the colony had not yet been fitted with RFID tags and so this system could could not take advantage of RFID tracking to monitor individual birds. This system suffered from complications with weatherproofing, radio communication and power supply, and was eventually destroyed by severe weather. A first draft of the updated system was deployed at St. Croix in 2019 but suffered from poor reliability and was also eventually destroyed by severe weather. Lessons learned from these attempts will be used to ensure that the latest version of the APMS will have a better chance of surviving and providing accurate data for researchers.

1.2 Project Objectives

The objective is to produce a prototype of a reliable monitoring system for use by SANCCOB staff for monitoring penguin populations. The system must record the weight of the birds, scan them for RFID identity tags, and capture video. The APMS is expected to function for extended periods of 3–9 months with minimal intervention. The system must be remotely accessible over a cellular internet connection so that basic maintenance, data retrieval and inspection tasks can be performed without requiring field work.

1.3 Scope and Scope Changes

Originally the scope of the project was:

- Produce 3 functioning prototypes of the APMS for SANCCOB to deploy in the field
- Ensure the prototype provides video, weight and RFID tag information at a suitable level of detail
- Collect detailed weight data from deployed AMPS'es for future research
- Design updated version of weight estimation algorithm to improve on empirical methods used by the existing APMS
- Design tools to process data acquired from field APMS'es

However, field testing of the system could not be performed due to the 2020 pandemic. The scope has been altered to account for this:

- Produce a functioning prototype AMPS for future installation at Stony Point penguin colony
- Ensure the prototype provides video, weight and RFID tag information at a high level of detail
- Provide tools for future work involving collected data
- Design tools to process data acquired from field APMS'es
- Test the APMS prototype to ensure that it will function for an extended field deployment

1.4 Plan of Development

Chapter 2 will cover why monitoring penguin populations may be important, followed by a brief overview of past implementations of automated monitoring systems of various forms. These implementations will be used as the starting point for the design decisions used for the rest of the report.

Equipped with the relevant background information, Chapter 3 will explain the hardware design process behind the development of the system: this starts with the design requirements provided by the SANCCOB, which will provide the basis for many design decisions. The description of the design process will work backwards, starting with a overview of the finalized design and the parts that make it up, before breaking down the reasons why each part was selected and what role it plays in the system.

Once the hardware design is finalized, Chapter 4 will discuss the implementation of software to take advantage of this hardware. Once again it will begin with an overarching view of the final design, followed by a short explanation of the function and purpose of each submodule.

Now that the full design of the system has been completed, Chapter 5 will discuss testing methods. The available test methods for both individual parts of the system and the entire APMS will be described, and the methods used to simulate operation for extended testing will be shown.

A simple portrayal of the results of testing follows in Chapter 6, covering the whether the individual modules were able to perform their functions before moving to the performance of the system as a whole during simulated testing. Each shortcoming of the design will be inspected and, in cases where errors were corrected, this will be noted.

Chapter 7 is a discussion of the results with regards to the design goals of the system. The performances of each subsystem and the system as a whole will be used to assess how well the APMS met its design goals, where there could be room for improvement, and whether there are any egregious failures.

Lastly, Chapter 8 draws conclusions with regards to the suitability of the APMS and recommendations on where future work in this area could be taken will be provided.

Chapter 2

Literature Review

2.1 General background to conservation efforts

The International Union for Conservation of Nature lists the African penguin, *Spheniscus demersus*, as an endangered species on the IUCN Redlist [1]. Their populations are subject to a number of threats, ranging from overfishing of their food resources, to oil spills, to climate-change-induced disruption. According to the IUCN, populations in South Africa have declined from about 70 000 pairs in 1984 to as few as 19 300 pairs in 2015, a loss of almost 73% of the population.

The Southern African Foundation for the Conservation of Coastal Birds (SANCCOB) is the primary body responsible for conservation efforts of sea birds in South Africa.

2.2 Notable past APMS implementations

Automated systems for monitoring penguin populations have been used by a number of groups to gather population data. Over the years these systems have focussed on many different aspects, from collecting weight and movement information to identifying individual birds to monitoring overall population sizes. Several works are discussed below.

2.2.1 1993: First use of an APMS by Kerry et al.

A 1993 study of Adélie penguins on Béchervaise Island in the Antarctic was carried out by Kerry et al. [2]. This study was the earliest use of an automated monitoring system for an extended study on seabird populations. Penguins in the target population were implanted with electronic identification tags, and a combination weigh-bridge and tag scanner was set up in a high-traffic region of the colony. A pair of infrared detection beams were used to determine the direction of birds movement, so that it could be determined whether the birds were moving towards their nests or their feeding grounds. These measurements could be used to assess the availability of food to the penguins, specifically how successful their foraging activities were over time and the impact of food availability on breeding.

Due to the open-air nature of the weigh bridge, the weight of penguins crossing the scale could not simply be recorded directly. This APMS made use of an empirical algorithm to guess the weight of birds as they crossed the bridge, which filtered out the effect of jumping and stepping as they crossed. This was necessary due to the limited storage space available on the device. This empirical algorithm had a consistent systematic error that could be corrected for, but this represents a complication in the system that would not be approached for many years.

The automated monitoring system represented a significant improvement for population monitoring. Researchers could monitoring a large population with greater ease, and there was no need to disrupt the population for weighing, which involved restraining the birds and could produce undesired behaviours in the population. Data from this study was extremely detailed compared to prior work, and was cited in numerous future works on automated studying and electronic tagging of seabird populations.

Kerry et al. were, however, limited by available technology of the era. Limitations of the 1993 APMS include:

- Scale could only be sampled at 8 samples per second, limiting data resolution
- Limited storage available forced all data to be processed as it was collected, preventing future analysis
- Untagged penguins could not be linked to their weights in any way
- Use of an empirical weight estimation algorithm makes it difficult to fully trust weight estimates produced

Nevertheless, this system was highly successful and saw much future use. By 2002, the fourth edition of the APMS was in used by many researchers, as noted by Clarke in a study on feeding habits [3]. The updated system introduced a higher sample rate and an improved estimation algorithm, however details on the actual implementation of the algorithm remain sparse.

Figure 2.1: An African penguin at the Two Oceans Aquarium. Note the arrangement of black markings on the chest.

2.2.2 2010: Computer vision for penguin identification by Sherley et al.

A 2010 study by Sherley et al. [4], studying the African penguin population on Robben Island, developed a computer-vision algorithm that aimed to identify African penguins based on the unique arrangement of markings on their chests, such as those seen in figure 2.1. The aim was to provide an non-invasive method of reliably identifying members of the penguin population that could be performed at scale.

The African Penguin Recognition System capture low framerate (6–8 frames per second) video of birds as they moved. These videos were analyzed with combined analytical computer vision and AI methods to accurately identify chest markings on penguins and assign each penguin a signature. The system was trained against birds with electronic tags, and featured the able to create records for new birds autonomously: this allowed it to track birds that had no electronic tagging at all.

The recognition system was able to, in the long term, provide accurate identification for over 90% of birds for which a favourable capture could be taken. Favourable captures could be taken for 44% of birds crossing the field of view of the camera, with an extremely low rate of false identifications, at 0.01%.

A recognition system such as this could be a valuable addition to future African penguin conservation efforts, as it would allow for large populations to be monitored with greater fidelity and ease than previously possible.

2.2.3 2015: Little penguins in Australia by Salton et al.

A 2015 study of the effect of body mass on breeding in the little penguin was carried out by Salton et al. [5]. This study made use of the updated APMS based on the work of Kerry et al. [2], and made use of a simple average to estimate the weight of birds crossing the weigh-bridge, which was possible as the weigh-bridge was installed with a low mesh tunnel constructed over it. This prevented the penguins from jumping, which produced much more consistent weight measurements. This project targeted a similar goal as previous works, monitoring bird weight as a proxy for food availability and colony health.

Is it notable that despite over 20 years separating the 2015 study from the creation of the APMS, this study was still affected by the limitations of the original system. The APMS had limited storage, forcing it to estimate weights immediately in order to save space. Untagged penguins could not be tracked in any way. In addition, the weight measuring algorithm is only reliable if the penguins can be prevented from jumping, which is not a practical requirement in many other environments where such an installation may be disruptive, expensive or even prevented by regulations.

2.2.4 2015: Improved algorithms by Afanasyev et al.

Afanasyev et al., working with the Macaroni penguin population on Bird Island, developed an analytical weight estimation algorithm for use with an APMS based on the one developed by Kerry et al. in 1993. They identified several flaws with past systems, citing low sampling frequencies, use of arbitrary empirical algorithms, and poor calibration practices. The paper proposed a physically-derived algorithm for deriving the weight of birds crossing a weigh-bridge.

The modified weigh-bridge samples the load cell in the weigh-bridge at 200 samples per second, and records all raw data directly rather than making calculations at measurement time. This allowed for testing of the new estimation algorithm as well as potentially making raw data available to future researchers.

The modified algorithm reduces the impact of the oscillating forces created by a walking penguin by finding the mean force measured during a bird crossing. This force can be treated as a simple estimation of the weight, however a further correction factor can be derived from this weight and the forces measured to provide a much more accurate measure of the weight.

The updated algorithm and storage policies of this study provide solutions to many of the weaknesses of the older monitoring systems, such as:

- Storing raw weight data for more thorough analysis
- Use of a physically derived algorithm improves quality of data
- Higher sample rates improve the quality of data used for estimates

In general, the work of Afanasyev et al. represented a significant improvement in the quality of automated monitoring systems.

2.2.5 2017–2019: SANCCOB installations in South Africa.

Over the course of 2017–2019, the SANCCOB installed two custom-made APMS devices at two African penguin colonies: One at Boulders Beach and one at St. Croix. These devices were developed primarily by Justin Pead of the University of Cape Town, and represent the forerunners of the system developed in this report.

The first APMS, installed in 2018, could only track the weights of penguins and had no means to identify individuals. It made use of a low-power Particle Electron microcontroller. This system made use of an empirical algorithm to estimate penguin weights, and featured simple network connectivity in order to provide real-time updates on the status of the device. This feature was added to simplify maintenance and reduce the need for costly field-work that would otherwise be needed to ensure the device was functioning. This installation provided data for several months before succumbing to issues with the solar power supply and weather damage.

A second version was developed over the course of 2018. On request from SANCCOB, a supplementary feature was to be added: a standalone video capture module. This feature was desired for two reasons: firstly to allow for the possibility of identifying untagged penguins, either manually or using a computer vision tool such as the one seen in Sherley et al. [4]. Secondly, video of penguins could be used to monitor the birds for signs of oil spills, which occasionally threaten the St. Croix colony [6].

The camera module for this project was developed by myself, in the capacity of a student of the University of Cape Town. The development of this system was informally documented in a blog post [7]. The video capture system was deployed as part of the APMS in early 2019, and ran successfully for several days, before suffering a software malfunction that rendered the monitoring system nonfunctional. The device continued to broadcast status messages for several months, until severe weather once again caused damage that resulted in system failure.

2.3 Key points

There have been a variety of approaches to the task of remotely monitoring penguin populations, and obtaining accurate weight information for individual members of the population over time is a powerful tool for assessing access to food. A common goal with automated systems is to reduce interaction with the birds, since capturing birds for weighing or tagging is both stressful to the animals and time consuming for researchers. While there have been advancements in the design of monitoring systems, such as improvements in sample rate and developments of new estimation algorithms, the data gathered remains very similar to that of projects from over 20 years ago, and this represents a point which could be improved.

Projects that make use of video data to help with monitoring exist, but they are presently standalone, only useful for tracking movements of the population, and do not provide any way to make accompanying measurements of weight. These techniques could prove very useful if they can be combined with the data collecting techniques of older monitoring systems, to provide multi-dimensional information about populations.

There is a space in the realm of possible monitoring systems that combines features of several different approaches that has not yet been successfully created. Prior works in South Africa to create this kind of combined monitoring system have failed due to two things: poor rough weather resistance and unreliable software. A system which aims to integrate more data collection techniques into a single system must contend with increasing complexity and increasing demands on resources. Reliability and stability must be a major focus of the design process.

Chapter 3

Design

This section details the functional requirements, provides an overarching view of the operation of the APMS, and explains how design choices were made to support those decisions. Many of these decisions were made for expediency rather than for optimal performance, and justifications for choices will be included. Descriptions of the software implementation of the APMS can be found in the Chapter 4, Implementation.

Due to nesting sites being protected under the Protected Areas bill [8], all equipment must be uninvasive and must not impede the movement of the birds in any way. This rules out, for example, tunnels used to guide birds, which would reduce false-readings from jumping, as seen in the APMS implementation by Salton et al. [5].

3.1 Requirements and goals

The design requirements of the new APMS were set forward by the SANCCOB. These requirements are:

- Must collect weight information about penguins who cross a weigh-bridge at a high resolution
- Must record the detection of electronic tags in penguins who cross a weigh bridge
- Must capture video of penguins who activate the scale or RFID sensors
- Must provide the ability to correlate video footage, RFID scans and weight data
- Must feature a system for remote access to the device for updates and monitoring
- Must integrate with existing monitoring infrastructure at installation sites

- Must be able to operate for extended periods of time without direct intervention

By integrating these features into a single monitoring system, it may be possible to take advantage of several advancements in the field such as the improved weight estimation algorithms proposed by Afanasyev et al. [9] and the computer-vision based identification methods developed by Sherley et al. [4].

3.2 General system architecture

The APMS consists of many interconnecting parts. In order to provide a clear overall image of how the various parts connect, a system diagram of the final design is provided in figure 3.1 overleaf. These parts will be dealt with in detail in the following sections. A detailed wiring guide can be found in Appendix A.

A brief (and colour–coded) description of the components is as follows:

- **Raspberry Pi**: The central computer for the APMS: controls all attached peripherals and sensors.
- **Webcam**: A USB camera which serves as a video capture device for the APMS. Mounted outside the equipment box.
- **RFID scanner**: A large loop antenna and RFID reader placed in the path of the penguins. Reads the serial number of any tagged penguins who pass through.
- **Scale**: An analog load-cell scale placed in the path of the penguins.
- **HX711**: An amplifier and analog to digital converter which makes the measurements of the scale usable by the Raspberry Pi.
- **4G HAT**: A 4G cellular networking device which provides internet connection to the Raspberry Pi.
- **Flash Storage**: A USB flash drive connected to the Raspberry Pi which stores any logs and video captures for easy removal.
- **DC-DC Converter**: Converts the 24V battery power to 5V power for use by the Raspberry Pi and its peripherals.
- **Solar Power Manager**: Charges the batteries using solar power.
- **Solar Panels**: Generate power for the system.
- **24V Batteries**: Large lead-acid batteries provide power when the sun isn't shining.

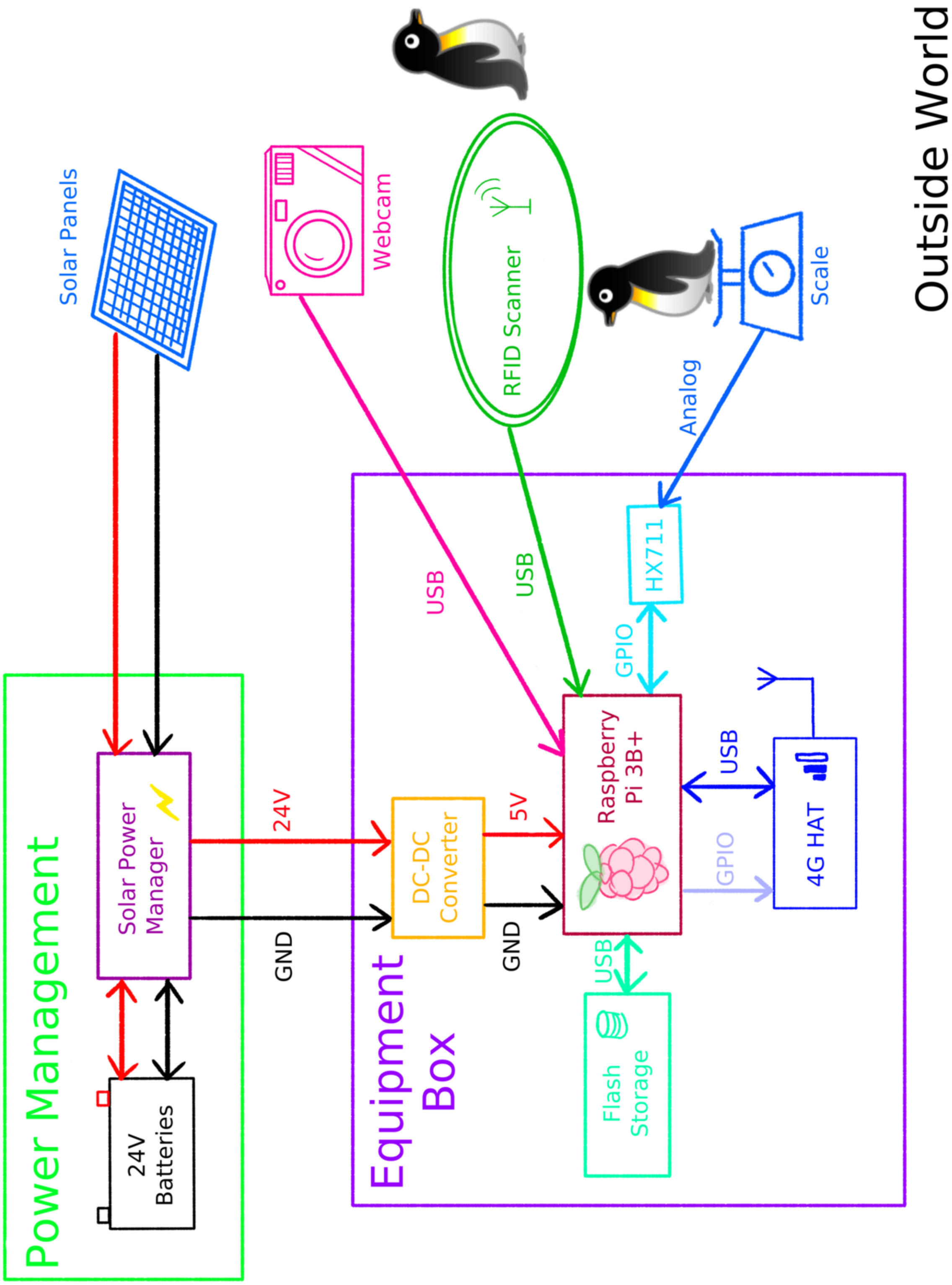


Figure 3.1: A general structural layout of the APMS, detailing the various devices involved and the method by which they are connected to each other. Not to scale.

3.3 New features

The most significant change featured in the new APMS is the need for video information. Capturing video is a computationally and space expensive task which requires much more advanced hardware and software than was available in historical APMS'es, which typically ran on a small, low-power microcontroller with only a few hundred megabytes of storage available.

A request was also made for bidirectional network connectivity to the APMS. The 2018 and 2019 versions of the APMS included a 2G cellular modem for posting statistics to a remote server, but was unidirectional, and provided no means for maintenance or pulling information off the device.

3.3.1 Upgraded computing hardware

The earlier work on the updated APMS [7] took advantage of the power of a Raspberry Pi 3B+ computer running Raspbian Linux for handling video information. The ample memory and processing power provided is suitable for light video processing, and it has a low maximum power usage of about 5W [10], easily within the capabilities of the existing power supply. The Pi includes a hardware encoder for H.264 video onboard, which improves the performance and further reduces the power demands of dealing with video content. Raspberry Pi's are plentiful around the engineering laboratories due to their use as a teaching aid for embedded systems courses.

The Raspberry Pi 3B+ (pictured in figure 3.2) features four USB 2.0 ports, which is sufficient to handle the power and bandwidth requirements peripherals that will be attached.

3.3.2 Video camera

The Raspberry Pi camera was not considered suitable for this task, as it connects to the Raspberry Pi over a short and delicate ribbon cable, making it unsuitable for the severe weather that the APMS must withstand and incapable of being placed at a vantage point away from the main body of the APMS.

The camera chosen is a Logitech C920 HD Pro webcam, pictured in figure 3.3. This model is popular for computer vision tasks and is capable of capturing full HD video under a wide variety of conditions. The webcam connects to the Raspberry Pi over USB, which allows for it to be easily extended for flexibility during installation. A weatherproof enclosure for the webcam will be required to protect it from the elements. The USB webcam communicates natively with the Linux kernel and software making use of the

Figure 3.2: The Raspberry Pi 3B+ single-board computer used for this project.

OpenCV Computer Vision library is used to capture frames from the camera.

The video data is captured using a Python program interfacing with the OpenCV library, which continuously captures frames from the camera and buffers them into memory. Details of this program are discussed in the Implementation chapter.

3.3.3 Cellular networking hardware

The new APMS is required to have remote maintenance and data access ability, so that the need for excursions to the installation site can be reduced. A variety of methods were considered to add cellular internet connectivity to the APMS.

The first consideration was the SIM868 HAT for the Raspberry Pi, produced by Waveshare [11]. This device provides a serial AT connection to a 2G GSM modem for the Raspberry Pi. However, creating a reliable internet connection with only Hayes AT commands is a complicated affair and would require a great deal of work to ensure reliability, and any software written to make use of the network would need to have custom features to communicate with a serial modem.

The chosen solution was the SIM7600E HAT for the Raspberry Pi, also produced by Waveshare [12] (see figure 3.4). This is a 4G LTE modem which provides both a serial

Figure 3.3: Logitech's C920 HD Pro webcam is a popular choice for computer vision tasks.

AT interface for the Raspberry Pi, but also presents as a Qualcomm MSM Interface device over USB. This allows existing software to communicate with the device as though it is any other network device, simplifying design and improving reliability. Software on the APMS can now treat the network connection like any other, which allows for simpler testing and for the use of traditional network programming techniques when making use of the LTE network.

In order to provide a maintenance and monitoring system, a simple reverse SSH tunnel can be created using an existing server. The server in use is currently a development server, but the software can be configured to use any other publicly accessible machine running an SSH server. The reverse SSH tunnel is maintained using a collection of scripts which continuously check the existing connection and restart it if it seems to have failed.

SSH is an extremely versatile protocol, and thanks to the ability to tunnel virtually any kind of network traffic over an SSH connection, a variety of tasks can be performed remotely on the APMS, from monitoring the function of the device to updating the software to copying records to a local machine for analysis.

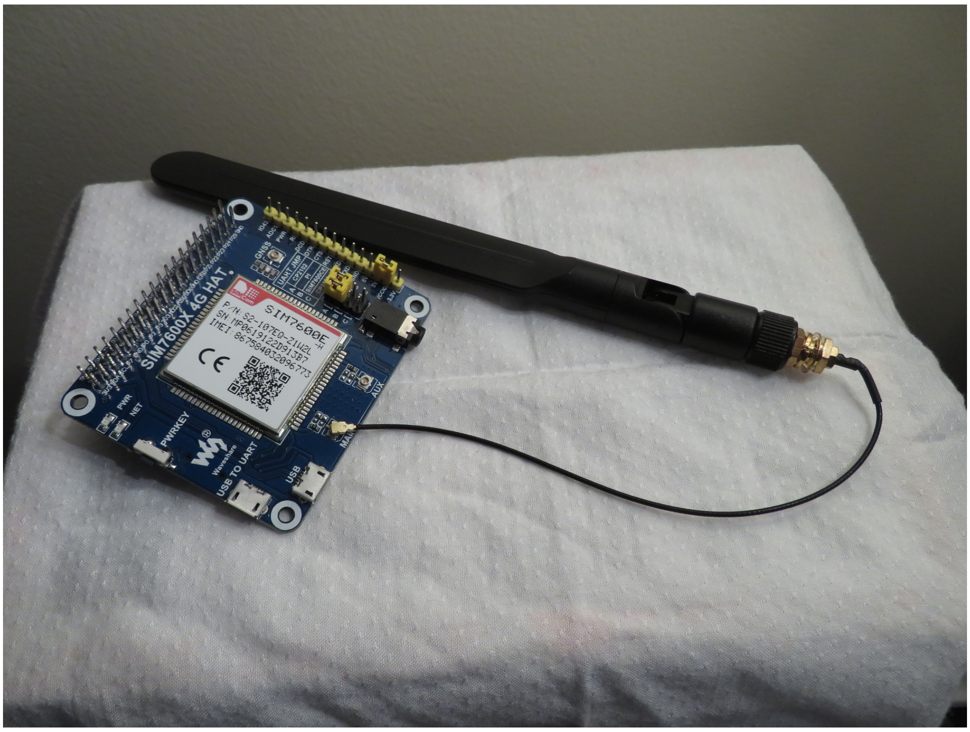


Figure 3.4: The SIM7600E is a 4G capable cellular modem with the ability to present as a QMI network device to a variety of systems.

3.4 Existing hardware

The new APMS is required to fit into the existing support hardware that exists at SANNCOB monitoring sites. These sites have have large 24V marine batteries and solar panels to provide power. The RFID scanner and scale unit in use have already been selected and verified for their safety and functionality on site. The system must integrate with these existing parts. In addition, many pieces of hardware are easily available due to their use in prior attempts, and reusing these parts will reduce the cost of testing. Pre-existing hardware and requirements for the project required some modification in order to integrate with some of the new features.

3.4.1 Weigh-bridge

A load-cell based package scale, the Measuretek PS-105 [13] (see figure 3.5), is currently in use for measuring the weight of the penguins on site. This scale is typically used for warehouse usage, but due to the standard loadcell used inside, it can be easily modified to communicate with an appropriate instrumentation amplifier and analog-digital converter (see figure 3.6 for examples). In order to read the measurements from the scale an HX711 load cell instrumentation amplifier will be used. Tatobari's HX711 Interface Library [14] for the Raspberry Pi is used to interface with the HX711 amplifier in software.

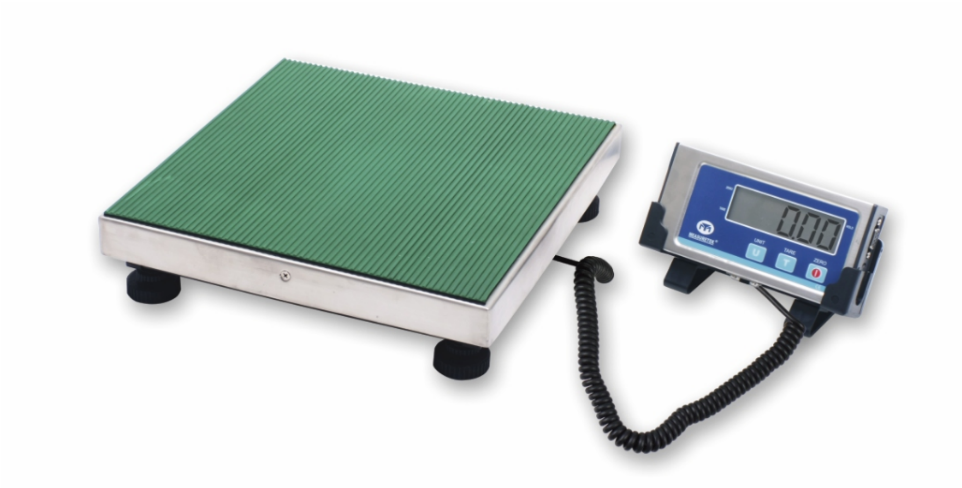


Figure 3.5: The PS-105 parcel scale. The display console must be removed for use with the HX711.

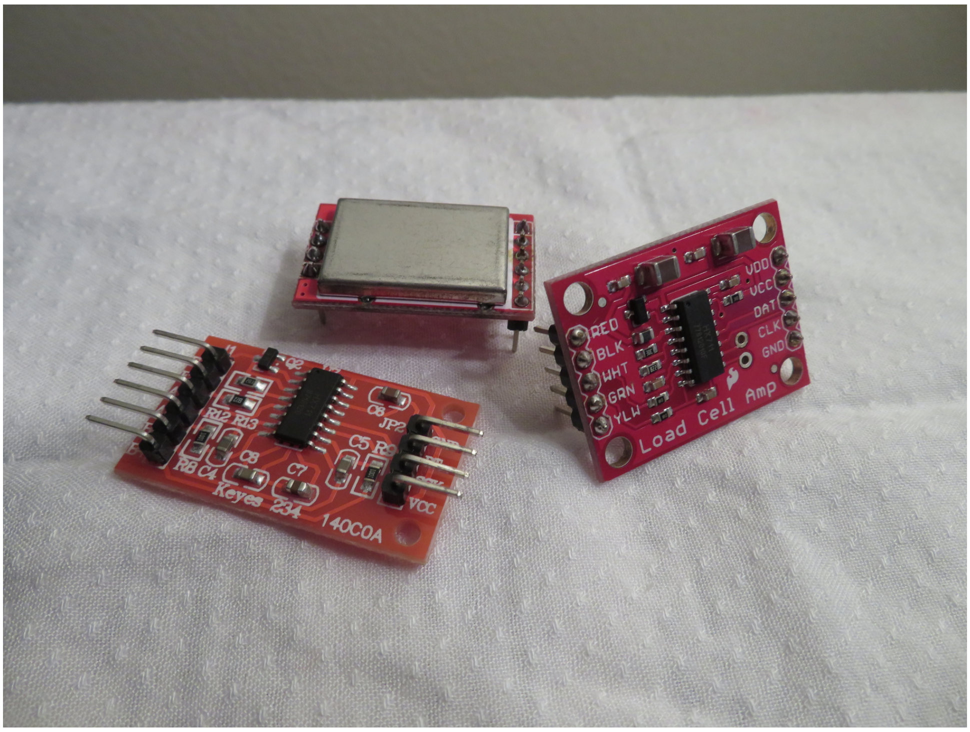


Figure 3.6: Several breakout boards based around the HX711 instrumentation amplifier, some pictured without shielding.

According to Afanasyev et al. [9], an increased sample rate is crucial to enabling accurate predictions of weight from the scale. To this end, the HX711 amplifier will be operated in a high sample rate mode, nominally 80 samples per second.

Research and experimentation performed while developing the precursors to this APMS showed that penguins do not step on the scale if it is left uncovered, and so when deployed on site it should be covered with a brown canvas wrapper to ensure that the birds do not try to avoid it.

3.4.2 RFID scanner

The RFID scanner in use is a ruggedized loop antenna (pictured in figure 3.8) designed by Biomark for underwater use [15]. It has been previously verified for scanning the RFID tags carried by some of the penguins. This antenna is a high-sensitivity model intended for monitoring electronic tags in fish, and is thus suitable for use in harsh saltwater conditions. The antenna attaches to a Biomark IS1001 reader board [16] (seen in figure 3.7), which provides power to the antenna and presents a USB serial interface for the Raspberry Pi to use.

3.4.3 Data storage

Previous iterations of the APMS used a microSD card or dedicated flash chip connected to the central microcontroller for storage. Dedicated flash was used in older systems and was typically very low capacity due to the cost of flash storage in the past. This forced designers to make estimates of penguin weight on site, throwing away the raw data to save space.

Due to the lack of available spare microSD card slots on a Raspberry Pi, this has been replaced with a commodity USB 2.0 flash drive, pictured in figure 3.9. A 64GB or larger flash drive plugged in to one of the USB ports of the Raspberry Pi provides an easy to access and relatively low-cost storage device. Thanks to the ever-decreasing cost of high-capacity flash storage, the new version of the APMS can afford to store both video footage of the animals, and the full sample data of any measurements made by the weight sensor. This will allow for more flexibility when analyzing data collected by the APMS in the future.

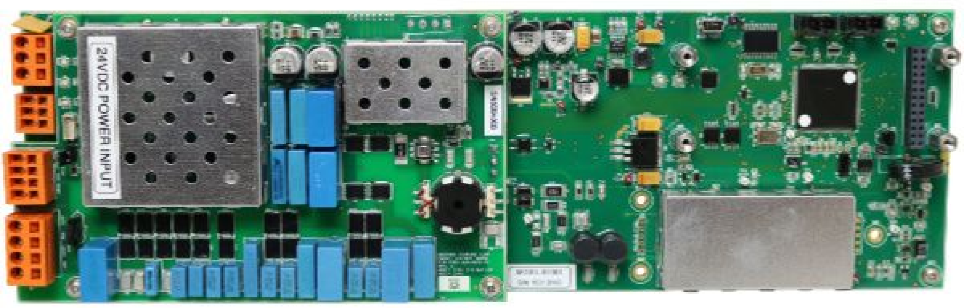


Figure 3.7: The IS1001 reader board provides power and a USB interface to the large RFID antenna

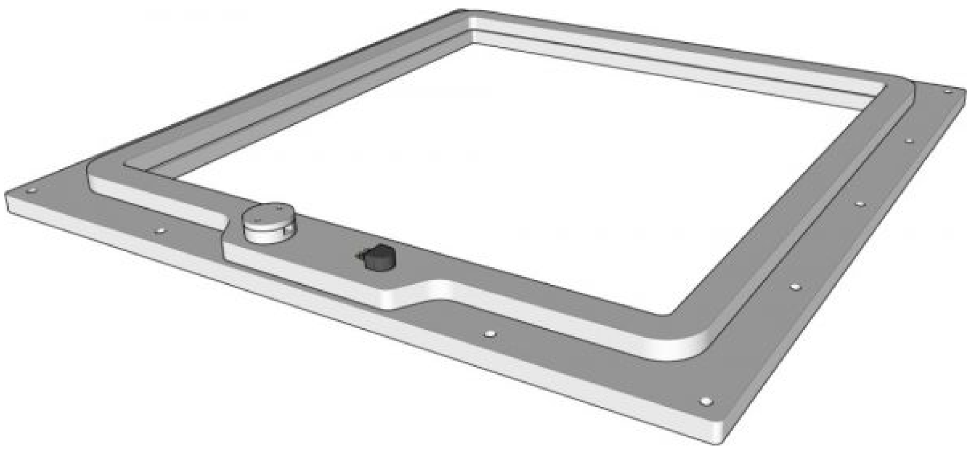


Figure 3.8: The ruggedized antenna used to detect the birds' electronic tags

Figure 3.9: One of the USB flash drives used for testing the APMS.

3.4.4 Power Supply

The only available source of power at many monitoring installations is a 24V solar power system with batteries. These are primarily used for the RFID antenna, which demands a great deal of power. The remaining functions of the APMS must make use of this supply. The power budget of the APMS (including safety factors) is described in table 3.1, with breakdowns for total power consumed and power drawn through the Raspberry Pi 3B+ ports, which has a power limit of 6 watts [17].

The APMS under a full load (i.e. compressing a video, measuring from the scale, capturing video, transmitting to the internet, and writing to disk) can consume up to 14.51 watts.

Device	Maximum Power Budget (W)
Raspberry Pi	7
HX711 amplifier	0.01
USB storage	1.5
USB webcam	3
SIM7600E HAT	3
RFID scanner	N/A Externally Powered
USB port subtotal	4.5
Total	14.51

Table 3.1: Breakdown of power consumption per component

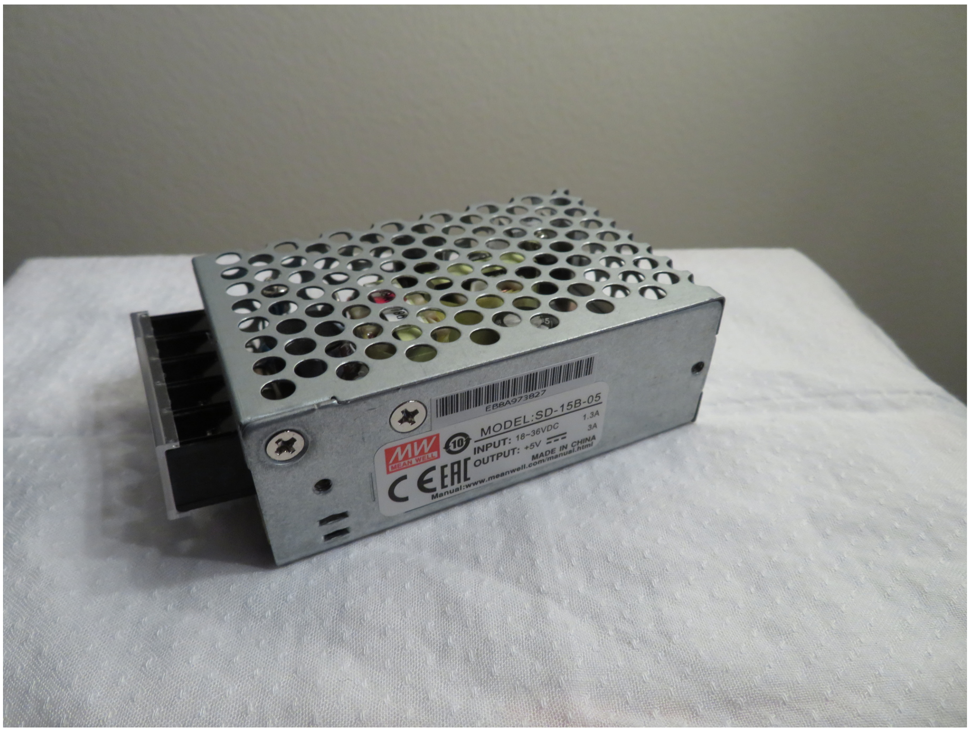


Figure 3.10: The Meanwell SD-15B-05 is able to provide 5V at up to 15W, well suited for the demands of the APMS.

Under testing (further detailed in the Testing section), it was found that it is crucial that the power supply chosen be able to provide this power at a voltage of 5V to avoid failure of the webcam.

The chosen power supply is a Meanwell SD-15B-05 power supply (pictured in figure 3.10), which takes 18–36V DC power from the batteries and converts it to 5V DC at up to 3A, thus having a rated power output of 15W. This is sufficient to provide power to all features of the APMS while staying well within its nominal power and voltage ratings.

Chapter 4

Implementation

This chapter covers the functions of the APMS provided by software and some detail as to how these features were implemented. It will discuss the core data collection modules, the mechanisms used to provide network access to the system, and the various tools used to ensure that the APMS is self-starting and self-repairing. All code discussed here can be found on GitHub at github.com/KaliumPuceon/APMS_code. Any code path references are given in `monospace` from the root directory of this source repository.

4.1 Data collection

The primary function of the APMS is to accurately collect data from the installation site. These functions are handled by a single Python 3 program. Python was chosen for its robustness to errors and quality garbage collection. Python is not a very efficient language compared to some other choices, but offers a high degree of reliability and reduces the chance that a programming error could cause a serious failure of the APMS. Python is a popular language for software development on the Raspberry Pi and there is a sizeable software ecosystem for it.

The following functions are implemented as independent threads of a core applications. This design allows for independent testing and verification of subsystems (discussed more in Chapter 5, Testing), as well as reducing the chance of a catastrophic failure.

The general structure of the various subsystems is summarized in figure 4.1

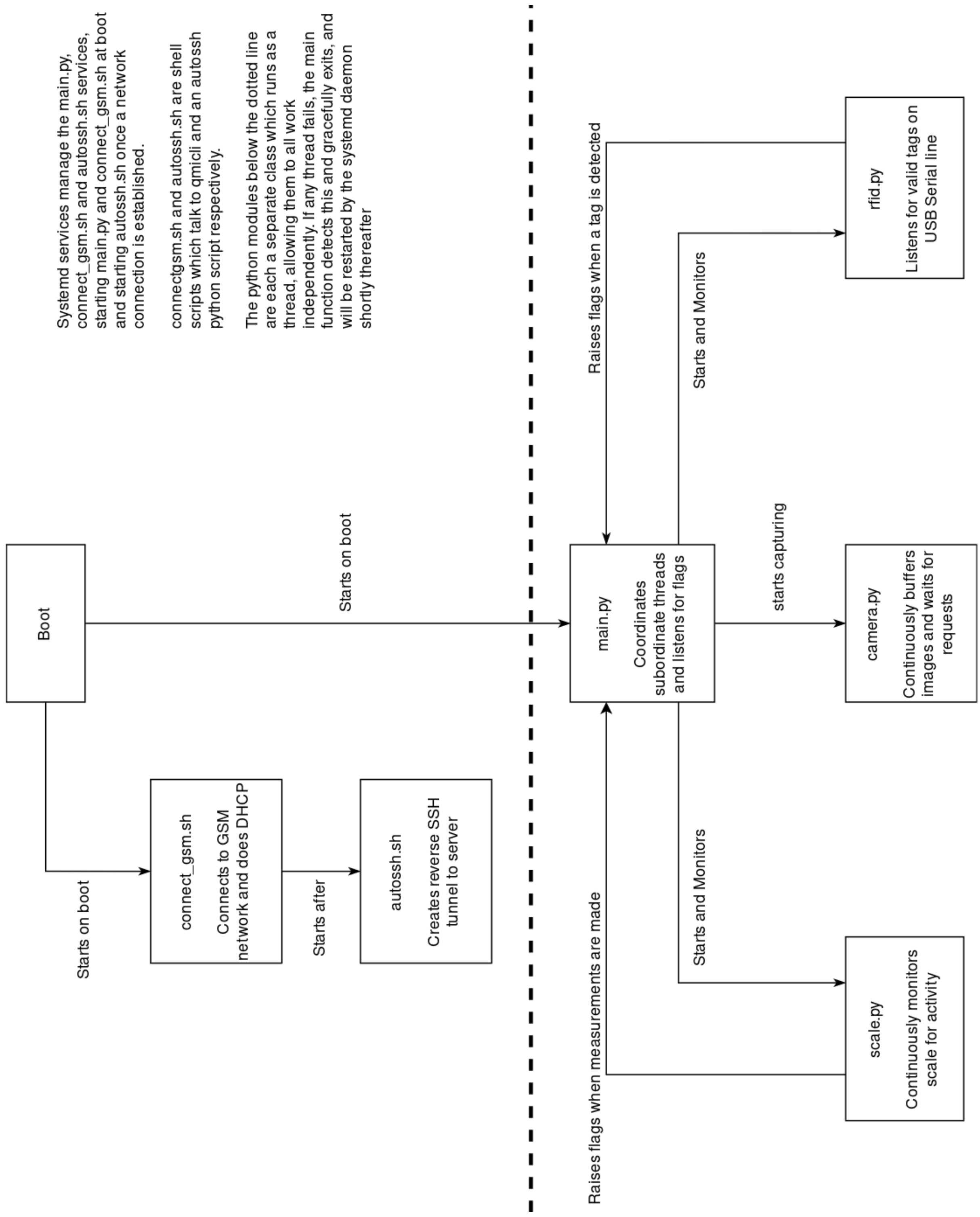


Figure 4.1: A broad overview of the structure, hierarchy and dependencies of the software driving the APMS

4.1.1 Scale measurements

Scale data is the highest priority data the APMS collects. This information can be used to monitor general changes in the population health, and even without any other data can be used to monitor general properties such as the levels of food available over time.

The implementation of this module can be found in `tuxcap_v2/scale.py`. The HX711 amplifier is accessed using the HX711 library for Raspberry Pi [14]. This library allows the program to rapidly sample the scale for data. The scale is sample at 80 samples per second continuously, and this value is compared against a threshold weight. If the weight exceeds the threshold, the current time is noted, and samples are recorded into memory until the weight once again drops below the threshold for a few seconds. This data is then recorded to the flash storage device under the timestamp. A flowchart describing this process can be found in figure 4.2.

When scale data is recorded, a global flag is set indicating that a weight measurement has begun. This flag is used by the coordinating process to control the camera recording.

4.1.2 RFID tag readings

RFID data is the second highest priority data the APMS collects. This information can be correlated against the measurements taken from the scale to track the weights of individual birds over time, which provides much finer information about the changes in food availability on the island, as well as tracking how active individuals are in the population.

The implementation of this module can be found in `tuxcap_v2/rfid.py`. The RFID scanner presents as a standard USB serial device. The RFID subsystem continuously listens for incoming messages. When a message is detected, the current time is recorded and it is logged to a file on the flash storage. A flowchart of this process can be found in figure 4.3.

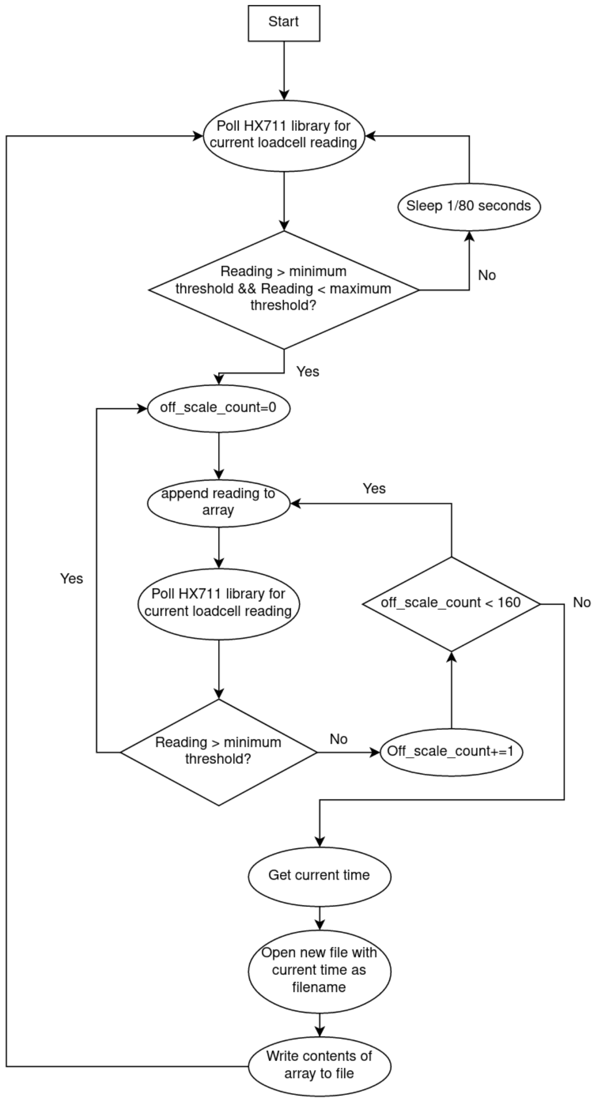


Figure 4.2: Flow chart showing general function of the weight recording subsystem

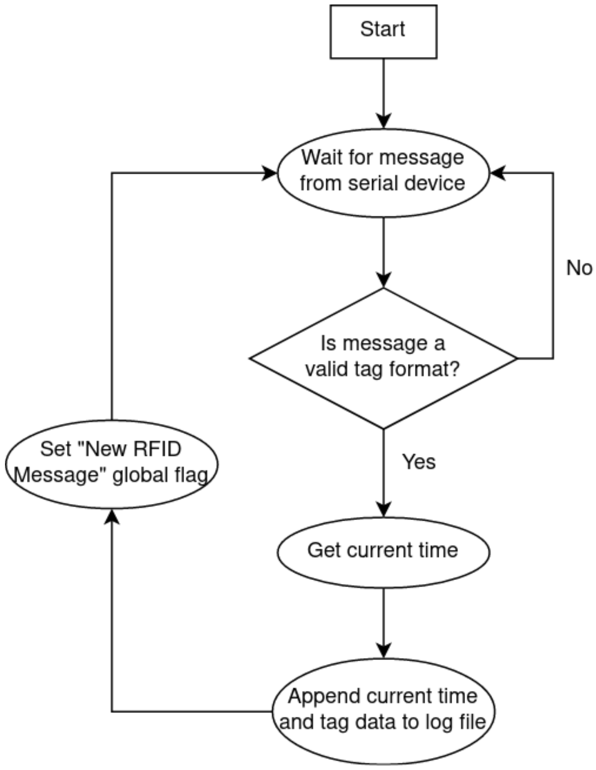


Figure 4.3: Flow chart showing general function of the RFID Tag subsystem

When an RFID tag is detected, a global flag is set indicating that a bird has been scanned. This flag is used by the coordinating process to control the camera recordings.

4.1.3 Camera capture

The video capture is the newest addition to the APMS, and is the lowest priority data collected, as well as being the most resource intensive data. Camera data can be used to add detail to RFID or scale captures that are ambiguous, and could be used to perform qualitative or computer-vision assisted assessments of colony health.

The Camera subsystem is implemented by `tuxcap_v2/camera.py`, and uses the high performance OpenCV library to communicate with the webcam. It continuously records video up to a length configured by the user into a ring-buffer in memory. These recordings are taken at 10 frames per second, in order to reduce the amount of space and computing power needed to handle them. When the camera subsystem receives a signal from the coordinating process, it records for a preconfigured amount of time, and then saves the entire ring buffer to disk as a video. If it is already in the process of creating a video when the signal is received, this signal is ignored. A flowchart of this process can be found in figure 4.4.

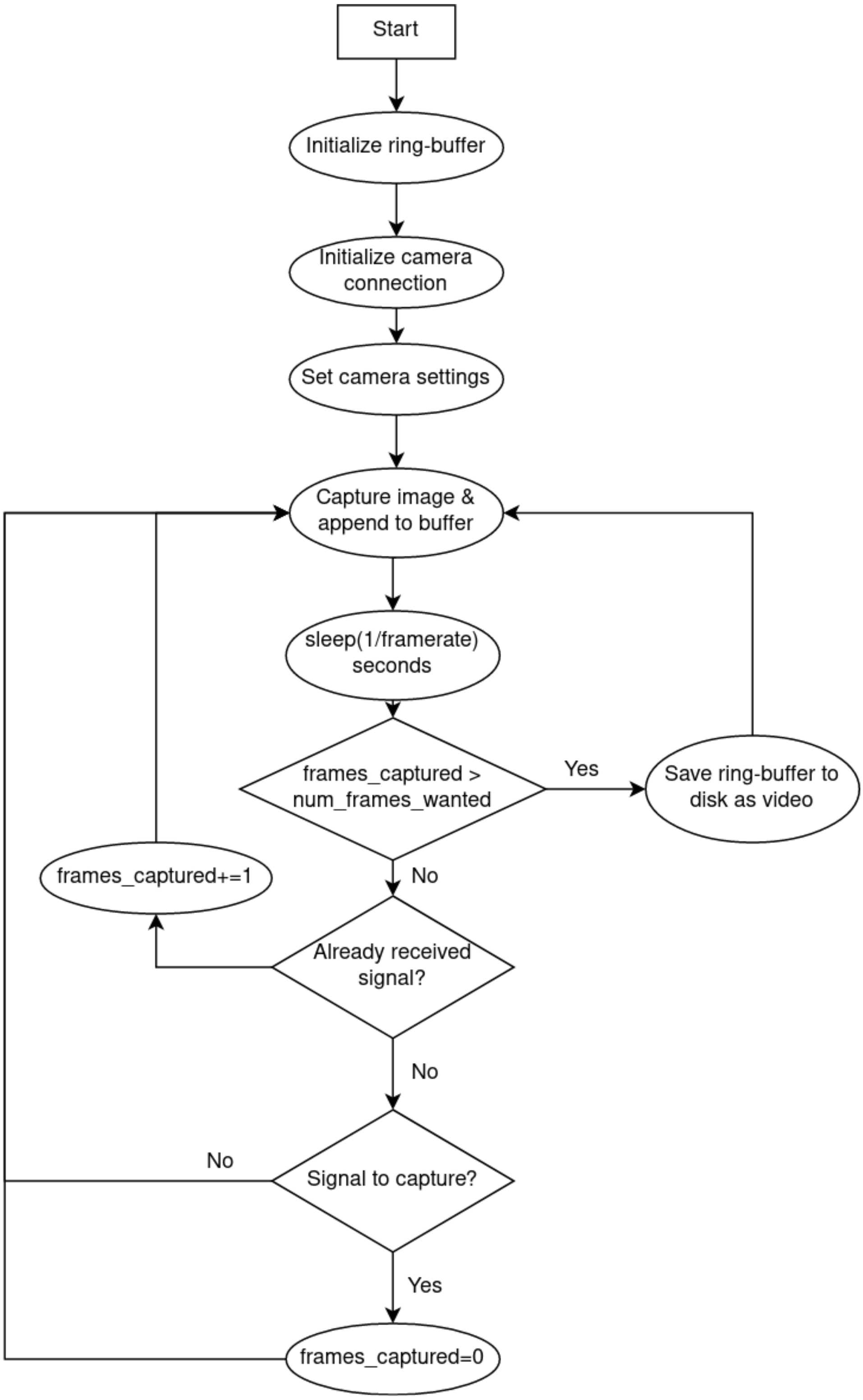


Figure 4.4: Flow chart showing general function of the Camera subsystem

Storing the images as a video allows for some inter-frame compression to be used, which significantly reduces the amount of disk space required to store the images without causing the level of detail loss that would come with compressing each image individually. In addition, thanks to OpenCV's support for the hardware H.264 video encoder on the Raspberry Pi, this compression is much more power efficient and faster than alternative methods.

The use of a ring-buffer ensures that not only is the period after the bird is detected captured, but the period before as well.

4.1.4 Coordination of subsystems

A coordinating process, implemented by `tuxcap_v2/main.py`, serves to connect all the subsystems and allow them to communicate. The coordinating process is started, and immediately intializes the three subsystems. It then monitors for flags from the RFID and scale subsystems. If a flag is raised by either subsystem, the coordinating process signals the camera subsystem to begin recording. The coordinating process then clears the flags and continues monitoring. In the event of an error, the coordinating process ensures that remaining processes are gracefully shut down so that they can be restarted.

4.2 Networking tasks

An important requirement for the new APMS was the ability to connect to the device and perform arbitrary tasks, ranging from data extraction to software updates. A series of tools were created to enable these tasks.

4.2.1 Establishing network connectivity

The SIM7600E modem HAT used for network connectivity requires several configuration steps in order to provide internet access. By default the device does not appear as a network device for the Raspberry Pi.

A script for controlling QMI network devices based on a script found on the Raspberry Pi forums [18] is used to establish the connection (documentation on how to properly configure QMI devices manually is extremely sparse, and determining the full set of commands took some trial and error). The full script can be found at `scripts/gsm.sh`. It configures the modem driver for network device usage, starts it, and performs a DHCP request so that it is provided an IP address on the cellular network. With this complete, the rest of the network features of the APMS can be started.

4.2.2 Creating access tunnels

In order to provide an access point for the APMS, a reverse-SSH tunnel is used. A reverse-SSH tunnel allows easy access to a device like the APMS, which will be assigned a random IP address by the cellular network, by relaying the connection through a known server.

This connection is created using a script which an be found at `scripts/connect.py`. This script checks for the existence of an SSH process, and if it is not found, starts an SSH process that establishes a relay to the server. If this connection is lost, the SSH process will exit. A user with the appropriate login credentials can SSH into the APMS'es Raspberry Pi via a selected port on the server, and can use this to run commands, inspect the system, or move files around using standard tools such as SFTP and SCP.

4.3 Self-starting and recovery

The APMS is expected to function for an extended period of time without direct intervention, and is designed to be assembled offsite as a sealed system to prevent ingress of saltwater and dirt which may damage the electronics. It must initialize automatically and gracefully handle faults such as losing network connectivity or unexpectedly hard reboots, since even with the addition of an access tunnel, the tunnel is only useful if it can be maintained. A number of methods are used to ensure that the APMS is able to recover on its own.

4.3.1 Monitoring tool recovery

The monitoring program is tied to a Linux systemd service, which starts at boot. Systemd is an advanced system management daemon which can be used to keep processes alive and gracefully handle failures. In the event of the monitoring process failing, systemd will wait a few seconds, and attempt to restart the process. If this fails for an extended period of time, systemd will reboot the entire APMS in an attempt to resolve the situation. The systemd service in control of this can be found in `systemd/apms.service`.

4.3.2 Network connection recovery

The network connections of the APMS are managed by their own systemd services, which are used to ensure that all actions occur in the correct order. On boot, `systemd/setupgsm.service` will run the network setup script, which sets up the QMI modem and gets a DHCP IP address on the cellular network. This service regularly checks if network service is available, and reconfigures the QMI modem if it is not.

The reverse-SSH tunnel script is run every hour by a cronjob. This script will attempt to create the tunnel if it does not exist. In this way, the APMS can recover from lost connections and will quickly re-establish connection to the server if, for example, network connection is lost.

Chapter 5

Testing

This chapter documents the procedures used to test the APMS. These tests were performed to ensure that the software and hardware driving the APMS was reliable over an extended period of time.

Field testing of the APMS was prevented by the 2020 pandemic. Testing performed was a series of modular tests to prove the function of each subsystem, as well as a comprehensive test of the entire APMS to measure the ability of the APMS to survive an extended deployment in the field.

5.1 Modular Tests

Each software subsystem of the APMS can be run standalone for testing purposes. This was used to verify the behaviour of each subsystem before a comprehensive test was performed. These tests were performed iteratively during the design process to ensure that the finalized design had the ability to perform all the tasks that were required of it.

5.1.1 RFID scanner

The RFID submodule can be run standalone, and will print any detections to the screen as well as logging them to the file. A sealed electronic tracking tag passed through the antenna should register a consistent serial number. Note that the RFID scanner chosen for this project is extremely sensitive, and testing performed indoors near sources of AC electricity may be unrepresentative of the true performance of the antenna.

5.1.2 Scale sensor

The scale measuring submodule will print out measurements when it has detected a weight present on the scale. Once the weight has been removed, it will save these readings to the appropriate log file. Placing a known weight on the scale allows for calibration of the scale sensor. Once the scale is calibrated, it can be tested by supplying new known weights and inspecting the readings produced by the subsystem to see if they are within acceptable limits of 50 grams (as suggested by the SANCCOB).

Care should be taken to ensure that the scale housing is aligned correctly, as the moveable part of the scale can sometimes scrape up against the side and become stuck when it is transported.

5.1.3 Video subsystem

With the webcam plugged in, running the video submodule will provide a text interface which can be used to manually trigger recordings and query the status of the buffered recordings. Entering the command "cap" will trigger a recording as though the RFID or scale has detected a bird, including ignoring duplicate events. The "show" command will show the status of the contents of the video buffer. Testing involves capturing a variety of scenes to test the quality of automatic focus and exposure, as well as ensuring captures are logged correctly.

5.1.4 Network connection

The provided `scripts/gsm.sh` script can be executed standalone in order to establish a network connection. If the connection has been successfully established, it will be possible to ping remote servers from the device as well as fetch web pages and access other web resources.

5.2 Comprehensive test

The scale and RFID subsystems were fitted with software-based dummy inputs that simulated events occurring at multiple points during the day, the APMS was connected to wall power using a conventional 2.1A, 5V USB power supply, and left to run for 14 days. During this time, the APMS was monitored over its remote connection and any faults or failures were assessed and, if necessary, corrected.

Figure 5.1: APMS as modified for testing. Note the use of a conventional AC-DC powersupply

5.2.1 Dummy input details

A data generating script can be found at `testing/run_test.sh`, and calls on `testing/data_generator.py` to provide simulated values. This script creates a simulated serial device that fed input to the RFID subsystem, as well as interacting with a dummy HX711 library (implemented as `tuxcap_v2/fake_hx711.py`) that has the same function call signature as the true library, allowing it to be dropped in to the original subsystem with minimal modification. The dummy inputs are considered lightweight enough that running them on the APMS itself will not significantly impact the performance of the device. The entire simulator script is kept alive with a systemd service, which can be found at `systemd/testrig.service`.

The data generator will randomly select if it should generate a scale event, an RFID event, both, or neither with configurable probabilities every 55 seconds. The probabilities for these events have been selected so that an average of 200 events will be simulated per day: the SANCCOB expects that this is approximately the number of daily events the final system will detect per day. Simulated RFID events feature randomly generated tag ID's, and simulated scale events feature randomly chosen weights which may or may not be within the threshold for the scale. Simulated weights are applied for random lengths of time within a given range. Each event is logged by the data generator to allow for comparisons.

5.2.2 Performance monitoring

The system is monitored as though it is installed at the intended site: unless there was a serious failure that rendered remote connection impossible, all checks on the system would be carried out over the cellular network connection to the device. On a daily basis the list of simulated inputs is compared against the records of inputs that the subsystems have logged. If there is a discrepancy, the system logs from around the time of the missing data can be checked for error messages or other information about the error.

5.2.3 Network access

At least once per day, an attempt will be made to log in to the device over the network. This serves the dual purpose of allowing the tester to inspect the function of the device and ensure that it is still detecting the simulated events, and ensuring that the network connection itself is working.

5.2.4 Video capture

The USB webcam was left attached to the system for the duration of the test. The camera is moved occasionally to provide a variety of different imaging situations which create a more varied range of possible images for the video processing subsystems, providing a slightly more realistic load on the system.

Chapter 6

Results

This chapter documents the outcomes of both per-module tests and the extended simulation testing performed. Each section deals with the outcomes of testing with regards to either a particular design goal or with regards to a failure that was discovered during testing.

6.1 Subsystem testing

Each individual subsystem was run standalone once development was finalized to ensure that each one provided the full range of expected functionality.

6.2 RFID scanner

In testing, the RFID scanner subsystem was able to identify and produce logs of tags reliably. These measurements were logged along with accurate times to the correct log file. Times are stored in second since Unix epoch so that it times are much easier to parse. An excerpt of a log file is shown below:

```
1589956735,019.3C98156EDB
1589957120,A99.40AB6520C9
1589957233,D89.2A08B5D4F5
1589957729,FF6.AF73A54C27
1589957841,DC1.45B83522F0
1589958502,A86.EC23D7A235
```

```
1589959108,ED8.1D0819AE79
1589959163,BEC.2162EB0EAA
1589959495,1FE.460ADCD389
```

6.3 Scale sensor

During testing, the weight sensor was calibrated by adjusting the calibration factor until a collection of known weights measured accurately on the scale to within 40 grams. The scale accurately detected the application of new weights, and faithfully logged every sample to disk when the weight was removed.

6.4 Video module

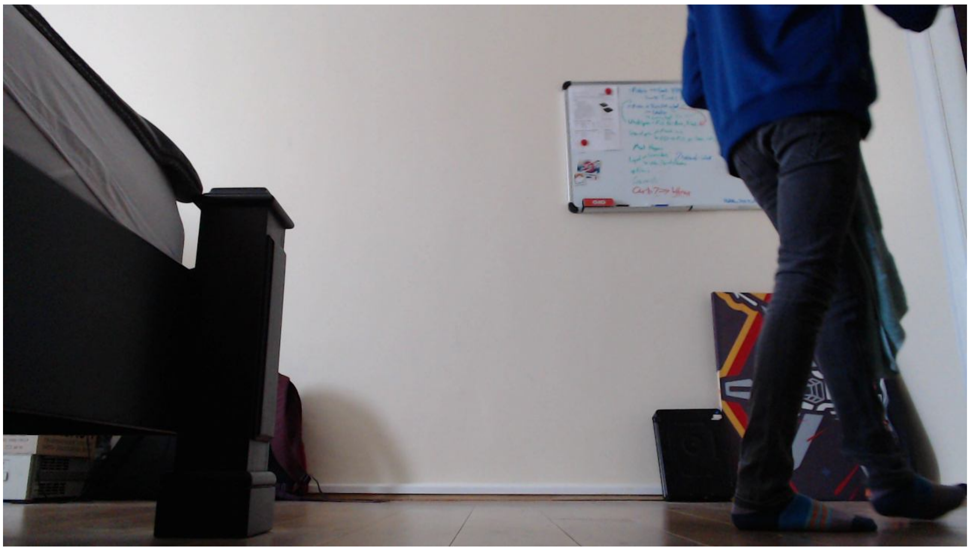
The video module was tested by capturing several different scenes, which allowed a useful measure of the size and space required for capturing video. These tests showed that a reasonable balance for capturing video is to capture 7 seconds of video, 3.5 second on either side of an event being detected, at 1280×720 pixels and 10 frames per second. 1280×720 pixels proves a reasonable balance between of file size and detail collected, and would provide suitable detail to identify penguin markings at close range. These values were reused for the comprehensive test. An example of a frame of footage captured by the video module during testing can be found in figure 6.1.

6.5 Comprehensive test

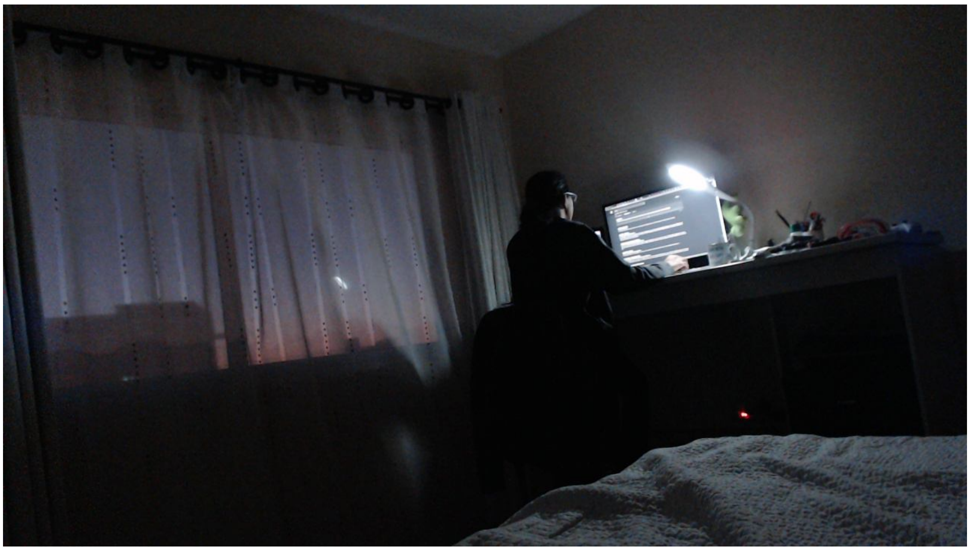
Several subsystem failures were detected during the run of the comprehensive system test. These failures can be attributed to various errors during design of the system or the test, and in many cases a correction to the error was added. This has been noted where it was done.

6.5.1 Loss of power to essential systems

During testing, several subsystems failed due to issues with the power supply. Due to the lack of a 24V power supply to test the system as it would be installed, a 2.1A rated USB power supply was used. As noted in under the Power Supply subsection, the full APMS can demand up to 14.51 watts of power under a heavy load. A 2.1A rated supply



(a) Capture frame of the author walking past the test rig during the day.



(b) Capture frame of the author sitting at their desk in a low-light situation.

Figure 6.1: Two example frames captured by the video submodule during testing.

Subsystem	Mode of Failure
Data collection	USB flash storage is effectively disconnected when insufficient voltage is maintained
Video capture	USB webcam enters an indefinite restart loop when insufficient power is available

Table 6.1: Subsystems which fail when faced with insufficient power supply

is only capable of supplying up to 10.5 watts of power at a stable voltage. Under high-power tasks such as capturing video or transmitting to the network, the APMS drew an excessive amount of current for the available power supply, causing the voltage to drop below the minimum rating of several devices. Details of which subsystems failed can be found in table 6.1.

These failures occurred due to an insufficient power supply. When installed as a complete system with the recommended 15W power supply, the APMS should be perfectly stable, even under a heavy load. In addition, thanks to the modular nature of the APMS software, when parts were disabled by a lack of power, the system continued to operate and would eventually regain full function when power requirements fell back into the range the available power supply could provide.

6.5.2 Loss of network connection

During early testing, it was found that a programming error resulted in the reverse-SSH tunnel not restarting appropriately in the event that network connection was lost. Such a failure would leave the device potentially functional, but would not allow remote access.

Correcting for this error was a simple matter of adjusting the SSH options used to create the reverse-SSH tunnel. By making use of the option **ServerAliveInterval**, it is possible to instruct the SSH client to fail and restart when a connection is lost for an extended period of time rather than continuously trying to reconnect. This allows the device to retry until a connection is reestablished.

6.5.3 Ease of access

Access to the APMS via the reverse-SSH tunnel was reliable, and provided an easy method for accessing data on the device as well as a means to inspect logs and adjust software. Downloads could be performed at up to 2 megabytes per second over the network, which is more than sufficient for the small files produced by the APMS. Due to the relayed nature of the SSH connection, latency is quite high, ranging from 0.5 to 4 seconds depending on connection quality.

6.5.4 Disconnection of flash storage

In the event that the flash storage became undervolted or otherwise temporarily disconnected, it was possible that the device would fail to notice and start saving data to the limited onboard storage of the Raspberry Pi. If this continued for an extended period of time, it may completely fill the system storage and cause instability. A correction for this error was added during testing which automatically checked for and remounted the flash storage. This effectively prevents any future failure from causing lasting damage.

6.5.5 Scale and Scanner subsystems

It is unfortunately impractical to accurately simulate a penguin walking across a scale for every single faked scale input, as this would require a much more complicated simulator running on the APMS. Nevertheless, simulated scale and scanner inputs to the relevant subsystems were accurately detected and led to appropriate data being logged in the correct log files, proving that at least the basic software function of these systems is complete.

6.5.6 Video capture data

During the run of the test, 2280 video files were recorded by the APMS successfully, representing the expected number of samples for a 10 day long run, with the remaining 4 days worth of video lost as victims of power brownouts, probability, and remote debugging periods during which data collection was paused. While these videos are unfortunately relatively static (it is once again impractical to supply a realistic simulation of a penguin, especially to a video camera), these videos provide a useful means to compare the storage space savings offered with video compression over storing individual images.

Each frame of video (captured at 1280×720 pixels), stored as a JPEG file, occupies an average of 200 kilobytes. The seven-second videos are recorded at 10 frames per second, so if these images are all stored uncompressed, each video would require 14 megabytes of storage space. With the advantages of inter-frame compression provided by H.264 encoding in an mp4 video, these images can be stored in less than 2 megabytes, or less than 15% of the space. This allows the storage required for a six-month deployment of the APMS to be reduced from over 500GB to as low as 72GB. Given the ready availability of 128GB flash storage in almost any computer store, this provides an extremely low cost and easily replaceable means of storing the data captured by the APMS.

Example frames taken by the video capture system can be seen in figure 6.1.

Chapter 7

Discussion

This chapter covers the results of testing as they concern the design requirements for the APMS, and how well the system met these requirements in testing. Some of the benefits or weaknesses of the system as implemented will be highlighted.

7.1 Modularity

Given the ease with which individual tests for subsystems could be performed, it can be reasonably claimed that the APMS succeeds in having a modular software design. Testing individual modules resolved many teething issues during design and development of the system, likely resulting in a more well-designed final system than would have been possible if only comprehensive testing was available.

7.2 Resiliency to failure

The APMS has proven that it is fairly resilient to failure. Under testing conditions with a sub-optimal power supply, it was still able to function for extended periods with minimal intervention. Many of the systems are able to perform their tasks ad infinitum provided that there is not catastrophic damage to the hardware that they depend upon to function. The use of a garbage-collected and memory-managed language for the implementation significantly reduces the chance of a software error in normal usage leading to a serious failure.

Several of the high impact failures that the system encountered during testing were due to the limited power supply used during simulation. Future testing would provide more representative results if it is performed using the recommended power supply as it would be installed during a deployment.

None of the failures detected during any tests were caused by faulty hardware. While there is no replacement for extended durability testing, this suggests that the selected hardware can be expected to function reliably when provided with adequate protection from the elements.

7.3 Ability to recover from failure

There were occasions where due to inadequate power supply, the entire APMS was forced to reboot. In these cases the system fully recovered to a working state, demonstrating that it is able to restart all services automatically when a hard failure occurs. In addition, thanks to systemd's ability to restart processes if they exit, individual processes responsible for monitoring have low downtime when an error occurs, as the systemd daemon will simply restart any failed subsystems.

7.4 Suitability for long-term deployment

The systems of the APMS show no performance degradation after running for several days, and so it is reasonable to assume that it will be suitable for extended deployments. Provided that the system is equipped with adequate storage (at least 500MB per day of deployment assuming 200 recording events per day) and power supply, there should be no real limit on how long the system can function. Regular check-ins by a technician over the remote maintenance connection would be helpful for ensuring that the system is operating well, and this task would be well supported by the remote networking features that were added.

The existence of some programming errors does show that more extended tests would be helpful in locating all possible points of failure. It is worth noting, however that several of the failures detected during the long-term test were detected while inspecting the system over the remote connection, with fixes for these problems issued remotely without any need for direct contact. This demonstrates a clear benefit for networking the system: software errors can be corrected easily and without the expense of travelling to the site.

7.5 Ease of use

The reverse-SSH connection used to provide access to the system is not a very user friendly system. Any task involving executing commands on the device remotely requires use of a command line interface. However, the use of SSH enables the use of graphical tools for connecting to SSH-accessible filesystems, which will be fully compatible with the reverse tunnel and can be used to easily access video captures and logs stored on the APMS remotely.

The relay server used for SSH is located in Amsterdam, and so the latency of the connection is quite high, anywhere from 0.5 to 4 seconds.This could become frustrating for a user who wishes to use the system interactively to view logs. Use of a relay server located within South Africa could improve this somewhat, however local server hosting costs are relatively high.

The APMS does present some usability improvements over past implementations: the use of a commodity USB flash drive for storage of logs means no specialized equipment is needed to access stored data, and it is provided in a simple format that requires no specialized software to work with.

7.6 Novelty of the updated APMS

This improved version of the APMS represents the first time that video recording and full network connectivity has been integrated with the traditional electronic tag and scale monitoring system targeted at monitoring sea birds. Video data offers the possibility for informing both direct conservation efforts and long term population studies.

It would be simple to provide a tool that pulls captured data off the APMS on a daily or even hourly basis, allowing the data to be analyzed in near-real-time in the event of a time-sensitive threat to a colony. This compares favourably to previous studies, which could only offload data to a researcher who undertakes a potentially expensive and time-consuming field trip.

Computer vision or manual screening could allow for the detection and triage of oil spills to be performed with greater speed and accuracy than before, and long term studies will now have the option of using the images of the birds to tie a weight reading to a bird directly, rather than relying on the electronic tags, which are highly accurate but which are only read by scanners sometimes.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

The aim of this research and design project was to produce a reliable prototype penguin monitoring system suitable for use with the African penguin colonies around South Africa. This implementation provides a novel assembly of features for use in ecology studies that may be able to provide researchers with previously unavailable data. This could be used to gain a more complete understanding of the challenges facing the African penguin.

This project demonstrates the versatility and performance available to users of modern embedded hardware: cheaply capturing and storing high resolution video with a embedded system was practically impossible only a decade or two ago. The cost of storing rich data has dropped steeply and it is now entirely reasonable to capture raw data from simple sensors, where in the past the compromise of on-device signal processing was required to save on storage space. It is worth considering if other similarly legacy research tools may benefit from similar hardware upgrades. The possibility to collect more detailed or even entirely new kinds of data could help improve the quality of research.

While the hardware of the APMS has been given a significant upgrade, the user interface still leaves much to be desired. A redesign or future updates to the system would do well to provide easy to use tools for inspecting the performance of the system, accessing logs and performing maintenance. Scientific equipment should aim to be easy to use and provide the user with access to the tools they need.

This report highlights some of the challenges involved in creating a device that is reliable in hardware and software, especially when it must be able to report on its own function. There are fortunately a great many tools for developing high reliability software, ranging from memory-managed programming languages to system management daemons. Effective use of these tools can produce hardy, self-repairing systems that operate unattended for extended periods of time.

The APMS as developed presently has only been through part of a full suite of testing, and further work will be required to prove that it is reliable enough (especially with regards to weather resistance) to be used in place of existing designs. While it would have been desirable to perform such tests for this project, this was unfortunately impossible. Such work should be pursued prior to any attempts to deploy the system long-term. This report did not deal with the design for enclosures or the effectiveness of network equipment in remote locations, which may present additional difficulties that will need to be accounted for. A more thorough design effort which includes assessing the suitability of enclosures and power supply systems would result in a fully integrated system that can be holistically evaluated for suitability. Once more holistic testing has determined if a modernized APMS can provide the reliability necessary for extended scientific work, it will be a useful source of information for conservation researchers.

8.2 Recommendations

The updated APMS as it stands offers a novel combination of data collection methods to conservation researchers. The space for future work with regards to high-performance wildlife population monitoring systems is quite large, as there are a wide array of ways these designs could be furthered or improved.

Future work with the APMS could involve writing tools for interacting with it, either directly, to provide more user friendly UI and data analysis using the always on access enabled by the networking features, or for interacting with the data dumps that can be taken off the USB storage when data is collected from the installation site by a field worker. The data as it is stored presently contains useful data but is not easy to visualize without some processing, and live visualizations that could be reviewed regularly could allow for real time monitoring of bird populations, or help researchers focus their efforts when working with video captures.

A future project could involve developing a set of enclosures and power supply hardware that is purpose built for rough weather. Previous projects have frequently been cut short by weather damage affecting the computer boards or power supply, and designing a hardy installation method that complys with all relevant environmental protection regulations would be a valuable addition.

Given the computational power available in modern embedded Linux single board computers, work could be done to include real-time computer vision processing of video data, allowing for the use of real time video identification techniques such as those developed by Sherley et al. [4]. By combining this with the existing RFID system it may be possible to provide real-time identification for the majority of birds that are detected without the need for tagging, further enabling work which uses real-time data from the APMS.

Lastly, it may be possible to take advantage of the video feed for awareness and outreach projects. A public outreach project which provides live feeds of the penguins may be an interesting attraction, and live online animal videos have had some success at a variety of institutions such as the Smithsonian National Zoo Live Cheetah Camera and the many bird nest cameras that have existed over the years. Such work could be used to raise awareness of the plight of the African penguin or even serve as the basis of a public fundraising project.

Bibliography

[1] *Spheniscus demersus: BirdLife international*, Aug. 2018. DOI: 10.2305/iucn.uk.2018-2.rlts.t22697810a132604504.en. [Online]. Available: https://doi.org/10.2305/iucn.uk.2018-2.rlts.t22697810a132604504.en.

[2] K. KERRY, J. Clarke, and G. ELSE, "The use of an automated weighing and recording system for the study of the biology of adélie penguins (pygoscelis adeliae)," *Proceedings of the NIPR Symposium on Polar Biology*, vol. 6, Jan. 1993.

[3] J. Clarke, K. Kerry, L. Irvine, and B. Phillips, "Chick provisioning and breeding success of adélie penguins at béchervaise island over eight successive seasons," *Polar Biology*, vol. 25, no. 1, pp. 21–30, Jan. 2002. DOI: 10.1007/s003000100307. [Online]. Available: https://doi.org/10.1007/s003000100307.

[4] R. Sherley, T Burghardt, P. Barham, N Campbell, and I. Cuthill, "Spotting the difference: Towards fully-automated population monitoring of african penguins spheniscus demersus," *Endangered Species Research*, vol. 11, pp. 101–111, Mar. 2010. DOI: 10.3354/esr00267. [Online]. Available: https://doi.org/10.3354/esr00267.

[5] M. Salton, C. Saraux, P. Dann, and A. Chiaradia, "Carry-over body mass effect from winter to breeding in a resident seabird, the little penguin," *Royal Society Open Science*, vol. 2, no. 1, 2015. DOI: 10.1098/rsos.140390.

[6] M. Matavire, *Refuelling under scrutiny as s.africa penguins hit by oil spill*, 2019. [Online]. Available: https://phys.org/news/2019-07-refuelling-scrutiny-safrica-penguins-oil.html.

[7] J. Singh, *I made a robot that takes pictures of penguins.* 2019. [Online]. Available: https://awful.cloud/blog/penguin_science.

[8] *National environmental management: Protected areas bill*, 2003. [Online]. Available: https://www.environment.gov.za/sites/default/files/legislations/nem_protected_areas39_0.pdf.

[9] V. Afanasyev, S. Buldyrev, M. Dunn, J. Robst, M. Preston, S. Bremner, D. Briggs, R. Brown, S. Adlard, and H. Peat, "Increasing accuracy: A new design and algorithm for automatically measuring weights, travel direction and radio frequency identification (rfid) of penguins," *PLOS ONE*, vol. 10, e0126292, Apr. 2015. DOI: 10.1371/journal.pone.0126292.

[10] J. Geerling, *Power consumption benchmarks*, 2020. [Online]. Available: https://www.pidramble.com/wiki/benchmarks/power-consumption.

[11] *Gsm/gprs/gnss/bluetooth hat for raspberry pi*. [Online]. Available: https://www.waveshare.com/sim7600e-h-4g-hat.htm.

[12] *4g/3g/2g/gsm/gprs/gnss hat for raspberry pi, lte cat4, for southeast asia, west asia, europe, africa*. [Online]. Available: https://www.waveshare.com/sim7600e-h-4g-hat.htm.

[13] *Ps-105 parcel scale*. [Online]. Available: http://www.measuretek.net/upload_pdf/PS-105-Parcel-Scale.pdf.

[14] Tatobari, *Hx 711 for raspberry pi*, 2018. [Online]. Available: https://github.com/tatobari/hx711py.

[15] *Low-pro square pit tag antenna*. [Online]. Available: https://www.biomark.com/low-pro-square.

[16] *Is1001 reader board*. [Online]. Available: https://www.biomark.com/is1001-reader-board.

[17] *Usb - raspberry pi documentation*. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/usb/README.md#powerlimits.

[18] mkrzysztofowicz, *4g hat: Raspberry pi forums*, 2013. [Online]. Available: https://www.raspberrypi.org/forums/viewtopic.php?p=1374909#p1450784.

Appendices

Appendix A

Wiring Manual

This chapter briefly documents the wiring required to assemble the APMS.

A.1 SIM7600E

The SIM7600E 4G HAT must be attached to the top of the Raspberry Pi as shown in figure A.1. Ensure that the yellow jumper blocks are set to A and 5V as shown in figure A.2.

Figure A.1: Connect the SIM7600E to the Raspberry Pi, ensuring the entire row of headers is attached firmly

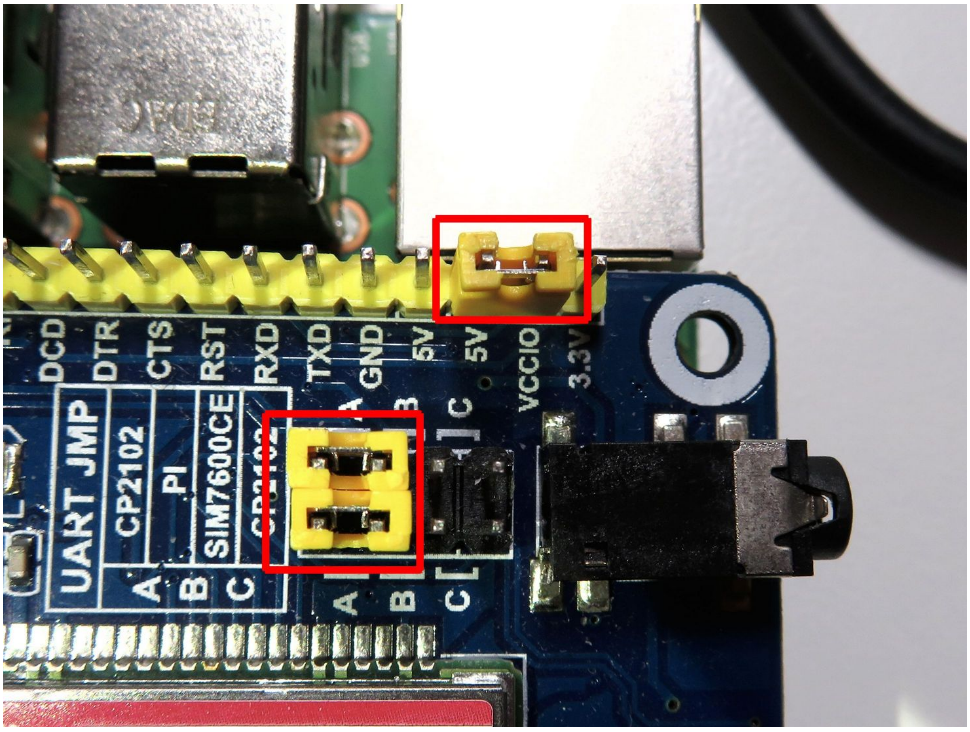


Figure A.2: The yellow jumper blocks should put the SIM7600E HAT into mode A, and select 5V for the VCCIO value

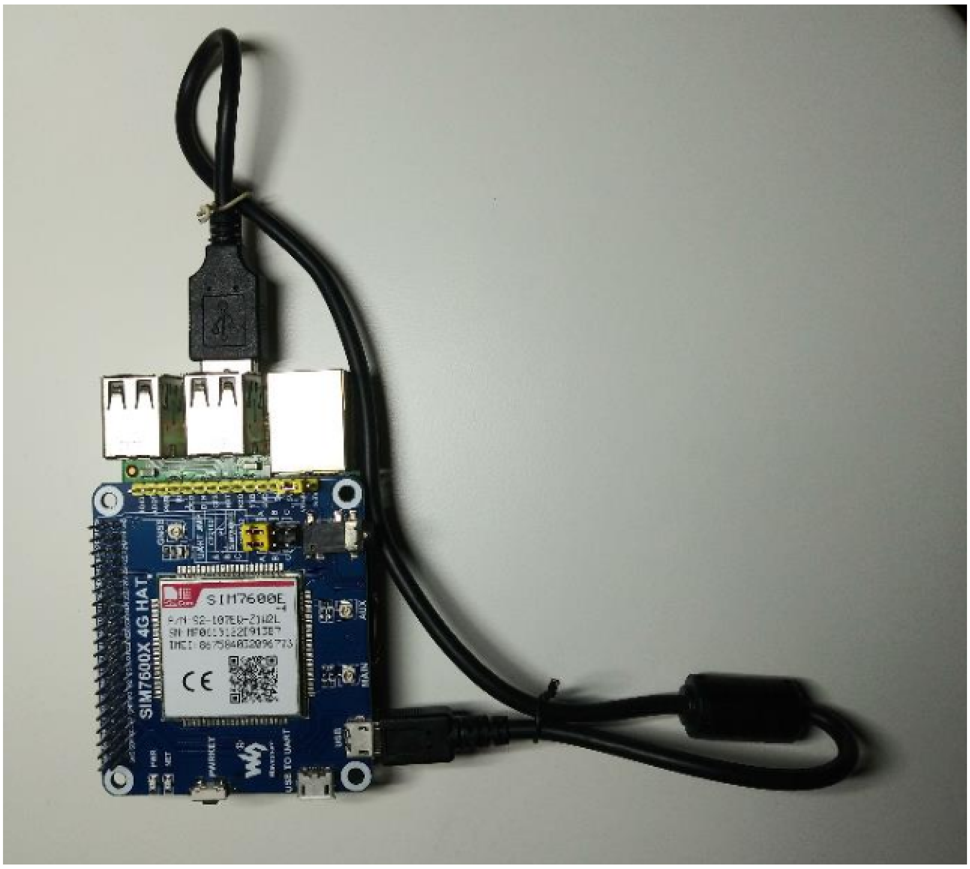


Figure A.3: Connect the 4G HAT to the Raspberry Pi over USB.

A.2 Connecting 4G HAT to Raspberry Pi over USB

Use the micro-USB cable provided with the SIM7600E HAT to connect the HAT to one of the Raspberry Pi's USB ports. Ensure that the 4G HAT is connected using the micro-USB port labelled USB, and not the one labelled "USB to UART", as shown in figure A.4 and figure A.3

Figure A.4: Ensure that the circled USB port is used.

A.3 HX711 board wiring

The HX711 breakout board is used to connect the package scale to the Raspberry Pi. In order to wire it to the scale, first cut the monitoring module off the top of the scale and strip back the cable to reveal a set of four wires: Red, Green, White and Black. These four wires connect to the scale-side of the HX711 board as shown in figure A.5

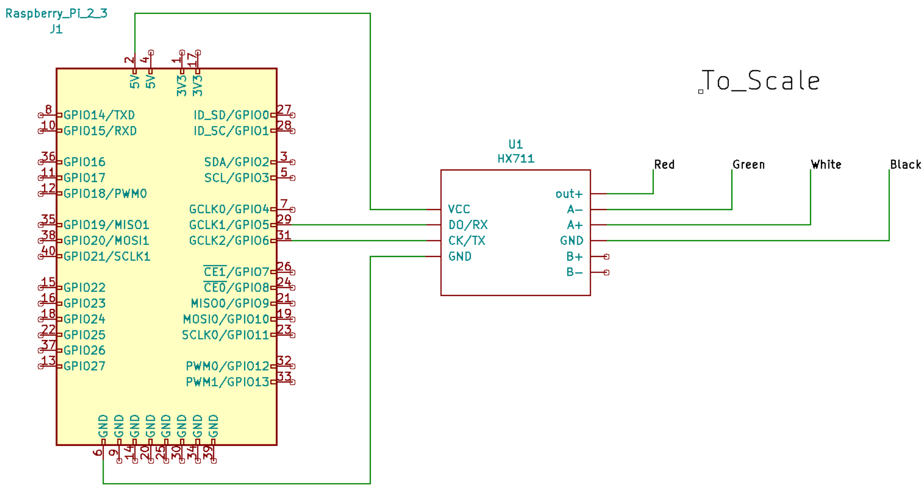
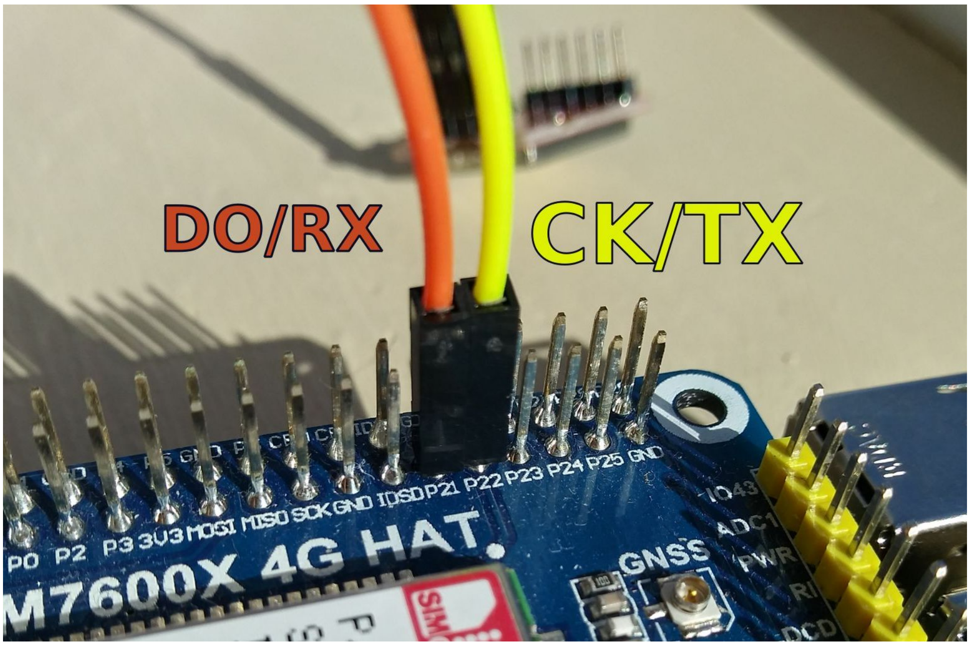


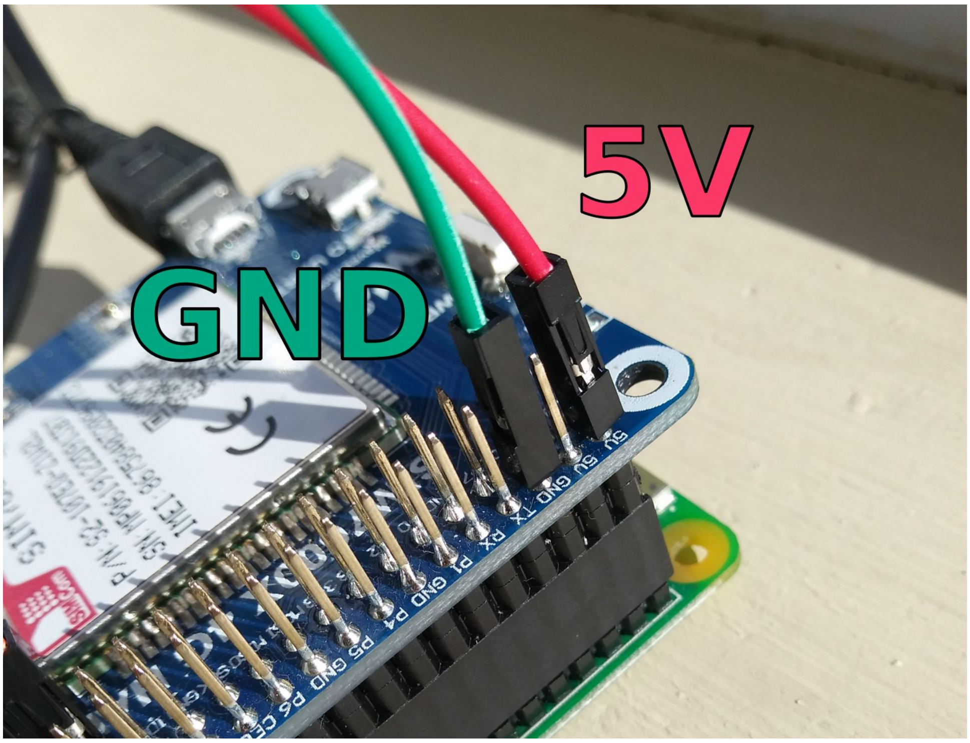
Figure A.5: Ensure that the circled USB port is used.

The power and data lines for the HX711 board connect to the Raspberry Pi. For clarity, the following photos of wiring are also provided in figure A.6.

- VCC: 5V (Pin 2)
- DO/RX: P21 (Pin 29)
- CK/TX: P22 (Pin 31)
- GND: GND (Pin 6)



(a) Data pins from the HX711 connected to the Raspberry Pi



(b) Power pins from the HX711 connected to the Raspberry Pi

Figure A.6: Visual aids for wiring the HX711 breakout board to the Raspberry Pi.

A.4 Power Supply

The system is powered with a Meanwell SD-15B-05 5V power supply. Cut and strip a Micro USB cable and attach the positive and ground wires to the V+ and V- terminals of the power supply, and wire the Input + and – terminals to the 24V power source. Test the supply with a 1.8 Ohm resistor and ensure that the adjustment is tuned to provide 5V at this current.

A.5 USB connections

The webcam, RFID scanner and flash drive must all be connected over USB to the remaining USB ports on the Raspberry Pi. These three can be connected to the remaining USB ports on the Raspberry Pi in whatever order makes the most sense for fitting inside the enclosure.

Appendix B

Software

All software written for this report can be found in a GitHub repository at **github.com/KaliumPuceon/APMS_code.**