Re-Entry Capsule Data Logging And Release System

by Cairin Michie

Prepared for P. Martinez
Department of Electrical Engineering
University of Cape Town

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.

SL17-01U
19 June 2017

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.

Abstract

This report describes the design and development of a release platform to perform a drop test of the MIRKA2 atmospheric re-entry capsule from a high-altitude balloon. The system is required to acquire and transmit data on the ambient conditions, to perform a controlled release of the MIRKA2 capsule and to document this release with video imagery. The system comprises various sensors connected to a Raspberry Pi Zero single-board computer. Communication is achieved over a LoRaWAN network with a ground station via an on-board LoRa module. A burn wire mechanism is used to release the MIRKA2 capsule at a specified altitude. The system also has two imagers to document the release of the capsule, one which is side-viewing, and the other which is downward-viewing. Thermal control of the system enables the functioning of temperature-sensitive electronics at the low temperatures prevalent in the upper atmosphere, and a parachute allows for the successful recovery of the system following the release of the capsule. Flow control of the various stages of the flight is used to ensure that the possible error conditions (e.g. premature rupturing of the balloon) do not result in mission failure. The functional elements are verified through a testing process, and the conclusion is drawn that the functional requirements have been achieved, and that the system is ready for flight tests. The report concludes with various recommendations for improvements of the system.

Acknowledgments

Prof. Peter Martinez for his dedication, and invaluable advice throughout this project.

My family for their continued support, advice, and motivation to perform to the best of my abilities.

My grandparents for their generosity, and being the exemplary role models they are.

Table of Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Background

A space object entering the Earth's atmosphere will experience aerodynamic heating that can cause it to disintegrate and burn up completely. A re-entry capsule is designed to allow the survival of atmospheric re-entry for a space object. In the case of the Soyuz spacecraft the re-entry capsule is a crewed vehicle. However, smaller versions of re-entry capsules have been designed to allow samples to be transported back to Earth.

The subject of this project is the design and construction of a controlled-release platform for the MIRKA2, a small re-entry capsule designed by the Small Satellite Student Group at the University of Stuttgart.

During the design process of re-entry capsules, high-altitude drop tests are required. The MIRKA2-ICV (Micro Return Capsule 2 - In-flight Communication Verification) is a high-altitude drop test for this capsule. This mission succeeds the previous MIRKA2-RX mission, which involved the release of the MIRKA2 capsule from a sounding rocket. Although the mission was a success, the capsule did not activate on ejection. Therefore the verification of the communication link on the MIRKA2 was not possible.

Hence, the following objectives for the MIRKA2-ICV experiment were defined: [2]

1. Verification of the in-flight communication with the Iridium satellite network; and
2. Verification of the capsule's activation mechanism.

1.2 Objectives

The objective of this report is to detail the design, fabrication and testing of a release system for the MIRKA2 re-entry capsule. This system will allow for the reliable release of the MIRKA2 capsule from a high-altitude balloon. The video-graphic documentation of this release will allow for the observation and analysis of the stability of the MIRKA2 capsule during free-fall. Transmission of flight-critical data is required for the recovery of the release platform, and to support the objectives of the MIRKA2-ICV mission.

1.3 Technical challenges

There are several technical challenges that arise from the nature of high-altitude balloon experiments. These are:

1. The balloon will ascend to a high altitude somewhere in the range of 25-35 km above sea level (ASL). During its ascent to this altitude, the balloon may also be carried a considerable distance from the launch point by winds that may be in excess of 30 $m.s^{-1}$ or 100 $km.h^{-1}$ [3]. The distance will present a challenge as the communication system will have limited power with which to transmit data. The ascent to this altitude would also expose the balloon payload to a wide range in temperatures.

2. The weight of the system (including the MIRKA2) may not exceed half the neck lift capability of the balloon, as this will prevent the balloon from achieving sufficient lift.

3. The stretch of a latex balloon decreases as it approaches breaking point. If the ascent rate is low, equilibrium between the upward force of the lifting gas (helium or hydrogen) and the resistance to stretching can be achieved. This will cause the balloon to stop expanding, and stop ascending. The balloon will not burst and will not be retrievable, as it will have floated out of the tracking range. Therefore, an ascent rate greater than 4 $m.s^{-1}$ is advisable to reduce the risk of floating.

4. The expenses incurred and time taken during the development and assembly of both the MIRKA2 capsule and the release platform will be wasted if the system fails. Although the system only needs to work once, it cannot fail.

Chapter 2

The MIRKA2 capsule

The MIRKA2 capsule is the smallest re-entry capsule developed to date. The small size of the capsule is attributed to the fact that it needs to fit inside a 1U CubeSat[1] module. The MIRKA2 capsule was developed by the Small Satellite Student Group at the University of Stuttgart, which is a society of students with engineering and scientific backgrounds.

Figure 2.1 shows isometric, side and top views of the MIRKA2 capsule, respectively. Detailed drawings of the interior assembly are shown in the MIRKA2-ICV Experiment Documentation [2].

Figure 2.1: Isometric, side and top views of the MIRKA2 capsule.

2.1 Purpose

After activation the capsule is required to transmit data, gathered by the on-board sensors, via the Iridium [4] satellite network. The data acquired details the behaviour of a light-weight ablative material called ZURAM which has been developed jointly by the German space agency DLR and the University of Stuttgart.

The MIRKA2 has the following on-board systems to ensure the required functionality:

- GPS receiver and antenna

[1] The CubeSat standard is a standard for the development of nanosatellites in standard size increments of 1U, or one unit, which is a cube sized 10 $cm \times 10\ cm \times 10\ cm$.

- Iridium transmitter and antenna
- Inertial measurement unit
- Thermopile
- Analog pressure sensor
- On-board micro-controller

2.2 Design

The MIRKA2 is an improvement on the capsule used in the MIRKA2-RX project. Changes to the MIRKA2 include improved electronics, software, and an improved activation mechanism.

2.2.1 Back shell

In the top view of the MIRKA2 capsule shown in Figure 2.1, four concentric sets of boreholes and recesses in the back shell are numbered. These boreholes and recesses serve the following functions:

1. An eyelet for a polyethylene cord to fasten the MIRKA2 to the release platform;
2. Three boreholes for aligning the MIRKA2 capsule to the release platform and to prevent rotational movement of the MIRKA2 capsule;
3. Four recesses for target contacts used in activating, providing power to, and interfacing with the MIRKA2 capsule; and
4. Three boreholes for connecting the back shell to the lower half of the capsule's structure.

2.2.2 Activation mechanism

Only one of the four sets of target contacts mentioned previously will be required for the purpose of activating the MIRKA2 on-board micro-controller and electronics. The activating mechanism consists of 2 spring-loaded pins. The MIRKA2 capsule has been designed such that when the activation pins are connected together, forming a short circuit, the MIRKA2 on-board micro-controller and electronics remain in a deactivated state. Therefore when the MIRKA2 is released, the spring-loaded pins will break contact with the target contacts and the MIRKA2 on-board micro-controller and electronics will activate.

The data-sheet for the activation pins is available in the MIRKA2 Release Experiment landing page [5] that also contains numerous other technical details pertaining to this project.

Chapter 3

High-altitude balloons as an atmospheric research platform

3.1 Motivation

High-altitude balloons provide an easy and cost-effective means for achieving near-space flight conditions, making high-altitude ballooning an attractive method of high-altitude research. High-altitude ballooning makes near-space research possible for institutions such as universities or small companies that would not normally be able to afford other means of space research. High-altitude ballooning also provides a cost-effective means for testing equipment that will one day operate in space.

3.2 History

High-altitude balloons have been used for atmospheric research since the 19^{th} century. In more recent years they have been used for a wide range of purposes, including tracking weather patterns, taking and transmitting high-altitude photographs of the Earth, and as research platforms.

3.3 Flight train

A high-altitude balloon has various characteristic components, which make up the 'flight train', as indicated in Figure 3.1. The payload, which is the raison d'être for the flight, will often consist of various components, and can even consist of multiple separate sub-payloads.

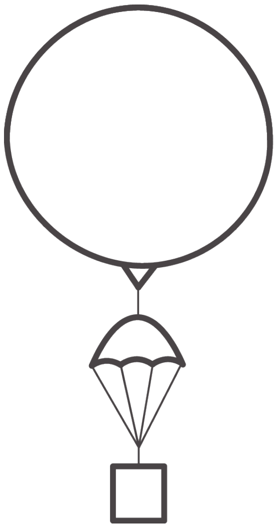

Figure 3.1: Typical flight train for a high-altitude balloon experiment.

High-altitude balloons are typically filled with hydrogen or helium. Hydrogen has a greater buoyancy, but is seldom used because of safety concerns.

Balloons come in various sizes, according to their capacity and weight. The balloon's flight parameters can be calculated to determine the neck lift and ascent rate for various payload weights.

3.3.1 Typical flight profile

High-altitude balloon flight profiles vary based on weather conditions such as wind, air pressure, and precipitation. Regardless of these factors, however, a successful balloon flight will typically have five distinct stages.

The launch stage lasts until the balloon is released. The launch stage is a critical stage in the flight, as it provides an opportunity to perform a final check of the various system functions. It is also the only stage that is influenced by the operators. The ascent makes up the second stage. The balloon burst is the third stage, at which point the payload will start its descent - the fourth stage. The final stage is the landing and recovery, in which the payload is recovered.

Figure 3.2 shows an example flight path prediction taken from the CUSF flight predictor [6]. In this image you can see the launch at the red marker, the ascent to the apogee or burst of the balloon, and the descent to the predicted landing point at the green marker.

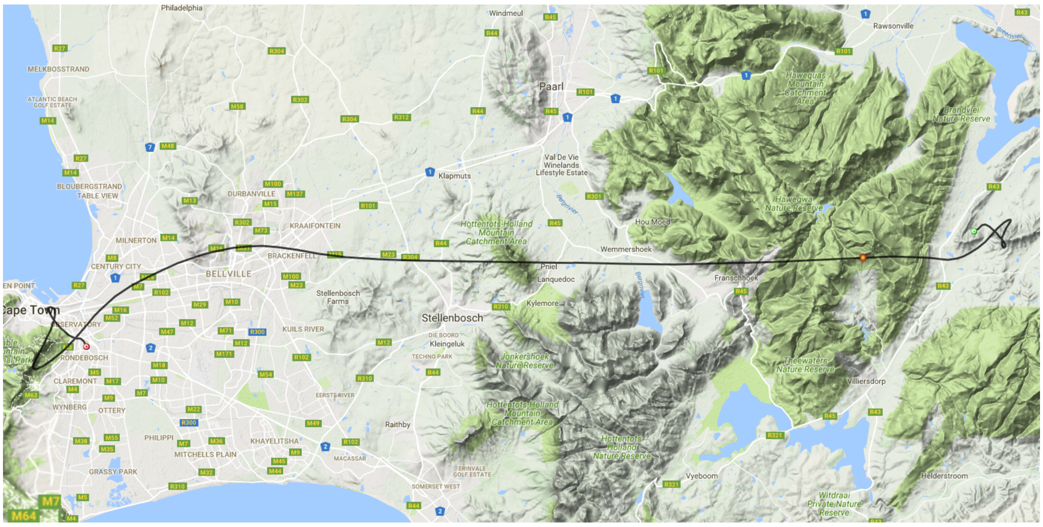

Figure 3.2: An example flight prediction for a balloon released in Cape Town with a westerly wind. The flight reached an apogee of 35 km ASL, with an ascent rate of 5 $m.s^{-1}$.

3.3.2 Balloon calculations

Before the flight of the balloon can take place, it is advisable to take certain precautionary measures. A number of calculations allow for prediction of the flight path and parameters that may be necessary for the success of the flight.

There are a variety of online tools available to facilitate the planning for a high-altitude balloon flight. A balloon burst calculator, for example the habhub balloon burst calculator [7], calculates the burst altitude, time to burst, ascent rate, volume of helium, and neck lift of the balloon using a target burst altitude or ascent rate, as well as the payload and balloon mass. This calculation is necessary to ensure that the balloon provides sufficient buoyancy to lift the payload to the desired altitude.

A parachute size calculator, for example the RocketReviews parachute calculator [8], can be used to determine size of a parachute necessary for a given payload mass. Using the size parachute thus calculated, one can calculate the descent rate of the payload using a descent rate calculator [9].

3.3.3 Atmospheric conditions calculations

Further predictions are advised regarding the atmospheric conditions on the launch day. A weather forecast [10] will allow for choosing of a clear sky day with minimal wind. Flight prediction software [6] should be used to predict the flight path of the balloon. This software uses predictions of weather patterns for the day of launch, as well as some of the calculations regarding the flight train. The frequent use of this software during the week leading up to launch day will enable more accurate prediction of the flight path.

Using the 1976 Standard Atmosphere Calculator [11] the temperature and pressure profile of the Earth's atmosphere can be estimated. This helps with the design of payloads that will be able to function properly in the wide range of temperature and pressure variations experienced during a flight.

Chapter 4

Requirements definition

This chapter provides an outline to the various design requirements of the system. This will serve as a guideline throughout the design, assembly and testing phases for this project.

4.1 Functional requirements

1. The system must provide telemetry during the flight, which must include location, temperature, and altitude.
2. The communication link must be duplex as a fail-safe, allowing an override of the release mechanism in the event of an early balloon burst or release failure.
3. The system must operate reliably between -60°C and 30°C, the range of atmospheric temperatures that the balloon will pass through.
4. The MIRKA2 capsule must be released at an altitude above 30 km ASL. The apogee of the balloon needs to be calculated on the flight day, and the MIRKA2 capsule must be released at least 2 km before the predicted apogee.
5. Video of the release of the MIRKA2 capsule must be captured, with one camera facing downward, and another camera viewing the MIRKA2 capsule from the side.
6. The system must have a mass of no more than 1500 g to ensure a fast ascent time to the capsule release altitude.
7. The payload must be recoverable. For this purpose a parachute and some form of mechanical shielding are needed in order to slow down the descent of the payload so that it survives landing.
8. The system must maintain power throughout the flight, and for a period of at least 3 hours after landing, to ensure recovery.

Chapter 5

Conceptual design

This chapter details the conceptual part of the design process of the platform data logging and capsule release system. The objectives of this chapter are to consider and evaluate certain design concepts, and compare them to the requirements specification in order to select design concepts to be further developed in the detailed design chapter.

5.1 System overview

This section details the subsystems, and their functional relationships, that are required to fulfil the functional requirements. These subsystems will form the basic elements upon which the conceptual design ideas are to be formed.

5.1.1 Micro-controller

The micro-controller will provide the central processing capability of the system, and an interface upon which the other electronic subsystems will function.

5.1.2 Control software

The control software will operate on the micro-controller, and will control the various subsystems. The software will control the operational flow of all operations in all phases of the flight.

5.1.3 Communication

The communication subsystem is responsible for communication between the balloon payload and the ground station. The primary function of this subsystem is to receive commands and transmit all the necessary data required for flight success.

5.1.4 Sensors

The sensory subsystem includes all the sensors used to log flight data and to visualise the release of the MIRKA2 capsule. This includes, but is not limited to, location, environmental, and imaging sensors.

The location sensor will provide information regarding the position (latitude, longitude and altitude) of the payload from which the movement during the flight can be obtained. The environmental sensory system will provide information regarding factors such as temperature and pressure of the environment surrounding the payload. The imaging system will be responsible for recording the release of the MIRKA2 capsule.

5.1.5 Thermal control

The electronic subsystems present on the payload will not function within the entire range of temperatures the system will be subjected to throughout the flight. Insulation and heating of the electronics will be necessary to avoid failure.

5.1.6 Release platform

The release platform will be responsible for supporting and releasing the MIRKA2 capsule. The release platform must contain the release mechanism that will be activated at the time of release.

5.1.7 Recovery

After the release of the MIRKA2 capsule, the release platform will return to Earth for recovery and reuse on a subsequent flight. The recovery system will enable the safe descent, landing, and subsequent retrieval of the payload.

Recovery of the payload will also allow downloading of recorded data that may not have been transmitted via the telemetry link, such as video footage or high cadence data.

5.1.8 Power supply unit

The power supply unit will provide sufficient power to the electronic subsystems. Power is required to operate the various subsystems throughout the flight, and also after landing for recovery purposes.

5.1.9 Conclusion

Figure 5.1 shows a block diagram of the functional subsystems that satisfy the functional requirements identified in the requirements definition chapter.

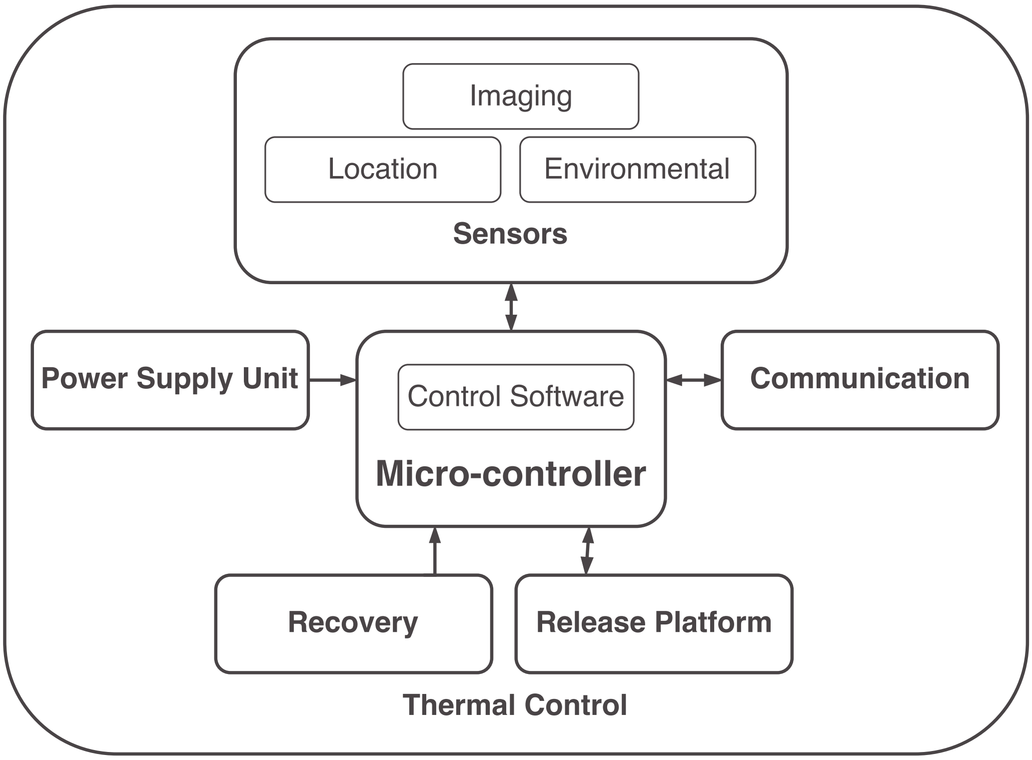

Figure 5.1: Block diagram showing the system's functional elements. The arrows pointing away from the micro-controller are control signals, and the arrows pointing towards the micro-controller are provision of data, power or protection.

5.2 Concept ideation

This section details the conceptual ideas formed based on the functional units described for the system. Each subsection presents various concept ideas to be evaluated.

5.2.1 Micro-controller

The following two concepts are considered for the micro-controller of the system.

Arduino

Arduino is an open-source electronics platform implementing specialised hardware and software. Arduino is programmed using the Arduino programming language [12]. The Arduino platform forms an abstraction layer between the programming language and the on-board micro-controller, instead of implementing an operating system.

Raspberry Pi

The Raspberry Pi is a small single-board computer. The Raspberry Pi features a Broadcom system on a chip, which includes an ARM-compatible CPU and an on-chip GPU. Raspbian [13] is the operating system used. This is a Debian-based Linux distribution [14]. Most programming languages are supported, and provide multiple ways to interface with sensory systems.

5.2.2 Control software

A programming language with which the developer is most comfortable, or has the most experience in, is preferable as this will reduce development time. However, for embedded applications various programming languages have certain advantages that need to be accounted for. These advantages include, but are not limited to, compatibility with the micro-controller, speed, and efficiency. Taking this into account, the following programming techniques were considered for the control software.

Serial

Serial programs contain a single ‘thread’ of control. This means that the flow of computation matches the code that has been written.

Concurrent

Concurrent programs have multiple threads of control. They operate on a logically simultaneous processing model, which does not imply multiple physical processing elements. This means that the flow of computation does not necessarily match the code that has been written. Concurrency can be achieved in many ways, including parallelism, switching, interrupts and threading.

Parallel

Parallel programs are executed using multiple, possibly independent, processing elements.

5.2.3 Communication protocol

The following three concepts are considered for the communication protocol of the system.

RTTY

Radio-teletype (RTTY) is a radio telecommunications system that was originally used to communicate between electro-mechanical teleprinters, and is now used by emulating teleprinters in software. This communication mechanism transmits serial data with 5-bit data encoding.

APRS

The Automatic Packet Reporting System (APRS) is a channel for Ham radio used to transmit digital information in real time. This method transmits packets that can be decoded by any receiver in range. This method is commonly employed by emergency services, such as police radios.

LoRaWAN

LoRaWAN^{TM} [15] is a proprietary Low Power Wide Area Network specification, which is commonly used in long-range Internet of Things applications. LoRaWAN^{TM} [15] allows for LoRa^{TM} [16] gateways to form a bridge between end-devices and a server.

5.2.4 Location sensors

For the purpose of this project, location information includes the geographical position as well as the altitude ASL. The following three concepts are considered for the geo-location sensors in the system.

GSM tracker

Global System for Mobile communications (GSM) is an open, digital cellular technology commonly used in transmitting mobile voice and data. Dedicated GSM trackers are available for purchase commercially, and transmit GPS data over the GSM network.

GPS

The Global Positioning System (GPS) utilises a medium Earth orbit (MEO) satellite constellation to provide location data, including altitude.

Iridium tracker

The Iridium [4] commercial constellation consists of 66 cross-linked Low Earth Orbit (LEO) satellites, providing high-quality voice and data connections over the Earth's entire surface. The satellites orbit the Earth at an altitude of 780 km ASL, this proximity provides shorter transmission paths, stronger signals, and lower latency. Dedicated iridium trackers are available for purchase commercially.

5.2.5 Pressure sensor

A pressure sensor is a device used to measure the pressure of liquids or gases. A pressure sensor can be used to estimate the altitude of the flight, as the pressure decreases with altitude.

Barometric

Barometric pressure sensors consist of a piezo-resistive sensor that can be used to calculate absolute altitude at a certain temperature.

Piezoelectric

Piezoelectric pressure sensors make use of piezoelectric crystal or ceramics to convert applied mechanical stress to an electrical output that can be used to calculate pressure.

5.2.6 Temperature sensor

External and internal temperature sensors can be used to monitor the temperature of the system, and could be used for active thermal control to ensure that the system remains within the nominal operating temperature range of the electronics. As most micro-controllers have on-board temperature sensors, the following are concepts for an external temperature sensor.

Thermistor

The thermistor is a thermally sensitive resistor that changes its physical resistance when exposed to changes in temperature. Thermistors are constructed from a ceramic type semiconducting material, such as oxides of nickel, manganese or cobalt.

Thermocouple

Thermocouples are thermoelectric sensors that utilise two dissimilar metals welded together at two junctions. One junction is kept at a constant temperature and forms the reference junction, while the other junction is used as the measuring junction. When there is a difference in temperature between the junctions, a potential difference will form across the junctions as a function of temperature.

Direct-to-digital

A direct-to-digital temperature sensor is a self-contained temperature sensing unit that will provide the micro-controller with the temperature digitally, as opposed to performing the calculations on the micro-controller. These sensors use diodes or transistors to detect changes in temperature.

5.2.7 Imaging sensors

The following concepts are considered for the imaging sensors of the system.

Embedded camera

A camera that is compatible with the micro-controller and can be controlled by the micro-controller.

FPV camera

First Person View (FPV) cameras are used to fly hobbyist quad-copters from the view of the quad-copter. They are built to be lightweight and durable, while providing good quality footage. These cameras can be controlled by a micro-controller, but do not have a self-contained power supply or SD card on which the video can be recorded.

Sports camera

A sports camera has high durability and a history of proven operation in the atmospheric conditions that will be experienced by the system. These cameras are typically enclosed in a waterproof casing, and will function autonomously from the rest of the capsule release system.

DSLR camera

Digital Single Lens Reflex camera would enable ultra high definition images and video content.

5.2.8 Thermal control

The following concepts are considered for the thermal control of the system.

Styrofoam box

A styrofoam box provides a layer of thermal insulation that can help mitigate the effects of the very low temperatures (e.g. -56°C at 20 km ASL) found at high altitudes. Styrofoam, a form of expanded polystyrene, is a lightweight material that provides good thermal insulation and impact protection.

Heating device

Additional warming of the payload can help in maintaining the system temperature within operational limits. This can be done using chemical hand warmers similar to the ones used by hikers.

Solar heating

Solar heating can be used in the form of solar panels and an electric heating system to maintain operational temperatures for the system.

5.2.9 Release platform

The following concepts are considered for the release platform mechanism of the system.

Trap door

A simple trapdoor mechanism that encases the MIRKA2 capsule, and opens the trap door at the time of release.

Burn wire

Ni-chrome wire, or other high resistance wire, can be used to cut through a polyethylene cord fastening the MIRKA2 to the release platform.

Pin and servo

A servo is a small electronic device that has a controllable output shaft. This shaft could be used to push a fastening pin out of place releasing the MIRKA2.

5.2.10 Recovery

The recovery system needs to take into account various aspects of the recovery process, namely the safe return of the payload back to Earth and the telemetry data to locate the payload. The communication of telemetry data has already been discussed, thus this section will focus on the return to Earth.

Parachute

A parachute can be used to slow the descent of an object experiencing free-fall. Parachutes come in many shapes and sizes and can be made for a specific payload.

Glider

The system could be placed within the fuselage of a glider. The glider would allow for the payload's descent to be controlled, either autonomously or manually to return the payload to a pre-determined recovery point on the ground.

Retro rockets

Retro rockets can be used to slow the payload before impact as is the case with the Soyuz capsule. The system would require attitude control to ensure that correct orientation was maintained for the rockets to be effective.

5.2.11 Power supply unit

Solar panels

Solar panels allow the system to use solar energy to power the system. The solar panels would charge low-capacity rechargeable batteries, which would power the system.

Batteries

Lithium-ion batteries provide a flat discharge voltage, enabling the system to run reliably for a specified amount of time.

5.3 Concept evaluation and selection

This section shows a comparison of the advantages and disadvantages of the conceptual ideas proposed in the previous section. This evaluation will aid in the selection of concepts for the functional elements of the system.

5.3.1 Micro-controller

Table I shows a comparison of the advantages and disadvantages of the micro-controllers considered for the system. Although both the Arduino and Raspberry Pi can capture video using a self operating third party camera, only the Raspberry Pi has the processing power to capture video using a dedicated camera. The Raspberry Pi has a better cost-to-performance ratio, but weighs marginally more than the equivalent Arduino board. To satisfy all the functional requirements the Raspberry Pi was chosen as the micro-controller for the system.

TABLE I

MICRO-CONTROLLER CONCEPTS EVALUATION.

Concept	Advantages	Disadvantages
Arduino	Arduino boards are relatively inexpensive. The Arduino software and hardware are open source. Light weight, and low power consumption.	Arduino boards are limited to sequential operation. Have significantly lower clock speeds. Arduino boards are not capable of capturing video.
Raspberry Pi	Raspberry Pis have significantly more RAM, storage space and computational power. They allow for multi-threaded applications, and can capture HD video with a range of cameras while performing other flight-sensitive functions.	Raspberry Pis have higher power consumption due to their increased performance. They are more expensive.

5.3.2 Control software

Table II shows a comparison of the advantages and disadvantages of the control software considered for the system. Taking into account that the Raspberry Pi was chosen as the micro-controller a concurrent approach to the control software was chosen. This decision was also based upon the requirement for multiple external devices needing to interface with the micro-controller simultaneously. It should be noted that the control software for each individual subsystem will still be executed sequentially.

TABLE II

Control software concepts evaluation.

Concept	Advantages	Disadvantages
Serial	Easy to program, and understand. Will always operate as expected.	Programs can only perform one function at a time.
Concurrent	Can simplify the separation of different processes in a system that need to be controlled simultaneously. Can improve the speed of computation and response time.	Most programming languages are inherently sequential, leaving the task of interleaving parts of the program to the programmer. Many challenges can arise, such as communication between processes, data synchronisation, and race conditions.
Parallel	Improves the performance of the program.	Not applicable to micro-controllers as they seldom have more than one processing element.

5.3.3 Communication protocol

Table III shows a comparison of the advantages and disadvantages of the communication protocols considered for the system. LoRaWAN^{TM} [15] was chosen as the communication protocol for the system to meet the functional requirements.

TABLE III

Communication protocol concepts evaluation.

Concept	Advantages	Disadvantages
RTTY	Simple and reliable technology that has been used since the 1920s for in-flight communication.	Can only transmit 32 symbols, as the messages are encoded into 5 bits.
APRS	Can transmit any form of telemetry. APRS packets can be received by anyone in range of the transmitter. Well established technology.	If packets are lost there is no method to request a re-transmit, as the communication link can only transmit messages and not receive.
LoRaWAN	Can transmit telemetry as well as SSDV images. It provides a higher bandwidth link than APRS or RTTY, or equivalently a longer range for the same bandwidth. Messages can only be decoded by another LoRa^{TM} [16] module. Can receive and transmit packets. Less expensive receiving equipment.	Is a proprietary protocol and requires specialised hardware.

5.3.4 Location sensors

Table IV shows a comparison of the advantages and disadvantages of the location sensors considered for the system. The GPS sensor was chosen as the location sensor for the system. The pressure sensor will also be used to determine altitude, and will therefore act as a backup for altitude determination in the event that the GPS subsystem fails.

TABLE IV

Location sensor concepts evaluation.

Concept	**Advantages**	**Disadvantages**
GSM	You can get a GPS fix on demand by calling the device. Low cost and no development time.	Do not work at high altitude as there is no signal from cellular towers available, making them useful at the beginning and end of the flight only.
GPS	Reliable. Trackers are available that work up to an altitude of 50 *km* ASL. Has a history of successful use in high-altitude balloon flights.	GPS can consume a large amount of power. They are susceptable to electromagnetic interference.
Iridium Tracker	Very accurate and fast.	Relatively expensive. Fail if they land upside down, as the antenna needs to be facing upwards. Only work up to 12 *km*.

5.3.5 Pressure sensor

Table V shows a comparison of the advantages and disadvantages of the pressure sensors considered for the system. The barometric sensor was chosen as the pressure sensor for the system.

TABLE V

Pressure sensor concepts evaluation.

Concept	**Advantages**	**Disadvantages**
Barometric	They are high-precision, low power consumption devices that are easily accessible. They are also small and light-weight.	Above an altitude of 9 km ASL the sensor's accuracy declines, as errors due to temperature cause fluctuations in readings.
Piezoelectric	These pressure sensors are extremely rugged, and useful at high temperatures.	They do not work with DC or steady-state conditions, as they are dependent on changes of state to generate electrical charge.

5.3.6 Temperature sensor

Table VI shows a comparison of the advantages and disadvantages of the temperature sensor considered for the system. The direct-to-digital sensor was chosen as the temperature sensor for the system.

TABLE VI

Temperature sensor concepts evaluation.

Concept	Advantages	Disadvantages
Thermistor	Very simple to implement. Relatively low cost of materials.	Non-linear, can have an exponential change in resistance with temperature, unless used in a voltage divider or Wheatstone Bridge arrangement.
Thermocouple	Very simple to implement. High speed response to changes in temperature. Widest range in temperatures of all the temperature sensors.	Would be hard to implement on the balloon experiment as it essentially measures a temperature difference between 2 points.
Direct-to-digital	Are quick and easy to implement as they only need to be incorporated into the control software. They are very accurate.	Response times to changes in temperature are not as fast.

5.3.7 Imaging sensors

Table VII shows a comparison of the advantages and disadvantages of the various imaging sensors considered for the system. The embedded camera was chosen as the imager for the system.

TABLE VII

Imaging sensor concepts evaluation.

Concept	**Advantages**	**Disadvantages**
Embedded	Fully controllable, from the size of the image or video to be taken to the timing of the footage capture. Extremely small and light-weight.	Less robust. Image quality is compromised due to sensor size.
FPV	Could potentially provide live footage if coupled to a dedicated FPV long-range transmitter. Are controllable.	Use a lot of power.
Sports	Good quality video and images. Very rugged, and can handle the temperature ranges experienced during flight.	Control using a micro-controller is not always supported. Can be heavy. Are more expensive.
DSLR	Very high quality images and video.	Very heavy and large. Cannot always be controlled externally.

5.3.8 Thermal control

Table VIII shows a comparison of the advantages and disadvantages of the thermal control techniques considered for the system. Both the styrofoam box and heating device were chosen as the thermal control for the system as they complement each other.

TABLE VIII

Thermal control concepts evaluation.

Concept	**Advantages**	**Disadvantages**
Styrofoam box	Provides thermal insulation, through better retention of heat. Provides additional impact resistance.	Does not provide heat, only allows the retention of heat.
Heating Device	Provides a source of heat.	Does not retain any of the heat that it generates.
Solar Heating	Renewable source of heat that can last for the duration of the flight.	Is heavier, and would require either multiple solar panels or attitude control to ensure the panels are directed towards the sun.

5.3.9 Release platform

Table IX shows a comparison of the advantages and disadvantages of the release platform mechanisms considered for the system. The burn wire mechanism was chosen as the release mechanism for the system as it is the simplest to implement, and the lightest.

TABLE IX

Release platform concepts evaluation.

Concept	**Advantages**	**Disadvantages**
Trap Door	Relatively simple system similar to the one used on the MIRKA2-RX mission.	Relatively heavy. May induce tumbling in capsule on release
Burn Wire	Light-weight mechanism that has been tested and used in space. Cheap and easy to implement.	Requires the MIRKA2 to hang from the bottom of the payload. The high heat generated could cause problems. Will need its own power supply with high current capabilities.
Pin and Servo	A reliable and easy method to implement.	Heavy. Would require alterations to the back plate of the MIRKA2's design.

5.3.10 Recovery

Table X shows a comparison of the advantages and disadvantages of the recovery concepts considered for the system. The parachute was chosen as the recovery medium for the system.

TABLE X

Recovery concepts evaluation.

Concept	Advantages	Disadvantages
Parachute	Reliable and easy to implement.	Crude mechanism designed to slow the descent, but will not break the fall entirely.
Glider	Allows for the payload to descend towards the chase car for easy recovery.	Very heavy system, requires dedicated flight controls. Long development lead time.
Retro Rockets	Allows smaller parachute to be used for more rapid descents.	Requires attitude control, fuel, and further research into the control of retro rockets. Very heavy system. Requires very precise altitude determination to determine when the rockets should be fired.

5.3.11 Power supply unit

Table XI shows a comparison of the advantages and disadvantages of the power supply units considered for the system. Batteries were chosen as the power supply unit for the system.

TABLE XI

Power supply concepts evaluation.

Concept	Advantages	Disadvantages
Solar	Renewable source of power that can last for the duration of the flight.	Is heavier, and would require either multiple solar panels or attitude control to ensure the panels are directed towards the sun.
Batteries	Simple and light weight.	Will run out eventually, with no way of generating power.

5.4 Flight operational flow

The operational flow control is a flight-critical capability of the release platform that will ensure the flight's success. Figure 5.2 shows a state diagram of the system throughout the flight, and depicts the operational flow under various error conditions. The flow control of the system will be controlled by the micro-controller, and will form the main thread of the system.

The system will use the altitude data collected by the GPS and pressure sensors to determine the change in vertical position of the payload. A premature balloon burst can be detected by a change in vertical velocity, more specifically the direction component of the velocity. The reason for using both metrics is to provide redundancy in case one of the individual sensors is functioning incorrectly. The system will continually cross check the sensor data, with previously collected data and individual sensor data to ensure that the data readings from the sensors are within acceptable limits.

The first phase of the flight is the execution of pre-flight system checks that will ensure that the telemetry data is being communicated as expected, and that all system components are functioning as expected. The system will then be reset and put into flight mode.

On launch all systems will be activated and a second set of checks will be conducted. This time the system will be powered by the power supply unit, and will remain powered until recovery. The thermal control heating device will be activated, and the flight system will start logging data from the sensors.

During the ascent phase of the flight the system will collect sensory data, check that the payload is still ascending and/or whether the drop altitude is reached, send telemetry, and repeat. If the payload starts descending before it reaches the burst altitude, an emergency release will be triggered. Otherwise the MIRKA2 capsule will be released normally at the predetermined drop altitude. A normal release is documented with three minutes of video footage, with the payload being released one minute after the start of recording. An emergency release will release the payload and start video footage simultaneously. After the release the payload will continue normal operation.

After the balloon bursts the recovery mechanisms will be activated. The payload will transmit more packets than on ascent to ensure a more accurate flight path is received by the ground station. Once the payload lands it will keep transmitting its final position until a response packet is sent to it from the ground station verifying the location has been received. At this point the payload will enter a sleep mode and will only transmit location based telemetry every 5 minutes.

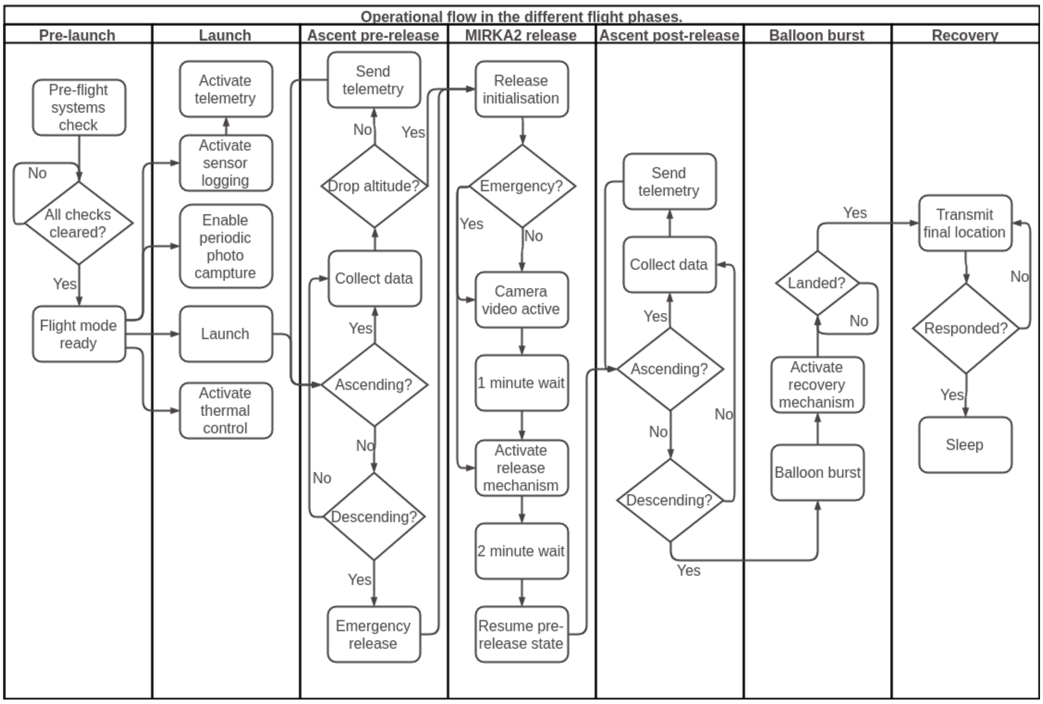

Figure 5.2: State diagram showing the operational flow control of the flight control system, with fault detection and recovery modes.

5.5 Conclusion

Figure 5.3 shows the functional elements of the system that were chosen to satisfy the functional requirements.

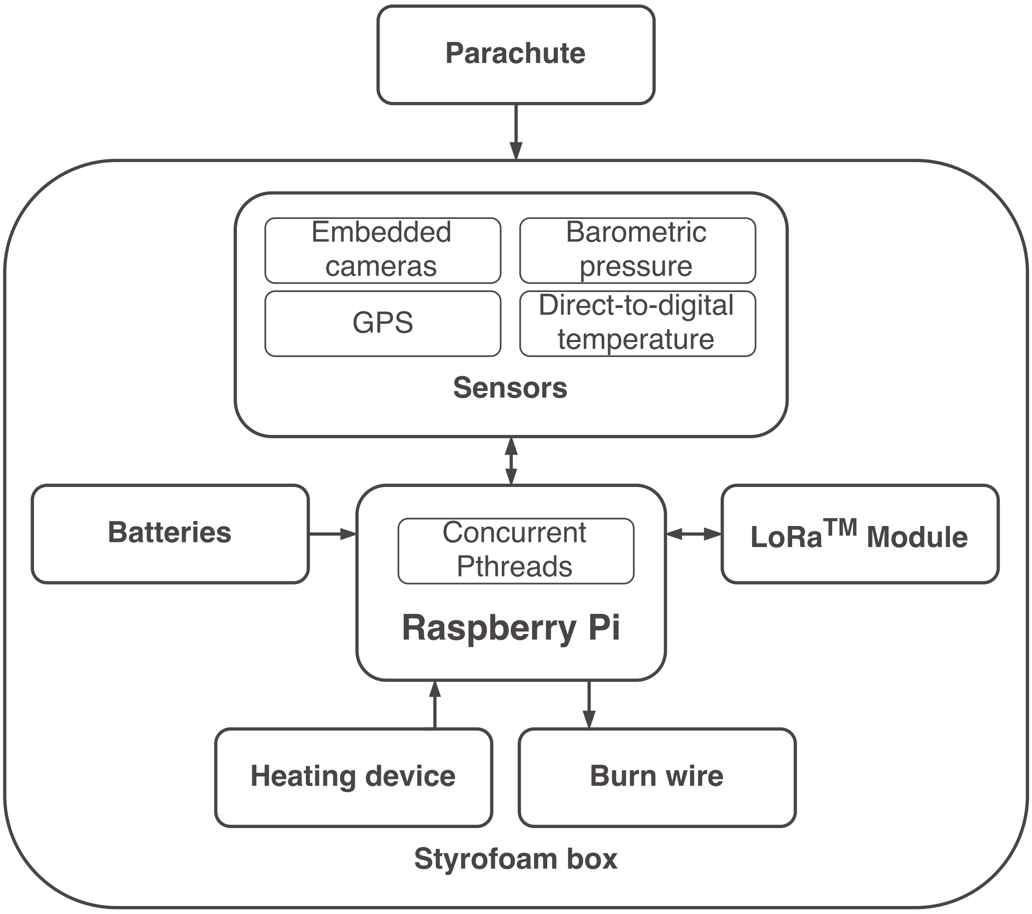

Figure 5.3: Complete concepts diagram showing the conceptual choices for the functional elements of the MIRKA2 capsule release system. The arrows pointing away from the Raspberry Pi are control signals, and the arrows pointing towards the Raspberry Pi are provision of data, power or protection.

Chapter 6

Detailed design

This chapter describes the final detailed design process of the MIRKA2 release system. The objectives of this chapter are to provide extended detail on the design's components chosen in the conceptual design phase.

6.1 Micro-controller

This section details the final design for the micro-controller of the system. The objectives of this section are to provide extended detail on the micro-controller chosen and the integration of the micro-controller within the system.

6.1.1 Detailed requirements

1. The micro-controller must provide adequate computational power, whilst maintaining low power consumption.
2. All telemetry data should be logged and stored on the micro-controller.
3. A ground station is required to demodulate the LoRaWAN [15] signal.

6.1.2 Raspberry Pi

The Raspberry Pi Zero [17] was chosen as the flight micro-controller. The Raspberry Pi A+ [18] was chosen as the ground station micro-controller.

The reasons for choosing the Raspberry Pi Zero [17] include:

- Being the lightest Raspberry Pi available;
- having the lowest power consumption;
- includes most of the core features of the other Raspberry Pis; and
- being more suited to the purpose of the flight than the other Raspberry Pis.

All Raspberry Pis have a GPIO header which allows for interfacing with external devices. Figure 6.1 [19] shows the GPIO pins with three numbering schemes, namely the physical pin numbers, the Broadcom pin numbers and the WiringPi [20] pin numbers. The Broadcom pins are the pins specified by the CPU on the Raspberry Pi Zero [17] and WiringPi [20] forms an abstraction layer that allows for easy programming of the GPIO.

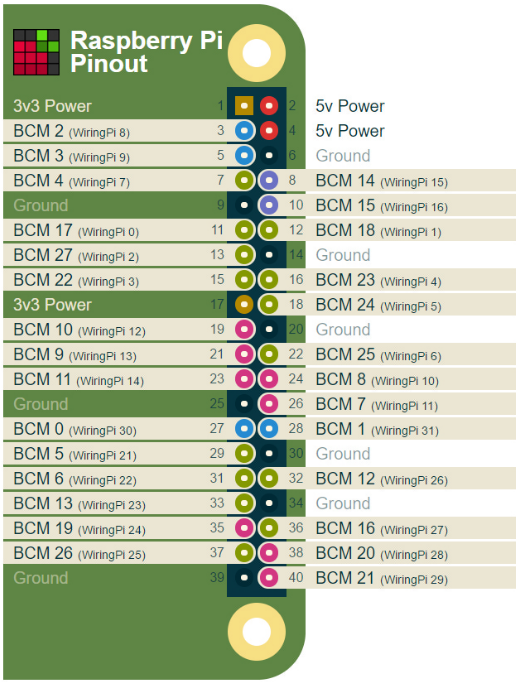

Figure 6.1: The Rapberry Pi pinout diagram, showing the location of the BCM and WiringPi numbering schemes with regards to the physical pin numbers.

Table XII lists the pins used on the two Raspberry Pis, and their designated use to be explained further in the sections detailing the devices that use the pins. The pin numbers in Table XII are the physical pin numbers. The table does not include the pins used for power.

TABLE XII

THE GPIO PINS USED ON THE FLIGHT AND GROUND STATION MICRO-CONTROLLERS, AND THEIR FUNCTIONS.

Pin	Raspberry Pi Zero	Raspberry Pi A+
3	SDA	
5	SCL	
7	ONE WIRE	
8	UBLOX RXD	
10	UBLOX TXD	
13	UBLOX SDA	
15	UBLOX SCL	
16	PHOTO	
18	VIDEO	
19	MOSI	MOSI
21	MISO	MISO
22	BURN	
23	SCLK	SCLK
26	CE1/NSS	CE1/NSS
31		LAN LED
32	CE1/DIO5	CE1/DIO5
33		INTERNET LED
35	WARN LED	
36	CE1/DIO0	CE1/DIO0
37	OK LED	CE1/DATA LED
38	GPS PPS	
40		CE1/DATA LED

6.1.3 PITS

The Pi in the Sky (PITS) is an open source project aimed at improving the state-of-the-art software and hardware for amateur high-altitude ballooning. They have developed a PCB that has been used successfully on over 70 flights. The PITS board can be mounted onto the Raspberry Pi Zero [17] GPIO using a stacking header and provides the following features:

1. Onboard 0.5 A buck/boost power regulator to provide power for an input voltage range from 1.4V to 5.5V;
2. UBlox GPS receiver [21] which is usable for altitudes up to 50 km ASL;

3. a temperature-compensated, frequency-agile, Radiometrix 434 MHz radio transmitter; and
4. a LoRa^{TM} [16] module.

6.1.4 Ground station

The ground station used to receive all telemetry communications consists of a Raspberry Pi A+ [18] and a LoRa expansion board. The ground station is connected to a PC through an ethernet port. This connection allows the ground station software to access the internet and display the groundstation user interface through an SSH connection.

6.2 Software control

This section details the final design for the software control of the system. The objectives of this section are to provide extended detail on the software control chosen and the integration of the software control within the system.

All the code for the control software is available at the Github repositories listed in Appendix B.

6.2.1 Detailed requirements

1. Multithreading is necessary for the micro-controller to interface with the many subsystems simultaneously.
2. Flow control of the flight must be implemented to handle possible fault conditions.

6.2.2 Pthreads

POSIX Threads (Pthreads) allow the programmer to create an independent stream of instructions that can be scheduled to run simultaneously and/or independently by the operating system. Pthreads are implemented using the C programming language pthread library, which defines pthreads as a set of types and procedure calls.

Pthreads are used extensively throughout the control software using a shared memory model, where each subsystem is executed on its own thread and all the threads have access to a global memory space as well as their individual memory spaces. Synchronised access to globally shared data is protected using mutexes. This approach allows the subsystems to operate simultaneously, block while waiting for I/O, respond asynchronously to events, or allow priority interrupts.

Tracker thread

The tracker thread is the main thread for the system. This is where all global variables and structs are initialised. All the other threads are configured and instantiated using this thread. This thread also gives the command to send telemetry packets, using the LoRa thread.

Pressure thread

The pressure thread is used to read the temperature and pressure from the BMP180 pressure sensor. The thread also calculates the temperature-compensated pressure value.

Camera thread

The camera thread controls the cameras throughout the flight, ensuring that the cameras function as specified in the flight operational flow section. This thread is also responsible for the control of the slave Raspberry Pi Zero [17] that controls the second camera. The slave will have a simplified version of this thread as its main program, that will take photos/videos when it receives commands through the GPIO from this thread. The slave will use the interrupt service routine (ISR) to receive commands asynchronously.

Both the camera threads will invoke shell commands to activate the camera in the background so as to prevent the control thread from being blocked for extended periods.

Temperature thread

The temperature thread controls the DS18B20 temperature sensor.

GPS thread

The GPS thread is responsible for obtaining GPS readings for the UBlox GPS receiver [21]. This thread also calculates various flight phases by using the GPS readings to determine the movement of the flight (i.e. the ascent, release, and recovery phases). The phases are used in the operational flow control of the flight.

LED thread

The LED thread updates the LEDs on the PITS board that show if the GPS signal is available. These LEDs are turned off after the flight reaches 2000 m ASL.

Log thread

The log thread simply logs the data recorded by the system to enable a review of the flight, and for development purposes.

LoRa thread

The LoRa thread handles the configuration and communication with the LoRa [16] module. The thread is responsible for packing the telemetry data required into LoRa packets, and handling the connection mode when a send request is made by the tracker thread.

Release thread

The release thread controls the release mechanism designed to release the MIRKA2 capsule. The thread is passed flight critical data from the pressure, temperature and GPS threads. The capsule release is handled by this thread under normal conditions and fault conditions.

6.2.3 Ground station

The ground station software uses the same techniques as the payload software, but is a simpler version that receives LoRa communication and provides a user interface.

6.3 Communication

This section details the final design for the communication of the system. The objectives of this section are to provide extended detail on the communication chosen and the integration of the communication within the system.

6.3.1 Detailed requirements

1. The direction and transmit power of the antenna must provide a reliable link to the ground station antenna.
2. The on-board electronics must provide reliable modulation to ensure maximum throughput to the ground station.

6.3.2 Antenna

A monopole ($\frac{1}{4}$ wave) ground-plane antenna is commonly used for high-altitude balloon flights. The antenna has a single monopole radiating element and 2 dipole ($\frac{1}{2}$ wave) radials to form the ground plane. It has an omnidirectional radiation pattern which is suitable to the use case for this project as it will provide equal transmit power in all downward directions.

The ground-plane acts as a shield and will prevent the communication system from interfering with the other electronic systems such as the GPS receiver. This will also increase the downward transmit power of the antenna as no transmit power will be wasted transmitting the signal upwards.

Figure 6.2 shows the radiation pattern of the antenna, and a model of the antenna used to simulate the radiation pattern. The positive $z-axis$ will be pointing downwards from the bottom of the payload.

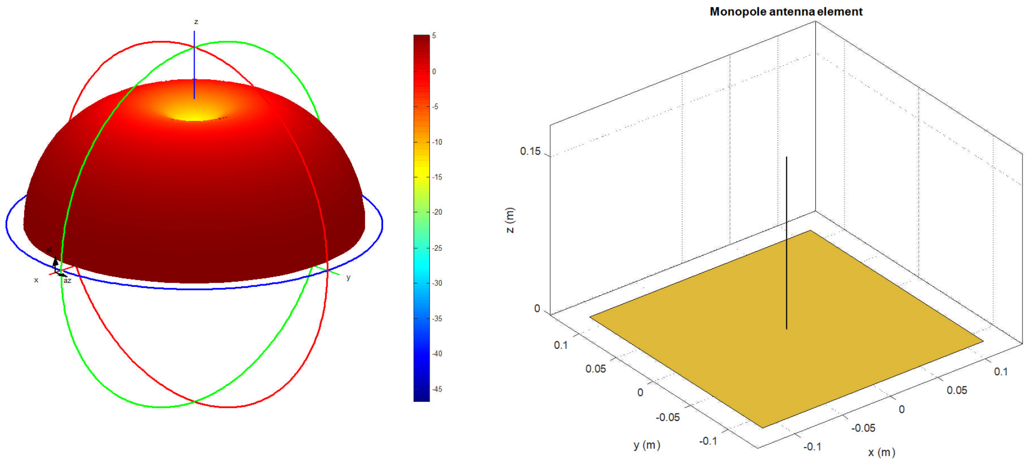

Figure 6.2: The monopole ground-plane antenna radiation pattern simulated in Matlab MathWorks, and the monopole antenna with ground plane.

The antenna is connected to the LoRa™ [16] module via a coaxial cable and SMA plug.

6.3.3 Ground station antenna

The ground station uses a 434 MHz stubby antenna with the specifications listed below.

- Gain: 2 dBi
- VSWR $<$ 1.5
- Power: 100 W
- Input Impedance: 50 Ω
- Length: 50 mm
- Polarization Type: Vertical

However, the ground station can use the same antenna specified for the payload.

6.3.4 LoRa

The LoRa^{TM} [16] module will feature on both the payload and ground station. The LoRa^{TM} [16] module will be mounted onto a PCB in both cases, the PITS board for the flight payload micro-controller and a LoRa expansion board for the ground station.

The LoRa^{TM} [16] module has a constant RF output of 20 dBm, and a high sensitivity down to -148 dBm.

6.4 Sensors

This section details the final design for the sensors of the system. The objectives of this section are to provide extended detail on the sensors chosen and the integration of the sensors within the system.

6.4.1 Detailed requirements

1. The internal and external temperatures of the payload casing must be measured.
2. The altitude of the flight must be obtained in real time.
3. The altitude of the flight must be verified by a minimum of two sources, to mitigate error.
4. Two cameras must be used to record the release of the MIRKA2 capsule.
5. Video of the release must be recorded in high-definition for later analysis.

6.4.2 Location

The UBlox GPS receiver [21] is used as the primary source of location-based data on the flight. The UBlox GPS receiver [21] is proven for altitudes up to 50 km, and is therefore a trustworthy source of flight-critical location data.

The UBlox GPS receiver [21] connects to the Raspberry Pi Zero [17] via pins 8, 10, 13, 15, and 28 (Table XII).

6.4.3 Pressure

The hypsometric formula in Equation 6.1 is used to calculate the relative altitude of a flight payload above a ground station. The formula uses the pressure P and temperature T[°C] of the atmosphere surrounding the flight payload. Here P_0 is the pressure at sea level.

$$h = \frac{((\frac{P}{P_0})^{\frac{1}{5.257}})(T + 273.15)}{0.0065} \tag{6.1}$$

After reaching 11000 m ASL altitude a temperature error will be introduced as there is a temperature inversion at this altitude. Above this the altitude Equation 6.1 can no longer be used. Temperature and pressure will be checked against a standard atmosphere [11], by allowing for a 3% tolerance for each measurement when correlating with the standard, to obtain a rough estimate of the altitude.

This rough estimate is used to check the GPS altitude readings. However, due to the nature of the inaccuracy in the calculation, the GPS readings will still be used as the golden standard in altitude measurement.

Regardless of the inaccuracy with the altitude calculation, the pressure in the atmosphere decreases with an increase in altitude. Therefore the pressure sensor can be used to determine if the flight payload is increasing or decreasing in altitude. This will help detect an early balloon burst, as the pressure will begin increasing as the payload starts to descend again.

The BMP180 [22] is the pressure sensor that will be used for the flight. The sensor has a self-contained temperature sensor for the altitude calculations.

The BMP180 [22] communicates with the Raspberry Pi Zero [17] using the I^2C pins 3 and 5 shown in Table XII.

6.4.4 Temperature

A direct-to-digital temperature sensor will be placed outside the casing of the flight payload to measure the atmospheric temperature.

The DS18B20 [23] connects to the Raspberry Pi Zero [17] via the 1-wire pin 7 shown in Table XII.

6.4.5 Imaging

Two Raspberry Pi cameras [24] will be used to capture the images and video on the flight. The cameras will connect to the CSI connectors on the Raspberry Pi Zero [17] flight and Raspberry Pi Zero [17] slave micro-controllers. The slave Raspberry Pi Zero [17] is used as a slave device to the main flight computer for the purpose of controlling the second camera, because the Raspberry Pi Zero [17] can only handle one camera at a time.

The Raspberry Pi Zeros [17] are connected via pins 16 and 18 from Table XII. The activation of photo and video footage is positive edge triggered. When pin 16 is set high by the flight Raspberry Pi Zero [17] the slave Raspberry Pi Zero [17] will capture a photo, and when pin 18 is set high by the flight Raspberry Pi Zero [17] the slave Raspberry Pi Zero [17] will capture video.

6.5 Thermal control

This section details the final design for the thermal control of the system. The objectives of this section are to provide extended detail on the thermal control chosen and the integration of the thermal control within the system.

6.5.1 Detailed requirements

1. Temperature must not be allowed to drop below -40°C inside the styrofoam box, this is the lowest operable temperature of all the payload electronics.
2. The styrofoam box must provide a housing for the electronics, and structure on which to mount the camera and mounting platform.

6.5.2 Styrofoam box

The styrofoam box will provide the main insulation from the harsh atmospheric temperatures in the Stratosphere. The styrofoam box will also provide impact protection for the electronic systems upon the return of the payload to Earth.

6.5.3 Heating device

Chemical hand-warmers, such as Grabber warmers [25], produce heat when exposed to air. The heat is produced due to a rapid oxidation process. These devices provide more than 30°C of heat, for more than seven hours of continuous heat production. The use of these devices allows the payload electronics to maintain a temperature above -40°C, when used in conjunction with the styrofoam box.

6.6 Release platform

This section details the final design for the release platform of the system. The objectives of this section are to provide extended detail on the release platform components chosen and the integration of the release platform within the system.

6.6.1 Detailed requirements

1. The release platform's mount for the MIRKA2 must provide sturdy support for the capsule and activation pins for capsule release.
2. The mounting platform must handle the temperature range of -60°C and 30°C in the atmosphere.
3. The mounting platform must weigh less than 75 g.
4. The release mechanism must operate safely, by not causing damage to the casing or compromising the other systems.
5. The release mechanism must release the payload efficiently, and must not compromise the power supply to the rest of the system.
6. The release mechanism must not take longer than 5 s to release the MIRKA2 capsule.

6.6.2 Burn wire

Nichrome wire can be used as a burn wire. The burn wire will be used to cut through a polyethylene cord which will be used to fasten the MIRKA2 capsule to the platform. The electrical resistance of a material [26] can be calculated using Equation 6.2, where ρ, L, and A are the resistivity, length, and cross sectional area of the wire respectively. The resistivity of nichrome at 20°C is $\rho = 1.5 \times 10^{-6}\ \Omega m$.

$$R = \frac{\rho L}{A} \tag{6.2}$$

However, the burn wire will not be operating at 20°C, but at -55°C. Thus we can use Equation 6.3 to calculate the resistance of nichrome at -55°C. The temperature co-efficient of resistance of nichrome is $\alpha = 0.4 \times 10^{-3}$. Substituting the resistance obtained in Equation 6.2 into R_{ref}, $T_{ref} = 20$°C, and $T = -55$°C.

$$R = R_{ref}[1 + \alpha(T - T_{ref})] \tag{6.3}$$

Using a 17 mm long 0.1mm diameter wire, a resistance of $R_{ref} = 3.25\Omega$ is obtained. Using Equation 6.3 the resistance of the nichrome burn wire at -55°C is $R = 2.27\Omega$.

Figure 6.3 shows the circuit diagram of the burn wire mechanism. For the specific use case of the burn wire, $R_W = R = 2.27\Omega$. Using a single Energizer L91 [27] Ultimate Lithium AA cell rated at a nominal voltage of 1.5 V the collector current I_c can be calculated to be 0.66 A. Using Figures 9 and 10 on the TIP122 data sheet [1], it can be seen that I_c and DC current gain β have a linear relationship for 0.1 $A < I_c <$ 2 A at -55°C, less gain is required to drive a collector current at -55°C than at 25°C, and at 25°C $I_b = 1mA$ is sufficient to saturate the transistor and drive $I_c = 2A$. Therefore, a 3.3 V GPIO pin on the Raspberry Pi Zero [17] with a max current rating of $I_b = 16\ mA$ will successfully saturate the transistor. Pin 22 from Table XII was chosen. A current limiting base resistor $R_B = 220\Omega$ is used. It should also be noted that the burn wire circuit will be inside the styrofoam box, and will not operate at -55°C, but at the temperature inside the styrofoam box. The thermal control mechanisms will regulate the temperature inside the styrofoam box to be warmer than the ambient temperature of the atmosphere at release altitude.

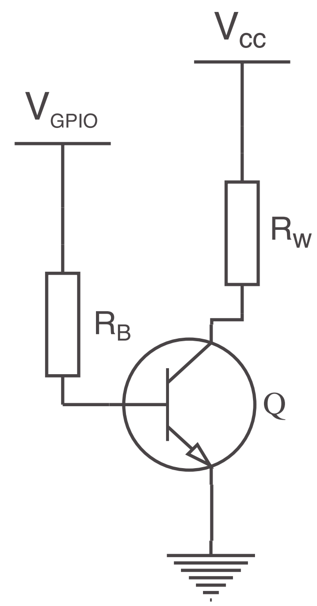

Figure 6.3: Circuit diagram showing the components in the burn wire circuit. V_{CC} = Burn wire power supply. V_{GPIO} = Control signal supplied by Raspberry Pi GPIO pin. R_W = Burn wire resistance. R_B = Current limiting base resistor. Q = TIP122 [1] (NPN) Darlington transistor.

6.6.3 MIRKA2 platform

The MIRKA2 platform was designed and modelled using Autodesk Inventor [28]. Figure 6.4 shows two orthographic views of version one of the mounting platform design. This design was 3D-printed with acrylonitrile butadiene styrene (ABS), a thermoplastic polymer, using fused deposition modelling (FDM) technology. The printing process was not suitable for such small parts, and the three locator pins were flimsy and unreliable. This prompted a second print using a different material and technology. This provided the opportunity for improvements to the design. The second version of the platform was printed using selective laser sintering (SLS) which prints polyamide (nylon). Figure 6.5 shows two orthographic views of version two of the mounting platform design.

The changes made to the design of the platform can be seen in the differences between Figure 6.4 and Figure 6.5. The alignment pins were replaced with boreholes to use metal pins, removing the reliability issues of the printed pins. The supportive structure for the polyethylene cord was raised by 1 mm to give more space for the burn wire. The centre platform diameter was increased and boreholes for the burnwire bolts were included. Initially the burn wire was going to be mounted onto a PCB in the centre, but the PCB conponents were too large. The access to the top-side of the shorting pins was opened up, for easier implementation. The centre ribs were changed to increase the strength of the structure, and various material reduction cut-outs were incorporated in the design, where possible, to reduce the weight of the platform.

Figure 6.6 shows bottom, side, and top views of version two of the platform.

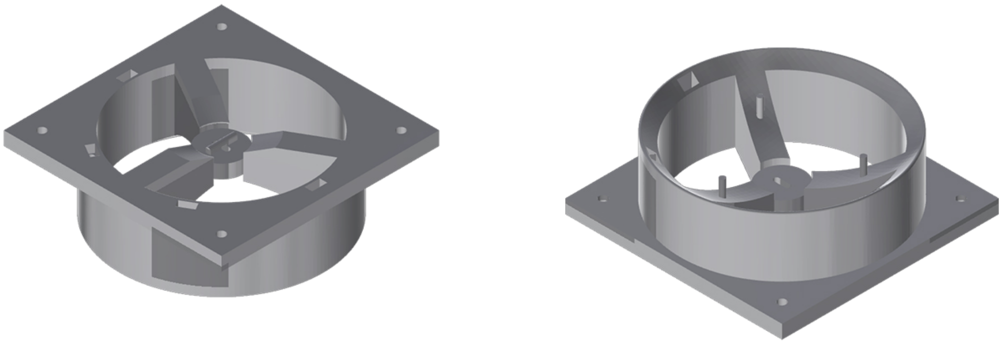

Figure 6.4: Two orthographic view angles of the MIRKA2 mounting platform version 1. The first view shows the platform oriented as it will be during the flight, and the second view shows the structure from below.

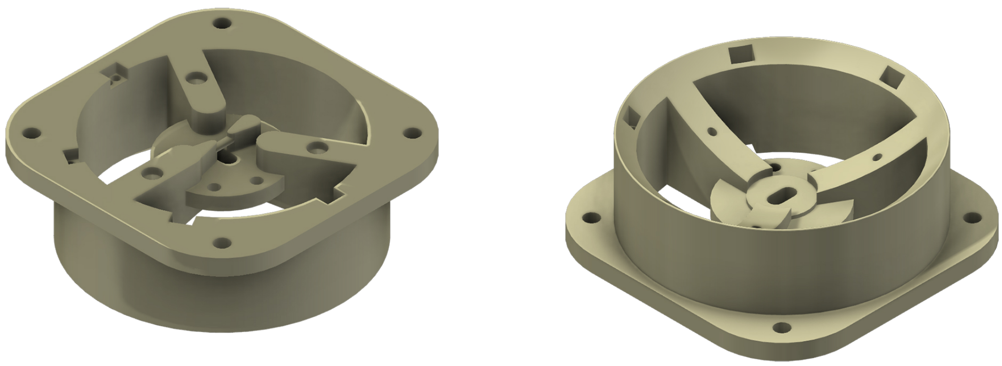

Figure 6.5: Two orthographic view angles of the MIRKA2 mounting platform version 2. The first view shows the platform oriented as it will be during the flight, and the second view shows the structure from below.

Figure 6.6: Bottom, side and top view of the MIRKA2 mounting platform version 2. From the top view the supporting structure for the polyethylene cord used to secure the MIRKA2 capsule can be seen. Three boreholes for the alignment pins, and two boreholes for the burn wire bolts can be seen in both the top and bottom views.

6.7 Recovery

This section details the final design for the recovery of the system. The objectives of this section are to provide extended detail on the recovery method chosen and the integration of the recovery method within the system.

6.7.1 Detailed requirements

1. The payload must survive the burst of the balloon.
2. The payload must survive the trip back to Earth.
3. The payload must survive the landing.
4. Telemetry data must be transmitted for the duration of the flight.

6.7.2 Parachute

The parachute for the system must produce sufficient drag to allow for a safe landing. A round 75 cm diameter parachute with a drag coefficient of 0.75 will slow the payload's descent to approximately 3.89 meters per second. The payload mass is 674 g, excluding the 660 g mass of the MIRKA2 capsule (see Table XIV). This is sufficient as the styrofoam box will provide extra padding upon landing.

6.7.3 Telemetry

Telemetry data will be transmitted at 5 s intervals after the balloon bursts. However, at an altitude of 2000 m ASL the time interval between messages will be reduced to increase the accuracy of the location data received by the ground station.

6.8 Power supply unit

This section details the final design for the power supply unit of the system. The objectives of this section are to provide extended detail on the power supply unit chosen and the integration of the power supply unit within the system.

6.8.1 Detailed requirements

1. Power to the system should be maintained for the duration of the flight, and for at least 3 $hours$ after landing.
2. The power source should provide a steady voltage so that the on-board micro-controllers maintain optimal performance.

6.8.2 Energizer L91

Energizer L91 [27] Ultimate Lithium AA cells were chosen as the power source for the flight. Each cell provides constant continuous discharge drain, giving a flat discharge profile. These cells operate reliably at lower temperatures, and have successfully been used on previous high-altitude balloon flights. Three cells will be connected in series to form a 4.5 V power supply. A voltage regulator circuit will be used to ensure that the Raspberry Pi flight computer is not damaged. This damage can be caused when the current from the power supply exceeds the maximum safe working limit of the Raspberry Pi flight computer.

A single Energizer L91 [27] Ultimate Lithium AA cell will be used to power the burn wire.

6.8.3 Power consumption

Table XIII shows the milliamp-hour capacity of the system required to power the electronic systems for a six-hour flight. These calculations are done assuming that everything is running for the full six hours at the maximum current ratings for each device, and are therefore worst-case calculations. The cameras will only be in operation to take photos periodically every 2 minutes at 2 seconds operation time per photo, and three minutes of video at release. The Raspberry Pi Zero [17] controlling the camera will be idle for the period when its camera is not operating.

According to the Energizer L91 [27] Ultimate Lithium AA cell's data-sheet a single cell has a milliamp-hour capacity of 3250 mAh for a continuous discharge drain of 1000 mA at 21°C. However, for a constant current discharge of 1000 mA at the maintained lowest temperature of -20°C the capacity is reduced to 2600 mAh; the capacity increases with a decrease in current discharge and increase in temperature. The worst case total capacity required by the system is shown as 2376.03 mAh and fits within the bounds of the worst case single cell's capacity. The reason for considering only one cell's characteristics in determining capacity is because capacity only increases when cells are connected in parallel, not series.

TABLE XIII

MAX CURRENT DRAWN AND mAh CAPACITY REQUIRED.

Component	**Current** [mA]	**Time** [h]	**Capacity** [mAh]
Raspberry Pi Zero 1	150	6	900
Raspberry Pi Zero 2	150	0.15	22.5
	70	5.85	409.5
Camera x2	250	0.15	37.5
DS18B20	1.5	6	9
BMP180	5×10^{-3}	6	0.03
UBlox	70	6	420
LoRa	90	6	540
Total	961.505		2376.03

6.9 System

This section details the final design for the system. The objectives of this section are to provide extended detail on the micro-controller chosen and the integration of the micro-controller within the system.

6.9.1 Detailed requirements

1. The mass of the system must be less than 1500 g.

6.9.2 Mass calculations

Table XIV shows the mass in grams of all the system components. The miscellaneous component includes the antenna, mounting screws, and polyethylene cord used to attach the balloon to the platform.

TABLE XIV

MASSES OF THE SYSTEM COMPONENTS.

Component	**Mass** [g]
Raspberry Pi Zero	9
PITS Zero	60
Camera x2	6
DS18B20	9
BMP180	3
Cells	60
MIRKA2	660
MIRKA2 mount	54
Parachute	75
Hand Warmers	60
Styrofoam box	150
Miscellaneous	200
Total	1334

6.10 System integration

Figure 6.7 shows a block diagram of the integrated subsystems that satisfy the functional and detailed requirements.

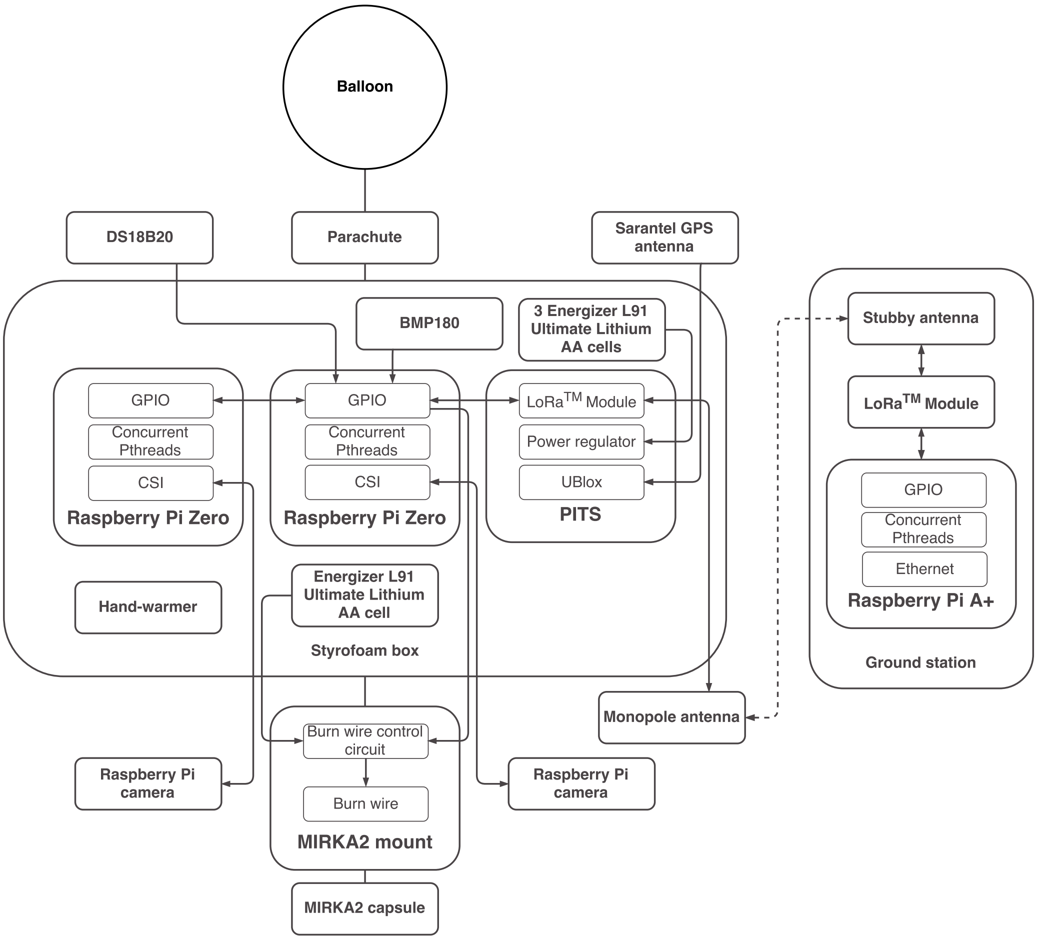

Figure 6.7: Block diagram showing the integrated subsystems as specified in the detailed design. The solid arrows specify control, data, and/or power signals. The dashed arrow specifies the communication link between the payload and ground station. The line connectors without arrow heads specify a mechanical connection.

Chapter 7

Assembly process and unit testing

This chapter details the assembly process and unit testing of the MIRKA2 release system. The objectives of this chapter are to provide detail on the methods used to fabricate and test the individual system components specified in the detailed design phase.

7.1 Release platform

The release platform was modelled using Autodesk Inventor [28]. This software was used to conduct a stress analysis of the design, and produce the necessary files to 3D-print the platform. The simulated stress analyses of the two versions can be accessed on the MIRKA2 Release Experiment landing page [5]. The final version of the platform was then printed using SLS polyamide. Figure 7.1 shows the final assembly of the platform version 2 with the fastening bolts, pins and MIRKA2 capsule. The mounts for the burn wire include two nuts clamping the wires between washers. The control wires are also clamped by the washers, with the rest of the control circuit inside the styrofoam box. The long bolts will be be used to fix the platform to the styrofoam box.

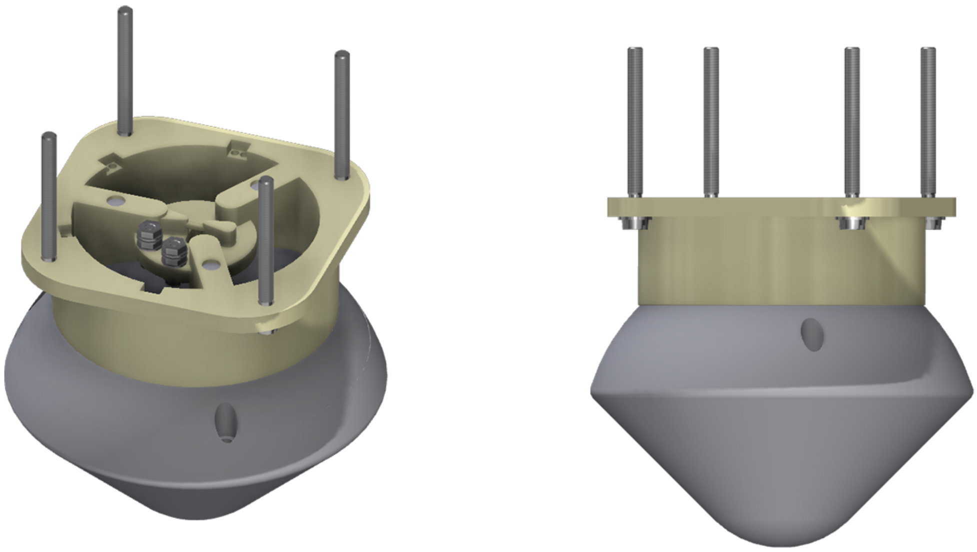

Figure 7.1: Full assembly showing the release platform with fastening bolts, pins and the MIRKA2 capsule.

7.2 Payload antenna

The UKHAS Wiki Payload antenna guide [29] was used as a guideline for the fabrication of the necessary monopole ground plane antenna.

7.3 Micro-controller

The following section describes the setup process for the Raspberry Pi Zero [17] to be used as the payload micro-controller.

7.3.1 Raspbian configuration

Raspbian Jessie Lite [13] was installed on the SD card, to be inserted into the Raspberry Pi Zero [17]. The Raspbian Jessie Lite image [13] was flashed onto the SD card using Etcher [30]. After the installation of Raspbian [13] a keyboard and monitor were attached to the Raspberry Pi Zero [17] for initial configuration. The Raspberry Pi Zero [17] was booted and logged into using ‘pi’ and ‘raspberry’ as the login username and password respectively. Listing 7.1 displays the command used to enter configuration mode for Raspbian [13]. In ‘Interfacing Options’ The following were enabled:

- Camera
- SSH (Allows connection over a USB wired LAN adapter)
- SPI
- I^2C
- 1-Wire support

Listing 7.1: Raspbian configuration command.

```
sudo raspi-config
```

After this completed the Raspberry Pi Zero [17] rebooted. The USB wired LAN adaptor was used from this point onwards to connect through an SSH link for programming the Raspberry Pi Zero [17]. The same login details apply.

7.3.2 Serial port

The serial port needed to be re-purposed to access the GPIO pins (instead of bluetooth using those specific GPIO pins). Listing 7.2 displays the commands used to disable the serial port login and to prevent the kernel from using the serial port. `console=serial0,115200` was removed from `/boot/cmdline.txt` while in nano [31].

Listing 7.2: Disable serial port commands.

```
sudo systemctl mask serial-getty@ttyAMA0.service
sudo nano /boot/cmdline.txt
```

Listing 7.3 was executed, adding `dtoverlay=pi3-disable-bt` to the end of `/boot/config.txt` while in nano [31].

Listing 7.3: Remap serial port commands.

```
sudo nano /boot/config.txt
sudo systemctl disable hciuart
```

7.3.3 Git

Git [32] was installed to allow the installation and updating of various programs. Listing 7.4 gives the command used.

Listing 7.4: Git installation command.

```
sudo apt-get install git
```

7.3.4 PIGPIO

PIGPIO [33] was installed to give access to development files for compilation. Listing 7.5 gives the commands used to install the development drivers.

Listing 7.5: PIGPIO installation commands.

```
cd ~
wget abyz.co.uk/rpi/pigpio/pigpio.zip
unzip pigpio.zip
cd ~/PIGPIO
make
sudo make install
```

7.3.5 Wiring Pi

Wiring Pi [20] is a library used for interfacing with the GPIO pins and is installed using the command in Listing 7.6

Listing 7.6: Wiring Pi installation commands.

```
cd ~
git clone git://git.drogon.net/wiringPi
cd ~/wiringPi
./build
```

7.3.6 SSDV

Slow scan digital video (SSDV) allows for the conversion between JPG and SSDV image formats. SSDV is a better format for transmitting images over an unreliable link. SSDV [34] was installed using the commands in Listing 7.7.

Listing 7.7: SSDV installation commands.

```
cd ~
git clone https://github.com/fsphil/ssdv.git
cd ~/ssdv
sudo make install
```

7.3.7 Control software

The PITS [35] tracker software is open source software that is available to use with the Pi in the Sky board. This software was altered and built upon such that the software performed as designed in the detailed design. Listing 7.8 shows the commands used to install the control software. The build process compiles and links the tracker program, creates a configuration file, and sets the software up to start automatically when the Raspberry Pi boots.

Listing 7.8: Control software installation commands.

```
cd ~
git clone https://github.com/cairinmichie/pits.git
cd ~/pits
./build
```

7.3.8 Software update

To download and build the latest software, the commands in Listing 7.9 were used. This method was used primarily during development, but can be used to set up new devices. This will update the software to the latest version in the repository.

Listing 7.9: Git pull command used to upate the tracker software.

```
cd ~/pits
git pull origin master
cd ~/tracker
make
```

7.3.9 Testing

Listing 7.10 shows commands that can be used to restart the tracker program.

Listing 7.10: Commands used to restart the tracking program.

```
sudo killall startup
sudo killall tracker
cd ~/pits/tracker
sudo ./tracker
```

The testing process was used during the development of the control software and hardware. The testing process comprised the following steps:

1. Kill the tracker program using the first two commands in Listing 7.10;
2. Update the software and remove power;
3. Add or remove hardware as needed;
4. Connect power and boot;

Additionally, on boot the tracker would blink the red WARN LED. This kept blinking until the GPS subsystem had connected to four or more satellites, at which point the green OK LED started blinking.

The LoRa connection tests were done after the setup of the base station.

7.4 Camera software

The following section describes the setup process for the second Raspberry Pi Zero [17] to be used as the micro-controller for the second camera. The following installation steps are repeated from section 7.3:

- Raspbian Configuration

- Wiring Pi
- Software Update (for camera software)

The camera software for the second Raspberry Pi Zero [17] was installed using the commands in Listing 7.11

Listing 7.11: pits-camera installation commands.

```
cd ~
git clone https://github.com/cairinmichie/pits-camera.git
cd ~/pits-camera
./build
```

7.5 LoRa base station

The following section describes the setup process for the Raspberry Pi A+ [18] to be used as the base station receiver. The following installation steps are repeated from section 7.3:

- Raspbian Configuration
- Wiring Pi
- Software Update (for base station software)

7.5.1 Dependencies

Certain dependencies are needed for the base station software. The curl library is used for internet connection, and the ncurses library for the screen display. These are installed using the commands in Listing 7.12.

Listing 7.12: Dependency installation commands.

```
sudo apt-get install libcurl4-openssl-dev
sudo apt-get install libncurses5-dev
```

7.5.2 Gateway software

The lora-gateway [36] software was forked and altered to achieve the goals of the project. The forked repository was then used to install the base station control software via the commands in Listing 7.13.

Listing 7.13: Base station control software installation commands.

```
cd ~
git clone https://github.com/cairinmichie/lora-gateway.git
cd ~/lora-gateway
make
```

7.5.3 Configuration

The file `gateway.txt` contains the configuration settings of the LoRa^{TM} module [16] on the base station.

7.5.4 Tests

To test the gateway and the communication link, both the Raspberry Pi Zero [17] and the Raspberry Pi A+ [18] were configured to use LoRa frequency = 434.275 MHz and LoRa mode 1. Figure 7.2 shows the results obtained on the base station for a basic communication test. This test was run indoors in a building with corrugated metal roofing, which prevented the GPS sensor from finding satellites.

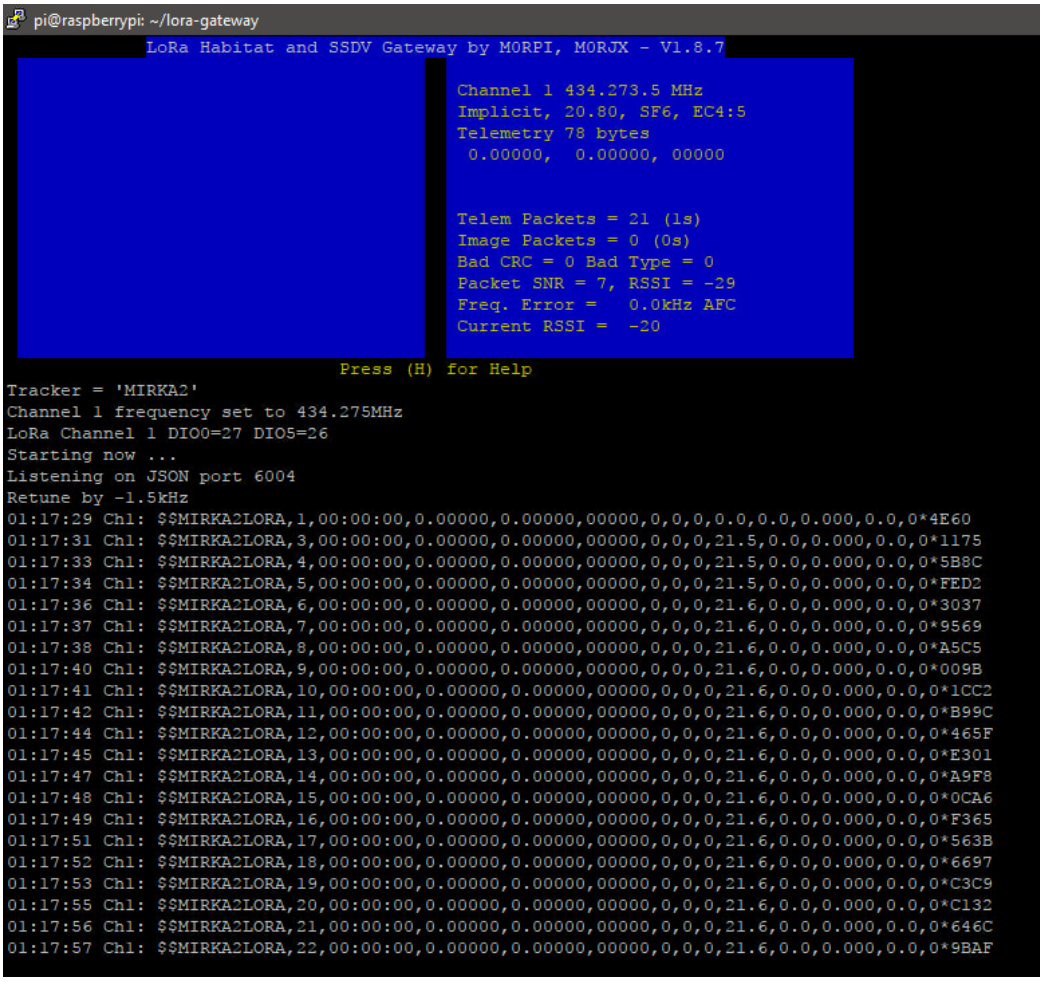

Figure 7.2: The working base station test, showing incoming packets from the flight micro-controller. The base station is using LoRa channel 1. There had been 1 s since the last packet had arrived when the screenshot was taken. The signal-to-noise ratio was 7. No packets were lost. THe received signal strength indicator (RSSI) has units dB.

7.6 Burn wire

The burn wire circuit was first tested in the laboratory to ensure that it worked as expected. This test was done on a breadboard, so that components could be swapped out easily if needed. The burn wire succeeded in cutting through a piece of polyethylene cord consistently. The time taken to cut through the cord varied slightly but never exceeded the 5 s requirement.

Chapter 8

Integrated system testing

This chapter details the integrated system testing of the MIRKA2 release platform. The objectives of this chapter are to provide detail on the testing processes used to ensure that the system specified in the detailed design will work according to the requirements.

8.1 Telemetry

This section details the tests run on the communication and sensor systems.

8.1.1 Battery test

The battery test was used to determine how long the single Raspberry Pi Zero [17] and PITS board could operate under battery power, while collecting and transmitting telemetry data.

The test was accomplished by setting up both the base station and the payload micro-controllers only, and running them until the power supply to the payload electronics failed. This test was not run using any of the sensors. The test was run outside, so that the GPS signal could be received.

The test started at 13h15 and ran until the system ceased communications due to power supply failure of the payload electronics at 8h45 the following morning, giving a system up-time of 19 hours and 30 minutes.

8.1.2 Integrated test

The integrated test was completed with the final electronic payload as stipulated in the detailed design chapter. This test was implemented in the same manner that the battery test was, albeit with all the sensors, thus testing both the sensors' functionality and the system's ability to run multiple sensors concurrently.

The test started at 15h58 and ran until the system ceased communications due to power supply failure of the payload electronics at 23h28 the same day, giving a system up-time of 7 hours and 30 minutes.

8.1.3 Integrated freezer test

The integrated freezer test was implemented in the same manner that the integrated test was, albeit in a freezer, thus testing the payload electronics and power supply at sub-zero temperatures. The ambient temperature inside the freezer was set to -19.5°C (measured by the DS18B20 temperature sensor). The payload electronics were placed into a plastic box which protected the electronics from moisture damage and provided minimal insulation.

The communication link was not affected by the freezer, and the same SNR and RSSI were achieved as the other tests throughout this test. However, out of the 22644 packets received by the ground station 43 packets had CRC failure. The Raspberry Pi Zero [17] reached a minimum temperature of -12.6°C (measured by the Raspberry Pi Zeros [17] on-board temperature sensor). The BMP180 pressure sensor reached a minimum temperature of -17.6°C, and measured a constant pressure of 102.4 kPa. The test started at 13h29 and ran until the system ceased communications due to power supply failure of the payload electronics at 21h55 the same day, giving a system up-time of 8 hours and 26 minutes.

8.1.4 Communication link

To test the communication link the Raspberry Pi Zero [17] flight micro-controller was placed at the top of Lion's Head. To maintain line of sight as much as possible the ground station micro-controller was kept at a location near the base of Lion's Head. Figure 8.1 shows the line of sight path between the ground station and the flight micro-controller at the top of Lion's Head. The path was generated using a python script [37].

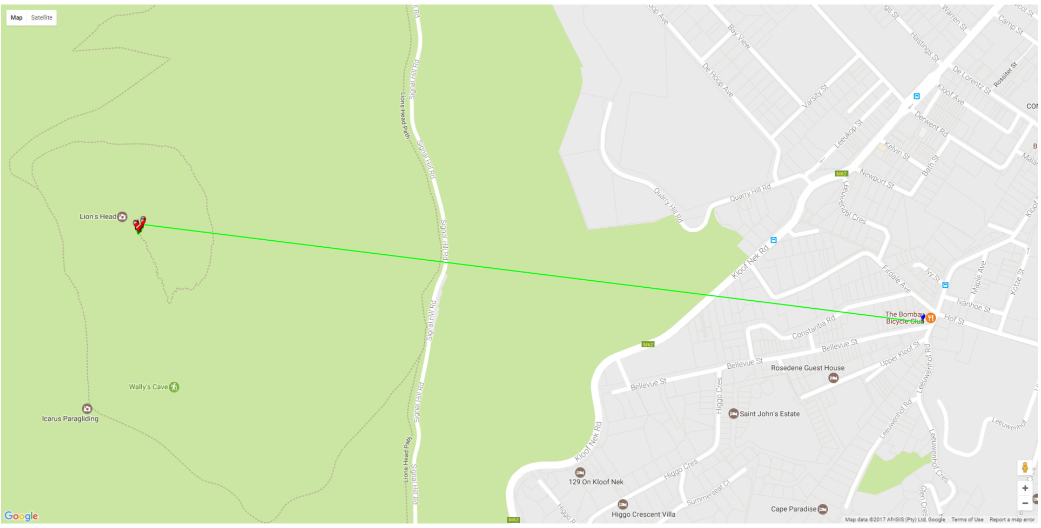

Figure 8.1: The line of sight path between the ground station, and the flight micro-controller at the top of Lion's Head. The blue marker denotes the position of the ground station, and the red markers denote GPS coordinates sent by the flight micro-controller as received by the ground station.

The article *To Zero, and Beyond* [38] written by Dave Ackerman (one of the developers at Pi in the Sky) shows the PITS Zero board transmitting an image from an altitude of 41837 m ASL. The communication system used on the flight payload in the article was the same as the system designed in this report, further verifying the capabilities of the communication system.

8.2 Release

To test the release platform the back shell of the capsule was 3D-printed, and a weight was tied to it to ensure that the mass was equivalent to that of the MIRKA2 capsule. This weight was then mounted to the release platform with the burn wire to simulate flight conditions, thus testing both the strength of the release platform and the burn wire circuit. The control software was altered to activate the burn wire at a specified time, for the duration necessary to cut through the polyethylene cord. This test verified proper operation of the release mechanism when activated by the GPIO pin of the flight controller. The test was run a total of 10 times, and was successful every time.

Further testing on the strength of release platform was performed to validate the simulation results. This was achieved by increasing the weight attached to the release platform. The weight was increased in increments, and the maximum weight tested was double the weight of the MIRKA2 capsule. The release platform was left for 4 hours to simulate worst case flight conditions. There was no visible deformation or stress on the release platform during and after the test.

Chapter 9

Conclusions and recommendations

This chapter details the conclusions and recommendations of the design of the MIRKA2 release platform. The objectives of this chapter are to provide analysis of the design, and give recommendations for future improvements.

9.1 System design

During the design process, many design choices were made that ultimately led to the final design. This section analyses the various design decisions made, and provides recommendations for future work regarding the design of the system.

9.1.1 Micro-controller

The micro-controllers chosen for the system provided adequate computational power, and maintained low power consumption during the testing. They provided a reliable platform upon which the rest of the system could depend upon. As all the requirements for this subsystem were met, it is not recommended that the design decisions regarding the micro-controllers be changed.

9.1.2 Control software

The control software was implemented as a multithreaded program that provides an interface for the sensors of the system. The flight control software is functional and meets all the requirements that were set out for it.

The control software could be expanded by allowing for the transmission of SSDV images as a verification of the release of the MIRKA2 capsule. Better altitude determination algorithms could be developed to improve the trackability, and therefore the safety of the flight. Full communication between the two payload Raspberry Pi Zeros [17] could be established, allowing for the sensors to be more evenly distributed between the micro-controllers. Although these extensions would improve the flight control quality, the success of the flight is not dependent on them.

9.1.3 Ground station

The ground station software in its current state is very simple. It displays only flight-critical data in a very compact format. If the ground station is to be used to its full potential, it is recommended that it is expanded by uploading the telemetry data to a server to create a real-time sensor dashboard, and plotting the location data received from the flight micro-controller onto a map in real time. These two improvements will provide an easier way to view data as it is received, and will allow for multiple chase vehicles as the flight data will be available over an internet connection (where wireless signal is available, or if a satellite terminal is used).

9.1.4 Communication

The communication technique chosen in the design process provides a reliable link over which all telemetry can be transmitted. The LoRaWAN [15] network protocol allows for a robust system that ensures the longest range and maximum throughput.

Currently the system uses a duplex connection to allow the ground station to request re-sending of lost packets from the flight micro-controller. It is recommended that a control protocol should be implemented to allow for an override of the release mechanism from the ground station. This will provide an extra safety mechanism to ensure the success of the flight, in the event of unforeseen faults, or in the event that the balloon starts to drift toward an area over which it would be unsafe to release the capsule.

9.1.5 Sensors

The sensor systems allow for measuring various atmospheric data such as internal and external temperatures and the ambient air pressure. The system provides a metric for altitude, and provides a fail-safe for false readings regarding altitude. The sensors provides a means to video the release of the MIRKA2 capsule. Great care was taken in the selection of the sensors, to provide maximum accuracy without compromising mass, or cost.

The sensors meet all the requirements of the system. It is recommended that a real-time overlay of the telemetry and sensory data should be created for the video footage. This would allow for easy identification of the system conditions at release. Alternatively the telemetry data could be added to the meta-data of the video file.

9.1.6 Thermal control

The thermal control of the system is adequate, and will maintain operational temperatures for the duration of the flight. The use of hand warmers placed inside an insulating styrofoam box has worked successfully on previous high-altitude balloon flights.

9.1.7 Release platform

The release platform formed an integral part of the design. The platform was carefully designed to meet the strict mass and strength specifications to support the MIRKA2 capsule. The stress analysis and testing of the platform show that the materials chosen in the final design worked as simulated. The manufacturing process allowed for the design to be small and intricate whilst maintaining the necessary structural integrity. The burn wire mechanism was designed to be as simple as possible to reduce risk of failure. The release mechanism was designed to operate separately from the rest of the system, so as not to compromise the power supply to the micro-controllers.

9.1.8 Power supply unit

The power supply was able to provide adequate power to the system for the required length of time and was suitable for use within the strict requirements of the system.

9.1.9 Flow control

The flow control methods specified provide adequate fault correction to satisfy the functional requirements of the system. However, in the event the balloon does not burst the system will not be recoverable. It is recommended that a second burn wire should be placed above the parachute to ensure that the system would be recoverable in the event the balloon does not burst.

9.1.10 System

The integrated system fulfils all the functional and detailed requirements set out in this report. The bill of materials in Appendix A shows all the parts used for the project and their associated costs. Photos of various parts are available on the MIRKA2 Release Experiment landing page [5].

9.2 Testing

During the testing process, the design was assembled, integrated, and tested. This section analyses the various tests, and provides recommendations for future work regarding the testing of the system.

9.2.1 Telemetry

The integrated freezer test verified that the payload electronics function as expected at temperatures as low as -19.5°C. The thermal control methods detailed during the design process will maintain an ambient temperature close to the tested temperature at the coldest point during the flight.

It is recommended that the full system, with insulation and heating devices, is tested in an industrial freezer to maintain temperatures closer to -60°C, thus testing the worst case atmospheric conditions.

9.2.2 Release

The release tests verified the correct functioning of the burn wire, and the strength of the release platform.

However these tests were not completed at low temperatures, therefore it is recommended that the release system should be tested at the low temperatures found in the upper atmosphere.

9.3 Flight

This section provides recommendations regarding the flight of the high-altitude balloon.

9.3.1 ATNS

Correspondence with air traffic and navigation services (ATNS) is recommended to ensure that the flight falls within the regulations for high-altitude balloons. This will help in the choice of launch position, ensuring that the balloon does not enter the flight paths of other vehicles.

9.3.2 Flight location

It is recommended that the balloon should be launched from an isolated area. This will reduce the chance of injury or damage incurred by the MIRKA2 capsule or the flight payload when they return to Earth. Local authorities should be alerted to the flight in a timely manner so that any people in the landing areas can be forewarned.

References

[1] “Plastic Medium-Power Complementary Silicon Transistors,” onsemi.com/pub/Collateral/TIP120-D.PDF.

[2] KSat e.V., “MIRKA2-ICV Experiment Documentation,” 2017, cairinmichie.github.io/MIRKA-release/Documents/MIRKA2-ICV-Documentation.pdf.

[3] T. Abe, T. Imamura, N. Izutsu, and N. Yajima, *Scientific Ballooning Technology and Applications of Exploration Balloons Floating in the Stratosphere and the Atmosphere of Other Planets.* Springer Science & Business Media, 2009, ch. 4, pp. 84–86.

[4] “Iridium Satellite Communications — Home,” iridium.com.

[5] “MIRKA2 Release Landing Page by cairinmichie,” cairinmichie.github.io/MIRKA-release.

[6] “CUSF Landing Predictor 2.5,” predict.habhub.org.

[7] “Habitat Burst Calculator,” habhub.org/calc.

[8] “Parachute Size Calculator,” rocketreviews.com/parachute-size-calculator.

[9] “Descent Rate Calculator,” rocketreviews.com/descent-rate-calculator.

[10] “Ventusky - Wind, Rain and Temperature Maps,” ventusky.com.

[11] “1976 Standard Atmosphere Calculator,” digitaldutch.com/atmoscalc/tableatmosphere.

[12] “Arduino - Reference,” arduino.cc/en/Reference/HomePage.

[13] “Download Rasbian for Raspberry Pi,” raspberrypi.org/downloads/raspbian.

[14] “Debian - The Universal Operating System,” debian.org/.

[15] “LoRa™ Technology,” lora-alliance.org/What-Is-LoRa/Technology.

[16] “RFM95/96/97/98(W) - Low Power Long Range Transceiver Module ,” hoperf.com/upload/rf/RFM95_96_97_98W.pdf.

[17] “Raspberry Pi Zero - Raspberry Pi,” raspberrypi.org/products/pi-zero.

[18] “Raspberry Pi 1 Model A+,” raspberrypi.org/products/model-a-plus.

[19] “Raspberry Pi GPIO Pinout,” pinout.xyz/pinout/wiringpi.

[20] “WiringPi,” wiringpi.com.

[21] “MAX-M8,” u-blox.com/sites/default/files/MAX-M8-FW3_DataSheet_%28UBX-15031506%29.pdf.

[22] “BMP180 Digital pressure sensor,” ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP180-DS000-121.pdf.

[23] “DS18B20 Programmable Resolution 1-Wire Digital Thermometer,” datasheets.maximintegrated.com/en/ds/DS18B20.pdf.

[24] “Camera Module - Raspberry Pi,” raspberrypi.org/products/camera-module-v2.

[25] “Grabber warmers,” grabberwarmers.com.

[26] R. D. Knight, *Physics For Scientists and Engineers*, 3rd ed. Pearson, 2013, ch. 30, pp. 879–880.

[27] “Energizer Lithium Ultimate Datasheet,” cairinmichie.github.io/MIRKA-release/Documents/Energiser-L91.pdf.

[28] “Inventor — Mechanical Design & 3D CAD Software — Autodesk,” autodesk.com/products/inventor/overview.

[29] “guides:payload_antenna,” ukhas.org.uk/guides:payload_antenna.

[30] “Etcher,” etcher.io.

[31] “GNU nano,” nano-editor.org.

[32] “Git,” git-scm.com.

[33] “pigpio library,” abyz.co.uk/rpi/pigpio.

[34] “SSDV,” github.com/fsphil/ssdv, a robust version of the JPEG image format, for transmission over an unreliable medium.

[35] “Pi in the Sky Telemetry Software,” github.com/PiInTheSky/pits.

[36] “LoRa-Gateway Software,” github.com/PiInTheSky/lora-gateway.

[37] “Python mapping software,” github.com/cairinmichie/pits-mapping.git.

[38] “To Zero, And Beyond,” daveakerman.com/?p=1943.

Appendix A

Bill of materials

TABLE XV

Bill of materials, with costs.

Item	**Quantity**	**Total Cost** $[R]$
Antenna	2	0.00
BMP180	1	158.00
Burn wire	1	75.10
DS18B20	1	195.00
Energizer L91	4	299.80
Handwarmers	2	100.00
LoRa module	1	431.66
Nuts and bolts	1	75.50
Parachute	1	141.64
Pi camera	2	1041.62
PITS board	1	3925.69
Polyethylene	1	70.00
Raspberry Pi A+	1	392.41
Raspberry Pi Zero	1	79.80
Release platform	1	1055.87
Styrofoam box	1	39.95
Total		8082.04

Appendix B

Github code repositories

github.com/cairinmichie/pits.git

github.com/cairinmichie/pits-camera.git

github.com/cairinmichie/lora-gateway.git

github.com/cairinmichie/pits-mapping.git