SUNSET

Stellenbosch University Near-space Engineering Testbed

Alexander Olaf Erlank
15289958

STUDY LEADER: Arno Barnard

October 2011

Report submitted in partial fulfilment of the requirements of the module Project (E) 448 for the degree Baccalaureus in Engineering in the Department of Electrical and Electronic Engineering at the University of Stellenbosch

Acknowledgements

Arno Barnard for his guidance and support and willingness to give up his Saturdays to help with tests

Leanne and Pam Pienaar for their excellent sewing

JP Meijers for lending me his amateur radio equipment

Everyone who volunteered to help on launch day.

Declaration

I, the undersigned, hereby declare that the work contained in this report is my own original work unless indicated otherwise

Signature Date

Abstract

Satellites provide many valuable services, but the cost of getting hardware into space can be prohibitive. A platform taken into the upper atmosphere by a Hydrogen filled balloon has the potential to provide many of the same services and data, at a fraction of the cost. This paper describes the design, implementation and testing of such a platform, named the Stellenbosch University Near-space Engineering Testbed. The SUNSET payload was designed to be low cost, lightweight, expandable and reusable. A Parallax Propeller microcontroller acts as the flight computer and is interfaced to a GPS sensor, a GSM modem, a VHF radio transmitter, an SD card, a camera and two temperature sensors. The SUNSET payload makes use of the amateur radio frequencies and the Automatic Position Reporting Service (APRS) to transmit telemetry and live images to the ground station during its flight. Custom ground station software to display the data from the SUNSET payload was also developed. An airframe was designed and manufactured to thermally insulate the electronics, and to protect them during the payload's fall back to earth. The completed payload was proven to be flight ready by several tests and two tether flights.

Satelliete verskaf baie waardevolle dienste, maar die koste om n satellite in die ruimte te kry is baie hoog. 'n platform wat tot in die die boonste atmosfeer geneem word deur' n waterstof gevulde ballon het die potensiaal om baie van dieselfde dienste en inligting te verskaf, teen 'n baie laer prys. Hierdie verslag beskryf die ontwerp, implementering en toetsing van so 'n platform. Die platform is SUNSET (Stellenbosch University Near-space Engineering Testbed) genoem. SUNSET is ontwerp om goedkoop, liggewig, uitbreibaar en herbruikbaar te wees. 'n Parallax Propeller mikroverwerker tree op as die vlug rekenaar en is gekoppel aan 'n GPS sensor, 'n GSM modem, 'n VHF radio sender, 'n SD-kaart, 'n kamera en twee temperatuur sensore. SUNSET maak gebruik van die amateur-radio frekwensies en APRS (Automatic Position Reporting Service) om telemetrie en kamera beelde aan die grond stasie te stuur, tydens sy vlug. Grondstasie sagteware om SUNSET se data te vertoon is ook ontwikkel. 'n liggewig struktuur was ontwerp en vervaardig om die elektronika termies te isoleer en om die elektronika te beskerm tydens SUNSET se val terug aarde toe. Deur toetse was dit bewys dat SUNSET vlug gereed is.

Acronyms

SUNSET	Stellenbosch University Near-space Engineering Testbed
ARHAB	Amateur Radio High Altitude Ballooning
GPS	Global Positioning System
VHF	Very High Frequency
UHF	Ultra High Frequency
HF	High Frequency
APRS	Automatic Position Reporting System
RISC	Reduced Instruction Set Computer
ARM	Advanced RISC Machine
CHDK	Canon Hack Development Kit
AGL	Above Ground Level
ASL	Above Sea Level
SMS	Simple Message Service
NiMH	Nickel-metal hydride
NiCd	Nickel-cadmium
ATV	Amateur Television
GSM	Global System for Mobile Communications
MIT	Massachusetts Institute of Technology
A/D	Dnalogue to Digital
ADC	Analoge to Digital Converter
UART	Universal Asynchronous Receiver/Transmitter
CPU	Central Processing Unit
IDE	Integrated Development Environment
RAM	Random Access memory
ROM	Read Only Memory
PWM	Pulse Width Modulation
PAL	Phase Alternating Line
NTSC	National Television System Committee
DFSK	Distributed Frequency Shift Keying
NRZI	Non return to zero, inverted
AFSK	Audio Frequency Shift Keying
OSI	Open Systems Interconnection
OBEX	Parallax Propeller Object Exchange
EEPROM	Electronically Eraseable Programmable ROM
DIP	Dual In-line Package
LQFN	Low-profile Quad Flat Package
QFN	Quad Flat Package
IC	Integrated Circuit
PLL	Phase Locked Loop
UI	Unnumbered Information
SSID	Service Set Identifier
FCS	Frame Check Sequence

I/O	Input/Output
NMEA	National Marine Electronics Association
CMOS	Complementary metal–oxide–semiconductor
AGWPE	AGW Packet Engine
PCB	Printed Circuit Board
CNC	Computer Numerical Control
TNC	Terminal Node Controller
GUI	Graphical User Interface
ESL	Electronic Systems Laboratory

Contents

List of Tables

List of Figures

1. Introduction

Global communication, weather prediction, earth observation, disaster response and management and military reconnaissance all rely on the services and data provided by satellites. While a platform in orbit can provide many valuable services, the cost of getting hardware into space can be prohibitive. A platform taken into the upper atmosphere by a balloon has the potential to provide many of the same services and data, at a fraction of the cost. A balloon-borne platform has many other advantages, as discussed in Section 2.4.

The aim of this project is to develop and test a reusable, expandable, high-altitude, balloon-borne platform named the Stellenbosch University Near-space Engineering Testbed (SUNSET). The purpose of this project is not to perform any specific experiment while in the upper atmosphere. Rather, this project focuses on designing and testing an expandable platform which can carry future experiments into the upper atmosphere. The essential components of such a platform include a programmable flight computer, a long distance communications link, a GPS sensor to enable recovery and reuse, batteries and a thermally insulated airframe. These components will be referred to as the SUNSET payload.

The SUNSET payload, together with an example experiment consisting of two temperature sensors and a camera, will be designed, manufactured and tested. A ground station which can track and display the data from the SUNSET payload will be developed, too. The payload will be designed to be as light weight and low cost as possible. After manufacture, sub-system and tether tests will lead up to a full, untethered flight. To be successful, the SUNSET payload must remain in constant communication with the ground station throughout its flight, return sensor data, land safely and be successfully tracked and recovered.

This report consists of five sections. Background information is given in Section 2. Section 3 describes the design of the SUNSET payload and ground station. Section 4 describes the implementation and manufacture of the payload. The testing procedures and results are given in Section 5. Finally, Section 6 contains the conclusion and recommendations.

2. Literary Review

2.1 Weather Balloons

A weather balloon is a balloon made from a flexible latex material which is designed to carry a small electronics package, called a radiosonde, into the upper atmosphere. Weather balloons can reach altitudes of 35km or more [1]. This area of the atmosphere, above 23km and below 100km, is known as “near-space”, due to its proximity to the boundary between our atmosphere and space. The International Aeronautical Federation defines this boundary to be at 100km above the earth’s surface [2]. The conditions in near-space are extreme and very similar to those found beyond our atmosphere, in outer space [3]:

- The sky is black and the curvature of the earth is visible.
- The air pressure is 97% lower than on the earth’s surface.
- The cosmic flux is 200 times greater.
- The temperature is typically around -40 degrees centigrade.

One of the first people to employ weather balloons for scientific study was a French meteorologist named Léon Teisserenc de Bort [4]. He began launching weather balloons in 1896 from his observatory in Trappes, France. During his experiments, which involved the launching of hundreds of weather balloons, Léon Teisserenc de Bort discovered the tropopause and stratosphere.

Weather balloons are launched to measure atmospheric conditions to aid in weather prediction. These weather balloons carry a small electronics package which transmits temperature, pressure, wind speed and direction readings back to a ground station. From case studies included in section 2.3, it appears that weather balloons take an hour or two to reach their maximum altitude. At this point, the low atmospheric pressure has caused the balloon's diameter to expand up to ten times its original inflated diameter. The balloon bursts at maximum altitude and the radiosonde is lost as it falls back to earth. The radiosondes used for weather prediction are typically not tracked or recovered. Approximately 800 locations around the world release weather balloons twice daily [1].

2.2 Amateur Radio

Amateur radio, or ham radio, is a popular hobby practiced by many around the world. It involves the use of designated radio frequencies for exchanging messages for recreation, experimentation and emergency communication. The term 'amateur' refers to the non commercial use of radio communication, as opposed to commercial public broadcast radio.

Amateur radio operators make use of a number of different modes of communication. The most common mode of communication is voice. Using an amateur radio as a 'walkie-talkie' has several advantages over similar technologies such as cell phones. Firstly, amateur radio communication is free. Secondly, it can be used anywhere in the world without the reliance on a network infrastructure. However, another amateur radio operator needs to be in range in order for a message to be heard. Cell phones usually need to be within about three kilometres of a cell phone tower, but certain amateur radio signals can travel hundreds of kilometres. Besides voice, many other forms of media can be transmitted using amateur radios. Images (Slow Scan TV), video (Amateur TV), Morse code and data (Packet) are routinely transmitted across the amateur radio frequencies.

One of the main challenges of amateur radio is trying to make contact with other operators over long distances. One method used by amateur radio operators to extend their operating range is to setup and make use of repeaters. Repeaters are unmanned stations which retransmit any amateur radio signals received. Repeaters are usually situated on high terrain, such as on hills and mountains, to increase their range. Making use of repeaters, amateur radio operators can communicate with non-line-of-sight stations by having their signal hop from one repeater to another.

In an effort to communicate over ever greater distances, amateur radio operators began attaching repeaters to weather balloons. The first recorded launch of amateur radio equipment on a weather balloon was by the Llamari program in Finland in 1967 [5]. Ralph Wallio called the new hobby: Amateur Radio High Altitude Ballooning (ARHAB) [6]. Soon thereafter, temperature sensors, pressure sensors, cameras and other experiments began to get added to weather balloon payloads. Unlike the radiosondes launched by weather forecasters, amateur radio operators want to recover their equipment after the balloon bursts. In recent years, the amateur radio operators have begun to include Global Positioning System (GPS) sensors in their weather balloon payloads to aid recovery. The payload can then transmit its coordinates back to the ground using a data mode of communication over the amateur radio frequencies.

As microelectronics and GPS technologies became cheaper and more readily available, the hobby has become more widespread. Today, many universities, amateur radio clubs and individuals have successfully constructed and launched electronics packages attached to weather balloons [6].

2.3 Case Studies

The following four case studies were chosen as they give a good indication of the technologies involved and accomplishments of near-space weather balloon launches to date.

Table 1 Case Study 1: HABEX

HABEX – High Altitude Balloon Experiments [7]	
Launch Date	24 July 2010
Organisation	Sci-Bono Discovery Centre in Johannesburg
Aim/Purpose	High school outreach. Various experiments and amateur radio equipment
Mechanical	• 2m diameter latex filled with helium • Payload weight: 1.2kg • 1.2m diameter parachute
Electrical	• U-Blox Neo 6Q GPS module • RadioMetrix SHX1 500mW vhf transceiver • MX614 Modem Chip • Automatic Position Reporting System (APRS) utilised • PIC24FJ48GA004 • Amateur radio repeater and HF beacons
Flight Details	• Flight 1 launched from Klerksdorp airport - APRS failure. GPS coordinates did not show up in telemetry • Flight 2 successful - Altitude reached: 35 000 m - Duration of flight: 50 mins - Distance travelled: 174km

Table 2 Case Study 2: BEAR3

BEAR3 - Balloon Experiments with Amateur Radio [8]	
Launch Date	22 August 2007
Organisation	Group of individual Amateur radio enthusiasts in USA
Aim/Purpose	Altitude Record Attempt. Aiming for 130k feet
Mechanical	• 1500g balloon filled with hydrogen • Payload weight: 95.8g • Custom foam payload enclosure
Electrical	• Trimble Lassen iQ GPS • Byonics Micro-Trak 300 (off-the-shelf APRS encoder and transmitter utilising 300mW vhf radiometrix transmitter) • 4x AAA Lithium batteries
Flight Details	• Altitude reached: 35 474 m • Duration of flight: 2hrs 39 mins • Distance travelled: 76.7km

Table 3 Case Study 3: HALO

HALO – High Altitude Object [9]	
Launch Date	8 October 2007
Organisation	Individual in Canada
Aim/Purpose	Personal project. High Altitude Photography
Mechanical	• 1200g Totex sounding balloon filled with helium • Payload weight: 1.5 kg • 1.22 m diameter parachute
Electrical	• Telit GM862-GPS cellular module with built-in GPS and built-in python interpreter • Long range 900Mhz XTend transceiver. Commands can be uplinked • Verdex 600MHz single-board ARM computer • Servo tilt Canon PowerShot A510 camera • Temperature and pressure sensors • $LiSO_2$, 5 cells in series resulting in 15V, giving 8-16 hrs operating time
Flight Details	• Altitude reached: >30 000m • Duration of flight: 2 hrs • GPS stopped working above 24km • 900MHz modem reception failed beyond 5km • SMS successfully sent upon landing • Parachute got tangled with balloon remnants, resulting in hard landing • 269 pictures, 58 videos returned

Table 4 Case Study 4: ICARUS

Icarus [10]	
Launch Date	2 September 2009
Organisation	Massachusetts Institute of Technology (MIT) students
Aim/Purpose	Cheapest near-space launch. $150 (R1000) including launch High Altitude photography
Mechanical	• Payload weight: 800g • 350g Kaymont weather balloon, filled with Helium • Styrofoam container, containing a chemical handwarmer packet and crumpled newspaper • 1.5 m diameter parachute
Electrical	• Motorola i290 GPS enabled cellphone plus booster antenna • AccuTracking software running on the cellphone • Duracell-Lithium AA backup batteries • Canon A470 camera running CHDK firmware
Flight Details	• 29.8 km • Altitude reached: 29 870 m • Duration of flight: 5 hrs • Distance travelled: 32km

The following lessons can be learned from these and other noted case studies:

- Radiometrix transmitters have proven flight experience. 300mW in the Very High Frequency (VHF) band seems to be enough power to ensure reliable communication while line of sight is possible. Modems that transmit on higher frequencies (eg 900Mhz) and can manage the distances required for near-space communication are too expensive and power hungry to be used onboard SUNSET. The APRS packet format allows any amateur radio operator to receive tracking packets, allowing him to aid in the recovery effort.

- From the experiences of HALO and Cygnus [11], it seems that cell phones do not work above about 500m above ground level (AGL). This is probably due to the radiation pattern of cell phone tower antennas, which focus their transmitted energy horizontally [12]. However, they are useful for sending an SMS containing the payload's location upon landing, especially if it lands out of line of sight.

- Some GPS modules, especially older ones, do not work up to 30km altitude. This is a safety feature present in their firmware to prevent their use in missiles. Newer GPS modules are less problematic, but their specifications should be checked.

- BEAR3 and Icarus showed that off-the-shelf, non-rechargeable lithium batteries fare well in the cold conditions of near-space. They are much lighter than most equivalent alkaline, Nickel-metal hydride (NiMH) and Nickel-cadmium (NiCd) batteries, and much cheaper than rechargeable lithium ion batteries. Lithium AA batteries are also readily available.

- Various processors have been used successfully in near-space. Cosmic radiation does not seem to be a notable problem, as it is not mentioned in any of the case studies. The short duration of the payload's exposure to near-space minimizes the probability of damage from cosmic radiation.

- Styrofoam containers provide adequate thermal insulation for the electronics and batteries against the cold temperatures of near-space.

2.4 The Benefits of Entering Near-space

Some of the benefits of near-space flights for weather prediction and amateur radio communication have already been mentioned. However, there are several other reasons for designing near-space payloads and entering near-space:

- **Cheaper access to outer space**. The idea of launching a small rocket from a high altitude balloon originated in the 1940s [2]. This combination has become known as a "rockoon". Launching from a balloon gives the rocket a 30km head start in altitude. But more importantly, by starting at this altitude, a rocket does not have to waste fuel fighting against air friction. This allows rockets to be smaller, safer and cheaper. Unfortunately, a balloon borne launch platform is relatively unstable. JP Aerospace, Cambridge University and several other organisations are currently investigating rockoons as an option for delivering small satellites into orbit [2].

- **High altitude photography**. Images taken from near-space cover several hundred kilometres more than images taken from ordinary aircraft. These high altitude images are useful for geographical mapping, agriculture, fire tracking, and surveillance. Using a near-space balloon payload to capture these images is far more cost efficient than employing a satellite, and can be achieved quickly and by almost anyone.

- **Testing equipment for outer space**. It is much cheaper to send equipment into near-space than it is to get it into orbit. This fact, combined with the similarities between the conditions in near-space and outer space, makes near-space a perfect testing ground for new, space-destined equipment. Surviving the extreme cold, low atmospheric pressure and high cosmic radiation levels are good tests for space-destined equipment. The opportunity to design near-space payloads at university can give students a good idea of the difficulties involved in designing spacecraft.

- **Atmospheric measurements.** Payloads attached to weather balloons can take atmospheric measurements at various altitudes. Pollution can be monitored by measuring carbon-dioxide, methane and sulphur compounds in the air, as well as particle concentrations. The ozone layer resides between 15 and 35km above the earth's surface [13]. This is an area known as the stratosphere. Planes typically fly far below these altitudes, which make near-space balloons essential for monitoring the ozone layer. Another benefit of near-space payloads is their ability to be recovered. The expensive equipment can be reused and collected samples can be further studied on the ground.

- **Cosmic Ray experiments.** The majority of cosmic particles from the sun collide with molecules in our atmosphere and break up into new particles. Near-space payloads are perfect for studying these cosmic particles as measurements can be taken above most of the atmosphere. These particles and their interactions with our atmosphere are very interesting to elementary particle physicists and planetary scientists.

- **Astronomy.** The atmosphere distorts the images captured by ground based telescopes. The largest ground based telescope has an 11m diameter mirror, but its images are still not as sharp and clear as those captured by the Hubble Telescope's 2m diameter mirror. The Hubble Telescope's advantage is that it sits in orbit above most of the distorting atmosphere. Telescopes taken into near-space by a balloon will be above the distorting atmosphere. One such project is called BLAST (Balloon-borne Large-Aperture Submillimeter Telescope). During three successful flights, BLAST used its 2m diameter mirror to investigate the formation of stars [14].

- **Disaster Relief.** A balloon carrying high resolution cameras or amateur radio repeaters could be very useful during disaster management activities. A satellite can take several hours to get above a disaster stricken area, whereas a balloon can be launched in a matter of minutes. High resolution images from such a balloon could be used to see the full extent of damage and aid in rescue and cleanup activities. Often cell phone towers and power distribution are affected during a natural disaster. Communication on amateur frequency bands is useful in such a situation. A repeater on a near-space balloon can act as a long distance relay. Pre-recorded public information messages can also be broadcast from such a balloon during communication blackouts.

2.5 SUNSET Innovations

In an effort to continue the exploration of near-space and the development of payloads capable of operating in near-space, the SUNSET payload will not merely repeat what has been done before.

Firstly, the SUNSET project will investigate the benefits of using a novel microcontroller, named the Parallax Propeller, as its main flight computer. This topic is further explored in section 2.6.

Secondly, the SUNSET payload will attempt to transmit live images back to the ground station during the flight. The majority of near-space payloads save images to memory cards. These images can only be seen if the payload is recovered. The few near-space payloads which have transmitted live images to ground used Amateur Television (ATV). This is an analogue mode of video transmission which requires powerful transmitters and large amounts of power. SUNSET will attempt to digitally transmit images back to the ground station. Even if the payload is lost during descent, the images will have been saved. This feature is explained in more detail in the Section 3.2.7.

Thirdly, SUNSET will utilise redundant communication systems. An amateur radio transmitter will provide live telemetry and tracking data by making use of the AX.25 and APRS protocols described in section 2.7. As a backup, a GSM modem will send the recovery team an SMS with the payload coordinates once it lands. This has been done before, for example by project HALO, but will be accomplished in SUNSET with very cheap, off-the-shelf components. The SMS backup is important as the payload could land behind a mountain, preventing the radio telemetry from being received.

Finally, SUNSET will be designed, constructed and launched on a very tight budget. Typical launches can cost up to $500 (R3500) [9]. The cheapest launch seems to have been achieved by the MIT students of project Icarus [10], who spent a total of $150 (R1000). A budget of R750 is available for the construction of the SUNSET payload, excluding the costs of launch.

2.6 Flight Computer

The flight computer is the main processor which controls the sequence of actions taken by the payload. This includes taking images, reading sensors and transmitting telemetry. There are several important specifications and limitations which must be taken into account when choosing a processor for the flight computer. The most important of these are: weight, number of input and output ports (I/O ports), power consumption, and onboard peripherals such as analogue to digital converters (ADCs) and Universal Asynchronous Receiver/Transmitters (UARTs). The flight computer's specifications must be closely matched to its intended job. A flight computer which is too powerful wastes resources such as battery capacity. On the other hand, a processor which is too weak will need additional hardware peripherals to perform important tasks. Using a laptop as a flight computer is unfeasible due to weight and cost restrictions. ARM processors are very powerful and can be found in many smartphones. However, due to time restrictions it was decided that learning a new platform was too risky. Additionally, the high clock speeds and processing power provided by ARM microcontrollers is not necessary for the SUNSET payload.

Three different microcontrollers were considered for SUNSET's flight computer: the Renesas R8C, the Microchip PIC range, and the Parallax Propeller. Table 5 briefly compares the specifications of each microcontroller:

Table 5 Comparison of Potential Flight Computer Microcontrollers

	Microcontrollers		
	Renesas R8C/10 [15]	PIC24FJ48GA004 [16]	Parallax Propeller [17]
Architecture	16 bit	16 bit	32 bit
Num CPU cores	1	1	8 (called cogs)
Num I/O ports	22	35	32
CPU speed	16MHz, 16MIPS	8MHz, 16MIPS	80MHz, 20MIPS x 8 cogs
Program memory	16KB	48KB	32KB
Temp range	-40 to 85°C	-40 to 125°C	-55 to 125°C
A/D converter	8 channels, 10 bit	10 channels, 10 bit	none
Onboard UART	1	2	UART can be emulated by dedicating a cog
Average power consumption	8mA	15mA	10mA per active cog
Price per IC [18]	R39.05	R39.17	R128.84
IDE setup	difficult	medium	easy
Programmer	Expensive, external hardware	External hardware	No programmer required
Other reasons for consideration	Previous experience with Renesas range	Previous experience with PIC range. Proven flight experience aboard HABEX.	Previous experience with Parallax Propeller. More below

The Renesas and PIC microcontroller ranges are well known and widely used. The Parallax Propeller, however, is a lesser known microcontroller with unique and powerful features.

The Parallax Propeller chip was designed and developed by Parallax Inc for embedded systems. It has a unique architecture consisting of eight individual, 32-bit RISC CPU cores, called cogs, which communicate with each other and communal peripherals through a central hub. Each cog has access to its own 2KB of RAM, divided into 512 32-bit words. Both data and instructions can be stored in this memory. Each cog also has access to 32KB RAM and 32KB ROM of shared memory. Access to this memory and other communal peripherals is controlled in a round robin fashion. On-board peripherals include special counters and signal generation hardware to aid in video, and Pulse Width Modulation (PWM generation). A bitmap font, logarithm, antilog and sine tables exist in ROM.

The Parallax Propeller can be programmed using a high level language developed for the Propeller named Spin, or low level Propeller assembly. Spin is an object-based, interpreted language. The programmer has full control over what code runs on which cog. No splitting and assigning of tasks is done by the compiler or processor. This structure works well in combination with the object-oriented nature of the Spin programming language. Almost all propeller assembly instructions take four clock cycles. This makes the programming of timing dependant tasks easier.

A large collection of official and user written software libraries, or objects, exists in a central database available online. Examples include: PS2 mouse and keyboard drivers, PAL/NTSC video generation libraries, Secure Digital (SD) card drivers, an object which emulates a Bell 202 audio modem and many communication protocols including I2C and RS232.

As can be seen from Table 1, each of the microcontrollers has its strengths and weaknesses. Those of the Parallax Propeller are listed and explained in more detail below:

Advantages:

- Software on the powerful Parallax Propeller can replace hardware. Less hardware means easier and shorter hardware development time, fewer points of failure and more reliability.
- Multiple cogs make extending software easy. New features run on new cogs.
- Multiple cogs are good for interfacing to multiple peripherals simultaneously.
- Multiple cogs make software development modular.
- No unpredictable interrupts required.
- No external programming hardware required.
- Power consumption scales with number of active cogs. Unused cogs use almost no power.
- Military grade temperature range: -55 °C to +125 °C.
- Large, central database of official and user contributed software libraries.
- Simple to setup and use integrated development environment.

Disadvantages:

- Expensive
- New users need to learn the Propeller Spin programming language
- No hardware analogue to digital converters.
- Relatively large power consumption when many cogs are running.

Fortunately, each of the disadvantages can be overcome. The expense of the Parallax Propeller will be offset by the simpler supporting hardware and lack of an expensive external programmer. Previous experience with the Spin language has shown that it is simple to learn and use, and the large collection of existing software objects makes writing complex code easier. The lack of hardware analogue to digital converters can be overcome with clever software and minimal, passive external components. The higher power consumption of the Propeller is not a major concern, as the calculations in Section 3.2.8 show that the onboard batteries will be able to power SUNSET for more than twenty hours. The Propeller's power consumption can also be minimized by using cogs sparingly.

The Parallax Propeller has a large list of advantages, and each of its disadvantages can be overcome, making it the best choice for the SUNSET flight computer.

2.7 AX.25 Packet and APRS

AX.25 is a data link layer protocol designed for use by amateur radio operators. Its full specification can be found in the document: *AX.25 Link Access Protocol for Amateur Packet Radio* [18] . It is derived from the X.25 protocol and implements the first, second, and often third layers of the Open Systems Interconnection (OSI) networking model. The AX.25 protocol is responsible for transporting data, encapsulated in frames, between nodes and detecting transmission errors. AX.25 supports both connected and connectionless modes of operation. Although the protocol does not officially describe the physical layer implementation, AX.25 is used almost exclusively with 300 baud Bell 103 tones on High Frequency (HF), 1200 baud Bell 202 tones on VHF and 9600 baud G3RUH Distributed Frequency Shift Keying (DFSK) on VHF and Ultra High Frequency (UHF) [19]. The AX.25

protocol describes the use of NRZI encoding, bit-stuffing and defines the format of frames. Further details of this protocol are explained in Section 3.2.2.

The AX.25 protocol describes a complete, point to point only, networking layer. Alone, this specification is enough to enable keyboard to keyboard communication between packet stations. Digipeaters are able to forward messages allowing communication between stations which are not in range of each other. Any Layer 3 protocol can be implemented on top of AX.25.

In recent years, with the decrease in cost of GPS devices, Automatic Position Reporting Service (APRS) has become a popular application of the AX.25 protocol. The APRS protocol is fully described in the document: *APRS Protocol Reference Version 1.0* [20]. APRS packets are broadcast as connectionless AX.25 frames, called unnumbered information frames. Anyone within range of the transmitter can receive APRS packets. APRS was first developed by Bob Bruninga in the late 1980's for real time communication of short messages to the immediate surroundings. There are four types of APRS packets: position, status, messages and queries. Position packets contain the coordinates of a fixed or mobile station, derived from a GPS unit. These packets have many optional fields, too, including: altitude, speed and heading. APRS position packets received by a station can be displayed automatically on a map using computer software. Status and Message packets can contain any text up to 256 characters. These packets are often used to transmit telemetry from unmanned weather stations.

All APRS packets have a field specifying a path. This path refers to the sequence and number of digipetears that the packet is desired to follow. Every packet is only allowed to be repeated by a digipeater a finite amount of times, otherwise the digipeaters will repeat each other and clog up the system. More details about the path field are described in Section 3.2.2. In South Africa, all APRS packets on the VHF band are transmitted on 144.800MHz. In urban areas this band can get quite congested. APRS stations need to ensure that they do not transmit too often or they risk clogging up the entire APRS system. A few APRS stations, called I-Gates, can also forward packets onto the internet for display on a global APRS website at http://aprs.fi.

SUNSET makes use of the APRS protocol to allow live, long distance tracking of the payload, and live telemetry. Making use of an existing protocol, such as APRS, has many advantages. Firstly, a large range of APRS decoding and plotting software for PCs exists. Secondly, a country wide system of digipeaters and I-Gates will help to propagate and detect the packets coming from the SUNSET payload. Even if the SUNSET packets are not heard directly by our ground station, there is a good possibility that they will be digipeated or forwarded onto the internet for later viewing. Finally, using the APRS protocol and standard amateur radio frequencies allows standard equipment to be used for the ground station. Anyone with a ham radio and a computer will be able to decode the SUNSET packets without additional hardware or software.

3. Design

3.1 Overview

The SUNSET project will consist of three sections: The payload electronics, the flight string and the ground station. The payload electronics includes a microcontroller, multiple sensors, an SD card interface, radio transmitters and a battery pack. The payload electronics are described in detail in section 3.2. The flight string, described in section 3.3, consists of a latex weather balloon connected by nylon string to a parachute and an insulated payload container. The ground station, described in section 3.4, is comprised of a laptop computer running third party and custom software, a VHF amateur radio receiver and an antenna.

3.2 Payload Electronics

A system diagram of the payload electronics is presented below:

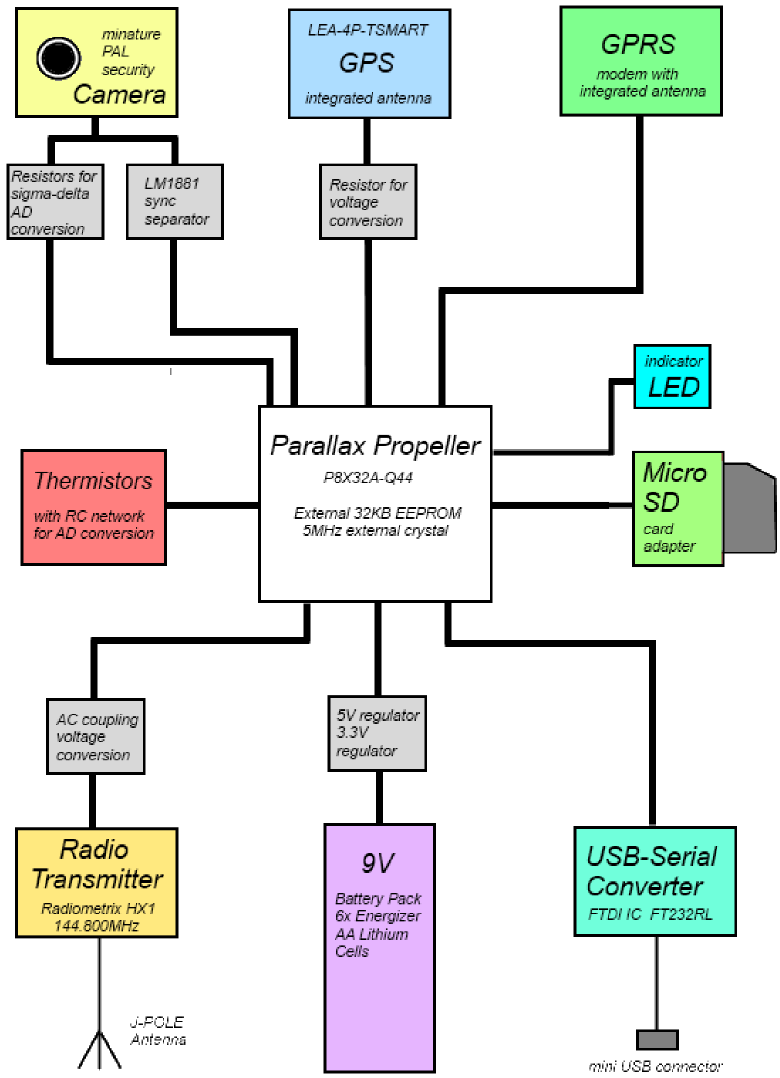


Figure 1 System diagram of payload electronics

Each of the components displayed in Figure 1, and their interface to the microcontroller acting as flight computer, are explained in detail in the sections that follow. All Propeller software objects mentioned are available from the Parallax Propeller Object Exchange (OBEX) on the Parallax website.

3.2.1 Flight Computer

The Parallax propeller runs on 3.3V. It is available in three different packages: a 40 pin DIP, a 44 pin LQFP and a 44 pin QFN. The Parallax Propeller has been rated to military temperature specifications. The P8X32A is currently the only model in the Parallax Propeller range. However, at the time of writing this report, the Propeller 2 is under development [21].

The basic supporting circuitry for the Parallax Propeller is used as described on the Demo Board schematic, supplied in the Propeller datasheet.

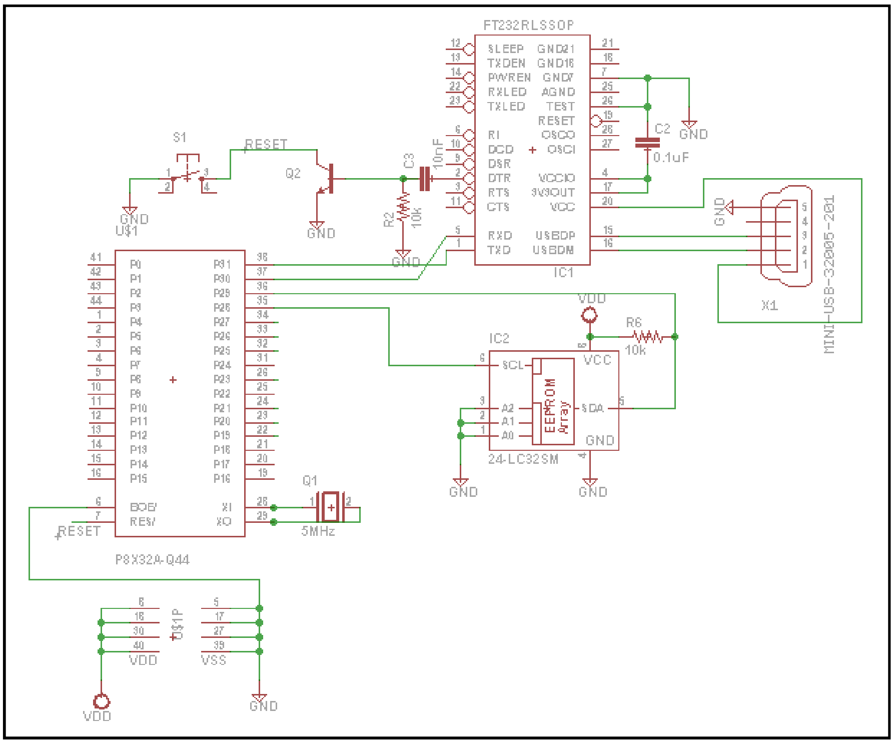


Figure 2 Schematic diagram of the Parallax Propeller supporting circuitry

The component labelled P8X32A-Q44 in Figure 2 is the Parallax Propeller in an LQFP package. This package is chosen because it is much lighter and smaller than the DIP version, but also much easier to solder than the QFN version. The Propeller integrated circuit (IC) has 4 VDD and 4 GND connections. A 100nF decoupling capacitor (not shown in the schematic above) is connected between each VDD and GND close to the IC.

To ensure the best compatibility with existing software objects, and to make timing intensive functions possible, the Propeller will be clocked at 80MHz. An external 5MHz crystal and a 16x phase locked loop (PLL) software setting enable this.

The Propeller's boot procedure, as described in its datasheet, must be understood to understand the other supporting components in Figure 2. Upon power up, Cog 0 loads a built in bootloader from ROM. The bootloader first checks for a serial connection to a host PC. If a connection is found, a program can be downloaded directly into the Propeller's main RAM, or written to an external EEPROM. If no connection is detected, the bootloader looks for an external EEPROM connected to the Propeller. If the EEPROM is found, the entire content of the EEPROM is loaded into main RAM. User written programs are loaded from the EEPROM in this way at every start up. Once the program exists in the Propeller's main RAM, Cog 0 is reloaded with a built in Spin Interpreter. Program execution begins, and various cogs are initiated and stopped during the program execution.

A 32KB I2C EEPROM (24-LC32SM) is connected to pins 35 and 36. All three address lines on the EEPROM are pulled low, and a 10kΩ pull-up resistor is added to the SDA line, as specified in the Propeller datasheet.

FT232RL is a USB to Serial converter IC. It allows the Propeller to be programmed by a personal computer (PC) using only a USB cable. The Propeller can also communicate with a host PC using this virtual serial port. The Propeller can be reset by pulling the reset pin low. This is achieved by either pressing a tactile switch, or by the DTR pin of the FT232RL IC via an NPN transistor. The Parallax Propeller Integrated Development Environment (IDE) pulses the DTR pin high just before programming starts, to reset the Propeller.

3.2.2 VHF Radio Transmitter

In order to make use of the AX.25 amateur radio protocol, data needs to be encoded as sound. This is normally done by a modem. Modem ICs exist, such as the MX614 used by HABEX, but they are becoming increasingly rare and expensive. A second option is to use the Parallax Propeller to perform the modem's function in software. This approach has the advantage of being more flexible and requiring less hardware.

Philip Pilgrim wrote an object for the Parallax Propeller which imitates a Bell202 modem [22]. The Bell 202 standard uses two different audible frequencies to represent ones and zeros. This is known as Audio Frequency Shift Keying (AFSK). A 2200Hz sine wave signifies a one, and 1200Hz signifies a zero. Data is encoded at 1200 bits per second. The Bell 202 Modem object generates a continuous sine wave which alternates between 1200Hz and 2200Hz. Figure 3, modified from Philip Pilgrim's Bell 202 object datasheet [22], shows the process of encoding a byte as sound:

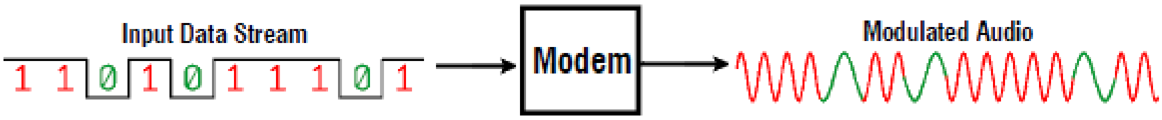


Figure 3 The Process of encoding data as sound

The biggest advantage of AFSK is its ability to transmit data across networks and through devices which were originally designed for voice. Examples include the telephone network, amateur radio equipment and computer soundcards. Computer software exists which can use the soundcard to sample AFSK encoded data, and decode it. This is process further described in Section 3.4.

The AX.25 and APRS protocols must be implemented on top of the Bell 202 standard. The document: *PIC-et Radio: How to Send AX.25 UI Frames Using Inexpensive PIC Microprocessors* by John Hansen describes all the steps necessary for a microcontroller to create valid APRS packets [23]. First, Philip Pilgrim's assembly code must be modified to add bit-stuffing and NRZI (Non-Return to Zero, Inverted) modulation, as required by the AX.25 protocol.

Bit-stuffing is the process of adding a zero to the data stream after five consecutive ones. This is done to ensure that the transmitter and receiver bit timing stays synchronised. Bit-stuffing is done for the whole data stream, except during flags (described below). NRZI modulation specifies that a one or zero is not represented by a specific state (or frequency). Instead, a change in state represents a zero, while no change in state represents a one. Figure 4 illustrates the concept:

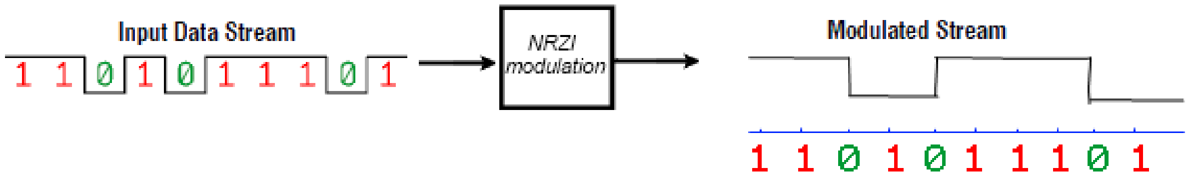


Figure 4 Example of NRZI modulation applied to a byte of data

Three changes of state can be seen in the modulated stream. These represent zeros. If the state does not change for a set amount a time (1/baudrate seconds), a one is interpreted.

Next the APRS protocol must be implemented using AX.25 Unnumbered Information (UI) frames. The AX.25 protocol divides each packet into seven fields, as shown in Figure 5. The fields include: Flag(s), Address, Control, Protocol Identifier (PID), Information, Frame Check Sequence (FCS) and Flag(s) again.

	AX.25 UI-FRAME FORMAT								
	Flag	Destination Address	Source Address	Digipeater Addresses (0-8)	Control Field (UI)	Protocol ID	INFORMATION FIELD	FCS	Flag
Bytes:	1	7	7	0–56	1	1	1–256	2	1

Figure 5 The composition of an AX.25 UI frame [20].

The following is a description of a complete AX.25 UI frame, as will be transmitted by the SUNSET payload:

The transmission begins with a flag. A flag is a special set of bits which tells the receiver that a new packet is starting or ending. Hence, there must be at least one flag between consecutive packets. A flag must contain a unique value which will never appear in ordinary data transmissions. The hex value 7E (01111110 in binary) is chosen (specified by the AX.25 protocol) because it contains six consecutive ones. Since bit-stuffing is not applied to the flag byte, the value 0x7E will never appear during data transmissions.

Next is the Address Field, which contains several amateur radio callsigns. First the destination callsign is transmitted, followed by the source callsign and a list of digipeaters. An example callsign is ZU1LEG-1. The first six characters constitute the callsign. The trailing number is known as the Service Set Identifier (SSID). The APRS protocol utilises the SSID field to specify the characteristics of the transmitting packet station. An SSID of 11 represents an airborne transmitter, such as the SUNSET probe. The AX.25 protocol specifies that every callsign must be exactly six characters long and shorter ones must be padded with spaces. Because the list of repeater callsigns can be variable in length (0-8), the receiving station has no idea when the address field is finished. This is solved by first shifting the bits of every character of the callsign one bit left. This leaves a zero in the least significant bit. Secondly, the following bit pattern is used to encode the SSID of every callsign: 011SSSSx. SSSS designates the SSID and the x is replaced by a one if it is the last callsign, or a zero if it is not.

In this way, by looking at the last bit of every callsign's SSID, the receiving station can determine whether this is the last callsign in the list.

Because APRS packets are not directed at any particular receiving station, a generic destination callsign "CQ" is used. The list of digipeaters, or path as it is also called, is left empty to comply with good APRS practice [24]. This is because a transmitter on a weather balloon has such a long line of sight that it does not need to have its packets digipeated. By not abiding to this rule of good practice, SUNSET's transmissions could potentially reach a vast number of digipeaters, clogging up the network with retransmissions. However, once the payload has landed, or is below a certain altitude (still to be decided after the accuracy of the GPS sensor's altitude measurements has been determined experimentally), the path will be changed automatically to include digipeaters. This will aid in recovery if the payload moves out of line of sight.

The Control and PID fields follow the address field. For UI frames, the control field is always 0x03 and the PID field is always 0xF0.

Next is the Information Field. The AX.25 protocol allows the Information Field to consist of up to 256 bytes. It does not specify what these bytes must be. SUNSET will transmit two different kinds of telemetry by putting different data into the information field. A telemetry packet begins the information field with the symbol '!'. This is followed by an ASCII string representing the balloon's current GPS coordinates, an ASCII string representing altitude, and two ASCII strings representing temperature measurements. All data is sent as ASCII strings so that it can be received and viewed by any third party packet software. An example of a SUNSET telemetry packet, as displayed by third party packet program MixW, is below:

```
ZU1LEG-4>CQ>UI,?,F0:!3358.50S/01850.50E-a120m+35+24
```

The second kind of telemetry that SUNSET will transmit is image data. These packets begin the Information field with the letter 'T'. Six lines of image data are sent in each packet. This is described in detail in section 3.2.7. Custom ground station software combines the data in these packets to recreate and display the images.

The second last field in an AX.25 packet is the Frame Check Sequence (FCS). The FCS is a unique value generated from the contents of the rest of the AX.25 packet. When a receiving station receives an AX.25 packet, it calculates the FCS and compares it to the one in the received packet. If the two match, the packet has been received without errors. The longer the packet, the smaller the chance that it will be received error free. There is no way to correct detected errors and because the SUNSET communication is one way, there is no way to ask for a retransmission. Packets which are received with errors are lost. Fortunately, the SUNSET payload will be transmitting regularly, so missed packets are not of major consequence. More information about the FCS algorithm can be found in the AX.25 specification document [18]. The few lines of code to compute the FCS on the Parallax Propeller was written by Dave Hein on the Parallax Propeller Forums [25].

An AX.25 packet is ended in the same way that it began, with a 0x7E flag.

The AX.25 packets, encoded in sound by the Bell202 object, have to be transmitted over the air by a radio. Two options were considered: A VHF JingTong handheld amateur radio, and a Radiometrix HX1 transmitter module. Either transmitter could have worked, but the Radiometrix module was chosen because it is cheaper, far lighter and has no unnecessary features. The following schematic shows the interface between the Radiometrix HX1 module, and the Parallax Propeller:

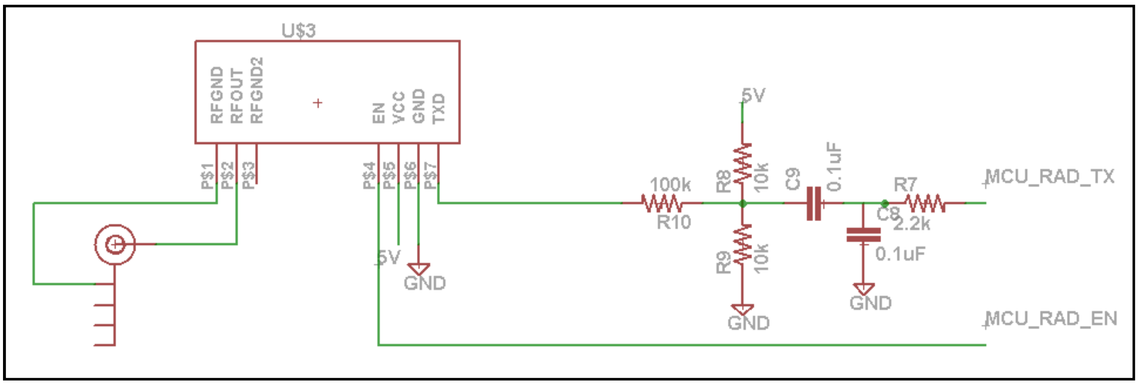


Figure 6 Radiometrix HX1 to Parallax Propeller interface

As shown in Figure 6, the Radiometrix HX1 module (on the left) is connected to the Parallax Propeller via an RC network. The Radiometrix HX1 requires 5V VCC and an input signal between 0-5V. Beginning from the right, R7 and C8 form a low pass filter. This is implemented as suggested by the Bell 202 object datasheet. The Bell 202 object outputs a varying pulse width modulated signal of which only the fundamental harmonic is let through by the low pass filter. C9 is an AC coupling capacitor. R8 and R9 form a voltage divider which centres the 0 – 3.3V signal from the Parallax Propeller around 2.5V for the Radiometrix HX1. Finally, R10 simply adjusts the amplitude, or volume, of the transmitted signal by forming a voltage divider with the Radiometrix's own 100kΩ input resistance [26]. This resistor will probably have to be chosen by experimentation. However, 100kΩ is a good starting value as it matches impedances with the Radiometrix HX1. The magnitudes of R8 and R9 are chosen as 10kΩ (1/10 the size of R10) so as not to influence the impedance matching done by R10.

The Independent Communications Authority of South Africa (ICASA), and the South African Radio League (SARL) split up the frequency spectrum and allocate bands to different activities. 144.800MHz is in the amateur radio band and is reserved for APRS packets. The Radiometrix HX1 is hard tuned to this frequency.

The Radiometrix HX1 transmitter needs an antenna to ensure reliable transmissions over long distances. The antenna must be lightweight, cheap, rugged, omni-directional and as high gain as possible. All antennas that require a ground plane are not feasible. The best option is a flexible J-Pole antenna with 3.7dB of gain [27]. A flexible J-Pole antenna can be constructed from cheap 300Ω twin lead TV cable (also used as FM antennas for radios). The antenna will hang below the payload. It is flexible, so it will not sustain any damage upon landing. It also has extensive near-space flight experience, having been used on Cygnus [11] and many other launches. A basic link budget calculation in Appendix D shows that the communication link should achieve a range of 100km or more with line of sight.

3.2.3 GSM Modem

A GSM modem will serve as a backup communication link. The Radiometric VHF transmitter should work over very long distances, provided there is line of sight between the payload and the ground station. However, once the payload lands, the usable range of the VHF transmitter will fall dramatically. This is even more problematic if the payload lands behind a mountain or in a ditch. In order to communicate its landing location to the recovery team, the SUNSET payload will begin sending SMS messages with its coordinates to the recovery team as soon as it gets below 500m altitude.

A cheap, Chinese, USB GSM modem can be modified so that it accepts AT Commands from the flight computer. The modem (SKU 12057) is available from DealExtreme for $24.90 (R201) [28]. This modem is chosen above other AT Command compatible options, such as GSM modules and old cell phones, because it is cheap, light, and easy to interface to. Michael Finch explains on his website [29] how to modify the modem. Figure 7, which has been slightly modified, is from his website:

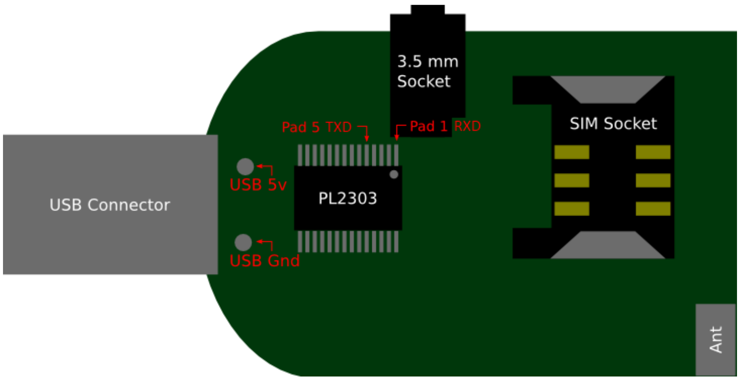


Figure 7 Serial connections to USB GSM modem [29]

The GSM modem consists of a BenQ M32 GSM/GPRS module and a USB to serial PL2303 IC. The USB host protocol is too complicated to implement on the Parallax Propeller. Fortunately, the BenQ M32 module can be controlled with 3.3V, 9600 baud serial commands by bypassing the PL2303 IC. The PL2303 IC will be desoldered, so that wires can be soldered onto the now empty pads. Figure 7 shows the important connection points. For simplicity, the SUNSET payload will only send SMS messages, not receive them. This means that only the RXD pin is necessary. The RXD pin is directly connected to one of the Parallax microcontroller I/O ports. The 5V and ground connections are connected to the SUNSET power regulator.

The BenQ M32 module accepts Hayes Commands, also known as AT Commands [29]. These are sent as strings using the Propeller object *Simple Serial*. Simple Serial emulates a serial port on any of the Propeller's I/O ports in software, but does not require its own cog. The following AT Commands are sent to the BenQ M32 module to send an SMS message:

Table 6 A List of useful AT Commands

AT Command	Description
AT	Get the modem's attention and detect baud rate
AT+CSMP=1,,0,0	Set the modem into plain text message mode
AT+CMGF=1	Use ASCII encoded text messages
AT+CSCA="+27829129"	Set the "message center" address for Vodacom RSA
AT+CMGS="+078399xxxx"	The destination number to send the SMS to
Your-message-text	Text message text terminated by ASCII character 26

3.2.4 GPS Sensor

The GPS sensor chosen for the SUNSET payload is the ublox LEA-4P reference board with integrated ceramic patch antenna. The LEA-4P is small (38 x 30 x 6mm), low cost (R245) and easy to interface to thanks to its CMOS/TTL interface.

The ublox LEA-4P can be configured by tying certain of its pins high or low, or by sending it serial commands. The default configuration (leaving all the pins unconnected) configures the LEA-4P to output position data 4 times per second, in the form of NMEA strings. The serial interface works at 9600 Baud. This is adequate for the needs of the SUNSET payload. No other configuration of the GPS sensor is needed. Therefore only the serial transmit pin is connected to a Propeller I/O pin, and the serial receive pin is connected to ground.

5V will be supplied to the GPS sensor by the SUNSET voltage regulator. A signal level conversion is needed between the GPS sensor and the Propeller. The LEA-4P outputs 0-5V serial signals, while the Parallax Propeller is a 3.3V device. Fortunately, the Parallax Propeller's designers have found that its I/O pins are 5V tolerant, thanks to parasitic clamping diodes [17]. As suggested by Parallax engineers on the Parallax Propeller forums, a 1kΩ current limiting resistor is put in series between the 5V GPS signals and the Propeller's input pin to protect the clamping diode [30].

Perry James Mole wrote a Spin object for parsing GPS NMEA strings, called *GPS_IO_mini*. *GPS_IO_mini* runs on its own cog, parsing NMEA strings in the background. *GPS_IO_mini* provides several methods which return strings representing latitude, longitude, altitude, and all other information contained in the NMEA data. It makes use of another Spin Object, written by Parallax, called *FullDuplexSerial*. *FullDuplexSerial*, unlike *Simple_Serial*, runs on its own cog in the background, emulating a hardware serial port. The GPS data can be continuously received, buffered and parsed in the background by *GPS_IO_mini* and *FullDuplexSerial*. When the flight computer wants to know its location, or wants to transmit a telemetry packet, it calls the methods in *GPS_IO_mini*.

The only operational limit mentioned in the datasheet is a maximum tracking speed of 515m/s. Hopefully this means that there is no altitude limit on the LEA-4P and that it will successfully retain GPS lock up to 30km altitude.

3.2.5 Temperature Sensors

Two temperature sensors will be included in the SUNSET payload. They will measure the inside and outside temperature as the balloon ascends into near-space. This data will be valuable to determine the effectiveness of the payload insulation. Since the aim of SUNSET is not to do a scientific study of the atmospheric temperatures, the accuracy of the sensors is not very important. Therefore it was decided that thermistors with approximately one degree Celcius of accuracy, would be accurate enough, while being cheap and easy to interface.

The Parallax Propeller does not contain any A/D converters. Fortunately, an alternative solution exists. An object named *RCTIME*, written by the Propeller developers, can do basic A/D conversion by timing the discharge of a capacitor through a variable resistance. In this case, the variable resistance is the thermistor. Figure 8 shows how a 47kΩ thermistor is connected to the Propeller, through a capacitor and a resistor, as specified by the *RCTIME* object:

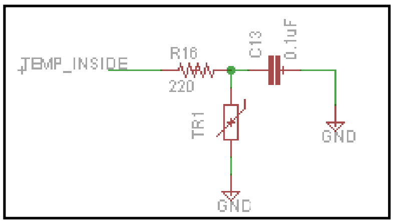


Figure 8 Interface between thermistor and Propeller

3.2.6 Secure Digital (SD) Card

A micro SD card will be used to store images and logs of GPS coordinates and temperature readings. The images will be saved as bitmaps, and the logs as text files, allowing any computer with an SD card reader to quickly view the stored information. The CLK, DO, DI and CS pins from the micro SD card slot are connected directly to Propeller I/O pins. VCC is connected to 3.3V. Although the SD card specification calls for pull-up resistors on all the data lines, several people on the Parallax forums have reported that the pull-up resistors are not necessary [31]. The interface was bread-boarded without pull-up resistors and worked without problems.

An object called *fsrw* was written for the Propeller by Tomas Rokicki and Jonathan Dummer. *Fsrw* provides FAT16/32 read/write access to SD cards. To log information, text files will be opened in append mode. *Fsrw* has methods for writing individual ascii characters to opened files. An object written by Perry James Mole (more information in Section 3.2.7), named *Pixelator_dumpscreen*, saves images taken with his *video_capture* object as bitmaps to the SD card.

3.2.7 Camera

One of the most exciting aspects of entering near-space is high altitude photography. Many near-space launches have resulted in beautiful images of the earth's curvature, or of the wide stretches of landscape below. Several near-space payloads, including HALO and Icarus, have successfully carried Canon cameras into near-space. These cameras are either flashed with special firmware which allows them to take an image automatically every few seconds, or the shutter button is electronically or mechanically pressed when signalled from the flight computer. While taking a complete point-and-shoot camera into near-space results in beautiful, high resolution images, these cameras are neither cheap, nor light. The flight computer also has no access to the images taken as they are stored directly onto an SD card. If the balloon is not recovered, all the images are lost.

The SUNSET payload will take a different approach to imaging. Instead of focussing on taking high resolution images, the SUNSET payload will attempt to transmit images back to the ground station during the flight. Megapixel resolution images are too large to transmit back to the ground station over a 1200 baud link. It will take too long to transmit each image. Therefore the images transmitted by SUNSET will have to be considerably lower in resolution. Perry James Mole wrote an object for the Parallax Propeller as part of his larger project, called *PropPixellator*, which allows it to interface to composite video sources [32]. It uses sigma-delta analogue to digital conversion to capture 64 by 240 pixels of data from a composite video frame directly into the Propeller's hub RAM. Each pixel contains a 6 bit greyscale value. While these images are very low resolution, they are small enough to be transmitted to the ground at 1200 baud. They are comparable in resolution to the earliest cell phone camera pictures. The quality and resolution is sufficient to see cloud formations, the curve of the earth against the black of space and land features such as mountains, trees and buildings.

The source of composite video will be a tiny, cheap (R150), off-the-shelf security camera. The camera is smaller than 2x2x2cm, and can be powered from 5-12V DC. There are several benefits to this approach as opposed to CMOS image sensor modules or other image sensor options. Firstly, any composite video source will be compatible with the SUNSET hardware. Secondly, the security camera has auto exposure, so the images it produces are always properly exposed, regardless of the lighting conditions. Finally, interfacing to image sensor modules is normally a huge project in itself. By using a composite video camera, the development time and cost of the SUNSET payload can be reduced considerably. The schematic below shows the interface between the Propeller and the video camera:

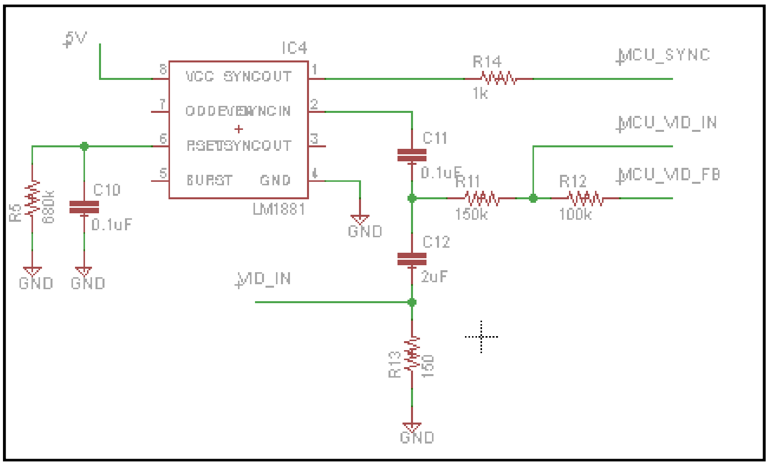


Figure 9 Parallax Propeller to video camera hardware interface

As shown in Figure 9, the main component of the camera-Propeller interface is the LM1881 IC. The LM1881 is a video sync separator. It extracts the new line and new video frame signals from the composite video signal. These sync signals are presented to the Propeller through a 1kΩ resistor (for clamping diode protection, as discussed for the GPS). R5, C10 and C11 are included as specified in the LM1881 datasheet. R13 is supposed to match the camera's output impedance to this circuit's input impedance, as suggested by the author of the *video_capture* object. The specified output impedance of the camera cannot be found. It is suspected to be 75Ω as this is common in television systems. However, the suggested circuit with 150Ω input impedance was bread-boarded and worked as is. C12 was added during bread-board testing, as it was found to be necessary. It AC couples the composite video signal, as opposed to a DC coupling. R11 and R12 form the hardware part of the sigma-delta A/D converter. They are included as specified by the author of the *video_capture* object. More can be read about sigma-delta A/D conversion in the Parallax Propeller application notes [33].

Because the SUNSET payload transmits on 144.800MHz, it may only transmit APRS packets. This means that the image data must be encoded into APRS packets, too. Each APRS packet can hold a maximum of 256 bytes in its Information field. It was experimentally determined that the two least significant bits of each pixel's data can be discarded without causing considerable loss of image quality. This allows two pixel's worth of data to fit into a byte. One row of image data contains 64 pixels, which fits into 32 bytes. 256 divided by 32 equals 8. Therefore 8 rows of image data can be sent in every APRS packet. However, to aid in reconstructing the image, a row number is included in every image data packet. For this reason, and the fact that packet reception errors increase with packet length, it was decided to put only 6 rows of image data into every APRS packet. When an image is taken, its data is immediately stored in the Propeller's hub RAM. Over approximately the next two minutes, 40 APRS packets containing the image data are transmitted. Custom ground station software, described in detail in section 3.4, recombines the data from the packets into an image. The images are also stored on the onboard SD card, described in section 3.2.7 in their 6 bit form.

The camera requires about 100mA at 5V during normal operation. Since this is a comparatively large current draw, and the fact that the camera will only be used for a few seconds at a time, it was decided that the Propeller should be able to turn the camera on and off. When the Propeller outputs 3.3V, a PNP transistor, and therefore camera, must be off. When the Propeller outputs 0V, the transistor must turn on, supplying the camera with enough current to turn on.

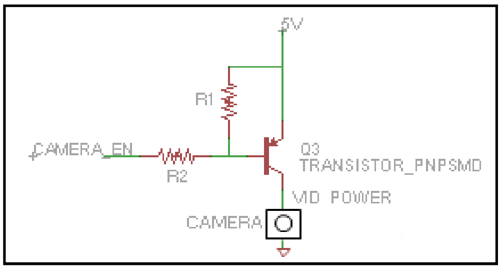


Figure 10 Camera on/off switch

Referring to Figure 10, and the PNP transistor's datasheet [34], R1 and R2 can be calculated for the 'on' case as follows:

$\beta = 200$, $V_{BE} = -1.2V$, $I_{EC} = 120mA$ (to be safely above 100mA)

$$I_B = \frac{IEC}{\beta} = \frac{120mA}{200} \qquad (1)$$

$$V_B = 5 + V_{BE} \qquad (2)$$

Using Kirchoff's current law:

$$\frac{VB-5}{R1} + \frac{VB-0}{R2} - I_B = 0 \qquad (3)$$

R1 is chosen as 1kΩ because it is a common value which is used in several places in the SUNSET payload.

Therefore, by rearranging the equation to make R_2 the subject, $R_2 = 2.11k\Omega$

Next, the 'off' case is tested. First, I_B is assumed to be zero. This means that R_1 and R_2 form a voltage divider.

$$V_B = (5 - 3.3) \cdot \frac{R2}{R1+R2} + 3.3 \qquad (4)$$

From the above equation it is calculated that V_B will equal 4.45V, making $V_{BE} = -0.55V$. This is more than $V_{BE(on)}$ of -1.2V specified by the datasheet, meaning the transistor will be off. Even if I_B is not zero, current flowing out of the base of the transistor will flow through R_2, further increasing V_B and V_{BE}. Thus, the transistor cannot be on. This circuit was tested and verified using LTSpice simulation software.

3.2.8 Power

As can be seen from Table 7, which lists the power requirements of the major SUNSET components, both 5V and 3.3V regulators are needed onboard. The power regulation circuit consists of an LM7805 5V regulator IC, followed by an LD33V 3.3V regulator IC. Decoupling and output capacitors are used as specified in the regulator datasheets.

Table 7 Individual power requirements of SUNSET payload components

Component	**Supply Voltage (V)**	**Power usage specifications**	**Usage onboard SUNSET**	**Average over 1 hour (mA)**
Parallax Propeller	3.3	Assume 4 cogs at 80MHz: 45mA	constant	45
Radiometrix HX1	5	TX supply current: 140mA Standby: not specified	96 packets/hr 3 seconds each. 6 images /hr 2 min each	25.2
GSM Modem	3.3	Standby: 3mA Transmit: 260mA	6 SMS's/hr Guess 2 seconds each	3.87
GPS	5	Sustained Supply Current: 36mA	constant	36
Camera	5	Specs not available 96mA measured	ON for 6 images/hr 2 min each	19.2
SD card	3.3	100mA max during write	96 + 6 writes/hr Guess 0.5 seconds each	1.42
Total				**130.69**

Both regulators will be screwed to the copper clad PCB, which will act as a heat sink and keep the other components on the PCB warm. The average total current consumed by the electronics of 130.69mA is well within the LM7805's range. According to Equation 5, the maximum current the LM7805 will have to supply is 677mA, which is well below the voltage regulator's specified maximum continuous current of 1.5A.

$$45+140+260+36+96+100\text{mA} = 677\text{mA} \qquad (5)$$

The SUNSET payload needs a supply voltage of at least 7.5V as the LM7805 has a maximum dropout voltage of 2.5V. Using a 9V power source will cause the voltage regulators to dissipate a lot of heat, which is useful for warming the payload electronics. Together, the regulators will dissipate 0.608W of heat.

Energizer lithium batteries have several properties which make them ideal for near-space flights [35]:

- Proven flight experience on BEAR3 and ICARUS
- Operating range specified to -40 degrees Celsius
- Extremely lightweight: 9V = 33.9g, AA = 14.5g
- High capacity: 9V = 750mAh, AA = 3000mAh (at a constant discharge rate of 200-250mA)
- Readily available and cheap compared to rechargeable lithium polymer batteries. 9V costs about R100 and 2xAA batteries cost about R50.

A 9V Energizer battery could power the SUNSET payload for approximately 750mAh/130.69mA = 5.7 hours. The other option is to use six AA Lithium batteries, which could power the payload for approximately 3000mAh/130.69mA = 23 hours. Either option is feasible. Six AA batteries will give the recovery team more time to find the payload if it lands out of line of sight.

3.2.9 Firmware

The firmware will be a combination of existing objects from the OBEX, modified objects, and custom objects and code written in Spin. The main code loop will continuously receive and parse GPS data, read the temperature sensors, take images and construct and transmit telemetry packets. A telemetry packet will be transmitted every thirty seconds. An image will be transmitted once every ten minutes. If the GPS sensor indicates that the payload is below a certain altitude, an SMS message will also be sent once every ten minutes. More detailed information on the firmware can be found in Section 4.2 of this report. A flow chart of the firmware is available in Appendix E.

3.3 Flight String

The flight string consists of an insulated electronics container (the payload), a parachute and a hydrogen filled, latex balloon, tied together with nylon string. The balloon will carry the payload up to about 30km altitude, at which point the balloon will burst and the payload will fall back to Earth, slowed by the parachute. Alternatively, the flight can be terminated by a release mechanism, described in section 3.3.4, which mechanically separates the payload from the balloon.

3.3.1 Insulated Electronics Container

Even in temperatures as cold as -50 degrees Celsius in near-space, the insulated electronics container must keep the electronic components above their rated minimum operating temperatures. These temperatures are listed in Table 8.

Table 8 Rated minimum operating temperature of SUNSET components

Component	Min operating temp	Notes
Parallax Propeller	-55	
Radiometrix HX1	-10	Produces heat while transmitting. Frequency is affected by temperature.
GSM Modem	Not specified	Only needs to function at low altitudes
GPS	-40	
Camera	Not specified	Lens is exposed to the air and not insulated. Not an essential component for successful recovery
LM1881	0	Will be positioned between the regulators and the radio transmitter for maximum warmth
SD card	-25 [36]	
Batteries	-40	

The essential components for recovery of the SUNSET payload are the Parallax Propeller, GPS sensor, batteries and either the Radiometrix HX1 or the GSM modem. While the Propeller, GPS sensor and batteries are rated to function at very low temperatures, other components, such as the camera need to be kept warm. The electronics container should be made of an insulating material, which is rigid enough to contain the electronics and cushion them on impact. The electronics container should also be aerodynamic to avoid excessive motion caused by winds. Many near-space launches, such as Icarus and BEAR3 have used polystyrene boxes with success.

SUNSET will use a section of "Pool Noodle" as its electronics container. First, the Pool Noodle will be cut in half length ways. Then hollows will be cut out for the various electronics components. Once the components are in place, the two halves of the Pool Noodle will be recombined and held together with elastic bands. The foam material of a Pool Noodle is similar to polystyrene, but provides more shock absorption. A Pool Noodle is also aerodynamic as it does not have any flat surfaces which can catch the wind. For extra insulation, one or two layers of "Space Blanket" can be wrapped around the Pool Noodle, separated by thin layers of sponge. The bubbles of air in the Pool Noodle material and in the sponge give the materials their good insulating properties. Finally, the Pool Noodle will be inserted into a brightly coloured material sleeve which will aid the recovery team in spotting the payload. The sleeve will be tied to the balloon with nylon string. Figure 11 depicts the structure of the electronics container.

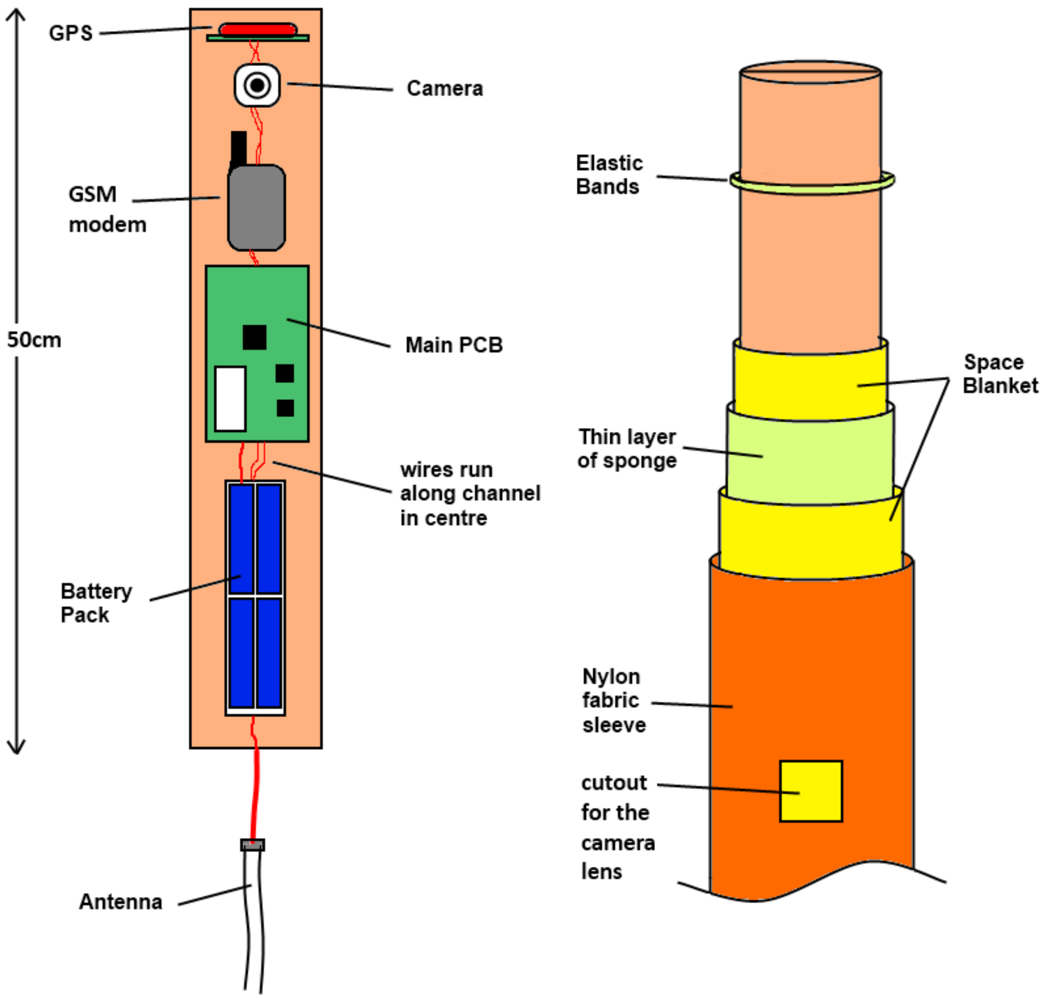


Figure 11 Diagram of the insulated electronics container

3.3.2 Parachute

The parachute needs to slow the descent of the electronics container as it falls back to Earth after being released from the balloon. The parachute will not need to deploy or unfold. It will hang open as the payload ascends and will flip around to slow the payload's descent as soon as the balloon bursts, or the release mechanism is activated. Many near-space projects have used model rocket parachutes. Unfortunately, due to time and monetary constraints, the SUNSET parachute will have to be constructed quickly from cheap materials.

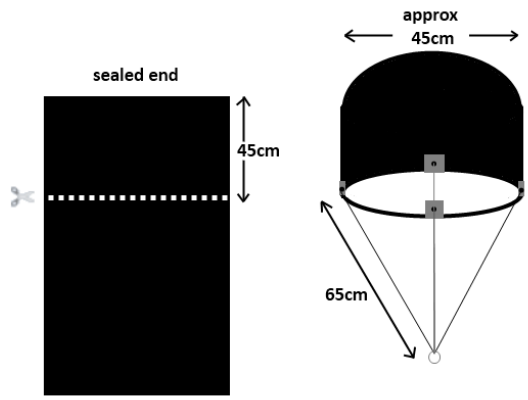


Figure 12 Plans to construct a parachute from a refuse bag

A black refuse bag can be turned into a parachute by following the steps shown in Figure 12. First a refuse bag is cut across its width 45cm from the sealed end. When inflated with air, the sealed end of the refuse bag forms a rough dome shape. Four pieces of kite string are attached to the refuse bag, evenly distributed around the open end's circumference. The attachment points are reinforced with strips of duck tape to avoid tearing. The other ends of the four strings are tied together with a keychain ring. This creates a parachute with a diameter of approximately 45cm. The materials cost less than R10. The following equation from a model rocketry website will give the descent speed of a payload using a parachute with the same diameter as this homemade parachute [37]:

$$v = \sqrt{(8mg)/\pi\rho CdD^2} \qquad (6)$$

Where m is the mass of the payload, g is the acceleration of gravity (9.81m/s^2), ρ is the density of air (1.22kg/m^2), Cd is the drag coefficient of the parachute (1.5), and D is the diameter of the parachute (0.45m) [37].

Model rocketry sites suggest descent velocities of no more than 4.57m/s (15feet/s) [38]. According to this equation, a payload with a mass of 250g will fall at 4m/s, and a payload with a mass of 500g will fall at 5.8m/s. Therefore, to comply with model rocketry standards, the SUNSET payload should weigh no more than 310g. However, tests will have to be conducted on the effectiveness of this proposed parachute and the extra shock absorption provided by the Pool Noodle electronics container.

3.3.3 Balloon

The latex weather balloon that will carry the SUNSET payload into near-space was acquired from the South African Weather Bureau. Unfortunately, its specifications are not known, except that it weighs 420g uninflated. The balloon will be filled with Hydrogen. Hydrogen is both cheaper and provides twice as much lift as Helium (not four times as much as would be suggested by the periodic table, because Hydrogen forms diatomic molecules). The only disadvantage of Hydrogen is that it is explosive when mixed with Oxygen. However, as long as adequate care is taken during the filling of the balloon to avoid static charge build-up, the pure Hydrogen inside the balloon does not pose a danger.

According to Nova Lynx [39], 420g does not seem to be a standard weather balloon size. However, they do provide tables listing the specifications of both 300g and 500g weather balloons. According to them, a 300g weather balloon, when filled with $2.1m^3$ of Hydrogen, will provide 2330g worth of gross lift. The payload's weight and the balloon's weight need to be subtracted from this to get the free lift. A maximum payload weight of 630g is suggested. The reason for the lower payload weight specification is to ensure a fast ascent. The more free lift is available, the greater the upwards force exerted on the payload and the faster the ascent. A fast ascent is good as it gives the balloon less time to travel in the wind.

The SUNSET payload will likely weigh far less than 630g, allowing the balloon to be filled to less than maximum. This will save time and money.

3.3.4 Release Mechanism

A novel release mechanism is proposed. A syringe will be half filled with air, before the tip is sealed. At this stage the plunger should be half extended. Attempts to pull the plunger out of the syringe will be met with resistance due to the fixed amount of air trapped in the syringe. However, as the atmospheric pressure around the syringe decreases, as occurs when going up in altitude, the volume of air inside the syringe will increase. Eventually, when the atmospheric pressure is low enough, the plunger will be forced out of the syringe. By attaching the balloon to the plunger, and the payload to the syringe body, an altitude release mechanism can be constructed. By varying the initial volume of air sealed into the syringe, the altitude of release can be set. This proposed release mechanism will require feasibility testing in a vacuum chamber. More information can be found in Section 4.4.3.

3.4 Ground Station

The ground station must receive, decode and display information transmitted from the SUNSET payload. The entire ground station must be portable and fit into a standard sized car, so that the recovery team can chase the SUNSET payload during its flight. The ground station should not consist of any specialised equipment. This will allow several different receiving stations to receive SUNSET transmissions for redundancy.

A JingTong VHF handheld radio, tuned to 144.800MHz, will receive the SUNSET transmissions. The JingTong radio will be connected to a flexible J-Pole antenna, mounted in a PVC tube. This configuration will allow the antenna to be stuck out of a car window.

The audio out jack of the JingTong radio will be connected to the microphone jack of a laptop computer. Two commonly used third party programs exist which can decode packet transmissions using the laptop's soundcard. AGW Packet Engine (AGWPE) and MixW can both emulate hardware terminal node controllers, forwarding the decoded data to other programs. Both AGWPE and MixW can be run simultaneously. MixW seems to be more robust in decoding packets. It is not as sensitive to the audio volume. However, there are far more mapping and logging programs available which interface to AGWPE. For this reason, both AGWPE and MixW will be used on the ground station. AGWPE will forward decoded packets to UIView32, which will use the GPS coordinates in telemetry packets to plot the balloon's track on a map. MixW will forward decoded packets to a custom written Visual Basic 2008 program, which will display the information from the telemetry packets, as well as reconstruct and display the images transmitted by SUNSET.

4. Implementation

4.1 PCB Design and Population

The circuits described in the Design section were implemented on a custom designed and manufactured, two-layer Printed Circuit Board (PCB). The free version of Eagle Layout Software version 5.11.0 was used to design the layout of the PCB and to generate the Gerber files. Initially the PCB was supposed to be long and narrow to fit into the Pool Noodle. Unfortunately, the free version of Eagle has a maximum PCB size of 10x8cm. With a maximum of 10cm in length, the PCB could not be made as narrow as originally planned. The Pool Noodle is 7cm in diameter, and the PCB ended up being 5.3cm wide. This leaves just under 1cm worth of insulation on either side of the PCB.

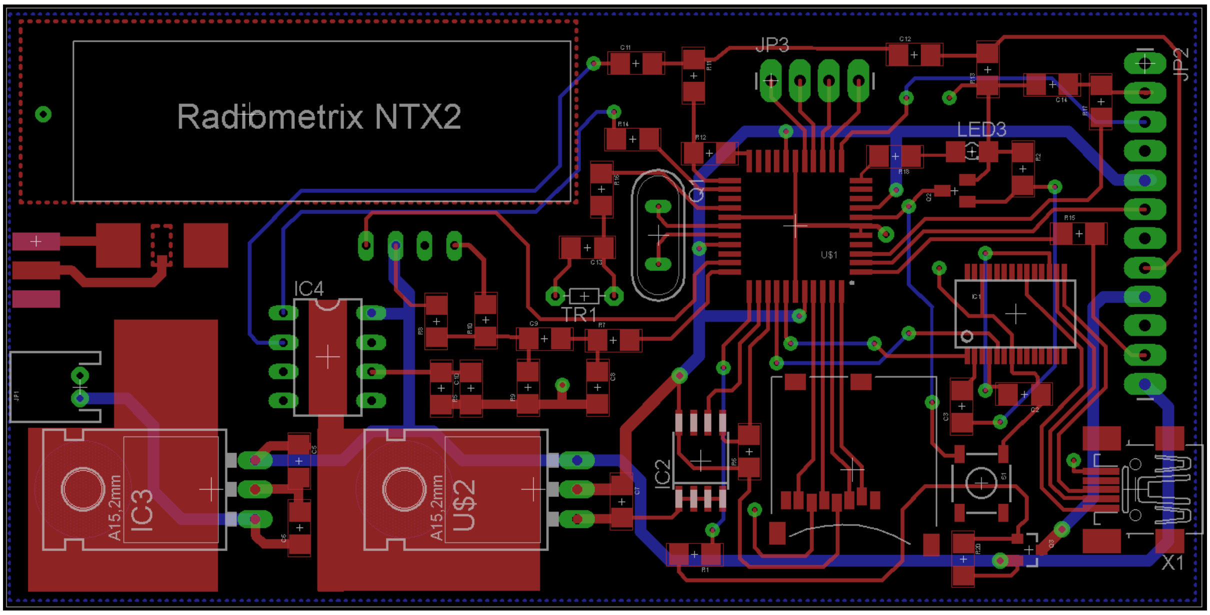


Figure 13 Eagle PCB design showing top and bottom layers simultaneously

Figure 13 shows both layers of the Eagle PCB design. Top and bottom PCB masks are available in the Appendix F. The power tracks were routed first, as the Parallax Propeller requires a good, clean power supply. All capacitors and resistors are 1206 surface mount components to minimise potential issues during component ordering. 1206 size components are also large enough to solder without specialised equipment. The amount of vias was minimized, as the Electronic Engineering Department of Stellenbosch University does not have the facilities to create PCBs with vias. Small wires have to be soldered into the position of vias. The bottom side is filled with a ground plane to make the layout process easier. A copper area is extended from beneath the 3.3V regulator, to beneath the LM1881 to transfer some heat and keep the LM1881 warm. A ground plane also exists under the Radiometrix HX1, as recommended in its datasheet [26].

All connections to peripherals were brought out to a twelve pin header on the right hand side. Starting from the top, the functions of the header pins are: gnd, external temperature sensor, spare I/O, gnd, 3.3V, GSM modem serial, gnd, video in, 5V, gnd, GPS serial, 5V. The header was designed in such a way that every peripheral can connect to the PCB with a three pin connector including power and data. The spare I/O pin can possibly be used to trigger a realease mechanism in future. Four more I/O pins are broken out to a header at the top right. These allow for future expansion. The internal temperature sensor, TR1, is mounted in the middle of the PCB, to monitor the temperature inside the insulated electronics container. A mini USB socket is mounted on the bottom right.

The PCB was manufactured by a PCB Computer Numerical Control (CNC) router. To prolong the life of the router bits, the router does not cut away excess copper, it simply cuts a thin line of insulation around every track. The PCB was populated in stages. Each stage was tested before the next stage was added. First the two voltage regulators and their associated capacitors were soldered on. Four 0805 size, 100nF decoupling capacitors were also added to the underside of the board near the Propeller power supply pins. These are not shown in Figure 13. The voltage regulation circuit was tested and confirmed to be working. A ground plane island existed beneath the Propeller IC. This was fixed with a bridging wire.

Next, a solder specialist in the Electronic Engineering Faculty mounted the Parallax Propeller IC, the FT232RL IC, the EEPROM IC and the USB socket. The 5MHz crystal, LED and two resistors were added, too. After a lot of trouble trying to solder the tiny reset transistor, and after reading through several posts on the Parallax forums about the reset transistor, it was concluded that the reset transistor could be removed and replaced with a single resistor without any negative effect on the SUNSET payload [40]. Therefore, instead of the transistor, 1kΩ resistor was put in series between the FT232RL's DTR pin and the Propeller's reset pin. Unfortunately, the wrong reset button had been ordered, and could not be included. This is not a major problem, as the reset button would have been removed for launch anyway. The removal of the reset switch for launch would have been a safety precaution to prevent the Propeller from being accidentally reset during turbulence. At this point, the PCB was connected to a computer with a USB cable and the Propeller was programmed to flash the LED. The test confirmed that the Propeller, EEPROM, and USB to Serial communication were working.

The next step was to add the Radiometrix HX1 and its interface to the Propeller. The Propeller was programmed to repetitively transmit a text APRS packet. A 421mm long piece of wire was attached to the Radiometrix to act as a temporary antenna. The JingTong radio was turned on and tuned to 144.800MHz. The transmission from the Radiometrix HX1 could be heard, but only barely. The volume could not be turned up any higher. After removing the 100kΩ volume control resistor on the PCB, the received transmission was louder, but still too soft. It was discovered that the best transmissions occurred when the Propeller I/O pin was connected directly to the Radiometrix's data pin. It appears that the process of transmitting, receiving and digitising the Propeller's transmission has enough of a low pass filter effect that the hardware low pass filter on the PCB is not necessary. The resistor voltage divider also appears to be unnecessary. The Radiometrix HX1 seems to be happy with a signal with a one volt swing around 1.65V, which is what the modified Bell 202 object outputs. Connecting the Propeller I/O pin directly to the Radiometrix HX1 makes the hardware even simpler.

Once the packet transmission was working, the twelve pin header and GPS were added. After integrating the code to read and parse the GPS, a problem was found. The GPS outputs 3.3V serial data, not 5V data as was expected. The 1kΩ series resistor was removed and the Propeller could successfully communicate with the GPS. At this point, a program was written which repeatedly reads the GPS sensor and transmits an APRS packet with the current coordinates. A test to determine communication distance and GPS tracking was run. This test is described in detail in the Testing and Results section.

Next, the camera interfacing circuitry was added to the PCB. The camera came with a built in 3.3V regulator. Unfortunately, it was found that this regulator did not work well with a 5V input voltage. The internal regulator was removed and replaced by an LD33V 3.3V regulator. This solved the problem.

Finally, the temperature sensor and RCTIME circuitry was added, and the solder specialist soldered on the micro SD card adapter. During the PCB layout process, one via mistakenly ended up underneath the SD card slot, preventing the SD card slot from being mounted flush against the PCB. Fortunately, this via could be removed as it related to the non-existent reset switch.

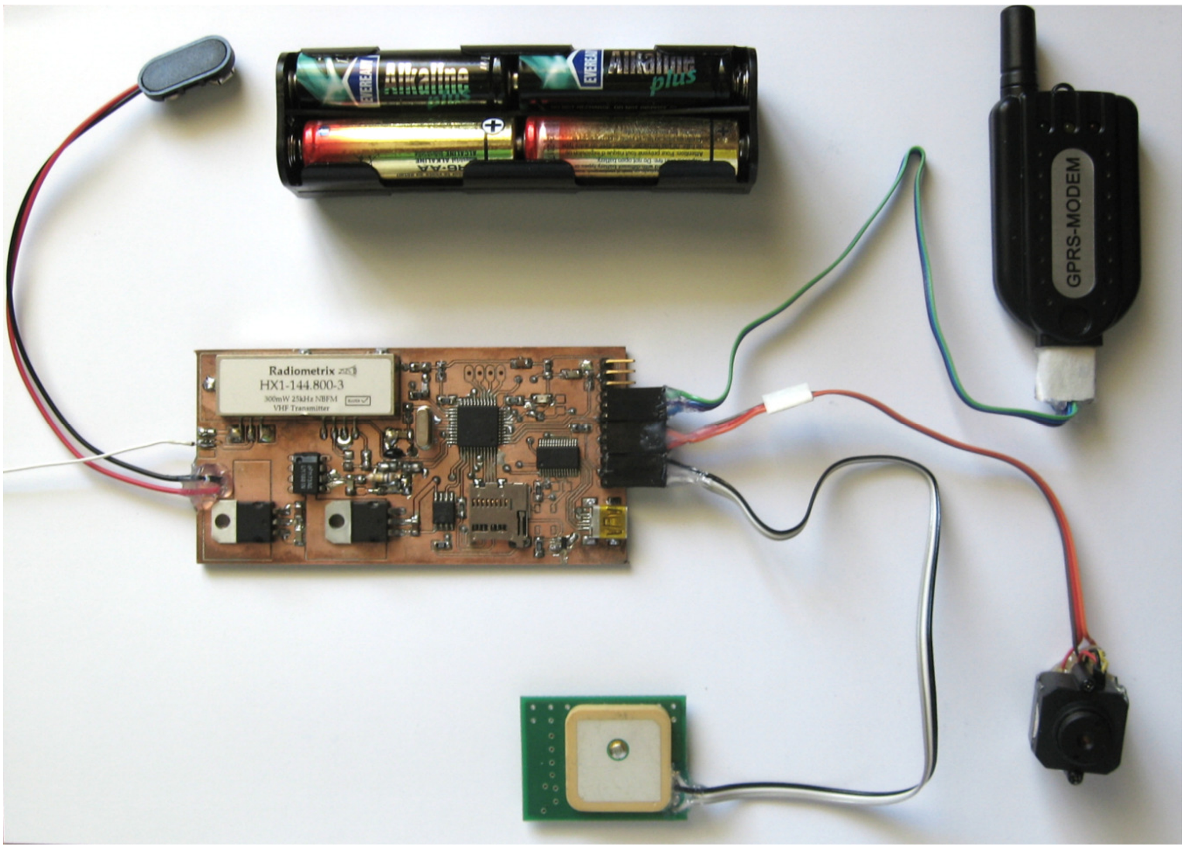


Figure 14 Completed PCB connected to GPS sensor, camera and GSM modem

Figure 14 shows the completed PCB and its peripherals. The three empty header pins are the connection to the external temperature sensor (not shown). The batteries shown in the image above are normal alkaline AA batteries, only used during testing. These will be replaced with Energizer Lithium batteries for launch.

4.2 Firmware

Appendix I contains a brief description of all the methods that compose the SUNSET firmware. The methods and objects are colour coded. Green objects and methods are almost entirely custom written. Yellow objects and methods contain modifications to code from the Parallax OBEX, and blue objects and methods are used as is from the Parallax OBEX. Any code which was not custom written contains the author's name in the source code.

Appendix I shows that approximately fifty percent of SUNSET's source code is composed of prewritten objects. As planned, this significantly reduced the firmware's development time. However, during the firmware development, one of the dangers of using other people's code became apparent. The *video_capture* object gave a lot of problems. It is very dependent on exact timing. Efforts to modify it to increase the resolution of captured images failed. The *video_capture* object is very sensitive to code changes, even to parts of the program which seem completely unrelated to it. The danger of using other people's code is that the source code is not fully understood, and changes can have unexpected consequences. During the first tether test, described in Section 5.2, the *video_capture* object got stuck in an endless loop for unknown reasons. As discussed below, concurrency problems were most likely to blame for the endless loop. After the test, an effort was made to better understand the *video_capture* object. Changes were made to the rest of the firmware so that the *video_capture* code was called in a manner and sequence very similar to the original author's program. This solved the problems with the *video_capture* object.

The Parallax Propeller's ability to run different methods on separate cogs was initially seen as an advantage. If one of the cogs got stuck waiting for a byte from the GPS, the other cogs could continue to function. This would prevent the whole system from locking up. However, during the first SUNSET tether test, concurrency issues kept appearing. All of the cogs can access the same Hub RAM. Although read and write instructions are atomic, multiple cogs accessing the same memory locations sequentially can still interfere with one another. Understanding the sequence of events in code with multiple threads becomes very difficult. After tether test 1, the extent of parallisation in the SUNSET source code was reduced to make the code easier to debug and more reliable. The GPS parsing was changed from running on its own cog, to running on the same cog as the main loop. This made the code more reliable, but could cause the whole program to freeze if the GPS is unexpectedly unplugged. This was judged to be acceptable, as if the GPS becomes unplugged, the hopes of finding the payload become very small anyway.

4.3 Temperature Sensor Calibration

Getting temperature measurements from the thermistors involves two transfer functions. The thermistor has a transfer function from temperature to resistance. The second transfer function is from the thermistor's resistance to the integer result of the RCTIME A/D conversion. Both of these transfer functions need to be characterised. Fortunately, the RCTIME object includes a list of resistance-A/D result pairs. By plotting these points on a graph in Microsoft Excel and curve fitting, the following transfer function was obtained:

$$R = 2.360x + 568.2$$

Where R represents the resistance of the thermistor in Ohms, and x represents the result of the RCTIME A/D conversion. To get the thermistor's transfer function was more difficult. The thermistor component has no markings on its body. It was thus not possible to find its datasheet. To characterise its transfer function, the thermistor was placed into a temperature controlled chamber. Measurements of the thermistor's resistance were made at temperatures ranging from 20 degrees Celcius to -120 degrees Celcius. Once again the datapoints were plotted on a graph in Excel, and a curve was fit to the result. The following transfer function was obtained:

$$T = -18.6\ln(R) + 496.6$$

Where T represents temperature in Kelvin, and R represents the thermistor's resistance. The first transfer function was easy to implement in software. However, calculating the natural logarithm on a microcontroller is difficult. Fortunately, the Parallax Propeller contains a logarithm lookup table in ROM. A method was found on the Parallax forums which makes use of this lookup table to calculate the natural logarithm. By making use of these two transfer functions, the integer result of the RCTIME A/D conversion is converted into a temperature measurement in Kelvin. The complete calibration values and graphs are available in the Appendix J.

4.4 Flight String

4.4.1 Insulated Electronics Container

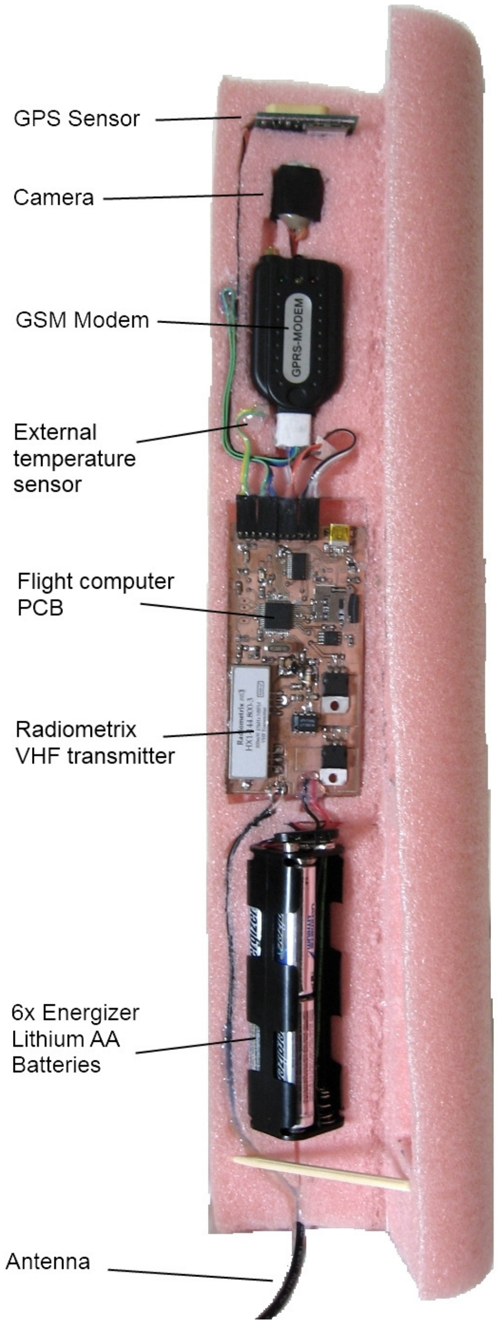


Figure 15 The component layout inside the insulated electronics container

Figure 15 shows the position of all the electronic components of the SUNSET payload. A 45cm section of Pool Noodle was cut in half and hollows were cut out for the electronic components, as designed. A knife was used to do the cutting. When the two halves of the Pool Noodle are brought together and held in place with elastic bands, the electronics are held firmly in place. The camera lens and the external temperature sensor protrude through the Pool Noodle to the outside (not visible in Figure 3). The antenna coaxial cable is glued to the Pool Noodle with hot glue. This is necessary as significant forces may be exerted on the antenna by winds. The antenna must not pull loose from the PCB.

A bright orange tube, with one closed end, was sown out of light-weight nylon fabric. A hole was cut on the side for the camera lens and on the bottom for the antenna. Two rubber bands put over the fabric sleeve hold the camera hole in place in front of the camera lens. The additional layers of insulation had to be left off for tether tests, as the outside temperature was in the high twenties.

4.4.2 Parachute

The parachute was implemented as designed. It will hang limp during ascent, and should open during descent. No deployment or unfurling of the parachute will be necessary. The parachute was successfully tested as described in Section 5.4.

4.4.3 Release Mechanism

The release mechanism was implemented using a syringe, as designed. By inserting sensor data from the HALO flight into the ideal gas equation, it is possible to predict how much the air in the syringe is expected to expand at different altitudes. Table 9 shows the predicted percentage increase in volume at different altitudes. From the table it is evident that even a minimal amount of air will be sufficient to cause the plunger to pop out of the syringe at high altitudes. Unfortunately, the engineering department's vacuum chamber was unavailable for tests during the development of the release mechanism. Therefore the release mechanism will have to be tested during the final flight.

Table 9 Expected expansion of the air inside the release mechanism syringe

Altitude (m)	Pressure (Pa)	Temperature (K)	Percent Increase (%)
2000	100k	303	0
15000	10k	228	752
30000	300	258	28300

4.4.4 Weight

The final weight of the SUNSET payload is 265g. The complete flight string, including payload, parachute and string, weighs 290g. Table 10 shows the payload weight breakdown. Compared to other near-space payloads, such as HALO, ICARUS and HABEX which all weigh 800g or more, the SUNSET payload is very light weight. SUNSET manages to be light weight while still including far more than just the essential tracking components. A 420g weather balloon should easily be able to lift the SUNSET payload.

Table 10 Payload weight breakdown

Component	Weight (g)
Insulated electronics container	30
PCB	35
GPS	15
6x AA Energizer Lithium batteries in a battery case	105
GSM modem	30
Camera	15
Other (including antenna)	35
Total	**265**

4.5 Ground Station

The ground station laptop runs four pieces of software. AGWPE decodes packets using the laptop's soundcard, and forwards them to UIView32. UIView32 uses the GPS coordinates in SUNSET telemetry packets to plot the balloons current position, and track, on a map. Meanwhile, MixW also decodes packets using the laptop's soundcard, and forwards them to a custom program written in Visual Basic 2008. UIView32 has been written to automatically detect and communicate with AGWPE. The custom SUNSET ground station software listens for decoded packet data on a serial port. MixW can emulate a hardware Terminal Node Controller (TNC) by outputting its decoded data on any specified serial port. It would be possible to connect MixW and the custom SUNSET ground station software using a serial cable if the laptop had two hardware serial ports. The more elegant solution is a software serial port bridge such as *MixW Serial Port Bridge*. This piece of software installs as a hardware device. In Windows Device Manager it appears under Multi-port serial adapters. MixW Serial Port Bridge allows you to specify two virtual serial ports. It will then connect the two with a virtual cable. MixW is configured to output its data on one of the two serial port bridge ports, and the custom ground station software listens on the other one.

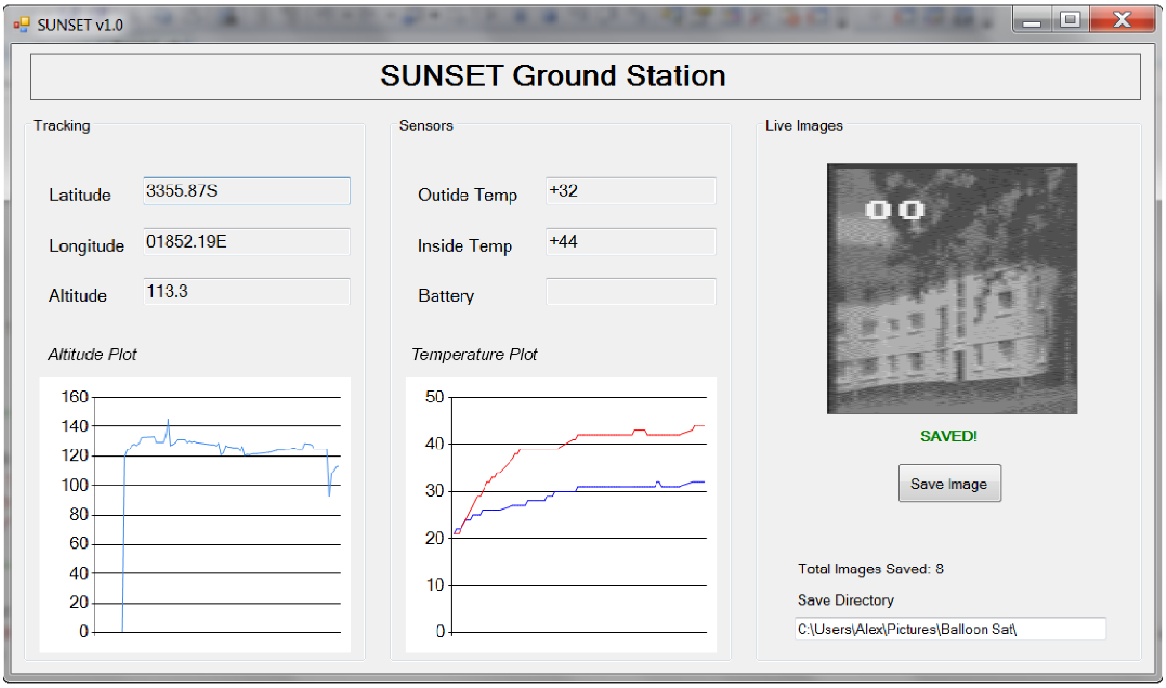


Figure 16 A screenshot of the SUNSET ground station custom software

Figure 16 shows a screenshot of the custom ground station software. The window is divided into three segments: Tracking, sensors, and live images. The Tracking segment displays the payload's current GPS coordinates and altitude. The graph at the bottom plots the payload's altitude against time. This graph gives a quick overview of the altitude data, making it easy to detect the moment the payload begins its decent. This tracking information serves as a backup to UIView32. UIView32 displays the payload's position on a map, and gives additional information such as speed and heading. The sensors segment displays the SUNSET payload's current internal and external temperature measurements. A plot of both temperatures versus time is plotted at the bottom of the segment. The red line represents internal temperature, and the blue line represents external temperature. A box to display the payload's current internal battery voltage is also included, but was never implemented in the SUNSET hardware. The live images segment, on the right, displays images line by line as they are transmitted by the SUNSET payload. The horizontal resolution is interpolated to 120 pixels for better viewability. Interpolation is achieved by making every odd pixel equal to the average value of its two neighbours. When either the last or second last lines of an image are received, the image is automatically saved to the laptop's hard drive. Alternatively, the Save Image button can be clicked at any time to save the current incomplete image.

The custom ground station software beeps whenever a packet is received. This is useful while preparing the payload for launch, as the audible beeps in the background are a reminder that the payload is still operating. The data from every packet received is appended to a text file which is stored on the laptop's hard drive.

The ground station hardware was implemented as designed.

5. Testing and Results

The full raw data from each test is available in the Appendix K.

5.1 Communication Range and GPS Test

This test was intended to verify the working of the GPS sensor, and to establish the range of the communication link. The ground station consisted of a laptop running AGWPE and UIView32, and a JingTong VHF radio connected to a 5/8 wave 2m amateur radio antenna. The ground station was placed on the roof of the five-story Stellenbosch Engineering building for maximum line of sight. The half populated SUNSET PCB and GPS sensor were placed inside the insulated electronics container and powered with six AA Alkaline batteries. The electronics container was suspended in the open sunroof of a car using string. A flexible J-Pole antenna mounted in a piece of PVC tubing was connected to the SUNSET PCB. The PVC tube containing the antenna was stuck out of the open sunroof of the car. The SUNSET PCB transmitted an APRS packet containing its current GPS coordinates every thirty seconds. While the ground station received, decoded and plotted the GPS coordinates on a map, the car containing the SUNSET PCB was driven towards Somerset West.

Figure 17 is two screenshots of UIView32. It shows the plotted track (in red) of the car containing the SUNSET PCB. The image on the left shows the accuracy of the GPS sensor. It is possible to determine the exact route taken by the car during testing. The image on the right is a zoomed out view of the same track. The car began moving before the GPS sensor had a position lock. The first position it reported was off by several kilometres. This caused the strange spike in the track below. After this one incorrect position report, the GPS sensor became more accurate, and UIView32 began correctly plotting the track of the car. The car began behind the engineering building, headed towards Somerset West, and came back to its starting point.

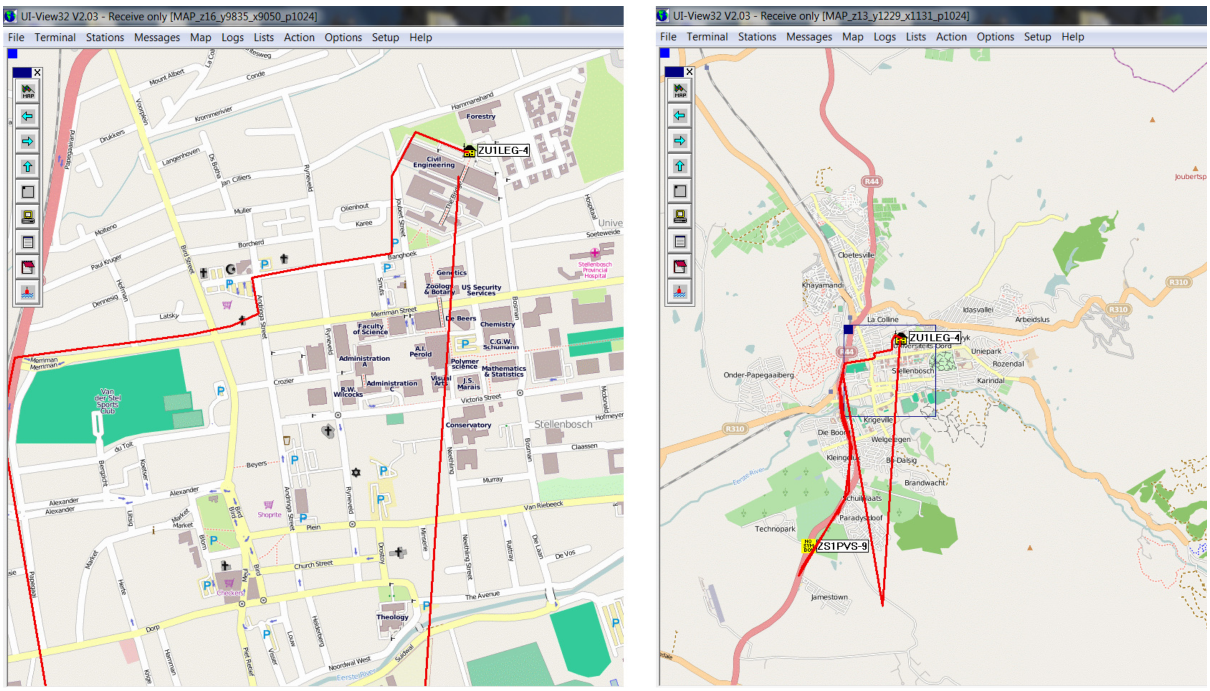


Figure 17 UIView32 screenshots displaying the track of a car containing the SUNSET PCB during GPS testing

From the image on the right it is possible to determine the maximum range achieved by the Radiometrix HX1. A packet was heard from just beyond Techno Park, which is a distance of approximately 5.1km. During the SUNSET flight, the communications link may have to work over a distance of greater than 100km. While the 5.1km achieved during this test is far less than 100km, it was achieved without line of sight. Personal experience has shown that a 2W JingTong radio which can barely make 20km without line of sight, can successfully communicate with the International Space Station more than 300km away when line of sight is possible [41]. Further, the Radiometrix HX1 transmitter has been successful on a number of near-space flights, including BEAR3 and SABLE3 [42]. During the flight of SABLE3, a packet from the Radiometrix HX1 transmitter was heard more than 600km away [42].

This test proved the working of the GPS unit and the radio communications. Despite a distance of only 5km without line of sight being achieved by the Radiometrix HX1, the test proved that the APRS communication system works, and should work satisfactorily during the launch.

5.2 Tether Test 1

Both tether tests took place on the grass field behind the engineering building. The first tether test took place on Saturday afternoon, 8 October 2011. The test was intended to simulate an actual launch. The SUNSET PCB was fully populated and all features were activated. The SUNSET payload was programmed to transmit a telemetry packet every ten seconds, and an image every two minutes. An SMS containing the GPS coordinates would also be sent every two minutes. The timing interval between transmissions was made short to speed up the test. The only components of the launch string which were missing were the parachute, release mechanism and external temperature sensor.

The weather balloon was filled with Hydrogen using a regulator. Due to a weak but constant wind (about 5km/h), the balloon had to be topped up a bit more to achieve a satisfactory lift. After the top up, the “nipple” on the top of the balloon had disappeared. The SUNSET electronics were switched on before the insulated electronics container was put into the fabric sleeve and was tied to the balloon using kite string. At approximately 5pm, the balloon and flight string rose 30m into the air, secured to the ground with a tether made of kite string.

Unfortunately, the test did not go well. Packets were successfully being received by the ground station. However, the telemetry indicated that the GPS sensor was not getting a position lock, even after thirty minutes with a clear view of the sky. Another problem occurred during image transmission. Although MixW indicated that all the image packets were being correctly received and decoded, only about fifty percent of the packets were being received by the custom ground station software. This resulted in images with many missing lines, as can be seen in Figure 18:

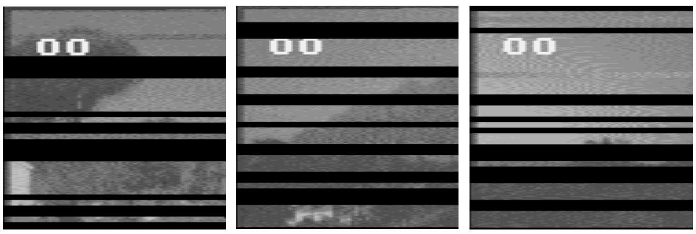


Figure 18 Images received during Tether Test 1 show large packet loss

A more serious anomaly also occurred during image transmission. After attempted quick-fixes to the firmware while in the field, the modified firmware was loaded onto the Parallax Propeller. This time the Parallax Propeller froze during image transmission, continuously broadcasting the same tone. This would be disastrous during an actual flight. The payload would be untrackable and the continuous broadcast would prevent other APRS transmissions in the area from being heard.

Despite these problems, the test did verify a few important aspects of the SUNSET project:

- The SUNSET payload is light enough to be lifted by a single 420g weather balloon.
- The camera onboard the payload functions and can transmit images. The packet loss problem was somewhere on the ground station side.
- The SUNSET payload successfully transmits temperature readings. These are plotted by the custom ground station software.
- Images and temperature readings are successfully saved to the micro SD card onboard the payload.
- The GSM modem functions correctly, as all the SMS messages sent by the payload were successfully received.

Tether Test 1 lasted approximately one and a half hours. After the test, the problems experienced during the test were investigated. The GPS parsing code on the Propeller was modified to execute in the main program loop, instead of on a separate cog. This solved concurrency issues. The problem of packet loss on the ground station was also tracked to concurrency problems in the custom ground station software. One thread receives serial data and stores it in an array. Another thread reads data from this array and displays it on the Graphical User Interface (GUI), before clearing the array. Packet loss occurred whenever these two threads tried to modify the data in the array simultaneously. The thread which displays the data was modified to only access the array when no data was present in the serial receive buffer. This solved the problem.

5.3 Tether Test 2

Tether Test 2 was conducted at 9am on Sunday, 9 October 2011. The outside temperature was in the high twenties, and there was no noticeable wind. After Tether Test 1, the balloon was not deflated. Instead it was stored indoors overnight. The balloon had no noticeable loss of lift when it was used again the following day for Tether Test 2. The test setup was identical to Tether Test 1, except for new firmware and updates to the ground station software.

This time everything worked perfectly. The GPS sensor got a position lock and complete images were received and displayed by the custom ground station software. The balloon was aloft for approximately forty five minutes, before being brought down. Figure 19 is a screenshot of the custom ground station software at the end of Tether Test 2. The altitude plot shows that the GPS altitude data is not very accurate. It varies by about 100m. Occasionally the GPS sensor momentarily lost its GPS lock, resulting in a zero altitude reading. The temperature plot shows that the inside temperature of the payload rose from 26 degrees Celsius to 51 degrees Celsius during the duration of the test. To further understand the effectiveness of the payload insulation, the external temperature sensor must be added and its measurements compared to those of the internal sensor.

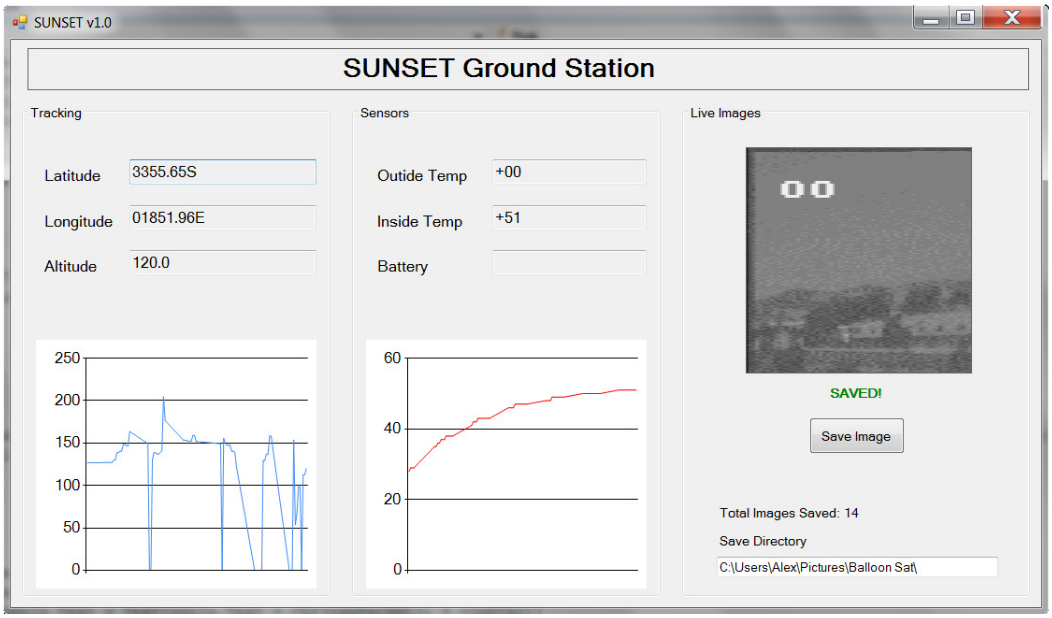


Figure 19 A screenshot of the custom ground station software at the end of Tether Test 2

Seven complete images were received during the duration of the test (each image is saved twice, once when the second last line is received and again when the last line is received). Figure 20 is a compilation of five of the images received by the ground station during Tether Test 2. The payload swayed gently beneath the balloon and captured images in all directions. While this test proved that images can be transmitted from the SUNSET payload, it also showed the limitations of the current system. The subject matter of the images can mostly be discerned, but the low quality of the images restricts their usefulness. The quality is good enough to see major features such as buildings, mountains and trees, but offers little detail.

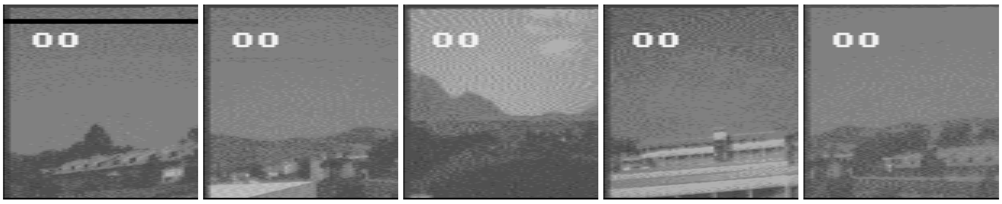


Figure 20 Five of the images received by the ground station during Tether Test 2

The digits overlaid on the images were intended to increment with every transmitted image. However, this had not yet been implemented at the time of the test. Despite the low quality of the images, the resolution should be good enough to return some interesting photos of the Earth's curved horizon, set against the black of space, during the final launch.

The image on the left in Figure 21 is a screenshot of UIView32 during Tether Test 2. The small icon of a house with the label ZU1LEG-4 represents the position of the SUNSET payload. It correctly displays the position of the SUNSET payload on the field behind the engineering building. UIView32 can be configured to plot the track of GPS positions on the map, but since the payload was not moving, this feature was disabled. GPSVisualizer is a website which can display GPS coordinates on a Google Maps map. After Tether Test 2, the text file containing recorded GPS coordinates on the SUNSET payload's micro SD card was uploaded to the GPSVisualizer website. The resulting map is on the right in Figure 21. The map shows that the GPS is not always very accurate. Some of the recorded coordinates are wrong by several hundred meters. However, the vast majority of the recorded points show the payload to be in the correct position. During SUNSET's flight, position reports with several hundred meter accuracy are acceptable. However, once the payload lands, accuracy to within ten meters is desired. There will be time to wait for several position reports to be transmitted once the payload has landed to ensure that stray position reports can be ignored.

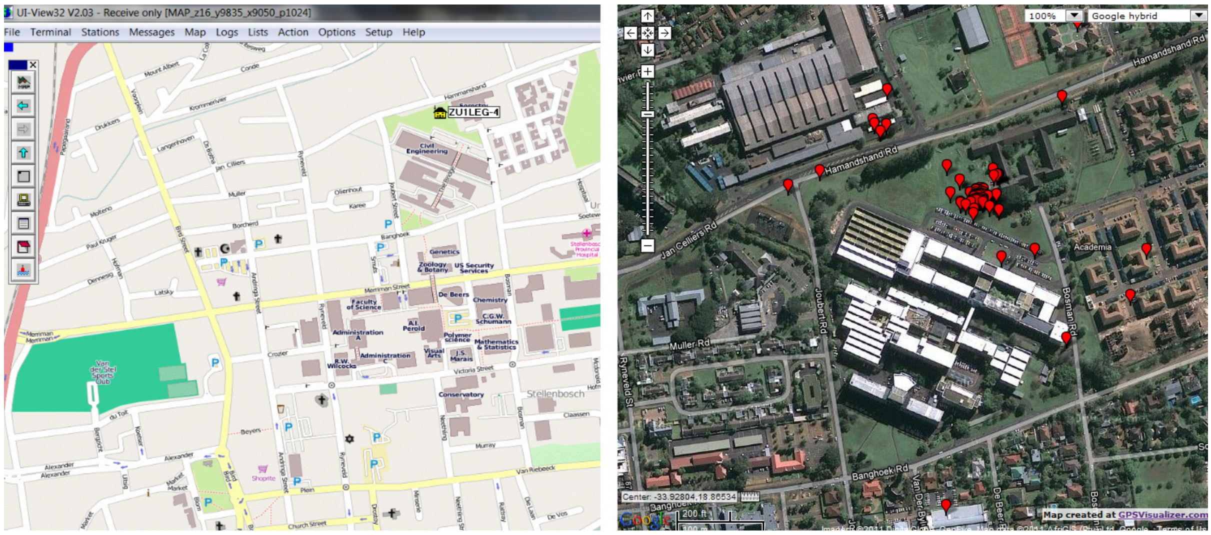


Figure 21 GPS location plots of the SUNSET payload during Tether Test Two. Right: UIView32 Left: GPSVisualizer

Tether Test 2 was a success. It proved the full operation of the SUNSET payload, with the exception of the external temperature sensor. No anomalies occurred during the forty five minute test.

5.3 Final Pre-Flight Test

The final pre-flight test included the external temperature sensor and logging on the ground station side. The custom ground station software was modified to write the data from every received packet to a file stored on the laptop's hard drive. This is an essential feature that ensures that the data is safe even if the custom ground station software crashes. The test was conducted on Monday, 17 October 2011, at 10am. Instead of wasting more Hydrogen by refilling the balloon to carry the payload aloft for this test, the payload was simply hung from a tree branch. The payload hung approximately 1.5m above the ground and had a clear view of the sky. The intervals between transmissions were set to those that will be used during the launch: thirty seconds between position reports, and ten minutes between SMS's and image transmissions. The test lasted forty minutes, during which the GPS maintained a lock. Telemetry and image packets were transmitted and received for the full duration of the test. Four images and four SMS's were received, and both temperature sensors reported their measurements.

The graph in Figure 22 shows that the internal temperature of the SUNSET probe settled approximately twelve degrees higher than the external temperature. The external temperature rose because the SUNSET payload hung in full sunshine. The internal temperature is seen to rise sharply by a few degrees periodically. This occurs during image transmission, as the Radiometrix HX1 gets hot during the long transmission. The insulation will have to be improved for the final flight, as twelve degrees of insulation is not enough.

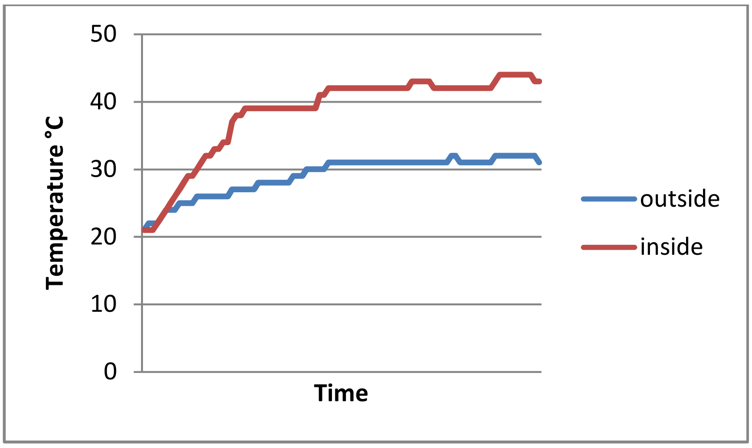


Figure 22 Graph of SUNSET internal and external temperature readings during the Final Pre-Flight Test

The Final Pre-Flight test was a success. No anomalies occurred. The SUNSET electronics are judged to be flight-ready.

5.4 Parachute Test

The parachute was constructed from a garbage bag, as designed. To test the parachute, a section of Pool Noodle was cut in half and hollows were cut out, just as for the insulated electronics container. However, instead of containing electronics, three eggs were put into hollows in the Pool Noodle. The section of Pool Noodle containing the eggs, shown in Figure 23, was put into a fabric sleeve and attached to the parachute. The eggs had the same amount of protection as the electronics onboard SUNSET will have.

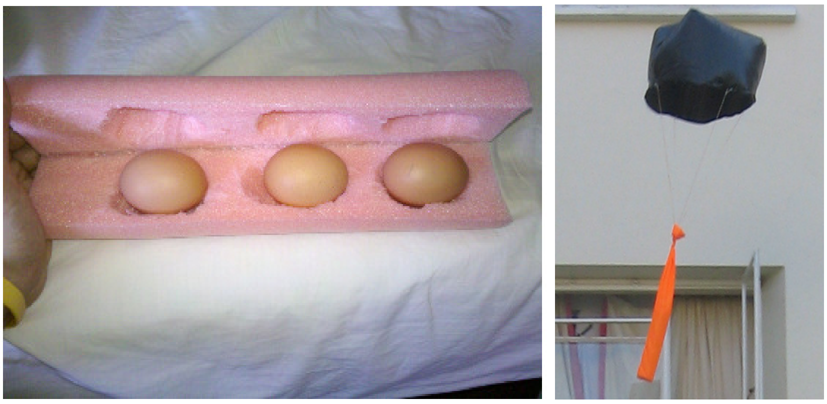


Figure 23 The eggs in a section of Pool Noodle and the parachute in action during the Parachute Test

The section of Pool Noodle containing the eggs, attached to a parachute, was thrown out of a third floor window. For the first drop, the parachute was held with the payload hanging below it, and then dropped. The parachute opened quickly and the eggs floated down to a safe landing on grass. For the second drop, the payload was held with the parachute hanging limp beside it. The payload was forcefully thrown horizontally out of the window. This time the parachute took a bit longer to open. However, it did open, and the payload floated down to a landing on a concrete surface. All three eggs survived again.

Together, the three eggs and the Pool Noodle weighed 210g. This is eighty percent of the weight of the SUNSET payload. If the parachute does not get tangled during descent, the probability of the SUNSET payload landing unharmed is high. The slight risk that the parachute does get tangled must be accepted.

5.5 Launch

5.5.1 Planning

Weather balloons are released twice a day from the Cape Town International Airport. To get a better understanding of the typical duration and flight path of these balloons, a live launch was attended at 12:30pm on Saturday, 22 October 2011. The weather was partly cloudy with a strong North Westerly wind (approximately 32km/h). The weather office at Cape Town International Airport releases, tracks and receives telemetry from the radiosondes attached to the weather balloons. They have a small hanger in which the balloons are filled, using gas from a Hydrogen generator.

The weather balloon was released after clearance from the aircraft controllers. The balloon ascended rapidly, but could be tracked with the naked eye for approximately ten minutes. The balloon initially headed South East. After approximately twenty minutes the balloon was heading due east. The balloon travelled roughly 150km east and reached an altitude of 23km above sea level before bursting, at which point the ground station stopped tracking the balloon. According to the staff of the weather office, the majority of their radiosondes land near Teewaterskloof. Regardless of the wind conditions at ground level, a jet stream seems to exist at higher altitudes which always carries the balloons eastwards. The radiosonde experienced temperatures colder than minus fifty degrees Celsius.

From this experience it is possible to make a prediction about SUNSET's flight path. SUNSET will be launched at 12:30pm on Monday, 31 October 2011. The SUNSET payload and parachute will be launched together with the weather office's radiosonde using a single weather balloon. This will allow the data returned by the SUNSET payload to be compared to that of the weather office's radiosonde. The primary SUNSET ground station, as described in Section 4.5, will remain at Cape Town International Airport throughout SUNSET's flight. A second, mobile ground station, consisting of a VHF amateur radio connected to a laptop, will wait near Teewaterskloof. The second ground station is responsible for receiving telemetry packets from the SUNSET payload during its fall back to earth. The mobile ground station will move towards the predicted landing site during SUNSET's descent. Stellenbosch University's satellite ground station, located in the Electronic Systems Laboratory (ESL), will also try to receive telemetry from the SUNSET payload during its flight.

The payload release mechanism will not be utilised on this flight. However, a half-filled, sealed syringe will be tied to the launch string. When the payload is recovered, the syringe will be inspected to see if the plunger separated from the syringe body during the flight. If it did, the theory behind the release mechanism will have been verified.

A pre-launch checklist is included in Appendix L.

5.5.2 Expected results

If all goes to plan, the SUNSET flight should last roughly two hours. The primary ground station is expected to receive telemetry for the full duration of the flight, possibly excluding the final moments of the descent. At this point, the SUNSET payload may fall behind mountains and break the line of sight. The ESL ground station should start receiving telemetry once the SUNSET payload has risen high enough (a few hundred meters) to be in the ESL ground station's line of sight. Five images and at least one hundred and twenty telemetry packets per hour are expected to be received during the flight. After the balloon bursts at peak altitude, the payload should fall back to earth at a speed no greater than

4.57m/s. It is expected to land in the Teewaterskloof area. The payload should survive the impact with the ground. During descent, the payload's antenna may be positioned unfavourably. It may become tangled or wrapped around the insulated electronics container or parachute. While not ideal, some packets should still be received by the ground stations. If not, the recovery teams will have to depend on the SMS's sent by the payload after landing. SMS's from the payload are expected once the payload is below 500m above ground level.

If all of the aims below are achieved, the mission will be declared successful:

- Reach an altitude of at least 10km above sea level.
- Receive telemetry from the payload throughout the flight.
- Receive at least one image during the flight.
- Successfully track the payload using its transmitted GPS coordinates throughout the flight.
- Receive at least one SMS from the payload after its landing.
- The payload must survive the landing.
- Recover the payload.
- Recover the data on the micro SD card.

If a majority, but not all of the aims are met, the mission will be declared partially successful.

5.5.2 Launch Day

The SUNSET payload was completed on schedule. Testing was completed and the SUNSET project was declared ready for launch two weeks before the due date of this report. Unfortunately, unfavourable weather and administration problems caused the launch date to be pushed back. The administration problems included finding a suitable launch time and location, and getting authorisation from the airport. While the results of the SUNSET launch are not included in this report, the launch is scheduled for Monday, 31 October 2011. Based on thorough testing, the launch is expected to be successful.

6. Conclusion and Recommendations

The aim of this project was to design and manufacture a reusable, expandable, near-space platform. The SUNSET payload consists of all the components necessary for such a platform.

The choice of the Parallax Propeller as flight computer simplified the design of the hardware and firmware. The hardware was simplified by emulating pieces of hardware in software. The firmware design was simplified due to the large database of pre-written objects available for the Propeller. The multiple core architecture of the Propeller caused unexpected problems. Before the Propeller is used in future launches, its architecture must be better understood. Concurrency can be beneficial, but also dangerous and unreliable when not well understood. The flight computer PCB was successfully designed and manufactured and has free I/O ports for the addition of future peripherals. The SD card interface and USB to Serial interface work satisfactorily.

The Propeller successfully performs the job of a modem in software. The AX.25 and APRS protocols are successfully implemented, allowing the telemetry from the SUNSET payload to be received and decoded by ordinary amateur radio equipment. The communication link achieved a range of 5.1km without line of sight, and is expected to work over more than 100km when line of sight is available. During testing, the communication linked proved to be very reliable. The communication subsystem, consisting of the Radiometrix HX1 and emulated AX.25 modem, is recommended for future near-space platforms. If a two way communication system is required, perhaps to control an onboard experiment, the Radiometrix HX1 could be replaced with the Radiometrix SHX1.

The example experiment included onboard SUNSET includes two temperature sensors and a camera. The temperature sensors were calibrated and have one degree Celsius accuracy. The camera can take images and store them on the flight computer's micro SD card, or transmit them to the ground station. The ability to transmit live images digitally from the SUNSET payload was proven, but the quality of the images reduces their usefulness. For future flights, the image quality should be improved. This can be achieved by sampling the video stream faster, possibly with a hardware A/D converter. Some form of compression should also be applied to the images to decrease their transmission time.

The GSM modem is simple to interface to and is a good backup communication system. Future improvements to the SUNSET payload could include the ability to receive information from the GSM modem, such as signal strength and the name of the nearest cell phone tower. This would be useful for recovery efforts if the GPS has failed. The ability to receive command SMS's through the GSM modem could also be added with little difficulty.

Energizer Lithium batteries are expensive, but very lightweight and rated to minus forty degrees Celsius. Unfortunately, their capacity and temperature tolerance were not officially tested. However, the same set of batteries was used for the duration of both tether tests and the final pre-launch test. This proves that they can power the SUNSET payload for at least three hours, which is longer than the expected duration of a near-space flight. Therefore, they are recommended for future near-space flights.

The thermal properties of the insulated electronics container could not be thoroughly tested. However, the Pool Noodle design successfully contains all the electronics and provides at least ten degrees Celsius of insulation. The overall weight of the payload (265g) is very low compared to the other near-space launches described in Section 2.3. A 420g weather balloon filled with Hydrogen can easily lift the payload. Suggested improvements to the payload container include a better method to connect the material sleeve to the flight string and a zipper on the side of the sleeve to allow easy access to the electronics. If a full test flight verifies the thermal insulation of the Pool Noodle container, the design is recommended for future near-space flights.

The custom ground station software successfully interfaces with MixW and displays the information contained in the SUNSET telemetry and image data packets. A good addition to the custom ground station software would be a Google Maps plugin which plots the balloon's location live on a map. This would negate the need for the third party software UIView32.

The design, manufacture and testing of the payload were completed on schedule. The total cost of the SUNSET payload was R1193, which is approximately 35% over budget. The cost of the small, individual electronic components was underestimated. The cost of the SUNSET payload is still far less than most of the near-space payloads described in Section 2.3. A cost breakdown is available in Appendix M. Although SUNSET could not be launched before the due date of this report, a launch date is set. The tests performed on the SUNSET payload predict that the flight will be successful.

References

[1] National Oceanic and Atmospheric Administration. (2010, May) NOAA National Weather Service Radiosonde Observations. [Online]. http://www.ua.nws.noaa.gov/factsheet.htm

[2] HobbySpace. (2011, October) Near-space: The Shore Of Our New Ocean. [Online]. http://www.hobbyspace.com/NearSpace/index.html

[3] Paul Verhage. (2005, January) Near-space. [Online]. http://www.parallax.com/Portals/0/Downloads/docs/cusapps/NearSpace0.pdf

[4] Encyclopædia Britannica. (2011, October) Léon Teisserenc de Bort. [Online]. http://www.britannica.com/EBchecked/topic/585710/Leon-Teisserenc-de-Bort

[5] Ilmari Project. (2002) Amateur Radio on Balloons - Ilmari History. [Online]. http://www.viestikallio.fi/ilmari/history/ilmari-historia.php?LANG=en

[6] Kieth Kaiser. (2011, June) Amateur Radio High Altitude Ballooning (arhab). [Online]. http://www.arhab.org/index.html#

[7] The HABEX Team. (2010) HABEX: High Altitude Balloon Experiments. [Online]. http://www.habex.za.net/about.html

[8] Barry Sloan. (2011, May) BEAR (Balloon Experiments with Amateur Radio). [Online]. http://bear.sbszoo.com/

[9] Alexei Karpenko. (2007, November) HALO: High Altitude Object. [Online]. http://www.natrium42.com/halo/flight2/#hardware

[10] Oliver Yeh, Justin Lee, and Eric Newton. (2010, May) 1337arts Photographs from near-space. [Online]. http://space.1337arts.com/

[11] Leszek Urbanski. (2006) Cygnus High Altitude Balloon. [Online]. http://moo.pl/~tygrys/balloon/

[12] Neha Kumar. (2011, May) Radiation Pattern of a Cell Tower Antenna. [Online]. http://neha-wilcom.blogspot.com/2011/05/radiation-pattern-of-cell-tower-antenna.html

[13] Charles Welch. (2011) The Ozone Hole. [Online]. http://www.theozonehole.com/ozonelayer.htm

[14] Mark Devlin. (2010) BLAST: Balloon-borne Large-Aperture Submillimeter Telescope. [Online]. http://blastexperiment.info/index.php

[15] Renesas. (2006, January) R8C/10 Group SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER. [Online]. http://www.glyn.de/data/glyn/media/doc/Datasheet_R8C_10.pdf

[16] Microchip. (2010) PIC24FJ64GA004 Family Datasheet. [Online].

http://ww1.microchip.com/downloads/en/DeviceDoc/39881D.pdf

[17] Parallax Inc. (2009, April) Propeller Datasheet v1.2. [Online]. http://www.parallax.com/Portals/0/Downloads/docs/prod/prop/PropellerDatasheet-v1.2.pdf

[18] William A. Beech, Douglas E. Nielsen, and Jack Taylor. (1998, July) AX.25 Link Access Protocol for Amateur Packet Radio. [Online]. http://www.tapr.org/pdf/AX25.2.2.pdf

[19] SYMEK GmbH. Packet-Radio-Modulation. [Online]. http://www.symek.com/g/pacmod.html

[20] The APRS Working Group. (2000, August) APRS Protocol Reference. Protocol Version 1.0. [Online]. www.aprs.org/doc/APRS101.PDF

[21] Parallax Inc. (2010, October) Propeller 2 Preliminary Feature List. [Online]. http://www.parallax.com/Propeller2FeatureList/tabid/898/Default.aspx

[22] Philip Pilgrim. (2009, March) Parallax Inc OBEX. [Online]. http://obex.parallax.com/objects/421/

[23] John Hansen, "PIC-et Radio: How to Send AX.25 UI Frames Using Inexpensive PIC Microprocessors," State University of New York, New York, Thesis 1998.

[24] Bob Bruninga. (2010, June) APRS SSID Standard Applications. [Online]. http://www.aprs.org/aprs11/SSIDs.txt

[25] Dave Hein. (2011, June) Parallax Forums. [Online]. http://forums.parallax.com/showthread.php?125684-Bell202-object-AX.25-and-APRS/page2

[26] Radiometrix. (2007, August) VHF Narrow Band FM 300mW Transmitter. [Online]. http://www.radiometrix.com/files/additional/hx1.pdf

[27] Glynn Rogers, *J-Poles Handbook*, 4th ed. Envington, Virginia: BUXCOMM Corporation, 1995. [Online]. http://www.packetradio.com/jpoles4ever.htm

[28] Deal Extreme. USB Tri-band GPRS Modem / Cell Phone Radio (GSM 900/1800/1900Mhz). [Online]. http://www.dealextreme.com/p/usb-tri-band-gprs-modem-cell-phone-radio-gsm-900-1800-1900mhz-12057

[29] Michael Finch. (2010, November) (Probably) The Cheapest way to add SMS to an Arduino. [Online]. http://finch.am/projects/arduinogsm/

[30] Parallax Inc. (2006, May) How to safely interface a 5v signal to the propeller? [Online]. http://forums.parallax.com/showthread.php?85474-How-to-safely-interface-a-5v-signal-to-the-propeller-See-Post-Reply-104

[31] Rayslogic. (2009) Secure Digital (SD) Card Interface. [Online].

http://www.rayslogic.com/propeller/programming/sd_card/sd_card.htm

[32] Perry James Mole. (2011, June) Parallax Forums, stupid video capture. [Online]. http://forums.parallax.com/showthread.php?98516-stupid-video-capture

[33] Parallax Semiconductor. (2011) AN008: Sigma-delta Analog to Digital Conversion. [Online]. http://www.parallaxsemiconductor.com/sites/default/files/appnotes/AN008-SigmaDeltaADC-v1.0_0.pdf

[34] Philips. (2005, February) BC807; BC807W; BC327 PNP general-purpose transistors data sheet. [Online]. http://www.mantech.co.za/datasheets/products/BC807-NXP.pdf

[35] Energizer. ENERGIZER L91 Product Datasheet. [Online]. data.energizer.com/PDFs/l91.pdf

[36] Transcend. Transcend microSDHC Card series. [Online]. www.transcendusa.com/support/./datasheet/TS16GUSDHC6.pdf

[37] Randy Culp. (2010, March) Parachute Descent Calculations. [Online]. http://my.execpc.com/~culp/rockets/descent.html

[38] Jordan Hiller. Model Rocket Parachute Descent Rate Calculator. [Online]. http://www.onlinetesting.net/cgi-bin/descent3.3.cgi

[39] Nova Lynx. Weather Balloons & Accessories. [Online]. http://www.novalynx.com/400-balloons.html

[40] Parallax Inc. (2011, July) Why do we need the transistor reset circuit for the USB devices? (FT232RL, CP2102). [Online]. http://forums.parallax.com/showthread.php?133362-Why-do-we-need-the-transistor-reset-circuit-for-the-USB-devices-(FT232RL-CP2102)

[41] Alexander Erlank. (2007) HAM RADIO. [Online]. http://students.ee.sun.ac.za/~leggs/main/ham_radio.html

[42] Barry Sloan. (2007, August) Micro-Trak 300 - V1.3 APRS Tracker. [Online]. http://www.sbszoo.com/bear/construction/traker/utrak1.3/utrak-V1-3.htm

Appendix A – Project Planning Schedule

Appendix B – Project Specification

UNIVERSITY OF STELLENBOSCH
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

AGREEMENT BETWEEN STUDENT AND STUDY LEADER RE MUTUAL RESPONSIBILITIES

Course: Project 448

Student: A.O. Erlank

Study leader: A. Barnard

Project: SUNSET – Stb. Univ. Near-Space Engineering Testbed.

1. It is the responsibility of the student to clarify aspects such as the definition and scope of the project, the place of study, research methodology, reporting opportunities and -methods (i.e. progress reports, internal presentations and conferences) with the study leader.

   Aims:

   – Design + build PCB with MCU + GPS + GSM + SD Card + Radio telemetry link.

   – Software (MCU) to demonstrate system functionality + interface to PC environment.

   – Packaging electronics in a High-Altitude-Ready configuration (balloon-type system)

   (Use other side of paper if necessary)

2. If the above aims are not achieved, the project may be considered incomplete.

3. It is the responsibility of the study leader to give regular guidance and feedback with regard to the literature, methodology and progress.

4. The rules regarding the handing in and evaluation of the project is outlined in the Study Guide and will be strictly adhered to.

5. The project leader conveyed the departmental view on plagiarism to the student, and the student acknowledges the seriousness of such an offence.

Study leader: [signature]

Student: [signature]

Date: 2 August 2011

Appendix C – Outcomes Compliance

ECSA EXIT LEVEL OUTCOMES		
Outcome	**Description**	**Section of Report**
1. Problem solving	Many problems, such as how to interface a camera to a microcontroller, were overcome during the design stage. New problems, such as concurrency issues, emerged and were solved during the implementation stage.	3 and 4
2. Application of scientific and engineering knowledge	Circuits were mathematically modelled and implemented. Algorithms and protocols were implemented in firmware.	3 and 4
3. Engineering Design	A thorough design process was followed	3
4. Investigations, experiments and data analysis	The SUNSET payload went through several tests. The resulting data was used to help troubleshoot problems. The temperature sensor was calibrated.	5
5. Engineering methods, skills and tools, including Information Technology	The internet and other information resources were successfully consulted before and during the design phase. Troubleshooting techniques were employed throughout. Physical tools such as soldering irons and screw drivers were used during the implementation phase.	Whole report
6. Professional and technical communication		Whole report
9. Independent learning ability	A lot was learnt during this project, especially during the literary review phase. The AX.25 and APRS protocols were studied and implemented. Information from a wide variety of sources was gathered, reviewed and utilised.	2,4

Appendix D – Radio Frequency Link Budget Calculations

The following calculations are based on the equations in Frank's (PA3GMP) document, titled: *Calculating a Link Budget*, available online at:
http://silverwolfenterprises.co.za/hvbextra/Linkbudget.pdf

The link budget is expressed by the equation below:

$$Prx = Ptx + Gtx - Lfs + Grx$$

where:

Prx = received power (dBm)
Ptx = transmitter output power (dBm)
Gtx = transmitter antenna gain (dBi)
Lfs = free space loss or path loss (dB)
Grx = receiver antenna gain (dBi)

AGWPE's technical documentation states that a signal strength of at least S3 (-109dBm) is required to decode a 1200 Baud AX.25 Packet. Therefore Prx = -109dBm minimum.

The Radiometrix HX1 transmitter has an output power of 300mW = 10log(300/1) = 24dBm.

The Radiometrix HX1 is connected to a flexible J-pole antenna with an approximate gain of 3.7dBm. The loss in the short piece of connecting coaxial cable is ignored.

The free space loss equation is given in *Calculating a Link Budget* as:

$$\text{Free Space Path Loss (dB)} = 32.44 + 20*\log(F(MHz)) + 20*\log(D(km))$$

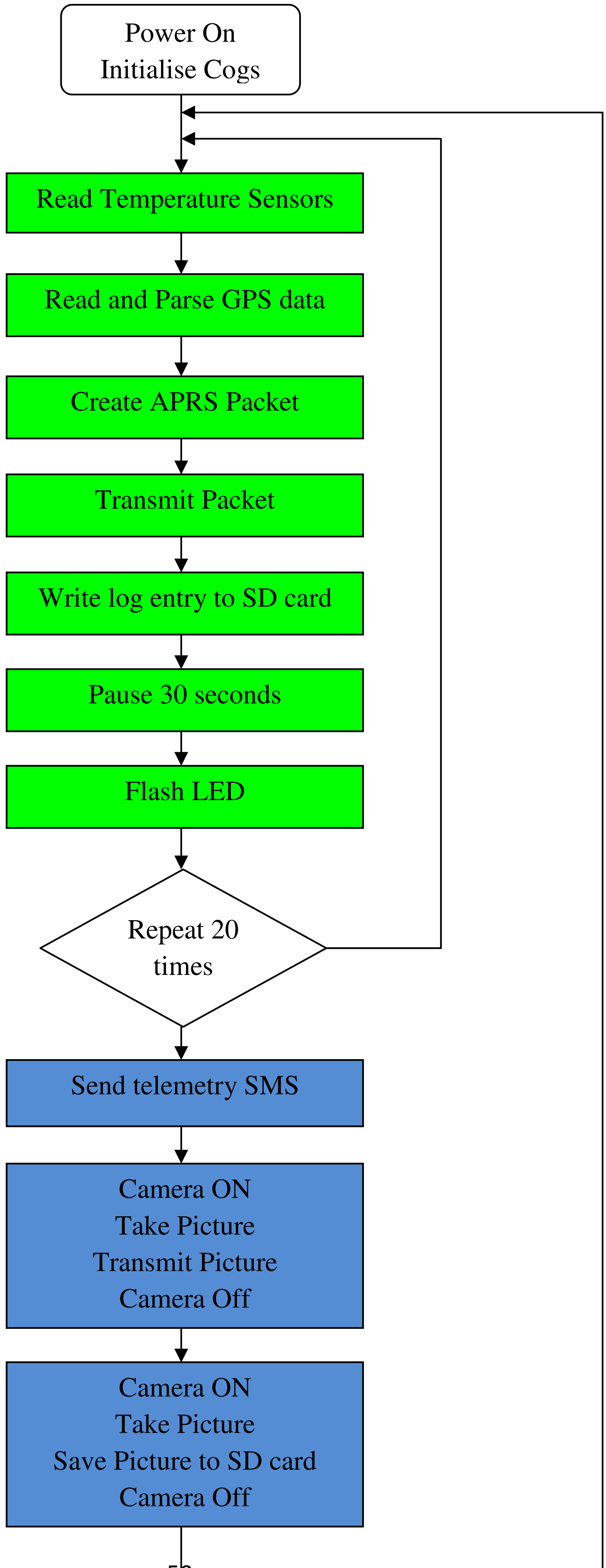
The free space loss is related to the distance between the transmitter and receiver (D) and the radio transmitter frequency (F). Substituting in various distance values gives the results in the table below:

Distance (km)	Transmitting Frequency (MHz)	Free Space Path Loss (dB)
10	144.800	95
50	144.800	109
100	144.800	115

The ground station receiver also has a flexible j-pole antenna with approximately 3.7dBm of gain. The small loss in the coaxial cable is ignored.

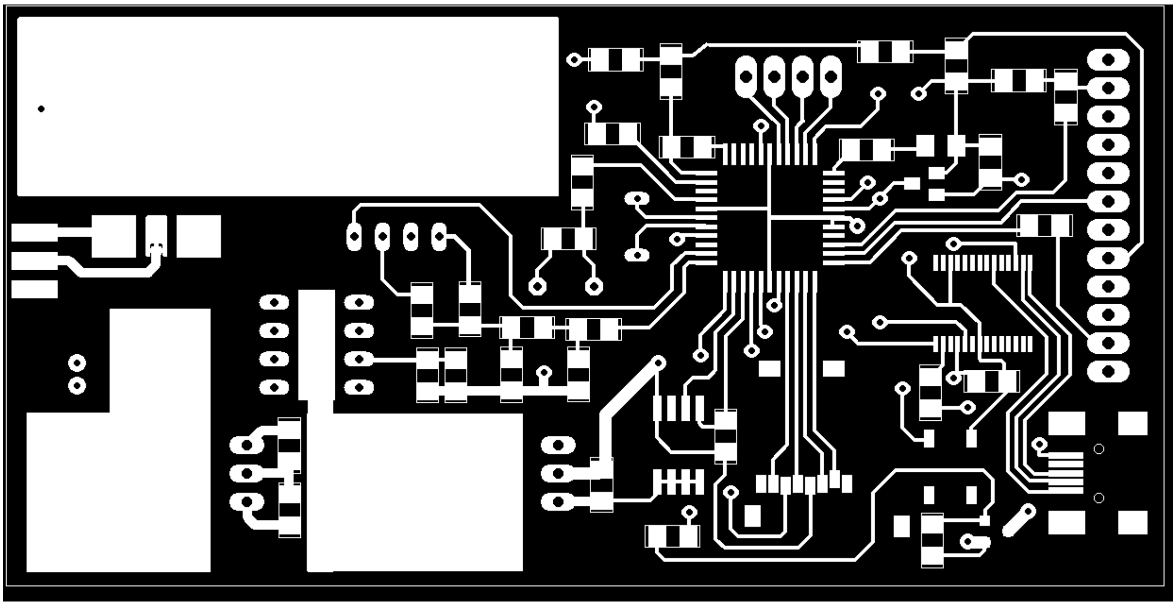
Rearranging equation 1 to make the free space loss (Lfs) the subject of the equation and substituting in values gives a maximum allowable Lfs of 140.4dBm. Even when the radio transmitter and receiver are 100km apart, the Lfs is still far below 140.4dBm. Therefore the communication link is expected to work over a range of over 100km.

Appendix E - Firmware Flowchart

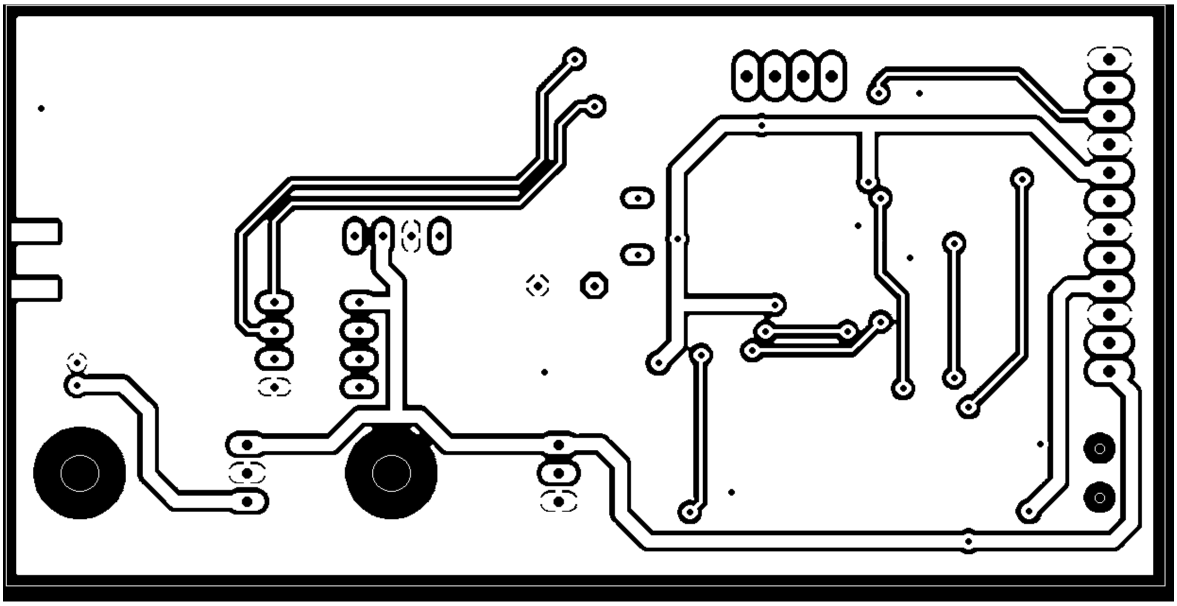


Appendix F – PCB Masks

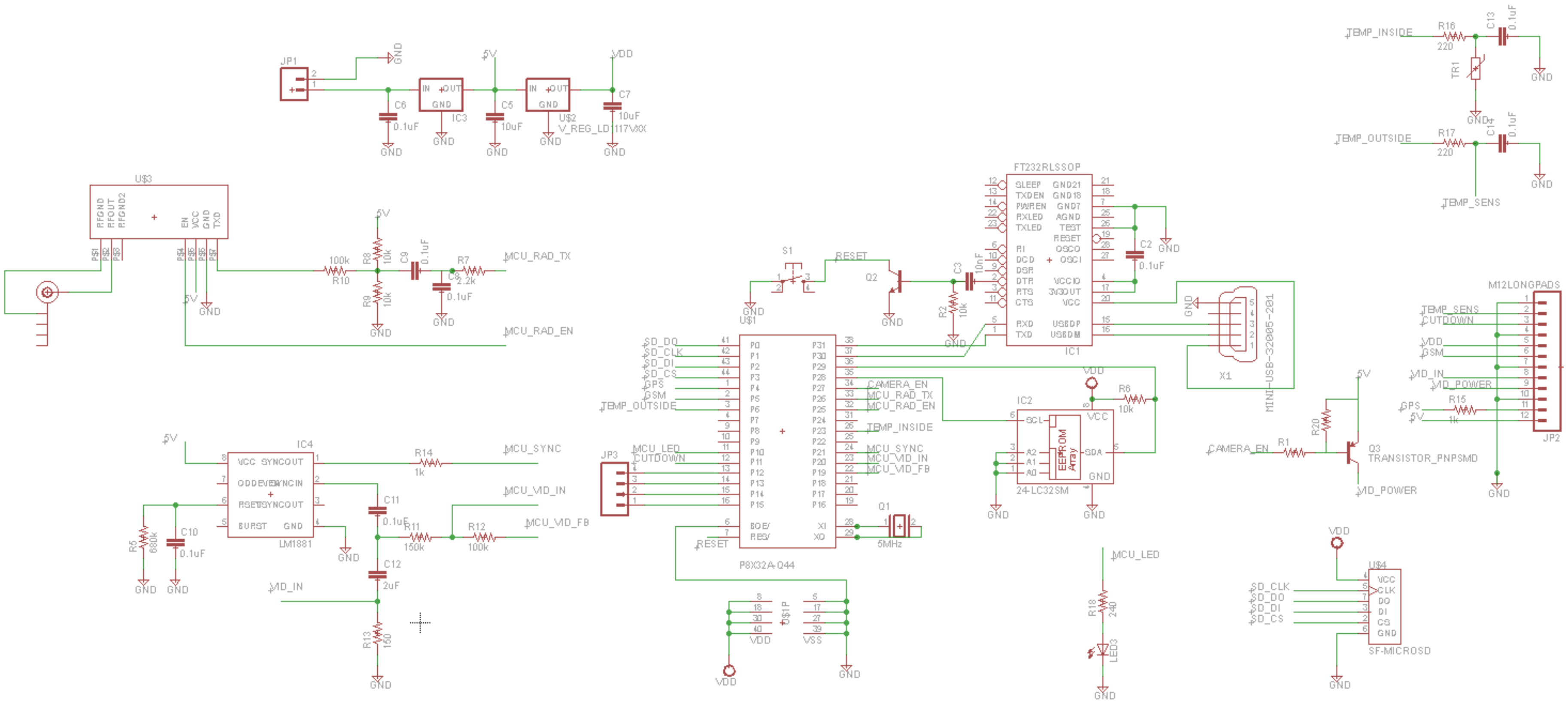
Top



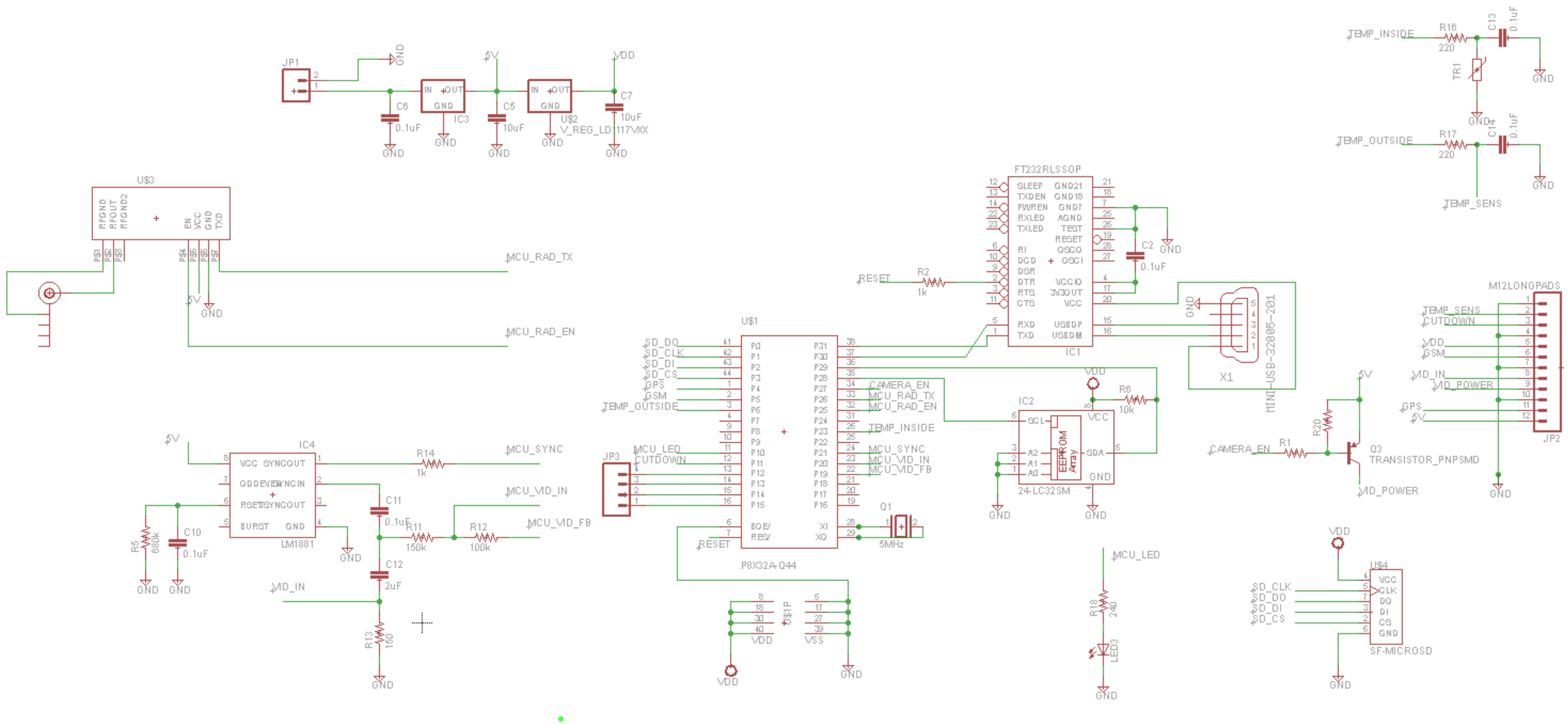
Bottom



Appendix G – Complete Circuit Diagram as Designed



Appendix H – Complete Circuit Diagram as Implemented



Appendix I – Table of Firmware Objects and Methods

Object name	Method	Description
SUNSET_main (cog 0)	initialise	First to run when the power is switched on. Initialises other objects.
	mainLoop	This loop repeats continuously, reading sensors and transmitting telemetry.
	pause	Pauses for given amount of seconds
	fillGPSPacket	Reads and parses GPS data. Creates a string containing GPS coordinates and temperature readings to be transmitted.
	addLogEntry	Adds a new entry to the temperature and GPS logs on the micro SD card.
	integerToDecimal	Returns a string representing a given integer.
	led	Turns the led on or off.
	addToPacket	Adds a given string to the array that will be transmitted (packet array).
	clearPacket	Clears the packet array.
	takeAndSendPicture	Turns the camera on, takes an image, and turns the camera off again. Then it transmits the image
	takeAndSavePicture	Same as above except saves the image to the micro SD card.
	sendImage	Creates and sends 40 packets each containing six lines of image data.
	digit	Overlays given digits onto the image.
	tempConversion	Converts the integer result of the RCTIME A/D conversion to an integer representing temperature in Kelvin.
	ln	Performs the natural logarithm.
ax25 (a modification of Bell202 object)	start_simple	Starts a new cog for the modem (below). Sets up output pin.
	modem (assembly language)	Runs on its own cog in the background (cog 1). Generates the tones necessary for Bell202 modulation. Modified with bit-stuffing and NRZI encoding.
	out	Adds a given byte to the modem transmit buffer.
	createPacket	Creates a correctly formatted APRS packet. Given string is put in the Information field.
	sendPacket	Encodes the packet created with createPacket, one byte at a time.
	computeCRC	Computes the checksum of the APRS packet.
GPS_IO_mini	start	Starts a new cog (cog 2) for serial communication with the GPS in the background. Sets up input pin. Modified to not start parsing on a new cog.
	readNEMA	Waits for a new NMEA string to begin, then reads it into a buffer. Parses the data into several

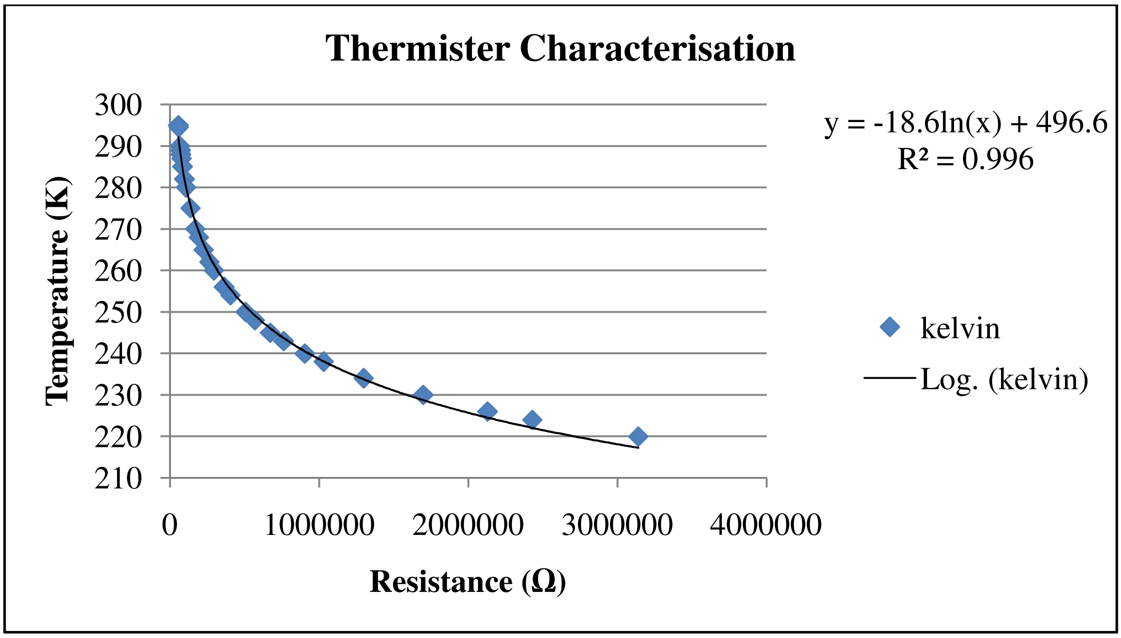
		strings representing latitude, longitude, etc. Modified to not run continuously on its own cog.
	Latitude Longitude GPSAltitude Valid	Various methods to return parts of the parsed GPS data. Modified to use NMEA RMC strings for Latitude and Longitude data instead of GGA strings. RMC strings seem to occur even when the satellite signal is very weak.
GPRS_Modem	init	Sets up the output pin for serial commands.
	sendSMS	Sends an SMS message containing the given string to the given number.
Logging	SaveBMP	Saves an image taken earlier to a BMP file on the SD card.
	LogGPS	Opens the file log.txt on the SD card in append .mode and adds a new entry containing current GPS coordinates.
	LogTemperature	Opens the file temperature.txt on the SD card in append mode and adds a new entry containing current temperature measurements.
video_capture	frames	Specifies the memory location to save image. Starts a new cog (cog 3) to do frame capture.
	entry (assembly language)	Performs very fast sigma-delta A/D conversion and stores the results in the memory location specified above. Synchronises with new line signals from the LM1881.
RCTIME	RCTIME	Performs foreground A/D conversion on specified pin.
fsrw safe_spi		Provides FAT16/32 SD card read/write methods.
FullDuplexSerial (cog 2)	rx	FullDuplexSerial runs on its own cog. Emulates a hardware UART. Used by GPS_IO_mini to interface to the GPS. Rx receives a byte from the receive buffer.
Simple_Serial	tx	Performs serial communication in a blocking fashion. Does not run on its own cog. Used to communicate with GPRS modem. Tx transmits a byte.
umath		Unsigned maths routines required by ax25.
Config		Pin assignments and other global configuration values.

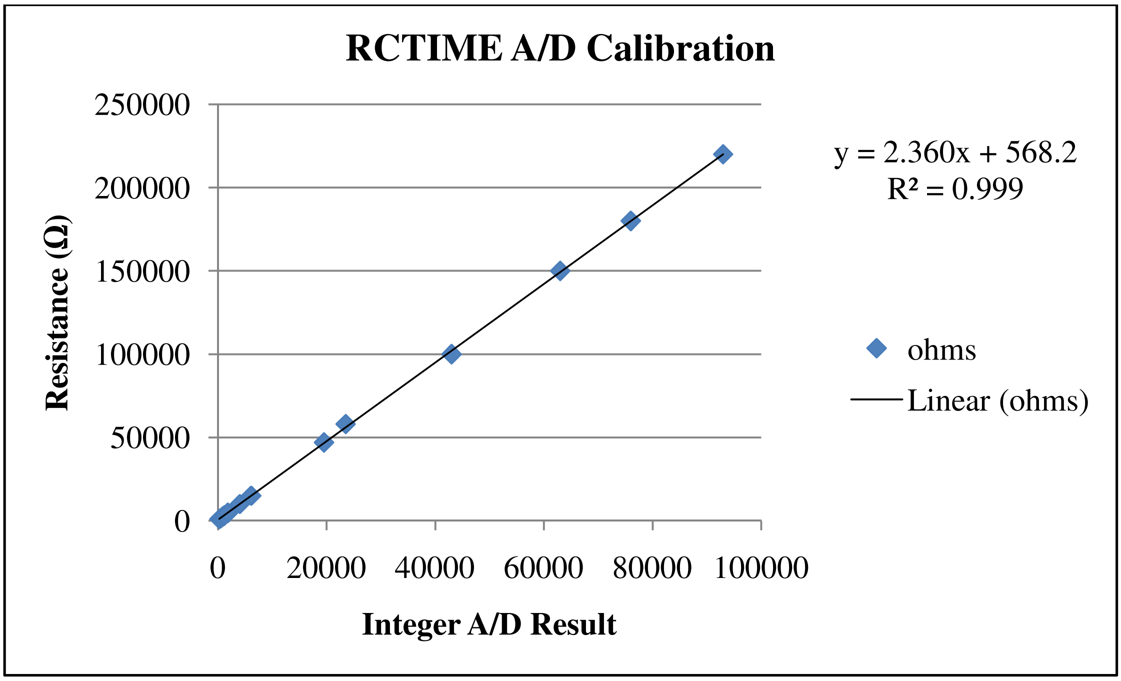
Appendix J – Thermister Calibration Data

Thermistor characterisation, RCTIME calibration

ohms	kelvin
57200	295
58400	294.5
66000	290
72000	289
74000	288
78000	287
85000	285
98000	282
108000	280
137000	275
170000	270
194000	268
226000	265
264000	262
294000	260
364000	256
406000	254
507000	250
569000	248
673000	245
763000	243
905000	240
1030000	238
1299000	234
1699000	230
2130000	226
2430000	224
3140000	220
4500000	216
4710000	214
6220000	210
8860000	205
12700000	200
18700000	195
37000000	190
59000000	183
154000000	170

AD result	ohms
240	1000
450	1500
750	2200
960	2700
1100	3000
1200	3300
1800	4700
4000	10000
6100	15000
19500	47000
23500	58000
43000	100000
63000	150000
76000	180000
93000	220000


Thermister Characterisation
300
290
280
270
260
250
240
230
220
210
Temperature (K)
0
1000000
2000000
3000000
4000000
Resistance (Ω)
y = -18.6ln(x) + 496.6
R² = 0.996
kelvin
Log. (kelvin)


RCTIME A/D Calibration
250000
200000
150000
100000
50000
0
Resistance (Ω)
0
20000
40000
60000
80000
100000
Integer A/D Result
y = 2.360x + 568.2
R² = 0.999
ohms
Linear (ohms)

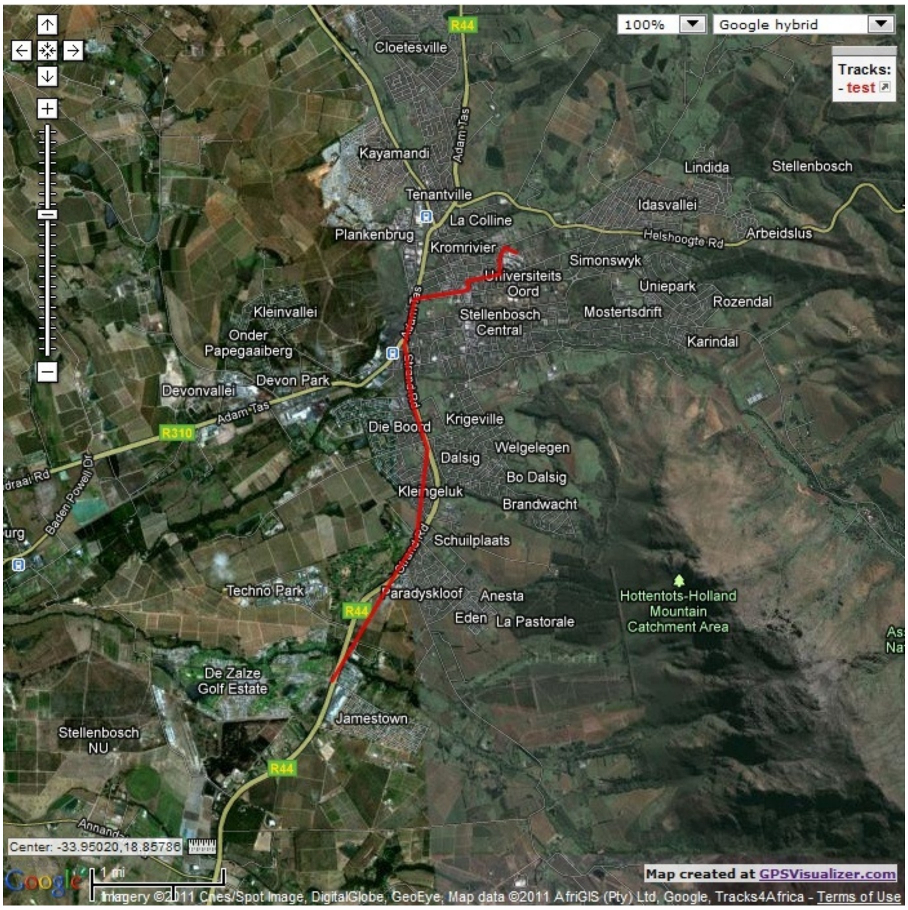
Appendix K – Raw Test Data not presented in the report

1. Communication Range and GPS Test

1.1 Raw packets received by ground station as displayed by UIView32:

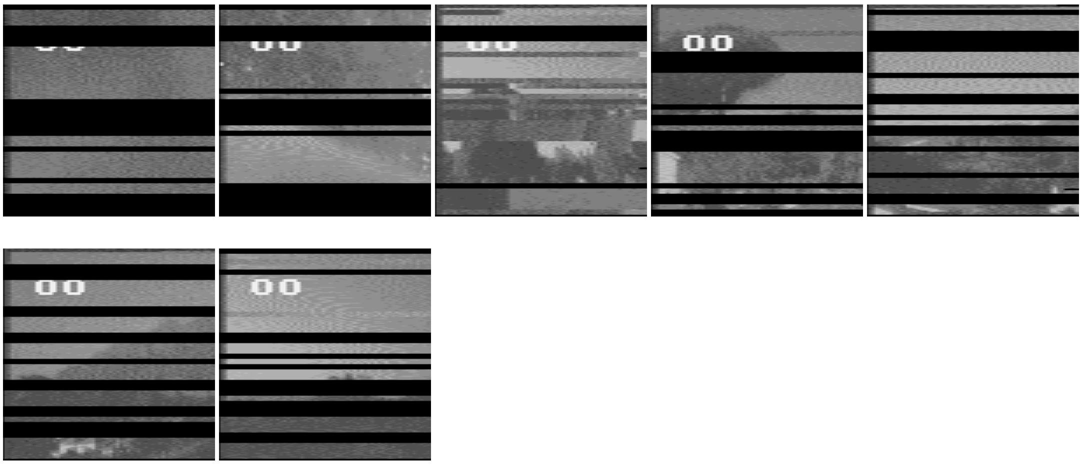
```
ZU1LEG-4>CQ>UI,?,F0:
!3358.50S/01850.50E-

ZU1LEG-4>CQ>UI,?,F0:
!3357.72S/01851.06E-

ZU1LEG-4>CQ>UI,?,F0:
!3357.58S/01851.17E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.96S/01851.27E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.65S/01851.14E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.48S/01851.11E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.47S/01851.11E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.35S/01851.10E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.28S/01851.08E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.15S/01851.12E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.14S/01851.12E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.14S/01851.12E-

ZU1LEG-4>CQ>UI,?,F0:
!3356.06S/01851.14E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.97S/01851.16E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.93S/01851.48E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.92S/01851.55E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.90S/01851.60E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.85S/01851.59E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.83S/01851.70E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.81S/01851.84E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.70S/01851.84E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.63S/01851.88E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.66S/01851.97E-

ZU1LEG-4>CQ>UI,?,F0:
!3355.66S/01851.98E-
```

1.2 Return Journey Raw Data Displayed by GPSVisualiser

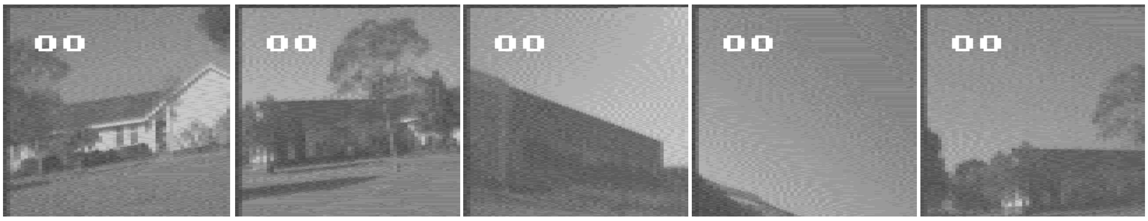


2. Tether Test 1

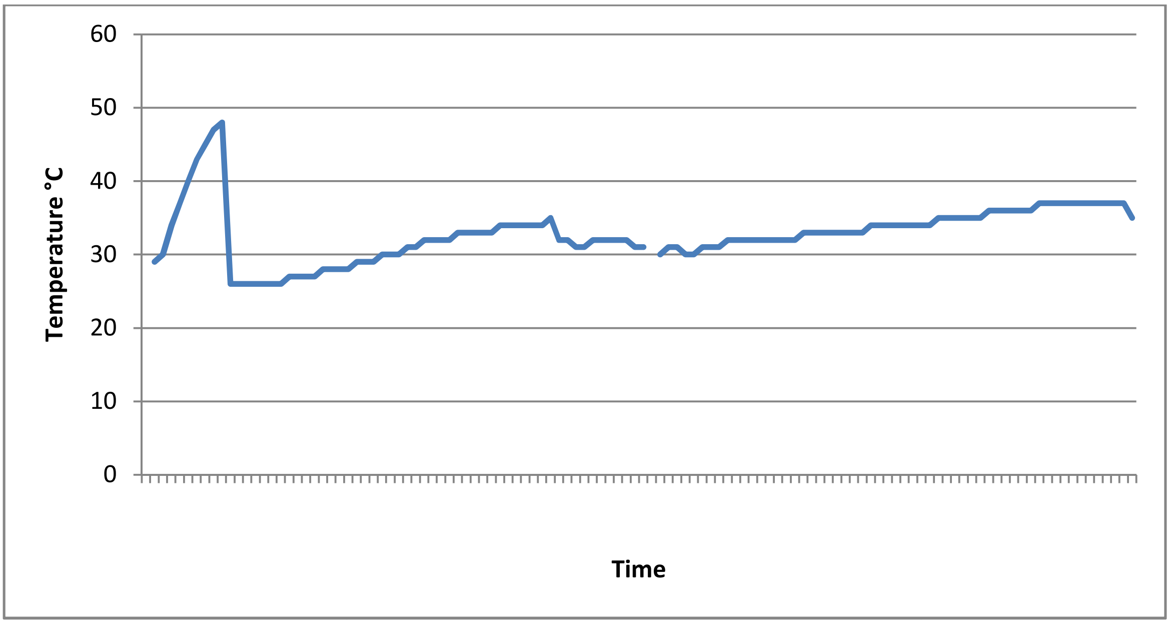
2.1 Raw images received by the custom ground station software



2.2 Selection of the images saved to the onboard micro SD card



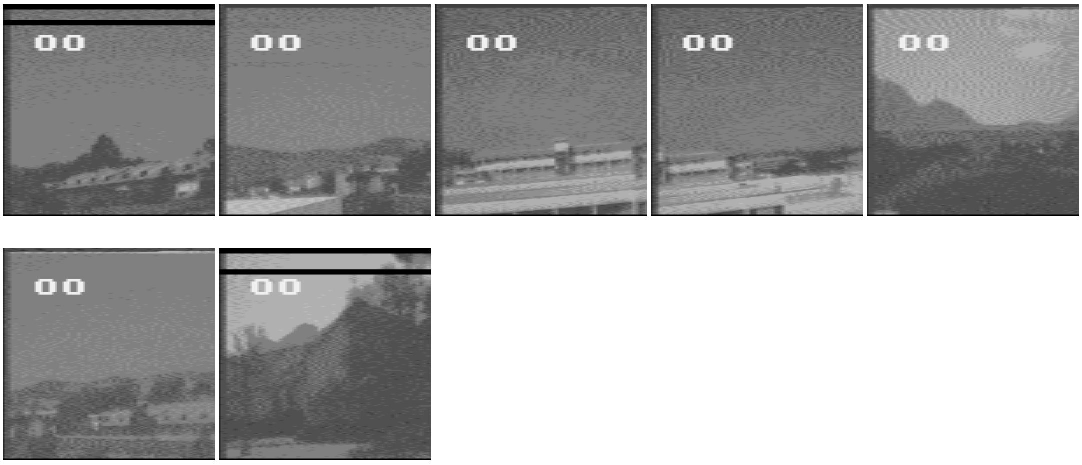
2.3 Plot of Internal Payload Temperature vs Time



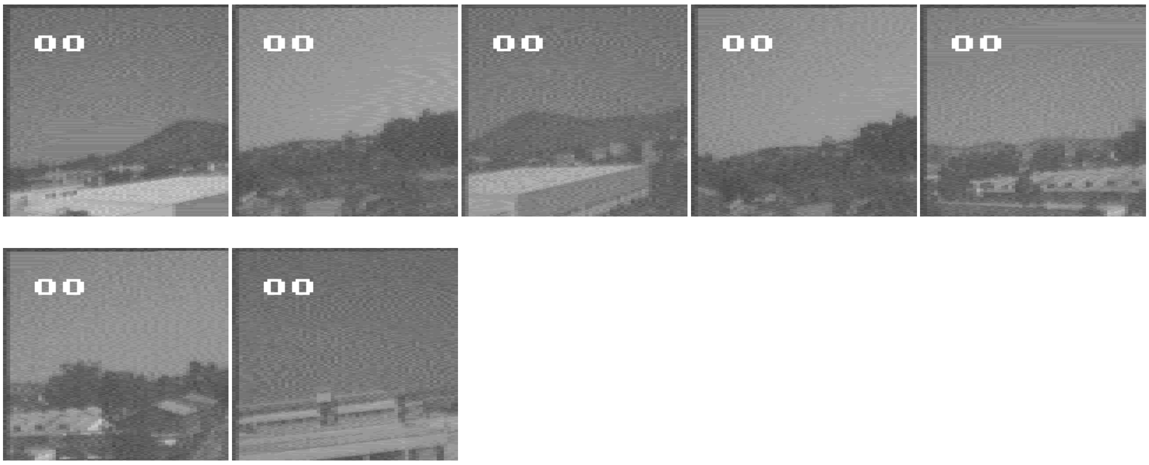
The initial spike in temperature was the result of the flight computer crashing and continuously transmitting the same tone. The balloon was brought down and the insulated electronics container was opened to reset the flight computer.

3. Tether Test 2

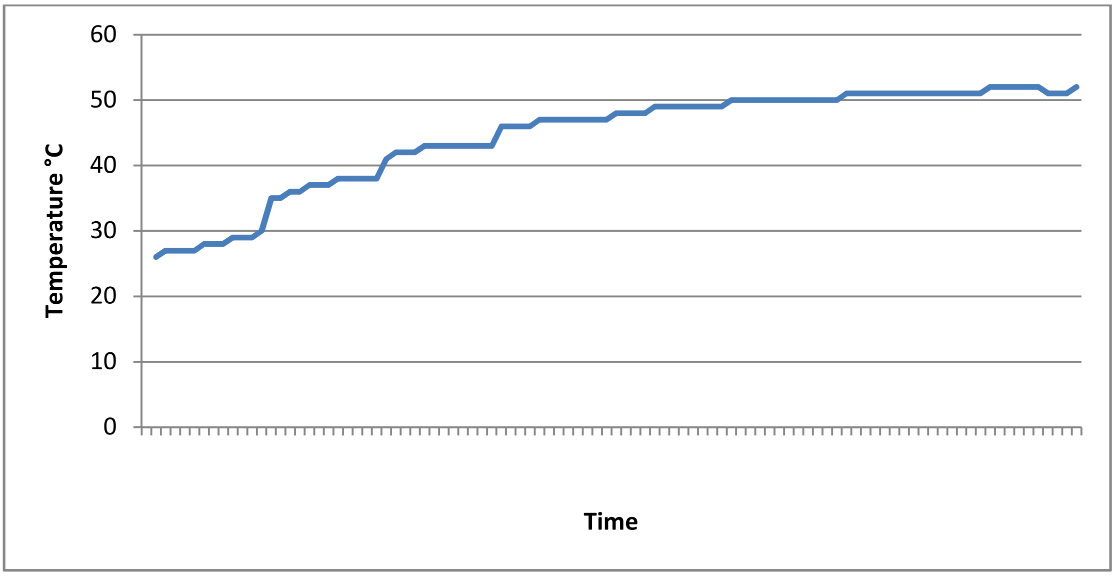
3.1 Raw images received by the custom ground station software



3.2 The images saved to the onboard micro SD card

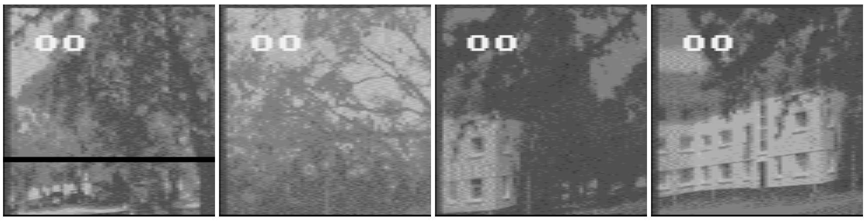


3.3 Plot of Internal Payload Temperature vs Time

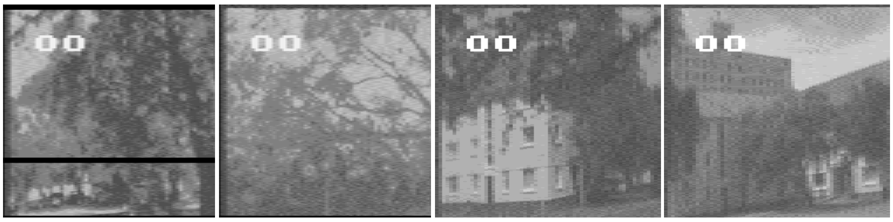


4. Final Pre-Flight Test

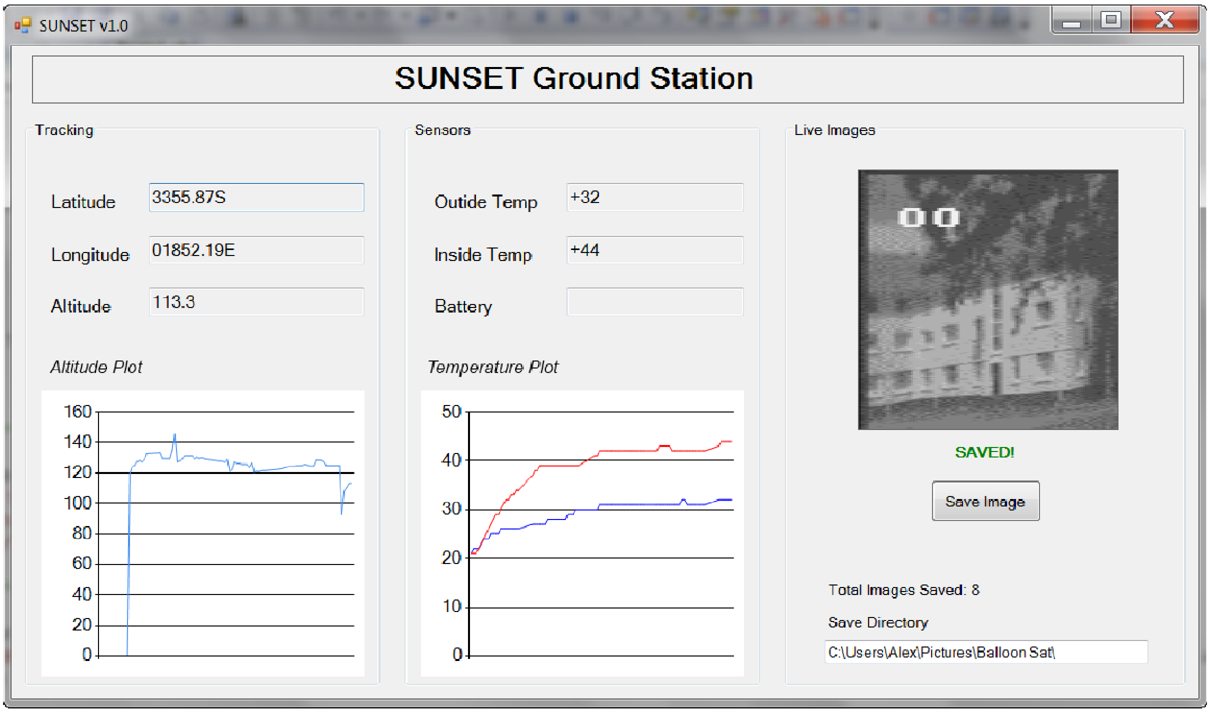
4.1 Raw images received by the custom ground station software



4.2 The images saved to the onboard micro SD card



4.3 Screenshot of the custom ground station software at the end of the test



The screenshot above shows that a GPS position lock was kept throughout the test. Both temperature sensors functioned correctly.

4.4 Content of the custom ground station software log after the test

Each log entry begins with the date, then the time, followed by the GPS coordinates and the two temperature measurements.

```
17102011_101102,no gps  ,no gps   ,no gps,+21,+21
17102011_101135,no gps  ,no gps   ,no gps,+22,+21
17102011_101209,no gps  ,no gps   ,no gps,+22,+21
17102011_101243,no gps  ,no gps   ,no gps,+22,+22
17102011_101317,no gps  ,no gps   ,no gps,+23,+23
17102011_101351,no gps  ,no gps   ,no gps,+24,+24
17102011_101425,no gps  ,no gps   ,no gps,+24,+25
17102011_101458,no gps  ,no gps   ,no gps,+24,+26
17102011_101532,no gps  ,no gps   ,no gps,+25,+27
17102011_101606,no gps  ,no gps   ,no gps,+25,+28
17102011_101640,no gps  ,no gps   ,no gps,+25,+29
17102011_101713,no gps  ,no gps   ,no gps,+25,+29
17102011_101748,3355.86S,01852.19E,119.8,+26,+30
17102011_101823,3355.86S,01852.19E,124.2,+26,+31
17102011_101858,3355.86S,01852.19E,124.8,+26,+32
17102011_101933,3355.86S,01852.19E,127.4,+26,+32
17102011_102008,3355.86S,01852.19E,128.3,+26,+33
17102011_102043,3355.86S,01852.19E,127.3,+26,+33
17102011_102118,3355.85S,01852.19E,128.9,+26,+34
17102011_102153,3355.86S,01852.19E,132.7,+26,+34
17102011_102501,3355.87S,01852.19E,133.2,+27,+37
17102011_102536,3355.87S,01852.19E,129.4,+27,+38
17102011_102611,3355.87S,01852.19E,129.4,+27,+38
17102011_102646,3355.87S,01852.18E,129.4,+27,+39
17102011_102721,3355.86S,01852.19E,129.4,+27,+39
17102011_102756,3355.86S,01852.19E,134.7,+27,+39
17102011_102831,3355.86S,01852.19E,145.4,+28,+39
17102011_102905,3355.87S,01852.19E,127.3,+28,+39
17102011_102940,3355.87S,01852.19E,128.0,+28,+39
17102011_103015,3355.87S,01852.19E,129.1,+28,+39
17102011_103050,3355.87S,01852.19E,131.3,+28,+39
17102011_103125,3355.87S,01852.19E,131.3,+28,+39
17102011_103200,3355.87S,01852.19E,131.3,+28,+39
17102011_103235,3355.87S,01852.19E,131.3,+28,+39
17102011_103310,3355.87S,01852.19E,129.2,+29,+39
17102011_103344,3355.87S,01852.19E,130.3,+29,+39
17102011_103419,3355.87S,01852.19E,129.5,+29,+39
17102011_103454,3355.87S,01852.19E,130.0,+30,+39
17102011_103529,3355.86S,01852.19E,129.3,+30,+39
17102011_103913,3355.87S,01852.19E,127.8,+30,+41
17102011_103948,3355.87S,01852.19E,127.8,+30,+41
17102011_104023,3355.87S,01852.19E,127.0,+31,+42
17102011_104057,3355.87S,01852.19E,128.8,+31,+42
17102011_104132,3355.86S,01852.19E,120.9,+31,+42
17102011_104207,3355.86S,01852.19E,122.1,+31,+42
17102011_104242,3355.90S,01852.17E,127.3,+31,+42
17102011_104317,3355.87S,01852.19E,126.3,+31,+42
17102011_104352,3355.87S,01852.19E,126.2,+31,+42
17102011_104427,3355.87S,01852.19E,125.8,+31,+42
17102011_104502,3355.87S,01852.19E,125.8,+31,+42
17102011_104537,3355.87S,01852.19E,125.8,+31,+42
17102011_104612,3355.87S,01852.19E,123.6,+31,+42
17102011_104646,3355.87S,01852.19E,126.6,+31,+42
17102011_104721,3355.86S,01852.19E,120.4,+31,+42
17102011_104756,3355.87S,01852.19E,121.1,+31,+42
17102011_104831,3355.87S,01852.19E,120.9,+31,+42
17102011_104906,3355.87S,01852.19E,121.4,+31,+42
17102011_104941,3355.87S,01852.19E,121.6,+31,+42
17102011_105016,3355.87S,01852.19E,121.6,+31,+42
17102011_105325,3355.85S,01852.17E,122.5,+31,+42
17102011_105400,3355.87S,01852.19E,122.9,+31,+43
17102011_105435,3355.86S,01852.19E,123.5,+31,+43
17102011_105510,3355.86S,01852.18E,123.9,+31,+43
17102011_105545,3355.86S,01852.19E,124.4,+31,+43
17102011_105620,3355.86S,01852.19E,124.4,+31,+43
17102011_105655,3355.86S,01852.19E,124.4,+31,+42
17102011_105729,3355.86S,01852.19E,124.7,+31,+42
17102011_105804,3355.86S,01852.19E,124.7,+31,+42
17102011_105839,3355.86S,01852.19E,124.9,+31,+42
17102011_105914,3355.86S,01852.19E,125.6,+32,+42
17102011_105949,3355.86S,01852.19E,125.5,+32,+42
17102011_110024,3355.94S,01852.27E,124.6,+31,+42
17102011_110059,3355.86S,01852.19E,124.6,+31,+42
17102011_110134,3355.90S,01852.22E,124.6,+31,+42
17102011_110209,3355.90S,01852.22E,128.6,+31,+42
17102011_110318,3355.86S,01852.20E,128.3,+31,+42
17102011_110353,3355.86S,01852.19E,127.7,+31,+42
17102011_110428,3355.86S,01852.18E,124.9,+31,+42
17102011_110737,3355.86S,01852.19E,124.9,+32,+43
17102011_110811,3355.86S,01852.19E,92.8,+32,+44
17102011_110846,3355.86S,01852.19E,108.2,+32,+44
17102011_110921,3355.86S,01852.19E,110.3,+32,+44
17102011_110956,3355.86S,01852.19E,113.0,+32,+44
17102011_111031,3355.87S,01852.19E,113.3,+32,+44
17102011_111106,3355.86S,01852.19E,113.8,+32,+44
17102011_111141,3355.86S,01852.19E,114.1,+32,+44
```

Appendix L – Pre-Launch Checklist

1. Ensure that the onboard micro SD card is in its socket, and empty.
2. Connect radio to laptop. Tune to 144.800MHz
3. Start MixW, AGWPE, UIView32 and the custom ground station software (CGSS)
4. Connect the SUNSET batteries
5. Verify that telemetry is being received and decoded by MixW and CGSS
6. Tape the batteries and seal the Pool Noodle with duct tape
7. Secure with elastics
8. Insert Pool Noodle into fabric sleeve. Ensure hole is in front of camera
9. Secure with elastics
10. Attach line and parachute
11. Tune radio volume for AGWPE. Check UIView32 for valid position
12. Turn on tracking in UIView32
13. Wait for one complete transmission cycle: telemetry, SMS, photo, telemetry
14. Has GPS locked on? If no, wait for GPS lock
15. Attach to balloon
16. Release!

Appendix M – Cost Breakdown

Component	Cost (R)
Pool Noodle	30
GPS Sensor	245
Radiometrix HX1	265
Camera	150
GSM Modem	154
Parallax Propeller	109.46
PCB electronic components	240
Total	1193