

# **Design of a small satellite UHF radio beacon for Identification, Telemetry, Tracking and Control**

*Travis J McKee<sup>1</sup>*

*The University of New South Wales at the Australian Defence Force Academy*

**The reduced cost of designing, manufacturing, launching and operating small satellites has seen a significant increase in the number of objects deployed into the low earth orbit space environment. Small satellites operated by organizations with little space experience has resulted in a large failure rate, increasing the number of space debris and creating more reliance on expensive space monitoring equipment to maintain space situational awareness. The aim of this study is to provide a concept design for a self-sustained satellite radio beacon that can transmit a unique identification and telemetry data, independent of any satellite system failures, to multiple dispersed ground stations which allows for tracking of the satellites position through trilateration. The satellite radio beacon system allows for: 1) the collection of telemetry data for fault analysis which could decrease the current satellite failure rate; and, 2) a radio signal for tracking using cost-effective ground stations irrespective of satellite system failures to reduce the reliance on space monitoring equipment. The study will utilize an agile, iterative design approach in which each component of the small satellite UHF radio beacon system (satellite radio beacon, communications link and ground receiving station) will be designed, tested and verified in sequential order. This study has produced a solderless breadboard prototype design for the satellite radio beacon that is capable of self-sustained operation in a ground environment. The communications link, using a LoRa radio module, is capable of the reliable transmission of data for the distances expected for a small satellite in a low earth orbit. The ground receiving station has proven the capability to receive the identification data, telemetry data and transmit a command for satellite control. Further work is required to reduce the uncertainty in the time measurement techniques by the ground receiving station to create an accurate tracking capability which would allow the continued development of the satellite UHF radio beacon system.**

## **Contents**

I. Introduction	3
II. Aim	3
III. Background	4
IV. The present study	4
A. Satellite radio beacon segment	4
1. Materials and methods	5
2. Results	5
3. Discussion	6
B. Communications link	7
1. Materials and methods	8
2. Results	8
3. Discussion	9
C. Ground receiving segment	10
1. Materials and methods	10
2. Results	11
3. Discussion	11
V. Conclusion	12
VI. Recommendations	12
VII. Acknowledgements	13
References	14

---

<sup>1</sup> PLTOFF, School of Engineering & Information Technology. ZEIT4501 & ZEIT4297.

## APPENDICES

Appendix A. Arduino Pro Mini processor testing method and results	A1
Appendix B. RFM96 LoRa radio module testing method and results	B1
Appendix C. Satellite radio beacons software flow chart and design considerations	C1
Appendix D. Satellite beacon electrical power regulation testing method and results	D1
Appendix E. Total beacon electrical power requirements and generation testing method and results	E1
Appendix F. Satellite beacon electrical power storage testing method and results	F1
Appendix G. Communication link testing method and results	G1
Appendix H. Communication link budget calculations	H1
Appendix I. Ground receiving station software flow chart	J1
Appendix J. Ground receiving station time measurement error investigation, testing and results	K1
Appendix K. Ground receiving station final verification testing method and results	L1

## Nomenclature

*Terms, Abbreviations and Acronyms:*

ACMA	= Australian communications and media authority
APM	= Arduino pro mini
BW	= Bandwidth
CR	= Coding rate
CRC	= Cyclic redundancy check
EEPROM	= Electrically erasable programmable read-only memory
EIRP	= Effective isotropic radiated power
FPGA	= Field-programmable gate array
FSPL	= Free-space path loss
GNSS	= Global navigation satellite system
IDE	= Integrated development environment
ISR	= Interrupt service routine
ITU	= International telecommunication union
LDL	= Low dropout
LED	= Light emitting diode
LEO	= Low earth orbit
LPWAN	= Low-power wide area network
NB-IoT	= Narrowband – Internet of things
PPS	= Pulse per second
RF	= Radio frequency
RTL-SDR	= Realtek - software defined radio
RX	= Receive
SDR	= Software defined radio
SF	= Spreading factor
SMA	= SubMiniature version a connector
SRAM	= Static random access memory
SSA	= Space situational awareness
TDOA	= Time difference of arrival
TT&C	= Telemetry, tracking and control
TX	= Transmit
UART	= Universal asynchronous receiver-transmitter
UHF	= Ultra high frequency
USB	= Universal serial bus
UTC	= Coordinated universal time

## I. Introduction

The reduction in economic and resource cost of designing, manufacturing, and launching a small satellite has led to an increased number of small satellites being operated in the Low Earth Orbit (LEO) space environment<sup>1</sup>. A greater number of objects in the LEO environment has resulted in new challenges for space situational awareness (SSA) and has led to a larger reliance on resource expensive space monitoring equipment to maintain situational awareness in the LEO environment. The reduced cost of launching a small satellite into the LEO environment is a result of releasing multiple small satellites from a single launch platform, this is commonly known as ride sharing. Ride sharing has resulted in upwards of 100 small satellites being released from the same launch vehicle in a small-time frame<sup>2</sup>. This has resulted in a reduction in SSA immediately after the launch of small satellites and throughout their operational life cycle due to difficulties in identifying individual satellites. The difficulties of identifying small satellites in the LEO space environment results in a greater demand on the limited ground-based optical and radar monitoring resources to maintain an SSA capability. The standardisation of small satellite manufacture has reduced the cost of production allowing government, educational and commercial organisations with little to no space mission experience to create and produce small satellite designs. The unique development and design process to meet the organisation requirements for each satellite has resulted in a 55% failure rate for academic institutions and a 23% failure rate for commercial industry<sup>3</sup>. The cause of a small satellite failure is difficult to determine as the failure can reduce the quantity of satellite telemetry data available for fault-finding. A deficiency of telemetry data can result in the determination of satellite failure being contributed to several causes of possible failure (typically 5-10 possible causes). The ride sharing launch produces additional difficulties in identifying an individual satellite immediately after release by the ground monitoring stations which leads to an increase in failure rates due to difficulties in creating the initial communication link with the individual satellite<sup>4</sup>. A satellite failure can result in the satellite becoming uncontrollable and/or difficult to track causing it to become a space debris object leading to an increased risk of a collision with other objects within the LEO space environment.

To reduce the small satellite failure rate, increase SSA in the LEO environment and reduce the reliance on space monitoring equipment a solution needs to be investigated that aims to provide better methods of obtaining satellite identification and telemetry data and provide a cost-effective method of tracking space debris, active and end of life satellites in the LEO space environment.

The purpose of this project will be to design a self-sufficient, independent satellite radio beacon system that can transmit satellite identification and telemetry data and receive control commands using a UHF radio signal that can be tracked using multiple geographically dispersed, cost-effective ground receiving stations.

The satellite radio beacon shall primarily allow for the identification of an individual satellite after launch and its operational life cycle providing greater SSA which leads to a lower risk of collisions in the LEO environment. Multiple graphically dispersed ground mounting stations constructed using low-cost, commercially available components are to be utilised to track the satellite via the beacon radio signal allowing the existing space monitoring systems to focus on other LEO space objects. The beacon will have a secondary function that can provide telemetry data for satellite on-orbit fault finding to facilitate the determination of causes of failure. Determining the actual cause of failure as opposed to having several possible causes of failure is expected to reduce the number of failures in future launches and operations. This system can be extended to include an alternative communications pathway that can be used to provide limited control of the other satellite systems to offer a redundancy system to correct on-orbit failures. Correcting an on-orbit failure can result in regaining control of the small satellite reducing the number of space debris objects in the LEO environment.

## II. Aim

The aim of this project is to design and produce a ground-tested satellite UHF radio beacon prototype and a cost-effective ground monitoring station prototype that can sustain a communications link for the distances required of a satellite in LEO. In order to achieve this aim, three aspects will be investigated, firstly the satellite radio beacon that is to be a self-sustained UHF communication system capable of operating independently of all other satellite systems for the duration of the satellites operation mission (until deorbit). Secondly, the UHF communications link which must be able to support the reliable transmission of satellite identification and telemetry data up to 2000kms to support the operation and monitoring of satellites in the LEO space environment. Finally, the ground monitoring station which must be capable of capturing the satellites identification data, telemetry data, recording the precise time of arrival of the radio signal and determining its own global location. The ground station will be able to pass on the captured data and measurements to a peripheral device for post processing using the time difference of arrival (TDOA) calculation technique to determine the position of the satellite through trilateration.

#### IV. Background

The theory of operation of the satellite beacon system is that the system contains its own power, processing and radio systems to ensure that it is self-contained and independent of the other satellite systems. Each radio beacon system contains an identifier in the form of a 16-bit address which allows for 65,536 unique addresses to be used simultaneously. Each radio beacon processor has the capability to be linked to other systems within the satellite to allow for the collection of satellite telemetry and health data or to pass a received command to another system. The 16-bit identification address and collected telemetry data are sent via the beacon radio to a ground receiving station to uniquely identify the small satellite and provide satellite telemetry data for tracking and fault-finding purposes. The radio beacon system can receive data from a ground receiving station to provide control of the beacon system and emergency control of the small satellite.

The ground station can receive and process the identification and telemetry data as well as providing a precise time of arrival of the received signal. The precise time of arrival for a unique small satellite beacon signal being received at three globally dispersed ground receiving stations allows the position of the satellite to be calculated. The global position of each ground station must be known with each station being synchronized to the same clock timing to allow for the TDOA calculation technique to approximate the satellites position as shown in Figure 1 (see right). The trilateration of position from three stations reduces the possible position of the satellite to two spatial locations with one being discarded as unfeasible due to it being below the surface of the earth. The uncertainty in the position of the satellite decreases as the number of ground stations used in the TDOA calculation increases.

A review of the current systems available has shown that solutions are provided for individual aspects of SSA in the LEO environment, but they do not take a holistic approach to the key parameters of a satellite telemetry, tracking and control (TT&C) system which includes satellite identification. There are several systems available that provide identification (CUBIT<sup>5</sup>, SOARS, Passive RF tag<sup>6</sup>, ELROI<sup>7</sup> and LEDSAT<sup>8</sup>) or telemetry (safety radio beacon<sup>9</sup>) and some which provide identification and telemetry (HYELT, RILDOS<sup>10</sup> and IRASAT1<sup>11</sup>). The existing systems deliver solutions for one or two parameters of the TT&C system which address either, but not both problems highlighted in the introduction. The proposed satellite radio beacon system will implement solutions for all four key parameters of a TT&C system that will address both identified problems.

The major orbital parameters that will be used for this project are based on a generic small satellite mission in the LEO environment. The most common orbits for a small satellite mission are sun-synchronous with an orbital height of 300-400kms and inclination of 52° or an orbital height of 500-800kms and inclination of 98°<sup>12</sup> with the later parameters used for the testing of this system. The orbital parameters used for testing results in an orbital period of roughly 90 minutes with the view window of each pass being in the vicinity of 8-10 minutes in which the satellite and ground station have a slant range of between 500kms to 2000kms.

A key goal of this project is to ensure that the design of the ground monitoring station can be carried out such that the costs, difficulty of construction and the difficulty in operating the equipment is minimized. The result of minimizing these aspects of the design will allow a broader spectrum of the public, particularly the growing space enthusiast community, to create their own ground monitoring stations. This allows a larger number of ground stations that are more geographically dispersed which has a two-fold effect of increasing the footprint and tracking reliability of the ground stations and increasing the public awareness and engagement.

#### V. The present study

The design of the satellite radio beacon system will be broken up into three major components: the satellite radio beacon, the communication link and the ground receiving station. The satellite radio beacon contains three sub-system: (1) Computer processing, (2) Radio transceiver, and (3) Power generation, storage, and regulation. The communications link is a standalone component while the ground receiving station will be separated into three sub-systems: (1) Computer processing, (2) Radio transceiver, and (3) GNSS module.

##### A. Satellite Radio Beacon

The first component of the system to be investigated is the feasibility of a satellite radio beacon that is a self-sustaining system and independent of all other satellite systems. The beacon must contain its own processor, radio and power generation, storage and regulation system to ensure it is self-sustained and can continue independent operation if there is a failure in another satellite system. The major constraint for the design is that the electrical power generated can meet the electrical requirements of each sub-system. The components used in each sub-system will be selected to ensure self-sufficiency, independence, cost-efficiency, ease of operation and minimization of size and weight.

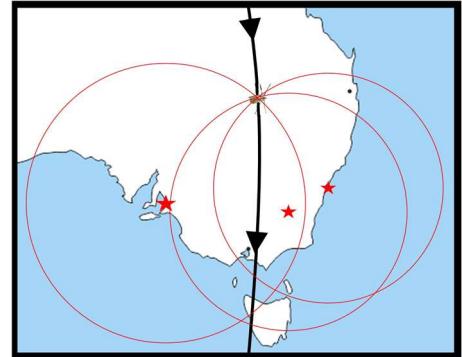


Figure 1 - Visual representation of TDOA trilateration with 3 stations

## 1. Materials and Methods

The first decision in the design process was to select the components for the initial design of the radio beacon with the focus being on components that have low power consumption and can meet the requirements of each sub-system. The solutions researched for the processor were a Teensy based microcontroller, an Arduino based microcontroller, a raspberry Pi and using a FPGA board with the 3.3V, 8MHz Arduino Pro Mini module (APM) using a ATMEGA328P<sup>13</sup> processor being selected for the initial design. The APM was selected as it fit all the requirements of the processor system, it has a proven space heritage, it is well resourced and can be operated with a low supply voltage and clock speed to reduce power consumption. The APM module contains an in-built power regulation system using a Low Dropout (LDO) voltage regulator which will be used in the initial design. Three radio systems that are capable of long-range communications were found during the initial investigation for the radio design solution: The LoRa spread spectrum system, SIGFOX Low-Power Wide Area Network (LPWAN) system and the NB-IoT LPWAN system. The LoRa radio system<sup>14</sup> was the selected medium for the initial design for it provided a superior point-to-point communications protocol, high immunity to noise and doppler shift, greater software support and can operate in the 70cm (430MHz, RFM96 module) and 33cm (915MHz, RFM95 module) band radio spectrum. A LoRa module breakout board designed by Boyan Nedkov<sup>15</sup> was used to allow for compatibility with a prototyping solderless breadboard. Solar power was the only system considered for the power generation system due to the difficulties of operating other power sources (Lithium-ion batteries, hydrogen fuel cells, nuclear power, thermo-photovoltaic cells, etc.) in a space environment. The solar panel selected for ground testing is a 0.5W, monocrystalline Silicon panel from Seeed<sup>16</sup> which has a typical output of 5.5V with a current of 100mA at 17% solar conversion efficiency. A capacitor-based energy storage system was selected for the initial design due to the launch isolation and ground testing requirements of a battery-based system.

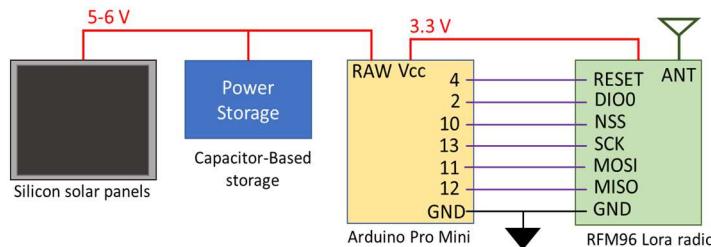


Figure 2 - Satellite radio beacon prototype design for initial testing

The initial design of the satellite radio beacon system was constructed on a prototyping breadboard (refer Figure 2 to the left) using Arduino IDE<sup>17</sup> software to operate the APM module with the LoRa radio module being driven by the Radiohead library developed by Airspayce<sup>18</sup>. To ensure that the beacon can meet the self-sufficient and independent requirements, the initial focus of the testing was to minimize the current consumption of the beacon sub-systems to reduce the number and size of solar panels required. The first set of tests carried out were to support the investigation of minimizing the current consumption of the processing sub-system (APM module) with the testing method and results detailed in Appendix A. The next investigation was to determine methods of minimizing the current consumption of the LoRa radio module which is supported by the testing method and results shown in Appendix B. The next step in the design process was to develop the software that will be utilised in the satellite radio beacon which is based upon the software flow chart and design considerations in Appendix C. The final investigation for current consumption minimization was carried out on the power regulation system with the testing method and results presented in Appendix D. At the conclusion of the current minimization investigation, testing was carried out to determine the total current requirements of the beacon system and methods to reduce the number solar panels required for the successful operation of the satellite radio beacon system which is set out in Appendix E. The final investigation of the beacon design was to develop the electrical power storage sub-system with the testing method and results shown in Appendix F.

## 2. Results

Table 1 - Average current consumption for the APM module processor and built-in regulator

Average current consumption for APM module power modes				
Input Voltage and resistor	RAW=5.0V,1Ω	RAW=3.3V,1Ω	Vcc=3.3V,1Ω	Vcc=3.3V,10Ω
power on current (mA)	50.67	13.78	13.51	14.66
powerDown current (mA)	42.25	10.00	10.00	11.43
powerSave current (mA)	42.30	10.00	10.00	11.43
powerStandby current (mA)	42.59	10.41	10.09	11.53
powerExtStandby (mA)	42.54	10.41	10.09	11.53
idle (mA)	46.15	10.86	10.58	12.20

A summary of the results from testing the APM module is shown above in Table 1 with the results indicating a higher than expected current measurement for the ATMEGA32P processor by 10mA for all modes, which is attributed to a green LED that constantly consumes 10mA when power is applied. The results show a significant increase in the APM current consumption if the supply input voltage is increased from 3.3V to 5V. The current consumed by the APM module is shown to be reduced when utilising the power saving modes of the *LowPower* library with the largest reduction occurring when the *powerDown* or *powerSave* modes are used.

Table 2 - RFM96 LoRa module average current consumption for each mode of operation

RFM96 LoRa module current measurements for each mode			
Series resistor value ( $\Omega$ )	1	10	20
Sleep current (mA)	0.70	0.62	-0.09
Receive current (mA)	11.00	10.00	11.67
No data transmit current (mA)	2.00	2.00	4.40
Idle current (mA)	2.00	2.00	2.00

The results for testing the current consumption of each mode of operation for the LoRa radio module is summarized above in Table 2 and the results show that the RFM96 LoRa module consumes the smallest amount of current when it is in *sleep* mode and the largest in the *receive* mode (unless radio packets are being transmitted).

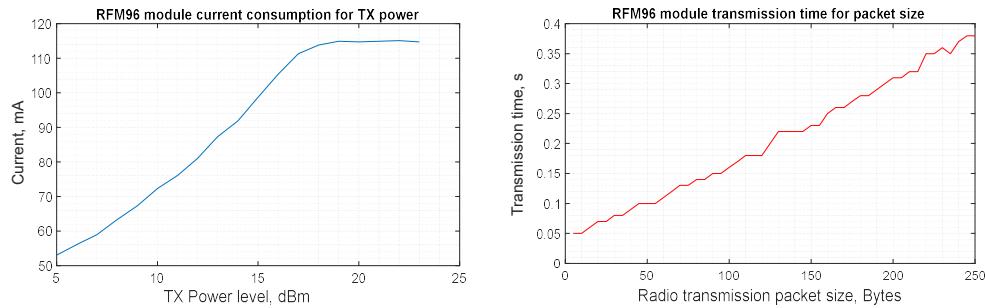


Figure 3 - RFM96 module current consumption for TX power level and transmission time for radio packet size results

The results of the current consumption measurements detailed above in Figure 3 shows the LoRa module current increases linearly (4.5mA/dBm) as the TX power increases until 17dBm is reached in which it plateaus at 117mA. The results indicate that the transmission time increases linearly by 10ms when the radio packet size increases by 5 bytes when using the (0) default RadioHead radio settings (medium data rate and range).

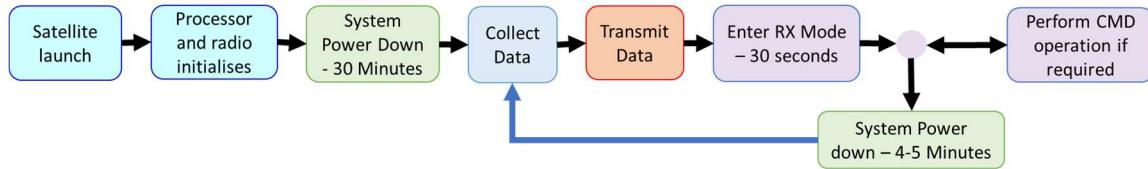


Figure 4 - Satellite radio beacon software cycle

The development of the software program for the satellite radio beacon is based upon the software cycle shown above in Figure 4. The validation process of the software cycle discovered that a small number of packets were not being received by a second LoRa module (The transmission of the radio packet was verified using an RTL-SDR<sup>19</sup> USB dongle driven by the Gqrx SDR<sup>20</sup> receiver software) with further investigation into this being carried out during the testing of the communications link. The receive function of the beacon software cycle was tested successfully with the radio beacon able to receive and execute a command. The APM module required 6 pins to be utilised (not including power, ground or connections for gathering telemetry data) for the successful operation of the developed satellite radio beacon software cycle.

Table 3 - Beacon total current consumption utilising different voltage regulators

Software cycle	MC5205	LM1086	LM3671	TS2940CZ
Launch (mA)	0.75	6.12	1.13	3.02
Collect Data (mA)	7.71	11.76	5.11	8.30
Transmit data (mA)	115.73	125.76	88.75	126.25
Receive mode (mA)	16.62	20.58	11.67	17.38
Idle mode (mA)	1.82	6.67	1.18	3.69

The results for the regulator testing is detailed above in Table 3 and reveal that the LM1086<sup>21</sup> and TS2940CZ<sup>22</sup> LDO regulators have a much higher current consumption than the MC5205<sup>23</sup> LDO regulator and LM3671<sup>24</sup> Step-Down DC-DC converter (buck converter) during all phases of the software cycle. The inbuilt regulator and buck converter have similar quiescent currents throughout the software cycle except for during the *transmit* phase in which the LM3671 buck converter consumes 27mA less current.

Table 4 - Total current consumption measurements of the satellite radio beacon

Software cycle	1Ω resistor	2Ω resistor	Average
Initialisation (mA)	55.58	55.40	55.49
Launch (mA)	1.50	0.75	1.13
Collect Data (mA)	5.41	4.89	5.15
Transmit data (mA)	88.75	88.66	88.70
Receive mode (mA)	11.94	9.82	10.88
low-power mode (mA)	1.48	1.12	1.30

The total current consumption of the satellite radio beacon for each phase is summarized on the previous page in Table 4 with tests showing that a single solar panel could not support the beacon operation if the TX power level was set above 10dBm without using an additional energy source. When 11mF of electrolytic capacitance was used as a supplementary energy supply to support the *transmit* phase then the radio beacon could be operated up to the maximum TX power (23dBm) when connected to one solar panel.

If the electrolytic capacitors are replaced with a single 5.5V,1F super-capacitor then the beacon system can be supported through all phases of the software cycle but a delay to the start of the software cycle is introduced. If a single super-capacitor is used, there is a 1-minute delay between when the solar panel starts to generate electrical power until there is enough energy to power all the sub-systems (2.6V potential) which increases to 8 minute and 20 seconds if the number of super-capacitors is increased to 5. The super-capacitor storage system reaches its full electrical potential after 27 minutes when the software cycle is 19 minutes into the *launch* phase. The average potential of the super-capacitor storage system after a one charging cycle was 6.3V which equates to 101 Joules of stored energy. The super-capacitor storage system, when fully charged, can support 1 hour and 14 minutes of beacon operations when no electrical power is being generated by the solar panel.

### 3. Discussion

The results of the investigation into the minimization of the current consumption of the satellite radio beacon has shown that the beacon is self-sustaining and can operate independent of all other satellite systems. The testing carried out found the following methods to minimize the total current consumption of the beacon system:

- Place the ATMEGA328P processor into *powerDown* or *powerSave* low power modes where possible and remove all superfluous components
- Place the LoRa radio into *sleep* mode where possible and reduce the length of the *receive* and *transmit* phase to the shortest time required
- Reduce the radio transmission power to as low as possible
- decrease the size of radio packets to the smallest size to transfer data
- Utilize a more efficient voltage regulator, LM3671 Buck converter

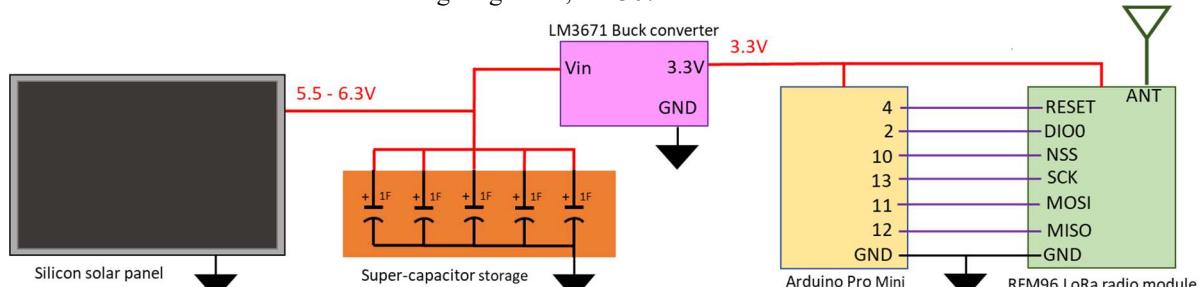


Figure 5 - Satellite beacon system configuration and connections for ground testing

One solar panel could not sustain operation of the beacon system without using an electrical storage system to support the spike in current consumed during the beacons *transmit* phase. The energy required to sustain a radio transmission was greater than the energy that could be supplied by one silicon solar panel. This led to the design of an energy storage system using five 1F super-capacitors that can support the maximum transmission power with one solar panel and can meet the no power to be stored requirements of the launch providers<sup>25</sup>. The storage system allows for 73 minutes of beacon operation when no power is being generated in low irradiance conditions and minimizes the space requirements of the satellite beacon by reducing the number of solar panels. The final design for the breadboard-based satellite radio beacon prototype is shown above in Figure 5.

The software cycle developed was based upon the software flow presented in Figure 4 on page 6 which must include a 30-minute period of radio silence immediately after release from the launch vehicle to meet the requirements of the launch provider (the delay caused by the super-capacitors can be included in this period). The radio beacon must have the ability to accept a command to silence the radio transmissions to meet the RF spectrum requirements<sup>26</sup> set by the ACMA and ITU which dictates that the software cycle must contain a receive phase. The communication link testing is required to be conducted to determine the LoRa radio settings and the format of the radio packet data before the satellite radio beacon software code could be finalized.

The connections to the APM module required to operate the beacon are shown above in Figure 5 which allows for a combination of analogue, digital and serial connections to other satellite systems to collect telemetry data. The basic software program uses 28% of flash memory and 50% of SRAM with 22k bytes of flash memory, 1k bytes of SRAM and 1k bytes of EEPROM remaining for use in the telemetry collection programming.

A waterfall design approach for the development of the satellite radio beacon system led to each component of the system being researched, designed and tested to ensure the beacon is self-sustained and independent from all other satellite systems. The beacon system can be constructed using a solderless breadboard with the final software configuration to be determined after the communication link testing.

## B. Communications Link

The communications link between the satellite radio beacon and ground receiving station must be able to sustain the reliable transfer of data for the expected slant range for a satellite in LEO. The orbital parameters for the missions in the LEO environment determine that the maximum distance to be supported by the data link is 2000kms which equates to a signal free-space path loss of 151.3dB. The TX power and radio configuration settings of the LoRa module will be determined based on the results of the link budget calculations.

### 1. Materials and Methods

The first set of tests to be carried out for the communication link is to investigate the dropped radio packets that was found during the software testing and determine if there are methods to mitigate these losses. A detailed description of the methods used to carry out the testing is found in Appendix G with the tests first determining if changing the settings of the LoRa radio module or test conditions affects the number of packets dropped, which is known as the Packet Error Rate (PER). Secondly, a set of tests were conducted to analyze the presence of bit errors in the transferred data. The final set of tests utilized functions within the *RHGenericDriver* and *RH\_RF95* files used in the RadioHead library that monitor the quality of transmitted and received data packets.

The next step in the communication link testing is to determine the radio settings that will allow the transmission of data to a satellite in LEO. The first step was to calculate the receiver sensitivity and link budget for the default and long-range radio settings available in the RadioHead library and compare against the values from the LoRa modem calculator tool available from Semtech<sup>27</sup>. A transmitter and receiver LoRa module were then connected with a series of cables and attenuators which simulate the Free Space Path Loss (FSPL) to verify the results of the calculations and determine the reliability of the transfer of data for the distance expected in an LEO mission. The methods used for testing and the results is detailed in Appendix G with the results and current RF regulations used to determine the radio settings and TX power that will be used for the final system testing.

The final step in the testing of the communications link is to perform all the necessary calculations to determine the link budget and compare against the estimates of the LoRa modem calculator tool. The method in which the communications link budget was calculated is detailed in Appendix H with the results of the link budget determining the TX power level and radio settings selected for LoRa radio module during the prototype testing.

### 2. Results

The investigation into the radio parameters and test settings show that the two settings that record a reduction in dropped packets is the Cyclic Redundancy Check (CRC) setting and the coding rate. If the CRC is turned on, then the number of packets dropped can be reduced by up to 50% and if the number of CRC check bits are increased from 1 to 4 then up to 75% reduction in dropped packets was observed. This reduction in the number of dropped packets is not observed consistently using the different testing parameters and as such is a general observation with no trend being identified. There was a total of 8 tests carried out that analyzed the content of each byte which found that there were no bit errors present in 79,904 bytes received. Using the functions within the *RHGenericDriver* file revealed that the reason for the LoRa data packets not being received cannot be determined. There were a total of 85 packets dropped when the CRC was on with 79 being determined to have errors and 6 not received at all and there were 12 packets dropped when CRC was turned off with 3 packets determined to have errors and 9 not received at all.

Table 5 - RadioHead Library default LoRa module settings with LoRa calculator bit rate and receiver sensitivity estimates

LoRa module radio configuration	Bandwidth	Coding rate	Spreading factor	Preamble length	Estimated Bit rate	Estimated receiver sensitivity
Medium range & data rate	125 kHz	4/5	7 (128 chips/symbol)	12	5469 bps	-124 dBm
Long range, slow data rate	31.25 kHz	4/8	10 (512 chips/symbol)	12	152.59 bps	-139.4 dBm
Long range, slow data rate	125 kHz	4/8	12 (4096 chips/symbol)	12	183.11 bps	-138 dBm

A summary of the RFM96 LoRa radio module configuration from the RadioHead library default settings with a bit rate and receiver sensitivity estimate from the LoRa modem calculator tool is presented above in Table 5. The results indicate that the (2) default settings could operate over a longer distance using a smaller portion of the radio spectrum and has a comparable bit rate to the (3) default settings.

Table 6 - LoRa calculator Link Budget estimates and testing measurements for the RadioHead default settings

Largest measured FSPL attenuation and distance for reliable data transmission													
Transmit power (dBm)		5			10			15			20		
RadioHead default setting	Receiver sensitivity (dB)	Estimated Link Budget (dB)	attenuation (dB)	Distance (kms)	Estimated Link Budget (dB)	attenuation (dB)	Distance (kms)	Estimated Link Budget (dB)	attenuation (dB)	Distance (kms)	Estimated Link Budget (dB)	attenuation (dB)	Distance (kms)
(0) Bw125Cr45Sf128	-124	129	131.14	195	134	137.14	390	139	141.14	620	144	145.14	980
(2) Bw31_25Cr48Sf512	-139.2	144.4	146.14	1100	149.4	151.14	1950	154.4	155.14	3100	159.4	161.14	6200
(3) Bw125Cr48Sf4096	-138	143	144.14	880	148	150.14	1750	153	155.14	3100	158	160.14	5500

Table 7 - LoRa calculator transmission time estimates and testing measurements for the RadioHead default settings

RFM96 LoRa radio transmit times for RadioHead default settings			
RadioHead default setting	(0) Bw125Cr45Sf128	(2) Bw31_25Cr48Sf12	(3) Bw125Cr48Sf4096
Estimated Ident TX time (s)	0.06	1.057	1.057
Estimated telemetry TX time (s)	0.183	3.416	2.892
Estimated total TX time (s)	0.423	7.643	7.119
Measured ident TX time (s)	0.03	0.59	1.01
Measured telemetry TX time (s)	0.09	2.02	3.04
Measured total TX time (s)	0.23	4.39	7.1

The results presented in Table 6 (previous page) and Table 7 (above) compare the estimated values from the LoRa calculator against the measured values obtained from the testing. The results indicate that the LoRa calculator link budget estimates are a fair representation of the measured results, while the transmission times were slightly unreliable. The results from testing the communications link are shown below in Table 8 and reveal that the (2) radio settings can sustain a communication link for a larger distance (1dB greater FSPL attenuation) for a smaller transmit time (40% less Time-on-Air) than the (3) settings. The results from the testing also indicated an approximate 20% increase in bit energy when comparing the (2) settings against the (3) settings.

Table 8 - Link budget calculations for the satellite beacon to ground station communications link

Link Budget calculations for the satellite radio beacon communications link at 2000kms													
RadioHead default settings used		(0) Medium Range Settings				(2) Long Range Settings				(3) Long Range Settings			
Parameter	Symbol	Downlink	Downlink	Downlink	Downlink	Downlink	Downlink	Downlink	Downlink	Downlink	Downlink	Downlink	Downlink
Transmitter Power (dBm)	Pamp (dBm)	5	10	15	20	5	10	15	20	5	10	15	20
Carrier Power Density (dBW)	C'	-173.77	-168.77	-163.77	-158.77	-173.77	-168.77	-163.77	-158.77	-173.77	-168.77	-163.77	-158.77
Bit Energy/Noise Density Ratio	E <sub>b</sub> /N <sub>0</sub>	-8.73	-3.73	1.27	6.27	4.382	9.382	14.382	20.382	3.590	8.590	13.590	18.590

A summary of the link budget calculations is presented above in Table 8 which shows that the communication link for the LoRa radio module has a positive bit energy to noise density ratio ( $E_b/N_0$ ) when using both long range settings in the RadioHead library. A  $E_b/N_0$  ratio above 10 maintains a reliable communications link which is achieved by setting the TX power to 15dBm or above for both RadioHead default long range settings.

### 3. Discussion

The investigation to determine if the number of packets dropped can be reduced could not provide a set of radio parameters or test conditions that produce consistent results. The testing showed a small portion of the packets not received were rejected by the LoRa module (payload/header CRC check errors, checksum errors, bad lengths, etc.) with the rest not being received by the module at all (no preamble detection). Further investigation into the settings for the LoRa radio module was conducted that allows the data that is received by the module to be passed to the Arduino module for processing regardless of any errors present which found that the proprietary nature of the SEMTECH software prevented any modification of the LoRa receive process.

The number of packets dropped by the LoRa module through the whole testing process was determined to be less than 1% of the number of packets transmitted. To prevent the loss of all data at the receiving station, the transmitted identification and telemetry data will be separated between several radio packets. The identification of each satellite is 16 bits (2 bytes) with each radio packet containing two sets of addresses preceded by a radio packet identifier (1 byte) for a total of 5 bytes. The packet will be repeated four times during each *transmit* phase of the radio beacon followed by one packet of 50 bytes containing the telemetry data. This transmission format was selected due to the inability of preventing radio packets from being dropped by the LoRa receiver and it reduces the probability of the satellite identification not being collected to 0.00001%.

The calculations for the communication link budget show a significant increase in the bit energy to noise ( $E_b/N_0$ ) when using the long range settings from the RadioHead library with the (2) settings providing a higher ratio. The smaller bandwidth (31.5kHz) of the (2) settings results in roughly 20% higher energy per bit as opposed to the (3) settings (125kHz) while using a smaller portion of the RF spectrum. The maximum TX power levels of the LoRa module are well below the maximum level that can be transmitted on the 435-438MHz RF spectrum as stipulated by the ACMA amateur license conditions. This allows the radio to be used at its maximum TX power (23dBm) with the decision made to initially use a 15dBm TX power to balance the beacons electrical power requirements while maintaining an  $E_b/N_0$  above 10. There are several losses (atmospheric, weather, scintillation, polarization, etc.) that are difficult to quantify that have not been included in the link budget calculations but there is tolerance in the  $E_b/N_0$  ratio to compensate for their losses. The testing revealed that a LoRa module with the (2) default settings has less Time-On-Air, occupies a smaller bandwidth of the radio spectrum and has higher bit energy contributing to the decision that the (2) default settings will be used for all further testing.

The testing carried out on the communications link has determined that the RFM96 LoRa radio module operating at 437MHz can support a reliable data link between a satellite radio beacon in LEO and a ground receiving station. The LoRa module uses the default (2) setting from the RadioHead library with a TX power of 15dBm to maintain a  $E_b/N_0$  above 10 for a 2000km data link. The testing found that the PER of the LoRa module requires separate packets of data be sent to include a redundancy that ensures the satellite identification is received at the ground station during every *transmit* phase of the radio beacon cycle.

### C. Ground Receiving Station

It has been shown that the communication link can support the reliable transfer of data between the satellite beacon in LEO and an Earth-based station which leads to the design of the ground receiving station. The major constraint for the ground receiver is to produce a design using cost effective components that has an acceptable level of tolerance in determining the position of the satellite. The position of the satellite can be estimated using the time difference of arrival of the transmitted beacon radio signal at three ground stations which requires a common timing source between the stations and an accurate method of measuring time.

#### 1. Materials and Methods

The first design decision was to utilize the Global Navigation Satellite System (GNSS) in the ground receiving station to provide the geospatial location of the ground station, the Coordinated Universal Time (UTC) for a common timing source and a pulse per Second (PPS) signal for synchronizing the processor clock timing. The initial design of the ground receiving station will include a global positioning development module based on the U-Blox NEO-7M<sup>28</sup> GNSS module as it has a positional accuracy of 2.5m, a PPS signal available on an external pin with a 30ns tolerance and can be connected to a 28dB GNSS Antenna for increased GNSS signal reception.

The second design decision was to utilize the RFM96 LoRa module on the same breakout board as used in the beacon prototype allowing the use of the same software and libraries to provide continuity between the two components of the system. The breakout board used in this design contains a SMA connector allowing for a simple 3.5dBi Helix antenna or a 12dBi Dipole antenna to be used for testing.

The last design decision before development begins was to select the computer processor that will accurately measure time to a precision required for this application and can drive the LoRa and GNSS modules. The solutions were limited to an Arduino based MC board to ensure a processor was selected that is well resourced, has enough peripherals and is easy to use. This reduces the available solutions to the Arduino Uno or Arduino Due development boards. The Uno was selected for the initial design as the component cost was substantially less with it being noted that the Due processor speed (84MHz) is much higher than the Uno (16MHz) and could increase the precision of time measurement if required. The initial design of the ground receiving station is shown in Figure 6 (see right) with the total cost for the ground station being under AU\$80.

After the initial design decisions were made, the next step was to develop the software code for the ground receiving station which is derived from the software flow chart detailed in Appendix I. The initial operation of the ground receiving station was planned such that as the ground station receives each packet (four identification and one telemetry radio packets) it produces a time stamp that marks the time of arrival referenced against the next GNSS PPS signal after the telemetry packet. The GNSS data is then captured which contains the current UTC time to a second precision and positional data of the ground station. The number of clock cycles between each packet time stamp and the PPS signal is used to determine the precise time of arrival of each radio packet with relation to the GNSS provided UTC. The timestamps, GNSS data (position and UTC), the identification data and the telemetry data are passed on to another device for further processing. The measured difference time of arrival, synchronization of GNSS UTC and known position of three ground receiving stations allows for the calculation of the estimated satellite position using a TDOA ranging technique.

The primary focus of testing the ground receiving station was to determine the possible sources of uncertainty in measurements that would determine the accuracy of the satellites estimated position. The initial research and testing of the system identified the following sources of uncertainty that could produce an error in the calculation of distance between the beacon and ground stations:

- 1) The resolution of the Arduino built-in timer function, *micros()*
- 2) The number of clock cycles taken to carry out an Interrupt Service Routine (ISR)
- 3) Oscillator frequency drift due to temperature, tolerances and other sources of error
- 4) Tolerance of the GNSS PPS signal
- 5) Accuracy of the ground station GNSS position
- 6) The time taken for the LoRa module built-in software to carry out integrity checks (time between when the signal is received and when it is made available to the Arduino module)
- 7) The resolution of the ATMEGA328P processor clock cycles using an external oscillator

The detailed description of how each uncertainty in measurement was investigated and tested with results are presented in Appendix J. The results from the testing was combined to determine the total uncertainty in measurement between two ground stations with the final verification testing being carried out to verify the total uncertainty in measurement with the testing method and results shown in Appendix K.

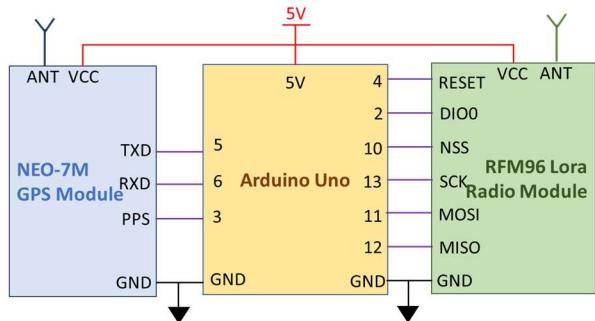


Figure 6 - Satellite ground receiving station final design

## 2. Results

The initial testing of the time measurement using the Arduino IDE software built-in function, *micros()*, revealed that the resolution of this measurement was found to be 4.096 $\mu$ s. The *micros()* function as implemented by Arduino has the output of the function being incremented only when the ATMEGA328P processor timer0 overflows resulting in the 4.096 $\mu$ s timing resolution which equates to distance calculation error of up to 1.23km.

The investigation into using the ATMEGA328P processor Timer1 to count the number of clock cycles between events revealed an uncertainty in the clock cycle count between successive events and the presence of a spike in the clock cycle count error (up to 190 clock cycles) occurring at irregular intervals. The results presented in Appendix J show that the difference in clock counts between uniformly timed events was less than 3 clock cycles for 97.3% of the measurements. When the spike in clock count errors were removed from the samples then the resultant statistics shows the uncertainty in clock cycles between events has a mean of 0.0266 and standard deviation of 1.1636 which results in 95% of the measurements being between -2.30 and 2.36 clock cycles.

The investigation into how the ISR were carried out revealed that the method for entering the routine for counting clock cycles (Timer1 ISR, *TIMER1\_COMPA\_vect*) and for the GNSS PPS pulse (External interrupt ISR, *INT1\_vect*) are exactly the same and uses the same amount of clock cycles to enter, execute and leave an ISR. The method of using an the clock counting ISR or external interrupt ISR is common between each ground station and does not change the difference in time for the TDOA calculation.

The tolerances for the errors in the GNSS PPS signal (30ns) and the GNSS positional measurement of the ground station location (10m) were found to be acceptable as the uncertainties are much smaller than the uncertainty in measuring time and no further testing was carried out.

The testing of the time difference in processing of two LoRa modules shows that the processing time difference has a mean of -0.1803 $\mu$ s with a standard deviation of 2.7966 $\mu$ s. The cumulative distribution function of the measured samples was used to determine that 92.36% of the measured values lie between  $\pm 5\mu$ s.

*Table 9 - Final verification testing total error in timing measurement between 2 ground station measurements*

	Measured values				Absolute values			
	mean ( $\mu$ s)	Std Dev ( $\mu$ s)	95% of data interval (ms)		mean ( $\mu$ s)	Std Dev ( $\mu$ s)	95% of data interval (ms)	
Address packet 1	19.816	638.83	-1.257844	1.297476	381.48	512.66	0	1.4068
Address packet 2	17.482	681.87	-1.346258	1.381222	409.35	545.46	0	1.50027
Address packet 3	38.799	652.44	-1.266081	1.343679	389.8	524.49	0	1.43878
Address packet 4	51.243	624.07	-1.196897	1.299383	379.93	498.02	0	1.37597
Telemetry packet	24.433	759.46	-1.494487	1.543353	419.14	633.66	0	1.68646
	<b>Averaged values</b>				395.14	250.2	0	0.89554

The results of the final verification testing are summarized above in Table 9 and show that the total absolute uncertainty in timing measurement has a mean of 380 $\mu$ s with a standard deviation of 520 $\mu$ s for the address packets and a mean of 419 $\mu$ s with a standard deviation of 634 $\mu$ s for the telemetry packet. This equates to 95% of the absolute measurement uncertainty of the address packets being less than 1.5ms and 95% of the telemetry uncertainty being less than 1.7ms which equate to a distance calculation error of 450km and 510km, respectively. If the absolute uncertainty in timing measurement is averaged between all five packets, then the mean is 395 $\mu$ s with a standard deviation of 150.2 $\mu$ s and 95% of the measured values below 0.9ms which is a distance calculation error of 270kms.

## 3. Discussion

The investigation into a method that would reduce the level of uncertainty utilizes the counter of the Timer1 clock cycles in the ATMEGA328P processor which has a resolution of 62.5ns. The testing carried out determined that there was a tolerance of  $\pm 3$  clock cycles with 97% confidence when using the clock counting method which equates to an uncertainty of 56.5m when calculating distance. When 4 PPS signals are averaged to determine the instantaneous oscillator frequency then the uncertainty increases to  $\pm 4$  clock cycles. This results in a very small tolerance in the length of clock cycle calculation (less than 2fs) and can be disregarded in the distance calculation uncertainty. The tolerance of the GNSS PPS signal is 30ns (distance calculation uncertainty of 9m) and the tolerance of the GNSS position is 10m which results in a total distance uncertainty of 19ms for the GNSS modules. The largest uncertainty is from the difference in processing time between two LoRa modules to process the same received radio packet and make it available to the Arduino module which was determined to be 5 $\mu$ s (with 92.6% confidence) which equates to an error in the distance calculation of 1.5km.

The results of investigating the error in timing measurements indicated that each ground station will have a total distance calculation uncertainty of 75m for the GNSS and Arduino module with the LoRa module introducing a 1.5km uncertainty for the difference in the LoRa processing times. The total expected error in distance calculations between two ground receiving stations is 1.575kms which equals a time measuring uncertainty of 5.25 $\mu$ s. The verification testing carried out between two ground stations shows the total uncertainty in timing measurements to be between 1.5-1.7ms which equates to a distance calculation error of 450-510kms.

The initial investigation for the large uncertainty in timing measurement tested the two external interrupts (the GNSS PPS signal and LoRa module *RXDone* interrupt) with the PPS signal found to be within tolerance while the LoRa module was found to have a processing time difference of up to 30 $\mu$ s (most were between 5-15 $\mu$ s) which

is  $25\mu s$  greater than the values found during previous testing. This increase of the LoRa processing time is equal to a distance calculation error of 9km which is much less than the error found in the verification testing. The large spike in clock counting error found during testing of the clock cycle counting algorithm was never greater than 190 cycles which is equal to an uncertainty of  $12\mu s$  or a distance calculation error of 3.5km which is much less than the final testing error but gives an indication to the cause of the error. The determined cause of the spike in the clock counting test was found to be due to the time taken (and number clock cycles) for the program to finish executing the current set of instructions before it enters the interrupt routines. The increased complexity of the ground receiver software and the two external interrupts used in the program taking precedence over the clock counting ISR causes a large increase in the number of clock cycles occurring before the Timer1 overflow count updates. To determine the reason for the large increase in error in counting clock cycles (1.5ms is equal to 24,000 clock cycles), an investigation into the method of implementing and calculating the clock cycles between events for the ground receiver station will be required with a focus on using timer capture modes for the ISR. This investigation to reduce the final error in timing measurement will be required to be carried out before the data acquired by the ground receiving station can be used reliably in calculating the estimated position of the satellite.

## VI. Conclusions

The aim of this project was to produce an initial prototype design for a self-contained and independent radio beacon that can transfer satellite identification and telemetry data from a small satellite in a low earth orbit to a ground receiving station. The project was extended to include a tracking function which uses the radio beacon signals time of arrival at multiple dispersed ground stations to estimate the position of the satellite using the TDOA trilateration calculation technique. The design of the satellite beacon prototype is presented on page 7 in Figure 5 and was the first step of the process with a focus on ensuring the beacon is self-sufficient, independent and the size of the beacon is minimized. The initial beacon prototype was designed, tested and verified to be capable of: 1) operating without an input from other satellite systems, 2) collecting telemetry data from other satellite systems, 3) sustaining normal beacon operation using a single silicon solar panel, and 4) executing a command transmitted from a ground station. The next step in the design process was to verify the LoRa radio modules could support the transmission of data over a communications link from a satellite in low earth orbit to a ground receiving station. The calculation of the communications link budget was supported by the tests conducted which show that the reliable transmission of data can be supported by the LoRa radio modules for slant range distances up to 2000kms. The data transmitted by the satellite beacon system contains a 16-Bit identification address in four identifiable, sequential radio packets followed by the satellite telemetry information in a separate radio packet. The information was split into separate radio packets to ensure that the satellite identification is received due to the high PER of the LoRa module. The final step was the design of the ground receiving station which is shown on page 10 in Figure 3, with the initial ground tests showing that it is capable of receiving the satellite identification and telemetry data. The testing of the satellites radio beacon signals time of arrival between two ground stations indicated an uncertainty of  $5\mu s$  in the time measurement which equates to a distance calculation error of 1.5kms. The final verification testing carried out using the developed ground station software demonstrated that the time of arrival data collected by two ground stations has a measured time uncertainty of 1.7ms which is a distance calculation error of 510kms. The amount of uncertainty in calculating the distance of the satellite from the ground receiving station is too large for this application with further work required to produce a more accurate time measurement technique in the ground stations. The concept of the small satellite radio beacon system has been proven for satellite identification, telemetry and control capabilities but the ground receiving station requires additional work to develop a reliable and accurate tracking function.

## VII. Recommendations

To conclude the initial design of the ground receiving station and ensure the satellite beacon system has a tracking capability requires further study to reduce the level of uncertainty in the received signals time of arrival between two ground stations. The uncertainty in time measurement by the ground station should be the focus of the investigation with the most likely contribution to uncertainty being found in the current ground station software program. The reduction of the uncertainty in time measurement to an acceptable level will allow for the development and verification of tracking the satellites position by using the time difference of arrival at multiple ground stations. This will then lead to the developing a TDOA algorithm that calculates the estimated position of the satellite within the determined uncertainties using the collected TDOA values and the latitude and longitude position of the ground stations. If the tracking capability is verified the development can continue to design a computer server system to transfer the required data between globally dispersed locations and develop a graphical user interface for displaying the position of the satellite.

The next step in the development of the satellite radio beacon is further investigation into the electrical components utilised in the prototype to ensure it is capable of operation in a space environment. Once the electrical components have been selected then a printed circuit board (PCB) design for the radio beacon can be developed to enable the execution of the recommended ground testing before space operations<sup>29</sup>.

### **Acknowledgements**

Firstly, I would like to say thank you to my wife, Christine and children for their unquestionable love and support throughout my studies and particularly through this research project. I would not have been able to complete my study without the time, support and understanding that they have provided through some demanding times.

I extend my sincerest gratitude to my thesis supervisors, Dr Edwin Peters and Dr Craig Benson, for their guidance and support through the project learning process. The brainstorming provided by Dr Craig Benson help to creatively shape the concept, which could only be transformed to reality with the time, dedication, knowledge and experience provided by Dr Edwin Peters.

This document would not exist without guidance and advice from Bianca Mister-Colmenares of the UNINSW Canberra ALL unit. I want to recognize the time, effort, expert knowledge and literary artistry she has provided to improve the presentation of this document and develop my writing skills.

I wish to extend my appreciation to Darryl Budarick of the Technology Support Group at UNINSW Canberra for the guidance, support and feedback in developing the hardware for this project.

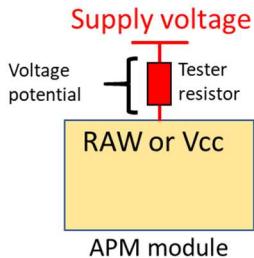
The expert guidance from Lisa Conti Phillipps of the UNINSW Canberra Library has taught me how to perform effective research and saved me many hours of unnecessary searching for which I am thankful.

## References

- <sup>1</sup> DELPOZZO, S., WILLIAMS, C. & DONCASTER, B. 2018. 2019 Nano/Microsatellite Market Forecast. 9.
- <sup>2</sup> WEKERLE, T., PESSOA FILHO, J. B., COSTA, L. E. V. L. D. & TRABASSO, L. G. 2017. Status and Trends of Smallsats and Their Launch Vehicles An Up-to-date Review. *Journal of Aerospace Technology and Management*, 9, 269-286.
- <sup>3</sup> VENTURINI, C. C. 2017. Improving mission success of CubeSats.
- <sup>4</sup> KELSO, T. S. 2017. Challenges identifying newly launched objects. *Proceedings of the International Astronautical Congress, IAC*, 6, 3898-3903.
- <sup>5</sup> PHAN, S. 2019. SRI International's CubeSat Identification Tag (CUBIT): System Architecture and Test Results from Two On-Orbit Demonstrations.
- <sup>6</sup> SVITEK, T. 2018. Passive RF Tag for Satellite Tracking. Available: <https://static1.squarespace.com/static/5c54e307fd67934e24b27846/t/5ca42cece5f0302a31f91c/1554263277115/RF+tag+white+paper+2018+02+01+public+release.pdf> (Accessed 07 oct 2019).
- <sup>7</sup> PALMER, D. M. & HOLMES, R. M. 2018. Extremely Low Resource Optical Identifier: A License Plate for Your Satellite. *Journal of spacecraft and rockets*, 55, 1014-1023.
- <sup>8</sup> CIALONE, G., MARZIOLI, P., MASILLO, S., GIANFERMO, A., FREZZA, L., PELLEGRINO, A., PIERGENTILI, F. & SANTONI, F. LEDSAT: A LED-Based CubeSat for optical orbit determination methodologies improvement. 5th IEEE International Workshop on Metrology for AeroSpace, MetroAeroSpace 2018 - Proceedings, 2018. 456-461.
- <sup>9</sup> LINHART, R. 2013. Safety radio beacon for the PilsenCUBE satellite. *Applied Electronics, AE, International Conference on*.
- <sup>10</sup> RIVERS, T. D., HESKETT, J. & VILLA, M. 2015. RILDOS: A Beaconing Standard for Small Satellite Identification and Situational Awareness.
- <sup>11</sup> HUMAD, Y. A. I., TAGELSIR, A. & DAFFALLA, M. M. 2017. Design and implementation of communication subsystem for ISRASAT1 Cube Satellite. 1-4.
- <sup>12</sup> POPESCU, O. 2017. Power Budgets for CubeSat Radios to Support Ground Communications and Inter-Satellite Links. *Ieee Access*, 5, 12618-12625.
- <sup>13</sup> Atmel corporation, “8-bit AVR® Microcontroller with 4/8/16/32K bytes In-System Programmable Flash”, ATMEGA328P datasheet, 2009
- <sup>14</sup> HopeRF Electronics, “RFM95/96/97/98(W) - Low Power Long Range Transceiver Module”, RFM95/96/97/98(W) datasheet, 2019
- <sup>15</sup> GitHub. 2020. *Atttxx/Rfm9x Breakout Board*. [online] Available at: [https://github.com/atttxx/rfm9x\\_breakout\\_board](https://github.com/atttxx/rfm9x_breakout_board) (Accessed 16 Feb 2020)
- <sup>16</sup> Seeed Studio, “0.5W Solar Panel 55x70”, 0.5W solar panel description, [online] Available at: <https://www.seeedstudio.com/0-5W-Solar-Panel-55x70.html> (Accessed 10 May 2020)
- <sup>17</sup> 2020. *Arduino IDE*. Arduino LLC. Available at: <https://www.arduino.cc/en/Main/Software> (Accessed: 13 Jan 2020).
- <sup>18</sup> Airspayce.com. 2020. *Radiohead: RH RF95 Class Reference*. [online] Available at: [https://www.airspayce.com/mikem/arduino/RadioHead/classRH\\_RF95.html](https://www.airspayce.com/mikem/arduino/RadioHead/classRH_RF95.html) (Accessed 09 Feb 2020).
- <sup>19</sup> RTL-SDR, “RTL-SDR Blog V3 Datasheet”, RTL-SDR datasheet, 2019
- <sup>20</sup> Csete, A., 2020. *Gqrx SDR*. Alexandru Csete OZ9AEC.
- <sup>21</sup> Texas Instruments, “LM1086 1.5-A Low Dropout Positive Voltage Regulators”, LM1086 datasheet, June 2000 [Revised Apr 2015]
- <sup>22</sup> TSC Electronics Ltd, “TS2940 1A Ultra Low Dropout Fixed Positive Voltage Regulator”, TS2940CZ datasheet, 2003 [Revised 2012]
- <sup>23</sup> Microchip Technology Inc, “MC5205 150 mA Low-Noise LDO Regulator”, MC5205 datasheet, 2017
- <sup>24</sup> Texas Instruments, “LM3671/-Q1 2-MHz, 600-mA Step-Down DC-DC Converter”, LM3671 datasheet, Nov 2004 [Revised May 2016]
- <sup>25</sup> INITIATIVE, N. C. L. 2017. NASA CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers. *NASA CubeSat Launch Initiative*.
- <sup>26</sup> AUSTRALIAN COMMUNICATIONS AND MEDIA AUTHORITY (2020). *Radiocommunications Licence Conditions (Amateur Licence) Determination 2015*. Available at: <https://libguides.ioe.ac.uk/c.php?g=482485&p=3299865> (Accessed: 14 May 2020).
- <sup>27</sup> 2019, LoRa Modem Calculator Tool, Semtech Corporation. [online] Available at: [https://semtech.my.salesforce.com/sfc/p/#E0000000JeIG/a/2R000000HUhK/6T9Vdb3\\_ldnElA8drIbPYjslwBbhIWUXej8ZMXtZXOM](https://semtech.my.salesforce.com/sfc/p/#E0000000JeIG/a/2R000000HUhK/6T9Vdb3_ldnElA8drIbPYjslwBbhIWUXej8ZMXtZXOM) (Accessed 29 Oct 2019)
- <sup>28</sup> U-Blox, “NEO-7 u-blox 7 GNSS modules”, NEO-7M datasheet, Nov 2014
- <sup>29</sup> INITIATIVE, N. C. L. 2017. NASA CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers. *NASA CubeSat Launch Initiative*.

## Testing Method

The Arduino Pro Mini (APM) module selected for testing contains a built-in voltage regulator which requires two separate tests to be carried out. The first will test the electrical current consumption of the APM module using the built-in regulator and the second will test the APM consumption for bypassing the in-built regulator. To test the power consumption in both tests, the voltage difference over a known valued resistor placed in series between the power supply and APM module will be measured to determine the amount of current drawn by the APM module as shown below in . This method of calculating the current will be referred to throughout this document as using a ‘tester resistor’.

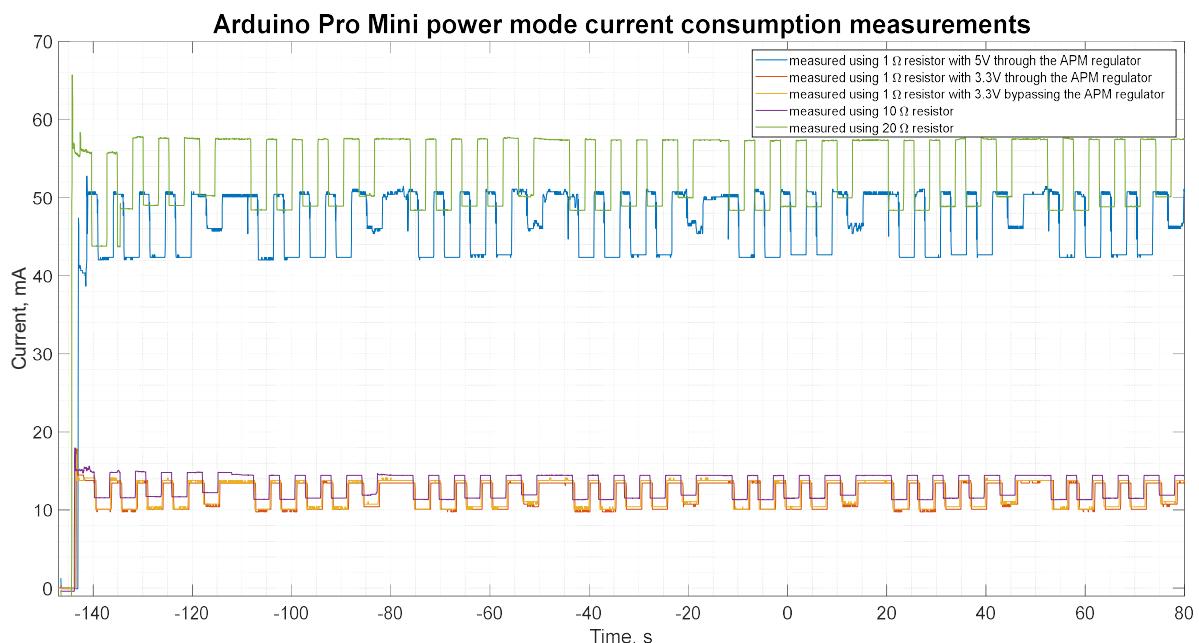


*Figure A1 - Tester resistor configuration for determining current consumption*

Firstly, a 3.3V and 5V power supply will be applied to the RAW pin of the APM module which feeds directly into the built-in fixed Low Dropout (LDO) voltage regulator (p/n MIC5205). This will measure the change in current consumed by the voltage regulator when the input voltage is the same as the output voltage and when there is a large difference between the input and output voltage. The next step will apply a 3.3V regulated voltage to the Vcc pin which bypasses the in-built regulation to check the change in current consumption.

Secondly, the ATMEGA328P processor used in the APM module is capable of being operating in different modes which consume different amounts of electrical current. The *lowpower.h* file developed by Rocketscream (<https://github.com/rocketscream/Low-Power>) is used to change the APM power operating mode. The test is carried out by cycling the APM module through the six modes of operation (*power on/normal mode, powerDown, powerSave, powerStandby, powerExtStandby and idle*) followed by a 5 second delay before the test cycle is repeated multiple times with each measurement averaged to increase the accuracy of the measurements.

## Results



*Figure A2 - Arduino Pro Mini current consumption measurements for each power mode operation*

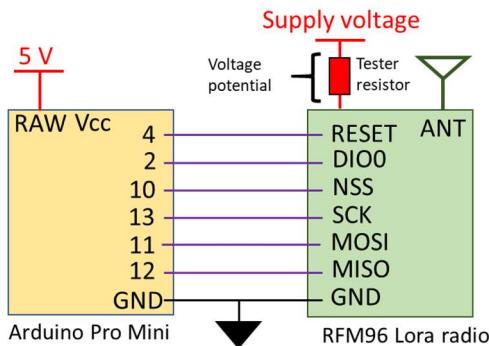
Appendix A  
Arduino Pro Mini processor testing method and results

*Table A1 - Arduino Pro Mini current consumption measurements tabulated results and averages*

Arduino Pro Mini power mode current consumption measurements and averages																	
1 Ohm Resistor measurement with Unregulated voltage (5V) on RAW pin							10 Ohm Resistor measurement										
Cycle number	1	2	3	4	5	6	7	Avg	Cycle number	1	2	3	4	5	6	7	Avg
power on current (mA)	50.76	50.43	51.10	51.09	50.09	50.76	50.43	<b>50.67</b>	power on current (mA)	15.00	14.60	14.60	14.60	14.60	14.60	14.60	<b>14.66</b>
powerDown current (mA)	42.01	42.01	42.35	42.35	42.35	42.35	42.35	<b>42.25</b>	powerDown current (mA)	11.60	11.40	11.40	11.40	11.40	11.40	11.40	<b>11.43</b>
powerSave current (mA)	42.01	42.35	42.35	42.35	42.35	42.35	42.35	<b>42.30</b>	powerSave current (mA)	11.60	11.40	11.40	11.40	11.40	11.40	11.40	<b>11.43</b>
powerStandby current (mA)	42.35	42.35	42.69	42.69	42.69	42.69	42.69	<b>42.59</b>	powerStandby current (mA)	11.70	11.50	11.50	11.50	11.50	11.50	11.50	<b>11.53</b>
powerExtStandby (mA)	42.35	42.35	42.69	42.69	42.69	42.69	42.69	<b>42.54</b>	powerExtStandby (mA)	11.70	11.50	11.50	11.50	11.50	11.50	11.50	<b>11.53</b>
idle (mA)	46.05	45.72	46.05	46.05	46.39	46.39	46.39	<b>46.15</b>	idle (mA)	12.20	11.90	11.90	11.90	11.90	11.90	11.90	<b>11.94</b>
1 Ohm Resistor measurement with Regulated voltage (3.3V) on RAW pin							20 Ohm Resistor measurement										
Cycle number	1	2	3	4	5	6	7	Avg	Cycle number	1	2	3	4	5	6	7	Avg
power on current (mA)	13.78	13.78	13.78	13.78	13.78	13.78	13.78	<b>13.78</b>	power on current (mA)	14.40	14.40	14.40	14.40	14.40	14.40	14.40	<b>14.40</b>
powerDown current (mA)	10.09	10.09	9.76	10.09	10.09	9.76	10.09	<b>10.00</b>	powerDown current (mA)	11.00	12.10	12.10	12.10	12.10	12.10	12.10	<b>11.94</b>
powerSave current (mA)	10.09	10.09	9.76	10.09	10.09	9.76	10.09	<b>10.00</b>	powerSave current (mA)	12.10	12.10	12.10	12.10	12.10	12.10	12.10	<b>12.10</b>
powerStandby current (mA)	10.41	10.41	10.41	10.41	10.41	10.41	10.41	<b>10.41</b>	powerStandby current (mA)	12.20	12.20	12.20	12.20	12.20	12.20	12.20	<b>12.20</b>
powerExtStandby (mA)	10.41	10.41	10.41	10.41	10.41	10.41	10.41	<b>10.41</b>	powerExtStandby (mA)	12.20	12.20	12.20	12.20	12.20	12.20	12.20	<b>12.20</b>
idle (mA)	11.09	11.09	10.77	11.09	11.09	10.43	10.47	<b>10.86</b>	idle (mA)	12.50	12.50	12.50	12.50	12.50	12.50	12.50	<b>12.50</b>
1 Ohm Resistor measurement with Regulated voltage (3.3V) on Vcc pin							20 Ohm Resistor measurement										
Cycle number	1	2	3	4	5	6	7	Avg	Cycle number	1	2	3	4	5	6	7	Avg
power on current (mA)	13.46	13.46	13.46	13.46	13.46	13.80	13.46	<b>13.51</b>	power on current (mA)	14.40	14.40	14.40	14.40	14.40	14.40	14.40	<b>14.40</b>
powerDown current (mA)	10.09	10.09	10.09	9.76	10.09	9.76	10.09	<b>10.00</b>	powerDown current (mA)	11.00	12.10	12.10	12.10	12.10	12.10	12.10	<b>11.94</b>
powerSave current (mA)	10.09	10.09	10.09	9.76	10.09	9.76	10.09	<b>10.00</b>	powerSave current (mA)	12.10	12.10	12.10	12.10	12.10	12.10	12.10	<b>12.10</b>
powerStandby current (mA)	10.09	10.09	10.09	10.09	10.09	10.09	10.09	<b>10.09</b>	powerStandby current (mA)	12.20	12.20	12.20	12.20	12.20	12.20	12.20	<b>12.20</b>
powerExtStandby (mA)	10.09	10.09	10.09	10.09	10.09	10.09	10.09	<b>10.09</b>	powerExtStandby (mA)	12.20	12.20	12.20	12.20	12.20	12.20	12.20	<b>12.20</b>
idle (mA)	10.43	10.43	10.77	10.77	10.43	10.43	10.77	<b>10.58</b>	idle (mA)	12.50	12.50	12.50	12.50	12.50	12.50	12.50	<b>12.50</b>

## Testing Method

The testing configuration for all three RFM96 LoRa module tests are the same with the current consumption of the module being calculated using a tester resistor in series between the power supply and the 3.3V pin on the LoRa module. The RFM96 software is configured such that the radio operates at 437Mhz in the LoRa packet mode with the (0) default radio settings are used [Bandwidth = 125 kHz, Coding rate = 4/5, Spreading factor = 7 (128 chips/symbol) and Cyclic Redundancy Check (CRC) on]. It is noted that the (0) default radio settings is for medium range, medium data rate applications but it will allow the current consumption of the radio to be characterized with the final radio settings being determined by the testing of the communications link. The configuration of the APM processor and LoRa radio module used for all LoRa consumption testing is shown below in Figure B1.



*Figure B1- LoRa radio testing configuration and connections*

The first test will determine the current consumption of the LoRa radio module during each mode of radio operation (*sleep, receive, transmit and idle*) available using the RadioHead library developed by Airspayce (<https://www.airspayce.com/mikem/arduino/RadioHead/>). Each mode of radio operation will be activated in sequence during one testing cycle which is repeated several times with different values of tester resistor to find the average current consumption. When the radio is set to *transmit* mode there is no data being transmitted from the LoRa module with the idle consumption of the transmit mode being checked and not the active mode which will be checked in the next test.

The second test will check the difference in current consumption of the LoRa module transmitting 30 bytes of data when the transmit power is increased from 5dBm to 23dBm. The transmit power is increased in 1dBm increments over several transmit cycles using different values of tester resistors to determine the average current consumption of the radio module.

The last test will measure the transmission time when the size of the transmitted radio packet is decreased from 250 bytes to 5 bytes in 5 byte increments when using a variety of transmit powers (5, 10, 15 and 20dBm). The current consumption will be measured using different tester resistors to average the results which are compared against the LoRa modem calculator tool supplied by the chip manufacturer, Semtech.

## Appendix B RFM96 LoRa radio module testing method and results

### Results

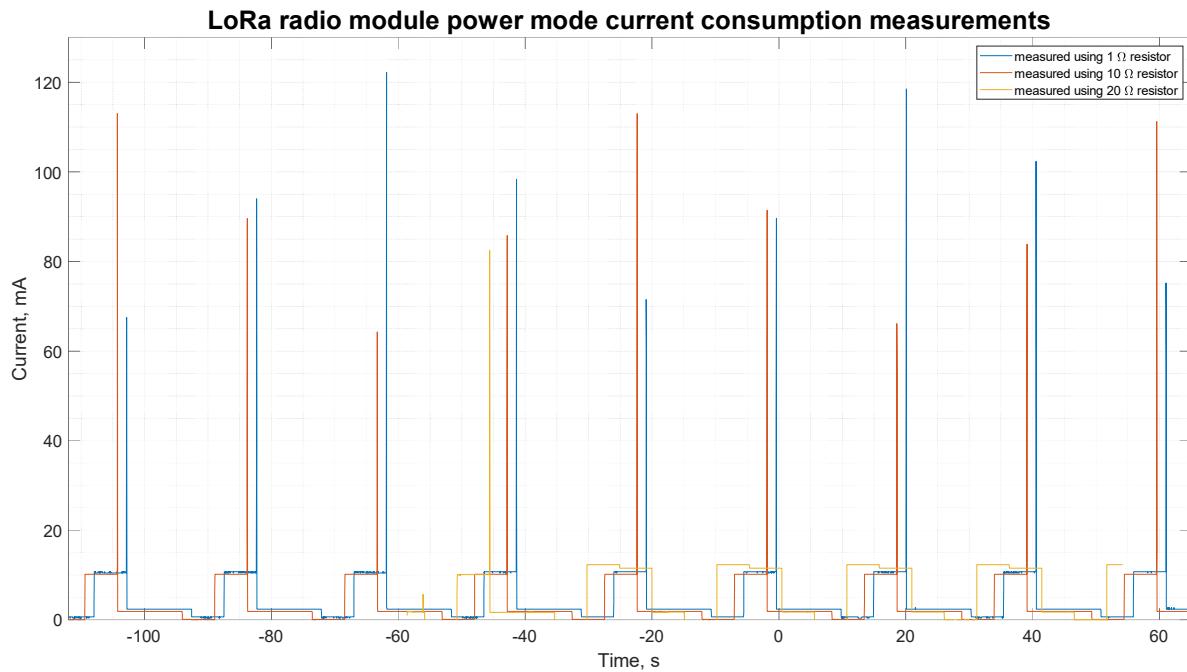
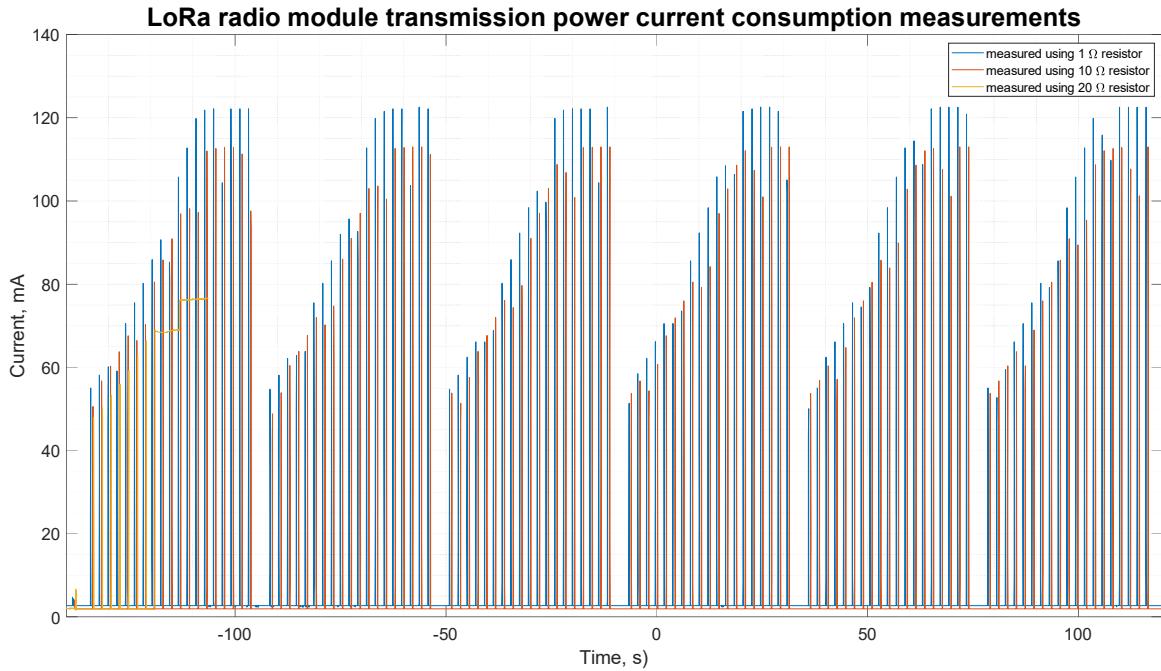


Figure B1 - RFM96 LoRa module current consumption measurements during each mode of operation

Table B1 - RFM96 LoRa radio module current consumption measurement for each mode of operation

RFM96 LoRa radio module power mode current consumption measurements										
Cycle number	1 Ohm Resistor measurement									
	1	2	3	4	5	6	7	8	9	AVG
Sleep current (A)	0.00070	0.00070	0.00070	0.00070	0.00070	0.00070	0.00070	0.00070	0.00070	<b>0.00070</b>
Receive current (A)	0.011	0.011	0.011	0.011	0.011	0.011	0.011	0.011	0.011	<b>0.011</b>
No data transmit current (A)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	<b>0.002</b>
Idle current (A)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	<b>0.002</b>
10 Ohm Resistor measurement										
Cycle number	1	2	3	4	5	6	7	8	9	AVG
Sleep current (A)	0.00036	0.00070	0.00070	0.00036	0.00070	0.00070	0.00070	0.00070	0.00070	<b>0.00062</b>
Receive current (A)	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0.010	<b>0.010</b>
No data transmit current (A)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	<b>0.002</b>
Idle current (A)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	<b>0.002</b>
20 Ohm Resistor measurement										
Cycle number	1	2	3	4	5	6	7	8	9	AVG
Sleep current (A)				-0.00032	-0.00015	-0.00015	-0.00015	0.00012	0.00012	<b>-0.00009</b>
Receive current (A)				0.010	0.012	0.012	0.012	0.012	0.012	<b>0.012</b>
No data transmit current (A)				0.002	0.020	0.001	0.001	0.001	0.001	<b>0.004</b>
Idle current (A)				0.002	0.002	0.002	0.002	0.002	0.002	<b>0.002</b>

Appendix B  
RFM96 LoRa radio module testing method and results



*Figure B2 - RFM96 LoRa module current consumption measurements for each transmit power available*

*Table B2 - RFM96 LoRa radio module current consumption measurement for each transmit power available*

RFM96 LoRa radio module transmit power current consumption measurements																				
Cycle no.	1 Ohm Resistor - Current (Amps)							10 Ohm Resistor - Current (Amps)							20 Ohm Resistor - Current (Amps)					
	1	2	3	4	5	6	Avg	1	2	3	4	5	6	Avg	1	2	3	Avg		
5dB	0.055	0.055	0.055	0.051	0.050	0.055	<b>0.054</b>	5dB	0.050	0.049	0.054	0.054	0.054	<b>0.053</b>	5dB	0.048	0.048	0.048	<b>0.048</b>	
6dB	0.058	0.058	0.058	0.058	0.055	0.053	<b>0.057</b>	6dB	0.057	0.054	0.051	0.057	0.057	<b>0.056</b>	6dB	0.050	0.050	0.051	<b>0.050</b>	
7dB	0.050	0.062	0.062	0.062	0.059	<b>0.060</b>	7dB	0.060	0.060	0.057	0.054	0.060	0.060	<b>0.059</b>	7dB	0.053	0.053	0.054	<b>0.053</b>	
8dB	0.059	0.063	0.066	0.066	0.066	<b>0.064</b>	8dB	0.064	0.064	0.064	0.061	0.057	0.064	<b>0.062</b>	8dB	0.056	0.056	0.056	<b>0.056</b>	
9dB	0.070	0.064	0.066	0.071	0.071	<b>0.069</b>	9dB	0.066	0.068	0.068	0.068	0.065	0.060	<b>0.066</b>	9dB	0.059	0.059	0.059	<b>0.059</b>	
10dB	0.076	0.076	0.069	0.071	0.076	<b>0.076</b>	10dB	0.068	0.072	0.071	0.072	0.072	0.069	<b>0.071</b>	10dB	0.063	0.063	0.064	<b>0.063</b>	
11dB	0.080	0.080	0.074	0.075	0.080	<b>0.078</b>	11dB	0.070	0.070	0.076	0.076	0.076	0.076	<b>0.074</b>	11dB	0.066	0.066	0.067	<b>0.066</b>	
12dB	0.086	0.086	0.086	0.086	0.079	0.080	<b>0.084</b>	12dB	0.080	0.075	0.074	0.080	0.080	0.080	<b>0.078</b>	12dB	0.069			<b>0.069</b>
13dB	0.090	0.092	0.092	0.092	0.092	0.086	<b>0.091</b>	13dB	0.086	0.086	0.080	0.080	0.086	0.086	<b>0.084</b>	13dB				<b>0.000</b>
14dB	0.082	0.096	0.098	0.098	0.098	<b>0.095</b>	14dB	0.091	0.091	0.091	0.084	0.084	0.091	<b>0.089</b>	14dB				<b>0.000</b>	
15dB	0.105	0.093	0.102	0.106	0.106	0.106	<b>0.103</b>	15dB	0.097	0.097	0.097	0.097	0.090	0.089	<b>0.095</b>	15dB				<b>0.000</b>
16dB	0.113	0.113	0.100	0.108	0.113	0.113	<b>0.110</b>	16dB	0.098	0.103	0.103	0.103	0.103	0.095	<b>0.101</b>	16dB				<b>0.000</b>
17dB	0.119	0.120	0.120	0.106	0.114	0.120	<b>0.117</b>	17dB	0.097	0.104	0.109	0.109	0.109	0.109	<b>0.106</b>	17dB				<b>0.000</b>
18dB	0.121	0.122	0.122	0.122	0.108	0.116	<b>0.119</b>	18dB	0.112	0.100	0.107	0.112	0.112	0.112	<b>0.109</b>	18dB				<b>0.000</b>
19dB	0.122	0.122	0.122	0.122	0.122	0.110	<b>0.120</b>	19dB	0.112	0.113	0.101	0.107	0.113	0.113	<b>0.110</b>	19dB				<b>0.000</b>
20dB	0.104	0.122	0.122	0.123	0.123	<b>0.123</b>	20dB	0.112	0.113	0.113	0.101	0.108	0.113	<b>0.110</b>	20dB				<b>0.000</b>	
21dB	0.122	0.104	0.123	0.123	0.123	<b>0.120</b>	21dB	0.112	0.113	0.113	0.113	0.102	0.108	<b>0.110</b>	21dB				<b>0.000</b>	
22dB	0.122	0.123	0.104	0.122	0.123	0.123	<b>0.120</b>	22dB	0.111	0.113	0.113	0.113	0.101	<b>0.111</b>	22dB				<b>0.000</b>	
23dB	0.122	0.122	0.123	0.105	0.121	0.123	<b>0.119</b>	23dB	0.098	0.111	0.113	0.113	0.113	<b>0.110</b>	23dB				<b>0.000</b>	

Appendix B  
RFM96 LoRa radio module testing method and results

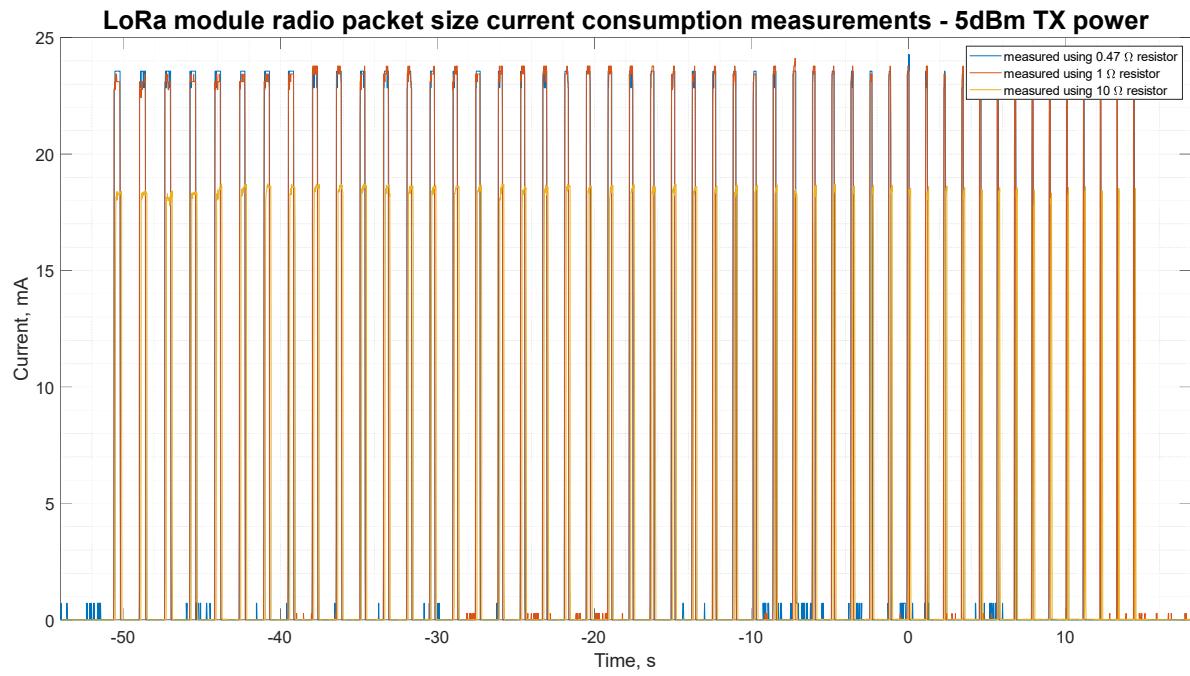


Figure B3 - RFM96 LoRa radio module transmission time for a 5-byte incremented radio packet with 5dBm Tx power

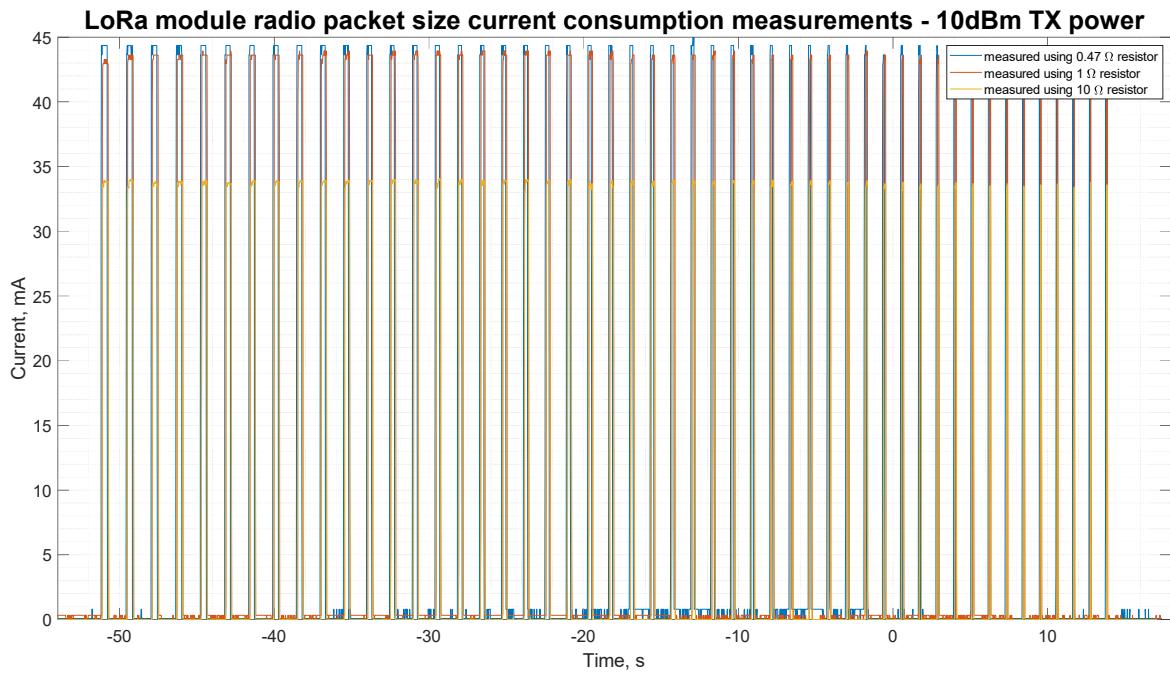


Figure B4 - RFM96 LoRa radio module transmission time for a 5-byte incremented radio packet with 10dBm Tx power

Appendix B  
RFM96 LoRa radio module testing method and results

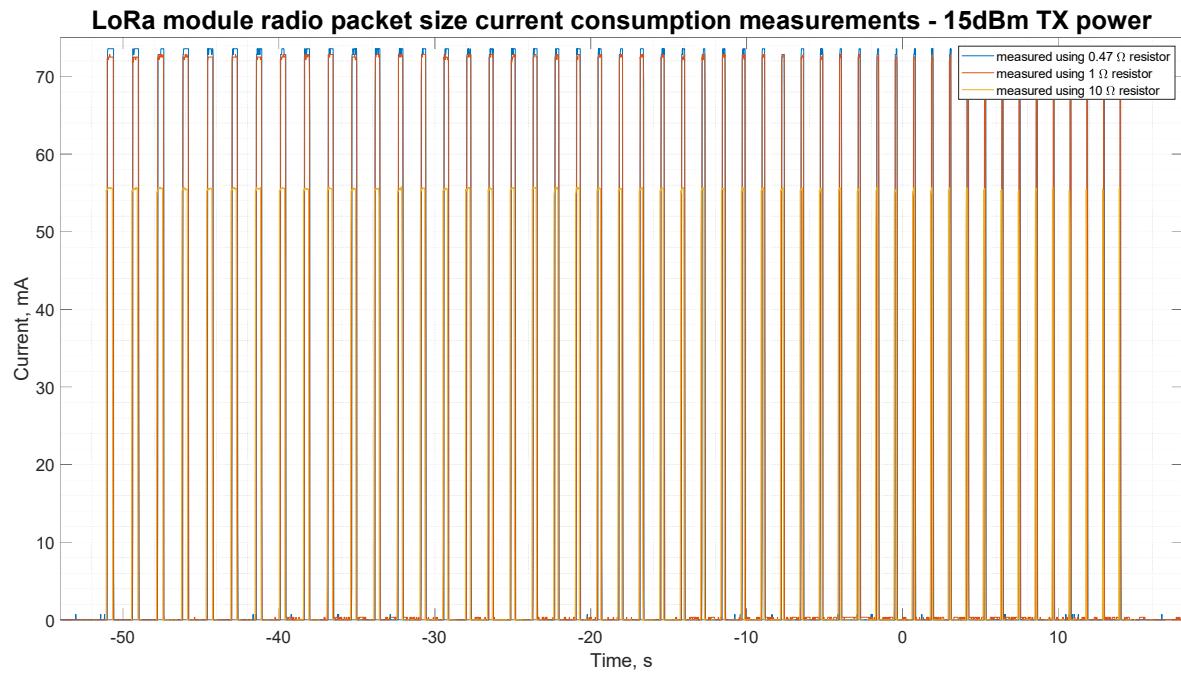


Figure B5 - RFM96 LoRa radio module transmission time for a 5-byte incremented radio packet with 15dBm Tx power

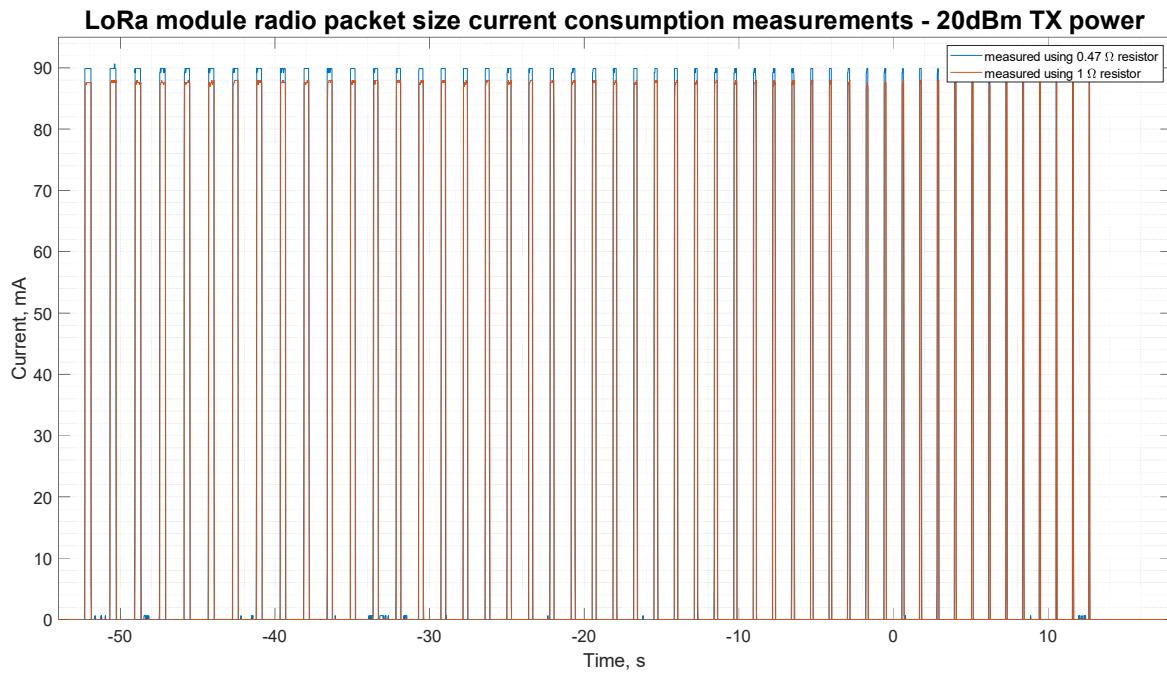


Figure B6 - RFM96 LoRa radio module transmission time for a 5-byte incremented radio packet with 20dBm Tx power

Appendix B  
RFM96 LoRa radio module testing method and results

*Table B3 - RFM96 LoRa radio module transmission time measurements for varying transmit packet size*

RFM 96 LoRa Radio transmission time measurements							
Byte Size	TX time (Sec)	Byte Size	TX time (Sec)	Byte Size	TX time (Sec)	Byte Size	TX time (Sec)
250	0.38	185	0.28	120	0.18	55	0.10
245	0.38	180	0.28	115	0.18	50	0.10
240	0.37	175	0.27	110	0.18	45	0.10
235	0.35	170	0.26	105	0.17	40	0.09
230	0.36	165	0.26	100	0.16	35	0.08
225	0.35	160	0.25	95	0.15	30	0.08
220	0.35	155	0.23	90	0.15	25	0.07
215	0.32	150	0.23	85	0.14	20	0.07
210	0.32	145	0.22	80	0.14	15	0.06
205	0.31	140	0.22	75	0.13	10	0.05
200	0.31	135	0.22	70	0.13	5	0.05
195	0.30	130	0.22	65	0.12		
190	0.29	125	0.20	60	0.11		

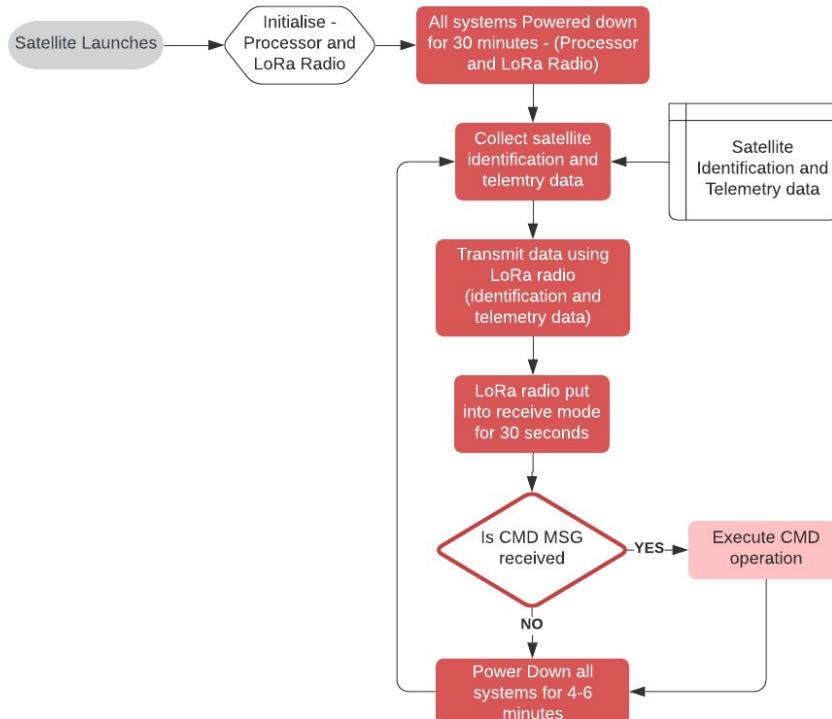
## Appendix C

### Satellite radio beacon software flow chart and design considerations

The major considerations when designing the software program are the requirement for no radio transmissions to occur for 30 minutes after launch, an inclusion of a period in the software cycle where the radio beacon can receive data and a variable period of time for when the radio beacon enters a power down state. There are several requirements that must be met when a satellite is released by a ride share launch provider with a major requirement being that no radio transmissions are to be carried out by the satellite for 30 minutes after the release from the launch vehicle. During the radio beacon software cycle, there must be a period where the LoRa radio can receive data from ground station. This will enable the radio beacon to receive a command from the ground to shut down the beacon radio transmissions which is a requirement of the International Telecommunication Union (ITU) when using the designated RF spectrum and also to receive a command that can be passed to another satellite system for emergency control. At the conclusion of the beacon software cycle, the amount of time that the system is powered down must vary to reduce risk of synchronisation occurring between multiple satellites. If the APM oscillator between 2 satellites were maintaining the same frequency and the radio beacon cycles where to align, then the transmissions from each satellite could interfere and cause a loss of transmission data from one of the satellites. To mitigate this risk of data loss, the power down time will be varied between each cycle to prevent synchronisation between satellite radio beacons occurring.

Testing was carried out to validate that the code developed in the Arduino IDE performs the necessary functions and processes defined in the software flow chart (refer Figure C1 below). This testing was carried out by running the radio beacon code through 100 software cycles to ensure that a 50-byte radio packet containing identification and telemetry data was sent from the radio beacon and received by a ground receiving station. The first 8 bytes of the radio packet contained the satellite identification ‘*TravSat1*’ and the remaining 42 bytes consisting of the telemetry data collected. A sample of satellite telemetry components was simulated such as measuring battery and solar panel voltage or checking light levels from several LEDs is performed during the collection of satellite telemetry data phase. The solar panel voltage is an incrementing counter from 0 to 100 to ensure that all packets are correctly received with any missed or dropped packets easily identified.

The software code was modified such that if a command was received from a ground station during the *receive* phase then a printed message was displayed on the Arduino serial monitor validating that the beacon can receive a command and execute a function.



*Figure C1 - Satellite radio beacon operational cycle software flow chart*

## Testing Method

The next investigation is reducing the current consumption of the beacon by utilising different electrical regulators for the electrical power system. The testing will be carried out by using a 5V, 1.5A wall power supply (the expected voltage for the solar panel is between 5-6V) being applied to a variety of regulators and running the satellite beacon through a shortened software cycle. The regulators that will be utilized for testing are:

1. MC5205 – APM module in-built Low-Noise LDO voltage regulator
2. LM1086 – LDO voltage positive regulator
3. LM3671 – Step-Down DC-DC converter (Buck converter)
4. TS2904CZ – Ultra LDO linear voltage regulator

The LoRa radio will use a TX power setting of 15dBm and will use the (2) RadioHead default settings [slow & long-range settings – BW = 31.25kHz, CR= 4/8, SF = 9 (512chips/symbol) and CRC on]. A  $1\Omega$  and  $2\Omega$  tester resistor will be placed in series with the power supply and regulator to measure the supply current with the testing configuration of the satellite beacon power regulator testing shown below in figure D1.

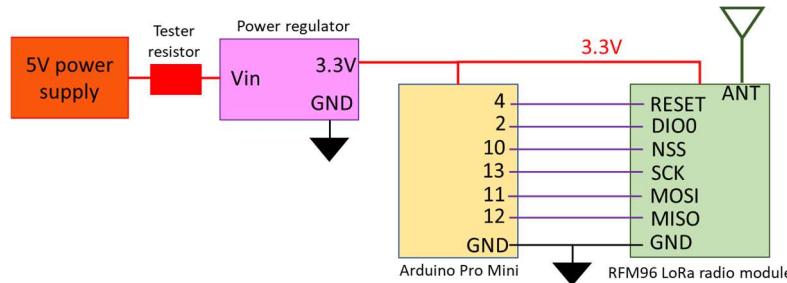


Figure D1 - Satellite beacon power regulator testing configuration

## Results

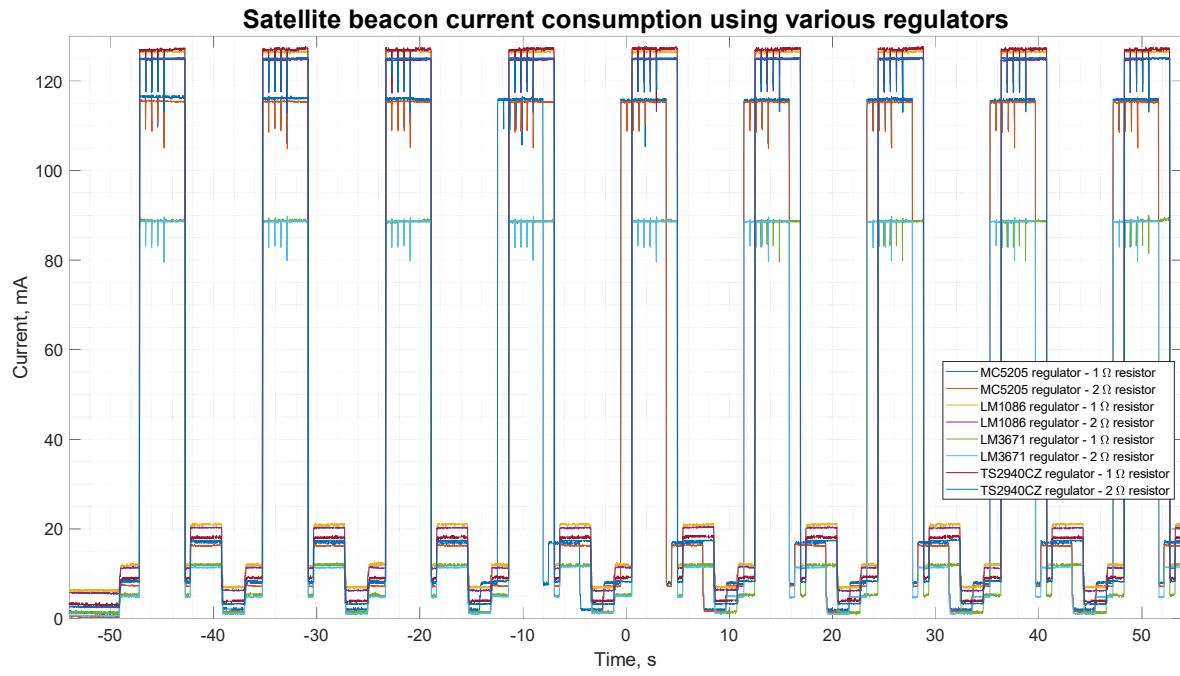


Figure D2 - Satellite beacon current consumption when utilising different voltage regulators

Appendix D  
Satellite beacon electrical power regulation testing method and results

*Table D1 - Satellite current consumption measurements and averages for different voltage regulators*

Satellite beacon current consumption using different regulators																			
MCS205 in-built regulator - 1 Ohm resistor								MCS205 in-built regulator - 2 Ohm resistor											
Cycle number	1	2	3	4	5	6	7	8	Avg	Cycle number	1	2	3	4	5	6	7	8	Avg
Launch (mA)	1.51								1.51	Launch (mA)	0.4								0.00
Collect Data (mA)	8.48	8.15	8.15	7.82	7.82	8.15	8.07	8.14	8.10	Collect Data (mA)	7.43	7.26	7.26	7.26	7.26	7.43	7.26	7.44	7.33
Transmit data (mA)	116.70	116.00	116.00	116.00	116.00	115.70	116.00	116.00	116.05	Transmit data (mA)	115.40	115.40	115.40	115.40	115.40	115.40	115.40	115.40	115.40
Receive mode (mA)	17.20	16.87	16.86	16.86	16.86	16.86	16.87	16.86	16.91	Receive mode (mA)	16.32	16.31	16.32	16.32	16.48	16.32	16.32	16.32	16.34
Idle mode (mA)	2.13	2.13	1.79	2.13	2.13	2.14	1.80	2.13	2.05	Idle mode (mA)	1.73	1.57	1.57	1.40	1.74	1.57	1.57	1.57	1.59
LM1086 external regulator - 1 Ohm resistor								LM1086 external regulator - 2 Ohm resistor											
Cycle number	1	2	3	4	5	6	7	8	Avg	Cycle number	1	2	3	4	5	6	7	8	Avg
Launch (mA)	6.48								6.48	Launch (mA)	5.75								5.75
Collect Data (mA)	11.85	12.17	11.84	12.18	12.18	12.18	12.18	12.18	12.10	Collect Data (mA)	11.46	11.29	11.29	11.46	11.46	11.61	11.46	11.29	11.42
Transmit data (mA)	126.70	126.70	126.70	126.70	126.70	126.70	126.40	126.40	126.66	Transmit data (mA)	124.90	124.90	124.80	124.80	124.80	124.90	124.90	124.80	124.85
Receive mode (mA)	20.89	20.88	20.89	20.89	20.89	20.88	20.88	20.89	20.89	Receive mode (mA)	20.34	20.34	20.17	20.34	20.34	20.18	20.17	20.28	20.28
Idle mode (mA)	6.81	7.15	7.15	7.15	6.82	6.82	7.15	7.15	7.02	Idle mode (mA)	6.43	6.26	6.25	6.26	6.43	6.26	6.43	6.26	6.32
LM3671 external regulator - 1 Ohm resistor								LM3671 external regulator - 2 Ohm resistor											
Cycle number	1	2	3	4	5	6	7	8	Avg	Cycle number	1	2	3	4	5	6	7	8	Avg
Launch (mA)	1.51								1.51	Launch (mA)	0.75								0.75
Collect Data (mA)	5.47	5.45	5.48	5.14	5.46	5.13	5.12	5.15	5.30	Collect Data (mA)	4.91	4.92	4.91	4.91	5.08	4.91	4.75	4.91	4.91
Transmit data (mA)	88.83	88.83	88.49	88.83	88.83	88.83	88.83	88.82	88.79	Transmit data (mA)	88.73	88.73	88.73	88.73	88.73	88.73	88.56	88.73	88.71
Receive mode (mA)	11.83	11.83	11.83	11.83	11.84	11.83	11.83	11.84	11.83	Receive mode (mA)	11.46	11.46	11.46	11.62	11.63	11.46	11.46	11.50	11.50
Idle mode (mA)	1.48	1.49	1.14	1.13	1.15	1.15	1.48	1.15	1.27	Idle mode (mA)	1.08	1.08	1.08	1.41	1.08	0.91	1.08	1.08	1.10
TS2940CZ external regulator - 1 Ohm resistor								TS2940CZ external regulator - 2 Ohm resistor											
Cycle number	1	2	3	4	5	6	7	8	Avg	Cycle number	1	2	3	4	5	6	7	8	Avg
Launch (mA)	3.47								3.47	Launch (mA)	2.58								2.58
Collect Data (mA)	9.16	9.17	9.17	9.17	9.50	9.49	9.16	9.16	9.25	Collect Data (mA)	7.42	7.43	7.26	7.26	7.43	7.43	7.26	7.29	7.35
Transmit data (mA)	127.40	127.40	127.00	127.40	127.40	127.40	127.40	127.40	127.35	Transmit data (mA)	125.10	125.30	125.10	124.90	125.10	125.10	125.30	125.30	125.15
Receive mode (mA)	17.87	18.21	18.20	18.21	18.21	18.20	18.21	18.21	18.17	Receive mode (mA)	16.31	18.32	16.48	16.32	16.32	16.32	16.32	16.32	16.59
Idle mode (mA)	3.81	3.80	5.08	4.14	4.13	4.14	3.81	3.80	4.09	Idle mode (mA)	3.24	3.25	3.24	3.25	3.42	3.41	3.24	3.25	3.29

## Testing Method

To determine the current drawn from the solar panels by the complete satellite radio beacon system (external LM3671 regulator, computer processor, LoRa radio module and attached components) during each phase of the software cycle, a tester resistor was placed between the solar panels and the breadboard positive power rail which supplies the power for all the other sub-systems components. The transmit power of the RFM96 LoRa radio module was set to 15dBm with the radio using the (2) RadioHead default settings (long range settings). The testing program used was a shortened version of the software cycle where the launch lasts for 5 seconds, the receive mode is 3 seconds and the idle/low power mode is 2 seconds.

The results from testing the current consumption of the satellite beacon provided evidence that the radio beacon was unable to be operated with one solar panel connected if the transmission power was greater than 10dBm as the current required for continuous operation was larger than the current being supplied by one solar panel. This prompted an investigation to find a solution where the transmit power can be increased while operating the beacon using one solar panel. This led to including five 2.2mF electrolytic capacitors in parallel with the solar panels which stores approximately 0.17 joules of energy that could be used during the transmit phase to stabilize the power system and provide enough energy for the transit current spike. The configuration of the satellite radio beacon used for the total power requirement and generation testing is presented below in Figure E1.

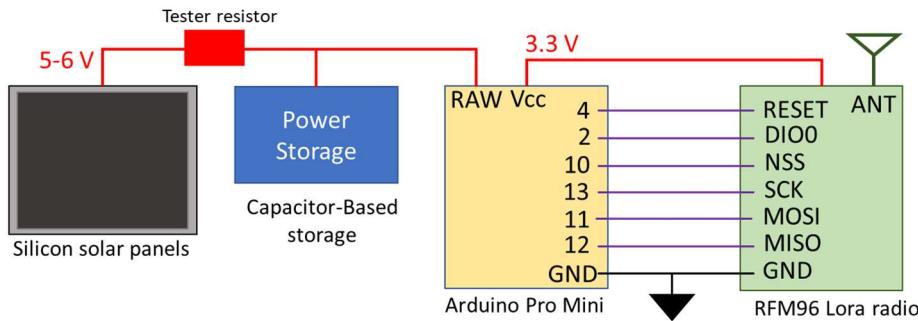


Figure E1 - Satellite radio beacon configuration for power requirement and generation testing

## Results

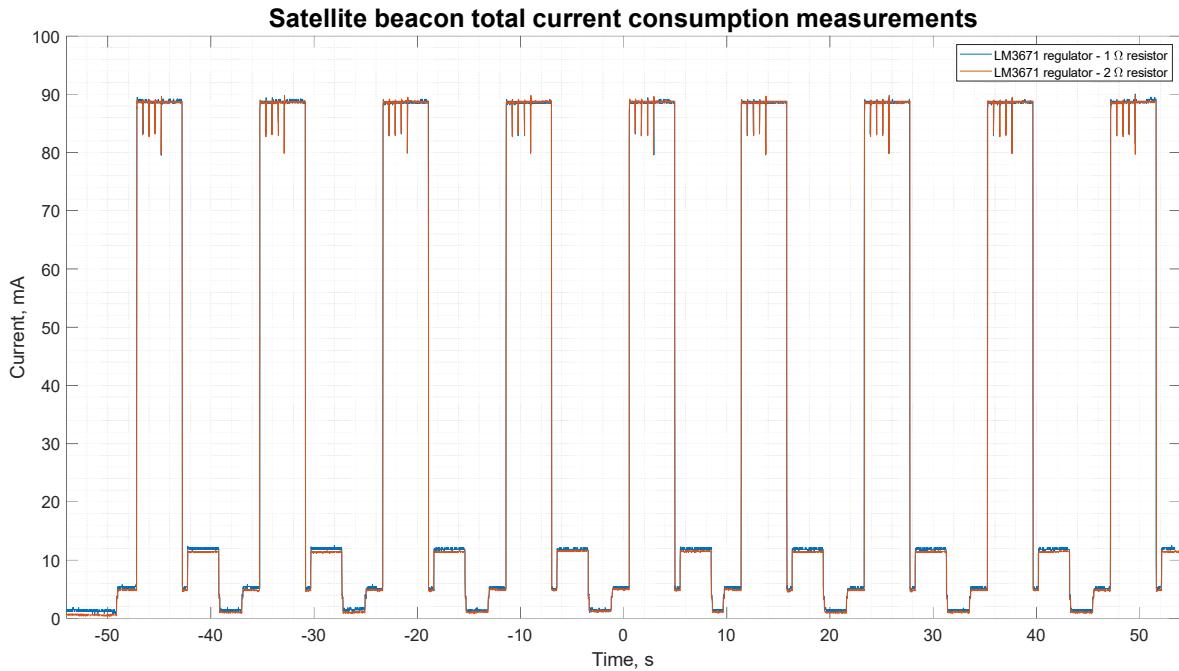


Figure E2 - Satellite beacon total current consumption during each phase of the software cycle

## Total beacon electrical power requirements and generation testing method and results

*Table E1 - Satellite beacon total current consumption measurements during each software cycle phase*

Satellite beacon total current measurements								
1Ω tester resistor								
Cycle number	1	2	3	4	5	6	7	Avg
Initialisation (mA)	<b>55.58</b>							<b>55.58</b>
Launch (mA)	1.50							<b>1.50</b>
Collect Data (mA)	5.44	5.44	5.14	5.46	5.46	5.46	5.46	<b>5.41</b>
Transmit data (mA)	88.83	88.88	88.70	88.56	88.56	88.83	88.88	<b>88.75</b>
Receive mode (mA)	12.17	12.17	11.83	11.83	11.84	11.84	11.87	<b>11.94</b>
low-power mode (mA)	1.48	1.48	1.48	1.48	1.48	1.48	1.48	<b>1.48</b>
2Ω tester resistor								
Cycle number	1	2	3	4	5	6	7	Avg
Initialisation (mA)	<b>55.4</b>							<b>55.4</b>
Launch (mA)	0.75							<b>0.75</b>
Collect Data (mA)	4.91	4.90	4.91	4.90	4.92	4.92	4.75	<b>4.89</b>
Transmit data (mA)	88.56	88.73	88.73	88.56	88.73	88.56	88.73	<b>88.66</b>
Receive mode (mA)	11.46	11.29	11.46	11.62	11.46	11.46	11.46	<b>9.82</b>
low-power mode (mA)	1.08	1.08	1.08	1.08	1.24	1.25	1.00	<b>1.12</b>

The testing of including 11mF of electrolytic capacitance to store energy for the transmit phase was conducted by running the beacon through the shortened software cycle with one solar panel connected while increasing the transmit power from 5dBm to 23dBm in 1dBm increments. The shortened beacon software cycle with the same settings as the previous test was used to test the addition of supporting capacitors in the power system. At each power level, the software cycle was performed 10 times with operation of the beacon being sustained for all transmit power levels when connected to a single solar panel and five 2.2mF capacitors. The weather conditions for the day were clear and sunny with the orientation of the solar panels were perpendicular to the sun. It is noted that solar panels, in general, can generate approximately 20% more power in a space environment as the sunlight does not have to penetrate the Earth's atmosphere. This will result in the beacon system having more power available when deployed in LEO with the excess providing a safety margin in the power generation system in lower irradiate conditions.

## Testing Method

The last investigation for the satellite beacon begins by determining if the inclusion of a super capacitor in parallel with the solar panel can sustain the transmit phase of the software cycle when using one solar panel. Secondly, a measurement of the voltage level and charge time of the super-capacitor storage system, consisting of five 1F capacitors, when connected to no load and 1 or 2 solar panels. The capacitor storage system is then connected to the Satellite radio beacon (with the solar panels are disconnected) to determine the length of time that software cycle can be run using only the energy stored in the capacitors. The full satellite beacon software cycle will be used without the 30-minute launch cycle, a transmit power of 15dBm and the (2) RadioHead default radio settings. The final step for the super-capacitor storage test will have it connected, with no electrical energy in the capacitors, to one solar panel and the satellite radio beacon which has the same radio settings as the previous test but will operate the full satellite radio beacon software cycle. The electrical potential of the capacitors will be monitored to measure the charging characteristics after a simulated launch and through the first cycles of the software program as well as to measure the time it takes for the system to contain enough energy to initialise post launch. The final configuration of the satellite radio beacon for testing is shown below in Figure F1.

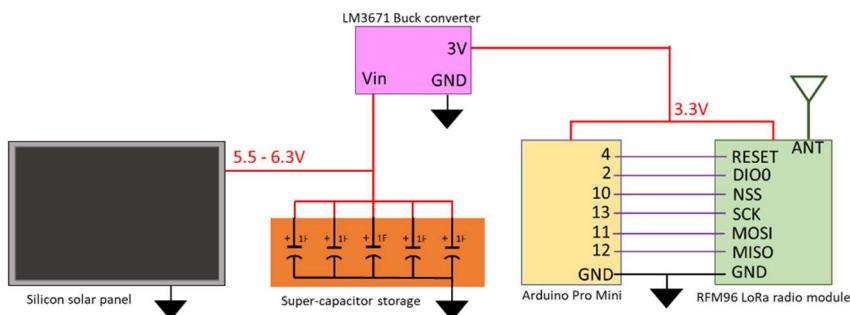


Figure F1 - Satellite radio beacon final configuration for testing

## Results

When a singular 5.5V, 1F super-capacitor replaced the 11mF electrolytic capacitor as the energy storage medium then the electrical storage system was able to support the radio transmit phase. The inclusion of the super-capacitor in the power system causes a delay for the radio beacon to start once power begin generating. When one solar panel is used to charge a single super-capacitor then it takes 1 minute for power to be applied to the system, and when the number of super-capacitors is increased to 5 then the time increases to 7 minutes.

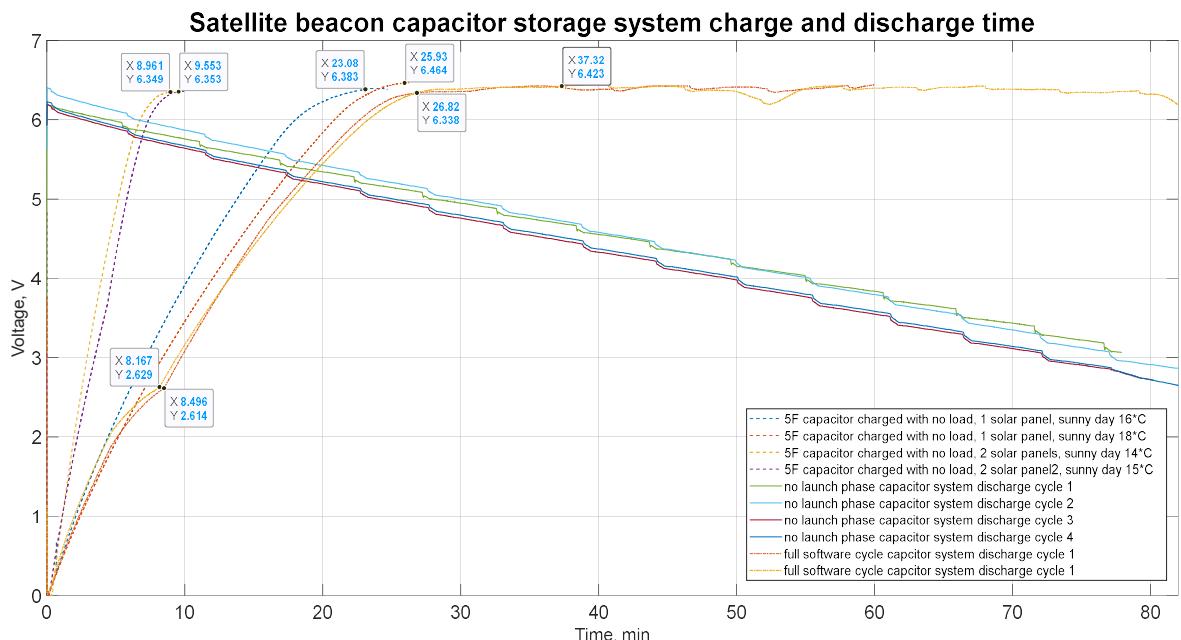


Figure F2 - Satellite beacon power storage system charge and discharge voltage with 5F super-capacitor system

Appendix F  
Satellite beacon electrical power storage testing method and results

The no-load charging time testing of the super-capacitors using one solar panel is detailed above in Figure F2, with the results showing that the average time taken for five 1F super-capacitors to charge to full capacity when no load is connected is 24 minutes and 30 seconds using one solar panel. If two solar panel are connected, then the charging time decreases to 9 minutes and 20 second.

The average voltage potential of the super-capacitor storage system after a charging cycle was 6.3V which equals 101 Joules of energy stored in the five capacitors. The average operating time before the super-capacitor could not support the beacon operation is 1 hour, 13 minutes and 43 seconds for one charge of the super-capacitor storage system. Discontinuities in the power supply would cause the Satellite radio beacon software to reset when the voltage potential of the super-capacitor storage system reduced below 3V which typically occurred during the transmit phase of the software cycle. The results show that there is an 8 minute and 20 seconds delay from when the solar panel first start to generate electrical power until there is enough energy to initialize the radio beacon hardware and software in which the voltage potential of the capacitors measures 2.6V. The super-capacitor storage system reaches it full electrical potential after 27 minutes of operation when the software cycle is approximately 19 minutes into the low power launch phase.

## Testing Method

1) The first set of tests that were performed to investigate the results of changing the radio parameters and test conditions of the communication link has on the number of packets that are dropped by the RFM96 LoRa radio module. The parameters for the radio and test conditions that will be adjusted are...

- Turning the Cyclic Redundancy Check (CRC) on and off
- Utilising different antennas (spring coil and wire dipole)
- Changing the distance between the transmitting and receiving stations
- Increasing the time delay between each radio packet transmit
- Utilising different coding rates (4/5, 4/6, 4/7 and 4/8)
- Increasing the amount of data in each packet (1 byte or 10 bytes per packet)
- Using different frequencies (437MHz, RFM96 module sand 915MHz, RFM95 module)

The RFM95 or RFM96 LoRa radio module was set to the default radio settings (BW is 125kHz and SF is 7, 128 chips/symbol) with the TX power level set to 5dBm and using a 437MHz or 915MHz radio frequency. The number of packets dropped were measured by counting the number packets received from transmitting 100 radio packets in a batch with each packet containing 1 byte. This process was repeated with 100 batches containing 100 packets for a total of 10,000 packets of data transmitted. An RTL-SDR dongle being driven by Gprx SDR receiver software was utilised during the testing to verify that each packet is transmitted. This testing being carried out is to verify if the radio packets are being transmitted and received without analysing the contents of the radio packets to verify the content of the data.

2) The second test conducted checks the number of packets that are dropped and verifies the contents of each packet to ensure that there are no bit errors in each byte. The method of testing is the same as the previous test (1 byte per packet, 100 packets per Batch and 100 Batches sent) with the default radio settings, 5dBm TX power and a 2m separation between transmitter and receiver. The test was carried out using the RFM95 module (915MHz) with the CRC on and off and using all the coding rates available.

3) The default radio settings for the RFM95 LoRa module (915MHz) was used with a 5dBm transmit power, 5/8 coding rate, using a 30dB attenuator between the transmitter and receiver and with the CRC turned both on and off. The method of testing was the same as the previous test for a total of 10,000 packets of data containing 1 byte. The *txGood* function was used in the transmitter from the *RHGenericDriver.h* file to count the number of packets successfully transmitted. The *rxGood* and *rxBad* functions from the *RHGenericDriver.h* file is used in the receiver to count the number of packets received successfully and the number of packets rejected due to errors.

4) The LoRa Modem Calculator Tool available from Semtech was used to estimate the major output parameters of the RFM96 radio module with the configuration being determined by the default radio settings available in the RadioHead Library, see Table G1 below.

*Table G1 - LoRa module configuration for the default RadioHead settings*

RadioHead default setting	Radio configuration	Bandwidth	Coding rate	Spreading factor	Preamble length
(0) - Bw125Cr45Sf128	Medium range & data rate	125 kHz	4/5	8 (128 chips/symbol)	12
(2) - Bw31_25Cr48Sf512	Long range, slow data rate	31.25 kHz	4/8	10 (512 chips/symbol)	12
(3) - Bw125Cr48Sf4096	Long range, slow data rate	125 kHz	4/8	12 (4096 chips/symbol)	12

The shortened beacon software cycle was used with the breadboard prototype using a transmit power of 15dBm to verify the transmit time of each RadioHead default setting. A Tester resistor was used to measure the current consumption of the beacon with the output being used to measure the transmit phase period to compare against the estimated value from the LoRa calculator.

5) To verify that the LoRa radio modules can operate with minimal data loss for the expected distances for a Low Earth Orbit (LEO), a transmitter and receiver were connected with a series of coaxial cables and attenuators that approximate the free-space path loss. The setup for testing the communication link is shown below in Figure G1, with the *A* and *B* attenuators being adjusted to change the free space path loss. The attenuation loss caused by the coaxial cables were calculated by measuring the length of the cable and using the datasheet for the expected cable loss (-60dB/m @ 437MHz) with the resultant values presented in red.

## Appendix G Communication Link testing method and results



Figure G1 - Test setup for verify the operation of the communications link for the free space path loss

The Generic “Hello World with RSSI” program found on the Adafruit website was used to send 30 bytes of data through the communications link with the combined value of the A and B attenuators being slowly increased until the receiver stops receiving consistent data. The highest value of attenuation that is achieved when the LoRa module is consistently receiving the “Hello World” message is determined to be the maximum value for the free-space path loss and is used to calculate the maximum operating distance for the reliable transfer of data. This test is repeated for the 0, 2 and 3 default radio settings using 5dBm, 10dBm, 15dBm and 20dBm transmit power.

## Results

1)

Table G2 - Observed number of dropped packets for varying radio parameters and test conditions

Test conditions - 10,000 packets sent, 100 packets per batch with 1 Byte per packet					
	Coding Rate	4/5	4/6	4/7	
<b>LoRa module Configuration</b>		<b>Number of packets dropped</b>			
CRC On, 100mS between Bytes, 915 Antenna, dist 50cm		4	13	7	8
CRC Off, 100mS between Bytes, 915 Antenna, dist 50cm		32	11	10	8
CRC On, 100mS between Bytes, 915 Antenna, dist 10m		45			
CRC Off, 100mS between Bytes, 915 Antenna, dist 10m		70			
CRC On, 100mS between Bytes, 433 Antenna (APM module), dist 2m		9			
CRC Off, 100mS between Bytes, 433 Antenna (APM module), dist 2m		9			
CRC On, 250mS between Bytes, 915 Antenna, dist 2m		10			5
CRC Off, 250mS between Bytes, 915 Antenna, dist 2m		7			9
CRC On, 500mS between Bytes, 915 Antenna, dist 2m		16			9
CRC Off, 500mS between Bytes, 915 Antenna, dist 2m		18			9
CRC On, No delay between Bytes, 915 Antenna, dist 2m *packet data checked*		10	10	13	11
CRC Off, No delay between Bytes, 915 Antenna, dist 2m *packet data checked*		5	12	16	15
Test conditions - 10,000 packets sent, 10 packets per batch with 10 Byte per packet					
	Coding Rate	4/5	4/6	4/7	
<b>LoRa module Configuration</b>		<b>Number of packets dropped</b>			
CRC Off, 100mS between Bytes, 915 Antenna, dist 2m		84			
Test conditions - 1,000 packets sent, 10 packets per batch with 10 Byte per packet					
	Coding Rate	4/5	4/6	4/7	
<b>LoRa module Configuration</b>		<b>Number of packets dropped</b>			
CRC On, 100mS between Bytes, 915 Antenna, dist 2m		5			
CRC Off, 100mS between Bytes, 915 Antenna, dist 2m		9			

2) The observed number of radio packets dropped is displayed above in Table G2 with the results indicating that the number of packets being dropped reduces when the CRC was turned on and if the number of CRC Bits was increased. There was a total of 80,000 packets of data sent and 79,904 packets received (79,904 bytes or 639,232 bits) with no bit errors observed in any of the bytes received.

3)

Table G3 - Testing statistics using the functions available in the RHGenericDriver.h file

Results from using the RHGenericDriver.h functions		
CRC Setting	On	Off
Number of packets Sent	10000	10000
Total txGood count	10000	10000
Number of packets received	9915	9988
Total rxGood count	9915	9988
Total rxBad count	79	3
Number of bytes not received	6	9

Appendix G  
Communication Link testing method and results

4)

*Table G4 - LoRa calculator tool estimated output values for the RadioHead (0) default setting*

LoRa modem calculator tool estimated radio parameters for RadioHead (0) default settings								
4 Byte identification radio packet								
TX Power (dBm)	TX Power (mW)	Bit Rate (bps)	Eb - Bit Energy (joules/bit)	Time on Air (mS)	Link Budget (dB)	Receiver sensitivity (dBm)	Transmit current (mA)	Transmit energy (J)
5	3.16	3125	0.00000101	59.9	132	-127	25	0.00494
10	10.00	3125	0.00000320	59.9	137	-127	31	0.00613
15	31.62	3125	0.00001012	59.9	142	-127	82	0.01621
20	100.00	3125	0.00003200	59.9	147	-127	125	0.02471
50 Byte telemetry radio packet								
TX Power (dBm)	TX Power (mW)	Bit Rate (bps)	Eb - Bit Energy (joules/bit)	Time on Air (mS)	Link Budget (dB)	Receiver sensitivity (dBm)	Transmit current (mA)	Transmit energy (J)
5	3.16	3125	0.00000101	182.78	132	-127	25	0.01508
10	10.00	3125	0.00000320	182.78	137	-127	31	0.01870
15	31.62	3125	0.00001012	182.78	142	-127	82	0.04946
20	100.00	3125	0.00003200	182.78	147	-127	125	0.07540
						Beacon Total transmisison time (mS)	422.38	
						Beacon Total transmisison time (S)	0.42238	

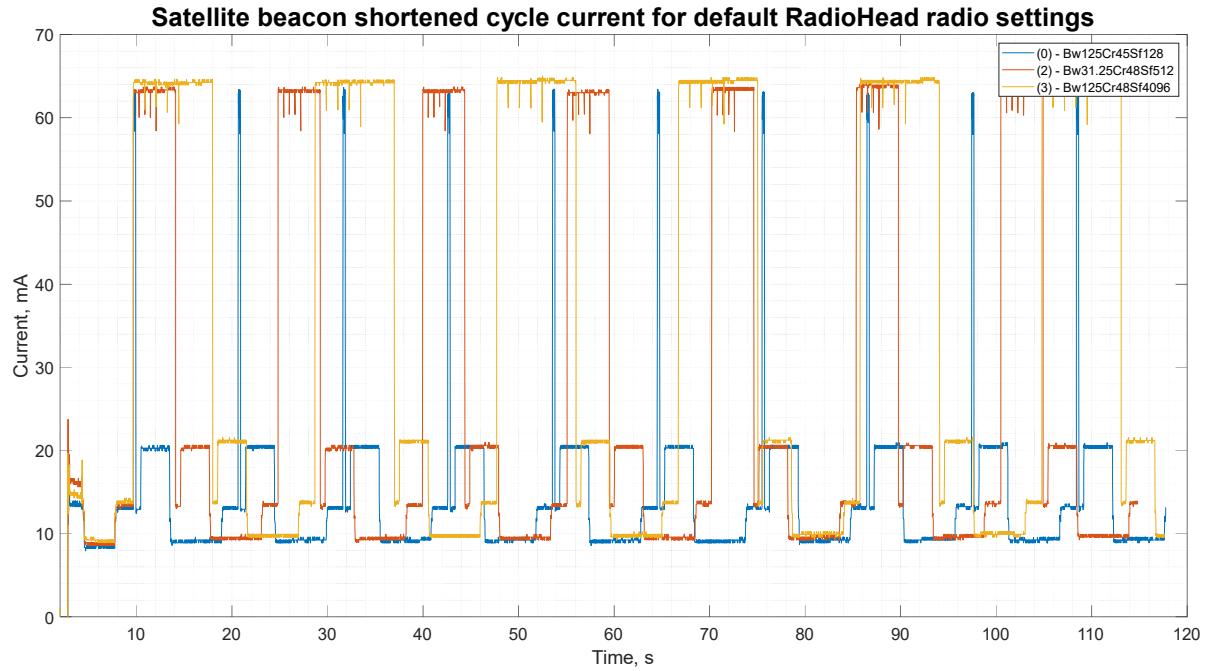
*Table G5 - LoRa calculator tool estimated output values for the RadioHead (2) default setting*

LoRa modem calculator tool estimated radio parameters for RadioHead (2) default settings								
4 Byte identification radio packet								
TX Power (dBm)	TX Power (mW)	Bit Rate (bps)	Eb - Bit Energy (joules/bit)	Time on Air (mS)	Link Budget (dB)	Receiver sensitivity (dBm)	Transmit current (mA)	Transmit energy (J)
5	3.16	152.59	0.00002072	1056.77	144.4	-139.4	25	0.08718
10	10.00	152.59	0.00006554	1056.77	149.4	-139.4	31	0.10811
15	31.62	152.59	0.00020724	1056.77	154.4	-139.4	82	0.28596
20	100.00	152.59	0.00065535	1056.77	159.4	-139.4	125	0.43592
50 Byte telemetry radio packet								
TX Power (dBm)	TX Power (mW)	Bit Rate (bps)	Eb - Bit Energy (joules/bit)	Time on Air (mS)	Link Budget (dB)	Receiver sensitivity (dBm)	Transmit current (mA)	Transmit energy (J)
5	3.16	152.59	0.00002072	3416.06	144.4	-139.4	25	0.28182
10	10.00	152.59	0.00006554	3416.06	149.4	-139.4	31	0.34946
15	31.62	152.59	0.00020724	3416.06	154.4	-139.4	82	0.92439
20	100.00	152.59	0.00065535	3416.06	159.4	-139.4	125	1.40912
						Beacon Total transmisison time (mS)	7643.14	
						Beacon Total transmisison time (S)	7.64314	

*Table G6 - LoRa calculator tool estimated output values for the RadioHead (3) default setting*

LoRa modem calculator tool estimated radio parameters for RadioHead (3) default settings								
4 Byte identification radio packet								
TX Power (dBm)	TX Power (mW)	Bit Rate (bps)	Eb - Bit Energy (joules/bit)	Time on Air (mS)	Link Budget (dB)	Receiver sensitivity (dBm)	Transmit current (mA)	Transmit energy (J)
5	3.16	183.11	0.00001727	1056.77	143	-138	25	0.08718
10	10.00	183.11	0.00005461	1056.77	148	-138	31	0.10811
15	31.62	183.11	0.00017270	1056.77	153	-138	82	0.28596
20	100.00	183.11	0.00054612	1056.77	158	-138	125	0.43592
50 Byte telemetry radio packet								
TX Power (dBm)	TX Power (mW)	Bit Rate (bps)	Eb - Bit Energy (joules/bit)	Time on Air (mS)	Link Budget (dB)	Receiver sensitivity (dBm)	Transmit current (mA)	Transmit energy (J)
5	3.16	183.11	0.00001727	2891.78	143	-138	25	0.23857
10	10.00	183.11	0.00005461	2891.78	148	-138	31	0.29583
15	31.62	183.11	0.00017270	2891.78	153	-138	82	0.78252
20	100.00	183.11	0.00054612	2891.78	158	-138	125	1.19286
						Beacon Total transmisison time (mS)	7118.86	
						Beacon Total transmisison time (S)	7.11886	

Appendix G  
Communication Link testing method and results



*Figure G2 - Satellite beacon current consumption for the default RadioHead radio settings*

*Table G7 - Measured transmission time for the default RadioHead LoRa radio settings*

Transmission time for default (0) RadioHead radio settings						Transmission time for default (2) RadioHead radio settings						
Cycle	Identification packet number	time (S)	Telemetry time (S)	Total time (S)		Cycle	Identification packet number	time (S)	Telemetry time (S)	Total time (S)		
1	0.03	0.03	0.04	0.03	0.24	1	0.58	0.60	0.59	0.60	2.02	4.39
2	0.03	0.03	0.04	0.03	0.22	2	0.58	0.60	0.58	0.60	2.02	4.38
3	0.03	0.04	0.04	0.03	0.23	3	0.57	0.60	0.58	0.60	2.03	4.38
4	0.03	0.03	0.03	0.04	0.23	4	0.57	0.60	0.60	0.59	2.02	4.38
5	0.04	0.03	0.03	0.04	0.24	5	0.57	0.60	0.60	0.60	2.01	4.38
6	0.03	0.03	0.04	0.03	0.22	6	0.59	0.58	0.60	0.60	2.03	4.40
7	0.03	0.04	0.04	0.04	0.23	7	0.60	0.60	0.60	0.60	2.00	4.40
Avg	0.03	0.03	0.04	0.03	0.23	Avg	0.58	0.60	0.59	0.60	2.02	4.39

Transmission time for default (3) RadioHead radio settings					
Cycle	Identification packet number	time (S)	Telemetry time (S)	Total time (S)	
1	1.17	1.19	1.18	1.19	3.54
2	1.17	1.20	1.19	1.19	3.53
3	1.21	1.19	1.18	1.19	3.53
4	1.17	1.19	1.19	1.20	3.53
5	1.18	1.19	1.19	1.18	3.53
6	1.20	1.20	1.10	1.20	3.60
Avg	1.01	1.02	1.00	1.02	3.04
					7.10

5)

*Table G8 - Estimated Link budget and measured FSPL attenuation for the default RadioHead LoRa module settings*

Largest measured FSPL attenuation and distance for reliable data transmission													
Transmit power (dBm)	5				10				15		20		
	RadioHead default setting	Receiver sensitivity (dB)	Estimated Link Budget (dB)	attenuation (dB)	Distance (kms)	Estimated Link Budget (dB)	attenuation (dB)	Distance (kms)	Estimated Link Budget (dB)	attenuation (dB)	Distance (kms)		
(0) Bw125Cr45Sf128	-127	132	131.14	195	137	137.14	390	142	141.14	620	147	145.14	980
(2) Bw31_25Cr48Sf512	-139.2	144.4	146.14	1100	149.4	151.14	1950	154.4	155.14	3100	159.4	161.14	6200
(3) Bw125Cr48Sf4096	-138	143	144.14	880	148	150.14	1750	153	155.14	3100	158	160.14	5500

## Testing Method

The Link budget for the communications link was broken up into 2 groups of calculations with the second group of calculations being carried out using two different methods. The first group of calculation was determining the link budget for the system hardware from transmitter to where the signal enters the receiver and the second group calculating from the receiver onwards including any software gains. A graphical representation of the Link budget is displayed below in Figure H1 followed by the equations used in the calculation.

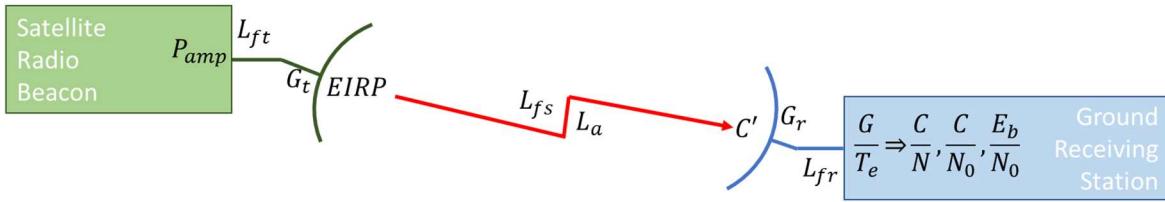


Figure H1 - Satellite radio beacon communications link budget representation

The first group of calculations from the transmitter amplifier to the receiver amplifier is as follows...

$$\text{Transmitter EIRP (EIRP) [dB]} - EIRP = P_{amp} - L_{bo} - L_f + G_t$$

$P_{amp}$  – Transmitter Power (dBW)

$L_{bo}$  – Back-off Loss (earth station only – for this assignment assume to be 0)

$L_f$  – feeder Loss (dB)

$G_t$  – Transmitter antenna Gain (dB)

$$\text{Free Space Loss (L}_FS\text{) [dB]} - L_{FS} = 92.44 + 20 \log(d_s \cdot f)$$

$d_s$  – slant range (kms)

$f$  – frequency (GHz)

$$\text{Carrier Power Density (C') [dBW]} - C' = EIRP - L_{FS} - L_A$$

$EIRP$  – Effective Isotropic Radiated Power (dB)

$L_{FS}$  – Free space Loss (dB)

$L_A$  – Atmospheric Loss (dB)

$$\text{Equivalent Noise Temperature (T}_e\text{) [dBK]} - T_e = 10 \log(T \cdot (NF - 1))$$

T – Receiver Environmental temperature (K)

NF – Receiver Noise Figure

$$\text{Receiver G/T}_e\text{ (G/T}_e\text{) [dBK}^{-1}\text{]} - G/T_e = G_r - T_e$$

$G_r$  – Receiver Antenna Gain (dB)

$T_e$  – Equivalent Noise Temperature (dBK)

The second set of calculations was performed in two ways, with the first method using the maximum bit rate figure obtained from the LoRa modem calculator tool. The bit energy to noise density ratio was calculated using the power to noise density ( $C/N_0$ ) value with the equations used being...

$$\text{C/N0 ratio [dB]} - C/\text{No} = C' - 10 \log(k) + \frac{G}{T_e} - L_{fr}$$

$C'$  – Carrier Power Density (dBW)

$L_{fr}$  – Receiver feeder Loss (dB)

$k$  – Boltzmann's constant ( $1.36 \times 10^{-23} \text{ J K}^{-1}$ )

$$\text{Eb/No Ratio [dB]} - Eb/\text{No} = C/\text{No} - 10 \log(f_b)$$

$f_b$  – maximum bit rate (bit/s)

The second method manually calculates the maximum bit rate from the bandwidth efficiency while using the Carrier to Noise (C/N) to calculate the bit energy to noise density ratio.

$$\text{Noise Density [dBW/Hz]} - N_0 = \frac{G}{T_e} - 10 \log(k) - L_{rf}$$

## Appendix H Communication link budget calculations

$G_r$  – Receiver Antenna Gain (dB)

$T_e$  – Equivalent Noise Temperature (dBK)

$L_{fr}$  – Receiver feeder Loss (dB)

$k$  – Boltzmann's constant ( $1.36 \times 10^{-23} Jk^{-1}$ )

$$\text{Raw bit rate [bit/s]} - R_b = SF * \frac{BW}{2 * SF}$$

SF – Spreading factor

BW – Bandwidth (Hz)

$$\text{Effective bit rate [bit/s]} - R_{b\ eff} = R_b * FEC$$

$R_b$  – Raw bit rate (bit/s)

FEC – Forward error correction code rate

$$\text{Bandwidth efficiency [bit/Hz]} - \eta = \frac{R_b}{BW}$$

$R_b$  – Raw bit rate (bit/s)

BW – Bandwidth (Hz)

$$\text{Total Noise Power [dBw]} - N = N_0 + 10 \log_{10}(BW)$$

$N_0$  – Noise density (dBw/Hz)

BW – Bandwidth (Hz)

$$\text{Carrier to Noise [dB]} - \frac{C}{N} = C' - N$$

$C'$  - Carrier Power Density (dBw)

$N$  – Total noise power (dBw)

$$\text{Eb/No ratio [dB]} - \frac{E_b}{N_0} = \frac{C}{N} + 10 \log_{10} \left( \frac{BW}{R_{b\ eff}} \right)$$

$\frac{C}{N}$  – Carrier to Noise (dB)

BW – Bandwidth (Hz)

$R_{b\ eff}$  – Effective bit Rate (bit/s)

## Appendix H

### Communication link budget calculations

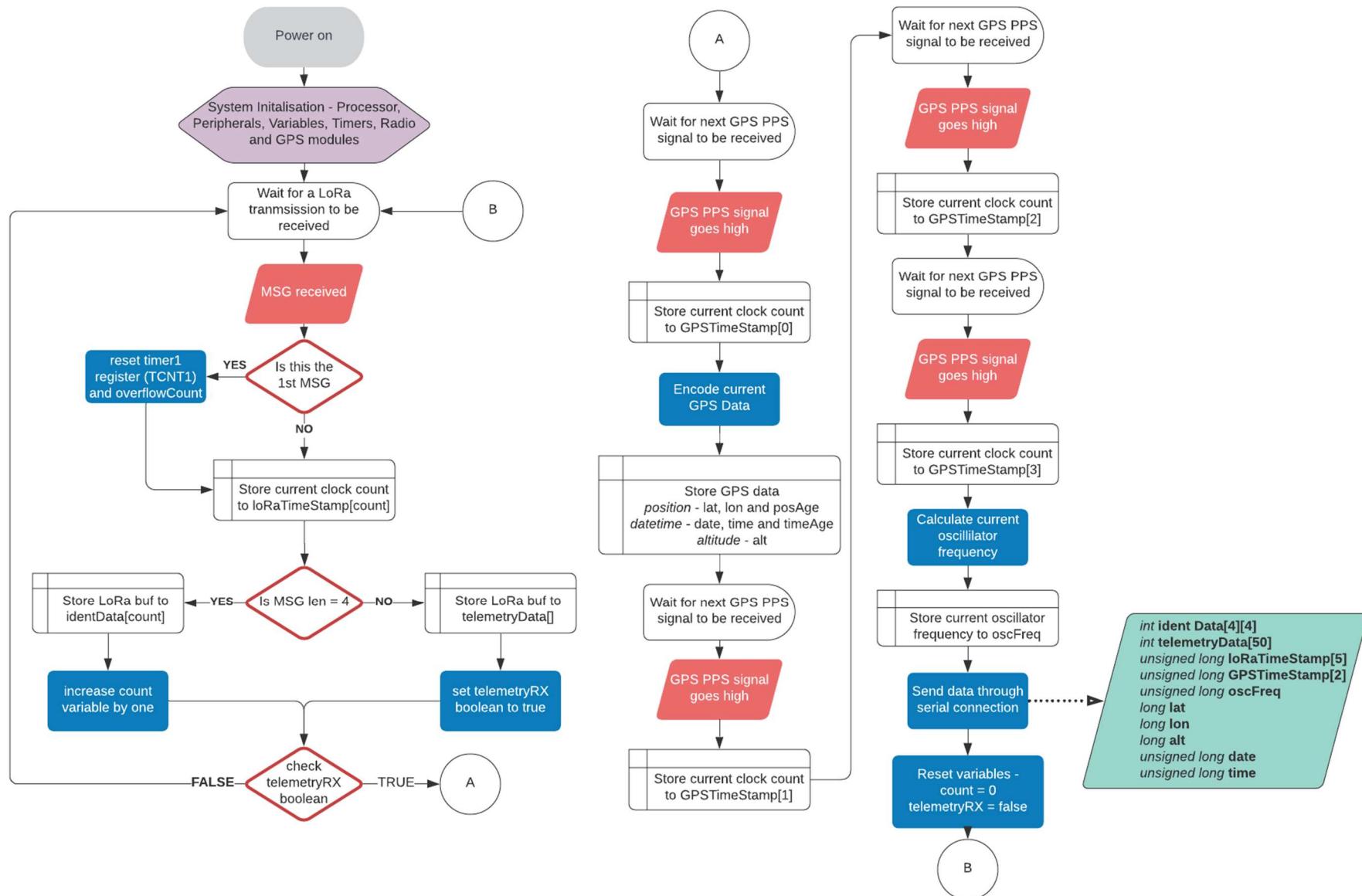
### Results

*Table H1 - Link Budget calculations for the RadioHead library default long range settings for a 2000km transmission path*

<b>Link Budget calculations for the satellite radio beacon communications link</b>										
LoRa module RadioHead default settings used			(2) Long Range Settings			(3) Long Range Settings				
Parameter	Symbol	Equation	Downlink	Downlink	Downlink	Downlink	Downlink	Downlink	Downlink	
Transmitter Power (dBm)	Pamp (dBm)		5	10	15	20	5	10	15	20
Transmitter Power (dB)	Pamp (dB)	Pamp = Pamp[dBm] - 30	-25	-20	-15	-10	-25	-20	-15	-10
Frequency (GHz)	f		0.437	0.437	0.437	0.437	0.437	0.437	0.437	0.437
Transmitter Antenna Gain (dBi)	Gt (dB)		3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
Transmitter Feeder Loss (dB)	Lft		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Transmitter EIRP (dB)	EIRP	EIRP = Pamp[dB] - Lf + Gt	-22	-17	-12	-7	-22	-17	-12	-7
Slant Range (km)	ds		2000	2000	2000	2000	2000	2000	2000	2000
Free Space Loss (dB)	Lfs	Lfs = 92.44 + 20Log(ds*f)	151.27	151.27	151.27	151.27	151.27	151.27	151.27	151.27
Atmospheric Loss (dB)	La		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Carrier Power Density (dBW)	C'	C' = EIRP - Lfs - La	-173.77	-168.77	-163.77	-158.77	-173.77	-168.77	-163.77	-158.77
Receiver Noise Figure	NF		6	6	6	6	6	6	6	6
Receiver Environmental Temperature (K)	T		290	290	290	290	290	290	290	290
Equivalent Noise Temperature (K)	Te	Te = T*(NF-1)	1450	1450	1450	1450	1450	1450	1450	1450
Equivalent Noise Temperature (dBK)	Te (dBK)	10^Log(Te[k])	31.61	31.61	31.61	31.61	31.61	31.61	31.61	31.61
Receiver Antenna Gain (dBi)	Gr (dB)		3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
Receiver G/Te (dBK-1)	G/Te	G/Te = Gr - Te[dBk]	-28.11	-28.11	-28.11	-28.11	-28.11	-28.11	-28.11	-28.11
Receiver Feeder Loss (dB)	Lfr		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Boltzmann's Constant (dBJK-1)	10*log(k)	k = 1.38E-23[JK-1]	-228.60	-228.60	-228.60	-228.60	-228.60	-228.60	-228.60	-228.60
<b>Method of calculation using the manually calculated maximum bit rate</b>										
Noise Density (dBW/Hz)	N0	N0 = G/Te - 10*log(k) - Lrf	-199.99	-199.99	-199.99	-199.99	-199.99	-199.99	-199.99	-199.99
bandwidth (Hz)	BW		31250	31250	31250	31250	125000	125000	125000	125000
FEC code rate	CR		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
spreading factor	SF		10	10	10	10	12	12	12	12
raw bit rate (bit/s)	Rb	Rb = SF*BW/(2*SF)	305.18	305.18	305.18	305.18	366.21	366.21	366.21	366.21
effective bit rate (bit/s)	Rb eff	Rb eff = Rb*CR	152.59	152.59	152.59	152.59	183.11	183.11	183.11	183.11
Bandwidth efficiency (bit/Hz)	$\eta$	$\eta = Rb/BW$	0.009766	0.009766	0.009766	0.009766	0.00293	0.00293	0.00293	0.00293
Total Noise Power (dBW)	N	N = N0 + 10*log(BW)	-155.04	-155.04	-155.04	-155.04	-149.02	-149.02	-149.02	-149.02
Carrier to Noise (dB)	C/N	C/N = C' - N	-18.73	-13.73	-8.73	-3.73	-24.75	-19.75	-14.75	-9.75
Bit energy to Carrier noise ratio	Eb/N0	Eb/N0 = C/N + 10*log(BW/Rb eff)	4.382	9.382	14.382	19.382	3.590	8.590	13.590	18.590
<b>Method of calculation using the LoRa calculator maximum bit rate</b>										
Carrier to Noise Density (dB)	C/No	C/No = C' + G/Te - 10Log(k) - Lfr	26.22	31.22	36.22	41.22	26.22	31.22	36.22	41.22
Maximum bit-rate (bit/s)	fb		152.60	152.60	152.60	152.60	183.10	183.10	183.10	183.10
Bit energy to Carrier noise ratio	Eb/No	Eb/No = C/No - 10Log(fb)	4.382	9.382	14.382	19.382	3.590	8.590	13.590	18.590

The green parameters are constants, the blue parameters are values obtained from component datasheets, the red parameters are assumed values and the black parameters are calculated values.

Appendix I  
Ground receiving station software flow chart



## Testing Method

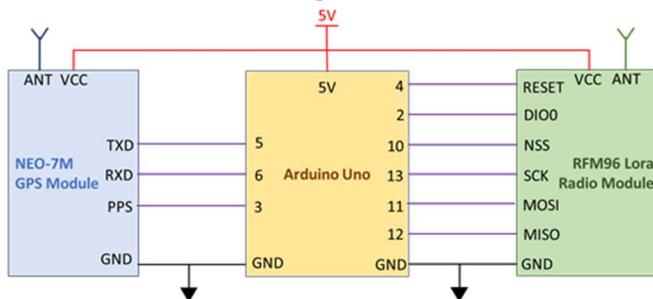


Figure J1 - Ground receiving station configuration and connections

**1) The resolution of the Arduino built-in timer function, *micros()*** – The initial development of measuring the time of arrival was carried out using the Arduino built-in timing function, *micros()*, which was found to have a resolution of  $4.096\mu\text{s}$  and this resolution would equal to a distance measurement error of 1.2kms. To overcome this error in measurement it was decided to utilise the clock cycles of the 16MHz external oscillator on the Arduino Uno board which could provide a smaller timing resolution of 62.5ns (18.75m distance measurement error).

This method was implemented by utilising the Timer1 counter in the ATMEGA328P which is a 16-bit counter (65,536 values) that will keep count of each individual clock cycle (62.5ns long). An Interrupt Service Routine (ISR) using *TIMER1\_COMPA\_vect* will be carried out at the top of the Timer1 register (4.096ms after the counter begins) in which a variable that keeps the count of the number of Timer1 overflows that occur. This count multiplied by 65536 combined with reading the current number of clock counts (TCNT1 variable) will give the total number of clock cycles that have occurred since the Timer1 counter was begun or reset.

To test if this method of counting clock cycles was a valid method, the number of clock cycles were counted between each GNSS PPS pulse which occurs every one second with a tolerance of 30ns. The tolerances and environmental factors that affect the frequency of the oscillator will cause the number of clock cycles between each 1 second pulse to not be the nominal value of 16,000,000 cycles. To test the repeatability of measuring the clock cycle, the number of cycles between pulses are compared to the previous value to determine the difference in the number of cycles. The configuration of the ground receiver station that will be utilise in all testing is shown above in Figure J1.

**2) The number of clock cycles taken to carry out an ISR** – The ATMEGA328P processor that is utilised in the Arduino Uno has a priority list for the interrupts that are available to be used with the highest priority being the reset interrupt followed by the external interrupt which is being used by the GNSS PPS signal in this project. The next interrupt priority is the Timer1 ISR which is being used to count the clock cycles. The execution for the interrupt process is the same for all AVR microcontroller (this includes the ATMEGA328P processor) which has been found to take 23 clock cycles to start the execution of the code in the routine (<http://www.gammon.com.au/forum/?id=11488>). This would introduce a delay of  $1.4375\mu\text{s}$  each time the time stamp is read for the message or GNSS PPS signals, but as the Interrupt Service Routine (ISR) is executed in the same way for every ground station then the delay can be included in each distance measurement calculation or ignored as it will not affect the time difference of arrival. The delay will be ignored as it is constant for all stations and will help to simplify the TDOA calculations.

**3) Oscillator frequency drift due to temperature, tolerances and other sources of error** – The oscillator used for processor timing in each ground station that will be used in calculating the position of the satellite will have a difference deviation from the expected frequency (16MHz). The frequency drift of an oscillator can be attributed to many sources such as age, local temperature, frequency stability, voltage stability, etc. This requires that the instantaneous frequency of the oscillator at each ground station be known such that the length of the clock cycle can be determined accurately. The PPS signal provided by the GNSS module will be used as a reference to determine the instantaneous frequency of a ground receiver station oscillator by counting the number of clock pulses between several PPS pulses to determine the average number of pulse in 1 second.

## Appendix J

### Ground receiving station time measurement error investigation, testing and results

**4) Tolerance of the GNSS PPS signal** – The data sheet for the U-Blox NEO-7N GNSS module lists a tolerance of 30ns for the PPS signal provided that equates to a position measurement error of 9m. An error of 30nS for the PPS signal is an acceptable level of tolerance for this application and as such will require no further investigation.

**5) Accuracy of the GNSS position of the ground station** – The data sheet for the U-Blox NEO-7N GNSS module lists a tolerance of 2.5m for the positional accuracy. An initial investigation found that this tolerance was generally achieved after operating the GNSS module for a larger period, but the error in position was greater when the unit is initially operated. The positional error was found to be no greater than 10m during the initial operation of the NEO-7N GNSS module and this uncertainty will be used for the error in the GNSS position of the ground station.

**6) The time taken for error checking in the LoRa module** – When the LoRa module received a signal then there is a processing routine carried out by the software built into the module where the validity of the preamble is checked, the radio packed is error checked, any error are corrected and then an interrupt routine is carried to produce a high signal on the DIO0 pin (*RXdone* interrupt). The time it takes to carry out this routine and produce the *RXdone* varied from module to module and signal to signal which requires testing into the expected time difference for the interrupt. The testing of the time difference in processing is carried out by connecting two ground receive stations to a single radio beacon using co-axial cable and a T-Piece connector and observing the time difference between each LoRa module when the *RXdone* interrupt goes from low to high. This test will be repeated for a large sample size to determine the expected means and variances in the time difference.

**7) The resolution of the ATMEGA328P processor clock cycles in the Arduino Uno** – The initial design decision to use an Arduino Uno for the processor assumed a small margin of error in the timing measurement. It was found during the development of the ground station that a compounding number of errors was causing a larger error in distance than originally assumed. An alternative processor, Arduino Due, was investigated as the external oscillator is an 84MHz clock which equates to a clock cycle of 11.9ns (3.57m error) and a reduction in the distance measuring error of over 15ms. The initial investigation into using the Arduino Due found some difficulties in accessing the timers and porting the existing code across which resulted in the decision to continue the development using the Arduino Uno. If the error in distance measurement is too large at the conclusion of the initial development with the Arduino Uno, then using the Arduino Due will be investigated further.

## Results

1)

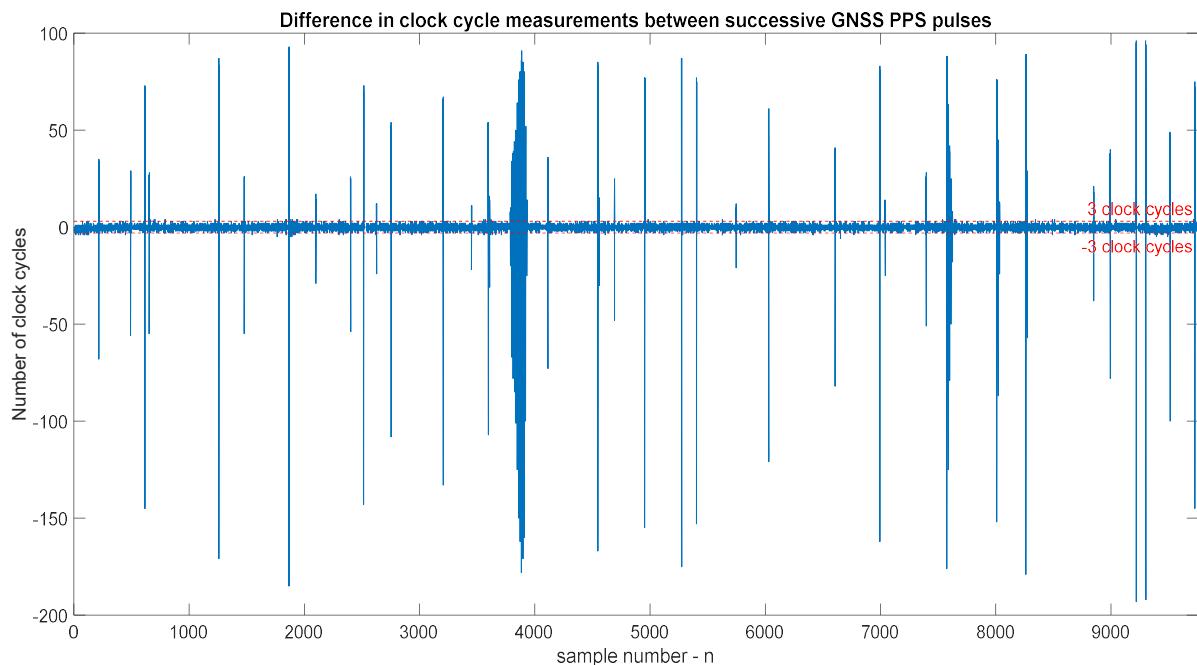
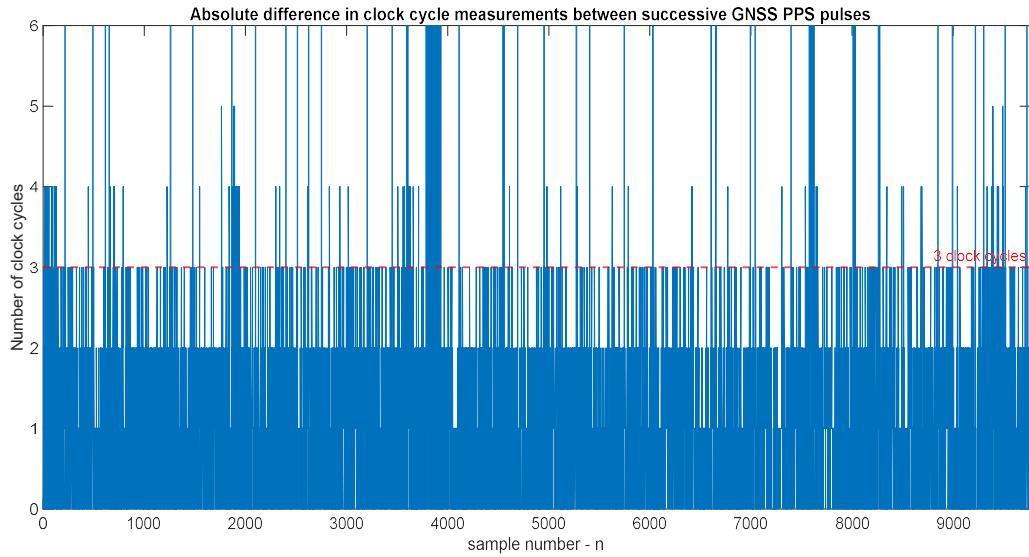


Figure J2 - Difference in clock cycle measurements between successive GNSS PPS pulses



*Figure J3 - Higher resolution absolute difference in clock cycles at the 3-cycle threshold*

The graph displayed above in Figure J1 shows a small number of spikes in the difference in clock cycles that are caused by incorrect ISR procedures, but most of the measurements are below 3 clock cycle which is shown above in Figure J2. The results show that when 9757 samples are tested for the clock cycle differences then 260 of the samples are above the 3-cycle threshold, which equates to an 2.7% error rate. If the threshold is increases to 4 clock cycles, then the error rate reduces to 1.8% and if the threshold is decreases to 2 clock-cycles then the error rate increases to 9.2%. The resultant statistics, including the spikes, has mean of 2.2404 and a standard deviation of 10.6291 which indicates that 95% of the absolute measurements are between 0 and 23.4986 clock cycles. The cause of the spikes in the measurements is a known phenomenon (discussed in the next paragraph) and can be removed from the statistics. When the spikes were removed then the absolute statistics has a mean of 0.9134 and standard deviation of 0.7213 which has 95% of absolute measurements between 0 and 2.356 clock cycles. If the statistics were determined on the measured values, then the mean is 0.0266 with a standard deviation of 1.1636 which results in 95% of the measurements being between -2.3006 and 2.3538 clock cycles.

The spikes present in Figure J2 (see above) show an increase in the error of clock cycles between 1-second uniform events by up to 180 clock cycles and can be contributed to the implementation of the ISR in the software. The Arduino IDE is based upon C programming in which the processor will complete the execution of the current list of command before an ISR is carried out. The number of clock cycles it takes to execute the current list of commands is variable which is shown in the differing values of the spikes. The software program developed for testing the number of clock cycles between the GNSS PPS signal uses two interrupt routines, an external interrupt and a clock compare interrupt, with the external interrupt occurring on the PPS signal and the clock compare used to count how many time the clock overflows. The ISR hierarchy dictates that the external interrupt occurs first then that ISR must be carried out before the clock compare ISR causing a delay in incrementing the clock overflow and significantly changing the number of clock cycles being counted between PPS signals.

6)

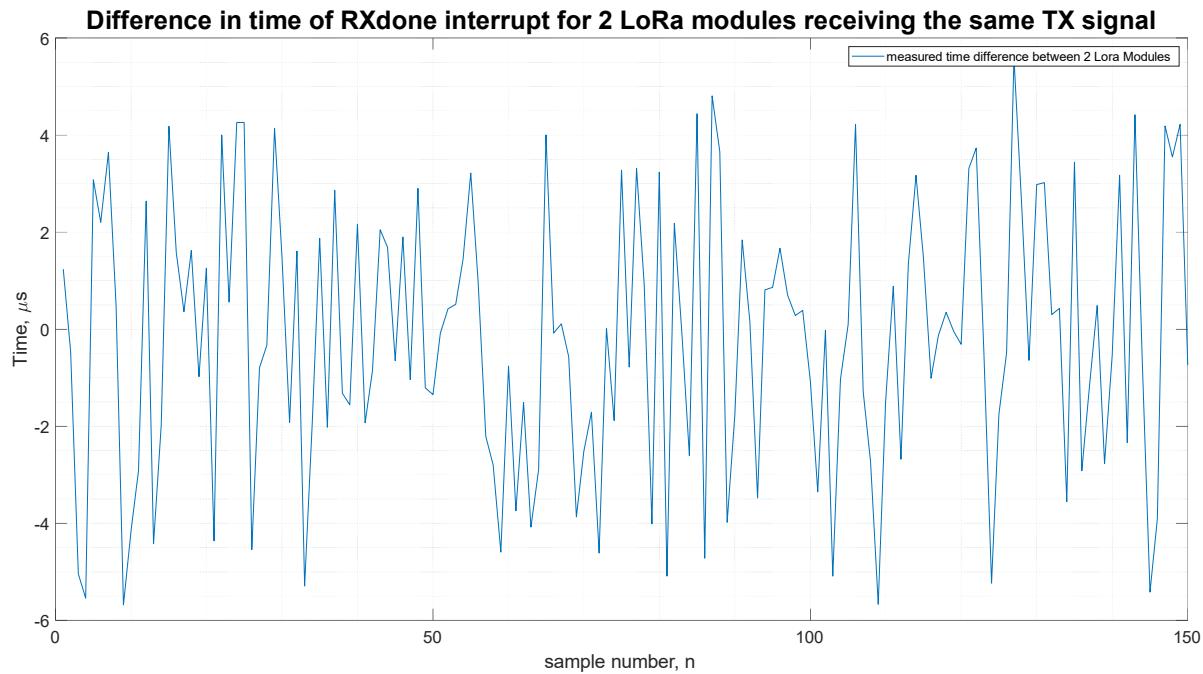


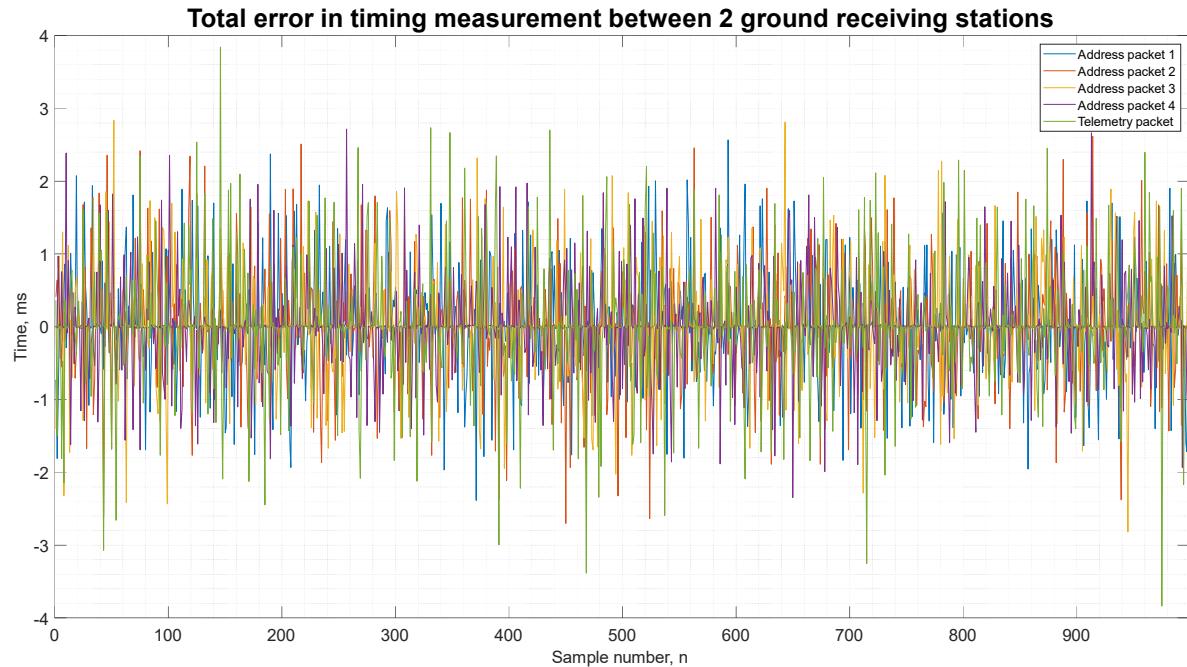
Figure J4 - Time differences for the RXdone interrupt between 2 LoRa module from a single TX source

The results in Figure J4 above show the values for the time difference measured between the *RXdone* interrupts for 2 ground stations when receiving a LoRa signal from a single transmit source. The resultant statistics taken from the absolute values of the collected data show that the difference in time between the LoRa modules has a mean of  $2.2901\mu s$ , a standard deviation of  $1.6042\mu s$  and a maximum difference of  $5.68\mu s$  with 95% of the data being between 0 and  $5.4985\mu s$ . When the statistics calculations are performed on the measured data the resultant mean is  $-0.1803\mu s$  with a standard deviation of  $2.7966\mu s$  which results in 95% of the measured data being between  $-5.7735\mu s$  and  $5.4129\mu s$ . Using the Cumulative Distribution Function of the measured data has found that 92.56% of the data lies between  $\pm 5\mu s$ , 96.56% of the data lies between  $\pm 5.5\mu s$  and 96.77% lies between  $\pm 6\mu s$ .

## Testing Method

The testing method used to verify the total uncertainty in measurements utilises the same test setup used to verify the difference in LoRa processing time where the satellite radio beacon is connected directly to two ground receiving stations via a coaxial cable and T-Piece. This direct connection ensures that the transmitted RF signal will arrive at each ground stations simultaneously in which no time difference of arrival should be observed in the measured timing data. The collected timestamps from the ground receiving station will not be the same due the uncertainties identified during the previous testing. A large sample of timestamps from the radio packet transmissions were collected by the ground receiving station and sent through a serial peripheral connection to be processed by MATLAB. The format of the data that is sent from the Arduino to MATLAB in order is: four address timestamps, one telemetry time stamp, four GNSS PPS timestamps, the instantaneous oscillator frequency as calculated in Arduino, the UTC date and UTC time of arrival of the first GNSS PPS timestamp. The number of clock cycle between the first GNSS timestamp to the address and telemetry time stamps are multiple by the inverse instantaneous oscillator frequency to determine the amount of time that has passed since the PPS signal where the GNSS data was collected. The amount of time between each timestamp for a single set of transmissions are compared against the results of the other ground receiver station to find the difference which represents the uncertainty in time measurement. The difference in the timestamps for each packet will be indicative of the total sum of all errors in measuring the difference time of arrival. The only uncertainty that is not included in this final total error is the positional tolerance of the GNSS which is much smaller than all other errors.

## Results



*Figure K1 - Total error in timing measurement between 2 ground receiving stations*

Appendix K  
Ground receiving station final verification testing method and results

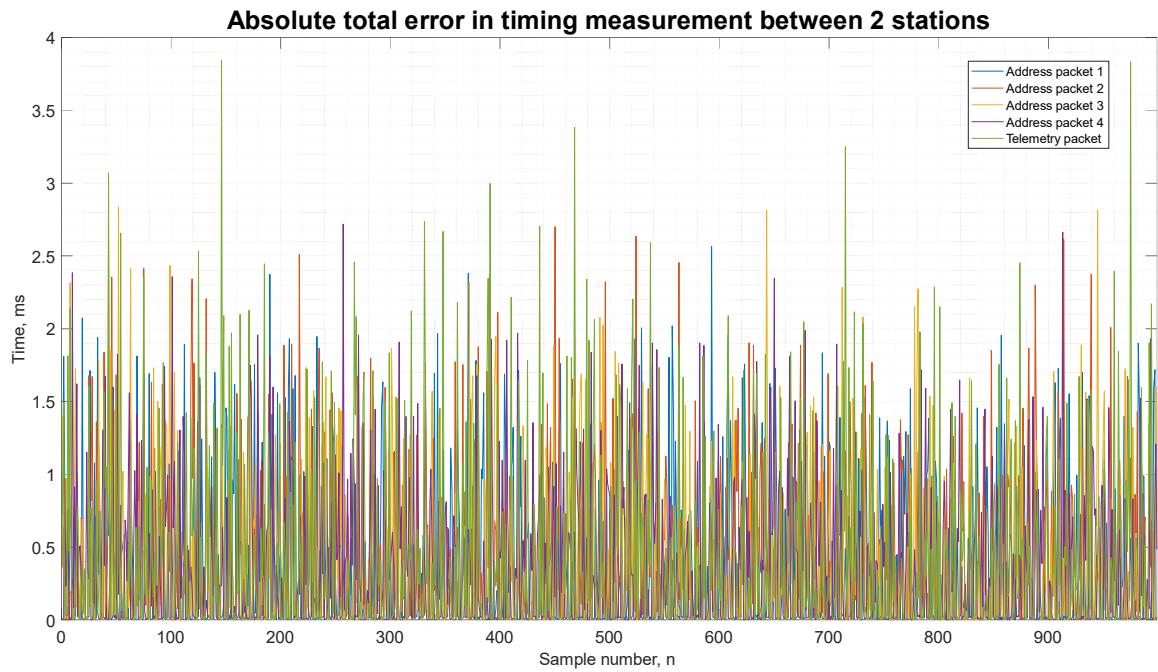


Figure K2 - Absolute total error in timing measurement between 2 ground receiving stations

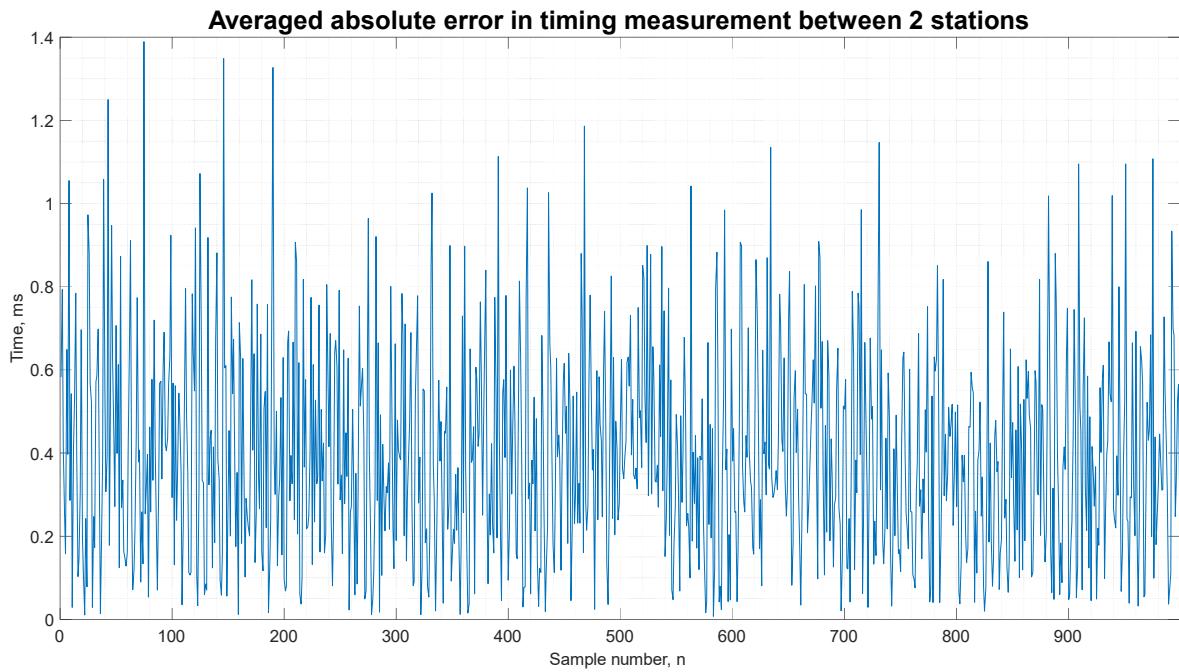


Figure K3 – Averaged absolute total error in timing measurement between 2 ground receiving stations

Table 9 - Total error in measurement statistics

	Measured values				Absolute values			
	mean ( $\mu$ s)	Std Dev ( $\mu$ s)	95% of data interval (ms)		mean ( $\mu$ s)	Std Dev ( $\mu$ s)	95% of data interval (ms)	
<b>Address packet 1</b>	19.816	638.83	-1.257844	1.297476	381.48	512.66	0	1.4068
<b>Address packet 2</b>	17.482	681.87	-1.346258	1.381222	409.35	545.46	0	1.50027
<b>Address packet 3</b>	38.799	652.44	-1.266081	1.343679	389.8	524.49	0	1.43878
<b>Address packet 4</b>	51.243	624.07	-1.196897	1.299383	379.93	498.02	0	1.37597
<b>Telemetry packet</b>	24.433	759.46	-1.494487	1.543353	419.14	633.66	0	1.68646
				<b>Averaged values</b>	395.14	250.2	0	0.89554