

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	
LoRa module Configuration		Number of packets dropped			
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	
LoRa module Configuration		Number of packets dropped			
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	
LoRa module Configuration		Number of packets dropped			
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 G1 - 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 G2 - 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 G3 - 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

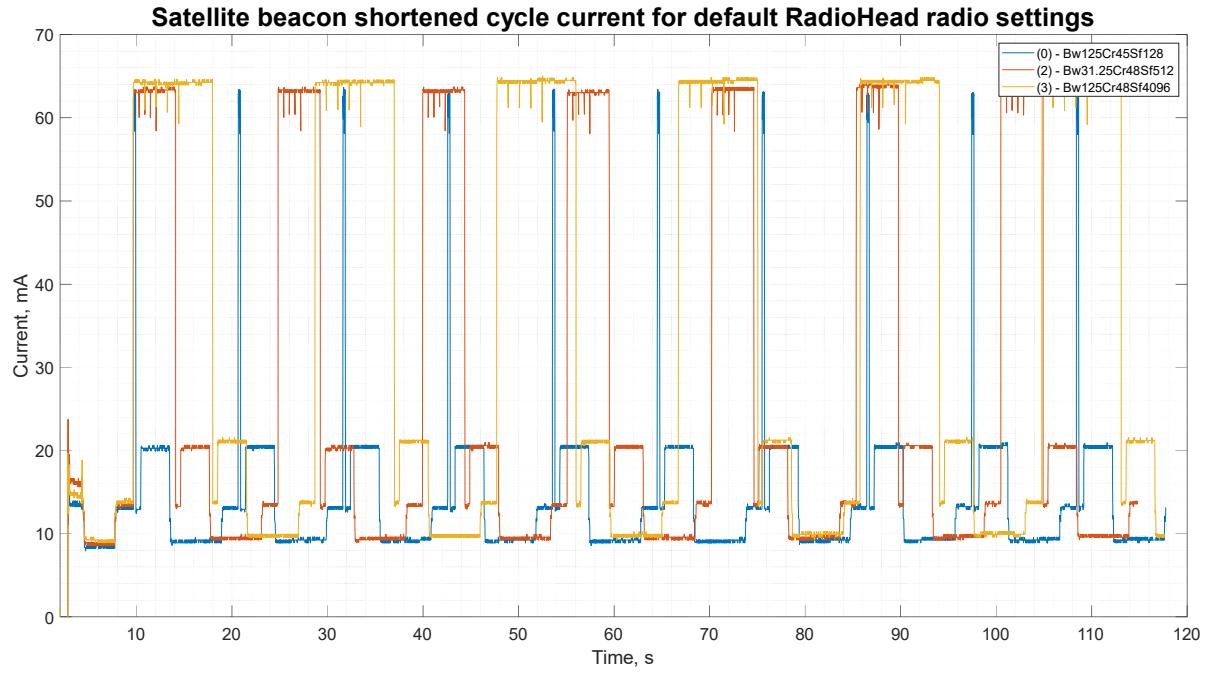


Figure G2 - Satellite beacon current consumption for the default RadioHead radio settings

Table G4 - 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 G5 - 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.

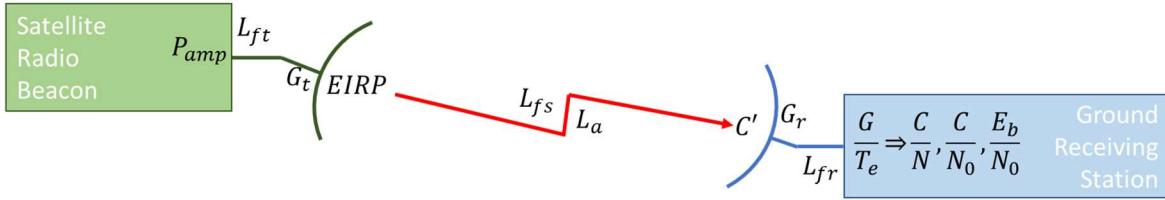


Figure H1 - Satellite radio beacon communications link budget representation

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_o) value with the equations used being...

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

C' – Carrier Power Density (dBW)

L_{fr} – Receiver feeder Loss (dB)

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

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

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.

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

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

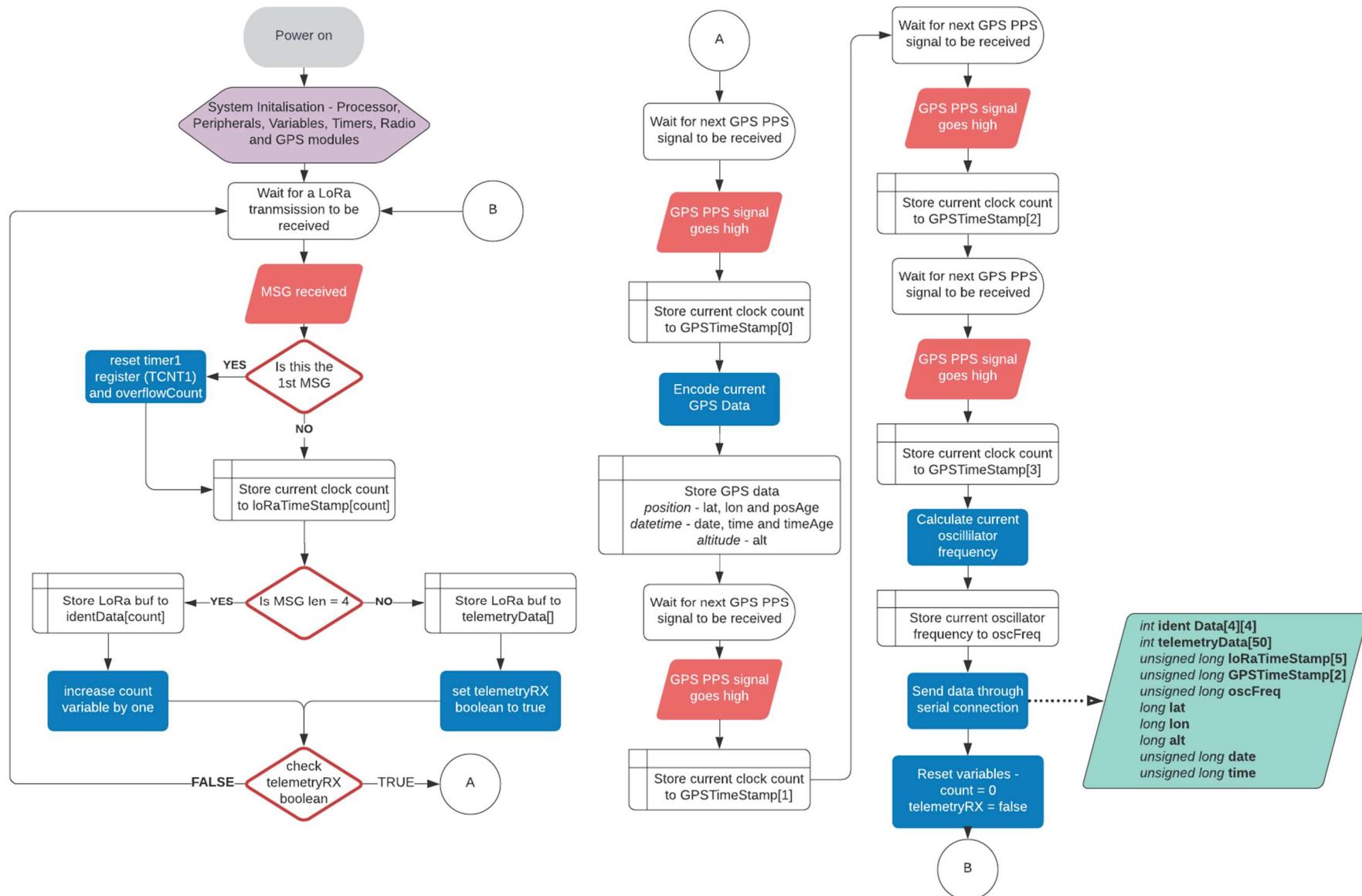
Results

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

Link Budget calculations for the satellite radio beacon communications link										
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
Method of calculation using the manually calculated maximum bit rate										
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)	η	$\eta = Rb/BW$	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
Method of calculation using the LoRa calculator maximum bit rate										
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

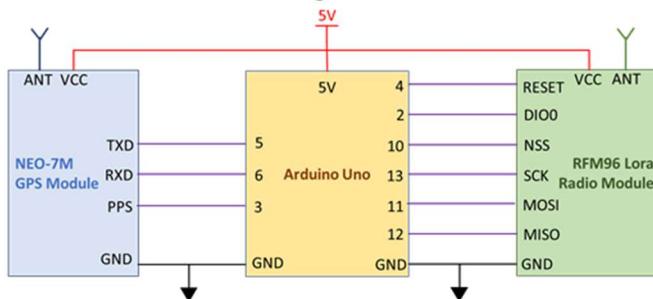


Figure J1 - Ground receiving station configuration and connections

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 1.2km s. 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)

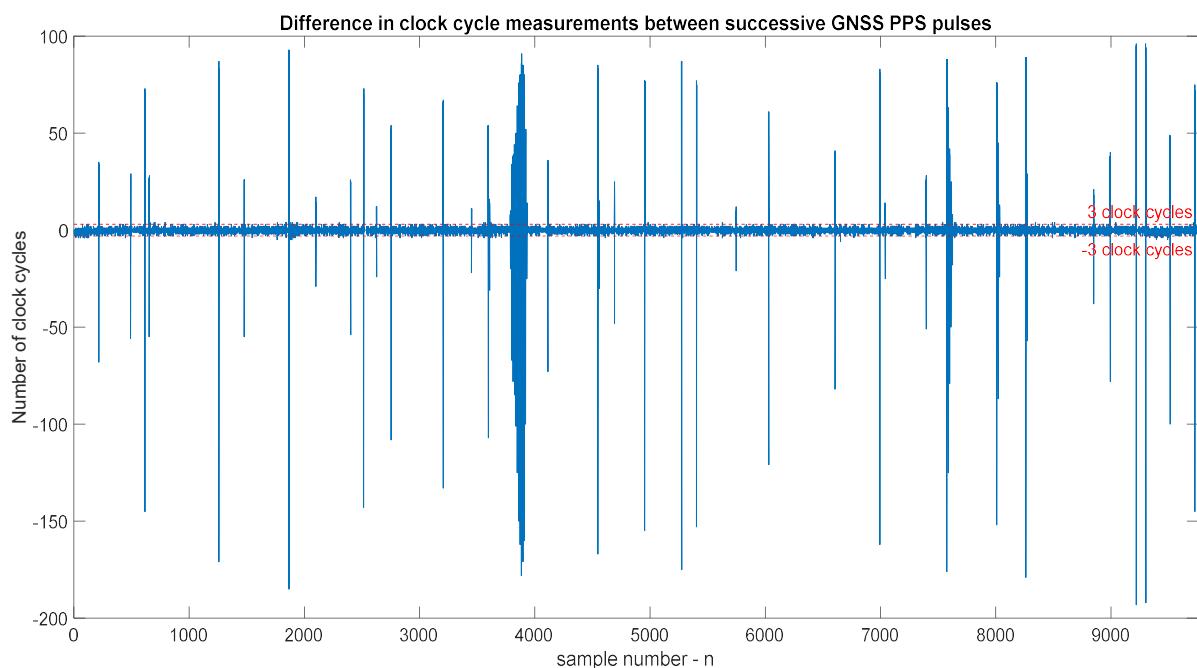


Figure J2 - Difference in clock cycle measurements between successive GNSS PPS pulses

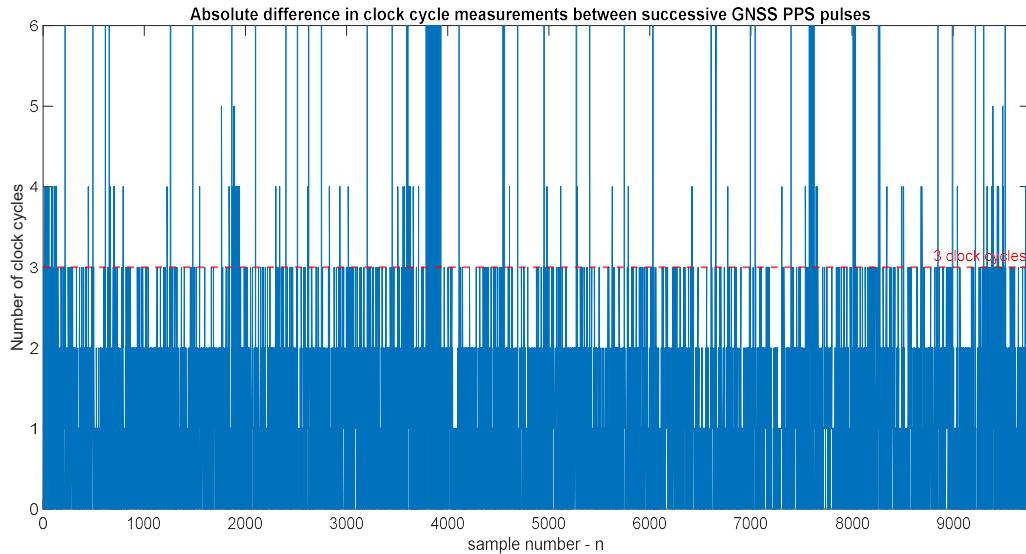


Figure J3 - Higher resolution absolute difference in clock cycles at the 3-cycle threshold

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)

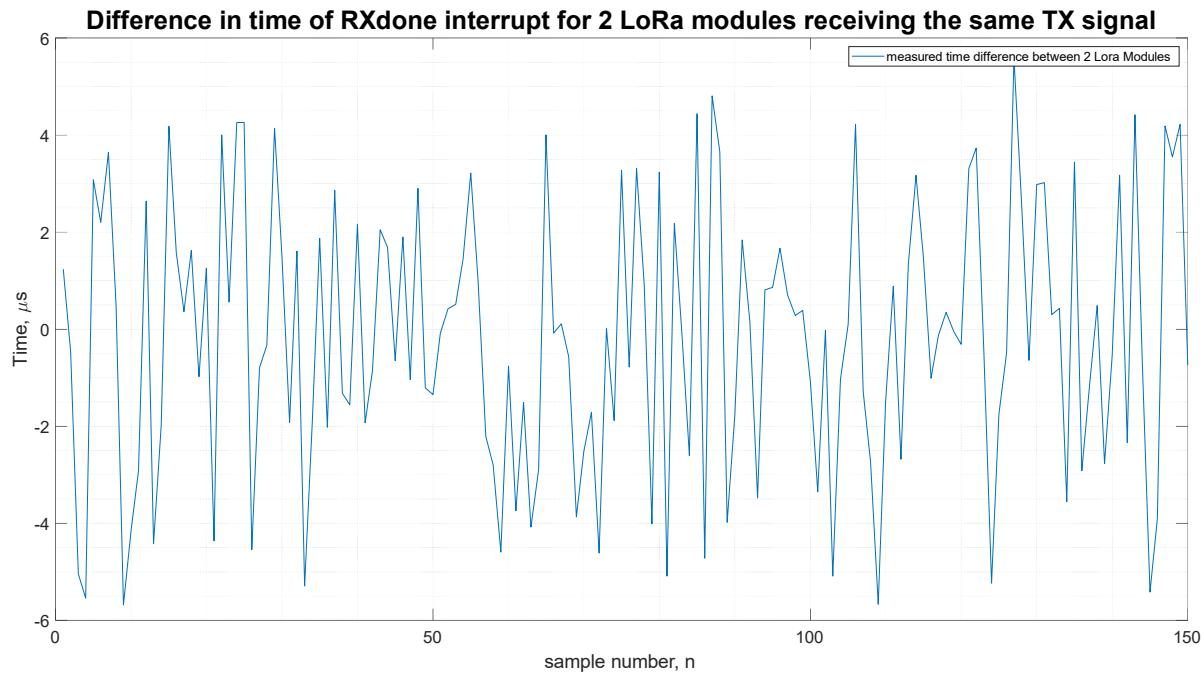


Figure J4 - Time differences for the RXdone interrupt between 2 LoRa module from a single TX source

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

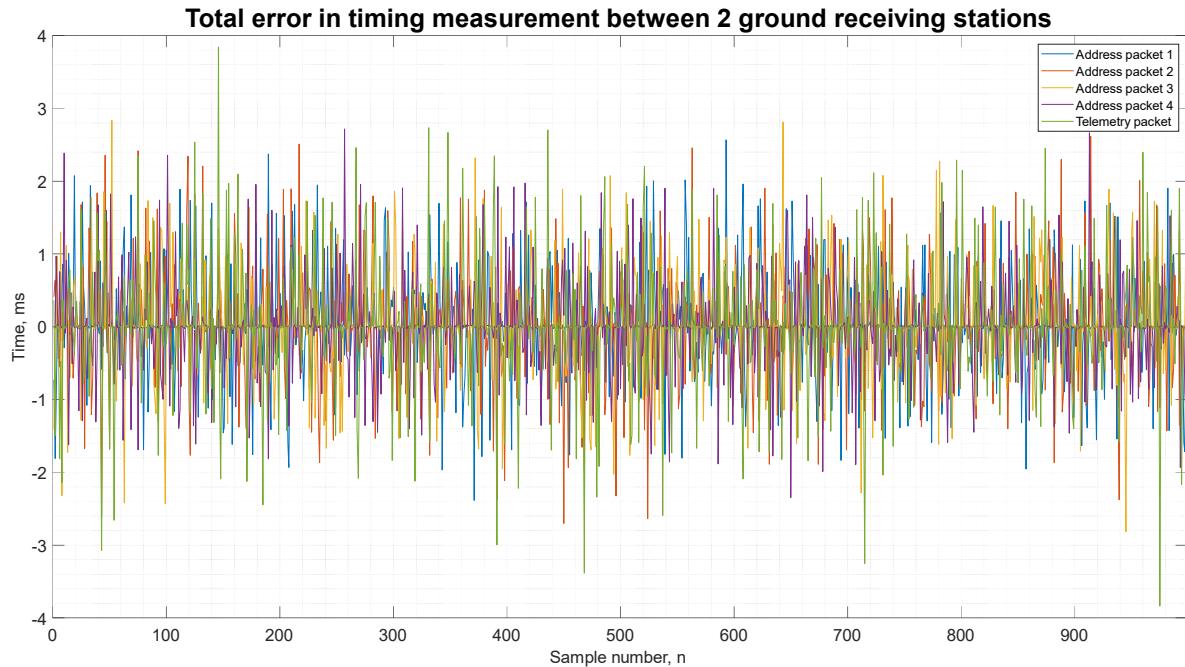


Figure K1 - Total error in timing measurement between 2 ground receiving stations

Appendix K
Ground receiving station final verification testing method and results

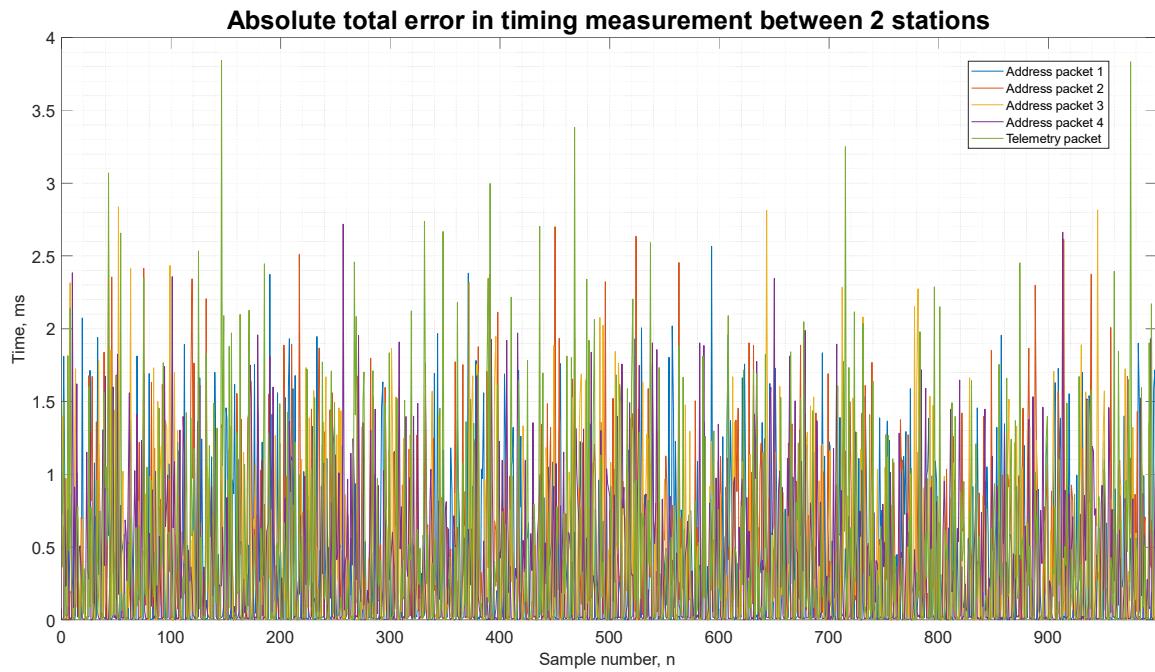


Figure K2 - Absolute total error in timing measurement between 2 ground receiving stations

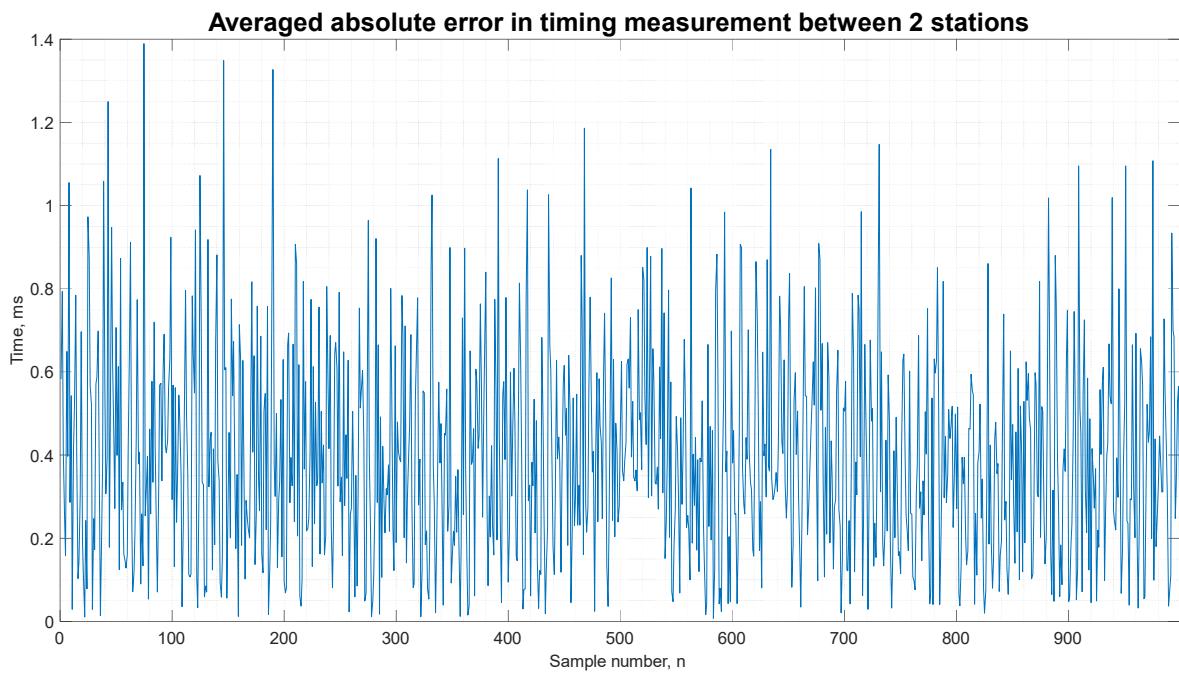


Figure K3 – Averaged absolute total error in timing measurement between 2 ground receiving stations

Table 6 - Total error in measurement statistics

	Measured values				Absolute values			
	mean (μ s)	Std Dev (μ s)	95% of data interval (ms)		mean (μ s)	Std Dev (μ 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
				Averaged values	395.14	250.2	0	0.89554