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'.

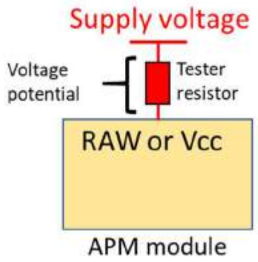


Figure A1 - Tester resistor configuration for determining current consumption

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

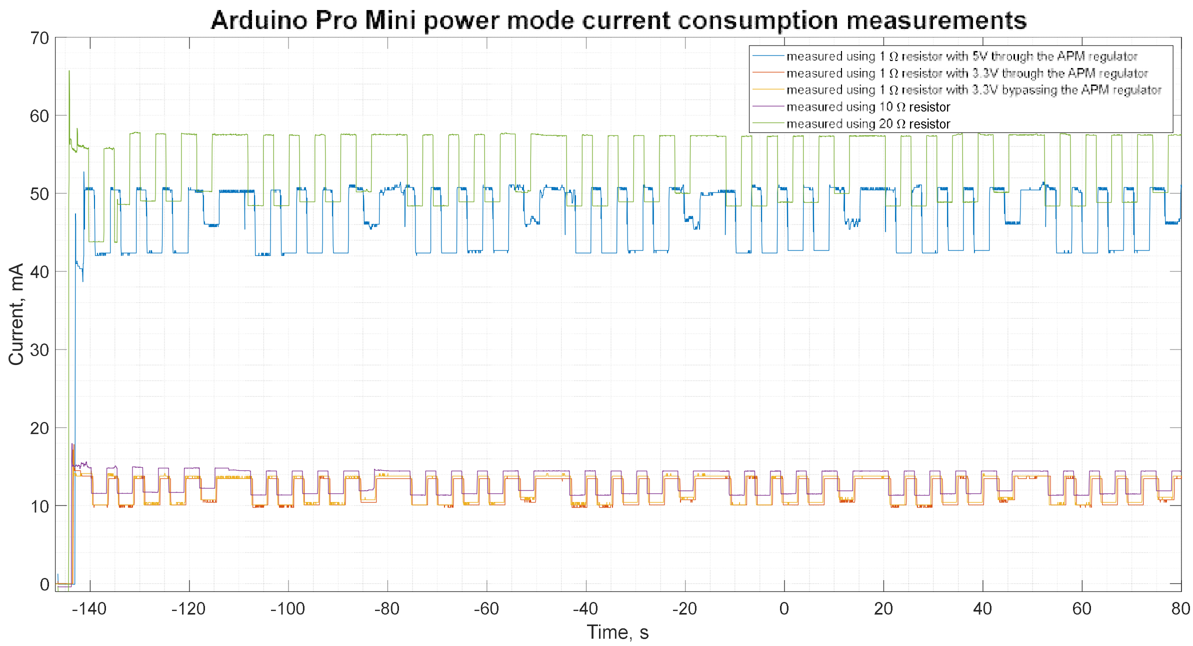


Figure A2 - Arduino Pro Mini current consumption measurements for each power mode operation

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								
Cycle number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Avg**
power on current (mA)	50.76	50.43	51.10	51.10	50.09	50.76	50.43	**50.67**
powerDown current (mA)	42.01	42.01	42.35	42.35	42.35	42.35	42.35	**42.25**
powerSave current (mA)	42.01	42.35	42.35	42.35	42.35	42.35	42.35	**42.30**
powerStandby current (mA)	42.35	42.35	42.69	42.69	42.69	42.69	42.69	**42.59**
powerExtStandby (mA)	42.35	42.35	42.69	42.69	42.69	42.69	42.29	**42.54**
idle (mA)	46.05	45.72	46.05	46.05	46.39	46.39	46.39	**46.15**

1 Ohm Resistor measurement with Regulated voltage (3.3V) on RAW pin								
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	**13.78**
powerDown current (mA)	10.09	10.09	9.76	10.09	10.09	9.76	10.09	**10.00**
powerSave current (mA)	10.09	10.09	9.76	10.09	10.09	9.76	10.09	**10.00**
powerStandby current (mA)	10.41	10.41	10.41	10.41	10.41	10.41	10.41	**10.41**
powerExtStandby (mA)	10.41	10.41	10.41	10.41	10.41	10.41	10.41	**10.41**
idle (mA)	11.09	11.09	10.77	11.09	11.09	10.43	10.47	**10.86**

1 Ohm Resistor measurement with Regulated voltage (3.3V) on Vcc pin								
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	**13.51**
powerDown current (mA)	10.09	10.09	10.09	9.76	10.09	9.76	10.09	**10.00**
powerSave current (mA)	10.09	10.09	10.09	9.76	10.09	9.76	10.09	**10.00**
powerStandby current (mA)	10.09	10.09	10.09	10.09	10.09	10.09	10.09	**10.09**
powerExtStandby (mA)	10.09	10.09	10.09	10.09	10.09	10.09	10.09	**10.09**
idle (mA)	10.43	10.43	10.77	10.77	10.43	10.43	10.77	**10.58**

10 Ohm Resistor measurement								
Cycle number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Avg**
power on current (mA)	15.00	14.60	14.60	14.60	14.60	14.60	14.60	**14.66**
powerDown current (mA)	11.60	11.40	11.40	11.40	11.40	11.40	11.40	**11.43**
powerSave current (mA)	11.60	11.40	11.40	11.40	11.40	11.40	11.40	**11.43**
powerStandby current (mA)	11.70	11.50	11.50	11.50	11.50	11.50	11.50	**11.53**
powerExtStandby (mA)	11.70	11.50	11.50	11.50	11.50	11.50	11.50	**11.53**
idle (mA)	12.20	11.90	11.90	11.90	11.90	11.90	11.90	**11.94**

20 Ohm Resistor measurement								
Cycle number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Avg**
power on current (mA)	14.40	14.40	14.40	14.40	14.40	14.40	14.40	**14.40**
powerDown current (mA)	11.00	12.10	12.10	12.10	12.10	12.10	12.10	**11.94**
powerSave current (mA)	12.10	12.10	12.10	12.10	12.10	12.10	12.10	**12.10**
powerStandby current (mA)	12.20	12.20	12.20	12.20	12.20	12.20	12.20	**12.20**
powerExtStandby (mA)	12.20	12.20	12.20	12.20	12.20	12.20	12.20	**12.20**
idle (mA)	12.50	12.50	12.50	12.50	12.50	12.50	12.50	**12.50**

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.

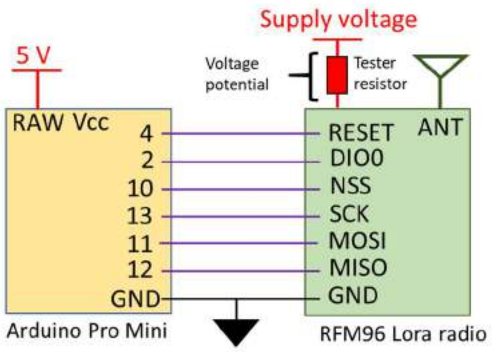


Figure B1- LoRa radio testing configuration and connections

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.

Results

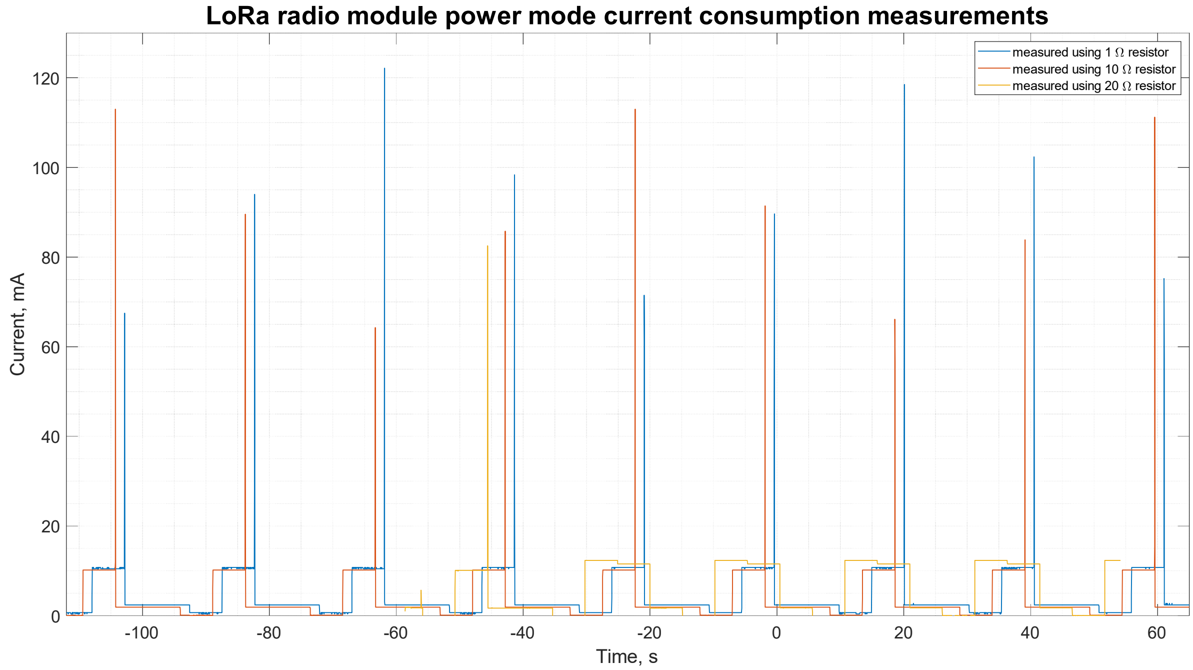


Figure B1 - RFM96 LoRa module current consumption measurements during each mode of operation

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

RFM96 LoRa radio module power mode current consumption measurements										
1 Ohm Resistor measurement										
Cycle number	**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	**0.00070**
Recieve current (A)	0.011	0.011	0.011	0.011	0.011	0.011	0.011	0.011	0.011	**0.011**
No data transmit current (A)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	**0.002**
Idle current (A)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	**0.002**
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	**0.00062**
Recieve current (A)	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0.010	**0.010**
No data transmit current (A)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	**0.002**
Idle current (A)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	**0.002**
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	**-0.00009**
Recieve current (A)				0.010	0.012	0.012	0.012	0.012	0.012	**0.012**
No data transmit current (A)				0.002	0.020	0.001	0.001	0.001	0.001	**0.004**
Idle current (A)				0.002	0.002	0.002	0.002	0.002	0.002	**0.002**

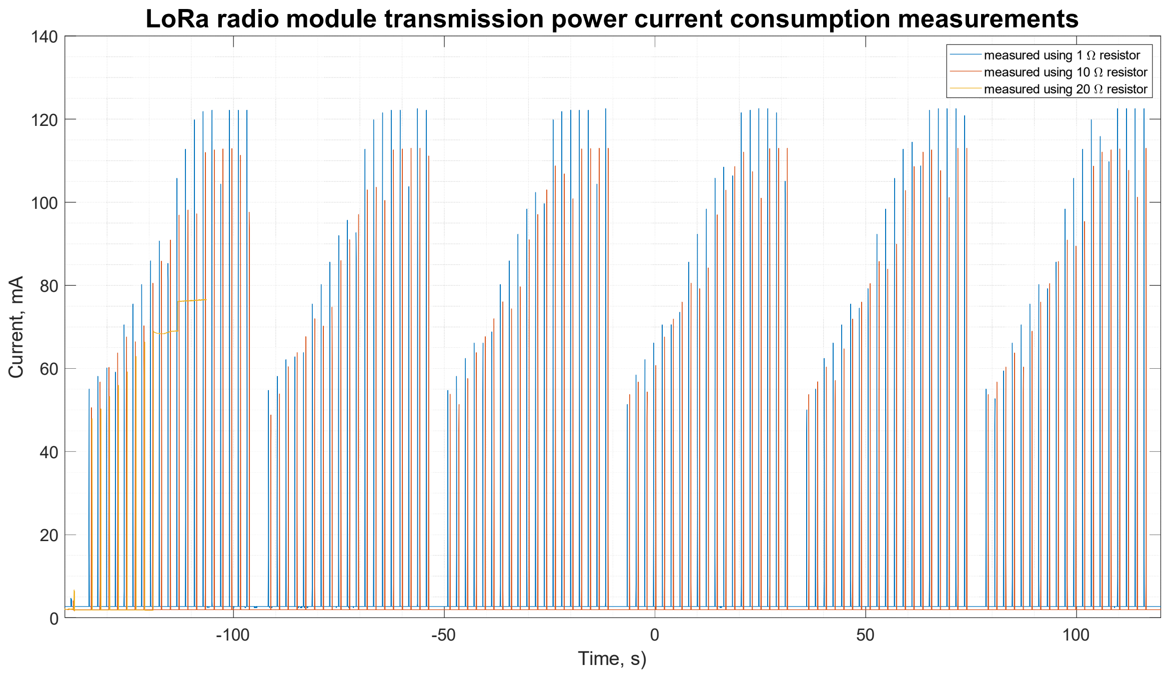


Figure B2 - RFM96 LoRa module current consumption measurements for each transmit power available

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

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

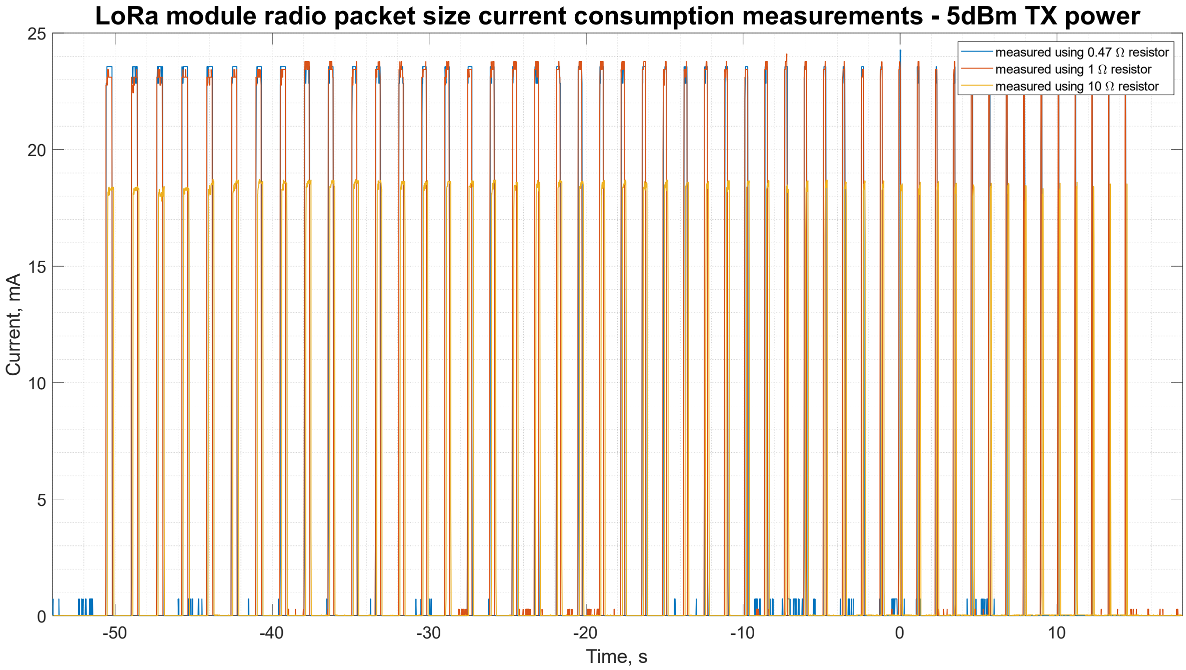


Figure B3 - RFM96 LoRa radio module transmission time for a 5-byte incremented radio packet with 5dBm Tx power

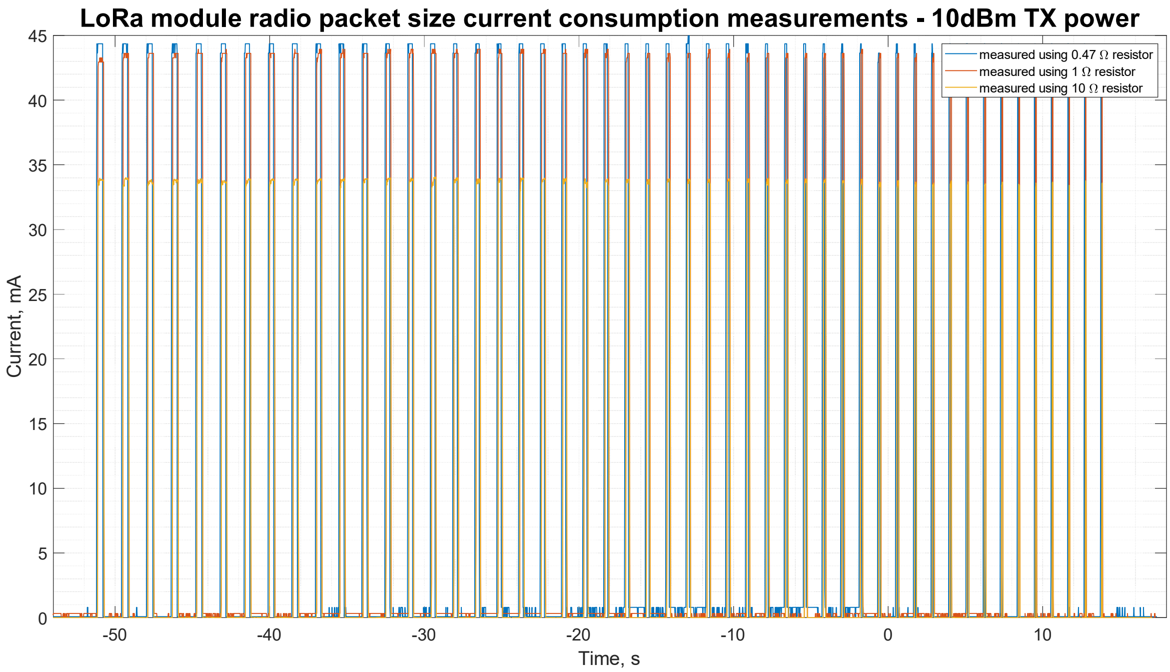


Figure B4 - RFM96 LoRa radio module transmission time for a 5-byte incremented radio packet with 10dBm Tx power

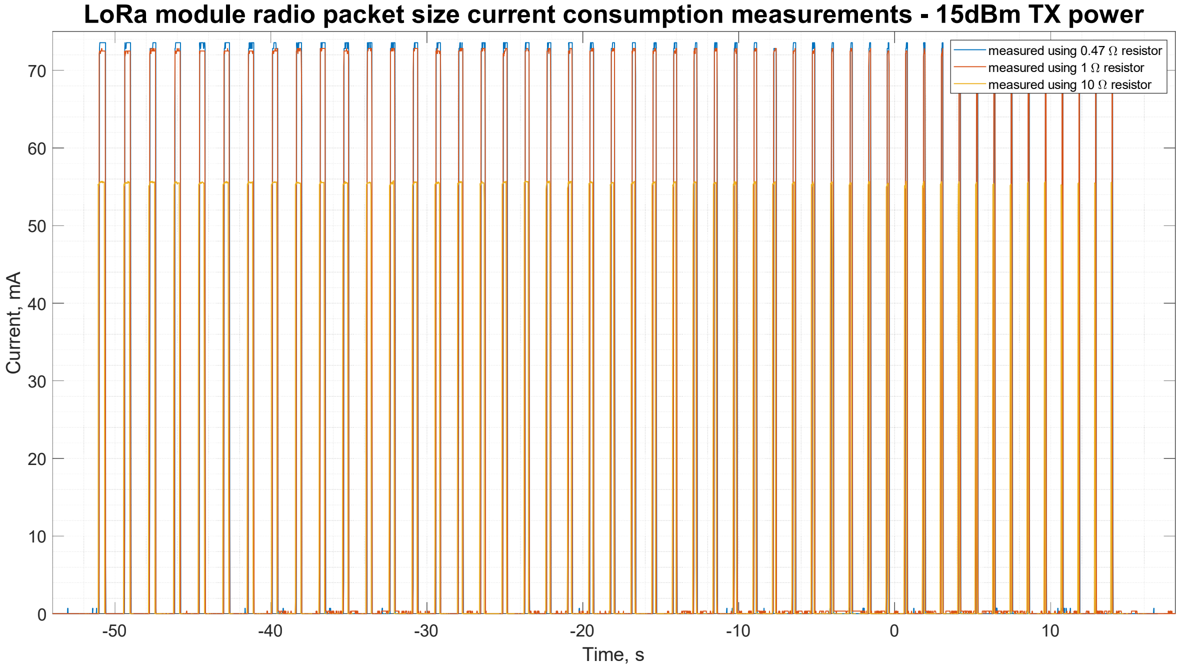


Figure B5 - RFM96 LoRa radio module transmission time for a 5-byte incremented radio packet with 15dBm Tx power

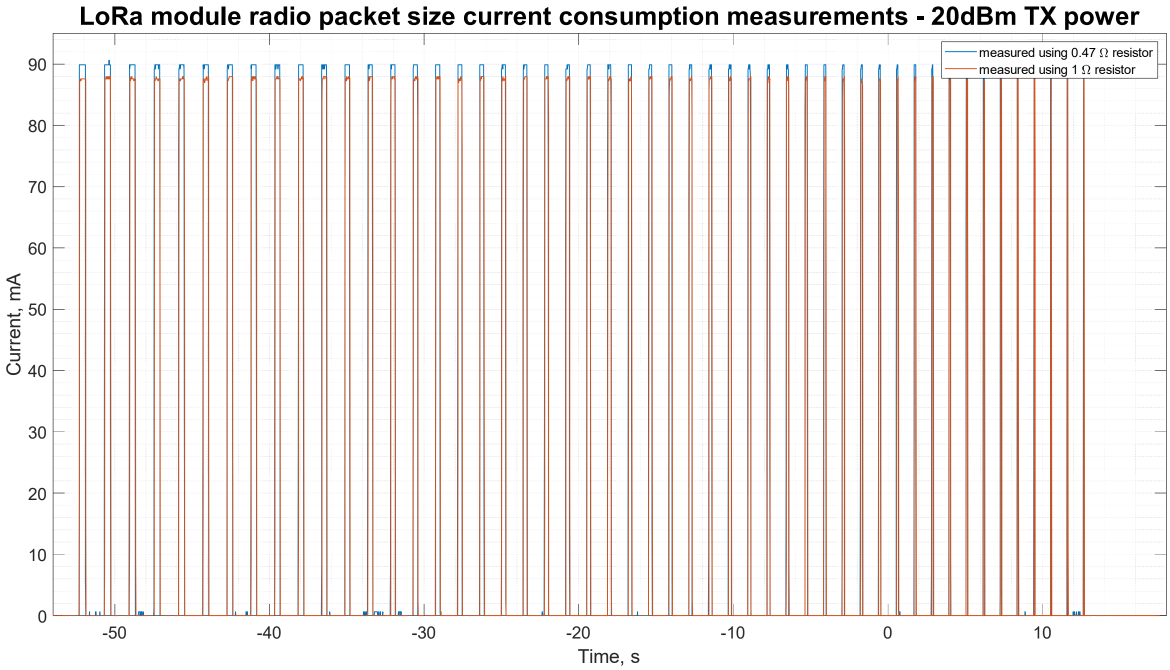


Figure B6 - RFM96 LoRa radio module transmission time for a 5-byte incremented radio packet with 20dBm Tx power

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

RFM 96 LoRa Radio tranmsission 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		

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.

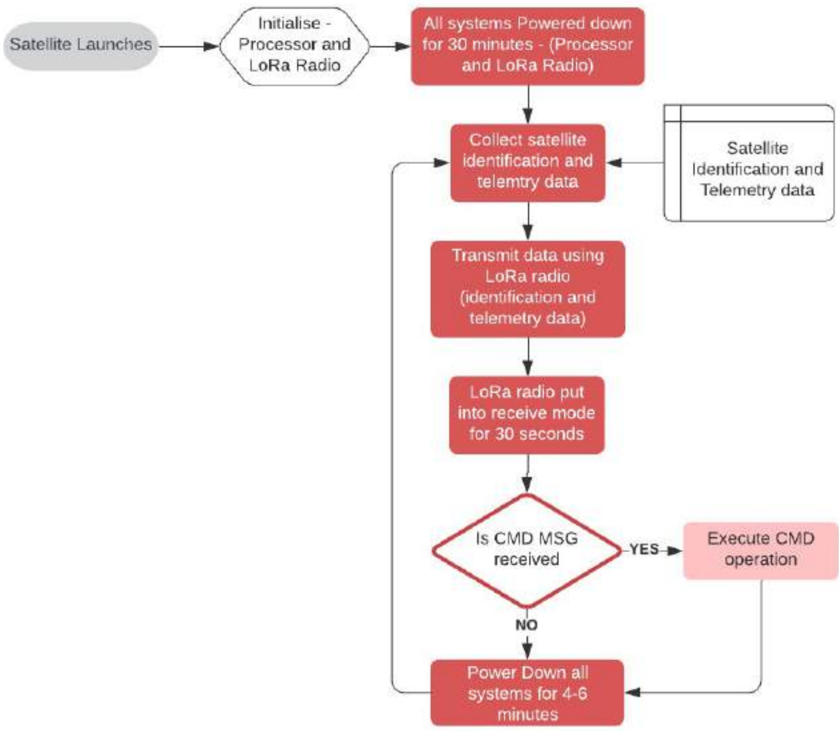


Figure C1 - Satellite radio beacon operational cycle software flow chart

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 1Ω and 2Ω 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.

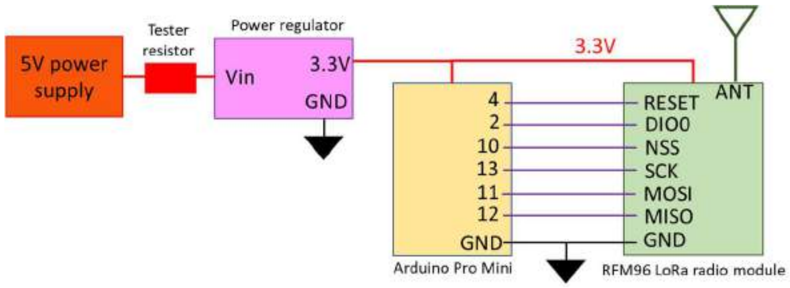


Figure D1 - Satellite beacon power regulator testing configuration

Results

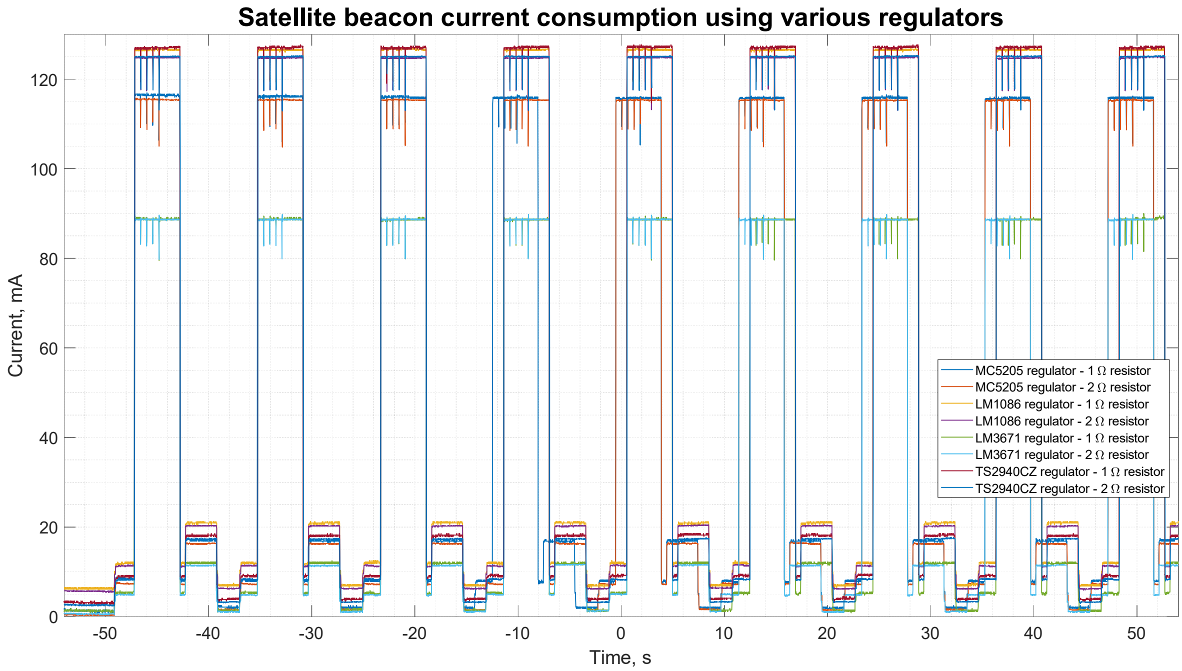


Figure D2 - Satellite beacon current consumption when utilising different voltage regulators

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

Satellite beacon current consumption using different regulators																			
MC5205 in-built regulator - 1 Ohm resistor										**MC5205 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	126.70	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	20.89	20.88	20.89	**20.89**	Receive mode (mA)	20.34	20.34	20.17	20.34	20.34	20.34	20.18	20.17	**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	11.46	**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 supplies 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.

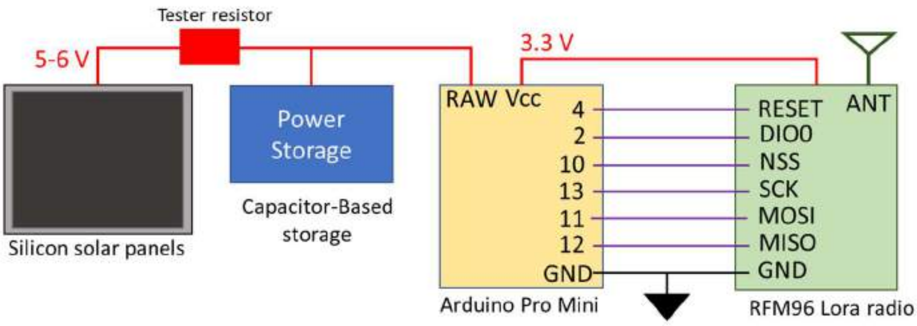


Figure E1 - Satellite radio beacon configuration for power requirement and generation testing

Results

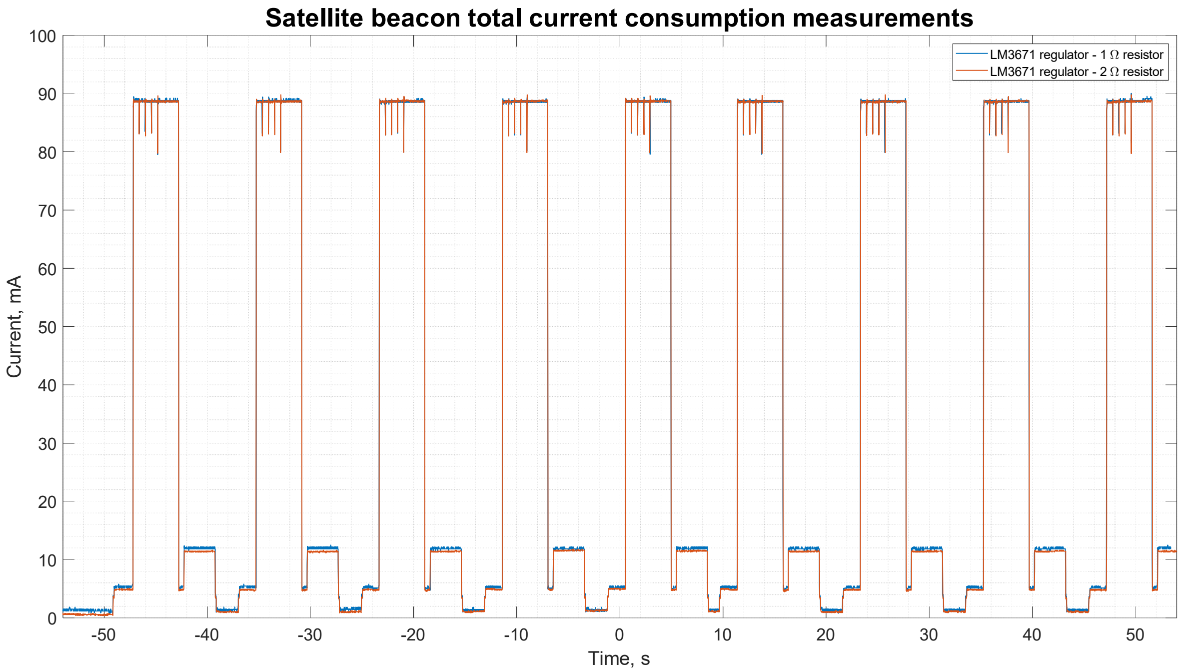


Figure E2 - Satellite beacon total current consumption during each phase of the software cycle

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

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.

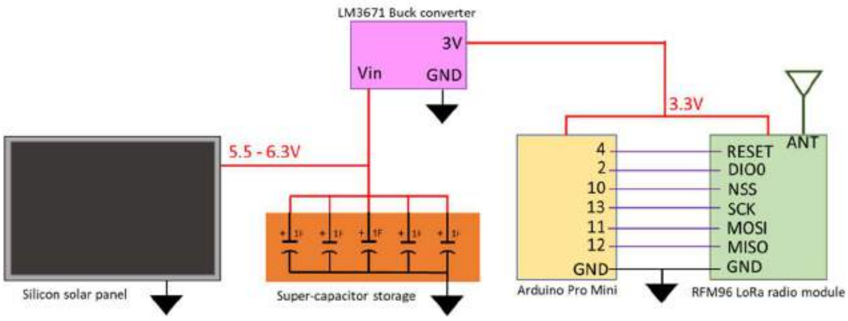


Figure F1 - Satellite radio beacon final configuration for testing

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.

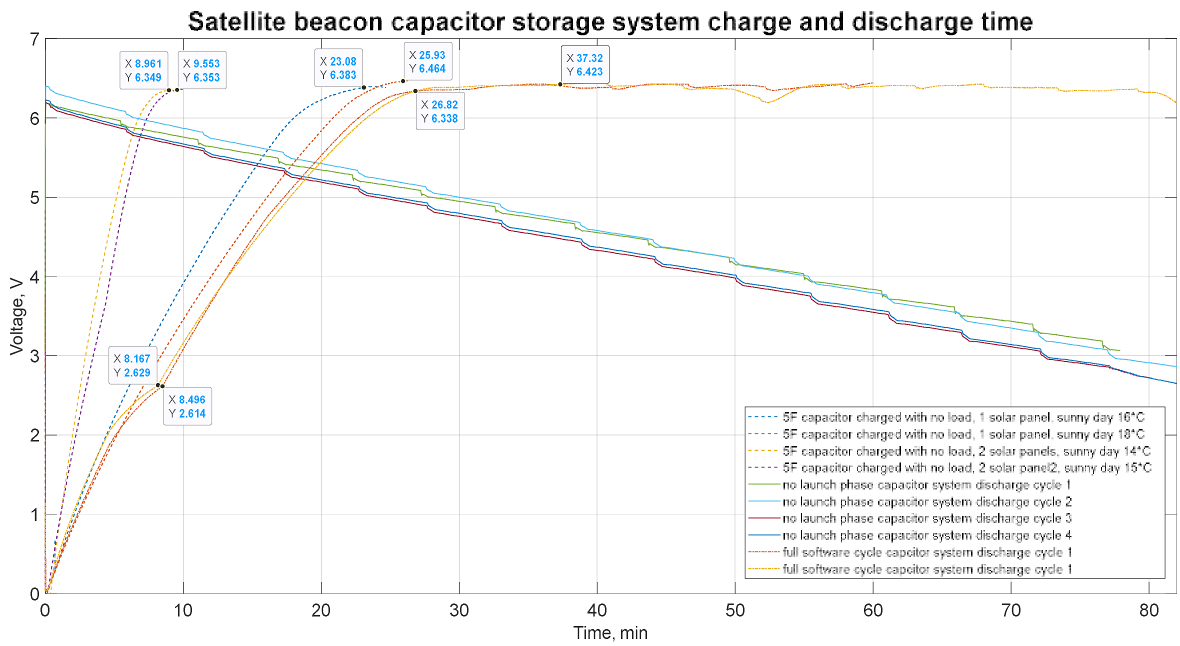


Figure F2 - Satellite beacon power storage charge and discharge voltage with 5F super-capacitor system

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.

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 receiver 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	4/8
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 mddule), 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 *packer 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	4/8
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	4/8
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

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

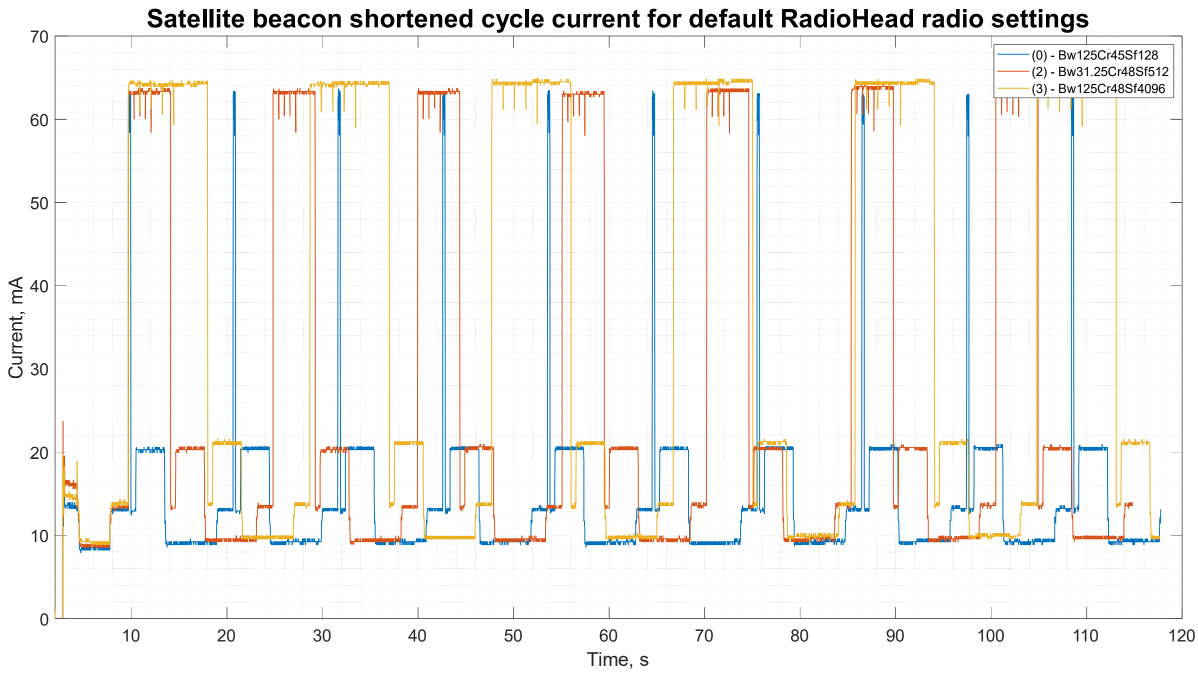


Figure G2 - Satellite beacon current consumption for the default RadioHead radio settings

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

Transmission time for default (0) RadioHead radio settings						
Cycle	Identification packet number time (S)				Telemetry time (S)	Total time (S)
1	0.03	0.03	0.04	0.03	0.10	0.24
2	0.03	0.03	0.04	0.03	0.09	0.22
3	0.03	0.04	0.04	0.03	0.09	0.23
4	0.03	0.03	0.03	0.04	0.10	0.23
5	0.04	0.03	0.03	0.04	0.10	0.24
6	0.03	0.03	0.04	0.03	0.09	0.22
7	0.03	0.04	0.04	0.04	0.08	0.23
Avg	0.03	0.03	0.04	0.03	0.09	0.23

Transmission time for default (2) RadioHead radio settings						
Cycle	Identification packet number time (S)				Telemetry time (S)	Total time (S)
1	0.58	0.60	0.59	0.60	2.02	4.39
2	0.58	0.60	0.58	0.60	2.02	4.38
3	0.57	0.60	0.58	0.60	2.03	4.38
4	0.57	0.60	0.60	0.59	2.02	4.38
5	0.57	0.60	0.60	0.60	2.01	4.38
6	0.59	0.58	0.60	0.60	2.03	4.40
7	0.60	0.60	0.60	0.60	2.00	4.40
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	8.27
2	1.17	1.20	1.19	1.19	3.53	8.28
3	1.21	1.19	1.18	1.19	3.53	8.30
4	1.17	1.19	1.19	1.20	3.53	8.28
5	1.18	1.19	1.19	1.18	3.53	8.27
6	1.20	1.20	1.10	1.20	3.60	8.30
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	Reciever 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	-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.

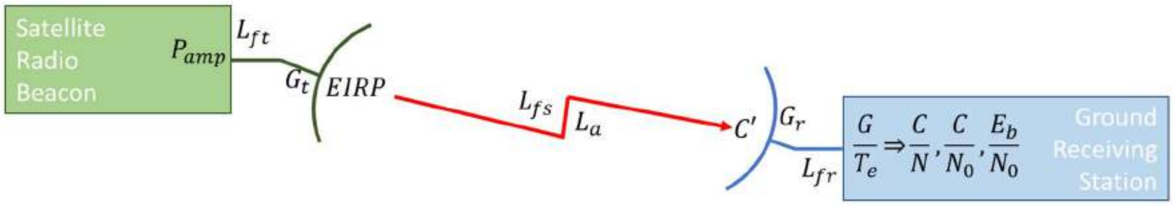


Figure H1 - Satellite radio beacon communications link budget representation

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

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)

Free Space Loss (L_{FS}) [dB] – L_{FS} = 92.44+20*log($d_s.f$)
d_s – slant range (kms)
f – frequency (GHz)

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)

Equivalent Noise Temperature (Te) [dBK] – T_e = 10*log(T*(NF-1))
T – Receiver Environmental temperature (K)
NF – Receiver Noise Figure

Receiver G/T_e (G/T_e) [dBK^{-1}] – 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…

C/N0 ratio [dB] – C/No = $C' - 10.\log(k) + \frac{G}{Te} - L_{fr}$
C' – Carrier Power Density (dBW)
L_{fr} – Receiver feeder Loss (dB)
k – Boltzmann's constant (1.36*$10^{-23}\ Jk^{-1}$)

Eb/No Ratio [dB] – Eb/No = C/No – 10log(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.

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

G_r – Receiver Antenna Gain (dB)
T_e – Equivalent Noise Temperature (dBK)
L_{fr} – Receiver feeder Loss (dB)
k – Boltzmann's constant (1.36*10^{-23} Jk^{-1})

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

SF – Spreading factor
BW – Bandwidth (Hz)

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

R_b – Raw bit rate (bit/s)
FEC – Forward error correction code rate

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

R_b – Raw bit rate (bit/s)
BW – Bandwidth (Hz)

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

N_0- Noise density (dBw/Hz)
BW – Bandwidth (Hz)

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

C' - Carrier Power Density (dBw)
N – Total noise power (dBw)

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)

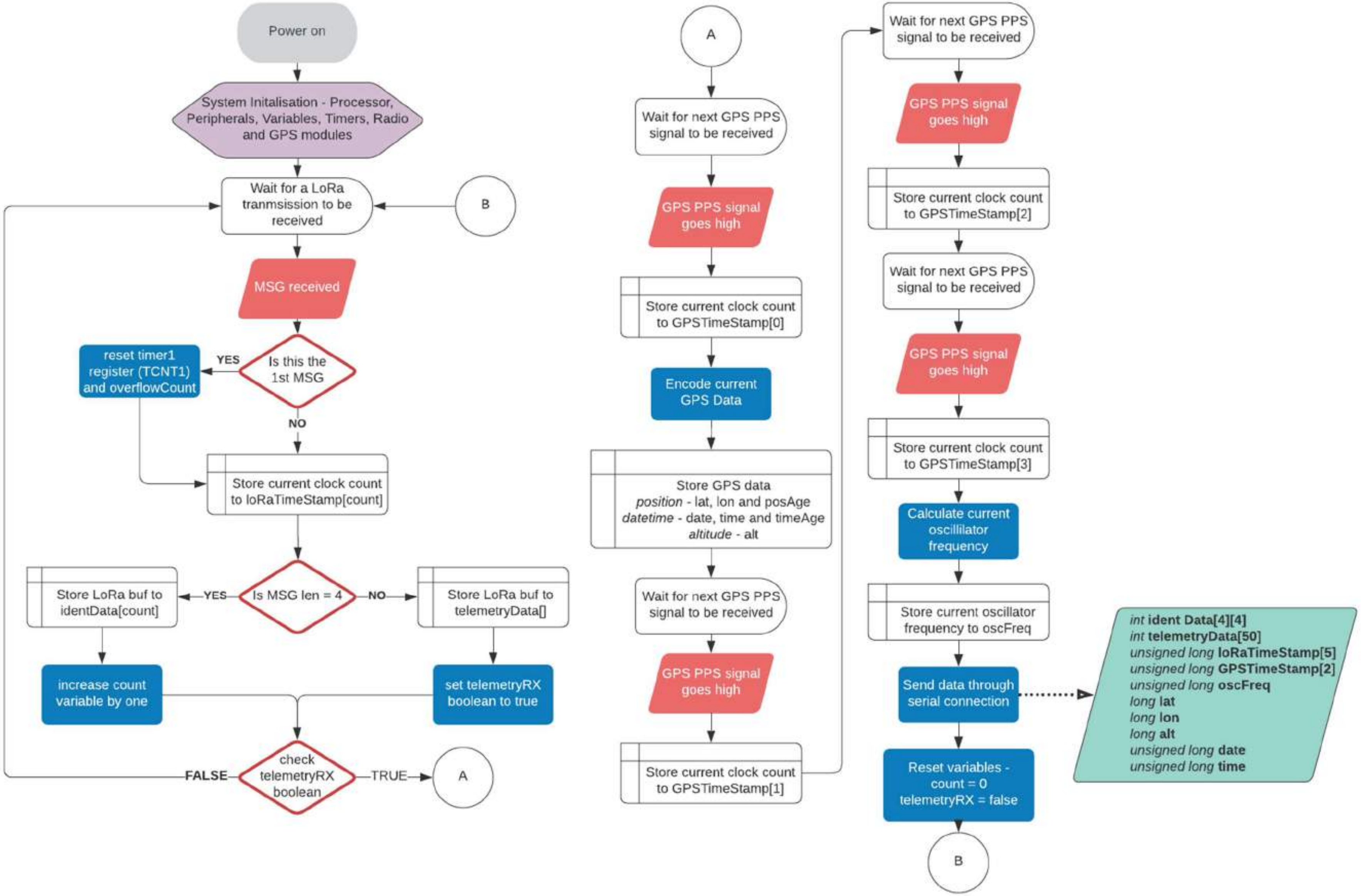
Results

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**	**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**	**η = 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

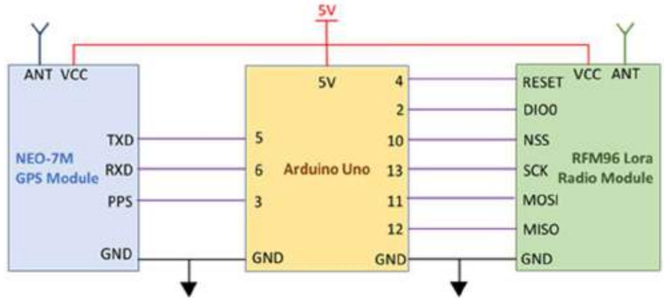


Figure J1 - Ground receiving station configuration and connections

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 4.096µs 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 1.4375µs 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.

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* interupt). 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 then 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)

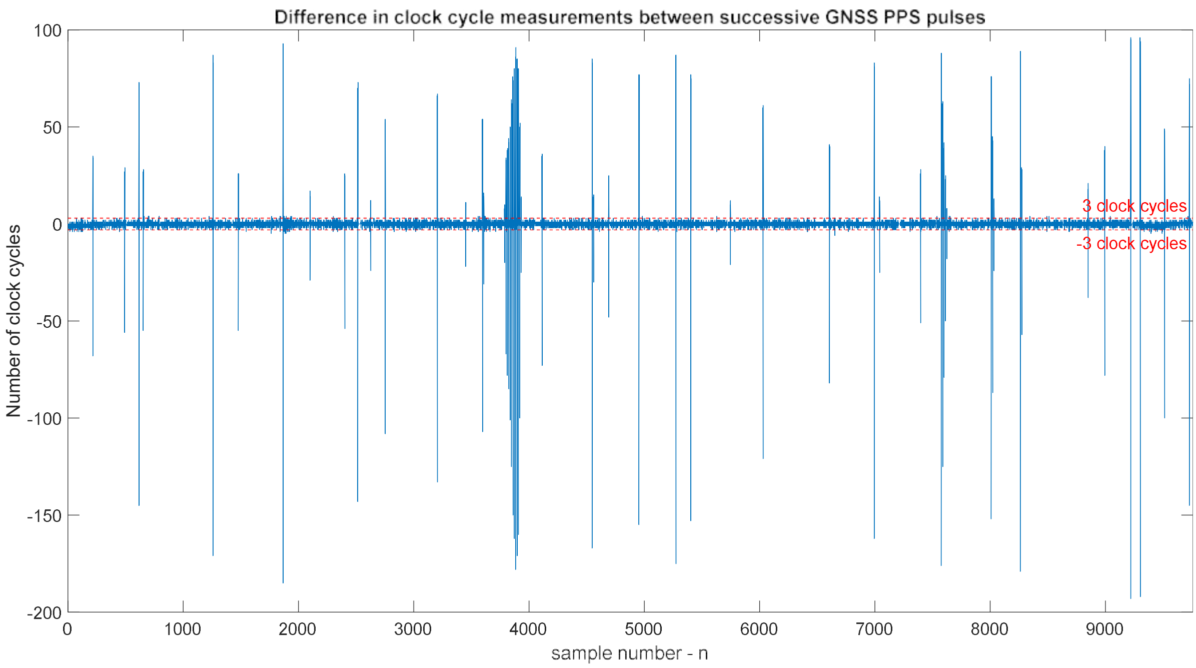


Figure J2 - Difference in clock cycle measurements between successive GNSS PPS pulses

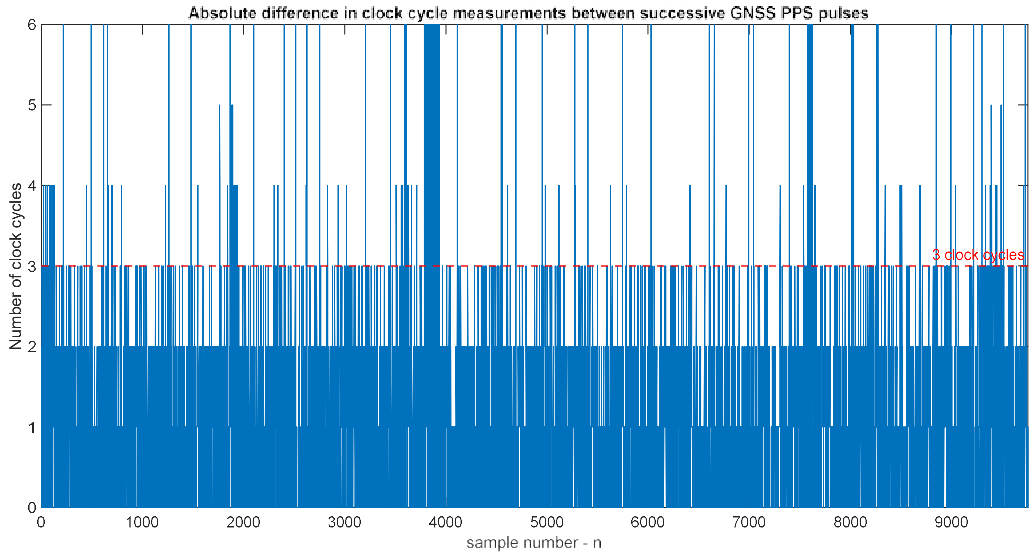


Figure J3 - Higher resolution absolute difference in clock cycles at the 3-cycle threshold

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)

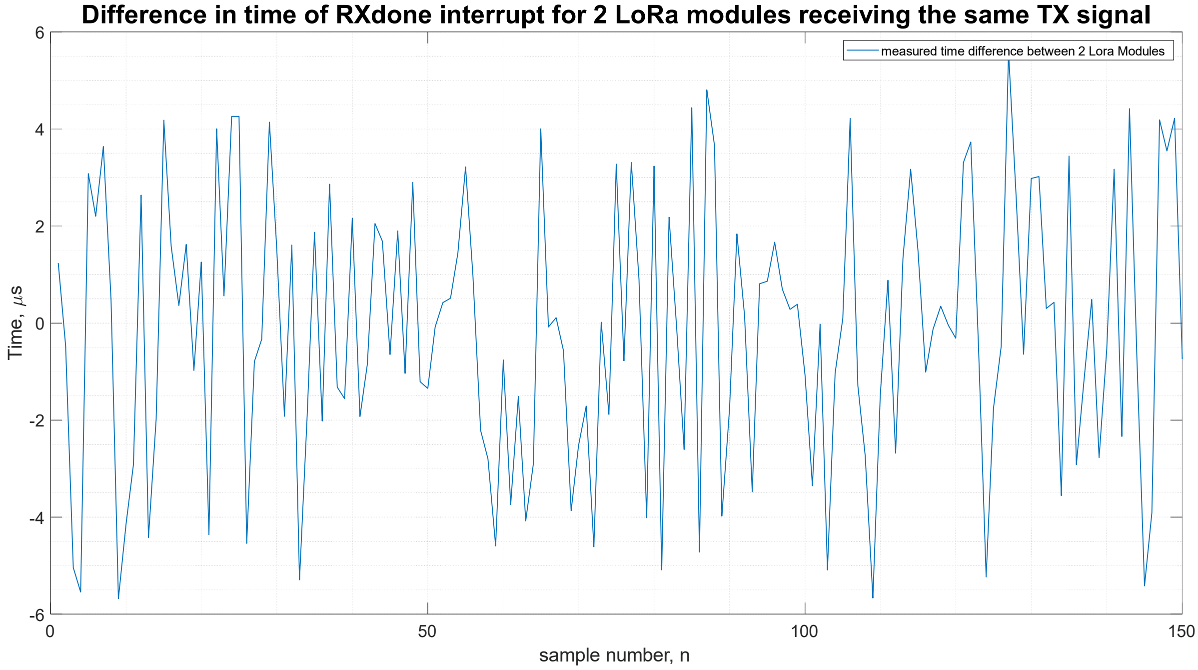


Figure J4 - Time differences for the RXdone interrupt between 2 LoRa module from a single TX source

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 2.2901µs, a standard deviation of 1.6042µs and a maximum difference of 5.68µs with 95% of the data being between 0 and 5.4985µs. When the statistics calculations are performed on the measured data the resultant mean is -0.1803µs with a standard deviation of 2.7966µs which results in 95% of the measured data being between -5.7735µs and 5.4129µs. Using the Cumulative Distribution Function of the measured data has found that 92.56% of the data lies between ±5µs, 96.56% of the data lies between ±5.5µs and 96.77% lies between ±6µs.

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

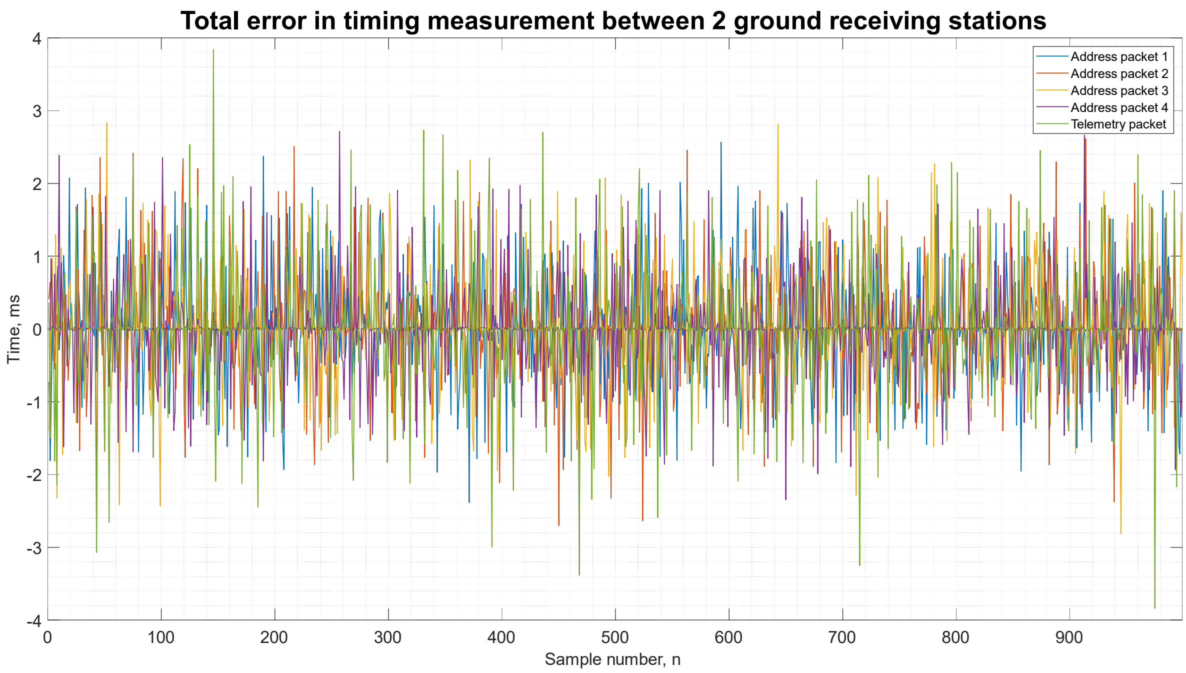


Figure K1 - Total error in timing measurement between 2 ground receiving stations

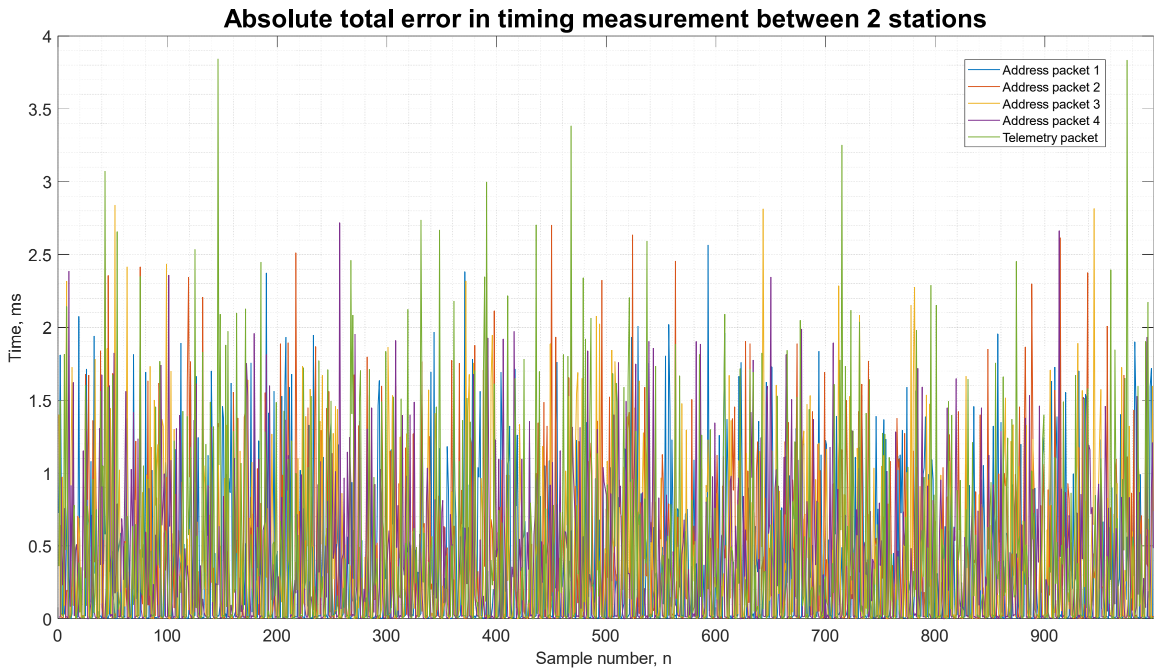


Figure K2 - Absolute total error in timing measurement between 2 ground receiving stations

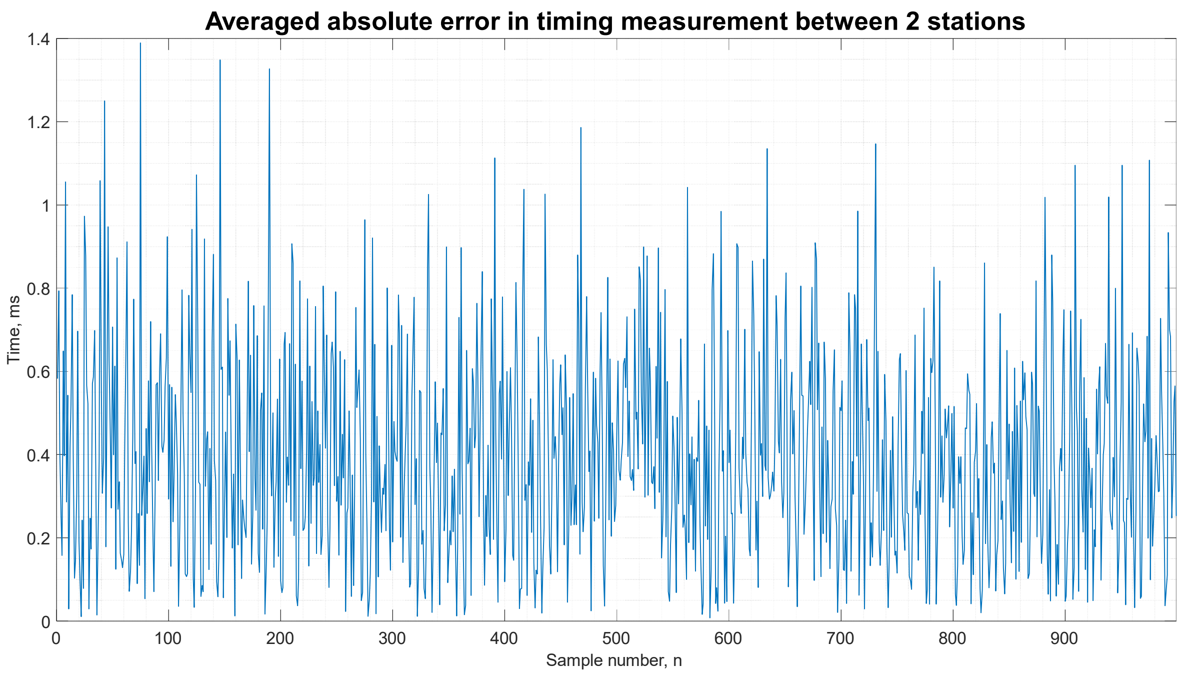


Figure K3 – Averaged absolute total error in timing measurement between 2 ground receiving stations

Table 9 - Total error in measurement statistics

	Measured values				Absolute values			
	mean (µs)	**Std Dev (µs)**	**95% of data interval (ms)**		**mean (µs)**	**Std Dev (µs)**	**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