4 - Satellite Software Test Plan

# Changelog

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author(s) | Confirmation Signature(s) | Description |
| 03/05/2019 | Richard Bamford | A drawing of a person  Description automatically generated | * Defined initial test plan with basic communications system tests and EEPROM tests. |
| 04/05/2019 | Richard Bamford |  | * Layout improvements. |
| 05/05/2019 | Richard Bamford |  | * Completed communication system tests. * Defined other systems outlined with bullet points. * EEPROM tests changed from tables to headers. |
| 05/05/2019 | Richard Bamford |  | Changelog added |
| 06/05/2019 | Richard Bamford |  | * Main program tests added. * Deployment sequence test added. * Hardware interface tests added. * Power control tests added. * Safety and security tests added * Automatic interval control tests added. * Main program test scripts added. * Deployment sequence test scripts added. * Power control test scripts added. * Automatic interval control test scripts added. * Safety and security test scripts added. * Hardware interface test scripts added. |
| 08/05/2019 | Richard Bamford, Gary Bamford |  | * Checking transmission times test script. * Checking on radio configured improperly test script. MAINPROGT3 |

Contents

[Changelog 1](#_Toc8737326)

[Document Key 5](#_Toc8737327)

[Document References 6](#_Toc8737328)

[Description 8](#_Toc8737329)

[Main Program 9](#_Toc8737330)

[MAINPROGT1 – Configuration and Global Variable initialization. 9](#_Toc8737331)

[MAINPROGT2 – Test Pin configuration is correct. 9](#_Toc8737332)

[MAINPROGT3 – Test radio error modes. 9](#_Toc8737333)

[MAINPROGT4 – Confirm Atmega328p has enough memory for the software. 10](#_Toc8737334)

[MAINPROGT5 – Check that the satellite is transmitting. 11](#_Toc8737335)

[MAINPROGT6 – Check that the satellite can receive transmissions. 11](#_Toc8737336)

[MAINPROGT7 – Check that the radio re-configures successfully. 11](#_Toc8737337)

[MAINPROGT8 – Check that the atmega328p can be delayed by up to 10 minutes. 11](#_Toc8737338)

[MAINPROGT9 – Check that the atmega328p can run the software system for at least a week. 12](#_Toc8737339)

[MAINPROGT10 – Check that the hardware watchdog is signalled correctly to keep the satellite alive. 12](#_Toc8737340)

[MAINPROGT11 – Check that the hardware watchdog does not switch the satellite off during custom delays. 12](#_Toc8737341)

[Deployment 13](#_Toc8737342)

[DEPLOYT1 – Test deployment sequence in scenarios; debugging, integration and Jettison. 13](#_Toc8737343)

[DEPLOYT2 – Test that the deployment sequence writes to EEPROM. 13](#_Toc8737344)

[DEPLOYT3 – Test that the deployment sequence does not run on power on if EEPROM address is set. 13](#_Toc8737345)

[DEPLOYT4 – Test that the deployment sequence mosfet enable time is suitable. 13](#_Toc8737346)

[DEPLOYT5 – Confirm that a power information packet is send before the deployment sequence is executed. 14](#_Toc8737347)

[DEPLOYT6 – Check that the deployment sequence transmits a success transmission. 14](#_Toc8737348)

[DEPLOYT7 – Check that the deployment system returns the correct value for its state. 14](#_Toc8737349)

[Power Control 15](#_Toc8737350)

[POWCONT1 – Check the battery charging boolean changes between true and false depending on the current battery temperature > 0 and < 0 Celsius. 15](#_Toc8737351)

[POWCONT2 – Check the battery charging circuit (MPPT) is switched off and on correctly. 15](#_Toc8737352)

[Automatic Interval Control 16](#_Toc8737353)

[AUTOINTT1 - Test that the intervals returned matches the given table of delay seconds. 16](#_Toc8737354)

[AUTOINTT2 – Test that maximum and minimum intervals do not cause errors. 16](#_Toc8737355)

[Hardware/Pin Interface 17](#_Toc8737356)

[HARDINT1 - Test bounds and readings for voltage reading maps 17](#_Toc8737357)

[HARDINT2 - Test that the battery voltage is read when disabled MPPT 17](#_Toc8737358)

[HARDINT3 - Test that the battery charging voltage is read when enabled MPPT 17](#_Toc8737359)

[HARDINT4 - Test that the satellite restarts when hardware watchdog signal is stopped. 17](#_Toc8737360)

[HARDINT5 - Test that the battery charging voltage reading is correct. 17](#_Toc8737361)

[HARDINT6 - Test that the battery voltage reading is correct. 18](#_Toc8737362)

[HARDINT7 - Test that the Solar cell voltage readings are correct. 18](#_Toc8737363)

[HARDINT8 - Test that the battery temperature reading is correct. 18](#_Toc8737364)

[HARDINT9 - Test that the board temperature reading is correct. 18](#_Toc8737365)

[HARDINT10 -Test that the watchdog heartbeat can be enabled and disabled. 18](#_Toc8737366)

[HARDINT11 - Test that the temperature resolution for the sensors works. 19](#_Toc8737367)

[Safety & Security 20](#_Toc8737368)

[SAFESECT1 - Test the satellite correctly handles lora radio codes. 20](#_Toc8737369)

[SAFESECT2 – Test the string check function returns false for invalid strings. 20](#_Toc8737370)

[SAFESECT3 – Test the deployment state eeprom values for errors and error handling. 20](#_Toc8737371)

[Communication 22](#_Toc8737372)

[COMMST1 - Testing the enable and disable transmission state 22](#_Toc8737373)

[COMMST2 – Testing the repeater 22](#_Toc8737374)

[COMMST3 – Testing the Ping-Pong feature 22](#_Toc8737375)

[COMMST4 – Testing the programmable callsign 23](#_Toc8737376)

[COMMST5 – Testing the transmission password protection 23](#_Toc8737377)

[COMMST6 – Testing the notification of satellite power on. 23](#_Toc8737378)

[COMMST7 – Testing the notification of satellite deployment. 23](#_Toc8737379)

[COMMST8 – Testing the eeprom reset message. 23](#_Toc8737380)

[COMMST9 – Testing the restart command 23](#_Toc8737381)

[COMMST10 – Testing the frequency band switching 24](#_Toc8737382)

[COMMST11 – Testing the manual deployment command 24](#_Toc8737383)

[COMMST12 – Testing the System Information command 24](#_Toc8737384)

[COMMST13 – Testing the notification of satellite shutting down. 24](#_Toc8737385)

[COMMST14 – Checking transmission times. 25](#_Toc8737386)

[Persistent Storage 26](#_Toc8737387)

[PT1 - Write 0 to eeprom at address 0 26](#_Toc8737388)

[PT2 - Write 1 to eeprom at address 0 26](#_Toc8737389)

[PT3 - Write 32767 to eeprom at address 0 26](#_Toc8737390)

[PT4 - Write -32768 to eeprom at address 0 26](#_Toc8737391)

[PT5 - Write 60000 to eeprom at address 0 26](#_Toc8737392)

[PT6 - Write -60000 to eeprom at address 0 27](#_Toc8737393)

[PT7 - Write 0 to eeprom at address 100 27](#_Toc8737394)

[PT8 - Write 1 to eeprom at address 100 27](#_Toc8737395)

[PT9 - Write 32767 to eeprom at address 100 27](#_Toc8737396)

[PT10 - Write -32768 to eeprom at address 100 27](#_Toc8737397)

[PT11 - Write 60000 to eeprom at address 100 28](#_Toc8737398)

[PT12 - Write -60000 to eeprom at address 100 28](#_Toc8737399)

[PT13 – Write 10 to EEPROM at address 0 and then read from EEPROM at address 0 28](#_Toc8737400)

# Document Key

Main Program – MAINPROG

Deployment – DEPLOY

Power Control – POWCONT1

Automatic Interval Control – AUTOINT

Hardware and Pin Interface – HARDINT

Safety & Security – SAFESECT

Communication – COMMS

Persistent Storage - PT

# Document References

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Specification ID | ID | Date Set | Date Completed | Tester Signature | Witness Signature |
|  | MAINPROGT1 | 05/05/2019 |  |  |  |
|  | MAINPROGT2 | 05/05/2019 |  |  |  |
|  | MAINPROGT3 | 05/05/2019 |  |  |  |
|  | MAINPROGT4 | 05/05/2019 |  |  |  |
|  | MAINPROGT5 | 05/05/2019 |  |  |  |
|  | MAINPROGT6 | 05/05/2019 |  |  |  |
|  | MAINPROGT7 | 05/05/2019 |  |  |  |
|  | MAINPROGT8 | 05/05/2019 |  |  |  |
|  | MAINPROGT9 | 06/05/2019 |  |  |  |
|  | MAINPROGT10 | 06/05/2019 |  |  |  |
|  | MAINPROGT11 | 06/05/2019 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Specification ID | ID | Date Set | Date Completed | Tester Signature | Witness Signature |
| COMMFS14, COMMFS4, COMMFS5 | COMMST1 | 04/05/2019 |  |  |  |
| COMMFS7, COMMFS15 | COMMST2 | 04/05/2019 |  | . |  |
| COMMFS2 | COMMST3 | 05/05/2019 |  |  |  |
| COMMFS3 | COMMST4 | 05/05/2019 |  |  |  |
| COMMFS6 | COMMST5 | 05/05/2019 |  |  |  |
| COMMFS1 | COMMST6 | 05/05/2019 |  |  |  |
| COMMFS13 | COMMST7 | 05/05/2019 |  |  |  |
| COMMFS10 | COMMST8 | 05/05/2019 |  |  |  |
| COMMFS11 | COMMST9 | 05/05/2019 |  |  |  |
| COMMFS16 | COMMST10 | 05/05/2019 |  |  |  |
| COMMFS17 | COMMST11 | 05/05/2019 |  |  |  |
| COMMFS9 | COMMST12 | 05/05/2019 |  |  |  |
| COMMFS12 | COMMST13 | 05/05/2019 |  |  |  |
|  | COMMST14 | 08/05/2019 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Specification ID | ID | Date Set | Date Completed | Tester Signature | Witness Signature |
|  | DEPLOYT1 | 05/05/2019 |  |  |  |
|  | DEPLOYT2 | 05/05/2019 |  |  |  |
|  | DEPLOYT3 | 05/05/2019 |  |  |  |
|  | DEPLOYT4 | 05/05/2019 |  |  |  |
|  | DEPLOYT5 | 05/05/2019 |  |  |  |
|  | DEPLOYT6 | 05/05/2019 |  |  |  |
|  | DEPLOYT7 | 05/05/2019 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Specification ID | ID | Date Set | Date Completed | Tester Signature | Witness Signature |
|  | POWCONT1 | 06/05/2019 |  |  |  |
|  | POWCONT2 | 06/05/2019 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Specification ID | ID | Date Set | Date Completed | Tester Signature | Witness Signature |
|  | AUTOINTT1 | 06/05/2019 |  |  |  |
|  | AUTOINTT2 | 06/05/2019 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Specification ID | ID | Date Set | Date Completed | Tester Signature | Witness Signature |
|  | HARDINT1 | 05/05/2019 |  |  |  |
|  | HARDINT2 | 05/05/2019 |  |  |  |
|  | HARDINT3 | 05/05/2019 |  |  |  |
|  | HARDINT4 | 05/05/2019 |  |  |  |
|  | HARDINT5 | 05/05/2019 |  |  |  |
|  | HARDINT6 | 05/05/2019 |  |  |  |
|  | HARDINT7 | 06/05/2019 |  |  |  |
|  | HARDINT8 | 06/05/2019 |  |  |  |
|  | HARDINT9 | 06/05/2019 |  |  |  |
|  | HARDINT10 | 06/05/2019 |  |  |  |
|  | HARDINT11 | 06/05/2019 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Specification ID | ID | Date Set | Date Completed | Tester Signature | Witness Signature |
|  | SAFESECT1 | 06/05/2019 |  |  |  |
|  | SAFESECT2 | 06/05/2019 |  |  |  |
|  | SAFESECT3 | 06/05/2019 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Specification ID | ID | Date Set | Date Completed | Tester Signature | Witness Signature |
| SYSFS12 | PT1 | 04/05/2019 |  |  |  |
| SYSFS12 | PT2 | 04/05/2019 |  |  |  |
| SYSFS12 | PT3 | 04/05/2019 |  |  |  |
| SYSFS12 | PT4 | 04/05/2019 |  |  |  |
| SYSFS12 | PT5 | 04/05/2019 |  |  |  |
| SYSFS12 | PT6 | 04/05/2019 |  |  |  |
| SYSFS12 | PT7 | 04/05/2019 |  |  |  |
| SYSFS12 | PT8 | 04/05/2019 |  |  |  |
| SYSFS12 | PT9 | 04/05/2019 |  |  |  |
| SYSFS12 | PT10 | 04/05/2019 |  |  |  |
| SYSFS12 | PT11 | 04/05/2019 |  |  |  |
| SYSFS12 | PT12 | 04/05/2019 |  |  |  |

# Description

Using an Arduino Uno with a SX1278 Dragino shield, satellite development kit and both connected to a PC. The Received result will be both serial values and transmitted values to check for differences.

# Main Program

### MAINPROGT1 – Configuration and Global Variable initialization.

#### Steps

1. Compile the software using Arduino version 1.8.9 and confirm the compilation is successful.
2. Power on the satellite.
3. Check the serial monitor for anomalies such as; powering off, freezing, malformed transmission messages.

#### Expected Result

1. Compilation successful.
2. LEDS light up and satellite powers on.
3. Satellite loops through the transmissions cycle with no anomalies in the serial monitor or ground station receiver.

#### Real Result

### MAINPROGT2 – Test Pin configuration is correct.

#### Steps

1. Power the satellite on.
2. Run the entire satellite software system.
3. Check that the data received is ok, deployment sequence executed, and hardware being controlled properly.

#### Expected Result

1. Receive a start up transmission.
2. Receive a system information packet and verify its values.
3. Confirm that the deployment hardware is switched correctly.

#### Real Result

### MAINPROGT3 – Test radio error modes.

#### Meaning

If the radio configuration can be configured incorrect but still carry on with normal operation, Is there any point in restarting it? For now we proceed as normal, since restarts can be manually initiated. 08/05/2019

#### Steps

1. Power on the satellite.
2. Wait for 15 transmission loop cycle and receive it with a ground station.
3. Power off the satellite.
4. Power on the satellite.
5. Transmit a PING command to the satellite.
6. Wait for PONG response.
7. Power off the satellite.
8. Power on the satellite.
9. Configure the radio incorrectly.
10. Wait for 15 transmission loop cycles (normal operation).
11. Power off the satellite.
12. Power on the satellite.
13. Transmit a PING command to the satellite.
14. Wait for PONG response.
15. Configure radio pins incorrectly.
16. Wait for 15 transmission loop cycles (normal operation).
17. Power off the satellite.
18. Power on the satellite.
19. Transmit a PING.
20. Wait for PONG.
21. Power off the satellite.

#### Expected Result

1. –
2. Receive messages on LoRa amateur band, LoRa ISM band, RTTY amateur band.
3. –
4. –
5. Satellite to receive the PING command.
6. Reception of a PONG command on a ground station.
7. –
8. –
9. Error code will be detected by the Safety & Security system.
10. No transmissions or receives.
11. –
12. –
13. No PING received by the satellite.
14. No PONG received by the ground station.

#### Real Result

### MAINPROGT4 – Confirm Atmega328p has enough memory for the software.

#### Steps

1. Compile the software using Arduino Ide version 1.8.9.
2. Upload the code to the satellite.
3. Run the entire program from 1 week sending it a cycle of commands.
4. Check for memory errors like freezing and crashing.

#### Command Cycle

1. Transmit callsign change for 1 minute.
2. Transmit repeat message for 5 minutes.
3. Transmit PING for 1 minute.
4. Transmit Restart for 5 minutes.

#### Expected Result

1. Compilation successful.
2. Uploaded correctly to the satellite atmega.
3. Satellite runs with no freezing or errors for a week.

#### Real Result

### MAINPROGT5 – Check that the satellite is transmitting.

#### Steps

1. Power the satellite on.
2. Receive the startup message on the ground station.
3. Receive the RTTY callsign message on the ground station after 10 cycles have completed.

#### Expected Result

1. Startup transmissions received on a ground station.
2. RTTY callsign message received on a ground station.

#### Real Result

### MAINPROGT6 – Check that the satellite can receive transmissions.

#### Steps

1. Power the satellite on.
2. Transmit a PING command for 5 minutes.

#### Expected Result

1. PONG command received on ground station.

#### Real Result

### MAINPROGT7 – Check that the radio re-configures successfully.

#### Steps

1. Power the satellite on.
2. Receive a LoRa ISM transmission, LoRa amateur transmission and FSK transmission.

#### Expected Result

1. Receive transmissions on the ground station.
2. No errors in satellite software.

#### Real Result

### MAINPROGT8 – Check that the atmega328p can be delayed by up to 10 minutes.

#### Steps

1. Program a 10 minute delay and 10 minute keep-alive delay.
2. Compile and upload the software.
3. Run program.

#### Expected Result

1. Satellite waits for 10 minutes with delay, satellite switches off due to hardware watchdog.
2. Satellite waits for 10 minutes with keep-alive delay, satellite stays powered on for 10 minutes.

#### Real Result

### MAINPROGT9 – Check that the atmega328p can run the software system for at least a week.

#### Steps

* MAINPROGT4

#### Expected Result

* MAINPROGT4

#### Real Result

* MAINPROGT4

### MAINPROGT10 – Check that the hardware watchdog is signalled correctly to keep the satellite alive.

#### Steps

* MAINPROGT4

#### Expected Result

* MAINPROGT4

#### Real Result

* MAINPROGT4

### MAINPROGT11 – Check that the hardware watchdog does not switch the satellite off during custom delays.

#### Steps

* MAINPROGT4

#### Expected Result

* MAINPROGT4

#### Real Result

* MAINPROGT4

# Deployment

### DEPLOYT1 – Test deployment sequence in scenarios; debugging, integration and Jettison.

#### Steps

1. Reset satellite EEPROM.
2. Power the satellite on with the ENABLE\_DEPLOYMENT\_SEQUENCE pre-processor definition present.
3. Reset satellite EEPROM.
4. Power the satellite on with the ENABLE\_DEPLOYMENT\_SEQUENCE pre-processor definition removed.

#### Expected Result

1. Deployment sequence skipped and satellite enters into the main loop.
2. Deployment sequence ran.

#### Real Result

### DEPLOYT2 – Test that the deployment sequence writes to EEPROM.

#### Steps

1. Reset the satellite’s EEPROM.
2. Power the satellite on with deployment enabled.
3. Switch the satellite off.
4. Power the satellite on with deployment enabled.

#### Expected Result

1. Deployment sequence runs.
2. –
3. Deployment sequence does not run.

#### Real Result

### DEPLOYT3 – Test that the deployment sequence does not run on power on if EEPROM address is set.

#### Steps

* DEPLOYT2

#### Expected Result

* DEPLOYT2

#### Real Result

* DEPLOYT2

### DEPLOYT4 – Test that the deployment sequence mosfet enable time is suitable.

#### Steps

1. Reset the satellite’s EEPROM.
2. Power the satellite with deployment on.

#### Expected Result

1. Check that the deployment hardware successfully deploys with the given MOSFET enable time of 1 second ON and 1 second OFF.

#### Real Result

### DEPLOYT5 – Confirm that a power information packet is send before the deployment sequence is executed.

#### Steps

1. Reset satellite’s EEPROM.
2. Power the satellite with deployment on.

#### Expected Result

1. Check after >3minutes that a power information packet has been received.

#### Real Result

### DEPLOYT6 – Check that the deployment sequence transmits a success transmission.

#### Steps

1. Power the satellite with deployment on.

#### Expected Result

1. Check that the deployment hardware successfully deploys with the given MOSFET enable time of 1 second ON and 1 second OFF.

#### Real Result

### DEPLOYT7 – Check that the deployment system returns the correct value for its state.

#### Steps

* DEPLOYT2

#### Expected Result

* DEPLOYT2

#### Real Result

* DEPLOYT2

# Power Control

### POWCONT1 – Check the battery charging boolean changes between true and false depending on the current battery temperature > 0 and < 0 Celsius.

#### Steps

1. Power the satellite with simulated environments of sub 0 and greater than 0.

#### Expected Result

1. Check that the satellite charging circuit is switched OFF when below 0.

#### Real Result

### POWCONT2 – Check the battery charging circuit (MPPT) is switched off and on correctly.

#### Steps

1. Power the satellite on.
2. Wait for a system information packet.

#### Expected Result

1. Power information packet should have correct values for both the battery voltage and battery charging voltage.

#### Real Result

# Automatic Interval Control

### AUTOINTT1 - Test that the intervals returned matches the given table of delay seconds.

#### Steps

1. Power the satellite on with a power supply set to the given table of voltages. OR Power the satellite on with a battery charged to specific amounts.
2. Receive 10 system information transmissions.

#### Expected Result

1. Calculate the average time between each transmission and compare is to the interval table. (remember to account for added time delay of transmissions cycle).

#### Real Result

### AUTOINTT2 – Test that maximum and minimum intervals do not cause errors.

#### Steps

1. Set the keep-alive delay value to minimum value in code.
2. Compile and upload.
3. Startup the satellite.
4. Wait for 5 minutes.
5. Set the keep-alive delay value to the maximum value in code.
6. Compile and upload.
7. Startup the satellite.
8. Wait for 5 minutes.

#### Expected Result

1. Satellite running at the minimum interval value without errors.
2. Satellite running at the maximum interval value without errors.

#### Real Result

# Hardware/Pin Interface

### HARDINT1 - Test bounds and readings for voltage reading maps

#### Steps

1. Power satellite on.
2. Wait for transmission loops.

#### Expected Result

1. Check system information packet to confirm correct voltage maps
2. Check system information packet to confirm correct current maps.

#### Real Result

### HARDINT2 - Test that the battery voltage is read when disabled MPPT

#### Steps

1. Power satellite on.
2. Wait for transmission loops.

#### Expected Result

1. Check system information packet to confirm correct battery voltage.

#### Real Result

### HARDINT3 - Test that the battery charging voltage is read when enabled MPPT

#### Steps

1. Power satellite on.

#### Expected Result

1. Check system information packet to confirm correct battery charging voltage.

#### Real Result

### HARDINT4 - Test that the satellite restarts when hardware watchdog signal is stopped.

#### Steps

1. Power satellite on.
2. Transmit a restart command.

#### Expected Result

1. Satellite restarts.

#### Real Result

### HARDINT5 - Test that the battery charging voltage reading is correct.

#### Steps

1. Power satellite on.

#### Expected Result

1. Check system information packet to confirm correct battery charging voltage.

#### Real Result

### HARDINT6 - Test that the battery voltage reading is correct.

#### Steps

* HARDINT2

#### Expected Result

* HARDINT2

#### Real Result

* HARDINT2

### HARDINT7 - Test that the Solar cell voltage readings are correct.

#### Steps

1. Power on the satellite.
2. Wait for transmission system information.

#### Expected Result

1. Confirm the voltage readings for the solar cells are correct in the system information packet.

#### Real Result

### HARDINT8 - Test that the battery temperature reading is correct.

#### Steps

1. Power on the satellite.
2. Wait for transmission system information.

#### Expected Result

1. Confirm the battery temperature reading is correct in the system information packet.

#### Real Result

### HARDINT9 - Test that the board temperature reading is correct.

#### Steps

1. Power on the satellite.
2. Wait for transmission system information.

#### Expected Result

1. Confirm the board temperature reading are correct in the system information packet.

#### Real Result

### HARDINT10 -Test that the watchdog heartbeat can be enabled and disabled.

#### Steps

* HARDINT4

#### Expected Result

* HARDINT4

#### Real Result

* HARDINT4

### HARDINT11 - Test that the temperature resolution for the sensors works.

#### Steps

1. Power on the satellite.
2. Wait for transmission system information.

#### Expected Result

1. Confirm the temperature readings are to 2 decimal places.

#### Real Result

# Safety & Security

### SAFESECT1 - Test the satellite correctly handles lora radio codes.

#### Steps

1. Power the satellite on with radio pins correct but antenna disconnected.
2. Run satellite for 5 minutes.
3. Power the satellite on with radio pins correct and antenna connected.
4. Run satellite for 5 minutes.
5. Power the satellite on with radio pins incorrect.
6. Run satellite for 5 minutes.

#### Expected Result

1. Antenna disconnected
   1. Error code be ERR\_NONE for radio startup on.
   2. Error code is ERR\_TX\_TIMEOUT for radio transmissions.
      1. Satellite restarts after a minute delay.
   3. Error code is ERR\_RX\_TIMEOUT for radio receive transmissions.
   4. Error code is N/A for configuration of radio.
      1. If an error code is returned, restart the satellite after a minute.
2. Antenna connected
   1. Error code be ERR\_NONE for radio startup on.
   2. Error code is ERR\_PACKET\_TOO\_LONG or N/A for radio transmissions.
   3. Error code is ERR\_RX\_TIMOUT, ERR\_CRC\_MISMATCH N/A for radio receive transmissions.
      1. On timeout the satellite continues execution.
      2. On CRC mismatch the satellite restarts after a minute.
   4. Error code is N/A for configuration of radio.
      1. If an error code is returned, restart the satellite after a minute.
3. Pins incorrect.
   1. Error code be ERR\_CHIP\_NOT\_FOUND for radio startup on.
      1. Satellite restarts after a minute delay.
   2. Error code is ERR\_TX\_TIMEOUT for radio transmissions.
   3. Error code is ERR\_RX\_TIMEOUT for radio receive transmissions.
   4. Error code is N/A for configuration of radio.
      1. If an error code is returned restart the satellite after a minute.

#### Real Result

### SAFESECT2 – Test the string check function returns false for invalid strings.

#### Steps

1. Power the satellite on.
2. Send callsign change command to >64 byte string.

#### Expected Result

1. No callsign change.

#### Real Result

### SAFESECT3 – Test the deployment state eeprom values for errors and error handling.

#### Steps

1. Reset satellite EEPROM.
2. Power the satellite on.
3. Note deployment messages serial println.

#### Expected Result

1. Deployment 0,0 results in an EEPROM wipe.
2. Deployment 0,1 results in nothing.
3. Deployment 1,1 results in nothing.
4. Deployment 1,0, results in EEPROM wipe.

#### Real Result

* SAFESECT1

# Communication

### COMMST1 - Testing the enable and disable transmission state

#### Steps

1. Disable transmissions. “FOSSASAT-17;”
2. Restart Command. “FOSSASAT-114;”
3. Enable transmissions. “FOSSASAT-18;”
4. Disable transmissions. “FOSSASAT-17;”
5. MANUALLY RESTART SATELLITE
6. Restart Command. “FOSSASAT-114;”
7. Enable transmissions. “FOSSASAT-18;”
8. MANUALLY RESTART SATELLITE.
9. Send PING command. “FOSSASAT-15;”

#### Expected result

1. –
2. Satellite restarts via a command.
3. –
4. –
5. Satellite restarts.
6. Satellite ignores restart command.
7. –
8. –
9. Satellite replies with PONG command. “FOSSASAT-16;”

#### Received Result

* TODO

### COMMST2 – Testing the repeater

#### Steps

1. Enable transmissions. “FOSSASAT-18;”
2. Send repeater message. “FOSSASAT-15;Repeater Message”

#### Expected result

1. Receive repeater message. “FOSSASAT-16;Repeater Message”

The satellite should re-transmit the given message on the next Communications System cycle.

Re-transmitted message will be received by the ground station.

#### Received Result

* TODO

### COMMST3 – Testing the Ping-Pong feature

#### Steps

#### Expected result

1. Receive “FOSSASAT-16;”

Pong to be transmitted on the next Communication System cycle.

#### Received Result

### COMMST4 – Testing the programmable callsign

#### Steps

#### Expected result

The transmitted callsign (sent start of every message) will change to the given string.

#### Received Result

### COMMST5 – Testing the transmission password protection

#### Steps

#### Expected result

#### Received Result

### COMMST6 – Testing the notification of satellite power on.

#### Steps

#### Expected result

1. Receive “FOSSASAT-11;”

The satellite to have all LEDs on and executing the program checked via serial communication.

#### Received Result

### COMMST7 – Testing the notification of satellite deployment.

#### Steps

#### Expected result

1. Receive “FOSSASAT-14;”

The antenna and solar panels to be deployed.

#### Received Result

### COMMST8 – Testing the eeprom reset message.

#### Steps

#### Expected result

Reset counter and transmission state wiped.

#### Received Result

### COMMST9 – Testing the restart command

#### Steps

#### Expected result

The satellite should switch off and then back on again.

#### Received Result

### COMMST10 – Testing the frequency band switching

#### Steps

#### Expected result

#### Received Result

### COMMST11 – Testing the manual deployment command

#### Steps

#### Expected result

If the antenna and solar panels have not been deployed automatically, they are deployed.

#### Received Result

### COMMST12 – Testing the System Information command

#### Steps

1. Transmit

#### Expected result

1. Receive “FOSSASAT-19;XXXX”

Receive charging voltage, battery voltage, solar cell a/b/c voltages, reset count, deployment state and board temperature packet.

#### Received Result

### COMMST13 – Testing the notification of satellite shutting down.

#### Steps

#### Expected result

1. Receive “FOSSASAT-12;”

The satellite to cease executing and to power off and power on again automatically.

#### Received Result

### COMMST14 – Checking transmission times.

#### Steps

1. For each protocol command, check the transmission is received for;
   1. 10 second,
   2. 30 second,
   3. 1 minute
   4. 5 minutes

#### Expected Result

1. The satellite to print to the serial monitor a successful response to each protocol command for each given time.

#### Real Result

# Persistent Storage

Since the persistent storage routines now write to multiple EEPROM locations and read from multiple locations and perform a most-common-value test, we need to test this function.

### PT1 - Write 0 to eeprom at address 0

#### Test Description

Write the integer value 0 to the Persistent\_Storage\_Set() at starting address 0.

#### Expected Result

6 values of 0 (2 byte integers) written to eeprom addresses 0-1, 100-101, 200-201, 300-301, 400-401, 500-501

#### Measured Result

### PT2 - Write 1 to eeprom at address 0

#### Test Description

Write the integer value 1 to the Persistent\_Storage\_set() at starting address 0.

#### Expected Result

Same as PT1 with 1 stored.

#### Measured Result

### PT3 - Write 32767 to eeprom at address 0

#### Test Description

Write the integer value 32767 (int max) to the Persistent\_Storage\_set() at starting address 0.

#### Expected Result

Same as PT1 with 32767 stored.

#### Measured Result

### PT4 - Write -32768 to eeprom at address 0

#### Test Description

Write the integer value -32768 (int min) to the Persistent\_Storage\_set() at starting address 0.

#### Expected Result

Same as PT1 with -32768 stored.

#### Measured Result

### PT5 - Write 60000 to eeprom at address 0

#### Test Description

Write the integer value 60000 to the Persistent\_storage\_set() at the starting address 0.

#### Expected Result

Same as PT1 with 60000 stored.

#### Measured Result

### PT6 - Write -60000 to eeprom at address 0

#### Test Description

Write the integer value -60000 to the Persistent\_Storage\_Set() at the starting address 0.

#### Expected Result

Same as PT1 with -60000 stored.

#### Measured Result

### PT7 - Write 0 to eeprom at address 100

#### Test Description

Write the integer value 0 to Persistent\_storage\_set() address 100

#### Expected Result

6 values of 0 (2 byte integers) written to eeprom addresses 100-101, 200-201, 300-301, 400-401, 500-501, 600-601

#### Measured Result

### PT8 - Write 1 to eeprom at address 100

#### Test Description

Write the integer value 1 to the Persistent\_Storage\_set() at starting address 100.

#### Expected Result

Same as PT7 with 1 stored.

#### Measured Result

### PT9 - Write 32767 to eeprom at address 100

#### Test Description

Write the integer value 32767 (int max) to the Persistent\_Storage\_set() at starting address 100.

#### Expected Result

Same as PT7 with 32767 stored.

#### Measured Result

### PT10 - Write -32768 to eeprom at address 100

#### Test Description

Write the integer value -32768 (int min) to the Persistent\_Storage\_set() at starting address 100.

#### Expected Result

Same as PT7 with -32768 stored.

#### Measured Result

### PT11 - Write 60000 to eeprom at address 100

#### Test Description

Write the integer value 60000 to the Persistent\_storage\_set() at the starting address 100.

#### Expected Result

Same as PT7 with 60000 stored.

#### Measured Result

### PT12 - Write -60000 to eeprom at address 100

#### Test Description

Write the integer value -60000 to the Persistent\_Storage\_Set() at the starting address 100.

#### Expected Result

Same as PT7 with -60000 stored.

#### Measured Result

### PT13 – Write 10 to EEPROM at address 0 and then read from EEPROM at address 0

#### Test Description

Write then Read the integer with Persistent\_Storage\_Get() at starting address 0.

#### Expected Result

Integer 0

#### Measured Result