Team AI - 26

AVIONICS INTEGRATION TEST BENCH

VERIFICATION & VALIDATION

ELEC 491 Capstone Project

University of British Columbia

11 April 2023

Table of Contents

Glossary

Term	Definition
ARINC	Aeronautical Radio Inc. (Communication Protocol)
CDC	Communication Device Class (for USB)
FIFO	First In First Out
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
I/O	Input/Output
IC	Integrated Circuit
LSB	Least Significant Bit
MCU	Microcontroller
MSB	Most Significant Bit
OS	Operating System
PCB	Printed Circuit Board
RX	Receive
SPI	Serial Peripheral Interface
TX	Transmit
USB	Universal Serial Bus

1. Project Context

1.1 Background Information

The Avionics Integration Test Bench, codenamed AeroSim, is a product for KF Aerospace that allows a computer to emulate various aircraft parameters transferred to avionics equipment and read data from said avionics equipment simultaneously. This allows KF Aerospace to emulate bi-directional data communication between avionics equipment and experiment with novel integration techniques prior to undertaking the expensive process of modifying an in-service aircraft.

For further information regarding the design and operation of the product, please refer to the Design Document.

The Verification and Validation Document seeks to test this system to verify functionality and validate conformity to client requests by checking against requirements set in the Requirements Document.

Thorough verification and validation needs to be performed on both the hardware and software to ensure a robust design that will allow the test bench to support engineers at KF Aerospace effectively.

2. Overview

This document outlines verification and validation tests in line with the requirements set out in the Requirements Document. The Requirements Document is part of the Milestone 2 submission for all references henceforth.

In this document, verification and validation are described as follows.

- *Verification:* Checking that the product is designed and developed as per the specified requirements.
- *Validation:* Checking that the product has met the client's needs and expectations.

Both verification and validation alike are critical components of quality assurance and management for the AeroSim Avionics Test Bench. Verification ensures that the developed or developing product meets the specified requirements outlined in the requirements document. These requirements are put in place in order to ensure that the desired functionality and usage is as expected by the client. Validation helps to ensure that all client needs and expectations are met, allowing for a dependable product to be easily integrated with the clients production.

3. Product Verification

3.1 Verification Plan Description

Outlined below in section 3.2 are various scheduled test plans that have been outlined with individual pre-requirements, test procedures, and test outcomes. The specific parent requirement(s) associated with each test case are specified and directly associated with said test case. In order to carry out an individual test case, one must first meet the pre-requirements. Afterwards the specific step by step test procedure is to be followed. Thereupon, unique results can be compared to the test cases expected outcome. If the observed/measured results differ from the specified outcome, the verification test case has failed.

Basic GUI functionality is to be verified through 8 test cases. GUI functionality, backend logic and user input checking are all verified through these tests. Backend software message transmission verification occurs through 7 test cases. Likewise, the working of serial and SPI communication will each have 2 individual test cases, and the FPGA logic and outputs will be verified in 4. Lastly, the verification of the discrete outputs, end-to-end communication and overall load and stress testing have their individual test cases associated with them.

3.2 Scheduled Test Plans

1. Basic GUI Functionality

- Number of Test Cases: 8

Test Plan Description: This test plan aims to verify all overarching features and general settings provided by the AeroSim GUI, such as theme selection, TX toggling, data selection and logging behaviour. These settings must function correctly in order to ensure the user has a seamless experience using the interactable elements on the GUI.

It is also important that users are prevented from executing actions that may potentially result in undesirable or undefined behaviour.

<table>
<tr><td colspan="6"><b>1.1. Test Case: Software booting state</b></td></tr>
<tr><td><b>Component(s):</b></td><td colspan="5">GUI</td></tr>
<tr><td><b>Parent Requirement:</b></td><td colspan="5">FR3.7</td></tr>
<tr><td><b>Test Overview:</b></td><td colspan="5">This test verifies the proper booting and shut down of the software’s GUI and backend such that the software is reset to its initial state.</td></tr>
<tr><td><b>Setup and Pre-Requirements</b></td><td colspan="5">● AeroSim software is installed.</td></tr>
<tr><td><b>Test Procedure:</b></td><td colspan="5">1. Start the AeroSim software.
2. Change bus configurations, toggle switches, and enter data at will.
3. Exit the AeroSim software.
4. Start the AeroSim software.
● Confirm that all configurations, switches, and data fields are off, blank, or reset to their initial states.</td></tr>
<tr><td><b>Expected Outcome:</b></td><td colspan="5">● After AeroSim software booting, all configurations, switches, and data fields are off, blank, or reset to their initial states.
● TX and RX switches are OFF.
● All Bus Bitrate drop down menus are initialised to OFF.</td></tr>
<tr><td><b>Date Executed:</b></td><td colspan="5">7 April 2023</td></tr>
<tr><td><b>Executed By:</b></td><td colspan="5">Team</td></tr>
<tr><td><b>Test Case Result</b></td><td>[✓ Pass]</td><td>Fail</td><td>Blocked</td><td>Incomplete</td><td>Not Executed</td></tr>
<tr><td><b>Additional Notes:</b></td><td colspan="5"></td></tr>
</table>

1.2. Test Case: Port settings and TX RX toggling

<table>
<tr><td><b>Component(s):</b></td><td colspan="5">GUI</td></tr>
<tr><td><b>Parent Requirement:</b></td><td colspan="5">FR3.3.6, FR3.3.7</td></tr>
<tr><td><b>Test Overview:</b></td><td colspan="5">This test verifies that TX and RX switches can only be toggled once a COM port is selected.</td></tr>
<tr><td><b>Setup and Pre-Requirements</b></td><td colspan="5">● AeroSim software is installed.
● AeroSim adapter is plugged in.</td></tr>
<tr><td><b>Test Procedure:</b></td><td colspan="5">1. Start the AeroSim Software.
2. Open the Global Settings menu shown by a ⚙ icon.
● Attempt to Toggle TX ON
● Attempt to Toggle RX ON
3. Select the correct COM port
4. Toggle the TX and RX switches ON.
5. Toggle the RX switch OFF.</td></tr>
<tr><td><b>Expected Outcome:</b></td><td colspan="5">● The TX and RX switches can be toggled once the COM port is selected, otherwise a warning message pops up and the toggles are reset back to the OFF state.
● Individual RX bus selections cannot be updated while RX is ON until RX is toggled OFF.</td></tr>
<tr><td><b>Date Executed:</b></td><td colspan="5">7 April 2023</td></tr>
<tr><td><b>Executed By:</b></td><td colspan="5">Team</td></tr>
<tr><td><b>Test Case Result</b></td><td>[✓ Pass]</td><td>Fail</td><td>Blocked</td><td>Incomplete</td><td>Not Executed</td></tr>
<tr><td><b>Additional Notes:</b></td><td colspan="5"></td></tr>
</table>

1.3. Test Case: Changing the COM port settings

Component(s):	GUI
Parent Requirement:	FR3.3.1, FR3.3.2
Test Overview:	This test verifies the software behaviour when COM port settings are changed.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in.
Test Procedure:	1. In the AeroSim software, open the Global Settings menu shown by a ⚙ icon. 2. Select a COM port. 3. Toggle the TX and RX switches ON. 4. Select a different COM port. ● Confirm that a warning message pops up asking to confirm the action. 5. Click Ok on the pop-up message
Expected Outcome:	● The TX and RX switches are automatically toggled OFF once the COM port settings are updated. ● The LOGGING ON/OFF will NOT be toggled OFF
Date Executed:	7 April 2023
Executed By:	Team
Test Case Result	[✓ Pass] Fail Blocked Incomplete Not Executed
Additional Notes:	

1.4. Test Case: Day/Night Mode Feature	
Component(s):	GUI
Parent	NFR4

<table>
<tr><td><b>Requirement:</b></td><td colspan="5"></td></tr>
<tr><td><b>Test Overview:</b></td><td colspan="5">This test confirms the software behaviour when the Day/Night mode feature is updated during transmission.</td></tr>
<tr><td><b>Setup and Pre-Requirements</b></td><td colspan="5">● AeroSim software is installed.
● AeroSim adapter is plugged in.</td></tr>
<tr><td><b>Test Procedure:</b></td><td colspan="5">1. In the AeroSim software, open the Global Settings menu shown by a ⚙ icon.
2. Update the flight parameter configurations at will, and toggle the TX and RX switches ON.
3. Toggle the Day/Night switch.
● Confirm that a warning message pops up asking to confirm the action.
4. Click Ok on the pop-up message</td></tr>
<tr><td><b>Expected Outcome:</b></td><td colspan="5">● The GUI theme is updated to a light layout if previously toggled in night mode, or a dark layout if previously toggled in day mode.
● The TX and RX switches are automatically toggled OFF.
● All flight parameter configurations and switches are reset to their initial state.
● The log file created during the RX toggling is closed.</td></tr>
<tr><td><b>Date Executed:</b></td><td colspan="5">7 April 2023</td></tr>
<tr><td><b>Executed By:</b></td><td colspan="5">Team</td></tr>
<tr><td><b>Test Case Result</b></td><td>[✓ Pass]</td><td>Fail</td><td>Blocked</td><td>Incomplete</td><td>Not Executed</td></tr>
<tr><td><b>Additional Notes:</b></td><td colspan="5">- Added dracula theme as well.</td></tr>
</table>

1.5. Test Case: RX log file

Component(s):	GUI
Parent Requirement:	FR3.3.7, FR3.3.8, FR3.3.9
Test Overview:	This test verifies the logging file status when RX is toggled.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in.
Test Procedure:	<u>Logging to File Enabled:</u> 1. In the AeroSim software, open the RX menu shown by a 🗁 icon. 2. Toggle the RX ON switch from the Global Settings menu shown by a ⚙ icon. 3. Click the Select Logging Location ● Confirm that a folder selection dialog pops up ● Select desired directory 4. Toggle LOGGING ON/OFF switch ON 5. Toggle the RX switch ON 6. Toggle the RX switch OFF ● Confirm that a log file has been created in the directory chosen <u>Logging to File Disabled:</u> 7. Toggle LOGGING ON/OFF switch OFF 8. Toggle the RX switch ON 9. Toggle the RX switch OFF <u>Logging to File With Invalid Directory:</u> 10. Toggle LOGGING ON/OFF switch ON ● Confirm that a folder selection dialog pops up

	● Cancel or close the folder selection dialog without selecting a directory ● Confirm that LOGGING ON/OFF toggle is reset to OFF 11. Toggle RX switch ON ● Confirm no logging file has been generated anywhere 12. Toggle the RX switch OFF.
Expected Outcome:	● User can select directory to generate the log file in using the LOGGING ON/OFF toggle switch ● LOGGING ON/OFF switch is set to OFF when user does not select a valid directory or cancels the dialog before selecting a directory ● No log file is created upon RX toggled ON if LOGGING ON/OFF switch is OFF ● A log file is created in user specified directory upon RX toggled ON if LOGGING ON/OFF switch is ON. ● The log file created during the RX toggling is safely closed after toggling the RX switch OFF. ● The generated log file is populated with both TX and RX information containing the following details: ○ Flight data parameters displayed in ARINC 429 word representation, separated by bit field (label, SDI, data field, SSM, and parity). ○ Physical units chosen for each flight data parameter (eg. metres, degrees, etc.). ○ Bus interface of the message transmitted or received and corresponding timestamp.
Date Executed:	10 April 2023

Executed By:	Isabelle André				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

1.6. Test Case: Maximum number of flight parameters on a bus	
Component(s):	GUI
Parent Requirement:	FR3.3.11
Test Overview:	This test verifies that the maximum number of 10 labels transmitted on a bus cannot be exceeded.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● COM port and bus 1 bitrate is configured, TX is ON.
Test Procedure:	1. Configure 10 different flight parameters to be sent on Bus 1, and toggle continuous send. 2. Attempt to configure another flight parameter for Bus 1. 3. Toggle the Bus 1 OFF on one of the 10 flight parameters being transmitted. 4. Configure another flight parameter for bus 1 and toggle continuous send.
Expected Outcome:	● The maximum amount of flight parameters allowed to be transmitted on a single bus is 10. ● After 10 toggles are selected for a single bus, selecting an eleventh toggle for that same bus will result in a pop-up

	error. The eleventh toggle will consequently be switched off.				
Date Executed:	7 April 2023				
Executed By:	Anthony Wang				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

1.7. Test Case: Data field range validation	
Component(s):	GUI
Parent Requirement:	FR3.3.10
Test Overview:	This test verifies that the user inputted data respects each flight parameter's accepted range.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● COM port and bus 1 bitrate is configured, TX is ON.
Test Procedure:	1. Enter -100 in the Altitude data field, configure a Bus Selection, and toggle Continuous Send.
Expected Outcome:	● Data field does not accept the value and resets the entered value to a previous valid value. If there is no previous valid value it will reset to the default placeholder value of 0. ● A new data message using the value displayed in the entry box is sent out
Date Executed:	7 April 2023
Executed By:	Anthony Wang

Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

2. Software Backend Message Transmission

- Number of Test Cases: 7

Test Plan Description: This test plan aims to verify the structure, validity, and transmission protocol of messages output by the GUI to the rest of the system. These tests are essential to ensuring the FPGA executes the correct behaviour intended by the user. The software in DEBUG Mode is used to view message construction and user logic directly on the command line, without the need to communicate with outside devices. This mode is used to ensure that message sequencing and construction is correct on the GUI side before being sent via serial.

2.1. Test Case: TX toggling message	
Component(s):	GUI, software backend
Parent Requirement:	FR3.3.6, FR3.3.12
Test Overview:	This test verifies the ARINC429 messages constructed from user inputs when TX is toggled before bus bitrates have been configured.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in.
Test Procedure:	1. Boot the AeroSim software in DEBUG mode. 2. Open the Global Settings menu shown by a ⚙ icon. 3. Select the correct COM port. 4. Open the bus bitrate config side menu within the Settings Tab and ensure that all buses are initialised as OFF.

	5. Update flight parameter configurations at will. ● Monitor the command line debug logs and confirm that no data messages are sent. 6. Toggle the TX switch ON. 7. Toggle the TX switch OFF.
Expected Outcome:	● No data messages are sent or shown on the command line debug screen before TX is toggled ON. ● A single TX ON message formatted as 1_000_0000 is shown on the command line debug message, followed by 6 bus bitrate configuration messages formatted as X_0_XXXXXX_XXXX_0010, where X is dependent on buses toggled. ● A single TX OFF message formatted as 0_000_0000 is shown on the command line debug message

Date Executed:	7 April 2023				
Executed By:	Anthony Wang				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

2.2. Test Case: Bitrate configuration messages	
Component(s):	GUI, software backend
Parent Requirement:	FR3.3.4, FR3.3.12
Test Overview:	This test verifies the ARINC429 messages constructed from user inputs when bus bitrates are configured.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in.

Test Procedure:	1. Boot the AeroSim software in DEBUG mode. 2. Open the Global Settings menu shown by a ⚙ icon. 3. Select the correct COM port. 4. Open the bus bitrate config side menu within the Settings Tab and select 100 kbps (high speed) from the Bus 1 Bitrate drop down menu, and 12.5 kbps (low speed) from the Bus 2 Bitrate drop down menu. 5. Ensure all flight parameters are unselected and at their initial state. 6. Toggle the TX switch ON. 7. Select 12.5 kbps (lowspeed) from the Bus 1 Bitrate drop down
Expected Outcome:	● No data messages are sent or shown on the command line debug screen before TX is toggled ON. ● Once TX is ON, the debug command line should show: ○ A TX ON message formatted as 1_000_0000. ○ A Bus 1 Bitrate configuration message formatted as 1_1_000001_XXXX_0010, where X is dependent on active bus toggles. ○ A Bus 2 Bitrate configuration message formatted as 0_1_000010_XXXX_0010. ○ 4 subsequent bus bitrate configuration messages formatted as X_0_XXXXXX_XXXX_0010 ● No data messages are sent or shown on the command line debug screen after TX is toggled ON. ● Once Bus 1 Bitrate is updated, a single bus bitrate configuration message should show on the debug command line as X_0_000001_0000_0010, where X is arbitrary.

Date Executed:	7 April 2023				
Executed By:	Anthony Wang				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

2.3. Test Case: Flight parameter data transmission	
Component(s):	GUI, software backend
Parent Requirement:	FR3.3.10
Test Overview:	This test verifies the ARINC429 messages constructed from user inputs when flight parameters are being transmitted.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in.
Test Procedure:	1. Boot the AeroSim software in DEBUG mode. 2. Open the Global Settings menu shown by a ⚙ icon. 3. Select the correct COM port. 4. Open the bus bitrate config side menu within the Settings Tab and select 100 kbps (high speed) from the Bus 1 Bitrate drop down menu. 5. Enter 100 in the Altitude (1013.25 mb) Flight Parameter data field. Toggle Bus 1 ON, and Continuous Send ON. 6. Ensure Update Rate is 50 ms for the parameter specified in the previous step 7. Toggle the TX switch ON. 8. Enter 100 in the Computed Airspeed Flight Parameter data field. Toggle Bus 1, and click the Single Send button.

Expected Outcome:	- No data messages are sent or shown on the command line debug screen before TX is toggled ON. - Once TX is ON, the debug command line should show: ○ A TX ON message formatted as 1_000_0000. ○ A Bus 1 Bitrate configuration message formatted as 1_1_000001_XXXX_0010, where X is arbitrary. ○ 5 subsequent bus bitrate configuration messages formatted as X_0_XXXXXX_XXXX_0010, where X is arbitrary. ○ A 56 bit data message with the first byte corresponding to 0000_0001 (command code). ○ Tx period = 0000110010 ○ Bus config = 000001 ○ ARINC Word = 10000000000000110010000011000001 - Once the Computed Airspeed parameter is sent as a single send message, the debug command line should show another 56 bit data message with the first byte corresponding to 0000_0001 (command code). ○ Tx period = 0000000000 ○ Bus config = 000001 ○ ARINC Word: = 10000001100100000000000001100001				
Date Executed:	7 April 2023				
Executed By:	Team				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

2.4. Test Case: Updating bus bitrate mid-transmission	
Component(s):	GUI, software backend
Parent Requirement:	FR3.3.4
Test Overview:	This test verifies the behaviour of ARINC429 message transmission when the bus bit rate is changed while data is being transmitted on a bus.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in.
Test Procedure:	1. Boot the AeroSim software in DEBUG mode. 2. Open the Global Settings menu shown by a ⚙ icon. 3. Select the correct COM port. 4. Open the bus bitrate config side menu within the Settings Tab and select 100 kbps (highspeed) from the Bus 1 Bitrate drop down menu. 5. Enter 100 in the Altitude (1013.25 mb) Flight Parameter data field. Toggle Bus 1 ON, and Continuous Send ON. 6. Toggle the TX switch ON. 7. Update the Bus 1 Bitrate configuration to 12.5 kbps (low speed).
Expected Outcome:	● No data messages are sent or shown on the command line debug screen before TX is toggled ON. ● Once TX is ON, the debug command line should show: ○ A TX ON message formatted as 1_000_0000. ○ A Bus 1 Bitrate configuration message formatted as 1_1_000001_XXXX_0010, where X is arbitrary.

	○ 5 subsequent bus bitrate configuration messages formatted as X_0_XXXXXX_XXXX_0010, where X is dependent on the active bus toggles. ○ A 56 bit data message with the first byte corresponding to 0000_0001. ○ Tx period = 0000110010 ○ Bus config = 000001 ○ ARINC Word = 10000000000000110010000011000001 ● Once the Bus 1 Bitrate is updated, the debug command line should show the following messages: ○ A Bus 1 Bitrate configuration message formatted as 0_1_000001_XXXX_0010. ○ A 56 bit data message with the first byte corresponding to 0000_0001. ○ Tx period = 0000110010 ○ Bus config = 000001 ○ ARINC Word = 10000000000000110010000011000001				
Date Executed:	7 April 2023				
Executed By:	Anthony Wang				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

2.5. Test Case: Ending the transmission of a flight parameter	
Component(s):	GUI, software backend
Parent Requirement:	FR3.4

Test Overview:	This test verifies the behaviour of ARINC429 message transmission from the software when the transmission of a flight parameter is stopped in the GUI.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in.
Test Procedure:	1. Boot the AeroSim software in DEBUG mode. 2. Open the Global Settings menu shown by a ⚙ icon. 3. Select the correct COM port. 4. Open the bus bitrate config side menu within the Settings Tab and select 100 kbps (highspeed) from the Bus 1 Bitrate drop down menu. 5. Enter 100 in the Altitude (1013.25 mb) Flight Parameter data field. Toggle Bus 1 ON, and toggle the TX switch on. 6. Toggle the continuous send ON . 7. Toggle the Altitude Rate Flight Parameter Continuous Send OFF.
Expected Outcome:	● No data messages are sent or shown on the command line debug screen before TX is toggled ON. ● Once TX is ON, the debug command line should show: ○ A TX ON message formatted as 1_000_0000. ○ A Bus 1 Bitrate configuration message formatted as 1_1_000001_XXXX_0010, where X is arbitrary. ○ 5 subsequent bus bitrate configuration messages formatted as X_0_XXXXXX_XXXX_0010, where X is arbitrary. ○ A 56 bit data message with the first byte corresponding to 0000_0001.

	○ Tx period = 0000110010 ○ Bus config = 000001 ○ ARINC Word = 10000000000000110010000011000001 ● Once the Altitude (1013.25 mb) Flight Parameter Continuous Send is toggled OFF, a message formatted with the two least significant Bytes as XX_000001_XXXX_0011, is observed on the debug command line.				
Date Executed:	7 April 2023				
Executed By:	Anthony Wang				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

2.6. Test Case: RX toggling messages	
Component(s):	GUI, software backend
Parent Requirement:	FR3.3.7
Test Overview:	This test verifies the behaviour of ARINC429 message transmission from the software when RX and logging is toggled ON and OFF.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in.
Test Procedure:	1. Boot the AeroSim software in DEBUG mode. 2. Open the Global Settings menu shown by a ⚙ icon. 3. Select the correct COM port. 4. Select which inbound RX buses to log and listen to.

	5. Toggle RX ON. 6. Toggle RX OFF.				
Expected Outcome:	- Once RX is toggled ON, the debug command line should show an RX message formatted as XXXX_XXXX_1_XXX_0101. - Once RX is toggled OFF, the debug command line should show an RX message formatted as XXXX_XXXX_0_XXX_0101.				
Date Executed:	10 April 2023				
Executed By:	Isabelle André				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

2.7. Test Case: Exiting the software during transmission and logging	
Component(s):	GUI, software backend
Parent Requirement:	FR3.7
Test Overview:	This test verifies the behaviour of the GUI and ARINC 429 messages when the software is terminated unexpectedly or exited.
Setup and Pre-Requirements	- AeroSim software is installed. - AeroSim adapter is plugged in.
Test Procedure:	1. Boot the AeroSim software in DEBUG mode. 2. Select the correct COM port, and toggle both TX and RX ON. 3. Exit out of the AeroSim software.

Expected Outcome:	● Once the software is terminated, the debug command line should immediately show an RX message formatted as 1111_1111_0_XXX_0100, and a TX message formatted as 0_000_0000. where X is arbitrary before ending the application				
Date Executed:	7 April 2023				
Executed By:	Anthony Wang				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

3. Serial Communication

- Number of Test Cases: 2

Test Plan Description: This test plan aims to verify the communication behaviour between the AeroSim GUI and the MCU. It is essential for ensuring communication between the GUI and MCU can be established and that messages sent can correctly be read or passed on by the MCU. It is also important that unexpected loss of communication events are handled properly to avoid undefined system behaviour.

3.1. Test Case: Adapter COM port appearing in software drop down	
Component(s):	AeroSim adapter (PCB, STM32, FPGA), AeroSim software
Parent Requirement:	FR3.1
Test Overview:	This test confirms that the AeroSim adapter can be detected as a COM port.
Setup and Pre-Requirements	● AeroSim software is installed.

<table>
<tr><td><b>Test Procedure:</b></td><td colspan="5">1. Plug in AeroSim adapter to laptop.
2. Open the Global Settings menu shown by a ⚙ icon.
3. Click the COM Port Selector drop down menu</td></tr>
<tr><td><b>Expected Outcome:</b></td><td colspan="5">- A COM port is shown in the drop down menu corresponding to the adapter.
- Adapter LED light is flashing</td></tr>
<tr><td><b>Date Executed:</b></td><td colspan="5">7 April 2023</td></tr>
<tr><td><b>Executed By:</b></td><td colspan="5">Team</td></tr>
<tr><td><b>Test Case Result</b></td><td>[✓ Pass]</td><td>Fail</td><td>Blocked</td><td>Incomplete</td><td>Not Executed</td></tr>
<tr><td><b>Additional Notes:</b></td><td colspan="5"></td></tr>
</table>

3.2. Test Case: Adapter Serial communication ending unexpectedly	
Component(s):	AeroSim adapter (PCB, STM32, FPGA), AeroSim software
Parent Requirement:	FR3.7
Test Overview:	This test verifies the behaviour of the AeroSim software and adapter if serial communication is cut unexpectedly during transmission.
Setup and Pre-Requirements	- AeroSim software is installed. - AeroSim adapter is plugged in.
Test Procedure:	1. Plug in AeroSim adapter to laptop. 2. Open the Global Settings menu shown by a ⚙ icon. 3. Click the COM Port Selector drop down menu. 4. Select bus 1 bitrate and select the TX toggle on. 5. Unplug the usb cord from the laptop

Expected Outcome:	● A pop-up error message appears indicating a loss of communication. ● The GUI software returns to its initial reset state. ● The MCU recognizes a loss of communication, and seizes all commands.				
Date Executed:	7 April 2023				
Executed By:	Anthony Wang				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

4. SPI Communication

- Number of Test Cases: 2

Test Plan Description: This test plan aims to verify the communication behaviour between the Aerosim MCU and FPGA. It is essential that messages passed on by the MCU from the GUI are of correct format and occur as expected. The MCU must also have the ability to receive data from the FPGA at any time, and correctly pass it back to the GUI.

4.1. Test Case: STM32 SPI message outbound	
Component(s):	PCB, STM32, GUI, software back-end
Parent Requirement:	FR1.1.2
Test Overview:	This test verifies that messages being outputted from the PCB contain the correct config fields and data to be transmitted to the FPGA.
Setup and Pre-Requirements	● AeroSim software is installed. ● PCB component of the adapter is plugged in without FPGA

	● Logic analyzer probing pins 25,26,27,28 of the PCB
Test Procedure:	1. On the AeroSim software, select the correct COM port, and toggle TX ON. 2. Toggle RX Bus 1 ON to enable logging on the bus, and toggle RX ON. 3. Configure the Altitude Rate flight parameter and toggle Continuous Send ON 4. Toggle RX OFF.
Expected Outcome:	● With SPI selected on the logic analyzer, the waveforms should show the MOSI, CS, and CLK channel pulsing according to each message sent. ● The logic analyzer should show the following messages being sent: ○ A 1 byte TX ON message ○ Six 2 byte bus bitrate config message ○ A 2 byte RX ON message ○ A 7 byte data message ○ A 2 byte RX OFF message ● Each of the above messages match with the user inputted data if any, as well as the 4 bit config.
Date Executed:	7 April 2023
Executed By:	Anthony Wang
Test Case Result	[✓ Pass] Fail Blocked Incomplete Not Executed
Additional Notes:	

4.2. Test Case: FPGA interrupt and SPI message inbound	
Component(s):	FPGA, avionics equipment (eg. GPS device)

Parent Requirement:	FR1.1.2				
Test Overview:	This test verifies that inbound messages received by avionics equipment and outputted from the FPGA via SPI are correctly formatted and reconstructed.				
Setup and Pre-Requirements	● Avionics equipment (eg. GPS device) ● Logic analyzer probing pins 25,26,27,28 of the FPGA.				
Test Procedure:	1. Connect avionics equipment (eg. GPS device) to PCB 2. Ensure FPGA and PCB are powered on as indicated by LED light. 3. Configure the avionics equipment to output some value/data. (eg. input heading into GPS device)				
Expected Outcome:	● The logic analyzer should show the incoming waveforms generated from the ARINC line receiver. ● A correct FPGA interrupt should be activated, to begin parsing the received data.				
Date Executed:	10 April 2023				
Executed By:	Isabelle André				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

5. FPGA Logic and Outputs

- Number of Test Cases: 4

Test Plan Description: This test plan aims to verify the structure, validity, and transmission protocol of messages output by the FPGA to the rest of the system. These tests are essential to ensuring that the PCB executes the correct behaviour intended, as well as for ensuring that data is correctly received from test equipment and transmitted back to the software. A logic analyzer is used to view specific signals of interest, and functionality of FPGA logic.

<table>
<tr><th colspan="6">5.1. Test Case: Minimum 4 bit time width between transmitted words on a bus</th></tr>
<tr><td><b>Component(s):</b></td><td colspan="5">AeroSim adapter (PCB, STM32, FPGA), AeroSim software</td></tr>
<tr><td><b>Parent Requirement:</b></td><td colspan="5">FR1.1.1</td></tr>
<tr><td><b>Test Overview:</b></td><td colspan="5">This test verifies that different data words transmitted on a single label are always transmitted at least 4 bits apart.</td></tr>
<tr><td><b>Setup and Pre-Requirements</b></td><td colspan="5">● AeroSim software is installed.
● AeroSim adapter is plugged in.
● Logic analyzer probing pins 22, 23 of the PCB.
● AeroSim software GUI has TX toggled ON and Bus 1 Bitrate is configured at 100 kbps (highspeed)</td></tr>
<tr><td><b>Test Procedure:</b></td><td colspan="5">1. Configure 10 flight parameters to be sent on Bus 1 and toggle Continuous Send.
2. Continuously monitor the logic analyzer waveforms.</td></tr>
<tr><td><b>Expected Outcome:</b></td><td colspan="5">● Zooming into the waveforms where data messages appear to overlap, there are always at least 4 bit wide cycles between data words sent.</td></tr>
<tr><td><b>Date Executed:</b></td><td colspan="5">9 April 2023</td></tr>
<tr><td><b>Executed By:</b></td><td colspan="5">Isabelle André</td></tr>
<tr><td><b>Test Case Result</b></td><td>[✓ Pass]</td><td>Fail</td><td>Blocked</td><td>Incomplete</td><td>Not Executed</td></tr>
<tr><td><b>Additional Notes:</b></td><td colspan="5"></td></tr>
</table>

<table>
<tr><th colspan="2">5.2. Test Case: Last word completes transmission before bus is reset</th></tr>
<tr><td><b>Component(s):</b></td><td>AeroSim adapter (PCB, STM32, FPGA), AeroSim software</td></tr>
</table>

Parent Requirement:	FR1.1.1
Test Overview:	This test verifies that data words are not truncated when transmission is reset.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● Logic analyzer probing pins 22, 23 of the PCB. ● AeroSim software GUI has TX toggled ON and Bus 1 Bitrate is configured at 100 kbps (highspeed)
Test Procedure:	1. Configure 10 flight parameters to be sent on Bus 1 and toggle Continuous Send. 2. Toggle TX OFF. 3. Repeatedly Toggle TX ON and OFF 10 times and monitor the logic analyzer waveforms.
Expected Outcome:	● The last data word transmitted always transmits in full with 32 bits.
Date Executed:	9 April 2023
Executed By:	Isabelle André
Test Case Result	[✓ Pass] Fail Blocked Incomplete Not Executed
Additional Notes:	No ARINC 429 word was seemingly truncated, however it is also possible that due to the latency between words, the TX ON/OFF command never overlapped with a word being transmitted.

5.3. Test Case: FPGA hardware reset button	
Component(s):	AeroSim adapter (PCB, STM32, FPGA), AeroSim software
Parent Requirement:	FR1.1.1

Test Overview:	This test verifies that data words are not truncated when transmission is turned off.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● Logic analyzer probing pins 22, 23 of the PCB. ● AeroSim software GUI has TX toggled ON and all Bus Bitrates are configured at 100 kbps (highspeed)
Test Procedure:	1. Configure a random flight parameter to be sent on each of the 6 buses and toggle Continuous Send. 2. Press the reset button located in the small pin hole of the adapter box.
Expected Outcome:	● Transmission from the adapter hardware stops. ● The last data word transmitted always transmits in full with 32 bits.
Date Executed:	9 April 2023
Executed By:	Isabelle André
Test Case Result	[✓ Pass] Fail Blocked Incomplete Not Executed
Additional Notes:	

5.4. Test Case: No SPI message inbound when RX is toggled OFF	
Component(s):	AeroSim adapter (PCB, STM32, FPGA), AeroSim software, avionics equipment (eg. GPS module)
Parent Requirement:	FR1.1.2
Test Overview:	This test verifies that RX messages from avionics equipment are not transmitted by the FPGA to the rest of the adapter when RX is toggled OFF.

Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● Logic analyzer probing pins 26,27,28,29 of the FPGA. ● RX is toggled OFF.				
Test Procedure:	1. Toggle RX Bus 1 ON. 2. Toggle RX switch ON				
Expected Outcome:	● Observing the logic analyzer waveforms, no SPI messages are sent from the FPGA until RX is toggled ON.				
Date Executed:	9 April 2023				
Executed By:	Isabelle André				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

6. Discrete Outputs

- Number of Test Cases: 3

Test Plan Description: This test plan aims to verify the functionality of the discrete outputs. It is essential that all open/ground and open/28V discrete signals function correctly in order to drive the two-state voltage or current sources.

6.1. Test Case: 28V discrete output	
Component(s):	AeroSim adapter (PCB, STM32), AeroSim software
Parent Requirement:	FR2.3
Test Overview:	This test confirms that the 28V discrete outputs can be controlled by the GUI and output the correct voltage.
Setup and	● AeroSim software is installed.

<table>
<tr><th colspan="6">6.1. Test Case: 28V discrete output</th></tr>
<tr><td><b>Pre-Requirements</b></td><td colspan="5">● AeroSim adapter is plugged in.
● Multimeter</td></tr>
<tr><td><b>Test Procedure:</b></td><td colspan="5">1. In the AeroSim software, toggle both 28V discrete outputs ON.
2. Using a multimeter, probe both 28V discrete outputs.
3. Toggle both 28V discrete outputs OFF and repeat the measurement.</td></tr>
<tr><td><b>Expected Outcome:</b></td><td colspan="5">● All of the discrete outputs are Open when toggled Open, and 28V when toggled 28V.</td></tr>
<tr><td><b>Date Executed:</b></td><td colspan="5">12 March 2023</td></tr>
<tr><td><b>Executed By:</b></td><td colspan="5">Team</td></tr>
<tr><td><b>Test Case Result</b></td><td>[✓ Pass]</td><td>Fail</td><td>Blocked</td><td>Incomplete</td><td>Not Executed</td></tr>
<tr><td><b>Additional Notes:</b></td><td colspan="5"></td></tr>
</table>

6.2. Test Case: Open discrete output	
Component(s):	AeroSim adapter (PCB, STM32), AeroSim software
Parent Requirement:	FR2.2
Test Overview:	This test confirms that the Open discrete outputs can be controlled by the GUI and output the correct voltage.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● Multimeter

Test Procedure:	1. In the AeroSim software, toggle all Open discrete outputs to GND. 2. Using a multimeter, probe all Open discrete outputs. 3. Toggle all Open discrete outputs OFF and repeat the measurement.				
Expected Outcome:	● All of the discrete outputs are Open when toggled Open, and grounded when toggled GND				
Date Executed:	12 March 2023				
Executed By:	Team				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

6.3. Test Case: Exiting the software during transmission	
Component(s):	AeroSim adapter (PCB, STM32), AeroSim software
Parent Requirement:	FR3.7
Test Overview:	This test confirms that all discrete outputs are automatically reset to Open when the AeroSim software is terminated.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● Multimeter
Test Procedure:	1. In the AeroSim software, toggle all Open discrete outputs to GND and 28V accordingly. 2. Exit out of the AeroSim software. 3. Using a multimeter, probe all discrete outputs.

Expected Outcome:	● All of the discrete outputs are reset to Open once the software is terminated.				
Date Executed:	12 March 2023				
Executed By:	Team				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

7. Load and Stress Testing

- Number of Test Cases: 2

Test Plan Description: This test plan aims to verify the bus loading capability of the adapter hardware and the physical robustness of the enclosed hardware. These tests are essential to ensuring that the device can withstand the stresses of everyday use by the user.

7.1. Test Case: Max loading of outputs	
Component(s):	AeroSim adapter (PCB, STM32), AeroSim software, avionics equipment
Parent Requirement:	FR1.1, FR1.1.1.1
Test Overview:	This test verifies the functioning of the adapter hardware at maximum load for a duration of time.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● Every TX and RX bus are plugged in avionics equipment
Test Procedure:	1. In the AeroSim software, toggle all Open discrete outputs to GND and 28V accordingly.

	2. Toggle all RX Buses ON. 3. Configure all 6 TX Bus rates to 100 kbps high speed. 4. Configure 10 Flight parameters corresponding to the piece of equipment connected and toggle Continuous Send. 5. Toggle RX and TX ON. 6. Wait 10 minutes.
Expected Outcome:	● Adapter continues to transmit data for 15 minutes. ● Adapter does not overheat and conserves a manageable temperature to the touch. ● Adapter continues to receive data from avionics equipment and logging to a log file.
Date Executed:	10 April 2023
Executed By:	Isabelle André
Test Case Result	Pass [✓ Fail] Blocked Incomplete Not Executed
Additional Notes:	Only tested for a single RX bus as only a single GPS module was available for testing. Full set up using avionics equipment remains untested. RX and TX buses are looped back into one another. Some crashing occurs at high load.

7.2. Test Case: Physical robustness	
Component(s):	AeroSim adapter (PCB, STM32, FPGA), adapter case
Parent Requirement:	NFR5
Test Overview:	This test verifies the minimum robustness of the adapter hardware and box to sustain minimal shock.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in.

<table>
<tr><td><b>Test Procedure:</b></td><td colspan="5">1. Vigorously shake the adapter.
2. Drop the adapter from a height of 1 metre.
3. Plug the adapter in.</td></tr>
<tr><td><b>Expected Outcome:</b></td><td colspan="5">● No adapter component has come loose or moved inside the case.
● The adapter status LED light lights up when the adapter is plugged in.
● The COM port can be detected.</td></tr>
<tr><td><b>Date Executed:</b></td><td colspan="5"></td></tr>
<tr><td><b>Executed By:</b></td><td colspan="5"></td></tr>
<tr><td><b>Test Case Result</b></td><td>Pass</td><td>Fail</td><td>[✓ Blocked]</td><td>Incomplete</td><td>Not Executed</td></tr>
<tr><td><b>Additional Notes:</b></td><td colspan="5">Will not be attempted at this time due to upcoming project deadlines and extra part order down time.</td></tr>
</table>

8. End-to-End System

- Number of Test Cases: 3

Test Plan Description: This test plan aims to verify the functionality of the system as a whole by testing end-to-end data transmission and receiving behaviour. This test plan is essential to ensuring the system outputs messages conforming to the ARINC-429 message protocol and can communicate both ways with avionics equipment that operates using the ARINC-429 message protocol.

8.1. Test Case: Data words are transmitted at the bitrate and transmit interval set	
Component(s):	AeroSim adapter (PCB, STM32, FPGA), GUI
Parent Requirement:	FR1.1.1.2

<table>
<tr><td><b>Test Overview:</b></td><td colspan="5">This test verifies that the adapter transmits data at the set bitrate and transmit interval set in the GUI.</td></tr>
<tr><td><b>Setup and Pre-Requirements</b></td><td colspan="5">● AeroSim software is installed.
● AeroSim adapter is plugged in.
● Logic analyzer probing each TX bus
● COM port is configured.</td></tr>
<tr><td><b>Test Procedure:</b></td><td colspan="5">1. Open the bus bitrate config side menu within the Settings Tab and select 100 kbps (highspeed) from the Bus 1 Bitrate drop down menu.
2. Enter a value in the Altitude Rate Flight Parameter data field. Toggle Bus 1 ON, and Toggle TX ON.
3. Toggle Continuous Send ON.
4. Repeat at a bus bitrate of 12.5 kbps (low speed).
5. Repeat for all TX buses</td></tr>
<tr><td><b>Expected Outcome:</b></td><td colspan="5">● The logic analyzer cursor shows that the transmitted data bit pulse width corresponds to the configured bitrate on the GUI.
● The logic analyzer cursor shows that the time between a transmitted word of a flight parameter and the next corresponds to the configured transmit interval on the GUI.
● All cursor measurements in the time domain are within a 1 ms error margin in the case of a single parameter transmitted on a bus.</td></tr>
<tr><td><b>Date Executed:</b></td><td colspan="5">7 April 2023</td></tr>
<tr><td><b>Executed By:</b></td><td colspan="5">Anthony Wang</td></tr>
<tr><td><b>Test Case Result</b></td><td>[✓ Pass]</td><td>Fail</td><td>Blocked</td><td>Incomplete</td><td>Not Executed</td></tr>
<tr><td><b>Additional Notes:</b></td><td colspan="5"></td></tr>
</table>

8.2. Test Case: All 32 flight data parameters can be transmitted on any bus	
Component(s):	AeroSim adapter (PCB, STM32, FPGA), GUI, avionics equipment
Parent Requirement:	FR1.1.1
Test Overview:	This test verifies the adapter can transmit any flight parameter data accurately on any bus
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● Logic analyzer probing each TX bus ● COM port is configured. ● Bus bitrate is configure for each bus as 100 kbps (highspeed), and TX is toggled ON ● Avionics equipment matching each flight parameters.
Test Procedure:	1. Enter an arbitrary value within range in the Altitude Rate Flight Parameter data field. Toggle all buses ON, and Continuous Send ON. 2. Repeat for all flight parameters. 3. Remove logic analyzer and replace with avionics equipment on TX buses. 4. Repeat the procedure.
Expected Outcome:	● The logic analyzer cursor shows that the 32 bit transmitted ARINC429 data corresponds to the converted user inputs. ● All cursor measurements in the time domain are within a 1 ms error margin in the case of a single parameter transmitted on a bus. ● User inputted data emulates the correct values on avionics equipment.

Date Executed:	7 April 2023				
Executed By:	Anthony Wang				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:					

8.3. Test Case: Avionics equipment RX communication	
Component(s):	AeroSim adapter (PCB, STM32, FPGA), GUI, avionics equipment
Parent Requirement:	FR1.1.2
Test Overview:	This test verifies that the adapter can receive data from avionics equipment and send data back to the software to be logged.
Setup and Pre-Requirements	● AeroSim software is installed. ● AeroSim adapter is plugged in. ● COM port is configured. ● Bus bitrate is configured for each bus as 100 kbps (highspeed), and TX is toggled ON. ● Each of the 8 RX buses are connected to an arbitrary avionics equipment (eg. GPS module). ● 6 of these devices must also be connected to one of the TX buses.
Test Procedure:	1. Toggle all RX buses ON. 2. Toggle RX ON. 3. Enter an arbitrary value within range for a flight parameter corresponding to the avionics equipment plugged into a bus.

	4. Toggle the bus for the flight parameter ON, and toggle Continuous Send ON. 5. Repeat with 3 different flight parameters for each TX bus.				
Expected Outcome:	● The log file has recorded all of the user’s actions on the GUI as well as any inbound RX messages received from avionics equipment ● Each event in the log file is time stamped.				
Date Executed:	10 April 2023				
Executed By:	Isabelle André				
Test Case Result	Pass	Fail	Blocked	[✓ Incomplete]	Not Executed
Additional Notes:	Only tested for each RX bus individually, as only a single GPS module was available for testing. The functioning of all RX buses in parallel was tested in a loop back test by connecting TX outputs to RX inputs, however full set up using avionics equipment remains untested. Test passed for each bus individually with the logging function.				

9. PCB Functionality

- Number of Test Cases: 5

Test Plan Description: This test plan aims to verify the functionality of all ICs on the PCB and the PCB itself. This test plan should be conducted before any integration with the FPGA and GUI logic.

9.1. Test Case: Correct Voltage Levels	
Component(s):	AeroSim PCB
Parent Requirement:	FR2.2

Test Overview:	This test verifies that all ICs have the correct power rails (5V and 3.3V) connected, and that the power rails are at the expected levels.
Setup and Pre-Requirements	● PCB can be powered via USB.
Test Procedure:	1. Power PCB via USB. 2. Use multimeter to verify voltage levels at every 5V, 3.3V, and GND input of ICs as seen on schematic and PCB footprints.
Expected Outcome:	● Correct 3.3V, 5V, and GND levels to be measured.
Date Executed:	10 February 2023
Executed By:	Nursultan Tugolbaev
Test Case Result	[✓ Pass] Fail Blocked Incomplete Not Executed
Additional Notes:	● Faulty LDO footprint that needs to be revised for Rev B. ○ The footprint downloaded from Ultra Librarian had the input and output pins swapped. ○ As a temporary solution, reworked LDO pins by hand. ○ Created RevB that fixed this issue. Design handed off to client for future manufacturing.

9.2. Test Case: Correct Operation of Line Drivers	
Component(s):	AeroSim PCB
Parent Requirement:	FR2.1.3 and FR2.1.4
Test Overview:	This test verifies that the line drivers and line receivers work as intended per ARINC 429 manual.
Setup and Pre-Requirements	● PCB can be powered via USB.

<table>
<tr><td></td><td colspan="5">● Function generator, oscilloscope, USB for oscilloscope captures</td></tr>
<tr><td><b>Test Procedure:</b></td><td colspan="5">1. Connect function generator (square wave, 3.3V, 12.5kHz or 100kHz) to TXx_BITy.
a. x stands for any line driver 1 to 6. y stands for BIT0 or BIT1.
2. For 12.5kHz speed, leave UDx_HL unconnected. For 100kHz speed, connect UDx_HL to a 3.3V debug pin.
3. Connect oscilloscope to TXx_OUT_A and TXx_OUT_B.
4. Power PCB via USB.
5. Turn function generator on.
6. Capture difference between TXx_OUT_A and TXx_OUT_B using MATH function on an oscilloscope.
7. Save oscilloscope capture.
8. Repeat steps 1 to 6 for every line driver (1 to 6), for both bits (0 and 1), and for both speeds (12.5kHz and 100kHz).</td></tr>
<tr><td><b>Expected Outcome:</b></td><td colspan="5">● Refer to Figure 2.9 in the Line Driver section of Design Document for expected operation of a line driver.
● When TXx_BIT0 is high, expect TXx_OUT_B to be high and TXx_OUT_A to be low.
● When TXx_BIT1 is high, expect TXx_OUT_A to be high and TXx_OUT_B to be low.
● When toggling and un-toggling UDx_HL, expect slope to change in accordance to ARINC spec.</td></tr>
<tr><td><b>Date Executed:</b></td><td colspan="5">10 February 2023</td></tr>
<tr><td><b>Executed By:</b></td><td colspan="5">Nursultan Tugolbaev</td></tr>
<tr><td><b>Test Case Result</b></td><td>[✓ Pass]</td><td>Fail</td><td>Blocked</td><td>Incomplete</td><td>Not Executed</td></tr>
</table>

Additional Notes:	None.

9.3. Test Case: Correct Operation of Line Receivers	
Component(s):	AeroSim PCB
Parent Requirement:	FR2.1.3 and FR2.1.4
Test Overview:	This test verifies that the line drivers and line receivers work as intended per ARINC 429 manual.
Setup and Pre-Requirements	● PCB can be powered via USB. ● Function generator, oscilloscope, USB for oscilloscope captures
Test Procedure:	1. Create a 6.6V differential signal pair using a function generator. a. Set CH1 to square wave, 3.3V, 12.5kHz or 100kHz. b. Set CH2 to square wave, -3.3V (use offset), same frequency and CH1. c. Create phase difference of 180 degrees. d. Connect differential pair across RXx_IN_A and RXx_IN_B. i. x stands for any line receiver pair 1 to 8. 2. Connect oscilloscope to RXx_BIT0 and RXx_BIT1. 3. Capture RXx_BIT0 and RXx_BIT1 and save it. 4. Repeat steps 1 to 6 for every line receiver (1 to 8), for both bits (0 and 1), and for both speeds (12.5kHz and 100kHz).

Expected Outcome:	● Refer to Figure 2.9 in the Line Driver section of Design Document for expected operation of a line receiver. ○ It is the opposite behaviour of a line driver. ● When RXx_IN_A is high and RXx_IN_B is low, expect RXx_BIT1 to be high and RXx_BIT0 to be low. ● When RXx_IN_A is low and RXx_IN_B is high, expect RXx_BIT1 to be low and RXx_BIT0 to be high.				
Date Executed:	11 February 2023				
Executed By:	Nursultan Tugolbaev				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:	None.				

9.4. Test Case: Correct Operation of GND/Open Discrete Output Control from MCU	
Component(s):	AeroSim PCB
Parent Requirement:	FR2.2.2
Test Overview:	This test verifies that GND/Open MOSFETs behave correctly when toggled by MCU.
Setup and Pre-Requirements	● PCB can be powered via USB. ● Firmware can be edited and flashed to the MCU. ● A 5V power supply is connected across GND of the PCB and a 1kOhm resistor in series with a single discrete output.
Test Procedure:	1. Power PCB via USB. 2. Connect programmer to MCU programming header of the PCB without a power supply from the programmer.

	3. Temporarily edit the device firmware to toggle one of the GND discrete output control pins at 200ms period, 75% duty cycle (on for 150ms, off for 50ms). 4. Use oscilloscope to verify voltage at the discrete output terminal connected to the 5V supply is between 0 and 0.2V (to account for Schottky diode drop) during the programmed on time, and 5V during the programmed off time. 5. Repeat steps 3 to 5 for the remaining 3 GND/open discrete outputs, moving the 5V supply and resistor to the pin under test each time (Make sure to de-power PCB while moving probes). 6. Flash the unmodified firmware image to the MCU.
Expected Outcome:	● Voltage at each discrete output terminal changes between 5V for 50ms and 0V-0.2V for 150ms as programmed.

Date Executed:	11 February 2023				
Executed By:	Nursultan Tugolbaev, Andrew Hanlon				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:	None.				

9.5. Test Case: Correct Operation of 28V/Open Discrete Output Control from MCU	
Component(s):	AeroSim PCB
Parent Requirement:	FR2.2.3
Test Overview:	This test verifies that 28V/Open MOSFETs behave correctly when toggled by MCU.

Setup and Pre-Requirements	● PCB can be powered via USB. ● Firmware can be edited and flashed to the MCU. ● A 28V power supply is connected across GND of the PCB and a 1kOhm resistor in series with a single discrete output.				
Test Procedure:	1. Power PCB via USB. 2. Connect programmer to MCU programming header of the PCB without a power supply from the programmer 3. Use multimeter to verify voltage on the 28V terminal connected to the 28V supply is 28V. 4. Temporarily edit the device firmware to toggle one of the 28V discrete output control pins at 200ms period, 75% duty cycle (on for 150ms, off for 50ms). 5. Use oscilloscope to verify voltage of the discrete output terminal toggles between about 28V (accounting for Schottky diode drop) during the programmed on time, and 0V during the programmed off time. 6. Repeat steps 3 to 5 for the other 28V/open discrete output, moving the 28V supply and resistor to the pin under test (Make sure to de-power PCB while moving probes). 7. Flash the unmodified firmware image to the MCU.				
Expected Outcome:	● Voltage at each discrete output terminal changes from 0V to ~28V when that discrete output is toggled by MCU.				
Date Executed:	11 February 2023				
Executed By:	Nursultan Tugolbaev, Andrew Hanlon				
Test Case Result	[✓ Pass]	Fail	Blocked	Incomplete	Not Executed
Additional Notes:	None.				

4. Product Validation

4.1 Product Validation Description

Outlined in section 4.2 are various validation criteria set in order to ensure that all the clients needs are met and satisfied. Each validation criteria is loosely associated with a requirement set out in the requirements document. In order to validate the specific case, the criteria outlined must be met. Meeting said criteria can be done in a qualitative or multifaceted requirement manner. This means that there is no specific single requirement that must be passed. Rather, all needs must be met in whatever fashion necessary.

4.2 Validation Criteria

1. Per requirement 2.1 outlined in Section 4.2 of Requirement Document:

Validation Criteria:	1.1. The minimum, desired, or stretch number of transmit and receive buses set by the client for the product have been met.		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

Validation Criteria:	1.2. All buses support both “high speed” (100kbit/s) and “low speed” (12.5kbit/s) communication. 1.2.1. The bit rate of each transmit bus is independent and configurable.

	1.2.2. Each receive bus is capable of receiving data at either bit rate without any configuration by the user.		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

Validation Criteria:	1.3. there are a minimum of 4 discrete GND/Open signal outputs. 1.3.1. If stretch goal has been implemented, there are a minimum of 4 discrete 28VDC/Open signal outputs.		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

Validation Criteria:	1.4. Power ON/OFF indicator LED is implemented		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

Validation Criteria:	1.5. System error indicator LED is implemented		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

Validation Criteria:	1.6. Hardware housing fully encloses the electronic components. 1.6.1. Chip programming connectors are enclosed. 1.6.2. ARINC429 and discrete signal bus connectors are accessible. 1.6.3. USB bus connector is accessible.		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

2. Per requirement 3.1 outlined in Section 4.3 of Requirement Document:

Validation Criteria:	2.1. A GUI allows the user to control the entire system. 2.1.1. Details of all control inputs present are outlined in requirement 3.4 of Section 4.3 of Requirement Document.
Date Validated:	7 April 2023

Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

3. Per requirements outlined in Section 5 of Requirement Document:

Validation Criteria:	3.1. System is designed to allow for future feature additions to improve or expand system functionality. 3.1.1. The software code is documented and allows for future edits. 3.1.1.1. Software tools are well-established, well-supported with documentation and active user bases. 3.1.2. The GUI is modular and allows for future data/control additions on the screen. 3.1.3. Hardware components are chosen with additional future usage of processing power, pins, etc. in mind.		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

Validation Criteria:	3.2. The system is as cost-effective as possible.

	3.2.1. Justification for each component choice is given as part of the proposed budget document.		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

Validation Criteria:	3.3. The GUI is intuitive and user-friendly. ***NOTE:*** *User-friendliness is defined by the feedback and preferences of the client*		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

Validation Criteria:	3.4. The system is robust, and can withstand everyday usage. ***NOTE:*** *Withstanding everyday usage is defined by the feedback and usages of the client*		
Date Validated:	7 April 2023		
Validated By:	Anthony Wang		
Status:	[✓ Meets Requirements]	Does Not Meet Requirements	Not Validated
Additional Notes:			

Appendix A: Team Information

Name	Initials	Tech Lead	Management Lead
Anthony Wang	AW	GUI software (front end)	Deliverable Organiser
Nursultan Tugolbaev	NT	Hardware	Internal Discussions / Meetings
Patric McDonald	PM	GUI software (back end)	Treasurer Minute taker
Andrew Hanlon	AH	Firmware	Client Communication
Isabelle André	IA	Digital logic design	Internal/External Communications

Appendix B: Project Report Contributions

Section	Major Content	Minor Content	Author	Reviewer
Section 1 and 2	NT PM		NT PM	PM
Section 3.1	PM		PM	NT
Section 3.2	IA AW NT	NT PM AW	IA	AW PM IA
Section 4.1	PM		PM	NT
Section 4.2	IA NT	PM	IA NT	PM
All sections formatting	AW IA NT	AW NT IA	AW IA NT	IA NT PM