This document explains the function of the IFJR, its schematic level design, its board level design, and its functional testing

IFJR

In-Flight JTAG Reprogrammer

Revision: 1.0.1

Gabe Adamson

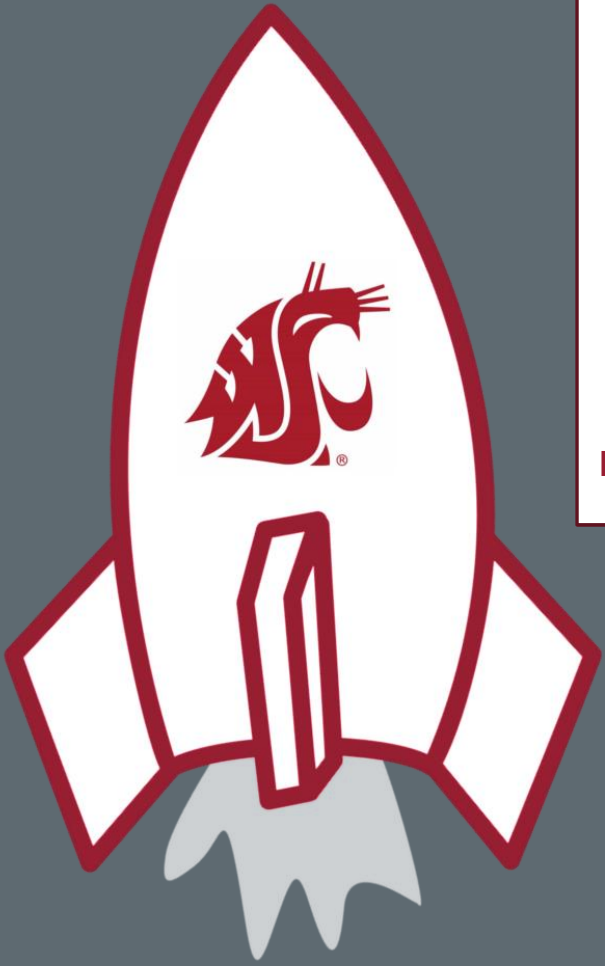

Table of Contents

1 Introduction

This document explains how the IFJR will fulfil the following Functions and conform to the following Requirements. This document refers to the Avionics Board version 1.1.

1.1 Function

The IFJR is responsible for programming subsystem microprocessors using JTAG.

1.2 Requirements

The requirements and design requirements for the IFJR can be found on GitHub.

2 Detailed Description

This section references the Avionics Board schematic. Page numbers will be listed and may have coordinates listed (number and letter combination found around the frame).

2.1 Functional Block Diagram

The block diagram can be found on the first page of the schematic.

2.1.1 IFJR Microcontroller

The IFJR receives commands from the IHU[1] to reprogram other microcontrollers while in orbit using JTAG.[2] The IFJR will alert the microcontroller before reprogramming.[3]

2.1.2 IFJR SD Cards

Two SD cards serve as redundant, non-volatile memory[4] to store data necessary to reprogram the other microcontrollers on the satellite. All data is stored in non-volatile memory in order to buffer the data and avoid corruption.

2.2 Schematic

2.2.1 IFJR Microcontroller

The IFJR microcontroller[5] (page 9) facilitates the reprogramming of microcontrollers. It was chosen for its ease of programming, and lower power consumption. The IFJR microcontroller communicates with other microcontrollers through a JTAG bus, and the IHU its two SD cards through two SPI bus. It communicates with IHU through an SPI bus.

2.2.2 SD Cards

The two SD Cards[6] (page 10) serve as redundant non-volatile memory for the IFJR Microcontroller. The IFJR microcontroller performs reads and writes through an SPI bus.

2.2.3 SPI Bus

The IFJR has two SPI buses (page 9, B2, B4). IFJR_BUS is for communicating with the IHU microcontroller, and IFJR_SPI is for accessing the SD cards.

[1] Requirement IFJR-005
[2] Requirement IFJR-002
[3] Requirement IFJR-003
[4] Requirement IFJR-004
[5] STM32L476RG
[6] MOLEX_5025700893

2.3 Board

The board shall be double layered with $1\ oz$ copper and ENIG finish. The board shall also conform to the dimensions specified by the CougSat Module Standard.

2.3.1 Layout Constraints

Unless specified in the following subsections, all signals shall use the default parameters below. Signals in the following subsections do not include their sense signals unless otherwise specified. Trace width can be broken if a trace needs to bottleneck down to a pin, the bottleneck shall be minimized.

Trace width:	$0.16mm$
Vias:	$\emptyset 0.3mm$, unlimited count
Separation:	$0.16mm$
Length:	unlimited

Devices with specific placement and routing considerations are called out on the schematic, see “CAD Note:”

2.3.1.1 JTAG - IFJR_[JTCK, JTDI, JTDO, JTMS], BUS_JTAG_[TCK, TDI, TDO, TMS]

Length:	Each node shall be length matched $\pm 10.0mm$
Stubs:	$< 10.0mm$

2.3.1.2 SPI - IFJR_SPI_[SCK, MOSI, MISO], IFJR_BUS_[SCK, MOSI, MISO]

Length:	Each node shall be length matched $\pm 10.0mm$
Stubs:	$< 10.0mm$

3 Testing

All tests shall be performed at room temperature and not under vacuum unless otherwise specified. If any modifications are performed, take note. Include enough information to understand circuit behavior and for others to replicate the results. Include any software written to execute the test and link it in the test notes section. Save all software, waveforms, etc. in a subfolder of the board’s test folder for each test.

- Waveforms shall be captured whenever appropriate
- Have the event take fill the screen (for fast events, zoom in; for slow events, zoom out)
- Label each channel accurately
- Only have bandwidth limiting if necessary for the test (this applies to the oscilloscope and probe settings)
- If ringing or overshoot occurs, use a ground spring or differential probe

Common test instructions can be found on the wiki.

3.1 Before First Power-On Check

Configuration:

3.1.1 Test Instructions

Measure the resistance of various points in reference to *PGND* located at the backplane. When measuring in circuit resistances, flip the probes and take the lower value.

3.1.2 Test Data

Node	Resistance		Node	Resistance
3.3V-5			IFJR_BUS_MISO	
BUS_JTAG_TCK			IFJR_SPI_SCK	
BUS_JTAG_TDI			IFJR_SPI_MOSI	
BUS_JTAG_TDO			IFJR_SPI_MISO	
BUS_JTAG_TMS			IFJR_JTMS	
IFJR_BUS_SCK			IFJR_JTCK	
IFJR_BUS_MOSI			IFJR_JTDO	

3.1.3 Test Notes

3.2 Reprogramming

Configuration:

This test evaluates the ability to reprogram another microcontroller

3.2.1 Test Instructions

Use the BUS_JTAG bus to upload an image to another microcontroller, validate the other microcontroller is properly programmed.

3.2.2 Test Data

Program another microcontroller via JTAG, validate the other microcontroller is properly programmed			
Programmer	Microcontroller Programmed	Passing Criteria	Pass / Fail
IFJR		*Microcontroller properly programmed*	

3.2.3 Test Notes

3.3 Command Response

Configuration:

This test evaluates the ability to respond to a command from the IHU.

3.3.1 Test Instructions

Send a signal on IFJR_BUS, use IFJR microcontroller to respond.

3.3.2 Test Data

Check if data is sent by IFJR microcontroller when requested on IFJR_BUS			
Device	Signal Received	Passing Criteria	Pass / Fail
IFJR microcontroller		*Data Received*	

3.4 Storage

Configuration:

This test evaluates the functionality of accessing storage.

3.4.1 Test Instructions

Write test package to both SD Cards and read back both test packages.

3.4.2 Test Data

Check if package sent is received when read			
Device	Package Received	Passing Criteria	Pass / Fail
IFJR SD0		*Package Received*	
IFJR SD1		*Package Received*	

3.5 IFJR Programming

Configuration:

This test evaluates the ability to program the IFJR microcontroller

3.5.1 Test Instructions

Connect a SWD programmer to the SWD header and upload an image, validate the IFJR is properly programmed. Connect a JTAG programmer to the backplane and upload an image, validate the IFJR is properly programmed.

Note: Follow the programming instructions on the wiki.

3.5.2 Test Data

Program the IFJR via SWD and JTAG, validate the IFJR is properly programmed		
Programmer	Passing Criteria	Pass / Fail
SWD	*IHU properly programmed*	
JTAG	*IHU properly programmed*	

3.5.3 Test Notes