Gabe Adamson

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



Table of Contents

[1 Introduction 2](#_Toc16596771)

[1.1 Function 2](#_Toc16596772)

[1.2 Requirements 2](#_Toc16596773)

[2 Detailed Description 3](#_Toc16596774)

[2.1 Functional Block Diagram 3](#_Toc16596775)

[2.1.1 Reprogramming 3](#_Toc16596776)

[2.2 Schematic 3](#_Toc16596777)

[2.2.1 IFJR Microcontroller 3](#_Toc16596778)

[2.2.2 SD Cards 3](#_Toc16596779)

[2.3 Board 3](#_Toc16596780)

[2.3.1 Layout Constraints 4](#_Toc16596781)

# 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.

## Function

The IFJR is responsible for programming subsystem microprocessors using JTAG.

## Requirements

The requirements and design requirements for the IFJR can be found on [GitHub](https://github.com/CougsInSpace/CougSat1-Readme/blob/master/CougSat1-Requirements.pdf).

# 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).

## Functional Block Diagram

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

### IFJR Microcontroller

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

### IFJR SD Cards

Two SD cards serve as redundant, non-volatile memory[[4]](#footnote-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.

## Schematic

### IFJR Microcontroller

The IFJR microcontroller[[5]](#footnote-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.

### SD Cards

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

### 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.

## Board

The board shall be double layered with copper and ENIG finish. The board shall also conform to the dimensions specified by the [CougSat Module Standard](https://github.com/CougsInSpace/CougSat1-Hardware/blob/master/CougSat1-Backplane/Documentation/CougSatModuleStandard.pdf).

### 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:

Vias: , unlimited count

Separation:

Length: unlimited

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

#### JTAG – IFJR\_[JTCK, JTDI, JTDO, JTMS], BUS\_JTAG\_[TCK, TDI, TDO, TMS]

Length: Each node shall be length matched

Stubs:

#### SPI – IFJR\_SPI\_[SCK, MOSI, MISO] , IFJR\_BUS\_[SCK, MOSI, MISO]

Length: Each node shall be length matched

Stubs:

# 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](http://cougs.space/wiki).

## Before First Power-On Check

**Configuration:**

### 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.

### Test Data

### Test Notes

## Reprogramming

**Configuration:**

This test evaluates the ability to reprogram another microcontroller

### Test Instructions

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

### 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* |  |

### Test Notes

## Storage

**Configuration:**

This test evaluates the functionality of accessing storage.

### Test Instructions

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

### 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* |  |

## IFJR Programming

**Configuration:**

This test evaluates the ability to program the IFJR microcontroller

### 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](http://cougs.space/wiki).

### 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* |  |

### Test Notes

1. Requirement IFJR-005 [↑](#footnote-ref-1)
2. Requirement IFJR-002 [↑](#footnote-ref-2)
3. Requirement IFJR-003 [↑](#footnote-ref-3)
4. Requirement IFJR-004 [↑](#footnote-ref-4)
5. [STM32L476RG](https://github.com/CougsInSpace/Resources/blob/master/SupplierDocuments/ST/STM32L476.pdf) [↑](#footnote-ref-5)
6. [MOLEX\_5025700893](https://github.com/CougsInSpace/Resources/blob/master/SupplierDocuments/04%20-%20Connectors%20SMT/04-0005/MOLEX_5025700893_sd.pdf) [↑](#footnote-ref-6)