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.

# 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.

### Reprogramming

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

## Schematic

### IFJR Microcontroller

The IFJR Microcontroller[[4]](#footnote-4) (page 8) 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 its two SD cards through an SPI bus.

### SD Cards

The two SD Cards (page 8, C1, C2, D1, & D2) serve as redundant non-volatile memory for the IFJR Microcontroller.[[5]](#footnote-5) The communicate with the IFJR Microcontroller through SPI.

## 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\_SPI0\_[MOSI, MISO]

Length: Each node shall be length matched

Stubs:

1. Requirement IFJR-005 [↑](#footnote-ref-1)
2. Requirement IFJR-002 [↑](#footnote-ref-2)
3. Requirement IFJR-003 [↑](#footnote-ref-3)
4. [STM32L476RG](https://github.com/CougsInSpace/Resources/blob/master/SupplierDocuments/ST/STM32L476.pdf) [↑](#footnote-ref-4)
5. Requirement IFJR-004 [↑](#footnote-ref-5)