Gabe Adamson

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

IHU

Internal Housekeeping Unit

Revision: 1.0.0



Table of Contents

[1 Introduction 2](#_Toc14299516)

[1.1 Function 2](#_Toc14299517)

[1.2 Requirements 2](#_Toc14299518)

[2 Detailed Description 3](#_Toc14299519)

[2.1 Functional Block Diagram 3](#_Toc14299520)

[2.1.1 Command 3](#_Toc14299521)

[2.1.2 Data Handling 3](#_Toc14299522)

[2.2 Schematic 3](#_Toc14299523)

[2.2.1 IHU Microcontroller 3](#_Toc14299524)

[2.2.2 SD Cards 3](#_Toc14299525)

[2.2.3 ADC 4](#_Toc14299526)

[2.2.4 Power 4](#_Toc14299527)

[2.2.5 I2C Bus 4](#_Toc14299528)

[2.3 Board 4](#_Toc14299529)

[2.3.1 Layout Constraints 4](#_Toc14299530)

# Introduction

This document explains how the IHU will fulfil the following Functions and conform to the following Requirements. This document refers to the Avionics Board version 1.0.1.

## Function

The IHU is responsible for the following:

* Managing and processing all forms of data on the satellite
* Prepare and interpret communications to and from the ground
* Keep and distribute the satellite’s time
* Enforce the current operation and mode of the satellite
* Perform periodic status inquiries on all subsystems

## Requirements

The requirements and design requirements for the IHU 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.

### Command

The IHU is responsible for executing commands sent to the satellite from the ground.[[1]](#footnote-1) This includes making sure all subsystems perform the current operation.[[2]](#footnote-2) The IHU also performs periodic inquiries on the status of all subsystems to ensure each subsystem is working as expected.[[3]](#footnote-3)

### Data Handling

The IHU collects and processes information about all subsystems and payloads.[[4]](#footnote-4) All collected data is stored in non-volatile memory in order to buffer the data and avoid corruption.[[5]](#footnote-5) The IHU also interprets all packages received from the ground[[6]](#footnote-6) as well as prepares all packages to be sent to the ground.[[7]](#footnote-7)

## Schematic

### IHU Microcontroller

The IHU Microcontroller[[8]](#footnote-8) (page 9, A4, B1, B2, B3, & B4) executes commands from the ground, collects and processes data from other subsystems, ensures subsystems are working properly, processes packages from the ground, and prepares packages to be sent to the ground. It was chosen for its ease of programming, and lower power consumption. The IHU microcontroller communicates with other microcontrollers and its ADC through two I2C buses, and its two SD cards through an SPI bus.

### SD Cards

The two SD Cards (page 10, C1, C2, D1, & D2) serve as redundant non-volatile memory for the IHU Microcontroller.[[9]](#footnote-9) The communicate with the IHU Microcontroller through SPI.

### ADC

The Analog to Digital Converter[[10]](#footnote-10), or ADC, (page 10, C1, C2, D1, & D2) monitors the temperatures of the components in the C&DH subsystem (IHU and IFJR). The IHU communicates with the ADC through the I2C0 bus.

### Power

The power inputs (page 11, C1, C2, D1, & D2) come from the backplane (from the Electrical Power Subsystem). The IHU uses the 3.3V-0 power source.

### I2C Bus

The IHU has two I2C buses (page 3 B2, B3, & B5). Both are for communicating with other microcontrollers, however the I2C0 bus also has the IHUADC. On the IHU buses, the IHU Microcontroller is the master served by the attached devices.

#### ADC

There is one ADC connected to the IHU Microcontroller. The ADC is located on the Avionics Board. Its address is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **ADC** | **I2C Address** | **AS1** | **AS0** |
| IHU ADC | 0x23 | H | L |

## 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:”

Trace width:

#### JTAG – JTAG-[TCK, TDI, TDO, TMS], BUS\_JTAG-[TCK, TDI, TDO, TMS]

Length: Each node shall be length matched

Stubs:

#### I2C – I2C\_[SDA, SCL], BUS\_I2C\_[SDA, SCL, IRQ]

Length: Each node shall be length matched

Stubs:

1. Requirement CDH-003 [↑](#footnote-ref-1)
2. Requirement CDH-009 [↑](#footnote-ref-2)
3. Requirement CDH-010 [↑](#footnote-ref-3)
4. Requirement CDH-006 [↑](#footnote-ref-4)
5. Requirement CDH-008 [↑](#footnote-ref-5)
6. Requirement CDH-005 [↑](#footnote-ref-6)
7. Requirement CDH-004 [↑](#footnote-ref-7)
8. [STM32L476RG](https://github.com/CougsInSpace/Resources/blob/master/SupplierDocuments/ST/STM32L476.pdf) [↑](#footnote-ref-8)
9. Requirement IFJR-004 [↑](#footnote-ref-9)
10. [AD7291](https://github.com/CougsInSpace/Resources/blob/cad-libraries/SupplierDocuments/27%20-%20Conversion%20IC/27-0003/ANALOG_AD7291.pdf) [↑](#footnote-ref-10)