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: 1.0.1



Table of Contents

[1 Introduction 3](#_Toc19379848)

[1.1 Function 3](#_Toc19379849)

[1.2 Requirements 3](#_Toc19379850)

[2 Detailed Description 4](#_Toc19379851)

[2.1 Functional Block Diagram 4](#_Toc19379852)

[2.1.1 IHU Microcontroller 4](#_Toc19379853)

[2.1.2 C&DH SD Cards 4](#_Toc19379854)

[2.1.3 C&DH ADCS 4](#_Toc19379855)

[2.2 Schematic 4](#_Toc19379856)

[2.2.1 IHU Microcontroller 4](#_Toc19379857)

[2.2.2 C&DH SD Cards 5](#_Toc19379858)

[2.2.3 ADC 5](#_Toc19379859)

[2.2.4 I2C Bus 5](#_Toc19379860)

[2.2.5 SPI Bus 5](#_Toc19379861)

[2.3 Board 6](#_Toc19379862)

[2.3.1 Layout Constraints 6](#_Toc19379863)

[3 Testing 6](#_Toc19379864)

[3.1 Before First Power-On Check 7](#_Toc19379865)

[3.1.1 Test Instructions 7](#_Toc19379866)

[3.1.2 Test Data 7](#_Toc19379867)

[3.1.3 Test Notes 7](#_Toc19379868)

[3.2 I2C Bus 7](#_Toc19379869)

[3.2.1 Test Instructions 7](#_Toc19379870)

[3.2.2 Test Data 7](#_Toc19379871)

[3.2.3 Test Notes 7](#_Toc19379872)

[3.3 SPI Bus 7](#_Toc19379873)

[3.3.1 Test Instructions 7](#_Toc19379874)

[3.3.2 Test Data 8](#_Toc19379875)

[3.3.3 Test Notes 8](#_Toc19379876)

[3.4 Storage 8](#_Toc19379877)

[3.4.1 Test Instructions 8](#_Toc19379878)

[3.4.2 Test Data 8](#_Toc19379879)

[3.5 Temperature Monitoring 8](#_Toc19379880)

[3.5.1 Test Instructions 8](#_Toc19379881)

[3.5.2 Test Data 8](#_Toc19379882)

[3.5.3 Test Notes 9](#_Toc19379883)

[3.6 IHU Programming 9](#_Toc19379884)

[3.6.1 Test Instructions 9](#_Toc19379885)

[3.6.2 Test Data 9](#_Toc19379886)

[3.6.3 Test Notes 9](#_Toc19379887)

# 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 1.1.

## 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](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.

### IHU Microcontroller

The IHU is responsible for executing commands sent to the satellite from the ground[[1]](#footnote-1), ensures subsystems are working properly,[[2]](#footnote-2) processes packages from the ground, and prepares packages to be sent to the ground. The IHU also performs periodic inquiries on the status of all subsystems to ensure each subsystem is working as expected.[[3]](#footnote-3)

### C&DH SD Cards

The C&DH SD cards serve as redundant, non-volatile memory to store collected and processed information from 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)

### C&DH ADCS

The Analog to Digital Converter, or ADC, converts the analog signals from the thermistors, which monitor temperatures of the components in the ADCS subsystem.

## Schematic

### IHU Microcontroller

The IHU microcontroller[[6]](#footnote-6) (page 11) communicates with other microcontrollers and its ADCs through two I2C buses, the In-Flight JTAG Reprogrammer (IFJR) through an SPI bus, and its two SD cards through a separate SPI bus. Reprogramming in orbit is achieved by the IFJR through a JTAG bus.

### C&DH SD Cards

The two SD Cards[[7]](#footnote-7) (page 12) serve as redundant non-volatile memory for the IHU microcontroller. The IHU microcontroller performs reads and writes through an SPI bus.

### ADC

The Analog to Digital Converter[[8]](#footnote-8), or ADC, (page 12) communicates with the IHU microcontroller using the required I2C communication protocol. Thermistors are supplied with a reference voltage supplied by the ADC.

### I2C Bus

The IHU has two I2C buses (page 11, B3). Both are for communicating with other microcontrollers, however the I2C1 bus also has the C&DHADC and a GPIO expander. The microcontrollers on the I2C0 bus are Comms, ADCS, and EPS. Payload is on the I2C1 bus. On the IHU buses, the IHU microcontroller is the master served by the connected devices.

#### ADC

There are two ADCs connected to the IHU microcontroller. One ADC is located on the Avionics Board, the other is off board. Its address is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **ADC** | **I2C Address** | **AS1** | **AS0** |
| IHU ADC | 0x23 | H | L |
| Off Board | 0x22 | H | NC |

#### GPIO Expander

There is a GPIO expander[[9]](#footnote-9) connected to the IHU microcontroller. It is located off board.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Device** | **I2C Address** | **A2** | **A1** | **A0** |
| GPIO Expander | 0x20 | L | L | L |

### SPI Bus

The IHU has two SPI buses (page 11, B2, B4). BUS\_SPI is for communicating with the IFJR microcontroller, and IHU\_SPI is for operating the C&DH 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:”

Trace width:

#### JTAG – IHU\_[JTCK, JTDI, JTDO, JTMS]

Length: Each node shall be length matched

Stubs:

#### I2C – BUS\_I2C0\_[SDA, SCL], BUS\_I2C1\_[SDA, SCL, IRQ]

Length: Each node shall be length matched

Stubs:

#### SPI – BUS\_SPI\_[SCK, MISO, MOSI], COM\_SPI\_[SCK, MISO, MOSI], IHU\_SPI\_[SCK, MISO, MOSI],

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

## I2C Bus

**Configuration:**

This test evaluates the circuit described in I2C Bus

### Test Instructions

Request a reply from a device on both I2C buses. Check if reply is received.

### Test Data

| Check if reply is received on BUS\_I2C0 | | | |
| --- | --- | --- | --- |
| Device | Reply Received | Passing Criteria | Pass / Fail |
|  |  | *Reply Received* |  |

| Check if reply is received on BUS\_I2C1 | | | |
| --- | --- | --- | --- |
| Device | Reply Received | Passing Criteria | Pass / Fail |
|  |  | *Reply Received* |  |

### Test Notes

## SPI Bus

**Configuration:**

This test evaluates the circuit described in SPI Bus

### Test Instructions

Request a reply from a device on both SPI buses. Check if reply is received.

### Test Data

| Check if reply is received on BUS\_SPI | | | |
| --- | --- | --- | --- |
| Device | Reply Received | Passing Criteria | Pass / Fail |
|  |  | *Reply Received* |  |

| Check if reply is received on IHU\_SPI | | | |
| --- | --- | --- | --- |
| Device | Reply Received | Passing Criteria | Pass / Fail |
|  |  | *Reply Received* |  |

### 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 |
| IHU SD0 |  | *Package Received* |  |
| IHU SD1 |  | *Package Received* |  |

## Temperature Monitoring

**Configuration:**

This test evaluates the functionality of the temperature monitoring system.

### Test Instructions

Heat thermistor location. Read resultant voltage through ADC.

### Test Data

| Check if temperature read by the ADC is the same as being supplied | | | | | |
| --- | --- | --- | --- | --- | --- |
| Device | Temperature Applied | Temperature Read | Error | Passing Criteria | Pass / Fail |
| IHU microcontroller |  |  |  | *Error < 2 ° C* |  |
| IFJR microcontroller |  |  |  | *Error < 2 ° C* |  |
| IHU SD0 |  |  |  | *Error < 2 ° C* |  |
| IHU SD1 |  |  |  | *Error < 2 ° C* |  |
| IFJR SD0 |  |  |  | *Error < 2 ° C* |  |
| IFJR SD1 |  |  |  | *Error < 2 ° C* |  |

### Test Notes

## IHU Programming

**Configuration:**

This test evaluates the ability to program the IHU microcontroller

### Test Instructions

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

Note: Follow the programming instructions on the [wiki](http://cougs.space/wiki).

### Test Data

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

### Test Notes

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. [STM32L476RG](https://github.com/CougsInSpace/Resources/blob/master/SupplierDocuments/ST/STM32L476.pdf) [↑](#footnote-ref-6)
7. [MOLEX\_5025700893](https://github.com/CougsInSpace/Resources/blob/master/SupplierDocuments/04%20-%20Connectors%20SMT/04-0005/MOLEX_5025700893_sd.pdf) [↑](#footnote-ref-7)
8. [AD7291](https://github.com/CougsInSpace/Resources/blob/cad-libraries/SupplierDocuments/27%20-%20Conversion%20IC/27-0003/ANALOG_AD7291.pdf) [↑](#footnote-ref-8)
9. [TCA9535](https://github.com/CougsInSpace/Resources/blob/master/SupplierDocuments/27%20-%20Conversion%20IC/27-0001/TI_tca9535.pdf) [↑](#footnote-ref-9)