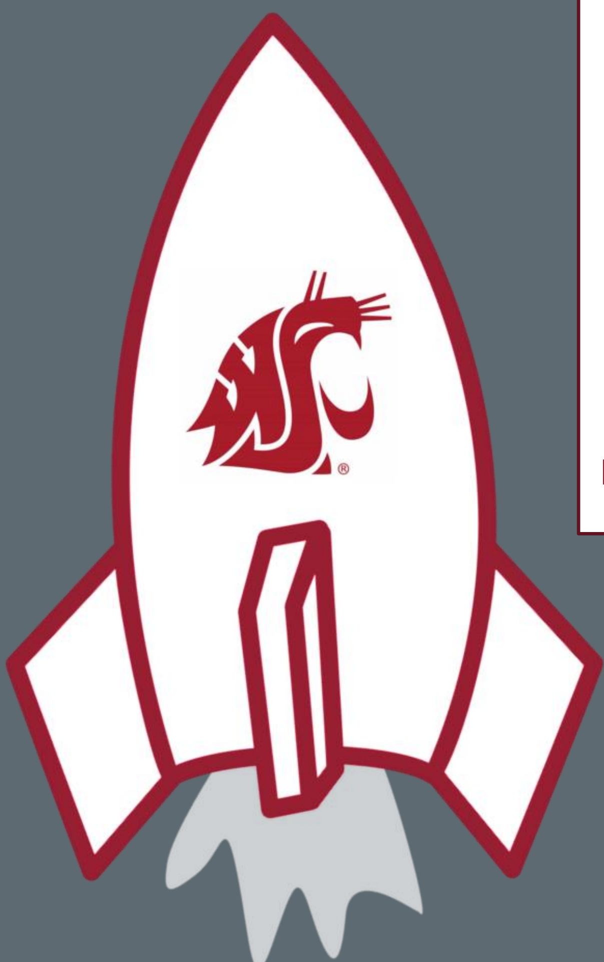
This document explains the function of the ADCS, its schematic level design, its board level design, and its functional testing

# ADCS

## Attitude Determination and Control System Design

Revision: 1.0.1

Gabe Adamson



# Table of Contents

# 1 Introduction

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

## 1.1 Function

The ADCS is responsible for the following:

- Operating the magnetorquer coils
- Collecting data on the position and orientation of the satellite

## 1.2 Requirements

The requirements and design requirements for the ADCS 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 ADCS Microcontroller

The ADCS microcontroller performs the main control loop. This involves handling the data received by the various sensors in the ADCS subsystem, as well as operating the three magnetorquer coils. The ADCS provides data collected to the Internal Housekeeping Unit (IHU) when requested[1] and rotates the satellite as directed by the IHU[2].

### 2.1.2 EEPROM

The two EEPROM cards serve as redundant, nonvolatile memory[3] that stores constants the main control loop needs to perform accurate rotations.

### 2.1.3 H-Bridges

The H-Bridges provide a controlled, variable current to the magnetorquer coils. Varying the current increases and decreases the strength of the resultant magnetic field produced by each coil. The bidirectional functionality of the H-bridges allows for the magnetic field to be produced in either direction on the axis.

### 2.1.4 Magnetorquer Coils

The Magnetorquer coils are located on the +Z, -X, and -Y panels. The ADCS microcontroller uses the H-bridges to control the amount of current flowing through each coil at a given time. The combinations of the three produced magnetic fields will be used to rotate the satellite in orbit.[4]

### 2.1.5 IMUs

The Inertial Measurement Units, or IMUs, sense changes in the satellite's acceleration, orientation, and gravity experienced.[5] This data is necessary to monitor the rotations being performed by the ADCS, as well as knowing what external forces to account for in calculations.

### 2.1.6 ADCs

The Analog to Digital Converters, or ADCs, convert the analog signals from the thermistors, which monitor temperatures of the components in the ADCS

[1] ADCS-011
[2] ADCS-010
[3] ADCS-006
[4] ADCS-009
[5] ADCS-003

subsystem, and the current passing through the magnetorquers into a digital, easier to read digital signal.

#### 2.1.7 GPS

The GPS provides the satellite's latitude and longitude coordinates above earth.[6] The GPS connects to an antenna on the +X panel.

### 2.2 Schematic

#### 2.2.1 Power Rails

Page 2 of the schematic illustrates the power rails the Avionics Board draws from. The ADCS subsystem draws power from 3.3V-2 and VBATT-5.

#### 2.2.2 ADCS Microcontroller

The ADCS microcontroller[7] (page 3) was chosen for its ease of programming, and lower power consumption. It communicates with the sensors through two I²C buses and a UART bus, and operates the coils by sending PWM signals to the three H-bridges. It communicates with the IHU through an I²C bus. Reprogramming in orbit is achieved by the In-Flight JTAG Reprogrammer (IFJR) through a JTAG bus.

#### 2.2.3 EEPROM

The two EEPROM[8] cards (page 4) serve as redundant, non-volatile memory for the ADCS microcontroller to store constants of the earth's magnetic field. To store a 16-bit value representing each point of latitude and longitude, 1 Mbit (128 Kbyte) was required. The EEPROM cards have the necessary I²C functionality.

#### 2.2.4 H-bridges

The H-bridges[9] (page 5) draw from VBATT-5 as well as 3.3V-2. VBATT-5 provides a maximum voltage of 4.1 volts. This allows the H-bridge to have a greater range of currents it can supply. The H-bridges are controlled by the ADCS microcontroller using PWM.

#### 2.2.5 IMUs

The IMUs[10] (page 4) communicate with the ADCS microcontroller using the required I²C communication protocol. There are three IMUs in the subsystem for parity.

#### 2.2.6 ADCS ADCs

The ADCs[11] (page 6) communicate with the ADCS microcontroller using the required I²C communication protocol. Two ADCs were necessary in the ADCS

---

[6] ADCS-004
[7] STM32L476RG
[8] M24M01-R
[9] DRV883x
[10] BNO055
[11] AD7291

subsystem in order to read every thermistor. Thermistors are supplied with a reference voltage supplied by the ADC that reads their signal.

### 2.2.7 GPS

The GPS[12] (page 7) sends the satellite's coordinates above earth via UART to the ADCS microprocessor.

### 2.2.8 Power & ADCS Jacks

The six Picolock jacks (page 11) provide the I²C connections for the six ADCs not on the Avionics board. The +X and -Z jacks also provide power to the ADCs they connect to. The power inputs (page 11) come from the backplane (from the Electrical Power Subsystem). The ADCS uses the 3.3V-2 power source.

### 2.2.9 I²C Bus

The ADCS has three I²C buses (page 3 B2, B3, & B5). Two are for communicating with sensors and ADCs, and one is to communicate with C&DH. On the ADCS buses, the ADCS Microcontroller is the master served by the attached devices

#### *2.2.9.1 EEPROMs*

There are two EEPROM cards connected to the ADCS Microcontroller. Both cards are located on the Avionics Board. The list of addresses follows:

| EEPROM | I²C Bus | I²C Address | E1 | E2 |
|---|---|---|---|---|
| EEPROM0 | ADCS_I2C0 | 0xA0 for A16 = 0<br>0xA2 for A16 = 1 | L | L |
| EEPROM1 | ADCS_I2C0 | 0xA4 for A16 = 0<br>0xA6 for A16 = 1 | H | L |

#### *2.2.9.2 IMUs*

There are three IMUs connected to the ADCS Microcontroller. All IMUs are located on the Avionics Board. The list of addresses follows:

| IMU | I²C Bus | I²C Address | COM3 |
|---|---|---|---|
| IMU0 | ADCS_I2C0 | 0x28 | L |
| IMU1 | ADCS_I2C0 | 0x29 | H |
| IMU2 | ADCS_I2C1 | 0x28 | L |

Note: Redundancies in addresses are accounted for by being on different busses

#### *2.2.9.3 ADCs*

There are eight IMUs connected to the ADCS Microcontroller. Two ADCs are located on the Avionics Board. The remaining 6 are each located on a different panel of the satellite. The list of addresses follows:

[12] Venus838FLPx-L

| ADC | $I^2C$ Bus | $I^2C$ Address | AS1 | AS0 |
|---|---|---|---|---|
| ADCS0 | ADCS_I2C0 | 0x20 | H | H |
| ADCS1 | ADCS_I2C0 | 0x22 | H | NC |
| +X | ADCS_I2C1 | 0x20 | H | H |
| -X | ADCS_I2C1 | 0x22 | H | NC |
| +Y | ADCS_I2C1 | 0x23 | H | L |
| -Y | ADCS_I2C1 | 0x2A | NC | NC |
| +Z | ADCS_I2C1 | 0x2B | NC | L |
| -Z | ADCS_I2C1 | 0x2C | L | H |

Note: Redundancies in addresses are accounted for by being on different busses

## 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 – ADCS_[JTCK, JTDI, JTDO, JTMS]*

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

#### *2.3.1.2 $I^2C$ – ADCS_I2C0_[SDA, SCL], ADCS_I2C1_[SDA, SCL], BUS_I2C0_[SDA, SCL, IRQ]*

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

#### 2.3.1.3 UART – GPS_[TX, RX]

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

#### 2.3.1.4 Power Traces – VBATT-5

PGND applies to between the H-bridges and the backplane

Length: Each node shall be length matched $\pm 40.0mm$
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-2 | | | ADCS_I²C0_SDA | |
| VBATT-5 | | | ADCS_I²C1_SCL | |
| AVDD-0 | | | ADCS_I²C1_SDA | |
| AVREF-0 | | | GPS_TX | |
| ADCS_BUS_SCL | | | GPS_RX | |
| ADCS_BUS_SDA | | | ADCS_JTDI | |
| ADCS_I²C0_SCL | | | ADCS_JTMS | |

| Node | Resistance | | Node | Resistance |
|---|---|---|---|---|
| ADCS_JTCK | | | ADCS_JTDO | |

#### 3.1.3 Test Notes

### 3.2 Command Response

**Configuration:**

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

#### 3.2.1 Test Instructions

Send a signal on the BUS_I²C bus, use ADCS microcontroller to respond.

#### 3.2.2 Test Data

| Check if data is sent by ADCS microcontroller when requested on BUS_I²C | | | |
|---|---|---|---|
| Device | Signal Received | Passing Criteria | Pass / Fail |
| ADCS microcontroller | | *Data Received* | |

#### 3.2.3 Test Notes

### 3.3 Orientation Determination

**Configuration:**

This test evaluates the functionality of the orientation determination system.

#### 3.3.1 Test Instructions

Rotate IMUs and record changes in Euler angles and angular rate. Move the IMUs and record changes in acceleration. Lift and lower IMUs and record changes in gravity vector.

#### 3.3.2 Test Data

#### 3.3.3 Test Notes

### 3.4 Location Determination

**Configuration:**

This test evaluates the functionality of the location determination system.

#### 3.4.1 Test Instructions

Move GPS around town, record data, compare to actual coordinates.

#### 3.4.2 Test Data

| Check if data from GPS is within acceptable error of actual | | | | | |
|---|---|---|---|---|---|
| Location Tested | Coordinates Measured | Coordinates Actual | Error | Passing Criteria | Pass / Fail |
| | | | | | |
| | | | | | |
| | | | | | |

#### 3.4.3 Test Notes

## 3.5 Storage

**Configuration:**

This test evaluates the functionality of accessing storage.

#### 3.5.1 Test Instructions

Write test package to both EEPROM and read back both test packages.

#### 3.5.2 Test Data

| Check if package sent is received when read | | | |
|---|---|---|---|
| Device | Package Received | Passing Criteria | Pass / Fail |
| EEPROM0 | | *Package Received* | |
| EEPROM1 | | *Package Received* | |

#### 3.5.3 Test Notes

## 3.6 Current Monitoring

**Configuration:**

This test evaluates the functionality of the current monitoring system.

#### 3.6.1 Test Instructions

Supply a varying current using the ADCS Microcontroller and the H-bridge. Read resultant current through ADC.

#### 3.6.2 Test Data

| Check if current read by the ADC is the same as being supplied | | | | |
|---|---|---|---|---|
| Current Supplied | Current Read | Error | Passing Criteria | Pass / Fail |
| 60 mA | | | | |
| 120 mA | | | | |
| 180 mA | | | | |

#### 3.6.3 Test Notes

## 3.7 Temperature Monitoring

**Configuration:**

This test evaluates the functionality of the temperature monitoring system.

#### 3.7.1 Test Instructions

Heat thermistor location. Read resultant voltage through ADC.

#### 3.7.2 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 |
| ADCS Microcontroller | | | | $Error < 2\,°C$ | |
| H-bridge X | | | | $Error < 2\,°C$ | |
| H-bridge Y | | | | $Error < 2\,°C$ | |
| GPS | | | | $Error < 2\,°C$ | |
| IMU0 | | | | $Error < 2\,°C$ | |
| IMU1 | | | | $Error < 2\,°C$ | |
| IMU2 | | | | $Error < 2\,°C$ | |

#### 3.7.3 Test Notes

## 3.8 Light Monitoring

**Configuration:**

This test evaluated the functionality of the light monitoring system.

#### 3.8.1 Test Instructions

Shine light on each photodiode. Read resultant currents through ADC.

#### 3.8.2 Test Data

| Check if current read by the ADC is the same as expected | | | | | |
|---|---|---|---|---|---|
| Photodiode | Light Supplied | Current Read | Error | Passing Criteria | Pass / Fail |
| +X | | | | | |
| +X | | | | | |
| +X | | | | | |
| +Y | | | | | |
| +Y | | | | | |
| +Y | | | | | |
| +Z | | | | | |
| +Z | | | | | |
| +Z | | | | | |
| -X | | | | | |
| -X | | | | | |
| -X | | | | | |
| -Y | | | | | |
| -Y | | | | | |
| -Y | | | | | |
| -Z | | | | | |
| -Z | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| -Z | | | | | |

### 3.8.3 Test Notes

## 3.9 ADCS Programming

**Configuration:**

This test evaluates the ability to program the ADCS microcontroller.

### 3.9.1 Test Instructions

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

Note: Follow the programming instructions on the wiki.

### 3.9.2 Test Data

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

### 3.9.3 Test Notes