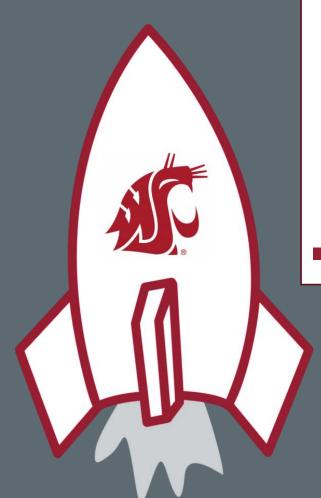
This document explains how the ADCS and the IHU are communication down to the byte by byte level.

# **ADCS** Interface

Detailed Description of the ADCS and IHU Interface

Revision: 1.0.4

Elijah Craig



## ADCS Interface

# Detailed Description of the ADCS and IHU Interface

# **Table of Contents**

1	Hardware Layer	2
	Software Layer	
	2.1 Location Data Request	
	2.2 Orientation Data Request	
	2.3 Temperature Data Request	
	2.4 Pulse Width Modulation Data Request	
	2.5 Current Data Request	4
	2.6 Orientation Command	4
3	Example Cases	4
	3.1 Orient away from Earth	





# 1 Hardware Layer

The hardware interface between the ADCS and the IHU is  $I^2C$ . It is a standard implementation of  $I^2C$  with a clock wire and a data wire. See <u>Wikipedia's  $I^2C$  article</u> for details on how it works.





## 2 Software Layer

The Attitude Determination Control System, located at 8b address 0xAC, is a slave board to the IHU. The IHU will request data from the ADCS to populate a telemetry packet. The ADCS data will be read at Upon receiving multiple commands, the reading done by ADCS will reflect the most recent command.

#### 2.1 Location Data Request

The latitude and longitude are represented by 32b signed integers with 100µmin/LSB. For more information on latitude and longitude, please read Wikipedia's page on the <u>Geographic Coordinate System.</u>

Index	0x00	0x01
Function	0×00:	0x00: Latitude 0x01: Longitude

#### 2.2 Orientation Data Request

Roll, Pitch, and Yaw are 16b unsigned integers. Zero represents  $0^{\circ}$ , and  $2^{16}$  represents  $2\pi$ .

Index	0x00	0x01
Function	0×01:	0x00: Roll
		0x01: Pitch
		0x02: Yaw

## 2.3 Temperature Data Request

This command is used by the IHU when building a telemetry packet. The temperature being represented with an 8b signed integer with 1°C/LSB.

Index	0x00	0x01
Function	0×02:	0x00: ADCS Temp

## 2.4 Pulse Width Modulation Data Request

The PWM is represented by a 16b unsigned integer.

Index	0x00	0x01
Function	0×03:	0x00: X PWM Out 0x01: Y PWM Out 0x02: Z PWM Out





#### 2.5 Current Data Request

The current is represented by a 16b signed integer with 150µA/LSB.

Index	0x00	0×01
Function	0x04:	0x00: X Current
		0x01: Y Current
		0x02: Z Current

#### 2.6 Orientation Commands

These are the general rotation commands the satellite will use to orient itself when aiming towards a location on Earth, a point in space, to position solar panels, or to protect the satellite from temperature related damage.

Index	0×00	0x01
Function	0×05	0x00: Roll
		0x01: Pitch
		0x02: Yaw

#### 2.6.1 Satellite Maneuvers for Earth

These commands pass in a latitude and a longitude as two 32b signed integers to find and maintain a fixed direction towards a location on Earth.

Index	0×00	0x01
Function	0x06	0x00: Point at coordinate 0x01: Point at ground station

#### 2.6.2 Satellite Maneuvers for Space Locations

For more information about the Equatorial Coordinate System, please go to Wikipedia's page on the subject.

Index	0×00	0x01
Function	0×07	0x00: Point [SIDE] at the Sun
		0x01: Aim solar panels
		towards Sun
		0x02: Point at equatorial
		location [RA, DEC]





#### **ADCS** Interface

## Detailed Description of the ADCS and IHU Interface

#### 2.6.3 Other Satellite Maneuvers

These functions are primarily for maintaining the health of the satellite. The function "Roast the Chicken" behaves similarly to a rotisserie, allowing each side to have protection and exposure to the Sun. Drifting, on the other hand ceases work of the ADCS stabilizers, and lets it tumble freely.

Index	0×00	0x01
Function	0x08	0x00: Roast the Chicken
		0x01: Point [FACE] towards
		Sun
		0x02: Point [FACE] away
		from sun
		0x03: Drift





## 3 Example Communication

#### 3.1 Simple Data Request for ADCS temperature

IHU: [0xAC]0x0200 [ADCS write] temperature? ADCS

ADCS reads temperature data and stores value in its buffer

IHU: [0x0F] [ADCS read]

ADCS:  $0 \times BFEC$  (40 \* 1°C/LSB = 40°C)

#### 3.2 Simple Repeated Data Request

IHU: [0xAC]0x0301 [ADCS write] PWM of Y

IHU: [0xAC]0x0200 [ADCS write] temperature? ADCS

ADCS reads temperature data and stores value in its buffer

IHU: [0x0F] [ADCS read]

ADCS:  $0 \times BFEC$  (40 \* 1°C/LSB = 40°C)



