



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

ADCS Interface

Detailed Description of the ADCS
and IHU Interface

Revision: 1.0.9

Elijah Craig

Table of Contents

1 Hardware Layer	3
2 Software Layer	4
2.1 Location Data Request	4
2.1.1 IHU to ADCS.....	4
2.1.2 ADCS to IHU.....	4
2.2 Orientation Data Request	4
2.2.1 IHU to ADCS.....	4
2.2.2 ADCS to IHU.....	4
2.3 Temperature Data Request.....	5
2.3.1 IHU to ADCS.....	5
2.3.2 ADCS to IHU.....	5
2.4 Coil Control Data Request.....	5
2.4.1 IHU to ADCS.....	5
2.4.2 ADCS to IHU.....	5
2.5 Coil Current Data Request.....	5
2.5.1 IHU to ADCS.....	5
2.5.2 ADCS to IHU.....	6
2.6 Orientation Commands.....	6
2.6.1 IHU to ADCS.....	6
2.6.2 ADCS to IHU.....	6
2.7 Satellite Maneuvers for Earth	6
2.7.1 IHU to ADCS.....	6
2.7.2 ADCS to IHU.....	6
2.8 Satellite Maneuvers for Space	7
2.8.1 IHU to ADCS.....	7
2.8.2 ADCS to IHU.....	7
2.9 Roast the Chicken	7
2.9.1 IHU to ADCS.....	7
2.9.2 ADCS to IHU.....	7
2.10 Drift	7

2.10.1 IHU to ADCS	7
2.10.2 ADCS to IHU	8
2.11 Two Line Elements	8
2.11.1 IHU to ADCS	8
2.11.2 ADCS to IHU	8
3 Example Communication	9
3.1 Data Request for Location	9
3.2 Repeated Data Request	9
3.3 Maneuver Request.....	9

1 Hardware Layer

The hardware interface between the ADCS and the IHU is I²C. It is a standard implementation of I²C with a clock wire and a data wire. See [Wikipedia's I²C article](#) 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 ADCS senses and controls the satellites orientation when in orbit. The IHU will periodically request data from the ADCS to prepare an updated telemetry packet. The ADCS data will be read at 0xAD. 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 geographic coordinates¹, represented by 32b signed integers with 100 μ m/LSB.

2.1.1 IHU to ADCS

The first byte represents the command ID, 0x00.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x00							

2.1.2 ADCS to IHU

The ADCS replies with an array of two geographic coordinates.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	Latitude Coordinate				Longitude Coordinate			

2.2 Orientation Data Request

Roll, Pitch, and Yaw are all angles, represented by 16b unsigned integers where zero indicates 0, and 2^{16} indicates 2π .

2.2.1 IHU to ADCS

The first byte represents the command ID, 0x01.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x01							

2.2.2 ADCS to IHU

ADCS replies with an array of three 16b unsigned integers.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
-------------	------	------	------	------	------	------	------	------

¹

<http://edndoc.esri.com/arcscde/9.2/concepts/geometry/coordref/coordsys/geographic/geographic.htm>

0x00	Roll Angle	Pitch Angle	Yaw Angle		
------	------------	-------------	-----------	--	--

2.3 Temperature Data Request

The temperatures are represented with an 8b signed integer with 1°C/LSB.

2.3.1 IHU to ADCS

The first byte represents the command ID, 0x02.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x02							

2.3.2 ADCS to IHU

The ADCS replies with an array of five 8b signed integers.

Byte Offset	0x00	0x01	0x02	0x03
0x00	Coil X Driver Temperature	Coil Y Driver Temperature	Coil Z Driver Temperature	ADCS (μController) Temperature
0x04	GPS Temperature			

2.4 Coil Control Data Request

The PWM is represented by a 16b unsigned integer.

2.4.1 IHU to ADCS

The first byte represents the command ID, 0x03.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x03							

2.4.2 ADCS to IHU

The ADCS replies with an array of three 16b integers.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	Coil X Driver PWM Out	Coil Y Driver PWM Out	Coil Z Driver PWM Out					

2.5 Coil Current Data Request

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

2.5.1 IHU to ADCS

The first byte represents the command ID, 0x04.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07

0x00	0x04							
------	------	--	--	--	--	--	--	--

2.5.2 ADCS to IHU

The ADCS replies with an array of three 16b currents.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	Coil X Driver Current		Coil Y Driver Current		Coil Z Driver Current			

2.6 Orientation Commands

This is the general rotation command using aircraft principle axes. The ADCS will maintain the request until the [drift](#) request is sent.

2.6.1 IHU to ADCS

The first byte is the command ID, 0x05, and all three parameters are angles, as defined in [Section 2.2](#).

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x05	Roll		Pitch		Yaw		

2.6.2 ADCS to IHU

The ADCS replies with an error code defined in Code SOP 6.4.1².

2.7 Satellite Maneuvers for Earth

The ADCS will maintain the request until the [drift](#) request is sent.

2.7.1 IHU to ADCS

There are two command IDs, each one associated with pointing either the camera or antenna at the point. This command passes in a latitude and a longitude as two 32b signed integers, with geographic coordinates found in [Section 2.1](#), to find and maintain a fixed point towards a location on Earth.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x06: Camera 0x07: Antenna	Latitude				Longitude [31:8]		
0x08	[7:0]							

2.7.2 ADCS to IHU

The ADCS replies with an error code defined in Code SOP 6.4.1³.

²<https://github.com/CougsInSpace/Resources/blob/master/StandardOperatingProcedures/Code.pdf>

2.8 Satellite Maneuvers for Space

This command passes in an 8b unsigned integer, and two 32b signed integer with 100 μ m/LSB representing the equatorial coordinate⁴. The ADCS will maintain the request until the [drift](#) request is sent.

2.8.1 IHU to ADCS

The first byte is the command ID, the parameters are the respective equatorial coordinates.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x07: Camera	Right Ascension				Declination [31:8]		
0x08	[7:0]							

2.8.2 ADCS to IHU

The ADCS replies with an error code defined in Code SOP 6.4.1⁵.

2.9 Roast the Chicken

The first byte represents the command ID, 0x08. These functions are primarily for maintaining the health of the satellite. The function "Roast the Chicken" behaves similarly to a rotisserie, evenly distributing the received power from the sun, aiding to normalize temperatures.

2.9.1 IHU to ADCS

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x08							

2.9.2 ADCS to IHU

The ADCS replied with an error code, defined in Code SOP 6.4.1⁶, indicating the satellite is now executing the maneuver "Roast the Chicken".

2.10 Drift

The drifting command halts the work of the ADCS's X, Y, and Z coils, and lets it tumble freely.

2.10.1 IHU to ADCS

The first byte represents the command ID, 0x09.

³ ibid

⁴ <https://lco.global/spacebook/equatorial-coordinate-system/>

⁵ <https://github.com/CougsInSpace/Resources/blob/master/StandardOperatingProcedures/Code.pdf>

⁶ ibid

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x09							

2.10.2 ADCS to IHU

The ADCS replies with an error code defined in Code SOP 6.4.1⁷.

2.11 Two Line Elements

The Keplerian orbital elements are transferred in the format of two-line elements (TLE)⁸.

2.11.1 IHU to ADCS

The first byte represents the command ID, the second is string length, and the remaining bytes contain the data in ASCII. The string does not include the null character.

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x10	String Length	Data	Data	Data	Data	Data	Data...

2.11.2 ADCS to IHU

The ADCS replies with an error code defined in Code SOP 6.4.1⁹.

⁷ ibid

⁸ <https://www.celestrak.com/NORAD/documentation/tle-fmt.php>

⁹ <https://github.com/CougsInSpace/Resources/blob/master/StandardOperatingProcedures/Code.pdf>

3 Example Communication

3.1 Data Request for Location

IHU: [0xAC]0x00 [ADCS write] location request
ADCS reads location data and stores it in its buffer
IHU: [0xAD] [ADCS read]
ADCS: 0x01ABD591FBCF4757 46.7309088°N, -117.1697756°E

3.2 Repeated Data Request

IHU: [0xAC]0x02 [ADCS write] temperature request
ADCS reads the temperature and stores it in its buffer
IHU: [0xAC]0x00 [ADCS write] location request
ADCS reads location data and stores it in its buffer
IHU: [0xAD] [ADCS read]
ADCS: 0x01ABD591FBCF4757 46.7309088°N, -117.1697756°E

3.3 Maneuver Request

IHU: [0xAC] 0x07008637B2FDD30C55 [ADCS write] maneuver request
RA: 210.660137° Dec: -59.166007°
ADCS reads requests and performs appropriately
IHU: [0xAD] [ADCS read]
ADCS: 0x00 Success