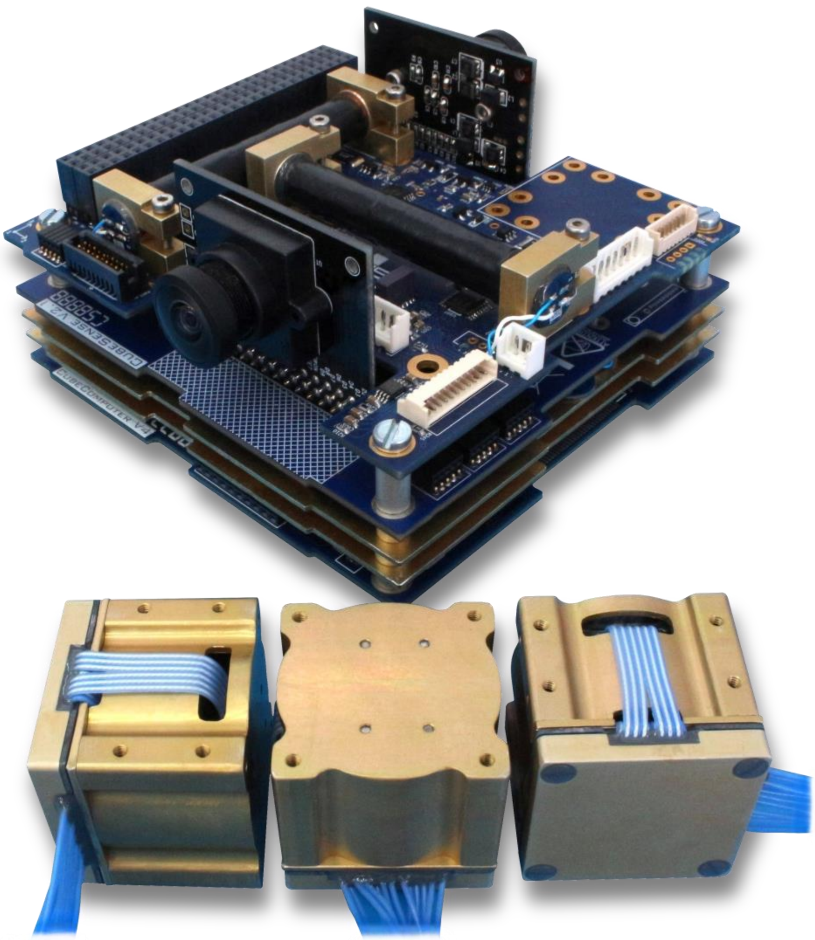
CubeADCS

The complete ADCS solution



Interface Control Document

Electronic Systems Laboratory

Contact Us

Phone : +27 21 808 9499
E-mail : info@cubespace.co.za
Web : www.cubespace.co.za
Facebook : /CubeSpaceADCS
Twitter : @CubeSpace_ADCS

Physical Address

CubeSpace
The LaunchLab
Hammanshand Road
Stellenbosch, 7600
South Africa

Universiteit Stellenbosch University

Document	CubeADCS Interface Control Document
Version	3.05
Domain	Public
Date modified	29 March 2018
Approved by	Name: Mike-Alec Kearney
	Signature:

Document Version History

Version	Responsible person(s)	Pages	Date	Description of change
3.0	MK	ALL	20/03/2017	V3 First draft
3.01	MK	ALL	04/07/2017	Merge 3-Axis & Y-Momentum
3.02	MK	27,28	04/07/2017	Added CAN details
3.03	CJG	All	26/07/2017	General Updates
3.04	MK	All	27/07/2017	Accept/Reject team changes
3.05	MK	24 26	31/10/2017	Fixed incorrect I2C address Changed Table 8 to correct TLM (240)
3.06	CJG	32	02/02/2018	Removed Command 70
3.07	CJG	12	12/02/2018	Added description for the SPI bus
3.08	CCH	All	29/03/2018	General language editing

List of Acronyms/Abbreviations

ACP	ADCS Control Program
ADCS	Attitude Determination and Control System
CSS	Coarse Sun Sensor
ESD	Electrostatic Discharge
I^2C	Inter-Integrated Circuit
MCU	Microcontroller Unit
MEMS	Microelectromechanical System
OBC	Onboard Computer
PCB	Printed Circuit Board
SBC	Satellite Body Coordinate
TC	Telecommand
TLM	Telemetry
UART	Universal Asynchronous Receiver/Transmitter

Table of Contents

1. Introduction

CubeADCS bundles provide attitude sensing and control capabilities to nanosatellites. These units consist of several CubeSpace modules integrated into a compact bundle. The ADCS onboard computer (OBC) can also serve as the main satellite OBC if necessary and a comprehensive attitude determination and control system (ADCS) software library can be provided with the CubeADCS bundle.

This document describes the characteristics of the CubeADCS unit as well as the mechanical and electrical interfaces to the bundle. CubeADCS units can be configured with different sensor and actuators for different mission requirements. This document covers all ADCS configurations and readers are to use their discretion in determining which parts are relevant to their various bundle configurations.

If the reader has doubt over which parts are relevant, please mail us at ***info@cubespace.co.za***.

2. Functional Description

2.1 System components

The CubeADCS unit can use a combination of magnetometer, coarse Sun sensor, fine Sun sensor, nadir sensor, MEMS rate sensor, and star tracker measurements to estimate the attitude of the satellite. Magnetorquers and reaction wheels are used to stabilize and control the satellite's attitude.

A CubeADCS unit consists of **up to four** integrated PC104-standard PCBs and several peripheral components, which are to be mounted separately. A basic diagram of the **complete** CubeADCS solution with all options is displayed in Figure 1.

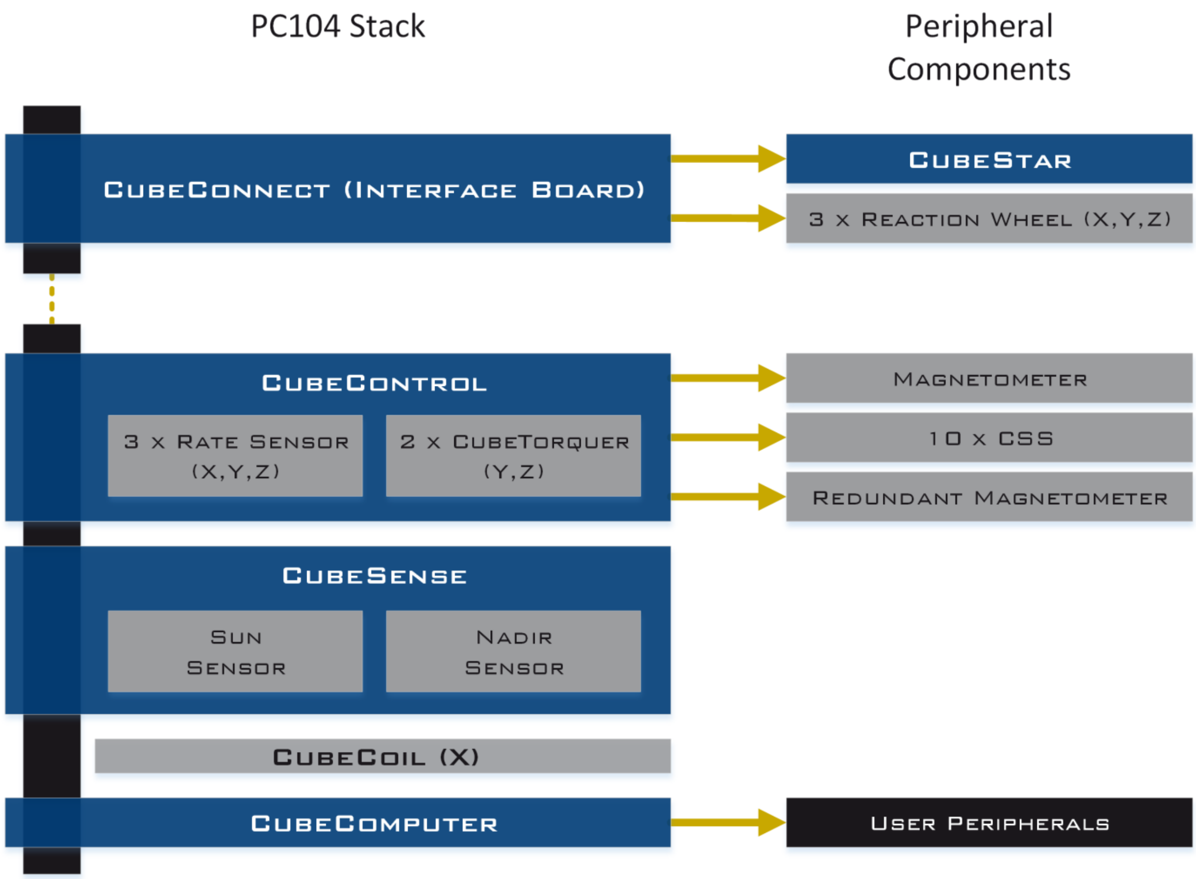
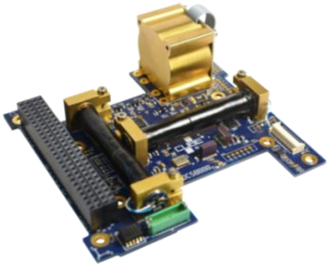
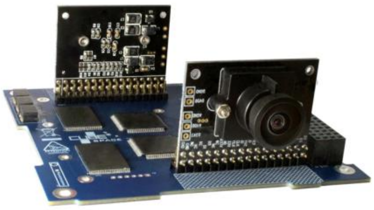
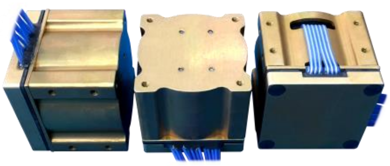
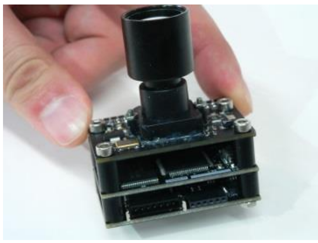


Figure 1 – System diagram of the complete CubeADCS solution.

A brief description of the core components of the CubeADCS is given in Table 1.

Table 1 – Modules of the CubeADCS solution.

Module	Description
CubeControl 	CubeControl is an actuator and sensor interface module for nanosatellites with advanced attitude control requirements. It is capable of controlling 3 magnetorquers and a momentum wheel. CubeControl can also interface with 2 magnetometers, 10 coarse Sun sensors, and 3 MEMS rate sensors. **NB: The momentum wheel mounted on CubeControl is not included in a CubeADCS 3-Axis bundle, but only in Y-Momentum**
CubeSense 	CubeSense is an integrated Sun and nadir sensor for attitude sensing. It makes use of two CMOS cameras – one dedicated to Sun sensing and another for horizon detection. The Sun sensor has a neutral density filter included in the optics. Both cameras have a field of view (FOV) of approximately 160°.
CubeComputer 	CubeComputer is a generic CubeSat OBC. It can perform the required ADCS functions, as well as the satellite's main OBC tasks. The module is based on ARM Cortex-M3 architecture and it also implements error detection and correction (EDAC) techniques.
CubeWheel 	CubeWheel is a compact standalone reaction wheel unit for nanosatellites. It provides the satellite with the ability to achieve 3-axis stability and 3-axis control. Each CubeWheel is magnetically shielded and is mountable in 3 axes. Various sizes are available to suit every need.
CubeStar 	CubeStar is a compact star tracker for nanosatellites. The module is based on ARM Cortex-M3 architecture and has significantly low power consumption.

2.2 Body axes definition

The coordinate system definition used by the CubeADCS bundle is shown in Figure 2, while the orbit coordinate system is displayed in Figure 3. The ADCS coordinate system is furthermore related to the satellite body coordinate (SBC) system through a transformation matrix. When the ADCS is controlling the attitude to zero roll, pitch, and yaw angles, the SBC system will coincide with the orbit coordinate system (referred to as the nominal orientation).

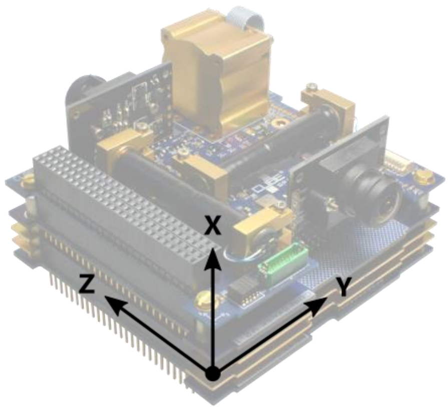


Figure 2 – ADCS coordinate system.

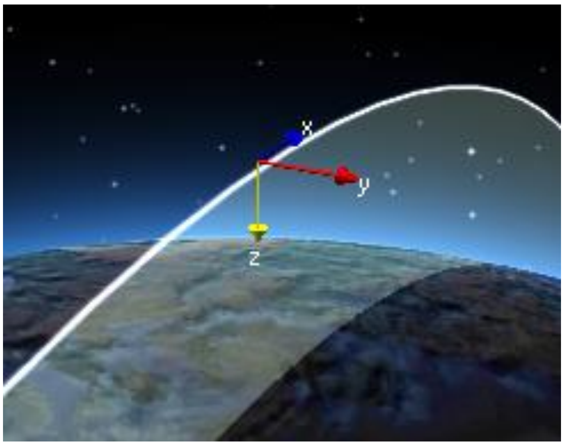


Figure 3 – Orbit coordinate system.

2.3 ADCS control loop

The ADCS controllers and estimators, as well as all internal communications inside the ADCS bundle, are executed by an Attitude Control Program (ACP) that runs on CubeComputer. This program runs a control loop at 1Hz.

If the ADCS bundle is configured to have the CubeComputer 'unlocked' to run flight software and serve as main OBC, the 'ADCS update' function in the ADCS software library must always be executed with the highest priority every 1s.

3. Electrical Interface

3.1 PC104 interface

The CubeADCS unit makes use of the standard PC104 header for electrical interfacing. The pin description of the PC104 bus is shown in Figure 4, as used by the CubeADCS 3-Axis unit.

H2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51
H1	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51

PC104 interface pins					
Communication					
	H1	1	CANL	CAN bus low	*option*
	H1	3	CANH	CAN bus high	*option*
	H1	21	I2C_SCL_ADCS	Internal I2C clock for all ADCS modules	required
	H1	23	I2C_SDA_ADCS	Internal I2C data for all ADCS modules	required
	H1	41	I2C_SDA_SYS	System I2C data for CubeComputer	required
	H1	43	I2C_SCL_SYS	System I2C clock for CubeComputer	required
	H1	17, 18, 19, 20	UART_1	Usable pins for UART_1 (RX or TX)	*option*
	H2	21, 22	UART_1	Usable pins for UART_1 (RX or TX)	*option*
	H1	33, 35, 39, 40	UART_2	Usable pins for UART_2 (RX or TX)	*option*
	H1	29	SPI_CLK	SPI Clock	*option*
	H1	30, 31	SPI_MOSI / MISO	SPI MOSI or MISO	*option*
	H1	32	SPI_CS	SPI Chip Select	*option*
Power					
	H2	29, 30, 32	GND	Ground connection for all modules	required
	H2	45, 46	V_Bat	Battery voltage bus	required
	H2	25, 26	5V_Main	Main 5 V supply	*standard option*
	H2	27, 28	3V3_Main	Main 3.3 V supply	*standard option*
	H1	47, 49, 51	5V_S	Switched 5 V supply options	*option*
	H1	48, 50, 52	3V3_S	Switched 3.3 V supply options	*option*
	H1	42	BUVIN	CubeComputer optional backup power supply	*option*
Internal ADCS pins					
	H1	2, 4, 6, 8	ENABLE	Enable lines for CubeADCS modules position 1	*standard option*
	H2	17, 18, 19, 20	ENABLE	Enable lines for CubeADCS modules position 2	*option*
	H1	5, 7, 9, 11	ENABLE	Enable lines for CubeWheel 1 - 4	*option*
	H1	13, 14, 15, 16	ENABLE	GPIO/ ADC or Enable Lines	*option*
	H1	10	ENABLE	Enable Line for CubeStar position 1	*option*
	H2	15	ENABLE	Enable Line for CubeStar position 2	*option*

Figure 4 – PC104 bus pin description.

3.2 Power

3.2.1 Power supply

The ADCS should be supplied with 3.3 V, 5 V and the raw battery voltage, $V_{battery}$. The 3.3 V and 5 V buses must be supplied by one of H1-48, 50, 52 or H2-27, 28 (3.3 V) and one of H1-47, 49, 51 or H2-25, 26 (5 V) ***– depending on the selection made by user in the Option Sheet***. The 3.3 V and the 5 V supply must be switched on within 10ms of each other.

The ADCS has internal power switches for the various individual components (CubeSense, CubeControl, CubeStar, and the CubeWheels), all of which are controlled by CubeComputer. The switch states are available as telemetry (TLM) and can also be toggled using a telecommand (TC).

If CubeComputer is used as the satellite's main OBC (alongside its function as ADCS OBC) it must be supplied with 3.3 V on H2-27, 28 (the "always on" 3.3 V).

Currents and voltages are measured at various points on the CubeADCS unit and the measurements can be obtained using TLM requests. The ADCS will also use these measurements to switch off components in case of a latch-up.

The battery bus (pins H2-45 and H2-46) supplies power to the CubeWheels and the magnetometer boom deployment circuitry.

3.2.2 Battery bus voltage

The CubeWheels have a turn-on threshold of 6.5 V, but lower voltages are tolerated without damage to the units. The maximum tested bus voltage is 16.0 V.

3.3 Communication

3.3.1 I^{2}C

The CubeADCS unit can communicate with other satellite subsystems using the System I^{2}C on the PC104 bus (H1-41, 43). If CubeComputer is the main OBC of the satellite, it will act as a master on the system I^{2}C bus. Alternatively, it will act as a slave on the system I^{2}C bus if it is only acting as ADCS OBC, responding to commands and telemetry requests (as detailed in the CubeADCS User Manual). CubeComputer can be populated with or without pull-up resistors on the system I^{2}C bus.

3.3.2 UART

The CubeADCS bundle has two UART channels (on CubeComputer), designated UART 1 and UART 2, which can be used to interface with the bundle. Both channels can be connected to the PC104 header and UART 2 is also accessible from the piggyback header on CubeComputer.

CubeSpace provides PC testing software called CubeSupport. CubeSupport communicates with CubeComputer via UART 1, which is also accessible via a 3-way 2.54 mm pitch square post female header. A UART-to-USB cable is supplied with each CubeADCS unit, which enables the user to connect the unit with a PC.

3.3.3 CAN

The optional CAN interface on CubeComputer is only available on the main PC104 header (H1-1, 3). The combination of a CAN transceiver and a CAN controller module on CubeComputer allows CubeADCS unit to interface at CAN bus voltage levels of 3.3 V or 5 V. A termination resistor between the CANH and CANL lines can be populated, if required by the user, or it can be configured at order.

3.3.4 SPI

The CubeComputer has an open SPI module which can be connected to the PC104 header. This optional SPI connection's intended use is for an Unlocked ACP (See Section 5). This SPI bus will allow the CubeComputer to be connected to slave nodes.

The ACP is supplied with a board support package which contains examples of SPI drivers. The user can make use of these examples to customize the SPI driver to suit their specific needs.

Please note that unlike the I2C, UART and CAN, the SPI cannot be used as a means to communicate with CubeComputer (ACP). It is provided to expand the CubeComputer's capabilities for users who intend to use an Unlocked version of the ACP.

3.4 Logging

The CubeADCS bundle has the ability to log data automatically and save it to the SD memory card on CubeComputer. CubeComputer is supplied with an industrial grade 1 GB SD card. For more detail regarding the logging, please consult the *CubeADCS User Manual*.

3.5 Peripheral components

As shown in Figure 1, the magnetometers and coarse Sun sensors interface with CubeControl via wired connections. The CubeStar and three CubeWheels interface with CubeComputer via the CubeConnect interface board in the standard configuration. All of the abovementioned connections are polarised for ease of use and the harness lengths are all configurable.

4. Mechanical Interface

4.1 PC104 component stack

Most CubeSpace components make use of the standard “self-stacking” PC104 CubeSat mechanical interface. The dimensions of the stacked components and mounting locations are displayed in Figure 5. The CubeADCS stack includes spacers to support the inter-PCB spacing.

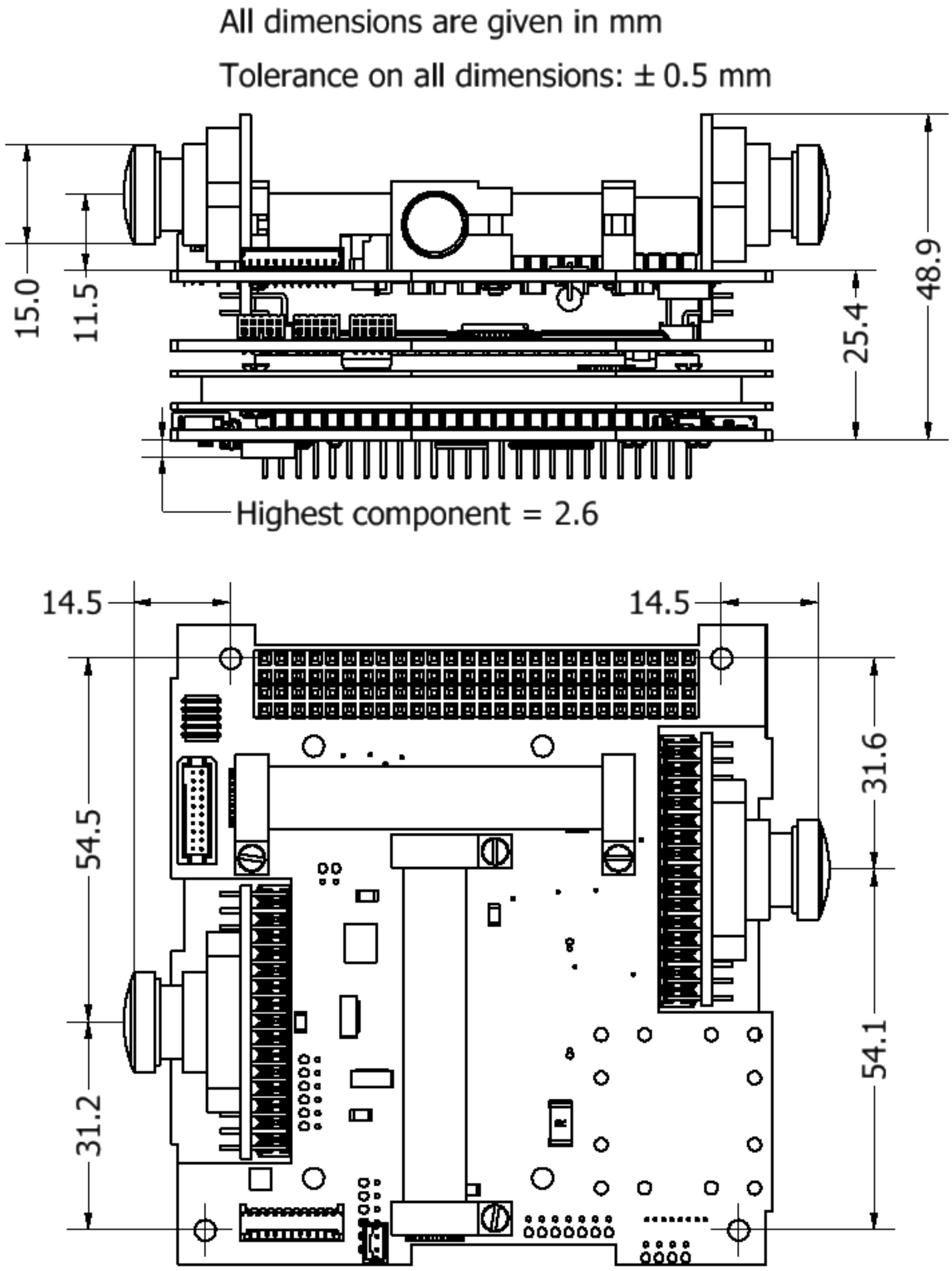


Figure 5 – Dimensions of the PC104 stack of the CubeADCS unit.

4.2 Reaction wheels

The mounting of the three CubeWheel reaction wheels will depend on the size of the CubeWheels. In the case of Small CubeWheels, the wheels can be (1) mounted on CubeConnect and integrated in the main PC104 stack; (2) mounted on CubeConnect and supplied on a separate PC104 stack, or (3) supplied loose, i.e. the user must mount the wheels.

4.2.1 Small CubeWheel

The Small CubeWheel unit has two sets of four M2 mounting holes on two different facets, as illustrated in Figure 6. The outer dimensions of the Small CubeWheel are 28 mm x 28 mm x 26.1 mm. Note that an additional 1.5 mm is required on one side of the CubeWheel for internal harnesses. The M2 mounting holes are fitted with A2 (304) stainless steel heli-coils to provide a stronger, more durable thread.

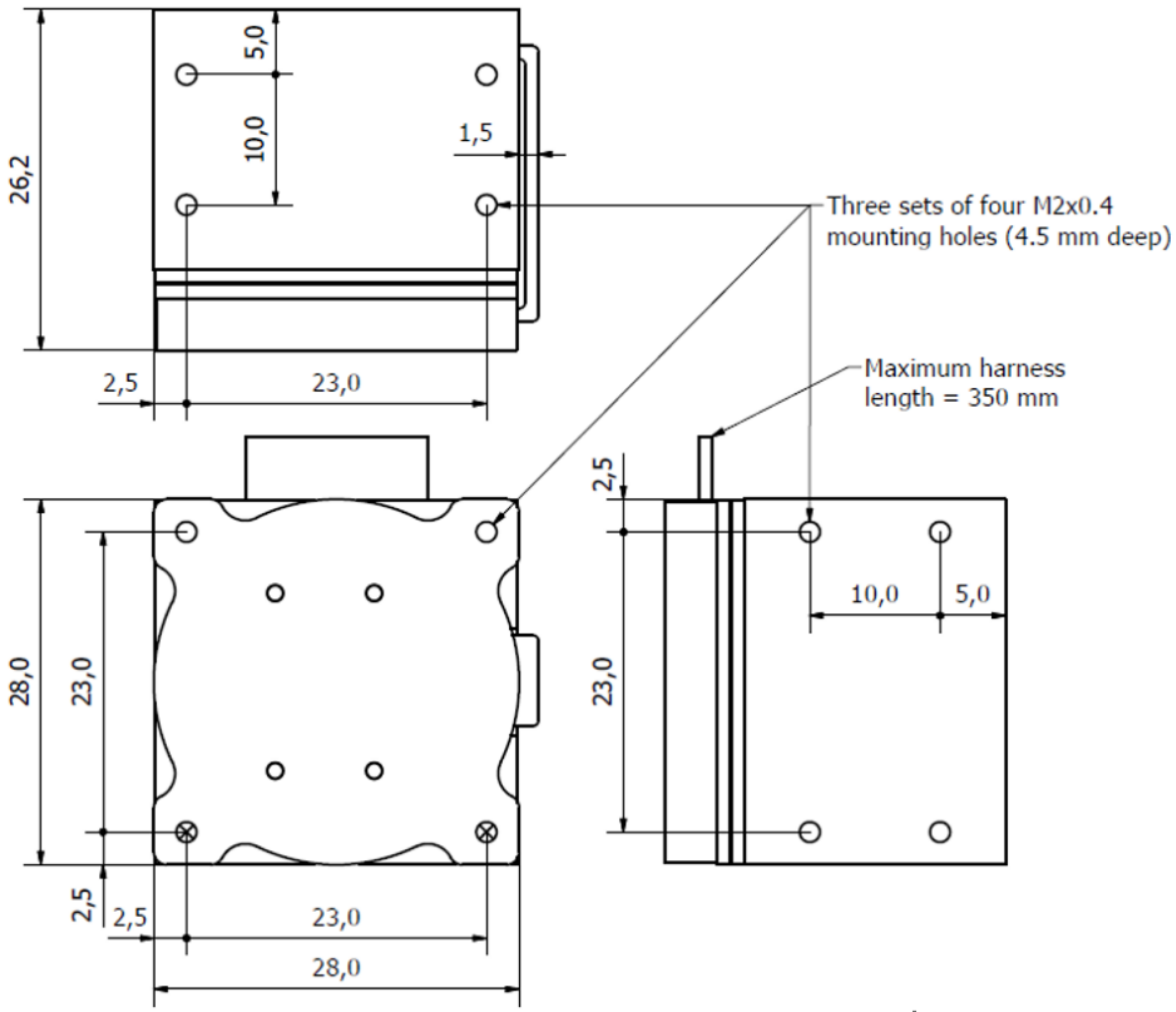


Figure 6 – Small CubeWheel mechanical interface.

4.2.2 Medium CubeWheel

The Medium CubeWheel unit has two sets of four M3 mounting holes on two different facets, as illustrated in Figure 7. The outer dimensions of the Medium CubeWheel are 46 mm x 46 mm x 31.5 mm. Note that an additional 1.5 mm is required on one side of the CubeWheel for internal harnesses. The M3 mounting holes are fitted with A2 (304) stainless steel heli-coils to provide a stronger, more durable thread.

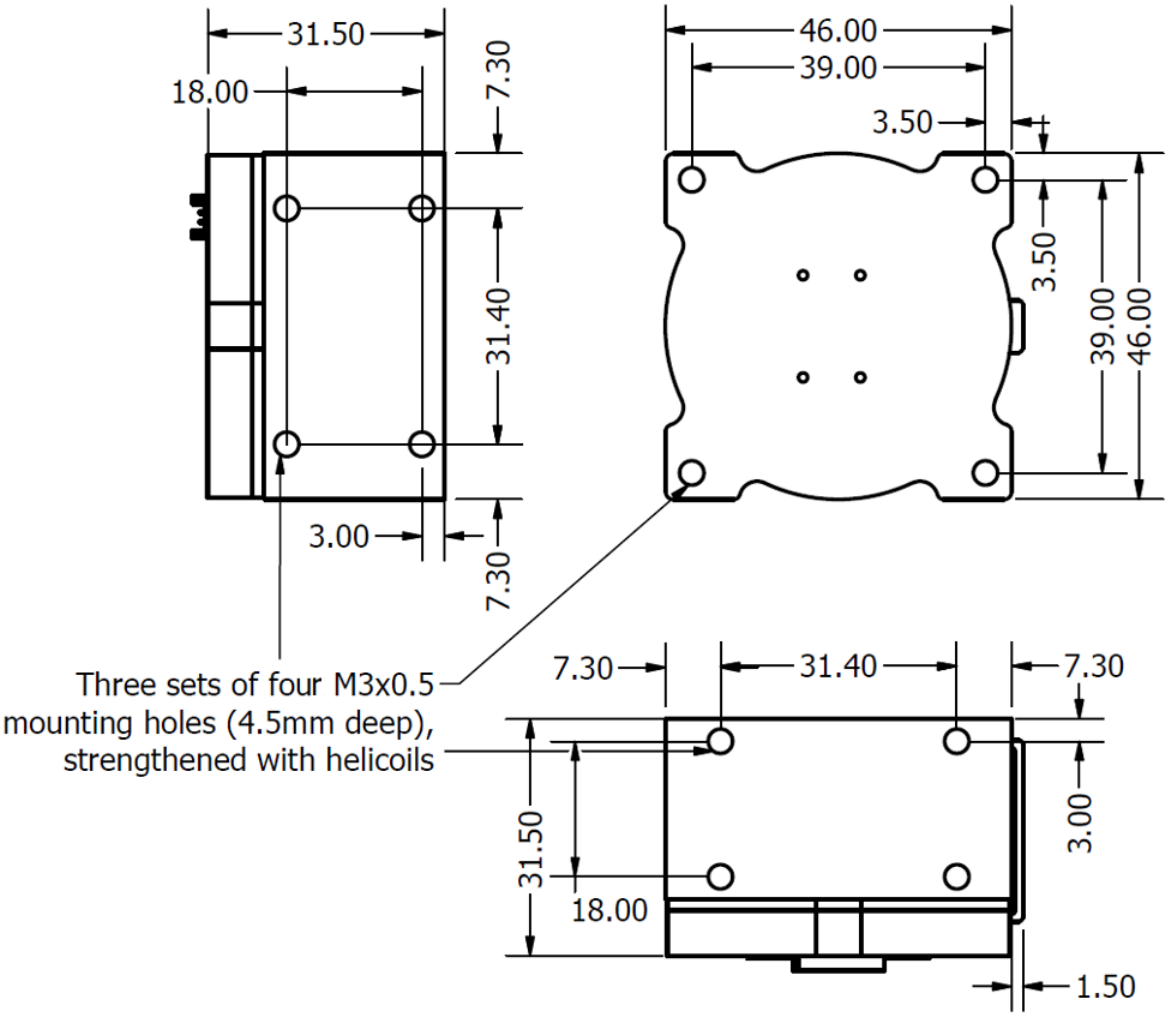


Figure 7 – Medium CubeWheel mechanical interface.

4.2.3 Large CubeWheel

The Large CubeWheel unit has two sets of four M3 mounting holes on two different facets, as illustrated in Figure 8. The outer dimensions of the Large CubeWheel are 57 mm x 57 mm x 31.5 mm. Note that an additional 1.5 mm is required on one side of the CubeWheel for internal harnesses. The M3 mounting holes are fitted with A2 (304) stainless steel heli-coils to provide a stronger, more durable thread.

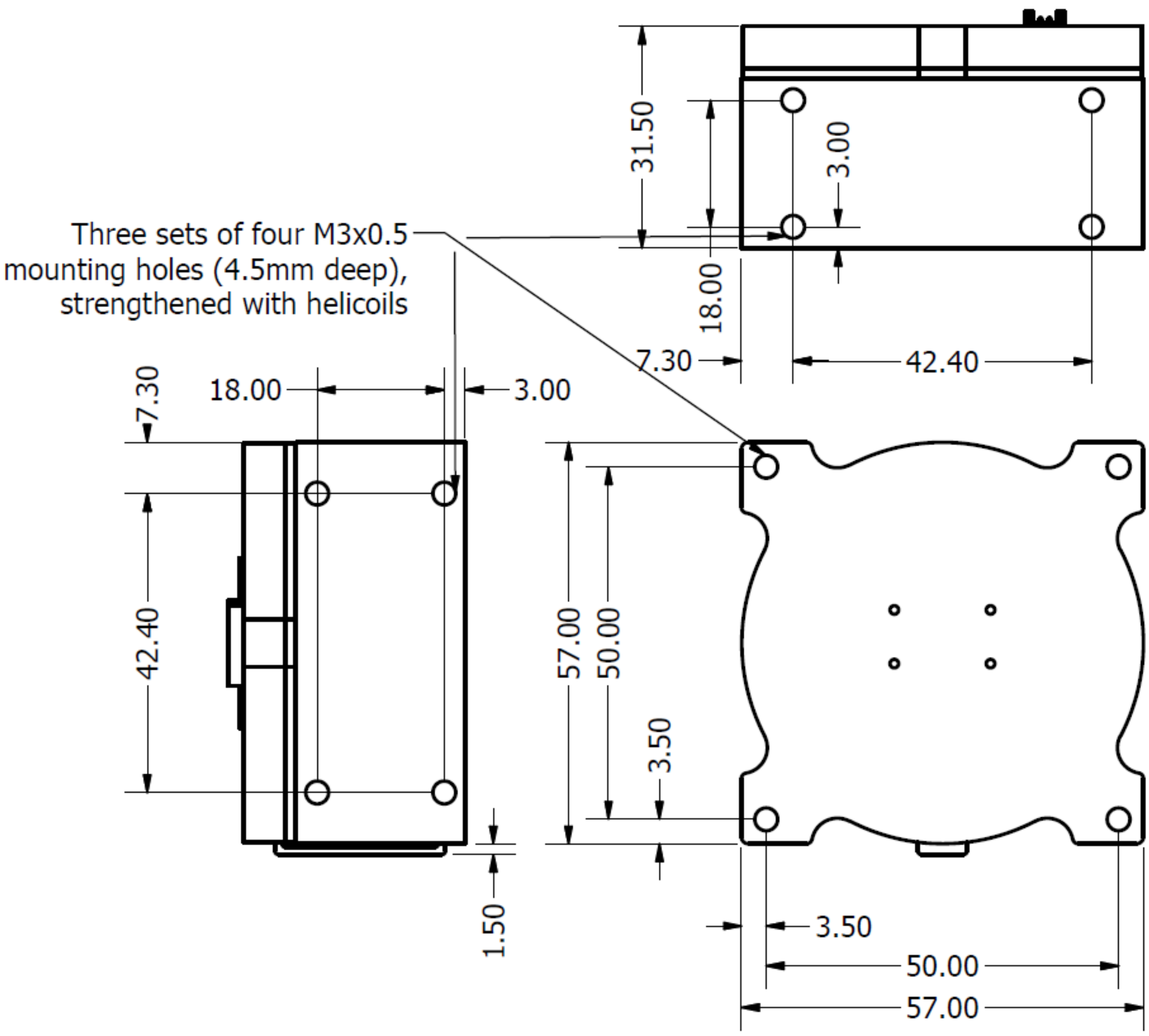


Figure 8 – Large CubeWheel mechanical interface.

4.3 Magnetometers

4.3.1 External deployable magnetometer

The primary magnetometer is housed on a small PCB, enclosed by an aluminum housing. The bare magnetometer housing measures 16.8 mm x 17.2 mm x 6.7 mm (as shown in Figure 9).

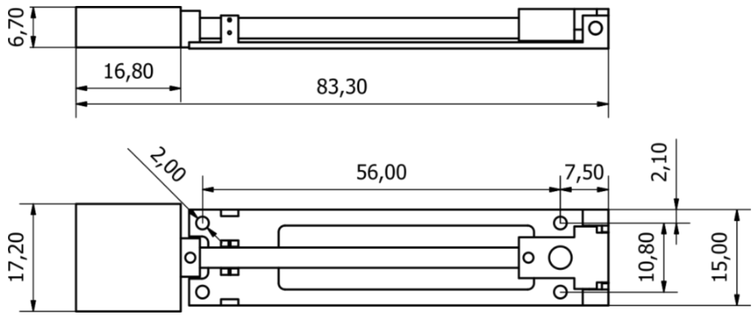


Figure 9 – Deployable magnetometer dimensions and mounting locations.

The magnetometer is supplied with a deployable boom by default. The boom separates the magnetometer 8 cm from the satellite body, limiting the effect of electromagnetic interference and lowering the measurement noise. Although the magnetometer can be supplied without the deployment mechanism, it is strongly recommended that the magnetometer should be deployable. This is owing to the magnetometer's important role during detumbling.

The deployable boom consists of a rod with the magnetometer in its enclosure at one end and a mounting bracket with a spring on the other, as illustrated in Figure 10. The bracket should be mounted on the outside of the satellite structure. The magnetometer will be supplied in the stowed position; deployment can be done in-flight via the appropriate TC.

The deployable magnetometer can NOT be refurbished, therefore a test deployment should not be done on the ground.

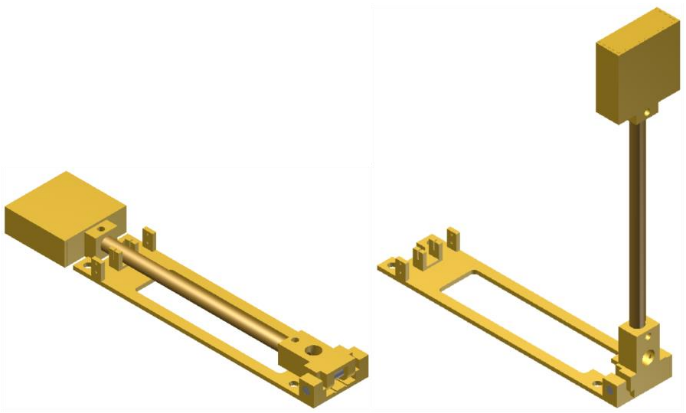


Figure 10 – Deployable magnetometer stowed (left) and deployed (right).

The primary magnetometer and the boom deployment actuation are connected to CubeControl using a single Omnetics Nano Circular 11-way in-line harness. The length between the CubeControl and the in-line connection, as well as the length between the magnetometer and the inline connection, can be configured to the user's requirements. The maximum total length of the harness is 500 mm. The in-line Omnetics connector has a diameter of 3.9 mm.

4.3.2 Redundant magnetometer

The CubeADCS unit offers an interface to an optional redundant magnetometer. The magnetometer is housed in the same enclosure as the primary magnetometer, but it is not deployable. The redundant magnetometer is connected to the CubeControl module via a 6-way harness (not in-line) and the maximum harness length is 400 mm.

4.4 Coarse Sun sensors

The CubeADCS unit can interface with up to ten coarse Sun sensor (CSS) photodiodes, each mounted on a small PCB. The dimensions of each CSS PCB are 3.8 mm x 10.8 mm and the height is 1.7 mm. The CSSs do not have mounting holes – they should be attached to the satellite body using epoxy. At least 6 photodiodes, each on a different facet of the satellite, are required to obtain a coarse Sun vector throughout the entire sunlit part of the orbit. The four additional sensors can be used for redundancy or in cases where sensors are shadowed or obstructed.

The ten CSSs are each connected to CubeControl by a Molex PicoBlade 2-way in-line connector to ease the satellite integration process. The ten 2-way harnesses are terminated in a single 20-way female connector, of which the male connector is populated on CubeControl. Figure 11 illustrates the above-mentioned wiring of the CSS units. The PCBs with the CSSs are connected to a short stub that is 50mm long. The long side of the harness can be configured to be up to 350mm, with 300mm being the default length.

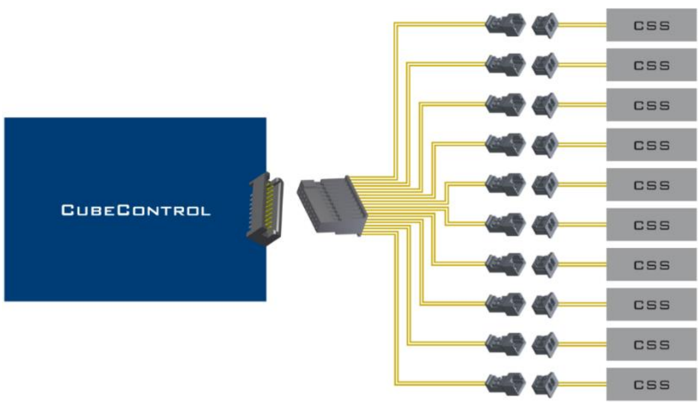


Figure 11 – Coarse Sun sensor wiring diagram.

4.5 Mass

The approximate masses of the CubeADCS components are shown in Table 2.

Table 2 – Mass of CubeADCS components.

Component	Mass (g)	Notes
PC104 components		
CubeControl V2.2	97	Including 2 CubeTorquer rods
CubeComputer V4.1	50	
CubeSense V2.2	79	Including 2 cameras
CubeCoil	46	
Spacers and epoxy	< 10	Depends on configuration
Peripheral components		
3 x Small CubeWheel	165	Excluding harnesses
3 x Medium CubeWheel	< 450	Excluding harnesses
3 x Large CubeWheel	< 660	Excluding harnesses
CubeConnect	46	
PC104 spacers for CubeConnect	28	Only applicable if Small CubeWheels are integrated in bundle
CubeStar	66	Including harness
Deployable magnetometer	15	Including harness and deployment mechanism
Redundant magnetometer	7	Including harness
10 x CSS	15	Including harnesses

5. Software Interface

5.1 CubeComputer software options

The computer in the CubeADCS unit can either run a standalone ADCS control program (ACP) or it can run an unlocked version of the ACP that allows for the addition of custom user functionality. The two options are illustrated in Figure 12.

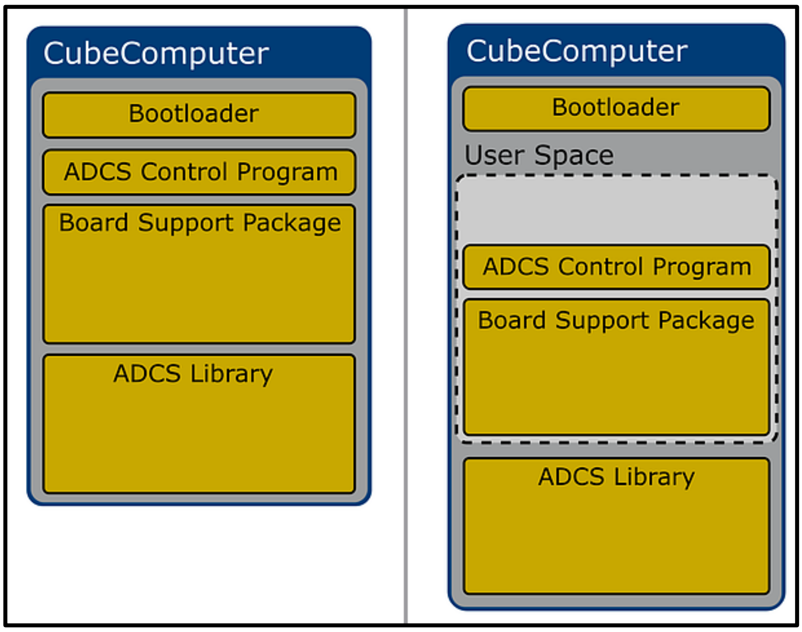


Figure 12 – CubeComputer software options.

In the standalone ACP case, the CubeComputer is commanded by a main OBC. The ADCS is controlled by sending telemetries and telecommands from the main OBC through one of the communication busses (I2C, UART or CAN). The complete listing of TCs and TLMs are shown in the Reference Manual.

In the other case, where the CubeComputer is unlocked for user development, the ADCS is controlled by making function calls to the compiled ADCS library. In this case the computer is typically used as ADCS, as well as main OBC in the satellite. For more information on the ADCS library's functions, see the *ADCS library Manual*.

5.2 Connecting with OBC

The ADCS stack can communicate using either a UART channel, an I^2C bus or a CAN bus. In all cases the telecommand and telemetry definitions are the same, but the protocol differs slightly.

Table 3 displays the meaning of the first byte of a message sent to the stack. This byte will determine whether the message is a telecommand or telemetry request, and it will also contain the ID of the telecommand or telemetry request. The first seven bits contain the ID. The most significant bit is the last one and it determines whether it is a telecommand or telemetry request.

Table 3 – Content of first byte of message.

Bit(s)	Meaning
7	0 = telecommand, 1 = telemetry request
0:6	Telecommand or telemetry frame ID

When considering the full byte identifier, telecommands' first byte will be in the range of 0 to 127 and telemetry requests in the range of 128 to 255.

5.3 UART

The standard specifications of the CubeADCS UART interfaces are given in Table 4. These specifications are, however, customisable if CubeComputer is the main OBC of the satellite.

Table 4 – Standard UART specifications for CubeComputer.

Parameter	Value
Baud rate	115200
Data bits	8
Parity	None
Stop bits	1

The CubeADCS uses a simple protocol which is a variation of the KISS protocol. The UART protocol makes use of start-of-message (SOM) and end-of-message (EOM) identifiers as shown in **Table 5**.

Table 5 – UART protocol message identifiers.

Escape character	0x1F
SOM identifier	0x7F
EOM identifier	0xFF

A message will therefore begin with the sequence 0x1F, 0x7F and end with the sequence 0x1F, 0xFF. Whenever data occurs in the message, where the data byte matches the escape character (0x1F), this will be replaced with the sequence 0x1F, 0x1F.

When decoding a CubeADCS UART message, on reception of the escape character, the byte following the escape character has the implications displayed in **Table 6**.

Table 6 – UART message decoding.

Byte received after escape character	Meaning
0x7F	Start of message
0xFF	End of message
0x1F	Data byte: 0x1F
other	Should not occur – error

When formatting a message to be sent to the CubeADCS UART, the same protocol applies. The ACP will set internal error flags to indicate that a protocol error occurred or an incomplete message was received (if a SOM identifier occurred without a preceding EOM identifier). These flags can be read via the *Communication Status* telemetry request. Once set, they will remain set until the *Communication Status* telemetry is requested.

5.3.1 Requesting telemetry

The form of a telemetry request to the CubeADCS (via the UART) is displayed in **Figure 13**.

0x1F	0x7F	TLM frame ID	0x1F	0xFF
Start-of-message identifier			End-of-message identifier	

Figure 13 – UART telemetry request.

The reply from the CubeADCS will then have the form displayed in **Figure 14**.

0x1F	0x7F	TLM frame ID	TLM byte 0	...	0x1F	0xFF
Start-of-message identifier					End-of-message identifier	

Figure 14 – UART telemetry reply.

5.3.2 Sending telecommands

Figure 15 displays the form of a telecommand to the CubeADCS (via the UART):

0x1F	0x7F	TC ID	TC data byte 0	...	0x1F	0xFF
Start-of-message identifier					End-of-message identifier	

Figure 15 – UART telecommand.

The CubeADCS will reply to the telecommand with an acknowledge message, as displayed in **Figure 16**.

0x1F	0x7F	TC Error flag	0x1F	0xFF
Start-of-message identifier		0 = no error 1 = invalid TC ID 2 = invalid parameters	End-of-message identifier	

Figure 16 – UART telecommand acknowledge.

The reply will contain a single data byte with the *TC Error* flag. This is the same flag that can be read via the *Telecommand Acknowledge* telemetry request. The receipt of the telecommand-acknowledge will indicate that the CubeADCS is ready to receive another telecommand. Sending another telecommand before the acknowledge will corrupt the telecommand buffer.

5.4 I²C

If CubeComputer is not used as the main OBC of the satellite (therefore it is only the ADCS OBC), it acts as a slave on the system I²C bus. The Computer has a default 7-bit I²C slave address of 0x57. To avoid confusion caused by different I²C addressing standards, an example of the bytes that will be written on the I²C bus during normal communications is displayed in Table 7 – CubeComputer (slave node) I²C addresses.

Table 7 – CubeComputer (slave node) I²C addresses.

	8-bit Hex byte	7-bit Hex address	Binary
I²C write	0xAE	0x57	0b0010 010**0**
I²C read	0xAF	0x57	0b0010 010**1**

More details regarding the communications interface (including protocols for TCs, TLM requests, and file downloading) can be found in the CubeADCS User Manual.

5.4.1 Requesting telemetry

Telemetry is requested from the ADCS over the I²C bus by performing either a combined write-read operation (repeated start condition) or a separate master write to select the requested telemetry, followed by a master read operation. These two methods are described below:

- The first write following the start condition is the write-address of the node (0xAE).
- This is followed by the telemetry frame identifier which is the ID of the TLM that should be read.
- In the case of a separate write then read, a stop condition will follow and the master will then issue another start condition.
- In the case of a combined write-then-read operation, the master will issue a repeated start condition (without a preceding stop condition).
- The read-address is then written by the master (0xAF).
- The master then issues a number of read cycles depending on the length of the telemetry frame.

Figure 17 – I²C Telemetry request. the form of a telemetry request.

S	0xAE	TLM frame ID	S	0xAF	TLM byte 0	…	P
	Master writes node address (write operation)	Master writes telemetry frame ID		Master writes node address (read operation)	Master reads first TLM data byte		

Figure 17 – I²C Telemetry request.

The length and content of the telemetry frames are shown in the ***CubeADCS Reference Manual*** and a summarized list of TLMs can be found in the ***CubeADCS ICD***.

As the master determines the number of bytes that are read, it is possible to read past the end of a telemetry frame or to read an incomplete telemetry frame. The ADCS will flag an error if an incorrect number of bytes are read for a given frame identifier. This flag is stored in the *Communications Status* telemetry frame (see ICD), and can be read using a telemetry request. The flag will remain set until the communication status telemetry frame is read.

5.4.2 Sending telecommands

Telecommands are given to the CubeADCS by performing a master write to the module, as found below:

- The first write following the start condition is the write-address of the node (0xAE).
- The first data byte (after the address byte) is the telecommand identifier
- This is followed by the telecommand parameters.

Figure 18 – I^{2}C Telecommand. the form of a telecommand.

S	0xAE	TC ID	TC data 0	...	P
	Master writes node address (write operation)	Master writes telecommand ID	Master writes data bytes		

Figure 18 – I^{2}C Telecommand.

The number and format of these parameters vary for each telecommand and are shown in the ***CubeADCS Reference Manual*** and a summarized list of TCs can be found in the ***CubeADCS ICD***.

The telecommand acknowledge status can be polled via a telemetry request (see ICD) to ensure that the CubeADCS successfully registered the telecommand provided to it.

Table 8 – Telecommand acknowledge telemetry frame.

Telemetry frame ID	240				
Name	Telecommand Acknowledge				
Frame length (bytes)	4				
Channels	**Offset**	**Length**	**Channel**	**Data type**	**Detail**
	0	8	Last TC ID	UINT	ID of last received TC
	8	1	Processed flag	BOOL	Flag to indicate if the last TC has been processed. Sending another TC while this flag is 0 may corrupt the TC buffer
	16	8	TC error status	ENUM	List of possible errors that could occur when processing a telecommand: 0 = No error 1 = Invalid TC 2 = Incorrect TC parameter length 3 = Incorrect TC parameter value

It is not a requirement that the telecommand acknowledge status has to be read following a telecommand, but an error will occur if another telecommand is sent before the *Telecommand Processed* flag (contained in the *Telecommand Acknowledge* telemetry frame) has been set. In this case the telecommand buffer will be overwritten, while the first telecommand is being processed, leading to corrupt telecommand data.

The *Processed* flag is not an indication of the telecommand execution status. It is only an indication that the module is ready for another telecommand to be sent.

The *Telecommand Acknowledge* telemetry frame also contains a *TC Error* flag. This flag will be set if an invalid telecommand ID was received for the last telecommand, if the number of data bytes were incorrect or if it contained invalid data.

The following sequence illustrates the actions that the master has to take to ensure proper telecommand execution:

- Send telecommand.
- Poll *Telecommand Acknowledge* telemetry until the *Processed*-flag equals 1. *Include a delay of at least 1ms to not overload the system with communications*

- Confirm telecommand validity by checking the *TC Error* flag of the last read *Telecommand Acknowledge* telemetry.
- Back to step 1 (if another telecommand is to be sent).

5.5 CAN

The CubeADCS unit can act as a slave or a master on a 1 Mbps CAN V2.0B bus.

5.5.1 CAN Protocol

The CubeSpace CAN protocol uses the extended frame format, consisting of 29 identifier bits. These identifier bits contain the information depicted in the Table 9.

Table 9 – CAN identifier frame format.

MSB			LSB
Message Type	**Channel/ID**	**Source Address**	**Destination Address**
5 Bits	8 Bits	8 Bits	8 Bits

The CAN Controller is set up to filter and mask based on the destination address bits of the identifier. If the `Destination address'-field of a received message matches the CubeADCS CAN address, the message is considered valid.

5.5.2 Standard Transactions

Table 10 displays the identifiers for standard CAN messages.

Table 10 – CAN message types.

Message Type	Identifier	Description
TC Request	0x01	Command/Request
TC Response	0x02	Acknowledge
TC Not Ack	0x03	Command Failure
TLM Request	0x04	Request
TLM Response	0x05	Response
TLM Not Ack	0x06	Request Failure
Extended TC Request	0x07	Multiple CAN Packet Command/Request
Extended TLM Response	0x08	

An example of a standard CAN message is shown in Table 11. In this example an Identification TLM packet (0x80) request is sent from a node with address 0x01 to a node with address 0x02.

Table 11 – Example CAN Identifier TLM Request.

MSB			LSB
Message Type	**Channel/ID**	**Source Address**	**Destination Address**
0x04	0x80	0x01	0x02

5.5.3 Extended Transactions

If the telecommand or the telemetry request response contains more than the maximum of eight data bytes, an Extended Telemetry Request or Extended Telecommand Response is used. The Message type for such TC or TLMs are defined in Table 10.

For these extended transactions, a protocol is implemented on the data bytes of the CAN packets. The 8th byte of the data of the extended transaction contains the number of CAN packets left which is part of the complete extended telemetry or telecommand request or response. Each packet will therefore only have 7 data bytes. Considering that the extended TLM or TC can consist of 256 (1byte) such packets, the total length of an extended TC or TLM is 1792 bytes. If a TC or TLM is longer than 1792 bytes, the file download mechanism must be used.

Table 12 – CAN Extended TC/TLM Data.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Packets Left

The last byte of the last packet must be zero indicating that there are no further CAN packets part of this transaction. A standard telecommand response, similar to the standard operations, is sent from the salve if the entire extended transaction has been received successfully.

5.6 Telecommands and telemetries

5.6.1 Telecommands

Table 13 lists the telecommands for commanding a CubeADCS unit. Note that the list of TCs is compatible with any CubeADCS bundle (Magnetic, Y-Momentum, or 3-Axis), and several TCs are therefore not relevant for the CubeADCS unit. The identification numbers and structures of the TCs are available once the user has purchased a CubeADCS unit.

Table 13 - List of telecommands.

ID	Name	Description	Length (bytes)
		General	
1	Reset	Perform a reset	1
2	Current Unix Time	Current Unix Time	6
3	Cache enabled state	Cache enabled state	1
4	Reset Log Pointer	Reset pointer to log buffer (from where LastLogEvent TLM is returned)	0
5	Advance Log Pointer	Advance pointer to log buffer (from where LastLogEvent TLM is returned)	0
6	Reset Boot Registers	Reset Boot Registers	0
7	Deploy Magnetometer Boom	Deploy magnetometer boom	1
10	ADCS Run Mode	Set ADCS enabled state & control loop behavior	1
11	ADCS Power Control	Control power to selected components	3
12	Clear Errors	Clear Latched Error Flags	1
13	Set Attitude Control Mode	Set attitude control mode	4
14	Set Attitude Estimation Mode	Set attitude estimation mode	1
15	Set Commanded Attitude Angles	Set commanded attitude angles	6
16	Set Magnetorquer Output	Set magnetorquer output (only valid if Control Mode is None)	6
17	Set Wheel Speed	Set wheel speed (only valid if Control Mode is None)	6
18	Trigger ADCS Loop	Trigger ADCS to perform one iteration of the control loop (only valid when ADCS Run Mode is Triggered)	0
19	Trigger ADCS Loop with Simulated Sensor Data	Trigger ADCS to perform one iteration of the control loop (only valid when ADCS Run Mode is Triggered)	90
20	ADCS Configuration	Current configuration	272
21	Set Magnetorquer Configuration	Set magnetorquer configuration parameters	3
22	Set Wheel Configuration	Set wheel configuration parameters	4

ID	Name	Description	Length (bytes)
23	Set Rate Gyro Configuration	Set rate gyro configuration parameters	3
24	Set CSS Alignment Configuration	Set photodiode pointing directions	10
25	Set CSS Scale Factor Configuration	Set photodiode scale factors	11
26	Set Sun Sensor Configuration	Set Sun sensor configuration parameters	15
27	Set Nadir Sensor Configuration	Set nadir sensor configuration parameters	15
28	Set Nadir Sensor Mask Configuration 1	Set Nadir Sensor Mask Configuration 1	8
29	Set Nadir Sensor Mask Configuration 2	Set Nadir Sensor Mask Configuration 2	8
30	Set Nadir Sensor Mask Configuration 3	Set Nadir Sensor Mask Configuration 3	8
31	Set Nadir Sensor Mask Configuration 4	Set Nadir Sensor Mask Configuration 4	8
32	Set Nadir Sensor Mask Configuration 5	Set Nadir Sensor Mask Configuration 5	8
33	Set Magnetometer Mounting Configuration	Set magnetometer mounting configuration parameters	6
34	Set Magnetometer Offset and Scaling Configuration	Set Magnetometer Offset and Scaling Configuration	12
35	Set Magnetometer Sensitivity Configuration	Set Magnetometer Sensitivity Configuration	12
36	Set Rate Sensor Configuration	Set Rate Sensor Offsets	7
37	Set Star Tracker Configuration	Set configurations of CubeStar	26
38	Set Detumbling Control Parameters	Set controller gains and reference values for Detumbling control mode	14
39	Set Y-Wheel Control Parameters	Set controller gains and reference value for Y-wheel control mode	20
40	Set Reaction Wheel Control Parameters	Set controller gains and reference value for reaction wheel control mode	8
41	Set Moments of Inertia	Set satellite moments of inertia	12
42	Set Products of Inertia	Set satellite products of inertia	12
43	Set Estimation Parameters 1	Set estimation noise covariance and sensor mask 1	16
44	Set Estimation Parameters 2	Set estimation noise covariance and sensor mask 2	14
45	Set SGP4 Orbit Parameters	Set SGP4 Orbit Parameters	64
46	Set SGP4 Orbit Inclination	Set SGP4 Orbit Inclination	8
47	Set SGP4 Orbit Eccentricity	Set SGP4 Orbit Eccentricity	8
48	Set SGP4 Orbit RAAN	Set SGP4 Orbit RAAN	8

ID	Name	Description	Length (bytes)
49	Set SGP4 Orbit Argument of Perigee	Set SGP4 Orbit Argument of Perigee	8
50	Set SGP4 Orbit B-Star Drag term	Set SGP4 Orbit B-Star Drag term	8
51	Set SGP4 Orbit Mean Motion	Set SGP4 Orbit Mean Motion	8
52	Set SGP4 Orbit Mean Anomaly	Set SGP4 Orbit Mean Anomaly	8
53	Set SGP4 Orbit Epoch	Set SGP4 Orbit Epoch	8
54	Set Tracking Controller Gain Parameters	Set controller gains for tracking control mode	12
55	Set Tracking Controller Target Reference	Set target reference for tracking control mode	12
56	Set Mode of Magnetometer Operation	Use of Main or Redundant magnetometer	1
63	Save Configuration	Save current configuration to flash memory	0
64	Save Orbit Parameters	Save current orbit parameters to flash memory	0
80	Save Image	Save and capture image from one of CubeSense cameras or CubeStar camera to SD card	2
100	Set Boot Index	Select which program to boot	1
101	Run Selected Program	Run Selected Program	0
102	Read Program Information	Read Program CRC, length	1
103	Copy Program to Internal Flash	Copy Program to Internal Flash	2
104	SD Log1 Configuration	Log selection and period for LOG1	13
105	SD Log2 Configuration	Log selection and period for LOG2	13
106	UART Log Configuration	Log selection and period for UART (unsolicited TLM)	12
108	Erase File	Erase File	3
112	Load File Download Block	Fill download buffer with file contents	8
113	Advance File List Read Pointer	Advance File List Read Pointer	0
114	Initiate File Upload	Initiate File Upload	2
115	File Upload Packet	File Upload Packet	22
116	Finalize Upload Block	Finalize Uploaded File Block	7
117	Reset Upload Block	Reset HoleMap for Upload Block	0
118	Reset File List Read Pointer	Reset File List Read Pointer	0
119	Initiate Download Burst	Initiate Download Burst	2
120	Hole Map 1	File Upload Hole Map 1	16
121	Hole Map 2	File Upload Hole Map 2	16
122	Hole Map 3	File Upload Hole Map 3	16
123	Hole Map 4	File Upload Hole Map 4	16
124	Hole Map 5	File Upload Hole Map 5	16
125	Hole Map 6	File Upload Hole Map 6	16

ID	Name	Description	Length (bytes)
126	Hole Map 7	File Upload Hole Map 7	16
127	Hole Map 8	File Upload Hole Map 8	16

5.6.2 Telemetry

Table 14 lists the telemetry frames that can be requested from a CubeADCS unit. Note that the list of TLMs is compatible with any CubeADCS bundle (Magnetic, Y-Momentum, or 3-Axis) and several TLMs are therefore not relevant for the CubeADCS unit. The identification numbers and structures of the TLM frames are available once the user has purchased a CubeADCS unit.

Table 14 - List of telemetry frames.

ID	Name	Description	Length (bytes)
General			
128	Identification	Identification information for this node	8
130	Boot Index and Status	Current selected boot index and status of last boot	2
131	Cache Enabled State	Cache enabled state	1
132	Boot And Running Program Status	Boot And Running Program Status	6
140	Current Unix Time	Current Unix Time	6
141	Last Logged Event	Last Logged Event (relative to pointer - adjusted via Advance and Reset TCs (3 & 4)	6
142	SRAM Latchup counters	SRAM Latchup counters	6
143	EDAC Error Counters	EDAC Error Counters	6
144	Communication Status	Communication status - includes command and telemetry counters and error flags	6
145	Current ADCS State	Current state of the Attitude Control Processor	6
146	Estimated Attitude Angles	Estimated attitude angles	6
147	Estimated Angular Rates	Estimated angular rates relative to orbit reference frame	6
148	Satellite Position (ECI)	Satellite position in ECI frame	6
149	Satellite Velocity (ECI)	Satellite velocity in ECI frame	6
150	Satellite Position (LLH)	Satellite position in WGS-84 coordinate frame	6
151	Magnetic Field Vector	Measured magnetic field vector	6
152	Coarse Sun Vector	Measured coarse Sun vector	6
153	Fine Sun Vector	Measured fine Sun vector	6
154	Nadir Vector	Measured nadir vector	6
155	Rate Sensor Rates	Rate sensor measurements	6
156	Wheel Speed	Wheel speed measurement	6
157	Magnetorquer Command	Magnetorquer commands	6
158	Wheel Speed Commands	Wheel speed commands	6

ID	Name	Description	Length (bytes)
159	IGRF Modelled Magnetic Field Vector	IGRF modelled magnetic field vector (orbit frame referenced)	6
160	Modelled Sun Vector	Modelled Sun vector (orbit frame referenced)	6
161	Estimated Gyro Bias	Estimated rate sensor bias	6
162	Estimation Innovation Vector	Estimation innovation vector	6
163	Quaternion Error Vector	Quaternion error vector	6
164	Quaternion Covariance	Quaternion covariance	6
165	Angular Rate Covariance	Angular rate covariance	6
166	Raw Nadir Sensor	Nadir sensor capture and detection result	6
167	Raw Sun Sensor	Sun sensor capture and detection result	6
168	Raw CSS 1 to 6	Raw CSS measurements 1 to 6	6
169	Raw CSS 7 to 10	Raw CSS measurements 7 to 10	6
170	Raw Magnetometer	Raw magnetometer measurements	6
171	CubeSense Current Measurements	CubeSense current measurements	6
172	CubeControl Current Measurements	CubeControl current measurements	6
173	Wheel Currents	XYZ Wheel current measurement	6
174	ADCS Temperatures	Magnetometer + MCU + Rate sensor temperature measurements	6
175	Rate Sensor Temperatures	Rate sensor temperatures	6
176	Raw GPS Status	Raw GPS status	6
177	Raw GPS Time	Raw GPS time	6
178	Raw GPS X	Raw GPS X position and velocity (ECI referenced)	6
179	Raw GPS Y	Raw GPS Y position and velocity (ECI referenced)	6
180	Raw GPS Z	Raw GPS Z position and velocity (ECI referenced)	6
181	Star 1 Body Vector	Star 1 Body Vector	6
182	Star 2 Body Vector	Star 2 Body Vector	6
183	Star 3 Body Vector	Star 3 Body Vector	6
184	Star 1 Orbit Vector	Star 1 Orbit Vector	6
185	Star 2 Orbit Vector	Star 2 Orbit Vector	6
186	Star 3 Orbit Vector	Star 3 Orbit Vector	6
187	Star Magnitude	Instrument magnitude of identified stars	6
188	Star Performance	Performance parameters of star measurement	6
189	Star Timing	Timing information of star measurement	6
190	ADCS State	Current ADCS state	48

ID	Name	Description	Length (bytes)
191	ADCS Measurements	Calibrated sensor measurements	72
192	Actuator Commands	Actuator commands	12
193	Estimation Data	Estimation meta-data	42
194	Raw Sensor Measurements	Raw sensor measurements	28
195	Power and Temperature Measurements	Power and temperature measurements	32
196	Adcs Execution Times	Returns information about execution times of ACP functions	8
197	ADCS Power Control	Control power to selected components	3
198	ADCS Misc Current Measurements	CubeStar and Torquer current measurements	6
199	Commanded Attitude Angles	Commanded attitude angles	6
200	Tracking Controller Target Reference	Tracking controller target reference	12
206	ADCS Configuration	Current configuration	272
207	SGP4 Orbit Parameters	SGP4 Orbit Parameters	64
210	Raw GPS Measurements	Raw GPS measurements	36
211	Raw Star Tracker	Raw Star Tracker Measurement	36
212	Star 1 Raw Data	Catalogue index and detected coordinates for star 1	6
213	Star 2 Raw Data	Catalogue index and detected coordinates for star 2	6
214	Star 3 Raw Data	Catalogue index and detected coordinates for star 3	6
215	Redundant Magnetometer Raw Measurements	Redundant magnetometer raw measurement data, temperature and magnetometer mode	9
218	Estimated Quaternion	Estimated quaternion set	6
219	ECEF Position	Satellite position in ECEF coordinates	6
220	ACP Execution State	Returns information about the ACP loop	3
230	Program Information	Program information including file size and CRC	8
231	Copy To Internal Flash Progress	Progress of copy to internal flash operation	1
233	Status of Image Capture and Save Operation	Status of Image Capture and Save Operation	2
235	SD Log1 Configuration	Log selection and period for LOG1	13
236	SD Log2 Configuration	Log selection and period for LOG2	13
237	UART Log Configuration	Log selection and period for UART (unsolicited TLM)	12

ID	Name	Description	Length (bytes)
240	Telecommand Acknowledge Telemetry Format	Telemetery frame with acknowledge status of the previously sent command	4
241	File Download Buffer with File Contents	File download buffer 20-byte packet	23
242	Download Block Ready	Status about download block preparation	5
243	File Information	File Information	12
244	Initialize Upload Complete	Initialize Upload Complete	1
245	Upload Block Complete	Finalize Upload Block Complete	1
246	Block Checksum	File upload Block CRC16 Checksum	2
247	Hole Map 1	File Upload Hole Map 1	16
248	Hole Map 2	File Upload Hole Map 2	16
249	Hole Map 3	File Upload Hole Map 3	16
250	Hole Map 4	File Upload Hole Map 4	16
251	Hole Map 5	File Upload Hole Map 5	16
252	Hole Map 6	File Upload Hole Map 6	16
253	Hole Map 7	File Upload Hole Map 7	16
254	Hole Map 8	File Upload Hole Map 8	16

6. Specifications

6.1 CubeADCS specifications

Table 15 displays the CubeADCS 3-Axis specifications.

Table 15 – CubeADCS 3-Axis specifications.

Specification	Value	Notes
Electrical		
3.3 V supply	3.23 V – 3.40 V	
5 V supply	4.85 V – 5.15 V	
$V_{battery}$	6.5 V – 16.0 V	
Physical		
Mass		
CubeADCS PC104 stack	282 g	See Section 4.5 for breakdown
Peripherals (Depends on configuration)	< 350 g < 630 g < 870 g	Incl. Small wheels, CubeStar Incl. Medium wheels, CubeStar Incl. Large wheels, CubeStar
Dimensions		
PC104 stack	90 x 96 x 52 mm (0.52 U)	Excl. CubeWheels
	90 x 96 x 75 mm (0.75 U)	Small CubeWheels mounted on CubeConnect, above CubeControl
	90 x 96 x 86 mm (0.86 U)	Small CubeWheels mounted on CubeConnect, below CubeComputer
Small CubeWheel	28 x 28 x 26.2 mm	
Medium CubeWheel*	46 x 46 x 31.5 mm	
Large CubeWheel*	57 x 57 x 31.5 mm	
CubeStar*	35 x 50 x 57 mm	Including lens
CSS	3.8 x 10.8 x 1.7 mm	Each photodiode PCB
Deployable magnetometer	16.8 x 83.3 x 6.7 mm	Stowed
Redundant magnetometer*	19 x 19 x 6.7 mm	
Outgassing		
Total Mass Loss (TML)	< 1.0 %	
Collected Volatile Condensable Material (CVCM)	< 0.1 %	
Vibration		
Sine sweep	5 - 100 Hz @ 2.5 g 100 - 125 Hz @ 1.25 g	Sweep rate: 2 oct/min Direction: X, Y, Z
Random vibration	20 Hz @ 0.01125 g^2/Hz 130 Hz @ 0.05625 g^2/Hz 800 Hz @ 0.05625 g^2/Hz 2000 Hz @ 0.015 g^2/Hz	Duration: 120 s RMS acceleration: 8.03 g Direction: X, Y, Z
Thermal		
Operating temperature range	-10°C to +60°C	

Storage temperature range	-10°C to +60°C	

* Optional extra on CubeADCS 3-Axis

6.2 Sensor specifications

Table 16 displays the sensor specifications of the various CubeADCS 3-Axis components.

Table 16 – Sensor specifications.

Specification	Value	Notes
Sun and nadir sensor		
Sun sensor measurement accuracy	< 0.2°	Entire FOV, 3-σ
Nadir sensor measurement accuracy	< 0.2°	Entire Earth visible in FOV, 1-σ
Field of view (both cameras)	150° – 200°	
Update rate	1 Hz	
Image size	1 MB	1024 x 1024 pixels
Image format	8-bit grayscale	
Coarse Sun sensors		
Measurement accuracy	< 10°	1σ
Update rate	1 Hz	
Magnetometers		
Measurement noise	< 50 nT	1σ (per X/Y/Z channel)
Update rate	1 Hz	Averaged output
Rate sensors		
Measurement noise	< 0.015 °/s	1σ
Update rate	1 Hz	Averaged output
Angular Random Walk	$0.28°/\sqrt{h}$	
Bias instability	24°/h	
CubeStar		
Mass	55g	Without mounting screws
Dimensions	50 x 35 x 55mm	The length of about 55mm can change by ±2mm. (Without baffle)
Operating Voltage	3.3V	
In-rush current peak	156 mA	Maximum of the three peaks at power-on.
In-rush current time	1 ms	Maximum time of any one of the three peaks
Data Interface	I^2C and UART	
Field of View	58° x 47°	Horizontal (X-axis) X Vertical (Y-axis) (Without baffle)
Star Catalogue Size	410	
Sensitivity Range	< 3.8	Star Magnitude
Sky cover:	99.71%	
Accuracy	0.01° - cross bore	
	0.03° - roll	

Update Rate	1 Hz	
Max Tracking Rate	0.3°/s	
Max Acquisition Time	3 s	
Image Size	1.3 MB	1280x1030

6.3 Actuator specifications

Table 17 displays the actuator specifications of the various CubeADCS 3-Axis components.

Table 17 – Actuator specifications.

Specification	Value	Notes
Magnetorquers		
Maximum magnetic dipole (rods)	0.2 Am^2	Maximum on-time of 800 ms
Maximum magnetic dipole (coil)	0.12 Am^2	Maximum on-time of 800 ms
On-time command resolution	10 ms	For a 1 Hz control period
Small CubeWheel		
Maximum momentum storage	1.7 mNms	
Maximum wheel speed	± 8000 rpm	
Maximum torque	0.23 mNm	
Medium CubeWheel		
Maximum momentum storage	10.0 mNms	
Maximum wheel speed	± 6000 rpm	
Maximum torque	1.0 mNm	
Large CubeWheel		
Maximum momentum storage	30.0 mNms	
Maximum wheel speed	± 6000 rpm	
Maximum torque	2.3 mNm	

6.4 Power consumption

Table 18 displays the power consumption of the various CubeADCS 3-Axis components.

Table 18 – Power consumption.

Component	Average current (mA)			Total average power (mW)
	3.3 V	5 V	$V_{battery}$*	
CubeComputer	60	-	-	198
CubeSense	10	14	-	103
CubeControl (torquers off)	75	-	-	248
CubeControl (torquers full on)	75	230	-	1398
CubeWheel				
Small, 2000 rpm	21	-	7	< 126
Small, maximum torque	21	-	80	710
Medium, 2000 rpm	21	-	7	< 126
Medium, maximum torque	21	-	TBD	TBD

Large, 2000 rpm	21	-	24	< 254
Large, maximum torque	21	-	TBD	TBD

* $V_{battery}$ = 8.0 V

7. Handling

⚠ Anti-static

The bundle contains a variety of static sensitive devices. Therefore, the appropriate electrostatic discharge (ESD) protection measures must be implemented. The unit must never be handled without proper grounding.

⚠ Cleanliness

It is recommended that the CubeADCS unit be handled in a clean environment. Therefore, an appropriate laminar flow workbench or a clean room of ISO class 8 or better, should be used.

⚠ Moisture

The unit should be kept free of moisture or liquids, as these could have corrosive effects on the electronics and electronic joints which may lead to degradation and loss of reliability of the circuits.

⚠ Shock

The unit must be handled with care. Dropping or bumping the unit should be avoided completely.

⚠ Camera lens cleanliness

The camera lenses should be kept clean and free of any dirt that may obstruct the images captured by the camera. Dust should be removed with a microfiber cloth. If required, the lens may be cleaned using ethanol and appropriate lens cleaning equipment. Unnecessary cleaning of the lens should be avoided.

⚠ Camera lens structural integrity

The camera sensors are aligned to be parallel with the CubeSense PCB. This is important, as misalignment of the cameras influence the performance of the system. Therefore, external forces on the camera modules should be avoided entirely.

⚠ Camera lens covers

The Sun and nadir optics are fitted with dust caps which should be removed before flight.

⚠ Reaction wheels

The aluminum housings of the reaction wheels should NOT be tampered with. Tampering with the housings may damage the wheels. No attempt should be made to loosen or remove the fasteners that secure the housings.