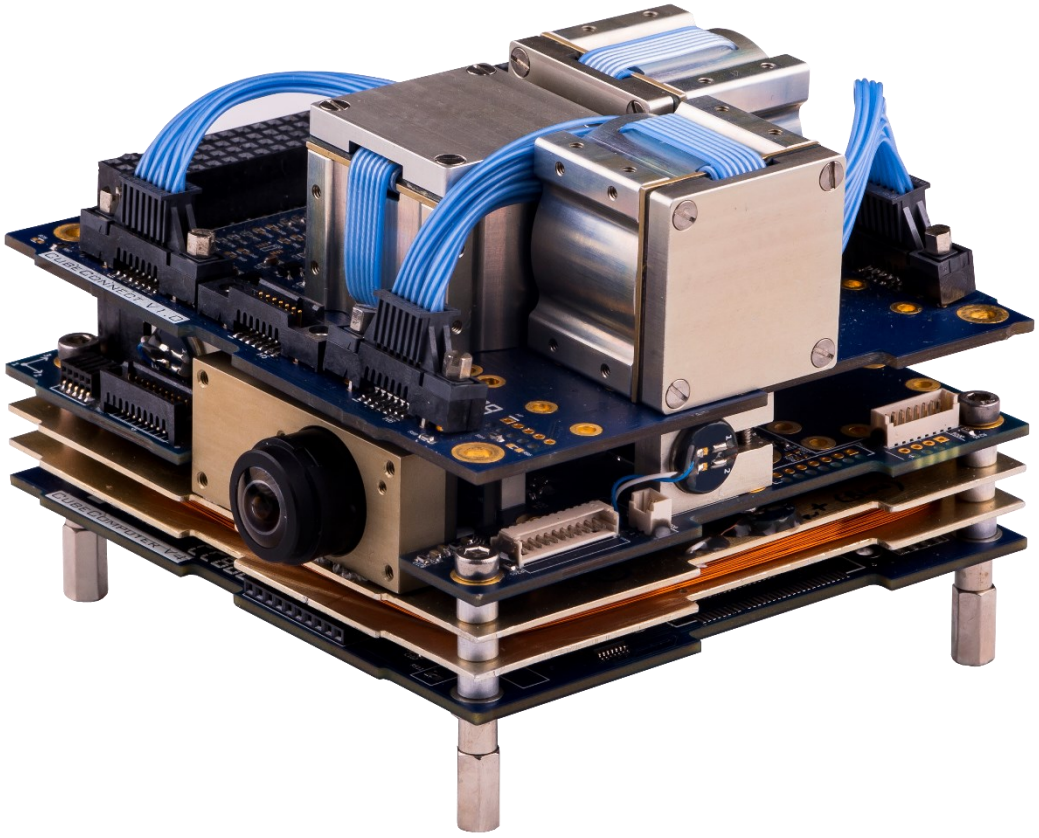
CubeADCS

The complete ADCS solution



Health Check

Contact Us

Phone (0027) 79 945 9957
E-mail info@cubespace.co.za
Web www.cubespace.co.za
Facebook /CubeSpaceADCS
Twitter @CubeSpace_ADCS

Physical Address

CubeSpace
Hammanshand Road
Stellenbosch
7600
South Africa

List of Acronyms/Abbreviations

ACP	ADCS Control Program
ADCS	Attitude Determination and Control System
CSS	Coarse Sun Sensor
DN	Digital Number (non-calibrated value)
ICD	Interface Control Document
MCU	Microcontroller Unit
PCB	Printed Circuit Board
TC	Telecommand
TLM	Telemetry
UART	Universal Asynchronous Receiver/Transmitter

List of Symbols

T_{amb}	Ambient/Room Temperature
I_{SRAM}	SRAM Current
$\omega_{x/y/z}$	The angular rate of rotation about a particular axis

Table of Contents

1. Health Check Details

CubeADCS unit number: CA2217

Date of CubeSpace Health Check: 2022/07/29

Name(s) of CubeSpace engineer(s): Marisa du Plessis

Signature(s) of CubeSpace engineer(s): 

Name of CubeSpace Manager: Justin Miller

Signature of CubeSpace Manager: 

Date of client Health Check:

Name(s) of client engineer(s):

Signature(s) of client engineer(s):

2. Component Summary

The table below contains the applicable information for all components present in this CubeADCS assembly.

Component	Serial Number	Applicable Version/Size
CubeComputer	SN2235	V4.6
CubeControl	CL2217	V2.5
CubeTorquer 1	CO2032	CubeCoil Single
CubeTorquer 2	CR2322	CubeRod Small
CubeTorquer 3	CR2323	CubeRod Small
CubeMag	MM2233	V3.0 – Deployable
CubeMag Redundant	N/A	N/A
Coarse Sun Sensors	-	Self-Assembly
Y-Momentum Wheel	N/A	N/A
CubeConnect	CN2229	V2.4 – Standard
CubeSense 1	CS22041	V3.1 – Sun
CubeSense 2	CS22043	V3.1 – Sun
CubeWheel 1	CW2532	V1.3.2 – Small
CubeWheel 2	CW2536	V1.3.2 – Small
CubeWheel 3	CW2535	V1.3.2 – Small
CubeWheel 4	N/A	N/A
CubeStar	N/A	N/A

3. Introduction

This document will provide the instructions and results of the health check of the **CubeADCS** unit. The instructions provided must be followed meticulously and the observed results must be indicated as is.

3.1 Handling

The CubeADCS unit contains electrostatic sensitive components. Under no circumstances should the unit be handled without anti-static protection.

The CubeADCS unit is a delicate mechanical assembly. Always handle with great care, preferably using clean anti-static gloves.

When handling the CubeADCS unit, always place the unit on an anti-static mat in a clean environment, as required for flight-model components.

3.2 Test requirements

The following items are required when performing the health check:

- Power supply (V_{max} > 10 V, I_{max} > 1 A).
- An appropriate power distribution platform with 5 V and 3.3 V regulators.
- Latest **CubeSupport** software (provided on USB with **CubeADCS**).
- The supplied UART-to-USB cable.
- The supplied **CubeADCS** unit and all peripherals.
- Any means of measuring the direction of a magnetic field, for example a normal field compass.
- A small bright light source to stimulate the Coarse Sun Sensors and Sun cameras (if any). Typically, a cell phone with a single LED flashlight can be used to create a narrow source.
- A broad bright light source to stimulate the nadir cameras (if any). Typically, a desk lamp covered with a cleanroom lint-free cloth will suffice (see Figure 1).

Figure 1 – Example of modified desk lamp

3.3 Preparation

Follow the instructions below to prepare the **CubeADCS** for the health check:

- Connect all peripherals to the **CubeADCS**. This normally includes:
  - Magnetometer
  - Coarse Sun Sensors
  - **CubeSense**/s (typically if 3-Axis and/or Y-Momentum capable)
  - Redundant Magnetometer (optional)
  - GPS (optional)
  - Multiple reaction wheels (typically if 3-Axis capable)
  - **CubeStar** (optional if 3-Axis capable)
- The **CubeADCS** is tested by **CubeSpace** at a standard battery voltage of **8 V**. To obtain similar health check results for comparison, it is recommended that the **CubeADCS** is supplied with the same battery voltage during customer health checks.
- Set the power supply to **8 V** and the current limit to I_{max} = **1 A** and connect the supply to the power distribution platform.
- Power down the power supply while it is being connected to the **CubeADCS**
- Connect power to the appropriate PC104 pins on the **CubeADCS** unit. Refer to the *CubeADCS Interface Control Document (ICD)* and the *Hardware Configuration Sheet* for the relevant PC104 pin locations.
- Connect the UART-to-USB cable to the UART connector on the **CubeComputer**, as shown in red in Figure 2. **NB: The black wire (Ground) should be connected to the pin closest to the corner of the PCB farthest away from the PC104 connector (right-hand pin of UART connector in Figure 2).**

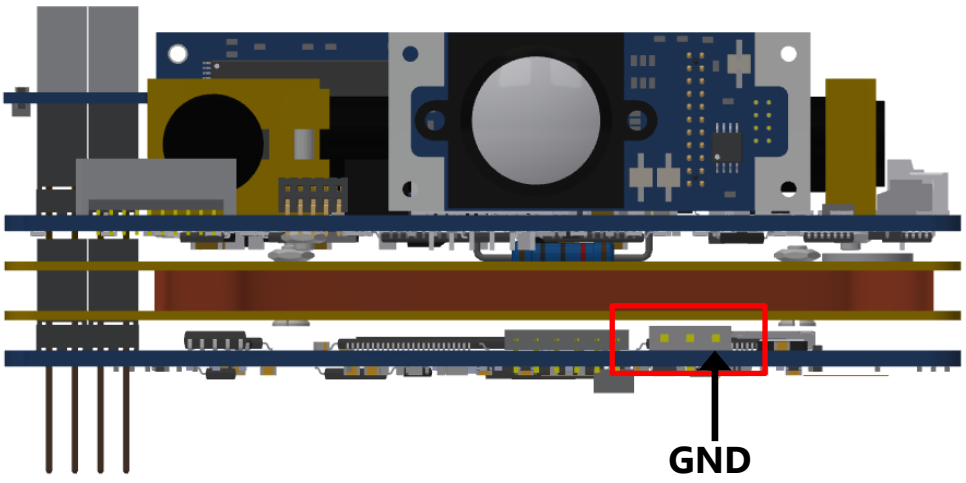


Figure 2 – Location of UART connector

4. CubeComputer

Once the preparation in the preceding section has been completed, the **CubeADCS** should be ready for the health check to start. Before powering on the **CubeADCS** and starting the test procedures in this document, please ensure the following:

- **CubeADCS** is connected to a power distribution platform with a 3.3 V and 5 V regulators and that the battery voltage line is connected to the correct PC104 pin on the **CubeADCS**.
- Power supply is set to 8 V.
- Power supply current limit is set to 1 A.
- Power supply is off.
- Cables/leads supplying power to **CubeADCS** are short and thick to keep series resistance as small as possible.
- **CubeADCS** is placed in a secure clean grounded area with enough room for users to move a light source around the ADCS, and the peripherals are placed for easy access and control.
- **CubeADCS** is connected to PC though UART-to-USB cable.
- Do not power on the **CubeADCS** until explicitly told to do so.

4.1 The Bootloader

The CubeSpace bootloader will always run first on startup. If the bootloader does not receive communications within **5 seconds**, it will automatically load the ACP program located in internal flash memory. Communicating with the bootloader within the first 5 seconds will cause the bootloader to keep running until the user explicitly instructs it to run the ACP program.

As a first step, connect to the bootloader and specify that the ACP should be loaded using the following steps:

- Before powering on the ADCS, get the **CubeSupport** application ready on the PC. The **CubeSupport** application is supplied on the USB along with the **CubeADCS** unit. Launch the **CubeSupport** application .exe which can be found in the **CubeSupport** folder on the CubeSpace USB. The **CubeSupport** application will open a window as shown in Figure 3.

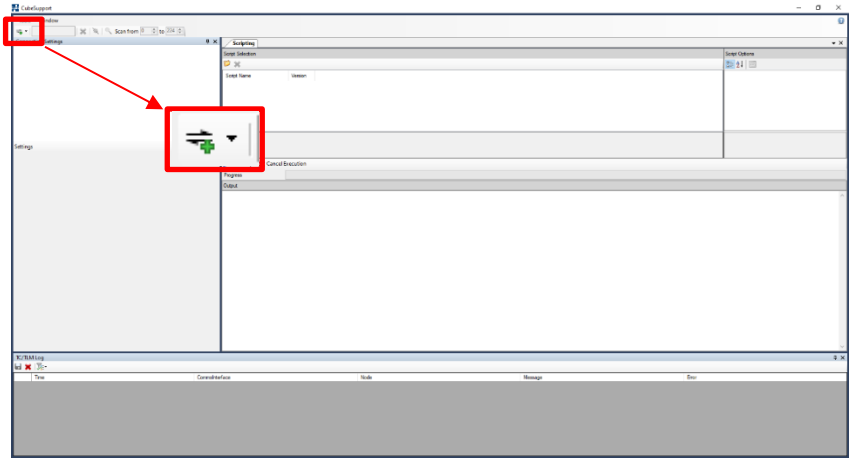


Figure 3 – Add connection to CubeSupport

- Click on the add connection icon  in **CubeSupport** as indicated in Figure 3. This will detect the UART-to-USB cable and list the connection.

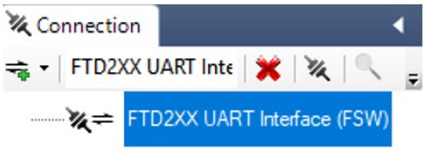


Figure 4 – New connection in list

- **Now power on the CubeADCS and wait one second for everything to power on.** Connect to the ADCS with **CubeSupport** within the first five seconds of power on by clicking on the connect button  . **CubeSupport** should connect to the bootloader and the bootloader menu should be displayed as shown in Figure 5.

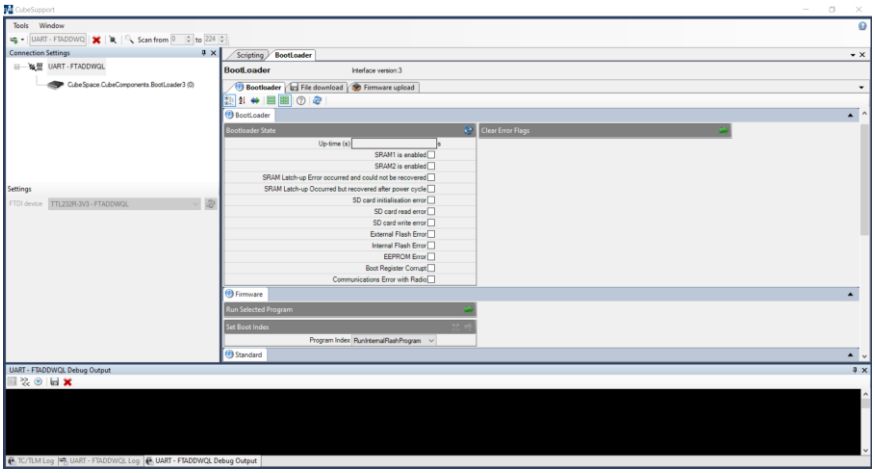


Figure 5 – CubeSupport bootloader interface

- On the ***Bootloader*** tab there are various telemetry windows which each display the bootloader information. Click on the refresh icon  to read the data from **CubeComputer**. Verify that the information is correct by comparing the fields with the corresponding data in the table below:

Test / Task	Expected Result	CubeSpace Result	Client Result
Bootloader → Bootloader State			
Up-time (s)	Incrementing every second	✓	
SRAM 1 is enabled	Checked	✓	
SRAM 2 is enabled	Checked	✓	
SRAM Latch-up Error occurred and could not be recovered	Unchecked	✓	
SRAM Latch-up Occurred but recovered after power cycle	Unchecked	✓	
SD card Error	Unchecked	✓	
External Flash Error	Unchecked	✓	
Internal Flash Error	Unchecked	✓	
EEPROM Error	Unchecked	✓	
Boot Register Corrupt	Unchecked	✓	
Communications Error with Radio	Unchecked	✓	
Standard → Boot and Running Program Status			
Cause of MCU Reset	PowerOnReset / SystemReqReset	✓	
Boot Cause	Unexpected	✓	
Boot Counter	Incrementing at every reset	✓	
Boot Program Index	RunBootloader	✓	
Firmware version (Major)	3	✓	
Firmware version (Minor)	1	✓	
Standard → Boot Index and Status			
Program Index	RunInternalFlashProgram	✓	
Boot Status	BootSuccess/BootNew	✓	
Standard – Satellite State → Communication Status			
Telecommand counter	0	✓	

Telemetry request counter	Incrementing after every refresh	✓	
All remaining communication flags	Unchecked	✓	

- In the Firmware section under the Set Boot Index sub-section, set the program index to *RunInternalFlashProgram* by clicking the option from the drop-down list as shown in Figure 6 where the header will change colour to orange.

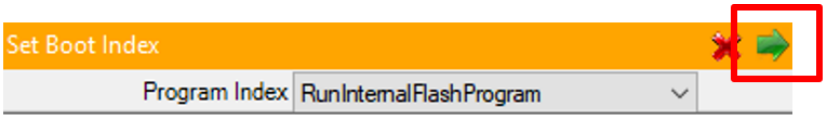


Figure 6 – Set boot index

- Transmit the command by clicking on the green arrow after setting the boot index.
- The ADCS is delivered with the ACP pre-loaded into the internal flash and setting the boot index to the internal flash will ensure that the application is executed.
- Power off the **CubeADCS** and disconnect the **CubeSupport** by clicking on the disconnect icon .
- Turn on the **CubeADCS** and wait for six seconds to pass before attempting to connect with **CubeSupport**. The bootloader will execute the ACP after five seconds. The **CubeSupport** application can connect to the ACP application and the **CubeSupport** application will display the window shown in Figure 7.

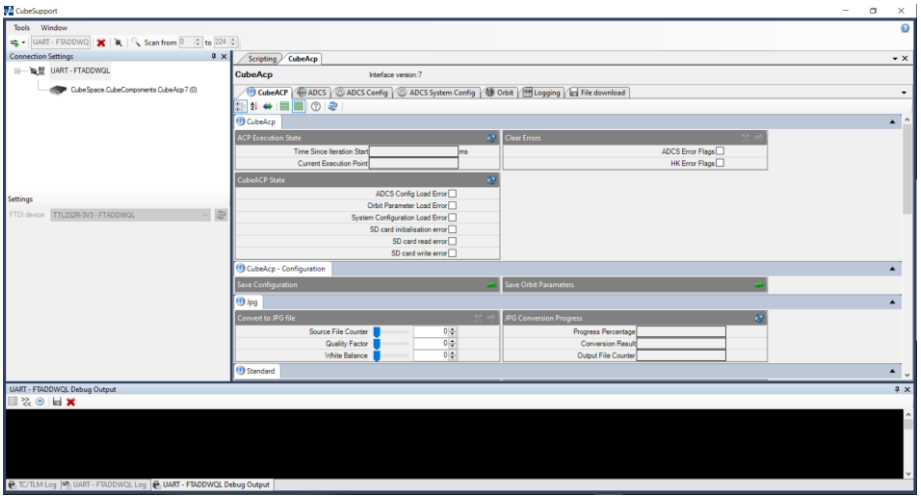


Figure 7 – CubeSupport connected to ACP

- Alternative to turning the **CubeADCS** off and on, the *Program Index* can be set, and the *Run Selected Program* command can be transmitted whilst in the bootloader interface. This will cause the bootloader to boot the application.

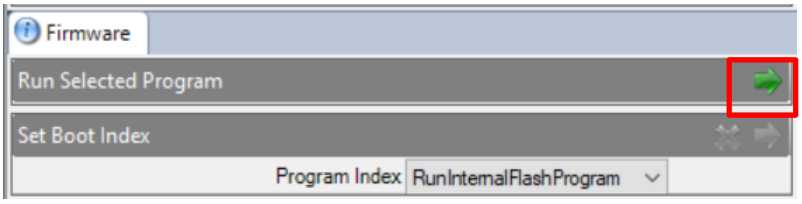


Figure 8 – Run selected program.

- An error message will be displayed reporting that no response was obtained since the bootloader started to boot the ACP.

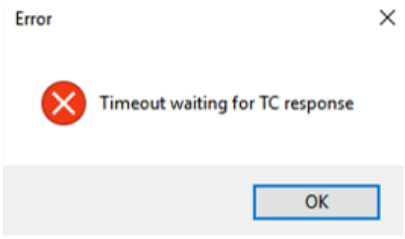


Figure 9 – Timeout Error

- Click "OK", disconnect  and reconnect  to connect to the ACP and show the ACP interface.

4.2 The CubeACP

Verify that **CubeSupport** connects to the ACP, then perform the tests that follow:

Test / Task	Expected Result	CubeSpace Result	Client Result
CubeComputer connects to the **CubeSupport** application.	✓	✓	

- Navigate to the ***CubeACP*** tab.
- Verify the following under the ***CubeACP*** tab by refreshing the data in the relevant box several times (click on the  button to refresh the data):

Test / Task	Expected Result	CubeSpace Result	Client Result
CubeACP → ACP Execution State			

Time Since Iteration Start	0-1000	✓	
Current Execution Point	Idle	✓	
Standard → Boot and Running Program Status			
Cause of MCU Reset	PowerOnReset / SystemReqReset / Unknown	✓	
Boot Cause	Unexpected	✓	
Boot Counter	Increment after every reset	✓	
Boot Program Index	RunInternalFlashProgram	✓	
Firmware Version (Major)	7	✓	
Firmware Version (Minor) *	Between 1 and 255	✓	
Standard → Identification			
Node type	10	✓	
Interface Version	7	✓	
Runtime (seconds)	Incrementing every second	✓	
Runtime (milliseconds)	0 – 1000	✓	
Standard → Boot Index and Status			
Program Index	RunInternalFlashProgram	✓	
Boot Status	BootSuccess	✓	
Standard - Satellite State → Communication Status			
Telecommand counter	0	✓	
Telemetry request counter	Incrementing	✓	
All remaining communication flags	Unchecked	✓	
Standard - Satellite State → Current Unix Time			
Time	Incrementing	✓	
Standard - Satellite State → SRAM Latchup Counters			
SRAM1 Latchups	0	✓	
SRAM2 Latchups	0	✓	
Standard - Satellite State → EDAC Error Counters			
Single SRAM upsets	0	✓	
Double SRAM upsets	0	✓	
Multiple SRAM upsets	0	✓	

* The minor version will depend on the software revision, and it does not impact the ACP interface. At the time of writing, the latest minor version is 1.

- Navigate to the ***Scripting*** tab
- Open the script named "*CubeADCS_StartHealthCheck.cs*" by clicking on the Open button ( ). This script should be supplied by **CubeSpace** on the accompanying USB drive.
- Go to the Script Options (see Figure 10), and select the appropriate options as follows:

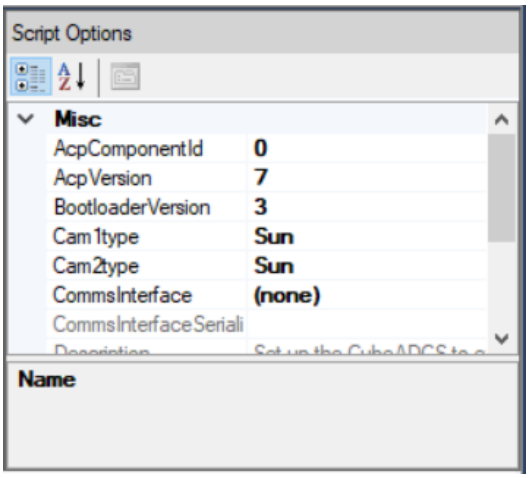


Figure 10 – Script Options

  - Ensure "AcpVersion" is the value shown in the table above at *Standard→Identification→ Interface Version*.
  - Change "Cam1type" and "Cam2type" to the options chosen in your Hardware Configuration sheet (also supplied on the USB drive).
  - At "CommsInterface" select the communication interface that you're currently using to communicate with the bundle.
  - At the option "xmlFileLocation" paste the path pointing to the AdcsConfig.*.xml file (supplied by **CubeSpace** on the USB drive) that contains your order's unique configuration. An example of such a path is "C:\CubeSpace\AdcsConfig.xml". This info can be found by right-clicking on the xml file and selecting *Properties*, as shown in Figure 11.

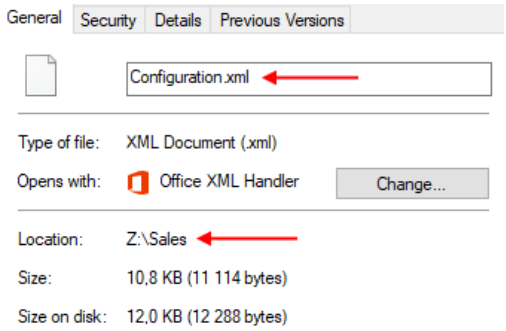
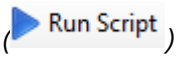


Figure 11 – Configuration File Path

- Click on *Run Script* () to execute the script. If it was successful, the *Output* should show the text "Result: Success!".
- If it failed, make sure you entered the path correctly and that the ADCS Config file was selected and not the System Config file.
- In the event the script continues to fail, please contact support@cubespace.co.za.
- Navigate to the ***ADCS*** tab.
- Verify the following under the ***ADCS*** tab by refreshing the data in the relevant box several times:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis → ADCS Power Control			
All nodes indicate PowOff	PowOff	✓	
ADCS 3-Axis - ADCS State → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsOff	✓	
All other states indicated with check boxes	Unchecked	✓	

- In the *ADCS 3-Axis → ADCS Run Mode* box, switch the run mode to *AdcsEnabled*. Note that the top of the box turns orange with a red cross and a green arrow once the relevant run mode has been selected. **Click on the green arrow to confirm and send the command to the CubeADCS unit**. Conversely, clicking on the red cross will ignore and discard the command.

Figure 12 – ADCS Run Mode

Confirm that the ADCS is enabled by reading the values indicated in the tables below (refresh the data in the relevant boxes).

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis → ADCS Execution Times			
Time to Perform ADCS update	105 ± 25 ms	97 ms ✓	
Time to Perform Sensor/Act Communication	3 ± 3 ms	0 ms ✓	
Time to Execute SGP4 Propagator	46 ± 10 ms	50 ms ✓	
Time to Execute IGRF Model	50 ± 10 ms	46 ms ✓	
ADCS 3-Axis - ADCS State → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsEnabled	✓	
Sun is Above Local Horizon*	Checked	✓	
All other Checkboxes**	Unchecked	✓	

* If ***Sun is Above Local Horizon*** is **not** checked, the **CubeADCS** expects to be in the eclipse part of its orbit according to the Unix time (or when ***Capture Status*** reads ***CaptureStartup***). This means the **CubeACP** will **not** sample the raw **CubeSense** outputs. If this happens, change the **CubeACP**'s Unix time to "move" the satellite out of eclipse. This is done at in the ***CubeACP*** tab in *Standard - Satellite State → Current Unix Time*. (Normal flight software is used for Health Check.)

**** Note that the following flags can be ignored, as they are dependent on the setup, environment, and stimulus:**

1. **Magnetometer Range Error**
2. **Cam1 Sensor Detection Error**
3. **Cam1 Sensor Range Error**
4. **Cam2 Sensor Detection Error**
5. **Cam2 Sensor Range Error**
6. **Course Sun Sensor Error**

Once all the parameters above have been verified, it can be confirmed that **CubeADCS** is operating normally. The following sections of this Health Check document will be dedicated to testing the functionality of the other ADCS nodes in the **CubeADCS** unit.

5. CubeSense

5.1 CubeSense1

Follow the instructions below to perform the **CubeSense1** health check:

- Navigate to the ***ADCS*** tab.
- Ensure sure that all nodes are selected *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box before proceeding.
- **Switch on CubeSense1** by selecting *PowOn* in the drop-down menu next to *CubeSense1 Power Selection* in the *ADCS 3-Axis → ADCS Power Control* box and transmit the command by clicking on the green arrow.
- Ensure that the **lens cap** is **on**.
- Navigate to the ***ADCS Config*** tab and go to the *ADCS 3-Axis – Configuration → CubeSense Configuration* block. Ensure that the ***Cam1 detection threshold*** is set to **150**. Next, ensure that the ***Cam1 sensor exposure time*** is set to **35** if **Cam1** is a nadir sensor, or **100** if **Cam1** is a Sun sensor. Set the ***NadirMaxBadEdges*** to **30** to make ground testing easier. Click on the green arrow to send these commands if necessary.
- Refer to *Appendix B: Typical CubeSense Images* Figure 19 to see an example of the typical image that would result in a successful detection for the Sun sensor and Figure 20 to see an example of the typical image that would result in a successful detection for the Nadir sensor.
- Navigate back to the ***ADCS*** tab and verify the following values:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS State → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsEnabled	✓	
CubeSense1 Enabled	Checked	✓	
Sun is Above Local Horizon*	Checked	✓	
All other checkboxes**	Unchecked	✓	
ADCS 3-Axis ADCS Power → CubeSense1 Current Measurements			
CubeSense1 3V3 current	19.5 ± 2.5 mA	17.3 mA ✓	
CubeSense1 Cam SRAM current	0.5 mA< I_{SRAM} <5 mA	1.2 mA ✓	
ADCS 3-Axis Raw Sensor Measurements → Raw Cam1 Sensor			
Cam1 centroid X	0	✓	
Cam1 centroid Y	0	✓	

Cam1 Capture Status*	CaptureSuccess	✓	
Cam1 Detection Result	DetectTooFewEdges or DetectSunNotFound	✓	

* If ***Sun is Above Local Horizon*** is **not** checked, the **CubeADCS** expects to be in the eclipse part of its orbit according to the Unix time (or when ***Capture Status*** reads ***CaptureStartup***). This means the **CubeACP** will **not** sample the raw **CubeSense** outputs. If this happens, change the **CubeACP**'s Unix time to "move" the satellite out of eclipse. This is done at ***CubeACP*** tab in *Standard* → *Satellite State* → *Current Unix Time.* (Normal flight software is used for Health Check.)
** Except for the aforementioned flags at the end of the **CubeComputer** section.

- Take **off** the **Cam1 camera**'s **lens cap**.
- Verify the following from the *ADCS* 3-Axis - *Raw Sensor Measurements* → *Raw Cam1 Sensor* block by testing the sensor with a light source (a dark environment will prevent false detections). If **Cam1** is a **nadir** sensor then a **large** light source should be used (e.g. a desk lamp), or if **Cam1** is a **Sun** sensor then a **small** light source should be used (e.g. narrow beam flashlight). Vary the distance between the light source and the sensor until consistent measurements are observed (normally ±150mm). If difficulties are experienced with the nadir sensor, the light source can be covered with white paper/cloth to create a more uniform light source. Finally, if no results are obtained for the nadir or Sun sensors, the exposure value can be adjusted.

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - Raw Sensor Measurements → Raw Cam1 Sensor			
Cam1 Capture Status*	CaptureSuccess	✓	
Cam1 Detection Result	DetectSuccess	✓	
Cam1 Centroid Y when placing the light close to the camera boresight	Close to zero (±100)	✓	
Cam1 Centroid Y when moving the light **up****	Increasing	✓	
Cam1 Centroid Y when moving the light **down****	Decreasing	✓	
Cam1 Centroid X when placing the light close to the camera boresight	Close to zero (±100)	✓	
Cam1 Centroid X when moving the light to your **right****	Increasing	✓	
Cam1 Centroid X when moving the light to your **left****	Decreasing	✓	

* If ***Sun is Above Local Horizon*** is **not** checked, the **CubeADCS** expects to be in the eclipse part of its orbit according to the Unix time (or when ***Capture Status*** reads ***CaptureStartup***). This means the **CubeACP** will **not** sample the raw **CubeSense** outputs. If this happens, change the **CubeACP**'s Unix time to "move" the satellite out of eclipse. This is done at ***CubeACP*** tab in *Standard* → *Satellite State* → *Current Unix Time.* (Normal flight software is used for Health Check.)
** When looking at the lens from the front with the user in the positive z-direction of the camera, as referred to in Appendix A at the end of this document.

- While keeping the light in the field of view of **CubeSense1**, navigate to *ADCS 3-Axis* → *Save Image* block and select *CamCam1* in the drop-down menu next to *Camera Select*. The *Image size* drop-down menu can be any value, but not selecting *Size0* will reduce the downloading time by lowering the image quality (*Size3* recommended). Capture the image by sending the command by clicking on the green arrow. The camera will capture an image after a delay of three seconds. Continue to hold the light in front of the camera for this duration.
- Navigate to *ADCS 3-Axis* → *Status of Image Capture and Save Operation* box. Refresh the box. The *Percentage Complete* will increase slowly, which indicates the process of the image being saved to the SD card from **CubeSense**'s memory.
- Once the *Percentage Complete* reaches 100%, navigate to the ***File download*** tab.
- Click on the refresh icon in the *Remote File System* block.

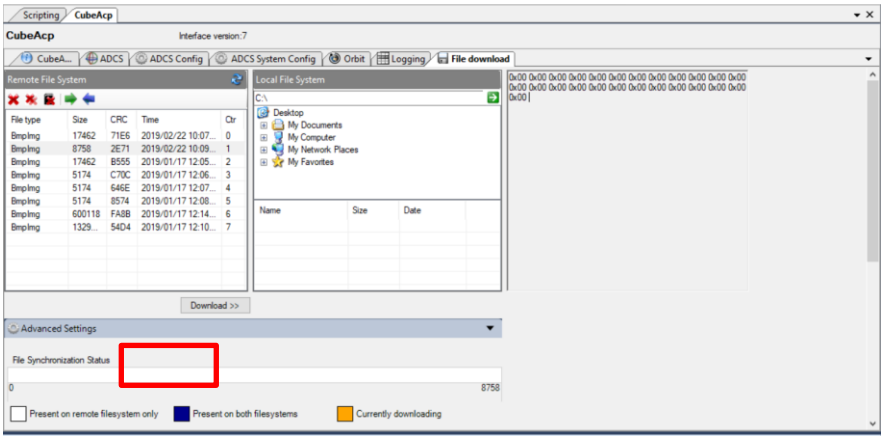


Figure 13 – *File download* tab.

- Click on the *BmpImg* file at the end of the list. Once the file is selected, the *File Synchronization Status* total size will change to the file size.
- Select a destination path in the Local File System and click the download button. A pop-up menu will appear where a location and file name can be specified, after which the download will start, and the *File Synchronization Status* load bar will progress.

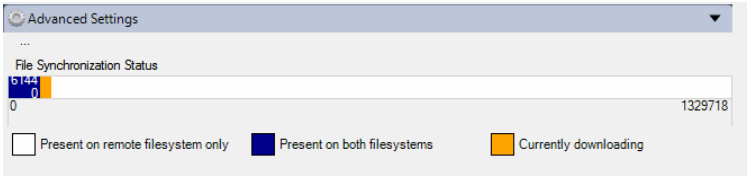


Figure 14 – Downloading a file.

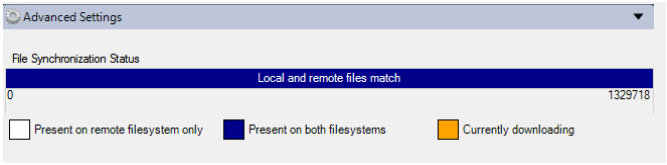


Figure 15 – File download complete.

- Verify that the light source is clearly visible in the image (large and round for a nadir sensor, or a small spot for a Sun sensor).
- Put the **Cam1** camera's **lens cap** back **on**.
- Switch **off CubeSense1** by selecting *PowOff* in the drop-down menu next to *CubeSense1 Power Selection* in the *ADCS 3-Axis → ADCS Power Control* box before proceeding.

5.2 CubeSense2

Follow the instructions below to perform the **CubeSense2** health check:

- Navigate to the ***ADCS*** tab.
- Ensure sure that all nodes are selected *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box before proceeding.
- **Switch on CubeSense2** by selecting *PowOn* in the drop-down menu next to *CubeSense2 Power Selection* in the *ADCS 3-Axis → ADCS Power Control* box and transmit the command by clicking on the green arrow.
- Ensure that the **lens cap** is **on**.
- Navigate to the ***ADCS Config*** tab and go to the *ADCS 3-Axis – Configuration → CubeSense Configuration* block. Ensure that the ***Cam2 detection threshold*** is set to **150**. Next, ensure that the ***Cam2 sensor exposure time*** is set to **35** if **Cam2** is a nadir sensor, or **100** if **Cam2** is a Sun sensor. Set the ***NadirMaxBadEdges*** to **30** to make ground testing easier. Click on the green arrow to send these commands if necessary.
- Refer to *Appendix B: Typical CubeSense Images* Figure 19 to see an example of the typical image that would result in a successful detection for the Sun sensor and Figure 20 to see an example of the typical image that would result in a successful detection for the Nadir sensor.
- Navigate back to the ***ADCS*** tab and verify the following values:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS State → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsEnabled	✓	

CubeSense2 Enabled	Checked	✓	
Sun is Above Local Horizon*	Checked	✓	
All other checkboxes**	Unchecked	✓	
ADCS 3-Axis ADCS Power → CubeSense2 Current Measurements			
CubeSense2 3V3 current	19.5 ± 2.5 mA	17.7 mA ✓	
CubeSense2 Cam SRAM current	0.5 mA< I_{SRAM} <5 mA	2.1 mA ✓	
ADCS 3-Axis Raw Sensor Measurements → Raw Cam2 Sensor			
Cam2 centroid X	0	✓	
Cam2 centroid Y	0	✓	
Cam2 Capture Status*	CaptureSuccess	✓	
Cam2 Detection Result	DetectTooFewEdges or DetectSunNotFound	✓	

* If ***Sun is Above Local Horizon*** is **not** checked, the **CubeADCS** expects to be in the eclipse part of its orbit according to the Unix time (or when ***Capture Status*** reads ***CaptureStartup***). This means the **CubeACP** will **not** sample the raw **CubeSense** outputs. If this happens, change the **CubeACP**'s Unix time to "move" the satellite out of eclipse. This is done at ***CubeACP*** tab in *Standard* → *Satellite State* → *Current Unix Time*. (Normal flight software is used for Health Check.)

** Except for the aforementioned flags at the end of the **CubeComputer** section.

- Take **off** the **Cam2 camera**'s **lens cap**.
- Verify the following from the *ADCS* 3-Axis - *Raw Sensor Measurements* → *Raw Cam2 Sensor* block by testing the sensor with a light source (a dark environment will prevent false detections). If **Cam2** is a **nadir** sensor then a **large** light source should be used (e.g. a desk lamp), or if **Cam2** is a **Sun** sensor then a **small** light source should be used (e.g. narrow beam flashlight). Vary the distance between the light source and the sensor until consistent measurements are observed (normally ±150mm). If difficulties are experienced with the nadir sensor, the light source can be covered with white paper/cloth to create a more uniform light source. Finally, if no results are obtained for the nadir or Sun sensors, the exposure value can be adjusted.

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - Raw Sensor Measurements → Raw Cam2 Sensor			
Cam2 Capture Status*	CaptureSuccess	✓	
Cam2 Detection Result	DetectSuccess	✓	
Cam2 Centroid Y when placing the light close to the camera boresight	Close to zero (±100)	✓	

Cam2 Centroid Y when moving the light **up****	Increasing	✓	
Cam2 Centroid Y when moving the light **down****	Decreasing	✓	
Cam2 Centroid X when placing the light close to the camera boresight	Close to zero (±100)	✓	
Cam2 Centroid X when moving the light to your **right****	Increasing	✓	
Cam2 Centroid X when moving the light to your **left****	Decreasing	✓	

* If ***Sun is Above Local Horizon*** is **not** checked, the **CubeADCS** expects to be in the eclipse part of its orbit according to the Unix time (or when ***Capture Status*** reads ***CaptureStartup***). This means the **CubeACP** will **not** sample the raw **CubeSense** outputs. If this happens, change the **CubeACP**'s Unix time to "move" the satellite out of eclipse. This is done at ***CubeACP*** tab in *Standard* → *Satellite State* → *Current Unix Time*. (Normal flight software is used for Health Check.)

** When looking at the lens from the front with the user in the positive z-direction of the camera, as referred to in Appendix A at the end of this document.

- Follow the previous steps shown at the end of the **CubeSense1** section again to download a **Cam2** camera image by selecting *CamCam2* in the drop-down list. Remember to keep the light source in the field of view of the camera for the three seconds of image capture.
- Verify that the light source is clearly visible in the image (large and round for a nadir sensor, or a small spot for a Sun sensor).
- Put the **Cam2** sensor's **lens cap** back **on**.
- Switch **off CubeSense2** by selecting *PowOff* in the drop-down menu next to *CubeSense2 Power Selection* in the *ADCS 3-Axis* → *ADCS Power Control* box before proceeding.

6. CubeControl

Note: CubeTorquer rod and coil current measurements are represented in terms of a digital number (DN) and not in terms of Amperes. To obtain the current value measured by either the Signal MCU or the Motor MCU in milli Amperes, multiply the DN by a scale factor of $\boldsymbol{k_{TORQ}} = 0.51$. The current measurement is not used internally by the **CubeADCS** during control of the satellite and is instead intended only as telemetry for the user.

In the case that the satellite is smaller than or equal to 3U in size and is utilising small **CubeRods** to actuate its magnetic control, no conversion factor is required (i.e. $\boldsymbol{k_{TORQ}} = 1$). The **CubeCoil** however, will still require the original conversion factor of $\boldsymbol{k_{TORQ}} = 0.51$ to obtain the current in milli Amperes – regardless of satellite size.

6.1 CubeControl Signal MCU

Follow the instructions below to perform the **CubeControl** Signal MCU health check:

- Ensure sure that all nodes are selected *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box before proceeding.
- Switch **on CubeControl Signal MCU** by selecting *PowOn* in the drop-down menu next to *CubeControl Signal Power Selection* in the *ADCS 3-Axis → ADCS Power Control* box and transmit the command by clicking on the green arrow.
- Verify the following under the ***ADCS*** tab:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS state → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsEnabled	✓	
CubeControl Signal Enabled	Checked	✓	
All other checkboxes*	Unchecked	✓	
ADCS 3-Axis - ADCS Power → CubeControl Current Measurements			
CubeControl 3V3 Current	40 ± 15 mA	33.69 mA ✓	
CubeControl 5V Current	5 ± 5 mA	4.88 mA ✓	
CubeControl Vbat Current	< 4 mA	1.46 mA ✓	
ADCS 3-Axis - ADCS Power → ADCS Temperatures			
MCU Temperature	T_{amb} ± 10°C	T_{amb} + 1.5 °C ✓	
Magnetometer Temperature	T_{amb} ± 10°C	T_{amb} + 2.2 °C ✓	

Redundant Magnetometer Temperature **	N/A	N/A	
ADCS 3-Axis - Raw Sensor Measurements → Raw CSS1 to 6			
CSS1	4 ± 4	✓	
CSS2	4 ± 4	✓	
CSS3	4 ± 4	✓	
CSS4	4 ± 4	✓	
CSS5	4 ± 4	✓	
CSS6	4 ± 4	✓	
ADCS 3-Axis - Raw Sensor Measurements → Raw CSS7 to 10			
CSS7	4 ± 4	✓	
CSS8	4 ± 4	✓	
CSS9	4 ± 4	✓	
CSS10	4 ± 4	✓	

* Except for the aforementioned flags at the end of the **CubeComputer** section and the *Sun is Above Local Horizon* flag.
** If the **CubeADCS is** equipped with a Redundant Magnetometer.

- Expose the coarse Sun sensors to a bright light, one by one.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - Raw Sensor Measurements → Raw CSS1 to CSS10			
Raw measurements when **exposed** to light	> 50 (each channel)	✓	

6.1.1 CubeMag with Signal MCU

Test the **CubeMag** as sampled by the Signal MCU by following the procedure below:

- Go to *ADCS 3-Axis → Set Mode of Magnetometer Operation → Magnetometer Mode* and select *SigMainMag*. Transmit the command by clicking the green arrow.
- Go to *ADCS 3-Axis- ADCS state → Current ADCS State* and ensure that the *Magnetometer Range Error* is **not** checked. If it is checked, then the magnetometer is unable to measure a sufficient/overpowering magnetic field. This can be solved by ensuring that there is no contact to an anti-static mat or by placing the magnetometers away from motors, power supplies, large ferromagnetic objects, *etc.*
- Familiarise the **axes** of **both** magnetometers, as shown in Appendix A at the end of this document.

- Navigate to *ADCS 3-Axis - Raw Sensor Measurements* and verify the operation of the magnetometers by referring to the *Raw Magnetometer* and *Secondary Magnetometer Raw Measurements* (Redundant Magnetometer) boxes. Choose an axis on the **main** magnetometer and point it in the positive direction of the magnetic field lines (north) to align the axis with the magnetic vector. Now rotate the **main** magnetometer around this axis. The chosen axis must remain positive while the other two axes will both go negative and positive though the rotation. Repeat this for all three axes to verify polarities.

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - Raw Sensor Measurements			
Raw Magnetometer shows varying values in X, Y and Z directions correctly	✓	✓	
Secondary Magnetometer Raw Measurements shows relatively constant values in X, Y and Z directions *	✓	✓	

* If the **CubeADCS** is **not** equipped with a Redundant Magnetometer the X, Y and Z values will all be zeros.

- Navigate to *ADCS 3-Axis – ADCS Measurements* → *Magnetic Field vecto*r. Rotate the **main** magnetometer and verify that the magnetic field vector displays both positive and negative in X, Y, and Z directions correctly. Fill the following table accordingly:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis ADCS Measurements → Magnetic Field Vector			
Magnetic Field can measure both positive and negative in X, Y and Z directions	✓	✓	

6.1.2 CubeMag Redundant

This section is not applicable to CubeADCS unit CA2217.

6.1.3 CubeTorquers with Signal MCU

Test the **CubeRods** and **CubeCoil** (if included in assembly) as controlled by the Signal MCU by following the procedure below:

- Navigate to the *ADCS 3-Axis - Actuator-Commands* → *Set Magnetorquer Output* box. Command the magnetorquer coil (X-axis) to maximum positive direction by setting the slider bar all the way to the right next to *Commanded X Magnetorquer duty cycle* and send the command by clicking on the green arrow.

- Confirm the current measurement and the direction of magnetic field in the table below. The direction can be confirmed by either placing a compass directly in line with the magnetorquer and observing the field direction or by using an external magnetometer. Note that the magnetorquer pulses on for a maximum of 0.8 seconds and then switches off.
- Change the command to maximum negative and confirm the current and direction again.
- Command the magnetorquer to zero to turn off the magnetorquer.
- Repeat these steps for the Y and Z magnetorquer rods and record the required values in the next table.

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Power → ADCS Misc/CubeControl Current Measurements			
(X) Magnetorquer Current	122 ± 12 DN	133.4 DN ✓	
X Magnetorquer direction of magnetic field with max positive command	+X	✓	
X Magnetorquer direction of magnetic field with max negative command	-X	✓	
(Y) Magnetorquer Current	57 ± 10 DN	65.7 DN ✓	
Y Magnetorquer direction of magnetic field with max positive command	+Y	✓	
Y Magnetorquer direction of magnetic field with max negative command	-Y	✓	
(Z) Magnetorquer Current	57 ± 10 DN	65.7 DN ✓	
Z Magnetorquer direction of magnetic field with max positive command	+Z	✓	
Z Magnetorquer direction of magnetic field with max negative command	-Z	✓	

- Ensure that all the magnetorquer duty cycles are set to zero (turned off) before continuing the health check.

- Switch **off CubeControl**'s **Signal MCU** by selecting *PowOff* in the drop-down menu next to *CubeControl Signal Power Selection* in the *ADCS 3-Axis → ADCS Power Control* box and transmit the command by clicking on the green arrow.

6.2 CubeControl Motor MCU

Follow the instructions below to perform the **CubeControl** Motor MCU health check:

- Ensure sure that all nodes are selected *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box before proceeding.
- Switch **on CubeControl**'s **Motor MCU** by selecting *PowOn* in the drop-down menu next to *CubeControl Motor Power Selection* in the *ADCS 3-Axis → ADCS Power Control* box and transmit the command by clicking on the green arrow.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS state → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsEnabled	✓	
CubeControl Motor Enabled	Checked	✓	
All other checkboxes*	Unchecked	✓	
ADCS 3-Axis - ADCS Measurements → Rate Sensor Rates			
X-rate can be positive or negative (tilt the unit)	✓	✓	
Y-rate can be positive or negative (tilt the unit)	✓	✓	
Z-rate can be positive or negative (tilt the unit)	✓	✓	
ADCS 3-Axis - ADCS Measurements → Rate Sensor Rates			
X Angular Rate (unit stationary)	$\lvert \omega_x \rvert < 2$ °/s	-0.018 °/s ✓	
Y Angular Rate (unit stationary)	$\lvert \omega_y \rvert < 2$ °/s	-0.668 °/s ✓	
Z Angular Rate (unit stationary)	$\lvert \omega_z \rvert < 2$ °/s	-0.275 °/s ✓	

* Except for the aforementioned flags at the end of the **CubeComputer** section and the *Sun is Above Local Horizon* flag.

6.2.1 CubeMag with Motor MCU

Test the main **CubeMag** as sampled by the Motor MCU by following the procedure below:

- Go to *ADCS 3-Axis* → *Set Mode of Magnetometer Operation* → *Magnetometer Mode* and select *MotorMainMag*. Click the green arrow to transmit the command.
- Rotate the **main** magnetometer and confirm that the values in the *ADCS 3-Axis - Raw Sensor Measurements* → *Raw Magnetometer* box.
- Navigate to *ADCS 3-Axis - Raw Sensor Measurements* and verify the operation of the magnetometer(s) by using the *Raw Magnetometer* and *Secondary Magnetometer Raw Measurements* (Redundant) boxes. Choose an axis on the **main** magnetometer and point it in the positive direction of the magnetic field lines (north) to align the axis with the magnetic vector. Now rotate the **main** magnetometer around this axis. The chosen axis must remain positive while the other two axes will both go negative and positive though the rotation. Repeat this for all three axes to verify polarities.

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - Raw Sensor Measurements			
Raw Magnetometer shows varying values in X, Y and Z directions correctly	✓	✓	
Secondary Magnetometer Raw Measurements shows zeros in X, Y and Z directions	✓	✓	

- Navigate to *ADCS 3-Axis – ADCS Measurements* → *Magnetic Field vecto*r. Rotate the **main** magnetometer and verify that the magnetic field vector displays both positive and negative in X, Y, and Z directions correctly. Once complete, fill in the following table accordingly:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis ADCS Measurements → Magnetic Field Vector			
Magnetic Field can measure both positive and negative in X, Y and Z directions	✓	✓	

6.2.2 CubeTorquers with Motor MCU

Test the **CubeRods** and **CubeCoil** (if included in assembly) as controlled by the Motor MCU by following the procedure below:

- Navigate to the *ADCS 3-Axis - Actuator-Commands* → *Set Magnetorquer Output* box. Command the magnetorquer coil (X-axis) to maximum positive direction by setting the slider bar all the way to the right next to *Commanded X Magnetorquer duty cycle* and send the command by clicking on the green arrow.

- Confirm the current measurement and the direction of magnetic field in the table below. The direction can be confirmed by either placing a compass directly in line with the magnetorquer and observing the field direction or by using an external magnetometer. Note that the magnetorquer pulses on for a maximum of 0.8 seconds and then switches off.
- Change the command to maximum negative and confirm the current and direction again.
- Command the magnetorquer to zero to turn off the magnetorquer.
- Repeat these steps for the Y and Z magnetorquer rods and record the required values in the following table.

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Power → ADCS Misc/CubeControl Current Measurements			
(X) Magnetorquer Current	129 ± 17 DN	139.6 DN ✓	
X Magnetorquer direction of magnetic field with max positive command	+X	✓	
X Magnetorquer direction of magnetic field with max negative command	-X	✓	
(Y) Magnetorquer Current	65 ± 11 DN	72.9 DN ✓	
Y Magnetorquer direction of magnetic field with max positive command	+Y	✓	
Y Magnetorquer direction of magnetic field with max negative command	-Y	✓	
(Z) Magnetorquer Current	65 ± 11 DN	72.6 DN ✓	
Z Magnetorquer direction of magnetic field with max positive command	+Z	✓	
Z Magnetorquer direction of magnetic field with max negative command	-Z	✓	

- Ensure that all the magnetorquer duty cycles are set to zero (turned off) before continuing the health check.

6.3 CubeControl Motor CubeWheel Health Check

This section is not applicable to CubeADCS unit CA2217.

7. CubeWheels

Note 1: Current measurements on **CubeWheels** are dependent on two main factors (among others):

- The temperature at which the test is conducted.
- How long the wheels have been stationary (i.e. not running) before performing this health check.

To obtain similar results as obtained by CubeSpace for the various CubeWheel current outputs, it is recommended to the user that each CubeWheel be run at 6000 RPM for at least 1 hour prior to performing the checks that follow in this section. This is the CubeWheel "burn-in" period, which will ensure that the difference in testing environments between the user and CubeSpace, will have little impact on the performance of the CubeWheels irrespective of temperature differences, etc.

Follow the instructions below to spin up the CubeWheels prior to the health check:

- Ensure sure that all nodes are selected *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box before proceeding.
- Switch on *CubeWheel1Power*, *CubeWheel2Power*, and *CubeWheel3Power Power Selection* by selecting *PowOn* in the *ADCS 3-Axis → ADCS Power Control* box.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS state → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsEnabled	✓	
CubeWheel1 Enabled	Checked	✓	
CubeWheel2 Enabled	Checked	✓	
CubeWheel3 Enabled	Checked	✓	
All other checkboxes*	Unchecked	✓	

* Except for the aforementioned error flags and *Sun is Above Local Horizon*.

- Navigate to the *ADCS 3-Axis – Actuator Commands → Set Wheel Speed* box. Set the commanded X, Y and Z speeds to 6000 rpm.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	6000 ± 5 rpm	✓	
Y Wheel Speed	6000 ± 5 rpm	✓	
Z Wheel Speed	6000 ± 5 rpm	✓	

- If all CubeWheels are spinning at 6000 RPM, set a timer for 1 hour and wait for the CubeWheels to complete their "burn-in" period.
- After the "burn-in" period has passed, navigate to the *ADCS 3-Axis – Actuator Commands → Set Wheel Speed* box. Set the commanded X, Y and Z speed to 0 rpm.
- After 10 s, verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	0	✓	
Y Wheel Speed	0	✓	
Z Wheel Speed	0	✓	

The CubeWheel "burn-in" is complete and the individual CubeWheel health checks can now commence.

7.1 CubeWheel1

Follow the instructions below to perform the **CubeWheel1** health check:

- Ensure sure that all nodes are selected *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box before proceeding.
- Switch on *CubeWheel1Power Power Selection* by selecting *PowOn* in the *ADCS 3-Axis → ADCS Power Control* box.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS state → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsEnabled	✓	
CubeWheel1 Enabled	Checked	✓	

All other checkboxes*	Unchecked	✓	

* Except for the aforementioned error flags and *Sun is Above Local Horizon*.

- Navigate to the *ADCS 3-Axis – Actuator Commands* → *Set Wheel Speed* box. Set the commanded X speed to 4000 rpm.
- After 10 seconds the wheel will settle to the commanded speed. Listen to the wheel to ensure that the correct one is spinning up and that the other two wheels are silent.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	4000 ± 5 rpm	✓	
Y Wheel Speed	0	✓	
Z Wheel Speed	0	✓	
ADCS 3-Axis - ADCS Power → Wheel Currents			
Wheel1Current	13 ± 7 mA	14.33 mA ✓	
Wheel2Current	0	0 mA ✓	
Wheel3Current	0	0 mA ✓	

- Navigate to the *ADCS 3-Axis – Actuator Commands* → *Set Wheel Speed* box. Set the commanded X speed to -2000 rpm.
- After 10 s verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	-2000 ± 5 rpm	✓	
Y Wheel Speed	0	✓	
Z Wheel Speed	0	✓	
ADCS 3-Axis - ADCS Power → Wheel Currents			
Wheel1Current	10 ± 4 mA	8.75 mA ✓	
Wheel2Current	0	0 mA ✓	
Wheel3Current	0	0 mA ✓	

- Navigate to the *ADCS 3-Axis – Actuator Commands* → *Set Wheel Speed* box. Set the commanded X speed to zero rpm.
- After 10 s verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	0	✓	
Y Wheel Speed	0	✓	
Z Wheel Speed	0	✓	
ADCS 3-Axis - ADCS Power → Wheel Currents			
Wheel1Current	6 ± 2 mA	5.84 mA ✓	
Wheel2Current	0	0 mA ✓	
Wheel3Current	0	0 mA ✓	

- Switch off *CubeWheel1Power Power Selection* by selecting *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box.

7.2 CubeWheel2

Follow the instructions below to perform the **CubeWheel2** health check:

- Ensure sure that all nodes are selected *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box before proceeding.
- Switch on *CubeWheel2Power Power Selection* by selecting *PowOn* in the *ADCS 3-Axis → ADCS Power Control* box.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS → ADCS state → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsEnabled	✓	
CubeWheel2 Enabled	Checked	✓	
All other checkboxes*	Unchecked	✓	

* Except for the aforementioned error flags and *Sun is Above Local Horizon*.

- Navigate to the *ADCS 3-Axis – Actuator Commands → Set Wheel Speed* box. Set the commanded Y speed to 4000 rpm.
- After 10 seconds the wheel will settle to the commanded speed. Listen to the wheel to ensure that the correct one is spinning up and that the other two wheels are silent.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	0	✓	
Y Wheel Speed	4000 ± 5 rpm	✓	
Z Wheel Speed	0	✓	
ADCS 3-Axis - ADCS Power → Wheel Currents			
Wheel1Current	0	0 mA ✓	
Wheel2Current	13 ± 7 mA	13.51 mA ✓	
Wheel3Current	0	0 mA ✓	

- Navigate to the *ADCS 3-Axis – Actuator Commands → Set Wheel Speed* box. Set the commanded Y speed to -2000 rpm.
- After 10 s verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	0	✓	
Y Wheel Speed	-2000 ± 5 rpm	✓	
Z Wheel Speed	0	✓	
ADCS 3-Axis - ADCS Power → Wheel Currents			
Wheel1Current	0	0 mA ✓	
Wheel2Current	10 ± 4 mA	8.82 mA ✓	
Wheel3Current	0	0 mA ✓	

- Navigate to the *ADCS 3-Axis – Actuator Commands → Set Wheel Speed* box. Set the commanded Y speed to zero rpm.
- After 10 s verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	0	✓	
Y Wheel Speed	0	✓	
Z Wheel Speed	0	✓	
ADCS 3-Axis - ADCS Power → Wheel Currents			

Wheel1Current	0	0 mA ✓	
Wheel2Current	6 ± 2 mA	5.82 mA ✓	
Wheel3Current	0	0 mA ✓	

- Switch off *CubeWheel2Power Power Selection* by selecting *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box.

7.3 CubeWheel3

Follow the instructions below to perform the **CubeWheel3** health check:

- Ensure sure that all nodes are selected *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box before proceeding.
- Switch on *CubeWheel3Power Power Selection* by selecting *PowOn* in the *ADCS 3-Axis → ADCS Power Control* box.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS → ADCS state → Current ADCS State			
Attitude Estimation Mode	EstNone	✓	
Control Mode	ConNone	✓	
ADCS Run Mode	AdcsEnabled	✓	
CubeWheel3 Enabled	Checked	✓	
All other checkboxes*	Unchecked	✓	

* Except for the aforementioned error flags and *Sun is Above Local Horizon*.

- Navigate to the *ADCS 3-Axis – Actuator Commands → Set Wheel Speed* box. Set the commanded Z speed to 4000 rpm.
- After 10 seconds the wheel will settle to the commanded speed. Listen to the wheel to ensure that the correct one is spinning up and that the other two wheels are silent.
- Verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	0	✓	
Y Wheel Speed	0	✓	
Z Wheel Speed	4000 ± 5 rpm	✓	
ADCS 3-Axis - ADCS Power → Wheel Currents			

Wheel1Current	0	0 mA ✓	
Wheel2Current	0	0 mA ✓	
Wheel3Current	13 ± 7 mA	14.65 mA ✓	

- Navigate to the *ADCS 3-Axis – Actuator Commands* → *Set Wheel Speed* box. Set the commanded Z speed to -2000 rpm.
- After 10 s verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	0	✓	
Y Wheel Speed	0	✓	
Z Wheel Speed	-2000 ± 5 rpm	✓	
ADCS 3-Axis - ADCS Power → Wheel Currents			
Wheel1Current	0	0 mA ✓	
Wheel2Current	0	0 mA ✓	
Wheel3Current	10 ± 4 mA	8.65 mA ✓	

- Navigate to the *ADCS 3-Axis – Actuator Commands* → *Set Wheel Speed* box. Set the commanded Z speed to zero rpm.
- After 10 s verify the following:

Test / Task	Expected Result	CubeSpace Result	Client Result
ADCS 3-Axis - ADCS Measurements → Wheel Speed			
X Wheel Speed	0	✓	
Y Wheel Speed	0	✓	
Z Wheel Speed	0	✓	
ADCS 3-Axis - ADCS Power → Wheel Currents			
Wheel1Current	0	0 mA ✓	
Wheel2Current	0	0 mA ✓	
Wheel3Current	6 ± 2 mA	5.35 mA ✓	

- Switch off *CubeWheel3Power Power Selection* by selecting *PowOff* in the *ADCS 3-Axis → ADCS Power Control* box.

8. CubeStar

This section is not applicable to CubeADCS unit CA2217.

Appendix A: CubeADCS Physical Axes

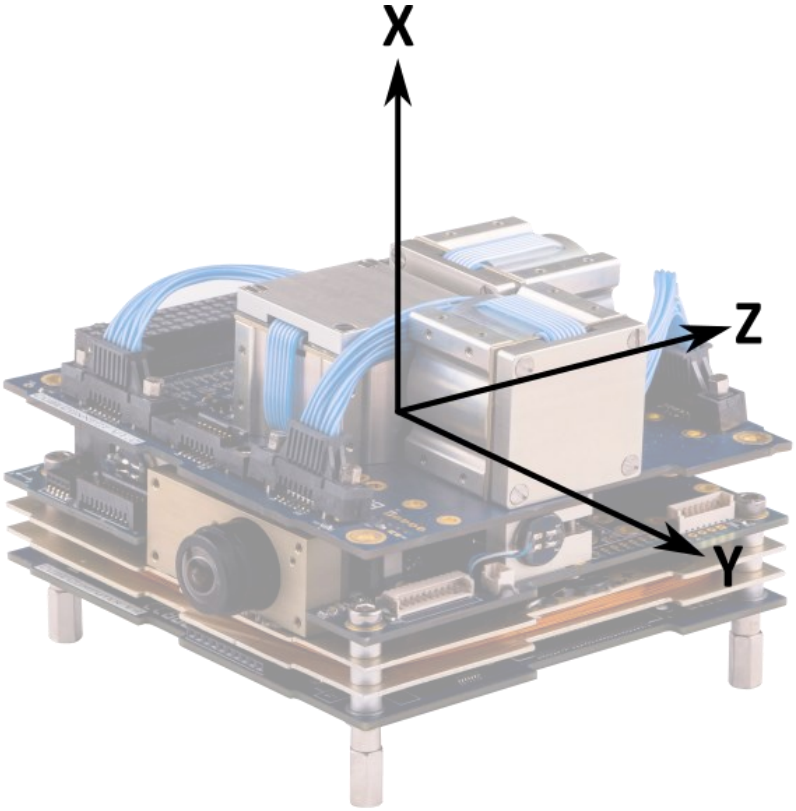


Figure 16 – CubeADCS Physical Axes

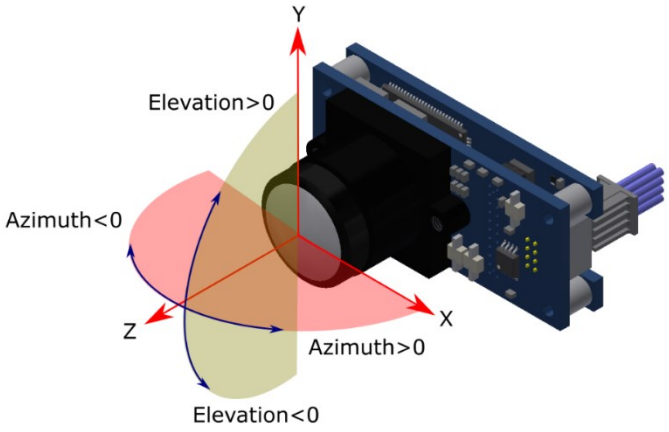


Figure 17 – Small integrated CubeSense V3.0 axes

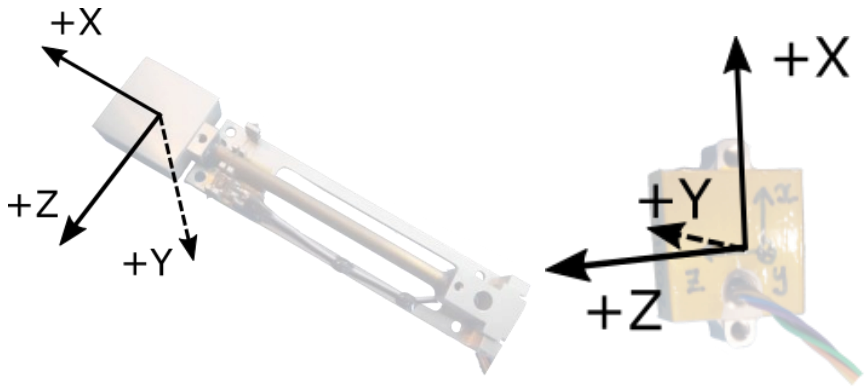


Figure 18 – Deployable and redundant magnetometer axes

Appendix B: Typical CubeSense Images

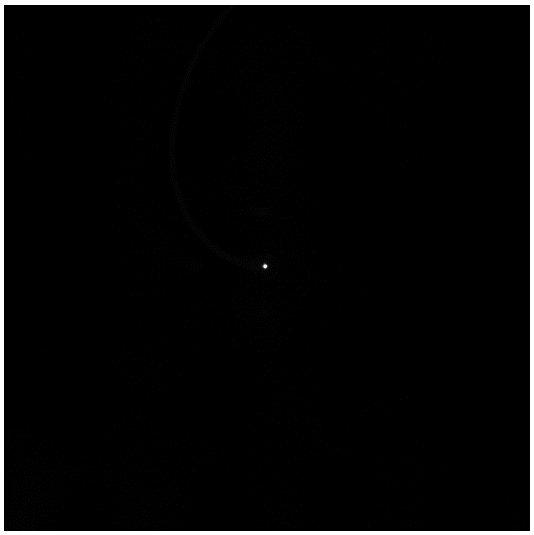


Figure 19 – Typical CubeSense sun sensor image for successful Sun detection

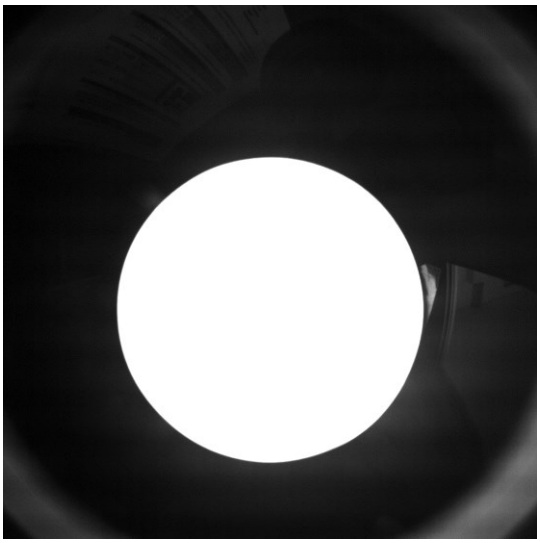


Figure 20 – Typical CubeSense nadir sensor image for successful Nadir detection