CubeSpace

Software Guide

Document number	CS-DEV.GD.SWB-01
Version	1.00
Date	2022/05/20
Prepared by	Oliver Gries, Christo Groenewald, Lourens Visagie
Reviewed by	CJG, CL
Approved by	Christo Groenewald
Distribution list	Confidential

Revision History

Version	Author(s)	Pages	Date	Description
1.00	Oliver Gries, Christo Groenewald	All	2022/05/20	This document supersedes and replaces the "Software Bundle Quick Start Guide" at version 10.0 to comply with CubeSpace document naming convention and for documentation consistency.

Reference Documents

[1] None

List of Acronyms/Abbreviations

ACP	ADCS Control Program
ADCS	Attitude Determination and Control System
CAN	Controller Area Network
COTS	Commercial Off The Shelf
CSS	Coarse Sun Sensor
CVCM	Collected Volatile Condensable Materials
DUT	Device Under Test
EDAC	Error Detection and Correction
EHS	Earth Horizon Sensor
EM	Engineering Model
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
ESD	Electrostatic Discharge
FDIR	Fault Detection, Isolation, and Recovery
FM	Flight Model
FSS	Fine Sun Sensor
GID	Global Identification
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
GYR	Gyroscope
I2C	Inter-Integrated Circuit
ID	Identification
LTDN	Local Time of Descending Node
LEO	Low Earth Orbit
MCU	Microcontroller Unit
MEMS	Microelectromechanical System
MTM	Magnetometer
MTQ	Magnetorquer
NDA	Non-Disclosure Agreement
OBC	On-board Computer
PCB	Printed Circuit Board
RTC	Real-Time Clock
RWA	Reaction Wheel Assembly
RW	Reaction Wheel
SBC	Satellite Body Coordinate
SOFIA	Software Framework for Integrated ADCS
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SSP	Sub-Satellite Point
STR	Star Tracker
TC	Telecommand
TCTLM	Telecommand and Telemetry (protocol)
TID	Total Ionizing Dose
TLM	Telemetry
TML	Total Mass Loss
UART	Universal Asynchronous Receiver/Transmitter

Table of Contents

Table of Tables

1 Introduction

This document is intended to be a Software Guide and serves as the entry point into the CubeSpace Software Release package. It guides the user through how to get started with navigating the folder structure and using the contents of the package.

The software release package contains all the tools and files required which, together with the physical CubeProducts, allow the user to get up and running with the CubeSpace Gen 2 ADCS System.

1.1 User's computer environment requirements

The user's environment requirements for accessing and utilising the Software package are shown in Table 1.

Table 1: User computer environment requirements

Item	Requirement
OS	Windows
Architecture	X86/x64
.Net framework	Latest version **Note:** CubeSpace's GUI tools are built using Microsoft .NET Framework and are therefore required to run on a Windows machine. Both x86 and x64 architectures are supported.

2 Versioning

All CubeSpace Gen2 ADCS Software is built around a common framework.

The supplied CubeSpace firmware, software tools, and documentation, together with the physical CubeProducts, form the CubeADCS.

Each CubeProduct making up the integrated CubeADCS system is individually versioned. Such a CubeProduct version changes independently to any of the other system CubeProducts, and represents changes made to that specific CubeProduct.

So too, the software package is versioned where a specific version represents a moment in time where all CubeProducts of the system have been tested and proven to function together. At this point, a Software Release package is generated with the associated Software package version and is released to the client.

2.1 The "manifest.txt" file

A file named "***manifest.txt***" is available in the root folder of the Software package. This file contains build information relating to the Software package which it forms part of. This information is provided as a means to easily verify the contents of this Software package, as well as for client support.

Descriptions of the contents of the "***manifest.txt***" are listed in Table 2.

Table 2: Contents of manifest.txt

Item	Description
version	Software Package version
date	Date of generating this Software Package.

Item	Description
time	Time of generating this Software Package.
git branch	Git branch that this Software Package was built from.
git hash	Git hash of HEAD from which this Software Package was built from.
git short hash	Git short hash of HEAD from which this Software Package was built from.
<application>-<component>:<version>	CubeProduct firmware versions for this Software Package

2.2 Software Package Version

The Software Package version is shown in the filename of the Software package. For confirmation of the software version, check the contents of the "***manifest.txt***" file.

A semantic versioning scheme is used consisting of four fields as shown in Table 3.

Table 3: System Version Fields

Field	Description
Major	Change breaking backwards compatibility
Minor	Feature addition
Patch	Bugfix on an existing feature
Build	Build number since previous patch

2.3 Firmware Version

The firmware versions for each of the applications of each CubeProduct is documented in the "***manifest.txt***" file.

A semantic versioning scheme is used consisting of three fields as shown in Table 4.

Table 4: Firmware Version Fields

Field	Description
Major	Change breaking backwards compatibility. Matches major PCB version of the CubeProduct
Minor	Feature addition
Patch	Bugfix on an existing feature

2.4 Application Version

Applications are the embedded software that execute on CubeProducts. The naming convention of these applications is common between products.

A list of application types is shown in Table 5.

Table 5: List of Application types

Application	Description
flash-bootloader	Responsible for loading new applications on to the device, as well as for FDIR.
control-program	Main application that performs the ADCS functions of the CubeProduct.

Application	Description
health-check	Performs low-level hardware checks to ensure the product is in good health.

2.5 Folder Structure

The Software release package is typically provided to the user as a zipped folder structure made available on (typically) a cloud based folder, shared between CubeSPace and the client, provided together with the CubeProducts for first deliveries. There-after, new Software release packages are made available via the agreed mechanism i.e. either via shared folder locations or FTP.

When the software release package is unzipped, the user is presented with the folder structure and associated documentation following below and which should be saved to a location from where the user intends to work from.

Notes:

1. The generic entries of type "cube-<*component*> *-x"* below are replaced in the software release package with folder names of the physical CubeProducts while the "x" denotes the major version of it. For example "cube-computer-5", or "cube-mag-4" and so on.
2. The generic entries of type "<*application*>*-x*" below are replaced in the software release package with folder names of the application type as defined in Table 5 while the "x" denotes the major version of it. For example "control-program-8"
3. Release packages are generated based on the hardware and software required by the user. Some of the folders listed below might not be present in a provided release package because it is optional.

… Software Guide
… manifest.txt
… **api**
    … \ cube- <*component*> - *x*
        … \ <*application*> - *x*
… **docs**
    … \ bootloader
        … \ Application Manual
        … \ Firmware Reference Manual
    … eos
        … \ User Manual
    … firmware
        … \ Common Firmware Reference Manual
… **firmware**
    … \ cube- <*component*> - *x*
        … \ <*application*>
            … \ missions
                … \ Mission Specific Config Files
            … \ Application Binary File
… **tools**
    … \ cube-support
    … \ cube-toolbox
    … \ eos

Please note that although “Mission Specific Config files” are provided in both binary and human readable format, the user is strongly advised not to edit these files if changes are required. Instead, the user should please contact CubeSpace directly to do so.