SSC20-S1-07

Integrating the Cubesat Space Protocol into GSOC's Multi-Mission Environment

Lukas Grillmayer, Saskia Arnold
German Aerospace Center (DLR)
Münchener Str. 20, 82234 Weßling, Germany; +49 08153 28 3407
lukas.grillmayer@dlr.de

ABSTRACT

CubeL is the first COTS cubesat to be operated by the German Space Operations Center (GSOC) utilizing cubesat space protocol (CSP). Scheduled for launch in summer 2020, it will initially be monitored and controlled via UHF using a compatible COTS ground segment to perform an IOD of "OSIRIS4CubeSat", a miniaturized OSIRIS space-to-ground laser communication terminal developed by DLR-KN in cooperation with Tesat-Spacecom. Afterwards, CubeL will be integrated into the GSOC multi-mission environment and be operated via S-Band. The GSOC ground segment architecture and software focuses on institutionally standardized communication, such as CCSDS frame (132.0-B-2) and packet standards (133.0-B-1) and the ECSS packet utilization standard (E-ST-70-41C). At the core of GSOC's multi-mission environment is the SCOS-2000 based monitoring and control system "GECCOS", which supports all satellite missions currently operated by GSOC. CubeL however depends on CSP for most communication. This page briefly introduces the CubeL mission and ground segment design, presents relevant protocols and the subsequent tailoring of CCSDS protocol features before describing the required CSP to CCSDS adapter, to enable communication between CubeL and GECCOS. For concept validation a minimal prototype is tested against the CubeL engineering model. This work concludes with a critical review of the chosen approach.

.

INTRODUCTION

For more than half a century, the GSOC has been collecting experience in operating a large variety of spacecraft, with recent activities ranging from unmanned low- and medium earth orbiting earth observation and science missions (e.g. TanDEM-X, GRACE Follow-on, Eu:CROPIS), over geostationary communication and navigation satellites (e.g. EDRS-C, HAG-1) to human spaceflight (e.g. ISS-Columbus) [1]. All satellites operated at GSOC make use of institutionally standardized communication protocols such as the CCSDS frame (132.0-B-2) and packet standards (133.0-B-1) and the ECSS packet utilization standard (E-ST-70-41C). Over the past two decades this commonality enabled GSOC to develop ground segment system architectures and tools which can be shared and reused by many projects with only minor adaptions. As a result these concepts and tools have been tested excessively and exhibit a high level of maturity, which form the basis of GSOC's multi mission (MUM) environment. This approach has proven to be cost-effective not only concerning software development and system maintenance but also for operations. Subsystem engineers and command operators have deep knowledge about features and functionality of MUM systems, so they are already familiar with the available toolset. This supports them in their effort to efficiently prepare and execute a mission.

GSOC is embedded within the German Aerospace Center (DLR) as part of the Institute for Astronaut Training and Space Operations. As DLR is a large research and development institution in Germany many applications of DLR are intended for the space environment and therefore it frequently has the need for in-orbit demonstrations (IOD). Most science driven satellites in low-earth-orbit (LEO) are host to novel hardware or software IODs with the purpose of showing technical feasibility and furthering the component's technology readiness level (TRL). With the ever increasing number of spacecraft [2], possibilities for "guest payload" IODs are increasing. However, these items are rarely a design driver and have little to no influence on orbit selection and mission timeline, which is determined by the main payload. While this scenario has the potential advantage of being low cost, this dependency comes with many disadvantages, such as unforeseeably long launch delays caused by the host satellite, or loss of flight opportunity due to delays within the IOD project. Dedicated cubesat IOD missions offer the chance to overcome the described issues while at the same time keeping cost reasonably low.

In late 2017 GSOC was tasked with preparing an operational concept for a DLR internal cubesat IOD mission, which was later called CubeL. Originally scheduled for launch in September 2018 this posed a challenge as there were only 6 months between proposal acceptance and launch, which is reflected in the mission profile. GSOC welcomed this opportunity to operate its first COTS cubesat and to gather experience with the "new space" approach to spaceflight. For CubeL to be able to be operated sustainably within the GSOC MUM environment and utilize integrated ground station networks it would have to communicate using supported protocols and frequency bands. Natively CubeL uses CSP via UHF-band whereas GSOC requires CCSDS protocols and is using S-, X-, and Ka-band on a daily basis. However this protocol was not a standard option supported by the spacecraft manufacturer and the available timespan did not allow for extensive development and test campaigns prior to launch. As a result the GSOC MUM integration has experimental characteristics. To avoid additional risk for the IOD mission and allocate more preparation time for the GSOC MUM integration it was decided to initially use a compatible COTS UHF-band ground segment and later transition to S-band based on predefined interfaces and protocols. Considering available COTS hardware options a software defined radio (SDR) and an S-band patch antenna were late additions to the CubeL satellite. The required onboard software would be developed by the manufacturer and the ground segment solution by GSOC, which is the focus of this work.

The following chapters briefly introduce the CubeL mission, satellite, and ground segment before discussing the design, and development process of the required CSP to CCSDS adapter. As of writing this paper, the CubeL project is in ECSS phase C/D as it experienced significant delays within the space segment and is scheduled for launch in summer 2020.

CUBEL MISSION

CubeL is an IOD mission of a highly compact and miniaturized laser communications terminal (LCT) called "OSIRIS4CubeSat" (O4C), which was developed by the DLR Institute for Communication and Navigation (IKN) in cooperation with Tesat-Spacecom (Tesat) as part of the Optical Space Infrared Downlink System (OSIRIS) program [2]. With the small size of 0.3U it is optimized for usage in nanosatellites following the cubesat design specification, offering high bandwidth space to ground data transmissions of up to 100 Mbit/s. This LCT enables nanosatellites to be utilized for missions requiring transmission of large amounts of data such as camera images. Following successful in orbit demonstration Tesat intends to make O4C commercially available, which will be marketed as "CubeLCT" [3]. After careful market evaluation by IKN the Danish nanosatellite manufacturer GomSpace A/S was selected as supplier for the 3U satellite, which will perform platform and payload assembly, integration, test, and verification (AITV) as well as launch and early orbit phase (LEOP). After successful handover, GSOC will continue operations for the duration of the mission, which is initially set to three years.

The CubeL mission goals are categorized in primary and secondary objectives, whereas the first 15 months of phase E2 are dedicated to the two primary objectives.

First is the IOD of the O4C terminal which aims to execute 40 successful optical links with data transmission to show reliable LCT link performance. The downlinked data shall include earth observation images taken by the onboard camera system. These initial activities are referred to as phase E2a and will be controlled and monitored using COTS UHF-band equipment.

Second primary objective is the integration of CubeL into GSOC MUM-environment using S-Band to utilize the same infrastructure, system architectures, software tools, and services used by other satellite missions operated by GSOC.

Secondary mission objectives include a demonstration of sending telecommands via laser uplink and a long term study of O4C to evaluate any potential degradation due to the space environment.

The CubeL satellite is a 3U cubesat manufactured by GomSpace A/S. Figure 1 shows a rendering of the satellite exterior prominently featuring four deployable UHF antennas, an S-band patch antenna, and a protruding full HD earth observation camera system. The middle unit hosts the O4C LCT module and a star tracker (not shown). All remaining surfaces are covered by solar cells to satisfy the power requirements during operations.

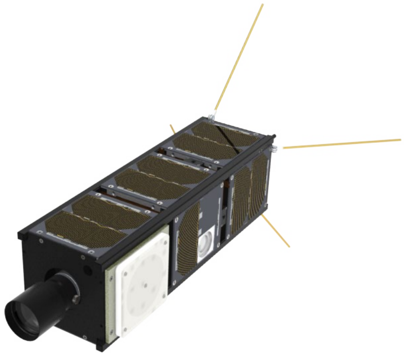


Figure 1: Rendering of the CubeL satellite [4]

CUBEL GROUND SEGMENT

The CubeL ground segment design is split into two phases, for UHF- and S-band operations. This approach was chosen due to the early stringent time requirements and the experimental nature of CubeL S-band communication and integration into the GSOC MUM environment.

Phase E2a UHF Ground Segment

During phase E2a, CubeL will be operated only via UHF using a compatible COTS ground segment, which is proven to work reliably out of the box as GSOC had neither integrated UHF ground stations nor a CSP capable monitoring and control system (MCS). Figure 2 shows an abstract overview of the minimalist E2a ground segment, which is further augmented by GSOC MUM components as described below. GSOC security policy requires external hardware and software to be isolated in separate networks and thus may not be integrated into the MUM environment directly.

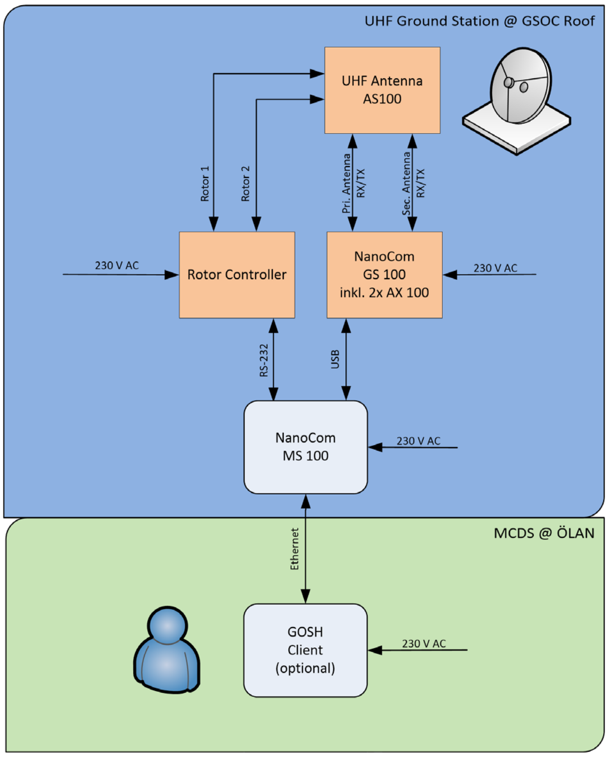


Figure 2: Overview of the E2a ground segment

Within this setup, CSP is the prevalent protocol used for communication between ground segment components and the space segment. The "CSP-Terminal" software is also referred to as GomSpace Shell (GOSH), which is the main TMTC interface used for operating the spacecraft. Further optional software additions include a telemetry visualization solution called "GSWeb", which is operated in parallel to the GSOC wide established "Satmon". Preceding usage in phase E2b, latter software can draw on existing network infrastructure for data delivery to Satmon@Home as well as previous user experience of all GSOC personnel and most principal investigators, requiring only little training. Further, the concurrent usage increases redundancy in phase E2a and allows for early validation efforts important for phase E2b.

During phase E2a the primary O4C IOD mission objective is performed. In parallel, the ground segment design for phase E2b will be implemented, not interfering with ongoing operations.

Phase E2b S-Band Ground Segment

The phase E2b ground segment design aims to extensively utilize existing GSOC MUM capabilities with as little custom additions as possible to accommodate the CubeL missions. All communication between space and ground shall be performed using S-band and CCSDS protocols. Figure 3 shows an overview of the fully integrated ground segment with instances of common GSOC MUM tools.

For operators, the realtime TMTC interface with all GSOC MUM missions is GECCOS [5], an MCS system based on ESA SCOS-2000, which is developed and maintained in-house. For telemetry visualization and analysis Satmon has been established as the tool of choice for operators, subsystem engineers, and principal investigators (PI) alike as it provides intuitive access to both realtime data with very little delay and large amounts of offline data with low retrieval latencies. The Multimission Offline Processing System handles the processing of recorded satellite telemetry which is excluded from the realtime data transmission channel and forwarded after a contact due to bandwidth limitations of some remote ground stations. Over the past decades the DLR ground station network has grown to a global network of ground station service providers and includes DLR operated sites in Weilheim, Germany (WHM) and O'Higgins, Antarctica (OHG), which allow for GSOC's fault tolerant and highly flexible ground segments.

Despite the conversion from UHF- to S-band based operations, the interface to PIs does not change. The ground segment design intends to make the transition as smooth and transparent as possible to not cause service outages, data gaps or any other inconvenience within the customer experience.

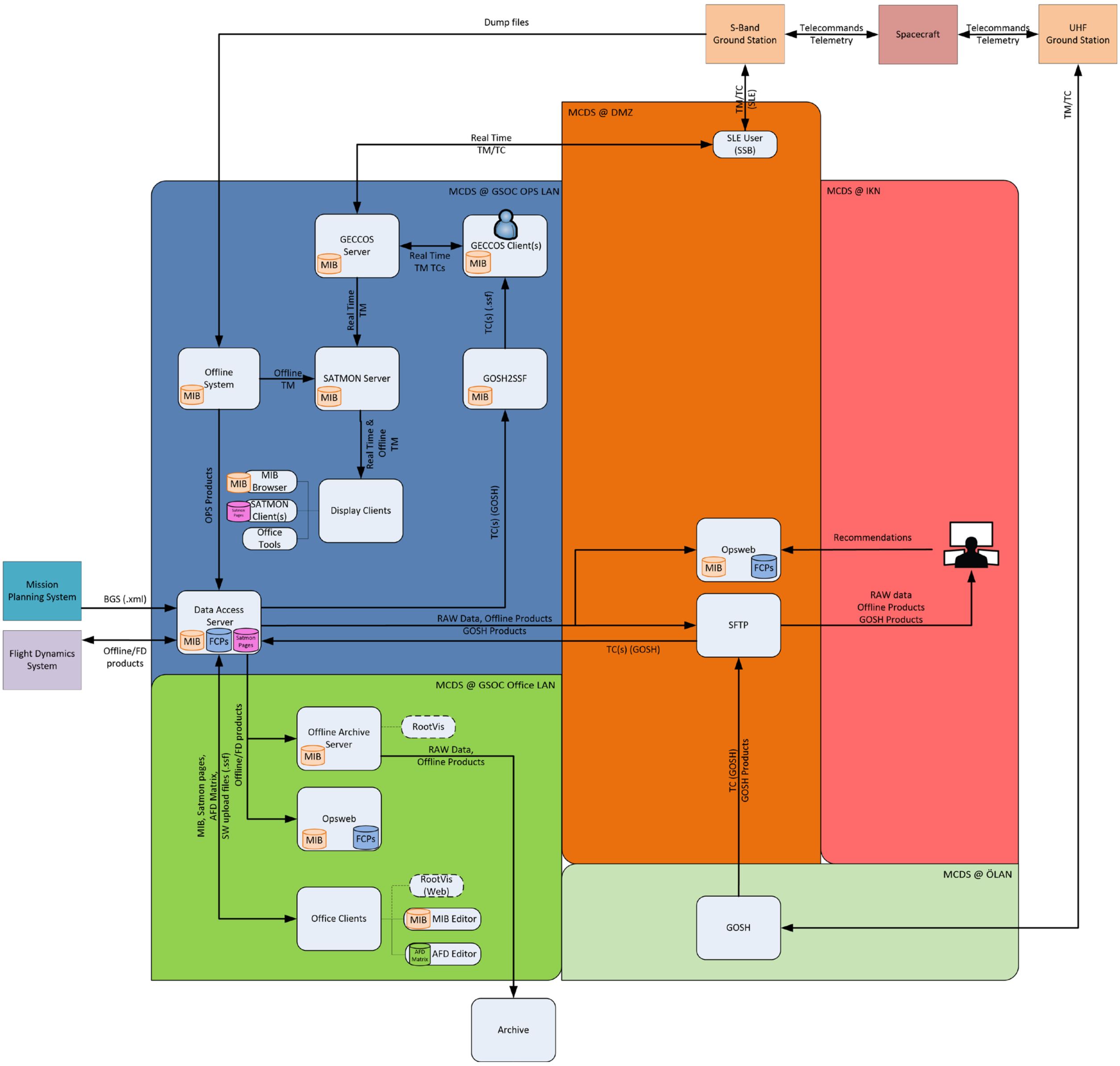


Figure 3: Overview of the E2b ground segment

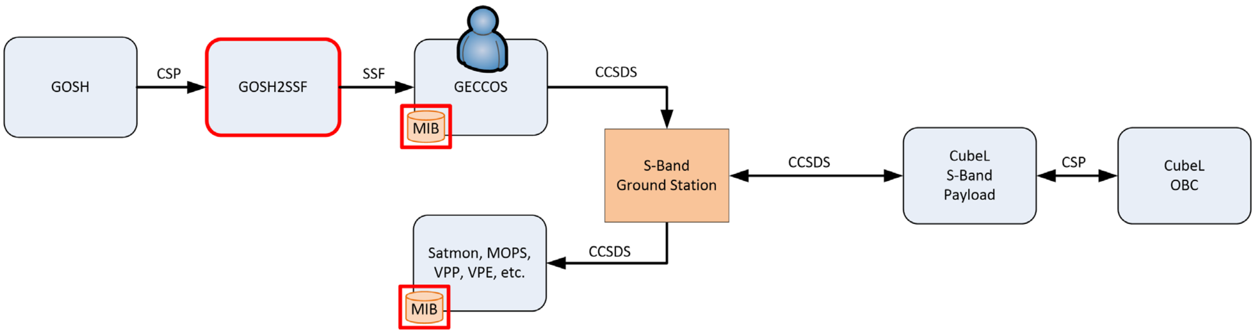


Figure 4: Phase E2b TMTC data flow and custom components highlighted in red

Figure 4 highlights the custom software components required for CubeL to interface with GSOC MUM. For initial ingestion of novel commands “GOSH2SSF” handles the conversion from GOSH CSP commands to GECCOS Saved Stack Files (SSF). For GECCOS to be able to encode and decode telemetry and commands it needs a description of all packets and parameters in form of the Mission Information Base (MIB).

By design it is intended for the CCSDS space packet protocol and is able to support the ECSS packet utilization standard (PUS). The CubeL system engineers aim to use the versatility of the MIB for it to serve as an adapter for the cubesat space protocol, allowing GECCOS to talk and interpret CSP in realtime. Within the GSOC MUM environment many tools require a MIB to decode, process, and visualize both realtime and offline telemetry.

PROTOCOLS

The following sections briefly introduce the protocols under discussions and highlights features relevant within the scope of this paper. This high level introduction is by no means complete and provides relevant standards and sources for further details.

Cubesat Space Protocol

The Cubesat Space Protocol (CSP) is a simplistic network and transport protocol, which was initially developed by Aalborg University in 2008 and is currently maintained by Aalborg students and GomSpace A/S. It is specifically designed for embedded systems such as the 32bit AVR Atmel microprocessor used by CubeL [6]. The protocol introduces a 32 bit header (see Figure 5) which contains source and destination addresses and basic means for authentication (HMAC), encryption (XTEA), UDP/RDP-like connections, and checksums (CRC). Since it is a very specialized and slim protocol well suited for networking and routing within static, small, low bandwidth and low latency systems, other features require additional protocols, header extensions or packet trailers. The CSP protocol library is open source and published on GitHub under LGPL 2.1 license [7, 8].

Bit offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Priority		Source					Destination					Destination Port						Source Port						Reserved				HMAC	XTEA	RDP	CRC
32	Data (0 – 65,535 bytes)																															

Figure 5: Cubesat Space Protocol header as used for CubeL

CCSDS and ECSS Protocol Stack for Uplink

For S-band communications, application of the following protocols standards and conventions are required. They are presented in a bottom up approach whilst omitting the physical layer channel convolution mechanisms. Due to the critical nature of telecommands, every layer contains some form of checksum to ensure the integrity of contained information.

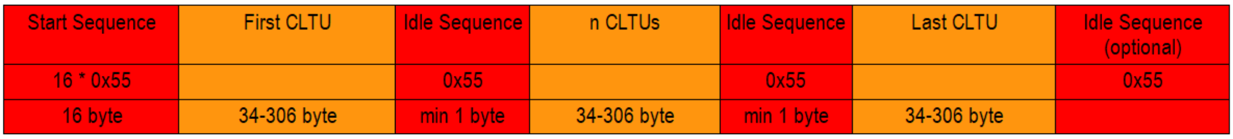
CCSDS 231.0-B-3 [9] introduces the communications link transmission unit (CLTU) as basic data structure for telecommands as seen in figures 7 and 8. An idle sequence of alternating ones and zeroes shall be used as separator between consecutive CLTUs or in case of no CLTU is ready to be transmitted (see Figure 6) aiming to improve the channel synchronization quality. Parity bits within the CLTU ensure correct transmission of contained instructions.

The TC space data link protocol CCSDS 232.0-B-3 [10] defines the transfer frame, containing information about addressed spacecraft and virtual channel (see Figure 9). It offers mechanisms for reliable delivery of TCs in the form optional of sequence control per virtual channel, which may be disabled under special circumstances such as recovery operations. Additionally this standard introduces the communication operation procedure (COP) management service also referred to as COP-1, which ensures that frames are received sequentially by reporting back the current transfer frame acceptance status via the Communication Link Control Words (CLCW) as part of telemetry transfer frames (see figures 14 and 15). This mechanism can be used to issue automatic retransmissions. CCSDS 232.1-B-2 discusses COP-1 in more detail [11]. Within the frame’s data field every TC must be contained within a TC segment as seen in Figure 10, which indicates if the carried instruction is split across consecutive frames.

Following the CCSDS 133.0-B-1 standard, the TC source packet contains the application process identifier (APID) for which the telecommand is destined [12]. It features a sequence counter which keeps track of the sent TCs per APID and states the length of the packet data field.

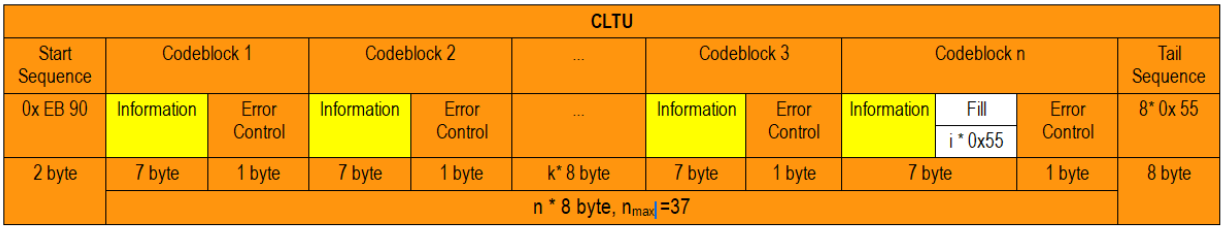
The source packet's data field begins with a secondary header (see Figure 12) according to ECSS-E-ST-70-41C packet utilization standard (PUS) which defines a set of commonly used services and subservices, and foresees ranges to map custom application capabilities [13]. It also states which TC acknowledges have been requested by the sender, which in turn will be transmitted in the form of PUS service 1 telemetry packets.

The contents of the source data field are encoded as documented by the MIB, which is defined in the SCOS-2000 Database Import ICD S2K-MCS-ICD-0001-TOS-GIC [14].



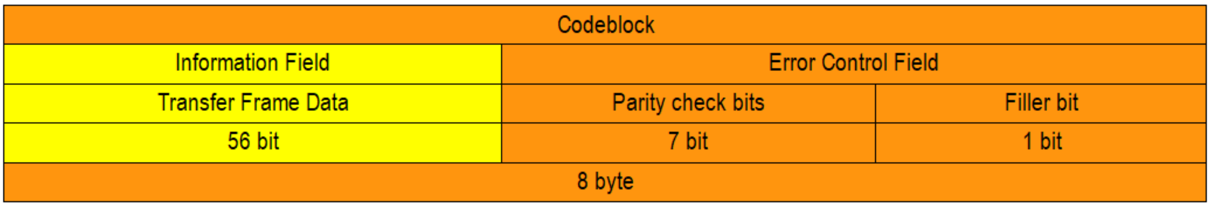
Start Sequence	First CLTU	Idle Sequence	n CLTUs	Idle Sequence	Last CLTU	Idle Sequence (optional)
16 * 0x55		0x55		0x55		0x55
16 byte	34-306 byte	min 1 byte	34-306 byte	min 1 byte	34-306 byte	

Figure 6: Physical layer



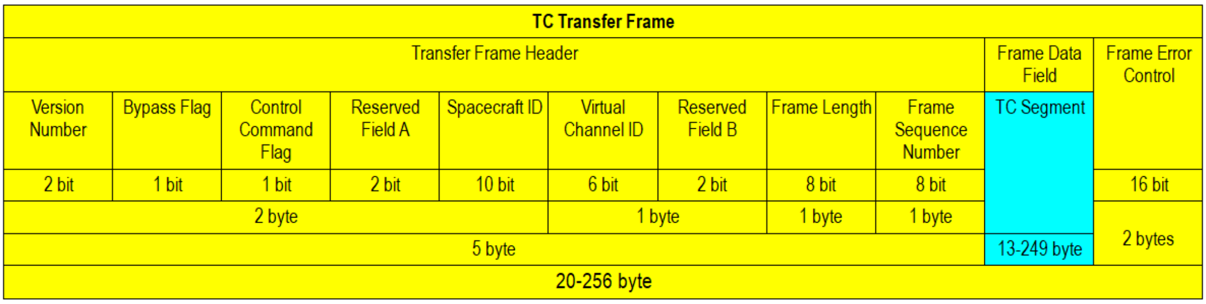
<table>
<tr><th colspan="12">CLTU</th></tr>
<tr><td>Start Sequence</td><td colspan="2">Codeblock 1</td><td colspan="2">Codeblock 2</td><td>...</td><td colspan="2">Codeblock 3</td><td colspan="3">Codeblock n</td><td>Tail Sequence</td></tr>
<tr><td rowspan="2">0x EB 90</td><td rowspan="2">Information</td><td rowspan="2">Error Control</td><td rowspan="2">Information</td><td rowspan="2">Error Control</td><td rowspan="2">...</td><td rowspan="2">Information</td><td rowspan="2">Error Control</td><td rowspan="2">Information</td><td>Fill</td><td rowspan="2">Error Control</td><td rowspan="2">8* 0x 55</td></tr>
<tr><td>i * 0x55</td></tr>
<tr><td rowspan="2">2 byte</td><td>7 byte</td><td>1 byte</td><td>7 byte</td><td>1 byte</td><td>k* 8 byte</td><td>7 byte</td><td>1 byte</td><td colspan="2">7 byte</td><td>1 byte</td><td rowspan="2">8 byte</td></tr>
<tr><td colspan="10">n * 8 byte, n_{max} =37</td></tr>
</table>

Figure 7: Coding layer: CCSDS CLTU



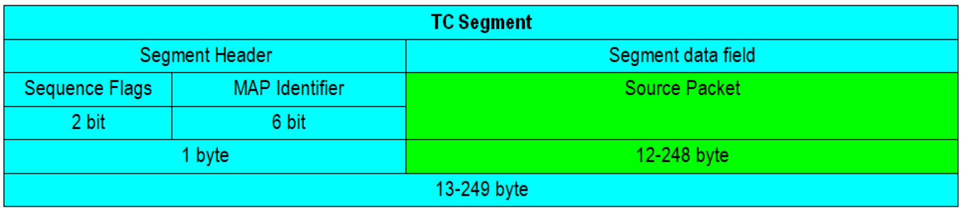
<table>
<tr><th colspan="3">Codeblock</th></tr>
<tr><td>Information Field</td><td colspan="2">Error Control Field</td></tr>
<tr><td>Transfer Frame Data</td><td>Parity check bits</td><td>Filler bit</td></tr>
<tr><td>56 bit</td><td>7 bit</td><td>1 bit</td></tr>
<tr><td colspan="3">8 byte</td></tr>
</table>

Figure 8: Coding layer (cont.): CCSDS CLTU codeblock



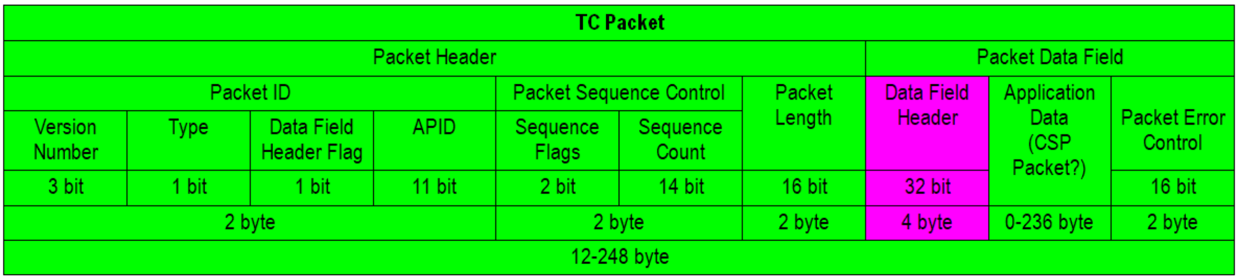
<table>
<tr><th colspan="11">TC Transfer Frame</th></tr>
<tr><td colspan="9">Transfer Frame Header</td><td>Frame Data Field</td><td rowspan="2">Frame Error Control</td></tr>
<tr><td>Version Number</td><td>Bypass Flag</td><td>Control Command Flag</td><td>Reserved Field A</td><td>Spacecraft ID</td><td>Virtual Channel ID</td><td>Reserved Field B</td><td>Frame Length</td><td>Frame Sequence Number</td><td rowspan="3">TC Segment</td></tr>
<tr><td>2 bit</td><td>1 bit</td><td>1 bit</td><td>2 bit</td><td>10 bit</td><td>6 bit</td><td>2 bit</td><td>8 bit</td><td>8 bit</td><td>16 bit</td></tr>
<tr><td colspan="5">2 byte</td><td colspan="2">1 byte</td><td>1 byte</td><td>1 byte</td><td rowspan="2">2 bytes</td></tr>
<tr><td colspan="9">5 byte</td><td>13-249 byte</td></tr>
<tr><td colspan="11">20-256 byte</td></tr>
</table>

Figure 9: Frame layer: CCSDS TC transfer frame



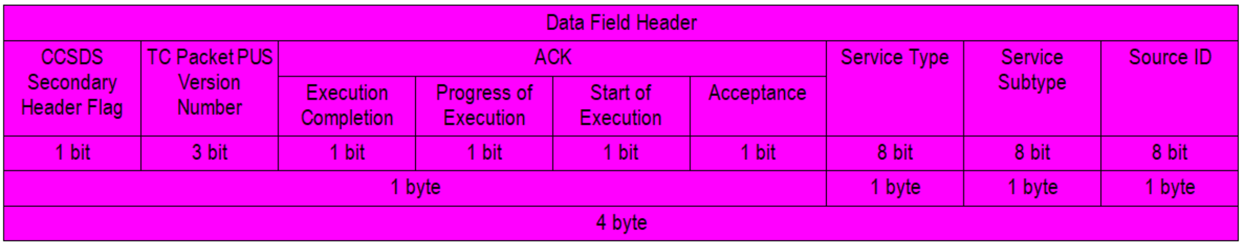
TC Segment		
Segment Header		Segment data field
Sequence Flags	MAP Identifier	Source Packet
2 bit	6 bit	
1 byte		12-248 byte
13-249 byte		

Figure 10: Segment layer: CCSDS TC segment



TC Packet									
Packet Header							Packet Data Field		
Packet ID				Packet Sequence Control		Packet Length	Data Field Header	Application Data (CSP Packet?)	Packet Error Control
Version Number	Type	Data Field Header Flag	APID	Sequence Flags	Sequence Count				
3 bit	1 bit	1 bit	11 bit	2 bit	14 bit	16 bit	32 bit		16 bit
2 byte				2 byte		2 byte	4 byte	0-236 byte	2 byte
12-248 byte									

Figure 11: Packet layer: CCSDS TC packet



Data Field Header								
CCSDS Secondary Header Flag	TC Packet PUS Version Number	ACK				Service Type	Service Subtype	Source ID
		Execution Completion	Progress of Execution	Start of Execution	Acceptance			
1 bit	3 bit	1 bit	1 bit	1 bit	1 bit	8 bit	8 bit	8 bit
1 byte						1 byte	1 byte	1 byte
4 byte								

Figure 12: Packet layer (cont.): ECSS PUS TC packet

CCSDS and ECSS Protocol Stack for Downlink

Similar to previous section the protocols involved in downlink communications are presented in a bottom up approach omitting the physical layer channel convolution.

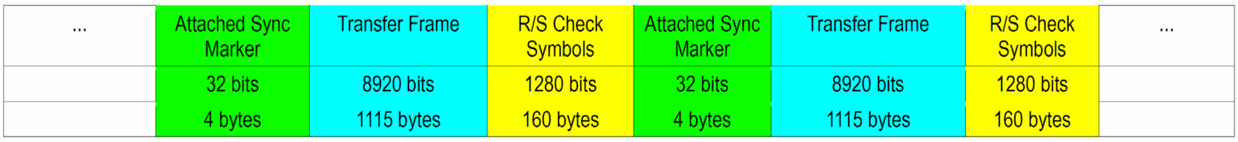
The basic data element within the utilized CCSDS telemetry protocol stack is the channel access data unit (CADU) according to CCSDS 131.0-B-3 [15]. Figure 13 shows two CADUs, each consisting of an attached sync marker, a transfer frame and optional Reed Solomon error correcting checksum (ECC). While the length of transfer frames is variable, it is commonly fixed. In essence markers and checksums shall ensure the correct transmission of transfer frames.

CCSDS 132.0-B-2 describes the TM space data link protocol, which defines the structure for TM transfer frames (see Figure 14) [16]. The frame header identifies spacecraft and data channel, before introducing the master channel frame counter (MCFC) and virtual channel frame counter (VCFC). If desired, data can be logically separated into individual channels, i.e. for time sensitive realtime bus data or recorded payload data. For every sent frame, the MCFC is incremented whereas the VCFC is only incremented within each defined channel. These two counters are commonly used to detect data gaps. The header then may indicate a secondary header (here ECSS PUS) and concludes with controls for source packet segmentation. The frame trailer starts with the command and control field (CLCW) which in essence contains information about received telecommands and serves as a feedback loop (see Figure 15). The trailing frame checksum is often optional if ECC mechanisms are used on a higher level.

On application layer the CCSDS 133.0-B-1 Space Packet Protocol introduces the TM source packet (see Figure 16) [12]. Commonly each subsystem or subcomponent is assigned a unique application identifier (APID) and data gaps on may be detected using the APID specific source sequence counter (SSC). The packet header further states the packet length to be expected in the packet data field.

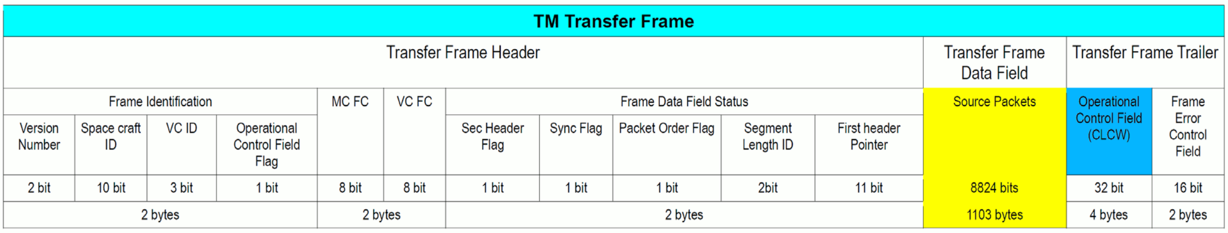
The TM source packet data field begins with the secondary header according to the ECSS-E-ST-70-41C packet utilization standard (PUS) [13]. The standard defines many packet types and subtypes, which correspond to certain aspects and capabilities of each application process. PUS also reserved a range for custom service types, which cannot be mapped to a standard service, allowing for much flexibility within the onboard application software implementation. The PUS header concludes with a timestamp, which is commonly used as sample time for all parameters to follow.

The structure for all packets and the parameters contained within the source data field is described by the MIB, which is structured according to the SCOS-2000 Database Import ICD S2K-MCS-ICD-0001-TOS-GIC [14] and commonly tailored for each SCOS-2000 MCS implementation.



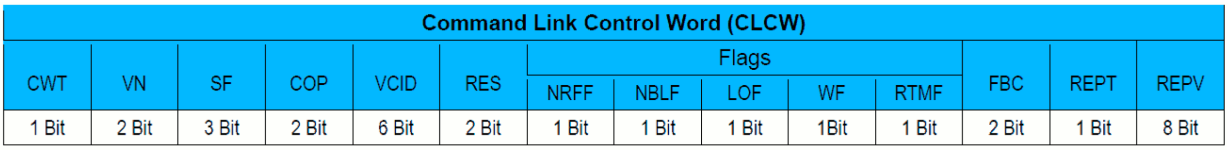
...	Attached Sync Marker	Transfer Frame	R/S Check Symbols	Attached Sync Marker	Transfer Frame	R/S Check Symbols	...
	32 bits	8920 bits	1280 bits	32 bits	8920 bits	1280 bits	
	4 bytes	1115 bytes	160 bytes	4 bytes	1115 bytes	160 bytes	

Figure 13: CCSDS channel access data unit



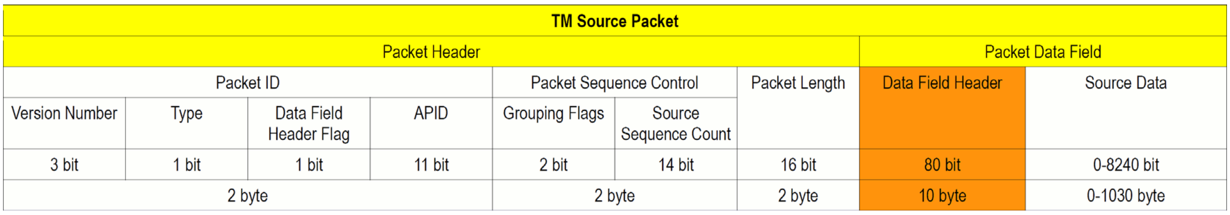
TM Transfer Frame													
Transfer Frame Header											Transfer Frame Data Field	Transfer Frame Trailer	
Frame Identification				MC FC	VC FC	Frame Data Field Status					Source Packets	Operational Control Field (CLCW)	Frame Error Control Field
Version Number	Space craft ID	VC ID	Operational Control Field Flag			Sec Header Flag	Sync Flag	Packet Order Flag	Segment Length ID	First header Pointer			
2 bit	10 bit	3 bit	1 bit	8 bit	8 bit	1 bit	1 bit	1 bit	2bit	11 bit	8824 bits	32 bit	16 bit
2 bytes				2 bytes		2 bytes					1103 bytes	4 bytes	2 bytes

Figure 14: CCSDS telemetry transfer frame



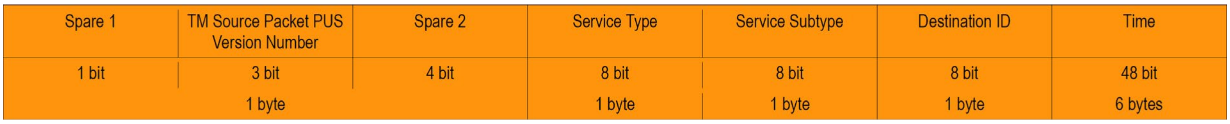
Command Link Control Word (CLCW)													
CWT	VN	SF	COP	VCID	RES	Flags					FBC	REPT	REPV
						NRFF	NBLF	LOF	WF	RTMF			
1 Bit	2 Bit	3 Bit	2 Bit	6 Bit	2 Bit	1 Bit	1 Bit	1 Bit	1Bit	1 Bit	2 Bit	1 Bit	8 Bit

Figure 15: CCSDS telemetry transfer frame CLCW



TM Source Packet								
Packet Header							Packet Data Field	
Packet ID				Packet Sequence Control		Packet Length	Data Field Header	Source Data
Version Number	Type	Data Field Header Flag	APID	Grouping Flags	Source Sequence Count			
3 bit	1 bit	1 bit	11 bit	2 bit	14 bit	16 bit	80 bit	0-8240 bit
2 byte				2 byte		2 byte	10 byte	0-1030 byte

Figure 16: CCSDS telemetry source packet



Spare 1	TM Source Packet PUS Version Number	Spare 2	Service Type	Service Subtype	Destination ID	Time
1 bit	3 bit	4 bit	8 bit	8 bit	8 bit	48 bit
1 byte			1 byte	1 byte	1 byte	6 bytes

Figure 17: CCSDS TM source packet secondary header following the ECSS packet utilization standard

TAILORING CCSDS

The previously introduced standards and conventions are rarely implemented to the fullest extent, as many features might not be useful for a particular mission. For this purpose a technical note was created to define the tailoring of used protocols. For CubeL it is important to keep the adapter implementation as simple as possible, thus many components have been excluded and fixed with static values. In the following subsections modifications to these modifications and their effects are discussed.

Uplink

To be compliant with SLE, it is sufficient, to implement only physical and coding layer (see Figure 6, 7, and 8). Of the CLTUs only the Start Sequence (0xEB 90) and the Tail Sequence (8 * 0x55) have to be present, the inner structure is not strictly necessary for SLE usage. To be compliant with the GSOC MUM MCS, the following TC layers have to be implemented as well.

The TC transfer frame (see Figure 9) can be set to sequence controlled data frames with frame sequence number 0x00 if the COP-1 protocol is not used. One virtual channel with fixed ID 0 is sufficient as the available ground station bandwidths are sufficient for the expected amount of data. The frame length must be set, however a fixed value is acceptable if only one size of TC packets is generated.

On segment layer, source packets shall neither be segmented or grouped. One segment shall contain one source packet (see Figure 10).

For simplicity it is sufficient to have one fixed application process ID, which allows for the sequence count to be just a single counter. Packet length may be fixed if only one size of TC packets is used.

The source packet secondary header shall indicate the usage of ECSS PUS, request no acknowledge report and have fixed PUS service type and subtype designators within the custom service range.

Downlink

For downlink, the TM transfer frame virtual channel identifier can be set to 0 as long as the data rate does not exceed 32kb/s. VC7 shall be used for idle frames if necessary. Further flags shall indicate the presence of the operational control field within the frame trailer, and the absence of the secondary header. Source packets in transfer frames shall always be inserted synchronously and in forward order. Further, there shall be no segmentation of source packets.

The CLCW can be set to all zeroes except if COP-1 is in effect. COP-1 is typically used at GSOC to simplify ground operations, but it is not strictly required. This would require the implementation of the report value (REPV). Further the implementation of the no RF available flag (NRFF) and no bit log flag (NBLF) is desired.

Within a TM source packet the presence of a data field header shall be indicated. Similar to uplink, the APID may be fixed to one value, allowing the source sequence count to be a single counter. Further every packet shall be standalone and not grouped. If only one packet size is used, the packet length can be fixed.

The ECSS PUS version 1 header shall indicate a fixed service type and subtype within the custom range. Finally, the provision of a timestamp in CUC format using 4 byte coarse time, 2 byte fine time, and GPS epoch without leap seconds is mandatory for CubeL, which is used for the sample time of contained telemetry parameters.

CSP TO CCSDS ADAPTER

The general idea is to create a simple CCSDS wrapper to include a CSP packet, which in turn carries payload data either foreseen by the protocol specification or proprietary content by GomSpace. Due to a non-disclosure agreement between DLR and GomSpace, protocol details cannot be discussed, thus a CSP conform ping packet [8] is used within the scope of this work. This wrapper shall be applied to both up- and downlink form a tunnel.

Onboard custom software created by GomSpace, which is running on the S-band SDR unpacks CCSDS packets and forwards the raw CSP packets to the bus. In reverse direction CSP packets will be wrapped according to above specification and sent to ground.

On ground, using the information from available CSP documentation in combination with some reverse engineering efforts, a MIB was crafted through which GECCOS is capable of formulating entire CSP TC packets including payload data without requiring any GECCOS software modifications. This way most functionality which does not require immediate feedback loops can be reproduced.

The resulting MIB is also used by other GSOC MUM tools to further process and visualize TMTC data. Throughout the preparation and setup of the phase E2a ground segment and development of operational procedures all telemetry parameters have been documented comprehensively allowing for an auto generated MIB baseline, which already includes TM parameters, allowing further development efforts to be focused on telecommands.

PROTOTYPE VALIDATION

For early conceptual validation a minimal prototype MIB was created consisting of a simple ping CSP telecommand, which shall be validated against the CubeL engineering model (EM).

To stay true to the CubeL low cost approach, the EM only features components relevant for O4C development. While lacking many subsystems such as ADCS, camera, battery modules, solar panels, and RF hardware it is sufficient for familiarization with the phase E2a commanding interface and operational flight procedure development for OBC, O4C and some EPS related commanding. As can be seen in Figure 18 the RF link is substituted by a wired CAN bus connection.

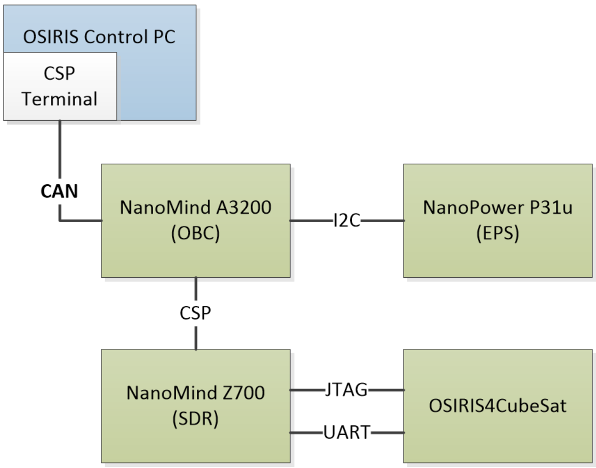


Figure 18: Reduced CubeL engineering model

Uplink

Due to situational project limitations, GECCOS and the EM setup could not be connected directly, thus a different approach was selected to record and playback the created data streams as described below.

The MIB was imported into GECCOS, through which a ping command was able to be loaded onto the command stack. Prior to sending, the GECCOS TC stream recording was activated. Since GECCOS usually interfaces with SLE service providers, the output stream includes an addition layer consisting of the Network Controller and Telemetry Router System (NCTRS) protocol. After stripping the NCTRS layer from the recording, raw CLTU data was exposed. The resulting file was then transferred to the control computer, which is connected to the EM. The individual CLTUs were then sent to the CubeL OBC via CAN. The successful reception and execution of sent commands was observed via enabled OBC debug utilities.

Downlink

As of writing this paper the onboard S-band software and corresponding documentation are still in development. Thus, the GECCOS telemetry reception will be tested once available. However using recorded onboard housekeeping telemetry files and the custom CubeL Offline Telemetry Processing System (COPS), which was developed for enabling the usage of Satmon during phase E2a, subsequent GSOC MUM tools for archiving and telemetry visualization were successfully validated.

CONCLUSION

In the course of this work relevant CSP, CCSDS and ECSS protocols, standards and conventions were briefly introduced. For integration of the CSP based CubeL cubesat into the GSOC MUM environment a CSP to CCSDS adapter was proposed, a prototype of which was implemented for concept validation against the CubeL EM. TCs sent using the GECCOS MCS were received and executed correctly by the CubeL OBC. The nature of the test setup limited the supported set of telecommands to stateless communication between MCS and EM. CSP services requiring closed feedback loops to continue an operation in progress, such as file transfers remain a challenge and might require software alterations.

Overall the chosen approach to quickly integrate a foreign protocol into the GSOC MUM environment is possible, however for small CSP packets to be individually wrapped as fixed length CCSDS packets the introduced protocol overhead is significant. This can only be compensated by a higher bandwidth connection, which will be the case for CubeL when transitioning from UHF- to S-band communication. Further it is to note that extensive documentation of TM and TC encodings should be a prerequisite for any protocol integration intending to be cost effective.

The flexibility of the GSOC MUM infrastructure and offered interfaces allows this seemingly crude but effective approach to be implemented rapidly within a few weeks. However if time and money permit a proper CCSDS protocol stack implementation within the space segment should be the preferred option for CubeL S-band communications. While not being necessarily perfect for cubesat applications, long established institutionalized standards have a lot of heritage and have matured over the years through regular revisions by globally collaborating expert working groups. Within the cubesat industry a widely adopted common communication standard has yet to be established. In the meantime GSOC is looking forward to increasing support for CCSDS compliant commercial communication solutions for cubesats.

References

1. DLR German Aerospace Center, About DLR, DLR Magazine Issue 158, p.56, July 2018
2. C. Schmidt, The OSIRIS Program at DLR, Free-Space Laser Communication and Atmospheric Propagation XXX. Vol. 10524. International Society for Optics and Photonics, February 15th, 2018.
3. Tesat-Spacecom GmbH & Co KG, CubeLCT, https://www.tesat.de/en/portfolio/products/690-cubelct
4. GomSpace, Nanosatellite Platform for German Aerospace Center's Demonstration of the OSIRIS Optical Demonstration Payload, https://gomspace.com/dlr.aspx
5. Stangl, Christian and Lotko, Boguslav and Geyer, Michael and Oswald, Michael and Braun, Armin (2014) *GECCOS – the new Monitoring and Control System at DLR-GSOC for Space Operations, based on SCOS-2000.* SpaceOps 2014 - 13th International Conference on Space Operations, 05.-09. May 2014, Pasadena, California, USA.
6. GomSpace A/S, Datasheet NanoMind A3200, December 10th, 2019, https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanomind-a3200_1006901.pdf
7. Wikipedia, Cubesat Sapce Protocol, https://en.wikipedia.org/wiki/Cubesat_Space_Protocol
8. GitHub, The Cubesat Space protocol, https://github.com/libcsp/libcsp
9. CCSDS, TC Synchronization and Channel Coding, CCSDS 231.0-B-3, September 2017
10. CCSDS, TC Space Data Link Protocol, CCSDS 232.0-B-3, September 2015
11. CCSDS, Communications Operation Procedure-1, CCSDS 232.1-B-2, September 2010
12. CCSDS, Space Packet Protocol, CCSDS 133.0-B-1, September 2003
13. ECSS, Telemetry and Telecommand Packet Utilization, ECSS-E-ST-70-41C, April 15th 2016
14. ESA, SCOS-2000 Database Import ICD, S2K-MCS-ICD-0001-TOS-GIC, Issue 5.2.5, November 20th, 2013
15. CCSDS, TM Synchronization and Channel Coding, CCSDS 131.0-B-3, September 2017
16. CCSDS, TM Space Data Link Protocol, CCSDS 132.0-B-2, September 2015