Proceedings of the Estonian Academy of Sciences,
2014, **63**, 2S, 222–231
doi: 10.3176/proc.2014.2S.03
Available online at www.eap.ee/proceedings

Design of the fault tolerant command and data handling subsystem for ESTCube-1

Kaspars Laizans[a,b*], Indrek Sünter[a,b], Karlis Zalite[a,b], Henri Kuuste[a,b], Martin Valgur[a,b], Karl Tarbe[b], Viljo Allik[a], Georgi Olentšenko[b], Priit Laes[b], Silver Lätt[a,b], and Mart Noorma[a,b]

[a] Tartu Observatory, Observatooriumi 1, 61602 Tõravere, Tartumaa, Estonia
[b] Institute of Physics, Faculty of Science and Technology, University of Tartu, Tähe 4-111, 51010 Tartu, Estonia

Received 15 August 2013, revised 26 February 2014, accepted 7 March 2014, available online 23 May 2014

Abstract. This paper presents the design, implementation, and pre-launch test results of the Command and Data Handling Subsystem (CDHS) for ESTCube-1. ESTCube-1 is a one-unit CubeSat, which will perform an electric solar wind sail experiment. The development process of the CDHS for ESTCube-1 was focused on robustness and fault tolerance. A combination of hot and cold hardware redundancy was implemented. Software, including a custom-written internal communications protocol, was designed to increase the system's fault tolerance further by providing fault detection and fall-back procedures. Tests were carried out to validate the implementation's performance and physical endurance. The final CDHS design is operational within the set requirements. Tests that verify fault tolerance of the system in orbit are suggested.

Key words: command and data handling subsystem, fault tolerance, redundancy, nanosatellite, CubeSat, ESTCube-1.

Acronyms and abbreviations
ADC – Analogue-to-Digital Converter
ADCS – Attitude Determination and Control Subsystem
ARQ – Automatic Repeat-Request
CAM – Camera subsystem
CAN – Controller Area Network
CDHS – Command and Data Handling Subsystem
COM – Communications subsystem
EPS – Electrical Power Subsystem
E-sail – Electric solar wind sail
FET – Field-Effect Transistor
FRAM – Ferroelectric Random Access Memory
HDLC – High-level Data Link Control
I^2C – Inter-Integrated Circuit
I/O – Input/Output
ICP – Internal Communications Protocol
MCU – Microcontroller Unit
MEMS – Microelectromechanical System
NOR – Negative OR logic function
OSI – Open Systems Interconnection
PCB – Printed Circuit Board
RAM – Random Access Memory
RTC – Real-Time Clock
SPI – Serial Peripheral Interface
SS – Sun Sensor
UART – Universal Asynchronous Receiver/Transmitter
USART – Universal Synchronous/Asynchronous Receiver/Transmitter
USB – Universal Serial Bus
XOR – Exclusive OR logic function

1. INTRODUCTION

Since the introduction of the CubeSat standard [1] small satellite development has seen rapid growth encompassing an increasing number of nations [2]. Advances in miniaturized integrated circuit technologies have improved environmental tolerances, stability, and functionality while decreasing the size and reducing power consumption of satellites. Such satellites as RAX [3], BRITE [4], SwissCube [5], CanX-2 [6], Delfi-C3 [7], AAUSAT-II [8], AAUSAT-3 [9], and COMPASS-1 [10] have demonstrated the versatility of CubeSats for fulfilling a variety of missions. With the increasing

* Corresponding author, kaspars.laizans@estcube.eu

scientific importance of CubeSat missions fault tolerance is now becoming critical. However, while the topic has been covered for larger spacecraft since the beginning of the space age [11], redundancy and fault tolerance of the Command and Data Handling Subsystem (CDHS) on board micro- and nanosatellites are yet to be discussed.

The fundamentals of the fault tolerant CDHS for CubeSats can be drawn largely from several decades of experience acquired by designing and operating large spacecraft. As discussed by Rennels [11], availability of commercial off-the-shelf single-chip microprocessors allows for implementing several processors for redundancy purposes. Furthermore, it allows for a distributed system design with fault recovery algorithms [12]. This general approach is also presented by McLoughlin et al. [13], who propose a parallel architecture that enables commercial devices to be incorporated into a computational unit with the total reliability approaching that of space-qualified alternatives. Nevertheless, little attempt has been made to present a CDHS design that incorporates the aforementioned ideas, advances in microelectronics, and CubeSat design specifications. On Delfi-n3Xt satellite, basic on-board computer functions were to be implemented on a microcontroller at the primary radio transceiver [14]. CanX-1 had error detection and correction on external memories and an implemented boot-strap code for carrying out basic spacecraft operations in the case of failure [6]. AAUSAT-3 avoided single points of failure by adopting a decentralized design philosophy where each subsystem has enough processing power and data storage to carry out its tasks [15].

In this paper we present a CDHS design with an improved fault tolerance scheme for ESTCube-1 [16], a one-unit CubeSat, which will perform an electric solar wind sail (E-sail) [17] experiment. Great emphasis during the ESTCube-1 project was put on fault tolerance, which resulted in the development of robust subsystems, including a CDHS. Here we describe the requirements that were derived from both the general CubeSat specifications and the specific scientific mission. We propose a combination of hot and cold hardware duplication redundancy methods as a possible solution for improving CubeSat fault tolerance. Software solutions ensure further reduction of risks by detecting failures and executing fall-back procedures. Finally, results from pre-flight tests are presented and discussed.

2. REQUIREMENTS

2.1. Satellite design requirements

From a mission-specific perspective, the CDHS on board ESTCube-1 is required to handle data needed for controlling the satellite and for the E-sail experiment [16]. The experiment involves a centrifugal unreeling of a 10 m long conductive tether. Next, a ± 500 volt potential is generated on the tether to observe its interaction with ionospheric plasma. The observation will be carried out indirectly by estimating changes in the satellite spin rate. Estimation is performed using a Kalman filter with readings from magnetometers, Microelectromechanical Systems (MEMS), angular rate sensors, and analogue sun sensors. The implementation of the Kalman filter on ESTCube-1 requires the input data measurements to be carried out 2.5 times per second [18].

Furthermore, the CDHS has to (1) store data from the Camera subsystem (CAM) in the form of multiple binary images of up to 600 KB each; (2) store housekeeping data of all subsystems; and (3) compile the beacon and data packets for downlink.

For redundancy purposes direct connections from all subsystems to the CDHS are preferred to sequential or daisy-chain type. As a backup method, direct connections between the subsystems bypassing the CDHS have to be implemented. Thus, a non-functioning subsystem cannot hinder the operation of any other node. This can be used in case the CDHS is rendered unusable or as a redundant communication channel if the primary data bus is damaged.

According to Bouwmeester and Guo [19], an Inter-Integrated Circuit (I^2C) communication bus has been favoured for the design of small satellites. However, a bus-busy state can occur while using the I^2C for inter-subsystem communications. If one or more subsystems are powered down during the communications, a mismatch in open-drain/collector configurations of devices can occur, which can lead to the bus-busy state blocking the whole bus. Other bus types used on small satellites are Universal Serial Bus (USB) and Controller Area Network (CAN) bus. The USB consumes much power, while the CAN bus is not supported by many devices. For these reasons the USB and CAN were not selected for ESTCube-1.

Due to physical limitations of the satellite, with a worst-case power production of 2 W, a 180 mW average and 250 mW peak power consumption constraint for the CDHS was set by the Electrical Power Subsystem (EPS) [20].

To ensure lower power consumption, the use of 3.3 V supply for the whole system was adopted at the cost of higher susceptibility to noise. The general trend of commercial-off-the-shelf devices is to reduce voltages even further with higher supply voltage devices becoming obsolete. To diminish the influence of digital noise on the signals, the use of analogue signals has been discouraged.

Furthermore, physical requirements were introduced primarily by the CubeSat standard for outer satellite dimensions of 10 cm $\times$ 10 cm $\times$ 10 cm with additional constraints for internal satellite structure derived from those (Table 1). Mass and environmental parameter estimates were set in early Phase B of the satellite development, based on the preliminary design. The vibration tolerance requirements were derived from launcher-specific documentation.

Table 1. Constraints for the internal structure of ESTCube-1

Requirement parameter	Maximum value
Dimensions	94 mm × 92 mm × 18 mm
Mass	60 g
Operating temperature	–10 to 70 °C
Vibration tolerance	
sine	22.5 g
PSD random	18 g
shock	1410 g

2.2. Software requirements

Logging housekeeping and mission data, boot-loader events, warnings, and errors has been set as the main task of the CDHS. Part of the mission would be carried out above the Earth's poles with no direct communications available. Thus, the command scheduler must be able to execute time-specific commands unsupervised.

The on-board software of the satellite is not required to be autonomous. All operations that involve a risk must be started manually or it must be possible for the satellite operator to stop these operations and disable them permanently.

The ability to upgrade the CDHS firmware via an access port device at the launch site as well as in orbit is needed. In-flight upgrades also provide flexibility regarding the satellite life-time schedule and functionality, which enables implementation of unplanned tasks in orbit. Such an approach allows for additional experiments with different attitude control algorithms and varying manners of detection of radiation effects on electronic devices.

In order to improve the fault tolerance of the system, it must be possible to enable, disable, and configure CDHS software components. Configurability also provides some flexibility without the need to upgrade the firmware.

3. SATELLITE DESIGN AND FAULT TOLERANCE IMPLEMENTATION

3.1. Fault tolerance

One of the main classic methods of increasing hardware fault tolerance is having hardware redundancy or multiplication of essential parts and components rather than modular result weighing. Having one of the duplicated or triplicated components damaged might lead to different results, depending on the general system architecture: either the system switches and starts to use another component (cold redundancy) or it experiences loss of performance while maintaining functionality (graceful degradation of the hot redundant system) [21]. A blend of a mixed hot–cold redundant system with software fault-check routines is used to create a system with enhanced fault tolerance.

3.2. Microcontrollers

Two STM32F103 Microcontroller Units (MCUs) are used in cold redundancy mode and switched by the EPS. Switching on one of the MCU's power buses powers up the MCU itself and drives the logic to enable the corresponding peripheral lines through the bus switches (Fig. 1). All the devices are available for both MCUs as hot redundant, enabling access to them by the other MCU as a backup processor in the case the main one is damaged.

3.3. Data buses

Direct connections between all subsystems and the CDHS were designed (Fig. 2). In particular, magnetometers and 3-axis MEMS angular rate sensors were duplicated in cold redundancy (Fig. 3). Readings from six sun sensors are digitized by two Analogue-to-Digital Converters (ADCs) and sent via two separate Serial Peripheral Interface (SPI) buses. Due to the limited amount of buses available the SPI buses have to be shared between the Attitude Determination and Control Subsystem (ADCS) and the CDHS internal memories and the Real-Time Clock (RTC).

Universal Synchronous/Asynchronous Receiver/Transmitter (USART) communications were chosen for communication between subsystems, which have their own MCU because it is robust and supported by the majority of microcontrollers. Apart from the standard USART, I^2C, and SPI buses additional digital Input/Output (I/O) signals are needed for subsystem-specific control and feedback.

3.4. Switching

All buses and peripheral lines are connected to both microcontrollers via SN74CB3Q16211DL bus switches. Bus switching is required in the case two MCUs are connected directly in parallel and the inactive MCU is able to sink all of the current injected into the bus by another microcontroller. By default, both processors are isolated from all peripherals and I/O lines. When the MCU is powered up from the EPS the respective bus switch logic is enabled allowing signals to pass through. Care must be taken to avoid bus switch logic being enabled for both microcontrollers at the same time. In this case, signals would be drained by the inactive MCU. Physically, outputs from bus switches are tied together, making a single input on the peripheral.

Each MCU has its own set of bus switches, through which peripherals are connected. Unlike the typical signal division where MCUs share one bus switch, this approach helps to alleviate the risk of a single bus switch causing the CDHS to fail as well as keeps current consumption to a minimum, because only half of the bus switches are active simultaneously.

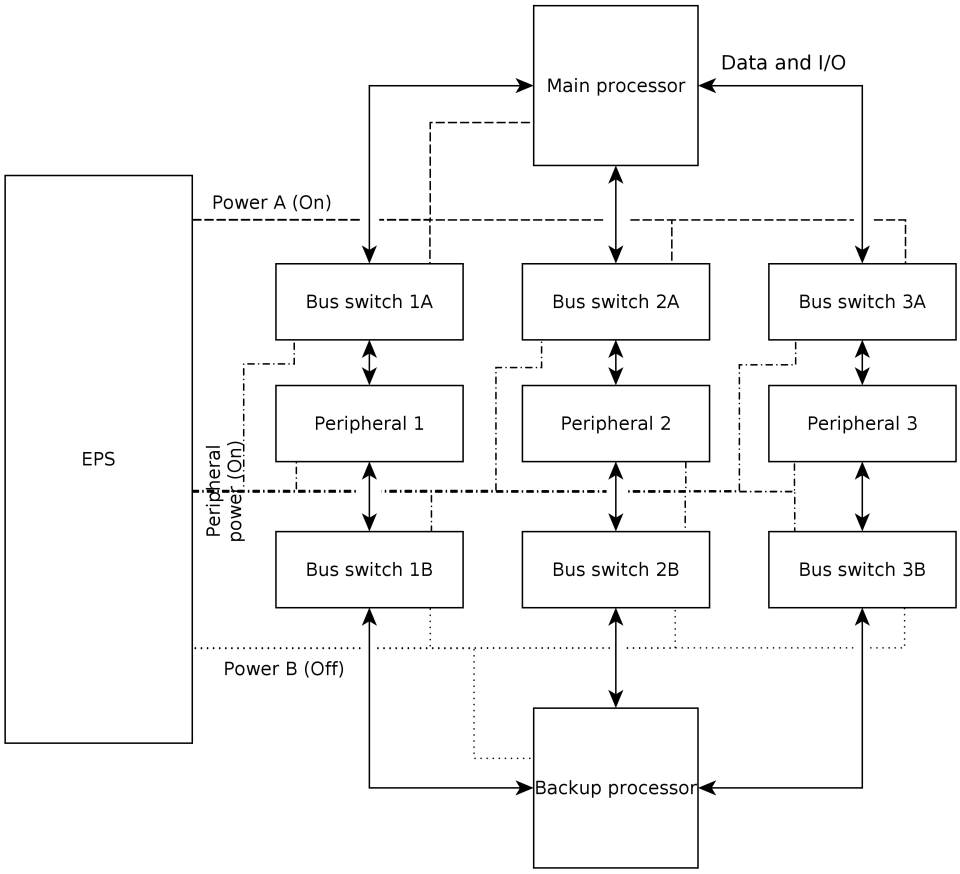

Fig. 1. Bus switch logic control via processor power buses. I/O – Input/Output, EPS – Electrical Power System.

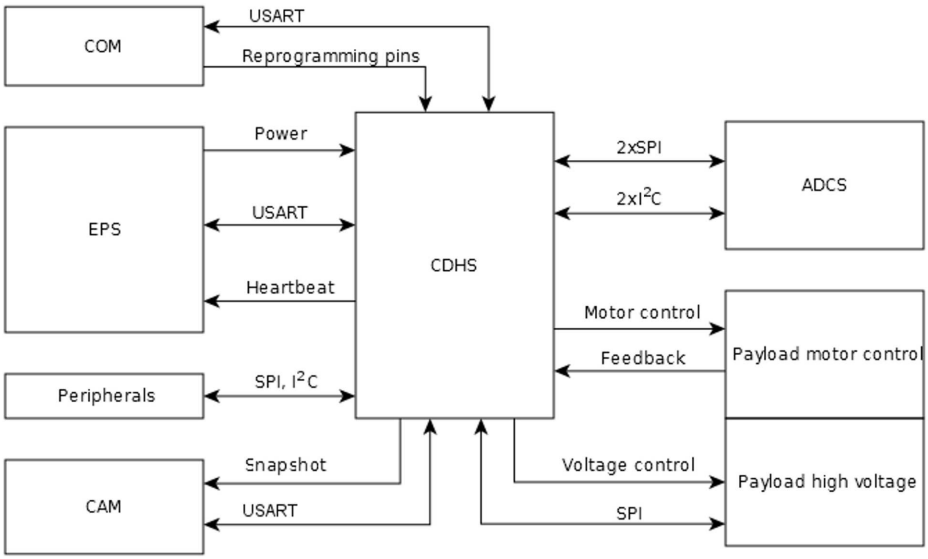

Fig. 2. Data channels of the CDHS. ADCS – Attitude Determination and Control Subsystem, COM – Communications subsystem, EPS – Electrical Power Subsystem, CAM – Camera subsystem, USART – Universal Synchronous/Asynchronous Receiver/ Transmitter, SPI – Serial Peripheral Interface, I^2C – Inter-Integrated Circuit.

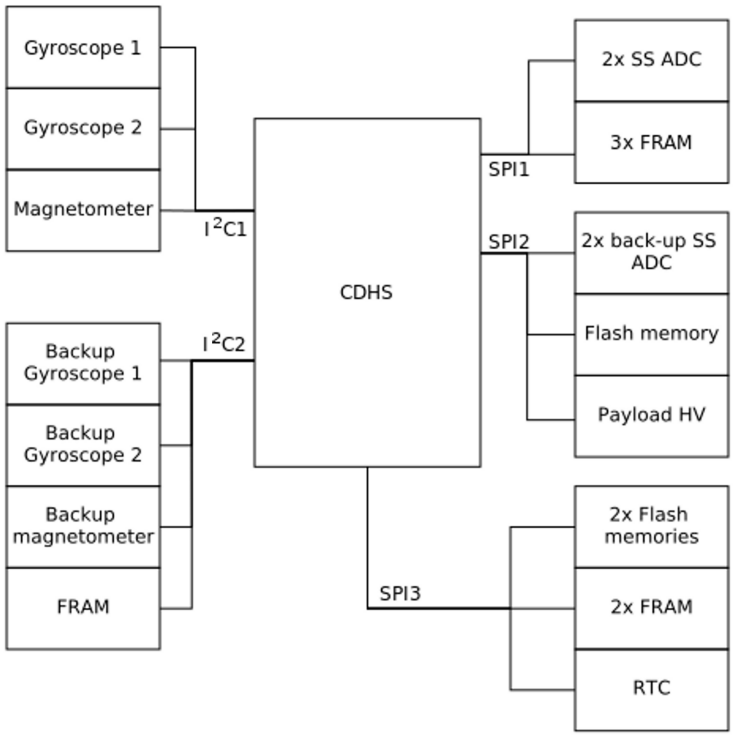

Fig. 3. Peripheral device distribution on data buses. I^2C – Inter-Integrated Circuit, SPI – Serial Peripheral Interface, FRAM – Ferroelectric Random Access Memory, ADC – Analogue-to-Digital Converter, SS – Sun Sensor, HV – High Voltage, RTC – Real-Time Clock.

According to SN74CB3Q16211DL bus switch specifications, each device can be considered as consisting of a relatively small number of Field Effect Transistors (FETs) (order of magnitude $<$100), which makes it a low-density device. A probability of such a low-density device experiencing a radiation-induced single upset event is much smaller than the same event happening within a microcontroller.

3.5. Storage

Storage of the CDHS and CAM firmware images, pictures, telemetry, and science data is one of the main tasks of the CDHS. Three fast Spansion SF25FL 128 Mbit SPI Flash memories are used for the storage of non-critical data such as logs of housekeeping and mission data, camera images, etc. Five Ramtron FM25 series ferroelectric 2 Mbit SPI Random Access Memory (RAM) modules store firmware images, ADCS constants, scheduler queues, and file system metadata for which data integrity is crucial. For redundancy purposes, memory devices are placed on all three SPI buses (Fig. 3), i.e. the subsystem internal memory bus and two SPI buses shared with the ADCS. Use of shared buses for redundancy purposes was deemed to be an acceptable risk due to the fact that the probability of high-density internal SPI peripheral breaking down is much higher than the probability of a bus becoming unavailable due to a damaged low-density secondary device [22].

Since the CDHS has external memory devices connected via serial buses, structures or variables in the external memory cannot be addressed directly. This poses an additional challenge for software development. All external memories are shared between both CDHS microcontrollers, and memory contents need to be backwards compatible with the different firmware images stored on board the CDHS. These features could be considered risks, which are minimized by extensive testing and use of safe fall-back default configurations for cases when backwards compatibility issues arise.

3.6. Real-time clock

Tether unreeling and camera image trigger events during the mission have to be scheduled, with a timing accuracy of 100 ms. A processor-independent external RTC is needed to periodically synchronize millisecond timers that are used for scheduling the CDHS events and commands. MAX DS3234 RTC, which is connected to the internal SPI bus (Fig. 3), was implemented. This device has integrated temperature-compensated crystal oscillator and voltage reference. Since the only device on the satellite directly powered by the batteries is the EPS, the start-up sequence of the CDHS had to include

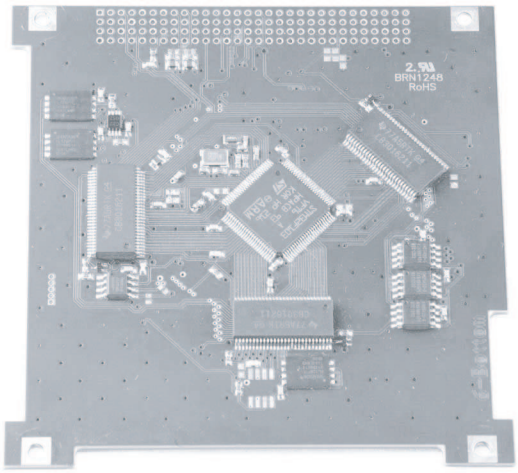

Fig. 4. Populated board, bottom side.

the polling satellite date and time from the EPS and reconfiguring its own RTC.

3.7. Printed circuit board and manufacture

The CDHS is assembled on a six-layer R1755V Medium Tg 170 Printed Circuit Board (PCB) (Fig. 4). No blind or buried vias are used to avoid their dislodging and short circuits generated by free floating via material. Board cut-outs were needed to provide space for sun-sensor cabling to the ADCS. The dimensions of the assembled board are 92 mm $\times$ 94 mm $\times$ 8.6 mm, and the mass of the populated board is 48.54 g.

4. SOFTWARE DESIGN

4.1. General design

The CDHS software has been designed to be configurable in orbit and to tolerate failure of attached devices. From the CDHS device configuration table, external ferroelectric RAM (FRAM), Flash memories, SPI RTC, and microcontroller internal peripherals can be enabled or disabled individually. In the case a device failure is detected on startup, the device is disabled automatically. A table of integer variables is used for setting processor clock frequency, fall-back modes for error logging and command scheduler, beacon transmission period, and other CDHS operational parameters. Coefficients required for attitude determination and control are stored in a table of floating-point values. Configuration variables are grouped by linker regions, which allows them to be easily synchronized with internal Flash pages, making changes non-volatile. A safe fall-back configuration is applied automatically in the case a corrupted active configuration passes CDHS limit checks and causes the CDHS to reset six times without sending a single telemetry frame to the ground.

For error logging, once the system has at least partially recovered again, there are multiple levels of storage to ensure the delivery of the list of errors. Errors during boot-up or interrupts are stored in a RAM section that retains its contents till the next CDHS power recycle. After the initialization of the external FRAM memories and file systems, the list of errors is appended to a larger error log file.

Software-induced fault tolerance by means of modular redundancy was not used due to the performance requirements and restrictions for the use of Flash and RAM. In the scope of this paper, the CDHS software is considered to be fault tolerant when the CDHS remains operable after experiencing software or hardware faults. Fault tolerance of the CDHS software does not necessarily include the ability to continue the operations that were interrupted by a fault.

4.2. Operating system and drivers

FreeRTOS was chosen as the operating system for the CDHS, mainly because of its low Flash and RAM footprints and low performance overhead. On the CDHS, FreeRTOS takes about 11 KB of Flash and 64 KB of RAM, most of which is reserved for the heap of dynamic memory management.

On top of a minimalistic hardware abstraction layer for STM32F1 and STM32F2, peripheral and device drivers were implemented for FreeRTOS. The Universal Asynchronous Receiver/Transmitter (UART), I^2C, and SPI drivers each have a daemon running in the background, which processes enqueued transactions and propagates the error codes back to the caller. For the CDHS and the CAM, exception handling was implemented in C.

4.3. File systems

In order to fulfil the storage requirements on the CDHS, two types of files are needed: (1) image files for CAM photos and calibration tables, and (2) circular journal files for housekeeping log entries, mission data, and buffering telemetry. For these types of files it is unnecessary to buffer block contents before erasure. Image files are erased and rewritten as a whole, whereas journal files are erased and overwritten block-wise, oldest entries first. Additional simplifications were made: 256 files per file system, hard or soft links, no directories, filenames are numeric, and files are of constant length. It was also decided to have only one volume per file system.

Since the CDHS has only 96 KB of static RAM, most of the available Flash file systems could not be used. The Transactional Flash File System was found to have exceptionally low requirements for RAM (less than 200 B). On the other hand, this system assumes a not OR (NOR) Flash memory with Single Level Cells to re-program individual bits of a Flash page. The

S25FL128P NOR Flash used on the CDHS is manufactured on MirrorBit technology. Furthermore, because of the limited amount of static RAM and the need for file systems for three serial NOR Flash devices with multiple bits per cell and six FRAM devices, custom Flash and RAM file systems with a low memory footprint had to be developed.

The Flash file system counts bad bytes and blocks by reading back the data written to the Flash. Mismatched bytes are recorded in the file system metadata. Single bytes are corrected on file reads. If more than ten bad bytes are detected in a single block, the block is marked as a bad block. Bad blocks are skipped on read and write attempts, resulting in data loss. The number of detected bad bytes and bad blocks shall be monitored in orbit.

As FRAM is more radiation tolerant than the Flash memory, the file system metadata and system files are stored on FRAM [23].

4.4. Internal communications protocol

A satellite-wide Internal Communications Protocol (ICP) was designed in addition to USART communications. The ICP was loosely based on the asynchronous version of the High-Level Data Link Control (HDLC), and is used by the EPS, CAM, Communications subsystem (COM) and the CDHS for inter-subsystem data and command transfer. The protocol covers both data link and network layers of the Open Systems Interconnection (OSI) model. The ICP's design was mostly constrained by the EPS: 16 MHz clock frequency, 2 KB of RAM dedicated to ICP, and no operating system.

The developed ICP inherits the HDLC's framing mechanism: packets are delimited by 0x7E and any 0x7E and 0x7D bytes occurring inside a packet are XORed with 0x20 and have 0x7D added before them. Similarly to HDLC, a 16-bit Fletcher's checksum is computed from the packet header, and the payload is included in the tail of the packet.

To ensure a reliable in-order delivery of packets, the Go-Back-N Automatic Repeat-Request (ARQ) data transmission protocol (a variant of sliding window protocol with the transmit window size of N and receive window size of 1) is used on direct links. The transmit window size is configurable, with a maximum size of 8 packets. The Go-Back-N ARQ was preferred over the simpler stop-and-wait ARQ protocol (transmit window size of 1) due to its higher throughput.

The routing mechanism of the ICP is very simple due to both the simple network topology and the tight RAM constraints. Routes from one subsystem to another are manually and statically defined as a list of data links to be attempted, ordered by preferability. After a packet sending times out and is re-sent a predefined number of times on a data link (usually three), sending via the next data link in the routing table is attempted. Since only a single loop exists in the network topology, infinite loops are avoided by refusing to route a packet through the subsystem from which the packet originated. The user is only acknowledged of a successful delivery of a packet over a data link and not of a delivery to the intended destination to keep the protocol lightweight.

4.5. Command scheduler

Commands received by the CDHS are scheduled by priority or executed immediately. The satellite's internal command structure supports for three different priorities: low, high, and immediate. For low and high priorities, separate scheduler queues are used. There is no queue for commands of immediate priority. On each scheduler update, either an immediate command is handled, or a command is fetched from either the low or high queue. It is guaranteed that high priority commands are handled twice as often as low priority commands and that no more than a single command is executed on each scheduler update.

Commands received by the CDHS are enqueued for handling in the command scheduler task. Although the satellite's internal command structure supports commands of three different priorities, this feature was rarely used and has been dropped from the command scheduler in favour of simplicity. Commands scheduling other commands are used to provide support for looped execution and conditionals.

All commands are stored with a Fletcher-16 checksum and queue entries with invalid checksum are discarded. To avoid memory corruption the whole queue is cleared in the case of invalid queue entry lengths. Clearing the queue stops active operations and prevents corrupt commands from being executed.

A date–time scheduler is used for executing commands at specified date and time. Date–time commands are attached to FreeRTOS millisecond timers and handled from the woken command scheduler task. Task wake-up time can be up to 1 millisecond. Due to possible clock drifts in the microcontroller, cumulative error might be caused in the FreeRTOS timers configured for long periods of time. The effects of microcontroller clock drifts shall be monitored in orbit.

4.6. Custom bootloader

For both the CDHS and CAM, a custom bootloader was designed and implemented. The bootloader allows for copying a firmware image from an external FRAM device and booting into a boot image chosen by the EPS. It processes a list of bootloader commands stored in the FRAM and logs status to a page in a microcontroller Flash. Before an attempt to boot a firmware image, its cyclic redundancy is calculated and compared against the one stored in the firmware image header. Without any bootloader commands or in the case of a problem with the FRAM, the bootloader selects the default firmware, based on the CDHS firmware select pin that can be toggled via the EPS.

5. HARDWARE TESTING

Several tests were carried out to ensure that the CDHS is able to survive the launch and operate in the space environment. These included functionality tests, performance and current consumption tests, as well as physical tests, such as vibration and thermal vacuum tests. During these tests, only hardware fault tolerance, meaning the ability to remain functional in the case of physical or electrical damage to components, was considered.

Functionality tests are a set of basic unit tests to ensure that hardware is functioning as intended. These involve communication tests with all attached devices, read and write tests on memory devices and the operating system with device drivers and error logging. The microcontroller supports overclocking of up to 128 MHz while remaining stable at room temperature. Instabilities of the overclocked microcontroller start to appear when the operational environment temperature exceeds 50 °C. At nominal 72 MHz frequency the device is able to operate within the ranges set in the specifications: from –20 °C up to 80 °C.

Various signal injection tests were performed on multiple engineering models and their components, including short-circuiting, stuck-at-fault, overvoltage, reverse polarity application, and electrostatic discharge. Results of these tests indicated that I^2C devices were most susceptible to noise on data lines and incorrect signal level applications, while the other devices continued to function without problems.

The performance of the CDHS was measured by running an in-house implementation of a Kalman filter [24] with predefined input values. The electric current consumption during the idle state with disabled peripherals is 47 mA. During the active phase when memory devices are accessed constantly, it increases to 54 mA. During the sleep mode with wake-up frequency of 1 kHz, the average current consumption decreases to 30 mA. Further improvements are possible with the implementation of the tickless sleep mode.

Vibration tests were performed according to the launch vehicle user manual of the launch provider Vega [25]. The resonant frequency of the populated board during the sweep-scan was found to be at 383 Hz. Such a high resonant frequency can be explained by the design where the placement of the 120 pin connector and large components (bus switches, microcontrollers) stiffens the construction. The functionality test performed after the vibration tests revealed one non-functioning flash memory module due to a loose solder joint connection. No other damage was observed after two repeated tests. Acceptance and qualification tests for launch were done on the system level [16,25].

The purpose of the thermal vacuum test was to measure the change in temperatures during the operation of the subsystem and to ensure that radiative and conductive heat dissipation were sufficient to keep the subsystem within the operational temperature range during uninterrupted operations. Physical connections of the subsystem for heat exchange consist of a main system bus connector (total conductive surface area of 43.5 mm^2) and four corner mounts (27.3 mm^2) with the total area of around 70 mm^2, which is negligible compared to the heat capacity of the PCB.

Temperature measurements were performed in a vacuum chamber at the level of 10e-4 mBar, which is enough to remove convectional heat dissipation. The engineering model of the subsystem was placed on a special aluminium heat transfer plate that emulates total heat capacity of the satellite. Contact between the heat transfer plate and the device tested was maintained only via special mounting rods through PCB corner mount slots, which represent the real assembly installation. Two calibrated temperature sensors were used: one placed on the microcontroller and the other on the aluminium heat transfer plate. In addition, the internal RTC temperature sensor and the internal microcontroller temperature sensor were tested. Temperature measurements were performed every 10 to 30 min while running the Kalman filter program.

The cooling of the prototype was done with a stream of evaporated liquid nitrogen as a heat removing transfer agent from the aluminium heat-exchange plate. During the cooling phase, the temperature of the heat-exchange plate dropped by ∼35 °C, while the board temperature (as registered by both the RTC and microcontroller internal sensors) dropped by ∼8 °C.

The heating of the board was performed for an hour after the cooling phase using an infra-red lamp located underneath the heat-exchange plate. A steep rise in the base plate temperature of ∼100 °C was observed for half an hour, then the CDHS PCB started to absorb heat energy in much higher amounts than it was able to radiate into the surrounding environment. The temperature of both the MCU and RTC rose by ∼40 °C, which is not critical. After the removal of the heat source all sensors cooled down, indicating successful heat dissipation. The whole process took about five hours, which is approximately three times longer than the expected orbital period. Temperature fluctuations in orbit should not be as severe and operational conditions are expected to be at ∼30% of the test values. The built-in temperature sensor was found to be imprecise due to the unstable reference voltage. Effects of thermal noise on the reference voltage source were found to be relatively large (up to 5% of 1.3 V). Other device inbuilt thermal sensors were found to be more accurate due to lower internal thermal variations.

6. DISCUSSION AND CONCLUSIONS

An operational CDHS was developed and tested for ESTCube-1. Redundancy and fault tolerance were emphasized during the development process. While the pre-launch tests carried out were not specifically designed to test the fault tolerance of the system, test

results indicate that a robust system has been developed. The CDHS operates in a stable manner within the physical limits set before development. Furthermore, thermal vacuum tests did not indicate any problems with either the vacuum environment or the ability to dissipate heat generated by the subsystem. Tests confirmed that excessive heat build-up in vacuum (which could disrupt operations of the system) was not probable. Even in case of direct infrared radiation heat, the capacity of the board was high enough to withstand heat build-up for a period of multiple orbits. Accumulated heat was rapidly dissipated via infrared radiation and thermal exchange through the physical connections. Temperature cycles did not impact performance and during the cycles changes in power consumption remained below 5%.

In the future, regular in-orbit functionality and memory integrity tests will be performed to acquire data on the reliability, performance, and radiation tolerance of the design. In addition, changes in current consumption, communication latencies, number of resets, and number of errors will be recorded. On the ground, these data will be correlated to the solar activity.

In future missions, improvements could be made to the subsystem to ease the implementation of advanced functionality and increase fault tolerance. Processor core with an inbuilt floating point unit support, cold redundant memory units based on microSD cards, ability to power down or power-recycle parts of the subsystem could be considered to be able to resolve a single-event latch-up in one device without having to power-cycle the whole CDHS.

ACKNOWLEDGEMENT

This research was supported by the European Social Fund's Doctoral Studies and Internationalisation Programme DoRa.

REFERENCES

1. *CubeSat Design Specification Rev. 12*. The CubeSat Program, Cal Poly SLO. California, 2009.
2. Woellert, K., Ehrenfreund, P., Ricco, A. J., and Hertzfeld, H. Cubesats: Cost-effective science and technology platforms for emerging and developing nations. *Adv. Space Res.*, 2011, **47**, 663–684.
3. Cutler, J., Bennett, M., Klesh, A., Bahcivan, H., and Doe, R. The radio aurora explorer – a bistatic radar mission to measure space weather phenomenon. In *Proc. 24th Annu. Small Satellite Conf*. Logan, Utah, 2010.
4. Deschamps, N. C., Grant, C. C., Foisy, D. G., Zee, R. E., Moffat, A. F. J., and Weiss, W. W. The BRITE space telescope: using a nanosatellite constellation to measure stellar variability in the most luminous stars. *Acta Astronaut.*, 2009, **65**, 643–650.
5. Borgeaud, M., Scheidegger, N., Noca, M., Roethlisberger, G., Jordan, F., Choueiri, T. et al. SwissCube: the first entirely-built swiss student satellite with an Earth observation payload. In *Small Satellite Missions for Earth Observation* (Sandau, R., Roeser, H. P., and Valenzuela, A., eds). Springer, 2010, 207–213.
6. Sarda, K., Eagleson, S., Caillibot, E., Grant, C., Kekez, D., Pranajaya, F. et al. Canadian advanced nanospace experiment 2: scientific and technological innovation on a three-kilogram satellite. *Acta Astronaut.*, 2006, **59**, 236–245.
7. Hamann, R. J., Verhoeven, C. J. M., Vaartjes, A. A., and Bonnema, A. R. Nano-satellites for micro-technology prequalification: the delfi program of delft university of technology. In *Selected Contributions of the 6th IAA Symposium on Small Satellites for Earth Observations*. Berlin, 2007, 319–330.
8. Nielsen, J. F., Larsen, J. A., Grunnet, J. D. et al. AAUSAT-II, A Danish Student Satellite. *I S A S Nyusu*, 2009.
9. Nielsen, J. D. and Larsen, J. A. Experiences and lessons learned during the Launch and Early Orbit Phase (LEOP) of AAUSAT-3. In *5th European CubeSat Symposium Book of Abstracts*. Brussels, 2013.
10. Scholz, A., Ley, W., Dachwald, B., Miau, J. J., and Juang, J. C. Flight results of the COMPASS-1 picosatellite mission. *Acta Astronaut.*, 2010, **67**, 1289–1298.
11. Rennels, D. A. Architectures for fault-tolerant spacecraft computers. *Proc. IEEE*, 1978, **60**, 1255–1268.
12. Aalbers, G. T., Gaydadhiev, G. N., and Amini, R. CDHS design for a university nano-satellite. In *Proc. 57th Int. Astronautical Congress*, Valencia, 2006, IAC-06-B5.7.05.
13. McLoughlin, I. V., Gupta, V., Sandhu, G. S., Lim, S., and Bretschneider, T. R. Fault tolerance through redundant COTS components for satellite processing applications. In *Proc. 2003 Joint Conf. of the Fourth Int. Conf. on Multimedia*, 2003, **1**, 296–299.
14. de Jong, S., Aalbers, G. T., and Bouwmeester, J. Improved command and data handling system for the Delfi-n3Xt nanosatellite. In *Proc. 59th Int. Astronautical Congress*, Glasgow, 2008, IAC-08.D1.4.11.
15. Nielsen, J. D. and Larsen, J. A. A decentralized design philosophy for satellites. In *2011 5th Int. Conf. Recent Advances in Space Technologies (RAST)*. Istanbul, 2011, 543–546.
16. Lätt, S., Slavinskis, A., Ilbis, E., Kvell, U., Voormansik, K., Kulu, E. et al. ESTCube-1 nanosatellite for electric solar wind sail in-orbit technology demonstration. *Proc. Estonian Acad. Sci.*, 2014, **63**(2S), 200–209.
17. Janhunen, P., Toivanen, P. K., Polkko, J., Merikallio, S., Salminen, P., Haeggström, E. et al. Electric solar wind sail: toward test missions. *Review Sci. Instruments*, 2010, **81**, 111301:1–11.
18. Slavinskis, A., Kvell, U., Kulu, E., Sünter, I., Kuuste, H., and Lätt, S. High spin rate magnetic controller for nanosatellites. *Acta Astronaut.*, 2013, **95**, 218–226.
19. Bouwmeester, J. and Guo, J. Survey of worldwide pico-

and nanosatellite missions, distributions and subsystem technology. *Acta Astronaut.*, 2010, **67**(7–8), 854–862.
20. Pajusalu, M., Ilbis, E., Ilves, T., Veske, M., Kalde, J., Lillmaa, H. et al. Design and pre-flight testing of the electrical power system for the ESTCube-1 nanosatellite. *Proc. Estonian Acad. Sci.*, 2014, **63**(2S), 232–241.
21. Johnson, B. W. Fault-tolerant microprocessor-based systems. *IEEE Micro*, 1984, **4**(6), 6–21.
22. Fleetwood, D. M., Winokur, P. S., and Dodd, P. E. An overview of radiation effects on electronics in the space telecommunications environment. *Microelectronics Reliability*, 2000, **40**(1), 17–26.
23. Wrachien, N. Advanced memories to overcome the flash memory weaknesses: a radiation viewpoint reliability study. Ph.D. dissertation. Dept. Inf. Eng., Padova University, Padova, Italy, 2010.
24. Slavinskis, A., Kulu, E., Viru, J., Valner, R., Ehrpais, H., Uiboupin, T. et al. Attitude determination and control for centrifugal tether deployment on the ESTCube-1 nanosatellite. *Proc. Estonian Acad. Sci.*, 2014, **63**(2S), 242–249.
25. *VEGA User's Manual*. Issue 3, 2006, France.

ESTCube-1 veatolerantse käsu- ja andmehaldussüsteemi projekteerimine

Kaspars Laizans, Indrek Sünter, Karlis Zalite, Henri Kuuste, Martin Valgur, Karl Tarbe, Viljo Allik, Georgi Olentšenko, Priit Laes, Silver Lätt ja Mart Noorma

On kirjeldatud ESTCube-1 käsu- ja andmehaldussüsteemi projekteerimist, selle tehnilist lahendust ning stardieelseid katsetusi. Süsteemi arendamisel pöörati erilist tähelepanu selle robustsusele ja veatolerantsusele. Riistvara komponentide puhul rakendati kombinatsiooni kuumast ja külmast liiasusest. Tarkvaras realiseeriti protseduurid vigade avastamiseks ja vea korral süsteemi töökorra taastamiseks. Lisaks riistvara töökindlusele simuleeritud rikete korral testiti ka käsu- ja andmehaldussüsteemi füüsilist vastupidavust laias temperatuurivahemikus ning kontrollitud vibratsiooni tingimustes. Artiklis on välja pakutud ideid süsteemi veatolerantsuse kontrollimiseks orbiidil.