IMPROVED COMMAND AND DATA HANDLING SYSTEM
FOR THE DELFI-N3XT NANOSATELLITE

S. de Jong, G.T. Aalbers, J. Bouwmeester
Chair of Space Systems Engineering, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands, Sybren.deJong@student.tudelft.nl, G.T.Aalbers@delfic3.nl, Jasper.Bouwmeester@tudelft.nl

Abstract
The Delfi-C$^3$ nanosatellite successor, Delfi-n3Xt, is currently under development at Delft University of Technology. This nanosatellite based on a three-unit CubeSat form factor has been improved through the implementation of a high-speed downlink, three-axis active attitude control and a single-point-failure free implementation of batteries in the electrical power system. Failure of the batteries will therefore not lead to failure of the primary mission as has, in the past, been the case with many other nanosatellite missions. The functional analysis and design of the command and data handling system (CDHS) of Delfi-C$^3$ and the improved CDHS architecture of Delfi-n3Xt are presented in this paper. The main design drivers for the CDHS of Delfi-C$^3$ were the available technology and the absence of batteries. These design drivers enforced specific hardware components which, however, resulted in undesired behavior during integration and testing. In particular low-speed devices on the bus were suppressing the performance of the CDHS and the high-speed systems of Delfi-C$^3$. Delfi-n3Xt requires a higher performance since much more data will be produced by the five payloads, stored and sent down with a high-speed downlink. The architecture presented in this paper complies with this, while it is based on the architecture of Delfi-C$^3$.

1 Introduction
The Delfi nanosatellite programme of the Delft University of Technology (TU Delft), in which more than 75 students have been participating, focuses on qualification of novel microtechnologies from the Dutch space industry [1]. The second satellite in this programme, Delfi-n3Xt, is currently in the first detailed design phase. Figure 1 shows the configuration of Delfi-n3Xt. This nanosatellite will be (10 x 10x 34) cm, a three-unit CubeSat. Compared to the first satellite in this programme, Delfi-C$^3$, Delfi-n3Xt will additionally be equipped with batteries, an active attitude determination and control subsystem (ADCS) and a high-speed S-band transmitter. Five innovative payloads [2, 3] will be accommodated for microtechnology space qualification and scientific research on space weather. Next to these mission goals, Delfi-n3Xt will be equipped with a radio amateur transponder to be used by the radio amateur community. This paper describes the preliminary architecture of Delfi-n3Xt which is based on the CDHS of Delfi-C$^3$.

The five payloads that will fly on Delfi-n3Xt are:

**Cool Gas Micropropulsion system (T$^3$µPS)**
This is a microsystems based thruster system for future small satellites and is under development by TNO, TU Delft and University of Twente.

**Multifunctional Particle Spectrometer (MPS)**
This radiation detector is under development by cosine Research BV, which is able giving the composition (type of particles, angle of incidence and energy) of the captured radiation.

**Space Flash Memory (SPLASH)**
NLR is developing a reliable and low-cost data storage
solution based on COTS flash memory cards that is medium tolerant to radiation.

Hydrogenated amorphous silicon solar cells
DIMES, which is part of TU Delft, does a lot of research and development on solar cells. This solar cell degradation measurement (SDM) in space will improve or verify lab research.

Efficient Nanosatellite Transceiver Module
ISIS BV is developing a transceiver module for future small satellites. This transceiver is based on the transceiver of Delfi-C^3 and will be qualified and tested on Delfi-n3Xt.

According to [4] the CDHS performs two major functions. It receives, validates, decodes, and distributes commands to other spacecraft systems and gathers, processes, and formats spacecraft housekeeping and mission data for downlink or use by an onboard computer. At Delfi-n3Xt, power switching systems, deployment systems and the onboard computer (OBC) are additionally part of the CDHS. The complete list of functional parts that belongs the CDHS of Delfi-n3Xt is:

- Databus:
  - Bus harness,
  - Bus control,
  - Error detection and correction (EDAC).
- OBC:
  - Command processing,
  - Software,
  - Data storage.
- Local subsystems controllers:
  - Power switching,
  - Housekeeping data collection.
- Deployment control systems:
  - Controlling,
  - Monitoring.

2 The Delfi-C^3 CDHS architecture

The successful Delfi-C^3, launched on 28 April 2008, is still operational. It functions currently as linear transponder for the radio amateur community. In the first three months, Delfi-C^3 performed the scientific mission. Delfi-C^3 is based on a battery free design. Absence of electrical power in eclipse and limited peak power were therefore large design drivers. Another design driver is single-point-failure (SPF) free design philosophy, no SPF may therefore threat the primary mission.

Since the orbit of Delfi-C^3 is a Sun synchronous orbit with eclipse, absence of power and a reboot occur every orbit period. The limited available power and the COTS OBC (the CubeSat kit FM340 board) lead to selecting low-power design choices: an I^2C bus and ultra low power microcontrollers and OBC.

2.1 Decentralized electrical power subsystem

Only one onboard computer is implemented in Delfi-C^3. The SPF free design philosophy forced the implementation of a redundant system to this OBC. A decentralized redundancy system is implemented in which every subsystem monitors the OBC’s activity. Once an OBC time-out is detected, every subsystem becomes an autonomous system and will only perform task to fulfil the primary mission. The overall satellite performance is reduced in this case. Another SPF free implication is decentralized power switching. Every subsystem continuously receive power from the main electrical power subsystem (EPS) and switching is done at every subsystem itself. A command from the OBC or an autonomous decision by the local EPS
Figure 2: Subsystem part of the decentralized EPS of Delfi-C\textsuperscript{3}. A fail-safe power regulator circuit powers the EPS microcontroller (PIC) which drives a MOSFET. Switching takes place on commands received from the OBC or autonomously by the EPS microcontroller software.

Limited available power due to the absence of batteries and multiple microcontrollers that have to be active all the time, requires ultra low power consumption the local microcontroller. Ultra low power microcontrollers (Microchip PICs) running on a very low-speed (31 kHz) are therefore implemented for this function. Figure 2 shows the subsystem part of the decentralized EPS.

2.2 Lessons learnt from Delfi-C\textsuperscript{3}

Major drawbacks of a single system with low-speed microcontrollers running at 31 kHz and high-speed microcontrollers running at 1 MHz and 20 MHz arose during integration and testing. Synchronization and data collection from all subsystems in one second showed to be problematic. Most issues were mitigated through software implementations, but the performance stays in practice limited to the about 50% of the slowest microcontroller.

The data frames from Delfi-C\textsuperscript{3} received on ground are tagged with an identifier that consists of the boot counter and a frame number that resets after each reboot. The boot counter is stored in the flash memory of the OBC and is limited by the write cycle endurance of this memory (about 100 000 times, which corresponds with 17 years). If the satellite will have frequent reboots due to power drops or other reasons, the lifetime will be shortened very quickly. A protection for this circumstance is implemented that prohibits updating the boot counter every reboot. During the first days of operation, frequent reboots were observed and the flash memory was ‘frozen’ by a telecommand. This resulted in lots of received data frames with equal boot counters. Since data frames are received and forwarded by a huge amount of radio amateur over the world, it is was very difficult to sort all, almost similar, data frames.

The major lessons for the CDHS design of Delfi-n3Xt are:

- Find a solution for high-speed and low-speed devices in a single system (architecture, hardware, software),
- Find a solution for really unique data frame identifiers,
- Find a solution where node cannot hang-up the bus,
- Make commanding of critical commands more reliable (i.e. with status feedback loop),
- Take precautions to bus line crosstalk.

3 The Delfi-n3Xt CDHS architecture

Not all design choices in the CDHS of the Delfi-C\textsuperscript{3} project were well performed due to rushes in the project. Thorough investigations and trade-offs for microcontrollers and bus standard were never performed. A complete preliminary CDHS design process of Delfi-n3Xt is therefore performed with the Delfi-C\textsuperscript{3} CDHS design in mind as heritage.

In the preliminary design of the CDHS of Delfi-n3Xt, the following parts are defined:

- Databus architecture and standard,
- Local power switching,
- Onboard computer and redundancy measure,
- Deployment control systems,
- High level onboard software.

These parts are all described in this section and the complete preliminary CDHS hardware architecture in shown in Figure 9 on the last page.

3.1 Databus architecture and standard

The databus distributes all commands and data to all systems. Several topologies are possible: star,
ring, mesh, line, tree, point-to-point and bus. In addition, communication types are parallel and serial communication. A bus topology with serial communication is chosen; a) to keep the wiring harness small with the high number of nodes (more than ten) and b) for high reliability. A parallel bus would require too much wiring and lacks reliability.

Since most available microcontrollers have \( \text{I}^2\text{C} \) and/or CAN (Controller Area Network) interfaces, these two standards are the best options for the bus. Both bus standard are described in the next sections. Wireless communication is discarded due to the complexity and electromagnetic compatibility (EMC) issues.

### 3.1.1 The \( \text{I}^2\text{C} \) bus

The \( \text{I}^2\text{C} \) bus is developed by Royal Philips and makes use of a clock line (serial clock (SCL)) and a data line (serial data (SDA)). Both lines are pulled up by resistors to the supply voltage. Devices on the bus can communicate by switching (pulling) the lines to the ground, this is also known as open drain design. The nodes on the bus can therefore communicate without adding power to the bus. This principle is shown in Figure 3. \( \text{I}^2\text{C} \) has its origins in consumer electronics and is mainly focuses on low power consumption and simplicity. A large number of peripherals are available, amongst others: I/O ports, memories, timers, temperature sensors, etc. Most \( \text{I}^2\text{C} \) devices support speeds from 0 bps up to 400 kbps. The protocol is simple and minimum hardware components are required for this standard.

Main advantages to use the \( \text{I}^2\text{C} \) standard in Delfi-n3Xt are:

- Many industry standard components available,
- Low power: typical 10 mW independent on the number of nodes,
- Simple protocol, simple hardware,
- Heritage from Delfi-C³ (tools in house).

Main disadvantages of the \( \text{I}^2\text{C} \) standard are:

- Low robustness,
- No bus protection to nodes causing the bus to hang using standard hardware,
- No built-in error detection and correction, only acknowledgement.

### 3.1.2 The CAN bus

The CAN bus originates from the automotive industry. The bus standard is therefore designed on reliability and robustness. The CAN bus makes use of one data channel, no clock line. This data channel uses a differential (twisted) pair (CAN\(_H\) and CAN\(_L\)) that limits the electromagnetic susceptibility (EMS) of the bus. The CAN protocol includes CRC checksum for every data package that makes the bus even more reliable. The datarate of the CAN bus is 40 kbps to 1 Mbps, the overhead is however large, minimal 40%.

Multiple components at a node are required for communication as shown in Figure 4. A controller is required to cope with the protocol and a transceiver for the physical layer. Both components require more power which can be reduced by decreasing the bus load (active time of the component). This can be done by increasing the bus speed while keeping the total amount of data equal. There are multiple types of CAN transceivers of which some have extra features to increase the robustness. Those features are: TX-time out to prevent a node hanging up the bus, bus wiring fault tolerance, slope control to reduce electromagnetic interference (EMI) and low power sleep modes. Most features come with the cost of limited data rate to 125 kpbs and higher power consumption at every node.

Main advantages to use the CAN standard in Delfi-n3Xt are:

- Robust (wire fault tolerant, TX time-out),
- Good EMC properties (differential pair, slope control).

Main disadvantages of the CAN standard are:

- More hardware components required, more complex,
Figure 4: CAN bus components and signal levels at both sides of the transceiver.

- No simple peripherals available, a microcontroller plus software is required at every node,
- Overall high complexity.

3.1.3 **I²C-CAN Trade-off**

The advantages and disadvantages of both standards differ much. A thorough study is done to figure out which standard is the best choice for Delfi-n3Xt. A large disadvantage is the use of the CAN bus for the decentralized power switching. It is not possible to design a system that is as low-power as the system applied in Delfi-C³ using CAN. The main reason for this is the high average power required by the nodes during communication since every node has to participate and high-speed microcontrollers are required to handle the CAN protocol and bit-timing (synchronization between the nodes). For SPF free design philosophy and heritage decentralized switching is preferred.

To solve these issues and find the best option, eight databus architecture options based on both bus standards were generated. Some architectures consist of multiple busses (ie. a high-speed bus and a low-speed bus), these options were more complex without adding reliability and were therefore discarded. Finally two strong options remained after the trade-off:

1. A single I²C bus with decentralized power switching, or
2. A single CAN bus with centralized power switching.

The first option was chosen. The main reasons are: the heritage of Delfi-C³ (transceiver, tools), simplicity on each subsystem (KISS), SPF free origin, risk on delay in the schedule and keeping the option open to use the MSP430 microcontroller as basic OBC part which does not has a built-in CAN controller. The latter reason is important since Delfi-C³ is equipped with an MSP430 based OBC. There is a good chance that the OBC of Delfi-n3Xt will again be based on the MSP430 since limited time is available for OBC design. Furthermore using I²C, local power switching of subsystems can be easily accomplished. Industry standard chips can be used for this function that do not even need software, thus lower complexity. The use of simple chips solves also the problem using high-speed and low-speed systems on a single bus. Table 1 shows the main conclusions whereon the choice for the I²C option is selected.

The bus voltage is set on the minimum standard value of 3.3 V (pull-up voltage) for power reasons. Higher voltages could make the bus more robust to interference. The bus will be a multi-master bus with the OBC as primary master and a controller on the primary radio as back-up.

An important condition that came with the choice for a single I²C was improving the reliability and robustness of the standard bus system. Measures are taken to make the I²C bus more reliable, these are based on the drawbacks of I²C with respect to the CAN standard. One of these measures is the implementation of a bus protector at every node that isolates a subsystem from the bus if it causes a hanging bus. A sample circuit [5] is shown in Figure 5. The basic component is a P82B96 I²C buffer that is made out of bipolar technology, which receives power from a timer circuit that monitors both the data and the clock line of the subsystem. The buffer enables also independent and flexible bus levels at both sides. A major drawback of

<table>
<thead>
<tr>
<th>Table 1: Main conclusions of the investigation of both options.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The I²C option (1)</td>
</tr>
<tr>
<td>KISS:</td>
</tr>
<tr>
<td>- Use of simple chips</td>
</tr>
<tr>
<td>- Architecture</td>
</tr>
<tr>
<td>Pros:</td>
</tr>
<tr>
<td>Heritage</td>
</tr>
<tr>
<td>MSP430 Compatible</td>
</tr>
<tr>
<td>Lower power</td>
</tr>
<tr>
<td>SPF free origin</td>
</tr>
<tr>
<td>Cons:</td>
</tr>
<tr>
<td>Lower robustness</td>
</tr>
<tr>
<td>Planning overdue risk</td>
</tr>
<tr>
<td>Less modular</td>
</tr>
<tr>
<td>More components</td>
</tr>
</tbody>
</table>
this circuit is the increase in power consumption of about 10 mW per active node. Testing of failure cases will figure out which nodes really require this protective circuit. A second reliability measure is redundant wiring that adds wiring interruption fault tolerance in the case of bad connections. This is already applied in Delfi-C$^3$ like shown in Figure 6.

3.2 Local Power Switching

Decentralized power switching of subsystems will again be applied. The problem with the low-speed EPS microcontrollers that drive power MOSFETs is tackled by applying industry standard I$^2$C chips like PCF8574 I$^2$C I/O expanders. This I/O expander (or I/O port) has a very low power consumption of 10 $\mu$A and 100 $\mu$A for standby and operating mode respectively (maximum ratings from the datasheet). This chip only needs a short command to toggle an output or read the status of an input. The I$^2$C bus speed is limited to 100 kbps by the use of the PCF8574. If this becomes a problem (which seems to be unlikely), the PCA8574 can be used that is capable to operate on a 400 kbps I$^2$C bus, this comes at the cost of a five times higher power consumption.

The drawback of using simple I/O chips is the absence of autonomy. Since Delfi-n3Xt has much more systems connected to the bus than Delfi-C$^3$ such autonomy would be too complex. Another measure for OBC redundancy will be taken as explained in the next section. The major advantages of the I/O chips are simplicity and the low power consumption in combination with high-speed bus communication.

3.3 Onboard Computer and redundancy measure

Several options were generated for the redundancy measure in the high-level OBC architecture design. After trading multiple, two strong options remained:

1. A full redundant OBC (two identical OBCs), and

2. Implementation of basic OBC functions at a microcontroller at the primary radio.

The first option is a common option in most satellites. Arbitration between both OBCs is required to decide which will control the satellite. A common method is a separate watch dog timer (WDT) module that monitors the OBC. Another WDT method, which is applied at Delfi-C$^3$, is the use of synchronization commands and timers. If several systems did not receive such a command, the system will switch to autonomous mode. This system is basically SPF free on hardware level and is therefore preferred. A major drawback of two identical OBCs is the probability of a single mode failure. To avoid this SPF a third backup system should be implemented which increases the complexity.

The second option not common but is also not rare in satellite designs. The primary transceiver will be equipped with basic OBC functionality in the telecommand decoder. A microcontroller at this transceiver will take over the basic or critical OBC functions if it fails. Arbitration as applied at Delfi-C$^3$ (synchronization commands) is the best method to select the active satellite controller. The major drawback of this option is the reduced performance if the main OBC fails. However, the whole system is simpler, has a better reliability and all satellite functions will be directly
accessible by telecommanding via the primary radio.

While focussing on simplicity, reliability and required functionality, the second option is chosen as redundancy option for the main OBC. Another reason to choose this option are the design improvements of the primary transceiver of which the implementation of a telecommand interpreter (TCI) at the receiver is intended. The OBC redundancy function will mainly have its design complexity in software. Investigations that are still open for this redundancy measure are the level of autonomy (i.e. only ground control, perform minimum mission goals, etc.) and the possibility for software updating.

3.3.1 Microcontroller selection

The selection of the main microcontroller for the OBC will take place in the first detailed design phase. Functions of the OBC are mainly processing commands, scheduling of tasks, management on anomalies and moving data. No difficult operations as data conversion or compression are required, the heaviest task will probably be error detection and correction functions. The heavy function of telecommand interpreting, which is basically sampling and decoding of telecommands, will take place on the receivers. The ADCS algorithms will be performed at a computer at the ADCS itself. According to this information, an MSP430 based OBC seems to be a good candidate, like applied at Delfi-C³. The design tools, software and good experience is available for this microcontroller.

3.3.2 Data storage

Unlike Delfi-C³, for the Delfi-n3Xt mission, it is intended that all produced payload data will be stored onboard and linked down to ground station(s). Mass data storage has therefore to be implemented in the CDHS. It has to be investigated which type of memory this should be. The data storage payload of NLR, SPLASH[2], can be the redundant system for this data storage.

3.3.3 Real Time Clock

One of the lessons learnt from Delfi-C³ is adding really unique identifiers to the data frames, preferably independent of a data storage system. A real time clock (RTC) or other time keeping or reference system is therefore a preferred feature. Another reason to implement this is the ability for proper scheduling of tasks. The best component for this function is an RTC as stand-alone chip. The PCF8593 I²C RTC chip is indicated as candidate to be tested in the first detailed design phase.

3.4 Deployment control systems

For the deployment and monitoring of deployables, the same I/O expanders will be applied as used for local power switching on the subsystems. For redundancy of these chips, it is wise to double the I/O expanders and place them parallel. The physical deployment and hold-down systems will be equal as applied on Delfi-C³ because of the simplicity and the good experience. These systems are the modular antenna boxes (MABs) [1] and solar panel hold down system which are based of the thermal knife principle: standard resistors cutting a Dynema wire.

3.5 Onboard Software

Delfi-n3Xt has five payloads and multiple mission goals [2]. To keep the software simple only three main modes are defined: Initiation mode, Operational mode and Safe mode. The first two modes are nominal modes and the latter mode will be active if the main OBC fails.

Initiation will be active directly after separation from the launcher. In this initiation mode, systems will be deployed and compulsory delays will be taken into account. The operational mode will be the nominal mode in which all experiments and tasks will be conducted according to a schedule. This schedule can be predefined, can be updated by the OBC and/or updated by telecommand. In safe mode the TCI will control Delfi-n3Xt and will become active if the main OBC fails or on telecommand. The TCI will continuously monitor the OBC and will take over control after time-out. In the first detailed design phase a software update possibility using the TCI will be investigated. Figure 7 shows the mode transitions with the transition conditions.

To perform all qualification and scientific tasks in one operational mode, task packages are defined. The mode breakdown with task packages is shown in Figure 8. Scheduling of these task packages will be based on the available power and data constraints. This schedule will be flexible by updating the schedule by ground control and a certain level
of autonomy. In the event of failure of batteries or ADCS, the schedule has to be adapted to the new constraints.

4 Summary

Delfi-n3Xt that will space qualify five microtechnology payloads will be equipped with an improved CDHS based on Delfi-C3\(^3\). The main issue of the databus of Delfi-C3\(^3\) was the presence of low-power, low-speed microcontrollers suppressing the performance of the databus. This problem is mitigated by replacing the low-speed microcontrollers by standard I/O ports that draw even less power.

A trade-off is performed between a \(\text{I}^2\text{C}\) and a CAN databus. Simplicity and heritage were the main reasons to implement a single databus based on the \(\text{I}^2\text{C}\) standard. Delfi-n3Xt will be controlled by a single OBC with redundancy measures in the primary radio. The OBC will use three modes and five task packages to perform all mission goals. The preliminary CDHS architecture of Delfi-n3Xt is shown in Figure 9 on the next page.

References


Figure 9: The preliminary CDHS design of Delfi-n3Xt.