Towards an innovative electrical interface standard for PocketQubes and CubeSats

Bouwmeester, J.; van der Linden, S. P.; Povalac, A.; Gill, E. K.A.

DOI
10.1016/j.asr.2018.03.040

Publication date
2018

Document Version
Accepted author manuscript

Published in
Advances in Space Research

Towards an Innovative Electrical Interface Standard for PocketQubes and CubeSats

J. Bouwmeester[1], S.P. van der Linden[1], A. Povalac[2], E.K.A. Gill[1]

jasper.bouwmeester@tudelft.nl, spvdlinden@gmail.com, povalac@feec.vutbr.cz, e.k.a.gill@tudelft.nl

[1]Delft University of Technology, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS, Delft, The Netherlands

[2]Brno University of Technology, Faculty of Electrical Engineering and Communication, Technicka 12, 616 00 Brno, Czech Republic

Abstract

Developers experience issues with the compatibility, connector size and robustness of electrical interface standards for CubeSats and PocketQubes. There is a need for a lean and robust electrical interface standard for these classes of satellites. The proposed interface standard comprises a linear data bus which is used for housekeeping data, internal commands and small-to-moderate payload data. A community based analytic hierarchy process is used for the trade-off of design options, resulting in the selection of RS-485 as standard data bus, mainly due to its low power consumption and high effective data throughput compared to other candidates. Several switched and protected battery voltage lines are distributed from the central electrical power subsystem unit to the other subsystems to enable a simple and efficient power distribution. The harness comprises a 14 and 9 pin stackable connector for CubeSats and PocketQubes, respectively, occupying very little board space.

1 Introduction

CubeSat and PocketQube Developers experience issues with the compatibility, connector size and robustness of electrical interface standards. This paper describes the process towards a lean electrical interface for CubeSats and PocketQubes which should tackle these issues. The primary objective of this paper is to select an appropriate data bus based on extensive analysis and (future) needs of satellite developers. The secondary objective is to show targets and aggregate results of prior studies towards the definition of a lean electrical interface standard.

In this paper, the results of an extensive trade-off for the electrical interfaces for PocketQubes and CubeSats are presented. The standard electrical interfaces typically comprise one or more digital data busses used for the transport of data between subsystem and power distribution lines. Optionally, an electrical interface standard can also comprise lines for baseband radio signals, analogue signals and general input/output.

Based on design targets specified in section 0, an appropriate standard data bus architecture is presented in chapter 2. Chapter 3 describes the trade-off process and chapter 4 provides trade-off results for the data bus. Chapter 5 provides a brief analysis on power distribution. In chapter 6 a new electrical bus interface standard for PocketQubes and CubeSats is proposed, which is lean, facilitates efficient power distribution and ensures inter subsystem compatibility. Finally, conclusions and a future outlook is provided in chapter 7.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

1.1 Background

In a worldwide survey on CubeSat electrical interfaces, it became clear that many CubeSat developers experience issues with the de-facto standard electrical interface based on the PC/104 connector, part of the PC/104 standard and the I^2C data bus (Bouwmeester et al., 2017). Documents which describe the pin allocation for PC/104 connectors for CubeSats do not exist and it was previously found that subsystems from different commercial suppliers use different pin allocations (Bouwmeester and Santos, 2014).

A proposal for a dedicated CubeSat electrical interface standard comes from UNISEC (Busch, 2015). It defines, amongst others, a standard 50 pin stacked connector between subsystems comprising power distribution at various voltage levels, several options for data interfaces (I2C, UART, JTAG), reset lines and several General Purpose Input/Output pins (GPIOs).

At this moment I^2C is dominant in CubeSats. However, many developers experience in-orbit issues with this bus (Bouwmeester et al., 2017). Specifically, in-orbit bus lockups of the I^2C data bus, the large connector, lack of a clear standardized power bus distribution and protection and lack of a fixed pin allocation were identified as key issues. The Delfi-C^3 CubeSat suffered from a high bit-error rate and bus lock-ups (Cornejo et al., 2009) with I^2C in-orbit. From these lessons learned, it can be concluded that the theoretical behavior of a data bus does not always apply in practice.

Another study proposes a split data and power interface using daisy chained connections (Riot et al., 2014) and call this the CubeSat Next Generation Bus (CNGB). For the data interface, the CAN bus was chosen with the high level of hardware supported features and extensive heritage in the automotive industry as main reasons. Details on the trade-off are, however, not provided. The paper mentions extensibility to larger than-3U-CubeSats as one of the programmatic goals. The split data and power connectors in a daisy-chained configuration is far from a small and lean solution and would not be suitable for smaller CubeSats or PocketQubes.

For PocketQubes, the only existing standard is PQ60 (Becnel et al., 2015). This standard is more clearly defined than the PC/104 implementation on CubeSats. It defines the connector, the pin allocation and the printed circuit board outline. It supports several different power outputs, SPI and I^2C data interfaces and many GPIOs. It uses a proprietary connector which is limited in current (0.2 A per pin).

The literature described above shows that most used and proposed electrical data busses are aimed at versatility, leaving a large design freedom to the subsystem developers. The disadvantage for these standards is that they do not guarantee compatibility and are far from optimal in terms of wiring harness. A lean standard with a minimum amount of clearly defined interfaces would counter these issues, but the lack of design freedom require a careful trade-off of the data bus and architecture for power distribution.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

1.2 Design Targets for a Standard Electrical Interface

Following the findings described in section 1.1, the following top level targets for electrical bus interfaces have been determined:

1. The interface is lean in volume and wiring harness.
2. The interface has a consolidated data bus and power distribution allocation.
3. The interface supports expected future performance demands.
4. The interface enables a high satellite power efficiency.
5. The interface is low in complexity.
6. The interface is expected to receive support in the community.
7. The interface is robust and reliable.

2 Standard Data Bus Architecture & Candidates

Before selecting an appropriate data bus or busses for an electrical interface standard, it is helpful to define a suitable data bus architecture for a typical CubeSat or PocketQube.

2.1 Data Bus Architecture

For this study, it is assumed that both satellite form factors make use of a distributed computing architecture, in which each physical subsystem of the satellite has its own microcontroller (or processor) to manage the local functionality. Some physical subsystems have components for which a digital interface is required, such as temperature sensors and reaction wheels. When they are physically implemented on the same board, a local data bus can be used, which can be of different kind and/or network topology (e.g. SPI). A central Onboard Computer manages the satellite by commanding the local microcontrollers and acquiring (housekeeping) data. Very advanced concepts, for example fractionated spacecraft or decentralized real time operations without a master node (central OBC), is considered out of scope for this study. While these concepts may have potential in the future, it is unlikely that these would receive wide community support in the short term.

For CubeSats and PocketQubes it is expected that for housekeeping data and internal commands, a linear bus connected to all physical subsystems will suffice. In a linear bus network topology, the same set of wires or lanes are used to connect multiple nodes on the bus together. This is different from a point-to-point bus, which can only connect two nodes together. A linear bus has the major advantage for very small satellites that the amount of wiring is limited when stacked connectors or some form of bus backbone is used. Secondly, the pin-out is fixed for all subsystems and the amount of potential nodes is not constrained by the amount of wiring.

A higher data rate of a linear data bus will support modest payloads connected to the same bus, which maintains a simple architecture. A linear data bus is, however, limited in speed because of cumulative electrical capacitance on the bus when adding nodes and the increasing demands on all nodes in terms of clock frequency and data handling capacity. Sophisticated and demanding payloads such as optical instruments produce, besides some modest housekeeping data, large amounts of payload data which may need to be stored and sent to selected ground stations over a high speed radio transmitter (Selva and Krejci, 2012). In a study on CubeSat science missions (Poghosyan and Golkar, 2017), it was found that high-speed radio links up to 100 Mbit/s are currently commercially available and being integrated in CubeSats. For these type of payloads it is expected that point-to-point busses will be required between the payload, potential data storage and a high speed radio.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

Wireless communication inside a CubeSat is not common(Bouwmeester et al., 2017), but a few experiments have been performed with a wireless sun sensor (de Boom et al., 2011) using a proprietary wireless standard. A custom optical variant of the CAN bus has even been demonstrated as main data bus (Arruego et al., 2016). The advantages of wireless communication become most apparent for sensors which are remote from the internal printed circuit board and could potentially be self-powered and thus completely wireless (Amini et al., 2009), e.g. sun sensors. Wiring, in this case, is typically a major burden. Whenever there is potential for these sensors to locally power themselves, wireless data busses may provide a great solution. In a previous study, Bluetooth 4.0 was evaluated as one of the current best options (Schoemaker and Bouwmeester, 2014). For data communication between the main subsystems, where a wired electrical interface is required for electrical power distribution, the potential reduction in wiring harness is limited while complexity would increase.

Figure 1 shows the proposed data bus architecture, which is considered to be appropriate to fulfill the requirements of many CubeSat and PocketQube missions in the near and long term future. All subsystems and payloads connect to a linear housekeeping bus which is mastered by the Onboard Computer. Low speed payloads can use this bus for payload data as well. Sophisticated payloads, together with data storage and a high speed transmitter, use point-to-point busses to make a high data throughput possible while relieving the onboard computer for its critical tasks. Remote self-powered wireless sensors connect to the OBC and/or ADCS through either wireless links or dedicated local data bus branches. It should be noted that individual CubeSats and PocketQubes can deviate in terms of amount and types of physical subsystems. The centralized concept, where the OBC manages the satellite as a master device is a starting point for further analysis. In this architectural concept the OBC can still be physically relocated, physically combined with other subsystems or taken over by a redundant backup system.

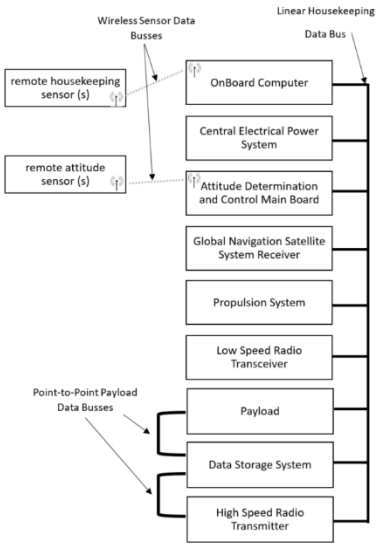

Figure 1. Proposed data bus architecture (example)

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

2.2 Linear Housekeeping Data Bus Candidates

The primary focus for this study is currently on data busses which are specified by a physical layer (ISO layer 1). As the number of existing busses and their variants is large, first a selection has been applied based on the targets described in section 0. Next to these targets, only data busses which are widely applied in terrestrial environments are considered. CubeSats and PocketQubes benefit from the associated wide availability of commercial integrated circuits, test equipment, documentation and user support for these data busses.

The candidates considered for the linear housekeeping data bus are: Inter-Integrated Circuit (I^2C), differential I^2C, Controller Area Network (CAN) and Recommend Standard 485 (RS-485).

I^2C is a single ended synchronous bus: it has clock and data lines (Leens, 2009). The lines are actively pulled high by a resistor (typically 4.7 kΩ) and have to be pulled low by its controller for communication. When applied in a small satellite, bus buffers need to be added to be able to isolate unpowered subsystems from the main data bus.

I^2C can be made differential by replacing the bus buffers by a dedicated differential driver (NXP Semiconductors, 2016), which yields four lines in total. As this is an easy-to-implement feature that slightly deviates from the standard while improving the robustness of the bus, this variant is added even though it is not widely implemented yet.

CAN is an asynchronous differential data bus developed for the automotive industry (Lawrenz, 2013). Some microcontrollers include a CAN controller, but most require an external controller connected to a local data bus that is supported internally by the microcontroller (e.g. SPI). An external differential driver is required in both cases.

RS-485 is an asynchronous differential data bus. It uses the Universal Asynchronous Receiver Transmitter (UART) that can be found on almost every microcontroller (Soltero et al., 2010). A dedicated external differential driver is required to make a RS-485 bus. This bus is the only one of the four options which is only specified on the physical layer and not on the higher OSI (Open System Interconnection) layers.

3 Trade-off Process for Housekeeping Data Bus

This chapter describes the trade-off method, criteria, test setup and community survey input.

3.1 Trade-off Method

Trade-offs with multi-disciplinary criteria are sensitive to errors and subjective scoring and weighting. Furthermore, a typical pitfall is to assign scores relative to the option space rather than the overall project or system scope. For example: a component trade-off leads to the discovery of several options ranging from € 2 to € 20. If the option space would be used to define a linear scoring range from 1 to 10, the individual score would be equal to the component cost divided by € 2. The cheapest option receives a score of 1 and the most expensive receives a score of 10. This may make sense for a € 100 mobile phone, but not for a 100 k€ satellite project.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

Methods dealing with some of the sensitivities of trade-offs exist, such as the well-established Analytic Hierarchy Process (AHP) (Saaty, 2008). This method provides a structured approach to derive criteria, relative weighting of these criteria and the grading of all options for each criterion. Saaty, however, also states that the *interpretation* of an option within a certain criteria, even if these itself are objective facts, is always subjective. The AHP method uses pair-wise comparisons between criteria and options to simplify the choices for the user. The fundamental scale used for these comparisons is presented in Table 1. Each pair-wise comparison enters together with its reciprocal in an n x n matrix, where n are the amount of options. When the table is filled, the normalized eigenvector of the matrix is calculated to provide the resulting priorities (weights) for the options. Different weights to multi-disciplinary criteria can lead to the most acceptable compromise between different subjective perspectives. While weighting between criteria are, per definition, subjective and require only high-level expertise, grading can be based on facts and requires more detailed insight into the topic. For the trade-off of the housekeeping bus it was chosen to derive the criteria and setup a grading table for the options per criterion between the authors of this paper and reviewed by several staff members at TU Delft with data bus experience. For the weighting between all criteria and the scoring of some criteria, the community is involved in the AHP using a questionnaire as elaborated in section 3.5.

Table 1. Fundamental Scale for Pairwise Comparisons in AHP (Saaty, 2008)*schoe*

Intensity of Importance	Definition	Explanation
1	equal	Two elements contribute equally to the objective
3	moderate	Experience and judgement moderately favor one element over another
5	strong	Experience and judgement strongly favor one element over another
7	very strong	One element is favored very strongly over another, its dominance is demonstrated in practice
9	extreme	The evidence favoring one element over another is of the highest possible order of affirmation

3.2 Derivation of Trade-off Criteria

In Figure 4 a first derivation of trade-off criteria is presented, which come from the design targets described in section 0.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

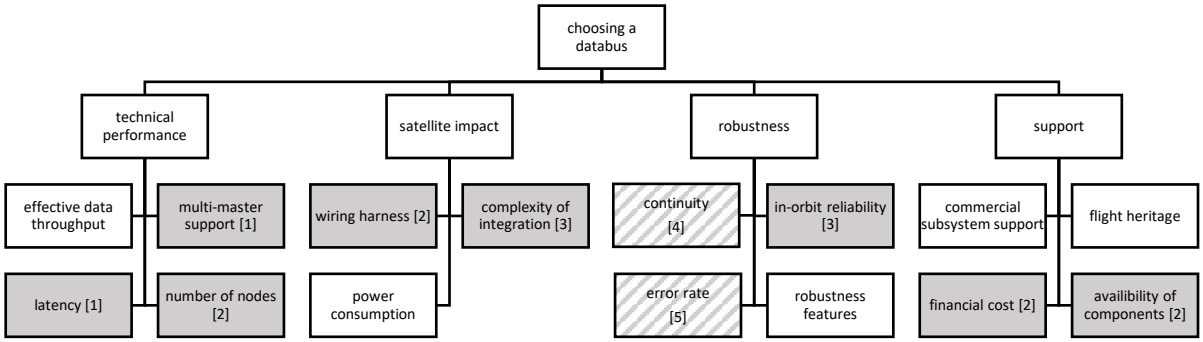

Figure 2. Derivation tree for criteria (grey/patterned boxes are omitted after theoretical/practical analysis)

Some of the identified criteria are omitted after theoretical analysis. These boxes are marked solid grey in Figure 2 and the number between brackets refer to the following reasons:

1. These criteria are not considered to be very important. A housekeeping data acquisition and commanding cycle in the order of 1 – 10 Hz, managed by the Onboard Computer as master, is a typical approach (Bouwmeester et al., 2007) that works very well and does not require low latency or multi-master support.
2. The difference between the data bus options for these criteria are considered to be too small or out of scope. RS-485 supports 32 nodes and the others even a few hundred. RS-485, CAN and I^2C require 2 wires and dI^2C just 4. For all busses, the required integrated circuits are widely available from different manufacturers and are all very low in cost (a few € / US$).
3. There is no good metric or data available for these criteria. For complexity of integration, there is too limited community experience for RS-485, dI2C and CAN to aggregate subjective input. Sufficient statistical input for these busses is also missing for in-orbit reliability, which would give I^2C an unfair disadvantage (Bouwmeester et al., 2017).

As a next step, initial laboratory tests (see section 3.4) have been performed to discover if the derived criteria can deliver appropriate results which can be used for comparison with a reasonable amount of effort. This lead to a further reduction in criteria after practical analysis, for which the boxes are marked patterned grey for the following reasons:

4. Continuity as criterion refers to the ability of the data bus to operate continuously with bus lockups or other events which cause temporary unavailability of the bus. The chosen metric for this is the amount of disruptive events per time unit, in which less than once per 24 hour would receive the highest grade. In the initial tests all four data busses did not show any such disruption, even when subjected to electromagnetic interfere (see next point).
5. Error rate as criterion refers to the number of (bit) errors per number of transactions or bits. The chosen metric was the Packet Error Rate (PER) which could be discovered by a check of the CRC in each transaction. A packet error would indicate one or more bit errors within the transaction. A PER of less than one-in-a-thousand would receive the highest grade. All four data busses were tested for about 30,000 transactions each. In ambient conditions none of them showed packet errors. Tests have also been performed at high computational load on the microcontrollers (continuously calculating pi) and when the microcontrollers receive interrupts (up to 1000 Hz with high interrupt priority). In all those tests, no packet errors have been detected. Finally, tests have been performed by injecting simulated Electro Magnetic Interference (EMI). First by a direct injection of white Gaussian noise on the bus

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

lines with capacitive coupling and a signal generator. All data busses withstood a noise injection up to 0.8 V RMS without any packet errors, but it must be noted that the peak-to-peak voltage levels generated by the used signal generator are, in this case, already beyond 10 V. This is significantly higher than the signal reference level of 3.3 V used by all data busses and beyond the electrical specification of their integrated circuits. Only at even higher noise levels, the busses showed packet errors and lock-ups. Lab experiments with a spare model of Delfi-C^3 and subsystems of Delfi-n3Xt (specifically the reaction wheels and magnetorquers) showed noise levels below 1 V. These satellites are not representative for all CubeSats and PocketQubes, but show that a sample selection of a few subsystems is not appropriate to identify EMI sources which do results in disruptions and communication errors. Other tests were performed to simulate power transients on lines with switching currents of several amperes, including in-rush currents of several tens of amperes. In all cases, there were no packet errors discovered. After several experiments it became clear that all busses are resilient to a significant amount of noise. Still, there is insufficient knowledge of EMI levels, characteristics and test methods which would be appropriate to simulate a wide scale of PocketQube and CubeSat configurations including more "exotic" components (e.g. pulsed plasma thrusters) within a reasonable amount of effort. It is therefore decided to omit test-based inputs for error rates and only focus on inherent robustness properties of the data busses themselves.

The experiences with the test setup are not in line with the in-orbit experiences with the I^{2}C data bus as described in section 1.1. During the development of the test setup and even the initial EMI testing, bus lock-ups and significant errors appeared on all tested busses. This resulted in the discovery of several flaws in the software drivers of the test setup which have been corrected appropriately. The test setup used for this paper is based on all the same microcontrollers and the software is extensively debugged, which is different than for Delfi-C^3 and potentially also for other flown CubeSats. In the specific example of Delfi-C^3, it was found that the clock speed of the microcontrollers, the I^{2}C software drivers and differences between the I^{2}C hardware drivers within the microcontrollers have caused disruption and significant error rates (Cornejo et al., 2009). It is expected that I^{2}C problems on Delfi-C^3 could have been solved before launch but would have required extensive testing, debugging of software and even changes to the hardware. The experiences show that in-orbit experiences cannot directly be projected to the intrinsic reliability of a data bus and that a fair comparison on reliability can only be performed if the both hardware and software are extensively tested and corrected for development errors and/or inadequate choices for relevant components.

The remaining criteria are worked out further and for some sub-criteria are added. Figure 3 presents the final trade-off criteria tree for choosing a data bus.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

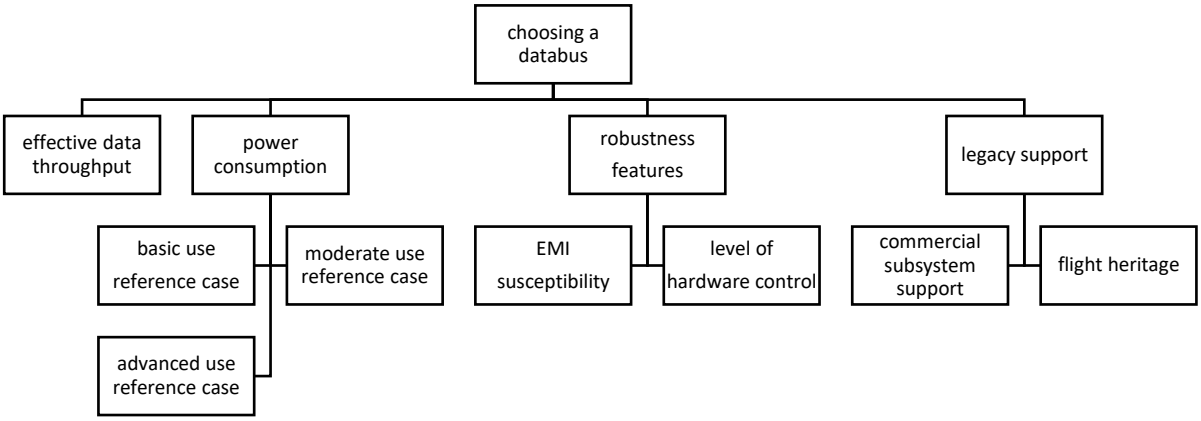

Figure 3. Final criteria tree for trade-off

Effective data throughput refers to the maximum amount of data which can be transferred over the bus from the master (OBC) to the slaves and back. It is the sum of all message content over the bus, excluding addressing, protocol overhead and timing delays.

The power consumption of the linear data bus is dependent on the number of nodes and the data throughput. As this may vary between missions, three reference use cases for the linear data bus have been defined:

Basic: a satellite with 5 subsystem nodes with a data and command cycle of 1 Hz. Payload could be a very low data rate sensor or a technology demonstration of (part of) a subsystem.

Moderate: a satellite with 9 subsystem nodes with a data and command cycle of 1 Hz. Payload could be similar to the basic case or could be sophisticated using dedicated point-to-point data bus(ses) as depicted in Figure 1.

Advanced: a satellite with 9 subsystem nodes at a relatively high data rate compared to the basic and moderate case. The high data rate can be attributed to a significantly higher data and command cycle and or a payload with moderate data rate which does not yet justify a dedicated point-to-point data bus. The effective data rate is fixed to approximately 250 kbit/s for this case, which was expected to be supported by the four chosen options.

The robustness features are EMI susceptibility and level of hardware control. The best attribute to judge EMI susceptibility on, based on the four options, is the difference between non-differential (I^2C) and differential (dI2C, RS-485 and CAN), where the latter is generally less susceptible due to common mode noise rejection. Testing under normal conditions did not show any errors including for regular I^2C. More intense EMI environments are unknown, so there is no quantitative metric based on value input possible for this criteria. It is therefore chosen to ask the community on their judgement, using the fundamental scale of AHP to determine the relative grades. For the level of hardware control, pairwise comparisons between three levels have been used:

- large part of the data protocol and potential error detection and failure handling needs to be implemented in the software (RS-485)
- a hardware controller for the full data protocol, but where the potential error detection and failure handling needs to be implemented in the software (I^2C & dI^2C)
- a hardware controller for the data protocol including internal error detection, correction and failure handling (CAN)

RS-485 required the full data protocol and any software error detection and correction to be

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

fully implemented in software. The UART and the differential driver only provides the physical layer. This means that the microcontroller needs to allocate relatively the highest amount of resources to the data bus and potential software bugs or interrupt/state control within the microcontroller could more easily lead to anomalies on the data bus compared to hardware control. I^2C and dI^2C do have the data protocol defined and implemented in the hardware controller. This will offload the microcontroller and is less prone to software bugs. CAN even has error detection and correction included in the hardware controller, which would make it most robust in this respect. However, the statements above are only true if the hardware controller has no flaws in the state-machine. Practical experience with I^2C shows that this is not always the case (Cornejo et al., 2009) and the high amount of bus lockups experienced by developers in orbit (Bouwmeester et al., 2017) may be an indication of a larger problem. Given the high degree of subjectivity in this matter, grading for this criterion is again based on the community judgement in pair-wise comparisons.

Finally, the legacy support of the data busses are taken into account. One sub-criterion is the commercial subsystem support. The rationale is that, of all available commercial subsystems, one can more easily and quickly adopt the wiring interface if the data bus is already supported. Alternatively, one can use a relatively simple interface-to-interface connector for the new proposed electrical interface standard compared to a situation where the subsystem does not yet support this data bus. The second sub-criterion is the flight heritage, which is based on the results of a survey performed on CubeSats (Bouwmeester et al., 2017). Both criteria are value based, but in terms of relative grades they do not have a direct technical impact on the satellite such as the effective data rate or power consumption. Therefore the community is asked to define the grading range for each.

3.3 Grading for Final Criteria

As next step, the grading is determined for the trade-off, which is presented in Table 2. The grade ranges for criteria using quantitative input are based on internal experience as well as studies of worldwide CubeSats (Bouwmeester and Guo, 2010).

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

Table 2. Grading table for linear housekeeping data bus

Criterion	Grade
Effective Data Throughput	$= \frac{D}{1000\ kbit/s}$ where D = effective data throughput *if D < 6 kbit/s → reject option*
Power Consumption	$= 1 - \frac{P}{T}$ P = total power consumption for data bus T = threshold For PocketQube / CubeSat: T_{basic} = 50 mW / 200 mW $T_{moderate}$ = 100 mW / 400 mW $T_{advanced}$ = 200 mW / 800 mW *if P > T → reject option*
Robustness Features	Fully AHP survey based, see section 0
Legacy Support	$= \frac{1 + (S-1) \cdot I}{S+1}$ S = AHP scale factor, see Table 1. $I_{COTS\ S/S\ support}$ = implementation rate fraction on commercial CubeSat or PocketQube subsystems in which a standard UART support counts half for RS-485 and regular I²C counts half for dI²C. $I_{flight\ heritage}$ = implementation rate fraction on CubeSats from survey (Bouwmeester et al., 2017).

The AHP method uses normalized grades and weights in which the individual grades for the options and the weights of the criteria need to add up to 1. Therefore, some of the grades from Table 2 need to be normalized before entering the next step of the trade-off. It also should be noted that community experience for CubeSats is also considered as input for PocketQubes as it involves flight heritage on very small satellites and public documentation on implementation lessons learned.

3.4 Housekeeping Data Bus Comparative Test Setup

This section describes the final test setup for the input for grading effective data throughput and power consumption.

The test setup comprises up to nine Texas Instrument's MSP432 microcontroller development boards. The MSP432 is a modern microcontroller which is chosen as the default controller for the Delfi-PQ PocketQube of TU Delft due to its low power over computational load ratio. The data bus specific hardware is placed on daughter boards which can be stacked on top of the development boards. A ribbon cable connects all boards. The power consumption is measured at the input power which is run to all boards by means of a high precision current meter. Before each test, the power consumption of each developments board is measured without the daughter boards. This value is subtracted from the measured power during the data bus tests. The complete setup is shown in Figure 4.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

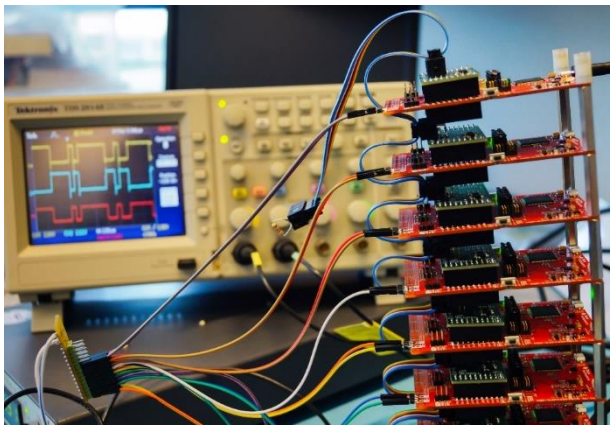

Figure 4. Test setup for data bus characterization

For I²C, the dedicated internal controller on the MSP432 is used and a data bus buffer is added per board. The circuit is represented in Figure 5. For differential I²C, the data bus buffer is replaced by a dedicated differential driver as shown in Figure 6. For RS-485, the UART of the MSP432 is used and a dedicated differential driver is added to the UART as shown in Figure 7. For CAN, both an external controller and a driver are required as shown in Figure 8. CAN is the only data bus under consideration which is not supported with an internal controller onboard the microcontroller chip. It has to be noted that there are some microcontrollers available with internal CAN controllers. This may positively influence the power consumption, but this will limit the choice of microcontrollers severely and may require major adaptations of existing subsystem designs. For all data busses, a list of potential components are selected which operate at 3.3 V level. From this list, the ones with the lowest power consumption according to the manufacturer specification is selected out of a list of options from different manufacturers. For all busses, bias and termination resistors are chosen following the recommended specification to ensure optimal behavior and noise rejection.

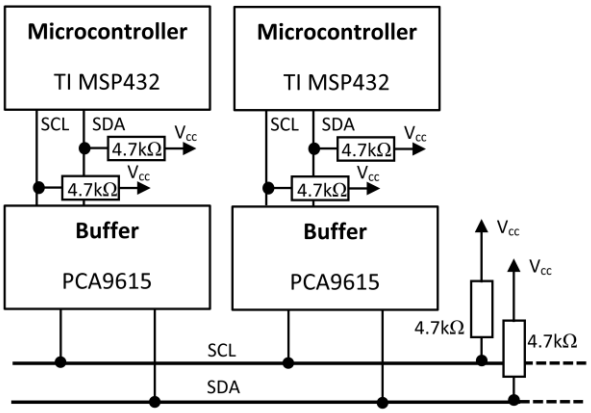

Figure 5. I²C circuit

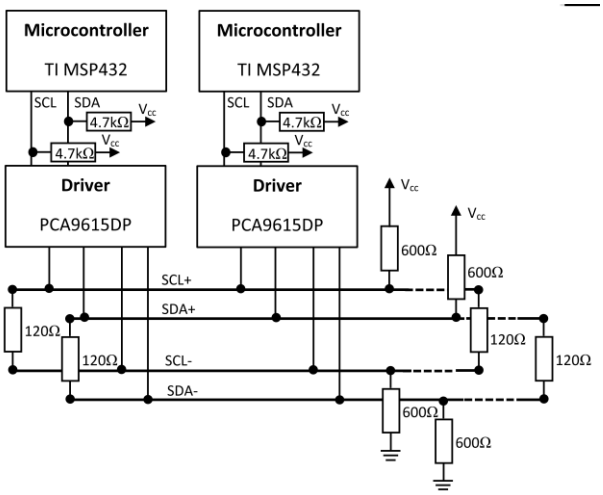

Figure 6. dI²C circuit

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

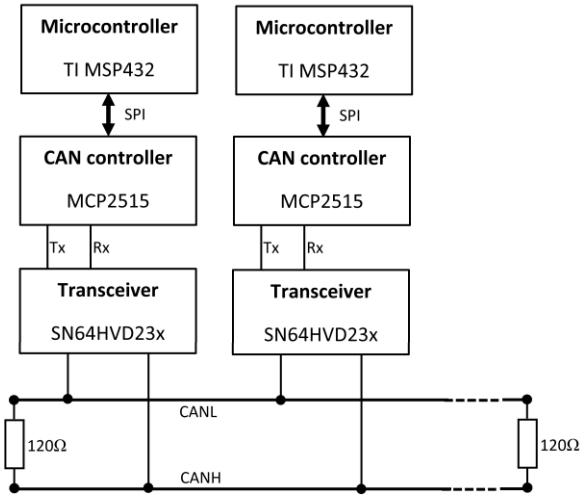

Figure 7. CAN circuit

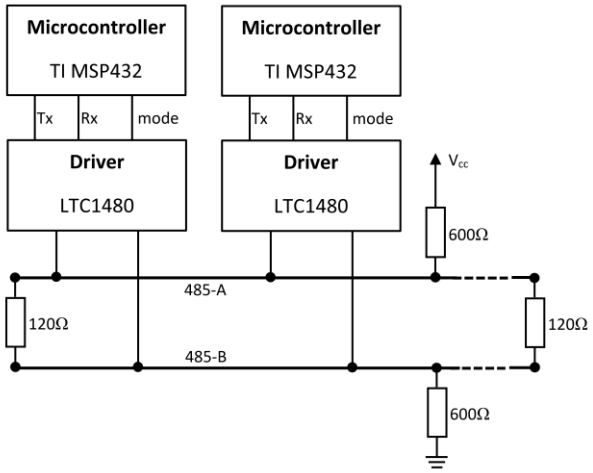

Figure 8. RS-485 circuit

For testing the power consumption and throughput efficiency, a reference case communication scenario has been established in Table 3 which is based on both the architecture and example provided in Figure 1. It is assumed that this standard communication set is cyclic at 1 Hz. The data packet size are based on experience with Delfi satellites and commercial CubeSat hardware. Large packets, not supported by a data bus (e.g. CAN), will be broken up in sequential packets.

Table 3. Reference communication set for linear housekeeping data bus

Source Node		Recipient Node		Size [bytes]
1.	OBC	3.	EPS	2
3.	EPS	1.	OBC	30
1.	OBC	4.	ADCS	2
4	ADCS	1.	OBC	120
1.	OBC	6.	GNSS	2
6.	GNSS	1.	OBC	30
1.	OBC	7.	propulsion	2
7.	propulsion	1.	OBC	10
2.	OBC	2.	H/K radio	2
2.	H/K radio	1.	OBC	10
1.	OBC	5.	payload	2
5.	payload	1.	OBC	10
1.	OBC	9.	data storage	2
9.	data storage	1.	OBC	10
1.	OBC	8.	P/L radio	2
8.	P/L radio	1.	OBC	10
1.	OBC	9.	data	250
1.	OBC	2.	H/K radio	250
		Total of node 1-5, 9 packets:		*428*
		Total of node 1-9, 18 packets:		*746*

While the reference communication set is a realistic representation of the architecture and subsystem structure provided in Figure 1, it does not apply for satellites with modest payloads that may not require dedicated payload data busses. Also, the frequency of 1 Hz is arbitrary and can be higher or lower depending on the specific needs of the mission. To determine the maximum effective throughput of the data bus, the set in Table 3 is simply looped continuously without pause. For satellites with payloads using relatively large data packets, the average overhead may decrease and thus the effective throughput maybe higher. It is, however, expected that the variations for different

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

scenarios will not lead to very large deviations in outcome and will be even more marginal, in a relative sense, between data busses.

3.5 AHP Questionnaire for Community Input

A questionnaire has been set up and sent in March 2017 to 36 and 453 members of the PocketQube and CubeSat community respectively. It has been decided to keep these communities separate, as the characteristics of these two different form factors are very different (in terms of volume, power, sophistication of payloads, flight heritage, etcetera). The questionnaire was sent out in March 2017 and had a response of 34 participants from the CubeSat community, representing 30 different development parties from around the world. Likewise, there were 15 participants representing 10 different development parties from the PocketQube community.

All questions provide input for the mutual weighting of sub-criteria followed by the main criteria in pair-wise comparisons using the AHP scale (see Table 1). Some of the final grades and all mutual weights are determined using the input and an Excel-based tool (Goepel, 2013) that calculates the AHP output.

4 Housekeeping Data Bus Results

4.1 Power Consumption

The test results on the power consumption of the data busses are presented in Figure 9 and Figure 10. The graphs shows the total power consumption of each data bus for the amount of bus nodes attached. The standard deviation between the four independent test runs for all test points is 6.5 mW. The confidence interval can be determined by:

$$\left(\bar{x} - z^* \frac{\sigma}{\sqrt{n}} \ , \bar{x} + z^* \frac{\sigma}{\sqrt{n}}\right) \quad (1)$$

where

$\bar{x}$ = mean
z* = confidence interval index

σ = standard deviation
n = number of measurements

The 95% confidence interval (z*=1.96) for the four test runs (n=4) is +/- 6.4 mW for the data presented in Figure 9 and Figure 10.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

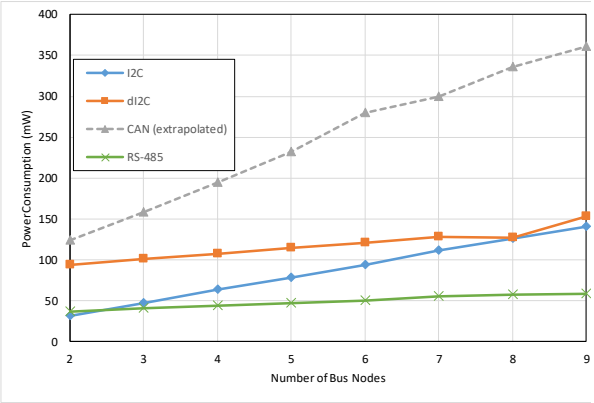

Figure 9. Power consumption for one communication set per second

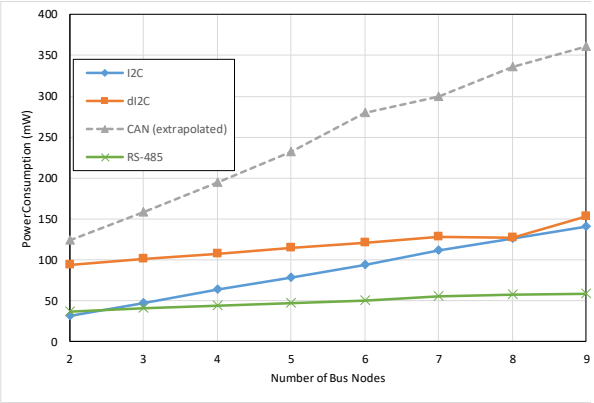

Figure 10. Power consumption for 250 kbit/s

For the trade-off, the input data are taken from the 5 node and 9 node points in Figure 9 and the 9 node points from Figure 10. The values are presented in Table 4. The reference use cases are described in section 0 and elaborated in section 3.4. The power consumption for 9 nodes at the maximum data throughput is also provided.

Table 4. Power consumption for the trade-off use reference cases

	Power Consumption [mW]			
Use Case	**I²C**	**dI²C**	**CAN**	**RS-485**
basic	52	36	139	9
moderate	95	63	268	11
advanced	141	153	362*	59
maximum	139	154	318	108

* extrapolated from maximum data rate of 136 kbit/s and idle consumption

The grades are calculated by entering the data from Table 4 into the grade equation in Table 2. As a next step, the grades have been normalized to the sum of one (required by AHP) and are subsequently multiplied by the calculated relative weights per participant following from the community survey. This yields individual priorities (grades) for the criterion of power consumption. For CubeSats, the mean weight of all participants are 0.29 for the basic, 0.31 for the moderate and 0.40 for the advanced use reference case. For the PocketQubes these are 0.42, 0.16 and 0.42 respectively. The priorities are presented in Figure 11 which shows a boxplot for the spread of individual priorities. The end of the legs show the minimum and maximum, the end of the boxes show the first and third quartile of all participants and the line in the middle shows the median. Additionally, the cross shows the mean of all participants and the dot shows the relative amount of participants for which the specific data bus received the highest priority.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

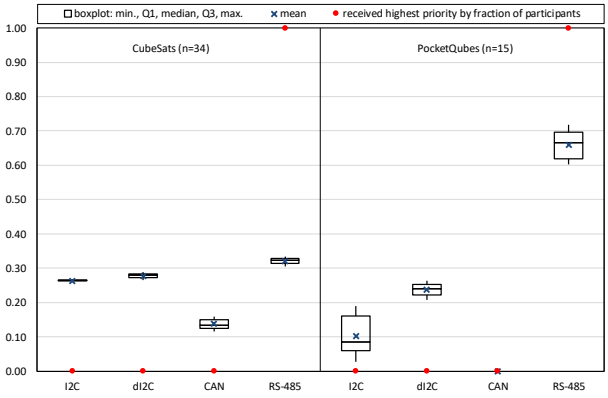

Figure 11. AHP priorities on power consumption

From Figure 11 it can be concluded that for PocketQubes, RS-485 has a clear advantage over the other busses. CAN, on the other end, does not meet the rejection threshold and should therefore be omitted as option for PocketQubes. Because of limitations of the AHP method, it still is included in the final trade-off with the grade for this criterion set to zero. For CubeSats, the spread of priorities for this criterion is significantly less, which can be explained by the higher reference power levels as presented in Table 2.

4.2 Effective Data Throughput

Table 5 provides the effective data throughput at the advanced reference case which is used for input of the trade-off. The initial grades based on Table 2 are normalized to calculate the AHP priority.

Table 5. Effective data throughput at advanced reference case

Data Bus	Baud rate of controller	Expected Data Efficiency	Measured Effective Data Throughput	Data Efficiency	AHP priority
I^2C	400 kHz	80%	248 kbit/s	62%	0.20
dI^2C	400 kHz	80%	258 kbit/s	65%	0.21
CAN	1 MHz	51%	136 kbit/s	14%	0.11
RS-485	1 MHz	79%	600 kbit/s	60%	0.48

I^2C, dI^2C and RS-485 both have a theoretical calculated data efficiency of about 80% for the communication set in Table 3. The measured efficiencies are lower, which can be attributed to the latencies of about 20% of total transaction time within the microcontroller of handling the data.

One of the reasons for the relatively low effective data throughput and also low data efficiency of CAN be found in the protocol overhead. A CAN frame with the maximum of 64 bits of message content is, in total, 114 bits (for the base frame format) including protocol overhead, so the efficiency is at best 56%. For a small 16-bit message, the total CAN frame is 66 bits, yielding an efficiency of 24%. Due to Non-Return-to-Zero (NRZ) encoding, bit stuffing is needed, which reduces efficiency up to 20%. The expected data rate in Table 5 is based on the communication set in Table 3 and 10% bit stuffing. Including the probable latency factor of the microcontroller, one would still expect an efficiency of approximately 40%. The best explanation for the gap between theory and test results are the latencies caused by the additional SPI interface between the microcontroller and the CAN controller. There is thus a potential gain in effective data throughput if internal controllers are used.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

The sensitivity of final trade-off for a theoretical improvement up to 400 kbit/s for CAN is investigated in section 4.5.

4.3 Robustness Features

For determining the priorities on the main criterion 'robustness features', the AHP community survey is used for prioritization of the sub-criteria. This is explained in section 0. The priorities are shown in Figure 12. CAN receives the highest priorities since it is a differential bus and has a high degree of hardware control. The mean relative weighting between the two sub-criteria is almost equal for CubeSats and PocketQubes, leading to a balance between I^2C and RS-485 and a slight advantage for dI^2C.

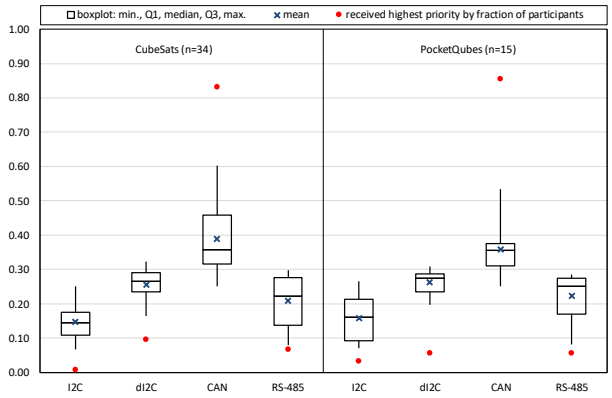

Figure 12. AHP priorities on robustness features

4.4 Legacy Support

The input used for the legacy support is presented in Table 6. For CubeSats, a wide survey of the market has been performed within this study with a large variety of commercial suppliers and (for each supplier) different subsystems. In total 56 different main physical subsystems coming from 23 different manufacturers have been selected. For the grade input, dI^2C receives 50% of result for I^2C support and whenever UART is mentioned instead of RS-485 explicitly, this is counted for 50% as well. The rationale is that the change from I^2C to dI^2C and generic UART to RS-485 require small modifications for which a major part of the legacy support is maintained. For PocketQubes, only 3 commercial systems were found. This is very low, making this a sensitive input for which the impact on the final result will be checked.

Table 6. Grade input data for data bus legacy support

	CubeSat flight heritage (n=56)	**CubeSat commercial subsystem support (n=52)**	**PocketQube commercial subsystem support (n=3)**
I2C	78%	40%	60%
dI2C	39%	20%	30%
CAN	5%	20%	0%
RS-485	4%	20%	10%

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

For this criterion, the grade input data is scaled to the AHP range as determined from the survey (see Table 2). The priorities for the legacy support are provided in Figure 13. I^2C receives the highest priorities, which can be explained by the input data. However, the levels of priorities are reduced in range compared to the input values as for both sub-criteria and both satellites form factors, the mean importance is rated moderate to strong. Some participants have given equal priority to each level of support, which is the reason that I^2C does not score 100% of the received highest priorities.

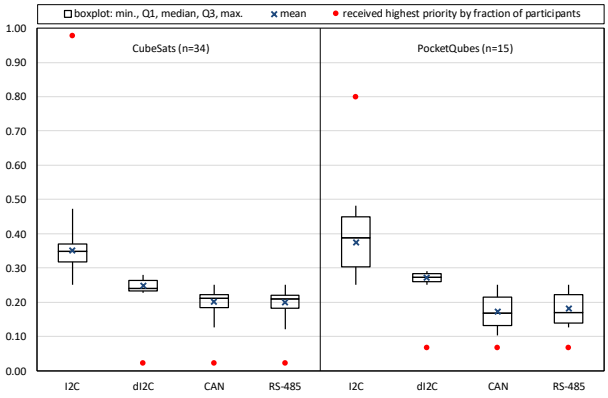

Figure 13. AHP priorities on legacy support

4.5 Final Trade-Off

Finally, the weights between the four main criteria are determined using the AHP community survey and provided in Figure 14. The relative priority of each criterion is multiplied by its relative weight and summed for each option, leading to the final priorities as provided in Figure 15.

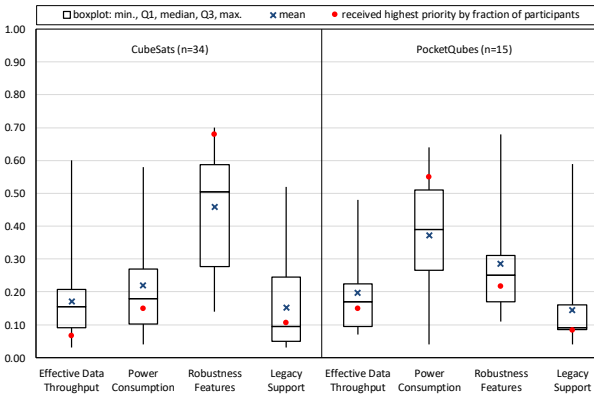

Figure 14. AHP weights of main criteria

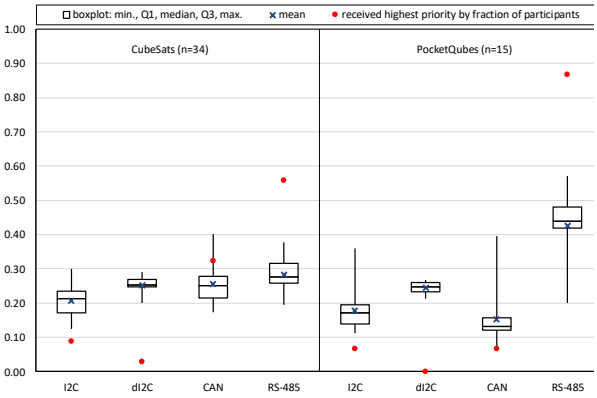

Figure 15. Final AHP priorities

For PocketQubes, RS-485 received the highest priority for 12 out of 14 participants. This is explained by the high relative weight for power consumption in combination with the high relative priority on this criterion for RS-485.

For CubeSats, RS-485 also received the majority of highest priorities (19/34), followed by CAN (11/34). For CubeSats the criterion 'robustness features' received a high weight, which is in favor of

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

CAN. Still, the combined weights on effective data throughput and power consumption and the relative good performance of RS-485 on these aspects swings the trade-off for many participants towards this data bus.

As mentioned in section 4.4, the trade-off is potentially sensitive to the limited available commercial subsystems for PocketQubes for this study. If the sub-criterion would be omitted, the final priorities only changes slightly in favor of CAN and RS-485 while the distribution of highest priorities over the data bus options remain the same.

As mentioned in section 4.2, the effective data throughput of CAN in the test setup has been found to be significantly lower than expected. If this data rate would be improved to a theoretical data rate of 400 kbit/s, CAN would receive the highest priority by 13 out of 34 participants for CubeSats, while RS-485 would drop to 16 out of 34 participants. For PocketQubes, there is no effect on the final outcome of highest priorities.

5 Electrical Power Distribution

In a previous study on the distribution of electrical power in CubeSats (Bouwmeester and Santos, 2014), the following conclusions and recommendations for a new interface standard were made:

- Limit the amount of supply voltages and fix the topology for all subsystems.
- Limit the amount of conversion steps needed.
- Fix the pin definitions such that incompatibility cannot occur.
- Fix the range of variable bus voltages, which is e.g. used by the battery.
- Use flex-rigid wiring in combination with side-mounted connectors to save board space.

The study concludes with two suggested options, of which the most simple and power efficient solution (based on the design targets in section 0) is chosen for the proposed interface standard in this paper. In Figure 16, a schematic overview of the power distribution is presented in which the unregulated battery bus is distributed via 4 or 8 configurable current protected switched outputs. Regulation occurs at the subsystems locally.

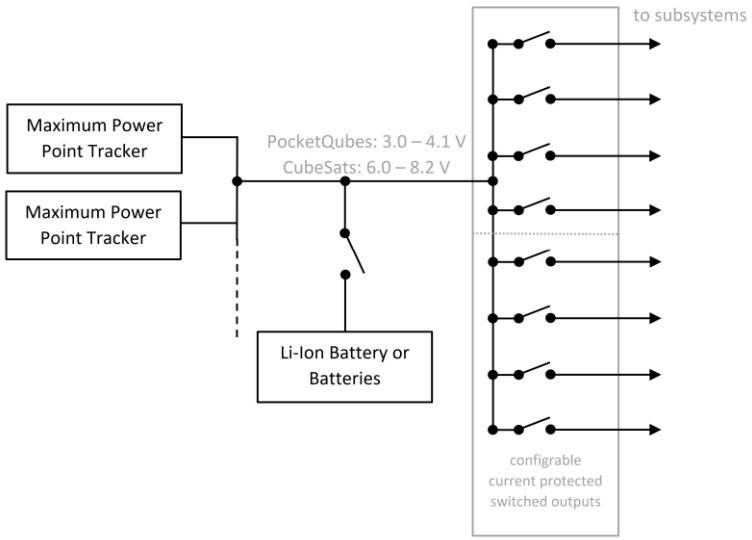

Figure 16. Power Distribution Schematic Overview

Single event upset and software state errors can lock up a data bus or halt the operations of the

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

OBC. In the current philosophy, subsystem redundancy concepts are omitted. The central EPS can solve some issues with a power cycle of the full satellite, either at a default fixed interval (e.g. once per day) or when it does not receive e.g. a repeating synchronization message for a while from the OBC. Still, such methods do not mitigate all errors, such as on the central EPS itself. A reset line from the primary radio receiver to the EPS is recommended. The radio receiver should be able to decode a reset tele-command and pull the reset line high. The line is pulled low by a resistor and a decoupling capacitor near the input at the central EPS unit. At the central EPS, the power of the EPS microcontroller and all distribution lines are taken down for a few seconds to enforce a true power cycle of all systems.

6 Proposed Electrical Interface Standard

Based on the trade-off results on the data bus and the analysis on the power distribution as well as the design targets stated in section 0, the simplest solution for an electrical interface standard is defined and presented in Figure 17. For the PocketQube, a 9 pin interface connector is defined (the first 9 pins in the figure) and is called PQ9. For CubeSats, a 14 pin connector is defined in similar fashion, by adding 4 power distribution lines, and is called CS14. The first nine pins of CS14 are similar to PQ9, but due to the different voltage range, not identical. However, since the power distribution requires local regulation, it is very well possible that the local DC-DC convertors can handle the entire input range from 3.0 to 8.2 V. This would create an opportunity to easily create a CubeSat version of a PocketQube system or to stack several of these PocketQube boards on a CubeSat motherboard.

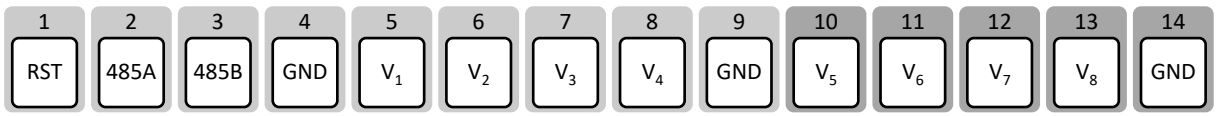

Figure 17. PQ9 (pin 1-9) and CS14 (pin 1-14) interface connector

Table 7. Pin allocation for PQ9/CS15 standard interface

Pin	Signal	Allocation
1	RST	system reset line (60 Ω to gnd)
2	485-B	RS-485 inverting signal
3	485-A	RS-485 non-inverting signal
4	GND	Ground
5	V_1	rec.: OBC (PQ: + Radio)
6	V_2	rec.: ADCS (PQ: + GNC)
7	V_3	rec.: propulsion
8	V_4	rec.: primary payload(s)
9	GND	Ground
10	V_5	rec.: radio
11	V_6	rec.: GNC
12	V_7	rec.: data storage & payload data transmitter
13	V_8	rec.: secondary payload(s)
14	GND	ground

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

In a previous study (Bouwmeester et al., 2017), a flex-rigid backbone in combination with side-mount connectors was suggested for wiring harness with main rationale to limit the amount of board space. However, since the number of pins selected in this paper is very low, such a solution would not be optimal in terms of board space. The final type of connectors and/or wiring harness chosen is a single row 2 mm pitched stackable pin header connection. These connectors are low in cost, available in different stack heights, sold by different manufacturers and proven in space since they are very similar to the PC/104 connector. The mechanical outline of the printed circuit boards for PQ9 and CS14 are shown in Figure 18 and Figure 19. A hardware example of PQ9 is provided in Figure 20. When comparing PQ9 to PQ60 it has about 15% of the pins and 30% of the connector footprint area. For CS14 compared to PC/104 this is 13% and 8% respectively.

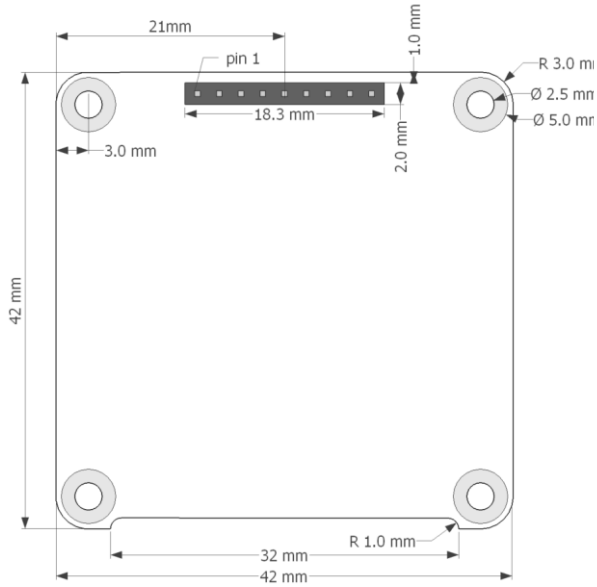

Figure 18. PQ9 printed circuit board outline

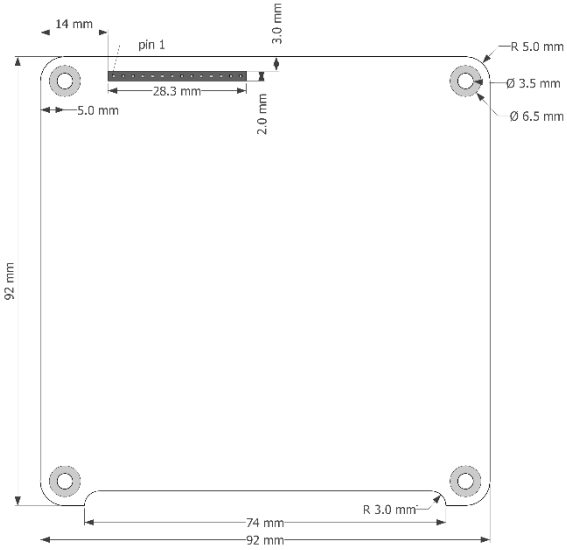

Figure 19. CS14 printed circuit board outline

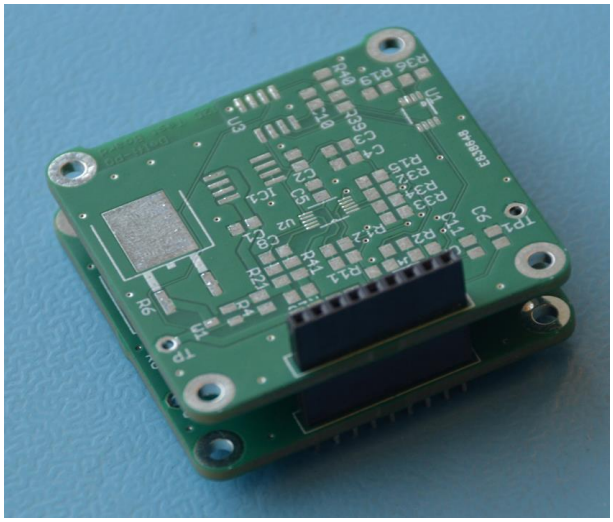

Figure 20. PQ9 PocketQube boards with stackable pin connector

7 Conclusions and Outlook

A proposal for an electrical interface standard for CubeSats and PocketQubes has been

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

established. The main target is towards a lean standard which meets expected future demands as opposed to existing versatile standards which exhibit the risk of incompatibility between subsystems from different developers.

Based on the defined set of selection criteria, community survey input and the AHP trade-off method, RS-485 is favored as housekeeping data bus for both PocketQubes and CubeSats. Tests results show that it outperforms I^2C, dI^2C and CAN in terms of power and effective data throughput. In terms of robustness features, it comprises differential signaling, but a low level of hardware control. In terms of legacy support is scores relatively low, but this is a criterion which can easily be improved in the future if the proposed electrical interface is adopted by multiple parties. For a future study it is recommended to test the RS-485 bus for very high data rates such that it can be used as a point-to-point payload data bus for demanding payloads (or RS-422, which is very similar in point-to-point configuration), data storage and high speed radio transmitters. Also, it is recommended to perform in-orbit tests with self-powered sensors over a wireless Bluetooth Low Energy connection to be able to reduce wiring harness to components which cannot be integrated in the internal stack of subsystems.

Power distribution can best be done by supplying the unregulated battery voltage over switched and protected lines to (groups) of subsystems. This limits the number of pins used and reduces conversion losses. Power protection features and duty cycling of subsystems to save power can be implemented at the central EPS unit. Together with the chosen data bus, this yields a 9-pin (PQ9) and 14-pin (CS14) standard electrical interface for PocketQubes and CubeSats respectively. PQ9 has only 15% of the electrical interface lines compared to PQ60, CS14 only 13% compared to PC/104. This saves in both cases significant board space, but more importantly leads to a very lean interface which with a lower risk for incompatibilities between physical subsystems. However, it comes at the cost of versatility and developers freedom.

An important assumption made in this study is that CubeSats and PocketQubes do not use redundancy for main subsystems. While the proposed interfaces do not prohibit this per se, the lack of a redundant data bus and the limited amount of power distribution lines make a true single-point-of-failure free design impossible. A follow-up study is recommend to investigate the impact on the overall reliability for these small satellites under these assumptions.

A topic not addressed in this paper is the development of (mega) constellations of very small satellites. Present day examples are the Flock CubeSats from Planet (Boshuizen et al., 2014) and the Lemur CubeSats from Spire (Hand, 2017). In relation to an electro-mechanical interface standard, it is expected that technical criteria are more important than community support. The rationale behind this expectation is that the main players have sufficient finances to develop many iterations of the spacecraft before the final mission and have the financial means to optimize the satellite and when necessary customize subsystems and even the interfaces to enhance the performance of the satellite. Next to this, the financial aspect of series production in relation to the electro-mechanical interface becomes important. The proposed PQ9 and CS14 interface standards can be implemented with just a few very cheap components (few Euros/Dollars) and assembly will take only a few minutes by an solder expert or can even be fully robotized.

Next steps are to define the data protocol for RS-485, the electrical characteristics of the reset line and to perform extensive testing with engineering models of PocketQube systems using the PQ9 interface. The final goal is to publicly release documentation on the new interface standards PQ9 and CS14.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

Acknowledgements

The authors would like to thank all participants in the AHP survey, all Delfi-PQ team members for their critical review and inputs of the trade-off process and Laurens Buijs for his contribution in testing the data busses. Part of this research is supported by the National Sustainability Program of the Czech Republic under grant LO401.

8 References

Amini, R., Gaydadjiev, G., Gill, E., 2009. Smart power management for an onboard wireless sensors and actuators network, in: AIAA Space 2009 Conference and Exposition.

Arruego, I., Rivas, J., Martinez, J., Martin-Ortega, A., Apestigue, V., De Mingo, J.R., Jimenez, J.J., Alvarez, F.J., Gonzalez-Guerrero, M., Dominguez, J.A., 2016. Practical application of the Optical Wireless communication technology (OWLS) in extreme environments, in: IEEE International Conference on Wireless for Space and Extreme Environments, WiSEE 2015. https://doi.org/10.1109/WiSEE.2015.7392981

Becnel, E., McAndrew, S., Strass, L., Walkinshaw, T., Worrall, K., 2015. PQ 60 Standard Document (v1.1).

Boshuizen, C.R., Mason, J., Klupar, P., Spanhake, S., 2014. Results from the Planet Labs Flock Constellation, in: 28th Annual AIAA/USU Conference on Small Satellites. AIAA, Logan, p. SSC14-I-1.

Bouwmeester, J., Amini, R., Hamann, R.J., 2007. Command and data handling subsystem for a satellite without energy storage: Delfi-C3, in: 58th International Astronautical Congress 2007. IAF.

Bouwmeester, J., Guo, J., 2010. Survey of worldwide pico- and nanosatellite missions, distributions and subsystem technology. Acta Astronaut. 67, 854–862.

Bouwmeester, J., Langer, M., Gill, E., 2017. Survey on the implementation and reliability of CubeSat electrical bus interfaces. CEAS Sp. J. 9. https://doi.org/10.1007/s12567-016-0138-0

Bouwmeester, J., Santos, N., 2014. Analysis of the Distribution of the Electrical Power in CubeSats, in: The 4S Symposium. ESA, Valetta.

Busch, S., 2015. CubeSat Subsystem Interface Definition (v0.3). Wuerzbrug.

Cornejo, N.E., Bouwmeester, J., Gaydadjiev, G.N., 2009. Implementation of a Reliable Data Bus for the Delfi Nanosatellite Programme, in: 7th IAA Symposium on Small Satellites for Earth Observation. IAA, Berlin.

de Boom, C.. W., van der Heiden, N., Sandhu, J., Hakkesteegt, H.C., Leijtens, J.L., Nicollet, L., Bouwmeester, J., van Craen, G., Santandrea, S., Hannoteau, F., 2011. In-Orbit Experience of TNO Sun Sensors, in: 8th International Conference on Guidance, Navigation and Control. ESA, Karlovy Vary.

Goepel, K.D., 2013. Implementing the Analytic Hierarchy Process as a Standard Method for Multi-Criteria Decision Making In Corporate Enterprises – A New AHP Excel Template with Multiple Inputs, in: Proceedings of the International Symposium on the Analytic Hierarchy Process. Creative Decisions Foundation, Kuala Lumpur, pp. 1–10.

Hand, E., 2017. CubeSat Networks Hasten Shift to Commercial Weather Data. Science (80-.). 357, 118–119. https://doi.org/10.1126/science.357.6347.118

Lawrenz, W., 2013. CAN System Engineering. Springer-Verlag, London. https://doi.org/10.1007/978-1-4471-5613-0

Leens, F., 2009. An introduction to I2C and SPI protocols. IEEE Instrum. Meas. Mag. 12, 8–13. https://doi.org/10.1109/MIM.2009.4762946

NXP Semiconductors, 2016. PCA9615 (Datasheet).

Poghosyan, A., Golkar, A., 2017. CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions. Prog. Aerosp. Sci. 88, 59–83. https://doi.org/10.1016/j.paerosci.2016.11.002

Riot, V., Simms, L., Carter, D., Decker, T., Newman, J., Magallanes, L., Horning, J., Rigmaiden, D., Hubbell, M., Williamson, D., 2014. Government-owned CubeSat Next Generation Bus Reference Architecture, in: AIAA/USU Conference on Small Satellites. AIAA, Logan.

Saaty, T.L., 2008. Decision making with the analytic hierarchy process. Int. J. Serv. Sci. 1, 83. https://doi.org/10.1504/IJSSCI.2008.017590

Schoemaker, R., Bouwmeester, J., 2014. Evalution of Bluetooth Low Energy Wireless Internal Data Communication for Nanosatellites, in: The 4S Symposium. ESA, Valetta.

Selva, D., Krejci, D., 2012. A survey and assessment of the capabilities of Cubesats for Earth observation. Acta Astronaut. 74, 50–68. https://doi.org/10.1016/j.actaastro.2011.12.014

Soltero, M., Zhang, J., Cockril, C., Industrial, H.P. a, Zhang, K., Kinnaird, C., Kugelstadt, T., 2010. RS-422 and RS-485 Standards Overview and System Configurations, Configurations.

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040

Annex: Table of figures

Published in 'Advances in Space Research', available at: https://doi.org/10.1016/j.asr.2018.03.040