VOL. 11, NO. 10, MAY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DESIGNING A LOW COST CUBESAT'S COMMAND AND DATA HANDLING SUBSYSTEM KIT

S. F. Sabri[1], S. S. Yuhaniz[1] and K. Kamardin[2]
[1]National Space Agency of Malaysia, Malaysia Space Center, Banting, Selangor, Malaysia
[2]Advanced Informatics School, University Technology of Malaysia, Kuala Lumpur, Malaysia
E-Mail: sfadlie@gmail.com

ABSTRACT

Commercial CubeSat kit is widely used to simplify CubeSat design process and shortened development schedule by providing standard and reliable hardware architecture. However, project team still needs to coordinate subsystems integration. It becomes more difficult for undergraduate students because they may not familiar with the components used in the kit. Price of the kits is also expensive, especially when converted to Malaysian Ringgit. Research is done to investigate suitable hardware and software architecture and design low cost Command and Data Handling subsystem (C&DH) kit with short learning curve. The kit will consist of hardware and software that is popular among university students and allow less coordination during development. This paper presents the current C&DH architecture for CubeSat and the selection of hardware and software components for the C and DH kit. The kit will be cost effective and convenient to be used by students.

Keywords: CubeSat, command and data handling, microcontroller.

INTRODUCTION

CubeSat is miniature satellite with size of 10 cm cube and a mass of up to 1.33 kg [1]. It aims is to promote a low cost platform but fast development approach in building a working satellite. Since the first CubeSat launched in 2003 until December 2013, more than 200 CubeSats have been launched to space [2]. From these CubeSats, more than 160 CubeSats were developed by universities worldwide.

Universities use CubeSats project as educational tool to train their students and improve the satellite-building capabilities of that particular university. Training of the student is as important as CubeSat's nominal mission [3].

Most of university will proceed with next CubeSat mission after successfully launched and operated their first CubeSat. The mission will be more scientific compared to the first one. This has shown how the program can be continuously run and improve the capability of university in satellite development. The knowledge and hands-on experience gained by students during the project will also benefit them in future.

Although the size of CubeSat is very small, its subsystems and development process is similar to bigger size satellite. Every subsystem requires team members from different background such as electronic engineering, electrical engineering, mechanical engineering, computer science, physic and etc. In university, it is difficult to gather multidisciplinary team members from different departments and faculties to work in a project for a long period. Besides that, development cost also quiet high to be borne by university.

One of the available options to solve these problems is by using CubeSat kit. The kit consists of CubeSat structure, development board, flight model board and sample codes for Command and Data Handling (C and DH) subsystem. Other subsystems can be bought separately based on requirements specification. By having the kit, development time and cost can be reduced.

However, the CubeSat kit price is still high, especially when converted to Malaysian currency. Table-1 shows the price of C and DH kit from four suppliers. The price is only for C and DH board and sample codes. Real Time Operating System (RTOS) and compiler will cost a lot more.

Table-1. Commercial C&DH kit price

Supplier	**Price** (MYR) Conversion Rate: USD$1 = MYR3.80
Gomspace	19,909
Pumpkin Inc.	30,438
Clyde Space	91,315
Tyvak Nano-Satellite Systems LLC.	20,925

Even though the kit is used, a lot of programming and integration need to be done. Students also have a problem when they are not familiar with the electronic components used in the kit. More time is needed to learn on how to use the component. This will delay development schedule.

CubeSat kit already provides fix connection type. Therefore, every subsystem's team must limit their design within this constraint to avoid problems during integration. C and DH subsystem is connected to all subsystems in the CubeSat. C and DH function is to receive, validate, decode, and distribute commands to other subsystems and gather, process, and format spacecraft housekeeping and mission data for downlink or use by an on-board computer [4][5]. Thus, C and DH team need to manage and coordinate the connection port to every subsystem and ensure everyone get what they are required. It is not easy to be done especially when the subsystems are developed by different universities at different geographical location.

Therefore, there is a need to have a C and DH kit that is easy to use, suitable for any subsystems design and cost less than market price. The kit should provide standard connection type to subsystems, use free RTOS and only need free or cheap compiler software.

This paper will explain the research done to identify suitable architecture and components to meet above objectives. The paper is organized as follows. In section 2, the hardware architecture and components used by other CubeSat's mission are discussed. In section 3, the software architecture and components are presented. The suitable architecture and components selected for C and DH kit are presented in section 4. Finally section summarized our work of this research.

HARDWAVE ARCHITECTURE AND COMPONENTS

CubeSat size limits available space for boards and components. There are two constraints need to be considered [6]. First, the dimension and mass of the C and DH must be small and light enough to be accommodated in a 10 cm X 10 cm area. Second, power consumption must be ensured to be as low as possible. C and DH normally takes around 10% of total power budget [7] or less than 100mW [8]-[9].

Complexity of the C and DH subsystem depends on team members' experiences. Most CubeSat project are prepared at institution of higher education or university level. Therefore, Keep It Simple Stupid (KISS) principle is widely used to prevent errors and problems on the design [10]. There are three types of architecture widely used in C&DH, namely: Centralized architecture, Parallel architecture and Distributed architecture.

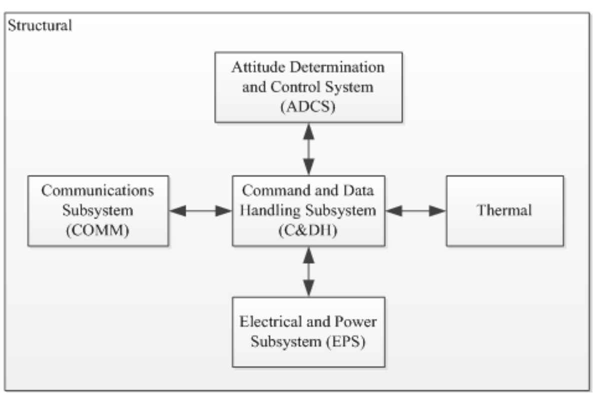

Figure-1. Centralized architecture.

Centralized architecture [11][15]is the most implemented architecture where every subsystem is connected to single microcontroller in C&DH as shown in Figure-1. Commercial C&DH kit are designed for this architecture. The advantages of centralized architecture are simplicity in its implementation and reduction in components' cost since only one processor is used. Power consumption can be low depending on the type of processor used.

In centralized architecture, developers need to manage the interface by ensuring interface is available and sufficient for all subsystems. If the interface is insufficient, software solution can be used to cover the inadequacy. However, programming codes can be more complex, and may lead to difficulty in error detection and rectification. Another difficulty that may arise with centralized architecture is single point failure. If the processor behaves abnormally or experience failure, this can affect the whole CubeSat operation.

Second type of architecture used for C&DH is parallel architecture [17]-[18]. Figure-2 shows the subsystem's connection in parallel architecture. In this architecture, both microcontrollers can operate simultaneously or only one microcontroller is operating at a time. If only one microcontroller operated, C&DH can swap to other microcontroller either by using power subsystem or watch dog module.

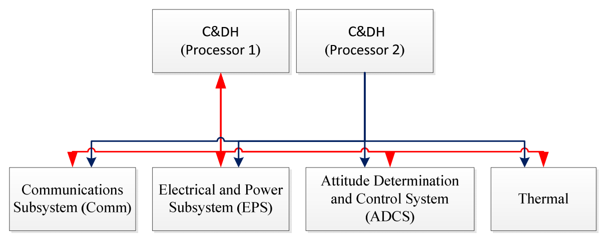

Figure-2. Parallel architecture.

Parallel architecture provides a good mechanism in preventing single point failure. Whenever one microcontroller fail, the alternate microcontroller shall be in charge and take over master's role. Two microcontrollers result in higher power consumption. Cold redundancy is a method in reducing power consumption since only one microcontroller is operational at one time. The disadvantage of the architecture is both microcontrollers may fault simultaneously due to same reason such as bug in codes which may prevent microcontroller to operate at nominal level. This bug may affect both microcontrollers since both microcontrollers are similar models, hence both have same codes. One of the solutions recommended for this problem is to use two different microcontrollers [18] at one time. However, developer needs to work with two set of programming codes for the same task, which may lead to more cost to be incurred and lengthy time frame.

Final architecture is distributed architecture. In distributed architecture, every subsystem has its own microcontroller. Figure-3 illustrates the architecture. The architecture is quite similar to centralized architecture whereby single point failure can occur at master or primary microcontroller, but in distributed architecture, the workload is greatly reduced [20][23].

 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

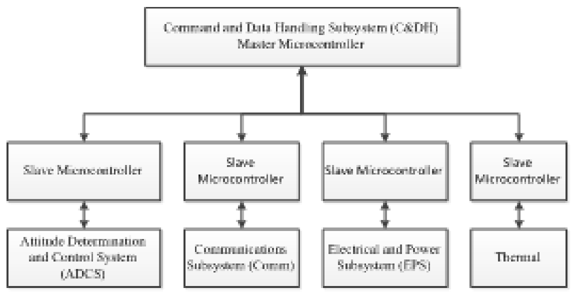

Figure-3. Distributed architecture.

Distributed architecture has two significant advantages. First, it allows developers to work separately and integrate their subsystems that comply with the standard interface provided. Second, it speeds up development time by reducing the workload of C&DH, and the codes' complexity.

On the other hand, this architecture requires more power and space in the CubeSat. Proper management is needed in ensuring that planned power and mass is used at optimal stage. Thermal subsystem needs to consider the location of microcontrollers in allowing high power controller's ability to dissipate heat with no difficulty.

From these architectures, the most important component is microcontroller. Developers are given freedom to use any microcontroller for any architecture. The crucial factor to be considered is the microcontroller's specifications. Parameters that need to be considered for microcontroller selection are Read Only Memory (ROM) size, Random Access Memory (RAM) size, bus interfaces such as USART, SPI, I2C and CAN, power consumption and support.

These parameters are depending on the complexity of the mission. Bigger size of ROM and RAM will allow bigger size programs and more complex processing. For CubeSat, less power consumption is considered better. Finally, another factor to be considered when choosing a microcontroller is the support. Most universities offering courses related to microcontrollers used these types of microcontrollers in their CubeSat so that student can get support easily [24].

SOFTWARE ARCHITECTURE AND COMPONENTS

C and DH software must be able to send commands to specific subsystem according to scheduled time. In nominal operation, C and DH shall execute task according to schedule. Most of the tasks are executed by other subsystems. Therefore, C and DH need to delegate the task accordingly and ensure right command is sent at the right time.

C and DH also must be able to collect data from subsystems and arrange it into a specific format. The data is then stored in a storage until the time where CubeSat receives command from ground to transmit the data. The format is important in ensuring the received data is understood by ground software.

Software architecture also can be classified to three types: Single loop architecture, Centralized architecture and Layered architecture. Single loop architecture is easy to be implemented [7], [25]. The loop continuously runs a task to monitor the telemetry, checks communication and retrieves the attitude readings from ADCS. Whenever C&DH detect anomaly, it will call appropriate subroutine to handle the situation. Programmers only need to arrange all tasks in a sequence. All situations must be taken into consideration in ensuring the codes are able to handle glitches accordingly. This architecture is only suitable for simple mission that only has small number of operation modes.

In centralized architecture, the software is developed in modules based on the subsystem in the CubeSat [26]. This implementation allows every module to be developed in parallel way, whereby each module is known as subsystem software and will interact with C&DH software. Subsystem software contains codes that directly interface with hardware in respective subsystem. C&DH codes will only handle overall CubeSat mission and processing commands from ground station.

The third architecture, divided the software to three layers [22]. The bottom layer is called as Hardware Access Layer (HAL), which controls the hardware and contains drivers for the peripherals used in the CubeSat. Second layer is distributed kernel layer. This layer is responsible for network and CubeSat's health monitoring, and task handling. Finally, the top layer is application layer where the mission is controlled by handling subsystems operation. This approach offers good reusability because it is independent from hardware implementation and it is also designed for distributed hardware architecture. The drawback of this approach is the software is more complicated compared to software for centralized hardware architecture.

Main component need to be considered in software design is Real Time Operating System (RTOS). RTOS is an operating system that handles multitasking applications to meet its deadline requirements. Besides getting accurate result, RTOS is also important in producing result within specified time. In C&DH, the usage of RTOS fully depends on software complexity. For system that implements single loop architecture, RTOS is not needed. However, in instances whereby system needs to handle a lot of tasks that are triggered by events or time, RTOS is required.

Salvo RTOS [9] [12] is an RTOS sold together with Pumpkin Inc CubeSat Kit. It is quite popular among CubeSat developers, especially those who are using C&DH kit from Pumpkin. Salvo PRO versions are priced at USD1250 for one year licence. Supports and helps are very limited because user community is small.

FreeRTOS [6] [27] is a free licence RTOS that requires low RAM and ROM footprint. ESTCube-1 used FreeRTOS to divide commands in three categories; low, high, and immediate. Command with immediate priority will be processed immediately, while others shall be processed accordingly based on their scheduler queues. The RTOS works well for CubeSat operation.

UWE CubeSat used a Linux version of RTOS called uClinux [25]. It is a free RTOS and provides same functionality as Linux operating system. uClinux exhibits its advantages by providing faster execution and shorter times for context switching. The only drawback is it requires huge ROM and RAM space due to Linux configuration.

Another important component is development environment for microcontrollers. It refers to development software used to write, compile, debug, and upload the programs to microcontroller. The software may only suitable for certain brand of microcontrollers. It is available for free or commercially licensed. For example, Keil uVision is suitable for ARM microcontroller. Its free version only allows codes that are less than 32kB in size. This differs with MPLAB IDE for PIC microcontrollers, which are available for free without any limitation. It is considered crucial to choose development environment that are free and provides full functionality and tools, in ensuring that development process are smooth and easy. Other factor to be considered is good technical support from the supplier and user communities, in ensuring that any problem encountered during development can be solved as soon as possible.

COMPONENTS SELECTION FOR C&DH KIT

The objective of this research is to design low cost C&DH kit with short learning curve. The kit consists of all components discussed in Section 3 and 4. Therefore, gap exists to identify suitable components combination. The selected components must not only be cost effective and convenient to use, but is also reliable for CubeSat mission. Cost effective means component's price is affordable for student's project. Convenient to use means user can get sufficient information and resources to use the components, either from supplier or users' community.

Distributed architecture is chosen as kit architecture because it allows flexibility in connecting C&DH and other subsystems. Power consumption may be an issue here because five microcontrollers are needed for CubeSat bus. However, with the advancement of technologies for solar cells and the usage of 3U structure, more power can be generated for CubeSat mission. Therefore, power consumption for five microcontrollers is considered not a burden to EPS.

Distributed architecture allows single data bus to be used to connect all subsystems. This has simplify the connection and reduces drivers' overhead during software development. Considering payload data size, SPI and I2C would be the best choice for data bus. SPI offers fast data rate and simple software configuration but requires seven wires connection for four subsystems. On the other hand, I2C only requires two wires but slave selection is done using software which can be considered risky for amateur programmer. Having taken into consideration, the knowledge level of users and effort in simplifying software's complexity, SPI is chosen for this kit.

Arduino Mega 2560 board with ATmega2560 microcontroller is chosen for C&DH subsystem and other subsystems will use Arduino Uno with ATmega328 microcontroller. The boards are prevalent for hobbyists and suitable for various types of projects. ATmega2560 is equipped with 256 kB flash memory and 8 kB SRAM, which is expected to perform better than most of microcontrollers listed in Table-2. It can also support RTOS and its development software, Arduino Software is free to use. ATmega microcontroller's family has been used on few CubeSats such as MCubed, NCube1 and NCube228.

Due to distributed hardware architecture, software architecture shall follow this method, and as such, shall be designed in distributed way. C&DH software is expected to be used to manage scheduling process, delegate commands from ground, collect subsystem's health data and handle communication between subsystems. Subsystem software will focus on components' level and can be developed independently from others. Interface Control Document (ICD) will be written as guidelines for every subsystem to connect to C&DH. ICD shall contain data format for command and telemetry to standardise the communication between subsystems. This architecture allows parallel development of every subsystem because codes for every subsystem are totally separated. FreeRTOS is chosen as RTOS for the kit because of its advantages; free licence and small size of ROM and RAM's requirements.

CONCLUSIONS

It can be concluded that the current solution for a CubeSat project is very costly, difficult to implement with coding errors and needs a high integration between the team members that leads to difficulty to work separately. The main components of developing a C and DH subsystem in a CubeSat have been reviewed and compared. From there, suitable selections of hardware architecture, microcontrollers, software architecture and RTOS were identified. This will help more Malaysian universities to involve actively in CubeSat development. This will ensure a continuous CubeSat project for the country and more students can be trained in satellite technology. This will provide a trained satellite engineers for Malaysia in the future.

FUTURE WORKS

Based on the output of this research, the kit will be developed. The performance of the kit will be evaluated and compared to the existing commercial C&DH kit. Besides that, cost analysis will be done to ensure the kit achieve its objectives.

ACKNOWLEDGEMENTS

This study was supported by a grant from UTM Research University Grant (RUG), under VOT number 08H23.

REFERENCES

[1] California State Polytechnic University. CubeSat Design Specification Rev. 13. 2014.

[2] M. A. Swartwout. 2014. Secondary payouts in 2014. 1-12.

[3] M. A. Swartwout. 2004. University-Class Satellites: From Marginal Utility to 'Disruptive' Research Platforms. 18th Annual AIAA/USU Conference on Small Satellites.

[4] M. Fadly, O. Sidek, M. A. Md Said. 2010. Development of Atittude Determination for Student Nano-Satellite INNOSAT. Regional Conference on Mechanical and Aerospace Technology. 518-1-518-9.

[5] S. D. Jong, G. T. Aalbers, J. Bouwmeester. 2008. Improved Command and Data Handling System for the Delfi-N3xt Nanosatellite. 59th International Astronautical Congress. 1-8.

[6] K. Laizans, I. Sünter, K. Zalite, H. Kuuste, M. Valgur, Tarbe, M. Noorma, 2014. Design of the fault tolerant command and data handling subsystem for ESTCube-1. Proceeding of Estonian Academy of Science. 63: 222-231.

[7] S. Waydo, D. Henry, M. Campbell. 2002. CubeSat design for LEO-based Earth science missions. IEEE Aerospace Conference Proceedings. 1: 1-435.

[8] Y. K. Chang, J. H. Park, Y. H. Kim, B. Y. Moon, M. I. Min. 2003. Design and development of HAUSAT-1 picosatellite system (Cubesat). International Conference on Recent Advances in Space Technologies. 47-54.

[9] R. Mahmood, K. Khurshid. 2013. Design techniques for on-board software of nano-satellites. 6th International Conference on Recent Advances in Space Technologies (RAST). 517-522.

[10] L. Alminde, M. Bisgaard, D. Bhanderi, J. D. Nielsen. 2005. Experience and Methodology gained from 4 years of Student Satellite Projects. International Conference on Recent Advances in Space Technologies. 94-99.

[11] B. A. Larsen, D. M. Klumpar, M. Wood, G. Hunyadi, S. Jepsen, M. Obland. 2002. Microcontroller design for the Montana Earth orbiting pico-explorer (MEROPE) CubeSat-class satellite. IEEE Aerospace Conference Proceedings. 1: 1-487-1-492.

[12] D. Schor, J. Scowcroft, C. Nichols. 2009. A command and data handling unit for pico-satellite missions. Canadian Conference on Electrical and Computer Engineering. 874-879.

[13] D. L. Bekker, T. A. Werne, T. O. Wilson, P. J. Pingree, K. Dontchev, J. C. Heywood. 2010. A CubeSat design to validate the Virtex-5 FPGA for spaceborne image processing. IEEE Aerospace Conference. 1-9.

[14] W. Guareschi, J. Azambuja, F. Kastensmidt, R. Reis, O. Durao, N. Schuch, G. Dessbesel. 2013. Configurable test bed design for nanosats to qualify commercial and customized integrated circuits. IEEE Aerospace Conference. 1-7.

[15] A. Shah, S. Afaq, U. Arshad. 2013. General system design of Cubesat in LEO for IR imaging. International Conference on Aerospace Science & Engineering (ICASE). 1-5.

[16] A. S. Rani, R. R. Santhosh, L. S. Prabhu, M. Federick, V. Kumar, S. Santhosh. 2010. A survey to select microcontroller for Sathyabama satellite's On Board Computer subsystem. Recent Advances in Space Technology Services and Climate Change (RSTSCC). 134-137.

[17] H. Ashida, K. Fujihashi, S. Inagawa, Y. Miura, K. Omagari, Miyashita, N. Kawai. 2010. Design of Tokyo Tech nano-satellite Cute-1.7+ APD II and its operation. Acta Astronautica. 66(9):1412-1424.

[18] P. Fiala, A. Vobornik. 2013. Embedded microcontroller system for PilsenCUBE picosatellite. IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS). 131-134.

[19] A. R. Aslan, A. Sofyali, E. Umit, C. Tola, I. Oz, S. Gulgonul. 2011. TURKSAT-3USAT: A 3U communication CubeSat with passive magnetic stabilization. 5th International Conference on Recent Advances in Space Technologies (RAST). 783-788.

[20] Lanclos, C. Pellerin. 2014. CAPE II: Design, development, launch, and on-orbit operation of an experimental picosatellite. IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC). 6C5-1 – 6C5-10.

[21] S. A. Asundi, N. G. Fitz-Coy. 2013. Design of command, data and telemetry handling system for a distributed computing architecture CubeSat. IEEE Aerospace Conference. 1-14.

[22] Mitchell, J. Rexroat, S. A. Rawashdeh, J. Lumpp. 2014. Development of a modular command and data handling architecture for the KySat-2 CubeSat. IEEE Aerospace Conference. 1-11.

[23] T. K. Priyambodo, A. E. Putra, M. Asvial, R. E. Poetro, G. Hendrantoro, E. Pitowarno, S. Kuswadi, G. S. Prabowo, A. Nugroho. 2014. IiNUSAT-1: The 1st Indonesian Inter-University Nano-Satellite for Research and Education. International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES).

[24] R. Canales, H. Bedon. 2010. First steps to establish a small satellite program in Peru. IEEE Aerospace Conference. 1-14.

[25] M. Schmidt, K. Schilling. 2008. An extensible on-board data handling software platform for pico satellites. Acta Astronautica. 63(11): 1299-1304.

[26] S. Johl, E. Glenn-Lightsey, S. M. Horton, G. R. Anandayuvaraj. 2014. A reusable command and data handling system for university cubesat missions. IEEE Aerospace Conference. 1-13.

[27] Sünter. 2014. Software for the ESTCube-1 command and data handling system. Doctoral dissertation, University of Tartu.

[28] L. T. Lumbwe, 2013. Development of an onboard computer (OBC) for a CubeSat. Doctoral dissertation. Cape Peninsula University of Technology.