Standardized, flexible interface design for a CubeSat bus system

著者	Turtogtokh Tumenjargal
year	2019
その他のタイトル	キューブサットバスシステムのための標準化・適応性インターフェース設計
学位授与年度	令和元年度
学位授与番号	17104甲工第485号
URL	http://hdl.handle.net/10228/00007820

Thesis of Doctor of Philosophy

“STANDARDIZED, FLEXIBLE INTERFACE DESIGN FOR A CUBESAT BUS SYSTEM”

by

Turtogtokh Tumenjargal

Supervisor:

Professor Mengu Cho

Laboratory of Spacecraft Environment Interaction Engineering

(LaSEINE)

Kyushu Institute of Technology

Kitakyushu 2019

Publications

Below is a list of journal articles, conference papers produced prior to publication of this work:

1. “Development Status of Joint Global Multi-Nation Birds CubeSat Constellation Project”, Proceeding of 第 60 回宇宙科学技術連合講演会講演集, Hakodate, Japan. (2016 年 9 月)

   著者 T. Tumenjargal, JGMNB Project Members, JGMNB Partners, and M. Cho

2. “Software configurable backplane interface design for a CubeSat bus system”, Proceeding of 第 61 回宇宙科学技術連合講演会講演集, Niigata, Japan. (2017 年 10 月)

   著者 T. Tumenjargal, Y. Tokunaga, E. Dashdondog, G. Maeda, S. Kim, H. Masui, and M. Cho

3. “Design and development of a reconfigurable backplane board for 1U CubeSat”, International Workshop on Lean Satellite, Kitakyushu, Japan. (2018 年 1 月)

   著者 T. Tumenjargal, S. Kim, Y. Tokunaga, S. Yoshida, T. Handa, H. Masui and M. Cho

4. “CubeSat bus interface with Complex Programmable Logic Device”, ACTA ASTRONAUTICA, (2019 年 4 月)

   著者 T. Tumenjargal, S. Kim, H. Masui, M. Cho

Abstract

Since the 2000s, small satellite launches have increased rapidly each year and the number of players in this field is strongly linked to the popularity of the CubeSat standard around the globe. Highlights of its achievements are often the compatibility of launches via a standardized deployer (i.e. POD), shorter development times and lower costs than conventional large satellites. CubeSats are not just popular instruments for educating students in space research and engineering, but also enable us to demonstrate challenging technologies in a cheaper and quicker way and carry out scientific research in the field. But the success of CubeSat's mission often fails. Improvements in reliability and prevent poor workmanship are necessary.

The CubeSat standard enabled the small satellite market to expand enormously. In fact, a modular spacecraft deployer which can be attached to many different launch vehicles as a secondary payload was the key technology for the CubeSat Standard. To date, only external CubeSat interfaces, especially the mechanical interface, have been standardized. CubeSat needs a standardized internal interface to take advantage of the modularity. It will contribute to cost reduction and development time. One key to cutting costs and delivery time is a standardized internal interface for different CubeSat missions. In three CubeSat projects at the Kyushu Institute of Technology in Kyutech, a backplane interface approach, proposed as UWE-3 by the University of Würzburg in Germany, has been implemented to reduce the time for development and assembly. The backplane approach also helped to reduce the risk of harnessing faults. In order to satisfy the mission requirements, however, modifications to the proposed standard interface board were necessary for each CubeSat project.

The thesis proposes a new idea of a Software-Configurable Bus Interface (SoftCIB) with a backplane board to obtain more flexibility, particularly for data connections. Instead of hardware routing, a Complex Programmable Logical Device (CPLD) was used to reprogram the bus interface on the PCB. The following advantages will be offered by the standardized backplane interface board: (1) less harness, (2) ease of assembly and disassembly (3) compatible with different CubeSat projects and (4) flexible for routings. We can use the SoftCIB again to reduce the cost and development of the interface boards, rather than designing and making new interface boards for new CubeSat projects. Various projects have various payloads for missions and interface requirements. The high flexibility of SoftCIB's interface allows one to select either the same or a different subsystem board such as an OBC or EPS.

A functional test with a breadboard module validated the concept. A radiation test has shown that the selected CPLD is strong enough to maintain total ionization doses in low Earth orbit of more than 2 years. The system level verification has been carried out in the engineering model of the BIRDS-3 project at Kyutech.

Acknowledgment

I would first like to thank my supervisor Prof. Mengu Cho. His questions, advice and recommendations are always thoughtful and helpful. Meeting with Cho sensei always seems not enough, not because of short time, but because of worth time. He charges my brain every time we meet. I am proud to be one of his ordinary students.

Thanks to my co-supervisors, Prof. Keiichi Okuyama, Prof. Kenichi Asami and Prof. Kazuhiro Toyoda for giving me valuable advice on my research. Thanks to Maeda sensei, Dr. Masui, Dr. Kim, Dr. Kateryna, Tsukinari san, Shirakawa san, Kawano san, Yamauchi san, Professor Ulam-Orgikh from NUM and other staff and professors at LaSEINE. My work and life in Japan would be more difficult without the help of every single one of you.

I should also like to thank the BIRDS-1, BIRDS-2, BIRDS-3 and SPATIUM-I project members for promoting a standardized CubeSat interface board and facing the real challenges of practical implementation of a backplane approach. I would like to acknowledge Dr. Stephan Busch and Prof. Klaus Schilling of the University of Würzburg for their useful discussions about backplane-based CubeSat design. Also, without the support of Prof. Takamiya of the Kyoto University Research Reactor Institute, and Prof. Iyomoto of the Center for Accelerator and Beam Applied Science at Kyushu University, the radiation tests would not have been done.

The research is partially supported by the “Coordination Funds for Promoting Aerospace Utilization” from the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT).

I would like to express my gratitude to the Mongolia-Japan higher Engineering Education Development (MJEED) project for covering the expenses for my PhD study. I would like to thank the president of the National University of Mongolia, Dr. R.Bat-Erdene, professor D.Ulam-Orgikh, professor R. Tsolmon, professor B. Ochirkhuyag, Dr. T.Begzsuren, Mrs. Sh. Mendbayar, Dr. B.Suvdantsetseg, Ms. N.Enkhjargal and others who support the foundation of space technology in my home country.

I want to thank my friends, D. Erdenebaatar, D. Amartuvshin, U. Tuguldur, B. Battuvshin, S. Purevdorj, N.Oyungerel, E.Namkhainyambuu, B.Dondogjamts, Adebolu Ibukun, Tejumola Taiwo, Apiwat Jirawattanaphol, Dima Faizullin, Atomu Tanaka, Hiroshi Fukuda, Rafael Rodríguez León, Sidi Asal, Isai Fajardo Tapia, Juan José Rojas, Abhas Maskey, Mariela Rojas Quesada, Rodrigo Cordova, Necmi Cihan, Bonsu Benjamin, Ernest Matey, Quansah Joseph, Nakamura Naoki, Tokunaga Yasuhiro, Raihana S.I. Antara, Abdulla Hil Kafi, Maisun Ibn Monowar, Reuben Jikeme Umunna, Joven Javier, Syazana Basyirah, Yeshey Choden, Adrian Salces, Hasif Azami, Cheki Dorji, Kiran Pradhan, Yasuhiro Tokunaga, Daiki Yamaguchi, Tomoki Uemura, Withanage Dulani Chamika, Tharindu Lakmal Dayarathna, Pooja Lepcha, Yuta Kakimoto, Makiko Kishimoto, Yuji Sasaki, Hari Ram Shrestha, Rahmi Rahmatillah, Syahrim Azhan, and others, who was the important part of my study and life in Japan.

Last but not least, I would like to thank my parents and my lovely wife for their support and making this study possible.

Table of Contents

1. Introduction

This chapter describes background and motivation for this research about standardized, flexible interface design for a CubeSat bus system. Presents a vision of this work and the objectives to be achieved. An overview of the thesis is also described in this chapter.

1.1. Research motivation

Since the 2000s, small satellite launches have increased rapidly each year and the number of players in this field is strongly linked to the popularity of the CubeSat standard [1] around the globe [2], [3]. Highlights of its achievements are often the compatibility of launches via standardized deployers (i.e. POD), shorter development times and lower costs than conventional large satellites. CubeSats are not just popular instruments for educating students in space research and engineering, but also enable us to demonstrate challenging technologies in a cheaper and quicker way and carry out scientific research in the field [4], [5]. But many CubeSats have failed to achieve its full mission. The great majority of the failed missions of CubeSats projects were run by universities [2]. Improving reliability and preventing the workmanship error are necessary [6], [7].

Only 17.4% of total 288 CubeSats have achieved their full mission success, 25.7% achieved partial mission success in the past from 2000 to 2015, according to [7]. The reason behind the CubeSat mission failure statically studied and explained in [2]. The vast majority of the failed CubeSat missions have been university-led projects, a first time CubeSat developers or mainly built by students who have low resources, a strong willingness to try risky missions. The author of [2] concluded that the major reason behind the failure is lack of integration and system level testing on the ground. Furthermore, the test is essential to demonstrate the whole operation sequence “as you fly”, also called end-to-end test. In order to find out the problems and make CubeSats ready to fly, developers need to spend little time on the development compare to integration and system level functional testing.

The issue, low rate of mission success of CubeSat class satellites, draws the attention of not only universities and research institutes, but also commercial companies and government agencies. In the USA, a group of people from leading aerospace companies investigated the causes of CubeSat mission failure and produced recommendations for improving mission success of CubeSats [8]. The research team questioned CubeSat developers in academia, industry, and government-funded research centers to discover reasons for poor success rates of CubeSat missions. They collected data from 23 organizations, analyzed, and generated 8 recommendations on how to improve mission success of CubeSats. Valuable information such as lessons learned during the development and hypothesis of possible failures were collected. However, the root causes of the on-orbit anomalies have not clearly determined in most of the cases. That recommendation addressed to management aspects of CubeSat development. More importantly, recommendation no.1 suggest that define the scope, goals, and success criteria at the beginning of the program. In other word, allow fewer changes as much as possible during the development because even tiny changes ultimately led to costly failures. Also, recommendation no.2 states that CubeSat developers need to limit time for development and spend more time (1/3 to 1/2 of the overall schedule) for integration and testing because of time constraint. In recommendation No. 4, they suggest simple and robust design because of the most CubeSat projects lack time and resources to fully analyze and test complicated systems to mitigate risk. This shows that there are needs to reduce the development of the CubeSat project. Other important points in these recommendations are: risk-based mission assurance, team experience, spare components, minimum mission assurance tests, and a performance check for purchased components.

The CubeSat standard enabled the small satellite market to expand enormously in the past two decades. More than 700 CubeSats already launched into space [9], [10]. In fact, a modular spacecraft deployer which can be attached to many different launch vehicles as a secondary payload was the key technology for the CubeSat Standard. To date, only external CubeSat interfaces, especially the mechanical interface, have been standardized. CubeSat needs a standardized internal interface to take advantage of the modularity. It will contribute to cost reduction and development time. One key to cutting costs and delivery time is a standardized internal interface for different CubeSat missions.

In this scope, several attempts at achieving that have been made and the most well-known interface used for CubeSats is the PC/104. Which have the same form factor of PC/104 embedded computer standard, but modified signals buses for CubeSats. In a recent survey [11], however, it was concluded that the PC/104 has design issues, namely that a smaller connector with fewer pins is needed. A further problem with the PC/104 interface is the difficult procedure when removing or replacing a central board (of a sandwich architectural design). To reach the targeted board all the top or bottom boards must be removed.

The University of Würzburg in Germany has developed and demonstrated a backplane interface board for a 1U CubeSat called UWE-3 [12], [13]. The time required for assembly and disassembly is much shorter than that for a PC/104, as there is very little need for wire harnessing. Similar architecture to UWE-3 was implemented in the Joint Global Multi-Nation Birds Project at the Kyushu Institute of Technology (Kyutech) known as BIRDS-1[14]. BIRDS-1 is a constellation of five 1U CubeSats. All five satellites have an identical design.

However, the standard UWE-3 interface required a number of modifications to adapt to the BIRDS-1 requirements. Changes were not only introduced by requirements of the BIRDS-1 mission but also specific requirements relating to the use of J-SSOD[15] to deploy satellites from the ISS. Two new projects followed BIRDS-1. BIRDS-2 is a constellation of three 1U CubeSats [15]. SPATIUM-I is a 2U CubeSat but, internally, the PCBs are housed only in a 1U volume with another 1U mass dummy [16]. To save on development time, BIRDS-2 and SPATIUM tried to incorporate as much of the BIRDS-1 backplane design as possible, with little success. Many changes occurred on the backplane electrical connections from one project to another project. The changes led to a redesign of the backplane hardware and to decrease its advantages. Each time, changes were made to the pin assignment of the PCBs and the backplane circuit needed rerouting, resulting in lengthy communications with the manufacturers and a delay in the product delivery. The reasons behind those changes were explained in published work [16]. In short, the reasons were: *launcher environment, different module interfaces, accepting ambitious missions, discontinuity of COTS product,* and *specific design requirements.*

The thesis proposes a new idea of a Software-Configurable Bus Interface (SoftCIB) with a backplane board to obtain more flexibility, particularly for data connections. Instead of hardware routing, a Complex Programmable Logical Device (CPLD) was used to reprogram the bus interface on the PCB. The following advantages will be offered by the standardized backplane interface board: (1) less harness, (2) ease of assembly and disassembly (3) compatible with different CubeSat projects and (4) flexible for routings. We can use the SoftCIB again to reduce the cost and development of the interface boards, rather than designing and making new interface boards for new CubeSat projects. Various projects have various payloads for missions and interface requirements. The high flexibility of SoftCIB's interface allows one to select either the same or a different subsystem board such as an OBC or EPS.

1.2. Research aims and objectives

The vision of this research is to make a contribution in a way of "building CubeSats faster and better" while keeping the cost low. The research aims to shorten development time, assembly time and reduce risk of human errors associated with the interface by taking advantages of the modularity in the CubeSat industry. The goal of the thesis work is to develop and demonstrate the standardized interface board for 1U CubeSat which can be utilized for different CubeSat missions.

1.3. Overview of the thesis

This thesis describes a flexible, standardized electrical interface board for CubeSats. Chapter 2 provides insight into the small satellite community, standardization of interface which is an important part of modularity for any systems including spacecraft, to reduce development and assembly times and costs of CubeSat project. This is achieved by reviewing the literature available in this field. The applications of CPLD for the small satellites are also reviewed in this chapter. Chapter 3 presents the pre-study of this work which was the experience of implementation of backplane approach for real projects. Chapter 4 introduces the concept, validation, and description of Software Configurable backplane Interface board for 1U CubeSats, which potentially reduces the times of development and assembly, and gives more options (flexibility) for the developer to choose modules for their application. The verifications of the SoftCIB are presented in Chapter 5 including various kind of test results. Finally, the conclusions of this thesis are presented in Chapter 6.

2. Background

2.1. Small satellite

The first satellite that mankind sent into space was Sputnik, developed by the Soviet Union, launched in 1957. However, Sputnik was a comparatively heavy 84 kg satellite, which carries out a simple significant space mission to transmit radio signals at two frequencies to determine its presence from Earth. Its 1-watt radio power is the level still used most often for small Earth orbits, and the short life span of its mission was essential to its simple design. The Explorer, first American satellite, was 14kg satellite launched in 1958. This satellite had also a single mission, with a lifetime of three months. Thanks to this mission the existence of regions subsequently known as Van Allen Radiation Belts was detected. There were a few small satellites had been developed, and achieved valuable successes in the early years of space flight. For example, first weather satellite, Tiros (122 kg) and the first geosynchronous communications satellite, Syncom (35kg) was launched in the early 1960s. Since that time, tech advances, Space Races, defense application of large rockets for missile delivery and the desire to put people into space and the surface of the Moon, led to bigger rockets and spacecraft in the United States and the Soviet [17]. Since then, the capability of the launch vehicle increased, simple small satellites have no further interests, until the late nineties. There was no apparent small satellite launched in between 1963 and 1996 [4].

The growth of the small satellite industry's expansion was enormous in just the last two decades. Figure 2-1 shows the number of launches of nanosatellites[1] per year. This graphics only show the statistical numbers of the satellite up to 10kg. We can see that more than 200 satellites are launching by year, nowadays. Figure 2-2 shows the statistical numbers and variations of nanosatellite builder around the globe. The companies and universities are the major developers in this field. Moreover, many others such as space agencies, military, non-profit organizations, institutes, schools, and even individuals are building small satellites [4]. This players, except big companies and space agencies, would never come to this industry if the satellites were big, costly, and took a long time to develop.

[1] The term Nanosatellite implies all CubeSats up to 27U, and 100g to 10kg Non-CubeSat class satellites by https://www.nanosats.eu/

Small satellites have certain design considerations that are different from traditional big satellites. The reduced part count is the most important engineering change due to a smaller scale. Fewer parts are less likely to result in component failure. Redundancy, the first line in defense, has limited or no place on smaller spacecraft against failure. As small satellites tend to perform shorter missions on the lower orbit they do not need to select parts that restricted by radiation durability or long lifetime requirements. The modern components generally have less electric power, smaller masses and volumes, lower costs and easier to work with, which further reduces their complexity. The complexity is the worst rival of the reliability. Although they are often developed by students or amateur teams with limited experience, their simplicity makes satellite reliability comparable with big spacecraft. Moreover, typically, the temperature gradient over a small satellite is not significant in combination with its small size. Together with typically low altitudes, they can have very low power transmission, but directional antennas are seldom used. The most driving force of small satellite design begins with payload power requirements. Because the surface area and power generation are limited by size. Volume and mass of the battery are derived directly from the power requirements. The smaller room for the satellite launch is often a fixed weight and volume, and there are no savings as a consequence of minimizing the size of the room.

Even a smaller team, for example, 5 to 15 people, can design and develop a small satellite in 2 years, compared to thousands of people who have to contribute to a large space program. Every person in the group can assess, within their responsibility, the impact of changes on subsystems in contrast to a large program where requirements have to be met at nearly any cost. Also a small satellite program focus on testing. Management is essential to implement the analysis, design, development and integration phases promptly in order to preserve testing time, personnel and financial resources. Most subsystems are only tested on the desk, not in the special facilities, like thermal vacuum chambers. Because, the entire satellite is small, system level tests can be done with accepted risks that some subsystem may fail during the test. In all cases, the cost of a subsystem failure in a system is not large because it is not so difficult to decompose a small, simple spacecraft in order to uncover a failed component [17].

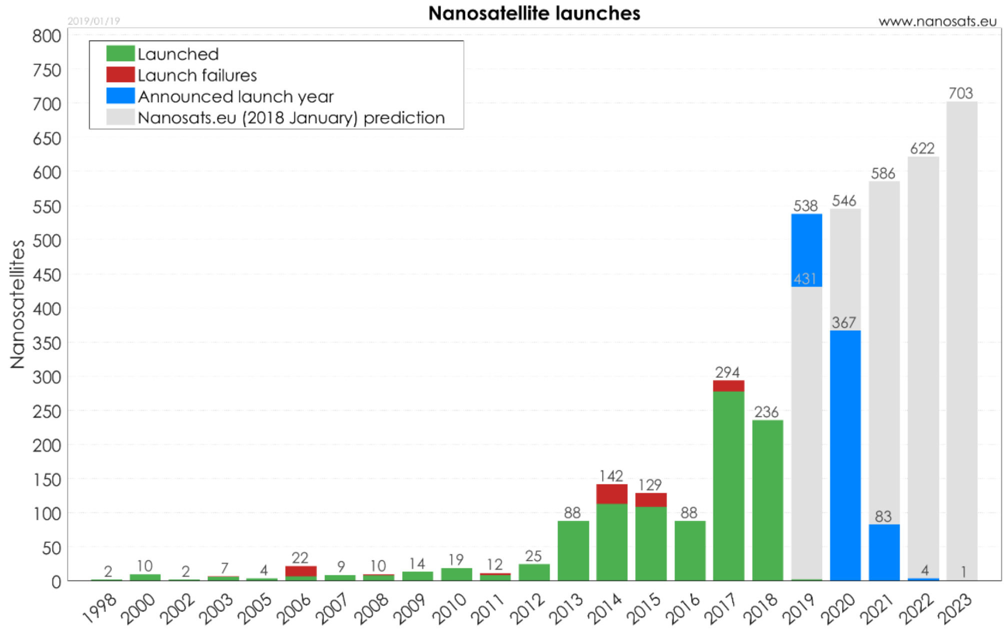


Figure 2-1. Nanosatellite launches by years since 1998, and prediction of the launch for the upcoming 5 years.

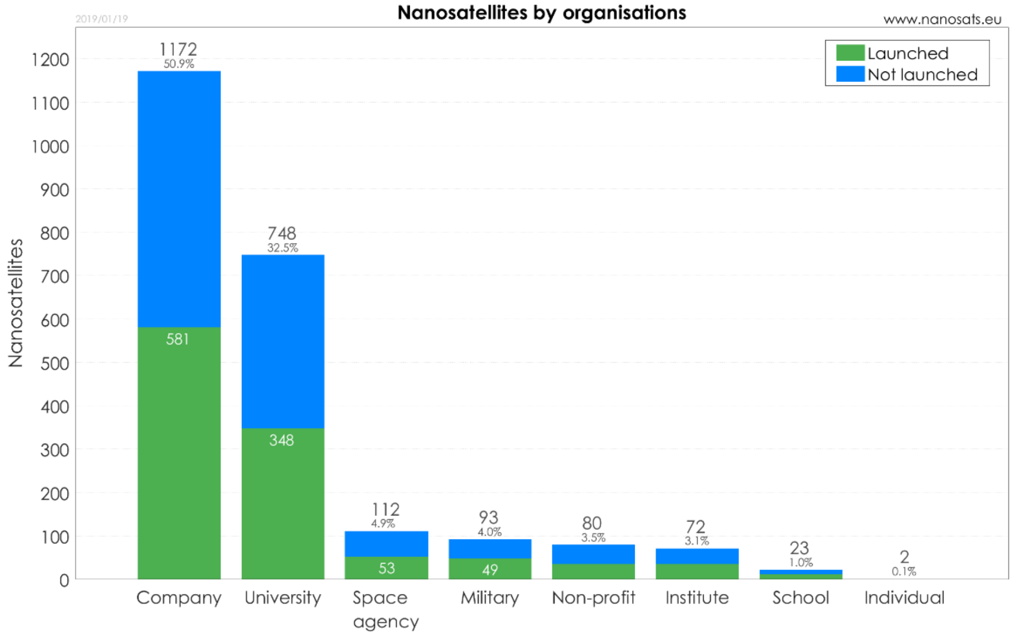


Figure 2-2. Type of the organization that builds Nanosatellite.

2.2. Lean Satellite

The word "Lean Satellite" is appeared, historically, almost 15 years later than a word "CubeSat" or Pico satellite [3]. The recent rapid growth of small satellite launches increased the demand for suitable definition of small satellites not just by judging spacecraft's size or mass. Because, in most cases, mass and size are the results of the mission, satellite program, or philosophy of the design. Lean Satellite approach explains the basic methodology used to develop CubeSat and other small satellites. Thus, CubeSat is a type of Lean Satellite.

A lean satellite, equivalent to "lean satellite program" or "lean satellite mission", is defined as: *"a satellite that utilizes non-traditional, risk-taking development and management approaches with the aim to provide value of some kind to the customer at low-cost and without taking much time to realize the satellite mission"* in the report of IAA study group [18].

This design philosophy is different from a traditional space system development and management. The traditional way of space system development often says that; reliability is the top priority, "failure is not an option". This leads to investing extremely high cost and a long time for the project. On the other hand, Lean satellite approach accepts a certain level of risk and uses non-space-graded COTS components in order to reduce the cost. Lean satellites aim to deliver value to the stakeholders at minimum cost and in the shortest possible time by minimizing waste. As a result, lean satellites project needs to be manageable by a small number of people, satellite size becomes smaller, and satellite lifetime became shorter.

2.3. A CubeSat Technology

CubeSat is a tiny spacecraft which ejects into space from standardized deployer (sometimes called as dispenser) attached to many different kinds of launch vehicle and spacecraft including International Space Station (ISS). The standard size of CubeSat is referred to as "unit" which 1U (one unit) is equal to 10 cm cubic volume with a mass of less than 1.3 kg. CubeSats size is scalable as it can be many times 1U. Practically the most common sizes of CubeSat are 1U, 1.5U, 2U, 3U, and 6U. CubeSats basic information about concept and processes can be found from this [1], [19], [20] source. In 2017, ISO standard (ISO 17770) addresses CubeSats, CubeSat Deployer and related verification of assurance/quality terms and metrics have approved. Example of CubeSat size classification is shown in Figure 2-3.

The idea originated from OPAL (Orbiting Picosatellite Automated Launcher) program, the Stanford University's first student-built satellite. This program demonstrated that the small satellites can be built by students with low cost using non space grade commercial-of-the-shield components. Furthermore, those small satellites can be launched and ejected into space with the help of container mother satellite (or deployer) which is the secondary payload of any launch vehicle [21]. Based on the experience of successful demonstration of OPAL, CubeSat development program started in 1999, a collaboration between Stanford University and California Polytechnic State University (Cal Poly). A first standardized CubeSat deployer, P-POD (Poly Picosatellite Orbital Deployer), was built by Cal Poly [22]. An updated version of P-POD is shown in Figure 2-4. Later, not only universities but also other organizations including government agencies and commercial companies have developed standardized CubeSat deployer equivalent to P-POD. For example, T-POD (Tokyo Picosatellite Orbital Deployer) developed by Tokyo University, X-POD (eXperimental Push Out Deployer) developed by the University of Toronto, SPL (Single Picosatellite Launcher) developed by AFW (Astro und Feinwerktechnik Adlershof GmbH) in Germany, ISIPOD (ISIS Payload Orbital Dispenser) developed by company named ISIS (Innovative Solutions In Space), NLAS (Nanosatellite Launch Adapter System) of NASA, NRCSD (NanoRacks CubeSat Deployer) developed by NanoRacks LLC, and J-SSOD (JEM Small Satellite Orbital Deployer) of JAXA.

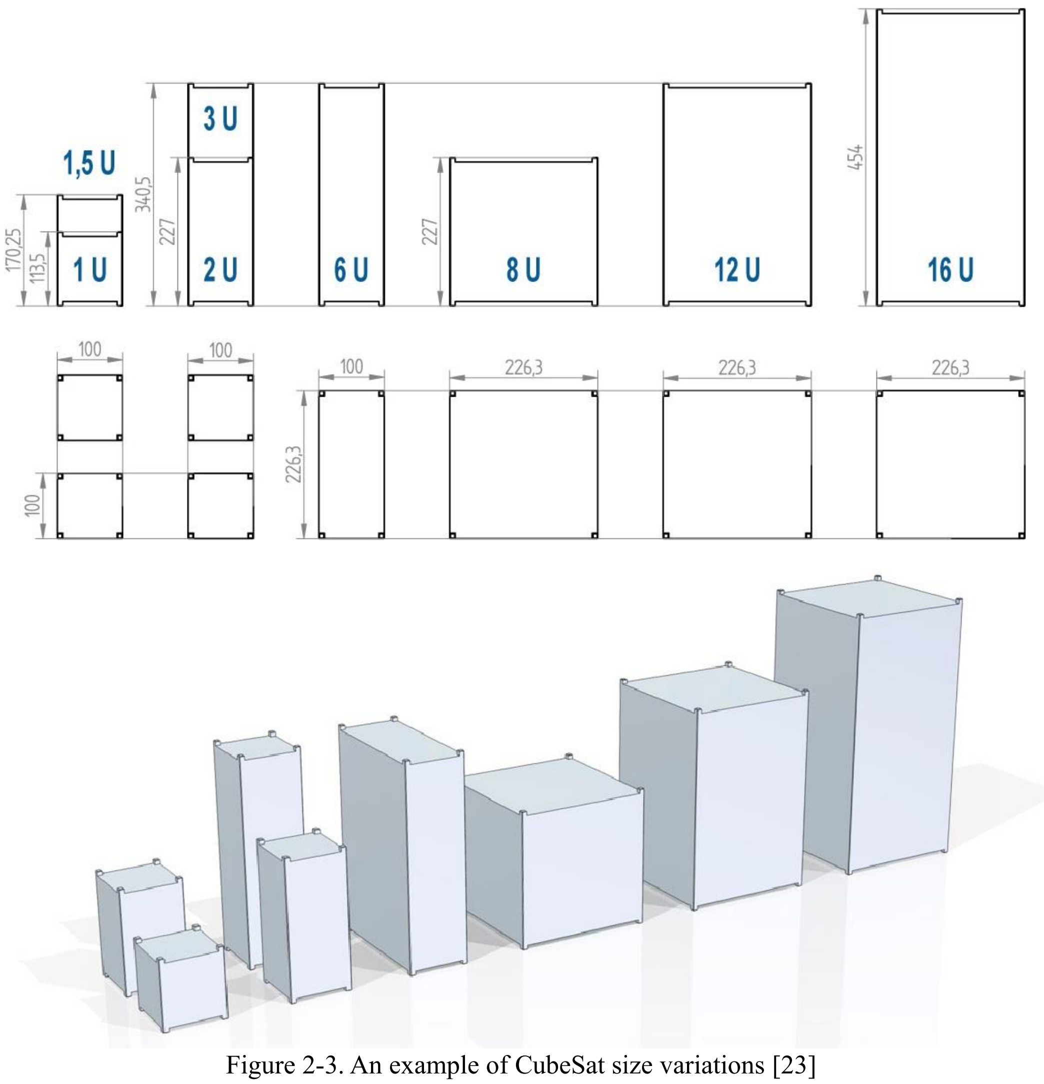


Figure 2-3. An example of CubeSat size variations [23]

The CubeSat, new generation of small satellite, enabled Universities and private groups such as amateur radio clubs to access to space. One of the main merits of the CubeSat is an educational opportunity for engineering students to understand the system engineering with project-based program. Through the CubeSat project, students can participate and learn all the processes of space system engineering such as design, development, fabrication, integration, testing, and operation after launch [24]. CubeSats also can demonstrate technologies that never flown in space before or they can conduct scientific experiments in space by a cheaper, easier and faster way. Not only universities but also commercial companies and government agencies.

Figure 2-4. Poly Picosatellite Orbital Deployer (P-POD) developed by Cal Poly [23]

Furthermore, the Constellation (multiple satellites in orbit) of CubeSats can give competitive solutions against traditional big satellites because of its low cost and fast delivery time. For example, emerging start-up companies “Planet Labs” and “Spire Global” benefit from the market by sharing their data such as earth observation, weather data collection, navigation, and tracking systems, etc. Where the data collected by more than a hundred CubeSats flying in space.

2.4. Modularity

There are many definitions of modularity. The research [25] reviewed and discussed definition of modularity. The only complete consensus was that everyone believes that a modular product consists of modules, building blocks. The more modular a product compared to lying independently the more components that fit into these modules. Modularity defined by Baldwin and Clark [26] is that the construction of a complex product or process from smaller subsystems, independently designed but functioning as a whole. Fundamentally, in any technological design, there are a lot of benefits in modularity. Researches [25]–[29] discussed benefits and positive effects of modularity in the design of products and systems. In this [30] research, the author described a series of potential advantages of modularity underpinning many industrial practices. Some of the advantages concern design and production saving, while others relate to customer reactivity and other relate to design and production system organization and operation. The potential benefits of the modularity articulated as follows.

- Component economies of scale
- Product change
- Product variety
- Flexibility in use
- Order lead-time
- Decoupling of tasks
- Design and production focus
- Component verification and testing
- Differential consumption
- Ease of product diagnosis, maintenance, and repair

2.5. Modularization of Space system

Modular design or architectures have been applied to the space elements in the past, even before the technological advancement was not matured like today. However, the motivation behind the implementation of modularity for spacecraft was not much different than today: in short, to lower the cost, and save time.

In the 1970's, NASA has introduced The MultiMission Modular Spacecraft (MMS) [31] design concept which later have been implemented to several space missions in between 1980 to 1992, including Solar Maximum Mission, Landsat 4, Landsat 5, Upper Atmosphere Research Satellite (URAS), Extreme Ultraviolet Explorer (EUVE), and TOPEX/Poseidon missions. The MMS was the spacecraft design which intended to be used for different missions by taking advantages of modular subsystem design. The idea of the MMS was to cut the redesigning processes of the modules or subsystems which most spacecraft fundamentally shall have, and do the same job among them, such as, attitude control, communications, data handling, and electric power, and structure subsystems. To save costs, MMS were tried to maximize the use of standard components, and a standard set of subsystem modules. The MMS also designed to be compatible with Space Shuttle to take a service (replacing or retrieving the modular subsystems or/and instruments) on orbit. To do so, interface standardization has been applied to MMS, which resulted in the modules of the spacecraft become interchangeable. The actual cost savings are not reported clearly. However, reduction of the Integration and Testing (I&T) times were the notable benefit of MMS [32]. The reasons for the abandonment of the MMS has summarized in this [33] source. Firstly, there was no core technology program for the MMS that would advance modular components. Therefore the performance of the MMS left behind, some of the subsystems needed to be modified or improved to meet the mission requirements of the subsequent missions, over time. Secondly, the MMS program relied on the Space Shuttle. Concerns of the costs and risks of Space Shuttle leads to unpleasing launch and service for the unmanned spacecraft. Only the expensive observatories and Space Stations have a token that chances. The decision to not use polar orbit launch site for the Space Shuttle was also impacted. Finally, optimization of system mass was always crucial to lowering the cost of launch.

Another application of modularity in a more modern space system was Modular, Reconfigurable, and Rapid-response (MR^2) space systems introduced by NASA. Space MR^2 systems are a paradigm shift in the design, construction, integration, testing, and flowning of space assets of all sizes. A proposed testbed for the development and implementation of principles and best practices for spacecraft with MR^2 support infrastructure was the Remote Sensing Advanced Technology (RSAT). This testbed application aims at producing a micro-satellite with high performance and wide applicability, with production-oriented, cost-effective and scalable remote sensing, which is weighted around 100 kg. The building process in 7 days of a spacecraft requires a methodical and concerted view of the integration of systems and of test processes. The key design rules of the MR2 was the plug-and-play interfaces. The Idea behind it was that the integration and testing of spacecraft cannot be much more than the assembly of a personal computer, particularly from an electrical and software perspective, for the standard plug-and-play interface. Standardization is also required of mechanical (including thermal) and fluid interfaces and examples can be provided from the car industry and elsewhere [34].

The lessons learned from the MMS has been applied to NASA's following concept called Modular, Adaptive, Reconfigurable System (MARS). MARS was a broad system-level architecture that consists of all components of a space mission extended life-cycle, including ground and space segments [33]. Modularity used in MARS was not only in the primary subsystem level, compared to MMS. Modules can be a MEMS device, electronic chip, card, box, subsystem, system, or system-of-systems. The adaptive concept was also implemented in MARS, whereas MMS did not have. Moreover, reconfigurability was better applied to MARS than the former. Because MMS system was designed to serve a specific kind of missions while MARS system is entirely changeable in order to apply to different missions. By an introduction of MARS, according to this research [33], “modular”, “adaptive”, and “reconfigurable” concept are described as below.

Modular: MARS framework contains selectable mechanical, electrical, and programming segments that might be utilized and re-utilized in quantized numbers. A system must evolve with the modular (or quantum) components used to incorporate technological advances and accept standard interfaces and plug-and-play principles. MARS modular components and systems must generate intelligent units collectively (and individually), referring to their ability to assemble on the ground or on orbit into larger components or systems. There are different levels of integration for a module from a chip to a card, box, subsystem, system of systems. Any module can perform a "function" either by itself or as a conglomeration system, as an extension of the above.

Adaptive: Simply put, the adaptive control system is capable of learning and acting from the environment. An adaptive MARS system would enable the reconfiguration, either precipitation through a prior definition or an autonomous field requiring a previously unexpected event, of its mechanical, electrical or computer features to change requirements. By the time of MARS, this was a completely new concept and was not supported by the design of the MMS.

Reconfigurable: In order to apply to a host of missions, the system must be able to morph. It must also be easy to manufacture, integrate, test and launch and can work alone or as a collective element, physically detached, or connected. It varies from adaptability as in reconfiguration is normally dependent on a lot of realized limit conditions. A system can be reconfigured, but in certain situations, it cannot adapt to changing requirements.

The design philosophy of the MARS has been advanced and differs from the MMS in several ways.

- MARS Spacecraft and Systems intended to take advantage of commercial standards for production, computing and communication technologies worth multiple billion dollars. This leads to sustainable systems, where not just the government contributes funding to maintain technological relevance.

- Another difference was that the modular design architecture must be able to move forward together with improvements in technology. This means that the "module" may contain technology that is relevant to its time and not fictionally restricted to any "standard." Although MMS was also fundamental, no technology program, allowing for technological development, has ever been established to make modules stagnant.
- At the interface and not at the subsystem or system level, standardization is implemented in the MARS compared to MMS. Commercial standards such as Ethernet, Firewire, USB, and others should be used by electric interfaces. Mechanical interfaces also needed to be standardized, and it should be sufficiently flexible to accommodate different design settings. This was the important key element for the modularity.
- The spacecraft operation system, flight software, and data transport systems were implemented as unique for the MMS. Whereas MARS was different; used operation system with open source code, flight software based on a layered architecture, and distributed command and flow of data on the Internet.

NASA conducted a study with the Applied Physics Laboratory (JHU / APL) of John's Hopkins University in October 2007 to revitalize the Small Explorer (SMEX) program which launched 9 satellites at that time, since 1992 [35]. The aim of the study (SmallSat Modular Hardware and Software Standardization) was to introduce innovative methods for developing small scientific research spacecraft with low cost and high reliability. JHU / APL determined the high-level guidelines which could reduce the cost of SmallSat missions, to be driven by SmallSat standards. The guidelines include the following important points.

- To increase standard interfaces per launch at low integration level ;
- Deviate only enough from current practice, to benefit from existing practice without imposing prohibitive costs based on the number of future production units;
- Enable mission changes to be localized;
- Concentrate on reuse as a facilitator for rapid development and time reduction of I&T;
- Take into account elements outside development of the engineering life cycle, like requirements and testing; and
- Enable modular component development which can be combined to form the required system.

It was found that standardization at the slice level for hardware, and functional level for software efforts and reduces the time required to develop SmallSats through this study.

Stacking systems with various form factors, and modular hardware and software interfaces were chosen by JHU/APL in order to achieve the goal. The entire concept of the hardware standard is based on a slice-level modular structure that allows the modular substitution and modular addition of stacking slices.

2.6. Standardization of the interface

The decomposability of components and interface compatibility issues must be taken seriously in a modular design strategy (as opposed to an integrative design strategy). It is impossible to take separately a modular product design (or modular product architecture) and interface standardization. Because the interface standardization is an important part of the success of modularity [36][37]. However, the effectiveness of the interface standardization defends on what level of modularization it is. Modularization characteristic curve proposed by this [37] research explains the relationships between different levels of modularity and interface constraints (see Figure 2-5). The chance of modularization decreases non-linearly from component level to system level with increasing interface constraints. The opportunity for Modularization and Interface constraints are correlated with every level of modularity. At the component level, interface restrictions tend to be low. Which means that the interface standardization at this level cannot give many benefits to the modularity. While at the subsystem level, interface constraints become high and modularization is more restricted. We can see from the graph that very small amount of change in interface constraints (c) at subsystem level contributes the same amount of opportunities for Modularization with component level, which requires a large range of change in interface constraints. Therefore, sub-system level interface constraints are very sensitive than component level constraints. In other word, interface standardization at the subsystem level is much more desirable.

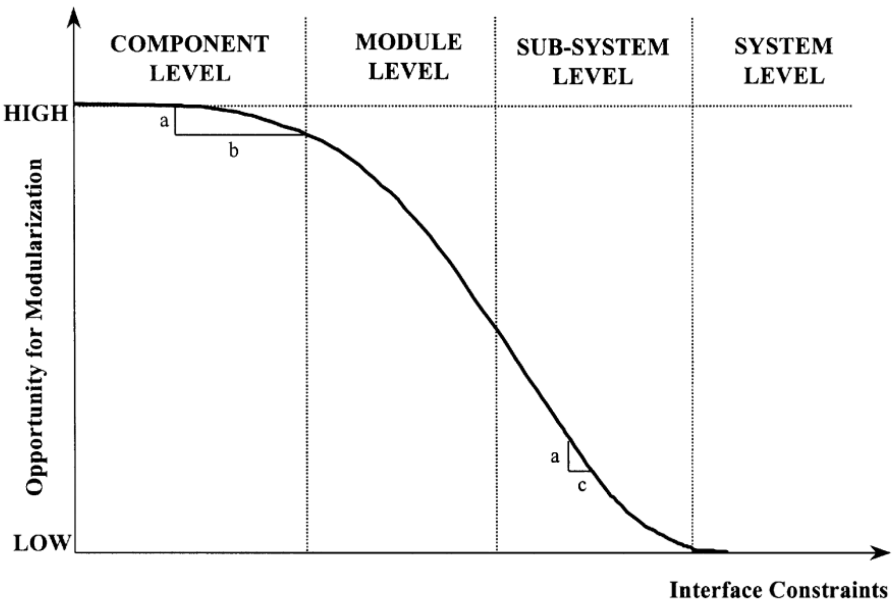
In general, not only the CubeSat domain, standardization have positive effects on the technology market. The functions of standards are classified into four categories for purposes of economic impact assessment. The basic functions of standards according to [38]:

Quality/reliability – Specification of acceptable product or service performance along one or more dimensions, such as function levels, changes in performance, a lifetime of service, efficiency, security, and environmental impact, has been developed for standards. A standard which defines a minimum performance level often offers the starting point for competition in an industry.

Information standards – Standards support the provision, in the form of publications, digital databases, terminologies, test and measurement methods, of evaluated scientific and engineering information for the description or quantification of product attributes.

Compatibility or interoperability – The standardized interface between elements of a larger system usually manifests compatibility or interoperability. The design of the components themselves does not influence an effective interface standard

Variety reduction – Standards restrict a product to a specific range or number of features, such as size or quality. This is the traditional function of the standard which reduces variation to attain economies of scale.



Note: *a* represents a change in opportunity for modularization.
b and *c* represent changes in interface constraints.

Figure 2-5. The modularization characteristic curve [37]

2.7. Interface standardization for CubeSats

CubeSat uses COTS components, thus there are options to choose some particular subsystem from the market. For example, there are 25 companies who sell products (subsystems and modules) for CubeSat listed in the www.cubesatshop.com web site. CubeSats even uses non-flight heritage components and modules which is not invented for CubeSat. Since there are many vendors, modularity is one of the keys to save time in development. But, it requires interface standardization. So far, standardization only applies to the external interface of CubeSat. Therefore, to reduce development time, an internal bus system can have a common standard which connects modules and subsystems together electrically and mechanically.

Within this range, the most well-known interface used for CubeSats is the PC/104. Which have the same form factor of PC/104 embedded computer standard, but modified signals buses for CubeSats. Figure 2-6 and Figure 2-7 shows the example of CubeSat boards with PC/104 interface connectors available on the market. Commercial companies such as GOMspace and Pumpkin Inc makes products that have the interface of PC104 form factor. This interface connector allows subsystems and modules to connect each other by stack through PC/104 connectors. The drawback of the PC/104 was the troublesome procedure when a middle board (of sandwich architecture) must be removed or replaced. All of the top or bottom boards must be removed to reach the targeted board. Also, some research [11] concluded that a smaller connector with less pins is needed.

Figure 2-6. GNSS Receiver Module with PC104 interface, by Pumpkin Inc.

Figure 2-7. Onboard Computer board with PC104 interface, developed by ISIS.

Another approach that CubeSats use dedicated interface PCB to connect all the subsystems inside CubeSat is the backplane approach. It is totally different from the stackable approach that using PC/104 form factor, because, in stackable approach, PC/104 connectors placed on the subsystem boards, and the subsystem boards connect to each other. Some people may think that the backplane approach uses additional PCB, thus it took more space than PC/104. In reality, there is no advantage for PC/104 over backplane in terms of the volume space. A backplane interface board for the 1U CubeSat has developed and demonstrated with UWE-3 satellite by the University of Würtburg, Germany [12], [13]. As there are very few wire harnessing requirements, the time for assembly and disassembly will be much shorter than a PC/104. Figure 2-8 and 2-9 shows the concepts and processes of UWE-3.

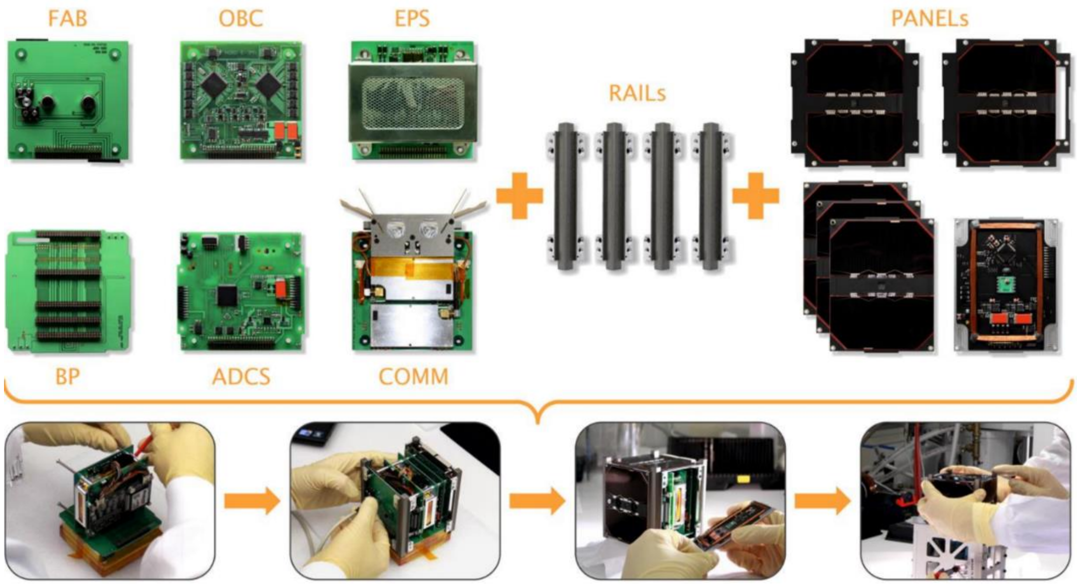


Figure 2-8. The process of UWE-3 being assembled [13]

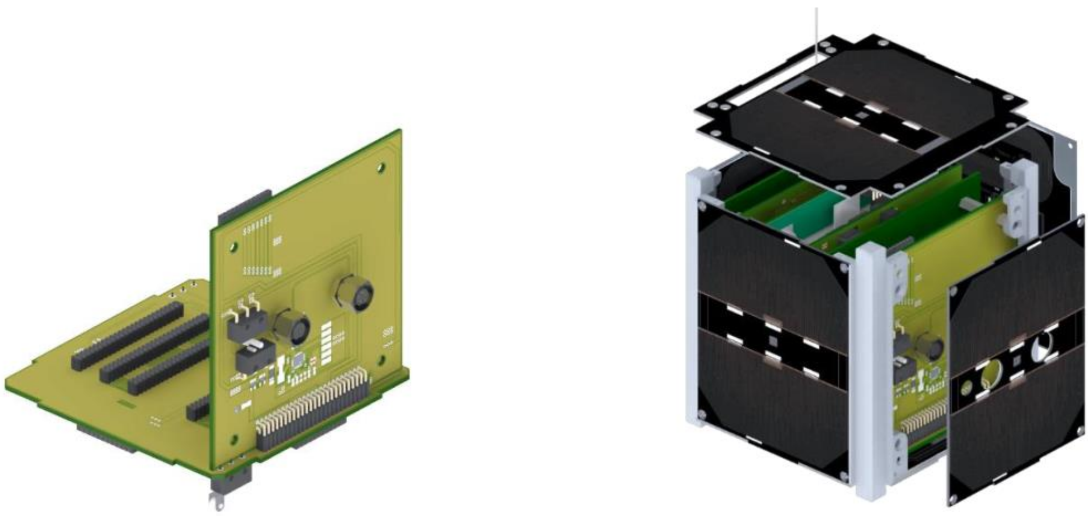


Figure 2-9. Concept of Backplane bus interface introduced by UNISEC global [13].

2.8. Complex Programmable Logic Device (CPLD),

A Programmable Logic Device (PLD) is a device that allows the developer to customize the logic cells to form an electronic digital circuit in the same packed IC. In general, the first programmable ICs were known as PLD. A Complex Programmable Logic Device (CPLD) is a more sophisticated versions of the PLD, and it is one of the most common devices of its kind, nowadays The CPLD comprises a number of logic blocks (occasionally called functional blocks) each containing a macro-cell and PLA or PAL system, and a global programmable interconnection in the central to the design. Besides, FPGA is also a complex form of PLD for much larger design, but the architecture was developed with different fundamental concept. The architecture of FPGA was developed based on a normal array of basic programmable logic cells and a programmable interconnect matrix neighboring the logic cells [39]–[41].

2.8.1. CPLD application for miniaturized satellites

A few missions that have been utilized CPLD for space applications in the past. ARISSat-1 was an amateur radio satellite developed by AMSAT. It weighted around 30kg. The satellite was launched to the ISS in January 2011 and released into space during a spacewalk on February 2011. Altera MAX II CPLD (EPM570T144C5) was integrated into an important unit of the satellite called Integrated Housekeeping Unit (IHU). The brain of the satellite, IHU, was responsible for all the events of the satellite, for example, when taking a picture, sending greetings from space and activating experiments. The purpose of the utilizing CPLD was that it used as a glue logic between the SDRAM and the MCU onboard satellite [42], [43].

OPTOS was a low-cost 3U CubeSat Project with a technology demonstration missions of the Instituto Nacional de Tecnica Aerospacial (INTA), the Spanish Space Agency. OPTOS had five payloads from various scientific fields: Giant MagnetoResistive sensors for magnetic field measurement, Evaluation of total radiation dose using commercial RadFET, Optoelectronic radiometer for protons, Microphotonic devices for temperature measurement, and Low resolution (≈200 m) CMOS camera. Cool Runner II family CPLD was used for the interface devices which provides subsystem control of the platform (PDU, ADCS, heat sensors) and payloads [44], [45].

FUNcube-1 (AO-73) educational 1U CubeSat Project with amateur radio transponder. Launched on 21 November 2013, FUNCube-1 continues to perform well after more than four years in orbit. The telemetry has been received and decoded by over 1000 stations, including many at schools and schools worldwide. FUNcube-1 uses a Xilinx CPLD as a command decoder [46], [47].

CPLD has been used as a digital filter for the scientific payload REPTile, The Relativistic Electron and Proton Telescope integrated little experiment, which is a solid-state particle detector designed to measure solar energetic protons and relativistic electrons in Earth's outer radiation belt. The REPTile was the main payload of the CSSWE (The Colorado Student Space Weather Experiment) satellite. CSSWE was a 3U CubeSat developed at the University of Colorado [48], [49].

Another application of the CPLD was introduced with AAReST MirrorSat, which is a microsatellite designed to demonstrate the autonomous assembly and reconfiguration of the space telescope. A fault tolerant nanosatellite On-Board Computer (OBC-II) for AAReST is developed at the University of Surrey. The OBC-II consisted of two redundant Raspberry Pi Compute Module 3 and an external CPLD. Xilinx XC9500XL device was selected for their application. The main purpose of implementing the CPLD was to switch between those two Raspberry Pi modules in case of failure and power sequencing. However, there were other tasks which CPLD had to perform. The task includes; CPLD controls the raspberry Pi modules' power input, generates reset signals for two modules of Raspberry Pi, two USB hubs and a WIFI device, acts as a smart router for the I2C Bus and Payload UART, monitor the current information in digitized form from the ADCs [50].

3. Implementation of Backplane approach for CubeSats

3.1. BIRDS-1 Project

The Joint Global Multi-Nation Birds Project, known as BIRDS-1, was a constellation of five 1U CubeSats which developed by 15 students from six countries including Japan, Ghana, Mongolia, Nigeria, Bangladesh and Thailand at the Graduate School of the Kyushu Institute of Technology (Kyutech), operated from seven ground stations across the world. All the processes, designing, manufacturing, integration, testing, and operation of the BIRDS satellites were carried out by students. The mission statement of the BIRDS project was "By successfully building and operating the first satellite of the country make the first step toward indigenous space program". The BIRDS project intended to establish a basis for the sustainable space program: accumulate human resources in universities and initiate a research and education program in the host universities. Satellites were developed within two years, more specifically, the project starts in October of 2015, and satellites delivered to JAXA in February 2017. On 3 June 2017, CubeSats launched inside Dragon spacecraft by Commercial Resupply Service mission (CRS-11) to the International Space Station (ISS). From there, they ejected into space by Japanese experimental module KIBO on the ISS. After almost two years of operation, satellites entered the Earth's atmosphere in May 2019. Timeline of the project is illustrated in Fig. 3-1. Ground stations at 7 countries were the main participants of the satellite operation. Photography of the flight model of the BIRDS-1 satellites shown in Fig. 3-2.

BIRDS-1 is an experimental lean satellite approach that is focused primarily on the search for the delivery of value and reduced waste through minimum costs. All development works were carried out in a radius of 30 meters, reducing waste associated with transport and communication dramatically. With the benefit of placing a student together in a single room, face-to-face communication and much less e-mail use are greatly enhanced unless all members of the team are required to communicate [14].

.

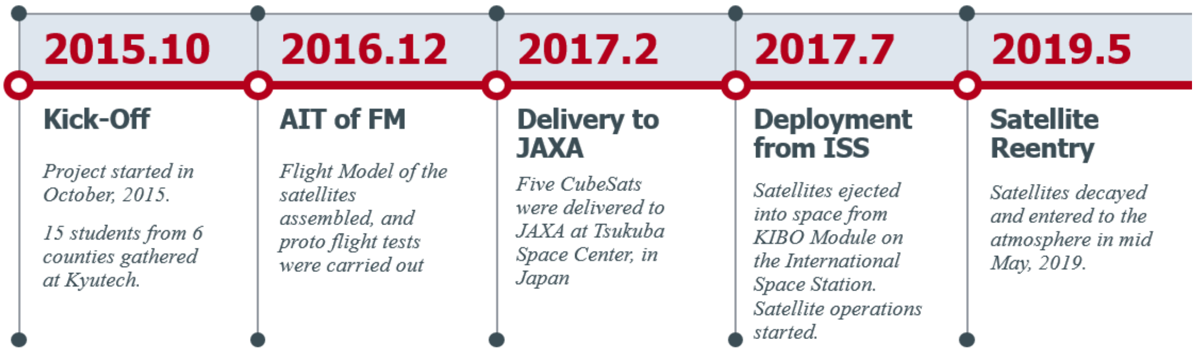


Figure 3-1. The timeline of the BIRDS-1 Project

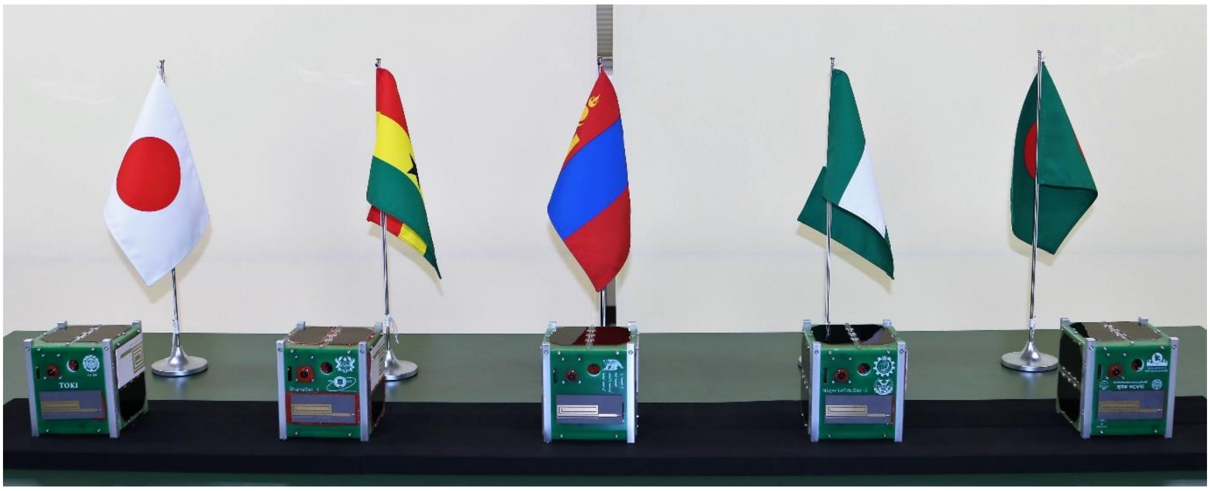


Figure 3-2. Flight model of all five CubeSats of BIRDS-1.

3.1.1. BIRDS-1 Missions

Apart from the project mission, BIRDS-1 CubeSats have total of six missions. The primary mission was an Earth Observation mission which aims to take photography of homeland of each stakeholder. Two cameras (SCAMP with 0.3MP and OV5642 with 5MP) were the payloads of this mission on-board. Next mission was Digi-Singer (SNG) mission. The goal of this mission was to exchange of song from satellites to Ham Radio receivers through UHF band. The measurement of Single Event Latch-up (SEL) in orbit by taking the log of microcontroller reset events over a period of time was the third mission. The fourth mission was the Determination of Satellite Precise Location (POS) using analysis of Time of Arrival (TOA) from time lag at different ground stations. Using data from the POS mission, with orbital analysis, atmospheric density shall be measured, which was the fifth mission. And the final mission was to demonstrate ground station network for CubeSat constellation by Amateur radio frequency.

3.1.2. BIRDS-1 Bus system

The architecture of the BIRDS-1's bus system is based on a space-proven satellite bus, HORYU-II, and HORYU-IV, designed and developed at the Kyutech. The specification of the bus system is shown in Table 3-1. SPI bus communication protocol has selected the data bus line between the subsystems. In a few occasion, UART was used. Flash memories played an important role in the transferring data from one microcontroller to another. Instead of directly sending data to each other, we used shared flash memories to keep data. This method may reduce the data transmitting speed, however, the advantages over data speed were that it was safer and multiple master device can be connected to the bus line. This approach was inherited from HORYU projects. Photography of the BIRDS-1's internal and external configurations are shown in Fig.3-3 and 3-4. All the bus systems and subsystems stacked on the single backplane interface board. Block diagram of communication protocols between subsystems is illustrated in Fig.3-5.

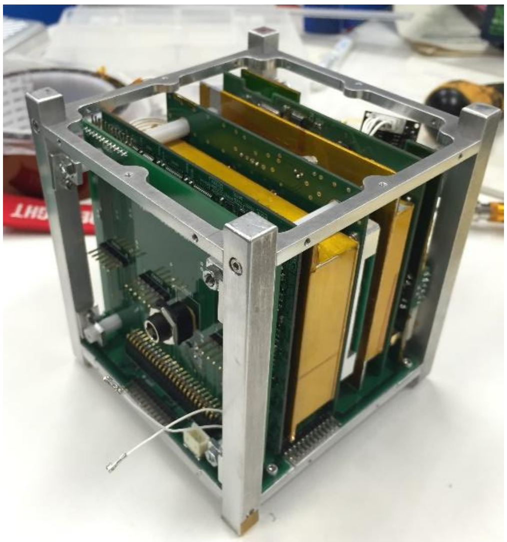


Figure 3-3. Internal configuration of the BIRDS-1 CubeSat

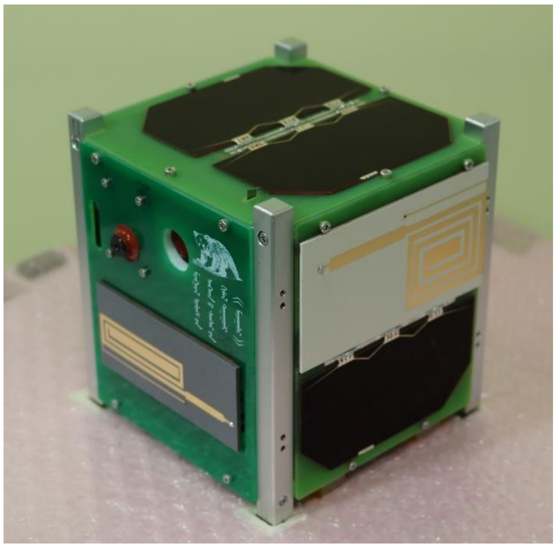


Figure 3-4. External view of the BIRDS-1 CubeSat.

OBC of BIRDS-1 consisted of two 16-bit H8 microcontrollers and two 8-bit PIC microcontrollers. One of the H8, named Main H8, acts as the master control of the mission modes of satellite and communicates through its dedicated flash memories with all subsystems via SPI bus. OBC collects housekeeping data and transmits to the COM system. Another H8, called COM H8, was a central processor of the COM system, and responsible to provide reliable communication between Ground Station (GS) and satellite. COM H8 controls the RX and TX modems, and radio transceiver to receive command and send telemetry data based upon Main H8 or GS request. One of the PIC microcontrollers was dedicated only for power reset in case Single Event Latch-up (SEL) occurs on the H8 microcontrollers. Another PIC microcontroller was dedicated only for the CW transmission.

EPS has 5 solar arrays that have an efficiency of about 29 percent over external structural panels. In order to provide an adjusted voltage level and control over battery charging, booster converters, as battery charging regulators (BCR), have been placed between a solar array and a bus system. With a capacity of 3600mAh, BIRDS uses three series and two parallel Nickel Metal Hydride (NiMH) Eneloop ® batteries. As separation switches, two parallel switches were used. 3.3V, 5V, and unregulated bus voltage can be delivered from the EPS.

Communication subsystem of BIRDS-1 consisted of UHF radio transmitter with 9600bps, UHF radio transmitter with 1200bps, and VHF radio receiver with 1200bps, one deployable monopole UHF antenna, one UHF patch antenna, and one VHF patch antenna. The command is received from the ground-station via VHF (145 MHz-146 MHz) band, with 1200bps of AFSK as a modulation scheme, whereas telemetry and mission data is transmitted by UHF to the ground station (435 MHz-438 MHz).

To ensure the satellite is aligned with Earth's magnetic field, BIRDS' attitude has been passively controlled. A permanent magnet was used to generate torque which the magnetic moment was parallel to the magnetic field line of the Earth. In order to coincide with the geomagnetic vector, permanent magnets are mounted on the satellite's Z-axis. A high magnetic permeable perm-alloy is used on the satellite as a material for the hysteresis damper to dampen the generated motion oscillations. ADCS had 3-axis gyro sensors, a magnetometer, and solar panel outputs to detect the Earth's orientation during the CAM mission.

Table 3-1. Specifications of the BIRDS-1 bus system

Onboard Computer Subsystem (OBC)	- Renesas H8 36057F microcontroller - PIC16F1787 microcontroller - Main data protocol: SPI
Electric Power Subsystem (EPS)	- NiMH battery (3S2P, System 4V, 3800mAh) - Output voltages: +3.3V, +5V and unregulated - Nominal output power: ~ 2W - Peak output: ~ 5W

Communication Subsystem (COM)	- Downlink: UHF (Amateur Radio) 437.375MHz (GMSK 9600bps and AFSK 1200bps) - Uplink: VHF (Amateur Radio)
Attitude Determination and Control Subsystem (ADCS)	Sensors: - Gyroscope - Magnetometer - Solar panel outputs Passive control: - Permanent Magnet - Hysteresis damper

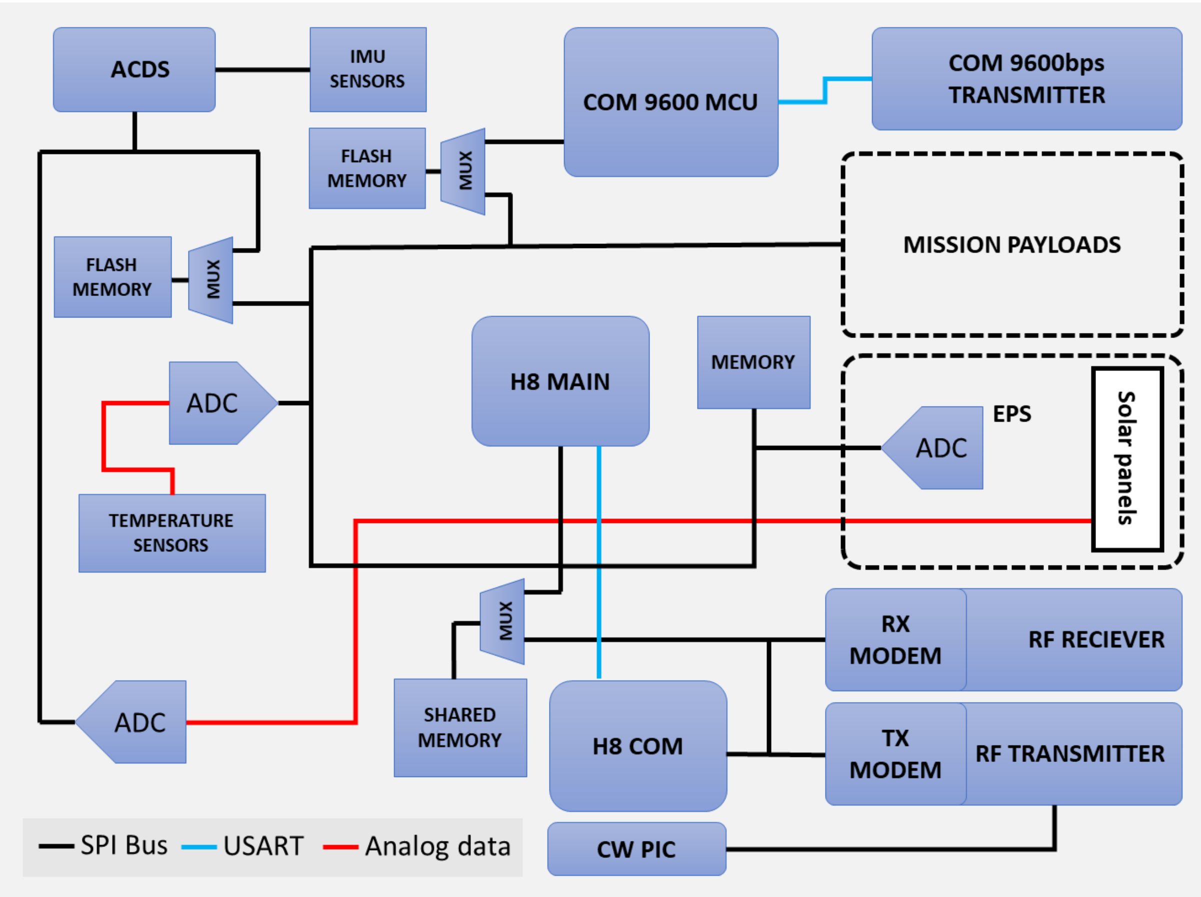


Figure 3-5. Block diagram of the communication protocols of the BIRDS-1

3.2. Backplane Interface Board

Since 15 students building five satellites of BIRDS-1 project, it was crucial to save time in development and testing by any mean. For that reason, all satellites have identical design. The

BIRDS-1 implemented backplane approach which is similar to UWE-3. The reason for choosing this technique was that the satellite configuration should reduce harness and be modularized to support robustness, rapid satellite development, integration and testing, and simple maintenance, and subsystem replacement. Picture of BIRDS-1 backplane is shown in Fig.3-6, and pin assignment shown in Fig.3-7. BIRD-1 backplane interface board not only had all the connectors for the subsystems and modules such as five 50-pin connectors, one 12-pin battery connector, two deployment switch connector, and four solar panel connectors but also had temperature sensor, and ADC (Analog to Digital Converter) circuits to measure the solar panel outputs in the back side of the backplane board. Subsystems that connected to backplane interface board does not use a harness, instead, rigid connectors are directly soldered on the PCBs of each subsystem. The 50-pin connector of the backplane accepts the Front Access Board (FAB), single board for OBC and EPS subsystems, radio transmitter board for 9600bps, Mission board (MSN), antenna deployment subsystem's board (ANT). The backplane interface board was four layer PCB with the size of 97 mm by 99 mm, 1.2mm thick.

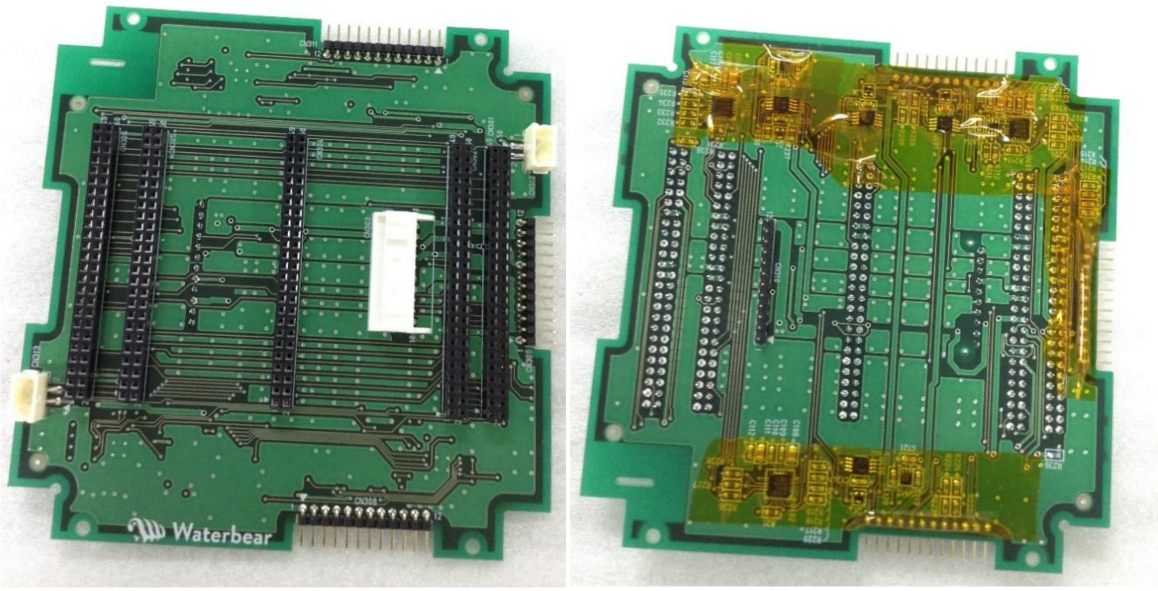


Figure 3-6. Backplane board of the BIRDS-1. Top view (left), bottom view (right).

Front Access Board (FAB)		Electric Power Subsystem (EPS) and On-board Conputer Subsytem (OBC)		Battery connector		Communication board (COM)		Mission board (MSN)		Antenna Board (ANT)	
Pin No.	Connections	Pin No.	Connections	Pin No.	Connections	Pin No.	Connections	Pin No.	Connections	Pin No.	Connections
1	TX-MAIN-PC	1	TX-MAIN-PC	1	GND_ROOT	1		1	COM_96_PGC	1	COM_96_PGC
2	RX-MAIN-PC	2	RX-MAIN-PC	2	GND_ROOT	2		2	COM_96_PGD	2	COM_96_PGD
3	NMI-MAIN-PC	3	NMI-MAIN-PC	3		3		3	COM_96_RESET	3	COM_96_RESET
4	NMI-COM-PC	4	NMI-COM-PC	4	PWR_BAT	4		4	COM_96_VDD	4	COM_96_VDD
5	PIC-ICSPDAT	5	PIC-ICSPDAT	5	PWR_BAT	5		5	CAM_OV_3V3	5	CAM_OV_3V3
6	PIC-ICSPCLK	6	PIC-ICSPCLK	6	THERMAL_CNT_BAT	6		6	CAM_OV_TX	6	CAM_OV_TX
7		7	GIO_6	7	SUP_3V3(NC)	7		7	GIO_6	7	
8		8	9600 PWR ON/OFF	8	BATT_TEMP	8	9600 PWR ON/OFF	8	CAM_OV_RX	8	CAM_OV_RX
9		9	SSO-M	9	GND_SYS	9		9	SSO-M	9	RES_PIC_PGC
10		10	SSCK-M	10	SUP_5V0	10		10	SSCK-M	10	RES_PIC_PGD
11		11	SSI-M			11		11	SSI-M	11	RES_PIC_RESET
12		12	CS-M			12		12	CS-M	12	RES_PIC_VDD
13	GND_SYS	13	GND_SYS			13	GND_SYS	13	GND_SYS	13	GND_SYS
14	GND_SYS	14	GND_SYS			14	GND_SYS	14	GND_SYS	14	GND_SYS
15	SUP_5V0	15	SUP_5V0			15	SUP_5V0	15	SUP_5V0	15	
16	SUP_5V0	16	SUP_5V0			16	SUP_5V0	16	SUP_5V0	16	
17	RES-MAIN-ON	17	RES-MAIN-ON			17	COM96_TEMP	17	SNG_SI	17	SNG_SI
18	RES_COM	18	RES_COM			18		18	SNG_RST	18	SNG_RST
19	RX-COM-PC	19	RX-COM-PC			19	COM96_TX	19	COM96_TX	19	
20		20	GIO_7			20		20	GIO_7	20	
21	TX-COM-PC	21	TX-COM-PC			21	COM96_RX	21	COM96_RX	21	
22		22	THERMAL_CNT_BAT			22		22	SNG_SCK	22	SNG_SCK
23		23	SUP_UNREG			23		23	SUP_UNREG	23	SUP_UNREG
24		24	SUP_UNREG			24		24	SUP_UNREG	24	SUP_UNREG
25	SUP_3V3	25	SUP_3V3			25		25	SUP_3V3	25	SUP_3V3
26	SUP_3V3	26	SUP_3V3			26		26	SUP_3V3	26	SUP_3V3
27		27	RES_PIC_PGC			27		27	ADCS_PGC	27	ADCS_PGC
28		28	RES_PIC_PGD			28		28	ADCS_PGD	28	ADCS_PGD
29	SCVOLTAGE_+Z	29	PWR_BAT			29		29	ADCS_RESET	29	ADCS_RESET
30	SCCURRENT_+Z	30	PWR_BAT			30		30	ADCS_VDD	30	ADCS_VDD
31	DC_CHARGE	31	POWERSC_Y			31		31	ANT_PW_ACK	31	ANT_PW_ACK
32		32	POWERSC_X			32		32	BP_TCK	32	
33	POWERSC_+Z	33	POWERSC_Z			33		33	ANT_M1	33	ANT_M1
34		34	RES_PIC_RESET			34		34	BP_TDI	34	
35	GND_ROOT	35	GND_ROOT			35		35	ANT_M2	35	ANT_M2
36	GND_ROOT	36	GND_ROOT			36		36	BP_TDO	36	
37		37	GND_CTL-1			37		37	ANT_FB1	37	ANT_FB1
38		38	RES_PIC_VDD			38		38	BP_GIO5	38	
39	OBC3.3V	39	OBC3.3V			39		39	ANT_FB2	39	ANT_FB2
40	PIC-MCLR/VPP	40	PIC-MCLR/VPP			40		40	BP_TMS	40	
41		41	SUP_CTL-1			41		41	SNG_S0	41	SNG_S0
42	SUP_CTL-2	42	SUP_CTL-2			42		42	CAM_OV_RST	42	CAM_OV_RST
43		43	GIO_1			43		43	GIO_1	43	
44		44	Reserved			44		44	SCAMP_PGC	44	SCAMP_PGC
45		45	GIO_2			45		45	GIO_2	45	COM-H8-GIO_1
46		46	GIO_3			46		46	GIO_3	46	COM-H8-GIO_2
47		47	COM-H8-GIO_1			47		47	SCAMP_PGD	47	SCAMP_PGD
48		48	GIO_4			48		48	GIO_4	48	
49		49	COM-H8-GIO_2			49		49	SCAMP_RESET	49	SCAMP_RESET
50		50	GIO_5			50		50	SCAMP_VDD	50	SCAMP_VDD

Figure 3-7. Interface definitions of the BIRDS-1 backplane

The original idea that BIRDS-1 tried was to use UWE-3 interface board directly as possible to save on development time. However, during the development, it was necessary to change the interface board. Changes took place not only in the physical form and dimensions, the numbers and locations of the connectors but also in the routing of the PCB design. The reasons that making those changes was, firstly, came from the launcher requirements. The Dnepr rocket launched the UWE-3 CubeSat, while the ISS released BIRDS-1 satellites. BIRDS-1 needed to change the deployment switches because of specific requirements associated with the J-SSOD[15] for the ISS release. Secondly, the difference of communication bus protocol leads to changes in the interface connection. A Serial Peripheral Interface (SPI) was the main BIRDS-1 protocol for communication between subsystems, and it required more communication lines, depending on how many slave units in the bus compared to I2C protocol which used for UWE-3. In order to maximize the efficiency of line utilization, some connections are cut that does not need to be accessed by every connector and the empty pins were used for the new connection. Thirdly, acceptance of ambitious missions affected to the backplane interface. BIRDS-1 had four mission payloads onboard to accomplish total of six missions. Six microcontrollers were placed on the MSN due to space limitation, and MSN was not exposed to outside. It wasn't even nearest to the outside solar panels. External access after integration to program / debug each microcontroller required further backplane connections. To do so, many connections have been made only between Rear access board and MSN on the backplane. Similar to the example above, it is useless and wasteful to connect those pins to the other boards.

Table 3-2 represents results on the basis of the number of interface connections which have been changed to meet the BIRDS-1 Backplane from the UWE-3 Backplane. In the table, numbers in each cell represent the number of electrical connections between any two pins of any connector on the PCB board.

Table 3-2. Number of changes that made on the backplane

Connections	Unmodified	Connections deleted	New connections	Number of BIRDS-1 backplane connections
Analog	32	54	10	42
Digital	50	135	17	67
Total	82	189	27	109

3.3. BIRDS-2 and SPATIUM Project

In order to save development time, two projects which followed BIRDS-1, BIRDS-2 and SPATIUM-I attempted to incorporate as much of a backplane BIRDS-1.

BIRDS-2 was a constellation of three 1U CubeSat, developed at Kyutech by 11 graduate students from four different countries including Malaysia, Philippines, Bhutan, and Japan. BIRDS-2 satellites had total five missions as well as Earth observation Camera, Automatic Packet Reporting System Digipeater (APRS-DP) and Store and Forward, GPS chip demonstration, Single Event Latch-up (SEL) measurement, and measurement of the magnetic field in space. CubeSats deployed from the ISS in August 2018 [51], [52].

SPATIUM-I was 2U CubeSat, although the PCBs internally contain only 1U volume with a second 1U dummy mass. The SPATIUM-1 purpose involves the use of a Chip Scale Atomic clock (CSAC) onboard satellite to scientifically study the ionosphere, to provide a three-dimensional real-time ionosphere plasma density mapping by CubeSats constellation. SPATIUM-I also developed at Kyutech and released from the ISS in October 2018 [53].

Backplane interface boards of SPATIUM-1 and BIRDS-2 are shown in the Fig.3-8. The efforts to use BIRDS-1 backplane to following those project made very little success. There were a number of changes to the BIRDS-1 backplane's electric connections to fit the next projects, which the hardware changes, significantly reduced the benefits of interface board because of redesigning and reproducing time of PCBs. The changes that made on BIRDS-1 to accommodate with BIRDS-2 and SPATIUM-1 is shown in Table 3-3.

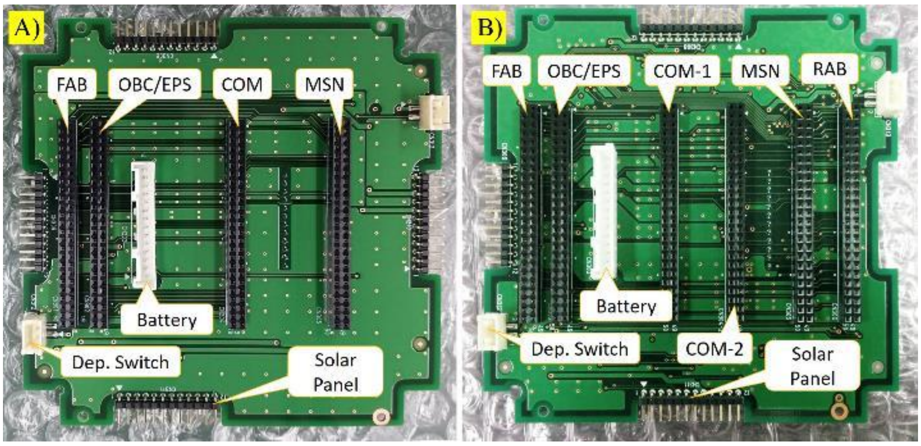


Figure 3-8. Backplanes of SPATIUM-I (A) and BIRDS-2 (B)

Table 3-3. A number of changes have made on the backplane from BIRDS-1 to SPATIUM-1 and BIRDS-1

Connections	Not modified	Connections removed	New connections	Number of connections on backplane after modification
BIRDS-2 backplane	81	22	33	114
SPATIUM-I backplane	22	87	39	61

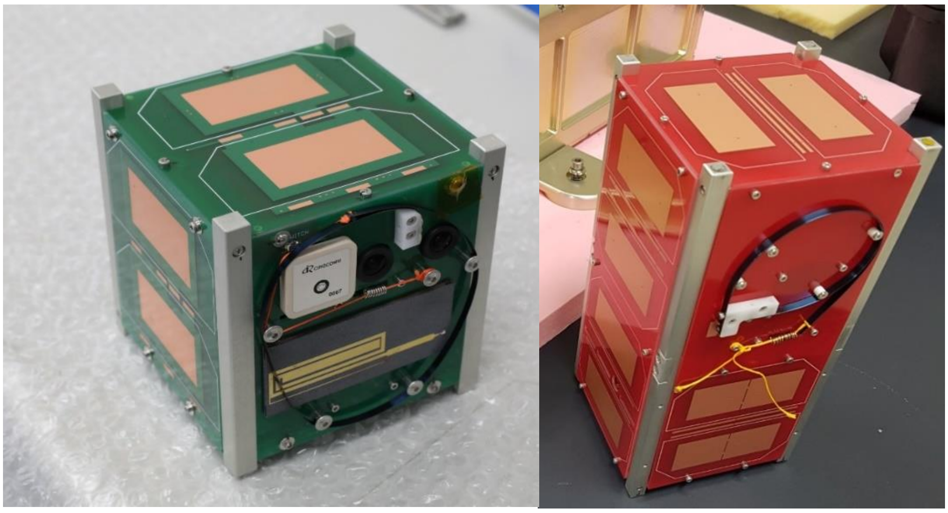


Figure 3-9. Engineering Model of BIRDS-2 (left), Engineering Model of SPATIUM-1 (right)

4. Purposed interface – The SoftCIB

Software Configurable Interface Board (SoftCIB) [16] is a harnessless, highly flexible and backplane style interface board for 1U CubeSats. The design, manufacture, and testing of any CubeSat subsystem take time and resources, while CubeSat projects are often finalized in short time. SoftCIB's objective is to reduce the costs and development time of the CubeSat projects that involved the redesign of interface boards and changes to the interface connection because of mission requirements. The key technology of the SoftCIB is that Complex Programmable Logic Device (CPLD) utilized on the PCB. CubeSat builders can easily define electric interface connection by programming CPLD using Hardware Description Languages (i.e., VHDL or Verilog) to meet their mission requirements. The SoftCIB allows builders of CubeSat to change the interface connection at any time without changing the hardware, as the software is only changed when the CPLD is reprogrammed. CubeSat developer, who utilizes SoftCIB for their project, shall benefit at least in the following ways.

- No Harness – as backplane board, all the harnesses made on the PCB, there is no need to connect cables, or wires for harnessing. Which shall help to reduce workmanship errors. In the CubeSat project, especially at the university, the workforce often composed of students who have not mastered in making spacecraft.
- Quick assembly and disassembly – Compare to a stackable structure like a sandwich, accessing to the particular board is much easier with backplane structure. In sandwich structure, we need to remove all the top or bottom boards to reach the target board while we can directly remove or assemble any with the backplane.
- Flexible interface definition – new method introduced with SoftCIB is that we can change the interface connection at any time, by only changing the software. Thanks to very simple software, ICD can be implemented to SoftCIB within one hour. Once CPLD software changed, then the interface connection already changed. No other works need to be done.
- Compatibility among different CubeSat projects – since CPLD solves the most important issue, interface change due to different requirement, other projects can implement SoftCIB. Time and costs that spent on designing, manufacturing, and the testing new board could be saved.

4.1. Concept of operation

Conceptual image of the SoftCIB is shown in Fig.4-1. The example in this picture shows the interface between only two subsystems. In the left picture, Project A shows that interface connection between OBC and Mission boards. Many connections are going through the CPLD is colorized by blue on the CPLD. That is the connections defined by the software. So, then the right picture, Project B uses the same backplane but different OBC and different Mission boards. Only the software-defined interface on CPLD has been replaced. Or, even in the same project, one of the boards that require different interface connections can be replaced. For instance, Mission board A can be replaced by EPS. However, this picture only shows the interface between two subsystems, all the interfaces except power lines between many different subsystems can be managed by single CPLD. That is the beauty of the SoftCIB.

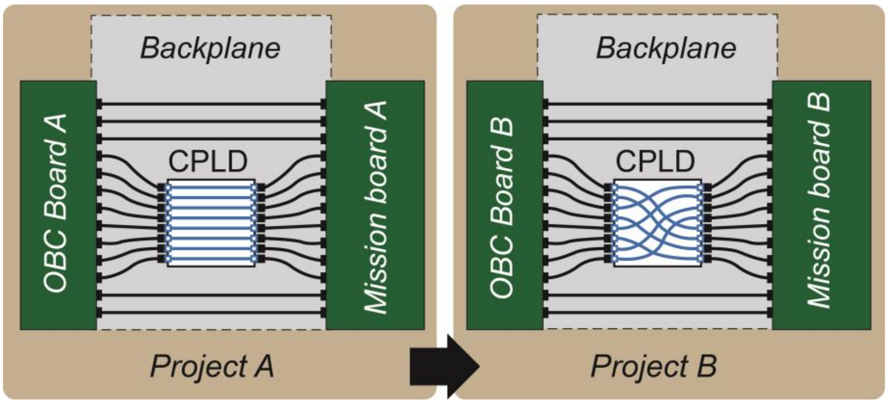


Figure 4-1. Conceptual utilization of SoftCIB

4.2. Design considerations of SoftCIB

Most of the design concerns are correlated with CPLD on the SoftCIB. Due to the very limited power and size budget of 1U CubeSats, the energy consumption and available space of a CPLD in the backplane are the major concerns for the SoftCIB. CPLDs should not be used for heavy processing or calculation onboard CubeSats far more so than a Field Programmable Gate Array (FPGA). FPGA's are often the challenge of high power consumption for 1U CubeSat missions [54]. We set 100mW as an acceptable maximum power consumption because this range of power consumption is manageable for the 1U CubeSat. Also, radiation effects on the CPLD are not a small issue, since CPLD is active semiconductors device onboard the satellite. The CPLD shall have enough strength against radiation in LEO for at least 2 years which is a typical lifetime of CubeSats. Furthermore, CPLD shall have the capability to handle digital signals which can be TTL or CMOS. The mechanical design of the SoftCIB shall be similar to the backplane board of BIRDS-2 and BIRDS-1. This means that there will be 5 to 6 units of 50-pin connector, solar panel connectors, and deployment switch connectors.

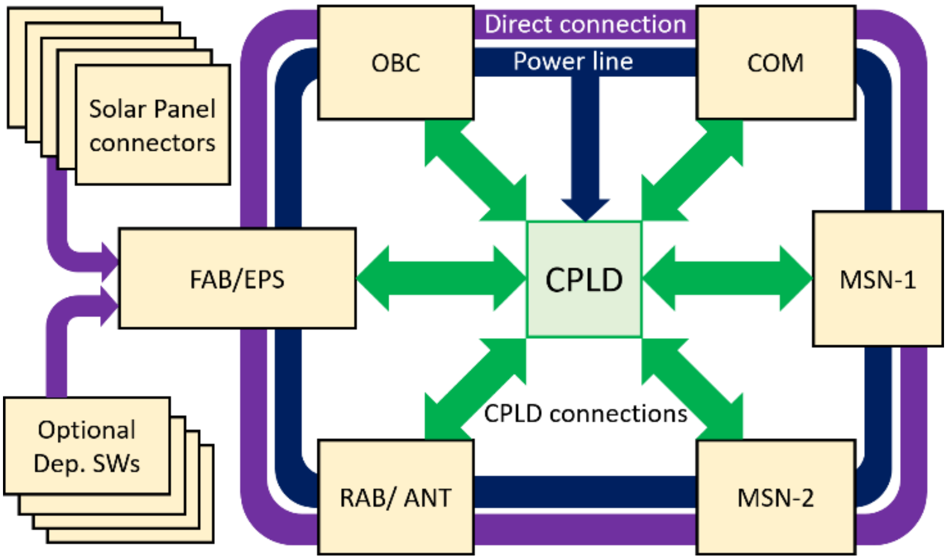


Figure 4-2. Block diagram of SoftCIB

In Figure 4-2, a simplified SoftCIB block diagram is displayed. The SoftCIB consists of general purpose six units of 50-pin connectors which refer to different subsystems such as OBC, COM, EPS or Front Access board (FAB), Mission board -1 (MSN-1), Mission board -2 (MSN-2), and Rear Access Board (RAB) or ANT. For hosts of various subsystems or modules, 50-pin connectors are used. The CPLD on SoftCIB will manage most data connections, excluding power lines, represented by green arrows in the Fig.4-2. Therefore, for any data connection between different boards, the CPLD serves like electrical routing. All power lines in Fig.4-2 are shown in dark blue. Our past CubeSat mission experience has shown that 5V, 3.3V, and unregulated power lines that connected to a 50-pin connector's fixed position was undesirable to change. Moreover, some data lines in violet in Fig.4-2, are routed directly among the 50-pin connectors. These direct links can be used for critical data connections, for instance, it could be utilized to control the communication subsystem. Besides the CPLD, Solar Panels are connected to an EPS board through printed wired on PCB. The SoftCIB also connects the deployment switches to the EPS using a harness. In order to maximize flexibility, two optional MSN positions, and four optional deployment switch connectors are placed on the SoftCIB. A temperature sensor required to measure backplane temperature.

4.3. Selection of the CPLD

Basically, not only CPLD but also FPGA can do the job which required for SoftCIB. However, through the study, it was known that CPLD is more suitable for this application. Several FPGAs from different manufacturers have been studied, for example, SPARTAN-6, -7, and ARTIX-7 from Xilinx Inc., Cyclone from Intel, PolarFire from Microsemi, ECP and CrossLink from Lattice Semiconductor Corporation. They, FPGAs, were powerful in terms of computing and processing which was not part of the main selection criteria. Moreover, this powerfulness comes with the sacrifice of energy consumption. And FPGAs equipped with a lot of logic cells which most of them will not be used in our application.

Trade-off study has been done to select the CPLD. A number of factors have been set for the selection criteria considering its price, power consumption, temperature range, physical size and number of pins. Information about power consumption was very difficult to find. Initial information about power consumption was collected from the manufacturers' web page by surfing the internet. Power consumption of few competitors actually measured at the laboratory. Another factor, easiness of the manufacturability of the PCB backplane is also considered. Because a package of the device will significantly influence the cost of PCB manufacturing. For example, devices with Ball Grid Array (BGA) packages make the PCB board more expensive due to its difficulties for soldering, and inspection. In addition, it was not good for the development, mostly developer needs to use a socket. Lastly, the evaluation board of the particular device was also considered.

The table 4-1 below is the summary of the trade-off study which sorted by market price (low to high) from Digikey [55]. It shall be noted that this table does not represent the final decision. This shows the potential competitors for the selection. An ispMACH®4000ZE (4256ZE-7TN144I) the device of the Lattice Semiconductor Corporation has been selected for the CPLD of the SoftCIB considering all the aspect before mentioned.

Table 4-1. Candidate CPLDs that sorted by unit prices

Manufacturer Part Number	Manufacturer	Unit Price (USD)	Series	Delay Time tpd(1) Max	Voltage Supply - Internal	Number of Macrocells	Number of I/O	Operating Temperature	Supplier Device Package
5M240ZT144I5N	Intel	6.8	MAX® V	7.5ns	1.71V ~ 1.89V	192	114	-40°C ~ 100°C (TJ)	144-TQFP (20x20)
LAMXO640E-3TN144E	Lattice Semiconductor	12.6	LA-MachXO	4.9ns	1.14V ~ 1.26V	320	113	-40°C ~ 125°C (TA)	144-TQFP (20x20)

5M570ZT144I5N	Intel	13.2	MAX® V	9.0ns	1.71V ~ 1.89V	440	114	-40°C ~ 100°C (TJ)	144-TQFP (20x20)
LC4256ZE-7TN144I	Lattice Semiconductor	15.9	ispMACH® 4000ZE	7.5ns	1.7V ~ 1.9V	256	96	-40°C ~ 105°C (TJ)	144-TQFP (20x20)
LC4256V-10TN144I	Lattice Semiconductor Corporation	22.1	ispMACH® 4000V	10.0ns	3V ~ 3.6V	256	96	-40°C ~ 105°C (TJ)	144-TQFP (20x20)
LC4256ZC-75TN176I		23.3	ispMACH® 4000Z	7.5ns	1.7V ~ 1.9V	256	128	-40°C ~ 105°C (TJ)	176-TQFP (24x24)
LC4256ZC-75TN176E		25.7		7.5ns	1.7V ~ 1.9V	256	128	-40°C ~ 130°C (TJ)	176-TQFP (24x24)
XA2C256-8TQG144Q	Xilinx Inc.	28.2	CoolRunner II	7.0ns	1.7V ~ 1.9V	256	118	-40°C ~ 105°C (TA)	144-TQFP (20x20)
LC4256V-75TN144I	Lattice Semiconductor	31.4	ispMACH® 4000V	7.5ns	3V ~ 3.6V	256	96	-40°C ~ 105°C (TJ)	144-TQFP (20x20)
EPM570T144I5N	Intel	31.6	MAX® II	5.4ns	2.5V, 3.3V	440	116	-40°C ~ 100°C (TJ)	144-TQFP (20x20)
EPM570GT144I5N		31.6		5.4ns	1.71V ~ 1.89V	440	116	-40°C ~ 100°C (TJ)	144-TQFP (20x20)
LC4256V-75TN176I	Lattice Semiconductor	31.8	ispMACH® 4000V	7.5ns	3V ~ 3.6V	256	128	-40°C ~ 105°C (TJ)	176-TQFP (24x24)
EPM570GT144I5	Intel	34.8	MAX® II	5.4ns	1.71V ~ 1.89V	440	116	-40°C ~ 100°C (TJ)	144-TQFP (20x20)
EPM570T144A5N		38.0		5.4ns	2.5V, 3.3V	440	116	-40°C ~ 125°C (TJ)	144-TQFP (20x20)
LC4384V-10TN176I	Lattice Semiconductor Corporation	43.4	ispMACH® 4000V	10.0ns	3V ~ 3.6V	384	128	-40°C ~ 105°C (TJ)	176-TQFP (24x24)
LC4256V-5TN144I		44.0		5.0ns	3V ~ 3.6V	256	96	-40°C ~ 105°C (TJ)	144-TQFP (20x20)
LC4256V-5TN176I		44.4		5.0ns	3V ~ 3.6V	256	128	-40°C ~ 105°C (TJ)	176-TQFP (24x24)
XA2C384-11TQG144Q	Xilinx Inc.	49.2	CoolRunner II	9.2ns	1.7V ~ 1.9V	384	118	-40°C ~ 105°C (TA)	144-TQFP (20x20)
LC4384V-75TN176I	Lattice Semiconductor	61.4	ispMACH® 4000V	7.5ns	3V ~ 3.6V	384	128	-40°C ~ 105°C (TJ)	176-TQFP (24x24)

4.4. Concept validation

A functional test with the Bread-Board Module (BBM) of the SoftCIB was performed to validate the concept. In this test, some boards (OBC, FAB, and MSN) of the TableSat versions of the BIRDS-1 CubeSat was used to confirm the interface connection through the CPLD. Evaluation board of the ispMACH®4000ZE had been used for the BBM as shown in the Fig.4-3. The BBM interface is programmed to substitute the backplane of BIRDS-1. The BBM was fed by a DC supply instead of batteries. Table 4-2 provides an overview of the functional test result. All test results were good, as shown in the table as "○". The SPI and UART communications between different microcontrollers among different subsystem boards were checked, and no anomalies detected. Again, general logic "High" and "Low" signal sent from OBC to control multiplexer of the MSN board, no failure recorded. The BBM board consumed about 36 mW of total power.

After that, a prototype board was produced (six-layer PCB with the size of 96.8 mm × 96.8 mm × 1.6 mm) and similar test with BBM was conducted. Figure 4-4 shows the photography of the prototype board with OBC/EPS, MSN, and RAB of BIRDS-2 installed. The results of the functional test were represented in the last column of Table 4-2. Test results have been good, as we can see.

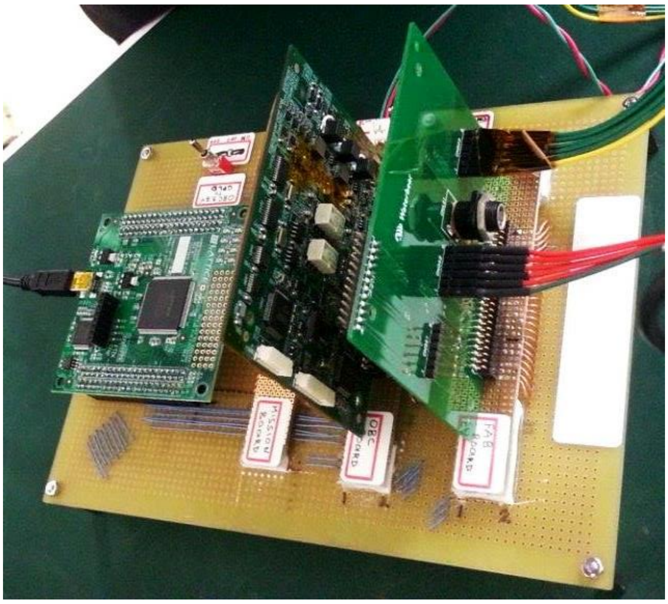


Figure 4-3. BreadBoard Model of the SoftCIB

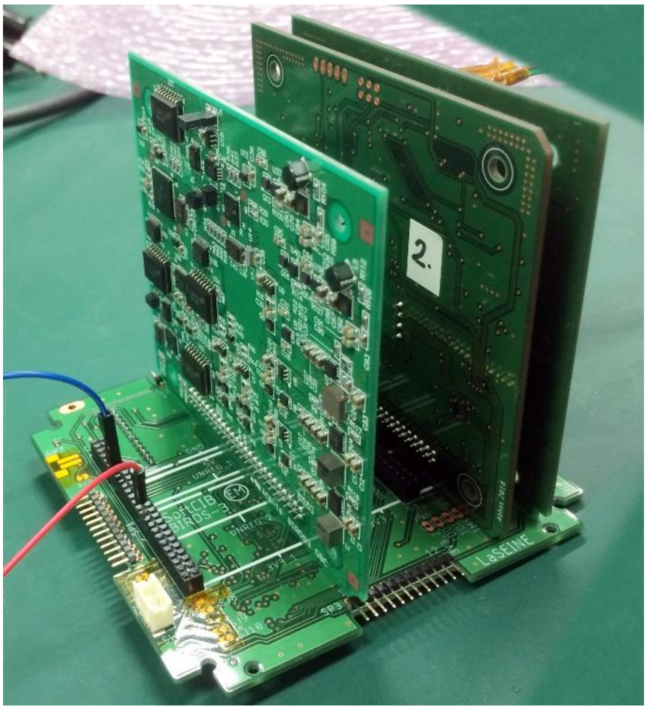


Figure 4-4. Prototype board of the SoftCIB with OBC, MSN, and RAB of BIRDS-2.

Table 4-2. An overview of the functional test results

Functions	BBM board	Prototype board
UART – OBC microcontroller and PC *(115200 bps)*	○	○
UART – COM microcontroller and PC *(115200 bps)*	○	○
UART – COM and OBC microcontrollers *(9600 bps)*	○	○
SPI – Flash Memory and OBC microcontroller *(1Mbps)*	○	○
SPI – Mission Board Memory and OBC microcontroller *(1Mbps)*	○	○
COM microcontroller reset through CPLD (H/L signal)	○	○
OBC microcontroller reset through CPLD (H/L signal)	○	○
OBC board and Mission board (H/L signal)	○	○

4.5. Software

The SoftCIB uses much simpler algorithms and software compared to the FPGA. The software's basic function is to define and assign input/output pins in VHDL which are generally defined by interface control document (ICD) of the satellite project. The CPLD acts in most cases as a simple digital follow-up circuit as the harness is replaced. For instance, one of the output pins follows the input pin logic.

4.6. Flexibility

SoftCIB has 63 connections which are permanently routed to the PCB board and the CPLD can configure 46 flexible connections. The term "connection" above refers to one signal line between the two ends of the 50-pin connector (pins). For this purpose, the connections between the 96 pins of the different 50pin connectors can be routed by CPLD pins (total number of flexible connections are 46). Also, the PCB has 31 permanent electric routes which govern all power lines, including 3.3 Volts, 5 V, unregulated voltage, the battery raw power and ground lines.

5. Testing campaign

5.1. Performance Testing

Considering the functions of CPLD, following two types of tests have done to check the performance of the SoftCIB. At first, it was necessary to find out the signal delays between inputs to the output of the CPLD. To check this performance, only two pins (as a single output and single input) of the CPLD were used. CPLD was programmed to follow the logic state of the input pin to the output. Then, the clock signal (logic state of high is 3.3 V, and 0 V corresponds to the logic state of low) was given from the functional signal generator at the input, and the output was measured by an oscilloscope. Due to the availability of equipment, the input signal frequency was increased to 40 MHz while monitoring the output. Schematic of the test setup is shown in the Fig.5-1. Comparison of the input and output signals measured on the oscilloscope shown in the Fig.5-2. It should be noted that in Fig.5-2, the signal forms are not identical and are not precisely square as the capacity of signal probes did not well compensate for the measurement of high frequencies. Nevertheless, the CPLD output follows the input without interruption or a loss of signal, and a signal delay was around 9 nanoseconds. We can see from the Fig.5-2 that the delay is consistent.

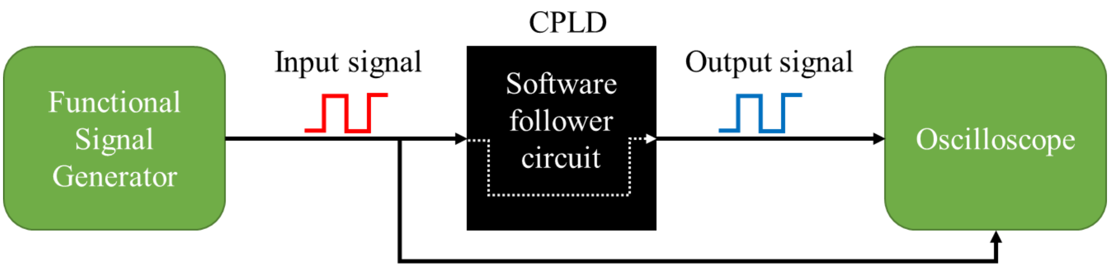


Figure 5-1. Test configuration for the simple input and output test

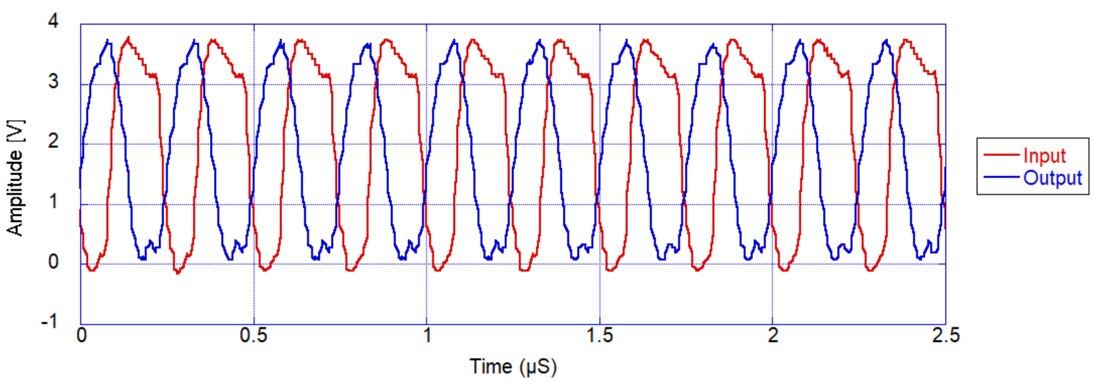


Figure 5-2. Comparison of the signals at the input and output of the CPLD pins

Next test was to determine the highest speed of communication that a CPLD can handle. The actual SPI communication test has been conducted to check this performance. CPLD serves as a bridge connection of the SPI between microcontroller and camera module. For this particular test, a master device was the Arduino Board and a camera module (OV5647) was the slave SPI device. CPLD was programmed to connect the four signal lines of the SPI — CLK (Clock signal), CS (Chip Select), MOSI (Master Output & Slave Input), and MISO (Master Input & Slave Output) — in between camera module and microcontroller. The other lines for controlling the camera module were not passed through the CPLD. The test setup is shown in the Fig.5-3.

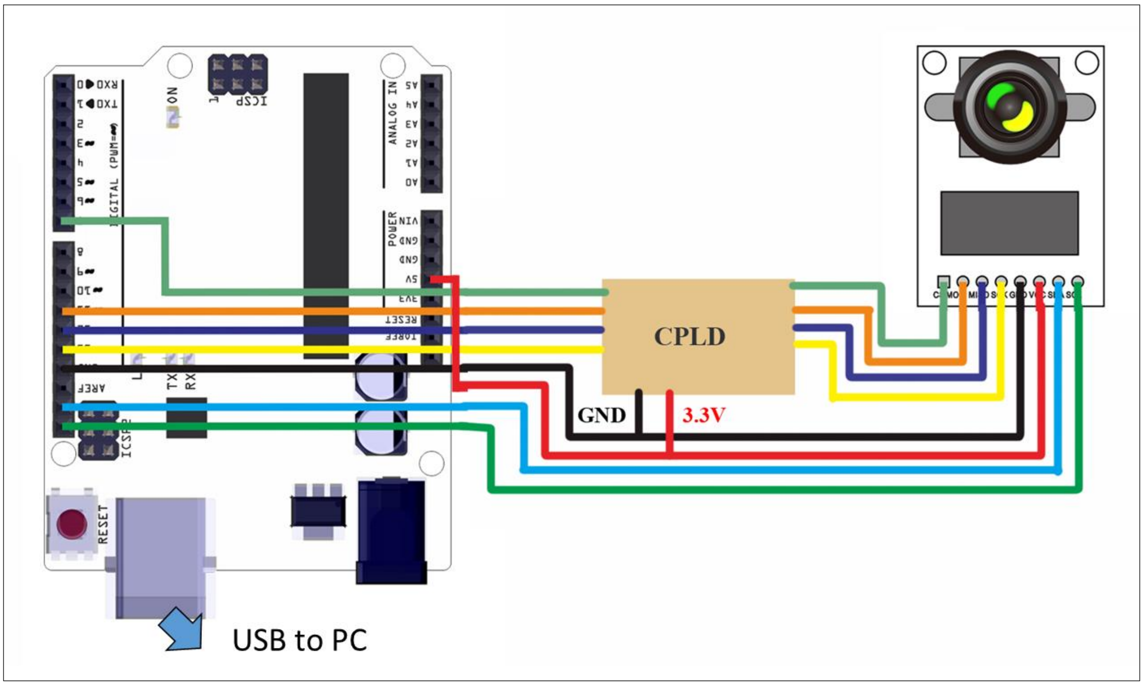


Figure 5-3. Setup for the SPI communication test using Arduino and OV camera module

Test sequence was: (1) the command sent from the Arduino board to take pictures to the camera module, (2) camera took a picture, (3) send back the image in JPG format via SPI, (4) Arduino receives the image file from Camera module, (5) send it to PC by USB, and (6) to check the image quality on the monitor. Multiple images were taken and downloaded on the PC at each of the SPI speeds (1, 2, 4 and 8 Mbps). The testing limit of the SPI speed was limited due to OV5647 camera module which the highest speed was 8 Mbps according to its datasheet. If the final images could be shown by a simple JPEG viewer program, it is considered that information was lost during the SPI communication. Because, even a single bit error can make the JPEG image non-restorable and step (6), USB communication to PC, was very reliable. To compare the test result, another test which uses a normal jumper cable instead of CPLD have done by the same test sequence. The results of the test are listed in Table 5-1.

Yet the results indicate that 8 Mbps SPI communication failed completely in the ' CPLD ' test and failed partially in the ' CPLD-free ' test. Failure of the 8Mbps was investigated following this test. In order to do this, all 4 inputs and the 4 CPLD outputs were simultaneously monitored. Three lines of SPI had no problem. The outputs provided the same logic as the inputs. However, the only camera module output signal line, MISO, was problematic. As Figure 5-4 shows, the signal at the CPLD input and output was somewhat different. The red color is the input signal for the CPLD, i.e. the camera module's output, while the blue color is the output signal of the CPLD connected to the Arduino board. We can see that the falling edge of the input signal has some noise and it is not straight. This can be explained as below.

The maximum value for a logic low signal for the CPLD (Family Lattice ispMACH4000ZE) is 0.8 V so that a logical output signal is cannot become low unless the input signal became below 0.8 V. The logically high output signal on the graph is, therefore, longer than the input signal. This time difference is crucial if a bit length becomes shorter or if the communication speed becomes higher. Therefore this leads to bit errors and communication failures. The conclusion is that the OV5647 camera module's output signal is not good for a high-frequency digital signal. The CPLD transmitted the information but failed to track the signal's falling edge.

Table 5-1. Result of the SPI communication test

SPI communication speed	Success rate of "CPLD" test	Success rate of "CPLD-free" test
1 Mbps	100%	100%
2 Mbps	100%	100%
4 Mbps	95%	95%
8 Mbps	0%	50%

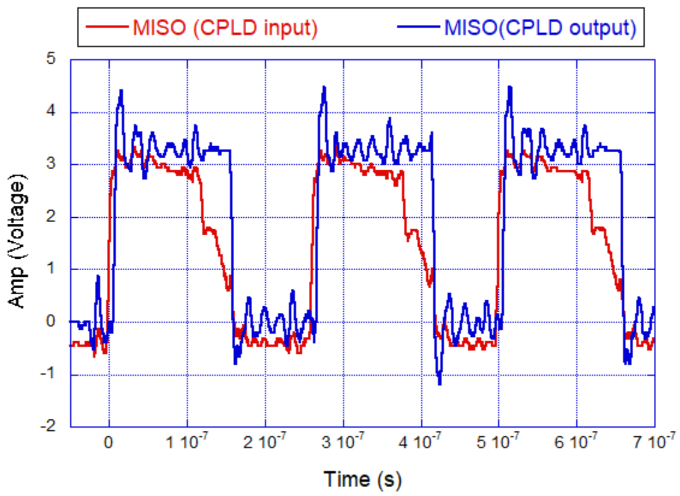


Figure 5-4. MISO signal at the input and output of the CPLD

5.2. Hot and Cold start test

Any active electronic devices need to be verified that system, module, or units can be turned on after the separation/deployment from launcher or ISS in space. A satellite, or some device, modules, or units of the satellites could be heated or cooled when the satellites ejected into space. In any cases, the satellite needs to start normally as it is intended. To confirm that CPLD can be turned on in space after the deployment, Hot/Cold start tests have done for SoftCIB. In general, for the system level test, temperature values shall be determined based on thermal analysis. However, for the qualification test at the unit level can be done as recommended by ISO 19683:2017. In this test, the starting temperatures for the CPLD were -35ºC for a cold start, and 65ºC for the hot start test. Thermal static chamber has been used for this test and SoftCIB was the only test article. The test setup is shown in the Fig.5-5. And the temperature profile of the test is shown in the Fig.5-6. PC that outside of the chamber were connected to CPLD to run and monitor the test. The power source to the CPLD was given from external Power Supply. The thermocouple was placed on the CPLD as shown in the Fig5-7 to measure the temperature. Starting sequences as functional tests have done at the high and low temperatures. Test was conducted when temperatures reached the target point and become steady. The test result was good, there were no issues. Both, cold and hot temperatures did not cause any abnormally to the starting of the CPLD.

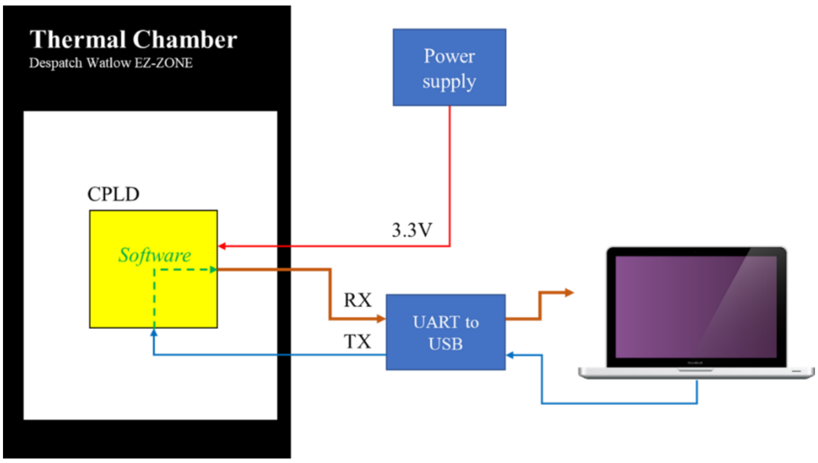


Figure 5-5. Hot/Cold start test setup

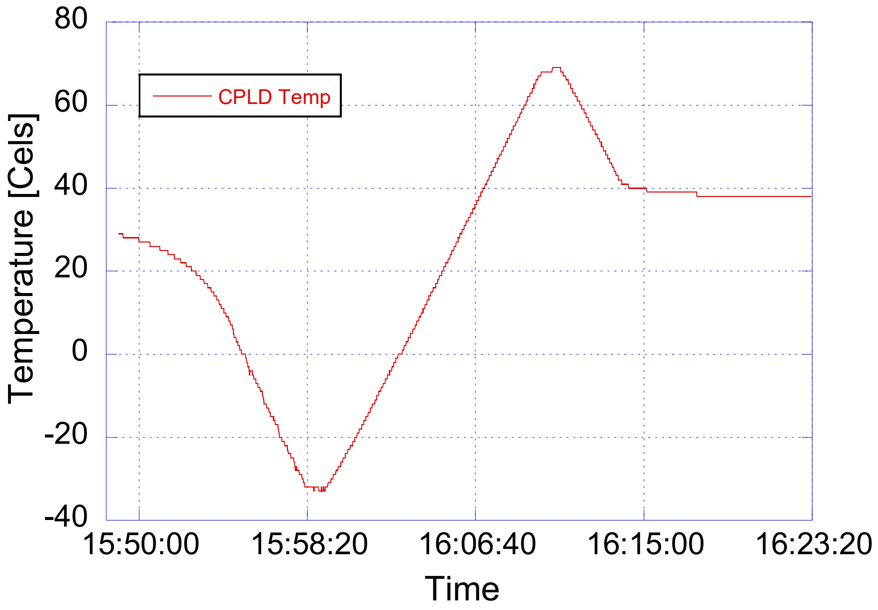


Figure 5-6. Temperature profile of the Hot/Cold start test

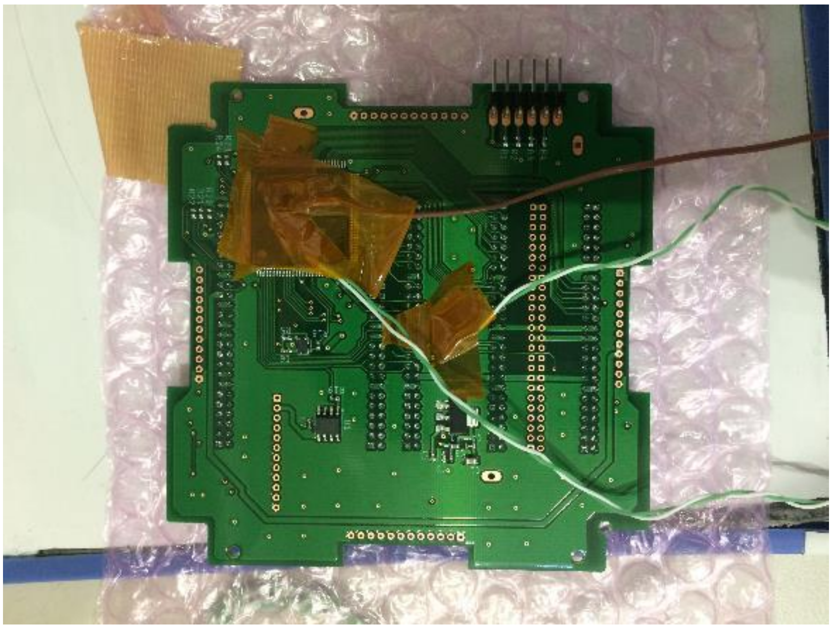


Figure 5-7. Thermocouples placed on the CPLD of SoftCIB

5.3. Radiations Tests

Space is a harsh environment not only in terms of temperature and pressure but also radiation. Therefore, any electronic devices on the spacecraft have to be operated in that radiation environment. When microelectronic technology advanced as they became smaller and powerful (a large number of transistors fit in a small area), then the vulnerability against the radiation has been raised. There is three main sources of the radiations in space where the charged particles come from, which includes Galactic Cosmic Rays, Solar Flares, and Van Allen Belts. Unusually, these energetic particles have an immediate effect if an electronic component is affected. Many spacecraft have failed because of this space radiation. Therefore, this is a serious problem, especially those small satellites which uses non-space-grade COTs components.

5.3.1. Total Ionizing Dose Test

It has been studied that Complementary Metal–Oxide–Semiconductor (CMOS) devices are vulnerable under the space radiation. One of the main issue for the CMOS device in space is the Total Ionizing Dose (TID) effect. The trapped charge induced by radiation has been built in the gate oxide of the CMOS transistors which causes the threshold voltage to shift (in other words, a change in the voltage which must be applied to turn the device on). If this shift is large enough, it is not possible to switch off the device, even when the zero volts are being applied. Because of this effect, devices have failed to function correctly [56]. To ensure that a Commercial off-the-shelf (COTS) CPLD can survive for the typical CubeSat life in a space radiation climate, a total ionization dose test was necessary. The CPLD was the only critical electronic part in SoftCIB, not the whole SoftCIB board was tested. Three CPLD-samples were tested under 3 different dosing conditions (part number: LC4256ZE7TN144I). Cobalt-60 (Co60) was the source of radiation and radiation doses were calculated by distance, as illustrated in Table 5-2. Test conditions set by the unit qualification test level defined in the ISO standards (ISO-19683:2017) [57]. The test has done usingthe facility of the Center for Accelerator and Beam Applied Science of Kyushu University in Japan. The test configuration is shown in Fig.5-8 and actual photography of the test setup shown in Fig.5-9. This test did not take into account performance degradation. The criterion of the test was simply passed or fail.

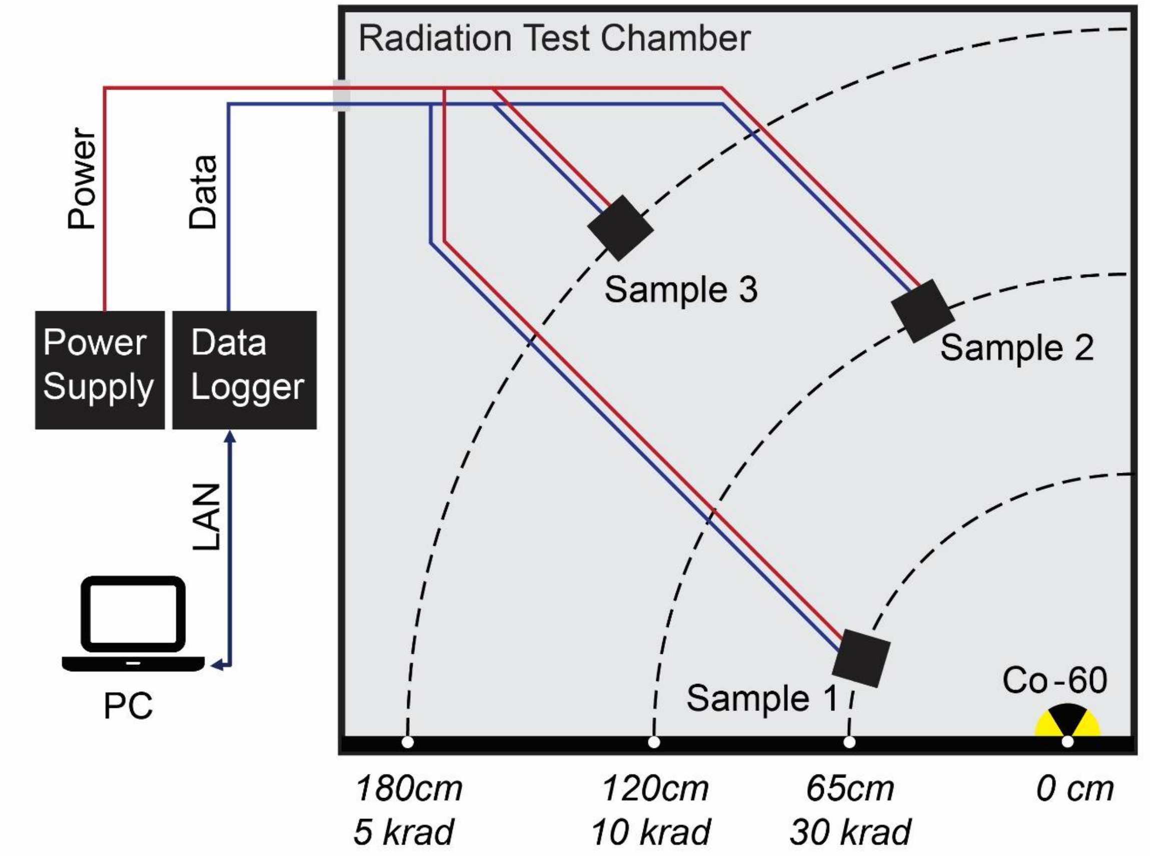


Figure 5-8. TID radiation test setup

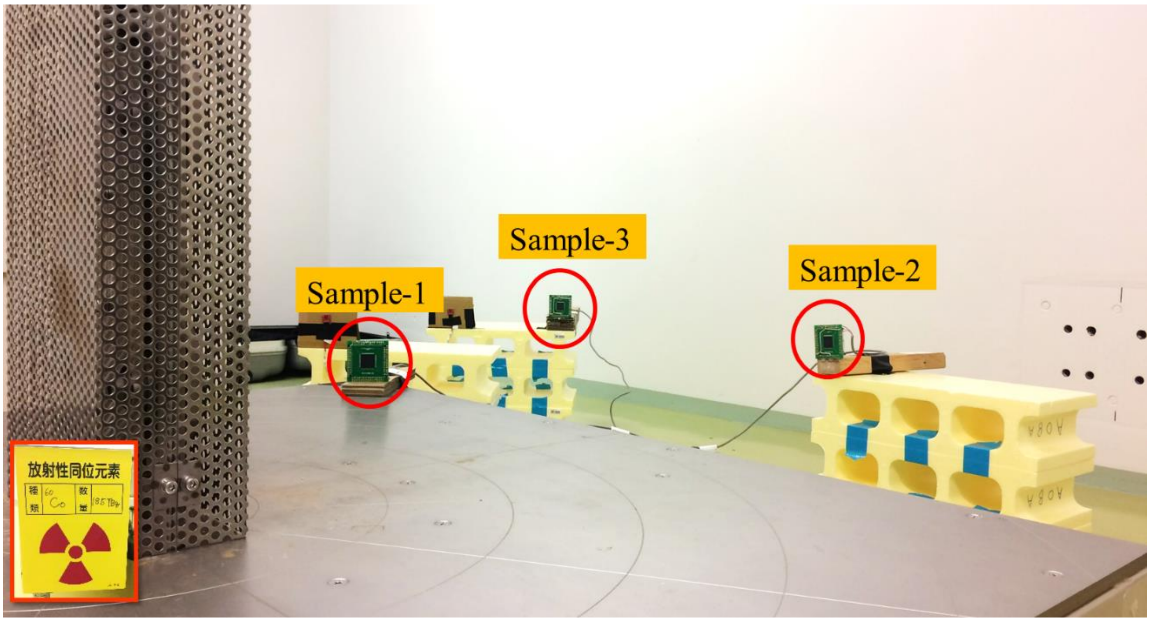


Figure 5-9. Photography of the TID test configuration

The methodology of tests is shown in the Fig.5-10. In order to monitor and record the test results, a signal generator and data logger both implemented on the single Raspberry Pi board was used. CPLD was programmed to make several follower circuits, input-to-output connections as green color on the Fig.5-10. Jumper cables represents purple color in the Fig.5-10, were used to connect one output to the next input of the CPLD making a chain. Only the very first input pin, where the signal comes to the CPLD, and very last output pin, where the signal going out from CPLD, of the chain was connected to the raspberry pi board. Which means that the CPLD has transmitted only the clock signal from the Raspberry Pi, many times through itself, and returned to Raspberry Pi. If there is a problem with one of the connections inside CPLD due to total ionizing dose, the data will be lost. This was the technique that we considered the test result as a "failure" if data was lost.

Table 5-2. Radiation dose estimated at a different distance from the radiation source

Test Article	Sample No.	Distance from the radiation source (*cm*)	Radiation dose (*krad*)
LC4256ZE7TN144I	1	65	30
	2	120	10
	3	180	5

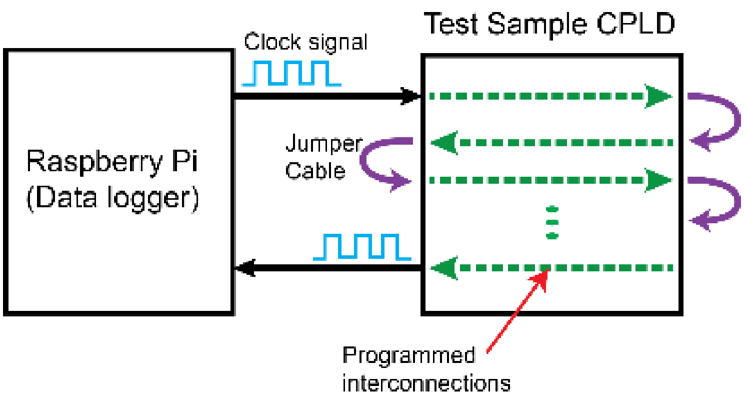


Figure 5-10. Method of the TID test

5.3.1.1. Result of the TID test

All of the CPLDs operations were normal under the different radiation dose levels up to 30 Krad, which was 3 times higher than the unit qualification test level of ISO standard [57]. No failure was found during the test. Based on the test result, we can say that the selected CPLD for the SoftCIB has enough strength against radiation to survive in LEO (Low Earth Orbit) within an average lifetime of CubeSat.

5.3.2. Single Event Effect Test

Another effect of the space radiation that may cause failure in the electronics device is the Single Event Effects (SEE). In contrasts to the cumulative effect of radiation such as TID, the SEE is induced by a single particle. SEE occurs if high energy particles pass through a sensitive volume of an electronic device, and deposit enough charge to interrupt the operation of an instrument. The results vary from recoverable effects to catastrophic system failures, depending on the vulnerability of the electronic component, the operation of the component and the timing of SEE [58]. Therefore many different effects that can be the result of SEE which namely [59]:

- Single Event Latch-up (SEL)
- Single Event Upset (SEU)
- Single Event Functional Interrupts (SEFI)
- Single Event Burnout (SEB)
- Single Event Gate Rupture (SEGR)

The CPLD must also be tested for SEEs because the SoftCIB has the potential single point of failure. The SEU and SEL were the focus of the test. Because we determined that they have the potential high impact on SoftCIB operation.

The test was performed with a californium-252 (252Cf) radioisotope using the methodology described in this [60] work. Since a 252Cf testing facility is more conveniently available and less costly than a particle accelerator testing plant. This particular test has done at the Kyoto University Research Reactor Institute.

The purposes of the test were:

- To detect the SEL through current measurement.
- To observe SEU in the non-volatile configuration memory by bit changes,
- To confirm that the CPLD could recover via power reset from an SEL
- To estimate the rate of minimum SEL occurrence in orbit.

The test setup and photography of the general external view of the test chamber is shown in Fig.5-11, and 5-12. The radiation source was placed over the test article inside the vacuum chamber after pressure decreases lower than 30 Pa. To verify the operation, read and write all configuration bits to the CPLD memory, the CPLD was connected to the PC via the JTAG debugger. By comparing the data of the CPLD memory before and after radiation exposure, we can identify an SEU event. The current is measured by a current sensor and its output is monitored on the oscilloscope and PC using DAQ which outside to the chamber. The relay switch was employed for the PC-controlled power reset. Photography of the chamber's internal view can be seen from Fig.5-13 after the setup completed. Radioactivity of 252Cf was 13.43 μCi and a circle with 10-mm radius freed 3.19 x104 ions in every second. It is estimated that the ion flux at CPLD, which is 10 mm from the source), was 593 (ions/cm^2/s) with 3 percent error margin. The top of the plastic package of the CPLD was removed due to the heavy ions from 252Cf cannot penetrate this package. The test article was an LC4256ZE7TN144I device from the ispMACH4000ZE family CPLD of the Lattice Semiconductor. Figure 5-14 shows the photograph of the CPLD under the test. We can see that the actual area of the semiconductor was about 3.75 mm × 3.25 mm.

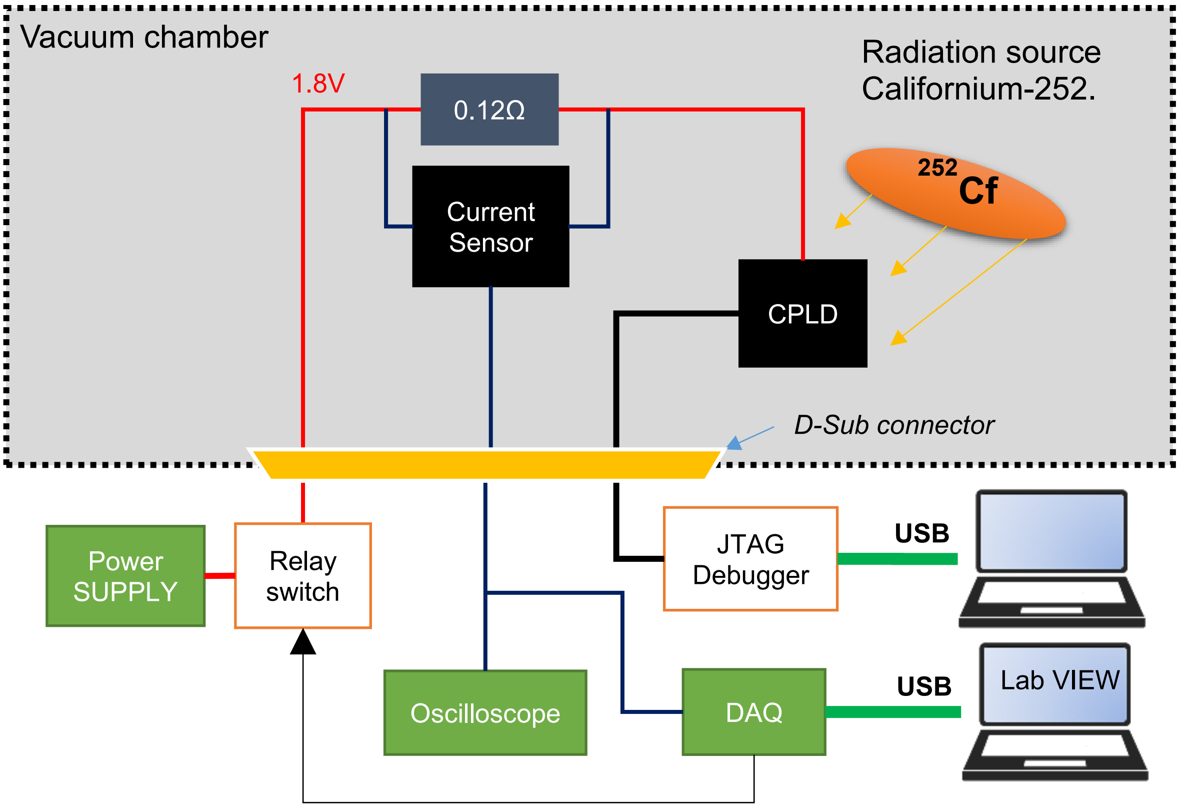


Figure 5-11. Schematic of the ^{252}Cf test setup

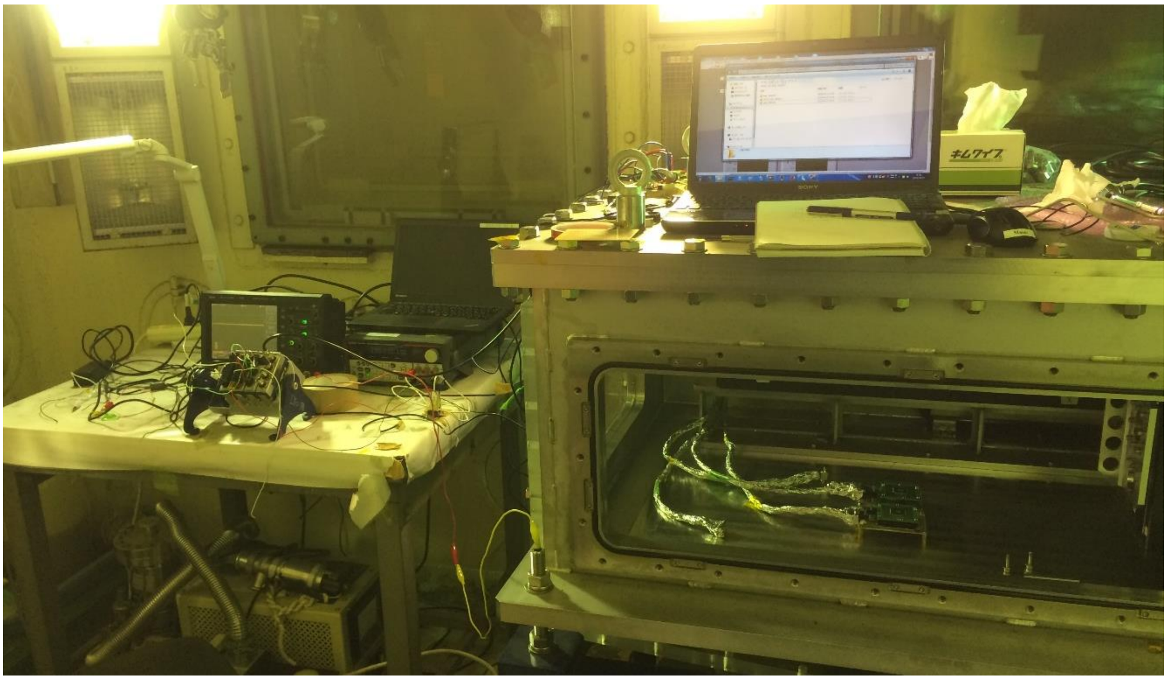


Figure 5-12. General view of the SEL test chamber

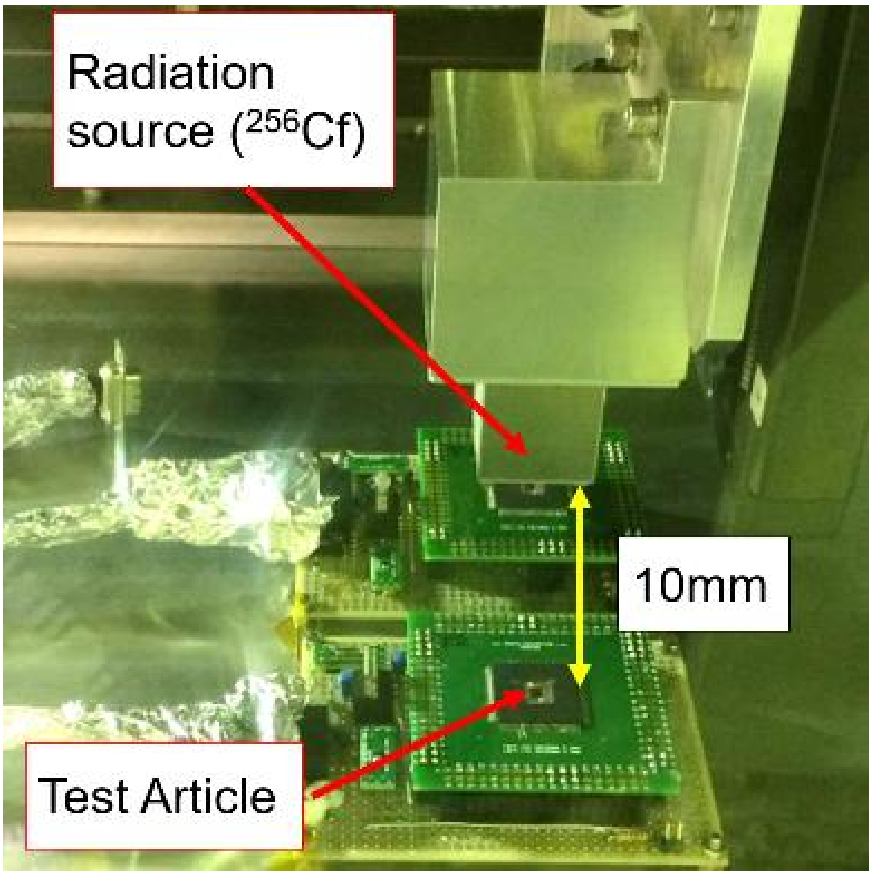


Figure 5-13. Internal view of the SEL test chamber

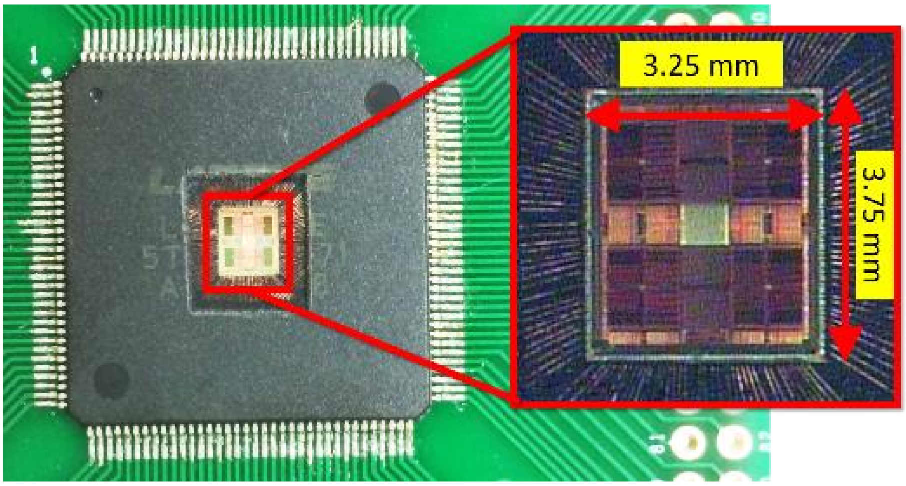


Figure 5-14. CPLD after the plastic cover removed

5.3.2.1. Test Result

During the test, no SEUs were observed. After exposure to radiation, the memory bits were not changed. Ions of 252Cf fission are typically above the SEU threshold level, with a mean Linear Energy Transfer (LET) of 43MV/mg/cm2 [61]. The CPLD was bombed in total $1{,}82 \times 10^6$ ions during the test, but there was no observation of SEU. The cross-section of the SEU, therefore, is lower than 1.37×10^{-7} (1/ion/device). Due to the fact that protons – the major radiation particle causing SEU in LEO – can't simply compute SEU events from a heavy ion particle test result. It is very promising that no SEU has been observed.

However, SEL has been observed multiple times. Figure 5-15 illustrates an example of current increases due to a SEL event during the test. An SEL in the graph happened at 92 s. Before the SEL, current consumption was 4 mA, and after the SEL, it was increased to 34 mA. This example shows an approximate 30-mA current jump. Table 5-3 summarizes the results of all observations. There are two categories of current jumps, approx. 30 mA and 14 mA. For the event of SEL mean time is respectively 582 and 1529 s. Figure 5-16 shows a histogram of the jumps.

The minimum SEL in orbit is calculated by replacing our value with that of the previous H8 microcontroller test [60]. For the H8 microcontroller, the relationship between the cross-sections in the 252Cf test and in orbit is used as a reference. The SEL rate in one year on orbit is estimated at those to current jump separately. The probability of the number of event in orbit was 0.0027 by 14 mA current jump, 0.0072 at the 30 mA current jump. The behavior of the SEL event was as predicted; after the current jump, current values remains at that high level.

By completely cutting the power of CPLD, the current going back to the normal state as it was consumed before the event. We could not detect the Single Event Functional Interrupts or any malfunctions during the test.

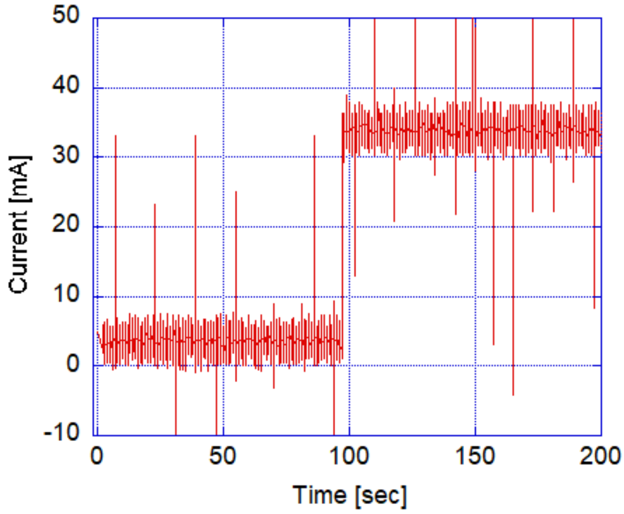


Figure 5-15. An example of the current increase due to SEL

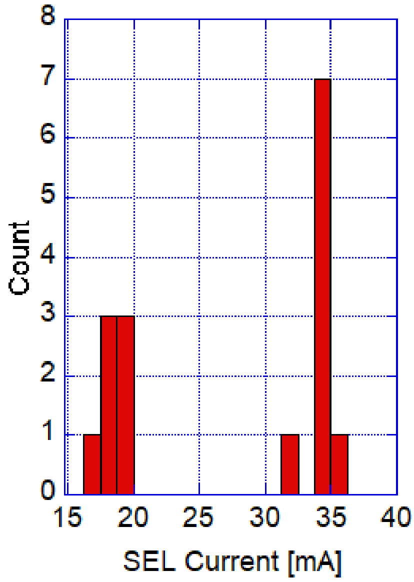


Figure 5-16. Histogram of the current jumps of the CPLD

Table 5-3. Summary of the SEL test result

Target test Sample	Total exposure time (hour)	Nominal current (mA)	SEL current [mA]	Time until SEL occur (sec)
T1	3	4	17	3540
			34	1440
			19	1200
			32	1020
T2	1	4	34	252
			18	487
			34	282
			36	152
			18	568

T3	2	4	34	47
			19	894
			18	1638
			34	850
			34	414
			19	2378
T4	1	4	34	779

5.4. Thermal Vacuum Test

The Thermal Vacuum Test (TVT) for SoftCIB have done at the system level, after integration with the BIRDS-3. The TVT test has done twice as an Engineering Model (EM), and Flight Model (FM) of BIRDS-3 CubeSat using a small vacuum chamber in the Center for Nanosatellite Testing (CENT) at Kyutech. Figure 5-17 and 5-18 illustrated the test configuration and a photograph of the general view of the EM testing. Thermocouples have attached to the satellite to monitor the temperature. From the internal components including CPLD of SoftCIB to the external panels of the satellite, a total 20 thermocouples were attached (see Fig.5-19). The DAQ that connected to the PC was used to collect pressure and temperature data. There are two external power supplies; one was used to supply the heater (PS1 in Fig. 5-17), and another one was used to supply the satellite instead of solar panels (PS2 in Fig. 5-17). As ON/OFF control for the satellite, a deployment switch was used. All the functional tests were controlled from Ground Station (GS) which placed beside the thermal vacuum chamber. The UHF transceiver of the satellite was connected to the ground station via RF cable with an attenuator. During the test in all conditions, the chamber pressure was lower than 1 x 10^{-3} Pa.

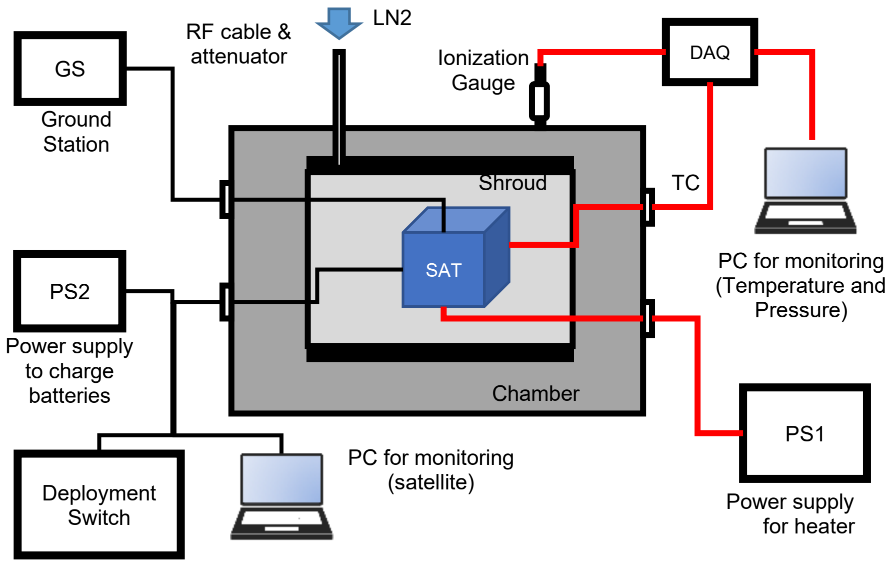


Figure 5-17. Schematic of TVT test setup

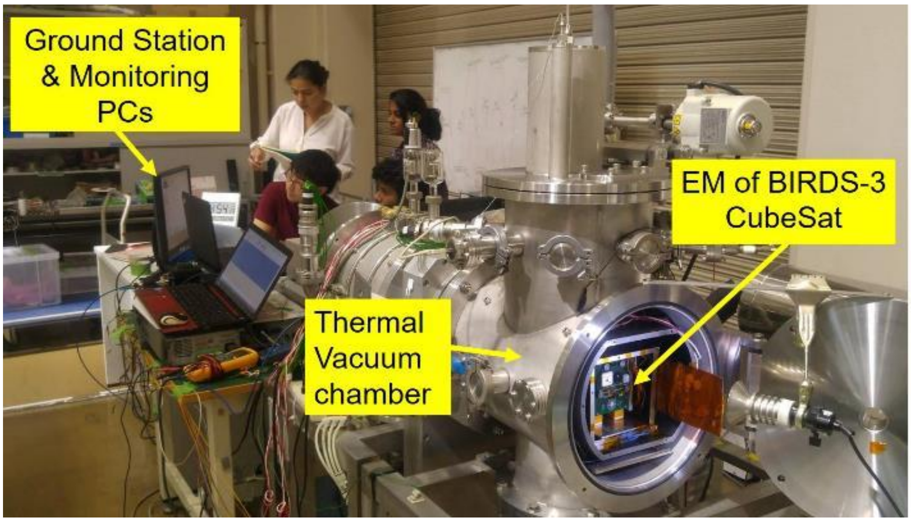


Figure 5-18. Photography of the general view of the TVT

The chamber temperature was controlled by an external panel's temperature and a total of four thermal cycles have done for EM testing. The highest temperature was 55ºC, and the lowest was -45 ºC. Before doing the functional tests at hot and cold conditions, the chamber had soaked for 30 minutes. The actual temperature profile is displayed in Fig.5-21 during the EM test. The last cycle of the test was extended to check the battery operation in minus 10 degree Celsius. Dark green color represents the Minus_Y in the Fig.5-21, and it was the controlling temperature where thermocouple attached to an external panel of –Y side of the satellite. Other temperatures of the satellite subsystems are also shown in Fig.5-21, for example, temperatures of the OBC, UHF transmitter, MSN-2, and the battery box were illustrated. The red color of the graph displays the temperature of the CPLD. The CPLD temperature was -42°C at coldest and 67°C at hottest during the test. All of the functional tests succeeded and there was no anomaly with the CPLD. The result of the functional test is shown in Table 5-4. The indicator "O" means that there were no errors or abnormality, "Δ" means that there was a partial failure. We can see that the fourth row of the table, the function "Interface between OBC and other Subsystems" had some partial failure during the test. Abnormality was investigated and it was not related to SoftCIB.

Table 5-4. Summary of the functional test results of TVT

Sl. No	Function	Details	Check (Overall)
1	SW2		O
2	Serial monitor	Main PIC	O
3	The interface between OBC and other Subsystems	OBC (60°C)	Δ
4	Reset	Reset PIC ←→ Main PIC	Δ
5	Battery Heater	ON/OFF	O

6	Beacon transmission (CW)	Received at GS	O
7	HK data		O
8	Uplink Command	CAM: activate/ deactivate	O
9		LDM	Δ
10		ADCS	O
11	Downlink	CAM: activate/ deactivate	O
12		LDM	Δ
13		ADCS	O
14	Antenna Deployment	In the worst cold (-X panel: -42°C)	O

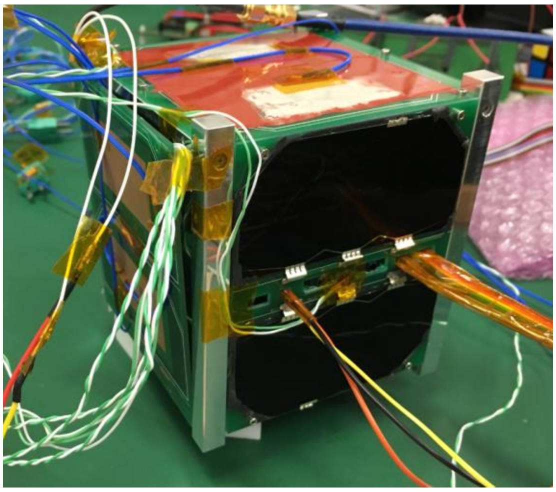


Figure 5-19. After the thermocouples were attached to the BIRDS-3 EM

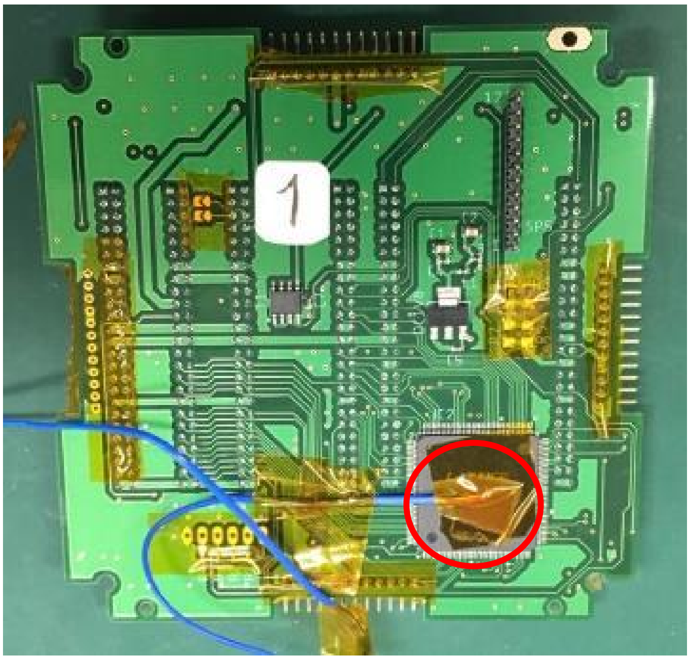


Figure 5-20. Thermocouple attached on the CPLD of SoftCIB

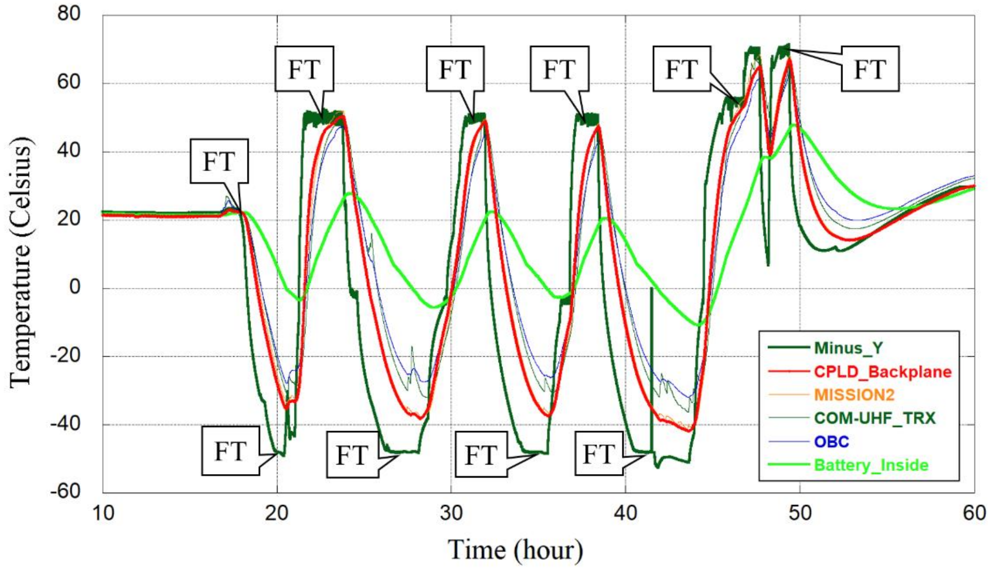


Figure 5-21. The temperature profile of the TVT

5.5. Vibration Test

The vibration tests were also not conducted at the component level or subsystem level. Instead, the test has done at the system level which was EM and FM of BIRDS-3. As it is CubeSat, subsystem failure can be easily identified after the vibration, even the CubeSats completely assembled or tested with the whole system at once. Figure 5-22 shows the photography of three FM CubeSats of BIRDS-3 integrated into the single jig and placed on the vibration shaker.

Random vibration and a quasi-static acceleration test were performed using the Qualification Test (QT) level JEM Payload Accommodation Handbook[15] for HTV, SpaceX and Orbital Cygnus launcher profiles. The overall Grms was 6.8 for this random vibration test, while the quasi-static acceleration in all directions was 22.6(G). Test levels are shown in Table 5-5. In all axes before and after vibration tests, there were no significant changes in structural natural frequency. Prior to and after vibrational testing, functional tests have been done and no anomalies have been identified.

Figure 5-22. Flight Models of BIRDS-3 CubeSats installed on the vibration shaker with a single 3U jig.

Table 5-5. Vibration test levels used for EM of BIRDS-3

RANDOM QT			
Direction	**Freq. [Hz]**	**QT**	
		Time [sec]	**Overall Grms value [Grms]**
Vertical axis (Y)	20~2000	120	6.8
The horizontal axis (X, Z)	20~2000		6.8 6.8
SINE BURST QT			
Direction	**Freq. [Hz]**	**QT**	
		Number of waves	**Acceleration [G]**
The vertical axis (Y)	10~40	10 or more	22.6
The horizontal axis (X, Z)	10~40		22.6 22.6

6. On-orbit demonstration

6.1. Implementation to the BIRDS-3 Project

BIRDS-3 project, the successor of the BIRDS-1 and BIDRD-2, at Kyutech have implemented the SoftCIB for their mission. The project BIRDS-3 is a constellation of three 1U-CubeSats that was built and designed by students who came from four different countries including Nepal, Sri Lanka, Bhutan, and Japan [62], [63]. A total of 5 missions will be carried out by BIRDS-3; demonstration of the LoRa Module (LDM) on-orbit, Imaging Earth from the space (CAM), Active Attitude Determination and Control Mission (ADCS) using 3-axis Magnetorquer, demonstration of SoftCIB on orbit, and Glue Mission which aim to replace space-qualified adhesive, Room Temperature Vulcanizing-Silicon (RTV-S) 691, by commercial alternative that is cheaper and more available on the market.

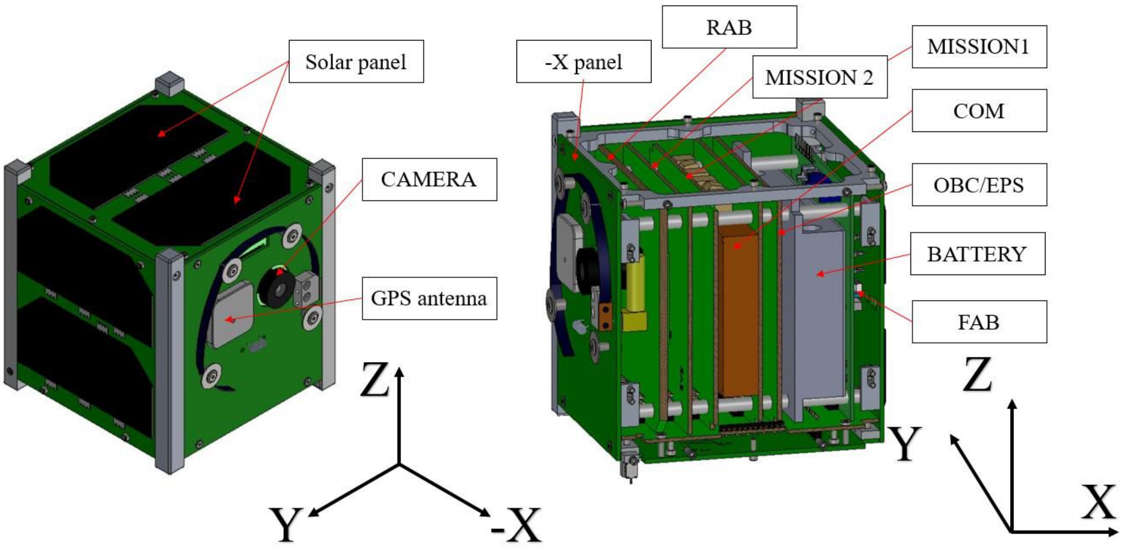


Figure 6-1. An external configuration (left), and internal configuration (right) of the BIRDS-3.

Three CubeSats had the same design except for backplanes. An external and internal configuration of the BIRDS-3 is shown in the Fig.4-5. Two satellites of the BIRDS-3, NepaliSat-1, and Raavana-1, equipped with a normal backplane which is similar to the BIRDS-1 and -2. Another satellite named “Uguisu” carried the SoftCIB. However, the definition of the interfaces is exactly the same to each other. Flight Models of the BIRDS-3 CubeSats shown in the Fig.4-6. The comparison image of the backplanes boards is shown in the Fig.4-7. During the development of the Engineering Model (EM) and Flight Model (FM), a number of tests have been done including space and launch environmental tests. No anomaly was found due to SoftCIB. The maximum power consumption of this SoftCIB was approximately 40 mW.

Figure 6-2. Flight Model of all BIRDS-3 CubeSats; Raavana-1 (left), NepaliSat-1(middle), Uguisu (right)

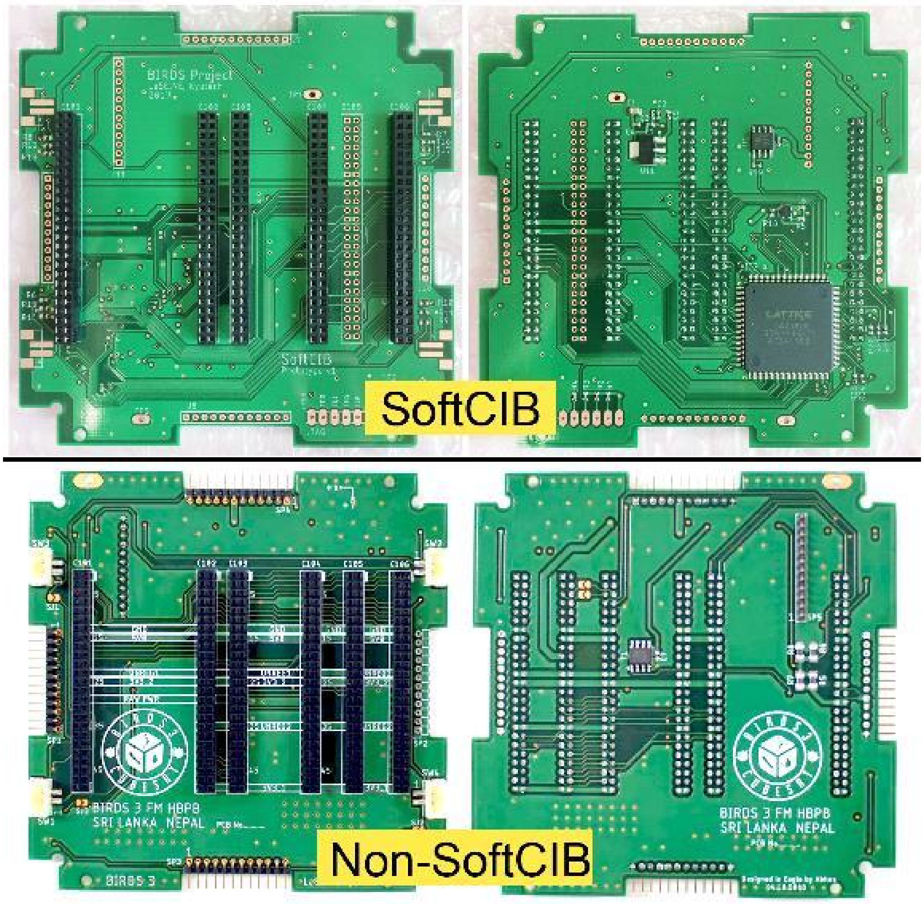


Figure 6-3. Comparison of the Backplane boards of the BIRDS-3.

6.2. The SoftCIB of BIRDS-3

Uguisu satellite of BIRDS-3 uses backplane, however, not all the CPLD pins were used. Total 25 digital connection (consumed 50 pins of the CPLD) were handled by CPLD on the SoftCIB for BIRDS-3 and 58 pins of CPLD were not used. The CPLD plays an important role to connect between different mission payloads and OBC. For example, UART communication between LDM mission and OBC, UART communication between ADCS and OBC, UART communication between CAM and OBC, SPI communication between Flash memory and OBC, UART communication between CAM and external access port were made by CPLD as it programmed. Figure 6-4 shows the diagram of the connections made by CPLD for Uguisu satellite. Four UART connections which the each UART have Tx and Rx, one SPI with four bus lines and one digital High/Low signals were implemented through CPLD. The rest of the interface connections of the Uguisu satellite were directly routed (hardware) on the backplane PCB board. Only the UART for accessing from PC is not used in space, while all other connections are going to be tested in space.

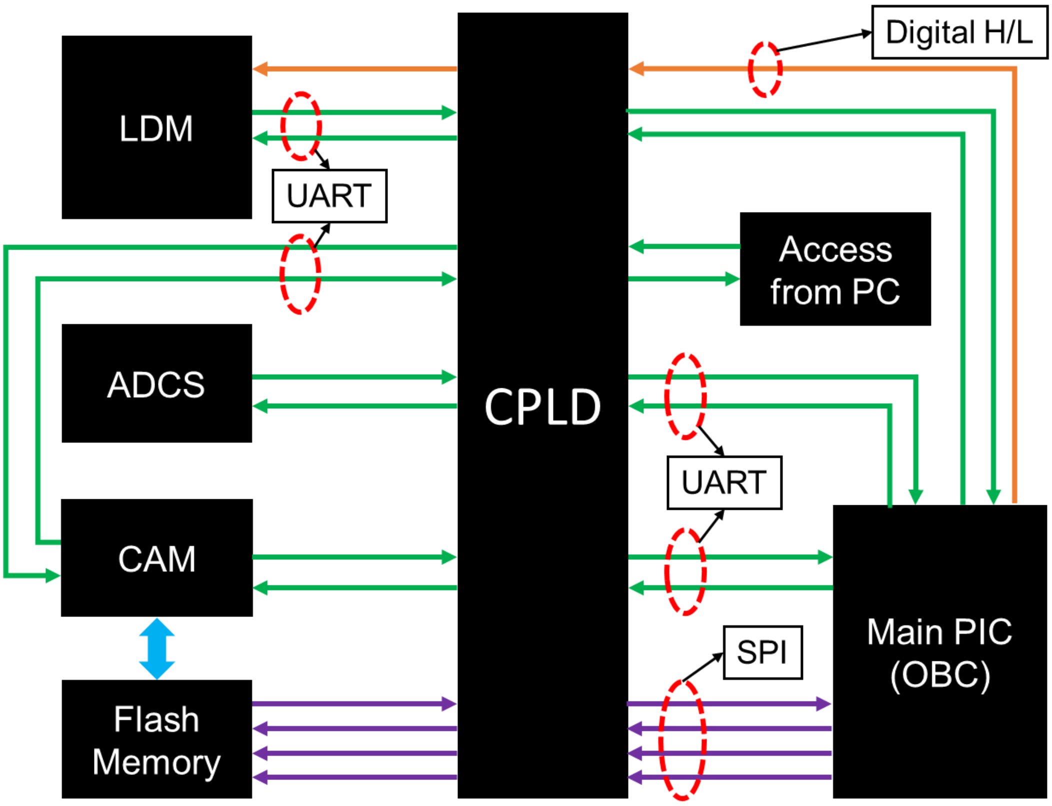


Figure 6-4. Diagram of the CPLD connection of BIRDS-3.4

The basic method for the on-orbit testing is simply activate the mission or subsystem by commanding from Ground Station and download the data to Ground Station via UHF band.

Analyzing the data for each mission or subsystem after telemetry data downloaded to the Ground Station, we can say that the communication between subsystems was successful or not.

6.3. Launch, deployment, and operation of BIRDS-3

BIRDS-3 CubeSats have launched to the ISS by CRS-11 mission with Antares rocket in April 2019 [64]. Satellites injected into space using the Japanese Experiment Module (Kibo) of the International Space Station on 17th of June, 2019 [65]. Satellites are operating from BIRDS ground station network which has many ground station in BIRDS partner countries all over the world. The ground station at Kyutech is acting as a primary ground station of the satellite operation.

Figure 6-5. Antares rocket launched on 17th April, 2019 carrying Cygnus Cargo Ship. (Image credit: Bill Ingalls/NASA)

Figure 6-6. Cygnus Cargo Ship arrived to ISS on 19th April, 2019 [64].

Figure 6-7. Deployment of BIRDS-3 satellites from ISS. NepaliSat-1 was first to deploy, followed by Raavana-1 and Uguisu satellite [65].

Figure 6-8. UHF antenna for BIRDS ground station at Kyutech.
BIRDS-3 uses UHF band for uplink and downlink. Image source [66].

Figure 6-9. Team members succeeded first uplink command on 19th of June, 2019.

Signals received from all three satellites in multiple ground stations of the BIRDS Ground Station Network. The first uplink command was successful on the 19th of June at Kyutech Ground Station [66].

6.4. Result of on-orbit demonstration

Each mission of the Uguisu satellite that using software-defined interface with SoftCIB have performed to check the operation in space. The operation report shows that the CPLD is properly working in space just like the test on the ground. The data downloaded on the ground from CAM mission as separate packets of pictures in few passes over the ground station. Later the data combined and recreated the image correctly. This shows that the CPLD has done its job properly in space. With CAM mission data it is clear that UART communication between CAM and OBC, and SPI bus communication between Flash memory and OBCs were worked. The example of low-resolution image downloaded on the ground station from Uguisu satellite is shown in Fig.6-10. Similar to CAM mission, other missions are also successfully demonstrated in space and no issues were detected due to the interface. To prove the successful communication between subsystems through the CPLD, a number of data packets were downloaded from each address of the mission memory and decoded on the ground. For example, part of data recorded by ADCS mission is shown in the Table 6-1. This data included the Earth's magnetic field and satellite's rotation for each axes with a timestamp which were measured by the magnetometer and gyroscope onboard the satellite. The summary of the all result from the on-orbit demonstration are shown in the Table 6-2.

Table 6-1. Example of the data downloaded from the ADCS subsystem.

Time [sec]	**MAG_X (mG)**	**MAG_Y (mG)**	**MAG_Z (mG)**	**GYRO_X (deg/s)**	**GYRO_Y (deg/s)**	**GYRO_Z (deg/s)**
8926	-244.732	339.648	-163.968	5.6	-3.3	13.8
8927	-185.684	378.932	-124.684	5.3	-3.3	14.0
8929	-128.832	386.252	-92.964	5.5	-3.2	13.8
8930	-51.972	362.096	49.044	5.4	-3.0	13.8
8932	0.732	316.712	-43.188	5.5	-3.1	14.1
8933	38.796	247.416	-40.748	5.3	-3.0	14.0
8935	47.092	177.876	-53.68	5.5	-3.2	14.1
8936	26.108	103.944	-83.448	5.5	-3.1	14.0
8938	-10.736	59.048	-112.972	5.8	-3.0	14.1
8939	-78.812	29.524	-154.452	5.7	-3.3	14.1
8941	-137.616	33.428	-184.708	5.5	-3.2	14.1
8942	-209.596	70.516	-214.476	5.9	-3.2	14.2

Figure 6-10. The example of an image that downloaded from the Uguisu satellite.

Table 6-2. Summary of the on-orbit demonstration result

Connections through the CPLD	Result
UART communication between LDM mission and OBC	Successful. Proven by an acknowledgement from the satellite via UHF.
UART communication between ADCS and OBC	Successful. Data successfully downloaded on the GS and analyzed.
UART communication between CAM and OBC	Successful. Images were taken. Proven by the downloaded image on the GS.
SPI communication between Flash memory and OBC	Successful. Images downloaded and recreated correctly on the ground.
LDM mission ON/OFF switch signal (Digital High/Low signal)	Successful. The mission started successfully by a command from the GS. Proven by an acknowledgement from the satellite.

7. Conclusions

A CubeSats have been built by many different entities to achieve a wide range of missions. A growing number of launches are not decreasing in the near future. The market scale of this field is getting bigger and bigger. Even though, the initial purpose of the CubeSats is still kept as important. A lot of universities and institutions were involving into this job, building a CubeSats, with the workforce who has not professional yet. Many of their CubeSats' mission has failed after launch. A highly important factor that influences the success of CubeSats is time constraints. CubeSat projects need to spend more times to testing than development and assembly. Because of limited time, on ground testing was not complete enough to find out the potential failure after launch in many projects. The author determined that the "modular approach" could help to solve this issue, and intensively studied the topic which has significant benefits to the technology and market. Also, it was found that the important key to succeeding in modularity is the interface standardization, in general. However, not all the interface standards are good, some may not support or promote the advancement of technology. People in the field of space industry have already realized the importance of modularity and interface standardization. Many attempts have done, and many projects were implemented before, in the space industry. The CubeSat was the successful example of the interface standardization. However, inside the CubeSat, not external interface, has not been achieved a big success in the author's point of view. There is room to cut the times of development and integration by implementing a standardized internal board which could be utilized to the different CubeSat mission with different payload.

This work introduces a new method with the backplane interface board (so-called SoftCIB) to reduce the times of development and assembly of the CubeSat project. The SoftCIB was also aimed to reduce workmanship error due to harnessing. This new interface board has developed, tested and implemented to the real CubeSat project.

The key technology to the SoftCIB is the use of CPLD. With the help of the programmable logic device, the user of the SoftCIB can change the interface connection in a very short time, practically less than one hour, by reprogramming the device. Otherwise, the interface changing on the backplane board takes a significant amount of time. Sometimes, the interface changing issue can arise suddenly during development. In most cases, including BIRDS-1 and BIRDS-2 CubeSat projects, interface changing issues were uncertain at the beginning of the project. The reason for that was most of the members of those projects were not professional and never involved before building a satellite. When the interface board needed to change during the development, it took time to manufacture new PCB. Again, it took time to test a new board.

The 42% of all data lines on the interface board became programmable. There are some permanently routed connections on the PCB for the power connection to the subsystems and for the critical signal lines to communication subsystems.

Since, developing a new interface board which consists of COTS component, certain design considerations and performance issues should be addressed. For example, CPLD power consumption was the biggest question due to limited power budget of the CubeSat. The results of the tests showing that the power consumption is an acceptable range as it is 40mA. Other questions also addressed, for instance, the signal delay between inputs and outputs was approximately 9ns, and highest SPI signal speed that CPLD can handle without interruption was 4MHz. SoftCIB functions are tested with different subsystem boards and result shown that CPLD is capable of doing the "software routing" for simple digital logic signals (High/Low), UART, and SPI signals.

Furthermore, space environmental test has done to ensure the SoftCIB withstand the space condition. The operation of the SoftCIB have checked under thermal vacuum condition at lowest -42°C and at highest 67°C. The CPLD of the SoftCIB can take radiation dose up 30krad without failure. Which means that the SoftCIB can withstand space radiation in LEO for average CubeSat mission lifetime. It was proven that the SEL event will occur in CPLD, by heavy-ion test. The current was increased due to SEL and it kept drawing the current until the power completely cut. However, the increase of current was not severe and it was 14mA to 30mA. Based on the previous study that has done at Kyutech, the on-orbit probability for SEL have calculated. The estimated number of SEL event on orbit per year was 0.0072. The SEU has never detected by heavy-ion test.

The SoftCIB has integrated with one of the BIRDS-3 CubeSat. And tested for the launch environment. Random vibration and a quasi-static acceleration test were performed at the system level. The overall Grms was 6.8 for random, 22.6 G for the quasi-static acceleration in all directions. BIRDS-3 satellites launched by CRS-11 mission with Antares rocket in April 2019 to the ISS. They will be deployed about the time of this thesis been submitted. Based on the BIRDS-3 result, we can add the CPLD (ispMACH®4000ZE family 4256ZE-7TN144I) to space tested COTS component list.

7.1. Future work

The SoftCIB is designed and implemented for the 1U CubeSat form factor. Scaling the SoftCIB for bigger CubeSat form factors should be the study further. Theoretically, the advantages shall be the same as it is bigger. However, the design concern, especially mechanical dimensions may become difficult or more thoughtful.

So far, SoftCIB interface tested for the digital logic signals, UART, and SPI communication protocols. The capability of the SoftCIB for other bus communication, such as I2C and CAN shall be studied further.

It may not enough to see the result by implementing it to only one CubeSat project, thus, at least one more CubeSat should experience the SoftCIB to make it good standardized.

Finally, the method (software reconfigurable interface) used for SoftCIB can be used to other space missions dismissing backplane board. Using this method, people may build the satellite which can configure its interconnections on-board, and on-orbit (or after assembled) to accomplish their mission.

8. References

[1] Cal Poly SLO, *CubeSat Design Specification*, REV 13., vol. Rev. 13. San Luis Obispo, CA, 2015.

[2] M. Swartwout, “The first one hundred CubeSats : A statistical look,” *J. Small Satell.*, vol. 2, no. 2, pp. 213–233, 2013.

[3] M. Cho, M. Hirokazu, and F. Graziani, “Introduction to lean satellite and ISO standard for lean satellite,” *RAST 2015 - Proc. 7th Int. Conf. Recent Adv. Sp. Technol.*, pp. 789–792, 2015.

[4] J. Bouwmeester and J. Guo, “Survey of worldwide pico- and nanosatellite missions, distributions and subsystem technology,” *Acta Astronaut.*, vol. 67, no. 7–8, pp. 854–862, 2010.

[5] K. Woellert, P. Ehrenfreund, A. J. Ricco, and H. Hertzfeld, “Cubesats: Cost-effective science and technology platforms for emerging and developing nations,” *Adv. Sp. Res.*, vol. 47, no. 4, pp. 663–684, 2011.

[6] M. Swartwout and C. Jayne, “University-Class Spacecraft by the Numbers: Success, Failure, Debris. (But Mostly Success.),” *Proc. AIAA/USU Conf. Small Satell. Conf. Small Satell.*, no. August, 2016.

[7] A. K. Nervold, J. Berk, J. Straub, and D. Whalen, “A Pathway to Small Satellite Market Growth,” *Adv. Aerosp. Sci. Technol.*, vol. 1, no. June, pp. 14–20, 2016.

[8] C. C. Venturini, M. Tolmasoff, and R. D. Santos, “Improving Mission Success of CubeSats,” in *U.S. SPACE PROGRAM MISSION ASSURANCE IMPROVEMENT WORKSHOP*, 2017.

[9] Erik Kulu, “Nanosatellite and CubeSat Database.” [Online]. Available: http://www.nanosats.eu/. [Accessed: 16-Dec-2017].

[10] M. Swartwout, “CubeSat Database.” [Online]. Available: https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database. [Accessed: 17-Dec-2017].

[11] J. Bouwmeester, M. Langer, and E. Gill, “Survey on the implementation and reliability of CubeSat electrical bus interfaces,” *CEAS Sp. J.*, vol. 9, no. 2, pp. 163–173, 2017.

[12] S. Busch, “Robust, Flexible and Efficient Design for Miniature Satellite Systems,” Julius Maximilian University of Würzburg, Germany, 2016.

[13] UNISEC Europe, “CubeSat Subsystem Interface Definition (Proposal).” pp. 0–10,

2015.

[14] T. R. Tejumola, M. B. Project, P. Birds, and M. Cho, “Joint Global Multi-Nation Birds ; Developing Nations ’ Testbed for Space Technology towards Sustainable Space Program,” in *7th Nano-Satellite Symposium*, 2016, pp. 1–5.

[15] Japan Aerospace Exploration Agency (JAXA), *JEM Payload Accommodation Handbook Small Satellite Deployment Interface Control Document*, vol. 8. 2015.

[16] T. Tumenjargal, S. Kim, H. Masui, and M. Cho, “CubeSat bus interface with Complex Programmable Logic Device,” *Acta Astronaut.*, vol. 160, no. April, pp. 331–342, Jul. 2019.

[17] J. R. Wertz, D. F. Everett, and J. J. Puschell, *Space mission engineering : the new SMAD*. Microcosm Press, 2011.

[18] M. Cho and F. Graziani, Eds., *Definition and Requirements of Small Satellites Seeking Low - Cost and Fast - Delivery*. International Academy of Astronautics, 2017.

[19] Cal Poly, “CubeSat.” [Online]. Available: http://www.cubesat.org/. [Accessed: 06-Aug-2018].

[20] NASA, “Basic Concepts and Processes for First-Time CubeSat Developers. NASA CubeSat Launch Initiative,” no. October. p. 86, 2017.

[21] J. Cutler and G. Hutchins, “OPAL : Smaller , Simpler , and Just Plain Luckier,” *14th Annu. Conf. Small Satell.*, p. 4, 2000.

[22] J. Puig-Suari, C. Turner, and W. Ahlgren, “Development of the standard CubeSat deployer and a CubeSat class PicoSatellite,” in *IEEE Aerospace Conference Proceedings*, 2001, vol. 1, pp. 347–353.

[23] “ECM space technologies GmbH.” [Online]. Available: https://www.ecm-space.de/index.php/launch-adapters-h/cubesat-sizes. [Accessed: 26-May-2019].

[24] H. Heidt, J. Puig-Suari, A. S. Moore, S. Nakasuka, and R. J. Twiggs, “CubeSat: A new Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation,” *AIAA/USU Conf. Small Satell.*, pp. 1–19, 2000.

[25] J. K. Gershenson, G. J. Prasad, and Y. Z. H. A. Ng, “Product modularity : definitions and benefits,” no. September, pp. 295–313, 2003.

[26] C. Y. Baldwin and K. B. Clark, “Managing in an Age of Modularity,” *Harv. Bus. Rev.*, 1997.

[27] C. Huang and A. Kusiak, “Modularity in Design of Products and Systems,” *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, vol. 28, no. 1, pp. 66–77, 1998.

[28] G. Erixon and A. Arnstrom, “Modularity - the Basis for Product and Factory

Reengineering," *CIRP Ann.*, vol. 45, no. 1, pp. 1–6, 1996.

[29] P. Gu and M. Hashemian, "An Integrated Modular Design Methodology for Life-Cycle Engineering," vol. 46, no. 1, pp. 71–74, 1997.

[30] K. Ulrich, "Fundamentals of Product Modularity," in *Management of Design*, Dordrecht: Springer Netherlands, 1994, pp. 219–231.

[31] R. O. Bartlett and F. J. Cepollina, "The Multimission Modular Spacecraft for the 80's," in *Meeting on Space Shuttle Missions of the 80's; Aug. 26-28,* 1975.

[32] J. R. FALKENHAYN EDWARD, "Multimission modular spacecraft (MMS)," in *Space Programs and Technologies Conference*, American Institute of Aeronautics and Astronautics, 1988.

[33] J. Esper, "Modular, Adaptive, Reconfigurable Systems: Technology for Sustainable, Reliable, Effective, and Affordable Space Exploration," *AIP Conf. Proc.*, vol. 746, no. 1, pp. 1033–1043, Feb. 2005.

[34] J. Esper, J. Andary, J. Oberright, M. So, and J. Hauser, "Modular , Reconfigurable , and Rapid Responsive Space Systems : the Remote Sensing Advanced Technology Microsatellite," *2nd Responsive Sp. Conf.*, no. 301, pp. 1–7, 2004.

[35] G. Cancro, P. Eisenreich, G. Oxton, S. Ling, and K. Balon, "NASA SmallSat modular hardware and software standardization," in *2009 IEEE Aerospace conference*, 2009, pp. 1–19.

[36] R. Sanchez and J. T. Mahoney, "Modularity, flexibility, and knowledge management in product and organization design," *Strateg. Manag. J.*, vol. 17, no. S2, pp. 63–76, 1996.

[37] J. Hsuan, "Impacts of supplier-buyer relationships on modularization in new product development," *Eur. J. Purch. Supply Manag.*, vol. 5, no. 3–4, pp. 197–209, 1999.

[38] G. Tassey, "Standardization in technology-based markets," *Res. Policy*, vol. 29, no. 4–5, pp. 587–602, 2000.

[39] C. Maxfield, *The Design Warrior's Guide to FPGAs*. Elsevier, 2004.

[40] K. Skahill and J. Legenhausen, "VHDL for Programmable Logic," p. 593, 1995.

[41] I. Grout, "Introduction to Programmable Logic," in *Digital Systems Design with FPGAs and CPLDs*, Newnes, 2008, pp. 1–41.

[42] Steve Bible, "Chips in Space: Let's look inside ARISSat-1 (part 1) | EE Times," 2011. [Online]. Available: https://www.eetimes.com/author.asp?doc_id=1285317#. [Accessed: 19-May-2019].

[43] M. T. Severance, J. Tate-Brown, and C. L. McArthur, "NASA Education Activities on

the International Space Station: A National Laboratory for Inspiring, Engaging, Educating and Employing the Next Generation," Sep. 2010.

[44] P. Cabo and I. Lora, "'OPTOS: A pocket-size giant' (Mission, Operation & Evolution)," in *Proceedings of the 23nd Annual AIAA/USU Conference on Small Satellites*, 2009.

[45] I. Lora, "INTA PICOSATELLITE OPTOS MISSION & SUBSYSTEMS," in *4th Annual CubeSat Developers Workshop 2007*, 2007.

[46] "Introduction | Welcome to the FUNcube Web Site." [Online]. Available: https://funcube.org.uk/introduction/. [Accessed: 19-May-2019].

[47] Bob Beatty, "An Introduction to Cubesats."

[48] Q. G. Schiller, A. Mahendrakumar, and X. Li, "REPTile : A Miniaturized Detector for a CubeSat Mission to Measure Relativistic Particles in Near-Earth Space," in *24th Annual AIAA/USU Conference on Small Satellites*, 2010.

[49] D. Gerhardt, S. E. Palo, Q. Schiller, L. Blum, X. Li, and R. Kohnert, "The Colorado Student Space Weather Experiment (CSSWE) On-Orbit Performance," *J. Small Satell.*, vol. 3, no. 1, pp. 265–281, 2013.

[50] P. Ramaprakash, "AAReST MirrorSat Payload Interface Computer Version II," University of Surrey, 2017.

[51] A. C. Salces, S. B. M. Zaki, S. Kim, H. Masui, and M. Cho, "Design, Development, Testing and On-orbit Performance Results of a Low-cost Store-and-Forward Payload Onboard a 1U CubeSat Constellation for Remote Data Collection Applications," in *69th International Astronautical Congress (IAC)*, 2018.

[52] Kyushu Institute of Technology, "BIRDS-2 CubeSat Project," 2017. [Online]. Available: http://birds2.ele.kyutech.ac.jp/. [Accessed: 11-Jan-2018].

[53] K. Aheieva *et al.*, "Space timing reference option for space applications provided by Space Precision Atomic-clock TIming Utility Mission satellite 'SPATIUM,'" in *Joint Conference: 31st ISTS, 26th ISSFD and 8th NSAT*, 2017.

[54] S. S. Arnold, R. Nuzzaci, and A. Gordon-Ross, "Energy budgeting for CubeSats with an integrated FPGA," in *IEEE Aerospace Conference Proceedings*, 2012.

[55] "DigiKey Electronics - Electronic Components Distributor." [Online]. Available: https://www.digikey.com/. [Accessed: 18-May-2019].

[56] T. R. Oldham and F. B. McLean, "Total ionizing dose effects in MOS oxides and devices," *IEEE Trans. Nucl. Sci.*, vol. 50 III, no. 3, pp. 483–499, 2003.

[57] The International Organization for Standardization, "Space systems — Design

qualification and acceptance tests of small spacecraft and units,” *ISO-9683:2017*, 2017.

[58] National Academies of Sciences and Engineering and Medicine, “Testing at the Speed of Light: The State of U.S. Electronic Parts Radiation Testing Infrastructure,” National Academies Press, Washington, D.C., Jun. 2018.

[59] D. Sinclair and J. Dyer, “Radiation Effects and COTS Parts in SmallSats,” *27th Annu. AIAA/USU Conf. Small Satell.*, pp. 1–12, 2013.

[60] T. Tomioka, Y. Okumura, H. Masui, K. Takamiya, and M. Cho, “Screening of nanosatellite microprocessors using californium single-event latch-up test results,” *Acta Astronaut.*, vol. 126, pp. 334–341, 2016.

[61] J. H. Stephen, T. K. Sanderson, D. Mapper, J. Farren, R. Harboe-Sorensen, and L. Adams, “A Comparison of Heavy Ion Sources Used in Cosmic Ray Simulation Studies of VLSI Circuits,” *IEEE Trans. Nucl. Sci. vol. 31, issue 6, pp. 1069-1072*, vol. 31, pp. 1069–1072, Dec. 1984.

[62] A. Maskey *et al.*, “Overview of BIRDS-3 Satellite Project involving Sri Lanka, Nepal and Japan,” in *Proceedings of the 62nd Space Sciences and Technology Conference*, 2018.

[63] “BIRDS 3 PROJECT | Japan, Nepal, Sri Lanka.” [Online]. Available: https://birds3.birds-project.com/. [Accessed: 18-May-2019].

[64] Chris Gebhardt, “NG-11 Cygnus, S.S. Roger Chaffee, brings the science to ISS – NASASpaceFlight.com,” 2019. [Online]. Available: https://www.nasaspaceflight.com/2019/04/ng-11-cygnus-brings-science-iss/. [Accessed: 30-Jul-2019].

[65] Japan Aerospace Exploration Agency, “Four CubeSats successfully deployed from "Kibo"! : Experiment - International Space Station - JAXA.” [Online]. Available: http://iss.jaxa.jp/en/kiboexp/news/190625_jssod11.html. [Accessed: 29-Jul-2019].

[66] G. Maeda, “BIRDS Project Newsletter,” no. 41, pp. 1–121, Jun-2019.

Appendix A

Definition of abbreviations

ADC	Analog to Digital Conversion
ADCS	Attitude Determination and Control System
AFW	Astro und Feinwerktechnik Adlershof GmbH
AMSAT	Amateur Radio Satellite
APRS	Automatic Packet Reporting System
BBM	Bread-Board Module
BCR	Battery Charging Regulator
BGA	Ball Grid Array
CENT	Center for Nanosatellite Testing
CLK	Clock Signal
CMOS	Complementary Metal-Oxide-Semiconductor
COTS	Commercial off The Shelf
CPLD	Complex Programmable Logical Device
CRS	Commercial Resupply Service
CS	Chip Select
CSAC	Chip Scale Atomic Clock
CSSWE	The Colorado Student Space Weather Experiment
DAQ	Data Acquisition
EM	Engineering Model
EPS	Electrical Power System
EUVE	Extreme Ultraviolet Explorer
FAB	Front Access Board
FM	Flight Model
FPGA	Field-Programmable Gate Array
GNSS	Global Navigation Satellite System
GPS	Global Position System
GS	Ground Station
I & T	Integration and Testing
ICD	Interface Control Document
IHU	Integrated Housekeeping Unit
INTA	Instituto Nacional de Tecnica Aerospacial
ISIPOD	ISIS Payload Orbital Dispencer

ISIS	Innovative Solutions In Space
ISO	International Standardization Organization
ISS	International Space Station
I2C	Inter-Integrated Circuit
JAXA	Japan Aerospace Exploration Agency
JHU/APL	Applied Physics Laboratory of John's Hopkins University
J-SSOD	JEM Small Satellite Orbital Deployer
JTAG	Joint Test Action Group
LEO	Low Earth Orbit
LET	Linear Energy Transfer
MARS	Modular, Adaptive, Reconfigurable System
MCU	Microcontroller Unit
MEMS	Micro-Electro-Mechanical-System
MMS	Multi Mission Modular Spacecraft
MISO	Master Input Slave Output
MOSI	Master Output Slave Input
MR2	Modular, Reconfigurable, and Rapid-response
MSN	Mission Board
NASA	National Aeronautics and Space Administration
NiMH	Nickel Metal Hydride
NLAS	Nanosatellite Launch Adapter System
NRCSD	NanoRacks CubeSat Deployer
OBC	On-Board Computer
OPAL	Orbiting Pico-satellite Automated Launcher
PCB	Printed Circuit Board
PLD	Programmable Logic Device
PDU	Power Distributive Unit
P-POD	Poly Picosatellite Orbital Deployer
RAB	Rear Access Board
RSAT	Remote Sensing Advanced Technology
RTV-S	Room Temperature Vulcanizing-Silicon
SDRAM	Synchronous Dynamic Random Access Memory
SEB	Single Event Burnout
SEE	Single Event Effect

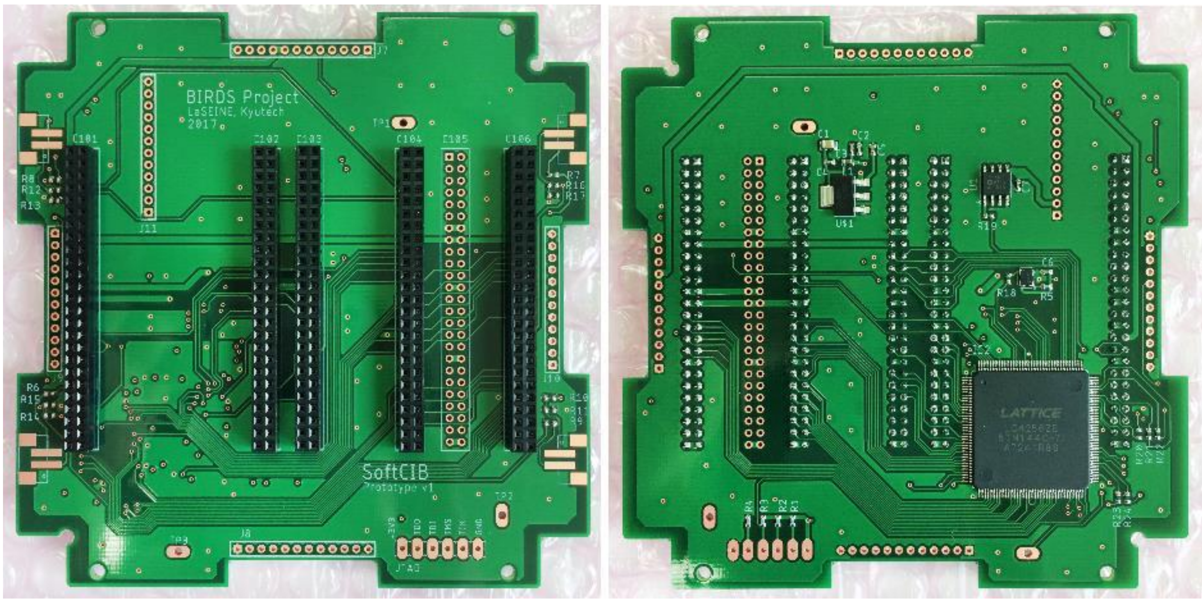
SEFI	Single Event Functional Interrupts
SEGR	Single Event Gate Rupture
SEL	Single Event Latch-up
SEU	Single Event Upset
SMEX	Small Explorer
SoftCIB	Software Configurable Bus Interface
SPI	Serial Peripheral Interface
SPL	Single Picosatellite Launcher
TID	Total Ionizing Dose
TOA	Time of Arrival
T-POD	Tokyo Picosatellite Orbital Deployer
TTL	Transistor-Transistor Logic
TVT	Thermal Vacuum Test
UART	Universal Asynchronous Receiver/ Transmitter
UHF	Ultra High Frequency
URAS	Upper Atmosphere Satellite
USB	Universal Serial Bus
VHF	Very High Frequency
X-POD	eXperimental Push Out Deployer
WIFI	Wireless Fidelity
QT	Qualification Test

List of Figures

List of Tables

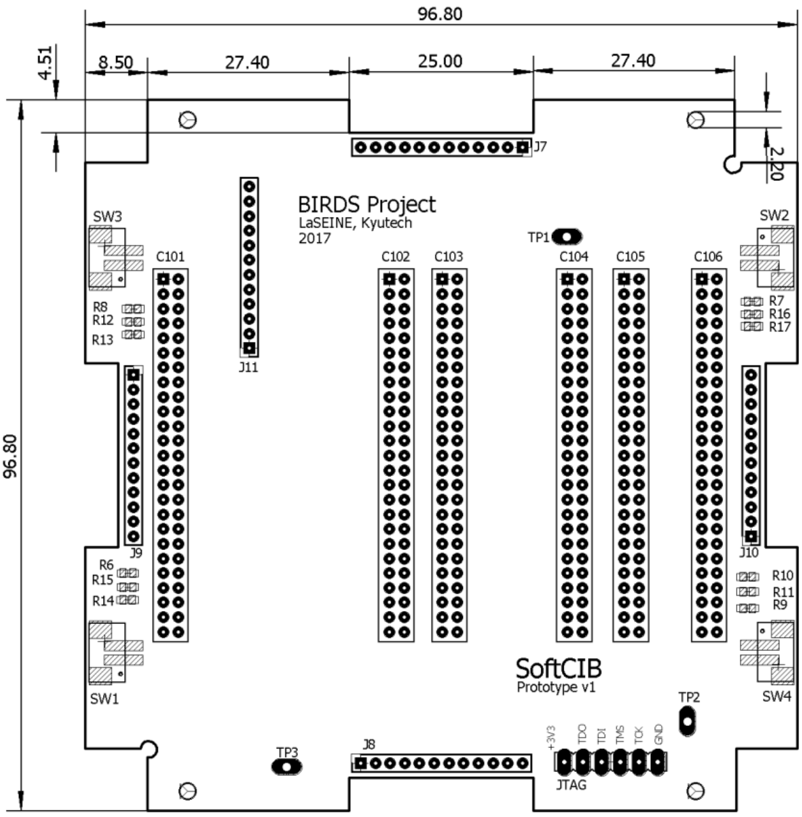
Appendix B:
Design specifications of SoftCIB

Physical Characteristics



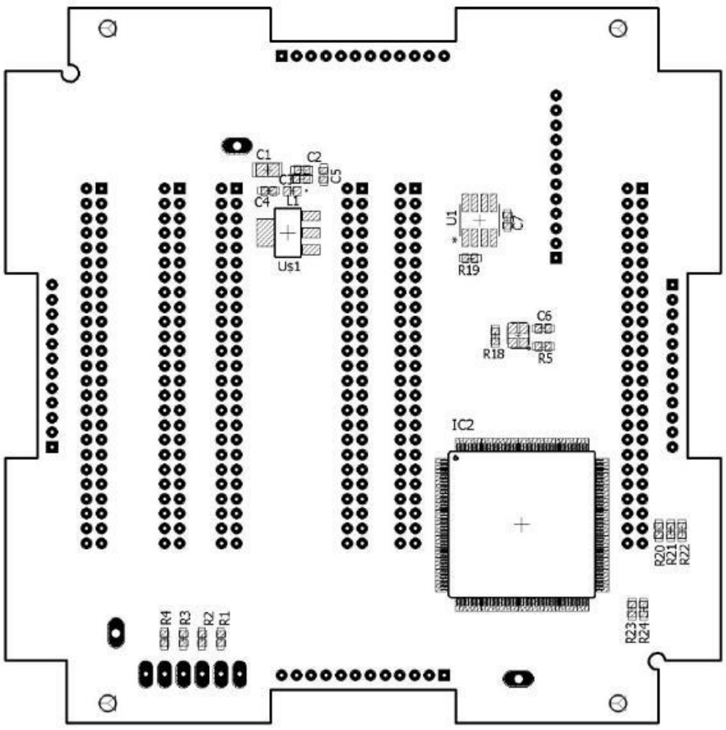
Picture of SoftCIB board (version No.1). Left: Front side, Right: Back side

Simplified Mechanical Layout[2]



[2] Dimensions in millimeters

Front side view



Back side view

Connector reference

No.	Connector name	Number of pin	Connector description	Purpose of connectors
1	C101	50	Front access board (FAB)	Subsystem boards to connect CubeSat bus
2	C102	50	Electric Power System / On Board Computer	
3	C103	50	Communication Board	
4	C104	50	Mission Payload Board	
5	C105	50	Mission Payload Board	
6	C106	50	Rear Access Board / Antenna Board	
7	SW1	2	Deployment switch connector	Deployment switches
8	SW2	2	Deployment switch connector	
9	SW3	2	Deployment switch connector	
10	SW4	2	Deployment switch connector	
11	J7	12	+X Solar panel connector	Solar Panels or External Panels
12	J8	12	-X Solar panel connector	
13	J9	12	+Y Solar panel connector	
14	J10	12	-Y Solar panel connector	
15	J11	12	-Z Solar panel connector	
16	JTAG	6	Debugger/programmer connector	Programming the CPLD

Signal descriptions

Subsystem board connectors (50 pin)

C101 (FAB)			
Signal name	PIN Number		Signal name
Prog_GIO_1	1	2	Prog_GIO_2
Prog_GIO_3	3	4	Prog_GIO_3
Prog_GIO_5	5	6	Prog_GIO_6
FAB_to_RAB GIO 1	7	8	FAB_to_RAB GIO 2
FAB_to_RAB GIO 3	9	10	FAB_to_RAB GIO 4
FAB_to_RAB GIO 5	11	12	FAB_to_RAB GIO 6
GND_SYS	13	14	GND_SYS
SUP_5V0	15	16	SUP_5V0
FAB_to_OBC GIO 1	17	18	FAB_to_OBC GIO 2
FAB_to_OBC GIO 3	19	20	FAB_to_OBC GIO 4
POWERSC_+X	21	22	TEMP_2 (+X panel)
SUP_UNREG_1	23	24	SUP_UNREG_1
SUP_3V3_2[3]	25	26	SUP_3V3_2
POWERSC_+Y	27	28	TEMP_1 (+Y panel)
RAW_POWER	29	30	RAW_POWER
POWERSC_-Y	31	32	TEMP_5 (-Y panel)
POWERSC_-Z	33	34	TEMP_3 (-Z panel)
SUP_UNREG_2	35	36	SUP_UNREG_2
POWERSC_-X	37	38	TEMP_4 (-X panel)
Kill SW	39	40	DEP_SW-1
DEP_SW-2	41	42	CPLD-PIN 1 / DEP_SW-3
CPLD-PIN 2 / DEP_SW-4	43	44	CPLD-PIN 3 / TEMP_6
CPLD-PIN 4	45	46	CPLD-PIN 5
CPLD-PIN 6	47	48	CPLD-PIN 7
SUP_3V3_1	49	50	SUP_3V3_1

C102 (EPS&OBC)			
Signal name	PIN Number		Signal name
Prog_GIO_1	1	2	Prog_GIO_2
Prog_GIO_3	3	4	Prog_GIO_3
Prog_GIO_5	5	6	Prog_GIO_6
OBC-COM 1	7	8	OBC-COM 2
FAB_to_RAB GIO 3	9	10	FAB_to_RAB GIO 4
FAB_to_RAB GIO 5	11	12	FAB_to_RAB GIO 6
GND_SYS	13	14	GND_SYS
SUP_5V0	15	16	SUP_5V0
FAB_to_OBC GIO 1	17	18	FAB_to_OBC GIO 2
FAB_to_OBC GIO 3	19	20	FAB_to_OBC GIO 4
CPLD-PIN 8	21	22	CPLD-PIN 9
SUP_UNREG_1	23	24	SUP_UNREG_1
SUP_3V3_2	25	26	SUP_3V3_2
CPLD-PIN 10	27	28	CPLD-PIN 11
RAW_POWER	29	30	RAW_POWER
CPLD-PIN 12	31	32	CPLD-PIN 13
CPLD-PIN 14	33	34	CPLD-PIN 15
SUP_UNREG_2	35	36	SUP_UNREG_2
CPLD-PIN 16	37	38	CPLD-PIN 17
Kill SW	39	40	DEP_SW-1
DEP_SW-2	41	42	CPLD-PIN 18
OBC-COM 3	43	44	OBC-COM 4
OBC-COM 5	45	46	OBC-COM 6
OBC-COM 7	47	48	OBC-COM 8
SUP_3V3_1	49	50	SUP_3V3_1

C103 (COM)			
Signal name	PIN Number		Signal name
COM_to_RAB GIO1	1	2	COM_to_RAB GIO2
COM_to_RAB GIO3	3	4	COM_to_RAB GIO4
COM_to_RAB GIO5	5	6	COM_to_RAB GIO6
OBC-COM 1	7	8	OBC-COM 2
FAB_to_RAB GIO 3	9	10	FAB_to_RAB GIO 4
FAB_to_RAB GIO 5	11	12	FAB_to_RAB GIO 6
GND_SYS	13	14	GND_SYS
SUP_5V0	15	16	SUP_5V0
CPLD-PIN 19	17	18	CPLD-PIN 20
CPLD-PIN 21	19	20	CPLD-PIN 22
CPLD-PIN 23	21	22	CPLD-PIN 24
SUP_UNREG_1	23	24	SUP_UNREG_1
SUP_3V3_2	25	26	SUP_3V3_2
CPLD-PIN 25	27	28	CPLD-PIN 26
CPLD-PIN 27	29	30	CPLD-PIN 28
CPLD-PIN 29	31	32	CPLD-PIN 30
CPLD-PIN 31	33	34	CPLD-PIN 32
SUP_UNREG_2	35	36	SUP_UNREG_2
CPLD-PIN 33	37	38	CPLD-PIN 34
CPLD-PIN 35	39	40	CPLD-PIN 36
CPLD-PIN 37	41	42	CPLD-PIN 38
OBC-COM 3	43	44	OBC-COM 4
OBC-COM 5	45	46	OBC-COM 6
OBC-COM 7	47	48	OBC-COM 8
SUP_3V3_1	49	50	SUP_3V3_1

[3] SUP_3V3_2 line provides 3.3V to the power of CPLD

Subsystem board connectors (50 pin) – (Continued)

C104 (MSN-1)			
Signal name	PIN Number		Signal name
COM_to_RAB GIO1	1	2	COM_to_RAB GIO2
COM_to_RAB GIO3	3	4	COM_to_RAB GIO4
COM_to_RAB GIO5	5	6	COM_to_RAB GIO6
FAB_to_RAB GIO 1	7	8	FAB_to_RAB GIO 2
FAB_to_RAB GIO 3	9	10	FAB_to_RAB GIO 4
FAB_to_RAB GIO 5	11	12	FAB_to_RAB GIO 6
GND_SYS	13	14	GND_SYS
SUP_5V0	15	16	SUP_5V0
CPLD-PIN 39	17	18	CPLD-PIN 40
CPLD-PIN 41	19	20	CPLD-PIN 42
CPLD-PIN 43	21	22	CPLD-PIN 44
SUP_UNREG_1	23	24	SUP_UNREG_1
SUP_3V3_2	25	26	SUP_3V3_2
CPLD-PIN 45	27	28	CPLD-PIN 46
CPLD-PIN 47	29	30	CPLD-PIN 48
CPLD-PIN 49	31	32	CPLD-PIN 50
CPLD-PIN 51	33	34	CPLD-PIN 52
SUP_UNREG_2	35	36	SUP_UNREG_2
CPLD-PIN 53	37	38	CPLD-PIN 54
CPLD-PIN 55	39	40	CPLD-PIN 56
CPLD-PIN 57	41	42	CPLD-PIN 58
CPLD-PIN 59	43	44	CPLD-PIN 60
CPLD-PIN 61	45	46	CPLD-PIN 62
CPLD-PIN 63	47	48	CPLD-PIN 64
SUP_3V3_1	49	50	SUP_3V3_1

C105 (MSN-2)			
Signal name	PIN Number		Signal name
COM_to_RAB GIO1	1	2	COM_to_RAB GIO2
COM_to_RAB GIO3	3	4	COM_to_RAB GIO4
COM_to_RAB GIO5	5	6	COM_to_RAB GIO6
FAB_to_RAB GIO 1	7	8	FAB_to_RAB GIO 2
FAB_to_RAB GIO 3	9	10	FAB_to_RAB GIO 4
FAB_to_RAB GIO 5	11	12	FAB_to_RAB GIO 6
GND_SYS	13	14	GND_SYS
SUP_5V0	15	16	SUP_5V0
CPLD-PIN 39	17	18	CPLD-PIN 40
CPLD-PIN 41	19	20	CPLD-PIN 42
CPLD-PIN 43	21	22	CPLD-PIN 44
SUP_UNREG_1	23	24	SUP_UNREG_1
SUP_3V3_2	25	26	SUP_3V3_2
CPLD-PIN 45	27	28	CPLD-PIN 46
CPLD-PIN 47	29	30	CPLD-PIN 48
CPLD-PIN 49	31	32	CPLD-PIN 50
CPLD-PIN 51	33	34	CPLD-PIN 52
SUP_UNREG_2	35	36	SUP_UNREG_2
CPLD-PIN 53	37	38	CPLD-PIN 54
CPLD-PIN 55	39	40	CPLD-PIN 56
CPLD-PIN 57	41	42	CPLD-PIN 58
CPLD-PIN 59	43	44	CPLD-PIN 60
CPLD-PIN 61	45	46	CPLD-PIN 62
CPLD-PIN 63	47	48	CPLD-PIN 64
SUP_3V3_1	49	50	SUP_3V3_1

C106 (RAB)			
Signal name	PIN Number		Signal name
COM_to_RAB GIO1	1	2	COM_to_RAB GIO2
COM_to_RAB GIO3	3	4	COM_to_RAB GIO4
COM_to_RAB GIO5	5	6	COM_to_RAB GIO6
FAB_to_RAB GIO 1	7	8	FAB_to_RAB GIO 2
FAB_to_RAB GIO 3	9	10	FAB_to_RAB GIO 4
FAB_to_RAB GIO 5	11	12	FAB_to_RAB GIO 6
GND_SYS	13	14	GND_SYS
SUP_5V0	15	16	SUP_5V0
CPLD-PIN 65	17	18	CPLD-PIN 66
CPLD-PIN 67	19	20	CPLD-PIN 68
CPLD-PIN 69	21	22	CPLD-PIN 70
SUP_UNREG_1	23	24	SUP_UNREG_1
SUP_3V3_2	25	26	SUP_3V3_2
CPLD-PIN 71	27	28	CPLD-PIN 72
CPLD-PIN 73	29	30	CPLD-PIN 74
CPLD-PIN 75	31	32	CPLD-PIN 76
CPLD-PIN 77	33	34	CPLD-PIN 78
SUP_UNREG_2	35	36	SUP_UNREG_2
CPLD-PIN 79	37	38	CPLD-PIN 80
CPLD-PIN 81	39	40	CPLD-PIN 82
CPLD-PIN 83	41	42	CPLD-PIN 84
CPLD-PIN 85	43	44	CPLD-PIN 86
CPLD-PIN 87	45	46	CPLD-PIN 88
CPLD-PIN 89	47	48	CPLD-PIN 90
SUP_3V3_1	49	50	SUP_3V3_1

Note:

Digital signals connected to CPLD
Signals lines that not connected to CPLD, permanent routing on the PCB
Power lines, permanent routing on the PCB

Solar panel connectors

J7	
PIN	**Signal Name**
1	GND
2	SUP_3V3_1
3	N/C
4	SP_BUS_1
5	SP_BUS_2
6	N/C
7	N/C
8	POWERSC_-X
9	POWERSC_-X
10	TEMP_2
11	N/C
12	GND

J8	
PIN	**Signal Name**
1	GND
2	SUP_3V3_1
3	N/C
4	SP_BUS_1
5	SP_BUS_2
6	CPLD-PIN 41
7	CPLD-PIN 42
8	POWERSC_-X
9	POWERSC_-X
10	TEMP_4
11	N/C
12	GND

J9	
PIN	**Signal Name**
1	GND
2	SUP_3V3_1
3	N/C
4	SP_BUS_1
5	SP_BUS_2
6	N/C
7	N/C
8	POWERSC_-X
9	POWERSC_-X
10	TEMP_1
11	N/C
12	GND

J10	
PIN	**Signal Name**
1	GND
2	SUP_3V3_1
3	N/C
4	SP_BUS_1
5	SP_BUS_2
6	N/C
7	N/C
8	POWERSC_-X
9	POWERSC_-X
10	TEMP_5
11	N/C
12	GND

J11	
PIN	**Signal Name**
1	GND
2	SUP_3V3_1
3	N/C
4	SP_BUS_1
5	SP_BUS_2
6	CPLD-PIN 39
7	CPLD-PIN 40
8	POWERSC_-X
9	POWERSC_-X
10	TEMP_6
11	N/C
12	GND

Note:

(yellow)	*Digital signals connected to CPLD*
(grey)	*Signals lines that not connected to CPLD, permanent routing on the PCB*
(pink)	*Power lines, permanent routing on the PCB*

CPLD programming/Debugging

Connector Name	JTAG					
PIN number	1	2	3	4	5	6
Signal Name	SUP_3V3_2	TDO	TDI	TMS	TCK	GND