www.adeepakpublishing.com

Kang, J. S. et al. (2023): JoSS, Vol. 12, No. 1, pp. 1187–1200
(Peer-reviewed article available at www.jossonline.com)

www. JoSSonline.com

Hardware Architecture for Evolutionary SmallSat Pedagogy and Space Workforce Development

Spencer Temkin and Jin S. Kang
Aerospace Engineering Dept., United States Naval Academy, Annapolis, MD US

John Moore
Institute for Earth Observations, Palmyra, NJ US

Maximillian Holliday
Dept. of Materials Science and Engineering, Stanford University, Stanford, CA US

Abstract

This paper presents an evolutionary pedagogy for training small satellite developers with an example hardware architecture that implements progressive learning with emphasis on inexpensive COTS hardware, culminating with work on actual flight articles. Small satellite development involves two parallel paths – hardware and software. Although learners may not have access to flight hardware during the early stages of their curriculum, it is a fairly simple matter to integrate a common software programming environment throughout. The proposed architecture uses a coding environment entirely in Python throughout the curriculum. In its MicroPython and CircuitPython implementations, it can be used as a microcontroller language, replacing C-based languages quite effectively. Hardware training still benefits from an incremental approach. The proposed architecture is a three-tiered approach to hardware training, gradually transitioning from working with a fully integrated spacecraft simulator to component fabrication and integration with a flight-ready vehicle. An example Tier 1 spacecraft simulator would be the A^{3}Sat device that is based on a Raspberry Pi processor and can easily be assembled by mid-to high-school students. A Tier 2 device is an Adafruit Metro M4 microcontroller board, which can run MicroPython or CircuitPython natively through its onboard ARM microcontroller. An ideal Tier 3 device is the PyCubed CubeSat open-source architecture, which is a flight-ready CubeSat avionics package in the $200 (USD) range. Learners would then finally have the opportunity to train on the same hardware they can use to implement their flight vehicle. At this tier, learners would focus on payload development, designing and building their own components while using the skills learned previously to integrate the hardware and software with the PyCubed bus. This paper will detail the hardware architecture at the three Tiers and propose how they can be integrated effectively into space system education pedagogy.

Corresponding Author: Jin S. Kang – kang@usna.edu

Publication History: Submitted – 03/10/22; Revision Accepted – 12/08/22; Published – 02/02/23

1. Introduction

It is a widely cited adage that the best way to provide training is to "train like you fight." In the case of educational pedagogy for small satellite developers, this would ideally entail using actual flight-rated hardware and operational techniques throughout a training curriculum. For several reasons, including cost and availability, this is not usually practical. This paper presents an evolutionary pedagogy for training small satellite developers with an example hardware architecture that implements progressive learning with emphasis on inexpensive COTS hardware, culminating with work on actual flight articles. Small satellite development involves two parallel paths—hardware and software. Although learners may not have access to flight hardware during the early stages of their curriculum, it is a fairly simple matter to integrate a common software programming environment throughout. Traditional methods often involve starting in a very high-level graphical coding language, like Scratch, and gradually transitioning through multiple languages to a low-level hardware optimized language, like C/C++. This often leads to distraction and obstacles to comprehension when learners are forced to focus more on relearning new coding syntax rather than focusing on data and control flow.

Rapid development of CubeSats over the past two decades (1999–present) has occurred, ranging from research to significant mission integration. The capabilities of CubeSats continue to expand and are being deployed in a wide range of sophisticated scientific and commercial missions, demonstrating that CubeSats have earned a legitimate place in the Aerospace Enterprise. Accordingly, leveraging the CubeSat platform for student education in satellite technology fits well with project-based learning initiatives. Involvement in CubeSat or related projects is beneficial for STEM education at all levels from middle school to university.

There are many examples of CubeSat programs being integrated into STEM (Science, Technology, Engineering, and Math) education, as it brings many benefits. In particular, CubeSat programs provide a great framework for introducing new concepts, hardware, and software to students in both classroom and laboratory settings. This is especially helpful for engineering courses. Space systems incorporate multiple topics, including RF (radio frequency) communications, power regulation and management, environmental control, digital signal processing, microprocessor control, A/D (analog-to-digital) and D/A (digital-to-analog) conversion, and remote sensing into a single package (Holman et al., 2014). In engineering classrooms, new students who are not yet familiar with complex hardware can be introduced to space system concepts using experiments and student projects involving sensors, remote actuators, and wireless communications (Holman et al., 2014). Another example is Massachusetts Institute of Technology (MIT)'s integration of a CubeSat program into their three-semester course that uses the development of a CubeSat-based science mission as its core teaching method (Smith et al., 2011). Designing and building a CubeSat served as their Capstone Design curriculum in the undergraduate program. "This project-based approach gives students essential first-hand insights into the challenges of balancing science requirements and engineering design" (Smith et al., 2011). The United States Naval Academy (USNA) also has implemented small satellite development as a centerpiece for Capstone Design curriculum in the Astronautics Track (within the Aerospace Engineering Department) since 2001. There have been 17 space payloads launched into space, and seven of them have been CubeSats (Kang et al., 2021). CubeSat development projects have been serving as the core of the Capstone Design curriculum's "Conceive - Design - Implement - Operate" (CDIO) concept, providing students with valuable hands-on engineering experiences (Kang et al., 2021; Gregory et al., 2020). The value of such programs is well recognized, and also serves as a main element in ABET criteria (Shiroma et al., 2003).

These education and training benefits of CubeSat programs extend to K-12 as well (Moore, 2013). More K-12 schools are adapting CubeSat or CubeSat-based modules into their education programs in promoting STEM interests among students. The Weiss School, which is a middle school, has an active CubeSat program, and has successfully launched CubeSats and other space payloads (Lyons et al., 2018). The Thomas Jefferson High School for Science and Technology is currently developing a 2U CubeSat that will perform communication linkage tests with the Iridium constellation, scheduled to launch in 2022 (David and Zaman, 2018). These are examples of many programs throughout the world that integrate elements of space flight hardware in educating and training the future space workforce. While CubeSat platforms have drastically reduced the bar for developing payloads for space, thus aiding in the development of the future space workforce, the bar may be still too high for many programs that are not established in space hardware and software development. Typical middle and high schools will have much difficulty in implementing CubeSat or CubeSat-like programs due to the complexity of the system, cost, and lack of expertise.

The United States Naval Academy's Small Satellite Program (NASSP) has been actively developing and launching space payloads since 2001, and has involved hundreds of undergraduate students in the design, development, integration, testing, and operation aspects of space systems engineering. The Naval Academy is an undergraduate-only institution, and thus has fostered a satellite engineering program centered around projects that are purely managed and run by students. NASSP and its space system projects have become more mature and streamlined over the years where the current Capstone Design operation is very close to achieving a "One CubeSat a Year" model. However, even such an established program as NASSP still has difficulty in educating students to the level where reliable space hardware can be easily developed. To overcome these obstacles, a tiered approach is proposed where the students working their way through the program will be given an opportunity to master different aspects of the key concepts.

To alleviate some of these challenges in integrating satellite development programs into education, a different pedagogy is proposed. The proposed architecture uses a coding environment entirely in Python throughout the curriculum. In its MicroPython and CircuitPython implementations, it can be used as a micro-controller language, replacing C-based languages quite effectively. Hardware training still benefits from an incremental approach. The proposed architecture is a three-tiered approach to hardware training, gradually transitioning from working with a fully integrated spacecraft simulator to component fabrication and integration with a flight ready vehicle. An example Tier 1 spacecraft simulator would be the A^{3}Sat device that is based on a Raspberry Pi processor and can easily be assembled by mid-to high-school students. Its physical layout and block systems diagram resembles a flight ready spacecraft and serves as a conceptual introduction. A Tier 2 device is an Adafruit Metro M4 microcontroller board, which can run MicroPython or Circuit-Python natively through its onboard ARM microcontroller. This resembles an on-board computer (OBC) that learners might use in a flight vehicle and allows them to connect and control multiple COTS components to perform subsystem level integration on their own, while still having a framework to work within. An ideal Tier 3 device is the PyCubed CubeSat open-source architecture, a flight-ready CubeSat avionics package in the $200 range. Learners would then finally have the opportunity to train on the same hardware they can use to implement their flight vehicle. PyCubed includes most of the required bus subsystems in a CubeSat format and incorporates the same processor as the Metro M4 board, allowing learners to directly migrate software developed on one to the other. At this tier, learners would focus on payload development, and designing and building their own components while using the skills learned previously to integrate the hardware and software with the PyCubed bus. This paper will detail the hardware architecture at the three Tiers and propose how they can be integrated effectively into space system education pedagogy.

2. Pedagogical Foundations

The goal of this pedagogical approach is to apply an evolutionary curriculum for small satellite and space system design with a consistent learning environment. An often-heard complaint from students is that concepts and techniques learned in one class are not applied in future courses and that they must often relearn the same techniques using different methods of application. An example of this is computer coding being taught in a learning language in one course and then applied in a different course which requires a different coding language. Although the basic concepts may be the same, time spent "getting back to speed" detracts from the amount of class time available for practice and mastery. An overarching goal of this architecture is to utilize a consistent hardware and software framework which builds on existing proficiency while adding new skills at each level to minimize the amount of "rework" required.

A notional undergraduate program using the proposed tiered architecture would take three to four years to complete, with the first year or prerequisite courses being required to impart basic engineering skills. Prerequisite skills may include basic computer programming, circuits and digital electronics, as well as fundamental lab skills. Alternatively, these skills could be taught concurrently with Tier 1 hardware lessons. It is envisioned that instructors would tailor lab activities to apply to specific goals and applications specific to their programmatic student outcomes to maximize synergy between lab work and class work. For example, a program that focuses on systems engineering may feature activities involving commercially purchased components emphasizing hardware integration factors, while a program with focus on electronics engineering may design its own components to interact with the existing hardware architecture in a prescribed way. Although not expressly designed as such, the evolution from Tier 1 to Tier 3 hardware roughly follows the classical composition of Bloom's taxonomy (Bloom, 1956). During the first year, students would cover engineering basics necessary to support future work with the hardware. This covers the base of "knowledge" and begins the "comprehension" level of the taxonomy.

Tier 1 learning is focused on making use of the hardware in a functional way, without requiring detailed design or integration. Most components may be of the "plug and play" variety; the value to learning is in the ability to begin to see how to use them together to make a whole and what is required at these top-level interfaces. At this level, Bloom's "comprehension" should be mature and "application" is emphasized.

Tier 2 learning begins to be focused on students making their own parts to fill a specific need or designing ways to integrate parts that are not specific to each other. This level of detail-oriented learning should impart deeper understanding of how the components interact and the considerations given to selecting a particular solution. At this level, strong "application" skills are needed, "analysis" is cultivated, and "synthesis" begins to come into play.

At the final tier, student learning is focused on representative or actual flight hardware to design and implement a space mission. External goals and motivations may now drive student learning to apply their prior skills and make top level design decisions. The entire continuum of Bloom's taxonomy comes into play, with an emphasis on "synthesis" and "evaluation." It may also be noted that this curriculum design adheres to the traditional "V" model of systems engineering, in that it begins with a very broad conceptual level of knowledge, progresses to a granular, detailed level in Tier 2 and then returns to a broader, mission focused scope in Tier 3. The relationships between the tiered architecture and other traditional pedagogical markers are shown in Table 1.

As in any project-based, hands-on learning program, having sufficient resources is critical to success. In the notional undergraduate program described above, the ideal setup is to have lab staff who are intimately familiar with the hardware and codes that are implemented on the hardware. This will ensure that

Table 1. Correlation Between Proposed Architecture and Other Pedagogical Markers

Architecture Tier	Learning Objectives	Bloom's Taxonomy	Learning Scope
Fundamentals	Basics and Prerequisites	Knowledge, Comprehension	Very Broad and Interdisciplinary
Tier 1	Components and Applications	Comprehension, Application	Broad and Interdisciplinary
Tier 2	Design and Fabrication, Subsystem Level Integration	Application, Analysis	Specific- and Space-Focused
Tier 3	Flight Specific Hardware, System and Vehicle Level Integration	Synthesis, Evaluation	Broad- and Space-Focused

consistent support is maintained throughout the class years and sections. The roles of faculty instructors will be to provide context to the lab activities, supplying the relevance to the lessons. This also makes it possible for the teachers to concentrate on the content delivery without needing to train on the hardware implementation and coding. In the case of the lab staff, a concentrated three-month period of assembling the hardware for the three Tiers, in addition to running through the lab activities themselves, have shown to be sufficient to prepare them for supervising student lab activities. In terms of lab resources, all three Tiers can be implemented with a typical engineering lab setup which includes soldering equipment, power supplies, multimeters, etc. Special equipment, such as clean room, thermal vacuum chamber, spectrum analyzer, etc., only becomes necessary if Tier 3 is to lead to space flight hardware.

3. Hardware Architecture Description

3.1. Tier 1-A^3 System

In collaboration with Burlington County Bridge Commission at Palmyra Cove Institute for Earth Observation, an environmental data kit for schools that comes in the shape of a CubeSat is being developed. This 1U CubeSat form-factor education unit is called A^3 Sat (Acquire – Analyze – Apply). It uses a Raspberry Pi 3b+ board as its main processor due to its advantages in power consumption, cost, and versatility. Pimoroni Enviro + board was selected as the main payload sensor board. This board can measure temperature, pressure, humidity, light, proximity, gas, and sound. Other GPIO (General Purpose Input/Output) pins are also available for the end user to implement additional payloads as needed. A^3 Sat also includes an IO imager and an infrared/thermal camera, to provide the students with understanding of different aspects of the imaging missions. A completed A^3 is shown in Figure 1. The criteria and constraints applied to developing A^3 is shown in Table 2. A^3 Sat comes with a detailed construction manual, satellite software, ground station software, and CAD (computer-aided design) files for 3D printing structure components.

A^3 Sat was designed and developed to both strengthen existing curricula taught in classrooms and to incorporate topics commonly missed in typical engineering classes. Incorporating a wide variety of fields simply in its construction, domains such as computer science, mechanical engineering, spatial structures, electrical engineering, and material science are embedded within it, allowing students to explore these fields and build vital technical skills. Extending outwards from these topics, the nature of satellites and

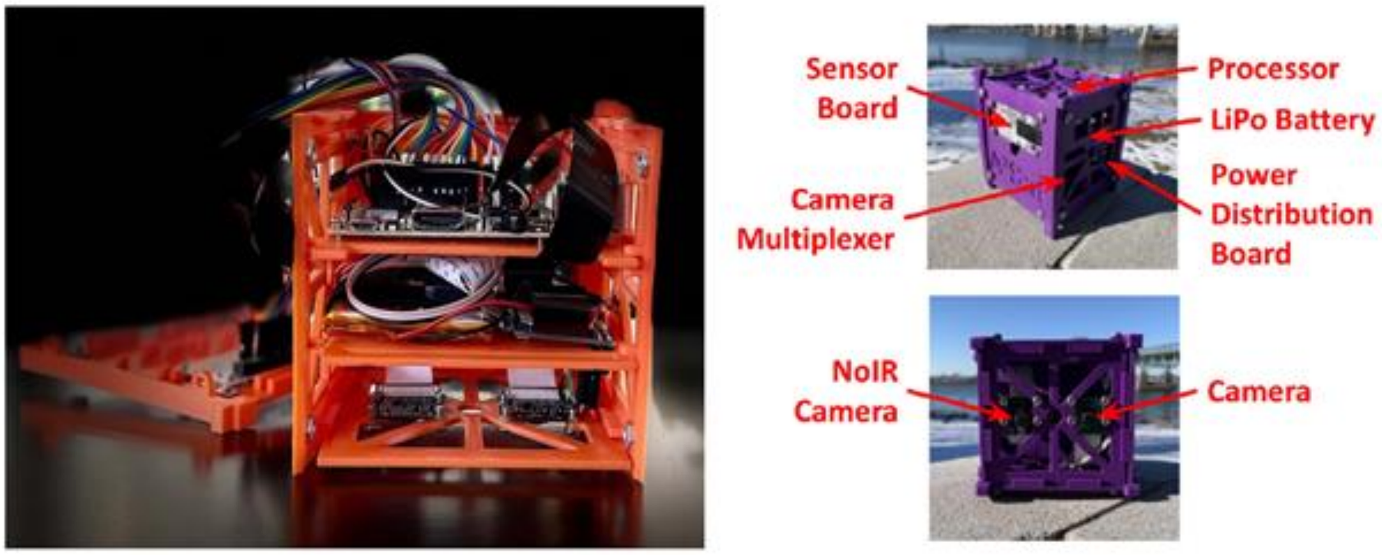
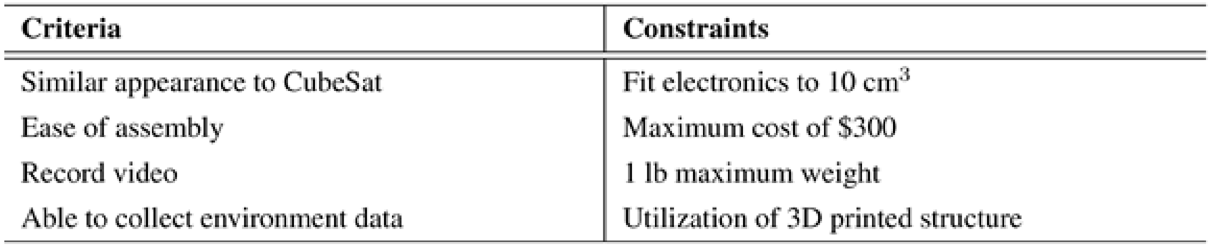


Figure 1. Images of A^3.

Table 2: Criteria and Constraints for A^3 Development



Criteria	Constraints
Similar appearance to CubeSat	Fit electronics to 10 cm^3
Ease of assembly	Maximum cost of $300
Record video	1 lb maximum weight
Able to collect environment data	Utilization of 3D printed structure

their close intricate ties to data is what will further their understanding of the overall space systems. In this model, the focus is more on the collection of data from environmental aspects and less on the flight hardware development itself. This allows students to further understand such topics as chemical compounds and concentrations, atmospheric phenomenon, geographical data sets, photons and wavelengths, and other physical science topics integral in both foundational and advanced knowledge of the scientific world. This also provides a context for what the impact of future space missions will be. As this data is collected, students gain the ability to map, plot, deeply analyze and interpret the data, catalyzing the process of scientific thinking and experimentation. The A^3 Sat then serves as a gateway for students to immerse themselves in STEM fields far out of reach, developing schools and minds alike with the processes and methodologies utilized by the world's leading scientists, and further establishing a foundation for the new generation to build upon.

The goal of the A^3 is to give pre-college students an opportunity to experience and experiment with data and environmental science. Students should have a tool they can use to acquire environmental data so they can analyze and apply it later. While these devices are not flight hardware, they have a similar appearance to a real CubeSat, and allow students to collect relevant data. The ultimate goal is to be able to provide schools with these kits so that they may be able to participate in studies and challenges that involve constructing them and collecting data.

3.2. Tier 2 -Onboard Computer Development

Metro M4 Express board running MicroPython as its programming language was chosen as the OBC (onboard computer) simulator in Tier 2. Then main goal of Tier 2 is to provide a bridge between Tier 1 hardware to the flight-ready hardware of Tier 3 while providing an opportunity to further understanding of the Command and Data Handling architecture. Arduino compatible boards are in common usage and are somewhat of a *de facto* standard in teaching environments, so much so that many students now have experience using them at the high school level. The Python programming environment can be difficult to implement on Arduino microcontrollers though, and the C/C++ based Arduino scripting language is substantially different from the object-oriented nature of Python so as to make it undesirable.

MicroPython offers an alternative to C/C++ based microcontroller programming. It is a microcontroller optimized software implementation of Python 3 written in C. One of the great advantages of using a Python-based system in a learning environment is that it is an interpreted language, rather than compiled. The MicroPython development environment features an interactive read-evaluate-print-loop (REPL) that will execute code without compiling, greatly accelerating the testing and development cycle for new learners. In small satellite, low power systems, the direct hardware control and efficiency of low-level languages like C cannot always be effectively substituted by Python. Since the underlying implementation of MicroPython is in C, the environment is also extensible with C/C++ code for applications where direct low-level hardware interaction and control is desirable. In this way, efficient code execution can still be achieved while the beginner coders can learn quickly. While this is not the most effective way for preparing the future workforce where C is still widely used in the industry, the decision was made to opt for ease of implementation. The goal as an undergraduate program in this case was not to make students expert in an assumed skill set, but to put them in the position to "learn to learn" so that they can move and self-actualize as needed.

CircuitPython is an open-source fork of MicroPython developed and maintained by Adafruit Industries. It is specifically targeted at students, and runs on a variety of microcontroller boards. It has the additional advantages of being the native environment designed into the PyCubed board, and the capability of running on Raspberry Pi processors, such as those used in the A^3 platform. Students who have previous experience using Python-based programming on microcontrollers are most likely to have encountered it by using CircuitPython.

The Adafruit Metro M4 Express microcontroller board was selected as an ideal Tier 2 device. As a current Adafruit product, it is optimized to support CircuitPython, but can also run Arduino IDE. It features an Atmel/Microchip Technology ATSAMD51J19 120MHz ARM Cortex M4 microprocessor, nearly identical to the ATSAMD51J20 included in the PyCubed satellite board (differing only in onboard flash memory size). This should allow software that is developed on the Metro card to be directly ported to PyCubed applications. Thus, it can serve as both a learning tool and a development tool for PyCubed projects. The Metro M4 also is designed with an "Arduino compatible" layout with an identical footprint and similar pin diagram. All Arduino compatible components should also be compatible with the Metro board, including shields and hats. It also features hardware support for SPI, UART and I2C, allowing straightforward integration with most peripherals. The Metro M4 board layout is shown in Figure 2.

Like other Arduino style boards, the Metro M4 supports 3.3V logic and 5V power and can be powered through the built in Micro USB port or the onboard 2.1mm DC jack. It has pulse width modulation (PWM) outputs on 22 of its 25 GPIO pins, analog inputs through eight pins via a 1 MSPS A/D converter and true analog out through two pins via D/A converter. Another interesting feature that has potential use for small satellite development is built in support for crypto engines including AES (256 bit) encryption. A very useful feature for troubleshooting in the learning environment is a surface mounted RGB NeoPixel LED, which can be programmed to give visual feedback of board operations. The Adafruit Metro M4 board is readily available and costs less than $30 per device.

The selection of an OBC development board was driven by the desire to use a microcontroller board, which allows an easy transition from the A^3 platform (Tier 1) and maintains as much commonality with the PyCubed satellite board (Tier 3) as possible, while possessing a flexible and accessible form factor. While it may be possible to directly transition to the PyCubed board from Tier 1 hardware, a complete transition to a fully open-ended project environment may not be ideal for student learning. The PyCubed board includes most of the subsystems already integrated on a single board. This may make it difficult for the students to understand how the subsystems interact together when all they see is a single electronics board. It is also difficult to characterize individual components on PyCubed as components are mixed together as randomly dispersed, small surface-mount components. A modular board such as the M4 Express board can be a more effective tool in teaching students how individual components are integrated together. Desired subsystems or sensors can be added to the board as a separate module, enabling testing and characterizing at the component level. This setup also provides a good

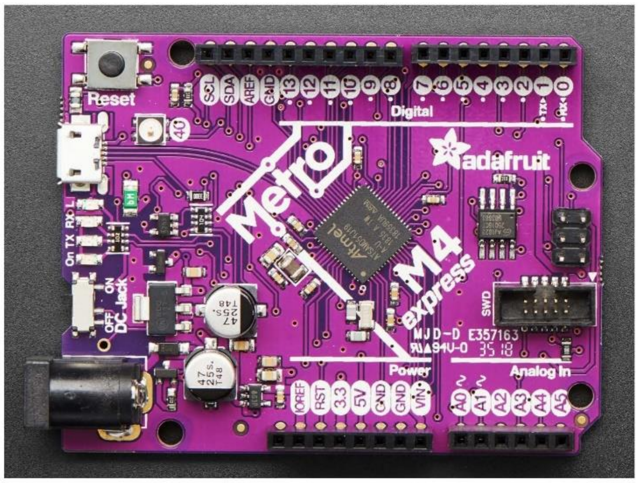


Figure 2: Picture of Adafruit Metro M4 express board (from Adafruit, at https://learn.adafruit.com/assets/85521).

opportunity for the students to further their understanding of command and data handling architecture of a satellite. After learning about how to handle peripherals and other subsystems/components in Tier 2, the students are better prepared to tackle an integrated package/system such as the PyCubed board.

3.3. Tier 3 -CubeSat Flight Hardware Training

PyCubed is single-board solution to CubeSat avionics developed by Max Holliday and his team at Stanford University (Holliday et al., 2019). It is a complete hardware and software combination stack for CubeSats that addresses many common pitfalls of small satellite building. A single board includes most of the core bus electronics, including the onboard computer, communication radio, electrical power management system, sensors, and connections for additional payloads. With an addition of structures, solar panels, and batteries, it can be made into a complete, flight-model CubeSat bus. Figure 3 shows top and bottom of the PyCubed board layout (PyCubed.org, 2022).

Another advantage of the board is that the software is based on Python, making it much easier for students not familiar with software to be able to more easily start on satellite development. PyCubed is designed to use CircuitPython, which allows Python language to be implemented on microcontrollers, resulting in drastic reduction in power consumption by the main processor. The CircuitPython architecture is an open-source effort led by Adafruit and targeted towards beginner programmers. Although it is easy to get started, CircuitPython is capable of enough depth to allow command and control of the spacecraft. The PyCubed mainboard does not have a traditional attitude determination and control (ADCS) unit. However, PyCubed has an excellent inertial measurement unit (IMU), and if a traditional ADCS is needed it can be added to to the spacecraft and interfaced to PyCubed using the abundant payload and GPIO connectors. The mainboard also does not incorporate a standard PC/104 bus connector, but has many breakout pins and connectors that can be leveraged to easily integrate with other hardware (PyCubed.org, 2022).

PyCubed has many great advantages as a core of a flight-model CubeSat. It implements high-reliability design practices, low-cost components that are radiation tolerant, and have good documentation. A PyCubed board costs approximately $250, as compared to typical COTS (commercial off-the-shelf) setups for a CubeSat that cost approximately $30,000 for similar functions. While the components are not radiation hardened, each component on the board was chosen based on the radiation tolerance data, making it a more radiation tolerant COTS solution. Numerous versions of these boards have also flown in space, and have shown good results (Holliday et al., 2019). Four recent spacecraft from NASA Ames and Stanford University have successfully used PyCubed for their flight avionics, including the 3U ”KickSat-2” mission in

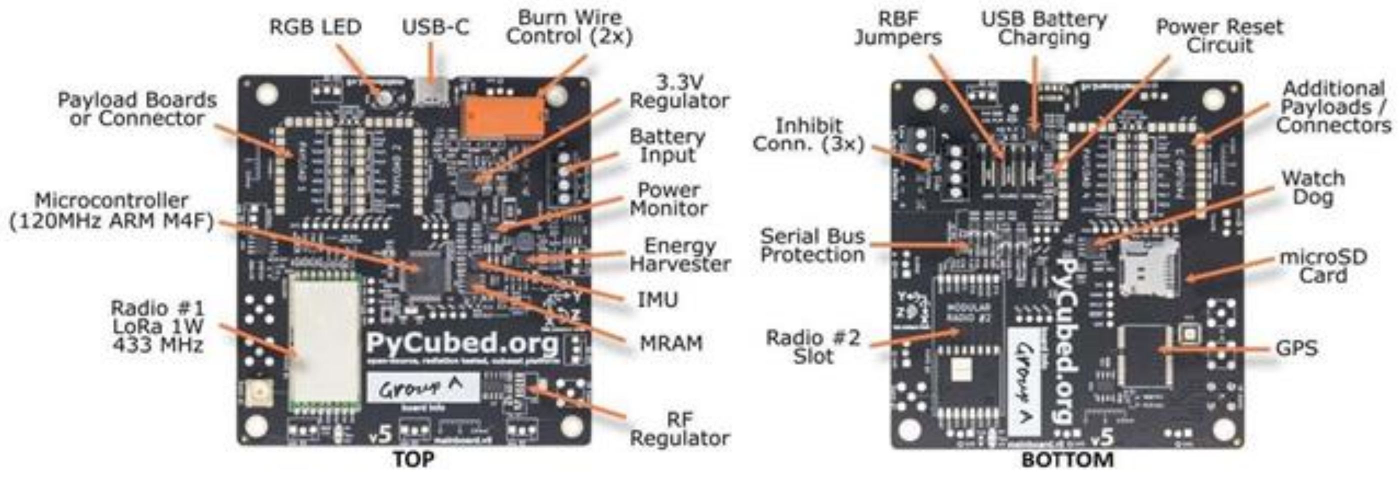


Figure 3. Images of PyCubed board showing key features.

2019 (Holliday et al., 2019) and three 1U CubeSats for the ”V-R3x” mission in 2021. University of Hawaii at Manoa’s Ke Ao 1U CubeSat is another good example of an implementation of PyCubed in CubeSat projects (Young et al., 2021). Many university courses and student-led teams are using PyCubed for CubeSat development in 2021.

4. Implementation

4.1. Current Curriculum Setup and Challenges

Due to the unique demands of the USNA undergraduate program, students do not declare majors until the end of their freshman year, and do not normally begin taking major-specific courses until the beginning of their sophomore year. As such, the astronautics course of study is limited to three years of specialized coursework spread over sophomore to senior years. The astronautics course of study itself is a specialized subset or “Track” within the aerospace engineering degree program. After a student has declared an intention to pursue a degree in aeronautical engineering, they then take two courses during sophomore year which introduce them to the fundamentals of both tracks, along with other engineering fundamentals courses. At the end of sophomore year students then choose to specialize in either the aeronautical track and continue to study atmospheric craft, or astronautics and study exo-atmospheric craft, with further courses tailored to each course of study. In either case, students are awarded an ABET accredited Bachelor of Science degree in aerospace engineering upon successful completion of their chosen degree track. Table 3 gives a summary of the relevant courses for the Astronautical Engineering Track at the Naval Academy.

Although there are certain advantages to delaying the declaration of a major until some coursework has already been completed, this system generally precludes the ability of students to begin specialized classes until after freshman year. The engineering college at USNA, known as the School of Engineering and Weapons, is only one of the three schools within the “university,” but the US Navy puts a high emphasis on engineering skills and all students are expected to complete core courses in chemistry, physics, calculus, differential equations, thermodynamics, and electrical engineering, along with other core classes. Most of these will have been completed before students begin their astronautics course of study after sophomore year. All aerospace students, both aeronautical and astronautical tracks, will additionally complete courses in introductory aeronautics and astronautics, statics, dynamics, materials, structures, computational methods and gas dynamics by the end of their sophomore year before specializing in their given track. Astronautical track students will then go on to study astrodynamics, attitude dynamics and control, power and communications, rocket propulsion, space weather, and associated labs during their junior and senior years. All students are also required to complete a spacecraft vehicle Capstone Design design project during their senior year.

NASSP Capstone Design projects typically involve the design, development, and construction of a CubeSat spacecraft. A nominal design project includes all aspects of satellite project management, from concept to launch and operations over the course of senior

Table 3: Astronautical Engineering Studies at United States Naval Academy

Class Year	Relevant Courses
1	Chemistry, Calculus I & II
2	Physics, Calculus III, Differential Equations, Intro to Aeronautics, Intro to Astronautics, Statics, Dynamics, Materials, Computational Methods
3	Electrical Engineering, Structures, Thermodynamics, Astrodynamics, Gas Dynamics, Communications & Power, Attitude Dynamics & Control
4	Rocket Propulsion, Space Weather, Space Systems Engineering Lab, Spacecraft Vehicle Design, at least two additional major electives

year. This ambitious operational schedule benefits from a standardized architecture like that proposed here. The compressed nature of majors studies at USNA requires the type of integrated curriculum approach exemplified by the proposed tiered architecture to achieve maximum success, but other more traditional curricula will also benefit. Additionally, due to the engineering emphasis at USNA, all non-engineering major students must take a survey-style course in an engineering field. The type of introductory level engineering taught in these courses would also be a natural fit for a Tier 1-type approach.

4.2. Curriculum Implementation Plan

The compressed nature of the astronautical major curriculum at USNA does not at first seem an ideal fit for the proposed pedagogical ideal of the three to four-year course of study built into the tiered approach. Operating within these constraints leaves only limited opportunities to maximize use of the Tier 1 hardware in introductory level courses. Tier 2 would be focused on during the second year, while Tier 3 would be relied on for the third and fourth years. With some minor modifications to the curriculum, a more level reliance on the three hardware tiers could be achieved and the overall program enhanced. This approach would also more closely align with other schools that provide a full four years of study within the major and might serve as a model for those type of programs wishing to adopt it. This concept is depicted in Figure 4.

The first step is to maximize opportunities to use the Tier 1 hardware in early coursework and lay the foundations for the tiered architecture and the more specialized hardware to follow. Because the A^3 system is “full up” and requires no further development from students to be effective, it presents many possibilities for incorporation into early coursework. These applications need not be space related or even engineering in nature. Any STEM oriented course which desires to include an applied laboratory activity may make use of the A^3 hardware as a platform. For example, one of the proposed applications which the A^3 system was expressly designed for was to take environmental data readings. It is easy to imagine ways in which this functionality might be used, even outside of an explicitly

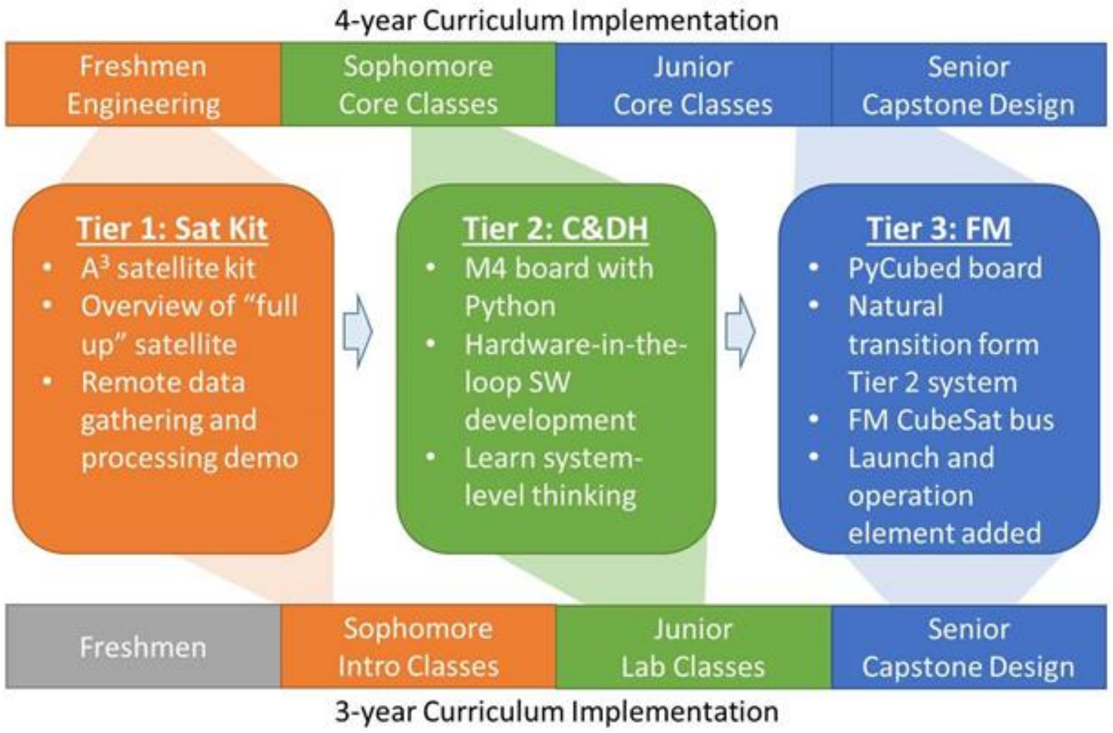

4-year Curriculum Implementation
Freshmen Engineering
Sophomore Core Classes
Junior Core Classes
Senior Capstone Design
Tier 1: Sat Kit
A³ satellite kit
Overview of “full up” satellite
Remote data gathering and processing demo
Tier 2: C&DH
M4 board with Python
Hardware-in-the-loop SW development
Learn system-level thinking
Tier 3: FM
PyCubed board
Natural transition form Tier 2 system
FM CubeSat bus
Launch and operation element added
Freshmen
Sophomore Intro Classes
Junior Lab Classes
Senior Capstone Design
3-year Curriculum Implementation

Figure 4. Depiction of how the Tiered Curriculum would fit in three-year and four-year engineering programs.

engineering context. By fostering familiarity with the equipment and focusing on operations and applications, even courses that are only peripherally related to the major course of study can support the follow-on implementation of the tiered architecture. A sound basis in programming and scripting controls for the A^3 is an added benefit of using the hardware as much as possible, and these skills are beneficial to all courses of study in a modern environment.

Early engineering fundamentals and majors courses can then make use of the Tier 2 hardware. Initial emphasis may be on more detailed programming techniques for direct control of hardware and systems integration. Specific applications may then include incorporating other hardware components and small-scale design projects. This type of hardware makes an ideal platform for the kind of lab intensive courses which tend to make up the middle part of such curricula. With a wide range of Arduino compatible components available and many example projects illustrated online, an application for the Tier 2 hardware can be found that is appropriate to most courses.

Since the Tier 3 hardware is actually a flight-ready system, any course that focuses on real-world spacecraft development can greatly benefit from its implementation. Once students have reached a suitable level of proficiency on the Tier 2 system, they should have no difficulty naturally transitioning to Tier 3, since the programming environment and hardware techniques are identical. Ideally, a Tier 3 supported course will have no need to re-teach any of these skills and will instead be able to focus additional time on the top-level, higher cognitive skills required in Capstone design level engineering. If the launch schedule is conducive to it, students should be able to learn on, practice on and then actually launch and operate the same piece of hardware in space. A condensed schedule where students accomplish all of this in only one year is greatly simplified by the tiered technology approach where broad, operational and mission-oriented skills from Tier 1 are combined with specific and specialized hardware skills from Tier 2 to successfully implement a space-worthy system using Tier 3.

4.3. Implementation at USNA

At USNA, an optimized approach would begin by incorporating the Tier 1 hardware into as many early classes as possible. At a minimum, this would include an introductory engineering fundamentals course taken during freshman year, which is currently being offered in a pilot program. Other classes scheduled during freshman year may adopt the technology for their own ends to further familiarity with the system. The non-major "Introduction to Aeronautics" course offered to students from the schools outside engineering should also find the Tier 1 technology a natural fit. Use of the Tier 1 system may be extended into the beginning of the sophomore year courses to provide continuity, with a transition to Tier 2 when appropriate. Classes that provide a natural transition point are during the Intro to Aeronautics and Intro to Astronautics courses. This course has lab components that further explore key course contents such as attitude control systems, communication systems, and power systems. They were previously performed as individual labs and unrelated hardware setups. With the implementation of A^3, the students will be seeing a consistent platform that resembles an actual CubeSat, and will have the opportunity to understand how different components interact with each other, while still achieving the learning objectives of how individual subsystems function. Other applications may include things like using a Tier 2 system to monitor accelerometers during a Dynamics class lab, or to collect data from strain gauges during a Materials class lab.

Tier 2 hardware may continue to be used in junior year courses, like Electrical Engineering, where it would make an excellent platform for circuit experimentation. Again, beginning with continuity from the previous year's equipment should help with students making the adjustment from one year to the next while reinforcing the previous lessons before moving on to a new setup. Tier 3 hardware may be introduced and used to good effect in the Power and Communications class and the Attitude Dynamics and Control class. The PyCubed board is a natural fit for these. With its built-in radio and power management system to experiment with power and communications, and the onboard IMU and magnetometer for attitude determination, existing learning objectives for these classes can be easily adapted to lab activities that not only reinforce classroom lessons, but also prepare students directly for the skills they will need to apply in further spacecraft design. One example is a combined lab where students will be experimenting with the Metro M4 Express board that is programmed (by students) to perform sun-pointing using solar panels, a battery pack, and a reaction wheel. The satellite, hung from the ceiling using a fishing wire, uses the reaction wheel to orient itself towards the light source. Adding and integrating external radios, solar panels, batteries, other sensors or attitude control devices will further expand the range of options available to support these, and follow-on classes.

During senior year, it is expected that most students will use the same Tier 3 hardware to provide the main bus for their spacecraft designs. Tier 2 boards may also be used to troubleshoot and develop integration techniques for payloads and additional components prior to installing on the spacecraft. At this level, all the prior knowledge students have gained while working with the integrated tiered technology architecture will come into play and they will realize the full benefits of the approach. The primary aim is that students at this phase will no longer have to spend flight a good portion of their time learning and developing their lab techniques and procedures to troubleshoot their hardware and will instead spend that time more productively in the actual troubleshooting process. Tier 2 and 3 systems may be used throughout the Space Systems Engineering Lab course for targeted classroom activities allowing instructors to maintain some control over the learning flow while also allowing these lessons to be directly applied to the Spacecraft Vehicle Design projects, no matter the real-world applications being addressed, since the hardware implementation is the same. Using such a system, it would be very difficult for an instructor to develop a lab activity which did not directly support the final Capstone Design project, thus realizing the ultimate goal of having all coursework tied to applications which support real-world ends and allowing students to cement their knowledge and realize the true value of all their classwork. The main satellite lab course consists of experiments with power, communication, command and data handling, and attitude control subsystems. Instead of separate setups designed for each lab activity, the student groups will be given PyCubed boards that they will characterize throughout the semester. Starting with C&DH labs, the students will modify the skeleton flight software to meet their design goals, leveraging the lessons learned in Tier 2 labs. The solar panels and the battery pack will be added and their performances will be characterized next. A communication system will also be implemented where the teams experiment with different protocols, modulations, and antenna designs. Fully characterizing the link performance is a key aspect of the lab. Integrating the systems together, ADCS will be integrated and the satellite overall performance will be demonstrated as a form of the lab course's final project. By the end of the first semester in their senior year, the students will have a fully developed, integrated, and tested CubeSat bus that they will integrate with the satellite payload that is being developed by the student teams as a part of their year-long Capstone Design curriculum. This allows students a complete understanding of the satellite bus system, as well as being able to concentrate on the payload development for their senior design project.

5. Conclusion

The capabilities of CubeSats continue to expand, and their usefulness as STEM education tools at all levels of education has been well proven. In a pedagogy for small satellite developers, an ideal program will entail use of actual flight rated hardware. For several reasons such as cost and availability, however, this may not always be feasible or practical. To overcome some of these challenges, a tiered curriculum approach is proposed, where the coding environment will be kept constant throughout the curriculum while the hardware provided to the students will evolve into a full flight-model by the end of the curriculum.

The proposed architecture uses a coding environment entirely in Python throughout the curriculum. In its MicroPython and CircuitPython implementations it can be used as a micro-controller language, directly replacing C-based languages like Arduino. Hardware training still benefits from an incremental approach where a three-tiered approach of gradually transitioning from working with a fully integrated spacecraft simulator to component fabrication and integration with a flight ready vehicle is adopted. This pedagogical approach allows students to maintain continuity and grow their development skills in software, and also slowly build up to the actual flight hardware that they would use to build a flight CubeSat in their senior year. Each Tier is geared towards delivering the education and training elements at each step while using the right tools, such that the cost can be minimized and the lessons made much more approachable. Students working their way through the proposed curriculum will not have difficulty transitioning to the final flight model development since the programming environment and hardware techniques are identical. The students will be able to learn on, practice on and then actually launch and operate the same piece of hardware in space, while keeping the bar-to-entry low for universities and even mid-and high-schools for project-based, STEM education.

References

Bloom, B. S. (1984): Taxonomy of Educational Objectives: The Classification of Educational Goals, Handbook 1: Cognitive Domain (2nd ed.). New York, NY, Addison-Wesley Longman Ltd. ISBN-10: 0582280109.

David, L. and Zaman, A. (2018): Simulating Iridium Satellite Coverage for CubeSats in Low Earth Orbit, presented at the 32nd Ann. AIAA/USU Conf. on Small Satellites, Logan, UT, Aug. 8. Paper SSC18-PI-06. Available at: https://digitalcommons.usu.edu (accessed Nov. 5, 2022).

Gregory, J. M. et al. (2020): Applying a Model-Based Approach to Develop a Standardized Template for CubeSat-Class Satellites, in *Proc. 2020 IEEE Aerospace Conf.*, Big Sky, MT, pp. 1-11. doi: 10.1109/AERO47225.20209172748.

Holliday, M. et al. (2019): PyCubed: An Open-Source, Radiation-Tested CubeSat Platform Programmable Entirely in Python, presented at the 33rd Ann. AIAA/USU Conf. on Small Satellites, Logan, UT, Aug. 8. Paper SSC19-WKIII-04. Available at: https://digitalcommons.usu.edu (accessed Nov. 11, 2022).

Holman, W. T. et al. (2014): The Small Satellite (CUBESAT) Program as a Pedagogical Framework for the Undergraduate EE Curriculum, in *Proc. 2014 ASEE Ann. Conf. & Expo., Indianapolis, IN*, pp. 24.1245.1–24.1245.12. doi: 10.18260/1-2-23178.

Kang, J. et al. (2021): Creating Future Space Technology Workforce Utilizing CubeSat Platforms: Challenges, Good Practices, and Lessons Learned, presented at *AIAA SciTech 2021 Forum* (Virtual Event). Paper AIAA 2021-1437. doi: 10.2514/6.2021-1437.

Lyons, R. et al. (2018): WEISS-SAT1: A Student Developed Astrobiology Payload for Small Satellite Microgravity Research, presented at the 32nd Ann. AIAA/USU Conf. on Small Satellites, Logan, UT, Aug. 8. Paper SSC18-WKIX-07. Available at: https://digitalcommons.usu.edu (accessed Mar. 5, 2022).

Moore, J. D. (2013): Integrating Small Satellites into the United States' K-12 STEM Education Discussion, *J. of Small Satellites*, Vol. 2(2), pp. 201–211. Available at: www.jossonline.com (accessed Oct. 1, 2021).

PyCubed.org (2022): PyCubed: An Open-Source, Radiation-Tested SmallSat Framework Programmable Entirely in Python. Available at: https://pycubed.org (accessed: Nov. 11, 2022).

Shiroma, W., Ohta, A., and Tamamoto, M. (2003): The University of Hawaii Cubesat: a Multidisciplinary Undergraduate Engineering Project, in Proc. 33rd Ann. Frontiers in Education, Westminster, CO, pp. S3A 7–S3A 11. doi: 10.1109/FIE.2003.1265973.

Smith, M. W., Miller, D. W., and Seager, S. (2011): Enhancing Undergraduate Education in Aerospace Engineering and Planetary Sciences at MIT Through the Development of a CubeSat Mission, in *Proc. SPIE Optical Engineering + Applications*,

p. 8146OS. San Diego, CA. doi: 10.11117/12.896130.

Young, L. et al. (2021): Ke Ao: A Low-Cost 1U CubeSat for Aerospace Education and Research in Hawaii, presented at the 35th Ann. AIAA/USU Conf. on Small Satellites, Logan, UT, Aug. 8. Paper SSC21-P2-44. Available at: https://digitalcommons.usu.edu (accessed Mar. 5, 2022).