

CubeFlow: Training for a New Space Community

Craig J. Kief¹

**Configurable Space Microsystems Innovations and Applications Center (COSMIAC)
Albuquerque, NM 87106**

**Jacob Christensen² and Bryan Hansen³
Space Dynamics Lab (SDL)
Logan, UT 84341**

**Jesse Mee⁴
Air Force Research Laboratory
Space Vehicles Directorate
Kirtland Air Force Base, NM 87117**

ABSTRACT

Many organizations (academia, industry and government) make high quality satellite components; however, very few organizations make entire satellites well. Those that can successfully create entire satellites, often take years to design and deploy “Swiss watch,” one-of-a-kind satellites. The federal government wants a way to capitalize on all of these organization’s quality components in a quick and efficient manner. To be more responsive to the military and emergency responder’s needs, rapid satellite development and deployment is critical. There is a need for a method to go from pushbutton mission design to off the shelf components (that all seamlessly integrate) in a rapid fashion. Under sponsorship by the Operationally Responsive Space (ORS) office, the Air Force Research Laboratory (AFRL) developed a modular, nanosatellite, plug-and-play (PnP) approach where hardware and software modules can be rapidly merged to form functional satellites. The Stanford/Cal Poly CubeSat and Poly-Picosatellite Orbital Dispenser (PPOD) standards have revolutionized the way that small satellites are developed and deployed. AFRL wants to capitalize on this momentum to advance the concepts and goals of rapid space. Small satellites are an excellent test bed for larger spacecraft. The combination of the AFRL’s PnP design paradigm and the CubeSat standards has resulted in the creation of a CubeFlow program and CubeFlow training. The basis of the electrical and software infrastructure is the AFRL Space PnP Avionics (SPA) technology. Many have complained about the complexity of developing components that conform to the SPA standards. To alleviate this, a secure, web-based, design system has been created that allows convenient access for developing design configurations and coordinating the offerings of a community of component developers. This system provides a simple development flow through which component manufacturers can easily and efficiently create a PnP module. This stems from the idea that minimizing the amount of code that a developer must produce and also minimizing human error through constant validation will greatly increase efficiency. The hope is that those trained in the CubeFlow courses will gain the skills needed to produce useful PnP components and allow the PnP community to expand. Currently, there are a number of organizations and universities that have expressed interest in nano-satellite programs and rapid space development. CubeFlow is intended to address the issue that, due to lack of funding or capability, it is rare that a single organization or university would be able to research and develop all the necessary components for a small satellite. If a community can be built around an accepted standard such as PnP, then it may be possible to coordinate efforts in such a way that no longer would a single entity be tasked with the development of an entire satellite - but rather a single module or component. It is believed that this will not only lead to faster development, but higher quality satellites.

¹ Deputy Director, COSMIAC, 2350 Alamo Avenue SE, Ste. 100, Albuquerque, NM 87106 AIAA Senior Member

² PhD Student, Computer Science Department, 4205 Old Main Hill, Logan, UT 84341, AIAA Student Member

³ Software Engineer, Engineering Division, 1695 N Research Park Way, North Logan, UT 84341

⁴ Electrical Engineer, RVSE, Kirtland AFB, NM 87117

BACKGROUND

Traditional satellite development is both costly and generally has a timeframe measured in years. The Air Force Research Laboratory's solution to drastically reduce this development time is to establish a plug-and-play standard. The proposed standard is known as Space PnP Avionics (SPA). This standard allows for self-discovery and self-configuration of both hardware components and software applications within a satellite network. The problem that plagues any new standard is always adoption. The added complexity that comes with creating truly plug-and-play components adds substantial complexity to the development process. This added initial complexity makes it even more difficult to convince others to adopt the SPA standard. AFRL's dream of having independently developed PnP components that can be pulled off a shelf and be rapidly integrated into a satellite will remain an impossibility until a sufficient number of organizations are trained and willing to support the standard.

While the SPA standards have been available for several years [1, 2, 3], what was missing was a structured way to train individuals and organizations on how to implement or utilize the standard. A training course had to be created from the ground up to advance these concepts. To keep costs low and allow for a true hands-on training experience, it was decided that the CubeSat standard would be used for the training. This small form factor satellite is a newer development in the satellite community that is revolutionizing the way small satellites are created and deployed. To increase the effectiveness of the course, an online web portal was created which would allow attendees to utilize a set of development tools to greatly simplify the creation of PnP satellite components during the course.

The main objective of the course was for attendees to acquire the knowledge needed to implement PnP components and assemble a simple CubeSat system from PnP components they developed by working through hands-on examples. A survey was performed of existing courses to gain insight and avoid creation of a course that previously existed [11-13]. Another important objective was to promote a community of developers to create interchangeable components. The entire course was developed around the PnP component development process. The PnP component development process can be thought of as a flow, where each step logically flows to the next. This flow as applied to CubeSats is known as CubeFlow. CubeFlow component development consists of the three main phases described below:

1. Creating an eXtensible Transducer Electronic Data Sheet (xTEDS), which allows the component to self-describe. The xTEDS document fully describes all of the data a component can produce, any commands it accepts, as well as any services it can provide. This document is stored in standard XML and can be quite verbose. The xTEDS is a central piece of the PnP model and is viewed by many as complex and daunting. The CubeFlow process alleviates this by allowing the xTEDS to be quickly created through a dialog-based web application, which insures the completeness and correctness of the document. The xTEDS can be thought of as a datasheet. Every manufacturer that sells a component provides a datasheet for customers to understand how the device reacts to specific inputs. The xTEDS is no different.

2. Creating the interface that allows the component to be incorporated into a PnP system. In a SPA system, this interface is known as an Appliqué Sensor Interface Module (ASIM). The ASIM holds the xTEDS. The ASIM is implemented as a simple hardware/software layer between a physical device and the satellite PnP bus software known as the Satellite Data Model (SDM). In the CubeFlow system, most of the functionality needed to provide these interfaces can be inferred from the xTEDS document. This allows the majority of this interfacing code to be automatically generated, leaving the component developer to write only the code to extract data from the sensor or device. This software is flashed onto a simple microcontroller that is connected both to the physical sensor or device and also to the PnP satellite bus (see Figure 1).

3. Writing an application to exercise all of the component's functionality. Components in a system need to have an application which controls and samples data from those components. In a SPA environment, an application needs to do this utilizing the SDM software. The component must be accessed based on its capability and not its physical location on the bus. However, the xTEDS already fully describes this interface and an entire application can be auto-generated. This application is capable of fully exercising all of a device's functionality as defined in its xTEDS, allowing the developer to complete the development flow by testing out the functionality of the device.

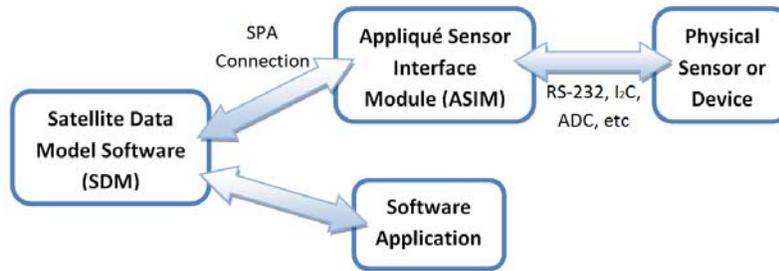


Figure 1 Basic SPA System Used in CubeFlow

Under the direction of the Configurable Space Microsystems Innovations and Applications Center (COSMIAC) and the Space Dynamics Laboratory (SDL), a pilot course was created to teach these topics. The first course was presented in Logan, Utah, at the SDL in May 2009. This course was attended by 12 different organizations including the Air Force Institute of Technology, Naval Postgraduate School and two NASA Centers. The two-day course contains an approximately 50/50 division between traditional presentation style education and hands-on training. The average student has a very limited knowledge of CubeSats or SPA specifics.

The remainder of this paper is organized as follows. In the next section, we discuss the past courses. The web tools are presented as well as a breakdown of training materials. Future training equipment will be discussed. Finally, we discuss current project status.

A. PAST COURSES

To further expand the knowledge of Plug-and-Play (PnP) in an attempt to broaden the acceptance of this standard, we put together a two-day course integrating hands-on training and traditional presentations. With the support of the Space Dynamics Laboratory (SDL), and the Configurable Space Microsystems Innovations and Applications Center (COSMIAC), we created a collection of presentations discussing CubeSats, available hardware, online resources, and the underlining key concepts behind the standard. In addition to the training, we wanted to make sure that each participating organization would go home with not only the education, but the appropriate resources to replicate the demonstrations and then hopefully develop their own PnP components. Consequently, each organization was given a flight emulation kit. Included in each kit was all the necessary hardware to develop a PnP sensor, communicate with the satellite data model (SDM) via a Space PnP Avionics USB (SPA-U) connection, and replicate a ground station between the satellite and the included laptop through an 802.11 wireless connection. Each organization was also given access to SDL's online resources enabling them to easily create ASIM source code, xTEDS, and SDM tester applications.

1. Logan, Utah – May 2009

The first training course was presented in Logan, Utah, at the SDL in May 2009. This two-day course was presented to more than 50 attendants from 12 different organizations including the Air Force Institute of Technology, Naval Postgraduate School and two NASA Centers. Among the keynote speakers were Dr. Jim Lyke from the Air Force Research Laboratory, and CubeSat pioneer, Bob Twiggs, a professor from Stanford University. Following two days of technical presentations, demonstrations, hands-on training, speakers, coordinators and attendees engaged in informative discussion, and provided the instructors with constructive criticism and positive feedback. The first steps to developing a Cubeflow community had been taken and new challenges became evident. These included a number of technical challenges such as minimizing the total power consumption of the PnP ASIM, as well as management challenges such as keeping the newly expanded Cubeflow community together. In response to the rapidly expanding community challenge, we have begun sending out a quarterly Cubeflow newsletter discussing current events in the Cubeflow community. An attempt to address the ASIM power concerns has also been addressed and is the primary driving force behind the development of a lower bandwidth, much lower power SPA-1 protocol. The first training course proved to be a wonderful utility for allowing the trainees to show us where to focus our development efforts and was unanimously concluded to be an overwhelming success.

2. Albuquerque, New Mexico – October 2009

The second course was held at the University of New Mexico in Albuquerque in October 2009. With the guidance and suggestions from the first course, we made modifications to the training material including improvement of the presentations content and appearance, and improvements to the flow of the hands-on exercises. This course was paid for by AFRL for the winners of the Advanced Plug-and-Play Technologies (APT) contract and was attended by six

different contractor teams. The attendees sat through technical presentations and discussions of various PnP concepts and received hands-on training including the development of xTEDS, ASIMs and SDM compliant test applications. Each contracting team was sent home with a flight emulation kit identical to the Logan, UT course. Again, this course was concluded with technical discussions and suggestions for future courses.

3. Albuquerque, New Mexico – October 2009

The next course occurred immediately after the APT contractor course. This was the first “paid course,” where organizations paid to attend. It was attended by eight different organizations/companies including Lockheed Martin and Sandia National Laboratories. Inclusive in this training was a new flight emulation kit that incorporated some minor changes and some hardware upgrades. This was the first CubeFlow training based on a 2U satellite (the two previous courses were based around 3U satellites). This new kit also included a revision of the SPA-U router hub that was much more robust than the original, and a better demonstration sensor assembly for reduced frustration. This course was two full days of collaboration, technical presentations and hands-on training.

Web Tools

A series of web tools have been developed that greatly reduce the difficulty and complexity of the PnP component development process. This accelerated development flow takes a component from conception and design all the way through testing in a complete PnP environment. As these web tools mature, it is hoped that the design flow will not only encompass the PnP component design process, but the design of and configuration of entire CubeSats.

A. xTEDS Developer

The xTEDS Developer is an online development environment for creating and maintaining xTEDS documents. This tool allows component developers to quickly create the electronic datasheets that fully describe their device’s capabilities in a PnP system. During development, an xTEDS is constantly checked against the xTEDS XML schema to make sure that no errors are present and that it conforms to the xTEDS standard. This controlled development process ensures that the xTEDS will be valid when embedded within the device. Once complete, an xTEDS document is stored in an online repository where it can be shared with other users and satellite developers, made easily available for use in subsequent web tools, and opened again for editing or maintenance.

B. ASIM Wizard

The ASIM Wizard is the online tool that facilitates the writing of the code that interfaces between the PnP network and the actual physical device [7]. This tool is very important as it removes the complexity from developing an ASIM. This web tool generates all of the source code needed to interface with the PnP satellite network by examining the contents of the associated xTEDS document. The only code that is not provided to the device developer is the code to physically pull the data or readings from the developer’s device. This code would have to be written regardless of the development paradigm. The hope is that using this tool should make the idea of developing an inherently more complex PnP component easier.

C. SDM App Wizard

After a component developer has written an xTEDS and programmed their ASIM, the last step would be to actually test their newly created PnP component on a PnP network. The SDM App Wizard is an online tool, which looks at the xTEDS document, generates, and compiles a complete C++ application which communicates with the PnP satellite bus to both control the ASIM, as well as receive and understand any data the ASIM produces [4,5].

Web-BASED Community Building

Another CubeFlow objective is to develop a community of PnP component developers. To facilitate the development of this community, several features have been added to the website that hosts the CubeFlow development tools. These features include both a central location to view news and updates relating to CubeFlow, as well as a set of forums where users can post questions or discuss all things related to PnP development. All of the documentation related to this development is also posted on this same Web site. Users are also allowed to upload and share files and documents with other users. As of January 2010 there were 187 unique accounts registered on this web site.

TRAINING MODULES

Although the long-term goal is to videotape the entire course, currently only the soft versions of the presentations are available through the Internet. The modules are broken down into two large categories: presentations and practical exercises. Although there are supplemental and additional presentations that add to the course, the main course consists of 11 presentations and three practical (or laboratory) sessions.

A. Presentations

1. SPA and CubeSat Overview

What is necessary for the course to be successful is a basic level of understanding on all the acronyms and also the “forest level” view from the point of view of AFRL. It begins by giving an explanation of the three main components of CubeFlow: SDM, ASIM and xTEDS. It gives enough detail for the attendees to be able to understand the reason for the three components and the way they interface together. After this is completed, the next phase is to explore CubeSats. This is critical since many of the attendees have no understanding of the CubeSat progress and its international reach. What is important here is how AFRL has modified the generic CubeSat structure to accommodate what they refer to as “facets” or Maximal Symmetric Envelopes (MSE).

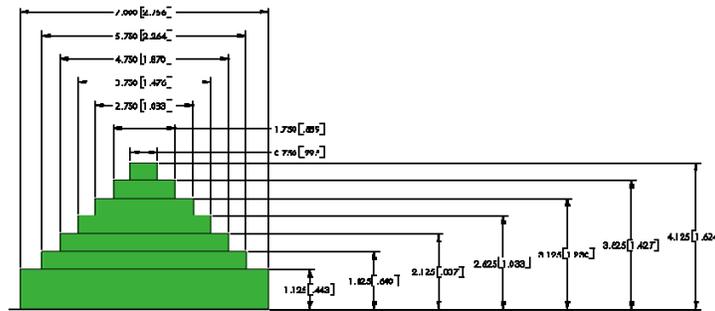


Figure 2 MSE Approach

The MSE approach has two main advantages. The panels have been designed so that each one can be developed as a separate nanomodule (integrated and assembled individually), then brought together and integrated to form spacecraft. In addition, MSE gives the ability to build larger satellite structures out of smaller panels. It is important for the students to understand the path ahead from CubeSats to larger satellites [14]. Current developments include miniaturizing and hardening the ASIMs. With this miniaturization, it becomes possible to partition the space allotment as designated above so that multiple components could share the same faucet while still complying with the MSE standard.

2. SDM and PnP

This presentation explains how the PnP paradigm allows for larger satellites to be built from smaller components and the overall view for how this can be accomplished from the AFRL perspective.

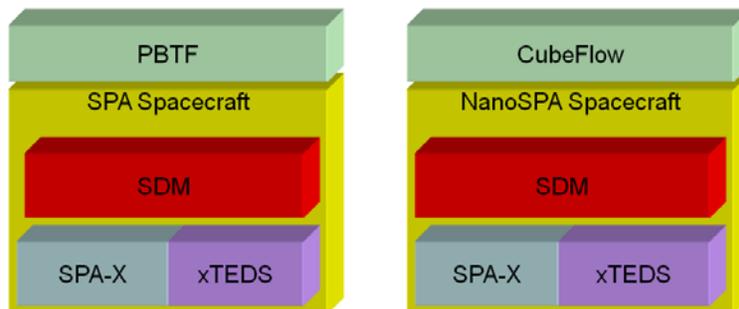


Figure 3 PnP Spacecraft

There are two parallel tracks to PnP spacecraft. For larger spacecraft there is a system called Pushbutton Tool Flow (PBTF) that will allow developers to create larger systems from individual components. There is also a smaller satellite tract based on CubeSats that is referred to as CubeFlow. The key to remember is that both systems are based on the PnP and SPA standards.

This presentation also develops the SPA protocol standards for the students. It explains the standards that are currently used (SPA-U and SPA-S) as well as the standards that are in development (SPA-1 and SPA-O).

3. CubeSat Launch and Development

Many of the students may be familiar with satellite basics as presented in the Space Mission Analysis and Design (SMAD) manual [15]; however, many are not familiar with CubeSats. This presentation offers an overview. The presentation was developed in collaboration with Dr. Jordi Puig-Suari of Cal-Poly and explains the possibilities and limitations related to CubeSats. However important the CubeSat is, the real driving factor is actually the launcher called the P-Pod. The standards related to these small satellites are driven by a desire to keep the size, weight and cost at a level where this type of development can still be accomplished through a university led effort. The standards are also written to keep them simple enough that a student can read, understand and comply with them. CubeSats are also becoming enablers of new technology and as such they are being embraced by a number of organizations including government, industry and academia from many different countries. Part of the challenge of CubeSats are the need to think smarter about how missions are accomplished instead of just thinking bigger.

4. CubeFlow Development Kits

This presentation delivers the first exposure to the actual equipment that will be used in the course. There are two kits that can be used interchangeably in the course. The first is the flight emulation kit and the second is the bench kit.



Figure 4 Flight Emulation Kit

The flight emulation kit contains all the items needed to create individual modules and then configure the entire satellite as demonstrated in the training course. This kit contains an aluminum satellite, which is beneficial for training and for demonstrating the ideas of CubeSats, but is not a “flight” hardware unit. The wireless module allows for configuring the satellite remotely, which is very beneficial in demonstrating the potential capabilities of configuring (or reconfiguring) the system in flight.



Figure 5 Bench Kit

The Bench Kit contains all the items needed to develop an xTEDS, ASIM, and SDM to demonstrate their functionality. This kit is designed more for academic institutions that might be interested in developing training programs in CubeFlow at their schools. It contains inexpensive commercial parts but has the same functionality as the more expensive flight kit.

Either kit can be used for the training. The actual kit components are individually explained in this presentation. Students are given history on the ASIMs and presented with the vision of where the ASIM development is headed as well as the history of how ASIMs have come from entire board solutions to small, single-chip devices.

5. Introduction to CubeSat Tool Flow

This section begins the introduction of the CubeFlow system, as well as explaining how the course will proceed. The ASIM is identified as the device that will handle all bus demands for the satellite or training system. All associated libraries are identified. In addition to the basics of the SDM, there is also an introduction to the capabilities that are present in the “tester applications.” This application allows ASIM developers to be able to test their devices after their creation.

6. xTEDS Introduction

The xTEDS stands for the Extensible Transducer Electronic Data Sheet. It is an XML document that is human readable. It is the data sheet that hardware developers can share between each other. It can be parsed and allows system developers to easily understand what the expected inputs and outputs are for a developed piece of hardware.

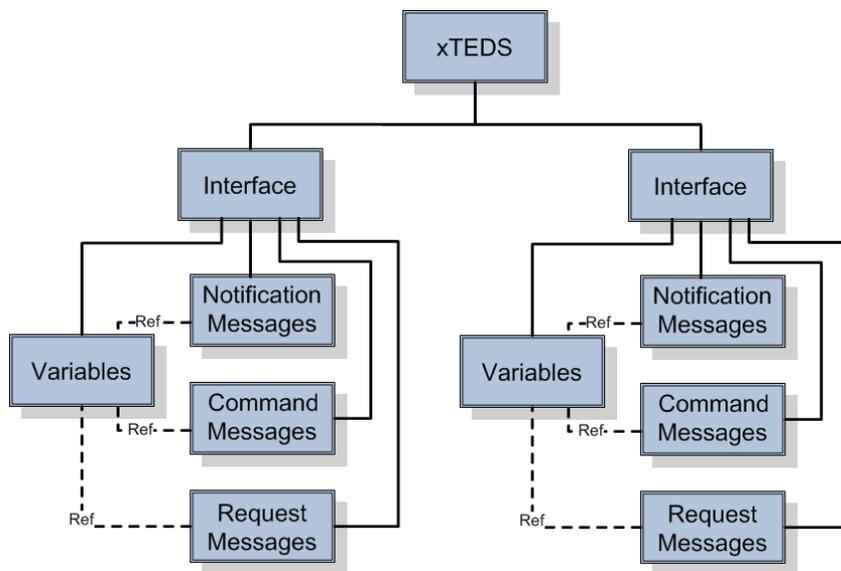


Figure 6 xTEDS Structure

Figure 6 shows the structure and the three types of messages that are allowable under this system. Online tools allow this document to be easily created while greatly reducing the chance of errors. Another critical aspect to the xTEDS creation is the use of the Common Data Dictionary (CDD). The CDD is a living document that is controlled by the community that provides developers with “common” terms for ensuring that all users stick to standards for module interfacing.

7. ASIM Coding Tools

The ASIM holds the xTEDS and acts as the physical interface to the entire system.

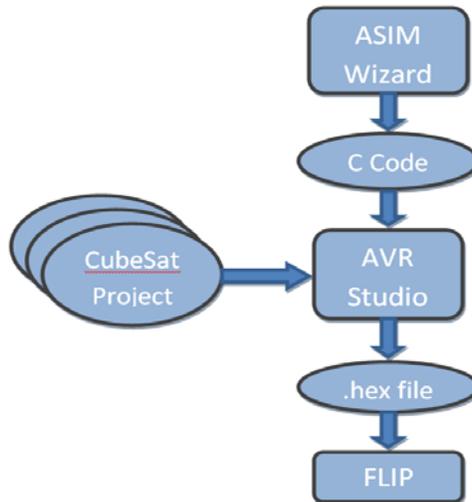


Figure 7 ASIM Programming Flow

There are two types of ASIMs used in the course. The first is the GE MiniASIM and the second is the Atmel USBKey. Both of these devices are programmed using two open source programming tools. The design flow (as shown in Figure 7) starts with the ASIM online wizard. The next two tools are the AVR Studio and the Flexible In-system Programmer (FLIP). AVR Studio is Atmel’s feature rich IDE code development environment and FLIP is used to actually program the ASIM using a common USB cable.

8. SDM Application Introduction

Once the students understand the module components associated with xTEDS and ASIMs, then it is necessary to explain the “system” concept. The backbone of the CubeFlow satellite is the Satellite Data Module (SDM). The students must understand that there are two types of users from the perspective of the SDM: Device Developers and Application Developers. The device developers often utilize the ASIM Tester Application to test or exercise all of the ASIMs capability within the SDM environment. The application developers write software modules to utilize the various modules to accomplish on orbit missions.

9. Building Your Own ASIM

Although there are two types of ASIMs utilized in the course, the students have to understand there are virtually limitless types of ASIMs ranging from standard 8051 microprocessors to programmable logic soft core implementations. All ASIMs have the same messaging protocol but don’t necessarily have the same hardware.

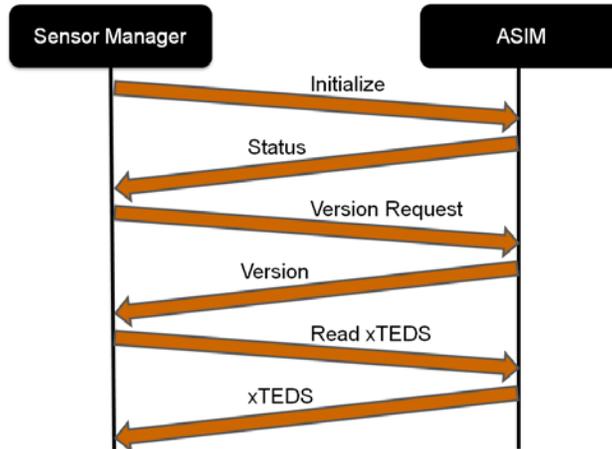


Figure 8 ASIM Registration Protocol

This presentation explains the ASIM message syntax to the students. It also explores the ASIM handshaking scheme as shown in figure 8.

10. SDM Resources

The entire repository of the SDM development system is located on servers at the Space Dynamics Laboratory. This presentation introduces the student to all of these resources. There is documentation on standards and various items related to system development. The site contains recorded videos related to all the tutorials. The students are given login and password information and shown how to maneuver through the various Web site sections [6, 8].

11. xScale Cross Compiling

The CubeSat development kits use an xScale-based, single-board computer. This presentation shows the users how to utilize the cross compilation environment for that architecture. This environment is provided for them on a Linux virtual machine installed on the laptops which are used during the course. While this knowledge is not necessary in the duration of the course, it will be of use to those who wish to do any further software development after the course.

B. Laboratory Sessions

The laboratory section is the “hands-on” portion of the two-day course. The next section will examine each of the three lab sessions.

1. Lab1. Thermister Test

This lab is designed to familiarize students with the skills needed to create their own xTEDS, program an ASIM and test it with a software application [9, 10]. The students watch the experiment first and then (with the assistance of instructors) do the activity. The project has the students utilize a temperature sensor on the GE ASIM and then display the temperature through an SDM application in both Fahrenheit and Celsius.

2. Lab 2. Integrating Off-Board Device

The second laboratory section builds on lab one. In the first lab, the students developed a system that did temperature sensing and reporting utilizing the AD7415 temperature sensor that is mounted on the GE ASIM. In this second lab, the students are provided an external AD7415 temperature sensor mounted on an adapter that provides them with the capability to hard wire in the chip (emulating an external sensor) to an existing ASIM device.

3. Lab 3. Create Your Own ASIM

This final lab is where students are allowed to take all the skills they have mastered throughout the course and all the available hardware to create their own system. This section utilizes the AT90USBKey to develop a temperature sensor and display the output temperatures. The AT90USBKey board provides full functionality (as an ASIM) but

in addition, it also has a wide variety of input and output capabilities. It has a five-position joy stick, LEDs and temperature sensors. This lab allows students to explore all these capabilities while creating a working system.

FUTURE TRAINING EQUIPMENT

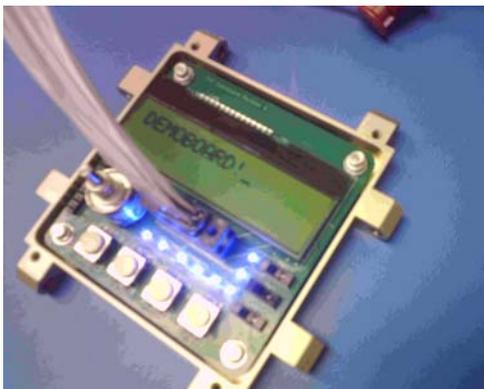


Figure 9 CubeFlow Demonstration Board

One of the first components developed to assist with this training is the CubeFlow Demonstration Board. The board is a multi-function training module designed for PnP small satellite training. The primary purpose is to demonstrate a range of basic commands, receive messages from the SDM and to improve the CubeFlow training. To accomplish this task, a number of sub-components were connected to the input and output pins of a GE ASIM and it was then mounted into a 1U facet of a small satellite. Then, using the CubeFlow Tools, the xTEDS, ASIM source code, and generated SDM application are developed. The end result is a module equipped with a LCD character display, LEDs, buttons, switches, a buzzer, and a temperature sensor. With this module, training can be performed on a variety of demonstrative applications, enhancing the impact of the training course. While the board is not currently a part of the flight emulation kit, it is demonstrated in the course and plans to include it in future courses are under review.

FUTURE COURSES

After thoroughly examining the past course surveys and taking careful consideration of suggestions from discussions with attendees, we have made significant changes to the future courses technical training material and flight emulation kit hardware. In the previous three courses, the SDM software was loaded into the Gumstix board from the Vulcan Wireless corporation. This module was used for both command and data handling and communication to the ground station “Laptop” via an 802.11 wireless radio. Connected to the Vulcan board via SPA-U was a four-port routing hub giving access to potentially four different PnP sensors. In the new flight emulation kit, the command and data handling and communication have been separated. In addition, the SPA-U routing hub has been combined with the command and data handling module. This was accomplished with hardware from the Data Design Corporation called the NanoCDH. NanoCDH integrates a 32-bit OMAP microprocessor with a six port SPA-U robust hub effectively condensing three facets worth of hardware into one. Consequently, a separate PnP radio module is now required and currently under development. Among the many advantages to the NanoCDH is a means of bridging the gap between SPA-U and SPA-S. This has been accomplished with an addition to the NanoCDH called the NanoBridge. This alleviates a lot of the concern for people that require more bandwidth than is available with SPA-U. Conversely, for low-bandwidth modules such as temperature sensors, another SPA protocol is rapidly being developed. This protocol is called SPA-1 and is based on an I²C two-wire interface.

In addition to improving the essential hardware for PnP compatibility, we have also developed a better training model in an attempt to increase the impact of the hands-on portion of the course. As discussed above, the CubeFlow demonstration board offers a wide range of devices and shows the potential available to the attendees. These course attendees get a deeper understanding of the different mechanisms involved in SPA communication.

We hope to integrate a NanoCDH, PnP communication module, and a demonstrations board in the upcoming CubeFlow courses. We look to begin these courses and incorporate the changes as soon as May 2010.

Summary

As of November 2009, approximately 29 different organizations have been formally trained on the CubeFlow system and there were over 185 registered users on the CubeFlow website. The experience gained from the past training courses will help to make future courses even better. The objectives of training organizations in the process of developing PnP components are starting to be realized. Community development is also continuing to move forward. With the upcoming advances in the infrastructural elements, including the miniaturization of ASIM and the development of flight-worthy subsystems, this growth should continue. The accomplishments achieved to date are indeed encouraging.

References

- [1] D. Lanza, J. Lyke, P. Zetocha, D. Fronterhouse and D. Melanson, "Responsive Space Through Adaptive Avionics," Responsive Space Conference, Los Angeles, CA, April 19-22, 2004.
- [2] D. Fronterhouse1, J. Lyke and S. Achramowicz, "Plug-and-play Satellite (PnPSat)," Responsive Space Conference, Rohnert Park, CA, May 7-10, 2007.
- [3] T. Morphopoulos, L.Hansen, J. Pollack, J. Lyke and S Cannon, "Plug-and-Play – An Enabling Capability for Responsive Space Missions," Responsive Space Conference, Los Angeles, CA, April 19-22, 2004.
- [4] K. Sundberg, S. Cannon and T. Hospodarsky, "A Satellite Data Model for the AFRL Responsive Space Initiative," Small Sat Conference, 2006.
- [5] M. Martin, D. Fronterhouse and J. Lyke, "Implementation of a Plug and Play Satellite Bus," SmallSat Conference, 2008.
- [6] CubeFlow Course Related Documentation. Available at: <http://gonzales.cs.usu.edu/drupal/node/49>
- [7] xTEDS Schema. Available at: <http://gonzales.cs.usu.edu/drupal/node/24>
- [8] CubeFlow Lab Walkthrough. Available at: <http://gonzales.cs.usu.edu/drupal/node/58>
- [9] CubeFlow Online Tool Suite. Available at: <http://gonzales.cs.usu.edu/drupal/node/9>
- [10] Dukeworks website with SPA standards. Available at: www.dukeworks.org
- [11] Space, Satellite & Aerospace Engineering Courses and Technical Seminars. Available at: http://www.atcourses.com/space_courses.htm
- [12] IET Satellite Communications summer school - training course. Available at: <http://www.satsig.net/satellite-training-iee.htm>
- [13] Application Technology Strategy Inc. Education and Training. Available at: <http://www.applicationstrategy.com/education.htm>
- [14] C. McNutt, R. Vick, H. Whiting and J. Lyke, "Modular Nanosatellites – (Plug-and-Play) PNP