A SPA-1 Enabled Plug-and-Play CubeLab for ISS Payloads

Zachary Jacobs*, Samir Rawashdeh†, James E. Lumpp Jr.‡

University of Kentucky, Lexington, KY 40506

This paper describes the design and development of a system to provide a Space Plug-and-play Avionics (SPA) interface in a CubeLab form-factor to extend the capabilities of the International Space Station (ISS) to interface directly with SPA-1 compliant devices. The CubeLab is a new payload standard for access to the ISS for small, rapid turn-around microgravity experiments. CubeLabs are small (less than 16”x8”x4” and under 10kg) modular payloads that interface with the NanoRacks Platform aboard the ISS, receive power and transfer data using plug-and-Play Universal Serial Bus (USB) standard. The SPA architecture is a modular technology for spacecraft that provides an infrastructure for modular satellite components to reduce the time to orbit and development costs for satellites. The new system described in this paper allows developers to easily operate their SPA-1 based experiments and payloads aboard the ISS. In addition, developers of new SPA-1 devices can rapidly access the microgravity environment of space.

I. Introduction

This paper presents a CubeLab bus system, designed by the Space Systems Lab at the University of Kentucky (SSL), that is Space Plug-and-play Avionics (SPA) compatible. This infrastructure would make the International Space Station (ISS) capable of interfacing with SPA devices for the first time. Microgravity experiments can be developed using the SPA self-describing standard; the CubeLab bus can operate the experiment using the embedded device description, and would allow commanding from Earth for more involved experiments. The CubeLab bus can also be used to rapidly and inexpensively increase the Technology Readiness Level (TRL) of spacecraft avionics by operating in the microgravity environment on the ISS.

The CubeLab standard offers an opportunity for relatively inexpensive access to the International Space Station for small payloads. The CubeLab standard provides simple power and communications interfaces for payloads by leveraging the USB standard. Power for the payload is provided through a USB type-B connection and all data transfers use the USB Mass-Storage protocol§, which is well supported by many hardware manufacturers.

SPA, developed by the Air Force Research Lab (AFRL), is a modular technology for spacecraft components based on “plug-and-play” technology. AFRL has been working to achieve Operationally Responsive Space (ORS) by reducing the cost and time to launch a spacecraft. This effort has led to systematic reductions in time for all aspects of developing and launching a spacecraft. This requires standardized systems to rapidly assemble a spacecraft and integrate it with a payload. Given the complexity of the different systems on a spacecraft, AFRL developed a “plug-and-play” (PnP) standard, called SPA, to facilitate rapid integration. SPA implements a self-organizing network of modules where components are self-describing and attach to a standardized data and power bus. This allows a satellite to be assembled by connecting the automatically discoverable devices to the SPA network, and focusing the development time on the application that uses the data and services from the SPA modules. Ideally, this alleviates integration time by eliminating module and bus design, given that SPA-enabled modules that meet the mission requirements are available.

* Graduate Research Assistant, Electrical and Computer Engineering, 453 F. P. Anderson Tower.
† PhD Candidate, Electrical and Computer Engineering, 453 F. P. Anderson Tower.
‡ Associate Professor, Electrical and Computer Engineering, 453 F. P. Anderson Tower.
II. Background

A. The CubeLab Standard

The CubeLab standard leverages the heritage and developer experience with the CubeSat form factor for free-flying satellites so that, in general, CubeSat structures, products, and hardware modules can serve as CubeLab components as well. A single unit (1U) CubeLab is specified as a payload that is a 10 cm cube or smaller. CubeLabs, like CubeSats, can be designed as multiple units where more volume is needed for the experiment. The CubeLab Interface Control Document (ICD) can be referred to for exact size constraints**.

The NanoRacks Platform serves as the interface between individual CubeLab Modules and the ISS, providing mechanical mounting points and electrical connections for power and data connectivity. The NanoRacks Platforms are installed inside an EXPRESS rack locker onboard the ISS. Figure 1 shows an exploded view of the installation configuration of a set of possible CubeLab variants and the NanoRacks Platform.

Each of the two NanoRacks platforms on orbit can accommodate 16U worth of CubeLab payloads in any configuration. CubeLabs of any size (1U, 2U, 3U, 4U, etc.) can be installed onto a NanoRacks Platform. The front panel of the NanoRacks Platform that is visible to the astronauts contains 16 USB type B connectors and a standard ISS power connector. The 16 USB connectors on the front panel provide the astronaut with a direct data connection for commanding and experiment data retrieval between the CubeLabs that are installed on the NanoRacks Platform and the EXPRESS Laptop Computer (ELC). The standard ISS power connector on the front panel connects to the 28V power available on the ISS. The NanoRacks Platform provides voltage step-down, distribution, filtering, and isolation of the 5V connections to each of the 16 USB ports on the side of the NanoRacks Platform. As of May 2012, there are two NanoRacks Platforms installed on the ISS. Figure 2 shows Astronaut Shannon Walker after the successful installation of NanoRacks Platform-2, CubeLab-3 and CubeLab-4. The front panels of each NanoRacks platform can be seen in the figure.

Several CubeLab experiments have been operated on the ISS to date. The University of Kentucky Space Systems Laboratory has flown FIRSTLab, CubeLab-2, CubeLab-3, and CubeLab-4. These 1-Unit CubeLabs performed hardware radiation susceptibility experiments, which are leveraged in the design of the CubeLab bus and tested the operation of the Nanoracks Platforms. In addition to The University of Kentucky's payloads, a variety of commercial and educational payloads have flown.

B. Space Plug-and-play Avionics (SPA)

SPA was developed to reduce the complexity of connections between avionics components on spacecraft. Ultimately, this led to the development of a PnP-style architecture that defines hardware and software connections and the interactions between the parts in the system. The SPA architecture defines the following: SPA components, SPA interfaces, ASIMs, SPA Networks, SPA systems, SPA middleware, Ontology, and System Conventions. SPA hardware is referred to as a device or a component and SPA software is referred to as application. The properties of devices and applications are described in xTEDS (extensible Transducer Electronic DataSheet). SPA interfaces define the physical layer which could be based on an optical network (SPA-O), the SpaceWire network (SPA-S), USB (SPA-U), or I²C (SPA-I). The purpose of the ASIM (Applique Sensor Interface Module) is to provide an interface between non SPA-compliant spacecraft components and the SPA network. It generally contains a processor that handles the translations and memory storage necessary to store the xTEDS. SPA components are hardwired together through the use of the physical layer associated with the standard type to form a SPA network. Networks of differing SPA interfaces can be created through the use of a SPA bridge. A SPA system is a network of SPA components. The SPA network can consist of an entire spacecraft platform or just a subset of a larger non-SPA network. SPA middleware consists of a software component referred to as an SDM (Satellite Data Manager). The SDM provides a service that allows SPA devices to discover and join the SPA network. The SDM can loosely be referred to as the “traffic cop” that discovers new devices on a network and records the capabilities of the new device in a data registry so existing SPA devices can have access to the new data available.

Current examples of SPA technology

To date, only a few spacecraft have taken advantage of SPA technology. The initial attempt at building a satellite based entirely on the principles of SPA was initiated by AFRL with PnPSat-1. The design techniques used on PnPSat-1 employed modular components whenever possible. This included structure items as well as electrical items which took advantage of the modularity of SPA. PnPSat-1 was originally scheduled to be launched in 2009 but is now being used strictly for education.

RAMPART is a CubeSat that contains an additional secondary circuit board with the intent of gathering radiation performance statistics on three types of SPA-1 modules. There are three ASIMs onboard: two radiation hardened ASIMs (one made in the US and one made in Sweden at ÅAC Microtec) and one commercially available PIC. The orbit of this CubeSat is such that the apogee is 1200km and should provide an adequate radiation dose to test the radiation susceptibility of the ASIMs onboard. RAMPART is in the process of being manifested on the Falcon-1e launch that currently has no launch date.

QuadSat-PnP is a nanosatellite under development by a coordinated effort between University of Applied Sciences Bremen, OHB System, ÅAC Microtec in Sweden, and the AFRL in the United States. This nanosatellite is the first in its class to be based completely on a PnP model, specifically SPA. The SPA-enabled subsystems include a TDRS software-definable radio, a power distribution unit, an inertial measurement unit, and other power control and satellite management units. As of the publication of this paper, there was no firm launch date for QuadSat-PnP.

TechEdSat is a CubeSat being built by San Jose State students, NASA Ames Research Center, and ÅAC Microtec to demonstrate SPA technology on a CubeSat and to be one of the first CubeSats to be deployed from the ISS. TechEdSat will also investigate using Iridium and ORBCOMM satellite to satellite communications as a means of eliminating a physical ground station from the mission requirements of a nanosatellite mission. The satellite is based on SPA hardware and software that has been developed by ÅAC Microtec.

COSMIAC is currently building a CubeSat, named Trailblazer, which will serve as a proof-of-concept for SPA technology. This satellite is based on the SPA-1 standard and will work to adapt existing COTS technologies to the SPA network. This will prove the feasibility of adapting commonly used satellite components to operate on the SPA network. The overall mission timeline of the satellite is to go from design to delivery in under a year. This will validate the reduced schedule claims that are part of the SPA paradigm. A SPA-1 ASIM is being developed and is based on a PIC microcontroller. There will be two SPA-based payloads on the satellite: a dosimeter and a rapid prototyped IMU. Trailblazer is currently manifested on the ELaNa IV mission and is scheduled to be launched in 2012.

III. Plug and Play Capable CubeLab Bus

A. Application of SPA in CubeLabs

Due to the large breadth of complexity of avionics used in spacecraft, data rates and power requirements vary greatly. This has led to the creation of four different SPA interfaces. Starting with the most complex and power-capable, the physical layer of the SPA interfaces are as follows: SPA-O, SPA-S, SPA-U and SPA-1. SPA-1 is geared towards simple devices with minimal power requirements. This standard fits well with the power and size constraints of the CubeLab standard, therefore this paper will focus only on the SPA-1 standard. Other SPA devices of different interfaces would be compatible with the SPA-1 implementation using an appropriate bridge, but would ultimately be limited by data throughput and power availability by the SPA-1 connection back to the CubeLab bus.

SPA is typically used to rapidly assemble spacecraft, where the majority of the time is spent in application development to create a completed and configured spacecraft. In contrast, the general-purpose aspect of the SPA-enabled CubeLab bus makes the paradigm significantly different. The CubeLab bus is pre-developed to support a general SPA device, and the assembly with the SPA experiment is done by the astronaut on orbit where the application is a configuration script for the CubeLab bus that can be uploaded to operate the specific device. Collecting experiment data using the NanoRacks system is dependent on astronaut interaction, which can be intermittent. Therefore, a level of autonomy is required to interact with the SPA experiment, as well as the ability to command and reconfigure the CubeLab bus depending on the SPA experiment’s need.

B. Concept of Operations

Figure 3 shows the data flow between a CubeLab onboard the ISS and the CubeLab developer on Earth. Data are generated on the SPA payload and stored on the CubeLab bus. Upon request for a data collection, an astronaut makes a USB connection between an ELC and the appropriate port on the front panel of the NanoRacks Platform. The CubeLab bus enumerates on the ELC as a USB Mass Storage Device and the CubeLab bus data files become available to the astronaut on the ELC. The astronaut transfers the data files over to the local ELC. Once complete, the Payload Rack Officer (PRO), which is part of the Huntsville Operations Support Center (HOSC), remotely transfers the data from the ELC through the Tracking and Data Relay Satellite System (TDRSS) and into Principal Investigator Microgravity Services (PIMS). The files are then securely transferred to the CubeLab development team.

The default operation and downlink path for the SPA-enabled CubeLab is based on the xTEDS. The xTEDS, as the SPA device’s self-description datasheet, identifies the device’s generated data, its data needs, and software functionality. The CubeLab bus concept is to use the information in the xTEDS to generate a command set which emulates the SPA-1 protocol. The CubeLab bus uses the information contained in the xTEDS to operate the device.
according to its datasheet; it requests the data the device generates at the rate it describes and logs those within the CubeLab bus for later download to the payload developer.

The uplink path provides greater flexibility for experiments that may not find the concept of operations based on the xTEDS sufficient. In the spirit of a general purpose solution, the CubeLab bus can be configured and provide additional functionality by uploading configuration scripts to its file system. Astronaut time can be scheduled to connect to the CubeLab bus and load a configuration script that has been uploaded to the ISS via TDRSS, similar to the download operation. The configuration scripts can customize the CubeLab bus operation to fit a certain SPA experiment’s needs by making SPA requests that are not described in the xTEDS.

C. System Requirements

The main motivation behind the original design description of the CubeLab bus was to accommodate generic payloads in the CubeLab form factor. The main driver behind the design of the CubeLab bus was the NASA Ames Research Center mission, Microsatellite in-situ Space Technology (MisST). The MisST mission was a free flyer 2U CubeSat payload which studied the effects of microgravity on C. Elegans. The SSL was interested in adapting the payload of MisST to operate in the NanoRacks platform. This mission required the use of a high power imager and had strict thermal constraints. This lead to the development of an extended power system for the CubeLab bus which is capable of charging a NiMH battery pack that can be used to provide higher instantaneous power than the power limitations listed in the CubeLab bus ICD.

The microorganisms in the AMES payload have specific temperature requirements to live, so temperature control becomes important. Additionally, good biological research requires a fairly constant temperature throughout the duration of a study, so even if the ambient temperatures are adequate for the microorganism to live, the normal fluctuations of ambient temperatures on the ISS would not be suitable for biological research. It was decided for the CubeLab bus to provide a less-than-ambient temperature using a peltier thermoelectric cooler. This cooling effect would be leveraged and the MisST payload would have used heaters to raise the temperature to a known state around ambient temperature. This method of cooling and heating provides a precise method of temperature control for the mission.

The design requirements for the CubeLab bus are mainly motivated by supporting the NASA Ames payload in the CubeLab form factor as described above, along with supporting SPA-1 based payloads and several derived requirements from the CubeLab Interface Control Document, the USB standard, and the SPA-1 standard. The requirements can be summarized as follows, and the following sections discuss the solutions to the major challenges.

**CubeLab Specification and NanoRack requirements**

1. The CubeLab and its payload shall consume less than 400mA continuous at 5VDC.
2. The CubeLab bus shall adhere to the USB Mass Storage Device class when a laptop is connected to the NanoRack front panel.

**Interface Requirements**

3. The CubeLab bus shall provide a mechanical attachment point for the SPA payload.
4. The CubeLab bus shall not interrupt USB enumeration during astronaut data collection.
5. The CubeLab bus shall not interrupt payload (SPA device) services during astronaut data collection.
6. The CubeLab bus shall provide the PnP mechanisms associated with the Satellite Data Model (SDM) for a SPA-1 network.

**Functional Requirements**

7. The CubeLab bus shall prevent data loss during astronaut data collection.
8. The CubeLab bus shall be capable of interrogating SPA-1 devices to obtain data requested by the designer and log it for later downlink.
9. The CubeLab bus shall be capable of dynamically commanding a SPA-1 payload.

IV. System Design and Features

A. Design Overview

A CubeLab Bus hardware and software solution that meets the requirements described in the remainder of the paper. The hardware chosen to provide the USB Mass Storage Device interface to the ELC and the data retention of
experimental data is discussed. A modular and feature-rich software design that meets the constraints and requirements is also described. Figure 4 shows the overall hardware block diagram of the system.

![Figure 4. Block Diagram of CubeLab bus showing support for SPA device.](image)

B. CubeLab Interface to NanoRack Platform

A major element of the CubeLab bus design is the interface to the NanoRack Platform which includes the subsequent data connection to the ELC. As shown in Figure 1, each CubeLab plugs into the NanoRack Platform at a USB attachment point. This connection provides the CubeLab with regulated 5V power and a USB data connection when an ELC connection is present. The ELC connection to the NanoRack Platform is made by using the front panel USB connection corresponding to the CubeLab (also shown in Figure 1). The front panel USB connector is directly routed to the CubeLab with the exception of noise and protection circuits that are transparent to the data connection.

To provide CubeLab experiments with uninterrupted power, the NanoRack Platform was designed to provide continuous power to the side panel USB connections whenever the NanoRack is turned on. Due to the persistent power on the USB interface, the CubeLab bus must detect astronaut presence using the USB data lines. The detection of data lines is discussed further in the next section. In order not to require special software or driver installation on the ELC, the CubeLab is expected to respond to astronaut interaction and appear as a USB Mass Storage Device.

To fulfill Requirement 2 listed above, the USB stack and Mass Storage Device functionality were implemented using a USB-to-SD card reader with bypass IC. This chip provides support for full-speed USB communications with an SD card using the Mass Storage Device Class. Through use of a “pass-through” mode, the card reader IC also provides an interface from a microcontroller to the SD card. This setup allows the microcontroller to use the SD card to store experimental data and also provide access to the same data by using the ELC.

The CubeLab bus must also provide a mechanical attachment to the NanoRack Platform, per Requirement 3. A 2U size was chosen for the bus so there are two mechanical attachment points through the form of two USB type-B plugs which stick out of the side panel of the NanoRack Platform. Since the force of gravity is negligible on the ISS, the friction of the USB connection between the NanoRack Platform and the CubeLab bus will be enough to hold the CubeLab in place. This has been tested numerous times on orbit with no adverse side effects. A mechanical enclosure has been designed with encloses the CubeLab bus board and provides mechanical attachment points for SPA-1 payloads. Initial designs can be seen in Figure 5 below.
C. Astronaut Access Transparency to Payload

The CubeLab bus is required to isolate the astronaut data collection functionality from the payload operation as shown in Requirements 4 and 5 listed above. Initial designs employed a single memory card that is used to store data from the payload and given to the card reader chip for USB data collection. This design however makes the SD card unavailable to support payload operations during astronaut data collection. A secondary data storage buffer was proposed that ensures that the main microcontroller services the payload without latency when a USB connection is present (Requirement 7).

As discussed previously, a USB-to-SD card reader with bypass chip was chosen to provide full-speed USB communications from the ELC to an SD card and also provide an interface for the microcontroller to store data on that SD card. When a data collection activity is scheduled, an astronaut will connect an ELC to the front panel of the NanoRacks Platform. This action causes USB-to-SD card reader chip to enumerate the SD card as a Mass Storage Device. This is known as card reader mode. Any experimental data which are generated is stored in a buffer for the duration of the data collection activity. Upon completion of the data activity, the ELC is disconnected and the experimental data in the buffer is then transferred to the SD card for future retrieval on a subsequent data collection activity. Transferring experimental data to the SD card is known as pass-through mode. Since Requirement 4 states that the bus shall not interrupt USB enumeration, access time will be carefully planned to ensure minimal time is spent in pass-through mode. This planning ensures USB enumeration is not delayed when card reader mode is entered.

D. Payload Interface and SPA Compatibility

With respect to the SPA device, the main purpose of the CubeLab bus is to provide PnP mechanisms associated with the SDM as shown in Requirement 6 above. Many different architectures were investigated for the CubeLab bus to handle the interaction with the SPA-1 device. Throughout that process, an ASIM based on an 8051 was developed to run on the SPA network. The xTEDS that was developed used a notification message to report the processor temperature in Celsius and Fahrenheit to the SPA network. This was used as a baseline to develop the rest of the system. To test one SPA-1 device, the entire SDM functionality in the bus is not needed. The only parts of the SPA-1 device to be tested would be known before being sent to the ISS and could be available by subscribing to

![Figure 5. Exploded view of CubeLab bus enclosure](Image)
notification messages or sending command messages (Requirement 8). The bus does not provide the full functionality of the SDM, but through the use of upload scripts and the CubeLab bus emulating the SPA-1 protocol, the differences are transparent to the SPA-1 device.

The parts of the SPA-1 components that were to be tested could be tested through sending and receiving a series of commands that are dictated by the SPA-1 protocol. We were very familiar with this protocol through the work completed by building the 8051-based ASIM. These commands could be generated on Earth, uploaded through the upload scripts mentioned in the previous section and executed at a predetermined time. The SPA-1 device would respond as appropriate and the CubeLab bus would record the data that are available. During the next astronaut data collection activity, astronauts could retrieve the data and disseminate to the SPA-1 device designer.

This system would have an identical ground model which could be used to verify the commands which were set up using the upload scripts. This also provides the SPA-1 researcher with baseline data for their research to compare with the data obtained in microgravity on the ISS.

Since the physical connection has not been standardized for SPA-1, the CubeLab bus is designed to have industry standard 0.1 in. spacing male header pins for connection of the SPA connector. A 50-pin header is also available for connection as seen in Figure 5. Mounting-hole locations on the mechanical enclosure can also be modified on a per payload basis.

V. Discussion

Previous sections have discussed the interfaces between the ELC, the NanoRack, the CubeLab bus, and the SPA-1 device. This section further expands that discussion to include the current designs being used and details required for operating and obtaining data from the SPA-1 device on the ISS.

Sample SPA device description

A typical mission is profiled in this discussion section to familiarize the reader with the entire process of designing a SPA device to be tested using the CubeLab bus system on the ISS. A simple temperature sensor will be used which is capable of reporting the temperature in Celsius using a periodic notification message at a rate of 1Hz and also receiving a command message which will toggle an LED on the SPA device.

Initially, the SSL works with the SPA device designer to come up with a command set using the SPA-1 protocol which can be used to extract data from the temperature sensor. This command set will be based on the data which are available as indicated in the xTEDS. The example temperature sensor could have an xTEDS as seen in Figure 6 below.

```xml
1 <xml version="1.0" encoding="utf-8" />
2 <xTEDS name="ExampleDevice" version="1.0">
  3 <Interface name="ExampleInterface" id="1">
    4 <Variable name="celsius" kind="temperature" format="FLOAT32" />
  5 </Interface>
  6 <Notification name="GetTemperature" id="1" negation="PERIODIC" negrate="1.000" >
    7 <VariableRef name="celsius" /> 
  8 </Notification>
  9 </Interface>
10 </xTEDS>
```

Figure 6. Example of simple xTEDS

As shown in the xTEDS, there are two actions that can be performed on the temperature sensor: request/cancel data subscription and send a command message. The SPA-1 protocol will be used to request a data subscription as shown below in Table 1. The Interface ID and the Message ID will come directly from the xTEDS and be programmed into the upload scripts. The request message is sent from the CubeLab bus to the SPA device and the response is sent from the SPA device to the CubeLab bus. In this instance, any data received will be logged directly to the SD card or data buffer on the microcontroller (depending on the state of astronaut interaction). If the device designer wants to run a variety of experiments, the upload script architecture allows any combination of subscriptions to data messages. Table 1 also lists the command that will be used to cancel a data subscription. No reply from the sensor is expected.
A similar method will be used to send a command message as seen in Table 2. The request for a command message is sent from the CubeLab bus to the SPA device and a response containing a status byte is sent from the SPA device to the CubeLab bus. Given the nature of the command message shown in the sample xTEDS (ToggleLED), it might be difficult to verify whether the command is actually being executed on the SPA device or not. In this case, GPIO pins will be available on the CubeLab bus which can be connected to appropriate positions on the SPA device to verify if the command is being executed properly. Data from GPIO pins can also be logged to a file.

### Table 1. Commands to request data subscription

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
<th>Payload length</th>
<th>Interface ID</th>
<th>Message ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>M</td>
<td>2 (2 bytes)</td>
<td>(1 byte)</td>
<td>(1 byte)</td>
<td>n/a</td>
</tr>
<tr>
<td>Response</td>
<td>D</td>
<td>2 + n (2 bytes)</td>
<td>(1 byte)</td>
<td>(1 byte)</td>
<td>(n bytes)</td>
</tr>
<tr>
<td>Request</td>
<td>C</td>
<td>2 (2 bytes)</td>
<td>(1 byte)</td>
<td>(1 byte)</td>
<td>n/a</td>
</tr>
</tbody>
</table>

As mentioned previously, the upload script architecture will be used to retrieve and store data from the SPA-1 device. This upload script will contain (in byte format) the messages that must be sent to the SPA device to subscribe to notification messages, cancel subscription of notification messages, and execute command messages.

The CubeLab bus is currently under development and has been tested to fully operate on a SPA-1 network as an 8051-based ASIM. The test setup for development of the CubeLab bus on the COSMIAC CubeFlow kit which provides a SPA-1 network can be seen in Figure 7 to the right. The software drivers for a generic payload have been written which include controlling the USB-to-SD card reader with bypass chip to switch between card reader and pass through modes, SPI drivers for the microcontroller to write data to the SD card on board, drivers for a real-time clock onboard which can be used during data collection and commanding, and file system drivers for the microcontroller to write to the SD card using the FAT16 file system.

![Figure 7. Desktop development of SPA-1 based CubeLab](image)
VI. Conclusion

Through the work described in this paper, the SSL has demonstrated a rapid, inexpensive method for designers of SPA-1 devices to operate their hardware in microgravity. The SPA-1 compatible CubeLab bus allows SPA-1 devices to be flown to the ISS and installed in the NanoRacks Platform to provide quick experimental data return. The structure of the bus allows black box-style data collection from a USB Mass Storage Device when astronauts collect experiment data. The bus structure also prevents loss of experimental data during astronaut data collection. Through the use of upload scripts, a wide variety of tests can be performed on-orbit and can be changed after the SPA-1 device has been launched.

References


