BIRDS-X ICD v1.4

BIRDS-X APRS Payload Competition

Interface Control Document (ICD)

Satellite Interfaces and Configuration

Laboratory of Lean Satellite Enterprises and In Orbit Experiments
Kyushu Institute of Technology
Kitakyushu 804-8550, Japan
https://kyutech-laseine.net/english/

v1.4 21 August 2023

Contents

List of figures

List of Tables

Acronyms

2U CubeSat	2-Unit CubeSat
AF	Audio Frequency
AFSK	Audio Frequency-Shift Keying
API	Application Programming Interface
APRS	Automatic Packet Reporting System
ARDC	Amateur Radio Digital Communications
BIRDS	Joint Global Multi-Nation Birds Satellite project
BPB	Backplane Board
CAD	Computer-Aided Design
COMM	Communication System
COTS	Commercial-Off-The-Shelf
CPLD	Complex Programmable Logic Device
DIO	Digital Input/Output
EM	Engineering Model
EPS	Electrical Power System
FAB	Front Access Board
FM	Flight Model
FM	Flash Memory
FM	Frequency Modulation
GND	Ground
GS	Ground Station
GT	Ground Terminal
ICD	Interface Control Document
ISS	International Space Station
J-SSOD	JEM Small Satellite Orbital Deployer
JAXA	Japan Aerospace Exploration Agency
Kyutech	Kyushu Institute of Technology
MCU	Microcontroller Unit
OBC	On Board Computer
OCP	Overcurrent Protection
PCB	Printed Circuit Board
RAB	Rear Access Board
RF	Radio Frequency
Rx	Reception
RGS	Reference Ground Station
RGT	Reference Ground Terminal
Rx	Receive

SINAD	Signal to Noise and Distortion
SPI	Serial Peripheral Interface
TNC	Terminal Node Controller
TRX	Transceiver
Tx	Transmit
UART	Universal Asynchronous Receiver-Transmitter
UHF	Ultra-High Frequency
VHF	Very High Frequency

Units

bps	Bits Per Second
dBm	Decibel-milliwatt
g	Gram
kHz	Kilohertz
mA	Milliampere
MHz	Megahertz
mm	Millimeter
W	Watt
Who	Watt-Hour
V	Volt
mAh	Milliampere Hour
Ω	Ohm

BIRDS-X Introduction and Scope

BIRDS-X project is a 2U CubeSat (100mm x 100mm x 226.5mm) dedicated to amateur radio communication. Our stakeholders are Kyushu Institute of Technology (Kyutech) and the Association for Radio Digital Communications (ARDC). The goal of our project is to bring diversity to the space sector and democratize the usage of space.

One of the missions in the BIRDS-X satellite project is to select five Automatic Packet Reporting System (APRS) Payload designs through open competition. The objective of the APRS payload competition is increasing the users of amateur radio community, as well as helping people to get involved in the design, development, and operation of that payload, resulting in the improvement of technical skills and democratization of space.

The mission aims to fulfill the objectives linked to APRS communication: creating outreach for the radio amateur community and developing the APRS technology in space to improve its accessibility. By having a public competition, we aim to engage a maximum number of participants with amateur radio communication and at the same time to stimulate the development of communication technology.

The competition will be held in three phases as shown in Figure 1. Qualifying teams from the first phase will progress to development of payload for the next phases until five final APRS payloads are selected to be launched in the BIRDS-X satellite.

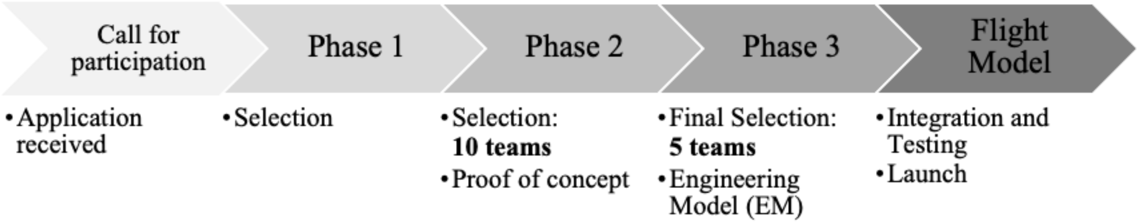
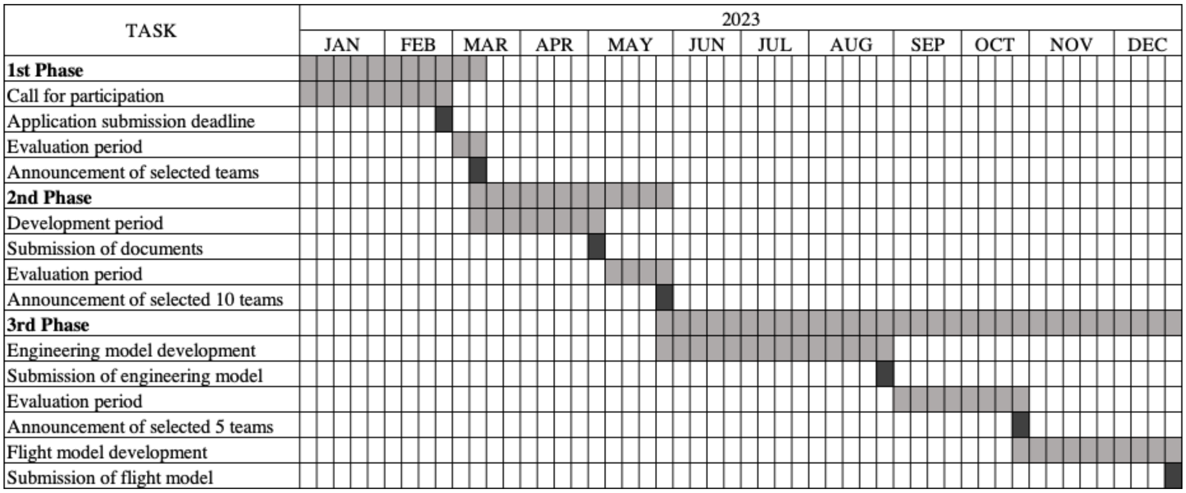


Figure 1. APRS payload competition phases

The timeline for payload development will be according to the schedule shown in Figure 2. At the end of the competition, the selected teams will have to deliver a fully functional APRS payload board. The payload must comply with all the requirements mentioned in this Interface Control Document (ICD) that will allow easy integration of the payload to the satellite.



TASK	2023																																																			
	JAN					FEB				MAR				APR				MAY					JUN				JUL				AUG					SEP				OCT				NOV					DEC			
1st Phase	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨																																									
Call for participation	▨	▨	▨	▨	▨	▨	▨	▨	▨																																											
Application submission deadline									■																																											
Evaluation period										▨	▨																																									
Announcement of selected teams											■																																									
2nd Phase											▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨																														
Development period											▨	▨	▨	▨	▨	▨	▨	▨																																		
Submission of documents																		■																																		
Evaluation period																			▨	▨	▨	▨																														
Announcement of selected 10 teams																						■																														
3rd Phase																						▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Engineering model development																						▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨																	
Submission of engineering model																																			■																	
Evaluation period																																				▨	▨	▨	▨	▨	▨	▨	▨									
Announcement of selected 5 teams																																											■									
Flight model development																																											▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Submission of flight model																																																				■

Figure 2 Schedule for APRS Payload Competition

Related Documents

1. BIRDS Platform ICD, Rev. F, March 2021.
2. JEM Payload Accommodation Handbook, Vol.8 Rev. D1, July 2020.
3. CubeSat bus interface with Complex Programmable Logic Device, Acta Astronautica, 160(2019), 331–342.
4. APRS Protocol Reference version 1.0, APRS Working Group. Tucson Amateur Packet Radio Corp, August 2000.
5. APRS Protocol Reference 1.0
6. APRS SPEC 1.1 addendum
7. APRS SPEC 1.2 addendum

BIRDS-X Platform Outline

BIRDS-X satellite architecture is divided into two sections: the BUS and the mission payloads. The BUS is comprised of the:

1. On-Board Computer (OBC/EPS)
2. Battery Box
3. Front Access Board (FAB)
4. Rear Access Board (RAB)
5. Communication Board (UHF Transceiver)
6. Back Plane Board (BPB)
7. Antenna Panel Board

Figure 3 shows the internal configuration and layout of the BIRDS-X satellite platform.

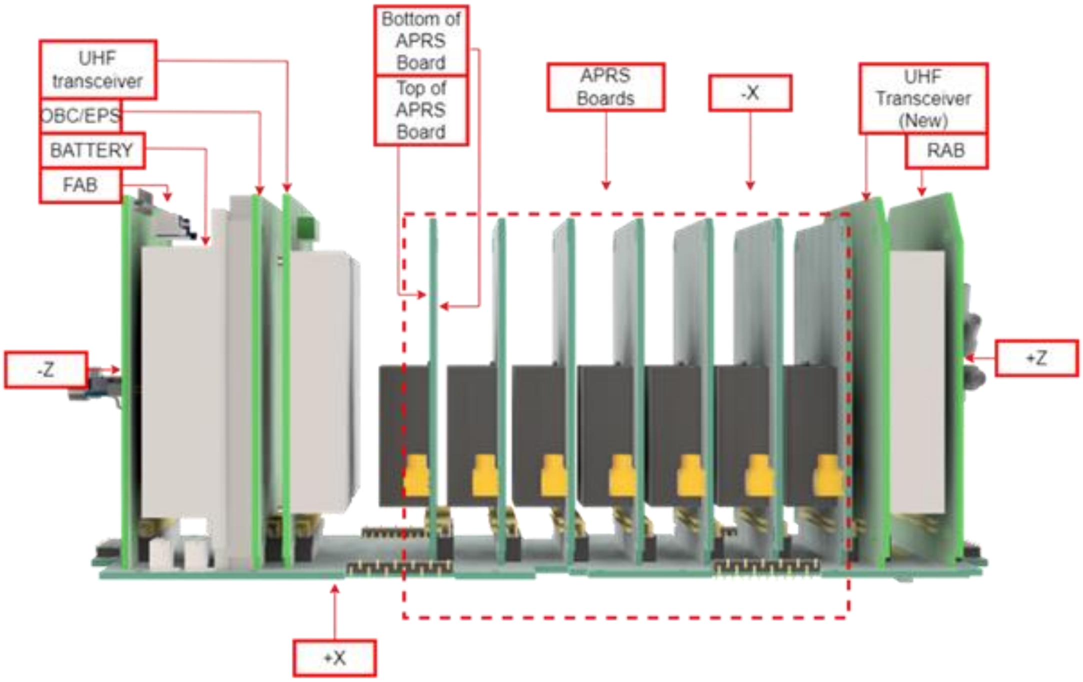


Figure 3 Internal configuration of the satellite

The platform uses a backplane to provide electrical and mechanical interface to the internal boards. The backplane consists of 50-pin female connectors (A3C-50DA-2DSA) for BUS system and payloads to be plugged to the backplane for integration.

BIRDS-X System Diagram

Figure 4 shows the system diagram of the whole BIRDS-X satellite.

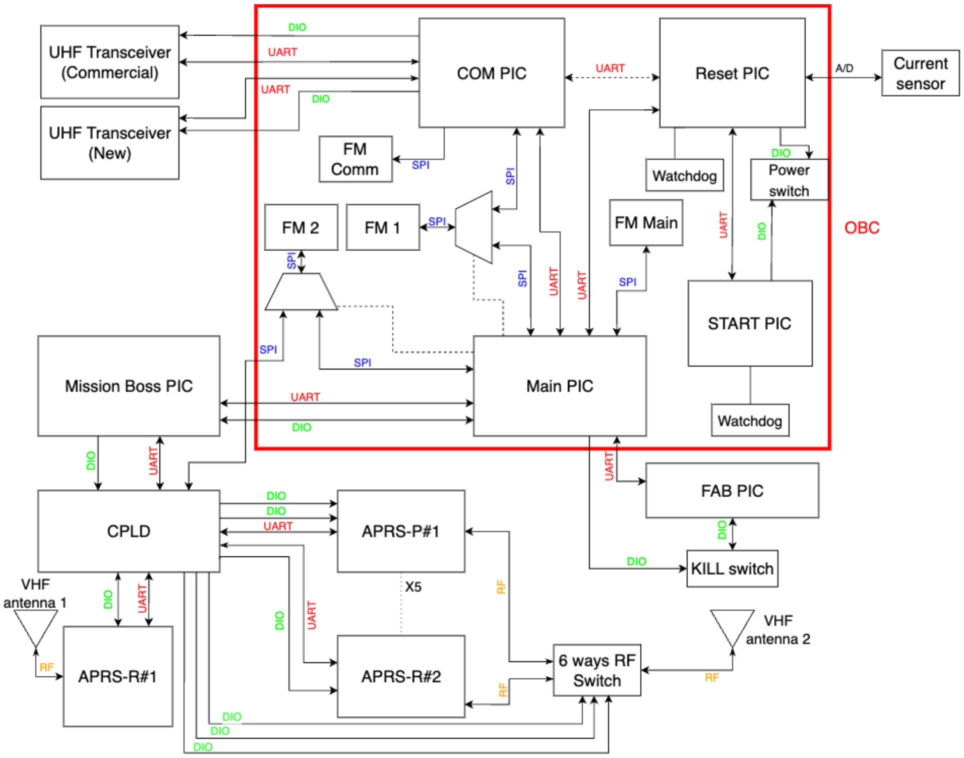


Figure 4 System Diagram of the satellite

BIRDS-X Satellite Dimension

The dimensions of the BIRDS-X satellite are detailed in Table 1.

Table 1 Satellite dimensions

Size	Satellite dimensions [mm]	Rail dimension [mm]	Tolerance [mm]
2U	(X, Y, Z): (100, 100, 227)	8.5 x 8.5	±0.1

The dimension of the platform conforms to CubeSat standard. The ± Z rail ends, and the edges of the rail sides are rounded at 1mm radius of curvature. Also, the rail surface conforms to MIL-A-8625 Type3; hard anodized coating has been made at 10 µm or more. The coordinate system of the BIRDS-X platform conforms to the CubeSat deployment system (J- SSOD-R, E-SSOD, etc.) and defined as follows:

1. The coordinate system of the platform: (X, Y, Z)
2. The origin is at the geometric center in the nominal dimensions of the platform.

Figure 5 to Figure 7, show the CAD images of the CubeSat along with the coordinate system.

Figure 5 shows the bottom part of the satellite, on this side the deployment switches are located at the end of the rails, at the same time on the -Z external panel are located some cuts for the acces ports of the FAB board, which differentiates it from the top side that doesn't require this access ports and deployment switches.

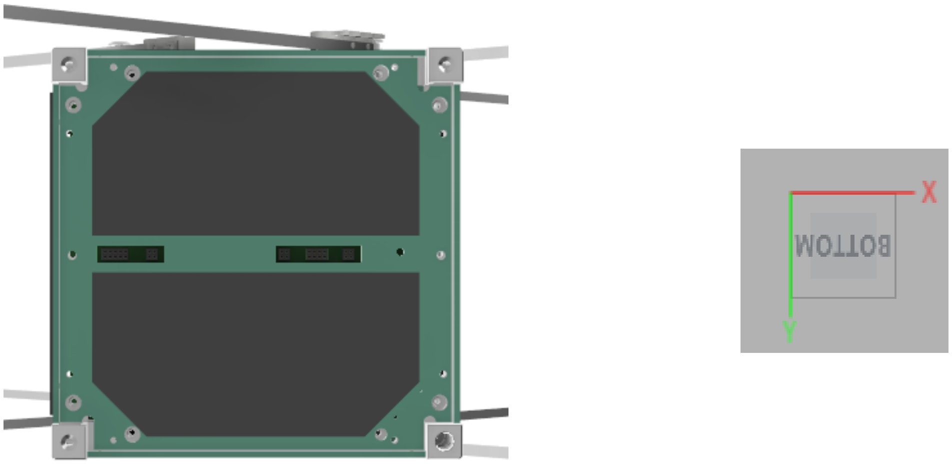


Figure 5 Bottom view of the satellite

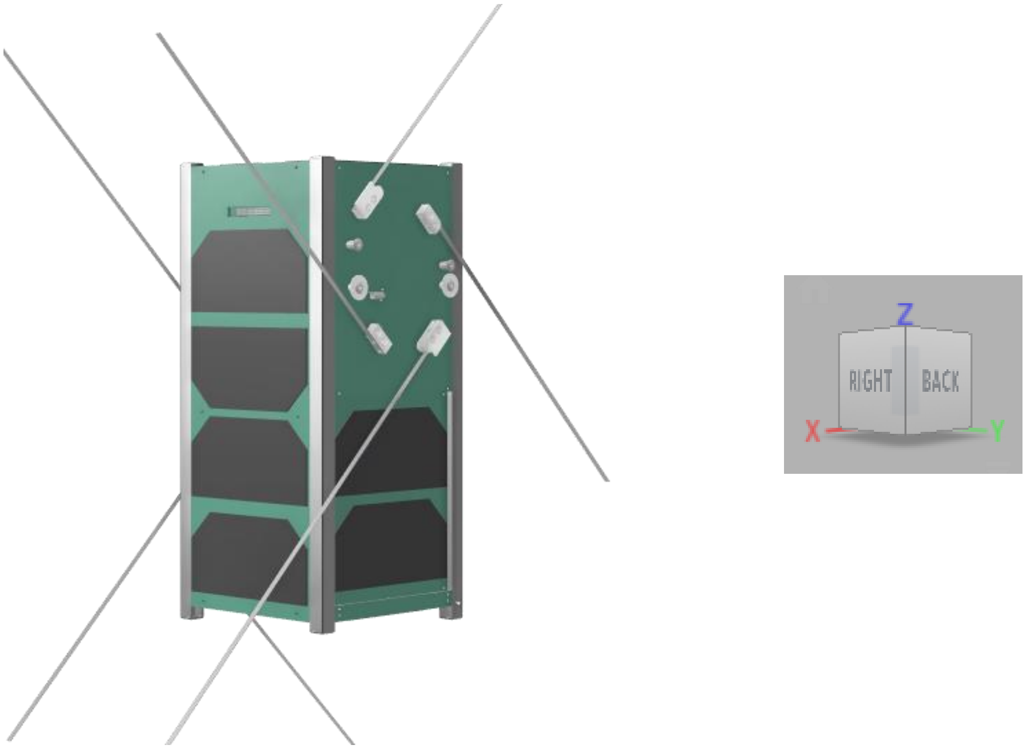


Figure 6 External view of the satellite

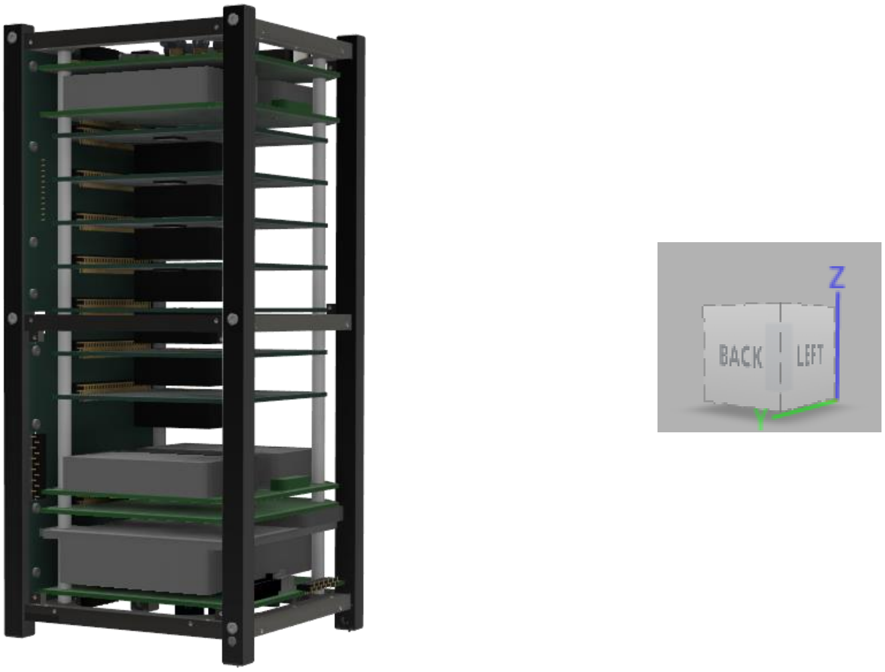


Figure 7 Internal configuration of the satellite

Payload Requirements

Mechanical Interface

The APRS payload must be designed with the dimensions detailed in Figure 8 to Figure 10. The dimensions of the APRS payload board must be 79 mm x 81 mm and must use a 50-pin male connector (NRPN252PARN-RC, LPC- 50M2LG, or equivalent) for connection with the backplane. In addition, the platform secures each board with four rods. Therefore, 3.8mm diameter holes must be included at four corners of the board for passing the long rods.

The maximum thickness of the payload board must not exceed 12mm on the opposite side of the 50-pin connector, excluding the board thickness (1.6mm). The total thickness of the PCB board should be 13.6 mm. The sizes and positions of the boards and the position of the 50-pin connector are detailed in Figure 8. The antenna connector must be right angle MCX connector (60611002111502 from Würth Elektronik, or equivalent).

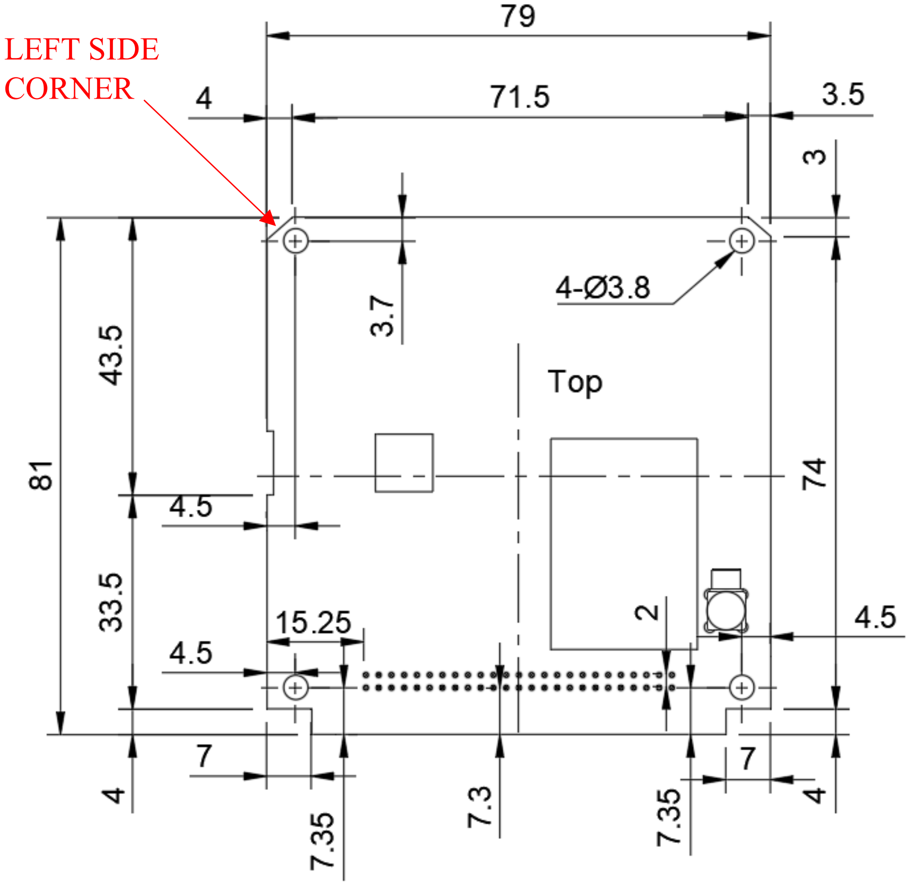


Figure 8 APRS payload board size requirements: top view

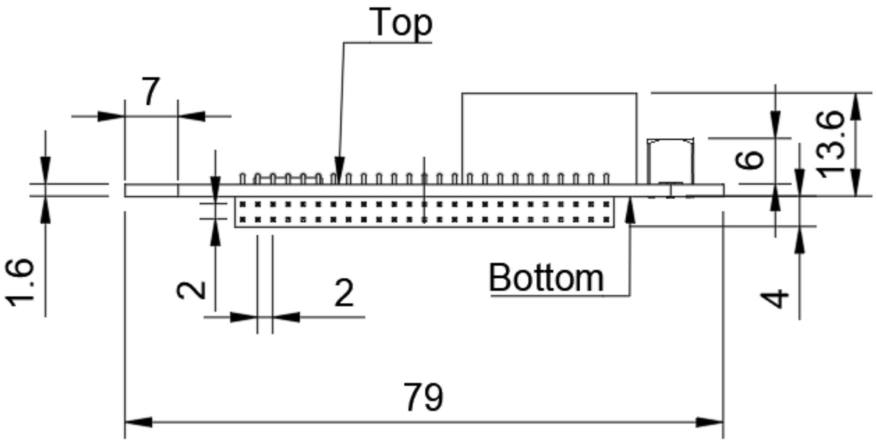


Figure 9 APRS payload board size requirements: front view

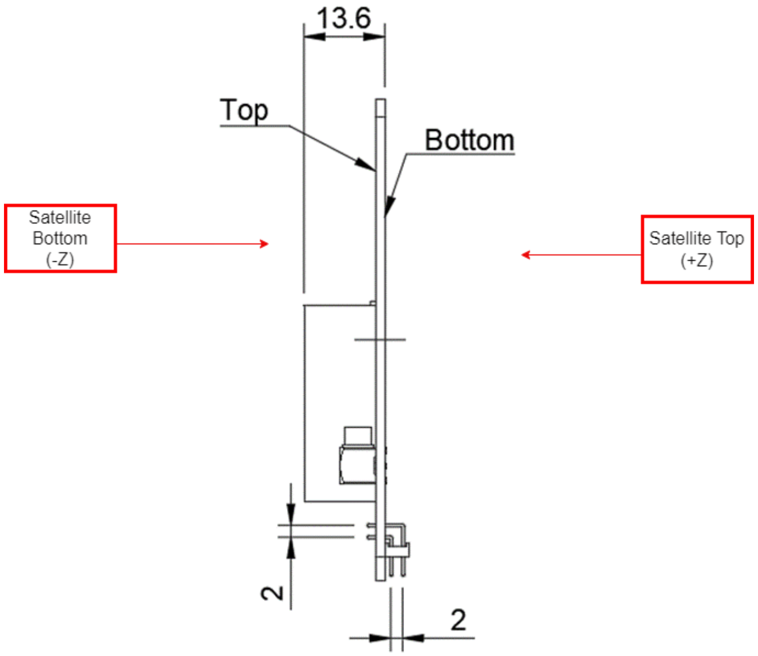


Figure 10 APRS payload board size requirements: side view

All the components must be placed on the top side of the PCB as Figure 8, the side which has the larger cut at the top left corner, and a cut marked on the left side only for reference. The 50-pin male connector must be soldered on the bottom side of the PCB. The maximum allowed mass of the APRS payload will be 90g. The pin assignment of the 50-pin connector is detailed in the next sub-section.

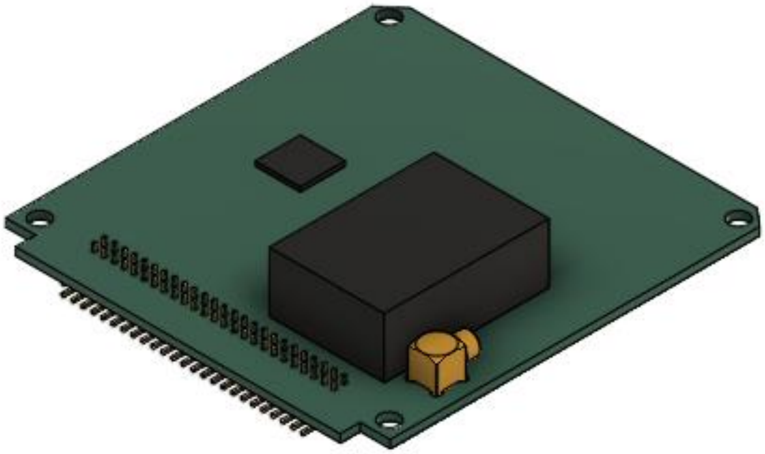


Figure 11 Isometric view of the payload

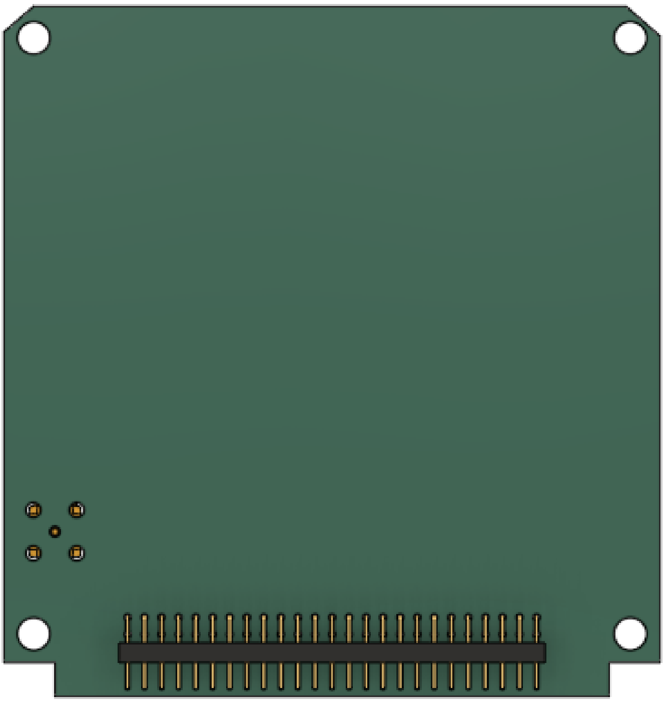


Figure 12 Bottom view of the payload

The dimensions and mass are subject to minor changes, for any changes made, it will be notified and will be included in the next versions of the ICD. Please check the website for the latest version of the ICD.

Electrical Interface

1. Pin Assignment

The pin assignment for 50-pin connector (NRPN252PARN-RC, LPC- 50M2LG, or equivalent) on the APRS payload board must follow the assignment as shown in Table 2.

Table 2 Pin assignment for the APRS payload board

Signal Name	Pin Number		Signal Name
Programming/debug #2	2	1	Programming/debug #1
	4	3	Programming/debug #3
	6	5	
	8	7	
	10	9	
	12	11	
GND_SYS	14	13	GND_SYS
SUP_5V0	16	15	SUP_5V0
UART (APRS to Mission Boss)	18	17	UART (Mission Boss to APRS)
DI/O_2 (5V0 OCP control)	20	19	DI/O_1 (3V3 OCP control)
	22	21	
SUP_UNREG_1	24	23	SUP_UNREG_1
SUP_3V3_2	26	25	SUP_3V3_2
	28	27	
	30	29	
	32	31	
	34	33	
SUP_UNREG_2	36	35	SUP_UNREG_2
	38	37	
	40	39	
	42	41	
	44	43	
	46	45	
	48	47	
SUP_3V3_1	50	49	SUP_3V3_1

The legend for connection type is as shown below:

🟫 Digital connection (fixed)
🟨 Digital connection (configurable through CPLD)
🟦 Analog connection
⬜ Power connection

Pin number 1 on the payload board is as marked in the following Figure 13. To avoid any confusion, the cuts on the top corners are made asymmetrical. Pin number 1 is the upper first pin from the left on the 50-pin connector when the bigger cut corner is on the top left-hand side. **Double or triple check shall be made so that the pin assignment is correct and the 50pin connector is attached to the correct side of PCB.**

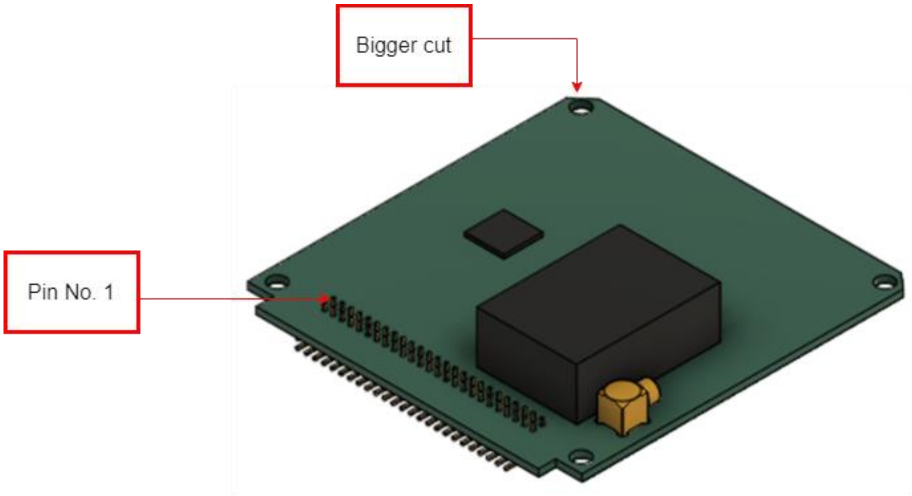


Figure 13 Pin number 1 assignment

2. Power Lines

The payload can use two power lines from the BUS system; 3.3V line (pin number 25 and 26) and 5V line (pin 15 and 16) from 50-pin connector to power the board. Before tapping the power lines to power the APRS payload, a switch preferably an overcurrent protection (OCP) with an enable pin for example LTC4361 or equivalent must be implemented for each power line as shown in Figure 14. The enable line of OCP for 3.3V line must be routed to pin number 19 and for 5V line must be routed to pin number 20 on the 50-pin connector. This allows the BUS to control the operation of the APRS payload on-orbit. **Any part of the payload MUST NOT operate before the implementation of OCP.** If any team decides to use either 3.3V or 5V then only one switch can be implemented on the board following the pin assignment for each kind of enable line.

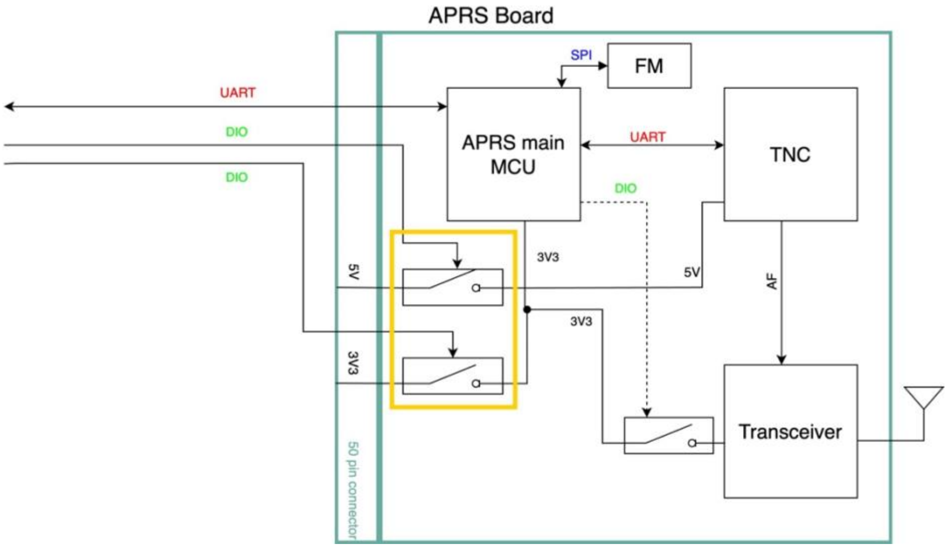


Figure 14 Implementation of switch for power tapping

The power consumption of the APRS payload design must not exceed **300mW** during receive mode and **1750mW** during transmit mode.

Data Transfer Interface

The APRS payload will communicate and transfer mission data to Mission Boss using UART protocol at a baud rate of 9600bps (8 bits-no parity-1 stop bit). Pin number 17 and 18 of the 50-pin connector are assigned for the UART communication; Pin 17 for Rx (Mission Boss to APRS) and Pin 18 for Tx (APRS to Mission Boss). Pin numbers 1-3 are assigned for programming of the main microcontroller unit (MCU) through the external access ports once the satellite is assembled.

Figure 14 also shows a typical example of APRS payload board. The APRS payload must be a modular system; it must be able to decode the APRS packets received from ground terminals in AX.25 protocol and store the received packets in its own flash memory during mission execution. The stored mission data must be transferred to the Mission Boss only through UART communication. After the satellite is assembled, the call sign for the Terminal Node Controller (TNC) must be able to be updated by using access ports which also are used to reprogram the APRS MCU.

APRS Mission

The payload shall work in two mission modes:

- Digipeater: RF signals containing APRS messages from amateur radio users or ground terminals shall be received by the transceiver, decoded by the TNC, and sent back to Earth immediately. These messages are not stored on the flash memory.

- Store and Forward: RF signals containing APRS messages are decoded by the TNC, sent back to Earth and at the same time are saved on the flash memory. The stored data packets will be transferred to the mission boss and later downloaded to the ground station for analysis.

Participants could add sensors to the payload design if it complies with the power budget, mass, and volume requirements. The data collected from the sensors can be communicated to the mission boss only through the UART communication.

Digipeating

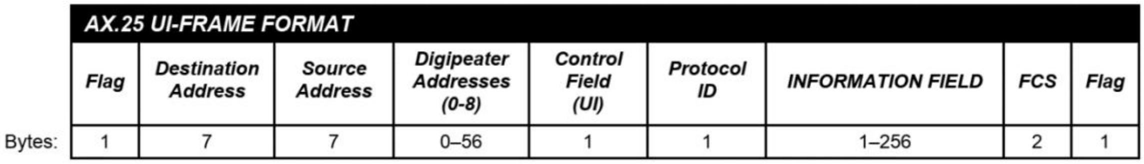
Communications between the APRS payload users from the amateur radio community are performed by relaying the RF signals with APRS packets incoming to the satellite in a process called “digipeating”. A combination of a Terminal Node Controller (TNC) to handle the APRS packets in AX.25 protocol and an RF transceiver is required.

1. Terminal Node Controller

The process of disassembly and assembly of APRS packets using the AX.25 data layer protocol is made by a TNC, this device is responsible for providing the data layer according to the AX.25 specifications and converting the digital data to audio tones by AFSK modulation to send/receive APRS packets at a rate of 1200 bit/s.

TNC implementation on the APRS payloads can be by hardware or software, in any case, it shall be able to perform the encoding/decoding process of APRS packets according to the APRS Protocol Reference 1.0 and its addendums APRS SPEC 1.1 and APRS SPEC 1.2.

An example of AX.25 UI (Unnumbered Information) frame format extracted from APRS Protocol Reference 1.0 is shown below:



AX.25 UI-FRAME FORMAT									
	Flag	Destination Address	Source Address	Digipeater Addresses (0-8)	Control Field (UI)	Protocol ID	INFORMATION FIELD	FCS	Flag
Bytes:	1	7	7	0–56	1	1	1–256	2	1

Figure 15 Example of AX.25 UI (Unnumbered Information) frame format

TNC shall provide the destination address in the AX.25 UI frame for the APRS device identifier. This identifier provides a unique number for the hardware manufacturer or software developer and the version.

Example:	APK004	for Kenwood TH-D74 Transceiver
	APTTxx	for Byonics Tiny Track
	APDWxx	DireWolf TNC software
	APZxxx	Experimental (requires notification before use)

This process is performed by software and hardware TNC available. Still, in case of a new development of TNC implementation for APRS different from those available nowadays, it is suggested to initiate the request for a new identifier. For more details about creating an APRS device ID, refers to (https://github.com/aprsorg/aprs-deviceid/issues).

2. TNC Settings

Since many TNC implementations exist, the following table represents the required parameters on the TNC settings for managing APRS packets, except for the parameters MYCALL and BTEXT, INFO and presented as an example. Besides the parameters presented in this section, all other TNC settings available shall be provided by the competition participants, and they must be available to be set during ground tests before satellite integration.

Table 3 Example of TNC parameters

Parameters	Values	Units
MYCALL	BIRDSX	
PATH	WIDE1-1	
DWAIT, QUIET	0	ms
PERSIST	63	
BTEXT, INFO	“Hello World”	
SLOTIME	15	ms
TXDELAY, TXD	30	ms
DUPETIME, DEDUPE	30	s

MYCALL: Satellite callsign.

PATH: Route that the APRS packet must follow before it expires.

DWAIT, QUIET: Time required before resuming transmissions without carrier signal.

PERSIST: Value is required in conjunction with SLOTIME to start transmission when the carrier signal is undetected.

BTEXT, INFO: Text transmitted as beacon signal by the APRS payload.

SLOTIME: Time required after a random number is generated before resuming transmission and compares it with PERSIST value.

TXDELAY, TXD: Time between radio lines is activated, and the audio signal is sent to the transceiver.

DUPETIME: The time of a previously digipeated packet is compared with the incoming packets to avoid duplicate transmission.

3. Beacon Message

While the APRS payload is active, a beacon signal with an APRS packet is required every 60 seconds. Beacon timing settings, as other parameters mentioned above, must be able to be set for testing purposes on the ground before satellite integration. APRS beacon data formats available refer to APRS Protocol Reference 1.0 Appendix 1.

4. RF transceivers

The table below shows the minimum specifications required for RF transceivers in the APRS payload.

Transmitter	
Central frequency	145.825 MHz
Frequency deviation (max)	± 3 kHz
Power output	≤ 33 dBm
Frequency accuracy	± 3 kHz
Bandwidth (max)	12.5 kHz
Spurious emissions	FCC part 97
Receiver	
Sensitivity @ 12dB SINAD	-120 dBm or better
Antenna impedance	50 Ω
Duty cycle	20% Tx; 80% Rx

Table 4 Minimum requirements for RF transceivers

Activating the transceiver Push-To-Talk (PTT) lines must be performed by appropriate circuitry; any other way, like activation by using Vox-On-Transmission (VOX), is forbidden.

It is mandatory to payloads to provide adequate matching circuits for 50 Ω antenna output.

Store & Forward

During APRS digipeating operations, payload must perform data capture from incoming APRS packets when required. The process for data storage operation shall be activated by commands received from satellite's Mission Boss. It is mandatory for competition participants to provide commands to perform this operation as well as commands for data transfer to Mission Boss using the physical connections and settings mentioned in Data Transfer Interface section. Details about command format will be described on Command for All Payload section.

Memory requirements

It is required for payloads to allocate memory storage for a 24/7 continuous operation and provide the measures to prevent memory shortage during APRS packet capture, as well other payload missions. APRS packet saved on the payload memory must be decoded packets from TNC and shall not be mixed with other payload missions data when is transferred to the satellite by commands from the Mission Boss.

Payload Commands

1. Command format

The command for the Dragonfly satellite is composed of 15 bytes as shown in Table 5. The role of each byte is shown in the following.

Table 5 Command format for Dragonfly satellite

Header	Command Format	Command ID	Reservation Time	Detailed Command	Command Header	Mission Command Data							Command Footer	CRC	
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F

- **Header (1 byte)** - Leading byte for the command signal. Set by the GS Software.
- **Command Format (1 byte)** - Determines which PIC the command is designated.
- **Command ID (1 byte)** - Determines the destination of the command.
- **Reservation Time (1 byte)** - Designates the reservation time for specific missions. Delays command will be executed for the specified time.
- **Detailed Command (1 byte) -** Determines which command will be executed. The upper 4 bits show the mission mode, and the lower 4 bits show each command's contents.
- **Command Header (1 byte)** – Encapsules the command as a header. This byte should be 0xE0.
- **Mission Command Data (7 bytes)** - Describes the command data to be included.
- **Command Footer (1 byte)** – Encapsules the command as a footer. This byte should be 0xED.
- **CRC (2 bytes)** - Cyclic Redundancy Check. For error checking.

Command ID, **Reservation Time**, **Command Header**, **Detailed Command**, **Command Footer**, and **Mission Command Data** will be sent to Mission Boss (11 bytes in total). And **Command Header, Mission Command Data, and Command footer** will be sent to each Mission MCU (**9** bytes). Moreover, you can receive **Detailed Command** (1 byte) if you require that information.

2. General requirement

The competitors need to define all the commands to operate their payload. Here are several terms which are needed to follow.

Table 6 Command Array

Mission Command Data						
No.0	No.1	No.2	No.3	No.4	No.5	No.6

- Commands should form the array of **hex**.
- Commands must be composed by **7 bytes**.
- The command needs to start from **0xE0** as a header of commands.
- The command needs to finish with **0xED** as a footer of commands.
- Header and footer must not be included in the Mission Command Data.
- APRS MCU will receive 9 bytes in total including header and footer.

3. Example of Command List and API

Here are some examples of the commands. In the API, you must mention which board component will perform what action and what happens when using that command. If the command asks the satellite bus to do some tasks, you must be clear with it.

Table 7 Command Example

Number	Command header	Mission Command Data							Command footer	Description
		No.0	No.1	No.2	No.3	No.4	No.5	No.6		
1	0xE0	0xA0	0x00	0x00	0x00	0x00	0x00	0x00	0xED	Turn off the sensor1
2	0xE0	0xAA	0x00	0x00	0x00	0x00	0x00	0x00	0xED	Turn on the sensor1
3	0xE0	0xCC	T [2 bytes]		0x00	0x00	0x00	0x00	0xED	Switch to waiting mode and stay T sec
4	0xE0	0xEE	Adress [4 bytes]				0x00	0x00	0xED	Clear flash memory data

Description

1. Turn off the sensor1:
   This command is used to turn off the temperature sensor on the board. Once the MCU receives this command, it immediately turns off the enable pin and shuts off.

2. Turn on the sensor1:
   This command is used to turn on the temperature sensor on the board. Once the MCU receives this command, it immediately turns on the enable pin and activates it.

3. Switch to waiting mode:
   When the MCU receives this command, it stops its mission mode and shifts to waiting mode. In this mode, the MCU will keep listening to the command from Mission Boss. But after T sec is passed without any command from Mission Boss, the MCU will start working in digipeating mode.
   The third byte (No.2) and fourth (No.3) are used to express the waiting time. These 2 bytes show the four digits number in hex. (e.g., E0, CC, 01, 2C, 00, 00, ED → wait 012C sec = 300 sec in decimal in waiting mode)

4. Clear flash memory data:
   This command is used to clear the flash memory data. Once the MCU receives this command, it will clear the 1000 bytes of data in flash memory from the designated address.
   4 bytes from the third byte (No.2) to the sixth (No.5) are used to express the address point.

4. Common command requirement

Our satellite has 7 APRS payloads, and there are some common commands. Therefore, we must use common commands for general operations such as the activation or deactivation of the boards. The following table shows all the common commands (6) we require participants to use.

Table 8 Common Command List

Command header								Command footer	Description
	No.0	No.1	No.2	No.3	No.4	No.5	No.6		
0xE0	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xED	Turn off the payload
0xE0	0x0E	0x00	0x00	0x00	0x00	0x00	0x00	0xED	Turn on DP*[1] mode
0xE0	0x01	0x00	0x00	0x00	0x00	0x00	N*[2]	0xED	Real-time downlink on DP mode
0xE0	0x1E	0x00	0x00	0x00	0x00	0x00	0x00	0xED	Turn on S&F*[3] mode
0xE0	0x11	0x00	0x00	0x00	0x00	0x00	N*[2]	0xED	Real-time downlink in S&F mode
0xE0	0x12	0x00	0x00	0x00	0x00	0x00	N*[2]	0xED	Transfer N packets of data to the satellite bus

*[1] Digipeat

*[2] The number of packets to be sent

*[3] Store-and-forward

Note

- The first byte (No.0) distinguishes each command to process. The APRS MCU will be required to recognize what process it needs to do.
- Some commands use a seventh byte (No.6) to designate the number of packets to downlink or transfer. One packet can contain **200 bytes or less** of row data. It also required to encoupsel it with header and footer.

5. Command definition for competitor’s unique mission

As shown in the previous section, 7 bytes of space are set for any command contents in hex. It is required to **define all commands** you need for your unique mission **by following general requirement**.

Evaluation scenario in orbit

After the Dragonfly satellite is released from ISS, we start the payload performance evaluation immediately. There are two test processes to evaluate both abilities including transmitting and receiving.

1. Transmitting performance test

To evaluate only the transmitting performance of the APRS payload, we use a beacon to do it. During the performance test period, BIRDS-X will continuously listen to the beacon coming from the APRS board. At the same time, the team will record the elevation angle data we can get from GS. After the satellite goes out of the communication range, we will analyze the result and evaluate it. The test flow is shown next.

I. Send command from GS to turn on APRS board in digipeating mode.
II. The satellite turns on the APRS board in digipeating mode.
III. APRS board transmits a beacon periodically.
IV. GS receives the beacon and records the elevation angle.
V. After a specific time, send a command from GS to turn off the APRS board.
VI. The satellite turns off the APRS board.
VII. Analyze the result.

2. Receiving performance test

To evaluate receiving performance of the APRS payload, we use digipeating to do it. During the performance test period, we continuously uplink APRS packets and listen to the digipeated packets. At the same time, we record the elevation angle data we can get from GS.

After the satellite goes out of the communication range, we will analyze the result and evaluate it. It is shown the test flow in the following sentence.

I. Send command from GS to turn on APRS board in digipeating mode.
II. The satellite turns on the APRS board in digipeating mode.
III. The GS transmits APRS packets periodically.
IV. The APRS board receives it and digipeats.
V. GS receives the digipeated packets and records the elevation angle.
VI. After a specific time, send a command from GS to turn off the APRS board.
VII. The satellite turns off the APRS board.
VIII. Analyze the result.

Notes

1. Each APRS payload must be able to survive in a space environment. Although the space environment tests will be performed at Kyutech, the components selected for the payload board must be able to operate within the temperature range of -15°C to +55°C.
2. Each APRS payload shall do no harm to the rest of the satellite and its components, such as other APRS payloads and the satellite bus system, during its nominal operation and even during faulty conditions.
3. Each APRS payload must be delivered with an Application Programming Interface (API). The team is also required to develop software for the designed payload. The commands for activating the mission will be provided in detail in the next version of the ICD. The payload will not be considered as “delivered” without a working software.
4. Functional test reports and test environment must be submitted for the designed payload.
5. The selected team for FM submission will have the opportunity to use the payload after the execution of the on-orbit competition, this will be subjected to coordination with the BIRDS-X team in advance.

IMPORTANT

The ICD will be updated as needed throughout the period leading up to the selection of five payload boards. The up-to-date version will be maintained in the BIRDS-X Project Website (https://birds-x.birds-project.com). All participants are responsible for working with the correct version by checking the revision date.

Laboratory of Lean Satellite Enterprises and In Orbit Experiments Kyushu Institute of Technology

Kitakyushu 804-8550, Japan

https://kyutech-laseine.net/english

BIRDS-X Satellite Project - KyuTech

birdsx_satellite_project

BIRDS-X Satellite Project