

Custom ThinSat Embedded Computer System Design

Dr. Kathryn Wingate, Lara Buri, Lauren De Moudt, Cole Kenny, Manuel Lindo, Matthew McCallum, Grant Novota, Lucas Perry, Austin Scheck, Jake Schroeder, Katie Steward, Srikanth Venkataraman, Trevor Weschler
University of Colorado Aerospace Engineering Sciences, Boulder, Colorado, 80303

Expenditures to access space are slowly coming down; however, development, testing, launch, and operating costs of spacecraft and small satellites still remain a barrier to entry for the spaceflight industry. Enter the ThinSat: a 1/7U pico-satellite developed by Virginia Space with the goal of providing a low-cost, lightweight payload platform for space missions. The Thinsat Operational Management of Commands And Telemetry (TOMCAT) team's objective is to outfit a ThinSat with an on-board communication system that receives commands and transmits data from the payload to the ground. With the payload area being only half of the ThinSat bus, it becomes difficult to incorporate all of the necessary hardware for system operation. Current COTS space flight boards are either too expensive or will not fit on the ThinSat bus. Therefore, a custom printed circuit board was designed and manufactured to conform to the shape of the payload space while completing all mission tasks. This board hosts all flight software, supports the payload (a 9-axis inertial measurement unit), interfaces with a software defined radio, and distributes power across necessary payload systems. One of the main challenges outlined in this paper is integrating the BeagleBone Black microcontroller with the custom board such that the custom PCB can perform the specific function of operating the ThinSat. Also outlined are the test procedures for the custom board, including observing the indicators for a successful flash of the Linux kernel onto the board and conducting software defined tests that check if the microcontroller is interfacing with the payload correctly. As a whole, this paper will outline the design, manufacturing, testing, and integration of a custom printed circuit board for the ThinSat payload space.

I. Nomenclature

<i>TOMCAT</i>	=	Thinsat Operational Management of Commands And Telemetry
<i>PCB</i>	=	Printed Circuit Board
<i>COTS</i>	=	Commercial Off The Shelf
<i>MCU</i>	=	Micro-Controller Unit
<i>IMU</i>	=	Inertial Measurement Unit
<i>SDR</i>	=	Software Defined Radio
<i>eMMC</i>	=	Embedded Multi-Media Card
<i>EEPROM</i>	=	Electrically Erasable Programmable Read-Only Memory
<i>MEMS</i>	=	Micro-Electro-Mechanical System
<i>DDR</i>	=	Double Data Rate
<i>CCSDS</i>	=	Consultative Committee for Space Data Systems
<i>API</i>	=	Application Programming Interface
<i>FSW</i>	=	Flight Software
<i>USB</i>	=	Universal Serial Bus
<i>SMD</i>	=	Surface Mount Device
<i>I/O</i>	=	Input/Output
<i>LED</i>	=	Light Emitting Diode
<i>JTAG</i>	=	Joint Test Action Group
<i>PC</i>	=	Personal Computer
<i>OS</i>	=	Operating System
<i>OSD3358C – Sip</i>	=	Octavo Systems Complete System-in-Package Surface Mount Device

II. Introduction

THE Thinsat Operational Management of Commands And Telemetry project, or TOMCAT, is a senior design project at the University of Colorado Boulder sponsored by Raytheon and managed by a team of 12 students in the Aerospace Engineering Sciences and Electrical and Computer Engineering Departments. The aim of the project is to develop and test a small communications system within the confines of a ThinSat bus [1]. While the focus is on the communication system, which utilizes a software defined radio (SDR), the team also designed the power system, science payload, flight computer, and flight and ground software systems for the small satellite.

This paper will detail the design of the TOMCAT Payload Board, which integrates the power distribution system, the science sensor, and the flight computer that hosts the flight software and signal processing software. The power distribution system takes power from the payload battery, a 2S2P custom Li-Ion pack made of 18500 cells, and regulates and distributes power to other satellite subsystems. The science sensor is an Xsens MTi-3 AHRS inertial measurement unit feeding 3-axis free acceleration and temperature data (though additional data points are available, TOMCAT does not utilize them in order to reduce communication and computing requirements).

The flight computer is based on the Industrial version of the BeagleBone Black, a consumer-grade, open-source microprocessor similar to a Raspberry Pi. This allows the TOMCAT satellite to run desktop versions of Linux operating systems, leading to simplified flight software development. Further, the BeagleBone Black allows for high powered computing in a small form factor.

The motivation for the Payload Board is to integrate all of these systems in a single package. Additionally, while the consumer BeagleBone Black is small, the development board it's integrated on is still too large to fit within the ThinSat payload area. The TOMCAT Payload Board takes only the necessary parts of the BeagleBone Black and integrates them to the custom board which fits within the payload area.

III. Design Objectives

A. Project Overview

Raytheon has tasked the TOMCAT team to outfit a ThinSat with a custom communications system designed and built by the team that can send telemetry from a science payload to the ground and can receive commands from the ground. The TOMCAT project will provide a platform by which Raytheon can test communication with its ground stations. The TOMCAT ThinSat must be able to generate telemetry that is processed by an on-board computer and packetized into CCSDS telemetry packets before being sent to the ground. The team will also develop a ground station that can receive the CCSDS formatted telemetry packets and send commands to the ThinSat. Both the ground station and the ThinSat will utilize software defined radios as the primary communication hardware. To support the communications system, the team must also design subsystems typical of any spacecraft, including power, structures, and ground and flight software.

B. Project Objectives

In order to verify the communications link between the TOMCAT ThinSat and the TOMCAT ground station, three levels of success have been established. Level 1 success is defined by the ability to establish wired communication between two SDRs and send and receive CCSDS formatted packets. Level 2 success requires the wireless communication of the SDRs with a fully integrated system. This means that ground and flight software handle the packetization of commands and telemetry and science data from the payload can be processed by the on-board embedded computer system. Further, the system must be powered by batteries and a power regulation system must be in place to distribute power to all necessary components. Level 3 success is defined by establishing long-range communication with the ThinSat throughout a high-altitude balloon flight. Like Level 2 success, Level 3 success requires the ThinSat and all its subsystems to be fully operational to support the two-way communication of the SDRs throughout the flight test. To meet Levels 2 and 3, the payload board must be able to house the flight software and be fully integrated with the other subsystems on the ThinSat. The payload board is, therefore, an integral part of the success of TOMCAT's mission.

A more detailed visual of the TOMCAT's end goal balloon flight is shown in the CONOPS below. TOMCAT will be launching on along on a balloon string hosted by the University of Colorado at Boulder's Gateway to Space class. The ground station will start communicating with the ThinSat while it is on the ground so a two-way communication link can be verified. On-board the ThinSat, the ground commands will travel from the software defined radio to the microcontroller unit (MCU) to the inertial measurement unit (IMU), which will execute the command and send rate telemetry in return. Throughout the duration of the launch, the ground system will be in constant communication with

the ThinSat. At the peak altitude of the flight (around 30km), the balloon will automatically burst due to pressure, and a parachute will automatically deploy. During the descent, the ground station will continue to maintain two-way communication.

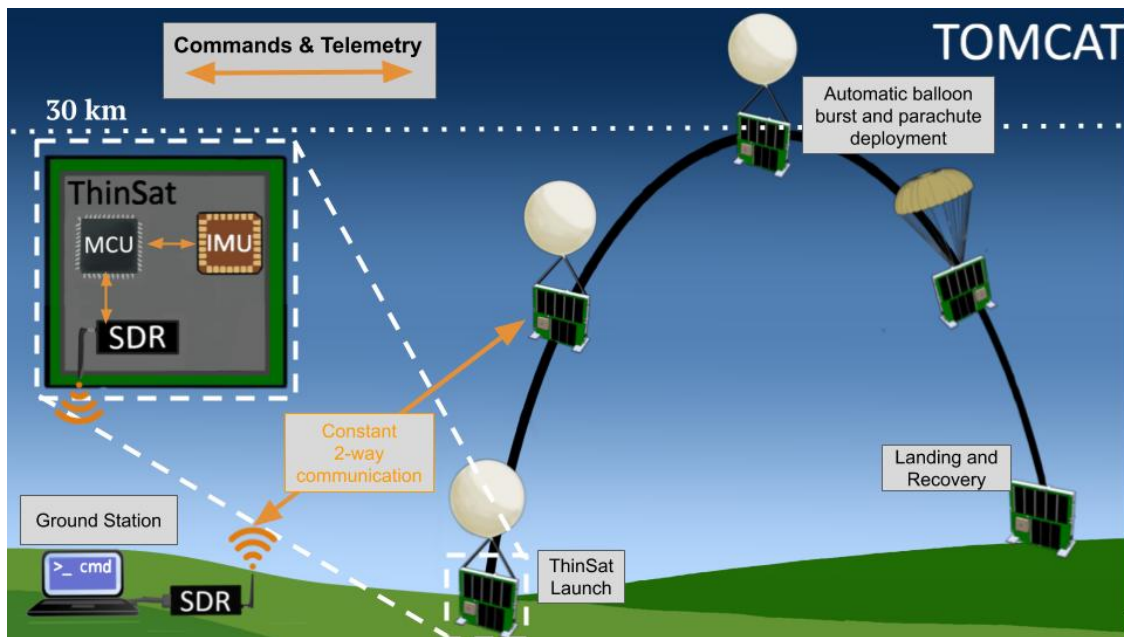


Fig. 1 TOMCAT Mission Concept of Operations

C. Payload Board Requirements

1. Payload Board Size and Shape

In order to conform to the unique shape and size of the ThinSat frame, the designed payload board is required to fit inside the ThinSat payload area as defined by the ThinSat ICD[2]. This is shown in Figure 2. This means the board must have dimensions of 108mm x 52mm and fit into the shape given in the student payload space of the ThinSat structure, shown in green in Figure 2. In order to maintain an acceptable mass for the high-altitude balloon launch and to conform to the ThinSat ICD, the designed payload board is required to have a mass of less than 55 grams.

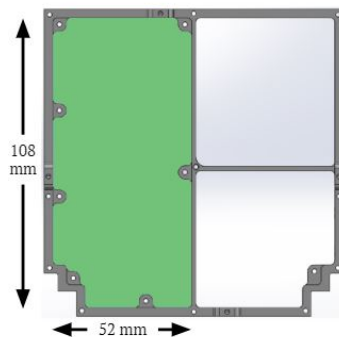


Fig. 2 Dimensions of the Payload Space

2. Payload Computer Software Requirements

The microcontroller unit, SDR software, and flight software (FSW) are required to perform all necessary software tasks for the balloon flight. This means that the on-board computer must receive raw telemetry from the science payload and store that telemetry, including IMU telemetry, payload health and safety telemetry, and command counts. Storing all telemetry allows the team to validate stored telemetry data with down-linked data from the flight test.

Per Raytheon's request, the team is required to run the software system KubOS for all flight software activities. The payload computer must be able to run this program and all other software needs in order for project success.

3. Debugging Requirements

The payload board must be designed with debugging mechanisms that allow for the detection of faulty components. This will allow the team to test the board for issues and ensures more reliable manufacturing.

4. Power Distribution Requirements

The payload board must house the necessary components to regulate power from the battery. This ensures that each component on the Thinsat receives the correct amount of power. Power must be regulated and sent to the IMU, communications hardware, and be used to power the payload board microcontroller.

IV. MCU and FSW Selection

A. MCU

To meet the team's objective of outfitting a Thinsat with an on-board communication system that receives commands and transmits data from the payload to the ground, the team needed to select a micro-controller unit (MCU) to provide all of the control functionality necessary for the operation of the ThinSat subsystems. The MCU needs to operate within the power budget set by the system's power supply and be able to drive various power outputs to the subsystems it is controlling, such as the SDR and IMU. Additionally, it must be capable of meeting all the real-time processing requirements that are set for the system to function as expected. A trade study was conducted to choose the MCU, which resulted in the selection of the AM3358. This MCU is capable of running the Linux kernel of the chosen KubOS operating system (discussed below), operates within the power budget of the system, has all of the necessary I/O for interfacing with the ThinSat subsystems, and can be found on various development boards like the BeagleBone Black. Furthermore, this MCU is included in the Octavo Systems OSD3358 C-Sip (Complete System-in-Package) surface mount device [3], which integrates the MCU with many of the necessary components required to have a functioning embedded computer system. These components include DDR3 memory, power management, EEPROM, eMMC non-volatile storage, MEMS oscillator, and other passive components. Choosing this surface mount device for our payload board helps lower the number of discrete subsystems that need to be assembled and reduces the complexity of the PCB design.

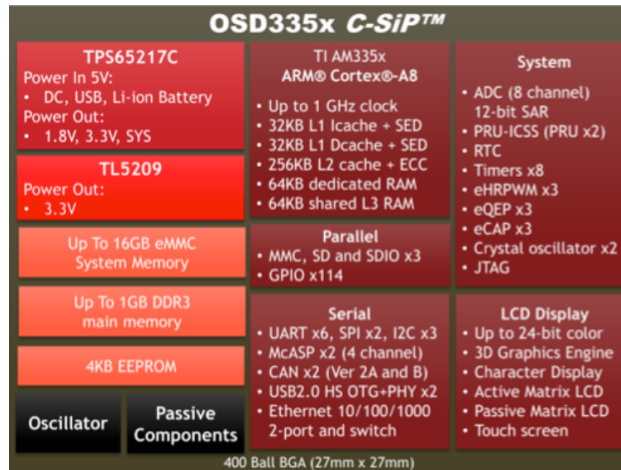


Fig. 3 OSD335x Diagram [3]

B. Flight Software

The purpose of the flight software is to perform all necessary tasks for the balloon flight, according to the functional requirements for the TOMCAT mission. There were various options available to the team from creating a custom software fulfilling only basic functionality to adopting an open-source full-grade flight software and adapting the latter to the mission's requirements. Eventually, the final decision grew from a customer's desire to extend capabilities for future missions, which narrowed the options to a software platform called KubOS. This software service provides a framework satellite developers can build upon to interface with science instruments, receive CCSDS packetized commands from the ground through a radio interface, and execute internal procedures that fulfill the space industry's software best practices.

The KubOS open-source platform is a collection of different software layers including APIs, services, and mission applications. The team will be modifying and adapting all of these layers to meet the project’s desired levels of success including an IMU service and API to collect and transmit IMU telemetry, a mission application to store IMU telemetry, a mission application to send a health beacon on regular time intervals, and lastly, a local communications service to interface with the software defined radio. All of these can be seen in Figure 4.

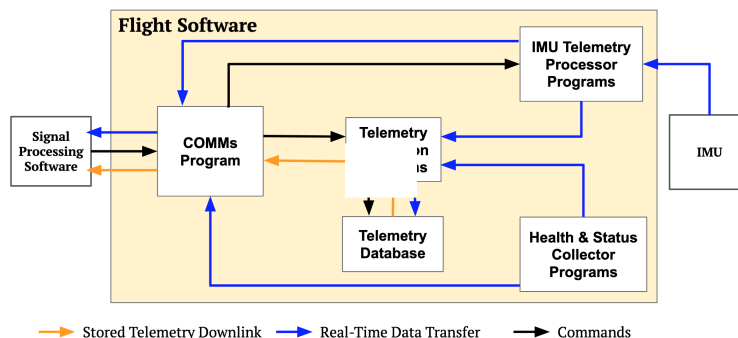


Fig. 4 FSW Development Modules

All of these programs will be modularized to remove inter-dependency and each development stage will involve unit and functional testing. The IMU Telemetry processor program's objective is to read data from the IMU, parse the IMU's proprietary Xbus packet through I²C low level communication protocol, create and fill an output dictionary, and make the formatted data available to the remaining programs including the Telemetry Collection Program and COMMS Program. Moreover, the Telemetry Collection Program systematically collects data from the IMU and Health Program and stores them in the telemetry database. Afterwards, the data in the database can be accessed through ground station commands. The Health and Status Collector Program assess the overall health of the software components

by pinging these on a regular basis. This health data can either be downlinked to the ground station or stored in the telemetry database for post flight analysis. Lastly, the COMMs Program integrates the flight software components with the software defined radio and signal processing software. It makes sure the uplink commands are in CCSDS format, depacketized, and transmitted to the appropriate subsystem/program, then it receives the system's response, while repacketizing it in the appropriate CCSDS format.

V. Payload Board Subsystems

The payload board houses many subsystems that work together with the MCU to form TOMCAT's embedded computer system, which is designed to meet the functional requirement set by the team that the payload board shall perform all necessary hardware and software tasks for system operation. The major subsystems discussed in this paper are the IMU, power management, USB circuitry, non-volatile storage, and debug circuitry.

A. Inertial Measurement Unit (IMU)

The MCU interfaces with the IMU via an I²C connection. The I²C pins of the IMU are specified in its datasheet (pins 23 and 24) and are connected to the I²C clock and data I/O of the MCU, as well as the 3.3V and 1.8V outputs of the power management system. Furthermore, pull-up resistors were included in the design to preserve the integrity of the signals traveling across the I²C lines with both the IMU device and power management system connected to them. These components are shown in the schematic in Figure 5.

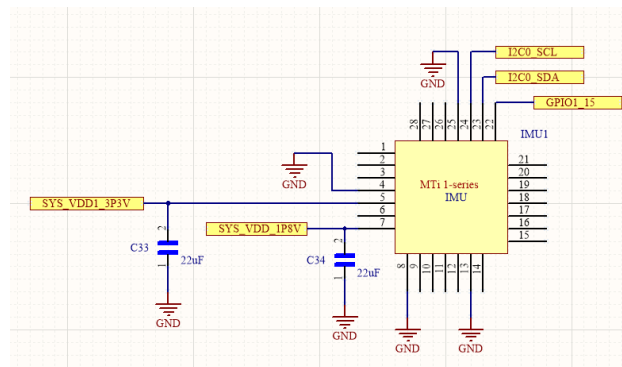


Fig. 5 IMU Schematic in Altium Designer

The IMU itself is an Xsens MTi-3 Attitude Heading and Reference System, or AHRS. It acquires a variety of data, including temperature, three-dimensional acceleration including free acceleration, high-rate acceleration, and delta-v, three-dimensional orientation including quaternion, euler angles, and the rotation matrix, three-dimensional angular velocity including rate of turn, delta-q, and high-rate of turn, and three-dimensional magnetic field. This sensor was selected for its expansive data set, ease of integration/configuration, and manageable mass and power requirements. Additionally, the sensor's nominal settings output data at a rate of 800 Hz, far above the team's requirement of 1 Hz. The data communicated to the signal processing software and SDRs can be customized for a variety of missions and provides the team's sponsor with flexibility for future applications. As stated earlier, only acceleration, free acceleration, rotation speeds, and temperature will be communicated for TOMCAT's testing in order to reduce strain on the flight software and communication systems.

B. Power Management

The MCU is connected to a power management integrated circuit (TPS65217C) inside of the OSD3358 C-Sip via an I²C interface. This produces the 5V, 3V, and 1.8V outputs needed by the various subsystems inside the ThinSat and meets the power distribution requirements for the project. Most of the functionality of this integrated circuit can be controlled programmatically through the I²C connections, such as the internal buck and boost converter output voltage, linear regulator output voltage, power up/down sequences, and over current thresholds. In addition to the I²C connections, more signal lines need to be connected between the MCU and power management so that the MCU can be

used to regulate power. The team also added mechanical power and reset buttons to physically control the power on/off and resetting of the board through the power management integrated circuit.

C. USB Circuitry

This design includes two USB connectors; one for interfacing with a computer and another for interfacing with the SDR. These USB connections follow the same standard for all USB A connectors and fuses were added on the power line to protect the board from any power spike. It was necessary to design one of the USB connectors as a host port to allow a computer to supply power and transfer data the the MCU. The other USB port was designed in an on-the-go configuration, where the state could change between a host and client port depending on the data passing through the USB connector. Configuring the USB ports was done by following the USB configuration table below (Fig. 6), supplied by Octavo Systems in their design documentation for the OSD3358.

	Configuration	
	Through Hardware	Through Software*
USB Host	Ground USBx_ID	Set IDDIG bit of USBxMODE register to 0
USB Client	Leave USBx_ID floating	Set IDDIG bit of USBxMODE register to 1
USB OTG	USBx_ID is directly controlled by USB Cable Connector	Set IDDIG bit to either 0 or 1 based on mode requirement

Fig. 6 USB Configuration Table [3]

D. Non-Volatile Storage

KubOS is built on a Linux kernel, so in order for the PCB to be capable of running this operating system, non-volatile storage such as a micro-SD card slot and eMMC is required and included in the board design. Luckily, the OSD3358 SMD has a 4GB eMMC internally integrated, so an external eMMC does not need to be mounted to the PCB. This leaves the team with the need to design an external micro-SD card connection, which was done by utilizing the datasheets of both the micro-SD card part and the OSD3358, and by following good electrical engineering design practices, such as including decoupling capacitors in parallel to reduce the mounting inductance of the part. With both the eMMC and micro-SD memory components, KubOS is able to be flashed to the embedded computer system's memory and used as it's operating system.

E. Debug Circuitry

The embedded computer system on the payload board must be designed with debugging mechanisms to assist with bringing up the board and troubleshooting any problems that may arise. Debugging circuitry on the board includes debug LEDs, a JTAG header, multiple probing points, and jumper pins. The debug LEDs are connected to various signal lines on the OSD3358 to indicate if the MCU hardware is following the expected behavior and probing points are placed on signal and power lines where expected voltages and currents are known, so that measurements may be taken with a digital multi-meter to compare the measured values with what is expected. Furthermore, a JTAG header is included because JTAG is an industry standard for verifying designs and testing PCBs after manufacture and assembly. This JTAG header is a common hardware interface that provides a PC with a way to communicate directly with the MCU and other SMDs on the payload board. The design also includes jumper pins, which allow signals like power and reset to be easily set by hand.

VI. Payload Board Shape and Layout

The size and shape of the payload board is constrained by the payload space specified in the Thinsat design. The frame is split down the middle, dividing it into the payload area and an area for support hardware that comes with the frame. A CAD model of the frame was provided to the team by Raytheon, which allowed for the shape of the payload board to be modeled. The outline of the payload board was chosen to take up as much of the payload area as possible to allow for the maximum amount of room on the board for the components. The outline of the board can be seen in Figure 7. The overall outline is the shape of the payload area, except for an area on the same end of the board as the text that is taken up by a connector for the hardware in the other half of the ThinSat. The semicircular indents are there to

make room for the bolts that will hold the PCB to the plate at the bottom of the payload area. The notch at the same end of the board as the MCU is there to give space for some connectors to the PCB.

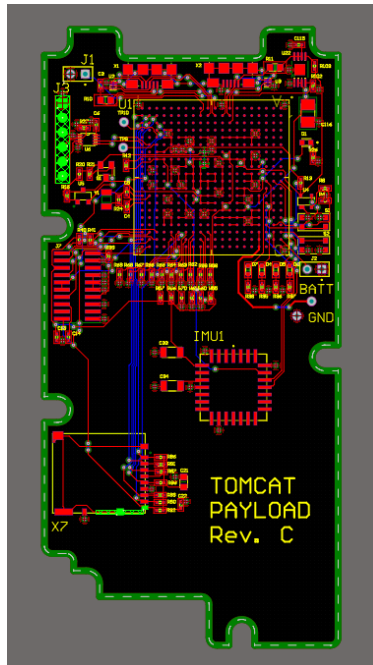


Fig. 7 Payload PCB Layout in Altium Designer

The PCB board layout design created in Altium fulfills the connectivity requirement, where all of the connections between components specified in the schematics have copper traces making these connections. Furthermore, the copper trace widths are varied so that 6 mil traces are used for signals and 10 mil traces are used for power and ground connections. A lab experiment was done to test how much current a copper trace could handle before it began smoking and the results were that a 6 mil trace could handle 1.2 A of current and a 10 mil trace could handle over 5 A. The signal currents will remain well below half of this 1.2 A limit for a 6 mil trace and the power currents will also remain below half of the 5 A limit for a 10 mil trace. The layout design was also created with the goal of maintaining signal integrity. This was done by following good electrical engineering design practices such as keeping traces as short as possible, using large power and ground planes, and routing the traces to keep mutual loop inductance at a minimum. Mutual loop inductance is where the changing current in one copper trace induces a magnetic field, which then induces a voltage in adjacent traces and therefore creates unwanted current in the adjacent traces. Avoiding large current loops in the layout, as well as routing traces on the top layer of the board in opposite directions from the traces in the bottom of the board are some of the strategies used to mitigate mutual loop induction and keep the PCB signal noise at a minimum.

VII. Payload Board Power

The Payload Board manages, regulates, and distributes all power from the battery to each individual subsystem. The design of the power system was primarily derived from the requirement that the payload shall survive and operate in the environment during all stages of the balloon flight. The Thinsat is required to maintain functionality throughout the duration of a two hour balloon flight, necessitating that the battery be able to fully power every system even in the extreme environment of near-space.

In order to begin to determine the power requirements of the system, voltage, current, and maximum power requirements were compiled from the data sheets of each component into a system-wide power budget. Because component and system-level testing requires the knowledge of the actual power requirements and fluctuations of the system over time, the maximum power usage of each component is used in the calculations as both a way to account for the worst-case-scenario, and because every component will be operating at near its full capacity during operation. The power requirements of each of the main components in the ThinSat is shown in Figure 8.

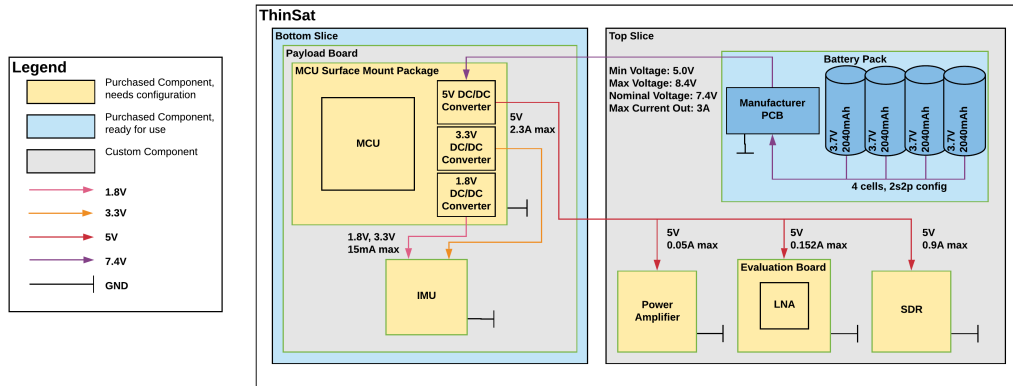


Fig. 8 System Level Power Distribution FBD

Because the communications system takes a significant amount of power to transmit data at the power required of the mission, a large enough battery and efficient power management is required to make the most out of such a small and compact system. Almost every powered component on the TOMCAT ThinSat requires a constant 5V to function, with the IMU and other sub-components on the PCB requiring 1.8V and 3.3V. While batteries are rated at a given voltage, such as 7.4V, their actual voltage varies based on their capacity, temperature, age, current draw, and a number of other variables. Therefore, it follows that the power management system on the payload board must be able to take in a range of DC voltages and output a constant voltage that is still able to accommodate the current draw requirements of each component.

Since the power distribution components on the PCB are crucial to the functionality of every powered component on the TOMCAT ThinSat, extensive testing to ensure that the full power requirements of each component can be met is required. This will be done first by connecting the PCB to a power supply and varying the input voltages to simulate the changes in voltage of the battery throughout its discharge. The DC-DC converters will be connected to power resistors to simulate the power draw of each component, and the current will be measured through each converter to ensure enough power is being supplied. The PCB will then be connected to every powered component, including the SDR, power amplifier, and LNA to ensure that each device can be powered properly. Once component level testing of the PCB is finished, full system integration testing begins, where all components will be assembled into the Thinsat casing with all mission software running in an environment similar to that at 30km altitude.

VIII. Payload Board Manufacturing

The vendor PCBWay was selected to produce the team's custom printed circuit board on account of their reliability, cost effectiveness, and history of successfully producing PCBs for the University of Colorado's electrical engineering courses for multiple years.

The team's electrical engineering lead submitted the layout of the custom TOMCAT PCB to PCBWay engineers in early February, providing ample time for review and production to occur on the vendor's end before slippage in the team's development schedule occurred. Over the course of a month, the custom PCB was reviewed, accepted for manufacturing, manufactured, and shipped from the vendor's factory.

The PCB was specified with dimensions of 107.35 by 51.08 mm and produced in four layers, totaling a thickness of 1.6 mm. Finished copper coats the surface of the PCB. All things considered, the final price of the PCB was \$354, with the components costing \$206, the labor and manufacturing costing \$126, and shipping costing \$22.

IX. Payload Board Bring-Up, Troubleshoot, and De-Bug

The PCB is designed to make the bring-up, troubleshooting, and de-bugging process as easy as possible. The payload board bring-up process involves hand soldering the remaining SMDs that were not assembled by the manufacturer. The procedure for doing this is the following: first, solder is added to the pads on the PCB with a soldering iron and with the help of generous amounts of flux. Next, a hot air gun is used to re-flow solder the SMDs onto the board, which are put into place using tweezers. Each solder joint is carefully inspected to ensure that the SMDs are in contact with the

pads and that there are no cold solder joints. To help ensure a smooth bring-up process, many things were taken into consideration during the design process. For example, the design avoids the use of SMDs that are too small to solder, such as parts in packages that are smaller than the 0402 size, which was chosen based on experience. The design also includes strategically placed thermal vias for heat dissipation and adequate spacing between parts so that they can be comfortably placed without the risk of bumping adjacent parts. Once the payload board is assembled, a kernel may be flashed onto a micro-SD card and inserted into the slot on the board. To simplify the process of verifying the payload board's design, one of the Beaglebone Linux operating systems may be used initially to test that the embedded computer system behaves as expected. Once the micro-SD card containing the OS is inserted, the reset button is held for a few seconds and then the MCU is expected to boot the loaded Linux kernel. This is indicated by some of the debug LEDs and can also be confirmed by presence of the board's Linux terminal when plugged into a PC. If the board does not boot or interface with the subsystems as expected, the debugging circuitry contains tools to help the troubleshoot and debug process. Once the payload board is confirmed to be working with the Beaglebone Linux OS, it may be swapped out with KubOS following the installation procedures written in the official documentation.

X. Conclusion

The TOMCAT Payload Board allowed the integration of numerous satellite subsystems onto one convenient part for easy installation and debugging. By successfully fitting into the provided payload area, serving as a means of power distribution, and meeting computation requirements for both the flight and signal processing software packages, the Payload Board allowed the TOMCAT satellite to fulfill major project performance requirements. Further, with the inclusion of debug circuitry, issues during manufacturing and integration can be easily identified, isolated, and fixed.

In the short term, additional focus will be placed on build-up and integration of the board with other payload subsystems, as delivery of the 3rd and nominally final revision is still pending. These steps will allow the TOMCAT team to meet Level 2 and Level 3 success criteria.

Longer term, development of such high-powered computing in a form factor which is sub-1U has wide-reaching implications on the spaceflight industry. Existing options boast lower performance, narrower environmental constraints, and costs that are orders of magnitude greater than the TOMCAT board described in this paper [4][5]. Further development on this design could result in cheaper access to high-profile missions not previously possible, like online or distributed state estimation and formation flying. Additional development could increase functionality, reliability, and technology readiness level.

Acknowledgments

The TOMCAT team would like to thank Mark Werremeyer and Sheldon Clark at Raytheon for providing advice and financial support for the development of this project. The team would also like to thank our advisor Dr. Kathryn Wingate for her help and feedback throughout the process, as well as the rest of the PAB and Dr. Nick Rainville.

References

- [1] Robert, T., et al., "The ThinSat Program: Flight Opportunities for Education, Research and Industry," *SmallSat Conference 2018*, August 2018. <https://doi.org/SSC18-WKIX-05>.
- [2] Daily, J., Orvis, M., and Voss, H., "ET-Sat Interface Control Document (ICD)," *NearSpace Launch (NSL) Information*, December 2018.
- [3] *OSD335x System-in-Package A New Era of Integration and Flexibility based on AM335x*, 2020. https://octavosystems.com/octavo_products/osd335x/.
- [4] *ISIS On board computer - Brochure*, 2020. <https://www.isispace.nl/wp-content/uploads/2016/02/IOBC-Brochure-web-compressed.pdf>.
- [5] *EnduroSat OBC Type 1 - Datasheet*, 2020. https://www.endurosat.com/modules-datasheets/OBC_Datasheet_Rev1.1.pdf.