

A large, faded watermark of the NASA logo is centered on the page. It features the word "NASA" in its characteristic serif font, with a red swoosh and a white orbital path around it, all set against a light blue circular background with white stars.

Iridium 9523 Modem Development Platform and Integration

Tessa Aus, Jordan Lynn, David Handy, Jonathan Hanson, Christopher Ocker

Table of Contents

Executive Summary	3
Project	3
Achievements	3
Background	3
Objectives	4
Planning	4
System Design	5
Hardware Design	5
Software Design	6
Testing	6
Hardware Testing	6
Software Testing	7
Growth	7
Appendices	9

Executive Summary

Project

The Iridium 9523 Modem Development Platform and Integration project pursues two goals: to create an easy-to-implement hardware and software interface for developing systems utilizing the 9523 Modem, and to apply these interfaces to integrate the 9523 Modem with the TechEdSat (TES) project at NASA Ames Research Center (ARC). Current satellite communication schemes on TES use an Iridium 9602 Modem equipped with only a Short Burst Data (SBD) mode to transmit single data packets at regular intervals. Upgrading to the 9523 Modem enables the Dial-Up data streaming mode that is capable of significantly increasing throughput at critical intervals of flight.

Achievements

Our team has completed a full software interface for the 9523, completed all but one of our hardware interface objectives, and created an implementation plan for using the 9523 for data streaming on-board a generic Low Earth Orbit spacecraft. The software library and associated documentation are packaged and provided to ARC for flight software design. Iterating development board designs, the physical interface and power connections to the Iridium 9523 modem have been created and validated. Data interfacing was designed to provided specification, but was not successfully verified will require further, direct assistance from the hardware manufacturer. Application engineers will be able to take advantage of all of our progress to greatly simplify the process of integrating the Iridium 9523 Modem on a spacecraft.

Background

The TechEdSat (TES) series of nano-satellites, engineered at NASA Ames Research Center, utilize the Iridium satellite network for two-way communication. The current communication system uses an Iridium 9602 modem to provide uplink and downlink via its short burst data (SBD) capability. SBD allows for up to 340 bytes of data to be sent from the modem in each message and up to 270 bytes of data to be received by the modem.

Due to the nature of TES missions (the satellites themselves do not survive reentry into Earth's atmosphere), the only data that can be collected is that which is downlinked while the satellite is in orbit. Therefore, a higher data rate and throughput is desired. This would simplify and accelerate the process of downloading data such as photographs. In addition to SBD, the Iridium network supports dial-up data. The Iridium 9523 modem features both SBD and dial-up capabilities, allowing for backward compatibility with the existing Iridium 9602 system. The Iridium dial-up system currently supports data rates of up to 2.4 kbit/s.

The team received an Iridium 9523 modem in the A3LA-RS package from NAL Research, a vendor of Iridium services. This package included a carrier printed circuit board (PCB) for the modem which included a SIM card socket, boost converter circuitry, and a TTL/RS-232 serial converter. The carrier board wanted by NASA Ames includes the same functionality but eliminates the need for the serial conversion (as the Teensy microcontroller uses TTL serial).

Objectives

The objectives of this project were organized around the 9523 and to allow final integration with TechEdSat(TES). The major goal was to create a complete and successful dial up data system for the planetary probes NASA Ames uses on a daily basis. To allow a successful dial connection the team needed to create two separate items to allow a final successful project. The first entailed the required hardware to allow the bit exchanging and signal transferring power to allow proper connection with the Iridium satellite constellations. In addition parallel to the creation of the hardware, software needed to be created to allow the proper communication and bit transferring. A stretch goal was additionally made in the software section to create a library for future generations of Senior Design as well as TES to use to better understand the software and functionality of the system at hand.

Once completed the team would test multiple flights using an air balloon to demonstrate proper connection and transferring of data with the new board and software. Once completed and tested thoroughly syncing up with NASA Ames would commence to demonstrate the usability of the product to the head leaders in TES.

Planning

Planning was separated into two major categories, expense budgeting and time management of the team's final goal. Due to to last years team additional money was added to the overall budget as a safety padding in case of unseen circumstances like equipment failure. With this funding and additional funding from NASA Ames the budgeting was separated into two main subcategories; development and travel.

Development was primarily used for the creation of the various dev board revisions, teensy's, and cabling/ material associated with the interface. The multiple teensys where purchased to allow a safety precaution to the potential failure of rigorous testing on the hardware. In addition throughout the 2016/2017 school year three revisions were ordered for the dev board because of various failure points in the design. With these and various materials associated with the creation and development of the board the estimated cost of the products purchased was 3,460.00 leaving and additional 540.00 dollars unspent.

The second sub category for expenses was the travel expenditures. These expenditures entailed the cost of flight, and lodging to the NASA Ames Research Center to integrate the final design with the current system intact on sight in Mountain View, California. The provided monetary support was 7,889.00 dollars and after expenses for traveling where complete a remainder of 829.00 dollars was left to pass on to next year's Senior design Team.

All traveling and budgeting were recorded on a Gantt Chart to allow a streamlined set of time frames to follow to make the project's end goal successful. The schedule was separated into a Fall and Spring timeframe in which the fall was dedicated to the creation and design of the Dev Board and the spring for testing and integration.

The largest components for the fall was the time frame to create both the architecture for the software and the the hardware design/ creation of the board. All of these dates were accomplished for the fall time line to create an optimal environment for spring testing. Though the fall was successful in its time frame major setbacks occurred in spring. With the issues associated with the architecture of the board the timeframe was pushed back for testing because of the unviability of the boards function. This lead to the board not being successful in its operations. Opposingly the

software team was successful in following the timeline of the Spring semester. Operational software was created as well as a library for future students on this continuing project.

System Design

Implementing Iridium's dial-up data streaming mode as an end-to-end communication system required a paradigm shift from the previous satellite communication system on TES. In its flight operations, the latest iteration of TES utilized the Iridium 9602 Modem as its sole uplink and downlink mechanism. The 9602 affords a smaller, lighter daughter board at the cost of only offering short burst data (SBD) capability. Fixed-size packets were transmitted at one minute intervals for data collection, and packets could be generated on the ground to asynchronously deliver commands.

9602 transmissions would start a Comma Separated Value file of raw data values passed in over a serial interface from the controller. A packet would travel via Iridium's proprietary satellite network, arriving at a node near the Iridium Gateway ground station. After being relayed to the ground station, the data payload on the packet would be attached to an email and sent to a specified email address for the service subscription. In this way, the controller supplied raw data and the user received it via email with Iridium handling all transmission protocol.

While leveraging the SBD capabilities of an Iridium Modem does not require any protocol design considerations, implementing the dial-up capabilities does require this layer of complexity. Dialing-up involves establishing a connection from the modem to the Iridium Gateway, but once that connection is established the packet flow is controlled by Internet Protocol (IP). Rather than resulting in an email, dial-up datagrams are delivered to The Internet, meaning the IP encapsulation is resolved to some host.

Going from an email based system to an IP based system requires two fundamental changes. First, the controller on the spacecraft must be providing IP packets to the modem for transmission. This is a massive change, as creating and maintaining an IP stack is a serious resource sink for a microcontroller. Second, the IP packets must resolve to some user-designed host serving as a ground station. Creating and hosting server software to handle these IP packets is a significantly more involved logistical challenge as compared to the email receipt of data.

Hardware Design

The primary hardware objective was a development platform for the Iridium 9523 modem to interface with a Teensy microcontroller using TTL serial, and to support the development of a generic software library. The Iridium 9523 development board was designed using EAGLE CAD software.

The initial revision of the development consisted of a footprint for the Iridium 9523 modem, breakout of RX/TX, 5V/GND, three status LEDs, SIM card socket, and boost converter circuitry. The boost converter circuit was simulated in LTspice to ensure the boost from 5V input to 27V would operate correctly.

The second revision fixed an incorrect pad size on the circuit board and added thermal vias underneath the boost converter. The third revision added additional pins to breakout the rest of the 9-wire serial connection in an attempt to get serial communication with the modem functioning.

Software Design

The software portion of the project was implementing a library for the Iridium 9523 modem that would perform all of the tasks as the Iridium 9602 modem with the addition of allowing streaming data using the 9523's dial-up capabilities. The reason for switching to the 9602 modem was that it only supported short burst data which sends a message similar to a text, but it can't be queued so it is impossible to send lots of data very fast. With the dial-up connection, the 9523 modem allows for a much higher bandwidth though it is still dial-up speed. So our project essentially turned into adding a dial up function to the existing library, but with a little more information it became much more than that.

The Teensyduino first creates an 9523 object that dials up to the Iridium network. If that is unsuccessful it will keep trying. After it has made a connection, if the Arduino receives the command to start a dial-up connection the TCP/IP stack starts packaging the data from whatever sensors and systems are on the TES. After the TCP packages have created, they are passed to the modem and sent to the Iridium Constellation which bounces that information to the Iridium ground station in Phoenix, Arizona. The Iridium ground station then handles the packets and sends them to the client.

For our Iridium 9523 modem library, we wanted it that would be easy to inherit and use. A major reason we wanted this was because the original Iridium 9602 modem library we inherited from NASA AMES was not so. To prevent the past from becoming the future we set a few goals that would lessen the burden for any team that would have to search the library. Those goals were to keep the cyclomatic complexity of the code below a 10, have a comment for every other line of code, and to create a JavaDocs style reference guide detailing files and functions. We stayed true to our goals except for the comments per line, but on the other fronts the code stayed below 10 on its cyclomatic complexity and using a free tool called Doxygen, the library was turned into a HTML reference guide.

Testing

Hardware Testing

Hardware testing was performed independently on each portion of the development board. Serial communication was tested with both a Teensy microcontroller and a USB to serial converter and Linux computer. Boost converter circuitry was verified on an oscilloscope.

Prior to testing the Iridium 9523 modem on the development board we created, the modem was tested within the A3LA-RS package. This allowed us to verify that the modem was functional, responding to AT commands and sending SBD messages. The setup for these tests used a custom DB15 wiring harness connecting the Iridium 9523 modem to both a TTL/RS-232 serial converter and 5V power supply. The Teensy microcontroller was then connected to the other side of the serial converter.

To test serial communication with the Iridium 9523 modem when mounted on the development board, the RX pin on the modem was connected to the TX pin on the Teensy microcontroller the TX pin on the modem was connected to the RX pin on the microcontroller. The Iridium 9523 modem was once again connected directly to the same 5V power supply. This setup eliminated the need for the DB-15 wiring harness and large TTL/RS-232 serial converter.

Unfortunately, although the modem was proven to be functional (responding to AT com-

mands) when mounted in the A3LA-RS, when attached to the development PCB we were unable to get a response back from the modem after sending any AT command. We were able to verify by checking the current draw of the modem that it was powered on and followed the same boot-up sequence we observed with it mounted in the A3LA-RS. The absence of status LEDs or any other debugging information on the modem itself made further troubleshooting difficult.

Steps we took to attempt to alleviate or at least understand the communication issue were swapping RX/TX pins, viewing the serial output from the Teensy to ensure it was valid TTL, and viewing the serial output from the Iridium 9523 modem (which was nonexistent).

Using an oscilloscope, we were able to test the performance of the boost converter circuitry. The scope wires were attached to the VBOOST and GND lines on the development board and an external power supply was used to pull the enable line of the boost converter high. The 5V to 27V boost output is shown in Figure 1. The output holds steady at 27V (tested independent of modem, without load).

Software Testing

Software testing began after receiving last year's 9602 modem library. It didn't compile right away, but that was because of some strange syntax error. After verifying the library, the first semester was spent figuring out how to communicate to the Iridium network. After attaining a reference guide to AT commands that could be used on the modem, the AT+CSQ became the bread and butter of testing. AT+CSQ essentially tries to ping the Iridium constellation and reports back with a signal strength. Unfortunately being on campus hampered the signal strength because of interference. On the rare occasion where the modem had a signal dialing up became the problem.

There was limited documentation readily available about how to dial up so NAL sent the team some information on how to set it up with Windows and with Linux. This allowed a dial up connection from a laptop but was no use to the project since we need to use an embedded system. The second semester's testing consisted of trying to set up a dial-up connection to Iridium using the Teensyduino, but ultimately we were not getting the responses we wanted. The packets that are sent from the Teensy are supposed to end up at a predefined address, but no such packets arrived. The team is unaware why the dial-up connection isn't working as planned and it likely because of a handshake process that is unknown. Despite the lack of success, the account was charged for dial-up data at some point during the project by Iridium meaning that some data was transferred.

Growth

The focus of the work conducted in this project has been to lay the foundation for future planetary probe teams to leverage the Iridium 9523 Modem's rich feature set. Our efforts have been largely successful, and while that is intrinsically exciting the real value is in how it enables project growth.

The only housekeeping task being handed off is the final step of verifying the data interface of the hardware development template board. Records of our correspondence with the vendor, our characterization of the design and its performance, and of our attempted solutions have been gathered and passed on to the team at ARC so that they may proceed as informed as possible. Due to their position, it will likely prove easier for them to collaborate with the manufacturer to identify the issue with the data interface.

Design teams with an existing 9523 field-application board may immediately begin taking advantage of the software library we have produced. Creating and testing flight operations software will require significantly less development time with the high-level functionality already ready to implement.

Testing our end-to-end system with a working 9523 field-application board will also provide a practical demonstration of the ground station server software. This software is designed to our expected use, but has not received serious test and debug time due to the lack of a successful hardware data interface. Going forward, perfecting this receiver software will be critical for recovering gracefully from disconnects or stitching received data back into images.

Appendices

Team Rocket Senior Design Data Flow Diagrams

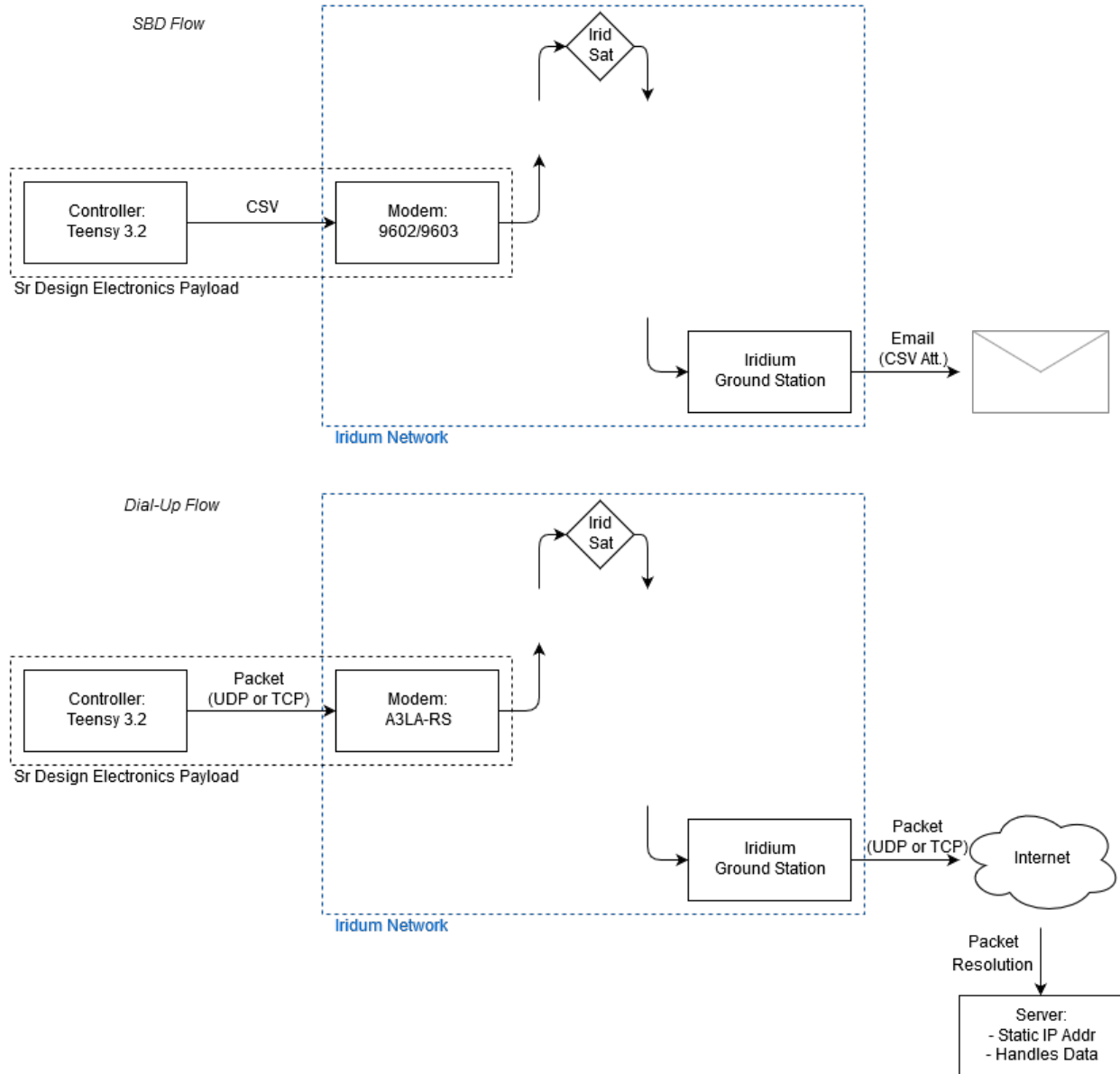


Figure 1: System Design Data Flow

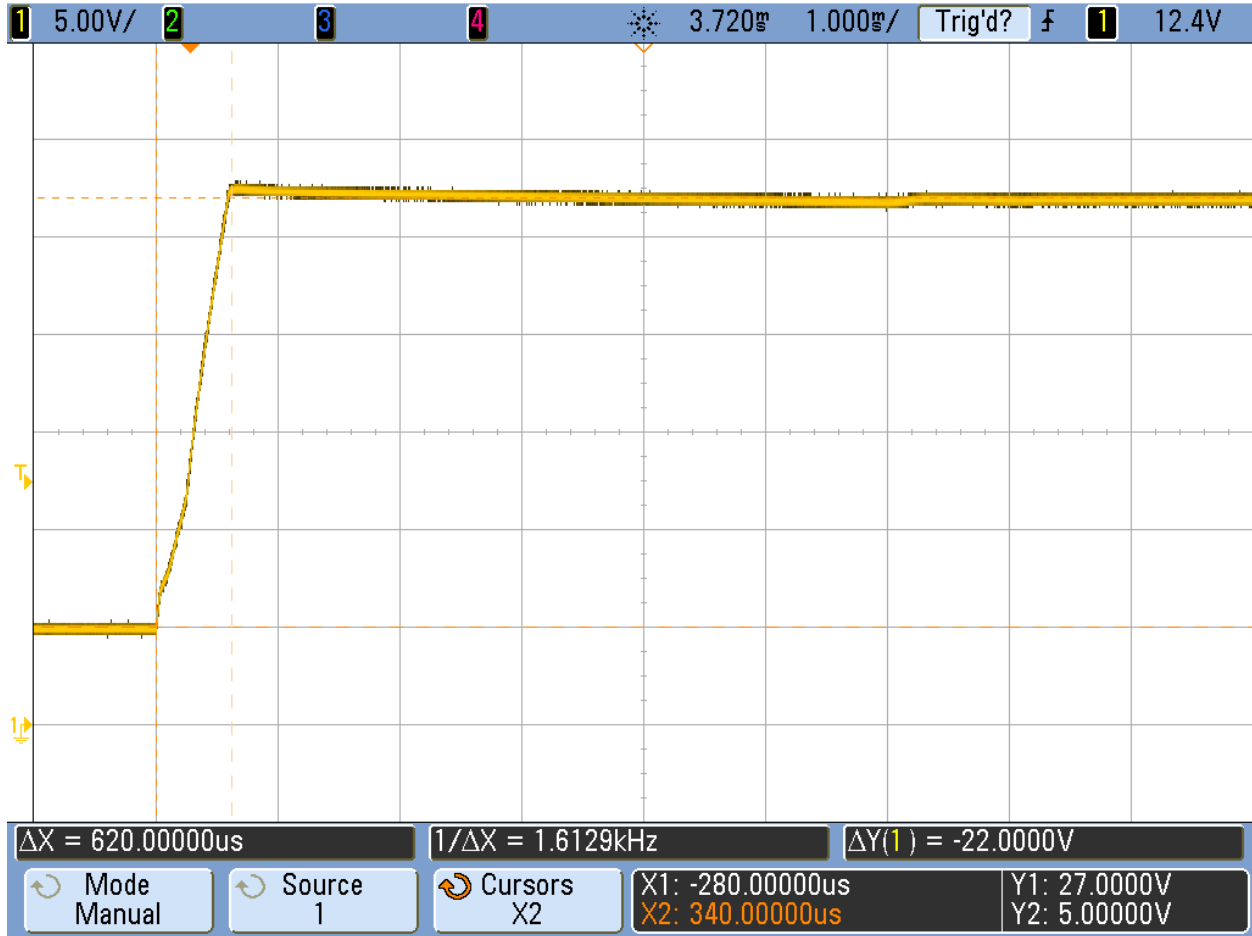


Figure 2: Boost Converter Performance