

ELECTRICAL HARDWARE AND FIRMWARE COMPUTER DESIGN FOR
APPLICATIONS IN SOLAR FLARE X-RAY EXPERIMENTATION BASED ON STM32
MICROCONTROLLERS

by

Jackson Chaney Novak

A professional paper submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

Month and Year ETD is accepted by Grad School (e.g. December 2022)

DEDICATION

To my wife, Sophie. Thank you for being there for me throughout this whole process. I couldn't have done it without your support.

ACKNOWLEDGEMENTS

Thank you to Brock LaMeres for advising me in my degree and reviewing this paper

Thank you to Larry Springer, Charles Kankleborg and Tyler Holiday for advising and reviewing my work

Thank you to Jake Davis and Cole Arthur for your mechanical designs

Thank you to Sara Dolan for assisting me in the electrical design process

Thank you to Sarah Pawlowski and Chris Entzel for assisting in testing and debugging

TABLE OF CONTENTS

INTRODUCTION	1
DESIGN TARGET AND MISSION OVERVIEW	2
CARRI-SUN Overview	2
Scientific Target	2
Electrical Design Target	3
SFXTI Overview	3
Scientific Target	4
Electrical Design Target	4
DESIGN IMPLEMENTATION	6
Design Considerations for Space Applications	6
CARRI-SUN Implementation	7
Hardware Design	7
Firmware Design	8
Support Software Design	11
SFXTI Implementation	13
Hardware Design	13
Firmware Design	14
Support Software	17
RESULTS AND DISCUSSION	19
Results of the CAPRI-SUN Design	19
Testing Process	19
Results of the SFXTI Design	20
Testing Process	20
CONCLUSIONS AND RECOMMENDATIONS	22
Computer Design	22
Shortcomings	22
Advantages	22
Future Projects	23
BUBO	23
IMPRESS	24

LIST OF FIGURES

1. Figure 1 CAPRI-SUN Hardware Block Diagram	5
2. Figure 2 CAPRI-SUN Firmware order of operations	7
3. Figure 3 QT Designer GUI for CAPRI-SUN	9
4. Figure 4 SFXTI Hardware Block Diagram	11
5. Figure 5 SFXTI Support Software GUI	15

GLOSSARY or NOMENCLATURE

Hi-C; High-resolution Coronal Imager

MSU; Montana State University

SSEL; Space Science and Engineering Laboratory

CAPRI-SUN; high cadence low-energy passband x-ray detector with integrated full-sun field of view

ADC; analog to digital converter

GPS; Global Positioning System

SFXTI; solar flare x-ray imaging investigation

ISS; International Space Station

STP; Space Test Program

CPU; central processing unit

ONFI; Open nand flash interface

SPI; Serial peripheral interface

UART; Universal asynchronous receiver transmitter

USB; Universal serial bus

GUI; Graphical user interface

DMA; Direct Memory Access

CSV; Comma Separated Value

BUBO; student-built ultraviolet burst observer

IMPRESS; impulsive phase rapid energetic solar spectrometer

ABSTRACT

This paper covers the electrical and firmware design process for two missions carried out by the Montana State University, Space Science and Engineering Laboratory. The missions were the NASA HI-C Flare subsystem, CAPRI-SUN, and the NASA funded SFXTI payload. It covers the full design process, including requirements definition, hardware and firmware design and the full testing process for both projects. Additionally it discusses some of the challenges involved with designing hardware for space applications. Both systems use a similar computer architecture, which is based on an embedded STM32 microcontroller. The two missions, however, target significantly different scientific instruments. After the success of the design for the HI-C mission it was desired to reuse the hardware design for SFXTI with new firmware in order to meet tight deadlines. This decision had one major pitfall, namely, where as the STM32s bare metal programming excelled in keeping the system robust through simplicity for CAPRI-SUN, it fell short on SFXTI due to the lack of USB drivers and scheduling macros that a Linux system provides. In the end, it was necessary to augment the SFXTI design with a second computer system, which ran LINUX, in order to pass the data from the instrument to the main communications computer.

INTRODUCTION

In the modern day of computers and electronics we have come to rely on the idea that the internet will continue to function and that our online data will still exist every time we log in. One factor that threatens this expectation is the solar flare. Although very rare, it is possible that a large enough flare could disrupt many of the electronic systems we use every day. One way to mitigate this threat would be predicting when a solar event might occur and making necessary preparations. The problem with this plan lies in the fact that solar flares are not well understood, and cannot be predicted well. Oftentimes, the best way to predict a phenomenon is to understand the underlying mechanisms that drive it. Montana State University's physics department seeks to gain insight into solar flare behavior through the measurement of the x-ray photons which are released during a solar flare event. The experiments being run aim to measure the photon count, and intensity in ways they had not previously.

So how does an experimenter go about measuring these photons? X-Ray photons have a hard time getting through the atmosphere meaning these experiments cannot be run from earth. In order to mitigate this problem the experiments can be sent on high altitude balloons, sounding rockets or even launched all the way to orbit, which allows for a longer observation time. Sending an experiment to space comes with a number of other challenges such as thermal considerations, vacuum survivability and large vibrations which must be sustained during launch. Additionally, since many of the systems which might be used to run the experiment on the ground, such as computers and power supplies, cannot be sent to space due to size and weight, custom hardware and software must be designed to run the experiment during flight. This paper covers the design of these custom systems which run the experiments during their

flight and pass the data back to the ground for later analysis. The design considerations and challenges as well as the testing process are discussed at length in order to detail the work associated with such a undertaking and inform future designers as to what has worked and what hasn't during the design process.

DESIGN TARGET AND MISSION OVERVIEW

CARRI-SUN Overview

The NASA High-resolution Coronal Imager Flare, Hi-C Flare, is a sounding rocket payload which is targeting high-resolution images of the sun during a solar flare event. The payload would take images of the sun for around fifteen minutes. The Montana State University Space Science and Engineering Laboratory, MSU SSEL, designed a secondary payload which would ride along with the Hi-C mission called high cadence low-energy passband x-ray detector with integrated full-sun field of view or CAPRI-SUN.

Scientific Target

This payload measures the power density of soft x-rays during a solar flare event. This main scientific difference between CAPRI-SUN and previous experiments is the sampling frequency, which is one kilo-hertz. Previously measurements had only been made at around half a hertz or a sampling period of two seconds. The scientific goal for the project is to determine what, if any, high frequency information is present in the x-ray solar flux. This target would be achieved using the use of the AXUV100G Photodiode as the scientific instrument.

Electrical Design Target

In order to enable the measurement described above two main hardware systems were developed. Firstly, an analog filter was designed to take in small currents from a photo-diode, amplify them to a workable range with a trans-impedance amplifier and filter them to eliminate aliasing from frequencies above five hundred hertz. The resultant output of the analog filter was then moved to the input of an analog to digital converter or ADC, called the ISL26104, built by Renesas.

Secondly, a digital computer system was designed to manage the whole system. Namely the computer was responsible for acquiring the data from the ADC, storing the data, synchronizing the data to a GPS timestamp and packaging the data for transmission to the ground station. Firmware was designed for the computer in order to make all of the above tasks possible.

Finally a power management system was designed to handle filtering and conversion of the input twenty-eight volt direct current line. The required voltages were plus and minus fifteen volts for the analog system, five volts as digital system power and three point three volts for the microcontroller. As mentioned the filtering on the input power was important because several other payloads on the Hi-C Flare mission share the same input power rail and could be sensitive to power line noise.

SFXTI Overview

The NASA funded mission for solar flare x-ray imaging investigation. SFXTI, was designed with a similar target to CAPRI-SUN. This mission would be flown on a larger payload

to the International Space Station, ISS, where it would continuously take data for one year. Due to the one year duration of the experiment SFXTI would be able to measure a number of solar flare events. This mission was set to be delivered under a faster timeline than typical SSEL projects and would therefore attempt to use heritage designs from several other SSEL projects.

Scientific Target

SFXTI would measure individual photons and count them into bins, each bin containing the number of photons registered within a given energy range over the elapsed period. The counts would be sampled at thirty-two hertz giving both time/ frequency resolution as well as spectral resolution. This science target would be achieved using a silicon photo-multiplier measurement system designed by Bridgeport Instruments called a SiPM-Morpho.

Electrical Design Target

For the SFXTI electrical design, reuse of previous hardware and firmware was crucial as the timeline for the project was short. For the hardware design, the target was to re-use as much of the design from CAPRI-SUN as possible. This would involve only a couple quick changes to the schematic and a redesigned layout to fit the SFXTI form factor and connector types.

The SiPM-Morpho instrument used on SFXTI is the same device used on another MSU run project, IMPRESS. The reuse of an already tested system allowed the project to rely on the operational and calibration testing which had already been completed by the physics team here at MSU. Additionally, the plan was to reuse code written by a software team at the University of Minnesota, originally designed for a computer being developed by University of Minnesota, to interface the MSU built computer with the instrument on SFXTI.

DESIGN IMPLEMENTATION

Design Considerations for Space Applications

When designing systems for space there are several main considerations. Firstly all systems need to have low outgassing to withstand vacuum conditions. This is especially crucial for metals, such as cadmium, which can end up creating shorts in the electronics as the metal dust spreads. Additionally, electrolytic capacitors can explode in vacuum and must be avoided. As for cabling, NASA has a set of insulation types which can be used in vacuum. For both projects discussed in this paper, PTFE insulated wire is used for all cabling.

Another consideration when designing for space is vibrations. During launch, the system experiences a broad band vibrational load. Mounting of the boards within the spacecraft is important as to avoid anything within the payload breaking during launch. Careful consideration for mounting hole location is taken along with the models created by the electrical team to mitigate the chance of these failure conditions.

For SFXTI, a number of requirements were set by the STP team to achieve space readiness. Specific wire types and gauges as well as vibration testing requirements were met in order to uphold these requirements.

CARRI-SUN Implementation

Hardware Design

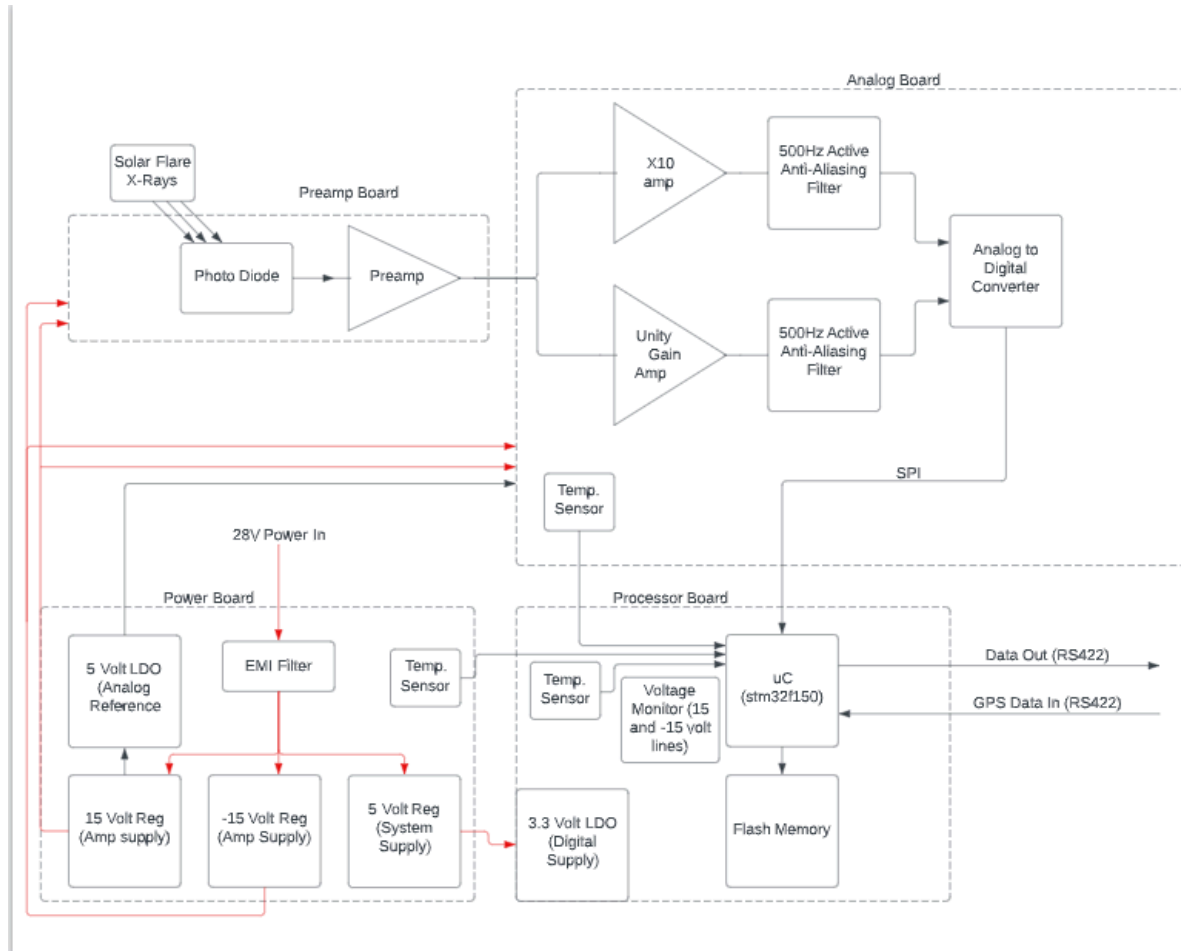


Figure 1 CAPRI-SUN Hardware Block Diagram

For the CAPRI-SUN project, four separate boards were used to fit the electrical design into the space. Firstly a transimpedance amplifier was used to boost the small currents output by the AUVX100G photodiode into a measurable voltage. This signal was then routed through a coax cable to the analog board which held two anti-aliasing filters, one having a gain of one and the other a gain of ten. This allowed the system to use a lower gain for large flares which would saturate the higher gain channel and the higher gain for smaller flares which would not be visible

in the low gain channel. The analog board also contained a high speed analog to digital converter which used an external oscillator to synchronize its sample time to precisely one milli-second.

The third board, named digital, contained an STM32L4 microcontroller by STMicroelectronics, a MT29F16G08 sixteen giga-byte flash memory device built by Micron and two SN65HVD RS422 transceivers by Texas Instruments. The microcontroller communicates with the analog board via a serial peripheral interface in order to receive the data. The flash memory chip was used to store data during flight such that it could be recovered after landing. The RS422 transceivers handled telemetry output to the spacecraft to be sent to the ground station and reading of global positioning system, GPS, packets in order to link the measurements to a specific time.

The final board was named power and was responsible for filtering of the input power. An electromagnetic interference, EMI, filter was used to protect the spacecraft from power line ripple caused by the CAPRI-SUN instrument. Three isolated dc/dc converters, purchased from Calnex, were used to convert the input twenty-eight volts to fifteen and minus fifteen for the analog system as well as five volts for the digital system.

Firmware Design

The main purpose of the CAPRI-SUN firmware was to facilitate collection of the measured data. The ADC used on the analog board would pull the master in slave out line low when a measurement was ready to be collected. The microcontroller would then toggle the clock line and read in the data. Additionally at start-up the SPI bus was used to send initialization commands to the ADC. Namely it would select a channel, set the sample rate, and set the ADC's internal gain.

In order to organize the data for both storage within the onboard flash memory and for packets being sent to the ground station data was stored in one second chunks. This means for each packet there was one thousand samples of twenty-four bit data from the ADC. This partitioning was driven primarily by the flash memory page size for 4098 bytes. Each sample was paired down to twenty-three bytes in order to allow for a gain bit. The gain bit was either a one or a zero depending on whether the high gain filter or the low gain filter was used for the sample.

Along with the one second partition of data each packet also contained a GPS timestamp. Once a one second data stream had begun the firmware would begin scanning for GPS packets. The GPS system used several packet types each being sent at different rates. Because of the need for only timestamp data, the system would wait until the TS, or timestamp, packet headed was read in. The system would then save the timestamp as well as an indicator for which sample, within the one second partition, the timestamp was received in conjunction with. This allows for the synchronization of the input samples to real time and indicates in the analysis if the system is running slow or fast.

Sample Count	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	...	997	998	999
Other Operations	Begin Data Transfer (by DMA) Save Previous Packet to flash								Scan for GPS							Save time stamp from GPS and sync byte (16 in this case)							Wait						

Figure 2 CAPRI-SUN Firmware order of operations

Once a one second packet has completed the system begins a packet transfer to the spacecraft to be sent to the ground station using the direct memory access system built into the microcontroller. This allows the system to transfer the data without the use of the main CPU.

While the data is transferred the main CPU saves the packet to flash memory as well. This used a sixteen wire interface consisting of an eight wire bus for data transfer and eight wires for control to indicate to the flash memory device what is being sent. This interface was handled primarily by a high level driver that is free to download from Micron called open nand flash interface, ONFI. Low level control was handled by only a few methods which set the pinouts allowing the high level interface to communicate with the hardware. In order to manage the flash memory a pointer was saved in the first available flash page. On boot the firmware would read this page, indicating the last page which had been written with data, and know which page to write next. This means that data can be stored over the course of multiple power cycles of the system without overwriting until the full sixteen giga-bytes were filled.

Support Software Design

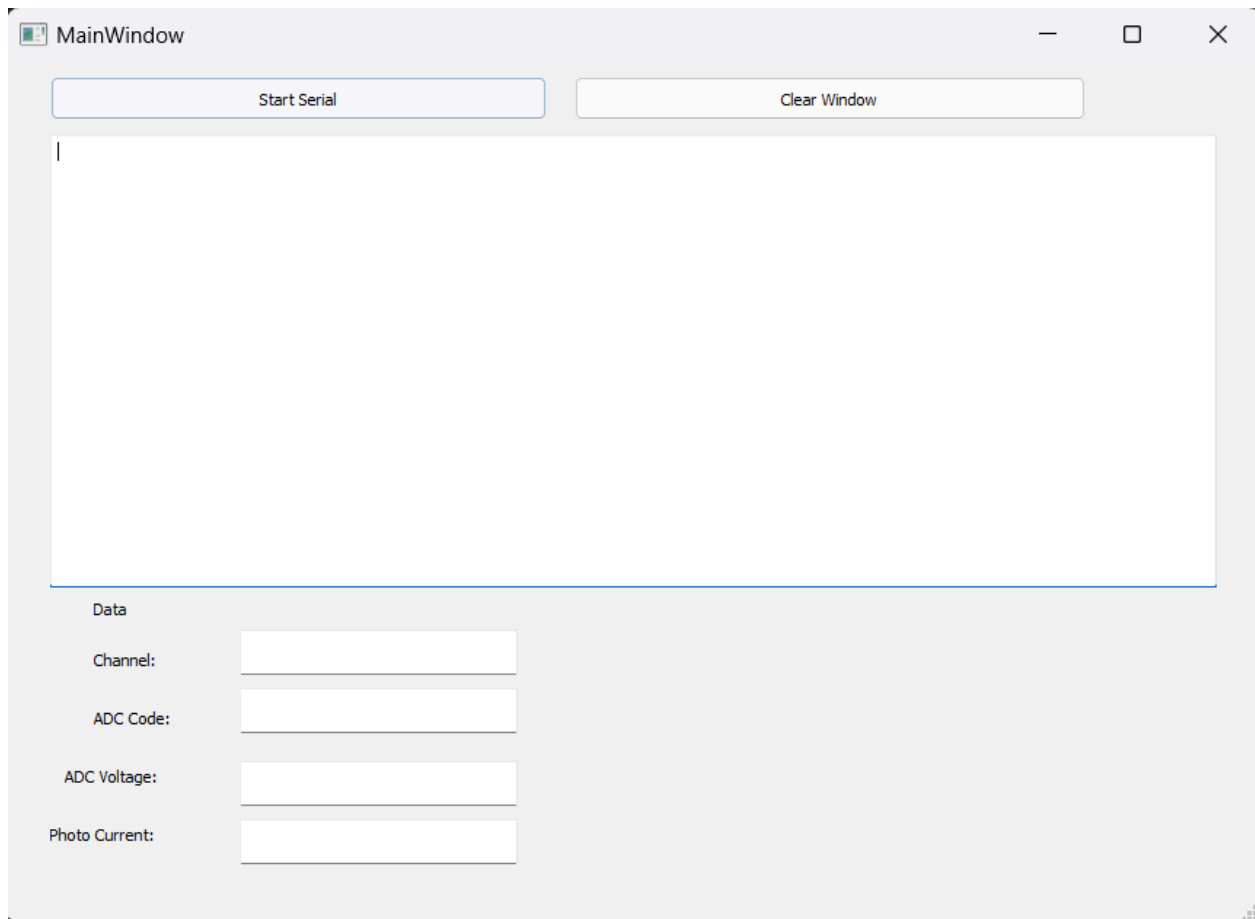


Figure 3 QT Designer GUI for CAPRI-SUN

In order to facilitate the debugging and testing process of the CAPRI-SUN firmware and hardware a python serial monitor and GUI was developed. The program was set up to run a QTDesigner graphical user interface, GUI, along with accessing a serial port on the computer it is run on to communicate with the CAPRI-SUN computer. The graphical editor for QTDesigner allows a GUI to be built with drag and drop widgets and saved as a “.ui” file. This file can then be converted to a python file containing the GUI information with a simple command line input. Since the CAPRI-SUN instrument is not set up to receive any commands, it simply begins sending data to the UART port which can be read by the python script. As data arrives it displays

all information it receives in the main window of the GUI. Additionally, the python program parses the channel and ADC code from one value in each packet. The ADC code is also converted both to a voltage value as well as a photo current value to help with readability for the user. These values update once per second, the same as the packet rate, allowing the user to track approximate values over time when testing. This also allowed the firmware developer to read the computer's output directly in order to ensure the communication protocol is working correctly. While the GUI is running all received data is stored to a comma separated value, CSV, file which can be used for later data processing.

SFXTI Implementation

Hardware Design

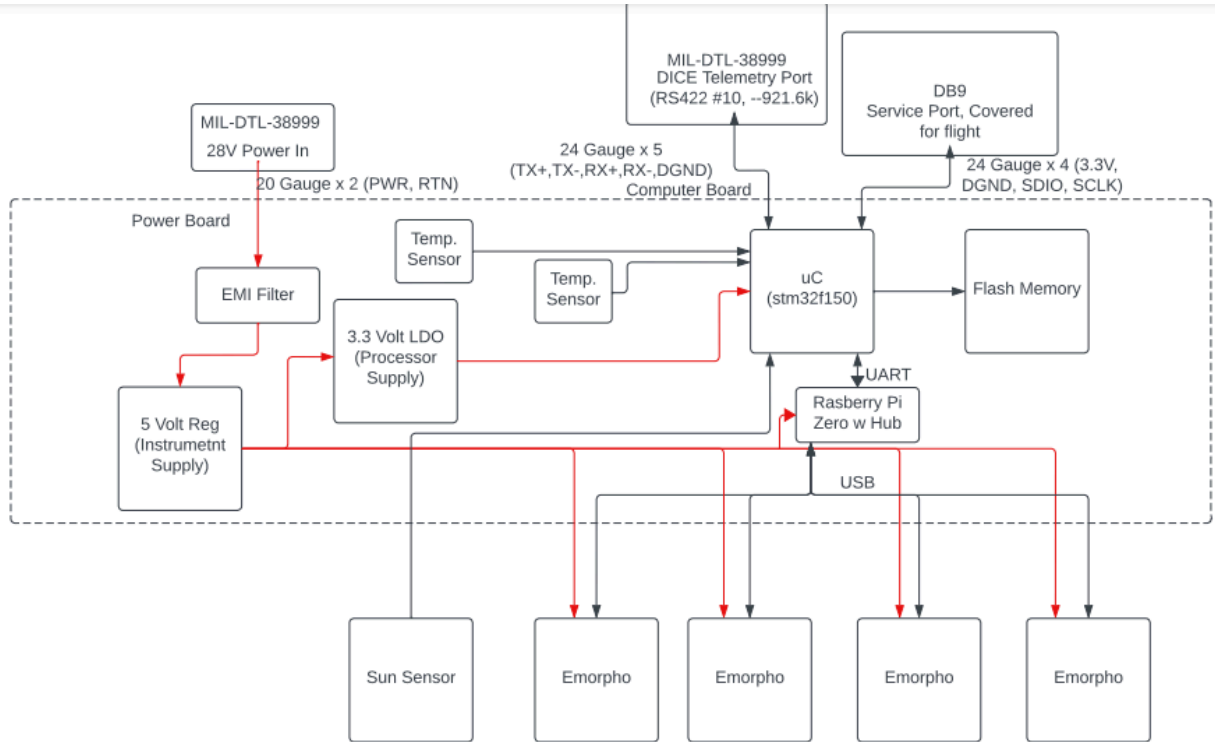


Figure 4 SFXTI Hardware Block Diagram

For SFXTI the computer and power boards from CAPRI-SUN were combined into one board. The fifteen volt regulators were also moved as only a five volt rail was required to run the system. Additionally, since the system would spend around one year in space several engineering sensors were added to increase the information known about the system. Namely voltage and current monitors were added to the five volt rail, several temperature sensors were added to the board and an external sun sensor was added to understand the solar power on the detector. Each of these engineering data sensors were routed to individual ADC capable pins on the microcontroller.

As will be discussed further in the SFXTI firmware design and discussion sections, an issue with the firmware required the addition of a raspberry-pi to the system. Since the computed board was already designed and tested as an interface to the spacecraft, the raspberry-pi simply acted as a middle man between the SiPM-Morphos and the computer board. A UART line was run between the two boards to allow the raspberry-pi to communicate the data to the computer board. In addition to the raspberry-pi a usb hub board, which was installed to the raspberry-pi as a hub, was used to split the USB interface to each of the four instruments.

Firmware Design

In order to meet a testing deadline set by STP for the experiment to spacecraft interface, the communication protocol was the first thing developed for SFXTI. This protocol involved receiving and acknowledging several packet types including Health and Status, Command, Low Rate Telemetry, High Rate Telemetry and GPS all over a single 921600 baud RS-422 port. Each packet type had a specific length and structure as well as specific requirements for the response packets. In order to manage this interface without bogging down the CPU a DMA was used to pass the incoming bytes directly to a circular buffer in memory. Whenever the CPU was free from other tasks it would begin reading the buffer searching for a packet header. The CPU would first compare the pointer to the byte it most recently read to the current DMA write pointer. If the pointers were the same there was no data to be read, if not it would begin checking a single byte at a time. Initially it would simply skip bytes until the first byte of a header was found. It would then check for the next byte of the header, starting over if the byte had the wrong value. Once a valid header was found it would read in the standard packet (device ID and timestamp) then read in the packet type. Once the packet type was received the firmware knows how many total bytes

to read in and how to respond. It would then calculate the cyclic redundancy check, CRC, and compare it to the CRC value received from the spacecraft. Finally it would form the response packet, including a CRC, and send it. Health and Status was a response packet containing engineering data about the system. Command was used to send information to the experiment and only required a simple acknowledgement. Low rate telemetry was a one hertz data request which could be filled with the data of the experimenter's choice. High rate telemetry was similar to the low rate telemetry however, it did not require a request and could be sent at will. Finally the GPS packet contained time and position data and only needed to be acknowledged once at start-up.

As mentioned in the target plan for SFXTI, the original firmware plan was to reuse the code written by the University of Minnesota on the computer board in order to interface with the SiPM-Morphos. This plan did not pan out as the code was heavily reliant on Linux scheduling. The microcontroller on the computer board was not able to run Linux and the necessary changes in order to make the code run on a bare metal system were deemed too time consuming from a development perspective. In order to get around this issue it was decided that a raspberry-pi computer would be inserted between the computer board and the SiPM-Morphos. The raspberry-pi was programmed with C++ primarily using code written by the University of Minnesota. The existing code was designed to read and write to each of the status, control and data registers in the SiPMs. In order to make the system flight ready, the order of messages sent to the instruments had to be designed. On start-up the system first connects the USB system to each device via their static serial number. It then sends a bin mapping to each instrument which acts as an initialization for each device. Next it sends a start command to each device, telling it to

begin taking data. Once all four devices have begun taking data the code begins a round-robin data collection. Starting with the first device it first requests the status register which informs the raspberry-pi as to how many samples have been taken since the last data request. It then requests data from the device according to the number of samples taken. Each time it receives a sample it forms a UART packet which is sent to the computer board. After all samples have been read it moves to the next instrument and repeats the same process. Once all four devices have been read it returns to the first device and starts the full process again. Meanwhile, as packets are received over the UART line by the computer board, the microcontroller saves the data into a buffer to be sent as high rate telemetry packets. Each high rate telemetry packet is large enough to contain four samples and therefore waits to fill the packet before sending.

Support Software

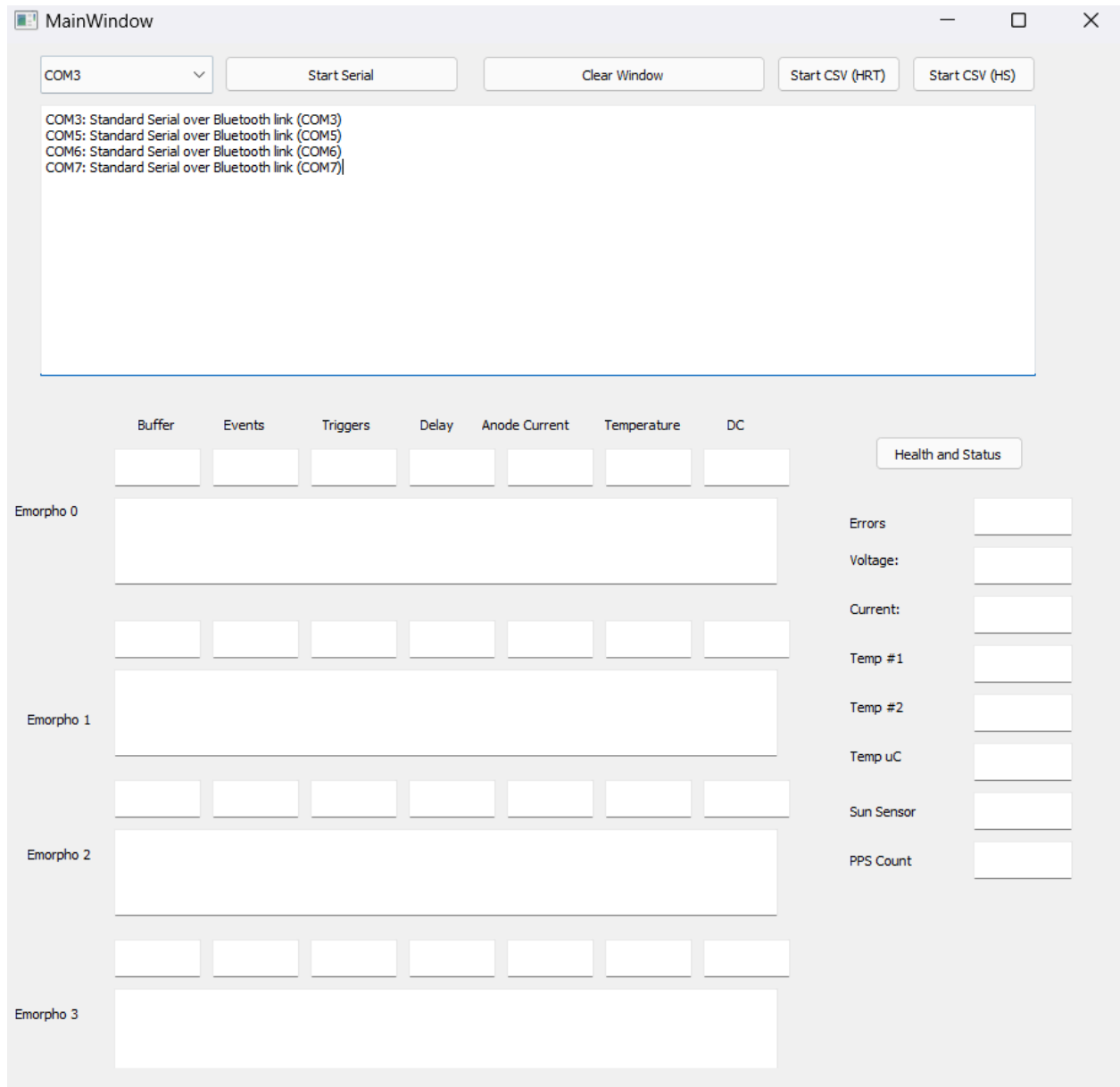


Figure 5 SFXTI Support Software GUI

Support software for the SFXTI instrument was built using the same methodology as the CAPRI-SUN support software. The same basic template was used where all information output by the computer is output to the main window of the GUI. For ease of use, COM port selection

was added to the GUI, as opposed to within the script code itself meaning the script could be converted to an executable file and run on any computer without need to access the code. Similarly to CAPRI-SUN the input packets are processed and displayed at an update frequency which can be read by the user. Status registers are displayed separately for each instrument as well as a single set of bin counts, which are displayed in the larger box. Additionally, since the SFXTI instrument is set up to receive commands, a button was added to request a health and status packet. When the response is received the data requested is displayed in the boxes underneath the request button. Two buttons, shown at the top of the GUI, allow the user to begin saving data and/or health and status packets to their own respective CSV files. While running the script automatically requests health and status packets once per second so the user can plot the status values over time after the fact. This script was used extensively throughout the testing and debugging process for the instrument as well as relied upon for comprehensive performance tests during final integration, thermal and thermal vacuum tests as well as post vacuum test.

RESULTS AND DISCUSSION

Results of the CAPRI-SUN Design

Testing Process

In order to test the systems responsiveness to x-ray radiation an Iron-55 radioactive source was pointed at the detector. This testing process proved more difficult than expected as the detector stack did not seem to respond to the source at the expected distance. One major issue with the use of a radioactive source was knowing the actual power being output by the detector. Although the initial quantity of the source material and the decay rate are known quantities the estimated source strength still had a fair amount of uncertainty associated with it. In order to determine why the source was not producing a response a nano-amp current monitor was placed at the output of the photo-detector. After performing measurements at several distances from the detector it was determined that the source power was as expected. After further inspection of the filter it was realized that an incorrect number had been used for the power of x-rays during solar flares. The value used was around two orders of magnitude larger than the true power and thus our signal amplifiers were not strong enough to properly resolve smaller flares. We would still be able to resolve a bigger flare size if one were present at the time of launch, however, if the launch occurs during a smaller flare no data would be acquired.

One other smaller issue found during the testing process was the presence of a spike at 2.5 volts. It was later realized that this spike was due to a desynchronization between the computer board and the python script when the serial buffer overflowed, causing a header packet to fall in the center of a data packet. These faulty packets can easily be thrown out of the data set.

Additionally, during real conditions when the python script is not being used, the data will be saved without a buffer and therefore this issue will not occur.

Overall the system performed as expected, albeit with significantly less gain than it should have had. The data was acquisitioned at the correct rate and both stored and sent out, the filter was stable and the system survived further thermal-vacuum and vibration testing. The System was packaged up and delivered in August of 2023.

Results of the SFXTI Design

Testing Process

During the initial testing process the system was primarily set up on a bench with no mechanical components. Additionally the scintillator portion of the detector stack was not used. This means although the bins could be read from the SiPM-Morphos, no counts would be present. The system used consisted of a bench power supply, initiating the power supply of the STP spacecraft, the computer board, raspberry-pi which was also connected to a monitor, usb-hub hat, and between one and four SiPM-Morphos connected by USB cables to the hub. During the testing process, several SiPM-Morphos were found to have power draw above the specified level. These devices were set aside for later testing and replaced with backups until all four flight boards were operating within specification.

Once the SFXTI designs was shown to work on the bench several sets of testing had to occur. In order to begin attaching the scintillator to the SiPM-Morphos the system had to be integrated into the mechanical design. This prevented overvoltages from occurring on the detector which could break the system. After the system was integrated and sealed with metal tape the system was tested with several radioactive sources. Counts were registered on the

detectors at higher rates than ambient giving some reasonable confidence that the system was functioning. Next the system had to be de-integrated and subsequently re-integrated with epoxy used to lock all of the bolts into place. After doing so, start-up voltage and current transients were measured to ensure the system met the STP power line requirements. After these tests the system went to thermal-vacuum. After one thermal cycle in the vacuum chamber it was determined that there was not enough time to perform all of the testing in the vacuum chamber as planned. The system was moved to the tenney thermal chamber to perform the remainder of the hot and cold start and survivability tests. Finally the system was brought to Huston for vibration testing and delivery. After each of the tests a comprehensive performance test was run on the system to ensure proper operation and power margins.

CONCLUSIONS AND RECOMMENDATIONS

Computer Design

Shortcomings

The Computer design was successful overall, however, it did have several clear issues. Firstly, for the CAPRI-SUN project, no engineering sensors, such as voltage, current or temperature, were used which could have been advantageous from a debugging perspective should anything go wrong after integrations. This was fixed for the SFXTI project, however, the SFXTI computer board had a much more impactful issue, in that the USB port was not capable of performing complex operations. The USB port on the STM32 microcontrollers is designed primarily for use as a slave device or for communication with other STM32 microcontrollers. The driver library is otherwise lacking for this port and would require extensive work in order to be used as a master device. Additionally, since the STM32 does not have an easy way to run linux, it can be difficult, as it was in the case of SFXTI, to integrate firmware which is designed for use with an operating system.

Advantages

The major advantage brought by the STM32 based computer board is ease of development. The development environment provided by STMicroelectronics has a graphical interface for selecting pin functionality, creating timers, as well as setting up the clock dividers which then generates the majority of the setup code for the user. This allows the developer to spend less time pouring through the data sheet and deciphering the initialization files. The developer can instead move straight to developing the C code, resembling a software

implementation one would see on a typical computer. This graphical interface can even be used during the schematic design process to ensure the pinout designed in hardware is compatible with the microcontroller's functionality.

Future Projects

Each of the Projects discussed in this paper have a future project based on the same instrument. For CAPRI-SUN and a project called the student-built ultraviolet burst observer, BUBO, will use the same photodetectors, amplifier stages and analog converter with more channels implemented with different filters allowing for different photo energy ranges to be detected. The impulsive phase rapid energetic solar spectrometer, or IMPRESS, will use the same instrument as SFXTI, along with the computer designed by the University of Minnesota, with several different filters allowing it to measure x-ray counts across a larger range of powers.

BUBO

For the BUBO project several changes should be made to the analog filters. Having a lower gain in the amplification stage is advantageous because it contributes to a higher signal to noise ratio. The ADC used can be set to measure a smaller input range, specifically plus and minus forty mili-volts, as opposed to the plus and minus two and a half volts used for CAPRI-SUN. This means the gain can be significantly decreased on the amplifier. Additionally, the gains can be set correctly based on the true solar irradiance values, correcting the sensitivity error in the CAPRI-SUN design.

IMPRESS

The impressive design, being based on the University of Minnesota computer, cannot gain much insight from the SFXTI design. The majority of the new designs for SFXTI are specific to the STP requirements and therefore do not apply to IMPRESS. WE can, however, learn for future designs which require complex USB interfaces, that using a linux based computer can significantly decrease the work required.