

Dissertations and Theses

4-2019

Planar Ion Probe for Low-Latitude Ionosphere/Thermosphere Enhancements in Density Cubesat Mission

Liam Owen Gunter

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Engineering Physics Commons](#)

Scholarly Commons Citation

Gunter, Liam Owen, "Planar Ion Probe for Low-Latitude Ionosphere/Thermosphere Enhancements in Density Cubesat Mission" (2019). *Dissertations and Theses*. 445.

<https://commons.erau.edu/edt/445>

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

PLANAR ION PROBE FOR LOW-LATITUDE
IONOSPHERE/THERMOSPHERE ENHANCEMENTS IN DENSITY
CUBESAT MISSION

BY
LIAM OWEN GUNTER

A Thesis
Submitted to the Department of Physical Sciences
In partial fulfillment of the requirements
for the degree of
Master of Science in Engineering Physics

04/2019
Embry-Riddle Aeronautical University
Daytona Beach, Florida

© Copyright by Liam Owen Gunter 2019
All Rights Reserved

PLANAR ION PROBE FOR LOW-LATITUDE
IONOSPHERE/THERMOSPHERE ENHANCEMENTS IN DENSITY
CUBESAT MISSION

by

Liam Owen Gunter

This thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Aroh Barjatya, Associate Professor of Engineering Physics, Daytona Beach Campus, and has been approved by the Thesis Committee. It was submitted to the Department of Physical Sciences in partial fulfillment of the requirements of the degree of
Master of Science in Engineering Physics

THESIS COMMITTEE:




Dr. Aroh Barjatya,
Committee Chair



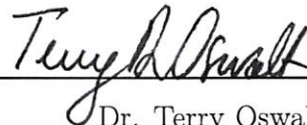
Dr. Bereket Berhane,
Committee Member



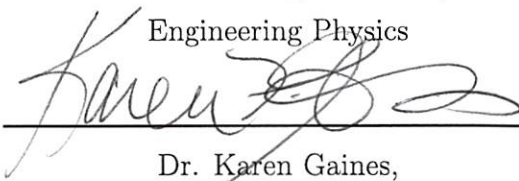
Dr. Charles Lee,
Committee Member



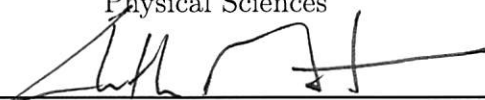
Dr. Matthew Zettergren,
Graduate Program Chair,
Engineering Physics



Dr. Terry Oswalt,
Department Chair,
Physical Sciences



Dr. Karen Gaines,
Dean, College of Arts and Sciences



Dr. Christopher Grant,
Vice Provost of Academic Support

Abstract

One of the crucial measurements for characterizing any space weather event is absolute plasma density and plasma density fluctuations, both spatially and temporally. Langmuir probes are the oldest and most proven instruments for these in-situ measurements. This thesis enumerates the development of a miniaturized low-noise Langmuir probe intended for a dual CubeSat mission to study equatorial temperature and wind anomaly in the Earth's ionosphere.

The Langmuir probe instrument developed is of a planar geometry and fix biased in the ion saturation region, i.e. negative w.r.t. spacecraft chassis. Operating the Langmuir probe in the ion saturation region avoids excessive spacecraft charging on small spacecraft platforms, while also avoiding high-risk deployables. Specific emphasis is placed on minimizing the physical footprint, power consumption, and measurement noise levels of the instrument, all while maintaining high measurement cadence. In this way, the device is also intended to be functionally dynamic and easily modifiable for future missions requiring similar instrumentation.

The effort toward this thesis included circuit design and simulation in National Instruments' Multisim, printed circuit board layout design in National Instruments' Ultiboard, instrument firmware development in Texas Instruments' Code Composer Studio, mechanical design of the system in Dassault's CATIA and SolidWorks, test article board population, extensive instrument testing, and temperature calibration in a TestEquity thermal chamber.

Acknowledgments

The work enumerated here was conducted between May 2017 and May 2019, under the direction of Dr. Aroh Barjatya. While he is a very challenging professor, I couldn't have asked for a better, more capable, more dedicated research advisor and mentor. My parents Bud and Jeanene, brother Benjamin, and grandparents Dora and Doyle Gunter were also steadfast sources of encouragement and comfort throughout the project, and thus have my undying appreciation. An additional thanks goes to Austin Ogle for his contributions to writing the instrument firmware for testing and developmental purposes, as well as troubleshooting assistance throughout a large portion of the project. Nicholas Purvis with his technical advice and hands-on help with circuitry, especially difficult soldering jobs, was an invaluable resource throughout testing. Kyle Hrenyo and his temperature chamber work also made a smooth survival testing and calibration happen. I would also like to thank Miguel Martínez and Marcos Diaz from the University of Chile for comparative instrument design discussion and correspondence. The whole team from the Aerospace Corporation I interfaced with, in particular Herach Ayvazian, also have my thanks. The faculty throughout my academic career, especially the Department of Physical Sciences, are too innumerable to mention here, but their academic contributions are valued more than they know. Additionally, Susan Adams deserves high praise for her overwhelming behind-the-scenes work for the researchers, including myself, in our Department.

I also greatly appreciate the financial support from the Embry-Riddle Office of Undergraduate Research, and their funding that brought my research to the 2018 Fall Meeting of the American Geophysical Union.

My sincerest thanks to you all!

Nomenclature

ADC Analog to Digital Converter

CFR Configuration Register

CONOPS Concept of Operations

CRC Cyclic Redundancy Check

DICE Dynamic Ionosphere CubeSat Experiment

DMM Digital Multimeter

EIA Equatorial Ionization Anomaly

EMI Electromagnetic Interference

ETWA Equatorial Temperature and Wind Anomaly

eUSCI Enhanced Universal Serial Communication Interface

GPIO General Purpose Input/Output

IC Integrated Circuit

IT Ionosphere-Thermosphere

IV Current-Voltage

LDO Low Dropout

LEO Low Earth Orbit

LLITED Low Latitude Ionosphere/Thermosphere Enhancements in Density

MCU Microcontroller

MISO Master In Slave Out

MMCX Micro-Miniature Coaxial

MOSI Master Out Slave In

MTeX Mesosphere Turbulence Experiment

PCB Printed Circuit Board

PDR Preliminary Design Review

PIP Planar Ion Probe

SAIL Space and Atmospheric Instrumentation Laboratory

SAR Sequential Approximation Register

SBW Spy-Bi Wire

SNR Signal to noise ratio

SPI Serial Peripheral Interface

SWaP Size Weight and Power

UART Universal Asynchronous Receiver-Transmitter

Contents

Abstract	iv
Acknowledgments	v
Nomenclature	vi
1 INTRODUCTION	1
1.1 LLITED Mission/Requirements	2
1.2 Langmuir Probe Background/Historical Survey	6
2 PLANAR ION PROBE DESIGN	16
2.1 General System Architecture	16
2.2 Low Noise Design Practices	21
2.2.1 Trace Design	22
2.2.2 Power Supply Decoupling	22
2.2.3 Analog-Digital Separation	23
2.2.4 Voltage Planes	23
2.2.5 Mechanical Design Considerations	24
2.2.6 Material Properties	25
2.3 Detailed Design	25
2.3.1 Analog Signal Flow/Processing	26
2.3.2 Housekeeping	38
2.3.3 Software Architecture	46
2.4 Mechanical Design	52

3	TESTING	57
3.1	Multisim Simulation Analysis	57
3.2	Low Noise Testing	59
3.2.1	v1.3 Oscilloscope/Multimeter Noise Analysis	60
3.2.2	v2.0, v2.1 and v3.0 Low Noise Verification	63
3.3	Power Consumption Characterization	64
3.4	Voltage Supply Permutation Survival/Inrush Current Testing	68
3.5	Temperature Survival Testing	71
4	CALIBRATION	75
4.1	Environment Chamber Temperature Calibration	75
5	CONCLUSIONS	82
A	Circuit Schematic	86
B	Circuit Layout	91
C	Sensor Layout	95
D	Mechanical Design	100
E	Instrument Firmware	106
F	Testing Interface Software	117
G	Calibration Software	128
H	Calibration Data	140
H.1	0b000 Calibration Coefficients	140
H.1.1	Bias Calibration	140
H.1.2	Instrument Calibration	140
H.2	0b001 Calibration Coefficients	141
H.2.1	Bias Calibration	141
H.2.2	Instrument Calibration	141

H.3	0b010 Calibration Coefficients	142
	H.3.1 Bias Calibration	142
	H.3.2 Instrument Calibration	142
H.4	0b011 Calibration Coefficients	143
	H.4.1 Bias Calibration	143
	H.4.2 Instrument Calibration	143
H.5	0b100 Calibration Coefficients	144
	H.5.1 Bias Calibration	144
	H.5.2 Instrument Calibration	144

List of Tables

1.1	PIP parameters and requirements at project beginning and end [Bishop et al., 2016].	5
2.1	PIP connector pinout and descriptions.	21
2.2	PIP 0b000 light sensor calibration.	46
2.3	PIP command dictionary.	49
2.4	PIP data packet format, with appended CRC-16 bits.	49
3.1	PIP power consumption for each supply and total for selected loads. .	67
3.2	PIP inrush current, steady-state current, and not to exceed inrush duration for recommended power sequence.	71

List of Figures

1.1	Data from the CHAMP accelerometer (top panel; 10^{-12} kg/m ³), planar Langmuir probe (middle panel; 10^{12} m ⁻³), GPS RO sensor (bottom panel; TECU). Crosses and dashes mark the location of crests and troughs, respectively. Adapted from Figures 1, 3, 6 of Lei et al. [2010].	3
1.2	One day (14 October 2005) of measurements of IGS data from STREAK mission superposed on data from the GUVI imager on the TIMED spacecraft [Clemmons et al., 2013].	3
1.3	Generic Langmuir probe curve [Barjatya, 2007].	8
1.4	Langmuir probe operating regimes and geometry dependence [Barjatya, 2007].	9
1.5	Portrayal of PIP sensor collecting current from ambient plasma on LLITED ram face.	12
1.6	Langmuir probe electrometer circuitry block diagram.	13
1.7	LLITED satellite configurations after launcher separation (solar panels stowed) and during nominal science mission activities (solar panels deployed).	15
2.1	PIP subsystem assembly, including the sensor PCB and instrument PCB, in relation to the LLITED ram plate.	17
2.2	PIP instrument/sensor board MMCX interface connectors [Amphenol Connex, 2003, 2006].	18
2.3	PIP hardware block diagram, showing signal flow and how it manifests on the physical v2.1 instrument.	19
2.4	PIP off-board connector pinout cartoon [Samtec].	20

2.5	Accepted PCB trace corner format.	22
2.6	PIP analog section, displaying all four layers of bias planes.	24
2.7	PIP inner copper 2 layer, with ground plane extent annotated.	24
2.8	PIP transimpedance amplifier subsystem circuit schematic.	28
2.9	PIP instrumentation amplifier subsystem circuit schematic.	30
2.10	PIP filter/diode protection subsystem circuit schematic.	32
2.11	PIP ADC interfaced with MCU subsystem circuit schematic.	34
2.12	ADS8353 conversion data read protocol timing diagram [Texas Instru- ments, 2014a].	35
2.13	ADS8353 CFR register settings information [Texas Instruments, 2014a].	36
2.14	ADS8353 REFDAC register settings information, applicable to both REFDAC A/B [Texas Instruments, 2014a].	37
2.15	PIP temperature sensor housekeeping circuit schematic.	39
2.16	PIP temperature sensor housekeeping Multisim simulation.	39
2.17	PIP +5 V/+15 V housekeeping circuit schematic.	40
2.18	PIP +5 V housekeeping Multisim simulation.	41
2.19	PIP +15 V housekeeping Multisim simulation.	42
2.20	PIP -15 V housekeeping circuit schematic.	43
2.21	PIP -15 V housekeeping Multisim simulation.	44
2.22	PIP light sensor housekeeping circuit schematic.	45
2.23	PIP light sensor housekeeping Multisim simulation.	47
2.24	PIP state machine, displaying the command response algorithm paths.	48
2.25	PIP interrupt and sample measurement timing diagram showing the sequence of events for command and data transfer (measure command shown).	50
2.26	Measurement command and instrument response, captured using an oscilloscope.	51
2.27	Error command and instrument response, captured using an oscillo- scope.	51

2.28	Shutdown command and instrument response, captured using an oscilloscope.	52
2.29	PIP’s finalized envelope design, acting as a protective Faraday enclosure for the instrumentation.	53
2.30	PIP sensor board interconnect between MMCX and copper planes magnified, copper bottom left and copper top right.	53
2.31	PIP sensor board assembly photograph (left is top and right is bottom), finalized for vibration testing.	54
2.32	The LLITED ram plate with feedthrough detailed, for passing through the MMCX signal connection for PIP and easing clearance concerns on the AD549.	55
3.1	PIP Multisim simulation confirming intended signal flow.	58
3.2	PIP Multisim AC frequency simulation, showing dB magnitude and phase in ADC signal.	58
3.3	PIP Multisim simulation varying power supply offsets from nominal voltage.	59
3.4	PIP noise characterization testing with oscilloscope/multimeter block diagram.	60
3.5	PIP v1.3 oscilloscope measured noise versus frequency data.	61
3.6	PIP v1.3 DC signal characterization.	62
3.7	PIP v1.3 AC noise characterization.	62
3.8	PIP noise characterization testing with oscilloscope/multimeter block diagram.	63
3.9	PIP v3.0 0b000 instrument DC characterization in terms of input current.	64
3.10	PIP v3.0 0b000 low gain noise in terms of input current.	65
3.11	PIP v3.0 0b000 high gain noise in terms of input current.	65
3.12	PIP power consumption testing block diagram.	66
3.13	PIP power consumption, given input load signal current.	67
3.14	PIP inrush current testing block diagram.	69

3.15	PIP +15 V supply inrush current.	69
3.16	PIP -15 V supply inrush current.	70
3.17	PIP +5 V supply inrush current.	70
3.18	PIP survival testing block diagram.	72
3.19	PIP survival test theoretical temperature profile.	73
3.20	PIP survival test temperature data from the first successful test. . .	74
4.1	PIP bias calibration testing block diagram.	76
4.2	PIP bias calibration fit error on board #0b100 instrument.	77
4.3	PIP instrument calibration testing block diagram.	78
4.4	PIP calibration fit error on board #0b000 instrument low gain channel.	79
4.5	PIP calibration fit error on board #0b000 instrument high gain chan- nel.	79
4.6	PIP calibration fit error on board #0b000 instrument full low gain channel.	80
4.7	PIP calibration fit error on board #0b000 instrument full high gain channel.	81

Chapter 1

INTRODUCTION

In the hierarchy of Earth's atmospheric layers, the ionosphere is an ionized region generated via solar radiation that spans the upper regions of the mesosphere, the entirety of the thermosphere, and into the lower regions of the exosphere. Quantitatively this translates to a region approximately 80 km to 1000 km in altitude above Earth's surface. The ionosphere forms the conductive, ionized half of the linked Ionosphere-Thermosphere (IT) system, and is bounded at its farthest space-ward extent by the magnetosphere. The ionosphere contains distinctly unique environments in its polar, equatorial, and mid-latitude zones. While the polar ionosphere is host to well known auroral activity that is visible to the naked eye, the low-latitude equatorial ionosphere is also host to crucial space weather phenomena albeit that is invisible to the naked eye. For example, the low-latitude region is home to the Equatorial Ionization Anomaly (EIA), a localized increase in plasma density seen on either side of the magnetic equator stemming from $\mathbf{E} \times \mathbf{B}$ plasma drift, as well as plasma bubbles embedded in the EIA which lead to scintillations in any radio signals being transmitted through the ionosphere. These scintillations have direct effect to quality of life on earth, most critically disrupting the reception of GPS signals. All of these various phenomena are investigated in-situ by rockets and satellites to help us better understand, model, and predict space weather.

Low-Latitude Ionosphere/Thermosphere Enhancements in Density (LLITED) CubeSat mission is a NASA funded project to investigate and better understand the equatorial ionosphere. This chapter first presents a brief introduction to the LLITED mission's motivation, requirements, and mission design. We then present theoretical background and historical context of Langmuir probe theory and application, as well as a brief introduction to the architecture of the Planar Ion Probe (PIP), Embry-Riddle's contribution to LLITED CubeSat mission for measuring absolute ion density. The detailed design, testing, and calibration are the subject of this thesis.

1.1 LLITED Mission/Requirements

Coupling between Earth's Ionosphere-Thermosphere regions has recently been observed to be more complex than previously expected. IT interactions in the low-latitude dusk-side manifest through two dominant phenomena: the Equatorial Ionization Anomaly (EIA) and the Equatorial Temperature and Wind Anomaly (ETWA), shown in Figures 1.1 and 1.2. The EIA has been extensively observed and modeled, while the ETWA is relatively unknown. It is suspected that there exists a causal link between the two phenomena, but current models struggle to close the problem without a large supply of coincident data pertinent to both the EIA and ETWA.

The Low Latitude Ionosphere/Thermosphere Enhancements in Density mission, a NASA funded grant contracted to The Aerospace Corporation, intends to provide the first coincident measurements of both neutral and ion density in the low-latitude region of the 400 km ionospheric region. This is done in an effort to fill the gap present in current ETWA observational data.

The lack of data on the ETWA is due mainly to poor coverage of the region with properly instrumented spacecraft. Specifically, appropriate ETWA observation should nominally involve both neutral (thermosphere) and plasma (ionosphere) measurements. LLITED thus intends to be the first mission to collect simultaneous neutral/ion data pertinent to IT interactions at lower altitudes, expanding the knowledge base regarding the ETWA. As such, the mission will collect neutral density, ion density, and total electron content profiles of the low-latitude IT region.

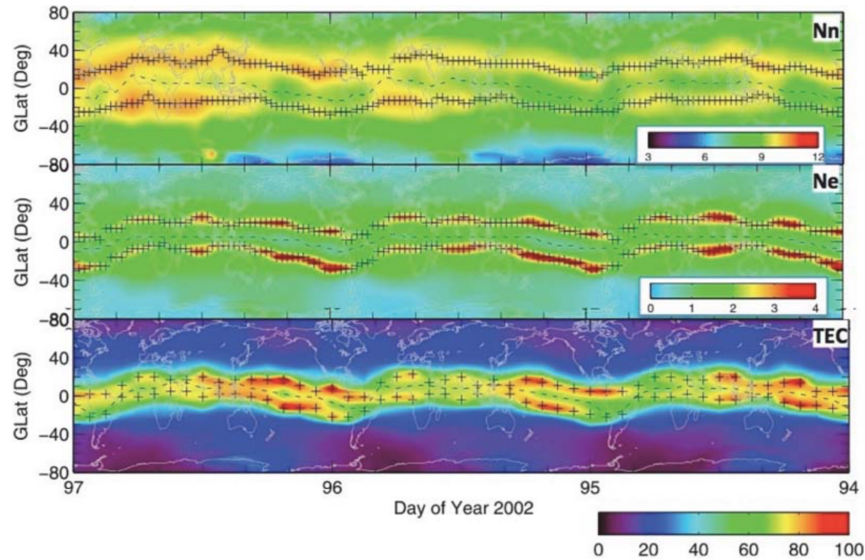


Figure 1.1: Data from the CHAMP accelerometer (top panel; 10^{-12} kg/m^3), planar Langmuir probe (middle panel; 10^{12} m^{-3}), GPS RO sensor (bottom panel; TECU). Crosses and dashes mark the location of crests and troughs, respectively. Adapted from Figures 1, 3, 6 of Lei et al. [2010].

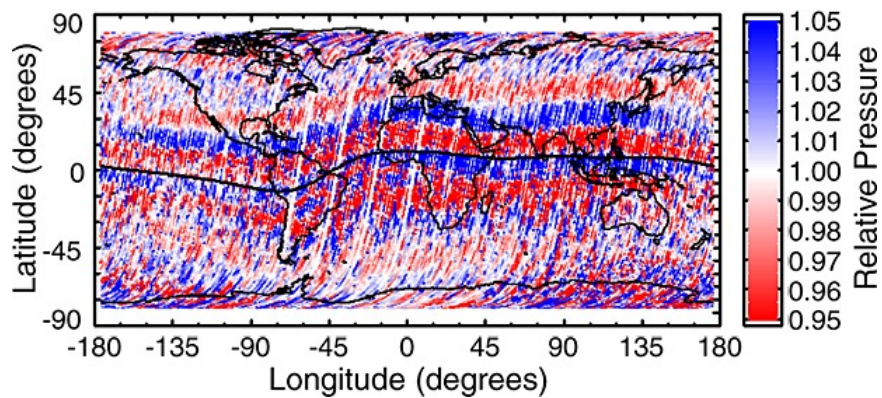


Figure 1.2: One day (14 October 2005) of measurements of IGS data from STREAK mission superposed on data from the GUVI imager on the TIMED spacecraft [Clemmons et al., 2013].

LLITED addresses these questions through its suite of three payloads: neutral density measuring ionization gauge sensor MIGSI, ion density measuring Langmuir probe PIP, and total electron content measuring radio occultation sensor CTECS. The LLITED CubeSat bus, MIGSI, and CTECS are being built by the Aerospace Corporation, whereas PIP is being provided by Embry-Riddle’s Space and Atmospheric Instrumentation Laboratory (SAIL). The mission design consists of twin 1.5U CubeSats expected to be launched in a polar 350 – 450 km low Earth orbit (LEO) in 2020. This orbit provides global coverage of low-latitude ETWA regions through the three instruments. Observations of the low-latitude regions specifically necessitate that the instruments only operate 25% – 45% of the orbital period. Additionally, the dual satellites travel with nominally 1/2 an orbit separation, additionally allowing for temporal observations of ETWA.

LLITED directly addresses a key science goal from the Heliophysics Decadal Survey: “Determine the dynamics and coupling of Earth’s magnetosphere, ionosphere, and atmosphere and their response to solar and terrestrial inputs.” LLITED will characterize and improve our understanding of the ETWA, provide insight into the coupling physics between the ETWA and the EIA, and increase our knowledge of the dusk-side dynamics that may influence space weather. LLITED will make progress towards answering three distinct questions:

1. What is the mesoscale variability of the ETWA as a function of season, and longitude/latitude as well as its relationship to EIA heating?
2. What is the relationship between neutral wind (i.e. tides) and the EIA zonal structure?
3. Are the small-scale wave fluctuation in neutral atmosphere quantities, such as those observed by earlier missions exhibited in the ionospheric density?

In-situ ion density measurements are provided by Embry-Riddle’s PIP instrument. Due to its intended CubeSat platform, PIP is an ultra low size, weight, and power (SWaP) electrometer built on a platform of two populated printed circuit boards (PCBs) equipped with a planar Langmuir probe sensor. Its simple fixed-bias system

enables low-noise measurements of absolute in-situ ion density without the side effects associated with a sweeping probe on a small platform. If needed in the future, the design is modular enough that it can be deployed as a patch on multiple faces of a CubeSat, further reducing attitude control requirements and enabling the study of the wake structure around the spacecraft.

Parameter	At Proposal	At PDR
Performance	$2 \times 10^9 \text{ m}^{-3}$ to $2 \times 10^{13} \text{ m}^{-3}$	
Resolution	$2 \times 10^8 \text{ m}^{-3}$	
Sample Rate	1 Hz	100 Hz
Heritage	DICE CubeSat, MTeX Rockets	
TRL	4	5
Cell Size	Mission Op.: 1 km	$\sim 80 \text{ m}$
Voltage	+5 V	+5 V, $\pm 15 \text{ V}$
Average Power	250 mW	125 mW mid-density, 250 mW max
Mass	$\sim 220 \text{ g}$	50 g (PCBs, fasteners, and envelope)
Volume	Board: $\sim 10 \times 10 \times 1 \text{ cm}$ Plate: $\sim 5 \times 6 \times 0.5 \text{ cm}$	Board: $7.02 \times 2.4 \times 1.1265 \text{ cm}$ Plate: $6.6 \times 4.6 \times 0.16 \text{ cm}$
Duty Cycle	60%	
Data Rate	48 b/s	4800 b/s

Table 1.1: PIP parameters and requirements at project beginning and end [Bishop et al., 2016].

Table 1.1 provides a summary of PIP parameters and requirements at the time of project proposal and again at Preliminary Design Review (PDR). From this simple comparison, PIP’s design changes throughout the course of the project and low SWaP characteristics are easily visible.

The dual nature of the LLITED satellites allows for observations of temporal evolution in the EIA and ETWA phenomena with $\sim 45 \text{ min}$ resolution. After nominal launcher separation, CubeSat A will deploy its stowed solar panels and induce greater drag than CubeSat B for approximately a month. Figure 1.7 depicts these stowed and deployed configurations. The spacecrafts’ mutual separation will increase throughout this duration until they are 180° in phase, at which time CubeSat B will deploy its solar panels. Nominal science mission operations then proceed following these events.

As per the mission concept of operations (CONOPS), when in science mode the

PIP instrument will be polled at 100 Hz. The instrument will determine ion density between $2 \times 10^9 \text{ m}^{-3}$ to $2 \times 10^{13} \text{ m}^{-3}$, at a minimum resolution of 10% accuracy. Thus, at the lowest end of its dynamic range, PIP is able to detect variations on the order of $2 \times 10^8 \text{ m}^{-3}$. Absolute ion density measurements are crucial to the success of the LLITED mission in characterizing the ETWA. We next give a brief background of Langmuir probe theory and how it applies to the design and implementation of PIP.

1.2 Langmuir Probe Background/Historical Survey

Electric probes, in the form of exposed metal electrodes immersed in plasma, have been in use for just under a century to investigate the physical properties of plasma Langmuir and Mott-Smith [1924]. Eponymously named due to their first use by Irving Langmuir in 1924, Langmuir probes are ubiquitously used in studies of plasma, both in laboratory setting and in space. They are flown on almost every geospace sounding rocket as well as a large number of satellites as a means to do in-situ measurements of plasma density and temperature. The Langmuir probe instrument designs differ in number and geometry of electrodes, electrode bias potential, and time varying bias on the electrodes. These different modes and implementations allow measurements of ion density, electron density, and electron temperatures.

In the simplest sense, Langmuir probe operating principle is application of a bias (referencing spacecraft chassis ground) on a probe immersed in plasma and collection of the resulting current. Conventionally, current sourced ‘from the probe into the plasma’ is considered positive, while current sunk ‘from the plasma into the probe’ is negative. As the direction of current is the direction of ion movement, ion collection current by the probe is negative and electron collection current is positive. The collection current is a summation of several different sources. These include plasma thermal current, vehicle ram velocity current, photoelectron emission current, and other currents associated with myriad minor plasma effects. From the resulting current-voltage (IV) characteristic the physical plasma parameters such as density and temperature are ascertained.

In the absence of any photoelectron current emission and energetic particle current

collection, the largest component of collection current by a stationary electric probe is the thermal plasma current. Assuming a Maxwellian distribution of velocities for plasma of singly charged ions and electrons of density n , the thermal current collection per species j to probe of area A_{probe} at plasma potential is found to be

$$I_{thermal,j} = n_j q_j A_{probe} \sqrt{\frac{k_B T_j}{2\pi m_j}}, \quad (1.1)$$

where the thermal current is dependent upon the species temperature T_j and particle mass m_j , and is scaled by the Boltzmann constant k_B . It is important to note that the “plasma potential” is not an assumed or equilibrium position, but rather is the point where no external fields exist between the probe surface and the bulk plasma and all thermal particles hitting the probe are collected by the probe. Thermal electron collection current is larger than thermal ion collection current due to the lighter mass of the electrons and consequently higher electron thermal velocity. Thus, at the plasma potential the current collection is positive. If the probe is left to ‘float’ then the surface eventually achieves a net negative voltage such that some of the thermal electron current is repelled until the electron and ion collection current magnitudes are equal and the net current is zero (i.e. equal negative and positive charges are collected). This potential is called as the floating potential [Barjatya, 2007].

The floating potential and plasma potential break the IV curve into three distinct regimes: the ion saturation region where the probe bias is significantly negative relative to the floating potential, the electron saturation region where the probe bias is more positive than plasma potential, and the electron retardation region where the probe bias is between the floating potential and plasma potential. Figure 1.3 illustrates these regions and their potential positions within a Langmuir probe IV curve.

In the retardation region, where the probe potential is just slightly negative compared to plasma potential, the probe repels electrons and attracts ions. Yet the current is still positive because the electron collection current continues to be larger than ion collection current owing to electron’s lower mass and higher thermal velocity

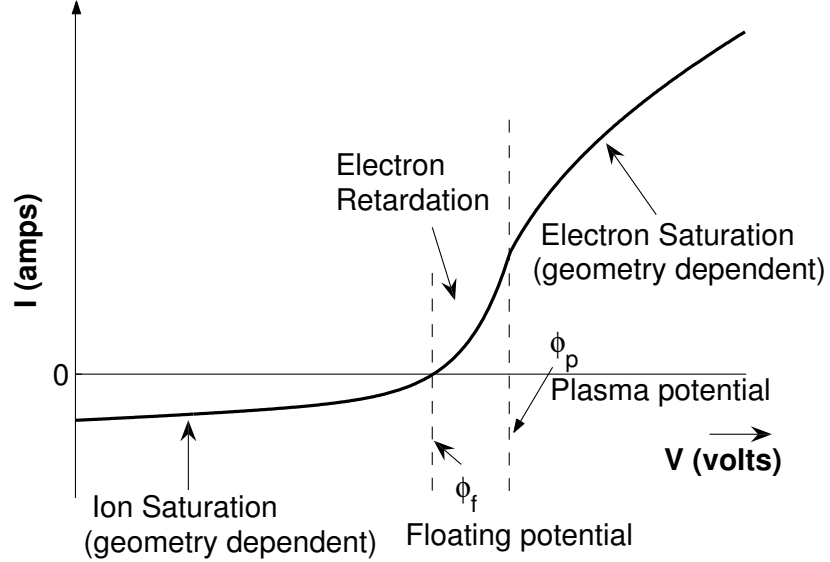


Figure 1.3: Generic Langmuir probe curve [Barjatya, 2007].

compared to ions. Thus, the retardation region current collection is related to the average kinetic energy of the electrons, which is the definition of electron temperature. In other words, analysis of the IV curve in the electron retardation region helps ascertain electron temperature. The relationship governing probe current in this electron retardation region is

$$I_e(\phi) = I_{thermal,e} \exp\left(\frac{e(\phi - \phi_p)}{k_B T_e}\right), \quad (1.2)$$

where ϕ is the potential relative to the plasma potential ϕ_p , e is fundamental electron charge, and T_e is electron temperature [Barjatya, 2007].

The ion and electron saturation regions are geometry dependent. For cylindrical and spherical probe geometries the collection current in the saturation regions increases with applied voltage, whereas for a planar geometry the saturation current is flat. This is shown in Figure 1.4. Besides being dependent on the applied potential, the saturation region current is also directly proportional to the ambient plasma density. Thus, analysis of the saturation regions help ascertain plasma density.

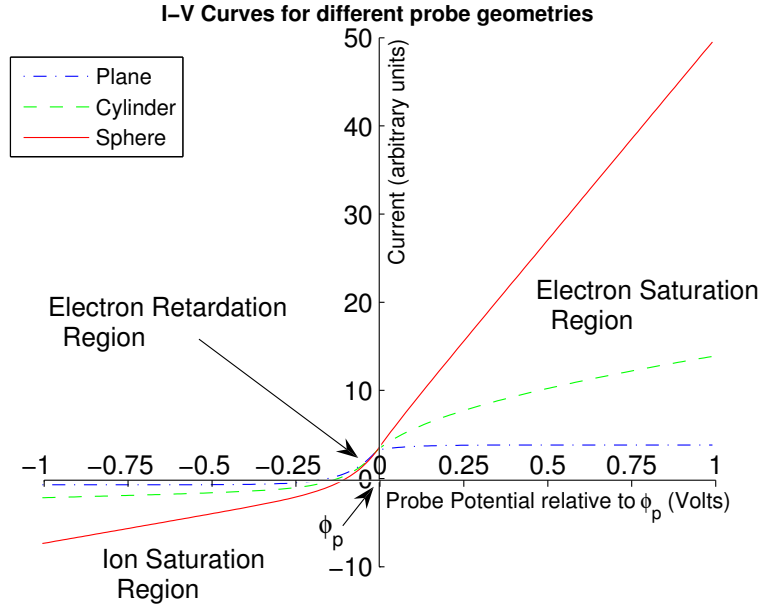


Figure 1.4: Langmuir probe operating regimes and geometry dependence [Barjatya, 2007].

The saturation region current collection equation is given by

$$I_j(\phi) = I_{thermal,j} \left(1 + \frac{q_j(\phi - \phi_p)}{k_B T_j} \right)^\beta, \quad (1.3)$$

where β is probe geometry dependent factor. $\beta = 1/2$ and $\beta = 1$ denote cylindrical and spherical probes, respectively [Barjatya, 2007]. $\beta = 0$, representing flat plate probes, denotes that ion and electron saturation region current collection asymptotically approaches a constant value with increasing magnitude of the absolute voltage. This constant thermal current property in the ion saturation region is used to great effect in the LLITED PIP instrument, as detailed later below.

From these Equations 1.1, 1.2, and 1.3, it is clear that given a Langmuir probe with known geometry, a distinctive and predictable IV curve would be generated from sampling the probe current at a variety of potentials. Plasma temperature and density can then be determined via analysis of this curve. To achieve this variation in potential, probes are often swept rapidly through many voltages up and down.

It is important to note though that the probe, spacecraft and plasma form a single electrical circuit. As the probe sweeps positive to collect more and more electrons, the spacecraft chassis ground has to collect equal number of ions. But as the ion thermal current is much smaller than the electrons, the chassis is unable to collect the requisite ion current. As a result of this excess in electron collection, the whole spacecraft starts charging negative. Or in the other words, the electrical ground of the spacecraft becomes unstable as the Langmuir probe sweeps positive. According to Szuszczewicz [1972], for a sweeping probe to avoid upsetting the spacecraft's floating potential and affecting other electronics and instruments on-board the spacecraft, the ratio of vehicle chassis exposed area to biased probe exposed area must exceed 10,000 : 1. For vehicles like large spacecrafts and somewhat less so for sounding rockets, this requirement is trivial to satisfy. However, for small satellites like CubeSats, this requirement is very difficult to satisfy while retaining a high SNR. With a limited spacecraft chassis surface area on the CubeSat, on the order of hundreds of square centimeters (that is, if the *entire* surface is conductively finished), this necessitates a sensor probe area of *hundredths of a square centimeter*. This small of a probe area would lead to an abysmally low SNR due to contributions of other noise effects in the circuitry and sensor design. Thus, a sweeping Langmuir probe implementation on a CubeSat is not possible.

A probe fixed bias in ion saturation region effectively removes this spacecraft charging problem, and allows for equipping the spacecraft with a large sensor to retain high SNR. That is because a large ion collection current by the Langmuir probe is easily matched by equivalent electron collection by the spacecraft chassis. PIP was thus designed with a fixed-bias system. PIP holds its constant bias via a reference integrated circuit (IC) well into the ion saturation region, at -7 V w.r.t. to the spacecraft chassis which is itself at the floating potential.

For an ionospheric plasma temperature of 1000 K ($T_i = T_e$), thermal velocity of ions is 290 m/s and that of electrons is 49,000 m/s. The ion velocity is much smaller than the spacecraft orbital velocity of 8000 m/s. This situation of $V_i < V_{sc} < V_e$ is called as a mesothermal plasma current collection. Under this situation, in addition to the ion thermal current collection, we also have to take into account the ion ram

velocity current. This current, which represents a probe sweeping up ions in its path as it moves at a speed faster than thermal velocity, is given by a simple linear relation

$$I_{ram,i} = n_i q_i u A_{cross} = n_i q_i u \cos \theta A_{probe}, \quad (1.4)$$

where n_i is ion density, q_i is ion charge, u is spacecraft orbital velocity, and A_{cross} is the projected cross-sectional area that the probe of area A_{probe} forms with the ram direction when its surface normal is offset from ram direction by angle θ .

As the instrument name implies, Planar Ion Probe (PIP) is a flat plate probe collecting ion current. A conducting, planar rectangle of hard gold is mounted externally on the ram face of the spacecraft. In addition to a flat plate Langmuir probe experiencing a nearly constant asymptotic ion saturation current due to the fixed-bias given by equation 1.3, the flat plate directed specifically in the ram velocity direction also collects current given by equation 1.4. Figure 1.5 provides an illustration of the operating principle of the PIP instrument that will be detailed in the next chapter.

For the LLITED project, the ram current due to the CubeSat velocity in low earth orbit is expected to be the predominant collection current. For example, at the minimum ion density requirement for LLITED ($n_i = 2 \times 10^9 \text{ m}^{-3}$) and assuming singly ionized oxygen ions at temperature 1000 K, the thermal current collected by the PIP sensor is on the order of 73 pA, whereas the ion ram current is expected to be on the order of 2.0 nA. Thus, the ion ram current is more than an order of magnitude larger than the ion thermal current at the lowest expected plasma density. At the highest expected plasma density, ion ram current is $20 \mu\text{A}$ versus a thermal current of $0.73 \mu\text{A}$, making ram current larger than ion thermal current by more than an order of magnitude. Knowing the spacecraft velocity and ram cross section area of the probe, one can then directly determine the absolute ion density from the measured current, assuming all ions are singly charged (which is true for ionospheric conditions). Thus, our instrument is designed to specifically target the ion ram current collection source.

Furthermore, we can define η as the ratio of ram to thermal current

$$\eta = \frac{I_{ram}}{I_{thermal,i}} = u \cos \theta \sqrt{\frac{2\pi m_i}{k_B T_i}}, \quad (1.5)$$

which is a function of only four parameters: spacecraft velocity u , ion mass m_i , ion temperature T_i , and angle to ram direction θ . For θ of 0° , spacecraft velocity of 8 km/s, and singly charged atomic oxygen ion of 1000 K, this convenient unitless ratio becomes:

$$\eta \approx 27.$$

An $\eta > 10$ implies that the thermal current collected by the probe is less than 10% of ion ram current collection. η drops from 27 to 10 when the ram angle offset is 68° . This implies that the PIP sensor can be offset from the ram by as much as 68° and yet maintain an order of magnitude more ram current than thermal current.

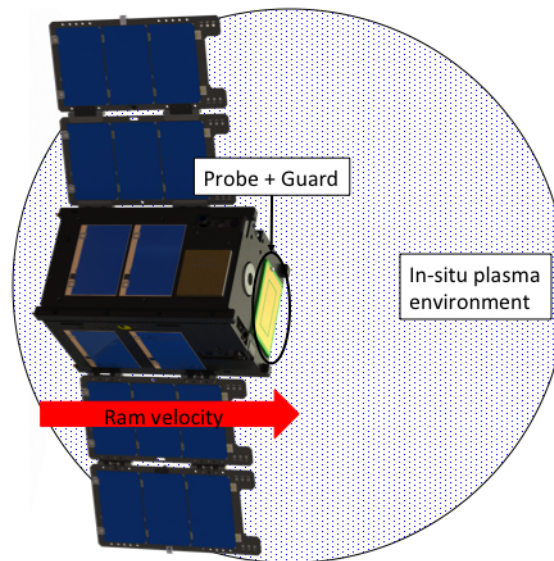


Figure 1.5: Portrayal of PIP sensor collecting current from ambient plasma on LLITED ram face.

Regardless of the operating mode of the Langmuir probe (geometry and sweeping/static potential), a consistent design aspect is the need to accurately measure the collected current. Typically, Langmuir probe circuitry is designed to convert the incoming current into a voltage with a pre-amplifier, and then further process and eventually digitize this voltage. Operational amplifiers, in a transimpedance configuration are a common and effective method of performing this current-to-voltage conversion. As shown in Figure 1.6, the circuitry block diagram shown has the probe sensor and

guard attached to the inverting and non-inverting inputs of the transimpedance amplifier, respectively. Since op-amps theoretically have a virtual ground between their inputs, the guard potential which is set to high accuracy by an external reference chip forces the probe sensor input, and therefore the whole sensor plate, to the reference voltage. The incoming collected probe current (which as was discussed mainly stems from ion ram collection), thanks to the extremely low input bias current characteristic of op-amps, is forced not into the input pin, but rather through the feedback resistor attached to the transimpedance amplifier. The current flowing over this known resistor produces a voltage drop, which is added at the output to the probe's voltage bias on the non-inverting input. Following this pre-amp, a second amplifier is used to subtract the probe bias voltage and flip the polarity from the negative regime of the transimpedance amplifier voltage inputs to a positive regime suitable for analog-to-digital conversion. Thus, all that remains after this two-stage amplifier system is a positive voltage with theoretically zero offset that is directly proportional to collected current, and is therefore *directly proportional to in-situ ion density*. In the PIP configuration, this takes the form of a dual-channel instrumentation amplifier that creates a high and a low gain channel for covering the entire instrument dynamic range with high precision. Following this, a filter cuts off high frequency noise from the signal and a pair of protection diodes clamps the signal lines to ground and supply voltage, before feeding into an analog-to-digital converter (ADC). Thus, with this architecture, high cadence, low noise, and low power measurements of absolute ion density based on collected ram current is possible with PIP.

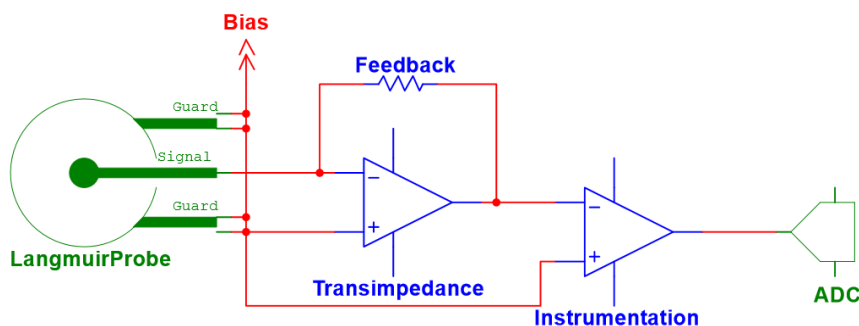


Figure 1.6: Langmuir probe electrometer circuitry block diagram.

Langmuir probes are widely flown on almost every sounding rocket experiment, and a fair amount of science spacecraft. Previous Embry-Riddle specific experience with flying Langmuir probes includes the German WADIS rocket campaign and NASA Mesosphere-lower thermosphere Turbulence eXperiment (MTeX) sounding rocket experiments, as well as the CubeSat-based Dynamic Ionosphere CubeSat Experiment (DICE) investigation. MTeX was a 2015 mesospheric sounding rocket launched from Wallops Island that featured three different Langmuir probe implementations: a conventional sweeping Langmuir probe, a multi-surfaced set of spherical probes fixed-biased in the electron saturation region capable of detecting atmospheric smoke particles, and a multi-needle probe biased at several points in the electron saturation region capable of making electron density measurements immune to low levels of spacecraft charging. All featured Embry-Riddle student design effort and acted as heritage sources of flight information and lessons for the LLITED PIP payload.

LLITED's PIP instrument draws heavily from the most recent of these experiments, MTeX, for some heritage part selection, design legacy, and recycling of testing apparatuses and procedures. Selections like signal connector style, instrumentation amplifier, protection diodes, and temperature sensor are identical to the MTeX boards, and use of the MTeX instrument power system was used for bench-top testing throughout the project. The low power microcontroller and analog to digital converter, the low noise front end transimpedance amplifiers, and the small form factor of the PIP sensor are the most important contribution of this work to further the Langmuir probe design from a sounding rocket platform to a low SWaP CubeSat platform.

The PIP instrument's development, design, and testing constitutes the scope of this thesis. The development work is enumerated in several chapters and appendices:

- In the next Chapter 2 we detail the PIP mechanical, electronic, and software design. Signal flow, component selection, key engineering design decisions, and instrument functional algorithm are also discussed in great detail.
- The subsequent Chapter 3 details the rigorous testing of the instrument to characterize its performance. Early Multisim circuit simulations verifying proof

of concept of the instrument signal flow, characterization of how low instrument noise is found to be, measured power consumption, power supply activation survival and measured inrush current, and wide thermal range survival tests are presented.

- We then document in Chapter 4 the meticulous calibration procedure of the instrument. The two part calibration process using a temperature chamber across functional temperature extremes, including data analysis with calibration fit coefficients and errors from theoretical are presented.
- Penultimately, a summary and conclusion of the thesis efforts are discussed, including alternate implementations of the technology and methods of improving future designs.
- Finally, the appendix incorporates the mechanical and electrical schematics for the envelope and boards as well as the C and Matlab code for flight and ground support, respectively.

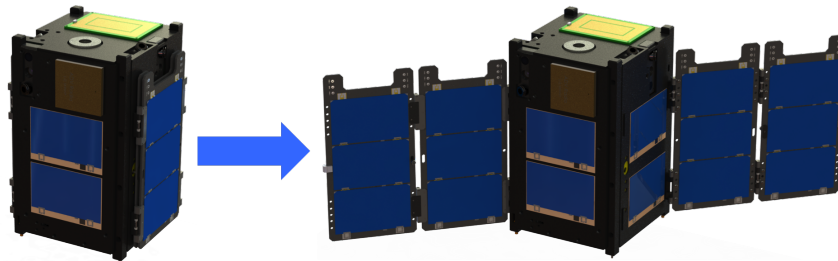


Figure 1.7: LLITED satellite configurations after launcher separation (solar panels stowed) and during nominal science mission activities (solar panels deployed).

Chapter 2

PLANAR ION PROBE DESIGN

This chapter summarizes PIP design aspects, and the choices that went into the design decisions. Included is an in-depth discussion of the general architecture, mechanical design, component selection, low noise design practices, and detailed design of the instrument. In this way, all aspects of the design are discussed in terms of mission requirements and spacecraft development and its impact on PIP.

2.1 General System Architecture

The PIP design is very straightforward: two PCBs, one acting as the Langmuir probe proper mounted externally and a second acting as the instrumentation payload processing the sensor's collected current mounted internally in the CubeSat. The LLITED ram plate separates the two PCBs and the internal board is surrounded on all sides by a solid aluminum envelope. This sensor-instrument-envelope-ram plate assembly can be integrated and mounted as a single unit on the CubeSat, with just a single external 10-pin connector providing communication, power, and programming capability with the spacecraft. Figure 2.1 visualizes the PIP subsystem on its own and within the larger LLITED spacecraft architecture.

The board exchanges signals, voltages, and communications with components outside its protective Faraday enclosure through two connectors: an ion current signal interface that carries the collected Langmuir probe signal current from the external

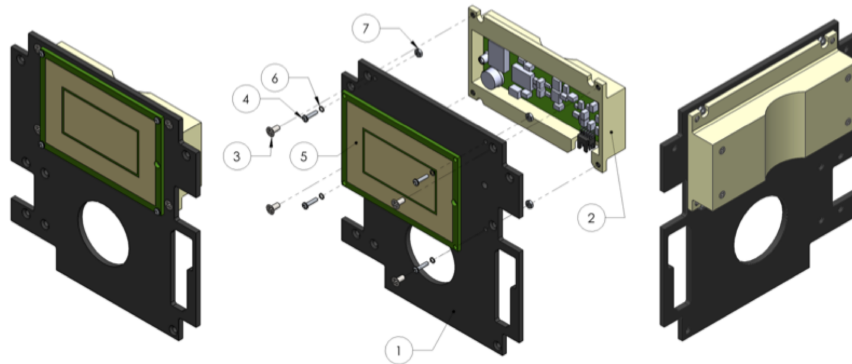


Figure 2.1: PIP subsystem assembly, including the sensor PCB and instrument PCB, in relation to the LLITED ram plate.

sensor board to the instrumentation for processing, as well as a ribbon cable connector carrying supply voltages, communications, and programming lines between the instrumentation and the payload interface board of the spacecraft.

The signal connection method chosen is a pair of male/female (P/N 262114/262104) Amphenol RF micro-miniature coaxial (MMCX) connectors that directly mate between the instrument and sensor boards, shown with the female instrument board connector on the left and male sensor board connector on the right in Figure 2.2. The coaxial nature of the connector allows the probe guard/reference voltage to surround the signal voltage on all sides, helping to protect the signal line from leakage currents and externally coupled noise. At only 4.5 mm above-board receptacle height, the MMCX is also an extremely small connector, which was necessary given the small space allocated to PIP in the CubeSat. Initially, both the instrument and sensor boards were designed with surface-mount MMCX connectors, but during testing these were found to be quite weak, breaking after a handful of connection cycles (and at the worst development times right before conferences!), and thus through-hole options were subsequently selected. A low noise RG188 cable with a male MMCX adapter on one end and a generic banana plug on the other allowed for interfacing with the circuitry during the majority of instrument testing.

In terms of the instrument PCB, an analog front-end conditions the collected probe current for digital conversion, while the subsequent digital half performs this

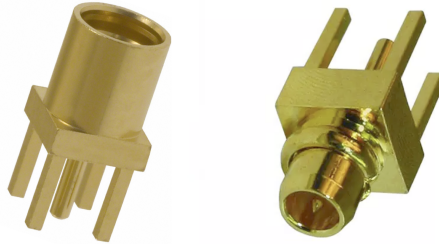


Figure 2.2: PIP instrument/sensor board MMCX interface connectors [Amphenol Connex, 2003, 2006].

analog-to-digital conversion. The digital half also controls data flow into and out of the instrumentation and responds to spacecraft commands, while with peripheral temperature, voltage, and light level sensors provides telemetry about the system health status. These two halves are located on the same PCB, though they are physically segregated to two ends of the board as is discussed in Section 2.2, where low noise design practices are discussed in greater detail.

The analog amplifiers are supplied ± 15 V, while the housekeeping and ADC analog supply use the +5 V source, and a low dropout regulator converts this +5 V to a +3.3 V level used in the microcontroller (MCU) and ADC digital supply.

Because the +3.3 V microcontroller requires a supply voltage not provided by the spacecraft, the +5 V line was sourced to provide this via a low dropout (LDO) regulator. The ADP3300 LDO regulator provides this functionality with a high ceiling of 100 mA maximum load current and a low dropout voltage maximum of 70 mV given a 10 mA load. Its small 2.80 mm \times 2.90 mm footprint is also beneficial for the dense digital section of the board [Analog Devices, c].

An initial pre-amp in transimpedance configuration converts the small incoming sensor current (~ 2 nA to $\sim 20,000$ nA) into a voltage. This voltage then has the bias voltage subtracted from it and is flipped into a positive range by a second, cascaded instrumentation amplifier. This instrumentation amplifier also separates the signal into two channels for higher resolution dynamic range coverage. A generalized block diagram enumerating this PIP instrumentation architecture is shown in Figure 2.3.

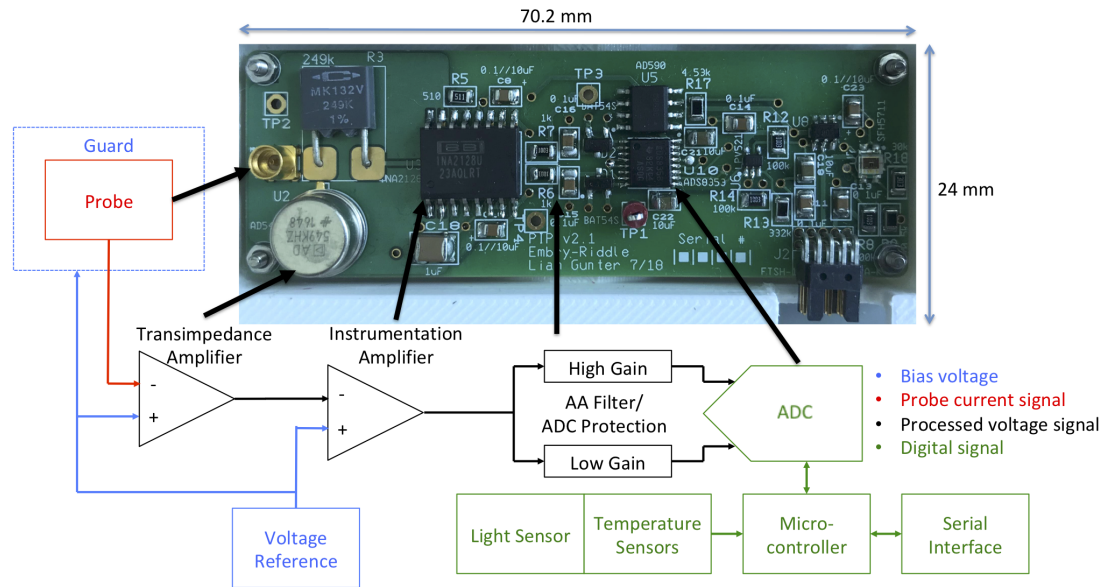


Figure 2.3: PIP hardware block diagram, showing signal flow and how it manifests on the physical v2.1 instrument.

Both commands and power are supplied from the spacecraft bus. The instrument-to-spacecraft interface chosen was a Samtec FTSH-105-01-F-D-RA-K connector. The Aerospace Corporation recommended this adapter due to prior experience with the product line and its 5.08 mm above-board height. The specific part has 10 conductors and a polarized keyed shroud, so backward connection of the ribbon cable *at the instrument* is prevented. The complementary FFSD ribbon cable line has varied options, making selection of the appropriate ribbon cable for spacecraft routing quite simple. This connector provides ground, power connections (+5 V, ± 15 V), universal asynchronous receiver-transmitter data communication (UART, TX/RX), and Spy-Bi-Wire (SBW) programming (+3.3 V sense/source, SBW data, SBW clock) to the payload. Figure 2.4 annotates the pinout visually on the connector for reference.

In addition to programming lines used during development and debugging, only two conductors (RX/TX) besides ground are required for UART. The instrument is supplied power (± 15 V and +5 V) from the spacecraft bus. That said, the instrument does regulate this +5 V supply to a lower +3.3 V digital supply line. Other than this, the power system is very straightforward: analog components and amplifiers

use the $\pm 15\text{ V}$ supplies, the ADC analog supply and housekeeping components use the $+5\text{ V}$, and the MCU and ADC digital supply use the onboard regulated $+3.3\text{ V}$. Three extra conductors on the spacecraft connector were allocated for programming purposes, as the instrument MCU is capable of programming via a SBW interface protocol, needing only a $+3.3\text{ V}$ source/sense connection, SBW data line, and SBW clock line. Learning from the lessons of MTeX, integrating the header pins necessary for programming the MCU directly into the off-board connector was a wise investment in terms of time and hassle during testing and development. Figure 2.4 provides a visual depiction of the spacecraft connector and its pinout, with more detailed pinout descriptions listed in tabular fashion in Table 2.1.

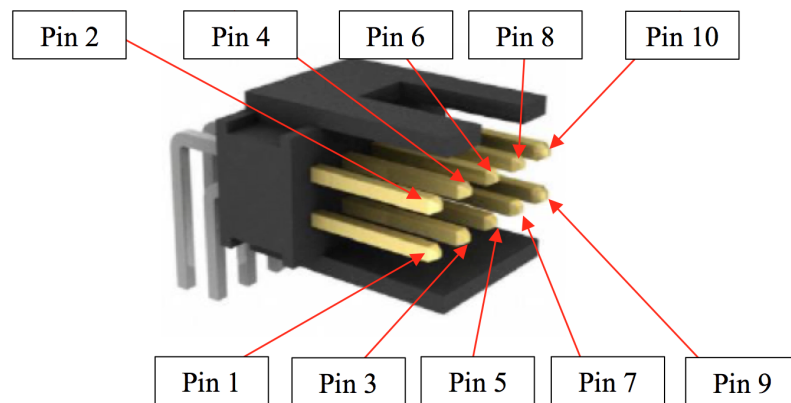


Figure 2.4: PIP off-board connector pinout cartoon [Samtec].

In order to provide industry-grade soldering capability and hasten development time, it was decided to have the PIP flight boards fabricated *and assembled* by the contracted PCB house used throughout the project for fabricating engineering test articles. With professional facilities, thus, IPC Class 2 standards, as well as traceability can be maintained. Five boards were ordered, allowing for one to be retained as an engineering test article, two to be used as flight boards, and two to act as flight board backups. The boards were numbered with permanent serial numbers (0b000, 0b001, 0b010, 0b011, and 0b100) both via physical silkscreen/permanent markings on the board surface as well as a static board ID (mentioned above) stored in software and communicated during each science data packet.

Pin #	Name	Description
1	TX	UART instrument transmit, 115.2 kbps
2	RX	UART instrument receive, 115.2 kbps
3	NC	No connection
4	-15VDC	-15 V DC power external supply
5	GND	Ground
6	+5VDC	+5 V DC power external supply
7	+3V3DC	+3.3 V sense (reserved for programming)
8	+15VDC	+15 V DC power external supply
9	RST	SBW TDO/TDI (reserved for programming)
10	TEST	SBW TCK (reserved for programming)

Table 2.1: PIP connector pinout and descriptions.

Each instrument board is survival tested and calibrated across temperature, as discussed in Sections 3.5 and 4.1, respectively. Each board also underwent 48 hours of $< 1 \times 10^{-4}$ Torr outgassing in an Embry-Riddle vacuum chamber. These provide confirmation that the instrument will survive and perform predictably during vacuum conditions and thermal cycling on-orbit in LEO. Additionally, as per integration procedures, the instruments underwent vibration testing on a subsystem and fully integrated LLITED basis at The Aerospace Corporation. Thus, the instrument is shown to be able to survive the vibrational environment during launch operations as well.

2.2 Low Noise Design Practices

In order to reach the low noise floor necessary for the mission requirements, careful attention was given to low noise design practices. These practices limit the amount of noise coupled from external sources, noise coupled from internal sources, and noise generated onboard the instrument.

2.2.1 Trace Design

Proper trace design can minimize electromagnetic interference (EMI). The most obvious is the minimization of trace length. Since the amplitude of noise picked up by a trace is proportional to the area created by the trace and its ground return path, minimizing this length and total area is critical. This concept also links back to component placement: orienting ICs on the proper side of the board and ensuring they are rotated for optimal signal direction helps prevent unnecessarily lengthy routing. Furthermore, to prevent signal reflection, trace corners are never taken with direct 90° angles - chamfered pairs of 45° angle turns is universal PCB practice, and even then are avoided in favour of straight traces when possible. Inside the analog segment of the board, the sensitive signal lines are taken at arced corners, an even better (though more difficult from a manufacturability standpoint to implement) trace corner technique. These trace geometries are visually depicted in Figure 2.5. Additionally, complementary traces (clock and data) for programming and ADC communications were carefully length matched as closely as possible. This helps prevent propagation delay mismatches, and thus communications errors in programming and signal measurement.



Figure 2.5: Accepted PCB trace corner format.

2.2.2 Power Supply Decoupling

Decoupling of components is one of the most important aspects of noise mitigation. All ICs are subject to power supply fluctuations, and thus removing these fluctuations

before they affect signal integrity is paramount. In this spirit, each IC is decoupled with a smoothing capacitor at each power supply pin. $0.1\ \mu\text{F}$ decoupling capacitors are utilized on the analog amplifiers. All other ICs feature their respective datasheet recommended decoupling capacitors, and even the FTSH connector supplying power also has $0.1\ \mu\text{F}$ decoupling capacitors near the power input pins to remove power line noise immediately after introduction to the instrument board environment.

2.2.3 Analog-Digital Separation

The analog and digital sections of the board are separated into two distinct regions. This separation prevents coupling of digital signals with analog ones by virtue of physical distance and component segregation. This isolation of the analog signals and components is also accompanied with analog bias planes discussed below and selective conformal coating on only the digital side. Incidentally, the development of the board roughly followed the process of filling the board from the one side with the probe signal interface and analog end to the other with the digital and spacecraft interface end.

2.2.4 Voltage Planes

Another advantageous tool for noise mitigation is the use of voltage planes in the design. In this case, two sets of planes exist on PIP: a bias plane connected to the -7 V probe reference voltage and a ground potential plane. The analog section of the board incorporates the bias plane on each of the four copper layers, shown in sequence from left to right in Figure 2.6 as copper top, copper inner 1, copper inner 2, and copper bottom layers. The sensitive current signal lines are, where possible, routed through inner layers. These steps ensure that the signal net is surrounded on the same layer, as well as by layers above and below with voltage planes at the prescribed bias voltage. Leakage currents from the signal line are thus minimized, given that the current signal is surrounded by a nearly identical potential (while still separated by high volume resistivity of the PCB dielectric itself). A ground plane across the entire post-instrumentation amplifier region of the board allows for low-impedance ground

return paths, as shown in Figure 2.7. The ground plane simplifies the circuit layout, allowing for grounding of ICs directly with a single via near the component ground pin. This ground plane occupies the copper inner 2 layer of the layout.

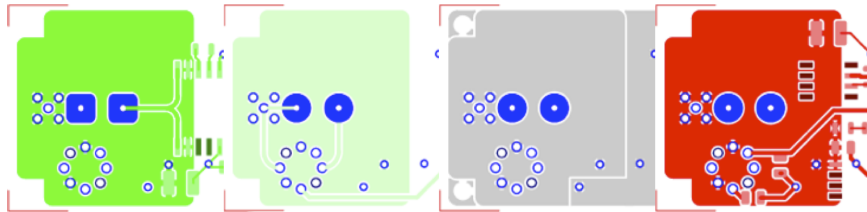


Figure 2.6: PIP analog section, displaying all four layers of bias planes.

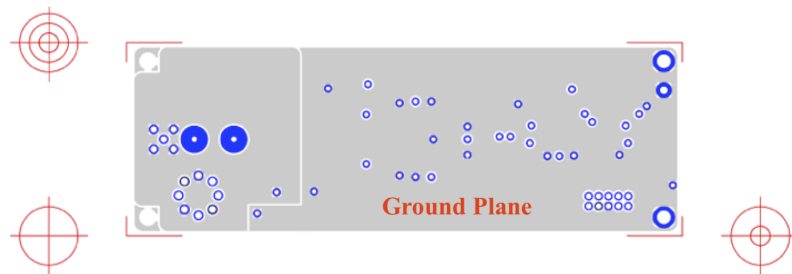


Figure 2.7: PIP inner copper 2 layer, with ground plane extent annotated.

2.2.5 Mechanical Design Considerations

Of note is that the compact size of the instrumentation electronics board lends itself positively to low noise generation and pickup of external EMI: the smaller the circuitry spread, the smaller the inherent collected and emitted noise. In order to remove noise from the rest of the spacecraft (or laboratory test environment), the entire instrument is encased in its own Faraday enclosure. This also simplifies mechanical mounting and modularity of PIP. Through two of the PCB mounting screws, this chassis ground is linked to the circuitry ground plane. This effectively separates the instrument from the rest of the CubeSat payload in terms of EMI, by surrounding it on all sides with a solid conductive grounded chassis. This envelope was discussed in greater detail and illustrated in the mechanical design Section 2.4.

2.2.6 Material Properties

The PCB material utilized throughout most early testing, as well as heritage from the previous SAIL Langmuir probes, was FR4. Upon material property research and confirmation with the Aerospace Corporation during the instrument peer review, polyimide was selected as a new circuit board material for v3.0 (and therefore the flight boards derived from this batch). A definite improvement in noise performance was observed between the v2.1 and v3.0 test articles, due to both an improvement in solder quality from the industry-grade job performed by the contractor and also from the higher volume resistivity polyimide. Additionally, heritage thick film Caddock resistors were used for transimpedance feedback resistance. Heritage experience has suggested low current noise emanating from these resistors in particular.

In terms of the AD549 transimpedance amplifier's specific high-resistivity requirements, the datasheet-specified required $10^{15}\Omega$ resistance between signal and supply lines is approached by manufacturing the PIP flight boards out of polyimide material and maintaining high distance between the supply lines from sensitive analog signals. Polyimide has a typical volume resistivity on the order of $\rho = 10^{17}\Omega \cdot \text{cm}$, compared to FR-4's of $\rho = 10^{14}\Omega \cdot \text{cm}$. Along these same lines, the conformal coating applied to the board must be kept away from this amplifier (as well as the rest of the analog section) for reasons of it having relatively low resistivity [Dupont, 2017, Keithley, 2014].

2.3 Detailed Design

The components for PIP were chosen with emphasis placed on low SWaP characteristics. Surface mount components were greatly preferred, due to consuming less board space and having lower electrical noise by being on only a single side of the board. Additionally, mixing of surface mount and through-hole components can significantly increase assembly complexity and cost by adding an additional technician-necessary step after top and bottom part placement and solder reflow. The instrument's passive components (resistors and capacitors) where possible were chosen in 0805 size

packages for the purposes of consistent soldering, use of pre-existing laboratory stock, and small footprints (but not too small to prevent hand soldering).

2.3.1 Analog Signal Flow/Processing

PIP went through four main phases of electrical design, manifesting as revisions v1.3, v2.0, v2.1, and v3.0. Revision v3.0 was the final flight board level design, implementing the collection of lessons learned from previous iterations, fabricated with high-quality electroless nickel immersion gold finish on a polyimide substrate, and turn-key assembled by our PCB contractor conforming to IPC class 2 standard.

In the most basic sense, any Langmuir probe is simply an electrometer: a current collected from an electrode immersed in plasma is measured and conclusions regarding the plasma state can be drawn from that reading, given information about the potential and geometry of the electrode.

Section 1.2 provided a more in-depth review of the plasma density-to-current collected relationship. The probe surface (immersed electrode) is held at a significantly negative potential (-7 V) relative to spacecraft ground. In this way, the probe operates inside the ion saturation region of the Langmuir probe IV curve in Figures 1.3 and 1.4. As discussed, this region provides an asymptotically constant thermal probe current component that is directly proportional to density. However, as also previously mentioned, PIP has been designed such that this thermal current is negligible: instead, the instrument primarily collects ions from direct ram velocity collection. This collected current thus becomes a linearly proportional analog to density, culminating in Equation 1.4.

The collected current is fed through a pair of MMCX connectors, which when fully assembled, directly joins the male connector of the sensor board to the female connector on the instrument PCB. The signal remains protected all the way through transmission, and has a very small cross section that can fit into the small PIP form factor within LLITED.

Voltage Reference

The voltage reference is an integral part of the system: biasing the guard plane and acting as a driving input to the transimpedance amplifier. Initially the selected bias potential was selected to be -10 V , but after breadboard testing it was found that the transimpedance and instrumentation amplifier chain could not process the signal effectively given only $\pm 15\text{ V}$ supplies. Thus, a -7 V potential was settled on. Negative voltage references are more specialized and difficult to find compared to their positive counterparts. The LT1021 was the only reference product line capable of both operating in negative shunt mode and possessing a 7 V product, the LT1021-7. Its modest 1.5 mA maximum supply current coupled with its typical temperature drift of $3\text{ ppm}/^\circ\text{C}$ and $4.0\text{ }\mu\text{V}_{\text{rms}}$ maximum output noise also makes it a suitable selection [Linear Technology].

Transimpedance Amplifier

The first stage in the electrometer design is conversion of the current signal into a voltage difference capable of being further processed. An initial AD549 operational amplifier in transimpedance configuration with a $249\text{ k}\Omega$ feedback resistor effects this conversion. The AD549 transimpedance amplifier chosen for PIP was not a heritage component. However, the impressive 100 fA maximum low input bias current made it ideal for LLITED purposes. Collaboration with the University of Chile team building a CubeSat Langmuir probe also helped make this decision, as the OPA128 amplifier they had success with was a pin-for-pin replacement for the AD549, but with a larger 150 fA maximum input bias current [Analog Devices, a].

The TO-99 metal can package that the AD549 uses is somewhat unique in that its metal can shell potential is accessible via a guard pin, allowing the bias voltage to shield the signal even inside the transimpedance amplifier. Note that the datasheet specifies that typical PCB material (e.g. FR-4) has inadequate volume resistivity to fully capitalize on the low noise of the amplifier, due to parasitic spurious currents [Analog Devices, a]. These concerns are mitigated through the material properties selections of polyimide, as discussed in Section 2.2.6.

The non-inverting op-amp input is connected to the voltage reference and sensor guard. The virtual ground between op-amp inputs thus forces the inverting input, connected to the probe collector and feedback resistor, to also assume the reference potential. Given the theoretically zero current entering op-amp inputs (maximum 100 fA input bias current for the AD549), the probe current signal is forced to divert through the feedback resistor instead. This current flow generates a potential drop across the feedback resistor, proportional to the current and resistance, meaning that at the first stage amplifier output the voltage must be more negative than the amplifier inputs and reference voltage. Figure 2.8 is a snapshot of the v3.0 instrumentation circuitry detailing specifically the transimpedance amplifier setup.

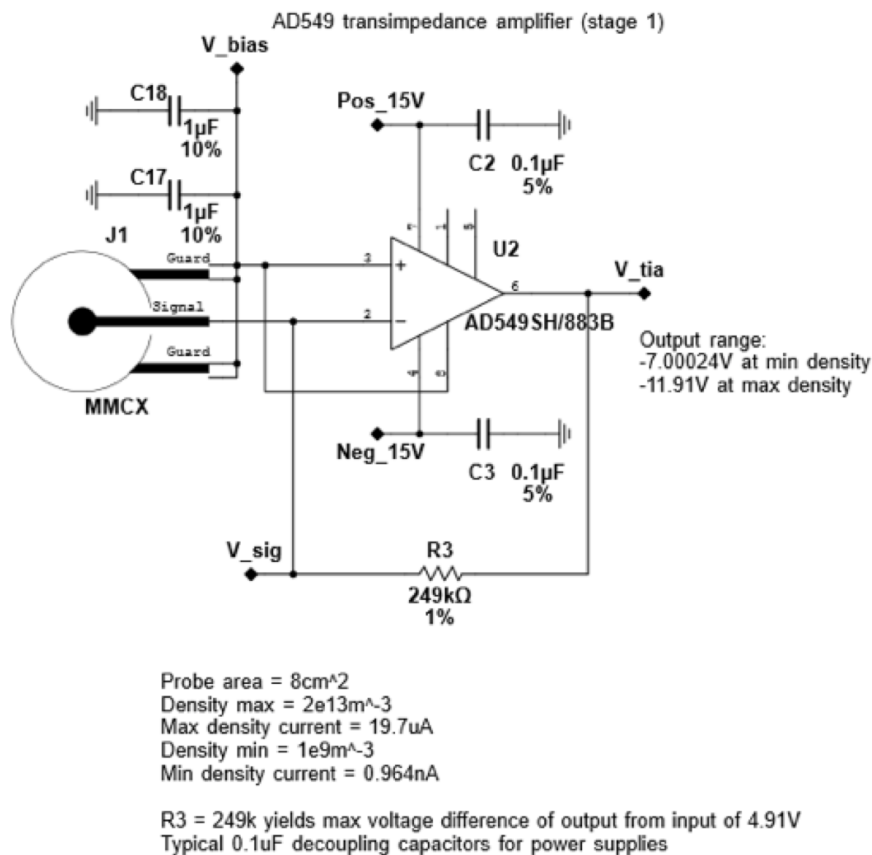


Figure 2.8: PIP transimpedance amplifier subsystem circuit schematic.

Feedback Resistor

As mentioned, the first stage amplifier, by virtue of being in transimpedance configuration, requires a feedback resistor to convert the input current into a voltage difference. This feedback resistor is chosen with special consideration, given its central role in the signal chain. The primary concern when selecting this resistor is the noise implications it has for the signal. While resistors in general exhibit thermal Johnson-Nyquist noise uniformly regardless of type, the current noise is dependent upon the physical construction of the resistor. Given past SAIL testing, MK132 series Caddock thick film resistors were found to have superior current noise characteristics. These resistors are also heritage to the MTeX payloads and a significantly sized existing laboratory stock cemented the decision to utilize this series of resistor. Note that the pads on the board are sized significantly larger than necessary to accommodate the through-hole MK132 leads for future use of alternative surface mount resistors, since the MK132 line is out of production.

Instrumentation Amplifier

The second stage of the electrometer should nominally perform two functions on the first stage output: subtract the reference voltage and flip the polarity of the signal so an ADC can digitize it. The INA2128 instrumentation amplifier is well-suited to perform both of these tasks. A heritage component from MTeX, and with two channels, a supply range accepting ± 15 V, and a maximum supply current of 1.5 mA, the INA2128 can provide both channels of instrumentation for less than 45 mW. The large 16-pin SOIC package also takes up a significant amount of area on the board. It is a proven, though somewhat resource-intensive instrumentation amplifier option [Texas Instruments, 2007].

The voltage signal is input to the inverting input and the reference bias voltage is fed to the non-inverting input. However, in order to precisely observe the intended four-decade scale dynamic range on a 16-bit ADC, separating the signal into a high and low gain channel is a practical requirement. The high gain channel observes the

Filtering/Protection

The third stage of the electrometer conditions the channel signals to safely interface with the ADC: constraining the signal cutoff frequency and clamping the voltage to prevent over-voltage/reverse-voltage at the ADC input. Cutoff frequency f_c , regulated by a low pass resistor-capacitor filter, is defined by the well-known relationship,

$$f_c = \frac{1}{2\pi RC} \quad (2.1)$$

where R and C are filter resistance and capacitance, respectively.

Selecting a cutoff frequency constrains this RC product. In PIP’s case, the cutoff frequency was chosen to be nominally 2 kHz. Limiting the power dissipation through the filter in the case of maximum measured instrumentation amplifier output of 14.1 V shorting to the +5 V line through the protection diodes subsequently fully constrains the resistance/capacitance selection. This selection was made to be a 100 k Ω and 1 nF combination. With this arrangement, an actual cutoff of 1.59 kHz is obtained, and a maximum of 0.774 mW dissipated. Testing with alternative selections, such as a 23.4 kHz cutoff or no filter at all led to much higher noise incident in the readings, and thus were not chosen.

In the same line of thought as conditioning the signal voltages in frequency for processing before the ADC, protection diodes short over-voltage and reverse-voltage signals to the supply and ground lines, respectively, in order to prevent these extremely high or low voltages interfacing with the sensitive ADC internal circuitry. Internally, though most ADCs do include some Schottky diodes to protect the sensitive circuitry (indeed, the ADS8353 has this feature), these diodes are typically not power-rated for more than brief transient surge protection (e.g. electrostatic discharge mitigation). MTeX heritage BAT54 diodes, popular general-purpose Schottky diodes with an adequately small forward voltage of 0.24 V at 0.1 mA were chosen. Since each signal channel needed to be capable of shorting to both +5 V and ground to protect the ADC, the BAT54S “series” variant was chosen so that a single component adequately maintains the input to the ADC below the required 0.3 V ground/supply voltage deviation specified in the ADS8353 datasheet [Diodes Incorporated, 2016,

Texas Instruments, 2014a].

It was found late in development that the small 1 nF capacitance used in the filter was not adequately large to accumulate enough charge in its role as a “flywheel” filter to fill the ADC sample and hold circuitry, resulting in a sag in output counts proportional to the number of oversamples taken. Troubleshooting resulted in a solution without hardware change by adjusting software delay times between successive samples sufficiently to allow the filter to charge and settle to get quasi-stable readings. A 4800 cycle delay time (300 μ s) between samples and a reduction from 32 \times to 16 \times oversampling was considered acceptable and adopted.

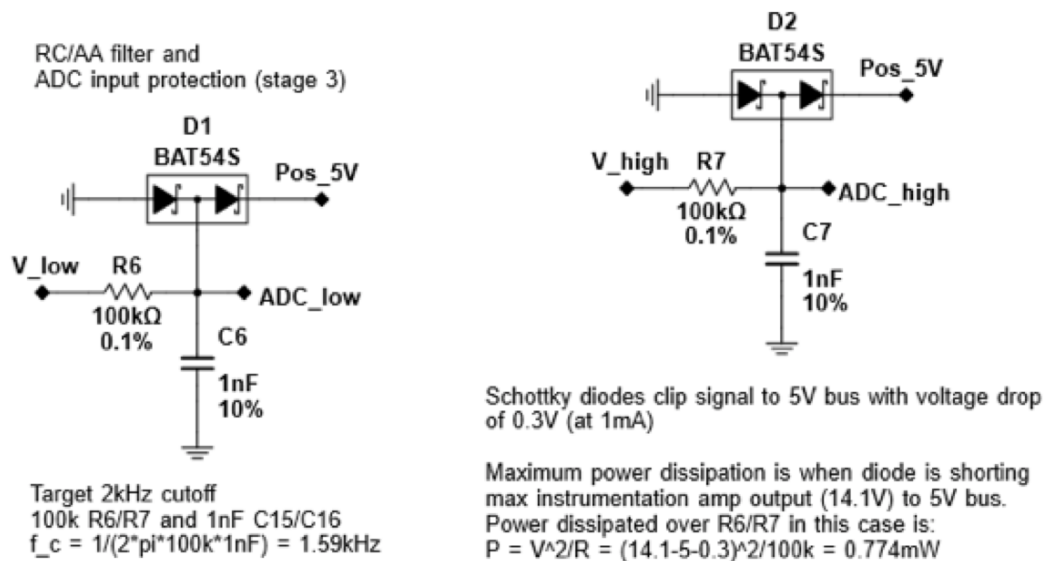


Figure 2.10: PIP filter/diode protection subsystem circuit schematic.

Future implementations of PIP, in order to avoid losing out on computation time during these delay cycles and be able to sample faster, should rectify this phenomenon. This can likely be accomplished by using a low power buffer amplifier stage following the instrumentation amplifier. Figure 2.10 overviews both the filter and diode protection scheme for high and low gain channels on PIP.

Analog to Digital Conversion

The ADC takes this processed signal voltage, and produces a digitized numerical code output. Given an ADC reference voltage (different from the instrument bias signal), the fraction of the ADC input voltage compared to this reference is mapped to a 16-bit numerical scale. With the PIP system, either high or low gain channels inputting 0 V to the ADC will result in zero counts, +5 V will result in 65,535 counts, and any voltage between 0-5V will linearly map within this number scale.

The ADC selection criteria involved several factors. Sequential approximation register (SAR) type ADCs were considered primarily, because of their proven heritage and ubiquity in the ADC market. A single ADC possessing two channels, 16-bit resolution, and a sample rate that can support oversampling 16 times at 100 Hz are required. Additionally, in order to prevent spatial separations of low vs. high gain measurements, the measurements must occur simultaneously. If they are not sampled simultaneously, the slight delay between samples will translate to a physical separation: a 100 μ s delay is mapped to ~ 1 m position change, which is deemed unacceptable. Use of the +5 V line for the analog supply is preferred, and an ADC with an internal reference and the capability to directly (without level shifting) interface digitally with the +3.3 V microcontroller electronics were preferred. Serial peripheral interface (SPI) or I²C, the protocols that the selected microcontroller supports on its communications module not dedicated to spacecraft UART data transfer, are also a requirement. The ADS8353 fits all these criteria, and possesses a respectable 89 dB SNR and sampling rate up to 1 M samples per second. The SAR converter also has two reduced power modes, one of which is used in operation to save on PIP energy expenditure. Figure 2.11 visually communicates the ADS8353 circuitry setup, and how the SPI communications lines interface with the MCU [Texas Instruments, 2014a].

The ADS8353 communicates via SPI, an extremely common four-wire serial bus communication protocol used to link “master” and “slave” ICs and enable high-speed intra-board communication. SPI involves four communication lines: a clock signal which coordinates bit exchange timing, a master out slave in (MOSI) master-to-slave direction communication line, a master in slave out (MISO) slave-to-master direction communication line, and a channel select line that notifies slave ICs of the master

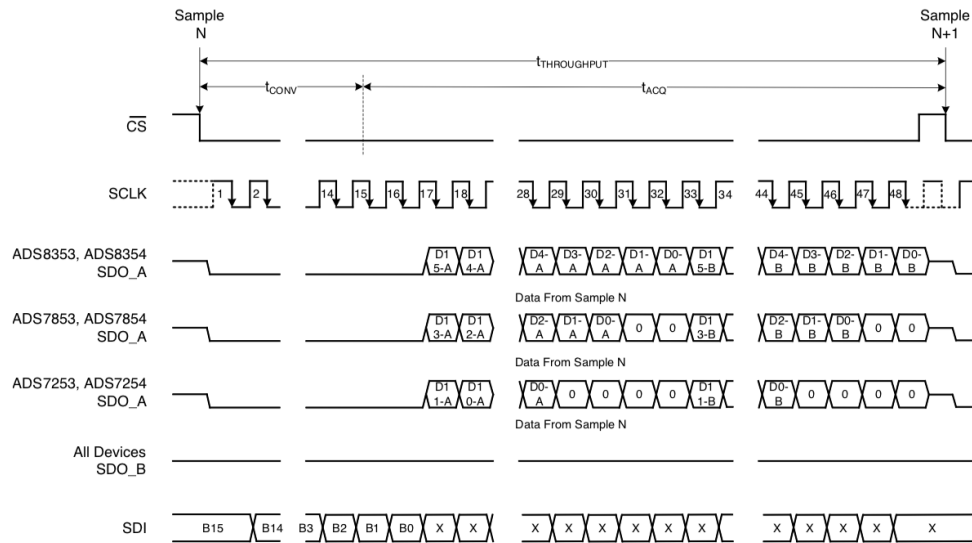


Figure 2.12: ADS8353 conversion data read protocol timing diagram [Texas Instruments, 2014a].

In order to change configurable settings in the ADC, instead of sending zeros in the first 16 bits sent to the ADS8353, the MCU actually addresses a destination register and rewrites settings. Thus, before any measurement PIP makes, the MCU ensures that the configuration register (CFR) and two internal references REFDAC A/B are loaded with the necessary settings described below. The internal references are utilized to provide stable +2.5 V reference sources for the channels, and the CFR is configured to extend the ADC input range to $2\times$ this internal reference (+5 V). These CFR and REFDAC A/B register settings are enumerated in Figures 2.13 and 2.14, respectively.

Within the measurement algorithm, initially the configuration registers and references are selected to reflect the above design choices. Following this, 16 consecutive conversions are carried out and stored in the MCU, buffered temporally by the aforementioned software delays in the MCU. Following these, a low power standby mode is entered by once again accessing the CFR.

15	14	13	12	11	10	9	8
WRITE/READ	0	ADDR1	ADDR0	RD_CLK_MODE	RD_DATA_LINES	INPUT_RANGE	0
7	6	5	4	3	2	1	0
INM_SEL	REF_SEL	STANDBY	RD_DATA_FORMAT	0	0	0	0

Bit	Field	Type	Reset	Description
15	WRITE/READ	W	0h	These bits select the user-programmable register. 1000 = Select this combination to write to the CFR register and to enable bits 11:0
14	0	R/W	0h	
13	ADDR1	R/W	0h	
12	ADDR0	R/W	0h	
11	RD_CLK_MODE	R/W	0h	This bit provides clock mode selection for the serial interface. 0 = Selects 32-CLK mode (default) 1 = Selects 16-CLK mode (Note that the ADS8353 only supports 32-CLK mode. This bit is ignored for the ADS8353.)
10	RD_DATA_LINES	R/W	0h	This bit provides data line selection for the serial interface. 0 = Use SDO_A to output ADC_A data and SDO_B to output of ADC_B data (default) 1 = Use only SDO_A to output of ADC_A data followed by ADC_B data
9	INPUT_RANGE	R/W	0h	This bit selects the maximum input range for the ADC as a function of the reference voltage provided to the ADC. See the Analog Inputs section for more details. 0 = FSR equals V_{REF} 1 = FSR equals $2 \times V_{REF}$
8	0	R/W	0h	This bit must be set to 0 (default)
7	INM_SEL	R/W	0h	This bit selects the voltage to be externally connected to the INM pin. 0 = INM must be externally connected to the GND potential (default) 1 = INM must be externally connected to the FSR_ADC_x / 2 potential
6	REF_SEL	R/W	0h	This bit selects the ADC reference voltage source. Refer to the Reference section for more details. 0 = Use external reference (default) 1 = Use internal reference
5	STANDBY	W	0h	This bit is used by the device to enter or exit STANDBY mode. Refer to the STANDBY Mode section for more details.
4	RD_DATA_FORMAT	R/W	0h	This bit selects the output data format. 0 = Output is in straight binary format (default) 1 = Output is in twos complement format
3:0	0	R/W	0h	These bits must be set to 0 (default)

Figure 2.13: ADS8353 CFR register settings information [Texas Instruments, 2014a].

15	14	13	12	11	10	9	8
WRITE/READ	0	ADDR1	ADDR0	D8	D7	D6	D5
7	6	5	4	3	2	1	0
D4	D3	D2	D1	D0	0	0	0

Bit	Field	Type	Reset	Description
15	WRITE/READ	W	0h	These bits select the configurable register address. 1001 = Select this combination to write to the REFDAC_A register 1010 = Select this combination to write to the REFDAC_B register
14	0	R/W	0h	
13	ADDR1	R/W	0h	
12	ADDR0	R/W	0h	
11:3	D[8:0]	R/W	0h	Data to program the individual DAC output voltage. Note: These bits are valid only for bits 15:12 = 1001 or bits 15:12 = 1010. Table 8 shows the relationship between the REFDAC_x programmed value and the DAC_x output voltage.
2:0	0	R/W	0h	This bit must be set to 0 (default)

REFDAC_x VALUE (Bits 11:3 in Hex)	B[2:0]	Typical DAC_x OUPUT VOLTAGE (V) ⁽¹⁾
1FF (default)	000	2.5000
1FE	000	2.4989
1FD	000	2.4978
—	—	—
1D7	000	2.45
—	—	—
1AE	000	2.40
—	—	—
186	000	2.35
—	—	—
15D	000	2.30
—	—	—
134	000	2.25
—	—	—
10C	000	2.20
—	—	—
0E3	000	2.15
—	—	—
0BA	000	2.10
—	—	—
091	000	2.05
—	—	—
069	000	2.00
—	—	—
064 to 000	000	Do not use

Figure 2.14: ADS8353 REFDAC register settings information, applicable to both REFDAC A/B [Texas Instruments, 2014a].

2.3.2 Housekeeping

The MCU runs on a supply voltage of +3.3 V, provided by a LDO IC regulating the +5 V line down. The internal MCU ADC system leverages an internally generated +1.5 V reference. With this reference, peripheral measurements of system health checks on PIP are performed.

Temperature

PIP was initially rated for a conservative operational temperature range of -25°C to $+50^{\circ}\text{C}$ in the CONOPS and initial ICD. However, during Aerospace thermal analysis development, this upper bound was changed to $+60^{\circ}\text{C}$. Measuring the onboard temperature is key to calibrating the instrument. The MTeX heritage AD590 temperature transducers utilized have a linear $1\ \mu\text{A}/\text{K}$ current response [Analog Devices, b].

It was decided that to simplify the design and allow for straightforward upper temperature range changes to use a single gain resistor (as opposed to an amplifier configuration) that directly converts current to voltage read by the internal MCU ADC. This design sacrifices the lower range of the ADC range (and therefore limits resolution to approximately $\sim 0.3\ \text{K}$), but ensures easy temperature range modifications, as displayed after the events of the thermal meeting with Aerospace. Accommodating the $+60^{\circ}\text{C}$ upper bound temperature was carried out by switching the original $4.53\ \text{k}\Omega$ temperature feedback resistor with an updated $4.22\ \text{k}\Omega$ one. Figure 2.15 shows the PIP temperature sensor circuitry design.

Figure 2.16 demonstrates simulated temperature sensor output from the temperature circuitry. The abscissa represents the expected transducer current from the AD590 for expected design temperatures of the instrument, while the ordinate represents the MCU ADC input voltage level polled for housekeeping purposes. As is apparent, up to $331\ \mu\text{A}$ (equivalent to $331\ \text{K}$), the ADC input voltage is within the $+1.5\ \text{V}$ range available for measurement. Note that this simulation is accurate for the original $4.53\ \text{k}\Omega$ setup.

Note that the temperature sensors are not calibrated for determination of exact board temperature. This is due to the fact that mission data analysis *needs no*

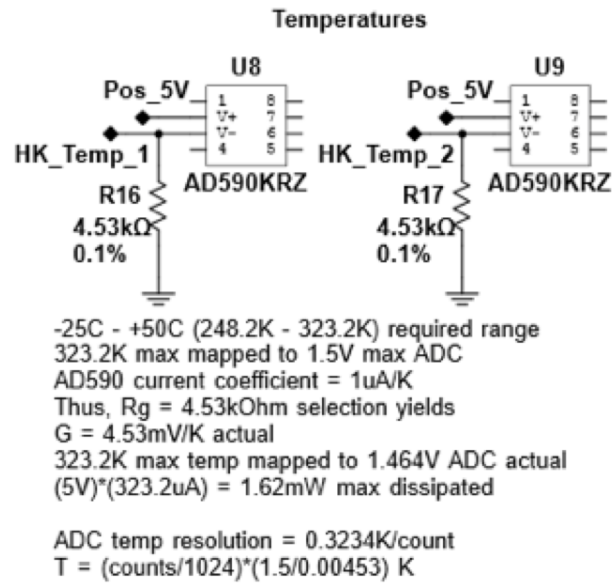


Figure 2.15: PIP temperature sensor housekeeping circuit schematic.

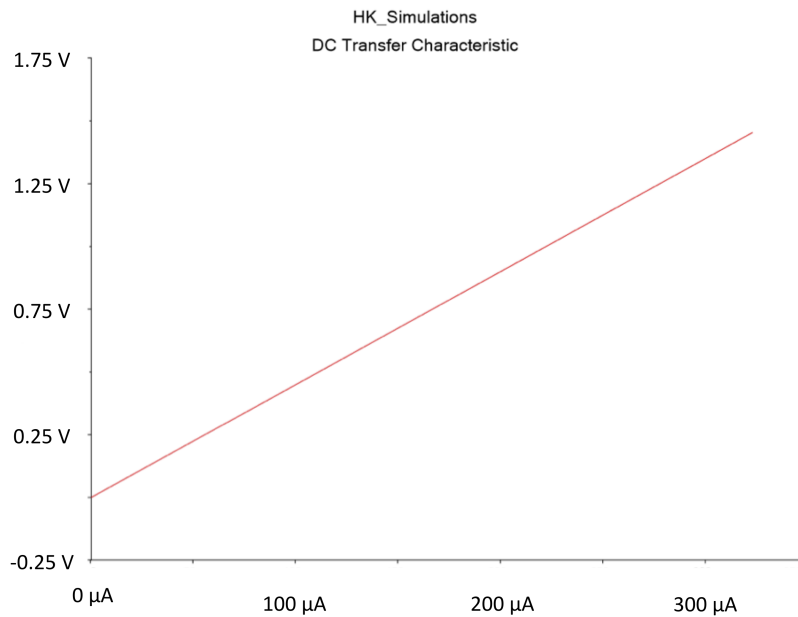


Figure 2.16: PIP temperature sensor housekeeping Multisim simulation.

knowledge of the specific temperature of the board, but rather relies on the relationship between housekeeping temperature counts and science channel data during calibration to properly characterize the sensor current measured. That said, for reference, the theoretical transfer function from counts x to sensed temperature T for the updated 4.22 k Ω setup is,

$$T(x) = \left(\frac{x}{1024} \right) \left(\frac{1.5}{0.00422} \right) \text{ [K]}. \quad (2.2)$$

Supply Voltages and *Quis Custodiet Ipsos Custodes?*

A Latin phrase attributed to Juvenal in the 2nd century CE, loosely translating to “who will guard the guards themselves?” this adage highlights a practical problem in the scientific and engineering problems of system redundancy and fault-tolerance, since embedded systems monitor their own status with onboard sensors, which are themselves subject to changes in the status of the device.

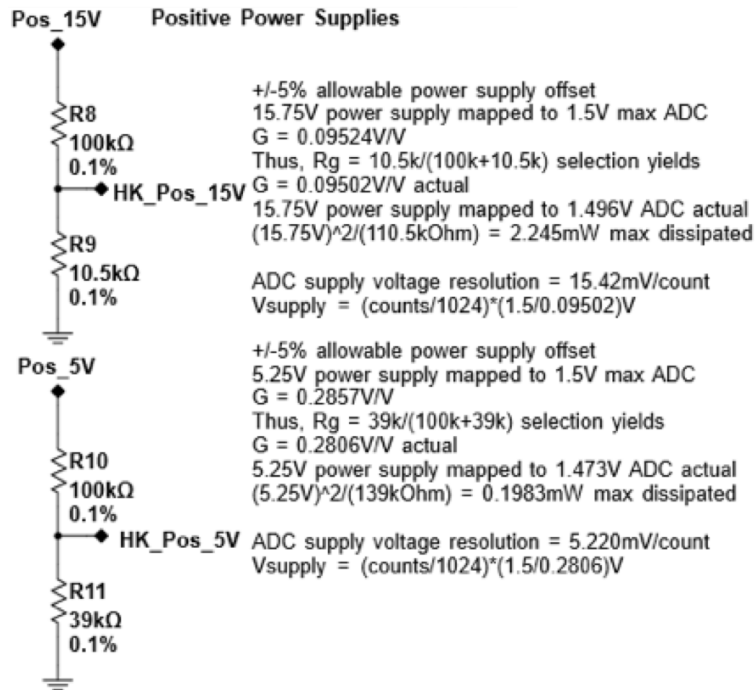


Figure 2.17: PIP +5 V/+15 V housekeeping circuit schematic.

The PIP instrumentation monitors its own conditions and uses this as a health check, while necessarily relying upon supply voltages to make these very measurements. We must be careful to functionally separate these housekeeping systems as much as possible from the supply voltages they are powered from, especially in fault or under-voltage failure modes.

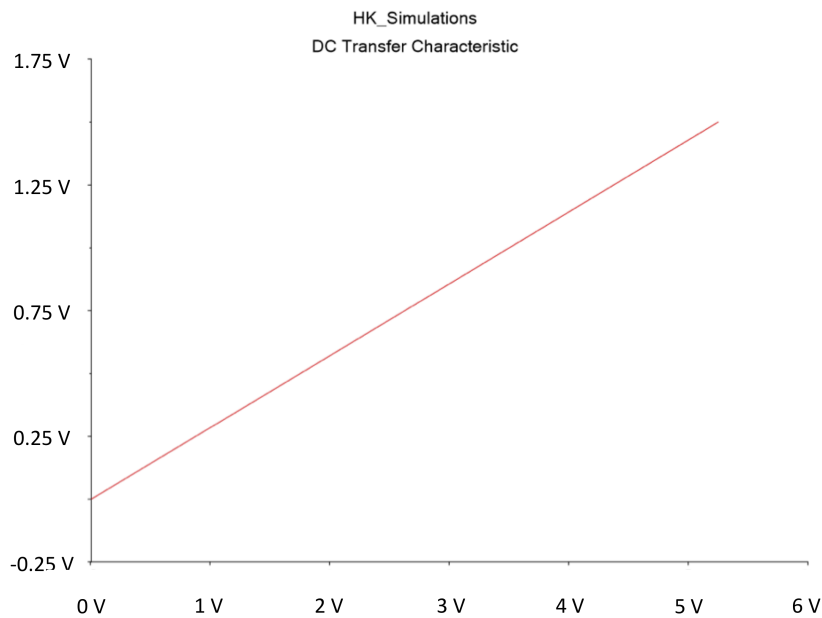


Figure 2.18: PIP +5 V housekeeping Multisim simulation.

This is accomplished thanks to the fact that the MCU ADC utilizing its internal +1.5 V reference can be maintained at MCU supply voltages as low as +1.8 V, according to the MCU datasheet. This means that operation of all the housekeeping channels can be maintained, and compensated for with the nominally +5 V supply that can drop as low as +1.8 V.

Housekeeping the two positive +5 V and +15 V supply voltages was straightforward. Voltage dividers scale the maximum expected voltage into the MCU ADC range, and allow for direct MCU measurement of the reduced divider voltage. From this we can scale up appropriately to find the voltage and determine the supply voltage level. These channels' circuitry are displayed in Figure 2.17 and the results

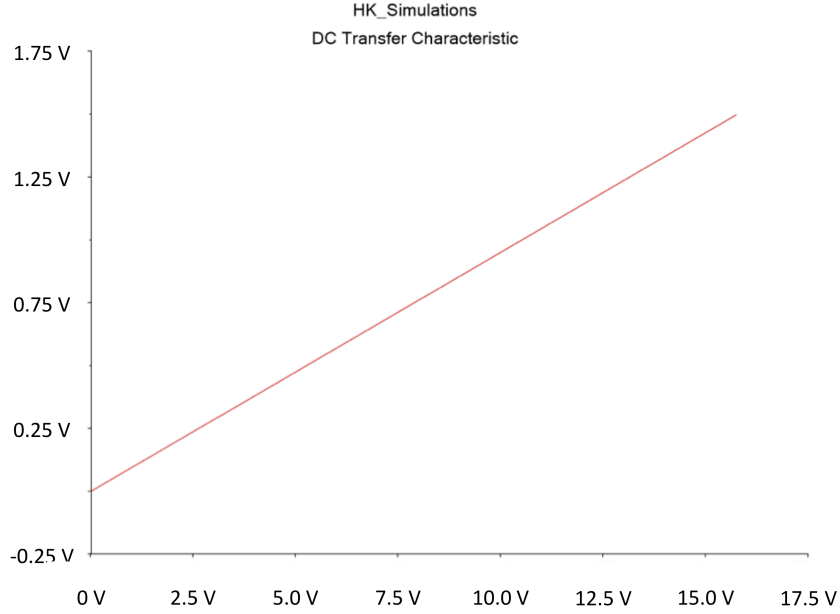


Figure 2.19: PIP +15 V housekeeping Multisim simulation.

of +5 V and +15 V Multisim simulations are shown in Figures 2.18 and 2.19. The abscissa shows the supply voltage, while the ordinate shows the MCU ADC input voltage response. As can be seen, the response maps the two supplies to the +1.5 V MCU ADC housekeeping input range, with a 5% margin above the expected supply voltage (+5.25 V and +15.75 V).

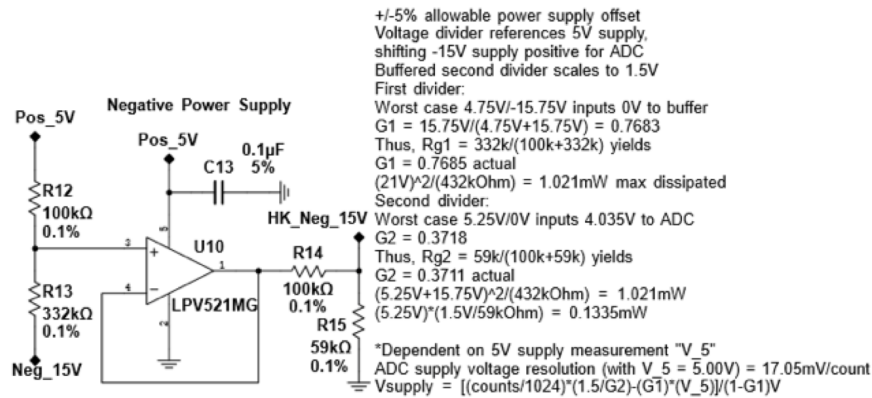
For reference, the theoretical transfer function from counts x to sensed +5 V supply voltage V_{+5} is,

$$V_{+5}(x) = \left(\frac{x}{1024} \right) \left(\frac{1.5}{0.2806} \right) [\text{V}], \quad (2.3)$$

and the transfer function from counts x to sensed +15 V supply voltage V_{+15} is,

$$V_{+15}(x) = \left(\frac{x}{1024} \right) \left(\frac{1.5}{0.09502} \right) [\text{V}]. \quad (2.4)$$

The negative supply voltage was more involved to study. The problem stems from

Figure 2.20: PIP $-15V$ housekeeping circuit schematic.

the fact that the voltage needs to be flipped to a positive voltage capable of being read by the positive ADC in the MCU. To do this directly with an op-amp requires the op-amp to operate with one rail supplied by the $-15V$ line itself.

Instead, an alternative system was developed where an initial voltage divider between the $+5V$ and $-15V$ line is designed to produce a positive output. A buffer amplifier feeds this to a second stage divider that scales the output voltage down for reading into the $+1.5V$ MCU internal ADC. This was considered acceptable, since the $+5V$ line is known with high confidence from its own housekeeping measurement, and proper $+5V$ behavior is required for the $+3.3V$ regulator to operate the MCU (and therefore its internal $+1.5V$ ADC reference) anyway, since if the $+5V$ fails, the MCU will not function regardless.

The intermediate amplifier required to buffer the signal between the first stage divider and the second-stage dividers was chosen for its small size and ultra-low power. The chosen LPV521 amplifier was tested on Multisim and found to be functional in the desired range of $-15V$ supply voltage. First tested on the integrated v2.1 board without prior breadboard testing, the results reflected the Multisim results during the power supply cycling permutation survival testing output data. This particular amplifier was specifically chosen thanks partially to its existence in the Multisim library, $+5V$ /ground tolerant supply, and $< 1\mu A$ supply current [Texas Instruments, 2014b].

Figure 2.20 shows the schematic of the $-15V$ polling scheme, while Figure 2.21

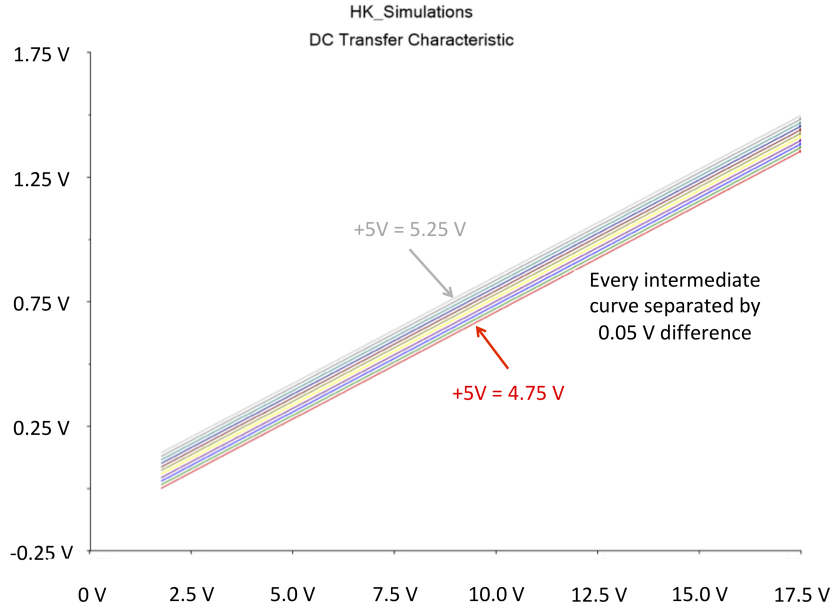


Figure 2.21: PIP -15 V housekeeping Multisim simulation.

shows the Multisim simulated output of the -15 V housekeeping circuitry, given varying $+5$ V supply voltages. The abscissa shows the -15 V supply voltage, while the ordinate shows the MCU ADC input voltage response given several values of the $+5$ V supply. As can be seen, the response maps the supply to the $+1.5$ V MCU ADC housekeeping input range, up to a 5% margin below the expected supply voltage (-15.75 V). Spreadsheet simulations also backed up these Multisim simulations, and benchtop testing throughout the rest of development results never yielded unexpected voltage values.

For reference, the theoretical transfer function from counts x to sensed -15 V supply voltage V_{-15} is,

$$V_{-15}(x, V_{+5}) = 4.320 \left(\left(\frac{x}{1024} \right) \left(\frac{1.5}{0.3711} \right) - (0.7685) (V_{+5}) \right) \text{ [V]}, \quad (2.5)$$

where V_{+5} is the calculated $+5$ V supply voltage value, determined from previous housekeeping measurement.

Light Sensor

The advisor of this thesis, as well as an engineer consulted on the PIP peer review design from the AFRL have both run into issues historically with photoelectron current on Langmuir probes. In order to provide some additional data on this front, a visible light optical sensor was added to the design. The optical visible light sensor, being a component not included on any previous designs, as well as a type of sensor practically unknown to the author, was chosen for its geometry (upward sensing surface mount device), small $2.35 \text{ mm} \times 2.95 \text{ mm}$ footprint, flexible supply voltage between 2.3 V to 5.5 V, and visible spectrum response with a center wavelength response at 555 nm. This SFH5711 light sensor is additionally manufactured by OSRAM, a recognized world leader in light sensor technology [Osram Opto Semiconductors, 2013].

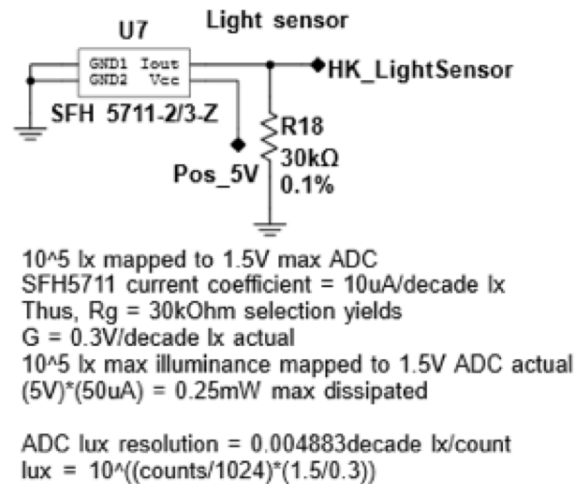


Figure 2.22: PIP light sensor housekeeping circuit schematic.

As orbital light sources may be variable in intensity, the light sensor setup was calibrated with a consistent light source to determine only the angular response of the assembled physical subsystem (PCBs, LLITED end plate, and Faraday enclosure). From this angular calibration, inferences about possible photoelectron currents can be made and compensated for in the post-flight data. Measurements of the light sensor output due to a stationary phone-based LED source in a light-absorbent enclosure at controlled incident angles were made at 15° intervals with the v3.0 test article, and

recorded for reference during mission data analysis. Table 2.2 enumerates the angular calibration data from testing, while Figure 2.22 illustrates the electrical schematic of the light sensor subsystem. A Multisim simulation in Figure 2.23, testing the MCU ADC input voltage for this housekeeping channel given varying currents from the SFH5711 light transducer, shows that the expected illuminances map to the +1.5 V ADC linearly as expected. For reference, the theoretical transfer function from counts x to sensed light sensor incidence E in lux is,

$$E(x) = 10^{\left(\left(\frac{x}{1024}\right)^{\left(\frac{1.5}{0.3}\right)}\right)} \text{ [lux]}. \quad (2.6)$$

θ ($^\circ$)	Output (counts)
0	600
15	348
30	331
45	238
60	85
75	0
90	0

Table 2.2: PIP 0b000 light sensor calibration.

2.3.3 Software Architecture

The MSP430 family of MCUs were selected due to their reputation for ultra-low power consumption. The MSP430FR4133 selected utilizes +3.3 V supply voltage to achieve down to $126 \mu\text{A}/\text{MHz}$ (with PIP implementing an operating clock of 16 MHz). This choice necessitated the consideration of peripherals and connections that did not exceed the low supply voltage and thus damage the MCU [Texas Instruments, 2014c].

The 48-pin package variant selected allows for simultaneous UART communication with the spacecraft, SPI communication with the ADC, and 6 internal ADC channels for housekeeping. Future PIP functionality expansion is supported through the 32 unused general purpose input/output (GPIO) pins. It should be noted that the MSP430FR4133 is not rad-hardened; future implementations of a PIP derivative can

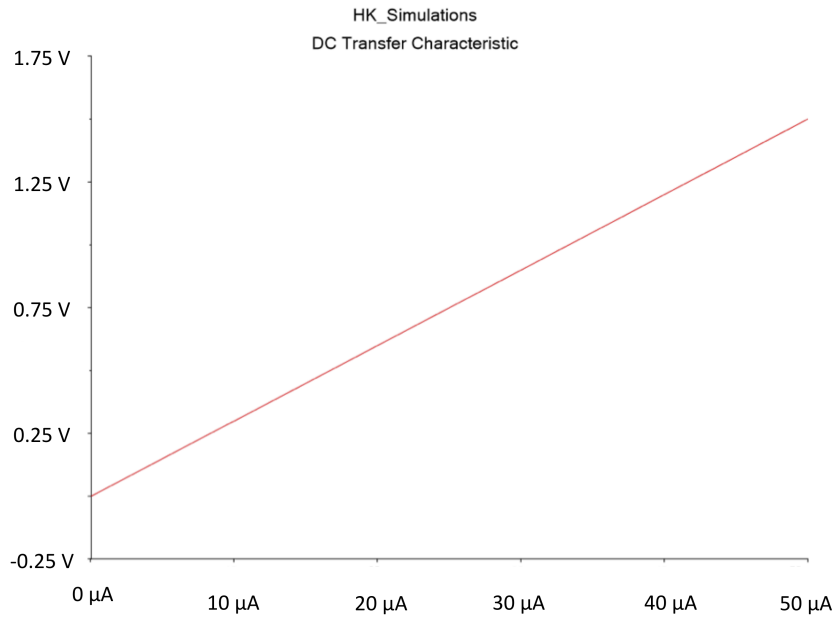


Figure 2.23: PIP light sensor housekeeping Multisim simulation.

benefit from the fact that the MSP430 has rad-hardened options available, and PIP operates with an onboard watchdog timer and spacecraft-level monitoring/resetting protocols to resolve single event upsets. On-orbit reprogramming is also possible through the spacecraft payload interface board, since the programming lines are included directly in the FTSH connector [Texas Instruments, 2014c].

In short, the MCU software must receive UART commands from the spacecraft, configure and sample the external ADC, average this data, poll housekeeping ADC channels, and organize and output this data back out over UART. Between spacecraft measurement commands, the system should save power by entering optimized power modes. The basic state machine describing this algorithm for PIP is shown in Figure 2.24, while the firmware itself is catalogued in Appendix E.

LLITED CONOPS dictates that the instrument operate in an asynchronous, interrupt-driven polling fashion, in a mission science mode where density measurements are collected at 100 Hz. Chosen in part to be as modular as possible for other

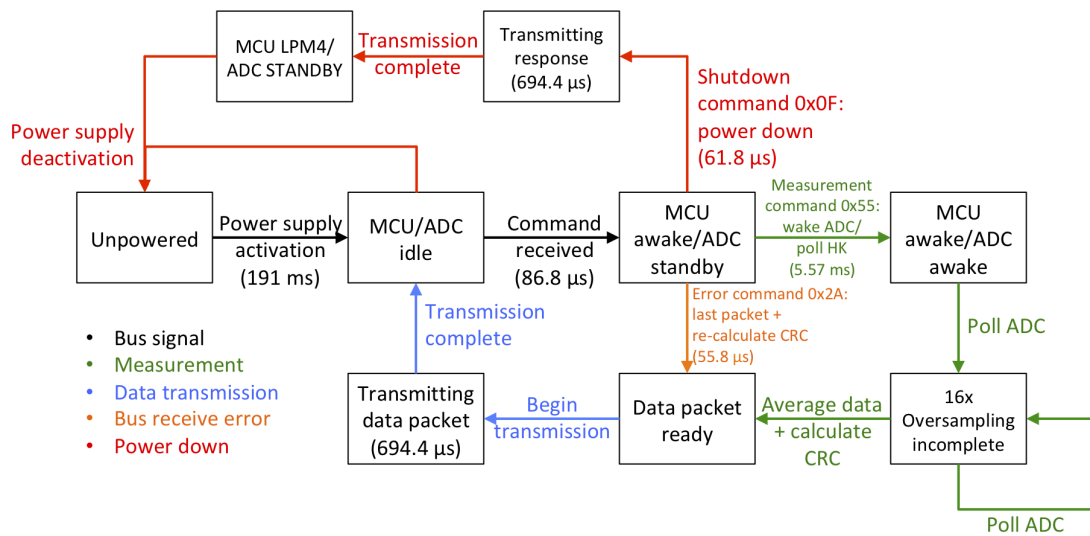


Figure 2.24: PIP state machine, displaying the command response algorithm paths.

vehicle implementations, this design decision also greatly simplifies the interface required for PIP. With this science measurement rate, the mission science requirements are more than met, while maintaining reasonable instrument activity/response time margin. This means that PIP communicates with the spacecraft bus via a UART at 115.2 kbps baud rate, hibernating in a nonfunctional mode until a detected UART-driven interrupt triggered by a spacecraft command spurs an instrument action and subsequent response.

There are two interrupt service routines defined in PIP: a UART driven command routine, as well as a watchdog timer interrupt that resets the timer so that the instrument does not reset. The watchdog ensures that if the MCU anomalously gets hung up (and can't service the watchdog reset timer), the MCU automatically resets into operational mode. When not servicing either of these routines, the instrument algorithm defaults to an infinite operational loop, where no action is taken. This all occurs after power-on and initial configuration of GPIO and communication interfaces. UART commands trigger an interrupt and bring the PIP to action.

The command takes the form of a single 8-bit command word, which is looked up in an onboard instrument command dictionary, provided for reference in Table 2.3. As can be seen, the binary format of each command is distinctly different from the

Command	ASCII	Hex	Binary	Description
Measure	U	0x55	0b01010101	Single science measurement
Error	*	0x2A	0b00101010	Resend last packet
Shutdown	SI	0x0F	0b00001111	Power down instrument

Table 2.3: PIP command dictionary.

others to avoid accidental bit flips triggering alternative command sequences. A measurement command samples a single 16-oversampled science measurement, an error command requests a resend of the last data packet with a recalculated cyclic redundancy check (CRC - discussed in more detail following) in the case of a spacecraft-received communication error, and a shutdown command prepares the instrument for its supply lines to be shut off by the spacecraft.

Byte #	Bit #	Name	Description
1	1-3	Board ID	Instrument-specific 3 bit serial number
1	4-6	Housekeeping ID	Housekeeping channel identifier
1	7-8	Housekeeping MSB	2 MSBs of 10-bit housekeeping channel
2	1-8	Housekeeping LSB	8 LSBs of 10-bit housekeeping channel
3	1-8	High Gain Channel Byte 1	Upper byte of 16-bit high gain channel
4	1-8	High Gain Channel Byte 2	Lower byte of 16-bit high gain channel
5	1-8	Low Gain Channel Byte 1	Upper byte of 16-bit low gain channel
6	1-8	Low Gain Channel Byte 2	Lower byte of 16-bit low gain channel
7	1-8	CRC Byte 1	Upper byte of 16-bit CRC
8	1-8	CRC Byte 2	Lower byte of 16-bit CRC

Table 2.4: PIP data packet format, with appended CRC-16 bits.

Once the command is received, the appropriate action is performed. In the case of a shutdown command, this just puts the MCU and ADC into a low power state in preparation for the spacecraft to power the instrument supply lines down. In the case of either a measurement or error command, PIP returns a 48-bit science data and 16-bit CRC packet to the LLITED bus. Table 2.4 displays the internal bit contents of a six byte science packet, as well as the appended two bytes of CRC. The 16-bit CRC simply acts as a checksum for the science data, to ensure transmission fidelity and allow for fault tracing. The CRC follows the CRC-16-CCITT-FALSE standard,

with a polynomial $0x1021$ ($x^{16} + x^{12} + x^5 + 1$), a final XOR of $0x0000$, and an initial value of $0xFFFF$.

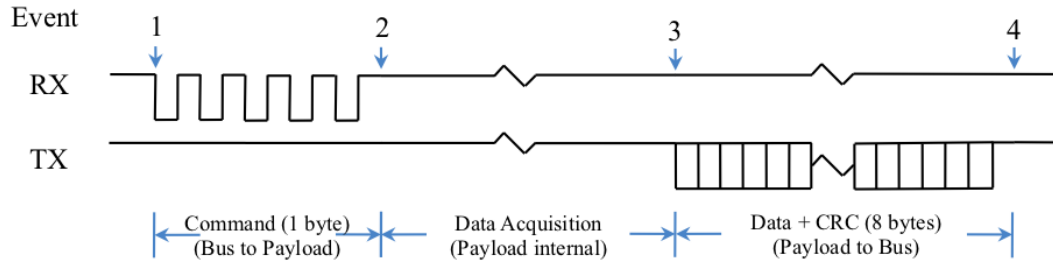


Figure 2.25: PIP interrupt and sample measurement timing diagram showing the sequence of events for command and data transfer (measure command shown).

Figures 2.25, 2.26, 2.27, and 2.28 provide detailed timing diagram and oscilloscope confirmations of measurement timings for a PIP science measurement communication exchange. Note that the instrument was connected to a $56\text{ M}\Omega$ load for all of these measurements. Blue denotes the bus command line and red denotes the instrument TX line. Cursors measure the time duration required for the numbered events in Figure 2.25 to be:

- Event 1-2 = $86.8\ \mu\text{s}$
- Event 2-3 =
 - $5.57\ \text{ms}$ for measurement command
 - $55.8\ \mu\text{s}$ for error command
 - $61.8\ \mu\text{s}$ for shutdown command
- Event 3-4 = $694.4\ \mu\text{s}$

These numbers translate to complete command and response times of $6.37\ \text{ms}$, $837\ \mu\text{s}$, and $843\ \mu\text{s}$ for measurement, error, and shutdown commands, respectively. Since the instrument is asynchronous, the spacecraft operator can lower this polling frequency arbitrarily without affecting PIP functionality. In the case of a prematurely timed command being sent before a PIP conversion is completed, the time for PIP

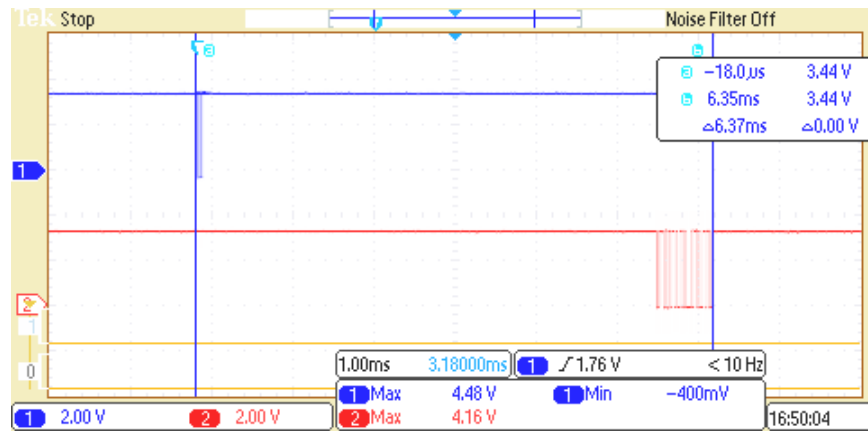


Figure 2.26: Measurement command and instrument response, captured using an oscilloscope.

to send a response to the original command and return to default idling state will simply be delayed by the time it takes the receive interrupt to be serviced, and will not result in the second, premature command being acted upon. Additionally, at 0.837 ms in duration, an error command/response time is fast enough to take place with a 2.793 ms margin following a measure command/response exchange that was in error before the next 100 Hz measurement must be initiated. Thus, the error command can be implemented without delaying subsequent measurements.

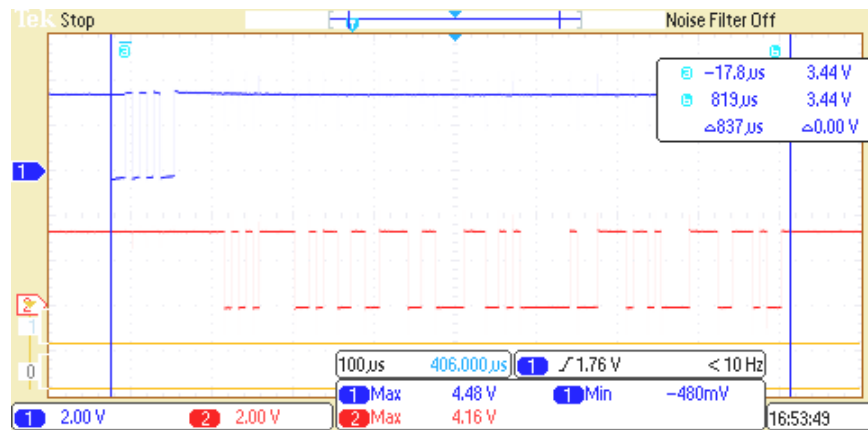


Figure 2.27: Error command and instrument response, captured using an oscilloscope.

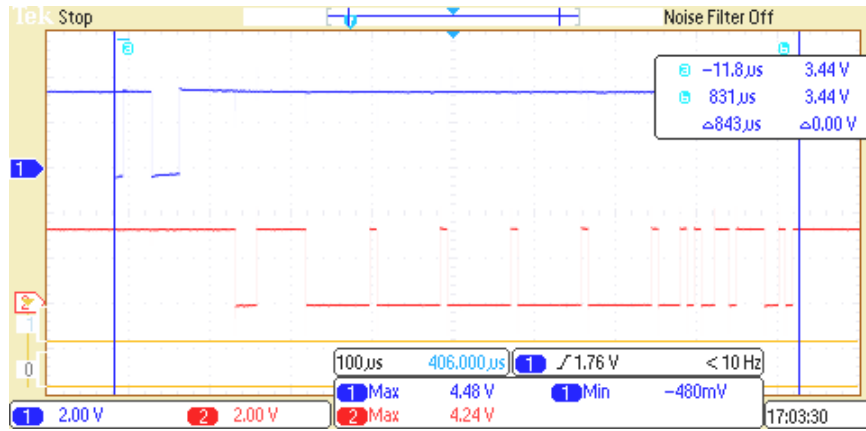


Figure 2.28: Shutdown command and instrument response, captured using an oscilloscope.

2.4 Mechanical Design

The PIP instrument went through several distinct mechanical design changes throughout the instrument development life. Ultimately, an independent, enclosed box concept was chosen for several reasons: it removed an unnecessarily oversized CubeSat board from the avionics stack, surrounded the sensitive instrument circuitry with its own Faraday enclosure, and made the instrument modular. Coupled with the ultra-low power consumption, such modularity allows for effectively drop-in implementation on other similarly sized spacecraft or rockets - varying the physical size of the sensor on the exterior of the vehicle and modifying the feedback resistor tune the application regime.

From the allotted envelope exterior, and designing with a 2 mm Faraday cage wall thickness and 1 mm clearance from board edge to envelope, the PCB was sized to a 24 mm \times 70.2 mm area. In terms of the Faraday enclosure, the middle of the PIP envelope has a cylindrical arc cutout removed in order to provide space for MIGSI. In terms of PIP design, this resulted in a self-imposed “no-go zone” for components near the middle of the bottom layer of the board. Vias and traces were routed through this region (as they are flush with the PCB surface), but ICs and passives were maintained away from this region in order to prevent component contact with the chassis. The design possesses adequate real estate to afford this limitation for

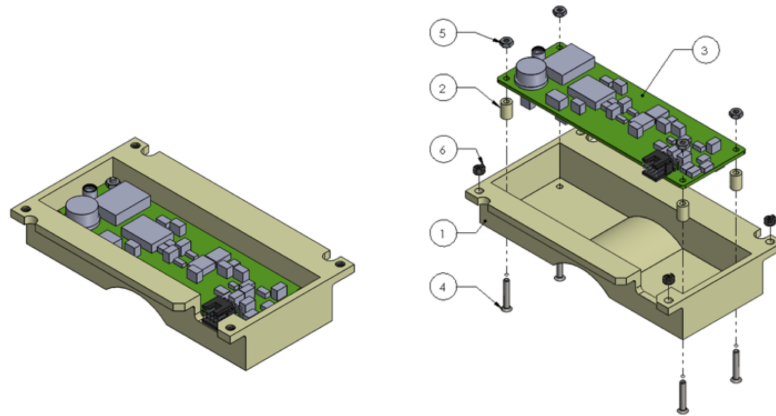


Figure 2.29: PIP's finalized envelope design, acting as a protective Faraday enclosure for the instrumentation.

The Aerospace Corporation. The final envelope design externally measured 30 mm by 76.2 mm and a 13 mm height, with a 2.286 mm tall mounting lip of 43.18 mm \times 76.2 mm extent. Figure 2.29 enumerates the finalized, post-ICD design, and full schematics for the envelope are available in Appendix D.

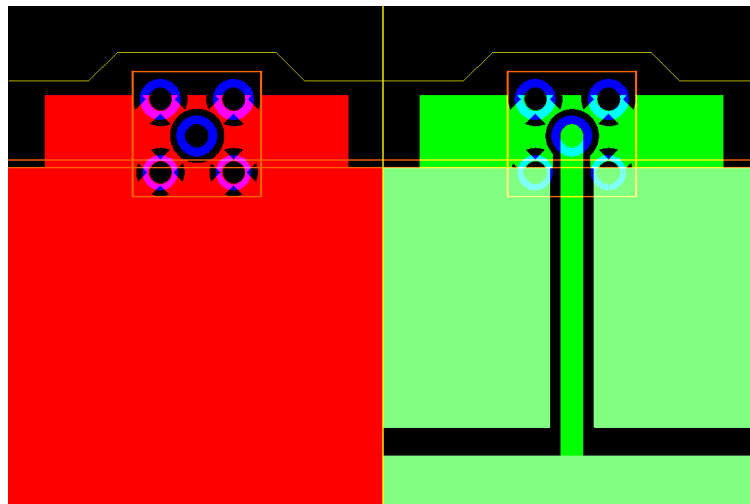


Figure 2.30: PIP sensor board interconnect between MMCX and copper planes magnified, copper bottom left and copper top right.

The sensor design was relatively straightforward. It was decided that since gold plating is widely available commercially from PCB manufacturers, it was simplest

to just utilize a gold-plated PCB on the external of the CubeSat as the sensor, in the form of a 46 mm \times 66 mm rectangular polyimide PCB. The interior of the sensor features a rectangle of 2 cm \times 4 cm dimensions as the Langmuir probe electrode and around this is a 9 mm wide rectangular electrode that acts as the guard. The guard electrode's corners are rounded with 4.5 mm radius corners to keep significant margin from the corner mounting screws. The final design also adopts a small 1 mm \times 7.5 mm protrusion near the MMCX connector to cover the feedthrough cutout in the ram plate.

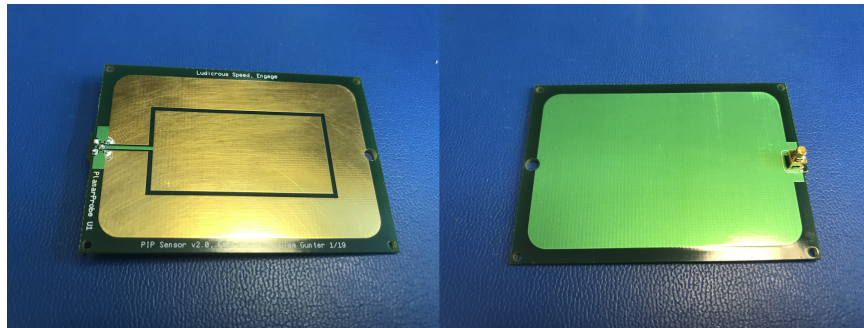


Figure 2.31: PIP sensor board assembly photograph (left is top and right is bottom), finalized for vibration testing.

The sensor board, being exposed to the external space environment, was hard-gold coated for two primary reasons. First, it provides chemical corrosion protection from atomic oxygen present in the LEO environment. Additionally, it provides a consistent photoelectric effect work function across the surface of the probe. This photocurrent effect is not negligible on such an exposed metallic surface, and is accounted for in data analysis by use of light sensor housekeeping telemetry. The gold coating thus provides protection and consistency for the instrument's sensor, though gold's high ratio of absorptivity to emissivity made it slightly more challenging than typical for The Aerospace Corporation to close the thermal problem on the ram plate with PIP.

Between the instrument PCB and sensor PCB, the signal must be routed via a pair of corresponding MMCX connectors. This connector is a through-hole connector, and thus has its 5 pin footprint placed on the far edge of the sensor board. Two of its guard pins are embedded in the exposed guard electrode, while the other two pins

are connected via a plane under the soldermask. All the pins feature thermal relief “x” patterns to make soldering more convenient for technicians. The central MMCX connector is routed to the central electrode via a soldermask protected trace that passes through a 1.524 mm gap in the guard electrode. Figure 2.30 enumerates these features in the finalized sensor board design, and Figure 2.31 is a photograph of the assembled sensor board prior to integration.

Additionally, the most protruding PIP instrument component and directly adjacent to this feedthrough MMCX pair, the AD549 transimpedance amplifier, at 5.842 mm height is 0.339 mm taller than the allocated clearance above the top layer of the board to the ram plate ceiling. Thus, a cutout in the ram plate for the AD549 is also necessary. To accommodate these two components, a generously sized joined cutout in the LLITED ram plate was incorporated into the design. Figure 2.32 visualizes this cutout, available for clearance around the AD549 and MMCX.

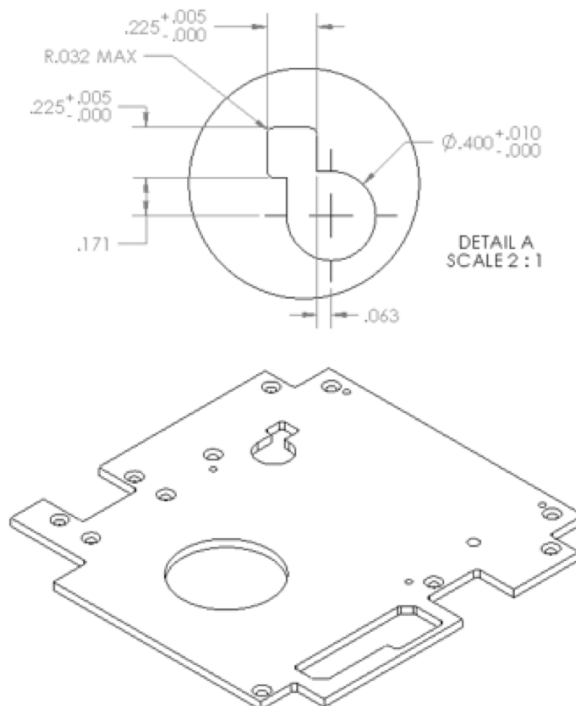


Figure 2.32: The LLITED ram plate with feedthrough detailed, for passing through the MMCX signal connection for PIP and easing clearance concerns on the AD549.

As per Aerospace Corporation specifications, the fasteners used for PIP were all chosen to be United States defense standard compatible. The PCB-to-envelope and envelope-to-ram plate mounting fasteners are both flathead screws inserted into countersunk holes. This provides a flush screw profile for the envelope and ram plate surfaces, simplifying mechanical design and minimizing interference with other components. Within the envelope, the PCB is separated from the envelope floor via 0.1875" mechanical spacers. With the flathead screws, spacers, and narrow nuts, the PCB is secured to the envelope. These contacts also provide a simple circuit connection to chassis ground, as the two 1.778 mm diameter-mounting holes on the digital end are plated with a 0.5 mm annular copper ring that is in contact with the ground plane. Along with the analog components, these two mounting holes are exempted from conformal coating application, in order to maintain conductivity.

Chapter 3

TESTING

3.1 Multisim Simulation Analysis

Prior to v1.3, several analyses were conducted via Multisim simulations in order to gauge the power supply sensitivity and general signal flow for PIP, before physical components were ever tested. For instance, a signal flow test is shown in Figure 3.1. This tested varying a current source feeding into the AD549, feedback resistor, INA2128, and RC filter/diodes. The red and green curves (green is hidden behind the purple) denote the processed ADC input voltages, and they properly translate the input current signal from 2 nA to 20,000 nA into two 0 – 5 V channel signals for digital processing.

Additionally, the power supplies from the LLITED bus are required to be within $\pm 1\%$ and the peak-to-peak power ripple shall not exceed $\pm 1\%$ across 500 kHz to 1.5 MHz for the ± 15 VDC input. The circuitry was simulated with both AC noise in the circuit and offset on these supplies, in order to see any possible ill effects on signal integrity. Figure 3.2 shows an example AC frequency analysis (amplitude top, phase bottom) on the signal lines, and where frequency-varied noise effects the signal the most, with an input signal of 100 nA. Note that the y-axis in the amplitude plot is in dB, so the fact that the processed ADC input signals barely exceed -50 dB implies high noise immunity.

Simulations varying the power supply voltages across their allowed $\pm 1\%$ offset

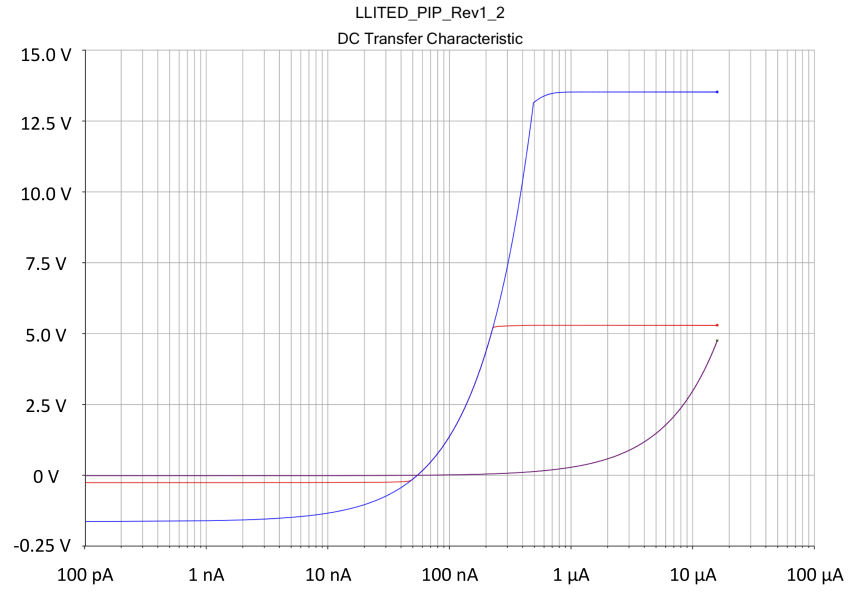


Figure 3.1: PIP Multisim simulation confirming intended signal flow.

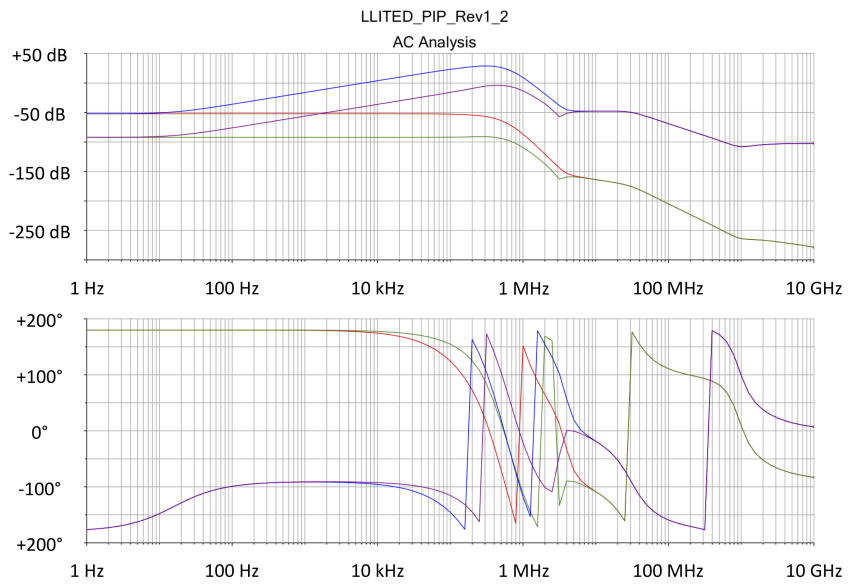


Figure 3.2: PIP Multisim AC frequency simulation, showing dB magnitude and phase in ADC signal.

variation show that the processed voltage signal, regardless of input load, barely exceeds approximately a 1 mV total difference. Figure 3.3 shows variation of the +15 V supply and the separate lines represent variations of the -15 V along their extremes, at a 10,000 nA input current. With such a 1 mV possible maximum variation from power supply offset, this implies the power supplies can impact the ADC processed signal by approximately 13 counts. This was considered an acceptably low impact on the final voltage signal.

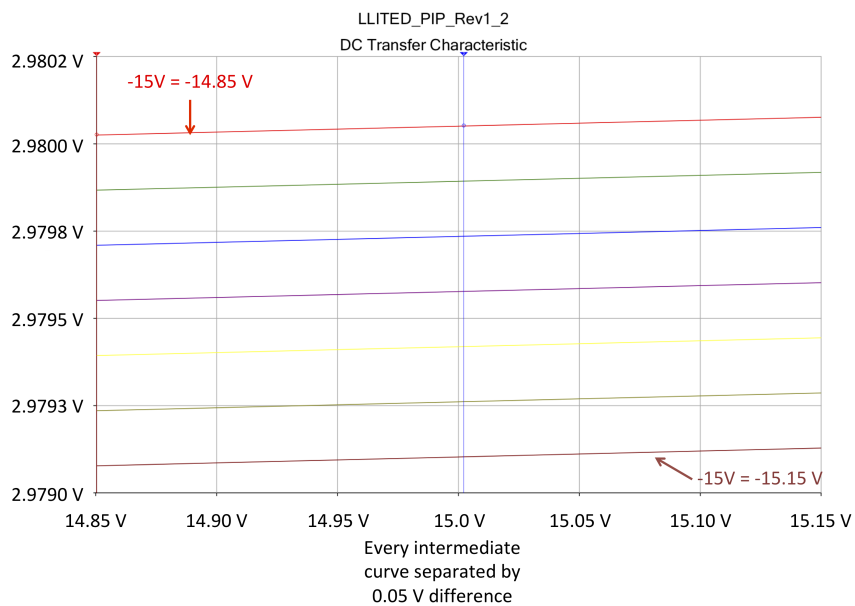


Figure 3.3: PIP Multisim simulation varying power supply offsets from nominal voltage.

3.2 Low Noise Testing

In addition to SWaP conservation, low noise was also of utmost importance in the design of PIP. Lower noise translates to a higher instrument SNR, and also importantly the instrument noise floor forms the effective minimum resolution achievable for measurements. Thus, characterizing the noise level during development was an important indicator of design effectiveness.

3.2.1 v1.3 Oscilloscope/Multimeter Noise Analysis

Since the v1.3 board lacked an onboard ADC or MCU as it was a functional prototype of the analog end and real estate management, noise analysis on this board was conducted using external instruments. An MSO2014 oscilloscope was utilized in order to observe the output signal and any possible noisy frequencies stemming from the voltage supply lines. Purposefully inputting noise into the supplies while observing the output signal was also carried out. A DMM4050 digital multimeter (DMM) with AC/DC voltage measurement capability was utilized to characterize the output signal of the instrument. DC functionality yielded the steady-state output voltages that would be input to the ADC given varying current loads, ensuring that the instrument is properly mapping the two channels to the +5 V ADC range. AC functionality measures the RMS voltage level, whose variation gauges how much noise would be input into the ADC.

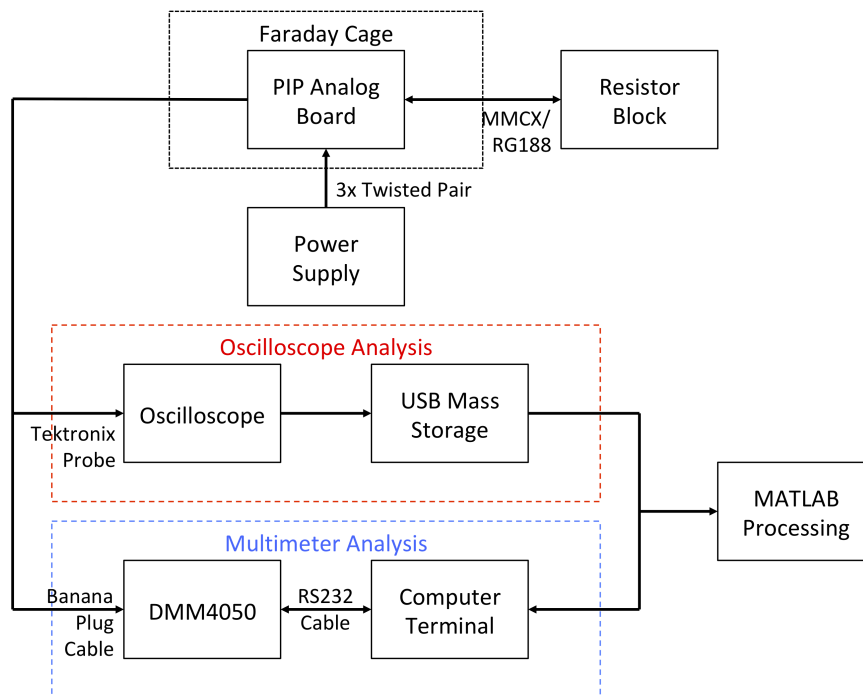


Figure 3.4: PIP noise characterization testing with oscilloscope/multimeter block diagram.

These oscilloscope and multimeter instrument testing schemes are visually illustrated in Figure 3.4, while the oscilloscope, multimeter DC, and multimeter AC data are shown in Figures 3.5, 3.6, and 3.7, respectively. Their results are indicative of properly performing signal processing schemes and acceptably low noise levels. Additionally, the oscilloscope suggests that no major frequency artifacts manifest in the output signal, aside from a small peak on the high gain channel at 50 kHz, a harmonic of the MTeX power supply board utilized during testing for powering PIP.

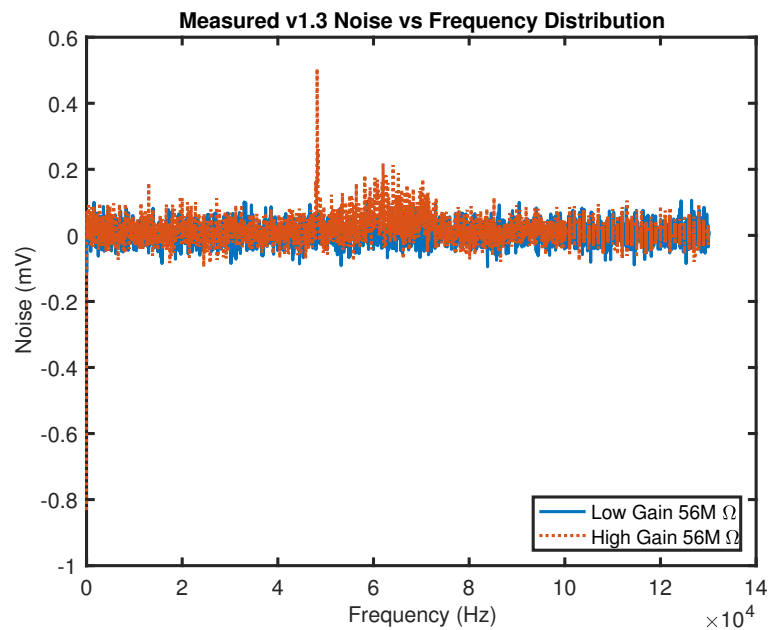


Figure 3.5: PIP v1.3 oscilloscope measured noise versus frequency data.

The oscilloscope data used the Fourier transform function in dBV of the oscilloscope in order to frequency-map possible problems, which was later processed by a Matlab analysis script. This script calculated actual noise voltage from the raw dBV values, and then subtracted data from an unpowered test, in order to remove noise not emanating from operation of the instrument. This measured noise for both channels was effectively below 0.1 mV, which is acceptably small noise, especially given that this setup occurred on an early test board. It also verified that no major unexpected frequencies were causing too much coupled noise.

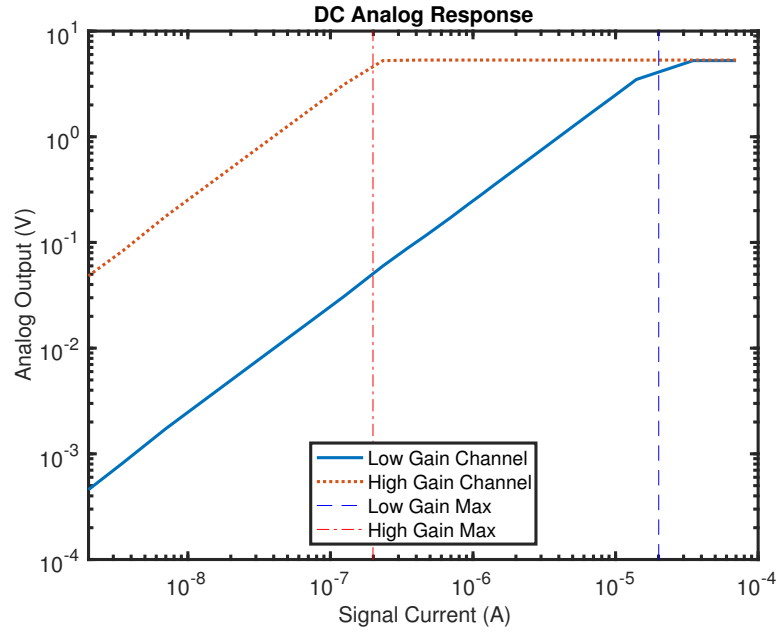


Figure 3.6: PIP v1.3 DC signal characterization.

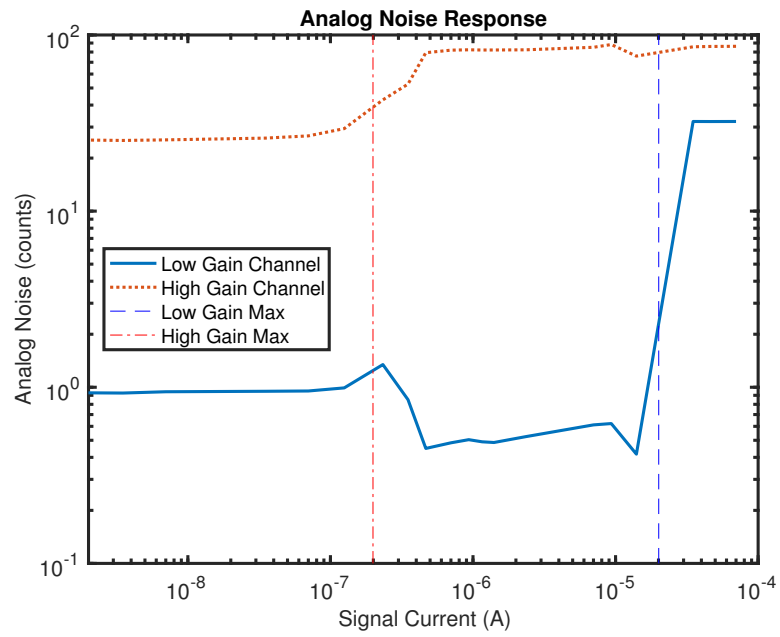


Figure 3.7: PIP v1.3 AC noise characterization.

3.2.2 v2.0, v2.1 and v3.0 Low Noise Verification

With an onboard ADC interfaced with the instrument's MCU, v2.0/v2.1 testing was significantly more straightforward than previous noise testing. A simple Matlab script collecting consecutive instrument readings and instructing the test technician to cycle through the load resistors collected the data. From these recorded values, an average at each resistor load yields the DC curve of output counts versus input signal current and the standard deviation at each resistor yields an instrument noise value across the dynamic range of input signal current. After receiving the v3.0 flight boards from the fabrication/assembly contractor, noise testing was repeated to ensure its low noise characteristics with the updated design, industry standard soldering job, and new polyimide material.

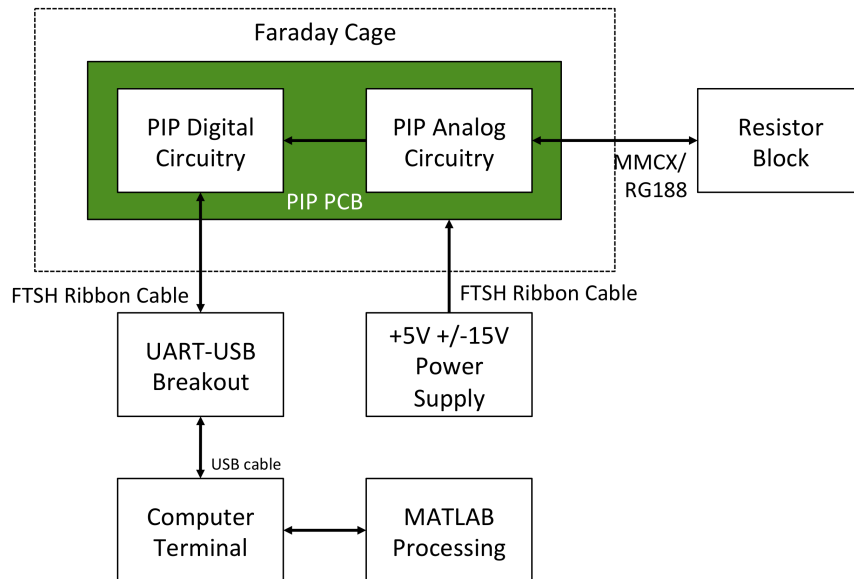


Figure 3.8: PIP noise characterization testing with oscilloscope/multimeter block diagram.

Figure 3.8 shows the testing block diagram for this round of testing, while Figure 3.9 shows the DC transfer profile of PIP, and Figure 3.10 and 3.11 display the noise levels for low and high gain as a function of instrument input current, respectively. It is obvious that the noise levels seen are at least an order of magnitude lower than the intended measurement readings, and thus the noise level is at an acceptable threshold

to tolerate for science operations.

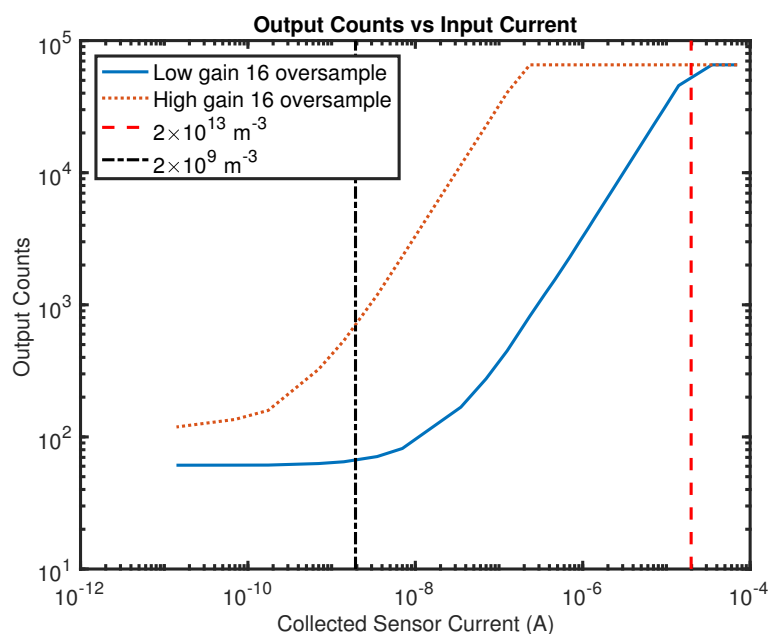


Figure 3.9: PIP v3.0 0b000 instrument DC characterization in terms of input current.

3.3 Power Consumption Characterization

Given the instrument’s low SWaP design policy, as well as the necessity to provide The Aerospace Corporation the instrument’s power draw for CubeSat power system design purposes, characterizing the current draw of the instrument was of utmost importance once the hardware/software design were nearly mature.

Because the PIP instrument must be externally fed well-regulated power supplies (up to ICD standards), a recycled MTeX power board was utilized for generating these voltages during all of development/calibration testing. MTeX power boards are capable of providing the necessary ± 15 V and +5 V voltages by properly adjusting board settings via a jumper resistor.

Measuring the current draw consisted of attaching a multimeter in parallel with the instrument on its supply voltage lines while operating the instrument at 100 Hz measurement rate (measurement command provided from an external MSP430FR4133

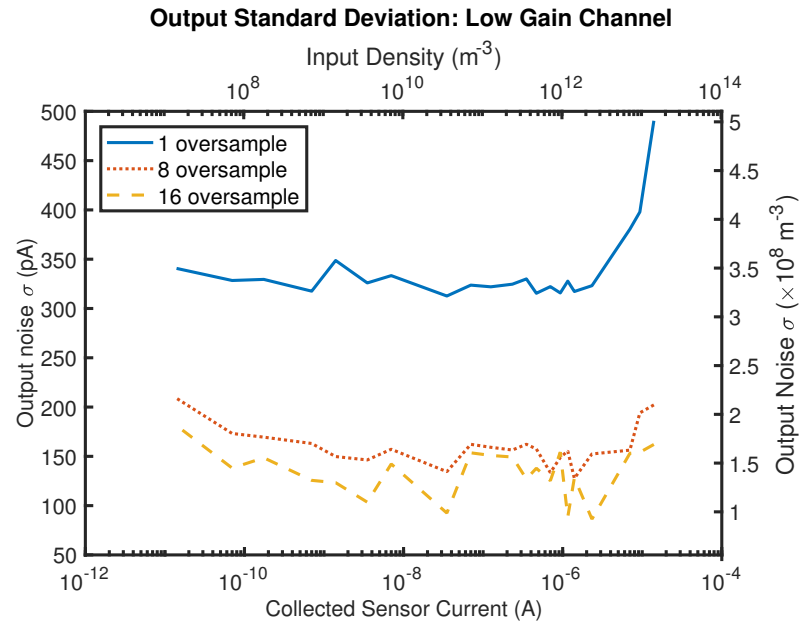


Figure 3.10: PIP v3.0 0b000 low gain noise in terms of input current.

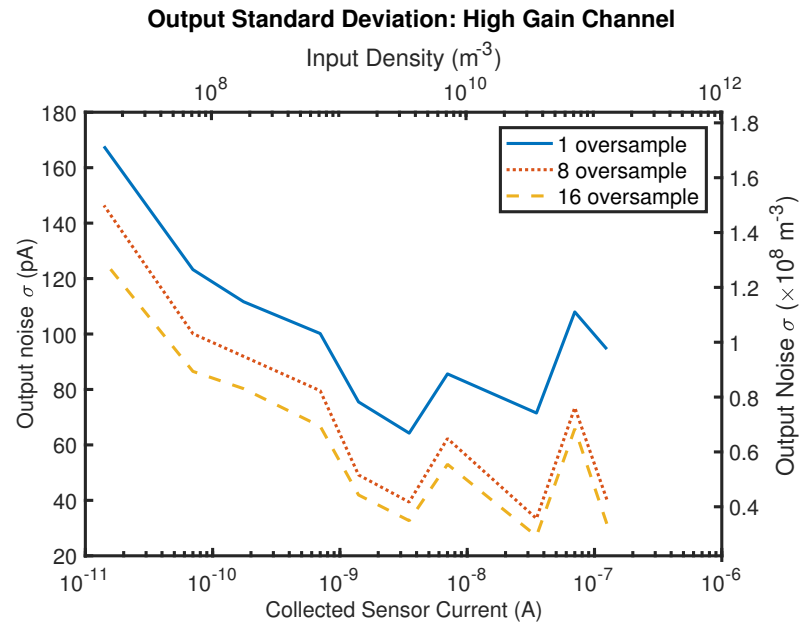


Figure 3.11: PIP v3.0 0b000 high gain noise in terms of input current.

development board). Power draw is obviously dependent upon the signal load PIP experiences, and thus characterizing the power consumption properly involves varying the input signal load. The ammeter measurement process was repeated for the three supplies and over several different resistor loads.

Figure 3.12 displays the block diagram setup for this power consumption testing, Figure 3.13, while Table 3.1 lists selective current draws for each supply. See *PIP_PowerConsumption.xlsx* for the full consumption study data. Note that for these tests, the supply voltages were measured to be +4.95 V, +14.89 V, and -14.83 V. Thus, the calculated total power draw for each supply and for the instrument in total is also included in this table.

As is obvious in Figure 3.13, the total power consumption is at or below ~ 140 mW for nearly all of the dynamic range of the instrument. When 10^{13} m^{-3} densities are reached, however, this power draw increases dramatically toward the instrument power requirement maximum of 250 mW at the end of the dynamic instrument range.

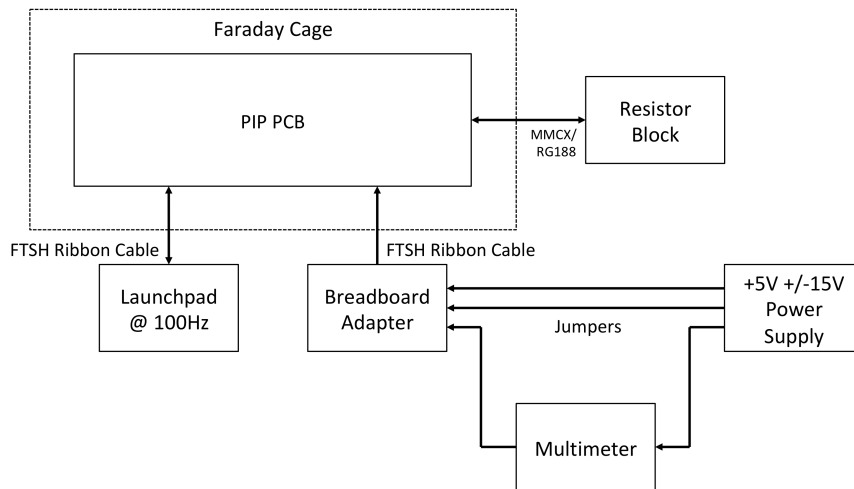


Figure 3.12: PIP power consumption testing block diagram.

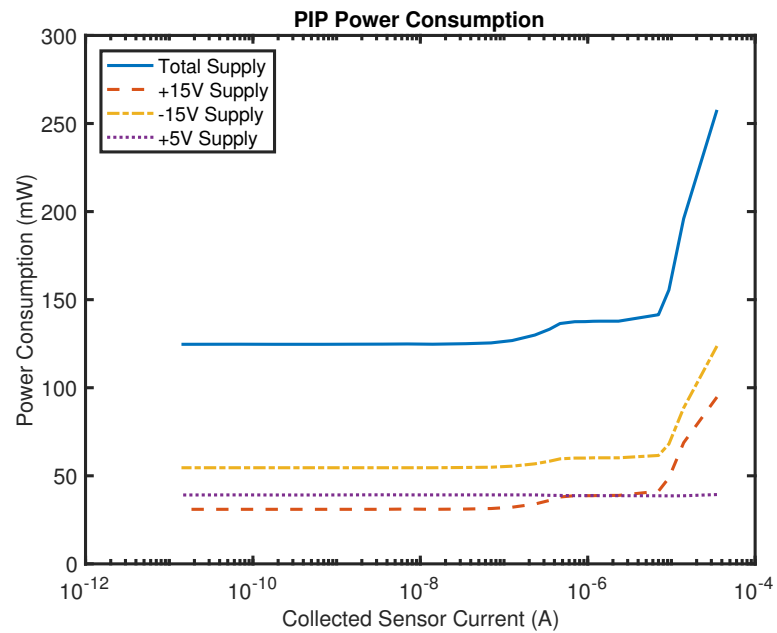


Figure 3.13: PIP power consumption, given input load signal current.

Load (nA)	-15 V (mW)	+15 V (mW)	+5 V (mW)	Total (mW)
14,000	88.5351	68.7918	38.6595	195.9864
125	55.4642	32.1624	39.1545	126.7811
7	54.5744	31.1201	39.1545	124.8490

Table 3.1: PIP power consumption for each supply and total for selected loads.

3.4 Voltage Supply Permutation Survival/Inrush Current Testing

Similar to the Section 3.3, the instrument had to be tested to ensure it could survive any permutation of voltage supply activations. Additionally, circuitry typically has an inrush current associated with the transient power switching process, where the electronics consume larger currents than typical operation while they are initializing. Measuring this inrush current for PIP and ensuring its survival given any supply voltage activation permutation was also tested. The collected information was used to inform The Aerospace Corporation on the most efficient way to activate the power supplies from a power draw perspective.

A DMM6500 multimeter was used for collecting inrush current data. The basic setup consisted of this multimeter in parallel with a supply at a time in ammeter configuration. Stepping through all the supply permutations, the line of interest was disconnected and reconnected three times each by the technician. The visual display on the touchscreen LCD of the DMM6500 allowed for confirmation of the connection and inrush. A simple USB storage device was used to offload the data for further manual Excel processing.

With these tests iterating several power cycles using every activation permutation, it was proven that the instrument can survive any supply line activation order. Matlab was also used to poll the instrument during these tests, so that data/housekeeping readings were recorded indicative of all the partially powered states. In this way, interoperability testing or on-orbit troubleshooting can be identified if the instrument is in a partially powered error state.

Figure 3.14 shows a block diagram of this setup. The complete collection of data is enumerated in the *InrushData.xlsx* and *InrushAnalysis.docx* documents, but Figure 3.15, 3.16, and 3.17 show the inrush for the +15 V, -15 V, and +5 V supplies respectively, given the Embry-Riddle recommended activation sequence in the same order. Blue, orange, and yellow curves represent the three trials for each test. Note that these assume expected mid-density (10^{11} m^{-3}) input currents, and that the time of 0 s corresponds with the initial current peak where the inrush is assumed to begin.

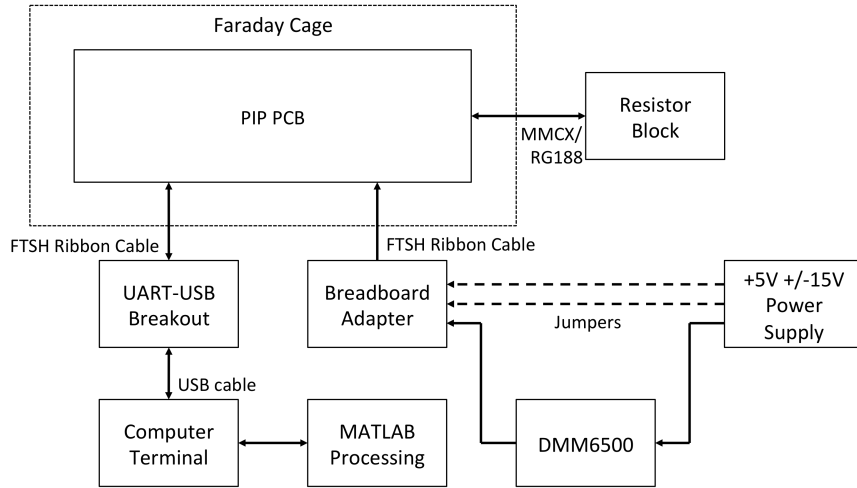


Figure 3.14: PIP inrush current testing block diagram.

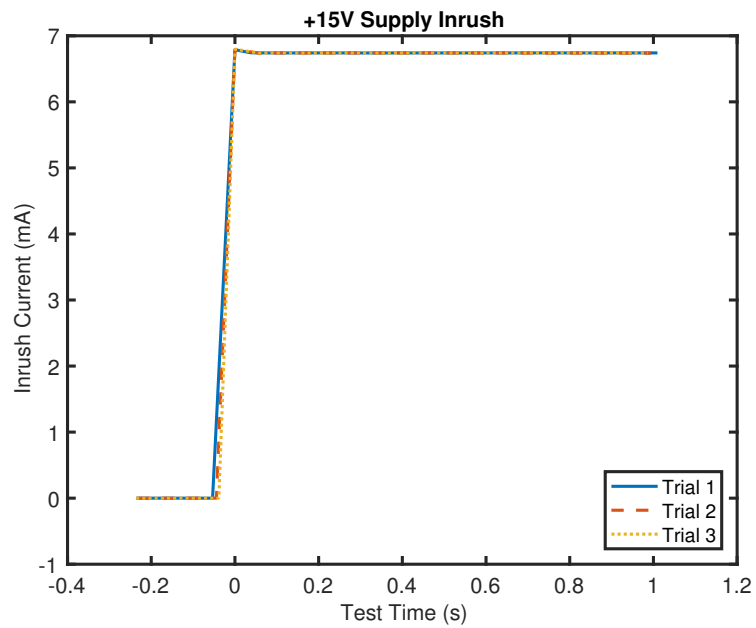


Figure 3.15: PIP +15 V supply inrush current.

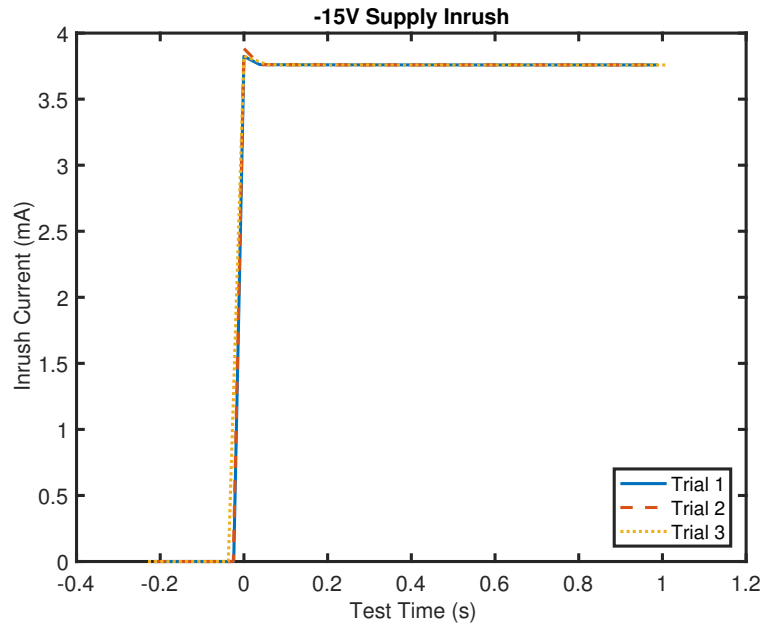


Figure 3.16: PIP -15 V supply inrush current.

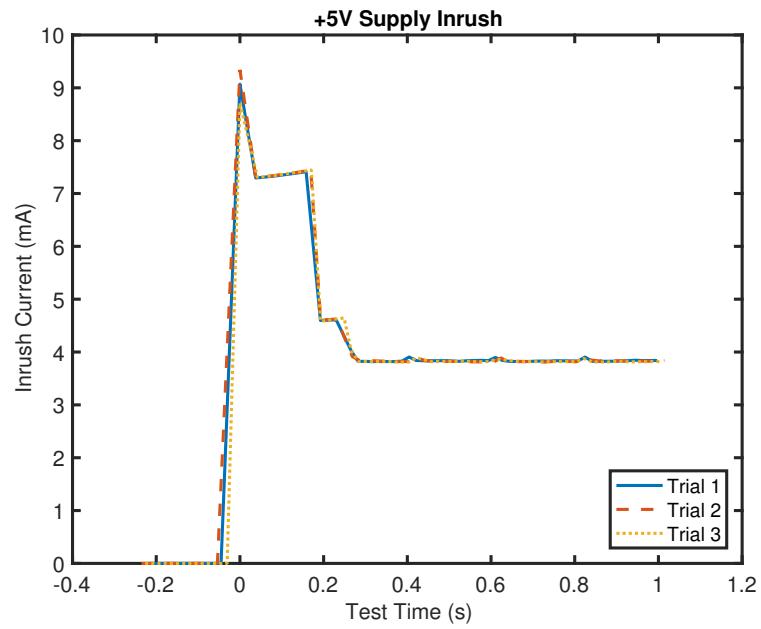


Figure 3.17: PIP +5 V supply inrush current.

Table 3.2 summarizes the information from this inrush testing by specifying maximum current, steady-state average current, and the not to exceed duration of the inrush, for both mid-density and high-density (10^{13} m^{-3}). Note that not to exceed duration is measured from the initial current peak to the steady state current region.

Density (m^{-3})	Supply	Peak (mA)	Average (mA)	Not To Exceed (ms)
10^{11} m^{-3}	+15 V	6.9	6.8	60
10^{11} m^{-3}	-15 V	3.9	3.8	60
10^{11} m^{-3}	+5 V	10	4	300
10^{13} m^{-3}	+15 V	6.9	6.8	60
10^{13} m^{-3}	-15 V	8.5	8.4	60
10^{13} m^{-3}	+5 V	10	4	350

Table 3.2: PIP inrush current, steady-state current, and not to exceed inrush duration for recommended power sequence.

3.5 Temperature Survival Testing

In order to ensure that the flight boards can survive the thermal cycling that LLITED will see on-orbit, a survival test procedure was conducted on all flight boards. This section provides insight into that procedure conducted.

A TestEquity 115A environment chamber was used for the thermal testing of the PIP assembly. This chamber is capable of executing pre-programmed temperature profiles across a temperature range of -73°C to $+175^\circ\text{C}$. This chamber is used for PIP flight instrument survival and temperature calibrations. A dry air purge is used to ensure condensation does not develop at low temperatures. Figure 3.18 explains the block diagram of the test setup.

After input from The Aerospace Corporation and internal discussion, the PIP v3.0 thermal survival testing profile is designed to the cycle displayed in Figure 3.19, with programmed chamber temperature, functional temperature extremes, and locations of functional tests listed. As shown it dwells significantly at survival temperature extremes for long durations at the beginning and end, while repeating 20 short duration cycles reminiscent of orbital temperature swings in the middle.

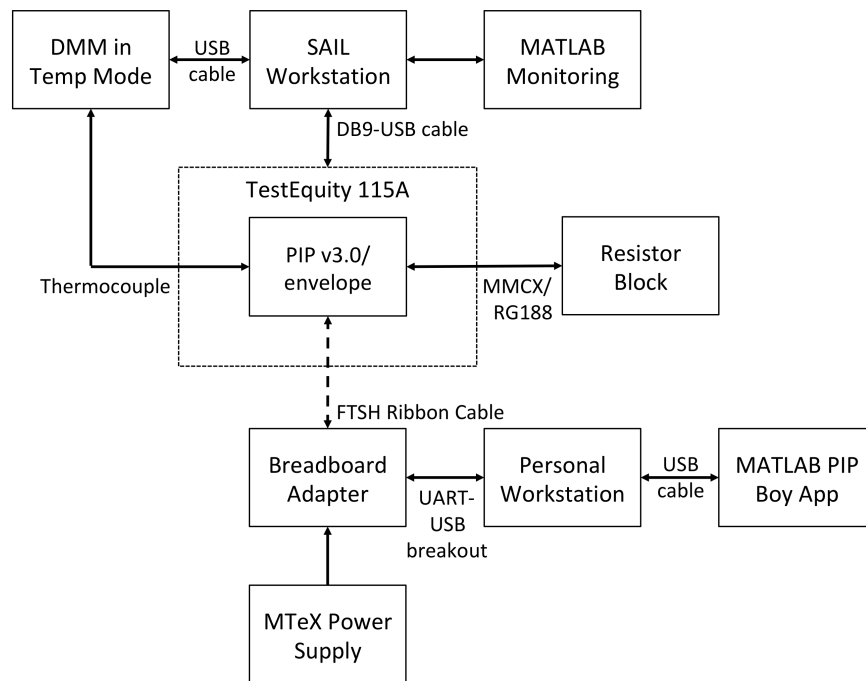


Figure 3.18: PIP survival testing block diagram.

During initial survival testing, Matlab was used to capture the temperature profiles from the TestEquity 115A output while also obtaining separate, DMM6500 multimeter-based measurements (in temperature mode with a type K thermocouple attached inside the PIP envelope). The Matlab script necessary to do so used the DMM6500 multimeter as a USB VISA object and the TestEquity 115A as a Modbus object. Measuring the temperatures consisted of simply requesting a “:READ?” query from the DMM, and reading Modbus holding register 101 on the 115A. This data was collected to ensure that the actual temperature the instrument experienced met the testing requirements. Figure 3.20 shows the first successful test data, clearly indicating that the payload temperature is settled with the chamber temperature and that the extremes were reached reliably for all cycles and dwell periods.

Functional tests performed at specified dwell periods ensured reliable functionality of the first instrument tested throughout the duration of the test. Following this, the rest of the boards were simply cycled in these temperature extremes and tested after the entire test was complete. All boards were found to be functional after the tests.

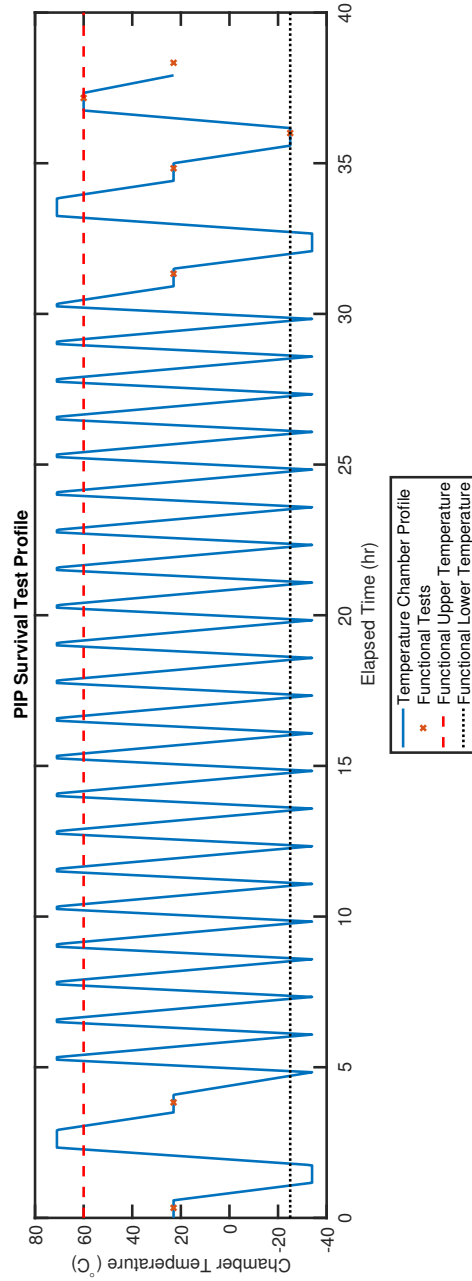


Figure 3.19: PIP survival test theoretical temperature profile.

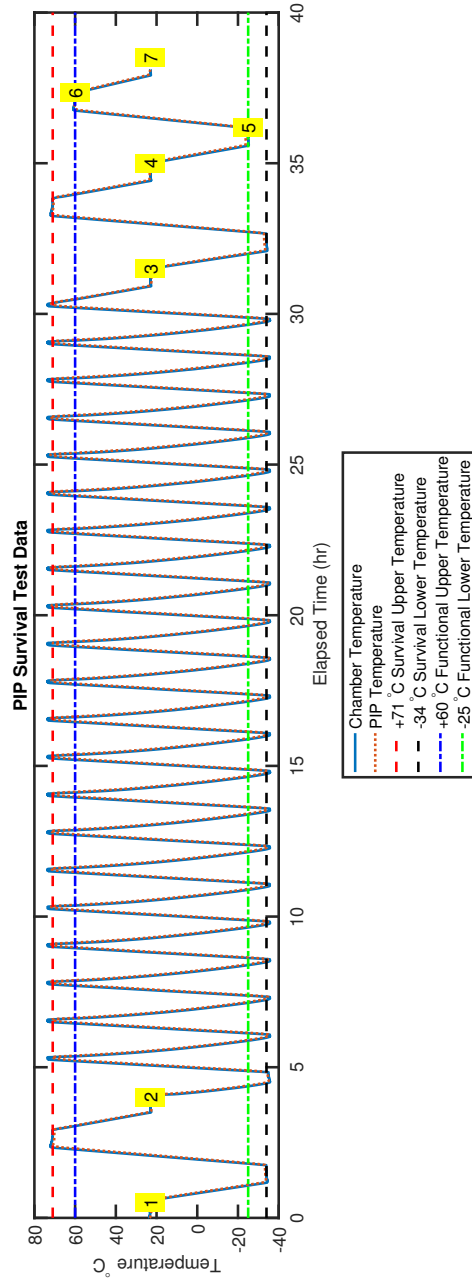


Figure 3.20: PIP survival test temperature data from the first successful test.

Chapter 4

CALIBRATION

4.1 Environment Chamber Temperature Calibration

The same TestEquity 115A environment chamber as used in Section 3.5 was utilized for calibrating the science data behaviour of the flight boards in terms of temperature. Due to the boards' intended environment of the LLITED CubeSat, a calibration across the entire functional thermal environment is central to ensuring that the PIP behaviour is predictable and properly accounted for.

This process is accomplished in two stages: a bias calibration which characterizes the instrument bias voltage as a function of temperature, and the calibration proper where the loads are varied while temperature is also varied. Each of these steps has a Matlab script effectively continuously polling the PIP instrument during the thermal chamber cycling, and a second analysis script that processes these and forms fit calibration values for the collected datasets.

Figure 4.1 block diagrams the bias calibration process, where the bias is measured using an external DMM6500 operated over USB throughout a calibration temperature profile subjecting the instrument throughout its functional -25°C to $+60^{\circ}\text{C}$. This entire time, measurements are taken with PIP, so that onboard housekeeping temperature channels are recorded throughout the thermal cycle. The post-analysis then uses the data to form a fit function that approximates the bias as a function of the two temperature channels to within acceptable tolerances. The bias V_{bias} fit is

found given fit coefficients K_- with the equation,

$$V_{bias} = (K_{offset}) + (K_{T_1})(T_1) + (K_{T_2})(T_2) + (K_{T_1^2})(T_1^2) + (K_{T_2^2})(T_2^2), \quad (4.1)$$

where T_- denotes temperature channel counts.

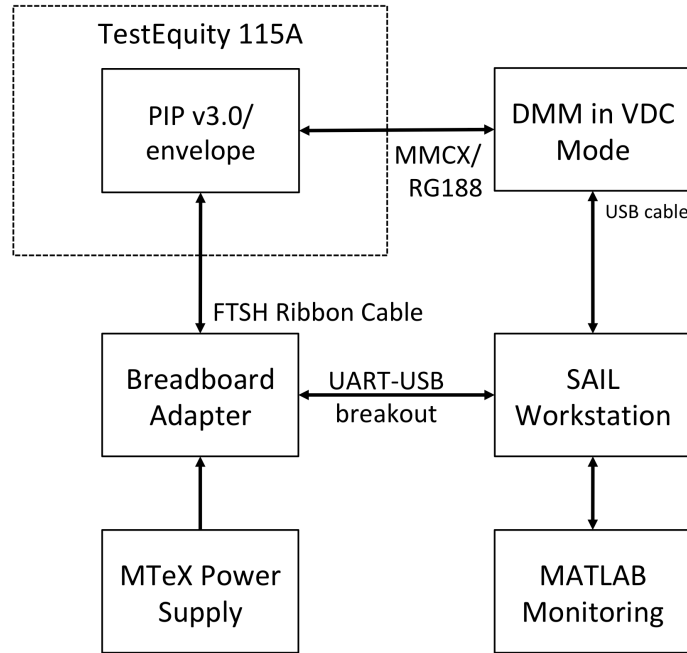


Figure 4.1: PIP bias calibration testing block diagram.

As can be seen, the chosen fit equation utilizes both linear and square terms for the temperature channel terms, in order to more accurately represent the slightly nonlinear edges of the bias variation with temperature. Including just linear terms resulted in a very similar error, but square was ultimately chosen for the aforementioned reason. Figure 4.2 shows both linear and square bias calibration fit error from theoretical for the board #0b100 instrument. With a fit error not exceeding $+100/-80 \mu\text{V}$, the fit predicts the bias potential to within 0.0015% accuracy. This implies high confidence in the fit, and therefore the calculated load current produced by the bias voltage with the high accuracy resistor block during actual calibration.

Figure 4.3 block diagrams the calibration process, where the input load is varied across 6 resistors and 500-measurement bursts with PIP are taken on these different

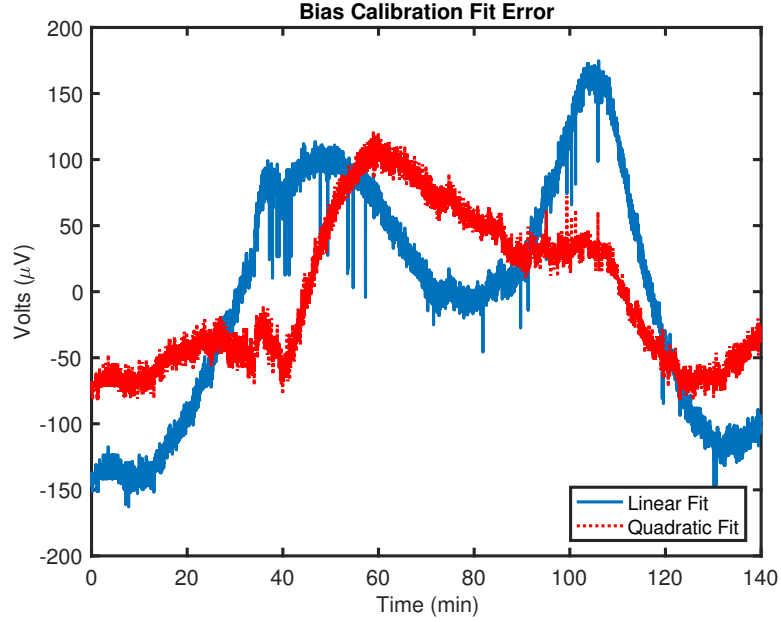


Figure 4.2: PIP bias calibration fit error on board #0b100 instrument.

loads, collecting science counts and all housekeeping channels simultaneously. While this process is repeated as many times across the 6 resistors analyzed as time allows the user, the thermal chamber is also taken through the same calibration thermal cycle as used in the bias calibration. Post-processing uses high-accuracy resistor measurements and predicted bias values, along with the collected science counts in order to fit the collected data. In the end, current I_{meas} as a function of science counts and both housekeeping channels is calculated, according to the relationship,

$$\begin{aligned}
 I_{meas} = & (K_{offset}) + (K_{ch})(C_{ch}) + (K_{T_1})(T_1) + (K_{T_2})(T_2) \\
 & + (K_{C_{ch}^2})(C_{ch}^2) + (K_{T_1^2})(T_1^2) + (K_{T_2^2})(T_2^2) \\
 & + (K_{C_{ch}^{0.5}})(C_{ch}^{0.5}) + (K_{T_1^3})(T_1^3) + (K_{T_2^3})(T_2^3), \quad (4.2)
 \end{aligned}$$

where K_- denotes various calculated coefficients, C_{ch} is science channel counts, and T_- is temperature channel counts. This fit is ensured to be acceptably within a certain level of error from the actual value.

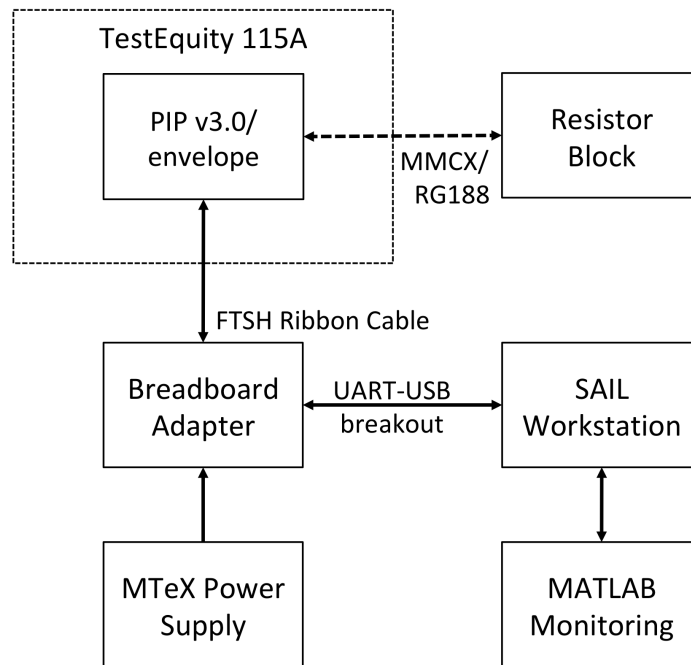


Figure 4.3: PIP instrument calibration testing block diagram.

Post-processing involves the Matlab script prepares data for fitting. In this step, the prior recorded 20-byte calibration packets from PIP are parsed into temperature and science data. The last 200 samples of each resistor trial are averaged into 4 data points to remove noise, and then a calculated current is obtained for each of these datapoints from predicted nonlinear bias fit and accurate resistor value. The resulting data vectors are then fit with the calculated current. As can be seen, the nonlinear fit equation has square and cubed temperature and square and square-rooted science channel terms (in addition to the linear temperature/science channel terms). That said, both a linear fit and this nonlinear fit are calculated and can be used depending on analysis preference. Figures 4.4 and 4.5 display the errors stemming from these for the averaged datapoints used for the original fit for low and high gains, respectively. Note that the curves are sawtooth patterned because there are four datapoints for each resistor and the resistors have significantly different errors.

It is obvious that the nonlinear fits for the instrument can has an error on the low gain channel that does not exceed ± 40 nA and an error on the high gain channel of less

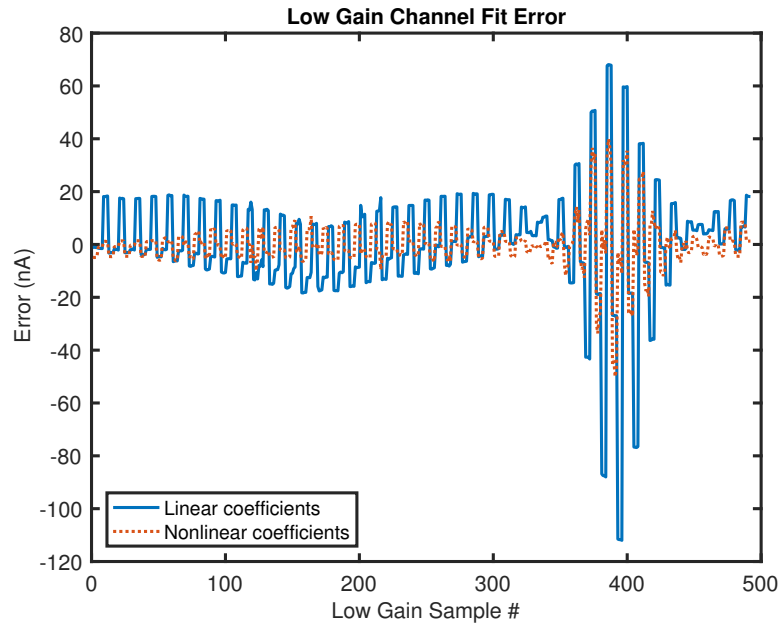


Figure 4.4: PIP calibration fit error on board #0b000 instrument low gain channel.

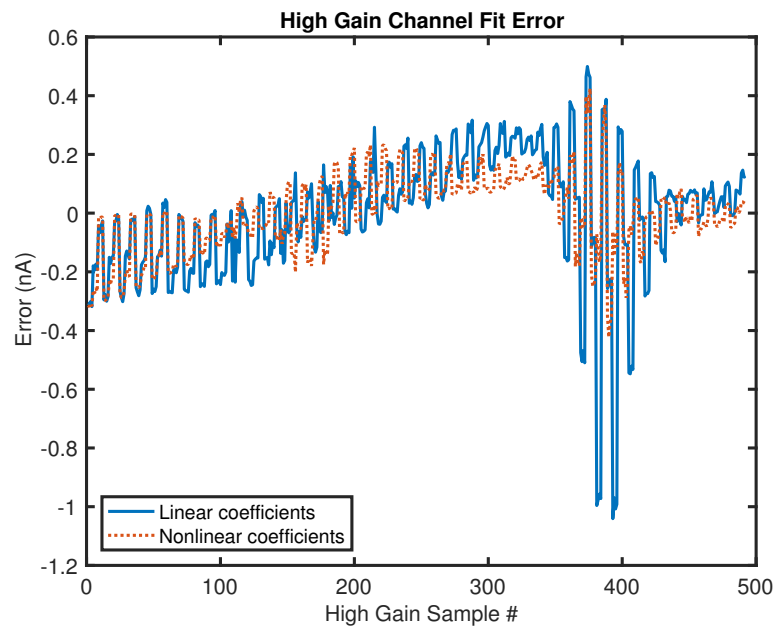


Figure 4.5: PIP calibration fit error on board #0b000 instrument high gain channel.

than ± 0.4 nA. These represent errors an order of magnitude lower than the region of interest for each channel, and are considered adequate for science measurement purposes.

Note that instrument board #0b000 in particular has a transient glitch where temperature channel 2 at approximately below freezing will occasionally read 0d639. A running average for temperature housekeeping can fix this problem, and other instruments do not manifest this problem regardless, so the problem is ignorable. This is observable when we use the fit equations on the full dataset recorded (no smoothing averages to accommodate transient problems). That said, the fits still do extremely well at maintaining error effectively an order of magnitude below the bottom of the intended dynamic range of each channel, as observed respectively for low and high gain in Figures 4.6 and 4.7.

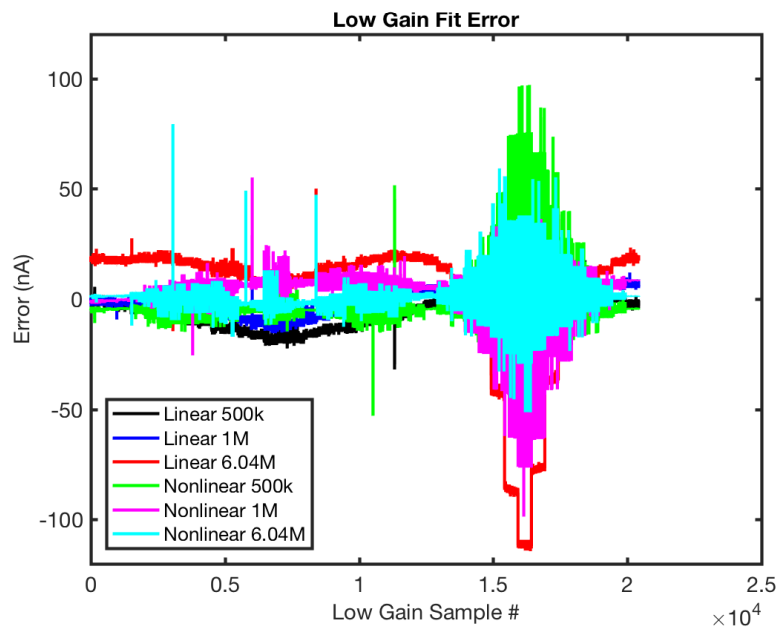


Figure 4.6: PIP calibration fit error on board #0b000 instrument full low gain channel.

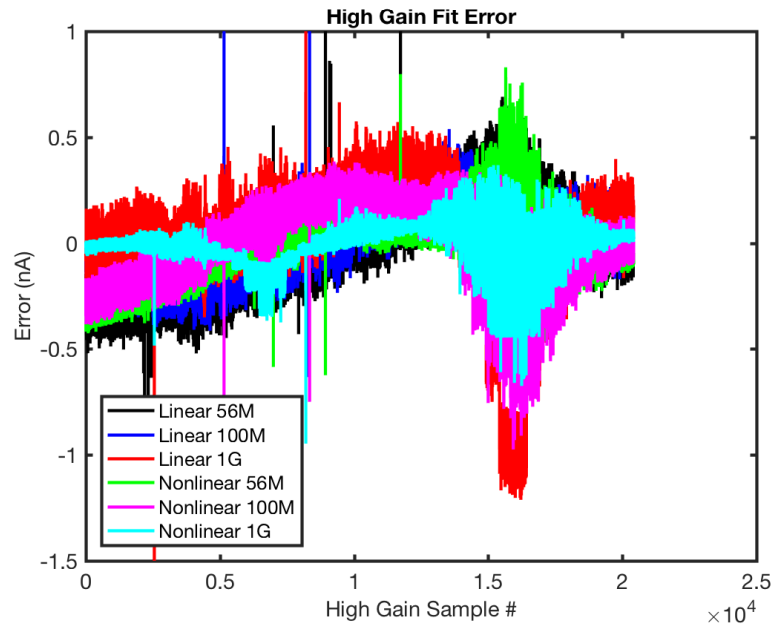


Figure 4.7: PIP calibration fit error on board #0b000 instrument full high gain channel.

Chapter 5

CONCLUSIONS

This thesis has enumerated and culminated in the development and testing of the Embry-Riddle PIP instrument built for the LLITED Mission. PIP is an ultra low SWaP and low noise instrument to measure in-situ absolute ion density. Along with the MIGSI and CTECS instrumentation, the mission intends to construct a complete dataset of information pertaining to the ETWA for anchoring models and future investigations of the phenomenon. As of this writing manifested for launch in 2020, the pair of LLITED CubeSats have a planned initial six month mission operations period, with at least three years of operational lifetime expected. Thus, long term observation of the ETWA with high (80 m) spatial resolution and temporally (45 min) separated spacecraft will be carried out.

This thesis involved mechanical, analog, digital, software, and printed circuit board design. Extensive electrical and thermal testing was also carried throughout all stages of development, with a final calibration characterizing the flight instruments across the entire operational temperature range. The information and data surrounding these designs and tests are documented in the appendix of this thesis and in the SAIL OneDrive cloud file storage system.

The PIP instrument has been designed specifically with modularity and future expansion of capability in mind. Beyond LLITED, the instrument is capable of being deployed on other vehicles, CubeSat and sounding rocket alike. The instrument's low power consumption, small physical footprint, simple electronics, nearly universal

serial interface, and bus-determined sample rate make PIP an excellent candidate for being applied as a patch to the exterior of multitudes of spacecraft. Identical instruments applied to opposing spacecraft surfaces can enable absolute ion density measurements on spinning/tumbling spacecraft, as well as allow for spacecraft wake structure experiments.

The entire development was an incredible learning experience. Among the more generic comments that can be made like how the author's engineering skills, including spreadsheet analysis, circuit simulation techniques, low noise design techniques, PCB fabrication industry-standard practices, proficiency with embedded C software, use of bench-top test equipment, and an appreciation for the difficulty of spacecraft design and systems engineering were advanced, there were several key lessons learned from the PIP development. The iterative nature of the engineering process was incredibly important, exemplified in particular by the nature of the mechanical envelope evolution; Embry-Riddle and the Aerospace Corporation exchanged months of communications to settle on the finalized envelope design of PIP, spanning four major concept designs. This envelope design, of course, determined the PIP PCB size, MMCX spacing, FTSH connector selection and placement, and ram plate design, so there was also a large feedback component of this iterative process. Furthermore, the electrical circuitry design went through two Multisim simulated revisions and four physically implemented revisions, while the gold-plated sensor board went through three physically implemented revisions. Additionally, while PIP development provided soldering and prototyping practice, contracting the assembly of the flight boards demonstrated how project funding can be exchanged to reduce product development and deployment time, to great effect.

Recommendations for follow-on development to this design would include adding a feedback capacitor to the transimpedance amplifier, improving the RC filter into ADC system so that software-programmed delays are unnecessary (possibly through a different instrumentation amplifier or a larger flywheel capacitor, making sure to maintain the chosen cutoff frequency), and implementing GPIO controlled relay switches that can toggle the analog ± 15 V lines to the amplifiers to improve average power consumption.

Bibliography

- Amphenol Connex. *MMCX P.C.B. Mount Jack*, nc edition, 3 2003. Drawing.
- Amphenol Connex. *MMCX P.C.B. Mount Plug*, 5 2006. Drawing.
- Analog Devices. *AD549 Ultralow Input Bias Current Operational Amplifier*, k edition, a. Datasheet.
- Analog Devices. *AD590 2-Terminal IC Temperature Transducer*, g edition, b. Datasheet.
- Analog Devices. *ADP3300 High Accuracy anyCAP 50 mA Low Dropout Linear Regulator*, c edition, c. Datasheet.
- Aroh Barjatya. *Langmuir Probe Measurements in the Ionosphere*. PhD thesis, Utah State University, 2007.
- Rebecca L. Bishop, James Clemmons, and Aroh Barjatya. Low-latitude ionosphere/thermosphere enhancements in density (llited) mission. Technical report, The Aerospace Corporation, El Segundo, CA, 2016. Proposal documentation.
- J. H. Clemmons, R. L. Walterscheid, A. B. Christensen, and R. L. Bishop. Rapid, highly structured meridional winds and their modulation by non migrating tides: Measurements from the streak mission. *Journal of Geophysical Research: Space Physics*, 118(2):866–877, 2013.
- Diodes Incorporated. *BAT54 Surface Mount Schottky Diode*, 32-2 edition, 11 2016. Datasheet.

Dupont. *Dupont Kapton Summary of Properties*, 2017. Datasheet.

Keithley. *Low Level Measurements Handbook*. Tektronix, 7 edition, 2014.

Irving Langmuir and H. M. Mott-Smith. Studies of electric discharges in gas at low pressures. *General Electric Review*, page 616, 1924.

Jiuhou Lei, Jeffrey P. Thayer, and Jeffrey M. Forbes. Longitudinal and geomagnetic activity modulation of the equatorial thermosphere anomaly. *Journal of Geophysical Research: Space Physics*, 115(A8), 2010.

Linear Technology. *LT1021 Precision Reference*, 1021fc edition. Datasheet.

Osram Opto Semiconductors. *SFH5711 High Accuracy Ambient Light Sensor*, 1.1 edition, 9 2013. Datasheet.

Samtec. *FTSH-1XX-XX-XX-D-XX-XX*, bo edition. Drawing.

Edward P. Szuszczewicz. Area influences and floating potentials in langmuir probe measurements. *Journal of Applied Physics*, 43(3):874–880, 1972.

Texas Instruments. *INA2128 Dual, Low Power Instrumentation Amplifier*, sbos035a edition, 4 2007. Datasheet.

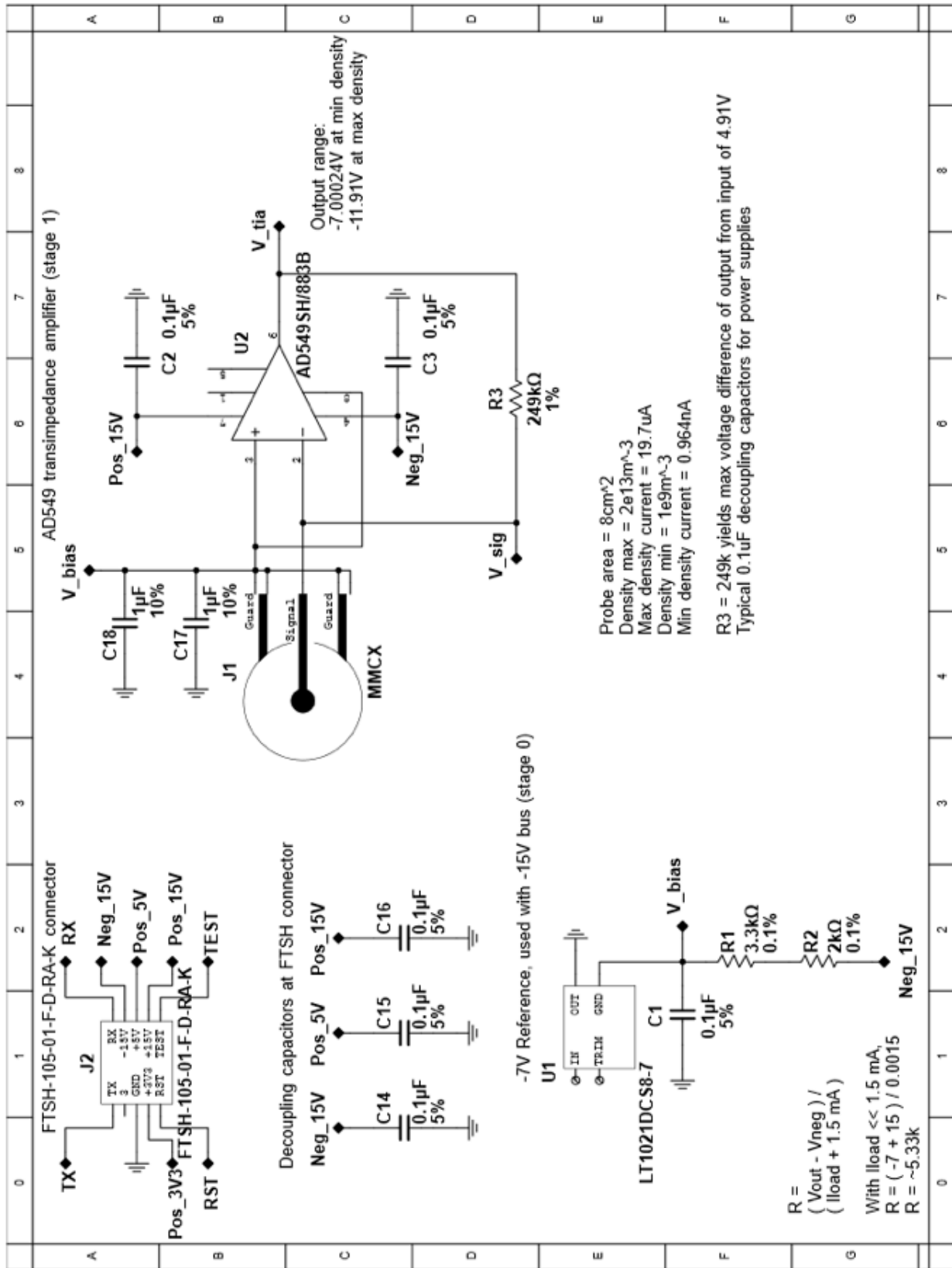
Texas Instruments. *ADSxx53 Dual, High-Speed, 16-, 14-, and 12-Bit, Simultaneous-Sampling, Analog-to-Digital Converters*, sbas584b edition, 8 2014a. Datasheet.

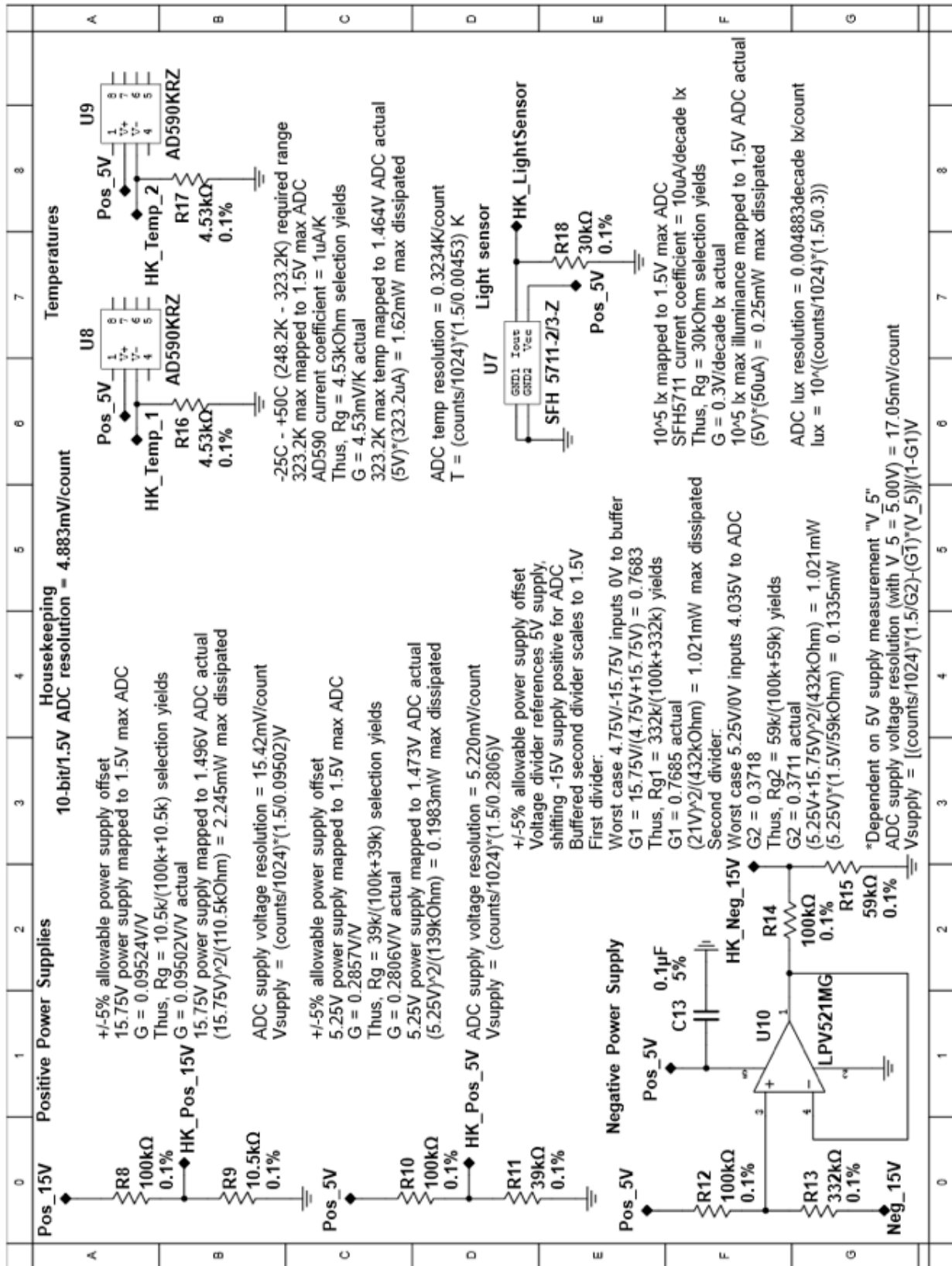
Texas Instruments. *LPV521 NanoPower, 1.8-V, RRIO, CMOS Input, Operational Amplifier*, snosb14d edition, 12 2014b. Datasheet.

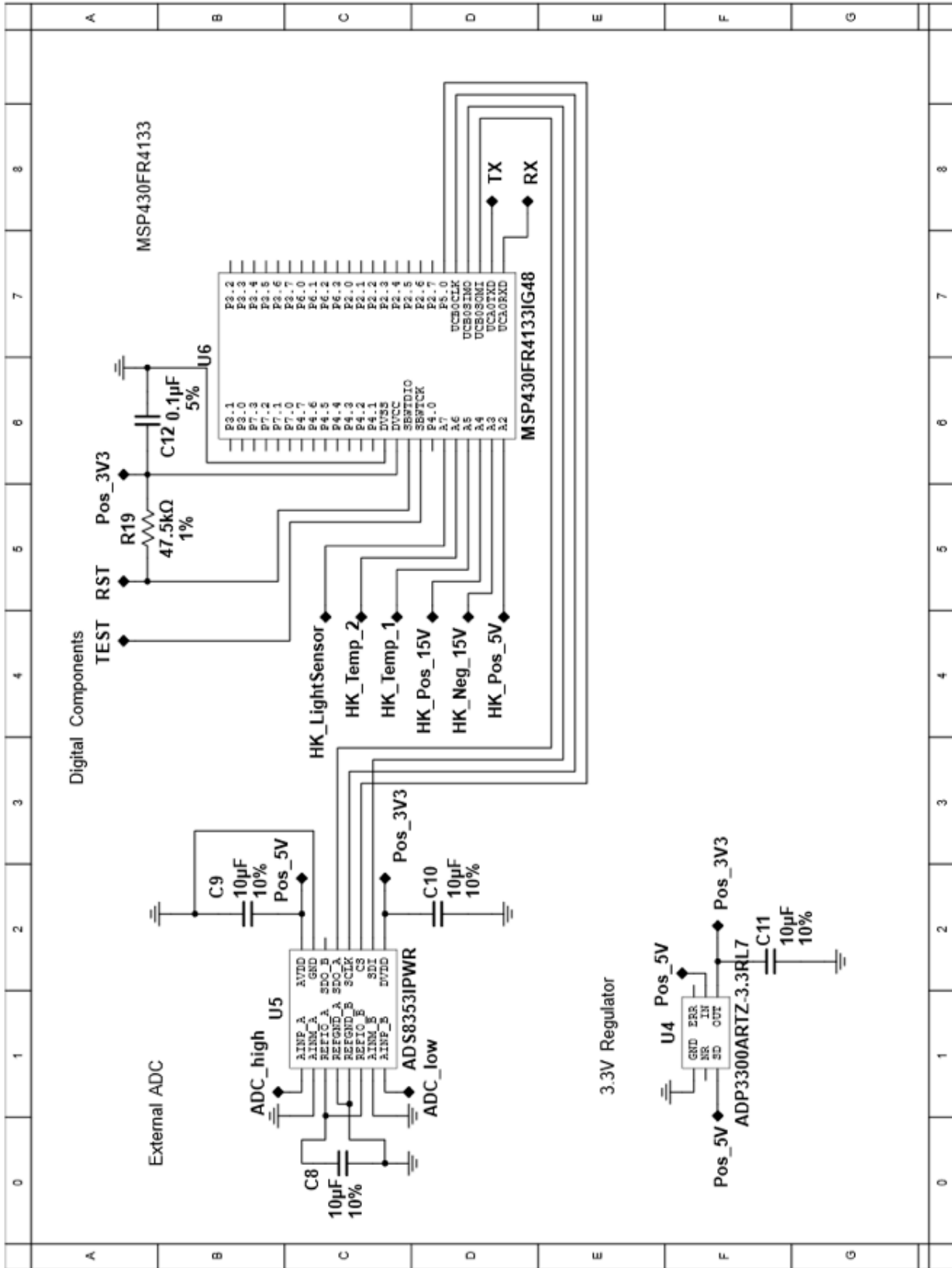
Texas Instruments. *MSP430FR413x mixed-signal microcontrollers*, slas865d edition, 10 2014c. Datasheet.

Appendix A

Circuit Schematic







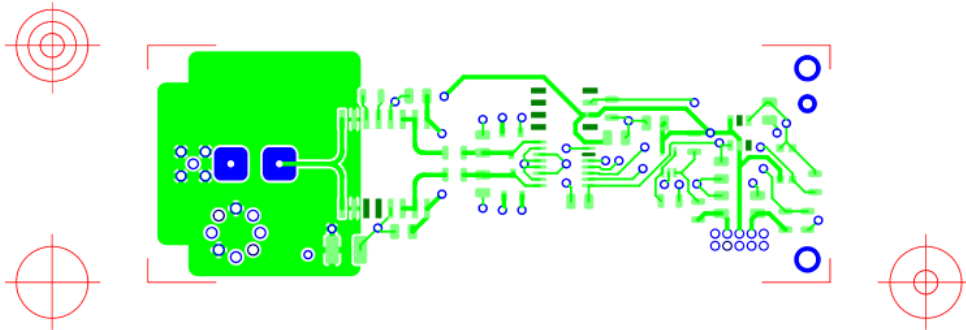
Appendix B

Circuit Layout

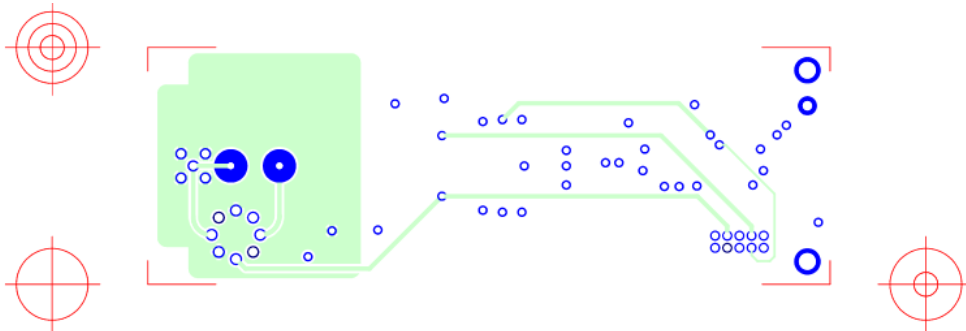
Layout:



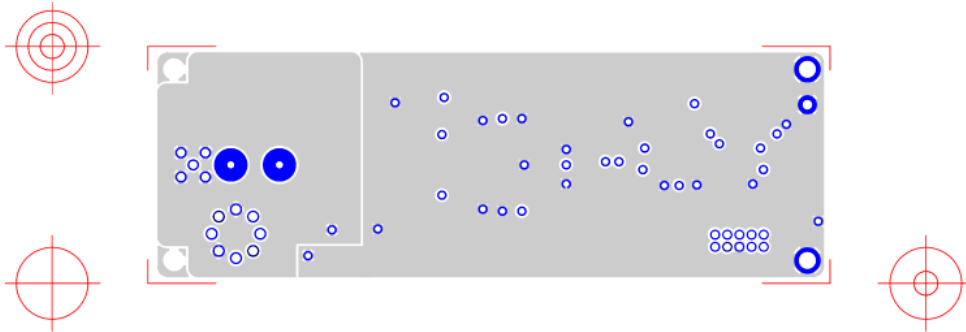
Board outline



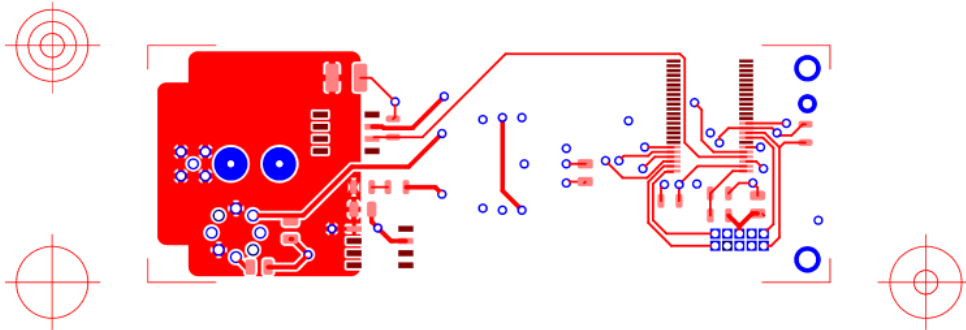
Copper top



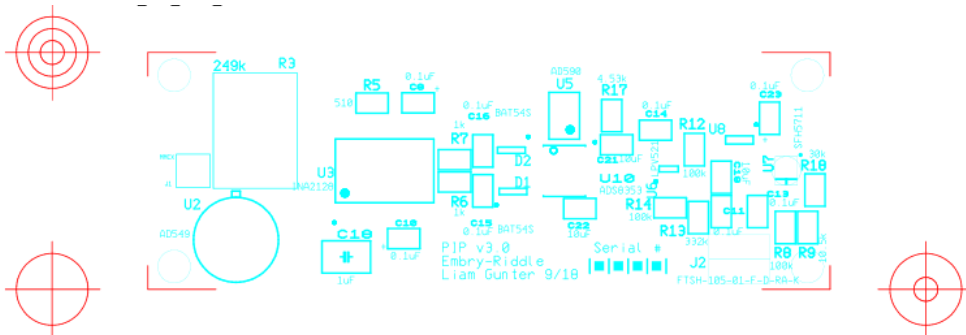
Copper inner 1



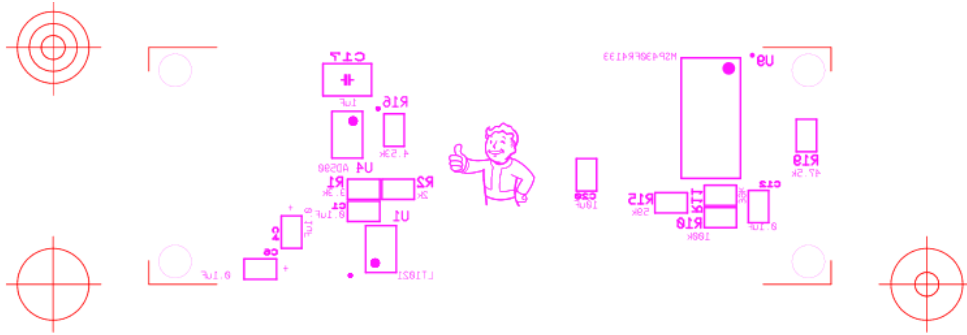
Copper inner 2 (ground)



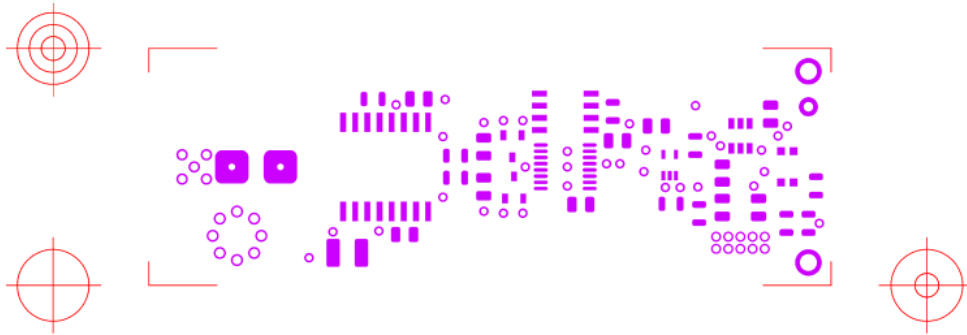
Copper bottom



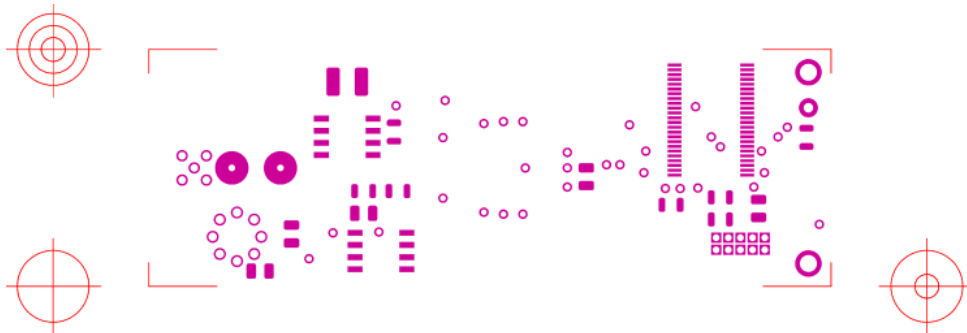
Silkscreen top



Silkscreen bottom



Soldermask top



Soldermask bottom

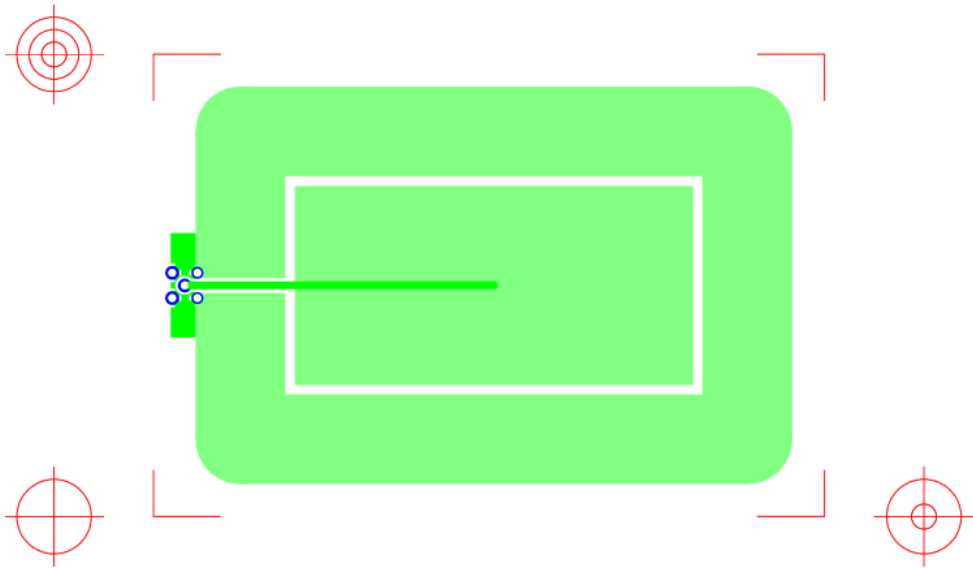
Appendix C

Sensor Layout

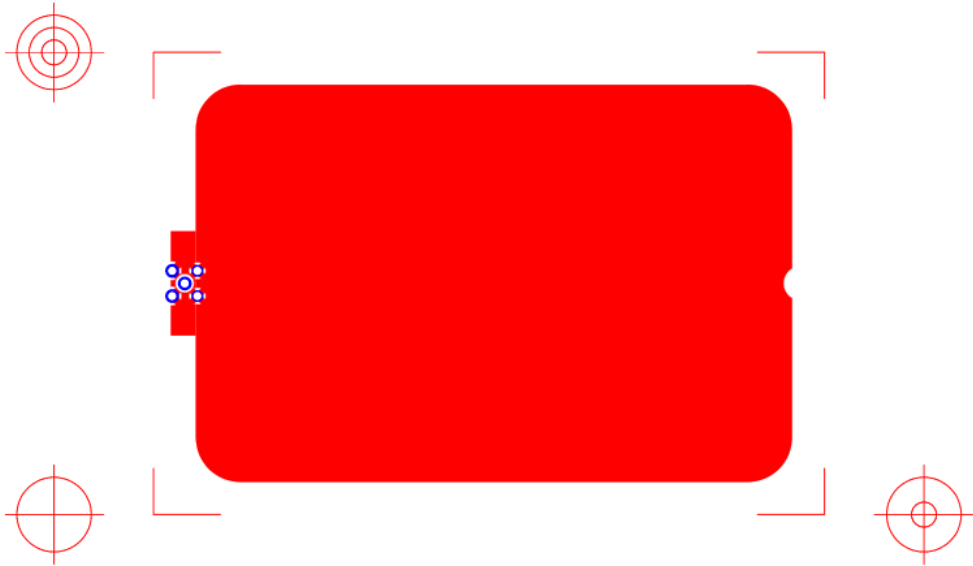
Layout:



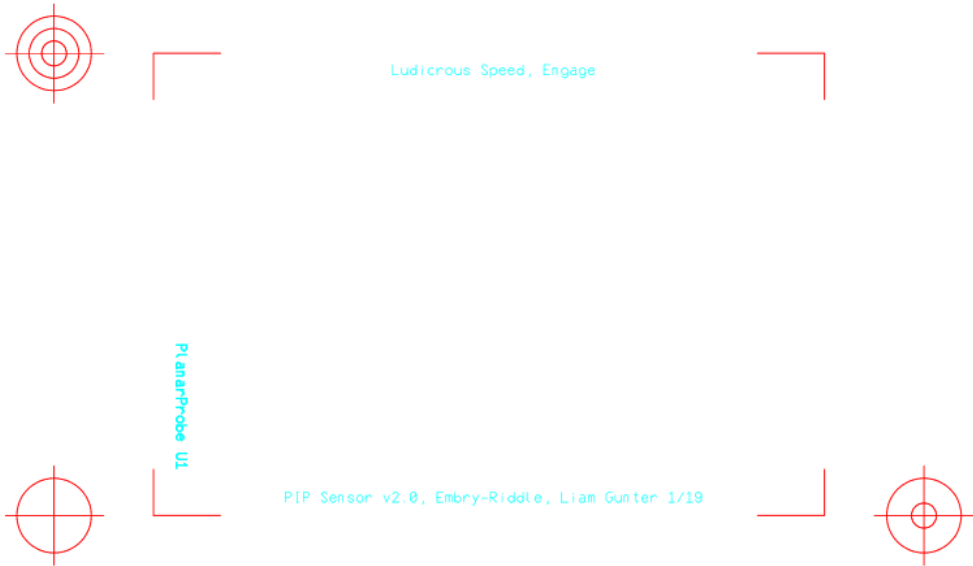
Board outline



Copper top



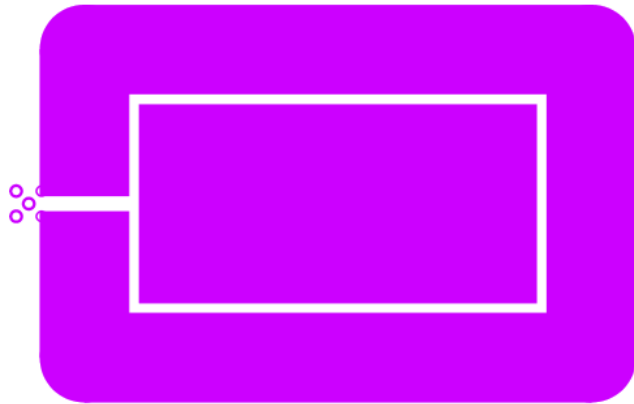
Copper bottom



Silkscreen top



Silkscreen bottom



Soldermask top



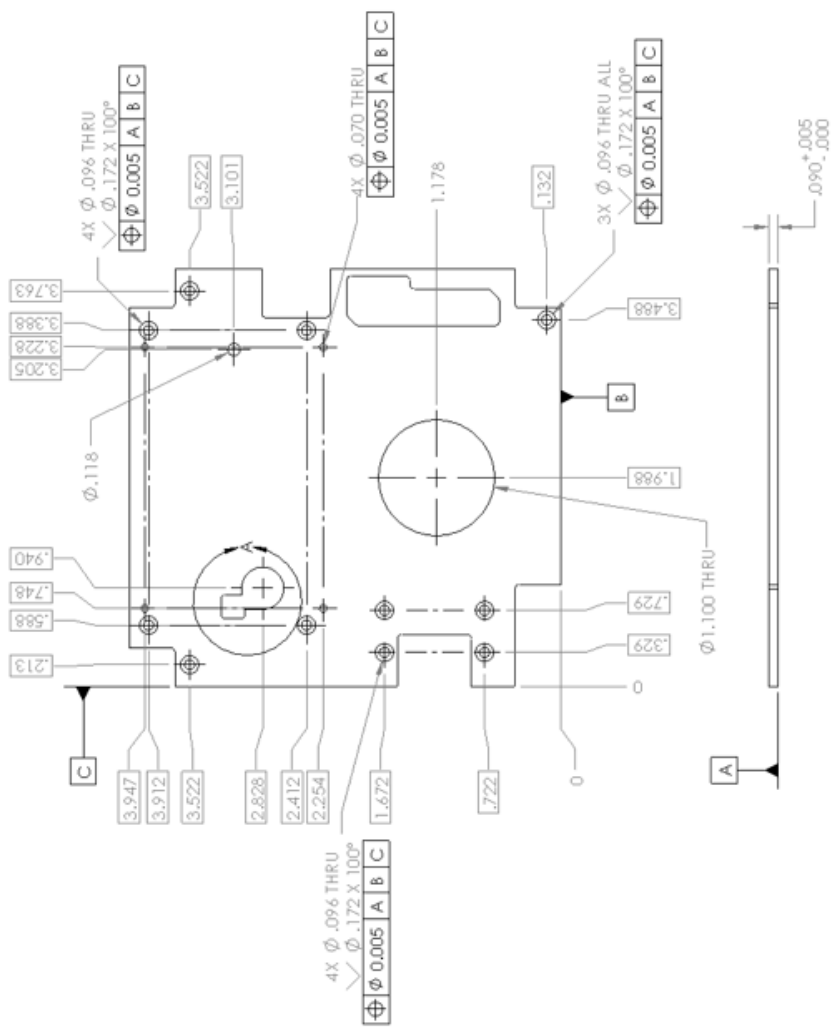
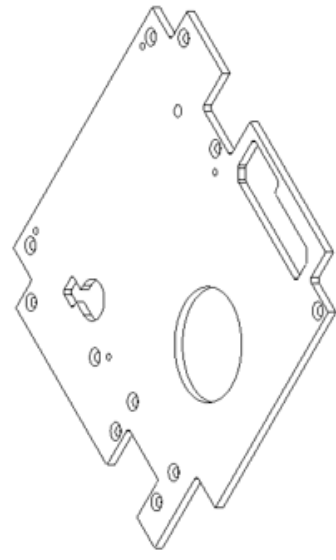
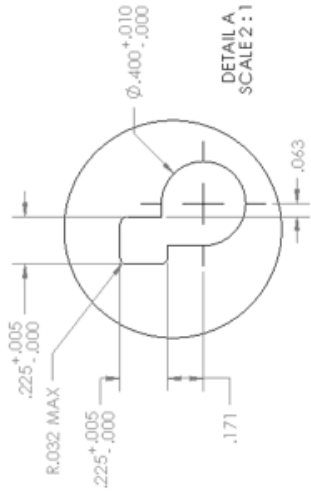
Soldermask bottom

Appendix D

Mechanical Design

APPENDIX D. MECHANICAL DESIGN

REVISIONS		DATE
REV	DESCRIPTION	DATE
A	INITIAL RELEASE	9/10/2018



	UNLESS OTHERWISE SPECIFIED:	DIMENSIONS ARE IN INCHES DIMENSIONS FOR HOLE LOCATIONS ARE TO BE TAKEN FROM THE CENTER OF THE HOLE UNLESS OTHERWISE SPECIFIED DO NOT SCALE DRAWING GENERAL TOLERANCES: FRACTIONAL: .015 DECIMAL: .004 ANGLES: .15 HOLE LOCATIONS: .004 CONCENTRICITY: .004 SURFACE FINISH: .125
	COPYRIGHT © 2018 THE AEROSPACE CORPORATION PROPRIETARY AND CONFIDENTIAL INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF AEROSPACE CORPORATION. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF AEROSPACE CORPORATION IS PROHIBITED.	TITLE: END PLATE, +Z

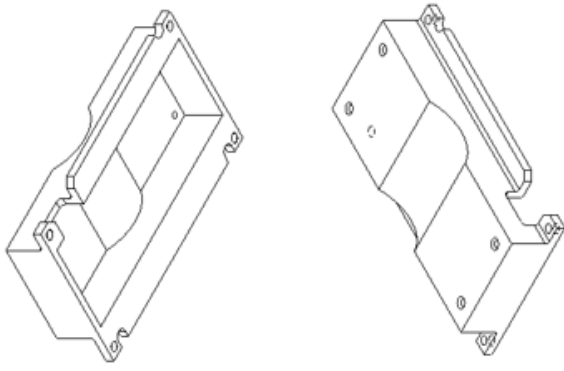
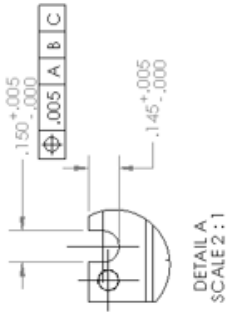
- NOTES:**
1. MATERIAL: ALUMINUM 6061 T6
 2. FINISH: BLACK ANODIZE PER MIL-A-8625 TYPE II CLASS 2
 3. USE CAD MODEL FOR DIMENSIONS NOT SHOWN IN THIS PRINT WITH ± 0.010 TOLERANCE
 4. USE CERTIFIED MATERIAL. ATTACH MATERIAL CERTIFICATION TO PRINT.
 5. INSPECT ALL GIVEN DIMENSIONS EXCLUDING ALL HOLE DIMENSIONS AND WRITE RESULT ON PRINT. DELIVER PRINT WITH PART.

REV	DWG. NO.	SCALE	WEIGHT
A	28845	1:1	51.231

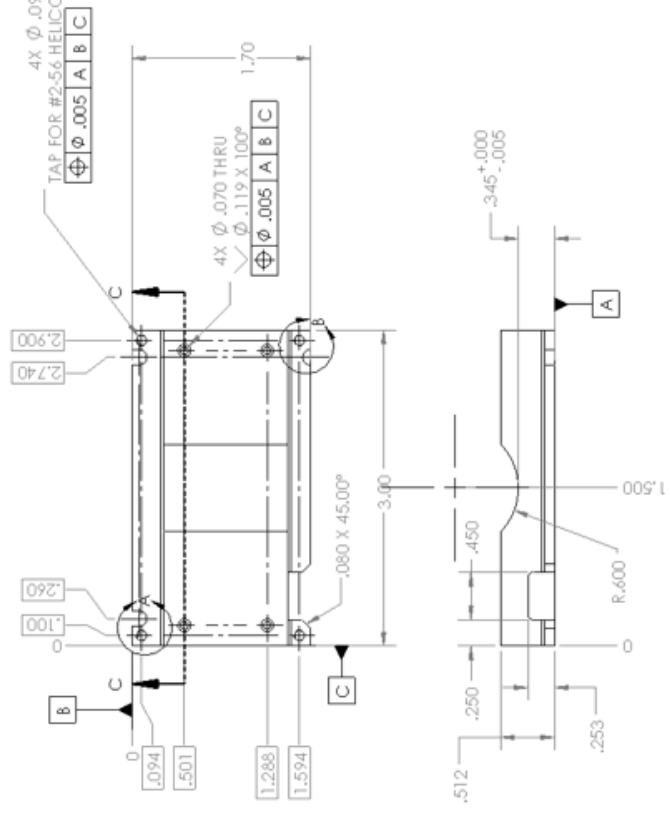
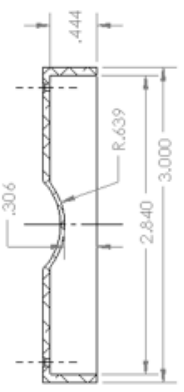
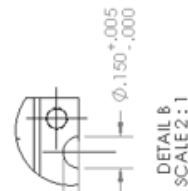
APPENDIX D. MECHANICAL DESIGN

REVISIONS	
REV	DESCRIPTION
A	INITIAL RELEASE

DATE
9/11/2018

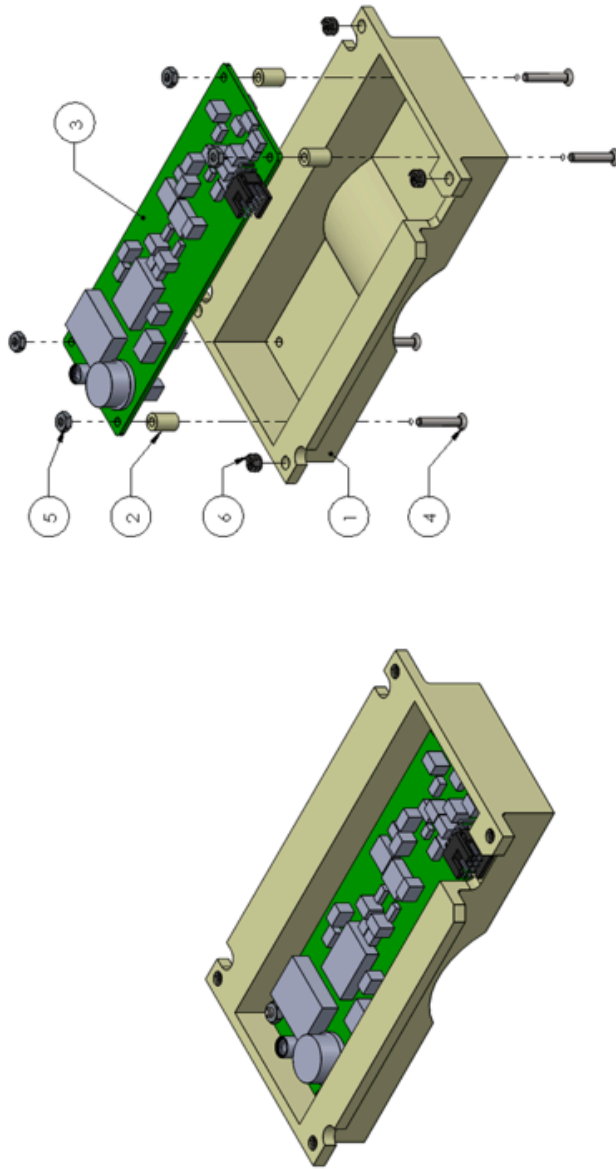


4X $\varnothing .094$ THRU
TAP FOR #2-56 HELICOIL INSERT = 1.0" DIA.

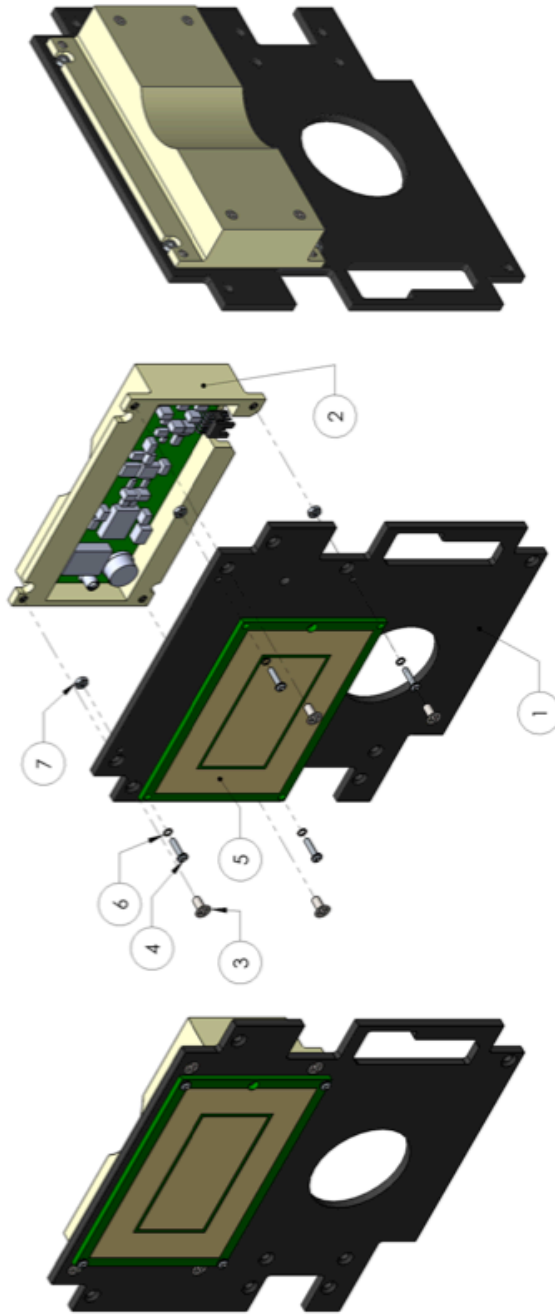


	UNLESS OTHERWISE SPECIFIED:	TITLE:
	DIMENSIONS ARE IN INCHES UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN MILLIMETERS UNLESS OTHERWISE SPECIFIED DO NOT SCALE DRAWING	EMI SHIELD, PIP
THE AEROSPACE CORPORATION PROPRIETARY AND CONFIDENTIAL DRAWING IS THE PROPERTY OF AEROSPACE CORPORATION REPRODUCTION IN WHOLE OR IN PART OR ANY MANNER WITHOUT THE PERMISSION OF AEROSPACE CORPORATION IS PROHIBITED.	GENERAL TOLERANCES: FRACTIONS: .001 DECIMALS: .005 ANGLES: .001 HOLE POSITION: .001 CONCENTRICITY: DATUM B .001	SIZE DWG. NO. B 31326
		SCALE: 1:1 WEIGHT: 26.812 SHEET 1 OF 1

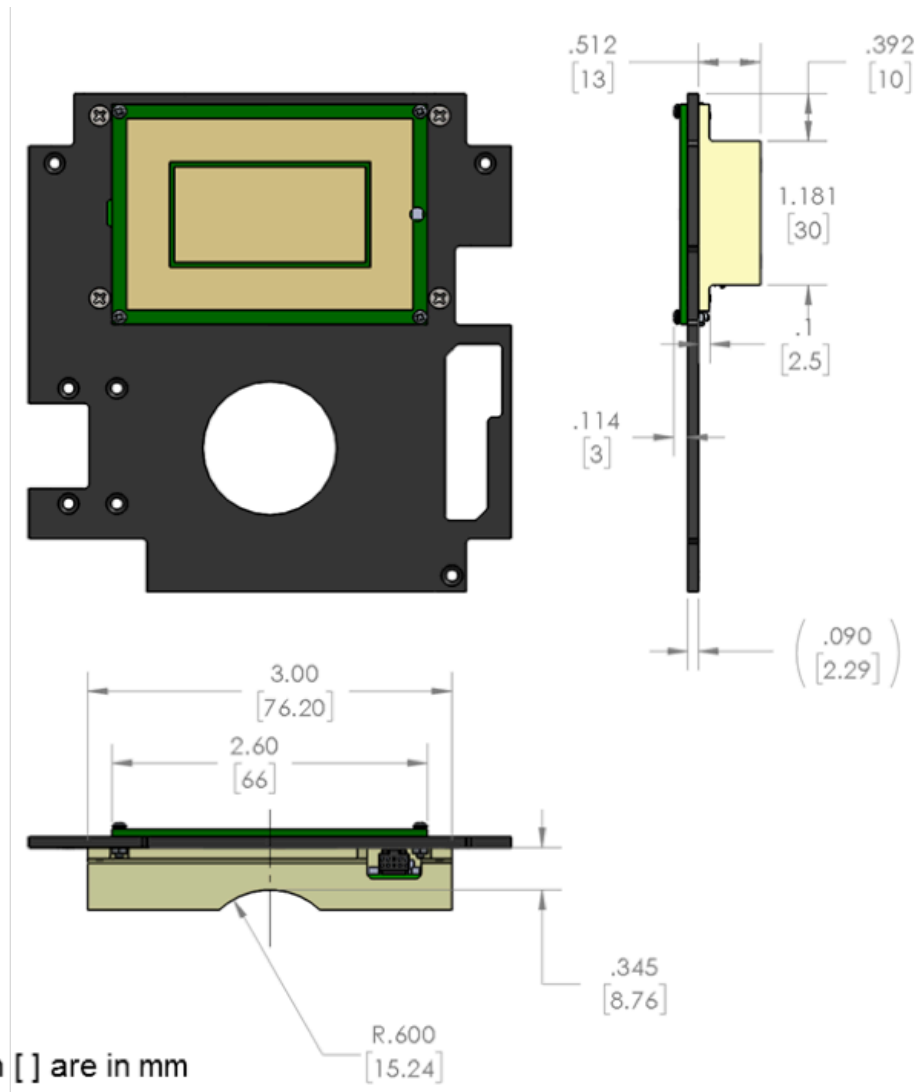
- NOTES:**
1. MATERIAL: ALUMINUM 6061-T6
 2. FINISH: CHEM. CONV. COAT PER MIL-DTL-5511F CLASS 1A (GOLD)
 3. USE CAD MODEL FOR DIMENSIONS NOT SHOWN IN THIS PRINT WITH $\pm .010$ TOLERANCE
 4. INSPECT ALL GIVEN DIMENSIONS INCLUDING ALL HOLE DIMENSIONS AND WRITE RESULT ON PRINT. DELIVER PRINT WITH PART.
 5. USE CERTIFIED MATERIAL AT TACH MATERIAL CERTIFICATION TO PRINT.



ITEM NO.	NUMBER	DESCRIPTION	QTY.
1	31326	PIP EMI SHIELD	1
2	31329	SPACER, #0 SCREW, 0.13"ODX0.1875"L, CHEM. CONV. COAT PER MIL-DTL-5541F CLASS 1A (GOLD)	1
3	31328	PIP PCBA	4
4	MS51960-5	SCREW, 0-80X.375L, PHILLIPS FLAT HEAD, 82° COUNTERSINK ANGLE	4
5	31504	NUT, 0-80, NARROW (SMALL PATTERN), 18-8 STAINLESS STEEL	1
6	MS21209-C0210	LOCKING HELICAL INSERT, 2-56 X 0.086"L	4



ITEM NO.	NUMBER	DESCRIPTION	QTY.
1	28845	END PLATE, +Z	1
2	31330	PIP PCB+EMI SHIELD ASSEMBLY	1
3	NAS 662C-2R3	SCREW, 2-56x.19L FLT HD	4
4	MSS1958-122	SCREW, PHILLIPS ROUNDED HEAD, 0-80 X 0.25" LONG, 18-8 STAINLESS STEEL	4
5	31327	LARGE SENSOR, PIP	1
6	NAS 620-C0	WASHER, #0, .100D	4
7	31504	NUT, 0-80, NARROW (SMALL PATTERN), 18-8 STAINLESS STEEL	4



Appendix E

Instrument Firmware

main.c

```

1 /*-----
2 * LLITED Mission Planar Ion Probe (PIP) instrument flight software
3 * Written by Liam Gunter and Austin Qale
4 * Embry-Riddle Aeronautical University
5 * Space and Atmospheric Instrumentation Laboratory
6 * 2018 - 2019
7 -----*/
8
9 // Specifying board ID# constant
10 #define BOARD_ID      0b10000000
11
12 // Specifying operation mode: 0 = flight mode, 1 = calibration mode
13 #define CALIBRATION   1
14
15 // CRC-16-CCITT-FALSE: x16 + x12 + x5 + 1
16 #define MSG_LENGTH    6
17 #define CRC16_POLY    0x1021
18 #define CRC16_FINAL_XOR 0x0000
19 #define CRC16_INIT    0xFFFF
20
21 // Specifying oversampling constant and right shift division factor
22 #define NOP            16
23 #define RS             4
24
25 // Specifying timer A compare register value: 15000 cycles at 16MHz = 0.9375ms for watchdog "kick the dog" timer
26 #define TIMER_COUNT   15000
27
28 // Specifying housekeeping maximum index
29 #define HKMAX         6
30
31 // Including requisite libraries
32 #include <msp430.h>
33 #include <string.h>
34 #include <stdio.h>
35 #include <driverlib.h>
36
37 // Declaring functions
38 void SYSTEMinit();
39 void writeUART(char* sentence);
40 int analog (int channel);
41 void GPIOinit();
42 unsigned short crc16MakeBitwise2(unsigned short, unsigned short, uint8_t*, unsigned int);
43
44 // Declaring global variables
45 volatile uint8_t CMD_FG = 0; // Command flag variable
46 volatile uint8_t CMD_STR = 0; // Command string variable
47 //volatile uint16_t TIMER_COUNT = 15000; // Enable for use instead of constant TIMER_COUNT for verifying watchdog functionality
48
49 // Main function
50 int main(void)
51 {
52     // Disabling watchdog timer, page 359 in MSP430FR4xx and MSP430FR2xx Family User's Guide
53     WDTCTL = WDTPW | WDTHOLD;
54
55     // Declaring function variables
56     int j; // Oversampling while loop counter variable (and calibration housekeeping counter)
57     uint16_t value[4]; // ADS8353 SPI data response storage variables
58     long average1; // High gain channel running total variable
59     long average2; // Low gain channel running total variable
60     uint8_t data[8]; // Data array for sending over UART (6 bytes science data + 2 bytes CRC) variables
61     uint16_t HK; // Housekeeping channel data value variable
62     uint8_t HKID = 0; // Housekeeping ID number variable
63     uint8_t mode = 0; // Mode ID for while loop variable
64     unsigned short crc16; // CRC calculated checksum variable
65     uint16_t HK_Calibration[HKMAX]; // Calibration housekeeping array (used for calibration)
66
67     // Perform system initialization function
68     SYSTEMinit();
69
70     // Enable interrupts (timer and UART)
71     __bis_SR_register(GIE);
72
73     // Infinite operational loop
74     while(1)
75     {
76         // If ISR has activated command flag

```

```

main.c

77  if (CMD_FG == 1)
78  {
79      // Command lookup dictionary
80      switch(CMD_STR)
81      {
82      case 0b01010101: // ASCII 'U': measurement, 16 oversample science data measurement request
83          mode = 1;
84          break;
85      case 0b00101010: // ASCII '*': error, resend packet with recalculated CRC request
86          mode = 2;
87          break;
88      case 0b00001111: // ASCII 'S' (shift in control character): shutdown, prepare for imminent power down request
89          mode = 3;
90          break;
91      default: // Any unrecognized command sequence: no action or response
92          mode = 0; // "I don't understand the question and I won't respond to it"
93          break;
94      }
95
96      // Mode 1 response (measurement)
97      if (mode == 1)
98      {
99          // Initializing important variables to 0
100         average1 = 0;
101         average2 = 0;
102         data[0] = 0;
103         data[1] = 0;
104         data[2] = 0;
105         data[3] = 0;
106         data[4] = 0;
107         data[5] = 0;
108         data[6] = 0;
109         data[7] = 0;
110
111         //**** Configuring external ADC *****/
112         // Configuring write to ADS8353 SPI settings, page 600 in MSP430FR4xx and MSP430FR2xx Family User's Guide
113         UCBOCTLW0 |= UCSWRST; // Allows below bits to be edited
114         UCBOCTLW0 &=~ UCCKPH; // Clearing SPI clock phase so rising edge separates bits (data is changed
115                               // by MSP430 on rising edge and latched by ADS8353 on falling edge)
116         UCBOCTLW0 &=~ UCCKPL; // Clearing SPI clock polarity so inactive low
117         UCBOCTLW0 &=~ UCSWRST; // Locks bits again
118
119         // Configuring CFR (configuration register) 0b 1000 0110 0100 0000, page 36 in ADS8353 datasheet
120         P5OUT &=~ BIT0; // Activates channel select
121         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
122         UCB0TXBUF = 0x86; // Accessing CFR and setting 32-CLK mode, use only one serial line,
123                               // and full-scale range 2 x Vref
124         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
125         UCB0TXBUF = 0x40; // Setting lower input to ground, internal reference, no standby,
126                               // and straight binary data format (as opposed to two's complement)
127         // Send four bytes of zeros to complete frame
128         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
129         UCB0TXBUF = 0x00; // Transmits 8 zeros
130         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
131         UCB0TXBUF = 0x00; // Transmits 8 zeros
132         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
133         UCB0TXBUF = 0x00; // Transmits 8 zeros
134         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
135         UCB0TXBUF = 0x00; // Transmits 8 zeros
136         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
137         P5OUT |= BIT0; // Deactivates channel select
138
139         _delay_cycles(24); // Wait time (1.5us) required, determined from trial and error during v2.0
140
141         testing
142
143         // Configuring REFDAC A (internal reference) 0b 1001 1111 1111 1000, page 37 in ADS8353 datasheet
144         P5OUT &=~ BIT0; // Activates channel select
145         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
146         UCB0TXBUF = 0x9F; // Accessing REFDAC A and setting to 2.5V
147         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
148         UCB0TXBUF = 0xF8; // Setting to 2.5V
149         // Send four bytes of zeros to complete frame
150         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
151         UCB0TXBUF = 0x00; // Transmits 8 zeros
152         while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
153         UCB0TXBUF = 0x00; // Transmits 8 zeros

```



```

main.c

152 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
153 UCB0TXBUF = 0x00; // Transmits 8 zeros
154 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
155 UCB0TXBUF = 0x00;
156 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
157 P5OUT |= BIT0; // Deactivates channel select
158
159 // Configuring REFDAC B (internal reference) 0b 1010 1111 1111 1000, page 37 in ADS8353 datasheet
160 P5OUT &=~ BIT0; // Activates channel select
161 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
162 UCB0TXBUF = 0xAF; // Accessing REFDAC B and setting to 2.5V
163 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
164 UCB0TXBUF = 0xF8; // Setting to 2.5V
165 // Send four bytes of zeros to complete frame
166 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
167 UCB0TXBUF = 0x00; // Transmits 8 zeros
168 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
169 UCB0TXBUF = 0x00; // Transmits 8 zeros
170 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
171 UCB0TXBUF = 0x00; // Transmits 8 zeros
172 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
173 UCB0TXBUF = 0x00; // Transmits 8 zeros
174 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
175 P5OUT |= BIT0; // Deactivates channel select
176 /**** Done configuring external ADC *****/
177
178 // Configuring read from ADS8353 SPI settings, page 600 in MSP430FR4xx and MSP430FR2xx Family User's Guide
179 UCB0CTLW0 |= UCSWRST; // Allows below bits to be edited
180 UCB0CTLW0 |= UCCKPH; // Setting SPI clock phase so falling edge separates bits (data is changed on
181 // falling edge by ADS8353 and captured by MSP430 on rising edge)
182 UCB0CTLW0 &=~ UCCKPL; // Clearing SPI clock polarity so inactive low
183 UCB0CTLW0 &=~ UCSWRST; // Locks bits again
184
185 delay_cycles(12000); // ADC warmup time (750us), determined from trial and error during testing v2.1
186 // with Dr. Barjatya
187
188 // Oversampling loop
189 for (i = 0; i < NOP; i++)
190 {
191 // Takes single external ADC measurement, pages 35-43 in ADS8353 datasheet
192 P5OUT &=~ BIT0; // Activates channel select line
193 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
194 UCB0TXBUF = 0x00; // Transmits 8 zeros
195 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
196 UCB0TXBUF = 0x00; // Transmits 8 zeros
197 (void)UCB0RXBUF; // Just in case, clears rx buffer
198 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
199 UCB0TXBUF = 0x00; // Transmits 8 zeros
200 while(UCB0STATW & UCBUSY); // SPI_B0_Buffer_Ready?
201 value[0] = UCB0RXBUF; // Stores 8 bits of ADC data (high gain MSBs)
202 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
203 UCB0TXBUF = 0x00; // Transmits 8 zeros
204 while(UCB0STATW & UCBUSY); // SPI_B0_Buffer_Ready?
205 value[1] = UCB0RXBUF; // Stores 8 bits of ADC data (high gain LSBs)
206 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
207 UCB0TXBUF = 0x00; // Transmits 8 zeros
208 while(UCB0STATW & UCBUSY); // SPI_B0_Buffer_Ready?
209 value[2] = UCB0RXBUF; // Stores 8 bits of ADC data (low gain MSBs)
210 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
211 UCB0TXBUF = 0x00; // Transmits 8 zeros
212 while(UCB0STATW & UCBUSY); // SPI_B0_Buffer_Ready?
213 value[3] = UCB0RXBUF; // Stores 8 bits of ADC data (low gain LSBs)
214 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
215 P5OUT |= BIT0; // Deactivates channel select line
216
217 // If on any sample except last, delay 4800 cycles to let RC filter settle
218 if (j != (NOP-1))
219 {
220 delay_cycles(4800); // RC filter settling time (300us), determined from trial and error during
221 // testing v2.1 with Dr. Barjatya
222 }
223
224 // Add running totals from each channel
225 average1 += (value[0] << 8) + value[1];
226 average2 += (value[2] << 8) + value[3];
227 }

```

```

main.c

228
229 //***** Configuring external ADC *****/
230 // Configuring write to ADS8353 SPI settings, page 600 in MSP430FR4xx and MSP430FR2xx Family User's Guide
231 UCB0CTLW0 |= UCSWRST; // Allows below bits to be edited
232 UCB0CTLW0 &=~ UCCKPH; // Clearing SPI clock phase so rising edge separates bits (data is changed
233 // by MSP430 on rising edge and latched by ADS8353 on falling edge)
234 UCB0CTLW0 &=~ UCCKPL; // Clearing SPI clock polarity so inactive low
235 UCB0CTLW0 &=~ UCSWRST; // Locks bits again
236
237 // Putting external ADC into standby mode 0b 1000 0110 0110 0000, page 36/45-46 in ADS8353 datasheet
238 P5OUT &=~ BIT0; // Activates channel select
239 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
240 UCB0TXBUF = 0x86; // Accessing CFR and setting 32-CLK mode, use only one serial line,
241 // and full-scale range 2 x Vref
242 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
243 UCB0TXBUF = 0x60; // Setting lower input to ground, internal reference, standby,
244 // and straight binary data format (as opposed to two's complement)
245 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
246 UCB0TXBUF = 0x00; // Transmits 8 zeros
247 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
248 UCB0TXBUF = 0x00; // Transmits 8 zeros
249 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
250 UCB0TXBUF = 0x00; // Transmits 8 zeros
251 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
252 UCB0TXBUF = 0x00; // Transmits 8 zeros
253 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
254 P5OUT |= BIT0; // Deactivates channel select
255 //***** Done configuring external ADC *****/
256
257 // Right shift accumulated running totals into an average for each channel
258 average1 = average1 >> RS;
259 average2 = average2 >> RS;
260
261 // Assign data bytes by direct assignment of LSBs and right shifting to get MSBs
262 data[2] = average1 >> 8;
263 data[3] = average1;
264 data[4] = average2 >> 8;
265 data[5] = average2;
266
267 // Loops housekeeping ID through 6 options: +5V, -15V, +15V, Temp1, Temp2, light sensor
268 HKID = HKID % HKMAX;
269
270 // Polling current housekeeping ID cycle channel
271 HK = analog(HKID + 2);
272
273 // Constructing data byte 1 (board ID, housekeeping ID, housekeeping MSB)
274 data[0] = BOARD_ID + (HKID << 2) + (HK >> 8);
275
276 // Constructing data byte 2 (housekeeping LSB)
277 data[1] = HK;
278
279 // Incrementing housekeeping ID cycle
280 HKID++;
281 }
282
283 // Continuing mode 1 (measurement) and beginning mode 2 (error)
284 if ((mode == 1) || (mode == 2))
285 {
286 // Calculating CRC-16-CCITT-FALSE given 6 science data bytes
287 crc16 = crc16MakeBitwise2(CRC16_INIT, CRC16_POLY, data, MSG_LENGTH);
288 data[6] = crc16 >> 8;
289 data[7] = crc16;
290
291 // Transmitting data + CRC over UART, page 573/580-581/586-587/591 in MSP430FR4xx and MSP430FR2xx Family User's Guide
292 while (!(UCA0IFG & UCTXIFG)); // Polls transmit flag
293 UCA0TXBUF = data[0]; // Transmits data byte 1
294 while (!(UCA0IFG & UCTXIFG)); // Polls transmit flag
295 UCA0TXBUF = data[1]; // Transmits data byte 2
296 while (!(UCA0IFG & UCTXIFG)); // Polls transmit flag
297 UCA0TXBUF = data[2]; // Transmits data byte 3
298 while (!(UCA0IFG & UCTXIFG)); // Polls transmit flag
299 UCA0TXBUF = data[3]; // Transmits data byte 4
300 while (!(UCA0IFG & UCTXIFG)); // Polls transmit flag
301 UCA0TXBUF = data[4]; // Transmits data byte 5
302 while (!(UCA0IFG & UCTXIFG)); // Polls transmit flag
303 UCA0TXBUF = data[5]; // Transmits data byte 6

```

```

main.c

304 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
305 UCA0TXBUF = data[6]; // Transmits CRC byte 1
306 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
307 UCA0TXBUF = data[7]; // Transmits CRC byte 2
308
309 /**** Enabled during calibration for full housekeeping output *****/
310 if (CALIBRATION == 1)
311 {
312 // Reading each housekeeping channel... make sure to declare HK_Calibration[HKMAX] at beginning of main
313 for(j = 0; j < HKMAX; j++)
314 {
315 HK_Calibration[j] = analog(j + 2);
316 }
317
318 // Transmitting full housekeeping data in format:
319 // +5 MSB | +5 LSB | -15 MSB | -15 LSB | +15 MSB | +15 LSB | T1 MSB | T1 LSB | T2 MSB | T2 LSB | LS MSB | LS LSB
320 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
321 UCA0TXBUF = HK_Calibration[0]>>8; // Transmits housekeeping
322 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
323 UCA0TXBUF = HK_Calibration[0]; // Transmits housekeeping
324 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
325 UCA0TXBUF = HK_Calibration[1]>>8; // Transmits housekeeping
326 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
327 UCA0TXBUF = HK_Calibration[1]; // Transmits housekeeping
328 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
329 UCA0TXBUF = HK_Calibration[2]>>8; // Transmits housekeeping
330 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
331 UCA0TXBUF = HK_Calibration[2]; // Transmits housekeeping
332 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
333 UCA0TXBUF = HK_Calibration[3]>>8; // Transmits housekeeping
334 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
335 UCA0TXBUF = HK_Calibration[3]; // Transmits housekeeping
336 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
337 UCA0TXBUF = HK_Calibration[4]>>8; // Transmits housekeeping
338 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
339 UCA0TXBUF = HK_Calibration[4]; // Transmits housekeeping
340 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
341 UCA0TXBUF = HK_Calibration[5]>>8; // Transmits housekeeping
342 while(!(UCA0IFG&UCTXIFG)); // Polls transmit flag
343 UCA0TXBUF = HK_Calibration[5]; // Transmits housekeeping
344 }
345 /**** End of calibration full housekeeping output *****/
346 }
347
348 // Mode 3 response (shutdown)
349 if (mode == 3)
350 {
351 // Constructing data bytes to acknowledge shutdown imminent (HKID = 0b111 and HK = high gain = low gain = 0)
352 data[0] = BOARD_ID + 0b00011100;
353 data[1] = 0x00;
354 data[2] = 0x00;
355 data[3] = 0x00;
356 data[4] = 0x00;
357 data[5] = 0x00;
358
359 // Calculating CRC-16-CCITT-FALSE given 6 science data bytes
360 crc16 = crc16MakeBitwise2(CRC16_INIT, CRC16_POLY, data, MSG_LENGTH);
361 data[6] = crc16 >> 8;
362 data[7] = crc16;
363
364 // Ensuring external ADC is in standby mode
365 /**** Configuring external ADC *****/
366 // Configuring write to ADS8353 SPI settings, page 600 in MSP430FR4xx and MSP430FR2xx Family User's Guide
367 UCBOCTLW0 |= UCSWRST; // Allows below bits to be edited
368 UCBOCTLW0 &=~ UCCKPH; // Clearing SPI clock phase so rising edge separates bits (data is changed
369 // by MSP430 on rising edge and latched by ADS8353 on falling edge)
370 UCBOCTLW0 &=~ UCCKPL; // Clearing SPI clock polarity so inactive low
371 UCBOCTLW0 &=~ UCSWRST; // Locks bits again
372
373 // Putting external ADC into standby mode 0b 1000 0110 0110 0000, page 36/45-46 in ADS8353 datasheet
374 P5OUT &=~ BIT0; // Activates channel select
375 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
376 UCB0TXBUF = 0x86; // Accessing CFR and setting 32-CLK mode, use only one serial line,
377 // and full-scale range 2 x Vref
378 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
379 UCB0TXBUF = 0x60; // Setting lower input to ground, internal reference, standby,

```

```

main.c

380                                     // and straight binary data format (as opposed to two's complement)
381 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
382 UCB0TXBUF = 0x00; // Transmits 8 zeros
383 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
384 UCB0TXBUF = 0x00; // Transmits 8 zeros
385 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
386 UCB0TXBUF = 0x00; // Transmits 8 zeros
387 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
388 UCB0TXBUF = 0x00; // Transmits 8 zeros
389 while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
390 P5OUT |= BIT0; // Deactivates channel select
391 /**** Done configuring external ADC *****/
392
393 // Disabling watchdog timer, page 359 in MSP430FR4xx and MSP430FR2xx Family User's Guide
394 WDTCTL = WDTPW | WDTHOLD;
395
396 // Disable interrupts (timer and UART)
397 __bic_SR_register(GIE);
398
399 // Transmitting data + CRC over UART, page 573/580-581/586-587/591 in MSP430FR4xx and MSP430FR2xx Family User's Guide
400 while (!(UCA0IFG&UCTXIFG)); // Polls transmit flag
401 UCA0TXBUF = data[0]; // Transmits data byte 1
402 while (!(UCA0IFG&UCTXIFG)); // Polls transmit flag
403 UCA0TXBUF = data[1]; // Transmits data byte 2
404 while (!(UCA0IFG&UCTXIFG)); // Polls transmit flag
405 UCA0TXBUF = data[2]; // Transmits data byte 3
406 while (!(UCA0IFG&UCTXIFG)); // Polls transmit flag
407 UCA0TXBUF = data[3]; // Transmits data byte 4
408 while (!(UCA0IFG&UCTXIFG)); // Polls transmit flag
409 UCA0TXBUF = data[4]; // Transmits data byte 5
410 while (!(UCA0IFG&UCTXIFG)); // Polls transmit flag
411 UCA0TXBUF = data[5]; // Transmits data byte 6
412 while (!(UCA0IFG&UCTXIFG)); // Polls transmit flag
413 UCA0TXBUF = data[6]; // Transmits CRC byte 1
414 while (!(UCA0IFG&UCTXIFG)); // Polls transmit flag
415 UCA0TXBUF = data[7]; // Transmits CRC byte 2
416
417 // Enter low power mode 4: CPU/MCLK, FLL, ACLK off, page 38 in MSP430FR4xx and MSP430FR2xx Family User's Guide
418 __bis_SR_register(LPM4_bits);
419 }
420
421 // Reset command flag, command string, and mode identifier
422 CMD_FG = 0;
423 CMD_STR = 0;
424 mode = 0;
425 }
426 }
427 }
428
429 // Timer A0 interrupt service routine (kicking the dog!)
430 #if defined(__TI_COMPILER_VERSION__) || defined(__IAR_SYSTEMS_ICC__)
431 #pragma vector = TIMER0_A0_VECTOR
432 __interrupt void Timer_A (void)
433 #elif defined(__GNUC__)
434 void __attribute__((interrupt(TIMER0_A0_VECTOR))) Timer_A (void)
435 #else
436 #error Compiler not supported!
437 #endif
438 {
439     WDTCTL = WDTPW | WDTSSSEL_0 | WDTCNTCL | WDTIS_4; // Clears watchdog timer and sets back to previous settings, page 359 in
440 // MSP430FR4xx and MSP430FR2xx Family User's Guide
441     TA0CCR0 += TIMER_COUNT; // Setting TA0CCR0 to 15000/16MHz = 0.9375ms (less than 1/2 watchdog period of
442 // 2.05ms), page 388 in MSP430FR4xx and MSP430FR2xx Family User's Guide
443     //TIMER_COUNT += 100; // TIMER_COUNT global variable increment, used to simulate latchup condition
444     that // prevents timer ISR from servicing watchdog (enable for watchdog verification)
445 }
446
447 // UART interrupt service routine
448 #if defined(__TI_COMPILER_VERSION__) || defined(__IAR_SYSTEMS_ICC__)
449 #pragma vector=USCI_A0_VECTOR
450 __interrupt void USCI_A0_ISR(void)
451 #elif defined(__GNUC__)
452 void __attribute__((interrupt(USCI_A0_VECTOR))) USCI_A0_ISR (void)
453 #else
454 #error Compiler not supported!

```

main.c

```

455 #endif
456 {
457     // All UART interrupt vectors from page 597/601-602 in MSP430FR4xx and MSP430FR2xx Family User's Guide
458     switch(__even_in_range(UCA0IV,USCI_UART_UCTXCFIFG))
459     {
460     case USCI_NONE: break;
461     case USCI_UART_UCRXIFG: // If RX buffer full flag set
462         UCA0IFG &=~ UCRXIFG; // Clear interrupt
463         CMD_STR = UCA0RXBUF; // Records command word
464         CMD_FG = 1; // Activates measurement command flag
465         break;
466     case USCI_UART_UCTXIFG: break; // If TX buffer empty flag set, do nothing
467     case USCI_UART_UCSTTIFG: break; // If start bit received flag set, do nothing
468     case USCI_UART_UCTXCFIFG: break; // If transmit complete flag set, do nothing
469     default: break; // Else, do nothing
470     }
471 }
472
473 // System initialization function
474 void SYSTEMinit()
475 {
476     // GPIO initialization to all outputs, pullup/pulldown resistors enabled, initialized to low output
477     GPIOinit();
478
479     // If enabled, sets internal pullup resistor for reset pin. External resistor makes this unnecessary
480     // SFRPCR |= SYSRSTUP + SYSRSTRE;
481
482     // Configure GPIO specific to PIP application, page 9 in MSP430FR4133 datasheet and page 306-307/327-328 in MSP430FR4xx and
483     // MSP430FR2xx Family User's Guide
484     P5DIR = 0x00; // Setting all port 5 DIR to input, required for USCIB function
485     P5DIR |= BIT0; // Setting SPI CS pin (pin 5.0) as output
486     P5OUT |= BIT0; // Setting SPI CS pin high (idling SPI state, disable when documenting timing)
487     P5SEL0 = 0x0E; // Setting pins 5.1, 5.2, 5.3 to primary non I/O function (SPI)
488     P1SEL0 |= BIT0 + BIT1; // Setting UART pins (1.0/1.1) to primary non I/O function (UART)
489
490     // Clock configuration, page 38/113-118/126-130/134/295/310 in MSP430FR4xx and MSP430FR2xx Family User's Guide
491     FRCTL0 = FRCTLPW | NWAITS_1; // Configures wait state - required for speeds above 8 MHz
492     __bis_SR_register(SCG0); // Disable FLL
493     PMSCTL0 &=~ LOCKLPM5; // Unlock I/O pin and other clock configurations
494     CSCTL3 |= SELREF__REFOCLK; // Setting REF0 as FLL reference source and FLL reference divider = 1
495     CSCTL0 = 0; // Clearing DCO and MOD registers
496     CSCTL1 &=~ (DCORSEL_7); // Clearing DCO frequency select bits initially
497     CSCTL1 |= DCORSEL_5; // Setting DCO = 16MHz
498     CSCTL2 = FLLD_0 + 487; // Setting FLL loop divider = 1 and FLL multiplier = 488 (16MHz / 32.768kHz = 488.3)
499     __delay_cycles(3); // Delay (0.1875us), recommended from example code
500     __bic_SR_register(SCG0); // Re-enable FLL
501     while(CSCTL7 & (FLLUNLOCK0 | FLLUNLOCK1)); // FLL locked
502     CSCTL4 |= SELMS__DCOCLKDIV; // Setting default DCOCLKDIV SMCLK source
503
504     // 115.2kBaud from 16MHz clock UART configuration,
505     // page 577/583/585/588-591/593-595/600 in MSP430FR4xx and MSP430FR2xx Family User's Guide
506     UCA0CTLW0 |= UCSWRST; // Setting software reset enable bit allows UART to be configured
507     UCA0CTLW0 |= UCSSEL__SMCLK; // Setting no parity, LSB first, 8-bit data, one stop bit, UART mode, asynchronous
mode,
508     // SMCLK (16MHz) as UART clock source, reject erroneous characters from setting flag,
509     // no break characters, no dormant mode, no address mode, no address mode, no break mode
510     UCA0BR0 = 8; // Setting UCBR0 to 8 from table 22-5, page 589-590 in MSP430FR4xx and MSP430FR2xx
511     // Family User's Guide
512     UCA0MCTLW = 0xF700 | UCBRF_10 | UCOS16; // Setting UCBSR0 to 0xF7, UCBRF0 to 10, and enabling UCOS16 oversample mode from
513     // table 22-5, page 589-590 in MSP430FR4xx and MSP430FR2xx Family User's Guide
514     UCA0CTLW0 &=~ UCSWRST; // Initialize eUSCI
515     UCA0IE |= UCRXIE; // Enable USCI_A0 RX interrupt
516
517     // Configuring Internal ADC, page 107/109/534/536-538/546/551-555/557/561 in MSP430FR4xx and MSP430FR2xx Family User's Guide
518     ADCCTL0 |= ADCSHT_2 | ADCON; // Setting ADC S&H time to 16 ADCCLK cycles, single conversion, ADC on,
519     // conversion disabled, and start conversion disabled
520     ADCCTL1 |= ADCSHP | ADCSSEL_0 | ADCCONSEQ_0; // Setting ADCSC trigger, pulse sample mode, no input inversion, no clock divider,
521     // MODCLK (~5MHz) ADC clock source, and single-channel single conversion mode
522     ADCMCTL0 |= ADCSREF_1; // Setting + reference voltage to internal 1.5V VREF and - reference voltage to
523     // AVSS (GND)
524     ADCCTL2 |= ADCRES_1; // Setting no divider, ADC resolution to 10-bit, unsigned data format,
525     // and higher sampling rate
526     PMMCTL0_H = PMMPW_H; // Unlock the PMM registers
527     PMMCTL2 |= INTREFEN; // Enable internal reference
528     __delay_cycles(400); // Delay (25us) for reference settling, recommended from example code
529

```

```

main.c
530 // Configuring SPI settings, page 596-600/610-611/613 in MSP430FR4xx and MSP430FR2xx Family User's Guide
531 UCB0CTLW0 |= UCSWRST; // Setting software reset enable bit allows SPI to be configured
532 UCB0CTLW0 |= UCMSB | UCMST | // Setting MSB first and master mode
533 UCMODE_2 | UCSYNC | UCSSEL_2 | // Setting 4-pin SPI STE active low, synchronous mode, and SMCLK (16MHz) clock source
534 UCSTEM; // Setting STE pin used as 4-wire slave enable
535 UCB0CTLW0 &=~ UCCKPH; // Clearing SPI clock phase so rising edge separates bits (data is changed by MSP430
536 // on rising edge and latched by ADS8353 on falling edge)
537 UCB0CTLW0 &=~ UCCKPL; // Clearing SPI clock polarity so inactive low
538 UCB0BRW = 0x00; // Setting SPI clock divider = 0: f_bitclock = f_brclk
539 UCB0CTLW0 &=~ UCSWRST; // Clearing software reset enable bit locks SPI configuration
540
541 /***** Configuring external ADC *****/
542 // Configuring CFR (configuration register) 0b 1000 0110 0100 0000, page 36 in ADS8353 datasheet
543 P5OUT &=~ BIT0; // Activates channel select
544 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
545 UCB0TXBUF = 0x86; // Accessing CFR and setting 32-CLK mode, use only one serial line,
546 // and full-scale range 2 x Vref
547 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
548 UCB0TXBUF = 0x40; // Setting lower input to ground, internal reference, no standby,
549 // and straight binary data format (as opposed to two's complement)
550 // Send four bytes of zeros to complete frame
551 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
552 UCB0TXBUF = 0x00; // Transmits 8 zeros
553 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
554 UCB0TXBUF = 0x00; // Transmits 8 zeros
555 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
556 UCB0TXBUF = 0x00; // Transmits 8 zeros
557 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
558 UCB0TXBUF = 0x00; // Transmits 8 zeros
559 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
560 P5OUT |= BIT0; // Deactivates channel select
561
562 _delay_cycles(24); // Wait time (1.5us) required, determined from trial and error during v2.0 testing
563
564 // Configuring REFDAC A (internal reference) 0b 1001 1111 1111 1000, page 37 in ADS8353 datasheet
565 P5OUT &=~ BIT0; // Activates channel select
566 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
567 UCB0TXBUF = 0x9F; // Accessing REFDAC A and setting to 2.5V
568 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
569 UCB0TXBUF = 0xF8; // Setting to 2.5V
570 // Send four bytes of zeros to complete frame
571 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
572 UCB0TXBUF = 0x00; // Transmits 8 zeros
573 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
574 UCB0TXBUF = 0x00; // Transmits 8 zeros
575 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
576 UCB0TXBUF = 0x00; // Transmits 8 zeros
577 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
578 UCB0TXBUF = 0x00; // Transmits 8 zeros
579 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
580 P5OUT |= BIT0; // Deactivates channel select
581
582 // Configuring REFDAC B (internal reference) 0b 1010 1111 1111 1000, page 37 in ADS8353 datasheet
583 P5OUT &=~ BIT0; // Activates channel select
584 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
585 UCB0TXBUF = 0xAF; // Accessing REFDAC B and setting to 2.5V
586 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
587 UCB0TXBUF = 0xF8; // Setting to 2.5V
588 // Send four bytes of zeros to complete frame
589 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
590 UCB0TXBUF = 0x00; // Transmits 8 zeros
591 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
592 UCB0TXBUF = 0x00; // Transmits 8 zeros
593 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
594 UCB0TXBUF = 0x00; // Transmits 8 zeros
595 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
596 UCB0TXBUF = 0x00; // Transmits 8 zeros
597 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
598 P5OUT |= BIT0; // Deactivates channel select
599
600 // Configuring CFR (configuration register) 0b 1000 0110 0110 0000, page 36/45-46 in ADS8353 datasheet
601 P5OUT &=~ BIT0; // Activates channel select
602 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
603 UCB0TXBUF = 0x86; // Accessing CFR and setting 32-CLK mode, use only one serial line,
604 // and full-scale range 2 x Vref
605 while((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag

```



```

                                main.c
606  UCB0TXBUF = 0x00;                // Setting lower input to ground, internal reference, standby,
607                                     // and straight binary data format (as opposed to two's complement)
608  // Send four bytes of zeros to complete frame
609  while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
610  UCB0TXBUF = 0x00;                // Transmits 8 zeros
611  while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
612  UCB0TXBUF = 0x00;                // Transmits 8 zeros
613  while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
614  UCB0TXBUF = 0x00;                // Transmits 8 zeros
615  while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
616  UCB0TXBUF = 0x00;                // Transmits 8 zeros
617  while ((UCB0IFG & UCTXIFG) == 0); // Polls transmit flag
618  P5OUT |= BIT0;                   // Deactivates channel select
619  /**** Done configuring external ADC ****/
620
621  // Timer configuration, page 370/372-373/376/382/384-389 in MSP430FR4xx and MSP430FR2xx Family User's Guide
622  TA0CTL |= MC_0;                  // Setting stop mode (halting timer just in case reset did not default to this state)
623  TA0CTL |= TACLR;                 // Clearing TA0R, clock divider state, and the counter direction
624  TA0CCTL0 &= CAP;                // Setting capture mode to compare (just in case reset did not default to this state)
625  TA0CCR0 = TIMER_COUNT;          // Setting TA0CCR0 to 15000/16MHz = 0.9375ms (less than 1/2 watchdog period)
626  TA0CCTL0 |= CCIE;               // TA0CCR0 interrupt enabled
627  TA0CTL |= TASSEL_2 | ID_0 | MC_2; // Setting SMCLK, no divider, and entering continuous mode
628
629  // Clears watchdog timer and selects operating settings, page 359 in MSP430FR4xx and MSP430FR2xx Family User's Guide
630  WDTCTL = WDTPW | WDTSEL_0 | WDTCNTL | WDTIS_4; // SMCLK selected (16MHz), watchdog mode, counter cleared,
631                                               // 1/16MHz * 32768 = 2.05ms WDT period
632
633  // Bringing SPI CS line high after initialization (enable when documenting timing with oscilloscope...
634  // make sure to disable initial high at beginning of initialization function)
635  //P5OUT |= BIT0;
636 }
637
638 // Internal ADC sampling function
639 int analog (int channel)
640 {
641     // Perform single internal ADC conversion from selected channel,
642     // page 551-552/556-557/561 in MSP430FR4xx and MSP430FR2xx Family User's Guide
643     ADCCTL0 &= ADCENC;            // Un-enables ADC so that input channel can be changed
644     if (channel == 2)              // Changes input channel without changing reference voltage
645     {
646         ADCMCTL0 = ADCINCH_2 + ADCSREF_1; // A2 = +5, V with + reference voltage to internal 1.5V VREF
647                                             // and - reference voltage to AVSS (GND)
648     }
649     if (channel == 3)
650     {
651         ADCMCTL0 = ADCINCH_3 + ADCSREF_1; // A3 = -15, with + reference voltage to internal 1.5V VREF
652                                             // and - reference voltage to AVSS (GND)
653     }
654     if (channel == 4)
655     {
656         ADCMCTL0 = ADCINCH_4 + ADCSREF_1; // A4 = +15, with + reference voltage to internal 1.5V VREF
657                                             // and - reference voltage to AVSS (GND)
658     }
659     if (channel == 5)
660     {
661         ADCMCTL0 = ADCINCH_5 + ADCSREF_1; // A5 = Temp 1, with + reference voltage to internal 1.5V VREF
662                                             // and - reference voltage to AVSS (GND)
663     }
664     if (channel == 6)
665     {
666         ADCMCTL0 = ADCINCH_6 + ADCSREF_1; // A6 = Temp 2, with + reference voltage to internal 1.5V VREF
667                                             // and - reference voltage to AVSS (GND)
668     }
669     if (channel == 7)
670     {
671         ADCMCTL0 = ADCINCH_7 + ADCSREF_1; // A7 = Light Sensor, with + reference voltage to internal 1.5V VREF
672                                             // and - reference voltage to AVSS (GND)
673     }
674     ADCIFG &= ADCIFG0;            // Clears ADC conversion flag
675     ADCCTL0 |= ADCENC | ADCSC;    // Starts ADC conversion by enabling and starting conversion
676     while ((ADCIFG & ADCIFG0) == 0); // Polls for ADC conversion flag to be tripped
677     return ADCMEM0;               // Returns contents of ADC conversion memory register
678 }
679
680 // UART byte transmission function
681 void writeUART(char* sentence)

```

main.c

```

682 {
683 // UART TX information from page 597/601 in MSP430FR4xx and MSP430FR2xx Family User's Guide
684 int i;
685 for (i = 0; i < strlen(sentence); i++) {
686     while(!UCA0IFG&&UCTXIFG);
687     UCA0TXBUF = sentence[i];
688 }
689 return;
690 }
691
692 // GPIO initialization function
693 void GPIOinit()
694 {
695 // GPIO port configuration, page 306/327-328 in MSP430FR4xx and MSP430FR2xx Family User's Guide
696 // Ports 1-8 configured as outputs
697 P1DIR = 0xFF; P2DIR = 0xFF; P3DIR = 0xFF; P4DIR = 0xFF;
698 P5DIR = 0xFF; P6DIR = 0xFF; P7DIR = 0xFF; P8DIR = 0xFF;
699
700 // Ports 1-8 configured with pullup/pulldown resistors enabled
701 P1REN = 0xFF; P2REN = 0xFF; P3REN = 0xFF; P4REN = 0xFF;
702 P5REN = 0xFF; P6REN = 0xFF; P7REN = 0xFF; P8REN = 0xFF;
703
704 // Ports 1-8 configured with output registers set to 0 (outputs low)
705 P1OUT = 0x00; P2OUT = 0x00; P3OUT = 0x00; P4OUT = 0x00;
706 P5OUT = 0x00; P6OUT = 0x00; P7OUT = 0x00; P8OUT = 0x00;
707 return;
708 }
709
710 // CRC-16 calculator function
711 unsigned short crc16MakeBitwise2(unsigned short crc, unsigned short poly,
712     uint8_t *pmsg, unsigned int msg_size)
713 {
714 // TI provided CRC-16 calculator function
715 unsigned int i, j;
716 unsigned short msg;
717 for (i = 0; i < msg_size; i++)
718 {
719     msg = (*pmsg++ << 8);
720     for (j = 0; j < 8; j++)
721     {
722         if((msg ^ crc) >> 15) crc = (crc << 1) ^ poly;
723         else crc <<= 1;
724         msg <<= 1;
725     }
726 }
727 return(crc ^ CRC16_FINAL_XOR);
728 }
729

```


Appendix F

Testing Interface Software

PIP Boy Testing Interface GUI

```

classdef PIP_Boy < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        SetupPanel              matlab.ui.container.Panel
        CollectDataSwitchLabel  matlab.ui.control.Label
        CollectDataSwitch       matlab.ui.control.ToggleSwitch
        RunningLampLabel        matlab.ui.control.Label
        RunningLamp             matlab.ui.control.Lamp
        COMPortDropDownLabel    matlab.ui.control.Label
        COMPortDropDown         matlab.ui.control.DropDown
        SelectdesiredCOMportLabel  matlab.ui.control.Label
        SetCollectDataSwitchtoOnLabel  matlab.ui.control.Label
        ReviewincomingsdataLabel  matlab.ui.control.Label
        SetCollectDataSwitchtoOffLabel  matlab.ui.control.Label
        HousekeepingDataPanel    matlab.ui.container.Panel
        BoardIDEditFieldLabel    matlab.ui.control.Label
        BoardIDEditField        matlab.ui.control.NumericEditField
        Temp1EditFieldLabel      matlab.ui.control.Label
        Temp1EditField           matlab.ui.control.NumericEditField
        Temp2EditFieldLabel      matlab.ui.control.Label
        Temp2EditField           matlab.ui.control.NumericEditField
        CRCChecksumLampLabel     matlab.ui.control.Label
        CRCChecksumLamp         matlab.ui.control.Lamp
        Neg15VEditFieldLabel     matlab.ui.control.Label
        Neg15VEditField          matlab.ui.control.NumericEditField
        Pos15VEditFieldLabel     matlab.ui.control.Label
        Pos15VEditField          matlab.ui.control.NumericEditField
        Pos5VEditFieldLabel      matlab.ui.control.Label
        Pos5VEditField           matlab.ui.control.NumericEditField
        LightLuxEditFieldLabel   matlab.ui.control.Label
        LightLuxEditField        matlab.ui.control.NumericEditField
        ScienceDataPanel         matlab.ui.container.Panel
        UIAxes                   matlab.ui.control.UIAxes
        UIAxes2                   matlab.ui.control.UIAxes
        LowGainAverageEditFieldLabel  matlab.ui.control.Label
        LowGainAverageEditField    matlab.ui.control.NumericEditField
        LowGainSTDEditFieldLabel  matlab.ui.control.Label
        LowGainSTDEditField        matlab.ui.control.NumericEditField
        HighGainSTDEditFieldLabel  matlab.ui.control.Label
        HighGainSTDEditField        matlab.ui.control.NumericEditField
        HighGainAverageEditFieldLabel  matlab.ui.control.Label
        HighGainAverageEditField    matlab.ui.control.NumericEditField
    end

    properties (Access = private)
        t % Serial port
        N=50 % Number to average
        % CRC lookup table

        Crc_ui16LookupTable=[0,4129,8258,12387,16516,20645,24774,28903,33032,37161,41290,45419,49
548,...

53677,57806,61935,4657,528,12915,8786,21173,17044,29431,25302,37689,33560,45947,41818,542
05,...

50076,62463,58334,9314,13379,1056,5121,25830,29895,17572,21637,42346,46411,34088,38153,58
862,...

62927,50604,54669,13907,9842,5649,1584,30423,26358,22165,18100,46939,42874,38681,34616,63
455,...

```

```

59390,55197,51132,18628,22757,26758,30887,2112,6241,10242,14371,51660,55789,59790,63919,3
5144,...

39273,43274,47403,23285,19156,31415,27286,6769,2640,14899,10770,56317,52188,64447,60318,3
9801,...

35672,47931,43802,27814,31879,19684,23749,11298,15363,3168,7233,60846,64911,52716,56781,4
4330,...

48395,36200,40265,32407,28342,24277,20212,15891,11826,7761,3696,65439,61374,57309,53244,4
8923,...

44858,40793,36728,37256,33193,45514,41451,53516,49453,61774,57711,4224,161,12482,8419,204
84,...

16421,28742,24679,33721,37784,41979,46042,49981,54044,58239,62302,689,4752,8947,13010,169
49,...

21012,25207,29270,46570,42443,38312,34185,62830,58703,54572,50445,13538,9411,5280,1153,29
798,...

25671,21540,17413,42971,47098,34713,38840,59231,63358,50973,55100,9939,14066,1681,5808,26
199,...

30326,17941,22068,55628,51565,63758,59695,39368,35305,47498,43435,22596,18533,30726,26663
,6336,...

2273,14466,10403,52093,56156,60223,64286,35833,39896,43963,48026,19061,23124,27191,31254,
2801,6864,...

10931,14994,64814,60687,56684,52557,48554,44427,40424,36297,31782,27655,23652,19525,15522
,11395,...

7392,3265,61215,65342,53085,57212,44955,49082,36825,40952,28183,32310,20053,24180,11923,1
6050,3793,7920]
end

methods (Access = private)

end

methods (Access = private)

% Code that executes after component creation
function startupFcn(app)
% Making sure everything is closed and ports are clear
fclose('all');
delete(instrfind);

% Setting up dropdown refresh item
app.COMPortDropDown.Items=cellstr([seriallist 'Refresh']);
app.COMPortDropDown.Value=app.COMPortDropDown.Items(1);

% Setting up plots
app.UIAxes.DataAspectRatioMode = 'auto';
app.UIAxes2.DataAspectRatioMode = 'auto';
app.UIAxes.PlotBoxAspectRatio=[1.77 1 1];
app.UIAxes2.PlotBoxAspectRatio=[1.77 1 1];
end

% Value changed function: COMPortDropDown

```

```

function COMPortDropDownValueChanged(app, event)
    delete(instrfind);
    value = app.COMPortDropDown.Value;
    delete(instrfind);
    if strcmp(value, 'Refresh')==0
        app.t=serial(value);
        set(app.t, 'BaudRate', 115200);
        app.t.Timeout=1;
        fopen(app.t);
    end
    app.COMPortDropDown.Items=cellstr([serialist 'Refresh']);
end

% Value changed function: CollectDataSwitch
function CollectDataSwitchValueChanged(app, event)
    value = app.CollectDataSwitch.Value;
    % Resetting data counter and positive 5V value
    i=1;
    voltage_pos5=0;

    % Continuously polling instrument for data
    while strcmp(value, 'On')==1 && strcmp(app.COMPortDropDown.Value, 'Refresh')==0
        % Changing status light
        app.RunningLamp.Color=[0 0.75 0];

        % Collecting data point
        fprintf(app.t, '%c', 'U');
        x=fread(app.t, 8, 'uchar');
        % If no read, repeat data point collection
        while isempty(x)==1
            fprintf(app.t, '%c', 'U');
            x=fread(app.t, 8, 'uchar');
        end
        % Parsing data into 6 values
        x1=bitget(x(1), 8:-1:1, 'uint8');           % First byte values
        PIP_Data(i,1)=4*x1(1)+2*x1(2)+x1(3);       % Static board ID
        PIP_Data(i,2)=4*x1(4)+2*x1(5)+x1(6);       % Cycling housekeeping ID
        PIP_Data(i,3)=512*x1(7)+256*x1(8)+x(2);    % Housekeeping value
        PIP_Data(i,4)=256*x(3)+x(4);               % High gain channel
        PIP_Data(i,5)=256*x(5)+x(6);               % Low gain channel
        PIP_Data(i,6)=256*x(7)+x(8);               % CRC

        % (Hopefully not, lol!) updating board ID
        app.BoardIDEditField.Value=PIP_Data(i,1);
        %app.BoardIDEditField.Value=sprintf('0b%s', PIP_Data(i,1));

        % CRC-16-CCITT-FALSE
        % G(x) = x16 + x12 + x5 + 1, 0x1021 polynomial and 0xFFFF initial value
        checksum=hex2dec('FFFF');
        for j=1:length(x(1:6))
            ui8LookupTableIndex=bitxor(x(j), uint8(bitshift(checksum, -8)));

checksum=bitxor(app.Crc_ui16LookupTable(double(ui8LookupTableIndex)+1), mod(bitshift(checksum, 8), 65536));
        end
        % Updating CRC error lamp
        if checksum==PIP_Data(i,6)
            app.CRCChecksumLamp.Color=[0 0.75 0];
        elseif checksum~=PIP_Data(i,6)
            app.CRCChecksumLamp.Color=[1 0 0];
        end

        % Updating Housekeeping values
        if PIP_Data(i,2)==0 % +5V
            voltage_pos5=double((PIP_Data(i,3)/1024)*(1.5/0.28057554));

```

```

app.Pos5VEditField.Value=voltage_pos5;
if voltage_pos5>4.95 && voltage_pos5<5.05 % Inside 1%
    app.Pos5VEditField.FontColor=[0 0.75 0];
elseif voltage_pos5>4.75 && voltage_pos5<4.95 % Inside 5%
    app.Pos5VEditField.FontColor=[1 0.5 0];
elseif voltage_pos5>5.05 && voltage_pos5<5.25 % Inside 5%
    app.Pos5VEditField.FontColor=[1 0.5 0];
elseif voltage_pos5<4.75 || voltage_pos5>5.25 % Outside 5%
    app.Pos5VEditField.FontColor=[1 0 0];
end
elseif PIP_Data(i,2)==1 % -15V
    voltage_neg15=((PIP_Data(i,3)/1024)*(1.5/0.37106918))-
(0.76851852*voltage_pos5)/(0.23148148);
    app.Neg15VEditField.Value=voltage_neg15;
    if voltage_neg15>-15.15 && voltage_neg15<-14.85 % Inside 1%
        app.Neg15VEditField.FontColor=[0 0.75 0];
    elseif voltage_neg15>-15.75 && voltage_neg15<-15.15 % Inside 5%
        app.Neg15VEditField.FontColor=[1 0.5 0];
    elseif voltage_neg15>-14.85 && voltage_neg15<-14.25 % Inside 5%
        app.Neg15VEditField.FontColor=[1 0.5 0];
    elseif voltage_neg15<-15.75 || voltage_neg15>-14.25 % Outside 5%
        app.Neg15VEditField.FontColor=[1 0 0];
    end
elseif PIP_Data(i,2)==2 % +15V
    voltage_pos15=(PIP_Data(i,3)/1024)*(1.5/0.09502262);
    app.Pos15VEditField.Value=voltage_pos15;
    if voltage_pos15>14.85 && voltage_pos15<15.15 % Inside 1%
        app.Pos15VEditField.FontColor=[0 0.75 0];
    elseif voltage_pos15>14.25 && voltage_pos15<14.85 % Inside 5%
        app.Pos15VEditField.FontColor=[1 0.5 0];
    elseif voltage_pos15>15.15 && voltage_pos15<15.75 % Inside 5%
        app.Pos15VEditField.FontColor=[1 0.5 0];
    elseif voltage_pos15<14.25 || voltage_pos15>15.75 % Outside 5%
        app.Pos15VEditField.FontColor=[1 0 0];
    end
elseif PIP_Data(i,2)==3 % Temp 1
    temp_1=(PIP_Data(i,3)/1024)*(1.5/0.00422);
    app.Temp1EditField.Value=temp_1;
elseif PIP_Data(i,2)==4 % Temp 2
    temp_2=(PIP_Data(i,3)/1024)*(1.5/0.00422);
    app.Temp2EditField.Value=temp_2;
elseif PIP_Data(i,2)==5 % Light sensor
    light_lux=10^((PIP_Data(i,3)/1024)*(1.5/0.3));
    app.LightLuxEditField.Value=light_lux;
end

% Updating low/high gain data
if i>=app.N
    LowMean=mean(PIP_Data((i-app.N+1):i,5));
    HighMean=mean(PIP_Data((i-app.N+1):i,4));
    LowSTD=std(PIP_Data((i-app.N+1):i,5));
    HighSTD=std(PIP_Data((i-app.N+1):i,4));
    app.LowGainAverageEditField.Value=LowMean;
    app.HighGainAverageEditField.Value=HighMean;
    app.LowGainSTDEditField.Value=LowSTD;
    app.HighGainSTDEditField.Value=HighSTD;

    % Plotting low gain data
    app.UIAxes.XLim=[(i-app.N+1) i];
    if LowSTD~=0
        app.UIAxes.YLim=[(LowMean-3*LowSTD) (LowMean+3*LowSTD)];
    else
        app.UIAxes.YLim=[-1 1];
    end
    app.UIAxes.XTick=(i-app.N+1):i;

```

```

    app.UIAxes.YTickMode='auto';
    app.UIAxes.Position = [20 5 300 185];
    plot(app.UIAxes,1:length(PIP_Data(:,5)),PIP_Data(:,5))

    % Plotting high gain data
    app.UIAxes2.XLim=[(i-app.N+1) i];
    if HighSTD~=0
        app.UIAxes2.YLim=[(HighMean-3*HighSTD) (HighMean+3*HighSTD)];
    else
        app.UIAxes2.YLim=[-1 1];
    end
    app.UIAxes2.XTick=(i-app.N+1):i;
    app.UIAxes2.YTickMode='auto';
    app.UIAxes2.Position = [320 5 300 185];
    plot(app.UIAxes2,1:length(PIP_Data(:,4)),PIP_Data(:,4))
end

% Incrementing data counter
i=i+1;

% Poll switch value and update while loop exit if set
value=app.CollectDataSwitch.Value;
drawnow;
end

% Changing status light
app.RunningLamp.Color=[1 0 0];

% Writing collected data to storage file

end

% Close request function: UIFigure
function UIFigureCloseRequest(app, event)
    fclose('all');
    delete(instrfind);
    delete(app);
end
end

% App initialization and construction
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create UIFigure
    app.UIFigure = uifigure;
    app.UIFigure.Position = [100 100 640 480];
    app.UIFigure.Name = 'UI Figure';
    app.UIFigure.CloseRequestFcn = createCallbackFcn(app, @UIFigureCloseRequest,
true);

    % Create SetupPanel
    app.SetupPanel = uipanel(app.UIFigure);
    app.SetupPanel.TitlePosition = 'centertop';
    app.SetupPanel.Title = 'Setup';
    app.SetupPanel.FontWeight = 'bold';
    app.SetupPanel.Position = [1 291 260 190];

    % Create CollectDataSwitchLabel
    app.CollectDataSwitchLabel = uilabel(app.SetupPanel);
    app.CollectDataSwitchLabel.HorizontalAlignment = 'center';
    app.CollectDataSwitchLabel.Position = [11.078125 15 71 15];
    app.CollectDataSwitchLabel.Text = 'Collect Data';

```

```

% Create CollectDataSwitch
app.CollectDataSwitch = uiswitch(app.SetupPanel, 'toggle');
app.CollectDataSwitch.ValueChangedFcn = createCallbackFcn(app,
@CollectDataSwitchValueChanged, true);
app.CollectDataSwitch.Position = [38 55 20 45];

% Create RunningLampLabel
app.RunningLampLabel = uilabel(app.SetupPanel);
app.RunningLampLabel.HorizontalAlignment = 'right';
app.RunningLampLabel.Position = [121.03125 23 50 15];
app.RunningLampLabel.Text = 'Running';

% Create RunningLamp
app.RunningLamp = uilamp(app.SetupPanel);
app.RunningLamp.Position = [186.03125 20 20 20];
app.RunningLamp.Color = [1 0 0];

% Create COMPortDropDownLabel
app.COMPortDropDownLabel = uilabel(app.SetupPanel);
app.COMPortDropDownLabel.HorizontalAlignment = 'right';
app.COMPortDropDownLabel.Position = [11.03125 142 60 15];
app.COMPortDropDownLabel.Text = 'COM Port';

% Create COMPortDropDown
app.COMPortDropDown = uidropdown(app.SetupPanel);
app.COMPortDropDown.Items = {'Refresh'};
app.COMPortDropDown.ValueChangedFcn = createCallbackFcn(app,
@COMPortDropDownValueChanged, true);
app.COMPortDropDown.Position = [86.03125 138 164.953125 22];
app.COMPortDropDown.Value = 'Refresh';

% Create SelectdesiredCOMportLabel
app.SelectdesiredCOMportLabel = uilabel(app.SetupPanel);
app.SelectdesiredCOMportLabel.Position = [81 115 147 15];
app.SelectdesiredCOMportLabel.Text = '- Select desired COM port';

% Create SetCollectDatasswitchtoOnLabel
app.SetCollectDatasswitchtoOnLabel = uilabel(app.SetupPanel);
app.SetCollectDatasswitchtoOnLabel.Position = [81 95 179 15];
app.SetCollectDatasswitchtoOnLabel.Text = '- Set Collect Data switch to
'On'';

% Create ReviewincomingdataLabel
app.ReviewincomingdataLabel = uilabel(app.SetupPanel);
app.ReviewincomingdataLabel.Position = [81 75 132 15];
app.ReviewincomingdataLabel.Text = '- Review incoming data';

% Create SetCollectDatasswitchtoOffLabel
app.SetCollectDatasswitchtoOffLabel = uilabel(app.SetupPanel);
app.SetCollectDatasswitchtoOffLabel.Position = [81 55 179 15];
app.SetCollectDatasswitchtoOffLabel.Text = '- Set Collect Data switch to
'Off'';

% Create HousekeepingDataPanel
app.HousekeepingDataPanel = uipanel(app.UIFigure);
app.HousekeepingDataPanel.TitlePosition = 'centertop';
app.HousekeepingDataPanel.Title = 'Housekeeping Data';
app.HousekeepingDataPanel.FontWeight = 'bold';
app.HousekeepingDataPanel.Position = [261 291 380 190];

% Create BoardIDEditFieldLabel
app.BoardIDEditFieldLabel = uilabel(app.HousekeepingDataPanel);
app.BoardIDEditFieldLabel.HorizontalAlignment = 'right';
app.BoardIDEditFieldLabel.Position = [10.03125 142 53 15];

```

```

app.BoardIDEditFieldLabel.Text = 'Board ID';

% Create BoardIDEditField
app.BoardIDEditField = uieditfield(app.HousekeepingDataPanel, 'numeric');
app.BoardIDEditField.ValueDisplayFormat = '%.0f';
app.BoardIDEditField.Editable = 'off';
app.BoardIDEditField.HorizontalAlignment = 'center';
app.BoardIDEditField.Position = [78.03125 138 100 22];

% Create Temp1EditFieldLabel
app.Temp1EditFieldLabel = uilabel(app.HousekeepingDataPanel);
app.Temp1EditFieldLabel.HorizontalAlignment = 'right';
app.Temp1EditFieldLabel.Position = [18.03125 102 45 15];
app.Temp1EditFieldLabel.Text = 'Temp 1';

% Create Temp1EditField
app.Temp1EditField = uieditfield(app.HousekeepingDataPanel, 'numeric');
app.Temp1EditField.ValueDisplayFormat = '%.2f';
app.Temp1EditField.Editable = 'off';
app.Temp1EditField.HorizontalAlignment = 'center';
app.Temp1EditField.Position = [78.03125 98 100 22];

% Create Temp2EditFieldLabel
app.Temp2EditFieldLabel = uilabel(app.HousekeepingDataPanel);
app.Temp2EditFieldLabel.HorizontalAlignment = 'right';
app.Temp2EditFieldLabel.Position = [18.03125 62 45 15];
app.Temp2EditFieldLabel.Text = 'Temp 2';

% Create Temp2EditField
app.Temp2EditField = uieditfield(app.HousekeepingDataPanel, 'numeric');
app.Temp2EditField.ValueDisplayFormat = '%.2f';
app.Temp2EditField.Editable = 'off';
app.Temp2EditField.HorizontalAlignment = 'center';
app.Temp2EditField.Position = [78.03125 58 100 22];

% Create CRCChecksumLampLabel
app.CRCChecksumLampLabel = uilabel(app.HousekeepingDataPanel);
app.CRCChecksumLampLabel.HorizontalAlignment = 'right';
app.CRCChecksumLampLabel.Position = [232.03125 23 92 15];
app.CRCChecksumLampLabel.Text = 'CRC Checksum';

% Create CRCChecksumLamp
app.CRCChecksumLamp = uilamp(app.HousekeepingDataPanel);
app.CRCChecksumLamp.Position = [339.03125 20 20 20];
app.CRCChecksumLamp.Color = [1 0 0];

% Create Neg15VEditFieldLabel
app.Neg15VEditFieldLabel = uilabel(app.HousekeepingDataPanel);
app.Neg15VEditFieldLabel.HorizontalAlignment = 'right';
app.Neg15VEditFieldLabel.Position = [192.03125 62 52 15];
app.Neg15VEditFieldLabel.Text = 'Neg 15V';

% Create Neg15VEditField
app.Neg15VEditField = uieditfield(app.HousekeepingDataPanel, 'numeric');
app.Neg15VEditField.ValueDisplayFormat = '%.3f';
app.Neg15VEditField.Editable = 'off';
app.Neg15VEditField.HorizontalAlignment = 'center';
app.Neg15VEditField.FontColor = [1 0 0];
app.Neg15VEditField.Position = [259.03125 58 100 22];

% Create Pos15VEditFieldLabel
app.Pos15VEditFieldLabel = uilabel(app.HousekeepingDataPanel);
app.Pos15VEditFieldLabel.HorizontalAlignment = 'right';
app.Pos15VEditFieldLabel.Position = [194.03125 102 50 15];
app.Pos15VEditFieldLabel.Text = 'Pos 15V';

```



```

% Create Pos15VEditField
app.Pos15VEditField = uieditfield(app.HousekeepingDataPanel, 'numeric');
app.Pos15VEditField.ValueDisplayFormat = '%.3f';
app.Pos15VEditField.Editable = 'off';
app.Pos15VEditField.HorizontalAlignment = 'center';
app.Pos15VEditField.FontColor = [1 0 0];
app.Pos15VEditField.Position = [259.03125 98 100 22];

% Create Pos5VEditFieldLabel
app.Pos5VEditFieldLabel = uilabel(app.HousekeepingDataPanel);
app.Pos5VEditFieldLabel.HorizontalAlignment = 'right';
app.Pos5VEditFieldLabel.Position = [200.03125 142 44 15];
app.Pos5VEditFieldLabel.Text = 'Pos 5V';

% Create Pos5VEditField
app.Pos5VEditField = uieditfield(app.HousekeepingDataPanel, 'numeric');
app.Pos5VEditField.ValueDisplayFormat = '%.3f';
app.Pos5VEditField.Editable = 'off';
app.Pos5VEditField.HorizontalAlignment = 'center';
app.Pos5VEditField.FontColor = [1 0 0];
app.Pos5VEditField.Position = [259.03125 138 100 22];

% Create LightLuxEditFieldLabel
app.LightLuxEditFieldLabel = uilabel(app.HousekeepingDataPanel);
app.LightLuxEditFieldLabel.HorizontalAlignment = 'right';
app.LightLuxEditFieldLabel.Position = [8.03125 23 55 15];
app.LightLuxEditFieldLabel.Text = 'Light Lux';

% Create LightLuxEditField
app.LightLuxEditField = uieditfield(app.HousekeepingDataPanel, 'numeric');
app.LightLuxEditField.ValueDisplayFormat = '%.0f';
app.LightLuxEditField.Editable = 'off';
app.LightLuxEditField.HorizontalAlignment = 'center';
app.LightLuxEditField.Position = [78.03125 19 100 22];

% Create ScienceDataPanel
app.ScienceDataPanel = uipanel(app.UIFigure);
app.ScienceDataPanel.TitlePosition = 'centertop';
app.ScienceDataPanel.Title = 'Science Data';
app.ScienceDataPanel.FontWeight = 'bold';
app.ScienceDataPanel.Position = [1 1 640 290];

% Create UIAxes
app.UIAxes = uiaxes(app.ScienceDataPanel);
title(app.UIAxes, 'Low Gain Channel')
xlabel(app.UIAxes, 'Sample')
ylabel(app.UIAxes, 'Counts')
app.UIAxes.DataAspectRatio = [1 1 1];
app.UIAxes.DataAspectRatioMode = 'manual';
app.UIAxes.XLim = [0 1];
app.UIAxes.XLimMode = 'manual';
app.UIAxes.YLim = [0 1];
app.UIAxes.YLimMode = 'manual';
app.UIAxes.CLim = [0 1];
app.UIAxes.CLimMode = 'manual';
app.UIAxes.GridColor = [0.15 0.15 0.15];
app.UIAxes.GridColorMode = 'manual';
app.UIAxes.MinorGridColor = [0.1 0.1 0.1];
app.UIAxes.MinorGridColorMode = 'manual';
app.UIAxes.Box = 'on';
app.UIAxes.XColor = [0.15 0.15 0.15];
app.UIAxes.XColorMode = 'manual';
app.UIAxes.XTick = [0 0.5 1];
app.UIAxes.XTickMode = 'manual';

```

```

app.UIAxes.YColor = [0.15 0.15 0.15];
app.UIAxes.YColorMode = 'manual';
app.UIAxes.YTick = [0 0.5 1];
app.UIAxes.YTickMode = 'manual';
app.UIAxes.XGrid = 'on';
app.UIAxes.YGrid = 'on';
app.UIAxes.Position = [20 5 300 185];

% Create UIAxes2
app.UIAxes2 = uiaxes(app.ScienceDataPanel);
title(app.UIAxes2, 'High Gain Channel')
xlabel(app.UIAxes2, 'Sample')
ylabel(app.UIAxes2, 'Counts')
app.UIAxes2.DataAspectRatio = [1 1 1];
app.UIAxes2.DataAspectRatioMode = 'manual';
app.UIAxes2.XLim = [0 1];
app.UIAxes2.XLimMode = 'manual';
app.UIAxes2.YLim = [0 1];
app.UIAxes2.YLimMode = 'manual';
app.UIAxes2.CLim = [0 1];
app.UIAxes2.CLimMode = 'manual';
app.UIAxes2.GridColor = [0.15 0.15 0.15];
app.UIAxes2.GridColorMode = 'manual';
app.UIAxes2.MinorGridColor = [0.1 0.1 0.1];
app.UIAxes2.MinorGridColorMode = 'manual';
app.UIAxes2.Box = 'on';
app.UIAxes2.XColor = [0.15 0.15 0.15];
app.UIAxes2.XColorMode = 'manual';
app.UIAxes2.XTick = [0 0.5 1];
app.UIAxes2.XTickMode = 'manual';
app.UIAxes2.YColor = [0.15 0.15 0.15];
app.UIAxes2.YColorMode = 'manual';
app.UIAxes2.YTick = [0 0.5 1];
app.UIAxes2.YTickMode = 'manual';
app.UIAxes2.XGrid = 'on';
app.UIAxes2.YGrid = 'on';
app.UIAxes2.Position = [320 5 300 185];

% Create LowGainAverageEditFieldLabel
app.LowGainAverageEditFieldLabel = uilabel(app.ScienceDataPanel);
app.LowGainAverageEditFieldLabel.HorizontalAlignment = 'right';
app.LowGainAverageEditFieldLabel.Position = [51.03125 242 103 15];
app.LowGainAverageEditFieldLabel.Text = 'Low Gain Average';

% Create LowGainAverageEditField
app.LowGainAverageEditField = uieditfield(app.ScienceDataPanel, 'numeric');
app.LowGainAverageEditField.ValueDisplayFormat = '%.1f';
app.LowGainAverageEditField.Editable = 'off';
app.LowGainAverageEditField.HorizontalAlignment = 'center';
app.LowGainAverageEditField.Position = [169.03125 238 100 22];

% Create LowGainSTDEditFieldLabel
app.LowGainSTDEditFieldLabel = uilabel(app.ScienceDataPanel);
app.LowGainSTDEditFieldLabel.HorizontalAlignment = 'right';
app.LowGainSTDEditFieldLabel.Position = [71.03125 202 83 15];
app.LowGainSTDEditFieldLabel.Text = 'Low Gain STD';

% Create LowGainSTDEditField
app.LowGainSTDEditField = uieditfield(app.ScienceDataPanel, 'numeric');
app.LowGainSTDEditField.ValueDisplayFormat = '%.2f';
app.LowGainSTDEditField.Editable = 'off';
app.LowGainSTDEditField.HorizontalAlignment = 'center';
app.LowGainSTDEditField.Position = [169.03125 198 100 22];

% Create HighGainSTDEditFieldLabel

```

```

app.HighGainSTDEditFieldLabel = uilabel(app.ScienceDataPanel);
app.HighGainSTDEditFieldLabel.HorizontalAlignment = 'right';
app.HighGainSTDEditFieldLabel.Position = [370.03125 202 85 15];
app.HighGainSTDEditFieldLabel.Text = 'High Gain STD';

% Create HighGainSTDEditField
app.HighGainSTDEditField = uieditfield(app.ScienceDataPanel, 'numeric');
app.HighGainSTDEditField.ValueDisplayFormat = '%.2f';
app.HighGainSTDEditField.Editable = 'off';
app.HighGainSTDEditField.HorizontalAlignment = 'center';
app.HighGainSTDEditField.Position = [470.03125 198 100 22];

% Create HighGainAverageEditFieldLabel
app.HighGainAverageEditFieldLabel = uilabel(app.ScienceDataPanel);
app.HighGainAverageEditFieldLabel.HorizontalAlignment = 'right';
app.HighGainAverageEditFieldLabel.Position = [349.03125 242 106 15];
app.HighGainAverageEditFieldLabel.Text = 'High Gain Average';

% Create HighGainAverageEditField
app.HighGainAverageEditField = uieditfield(app.ScienceDataPanel, 'numeric');
app.HighGainAverageEditField.ValueDisplayFormat = '%.1f';
app.HighGainAverageEditField.Editable = 'off';
app.HighGainAverageEditField.HorizontalAlignment = 'center';
app.HighGainAverageEditField.Position = [470.03125 238 100 22];
end
end

methods (Access = public)

% Construct app
function app = PIP_Boy

    % Create and configure components
    createComponents(app)

    % Register the app with App Designer
    registerApp(app, app.UIFigure)

    % Execute the startup function
    runStartupFcn(app, @startupFcn)

    if nargin == 0
        clear app
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end

```

Appendix G

Calibration Software

PIP Bias Calibration Data Collection Software

```

% MATLAB script for communication with PIP and DMM6500 to collect bias
% voltages and temperature housekeeping throughout temperature cycling
% Liam Gunter 2019

%%
% Setup
clc;
clear all;
close all;
fclose('all');
delete(instrfind);
% Measurement duration
duration=2.33; % hours
% Measurement delay
delay=1; % seconds
% PIP trigger word
TriggerWord='U';
% DMM resource name
VisaResourceName='USB::0x05E6::0x6500::04386498::INSTR';
% Generate dated file name
date_time=fix(clock);
date_time_str=sprintf('%04d%02d%02d_%02d%02d',date_time(1),date_time(2),date_time(3),date_time(4),date_time(5));
file_str=sprintf('%s',mfilename);
text_str=sprintf('%s_%s.txt',file_str,date_time_str);
% Open file for recording data
record_file=fopen(text_str,'w');
% Create objects and establish connections
t=serial('COM25');
set(t,'BaudRate',115200);
fopen(t);
t.Timeout=1;
DMM6500=visa('ni',VisaResourceName);
fopen(DMM6500);
% Enable DMM voltage mode
fprintf(DMM6500,'*RST');
fprintf(DMM6500,':SENS:FUNC ''VOLT''');
% Start timer
tic;

%%
% Monitor temperature profile
while toc<(duration*3600)
    % Recording 5 PIP measurements to get both temperature
    % Printing trigger word to instrument
    fprintf(t,'%c',TriggerWord);
    % Reading 6 bytes return data
    x=fread(t,20,'uchar');
    % If timeout error occurs, repeat measurement request
    while isempty(x)==1
        fprintf(t,'%c',TriggerWord);
        x=fread(t,20,'uchar');
    end

    % Record instrument temperatures
    Temp1=256*x(15)+x(16);
    Temp2=256*x(17)+x(18);

    % Record DMM voltage measurement
    fprintf(DMM6500,':READ?');
    DMM_Voltage=str2double(fscanf(DMM6500));

```

```
% Writing data to file

fprintf(record_file, '%s\t%s\t%s\t%s\r\n', num2str(toc), num2str(DMM_Voltage, '%.5f'), num2str(
(Temp1), num2str(Temp2));

    % Printing readings to terminal for monitoring
    disp([num2str(toc), ': ', num2str(DMM_Voltage), ' ', num2str(Temp1), '
', num2str(Temp2)]);

    % Waiting delay period
    pause(delay);
end

%%
% Cleaning up
close all;
fclose('all');
delete(instrfind);
```

PIP Bias Calibration Analysis Software

```

% MATLAB PIP bias calibration analysis script
% Outputs bias calibration coefficient file
% Liam Gunter 2019

%%
% Setup
clc;
clear all;
% close all;
fclose('all');
delete(instrfind);
% ADC cutoff value
ADC_Cutoff=1020;
% User identified filename
date_time_str=sprintf('20190311_1911');
file_str=sprintf('PIP_BiasMeasurement');
Bias_Str=[string(sprintf('%s_%s.txt',file_str,date_time_str))];
% Open file for recording coefficient data
record_file_str=[string(sprintf('BiasCoefficients_%s.txt',date_time_str))];
record_file=fopen(record_file_str,'w');
% Reading data from file
BiasData=load(Bias_Str);

%%
% Organizing data into variables
Time=BiasData(:,1);
Bias=BiasData(:,2);
Temp1=BiasData(:,3);
Temp2=BiasData(:,4);

% Excluding nonlinear ADC temperature counts sections
ind=find(Temp1<ADC_Cutoff);
Time=Time(ind);
Bias=Bias(ind);
Temp1=Temp1(ind);
Temp2=Temp2(ind);
ind=find(Temp2<ADC_Cutoff);
Time=Time(ind);
Bias=Bias(ind);
Temp1=Temp1(ind);
Temp2=Temp2(ind);

% Constructing offset vector
Offset=ones(length(Bias),1);

% Calculating coefficients for linear temperature relationship
coeff=[Offset,Temp1,Temp2]\Bias;
calBias=coeff(1)*Offset+coeff(2)*Temp1+coeff(3)*Temp2;
error=Bias-calBias;

% Calculating coefficients for square temperature relationship
coeff2=[Offset,Temp1,Temp2,Temp1.^2,Temp2.^2]\Bias;
calBias2=coeff2(1)*Offset+coeff2(2)*Temp1+coeff2(3)*Temp2+coeff2(4)*Temp1.^2+coeff2(5)*Temp2.^2;
error2=Bias-calBias2;

% Plotting error for linear and square relationships
figure;
plot(Time./60, error*1e6,'LineStyle','-'); % Time in minutes, error in uV
hold on
plot(Time./60, error2*1e6,'r','LineStyle',':');
%ylim([-100 100]);

```

```
set(gcf, 'color', 'white');
set(gca, 'FontSize', 14);
set(gca, 'LineWidth', 2);
xlabel('Time (min)', 'fontsize', 14);
ylabel('Volts (\muV)', 'fontsize', 14);
title('Bias Calibration Fit Error', 'fontsize', 14);
set(findall(gca, 'Type', 'Line'), 'LineWidth', 2);
legend('Linear Fit', 'Quadratic Fit');
legend('Location', 'southeast');
hold off;

%%
% Recording bias coefficient data to file for reference
fprintf(record_file, '%s\t%s\t%s\r\n', num2str(coeff(1)), num2str(coeff(2)), num2str(coeff(3)
));
fprintf(record_file, '%s\t%s\t%s\t%s\t%s\r\n', num2str(coeff2(1)), num2str(coeff2(2)), ...
        num2str(coeff2(3)), num2str(coeff2(4)), num2str(coeff2(5)));

%%
% Cleaning up
fclose('all');
delete(instrfind);
```


PIP Instrument Calibration Data Collection Software

```

% MATLAB script for calibration on PIP
% Communication with PIP to collect test data for analysis
% Liam Gunter 2019

%%
% Setup
clc;
clear all;
close all;
fclose('all');
delete(instrfind);
beep on;
% Parameters
duration=2.33; % Hours
N=500;
TriggerWord='U';
BytesExpected=20;
resistor=[ 0.5
           1
           6.04
           56
           100
           1000]*1e6;
% Generate dated file name
date_time=fix(clock);
date_time_str=sprintf('%04d%02d%02d_%02d%02d',date_time(1),date_time(2),date_time(3),date_time(4),date_time(5));
file_str=sprintf('%s',mfilename);
text_str=sprintf('%s_%s.txt',file_str,date_time_str);
% Open file for recording data
record_file=fopen(text_str,'w');
% Create objects and establish connections
t=serial('COM25');
set(t,'BaudRate',115200);
fopen(t);
t.Timeout=1;
% Starting timer
tic;

%%
% Reading data from PIP for a specified number of trials for each resistor
j=1;
while toc<(duration*3600)
    disp(['Trial ',num2str(j),':']);
    for i=1:length(resistor)
        % Displaying resistor value to user and waiting for input
        beep;
        if resistor(i)<1e6
            disp(['Press key for ',num2str(resistor(i)/1e3),'kOhm resistor']);
        elseif resistor(i)<1e9 && resistor(i)>=1e6
            disp(['Press key for ',num2str(resistor(i)/1e6),'MOhm resistor']);
        else
            disp(['Press key for ',num2str(resistor(i)/1e9),'GOhm resistor']);
        end
        pause;
        beep;
        % Delaying 10 seconds to allow reading to settle
        pause(10);
        % Taking data from PIP
        for k=1:N
            % Printing trigger word to instrument
            fprintf(t,'%c',TriggerWord);
            % Reading [BytesExpected] bytes return data

```

```

x(k,:)=fread(t,BytesExpected,'uchar');
% If timeout error occurs, repeat measurement request
while isempty(x(k,:))==1
    fprintf(t,'%c',TriggerWord);
    x(k,:)=fread(t,BytesExpected,'uchar');
end
end
% Writing data to file
for k=1:length(x(:,1))
    fprintf(record_file,'%s\t%s\t%s\t%s\r\n',num2str(j),num2str(resistor(i)),...
        num2str(k),num2str(x(k,:)));
end
% Printing first and last measurement to terminal for verification
disp(['LG = ',num2str(256*x(1,5)+x(1,6)), ' | HG = ',num2str(256*x(1,3)+x(1,4)), '
| Temp1 = ',num2str(256*x(1,15)+x(1,16)), ' | Temp2 = ',num2str(256*x(1,17)+x(1,18))]);
disp(['LG = ',num2str(256*x(end,5)+x(end,6)), ' | HG =
',num2str(256*x(end,3)+x(end,4)), ' | Temp1 = ',num2str(256*x(end,15)+x(end,16)), ' | Temp2
= ',num2str(256*x(end,17)+x(end,18))]);
end
j=j+1;
end

%% Cleaning up
close all;
fclose('all');
delete(instrfind);

```

PIP Instrument Calibration Analysis Software

```

% MATLAB PIP instrument calibration analysis script
% Outputs instrument calibration coefficient file
% Liam Gunter 2019

%%
% Setup
clc;
clear all;
close all;
fclose('all');
delete(instrfind);
% User identified filename for pre-calibration data
date_time_str=sprintf('20190304_2025');
file_str=sprintf('PIP_Calibration');
Calibration_Str=[string(sprintf('%s_%s.txt',file_str,date_time_str))];
% Opening bias data file
bias_file=fopen('BiasCoefficients_20190304_1635.txt','r');
% ADC cutoff value
ADC_Cutoff=1020;
% Number of samples
N=500;
% Resistor
resistor=[ 0.5
           1
           6.04
           56
           100
           1000]*1e6;
measured_resistors=[ 0.499916
                    0.998433
                    6.07590
                    56.0700
                    100.230
                    1001.31]*1e6;
% Reading data from pre-calibration file
CalibrationDataFull=load(Calibration_Str);
CalibrationData=CalibrationDataFull;
% Reading bias data from bias file
bias_coeff=str2num(fgetl(bias_file));
bias_coeff2=str2num(fgetl(bias_file));

%%

% Calculating low/high gain and temperature counts from raw data
LowGain=256*CalibrationData(:,8)+CalibrationData(:,9);
HighGain=256*CalibrationData(:,6)+CalibrationData(:,7);
Temp1=256*CalibrationData(:,18)+CalibrationData(:,19);
Temp2=256*CalibrationData(:,20)+CalibrationData(:,21);

% Excluding saturated ADC temperature counts sections
ind=find(Temp1<ADC_Cutoff);
CalibrationData=CalibrationData(ind,:);
LowGain=LowGain(ind,:);
HighGain=HighGain(ind,:);
Temp1=Temp1(ind,:);
Temp2=Temp2(ind,:);
ind=find(Temp2<ADC_Cutoff);
CalibrationData=CalibrationData(ind,:);
LowGain=LowGain(ind,:);
HighGain=HighGain(ind,:);
Temp1=Temp1(ind,:);
Temp2=Temp2(ind,:);

```

```

% Isolating the indices of the last 200 samples of each trial resistor, and averaging
them to 4 values
ind1=find(CalibrationData(:,3)>=301 & CalibrationData(:,3)<=350);
x(:,1)=[ind1(1);ind1((find(~(diff(ind1)==1))+1))];
x(:,2)=[ind1(find(~(diff(ind1)==1)));ind1(end)];
ind2=find(CalibrationData(:,3)>=351 & CalibrationData(:,3)<=400);
x(:,3)=[ind2(1);ind2((find(~(diff(ind2)==1))+1))];
x(:,4)=[ind2(find(~(diff(ind2)==1)));ind2(end)];
ind3=find(CalibrationData(:,3)>=401 & CalibrationData(:,3)<=450);
x(:,5)=[ind3(1);ind3((find(~(diff(ind3)==1))+1))];
x(:,6)=[ind3(find(~(diff(ind3)==1)));ind3(end)];
ind4=find(CalibrationData(:,3)>=451 & CalibrationData(:,3)<=500);
x(:,7)=[ind4(1);ind4((find(~(diff(ind4)==1))+1))];
x(:,8)=[ind4(find(~(diff(ind4)==1)));ind4(end)];

% Constructing averaged vectors with these indices
for i=1:length(x(:,1))
    if ((CalibrationData(x(i,1),2)==resistor(1)) ||...
        (CalibrationData(x(i,1),2)==resistor(2)) ||...
        (CalibrationData(x(i,1),2)==resistor(3)))
        LG_ind((i*4-3):(i*4),1)=1;
    else
        LG_ind((i*4-3):(i*4),1)=0;
    end
    if ((CalibrationData(x(i,1),2)==resistor(4)) ||...
        (CalibrationData(x(i,1),2)==resistor(5)) ||...
        (CalibrationData(x(i,1),2)==resistor(6)))
        HG_ind((i*4-3):(i*4),1)=1;
    else
        HG_ind((i*4-3):(i*4),1)=0;
    end
    LG(i*4-3,1)=mean(LowGain(x(i,1):x(i,2)));
    LG(i*4-2,1)=mean(LowGain(x(i,3):x(i,4)));
    LG(i*4-1,1)=mean(LowGain(x(i,5):x(i,6)));
    LG(i*4,1)=mean(LowGain(x(i,7):x(i,8)));
    HG(i*4-3,1)=mean(HighGain(x(i,1):x(i,2)));
    HG(i*4-2,1)=mean(HighGain(x(i,3):x(i,4)));
    HG(i*4-1,1)=mean(HighGain(x(i,5):x(i,6)));
    HG(i*4,1)=mean(HighGain(x(i,7):x(i,8)));
    T1(i*4-3,1)=mean(Temp1(x(i,1):x(i,2)));
    T1(i*4-2,1)=mean(Temp1(x(i,3):x(i,4)));
    T1(i*4-1,1)=mean(Temp1(x(i,5):x(i,6)));
    T1(i*4,1)=mean(Temp1(x(i,7):x(i,8)));
    T2(i*4-3,1)=mean(Temp2(x(i,1):x(i,2)));
    T2(i*4-2,1)=mean(Temp2(x(i,3):x(i,4)));
    T2(i*4-1,1)=mean(Temp2(x(i,5):x(i,6)));
    T2(i*4,1)=mean(Temp2(x(i,7):x(i,8)));
    Res(i*4-3,1)=measured_resistors(CalibrationData(x(i,1),2)==resistor);
    Res(i*4-2,1)=measured_resistors(CalibrationData(x(i,1),2)==resistor);
    Res(i*4-1,1)=measured_resistors(CalibrationData(x(i,1),2)==resistor);
    Res(i*4,1)=measured_resistors(CalibrationData(x(i,1),2)==resistor);
end

LG_ind=find(LG_ind);
HG_ind=find(HG_ind);

% Calculating bias and current
Offset=ones(length(LG(:,1)),1);
CalBias=bias_coeff(1).*Offset+bias_coeff(2).*T1+bias_coeff(3).*T2;
CalBias2=bias_coeff2(1).*Offset+bias_coeff2(2).*T1+bias_coeff2(3).*T2+...
    bias_coeff2(4).*T1.^2+bias_coeff2(5).*T2.^2;
% Current=-CalBias./Res;
Current=-CalBias2./Res;

```

```

% Calculating coefficients
coeff_LG=[Offset(LG_ind),LG(LG_ind),T1(LG_ind),T2(LG_ind)]\Current(LG_ind);
coeff_HG=[Offset(HG_ind),HG(HG_ind),T1(HG_ind),T2(HG_ind)]\Current(HG_ind);

coeff_LG3=[Offset(LG_ind),LG(LG_ind),T1(LG_ind),T2(LG_ind),...
    LG(LG_ind).^2,T1(LG_ind).^2,T2(LG_ind).^2,...
    LG(LG_ind).^0.5,T1(LG_ind).^3,T2(LG_ind).^3]\Current(LG_ind);
coeff_HG3=[Offset(HG_ind),HG(HG_ind),T1(HG_ind),T2(HG_ind),...
    HG(HG_ind).^2,T1(HG_ind).^2,T2(HG_ind).^2,...
    HG(HG_ind).^0.5,T1(HG_ind).^3,T2(HG_ind).^3]\Current(HG_ind);

% Calculating theoretical current with coefficients
% Linear coefficients only
CalCurrent_LG=coeff_LG(1).*Offset(LG_ind)+coeff_LG(2).*LG(LG_ind)+coeff_LG(3).*T1(LG_ind)
+coeff_LG(4).*T2(LG_ind);
CalCurrent_HG=coeff_HG(1).*Offset(HG_ind)+coeff_HG(2).*HG(HG_ind)+coeff_HG(3).*T1(HG_ind)
+coeff_HG(4).*T2(HG_ind);

% Coefficients on channel and temp1/temp2 squares and cubes
CalCurrent_LG3=coeff_LG3(1).*Offset(LG_ind)+coeff_LG3(2).*LG(LG_ind)+...
    coeff_LG3(3).*T1(LG_ind)+coeff_LG3(4).*T2(LG_ind)+...

coeff_LG3(5).*LG(LG_ind).^2+coeff_LG3(6).*T1(LG_ind).^2+coeff_LG3(7).*T2(LG_ind).^2+...

coeff_LG3(8).*LG(LG_ind).^0.5+coeff_LG3(9).*T1(LG_ind).^3+coeff_LG3(10).*T2(LG_ind).^3;
CalCurrent_HG3=coeff_HG3(1).*Offset(HG_ind)+coeff_HG3(2).*HG(HG_ind)+...
    coeff_HG3(3).*T1(HG_ind)+coeff_HG3(4).*T2(HG_ind)+...

coeff_HG3(5).*HG(HG_ind).^2+coeff_HG3(6).*T1(HG_ind).^2+coeff_HG3(7).*T2(HG_ind).^2+...

coeff_HG3(8).*HG(HG_ind).^0.5+coeff_HG3(9).*T1(HG_ind).^3+coeff_HG3(10).*T2(HG_ind).^3;

% Calculating error in current
error_LG=Current(LG_ind)-CalCurrent_LG;
error_HG=Current(HG_ind)-CalCurrent_HG;

error_LG3=Current(LG_ind)-CalCurrent_LG3;
error_HG3=Current(HG_ind)-CalCurrent_HG3;

% Plotting low gain channel fit error
figure;
plot(error_LG/1e-9,'LineStyle','-');
hold on;
plot(error_LG3/1e-9,'LineStyle',':');
set(gca,'FontSize',14);
set(gca,'LineWidth',2);
xlabel('Low Gain Sample #','fontsize',14);
ylabel('Error (nA)','fontsize',14);
legend('Linear coefficients','Nonlinear coefficients');
legend('Location','SouthWest');
title('Low Gain Channel Fit Error','fontsize',14);
set(findall(gca,'Type','Line'),'LineWidth',2);
hold off;

% Plotting high gain channel fit error
figure;
plot(error_HG/1e-9,'LineStyle','-');
hold on;
plot(error_HG3/1e-9,'LineStyle',':');
set(gca,'FontSize',14);
set(gca,'LineWidth',2);
xlabel('High Gain Sample #','fontsize',14);

```

```

ylabel('Error (nA)', 'fontsize', 14);
legend('Linear coefficients', 'Nonlinear coefficients');
legend('Location', 'SouthWest');
title('High Gain Channel Fit Error', 'fontsize', 14);
set(findall(gca, 'Type', 'Line'), 'LineWidth', 2);
hold off;

% Data parsed in terms of resistors for second analysis
clear LowGain HighGain Temp1 Temp2;
for i=1:length(resistor)
    LowGain(:,i)=256*CalibrationDataFull((CalibrationDataFull(:,2)==resistor(i)),8)+...
        CalibrationDataFull((CalibrationDataFull(:,2)==resistor(i)),9);
    HighGain(:,i)=256*CalibrationDataFull((CalibrationDataFull(:,2)==resistor(i)),6)+...
        CalibrationDataFull((CalibrationDataFull(:,2)==resistor(i)),7);
    Temp1(:,i)=256*CalibrationDataFull((CalibrationDataFull(:,2)==resistor(i)),18)+...
        CalibrationDataFull((CalibrationDataFull(:,2)==resistor(i)),19);
    Temp2(:,i)=256*CalibrationDataFull((CalibrationDataFull(:,2)==resistor(i)),20)+...
        CalibrationDataFull((CalibrationDataFull(:,2)==resistor(i)),21);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ind=any(Temp2==639,2);
LowGain=LowGain(~ind,:);
HighGain=HighGain(~ind,:);
Temp1=Temp1(~ind,:);
Temp2=Temp2(~ind,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Offset=ones(length(LowGain(:,1)),length(resistor));

% Calculate bias and current from bias
CalBias=bias_coeff(1).*Offset+bias_coeff(2).*Temp1+bias_coeff(3).*Temp2;
CalBias2=bias_coeff2(1).*Offset+bias_coeff2(2).*Temp1+bias_coeff2(3).*Temp2+...
    bias_coeff2(4).*Temp1.^2+bias_coeff2(5).*Temp2.^2;
% Current=-CalBias./measured_resistors';
Current=-CalBias2./measured_resistors';

% Calculate current for each channel, linearly
CalCurrent_LG=coeff_LG(1).*Offset(:,:)+coeff_LG(2).*LowGain(:,:)+coeff_LG(3).*Temp1(:,:)+
coeff_LG(4).*Temp2(:,:);
CalCurrent_HG=coeff_HG(1).*Offset(:,:)+coeff_HG(2).*HighGain(:,:)+coeff_HG(3).*Temp1(:,:)+
coeff_HG(4).*Temp2(:,:);

% Calculate current for each channel, nonlinearly
CalCurrent_LG3=coeff_LG3(1).*Offset(:,:)+coeff_LG3(2).*LowGain(:,:)+...
    coeff_LG3(3).*Temp1(:,:)+coeff_LG3(4).*Temp2(:,:)+...
    coeff_LG3(5).*LowGain(:,:).^2+coeff_LG3(6).*Temp1(:,:).^2+...
    coeff_LG3(7).*Temp2(:,:).^2+coeff_LG3(8).*LowGain(:,:).^0.5+...
    coeff_LG3(9).*Temp1(:,:).^3+coeff_LG3(10).*Temp2(:,:).^3;
CalCurrent_HG3=coeff_HG3(1).*Offset(:,:)+coeff_HG3(2).*HighGain(:,:)+...
    coeff_HG3(3).*Temp1(:,:)+coeff_HG3(4).*Temp2(:,:)+...
    coeff_HG3(5).*HighGain(:,:).^2+coeff_HG3(6).*Temp1(:,:).^2+...
    coeff_HG3(7).*Temp2(:,:).^2+coeff_HG3(8).*HighGain(:,:).^0.5+...
    coeff_HG3(9).*Temp1(:,:).^3+coeff_HG3(10).*Temp2(:,:).^3;

% Calculate error from calculated current vs theoretical current
error_LG=Current(:,1:3)-CalCurrent_LG(:,1:3);
error_HG=Current(:,4:6)-CalCurrent_HG(:,4:6);
error_LG3=Current(:,1:3)-CalCurrent_LG3(:,1:3);
error_HG3=Current(:,4:6)-CalCurrent_HG3(:,4:6);

% Plotting low gain error curves per resistor
figure;

```

```

plot(error_LG(:,1)/1e-9,'k');
hold on;
plot(error_LG(:,2)/1e-9,'b');
plot(error_LG(:,3)/1e-9,'r');
plot(error_LG3(:,1)/1e-9,'g');
plot(error_LG3(:,2)/1e-9,'m');
plot(error_LG3(:,3)/1e-9,'c');
ylim([-150 100]);
set(gca,'FontSize',14);
set(gca,'LineWidth',2);
xlabel('Low Gain Sample #','fontsize',14);
ylabel('Error (nA)','fontsize',14);
ylim([-120 120]);
legend('Linear 500k','Linear 1M','Linear 6.04M','Nonlinear 500k','Nonlinear
1M','Nonlinear 6.04M');
legend('Location','SouthWest');
title('Low Gain Fit Error','fontsize',14);
set(findall(gca,'Type','Line'),'LineWidth',2);
hold off;

% Plotting high gain error curves per resistor
figure;
plot(error_HG(:,1)/1e-9,'k');
hold on;
plot(error_HG(:,2)/1e-9,'b');
plot(error_HG(:,3)/1e-9,'r');
plot(error_HG3(:,1)/1e-9,'g');
plot(error_HG3(:,2)/1e-9,'m');
plot(error_HG3(:,3)/1e-9,'c');
ylim([-1.5 1]);
set(gca,'FontSize',14);
set(gca,'LineWidth',2);
xlabel('High Gain Sample #','fontsize',14);
ylabel('Error (nA)','fontsize',14);
legend('Linear 56M','Linear 100M','Linear 1G','Nonlinear 56M','Nonlinear 100M','Nonlinear
1G');
legend('Location','SouthWest');
title('High Gain Fit Error','fontsize',14);
set(findall(gca,'Type','Line'),'LineWidth',2);
hold off;

%%
% Cleaning up
fclose('all');
delete(instrfind);

```

Appendix H

Calibration Data

Bias calibration coefficients in [V] and instrument calibration coefficients in [A].

H.1 0b000 Calibration Coefficients

H.1.1 Bias Calibration

K_{offset}	K_{T_1}	K_{T_2}	$K_{T_1^2}$	$K_{T_2^2}$
-7.0061	-3.3328×10^{-5}	2.6540×10^{-5}	2.1696×10^{-8}	-1.9172×10^{-8}

H.1.2 Instrument Calibration

Low Gain

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
6.7258×10^{-6}	1.0716×10^{-9}	-6.0037×10^{-7}	5.9293×10^{-7}	-5.9905×10^{-15}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
7.6488×10^{-10}	-7.5378×10^{-10}	-1.2887×10^{-7}	-3.2297×10^{-13}	3.1758×10^{-13}

High Gain

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
7.0384×10^{-8}	6.6857×10^{-12}	6.1039×10^{-9}	-6.2720×10^{-9}	-3.1684×10^{-17}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
-7.7632×10^{-12}	7.9407×10^{-12}	-5.5728×10^{-10}	3.2758×10^{-15}	-3.3366×10^{-15}

H.2 0b001 Calibration Coefficients**H.2.1 Bias Calibration**

K_{offset}	K_{T_1}	K_{T_2}	$K_{T_1^2}$	$K_{T_2^2}$
-7.0064	-5.9775×10^{-6}	-3.7608×10^{-7}	2.9097×10^{-9}	1.9377×10^{-10}

H.2.2 Instrument Calibration**Low Gain**

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
5.3719×10^{-6}	8.7859×10^{-10}	1.2495×10^{-7}	-1.3186×10^{-7}	-4.4975×10^{-15}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
-1.4467×10^{-10}	1.5441×10^{-10}	-9.6042×10^{-8}	5.4862×10^{-14}	-5.9335×10^{-14}

High Gain

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
-1.3797×10^{-8}	8.3312×10^{-12}	7.5675×10^{-10}	-6.2704×10^{-10}	-4.6093×10^{-17}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
-8.8102×10^{-13}	7.4516×10^{-13}	-8.1390×10^{-10}	3.3809×10^{-16}	-2.9098×10^{-16}

H.3 0b010 Calibration Coefficients**H.3.1 Bias Calibration**

K_{offset}	K_{T_1}	K_{T_2}	$K_{T_1^2}$	$K_{T_2^2}$
-7.0067	-1.7585×10^{-5}	2.6867×10^{-6}	7.9414×10^{-9}	-2.0826×10^{-9}

H.3.2 Instrument Calibration**Low Gain**

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
9.4784×10^{-6}	4.7912×10^{-10}	-7.1214×10^{-8}	3.9008×10^{-8}	-1.3637×10^{-15}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
8.1876×10^{-11}	-4.0856×10^{-11}	-2.8625×10^{-8}	-2.9796×10^{-14}	1.2511×10^{-14}

High Gain

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
2.8280×10^{-8}	6.9844×10^{-12}	2.8489×10^{-10}	-3.3264×10^{-10}	-3.5132×10^{-17}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
-3.4518×10^{-13}	4.1734×10^{-13}	-5.8872×10^{-10}	1.3690×10^{-16}	-1.6994×10^{-16}

H.4 0b011 Calibration Coefficients**H.4.1 Bias Calibration**

K_{offset}	K_{T_1}	K_{T_2}	$K_{T_1^2}$	$K_{T_2^2}$
-6.9996	-2.0500×10^{-5}	9.1933×10^{-7}	9.7842×10^{-9}	-7.2182×10^{-10}

H.4.2 Instrument Calibration**Low Gain**

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
5.9662×10^{-6}	1.0671×10^{-9}	-6.5188×10^{-8}	6.1607×10^{-8}	-5.9675×10^{-15}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
8.1725×10^{-11}	-7.7000×10^{-11}	-1.2793×10^{-7}	-3.4091×10^{-14}	3.2029×10^{-14}

High Gain

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
5.9989×10^{-8}	5.7841×10^{-12}	1.3511×10^{-9}	-1.5074×10^{-9}	-2.4457×10^{-17}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
-1.6554×10^{-12}	1.8334×10^{-12}	-4.0358×10^{-10}	6.6597×10^{-16}	-7.3480×10^{-16}

H.5 0b100 Calibration Coefficients**H.5.1 Bias Calibration**

K_{offset}	K_{T_1}	K_{T_2}	$K_{T_1^2}$	$K_{T_2^2}$
-6.9970	-2.6626×10^{-5}	-3.9571×10^{-7}	1.3352×10^{-8}	3.2886×10^{-10}

H.5.2 Instrument Calibration**Low Gain**

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
5.7346×10^{-6}	1.1015×10^{-9}	1.9649×10^{-9}	-3.4946×10^{-9}	-6.2189×10^{-15}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
-2.8454×10^{-12}	4.6925×10^{-12}	-1.3399×10^{-7}	1.2248×10^{-15}	-1.9602×10^{-15}

High Gain

K_{offset}	K_{ch}	K_{T_1}	K_{T_2}	$K_{C_{ch}^2}$
7.7478×10^{-8}	7.1709×10^{-12}	-2.3230×10^{-9}	2.1097×10^{-9}	-3.4912×10^{-17}
$K_{T_1^2}$	$K_{T_2^2}$	$K_{C_{ch}^{0.5}}$	$K_{T_1^3}$	$K_{T_2^3}$
2.8264×10^{-12}	-2.5536×10^{-12}	-6.4897×10^{-10}	-1.1178×10^{-15}	1.0014×10^{-15}