



# An Interactive Application Demo for the AD8295

A Major Qualifying Project Report submitted to the Faculty of the

Worcester Polytechnic Institute

In partial fulfillment of the requirements for the Degree of Bachelor of Science

By

---

Alihusain Y. Sirohiwala

---

Vasil K. Savov

---

Tahsin Hassan

Submitted to:

---

Dr. John McNeill, Advisor

## Abstract

The goal of this MQP was to design an interactive Application Demo Board for the Instrumentation Amplifier AD8295 that could be used in electronics trade-shows worldwide to demonstrate the versatility of this product. Instrumentation Amplifiers (In-Amps) are ideal for extracting and processing small differential signals from high common-mode voltages. Traditionally, these amplifiers have been used for applications in the medical and automotive industries. The AD8295 from Analog Devices is their latest in-amp product and comprises of a precision In-Amp, 2 uncommitted Op-Amps and a matched resistor pair all on the same substrate. This provides immense versatility and allows the product to be designed into a myriad of applications as a space saving, precise, and low power alternative.

## Table of Contents

Abstract .....	1
Executive Summary .....	9
Acknowledgments .....	12
Introduction.....	13
1. Background.....	15
1.1. Defining Project Goals .....	15
1.2. Instrumentation Amplifiers .....	16
1.3. The AD8295 – A System Level Solution .....	26
2. Brainstorming Applications .....	33
Application #1: Electrocardiogram.....	35
Application #2: Strain-Gage using AC excitation.....	37
Application #3: RTD Temperature Sensor .....	39
Application #4: Single Supply Barometer .....	41
Application #5: Voltage Controller Current Source.....	42
Application #6: Glove Mouse .....	44
Application #7: Pulse Oximetry .....	46
Application #8: Capacitive Sensing.....	47
Application #9: Temperature Based Dual-Axis Tilt-Meter .....	49
Discussion on Application Choice.....	50
3. Design Details .....	51
3.1. Overall Design .....	51
3.1.1. Sensor.....	51
3.1.2. Analog Multiplexer: The ADG659.....	52
3.1.3. The AD8295 .....	52
3.1.4. The MSP430F169.....	53
3.1.5. USB Chipset FT232 .....	54
3.1.6. Demo Laptop .....	54
3.1.7. Charge Pump .....	54
3.2. System Block Diagram Details .....	55
3.2.1. Sensor.....	55
3.2.2. Analog Multiplexer.....	61
3.2.3. Instrumentation Amplifier - AD8295.....	64
3.2.4. $\mu$ Controller - MSP430F169 .....	66
3.2.5. USB chipset – FT232RL .....	73
3.2.6. Charge Pump – Microchip TC7660 .....	74
3.2.7. Frequency Compensation .....	79
3.2.8. Demo Laptop and LabView GUI .....	88
Game rules .....	98
4. Measurements and Results .....	102
4.1 Sensor Measurements .....	102
4.1.1 Noise Characterization .....	103
4.1.3 Step Response .....	105
4.1.3 Sensor Angle Characterization .....	106
4.2 Sampling Scheme Evaluation .....	110
4.3 Calibration Scheme Evaluation.....	112
5. Conclusion .....	113
APPENDIX A – Detailed final Schematic .....	116

APPENDIX B – LabView Game .....	117
Setup for Maze.VI.....	117
APPENDIX C – Scope Reading.....	118
APPENDIX D – Tutorial for scope Reading.....	119
APPENDIX E – PCB Layout.....	130
APPENDIX F – MATLAB Script code for the filter design .....	133
APPENDIX G – Matlab Scripts for Noise .....	137
APPENDIX H – Measured Performance Characteristics of AD8295 .....	138
APPENDIX I – MSP430F169 Code .....	144
APPENDIX J – Part List .....	148
APPENDIX K – LabView Scope Reading Diagram.....	149
APPENDIX L – LabView Maze Game Diagram .....	150

## Figures

FIGURE 1: GENERAL INSTRUMENTATION AMPLIFIER.....	16
FIGURE 2: THREE OP AMP INSTRUMENTATION AMPLIFIER ARCHITECTURE.....	17
FIGURE 3 : TWO OP AMP INSTRUMENTATION AMPLIFIER ARCHITECTURE .....	19
FIGURE 4: MODELING INPUT BIAS CURRENTS.....	22
FIGURE 5: <i>INPUT COMMON-MODE RANGE VS. OUTPUT VOLTAGE, G = 1, VS = ±2.5 V, ±5 V, REF = 0 V</i> .....	24
FIGURE 6: BLOCK DIAGRAM OF AD8295 (LEFT) LFCSP PACKAGE DIMENSIONS IN MM (RIGHT) .....	26
FIGURE 7: AD8295 EVALUATION BOARD.....	29
FIGURE 8: DC GAIN TEST SCHEMATIC.....	30
FIGURE 9: $V_{IN}$ VS $V_{OUT}$ , $G = 10$ .....	30
FIGURE 10: BANDWIDTH MEASUREMENT SCHEMATIC.....	32
FIGURE 11: BANDWIDTH OVERLAY.....	32
FIGURE 12: PATIENT SIDE OF THE ECG SIGNAL-CHAIN.....	35
FIGURE 13: STRAIN GAGE APPLICATION (USING AC EXCITATION) .....	37
FIGURE 14: PRECISION SINGLE SUPPLY RTD THERMOMETER .....	39
FIGURE 15: SINGLE SUPPLY BAROMETER .....	41
FIGURE 16: VOLTAGE CONTROLLED CURRENT .....	42
FIGURE 17: X-IST DATA GLOVE USING FIBER OPTIC BEND SENSORS.....	44
FIGURE 18: PULSE OXIMETRY SETUP (LEFT) ABSORPTION SPECTRA FOR OXY/DEOXY-HEMOGLOBIN(RIGHT) .....	46
FIGURE 19: PROPOSED TOUCH-SCREEN (TOP), CIRCUIT CONFIGURATION (BOTTOM).....	47
FIGURE 20: DUAL AXIS ACCELEROMETER CONCEPT.....	49
FIGURE 21: TILT-SENSING APPLICATION.....	50
FIGURE 22: SYSTEM BLOCK DIAGRAM.....	51
FIGURE 23: SYSTEM LEVEL EXPLANATION OF SENSOR BLOCK .....	52
FIGURE 24: ANALOG MUX – SIMPLIFIED SCHEMATIC .....	52
FIGURE 25: AD8295 BASIC CIRCUIT .....	53
FIGURE 26: MSP430F169 – SIMPLIFIED SCHEMATIC .....	53
FIGURE 27: FT232 USB CHIPSET – SIMPLIFIED SCHEMATIC.....	54
FIGURE 28: CHARGE PUMP – SIMPLIFIED SCHEMATIC.....	54
FIGURE 29: A SAMPLE MEMSIC ACCELEROMETER PACKAGE (LEFT) THE MEMSIC 2-AXIS SENSOR (RIGHT).....	55
FIGURE 30: SENSOR VERSION1 (LEFT) SENSOR WHEATSTONE BRIDGE SETUP (RIGHT).....	56
FIGURE 31: NOISE MEASUREMENT - SENSOR VERSION1 .....	57
FIGURE 32: SETUP FOR STEP RESPONSE TEST .....	57
FIGURE 33: SENSORV1 +90 STEP (LEFT) SENSORV1 -90 STEP (RIGHT).....	58
FIGURE 34: SENSOR VERSION 2 .....	58
FIGURE 35: SENSOR V2 NOISE (LEFT) SENSOR V2 STEP RESPONSE +90(RIGHT).....	59
FIGURE 36: SENSOR VERSION 3 .....	59
FIGURE 37: FINAL SENSOR DESIGN SIDE VIEW (LEFT) AND TOP VIEW (RIGHT).....	60
FIGURE 38: FINAL SENSOR NOISE MEASUREMENT .....	61
FIGURE 39: TWO CHANNEL INPUT USING MULTIPLEXER .....	62
FIGURE 40: TWO CHANNELS INPUT USING TWO SEPARATE IN-AMPS.....	62
FIGURE 41: AD8295 CIRCUIT SCHEMATIC.....	64
FIGURE 42: PCB LAYOUT OF AD8295.....	65
FIGURE 43: MSP430X16X FUNCTIONAL BLOCK DIAGRAM .....	66
FIGURE 44: ADC12 BLOCK DIAGRAM.....	68
FIGURE 45: USART BLOCK DIAGRAM UART MODE .....	69
FIGURE 46: OUTPUT RESISTANCE OF THE DIGITAL TO ANALOG CONVERTER .....	70
FIGURE 47: DAC12 BLOCK DIAGRAM.....	71
FIGURE 48: FT232R BLOCK DIAGRAM.....	73
FIGURE 49: TC7660 CHARGE PUMP (LEFT) CONFIGURATION FOR A -5V OUTPUT (RIGHT) .....	75
FIGURE 50: OUTPUT CURRENT VS. OUTPUT VOLTAGE.....	76
FIGURE 51: EQUIVALENT CIRCUIT MODEL OF TC7660 .....	77

FIGURE 52: PSRR VS. FREQUENCY FOR AD8295 .....	78
FIGURE 53: ACQUIRED DATA SHOWING SLOW DYNAMIC RESPONSE.....	79
FIGURE 54: ORDER OF FILTERING .....	80
FIGURE 55: AVERAGING FILTER RESULT .....	81
FIGURE 56: COMPARISON BETWEEN THEORETICAL MODEL & ACQUIRED DATA .....	82
FIGURE 57: SYSTEM RESPONSES.....	84
FIGURE 58: STEP RESPONSE OF SYSTEMS IN CONTINUOUS TIME.....	84
FIGURE 59: STEP RESPONSE OF THE CONTINUOUS SYSTEM .....	85
FIGURE 60: Z PLANE REPRESENTATION OF THE INITIAL SYSTEM .....	86
FIGURE 61: Z PLANE REPRESENTATION OF THE FILTER SYSTEM .....	86
FIGURE 62: Z PLANE REPRESENTATION OF THE OVERALL FILTERED SYSTEM .....	87
FIGURE 63: VISA CONFIGURE            FIGURE 64: FRONT PANEL VISA RESOURCE NAME .....	89
FIGURE 65: VISA READ .....	89
FIGURE 66: ELAPSED TIME BLOCK      FIGURE 67: VISA CLEAR .....	90
FIGURE 68: WHILE LOOP.....	90
FIGURE 69: SEARCH/SPLIT STRING .....	90
FIGURE 70: VOLTAGE SUBVI .....	91
FIGURE 71: VOLTAGE SUBVI BLOCK DIAGRAM.....	91
FIGURE 72: STRING TO NUMBER.....	92
FIGURE 73: DIVIDE FUNCTION      FIGURE 74: MULTIPLY FUNCTION.....	92
FIGURE 75: INSERT INTO ARRAY (LEFT) EMPTY ARRAY (RIGHT) .....	92
FIGURE 76: ANGLE SUBVI.....	93
FIGURE 77: ANGLE SUBVI BLOCK DIAGRAM .....	93
FIGURE 78: AVERAGING FILTER      FIGURE 79: AVERAGING FILTER SUBVI.....	95
FIGURE 80: IIR FILTER BLOCK DIAGRAM .....	95
FIGURE 81: IIR HIGHPASS FILTER SUBVI.....	96
FIGURE 82: VOLTAGE MEASUREMENT GRAPH.....	96
FIGURE 83: ANGLE MEASUREMENT .....	96
FIGURE 84: SUB-BLOCK DIAGRAM FOR CREATING A 3D BOARD.....	97
FIGURE 85: SET TRANSLATION BLOCK .....	98
FIGURE 86: BLOCK DIAGRAM FOR THE IMPLEMENTATION OF GAME RULES .....	98
FIGURE 87: THE X AXIS UPPER BOUND ARRAY .....	99
FIGURE 88: X WITHIN WALL SUBVI .....	99
FIGURE 89: CHECK WHETHER THE BALL MOVEMENT IS LEGAL.....	100
FIGURE 90: OUR FINAL SENSOR ASSEMBLY.....	102
FIGURE 91: SENSOR VERSION3 X-AXIS NOISE (TOP) AND Y-AXIS NOISE (BOTTOM).....	103
FIGURE 92: SENSOR VERSION3 - Y-AXIS NOISE SPECTRUM (TOP) AND X-AXIS NOISE SPECTRUM (BOTTOM).....	104
FIGURE 93: FIGURE ILLUSTRATING 90 DEGREE STEP .....	105
FIGURE 94: STEP RESPONSE, X AXIS, +90 STEP .....	105
FIGURE 95: STEP RESPONSE, X AXIS, -90 STEP .....	106
FIGURE 96: ILLUSTRATING ANGLE CHARACTERIZATION PROCEDURE.....	107
FIGURE 97: Y-AXIS ANGULAR RESPONSE (TOP), Y-AXIS ANGULAR RESPONSE ZOOMED-IN (BOTTOM).....	108
FIGURE 98: X-AXIS ANGULAR RESPONSE .....	109
FIGURE 99: SAMPLING THE MULTIPLEXED SIGNAL .....	110
FIGURE 100: ZOOMED SAMPLING THE MULTIPLEXED SIGNAL.....	111
FIGURE 101: OUTPUT BEFORE (LEFT) AND AFTER (RIGHT) CALIBRATION .....	112
FIGURE 102: FRONT PANEL OF MAZE.VI.....	117
FIGURE 103: FRONT PANEL OF SCOPE_READING.VI.....	118
FIGURE 104: SMALL PORTION OF SCOPE_READING.VI FRONT PANEL .....	119
FIGURE 105: SMALL PORTION OF SCOPE_READING.VI BLOCK DIAGRAM .....	120
FIGURE 106: LAYOUT REVISION 1.0.....	130
FIGURE 107: TOP LAYER OF PCB REV 2.0.....	130
FIGURE 108: BOTTOM LAYER OF PCB REV 2.0.....	131

FIGURE 109: LAYOUT REVISION 2.0 ..... 131  
FIGURE 110: TOP LAYER OF PCB REV 2.0 ..... 132  
FIGURE 111: BOTTOM LAYER OF PCB REV 2.0 ..... 132  
FIGURE 112: VIN VS VOUT G=1 ..... 138  
FIGURE 113: VIN VS VOUT G=100 ..... 139  
FIGURE 114: VIN VS VOUT G=1000 ..... 140  
FIGURE 115: BANDWIDTH ..... 141  
FIGURE 116 BANDWIDTH MEASUREMENT SCHEMATIC ..... 141

## Tables

TABLE 1: AD8295 CMR CHARACTERISTICS.....	20
TABLE 2: AD8295 OFFSET CHARACTERISTICS .....	21
TABLE 3: AD8295 INPUT IMPEDANCE CHARACTERISTICS.....	21
TABLE 4: AD8295 INPUT CURRENT CHARACTERISTICS .....	22
TABLE 5: AD8295 NOISE CHARACTERISTICS .....	23
TABLE 6: TYPICAL SMALL SIGNAL BANDWIDTHS .....	24
TABLE 7: GAIN OF 10 EVALUATIONS:.....	30
TABLE 8: BANDWIDTH DATA - G=100 .....	31
TABLE 9: SUGGESTED ADI PRODUCTS FOR APPLICATION RESEARCH.....	33
TABLE 10: ECG SIGNAL PROFILE.....	35
TABLE 11: METRIC FOR PICKING APPLICATION .....	50
TABLE 12: COMPARING SOLUTIONS FOR MULTIPLE INPUTS .....	63
TABLE 13: ADC REGISTERS CONFIGURATION .....	69
TABLE 14: UART REGISTERS CONFIGURATION .....	70
TABLE 15: DAC REGISTERS CONFIGURATION .....	71
TABLE 16: INPUT OPERATING VOLTAGE RANGE FOR AD8295 .....	74
TABLE 17: GAIN OF 1 EVALUATION .....	138
TABLE 18: GAIN OF 100 EVALUATION .....	139
TABLE 19: GAIN OF 1000 EVALUATION .....	140
TABLE 20: G=1, BANDWIDTH MEASUREMENT .....	142
TABLE: 21 G=100 BANDWIDTH MEASUREMENT .....	142
TABLE 22: G=1000 BANDWIDTH MEASUREMENT .....	143

## Equations

EQUATION 1: GENERAL TRANSFER CHARACTERISTIC FOR THREE OP-AMP IN-AMP .....	17
EQUATION 2: TYPICAL TRANSFER CHARACTERISTIC FOR THREE OP-AMP IN-AMP .....	18
EQUATION 3: GENERAL TRANSFER CHARACTERISTIC FOR TWO OP-AMP IN-AMP .....	19
EQUATION 4: TYPICAL TRANSFER CHARACTERISTIC FOR TWO OP-AMP IN-AMP .....	19
EQUATION 5: TRANSFER CHARACTERISTIC FOR THE IN-AMP OF AD8295 .....	27
EQUATION 6: CALCULATING GAIN OF THE IN-AMP OF AD8295.....	27
EQUATION 7: OUTPUT $N_{ADC}$ OF THE ADC.....	68
EQUATION 8: OUTPUT OF THE DAC.....	71
EQUATION 9: BANDWIDTH – RISE TIME RELATIONSHIP.....	79
EQUATION 10: TIME CONSTANT – RISE TIME RELATIONSHIP .....	82
EQUATION 11: TRANSFER FUNCTION OF SYSTEM.....	82
EQUATION 12: Z TRANSFORM EXPRESSION OF HIGH PASS FILTER .....	85
EQUATION 13: IIR HIGH PASS FILTER EQUATION .....	85
EQUATION 14: RUNNING AVERAGE EQUATION.....	94

## Executive Summary

The purpose of this project was to develop an interactive application demo for an Analog Devices Inc (ADI) product. Analog Devices Inc. is a semiconductor company that has been a leading manufacturer of analog and mixed-signal, signal processing products for over forty years. As a member of the NECAMSID lab at Worcester Polytechnic Institute, ADI expressed interest in having an application-demo for one of its latest instrumentation amplifier products so that it could be used at electronics tradeshows to demonstrate the key features of the amplifier in an interactive manner.

An initial meeting with Oljeta Bida, Paul Blanchard (WPI alumnus and employees of Analog Devices Inc) and our advisor Prof. John McNeill lead to the establishment of general guidelines for the nature of the application demo to be designed. The AD8295, a new ADI instrumentation amplifier product was suggested as a candidate for the demo. This product was suggested for its versatility as it contained an instrumentation amplifier, two operational amplifiers and a matched resistor pair all on a single silicon die. Furthermore, it was requested that the demo be powered by and connected to a laptop computer, via USB, so as to make the setup requirements minimal. This would be a great asset for an applications engineer at ADI who would use this demo for electronics tradeshows worldwide.

Based on this general guideline, we began researching for applications that a demo could be made for. We started our search by first looking for already-in-use applications for instrumentation amplifier products. These included an Electrocardiogram demo, a Strain-Gage application, a Temperature Sensor with Linearization Circuit, and a Capacitive Sensing Application among several others. These ideas were then screened based on our project guidelines, constraints, and interactive appeal.

Given the sheer range of options available, we decided to spend significant time towards developing some of these ideas and thereafter picking the one that was most feasible. At the same time, we decided to work, in parallel, on the design and construction of a prototype printed circuit board (PCB) that would include core components of our signal chain (the AD8295, a microcontroller, USB solution, and the power solution) so as to iron out this process early in the

project, giving us more time to work on the research and development of our final application. In implementing this plan, we further pursued the following ideas: the RTD Temperature Linearization Circuit, the Capacitive Sensing Application, and a Temperature-Based Tilt-Meter.

Based on this research, we determined that the Capacitive Sensing Application would not be feasible given the scope of this project and the product we had chosen. The RTD Temperature Linearization Circuit would work well with the part selected, but it wasn't a very interactive application. Lastly, the Temperature-Based Tilt Meter provided promising results after initial prototyping and had the potential for being used as a dual-axis motion detector. This seemed a feasible and appealing choice as it had all the requirements we were looking for. Therefore, we decided to make a temperature-based tilt-meter for our Application Demo.

The temperature-based tilt-meter was based on an idea for an accelerometer currently being produced by a company called MEMSIC. This idea involved constructing a square heat source surrounded by thermistors on each side and enclosed in an air-tight cavity. When tilted, the temperature difference between opposite thermistor pairs would change, due to heat transfer by convection, as some function of tilt angle. This difference could be tracked with an instrumentation amplifier, and used as spatial co-ordinates to govern the motion of a "ball" through a maze on the laptop screen. This would make for an interactive game as our application. In addition, we would also provide a "scope-mode" that would display the acquired differential signals "real-time" for use at trade show demos.

We were now in a position to make design choices more specifically. We chose LabView as the software for the laptop portion of the design and work began on developing the game and scope-mode programs. We went through several iterations in the sensor design to improve the noise and bandwidth of our system. Our final design involved thru-hole resistors, thru-hole thermistors and a plexi-glass casing for an air-tight seal. Initial measurement results led to the realization that additional filtering would be required to sufficiently increase the bandwidth of the system. This filtering was implemented in LabView using a discrete high-pass filter.

For the AD8295 portion of our system design, we decided to multiplex two inputs (containing X, Y-axis information) to the instrumentation amplifier using an analog multiplexer. Additionally, we found uses for each of the other components in the product for our application. For the digital portion of our system design, we chose a Texas Instruments MSP430 microcontroller and interfaced it with a USB Chipset to provide serial data to the computer. All of these are discussed in the Design Details section of this report.

Our second and final PCB was then designed, incorporating our final design choices. It was significantly smaller (13cm X 7cm compared to 15cm X 10cm for the original one) in size as it contained only the features/components required by our application. Our final “Game” involved navigating a ball through a two dimensional maze, using the PCB as a dual-axis, tilt-based, input-device. We now conducted extensive testing with this board to verify the functionality of the system and characterize some of the specifications of the sensor such as noise, step response, and angular response. This verification showed that the system was functional and performed as expected. The game worked well although there was still about a second of ‘lag’ in the response of the system to angular motion. This was mostly attributed to the thermal time-constant inherent in the sensor’s design. This board, along with the necessary design details and schematics will be presented to ADI in completion of the project.

## Acknowledgments

This Project would not have been possible without the support of the guidance of many.

Firstly, we would like to thank the New England Center for Analog and Mixed Signal Integrated Circuit Design (NECAMSID) for sponsoring this project, specifically the companies Analog Devices in Wilmington MA, Allegro Microsystems in Worcester MA, BAE Systems in Nashua and Merrimack NH and Texas Instruments in Manchester NH, for providing the financial support necessary for carrying out this project.

Secondly, we would like to thank Paul Blanchard, Oljeta Bida and Michael O’Sullivan from Analog Devices, Wilmington for their constant support in all stages of this project. Their help with identifying and developing this project opportunity was instrumental to its success.

Thirdly, we would like to thank Patrick Morrison, Tom Angelotti and Bob Boyce for their help with acquiring parts and soldering IC’s for our prototypes.

Finally, we would like to extend our deepest gratitude to our project advisor, Professor John McNeill for his tutelage. He set a high standard of excellence for us from the very outset, and his mentorship motivated us to try and reach this standard.

## Introduction

This project explores the topic of Integrated Circuit(IC) Applications. It is aimed at trying to build an application circuit for an existing IC. Specifically, it investigates suitable applications for the AD8295 which is one of the latest Instrumentation Amplifier (In-Amp) IC's designed and manufactured by Analog Devices Inc. This product is a system-on-chip solution comprising of an Instrumentation Amplifier, two uncommitted op-amps and a laser trimmed resistor pair all on the same silicon substrate.

In-Amps can be viewed as closed loop voltage gain blocks usually designed for precise DC/low-frequency performance and are used in a variety of applications. Traditionally, In-Amps have been applied in the Medical (ECG, General Instrumentation) and Automotive Industries. However, any low frequency signal that needs to be extracted from high common-mode, noisy environment is a good candidate for being processed with an In-Amp.

This project was defined and proposed in collaboration with the Industrial Amplifier Products Group at Analog Devices. Former WPI graduates Paul Blanchard and Oljeta Bida served as contact persons from ADI. As such, it was a joint effort between ADI and WPI.

The overlying guideline assigned to the team was to develop a user-interactive application demo board that could be used to highlight the key advantages of the AD8295 over competing discrete solutions. This application demo would ideally be something that an Applications Engineer at ADI could use at Trade Shows worldwide as a tool to demonstrate the advantages of designing application circuits around the AD8295.

After going through the process of brainstorming a set of potential application demos based on the project timeline, resource constraints, and the key features of the product; we decided to implement a computer input device as our application. This device would take advantage of the AD8295's ability to process small differential voltages from a heat convection based tilt sensor to obtain X and Y spatial coordinates on a screen corresponding to tilt along the

respective axes of the sensor. Additionally, our application would have the functionality of being able to test the basic performance characteristic of the part to serve as a quick demonstration of its quality.

This report discusses the details of this project through all the major steps: conception, project definition, design, and evaluation.

## 1. Background

This project began as an open-ended topic: Designing an Application for an Instrumentation Amplifier Integrated Circuit. Through our meetings with representatives at Analog Devices, and our own research, we developed some guidelines and goals as to what our features and qualities our Application Demo should possess. These guidelines are discussed below. Furthermore, this chapter provides a basic technical overview of Instrumentation Amplifiers (topology, important specs, etc.) in general and the AD8295 in particular.

### 1.1. Defining Project Goals

Choosing one specific IC, from a variety of possibilities, helped us derive some of the application requirements. Most of these requirements emerged after several meetings within the team, with our advisor - Professor John McNeill, and with representatives from Analog devices - Oljeta Bida (designer of the part) and Paul Blanchard (product engineer for the part). It was stated in these meetings that the final goal was to have a demo board that could be used for showing the part's performance at different tradeshow – more specifically Electronica™ (Germany) and SensorsExpo™ (USA). Another guideline given to us was to make the application as interactive as possible so that it would appeal to people attending these trade-shows. On the design front this lead to incorporating a USB connection to a computer for data transfer and power supply in order to make the board as easy-to-use as possible. To be able to establish the data transfer to the computer we had to design a data acquisition system that will sample the output of the analog stage, process it and send it to the computer according to the USB protocol. Given that the AD8295 had multiple available features, our goal was to use as many of these for our application, as possible.

Following the goals and initial requirements, we started doing our background research on instrumentation amplifiers in general and the AD8295 in particular.

## 1.2. Instrumentation Amplifiers

### General Information

An Instrumentation amplifier is a closed loop gain block that has a differential input and an output that is single ended with respect to a reference terminal. It falls under the category of 'Difference Amplifiers', i.e., they amplify the voltage difference between their inputs and, ideally, they reject any voltages that are common to both inputs.

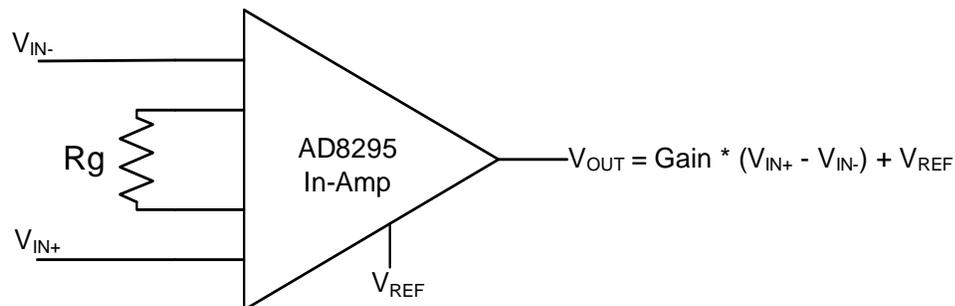


Figure 1: General Instrumentation Amplifier

Commonly the impedance of both the input terminals are balanced and have high values (usually in the  $G\Omega$  range or higher). The gain is determined by a feedback resistor network that is isolated from the signal input terminals. The gain resistor can be internal to the IC or it can be user set by an external resistor connecting two pins as shown on figure 1. Semiconductor manufacturers have made many advances since the first monolithic Integrated Circuits were fabricated. These include laser trimming of on-chip resistors to account for process variations and allow precise matching and accurate performance. These advances have led to a new generation of Instrumentation Amplifiers with enhanced common mode rejection (80dB-100dB).

### Different Architectures

Looking a little more in depth of the operation of the instrumentation amplifiers we can distinguish two main architectures – a two op-amp and a three op-amp topology.

### 3-Op Amp Instrumentation Amplifier

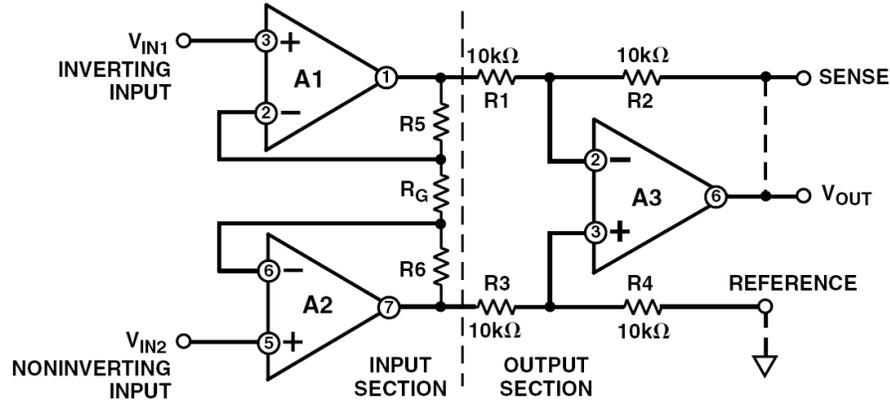


Figure 2: Three Op Amp Instrumentation Amplifier Architecture<sup>1</sup>

Figure 2 shows the architecture that has become the most popular configuration for instrumentation amplifier design. The classic 3-op amp in-amp circuit is a solution that has one op amp for the output that is subtracting the two inputs that are buffered from another two op-amps. Op amps A1 and A2 of figure 2 buffer the input voltage. In this configuration, a single gain resistor,  $R_G$ , is connected between the summing junctions of A1 and A2. The full differential input voltage will appear across  $R_G$  (because the voltage at the summing junction of each amplifier is equal to the voltage applied to its positive input). Since the amplified input voltage (at the outputs of A1 and A2) appears differentially across the three resistors,  $R_5$ ,  $R_G$ , and  $R_6$ , the differential gain may be varied by just changing  $R_G$ .

Another major advantage of this architecture is that once the subtractor circuit has been set up with its ratio-matched resistors, no further resistor matching is required when changing gains. The general transfer characteristic looks like this:

Equation 1: General Transfer Characteristic for three op-amp in-amp

$$V_{OUT} = \frac{1}{R_G} * V_{IN} \left[ \frac{A * R_6}{B} + R_5 \left( \frac{1}{B} - 1 \right) \right] + \left( \frac{A}{B} \right) * V_{IN}^+ - \left( \frac{1}{B} - 1 \right) * V_{IN}^- + V_{REF} \quad (1)$$

<sup>1</sup> A Designer's Guide to Instrumentation Amplifiers (3rd Edition) by Charles Kitchin and Lew Counts

Where:

$$V_{IN} = V_{IN}^+ - V_{IN}^-; \quad A = \frac{R_4}{R_3 + R_4}; \quad B = \frac{R_1}{R_2 + R_1};$$

We can see how that any imbalance in the resistor matching can lead to an asymmetry (the red terms) in the output. The gain resistor is affecting only the differential input. The standard ratio between the resistor is to make  $R_5 = R_6$ ,  $R_1 = R_3 = R_2 = R_4$  and we can see that when we plug this in the equation1 all the asymmetric terms are canceled and the transfer characteristic becomes:

**Equation 2: Typical Transfer Characteristic for three op-amp in-amp**

$$V_{OUT} = V_{IN} * \left(1 + \frac{2 * R_5}{R_G}\right) + V_{REF} \quad (2)$$

Since the voltage across  $R_G$  equals  $V_{IN}$ , the current through  $R_G$  will equal  $\frac{V_{IN}}{R_G}$ . Amplifiers A1 and A2, therefore, will operate with gain and amplify the input signal. Note, however, that if a common-mode voltage is applied to the amplifier inputs, the voltages on each side of  $R_G$  will be equal, and no current will flow through this resistor. No current flow through  $R_G$  will result in no current flow through  $R_5$  and  $R_6$  as well and amplifiers A1 and A2 will operate as unity-gain followers. Therefore, common mode signals will be passed through the input buffers at unity gain, but differential voltages will be amplified by the factor  $(1 + (2 R_F/R_G))$ . This means that the user may take as much gain in the front end as desired (as determined by  $R_G$ ) without increasing the common-mode gain and error. In fact the CMRR will theoretically increase in direct proportion to gain, which is a very useful property. Finally, because of the symmetry of this configuration, common-mode errors in the input amplifiers, if they track, tend to be canceled out by the output stage subtractor. These features explain the popularity of this configuration.

## 2-Op Amp Instrumentation Amplifier

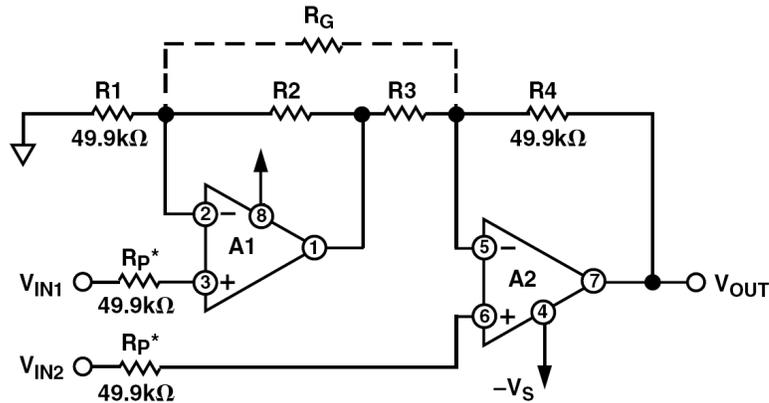


Figure 3 : Two Op Amp Instrumentation Amplifier Architecture

Another very common architecture is the two op-amp as shown on figure 3. It has the obvious advantage of requiring only two, rather than three, operational amplifiers and providing savings in cost and power consumption. However, the nonsymmetrical topology of the 2-op amp in-amp circuit can lead to several disadvantages, most notably lower AC CMRR compared to the 3-op amp design, limiting the circuit's usefulness. For this architecture the precise resistor matching also has a big influence in identical way. The transfer characteristic is:

Equation 3: General Transfer Characteristic for two op-amp in-amp

$$V_{OUT} = V_{IN} * \left( \frac{R_4}{R_3} + \frac{R_4}{R_g} + \frac{R_2 R_4}{R_g R_3} \right) + V_{IN}^+ - \frac{R_2 R_4}{R_1 R_3} V_{IN}^- \quad (3)$$

In order to get rid of the asymmetric term (Equation 3 in red) the ratios are matched such that  $R_1 = R_4$  and  $R_2 = R_3$  and the transfer characteristics is:

Equation 4: Typical Transfer Characteristic for two op-amp in-amp

$$V_{OUT} = V_{IN} * \left( 1 + \frac{R_4}{R_3} + 2 \frac{R_4}{R_g} \right) \quad (4)$$

## *In-Amp Properties*

In order to get a better understanding of instrumentation amplifiers we looked at the characteristics that can be used to compare In-Amps. Understanding this was important to our project in terms of brainstorming of application that can emphasize the performance of AD8295. For each specification we are showing the part from the datasheet of AD8295 In-Amp that describes it. It is really important that the results are only under certain condition. In this case:

$$V_S = \pm 15 \text{ V}, V_{\text{REF}} = 0 \text{ V}, T_A = 25^\circ\text{C}, G = 1, R_L = 2 \text{ k}\Omega$$

## *AC/DC Common Mode Rejection (CMR)*

Ideally an in-amp should amplify only the difference between the two inputs, regardless if there is a voltage that is common to both inputs, both AC and DC. At a minimum the Common Mode Rejection (**CMR**), should be high over the range of input frequencies that need to be rejected. A good CMR will be high at the frequencies of the power lines.

**Table 1: AD8295 CMR Characteristics**

Parameter	Test Conditions	A Grade			B Grade			Unit
		Min	Typ	Max	Min	Typ	Max	
COMMON-MODE REJECTION RATIO (CMRR)	$V_{\text{CM}} = -10 \text{ V to } +10 \text{ V}$							
CMRR, DC to 60 Hz	1 k $\Omega$ source imbalance							
G = 1		80			90			dB
G = 10		100			110			dB
G = 100		120			130			dB
G = 1000		130			140			dB
CMRR at 8 kHz								
G = 1		80			80			dB
G = 10		90			100			dB
G = 100		100			120			dB
G = 1000		110			120			dB

### Offset Voltage / Offset Voltage Drift

As with operational amplifiers (Op-Amp) a good in-amp must have low offset voltage. Even though there are different architectures for the in-amps, generally there are two stages – input and output stage; so the total output offset will be equal to the gain of the amplifier times the input offset plus the offset of the output amplifier.

Typical values are  $1\mu\text{V}/^\circ\text{C}$  for the input and  $10\mu\text{V}/^\circ\text{C}$  for output offset drift respectively. Even though the initial offset voltage may be taken care of with external trimming, the temperature drift cannot be adjusted out.

**Table 2: AD8295 Offset Characteristics**

VOLTAGE OFFSET	RTI $V_{OS} = (V_{OSI}) + (V_{OSO}/G)$					
Input Offset, $V_{OSI}$	$V_S = \pm 5\text{ V to } \pm 15\text{ V}$		120		60	$\mu\text{V}$
Over Temperature	$T_A = -40^\circ\text{C to } +85^\circ\text{C}$		150		80	$\mu\text{V}$
Average TC			0.4		0.3	$\mu\text{V}/^\circ\text{C}$
Output Offset, $V_{OSO}$	$V_S = \pm 5\text{ V to } \pm 15\text{ V}$		500		350	$\mu\text{V}$
Over Temperature	$T_A = -40^\circ\text{C to } +85^\circ\text{C}$		0.8		0.5	mV
Average TC			9		5	$\mu\text{V}/^\circ\text{C}$
Offset RTI vs. Supply (PSR)	$V_S = \pm 2.3\text{ V to } \pm 18\text{ V}$					
G = 1		90	110	94	110	dB
G = 10		110	120	114	130	dB
G = 100		124	130	130	140	dB
G = 1000		130	140	140	150	dB

As mentioned earlier these results are under specific test conditions. In this case the Gain is set to be equal to unity. Since usually much higher gain is used one should be aware that the drift offset will also be variable with the gain.

### Matched High Input Impedance

The impedance of the inverting and the non-inverting input terminals must be high and closely matched to one another. This is required to avoid loading the input signal source which can affect the input signal voltage. Input impedance values that are in the range  $10^9\Omega$  to  $10^{12}\Omega$  are typical.

**Table 3: AD8295 Input Impedance Characteristics**

INPUT				
Input Impedance				
Differential		100  2		100  2 $G\Omega  \text{pF}$
Common Mode		100  2		100  2 $G\Omega  \text{pF}$

### Low Input Bias and offset Current Errors

Having op-amps at its input-stage the instrumentation amplifier has bias currents that flow either into, or out of, its input terminals. Bipolar in-amps have base currents and FET in-amps have gate leakage currents. This bias current flowing through an imbalance in the signal source resistance will create an offset error. They can be modeled like current sources as shown on the figure bellow:

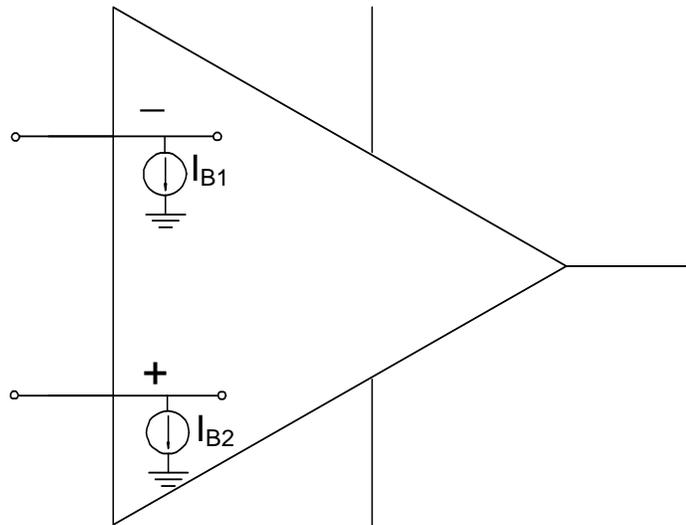


Figure 4: Modeling Input Bias Currents

Input offset current errors are defined as the mismatch between the bias currents flowing into the two inputs ( $I_{OS} = |I_{B1} - I_{B2}|$ ). The average value of the input bias currents is usually referred as the Input Bias Current ( $I_B = (I_{B1} + I_{B2})/2$ ). Typical values of input bias current for a bipolar in-amp range from 1 nA to 50 nA; for a FET input device, values of 1 pA to 50 pA are typical at room temperature.

Table 4: AD8295 Input Current Characteristics

INPUT CURRENT								
Input Bias Current			0.5	2.0		0.2	0.8	nA
Over Temperature	$T_A = -40^\circ\text{C to } +85^\circ\text{C}$			3.0			1.5	nA
Average TC			1			1		pA/°C
Input Offset Current			0.2	1		0.1	0.5	nA
Over Temperature	$T_A = -40^\circ\text{C to } +85^\circ\text{C}$			1.5			0.6	nA
Average TC			1			0.5	2	pA/°C

### Low Noise

To be able to handle very low level input voltages, an in-amp must not add its own noise to that of the signal. A minimum input noise level of 10 nV/√Hz @ 1 kHz (gain > 100) referred to input (RTI) is desirable.

Table 5: AD8295 Noise Characteristics

NOISE	RTI noise = $\sqrt{e_{NI}^2 + (e_{NO}/G)^2}$			
Voltage Noise, 1 kHz				
Input Voltage Noise, $e_{NI}$	$V_{IN+}, V_{IN-}, V_{REF} = 0V$	8	8	nV/√Hz
Output Voltage Noise, $e_{NO}$	$V_{IN+}, V_{IN-}, V_{REF} = 0V$	75	75	nV/√Hz

### Gain Selection

There are different ways for setting gain for an In-Amp. The use of a single external gain resistor is common and is the way the gain is set for AD8295, but an external resistor affects the circuit’s accuracy and gain drift with temperature. Some in-amps provide a choice of internally preset gains that are pin-selectable. For example AD621 has on chip pre-trimmed gain resistors for gains of 10 and 100. The AD621 is preset to a gain of 10. A single external jumper (between Pins 1 and 8) is all that is needed to select a gain of 100.

Another typical way to set the gain is through software. An example for this is AD8250 which can be configured for gains of 1, 2, 5 and 10.

Also there are fixed gain In-Amps like AD8225, which has a fixed gain of 5. Having fixed gain has the plus side that there is no interference with the gain resistor which results in less noise and better CMRR.

### Bandwidth

An instrumentation amplifier must provide bandwidth sufficient for the particular application. Since typical unity gain, small-signal bandwidths fall between 500 kHz and 4 MHz, performance at low gains is easily achieved, but at higher gains bandwidth becomes much more of an issue so it is a tradeoff the user have to choose from.

Table 6: Typical small signal bandwidths

In-Amp	Small Signal Bandwidth
AD8295	1200 KHz
AD624	1 MHz
AD8225	900 KHz
AD621	800 KHz

Table above shows typical values for small signal bandwidth for several Analog Devices instrumentation amplifiers.

### Input and Output Swing

Modern in-amps often need to operate on single-supply voltages of 5 V or less. So called rail-to-rail operation means that an amplifier’s maximum input or output swing is essentially equal to the power supply voltage. In fact, the input swing can sometimes exceed the supply voltage slightly, while the output swing is often within 100 mV of the supply voltage or ground (in the case of a single supply operation). Important as well is the common mode input voltage. On a datasheet it is common to see a graph like this that is characterizing the output limits as a function of the input common mode voltage:

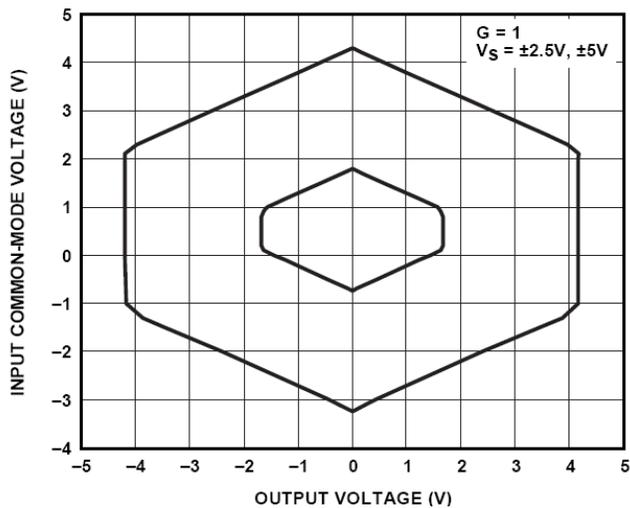


Figure 5: Input Common-Mode Range vs. Output Voltage,  $G = 1$ ,  $V_S = \pm 2.5 V, \pm 5 V$ ,  $REF = 0 V^2$

<sup>2</sup> AD8295 Datasheet [www.analog.com](http://www.analog.com)

## *Applications*

In-amps are very useful for a variety of applications but mainly are used for amplifying signals from low level output transducers in noisy environments. Signals from pressure sensors (barometers) or temperature transducers (RTDs), as well as signals from common bridge applications like load cells or strain gauges are all ideal for in amps. In-amps are also widely used in medical equipment such as EKG and EEG monitors and blood pressure monitors. Different possible applications are investigated in more depth in chapter 2 as a part of the brainstorming process for the project.

### 1.3. The AD8295 – A System Level Solution

The AD8295 is the latest In-Amp Integrated Circuit from Analog Devices. Released in October 2008, the product comprises of a precision Instrumentation Amplifier, two uncommitted op-Amps, and a matched 20kΩ resistor pair all on the same silicon substrate. This assortment of devices makes the product a system-on-chip solution and a versatile design choice for many Analog Front Ends (AFE's). The size of the LFCSP package is 4mm X 4mm which reduces PCB board space by half when compared to an SOIC package.

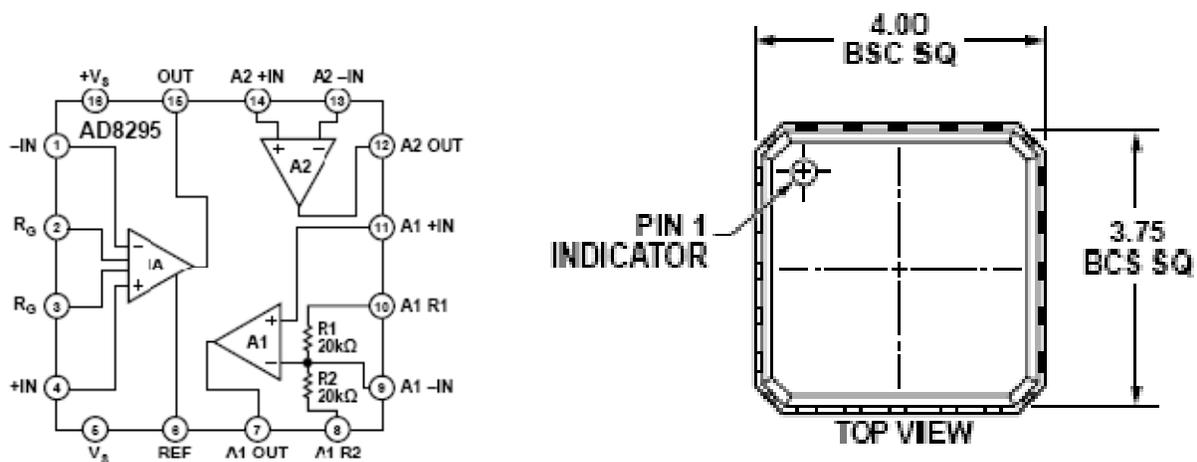


Figure 6: Block Diagram of AD8295 (left) LFCSP Package Dimensions in mm (right)<sup>3</sup>

The product is targeted for (but not limited to) the following applications: Differential Output Drive, Weigh Scales, Industrial Process Controls, Bridge Amplifiers, Precision Data Acquisition Systems, Medical Instrumentation, Strain Gauges, Transducer Interfaces, and Voltage-Controlled Current Sources. The following sections describe the main components of the AD8295.

<sup>3</sup> Taken From AD8585 Datasheet

## Instrumentation Amplifier

The Instrumentation Amplifier for the AD8295 is designed based on the classic three op-amp topology discussed in previous sections. The transfer characteristic is given by:

Equation 5: Transfer characteristic for the in-amp of AD8295

$$V_{out} = G * (V_{in+} - V_{in-}) + V_{ref} \quad (5)$$

Where the gain is a function of the gain resistor  $R_G$  and has the following equation:

Equation 6: Calculating Gain of the in-amp of AD8295

$$G = 1 + \frac{49.4k\Omega}{R_G} \quad (6)$$

The AD8295 defaults to  $G = 1$  when no gain resistor is used. Gain accuracy is a combination of both the  $R_G$  accuracy, the accuracy listed in the specifications, including accuracy over temperature. Gain error and gain drift are kept to a minimum when the gain resistor is not used.<sup>4</sup>

## Uncommitted Op-Amps

The AD8295 has two uncommitted op amps that can be used independently. These op amps allow simple pin-strapping for many common applications circuits. Op Amp A1 (Figure 6) has its inverting input connected to a precision 2:1 voltage divider resistor network. Because this network is internal to the IC, these resistors are closely matched and also track each other, over temperature variation. Op Amp A1 and the associated resistor network can be used to create either a non-inverting gain stage with gain of 2 or an inverting gain stage with gain of  $-1$  with excellent gain accuracy and gain drift. Op Amp A2 is the same as A1 in terms of design and specifications, the only difference is that it is not connected to the resistor network in any way. Both A1 and A2 have a gain-bandwidth product of 1MHz and a slew rate of 2.6V/ $\mu$ s. For further specs, please refer to the AD8295 datasheet.<sup>4</sup>

---

<sup>4</sup> [Appendix B: AD8295 Datasheet](#)

### *Internal Resistor Network*

The two 20k $\Omega$  resistors are very useful for an application that requires a well matched resistive divider. For example, they are ideal for use as a feedback network for one of the on-chip op-amps (A1) where gain accuracy is important. Additionally, the on-chip location of the resistors provides enhanced temperature drift performance. The resistors are specified as having a (worst-case, absolute) accuracy of 0.2% and a (worst-case, absolute) Temperature Coefficient of -50 ppm/ $^{\circ}$ C.<sup>5</sup> The resistors matching (ratio-metric) is given as 0.03% and the tracking Temperature Coefficient is given as 1ppm/ $^{\circ}$ C.

### *Reference Terminal*

The output voltage of the AD8295 instrumentation amplifier is developed with respect to the potential on the reference terminal. The REF pin is protected with ESD diodes, and the REF voltage should not exceed either +V<sub>S</sub> or -V<sub>S</sub> by more than 0.3 V. For best performance, the source impedance to the REF terminal should be kept below 1  $\Omega$ . Additional impedance at the REF terminal can significantly degrade the CMRR of the amplifier. When the reference source has significant output impedance (for example, a resistive voltage divider), buffering the signal before driving the REF pin is highly recommended for accurate reference voltage. Internal Op Amp A1 or A2 can be used for this purpose.<sup>5</sup>

### *PCB Layout for the AD8295*

The AD8295 is a high precision device. To ensure optimum performance at the PCB level, care must be taken in the board layout. The AD8295 pins are arranged in a logical manner to aid in this task. Traces from the gain setting resistor to the R<sub>G</sub> pins should be kept as short as possible to minimize parasitic inductance. To ensure the most accurate output, the trace from the REF pin should either be connected to the local ground of the AD8295 or to a voltage that is referenced to the local ground of the AD8295.<sup>5</sup>

---

<sup>5</sup> Appendix B: AD8295 Datasheet

### *Measured Performance Characteristics*

As part of our background research we decided to perform some basic characterization of the AD8295 using the Evaluation Board provided by ADI. This included DC gain tests and Bandwidth Measurements for different gain settings (G=1, 10, 100, 1000). For data pertaining to the DC Gain tests, see the Appendix on Measured Performance Characteristics. These tests were performed using the AD8295 Evaluation Board shown below. This board was designed by ADI for the purpose of bench characterization specifically for this part.



Figure 7: AD8295 Evaluation Board

### *DC Gain Measurements*

DC Gain Measurements<sup>6</sup> were for G=1, 10, 100, 1000 by choosing  $R_g$  values according to Equation. However, due to the limitation of standard valued resistors, an  $R_g$  value close to the theoretical value was used and this difference was accounted for while calculating the Gain. Next, a measured differential input voltage was applied to the in-amp, and the output was measured. A series of points like these were taken and the resulting data plotted. The Slope of the best-fit line was taken as the measured Gain. Below is the collected data and plot for G=10. The other plots are located in APPENDIX H.

---

<sup>6</sup> [Equipment Used: HP 34405A Digital Multimeter, HP 33120A Arbitrary Waveform Function Generator, HP 3631A Power Supply.](#)

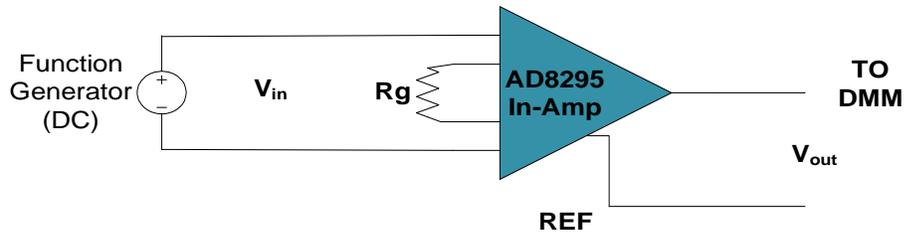


Figure 8: DC Gain Test Schematic

Calculated Gain (Based on measured  $R_g=5.026\text{kohms}$ ) = **10.829 v/v**

Measured Gain (Based on best-fit slope) = **10.808 v/v**

Offset (Based on Intercept of best-fit line using Excel Trend Function) = **-70 $\mu\text{V}$**

Table 7: Gain of 10 Evaluations:

G=10	
Vin	Vout
0.0000E+00	-3.1000E-04
1.0004E-01	-1.0810E+00
2.0005E-01	-2.1620E+00
3.0007E-01	-3.2430E+00
4.0001E-01	-4.3240E+00
1.0000E+00	-1.0808E+01
1.1000E+00	-1.1889E+01
7.0000E-01	-7.5660E+00

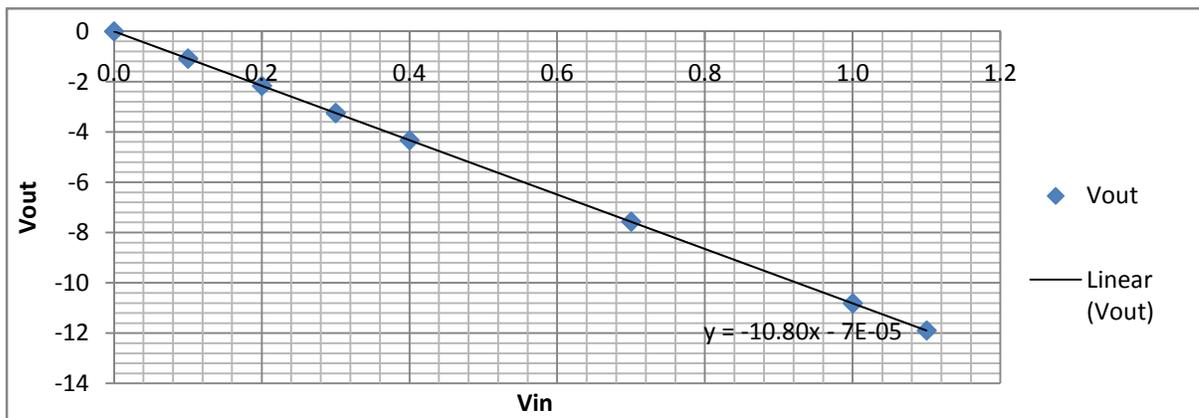


Figure 9:  $V_{IN}$  vs  $V_{OUT}$ , G = 10

## Bandwidth Measurement

The Bandwidth measurement<sup>7</sup> was essentially a measure of the gain of the In-Amp Vs. frequency. This was done for G= 1, 10, 100, 1000. Gain was measured by recording  $V_{IN}$  and  $V_{OUT}$  readings from the Oscilloscope, and calculating the gain from there. While making this measurement, especially for high gains, there was a tradeoff between maintaining sufficient signal resolution at the output at higher frequencies (>3dB) and exceeding the output range of the amplifier at DC. Thus, for these higher gains, a resistive voltage divider was used at the input to make the tradeoff good enough for adequate measurement across the frequencies of interest. For example, while measuring Bandwidth for G=100, a 0.01 attenuation divider was used. The sample data for G=10 is given below, along with a graph showing Bandwidth for all measured gains (overlaid with the corresponding datasheet TPC). Bandwidth data for the other gain values can be located in Appendix H.

Table 8: Bandwidth Data - G=100

Bandwidth G = 100						
Frequency	Vout	Vin	Gain	Gain(db)	Vin(gen)	Atten.Factor
1.00E+02	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02
2.00E+02	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02
3.00E+02	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02
4.00E+02	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02
1.00E+03	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02
2.00E+03	5.92E+00	6.04E-02	9.81E+01	3.98E+01	6.68E-01	9.04E-02
3.00E+03	5.92E+00	6.04E-02	9.81E+01	3.98E+01	6.68E-01	9.04E-02
4.00E+03	5.96E+00	6.12E-02	9.74E+01	3.98E+01	6.77E-01	9.04E-02
1.00E+04	5.92E+00	6.12E-02	9.68E+01	3.97E+01	6.77E-01	9.04E-02
2.00E+04	5.88E+00	6.12E-02	9.61E+01	3.97E+01	6.77E-01	9.04E-02
3.00E+04	5.83E+00	6.12E-02	9.53E+01	3.96E+01	6.77E-01	9.04E-02
4.00E+04	5.79E+00	6.12E-02	9.47E+01	3.95E+01	6.77E-01	9.04E-02
1.00E+05	5.02E+00	6.12E-02	8.21E+01	3.83E+01	6.77E-01	9.04E-02
2.00E+05	3.57E+00	6.12E-02	5.84E+01	3.53E+01	6.77E-01	9.04E-02
3.00E+05	2.53E+00	6.12E-02	4.14E+01	3.23E+01	6.77E-01	9.04E-02
4.00E+05	1.85E+00	6.12E-02	3.02E+01	2.96E+01	6.77E-01	9.04E-02
1.00E+06	4.47E-01	6.12E-02	7.31E+00	1.73E+01	6.77E-01	9.04E-02
2.00E+06	7.90E-02	6.04E-02	1.31E+00	2.34E+00	6.68E-01	9.04E-02
3.00E+06	1.94E-02	6.04E-02	3.21E-01	-9.86E+00	6.68E-01	9.04E-02
4.00E+06	6.40E-03	6.04E-02	1.06E-01	-1.95E+01	6.68E-01	9.04E-02

<sup>7</sup> Equipment Used: HP 34405A Digital Multimeter, HP 33120A Arbitrary Waveform Function Generator, HP 3631A Power Supply, Tektronix TDS2004B Oscilloscope.

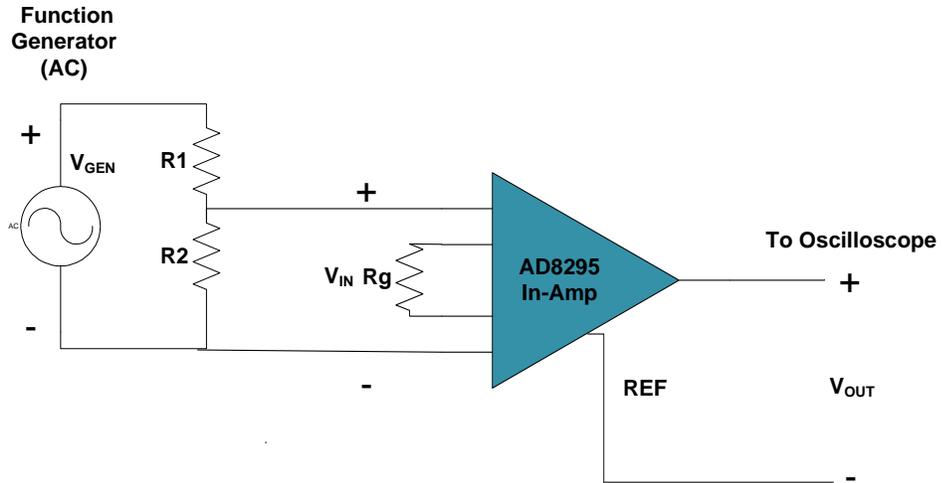


Figure 10: Bandwidth Measurement Schematic

Below is the measured result for Bandwidth for all the Gains measured (1, 10, 100, 1000). The data is overlaid with the corresponding Datasheet TPC.

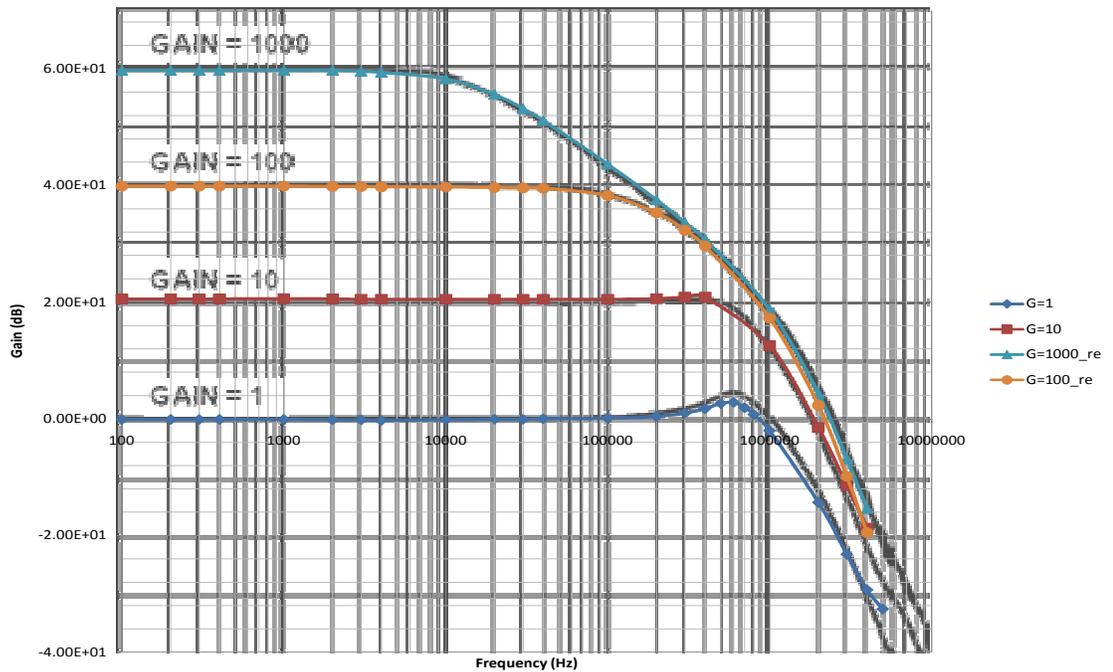


Figure 11: Bandwidth Overlay

## 2. Brainstorming Applications

One of the main goals of this project was to develop an application demo for an In-Amp product. This application demo would be used by ADI to demonstrate the qualities of their product at Electronics Trade Shows such as Electronica and Sensor Expo. The AD8295 was suggested as a good candidate for this project as its versatility (in-amp, two extra op-amps, and resistive divider) allowed it to be used in a variety of different applications. Thus, we began our brainstorming with the objective of finding applications that would be “good fits” for the AD8295. In addition, based on Oljeta Bida’s suggestion, we also looked at the datasheet’s of certain other in-amp products (listed in the table below).

As such, there were certain requirements that any Application would have to meet. These are listed below:

- 1) The Demo should be visually appealing, tactile and interactive so that a person visiting a trade-show would be attracted by it.
- 2) The performance of the Application Demo, should not present any safety issues unsuitable for a trade-show.
- 3) The implementation of the idea should make use of the unique capabilities of the AD8295 in-amp, such as the 20K $\Omega$  matched resistor pair. And the two extra uncommitted op-amps.
- 4) The Demo should make use of the usual advantages of using In-Amps High Gain, Common Mode Rejection, and Low Noise.
- 5) The scope of the project should be appropriate for completion on the accelerated A-B term schedule (4 months).

Table 9: Suggested ADI Products for Application Research

Product	Category
AD833X Family	Variable Gain Amplifiers
AD827X Family	Differential Amplifiers
AD822X Family	Resistor Gain In Amps
AD825X Family	Digitally Programmable In Amps
AD823X Family	Auto-Zero In-Amp & Op-Amp
AD8293	CMOS In-Amp
AD821X Family	Current Sensors

The next step we took was to try and research already-in-use applications for these products so as to come up with an idea for an improvement with the AD8295. We realized that there were a lot of applications to choose from and decided to split these up into categories, weight the choices with the requirements (1 through 5) listed above and pick the one that was most suitable. The Following is a list of the final contenders.

- 1) Electrocardiogram
- 2) Strain-Gage Using AC excitation
- 3) RTD Temperature Sensor Circuit
- 4) Single Supply Barometer
- 5) Voltage Controlled Current Source
- 6) Glove Mouse
- 7) Pulse Oximetry
- 8) Capacitive Sensing
- 9) Temperature Based Dual-Axis Tilt Meter

## Application #1: Electrocardiogram

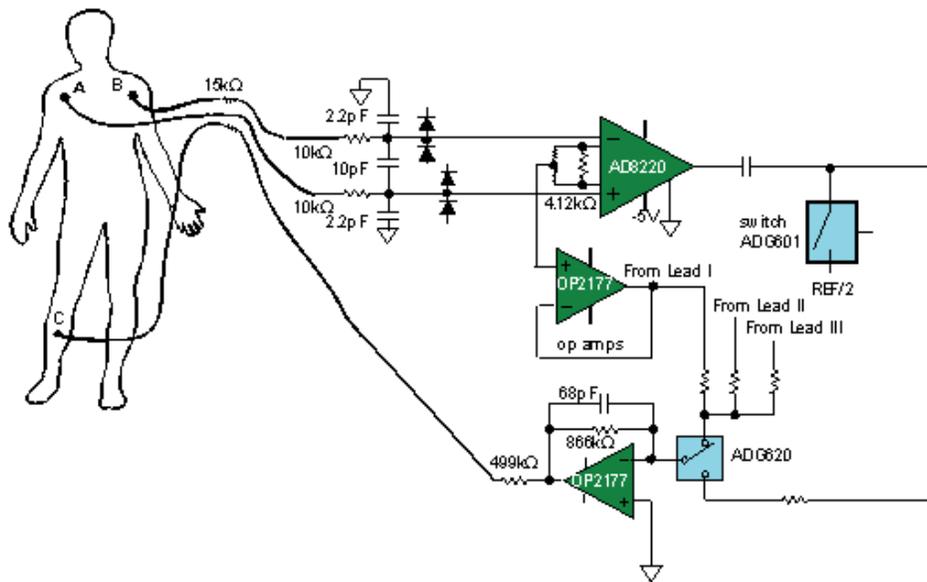


Figure 12: Patient side of the ECG signal-chain.<sup>8</sup>

In the figure above the application schematic describes the analog front end of an ECG monitor<sup>8</sup>. The differential ECG signal is amplified by the In-Amp (AD8220). The first op-amp OP2177 buffers the common-mode voltage taken at the midpoint of the AD8220 gain-setting resistors. The second OP2177 op-amp inverts the signal (and works as a low-pass filter), feeding it back into the right leg of the patient which cancels the common-mode signal by driving it back into the body.

The ECG signal, as measured from the patient's skin, has the following profile:

Table 10: ECG Signal Profile<sup>9</sup>

Signal Amplitude	Signal Frequency
0.1mV - 5mV	0.05Hz - 130 Hz

<sup>8</sup> Source: [www.analog.com](http://www.analog.com) (medicalapplication\_ECG.ppt)

<sup>9</sup> Measuring the ECG Signal with a mixed analog-digital application specific IC, [Hewlett-Packard Journal](#), Oct, 1991 by [Wolfgang Grossbach](#)

### *Advantages*

There were two reasons as to why we felt that ECG was a good application for the AD8295. Firstly, in the schematic in Figure 12 there are three primary components (AD8220 In-Amp, 2 X OP2177 op-amps) which could be replaced with the AD8295 alone. This would utilize the AD8295 entirely, as well as save board space on an end-user's circuit board. Secondly, the small signal amplitude and large common mode voltages presented by a human bodies ECG signal would be ideal for an In-Amp to process as it has both high gain and high CMRR (common mode rejection ratio) in the frequencies range of ECG

### *Disadvantages*

The primary disadvantage with the ECG was that it presented some difficulties in making the demo process quick and smooth. The person taking the demo would have to wear electrodes on his/her body and this would cause inconvenience and delays.

## Application #2: Strain-Gage using AC excitation

Strain gage measurements are often plagued by offset drift, 1/f noise, and line noise. The idea, proposed in [10], is to use an AC signal to excite the bridge to eliminate DC drifts and noise. The AD8221 gains the signal and an AD630AR synchronously demodulates the output waveform with respect to the known excitation signal. What results is a DC output proportional to the strain on the bridge. The output signal is devoid of all DC errors associated with the in-amp and the detector including offset and offset drift.

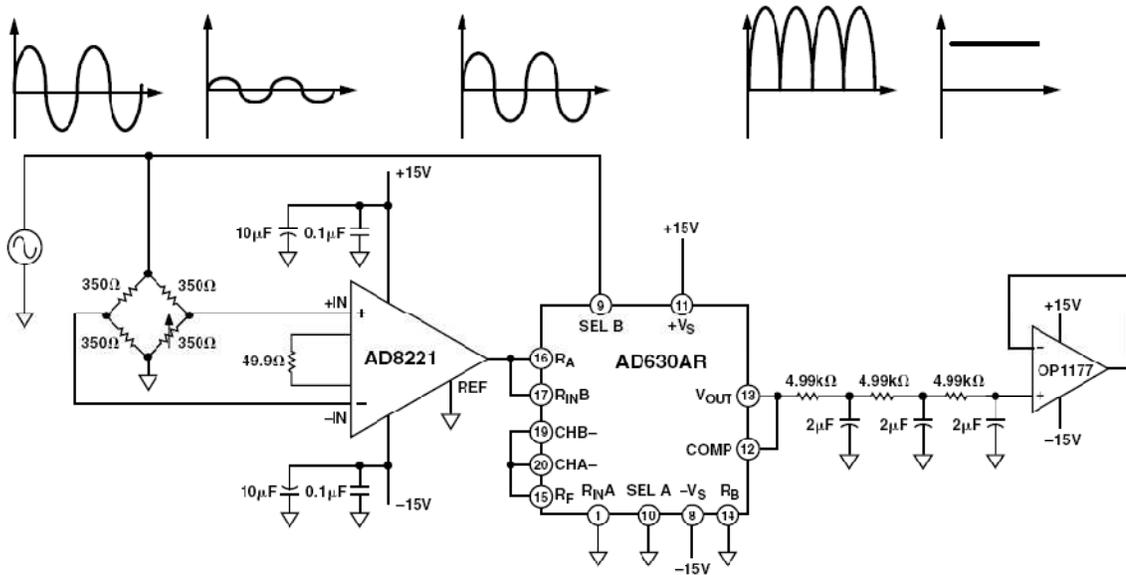


Figure 13: Strain Gage Application (Using AC excitation) <sup>10</sup>

A possible implementation of this idea could involve using the AD8295 in place of the AD8221 and incorporating the output buffer and a low-pass filter using the two active components (A1, A2) in the AD8295. Additionally, two of the Wheatstone-Bridge resistors could be provided by the AD8295 as well.

<sup>10</sup> Application Note AN-683, ADI, by Moshe Gerstenhaber and Stephen Lee

### *Advantages*

The implementation of the circuit with the AD8295 would incorporate the use of all of the additional components (namely: A1, A2 and the resistive divider). Thus it would be a good fit to demonstrate the utility of the product.

### *Disadvantages*

The appeal of the application from the perspective of being an interactive application demo was not very high. Additionally, in terms of the implementation, the idea was too restrictive and did not allow for the investigative research experience warranted by the MQP.

### Application #3: RTD Temperature Sensor

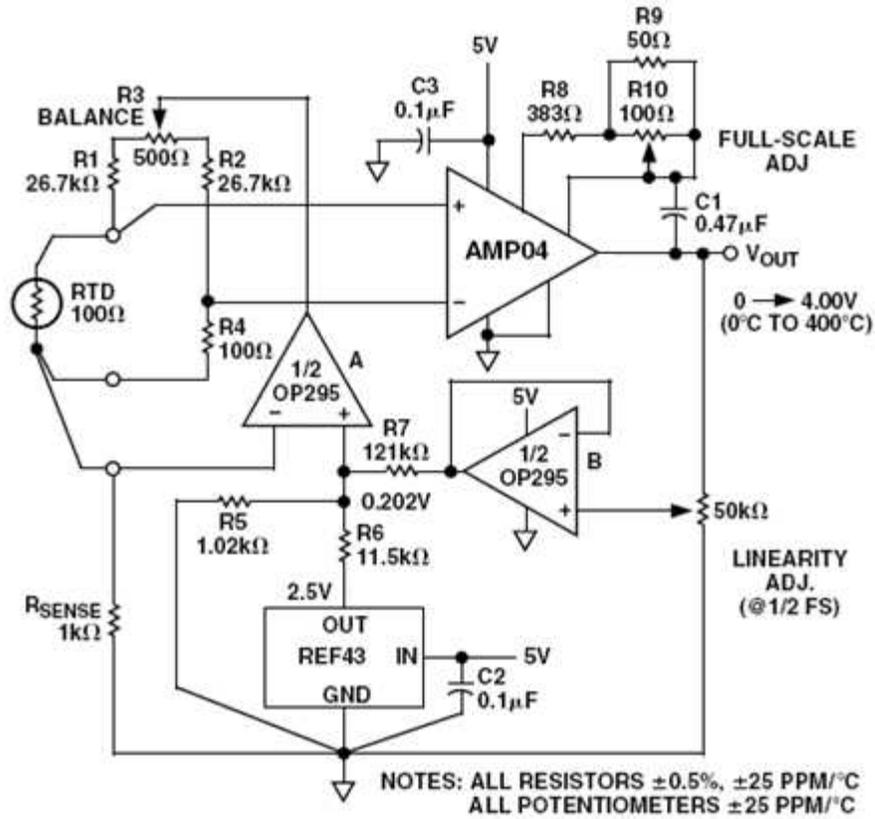


Figure 14: Precision Single Supply RTD Thermometer<sup>11</sup>

Resistance Temperature Detectors (RTDs) are temperature sensors that make use of the temperature dependence of a metal's resistance. They are used in a wide variety of temperature measurement and control applications. We found an application note that detailed the use of two Op-Amps and an In-Amp to make a linearized RTD Thermometer circuit.

The application is well suited for a precision temperature measurement from 0 Degrees to 400 Degree Celsius. As shown in Figure 14, one of the op-amps (B) is used in conjunction with the Linearity ADJ potentiometer to calibrate

<sup>11</sup> Source: AMP04 Datasheet

the temperature sensing circuit. The REF43 voltage regulator provides a reference voltage of 2.5 volts which is divided down to 0.202 volts for op-amp B to provide constant current to the RTD in a Wheatstone bridge configuration.

### *Advantages*

The RTD application provides us with the option of using the two uncommitted op-amps and enhances the space saving features of the AD8295. Also, the RTD circuit's temperature sensors are used in a wheat-stone bridge configuration which could incorporate the 20K precision resistors present in the AD8295.

### *Disadvantages*

This circuit was extensively simulated and we performed analysis to determine its overall transfer function and the motivation behind using certain components. We used this idea as a rigorous exercise in circuit analysis. However, the three-point calibration scheme and lack of user-interaction in the application were deterrents in its role as a live demo.

## Application #4: Single Supply Barometer

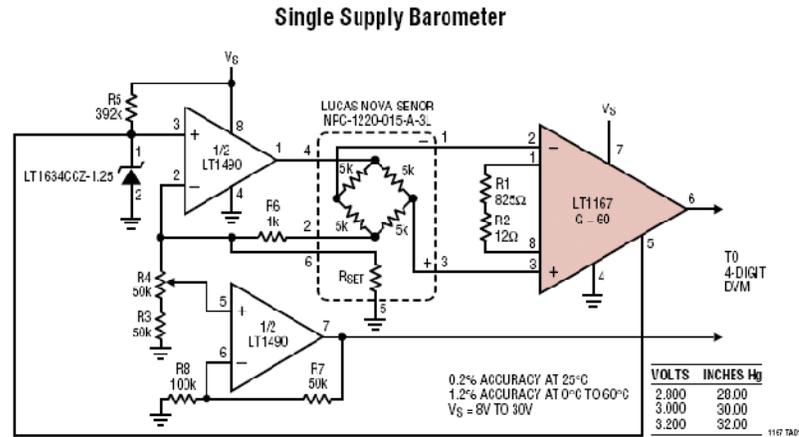


Figure 15: Single Supply Barometer <sup>12</sup>

The above circuit explores the use of an instrumentation amplifier in a Barometer application. In this case a Lucas Nova sensor (based on a Wheatstone-Bridge topology) is excited by an op-amp providing constant current (LT1490). Another Op-amp (LT1490) is used to calibrate the circuit. The in-Amp (LT1167) is used to process the signal from the transducer.

### Advantages

Barometer applications are a common use of in-amps since you want to extract a very small signal. And, as seen in the previous application ideas, the fact that the AD8295 can eliminate the need for additional components for the two op-amps and two bridge resistors is an added advantage.

### Disadvantages

Again, the primary disadvantage with this application was that it would not be very user-interactive for a Trade-Show Demo.

<sup>12</sup> Source: LT1167 datasheet

## Application #5: Voltage Controller Current Source

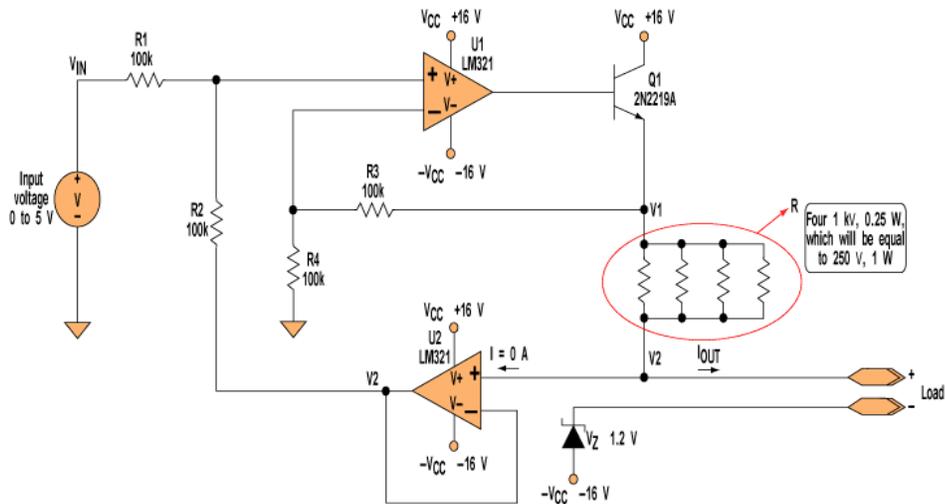


Figure 16: Voltage Controlled Current <sup>13</sup>

One challenge in circuit design is building a good current source, especially when the load is variable or the current must be controlled with a voltage source. The figure shows a simple, low-cost voltage-controlled current source using two op amps that provides a good range of current and maximum load. The idea is to apply a voltage on a reference resistor (or resistors) having a low thermal coefficient. The current passing through this resistor will be the output current. The First Op-Amp U1 adds the input voltage with V2, and second Op-Amp U2 buffers the load voltage. Depending on the application we decide we might need to drive a load with a voltage controlled current source.

<sup>13</sup> Source - <http://electronicdesign.com/Articles/Index.cfm?AD=1&ArticleID=9018> (last accessed 09/03/2008)

### *Advantages*

In the circuit given above, the two Op-Amps can be the ones we have inside the package of AD8295. This will take advantage of the uncommitted op-amps in the package, saving space on the board.

### *Disadvantages*

The voltage controlled current source can be a good addition to our project. But this alone cannot be an application for our project, since this idea does not use an In-Amp. Thus, we kept the idea as a possible addition to our project, but moved away from the idea of using this design as the core of our project.

## Application #6: Glove Mouse

Another idea for an application was to make a computer mouse that looked like a glove. The concept is fairly simple: there are going to be 5 strain gauges along the fingers of the glove. When bending different fingers, the strain gauge output could be associated with different actions on the computer like a click, scroll, zoom-in or zoom-out. The movement would be tracked with a three dimensional accelerometer, and a three dimensional gyroscope could be used for tracking rotational movements. After the glove (which is going to be the input stage) there would be a signal processing stage (possibly in-amp circuitry) followed by a standard ADC and subsequent USB transfer. In the programming process there can be included moves like bending all the fingers for calibration in and this can position the mouse in the middle of the screen. There are many more applications that can be achieved with this input device like having an artificial hand mimicking the movement of your hand.



Figure 17: X-IST data glove using fiber optic bend sensors<sup>14</sup>

There are companies that already offer similar gloves. VRealities is one of the distributors for variety of gloves for different applications. The use of five transducers in sensing finger movement, provides us with the option of using five AD8295, signifying the space saving attribute of the chip.

---

<sup>14</sup> <http://www.vrealities.com/x-ist.html>

### *Advantages*

This idea made for a very user-interactive and engaging Trade Show Demo. Furthermore, the use of strain-gages as sensors provided the possibility of some analog signal processing stages.

### *Disadvantages*

Firstly, the idea did not really require use of an In-Amp. There was a possibility of using in-amps to process the strain-gage signal but it should be noted that for the application, only “discrete” changes corresponding to “clicks” or “scrolls” would be tracked by the strain gage thus making a precise analog solution an overindulgence.

Secondly, given the time constrains of the project, a complex demo such as this might not be within the scope of our project timeline.

## Application #7: Pulse Oximetry

### Pulse Oximetry - Theory

From: IEEE Biomedical Engineering Handbook pgs 1349-1352

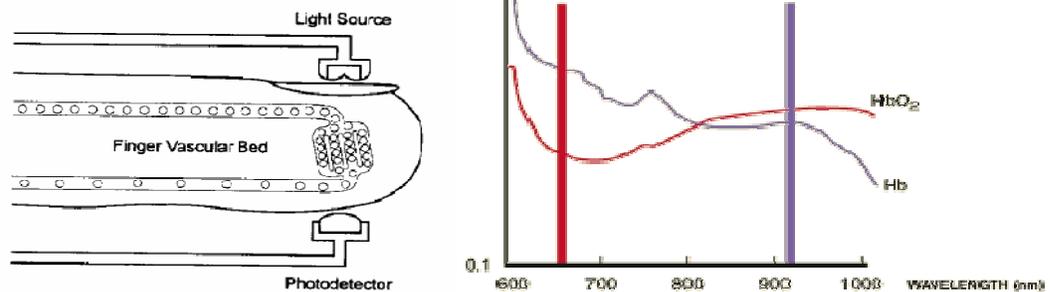


Figure 18: Pulse Oximetry Setup (Left) Absorption Spectra for Oxy/Deoxy-hemoglobin(Right)<sup>15</sup>

The Pulse Oximeter is a medical device that indirectly measures the oxygen saturation of a patient's blood, as well as the patient's pulse. A pair of small light-emitting diodes (LEDs) facing a photodiode through a translucent part of the patient's body (usually a fingertip or an ear-lobe) constitute the basic setup. Absorption of red (660nm) and infra-red (910nm) light differs spectrally between oxyhemoglobin and deoxyhemoglobin as shown below in, therefore from the ratio of the absorption of the red and infrared light, the oxy/deoxyhemoglobin ratio can be calculated.

### Advantages

The Pulse-Oximeter provides us with the option of being a non-invasive medical application that can be demonstrated at tradeshow. Furthermore, in this application there are uses of filtering, which can be achieved by employing the additional op-amps (A1 and A2) in the AD8295 as active filters.

### Disadvantages

The Pulse-Oximeter application involves use of amplifiers and possible active filtering. However, this application did not involve working with very small signals; the principal reason for using an In-Amp.

<sup>15</sup> <http://www.oximeter.org/pulseox/principles.htm>, last accessed: Sunday, November 23<sup>rd</sup>, 2008

## Application #8: Capacitive Sensing

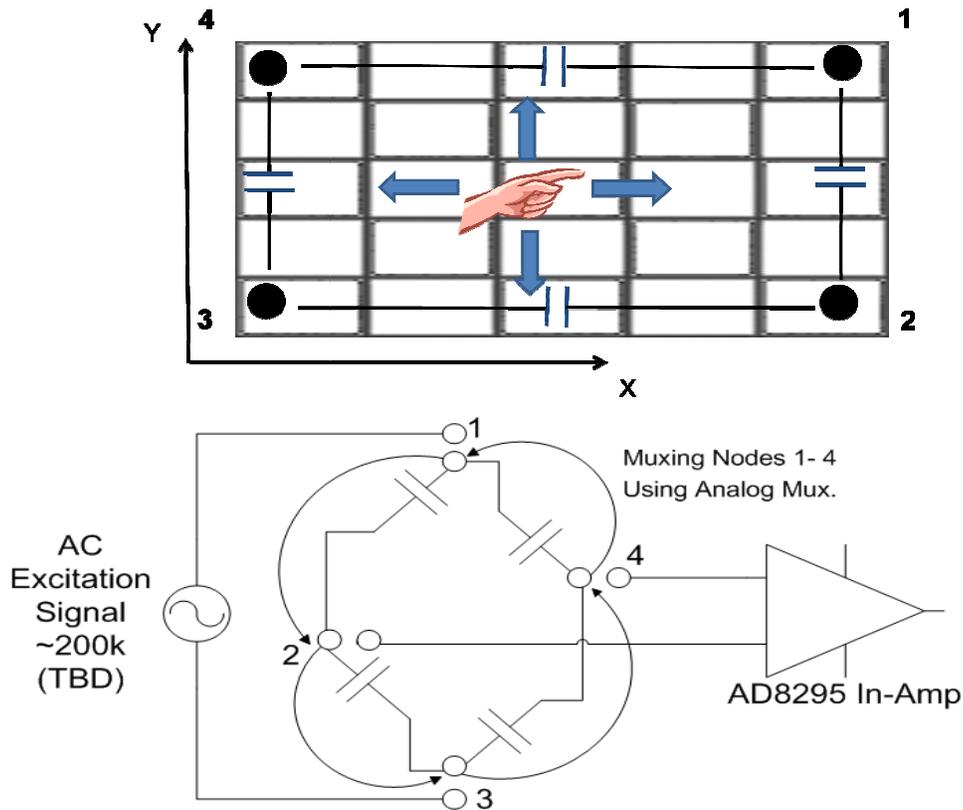


Figure 19: Proposed Touch-screen (Top), Circuit Configuration (Bottom)

We considered the idea of capacitive sensing as a way to track finger movement. The idea was to use copper patches in the four corners of a laptop screen and effectively form a touch screen. The patches would be arranged in a capacitive Wheatstone-Bridge configuration. The bridge would be excited with an AC signal source. As shown in Figure: 8, the 8295 in-amp would track the differential voltage across the mid-branches of the bridge. When a finger would be moved along this 'touchpad', the differential voltage developed across the bridge output would be proportional to the finger position. Pairs of capacitors could be used to achieve orthogonality and provide x and y-axis information. This output could then be used as position coordinates for a computer input-device.

### *Advantages*

This idea was definitely one of the top contenders from our list. The appeal for a simple solution to motion sensing that could be used as a simple attachment to a laptop would definitely have mass appeal. Also, the High CMR of the In-Amp made it ideal for processing a differential AC signal from a large common-mode voltage.

### *Disadvantages*

After initial prototyping of the circuit, we ran into problems with repeatability, broadband noise and found that we needed to connect the user's body to one of the copper patches to allow the finger capacitance to 'couple' into the input signal. So effectively the user would have to touch the pad in two places to use the application. These drawbacks, coupled with the scope of the project forced us to look for another application.

## Application #9: Temperature Based Dual-Axis Tilt-Meter

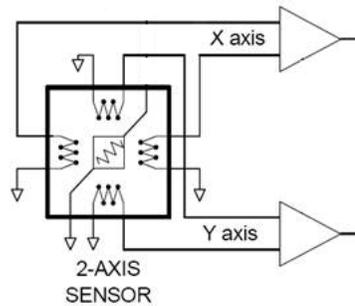


Figure 20: Dual Axis Accelerometer Concept<sup>16</sup>

This idea is based on a product made by MEMSIC, a semiconductor company that primarily produces accelerometer IC's. The concept, illustrated in Figure 20, involves using a resistor as a constant heat source and using thermistors (arranged as shown) to extract a differential voltage signal proportional to the temperature difference that is created between the thermistors when the sensor is tilted along the x or y axes. This temperature difference occurs due to heat transfer by convection when the sensor is tilted. The output voltage can be used to move a pointer on an x-y plane based on "tilt" along the x and y direction.

### *Advantages*

This application was a good fit for the AD8295 as it involved sensing a small differential signal (ideal for in-amp) and required a Wheatstone Bridge sensor configuration (ideal for two on-chip resistors and op-amp as excitation source). The idea was something relatively new and unique and would therefore appeal to people. Lastly, implementing this idea could employ all of the AD8295's key components.

### *Disadvantages*

The disadvantage with this idea was that the performance of the sensor would have to be characterized and improved through experimentation as this type of sensor had not been widely used yet.

<sup>16</sup>Taken From MXA2050A Datasheet ([www.memsic.com](http://www.memsic.com))

## Discussion on Application Choice

For the purpose of our project, we decided to pick one of these ideas to make our demo board based on its relevance, and project constraints mentioned earlier. Towards this, we prepared the following metric, with assigned weights for each project-requirement, to help us pick a final application.

Table 11: Metric for Picking Application

Application	Goals					TOTAL Score (Out of 100)
	Wt = 3	Wt = 2	Wt = 2	Wt = 2	Wt = 1	
	Interactive	Safe	AD8295 Friendly	In-Amp Friendly	Within Scope	
1) Electrocardiogram	2	2	4	5	5	66
2) Strain-Gage (AC Excitation)	2	5	5	5	5	82
3) RTD Temp. Sensor Circuit	1	5	5	5	5	76
4) Single Supply Barometer	2	5	5	5	5	82
5) Voltage Controlled I Source	2	5	3	1	5	58
6) Glove Mouse	5	5	3	3	3	80
7) Pulse Oximetry	5	5	2	2	4	74
8) Capacitive Sensing	5	5	4	5	2	90
9) Temp. Based Dual Axis Tilt Meter	5	5	5	5	5	100

From this table, it is clear that the Temperature Based Tilt Meter was the clear winner. For the trade show application, we proposed a tilt sensing game that would simulate the movement of a ball on a plane that was being tilted along the x-z and y-z planes (as shown below).

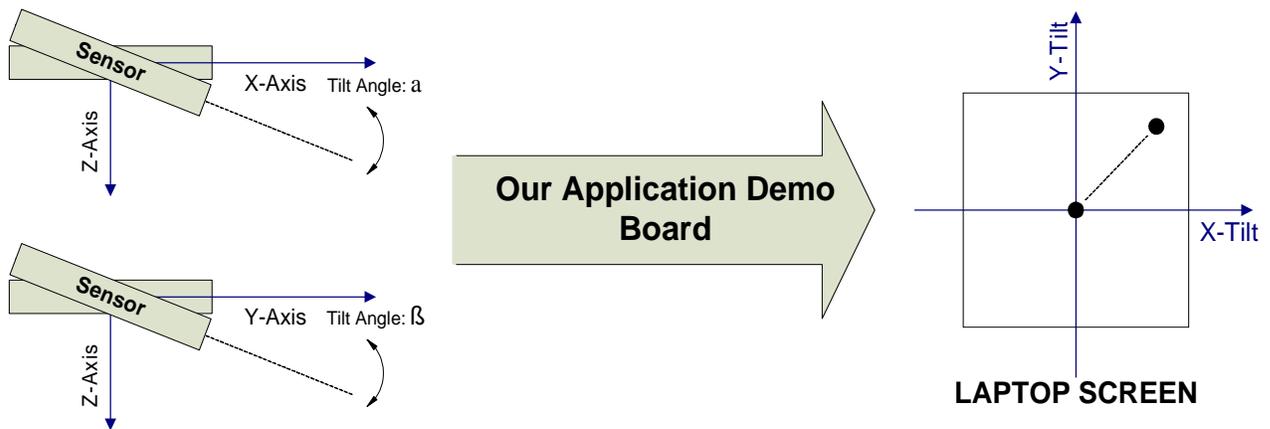


Figure 21: Tilt-Sensing Application

### 3. Design Details

This section describes the design process for our chosen IC application: A temperature based Tilt Sensor. Our basic system block diagram (organized by signal chain) is shown below in Figure 22. What follows is a brief overview of each block followed by a more detailed design description. The next section looks at the overall design of our project to provide an understanding of how the individual components come together.

#### 3.1. Overall Design

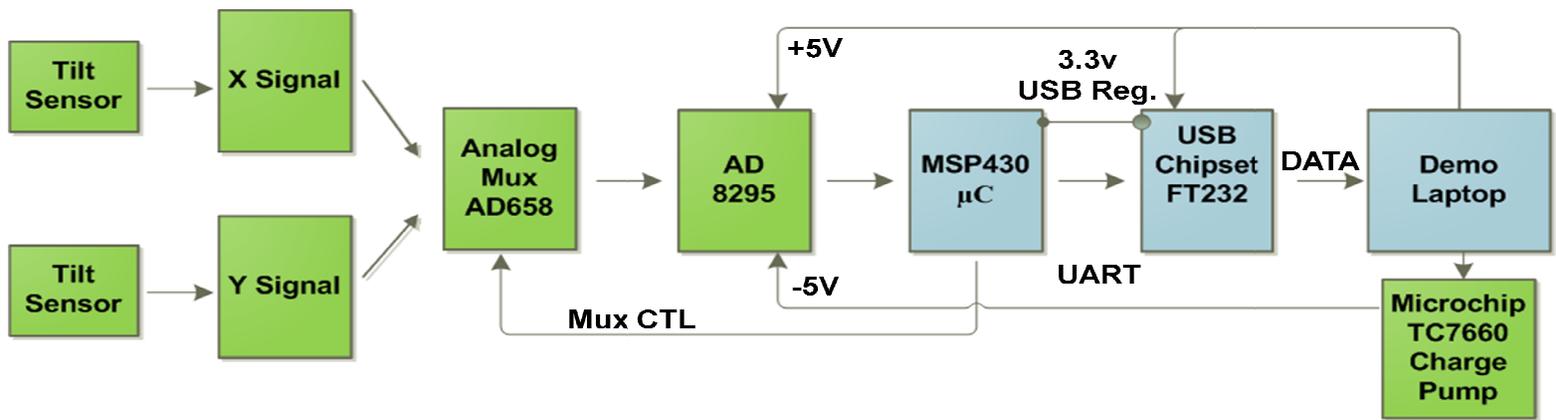


Figure 22: System Block Diagram

##### 3.1.1. Sensor

This block is comprised of an analog tilt sensor that employs the principle of heat transfer by convection to determine the acceleration along a given direction. This idea is based on a product manufactured by a company called MEMSIC which produces similar accelerometers, along with mixed-signal processing circuitry implemented on a monolithic IC. The sensor for our application is based on this idea and attempts to implement this idea with standard surface mount/through-hole components. Our goal is to obtain analog outputs proportional to the ‘tilt-angle’ between the x-axis and the vertical plane, as well as the y-axis and the vertical plane. These analog outputs would ultimately be used as co-ordinates that would track the tilt of the sensor on an X-Y plane being displayed real-time on a laptop screen.

The figure below illustrates the idea:



Figure 23: System Level Explanation of Sensor Block

### 3.1.2. Analog Multiplexer: The ADG659

In our design, the analog output voltage (proportional to tilt) is produced for the X and Y direction by two separate sensor outputs. Our design uses an Analog Multiplexer (the ADG658) to multiplex these sensor outputs to a single In-Amp. The basic circuit for this is as shown below.

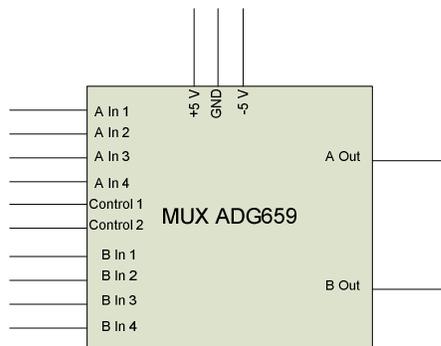


Figure 24: Analog MUX – Simplified Schematic

### 3.1.3. The AD8295

This design block serves multiple functions in our circuit. The included In-Amp serves as the signal conditioning amplifier for the tilt sensor inputs. One of the Uncommitted Op-Amps serves as a buffer that excites the top of our “sensor Wheatstone Bridge”, as shown in Figure. The two 20k resistors shown are in fact the on-chip resistors included

in the AD8295. Lastly, the other uncommitted op-amp is being used to drive the REF pin of the In-Amp, as shown in Figure 24. The output of the in-amp then goes to the next block in the signal chain – the MSP430 microcontroller.

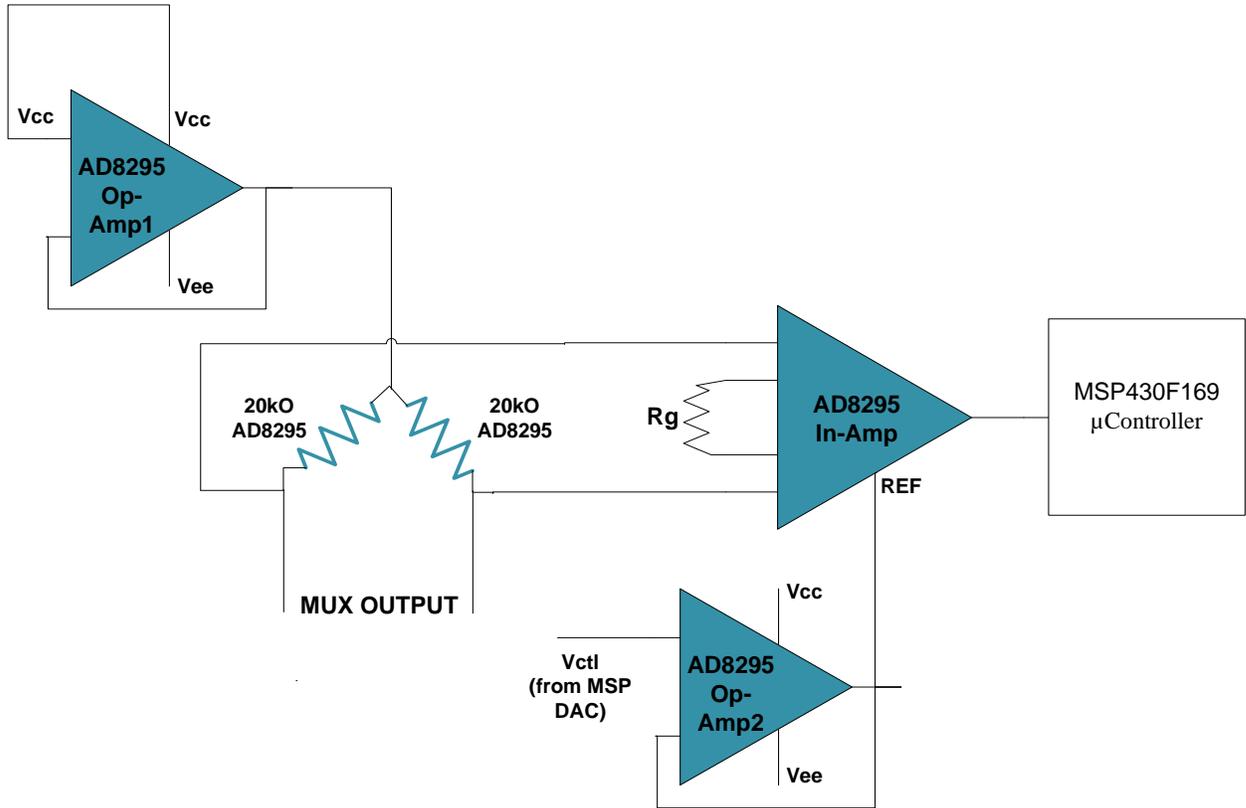


Figure 25: AD8295 Basic Circuit

### 3.1.4. The MSP430F169

This block comprises of the microcontroller module of our system and performs the following tasks:

1. The onboard 12 bit ADC samples the output of the in-amp and sends this data via a UART protocol to the next block (USB Chipset).
2. It controls the operation of the Analog Multiplexer by using digital control signals to select which channel to a multiplexer.
3. The onboard DAC is used to provide the input voltage that drives (through a buffer) the REF pin of the In Amp.

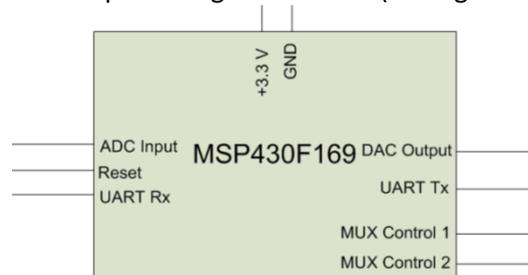


Figure 26: MSP430F169 – Simplified Schematic

### 3.1.5. USB Chipset FT232

This USB chipset serves as the means of communication between the laptop and our board. It accepts as its input a UART protocol data input and sends this via USB to the laptop. Additionally, this chipset also has an onboard 5v to 3.3v regulator so that it is ideal for powering up the MSP as well. This can be seen in the figure below:

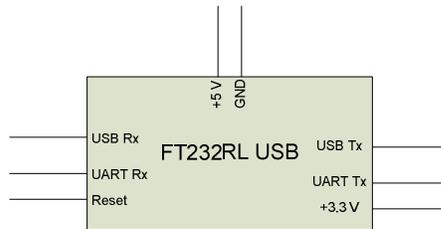


Figure 27: FT232 USB Chipset – Simplified Schematic

### 3.1.6. Demo Laptop

This will be the laptop that an Applications Engineer will bring to the tradeshow as part of this demo. It needs to be able to run the LabView Software Package (8.5) and needs to have a USB port. Other than this, there are no other minimum system requirements for this block.

### 3.1.7. Charge Pump

In order to be able to use the complete dynamic range of the ADC we needed to have the AD8295 work on dual rail supply. Since the requirement was to power the board only through USB which provides only +5V rail we had to use a charge pump to “create” the -5V rail. After appropriate research we chose the Microchip TC7660 Charge Pump.

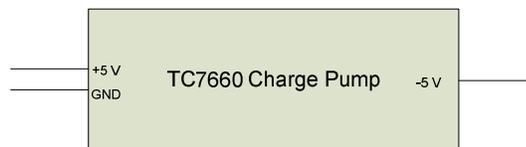


Figure 28: Charge Pump – Simplified Schematic

## 3.2. System Block Diagram Details

### 3.2.1. Sensor

The design of our sensor was based on an idea implemented for accelerometer IC's by a company called MEMSIC. MEMSIC is a start-up company that develops and accelerometers that combine their proprietary MEMS technology along with integrated mixed-signal processing in order to provide a low-cost yet reliable and accurate accelerometer IC fabricated on a standard CMOS process.

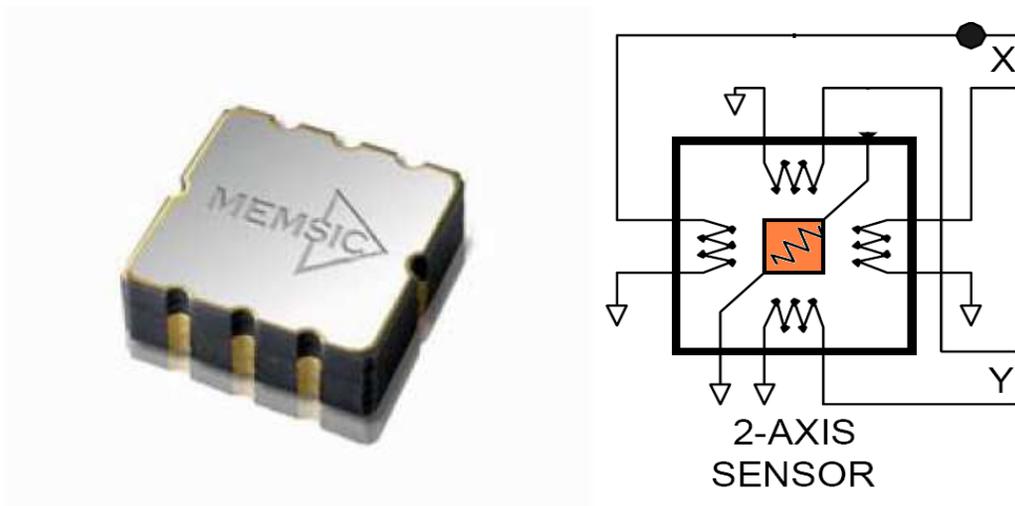


Figure 29: A sample MEMSIC Accelerometer Package (Left) The MEMSIC 2-Axis sensor (Right)<sup>17</sup>

The design of the MEMSIC accelerometers and inclination sensors contain no moving parts and are based on the principle of heat transfer by convection. The idea involves an air-tight cavity with pairs of thermistors on opposite sides of a square heat source. The differential voltage between these opposite thermistor pairs is tracked continuously. When tilted with respect to the z axis (along either x or y axes) a differential voltage develops across the respective thermistor pair. This voltage can be used to extract the angular position of the sensor. A similar technique is used to extract acceleration information from this setup.

<sup>17</sup> Taken from MEMSIC Application Note AN-00MX-001

For our application we decided to implement a similar sensor for the purpose of extracting the degree of tilt (along both x and y) with respect to the z-axis. This sensor would essentially provide the analog signal that, after being processed by the in-amp, would provide x-y directional control for navigating a ball through a maze in the LabView program of our Application Demo. The following sections proceed to describe the various revisions of the sensor

### Sensor Version 1

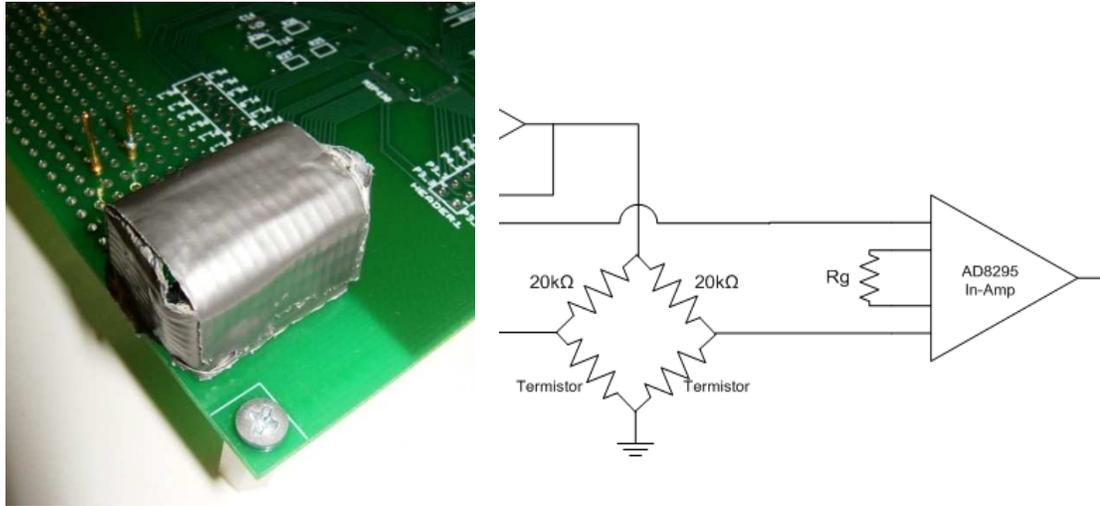


Figure 30: Sensor Version1 (Left) Sensor Wheatstone bridge Setup (Right)

Figure 30 shows our first attempt at the temperature-based tilt sensor and we used standard 5% tolerance 51Ω resistor as our heat source. We passed a current of about 100mA through this resistor to make it dissipate about 500mW of power. Adjacent to this sensor, on both sides, were standard 20kΩ US Sensor thermistors. These were used, along with the pair of 20K resistors from the AD8295, to form a Wheatstone bridge configuration.

To test the quality of this sensor, we performed two measurements - a noise measurement and a step response measurement.

For the Noise measurement the TDS 2004B oscilloscope was used at a timescale of 25s/div to look at the output voltage drift under rest condition. We 'zeroed' the output by applying a buffered voltage at the REF pin to compensate for the DC error of the sensor. The In-amp Gain was set to 25. Output Drift Vs Time (under room temperature in the lab)

showed that the output (with sensor resting on horizontal table) drifts from -600mV to 800mV (1.4V delta). This was pretty high for in order to get meaningful data and indicated that the sensor casing needed to be better isolated thermally.

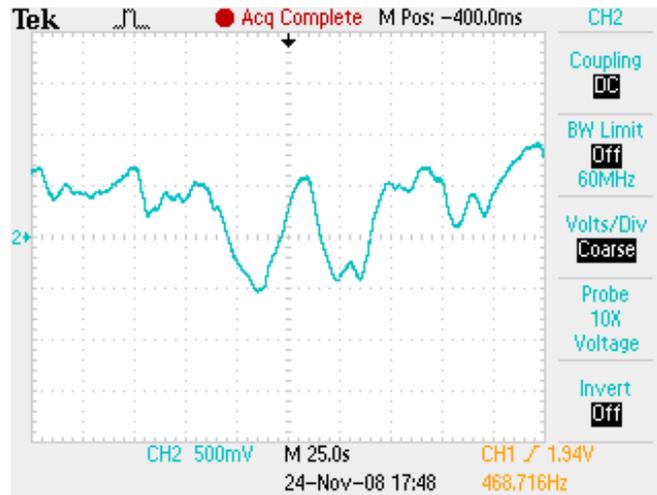


Figure 31: Noise Measurement - Sensor Version1

The Second test, Step Response, was performed to test the response of the sensor when tilted from 0 to +90 degrees and from 0 to -90 degrees. This output was recorded and is displayed below. Before each test, the sensor was allowed to rest for 1-2 minutes before tilting.

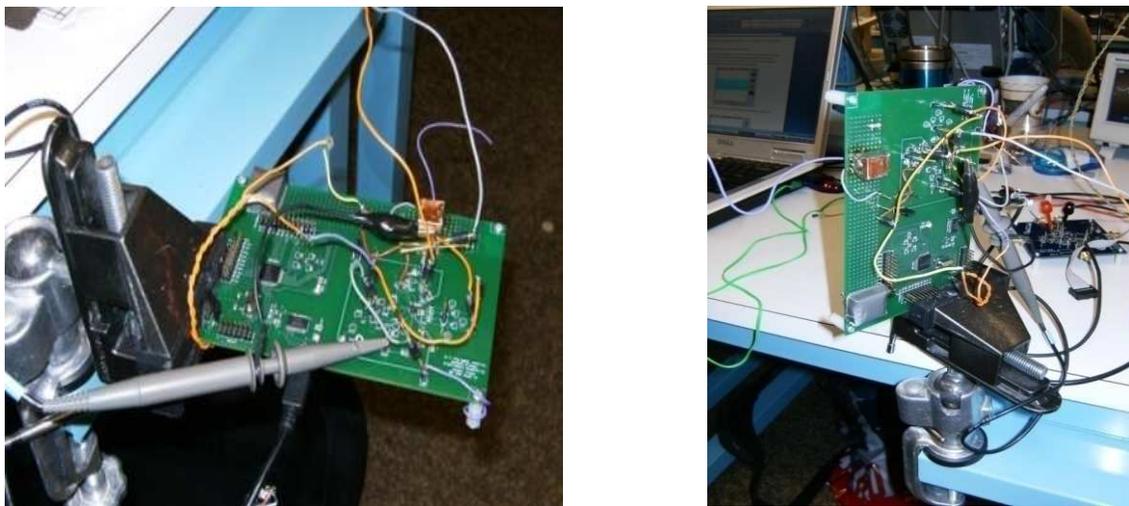


Figure 32: Setup for Step Response Test

The Sensor was first stabilized in the horizontal position, and then turned to +90° while acquiring output data on the scope. (t=25s/div) 50s (full-tilt on output). Similarly, data for -90° was also obtained.

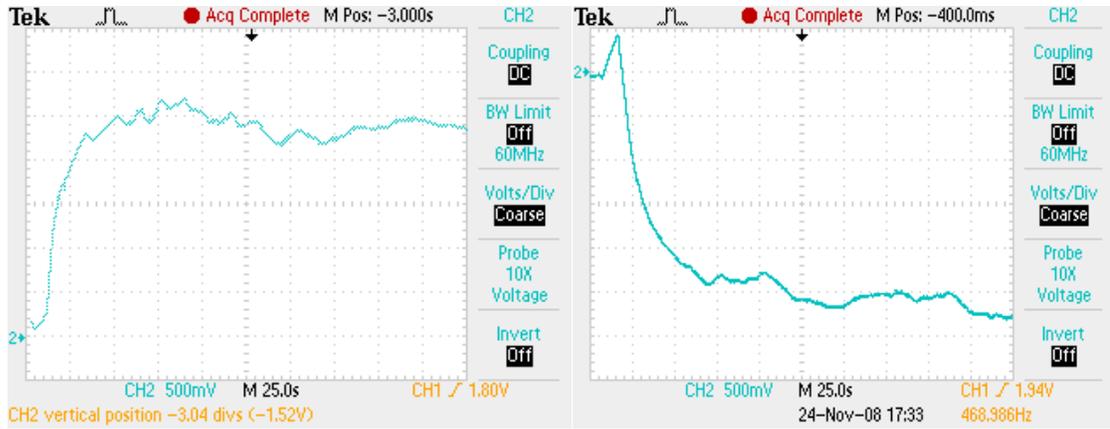


Figure 33: SensorV1 +90 Step (Left) SensorV1 -90 Step (Right)

These graphs showed that it took roughly 50 seconds for the output to rise or fall to full scale for a 90° step. This indicated that measures needed to be taken to speed-up the step response of the sensor. This process led to more revisions of the sensor design.

### Sensor Version2

To try and improve the isolation of our casing, we tried a similar setup but this time we used copper and solder to try and make an air-tight casing. Another motivation for using a metal casing was to check, if this affected the temperature profile inside the cavity favorably for our application. In other words, we were looking to see if the step response for this system was faster or not.

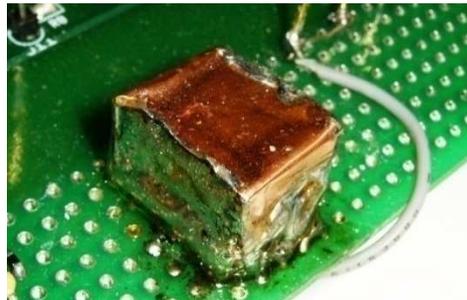


Figure 34: Sensor Version 2

This sensor, when subjected to the same tests as the previous, produced the following results: Noise: 200mV range, step response approximately 50 seconds.

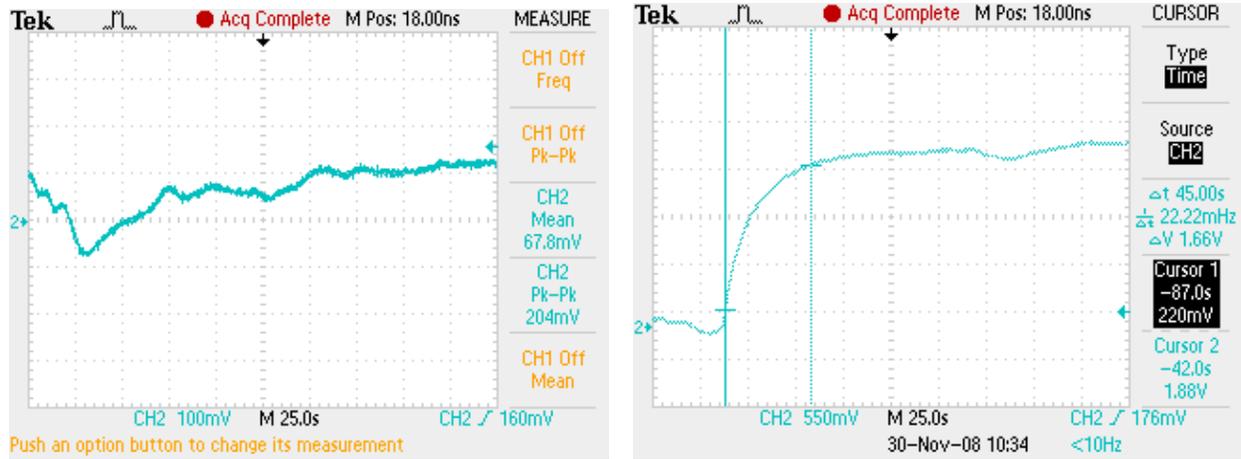


Figure 35: Sensor V2 Noise (Left) Sensor V2 Step Response +90(Right)

### Sensor Version3

Although the previous version did better with noise, we could still see that there was scope for improvement as blowing air on it still disrupted readings. We now tried a third sealing method employing a plexi-glass block and sealed the top with transparent tape (and later a piece of glass using glue).

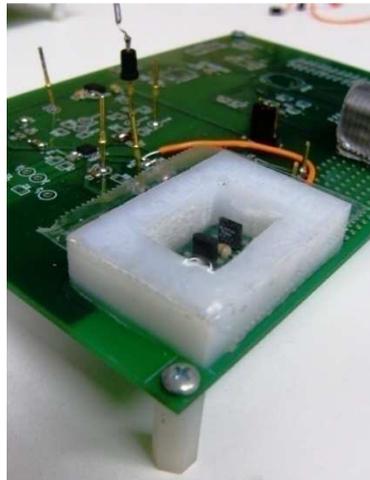


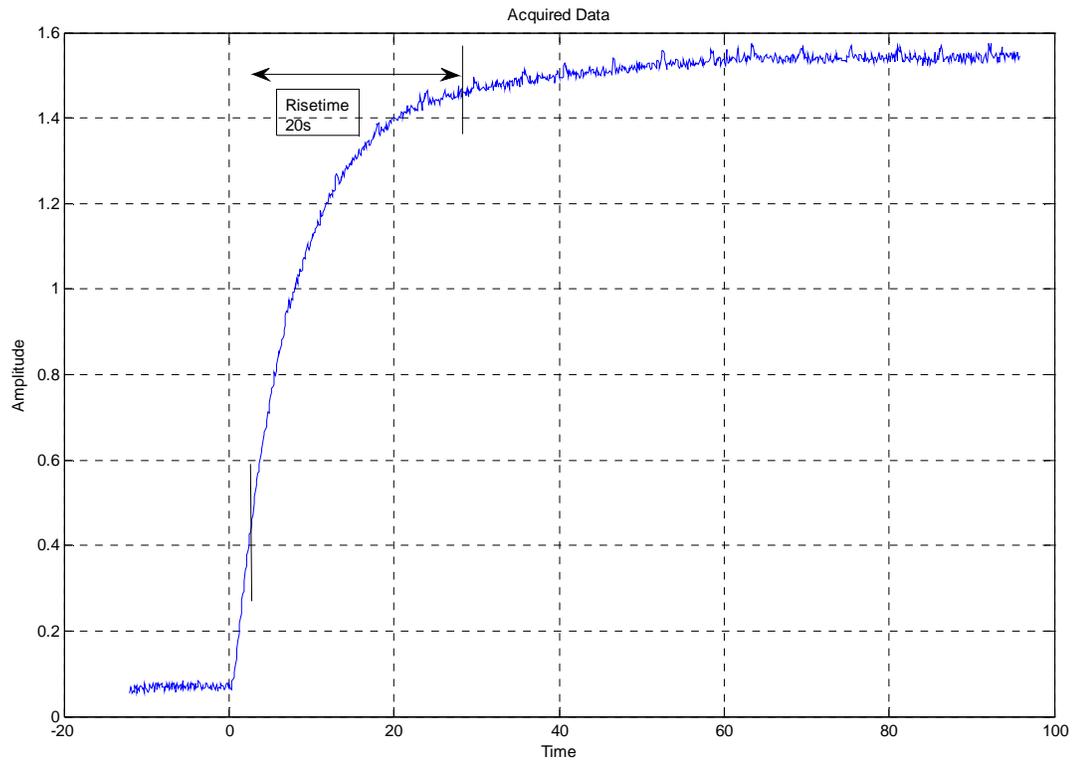
Figure 36: Sensor Version 3

This sensor performed better in terms of noise and eventually evolved into our final design.



Figure 37: Final Sensor Design Side view (Left) and Top view (Right)

This was our final and best revision and provided us with least noisy and fastest response to tilt. The step response time reduced to about 25-30 seconds.



The noise range (max – min) for a 20minute sample, was reduced to 80mV.

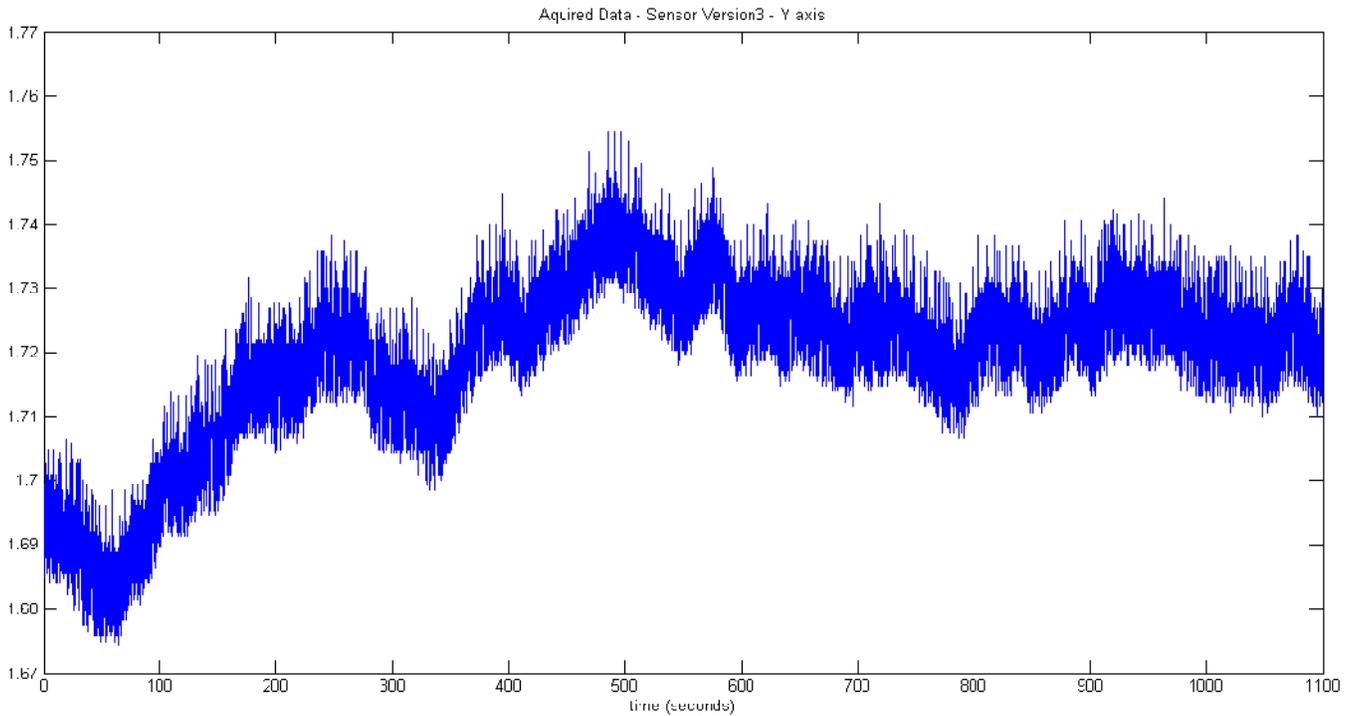


Figure 38: Final Sensor Noise Measurement

### 3.2.2. Analog Multiplexer

For our application, we decided to make the “tilt meter” to have 2 axis of freedom. This brought up the question – should we add another instrumentation amplifier (AD8295) and read the signal through two channels of the ADC, or we can use an analog multiplexer to constantly switch between the two pairs of thermistors corresponding to X and Y direction tilt relative to the direction of the board. The two possible circuits look like the following:

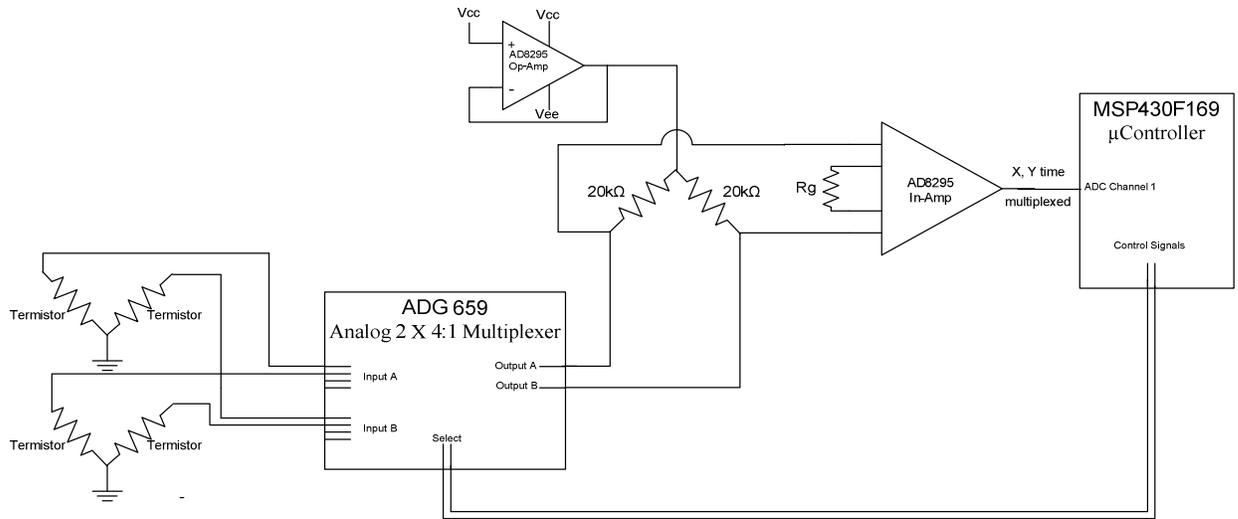


Figure 39: Two channel input using Multiplexer

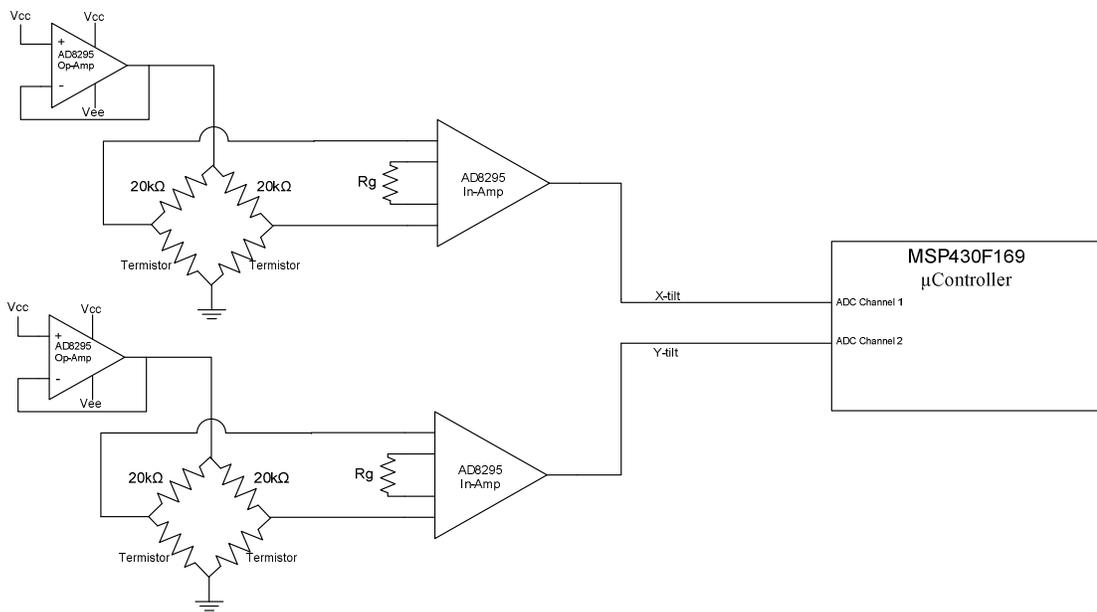


Figure 40: Two channels input using two separate In-Amps

To decide which one of these options were feasible, we made an analysis on the pros and cons of both solutions and the results were summarized in the table below:

Table 12: Comparing solutions for multiple inputs

Decision Criteria	2 x AD8295	1 x AD8295 + 1 x MUX ADG659
Price <sup>18</sup>	AD8295 = \$2.89	ADG659 = \$0.89
Board Space	4x4mm LFCSP	4x4mm LFCSP
Performance	As good as it can be	$R_{ON} \sim 100\Omega$ , $\Delta R_{ON} \sim 3\Omega$

So, as one can see, the only downside of using a multiplexer is the “on” resistance of the channels. This is really not an issue in our circuit since this resistance goes in series with the 20kΩ precise resistors from AD8295. As we can see the delta resistance between the channels is what we are concerned about and it is only 3Ω. We also considered the switching time for the multiplexer and the fact that we will have half of the samples for each channel but this was irrelevant because the bandwidth of the sensor is in orders of magnitude less than the rest of the circuit. So having the same performance and the same space at a lower price tipped the scales towards using the multiplexer solution.

Looking in more details about the ADG659, it was chosen for several reasons. We needed a multiplexer that was at least 2:1 and we have at least two of these on the same IC. We decided to choose one that is 4:1 in case the project needs to be further developed and other input signals are required. We were also looking for a multiplexer that can be easily interfaced with the existing circuit, for example with the μController that is going to drive it. ADG659 all digital inputs have 0.8 V to 2.4 V logic thresholds, ensuring TTL/CMOS logic compatibility when using single +5 V or dual ±5 V supplies, which is what we need to be able to drive it.

**PCB Considerations**

This part is not high speed, thus not requiring extensive PCB considerations. However this chip is part of our input signal, and this is why it is essential to keep the CMRR as low as possible. Thus, careful consideration was given to laying out the connections as symmetrical as possible. The IC can be bought in two packages – LFCSP and TSSOP. If needed, the dimensions can be found on the datasheet of the part and also the footprint of the AD8295 can be used if using the LFCSP package.

<sup>18</sup> Prices are corresponding to [www.analog.com](http://www.analog.com) as of 11/21/2008 and for the minimum quantities

### 3.2.3. Instrumentation Amplifier - AD8295

Our Final Design employed the use of a single AD8295 Instrumentation Amplifier. Sensor outputs from both the X-axis and Y-axis sensors were multiplexed to this in-amp using an ADG569 analog multiplexer. The precision AD8295 20K $\Omega$  resistor pair constituted the 'top' of the sensor Wheatstone bridge. The uncommitted op-amp A1 was used to drive the sensor Wheatstone bridge. Lastly, the uncommitted op-amp A2 was used to drive the REF-pin of the AD8295. The figure below details the AD8295 schematic.

The output of our Instrumentation Amplifier (Net 15 shown in the figure below) goes directly to the Analog to Digital Convertor within our microcontroller (The MSP430F169). Furthermore, the Non-inverting input of A2 is driven by the Digital to Analog Convertor present in the MSP430. This input, in turn, drives the REF pin of the Instrumentation amplifier. Since the In-Amp output is referenced with respect to the REF pin, this feature allows for a Calibration Scheme to be implemented for correcting Offset errors from our Wheatstone bridge sensor network.

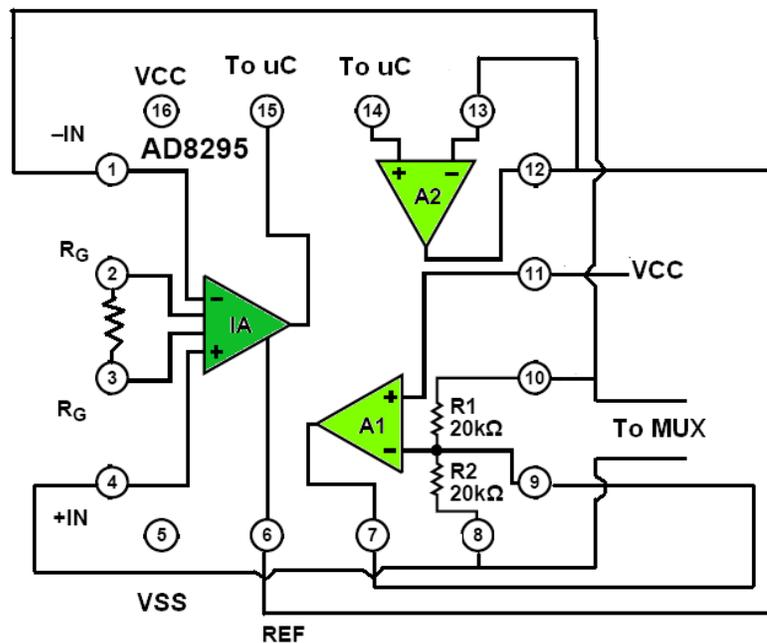


Figure 41: AD8295 Circuit Schematic<sup>19</sup>

<sup>19</sup> Adapted from AD8295 Datasheet, Connection Diagram, Page 1

One design choice that had to be made for the AD8295 was to pick an appropriate gain for the Instrumentation Amplifier. It was agreed to, that the full scale range of the sensor should correspond to the full scale range of the ADC, as this would maximize the resolution at the ADC output. Based on our experimentation, we found that at gain of 12, the output filled the range of the ADC for a full scale swing at the input.

### ***PCB CONSIDERATIONS***

There were certain aspects that needed careful consideration while laying out this device. The high impedance in-amp input lines needed to be symmetric to cancel-out offset errors. The in-amp output voltage line needed a long route to reach the MSP. Care was taken to minimize the effects of parasitic capacitance for these lines. Even though the frequency of the output would not be very high, noise could get coupled in through parasitic capacitances. The AD8295 featured a new LFCSP package with no metal underneath. This allowed for routing directly beneath the package and made certain key routes convenient while also saving board space. Care was also taken to use thicker traces for any power lines with in order to minimize any resistive losses.

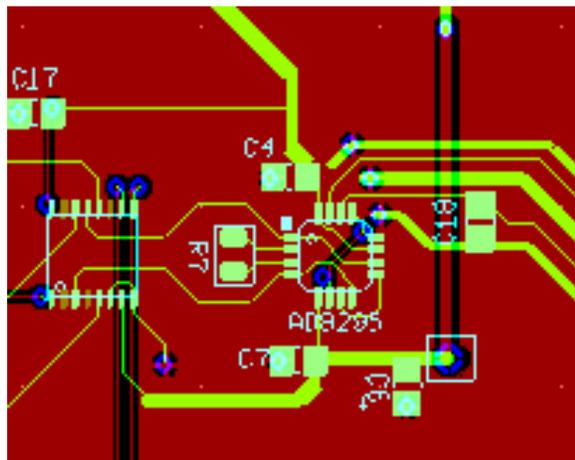


Figure 42: PCB Layout of AD8295

### 3.2.4. $\mu$ Controller - MSP430F169

One of the central components of the circuit is the processing unit. Because of the small time frame of the project we decided to use something that we were familiar with and that can fulfill our requirements. We decided to choose the Texas Instruments MSP430  $\mu$ Controller family. Choosing a specific IC within the  $\mu$ Controller family was decided according to the functional requirements. The main ones are:

- ADC (for acquiring readings)
- UART (to send data to PC)
- DAC (Driving the Ref Pin of the In-Amp)
- MUX controlling

After cross-referencing a list of these  $\mu$ Controllers with our requirements we still ended up with a smaller but still diverse list. We decided to use one that has relatively more computing and Input/Output resources in case we need to add more functionality fairly easily, and finally we picked up MSP430F169. The functional block diagram is shown in the figure below.

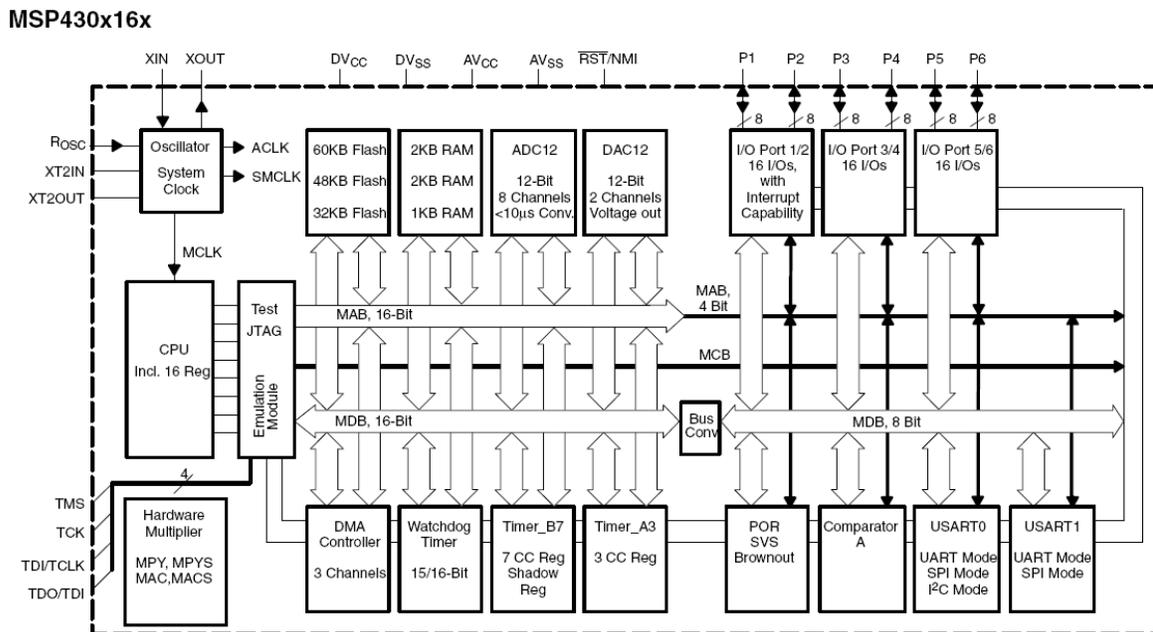


Figure 43: MSP430x16x functional block diagram<sup>20</sup>

<sup>20</sup> MSP430x15x, MSP430x16x, MSP430x161x -MIXED SIGNAL MICROCONTROLLER Datasheet

The basic features of this  $\mu$ Controller<sup>21</sup> (important to our design) are:

- Low Supply-Voltage Range, 1.8 V . . . 3.6 V
- Ultralow-Power Consumption(Active Mode: 330  $\mu$ A at 1 MHz, 2.2 V)
- 12-Bit A/D Converter With Internal Reference, Sample-and-Hold and Autoscan Feature
- Dual 12-Bit D/A Converters With Synchronization
- Serial Communication Interface (USART0)
- 60KB+256B Flash Memory, 2KB RAM

The separate functional sub-blocks that were used were further investigated. A complete explanation of all control registers used can be found in the appendix at the end of this report.

#### ***3.2.4.1. 12bit ADC***

In order to process the acquired information to a computer, the first step is to sample the analog signal. For this purpose we are using the 12-bit ADC inside the  $\mu$ Controller. The ADC12 module supports fast, 12-bit analog-to-digital conversions. The module implements a 12-bit SAR core, sample select control, reference generator and a 16 word conversion-and-control buffer. The conversion-and-control buffer allows up to 16 independent ADC samples to be converted and stored without any CPU intervention. The module is configured with user software and its block diagram is shown below.

---

<sup>21</sup> MSP430x15x, MSP430x16x, MSP430x161x -MIXED SIGNAL MICROCONTROLLER Datasheet

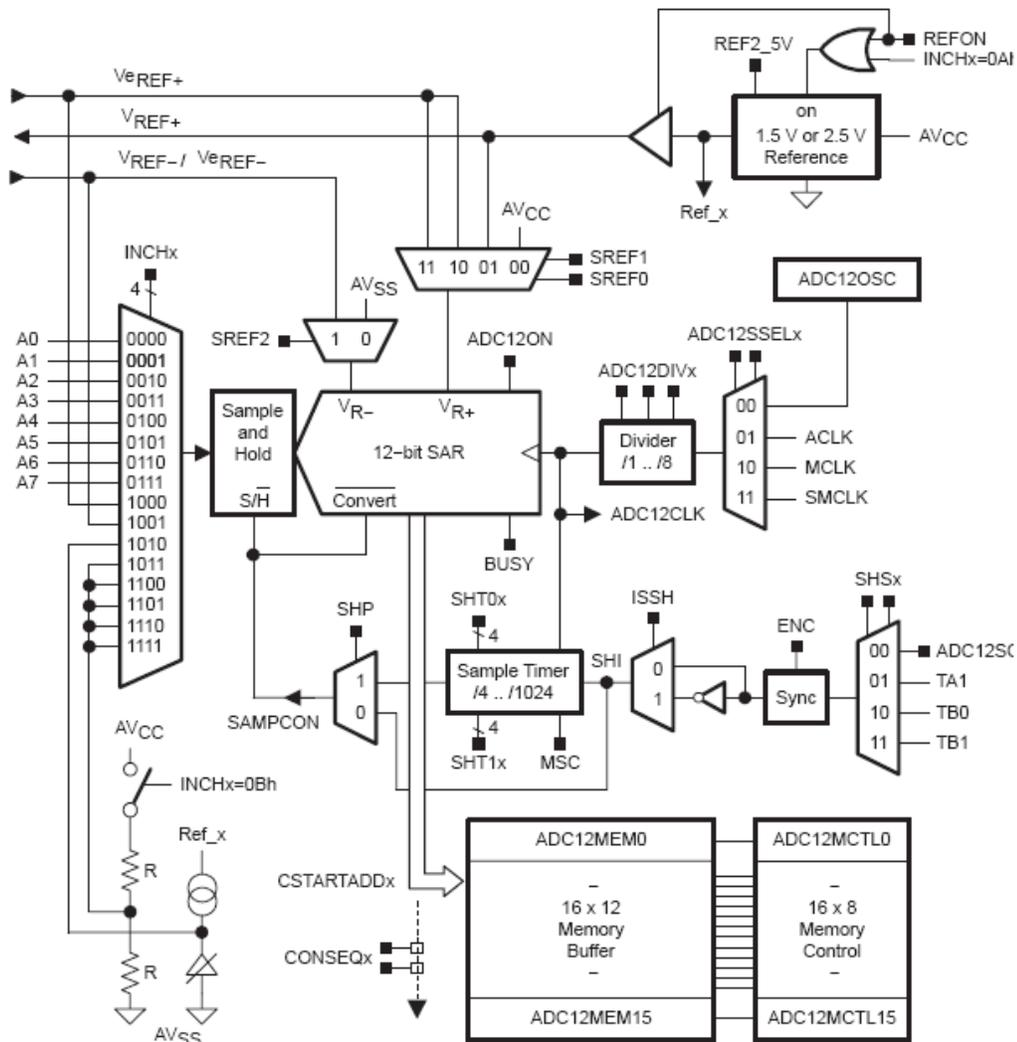


Figure 44: ADC12 Block Diagram<sup>22</sup>

The acquired sample is calculated by the following equation:

Equation 7: Output  $N_{ADC}$  of the ADC

$$N_{ADC} = 4095 * \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}} \quad (7)$$

Configuration of the ADC is done by setting its control registers. These are the main configurations that were used for our project (only the registers that have to be specifically set, for exact way to configure then look at the attached C code).

<sup>22</sup> MSP430x1xx Family User's Guide

Table 13: ADC Registers configuration

Register	Configurations in our project ( explanation )
ADC12CTL0, ADC12 Control Register 0	<b>SHT0x</b> = 0x0100 (256 ADC12CLK cycles, Speed up if needed)  <b>MSC</b> = ON (Multiple sample and conversion)  <b>ADC12ON</b> = ON (turn on the ADC)  <b>ENC</b> = ON ( enable conversions )
ADC12CTL1, ADC12 Control Register 1	<b>ADC12 SSELx</b> = 10 MCLK (Use the Master Clock = 8Mhz)  <b>CONSEQx</b> = Repeat-single-channel (Sequence mode)
ADC12IE, ADC12 Interrupt Enable Register	<b>ADC12IE</b> = 0x08 (Enable ADC12IFG.3, Interrupts)

### 3.2.4.2. UART

In order to send data to a PC we use the universal synchronous/asynchronous receive/transmit (USART) peripheral interface. It supports two serial modes with one hardware module. We configured it for operation in the asynchronous UART mode.

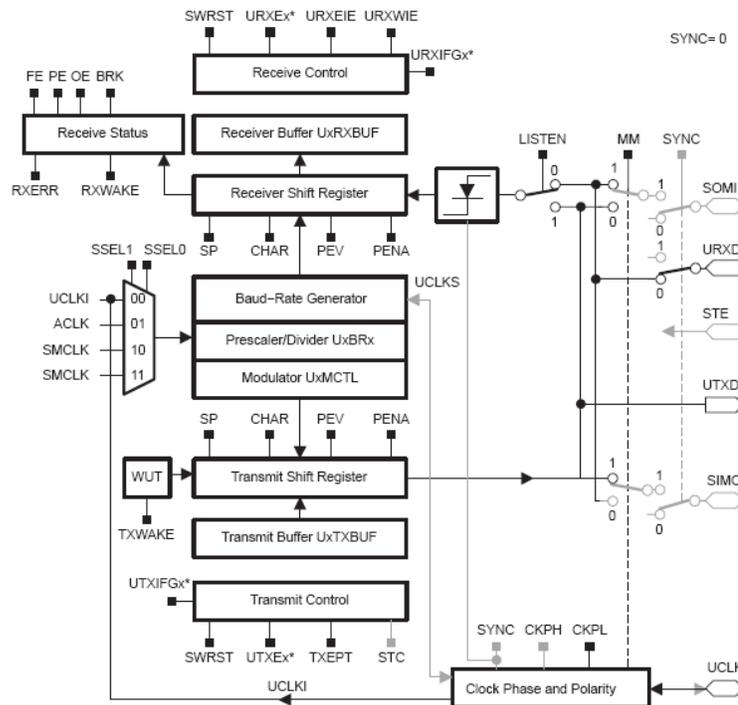


Figure 45: USART Block Diagram UART mode<sup>23</sup>

<sup>23</sup> MSP430x1xx Family User's Guide

In UART mode, the USART transmits and receives characters at a bit rate asynchronous to another device. Timing for each character is based on the selected baud rate of the USART. Transmit and receive functions use the same baud rate frequency. For setting the baud rate we use the Digitally Controlled Oscillator (DCO). For configuring the UART and the DCO we use the following registers.

Table 14: UART Registers configuration

Register	Configurations in our project ( explanation )
DMACTL0	<b>DMA0TSEL_0</b> – Software Trigger
DMA0CTL	<b>DMA0TSEL_0</b> – Single Transfer  <b>DMA0SRCINCR_3</b> – Source Address is Incremented  <b>DMA0SBDB</b> – Source and Destination Bytes  <b>DMA0EN</b> – DMA enable  <b>DMA0DSTINCR_0</b> – Destination address is unchanged

### 3.2.4.3. DAC

Since we want to get the maximum dynamic range of In-Amp, we are doing calibration procedure so that, when there is no tilt, the output corresponds to the middle of the ADC. This is done by using the digital to analog converter (DAC) inside the  $\mu$ Controller. Since the output of the DAC is not low impedance if we want to use the entire possible range, we can buffer it by using the second AD8295 uncommitted Op-Amp.

R <sub>O/P(DAC12)</sub>	Output Resistance (see Figure 22)	R <sub>Load</sub> = 3 k $\Omega$ V <sub>O/P(DAC12)</sub> = 0 V DAC12AMPx = 7 DAC12_xDAT = 0h	2.2V/3V	150	250	$\Omega$
		R <sub>Load</sub> = 3 k $\Omega$ V <sub>O/P(DAC12)</sub> = AV <sub>CC</sub> DAC12AMPx = 7 DAC12_xDAT = 0FFFh	2.2V/3V	150	250	
		R <sub>Load</sub> = 3 k $\Omega$ 0.3 V $\leq$ V <sub>O/P(DAC12)</sub> $\leq$ AV <sub>CC</sub> - 0.3 V DAC12AMPx = 7	2.2V/3V	1	4	

Figure 46: Output Resistance of the Digital to Analog Converter <sup>24</sup>

<sup>24</sup> MSP430x15x, MSP430x16x, MSP430x161x -MIXED SIGNAL MICROCONTROLLER Datasheet



In our case we used  $V_{REF}$  of 2.5V. We also keep the 2 different values for the reference voltages for the different channels. These values are calculated after the user presses the calibration button. Then 20 samples are taken from each channel and the average is calculated. Then the DAC values are calculated such that the instrumentation amplifier has output at the middle of the range of the ADC.

#### ***3.2.4.4. MUX controlling***

In order to control which signal we are reading through the ADC – the X or the Y component of the tilt, we are using two general purpose I/O pins to control the multiplexer. Doing this we can easily associate the current reading with the corresponding channel.<sup>25</sup>

#### ***PCB consideration***

The  $\mu$ Controller has a lot of pins (64) making it relatively hard to layout. A thing to be careful is the relatively high frequency clock (8 MHz). Crosstalk with nearby lines might occur so should be taken into account.

---

<sup>25</sup> C code is attached in the appendix for detailed information

### 3.2.5. USB chipset – FT232RL

One of the project statements was that the board we are designing should communicate to a computer using USB connection. Looking at what was available on the market we found a company called Future Technology Device International (FTDI) which offers a range of products to allow easy interfacing to devices over USB. After researching their products we found one chip (FT232RL) which has all the features we need. FT232RL is a USB to serial UART interface, which means it takes a UART input/output and “translates” it into USB standard having an integrated USB Protocol Engine. Furthermore, it provided an additional feature that moved it above the competitors - it has an integrated 3.3V voltage regulator, which we were in need to power the MSP430F169 since its maximum supply voltage is 3.6V and it couldn’t be powered directly from the USB.

## 2 FT232R Block Diagram

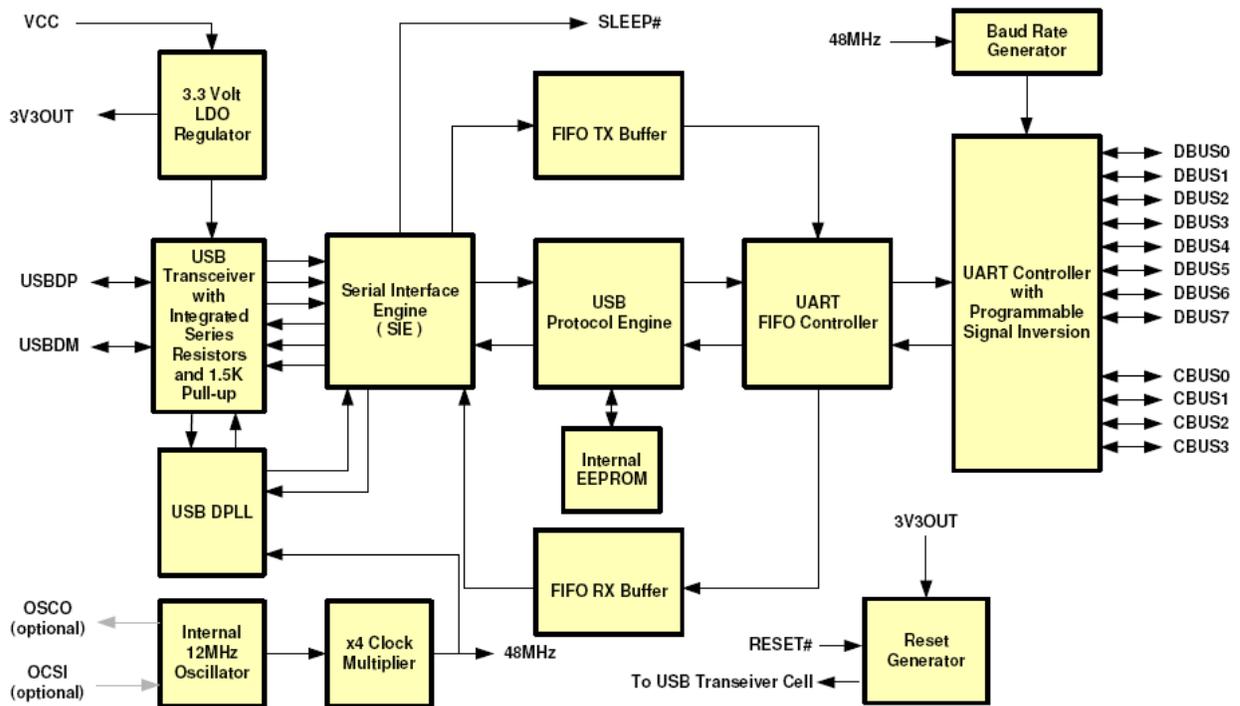


Figure 48: FT232R Block Diagram<sup>26</sup>

This chip does not require any specific configurations. After being connected to the  $\mu$ Controller over transmit receive pair (Tx/Rx) and plugged into a computer it automatically identifies itself under Microsoft Windows VISTA. For

<sup>26</sup> FT232R USB UART IC Datasheet <http://www.ftdichip.com>

Microsoft Windows XP, drivers need to be installed for proper operation, which can be easily found on the webpage of the manufacturer, accompanied by installation guidelines for various Operating Systems. Once the drivers are installed, every time you plug the device into a computer it automatically recognizes it as a serial COM port with a unique number. Each device is pre-programmed with unique USB serial number which the computer associates with a COM port number and every the device is connected it is always given exactly the same COM port.

**PCB considerations**

There are not any specific considerations for this IC. The only suggestion is to keep the USB data traces as symmetrical as possible and as short as possible to the USB connector on the board. Dimensions of the footprints for the different packages (28 pin SSOP and 32pin QFN) are available from the datasheet of the part.

**3.2.6. Charge Pump – Microchip TC7660**

*Motivation*

The AD8295 In-Amp provides the user with the option of using either dual rail or single rail power supply. There are certain advantages of using a dual supply over single supply. Dual supply configuration provides wider input dynamic range than single supply configuration.

The input operating voltage range specified for AD8295 is given below.

**Table 16: Input operating Voltage Range for AD8295**

Test Conditions	Grade A		Grade A		Unit
	Min	Max	Min	Max	
$V_s = \pm 2.3 \text{ V to } \pm 5 \text{ V}$	$-V_s + 1.9$	$+V_s - 1.1$	$-V_s + 1.9$	$+V_s - 1.1$	V

As seen from table 16 the operating voltage range for  $\pm 5\text{v}$  power rail will be higher than a 5v to ground power rail. The  $\pm 5\text{v}$  provides 6.9 volts of range (Grade A), and the 5v to ground power rail provides 2 volts of

range (Grade A). The ADC on the MSP430  $\mu$ Controller has input range of 3.2 volts and in order to take advantage of it we decided to use the dual supply configuration.

To use a dual supply configuration we needed to provide negative 5 volts rail to the chip. Since, the only power source is from a USB which provides us with a positive 5V rail, we had to use a charge pump that would provide the negative 5V supply. We looked at different charge pumps from a wide variety of options available in the market. Some of them provide programmable voltage output. But for simplicity of design we wanted to use one that requires no programming. Also there are others that provide much higher voltage range and output current than what we required. We only needed to supply the AD8295 which has a maximum power supply quiescent current of 2mA.

We wanted to find chips that have both DIP package and a smaller package like SOT or SOIC. We wanted to test with the DIP version on breadboard first, and then transfer the SOIC/SOT package to the PCB. After using these criteria, we were left with few options. From this pool of options we chose the TC7660 charge pump, because of its cost and availability.

### Schematic description

We decided to use TC7660 chip from Microchip for the charge pump.

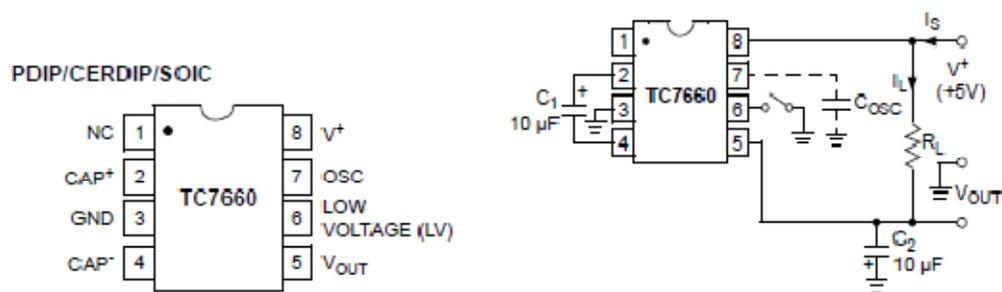


Figure 49: TC7660 charge pump (Left) Configuration for a -5V output (Right)

The TC7660 charge pump comes in both PDIP and SOIC version, and uses only two external capacitors for setting up negative voltage. A negative 5 volts output only requires two external 10 $\mu$ F capacitors.

However, we were concerned whether the charge pump will be able to supply a constant DC-DC conversion, or it will have ripple at the output; and if there is ripple at the output, what will be the effect on the In-Amp performance. To understand these effects we looked at the datasheets of TC7660 and AD8295.

### *Charge pump output Ripple.*

In the datasheet we found a graph of output current vs. output voltage for the TC7660 chip.

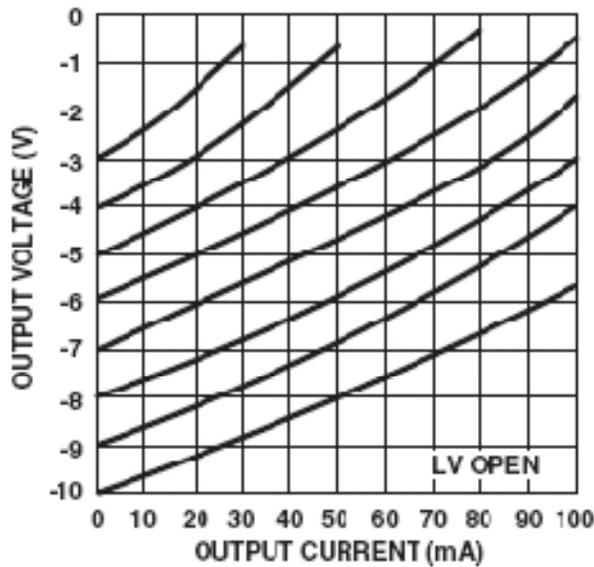


Figure 50: Output current vs. Output voltage

For 2mA of output current, the output voltage is around -4.85 volts for this chip. However, the graph did not specify the output resistance of the charge pump. On the other hand, the datasheet specifies output resistance in room temperature conditions. The maximum output source resistance for this chip in room temperature is 100 Ohms. The switched capacitor inverter can be modeled as a simple circuit shown below.

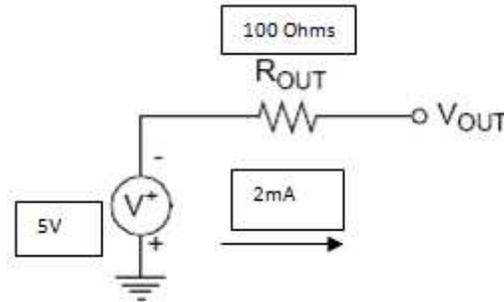


Figure 51: Equivalent circuit model of TC7660

From the AD8295 datasheet we found that the quiescent current drawn by the whole system is 2mA. So, we used the output current quantity of 2mA to calculate the output voltage. With a  $V^+ = 5\text{Volts}$  supply and  $R_{OUT} = 100\text{ Ohm}$  output source resistance, the charge pumps output voltage drops linearly with the current. For a 2mA output current, we can write the following equation:

$$5V - V_{out} = 2mA * 100$$

$$V_{out} = 4.8\text{ V (in the actual circuit it will be } - 4.8\text{ Volts)}$$

### AD8295 performance due to Ripple

To understand the power supply ripple effect on the AD8295, we looked at the PSRR plot of the chip. Power supply ripple rejection ratio (PSRR) is a measure of how well a circuit rejects ripple coming from the input power supply at various frequencies.

$$PSRR = 20 \log \frac{\text{Ripple input}}{\text{Ripple output}}^{27}$$

<sup>27</sup> <http://focus.tij.co.jp/jp/lit/an/slyt202/slyt202.pdf> - Understanding power supply ripple rejection in linear regulators

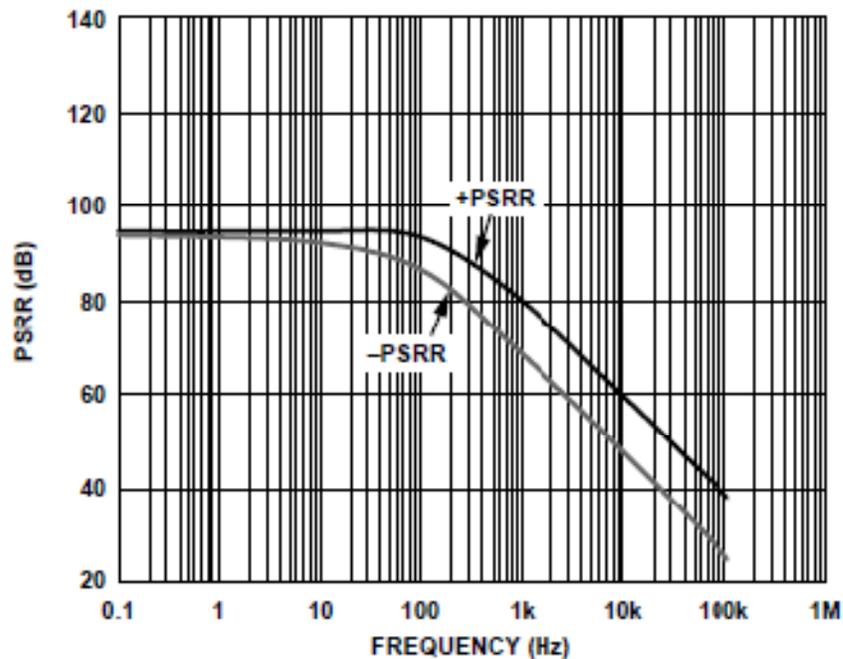


Figure 52: PSRR vs. Frequency for AD8295

From the PSRR graph of AD8295, we can see that the rejection ratio stays high even for a wide variety of frequencies (it is close to 50 db at 10 KHz). So, even though there might be ripple at the output of the charge pump, AD8295 will be able to perform well under those conditions. Therefore, we decided to use TC7660 charge pump for our application.

### PCB Consideration

The charge pump provides the negative power supply to the board. The traces and copper pours carrying current from charge pump to the next components should be made adequately wide to reduce generation of noise or unwanted voltage spikes<sup>28</sup>.

<sup>28</sup> [http://www.national.com/appinfo/power/files/national\\_power\\_designer114.pdf](http://www.national.com/appinfo/power/files/national_power_designer114.pdf) Best Layout Practices for Switching Power Supply.

### 3.2.7. Frequency Compensation

#### Motivation

The response of the tilt sensor is a function of physical properties of enclosed air and sensor electronics. Without any frequency compensation, we can see that the dynamic response of the tilt sensors is slower than desired. If we provide our system with a step function by tilting the prototype board from horizontal to vertical in Y axis, we see that the voltage settling time is approximately 20 seconds i.e. it will take approximately 20 seconds for the tilt sensor to change its output voltage , for some tilt angle variation of the board.

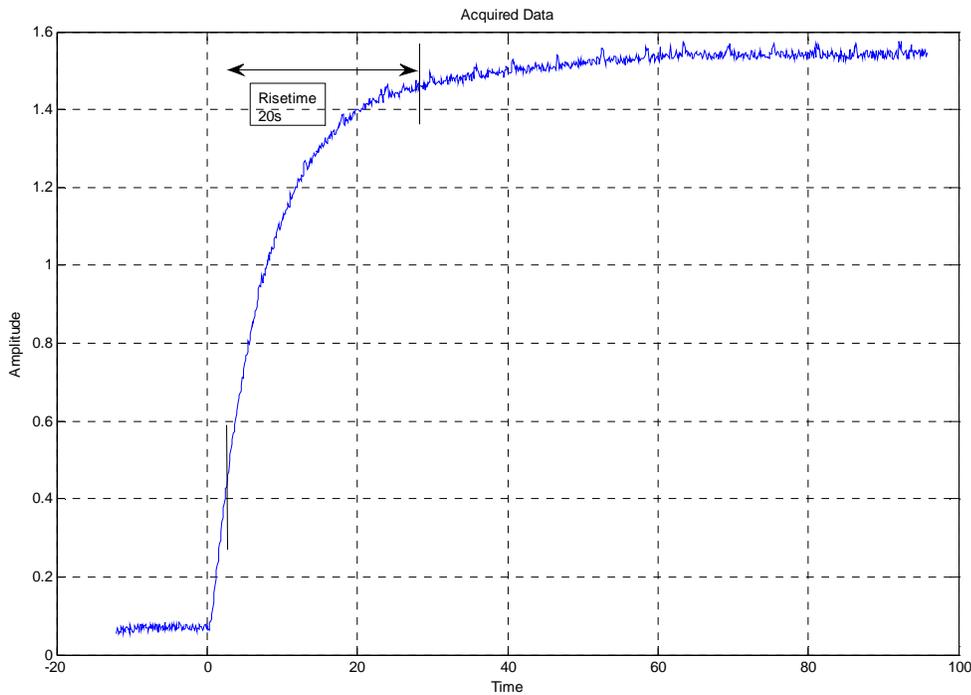


Figure 53: Acquired data showing slow dynamic response

Thus, to make the tilt sensor more sensitive, its dynamic response should be faster. Therefore, we need to decrease the rise time. To accomplish a lower rise time, we can use frequency compensation by relating to the following equation.

Equation 9: Bandwidth – Rise time relationship

$$Rise\ Time\ (s) = \frac{0.35}{Bandwidth} \quad (9)$$

Therefore, assuming that our tilt sensing board is a first order system, we can increase the Bandwidth of the overall system, and thereafter see a faster dynamic response. To achieve this we implemented a high pass filter.

### *Design Detail*

The overall frequency compensation system can be given by the following block diagram.

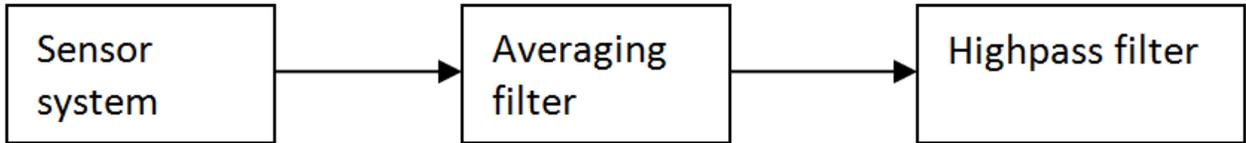


Figure 54: Order of Filtering

First our sensor system is fed through an averaging filter to reduce noise effects, and thereafter a high pass filter is used to increase the bandwidth of the overall system. We used LabView to employ the digital filtering for our system. We used the continuous time domain theoretical models, to find the digital filter (discrete time domain) coefficients for our system.

### *Continuous Time Domain*

#### *Averaging Filter*

The acquired data in an oscilloscope shows that the voltage couples noise from other sources. Thus to limit noise, we implemented an averaging filter. By trial and error we found that an averaging filter of twenty samples sufficiently smoothes the voltage spikes due to noise.

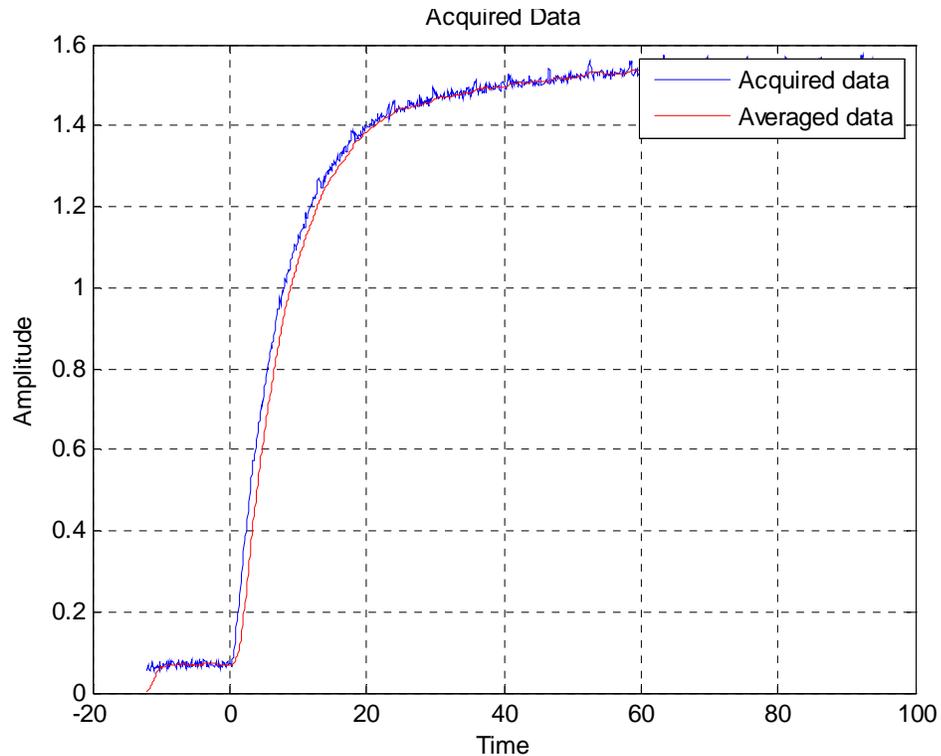


Figure 55: Averaging filter result

The above figure shows the acquired data and the data after twenty sample averaging. As we can see, the spikes in the data points are greatly reduced by the averaging filter.

### *Highpass Filter*

To increase the bandwidth we needed to implement a high pass filter. However, first we had to make a theoretical model of our tilt sensor system before we could proceed with the Highpass filter. We started our filter implementation by assuming that the overall system is a first order system. From the oscilloscope measurement we found the 10-90% rise-time to be 20 seconds.

Thus by implementing the following equation we found the time constant for the transfer function of our system.

Equation 10: Time constant – Rise Time relationship

$$\text{Time constant } (\tau) = \frac{\text{risetime}}{2.2} \quad (10)$$

Transfer function =  $\frac{1}{1+s*\tau}$  for a first order system. However we found that, the transfer function model (Equation 10) does not correspond to the acquired data, unless we use a gain of 1.55. Thus, the final theoretical model for the transfer function of our system is given by the following equation

Equation 11: Transfer function of system

$$\text{Transfer function of system} = \frac{1.55}{1+s*\tau} \quad (11)$$

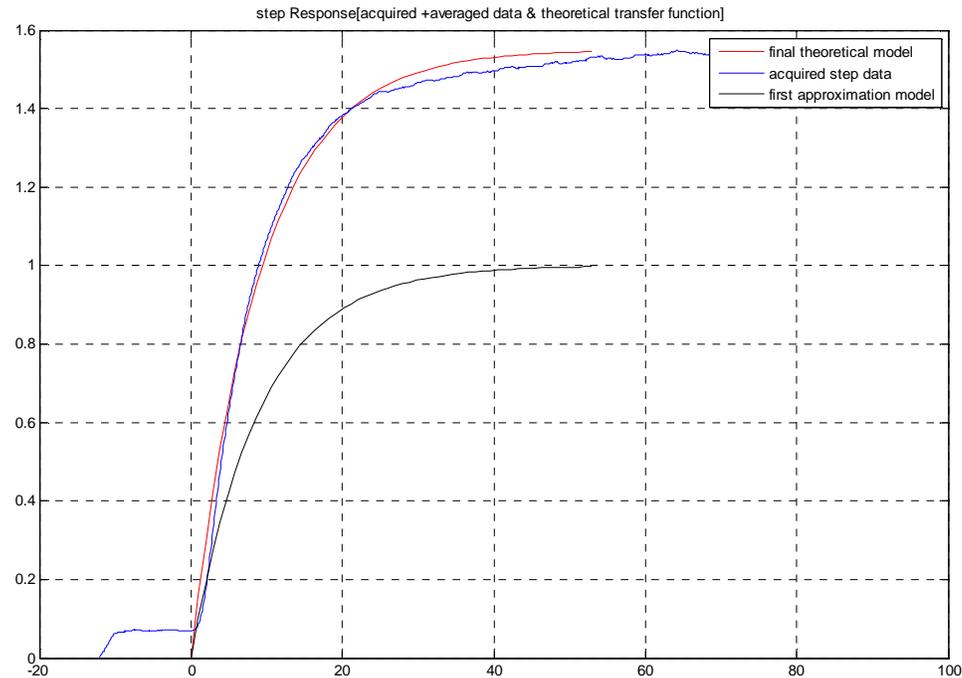


Figure 56: Comparison between theoretical model & acquired data

As we can see if we multiply our initial theoretical model by gain of 1.55, then the theoretical model step response closely corresponds to the acquired data step response. Once we had the good theoretical model to our system, we decided to increase the bandwidth of our system.

From transfer function in equation 11 we can obtain the bode plot for our overall system. The overall system response is marked in blue in the figure bellow. If we make the 3dB point of our system appear much later, then we will be able to increase the bandwidth of our system (marked in red).

To move the pole to higher frequency point, we need to pass our initial sensor system through a highpass filter. The filter (marked by green) has a gain of one, for low frequencies. Since we assumed a first order system we have one pole. The goal is to design the filter such that it will have a zero at the same frequency as the system pole and they will cancel out. The resulting total system is again a first order but has expanded bandwidth since it has a new pole – the one from the highpass filter. In other words for a given system with transfer function of  $\frac{1}{1+s*\tau_1}$ , we create a highpass

filter with the transfer function:  $\frac{1+\tau_1*s}{1+\tau_2*s}$  and the resulting transfer function is the multiplication of these two transfer

$$\text{functions } \frac{1}{1+s*\tau_1} * \frac{1+\tau_1*s}{1+\tau_2*s} = \frac{1}{1+s*\tau_2}$$

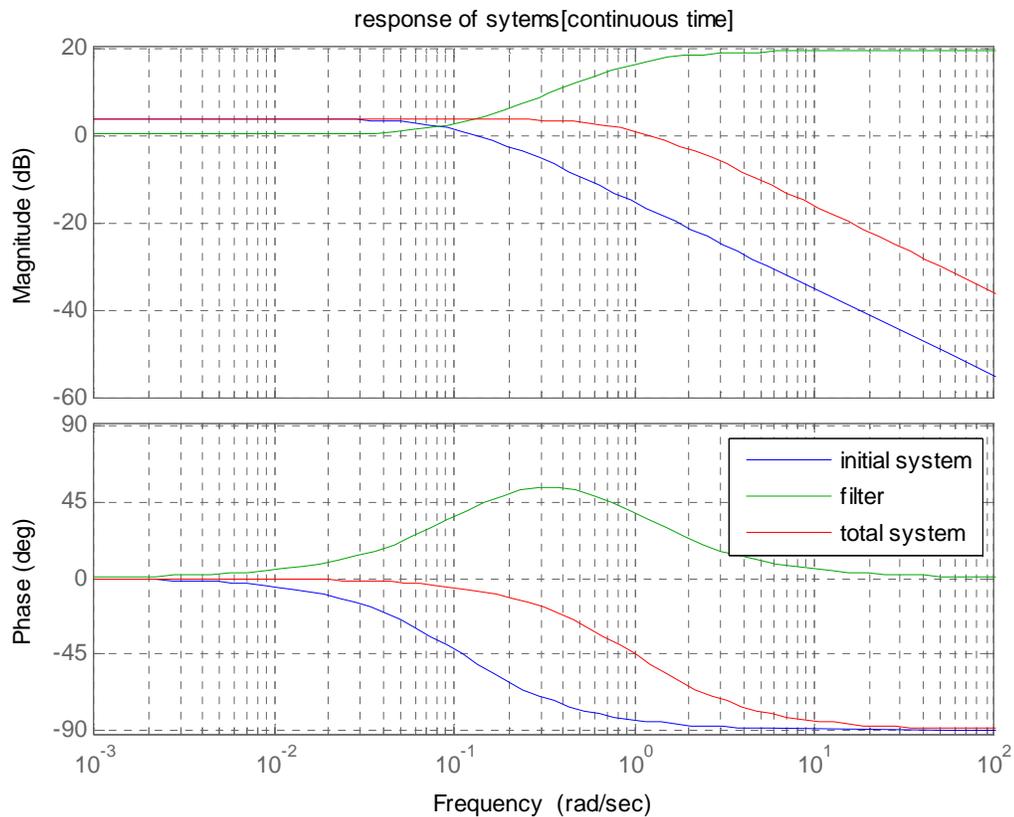


Figure 57: System responses

Once we had the transfer function of our filtered system<sup>29</sup>, we checked our theoretical assumptions by comparing between acquired data step response and filtered data step response.

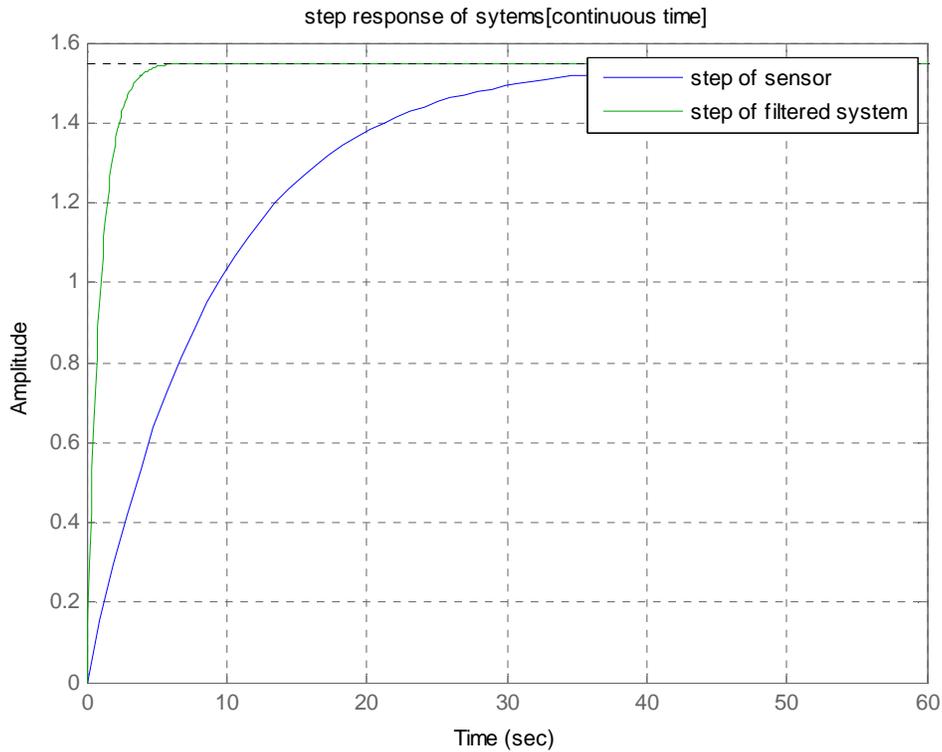


Figure 58: Step response of systems in continuous time

After we implement the high pass filter, we see from the above figure that the dynamic response time is faster than our initial system.

### Discrete Time domain

Since, we intended to use LabView to implement all the filtering methods discussed above; we needed to convert our continuous time domain model to discrete time domain model. We used the bilinear theorem in MATLAB to obtain the Z transform of our filter. The Z transform provided us with the following equation in the Z domain:

<sup>29</sup> The sensor signal passes through an LTI system filter. Thus the total response of the filtered system can be given by multiplying the two LTI systems.

Total system T.F. =sensor system T.F. \* filter system T.F.

[T.F.= Transfer function]

Equation 12: Z transform expression of High pass filter

$$\frac{8.706*Z-8.61}{Z-0.9048}$$

Once we had the Z transform, we converted the equation to a time domain representation of an IIR filter. The following equation provides the time domain representation of the IIR highpass filter of our system:

Equation 13: IIR High Pass filter equation

$$y[n]= 8.706x[n] -8.61x[n-1] +0.9048y[n-1] \tag{13}$$

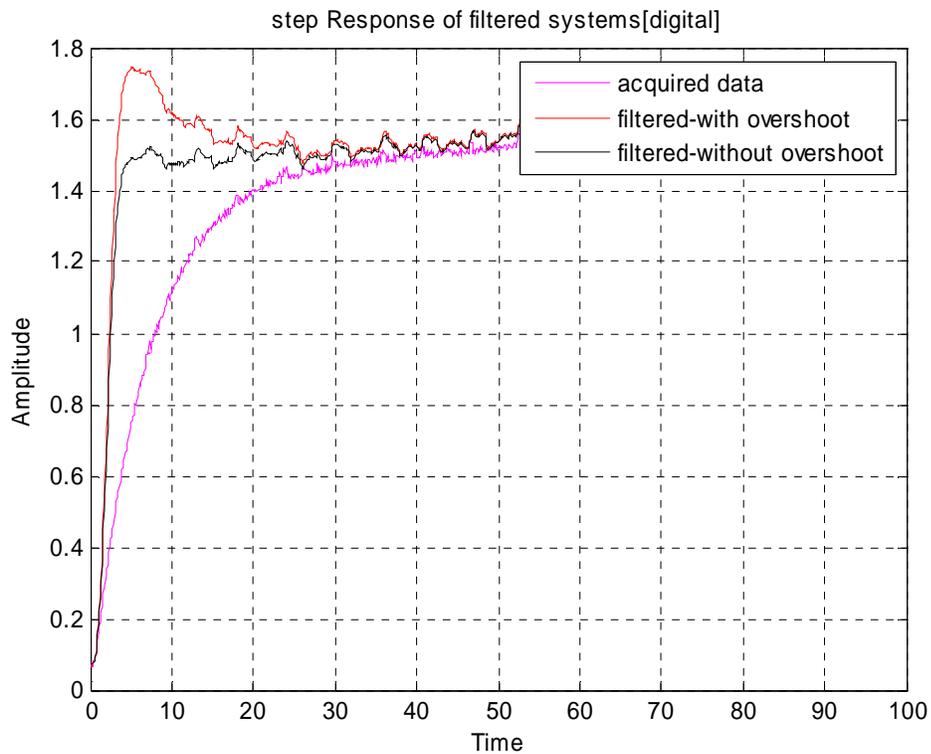


Figure 59: Step response of the continuous system

The overall response of our system in the discrete time domain is shown in the figure above. As we can see that the dynamic response is much faster; however, there is an overshoot before the signal settles towards its final value. Therefore, we changed the time domain equation constants; and found a corresponding response that does not have a spike. In our final LabView implementation, we used the filter equation, which does not have any big overshoot. Our unfiltered systems pole-zero plot is shown in Figure 60. After we introduce the highpass filter, we expect to see the

poles of the initial system canceled by the zero of the filter; and the zero of the initial system being cancelled by the pole of the system.

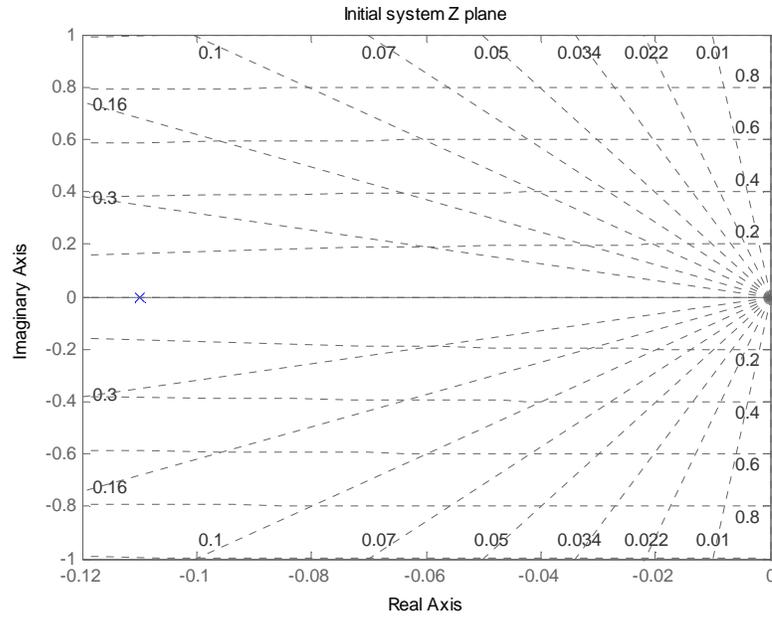


Figure 60: Z plane representation of the initial system

As we can see from the above figure, our initial first order system has only one pole.

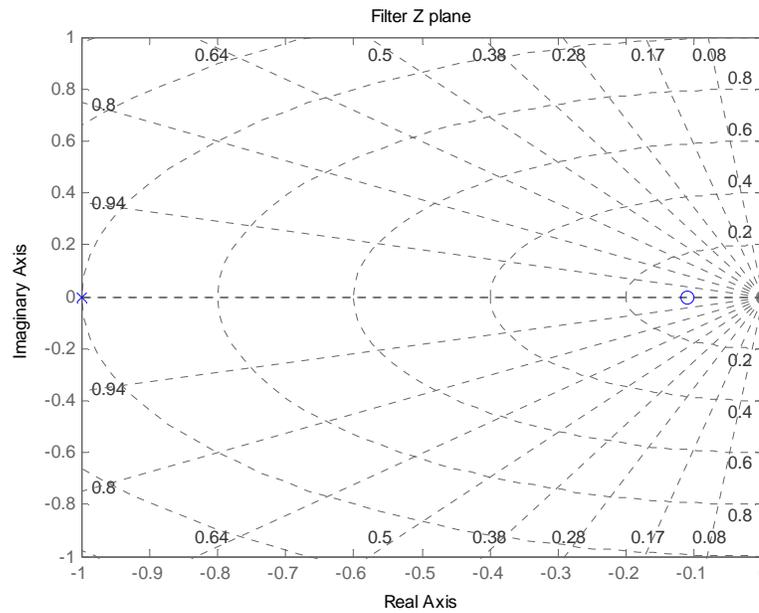
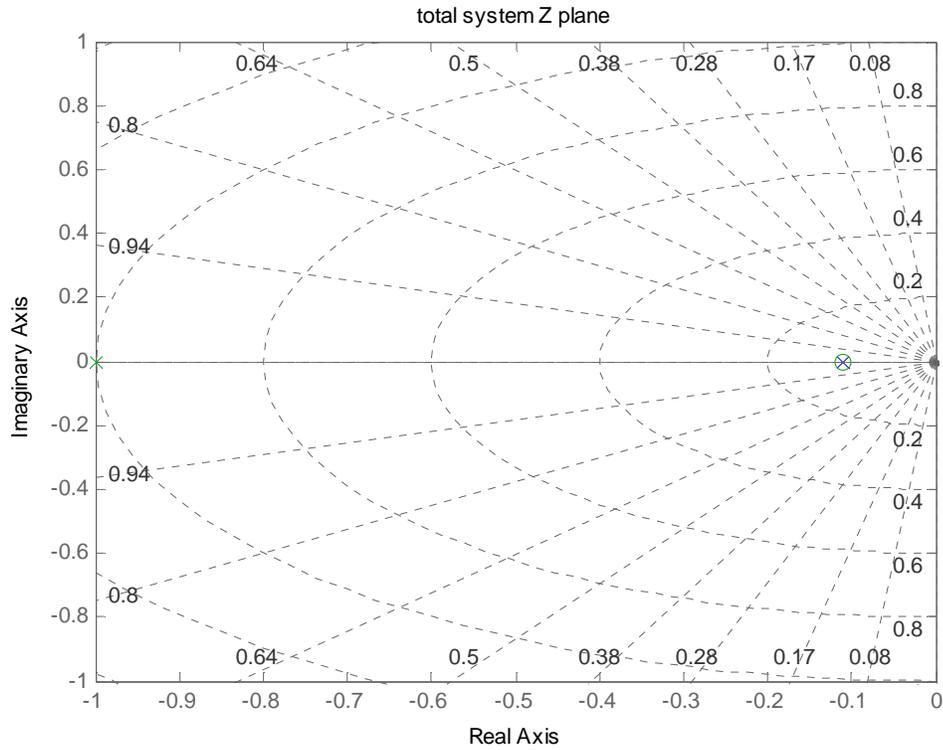


Figure 61: Z plane representation of the filter system

From the filter system pole-zero plot, we see that the highpass filter has one zero and one pole. The zero of the highpass filter system is used to cancel the pole of the initial system.



**Figure 62: Z plane representation of the overall filtered system**

From the above figure we can see the pole of the unfiltered system, is cancelled by the zero of the filter system. And only the pole of the highpass filter is available in the transfer function of the filtered system. This pole extends the bandwidth and increases the dynamic response of the system.

### 3.2.8. Demo Laptop and LabView GUI

#### *Design details for Scope Reading Instrument Control and Data Acquisition*

Creating LabView systems begins with an understanding of PC-based data acquisition. When data is collected from an instrument it passes through three areas of interest. At the lowest level a sensor detects a physical phenomenon and sends it back via a signal. Secondly, device known as a Data Acquisition (DAQ) card takes the signal and converts it into digital data and processed by the computer. Lastly, after the data has been collected, steps can be taken to make an analysis and a meaningful conclusion can be drawn. LabView provides the order and structure for these steps to occur.

For our project, the MSP430 provides the DAQ which converts signal from In-Amp to Digital data. The USB chipset provides a direct path for the data to be transferred back to the PC through the serial port.

LabView has the ability to easily communicate with external instrumentation such as oscilloscopes, microprocessors etc through GPIB, Serial ports. For our project, we decided to use the USB serial port for connecting to the PC, LabView. However, LabView has to recognize the USB communication port, for data transfer. To use serial port with LabView, we need to configure LabView for port communication with VISA layer. National instruments provide DAQ cards that are readily recognized by LabView and do not require any configurations. But if someone is using custom DAQ cards and using serial port to pass the data to PC, the easiest way for LabView to recognize the card, is to use **VISA Configure**.

**Visa Configure** initializes the serial port specified by **VISA resource name** to the specified settings. It tells LabView, to connect to a port specified by **resource name** for instrument control or data acquisition purpose. This control has just one output, the VISA name. Note that for all operations on the port you will need to pass around the

name from one control/function to the next; the name is used from here similarly to a file descriptor in general systems programming.

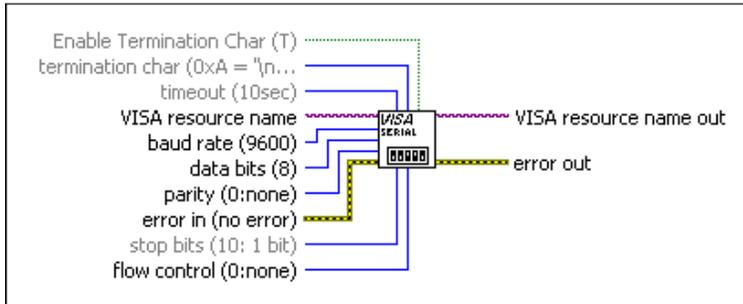


Figure 63: VISA Configure



Figure 64: Front Panel VISA resource name

Once LabView knows, which port to use for data communications, it will need to read the data. This is achieved by using the **VISA Read** Function. In order to read data from an external instrument all that is needed is the command query, telling what to do; a session ID, identifying which device to communicate with (in our the wiring to VISA Configure resource Name provides that information); and error messages, for debugging purposes, a number is connected telling how much data to receive, and a string is returned. All these are connected to the read function symbolized by R, as seen in figure below

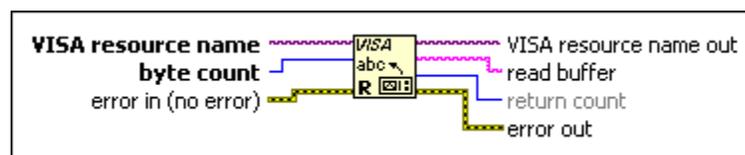


Figure 65: VISA Read

However, the VISA buffer can only hold certain amount of data. After the buffer gets filled , we need to clear/flash the buffer. Thus, we used VISA clear to flash the buffer after every 20 seconds. The Elapsed time is configured so that every 20 seconds it initiates a true case Boolean, which in turn initializes the VISA clear block to flash the VISA buffer.



Figure 66: Elapsed time block



Figure 67: VISA clear

### Processing Data

In our application the data received needs to be processed continuously. Thus, we will need a while loop which continuously reads the data. A While Loop executes its sub-diagram until a Boolean value wired to the conditional terminal is FALSE. LabView checks the conditional terminal value at the end of each iteration. The default value of the conditional terminal is FALSE, so if it is unwired, the loop iterates only once. For our design we wired the loop conditional terminal to the error Bus. So if an error occurs, then the Loop will stop execution.



Figure 68: While Loop

However to process the data we need more functional blocks. The First block inside the While Loop Sub-diagram is **Search/Split string**.

For our project, a typical data set that we get from the MSP430 would be `_23451234_45891237_;` where the first four characters after the Space are X axis Data, and the next four are Y axis Data. To process the data any further we need to split this string into two parts, one containing the X axis data and the other containing the Y axis data. Thus we use the Search/Split String function.

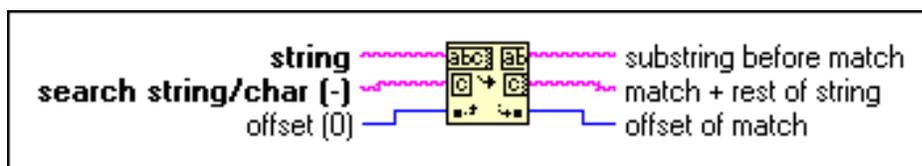


Figure 69: Search/Split String

Once we have the X axis and Y axis data, we need to perform two operations.

1. We have to convert the data to a voltage measurement for X axis and Y axis.
2. We have to convert the data to a tilt angle measurement for X axis and Y axis.

Since, we will be doing Voltage and angle calculation for both X axis and Y axis; we can use the same calculation functionality for both cases. Here, we can use the concept of a Sub-VI. You can easily create a subVI that can be reused many times in other VIs requiring the same functionality; editing the subVI one time updates its behavior for all VIs that use the subVI, which simplifies program design and maintenance. Moreover, the subVI forms the basic method by which you can create a hierarchical program.

### Voltage subVI

The Voltage subVI uses the string data and converts it to numerical data. Thereafter the subVI calculates the voltage corresponding to the data and provides a voltage array. The voltage data can be used to show the current voltage; whereas the voltage array can be used to show change of voltage over time. As an input the subVI only takes buffer from the scope\_reading.VI.



Figure 70: Voltage SubVI

The overall block-diagram of the Voltage subVI is given below.

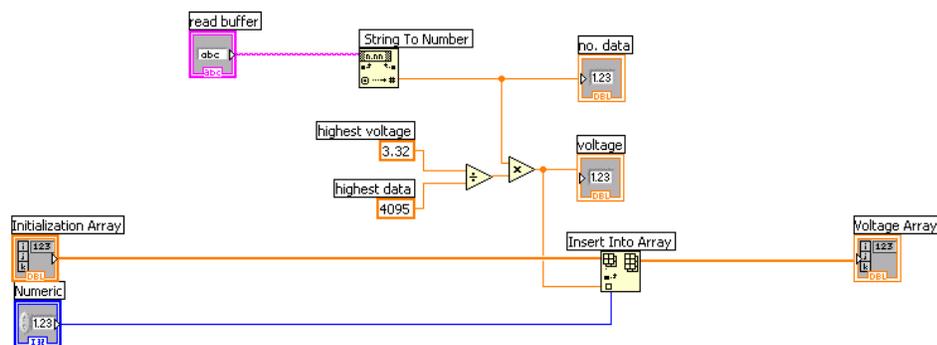


Figure 71: Voltage subVI block Diagram

As can be seen from the block-diagram the subVI has few different functions. The first of these functions is **String to Number**. For LabView to process the data, we must convert the String data to Numerical data. So we use, the LabView inbuilt function **String to Number** which converts the characters 0 through 9 in **string to a floating point Number**.

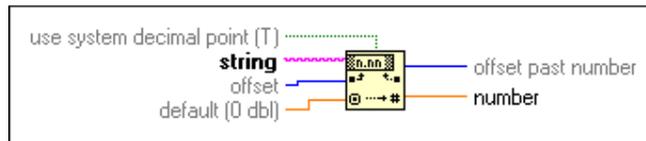


Figure 72: String to Number

Once we have the numerical data, we need to do calculations on the data to output a voltage value. The highest numerical data point that we get from the MSP430 is 4095, which also corresponds to 3.32 volts. So each data point corresponds to  $3.32/4095 = 0.00081074481$  Volts. If we multiply this resolution with the acquired numerical data, we can convert the acquired data to voltage.

To help us in completing the conversion, we can use the inbuilt **Multiply** and **Divide** function of LabView.

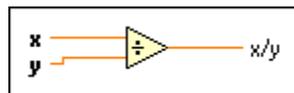


Figure 73: Divide Function

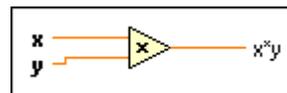


Figure 74: Multiply function

Once we have completed the conversion of Voltage, we need to store the values in an array for future use for the main VI (Scope\_reading.VI).

The **Insert into Array** function in LabView provides us with the option to store values in an array. However, we must initialize the array to an empty array first. This initialization is done in the main scope\_reading.VI with an empty array constant.

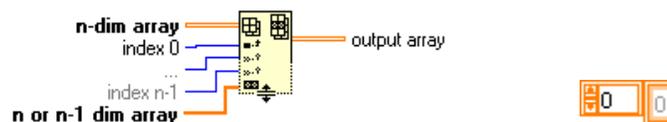


Figure 75: Insert into Array (Left) Empty Array (Right)

## Angle subVI

The Angle subVI uses the string data and converts it to numerical data. Thereafter, the subVI calculates the angle corresponding to the data and provides an angle array. As an input the subVI only takes buffer from the scop\_reading.VI.



Figure 76: Angle SubVI

The overall block-diagram of the Angle subVI is given below.

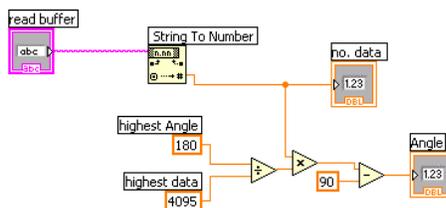


Figure 77: Angle SubVI block diagram

As mentioned above in the Voltage subVI, we use the **String to Number** function to convert the string to a numerical value. Once we have the Numerical data, we need to do calculations to output an angle value. Our board can measure a tilt of +90 to -90 a total of 180 degrees of tilt. The highest numerical data point that we get from the MSP430 is 4095. So each data point corresponds to  $180/4095 = 0.00081074481$  Volts. However, a data point of 4095 signifies an angle tilt of +90 degrees. If we multiply this resolution with the acquired numerical data and subtract -90 degrees, we can convert the acquired data to tilt angle. We use **Multiply**, **Division** and **Subtract** block to perform the pertaining calculations for the Tilt angle measurement.

## Frequency compensation

As discussed earlier in the frequency compensation section, we needed to incorporate filters in our LabView demo to see faster dynamic response. First, we incorporated the averaging filter to limit noise, in our system. Thereafter we created an IIR highpass filter to increase the bandwidth of our system.

## Averaging Filter

The average filter time domain expression is given by the following equation:

Equation 14: Running average equation

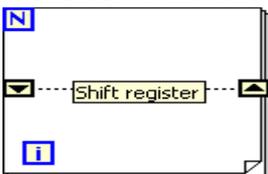
$$y(n) = \frac{1}{20} x(n) + \frac{1}{20} x + \dots \dots \dots \frac{1}{20} x(n - 19) \quad (14)$$

by inspecting the equation, we can see that to process the incoming voltage array data in a loop to do all the subtraction and addition. We take the voltage array from **voltage SubVI**, and process the array data in a for loop. Inside the **for loop** the counter is denoted by **i**, and the termination count is denoted by **N**.

The **array size** block determines the size of the incoming block. We use a subtract block with the array size block to determine the index position of previous sample. After we determine the index position of previous sample, we use the while loop to go through the **Index array** block and retrieves one element of the array at a time. We use the addition block in conjunction with the shift register<sup>30</sup> block to add up the average of twenty samples.

---

<sup>30</sup> For loop structures can have terminals called shift registers that one can use for passing data from the current iteration to the next iteration. Shift registers are local variables that feed forward or transfer values from the completion of one iteration to the beginning of the next iteration.



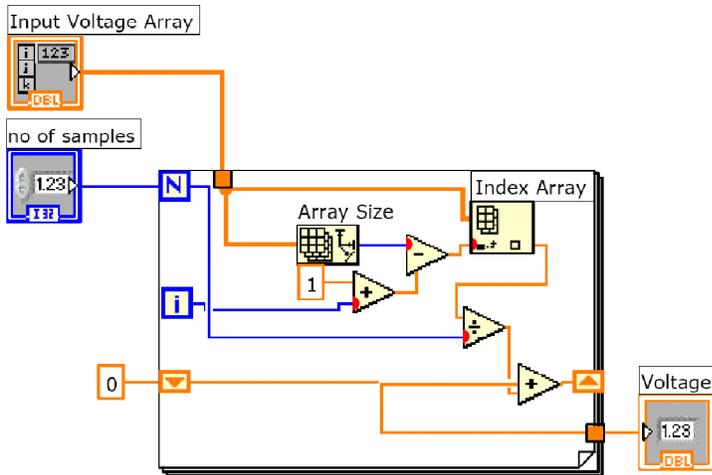


Figure 78: Averaging Filter



Figure 79: Averaging Filter SubVI

### IIR Highpass Filter

The IIR Highpass Filter expression is given by the following equation:

$$y[n] = 8.706x[n] - 8.61x[n - 1] + 0.9048y[n - 1]$$

A simplified block diagram of the filter is given below.

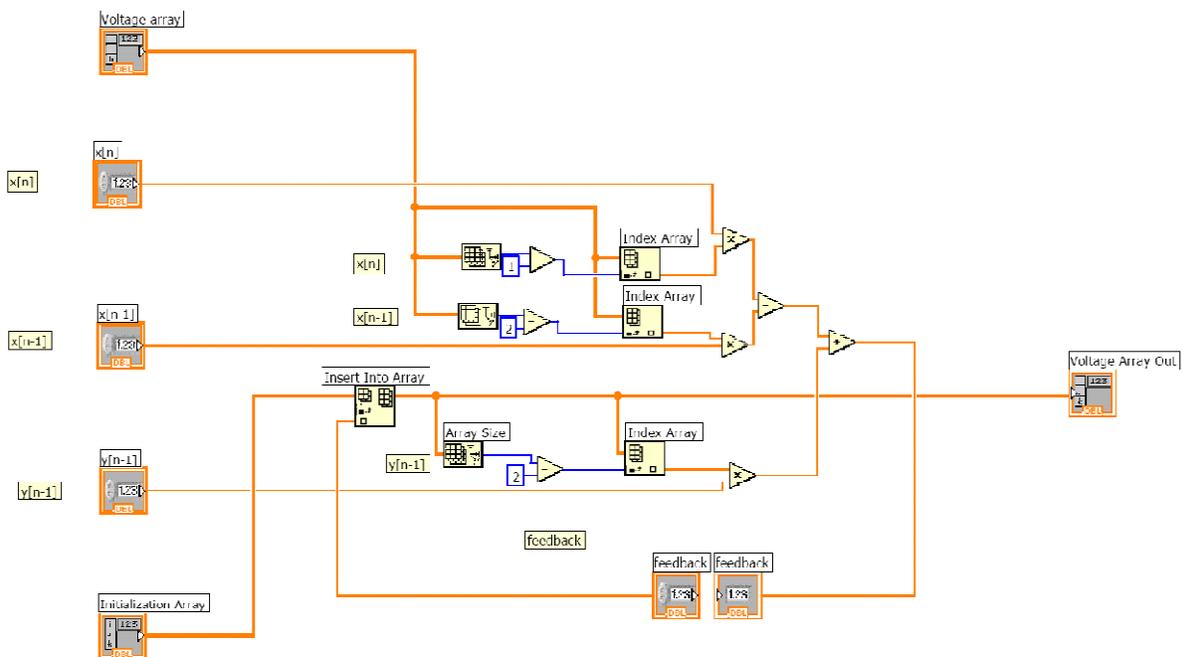


Figure 80: IIR filter block Diagram

As we have seen in the averaging filter, we use a combination of mathematical block, Index array and array size block to implement the IIR filter. We use the array size block to determine the size of the incoming array, and subtract one and two elements to determine index position of  $x[n]$  or  $x[n-1]$ . Thereafter, the index array block is used to retrieve element from the array. For the  $y[n-1]$  element it is fed back into the  $y[n]$  array, and therefore there is a feedback loop created (shown by the feedback blocks in the above figure).

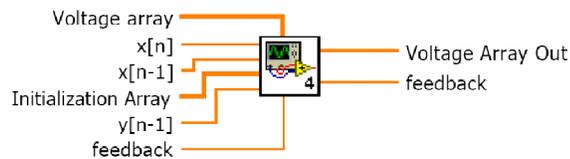


Figure 81: IIR Highpass Filter SubVI

### Displaying Data

Once we have calculated the voltage and tilt angle, we need to show the measurements graphically on the front panel. We use the chart indicator to show voltage measurements and the gauge indicator to show angle measurements.

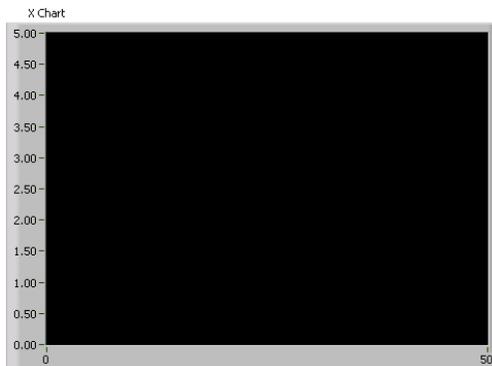


Figure 82: Voltage Measurement Graph



Figure 83: Angle Measurement

## Design Details for Maze Game

As an interactive demo for the application board, we decided to build a tilt sensor based game in LabView. The user moves the ball inside branching passages in the laptop screen to go to the rightmost corner of the screen. The ball movement is controlled by the tilt movement of our board.

The LabView game code can be broken down to the following parts:

1. Instrument Control and Data Acquisition
2. Processing Data
3. Frequency compensation
4. The game graphics
5. Ball movement rules
6. The game rules

The first three parts were implemented the same way, as discussed earlier in the scope reading section. The game graphics and game rules are discussed below.

## Game Graphics

It was essential that we create a 3D display for the game. Thus, we relied on the LabView 3D picture control VIs to provide us with a 3D display.

We needed to create a ground board, for the 3D display. LabView provides a **Create box** VI. We used this VI to create a board/ ground for the maze game with length of 180 units, width of 180 units and height of 1 unit. Thereafter, we used the **Create object** VI and added the previously created box as an object to the 3D display.

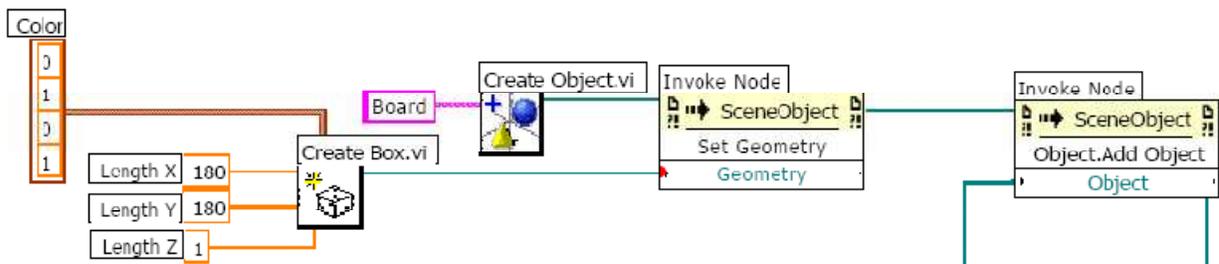


Figure 84: Sub-block diagram for creating a 3D board

In the same way we created the walls for the maze and created our own subVI called the **create Wall.VI**. However, we needed to move the walls in different positions. Otherwise all the walls would show up in the same center position (0, 0, 0) in the 3D display.

Thus, we used the Set translation to move the walls to our desired position. The set translation package clears any translations previously applied to an object in a 3D scene and translates the object by the vector you specify. This block performs an absolute translation from the object's initial position. The Set translation block can be found under set axis & boundary VI.

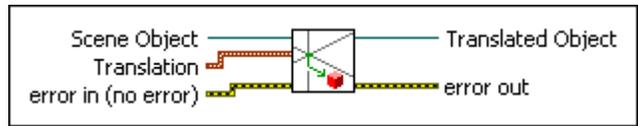


Figure 85: Set translation block

### Game rules

The ball has to move through different passages in the maze screen. However, if the ball movement causes the ball to hit a wall, we must make sure that the ball movement does not get updates. Another way of saying this would be, we should make sure that the ball does not go through a wall or outside the maze board. A block diagram of how we implemented these rules is given below.

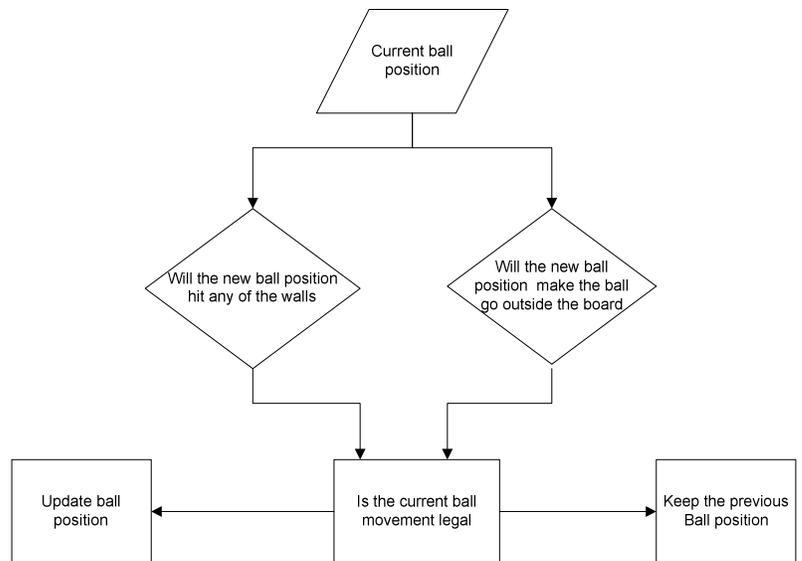


Figure 86: Block diagram for the implementation of game rules

If the current ball position causes the ball to hit a wall or go outside the board, we do not update the ball movement. To determine whether the ball has hit one of the walls, we needed to know the boundaries of the wall. Each wall has a certain length and width, and is centered at some unique co-ordinate position. For example, in the case of wall 1 length of 8 units, and is centered at 20 units. Therefore, the upper boundary for x position for wall 1 is at  $20+8/2=24$  and lower boundary is at  $20-8/2=16$ . We used the expression node to do these calculations. After we found the upper and lower boundary of each wall, we stored those values in an array.

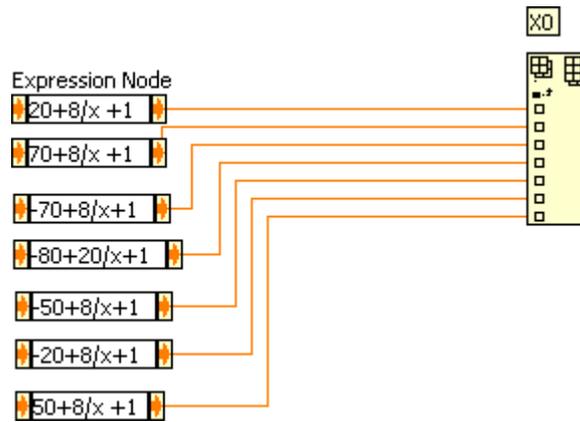


Figure 87: The x axis upper bound array

From the extract one element of the boundary arrays, and check whether the ball is within one of these walls. If the ball position is within one of these wall boundaries, then we know we cannot update the ball's current position. We use the within wall subVI to check whether the ball is within a wall boundary.

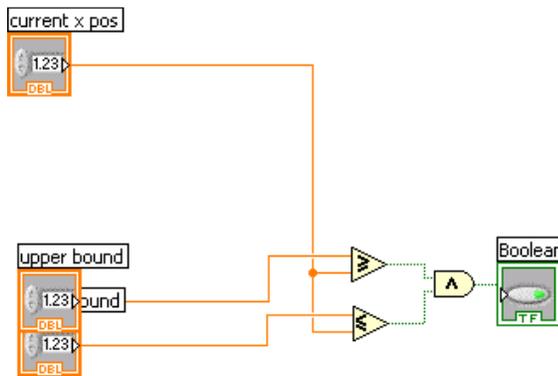


Figure 88: x within wall subVI

In above figure, we can see that one of the wall boundary value(current x pos)is compared to upper bound and lower bound values of the walls. At the end of this sub-block , it is determined whether the current ball position will intersect any of the walls.

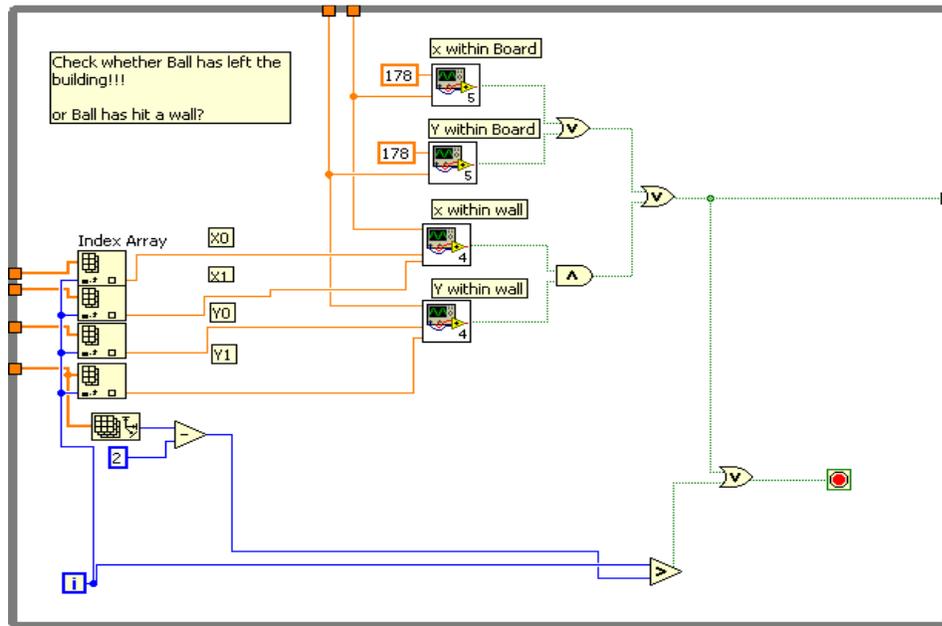


Figure 89: check whether the Ball movement is legal

In the same way, it is determined whether the current ball position will make the ball move outside the board. If either the ball position causes the ball to move out of the board or hit any of the maze walls, the current ball position will not be updated. In the figure above, we extract one boundary value at a time from the boundary array (x0,x1,y0,y1) and check whether the ball position causes the ball to move inside a wall or outside the board. If so, the while loop is ended and the ball position is not updated.

### *Ball movement rules*

The ball moves with the tilting of the board. However, instead of making the ball movement a discrete case, we wanted to make the ball movement continuous. Thus we implemented simple physics rules to attain continuous update to ball movement.

Current position  $x = \frac{at^2}{2} + vt + x_0$  (where  $x_0$  = previous position)

Current Velocity  $v = v_0 + at$  (where  $v_0$  = previous velocity)

Time  $t = \frac{1}{f_s}$

Acceleration  $a = \sin(\alpha) * g$   
 $\alpha$  = tilt angle of the board.  
 $g$  = gain, in this case.

Thus, the ball moves faster with a higher tilt angle, and the ball movement update happens in a more continuous mode.

## 4. Measurements and Results

Once we had the hardware working, we tested our design to determine if it was fully functional or not. While some of these tests were discrete in nature - checking for component connectivity after soldering and verifying that all IC's were powered up right, others were involved data collection to verify 'how well' the device performed a given function. In this chapter we discuss the results of these tests and describe the procedures in which these tests were performed. The following sections include test results for the noise, step-response, and angular response of our system. It also contains an evaluation of the sampling and calibration schemes implemented by our microcontroller.

### 4.1 Sensor Measurements

As discussed previously in the Design Details section, the process of developing the sensor involved several revisions of the layout and construction of the sensor's physical components. To determine how well a particular revision worked, we tested it for mainly two parameters – its noise performance and its step response. Once we arrived at our final sensor design, we also characterized the angular response of the sensor (for 10 degree steps from  $-90^\circ$  to  $90^\circ$ ). These measurements were used by us as yardsticks to determine how well our sensor performed.



Figure 90: Our Final Sensor Assembly

### 4.1.1 Noise Characterization

The noise measurement was performed to determine the variation in the output voltage of the sensor with the sensor lying flat on a horizontal surface. Ideally, this situation should correspond to the output voltage being constant at the “mid-range” of our ADC Full Scale Range. However, this is not the case mainly due to the response of the sensor to air-currents external to the sensor’s air-cavity. For the purpose of characterizing the noise of our final system, we placed our board on a flat surface, powered it up via USB, and collected output data for both the X and Y axis through the computer’s Hyper-Terminal for a period of about 20 minutes. This data was then used to determine the noise as a function of time, and as a function of frequency. The data was plotted in MATLAB, and the scripts used to produce these plots are located in the appendix.

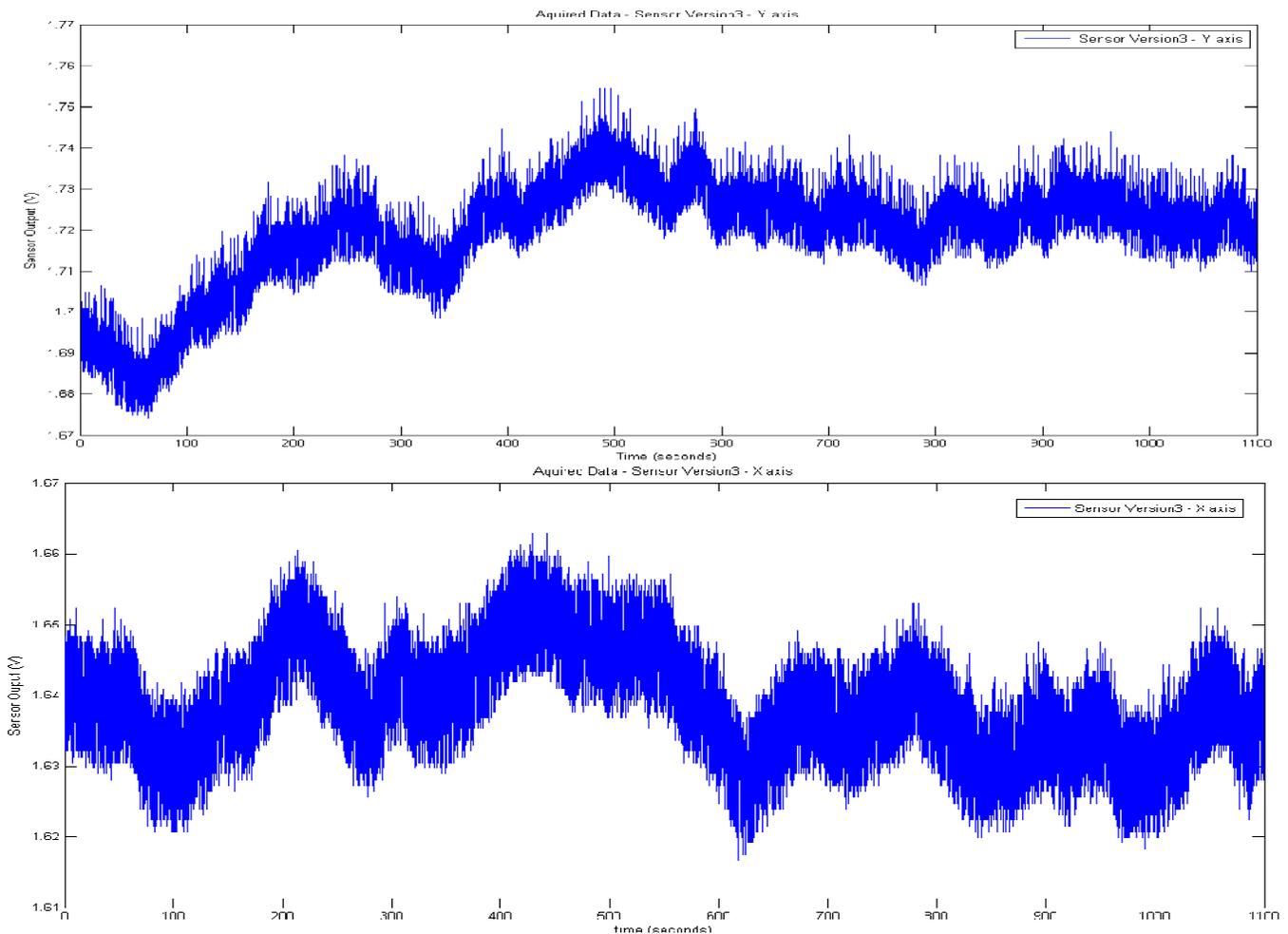


Figure 91: Sensor Version3 X-axis Noise (Top) and Y-axis Noise (Bottom)

For the X-axis, the noise range (max-min) was 42.6mV. The standard deviation was calculated at 8.5mV.

For the Y-axis, the noise range (max-min) was 80.6mV. The standard deviation was calculated at 13.4mV.

In order to study the noise components of our system with respect to frequency, we generated the following single-sided amplitude spectral plots from the same set of collected data. The MATLAB scripts for this process are located in the appendix.

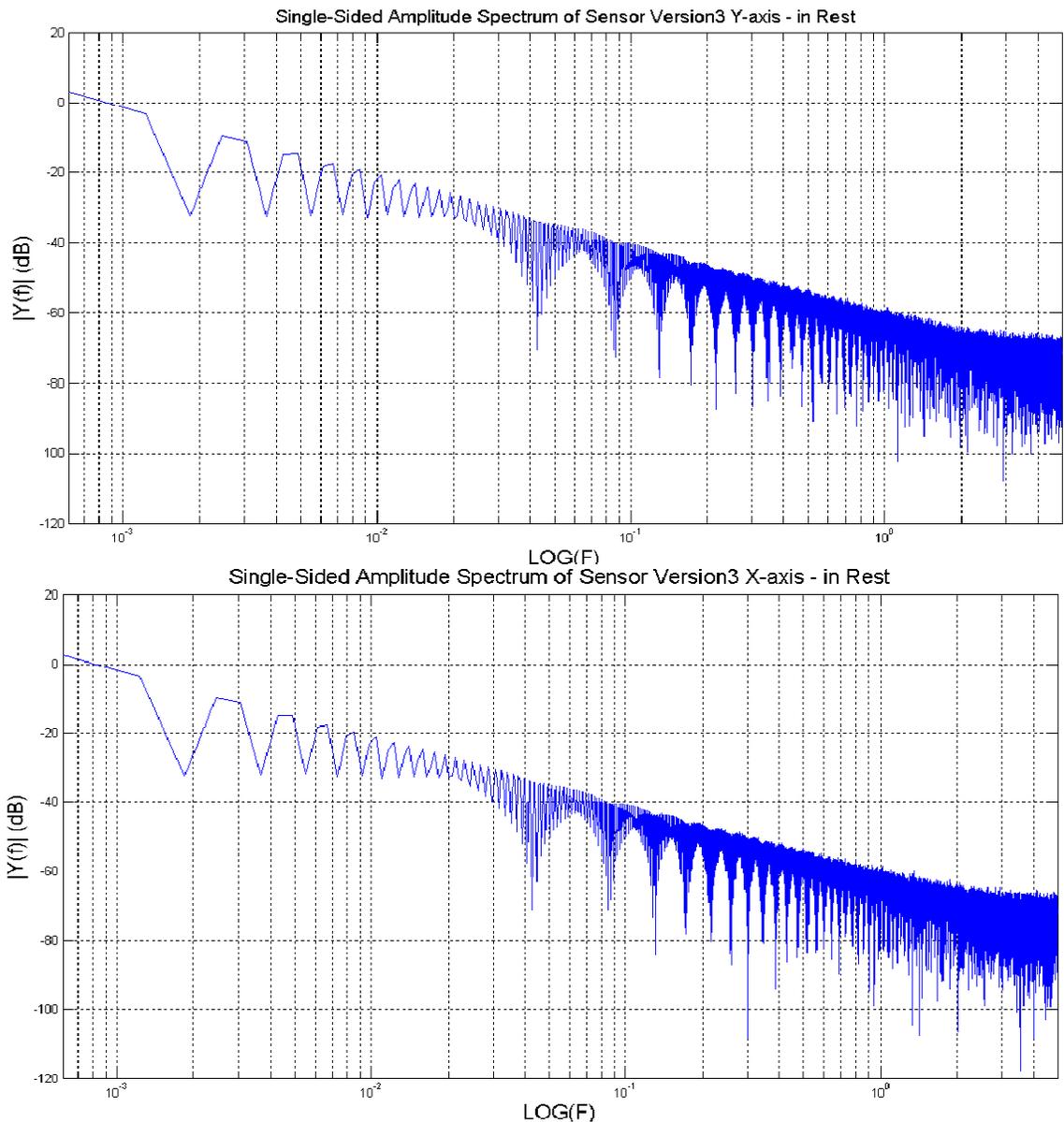


Figure 92: Sensor Version3 - Y-axis noise spectrum (Top) and X-axis noise spectrum (Bottom)

### 4.1.3 Step Response

To measure the response of our system to angular motion, we characterized its response to a 90° step (from a horizontal position to a vertical position).

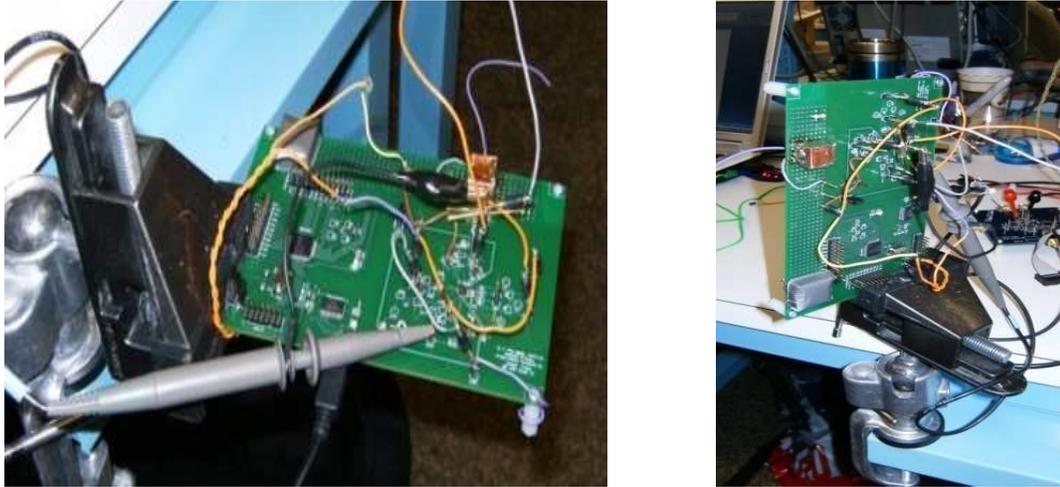


Figure 93: Figure illustrating 90 degree step

We collected this data through the USB port via the LabView “Scope Mode” Program, with the intention of capturing sensor outputs both before and after filtering. This data is shown on the next page for the X axis.

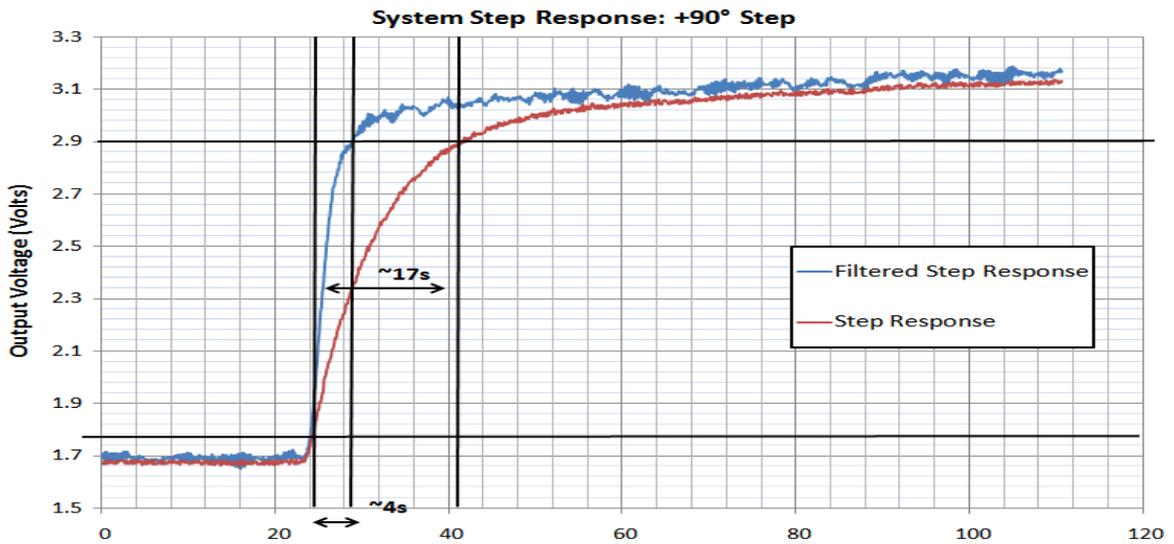


Figure 94: Step Response, X axis, +90 step

As shown in the figure above. The rise-time (10% - 90% of FSR) is about 17 seconds for the unfiltered and about 4 seconds for the filtered output.

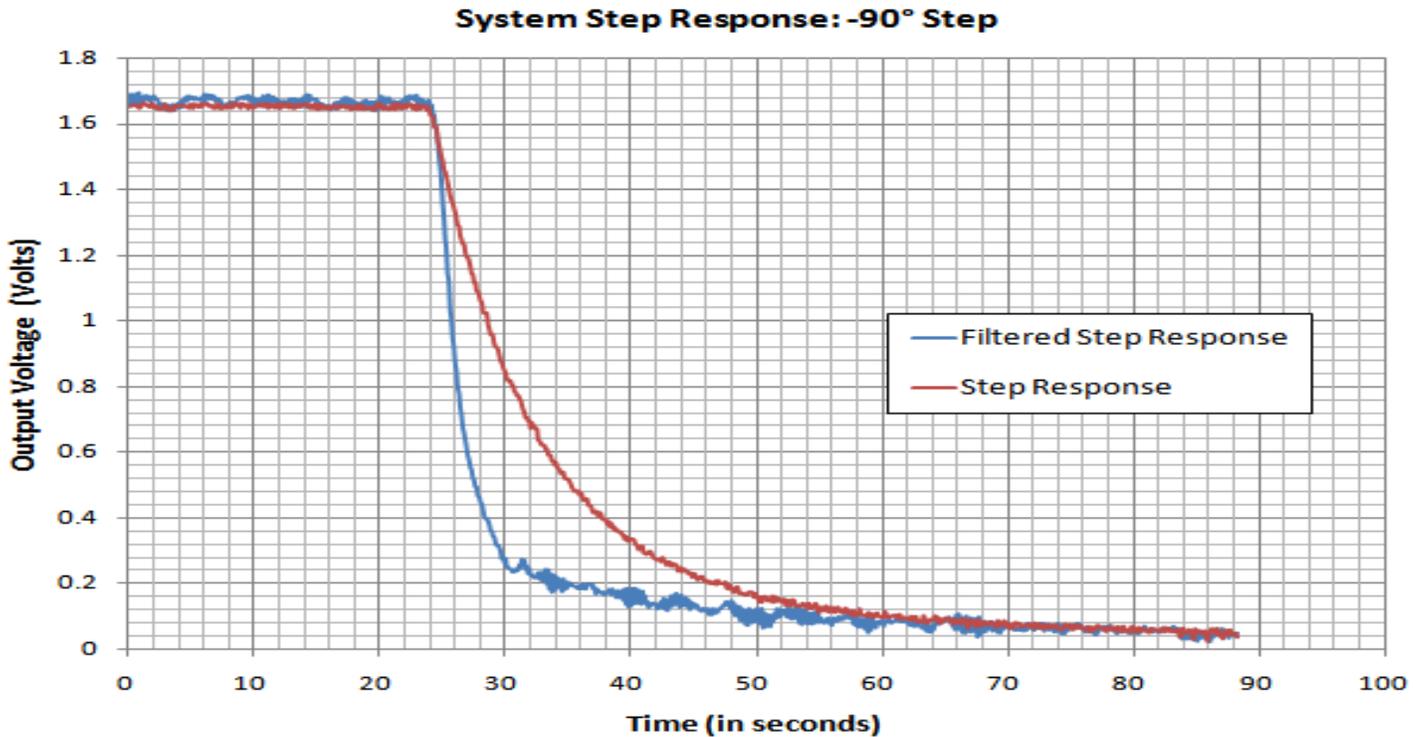


Figure 95: Step Response, X axis, -90 step

Similarly, data was collected for a -90 degree step. The figure above shows both the filtered and unfiltered output.

### 4.1.3 Sensor Angle Characterization

In order to obtain a better measurement for the sensors response to angular motion, we conducted a test similar to the step-response, but with much smaller angle variation of 10 degree steps. Again, this data was collected directly from the digital serial output of our system, via the USB port. Both the filtered and unfiltered outputs were captured. To provide these 10 degree angular steps, the board was set-up using a metal clamp fixture as shown below. For measurements along the orthogonal axis, the board was simply attached along its length as opposed to its breadth

(shown below). To produce an angular step, we simply rotated the set-up till the pointer indicated a ten degree shift along the protractor (shown below).



Figure 96: Illustrating Angle Characterization Procedure

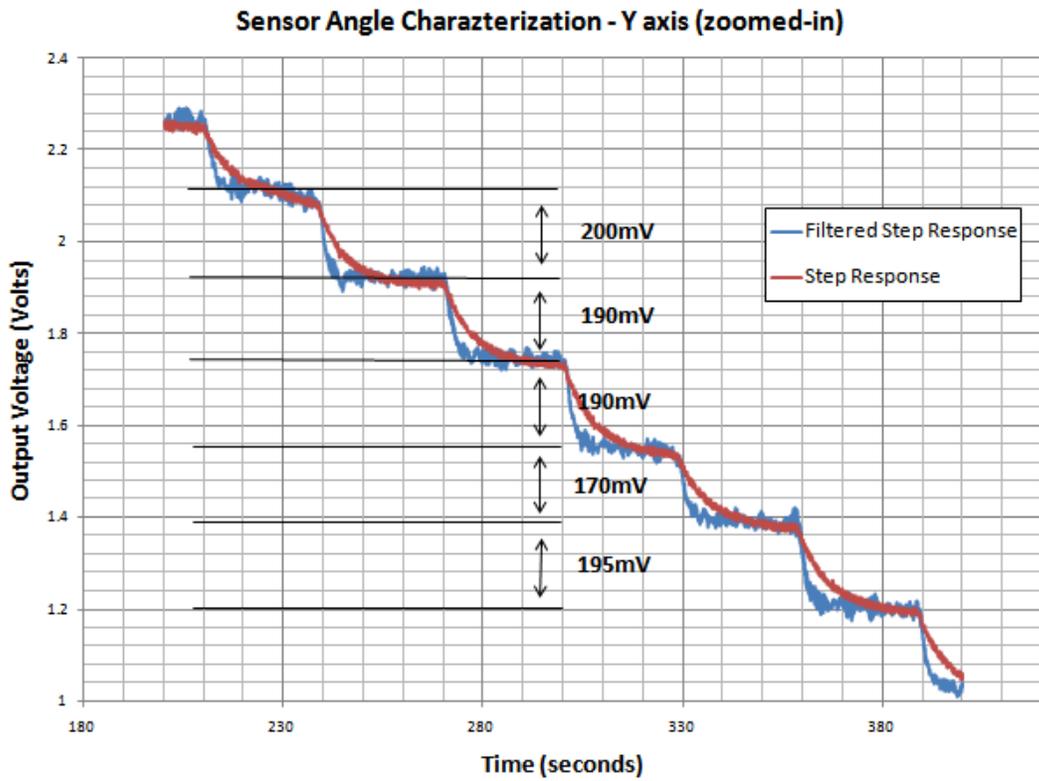
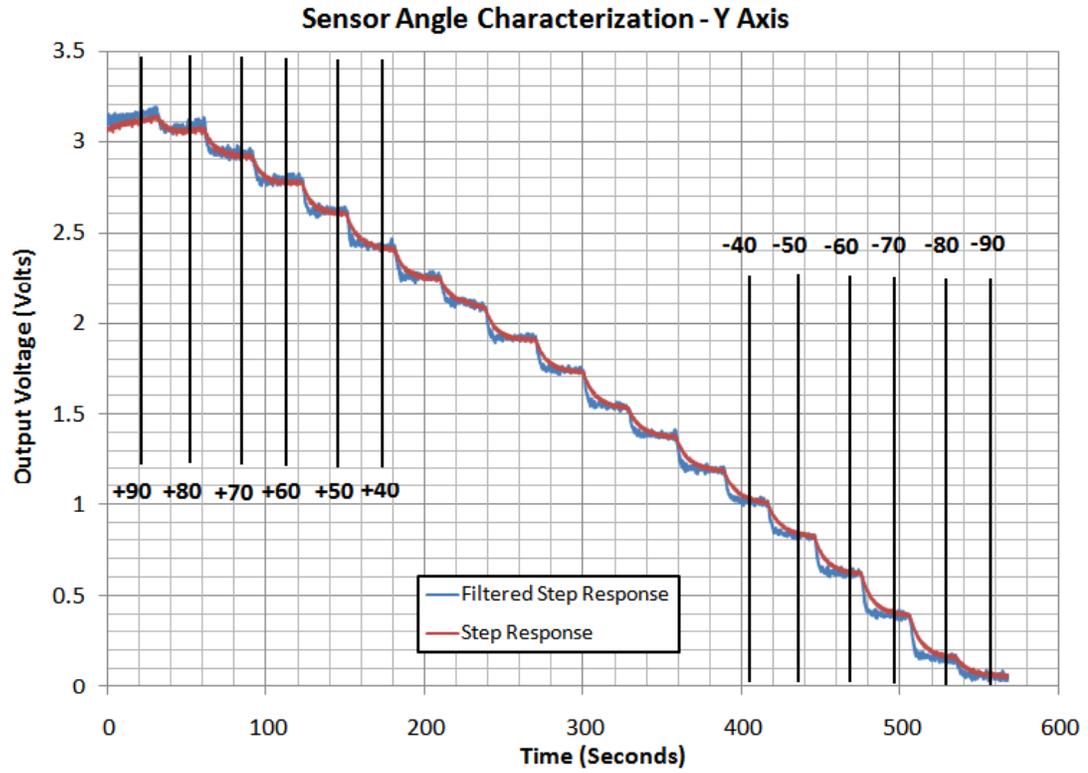


Figure 97: Y-axis Angular Response (Top), Y-axis Angular Response Zoomed-in (Bottom)

From the above graphs, we were able to determine that for most of the sensor output range, 10 degrees of angular movement corresponds to about 190 mV of output signal. This figure is an estimate based on the sensor performance in the  $\pm 60$  degrees region. However, the sensitivity of the sensor reduces for angle in approaching positive or negative 90 degrees. This is inherent of the sensor design. Data analysis from MEMSIC suggests that the sensor response is an arc tan function of angle. Although we did not have sufficient accuracy and resolution in our measurement to verify this, we did see that the step size decreased for the step corresponding to  $+90$  degrees and  $-90$  degrees. Shown below is the same graph for the X-axis sensor. This produced similar results as well.

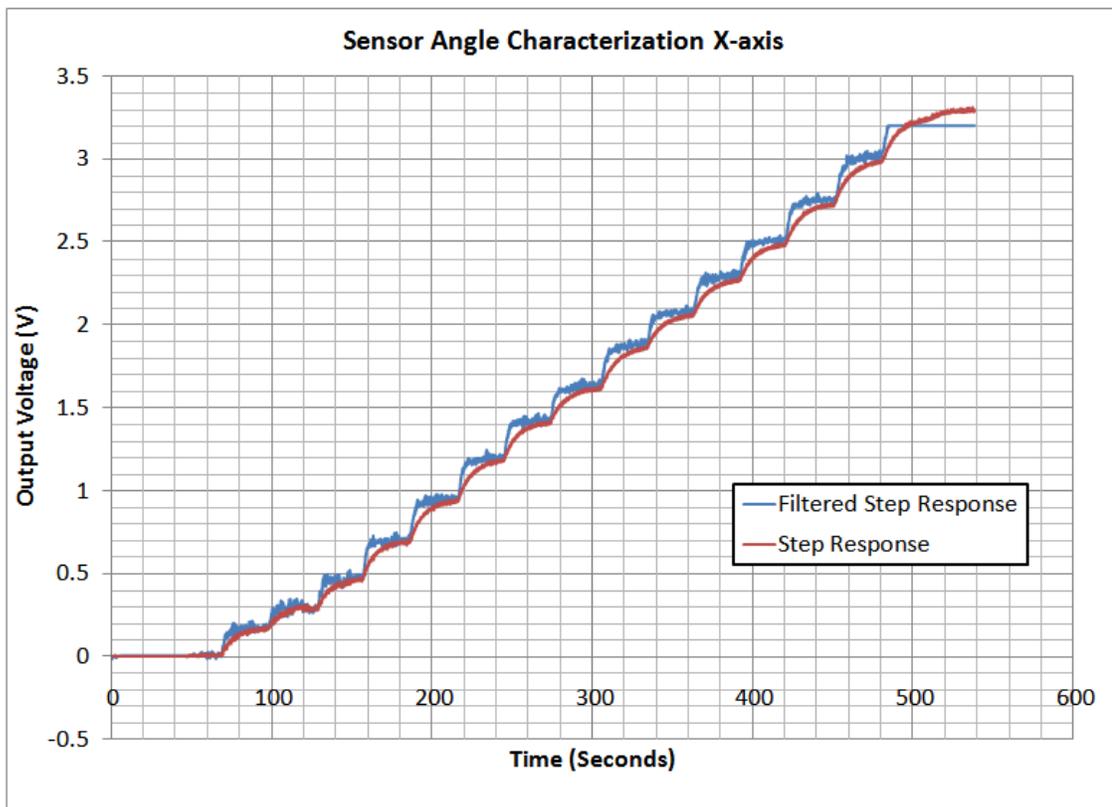


Figure 98: X-axis Angular Response

## 4.2 Sampling Scheme Evaluation

Since we are using one instrumentation amplifier and multiplexing between the two channels we came up with the following sampling scheme. On the figure bellow we can see the following signals:

- Channel 1** (yellow) – The control signal for the analog multiplexer
- Channel 2** (light blue) – General I/O pin on the  $\mu$ Controller. Sampling occurs on its rising edge
- Channel 3** (purple) – Output of the instrumentation amplifier

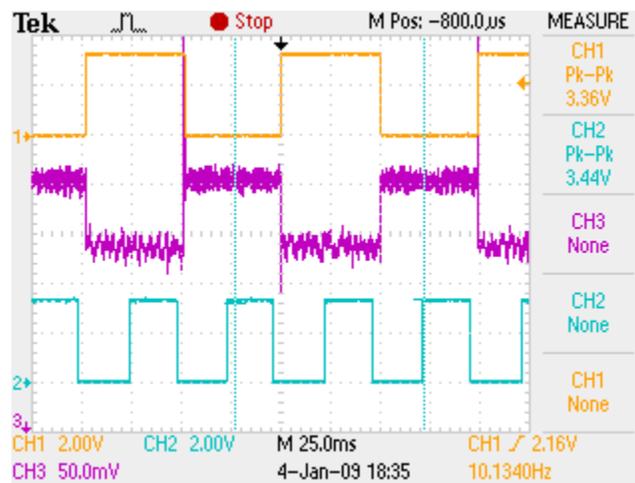


Figure 99: Sampling the multiplexed signal

Sampling frequency that we decided to use is 10 Hz. So each second we get 10 samples for both channels. From the figure above we can see that the sampling occurs after the switching has occurred and the output of the amplifier has settled down. When we look at a smaller time division on the figure bellow we can see the ringing that occurs at the switching of the inputs to the in-amp and this is the reason why we wait for about 25ms before we sample. This time is much greater than the settling time (less than 1ms) which guarantees that the sample is what we want.

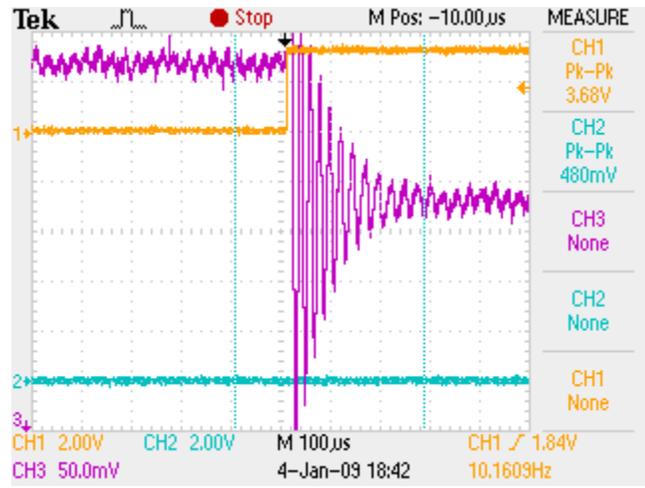


Figure 100: Zoomed sampling the multiplexed signal

Also when the channels are switched, the DAC output is also switched to provide the according voltage for the reference pin for the corresponding channel.

### 4.3 Calibration Scheme Evaluation

To take into account different values of the thermistors on the sensor side at equilibrium we perform a calibration procedure. It is very simple but still very effective. After the circuit is powered and left on an even surface that is not tilted in any direction to warm up and reach equilibrium, the calibrate button can be pressed. Twenty samples of each channel are taken and then the average of these samples is calculated for each channel correspondingly. The resulting value is used as value for the DAC that is driving the reference pin of the instrumentation amplifier. This calibration procedure is useful for several reasons, first the zero degree tilt in both directions is known and second by putting the output at the middle of the range of the ADC we get the maximum dynamic range. The result of the calibration can be seen on the figure bellow.

**Channel 1** (yellow) – The control signal for the analog multiplexer

**Channel 3** (purple) – Output of the instrumentation amplifier

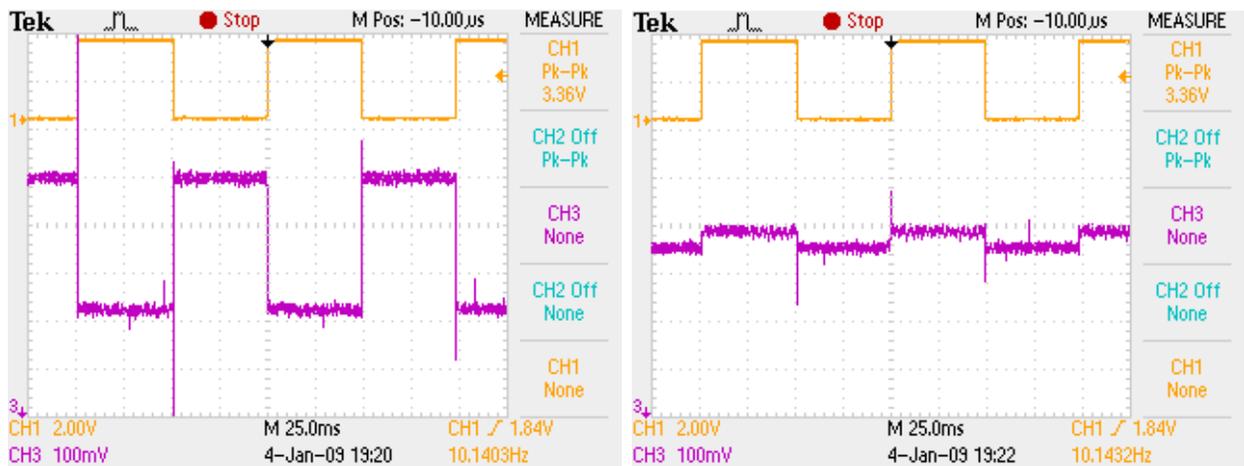


Figure 101: Output before (left) and after (right) Calibration

## 5. Conclusion

This project was open ended in the beginning which allowed us to touch different possible aspects of the design process. The starting point was choosing a specific IC and we chose one that was offered to us from one of the sponsors of the NECAMSID lab – Analog devices. The IC – AD8295, which was still not released to the market at the time, is a high precision instrumentation amplifier that also has two additional uncommitted operational amplifiers and a trimmed precision resistor divider network on the same silicon die which gives a lot of advantages. The stated final goal was to come up with an application emphasizing all its features and design an application board for it as well as appropriate software for a computer. This demo board would be used to present the part at different electronics trade shows that Analog Devices participates in.

The first step was doing initial research on instrumentation amplifiers and typical applications. This was assisted by several meetings between the team and the designer of the IC – Oljeta Bida and the product engineer – Paul Blanchard, and also the advisor of the project Prof. John McNeill. Several additional requirements were also derived – the application should be attractive to the users, it should be using USB connection for both data transfer and power, and interactive computer demo should be designed according to the application. After compiling list of possible applications they were weighted with appropriate criteria and the list was narrowed down to several applications that needed further research and tests to determine their feasibility.

In parallel to researching applications, different parts of the project were designed, that were required regardless the final application – this included data acquisition using a  $\mu$ Controller with integrated Analog to Digital Converter and its characterization, USB communication between the  $\mu$ Controller and the computer, interfacing the instrumentation amplifier with the  $\mu$ Controller, running dual supply for the AD8295 when we have only a single supply from the USB and properly interpreting the incoming data from a computer. Also several circuits were analyzed through analytical derivations and simulations to enhance the teams understanding of instrumentation amplifiers. A PCB board revision 1 was designed, fabricated and populated having the basic features available for testing different applications.

After further investigation we came to conclusions that some of the applications were not feasible in the time frame that we had, with the specific IC we chose or just not appropriate and interactive enough as an application demo. So the final application that was chosen was a temperature based tilt meter, using idea from a company called MEMSIC.

The principle of operation is the following – two thermistors which are a part of a Wheatstone bridge circuit are placed symmetrically around a resistor which acts like a heat source and all of them are sealed in a plastic box. When placed on a leveled surface both the thermistors are heated equally. When tilted one way or the other due to convection of the hot air from the heat source, one of the thermistors heats up more than the other. The resulting voltage difference is amplified by the instrumentation amplifier and then read through the ADC and transferred to a computer through the  $\mu$ Controller and USB chipset.

In the process of designing the sensor, we made four different casings from different materials and sizes. Also, different thermistors were used – surface mount, through-hole ones and also Resistive Temperature Detectors (RTDs). Comparing the different results, the final design was using a through-hole thermistors and a plexi-glass case. Since the response time of the system was slower than required for an interactive demo, we made a theoretical model of our system which was comparable to our measured results. Thereafter we concluded that to decrease the response time, we needed to increase the bandwidth of our system. To achieve this we designed a high pass filter at the output which was implemented as a discrete Infinite Impulse Response (IIR) filter on the computer. MATLAB scripts were written to assist us with the understanding of the theoretical model and also for its design.

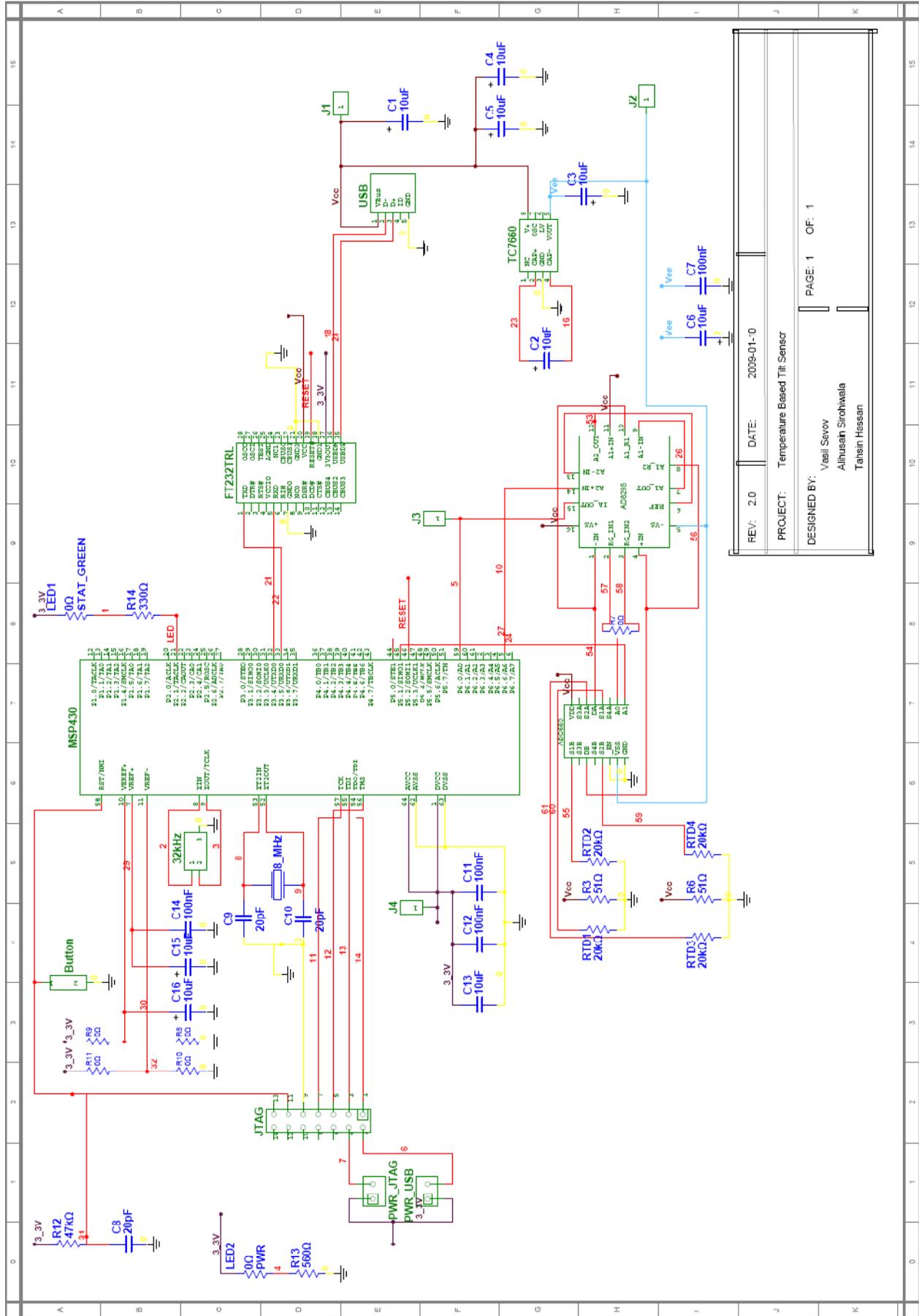
For the final design we decided to make the board for a 2-axis tilt sensor. Two pairs of this sensor were used orthogonal to each other. Instead of having two instrumentation amplifiers we used an analog multiplexer to switch between the two channels. Having the final design a PCB revision 2 was also designed, populated and detailed test were made. Also a calibration scheme was developed and implemented which allowed the maximum dynamic range for the ADC by calibrating the analog output at the middle of the range of the ADC after the sensor reaches equilibrium on a flat

surface. Both the non-filtered and the filtered results were observed and recorder and they were close to the theoretical expectations. Complete characterization from +90 to -90 degrees was made for both sensors.

On the computer side a LabView program was developed. It has two separate modes. One is having scope reading of both sensors before and after the filtering, whereas the other is the interactive one where the user moves a ball through a maze by tilting the surface of the maze so the ball moves with different velocity in different directions according to the tilt and the user has to move the ball to a specific place in maze.

Even though there is still some lagging in the final output response time, this project was successful. While the accomplishments of this project are significant, there are several areas of future research that were beyond the scope of the project. Things that can be done are making a better more detailed thermal model of the sensor and from that optimizing its design by creating a better casing for example. Improvements can also be made on the computer side by optimizing the LabView code and in case the price is concerned a less powerful and cheaper  $\mu$ Controller can be used and the code for it appropriately re-written.

# APPENDIX A - Detailed final Schematic



## APPENDIX B – LabView Game

### Setup for Maze.VI

1. On the front panel, there is a box named VISA resource Name. Once user clicks on the radio box, user has the ability to choose the connection port from the drop down menu. The user chooses the com-port the tilt sensing device is connected to.

2. The user clicks on the RUN button in top of the screen

3. Once the program is running the user can cause angle variation to the board. The user should see the ball moving because of tilting the board. The user should try to move the ball towards the rightmost corner of the board. And the game will display the message "YOU WON" at this point.

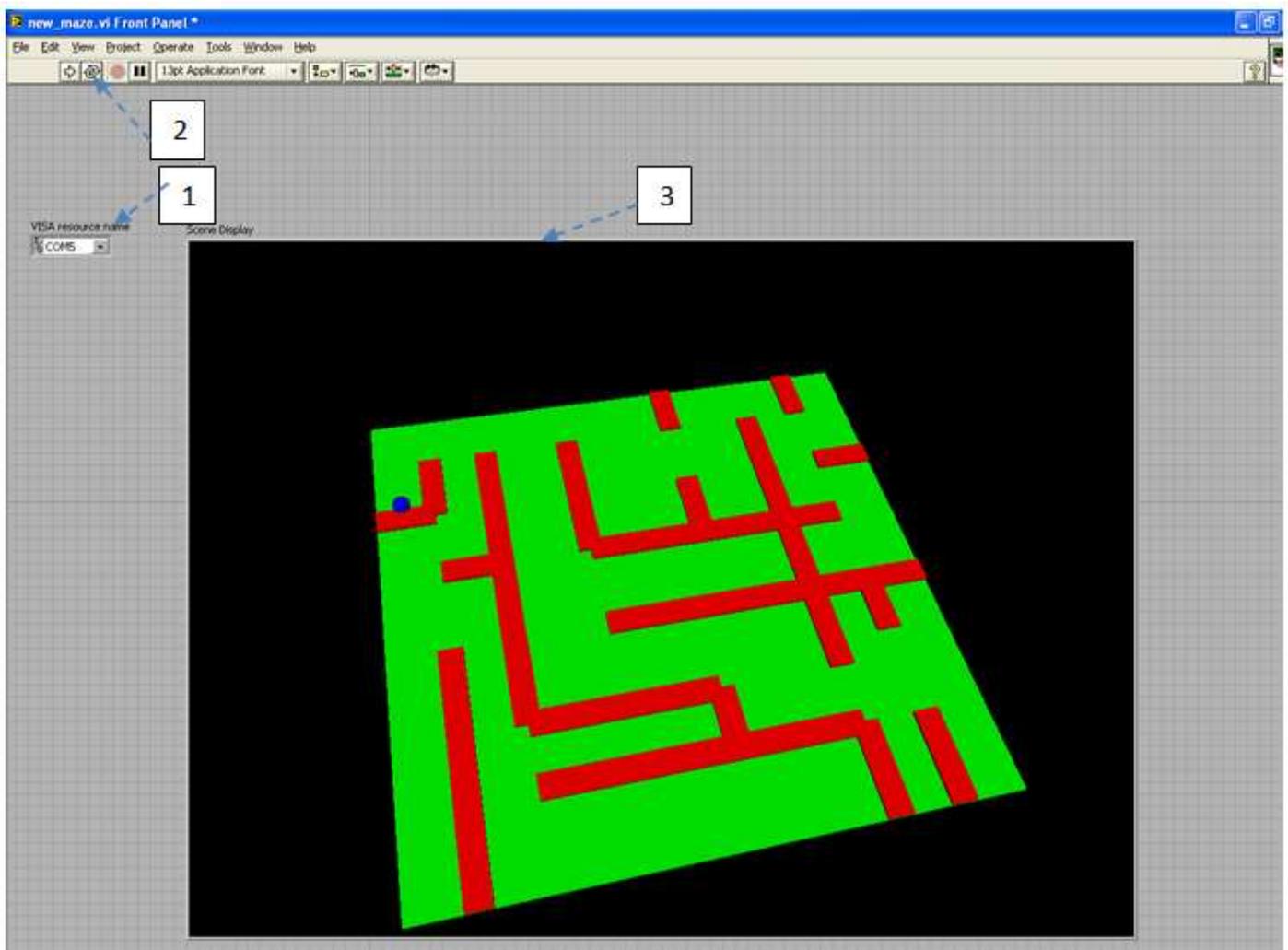


Figure 102: Front panel of maze.VI

## APPENDIX C – Scope Reading

### Setup for Scope Reading

1. On the front panel, there is a box named VISA resource Name. Once user clicks on the radio box, user has the ability to choose the connection port from the drop down menu. The user chooses the com-port the tilt sensing device is connected to.
2. The user clicks on the RUN button in top of the screen
3. Once the program is running the user can cause angle variation to the board. On the front panel, the user should see the changing pattern of the signal due to angle variation, also on the gauge indicator, the user will be able to see the current angle of the board.
4. The user can also see the current angle and voltage value as a numerical representation

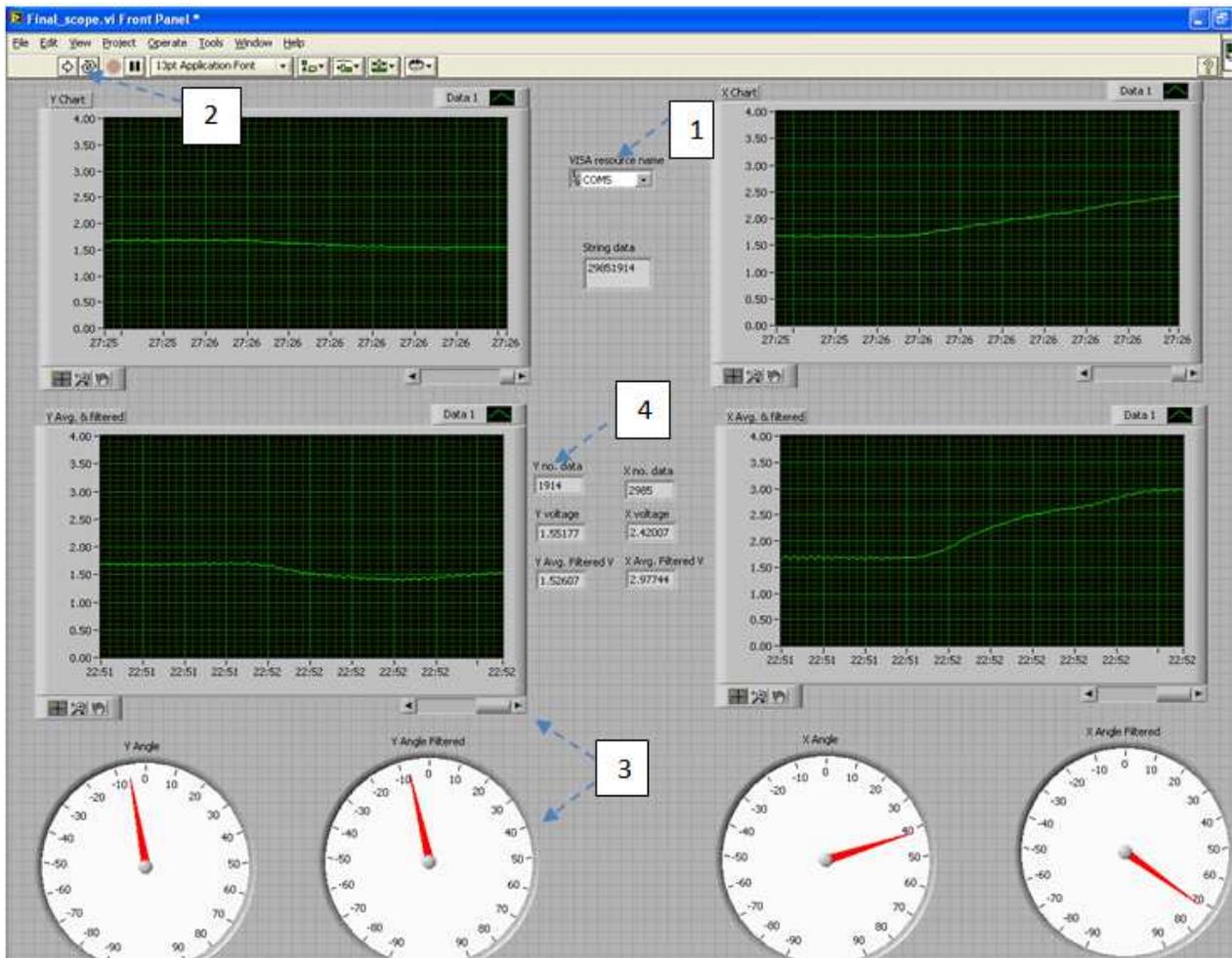


Figure 103: Front panel of Scope\_reading.VI

## APPENDIX D – Tutorial for scope Reading

### Parts of LabView

Every LabView program is comprised of two main parts a front panel- user interface, and a block diagram-the programming background. Once a program is created the user only needs to know how to setup the testing equipment and how to use the front panel. One can think of the front panel as a set of controls for the user to manipulate while the system is running and the block diagram as a collection of actions bound together like a flow-chart. When a LabView system is running, data is moved back and forth between the block diagram and the front panel.

The front panel of a LabView VI contains controls and indicators. As their name suggests, controls are the inputs into the system, and the indicators are the outputs. An example of a VI front panel can be seen in the following figure. Here one can see that the X graph, Y graphs etc. are the indicators/output and the VISA resource name (discussed later) is the input to the System. A **User Manual** for the **Scope\_reading.VI** is provided in the Appendix.

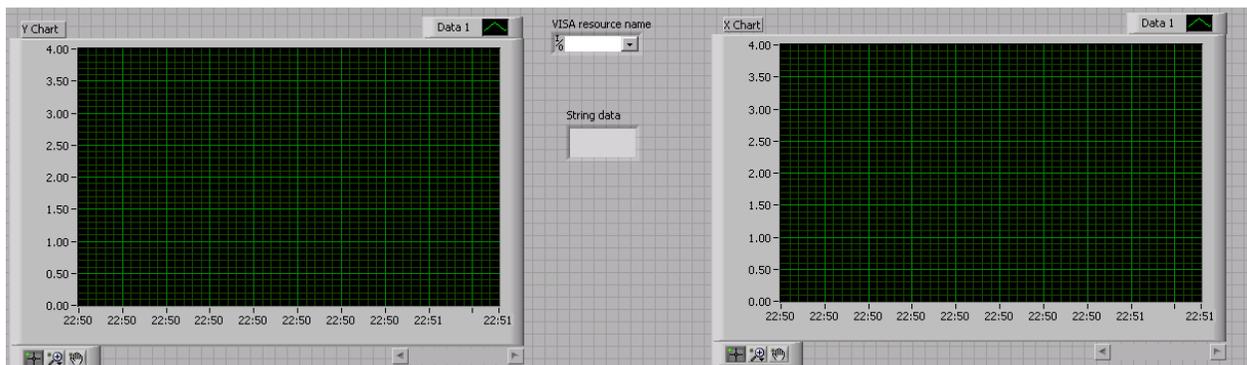


Figure 104: Small portion of Scope\_reading.VI Front panel

The Block diagram is where the user can link all of the parts of a LabView system together. Initially the user places the function icons that correspond to the actions they wish the system to take on the block diagram. They are then “wired” together in a set order to produce a coherent path through the system. An example of the Block Diagram can be taken from the Scope\_reading.VI. A description of the Block diagram is given in the following section.

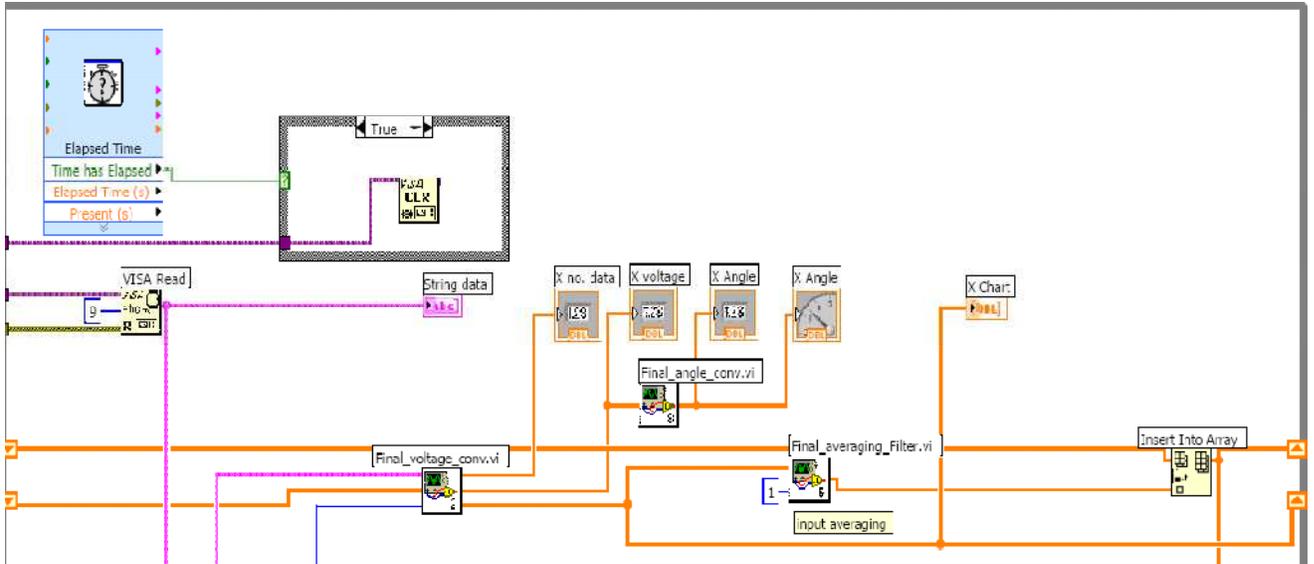


Figure 105: Small portion of Scope\_reading.VI block diagram

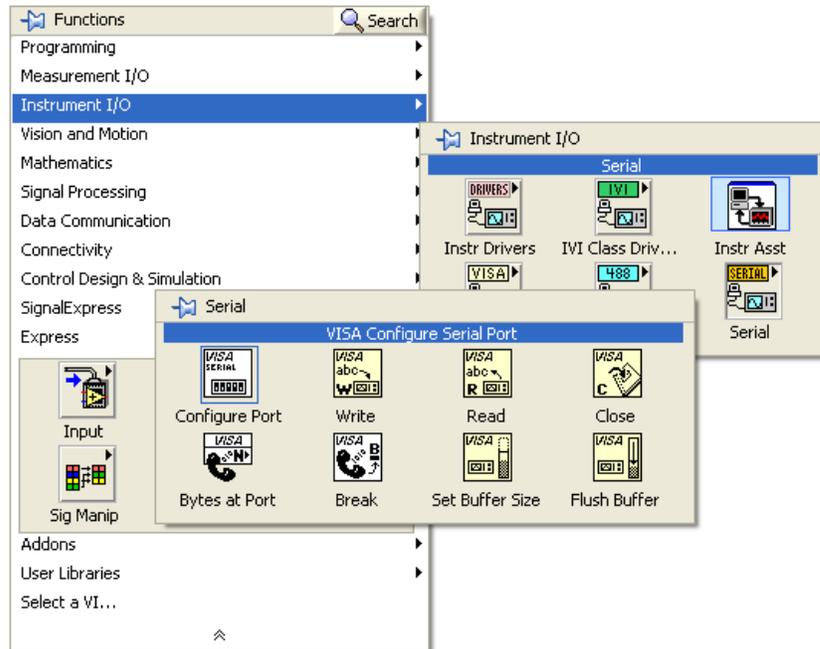
- a. We launch LabView by going to **Start >> Programs >> NI LabView 8.5**
- b. When the launcher loads we click on **Blank VI**.
- c. If the tool palette is not currently visible, go to **View** and click on **Tools Palette**.
- d. From the **File** menu, select **Save As** and save the file to your Hard disk (or USB drive) under a suitable name.

The file extension must be \*.vi.

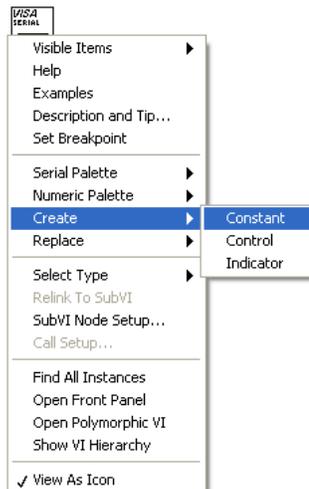
- e. With either the block diagram or front panel window selected, press <Ctrl+T> to tile the windows left and right. This way you can see both the block diagram and front panel at the same time.

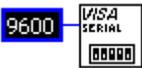
### VISA Configure Serial Port

1. Right click on the block diagram. The Function Palette shows up. From there choose **Instrument I/O > Serial > Configure Port**. You can also go to **Menu>View>Functions Palette> Instrument I/O > Serial > Configure Port**

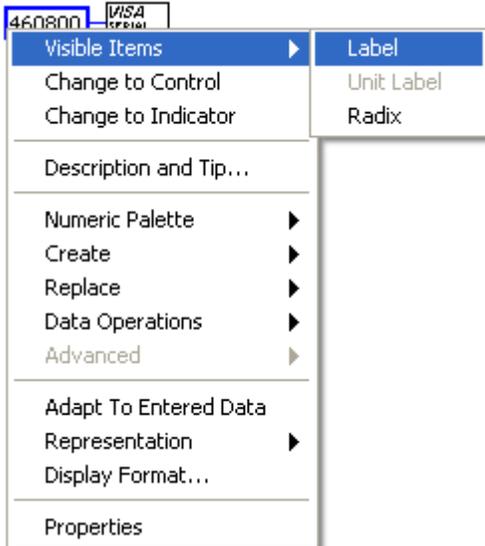


2.  The VISA Configure Port function shows up, Put Cursor on the function box  , the terminals show up.
3. Right click on one of the terminals and **create constant**.

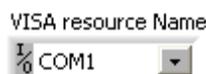


4. The default value shows up. 

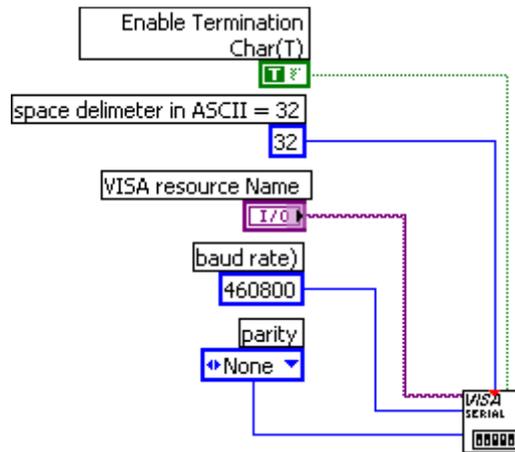
- Now you can click on the box, and change the value to 460800 (this is the speed we used in our system if needed other speeds can be used).
- Now you can right click to show the label of the Box you just created.



- baud rate
- In our application, the MSP430 provides two data points in a single set; one for the X axis, another for the Y axis, followed by the space delimiter. A typical data set would be `_23451234_45891237_`. The first four numbers after the space delimiter is the X axis data (2345) and the last four are Y axis data (1234).The next space tells LabView that a new set of data is coming. So we wired the VISA Configure, to **Enable Termination Char space (ASCII value= 32), so that LabView knows to differentiate between different sets of Data.**
- In the same way as before, we create constant for different Terminals
  - Enable Termination Char= T (signifies true)**
  - Termination char= 32,**
  - Parity= None.**
- Once we create the Visa Resource Constant, Note that a control box for this terminal shows up in the Front Panel.



11. All the other terminals are left to Default. The final figure for the VISA Configure Port should look like the following.



### VISA Read

12. Now once again by right clicking on the function Palette, we choose the VISA Read Function.

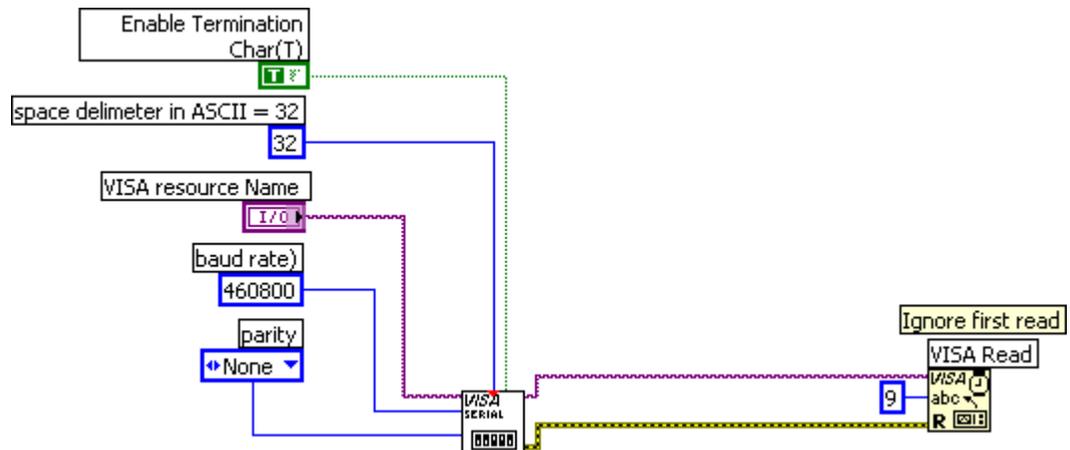
**Functions Palette > Instrument I/O > Serial > VISA Read.**

13. If we place the cursor on one of the terminals, the cursor changes to wiring tool . Wire the VISA Read Input **VISA resource Name** to the Visa Configure output terminal that says **VISA resource name out**.

14. In our application, each data set has nine string characters, eight of which are data points and one as a space delimiter. Thus, we set up the Read function to acquire only 9 bytes at a time. Right click on the terminal which reads **Byte count**, and create a constant. The default is 0, click on the box to change the **Byte count** to 9.

15. Also we connected the **error in** terminal of the VISA Read to error out VISA Configure.

16. The Final figure for this part should look like the following.



## While Loop

17. In our application the data received needs to be processed continuously, so we will need a While loop. To place the While loop on the block diagram, we go to functions palette and choose **Programming > Structures> While Loop**.

18. The figure looks like the following.



## Search/Split string

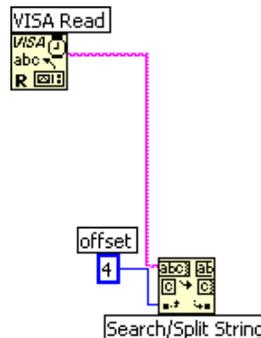
19. To process the acquired data any further we need to split this string into two parts, one containing the X axis data and the other containing the Y axis data. To place this block go to

**Functions> Programming >Strings> Additional Strings> Search/Split string**.

20. Once the Search/split String function is in the block Diagram, we wire the **String input** to the **read buffer output** of the 2<sup>nd</sup> VISA Read function.

21. In our application a typical buffer output will be 23451234; where first four digits are X axis data. So we create a constant for the **offset input** terminal and make the value equal to **4**.

22. The Final Block looks like the following

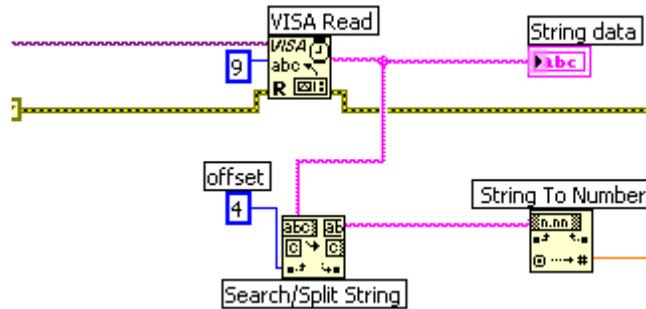


## Voltage SubVI

Once we have the data string for the X and Y axis; we need to convert the data string to pertinent voltage measurement. Since, we would use the same functionality for both x & y axis voltage measurement, it is possible for us to use a subVI block.

23. We needed to convert the data string to numbers, since we needed to do calculations on the data. So we used the **Fract/Exp String to Number**.

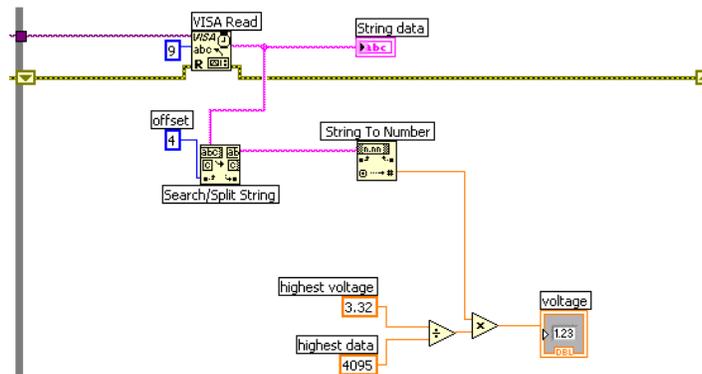
24. The block can be found under **Programming>String> String Number Conversion> Fract/Exp String To Number**.
25. The **String input** terminal is connected to **substring before match Output terminal of Search/Split string** block. At the end of this step the figure will look like the following.



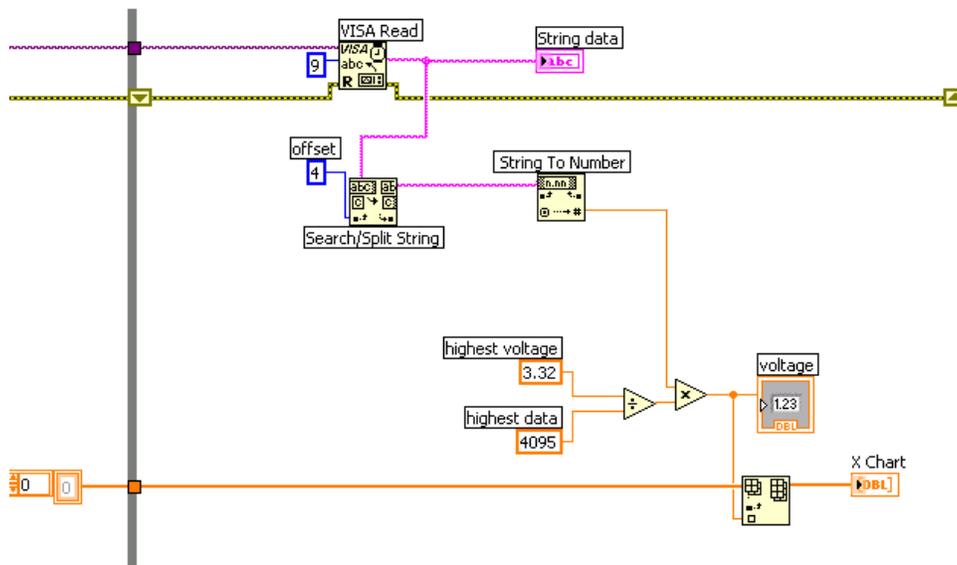
26. Once we have the numerical data for x/y axis; we need to convert the data to voltage.
27. The highest numerical data point that we get from the MSP430 is 4095, which also corresponds to 3.32 volts. So each data point corresponds to  $3.32/4095 = 0.00081074481$  Volts. If we multiply this resolution with the acquired numerical data, we can convert the acquired data to voltage. We can represent the equation as the following:

$$\frac{3.32}{4095} * \text{acquired data} = \text{voltage}$$

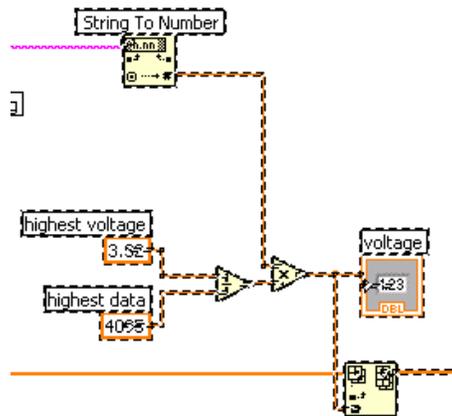
28. Under **Programming>Numeric** we will find the **Division** and **Multiply** block.
29. For the Division block create constants for both inputs. The division block has two constant x & y. and the resultant is x/y. For the x constant, we change it to 3.32 and for the y constant we change it to 4095. The resultant/ output of the division block is wired as an input to the multiply block .The multiply is connected to an indicator, which shows the voltage conversion.
30. At the end of this step, the block diagram should look like the following.



31. However, the Voltage indicator shows only one voltage conversion at a time. We need to show the voltage for a certain amount of time. So, we need to store the voltage measurements in an array. We used the **Insert into array** function.
32. The function can be found under **Programming> array**. We wire the **new element input** to the output of the Multiply block.
33. However, the array needs to be initialized to an empty array. So we create a constant by right clicking on the array input. The initialization should be done outside of the loop. The output of the array is connected to a chart. Right click on the front panel under **control> graph indicator>chart**.
34. The block diagram looks like the following after the last step.

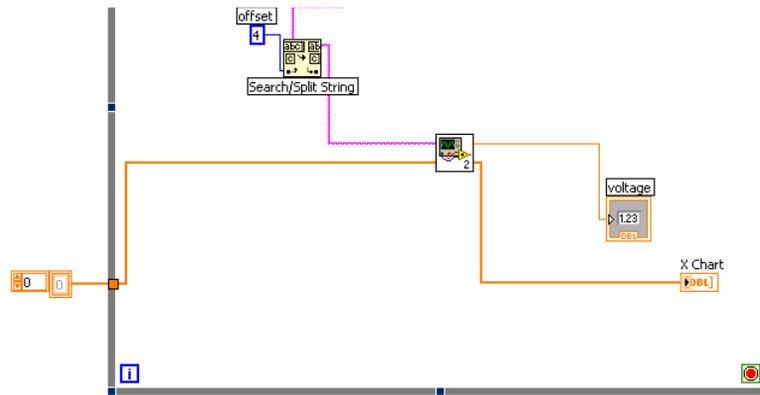


35. We can reuse the blocks after Search/split string block. So we created a subVI.

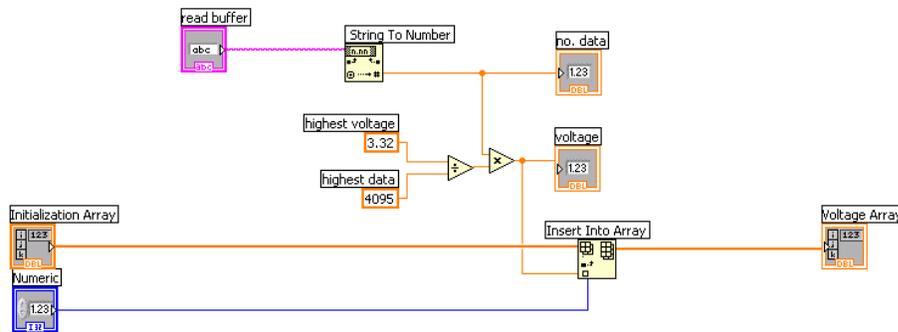


36. We choose the VI blocks we want to reuse, and then in the file menu choose **EDIT>Create subVI**, and name it as voltage subVI.

37. The main block diagram looks like following.



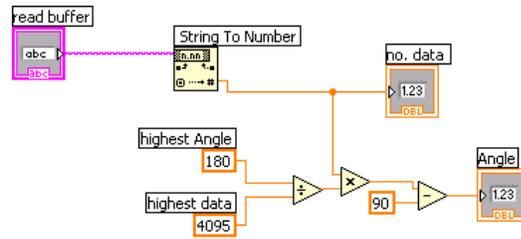
The voltage subvi automatically creates the terminals for it to connect to the main VI block.



### Angle subVI

38. We build the angle subVI following the same procedures of Voltage subVI. Only the multiply block inputs are changed.

39. Our board can measure a tilt of +90 to -90 a total of 180 degrees of tilt. The highest numerical data point that we get from the MSP430 is 4095. So each data point corresponds to  $180/4095 = 0.00081074481$  Volts. However, a data point of 4095 signifies an angle tilt of +90 degrees. If we multiply this resolution with the acquired numerical data and subtract -90 degrees, we can convert the acquired data to tilt angle. At the end of this step, the block diagram for the Angle subVI looks like following.



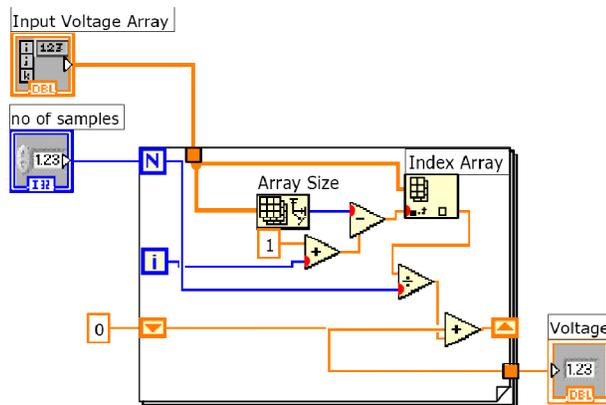
To show the angle measurements it is easier to use a gauge indicator, found under, **Programming> Numeric indicators> gauge**

### *Averaging filter subVI*

40. We use “for” loop to create an averaging filter. The “for” loop can be found under **Programming > Structures > for Loop**.

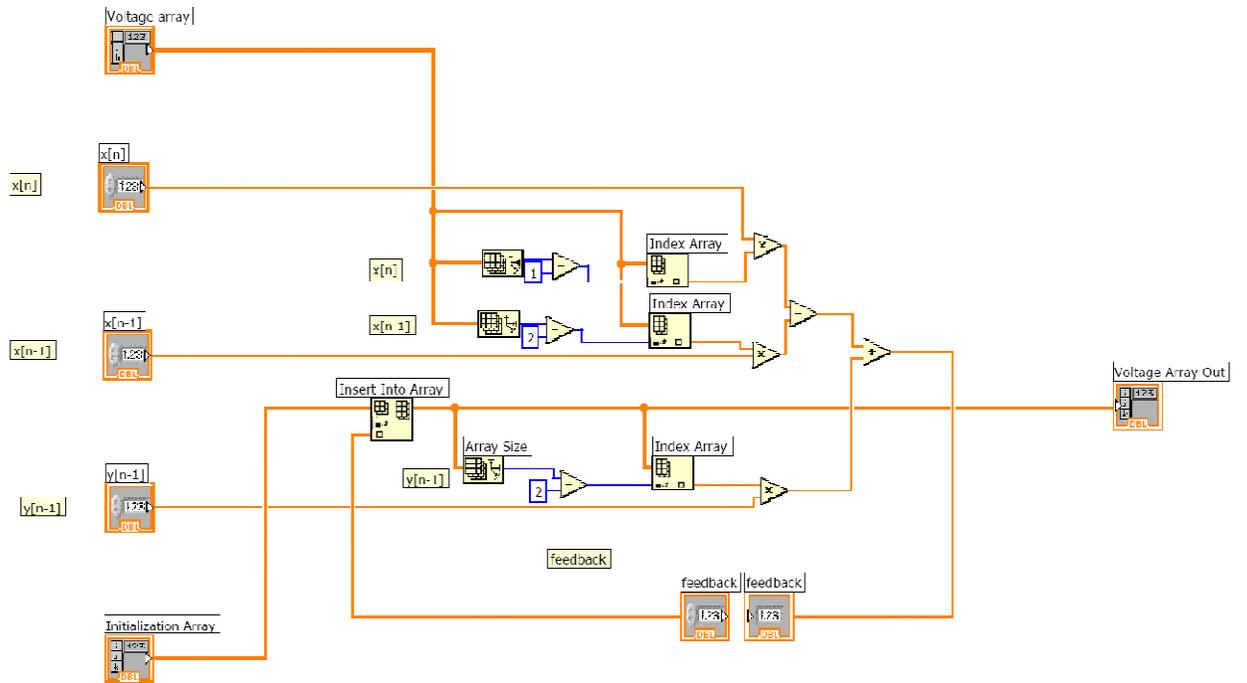
41. To create an averaging filter for 20 samples, we need to find previous 20 samples in an array and divide it by 20. Thus, we can use an **index array** to find each element with a certain index. This block can be found under **programming> array>index array**

42. We use the **Array Size** block to determine the size of the incoming array, and thereafter, if we subtract the number from 0 to 20, we will get the last index value in the incoming array for the past twenty samples



### *IIR highpass filter subVI*

43. The IIR highpass filter was created with the same elements in the Averaging filter subVI. Please refer to the IIR highpass filter block diagram and the IIR filter equation.



# APPENDIX E – PCB Layout

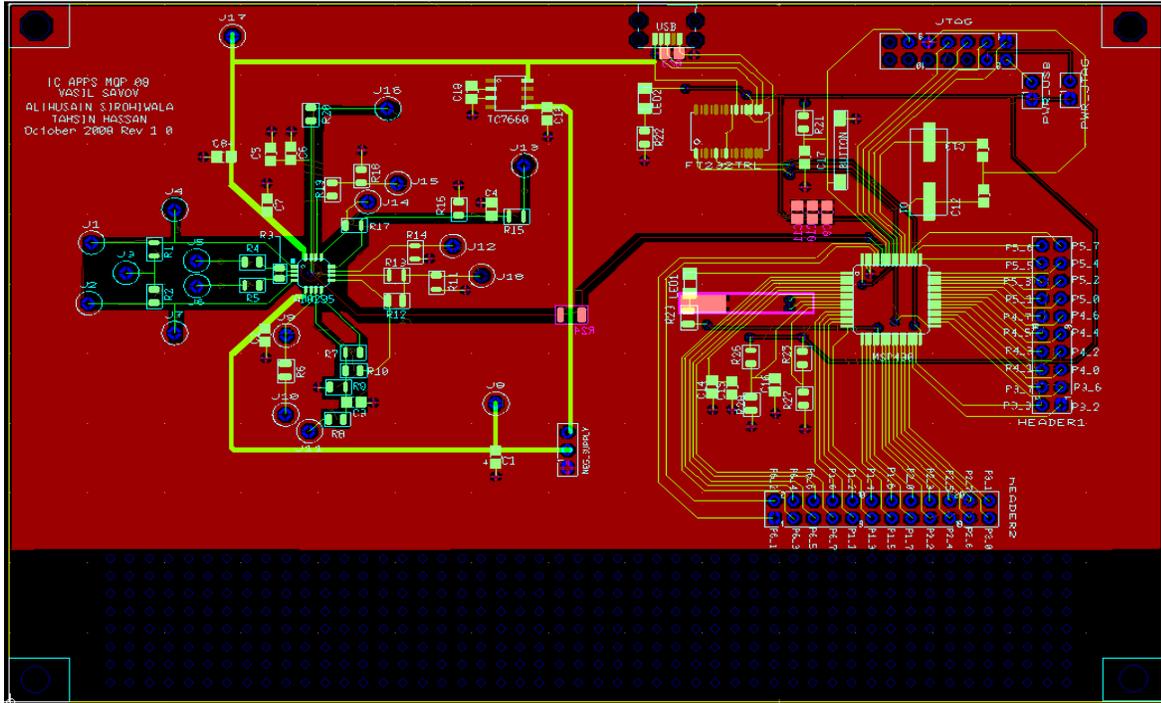


Figure 106: Layout revision 1.0

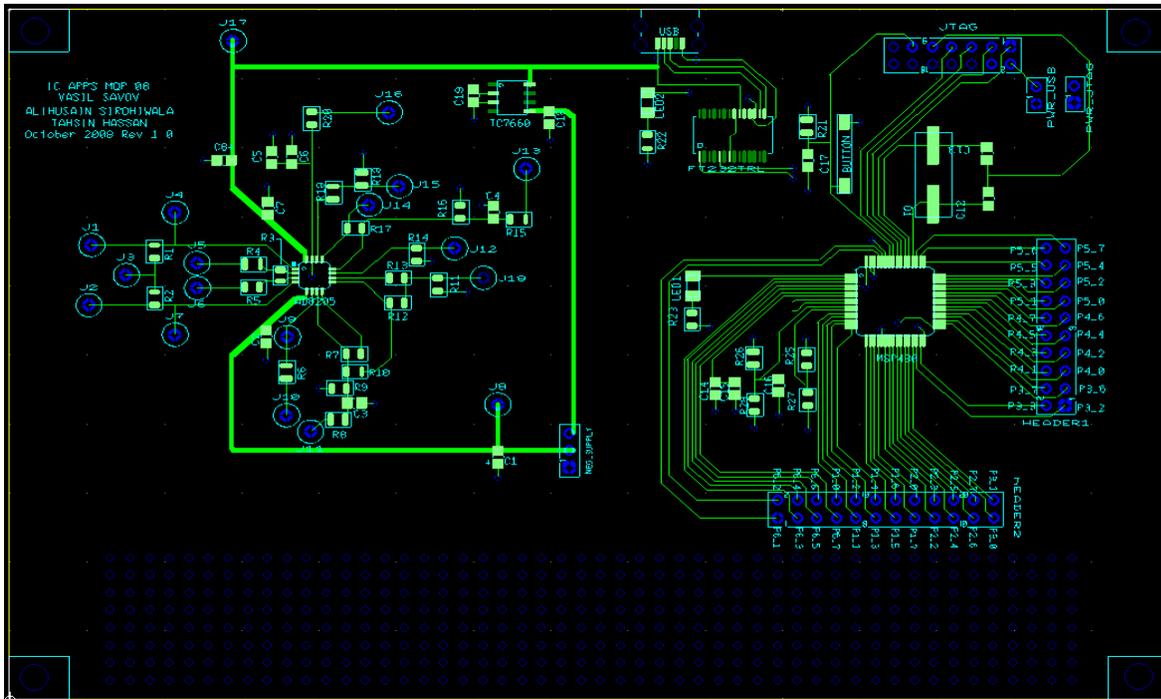


Figure 107: Top layer of PCB Rev 2.0

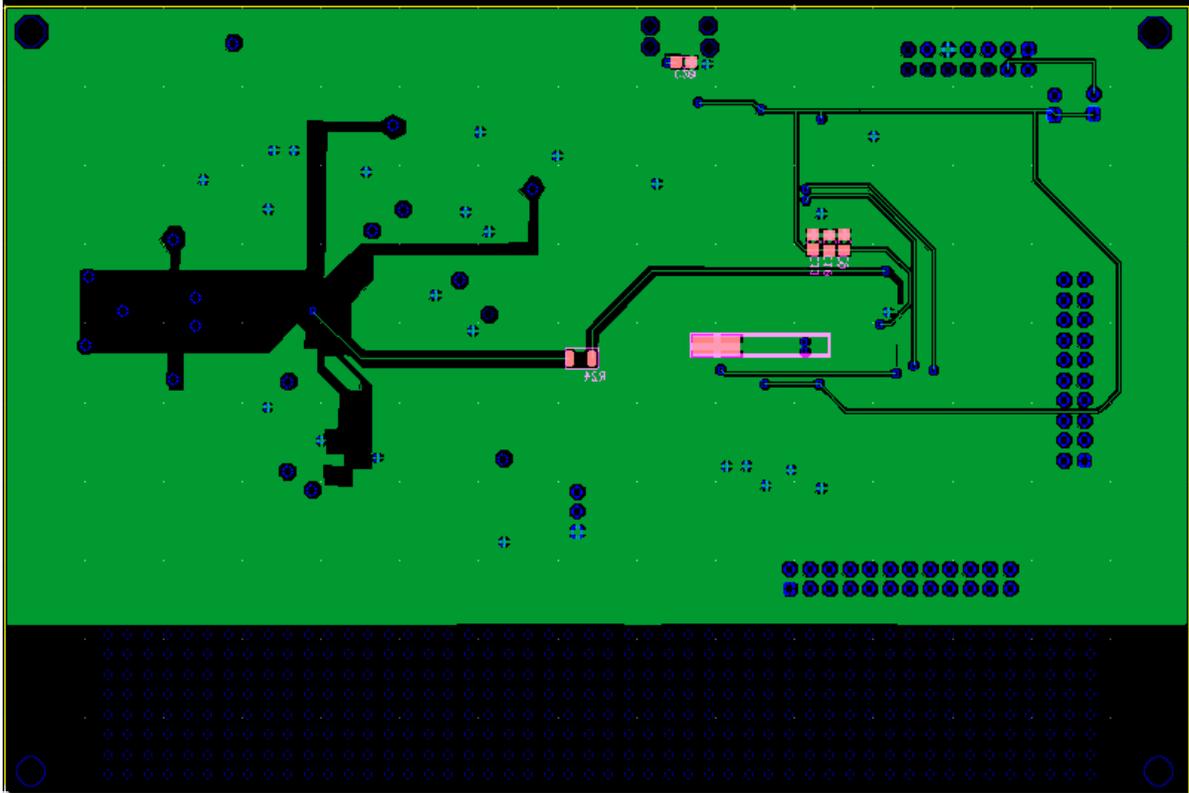


Figure 108: Bottom layer of PCB Rev 2.0

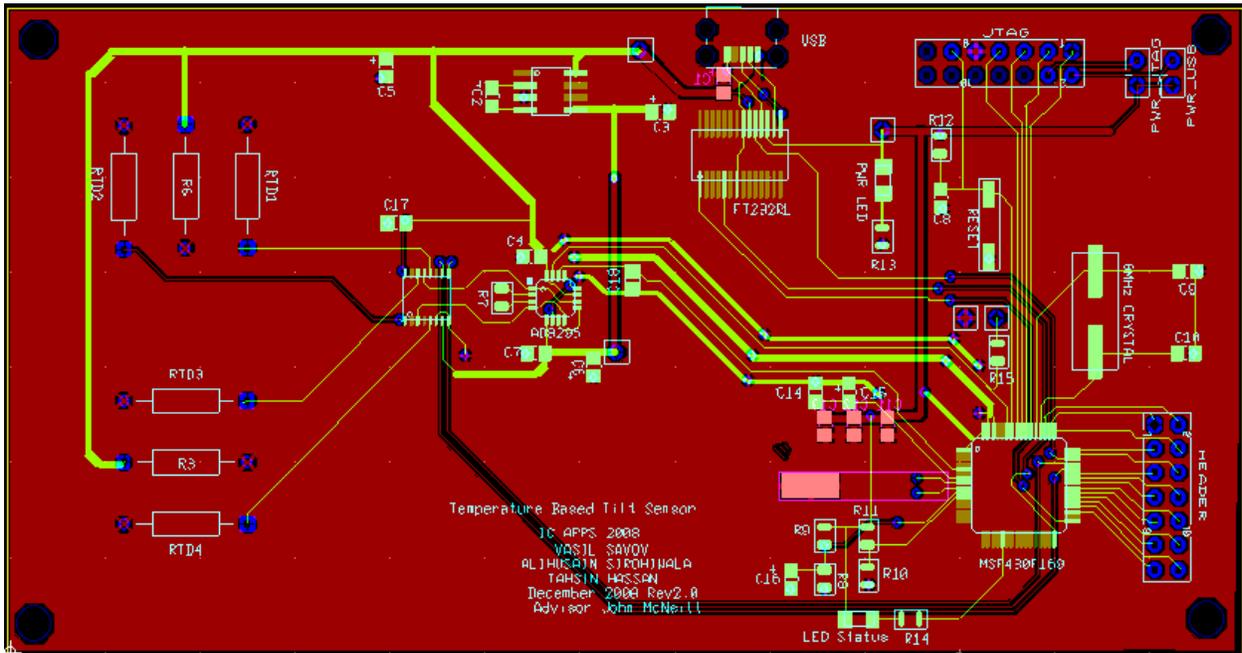


Figure 109: Layout Revision 2.0

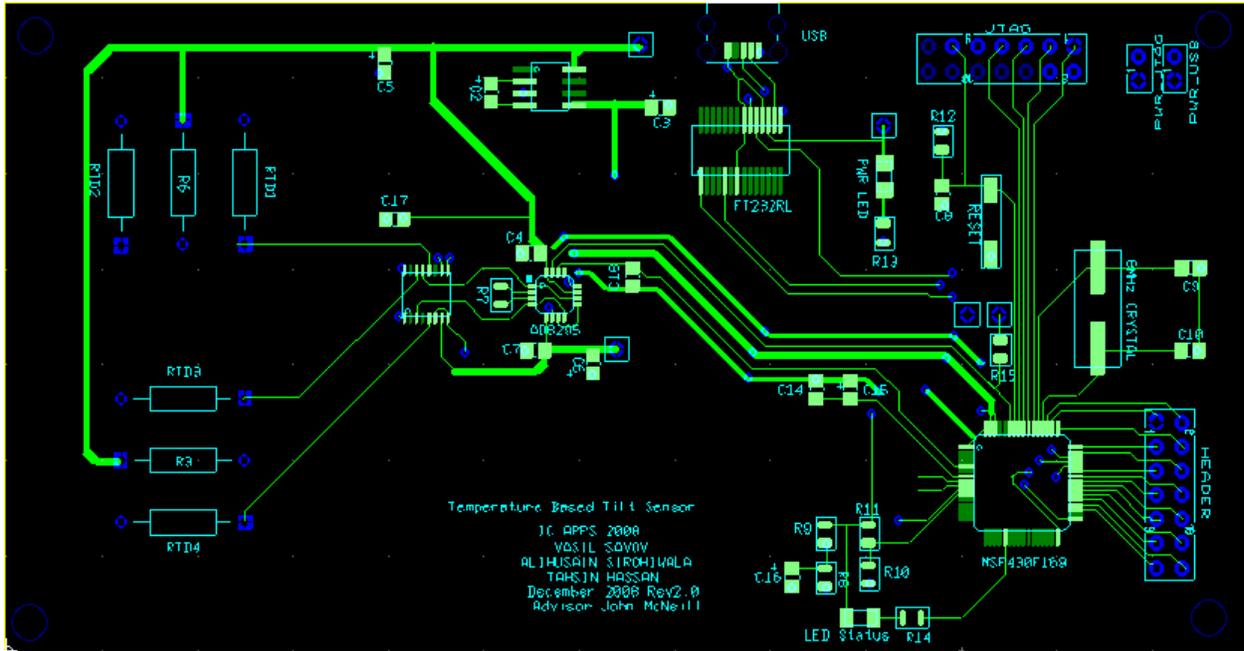


Figure 110: Top layer of PCB Rev 2.0

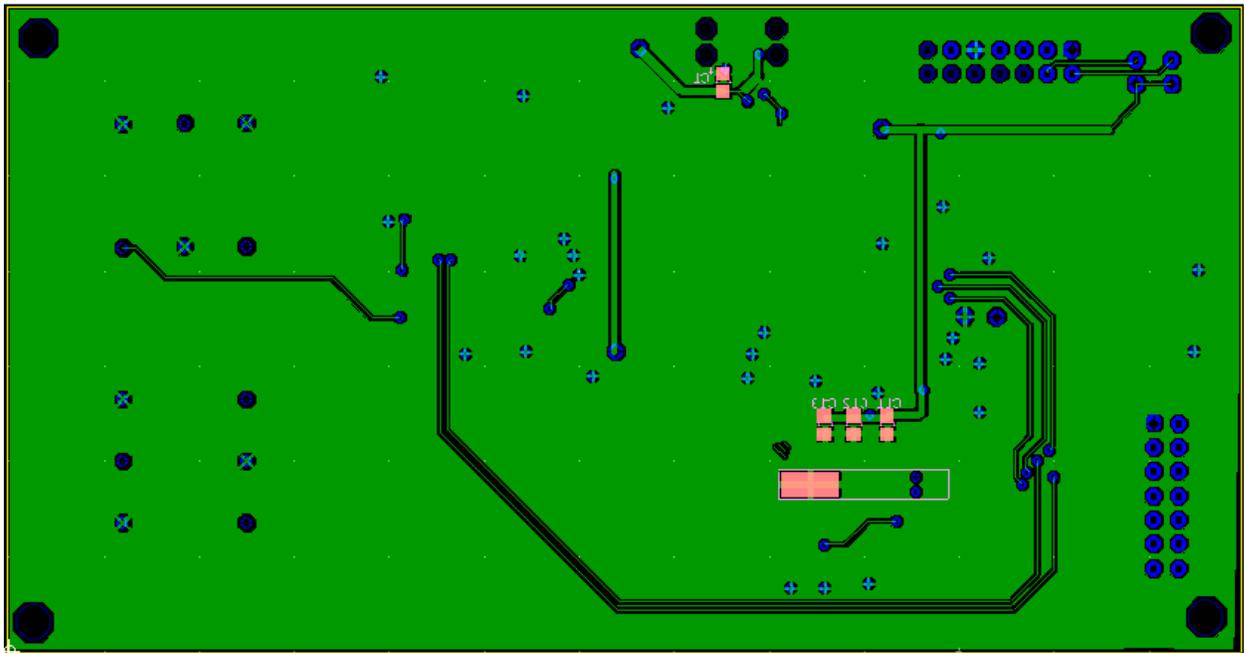


Figure 111: Bottom layer of PCB Rev 2.0

## APPENDIX F – MATLAB Script code for the filter design

```
%IC Application Team WPI 2008
%Dec 02,2008
clc
clear all
%read the Excel file
[time] = xlsread('Step_response_y.csv','B2:B1080'); %we are taking data
                                                    % for 100 seconds
[time] = time-12; % we subtract 12
                                                    % seconds to make
                                                    % the start time
                                                    % at zero seconds

[amplitude]= xlsread('Step_response_y.csv','D2:D1080')-1.6; % DC offset of 0
% We use an averaging filter to smooth out the acquired data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      1      1      1      1      1
% y(n)=-x(n) + -x(n-1)+ -x(n-2) + -x(n-3)+.....-x(n-19)
%      20      20      20      20      20
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
no_samples= 20;
a_average = 1;
b_average = 1/no_samples*ones(1 ,no_samples);
y_average = filter(b_average,a_average,amplitude);
% acquired data in scope
figure(1);
plot(time,amplitude);
hold on
plot(time,y_average,'r');
hold off
xlabel('Time');
ylabel('Amplitude');
title('Acquired Data');
grid on
legend('Acquired data', 'Averaged data')
% Note: As a first approximation we assumed the initial system to be
% first order;
risetime=20 % the 10-90% risetime is 20 seconds
tao=risetime/2.2; % time constant
%the transfer function of our system is given by the following
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Response of unfiltered initial system
%      Transfer function:
%      gain* 1
%      -----
%      tao* s + 1
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
gain =1.55; % by inspection we found that the gain of
           % 1.55 provides a close match between our
           % theoretical model and acquired data

num_initial = gain*[1]; % numerator
den_initial = [tao,1]; % denominator
Sys_initial=tf(num_initial,den_initial)% transfer function
```

```

Sys_initial_trial=tf(1,den_initial)

figure(2)
[sys_amp,sys_time] = step(Sys_initial);
[sys_amp_trial,sys_time_trial] = step(Sys_initial_trial);
plot(sys_time,sys_amp,'r')
hold on
plot(time,y_average,'b')
plot(sys_time,sys_amp_trial,'k')
hold off
grid on
title('step Response[acquired +averaged data & theoretical transfer function] ')
legend('final theoretical model', 'acquired step data', 'first approximation
model')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% highpass filter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Response of highpass filter[continuous time]
% Transfer function:
%   tao* s + 1
%   -----
%       s + 1
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
num_filter= [ tao 1];
den_filter= [1 1];
Sys_filter=tf(num_filter,den_filter)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% total filtered system
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Response of total system with the added filter[continuous time]
%Sys_total=Sys_initial*Sys_filter
% Transfer function:
%   gain
%   -----
%   s +1
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

num_total= [ gain];
den_total= [1 1];
Sys_total=tf(num_total,den_total)
figure(3)
bode(Sys_initial)
hold all
bode(Sys_filter)
bode(Sys_total)
hold off
grid on
title('response of systems[continuous time]')
legend('initial system ','filter','total system')
figure(4)
step (Sys_initial)
hold all

```

```

step(Sys_total)
grid on
hold off
title('step response of systems[continuous time]')
legend('step of initial ', 'step of filtered system')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Analog to Digital filter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Z transform
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fs=10 ;                               %sampling frequency of MSP
Ts=1/fs;
Sysd_filter = c2d(Sys_filter,Ts,'tustin')

%The next command compares the continuous and discretized step responses
figure(5)
step(Sys_filter, '-', Sysd_filter, '--')
title('step response of high pass filter[continuous+ digital]')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Transfer function:
%      Z transfer
%      8.706 z - 8.61
%      -----
%      z - 0.9048
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

TotalSize = size(amplitude,1);
y=ones(TotalSize,1);
y_trial=ones(TotalSize,1);

for i = 2:TotalSize
    % discrete time domain [n]
    % using Bilinear theorem we can convert the continuous time domain
    % representation to discrete time domain representation.
    % y[n]= 8.706x[n] -8.61x[n-1] +0.9048y[n-1]

y(i) =      8.706*y_average(i)-8.61 *y_average(i-1)+0.9048 *y(i-1);

    %after looking at the step respons of the system , we see that there is
    %a overshoot before the signal reaches it's settling value
    %thus by trial and error , we found a case where there is no overshoot
    %before the signal reaches it's settling value.
    y_trial(i)= 8.76 *y_average(i)-8.635*y_average(i-1)+0.876 *y_trial(i-1);
end

figure(6);
plot(time,amplitude,'m');
hold all
plot(time,y,'r');

```

```

plot(time,y_trial,'k');
hold off
axis([0 100 0 1.8])
xlabel('Time');
ylabel('Amplitude');
grid on
title('step Response of filtered systems[digital]');
legend('acquired data', 'filtered-with overshoot','filtered-without overshoot');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%zero pole plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Sysd_initial = c2d(Sys_initial,Ts,'tustin')

Sysd_total = c2d(Sys_total,Ts,'tustin')

% Set up vector for zeros

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Transfer function:
%      8.706 z - 8.61
%      -----
%      z - 0.9048
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
numz_filter=[8.706 -8.61];
denz_filter=[1 -0.9048];
z_filter = [roots(numz_filter)];
p_filter = [roots(denz_filter)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(7)
pzmap(Sys_initial)
sgrid
title('Initial system Z plane')

figure(8)
pzmap(Sys_filter)
sgrid
title('Filter Z plane')

figure(9)
sgrid;
pzmap(Sys_initial)
hold on
sgrid;pzmap(Sys_filter)
title('total system Z plane')

```

## APPENDIX G – Matlab Scripts for Noise

```
/**
-----IC APPLICATION TEAM - NECAMSID LAB WPI -----
-----MQP 2008-----
// Matlab Script For Plotting Noise Data and Noise FFT.
//
// IC Applications - 2008
/**

Fs = 10;           % Sampling frequency
T = 1/Fs;         % Sample time
L = 2500;         % Length of signal - number of samples
t = (0:L-1)*T;    % Time vector

figure(2)

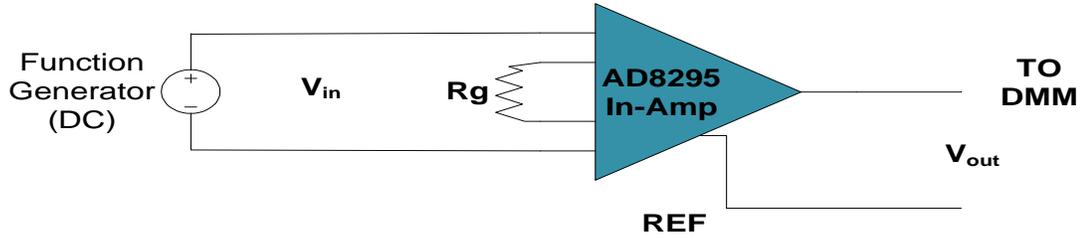
x1 = cast(Sheet2 , 'single');
subplot(2,2,1)
plot(t,x1)
title('Aquired Data - New Sensor')
xlabel('time (seconds)')

NFFT = 2^nextpow2(L); % Next power of 2 from length of y
X1 = fft(x1,NFFT)/L;
f1 = Fs/2*linspace(0,1,NFFT/2);
Y1 = 2*abs(X1(1:NFFT/2));

% Plot single-sided amplitude spectrum.
subplot(2,2,2)
plot(f1(1:length(f1)),20*log10(Y1(1:length(f1))))
title('Single-Sided Amplitude Spectrum of Sensor in Rest')
xlabel('LOG(F)')
ylabel('|Y(f)| (dB)')
```

# APPENDIX H – Measured Performance Characteristics of AD8295

## DC Gain of 1



Calculated Gain (Based on measured  $R_G=5.026\text{kohms}$ ) = **1 v/v**

Measured Gain (Based on best-fit slope) = **1 v/v**

Output Offset (Based on Intercept of best-fit line) = **-400 $\mu\text{V}$**

Table 17: Gain of 1 evaluation

G=1	
V <sub>in</sub>	V <sub>out</sub>
9.9900E-01	-1.0000E+00
1.9990E+00	-2.0000E+00
3.0000E+00	-3.0000E+00
4.0000E+00	-4.0000E+00
1.0001E+01	-1.0001E+01
1.0001E-01	-1.0007E-01
2.0000E-01	-2.0006E-01
3.0001E-01	-3.0006E-01

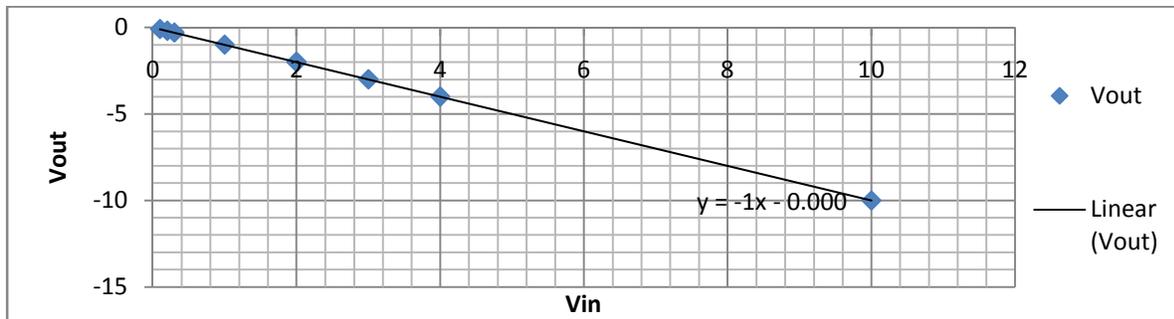


Figure 112: Vin vs Vout G=1

### DC Gain of 100

Calculated Gain (Based on measured  $R_G = 504.9\text{ohms}$ ) = **98.841 v/v**

Measured Gain (Based on slope) = **98.679 v/v**

Output Offset (Based on Intercept of best-fit line) = **-2.7mV**

Table 18: Gain of 100 evaluation

G=100	
Vin	Vout
0.0000E+00	-2.5000E-03
1.0003E-01	-9.8740E+00
5.0020E-02	-4.9380E+00
2.0010E-02	-1.9780E+00
3.0010E-02	-2.9640E+00
7.0020E-02	-6.9120E+00
1.2005E-01	-1.1849E+01

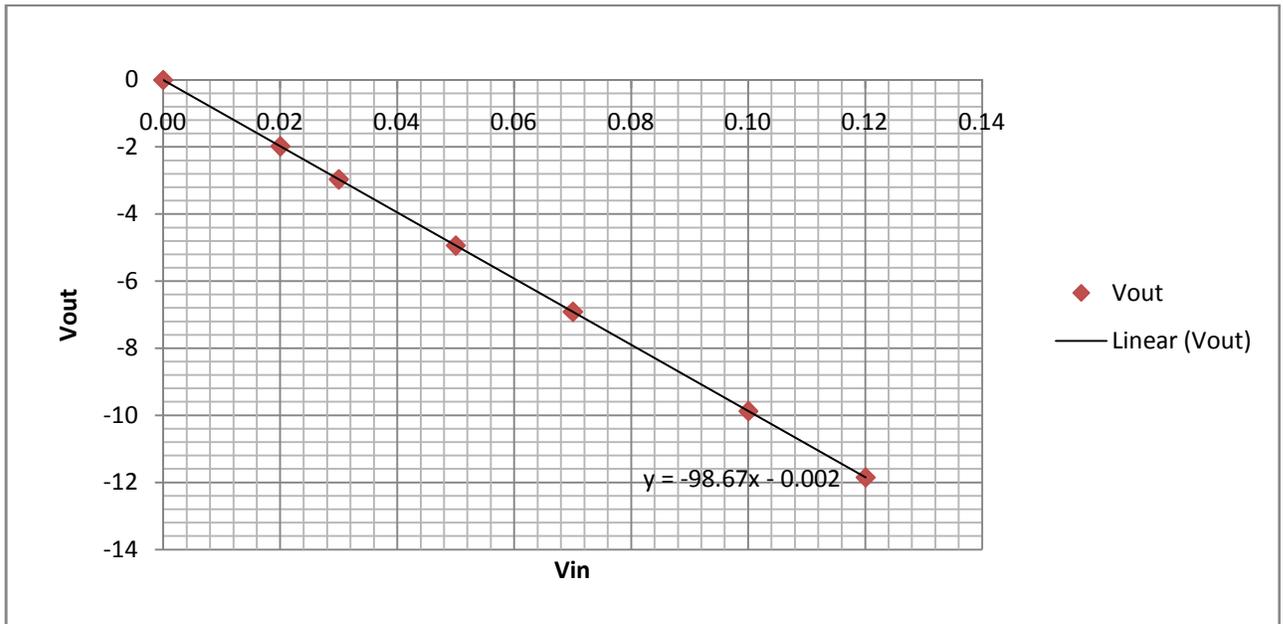


Figure 113: Vin vs Vout G=100

### DC Gain of 1000

Calculated Gain (Based on measured  $R_G=50.58\text{ohms}$ ) = **977.671 v/v**

Measured Gain (Based on slope) = **975.73 v/v**

Output Offset (Based on Intercept of best-fit line) = **-1.96mV**

Table 19: Gain of 1000 evaluation

G=1000	
Vin	Vout
0.0000E+00	-2.4000E-02
1.0100E-03	-1.0070E+00
2.0200E-03	-1.9890E+00
3.0200E-03	-2.9640E+00
4.0100E-03	-3.9280E+00
1.2010E-02	-1.1741E+01
9.0100E-03	-8.8100E+00

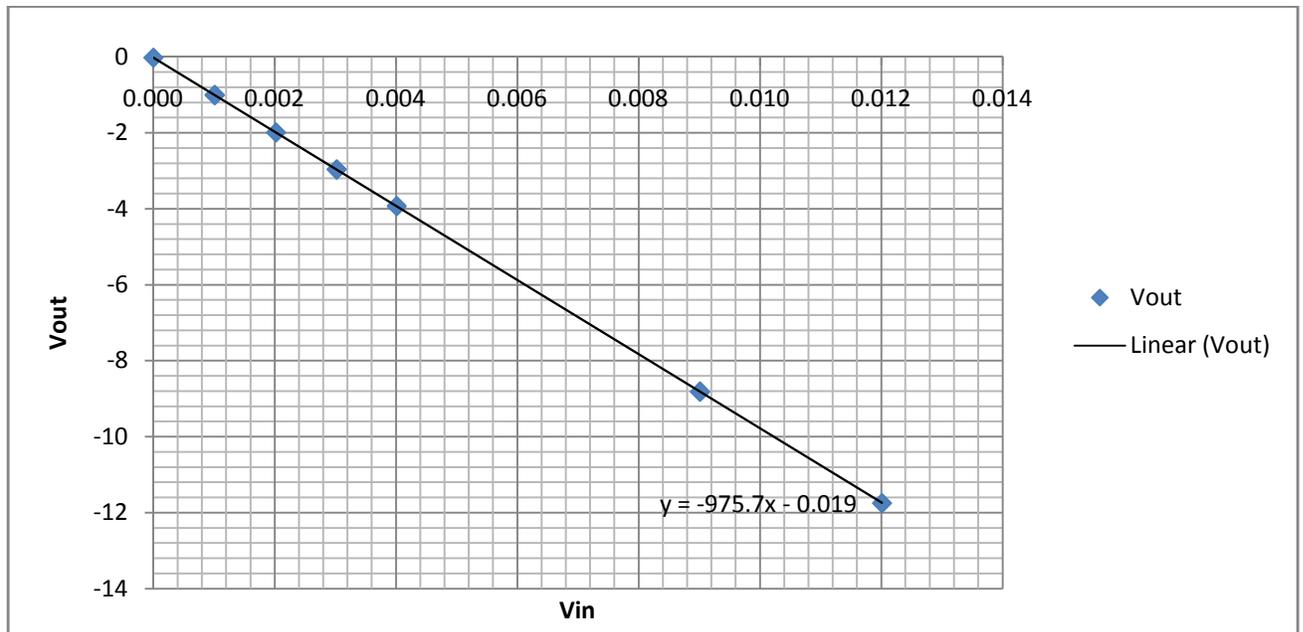


Figure 114: Vin vs Vout G=1000

## Bandwidth Measurement

BW measurement was done using the Tektronix TDS 2004B Oscilloscope.

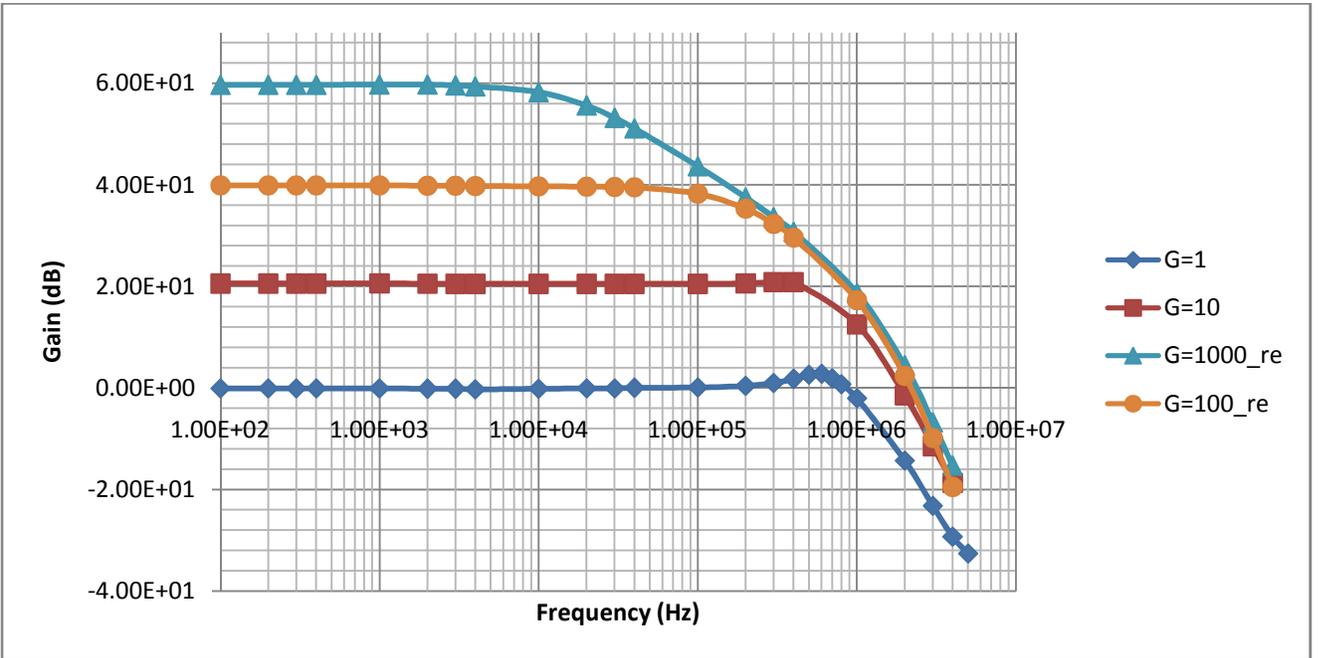


Figure 115: Bandwidth

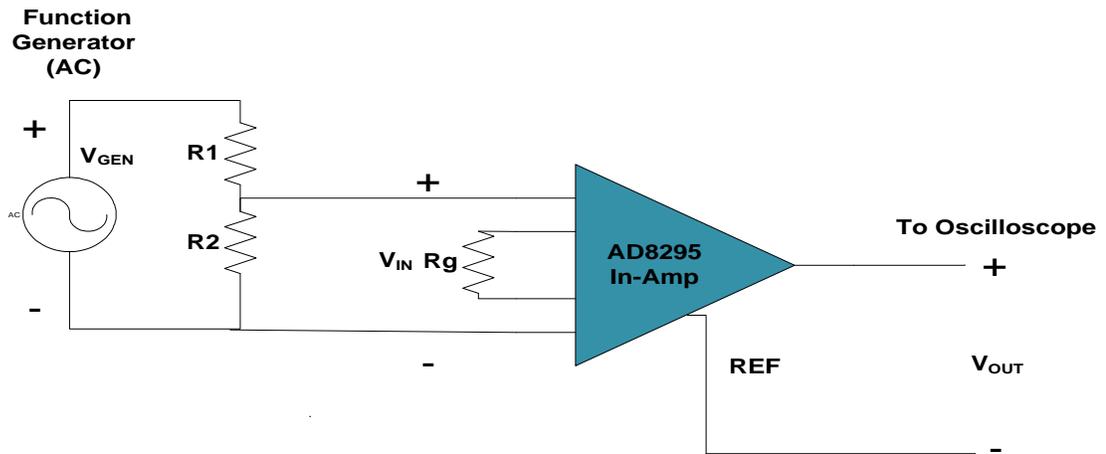


Figure 116 Bandwidth Measurement Schematic

**Table 20: G=1, Bandwidth Measurement**

G=1				
BW				
f	Vout	Vin	G	Gdb
1.00E+02	1.00E+00	1.01E+00	9.90E-01	-8.64E-02
2.00E+02	1.00E+00	1.01E+00	9.90E-01	-8.64E-02
3.00E+02	1.00E+00	1.01E+00	9.90E-01	-8.64E-02
4.00E+02	1.00E+00	1.01E+00	9.90E-01	-8.64E-02
1.00E+03	1.00E+00	1.01E+00	9.90E-01	-8.64E-02
2.00E+03	1.00E+00	1.02E+00	9.80E-01	-1.72E-01
3.00E+03	1.00E+00	1.02E+00	9.80E-01	-1.72E-01
4.00E+03	1.00E+00	1.03E+00	9.71E-01	-2.57E-01
1.00E+04	1.01E+00	1.03E+00	9.81E-01	-1.70E-01
2.00E+04	1.02E+00	1.03E+00	9.90E-01	-8.47E-02
3.00E+04	1.02E+00	1.03E+00	9.90E-01	-8.47E-02
4.00E+04	1.02E+00	1.02E+00	1.00E+00	0.00E+00
1.00E+05	1.03E+00	1.02E+00	1.01E+00	8.47E-02
2.00E+05	1.08E+00	1.03E+00	1.05E+00	4.12E-01
3.00E+05	1.14E+00	1.02E+00	1.12E+00	9.66E-01
4.00E+05	1.25E+00	1.02E+00	1.23E+00	1.77E+00
5.00E+05	1.38E+00	1.02E+00	1.35E+00	2.63E+00
6.00E+05	1.41E+00	1.02E+00	1.38E+00	2.81E+00
7.00E+05	1.27E+00	1.02E+00	1.25E+00	1.90E+00
8.00E+05	1.11E+00	1.02E+00	1.09E+00	7.34E-01
1.00E+06	8.08E-01	1.02E+00	7.92E-01	-2.02E+00
2.00E+06	1.96E-01	1.02E+00	1.92E-01	-1.43E+01
3.00E+06	6.96E-02	1.01E+00	6.89E-02	-2.32E+01
4.00E+06	3.46E-02	1.01E+00	3.43E-02	-2.93E+01
5.00E+06	2.32E-02	9.92E-01	2.34E-02	-3.26E+01

**Table: 21 G=100 Bandwidth Measurement**

Bandwidth G = 100							
Frequency	Vout	Vin	Gain	Gain(db)	Vin(gen)	Atten. Factor	
1.00E+02	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02	
2.00E+02	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02	
3.00E+02	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02	
4.00E+02	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02	
1.00E+03	5.92E+00	6.00E-02	9.87E+01	3.99E+01	6.64E-01	9.04E-02	
2.00E+03	5.92E+00	6.04E-02	9.81E+01	3.98E+01	6.68E-01	9.04E-02	
3.00E+03	5.92E+00	6.04E-02	9.81E+01	3.98E+01	6.68E-01	9.04E-02	
4.00E+03	5.96E+00	6.12E-02	9.74E+01	3.98E+01	6.77E-01	9.04E-02	
1.00E+04	5.92E+00	6.12E-02	9.68E+01	3.97E+01	6.77E-01	9.04E-02	
2.00E+04	5.88E+00	6.12E-02	9.61E+01	3.97E+01	6.77E-01	9.04E-02	
3.00E+04	5.83E+00	6.12E-02	9.53E+01	3.96E+01	6.77E-01	9.04E-02	
4.00E+04	5.79E+00	6.12E-02	9.47E+01	3.95E+01	6.77E-01	9.04E-02	
1.00E+05	5.02E+00	6.12E-02	8.21E+01	3.83E+01	6.77E-01	9.04E-02	
2.00E+05	3.57E+00	6.12E-02	5.84E+01	3.53E+01	6.77E-01	9.04E-02	
3.00E+05	2.53E+00	6.12E-02	4.14E+01	3.23E+01	6.77E-01	9.04E-02	
4.00E+05	1.85E+00	6.12E-02	3.02E+01	2.96E+01	6.77E-01	9.04E-02	
1.00E+06	4.47E-01	6.12E-02	7.31E+00	1.73E+01	6.77E-01	9.04E-02	
2.00E+06	7.90E-02	6.04E-02	1.31E+00	2.34E+00	6.68E-01	9.04E-02	
3.00E+06	1.94E-02	6.04E-02	3.21E-01	-9.86E+00	6.68E-01	9.04E-02	
4.00E+06	6.40E-03	6.04E-02	1.06E-01	-1.95E+01	6.68E-01	9.04E-02	

Table 22: G=1000 Bandwidth Measurement

BW G=1000							
f	Vout	Vin	G	Gdb	vin_gen	atten. factor	
1.00E+02	9.59E+00	9.91E-03	9.68E+02	5.97E+01	1.00E+00	9.91E-03	
2.00E+02	9.59E+00	9.91E-03	9.68E+02	5.97E+01	1.00E+00	9.91E-03	
3.00E+02	9.59E+00	9.91E-03	9.68E+02	5.97E+01	1.00E+00	9.91E-03	
4.00E+02	9.59E+00	9.91E-03	9.68E+02	5.97E+01	1.00E+00	9.91E-03	
1.00E+03	9.72E+00	1.00E-02	9.71E+02	5.97E+01	1.01E+00	9.91E-03	
2.00E+03	9.72E+00	1.00E-02	9.71E+02	5.97E+01	1.01E+00	9.91E-03	
3.00E+03	9.59E+00	1.01E-02	9.49E+02	5.95E+01	1.02E+00	9.91E-03	
4.00E+03	9.40E+00	1.01E-02	9.30E+02	5.94E+01	1.02E+00	9.91E-03	
1.00E+04	8.28E+00	1.01E-02	8.19E+02	5.83E+01	1.02E+00	9.91E-03	
2.00E+04	6.08E+00	1.01E-02	6.02E+02	5.56E+01	1.02E+00	9.91E-03	
3.00E+04	4.59E+00	1.01E-02	4.54E+02	5.31E+01	1.02E+00	9.91E-03	
4.00E+04	3.64E+00	1.01E-02	3.60E+02	5.11E+01	1.02E+00	9.91E-03	
1.00E+05	1.54E+00	1.01E-02	1.52E+02	4.37E+01	1.02E+00	9.91E-03	
2.00E+05	7.60E-01	1.01E-02	7.52E+01	3.75E+01	1.02E+00	9.91E-03	
3.00E+05	4.85E-01	1.01E-02	4.80E+01	3.36E+01	1.02E+00	9.91E-03	
4.00E+05	3.48E-01	1.01E-02	3.44E+01	3.07E+01	1.02E+00	9.91E-03	
1.00E+06	8.64E-02	1.00E-02	8.63E+00	1.87E+01	1.01E+00	9.91E-03	
2.00E+06	1.66E-02	1.00E-02	1.66E+00	4.40E+00	1.01E+00	9.91E-03	
3.00E+06	4.56E-03	1.00E-02	4.56E-01	-6.83E+00	1.01E+00	9.91E-03	
4.00E+06	1.68E-03	9.84E-03	1.71E-01	-1.54E+01	9.93E-01	9.91E-03	

## APPENDIX I - MSP430F169 Code

```
/******
//-----IC APPLICATION TEAM - NECAMSID LAB WPI -----
//-----MQP 2008-----
//This is the code used for the MSP430F169 uController for the temperature based tilt meter
//Main functionality:
//1. Sample the input signal from AD8295 Instrumentation Amplifier
// using the integrated 12bit Analog to Digital Converter (ADC)
//2. Multiplex between the two sensors
//3. Drive the Reference pin of the instrumentation Amplifier
// using the integrated 12 bit Digital to Analog Converter (DAC)
//4. Send the samples through the UART to the FT232RL which then transfers automatically
// the samples to a PC through USB
//!!!Baud Rate used is 460800 !!!
//
//
// IC Application Team - Vasil Savov
// December 10th 2008
// Rev 2.0
//*****
#include <msp430x16x.h>
#define SendConverted
#define Num_of_Results 2
#define DELTA 1024 // Target DCO = DELTA*(4096) = 4*1048576
#define SizeOfResults 9
static unsigned int results[Num_of_Results]; // Needs to be global
static unsigned int resultsAVG[Num_of_Results]; // Needs to be global
char ResultsString[9];
char count = 0;
void ADCconversion();
void RefCalibration();
void Set_DCO();
void Initialize();
void UARTsendData();
void Convert();
char ReadyToSend=0;
char WhichToSend=0;
int DAC_Channel1 = 2700; // The Ref voltage for channel 1
int DAC_Channel2 = 2700; // The Ref voltage for channel 2
char DAC_Calibrated = 0;
char CalCounter = 0;
//-----MAIN -----//
void main(void)
{
Initialize();
RefCalibration(); // Set the Voltage for the ref pin
ADCconversion(); // Start Sampling
ME1 |= UTXE0 + URXE0; // Enabled USART0 TXD/RXD
UCTL0 |= CHAR; // 8-bit character, SWRST=1
UTCTL0 |= SSEL1; // UCLK = SMCLK
UBR0 = 0x09; //
UBR1 = 0x00; //
UMCTL0 = 0x00; // No modulation
UCTL0 &= ~SWRST; // Initialize USART state machine
DMA0SA = (int) ResultsString; // Source block address
DMA0DA = U0TXBUF_; // Destination single address
DMA0SZ = SizeOfResults; // Block size (discard trailing 0)

UARTsendData();
while (1)
{
// Loop here forever - everything happens in the interrupts
}
}
//=====Function Definitions=====//
//-----Setting the 12 bit ADC -----//
void ADCconversion()
{
ADC12CTL0 = ADC12ON+SHT0_5 + MSC + REFON + REF2_5V; // Turn on ADC12, set sampling time
```

```

ADC12CTL1 = CSTARTADD_1 + SHP + CONSEQ_3; // Use sampling timer, set mode
ADC12IE = 0x01; // Enable ADC12IFG.0
ADC12CTL1 |= 0x0010; // |_ Use the MCLK
ADC12CTL1 &= 0xFFF7; // |
ADC12CTL0 |= ENC; // Enable conversions
ADC12CTL0 |= ADC12SC; // Start conversion
_BIS_SR(GIE); //Enable interrupts
}
//-----General Initialization -----//
void Initialize()
{
WDTCTL = WDTPW+WDTHOLD; // Stop watchdog timer
// MSP430F169 specific initialization
P1OUT = 0x10; // disable audio output stage
P1DIR = 0x3f; // P1.6/P1.7 inp, all other outp
P2OUT = 0x00; // clear P2 output register
P2DIR = 0xff; // unused pins as output
P3SEL = 0x30; // P3.4,5 = USART0 TXD/RXD
P3OUT = 0x00; // clear P3 output register
P3DIR = 0xd5; // all but I2C pins to output
P4OUT = 0x00; // clear P4 output register
P4DIR = 0xff; // unused pins as output
P5OUT = 0x03; // clear P5 output register
P5DIR = 0xff; // unused pins as output
P6OUT = 0x00; // clear P6 output register
P6DIR = 0xf0; // unused pins as output
P6SEL |= 0x0f; // Enable A/D channel A0
P5OUT&= 0xFB; // Reset the FT232 RL Procedure
P5OUT|= 0x04; // Reset is P5.2
Set_DCO(); // Set the DCO
BCSCTL1&=0x0F; // Turn XT2 on ; enter High Freq Mode ; ACLK div = 1
BCSCTL1 |=0x40; //
BCSCTL2&=0x8F; // Select the XT2 for the MCLK ; MCLK div = 1
BCSCTL2|=0x80; //
ResultsString[0]=0x20; // Set a separator for the send data
}
//-----Setting the DCO for the specific BAUD rate -----//
void Set_DCO()
{
unsigned int Compare, Oldcapture = 0;
BCSCTL1 |= DIVA_3; // ACLK = LFXTLCLK/8
CCTL2 = CM_1 + CCIS_1 + CAP; // CAP, ACLK
TACTL = TASSEL_2 + MC_2 + TACLK; // SMCLK, cont-mode, clear
while (1)
{
while (!(CCIFG & CCTL2)); // Wait until capture occurred
CCTL2 &= ~CCIFG; // Capture occurred, clear flag
Compare = CCR2; // Get current captured SMCLK
Compare = Compare - Oldcapture; // SMCLK difference
Oldcapture = CCR2; // Save current captured SMCLK
if (DELTA == Compare)
{
break; // Break loop
}
else if (DELTA < Compare)
{
DCOCTL--;
if (DCOCTL == 0xFF) // DCO is too fast, slow it down
if (BCSCTL1 & 0x07) // Did DCO role under?, Sel lower RSEL
BCSCTL1--;
else
break; // Error condition, break loop
}
else
{
DCOCTL++; // DCO is too slow, speed it up
if (DCOCTL == 0x00)
if ((BCSCTL1 & 0x07) != 0x07) // Did DCO role over? Sel higher RSEL
BCSCTL1++;
else
break; // Error condition, break loop
}
}
}

```



```

P5OUT |= 0x01;
DAC12_ODAT = DAC_Channel1;
for(int i=0;i<5100;i++)
{
i
++;
}
WhichToSend =0;
if(!DAC_Calibrated)
{
resultsAVG[0]+=results[0];
resultsAVG[1]+=results[1];
CalCounter++;
if(CalCounter == 20)
{
resultsAVG[0]=(resultsAVG[0]*1.328)/CalCounter;
resultsAVG[1]=(resultsAVG[1]*1.328)/CalCounter;
DAC_Channel1 = 5400 - resultsAVG[1];
DAC_Channel2 = 5400 - resultsAVG[0];
DAC_Calibrated =1;
}
}
else{
Convert();
UARTsendData();
while(!(U0TCTL&0x1))
{
}
}
_NOP();
}
for(int i=0;i<17000;i++)
{
i
++;
}
}

```

```

// Select Channel 2
// Select Output of the DAC to center Channel 2

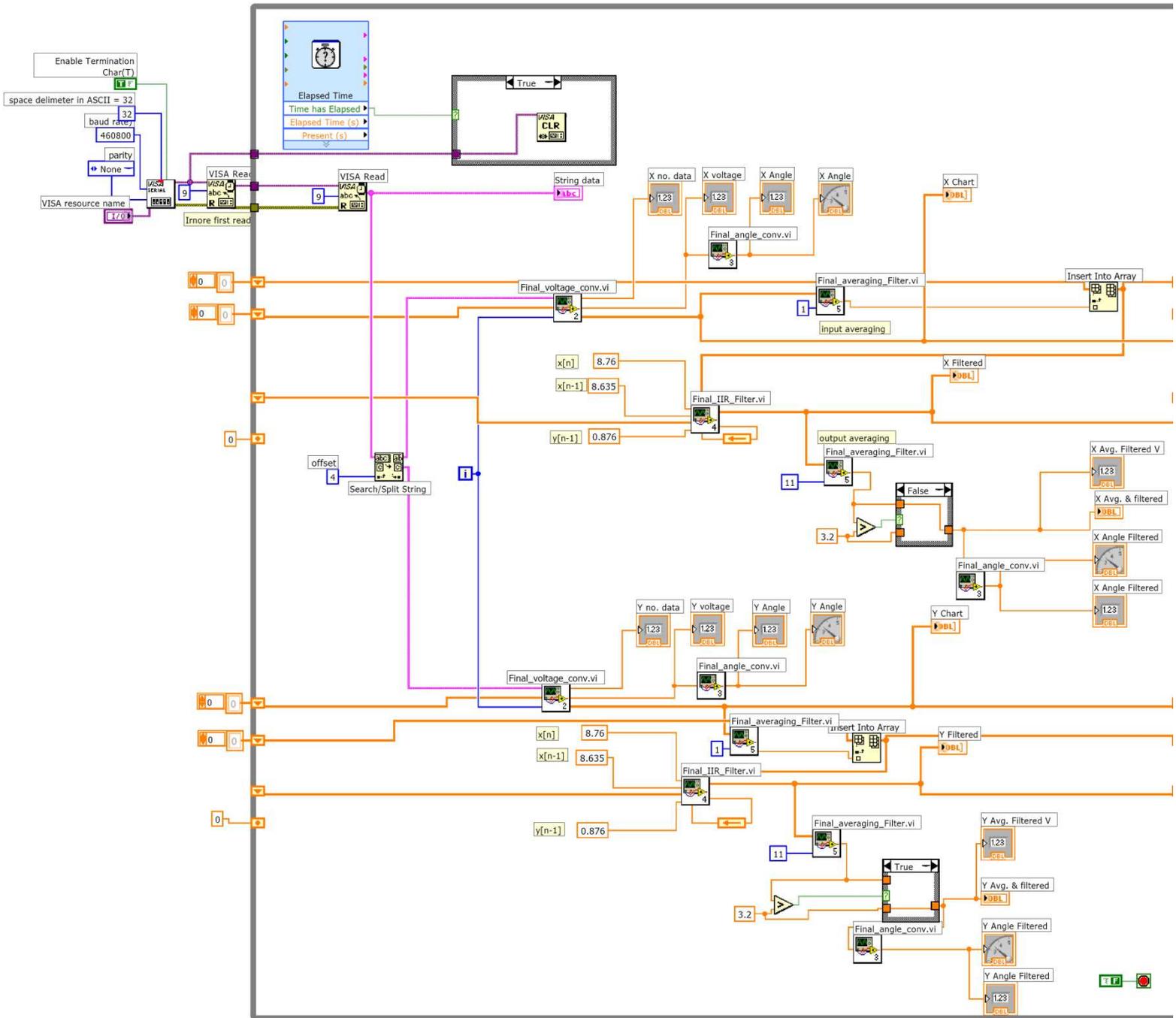
// Do the calibration of the Reference if not calibrated

```

## APPENDIX J – Part List

Part Name	Description	Value	Manufacturer Part No	Unit Price \$	Qty per Board
Crystal	Crystal	32.768 KHz	CFS308-32.768KDZBB	0.15	1
USB	USB connector	mini-B	51387-0578	1.74	1
Crystal	Crystal	8 MHz	HCM49-8.000MABJ-UT	0.75	1
TC7660	Charge pump	N/A	TC7660COA	0.96	1
LED2	Led	RED color	SML-LX1206IW-TR	0.2	2
Button	Tactile Switch	N/A	SWT3-G2GTR	0.54	1
AD8295	Instrumentation Amplifier	N/A	AD8295ACPZ-RL	2.89	1
FT232TRL	USB chipset	N/A	FT232RL	4.5	1
ADG659	4/8 channel MUX	N/A	ADG659YRUZ	0.86	1
MSP430	Microcontroller	N/A	MSP430F169IPM	13.19	1
Thermistor	NTC thermistor	20k Ohm	203JG1J	0.8	4
Capacitor	Tantalum	10uF	TAJR106M006R	0.65	6
	Ceramic	0.1uF	C0805C104M5RACTU	0.05	3
	Ceramic	20pF	GRM2165C1H200JZ01D	0.11	1
	Ceramic	47nF	C0805C473K5RACTU	0.06	1
Resistor	SMD resistor	47k Ohm	260-47K-RC	0.04	5
	SMD resistor	330 Ohm	260-330-RC	0.04	1
	SMD resistor	560 Ohm	292-549-RC	0.04	1
	¼ watt resistor	51 ohms	CFS1/4CT52R510J	0.05	2

# APPENDIX K – LabView Scope Reading Diagram



# APPENDIX L - LabView Maze Game Diagram

