# STRUCTURALLY INTEGRATED EMBEDDED SYSTEM

Thesis

Submitted to

The School of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Electrical Engineering

By

David L. Zeppettella

UNIVERSITY OF DAYTON

Dayton, Ohio

December 2011

STRUCTURALLY INTEGRATED EMBEDDED SYSTEM

Name: Zeppettella, David L.

APPROVED BY:

John G. Weber, Ph.D.
Adviser Committee Chairman
Professor
Electrical & Computer Engineering

John S. Loomis, Ph.D.
Committee Member
Associate Professor
Electrical & Computer Engineering

Ralph Barrera, D.E.
Committee Member
Adjunct Professor
Electrical & Computer Engineering

John G. Weber, Ph.D.
Associate Dean
School of Engineering

Tony Saliba, Ph.D.
Dean, School of Engineering
& Wilke Distinguished Professor

# ABSTRACT

STRUCTURALLY INTEGRATED EMBEDDED SYSTEM

Name: Zeppettella, David L.
University of Dayton

Advisor: John G. Weber

New additive manufacturing techniques such as Direct Write, combined with the continually decreasing size of electronic components has opened new application areas. One such application is the integration of electronic systems with an aircraft's mechanical structure for the purpose of monitoring structural health and sensing the aerodynamic conditions surrounding the vehicle. Data from such a system could be provided to a flight control system to enable new control algorithms to address conditions such as gust loads or simply to improve fuel efficiency through minute attitude adjustments. This paper investigates structural integration techniques and demonstrates through laboratory experiments that the concept is feasible.

To my family.

Shi-kin Hara-mitsu Dai-ko-myo

## ACKNOWLEDGEMENTS

English clergyman John Donne is the author of a passage containing the famous phrase "No man is an island, entire of itself", and nowhere is that sentiment more true than in the undertaking of an engineering effort. This thesis project could not have been accomplished without the assistance of many people in academia, government, and industry. First and foremost, I appreciate the knowledge and experience shared with me by my advisor, Dr. John Weber. His advice and guidance in everything from aircraft embedded systems to contract management was essential in the completion of this project. Additionally, I am grateful to my committee members Dr John Loomis and Dr. Ralph Barrera for reviewing the work.

I would also like to thank Dr. C. Paul Christensen of Potomac Photonics for his efforts in fabricating the Direct Write circuit boards used in this work. His willingness to make changes to the circuit layout and sheer determination to produce functional circuits were key factors in the success of the project.

I also would like to recognize my colleagues in the Air Vehicles Directorate of the Air Force Research Laboratory for the support I received during my work on this project. Mr. Brian Smyers of the Facility for Innovative Research in Structures Technology was essential in the design of test coupons and in lining up laboratory resources to carry out the mechanical tests. Thanks to Mr. Richard Wiggins and Lt. Erin Rogers for their help in operating the MTS equipment in the lab. Thanks as well to Mr. Bill Baron and Mr. Jim

Tuss, who were forced to carry a heavier workload for the Multifunctional Structures Team during my absence to conduct this research.

Finally, I would be remiss if I did not acknowledge the encouragement and understanding of my wife, Aneta, and my daughter, Carla. The time I spent working on this thesis often came at their expense.

## TABLE OF CONTENTS

Content Page

ABSTRACT ... iii

DEDICATION ... iv

ACKNOWLEDGEMENTS ... v

LIST OF FIGURES ... ix

LIST OF TABLES ... xi

CHAPTER 1 INTRODUCTION ... 1

CHAPTER 2 HARDWARE DESIGN ... 5

2.1 Power Supply ... 6

2.2 Bridge Circuit Theory and Design ... 7

2.3 Thermistor Circuit ... 13

2.4 Microcontroller ... 14

2.4.1 Timer Module ... 15

2.4.2 Analog-to-Digital Converter ... 17

2.4.3 $I^2C$ Communication Module ... 21

CHAPTER 3 SOFTWARE DESIGN ... 24

3.1 Microcontroller Firmware ... 24

3.2 Memory Utilization ... 30

3.3 LabVIEW Data Display Program ... 31

3.4 LabVIEW Front Panel ... 31

3.5 LabVIEW Block Diagram ... 32

CHAPTER 4 FABRICATION OF TEST ARTICLES ... 37

4.1 Laser Micromachining ... 37

4.2 Direct Write with Micro-Dispensing ... 38

4.3 Fabrication of Panels with Parasitic Electronics ... 39

4.4 Fabrication of Panels with Integrated Electronics ... 42

CHAPTER 5 TESTING ........ 45
5.1 Physical Set-Up ........ 45
5.2 Test Procedure ........ 47
CHAPTER 6 RESULTS ........ 49
6.1 Micro-Pen Direct Write ........ 49
6.2 Laser Micromachining ........ 53
6.2.1 Electrical Considerations ........ 53
6.2.2 Results of Mechanical Tests ........ 57
6.2.2.1 Specimen 1 57
6.2.2.2 Specimen 2 58
6.2.2.3 Specimen 3 60
6.2.2.4 Specimen 4 62
CHAPTER 7 INTERFACE TO THE 1553 AVIONICS BUS ........ 65
7.1 MIL-STD-1553 Overview ........ 65
7.2 Hardware Considerations ........ 68
7.3 Hardware Connections ........ 69
7.4 Firmware Considerations ........ 71
CHAPTER 8 FUTURE WORK AND CONCLUSIONS ........ 75
8.1 Future Work ........ 75
8.2 Conclusion ........ 77
BIBLIOGRAPHY ........ 78
APPENDICES ........ 81
Appendix A: Block Diagram of Microcontroller ........ 81
Appendix B: Block Diagram of Microcontroller ADC Peripheral ........ 82
Appendix C: Block Diagram of Microcontroller I2C Peripheral ........ 83
Appendix D: Firmware Structure ........ 84
Appendix E: Circuit Layout ........ 85

## LIST OF FIGURES


Figure Page

Figure 1.1 Physically embedded electronic circuit [7]. ........ 2

Figure 2.1 Schematic diagram of the application circuit. ........ 6

Figure 2.2 Quarter bridge strain gage circuit. ........ 8

Figure 2.3 Plot of bridge linearity. ........ 9

Figure 2.4 Block diagram of Timer 1. ........ 15

Figure 2.5 Waveform observed during measurement of Timer 1 period. ........ 17

Figure 2.6 Model of ADC input. ........ 20

Figure 2.7 Photograph of NI USB-8451 ........ 22

Figure 2.8 Photograph of conventional breadboard. ........ 23

Figure 3.1 Data management scheme implemented in firmware. ........ 25

Figure 3.2 Oscilloscope display showing 10 frames. ........ 26

Figure 3.3 Measurement of slack time in frame 10. ........ 27

Figure 3.4 Ten frame sequence after firmware change. ........ 29

Figure 3.5 Measurement of processing time in frame 10. ........ 29

Figure 3.6 Memory Usage Gauge for the application firmware. ........ 30

Figure 3.7 Front panel for the user interface. ........ 32

Figure 3.8 Beginning of the user interface block diagram. ........ 33

Figure 3.9 Data acquisition and display segment of block diagram. ........ 34

Figure 3.10 Block diagram for converting binary data to engineering units. ........ 35

Figure 3.11 Final segment of the user interface block diagram. ........ 36

Figure 4.1 Direct Write with the laser micromachining process[22] ........ 38

Figure 4.2 Deposition of material with a micro-dispensing nozzle. ........ 39

Figure 4.3 Photograph of circuit fabricated by Potomac Photonics. ........ 41

Figure 4.4 Completed test coupon for Potomac Photonics circuit. ........ 41

Figure List of Figures (Continued) Page

Figure 4.5 Composite lay-up concept incorporating a spacer layer. .......... 43
Figure 4.6 Layout proposed for panel with integrated electronics. .......... 44
Figure 4.7 Fiberglass panel produced by SI2 Technologies. .......... 44
Figure 5.1 Block diagram of test set-up. .......... 45
Figure 5.2 Photograph of laboratory physical set-up. .......... 46
Figure 5.3 Close up of test article in the loading fixture. .......... 47
Figure 6.1 Manufacturing flaw on Direct Write circuit trace. .......... 51
Figure 6.2 Cracks in Direct Write traces and Mylar dielectric. .......... 52
Figure 6.3 Profilometer scan showing a crack in a circuit trace. .......... 52
Figure 6.4 Potentiometer added to balance the bridge circuit. .......... 54
Figure 6.5 Noise signal observed on the conventional breadboard. .......... 56
Figure 6.6 Noise signal observed in a circuit fabricated with direct write. .......... 56
Figure 6.7 Photograph of microcontroller pins .......... 62
Figure 7.1 Single level 1553 network. .......... 66
Figure 7.2 Format of MIL-STD-1553 command word [32]. .......... 67
Figure 7.3 Format of the MIL-STD-1553 status word [32]. .......... 67
Figure 7.4 Communication between BC and RT[32]. .......... 68
Figure 7.5 Schematic for the Avionics Bus Interface. .......... 69
Figure 7.6 Organization of the Descriptor Table. .......... 74
Figure 8.1 Strain sensors fabricated on a flexible substrate. .......... 76

# LIST OF TABLES

Table Page

Table 3.1 Measured execution times for frame 10. ............ 28

Table 3.2 Measured execution times after firmware change. ............ 30

Table 5.1 Test sequence used for cyclic fatigue experiment. ............ 48

Table 6.1 Summary of test results. ............ 57

Table 6.2 Applied load and number of cycles for specimen 1. ............ 58

Table 6.3 Results for the first 6 load levels on specimen 2. ............ 59

Table 6.4 Results for the tests on specimen 2. ............ 60

# CHAPTER 1

# INTRODUCTION

The ever-diminishing size of electronic devices continues to open up new applications areas for electronic systems. In particular, as the capability of microprocessors has grown and integrated circuit package size has decreased, embedded systems have become essential in the operation of everything from convenience items such as cell phones and digital cameras to critical systems such as aircraft flight control systems and medical implants. The intent of this work is to propose yet another application for an embedded system in which the electronic system itself is physically integrated with a mechanical structure.

There is a great deal of interest in the use of structurally integrated electronics to monitor the health of complex mechanical structures such as vehicles and highway bridges [1,2,3]. The implementation of such systems could enable monetary savings for municipalities as well as offering safety benefits for the general public. However, in the case of vehicle-based applications, the use of integrated electronics to monitor the status of the structure presents an additional, and somewhat novel, benefit. Namely, the possibility of using structural status as an input to the vehicle control system.

The inspiration for this effort was drawn from previous work completed at MIT in 1991 and more recently at the Center of Excellence for Advanced Materials at the University of California San Diego (UCSD). The MIT research investigated techniques for embedding a single IC into a graphite composite and showed that electronic devices

can survive the composite fabrication process [4]. Research at UCSD showed that the presence of chip resistors in a 0805 package did not significantly degrade shear strength in a composite panel, however the fatigue life can be affected by the integration method [5]. An additional research effort conducted at UCSD involving finite element analysis showed that the inclusion of devices between the plies of a composite gave rise to resin pockets and caused stress concentrations in the immediate vicinity of the device. These stress concentrations were expected to cause failure of the laminate [6].

Additional work done at the Naval Research Laboratory (NRL) has shown that it is possible to create a functional electronic circuit that is embedded into a substrate [7]. The circuit in that effort was fabricated by embedding electronic components into a polyethermide board and using laser direct write to make circuit connections. Although the application circuit was a very simple astable multivibrator based on the LM555 IC from National Semiconductor, the project nonetheless proved the viability of embedding components into a material. Figure 1.1 is a photograph of the circuit, which consisted of the 555 timer in bare die format, along with additional passive components and two LEDs.

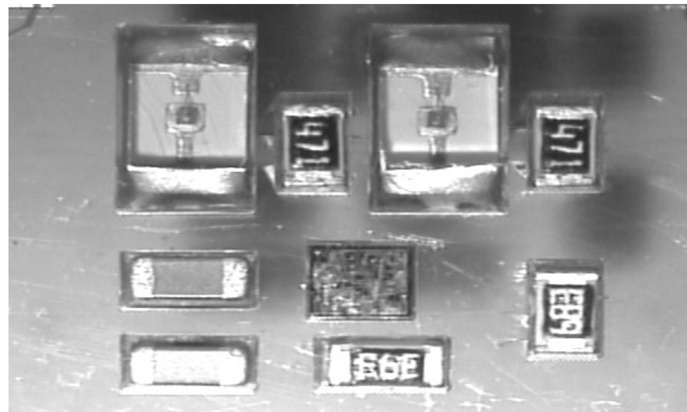


**Figure 1.1** Physically embedded electronic circuit [7].

The purpose of this thesis was to utilize the findings of the research outlined above to examine techniques for integrating an electronic system with a fiberglass

composite panel capable of carrying a mechanical load. Two separate integration methods were considered – a parasitic approach in which the electronics were fabricated on a polyimide film that was subsequently bonded to the test panel, and an embedded approach in which the application circuit was encased in the fiberglass panel itself. Because the application under consideration was structural state sensing, the sensors selected were a strain gage and a thermistor. It should be noted that the cost of these sensors was a significant consideration in their selection, and they would not be the best choice in a practical application.

The project was divided into three activities: design and testing of a conventional prototype circuit (breadboard) and associated software, construction of panels containing integrated electronics, and mechanical testing of panels with the integrated system operational.

One of the essential processes in this endeavor was the creation of the electrically conductive networks to form the circuit interconnections. As in the NRL research, Direct Write technology was incorporated for this task. This technology is an additive manufacturing technique in which a conductive material is deposited on a substrate to form a conductive path. Two different Direct Write processes were utilized in order to assess the suitability of each method for the application.

The primary focus of this effort was on the survivability of the microcontroller and supporting integrated circuits during the fabrication of the panel and subsequent fatigue testing. Although the electronics system was designed to produce reasonable accuracy, efforts were made to reduce the design to a minimally sufficient configuration in order to reduce the number of components that needed to be embedded in the panel. The rationale being that a less complex design would promote easier fabrication, and any measurement errors or performance enhancements could be addressed after the concept was validated.

This thesis is organized into 8 chapters including the introduction. Chapter 2 discusses the hardware design along with the pertinent microcontroller peripherals used in the design. Chapter 3 covers the software design including both the microcontroller firmware and a user interface program developed in LabVIEW. Chapter 4 describes the two Direct Write processes used in the research and discusses the fabrication of the mechanical test coupons used in experiments. Chapter 5 provides an overview of the physical test set-up and test procedure. Chapter 6 outlines the results obtained in mechanical tests. Chapter 7 gives a brief description of the MIL-STD-1553 avionics bus and provides a high-level design for interfacing an integrated electronic system to that bus. Chapter 8 concludes the paper.

# CHAPTER 2

# HARDWARE DESIGN

The electronic circuit used for the research is shown in Figure 2.1. The design consists of a strain gage in a Wheatstone bridge with a Texas Instruments INA333 instrumentation amplifier used to amplify the bridge output. A MAX4626 analog switch from Maxim Integrated Products is used to place a shunt resistor in parallel with the strain gage in order to validate circuit operation. To improve the Common Mode Rejection Ratio (CMRR) of the INA333, the amplifier's DC offset voltage is buffered by an operational amplifier. The schematic shows a Texas Instruments OPA2333 in this role; however, the LT6003 amplifier was substituted in one group of panels because of difficulty encountered in obtaining the OPA2333. The temperature sensor circuit is straightforward, consisting simply of a current limiting resistor in series with the thermistor. Circuit control, analog-to-digital conversion, and communication with an external computer are accomplished via a Microchip Technologies PIC24F64GA002 microcontroller. The following sections discuss the design in more detail.

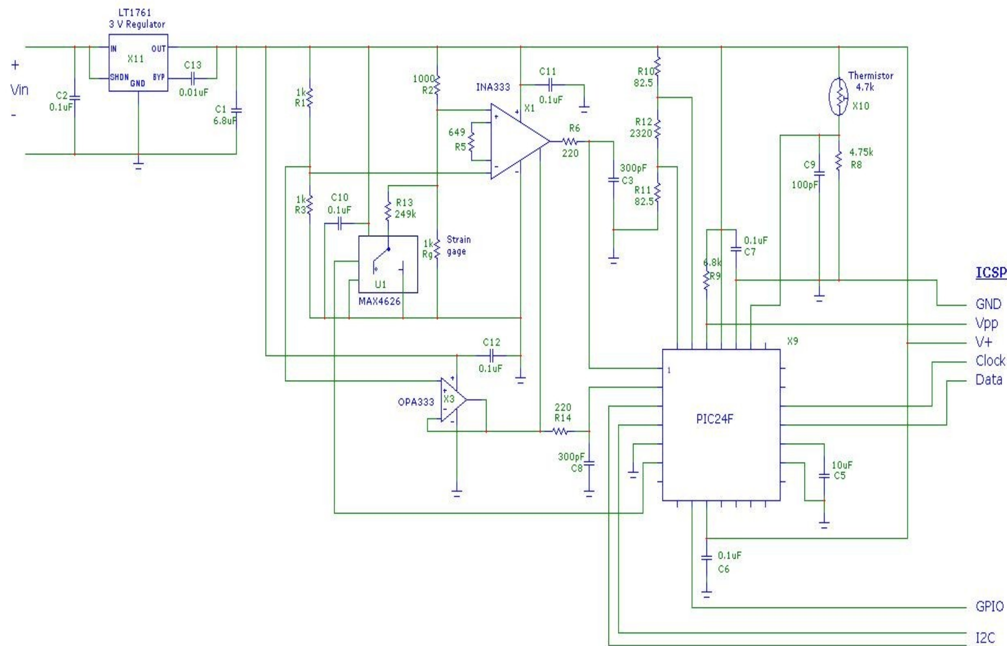



**Figure 2.1** Schematic diagram of the application circuit.

## 2.1 Power Supply

Since one of the design goals was to minimize the component count, no consideration was given to the use of a bipolar power supply. In recent years, manufacturers have been introducing new, low power analog ICs targeted at the wireless sensor network market. Hence, it was felt that the unipolar power supply restriction would not limit the number of ICs available for use in the design.

The most obvious parameter to consider when determining an acceptable power supply voltage is the input voltage range for the ICs used in the design. All of the analog ICs used in this work have a supply input voltage range of 1.8V – 5.5V. The microcontroller core logic requires 2.5 V, but the IC has an on-board voltage regulator that allows an operational voltage range of 2.5V – 3.6V. Thus, with respect to the input supply voltage parameter for the ICs in the design, the microcontroller limits the supply voltage to a maximum of 3.6V.

The sensor biasing circuits also need to be taken into account when considering supply voltage. The desire to minimize components dictates the use of DC circuits for sensor excitation; hence, both the strain gage and thermistor present a purely resistive load to the power supply. Consequently, power dissipation in the biasing networks needs to be considered, particularly in the case where the circuit elements will be encased in a nonconductive fiberglass panel. Additionally, as will be seen in following sections, the output of each sensor biasing circuit is directly proportional to the excitation voltage.

Ultimately, a supply voltage of 3V was chosen, and a Linear Technologies LT1761 voltage regulator was selected for the circuit. One useful feature of this component is that it provides reverse bias protection if the power supply leads are inadvertently reversed, thus eliminating the need for external diodes. The manufacturer's datasheet was used to determine values for the feedback and output capacitors.

## 2.2 Bridge Circuit Theory and Design

Signal conditioning circuits for strain gages typically involve the use of a Wheatstone bridge to detect small changes in the resistance of the gage. The topic of how to excite the bridge tends to be a point of continual debate in the instrumentation field. Excitation can be provided in a number of ways, but two widely accepted approaches for exciting the bridge are by applying either a constant voltage or a constant current. The first phase in designing the strain gage bridge circuit involved determining which approach was best for the application. Although the initial circuit concept used a constant current approach, research showed that because the strain gage being used here is of the metal foil type, the constant voltage method was the better choice in this application, as will be shown in the following discussion.

Since there is only one strain gage in the embedded circuit, the bridge circuit will contain one varying element as shown in Figure 2.2. This configuration is typically referred to as a quarter bridge circuit, and the output is given by

$$V_o = \frac{V_e}{4}\left(\frac{\Delta R}{R+\frac{\Delta R}{2}}\right) \quad (2.1)$$

where $V_o$ = Bridge output voltage
$V_e$ = Bridge excitation voltage
$R$ = Nominal value of bridge elements
$\Delta R$ = Resistance change of gage

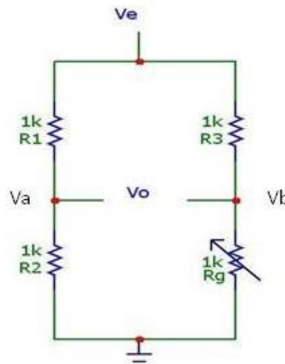



**Figure 2.2** Quarter bridge strain gage circuit.

One of the main concerns with this configuration is that the output of the bridge is nonlinear, as shown by the expression for output voltage[8]. The output equation was rearranged as shown in Equation 2.2, and the bridge output was plotted as shown in Figure 2.3.

$$\frac{\mathrm{Vo}}{\mathrm{Ve}} = \frac{1}{4}\left(\frac{\Delta \mathrm{R}}{\mathrm{R}+\frac{1}{\Delta \mathrm{R}}}\right) \quad (2.2)$$

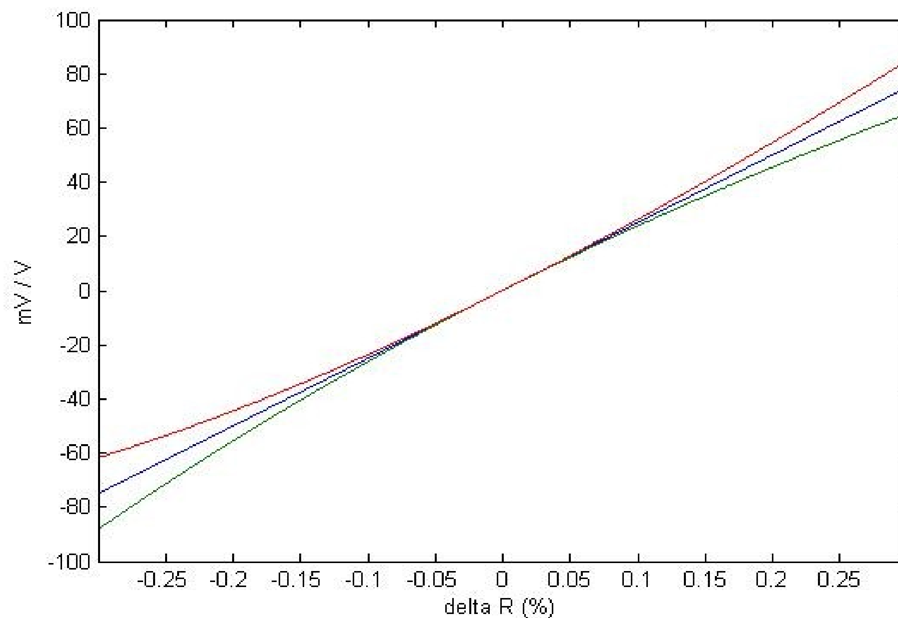



**Figure 2.3** Plot of bridge linearity.

The straight line in Figure 2.3 was created by omitting the 1/ΔR term in the denominator of the Equation 2.2, and the bottom-curved line shows the result of including this term. Hence, the linearity error for the bridge circuit is the difference between the two lower lines shown in Figure 2.3. It can be readily observed from the plot that the bridge non-linearity is most pronounced for large values of ΔR. For a strain gage, this condition would occur when the gage experiences high strain levels.

Until this point, we have considered only the bridge circuit, but the linearity of strain gage itself also needs to be considered. Metal foil strain gages also have a nonlinear characteristic. The nonlinearity of the gage, however, turns out to be opposite that of the bridge circuit, and thus the two effects act in a way to nearly cancel each other [9]. The nature of the strain gage nonlinearity is illustrated by the top curve in Figure 2.3. This “self compensation” effect is obtained with the strain gage in a constant voltage bridge, but the effect cannot be realized if the bridge is excited with a constant current [9].

High strain levels were not planned in the initial testing of the specimens containing embedded circuitry; however, the strain circuit was designed using constant voltage excitation in order to allow future testing at higher strain levels without changing the bridge excitation method. The strain values used in panel testing were not expected to go outside the range of +/- 6000 με, and the bridge amplifier circuit was consequently designed to operate in this range.

With the excitation method determined, the strain gage bridge circuit was analyzed to determine the required gain of the instrumentation amplifier. For a foil strain gage, the resistance of the gage and the measured strain are related as shown in Equation 2.3 [9].

$$\frac{\Delta R}{R} = GF * \epsilon \tag{2.3}$$

where

ΔR = change in resistance

R = unstrained gage resistance

GF = gage factor (sensitvity)

ε = strain

The gage factor for a metal gage is on the order of 2, and the resistance value selected for the design was 1 kΩ. Thus, the range of gage resistance for the expected strain of ± 6000 με can be calculated as follows

$$\Delta R = GF * \epsilon * R \tag{2.4}$$

$$\Delta R = 2 * 6000 * 10^{-6} * 10^{3}$$

$$\Delta R = 12\ \Omega$$

The range of resistance expected is thus 1000 ± 12 Ω. This result can be used to calculate the span of the bridge output, and subsequently the required amplifier gain.

Referring to the bridge schematic in Figure 2.2, the bridge output voltage, $V_0$, is the difference between $V_a$ and $V_b$. By applying voltage division, it can be seen that the bridge output can be represented as

$$V_o = \frac{V_{cc}}{2} - V_{cc}\left(\frac{R_g}{1000 + R_g}\right) \qquad (2.5)$$

The span of the bridge output voltage can be easily determined by substituting the maximum and minimum values of the expected gage resistance as calculated above. Substituting $R_g = 1012$ into Equation 2.5, and recalling that $V_{cc} = 3$ V, we have

$$V_o = \frac{3}{2} - 3\left(\frac{1012}{1000 + 1012}\right)$$

$$V_o = -8.95 \text{ mV}$$

Similarly, for $R_g = 998\ \Omega$,

$$V_o = \frac{3}{2} - 3\left(\frac{998}{1000 + 998}\right)$$

$$V_o = 9.05 \text{ mV}$$

Therefore, the span of the bridge output voltage is approximately ± 9mV.

With the span determined, information from the INA333 datasheet can be utilized to determine the gain resistor value. The datasheet specifies that the output of the INA333 can swing to within 50 mV of the supply rails. It is also important to note that with a supply voltage of 3V, the output of the instrumentation amplifier needs to be referenced to the mid-supply level to allow the output to swing in both directions to indicate positive and negative strain. This was accomplished by simply buffering the voltage divider from the inactive side of the bridge with an operational amplifier. Given the above, it becomes clear that the maximum swing in each direction can be determined by subtracting the limits from the reference level, as shown by Equation 2.6.

$$\text{V} = |\text{V}_{\text{ref}} - \text{V}_{\text{lim}}| \qquad (2.6)$$

Thus, the signal swing for the amplifier is calculated as V = |1.5 V – (3.0 -.05)V| = 1.45 V toward the positive rail, and, similarly, V = |1.5 V – (0+.05)V| = 1.45 V toward the negative rail. The design can be made more conservative by taking the signal swing to

be 1.4 V, and calculating the required gain as the ratio of the maximum amplifier swing to the maximum swing of the bridge output voltage

$$G = \frac{V_{o\ amp}}{V_{o\ bridge}} = \frac{1.4\ V}{9\ mV} = 155.5 \quad (2.7)$$

Once again referring to the INA333 datasheet, the gain equation is specified as

$$G = 1 + \frac{100\ k\Omega}{R_G} \quad (2.8)$$

Rearranging Equation 2.8 and substituting the gain value determined above, we determine the value of the gain resistor.

$$R_G = \frac{100\ k\Omega}{155.5 - 1} = 647.25\Omega$$

Referring to standard 1% tolerance resistor values, the closest value was determined to be 649 Ω. Using Equations 2.8 and 2.9, we determine the resulting gain will be 155.08 and the amplifier output swing will be 9mV * 155.08 = 1.396 V.

The issue of bridge balancing was also a consideration in the design of the bridge circuit. The desire to minimize circuit components and consequently reduce complexity of Direct Write circuit networks drove an initial design decision to eliminate as much bridge imbalance as possible through the use of precision bridge resistors and symmetrical traces in both arms of the bridge. Any remaining offset would then be removed through a shunt calibration procedure controlled by the microcontroller.

Although the maximum bridge deflection is 9 mV (corresponding to a strain of 6,000 με), the initial mechanical test were intended to produce a more moderate strain level of 2,000 με. Since this strain level is one third of the maximum level, the bridge output voltage will also be one third of maximum deflection as can be seen in Equation 2.9.

$$V_o = V_s\left(\frac{k*\varepsilon}{4}\right) \quad (2.9)$$

Thus, an output voltage of 3 mV was used to establish a value for the shunt resistor. The value can be approximated as shown in Equation 2.10 [10].

$$R_s \;=\; R_{ba}\;\left(\frac{V_s}{4*\,V_o}\right) \qquad (2.10)$$

where

$R_s$ = shunt resistor value
$R_{ba}$ = Bridge arm resistor value
$V_s$ = Bridge excitation voltage
$V_o$= Bridge output voltage

Substituting values of $V_s$ = 3V, $V_o$=3 mV, and $R_{ba}$ = 1kΩ into Equation 2.10 leads to a value of 250kΩ for the shunt resistor. A standard 1% value of 249kΩ was selected for this resistor.

An analog switch IC was selected as the means by which the shunt resistor would be connected in parallel with the strain gage, and a Maxim MAX4626 IC was selected. The MAX4626 functions in the normally open mode; with a logic high on the control pin required to close the switch. One pin of the microcontroller was configured as a digital output to operate the analog switch.

## 2.3 Thermistor Circuit

The thermistor circuit is a simple voltage divider consisting of a ballast resistor in series with a thermistor. Some consideration was given to the use of a linearization resistor in parallel with the sensor to create a linear output characteristic as well as expand the temperature range [11]; however, the desire to minimize hardware drove the decision to use a more involved software calculation to determine the temperature.

The primary issue in the design phase was determining a value for the ballast resistor to limit current through the thermistor for the maximum current condition. Because the thermistor is a negative temperature coefficient device, maximum current

occurs at the high temperature limit. For this application, the high temperature limit is determined by the maximum operating temperature of the ICs used in the design. The device datasheets were used to determine that the limiting factor is the operating range of the Maxim MAX4626 analog switch, which has a high temperature limit of 85 C.

To arrive at the value of the ballast resistor, the thermistor datasheet was consulted to obtain values for the maximum operating current, resistance value at 85 C, and maximum power dissipation. After determining these parameters, it is simply a matter of applying Ohm's law to arrive at an expression for the ballast resistor value.

$$R = \frac{V}{I_m} - R_T \qquad (2.11)$$

where

R = ballast resistor value

V = Power supply voltage

$I_m$ = Maximum thermistor current

$R_T$ = Thermistor resistance

In substituting the known values into the equation, a margin of safety was added by taking the maximum current to be 550 μA instead of the maximum current of 650 μA stated on the datasheet. A suitable value for the ballast resistor was determined to be 4.75kΩ after substituting the 3V power supply voltage and the 641Ω thermistor value at 85 C. The power dissipation for the thermistor can be determined using the maximum current and the resistance value at 85C. Carrying out the calculation ($I^2R$ = (550 μA)$^2$(641 Ω) = 193.9 μW) shows that the dissipated power is far below the specified rating of 200mW.

## 2.4 Microcontroller

The microcontroller selected for the application is a PIC24FJ64GA002, and a general block diagram for the PIC24FFJ64GA004 family is shown in Appendix A. The core is a 16 bit CPU utilizing the Harvard architecture with 64KB of program memory

capable of holding 22,000 instructions and 8KB of data memory. The controller has several options for clocking allowing for external or internal clock generation. Once again driven by the desire to minimize components, the internal Fast RC oscillator feature was utilized. This is an on-chip oscillator with a frequency of 8MHz that can be divided down to 31 kHz under software control or, alternatively, used in conjunction with a Phase Locked Loop frequency multiplier to attain clock speeds up to 32 MHz [12]. The clock was set to 4 Mhz for this project. The controller also has a number of peripheral modules, but only three of these resources were needed in the design – the analog to digital converter, two timers, and the Inter-Integrated Circuit ($I^2C$) communications port. The following sections discuss these modules and their role in the design.

### 2.4.1 Timer Module

Although the PIC24F contains 5 timers, only Timer 1 and Timer 2 were utilized in the project. Timer 1 (denoted TMR1 in Microchip's literature) plays a critical role because it is used to establish the desired sample rate for the analog signals. A block diagram of this module is shown in Figure 2.4. As shown in the figure, the module

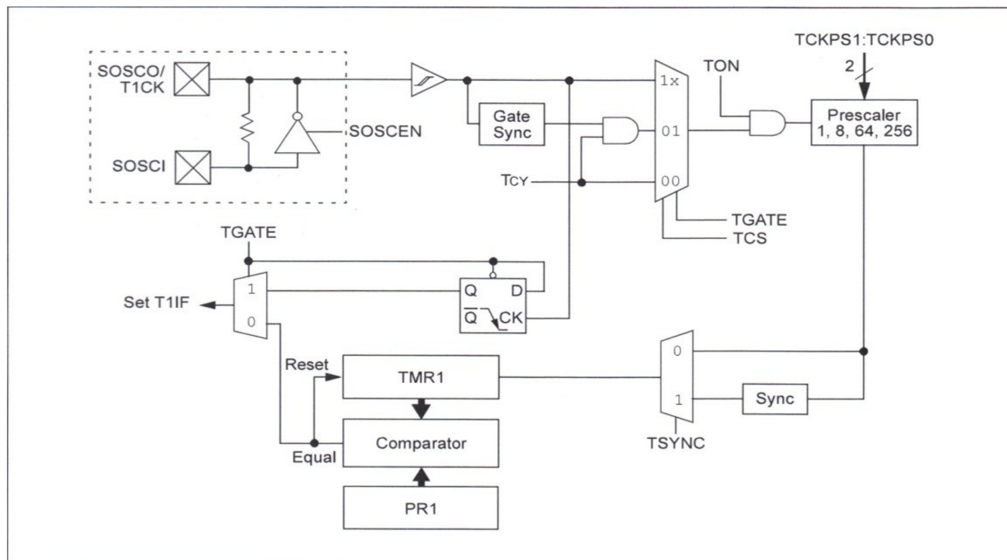



**Figure 2.4** Block diagram of Timer 1.

contains a period register, denoted PR1, which is used to configure the module to generate an interrupt whenever the contents of PR1 match the value of the TMR1 register. To establish a sampling rate at a specific interval, PR1 is loaded with a hexadecimal value representing the period. This value is determined by calculating how many timer increments occur over the sampling period while noting that TMR1 is incremented at half the processor's oscillator frequency, or 2 MHz. The sampling period was determined by assuming that the PIC24 is required to supply data to a notional flight control system at a rate of 250 Hz. In order to minimize aliasing errors, the analog signals would need to be sampled at ten times this value implying that a sampling frequency of 2.5 kHz is required. The value for the period register was determined by dividing the signal sampling period by the period of TMR1: 400μs / 500 ns = 800. This value is then converted to hexadecimal, and it is thus determined that the period register needs to be loaded with $320_{16}$. A short program was implemented on a prototype board to verify timing values, and a digital IO pin was toggled upon entry & exit from the timer 1 interrupt service routine. An oscilloscope was used to capture the signal on the IO pin, and Figure 2.5 shows the observed waveform. As can be seen in the figure, the test confirmed that the period was 400 μs.

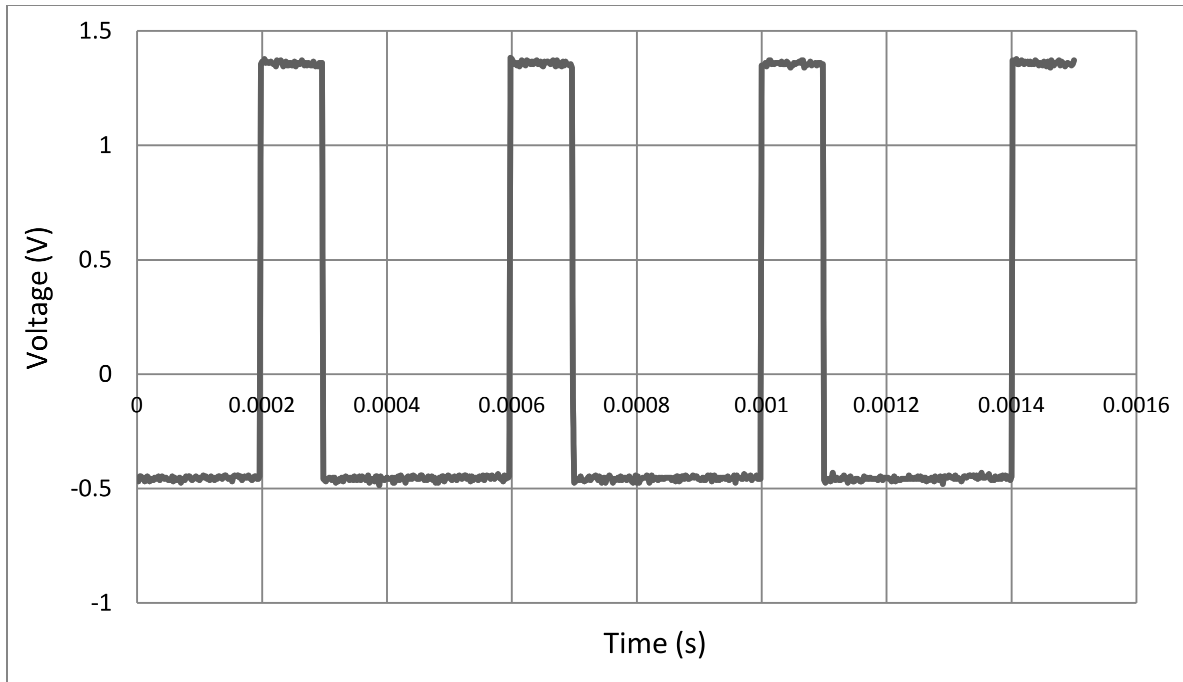



**Figure 2.5** Waveform observed during measurement of Timer 1 period.

An additional timer, Timer 2, was also utilized in the project. Although the PIC24 offers the option to combine Timer 2 and Timer 3 to create a 32 bit timer, Timer 2 was configured for 16 bit operation. This timer played a less critical role in the design, simply being used to create time delays during program execution. One example of where this function was used is to indicate completion of the microcontroller initialization sequence by flashing an LED with one of the digital output pins.

**2.4.2 Analog-to-Digital Converter**

A block diagram of the 10 bit Analog-to-Digital converter (ADC) is shown in Appendix B. There are numerous operational modes available with this module, and this flexibility results in some initial confusion when determining what mode to select and how to configure the ADC for the desired mode.[13] Although the ADC was used in a more basic mode, it was necessary to write values to five different special function

registers to initialize the ADC. The following list outlines the steps required to configure the ADC.

1. Select the pins of the PIC24F to be used for analog inputs.
2. Select data output format and set up automatic timing of conversions
3. Disable channel scanning
4. Select voltage reference, set buffer mode, and select multiplexer
5. Set ADC clock source ,automatic sample time, and select ADC conversion clock period

The following C language function shows how these steps were implemented and concludes with a line of code to enable the ADC module.

```
void initADC()
{
  AD1PCFG = 0xFDF0;          /* AN0 -> AN3 & AN9 are analog */
  AD1CON1 = 0x00E0;          /* Auto convert after sample period */
  AD1CSSL = 0;               /* No scanning */
  AD1CON2 = 0x6000;          /* Vref = external (for strain input) */
  AD1CON3 = 0x1F00;          /* sample time = 31 Tad; ADC conversion T = Tcy */
  AD1CON1bits.ADON = 1;      /* Enable the ADC module */
}
```

As mentioned previously in the strain gage circuit, the output of the instrumentation amplifier swings 1.395V about the mid-supply point. Thus the ADC input pin will see voltages between 100 mV and 2.895 V, and the input span of the ADC module is 0 – 3V. For maximum accuracy, the span of the transducer circuit output needs to match the span of the ADC.[14] To accomplish this, a resistor network was used to establish an external reference for the ADC when reading the strain gage circuit. Thus, each time the strain gage circuit is sampled, it is necessary to switch the ADC reference to the external one. For monitoring the temperature circuit and the voltage monitoring circuit, the ADC internal reference is used. The following C function was used to select the desired input channel, set the reference voltage, and read the ADC.

```
int readADC(channel)
{
  AD1CHS = channel;                          /* Select analog channel */

        if (channel == 3)
           AD1CON2 = 0x6000;             /* use external ref. only for strain channel */
        else
           AD1CON2 = 0x0000;             /* use internal ref. (Vdd, Vss)  */

        AD1CON1bits.SAMP = 1;           /* Begin sampling */
        while (!AD1CON1bits.DONE);     /* wait for conversion to complete */

        return ADC1BUF0;
}
```

Another consideration in the hardware design is the interface of the sensor circuits to the ADC. The schematic in Figure 2.1 shows an RC flywheel circuit on the output of each amplifier. The capacitor in these circuits serves the purpose of providing charge to the sample and hold capacitor in the ADC. Figure 2.6 shows the input model for the ADC as shown in the Microchip datasheet. We note from the input model that the sample and hold capacitor has a typical value of 4.4 pF and the sampling switch has a resistance associated with it at a typical value of 5kΩ. Also worth noting is that the pin has a typical capacitance in the range of 6 – 11 pF, although the data sheet states that this capacitance has negligible effect as long as the external resistance is less than 5 kΩ. The datasheet further states that the accuracy of the ADC is improved by limiting the external impedance to 2.5kΩ. To make the design more conservative, it was assumed that the external circuit would have to charge both the sample and hold capacitor and pin capacitance through the 5kΩ series resistor. Thus, the sample and hold capacitor was assumed to have a value of 4.4pF + 8pF = 12.4 pF, or approximately 13 pF.

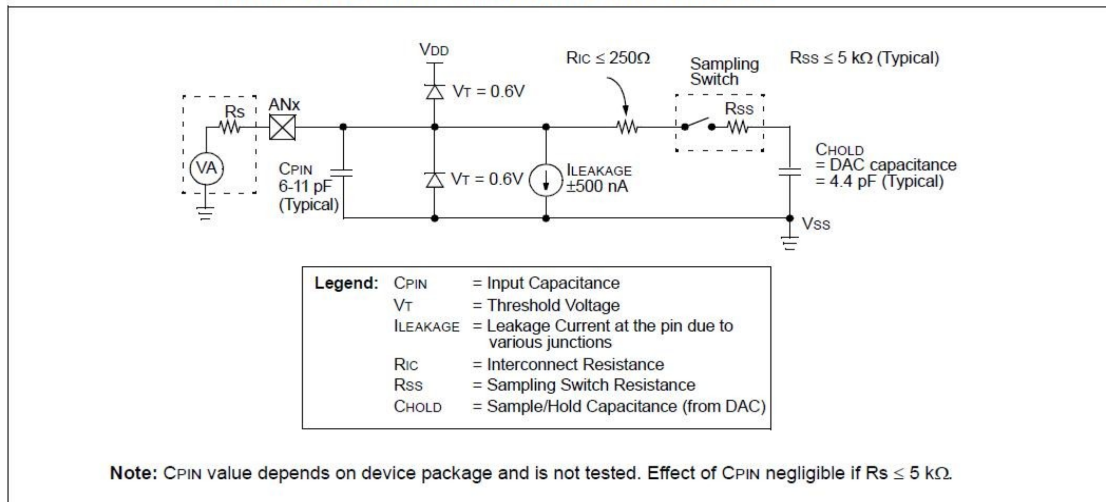



**Figure 2.6** Model of ADC input.

One approach to designing the interface circuit is to assume that 95% of the charge required to charge the sample and hold capacitor will come from the external capacitor and 5% will come from the amplifier.[15] The total charge, Q, required can be determined from the relation Q = CV where C is the capacitance and V is the voltage to which the capacitor is charged. The highest voltage will occur in the strain gage circuit at maximum bridge deflection of 9 mV, causing the amplifier output to swing to 2.8 V. Using this value, Q = 13pF x 2.8V = 36.4 pC. In order for the voltage across the external capacitor to sag by no more than 5% when charging the sample and hold capacitor, we again apply Q = CV and size the external capacitor according to the relation 36.4 pC = C x (.05 * 2.8 V). A value of 260 pF is obtained for C, which is not a standard value. Ultimately a 300pF, 5% capacitor was selected for this the external capacitor. Another design guideline is that the external capacitor should be at least ten times larger than the sample and hold capacitor.[15] A simple calculation shows that the design meets the guideline: 300 pF / 13 pF = 23.08.

The value of the external resistor is selected by considering the acquisition time for the ADC and the fact that the external RC circuit needs to settle in at least the same

amount of time. Additionally, For a 10 bit ADC, the internal RC circuit comprised of the switch resistance and sampling capacitor can be assumed to settle in eight time constants [15]. Thus we impose this requirement on the external RC circuit as well, and we can express the timing relation as shown in Equation 2.12.

$$t_{ext} = t_{acq} = 8\tau_{ext} \tag{2.12}$$

The acquisition time for the ADC can be determined from the Microchip datasheet as 900 ns. The time constant for the external circuit can now be determined using Equation 2.12, and is found to be 112.5 ns. With the time constant determined and the value of C previously chosen as 300 pF, a value for R can be determined by dividing 112.5 ns by 300 pF to obtain 270 Ω. Typical practice is to shorten the time constant of the external time circuit by 30% to allow for amplifier setting time.[15] Here a slightly more conservative resistor value of 220 Ω was selected which results in the flywheel circuit having a time constant 40% shorter than the acquisition time of the ADC.

In the case of the thermistor circuit, a buffer amplifier was not used in order to reduce complexity of the application circuit. However, a capacitor was added in parallel with the ballast resistor to provide charge to the ADC sample and hold capacitor.

### 2.4.3 $I^2C$ Communication Module

The final microcontroller peripheral used in the design was the Inter-Integrated Circuit ($I^2C$) module, and a block diagram of the module is shown in Appendix C. The sole purpose of including this feature in the design was to provide communications with a standalone computer during the mechanical testing phase.

A key component in the communications scheme is the $I^2C$ module that is used to connect the stand alone PC to the bus. In this case, a National Instruments USB-8451 was selected for this role, and the device is shown in Figure 2.7.

**Figure 2.7** Photograph of NI USB-8451.

One of the characteristics of the NI-8451 is that it functions only as an I$^2$C master. Consequently, the microcontroller I$^2$C module was configured for operation as a slave, and the C function used for this purpose is shown below.

```
void initI2C_slave(void)
{
        /* Note: slave mode does not require setting I2CBRG */
   I2C2MSK = 0x0000;        /* no address bits masked - full match required.  */
   I2C2ADD = 0x0010;        /* set up 7 bit slave address */
   I2C2CON = 0x9200;        /* load control register; set enable, sclrel, disslw  */
   IPC12 = 0x0030;          /* set interrupt priority to level 3 (7 = highest) */
   IFS3bits.SI2C2IF = 0;    /* clear slave I2C INT */
   IEC3bits.SI2C2IE = 1;    /* enable slave I2C INT */}
```

The function sets up the address mask, the slave address, and loads the module control register to enable I$^2$C operation. The interrupt priority is set lower than that of Timer 1 to ensure that data acquisition takes precedence over communication functions. The function then concludes by enabling the interrupt for the module.

The final step of the hardware design process was to construct a conventional breadboard circuit in order to validate the hardware design before beginning fabrication of the Direct Write circuit boards. This board also provided a firmware development platform while the Direct Write boards were under fabrication and also served as a reference for noise investigations on the Direct Write circuits.

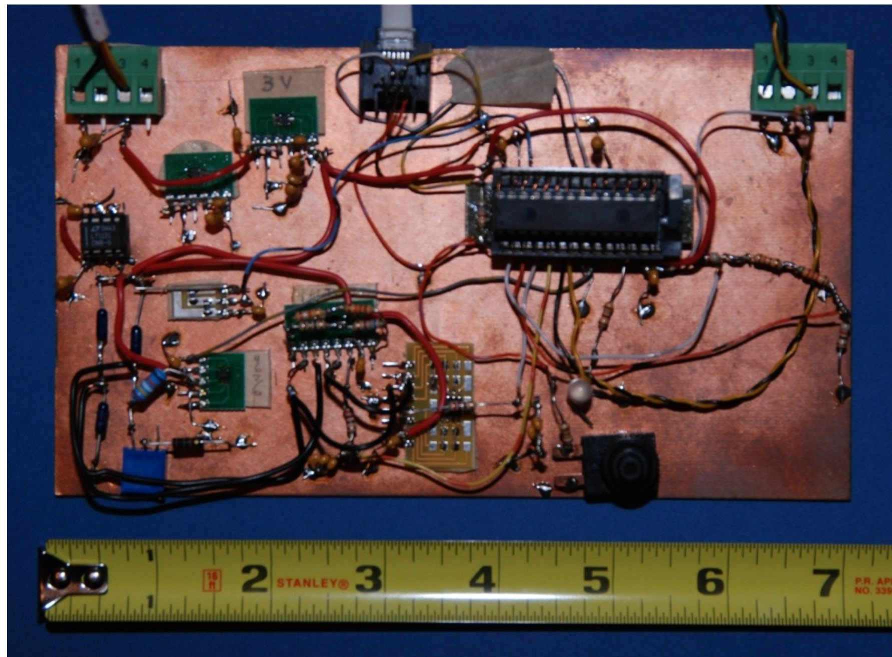



**Figure 2.8** Photograph of conventional breadboard.

# CHAPTER 3

# SOFTWARE DESIGN

The software developed for this project consisted of the microcontroller code written in C and a user interface for a PC that was created with National Instruments LabVIEW graphical programming environment. In both cases, the waterfall model of the software development model was used. This is a linear process involving requirements definition, design of data structures and algorithms, implementation in code, and testing.[16] The C compiler used was the MPLAB C30 evaluation version that is an add-on to Microchip Technologies' MPLAB development environment. An additional piece of development hardware that was critical in the testing and debugging of the C program was Microchip's REAL ICE emulator system.

## 3.1 Microcontroller Firmware

The embedded system in this project is mostly periodic because the data acquisition is being controlled by a timer interrupt, but users can create a` periodic tasks by starting and stopping the data display on the user interface. The implementation scheme for this is a foreground/ background system with data being collected and stored in the foreground, and all tasks related to data processing and communications handled in the background. A flow chart depicting the firmware structure is shown in Appendix D.

The firmware maintains a data structure for each signal being monitored that consists of a ten word vector and at least one integer data member. There is also a ten

byte vector that serves as a transmit buffer used to hold data for $I^2C$ data transmissions. The foreground data collection routine writes sample data to the buffer in the data structure associated with each sensor. When the buffers are full, a flag is set to initiate a data processing task in the background that calculates the average for each buffer and loads the result into the integer member of each data structure. Upon completion of the average value calculations, another background operation separates the words into bytes and loads them into the $I^2C$ transmit buffer. Figure 3.1 depicts this operation graphically.

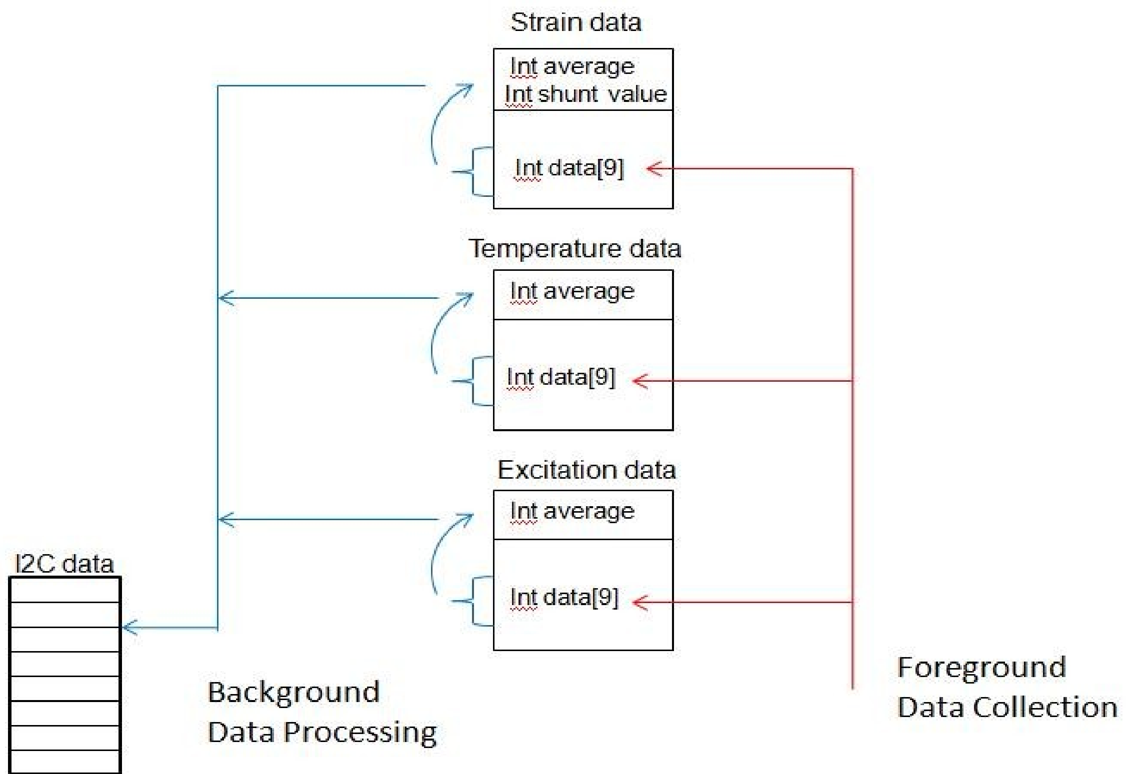



**Figure 3.1** Data management scheme implemented in firmware.

One way of conceptualizing the operation of the firmware is to consider the system mostly cyclic with nine identical frames followed by a single frame that contains a

data acquisition operation followed immediately by the background processing functions. The $I^2C$ interrupt generated by the user display software could be considered as a single a`periodic task handled by the system.

An important consideration in this model is the time required to complete the background processing. If this processing is not completed before the end of the frame period, the foreground process could corrupt the data set by writing a new value into the data buffers. Although this could be avoided by employing a double buffer scheme, the desire was to avoid this additional complexity in the firmware. To investigate any timing problems, code was inserted to toggle an LED at various points during firmware execution, and time measurements were made with an oscilloscope. Figure 3.2 is a screenshot of the oscilloscope trace showing the ten frame sequence. It is evident in the figure that the tenth frame utilizes most of the 400 µs period.

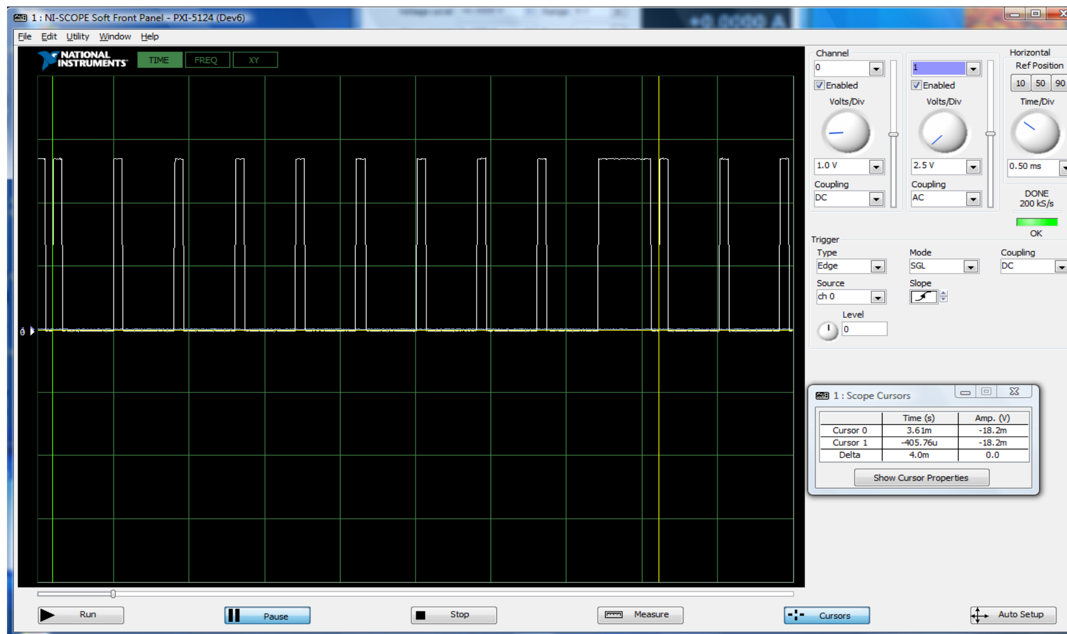



**Figure 3.2** Oscilloscope display showing 10 frames.

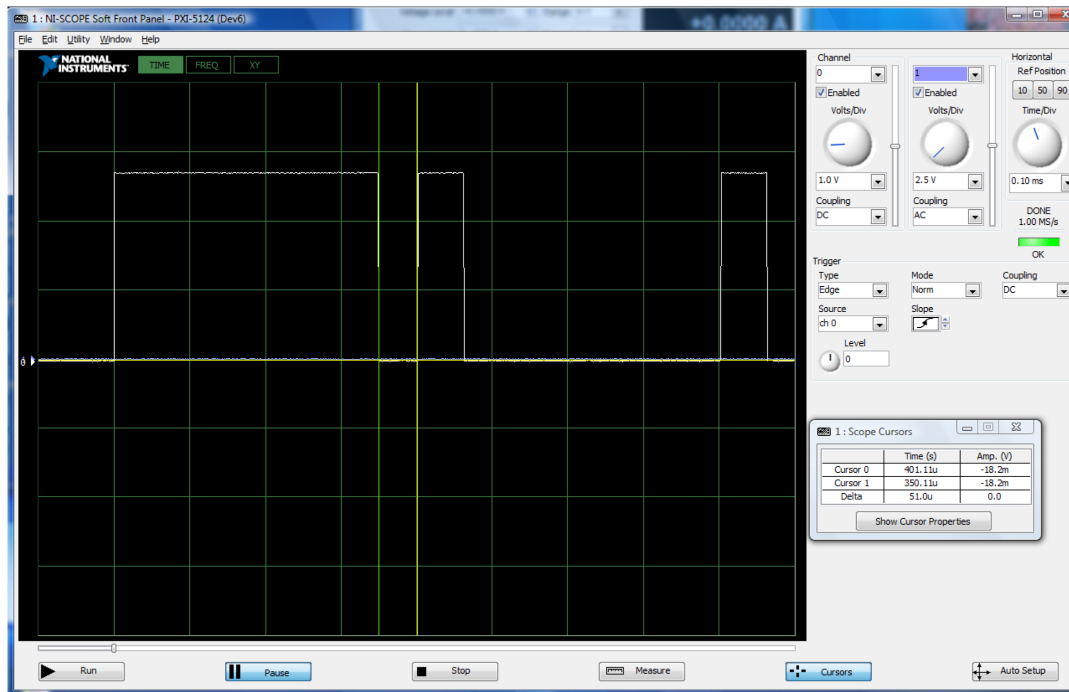



**Figure 3.3** Measurement of slack time in frame 10.

The oscilloscope was used to determine how much slack time remains at the end of the tenth frame, and a screen shot of this measurement is shown in Figure 3.3. The slack time at the end of the tenth frame was determined to be 51 μs using the measurement cursors on the oscilloscope.

Additionally, the execution time of the data acquisition routine, the averaging function, and the $I^2C$ buffer loading function were measured. The results obtained are summarized in Table 3.1, and the data shows that the averaging function is the operation with the longest execution time. It should be noted that the total execution time at the bottom of the table does not include overhead associated with the jump to subroutine on the Timer 1 interrupt.

Thus, although the scheme works in the current configuration, frame over-runs could be encountered should the need arise to monitor more data channels. The execution time of the averaging operation can be modeled as a linear function T(n) = cn,

where n is the size of the input and c is the time associated with arithmetic operations [16]. Consequently, the execution time will grow linearly as more channels are added.

**Table 3.1** Measured execution times for frame 10.

| **Function** | **Time (μs)** |
| --- | --- |
| Data Acquisition | 63.4 |
| Averaging Calculation | 240.0 |
| Load $I^2C$ Buffer | 29.2 |
| **Total** | **332.6** |

One alternative approach would be to update a running sum of each monitored signal during the data acquisition interrupt service routine (ISR), and then simply divide this sum by 10 to get the average in the tenth frame. This would eliminate the need to index through the data array to calculate the sum prior to the division. To investigate this concept, the firmware was modified, and Figure 3.4 is a screen shot of the resulting ten frame sequence. It is clearly evident that the processing load is more evenly distributed and that processing time in the tenth frame is greatly reduced.

The previously described timing measurements were repeated, and Figure 3.5 shows the measurement of the processing time in the tenth frame, which was 164 μs. Subtracting this value from the 400 μs frame period, we see that the processor is now idle for over half the frame period (236 μs), making more processor time available for monitoring additional signals or running analysis routines. Table 3.2 shows the execution times for the operations running in frame ten. The execution time for the data acquisition routine increased by only 8.6 μs, and the execution time of the averaging operation was reduced by a factor of 5 to 45.8 μs.

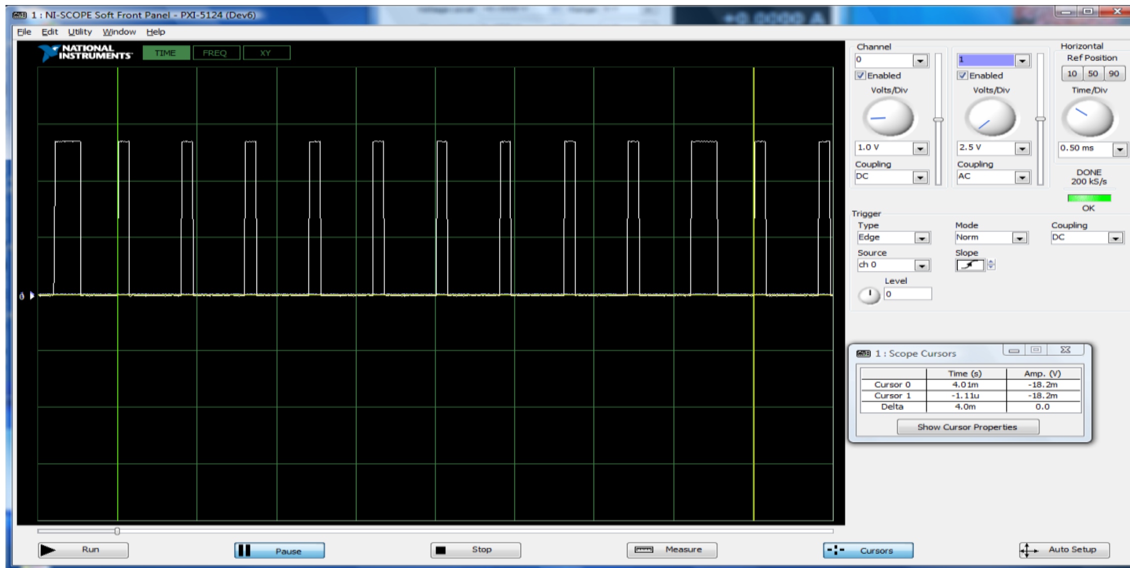



**Figure 3.4** Ten frame sequence after firmware change.

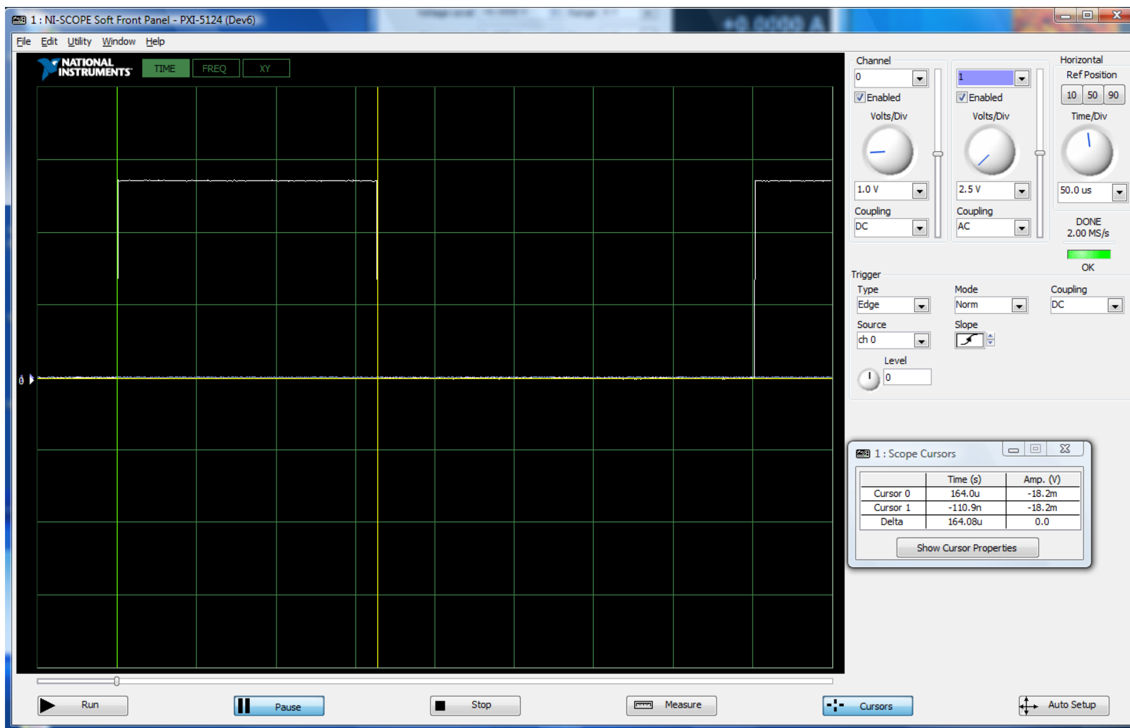



**Figure 3.5** Measurement of processing time in frame 10.

**Table 3.2** Measured execution times after firmware change.

| Function | Time (µs) |
|---|---|
| Data Acquisition | 72 |
| Averaging Calculation | 45.8 |
| Load $I^2C$ Buffer | 29.2 |
| **Total** | **147** |

## 3.2 Memory Utilization

The PIC24FJ64GA002 has a 22,000 word program memory and about 8 kB of data memory .[12] The Microchip development environment contains a feature known as the Memory Usage Gauge that graphically displays how much memory is used by the application code. Figure 3.6 shows the Memory Usage Gauge after compiling a release version of the code. The gauge indicates that 737 words out of 22,014, or about 3.3%, of the program memory has been used, and only 1.3% of the data memory has been used. Therefore there is a significant amount of space to expand system operation in the form of monitoring additional sensors and adding analysis functions to the current program.

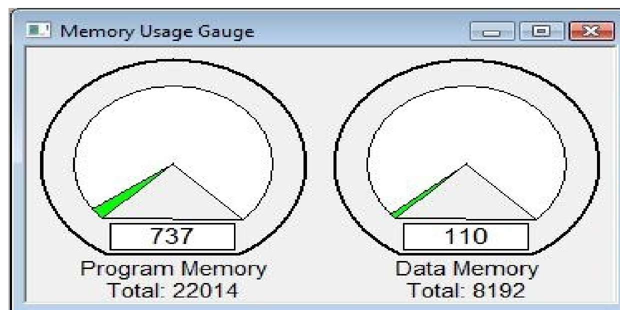



**Figure 3.6** Memory Usage Gauge for the application firmware.

## 3.3 LabVIEW Data Display Program

As mentioned previously, LabVIEW was used to implement a user interface. LabVIEW is a graphical programming environment consisting of a front panel and a corresponding block diagram that contains the source code. Any program or subroutine created with LabVIEW is known as a VI, which stands for Virtual Instrument. This nomenclature is derived from the fact that the front panel typically resembles a laboratory instrument. The program is constructed by interconnecting icons (called nodes) that represent various program functions with lines (known as wires) [17 ]. The characteristic that sets LabVIEW apart from a conventional programming language is the code is not executed sequentially, a given node executes only when all data is present at its input terminals.[18] This concept is known as “data flow”, and the non-sequential order of operations could cause unexpected results in some applications.[19] However, order of execution is not a critical factor in this case because the desire was to create a simple user interface to display data,.

## 3.4 LabVIEW Front Panel

LabVIEW front panels consist of two types of objects – controls and indicators. Controls typically take the form of knobs and switches found on laboratory equipment and allow the operator to input data to the program. Indicators simulate displays on laboratory equipment and are typically numerical displays, meters, and graphs.

The front panel for the user display is shown in Figure 3.7. The main features of the display created for this project are a graph and a thermometer. There are actually two graphs, one on each of two tabs in a tab control; thus allowing users to switch between plots of strain and temperature data. On the left side of the display, two numerical indicators display bridge excitation voltage and the shunt calibration result.

Three controls in the form of buttons allow the user to start and stop the display, log data to a file, and end program execution.

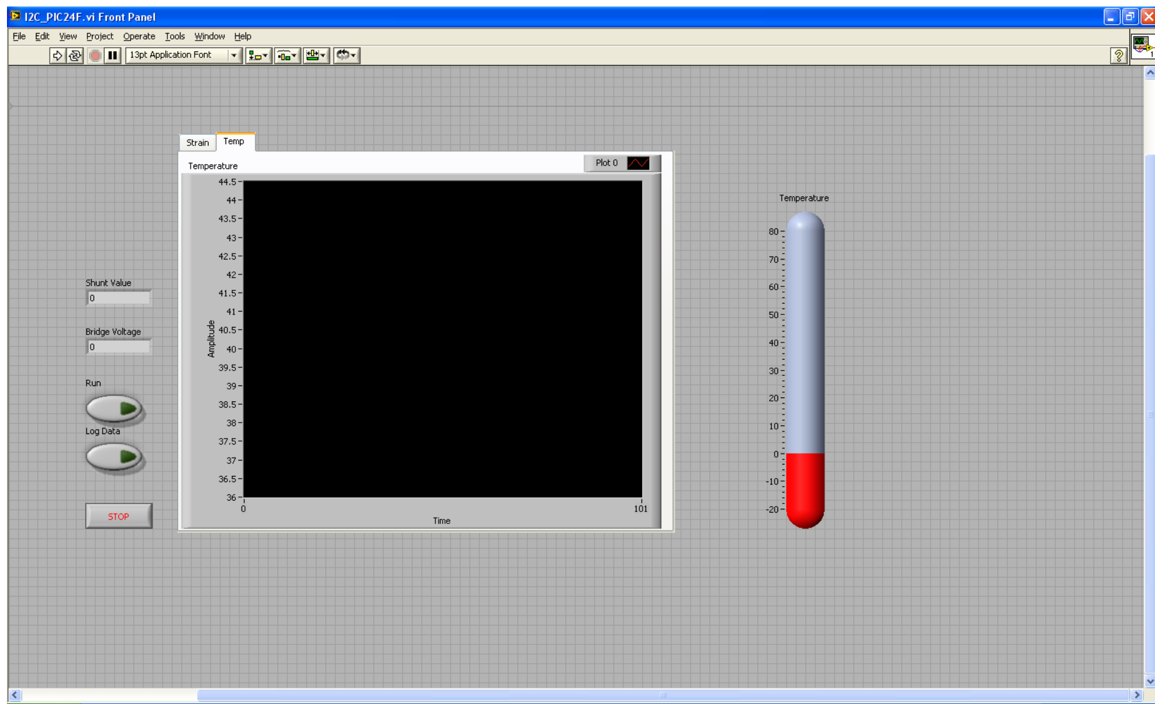



**Figure 3.7** Front panel for the user interface.

## 3.5 LabVIEW Block Diagram

Since the block diagram contains the code, it is important to utilize care in the layout of the diagram. Although the block diagram affords the programmer a great deal of flexibility, a haphazard diagram can result in problems understanding and maintaining the code, and also has efficiency implications. [21] Consequently, time was spent investigating design patterns and layout practices for the block diagram before any code was implemented. Given the simplicity of the user interface application, a continuous loop design pattern was selected. [21] Additionally, several best practices were identified as essential for promoting understanding of the block diagram. These practices include left to right data flow, limit nesting of structures to no more than three levels, and the use of comment blocks to describe the purpose of each portion of the

diagram. [20, 21] The following series of figures shows the block diagram for the user interface program and briefly discusses its operation.

Figure 3.8 shows the first part of the block diagram to be executed when the program is started. The user is prompted to enter a name for a file to which data will be written, and the file is created once the name is entered. The program then creates column headers for the data, and writes the headers to the newly created file. This is the only time these operations will occur as the program next goes into a ‘while’ loop.

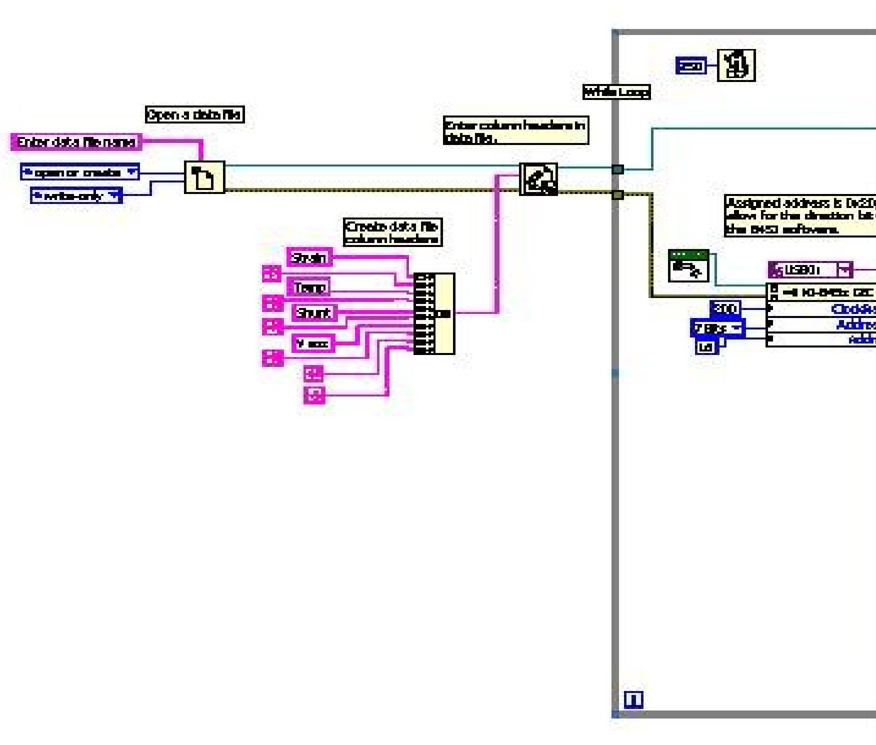



**Figure 3.8** Beginning of the user interface block diagram.

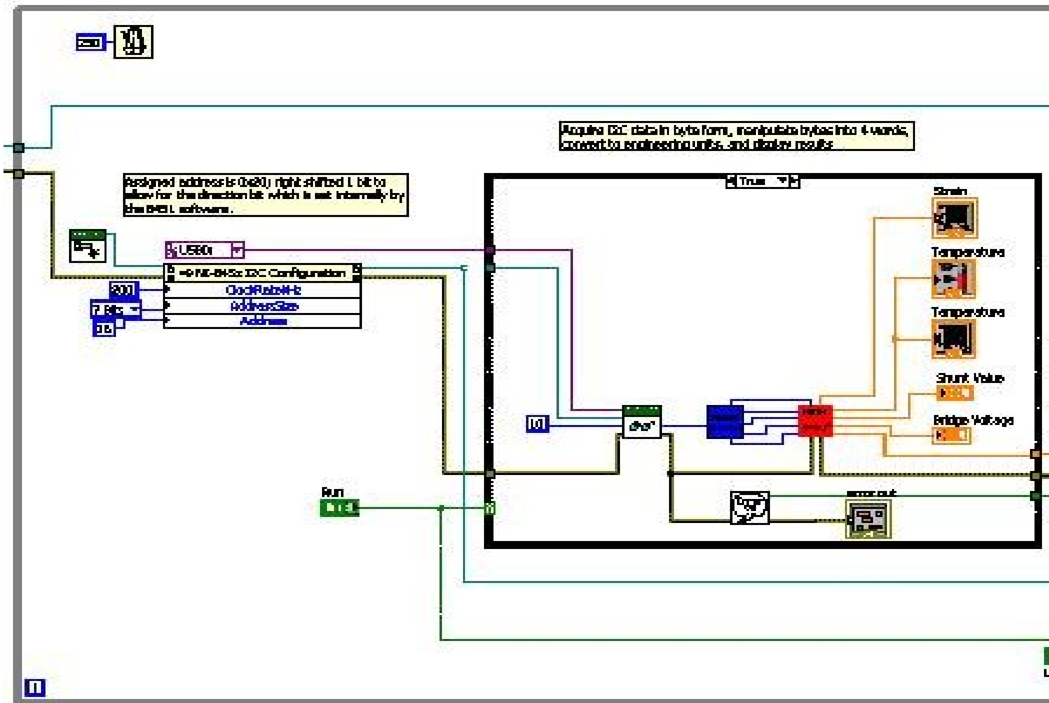



**Figure 3.9** Data acquisition and display segment of block diagram.

Figure 3.9 shows the portion of the block diagram that reads data from the NI USB-8451 I$^2$C module. The program obtains a reference to the module, and sets the address mode, address, and clock rate parameters for I$^2$C communication. When the run button is activated on the front panel, the program begins reading data from the module. Data from the module is formatted as a vector of bytes, and this data needs to be reformatted into words before the conversion to engineering units can be carried out. A separate VI was created for this purpose that outputs four words that are used as the inputs to a conversion routine. The conversion VI is shown in Figure 3.10, and the output of this VI is used to update the indicators on the front panel.

The key feature of the conversion VI is the LabVIEW formula node that is used to carry out the conversions. The three rectangles in the center of Figure 3.10 are the formula nodes. The formula node strongly parallels a function in the C language. Variables passed into the node are on the left side of the structure, a series of sequential

processing steps are entered inside the node, and the output is taken from the right side of the structure. The results are then returned to the user interface VI, and the front panel indicators are updated with new values. This interaction is shown on the right side of Figure 3.9, with the indicators arranged vertically on the right and wires connecting each indicator to an output on the conversion VI.

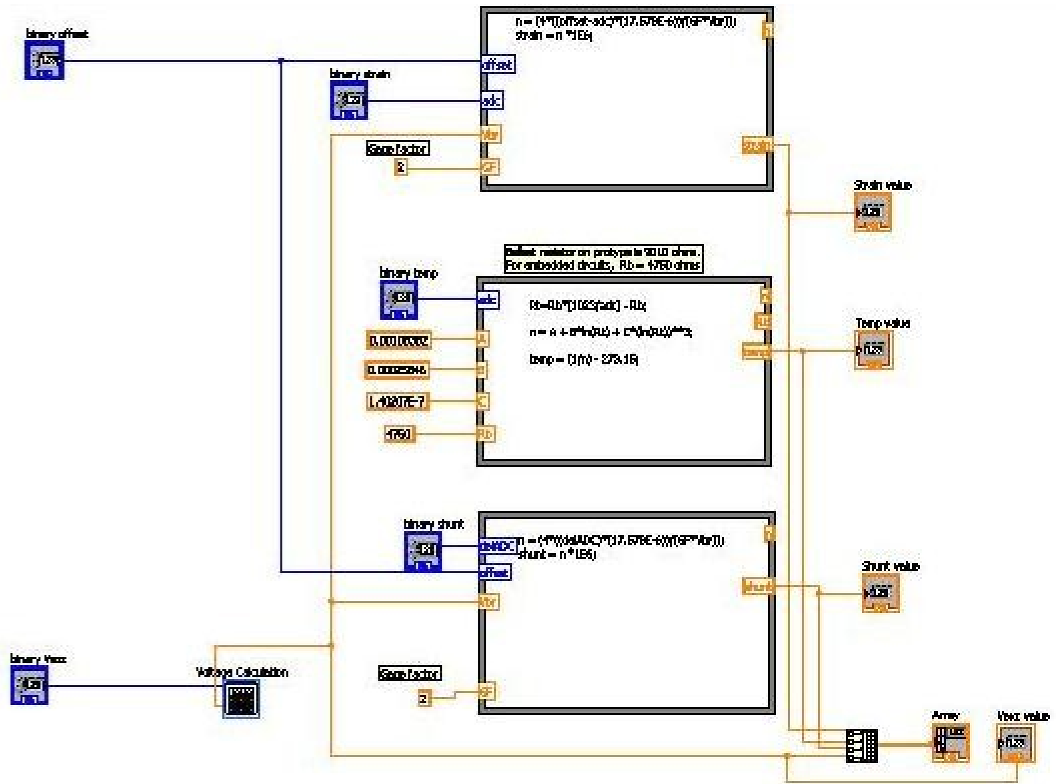



**Figure 3.10** Block diagram for converting binary data to engineering units.

The final segment of the block diagram is shown in Figure 3.11. This section of the block diagram is executed only when three conditions are true: the user interface is running, there are no errors on the $I^2C$ read operation, and the user selects the “log data” control on the front panel. When this condition exists, the output of the logic diagram on the bottom left of the figure will be true, and an array of data is formatted to three decimal places and written to a tab delimited text file. The logic circuit inside the loop at the lower right causes the program to halt if there is an $I^2C$ read error or the user

activates the stop button on the front panel. The icons outside the loop execute only when the program is terminated. The upper icon is a VI that closes the text file, and the icon on the bottom represents a VI that closes the $I^2C$ session and releases the reference for the USB-8451 module.

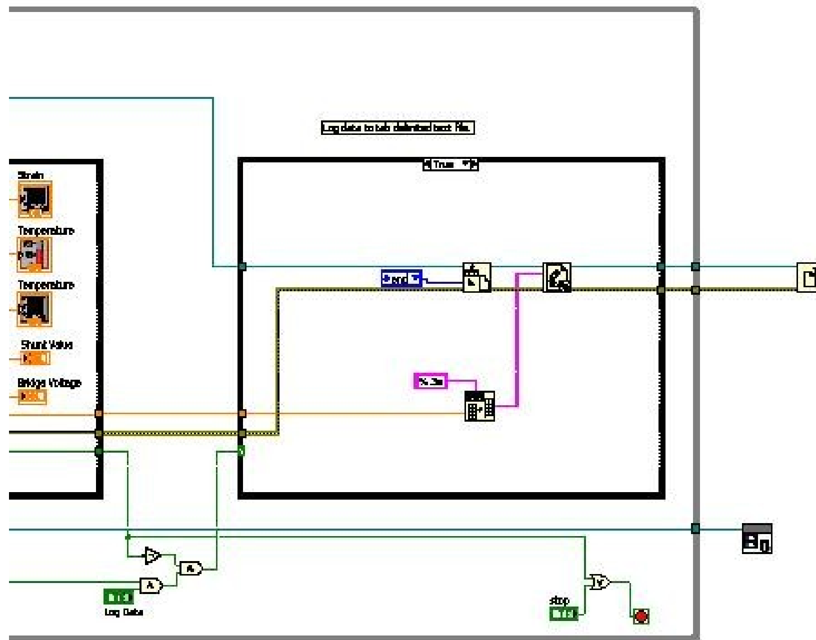



**Figure 3.11** Final segment of the user interface block diagram.

# CHAPTER 4

# FABRICATION OF TEST ARTICLES

As mentioned previously, Direct Write was used to create the conductive traces that form the electrical networks of the application circuit. The Direct Write process is the inverse of the conventional approach to printed circuit board fabrication. Circuit boards have traditionally been manufactured by masking off regions to form the desired conductive traces on a copper clad board, and subsequently employing an etching process to remove the unwanted copper. With Direct Write, a conductive material is deposited onto a dielectric substrate to form electrical networks. There are numerous techniques for accomplishing Direct Write, but this work employed only two methods: laser micromachining and micro-dispensing. We will briefly describe these techniques before discussing the fabrication of fiberglass panels.

## 4.1 Laser Micromachining

As is implied by the name of the process, laser micromachining involves the use of a laser to carve features into the surface of a substrate. This subtractive process is used to create tracks in the substrate in the shape of the desired circuit traces. The recessed areas are then filled with a conductive material to form the traces. This is illustrated on the upper left side of Figure 4.1, and the remainder of the figure shows the steps required to complete the process. A conductive material such as a nano-particle

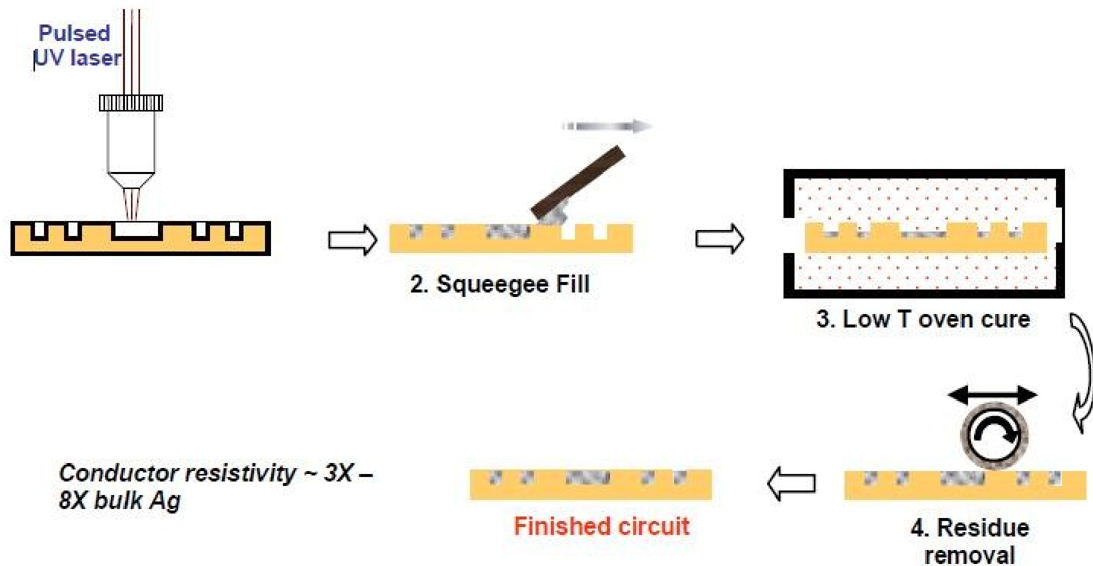



**Figure 4.1** Direct Write with the laser micromachining process[22].

silver paste is placed into the tracks, the substrate is placed into an oven to cure the conductive traces. Excess conductive material is removed from the surface of the substrate upon removal from the oven. Although not illustrated in the figure, a cover layer is typically applied over the top of the finished board to protect the traces. Final assembly of the circuit board is accomplished by bonding circuit components to their respective pads with conductive epoxy, and subsequently placing the entire assembly into an oven to cure the epoxy.

## 4.2 Direct Write with Micro-Dispensing

Unlike laser micro-machining which removes material from the surface of the substrate, micro-dispensing is an additive process, and it is the least complicated method for accomplishing materials deposition. A pump is used to drive a constant stream of silver-loaded paste through a needle or pen-like nozzle, and this dispensing mechanism is moved over the substrate to create the desired pattern. The print pattern is typically defined in a CAD file, and a computer is used to control the movement of the

print head in three dimensions[23]. Upon completion of the deposition process, the substrate is placed into an oven to evaporate the solvent and create conductive traces. Components are attached to the board with conductive epoxy in the same manner as was described for laser micro-machining. Figure 4.2 shows a micro-dispensing nozzle depositing a dot – dash pattern on a substrate.

**Figure 4.2** Deposition of material with a micro-dispensing nozzle.

## 4.3 Fabrication of Panels with Parasitic Electronics

Laser micromachining was used to fabricate four application circuits on a polyimide substrate, and, given the specialized nature of the equipment, Potomac Photonics of Lanham, MD was contracted to manufacture the circuits. The circuit layout used for the initial fabrication was determined after substantial collaboration, and the layout is shown in Appendix E.

Two features of the layout warrant some discussion. The first is that the board consists of a single layer and incorporates unconventional practices such as routing traces under passive components and around pins on some of the ICs. This was done

to reduce manufacturing complexity by eliminating via fabrication and the need to machine both sides of the substrate. However, the looping nature of the traces and lack of isolated analog and digital grounds present some concerns with respect to noise. Equation 3.1 is Faraday's Law that describes the relationship between a time varying magnetic field and the voltage induced in an open loop. [24, 25]. Thus, traces that

$$V = \frac{\partial}{\partial t} \iint \boldsymbol{B} \cdot d\boldsymbol{s} \qquad (3.1)$$

form loops could provide the mechanism for noise induction in the presence of a time varying magnetic field. Also, a common noise suppression practice in PCB layout is to use separate grounds for analog and digital circuits on the same board, and connect the two at only one point. [14, 15] This principle has been violated in the layout shown in Appendix E in an attempt to implement the circuit in a single layer.

Another feature of the design is the use of meandering Direct Write traces as variable resistors as a means of balancing the Wheatstone bridge. The circuit design called for the use of precision resistors to minimize bridge offset. However, Potomac Photonics was concerned that the traces would have enough resistivity cause an imbalance, and a symmetrical layout was not possible because of the need to provide terminals to connect the external strain gage. Also, difficulty was encountered in obtaining precision resisters in the 0201 package size (.024 x .012 in). Potomac Photonics suggested using one percent resistors and constructing trimmer resistors in the form of meandering parallel traces with numerous points exposed to allow the traces to be shorted together with conductive epoxy. The locations of the trimmer resistors is shown on the left side of the layout in Appendix E, and the shorting locations can be observed in Figure 4.3 as four matrices of silver points. The circuit shown in Figure 4.3 is the first of four that were manufactured by Potomac Photonics. It should be pointed

out that one shortcoming of this balancing scheme is that it can desensitize the strain gage should the need arise to add resistance in series with the gage.

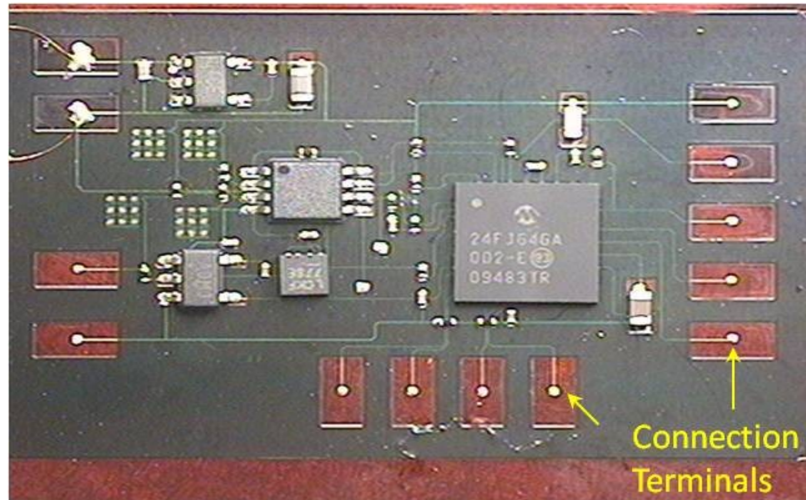



**Figure 4.3** Photograph of circuit fabricated by Potomac Photonics.

One concept as to how the circuit in Figure 4.3 might be applied to a vehicle structure is as a parasitic installation in the core area of a sandwich structure. Composite structures are typically constructed of carbon fiber or fiberglass laminates, and the circuit was consequently bonded to a G-11 Garolite test coupon with Vishay AE-10 Strain Gage Adhesive to simulate such an installation. Fabricating and testing a complete sandwich specimen was not possible because of cost constraints. The test coupon with attached electronics is shown in Figure 4.4.

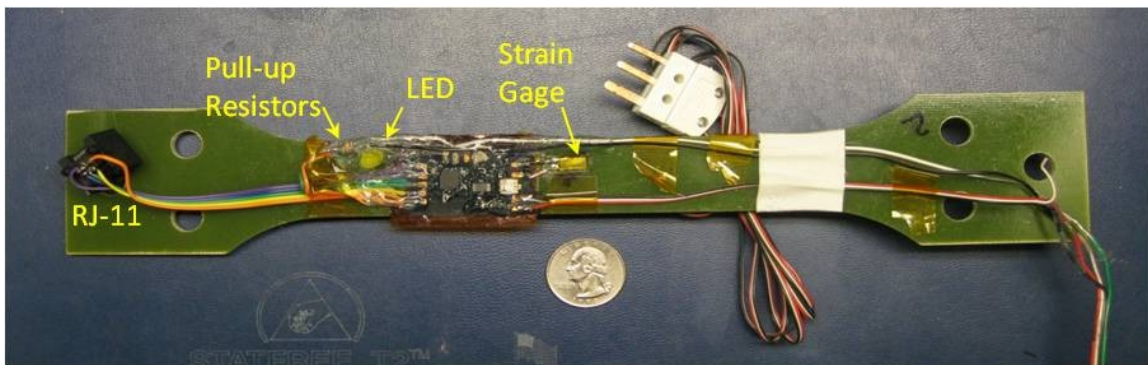



**Figure 4.4** Completed test coupon for Potomac Photonics circuit.

There were numerous external components required to enable testing, and these can also be seen in Figure 4.4. An RJ-11 connector is required for connection to the REAL ICE emulator/programmer, and an LED was connected to a digital I/O pin to enable a visual signal when the controller is reset. Two pull-up resistors were attached to the $I^2C$ lines along with wires for connection to the $I^2C$ module. Wire connections for a strain gage supporting laboratory data acquisition equipment and a power cable completed the external connections.

It is not possible to solder to the silver terminals on the Direct Write circuit (indicated in Figure 4.3), and this led to difficulty during assembly of the test coupons. A conductive epoxy from Chemtronics, CW2400, was used to attach all of the external components and wires. This epoxy requires a minimum drying time of four hours at room temperature, and limited tools in the hand assembly process limited the assembly procedure to two simultaneous bonds. At this rate, it took three standard business days to complete all of the external connections. Subsequent circuits from Potomac Photonics were supplied with copper tabs attached to all terminals to enable the use of solder. This expedited the assembly process somewhat; however, heat from the soldering iron occasionally caused the copper tabs to detach from the terminal. In these circumstances, the CW2400 epoxy was used to affect repairs; thus the four hour cure time was not completely eliminated from the process.

## 4.4 Fabrication of Panels with Integrated Electronics

SI2 Technologies was contracted to manufacture fiberglass panels containing integrated electronics using the micro-dispensing technique. The company has an in-house composites fabrication facility, thus SI2 was responsible for the entire fabrication process. The planned approach for production of the panels was to fabricate one specimen with all of the components mounted on the surface of the panel. This

specimen would allow for testing of the direct write process and functional tests of the completed circuit before proceeding with fabrication of the of the integrated panels. Taking into account the previously described research conducted by UCSD, SI2 agreed to construct the integrated panels using what UCSD describes as a “precision punch” technique.[5] This involves using one ply of the composite as a spacer between the layer containing the electronics and the remaining plies above the electronics layer. The spacer has material removed to accommodate the electronic components making up the circuit. The intent of this methodology is to minimize resin pockets surrounding the electronic components, and the concept is illustrated in Figure 4.5.

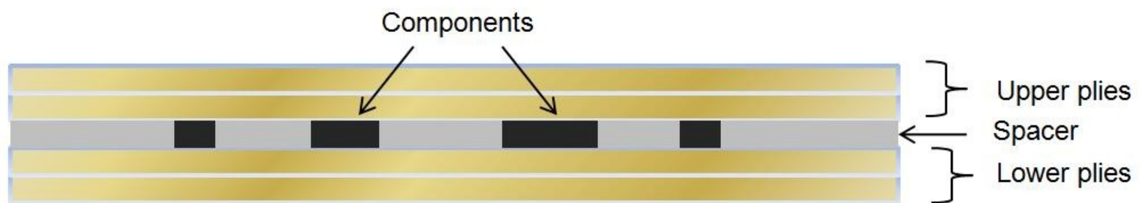



**Figure 4.5** Composite lay-up concept incorporating a spacer layer.

SI2 opted to use a subcontractor to produce the circuit layout, and the proposed layout is shown in Figure 4.6. In contrast to Potomac Photonics, SI2 used a two layer approach. The separation between the layers was accomplished by applying a dielectric material (Mylar) over the lower traces, and subsequently cutting holes in the dielectric to create vias.

A fiberglass panel with surface mounted components was fabricated by SI2, and this panel is shown in Figure 4.7. SI2 encountered a number of problems during manufacturing, and only the voltage regulator, microcontroller and thermistor circuit were implemented on the panel. There were numerous flaws in the Direct Write traces on this board, and the circuit was totally inoperable. Given the nature of the failures on what was the easiest specimen to fabricate, any further efforts by SI2 Technologies were

discontinued. The specific problems encountered will be discussed in detail in Chapter 6.

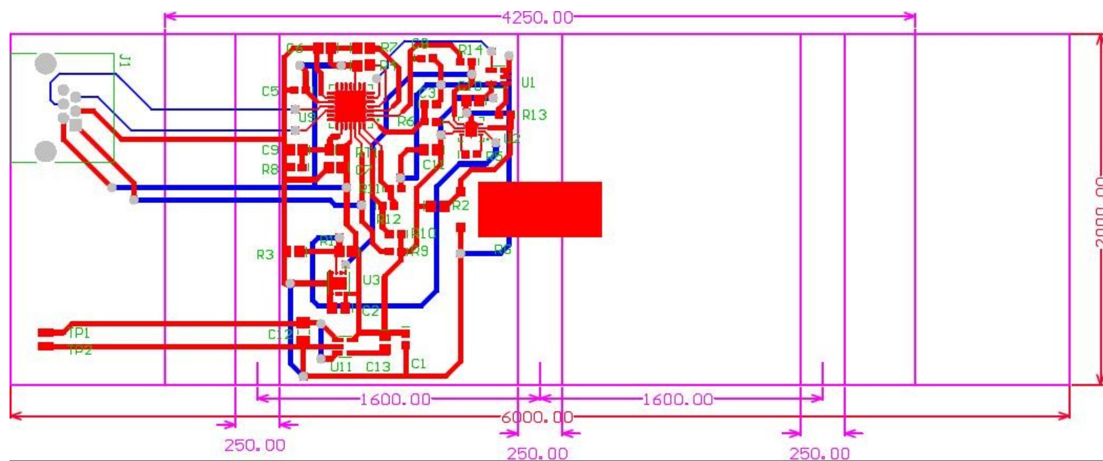



**Figure 4.6** Layout proposed for panel with integrated electronics.

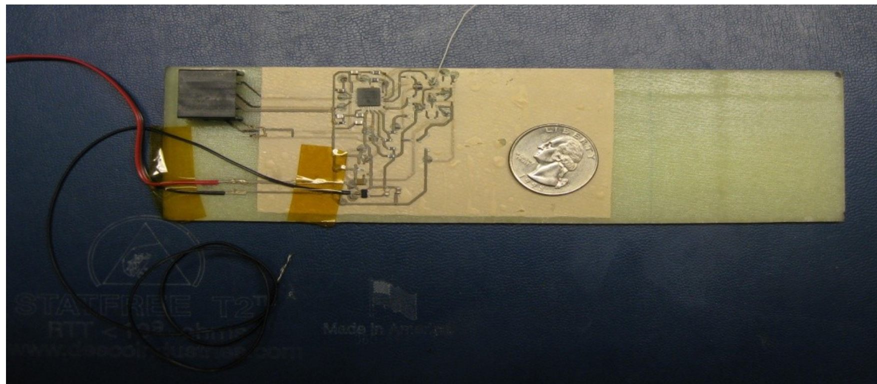


**Figure 4.7** Fiberglass panel produced by SI2 Technologies.

# CHAPTER 5

# TESTING

Testing of all specimens consisted of combined mechanical fatigue and system operational tests. The goal of testing was to investigate the durability of the electronic system by observing system operation while a cyclic load was applied to the panel. This section describes the physical laboratory set up and the test regimen that was applied to each specimen.

## 5.1 Physical Set-Up

The components comprising the laboratory set-up included a load frame with an attached load cell, a three point bending test fixture, an MTS hydraulic load control system, an Omega OM2-163 Bridgesensor signal conditioning card, and a computer to support the user interface for the MTS system. Figure 5.1 is a block diagram of the set-up, and a photograph of the apparatus is shown in Figure 5.2.

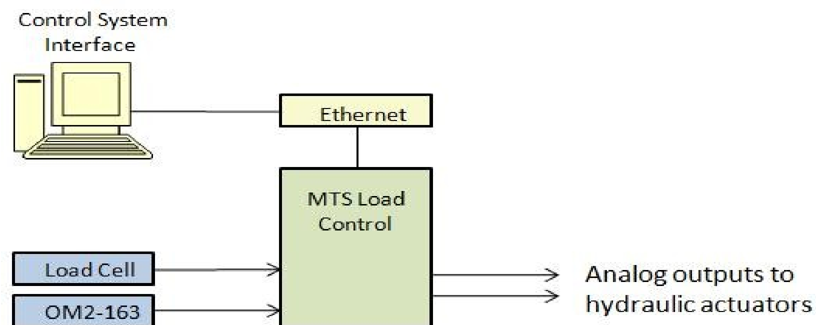



**Figure 5.1** Block diagram of test set-up.

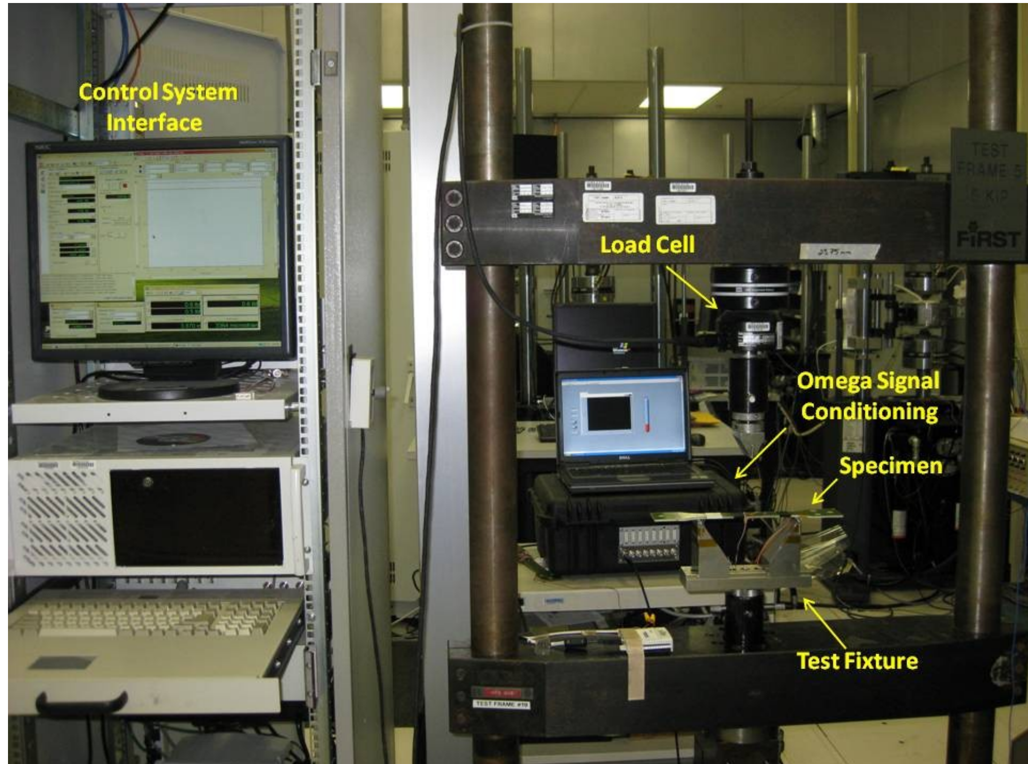



**Figure 5.2** Photograph of laboratory physical set-up.

A strain gage was attached to each test article directly underneath the electronic circuit under test. The strain gage was connected to the Omega Bridgesensor card, and the output of this signal conditioning board was connected to an analog input on the MTS control system. Thus, in addition to allowing input of load control parameters, the control system interface could also be used to observe real-time indicators displaying the applied load and the strain experienced by the electronic system under test.

A close up view of a test article mounted in the fixture is shown in Figure 5.3. All specimens were mounted with the electronic system facing downward, and the position of the electronics is indicated in the figure. Also visible at the lower left is the $I^2C$ communications module; a USB cable connects this module to the data display computer in the background. A bench top Hewlett Packard power supply was used to supply power to the electronic system, and this is also visible in the background.

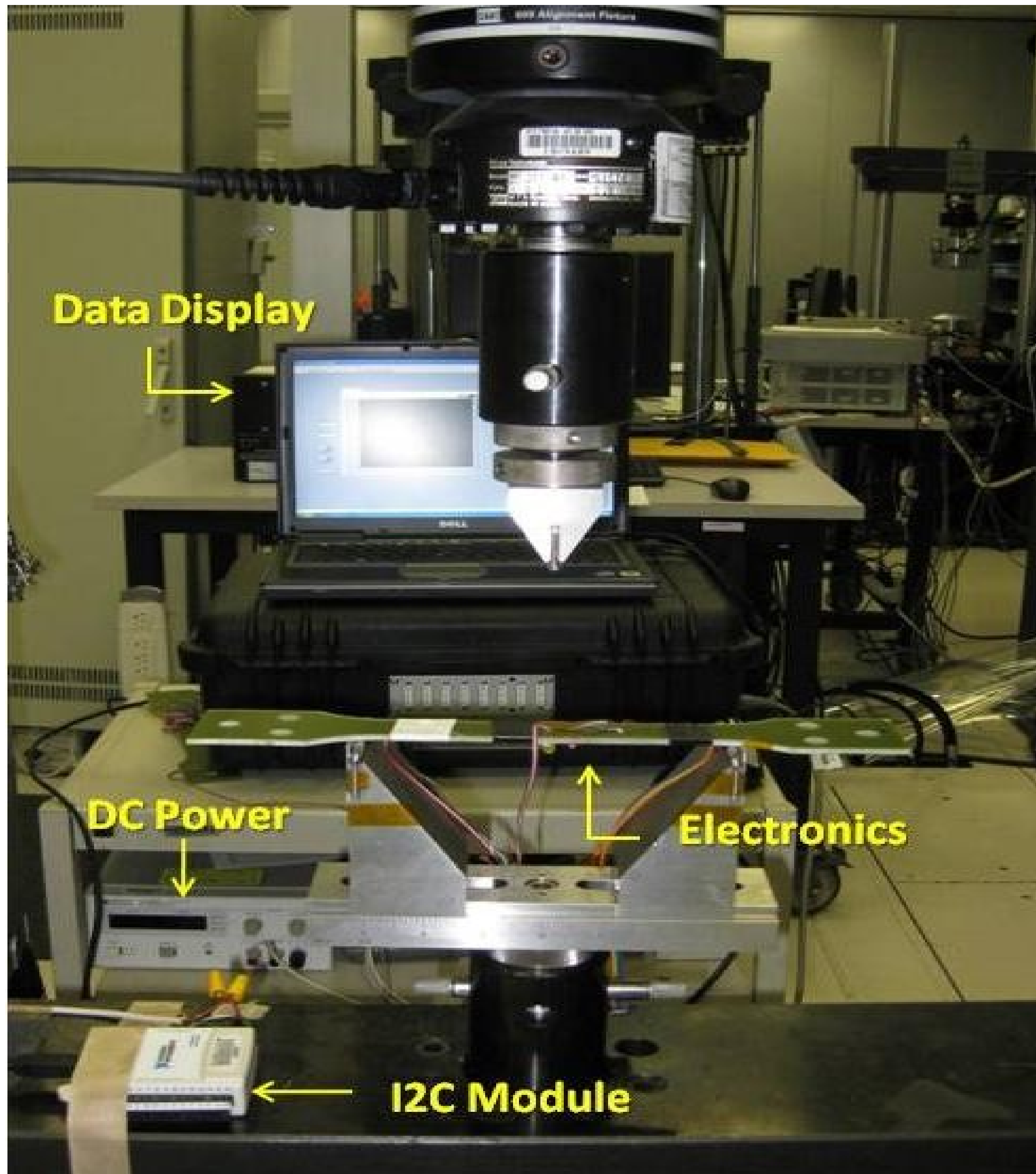



**Figure 5.3** Close up of test article in the loading fixture.

## 5.2 Test Procedure

Mechanical testing consisted of cyclic fatigue via the three point bending procedure as outlined in the ASTM D7264 standard. Experiments with a blank test coupon were used to determine a starting load condition. These experiments showed that a load of 4 lbs applied at the center of the coupon produced a strain of 360 – 390 microstrain in the region where the circuits were installed on the test articles. It was also

determined that a load increase of 2 lb caused a corresponding increase in strain of180 microstrain. This information was used to develop a test sequence in which the strain level was incremented at regular intervals throughout the test. Table 5.1 shows the sequence used for mechanical tests.

**Table 5.1** Test sequence used for cyclic fatigue experiment.

| Load (lb) | Number of Cycles | Strain (microstrain) |
|---|---|---|
| 4 | 2500 | 350 |
| 6 | 2500 | 530 |
| 8 | 2500 | 710 |
| 10 | 2500 | 890 |
| 12 | 2500 | 1070 |
| 14 | 2500 | 1250 |
| 16 | 2500 | 1430 |
| 18 | 2500 | 1610 |
| 20 | 2500 | 1790 |
| 22 | 2500 | 1970 |
| 24 | 2500 | 2150 |
| 26 | 2500 | 2330 |
| 28 | 2500 | 2510 |

The procedure itself was straight forward. After conducting bench testing to confirm that the test circuit was operational, the test article was placed into the three point bend fixture as illustrated in Figure 5.3. One final system check was performed, and then the MTS system was engaged to begin fatigue testing, The LabVIEW GUI was monitored during cycling to detect anomalous circuit performance due to the applied load. The test continued per Table 5.1 unless an error condition was observed in which case the test was halted, the number of cycles applied when the error occurred was recorded, and the cause of the error was investigated.

# CHAPTER 6

# RESULTS

Although the initial plan for this research was to investigate both an embedded and conformal mount approach for achieving structural integration, significant problems were encountered by SI2 Technologies in producing the Direct Write circuit traces. Thus, only the conformal mount circuits were available for mechanical tests. This section describes the problems encountered by SI2 Technologies and also covers the results from the mechanical testing conducted on the conformal mount circuits.

## 6.1 Micro-Pen Direct Write

As stated in chapter 4, the first circuit produced by SI2 Technologies was to be entirely surface mount in order to investigate the two layer approach as well as to enable troubleshooting of the application circuit. However, SI2 encountered significant problems in creating the circuit traces. The initial problem involved distortion of the land patterns for the ICs. SI2 deposited nano-particle silver paste onto an uncured piece of pre-impregnated fiber panel to create the electrical networks, and then applied heat to the panel to cure the silver paste. Although the heat cured the traces, it also caused a dimensional change in the panel, which resulted in a distortion of the land patterns for the ICs.

Ultimately, SI2 decided to deposit the silver paste onto a cured fiberglass panel in an effort to eliminate the distortion problem. However, serious problems were

encountered with the Direct Write traces in the form of cracking and spalling which caused opens in the conductive traces, and numerous short circuits occurred near vias and closely spaced traces.

Although these flaws were clearly visible under a microscope, SI2 delivered the test panel as a functional device. After completing laboratory tests that confirmed that the circuit was completely inoperable, SI2 was contacted to determine the cause of the defects. SI2 had no definitive explanation for the cracking and spalling, but speculated that the flaws may have been caused by an improper cure cycle; SI2 also stated that they had not observed any short circuits on the board [27]. At this point, the effort to fabricate the panels containing the embedded electronics was terminated.

The photograph on the left side of Figure 6.1 shows an example of some of the flaws present on the board produced by SI2. An open circuit in the Direct Write trace can be seen immediately to the right of the chip resistor on the left side of the photograph. To better illustrate the nature of this defect, an optical profilometer was used to map the surface of the board in the area of the flaw, and the results can be seen on the right side of Figure 6.1. The instrument had two optical pens available for measurements; one capable of measuring up to a height of 3.5 mm, and another with a maximum measurement height of 400 μm. The upper plot in Figure 6.1 was created with the 3.5 mm optical pen, and clearly shows a portion of the trace is missing. The lower plot was created using the 400 μm pen. The chip resistor is not visible in this plot because it is outside the measurement range of the pen; however, the defect in the trace is even more evident.

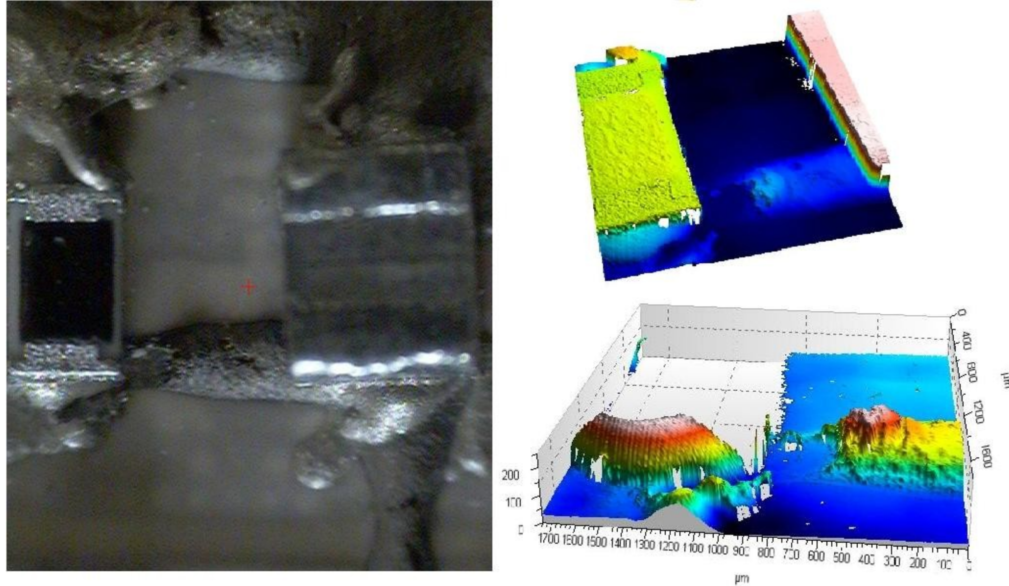


**Figure 6.1** Manufacturing flaw on Direct Write circuit trace.

Cracks in circuit traces were the most common flaw observed on the board, and the photo on the left in Figure 6.2 shows a significant crack in a trace. Also evident in both photos in the figure are cracks in the Mylar dielectric that insulates the traces on the upper layer from those on the lower layer. The cracks in the Mylar may support SI2's claim that the defects were caused by excessive heat application during the curing process.

Another possible cause of the defects is a difference in the coefficient of thermal expansion (CTE) between the underlying fiberglass composite panel and the Mylar dielectric. The data sheet for the fiberglass used to fabricate the panel shows the material has a CTE of 71 ppm / ˚C [28]. The CTE for the polyester film is 17 ppm / ˚ C [29]. Comparing these values, it can be seen that the CTE of the fiberglass is four times that of the dielectric film. It is reasonable to expect tears in the polyester film if the substrate expanded significantly more during the cure cycle than the film. Furthermore, as the film separated during this expansion, the overlying Direct Write traces would in turn be subject to damage.

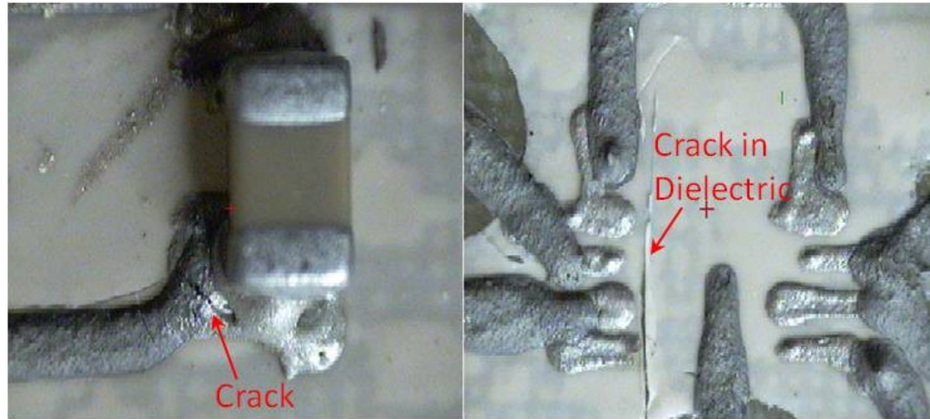



**Figure 6.2** Cracks in Direct Write traces and Mylar dielectric.

Although this explanation is somewhat hypothetical, the optical profilometer surface scan shown in Figure 6.3 supports the concept. The scan shows a circuit trace as it enters the pad for a passive component. A large crack is visible at the junction, and the crack can be seen to extend down into the dielectric in the foreground. It appears as though the Mylar cracked and caused the overlying trace material to separate along the length of the crack.

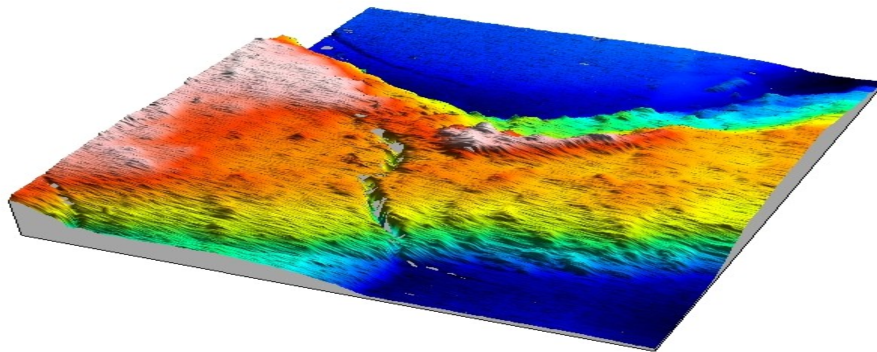


**Figure 6.3** Profilometer scan showing a crack in a circuit trace.

## 6.2 Laser Micromachining

There were four test circuits fabricated by Potomac Photonics using the laser Direct Write process. All of the circuits were functional, but minor problems were encountered in component bonding and in the balancing of the strain gage circuit. The following sections discuss results of electrical and mechanical testing for the four circuits.

### 6.2.1 Electrical Considerations

Measurements on the first board showed that it was not possible to use the meander line trim method to balance the strain gage bridge as described in chapter 4. The nano-particle trace material was more conductive than expected, and the voltage drops across the trimmer resistors was not enough to compensate for the imbalance caused by the 1% tolerance bridge resistors.

The voltage difference across the bridge was 38 mV, but the maximum adjustment that could be made by shorting the direct write trimming resistors was 17.4 mV. Thus, the bridge was still 20.6 mV out of balance and well outside the design goal of +/- 9 mV about the mid-supply point as discussed in chapter 2. As a result, a design change was made in the remaining three circuits to eliminate the direct write trim resistors. Instead, a potentiometer was added to the circuit as a means of balancing the bridge, as illustrated by the schematic in Figure 6.4 [30]. Although the modification shown in the figure initially allowed the bridge to be balanced, problems were eventually encountered with this scheme as well. The problem observed was that a rotation of the potentiometer did not result in any voltage change at the output of the instrumentation amplifier. The potentiometer was rotated frequently during bridge circuit checks both at the time of circuit assembly and during system tests prior to the start of mechanical

testing. It is believed that these manipulations led to a mechanical failure of the potentiometer.

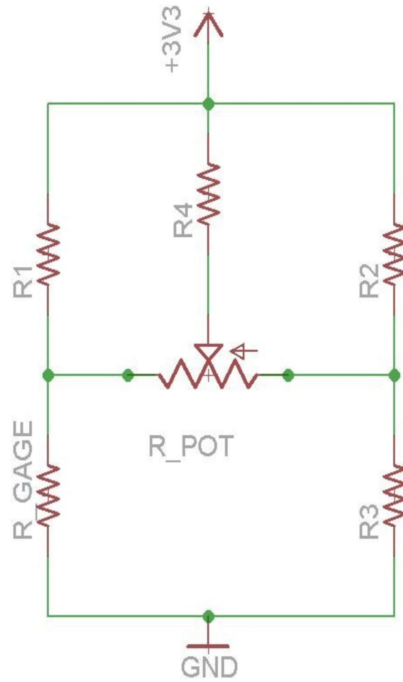



**Figure 6.4** Potentiometer added to balance the bridge circuit.

Power consumption was also investigated because of potential impact on a vehicle power system as well as limited ability to dissipate heat when the circuitry is integrated with a structure. During system testing, the current consumed by the circuit was monitored and found to be 9.8 mA. From the schematic in Figure 2.1, the main power draw can be seen as due to the bridge circuit, the series resistors that set up the ADC reference voltage, and the microcontroller. The remaining integrated circuits are micro-power devices that draw less than 100 μA combined, and the thermistor circuit will draw 317 μA at room temperature. Thus, the current expected is the sum of the currents for the bridge circuit (3 mA), the ADC reference resistors (1.2 mA), and the PIC24 running at 4 MHz (5 mA) [12], plus 400 μA for the micro-power devices and thermistor

circuit. The measured current of 9.8 mA is thus in close agreement with the calculated value of 9.6 mA.

With a current of nearly 10 mA and a supply voltage of 3V, the power consumed by the circuit is 30 mW. The heat generated by this amount of power is not expected to present a problem for dissipation; however, in an application with potentially 1,000 sensor nodes such as the notional one considered here, the vehicle power system would need to supply 30 W. This additional power requirement may add undesired weight to the vehicle. It should be noted, however, that nearly half of the current drawn by the application circuit is drawn by the resistive components in sensor circuits, which, as indicated in the introduction, would likely not be used in a practical application.

Noise was another concern with the approach of fabricating the circuit on a single layer using unconventional routing techniques. This was investigated by observing the main power bus on the direct write circuit with an oscilloscope and comparing it to the noise observed on the power bus of the breadboard circuit. Figure 6.5 shows the noise observed on the main power bus of the conventional breadboard circuit while Figure 6.6 shows a measurement of the same signal on a direct write circuit. Comparing the two plots, an 85 mV signal swing is observed on the direct write board while the breadboard had only a 7 mV noise envelop. Additionally, the large signal swings on the direct write circuit are periodic in nature, and occur 400 μs apart. This is the sample interval used to establish the 2.5kHz sampling rate. Thus, every time the microcontroller samples the analog signals, noise is injected on the system power bus of the direct write circuit. Since this does not occur in the breadboard circuit that was constructed with proper grounding procedures, it is evident that the routing of the direct write traces resulted in a noise issue.

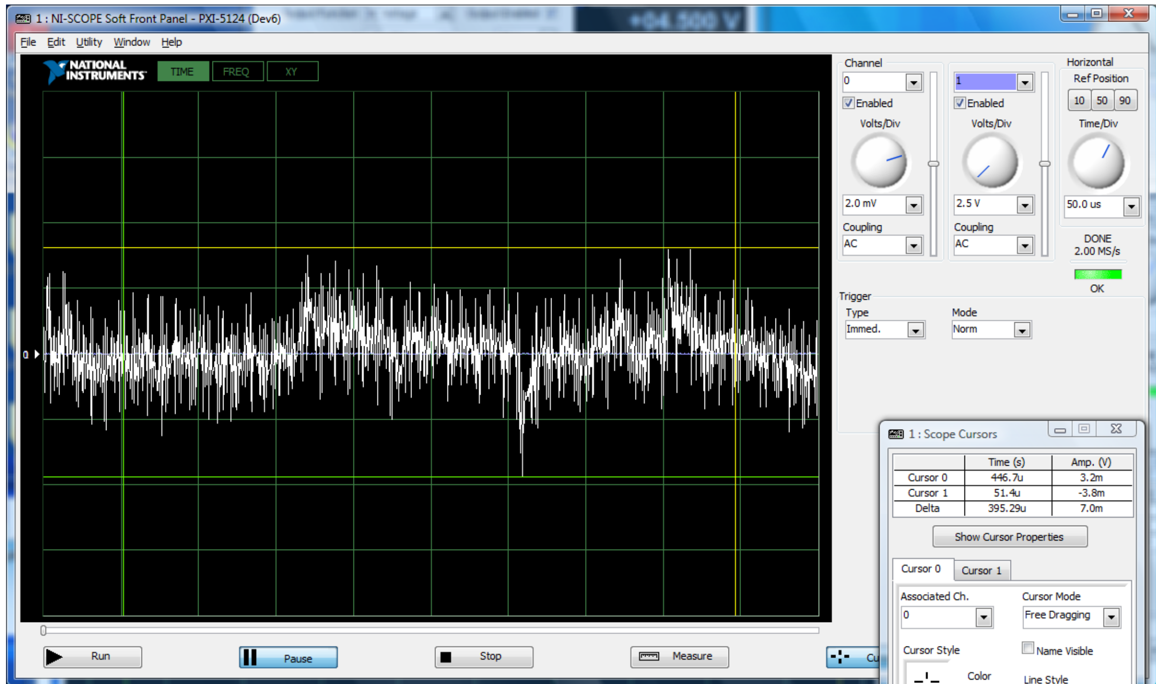



**Figure 6.5** Noise signal observed on the conventional breadboard.

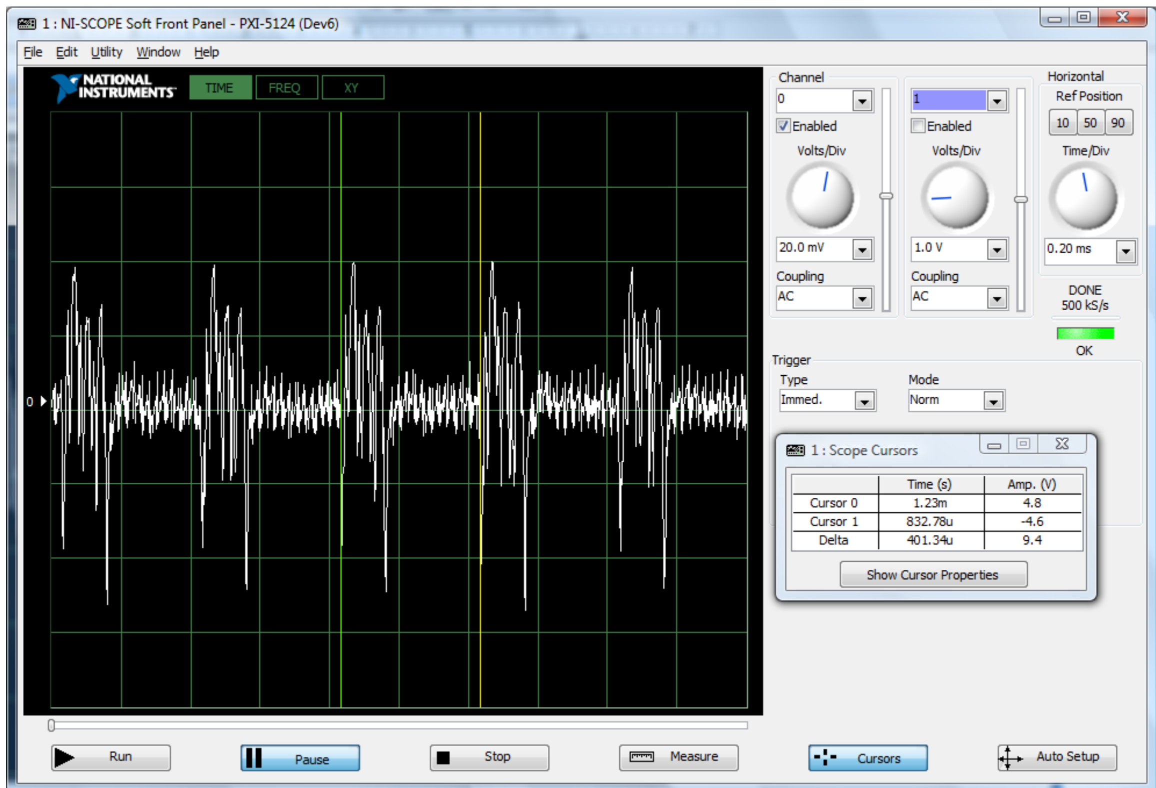



**Figure 6.6** Noise signal observed in a circuit fabricated with direct write.

### 6.2.2 Results of Mechanical Tests

Mechanical tests showed that the direct write circuits are rather durable despite no additional measures being taken to secure components to the board beyond the use of conductive epoxy to bond the component terminals to the pads on the board. Different issues were encountered with each test specimen, so the following sections will discuss each specimen individually. Table 6.1 presents a summary of the results obtained through mechanical testing.

**Table 6.1** Summary of test results.

| Specimen | Cycles Applied | Summary |
|---|---|---|
| 1 | 21,020 | Loose component caused an open circuit |
| 2 | 40,000 | Tested to 2500 με. Minor malfunctions observed. Fully operational upon completion of tests |
| 3 | 2,700 | Marginal power connection on microcontroller |
| 4 | 0 | Specimen destroyed during test set-up |

#### *6.2.2.1 Specimen 1*

During bench testing of the first specimen, a marginal connection was found on one of two zero ohm resistor that Potomac added to the ground trace in order to route the $I^2C$ traces to the terminals at the edge of the substrate. These resistors can be seen just below the lower left corner of the PIC24 in the layout shown in Appendix E. Circuit operation was restored by adding a small amount of conductive epoxy to strengthen this connection.

Table 6.1 shows the load applied to the specimen and the resulting strain applied to the test article. The circuit functioned normally measuring temperature and communicating with the GUI up through the 1,100 microstrain level. At 1,300 microstrain, occasional errors were observed in the temperature measurement in the form of sudden spikes from 24 C to 56 C. When the strain was increased to 1,500 microstrain, the processor rebooted one time during the applied 2,500 load cycles.

Finally, with a strain of 1,680 microstrain, a loss of communications occurred, and the circuit could not be re-started.

The specimen was removed from the fixture and taken to an electronics test bench for analysis. The cause of the failure was determined to be an open circuit at the zero ohm resistor connection that was repaired prior to testing. It was noted that circuit operation could be restored if downward pressure was applied to this node. Thus there was no component damage, the processor merely lost power due to the open circuit in the ground trace.

**Table 6.2** Applied load and number of cycles for specimen 1.

| *Load (lbf)* | Cycles Applied | Strain (με) |
|---|---|---|
| 4 | 4,500 | 360 |
| 6 | 2,500 | 560 |
| 8 | 2,500 | 760 |
| 10 | 2,500 | 960 |
| 12 | 2,500 | 1,100 |
| 14 | 2,500 | 1,300 |
| 16 | 2,500 | 1,500 |
| 18 | 1,520 | 1,680 |
| | 21,020 | |

#### *6.2.2.2 Specimen 2*

After observing the problem with the zero ohm resistors in the ground trace on the first specimen, Potomac was asked to remove these components and replace them with a bridge of conductive epoxy for all remaining circuits. This modification was made, and the second specimen was subjected to the mechanical testing procedure. Table 6.3 shows the results of the initial 14,675 load cycles applied to the specimen. No abnormal operation was observed until near the end of the 14 lb load level. There was an $I^2C$ communications failure near the end of the cycling for this condition, and the circuit appeared to be nonfunctional.

**Table 6.3** Results for the first 6 load levels on specimen 2.

| Load (lbf) | Applied Cycles | Strain (με) |
|---|---|---|
| 4 | 2,500 | 370 |
| 6 | 2,500 | 520 |
| 8 | 2,500 | 760 |
| 10 | 2,500 | 910 |
| 12 | 2,500 | 1,140 |
| 14 | 2,175 | 1,340 |
| | 14,675 | |

The specimen was taken back to an electronics laboratory for evaluation. The Microchip emulator was connected to the circuit, and it was determined that the processor was still functional. A break point was placed in the ISR for the $I^2C$ interrupt, and the GUI program was started in order to trigger an interrupt. The break point was activated, confirming operation of the processor's $I^2C$ peripheral. The emulator was used to conduct similar diagnostic procedures to confirm program operation, and no problems were found. The emulator was then removed, and the embedded system was operated and monitored in stand-alone mode for approximately 30 minutes with no anomalies observed. Based on this observation, a decision was made to resume mechanical testing to determine if the anomaly could be reproduced.

The mechanical tests were resumed starting several load levels below that where the communication failure occurred, and the number of applied cycles was reduced for these load conditions. Table 6.4 shows the results for the test. No other performance issues were observed until the 24 lb load level was reached. After 2,400 cycles at this level, the system supply voltage dropped to 2.5 V and a brown out reset occurred. However, the circuit returned to normal operation after the processor rebooted. The test continued until the 28 lb load level was reached. At this point in the testing process, it

was decided to add an additional 2,575 cycles in order to reach 40,000 total cycles with the last 5,000 cycles exposing the electronic circuit to 2,500 microstrain.

Upon completion of the cyclic testing, specimen 2 was still operational and testing was suspended in order to preserve the test article for further evaluation in the future.

**Table 6.4** Results for the tests on specimen 2.

| Load (lb) | Applied Cycles | Strain (με) |
|---|---|---|
| 8 | 750 | 740 |
| 10 | 1,500 | 940 |
| 12 | 1,500 | 1,150 |
| 14 | 1,500 | 1,340 |
| 16 | 2,500 | 1,540 |
| 18 | 2,500 | 1,720 |
| 20 | 2,500 | 1,900 |
| 22 | 2,500 | 2,000 |
| 24 | 2,500 | 2,180 |
| 26 | 2,500 | 2,370 |
| 28 | 2,500 | 2,550 |
| 28 | 2,575 | 2,550 |

#### *6.2.2.3 Specimen 3*

During bench testing of the third circuit fabricated by Potomac Photonics, a problem was encountered that was similar to the one experienced with the first circuit The circuit stopped functioning while using the PIC24 emulator to verify system operation.  After a visual inspection revealed no loose wires on the emulator connection terminals, downward pressure was applied to the PIC24 processor.  This action restored circuit functionality, and no further malfunctions were observed during bench tests.

Specimen 3 was then subject to the same cyclic fatigue testing as the first two test articles. After 1,600 cycles at the 4 lb load level, the microcontroller began to continuously attempt to reboot. The test was paused, the load was removed from the specimen, and the power was removed from the electronic circuit under test. With the test coupon resting in the test fixture, the power was restored to the circuit. Normal circuit operation was observed under static conditions for approximately 15 minutes, at which point the determination was made to resume the fatigue test under the same 4 lb load condition. Two additional resets were noted before the testing at the first load level was concluded at 2,500 cycles. Upon increasing the applied load to 6 lb, there was an immediate recurrence of the previously observed rebooting problem. The test article was again returned to the no-load, static condition, and, as was observed previously, the circuit functioned normally. Any further testing was suspended at this point because the problem appeared to be an intermittent connection caused by the applied mechanical load.

Potomac Photonics was consulted as to the potential cause of this problem, and the most likely cause was determined to be insufficient conductive epoxy on one of the power supply pins of the PIC24 processor. Potomac stated that they were concerned about creating a short by bridging two adjacent terminals with epoxy, and in this case they were likely too conservative with the epoxy. They also stated that this condition may be observable with a microscope if the board could be observed edge on. Figure 6.7 is a photograph of pins 22 -28 of the PIC24 taken in the manner suggested by Potomac Photonics.

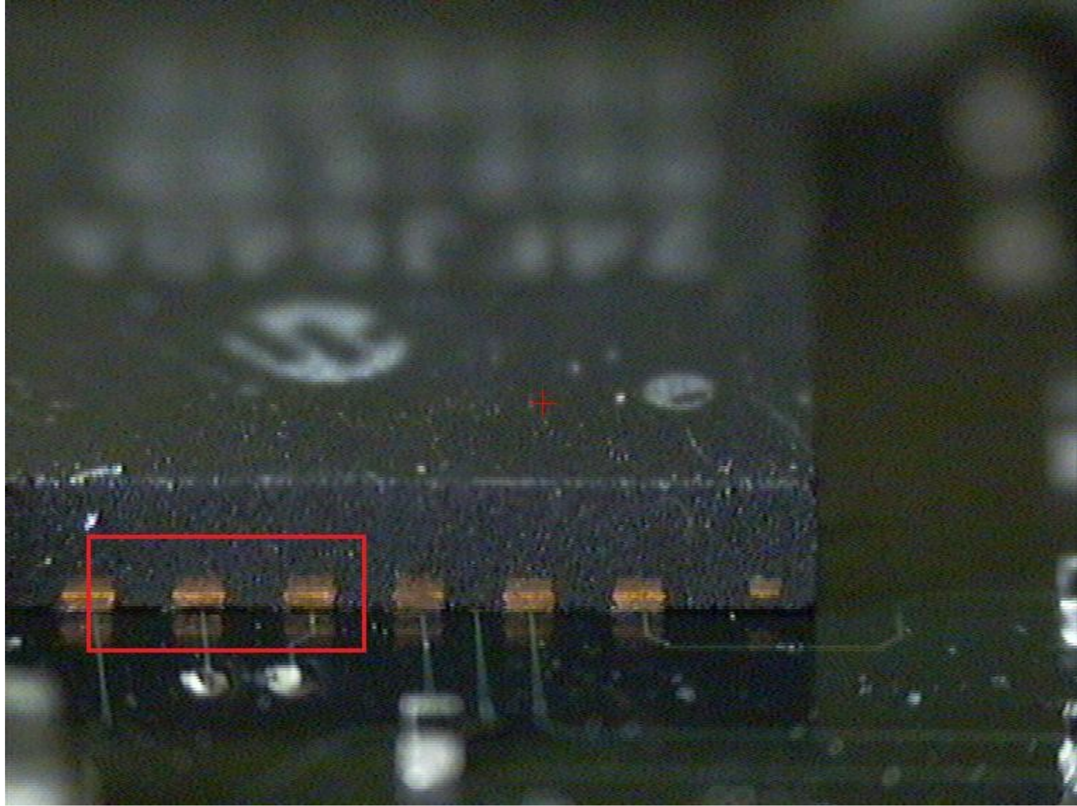


**Figure 6.7** Photograph of microcontroller pins.

The two pins on the right side of the box in the photograph are the power and ground connections. A gap between the IC and the substrate is clearly visible in the region indicated by the box, especially when compared with the three terminals immediately to the right. The intermittent power connection is thus seen to be due to an unevenness of polyimide substrate in the area under the power terminals combined with a fabrication practice of lightly applying the conductive epoxy to avoid short circuits.

#### *6.2.2.4 Specimen 4*

Since all of the problems observed during mechanical tests involved marginal connections, an attempt was made to address the problem on the fourth circuit. Potomac Photonics used a non-conductive epoxy to bond the electronic components to the board in addition to the conductive epoxy used on the electrical connections. The

intent of this approach was to increase the overall bond strength by adding additional bonding points while simultaneously reducing the mechanical load experienced by the bonds at the electrical connections.

Unfortunately, the fourth test article was destroyed during the set-up process for the mechanical test. Referring to the control system block diagram in Figure 5.1, the user interface is on a PC that communicates over Ethernet with the computer in the load control system. Between the end of tests for specimen 3 and the beginning of tests for specimen 4, an upgrade to the virus protection software was installed on the PC. Unbeknownst to laboratory personnel, this upgrade included a firewall feature that blocked communication between the user interface PC and the control system. As a result, the test operator experienced difficulty trying to set up the test rig for the fatigue test. While trying to diagnose the problem, the operator manually engaged the control system, which did not have the correct control parameters due to the now blocked communication path with the user interface. As a result, the control system drove the lower portion of the test fixture shown in Figure5.3 into the fixed point on the top of the fixture. This caused the test article to be completely severed into two pieces thereby eliminating any possibility of conducting the planned fatigue test.

Upon inspection, the circuitry itself did not appear to have any damage, but some of the wires allowing external connections to power and the $I^2C$ bus were torn from the connection points. Once all of the connections were restored, power was applied to the circuit, and it appeared to be functioning normally. Further analysis with the emulator showed that the program was running as expected, and inspection of various registers turned up no evidence of malfunction.

Given this surprising result, the focus of the remaining laboratory work shifted to determining how much strain the circuit experienced during the fracture event. One final experiment was designed to investigate this point. A fiberglass coupon identical to the

one destroyed in the previous test was instrumented with four strain gages applied between one end of the coupon and its mid-point. The control system was then programmed to increase the load level in 5 lb increments between 0 and 60 lb in an attempt to deliberately fracture the coupon and record the strain level at the point just before fracture. The coupon ultimately withstood the full 60 lb load without fracturing and the strain recorded in the region where the electronics were installed on specimen 4 was measured at 8,500 $\mu\varepsilon$. At this point the test was terminated to prevent possible damage to the mechanical test fixture. Hence, although the exact value of the strain experienced by the Direct Write circuit of specimen 4 during the fracture event remains unknown, it was determined through this experiment that the maximum value was in excess of 8.500 microstrain.

# CHAPTER 7

# INTERFACE TO THE 1553 AVIONICS BUS

Another factor that was considered during this work is communication of data between the structurally integrated electronics and flight control systems. Although it was not implemented in the hardware described previously, an effort was made to develop a conceptual design to interface the integrated electronics with a common data bus used on aircraft. For the purposes of this study, the MIL-STD-1553 data bus was selected as a typical bus with which an integrated sensing system would need to communicate. Before describing the interface design, a brief overview of the MIL-STD-1553 data bus will be given.

## 7.1 MIL-STD-1553 Overview

MIL-STD-1553 outlines the implementation of a 1 Mbps serial communications network for up to 32 nodes that is widely used as an avionics bus [31]. Although it is possible to implement networks consisting of multiple levels, single level topologies are the simplest and most widely employed, and this architecture is shown in Figure 7.1. One of the characteristics of the 1553 bus evident in the figure is the use of redundant busses. Data is transmitted on only one bus, and the second bus is only used in the event of a failure on the primary bus.

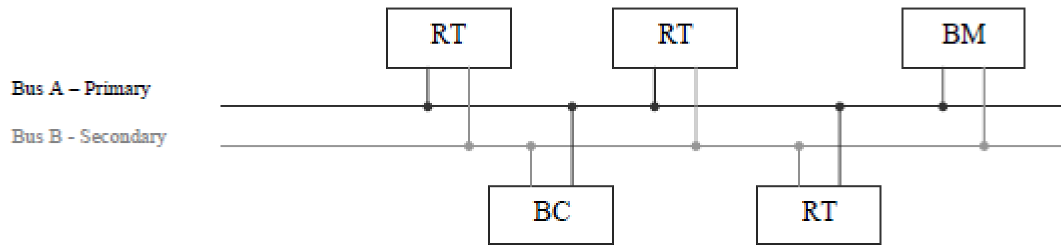



**Figure 7.1** Single level 1553 network.

As shown in the figure, there are three types of nodes possible on a network – a bus controller (BC), remote terminals (RT), and a bus monitor (BM). The bus controller manages all aspect of network operations by issuing commands to remote terminals to control both node actions (such as a restart) and traffic on the bus. Although a network can employ multiple BCs, only one can be operational at any time. [31] The remote terminals are interfaces to subsystems such as flight controls, navigation systems, and engine management. A bus monitor is device used to capture traffic on the bus for analysis, and may or may not be included on a given network.

Because a single cable makes up the network, only one node can transmit at any time while the remaining nodes receive. The network is therefore time division half duplex. All communication on the bus is initiated by the BC, and remote terminals only communicate with the BC or other RTs when directed by the BC. This is in contrast to other bus type networks, such as Ethernet, in which nodes may transmit at any time, and a bus arbitration scheme is employed if two nodes attempt to transmit at once. When the BC transmits a command, the receiving RT must respond with a status word as an acknowledgement.

Figure 7.2 shows the format of the command word. This word specifies the address of the RT to which the BC is sending the command in bits 4-8, and specifies if

**Figure 7.2** Format of MIL-STD-1553 command word [32].

the RT is to transmit or receive data with bit 9. Bits 10 – 14 contain the RT sub-address involved in the data transfer, and bits 15 – 19 specify the number of data words to be transmitted or received. A special case occurs if the sub-address field is 0 or 31 (0x1F). This condition indicates the command is a mode code, which is a bus management command. Under this condition, bits 15 – 19 specify the mode code type, and the value indicates the management function to be performed [32].

Figure 7.3 shows the format of the status word, with the purpose of each bit position indicated below each position. The status word contains the address of the transmitting RT, and a bit field that may not be entirely utilized in a given implementation. The more important bits are the Message Error, Service Request, and Terminal Flag bits [32].

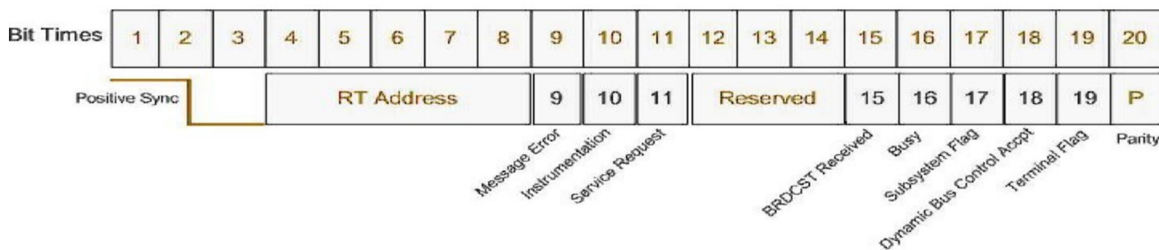



**Figure 7.3** Format of the MIL-STD-1553 status word [32].

The two words described above, along with a 16 bit data word, are used to construct all packets on the 1553 bus. The message types on the 1553 bus include BC broadcast, BC to RT, RT to BC, and RT to RT. For this discussion, only BC to RT and

RT to BC messages will be considered because it is assumed that the structurally integrated electronic system will have to supply data to a process on the computer acting as the BC. Figure 7.4 depicts this interaction. The BC first issues a command for a specified number of data words from the integrated electronics system. In response, the RT transmits a status word followed by the number of data words specified by the BC.

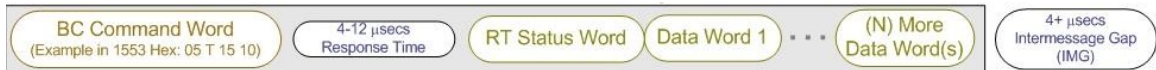



**Figure 7.4** Communication between BC and RT[32].

## 7.2 Hardware Considerations

The interface method proposed here is based upon the HI-6121 integrated circuit from Holt Industries, and the suggested approach is illustrated in Figure 7.5. The HI-6121 is a monolithic MIL-STD-1553 Remote Terminal that implements all functions of the MIL-STD-1553 protocol. One of the primary reasons for the selection of this device was the availability of a 4-wire Serial Peripheral Interface (SPI) port to interface with the host processor. The PIC24F microcontroller used in the application circuit is in a 28 pin package, and, a parallel interface, assuming it could be implemented with the unused I/O pins, would not represent the most efficient use of those pins. The SPI approach allows the HI-6121 interface to be accomplished with five I/O pins, and leaves pins available to expand the number of sensors monitored by the PIC24.

The most significant change to the application circuit would be the addition of a 50 MHz clock to support the HI-6121. A surface mount, 50 MHz silicon oscillator, such as the LTC6905-100 from Linear Technology, Inc. is one method of providing a clock source while adding only a single IC and bypass capacitor to the existing hardware. The 3.3V supply required by the HI-6121 would necessitate an additional voltage regulator,

but, as with the silicon oscillator, this could be accomplished with only two additional components.

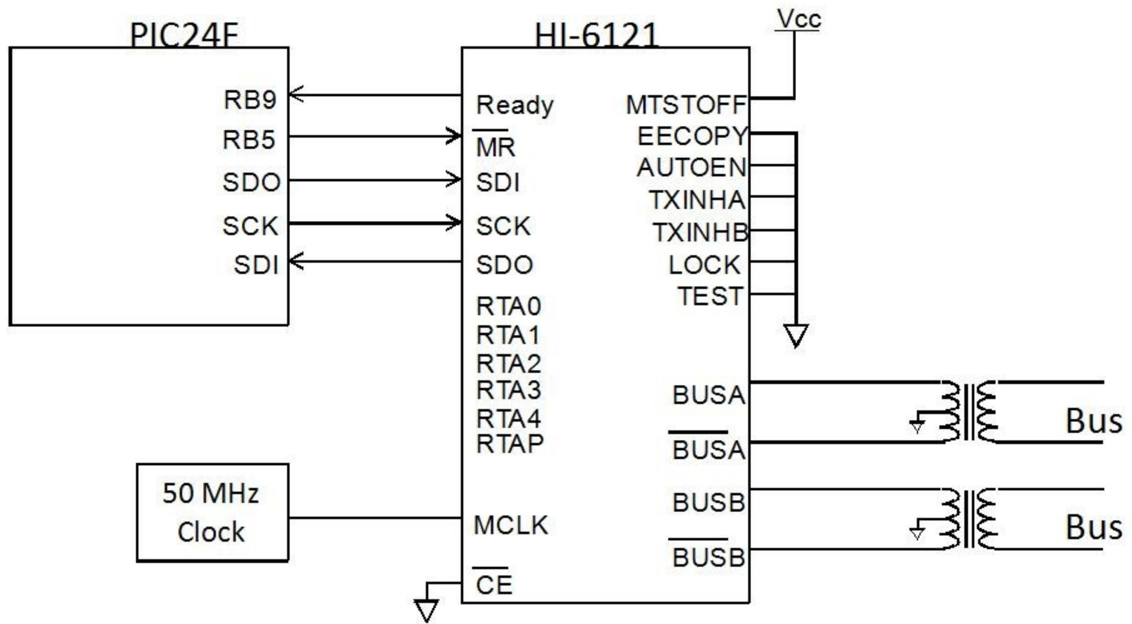



**Figure 7.5** Schematic for the Avionics Bus Interface.

Although the additional components to accomplish the bus interface do not add significant complexity to the hardware, a more significant concern is the power drawn by these devices. The clock IC draws 6mA while the HI-6121 draws an idle current of 4 to 10 mA and a 500 mA current draw in transmit mode. Thus, the current draw when the node is not transmitting will at least double in the best case. If a large number of nodes are to be incorporated into a vehicle, available vehicle power and heat dissipation could be issues.

## 7.3 Hardware Connections

The following is description of the pin connections required for the HI-6121. It should be noted that the HI-6121 offers numerous options to the user, and consequently is a complex device. The implementation described here is intended to be minimally

sufficient because a more complex design would require more knowledge of a specific system application.

The Ready pin on the HI-6121 is low anytime built-in tests are in process on the device. The host processor checks the status of this pin prior to accessing registers or RAM on the HI-6121. The host processor can initiate a hardware reset by driving the master reset pin low, or, alternatively, a software reset can be generated by setting a bit in one of the two configuration registers.

Communication between the host processor and the remote terminal is accomplished with the SPI protocol, and the processor acts as the as the master and provides the clock for data transfers. Three connections must be made between the devices to implement the SPI protocol.

The address of the remote terminal is established by pins RT4-RT0, and the RTAP pin is used to establish odd parity for the address on RT4-RT0. The status of the address pins is latched into an internal status register to establish the address of the remote terminal. The level of the LOCK pin is also latched into the status register and determines if the host processor can overwrite the address bits in the status register. In this case, it was decided to retain the ability to change the address in software, so the LOCK pin was grounded.

On the upper right side of the schematic, one pin is tied high while several others are connected to ground. The MTSTOFF pin is tied high to disable a test of RAM during a reset. The reset time is shortened by omitting this memory test.

The EECOPY and AUTOEN pins are used to initiate copying configuration data to and from external EEPROM. A choice was made to simplify the hardware design by configuring the HI-6121 with the host processor. Since no EEPROM is present, these pins were grounded to disable the function.

The TXINHA and TXINHB pins can be used to disable transmission on bus A and bus B. However, this can also be accomplished by setting the appropriate bits in one of the configuration registers, so the external pins were grounded. The pins and control bits are logically ORed, thus grounding of the pins does not eliminate the possibility of inhibiting bus transmission.

The TEST pin provides a means for the host processor to initiate a RAM self-test or loopback test. This feature was not implemented to conserve processor I/O pins, and the pin is thus connected to ground.

## 7.4 Firmware Considerations

Since it has been assumed that the RT will be sending a specified number of data words to the BC, the host processor needs to perform two major operations. The first is to initialize the internal registers and data structure on the HI-6121 to ensure proper operation of the RT. The second function is obviously to provide periodic data updates to a specific area of RAM on the RT to support communications with the BC.

The HI-6121 has 32 registers that are 16 bits wide, but not all of these require initialization. Some registers are reserved, some are read-only, and others may not be needed in a given application. For instance, one of the read-only registers is the Time Tag register. This is a counter that is used to generate a time stamp that is stored in memory along with data anytime RAM is updated. The Time Tag Utility Register can be used to load the Time Tag counter with a specific value, an action which may not be required for all applications. The four registers that need to be initialized in any application are Configuration Register 1, Configuration Register 2, Status Word Register, and the Descriptor Table Base Address Register. We briefly consider these registers.

Configurations Register 1 is the last register that would be written to during an initialization sequence because bit 8 in this register (STEX) is used to begin RT

operation.  Other bits that may be required are bits 9 – 11 (TTCK0 – TTCK2) which are used either to disable the Time Tag Counter or to select the clock source for the counter. Another bit to be aware of is bit 7, which can be used to initiate a software reset of the remote terminal.  Other functions controlled by this register are interrupts and bus disable.

Configuration Register 2 is used to load and reset the Time Tag counter, enable a 32-word data buffer for all receive commands, and establish time-out period for RT – RT receive commands.  Other options managed from this register include establishing terminal response to mode code commands received from the BC.

The 1553 Status Word Bits Register is used to establish appropriate bit levels in the status word sent by the RT in response to BC commands.  For example, if the application needed to set the Service Request bit in the Status Word, the processor would set the bit in this register, and the RT would transmit the Status Word upon receipt of the next BC command.

The Descriptor Table Base Address Register points to the beginning of the Descriptor Table.  This data structure is vital in terminal operations because it is used to set up the terminal's actions upon receipt of all commands from the BC.  Consequently, the effect of loading this register incorrectly or corrupting its contents during program execution could have a catastrophic effect on terminal operations.

The Descriptor Table itself warrants further discussion here because this structure must be initialized by the host processor and the data pointed to by this table also needs to be periodically updated.  Although this must be done for every valid command issued to the RT by the BC, we will illustrate only how the descriptor table would need to be initialized for the structurally integrated electronic system.  In order to do this, we first assume that the sub-address for the integrated electronic system is 2, and that it is desired to employ a double buffer scheme for data being sent to the BC.

The Descriptor Table is divided into four quadrants, with each quadrant dedicated to one of the four types of commands issued by the BC: receive mode commands, transmit mode commands, receive sub-address commands, and transmit sub-address commands. Each quadrant consists of 32 consecutive groups of four data words. Figure 7.6 illustrates this structure for the transmit sub-address quadrant, which is the quadrant that would be used for transmitting data from the structurally integrated electronic system to the BC. Since sub-address 2 is assumed to be used by the integrated system, block 2 needs to be initialized by the host processor. As indicated in the figure, each block of data words is comprised of a control word and 3 descriptor words. The control word is used to select a data buffering mode and any desired RT actions, such as interrupt generation, upon receipt of the sub-address. Thus, for the integrated electronics application, the processor would need to configure the control word to select double buffer mode. Once this mode is selected, the processor would need to load the address of the two data buffers into the first two descriptor words. The third descriptor word is related to responding to a BC broadcast message, which is not considered here.

Once the initial configuration is complete, the host processor would need to provide updates to the data buffers. This could be done after every tenth acquisition cycle with the current firmware design. The HI-6121 would handle all communications with the BC. When a transmit sub-address for the structurally integrated system is received, the HI-6121 would use the descriptor block for sub-address 2 to select the appropriate data buffer, and transmit the data to the BC.

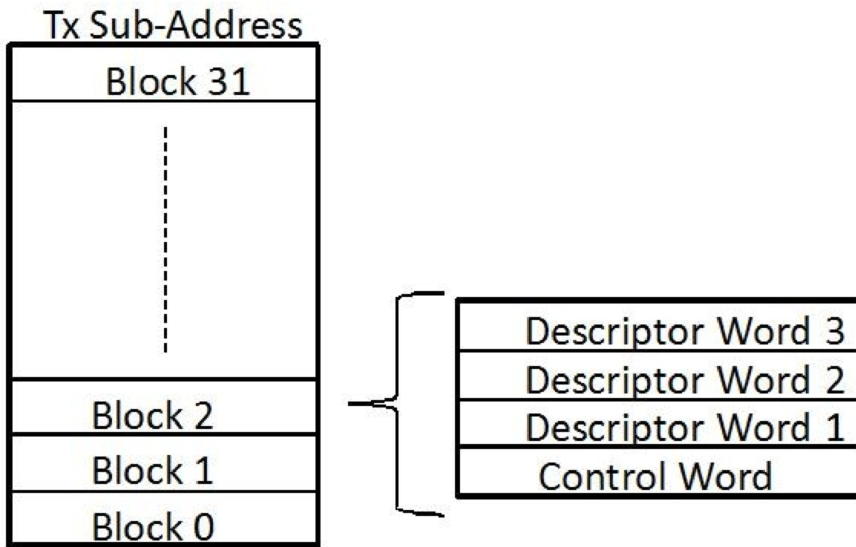



**Figure 7.6** Organization of the Descriptor Table.

# CHAPTER 8

# FUTURE WORK AND CONCLUSIONS

## 8.1 Future Work

The results outlined in chapter 6 show that the durability of a circuit board fabricated with Direct Write technology and commercial electronic components is surprisingly robust, and, as a consequence, the concept of a structurally integrated embedded system is feasible. However, further research is needed in several areas to address some of the problems encountered in this work.

One of the key areas needing refinement is circuit fabrication. Direct Write is a viable fabrication method, but the next step is to enable creation of vias between circuits on different composite plies or even between two sides of a polyimide substrate. This will enable structures containing integrated electronics to parallel the construction of conventional printed circuit boards; thereby taking advantage of noise mitigation practices used in the production of standard boards. The application circuit in this research used unconventional routing techniques and experienced undesirable noise levels as a result. Improving methods for fabricating Direct Write electrical networks is critical for both noise mitigation and enabling the implementation of more complex circuits.

Another aspect of the electronics fabrication process that needs further development is component attachment. All of the circuit malfunctions encountered in the

fatigue tests were attributed to circuit connections that opened when a mechanical load was applied to the test article. Thus, improved bonding techniques would enhance the ability of the circuit to remain operational while experiencing mechanical load. One bonding method that needs to be evaluated is the use of non-conductive epoxy to take some of the load off of the conductive epoxy on circuit connections. Also, further investigation of the intra-laminar mounting approach that was attempted in this research is warranted because encasing the circuitry in the structure itself may make connections between components and mounting pads more reliable.

Another important area for further work is a system demonstration that incorporates sensors representative of those that lend themselves to structural integration. Strain sensors on flexible substrates that could potentially be used as an intra-laminar inclusion are under development, and these devices provide the advantage of being fabricated as a full bridge [33]. Consequently, discrete bridge resistors and balancing potentiomters, such as those that created problems in this research, can be eliminated from the strain circuit. Figure 8.1 shows an example of these flexible strain sensors.

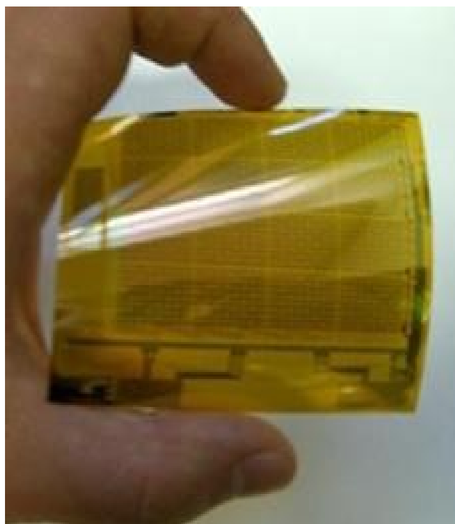


**Figure 8.1** Strain sensors fabricated on a flexible substrate.

## 8.2 Conclusion

In this thesis, an embedded system was integrated with a mechanical test panel and was proven to be capable of withstanding a significant strain level through laboratory testing. The key finding is that no circuit malfunctions were observed up to 1,100 microstrain, and, although malfunctions were encountered beyond this level, none of the test articles were significantly damaged in 60,000 load cycles. With the exception of one test article, all specimens were functional upon completion of mechanical tests, and the malfunction in the inoperable specimen was simply due to an open circuit. Also significant is the fact that one specimen was exposed to greater than 8,500 microstrain during a fracture event, and the embedded system sustained no permanent damage. The main point of failure found during testing was intermittent connections between electronic components and the circuit board.

# BIBLIOGRAPHY


1. Victor Giurgiutin (University of South Carolina), John H. Barnes, Dustin Thomas, (Air Force Research Laboratory), “Multifunctinal Vehicle Structural Health Monitoring Opportunities With Piezoelectric Wafer Active Sensors”, 2002, DTIC web site, URL http://handle.dtic.mil/100.2/ADA516846, Accessed September 3, 2009.

2. Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, Martin turon, “Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks”, Proceedings of the 5th International Workshop on Structural Health Monitoring, 2007, ACM Press, pp. 254 – 263.

3. Md. Nazmul Alam, Rashad H. Bhuiyan, Roger A. Dougal, Mohammod Ali, “Concrete Moisture Content Measurement Using Interdigital Near-Field Sensors”, IEEE Sensors Journal, Vol. 10, No. 7, July 2010

4. David J. Warkentin, Edward F. Crawley, Stephen D. Senturia, “Embedded Electronics for Intelligent Structures”, Master of Science Thesis, Massachuseetes institute of Technology, 1991.

5. Kristin Schaff, Patrick Rye, Sia Nemat-Nasser, “Optimization studies of self-sensing composites”, Proceedings of SPIE, vol. 6529 (2), pp. 65292X.1-65292X.5, March 2007.

6. Yi Huang, Sia Nemet-Nasser, “Structural Integrity of Composite Laminates with Embedded Microsensors”, Sensor Systems and Networks: Phenomina, Technology, and Applications for NDE and Health Monitoring 2007, edited by Kara j. Peters, Proc. Of SPIE Vol. 6530, 65300W, (2007)0277-786X/07.

7. A. Piqué, B. Pratap, S. A. Matthews, B. J. Karns, R. C. Auyeung, *M. Kasser*, M. Ollinger, H. Kim, S. Lakeou, and C. B. Arnold, “Laser-direct-write of embedded electronic components and circuits,” Photon Processing in Microelectronics and Photonics IV, eds. J. Fieret, P. R. Herman, T. Okada, C. B. Arnold, et al., International Society for Optical Engineering (SPIE), **5713**, 223-230 (2005)

8. Walt Kester, “Section 4-2 Bridge Circuits”, Op Amp Applications, Edited by Walter G. Jung, Analog Devices, 2002.

9. Karl Hoffman, “An Introduction to Measurement using Strain Gages”, Hottinger Baldwin Messtechnik, 1989.

10. IPC Systems LTD, “Basic Strain Measurements”, 2007, Sensorland.com, http://www.sensorland.com, Accessed September 12, 2009)

11. Walt Kester, James Bryant, Walt Jung “Section 4-5 Temperature Sensors”, Op Amp Applications, Edited by Walter G. Jung, Analog Devices, 2002.

12. PIC24FJ64G004 Familt Data Sheet,Microchip Technology, Inc., 2008, http://www.microchip.com, Accessed August 24, 2009.

13. Lucio Di Jasio, “Programming 16-bit Microcontrollers in C - Learning to Fly the PIC24”, Elsevier, Inc., 2007.

14. Ron Macini, “Op Amps for Everyone, Design Reference”, Second Edition, Newnes, 2003.

15. Michael Ashton, Rick Downs, Tim Green, Bill Klein, Lijoy Philipose, Robert Watson, “Precision Data Acquisition Seminar”,Texas instruments, 2004.

16. Clifford A. Shaffer, “Data Structures and Algorithm Analysis”, Prentice Hall, 2001.

17. Raj Kamal, “Embedded Systems: Archtiecture, Programming, and Design”, McGraw Hill, 2008.

18. Gary W. Johnson, Richard Jennings, “LabVIEW Graphical Programming”, Third Edition, McGraw-Hill, 2001.

19. Jeffrey Travis, Jim Kring, “ Labview for Everyone: Graphical Programming Made Easy and Fun”, Prentice Hall, 2007.

20. Richard Bitter, Taqi Mohiuddin, Matt Nowrocki, “LabVIEW: Advanced Programming Techniques”, CRC Press, 2001.

21. Peter A. Blume, “The LabVIEW Style Book”, Prentice Hall, 2007.

22. C. Paul Christensen, “Energy Storage, Electrical Distribution, and Packaging for Wireless Sensor Networks”, SBIR Final Report, National Science Foundation, 2010.

23. Alberto Pique, Douglas B. Chrisey, “Direct Write technologies for Rapid Prototyping: Applications to Sensors, Electronics, and Passive Coatings”, Academic Press, 2001.

24. Constantine A. Balanis, “Advanced Engineering Electromagnetics”, John Wiley & Sons, 1989.

25. Ramon Pallas-Areny, John G. Webster, “Sensors and Signal Conditioning”, Second Edition, John Wiley & Sons, 2001.

26. ASTM Standard D 7264/D 7264M -07, 2006 (2007), “Standard Test Method for Flexural Properties of Polymer Matrix Composite materials” ASTM International, West Conshohocken, PA, 2007, DOI: 10.1520/D7264_D7264M-07, www.astm.org.

27. SI2 Technologies, email to SelectTech Services (AFRL Procurement Contractor), October 2010.

28. Technical Data Sheet, “BT250E-1 Resin System” , Tencate Advanced Composites USA, Inc., http://www.tencate.com, accessed May 21 2010

29. Grafix Plastics, Mylar Film and Sheet Properties, http://www.grafixplastics.com/mylar_prop.asp, Accessed March 2011

30. James I Craig, “Electrical Resistance Strain Gage Circuits”, Georgia Tech school of Aerospace Engineering, http://www.soliton.ae.gatech.edu/people/jcraig/classes/ae3145/lab2/strain-gages.pdf, Accessed October 5, 2010.

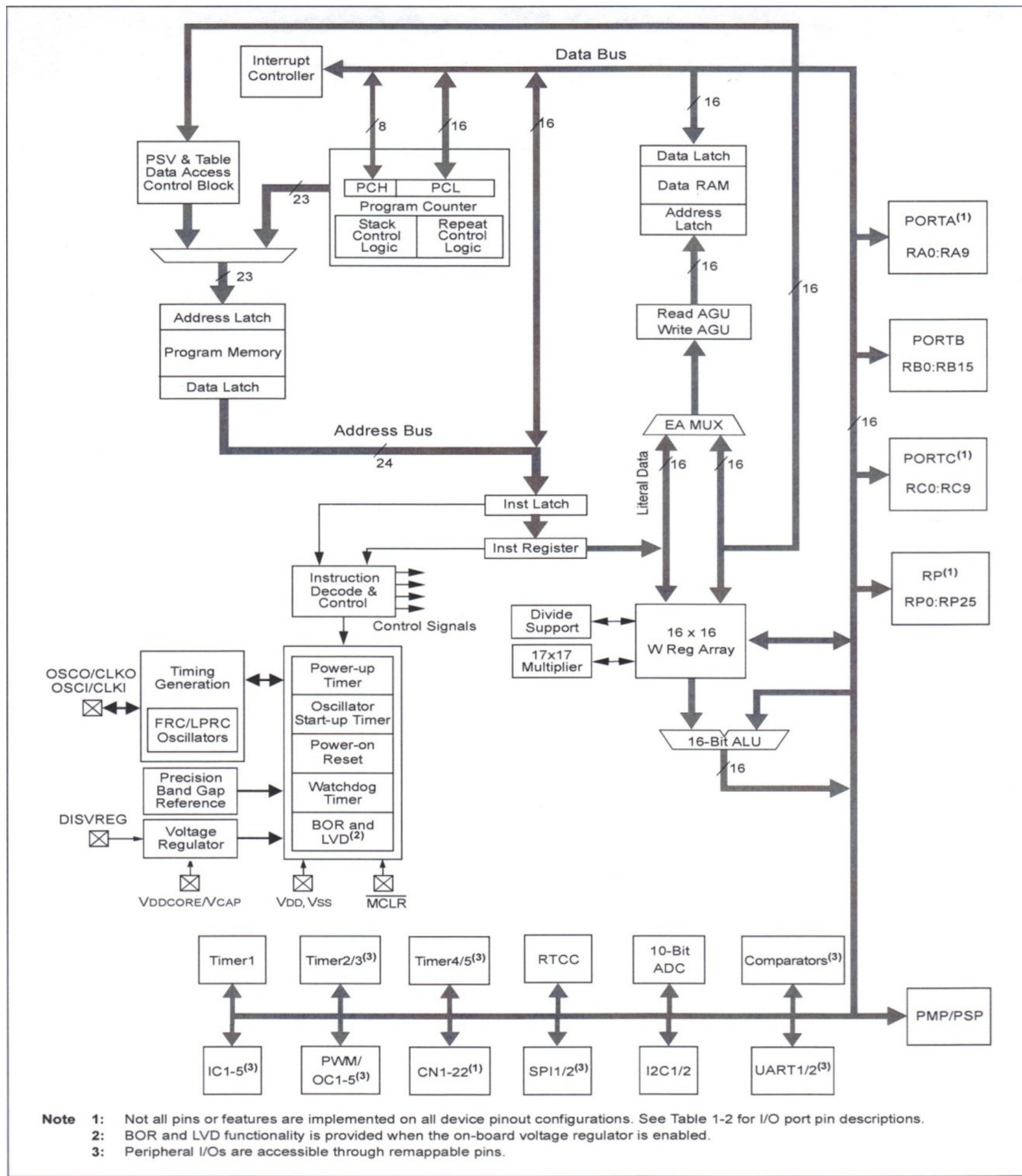
31. Avionics Interface Technologies, “MIL-STD-1553 Protocol Tutorial”, 2010, avittech.com, http://www.aviftech.com ; accessed February 16, 2010.

32. Alta Data Technologies, LLC, “MIL-STD-1553 Tutorial and Reference”, 2007, altadt.com, http://www.altadt.com; accessed February 19, 2010.

33. Thomas Jackson, “a-Si:H flexible substrate, Strain Sensor”, Penn State University, http://jerg.ee.psu.edu/research/pdf/StrainSensor_1208.pdf; accessed August 23,2011.

# APPENDICES

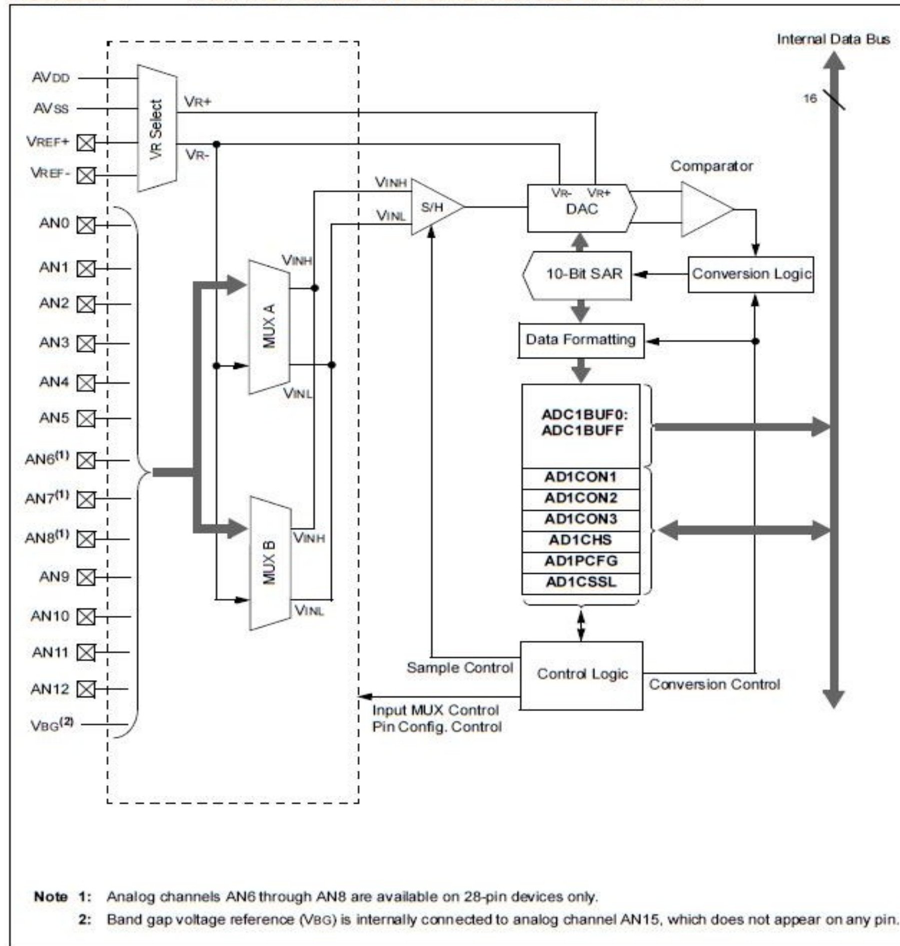
## Appendix A: Block Diagram of Microcontroller



## Appendix B: Block Diagram of Microcontroller ADC Peripheral



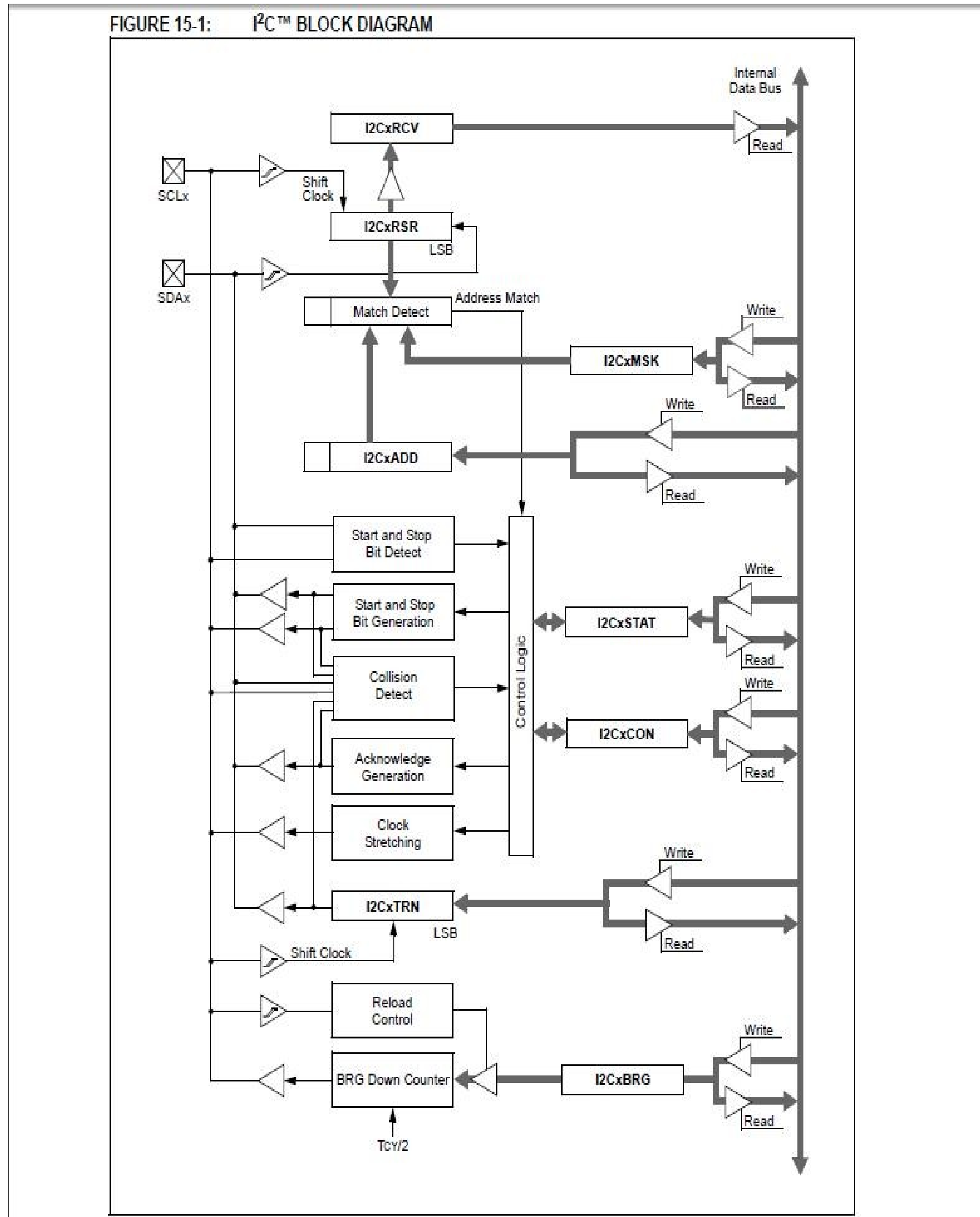
FIGURE 20-1: 10-BIT HIGH-SPEED A/D CONVERTER BLOCK DIAGRAM



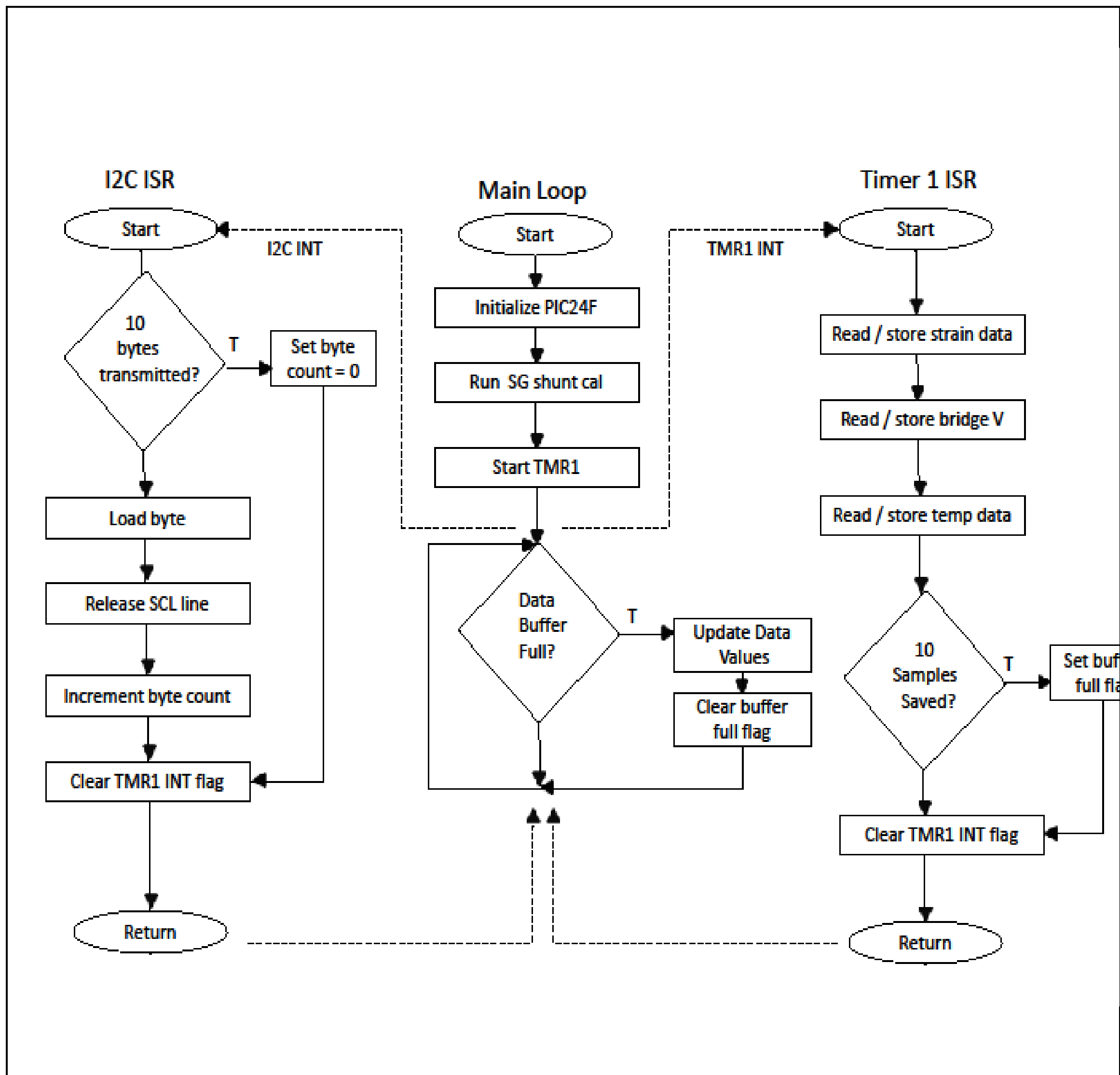

**Note 1:** Analog channels AN6 through AN8 are available on 28-pin devices only.
**2:** Band gap voltage reference (VBG) is internally connected to analog channel AN15, which does not appear on any pin.


© 2008 Microchip Technology Inc.

## Appendix C: Block Diagram of Microcontroller I2C Peripheral



## Appendix D: Firmware Structure



## Appendix E: Circuit Layout