University of Portland

Donald P. Shiley School of Engineering
5000 N. Willamette Blvd.
Portland, OR 97203-5798

Phone 503 943 7314
Fax 503 943 7316

Final Report

Project Goonies:
Spinal Fusion Strain Gage

Team Members:

Trever Hickey (Spring Team Lead)

Kelcey Lajoie (Fall Team Lead)

Ashley Martin

Caroline Nakaye

Faculty Advisors:

Dr. Robert Albright

Customer:

Dr. Deborah Schenberger

Revision History

Rev.	Date	Author	Reason for Changes
0.1	4 April 2012	K. Lajoie, C. Nakaye, A. Martin	Initial draft. Contains analog technical outcomes, digital technical outcomes, and analog process outcomes.
0.2	5 April 2012	T. Hickey	Added digital process outcomes and conclusion.
0.3	10 April 2012	A. Martin	Added edits to Digital Technical Outcomes
0.4	12 April 2012	T. Hickey	Edited Digital Outcomes & Conclusion. Still need to edit Figures, Table of Contents, References, and talk about data results in conclusion.
0.5	12 April 2012	C. Nakaye	Added edits to analog technical outcomes
0.6	13 April 2012	K. Lajoie	Added edits to Introductions, Milestones, and analog process outcomes
0.7	13 April 2012	A. Martin	Formatting and addition of data collection to conclusion
0.8	20 April 2012	T. Hickey, A. Martin	Added information about LabJack.
0.9	21 April 2012	C. Nakaye	Added Acknowledgements, Appendixes, and edited spinal fusion strain gage block diagram
1.0	25 April 2012	A. Martin	Final edits

Table of Contents

List of Figures

List of Tables

Acknowledgements

Team Goonies would like to thank the following people for their support and help to make the Spinal Fusion Strain Gage project a success. Without them the success of this project wouldn't be possible.

Dr. Robert Albright

Dr. Albright is the Faculty Advisor for Team Goonies. He helped the team stay on schedule and as well as providing assistance with project development, research and documentation revisions.

Dr. Joseph Hoffbeck

Dr. Hoffbeck provided Team Goonies with MPLAB and MATLAB debugging assistance and also proposed the use of the LabJack.

Dr. Wayne Lu

Dr. Lu provided PIC Microcontroller software that Team Goonies used in programming the PIC24. He also provided necessary debugging help for the PIC24.

Craig Henry

Craig Henry, the Electronics Technician at the Shiley School of Engineering provided material and assistance essential for construction of the project.

Dr. Timothy Doughty

Dr. Doughty provided Lab-View equipment to test with the analog circuitry as the team was waiting for the LabJack to arrive.

Patrick Hickey

Patrick Hickey, Trever's father, built the steel metal casing where the entire circuitry is enclosed.

Introduction

Spinal fusion is a surgical technique that joins two vertebrae and is usually done to relieve pain. When cartilage between the vertebrae wears away, the bone on bone contact causes discomfort and pain. The solution is to fuse the two vertebrae into one bone to eliminate movement between the two vertebrae. The spinal fusion surgery adds external hardware, which supplies structure, and bone material, which jumpstarts the body to begin forming bone tissue around the hardware, as seen in Figure 1. It takes between 6 and 18 weeks for bone to form and fuse to the existing vertebrae. Currently, the only way to measure the bone formation is using X-rays. Unfortunately, bone material can only be seen on an X-ray once it has mineralized, which does not occur right away. Also, it is hard to see the newly formed bone because the hardware is in the way. While the bone is forming the patient must be in a cast that extends from their arm pits to hips for many months, and in some cases up to a year.

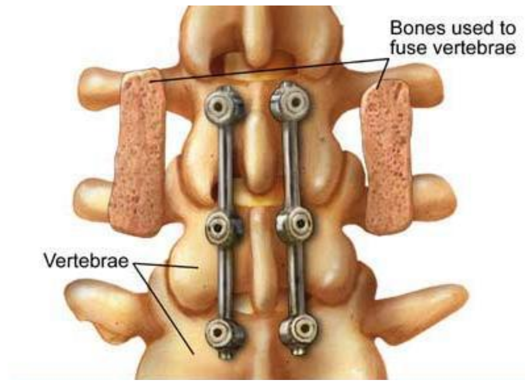


Figure 1. Illustration of Spinal Fusion.

Dr. Deborah Schenberger, a Mechanical Engineering professor at the University of Portland, has been researching an alternative solution to X-rays in measuring bone formation. She has developed a strain gage that can be implanted during the spinal fusion surgery to measures the strain in the metal hardware. Initially, the strain in the hardware will be high, because the hardware is carrying the load of the vertebrae. As bone tissue forms around the structure, the strain in the hardware will decrease. Eventually, once the bone has fully healed, the strain value will plateau. The strain gage will allow doctors to be sure that enough bone has formed to support the spine before allowing the patient's cast to be removed.

Use of the strain gage is still in the research phase. Dr. Schenberger has tested her design using deceased sheep spines and manually measuring the strain using a multimeter. Since a strain gage is essentially a variable resistor, she used a Wheatstone bridge and an amplifier, and measured the changes in voltage as she incrementally added bone cement, which simulates bone fusion. Since her research will still take several more years, and her current method of measurement is tedious, she asked our electrical engineering team to devise a way to more quickly measure and graph the output of the strain gage.

This year's project's goal was to connect to the strain gage as part of a Wheatstone bridge, filter, amplify the signal, convert the analog signal to digital, and graph the data in a computer program with MATLAB. Many challenges were faced along the way as this design was implemented.

This document outlines exactly what the final project includes in the Technical Outcomes section. It also highlights the differences between the design and the final product in the Process Outcomes section. The challenges that were faced and how they were overcome is also included in that section.

Technical Outcomes

The Spinal Fusion Strain Gage project consists of several components that all work together to collect data from the strain gages, process it and graphs it on a computer in form of strain against time. Figure 2 outlines how the different components interact.

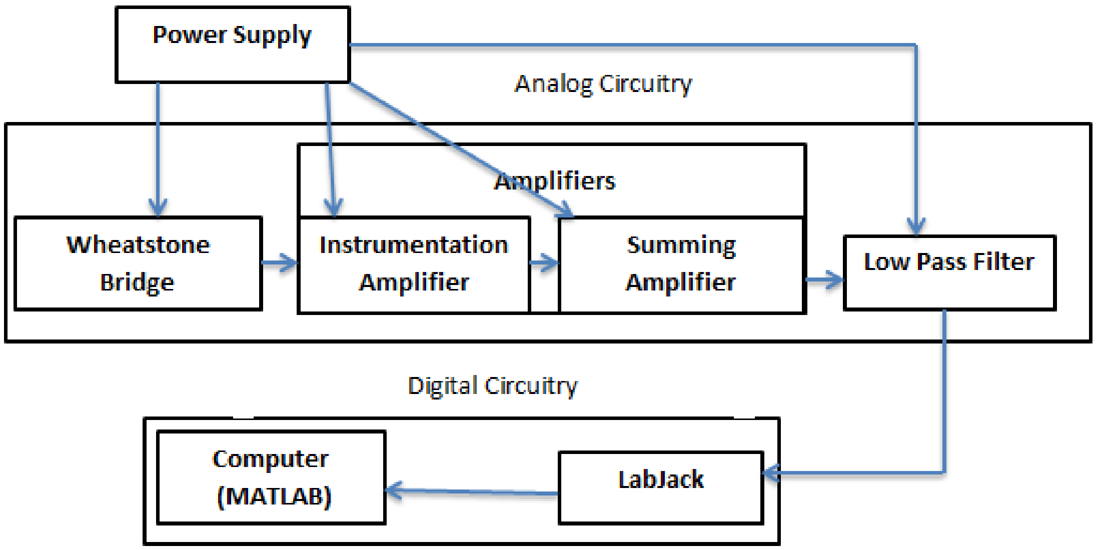


Figure 2. Spinal Fusion Strain Gage System Block Diagram.

The following section will discuss each of the components shown in Figure 2 above in greater detail. It will also describe how each component interacts with the component before and after it.

Hardware Components

Power Supply

The DC power supply has four voltage outputs. One output is a fixed 5 V output, which is used as the input voltage to the Wheatstone bridge and A/D Converter. The other two outputs are variable from 0 V to 35 V. For the purposes of our project, one variable output is set to 2.5 V and used as the DC offset input voltage to the summing amplifier. The other two variable outputs are a set to +15 V and -15 V, which are the bias voltages of the 741 operational amplifiers used in this project.

Wheatstone Bridge

The Wheatstone bridge contains two strain gages, each with nominal resistance of 350 Ω. One strain gage is placed on the load carrying hardware in such a way that it will measure the strain. The second strain gage is placed perpendicular to the load, and its purpose is to compensate for temperature affects. Since strain gages are extremely sensitive to small changes in temperature, using two strain gages in the Wheatstone bridge allows for changes in temperature to have little effect on the output of the Wheatstone bridge.

The Wheatstone bridge circuitry is complete with two fixed resistors of 330 Ω each. One of the strain gages is in series with a 10 Ω, multi-turn potentiometer, which accurately balances the bridge. When the bridge is balanced, the resistors satisfy the following ratio, giving an output voltage, V_{out}, of 0 V: $R_1/R_3 = R_2/R_4$

This describes the half-bridge configuration, which is illustrated in Figure 3.

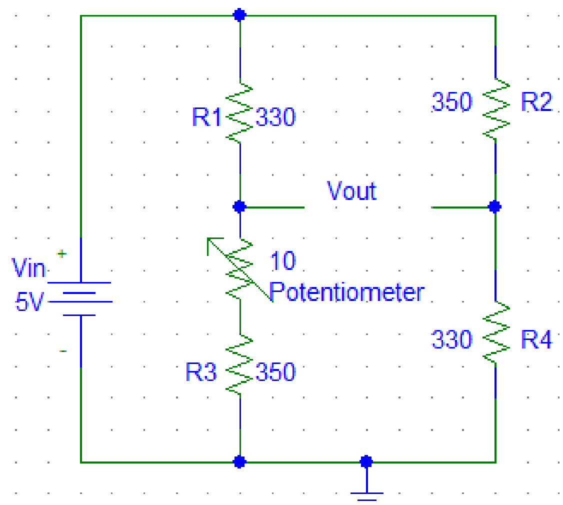


Figure 3. Wheatstone Bridge Circuitry Layout.

In Figure 2: R_2 and R_3 are the strain gages, R_1 and R_4 are 330 Ω resistors. The strain gage R_3 is in series with the 10 Ω potentiometer.

The strain gages act as variable resistors and the Wheatstone bridge measures the change in resistance that corresponds to a change in strain. The strain, ε, can be calculated from the change in resistance from the following formula:

$$\varepsilon = \Delta R/(R_g * F)$$

Where R_g is the nominal gage resistance, 350 Ω, and F is the gage factor, which is 2.11 for the strain gages used in this project. These values can be found on the packaging of the strain gage.

The output voltage, V_{out}, of the Wheatstone bridge is then measured, and is related to the change in resistance, ΔR, by the formula:

$$V_{out} = V_{in} * \Delta R/R_g$$

The output voltage can be calculated by the following formula:

$$V_{out} = (R_3/(R_3+R_4) - R_1/(R_1+R_2)) * V_{in}$$

The input voltage of the bridge circuit is 5 V, and is supplied by the power supply, that is discussed in the above section.

Amplifiers

The amplification circuitry consists of two stages of operational amplifier circuitry. The first stage is an instrumentation amplifier, and the second stage is the summing amplifier. The instrumentation amplifier consists of three operational amplifiers and resistors which amplify the voltage output of the Wheatstone bridge circuit by one thousand. The summing amplifier consists of a 741 operation amplifier and three 100 Ω resistors.

Since the input range of the PIC microcontroller's A/D converter is 0 to +5 V, in order to optimize this range the signal must be amplified and shifted up in voltage level. This was accomplished using a series of two amplifiers. The first stage is known as an instrumentation amplifier, which amplifies the signal by a factor of 1000. The second stage is a summing amplifier, which shifts the signal up by 2.5 V.

Instrumentation Amplifier

The instrumentation amplifier is shown in Figure 4 below.

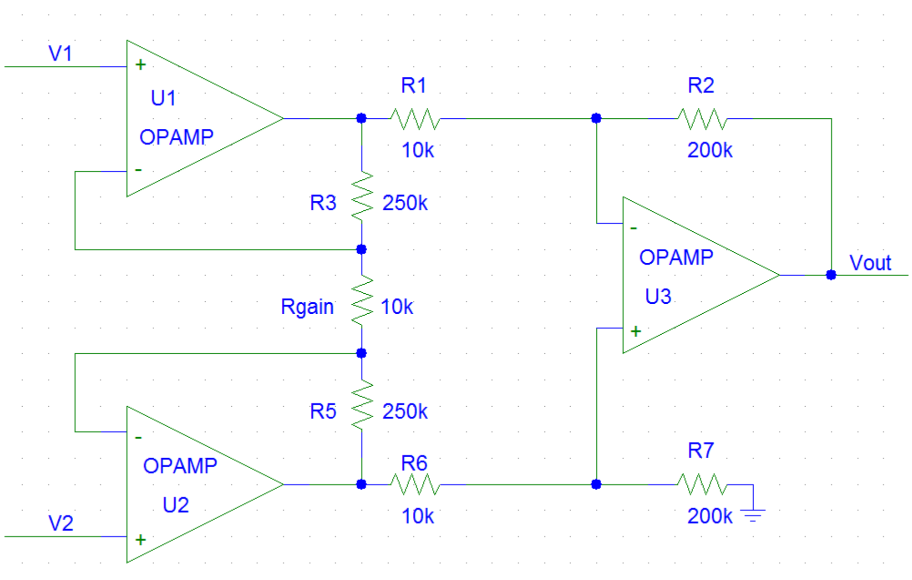


Figure 4. Instrumentation Amplifier Circuit Layout.

The circuit is made up of three 741 operational amplifiers and several resistors. The circuit is essentially a more advanced version of the simple differential amplifier, using input buffers to eliminate the need for impedance matching. The gain is calculated by the following formula:

$$\frac{V_{\text{out}}}{V_2 - V_1} = \left(1 + \frac{2R_1}{R_{\text{gain}}}\right)\frac{R_3}{R_2}$$

where $V_2 - V_1 = V_{out}$, the output voltage of the Wheatstone bridge. In order to achieve a gain of 1000 V/V, the resistor values have been chosen to be those listed in the following Table 1:

Table 1. Resistor Values For Instrumentation Amplifier.

Resistor Name	Resistor Value
R_1	250 kΩ
R_2	10 kΩ
R_3	200 kΩ
R_{gain}	10 kΩ

The resistor R_{gain} is a fixed resistor, but if the client requires a variable gain, it can easily be replaced by a potentiometer.

Summing Amplifier

The final stage is the summing amplifier which is a simple amplifier with a gain of 1, which shifts the input by a DC offset of 2.5 V. Since the output of the instrumentation amplifier is in the range +2.5 V to -2.5 V, and the input range of the A/D converter is 5 V, the signal is shifted up by precisely 2.5 V to meet the correct input range. The summing amplifier in Figure 5 shows how this is accomplished with a 741 operational amplifier and three 100 kΩ resistors.

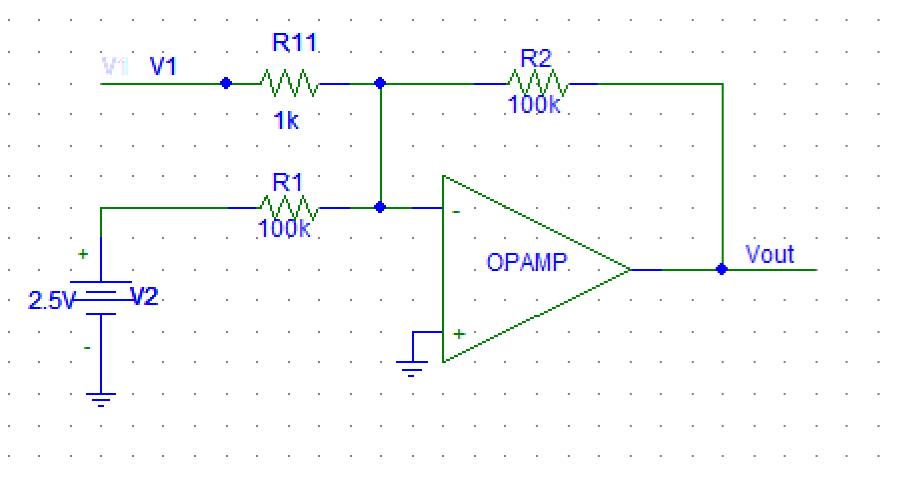


Figure 5. Summing Amplifier Configuration.

In Figure 5, V_{in} is the output of the instrumentation amplifier and V_3 is a steady 2.5V from the DC power supply. The output of the summing amplifier of unity gain is a signal shifted up 2.5 V.

The output of the summing amplifier is given by:

$V_O = -(V1 + V2)$

Low-Pass Filter

The final stage of the analog circuitry is a low-pass active filter, also known as an anti-aliasing filter. The filter circuitry is made up of a low-pass filter which consists of a 741 operational amplifier, a 350 µF capacitor, and two 100 kΩ resistors. The low-pass filter eliminates high-frequency noise and prevents aliasing of the analog signal when it is converted to digital. Figure 6 shows the circuit layout of the low-pass active filter. It is a simple first-order RC filter.

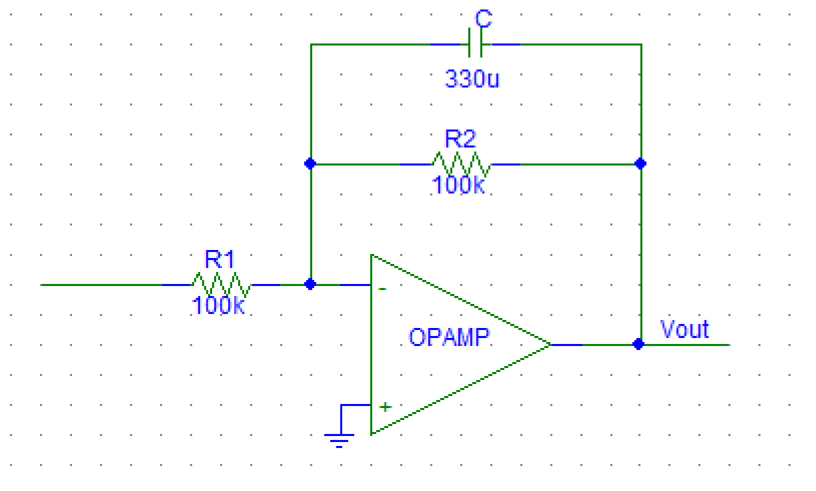


Figure 6. Low–Pass Filter Schematic.

The value of the capacitor determines the cutoff frequency of this filter using the equation below:

$$f_{cutoff} = \frac{1}{2\pi R2C}$$

Therefore the filter has a cutoff frequency of the filter is 0.00482 Hz. The value of the capacitor C_2 is a fixed capacitance, but if the client requires a different cutoff frequency, it can be easily replaced with another capacitor.

Another purpose of the low-pass filter is to remove conducted noise and prevent sampling aliasing, an effect that occurs if there are frequencies within the input signal that are greater than half the sampling frequency. The sampling frequency is significantly higher than the cutoff frequency chosen.

The output voltage as a function of radian frequency is given as follows:

$$V_{out} = \frac{-R_2 \quad * \quad 1}{R_1 (1 + j\omega R_2 C)} * V_{in}$$

Note the negative sign for both the summing amplifier gain and the low-pass filter gain. They cancel each other out so that our final input signal to the A/D converter has the same sign as the original output signal of the Wheatstone bridge.

Noise Reduction

Preliminary calculations show that the output voltage of the Wheatstone bridge will vary from +2.5 mV to -2.5 mV. Ideally, the circuitry has the resolution to measure ±1 με, which corresponds to a bridge output voltage of ±2.5 μV. With such a small signal, it was extremely important to minimize the noise in our circuit.

To modify the circuit and minimize noise, six different steps were taken that include; using low-noise resistors, shielded wire, bypass capacitors, steel metal casing, and grounding all voltages sources to a common ground. To reduce conducted noise, by pass capacitors are connected between each voltages source and ground.

First, the circuit is built using low noise resistors of 1/8 watt and 5% precision which are more resistant to noise. Additionally, twisted, shielded wires are used to connect from the bread board to the voltage supply and strain gages. This shielded twisted wire reduces severe electromagnetic radiation and pickup. Furthermore, we kept the leads in the circuit as short as possible because long leads act as antennas and pick up a variety of electric and magnetic interference.

To reduce conducted noise, bypass capacitors are connected between each voltage source and ground. The bypass capacitors also stabilize the input supply voltage from the power supply. All voltage sources are connected to a common ground to minimize radiated noise. Finally, the device is enclosed in a steel metal casing which blocks off any other electric and magnetic interference. A photo of the stell metal casing can be found in Appendix C.

LabJack

The LabJack is a 12-bit ADC that connects to a computer through USB. It provides drivers that can configure its settings for data collection in MATLAB. It is able to sample data every three milliseconds. Figure 7 below shows the LabJack.

Figure 7: LabJack

The output of the analog circuitry is imputed into the LabJack via a single wire. The internal ADC converts the analog signal into a digital signal that can be read by the computer. To make the LabJack perform the way it needed to, MATLAB was required.

The MATLAB script sets up the LabJack and collects data for a specified period of time and creates both a voltage vs. time and a strain vs. time graph from the collected data.

Computer

The analog circuitry is connected directly to a computer running Windows XP.

Since the LabJack has an internal analog-to-digital converter, the signal from our analog circuit is automatically converted. Also, because the LabJack is controlled through MATLAB nothing really needs to be done, aside from plugging everything together. A single wire from the analog circuitry is imputed into the LabJack and then by a USB cable, the LabJack is connected to the computer. The data collected and converted by the LabJack is then stored as a comma-separated (csv) file to be used later by MATLAB.

Software Components

User Interface Component

MATLAB

After the data collection was complete MATLAB was used to manipulate the data. We wrote a script that allowed for import of the file with the saved data and converted the output voltages into strain and plotted them on a graph of strain vs. time. The MATLAB script can be found in Appendix A.

Process Outcomes

The purpose of this section is to outline how the final product differs from the original design. Several changes were made to the design as problems were faced, mostly relating to noise in the analog system, and difficulties programming the Microchip PIC 24 microcontroller.

Milestones

In order to better judge our progress, weekly milestones were created at the time the design document was finalized at the end of the Fall Semester. As the Spring Semester began, it became clear that the proposed milestones needed to be reorganized to better suit the building and testing process. The following table shows how these milestones were restructured.

Table 2. Project Milestones.

	Description	Original	Description	New
1	Build Wheatstone Bridge	1/27/12	Build Amplifier/Filter	1/27/12
2	Build amplifier/filter	2/8/12	Build Low-Pass filter	2/3/12
3	MATLAB script to graph data	2/8/12	Build Wheatstone Bridge	2/8/12
4	Test each component individually	2/22/12	Develop function for serial communication	2/8/12

5	Connect components together	3/2/12	Test each component individually	2/17/12
6	Test/Debug components together	3/14/12	Develop function for A/D conversion	2/17/12
7	Test product on sheep spines	3/28/12	Test components using strain gage	2/24/12
8	First Draft of Final Report due	4/6/12	Connect Analog Components together	3/2/12
9	Final Draft of Final Report due	4/13/12	Test and Debug digital	3/7/12
10	Founder’s Day Presentation	4/17/12	Test and Debug digital and analog together	3/14/12
12	Peer evaluations and lab notebooks due	4/23/12	Develop MATLAB script to parse data	3/21/12
13	N/A	N/A	Complete first user test on sheep spines	3/28/12
14	N/A	N/A	First Draft of Final Report due	4/6/12
15	N/A	N/A	Final Draft of Final Report due	4/13/12
16	N/A	N/A	Founder’s Day	4/17/12
17	N/A	N/A	Peer evaluations and lab notebooks due	4/23/12

As Table 2 shows, the order of the Milestones was rearranged. Also, several milestones were added to better define the processes used to assemble and test the project. These changes helped the team stay on task. Our initial predictions while writing the design document did not fully account for how this project would be built. By making these adjustments early on, we were able to stay on task, for the most part. The delayed milestones will be discussed further in following sections.

Analog Outcomes

Since the project was implemented by separating the analog and the digital parts among the team members, the outcomes for the project are separated accordingly. In general, the analog part of the project went significantly smoother than the digital part. Very few changes were made to the initial design, and only a few minor problems were encountered, stemming from incorrectly placing components and significant sources of noise in the testing environment.

Assumptions

Several assumptions were made throughout the design process during the Fall Semester. One assumption that was fairly crucial was that the parts needed could be purchased and delivered quickly, and the parts functioned the way we expected them to. This assumption proved mostly valid, especially with regards to the analog circuitry. The analog circuit is comprised of mostly resistors, capacitors, and operational amplifiers. All of these components were readily available in the senior design laboratory, and did not need to be ordered. One part that needed to be ordered was the multi-turn potentiometer for the Wheatstone bridge. These were available for purchase online, were delivered very quickly after being ordered, and worked as expected.

The only part that presented some challenges in ordering was a low-noise operational amplifier. The parts supplied in the senior design laboratory are 741 operational amplifiers, which are not specifically design to be low noise. Ideally, low-noise operational amplifiers would help keep our circuit as accurate as possible. It was a challenge to find low-noise operational amplifiers to be purchased in the limited quantity we wanted. At the very least, the circuit required five operational amplifiers, but ideally we wanted to order around 25 to have as back-ups. All the suppliers we contacted only sold this product in large rolls of several hundred to a thousand, so we decided to use 741 operational amplifiers for our testing purposes.

In the design document, it was assumed that the strain gages that would be used would all be approximately the same and no calibration would need to be done on the strain gages to make them work with our project. In a sense, this assumption was true, because the strain gages themselves do not need to be calibrated, and in fact, cannot be calibrated. But each time the circuit is turned on, the potentiometer in the Wheatstone bridge must be adjusted to set the voltage output of the Wheatstone bridge to zero. Though this is slightly tedious for testing purposes, it is extremely important to zero the circuit before collecting data to make sure the data is as accurate as possible. The potentiometer is a very simple solution to adjusting the circuit to get reliable data every single time.

Another assumption made was that the Wheatstone bridge is the best method for compensating for temperature differences in the strain gages. Research throughout the building process confirms that this circuit is the most commonly used method when dealing with strain gages, so we are confident this assumption was valid. Another assumption was that the output of the Wheatstone bridge would be less than 2.5 mV. Once the circuit was built, it was easy to confirm that this was true. This also confirmed the assumption that this signal needed to be greatly amplified.

Changes

Very few changes were made to the analog circuit's design during its implementation. The only notable change made was the placement of the potentiometer in relation to the strain gages in the Wheatstone bridge circuit. Figure 8 shows how the potentiometer changed position.

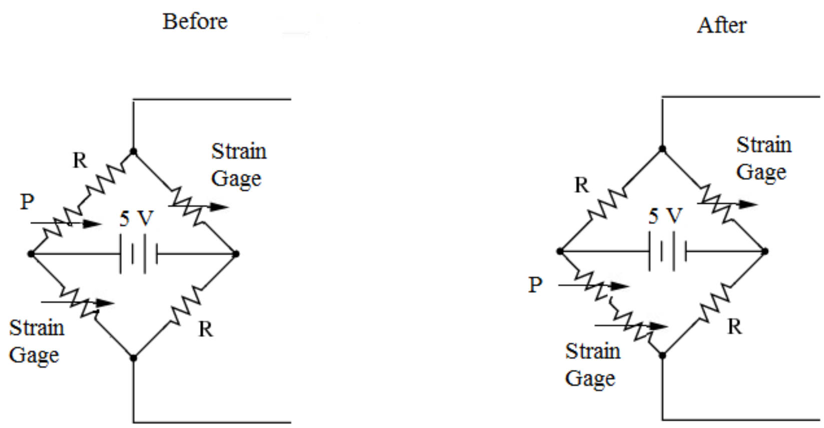


Figure 8. Change in Potentiometer Placement

Figure 8 shows the potentiometer, the variable resistor labeled as P, and to where it was moved. It was previously meant to be in series with one of the constant resistors, shown in Figure 1 as R, and the output taken between the potentiometer and the strain gage. Instead, we found the best results with the potentiometer in series with the strain gage, and the output taken between the constant resistor and the potentiometer.

A few minor additions were made to the circuit design to decrease the noise in the system. One significant addition was the use of bypass capacitors placed at the inputs of all the power supplies to the circuit board. Figure 9 shows how a bypass capacitor is connected.

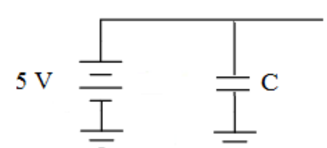


Figure 9. Bypass Capacitor Placement.

A bypass capacitor is placed between the positive terminal of a voltage source and ground, as shown in Figure 9. This keeps the voltage from jumping around too quickly, which introduces noise into the system. Since four different voltage levels were used for power supply, +15V, -15V, +2.5V and +5V, four bypass capacitors were needed to keep the input voltage as constant as possible.

Another addition to the design was to use shielded, twisted wires for all the power supply cables to the circuit board. The use of shielding helps block electric fields from introducing noise into the system. The shields are all tied to the same ground to minimize ground loops that can also add noise to the system. Twisting the wires helps minimize the effect of magnetic fields. Special cables were constructed using shielded, twisted wire to use for this project.

Problems Encountered

A few problems were faced as the analog circuit was built. The first problem faced was due to a misunderstanding about the pins for the 741 operational amplifier. Two similar pin layouts are shown in Figure 10.

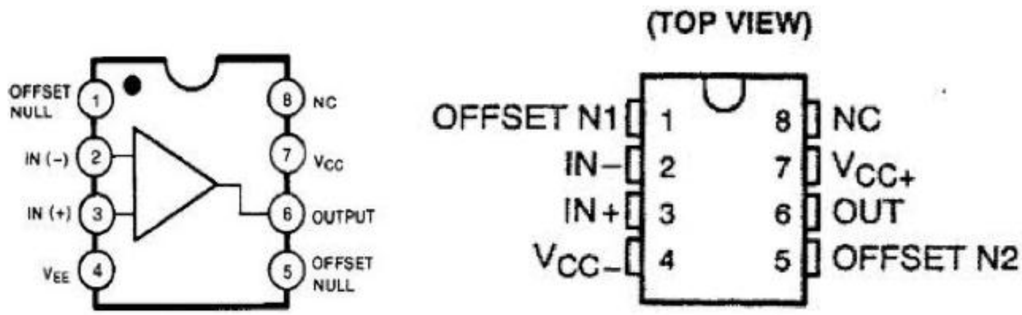


Figure 10. Two Pin Layouts for the 741 Operational Amplifier.

The first pin layout is what we initially were using to design our circuit board. As Figure 16 shows, this diagram does not specify which of the voltage pins, V_{CC} or $V_{EE,}$ is the +15 V DC input and which is the -15 V DC input. Mistakenly, the original circuit was wired with -15 V DC going to the V_{CC} pin, and +15 V DC going to the V_{EE} pin. As the second diagram clearly shows, this is the opposite of how a 741 operational amplifier should be biased. This mistake in wiring caused many 741 operational amplifiers to be fried. Once the mistake was realized, and the 741 operational amplifiers were replaced, the circuit worked correctly.

Unfortunately this problem caused the milestone for testing each component individually to be delayed from 2/17/12 to 2/24/12. Luckily, even though troubleshooting took a week, the testing of each component was very easy, and the following deadline, to test the components with the strain gage, could be completed the same day.

Another problem encountered was that 741 operational amplifiers are extremely sensitive and needed to be replaced fairly frequently. On several occasions, the replacement operational amplifier was placed one pin too high or low than it was supposed to, which caused severe problems. In one case, a 741 exploded because it was shifted down one pin, so that the +15 V DC power supply was going into the NC pin. After several mistakes, we learned to be especially careful in checking where to place the 741 operational amplifiers when they need to be replaced.

Another large problem this project constantly had to face was working in an extremely noisy laboratory environment. All the initial testing was done without implementing the low noise techniques, which caused the results to vary greatly, causing questions about whether or not the circuit was working correctly. Once bypass capacitors, shielded power cables, and a shielding metal box were included in the project, the circuit worked as expected. The noise was drastically noticeable on an oscilloscope when the gain of the circuit was measured on several occasions. On one occasion, if the lid to the box was open, the gain was roughly 1000, and if the lid of the box was closed, the gain decreased to about 500. This was a huge change, and it demonstrated how noisy the environment was where this project was being tested. The noise reducing techniques employed have helped alleviate this problem.

Risks

The risks associated with the analog portion of this project were not anticipated very well in the design document. The only risk that was acknowledged was that delivery of parts might take longer than anticipated. In reality, this was never a problem, since everything that was ordered from a manufacturer was delivered in less than three days.

The biggest risk that was not well accounted for was the likelihood that noise would significantly affect our signal. It was probably the biggest source of uncertainty in this project, and had it not been appropriately handled, could have made the project worthless. Luckily, by using a few noise reducing techniques, such as using bypass capacitors at the voltage source inputs, shielding the voltage source inputs with special cable, and enclosing the entire circuit in a steel box, the noise in the system was drastically reduced, so that the signal can be accurately measured.

Digital Outcomes

Assumptions

One of the greatest assumptions in the project was that an analog voltage signal could be sampled, converted to a digital value, and outputted to a computer. Although not in the assumptions of the functional specifications, the design assumed that using a Microchip PIC24 microcontroller would be similar to using the PIC18 microcontroller, a technology that the team members were familiar with. The team had a basic understanding of assembly language and believed there were enough similarities between architectures that it would be easy to program a PIC24. Although the PIC24 and PIC18 assembly languages are similar, their assembler directives and syntax are different enough that it took a long time to set up the code used on the chip.

Although the PIC24 microcontroller could be programmed in assembly language similarly to the PIC18, it is much more difficult to do so and is less documented. In order to get this working, the manufacturer of the PIC had to be contacted to help figure out how to program the PIC24 in assembly language.

Changes to Design

The primary change to the design is how the analog voltage of our circuit is collected and sent to the computer to be graphed.

Once the PIC24 microcontroller was programmed in assembly language, the primary focus of the project was getting serial communication working between the microcontroller and computer to make sure the collected analog data could be sent to the computer. Once the PIC was programmed, the appropriate signals were not being outputted to the I/O Expansion Board.

By contacting the manufacturer, the team learned that the PIC24 had a pin remapping module that needs to be configured to output pins to the board. Once the PIC24's pin layout was verified, connected to the I/O Expansion board, and had its pin remapping module set up, the serial signal was transmitted out of the desired pin.

Table 3. Pin Layout of I/O Expansion Board.

TABLE 1: HEADER INTERFACE MAPPING

J1 Pin #	J2 Pin #	J3 Pin #	J4 Pin #	J10 Pin #	J11 Pin #
1	65	65	65	1	—
3	68	68	68	4	—
4	73	73	73	—	1
5	67	67	67	3	—
6	74	74	74	—	3
7	116	116	116	5	—
8	66	66	66	—	4
9	115	115	115	6	—
10	27, 60	27, 60	27, 60	—	6
13	114	114	114	7	—
14	29, 62	29, 62	29, 62	—	5
15	113	113	113	8	—
16	30	30	30	—	8
17	112	112	112	9	—
18	28	28	28	—	7
19	111	111	111	10	—
20	105	105	105	—	10
21	110	110	110	11	—
22	106	106	106	—	9
23	109	109	109	12	—
24	99	99	99	—	12
25	98	98	98	13	—
26	100	100	100	—	11
27	104	104	104	15	—
28	97	97	97	14	—
29	103	103	103	16	—
30	89	89	89	—	14
31	90	90	90	17	—
32	—	—	—	—	13
35	85	85	85	18	—
36	—	—	—	—	16
37	86	86	86	19	—
38	—	—	—	—	15
39	87	87	87	20	—
40	96	96	96	—	18
41	88	88	88	21	—
42	95	95	95	—	17
44	94	94	94	—	20
45	35	35	35	23	—
46	93	93	93	—	19
47	37	37	37	24	—
48	—	—	—	—	21
49	39	39	39	25	—
50	—	—	—	—	24
51	33	33	33	26	—
52	—	—	—	—	23
54	102	102	102	—	26
58	101	101	101	—	25
62	—	—	—	—	29
63	77	77	77	29	—
64	14	14	14	—	30
65	—	—	—	31	—
66	13	13	13	—	31

J1 Pin #	J2 Pin #	J3 Pin #	J4 Pin #	J10 Pin #	J11 Pin #
67	—	—	—	30	—
68	—	—	—	—	32
69	—	—	—	32	—
70	12	12	12	—	33
71	43	43	43	33	—
72	11	11	11	—	34
73	44	44	44	34	—
74	40	40	40	—	36
76	38	38	38	—	38
79	49	49	49	36	—
81	50	50	50	35	—
83	17	17	17	38	—
84	6	6	6	—	35
85	18	18	18	37	—
86	8	8	8	—	37
87	—	—	—	42	—
88	2	2	2	—	41
90	4	4	4	—	43
91	3	3	3	41	—
92	20	20	20	—	45
93	5	5	5	44	—
94	19	19	19	—	47
95	7	7	7	43	—
97	1	1	1	46	—
101	79	79	79	47	—
102	—	—	—	—	52
103	80	80	80	48	—
105	81	81	81	49	—
106	52	52	52	—	42
107	82	82	82	50	—
108	51	51	51	—	44
109	—	—	—	51	—
110	34	34	34	—	46
111	—	—	—	52	—
112	36	36	36	—	48
113	45	45	45	53	—
114	—	—	—	—	51
115	46	46	46	54	—
116	—	—	—	—	53
117	—	—	—	55	—
118	72	72	72	—	54
119	—	—	—	56	—
120	71	71	71	—	56
123	—	—	—	57	—
124	70	70	70	—	58
125	—	—	—	58	—
126	69	69	69	—	60
127	83	83	83	59	—
128	—	—	—	—	55
129	84	84	84	60	—
130	78	78	78	—	57
132	—	—	—	—	59

By tracing the pin layout in Table 3 above, the PIC24's output pins were verified as properly connected.

Because the computer still didn't recognize the signal, research showed that the PIC24's Universal Asynchronous Receiver/Transmitter (UART) needed to be amplified and inverted using a device called a line driver. This would amplify the UART's signal to the RS-232 protocol so that the computer could properly understand the signal. The team decided to order the RSLink2 line driver because it provided pin connections that could be made with the I/O Expansion board, the line driver chip preconfigured with capacitors, and the outputs of the chip connected to a 9 pin female serial connector to easily interface with a computer. This device is shown in the Figure 11 below.

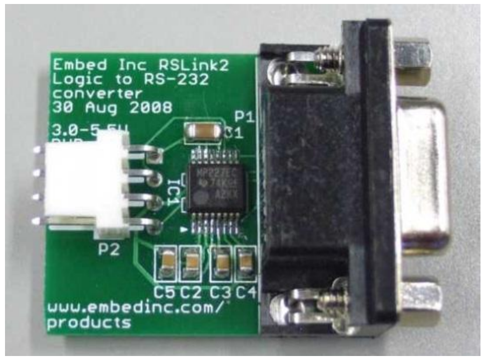


Figure 11. RSLink2 Line Driver.

By this point, the building phase became so focused on getting the PIC24's serial module working, it was decided that there would not be enough time to test and configure the PIC24's ADC module. As an alternative, it was decided to use an external ADC that could automatically convert an analog voltage to a digital signal, that way the analog data could easily be read and sent over the serial port to the computer.

Because of the many different models of ADC, it was decided to get one with 12 bits of precision and didn't require some special communication protocol [such as SPI or I2C] to read its output. An ADS7842EB chip was eventually ordered that had 12 bits of precision and parallel outputs to read its results. The layout and interface for how this chip works can be seen in the Figure 12 below.

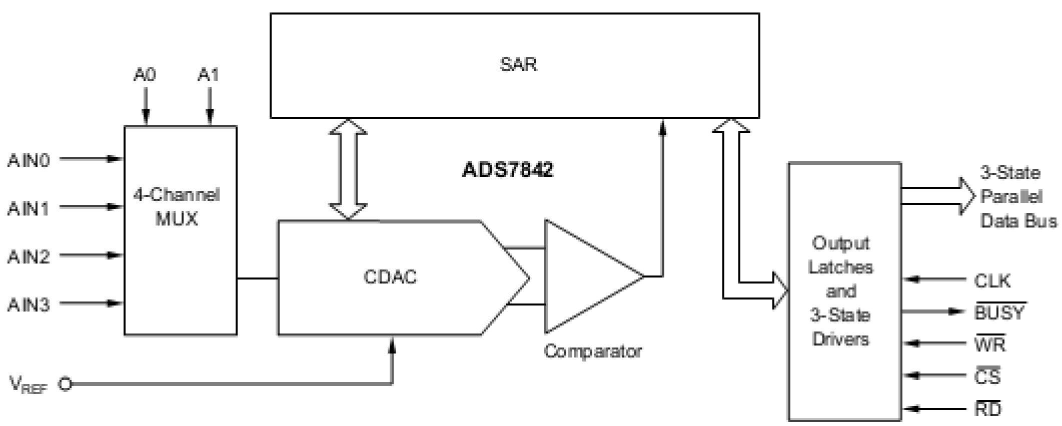


Figure 12. Block Diagram of How ADS7842EB [ADC] Functions.

Although the serial signal was now at the appropriate voltage levels using the line driver, the computer still could not recognize the signal coming from the UART. By contacting the manufacturer and referring to serial communication tutorials online, the team learned that some of the computer's default settings needed to be changed and that there are different types of serial cables; straight-through serial cables and cross-over null modem cables that change the pin layout between devices. The differences between how these cables are wired are shown in the Figures 13 & 14 below.

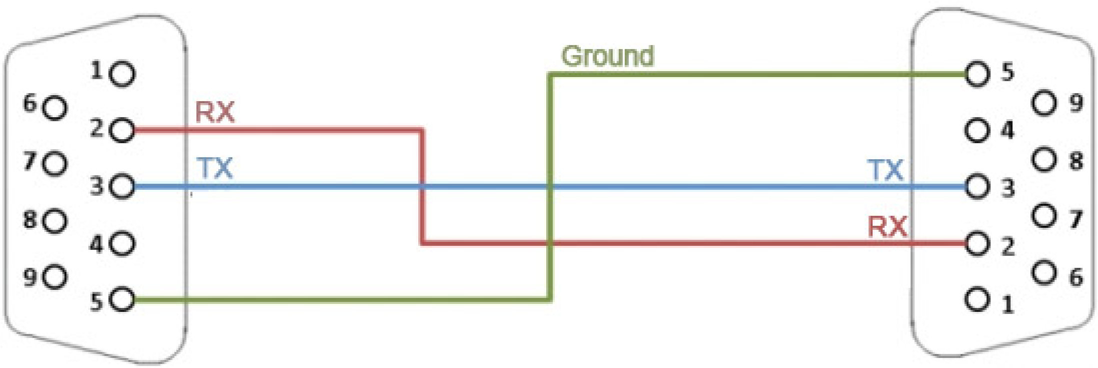


Figure 13. Wiring Layout of a Straight-Through Serial Cable.

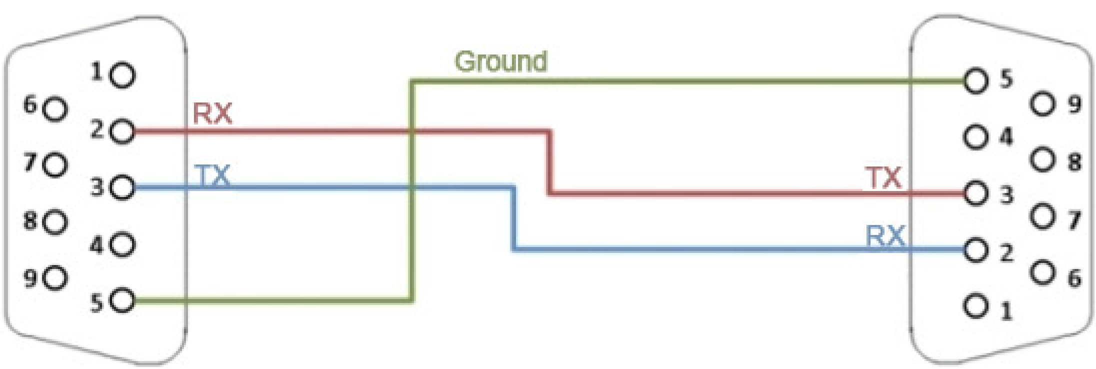


Figure 14. Wiring Layout of a Null Modem Cable.

Once the proper settings were disabled and the PIC24 microcontroller was connected to the computer using a straight-through serial cable, the computer was finally able to recognize the signal from the PIC24.

However, the characters the PIC24's UART were sending were not the characters the computer was receiving. The computer's COM port was verified that it was working correctly to ensure that the COM port was not broken. This was accomplished by making the COM port's transmitter send characters to its receiver. Because the computer received the characters that it sent, it was verified to be working correctly.

Once this was done, an attempt to send a limited number of characters and adjusting the baud rate to verify our results was made, but the proper characters were still not being received. Since it was April 27th and there were only a few weeks till Founder's Day, it was decided to stop using the PIC24 and use another device to accentuate data collection. It was decided to use a LabJack U3-HV module, a device that could collect analog voltages and output their results to a spreadsheet. This module can be seen below in Figure 15. A benefit to using this device was that it could be configured using MATLAB, so few changes would need to be made to our post process data analysis MATLAB script.

Figure 15. LabJack Module.

Once the drivers for the Lab Jack were installed, we were able to easily set it up and verify that the correct voltages were being collected by the LabJack.

Risks

The greatest risk to the project was spending so much time getting the PIC24's serial port working. Too much time was spent on this instead of immediately finding alternatives once problems arose. Alternatives should have been pursued sooner to give adequate time to finish the project. The reason this did not occur was the manufacturer was able to get the code to work, so progress was made by configuring the project to be made like theirs, hoping that it would eventually work. One of the ways this dilemma could also be avoided was to have more plans for alternatives and ways to integrate them into the design easily so they wouldn't have to be integrated hastily and at the last minute.

Resource Requirements

The biggest change in the budget occurred when the PIC24 microcontroller started having trouble using the serial port. The budget was adjusted for extra serial cables, the line driver to amplify the UARTs signal, and external ADC, but these were reasonable adjustments to the original budget. Once it was decided the PIC24 would not work, the budget now had to include the backup LabJack system because the design had changed, which drastically changed our budget.

Another resource underestimated was finding faculty who understood problems configuring the PIC microcontroller. Although documentation material existed in the form of contacting the manufacturer and manuals, there were few online tutorials for configuring the PIC24 microcontroller. Also the faculty with the most experience with PIC microcontrollers, Dr. Wayne Lu, was on sabbatical so he could not directly assist with the design. The team was able to consult with Dr. Lu by email though and he was able to help debug some errors within the code. Because of this limitation, it should have signaled the team to switch to an alternative sooner so that the device would be more familiar to available faculty and might have better tutorials and documentation.

Conclusions

Some testing on the SATEC loading machine in the School was done in order to determine that the device worked as expected. The expected outcome was that the strain on the device would have an initial start of zero strain and jump extremely high due to the metal device carrying all the strain. Overtime the strain on the device would decrease as a result of the bone growth and the device carrying less and less strain. Figure 16 on the next page displays some sections of the collected data. It is a tab-delineated spreadsheet that shows tension and then compression.

0.098000	2.768066
0.099000	2.777303
0.100000	2.798323

72.119000	2.822847
72.120000	2.842912
72.121000	2.845779
72.122000	2.856289
72.123000	2.871577

119.662000	2.969036
119.663000	2.975724
119.664000	2.983368
119.665000	2.994197
119.666000	3.004070
119.667000	3.009485
119.668000	3.001522
119.669000	3.016492

Figure 16. Data Collection.

In this figure the initial starting point is 2.7 volts due to certain circumstances, however it still shows that it starts at zero, jumps high and then slowly tapers off. It also illustrates that data was collected and our teams' specifications were indeed met.

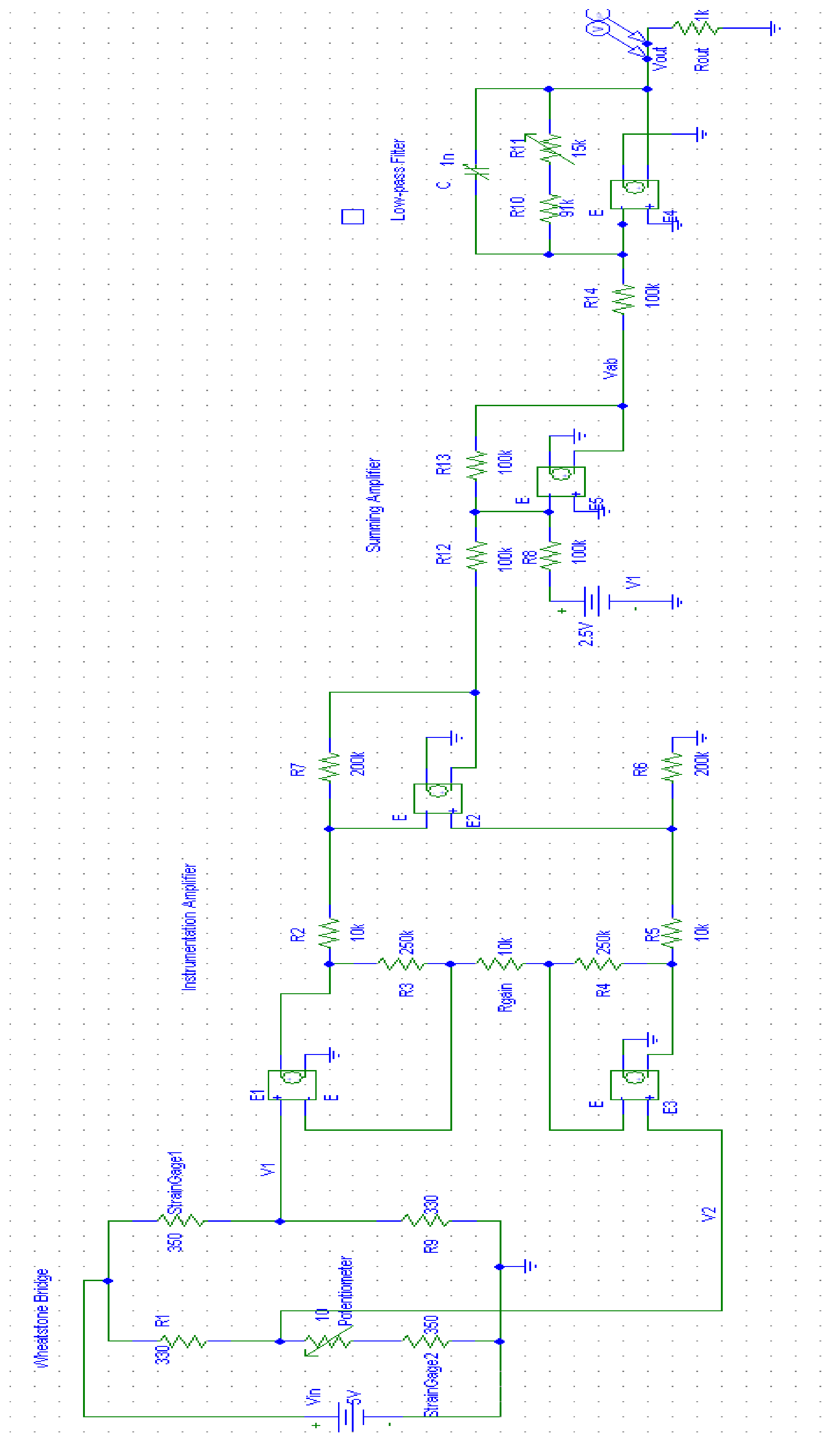
From this project, we were successfully able to design a spinal fusion strain gage reading and recording system. This project included both analog and digital components interfacing with each other to accomplish our data collection system. Although the analog design was very easy to build, there was trouble building our initial digital design that forced us to rely on alternative designs. From these experiences, we have learned to explore alternatives to designs much sooner, because if one problem exists that can't be worked around, then the final design will not work. We also learned a lot about serial communication, device interfacing, and noise reduction technique, which are important skills in real practice. For future Electrical Engineering students doing design projects, we recommend that they understand any programming technologies they may use in their projects during the Fall so that they don't run into trouble figuring out how to program their device in Spring.

Appendix A: MATLAB Script for the LabJack

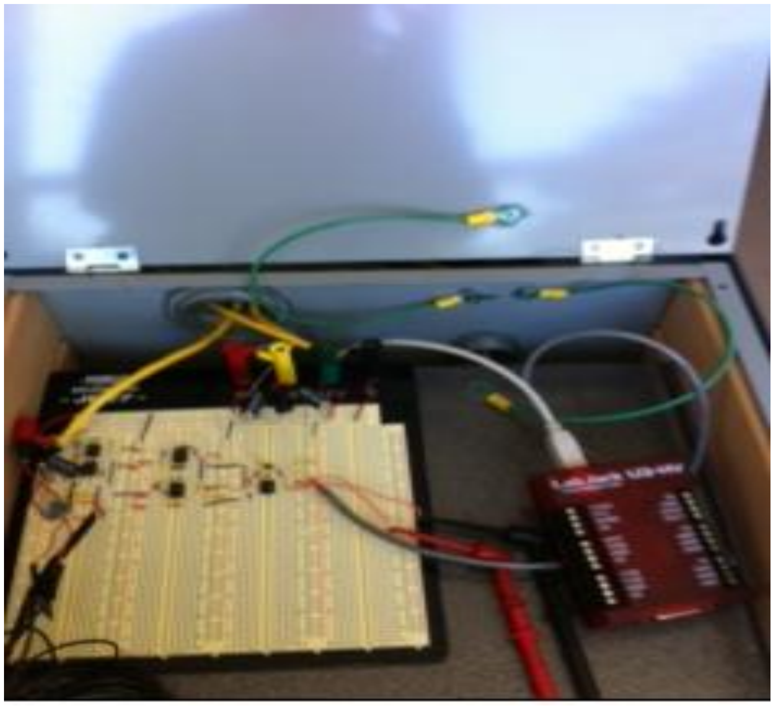
```
Function [M] = Collect_Data( run_time, gain, voltage_name,
strain_name, data_name)
% Function name: Collect_Data
%
% Function description: This fuction continually collects the data
for the
% specified run_time from the AIN0 input of the LabJack and stores
it
% into the matrix M. It graphs the voltage versus time, and the
strain
% versus time, and stores the data in an excel spreadsheet.
%
% Function Inputs:
% run_time is the total time of the trial in seconds
% gain is the gain of the amplifier circuit (1000 V/V generally)
% voltage_name is the name of the file to store the voltage vs time
graph
% strain_name is the name of the file to store the strain vs time
graph
% data_name is the name of the file to store the raw data
% ------------------------------------------------------------------
------

%initialize the LabJack. This code was provided by the LabJack
website

clc % Clears the command window
clear global % Clears all global variables

ljud_LoadDriver; % Loads LabJack UD Function Library
ljud_Constants; % Loads LabJack UD constant file
[Error ljHandle] = ljud_OpenLabJack(LJ_dtU3,LJ_ctUSB,'1',1); %
Returns ljHandle for open LabJack
Error_Message(Error) % Check for and display any Errros

%Start by using the pin_configuration_reset IOType so that all
%pin assignments are in the factory default condition.
[Error] = ljud_ePut(ljHandle, LJ_ioPIN_CONFIGURATION_RESET, 0, 0,
0);
Error_Message(Error)

%First some configuration commands.  These will be done with the
ePut
%function which combines the add/go/get into a single call.

%Configure FIO2 and FIO3 as analog, all else as digital.  That means
we
%will start from channel 0 and update all 16 flexible bits.  We will
%pass a value of b0000000000001100 or d12.
[Error] = ljud_ePut(ljHandle, LJ_ioPUT_ANALOG_ENABLE_PORT, 0, 12,
16);
```

```
Error_Message(Error)

% The following code was written by Team Goonies

% initialize variables
i = 1;
M = [0 0];

% Calculate number of samples  based on estimated 3ms per sample.
sample_count = run_time/(.003);

%Collect samples of data.
for i= 1:sample_count,
    %Get voltage sample
    [Error AIN0] = ljud_eGet(ljHandle,LJ_ioGET_AIN,0,0,0);
    Error_Message(Error)

    %Add voltage to row of matrix
    M(i,2) = AIN0;
    M(i,1) = i * .003;
    %Go to next sample.
    i = i+1;
end

%Graph the voltage vs time
plot(M(:,1),M(:,2))
xlabel('Time (s)');
ylabel('Voltage (V)');
title('Voltage vs Time Graph');
saveas(gcf,voltage_name,'tif')

%convert to strain
M(:,2) = M(:,2) - 2.5;%subtract 2.5V
M(:,2)   =   M(:,2)./gain;   %deamplify   by   dividing   by   gain   of
instrumentation amp
M(:,2)   =   M(:,2).*350/5;  %convert   to   change   in   resistance   by
multiplying by 350 and dividing by 5
M(:,2) = M(:,2).*2.11/350; %convert to strain by dividing by 350 and
multiplying by 2.11

%graph the strain vs time
figure
plot(M(:,1),M(:,2))
xlabel('Time(s)')
ylabel('Strain')
title('Strain vs Time Graph');
saveas(gcf, strain_name,'tif')

%save raw data in excel file
xlswrite(data_name, M)

end
```

Appendix B: Analog Circuitry PSPICE LAYOUT



Appendix C: Final Product in Metal Casing



Appendix D: User Manual

Step 1:

Make sure LabJack software is downloaded onto computer being used to collect data. The software can be found at the following website:

http://labjack.com/support/windows-ud

Choose the link that corresponds to the computer in use. It will likely be the first link.

Step 2:

Connect the LabJack via the USB port on the computer. This will then prompt you to install the hardware. Follow the steps in the set up window.

Step 3:

Download the Driver and Examples for MATLAB from the following website and unzip the folder:

http://labjack.com/support/ud/examples/matlab

Step 4:

Download the MATLAB script attached in the email. This script is named Collect_Data.m. This needs to be saved in the following subdirectory of what was downloaded in Step 3:

MATLAB_LJUD -> LJUD_FUNCTIONS

Step 5:

Start the MATLAB software.

Step 6:

Change Current Directory to the subdirectory titled LJUD_FUNCTIONS.

Step 7:

Solder the twisted, shielded wire we have supplied to you (this eliminates noise) to the strain gage solder pads. Connect the other end of this wire to the labeled places on the circuit board. Figure 1 highlights the places to connect. It does not matter which strain gage goes in which position. One strain gage should be perpendicular to the load, and one should be parallel to the load.

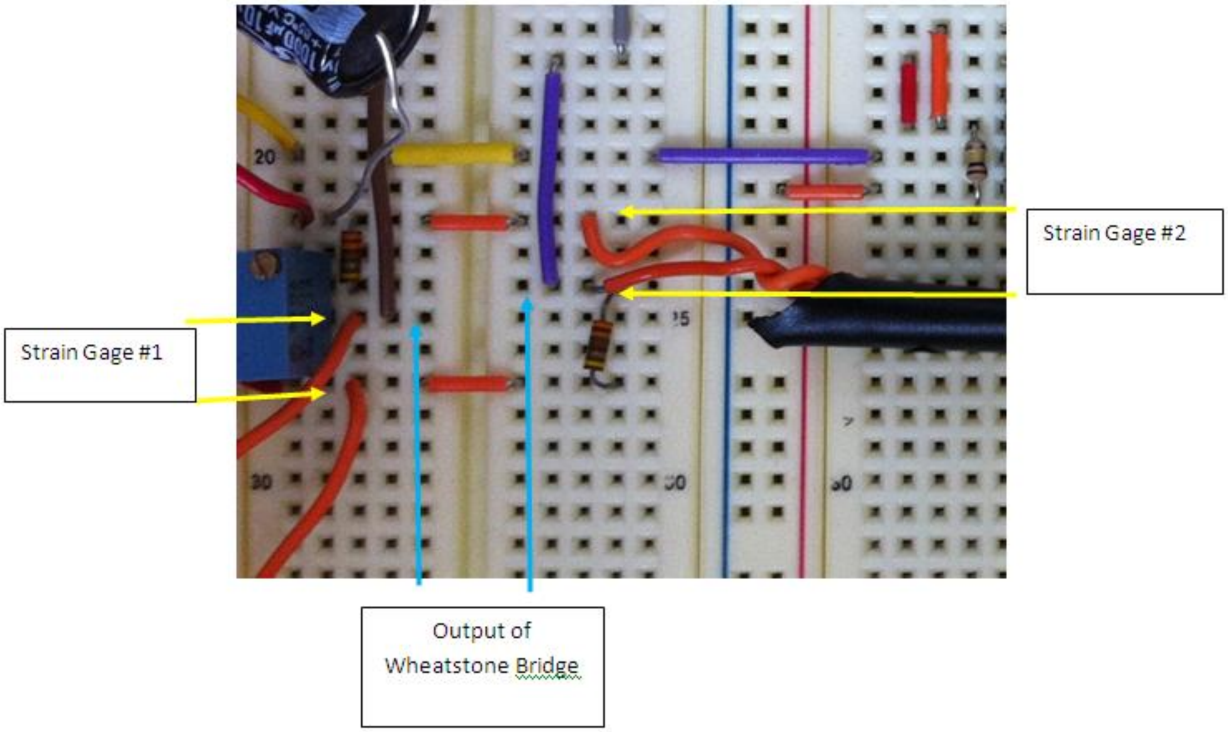


Figure 1. Wheatstone Bridge.

Step 8:

Load the spine gages with the hardware and the strain gages into the SATEC machine.

Step 9:

Turn on the voltage power supplies to the circuit, starting with one with red, yellow, and black leads. In general, you should avoid disconnecting the power cords attached directly to the power supplies. Disconnecting at the circuit board will eliminate any confusion, since they are color coded at the circuit board connection, and not at the power supply connection. Figure 2 shows how the power cords are attached to the circuit board, and Figure 3 shows how the power cords are attached to the power supplies.

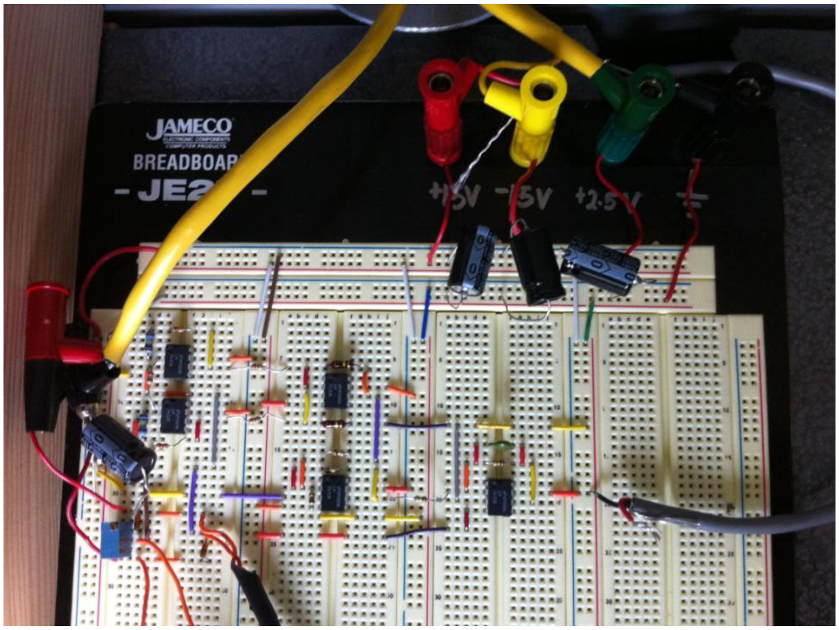


Figure 2. Circuit Board Connections.

Figure 3. Power Supply Connections.

Step 10:

Using a mulitmeter, on the millivolt setting, measure the output of the Wheatstone Bridge. On Figure 1, the output of the Wheatstone Bridge is where the blue arrows point.

Step 11:

While still monitoring the output of the Wheatstone Bridge, use a small non-metallic screw driver to turn the blue potentiometer until the output voltage is zero. This step must be done after the spine and hardware have been loaded into the SATEC machine.

Step 12:

Once you are ready to collect data, type the following in the MATLAB Command Window:

```
Collect_Data( run_time, gain, voltage_name, strain_name, data_name )
```

Where:

run_time is the time in seconds the trial will take.

Gain is the gain of the amplifier circuit, which will generally be 1000

Voltage_name is the file name of where you would like to store the voltage graph. For example, you could type 'voltage'

Strain_name is the file name of where you would like to store the strain graph. For example, you could type 'strain'

Data_name is the file name of where you would like to store the spreadsheet of raw data. For example, you could type 'data'

An example of how to use this function is:

```
Collect_Data( 60, 1000, 'voltage', 'strain', 'data' )
```

The files generated will be saved in the current directory that MATLAB is operating in. The graphs will be saved as .tif files, and the spreadsheet will be saved as an .xls file. The excel file does not have any header information. The first column generated is time in seconds, and the second column is strain in units of strain.

Step 13 (Troubleshooting errors):

Sometimes the first time the MATLAB code runs, it gives you an error. Try running it a second time, and hopefully it should work.

If you continue to have issues with the MATLAB code, you may have stored the files in different directories. Make sure you are in the correct directory, and all the MATLAB functions from the website, as well as the Collect_Data.m file are in the same place.

If you are continuing to experience problems, make sure you have only one set of these files. If the files exist two different locations on the same desktop, MATLAB doesn't know which one to use, and will give you an error. Make sure that only one copy of these files exists.

If the data is not coming out with reasonable numbers, there may be a problem with the 741 operational amplifiers. The easiest solution is to just replace all 5 with new chips.