

# **A System to Quantify Upper Limb Function**

By

Larissa Schudlo (0664526)

Electrical and Biomedical Engineering Design Project EE 4BI6  
Department of Electrical and Computer Engineering  
McMaster University  
Hamilton, Ontario, Canada

# **A System to Quantify Upper Limb Function**

By

Larissa Schudlo (0664526)

Department of Electrical and Biomedical Engineering  
Faculty Advisor: Dr. H. deBruin

Electrical and Biomedical Engineering Project Report  
Submitted in partial fulfillment of the requirements for the degree  
of Bachelor of Engineering

McMaster University  
Hamilton, Ontario, Canada  
April 22, 2010

Copyright © April 2010 Larissa Schudlo

## **Abstract**

Peripheral nerve damage in the upper extremity can reduce a person's range of motion, and must be immediately repaired. Post operative recovery must include a therapy program, and the assessment of motor function to ensure adequate rehabilitation.

Traditional assessments are qualitative, and can lead to inaccuracies in the evaluation process. Nerve injuries in the upper limb are prevalent in pediatric patients, which further increases the difficulty in providing an accurate assessment. A device that would allow clinicians to qualitatively assess a pediatric patient's range of motion using accelerometry, particularly the ability to pronate and supinate the forearm, would improve the accuracy and reliability of evaluating rehabilitation. The theory, hardware and software design, and experimental results of this instrumentation system are presented.

**Keywords:** peripheral nerve injury, upper extremity, functional recovery, range of motion, accelerometry, paediatrics, instrumentation system

## **Acknowledgements**

The author would like to thank the project supervisor, Dr. deBruin, for his continued support and encouragement during the development of this project. He helped to develop and expand the idea, provided guidance throughout the year, and involved the people in the MacHand Therapy Program with the project to give it a particularly meaningful purpose.

Recognition is also given to Carol DeMatteo from McMaster's School of Rehabilitation Sciences, and Tara Packham from the Mac Hand Therapy Program and Dr. Jamie Bain in the Brachial Plexus Clinic at the McMaster Children's Hospital. Carol and Tara helped determine the specifications for the instrumentation developed, and Carol and Dr. Bain provided the opportunity to meet and observe patients in the Brachial Plexus Clinic for project research.

Lastly, deep gratitude is expressed to Michael Willand, whose constant advice and guidance was fundamental in the completion of this project. He shared his extensive practical knowledge regarding electronics, answered numerous questions, and was extremely helpful in overcoming many of the challenges that arose during the course of the project. He also provided hardware components. His time and patience is sincerely appreciated.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Contents</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>Nomenclature</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Literature Review</b>	<b>3</b>
2.1 Traditional Clinical Assessment .....	3
2.2 Current Qualitative Assessment .....	3
<b>3 Statement of the Problem</b>	<b>5</b>
3.1 Project Objective .....	5
3.2 Methodology of Solution .....	5
<b>4 Experimental Procedures</b>	<b>8</b>
4.1 Theoretical Design .....	8
4.2 Design of Instrumented Hand Held Toy .....	8
4.3 Design of Base Station .....	11
4.4 Design of Software Data Acquisition Program .....	12
<b>5 Results</b>	<b>15</b>
5.1 Instrumentation Hardware Results .....	15
5.2 Accelerometer Calibration Results .....	16
5.3 LabVIEW Program Results .....	18
5.3.1 Displaying Past History Files .....	18
5.3.2 Displaying Current Results Being Recorded .....	19
5.3.3 Displaying Testing Results .....	20
<b>6 Conclusion</b>	<b>22</b>
6.1 Discussion .....	22
6.2 Future Recommendations .....	22
<b>A</b>	<b>25</b>
<b>A.1 Hardware Components and Specifications</b>	<b>26</b>
<b>A.2 Calibration of ADXL335 Accelerometer</b>	<b>27</b>
A.2.1 Accelerometer Parameters Required .....	27
A.2.2 Calibration Measurements .....	27
A.2.3 Measuring Angle of Rotation with an Accelerometer .....	28
A.2.4 Measuring Angular Velocity with an Accelerometer .....	30
<b>A.3 Wireless Communication</b>	<b>31</b>
A.3.1 RF Transceivers and Nrf2401 Arduino Library.....	31

A.3.2 Transmitter Microcontroller .....	31
A.3.3 Receiver Microcontroller .....	32
<b>A.4 LabVIEW SubVIs</b>	<b>38</b>
A.4.1 USB 4 Channel Read .....	38
A.4.2 Micro to Acceleration Conversion .....	39
A.4.3 Offset Calibration .....	39
A.4.4 Acceleration to Degrees .....	40
A.4.5 Angular Velocity .....	41
A.4.6 Drawing Rotation .....	41
A.4.7 History File Display .....	42
A.4.8 Measuring Rotation .....	44
 <b>Curriculum Vitae</b>	 <b>47</b>
 <b>References</b>	 <b>49</b>

## List of Tables

5.2	Accelerometer calibration results .....	16
5.2	Accelerometer tilt testing .....	17
5.2	Statistical data for accelerometer testing .....	17
A.1	Hardware components and specifications .....	26

## List of Figures

3.2	Methodology of instrumentation system .....	7
4.1	Theoretical design of instrumentation system .....	8
4.2	Theoretical hardware design: handheld device .....	11
4.3	Theoretical hardware design: base station .....	12
4.4	Front panel display of rotation angle .....	12
4.4	LabVIEW front panel: Evaluating patient range of motion .....	14
5.1	Final electronic circuit: Handheld device to measure range of motion .....	15
5.1	Final electronic circuit: Base station .....	16
5.1	Final electronic circuit: Handheld device .....	16
5.2	Accelerometer calibration experimental error .....	17
5.3.1	LabVIEW front panel: Displaying two sets of previous data .....	19
5.3.1	LabVIEW front panel: Displaying previous history .....	19
5.3.2	LabVIEW front panel: Displaying real time results .....	20
5.3.3	LabVIEW front panel: User testing results .....	21
A.2.2	Accelerometer Axes of Sensitivity .....	27
A.2.2	Accelerometer output response for orientation .....	28
A.2.3	Angle of rotation about z-axis .....	28
A.2.3	Signs of acceleration with orientation .....	30
A.4.1	LabVIEW SubVI: USB 4 Channel Read .....	38
A.4.2	LabVIEW SubVI: Microcontroller to Acceleration Conversion .....	39
A.4.3	Accelerometer calibration position .....	39
A.4.3	LabVIEW SubVI: Offset Calibration .....	40
A.4.4	LabVIEW SubVI: Acceleration to Degrees .....	40
A.4.5	LabVIEW SubVI: Angular Velocity .....	41
A.4.6	LabVIEW SubVI: Drawing Rotation .....	42
A.4.7	LabVIEW SubVI: History File Display for Rotation Data .....	43
A.4.7	LabVIEW SubVI: History File Display for Angular Velocity Data .....	43



## Nomenclature

3D	Three dimensional
ADC	Analog to digital converter
Case Structure	An 'if' statement in LabVIEW
Front Panel	The user interface of a LabVIEW program
$g$	Acceleration due to gravity, $9.81 \text{ m/s}^2$
LED	Light emitting diode
RF	Radio frequency
RS	Recommended Standard
SubVI	Sub Virtual Instrument, a function written in LabVIEW to be used in another VI
USB	Universal Serial Bus
VI	Virtual Instrument, a software program written in LabVIEW

# Chapter 1

## Introduction

The upper extremity of the human body is a complex system and its wide range of functions and movements is vital to many everyday activities. The arm and shoulder produce powerful forces and rotations, while the hand performs precise motions and manipulations as well as dynamic grasps. This intricate combination of fine and gross motor skills is achieved through the upper extremity's muscles and the nerves that supply them. These nerves extend from the brachial plexus in the shoulder down to the finger tips, branching along the path. These innervations allow for the vast motion, sensation and positioning feedback capabilities of the shoulder, arm, forearm and hand. Traumas, lesions and degeneration can lead to peripheral nerve damage resulting in loss of coordination and function of this versatile body system, and have a great impact on a person's everyday life - physically and socially. Simple motions such as grasping and pinching with the hand, pronation and supination of the forearm, and external rotation, flexion and extension of the arm can be impeded or even completely hindered.

Operative repair is necessary following nerve injury or trauma. During post operative recovery, muscle strength and control must be regained. It is necessary to implement a precise and effective rehabilitation program to ensure muscle wasting does not occur and permanently hinder the patient. During the course of the rehabilitation program, periodic assessments are needed to ensure the patient is effectively regaining muscle function. Therapists must evaluate the patient's upper limb abilities and provide feedback regarding progression and necessary changes to the rehabilitation program. Currently, functional evaluation techniques employed for the upper limb are primarily qualitative. Upon a therapist's observations, the patient's abilities are evaluated, and results are compared to previous qualitative, written results. However, this technique of evaluation is subjective, and highly depends on the therapist's abilities to detect even minute muscle activity. It can be difficult to compare a patient's current abilities to past history unless drastic changes have occurred. It also relies heavily on patient communication with the therapist. Although providing an accurate assessment for any patient can be difficult, these constraints can make the evaluation of paediatric patients an

even greater challenge. A quantitative measurement of a patient's muscle coordination and strength in the upper limb would drastically improve the accuracy and reliability of patient assessment, especially for children. Quantitative data could easily be stored and compared and help to reduce variability between different therapists. A more accurate patient assessment would improve post operative recovery and increase the effectiveness of an implemented rehabilitation program. The objective of this project was to develop such as system that allows for quantitative measurement of range of motion in the upper limb, particularly aimed toward paediatric patients.

## **Chapter 2**

### **Literature Review**

#### **2.1 Traditional Clinical Assessment**

Following nerve injury and repair in the upper limb, it is important to implement a therapy regime to regain muscle function and coordination. Post operative evaluation of a patient's strength and mobility is an essential part of the rehabilitation program as it indicates the effectiveness of the rehabilitation program and the patient's progression in recovery. Clinical tests for the upper limb include measurements of grip and pinch strength, range of motion and dexterity [1]. However, traditional evaluation primarily relies on qualitative data, and lacks in producing quantitative results that can easily be recorded and compared. Rather, varieties of grading scales are used to evaluate muscle function, coordination and control of the upper extremity. The patient is asked to perform various actions, and the therapist makes observations regarding the patient's ability. After assessing the patient, the therapist assigns a rating from the qualitative scale. However, these scales are subjective and tend to be poorly defined, allowing significant variations in muscle activity to go undetected. They rely highly on observations made by the therapist and patient; depend on the therapist's experience, and lack consistency and reliability [3]. An objective measure of motion analysis is needed to better manage progress and effectiveness of rehabilitation programs. Quantitative measurements would eliminate these factors that can influence and potentially bias a patient's results. Slight and minute variations in function would not go undetected, and this would ensure patient recovery is properly assessed.

#### **2.2 Current Qualitative Assessment**

Many devices and systems have been developed that quantitatively assess pinch and grip strength [1][2][6], and evaluate the range of motion and coordination of the fingers [1][7]. Range of motion is primary assessed using a goniometry. This device simply evaluates a patient's range of motion through measuring the angle extent which

they can rotate or move their limb. Hand dynamometers are a strain gauge type device that measure grip strength as a person squeezes the device. However, the current devices being used to assess a patient's motor function in the upper limb fail to pair functional assessment for both force and range of motion together. Further, these devices simply provide a measurement that must be recorded by an observer. They lack any kind of interface, such as a computer or memory storage hardware that would allow the patient to store the results each time the device is use and provide feedback and comparison with previous results. The devices currently being used also fail to incorporate any aesthetic appeal or motivational response during patient evaluation. This can be particularly unsatisfying for children that require functional assessment. The current methods of paediatric assessment are merely scaled versions of adult devices, and lack the appeal and creativity to intrigue and motivate children. A device that integrates the quantitative assessment of both range of motion and force of the upper limb, along with a computer interface and user feedback, and incorporates an aesthetic and motivation appeal would greatly improve the accuracy and reliability of functional assessment in upper extremity rehabilitation, particularly for paediatric patients.

## **Chapter 3**

### **Statement of the Problem**

#### **3.1 Project Objective**

The objective of this project was to design an instrumentation system to quantify muscle function in the upper extremity which would allow clinicians to quantitatively measure and objectively evaluate functional recovery in this limb. Working closely with the School of Rehabilitation Sciences and the MacHand Therapy Program at McMaster, a device was developed that combines a biomedical engineering approach to the current clinical therapy assessment techniques being used. The system was particularly aimed toward paediatric patients rehabilitating from trauma resulting in peripheral nerves. To assess the muscle control and range of motion, the hand held instrument measures a patient's ability to pronate and supinate their forearm by measuring the angle which they have rotated their wrist and the speed of rotation.

#### **3.2 Methodology of Solution**

In order to evaluate a patient's movement capabilities, a 3D accelerometer was used to acquire motion information. Using its three axes, the accelerometer can measure both tilting and dynamic acceleration, and these axial measurements are used to assess the movement. When the patient is asked to pronate or supinate their forearm, the accelerometer readings are processed to determine how much it was tilted from its initial position along one of its axes. The instrumentation system contains the accelerometer, along with other analog hardware necessary to acquire and transmit the data. The analog signal output by the accelerometer is delivered to a microcontroller to convert it to a digital signal. Using RF wireless communication, the data is then transmitted from the instrumented device to a base station a few feet away. Wireless transmission was necessary to ensure patient safety and complete isolation from a power supply and ground. The instrumented device is battery powered to make it wireless and to reduce the amount of noise in the signal transmitted. Once the data is received at the base station, it

is sent to analysis software on a computer through a USB connection. The data acquisition software was written in LabVIEW. A virtual instrumentation system was developed to process the received acceleration data and calculate the patient's angle and speed of rotation. After the appropriate calculations are made, the information is output on an interactive display to provide understandable, useful information about the patient's forearm movement.

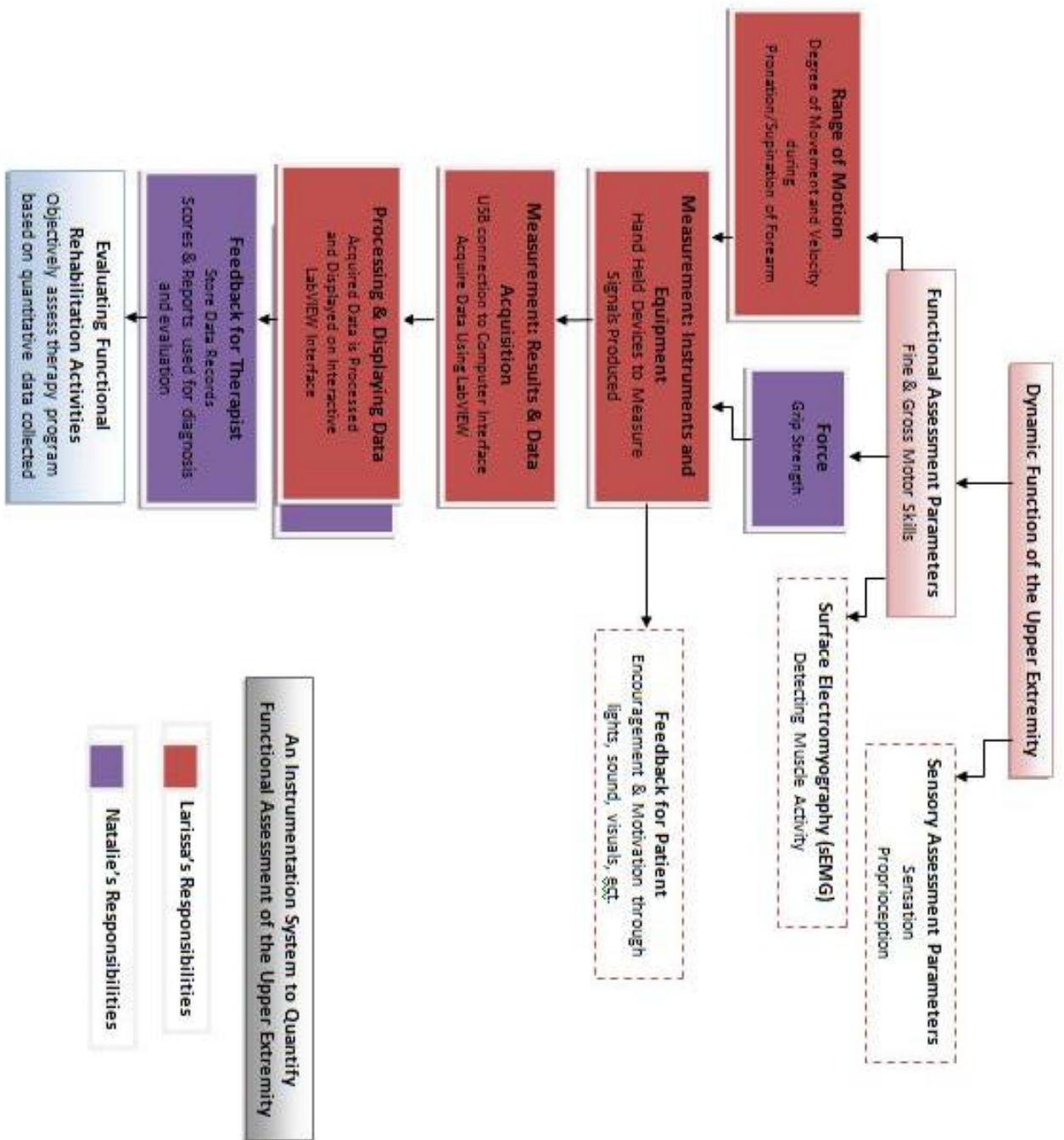


Figure 1 Methodology of Instrumentation System



# Chapter 4

## Experimental Procedures

### 4.1 Theoretical Design

The following is an outline of the theoretical design of the instrumentation system to and the experimental procedures used to implement the system (figure 2).

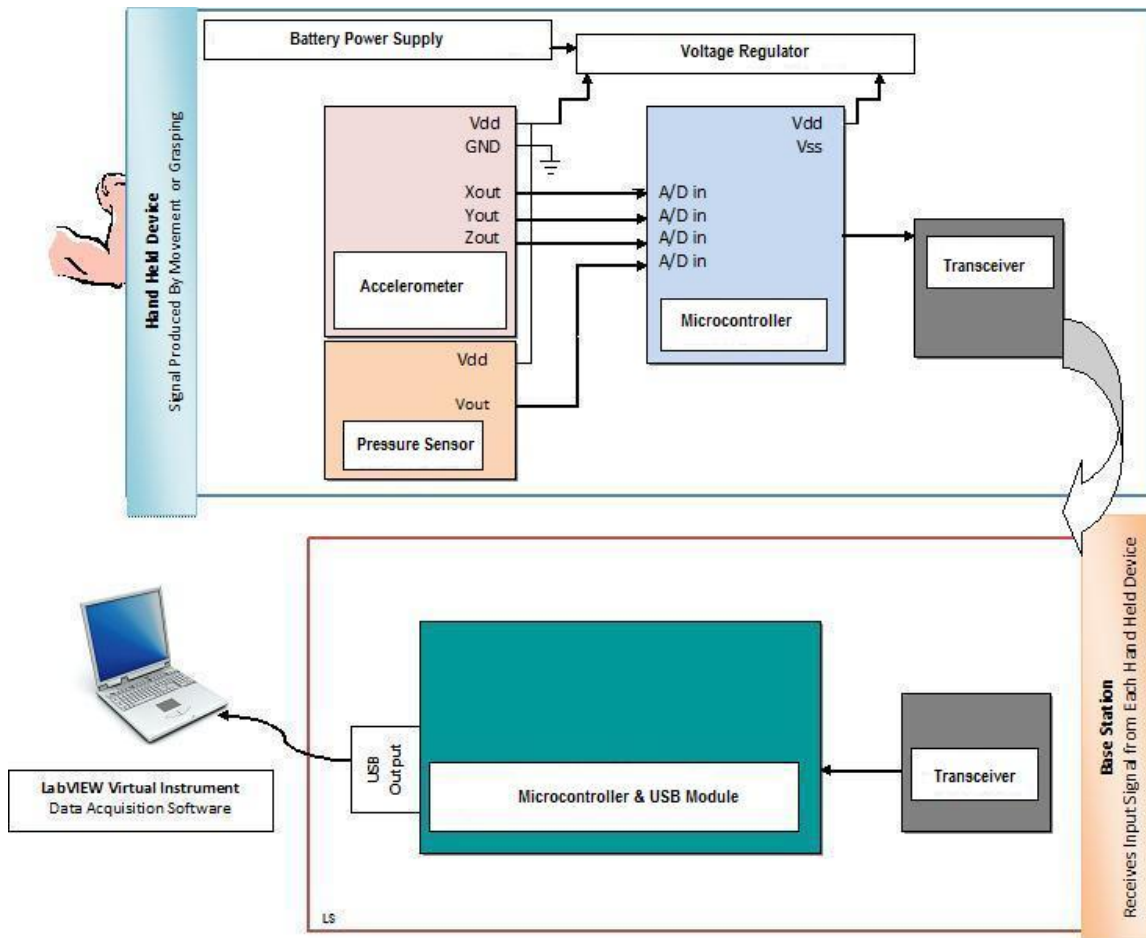


Figure 2 Theoretical design of instrumentation system

### 4.2 Design of Instrumented Hand Held Toy

The hand held toy is instrumented with the necessary analog hardware to capture a patient's range of motion, and transmit the data wirelessly to a computer. Specifications considered when choosing hardware components to instrument the toy included sensitivity, power consumption, foot print, weight, and cost. Because the patients have

limited mobility in their limb, the components needed to be light, yet provide a high degree of sensitivity to capture even minute motions. The components also needed to have a small footprint, in order to easily instrument a child's toy. Since the toy is wireless, it must be battery powered and the power consumption of the hardware in the toy had to be kept to a minimum. Lastly, costs were considered to ensure economically feasible decisions were made. Hardware components in the hand held toy include an accelerometer, a microcontroller, an RF transceiver, battery and battery holder. (The toy also contains a pressure sensor, to measure force. This piece of hardware was not my component. However, sampling and transmitting the data acquired from it was my also responsibility).

The primary component contained in the toy is a three axis accelerometer, used to capture the patient's mobility. The accelerometer contained in the instrumented toy is the ADXL335 accelerometer from Analog Devices. This three axis, analog accelerometer requires little power, provides sufficient sensitivity to capture motions produced by the patient, and has a very small footprint. The accelerometer is on a premade breakout board and headers were soldered on in order to make the accelerometer bread board compatible for prototyping.

In order to transmit the data acquired from the accelerometer and pressure sensor wirelessly, it must first be converted to a digital signal. A microcontroller was used to sample the data. The accelerometer breakout board contains a passive filter on the output with a cut off frequency of 50Hz. Therefore, according to Nyquist Sampling Theorem, the accelerometer signal from each axis must be sampled at minimum rate of 100Hz. The microcontroller was programmed to sample each axis of the accelerometer (as well as the signal from the pressure sensor) at approximately 1kHz, format the data, and output it to the RF transceiver. The microcontroller chosen was the ATMEGA328 with pre-programmed Arduino bootloader. This microcontroller also has small foot print, fairly low power consumption, and has sufficient memory and A/D channels required for its purpose. The only external hardware component needed for the microcontroller is a 16MHz crystal, to provide the clock input. This microcontroller was chosen specifically because it can easily be programmed using the Arduino Duemilanove USB module (which was used in the base station). To format and send the data to the RF transceiver,

an Arduino library specific for the transceiver chosen was used. This library provides functions to initialize and configure the transceiver to transmit data across the wireless link. Initialization includes activating the transceiver, selecting the appropriate wireless channel, setting the number of bits to be transmitted, and configuring the transceiver for transmit mode. Once the microcontroller samples all five inputs using five ADC channels, and formats the data, it is sent to a transceiver. The data channels are sampled continuously every millisecond.

To communicate the data wirelessly, RF transceivers were used. The transmitting receiver is contained in the hand held toy. The SFE Micro-Miniature 2.4 GHz transceiver was chosen. This transceiver is on a break out board that contains the Nordic Semiconductor nRF2401 transceiver, and an ultra compact chip antenna. This chip was selected for wireless communication because it has an extremely small footprint, requires a 3.3V supply (which is the same as the same as the other components in the hand held toy), and can be easily interfaced with the microcontroller selected to implement an RF protocol. Headers were soldered onto the breakout board so it could be used. As data is sent from the microcontroller to the transceiver, it is transmitted across the wireless channel to the receiving transmitter.

To power the components contained in the instrumented toy, a battery power supply was used. To maintain a constant 3.3V supply, two lithium coin cell batteries were used. The two batteries are held in a dual-cell coin cell battery holder. Each battery supplies 3V, and the total 6V produced is scaled to 3.3V using an NTE940 three terminal positive voltage regulator. This regulator required an external lithium ion capacitor in order for the component to work properly. The regulator has an internal, over-current shut down logic. Without the capacitor, upon applying power to the regulator, the power surge causes the regulator to shut off and it will not supply a voltage output. However, the capacitor limits this initial surge of power to the regulator, and allows for proper functioning of the component.

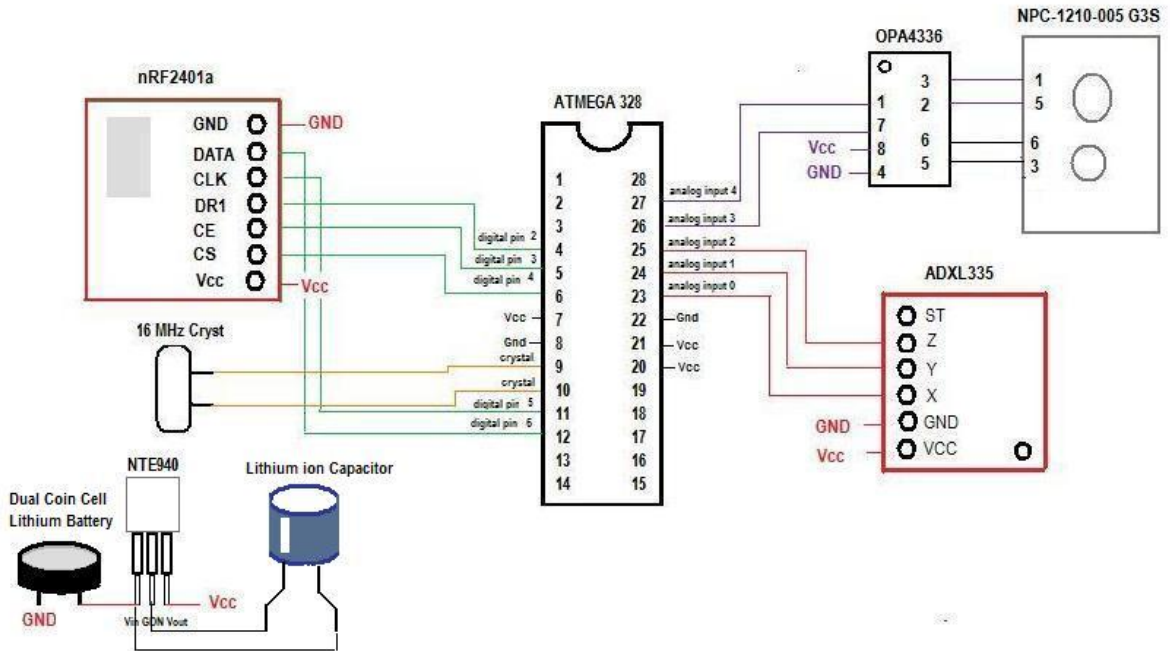


Figure 3 Hardware design for handheld device

### 4.3 Design of Base Station

The data transmitted from the handheld toy is received at the base station connected to a computer. The information is picked up by a transceiver identical to the transceiver in the hand held toy sending the data. This transceiver is also powered by a 3.3V supply. The transceiver is connected to a microcontroller and the received data is read by the microcontroller when it is available. However, this microcontroller has a 5V supply. To prevent damage to the transceiver, which can handle of maximum input voltage of 3.6V, any 5V output from the microcontroller must be translated to Vcc of the transceiver (3.3V). Three of the transceiver pins, Chip Select, Chip Enable, and CLK are connected to digital output pins of the microcontroller. To scale the 5V output from the microcontroller to 3.3V, a simple voltage divider is used for these three pins. All other pins on the transceiver are digital input pins for the microcontroller. The microcontroller has a minimum logic high input of  $0.6V_{cc}$ , which in this case is  $0.6 \times 5V = 3V$ . Therefore, any outputs from the transceiver to the microcontroller will be at a sufficient level ( $\approx 3.3V$ ) to be read correctly as logic high by the microcontroller.

The microcontroller in the base station is part of an Arduino Duemilanove USB module. This module is powered through USB, and provides the necessary hardware to

establish the USB-Serial port communication with the computer. This includes the FTDI chip, which emulates a serial port to the computer, so data can be sent and read using serial communication. The chip converts the serial data from the microcontroller to USB data packets using the RS-232 standard (voltage standard necessarily to communicate through a USB). The microcontrollers were easily programmed through the USB connection on the Duemilanove. This module also provides a 3.3V supply pin, which was used to power the transceiver. The microcontroller on the base station was programmed to read any available data from the transceiver and serially communicate it to the computer.

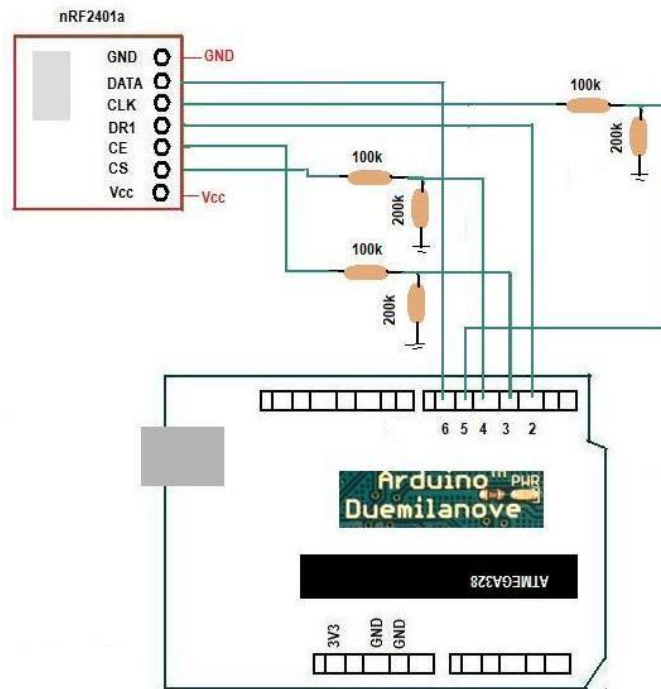


Figure 4 Hardware Design for Base Station

#### 4.4 Design of Software Data Acquisition Program

The software program used to process and display the data acquired from the accelerometer was written in LabVIEW. Once the data is communicated to the computer serially through USB connection, the LabVIEW program converts process the data to display the angle of tilt and angular velocity of rotation graphically and numerically as it acquired. To

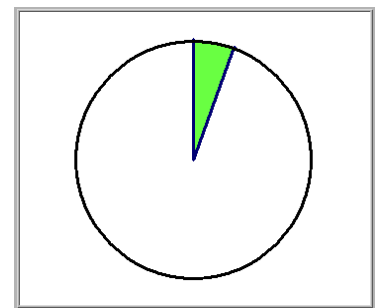


Figure 5 Front panel display of rotation angle

display the extent of rotation numerically, the acquired digital number is converted back to the original analog voltage value. The acceleration along each axis, X, Y and Z is determined from the analog voltages acquired from each axis. From these values, the number of degrees rotated from the z axis and the speed of rotation is calculated. This information is then displayed on the front panel of the LabVIEW program. To display the extent of rotation visually, a circle is displayed on the front panel (figure 3). As the accelerometer is tilted, a section of the circle is filled in (a pie piece) corresponding to the number of degrees rotated. Numeric indicators on the front panel also display the maximum and minimum angles that have been rotated to away from the zero degree axis established at the start, the total degrees that have been rotated through. Then angular velocity of rotation is displayed on a graph on the front panel, along with a numeric indicator that shows the current angular velocity. Once the session is complete, additional numeric indicators show the maximum and average speed of rotation.

Upon starting the program and selecting to measure range of motion, the user can select to either test the program or record/look up patient data. If the user selects to test the program, data is acquired and displayed. However, information is not saved. This feature allows the user to test the program and prepare for a recorded session. If the user selects to start a patient session, they can either display past data files or record new data. If the user selects to display past data files, they are prompted to select the files for rotation and/or angular files for up to two previous sessions. Rotational data is displayed graphically, and key numbers are also displayed. Angular velocity data is only displayed using key numeric values from the previous session(s).

If the user selects to record new patient data, they can choose to display the results from a previous session to provide motivation and comparison for the current session. Next the user is prompted to create new text files to save the data that will be acquired from this session – one for rotation data, and one for angular velocity data. Organizing the folders which these files are saved to identify the various patients falls under my partner's responsibility for the project. Once the files have been selected, the user can start recording the current session data. As the data is acquired, it is saved to the selected text files, and displayed on the front panel. If the user has selected to display past session information, both are visible for easy comparison.

Writing LabVIEW code to organize the files and folders for saving patient information did not fall under my responsibilities. However, the LabVIEW code which I wrote prompted the user to select these organized files for viewing of past history of data relating to range of motion, or create new files for recording new range of motion data (as previously described). This subVI was added to the overall software program for the instrumentation system, with only slight modifications made for file saving.

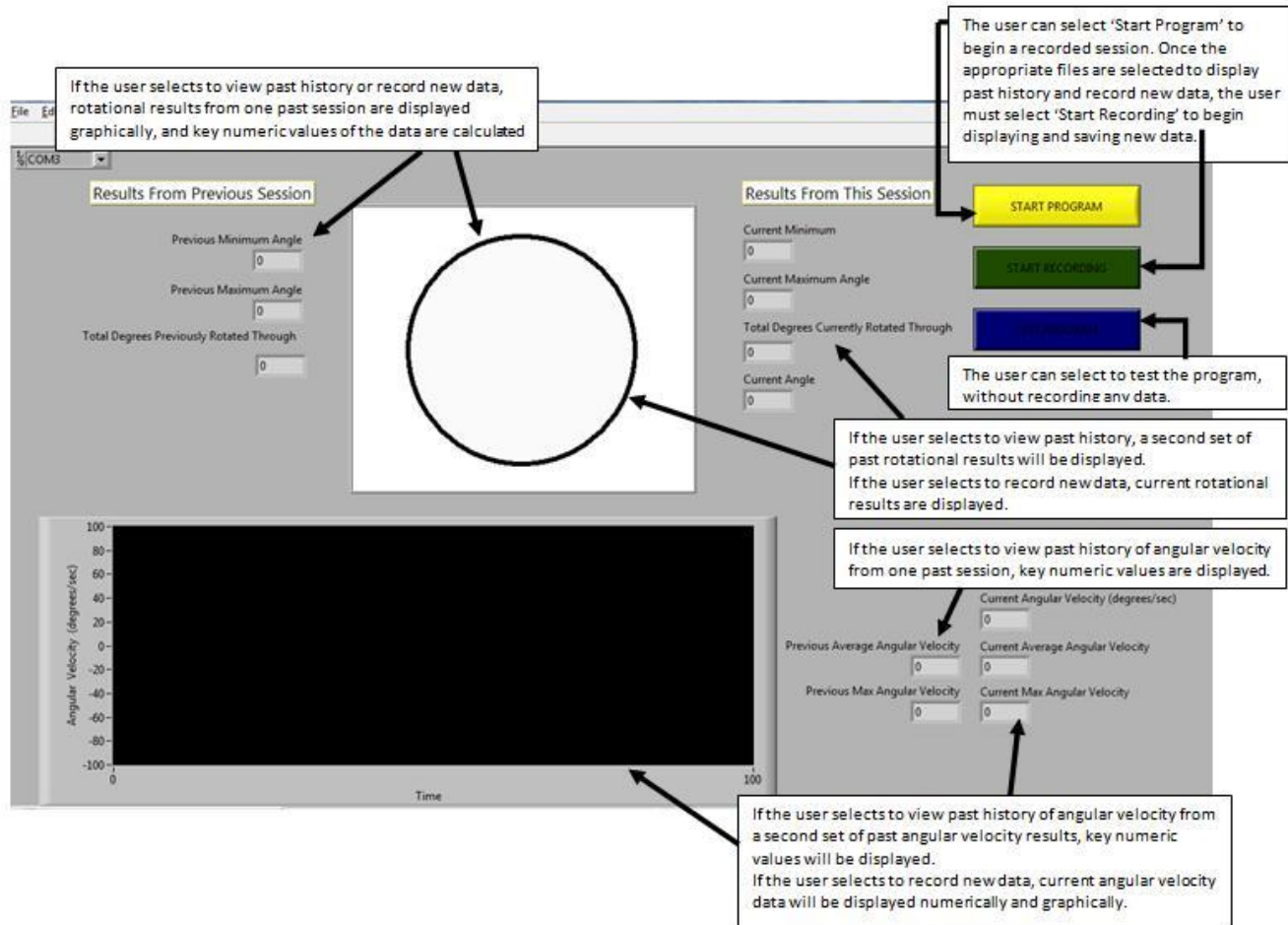


Figure 6 LabVIEW front panel upon selecting to evaluate patient's range of motion

## Chapter 5

### Results

#### 5.1 Instrumentation Hardware Results

The theoretical design of the hardware portion of the instrumentation system was implemented with the specified hardware components. The hardware components on the hand held toy were powered by a battery and the voltage regulator maintained a steady output of  $V_{cc} = 3.31$  volts. The three axes of the accelerometer and two outputs of the pressure sensor were successfully sampled by the microcontroller and transmitted wirelessly to the base station by the transceiver. The data was received by the transceiver on the base, and serially communicated to the computer. The validity of the data being sent was confirmed by rotating the accelerometer to a particular angle and comparing the measured results. The final circuits are shown (figure 7, figure 8 & figure 9).

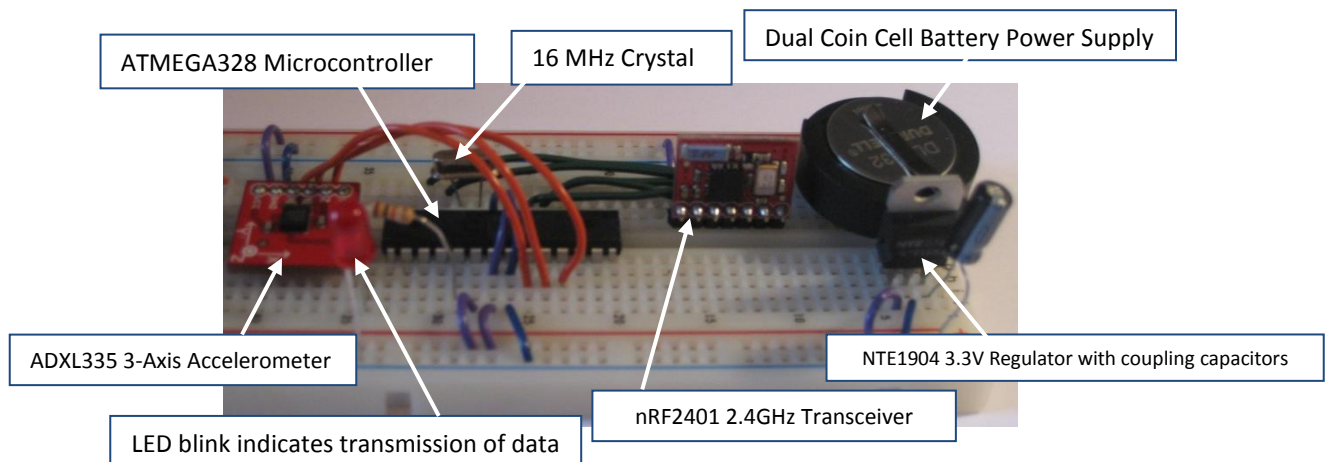


Figure 7 Final electronic circuit to instrument hand held device to measure range of motion



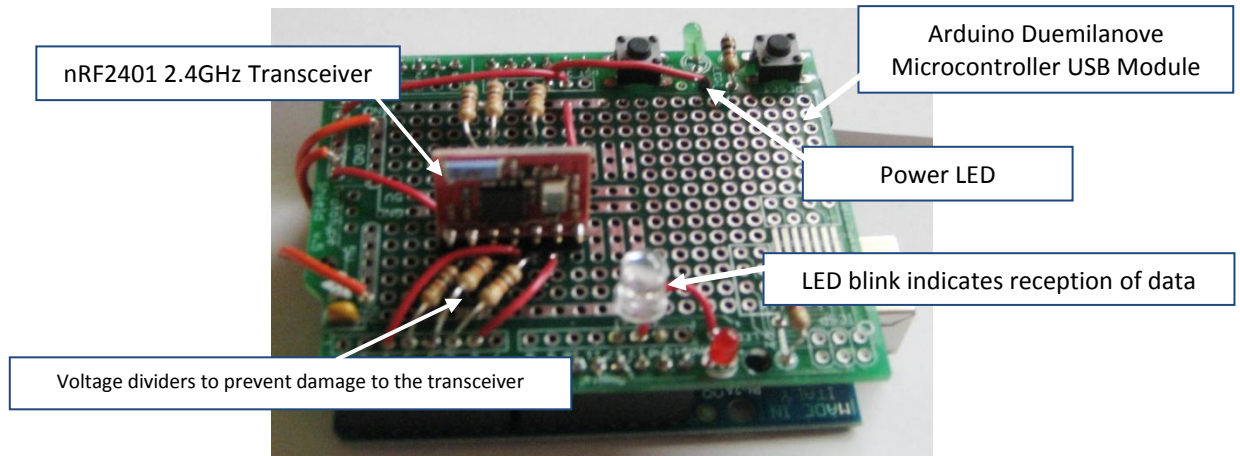


Figure 8 Final electronic circuit for base station

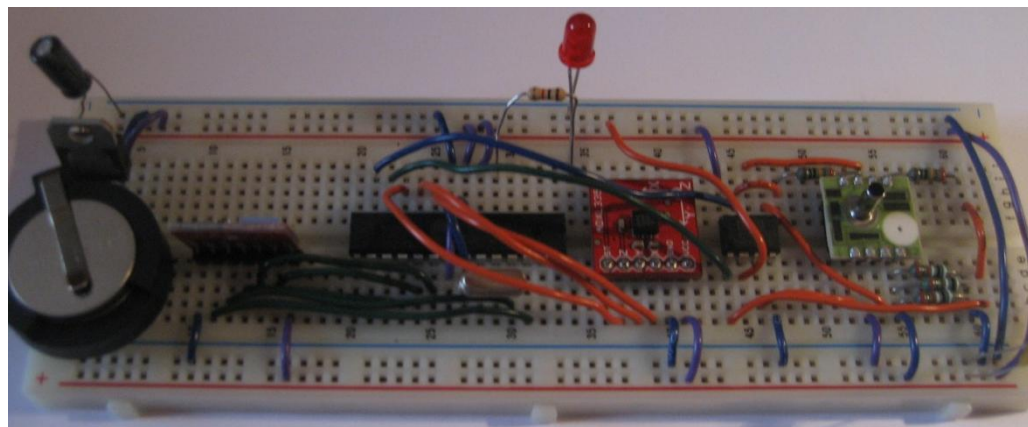


Figure 9 Final electronic circuit to instrument hand held device including hardware to measure both range of motion and pressure.

## 5.2 Accelerometer Calibration Results

The accelerometer was calibrated and the following results were obtained:

Table 1 Accelerometer Calibration Results

Axis	$V_{og}$ (V)	$V_{+1g}$ (V)	$V_{-1g}$ (V)	Sensitivity (V/g)
<b>X</b>	1.7132	2.0610	1.3646	0.3482
<b>Y</b>	1.6773	2.0369	1.3411	0.3479
<b>Z</b>	1.6849	2.0447	1.3631	0.3408

To test the accuracy of the accelerometer tilt measurements, the accelerometer was tilted to various angles in the clock wise direction. The data was read in through LabVIEW and was stored in a text file. For post processing, the data from each tilt was read into Matlab, and the average angle was found. The results obtained are as follows:

Table 2 Accelerometer tilt testing

Theoretical Angle	Experimental Angle
0°	0.2000
10°	9.2500
20°	19.4800
30°	30.2000
40°	40.7000
50°	47.9500
60°	62.6500
70°	69.4500
80°	80.3500
90°	88.9500
100°	100.2500
110°	109.5700
120°	120.3700
130°	131.0800
140°	142.0500
150°	150.0000
160°	158.7500
170°	165.8900
180°	175.5667

Table 3 Statistical data for accelerometer tilt testing

	Mean	Standard Deviation	Variance
<b>Tilt Error</b>	1.2102	1.2951	1.6773

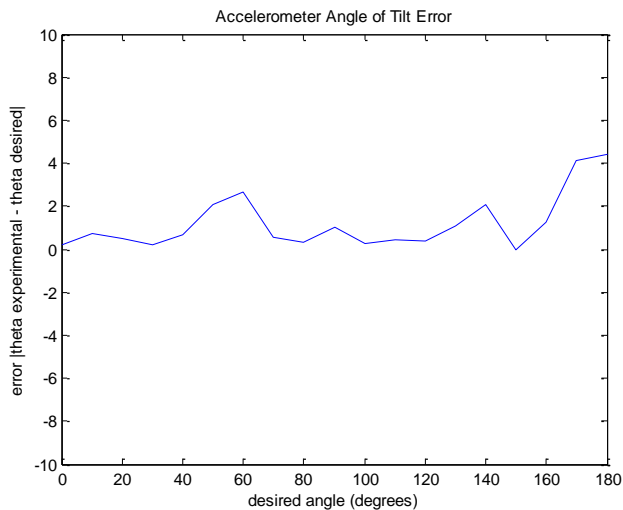


Figure 10 Error in experimental results measuring tilt with accelerometer

The measured experimental angles are close to the desired theoretical angles. The statistical measures of spread for the error data (mean, standard deviation and variance) indicate the measured angle is usually within one to two degrees of accuracy of the desired angle. This sensitivity is acceptable for the intended application. It can also be observed from the experimental data that, with a few exceptions, the error tends to increase as the angle of tilt increased. This slight non linearity of the accelerometer can be corrected using further software processing, such as implementing a look up table. The biggest error occurs when the accelerometer is rotated to measure angles between  $170^{\circ}$  and  $180^{\circ}$ . Similar results were obtained when the accelerometer was rotated in the counter clock wise direction. However, the angles measured were negative.

### **5.3 LabVIEW Program Results**

To test the LabVIEW subVI ‘Measuring Rotation’ and subsequently all the subVI’s it contains, all possible cases were tested. All results are accurate and the is processed and displayed as expected.

#### **5.3.1 Displaying Past History Files**

Upon selecting the ‘Start Program’ button on the front panel, the user is prompted to select between recording new data and viewing past history data. If the user selects to display past history files, files can be selected and the stored data is displayed on the front panel. If two sets of rotational and angular velocity data are selected to be displayed, the appropriate numeric indicators are filled in and the rotational results are also displayed graphically (figure #). If the user selects only to display only one set of results (both rotational and angular velocity data), only rotational data, or only angular velocity data, or any other combination of rotational and angular velocity results, the necessary values are displayed on the front panel (figure #).

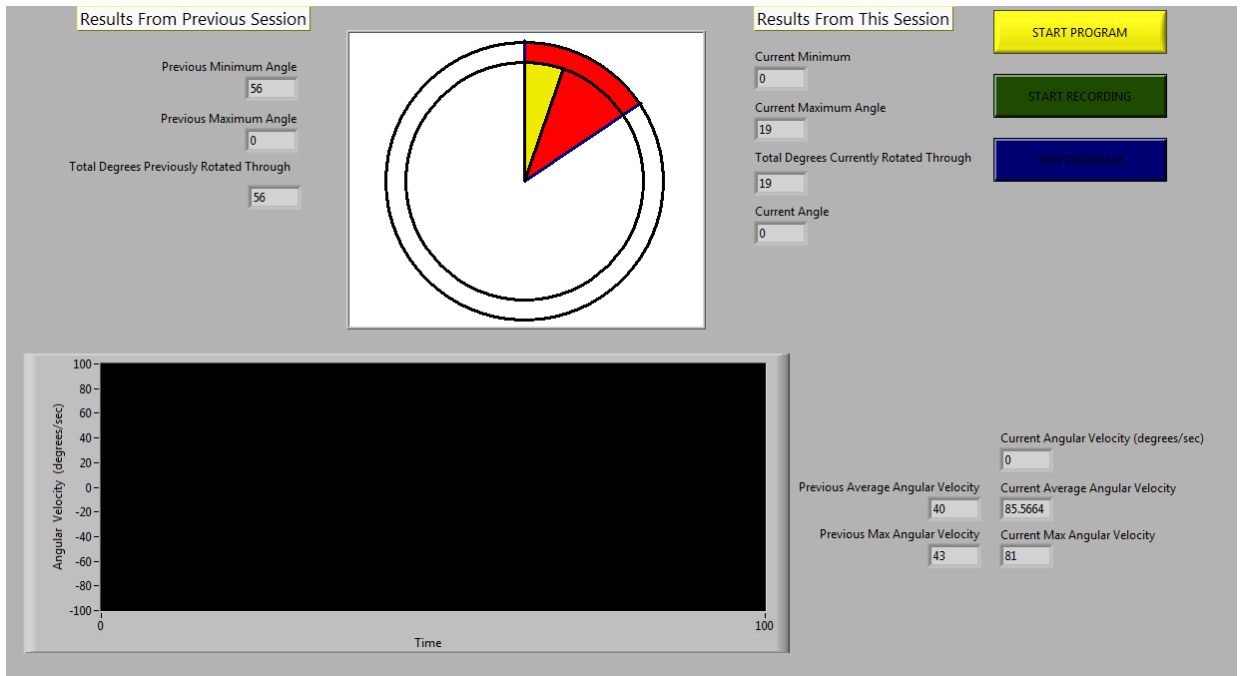


Figure 11 Displaying two sets of previous history data. Two sets of rotation and angular velocity results were selected to be displayed.

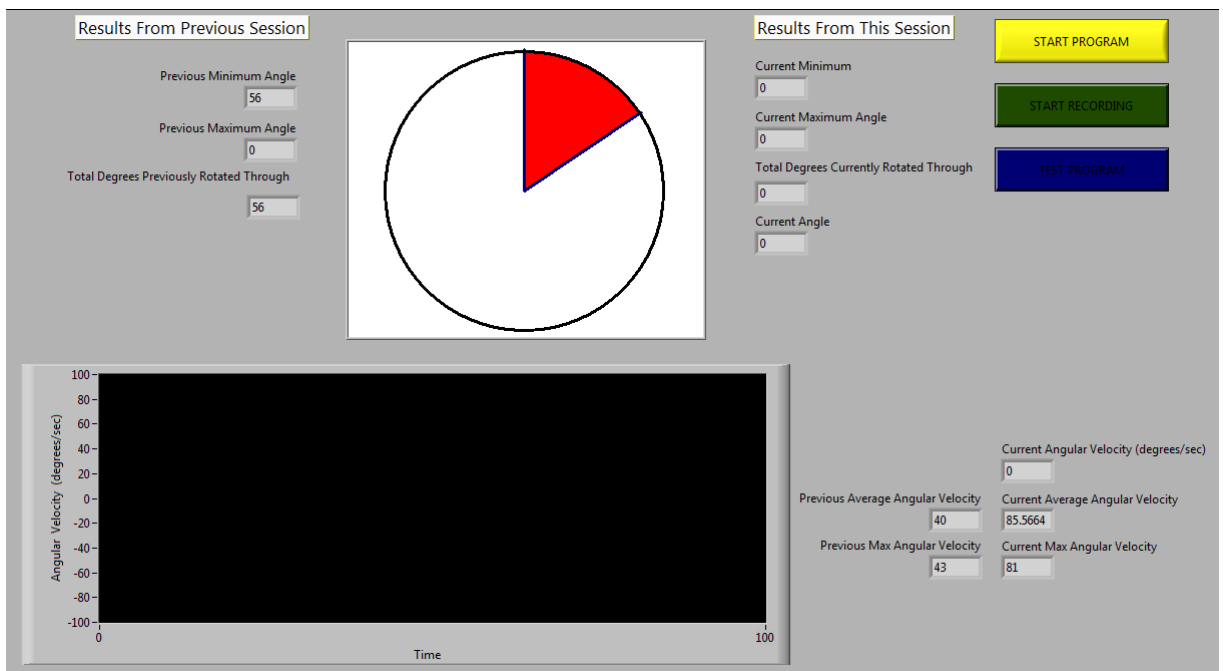


Figure 12 Displaying previous data history. One set of rotation results and two sets of angular velocity results were selected to be displayed.

### 5.3.2 Displaying Current Results Being Recorded

Upon selecting the 'Start Program' button, and selecting the record new data, the user can first select to display previously saved rotational and/or angular velocity results.

The previously selected files are displayed for the user to easily compare current results to a previous session, to provide motivation and track patient progression. Once the user selects the ‘Start Recording’, the data is obtained from the hand held toy, transmitted to the base station and sent through USB communication to the computer to be displayed on the front panel (figure #). The user can rotate the accelerometer in either direction (rotating in the clock wise direction will display a positive angle and rotating in the counter clock wise direction will display a negative angle).

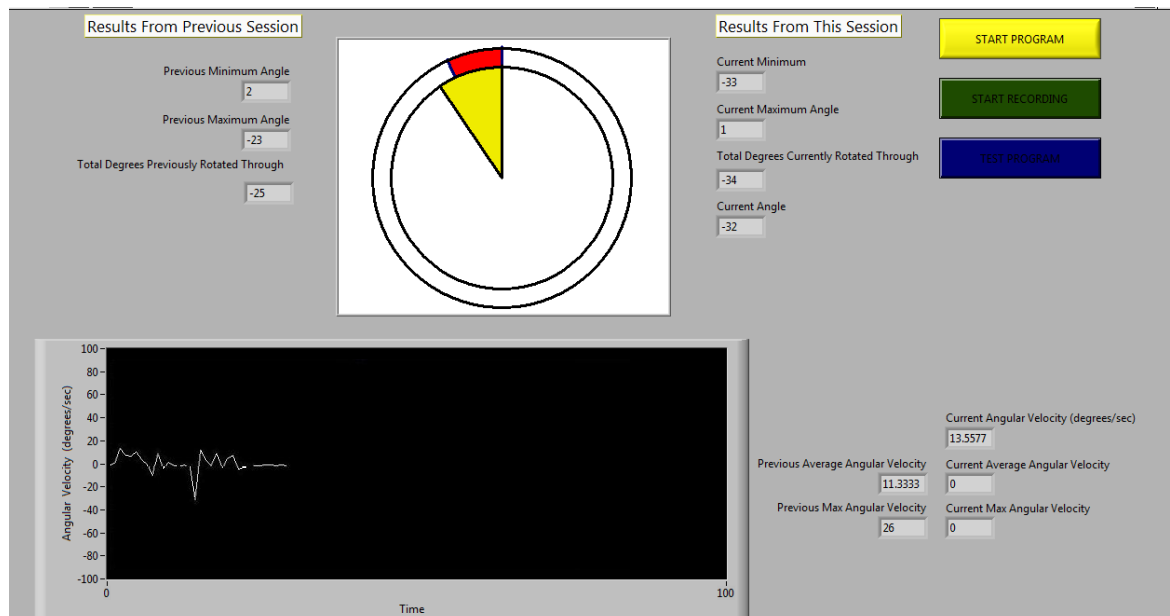


Figure 13 Displaying real time results. The user has selected to display a past set of rotation and angular velocity data. The current results are displayed as they are measured.

### 5.3.3 Displaying Testing Results

If the user selects to test the instrumentation system before recording any results, the ‘Test Program’ button must be selected. Upon the push of this button, the data is displayed on the front panel as it is received from the transducers in the toy without saving the results to a text file. The user is not give the option of displaying any previous results or selecting a text file to save the current results to. The data is simply displayed to allow the user to become familiar with the way the data is acquired from the transducer and displayed.

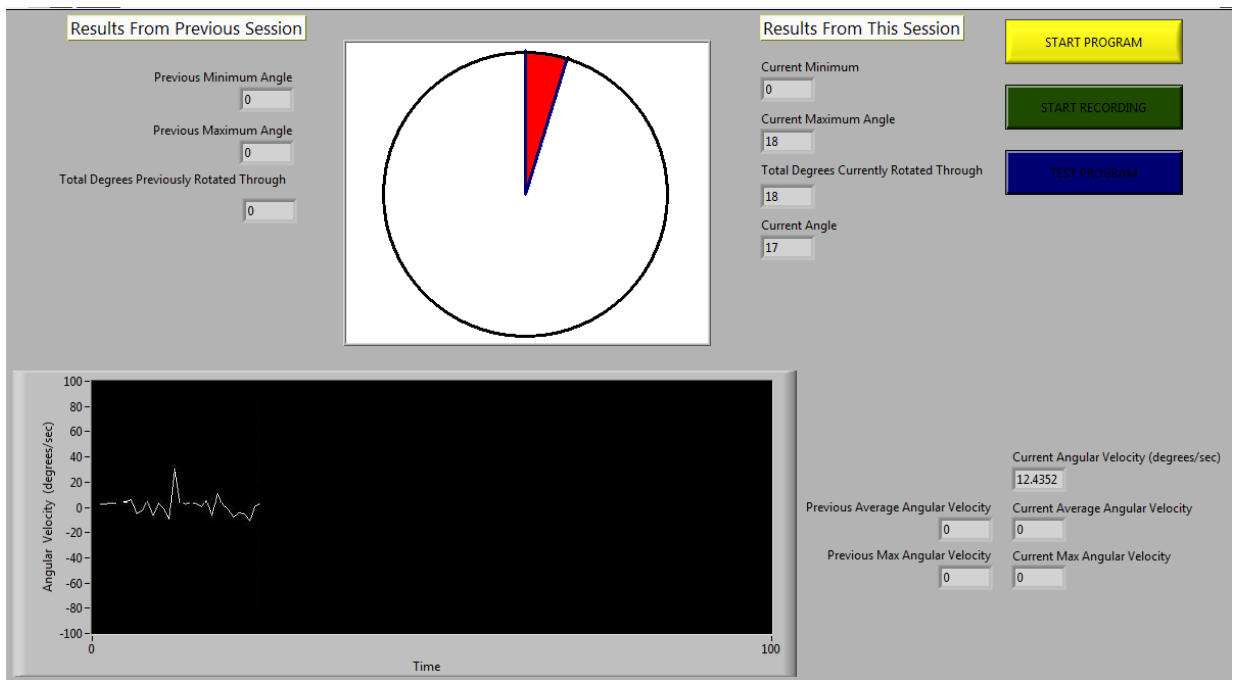


Figure 14 Displaying results as the user tests the program. Data is not recorded and previous results are not displayed. Only the current rotational and angular velocity numeric and graphical results are displayed.

## **Chapter 6**

### **Conclusion**

#### **6.1 Discussion**

An instrumentation to quantitatively measure the function of the upper limb can be beneficial in assessing muscle control and coordination in patients rehabilitating from peripheral nerve injuries. Such an instrumentation system particularly designed for paediatric patients was successfully designed and developed. This system provides the ability to quantitatively assess a patient's range of movement, and record one's improvements as they progress through their rehabilitation program. Through the use of accelerometry, the wireless device developed accurately assesses a person's ability to pronate and supinate their forearm and the speed which rotation is performed. The data is obtained from the accelerometer contained in a hand held toy, transmitted wirelessly to a base station, and communicated to a computer through a USB connection. The precise measurements of tilt and speed are calculated and graphically displayed on a computer using a software program written in LabVIEW. The software written allows for easy viewing and assessment of data, and allows the user to record the information for use in the future. This system effectively allows clinicians and therapists to assess a patient's functional recovery of the upper limb during a rehabilitation program and evaluate the effectiveness of the implemented therapy. All primary design objectives and deliverables were met, and the instrumentation system demonstrates feasibility for being used effectively for its intended purpose.

#### **6.2 Future Recommendations**

The instrumentation system was successfully developed and contained the basic components necessary to read, transmit and display the data measured. Although it is a good initial prototype, there are opportunities to improve and expand upon the device. Firstly, the accelerometer currently in the device measures +/- 3g's of acceleration. This rate of acceleration is much faster than the movement of the upper limb can achieve, especially for a child with peripheral nerve injuries. Using an accelerometer with such a

large range of measurement reduces the sensitivity of the measurements taken. Using an accelerometer with a smaller measurable range would improve the sensitivity of the device and allow for more precise measurements. Secondly, the device simply measures the angle and speed of rotation during pronation and supination of the forearm. However, the instrumentation can further be expanded by assessing other motions of the upper limb such as flexion/extension and internal/external rotation of the arm. To expand the device to incorporate the tracking of three dimensional positioning and speed, it may be beneficial to pair the accelerometer with another transducer such as a gyroscope. Thirdly, the device currently uses the Arduino Duemilanove USB Module to transmit received signals from the hand held toy to the computer. Although implementation using this module was straightforward and it allowed for easy programming of the microcontrollers used, problems have arisen during use. At times, the Arduino does not synchronize with the computer and the serial port is not validly read by the computer. In order to correct this problem, it was found that simply disconnecting and reconnecting does not help to synchronize the device with the computer and ensure it is recognized by the computer. LabVIEW must be closed and the Arduino must be disconnected and reconnected. This problem was found among other groups using the same device, and is likely an attribute of the Arduino module. To eliminate any synchronization problems, the Arduino module can be replaced with a self implemented microcontroller/USB interface using a microcontroller and an FTDI chip and other necessary hardware components such as voltage regulators and headers. Fourth, the microcontrollers used in the instrumentation system have Arduino boot loaders so they can easily be programmed using the Arduino module. As a result of this boot loader, the start up time of the device is longer than desired and data acquisition is delayed. When the serial port is opened through LabVIEW, the microcontroller reboots and runs its bootloader. It first checks to see if any attempts to reprogram the microcontroller are being made, and if not, will run the code programmed on it. If the computer tries to interrupt while the bootloader is running but does not upload a new program, the microcontroller will never exit the bootloader nor run its intended program. This also prevents the Arduino from synchronizing and effectively communicating with the computer. The microcontrollers do not require the boot loader and can be programmed using an external programmer yet still be used in the



Arduino USB module (if it is not eliminated). Avoiding the use of a boot loader will improve upon start up time, and could also improve the accuracy of the device. The first few samples taken from the accelerometer are used to calibrate the voltage offsets for each axes, and if the patient is unfamiliar with the time delay and the device is not in the appropriate position, inaccurate measurements could be made. Fifth, currently the hand held device uses a line voltage regulator to produce a constant 3.3v from the battery power supply. Line regulators tend to be inefficient and will drain the battery more quickly. To use the battery power more efficiently, a Charge Pump, a type of DC to DC Converter, should be used. This will prolong the life of the batteries in the hand held toy. Sixth, the current transceiver communication is unidirectional. Implementing bidirectional communication would improve the efficiency of the system. Upon power up of the hand held toy and the base station, bidirectional communication would also the two devices to synchronize and indicate to the user that the hand held toy was found and effectively communicating information to the base station. As the program is implemented right now, the toy simply transmits and it is assumed to be received by the base station immediately. Also, bidirectional communication would help increase power efficiency. Once the user selects to measure force or range of motion, the other transducer can be powered down to save on battery power. Lastly, to improve the wireless communication of the system, rather than using a USB connection and communication using RF transceivers, Blue Tooth technology can be used to communicate between the computer and the hand held device. This would improve the efficiency and speed of the system. Overall, basic improvements can be made to the system to increase the efficiency and accuracy of the device.

## **Appendix A**

## A.1 Hardware Components and Specifications

Table 4 Hardware Components and Specifications

Hardware Component	Ordered Part	Specifications
Accelerometer	Analog Devices ADXL335 (on break out board)	<ul style="list-style-type: none"> <li>• Analog, Low Powered 3 Axis Accelerometer</li> <li>• +/- 3g</li> <li>• Vcc = 3.3V</li> <li>• Bandwidth 50Hz</li> <li>• 3D motion and position capture</li> <li>• Tilt Sensing to measure rotation</li> </ul>
Base Station Microcontroller & USB Connection	Arduino Duemilanove USB module	<ul style="list-style-type: none"> <li>• Includes ATMEGA328 Microcontroller powered and programmed through USB connection</li> <li>• Includes USB jack and FTDI chip (required to convert serial to USB data) to establish USB-Serial Port Communication</li> <li>• Vcc = 5V</li> <li>• Logic High = 0.6Vcc = 3V or higher</li> <li>• Logic Low = 0.3Vcc = 1.5V or lower</li> <li>• 10 bit ADC, 8 A/D Channels, 23 I/O Pins</li> <li>• External 3.3V pin available</li> </ul>
Hand Held Device Microcontroller	ATMEGA328 with pre-programmed Arduino bootloader	<ul style="list-style-type: none"> <li>• Preprogrammed Arduino bootloader so it can be programmed using Arduino Duemilanove board</li> <li>• 10 bit ADC, 8 A/D Channels, 23 I/O Pins</li> <li>• Vcc = 3V</li> <li>• Logic High = 0.7Vcc = 2.31V or higher</li> <li>• Logic Low = 0.3Vcc = 0.99V or lower</li> </ul>
RF Transceivers	Nordic Semi Conductor nRF2401 High Speed Micro Transceiver with Chip Antenna (on break out board)	<ul style="list-style-type: none"> <li>• Vcc = 3.3V</li> <li>• GFSK Modulation 2.4 GHz</li> <li>• Low Output Power</li> <li>• Built in Chip Antenna</li> <li>• Compact Footprint</li> <li>• Up to 50 foot range transmission</li> </ul>
Power Supply (Hand Held Device)	Dual Cell Lithium Ion Coin Cell Battery	<ul style="list-style-type: none"> <li>• Total 6V output</li> <li>• Voltage regulator required for 3.3V supply</li> <li>• Small packaging</li> </ul>
3.3V Voltage Regulator	NTE 1904	<ul style="list-style-type: none"> <li>• Outputs a consistent voltage output of 3.3V and up to 1A of current.</li> </ul>
Crystal	16MHz Crystal	<ul style="list-style-type: none"> <li>• 16 MHz crystal to provide the clock signal on the microcontroller in the hand held toy</li> </ul>
Prototyping Shield	Adafruit Arduino Prototyping Shield	<ul style="list-style-type: none"> <li>• Prototyping shield made specifically for the Arduino Duemilanove</li> <li>• Brings all pins and buttons to surface of the prototyping board, and large space for soldering components.</li> </ul>

## A.2 Calibration of ADXL335 Accelerometer

### A.2.1 Accelerometer Parameters Required

In order to acquire accurate and high resolution measurements, the accelerometer parameters must be calibrated accurately. Each axis of the accelerometer has two parameters that need to be measured in order to convert analog voltage readings to acceleration reading correctly, namely Axis Offset and Axis Sensitivity. The ADXL335 Accelerometer outputs a voltage between 0 and  $V_{cc}$  volts for each axis, and zero acceleration for the axis occurs at approximately  $\frac{1}{2} V_{cc}$ . Therefore, each voltage output contains an offset that must be specifically calibrated for each axis. To convert a measured voltage reading to acceleration, Axis Sensitivity is also required, a conversion constant that must also be measured specifically for each axis. The approximate sensitivity of each axis depends on the type of accelerometer. The ADXL335 Accelerometer has an approximate Axis Sensitivity of  $330\text{mV/g}$ , where  $g$  is the acceleration due to gravity equal to  $9.81\text{m/s}^2$ .

### A.2.2 Calibration Measurements

To calibrate the ADXL335 accelerometer, the accelerometer was supplied by a voltage supply of  $V_{cc} = 3.3\text{V}$  on the Arduino Duemilanove Microcontroller Module, and the data was read in serially to the computer through USB. The microcontroller performing the analog to digital conversion has a  $5\text{V}$  supply, and a 10 bit A/D converter. Accelerometer calibration was complete with a voltage supply of  $V_{cc} = 3.3\text{ V (DC)}$ . To determine the sensitivity of each axis, the accelerometer was positioned in various ways such that outputs of  $-1g$  and  $+1g$  were captured for each axis. To determine the offset of each axis, the accelerometer was positioned to capture a  $0g$  output for each axis.

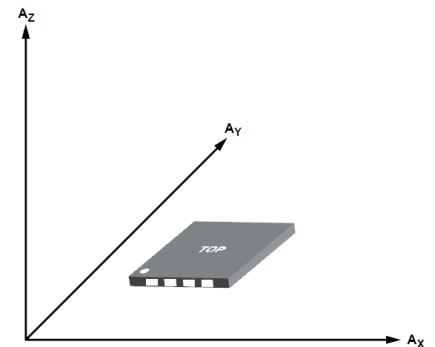


Figure 15 Axis of acceleration sensitivity of ADXL335

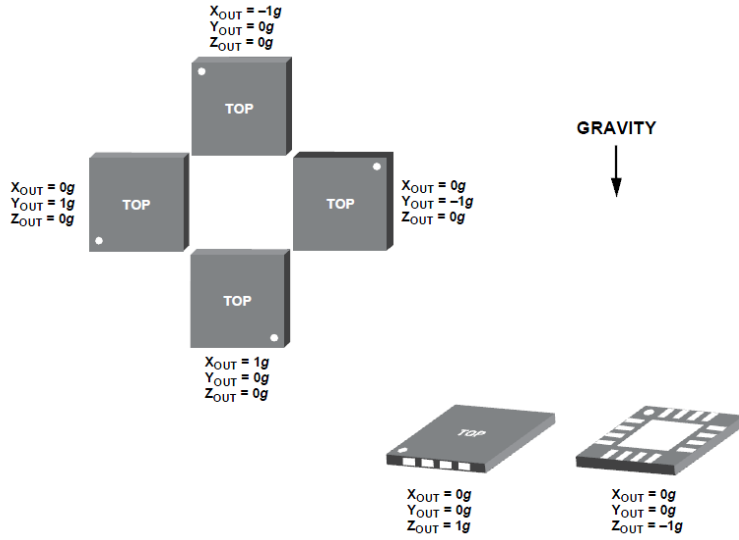


Figure 16 Output response of ADXL335 for a given orientation with respect to gravity.

Matlab was used to take 1000 samples of each accelerometer axis in each position, and the acquired data was averaged and calibration calculations using the following equations:

$$V_{in} = \frac{5}{2^{10}} (\text{Digital Input Value})$$

$$\text{Axis Sensitivity} = \frac{V_{+1g} - V_{-1g}}{2g} V/g$$

$$\text{Axis Offset} = V_{0g}$$

### A.2.3 Measuring Angle of Rotation using ADXL335 Accelerometer

To measure the angle of rotation of the ADXL335 about the z-axis, the maximum sensitivity is obtained by measuring tilt using all three axes of the accelerometer. Measuring tilt about the z-axis using all three axes of acceleration provides the greatest resolution, and the angle of tilt calculated is independent of the direction rotated from the z-axis. Using basic trigonometry to derive the equation (figure 9), rotation about the z-axis using calculated using the following equation:

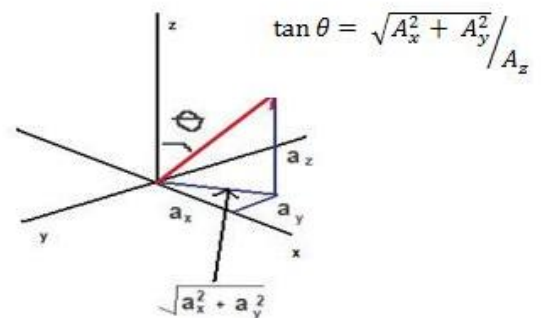


Figure 17 Calculating angle of rotation with z axis.

$$\theta = \tan^{-1} \left( \frac{\sqrt{A_x^2 + A_y^2}}{A_z} \right) \text{ degrees}$$

This equation will only provide 180 degrees of rotation ( $-90^\circ$  to  $+90^\circ$ ). However, to achieve a full 360 degrees of rotation ( $-180^\circ$  to  $+180^\circ$ ), the sign of the z-axis and y-axis accelerations must also be considered to properly display the data (figure 10).

The sign of the z-axis acceleration indicates if the angle is in the top half or bottom half of the Cartesian plane. If the sign of the z-axis is positive, the angle is in the top half of the Cartesian plane and the magnitude of the angle does not need to be modified. If the sign of the z-axis is negative, the angle is in the bottom half of the Cartesian plane. The magnitude of the angle must be modified by adding  $180^\circ$  to the calculated angle.

To determine the appropriate sign of the measured angle, the sign of the y-axis acceleration is used to determine which quadrant the angle of rotation is in. When the accelerometer is rotated in the first quadrant, the calculated angle  $\theta$  is between  $0^\circ$  and  $90^\circ$ , and the y-axis acceleration is positive. The calculated angle  $\theta$  can be displayed directly. When the accelerometer is rotated in the second quadrant, the calculated angle is again between  $0^\circ$  and  $90^\circ$ . However, the y-axis acceleration is negative. The negative of the measured angle is displayed. When the accelerometer is rotated in the third quadrant, the calculated angle  $\theta$  is between  $-90^\circ$  and  $0^\circ$ , and the y-axis acceleration is negative. When these requirements are met,  $180^\circ$  must be added to the negative angle  $\theta$ , and the result must be negated in order to properly display an angle of tilt between  $-90^\circ$  and  $-180^\circ$ . When the accelerometer is rotated in the fourth quadrant, the calculated angle  $\theta$  is between  $-90^\circ$  and  $0^\circ$ , and the y axis acceleration is positive. When these requirements are met,  $180^\circ$  must be added to the calculated angle  $\theta$  to properly display an angle between  $90^\circ$  and  $180^\circ$ .

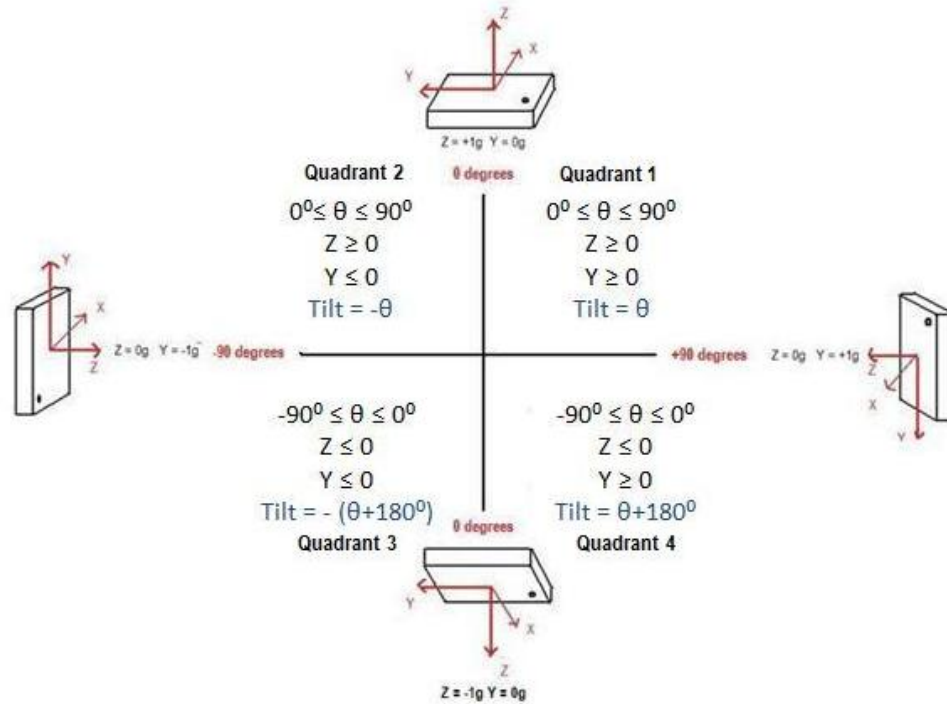


Figure 18 Sign of Y and Z accelerations used to determine the angle of tilt displayed

## A.2.4 Measuring Angular Velocity using ADXL335 Accelerometer

To measure angular velocity using the ADXL335, the change in angle per unit time was measured. The absolute angular velocity was calculated using the following formula:

$$\omega = \frac{|\theta_2 - \theta_1|}{\Delta t} \text{ degrees per sec}$$

Where  $\theta_2$  is the angle measured at  $t = t_2$ ,  $\theta_1$  is the angle measured at  $t = t_1$  and  $\Delta t = t_2 - t_1$ .

## **A.3 Wireless Communication**

### **A.3.1 RF Transceivers and Nrf2401 Arduino Library**

Using a pair of Nordic Semiconductor nRF2401a Micro Transceivers, RF communication was implemented to transmit the data from the hand held device to the base station. Microcontrollers were programmed to interface with each transceiver. Initially, to set up and test communication a simple program was written for three specific numbers to be sent from transmitter to receiver. After this program was successfully implemented and the wireless link could properly be established, a program was written to communicate the necessary accelerometer and pressure sensor data. The Nrf2401 Arduino library was used, developed specifically to configure the Nordic Semiconductor nRF2401 transceiver. For each of the microcontrollers, an Nrf2401 object was created. The following attributes and methods of the Nrf2401 object were used to configure the transceivers and send or receive data:

#### **Radio.remoteAddress**

This attribute of the Radio object sets the channel the transceiver will send data from (if in transmitting mode) or receive data from (if in receiver mode). The nRF2401 has 25 different channels, and to establish communication between a pair of transceivers the data must be sent across the same channel.

#### **Radio.data[]**

This attribute is an array of bytes that stores received or transmitted data. Up to 25 bytes can be stored.

#### **Radio.txMode(int messageSize)**

This method initializes and configures the transceiver to transmit data. The messageSize parameter specifies the number of bytes that will be transmitted.



**Radio.rxMode(int messageSize)**

This method initializes and configures the transceiver to receive data. The messageSize parameter specifies the number of bytes that will be received. It is important the message size to be transmitted equals the message size to be received.

**Radio.write()**

This method sends the data stored in Radio.data[] across the wireless channel to the address specified by Radio.RemoteAddress.

**Radio.available()**

This method returns true when data is available to be read and false when data is not available. The data channel is polled for incoming data using this method. When data is available, the DR1 pin will go high to indicate data is available.

**Radio.read()**

This method reads the data available from the appropriate channel and stores it in the Radio.Data[] array for the receiver.

**A.3.2 Transmitter Microcontroller**

The microcontroller contained in the handheld device was programmed to transmit data from the device to the base station. Five analog-to-digital converter channels are sampled every millisecond to digitize the analog signals output from the transducers in the device. The transceiver was configured as a transmitter, and programmed to send the ten bytes of data acquired to its receiver pair in the base station. The data read from each ADC channel must be stored in two bytes of data. However, the Data[] array is an array of bytes. Therefore, the data must be broken into bytes, transmitted, and then restored at the receiving end. For each transmission, an LED blinks to indicate data has been sent across the channel. The transmitter microcontroller was programmed with the following code:

```

//Created by Larissa Schudlo
//Transmitter Microcontroller Code
//1-way communication, sending 10 Bytes of data
//Send to address 1
//data sampled and sent every 1mS
////////////////////////////////////
// The circuit:      analog    0: x-axis
//                  analog    1: y-axis
//                  analog    2: z-axis
//                  analog    3: pressure sensor internal
//                  analog    4: pressure sensor ambient
//                  digital   2: DR1
//                  digital   3: CE
//                  digital   4: CS
//                  digital   5: CLK
//                  digital   6: DAT
//                  digital  13: blinking LED, to frame tx
////////////////////////////////////
#include "Nrf2401.h"           //include nRF2401
Transceiver Library
Nrf2401 Radio;                //declare the TXRX
object
const int xPin = 0;           // x-axis of the
accelerometer
const int yPin = 1;           // y-axis of the
accelerometer
const int zPin = 2;           // z-axis of the
accelerometer
const int pSensIpin = 3;      //internal pressure of the
pressure sensor
const int pSensApin = 4;      //ambient pressure of the
pressure sensor
int xAxis = 0;
int yAxis = 0;
int zAxis = 0;
int pSensI = 0;
int pSensA = 0;

void setup (void)
{
  Radio.remoteAddress = 1;    //address the data is sent to
  Radio.txMode(10);           //put the TXRX into Tx
mode, to send 10 bytes (2 per axis analog pin sampled)
  //digital pin 13 blinks twice on start up, accelerometer
requires 17ms power up
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);
}

```

```

delay(300);
digitalWrite(13, LOW);
delay(300);
pinMode(13, OUTPUT);
digitalWrite(13, HIGH);
delay(300);
digitalWrite(13, LOW);
    //disregard first few samples taken by ADC
    for (int i = 0; i < 3; i++)
    {
        xAxis = analogRead(xPin);
        yAxis = analogRead(yPin);
        zAxis = analogRead(zPin);
    }
}
void loop(void)
{
    //read the 3 axes of the accelerometer
    xAxis = analogRead(xPin);
    yAxis = analogRead(yPin);
    zAxis = analogRead(zPin);
    //read 2 outputs of the pressure sensor
    pSensI = analogRead(pSensIpin);
    pSensA = analogRead(pSensApin);
    //10 bytes of data to be sent to receiver, data must be
    broken up into bytes
    Radio.data[0] = highByte(xAxis);
    Radio.data[1] = lowByte(xAxis);
    Radio.data[2] = highByte(yAxis);
    Radio.data[3] = lowByte(yAxis);
    Radio.data[4] = highByte(zAxis);
    Radio.data[5] = lowByte(zAxis);
    Radio.data[6] = highByte(pSensI);
    Radio.data[7] = lowByte(pSensI);
    Radio.data[8] = highByte(pSensA);
    Radio.data[9] = lowByte(pSensA);

    digitalWrite(13, HIGH);                //frame the
    transmission with LED blink
    Radio.write();                          //writes the
    data in Radio.data to be sent across wireless link
    delayMicroseconds(1000);                //delay 1mS before sending
    again
    digitalWrite(13, LOW);                  //frame the
    transmission with LED blink
}

```

### A.3.3 Receiver Microcontroller

The microcontroller contained in the base station was programmed to receive data from the handheld device, and communicate it to the computer through serial port communication. Ten bytes of data are read by the receiver, and the high byte and low byte of each data value are put back together. The five values read from the transducers are sent to the computer through a serial port. In order to distinguish between the different accelerometer and pressure sensor inputs, the data is sent to the computer in the form: x###y###z###i###a###,. The LabVIEW data acquisition program parses the data, to ensure the correct value is associated with the correct transducer output. For each reception of data, an LED blinks to indicate data has been read from the channel. An additional LED blinks to indicate data when the data is sent to the computer. The receiver microcontroller was programmed with the following code:

```
//Created by Larissa Schudlo
//Receiver Microcontroller Code //1-way communication,
receiving 10 Bytes of data
//sent to address 1 //blink LED to frame rx, blink
second LED to frame serial communication
//Baud Rate = 9600 for serial communication with computer
//Data sent to computer in the form: x###y###z###i###a###
////////////////////////////////////
////
// The circuit: digital 2: DR1
// digital 3: CE
// digital 4: CS
// digital 5: CLK
// digital 6: DAT
// digital 13: blinking LED, to frame tx
// digital 12: blinking LED, to frame
serial communication
////////////////////////////////////
////
#include "Nrf2401.h" //include nRF2401 Transceiver
Library
Nrf2401 Radio; //declare the TXRX object
word x = 0; //x-axis acceleration
word y = 0; //y-axis acceleration
word z = 0; //z-axis acceleration
word pi = 0; //internal pressure
word pa = 0; //ambient pressure
```

```

void setup (void)
{
  Radio.localAddress = 1; //the address other radios will
  use to send data to this TXRX
  Radio.rxMode(10); //put into RX mode, to receive
  10 bytes
  Serial.begin(9600); //open serial port of computer,
  baud rate = 9600
  //digital pin 13 blinks twice on start up
  //if both transceivers power up at same time, rx will be
  ready before tx to get incoming data
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  digitalWrite(12, LOW);
  digitalWrite(13, HIGH);
  delay(10);
  digitalWrite(13, LOW);
  delay(10);
  digitalWrite(13, HIGH);
  delay(10);
  digitalWrite(13, LOW);
}
void loop(void)
{
  digitalWrite(13, HIGH); //frame RX with LED blink
  while(!Radio.available()); //wait for incoming data
  Radio.read(); //read the available
  data
  //data was sent byte by byte, pairs of bytes must be put
  back together
  x = word(Radio.data[0], Radio.data[1]); //x-
  axis acceleration
  y = word(Radio.data[2], Radio.data[3]); //y-axis
  acceleration
  z = word(Radio.data[4], Radio.data[5]); //z-axis
  acceleration
  pi = word(Radio.data[6], Radio.data[7]);
  //internal pressure
  pa = word(Radio.data[8], Radio.data[9]); //ambient
  pressure
  digitalWrite(13, LOW); //frame RX with LED blink

  digitalWrite(12, HIGH); //frame serial communication
  with LED blink
  //send received data to USB through serial communication
  Serial.print("x");
  Serial.print(int(x));
}

```

```
Serial.print("y");
Serial.print(int(y));
Serial.print("z");
Serial.print(int(z));
Serial.print("i");
Serial.print(int(pi));
Serial.print("a");
Serial.print(int(pa));
digitalWrite(12, LOW); //frame serial communication
with LED blink
}
```

## A.4 LabVIEW SubVIs

The following subVIs were written, and can be found in the main LabVIEW data acquisition program for the instrumentation system.

### A.4.1 USB 4 Channel Read

<div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto;"> <span style="color: red; font-weight: bold;">Usb Read</span> </div>	<b>Inputs:</b>	VISA Resource (COMM port for USB device)
	<b>Outputs:</b>	X-axis acceleration    Y-axis acceleration    Z-axis acceleration Internal Pressure    Ambient Pressure

The microcontroller on the base station is programmed to send data to the computer in the form: `x###y###z###i###a###` repeated continuously, where `x###` represent the data read from the x-axis of the accelerometer, `y###` represents the data from the y-axis of the accelerometer, `z###` represents the data from the z-axis of the accelerometer, `i###` represents the internal pressure data from the pressure sensor and `a###` represents the ambient pressure data from the pressure sensor. LabVIEW reads the data in as a string through the VISA Comm Port (LabVIEW's subVI to read USB data). In order to separate the different numbers, one byte is read in at a time until that byte contains an 'x.' Once an 'x' is read in, the next three bytes are read in, converted from a string to a double, and then passed out of the subVI as the value representing the x-axis acceleration. This is repeated for the y-axis, z-axis, and the pressure sensor data. Reading the data in this manner (ie – waiting for the correct string that represents the data that is to be passed out next) ensures the correct digital number is associated with the correct axis/pressure sensor, and full numbers are not read incorrectly.

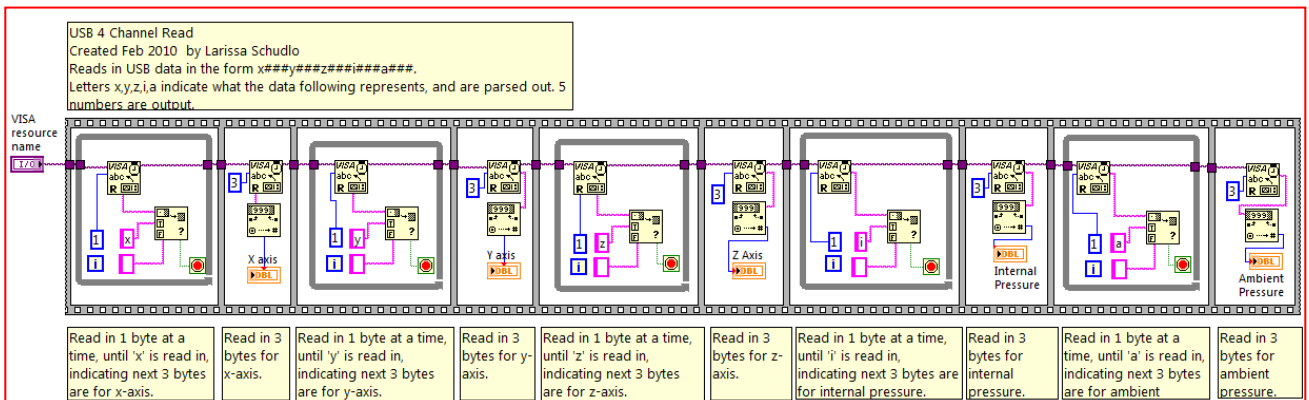


Figure 19 USB 4 Channel Read SubVI

## A.4.2 Micro to Acceleration Conversion



**Inputs :** Accelerometer Digital Input    Axis Voltage Offset    Axis Sensitivity  
**Outputs:** Acceleration

The microcontroller sends the digital number that was read in originally by the ADC on the handheld toy to the LabVIEW program. This data must be converted back to the original voltage, and then the acceleration can be calculated. The voltage supply on the handheld toy is  $V_{cc} = 3.3V$ , and the ADC has a 10 bit resolution (1023 different numbers can be represented). Therefore, to back convert to the original voltage read from the accelerometer:

$$V_{axis} = Digital\ Input * \frac{3.3V}{1024}$$

The voltage must then be converted to the acceleration of that axis, based on pre-calibrated axis sensitivity and offset voltages specific to that axis:

$$Acceleration = \frac{V_{axis} - V_{offset}}{sensitivity}$$

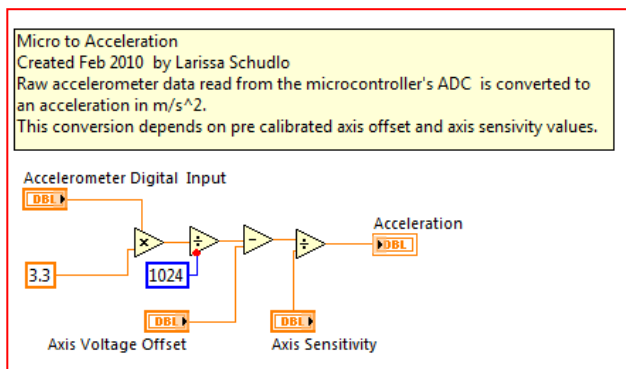


Figure 20 Microcontroller to Acceleration Conversion SubVI

## A.4.3 Offset Calibration



**Inputs:** VISA Resource (COMM port for USB device)  
**Outputs:** Y axis Offset    Z axis Offset    X axis Offset

In order to maintain optimal accuracy of the acceleration measured, the accelerometer offset voltage for each axis is calibrated before each use. The offset can shift at times, and recalibration is necessary to avoid offset calibration errors. To determine the offsets,

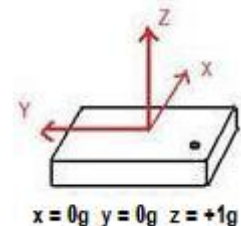


Figure 21 Accelerometer start position for calibration.



the first 10 data samples for each axis are read in and averaged. This value is then converted to the axis offset voltage. It is assumed the accelerometer will be positioned upright such that the y and x axis are in 0g positions and the z axis is in the +1g position (figure 21). Therefore the value read in for the x and y axes must only be converted from a digital number to the original analog voltage. Because the z axis represents 1g, the z-axis sensitivity must be subtracted from the original analog voltage to obtain the axis offset.

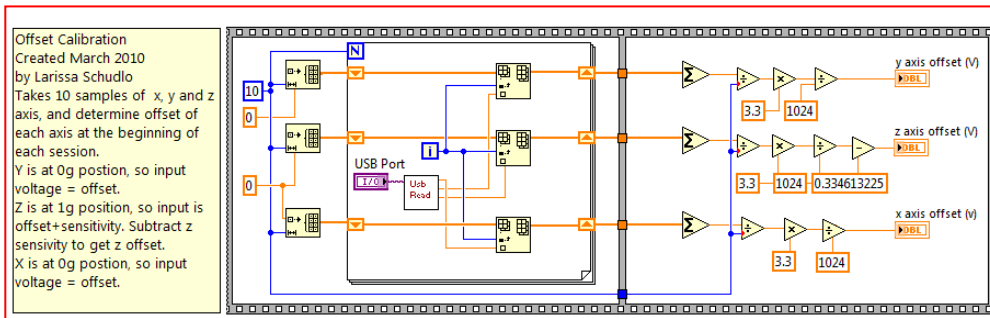


Figure 22 Offset Calibration SubVI

#### A.4.4 Acceleration to Degrees

acceleration to degrees	<b>Inputs:</b>	X Axis Acceleration	Y Axis Acceleration	Z Acceleration
	<b>Outputs:</b>	Angle of Tilt about Z Axis		

Using all three axes of acceleration, the angle of tilt from the z-axis is determined. In order to display angles between 0 and 180° correctly, the z-axis is used to determine the correct half of the Cartesian plane the angle is in.

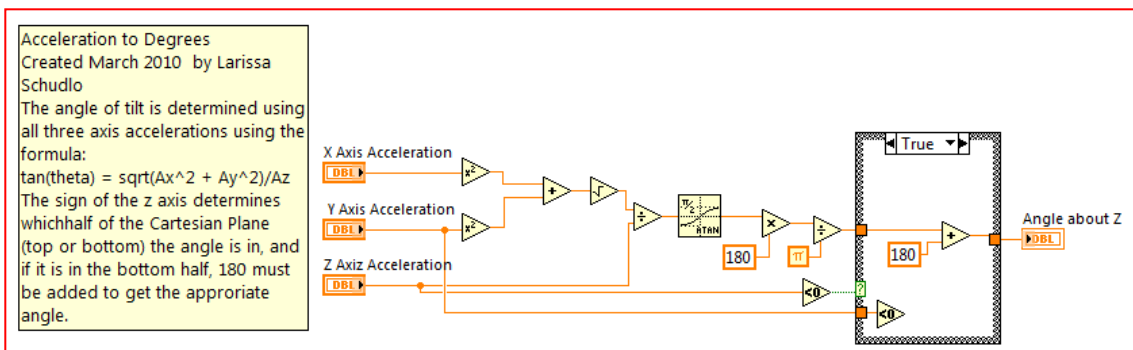


Figure 23 Acceleration to Degrees SubVI

## A.4.5 Angular Velocity

<b>Angular Velocity</b>	<b>Inputs:</b>	Current Angle	Previous Angle	Sampling Period
	<b>Outputs:</b>	Angular Velocity		

To determine the angular velocity, the degrees rotated through during the sampling period is calculated, and divided by the difference in time.

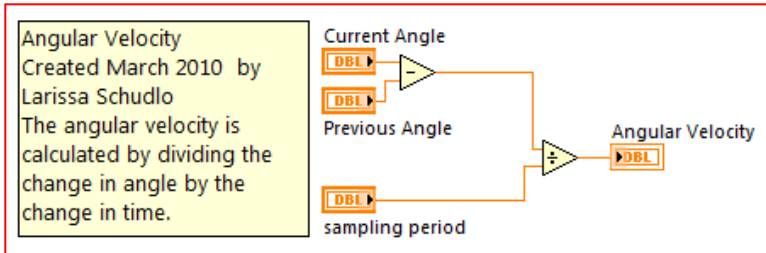


Figure 24 Angular Velocity SubVI

## A.4.6 Drawing Rotation

<b>drawing rotation</b>	<b>Inputs:</b>	Input Picture	Fill Colour	Outline Colour	Rotation
	<b>Outputs:</b>	Rotation Picture	Degrees of Rotation Y axis Acceleration		

The graphical display of rotation on the front panel is a 'Picture Function' in LabVIEW. An existing picture is passed into the SubVI so the new rotational data can be added to it. The current angle is passed in, and this amount of degrees is represented as a 'pie piece' filled in on the picture circle from zero in the Fill Colour. Using the radius of the circle, the angle of rotation and the sign of the y-axis, the 'pie piece' is outline using the Outline Colour. The altered picture is output, to be displayed on the 'Picture' on the front panel.

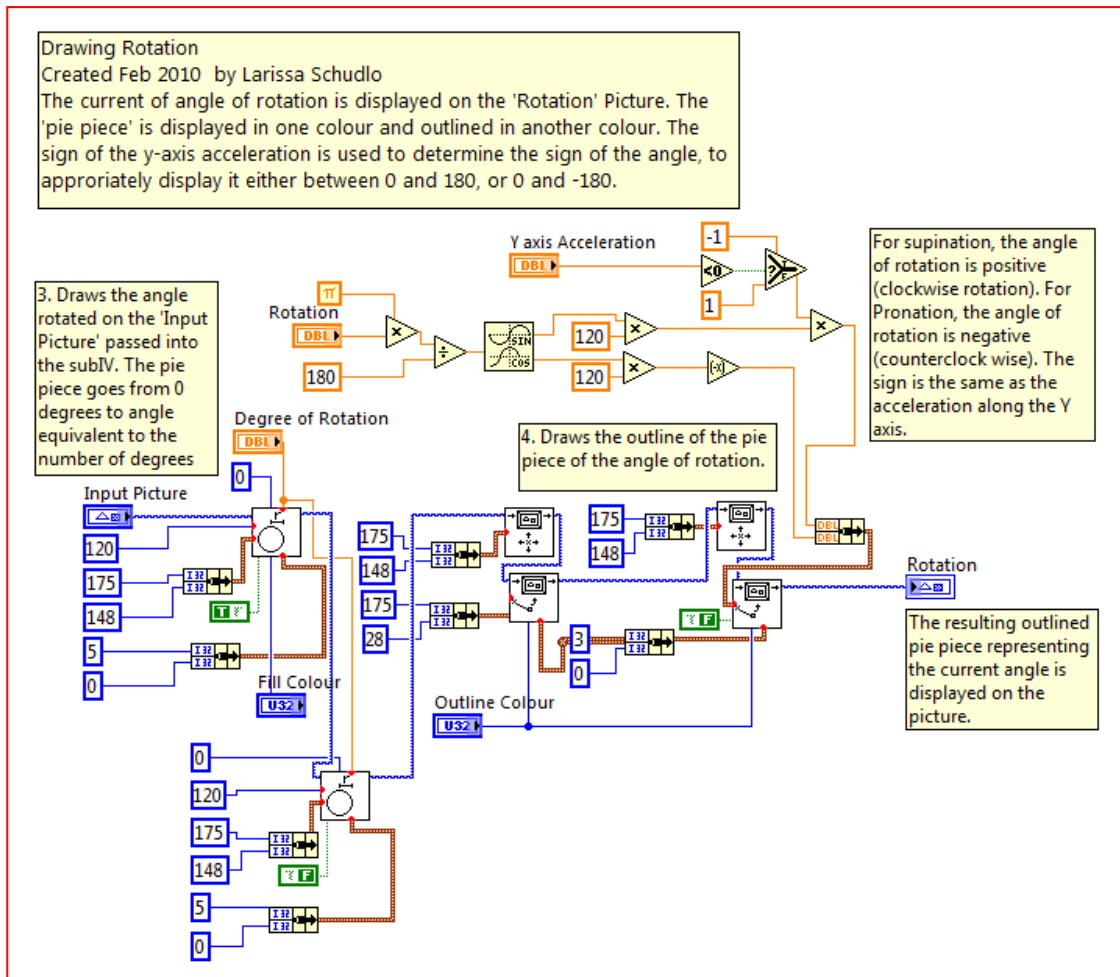


Figure 25 Drawing Rotation SubVI

### A.4.7 History File Display

History File Display	<b>Inputs:</b>	Input Picture	Fill Colour	Outline Colour	Radius of Circle
	<b>Outputs:</b>	Maximum Angular Velocity	Average Angular Velocity	Maximum Angle	Rotation Picture
		Minimum Angle	Total Degrees Rotated Through		

To display previous patient history, the user can display past rotational and/or angular velocity data. The user is first asked if they want to display rotational data. If the user selects 'yes', the file to be read is selected and the data is displayed on the 'Input Picture.' If the user selects 'no' or after they have displayed rotational data, then they are asked if they would like to display velocity data. If the user selects 'yes', they must select the appropriate file and the data is displayed. If the user selects 'no' or have displayed rotation data, the subVI closes leaving the data displayed on the front panel.

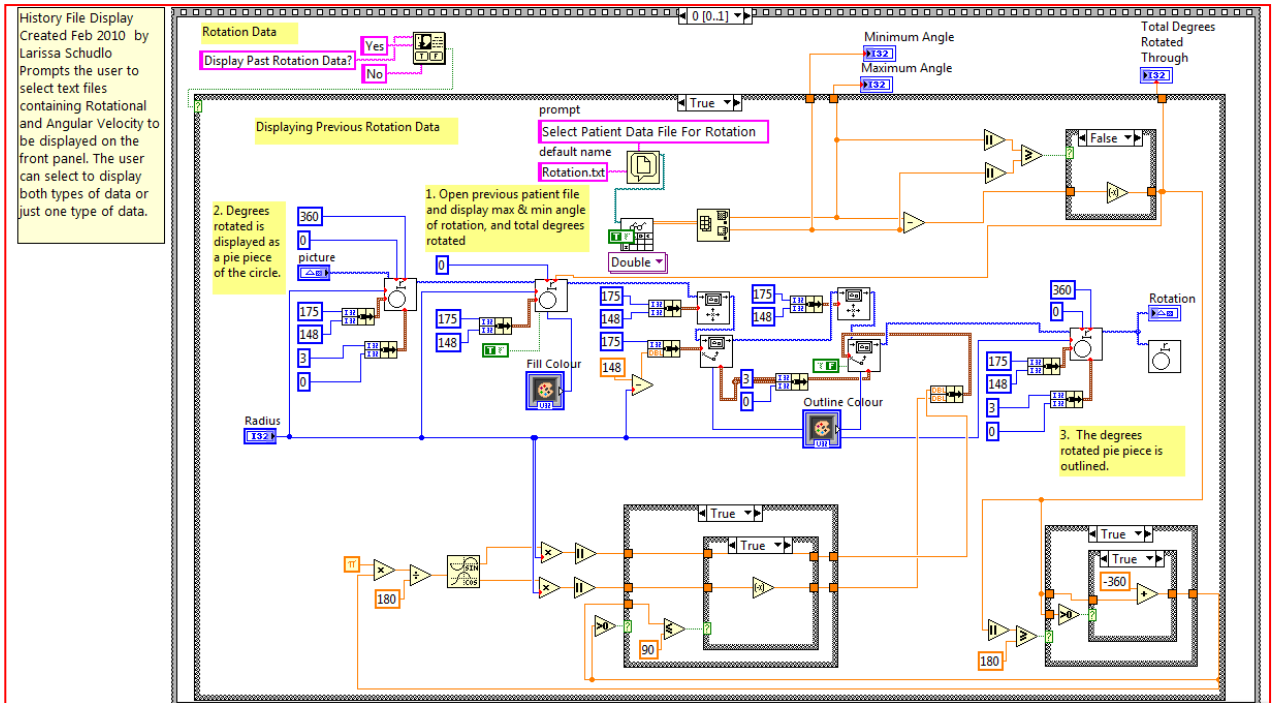


Figure 24 File Display SubVI - Rotation Data

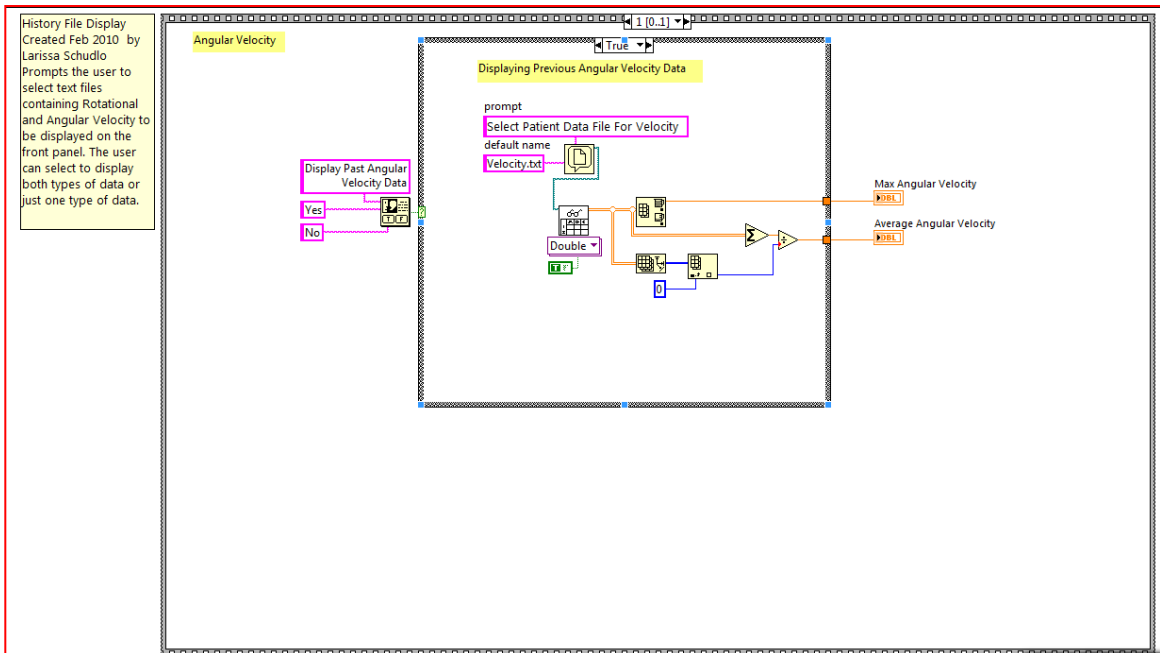


Figure 25 History File Display SubVI - Angular Velocity

## A.4.8 Measuring Rotation



<b>Outputs:</b>	Previous Minimum Angle	Previous Maximum Angle
	Total Degrees Previously Rotated Through	Current Maximum Angle
	Current Minimum Angle	Total Degrees Rotated Through Current
	Current Angle	Previous Max Angular Velocity
	Previous Average Angular Velocity	Current Angular Velocity
	Current Max Angular Velocity	Current Average Angular Velocity
	Rotation Display	Angular Velocity Display

This subVI was created to allow the user to test the activities, select past patient data or record new data after they have selected that they would like to evaluate the patient's range of motion. Upon start up, the patient can either select to test the program or select to start the program, which will start a session which they can either look up past data or record new data in. If the user selects to test the program, information is displayed on the screen. However, it does not save to a file. The sequence of events is similar to the sequence when the user selects to record new data (some of the numeric indicators are not displayed).

When the user selects the 'Start Program' button on the front panel, they are asked to select whether they want to look up past patient data or record new data. If the user selects to look up past patient data, the subVI 'History File Display' is called, and they are prompted to select the files they wish to display for up to two past sessions. Rotational data is displayed graphically and key numeric indicators are shown. Angular Velocity data is only displayed through key numeric indicators.

When the user selects 'Start Program', and chooses to record new data, they are first asked whether they would like to display the rotation and angular velocity results from a previous session. This provides motivation for the current session and allows for a quick comparison between sessions. If it is selected to display previous results, the subVI 'History File Display' is called, and the user is asked to select the appropriate files. Next, the user is prompted to create two new files, to save rotational data and angular velocity data for the upcoming recording. If the file already exists, the data will be appended to the end of the file. Once the files are created, the program is ready to start recording data. When the user is ready, data will start being displayed and recorded once the 'Start Recording' button is selected on the front panel. Once selected, this button will change colour and read 'Stop Recording.' It can be clicked during the session to stop the

recording at any time. Data to be displayed is received through serial communication. To read the data, LabVIEW has a 'VISA Class' which provides the necessary subVI's to read in serial data. On the front panel, there is a 'VISA Resource Name' indicator, and the correct serial port must be selected in order to obtain the USB data. The default port has been set for the computer the code was created on and does not need to be selected each time. Once the serial port is opened, the subVI 'Offset Calibration' is called to read in the first few data points and calibrate the accelerometer offsets. Following calibration, the subVI 'USB 4 Channel Read' is called ten times, and the three acceleration values are stored in an array and then averaged. The average values of the acceleration for each axis are passed to the subVI 'Micro to Acceleration' to convert the digital number to the actual value of acceleration. Then the resulting accelerations are passed to the 'Acceleration to Degrees' subVI to determine the angle rotated from the z-axis. The angle output will always be positive. To determine the correct sign of the angle for proper display on the front panel, the sign of the y-axis acceleration is considered. The sign of the current tilt angle takes the same sign of the y-axis acceleration, and it is adjusted accordingly. The angle is then used in several places. The subVI 'Drawing Rotation' is called to display the tilt information on the front panel graphically. The angle is passed the LabVIEW subVI's to format and append the data to a file, and to a case structure which updates and stores the current maximum and minimum angles. The minimum angle and maximum angle of rotation the patient has achieved, the total degrees rotated through, and the current angle of tilt are updated and displayed on the front panel each time new data is obtained. The angle is also used to determine angular velocity. To calculate angular velocity, the 'Angular Velocity' subVI is called, which calculates the change in angle divided by the change in time. Rather than calculating the change in angle each time a new angle is calculated, the difference between every six angles is calculated then divided by the change in time. To determine the difference in time, a LabVIEW timer is used. It is started when the first angle is obtained, and stopped when the sixth angle is obtained. The maximum angular velocity is updated and stored each time, and displayed on the front panel at the end of the session. The average angular velocity is also displayed at the end of the session by subtracting the difference between the maximum and minimum angle obtained during the span of the session, and divided by the length of the session

(determined by a timer started at the beginning). The angular velocity is also recorded in a text file. Once the user has completed their session, the recording can be stopped by selecting the 'Stop Recording' button.

# Curriculum Vitae

Larissa Schudlo

## Education

***Bachelor of Engineering – Biomedical and Electrical*** 04/ 2010  
McMaster University, Hamilton, ON

- Currently enrolled in fourth year
- Areas of Study include: Biomedical Instrumentation, Medical Imaging, Signal Processing, Microelectronics, Bioelectricity, Anatomy and Physiology
- Grade point average is 11.8 / 12

***St. Francis Secondary School - Ontario Secondary School Diploma*** 06/ 2006  
St. Catharines, ON

- Ontario Scholar and Valedictorian with a 97.6% grade point average
- Participated in Secondary School Athletics:
  - Volleyball, Basketball, Hockey, Badminton, Softball, Track & Field 2002-2006

## Academic Awards/Scholarships

- NSERC CGS Award 2010
- Dean's Honour list - McMaster University 2007, 2008, 2009
- Ontario Professional Engineers Foundation for Education Undergraduate Scholarship (\$1250) 2009
- Patrick Tan Academic Grant (\$1000) 2009
- NSERC Undergraduate Summer Research Award (\$4500) 2009, 2010
- Richard C. Newman Academic Grant (\$800) 2008
- Pollack Family Academic Grant (\$2000) 2007
- University Senate Scholarship (\$800) 2007, 2008
- Nortel Network Entrance Scholarship (\$6000) 2006 - 2007
- Ashbaugh Scholarship (\$12,000) 2006 - 2009

## Experience

***Biomedical Engineering Final Year Capstone Design Project*** 09/2009 – 04/2010  
Department of Electrical and Computer Engineering – McMaster University  
Hamilton, ON

- Currently developing an instrumentation system to quantify functionality in the upper extremity for therapists evaluating pediatric patients recovering from peripheral nerve injuries.
- Designing and developing the electronic hardware to instrument a handheld device to acquire signals from transducers including an accelerometer and pressure sensor, and transmit the data wirelessly to a computer.
- Developing a software program in LabVIEW to process the signals produced by an accelerometer through human movement and output the information numerically and pictorially on an interactive display.

***Laboratory Assistant*** 05/2009 – 08/2009  
Department of Electrical and Biomedical Engineering – McMaster University  
Hamilton, ON



- Explored electrical stimulation and sensory protection techniques to preserve muscle function following nerve injuries.
- Ran studies which included performing electrical muscle stimulation on rats, and used problem solving skills to fix any complications involving electronic equipment or anesthetic machine being used.
- Assisted in denervation surgeries performed on rats, and learned various methods of testing muscle strength and development, and tissue collection techniques.
- Used Matlab and LabVIEW to collect, analyze and display data.
- Designed amplifiers for recording bioelectric potentials, developed PCBs using EAGLE Layout Editor and assembled the finalized circuit boards.

***Laboratory Assistant***

05/ 2007 – 08/ 2007

Department of Chemical and Bioengineering - McMaster University  
Hamilton, ON

- Investigated release rates and activity of various drugs from polymeric materials for use in the eye
- Used problem solving and organizational skills to independently run experiments testing the release rate of over 15 drugs
- Ran Absorbance or Fluorescence Assays to obtain data, and then compiled and organized it in a usable form in Excel
- Gained experience using lab equipment including Autoclave machine, Coulter Counter Cell and Particle Counter, UV Absorbance Reader, and Fluorometer

**Extracurricular**

- Volunteer for McMaster Engineering I Recruitment Office 03/2009 – present
- Member of Engineering World Health at McMaster University 10/2008 - present
- Volunteer for May at Mac Open House and Scholar’s Reception 05/2009  
at McMaster University
- Volunteer for ESSCO High School Math and Physics Day 05/2009
- Participant in Canadian Cancer Society Relay for Life 3/2008
- Member of Golden Key Honour Society for academic achievement 9/2007 - present
- Play recreational volleyball and hockey 9/2002 – present
- Coached for the Niagara Rapids House League and Spikes Volleyball 2005-2006  
Programs for children from grade 4 to 8.
- Played for the Niagara Rapids Traveling Volleyball Club 2003 – 2006

## References

- [1] Fowler, N. K., & Nicol, A. C. “Functional and biomechanical assessment of the normal and rheumatoid hand,” Clinical Biomechanics, vol. 16, no. 8, pp. 660-666.
- [2] Giacomozzi, C. “New instrumental set for the assessment of the hand functionality,” Medical Biological Engineering Computation, vol. 41, no. 5, pp 513-515.
- [3] Huang, Y. Y., Low, K.H., & Lims, H.B. “Objective and Quantitative Assessment Methodology of Hand Functions for Rehabilitation,” presented at IEEE International Conference of Robotics and Biomimetics, Thailand, 2009.
- [4] Lee, J., Cho, S., Lee, J., Lee, K., & Yang, H., “Wearable Accelerometer System for Measuring the Temporal Parameters of Gait,” presented at 29th Annual International Conference of the IEEE EMBS, France, 2007.
- [5] Lui, H., & Pang, G., “Accelerometer for Mobile Robot Positioning,” IEEE Transactions of Industry Applications, vol. 37, no. 3, pp. 812 – 819.
- [6] Memberg, W.D., Crago, P.E., “Instrumented Objects for Quantitative Evaluation of Hand Grasp,” Journal of Rehabilitation Research and Development, vol. 43, no. 1, pp. 82-90.
- [7] Micera, S. “Functional assessment of hand orthopedic disorders using a sensorised glove: Preliminary results,” presented at IEEE International Conference on Robotics and Automation, Taiwan, 2003.
- [8] Yang, S., & Cho, S., “Recognizing Human Activities from Accelerometer and Physiological Sensors,” presented at IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Korea, 2008.