# INSTRUMENTATION FOR MICROSCALE MEASUREMENT AND CHARACTERIZATION OF BIO-FIBERS

by

Yeliz Karakaya

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Fall 2005

© 2005 Yeliz Karakaya
All Rights Reserved

UMI Number: 1430753

UMI Microform 1430753

Copyright 2006 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

# INSTRUMENTATION FOR MICROSCALE MEASUREMENT AND CHARACTERIZATION OF BIO-FIBERS

by

Yeliz Karakaya

Approved: _____
Jian-Qiao Sun, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Thomas S. Buchanan, Ph.D.
Chair of the Department of Mechanical Engineering

Approved: _____
Eric W. Kaler, Ph.D.
Dean of the College of Engineering

Approved: _____
Conrado M. Gempesaw II, Ph.D.
Vice Provost for Academic and International Programs

# ACKNOWLEDGEMENTS

I am so grateful for the guidance of my advisor, Professor Jian-Qiao Sun, during my research in University of Delaware. I thank him for having given me the opportunity to be one of his graduate students. I appreciate the challenges and freedom I have had while being a researcher, and the motivation and high expectations he has for those he advises. I am also thankful to him for his time, patience and understanding with my work.

I take this opportunity to thank my professor Professor Günay Anlaş in Boğaziçi University, Turkey, for having supported me with my decision to get a graduate degree in my profession abroad.

I am forever thankful to my parents who have always trusted and supported me with my decisions. Although you were so far away from me during my study in the US, I know that your hearts have always been here with me. You will always be the greatest inspirations in my life.

I also thank dear Jorge Garcia. Jorge, I enjoyed being in your class so much. You turned my frustrations into an achievement by sharing your knowledge enthusiastically in the course you taught. Your enthusiasm and willingness to help others encouraged me a lot to learn more in your class.

I greatly appreciate the gentleman, Steven Beard, who is responsible for the machineshop. Thanks so much Steve for your help, expertise and good natured conversation we had while working in your shop.

I must also thank to Chuck Hanavin for providing me with electrical equipment and components immediately in a friendly way whenever I asked for them.

And, Nick, thank you so much, too, for all the help. With your help during the limited time of my research, you certainly made an otherwise trying experience more enjoyable.

I also appreciate my friend John Fader for sharing his experience and comments on my design, and John and Mike for the good job they did helping me with machining of my parts.

Ercüment, I also want to thank you for all the support you have given in the end of my research.

I also thank my friends, the staff in the Mechanical Engineering department, and everyone else who showed me their friendliness during both rewarding and frustrating times for me at UD.

# TABLE OF CONTENTS

**LIST OF FIGURES** . . . . . **viii**
**LIST OF TABLES** . . . . . **xi**
**ABSTRACT** . . . . . **xii**

**Chapter**

**1 INTRODUCTION** . . . . . **1**

1.1 Fibers in General . . . . . 1
1.2 Applications of Fibers . . . . . 3
1.3 Mechanical Testing of Fibers and Standards . . . . . 6

1.3.1 Motivation for Mechanical Testing of Fibers at Microscale Level 7
1.3.2 Current Technologies and Techniques of Fiber Testing . . . . . 7
1.3.3 Strength and Deficiencies in Existing Technologies . . . . . 7

1.4 Scope of the Work . . . . . 8

**2 DESIGN OF TENSILE TESTER** . . . . . **9**

2.1 Design Goals . . . . . 9
2.2 Modeling and Simulation of the Tensile Tester . . . . . 10
2.3 Fabrication . . . . . 15

**3 INSTRUMENTATION** . . . . . **19**

3.1 Electrical Circuit Design and Communication of Sensors with the Microcontroller . . . . . 19

3.1.1 Force Sensor . . . . . 19
3.1.2 Displacement Sensor . . . . . 20

3.1.3 Voltage Reference . . . 20
3.1.4 Operational Amplifier . . . 22
3.1.5 Instrumentation Amplifier . . . 23
3.1.6 Variable Resistor Switch . . . 23
3.1.7 Electrical Stop Switches . . . 23
3.1.8 Devices for Data Conversion . . . 23

3.1.8.1 Multiplexer . . . 23
3.1.8.2 Analog to Digital Converter (ADC) . . . 24

3.1.9 Microcontroller . . . 24

3.2 Communication between the Microcontroller and the DC Motor . . . 26

3.2.1 Direct Current (DC) Motor . . . 26
3.2.2 Motor Driver . . . 26
3.2.3 Implementation of a PID Controller for Speed Control . . . 27

3.2.3.1 Designing the PID Controller Routine . . . 27
3.2.3.2 Experimental Tuning of Control Gains . . . 27

3.3 Data Transmission via Serial Communication . . . 28

3.3.1 LabVIEW Interface with the Microcontroller . . . 29

**4 EXPERIMENTATION . . . 35**

4.1 Sensor Calibrations . . . 35
4.2 Experimental Procedures on Nylon Fibers . . . 38
4.3 Mechanical Testing of Nylon Fibers . . . 40
4.4 Discussion . . . 41

**5 SUMMARY AND FUTURE WORK . . . 47**

5.1 Summary . . . 47
5.2 Future Work . . . 48

**BIBLIOGRAPHY . . . 49**

**Appendix**

**A THE MECHANICAL COMPONENTS AND DRAWINGS . . . . 52**
**B THE ELECTRICAL COMPONENTS . . . . . 65**
**C THE MAIN CODE . . . . . 67**

# LIST OF FIGURES


**1.1** Spider silk [20]. . . . . 2

**1.2** Distinction between a microfiber and some natural fibers [20]. . . . 3

**1.3** Air and water vapor permeability of microfibers [20]. . . . . 4

**1.4** Resistance of microfibers to wetting [20]. . . . . 5

**1.5** Evaporation of water through microfibers [20]. . . . . 5

**2.1** Model of the motor. . . . . 10

**2.2** Open loop step response of the model. . . . . 13

**2.3** Closed-loop step response of the model for the control in Equation (2.10) with different damping ratios. . . . . 14

**2.4** Tensile tester. . . . . 15

**2.5** Specimen placement. . . . . 16

**2.6** Bolts on the side of bridge supports and knobs. . . . . 17

**2.7** Tensile testing machine with the case. . . . . 18

**3.1** Model of the electrical design. . . . . 20

**3.2** The schematic design of the electrical circuit. . . . . 21

**3.3** The electrical circuit with microcontroller interface. . . . . 22

**3.4** The flow diagram of the main code. . . . . 25

**3.5** Serial communication data transfer lines [23]. . . . . 28

**3.6** Transmission of an 8-bit data by serial communication [23]. . . . . 29

**3.7** Start and stop bits [23]. . . . . 30

**3.8** 10-bit data stream with the addition of the start and stop bits. . . 30

**3.9** The front panel of the VI code. . . . . 31

**3.10** The main frame of the VI code. . . . . 32

**3.11** The first frame of the VI code. . . . . 33

**3.12** The last frame of the VI code. . . . . 33

**3.13** The detail of the code symbolized by the SubVI2. . . . . 34

**4.1** Force calibration curve. . . . . 36

**4.2** Speed calibration curve. . . . . 39

**4.3** A single fiber test specimen. . . . . 40

**4.4** Force versus displacement curve plotted automatically by the VI code. . . . . 41

**4.5** Single nylon fiber experimental force-elongation curve, I. . . . . 43

**4.6** Single nylon fiber experimental force-elongation curve, II. . . . . 44

**4.7** Single nylon fiber force-elongation curve, III. . . . . 46

**A.1** Platform (Moving plate) . . . . . 56

**A.2** Hanging plate . . . . . 57

**A.3** Fixed plate . . . . . 58

**A.4** Bridge supports . . . . . 59

**A.5** Resin rods . . . . . 60

**A.6** Mounting block . . . . . 61

**A.7** Base plate . . . . . 62

**A.8** Bottom base plate . . . . . 63

**A.9** Rack . . . . . 64

# LIST OF TABLES


**1.1** Index of fibers. . . . . 2

**2.1** Simulation parameters. . . . . 12

**2.2** Simulation results. . . . . 15

**4.1** Force calibration data. . . . . 36

**4.2** Instrumentation amplifier gain-set data. . . . . 37

**4.3** Displacement calibration data. . . . . 38

**4.4** Speed calibration data. . . . . 39

**4.5** Single nylon fiber experimental data resulting breaking force of $133.189 \pm 53.503mg$, elongation of $51.7 \pm 20\%$, and modulus of $617.1 \pm 194.2mg$. . . . . 42

**4.6** Increment in force and displacement measurements leading to resolutions of $7.676 \pm 1.510mg$ and $43.556 \pm 2.175\mu m$. . . . . 45

# ABSTRACT

Recent interest in very small scale fibers has driven testing of these materials for their mechanical properties. Due to the difficulty in handling specimens for testing and measuring small loads and displacements required for deformation, mechanical properties of these fibers have not been widely characterized. Existing technologies used for small scale fiber testing are the designs made for larger scale material testing. There exists a gap between the capabilities of existing technologies and a technology capable of truly quantitative small scale measurements. This work makes steps toward narrowing this gap by integrating small mechanical and electrical designs, enabling high resolution in small scale measurements.

In this work, the design development, modeling and simulation, fabrication, instrumentation and the calibration stages of the small scale tensile testing machine have been made. Conducting experiments on single nylon fibers, the mechanical properties of the material have been analyzed and the performance of the machine has been evaluated.

Design goals include having flexibility for testing of different size specimens at different testing speeds, elimination of factors affecting the accuracy of the tests, such as electrical noise, vibration, and airflow, building a PID controller to control the speed for different strain rate experiments and, implementation of safety precautions for saving the system from being damaged.

A MC9S0GB60 Motorola type microcontroller has been programmed as the base unit in the instrumentation. The instrumentation consists of three main parts: processing analog to digital conversion of the data, controlling the motor by Pulse

Width Modulation (PWM) method with a PID control, and implementing the RS232 serial communication protocol between the microcontroller and the PC.

The experimental data indicate the force and displacement resolutions to be 7.676 $\pm 1.510mg$ and 43.556 $\pm 2.175\mu m$ respectively.

Testing of one specimen takes less than a minute using this machine. Therefore, repetitive tests can be carried out efficiently and easily for the analysis of small scale fiber mechanical properties.

# Chapter 1

# INTRODUCTION

Much of the current rapid advancement in the bio-engineering, chemical engineering, and materials science requires characterization and manipulation of polymeric materials at ever smaller scales. Many of fiber applications are still in the research and development stage. When the challenges present in mechanical testing are overcome, more applications will be discovered.

This work details the design of a small unique tensile tester to measure force and displacement for the mechanical properties of bio-fibers.

## 1.1 Fibers in General

Any material that is characterized by having a length at least hundred times its diameter is called a fiber. All fibers are long strings of repeating chemical units. Some fibers come from ground plants that synthesize connected units of cellulose. Some are protein chains found on animals. Other fibers are spewed from insects and worms like spider webs and silk shown in Figure 1.1.

Manufactured fibers are of two types: cellulosic and synthetic. Cellulosic fibers are produced from raw materials from trees or plants. Synthetic fibers are produced from chemicals made from refined petroleum or natural gas. Table 1.1 shows basic natural and manufactured fibers.

All manufactured fibers can be engineered to produce desired qualities. They can be made many times thinner than natural fibers. Yarns from manufactured fibers are often called microfibers. Microfibers have a very good combination of

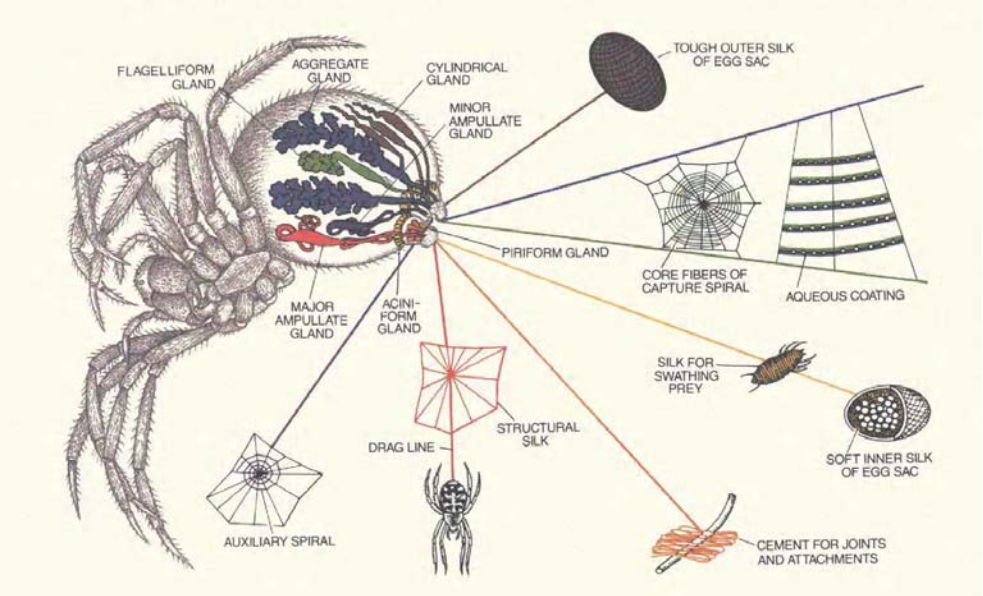


**Figure 1.1:** Spider silk [20].

**Table 1.1:** Index of fibers.

| Natural Fibers | Manufactured Fibers |
|---|---|
| Angora | Acetate |
| Camel Family | Acrylic |
| Cashmere | Elastoester |
| Cotton | Lyocell |
| Hemp | Microfibers |
| Linen | Nylon |
| Mohair | PLA Fiber (corn polymer) |
| Other Plant Fibers | Polyester |
| Ramie | Polyolefin (Olefin) |
| Silk | Rayon |
| Wool | Spandex |

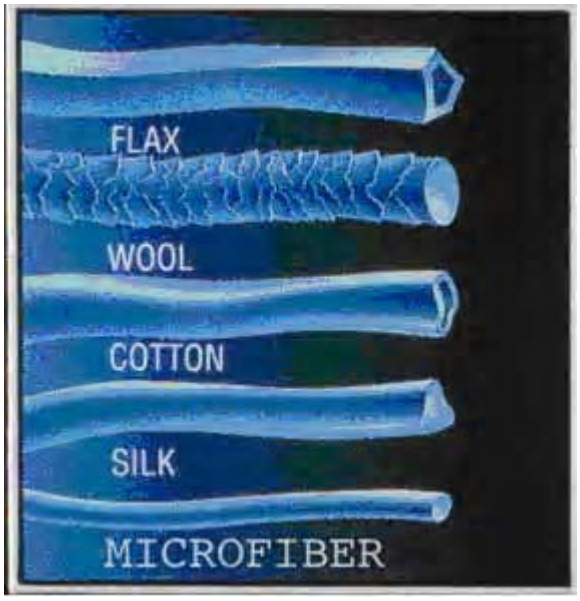


**Figure 1.2:** Distinction between a microfiber and some natural fibers [20].

quality and performance. Figure 1.2 shows that microfibers are even finer than natural fibers [20].

## 1.2 Applications of Fibers

Fibers are used in a variety of industrial applications such as biomedical, textile, nano-engineering, and traditional consumer products.

The biomedical industry holds many possible applications for bio-fibers. Medical devices, such as artificial arteries and antibacterial wound dressings, are based on manufactured fibers [20]. Tissue engineering makes a widespread use of living tissue replacements with bio-fibers. Due to the similarity between human tissue and biocompatible fibers, bio-fibers can be used for soft tissue prosthetic applications such as blood vessel, vascular and breast implants [31].

Another extensive use of manufactured fibers is in the textile industry. Textiles made from microfibers are much lighter than current fabrics available and still

**Figure 1.3:** Air and water vapor permeability of microfibers [20].

remain permeable to air and water vapor as shown by Figure 1.3.

Figure 1.4 shows the close packing property of microfibers which gives the fabric the ability to repel rain. The non-wetting surface of the fibers causes water to form beads. These beads are much larger than the spaces between the yarns. Therefore, water is effectively locked out.

Since the yarns are big enough for the passage of moisture vapor, fabrics from microfibers breathe well as illustrated in Figure 1.5.

The conductivity of manufactured fibers leads to their use in the nanoelectric machinery. A potential interest has been indicated for nanofibers to be used in micro and optoelectronics such as nanowires and photocells.

Use of manufactured fibers in traditional consumer industry is seen in the form of carpets, bedding, and window treatments [20].

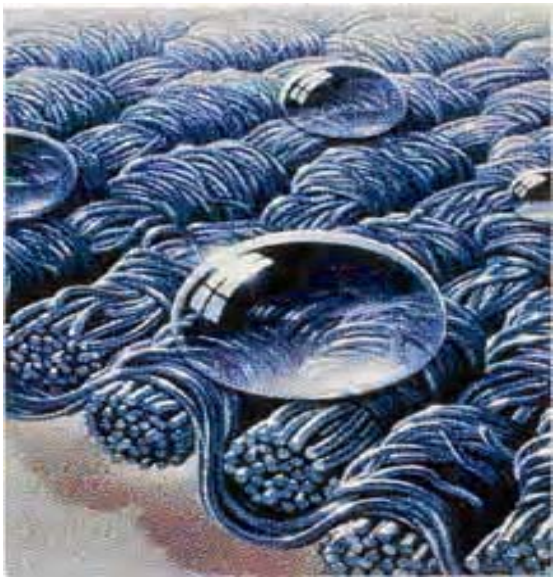

**Figure 1.4:** Resistance of microfibers to wetting [20].

**Figure 1.5:** Evaporation of water through microfibers [20].

## 1.3 Mechanical Testing of Fibers and Standards

Two procedures have been established for tensile testing of fibers: the single fiber test and the fiber bundle test. The fiber bundle test has the big advantage of being fast, but the fiber single test is more exact, if carried out carefully [5].

In a tensile stress–strain testing of fibers, the sample is elongated until it breaks. The stress is recorded as a function of extension. Stress–strain studies are usually at slow speed, of the order of mm/s.

Fiber testing is test dependent. Material parameters of fibers depend on temperature and humidity [22]. Testing must be done in the standard atmosphere, which is $21\pm1^0$C and $65\pm2\%$ relative humidity, since changes in temperature and humidity may affect the results [5].

Some mechanical tests on nonwoven mats of polymer fibers have been performed so far according to the ASTM D638, Standard Test Method for Tensile Properties of Plastics [6,21]. Mechanical testing has been performed on both natural and man-made textile fibers based on the ASTM D3822, Standard Test Method for Tensile Properties of Single Textile Fibers [14, 15, 19]. Spider silks have been tested in a manner similar to that of individual textile fibers [25,27]. Due to the small diameters of bio-fibers, there are currently no standards appropriate for them. However, the testing method for the tensile properties of natural and man-made single textile fibers comes closest to the testing method performed on bio-fibers.

The ASTM D76, Standard Specification for Tensile Testing Machines for Textiles, gives the specifications covering the operation of tensile testing machines used for testing textile fibers. It covers constant rate of extension (CRE-type), constant rate of transverse (CRT-type), and constant rate of load (CRL-type) machines. In both CRE and CRT types of machines, the pulling clamp moves at a uniform rate. However, while the other clamp on force measuring mechanism moves a negligible distance, less than 0.13 mm, in CRE types, it moves considerably to actuate the

force measuring mechanism in CRT types. In CRL types, the rate of increase of the force is uniform [4].

### 1.3.1 Motivation for Mechanical Testing of Fibers at Microscale Level

When scientists search for a new polymer for making high strength fibers, they have to consider many combinations of different chemicals and processing steps. Each time, they have to form and test the fiber to see if it performs as expected. However, it takes over a week to evaluate just one product.

Today, more than half of the fibers used across the globe are manufactured fibers [20]. As world economies and populations continue to expand, there will be greater need to produce better quality fibers. Therefore, there is an increasing need in the mechanical property analysis of very small scale fibers.

### 1.3.2 Current Technologies and Techniques of Fiber Testing

Traditional large scale testing machines such as Instron Machines and Universal Testing Machines have been used to perform mechanical tests on nonwoven mats of polymer fibers [10, 16, 24, 26, 34]. The Hopkinson tensile bar machine is also among the testing apparatuses that are used to study the mechanical behavior of fibers [33].

Although there are not many, some small scale apparatuses designed for fiber tensile testing also exist [29, 30].

### 1.3.3 Strength and Deficiencies in Existing Technologies

Currently, testing is mainly being performed on fiber mats and bundle fibers, not on single fibers due to their small diameters. Existing technologies are inadequate for mechanical testing on single fibers. In order to use the fibers for tensile tests, a method of handling them for the test is essential. As a result, most tensile testing systems for small scale fibers are custom-made incorporating various

load measuring devices such as precision load cells, cantilevers [10, 11] and atomic force microscope based nanoindentation system [12]. Other types of tests include using atomic force microscopes to perform a nanoscale three point bend test on a nanofiber [32]. Today, tensile tests of single fibers have been carried out under optical microscope systems with movable stage and cantilevers as load sensors [11, 13]. However, such systems are expensive and inefficient. They are not suitable for conducting a large number of tests.

### 1.4 Scope of the Work

The present project develops a test system for evaluating the mechanical properties of small scale fibers made of new bio-materials. The system is fully automated and enables fast and easy testing.

In Chapters 2 and 3, the mechanical and electrical designs of the tensile tester are explained. The experimental results obtained using the mechanical testing device are presented in Chapter 4. In Chapter 5, a summary of the work is presented and future work is discussed.

# Chapter 2

# DESIGN OF TENSILE TESTER

## 2.1 Design Goals

There are seven important goals that have been aimed to build the small scale fiber tester in this study: (1) Designing and machining mechanical components for the accurate motion and measurements in very small scales is extremely important. Alignment of the platform, where the uniaxial motion takes place, has great importance on force and displacement measurements. In addition, having the least possible friction coming out of the motion of mechanical components during experimentation is also important. (2) Since the expected force and displacement readings are in micro levels, sensors should have small measurement ranges and high resolutions. (3) In order to conduct experiments for different size of specimens at different testing speeds, the design should be flexible for necessary arrangements. (4) Saving the force sensor from being damaged during the placement of specimen on the tester is also crucial. (5) Any factors which might cause wrong or low resolution force or displacement measurements such as vibration, airflow, and electrical noise should be eliminated. (6) Since the speed has to be constant throughout the experiment according to the ASTM D3822 Standard, a practice of closed loop speed control has been aimed. (7) Implementation of safety measures to stop the motor automatically has also been aimed for the sake of the system.

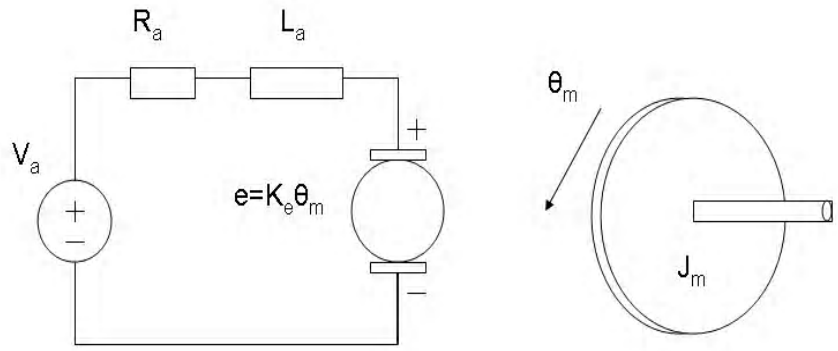


**Figure 2.1:** Model of the motor.

## 2.2 Modeling and Simulation of the Tensile Tester

To study the feasibility of speed control, we first simulate the model of the system. Figure 2.1 shows the electrical circuit of the motor and the rotor.

The governing equation for the electrical circuit is defined as

$$L_a \frac{di_a}{dt} + R_a i_a = V_a - K_e \dot{\theta}_m, \tag{2.1}$$

where $L_a$ is the inductance of the motor, $i_a$ is the current, $R_a$ is the internal resistance of the motor, $V_a$ is the input voltage, $K_e$ is the EMF constant, and $\theta_m$ is the angle swept by the rotor.

The rotational motion of the rotor based on Newton's 2nd law of motion is given by

$$J_m \ddot{\theta}_m + b\dot{\theta}_m = K_t i_a - T_l, \tag{2.2}$$

where $J_m$ is the rotational moment of inertia of the rotor, $b$ is the assumed rotational damping, $K_t$ is the torque constant of the motor, and $T_l$ is the load applied to the system.

According to Newton's 3rd law of motion $T_l$ and the internal force, $F_{in}$, are related to one another by

$$T_l = F_{in}R, \tag{2.3}$$

where $R$ is the radius of the pinion.

The translational motion of the platform based on Newton's 2nd law of motion is written as

$$M\ddot{x} = F_{in} - c\dot{x}, \tag{2.4}$$

where $M$ is the mass of the platform, $x$ is the displacement of the platform, and $c$ is the damping coefficient.

The relationship between the linear and angular speeds is given by

$$\dot{x} = R\dot{\theta}_m. \tag{2.5}$$

Applying Laplace transform, we obtain a relationship between the input voltage and the speed of the platform,

$$sX(s) = V(s) = \frac{K_t R V_a(s)}{As^2 + Bs + D}, \tag{2.6}$$

where $V(s)$ is the speed in s domain. For convenience in the later analysis, three shorthand parameters, $A$, $B$, and $D$ are defined as

$$\begin{aligned} A &= J_m L_a + MR^2 L_a, \\ B &= J_m R_a + bL_a + MR^2 R_a + cR^2 L_a, \\ D &= bR_a + cR^2 R_a + K_t K_e. \end{aligned} \tag{2.7}$$

From Equation (2.6), the governing equation of the model is obtained as

$$\ddot{v} + 2\zeta w_n \dot{v} + w_n^2 v = K_o v_a(t), \tag{2.8}$$

where $v$ is the speed in time domain, $\zeta$ is the damping ratio, $w_n$ is the natural frequency, and $K_o$ is the gain of the system. $w_n$, $\zeta$, and $K_o$ and are given by

**Table 2.1:** Simulation parameters.

| | |
|---|---|
| $L_a$ (motor inductance) | 0.062 H |
| $R_a$ (motor internal resistance) | 1 KOhm |
| $K_e$ (EMF constant) | 35.306 Nm /A |
| $J_m$ (rotational moment of inertia) | 14.38 x $10^{-10}$Nms$^2$/rad |
| $b$ (assumed rotational damping) | 0.001 Nms /rad |
| $K_t$ (torque constant) | $K_e$ in SI unit system |
| $R$ (radius of the pinion) | 5.5 mm |
| $M$ (mass of the platform) | 0.67kg |
| $c$ (assumed mechanical friction) | 0.001 Ns /m |

$$
\begin{aligned}
w_n &= \sqrt{\frac{D}{A}}, \\
\zeta &= \frac{B}{2\sqrt{AD}}, \\
K_o &= \frac{K_t R}{A}.
\end{aligned}
\tag{2.9}
$$

Equation (2.8) has been simulated for the open loop step response with the parameters in Table 2.1. Figure 2.2 shows the response plot obtained. The response has overshoot and steady state tracking speed error.

Consider a control given by

$$v_a = K_p v_r + K_d \dot{v} \tag{2.10}$$

where $v_r$ is the reference speed, $K_p$ and $K_d$ are the control gains. Then, we have

$$\ddot{v} + 2\left(\zeta + \delta\right) w_n \dot{v} + w_n^2 v = K_o K_p v_r, \tag{2.11}$$

where $\delta$ is defined as

$$\delta = \frac{K_d K_o}{2 w_n}. \tag{2.12}$$

In order to find $K_p$, we apply the Final Value Theorem (FVT)

$$\lim_{s \to 0} s G_{CL} \frac{1}{s} = 1, \tag{2.13}$$

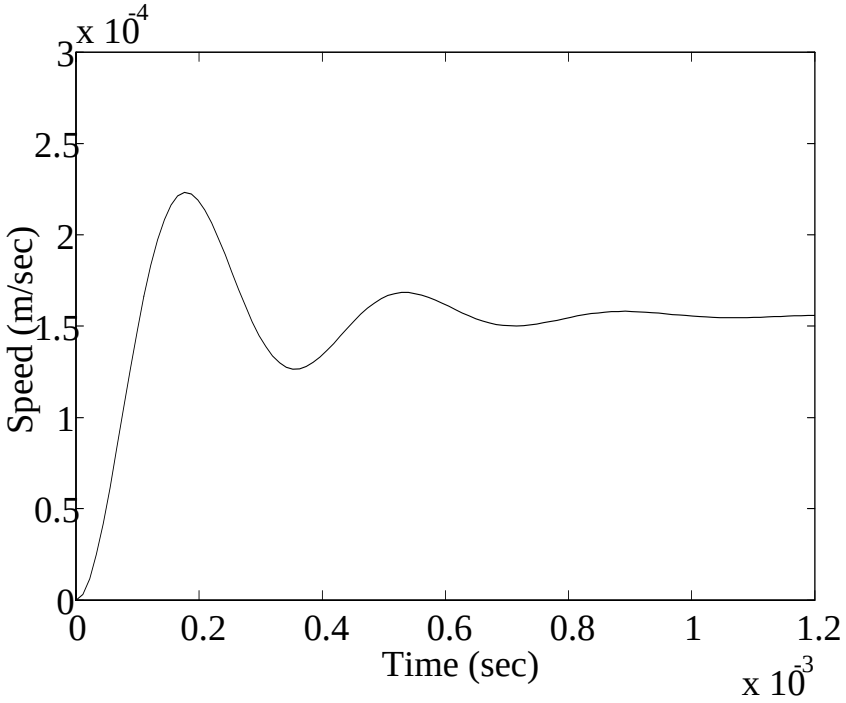


**Figure 2.2:** Open loop step response of the model.

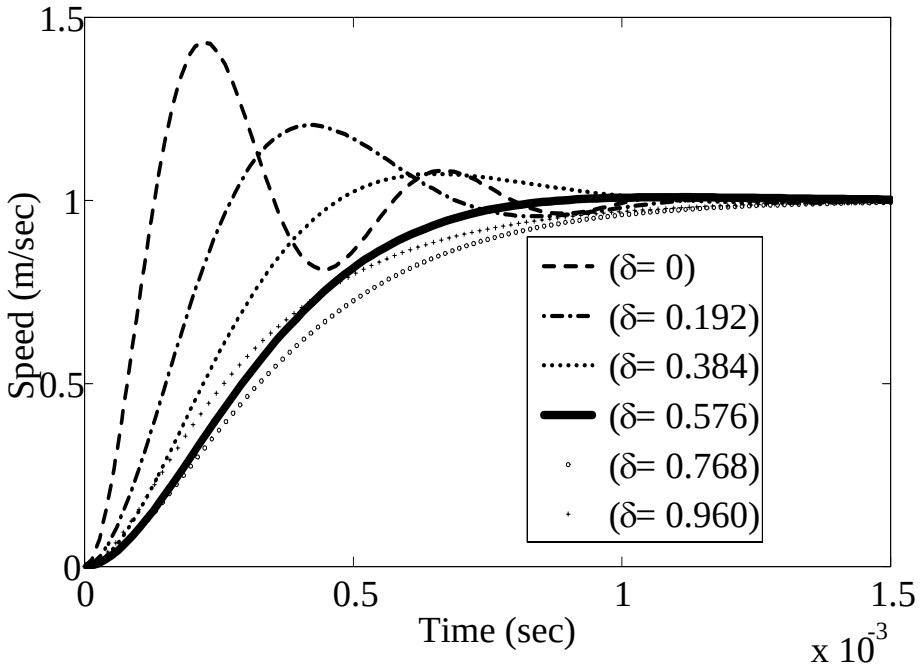


**Figure 2.3:** Closed-loop step response of the model for the control in Equation (2.10) with different damping ratios.

where $G_{CL}$ is the closed-loop transfer function,

$$G_{CL} = \frac{K_o K_p}{s^2 + 2w_n(\zeta + \delta)s + w_n^2}. \tag{2.14}$$

Hence,

$$\frac{K_o K_p}{w_n^2} = 1 \tag{2.15}$$

Simulations of Equation (2.11) with a step input are shown in Figure 2.3. $K_d$ is chosen to meet the overshoot requirement.

From the figure, we choose the step response having the smallest overshoot, leading to $\delta = 0.576$. The control gains are obtained and listed in Table 2.2.

The closed loop speed control for the system has been implemented with an assumed damping coefficient.

**Table 2.2:** Simulation results.

| | |
|---|---|
| $K_p$ (proportional control gain) | 6420 |
| $K_d$ (derivative control gain) | 0.236 |

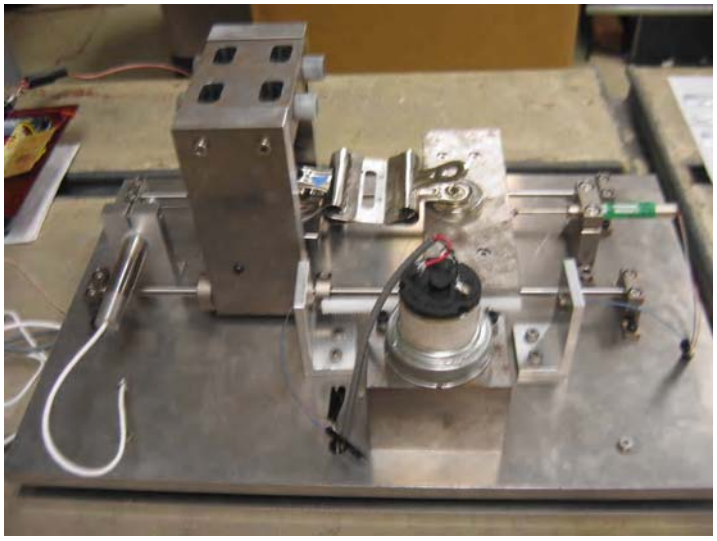

**Figure 2.4:** Tensile tester.

## 2.3 Fabrication

The testing machine shown in Figure 2.4 is designed for both single and mat fiber testing. All the mechanical components as well as the mechanical drawings are given in Appendix A.

The basic idea of the design is to move a platform on two shafts and hang down a plate by four strings. The specimen to be tested is placed on top of these two plates shown in Figure 2.5. The moving plate, or the platform, is designed so that it is attached to the displacement sensor on one end in order to retract and extract the sensor while moving forward and backward. It moves by the linear motion provided by a rack attached to it. The rotational motion of the Direct Current (DC) motor is converted into a linear motion via a pinion-rack.

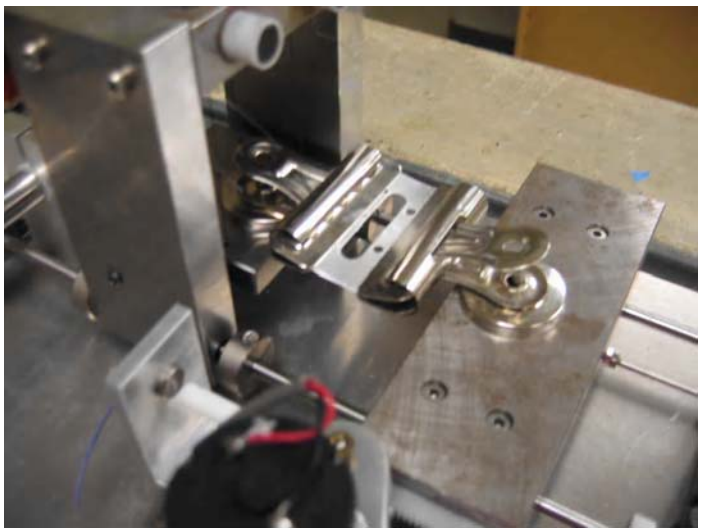

**Figure 2.5:** Specimen placement.

The hanging plate is attached to the force sensor by a small shaft mechanism. On moving, it applies a uni-axial force to the sensor.

High precision of the shafts that the moving plate slides on is important for the alignment of the platform and the reduction of the friction. In order to measure the uni-axial force applied to the force sensor accurately, the sliding mechanism has to be perfectly aligned. Also, when the platform slides forward and backward, the displacement sensor should not have any difficulty to get retracted or extracted. While friction cannot be completely eliminated, the shafts with high precision and polishing can reduce it.

The specimen is placed on top of the plates by two clips as shown in Figure.2.5. The clips are used to grip the fiber specimen. To prevent the clips from slipping during the experiment, a special clip having magnets on its bottom surface is used. Both plates are made of carbon steel to increase the magnetic force between the contact surfaces. Moreover, the clips are placed on top of the plates through two small-diameter rods keeping the clips in position and from slipping.

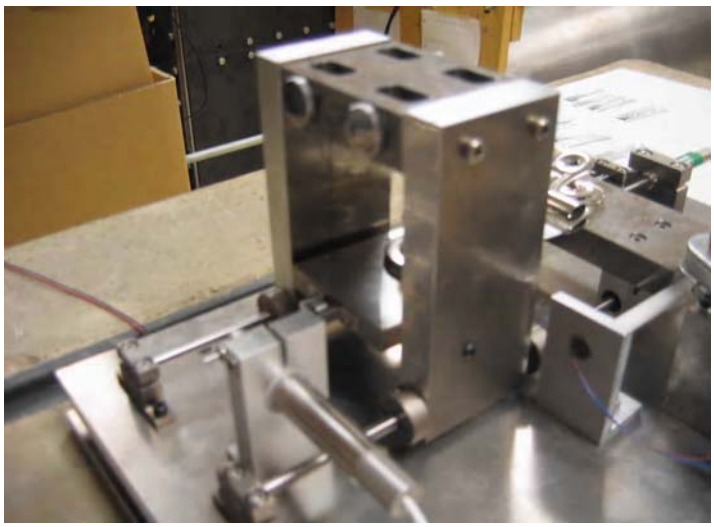

**Figure 2.6:** Bolts on the side of bridge supports and knobs.

Attention has to be given to save the force sensor from accidental high loading. Therefore, the two tapped holes are drilled on the sides of bridge support shown in Figure 2.6 to fix the hanging plate when placing the specimen.

The tensile tester is flexible for the test of different size specimens. The distance between the platform and hanging plate can be rearranged by sliding the bridge supports over the shafts. After determination of the desired distance between the plates, set collars are fixed so that the bridge supports do not slide. If needed, the altitude of the hanging plate can also be changed by two knobs shown in Figure 2.6.

The design also takes care of the vibration disturbances due to the DC motor. Four vibration isolators have been used to eliminate the vibration. In addition, a special concrete stand has been used to put the tensile tester on. It isolates the vibration created by the machines operating inside the building.

In order to keep the force measurements from being influenced by the airflow in the medium, an acrylic case shown in Figure 2.7 is used to cover the tester, when

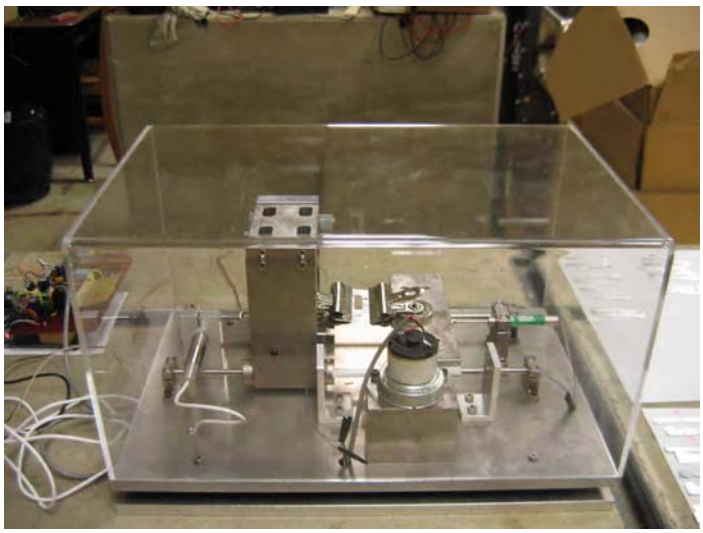

**Figure 2.7:** Tensile testing machine with the case.

experiments are being conducted.

# Chapter 3

# INSTRUMENTATION

Instrumentation has been implemented mainly in three different parts. The microcontroller is the base unit that communicates with the sensors, motor, and the PC. The diagram in Figure 3.1 shows these three different communications. Sections 3.1, 3.2 and 3.3 discuss each part separately.

## 3.1 Electrical Circuit Design and Communication of Sensors with the Microcontroller

All the electrical components used in the instrumentation are listed in Appendix B. The schematic design of the electrical circuit and the electrical circuit with microcontroller interface are shown in Figures 3.2 and 3.3 respectively.

In this section, each component used in the instrumentation is discussed.

### 3.1.1 Force Sensor

In this design, an SLCB-50 beam type load cell has been used as the force sensor. Beam type load cells are commonly employed for measuring low level loads [17]. As seen in Figure 3.2, the sensor is excited by 5 volt Direct Current (DC) voltage. It has infinite resolution and 50 gram measurement range. Infinite resolution means its output is analog. The smallest value this sensor can read is limited to the noise in the circuit and the resolution of analog to digital converter (ADC).

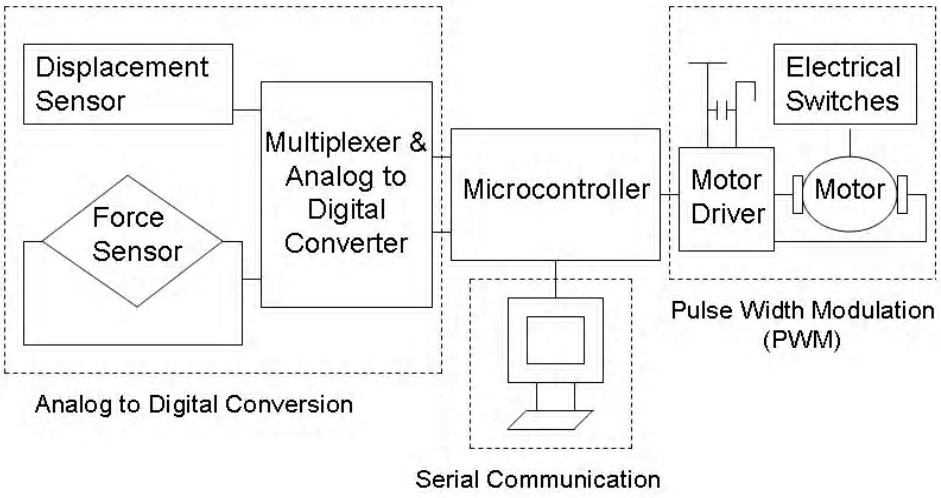


**Figure 3.1:** Model of the electrical design.

### 3.1.2 Displacement Sensor

An LP-20FQ type linear potentiometer measures the displacement in uni-axial direction in this design. It has 25 mm stroke and converts linear displacement into resistance, which is converted directly to a voltage signal [28].

### 3.1.3 Voltage Reference

Fluctuations that exist in the main line electricity are sometimes in excess of what is tolerable in a scientific application. Voltage references are used to reduce the fluctuations in DC voltage due to ripple, line voltage variations, and load current variations. A voltage reference basically takes its input as any voltage a few volts higher than the design DC output voltage and reduces it to a certain steady DC voltage [9].

Shown in Figure 3.2 is an ADR423 type voltage reference that regulates 6V DC input to a 3V clean DC voltage to feed the displacement sensor. Having a

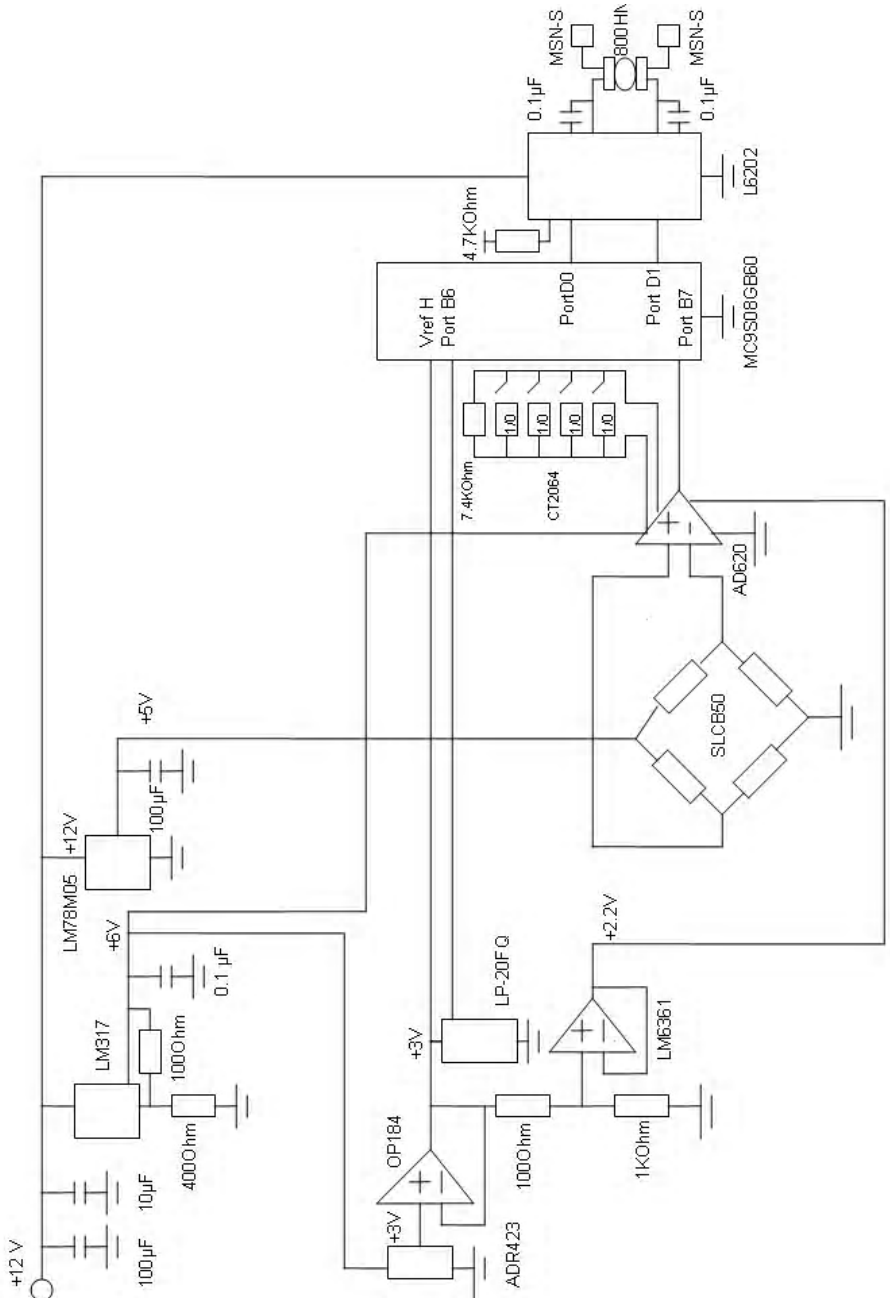


**Figure 3.2:** The schematic design of the electrical circuit.

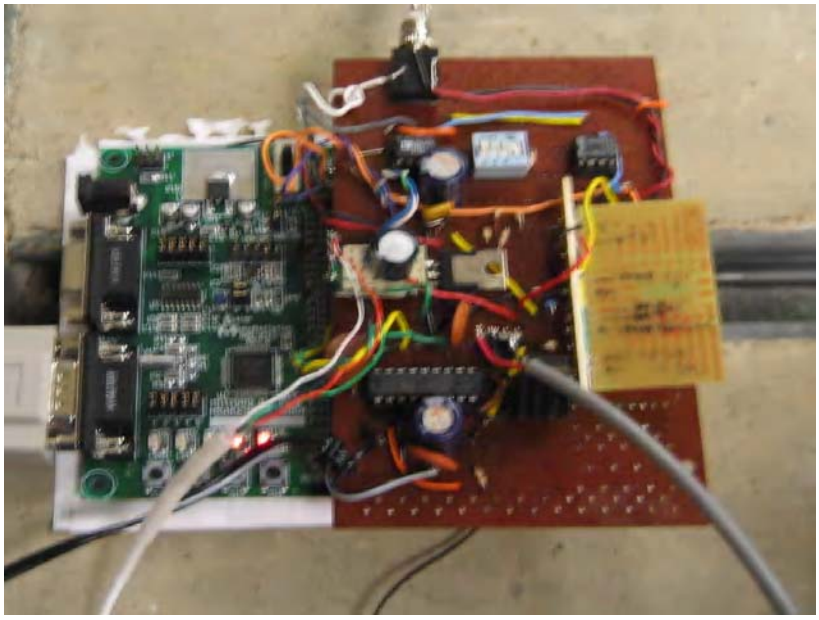

**Figure 3.3:** The electrical circuit with microcontroller interface.

similar functionality with a voltage reference, an LM78M05 type voltage regulator has been used to reduce 12V DC voltage to a 5V clean reference to supply the force sensor. Since the whole circuit is powered with only 12V DC voltage, an LM317 type variable regulator produces 6V DC voltage necessary for the instrumentation amplifier.

### 3.1.4 Operational Amplifier

Since the current drawn from the voltage reference ADR 423 is not capable of driving the displacement sensor, an OP184 type operational amplifier is connected in a way that it buffers its input to keep the reference from being loaded by the sensor. The other operational amplifier in the design is the LM6361 shown in Figure 3.2.

### 3.1.5 Instrumentation Amplifier

As very small level of forces is measured in this design, the output signals of the force sensor are not at the level to read by any device. The instrumentation amplifier AD620 amplifies the force sensor output to a detectable level. The gain of this amplifier is set by the addition of a single external resistor across its pins.

### 3.1.6 Variable Resistor Switch

A CT2064 type variable resistor switch shown in Figure 3.2 is used to change the instrumentation amplifier gain for testing different materials. The switch has four buttons. For any On (1) and Off (0) combination of buttons, the resistance across the external resistor of the instrumentation amplifier changes.

### 3.1.7 Electrical Stop Switches

Two MSN-S type electrical stop switches have been included in the design for a safety purpose. In case the motor does not stop at the limit positions of the motion, which is controlled by a piece of code written, the electrical switches stop the motor. Having a normally open circuit is their characteristic. When sensing the platform, they send a signal to stop the motor closing the circuit.

### 3.1.8 Devices for Data Conversion

#### 3.1.8.1 Multiplexer

Since the information from both force and displacement sensors has to be read one at a time, a multiplexer has been used in the design. A multiplexer is a switching device which enables each of the inputs to be sampled in turn [28]. In this application, it is a built-in function of the M68DEMO908GB60 Free Scale demonstration board. It passes the data to the analog to digital converter, after processing.

#### 3.1.8.2 Analog to Digital Converter (ADC)

The analog to digital conversion is discrete, taking place one number at a time. The ADC is also a built-in function of the demo board. It is a 10 bit ADC with 3.3V full scale range. So, it represents analog voltages in the uni-polar range between 0 and 3.3 with $2^{10}$ different binary values. It means that each bit step represents 3.3V/1024, 3.2 mV. This is the resolution of the ADC.

### 3.1.9 Microcontroller

The MC9S08GB60 Motorola microcontroller used in this study has 7 ports with different functionalities. The 7 ports have 64 pins in total. Each pin is programmed to functionalize the port it belongs to.

The Port B, TPM1 timer with Port D, and both SCI1 and SCI2 serial communication interface modules have been programmed using Assembly language. The flow diagram of the main code is shown in Figure 3.4. The full code is included in Appendix C.

The Port B pins are the ADC input pins. The Port B6 and B7 pins have been functionalized for displacement and force readings respectively. As seen in Figure 3.4, to reduce the noise existing in the sensors, filtering has been done by averaging the readings. For the force sensor, the Port B7 pin has been programmed to stop the motor, before the force readings reach the limit of the sensor.

Port D has two TPM modules, TPM1 and TPM2 known as timers. When any of these timers are enabled, the programmed Port D pins are controlled by that timer. Timing function controls frequency and range, which is the period between overflow limits of the time reference. The timing system is suitable for motor control in small appliances. A control bit in each timer configures all channels in that timer to operate as center-aligned Pulse Width Modulation (PWM) function to control the motor. In this design, the Port D0 and Port D1 pins have been enabled to operate the PWM module.

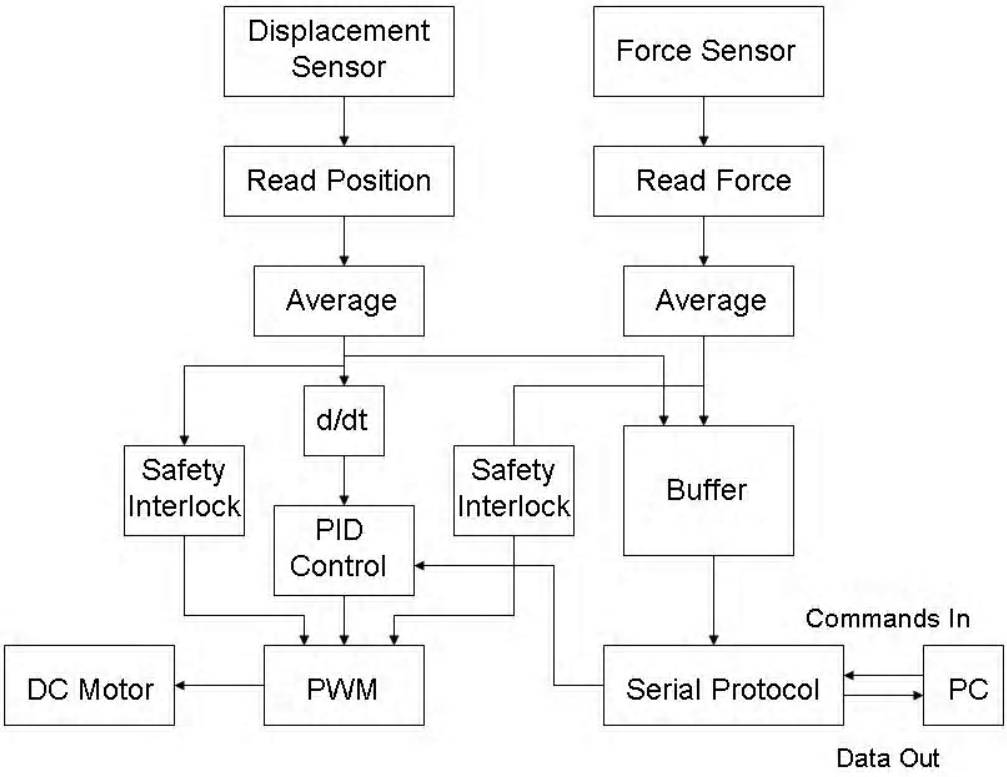


**Figure 3.4:** The flow diagram of the main code.

The serial communication interface modulus provide the RS232 serial communication between the microcontroller and the PC. The SCI1 has been programmed for debugging the main code only. Whereas, the SCI2 is always kept connected to the PC for the RS232 serial communication [7].

## 3.2 Communication between the Microcontroller and the DC Motor

Since a DC motor, an analog type of device, is used in this application, it is necessary to convert the digital binary output from the microcontroller into an analog form. There is no digital to analog converter on the demonstration board. It is the PWM output doing digital to analog conversion. Its converters reconstruct the digital signals into analog signals for the purpose of actuation.

### 3.2.1 Direct Current (DC) Motor

DC motors are capable of providing precise and continuous control of speed by varying the voltage applied to the motor [28]. The DC motor used in this design is a permanent magnet (PM) type 800HN Cramer model. The PWM feature of the microcontroller has been programmed to vary the voltage applied to the motor.

### 3.2.2 Motor Driver

Since the current drawn from the microcontroller is 60 mA per pin at best, it is not high enough to drive the motor. Therefore, an amplifier is needed to increase the power. For motor control applications, mostly this is achieved using H-Bridges as motor drivers. An L6202 type H-Bridge has been used in this design. It increases the current up to several amps at 50 volts, dissipating very little power. In this application, its pins also have been enabled to perform the power stage required for both speed and direction control [1].

### 3.2.3 Implementation of a PID Controller for Speed Control

In order to build the speed controller, three main elements are needed in this application: regulating the voltage applied to the motor to change its speed, measuring the absolute speed of the platform, and running a control algorithm to keep the speed constant. A LabVIEW VI code discussed in Section 3.3.1 has been written for changing the applied voltage. With the use of the displacement sensor and the clock of the microcontroller, a series of pulses with a frequency proportional to motor speed is obtained for the measurement of the absolute speed. The Assembly-program code for the Proportional Integral Derivative (PID) control has been written and included in Appendix C.

#### 3.2.3.1 Designing the PID Controller Routine

The PID control takes the setpoint of the speed from the user interface. An interrupt is used to measure the platform speed via the displacement sensor and the clock of the microcontroller. A periodic interrupt is used to work out the difference between the actual speed and the required speed. The speed error decides the duty cycle of the PWM output to control the motor.

#### 3.2.3.2 Experimental Tuning of Control Gains

Theoretical analysis of the system model provides a baseline for the control gains. Because the mathematical model of the system is based on simplifying assumptions, theoretically designed controls are often tuned experimentally. Here, we describe the tuning procedure that we developed for our control design.

First, the setpoint is set to the maximum speed with the integral and derivative gains at zero. Then, the proportional gain is increased so that the speed reaches the maximum possible before a speed oscillation occurs. The proportional gain has been found to be 10,000. After that, the setpoint is reduced to zero, and changes in setpoint are repeatedly applied while increasing the integral gain gradually until

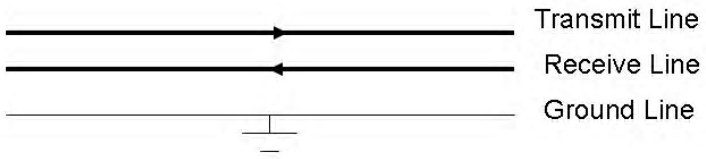


**Figure 3.5:** Serial communication data transfer lines [23].

the speed starts to overshoot. This gives the integral gain to be 10. Since the speed is already settled well, the derivative gain needs not be tuned again.

### 3.3 Data Transmission via Serial Communication

The PC and the microcontroller in the automatic system communicate with each other by an RS232 serial communication method launched by the LabVIEW VI code. Serial communication method transmits data bit by bit [18]. Only three communication lines are needed for data transfers between the PC and the microcontroller. These are transmit, receive, and a reference ground lines shown in Figure 3.5.

In serial communications, a high level is called a Mark state, while the low level is called the Space state. The output line is in a high state, which is often denoted as a 1. The transmitter signals the receiver that it is about to send a data byte by pulling the transmit line low to the space state, which is denoted as a 0. This falling edge is the signal for the receiver to get ready for incoming data. In RS232 communication, all data bits are sent and held at a constant baud rate, which is the frequency of data transmission measured in bits per second [23]. In the VI code, the baud rate is set to 115,200.

In order to ensure that the reference speed is received by the microcontroller

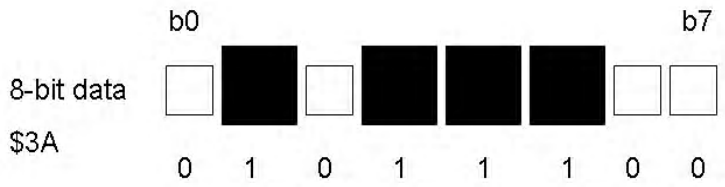


**Figure 3.6:** Transmission of an 8-bit data by serial communication [23].

correctly, the reference speed set by the LabVIEW is monitored using an oscilloscope. For a serial communication, the protocol demands that the least significant bit (b0) be transmitted first with the most significant bit (b7) last. By convention, time is represented as moving from left to right. Hence, an 8 bit data byte ($3A) or in binary representation (00111010), for example, is transmitted as (01011100) in reverse order shown by Figure 3.6.

The protocol also requires that the data byte be framed by two special bits, the start bit, a Space state, and the stop bit, a Mark state. The start bit follows the falling edge signal to the receiver and thus is a low 0. The stop bit signaling that the data byte transmission is completed is always high 1. The addition of the start and stop bits to the serial data byte generates a 10 bit data stream shown in Figure 3.7 [23].

The timing diagram of ($3A) is observed in the oscilloscope as seen in Figure 3.8.

### 3.3.1 LabVIEW Interface with the Microcontroller

LabVIEW, an acronym for Laboratory Virtual Instrument Engineering Workbench, is a graphical programming language to create complex test and measurement systems. Each VI is made up of three components: the front panel, the block diagram, and the icon and connector pane [2]. The front panel is built using a set of

**Figure 3.7:** Start and stop bits [23].

**Figure 3.8:** 10-bit data stream with the addition of the start and stop bits.

tools and objects. The front panel is the user interface that simulates the panel of a physical instrument. Figure 3.9 shows the front panel in this study.

Source code is written in the block diagram by using graphical representations of the functions in order to control the objects located on the front panel. The block diagrams in Figures 3.10, 3.11 and 3.12 resemble a flow chart.

VIs are hierarchical, when they are used as subprograms within other programs. The icon and connector pane allows the user to identify the VI with a name and symbol so that it can be submitted into another program as a SubVl. The piece of code shown by Figure 3.13 details the SubVI2 icon in Figure 3.10.

As seen in Figure 3.9, serial port address (COM1) is defined in LabVIEW serial communication applications. Defining the address correctly, LabVIEW sends the input commands to the microcontroller and receives the sensors data from the same address in return.

**Figure 3.9:** The front panel of the VI code.

LabVIEW provides a library of VISA Input/Output functions to establish the communication. Figure 3.10 shows that the VISA Open function initiates the VISA serial function with the defined baud rate of 115,200 baud, the 8 bit data byte, a stop bit of 1 and no parity. There are three input commands that LabVIEW sends to the microcontroller subsequently: s, r, and b commands. Pressing the run button on the front panel, the VI code sends the s command to the microcontroller shown in Figure 3.11. The microcontroller starts running the motor by this command. As seen in Figure 3.10, SubVI2 writes r command to the microcontroller. After gathering the force and displacement data, the microcontroller sends them to the PC. Since the data comes to the PC in ASCII characters and without being split into two, the Type Cast function converts ASCII characters into decimal numbers, and then the Index Array function splits the data into two elements, which are force and displacement elements. The force and displacement data are ready to plot force

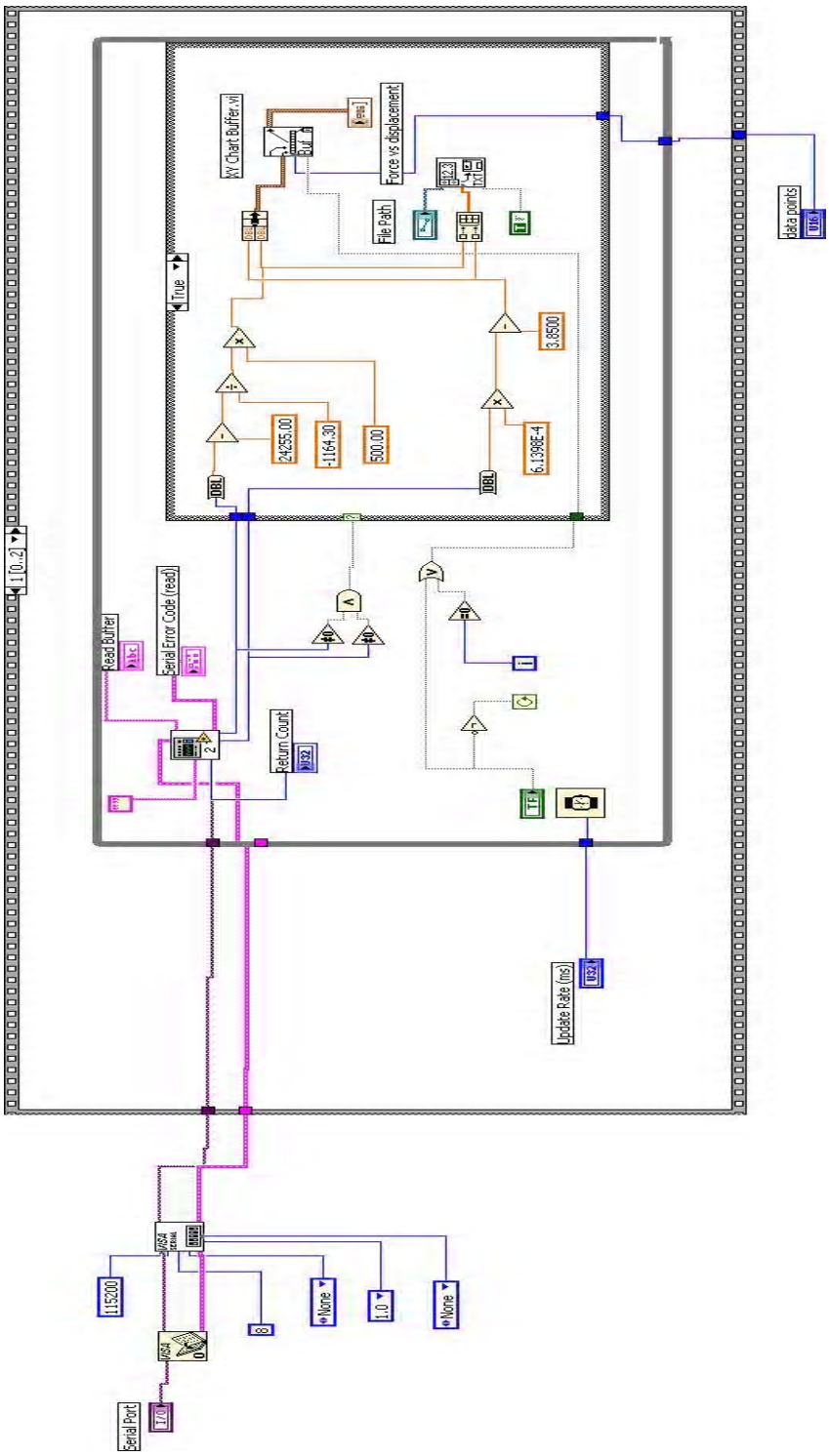


**Figure 3.10:** The main frame of the VI code.

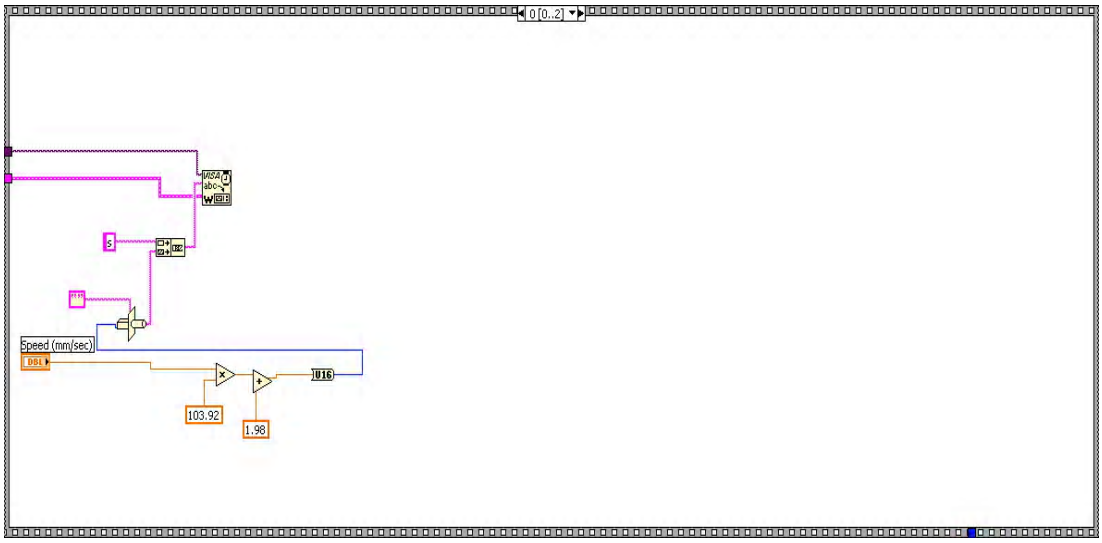


**Figure 3.11:** The first frame of the VI code.

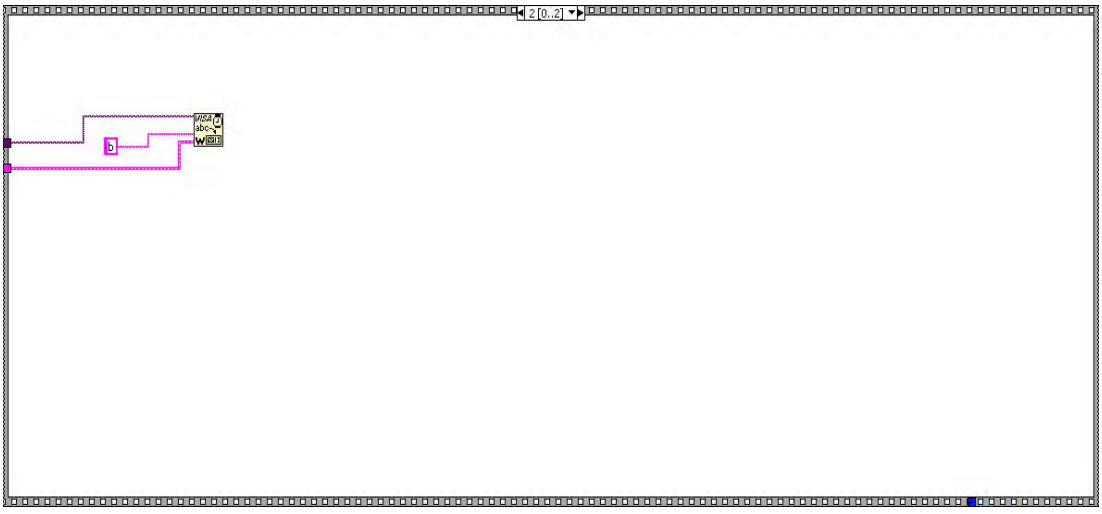


**Figure 3.12:** The last frame of the VI code.

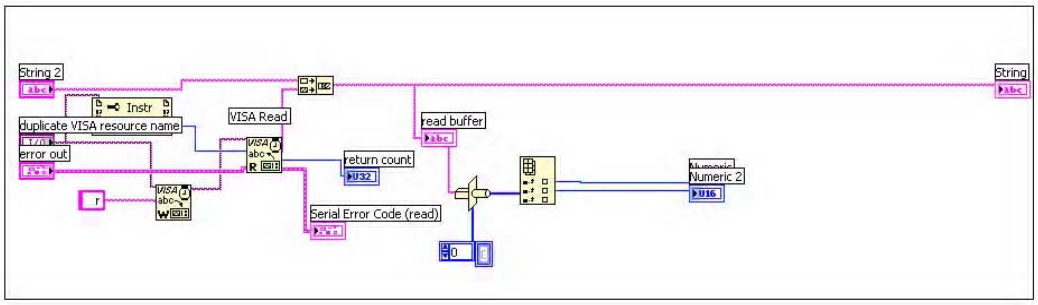


**Figure 3.13:** The detail of the code symbolized by the SubVI2.

versus displacement graph, and is written in a separate file for later analysis.

Pressing the Rewind button, the VI code jumps into the last frame in Figure 3.12 and writes b command into the microcontroller. The microcontroller sends a signal to run the DC motor in the reverse direction to bring the platform back to its original place.

In this design, the user is able to change the speed of the motor and the sampling rate of data collection, shown as the update rate in Figure 3.9. The serial error functions of LabVIEW have been used to alert the user in case of an error that happens during the execution of the VI code.

# Chapter 4

# EXPERIMENTATION

## 4.1 Sensor Calibrations

Due to the use of both analog and digital types of devices in the instrumentation, the sensor data has been converted from continuous to digital form. This requires the calibration of the force and displacement sensor reading as well as the input speed.

The sensitivity of the force sensor is 20mV/V [8]. Since it has been excited by 5V DC voltage, the maximum voltage read by the sensor is 100 mV. Therefore, known voltages in millivolt level have been applied for the calibration. Table 4.1 shows the force calibration values.

Using the values in the table, the force calibration curve is plotted in Figure 4.1. The force calibration equation reads,

$$Y_{FS} = -1164.3V_F + 24255, \tag{4.1}$$

where $V_F$ is the known voltage applied to the force sensor and $Y_{FS}$ is the sensor output. The accuracy of the force sensor calibration is calculated to be 99.85%.

Based on the calibration equation and the upper limit force reading, which is 50 gram, the VI code is modified to read force measurements in miligram.

The force calibration has been done, while all the buttons of the variable resistor discussed in Section 3.1 are On (1). Therefore, when testing materials at a

**Table 4.1:** Force calibration data.

| Voltage (mv) | Sensor Output |
|---|---|
| 0.355 | 23861.88 |
| 0.768 | 23337.96 |
| 1.530 | 22436.45 |
| 2.095 | 21810.25 |
| 2.832 | 20939.45 |
| 3.010 | 20765.59 |
| 4.228 | 19327.32 |
| 5.058 | 18359.55 |
| 6.035 | 17314.28 |
| 10.085 | 12527.06 |
| 13.352 | 8670.25 |

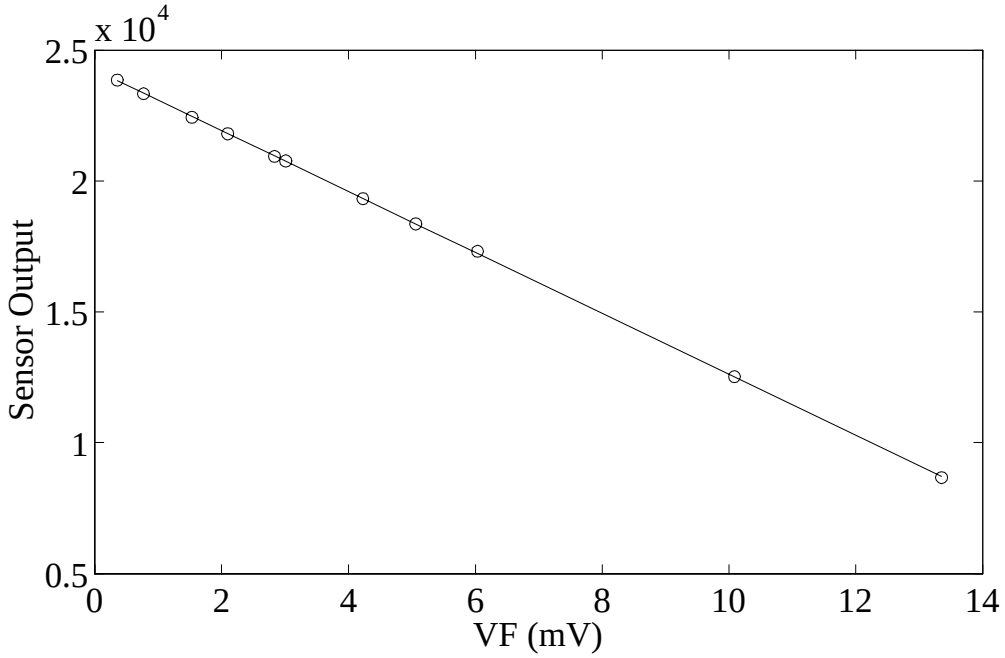


**Figure 4.1:** Force calibration curve.

**Table 4.2:** Instrumentation amplifier gain-set data.

| Switches | Gain |
|---|---|
| 0000 | 7.712 |
| 1000 | 14.431 |
| 0100 | 21.635 |
| 1100 | 28.353 |
| 0010 | 32.892 |
| 1010 | 39.594 |
| 0110 | 46.783 |
| 1110 | 53.497 |
| 0001 | 62.827 |
| 1001 | 69.516 |
| 0101 | 76.651 |
| 1101 | 83.333 |
| 0011 | 87.819 |
| 1011 | 94.720 |
| 0111 | 101.899 |
| 1111 | 108.602 |

different set of buttons, the resistance measured across the instrumentation amplifier pins, therefore, the gain changes. The gain is calculated by

$$Gain = \frac{49.4(KOhm)}{R(KOhm)} + 1, \tag{4.2}$$

where $R$ is the measured resistance [3]. The gain for each combination of the buttons is calculated to be in Table 4.2.

In the displacement calibration, the initial and final locations of the platform are measured in ten experiments as shown in Table 4.3.

Using the values in the table, the relationship between the sensor output and the measured displacement is found. The average slope is taken as the displacement calibration coefficient. Repeating the experiments several times to find the number for displacement readings to start from zero, we obtain the calibration equation

$$Y_{DS} = 1629L_D + 6270, \tag{4.3}$$

**Table 4.3:** Displacement calibration data.

| Initial | | Final | |
|---|---|---|---|
| Distance (mm) | Sensor Output | Distance (mm) | Sensor Output |
| 27.00 | 6119 | 13.16 | 28700 |
| 27.00 | 6119 | 13.26 | 28731 |
| 27.10 | 6196 | 13.17 | 28731 |
| 27.12 | 6146 | 13.25 | 28768 |
| 27.18 | 5908 | 13.30 | 28783 |
| 27.12 | 5980 | 13.25 | 28673 |
| 27.14 | 6061 | 13.18 | 28735 |
| 27.14 | 6025 | 13.02 | 28601 |
| 27.10 | 6032 | 13.20 | 28742 |
| 27.04 | 6176 | 13.14 | 28721 |

where $L_D$ is the displacement of the platform and $Y_{DS}$ is the output in the VI code. The accuracy of the displacement sensor calibration is calculated to be 98.02%.

For the calibration of speed, we have made six experiments in a range of 0 to $2^{10}$ at 80, 100, 150, 200, 300 and 400 voltages in digital form. This range is the maximum range of voltage that can be applied to change the speed of the DC motor in the design. This is a unitless range defined for the microcontroller to understand. The speed corresponding to each voltage is found and listed in Table 4.4. Using the data obtained, the speed calibration curve is plotted in Figure 4.2. The speed calibration equation reads

$$Y_E = 9.6x10^{-3}Y_V - 0.019, \tag{4.4}$$

where $Y_E$ is the corresponding speed in mm/sec and $Y_V$ is the voltage. The accuracy of the speed calibration is calculated to be 97.65%.

## 4.2 Experimental Procedures on Nylon Fibers

Fibers to be tested are prepared using aluminum covered paper frames. For the test of nylon fibers, a single fiber from a bundle is glued to the frame shown

**Table 4.4:** Speed calibration data.

| Speed (mm/sec) | Voltage |
|---|---|
| 0.787 | 80 |
| 0.891 | 100 |
| 1.413 | 150 |
| 1.924 | 200 |
| 2.908 | 300 |
| 3.799 | 400 |

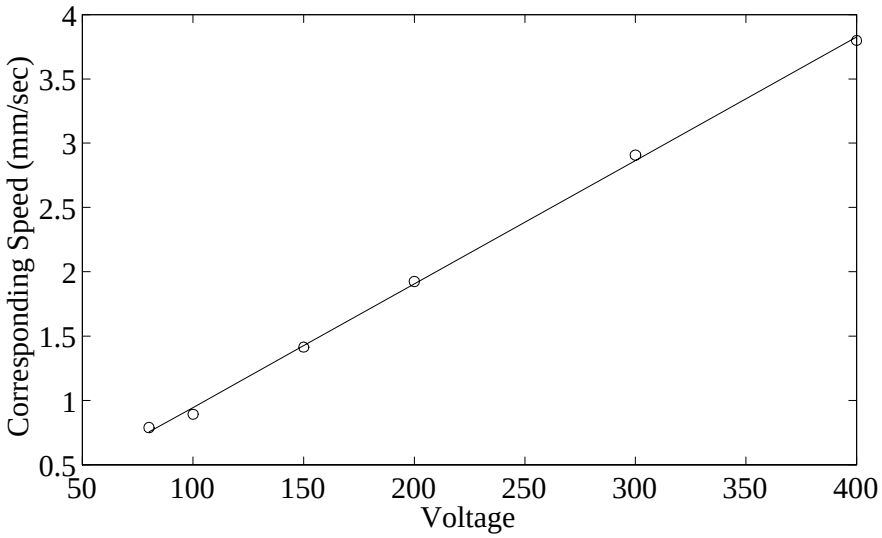


**Figure 4.2:** Speed calibration curve.

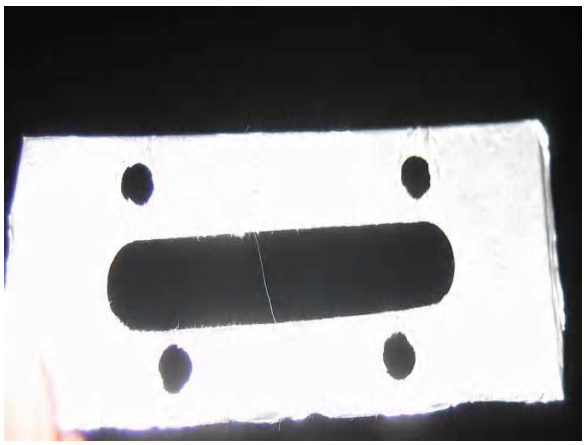

**Figure 4.3:** A single fiber test specimen.

in Figure 4.3. Removing slack without stretching the fiber is important when gluing. After fixing the frame to the moving plates, we cut both sides of the frame perpendicular to the fiber. We are ready to cover the tester and start pulling the fiber. The VI code is launched and the test speed is set. With the movement of the platform, fibers are stretched until they get broken. The force versus displacement curve seen in Figure 4.4 is automatically plotted on the computer screen.

### 4.3 Mechanical Testing of Nylon Fibers

Since the fiber bundles are much stronger than the single fibers, the breaking force of a bundle exceeds the maximum force reading of the force sensor. Therefore, a single fiber from a bundle is tested in this study.

Twenty samples are tested using the tensile tester at the standard atmosphere conditions. The fibers are stretched at a rate of 1 mm/sec. According to the ASTM D3822, Standard Test Method for Tensile Properties of Single Textile Fibers, elongation at break, $\lambda$, is defined as

$$\lambda = \frac{\Delta L}{L_0}, \tag{4.5}$$

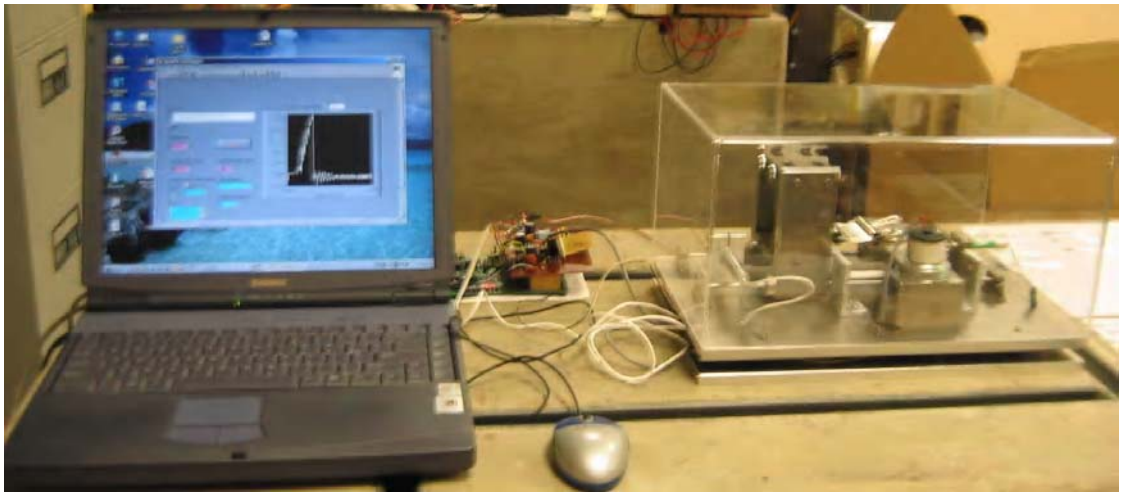

**Figure 4.4:** Force versus displacement curve plotted automatically by the VI code.

where $\Delta L$ is the displacement and $L_0$ is the original length of the fiber. Here, $\Delta L$ is the displacement at the breaking point.

In the ASTM D3822 Standard, initial modulus, $E$, is defined as the slope of the initial straight portion of the force-elongation curve,

$$E = \frac{F}{\lambda}, \tag{4.6}$$

where $F$ is the tensile force exerted on the fiber.

Table 4.5 shows the breaking force and the calculated values of elongation at break and initial modulus for each experiment. Figures 4.5, 4.6, and 4.7 show examples of experimental force-elongation curves of single nylon fiber.

## 4.4 Discussion

Based on the average increments in force and displacement measurements obtained in twenty experiments, the resolutions of force and displacement sensors are calculated to be in Table 4.6.

**Table 4.5:** Single nylon fiber experimental data resulting breaking force of $133.189 \pm 53.503mg$, elongation of $51.7 \pm 20\%$, and modulus of $617.1 \pm 194.2mg$.

| Force (mg) | Elongation (%) | Modulus (mg) |
|---|---|---|
| 88.368 | 44.2 | 704.2 |
| 133.302 | 46.6 | 839.1 |
| 117.324 | 58.6 | 464.8 |
| 119.436 | 80.1 | 470.2 |
| 86.966 | 25.5 | 640.1 |
| 87.593 | 44.2 | 674.7 |
| 133.302 | 46.0 | 807.4 |
| 117.324 | 57.4 | 546.6 |
| 117.838 | 80.9 | 572.8 |
| 86.966 | 25.5 | 632.6 |
| 116.525 | 99.9 | 193.8 |
| 103.971 | 33.7 | 560.8 |
| 123.829 | 35.1 | 649.3 |
| 115.039 | 54.3 | 196.3 |
| 253.023 | 48.8 | 867.8 |
| 115.897 | 29.2 | 604.0 |
| 123.829 | 45.6 | 703.8 |
| 252.386 | 48.6 | 845.8 |
| 117.838 | 80.4 | 489.5 |
| 253.023 | 48.4 | 879.4 |

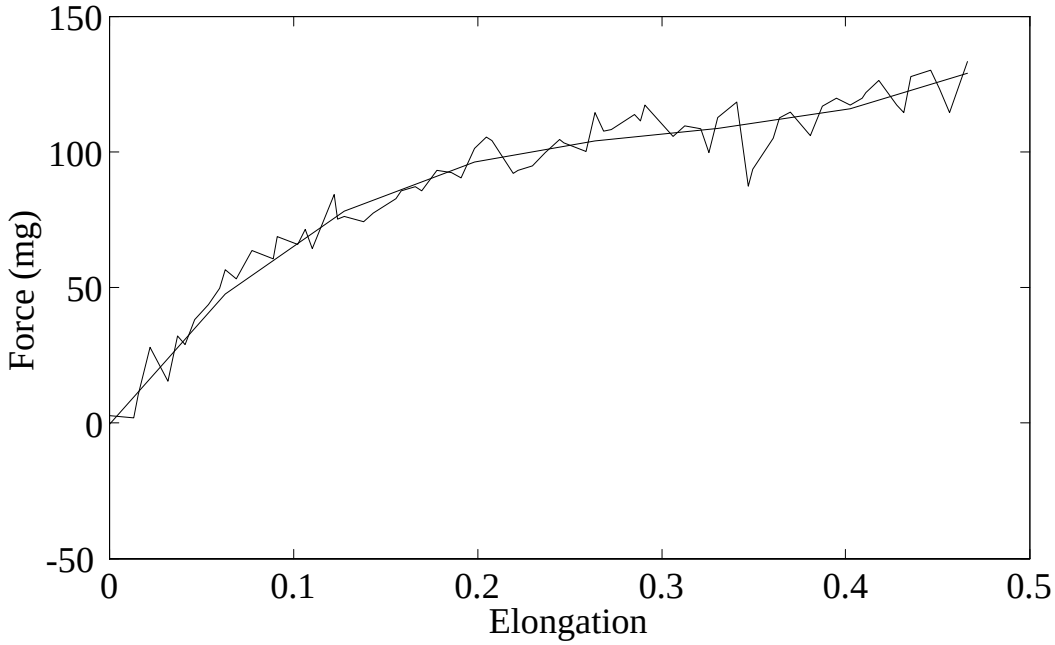


**Figure 4.5:** Single nylon fiber experimental force-elongation curve, I.

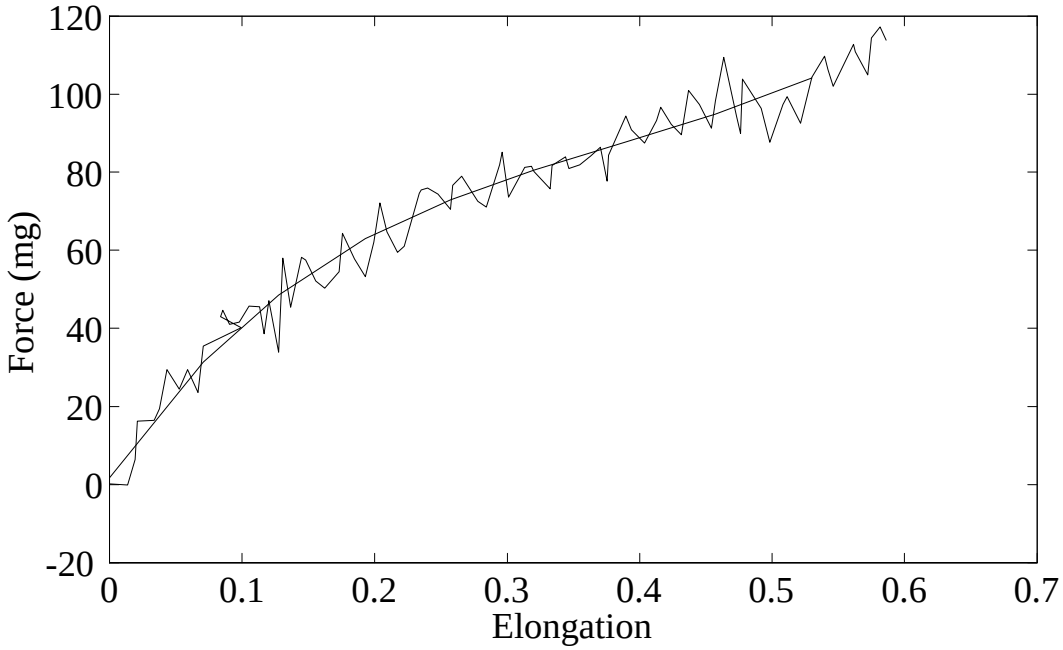


**Figure 4.6:** Single nylon fiber experimental force-elongation curve, II.

**Table 4.6:** Increment in force and displacement measurements leading to resolutions of $7.676 \pm 1.510mg$ and $43.556 \pm 2.175\mu m$.

| Force (mg) | Displacement ($\mu m$) |
|---|---|
| 6.214 | 48.318 |
| 6.612 | 42.271 |
| 5.917 | 44.056 |
| 8.277 | 42.633 |
| 10.764 | 39.537 |
| 7.247 | 42.530 |
| 7.735 | 41.743 |
| 6.248 | 41.420 |
| 9.389 | 43.202 |
| 10.973 | 39.537 |
| 7.545 | 44.697 |
| 6.933 | 44.269 |
| 7.247 | 43.686 |
| 7.202 | 44.465 |
| 8.613 | 46.239 |
| 5.236 | 43.130 |
| 6.693 | 41.986 |
| 7.693 | 46.060 |
| 9.163 | 45.408 |
| 7.818 | 45.925 |

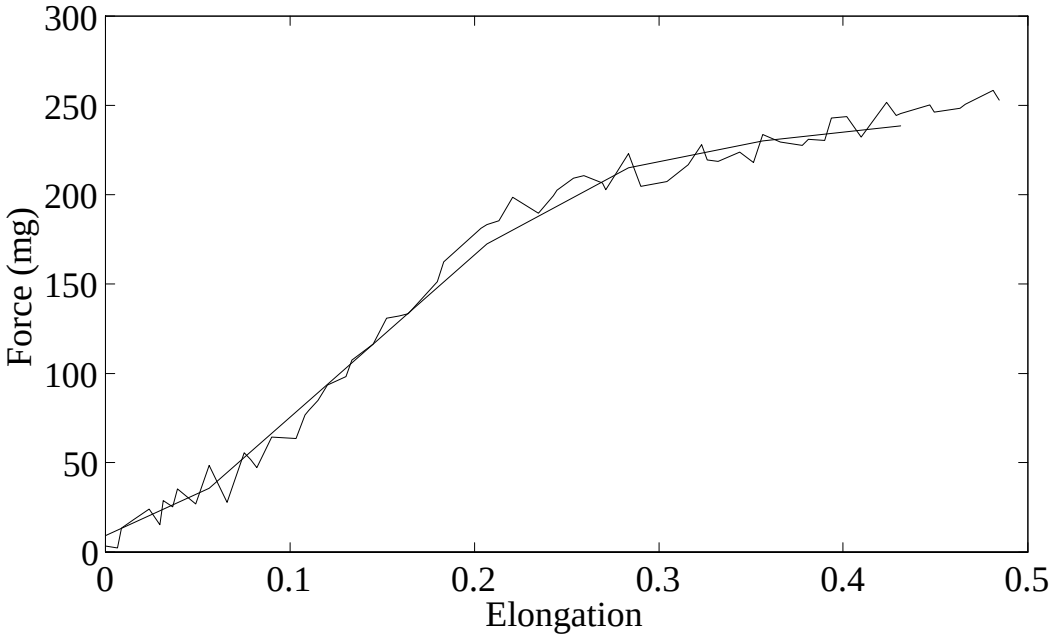


**Figure 4.7:** Single nylon fiber force-elongation curve, III.

# Chapter 5

# SUMMARY AND FUTURE WORK

## 5.1 Summary

With a rapid increase in the use of very small scale fibers in many different industries, there is a great need for testing these materials for their mechanical properties.

This work in mechanical testing has focused on building a small and accurate fiber tensile tester for microscale measurements. It is a unique fully-automated design. This machine has been built according to the ASTM D76 "Standard Specification for Tensile Testing Machines for Textiles".

The machine is composed of a force sensor and a displacement sensor. The instrumentation mainly includes a permanent magnet DC motor, a microcontroller (MC) and complementary electrical circuit elements. The control software has been written dealing with the analog to digital conversion, pulse width modulation (PWM) and serial communication features of the microcontroller. A PID control has been designed and implemented for regulating the speed of fiber testing.

The tester is rigorously calibrated. Single nylon fibers have been tested for their mechanical properties.

The tensile tester is a cost effective design for testing of small scale fibers. Testing of one specimen takes less than a minute using this machine. Many repetitive tests can be carried out for the mechanical properties of new small scale fibers effectively.

## 5.2 Future Work

Currently, the fiber frame is clamped and placed on the tester manually. This can be made automatically by a mechanism added to the system.

With some modification to the machine such as controlling the testing temperature, pressure or humidity, effect of external conditions on bio-fibers may also be studied for future work.

# BIBLIOGRAPHY


[1] *L6202 Dmos Full Bridge Driver Data Sheet.* SGS-Thomson Microelectronics, Inc., Italy, 1997.

[2] *LabVIEW User Manual.* National Instruments Corp., Austin, Texas, 1998.

[3] *AD620 Instrumentation Amplifier Data Sheet.* Analog Devices, Inc., Norwood, Massachusetts, 1999.

[4] ASTM D76-99: Standard Specification for Tensile Testing Machines for Textiles. 1999.

[5] ASTM D3822-01: Standard Test Method for Tensile Properties of Single Textile Fibers. 2001.

[6] ASTM D638-01: Standard Test Method for Tensile Properties of Plastics. 2001.

[7] *MC9S08GB/GT Microcontrollers Data Sheet.* Freescale Semiconductor, Inc., Chandler, Arizona, 2004.

[8] *SLCB-50 Mini-gram Beam Type Load Cell Data Sheet.* Load Cell Central, Monroeton, Pennsylvania, 2004.

[9] D. Barnaal. *Analog Electronics for Scientific Application.* Waveland Press, Inc., Prospect Heights, Illinois, 1989.

[10] I. S. C. Benloulo, J. Rodriguez, M. A. Martinez, and V. S. Galvez. Dynamic tensile testing of aramid and polyethylene fiber composites. *International Journal of Impact Engineering*, 19:135–146, 1997.

[11] J. S. Bergstrom and M. C. Boyce. Deformation of elastomeric networks: Relation between molecular level deformation and classical statistical mechanics models of rubber elasticity. *Macromolecules*, 34:616–626, 2000.

[12] M. Bognitzki, W. Czado, T. Frese, A. Shchaper, M. Hellwig, M. Steinhart, A. Greiner, and J. H. Wendroff. Nanostructured fibers via electrospinning. *Advanced Materials*, 13:70–72, 2001.

[13] D. Brown and J. H. R. Clarke. Molecular dynamics simulation of an amorphous polymer under tension 1. phenomenology. *Macromolecules*, 24:2075–2082, 1991.

[14] J. G. Cook. *Handbook of Textile Fibers:Man-Made Fibres*. Merrow Publishing Co. Ltd., Durham, England, 1984.

[15] J. G. Cook. *Handbook of Textile Fibres:Natural Fibres*. Merrow Publishing Co. Ltd., Durham, Englad, 1984.

[16] B. Ding, H. Y. Kim, S. C. Lee, C. L. Shao, D. R. Lee, S. J. Park, G. B. Kwag, and K. J. Choi. Preparation and characterization of a nanoscale poly(vinylalcohol) fiber aggregate produced by an electrospinning method. *Journal of Polymer Science Part B-Polymer Physics*, 40:1261–1268, 2002.

[17] M. A. Elbestawi. *Force Measurement*. CRC Press LLC, Inc., Boca Raton, Florida, 2000.

[18] R. S. Figliola and D. E. Beasley. *Theory and Design for Mechanical Measurements*. John Wiley and Sons, Inc., Toronto, 1995.

[19] M. Harris. *Handbook of Textile Fibers*. Harris Research Laboratories, Inc., Washington, District of Columbia, 1954.

[20] http://www.fibersource.com/Fiber.html. American Fiber Manufacturers Association. 2005.

[21] K. H. Lee, H. Y. Kim, M. S. Khil, Y. M. Ra, and D. R. Lee. Characterization of nano-structured poly(epsilon-caprolactone) nonwoven mats via electrospinning. *Polymer*, 44:1287–1294, 2003.

[22] A. Nechwatal, K. Mieck, and T. Reussmann. Developments in the characterization of natural fibre properties and in the use of natural fibres for composites. *Composites Science and Technology*, 63:1273–1279, 2003.

[23] B. E. Paton. *Sensors, Transducers and LabVIEW*. Prentice-Hall, Inc., New Jersey, 1999.

[24] G. D. Pins, E. K. Huang, D. L. Christiansen, and F. H. Silver. Effects of static axial strain on the tensile properties and failure mechanisms of self-assembled collagen fibers. *Journals of Applied Polymer Science*, 63:1442–1448, 1997.

[25] J. P. Rigueiro, C. Viney, J. Llorca, and M. Elices. Mechanical properties of single-brin silkworm silk. *Journal of Applied Polymer Science*, 75:1270–1277, 1999.

[26] J. Sandler, A. H. Windle, P. Werner, V. Altstadt, M. V. Es, and M. S. P. Shaffer. Carbon-nanofiber-reinforced poly(ether ether ketone) fibres. *Journal of Materials Science*, 38:2135–2141, 2003.

[27] A. Seidel, O. Liivak, S. Calve, J. Adaska, G. Ji, Z. Yang, D. Grubb, D. B. Zax, and L. W. Jelinski. Regenerated spider silk: processing, properties, and structure. *Macromolecules*, 33:775–780, 2000.

[28] D. Shetty and R. A. Kolk. *Mechatronics System Design*. PWS Publishing Company, Inc., Boston, Massachusetts, 1997.

[29] F. M. Shofner. Apparatus and methods for testing tension-elongation or cross-sectional properties of single fibers and multiple fiber bundless. *U.S. Patent 5167150*, 1992.

[30] F. M. Shofner. Apparatus and methods for testing tension-elongation or cross-sectional properties of single fibers and multiple fiber bundless. *U.S. Patent 5203206*, 1993.

[31] E. P. S. Tan, S. Y. Ng, and C. T. Lim. Tensile testing of a single ultrafine polymeric fiber. *Biomaterials*, 26:1453–1456, 2005.

[32] Y. Termonia and P. Smith. Kinetic-model for tensile deformation of polymers.5. effect of temperature on orientation efficiency. *Macromolecules*, 26:3738–3741, 1993.

[33] P. Verleysen and J. Degrieck. Measurement of the evolution of the axial strain distribution in hopkinson specimens. *Journal of Physics*, 110:501–506, 2003.

[34] J. Wang and K. J. J. Smith. The breaking strength of ultra-high molecular weight polyethylene fibers. *Polymer*, 40:7261–7274, 1999.

# Appendix A

# THE MECHANICAL COMPONENTS AND DRAWINGS

**Misumi Parts**

| Part Name | Quantity |
|---|---|
| SFJ5 – 288<br>Linear Shaft | 2 |
| SHPTM5 – 15<br>Shaft Holder | 4 |
| SHPTM8 – HC35<br>Shaft Holder<br>(Potentiometer Shaft Holder) | 1 |
| SHKSA8 – HC38<br>Shaft Holder<br>(Force Sensor Shaft Holder) | 1 |

| | |
|---|---|
| KPAAT5 –A50–B25–L30–X15–<br>Y36–F8–G16–V14–D8–NA4<br>Work Plate<br>(Stop Switcher Shaft Holder) | 2 |
| LHSSWS5<br>Linear Bushing<br>(Platform Bushings) | 2 |
| LMU5<br>Linear Bushing | 4 |
| SCA – 130-60-15-BC45<br>Guide Plate<br>(Platform, Figure A.1) | 1 |
| RDOC3 – 20<br>Rotary Shaft | 1 |
| RDOC3 – 25<br>Rotary Shaft | 1 |
| SCA – 58-60-10-BC55<br>Guide Plate<br>(Hanging Plate, Figure A.2) | 1 |

| | |
|---|---|
| SSA – 65-60-19-BC50<br>Guide Plate<br>(Fixed Plate, Figure A.3) | 1 |
| ALNS - 83-50-15<br>Aluminum Plate<br>(Bridge Support, Figure A.4) | 2 |
| RDJK10 – 60-MC4<br>Rotary Shaft<br>(Resin Rod, Figure A.5) | 2 |
| PSCS5 – 8<br>Set Collar | 2 |
| CLJJ10 – 12-10<br>Washer | 4 |
| NOB4 – 6<br>Knureing Knobs | 2 |
| SSFB – 60-40-46.3<br>Steel Block<br>(Motor Mounting Block, Figure A.6) | 1 |

| | |
|---|---|
| ALA – 345-250-10-BC222 | 2 |
| Guide Plate | |
| (Base Plates, Figures A.7 and A.8) | |

**LordMpd Parts**

| | |
|---|---|
| SMB003 – 0200-9 | 4 |
| Small Sandwich Mounts | |
| (Vibration Isolator) | |

**Sterling Instrument Parts**

| | |
|---|---|
| A 1Z 2MYZ0502204 | 1 |
| Pinion | |
| (4.75 mm Press Fit Bore Diameter) | |
| | |
| A 1M12MYZ0525A | 1 |
| Rack | |
| (90 mm long, Figure A.9) | |

**Potomac Display Parts**

| | |
|---|---|
| 345x222x175x3 mm | 1 |
| Acrylic Display Case | |

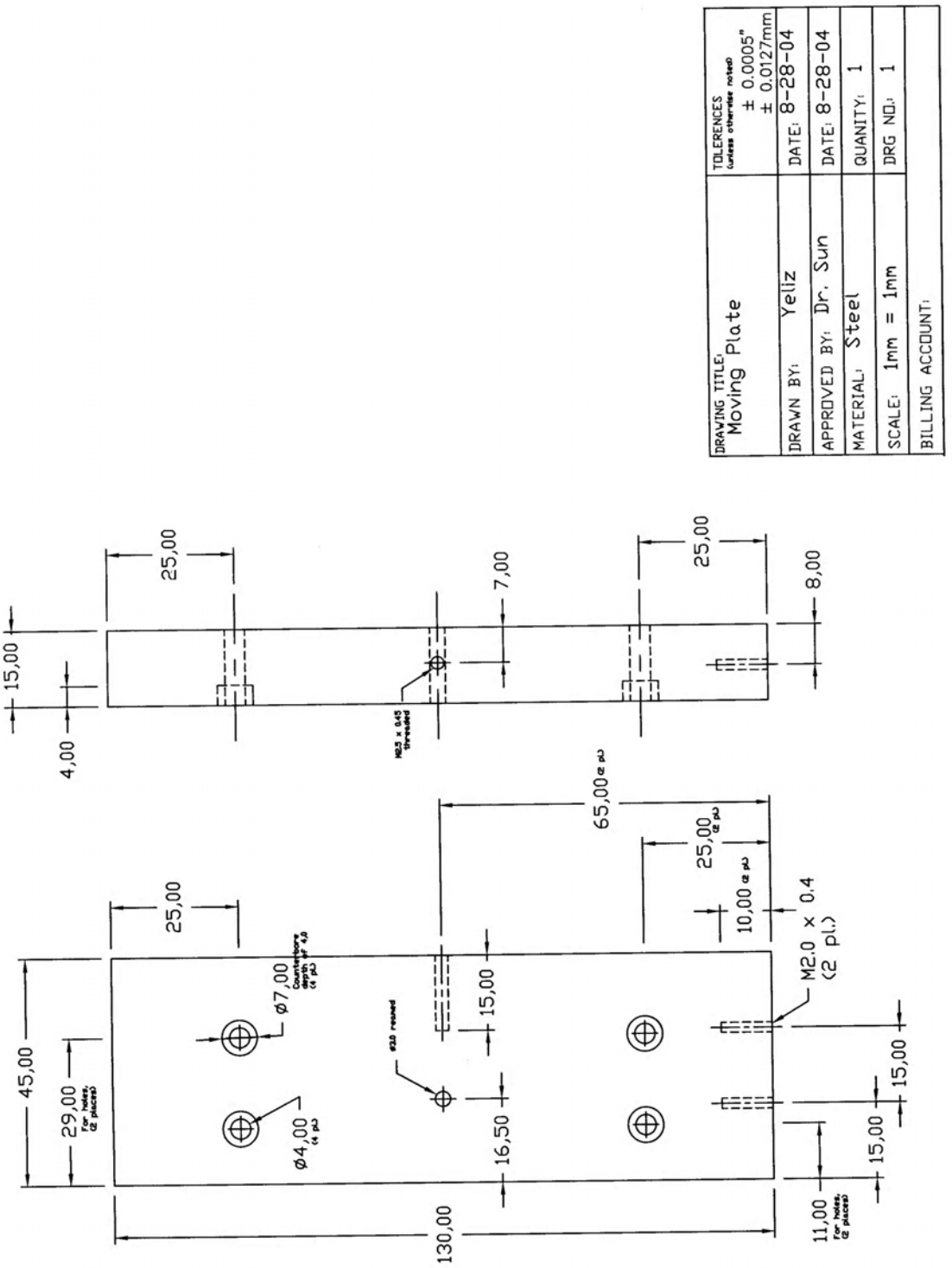


**Figure A.1:** Platform (Moving plate)

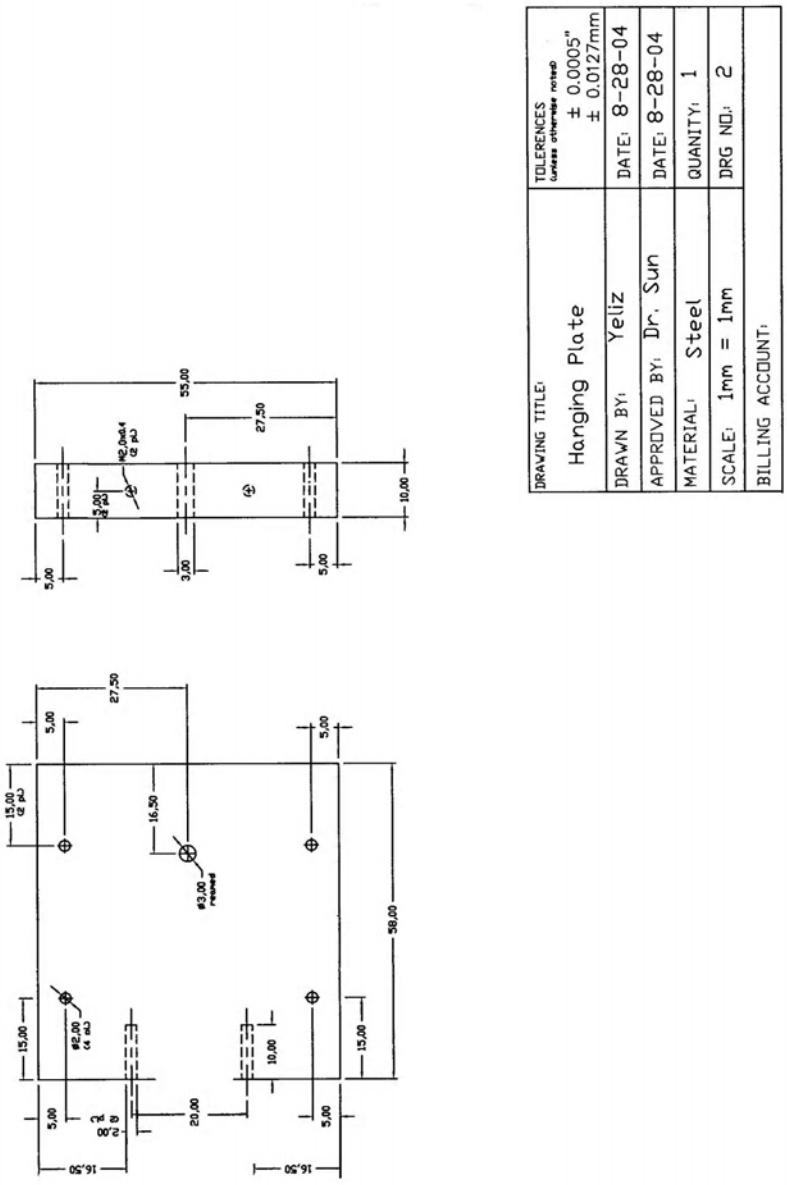


**Figure A.2:** Hanging plate

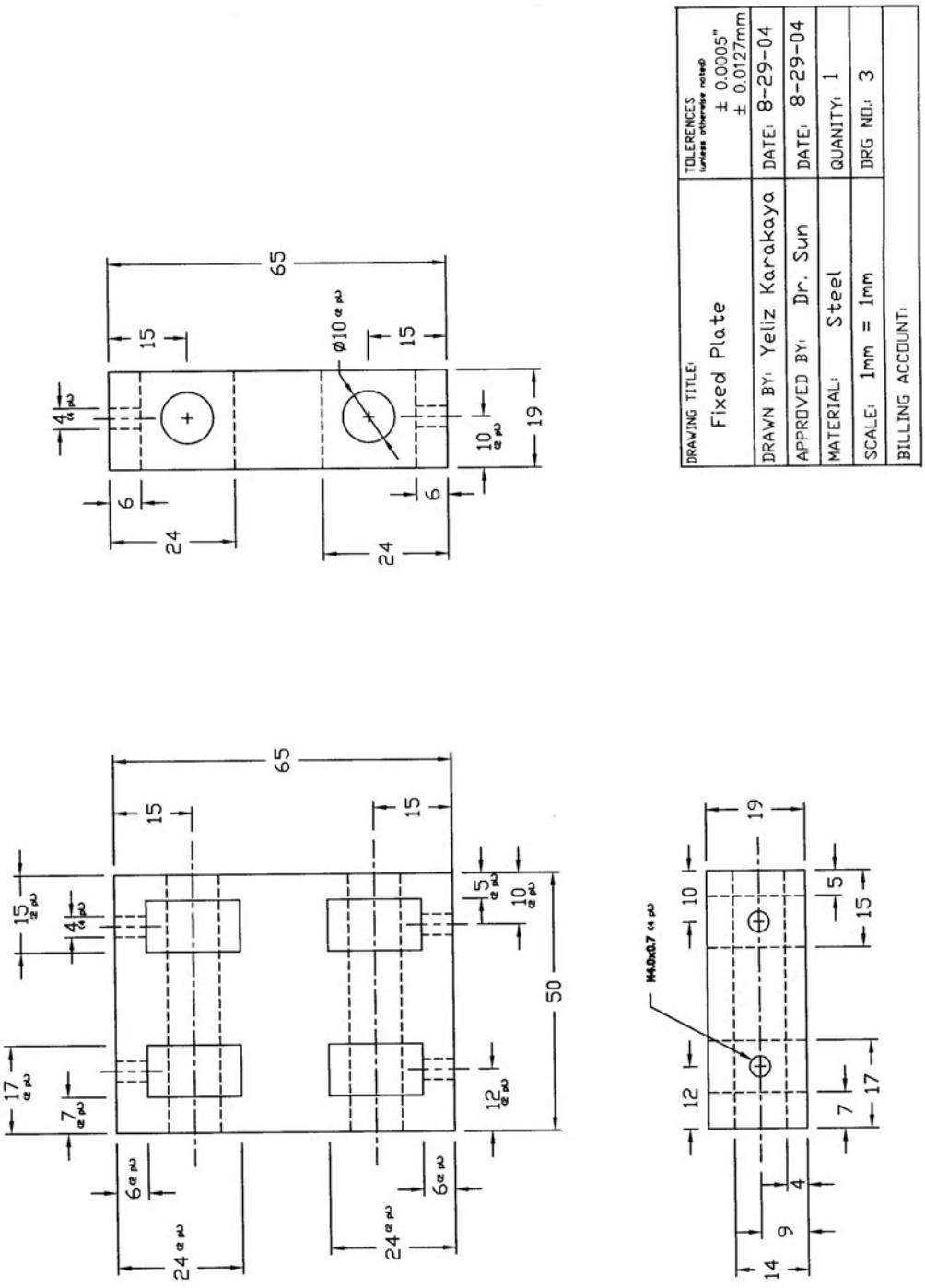


**Figure A.3:** Fixed plate

**Figure A.4:** Bridge supports

**Figure A.5:** Resin rods

**Figure A.6:** Mounting block

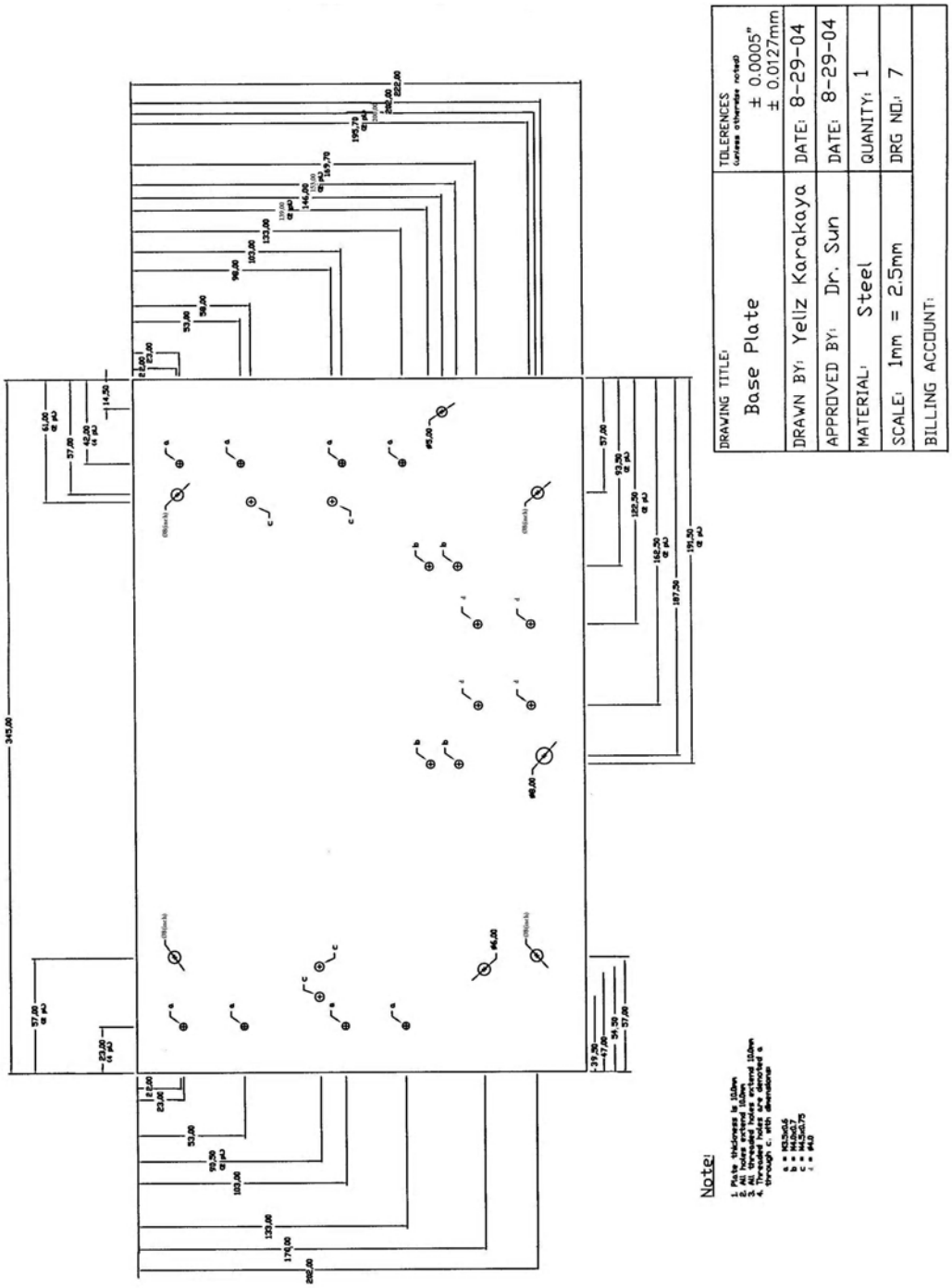


**Figure A.7:** Base plate

**Figure A.8:** Bottom base plate

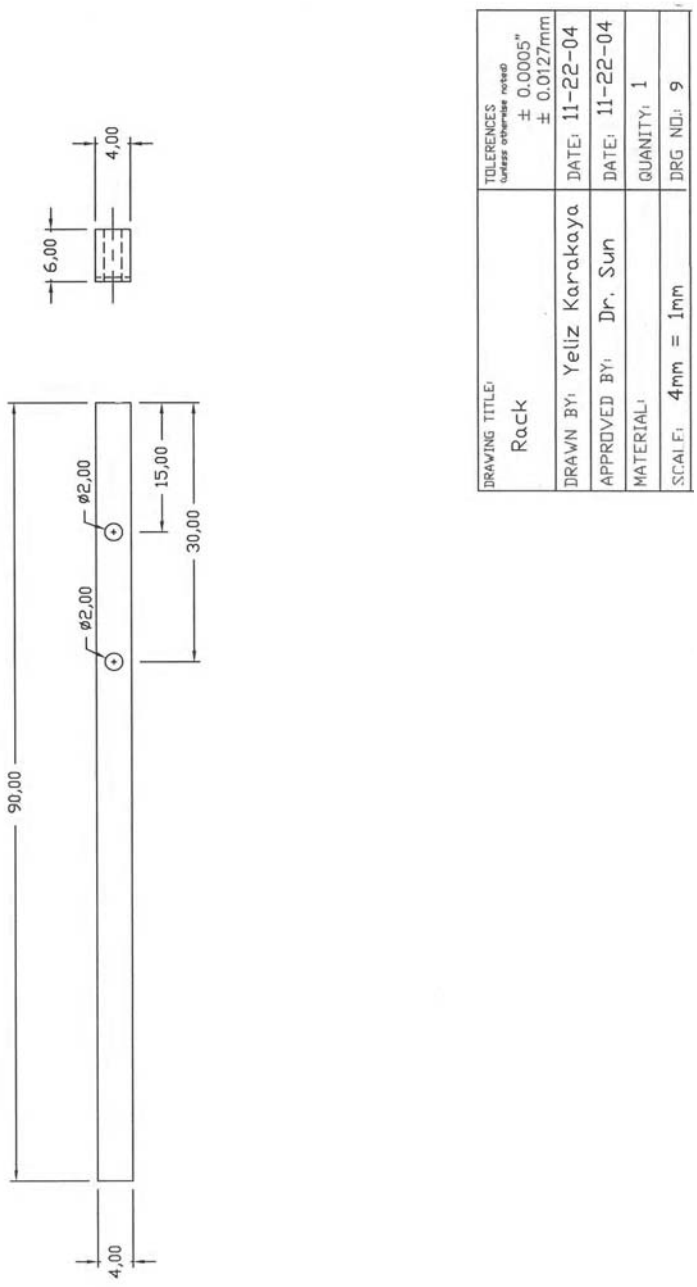


**Figure A.9:** Rack

# Appendix B

# THE ELECTRICAL COMPONENTS

**Digikey Parts**

| Part Name | Quantity |
|---|---|
| M68DEMO908GB60-ND<br>Demo Board for Motorola MC9S08GB60 | 1 |
| ADR 423AR – ND<br>Voltage Reference | 1 |
| AD620 – ND<br>Instrumentation Amplifier | 1 |
| 497 – 1420-5-ND<br>L6202 H-Bridge, Motor Driver | 1 |
| OP184FS – ND<br>Differential Op-Amp | 1 |
| LM317TNS-ND<br>Variable Regulator | 1 |

LM78M05CTNS-ND 1

Voltage Regulator

LM6361N-ND 1

Op-Amp

CT2064-ND 1

Variable Resistor Switch

**Misumi Parts**

MSN – S 2

Stopper Switch

**Midori USA Parts**

LP-20FQ 1

Displacement Sensor (Linear Potentiometer)

**Load Cell Center Parts**

SLCB-50 1

Force Sensor (Bending Beam Type)

**Cramer Parts**

800HN 1

Permanent Magnent DC Motor

# Appendix C

# THE MAIN CODE

**Main File**

```
        XDEF Entry


        ABSENTRY Entry
        Include 'GB60_registers.inc'    ; For the HCS908GB60
NULL                                    equ 0
buf_n_samples                           equ 4;
bytes_per_sample                        equ 4;
TX_buf_size             equ buf_n_samples * bytes_per_sample
timer_modh                              equ $08
timer_modl                              equ $00
sci_modh                                equ $00
sci_modl                                equ $0A
AVG_CNT_MAX                             equ 32      ; 5 bits
safety_limit_hi             equ 980*AVG_CNT_MAX
safety_limit_lo             equ 180*AVG_CNT_MAX
pos_limit_hi                  equ 900*AVG_CNT_MAX
pos_limit_lo                  equ 200*AVG_CNT_MAX
REV_FAST                equ 800
REV_SLOW                 equ 300
```

```
slow_range          equ 1000
stop_range          equ 30
cmd_rewind          equ 0
cmd_wait            equ 1
cmd_getS0           equ 2
cmd_getS1           equ 3
cmd_run             equ 4
cmd_stop            equ 5
pid_P equ 10000     ; signed fixed point numbers! xx.yy
pid_I equ 10
pid_D equ 0
pid_IMAX equ 500
MOTOR_MAX equ $7FFF
MOTOR_MIN equ $8000
ORG ZPRamStart
avg_pos             ds.w 1      ; averaged [position | force] samples
avg_force           ds.w 1
avg_cnt     ds.b 1      ; how many samples have been averaged?
TX_buf_in       ds.w 1      ; FIFO pointers
TX_buf_out      ds.w 1
TX_buf_tmp      ds.w 2      ; temporary transmit space
TX_state            ds.b 1
cmd_state           ds.b 1
cmd_speed           ds.w 1      ; controls speed, set by labview
set_speed           ds.w 1      ; controls speed, set by state controller
filter_in           ds.w 1
filter_tmp          ds.w 1
```

```
filter_out              ds.w 1
pid_int              ds.w 1
pid_diff              ds.w 1
lastpos              ds.w 1
; math routine I/O registers
in1              ds.w 1
in1b equ in1+1
in2              ds.w 1
in2b equ in2+1
outA              ds.w 1
outB equ outA+1
outC              ds.w 1
outD equ outC+1
     ORG RamStart
TX_buffer       ds TX_buf_size ; 3kbytes of ram (1000 samples worth)
ORG RomStart
;*****************************************************************
;* Entry - This is the point where code starts executing *
;* after a RESET. *
;*****************************************************************
Entry:
main:
     jmp      App      ; skip identification data


;---------------------------------
init_buf:
ldhx #(TX_buffer)
```

```
sthx TX_buf_in
clrh
clrx
sthx TX_buf_out
rts
;--------------------------------
out_buf:
psha
ldhx TX_buf_out
beq ob_exit
pshh
pshx
aix #bytes_per_sample
cphx #(TX_buffer + TX_buf_size)
blo ob_nowrap
ldhx #(TX_buffer)
ob_nowrap:
cphx TX_buf_in
bne ob_done
ldhx #NULL
ob_done:
sthx TX_buf_out
pulx
pulh
lda TX_buf_in       ; if inbuf == NULL, inbuf=old outbuf
bne ob_exit
lda (TX_buf_in+1)
```

```
bne ob_exit
sthx TX_buf_in
ob_exit:
pula
rts
;---------------------------------------
in_buf:
psha
ldhx TX_buf_in
beq ib_exit
pshh
pshx
aix #bytes_per_sample
cphx #(TX_buffer + TX_buf_size)
blo ib_nowrap
ldhx #(TX_buffer)
ib_nowrap:
cphx TX_buf_out
bne ib_done
ldhx #NULL
ib_done:
sthx TX_buf_in
pulx
pulh
lda TX_buf_out        ; if inbuf == NULL, inbuf=old outbuf
bne ib_exit
lda (TX_buf_out+1)
```

```
bne ib_exit
sthx TX_buf_out
ib_exit:
pula
rts

;--------------------------
init_timer:
sta timer1.H ; resets counter
mov #timer_modh, timer1.modulo.H
mov #timer_modl, timer1.modulo.L
; TPM1 channel0 is PWM output
mov #%00101000, timer1.ch0.status ; high-true PWM
; TPM1 channel1 is set to a constant 50% duty cycle
; and generate interrupts.
mov #%01010000, timer1.ch1.status
lda #timer_modh ; divide timer-mod by 2
lsra
sta timer1.ch1.H
lda #timer_modl
rora
sta timer1.ch1.L
mov #%01001000, timer1.status ; activate timer
rts
;-----------------------------
init_AD:
; AD converter is enabled, but no conversion is started
```

```
mov #%11000100, AD.setup
mov #%11000000, portB.ADmode ; portB-6 and 7
mov #%00000000, AD.status ; do not activate AD yet
clrh      ; init averages and count
clrx
sthx avg_pos
sthx avg_force
mov #AVG_CNT_MAX, avg_cnt
rts
;-------------------------------------
init_clk:
mov #%00111000, clk.setup.1
mov #%01111000, clk.setup.2
mov #%00001000, clk.filter.L
mov #%11100001, clk.filter.H
mov #%10000000, clk.trim
rts
init_sci:
clrh
clrx
sthx cmd_speed ; setspeed is to zero
clr TX_state ; state = 0, nothing's happening
mov #cmd_rewind, cmd_state
mov #sci_modh, sci.ch1.modulo.H
mov #sci_modl, sci.ch1.modulo.L
mov #%00000000, sci.ch1.control.1
mov #%10101100, sci.ch1.control.2
```

```
mov #0, sci.ch1.control.3
rts

;---------------------------------
SCI_rx_int:      ; SCI received char interrupt
pshh
bset 4, portG
mov #$FF, portF
lda sci.ch1.status.1
lda sci.ch1.data
ldx cmd_state ; WAIT, RUN, STOP, or REWIND
cpx #cmd_wait
bne not_wait
cmpa #'r'
beq is_r
cmpa #'s'
bne rx_done
inc cmd_state
bra rx_done
not_wait:      ; RUN, STOP, or REWIND
cpx #cmd_getS0
bne not_getS0
sta cmd_speed
inc cmd_state
bra rx_done
not_getS0:
cpx #cmd_getS1
```

```
bne not_getS1
sta cmd_speed+1
inc cmd_state
bra rx_done
not_getS1:
cpx #cmd_run
bne not_run
cmpa #'r'
beq is_r
cmpa #'b'
bne rx_done ; abort cycle
mov #cmd_stop, cmd_state
bra rx_done
is_r:
mov #1, TX_state ; state = 1, send first AD byte
jmp txone
not_run:      ; STOP or REWIND
cpx #cmd_stop
bne rx_done
cmpa #'b'
bne rx_done
mov #cmd_rewind, cmd_state
bra rx_done
rx_done:
bclr 4, portG
pulh
rti
```

```
;---------------------------------------
SCI_tx_int:      ; SCI transmit char complete interrupt
pshh
bset 3, portG
lda sci.ch1.status.1
lda sci.ch1.status.2 ; touch the status registers
lda TX_state                ; switch (TX_state)
beq txzero                   ; case (0):
cmpa #1

beq txone                   ; case (1):
cmpa #2
beq txtwo                   ; case (2):
cmpa #3
beq txthree                   ; case (3):
cmpa #4
beq txfour                        ; case (4):
clr TX_state               ; default:
bra txzero
txone:                                  ; send count
bset 7, sci.ch1.control.2 ; activate transmitter
jsr out_buf
cphx #NULL
beq txnull
mov x+, TX_buf_tmp          ; load transmit buffer and go
mov x+, TX_buf_tmp+1
mov x+, TX_buf_tmp+2
```

```
mov x+, TX_buf_tmp+3
bra transmit
txnull:
sthx TX_buf_tmp                 ; null out transmit buffer
sthx TX_buf_tmp+2
transmit:
mov TX_buf_tmp+2, sci.ch1.data ; send word 2 high byte
mov #2, TX_state
bra txdone
txtwo:
mov TX_buf_tmp+3, sci.ch1.data ; send word 2 low byte
mov #3, TX_state
bra txdone
txthree:                        ; send word 1 high byte
mov TX_buf_tmp, sci.ch1.data
mov #4, TX_state
bra txdone
txfour:                         ; send word 1 low byte
mov TX_buf_tmp+1, sci.ch1.data
clr TX_state
txzero:
bclr 7, sci.ch1.control.2       ; unrequested interrupt, turn them off
txdone:
bclr 3, portG
pulh
rti
;-------------------------------
```

```
SCI_err_int:
bset 3, portG
bset 4, portG
jsr init_sci
bclr 4, portG
bclr 3, portG
rti
;--------------------------------
init_filter:
clrh
clrx
sthx set_speed
    sthx pid_int
    sthx pid_diff
    sthx filter_out

    ldhx pos_limit_lo
    sthx lastpos
rts
;--------------------------------
filter:
pshh
pshx
lda cmd_state
ldhx filter_in
cphx #safety_limit_hi ; safety limits, must not be exceeded
bhi do_emstop
```

```
cphx #safety_limit_lo
blo do_emstop
cmpa #cmd_rewind      ; in rewind state?
beq do_rewind
cmpa #cmd_run         ; in run state?
bne do_stop           ; no: stop
cphx #pos_limit_hi    ; did it reach upper limit?
blo do_run            ; no: continue running
mov #cmd_stop, cmd_state
bra do_stop           ; yes: stop
; ———————— state commands
do_emstop:
mov #%11110000, portF
clrh
clrx
sthx pid_int
sthx filter_out
bclr 1, portD
bra filter_done
do_rewind:
mov #%11111110, portF
bsr rewind
bra do_pid
do_run:
mov #%11111011, portF
ldhx cmd_speed
bra do_pid
```

```
do_stop:
mov #%11110111, portF
clrh
clrx
; sthx pid_int
bra do_pid
do_pid:
sthx set_speed ; setup PID controller, and..
jsr PID
brclr 7, filter_out, pidpos
pidneg:                         ; pid output is negative
bset 1, portD ; set motor to be negative speed
bclr 7, filter_out
bra pidout
pidpos: ; pid output is positive
bclr 1, portD
pidout:      ; shift right by 4 bits
clc

ror filter_out      ; 16-bit rotate
ror filter_out+1
clc
ror filter_out      ; 16-bit rotate
ror filter_out+1
clc
ror filter_out      ; 16-bit rotate
ror filter_out+1
```

```
clc
ror filter_out      ; 16-bit rotate
ror filter_out+1
filter_done:
pulx
pulh
rts
;————————————————————-
rewind:
mov #%11111101, portF
cphx #(pos_limit_lo-stop_range)          ; is it too far left?
blo rw_right                ; yes: go right slowly
cphx #(pos_limit_lo+slow_range) ; position nearing lower limit?
bhi rw_fast
cphx #(pos_limit_lo+stop_range)
bhi rw_slow
rw_done:                                ; position within dead gap?
mov #cmd_wait, cmd_state          ; yes: stop
ldhx #0
bra rw_end
rw_fast:                    ; no: rewind quickly
ldhx #-REV_FAST
bra rw_end
rw_slow:
ldhx #-REV_SLOW
bra rw_end
rw_right:
```

```
ldhx #+REV_SLOW
bra rw_end
rw_end:
rts
;-------------------------------
PID:;;;;;;;;;;;;;;;;;;;;;;;;;;;; PID CONTROLLER
clrh
clrx

sthx filter_out
ldhx lastpos                ; curspeed = filter_in - lastpos
sthx in2                    ; lastpos = filter_in
ldhx filter_in
sthx in1
sthx lastpos
bsr sisub222
sthx in2 ; err = setspeed - curspeed
ldhx set_speed
sthx in1
bsr sisub222
sthx filter_tmp
;;;;;;;;;;;;;;;;;;;;;;;;;;;; PROPORTIONAL
sthx in1            ; P = pid_P * err
ldhx #pid_P
sthx in2
bsr mul_acc
;;;;;;;;;;;;;;;;;;;;;;;;;;;; INTEGRAL
```

```
ldhx filter_tmp
sthx in1                ; pid_int = pid_int + err
ldhx pid_int
sthx in2
jsr siadd222
cphx #pid_IMAX
bge int_sat_hi
cphx #(-pid_IMAX)
ble int_sat_lo
iacc:
sthx pid_int
sthx in1                ; I = pid_I * pid_int
ldhx #pid_I
sthx in2
bsr mul_acc
;;;;;;;;;;;;;;;;;;;;;;;;;;;; DERIVATIVE
ldhx pid_diff
sthx in1
ldhx filter_tmp ; D = pid_D * (pid_diff - err)
sthx in2
bsr sisub222
sthx in1
ldhx #pid_D
sthx in2
bsr mul_acc
ldhx filter_tmp
sthx pid_diff ; pid_diff = pid_err
```

```
rts
int_sat_hi:
ldhx #pid_IMAX
bra iacc
int_sat_lo:
ldhx #(-pid_IMAX)
bra iacc
mul_acc: ; filter_out = filter_out + (in1 * in2) / 256
jsr smul224
ldhx outB ; the bytes in question
lda outA                ; test byte above them for overflow
beq accumulate          ; hi byte zero (positive no overflow)
inca
beq accumulate          ; hi byte all ones (negative no overflow)
brclr 7, outA, psatacc  ; MSB set means negative
nsatacc:
ldhx #$8000             ; saturate negative overflow
bra accumulate
psatacc:                ; saturate positive overflow
ldhx #$7FFF
bra accumulate
accumulate: ; filter_out = filter_out + (outB:outC)
sthx in1
ldhx filter_out
sthx in2
bsr siadd222
sthx filter_out
```

```
rts
susub222:
sisub222: ; int sisub222(int, int): outA = in1 - in2
bsr neg_in2
bsr siadd222
rts
neg_in2: ; in2 = -in2
ldhx in2
cphx #$8000
bne n2_doneg
ldhx #$7FFF
sthx in2
bra n2_done
n2_doneg:
clra
sub in2b
sta in2b
clra
sbc in2
sta in2
n2_done:
rts
neg_in1: ; in1 = -in1
ldhx in1
cphx #$8000
bne n1_doneg
ldhx #$7FFF
```

```
sthx in1
bra n1_done
n1_doneg:
clra
sub in1b
sta in1b
clra
sbc in1
sta in1
n1_done:
rts
siadd222: ; int siadd222(int, int): outA = in1 + in2
clra
brclr 7, in1, nn1
inca
nn1:
brclr 7, in2, nn2
inca
nn2:
psha
lda in1b
add in2b
sta outB
lda in1
adc in2
sta outA
pula
```

```
cmpa #1
beq sao_done ;(-) + (+)
cmpa #0             ;
beq nn3
brset 7, outA, sao_done ;(-) + (-)
ldhx #$8000
bra sao_st
nn3:
brclr 7, outA, sao_done ;(+) + (+)
ldhx #$7FFF
sao_st:
sthx outA
sao_done:
ldhx outA
rts
smul224: ; long int smul224(short int, short int): outA:outC = in1 * in2
clra
psha
brclr 7, in1, sm_nn1
inc 1,sp            ; one is negative
bsr neg_in1
sm_nn1:
brclr 7, in2, sm_nn2
inc 1,sp            ; two are negative?
bsr neg_in2
sm_nn2:
lda in1b
```

```
ldx in2b
mul                 ; B x D
sta outD
stx outC
lda in1             ; A x D
ldx in2b
mul
add outC
sta outC
txa
adc #0
sta outB
clra
adc #0
sta outA
lda in1b            ; B x C
ldx in2
mul
add outC
sta outC
txa
adc outB
sta outB
lda outA
adc #0
sta outA
lda in1             ; A x C
```

```
ldx in2
mul
add outB
sta outB
txa
adc outA
sta outA
pula            ; see if we have to negate the result
rora
bcc notneg
clra            ; 4-byte negate!
sub outD
sta outD
clra
sbc outC
sta outC
clra
sbc outB
sta outB
clra
sbc outA
sta outA
notneg:
ldhx outA
rts
;---------------------------------
start_AD1:
```

```
pshh
bset 7, portG
bclr 7, timer1.status ; acknowledge timer interrupt
ldhx filter_out
sthx timer1.ch0.H ; load PWM value from filter output
; set AD mux to position sensor input and start new conversion
mov #$46, AD.status       ;(4)is "AD ON", (6) is the input pin
bclr 7, portG
pulh
rti
;--------------------------------
start_AD2:
pshh
bset 6, portG
bclr 7, timer1.ch1.status; acknowledge interrupt
; set AD mux to strain guage input and start new AD conversion
mov #$47, AD.status
bclr 6, portG
brset 6, portA, notbump
ldhx cmd_speed
sthx set_speed
bra ad2done
notbump:
brset 5, portA, notL
mov #cmd_rewind, cmd_state
notL:
brset 7, portA, ad2done
```

```
        mov #cmd_run, cmd_state
        ad2done:
        pulh
        rti
        ;-----------------------------------------------
        AD_done:            ; nesting interrupt!
        pshh
        bset 5, portG
        ldhx AD.H ; acknowledge AD interrupt (done automatically when value is
read out)
        lda AD.status
        and #$1F ; mask-select AD channel selector
        cmpa #$06 ; which AD sample is it?
        bne stress_sample ; if it's position sample,
        position_sample:
        bset 7, portG
        lda AD.L ;AVERAGE POSITION SAMPLE
        add avg_pos+1
        sta avg_pos+1
        lda AD.H
        adc avg_pos
        sta avg_pos
        bcc chkP_done           ; saturate, just in case
        ldhx #$FFFF
        sthx avg_pos
        chkP_done:
        dbnz avg_cnt, ad_end ; if all samples are averaged, continue
```

```
ldhx avg_pos
sthx filter_in
bclr 7, portG
cli
jsr filter                ; CALL PID CONTROLLER!
sei
bra ad_end
stress_sample:
bset 6, portG
lda AD.L ;AVERAGE STRESS SAMPLE
add avg_force+1
sta avg_force+1
lda AD.H
adc avg_force
sta avg_force
bcc chkF_done      ; saturate, just in case
ldhx #$FFFF
sthx avg_force
chkF_done:
tst avg_cnt
bne ad_end
mov #AVG_CNT_MAX, avg_cnt
jsr in_buf ;queque in output FIFO
cphx #NULL
bne ad_add ; if FIFO overflows,
jsr out_buf ; throw out oldest sample
jsr in_buf
```

```
ad_add:                         ; else
mov avg_pos , x+ ; move average into transmit queque
mov avg_pos+1 , x+
mov avg_force , x+
mov avg_force+1, x+
clrh
clrx
sthx avg_pos            ; clear averages
sthx avg_force
ad_end:
bclr 5, portG
bclr 6, portG
bclr 7, portG
pulh
rti
;*****************************************************************
;* Main_Init - This is the point where code starts executing *
;* after a RESET. *
;*****************************************************************
App:
sei
ldhx RamEnd ; adjust stack pointer properly
;txs
jsr init_clk
jsr init_timer
jsr init_AD
jsr init_filter
```

```
        jsr init_buf
    jsr init_sci
    ;;
    ; first, confirm PortA as input with pull-ups to read switches
        mov #$00, portA.dir      ; write to DDRA
        mov      #$F0, portA.pullup
    ; second, set PortF 0-3 as output to drive LEDs
        mov      #$0F, portF.dir      ; write to DDRF
        mov #$FF, portF
    ;;
        mov #$02, portD.dir

        mov #%11111000, portG.dir
        cli             ; enable interrupts (required to debug with the monitor)
    main_loop:
    ;       lda portA
    ;       and #$0F
    ;       nsa
    ;       sta portF
        sta      SIMRS; write any value to SIMRS to reset the watchdog/COP
timer

        wait

    bra main_loop
    ;---------------------------------
    spurious: ; dummy interrupt handler
```

```
nop
rti
;*****************************************************************
;* Interrupt Vectors *
;*****************************************************************
ORG VectTable
    fdb     spurious        ; RTI
    fdb     spurious        ; IIC
    fdb     AD_done         ; ATD conversion
    fdb     spurious        ; Keyboard
    fdb     SCI_tx_int      ; SCI2 transmit
    fdb     SCI_rx_int      ; SCI2 receive
    fdb     SCI_err_int     ; SCI2 error
    fdb     spurious        ; SCI1 transmit
    fdb     spurious        ; SCI1 receive
    fdb     spurious        ; SCI1 error
    fdb     spurious        ; SPI
    fdb     spurious        ; TPM2 overflow
    fdb     spurious        ; TPM2 channel 4
    fdb     spurious        ; TPM2 channel 3
    fdb     spurious        ; TPM2 channel 2
    fdb     spurious        ; TPM2 channel 1
    fdb     spurious        ; TPM2 channel 0
    fdb     start_AD1       ; TPM1 overflow
    fdb     spurious        ; TPM1 channel 2
    fdb     start_AD2       ; TPM1 channel 1
    fdb     spurious        ; TPM1 channel 0
```

```
        fdb     spurious        ; ICG
        fdb     spurious        ; Low voltage detect
        fdb     spurious      ; IRQ
        fdb spurious          ; SWI
        fdb     main            ; reset
    END
```

**Register File**

```
    ; 68HC908GB60 Equates
    ;*** Memory Map and Interrupt Vectors
    ;*****************************************
    RomStart:           equ $182C
    VectTable           EQU $FFCC
    RomEnd:             equ $FFBF
    ZPRamStart:         equ $0080
    ZPRamEnd:           equ $00FF
    RamStart:           equ $0100
    RamEnd:             equ $107F
    Rom1Start:          equ $1080
    Rom1End:            equ $17FF
    ;*** Ports
    portA               EQU $0000       ; data register
    portA.pullup        EQU $0001       ; 25 kOhm pullup resistor enable
    portA.slewrate      EQU $0002 ; slewrate control enable
    portA.dir           EQU $0003       ; output enable
    portB               EQU $0004       ; data register
    portB.pullup        EQU $0005       ; 25 kOhm pullup resistor enable
    portB.slewrate      EQU $0006 ; slewrate control enable
```

```
portB.dir          EQU $0007      ; output enable
portC              EQU $0008      ; data register
portC.pullup       EQU $0009      ; 25 kOhm pullup resistor enable
portC.slewrate     EQU $000A ; slewrate control enable
portC.dir          EQU $000B      ; output enable
portD              EQU $000C      ; data register
portD.pullup       EQU $000D      ; 25 kOhm pullup resistor enable
portD.slewrate     EQU $000E ; slewrate control enable
portD.dir          EQU $000F      ; output enable
portE              EQU $0010      ; data register
portE.pullup       EQU $0011      ; 25 kOhm pullup resistor enable
portE.slewrate     EQU $0012 ; slewrate control enable
portE.dir          EQU $0013      ; output enable
portF              EQU $0040      ; data register
portF.pullup       EQU $0041      ; 25 kOhm pullup resistor enable
portF.slewrate     EQU $0042 ; slewrate control enable
portF.dir          EQU $0043      ; output enable
portG              EQU $0044      ; data register
portG.pullup       EQU $0045      ; 25 kOhm pullup resistor enable
portG.slewrate     EQU $0046 ; slewrate control enable
portG.dir          EQU $0047      ; output enable
RTI.status         EQU $1808 ; system RTI setup register
KBI.status         EQU $0016 ; keyboard interrupt setup register
KBI.pin_enable     EQU $0017 ; keyboard pin
portA.kbimode      EQU $0017 ;
AD.setup           EQU $0050 ; ATDPU | DJM | RES8 | SGN | PRS
AD.status          EQU $0051 ; CCF | ATDIE | ATDCO | ATDCH (5)
```

```
AD.H                EQU $0052
AD.L                EQU $0053
AD.pin_enable       EQU $0054
portB.ADmode        EQU $0054
sci.ch0.status.1    EQU $001C
sci.ch0.status.2    EQU $001D
sci.ch0.control.1   EQU $001A
sci.ch0.control.2   EQU $001B
sci.ch0.control.3   EQU $001E
sci.ch0.data        EQU $001F
sci.ch0.modulo.H    EQU $0018
sci.ch0.modulo.L    EQU $0019
sci.ch1.status.1    EQU $0024
sci.ch1.status.2    EQU $0025
sci.ch1.control.1   EQU $0022
sci.ch1.control.2   EQU $0023
sci.ch1.control.3   EQU $0026
sci.ch1.data        EQU $0027
sci.ch1.modulo.H    EQU $0020
sci.ch1.modulo.L    EQU $0021
timer1.status       EQU $0030
timer1.H            EQU $0031
timer1.L            EQU $0032
timer1.modulo.H     EQU $0033
timer1.modulo.L     EQU $0034
timer1.ch0.status   EQU $0035
timer1.ch0.H        EQU $0036
```

```
timer1.ch0.L            EQU $0037
timer1.ch1.status       EQU $0038
timer1.ch1.H            EQU $0039
timer1.ch1.L            EQU $003A
timer1.ch2.status       EQU $003B
timer1.ch2.H            EQU $003C
timer1.ch2.L            EQU $003D
clk.setup.1             EQU $0048
clk.setup.2             EQU $0049
clk.status.1            EQU $004A
clk.status.2            EQU $004B
clk.filter.L            EQU $004D
clk.filter.H            EQU $004C
clk.trim                EQU $004E
SIMRS                   EQU $1800       ; System Integration Module Reset Status
register
```