Automatic Tensile Tester 4000

Fall 2010

Group 6

Alan Beauchamp

Devon Jackson

Justin Ewing

Introduction

1. Executive Summary…………………………………………………………………...1
2. Design Summary………………………………………………………………………2
   1. Hardware Design………………………...……………………………………......2
   2. Software Design….……………………………………………………………......3
3. Requirements………………………………………………………………………….4
4. Specification…………………………………………………………………………..5
5. Bill of Materials……………………………………………………………………….6
6. Project Milestones… ………………………………………………………………….7
7. List of Responsibilities….…………………………………………………………….8
8. MTS LandMark System……………………………………………………………….9
9. Reasearch…………………………………………………………………………….10
   1. Load Cell…………………………………………………………………………10
      1. Load Cell Parameters………..………………………………………………10
      2. Types……………………….……………………………………………...…11
      3. Load Cell Components…….………………………………………………...13
10. Strain Gauge…………………………………………………………………………14
    1. Strain Gauge Parameters…………………………………………………………15
       1. Gauge Length……………………………………………………...…………15
       2. Gauge Pattern……………………………………………………...…………16
       3. Gauge Resistance. ………………………………………………...…………17
       4. Gauge Wires. …………………………………………….………..…………18
    2. Strain Gauge Components………………………………………..….…………..18
11. Thermocouple……………………………………………………………..…………21
12. Resistance Temperature Detectors (RTD)…………………………..………………..24
13. Thermistors…………………………………………………………………………..25
14. Surface Temperature Sensors………………………………………………………...26
15. ENVIROSEAL™ HD Sensor…………………………………………………..……27
16. Resistance Temperature Sensors…………………………………………….…….…28
17. Amplifiers……………………………………………………………………..……..29
    1. INA 118………………………………………………………………………..…29
    2. LF 351 (non-inverting amplifier)………………………………………………...31
    3. AD 595………………………………………………………………………...…32
    4. Motor…………………………………………………………………………..…32
       1. AC……………………………………………………………………………32
       2. DC……………………………………………………………………………32
       3. Gears…………………………………………………………………………33
       4. Steppers………………………………………………………………….…...34
       5. Actuators………………………………………………….………………….36
18. Processor Type Comparison …………………………………………………………37
    1. Microcontroller ……………………………………………………………….…37
       1. Development boards and Chips analysis ……………………………………39
       2. Inter-Integrated Circuits……………………………………………………...39
       3. Arduino Series ………………………………………………………………39
       4. Homebrews Development Board…………………………………………….40
    2. Data logging & External Memory……………………………………………….40
       1. Data serialization…………………………………………………………….40
       2. USB Storage………………………………………………………………….42
       3. SD / MMC Card……………………………………………………………...42
       4. Logomatic v2 Serial SD Datlogger…………………………………………..43
    3. Data logging Prototype and development…………………………………..……43
    4. User & Communication Interface……………………………………..…………44
       1. Current GUI: TestWorks…………………………………………..…………44
       2. Prospective GUI……………………………………...………………………45
          1. Development Environments…………………………………...…………45
          2. Layout……………………………………………………………………46
          3. Web Interface………………………………………………….…………46
       3. Physical LCD interface with Push Buttons………………..…………………47
       4. USB to Serial…………………………………...……………………………49
       5. Proportional–Integral–Derivative Controller…………...……………………50
          1. Motor PID Control…………………………………………………….…50
          2. Hardware \w PID………………………………………………...………52
          3. Software PID…………………………………………………..…………53
       6. System Flow…………………………………………………………………54
          1. Applying a Linear Pressure………………………………………………55
          2. Constant Load……………………………………………………………56
          3. Wave………………………………………..……………………………57
    5. Power Sources……………………………………………………………………58
       1. Voltage Regulators………………………………………………...…………58
          1. 78XX Regulators…………………………………………….…………..59
          2. Tri-Mode Regulators…………………………………….……………….60
19. Design………………………………………………………..………………………63
    * 1. Op amp and Load Cell………………………….……………………………63
      2. Strain Gauge…………………………………………………………………66
      3. Thermocouple……………………………………………………………..…67
    1. Motor (Software)…………………………………………………...……………69
       1. Motor-Atemga328P Schematic………………………………………………70
       2. Analog Signals (software) …………………………………..………………71
       3. Analog devices to Atemga328P ………………………………………..……73
       4. Data logging class …………………………………………...………………73
       5. Micro SD Card - Microcontroller schematic……………………….. ………74
       6. User Interfaces ………………………………………………………………75
          1. Physical Interface …………………………………………..……………75
             1. Physical Interface class ………………..…………………….………77
          2. Web Interface…………………………………………………….………78
          3. USB to PC interface…………………………………………………...…78
       7. Files Classes and Data Structure ……………...………………..……………79
       8. Experiment Flow Algorithms………… …………………………………80
20. Prototyping and build instructions…………………………………………...………81
    1. PCB …………………………………………………………………………..….82
    2. Software …………………………………………………………………………83
       1. GUI …………………………………………………………….……………83
       2. Serial Communication ………………………………………………………83
       3. Atmega328p programming ………………….………………………………83
       4. Micro SD …………………………………………………….………………83
       5. Input and Output Signals Software………..…………………………………84
21. Testing……………………………………..…………………………………………84
    1. GUI Event Table…………………………………………………………………84
    2. Test Environment ……………………………………………………………..…85
    3. Stopping Criteria…………………………………………………………………86
    4. Individual Test Cases………………….…………………………………………86
22. Appendix

**Executive Summary**

We are designing and building the Automatic Tensile Tester 4000. The main scope of this project is to make this machine a small scale test device for creep, creep-fatigue, and creep crack growth testing. Creep is a deformation that occurs below the yield strength of a material at temperatures above 40% of the melting temperature.  Fatigue is the non-recoverable loss in strength that occurs when a material is subject to cyclic loading. When a material is subject to creep conditions and cyclic loading, creep-fatigue occurs, where both creep deformation and fatigue loss in strength reduce the life of a material. Often, the effects of creep and fatigue are idealized as damage. This device will be used to study those effects, because creep damage is the primary design consideration for steam and gas turbine OEM’s.

We will be working along with a University of Central Florida mechanical engineering department in order to complete this device. The mechanical engineering department will design and fabricate a small scale test device that can confer a force or displacement history to a sample-sized specimen. We as the electrical engineering group also have some major parts in the completion of this device. The main objective for us is to develop a plug and play system to transmit all the data from the environment using analog sensors which consist of the strain gauge, thermocouple, and load cell. A secondary objective is to control the force/displacement and to measure temperature on the specimen. Also the device will transmit data to a written text file so that the user is able to observe the specific force/displacement and temperature history. This written file will be stored on a micro SD card that will implement a fat16 file system to allow the data to be read on both the device and pc without conversion. Finally the user software interface must be suitable for both Windows and Mac.

The motivation for this project is education, because this machine can be used by students in the lab. Our sponsor has a bigger scale machine which cost about $20,000 a machine. The disadvantage of this machine is it takes up a whole computer while performing just one test. The device we are building will be able to be able to do one test but not take up a computer due to the fact it will be saving the data onto the micro SD card. This can be taken out anytime by first putting the system on pause and then plugged into any computer to observe the results of the testing. Also, the software that comes with the machine is not free or open-sourced but the software we are writing can be open-sourced.  The main advantages of are device is the low cost to build, which in turn means more machines for labs and students.

This device will have specifications and requirements which have been set by our sponsor. The device is the first generation of its kind. The technical approach to this project is as follows. We will be using a microcontroller which is the brains of the system. It will be programmed to read in and analyze all inputs and control the motor using a linear control system. The H-bridge will have a part in controlling the motor. We will be using strain gauges, and load cells to measure and control the force applied on the specimen. Thermocouples will also be used to measure the temperature on the specimen. Our sponsor in the future wants to expand the expectations of this device. For example, being able to run multiple tests, using one computer to control all the devices, or even more advanced options.

**Design Summary**

**Hardware Design**

Our project will be powered by output, US standard of 120 volts AC, from a wall unit whose output is run through an AC to DC converter. For the converter, we plan to use a power cord whose output is approximately 24 volts and 500 millivolts. From there, the individual components will be powered from the outputs of a 7805 regulator and a 7812 regulator which will be conditioned with two capacitors apiece. These systems will power the sensors, amplifiers and the microcontroller.

The load cell will receive its excitation signal from the 7812 regulator. This will make the total output voltage of the of the column load cell 24 millivolts. This output will go to an amplifier to improve the strength of the output to a maximum of no greater than 5 volts. This output will be analyzed by the microcontroller.

The strain gauge circuit will receive direct input from the 7812 regulator. The strain gauge is inserted into one side of a Wheatstone bridge formation. This quarter-bridge circuit will be used to measure the electrical output. The resistor value that is in series with the strain gauge will be of the same value as the initial strain gauge resistance. There will be two leads from the output of the circuit going to the microcontroller who will have the necessary.

We will be designing the motor portion of the project. Our group has decided to use an actuator which gives out 4000 lb of thrust to drive our machine. The motor is able to brake, go forward, and reverse. The motor will drive the thrust down onto the specimen, which is required for the experiment. We will be using an H Bridge to control the motor. The H Bridge will then connect to the microcontroller. The motor will then ultimately be able to be controlled thru our project box. That is one of the goals for the project. The motor will have to be calibrated and have a PID on it so that we can have the readings and make sure it is doing the job accurately.

We also have the task of installing a thermocouple. We will be using a Type K thermocouple. The thermocouple will need to be amplified so it has to be connected to an amplifier for it to work properly. The thermocouple will be placed on the machine and from there it will run into the amplifier, which supplies the voltage. The amplifier will then run from there to the microcontroller. The microcontroller will be programmed by us to run many parts of this project.

**Software Design Summary**

The software for the Automatic Tensile Tester 4000 is user friendly, reliable and effective. It is also open source, which will allow the user to edit and add to it as needed. For ease of portability the GUI will be created using Java in the backend and Swing of visual representation. The device will connect to a PC using a USB 2.0 A to B connection using java’s RxTx library to control hardware side of communication. For data logging the A.T.T 4000 will use a micro SD card equipped with the ability to be accessed using a fat16/32 file system.

An Atmel atemga328p microcontroller will be the brains of our system allow use to control the actuator movement. Read and analyze all analog and digital signals including but not limited to temperature, load, and strain. All code for the microcontroller will also be open-source allowing any developer to improve the system as they see fit.

The physical interface will also consist of a LCD 2 x 16 displays, which will display current status and state of the system. Screw terminal connection for easy interchanging and plug and play of components such as the actuators and thermocouple. The system is also equipped with multi buttons that will allow the user to pause, resume, and stop test.

Finally the system will consist of a read only web interface, which will allow a user to access information such as current load, temperature and other statuses of the current running experiment. This interface is read only for security reasons, protecting ones experiment from any malicious intent.

**Requirements**

* Low cost
* Accurate
* Portable
* Work without a computer connection
* Must be able to pause/re-edit/resume
* Must be able to data dump to or read data from USB
* Must be able to manually control entire experiment from box
* Must be able to program experiment from computer connection
* Control two material testers simultaneously
* Interfacing accessible on web (Linux, Windows)
* LCD must display status and cycle
* Implement fat16 file system for data logging
* Status light for experiment
* Wall powered
* Plug and play components
* Software must be user friendly and open source ‘

**Specifications**

* PCB - (6 sq. in. at base)
* Automatic Command/data retrieval for 1 hour
* Thermocouple- measure temperature accurate to 5o
* Load Cell- measure force accurate to ± 10%
* Be able to hold and test loads up to 4000lb
* Be able to run 3 tests (specified) for 15 minutes each
* Control box 8”x 6” and weight no more than 5 pounds
* Rate temperatures from 32oF to 212oF
* Accurately measure strain gauge factor ± 10%

**Bill of Materials**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **QTY** | **Cost** | **Total** | **Status** |
| Actuator Motor | 1 | $250.00 | $250.00 | Not Ordered |
| Atmega 328P | 2 | $4.30 | $8.60 | Not Ordered |
| 10k Resistor | 1 | $0.25 | $0.25 | Not Ordered |
| H-Bridge | 1 | $2.35 | $2.35 | Not Ordered |
| 3.3 K Resistor | 3 | $0.25 | $0.75 | Not Ordered |
| 1.8 K Resistor | 3 | $0.25 | $0.75 | Not Ordered |
| 5 Volt Regulator | 1 | $1.59 | $1.59 | Not Ordered |
| 3 Volt Regulator | 1 | $1.95 | $1.95 | Not Ordered |
| Sparkfun Serial LCD | 1 | $24.95 | $24.95 | Not Ordered |
| Ethernet Shield | 1 | $45.95 | $45.95 | Not Ordered |
| Micro SD Card | 1 | $9.95 | $9.95 | Not Ordered |
| SD Breakout Board | 1 | $9.95 | $9.95 | Not Ordered |
| Wall Wart 12 Volt | 1 | $5.95 | $5.95 | Not Ordered |
| Load Cell | 2 | $475.00 | $950.00 | Not Ordered |
| Strain Gauge | 2 | $330.00 | $660.00 | Not Ordered |
| Thermocouple | 2 | $5.00 | $10.00 | Not Ordered |
| PCB | 1 | $121.00 | $121.00 | Not Ordered |
| 10 nf Capacitors | 2 | $0.45 | $0.90 | Not Ordered |
| 100 nf Capacitors | 2 | $0.35 | $0.70 | Not Ordered |
| FT245RL Breakout Board | 1 | $14.95 | $14.95 | Not Ordered |
| USB Cable | 1 |  |  | Not Ordered |
| 8 X 6 Project Box | 1 | $6.99 | $6.99 | Not Ordered |
| Buttons | 6 | $0.50 | $3.00 | Not Ordered |
| LED | 10 | $0.35 | $3.50 | Not Ordered |
| Omp Amps | 3 |  |  | Not Ordered |
| Screw Terminals- 2pin | 4 | $0.95 | $3.80 | Not Ordered |
| Screw Terminals- 3pin | 1 | $0.95 | $0.95 | Not Ordered |
| Arduino | 1 | $29.95 | $29.95 | Not Ordered |
| Total |  |  | $2,168.73 |  |

**Project Milestones**

It is very important and critical to the project that we set these project milestones. We must also make sure they are clear and realistic. These milestones will keep our group focused, because it’s a timeline. Time must planned and used wisely for this project to complete on time. It will keep us from spending too much time on parts in the project. Our group does not want to be at the last minute trying to complete the project. If we plan ahead then we will have success. Also, getting started early allows time to fix any errors or mistakes occurring.

|  |  |  |
| --- | --- | --- |
| **Task Name** | **Start** | **Finish** |
| Research | 8/30 | 10/15 |
| Design | 8/30 | 12/4 |
| First Report | TBA |  |
| Design Review | TBA |  |
| Parts Acquisition | 2/15 |  |
| Programming Code | 1/1 | 3/21 |
| Build Interface | 1/1 | 4/3 |
| Install Motor | 2/4 | 4/3 |
| Install Sensors | 2/4 | 4/3 |
| Testing | 2/4 | 4/3 |
| Install Fix | 4/3 | 4/3 |
| Troubleshooting | 4/3 | 5/3 |
| Final Review | TBA |  |
| Demonstration | TBA |  |
| Final Presentation | TBA |  |
| Final Report | TBA |  |

**List of Responsibilities**

Our project requires a lot of responsibilities, but we have tried to divide them evenly among the team members. The tasks assigned to each team member just means they are mostly in charge of getting them done. But the other team members will also assist in helping to getting the task done. That way a team member doesn’t have to be all by themselves. This will also cause team members to learn every aspect of the project even though they’re not in full control of that task. Their might also be other task added to the responsibilities list later once the project design gets rolling.

**MTS Landmarktm Servo hydraulic Test Systems**

Figure 1 displays the current system being used in the labs. It is called the MTSLandmarktm Servo hydraulic Test System. It presents a wide selection of actions to execute a whole range of static and dynamic testing. Some of the options are durability, fatigue crack growth, high cycle fatigue, low cycle fatigue, fracture toughness, tension, compression, and more. Testworks is the software used to drive the machine. These machines come in different models such a floor standing and table top model. This machine is ideal for testing many different components. Load cells and strain gauges also make up this machine, which are used to measure the load applied to a specimen. Thermocouples are also utilized to measure the temperature. The usually cost for one of these machines is about $20,000. The only disadvantage is the machine needs a computer to run on, so that’s about a $2000 setback.



**Figure 1 -** MTS Servo Hydraulic Test Systems (waiting for permission from MTS)

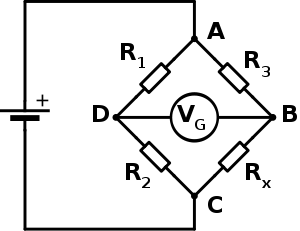
**Research**

**Sensors**

**Load Cell**

The load cell (force transducers) is the primary function of the MTS machine. It measures the force on a given material and converts it to an electrical output based on the electrical input on the machine. The load cell is just the measurement portion of that machine. This component measures the outside force on the material, while the strain gauge measures the physical reaction to the induced force.

The load cell uses variable resistors in a Wheatstone bridge set-up (figure 2) to produce the output voltages based on the size of the load. When the load is applied to the load cell the internal resistance changes and the output voltage changes in accordance to the input voltage and the new resistance of the load cell.



**Figure 2 –** Wheatstone bridge that controls output of load cell

Different grades of load cells in the same company general use a similar grade for their cells (i.e. 2 mV/V). Also, the different classes of load cells have different pressure resistances. This way, that difference tolerances produce

**Load Cell Parameters**

When looking for the right load cell for our application, we first researched what to look for. We found that one should select a load cell based on: the type and mode of operation, the number of load cells, the capacity the performance or accuracy level, the method of mounting, sealing level and material of construction and the cost.

The mode of operation is either in compression mode or tension mode. When in compression mode, the load is placed on top of the load cell and the load is measured. In tension mode, the load hangs freely from the load cell and the load is measured.

The performance of a load cell can be split into 3 main groups of parameters: time dependent, environment dependent, and mechanically dependent. Performance can be affected by any one of these things so it is important to take into account what kind and where the tests are being conducted because all of these conditions can significantly affect the accuracy of the load cell. When choosing a load cell it important to take all of these parameters into consideration. For our experiment, we will not be undergoing a wide range of temperatures because this will primarily be in a lab environment, so the temperature will stay in a relatively stable range. Therefore, only the other two parameters will be taken into account for the selection of the load cell.

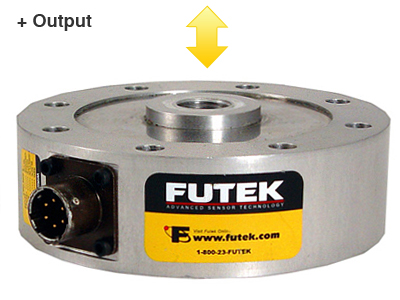
When taking into account the load cell capacity, the weight of the weighing structure, the maximum load and the addition overload from external factors.

Overload is when there is more force pressing against the load cell than can the rated capacity. Most load cells can safely withstand 50% more than their rated capacity but the overload is not made for sustaining this input for periods of time, nor is it made for shock loading. Extreme loading can damage parts of the load cell making it insufficient for use in a test environment.

**Types**

*Pancake*

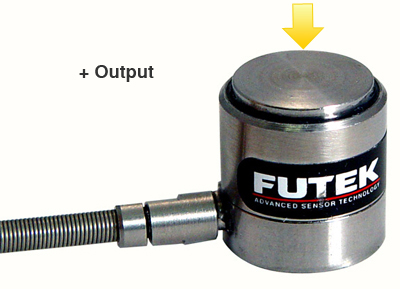
The pancake load cell (figure 3) is a force-centered load where the pressure is applied to the top of the load cell. This load cell seems designed for blunt constant force applied to a flat surface. Sustained pressure on a flat surface makes this effective for estimating the total force on the supports for a scaled bridge. Because it offers a flat surface for the measurement of force, it is an ideal load cell for the project.



**Figure 3-** pancake type load cell (waiting for permission from Futek)

*Column*

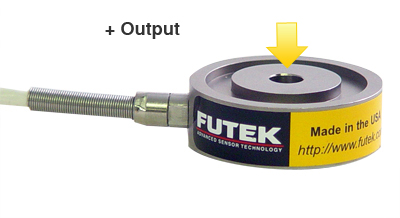
The column load cell (figure 4) is a shaft with a thick loading center made for broad and widely distributed weights. Unlike many of the center and point force load cells, this load cell can handle a plate, rod, and many other types of specimens that want need to be tested.



**Figure 4 –** column type load cell (waiting for permission from Futek)

*Donut Hole*

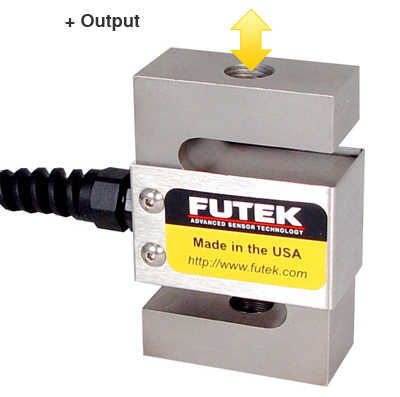
The donut hole load cell (figure 5) is another force centered load cell where, instead of being placed on top of the load cell, the material is run through the load cell and sealed with a screw or another fastening tool. This load cell is perfect for measuring the force applied to a fastener for a pipe and by large wall screws. Because our material tester is measuring the force applied to a blunt object, this load cell would not be ideal for our project.



**Figure 5 –** donut hole type load cell (waiting for permission from Futek)

*S-Beam*

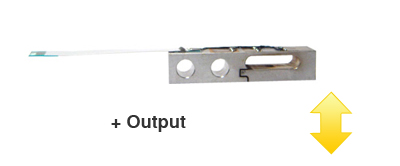
Known for its “S” shape design, this load cell (figure 6) is commonly used for heavy loads such as tank level, hoppers and truck scales. These applications are for very large objects, but they also make these cells in sizes available for our specific applications for the material tester. Our tests will not assume more than 1000 lbs of force on the load cell. Also, this particular load cell is very large and may not fit on the mechanical rig. Therefore, for our purposes, this specific piece will not be considers for our final load cell.



**Figure 6 –** s-beam load cell (waiting for permission from Futek)

*Shear Beam*

The shear beam (figure 7) is measured at the middle of the beam on the top-side and under-side where the strain is measured to figure the total force on the beam. This cannot function in the space that we are given, therefore this piece will not be considered.



**Figure 7 –** shear beam load cell(waiting for permission from Futek)

**Load Cells Components**

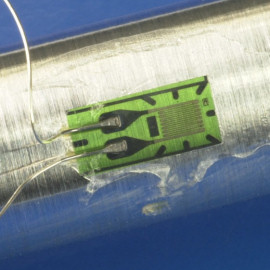
Transducer Techniques offers their MLC Series Load Cell, a column load cell that has four leads, two excitation and two signals, which relay the electrical signals to and from the load cell. The output of the cells is 2 mV/V and it has an excitation voltage of 10 volts DC. The terminal resistance is 350 Ω meaning that it will not draw as large of a current from the power source. It can handle temperatures from (-65 to 200F). Its radius is 2 inches meaning it can handle sizeable loads. Our desired rating is 5,000 lbs. and it is available for purchase. There is a 150% overload protection in cause of excess force.

The developer Interface Force offers a small Diameter High Capacity Load button. This is a high accuracy load cell that can function in temperatures from -40o C to 80o C. the rated output is 2 mV/V (+40%/-20%). The diameter of the push button where the load is applied is 1.05 inches. This makes it relatively small for something capable of handling forces up to 30,000 pounds of force. There is a 150% overload protection in cause of excess force.

Futek offers their Miniature Column Load Cell for loads of medium to high capacity. These cells can handle loads from 7,500 to 10,000 pounds of force directly pressed on the top of the cell with a 150% for the rating of the safe overload before failure. The maximum excitation voltage this load cell can receive is 18 volts in AC or DC and an internal bridge resistance of 350 Ω. The Futek brand cells have an operating temperature of -50o C to 93o C.

**Strain** **Gauge**

A strain gauge (figure 8) is a device that measures the displacement and deformation of a material as it undergoes changes in temperature and stress deviation. This is often used in conjunction with a load cell to monitor how the material deforms under different stresses. Stress is measured in change in inches per inch. The output of the system is an electrical signal that must be input into an equation to measure the action strain on the specimen.



**Figure 8 –** how strain gauge is applied to specimen (waiting for permission from doitpoms)

There is a variable resistor in the strain gauge that adjusts as the material changes the shape of the strain gauge. When the gauge expands, the resistor in the gauge also expands and the result of the expansion is an increased resistance due to the resistance formula R=ρ(L/A). This change in change in resistance leads to a change in output voltage which is what is read in the analysis. The output voltage is measured by the same standard as a load cell. The specifications with each part produce a chart that assists in determining the strain in change in inches per inch based on the output millivolts per volt induced. The reason the output is so small compared to the input is because of the following equation,, where Rg is represented by the present output of the strain gauge. The equations that determine how the resistance changes in reference to the change in length is called the gauge factor and it is defined by the equation, , where R is the initial resistance, L is the initial length, CR is the present change in resistance and CL is the present change in length. The combination of these two equations will tell us the change in length of the strain gauge. The gauge factor is generally a given for the strain gauges and is often times found to be a factor of 2.

**Strain Gauge Parameters**

Strain gauges offer many different options in their selection that affect their use, application, cost and size. These make each strain gauge suitable for different purposes and areas. This can affect the design in using one gauge that can handle all the different tests, or if you need multiple gauges to get all the necessary measurements that you need.

**Gauge Length**

The gauge length is better referred to as the strain sensitive length. The strain sensitive length is the size of the variable resistor that drives the main functions of the strain gauge. They can range from 0.2 mm to 100 mm. A length of 3 mm to 6 mm is used in most applications.

The shorter gauge lengths (less than 3 mm) are used when there is not enough space to mount a longer gauge or when you need to measure a specific area when there is a high strain gradient. Longer gauge lengths (greater than 6 mm) are used when high accuracy is needed. They are also easier to mount and install. Since they are larger, they are also less susceptible to the effects of heat dissipation. Longer gauges are also less expensive than the more extreme lengths of the gauge spectrum.

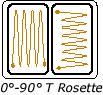
**Gauge Pattern**

The uniaxial strain gauge (figure 9) is used primarily used when there is only one direction in need to be tested and investigated. These are some of the least expensive strain gauges available and the most widely used. Multiple uniaxial gauges can do the job of a single more expensive combination gauge. These are also often less expensive than using a combination gauge as they are usually the least expensive gauges available.

http://www.efunda.com/designstandards/sensors/strain_gages/images/StrainGage_Uniaxial.gif

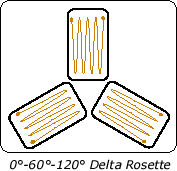
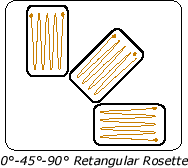
**Figure 9 –** formation of uniaxial strain gauge (waiting for permission from efunda)

The biaxial strain gauge (figure 10) is a combination strain gauge with one strain gauge oriented 90o degrees from the primary gauge. This gauge jointly measures strain in two different directions. It is said that this is desired when the principal stresses need to be investigated and the principal axes are known.



**Figure 10 –** formation of a biaxial strain gauge (waiting for permission from efunda)

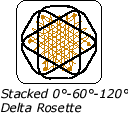
One selects a tri-strain (figure 11) gauge if you need to measure the principal stresses need to be investigated and the principal axes are unknown. These primarily come in two types, a 0o-45o-90o rectangular orientation and a 0o-60o-120o delta-triangular orientation.



**Figure 11 –** two formations of tri-strain gauges(waiting for permission from efunda)

There are two separate orientations for the combination gauges. The options are having a planar layout and a stacked layout. The planar layout has lower heat dissipation than the stacked layout. The planar layout is also more accurate than the stacked layout because each strain gauge is actually in contact with the surface.

The stacked layout is better when there is a high strain gradient and you want to record the strain in one area. The stack layout is also better when space is limited.



**Figure 12 –** stacked tri-strain gauge in delta pattern (waiting on permission from efunda)

**Gauge Resistance**

The resistance of the strain gauge is important when designing the Wheatstone bridge. The resistor in series has to be of the same resistance is the same as the initial resistance of the strain gauge. This also affects the sensitivity of the strain gauge. Even amongst strain gauges with the same gauge factor, the fluctuation in the gauges would be larger and there would be a wider range of resistances.

When you have a lower resistance (around 120 Ω) there is less of an issue with fatigue loading. Thinner wires are more susceptible to fatigue, so this can be negated with wires with a larger diameter. Also, gauge with a lower resistance usually cost less than higher resistance gauges.

The higher resistance gauges (common values are 350 Ω or 1000 Ω) generally offer a higher sensitivity than their lower resistance brethren. The problem is that these are generally better used with established systems for measurement such as a Wheatstone bridge substitution box.

**Gauge Wires**

Constantan alloy wire is the most inexpensive wire available for your gauge. It has slow response time to changes in strain, so it is better used for static strains or strains that do not vary much in short amounts of time. They aren’t good for extreme conditions and are better used in controlled environments like a laboratory. These are often the default choice for most gauges.

Isoelastic alloy wires carry a very high quality signal as they have a high signal-to-noise ratio. These are also good for measuring dynamic strains and quick cyclic loads. These are better for when the experiment is in an air-controlled laboratory.

Karma alloy is best when there is a very low temperature environment, often temperatures around absolute zero. These wires are also capable of self-temperature compensation, meaning that they are not very susceptible to change their resistance when there are changes in temperature.

Platinum wires are only used when there are measurements made at above temperatures of 230o or more.

**Strain Gauge Components**

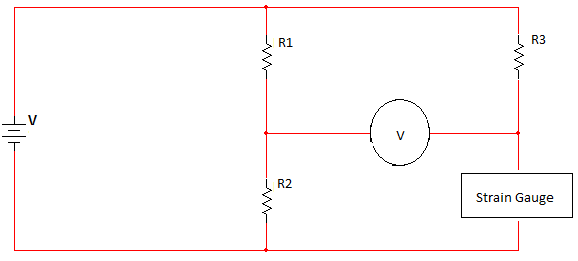
Texas instruments offer a general purpose gauge for general use, its F series. The foils are 0.003 mm to 0.007 mm thick. The adhesive on the back of the strain gauge is a thin coat of epoxy. The general operating temperature is from -20o C to 80o C. It uses a copper and nickel alloy foil. Its strain limit is 3% of the initial rating.

BCM offers three different types of gauges. They all are used to probe weak signals and to compensate for a significant non-linearity encountered in the manufacturing of transducers. The Class-A gauge is for high accuracy transducers. Class-B is for temperature compensated gauges that still have good accuracy. Class-C is for stress analysis in critical measurement conditions.

Omega offers pre-wired strain gauges in a 2- and 3-wire format with insulated leads with its KFG series. The gauge wires are made of Constantan foil. It also has gauges in sizes from 0.2 mm to 30 mm so it has enough choices for any application should we need them.

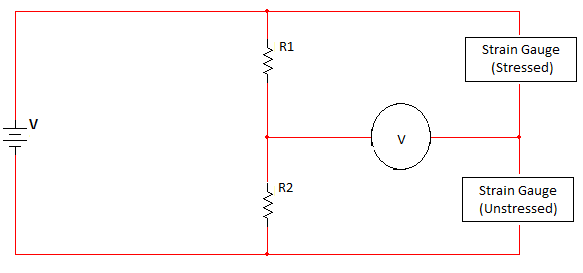
A strain gauge can be assimilated into the system in a variety of ways. The strain gauge should be integrated into a Wheatstone bridge format (figure2). The three formats are quarter-bridge (figure 13), half-bridge (figure 14), and full-bridge (figure 15) circuits. The half-bridge and full-bridge circuits are both more accurate and more sensitive than the quarter-bridge circuits. In a tubular specimen, it would be hard to find complimentary sections to where one strain gauge would be stressed and one strain gauge would be unstressed. Also, when there are more strain gauges there would be an increase in the cost of the project.

The quarter-bridge format is the lowest cost because you only need a single strain gauge. It is preferred in applications that do not have two areas that are opposing in strain. It only measures the tension and compression on one side. Since most gauges deal only with units of microstrain (ε = 10-3), the system is vulnerable to the effects of resistance increase due to heat.



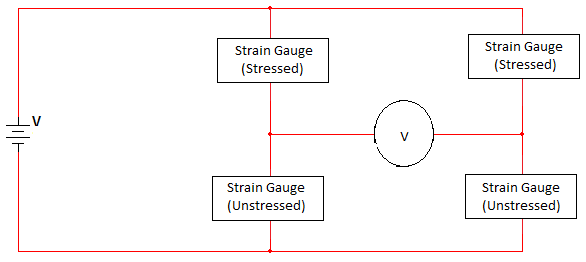
**Figure 13 –** strain gauge in a quarter-bridge

In many applications, the half-bridge is the preferred strain gauge of choice. For a test specimen where there both the stress and strain can be measured at concurrent points, this proves to be an ideal. Since there are two strain gauges on the same system, and they both undergo the effects of resistance increase due to heat, the reading is more accurate because the effects of heat are negated.



**Figure 14 –** two strain gauges in half-bridge

The full-bridge circuit can be used to measure the strain along two axes. They can be configured in many ways to find many different stresses and strains including shear, bending, and torsional. They combine the strain gauge pairs in different formations to measure different parameters.



**Figure 15 –** four strain gauges in a full-bridge

After researching the different formations and combinations of the strain gauge and Wheatstone bridge, we brought the ideas with their pros and cons to the sponsor of the project. He asked that we only use a single strain gauge, so we will be using a strain gauge in the quarter-bridge fashion. Also this simplifies the equations necessary to find the amount of strain associated with the load cell.

**Thermocouple**

A thermocouple is a type of industrial thermometer that is used to measure extreme temperatures it does this using a junction between two different metals that produce a voltage that is related to temperature difference. It was discovered by an Estonian physician named Thomas Seebeck discovered accidentally that the junction between two metals generates a voltage which is a function of temperature. The important effect named by the person who discovered it Seebeck effect is when energetic electrons at the hot end diffuse toward the cold end, pushing less energetic electrons along with them, resulting in a higher static potential difference. Figure 16 below displays a thermocouple.



**Figure 16 –** Thermocouple (waiting on permission from Wikipedia)

There are different configurations used in thermocouples. One is the classical thermocouple loop configuration where two different metal conductors are used. They are chemically electrically and physically compatible. They produce different potentials when they are given the same temperature. After the metals are connected then the loop is broken by picking a location where the temperature matches. Thermal gradients are unaffected by maintaining uniform temperatures where the leads connect. Also to keep stay thermocouples effects small the thermal gradients across the lead wires are avoided. Also to stop any remaining effects the leads are matched very well.

Another configuration is to omit the cold junction compensation, since in practice the configuration is accomplished by maintaining ice water slurry and actual cold junction which is rarely feasible. This configuration directly measures the potential across the two terminals. To complete the temperature measurements the final temperature has to be determined. There are two ways electronic cold junction compensation and independent cold junction measurement.

One advantage is that they are inexpensive and interchangeable and as mentioned above they can measure a wide range of temperature usually from -200 to 2600 degrees Celsius. Another advantage is that they can be brought into contact with what is being measured the only requirement is that they are correctly and safely grounded.

Even though they are very advantages they do have some problems creating disadvantages, one being that they are prone to corrode since they are made from two dissimilar metals. Any type of corrosion no matter how small will cause the results to be off. Also corrosion specially if is slight is hard to detect and hard to detect if the reading is off. Also depending on what type of thermocouple it is there are marginal errors. Therefore crating a wide range if accuracy especially at high temperatures and they are hard to recalibrate depending on the environment. Also because depending on what is being used for long wires might be requires making them expensive is they are used.

There are different types of thermocouples which can be chosen according to cost, availability, convenience, melting point, chemical properties, stability, output, and of course what type is best depending on what is being used for, meaning the range that is needed each having some advantages and disadvantages. They are listed below: Type K, Type E, Type J, Type N, Type B, Type R, Type S, Type T, Type C, Type M, and Chromel-golden/iron.

Type K is the most common general purpose type of thermocouple; it is inexpensive and has a range -200 to 1350 degrees Celsius. The metals used are nickel-chromium and nickel aluminum.

Type E which has a high output which makes it good for the study of production of very low temperatures usually considered being below -150 degrees Celsius of materials (cryogenics). They are also good for pharmaceutical and chemical applications that have very small amounts of ingredients and they critically have to be measured. Cryogenics usually use the scale Kelvin or the Rankine scale. Type E thermocouples are made from a combination of Chromel and Constantan with Chromel as the positive side and Constantan as the negative.

Type J has a more restricted range than Type K. They are made from a positive iron wire and a negative Constantan wire. They are susceptible to oxidation and rusting in sub-zero temperatures. They are also susceptible to temperatures above 760 Celsius which causes a magnetic transformation and causes permanent damage to the calibration. This thermocouple is usually used for old equipment that cannot take the newer thermocouples.

Type N is made from Nicrosil and Nisil. This thermocouple is made for high temperatures that go above 1200 Celsius they resist high temperature oxidation. The N type is suppose to be an improve K type. This is supposed to be cost effective compared to the B, R and S mentioned below.

The Following types are type B, R, and S. These following types of thermocouples are the “noble” metal thermocouples. They are most stable of all the thermocouples and they are usually only used for high temperatures measuring more than 300 Celsius.

Type B thermocouples are made from Platinum and Rhodium and as mentioned before are used for high temperature up to 1800 Celsius and are usually used above 600 Celsius. These thermocouples are pretty much useless below 50 Celsius because they produce the same output from 0 to 42 Celsius.

Type R are made from Platinum and Rhodium and are also for high temperatures

and measure up to 1600 Celsius. They have low sensitivity and high cost so they are not suited for over all purpose. Type R has similar uses than Type S since they offer an improved range of stability.

Type S are made from Platinum and Rhodium and are also suited for high temperatures up to 1600 Celsius, but they can usually with stand short time periods of 150 Celsius and continuous of 1450 Celsius. They need to be protected from high temperatures atmospheres to prevent metallic vapors reaching going in the tip which results in a degrees of the EMF generated. This thermocouples is usually use for calibration for the melting point of gold.

Types T are made from Copper and Constantan and are used to measure from -200 to 350 Celsius and are usually used in laboratories rather than industrial applications. They have no Curie point meaning that a ferromagnetic or a ferromagnetic material becomes paramagnetic on heating.

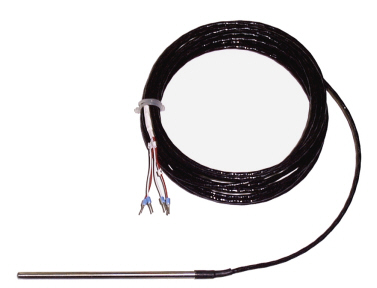
Type C is made from Tungsten and Rhenium; they are suited to measure 0 to 2310 Celsius range. This thermocouple is usually used in the lab because it must be operated in a vacuum or under inert gas, which is a non-reactive gas used during chemical synthesis. Hot tungsten burns when exposed to oxygen, this would happen if when a glass light bulb breaks.

Type M thermocouples are made from nickel alloy wire. There are used for vacuum furnaces for the same reason as the previous one. The upper temperature is limited to 1400 Celsius and is less used than the other types.

Chromel-gold/iron is used for cryogenic purposes and the sensitivity depends on the iron concentration. The lowest temperatures it was measure vary from 1.2 to 4.2 Kelvin.

**Resistance Temperature Detectors (RTD)**

Resistance Temperature Detectors or RTDs for short, and are also known as Platinum Resistance Thermometer or PRT for short. They are made of wire and thin film devices that measure temperature because of the physical principle of the positive temperature coefficient of electrical resistance metals meaning the hotter it becomes the bigger the value of the electrical resistance meaning that the impedance of certain metals alters the way the temperature falls and rises. They are slowly replacing thermocouples in industrial applications that need them for 600 Celsius of less because of their accuracy. They are usually used in air conditioning and refrigeration, food processing, stoves and grills, textile production, plastics processing, petrochemical processing, micro-electronics, air, gas and liquid temperature measurement. RTDs are wired in several ways from the simplest being two-wire to for wires. Figure 17 below displays a RTD.



**Figure 17 -** RTD

RTD have some advantages one being that they provide a reliable output over a period. The calibration is easier that other devices and they offer accurate reading for narrowed temperature. They have low drift. They have a wide operation range. Because of this RTDs should be used when accuracy and stability are highly important, when the accuracy has to be over a wide range and when standardization is wanted.

Some disadvantages is that they most inexpensive one are still somewhat expensive. They do have limitations to being used at higher temperatures than 660 Celsius. They do have errors the source of these errors are interchangeability, insulation resistance, stability, repeatability, hysteresis, stem conduction, calibration/interpolation, lead wire, self-heating, time response and thermal EMF. They have no point sensing, they are affected by shock vibration and they require usually three to four wire operation.

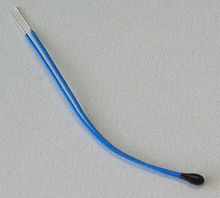
**Thermistors**

Thermistors are like thermocouples and RTD’s in some ways. In Figure 18 below shows the symbol for a thermistor. They are used as temperature sensors, and heating elements. They are resistors that vary due to temperature. They are made out of different materials then RTD’s. RTD’s are usually made out of metal materials, while thermistors are made from ceramic or polymer. Another difference is the temperature response. RTD’s can reach higher temperatures, unlike thermistors which have high accuracy with a certain range of temperatures. There exist a relationship between resistance and temperature. Which is linear and can be described in this equation: ΔR= k Δ T.

[](http://en.wikipedia.org/wiki/File:Thermistor.svg)

**Figure 18 -** NTC Thermistor

There are two types of thermistors. The first type is called a positive temperature coefficient (PTC) thermistor. To get this first type the coefficient k must be positive, which means the resistance increases while the temperature increases too. The second type is called the negative temperature coefficient (NTC). To get this second type it is vice versa. The k must be negative which means the resistance decreases while the temperature increases too. In Figure 19 below a NTC thermistor is shown

[](http://en.wikipedia.org/wiki/File:NTC_bead.jpg)

**Figure 19 -** NTC Thermistor (waiting on permission from USsensors.com)

**Surface Temperature Sensors**

Measuring temperatures is very hard, but surface temperature sensors can perform the job. They are very accurate. They have rapid response time. They are ideal for situations that require low thermal mass of flexibility. In Figure 20 below a surface temperature sensor is shown.



**Figure 20** - Surface Temperature Sensor (waiting on permission from USsensors.com)

**ENVIROSEAL™ HD Sensor**

This is another temperature sensor. The main advantage of this sensor is the ability to block out environmental factors, such as moisture, oil, and other contaminants. It can also provide accurate measurements in very movable environments. In Figure 21 below an ENVIROSEAL™ HD Sensor is shown

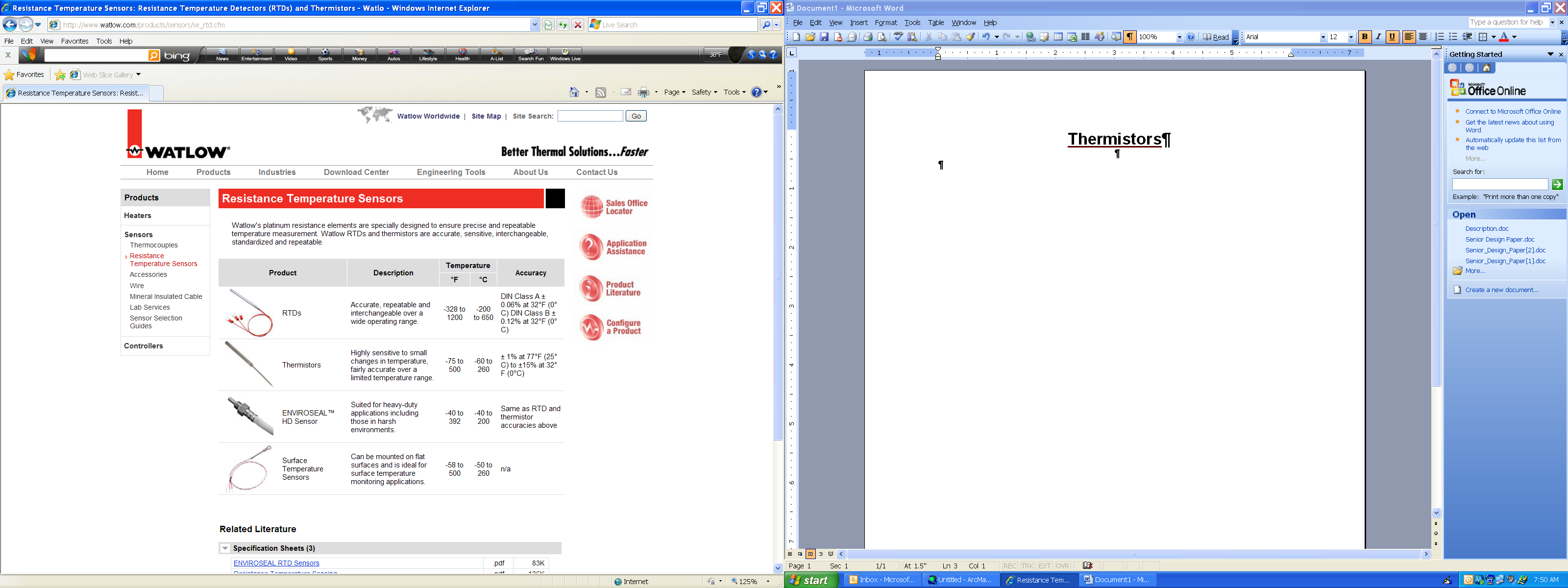


**Figure 21-** ENVIROSEAL™ HD Sensor (waiting on permission from USsensors.com)

**Resistance Temperature Sensors**

The following table 2 below shows a quick overview of the different types of thermocouples with max temperature and also accuracy percentages. Temperature may be controlled externally but we must still be able to read and notify the user of any malfunctions. Also our models must be able to withstand extreme pressures and heat.

As you can see the RTD type temperature sensors will allow for the largest range of temperatures at a reasonable accuracy level. Thermistors and the HD sensor have the low range available.

****

**Table 1 –** comparison chart of different temperature measuring components (waiting on permission from ussensors.com)

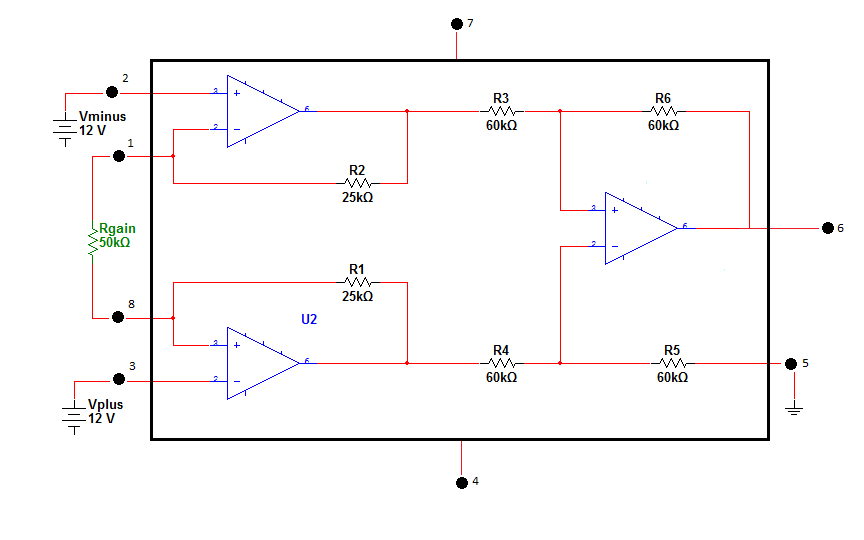
**Amplifiers**

**INA 118**

One of the primary properties of this amplifier is that it only requires one input resistor to produce a gain. It used a series of unspecified amplifiers to produce large gains. The gains can be anywhere from unity (Av=1) to ten thousand (Av=10000). The selling point of the amplifier was that is high accuracy, low power amplifier. It also provides good internal protection that will still work with voltages up to 40V without damage. It allows us to produce very large output signals without risking damage to the amplifier as a whole.

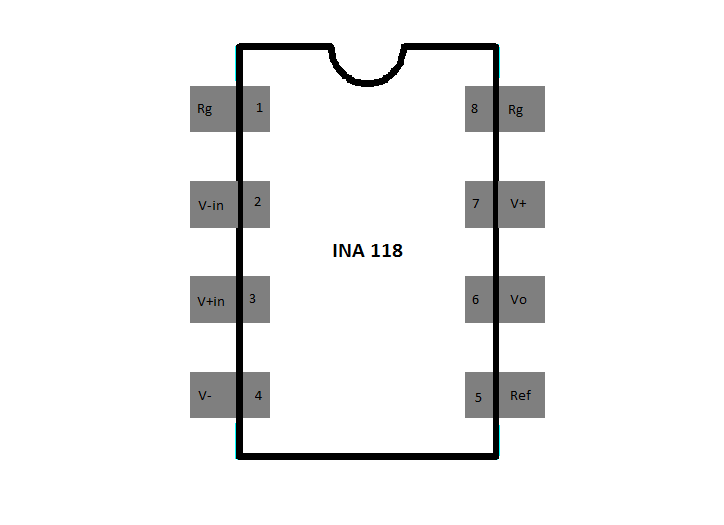
The INA 118 is an instrumentation amplifier (figure 22) that is structured to give gains from 1 to ten thousand (10,000) on the difference between the two inputs on the amplifier. The design has 3 internal amplifiers and 6 internal resistors to produce different the different gains with the equation Av = (1+2R1/Rgain)\*(R3/R2) where R1 = 25Ω, R2 = 60Ω and R3 = 60Ω. This allows for multiple set-ups.

The good thing about this particular circuit is that you can get a variety of gains just by changing the value of Rgain. This determines how much gain you can achieve and other than the voltage inputs of V1 (V+) and V2 (V-). This is achieved by using multiple internal operational amplifiers and internal resistors. the



**Figure 22 –** schematic of INA118

In figure 23 below, it is shown the exact pin layout of the INA118 instrumentation amplifier. As shown before, the resistor Rg is strewn back into the circuit as it is the primary factor in the gain. The Vo would go directly to the data-logging device.



**Figure 23 –** pin layout of INA118

**LF 351 (non-inverting amplifier)**

This operational amplifier is classified as a low noise, high speed JFET operational amplifier. This is also the amplifier that the researcher is most familiar with through class. With this class of amplifier, you must design your gain from scratch. With the basic set-up of a non-inverting operational amplifier, the gain would be set with the equation Av = (1+R2/R1).

All of these amplifiers would need to have a gain of one thousand (1,000) to be effective of the amplification of the sensors. With the INA 118, with the use of the speculation sheet, the gain of 1,000 can be achieved with the addition of a 50K resistor for Rgain. With the RF 351, it is wise to not use a resister value below 1K. Therefore, we will set the value of R1 to 1K and the value of R2 to around 999K or roughly 1M. These values will give the proper gain for the non-inverting amplifier.

Due to the smaller values necessary for the resistor values, we will choose the INA 118 because the resistor values necessary for the proper amplification of the circuit are less and because overall, less resistors are needed. When we have more resistors, there is more room for error in the circuit

**AD 595**

This amplifier will be used for the thermocouple. It will run thru from the thermocouple to the microcontroller. It can run from a voltage supply of 5 volts to 15 volts. This amplifier will be able to match up to the microcontroller. This amplifier is able to measure temperatures below zero degrees Celsius. This amplifier is already pre-calibrated and pre-trimmed to fit up to a Type K, which is the type we are implementing in our build. The AD595 can be used with Type T thermocouple inputs. This amplifier also comes with an extra feature, which is a failure alarm. The alarm comes on if any of the thermocouple leads are open. It also has a 10 mV for each volt in the output.

**Motors**

**AC Motors**

An AC motor is an electric motor which is powered thru alternating current.  The motor is based upon the rotary field of force. The usual AC motor is consisted of two parts. The first part is an outside stationery stator, which is comprised of the coils where the AC current is supplied, to in turn create the rotating magnetic field. The second part is the rotor which is located inside on the shaft, which provides the rotating field due to the torque supplied. The two main types of AC motors are synchronous motors, and induction motors. The induction motor has become the standard AC electro-magnetic motor. The synchronous motor rotates at the supplied frequency.  The magnetic field on the rotor is caused by current traveling thru the slip rings or by a permanent magnet.  The second type of AC motor is the induction motor. It rotates at just a little bit lower than the supplied frequency. The magnetic field on the rotor is caused by an induced current.

**DC Motors**

A DC motor is an electric motor which is powered thru direct current. There are many different types of DC motors such as brushed, synchronous, brushless, and uncommutated.  The DC brushed motors produce torque thru the DC power supplied to the motor.  The internal commutation, stationary permanent magnets, and rotating electrical magnets all follow the Lorentz force principle to make that happen. The low initial cost, high reliability, and simple control of motor speed are the advantages of the brushed DC motor. The high maintenance and low life-span for high intensity uses are the disadvantages that come along with the brushed DC motor. The synchronous DC motors consist of the brushless DC motor and the stepper motor. Both of the motors must have external commutation to generate torque.  The brushless DC motors consist of a rotating permanent magnet in the rotor, and stationery magnets on the motor housing.  The long life span, little or no maintenance, and high efficiency are the advantages of the brushless DC motor. The high initial cost and more complicated motor speed controllers are the disadvantages that come along with brushless DC motor. The uncommutated DC motors consist of not having commutation.

With a DC motor that only specs controlled are the revolutions and torque. The main advantage of a DC motors are that they can operate at very low voltages. Also, DC motors are more accurate for torque control then AC motors.  Another advantage is the designs used with a DC motors are more cost effective. DC motors have a faster response time compared to a stepper motor.

The main disadvantage of a DC motor is the required maintenance of the commutator and brushes. It requires this; because the surface needs to be well conducting or else it will wear the brushes down and also damage the commutator.

**Gears**

**Worm gears** are used when large gear reductions are needed. The worm gear can slow the rotational speed or allow higher torque to pass. The advantage of the worm gears is that worm can turn the gear without any problems, but the gear can’t turn the worm. This is possibly due to the angle of the worm gear, the worm being so shallow, the friction between the gear and the worm holds the worm in place. This is useful, because it acts as a braking system, which can reduce speed in less space than many other types of gearing. These gears provide large speeds ratios in comparatively short center distance, when they are properly installed meaning they are lubricated and properly (lubrication is a very important factor in to improve the efficiency) mounted they are the quietest and smoothest type of gears. Figure 24 below displays a worm gear setup.



**Figure 24 -** Worm Gear

The efficiency of worm gears depends on the helix angle and the type of material used. They are usually made from metal or plastic and a combination of metal and plastic, some materials they are usually made from are: brass and steel the most common also nylon and iron among others. These materials are used depending on what the application requirements since each have certain properties like how long they last, power transmission, noise, the amount of heat they can take and corrosiveness.

There some disadvantages of worm gears, one is safety were the worm gears can cause harm if used to lock mechanism that hold heavy loads because their reversing action. Another is lubrication, as mentioned before lubrication is very important, so these gears have to stay lubricated since their movement is dependent on the wheel gear faces sliding. These gears are not easy to lubricate and require lubricants with very high viscosity which makes them hard to filter. Also since these gears are usually made specialized for what they do they require a special lubricant to always be there depending on the equipment.

**Stepper Motors**

A stepper motor is a DC motor. There are several factors that affect the torque generation. The first one is the step rate. The second one is the drive current in the windings. The third factor is the drive design.

The stepper motor has many advantages to itself.  The first advantage is the motor still has full torque at a standstill position. The stepper also has precise positioning of movement, which makes them ideal for most applications especially digital applications. With the stepper motor comes along a great response to starting, stopping, and reversing. Another advantage is the life expectancy of the motor is longer. Another main advantage is the simple and cost effectiveness control of the motor due to the digital input pulses provided by the open-loop control. The stepper motor does not require tuning to obtain the correct motor function, they pretty much require to be plugged in and the wires to be wired to the motor driver and are very simple in design requiring no set replacement of part like other motors were for example brushes haven’t to be replaced. With the stepper motor it is possible to acquire low speed rotations. They are also more reliable and the errors in the positioning of the stepper do not add up and are less likely than other motors to be damaged by a mechanical overload.

There are two main disadvantages of the stepper motor. The first is if the motor is not controlled right than resonances can occur. The second one is the motor is hard to operate at high speeds. Some other disadvantages are that precise movement that is created by snapping the rotor from one position to another, which is called a detent, causes vibration making the stepper motor noisier than some other types of motors. Also they consume a lot of power given what their output is.

Steppers are generally commutated open loop also called non-feedback controller which as the name states there is no feedback. This means that the driver has no feedback on where the rotor actually is.  Open-loop types of controller computer its input using the current state. Because the Stepper motors do not have “memory” especially if the load inertial is high or there is a widely varying load they have to be over engineered if they are not then there is a risk that the motor will lose steps. Also because it cannot learn because it relies on what’s going on now, the motor cannot compensate for disturbances in the system.

There are different types of stepper motors. The variable-reluctance stepper, permanent-magnet stepper, and the hybrid synchronous stepper are compared in the table below.

|  |  |  |
| --- | --- | --- |
| **Permanent Magnet Stepper** | **Hybrid Synchronous Stepper** | **Variable Reluctance Stepper** |
| Higher Speed that HS | Lower Speed than PM |  |
| Perform well in close loop | No not perform well in close loop |  |
|  | Easier to use than PN |  |
|  | Maintain accuracy in open loop mode |  |
|  | Maintain reliability in open loop mode |  |
|  |  | Rotor that turns through a number of degrees and then stops |
| Permanent magnet rotor |  | Non-permanent magnetic poles on the ferromagnetic rotor |
|  |  | Has soft iron rotor with teeth or poles so positioned that they cannot simultaneously align with all the stir poles |

**Table 1** – Pro and Con list

**Actuators**

An actuator is a mechanical device used for moving or controlling a system. They are usually used when there is a need for circular applications. They operate by way of a source of energy, which can be from an electric current, hydraulic fluid pressure, pneumatic pressure.  Once receiving this source of energy the actuator transfers it into a form of motion. Many of them have more than one source of energy. Actuators are typically used in motors. They also can produce linear motion, rotary motion, or oscillatory motion. When the actuator is creating a linear motion that means it is creating motion in one direction. For rotary motion it is creating motion in a circular motion and as for oscillatory motion it is creating motion in the opposite directions at regular intervals. A linear actuator is also an actuator. It is impelled by a non-linear motion, which creates the linear motion or motion moving in one direction. There are different types of actuators. The mechanical, electro-mechanical, linear motor, piezoelectric, hydraulic, wax motor, segmented spindle, and MICA are compared in the table below.

|  |  |  |
| --- | --- | --- |
| **Actuator Type** | **Advantages** | **Disadvantages** |
| Mechanical | It is cheap. Also, it requires no power source. It is has identical behavior for extending or retracting. | It is manual operation only. |
| Electro-mechanical | It is cheap. It can be operated by automated. It had identical behavior for extending or retracting. It also has position feedback. | It has moving parts which tend to wear. |
| Linear motor | It has a simple design. Not a lot of moving parts. Can reach high speeds. It has Identical behavior for extending or retracting. | It has a low force. |
| Piezoelectric | It has real small motions. | It needs position feedback to be repeatable. It has short travels and low speeds. It needs high voltages and is expensive. They are only good for compression. |
| Hydraulic | It has real high forces. | They can leak. It needs position feedback to be repeatable. It requires an external hydraulic pump. Mostly only good for compression only. |
| Wax motor | It has very smooth operation. | They are not as reliable as others. |
| Segmented spindle | It is real compact. Also, the range of motion is greater than the length of actuator. | They have both linear and rotary motion. |
| Moving coil | It can control force, position, and speed. Also, can do high speeds and precise positioning. It also can have linear, rotary, and both motions. | It needs position feedback to be repeatable. |
| MICA (moving iron controllable actuator) | It has high forces, which can be controlled and controllable. | The strokes are limited. |

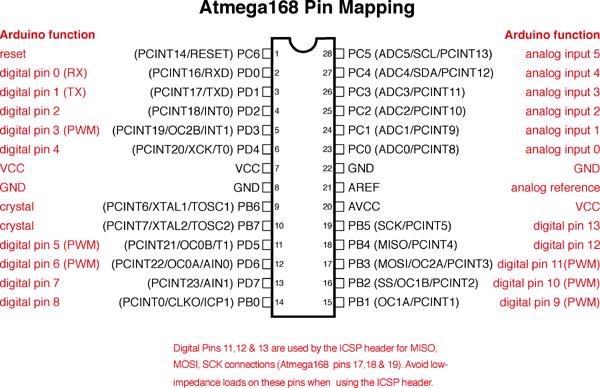
**Table 2 –** Actuator pro/con list

**Processor Type Comparison**

While doing research and through class experience we discovered three methods to operate as the brains of the Automatic Tensile Tester 4000. The three control methods we researched and considered were an 8-bit microcontroller, microprocessor, and field-programmable gate arrays (FPGA). We ranked our selection of technology based on ease of programming, cost, power consumption, I/O capabilities and peripheral availability and capabilities.

**Microcontroller**

Microcontrollers are special purpose microprocessor with limited functionality and low power consumption devices (Heath). They are used in automating and controlling most products such as microwaves, alarm clock, and embedded devices. These devices use a range from 8bit to 32bit instruction sets and come equipped with build in analog to digital converter, 10 to 20+ programmable I/O Lines, and as clock speed range from as low as 4MHz to as high as 20MHz. They come in DIP and surface mounted designs, there also a variety of peripheral such as thermocouples, H-Bridges for motor control, and Ethernet Chip that can be in easily interfaced with a microcontroller. Development kits range from $50 to $100 or a homemade development board can be made for less than $20. Figure 25 below shows an atmega168 pin mappings as you can see this IC consist of 10 analog-to-digital converters and 23 possible digital pin outputs or inputs.



**Figure 25 –** pin mapping of atmega168

***FPGA***

FPGAs usually come in 100+ pin chip, and are highly versatile. There inner designed consist of thousands of logic gates connected together, the basic logic cell of a FPGA “which consist of a small lookup table which is used to perform logic functions, a D-flip-flop and a 2-to-1 mux*”* (Jean P. Nicolle) using hardware description language (HDL) these logic gates can programmed with Verilog into different configuration and interconnections in order to perform different algorithms and instructions. A useful development kits ranges from as low $50 to as high as $2000.The FPGA Power consumption is usually 5V to 3.3V. Peripherals are highly available but even with some basic Verilog experience gained in class; the learning curve is extremely steep.

**Microprocessor**

Microprocessor devices are usually the heart and brains of computers and most smart phones; a microprocessor incorporates most or all of the functions of a computer's central processing unit (CPU) on a single integrated circuit (IC, or microchip)(Weik). Embedded Microprocessors typically can run Linux-based Operating system and can be programmed with a variety of high and low level programming languages such as C, java, python, etc… Peripherals such as an Analog-to-digital Converter can typically be interacted through USB, serial, and parallel ports. Microprocessors usually are high powered and costly, dissipate large amounts of heat when active. A useful microprocessor device such as the beagle board cost $149 for the base device which is fully functional with a 5v power supply; the beagle board which is powered by an OMAP 3530 Microprocessors.

**Design Selection Microcontroller**

The group decides to go with a microcontroller as the brain of our design, even though it is not the most powerful of the three, its basic functionality and easy integration, makes it the best fit for our project. We looked into two microcontroller company microchips’ 8-bit PIC® Microcontrollers and AVR’s megaAVR and tinyAVR series. Once again we ranked our selection of technology based on ease of programming and prototyping, learning curve, cost, power consumption I/O capabilities and peripheral availability and capabilities. Both microcontrollers scored equally when it comes to cost, power consumption and peripheral availability. Our decision came down to ease of programming and prototyping, and our group’s current knowledge and experience with the AVR series.

**Window Comparator**

The Automatic tester 4000 will require the microcontroller to constantly read analog signals outputs, most these signals come out of the device in mV and must be amplified from the original form into a max 5 V signal for the microcontroller to interpret the data. Because of these we must protect the microcontroller for getting damaged if one of the analog signals comes in to high. The Window Comparator for sparkfun we allow us to protect analog to digital pins from reading a to high voltages window comparator.

Each analog source will pass through this device before it reaches the microcontroller, if the voltage is out of range it will be logged and the user will be able to check after the experiment were the error appeared in the experiment.

**Development boards and Chips analysis**

For prototyping and development the group looked into the Arduino series and alternatives. The Arduino series come in a variety of boards, each is based on the 8-bit Atmel AVR, megaAVR series, the development board consist of pin extraction which allow for peripherals to be easily be plug in and out, the alignment of the pins also allow for shield to be placed on top of the board. Arduino are USB powered and programmed through the RS-232 when the Arduino bootloader is used. Arduino are programmed using wiring language or direct avr-gcc which is c ported library and complier. Many alternative Arduino clones exist that include a variety of different hack and modification to the original design. One can also configure their megaAVR series using a breadboard and programmed and powered externally.

**Inter-Integrated Circuit devices**

The atemega328p come with 2 I2C pins that will allow us to connect up to 112 devices to its bus. I2C devices’ are that incorporate full systems such as temperature amplification, calculation and conversion into one device with a 2 wire interface one for data and one for clock.

A developer can set each I2C device with its own address allow for packet to only be sent to the device with the unique address. The beauty of these devices is that most complicated circuits are simplified into a single chip.

The Trinamic TMC222 Microstep Stepper Motor Controller can be used as a middle interface between our microcontroller and actuator. I2C EEPROM - 256kbit chip can be use extend programming space and for data logging.

**Arduino Series**

The Arduino development board prices range from $30 to $50.Table 3 shows a quick over view of the Arduino Series with processor included. Each board clock speed is set to 16 MHz and is plug\_and\_play and interchangeable. The Arduino is usually programmed using the Arduino Ide which uses a wiring based language or C port, which we will be discussed later.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Arduino](http://arduino.cc/en/Main/Hardware) | Processor | Flash KiB | EEPROM KiB | SRAM KiB | I/O pins | P  W  M | ADC input pins |
| [Diecimila](http://arduino.cc/en/Main/ArduinoBoardDiecimila) | ATmega168 | 16 | 0.5 | 1 | 14 | 6 | 6 |
| [Duemilanove](http://arduino.cc/en/Main/ArduinoBoardDuemilanove) | ATmega328 | 32 | 1 | 2 | 14 | 6 | 6 |
| [Uno](http://arduino.cc/en/Main/ArduinoBoardUno) | ATmega328 | 32 | 1 | 2 | 14 | 6 | 6 |
| [Mega](http://arduino.cc/en/Main/ArduinoBoardMega) | ATmega1280 | 128 | 4 | 8 | 54 | 14 | 16 |
| [Mega2560](http://arduino.cc/en/Main/ArduinoBoardMega2560) | ATmega2560 | 256 | 4 | 8 | 54 | 14 | 16 |

Table 3 - Types of Arduino Board – (http://en.wikipedia.org/wiki/Arduino - waiting for reprinting permmsions)

**Homebrews Development Board**

A drawback we discovered with using an Arduino development board was the extraction level from the actual components circuitry and component interconnections. This may decrease prototyping time giving us a smaller learning curve when it comes to hooking up sensors and other peripheral. But increases the time it takes to transfer our product to a PCB and/or breadboard. Realizing this we decided to make our own basic development board. Our Homebrews Development board consists of a RS-232 port using a max232 for serial communication between our pc and microcontroller, pin outs extractions for peripheral interconnections, and an ICSP header for uploading programs.

The minimal components and a serial out are needed to communicate and debug, also it includes is an ICSP header for quick programming. Because the homebrew board was both cheap and will allow the team to go from development to production with minimal down time, we decided it would be the best for development and prototyping.

We also decided to go with the Atmega328p as our development and production chip, if a downgrade is need we will be able to interchange the atmega328p with an atmega168 which contain less flash memory. The homebrewed development board was created with all components placed on a Universal Component PC Board from RadioShack.

**Data logging & External Memory**

**Data serialization**

We looked into several data serialization methods and formatting. Methods such as comma separated variables, XML, and Netstrings. Per request of UCF mechanical engineering department the product should log data in a simple format while being able to be imported into a standard program such as excel or open office spreadsheet. With this in mind we decided that for data serialization the Automatic Tensile Tester 4000 data should be collected in the comma separated variables (.CSV) format. In table 4 we list output data, max and min string size.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Output** | **Type** | **Min Length** | **Max Length** | **Format** | **Example** |
| Cycle | Unsigned Long +1 | 2 | 10 | ##### | 456 |
| Strain | Float +2 | 2 | 5 | ###.#### | 55.5 |
| Load | Float +2 | 2 | 10 | ###.## | 100.9 |
| Temperature | Unsigned int +1 | 2 | 3 | ### | 77 |

Table 4 - Output data size and format

Being that CSV files are essentially basic text file were each character is 1 byte, allows us to calculate data storage usage. Equations xx and xx shows how to calculate the HD capacity need to run a test with a known cycle number and the max cycles possible with HD capacity known respectively.

Equation 1- Total size in kB of output files given number of cycles (top)

Maximum number of cycle a drive can hold given drive capacity (bottom)

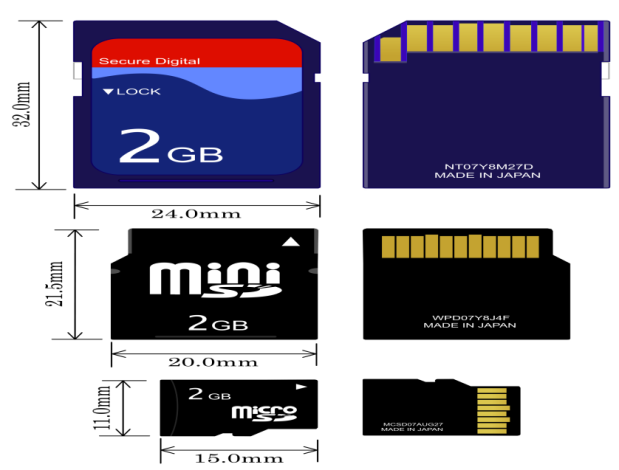
By specifications the auto tensile testing machine must be able to log data for an extended period of time without being connected to a computer terminal or network. In order to achieve this specification we would have to interface our system with a data storage device such as USB or SD / MMC Card. Our microcontroller would also have to implement a small file system that will give us the ability to connect the device into either the microcontroller or PC without doing data conversion. We ranked our selection of technology based on ease of interfacing with our microcontroller, file system implementation, storage capacity, and price.

**USB Storage**

The USB storage devices capacities range from a couple of Mbytes to ~2Tbytes. Most computers have plug and play interfaces to read and write to these devices using a native OS file system such as the FAT file system. To interface with the microcontroller one can use a “VDIP1 module which is a 24-pin DIP socket that provides access to all UART, SPI and FIFO interfaces” (Mouser Electronics, Inc.). The microcontroller implements read/write and drive navigation using serial input and output with a DOS like command like syntax. The VDIP1 module cost about $24.50 and an USB memory stick ranges from $7 to $300.

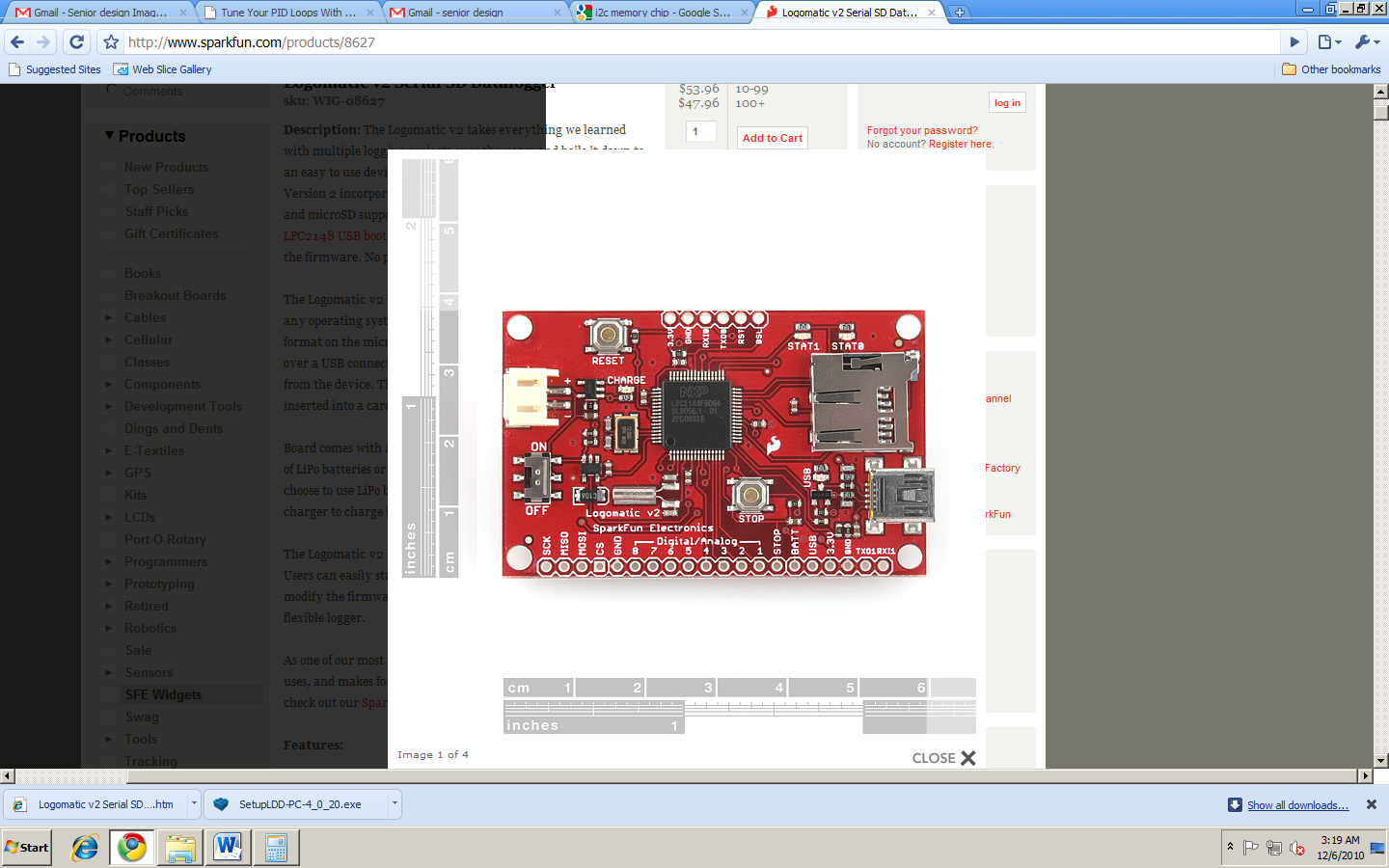
**SD / MMC Card**

SD (Secure Digital) Memory Card devices are used in many embedded devices such camera, and smart phones. SD cards are small memory device that hold up to capacity of up to 4GB and come in three size categories from largest to smallest they are SD, mini SD, micro SD. USB adapters are available for easy access to the SD device through a PC and most laptops come with SD card slot which allow for quick plug and play access to the SD device. Typically the smallest of the three which is the micro SD can be up converted to either a SD or mini through a physical adapter. Interesting enough one can use the same adapter to interface the SD card with the atmega328 microcontroller. One can either purchase an SD card module/breakout board ranging from $4 to $14 or solder connections directly on to the SD card or SD card adapter. Libraries are available to implementing a Fat16/32 file system on the microcontroller which will allow for easy swapping data between or device and a computer. The figure (26) below shows the three types of SD cards with size measurement.



**Figure 26 –** types and sizes of SD cards (

**Logomatic v2 Serial SD Datalogger**

Logomatic v2 Serial SD Datalogger (figure 27) is an easy to use device that incorporates both USB and micro-SD socket for data logging. It will connect to the serial interface of our microcontroller and can connect to a pc using the USB port, automatically being detected as a fat 16 file system. This device will allow the user to connect to the Automatic Tensile Tester 4000’s data information without disconnect the device or taking out the SD card. It can also log data from 9 analog device at a time, the firmware it open source and can be modified to fit our data logging need with a few hacks. Figure below shows the Logomatic v2 Serial SD Datalogger form spark fun. The device dimensions are 1.5x2.4" a small blue print which will not exceed our specification of small and portable. This device can interface with our microcontroller using the software serial library.

**Figure 27 –** view of Logomatic v2 Serial SD Datalogger (printed with permission from spakfun.com)

**Data logging Prototype and development**

The group decides to go with a SD card implementation for data logging on our device and also will implement a FAT file system using the Sdfat16 library available through the Arduino website. We decide to go with the SD card interface because of price and implementation. We were able to construct a prototype interface using a SD card adapter and an old IDE cable. The adapter will allow the group to run some preliminary data read and write tests using the microcontroller and sdfat16 library. For stability and in production our group may purchase a break out board to interface with our PCB layout. Transfer rate of SD devices range of 80–160 Mbit/s, which is more than enough speed for or device.

**User & Communication Interface**

Our device also must implement a basic graphical user interface, which will allow the user to, connect and disconnect to the device without restarting the system, read only access through a web interface and set test parameters to list a view. By specification the Automatic Tensile Tester 4000 must also include not only a GUI but also a physical interface. All though limited the physical interface must be able to pause and resume the system, and display statuses. A usage-to-interface table (5) is show below which lines up each specification with interface enable or not.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Specification | GUI | Web | Physical | Console (prototyping) |
| Start | X |  |  | X |
| Stop | X |  | X | X |
| Pause | X |  | X | X |
| Resume | X |  | X | X |
| Temperature reading | X | X | X | X |
| Strain  Gauge Reading | X | X | X | X |
| Motor  Directional Status | X | X |  | X |
| Motor Control | X |  |  | X |
| Transfer Data | X |  |  | X |
| Set PID Constants | X |  |  | X |

Table 5 - Usage-to-interface

**Current GUI: TestWorks**

Test Works is the GUI that is used for the current MTS Insight 5 in the mechanical engineering lab. The software comes complete with all one need to operate the MTS machine, and is very customizable. Currently the software cost a price which student do not have the funds or not willing to pay in order to get one on one time with the software. Though TestWorks is customizable it is still not open source. As seen in figure 28, Testworks is a tab based, window-sectioned interface with giant push buttons for control.

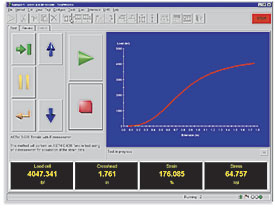


Figure 28 – Testworks 4 (waiting permission to reprint from MTS)

**Prospective GUI**

*Development Environments*

One minor specification is that the user interface should work in Windows and Mac environments. A Java based interface will allow for ease of portability and prototyping. The group plans on using the java Swing API; Swing is java’s primary API used for creating, programming, and positioning components on a canvas. Development would be done using NetBeans IDE: Java SE (standard edition) with Swing GUI Builder, which allows a drag and drop interface for placing and sizing components onto a canvas. For communication with the microcontroller the java communication API, this API will allow for OS native serial communication between the microcontroller using a USB to serial adapter implemented through a microcontroller peripheral interface or with and externally purchased adapter.

A C/C++ based interface, Qt by Nokia was also research as the SDK to develop our GUI. The SDK also allow for cross-platform compatibilities. Development would be done in Qt’s IDE Qt Creator, which allow for a drag and drop interface for component placement and window managing. Qt uses a signal/slot mechanism for connecting between objects. Slots are used to connect to objects and objects emit signals. For serial communication between the microcontroller and our GUI the Qt library QextSerialPort should be efficient. This software will allow the GUI to be used on Mac, Linux, and Windows platforms and allow for a high level of abstraction while developing.

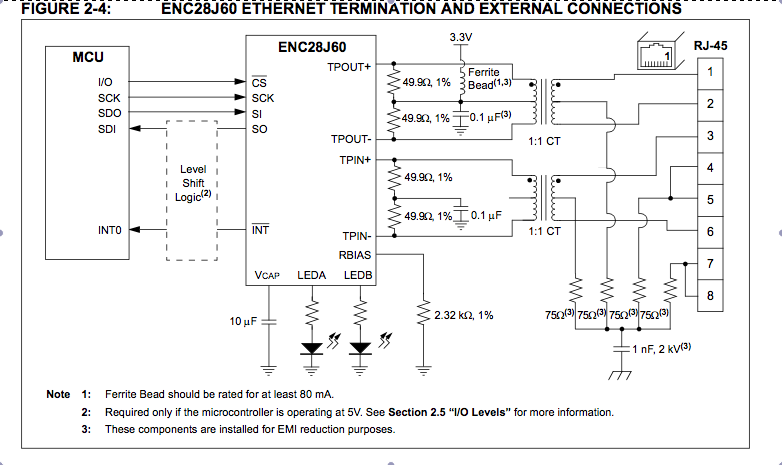
*Layout*

We decided to use a tab-based layout with each tab being a separate standalone widget. This will allow us to develop each specified functionality separate and then connect them together for production. Only one widget will be accessed at a time, this will allow for stability and reliability when communicating over the serial interface with the microcontroller. This tab based interface also allow for the end user to focus on one element of the system at a time, separates setup from operation and data logging, and minimize simple human error.

*Web Interface*

The automatic tensile tester 4000 also will include a read only web interface. Users will not be able to change any settings, or stop or pause the device. The reason for this is because the mechanical engineering department will not use the device with students in the labs but also use the device in production. An active web server would pose a security issue in the latter case, because present day malicious hacker’s exist who would love to destroy whatever they can. Seeing that each experiment may take months all precautions must be taken to not interrupt or corrupt the experiment. With this in mind we research two chips that would enable our device to do act as a web server Microchip ENC28J60 and WIZnet W5100.

The web server will show current status of the load cell and strain gauge, motor direction and current cycle.

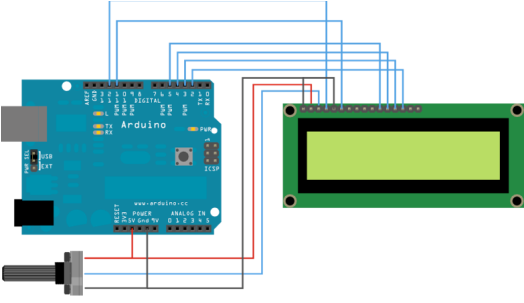
**For the automatic tensile tester 4000 we decide to implement our web server with the Microchip’s ENC28J60, in conjunction with Arduino Ethernet library. The ENC28J60 allows for a very basic TCP/IP stack implementation. The ENC28j60 come in a 20 pin dip chip, one needs a magjack Ethernet module which is a *RJ45 ethernet jacks with built in transformer (magnetics), status LEDs, and shielding* (sparkfun.com). Figure 29 shows the interconnection between the ENC28j60, microcontroller and Ethernet module.

**Figure 29 –** Interconnection ENC28j60, microcontroller and Ethernet

*Physical LCD interface with Push Buttons*

Lastly the automatic tensile tester 4000 by specification should continue to operate without being connected to a computer either through serial/USB or Ethernet. To accomplish this we decided to implement a LCD screen with a stop button, pause button, 4 buttons for menu navigation. A prototype layout of our LCD and button interface design does not include a start button incorporated in the physical design; the reason for this is that one must first setup the experiment before initializing the startup script.

Two different LCD interfaces for adding display to our project was considered, serial enable LCD or a traditional HD44780 parallel interface basic LCD. The main difference between the two is that the serial enable LCD connects through the microcontrollers RX pin while the basic traditional LCD takes 9 pins, 6 which interface with the microcontroller and 3 for power, ground, and contrast. A wiring diagram for the parallel LCD interface are shown below in Figure 30

**Figure 30 –** Parallel LCD connection to Arduino Board (waiting for permission from Arduino.cc)

For the project we decided to go with the serial enabled LCD mainly because of its ease of interfacing. But Instead of interfacing with the RX pin of our microcontroller we will use a digital pin out, leaving the RX available for communication with the GUI on the PC or MAC. For button we decided to use standard 4 pin push buttons, each pin will be solder with extension wires on to our PCB to the project box. Figure 31 below the diagram with measurement for the button we plan on using.

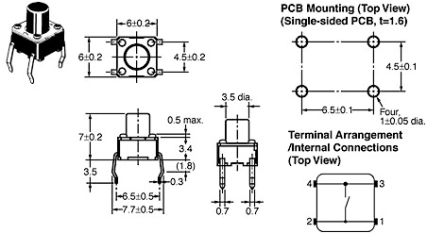


Figure 31 – Push button (waiting permission to reprint from www.sparkfun.com)

*USB to Serial*

For prototyping and development we will use a max232 and DSUB-9 interface to communicate in between our device and the development pc. In production we will use a USB to serial FTDI FT232RL breakout board. The FTDI FT232RL will allow the user to connect an USB mini cable to our devices when interfacing with the GUI on a PC. USB will be used in production rather than a DSUB-9 because USB comes standard on most computers, is more reliable, durable, and it’s usually plug and play. The alternative to using a USB to serial converter on the device side would be to use a serial to USB converter on the computer side. The FTDI break out board is shown in figure 32 below.

As see in the schematic diagram of the FTDI break out board the necessary RX and TX pin out are readily available for interfacing with our microcontroller. Also the board run at either 3 or 5 volts, and contains led which will allow the user to observe transmitting and receiving signals.

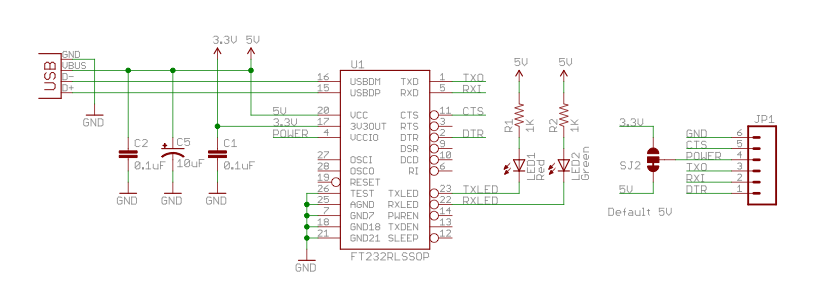


Figure 32 - FTDI schematic (waiting permission to reprint from www.sparkfun.com)

**Proportional–Integral–Derivative Controller**

The atmega328 will be the brain of the automatic tensile tester 2000 which will control running the motor, reading sensors, data logging, menu flow, and all algorithms and methods. We researched multiple methods for controlling each component of the system and will discuss our finding on motor controlling.

**Motor PID Control**

Linear control systems specifies two type of control systems, open-loop controller *which there is no direct connection between the output of the system and the actual conditions encountered; and closed-loop control system, a sensor monitors the output and feeds the data to a computer which continuously adjusts the control input as necessary to keep the control error to a minimum* (Wiki)*.* The automatic tensile tester 4000 will use a closed-loop control system for motor control when it comes to applying force to a test object. More specifically the commonly used PID feedback system, figure 33 show the basic block diagram of a PID closed loop system, which consist of the proportional, integral and derivative components.

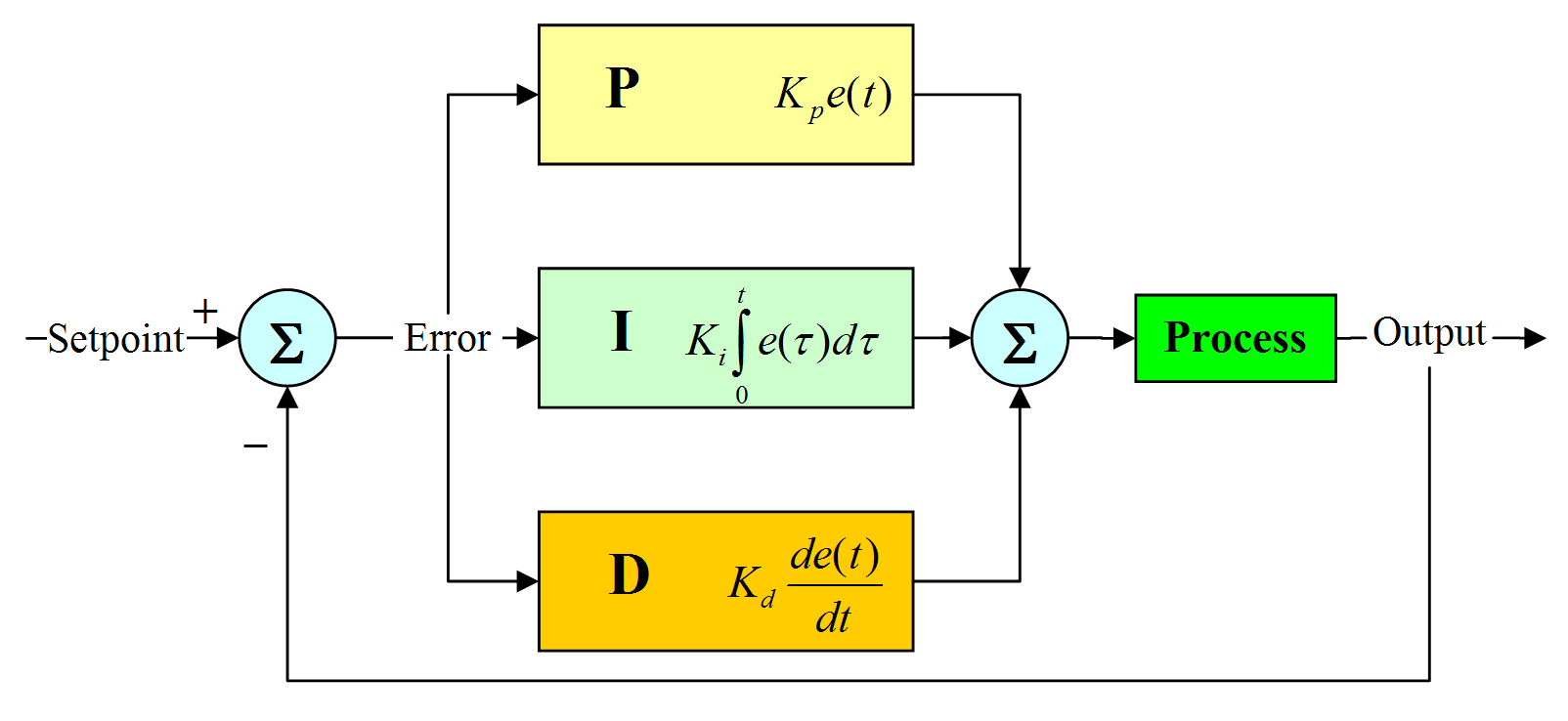


Figure 33 - FTDI schematic (waiting permission to reprint from wiki author SilverStar)

The load cell will be used to measure the pressure applied by the motor, and the execution file explained in the motor section, will determine the set point. *The proportional term is typically the error (Society of Robots)*, which in the automatic tensile tester 4000 would be the amount the load cell reading is off from the actual set point P = (desired force) – (load cell reading) . *The integral term is the accumulative error made over a set period of time (t)(Society of Robots),*in our case this would be I = sumof(error in load cell vs setpoint(t)/ t) *.* Finally *the derivative term is the change in error made over a set time period (t)(Society of Robots)*, which would be in our case (load cell measurement) i-1 - (load cell measurement) i / delta (t). Figure 34 shows a high level view of the flow chart of the PID interfaced with the motor, load cell and desired user input.



Figure 34 - FTDI high-level view of PID

As seen above all values are feed into the PID controller effecting the movement of the motor. PID controller can be either implement strictly through software or a mix of both hardware and software. For our project we took a look into both methods and decided which one was best to implement.

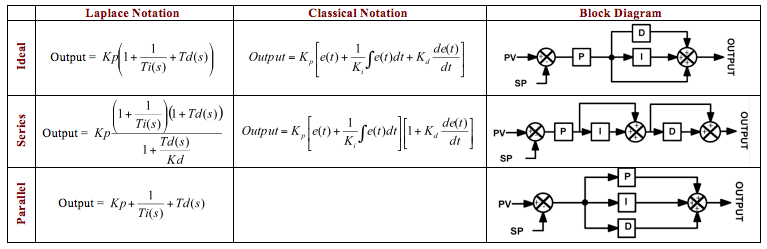
**Hardware \w PID**

The company Pololu makes a varitiey of motor controller board with PID control, the board we research was the Pololu Jrk 12v12 USB Motor Controller with Feedback. This model operates at a max volatage of 16V, interfaces with both usb and more importantly logic-level serial(for microcontroller), and as seen at the Jrk 12v12 includes a feedback input pin for PID controll. For communcation with the atmega328 the Jrk 12v12 Maximum logic voltage is 5v which should work perfect.

Pros with using a hardware enable PID motor controller are that most the upfront work is done from the board,prototyping can be monitored through the USB port and incldued software,and status light to indicate power, error and output.

Cons of using the Jrk 12v12 is that it cost $99.95 plus shipping which could hurt of requirement to be cheap and effective. If we wanted to upgrade our motor we are limited to 16v motors tha are capable with the controller. Finally the Jrk 12v12 might be an overload and restrict future development of the Automatic Tensile Tester 4000.

**Software PID**

****The second option we looked into was a simple algorithm that would allow us to implement PID controls using our microcontroller and an H-bridge .In order to begin development on a software based PID solution we look at the 3 general formats of the PID controller algorithm which are the ideal, parallel and series (interacting) each which can be seen in equations and block diagram from below in table 6.

**Table 6 –** PID Notation Diagram (waiting on permission from PIDtuning.net)

The only difference in the algorithms is how each element P, D, and I affect each other, which can be seen clearly in the block diagrams. Prototyping and development of the A.T.T 4000 will be done using the parallel algorithm because this will allow for a 1-to-1 analysis and trouble shooting of the PID control, each element can be removed and the effect of that element being removed and be seen directly in the output. Table 7 shows the effect that each elements and its constant have on the graph in Table 6, which is the response time of the PID controller.

The pros of implementing the simple parallel software based PID controller for the A.T.T 4000 is that it is cost effective, allow full control of our system, satisfy the open source requirement allowing the PID control to be modified in the future, and finally the hardware used to drive the motor should have no bearing on the

PID controller. The main downside of developing an in house PID software based controller is that it will greatly increase our development time.

We chose the software based PID controller option mainly because of price and the ability to modify it in the future if needed. A more detail review of the algorithm with pseudo code can be found in the design section.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | **Rise time** | **Overshoot** | **Settling time** | **Steady-state error** | **Stability**[[3]](http://en.wikipedia.org/wiki/PID_controller#cite_note-2) |
| ***Kp*** | Decrease | Increase | Small change | Decrease | Degrade |
| ***Ki*** | Decrease | Increase | Increase | Decrease significantly | Degrade |
| ***Kd*** | Minor decrease | Minor decrease | Minor decrease | No effect in theory | Improve if *Kd* small |

Table 7- PID constant and effects on reaching steady state

**System Flow**

By specification the A.T.T 4000 should be able to run 3 basic tests, which are waveform test, constant force and linear force. While researching and prototyping a single complex method that would implement each test automatically, we decided it would be best to development 3 basic methods. We include a high level description of each method with explanation below. Table xx show the pros and cons of single and multiple method approaches.

|  |  |
| --- | --- |
| **Single** | |
| PROS | CONS |
| Require less programing space | Hard to troubleshoot for single case |
| Highly dynamic | Extended development time |
| Non dependent on setup file | Complex algorithms |

Table 8 - pros and cons of single method approach for experiment execution.

|  |  |
| --- | --- |
| **Multi** | |
| PROS | CONS |
| New method can be added without effecting others | Setup File must be correct |
| Troubleshooting case by case | More writing more space |

Table 9 - pros and cons of multiple method approach for experiment execution.

|  |  |
| --- | --- |
| Can be developed quicker |  |

**Applying a Linear Pressure**

For the linear force test the A.T.T will apply a greater force each cycle till the object either breaks or reaches equilibrium. As seen in figure 35 this user must input that the test will be a continuous test. This criterion will be in the header of the input file or maybe set in the GUI. Pausing and data logging are not show in the figure but will be address later in the implementation in design section XX.

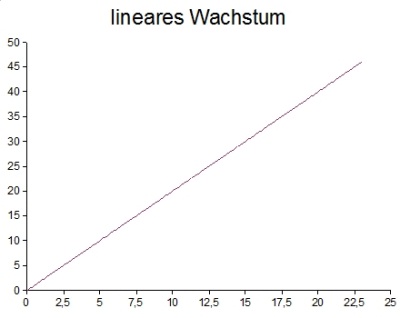


Figure 35 - block diagram of processing countinous increase pressure on a test speciemen(left) with corresponding graph (right)

*Constant* *Load*

For the constant load test the A.T.T 4000 will apply a load and hold it for a given number of cycles till the object either breaks or the cycles reached. This Load constant and number of cycles will be in the header of the input file or maybe set in the GUI. Data logging taken at 1 Mhertz/s. A flow diagram and graph can be seen in figure 36 below.

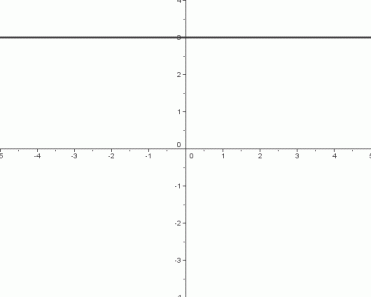


Figure 60 - block diagram of processing constant pressure on a test specimen (left) with corresponding graph (right)

**Wave**

The wave for test consists of compressing and decompressing an object by adding and removing pressure set by user applied by the motor. This method was the prototype for the single method approach. The input file will consist of multiple load entries, which can be generated by an excel file converted to CSV form. The method will pass the desired load through the PID controller until the desired load in constant for X cycles then update the desired load value from the input file. This process will continue until the final input file is reach or the object breaks. Data logging will be taken at H hertz. A flow diagram and graph can be seen in figure 37 below. This is a high level design; a more in-depth view of the methods, function, class, etc. will be discussed in the design section.

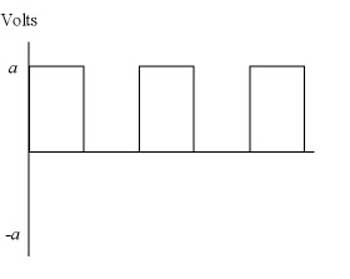


Figure 1 - blocks diagram processing pressures generated by a waveform on a test specimen (left) with corresponding graph (right)

**Power Sources**

To power the all the equipment, we will use a series of voltage regulators to transform the wall output of 120 volts to the various voltages we would need. We picked these voltages based on the specifications of each part. The load cell reported a maximum voltage of 18 volts for the excitation. We thought it would be easier to allow all of the sensors to have the same excitation voltage to prevent us from making multiple regulators for the input voltage. Each different excitation voltage level needs its own regulator. With the current set-up, only two regulators are needed.

We would use the following voltage parameter for our project. Each component and its required voltage are listed in table 10 below. This will allow us to design a effective power system with correct load parameters.

|  |  |
| --- | --- |
| 12V | 5V |
| Thermocouple | Microcontroller |
| Strain Gauge |  |
| Load Cell |  |
| Amplifier |  |

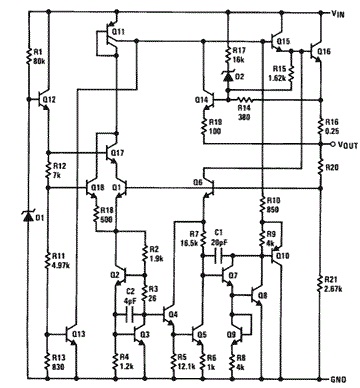
To achieve these varied voltages, we will use a wall wart (AC adapter) to change the output of the wall outputs to our circuits. The wart can be stripped and used as the positive and negative (ground) leads to the regulators that will be used to power the load cell circuits, strain gauge circuits and the microcontroller. The 12 volt line would be attached in parallel to the thermocouple, strain gauge, and load cell and those would be attached to a common ground. The 5 volt line will be solely attached to the microcontroller and will feed the negative terminal to the ground.

**Voltage Regulators**

The voltage regulators must be chosen in a number and a fashion that each component in our circuit receives a sufficient enough voltage from the regulator, but it doesn’t drive that same regulator too hard by drawing too much current. If too much current is drawn from any of the parts, then we could have a major system malfunction. We know that our microcontroller needs up to 5 volts to power it, and our load cell can take no more than 18 volts to power it. The strain gauge circuit is rated in an output to input ratio according to the Wheatstone bridge, so the input voltage shouldn’t matter, but we want a voltage signal that is large enough to be captured, read and analyzed by the microcontroller. Amplifiers take anywhere from 5 volts to 15 volts to run properly. Due to these aforementioned limitations

**78XX Regulators**

The LM78XX regulators (figure 38) are commonly used to introduce college students to the used of regulators. The reason for this is because they are very versatile, and can be used in a variety of ways including a fixed regulator and an adjustable regulator. Even with that, in some applications there is no need to use additional components because the regulator will output the specified voltage of that regulator regardless. These also have built-in protection against drawing too much power, overheating and short-circuits making them preferable choice in prototyping. If there is a failure in the system, neither the 78XX nor the other components will be harmed. A main disadvantage of the 78XX is that you can only step down the voltage as the input voltage must be at least 3 volts higher than the output voltage. This also means that 78XX regulator circuits are limited in the devices they can be powered by. Also the stepping down of the voltage means that there is power lost in the circuit somewhere, most likely in the regulator itself.



**Figure 38 –** regulator schematic

**Tri-Mode Regulators**

Tri-mode regulators are designed for portable battery use and applications that require precise output voltage, low supply current and high ripple rejection. Figure 39 shows how the comparators and the decoder allows for the tri-mode switch between its three modes, chip enable mode, fast transient mode and low power mode. This is will help with controlling power consumption from a battery while still allowing the system to continue logging data under non-optimal conditions or undesirable conditions such as power surges, human error, and storms thing system will give the Automatic Tensile Tester 4000 the ability to shutdown safety without damaging any components or lose valuable data.

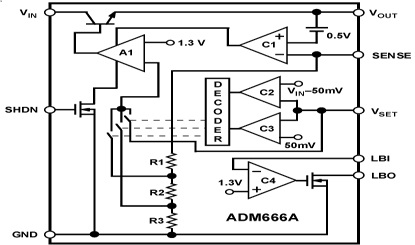


Figure 39 – tri-mode regulator schematic will help with controlling power consumption from the battery while still lallowing the system

an the future. ich will allow for the amp to be replace or change when new and better model are avaiblen he GUI and what goes o

Because of the ease of use, ease of installation and the low cost of the component, we choose to use the LM78XX regulators. The biggest reason for this is that each one is a voltage controlled current source, so the regulator will only give as much current as needed to keep the output at the voltage that is specified on the part. The maximum current output of these components is 1 ampere, which is a large current. This is good because the output will be holding 2 to 4 components apiece.

We have four components that we plan to use at the excitation voltage of 12 volts coming from the 7812 voltage regulator circuit. This could draw more current than one regulator can handle. If that happens to be the case, we will add another regulator of the same type to assist in powering all of the components.

The wall wart itself must be greater than three volts higher than the output of the highest regulator, so the output of our wall wart must be no less than 15 volts and have a sufficient enough current to power the amplifiers. It also allows us to possibly have enough of a cushion with the output current of the wall wart, and it will not overload the regulators.

We found that the output of the 78XX regulators needed conditioning and smoothing so we researched smoothing methods and determined that we could use those two different polarized capacitors to smooth the signal to almost DC levels to the point that the components can’t tell the difference. This seems like it saves money in not having to buy more components as have been used before and making the circuit easier to put together.

The reason we have a 12 volt circuit and a 5 volt circuit is because we have different power requirements for each of the components. The 12 volt circuit can be as high as 15 volts or as low 8 volts but we decided to go with the middle ground.

We want the electrical power signals to flow in the following manner as shown in figure 40. All information in the system is carried by voltage fluctuations cause by changes in resistances of the sensing components. For this reason, the powering and accuracy of the voltage and power requirements is very important.



**Figure 40 –** electrical signal flow chart

This shows how the power will through the circuit. The voltages that will be fed into each of the sensors are read by the last two numbers of the regulator. The output of the 12 volt regulator will be input into the op amp, the load cell, and the strain gauge circuit.

The op amp will be powered so that we can amplify the output of the load cell that it may be registered by the microcontroller. The op amp only needs the power so that it may run properly. Without the voltage provided by the regulator, the op amp cannot operate.

The load cell requires the 12 volts as an excitation voltage. The output of the load cell is an electrical signal made to be input into a linear equation given the characteristics of the load cell itself. All that needs to be done is create a source for the load cell and apply the force. When the force is applied, a higher voltage will be output. If the load cell is rated 2 mV/V then the maximum output will be 2 mV x Vexcitation.

The strain gauge circuit will be a Wheatstone bridge design with the strain gauge resistor as the variable resister driving the output voltage. The voltage must be high enough for the output voltage to be read by the microcontroller, yet small enough not to overpower the regulator, wall wart, or strain gauge itself.

The wall wart itself must be greater than three volts higher than the output of the highest regulator, so the output of our wall wart must be no less than 15 volts and have a sufficient enough current to power the amplifiers.

**Design**

**Load Cell**

The data sheet (figure 41) for the column load cell tells us that the maximum load that the load cell can handle without recalibration. Also necessary is the max excitation voltage of 18 volts which we shall not exceed. The 2 mV/V output rating is to be manipulated by 12 volts and therefore leading to a maximum output voltage of 24 mV. For design purposes, the most important information is the wiring code. This is immensely helpful in completing the amplification and powering circuits for the load cell.

|  |  |  |  |
| --- | --- | --- | --- |
| + Excitation | - Excitation | + Signal | * Signal |
| RED | BLACK | GREEN | WHITE |



Figure 41 – Load Cell dimensions (waiting on permission from Futek)

The load cell receives its excitation voltage from the 7812 regulator and outputs to the INA118 Instrumentation amplifier. This process is shown in figure 42 below.



**Figure 42 –** 7812 regulator flow chart

The amplifier (figure 43) is to receive its input signals from load cell and magnifies the difference between the two inputs. A resistor value is used to set the voltage gain. The load cell output 2mv per volt seen on the excitation voltage side, the microcontroller only values in the range of 0 to 5 volts. The plus and minus signal lines from our load see goes into pins 3 and 2 respectively. The plus and minus excitation voltage does into pins 4 and 7. The output to the microcontroller analog to digital conversion pin comes from pin 6. Finally pin 5 will be grounded to the chassis of our project box.

The load cells will be interchangeable in the final production and each load cell gives a different voltage out. We will incorporate a potentiometer that will extend off our project box to allow for varying the resistance.

Our GUI interface will allow the user to check these resistance values; we may be able to use an electronic potentiometer in version 2 of the automatic tensile tester 4000.

Finally we will use a DIP socket to insert the INA 118 into our PCB which will allow for the amp to be replaced or change when new and better model are available in the future.

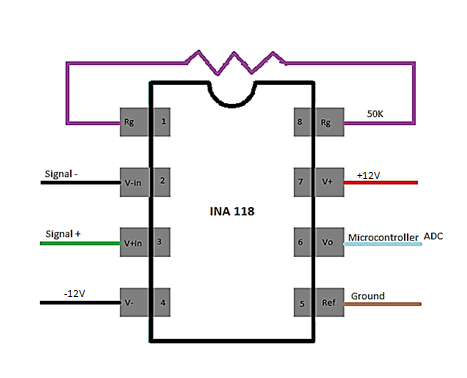


Figure 43 – INA118 Pin layout

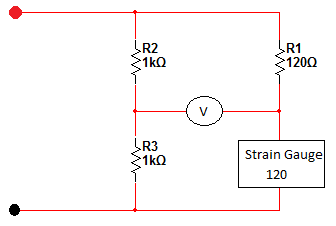
The following figure (figure) is a flow diagram detailing how the electrical signal travels to and from the amplifier. The resistor in the figure is shown to be feeding back into the amplifier. This is done in pins 1 and 8 to create the means for the gain. The load cell inputs the Signal + and Signal – inputs into 3 and 2 respectively, while the 7812 Instrumentation Amplifier is input to the 4 and 7 terminals. The output signal of the amplifier goes directly to the microcontroller where it will subsequently be analyzed.



**Figure 44 -** INA118 flow chart

**Strain gauge**

The strain gauge is in a quarter bridge Wheatstone bridge formation. The schematic (figure 45) of which is as follows. The resistor on the side with the strain gauge must be the same value as the initial resistance value of the strain gauge.



**Figure 45 –** values of strain gauge quarter-bridge

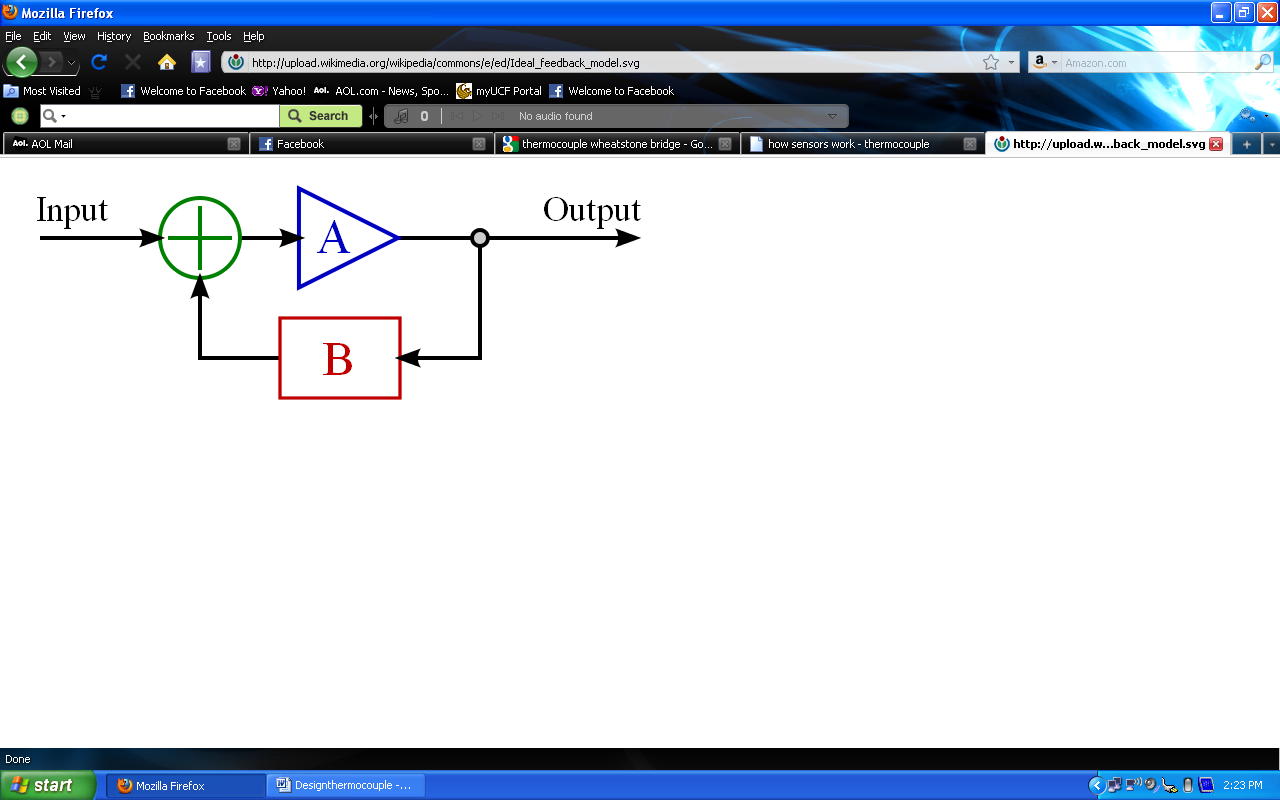
The following figure (figure 46) is a flow chart of the electrical signal. The filtered output from the instrumentation amplifier is the power source of the strain gauge circuit. The output of the strain gauge is then read by the microcontroller.



**Figure 46 –** strain gauge circuit flow chart

**Thermocouple**

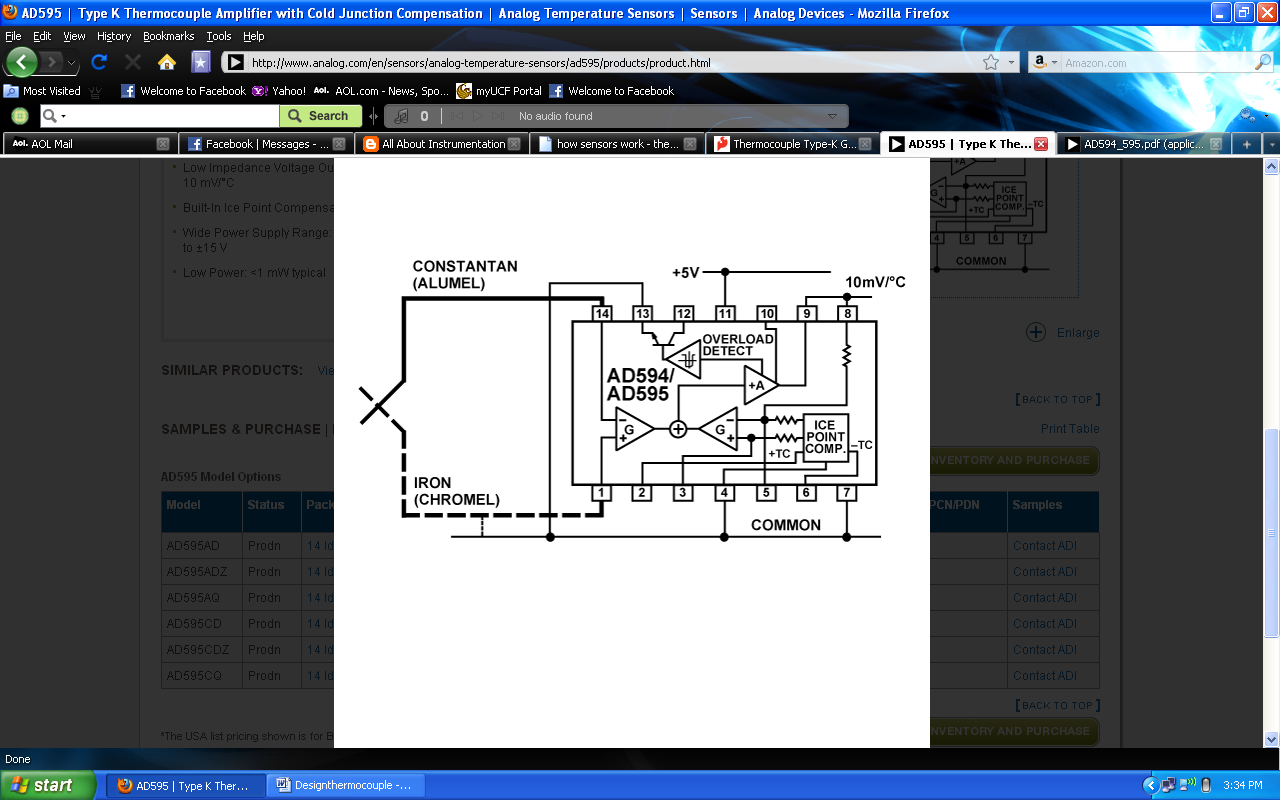
Our project calls for a thermocouple to be used. We decided to go with the thermocouple instead of an RTD. Due to the temperature requirements set by the sponsor. Also, because they generate a small voltage signal proportional to the temperature. The thermocouple has a small voltage output, so we have to amplify it. Since we will be cutting the thermocouple on and off we must use a Schmitt trigger type of circuit as shown in figure 47. These circuits usually come with amplification already.



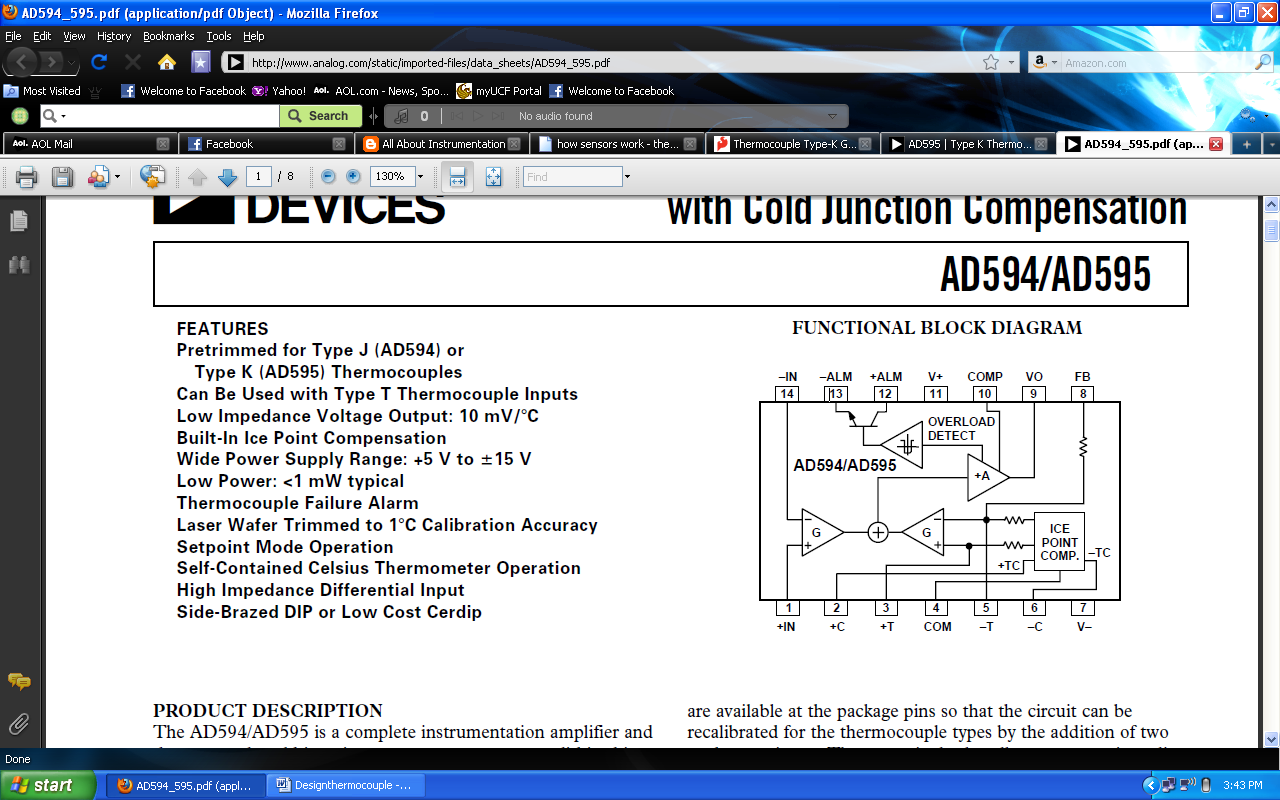
**Figure 47 –** Schmitt trigger for thermocouple (posted with permission from sparkfun.com)

To use the thermocouple right we must place one end of the thermocouple in contact with the process while the other end is at constant temperature. The one we have in contact with the process is now referenced as the hot while the other one is cold. There is a relationship between the total circuit voltage, hot and cold. The equations that explains that is: Circuit (emf) = Measurement (emf) - Reference (emf). When the circuit (emf) and reference (emf) are identified then the measurement (emf) can be solved and the relative temperature can be calculated. We will be using a Type-K glass braid insulated thermocouple from Sparkfun.

It will be connected to an amplifier which is called AD594/AD595 also found at Sparkfun. The configuration is shown below in figure 48a and 48b



**Figure 48a –** AD594/AD595 amplifier (posted with permission from sparkfun.com)



**Figure 48b -** AD594/AD595 amplifier (posted with permission from sparkfun.com)

The thermocouple will be connected into pins 14 and 1. Then 5 volts will be placed into pin 11. Then pin 9 will run into the microcontroller. The amplifier already has the needed resistors inside.

Motor (software)

In order to apply pressure to our test object we will use an actuator with the ability to apply 4000lb of pressure. The motor controller class will simply take care of moving the actuator forward and backwards. The motor controller class contains a PID\_Controller class for which the class diagram can be seen in figure 49. Because it is nearly impossible to apply an instantaneous amount of force to our samples we use a linear control system (PID class) that will take care of allow the system to reach the desired load for each cycle, with an acceptable error of plus or minus 5lbs. We will overview the design of the PID controller class and its important methods.



Figure 49 2- class diagram of PID together with motor

*Class Analogtodigital*

Void getP(), getI(), getD()

Input: necessary constants P,I or D according to functions

Function: computing Proportional Integral and Derivate constants for PID in section xx on motor control

Void SetOutput()

Function: Use P, I, D constant to figure out number of steps need to achieve the set point (desired Load )

Motor-Atemga328P Schematic

We will use an H-bridge (model: L293) as the interface between the atmega328p and motor x. The H-bridge is needed to control the direction of the motor and to apply the voltage needed to run the motor. Shown figure 50 is the circuit we will use to connect the motor to the H-bridge to the microcontroller. As seen in the figure 50, 5 volts will power vcc1 of the h-bridge IC and atmega328p, vcc2 will vary from 4.5 V to 36 V for the A.T.T 4000 we are using an x volt motor. The microcontroller will send 3 signals to the H-bridge using 3 pin digital pin an enable (active low), 1A and 2A.

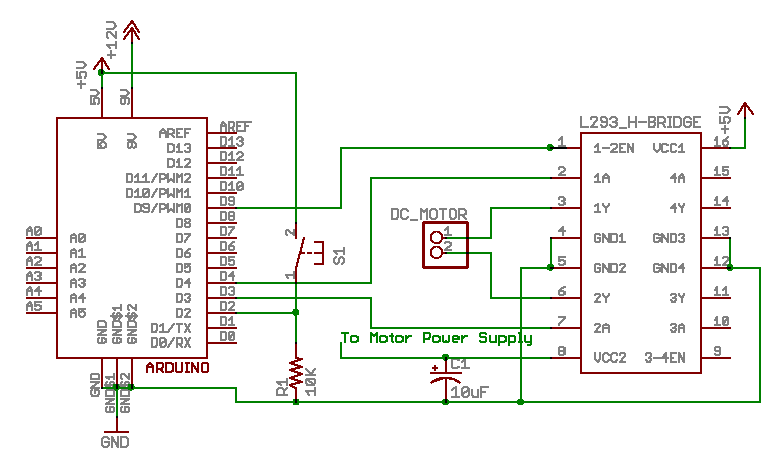


Figure 50 -3 Connecting Arduino H-Bridge and motor (printed with permission from sparkfun.com)

Analog Signals (software)

Each of our sensors for the A.T.T 4000 after being assembled standalone must be integrated in our microcontroller. In section xxx of our design section we discussed and showed diagrams of different circuits to get our various sensors to output the correct analog signal .The atmega328p come with 5 10-bit analog-to-digital converter pins, each of these pins can take a max of 5 volts as its input. The analog signal value received is a 10 bit number split into two 8-bit numbers, values ranging from 0 to 1023. Once we receive a reading from a particular sensor we must use the correct conversion equations stated in section xx, xx and xx for thermocouple, load cell, or strain gauges. The sensor class seen below in figure 51 consists of an abstract parent named “analogtodigital”, and three children classes “thermocouple”, “loadcell”, and “straingauge”. We will overview the design of each class and there important methods.

Figure 51 - class diagram of all analog to digital components

*Class Analogtodigital*

Void ReadValue()

Return: 16-bit number from 0 to 1023, which corresponds to the analog voltages, output from devices

Void ConvertToVoltage()

Input: value (the raw ACD value)

Return: Abstract class which will be used to convert the 16bit raw data to a voltage value in convertedValue(float) format XX.XX volts

*Class Thermocouple*

Void ConvertToVoltage()

Return: Converts raw ACD value into Fahrenheit (temp\_F) and Celsius (temp\_C). Conversion formula in section xx

*Class LoadCell*

Void LinearConversion(in X, in B, in M) Y = mx + b

Input: Values discussed in section XX for converting voltage to pounds

Return: Pound and Newton values corresponding to input voltage. Using linear Conversion formula that comes with load cell

Analog devices to Atemga328P

The schematic in figure 52 below shows the configuration of the 3 analog devices and the atmega38p. The load cell and thermocouple outputs require amplification before they can be sent to the microcontroller, this allow the signal coming out to max out at 5 volts. The amplification circuit show is a black box representation of the actually circuit show in previous sections. ADC ports on the atmega328p 0, 2, 4 receive signals from the load cell, thermocouple and strain gain respectively.

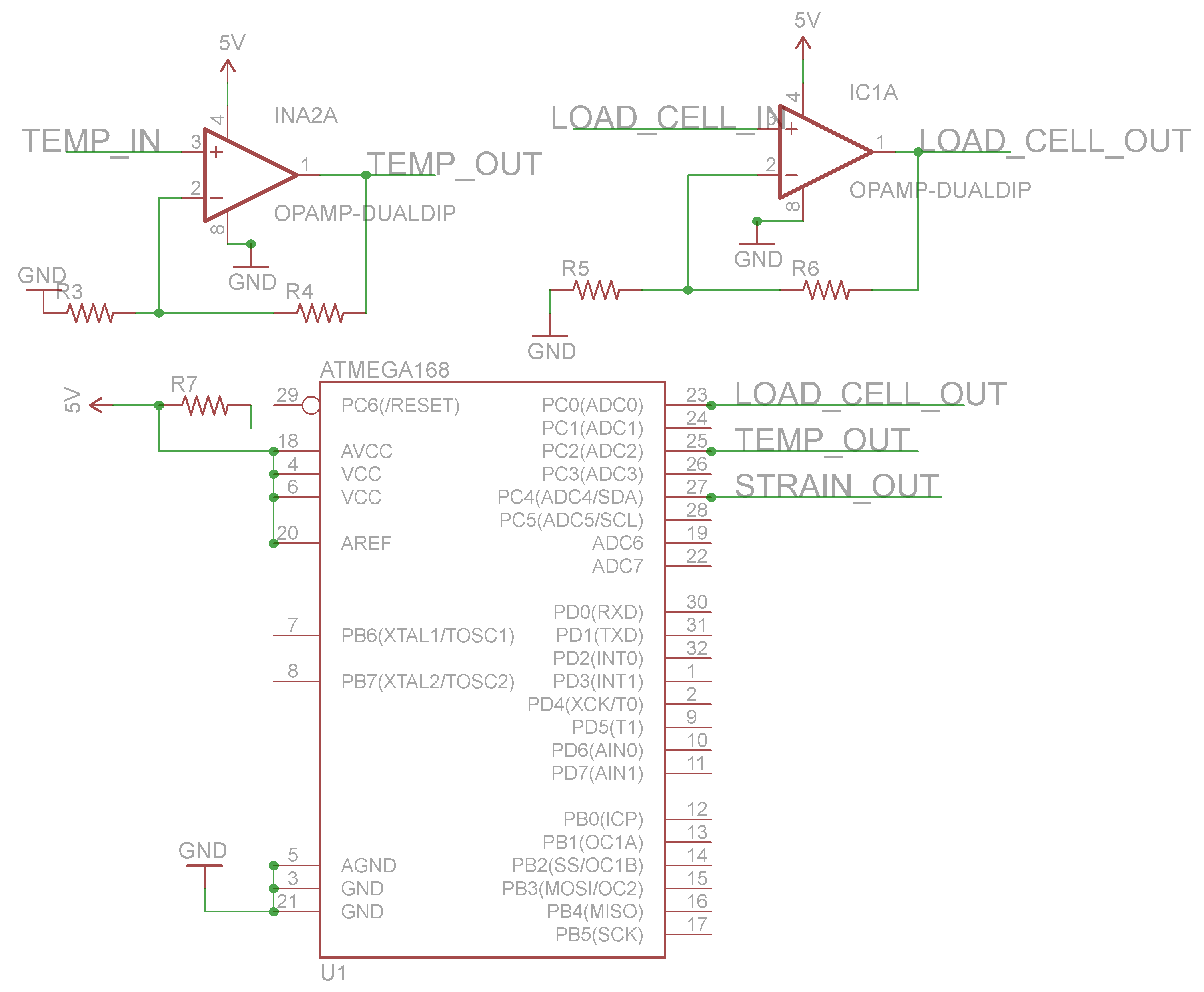
**

Figure 52 - schematic diagrams of Analog devices to Atemga328P

Data logging class

For data logging we will use a micro SD card with the *sdfatlib* library in the background to take care of our fat16/32 file system that will allow us to take the memory card out of our device and load it on any pc without any conversion or being connected to the device. The class diagrams for data logging can be seen in figure 53. The OutputFileStruct structure shown the figure consists of all our output variables as strings. We will overview the design of the data logger class and its methods.



Figure 53 - Class diagrams for data logging

*Class Datalogger*

Void serializeCSV(OutputFileStruct index)

Function: Convert an OutputFileStructs into a comma separated variable formatted string for logging

Void writeLog(String strLogFile)

Function: Runs serializeCSV on current Position of the data array. Then append a log entry to the end of the file located at strLogFile.

**Micro SD Card - Microcontroller schematic**

The schematic in figure 54 below shows the configuration of micro SD adapter interface to the atmega328p, which connects through the Serial Peripheral Interface Bus a 4-wire interface on the microcontroller. The micro SD card is a 3-volt signal device for power, inputs, and output because of this we will use voltage dividers to step down the 5-volt digital signal coming out of the atmega328p when communicating to the micro SD card. The atmega328p also sees 3 volts as a logical input high allowing allow the micro SD card to send information to the atmega328p without voltage regulation.

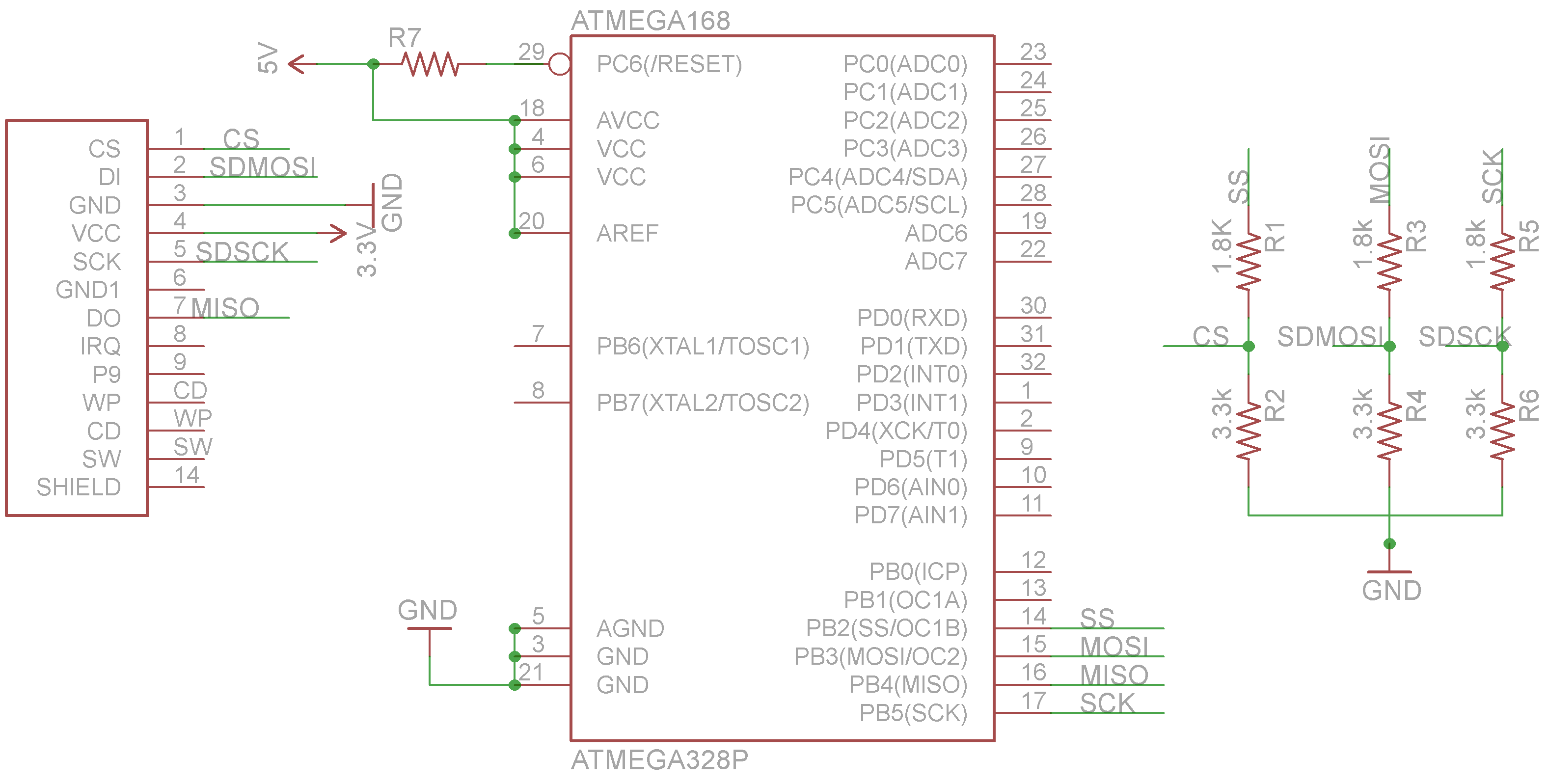


Figure 54 - schematic Micro SD to Atemga328P

**User Interfaces**

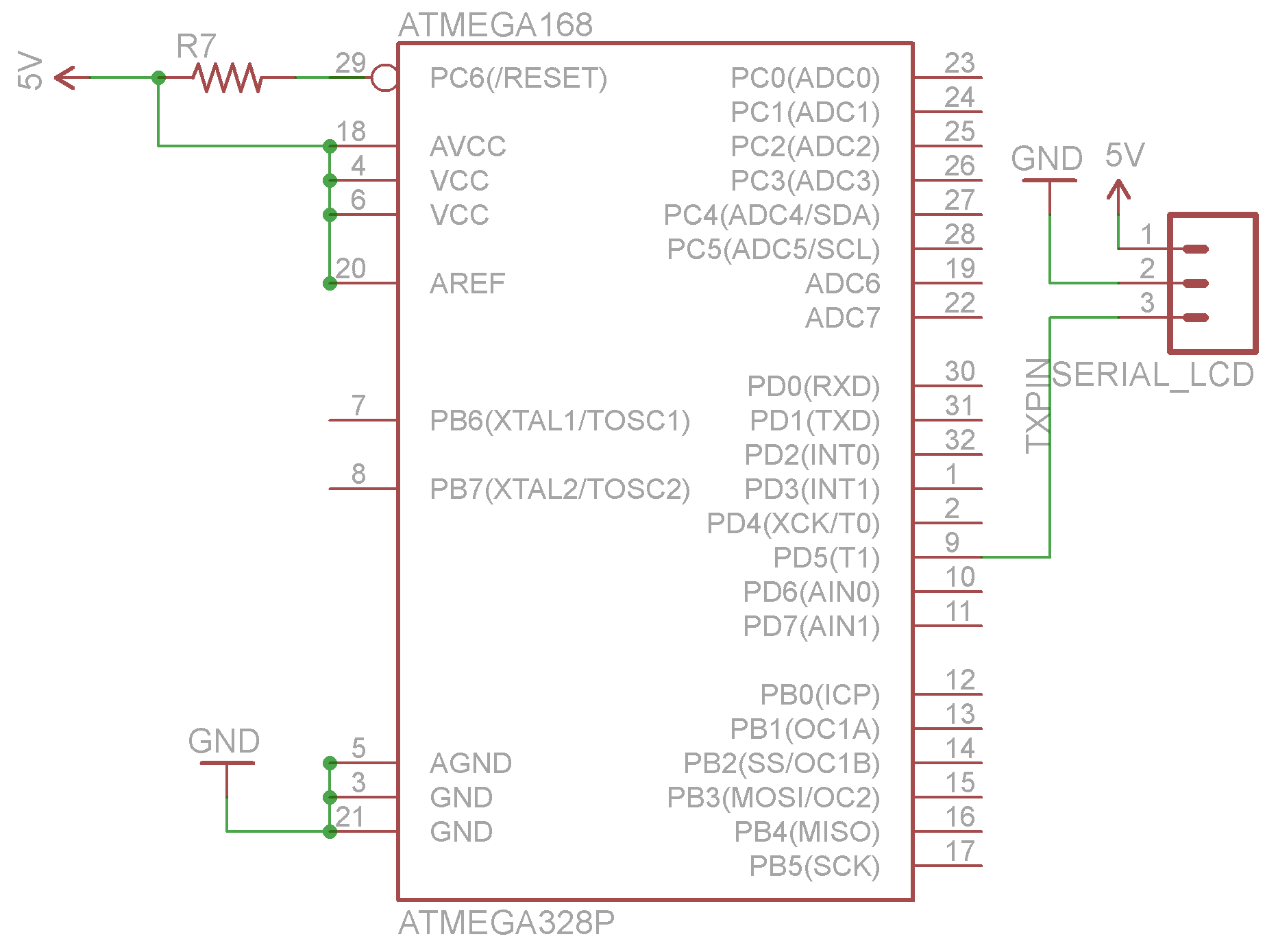
The ATT 4000 will consist of 3 interfaces, a physical, web, an USB to PC interface. A user will interact with the GUI through the USB port of their PC using a USB mini cable. The USB to PC interface will be the only full-featured interface available for the ATT 4000. The physical interface will allow the user to pause and resume an experiment but not start an experiment; a user will also be able to read current status values of the 3 analogs devices, cycles remaining and current cycle position on LCD display. Finally the web interface will only consist of the ability to read current status only. We will now discuss the design, implementation, and schematics of the each interface.

**Physical Interface**

The physical interface must be simple and user friendly, by specification the user must be able to tell if the system is running form 10 feet away, pause, stop, and resume experiment, and view status information from all sensors.

To achieve the first requirement of the status of the system form a distant we will incorporate 2 status LEDS. The first led will be a green status light to let the user know that the device is powered up and went through it basic power up routine without running into any errors. Once an experiment is running the green LED will blink slowly to let the user know that an experiment is running as expected. The second will be a green LED we will incorporate will be an error led which will let the user know when something has go wrong with the system. The LED will blink at different frequencies, for example blink twice every 5 seconds to let the use know the motor was not connected properly, and stay on continuous if the initial power on script does not execute successfully. The user manual will include all status light explanations and combinations. The LED will connect to digital pins 2 and 3 of the atmega328p as seen in the schematic diagram in figure 55.

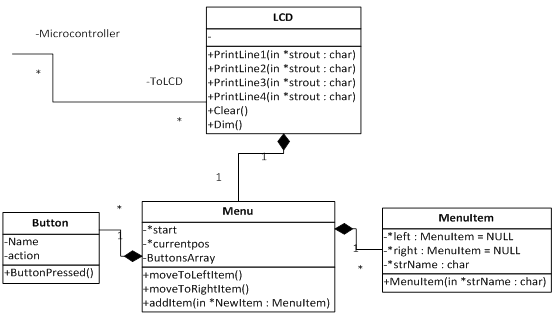
To achieve our second requirement of be able to pause, stop, and resume an experiment will incorporate 33mm push button for each function. When the pause button is pressed the system will finish its current cycle, log all data including current cycle, load, strain, and temperature and also save the current status structure on the micro SD. This will allow the user to press the resume button and have all information loaded, also after resuming the PID resume function will allow the pressure to return to its previous state. Finally the stop button will do the same as the pause but will not save any states. These LED will connect to digital pins 4 and 6 of the atmega328p as seen in the schematic diagram in figure 55.

**Finally we will incorporate a 2 by 16 serial LCD with a 33mm push button interface. This interface will let the user check the current status of all components of the system. The default screen for the A.T.T 4000 will display temperature in degrees Fahrenheit, load in pounds, and strain in unit less displacement. Pushing the up and down button will all the user to cycle through the LCD menu, each menu item will display more information about the experiment running including but not limited to PID constant value, remaining cycles and current cycle desired load. After 10 minutes of non-user interaction the LCD will default to the main screen, which displays basic information. The full schematic for the physical interface is shown in figure 55.

**Figure 55 –** Atmega328P full schematic for interface

**Physical Interface class**

For the physical interface we will implement LCD, Button, Menu and Menu Item classes which interaction can be seen in figure 56 below. Even though we will only use a 2 line LCD the LCD class will incorporate the ability to print to a 4 line LCD for future implementation of our device. We will also allow the device to dim after 15 minutes on none user interaction. The Button class will incorporate a dynamic ButtonPressed() method which perform the desired action such as pause the system, and navigating through the menus. Finally menu class will include an array of MenuItem class variable, which will be initialized using a menuitem list saved in the EPROM ram of the microcontroller; this will allow us to save programming space



**Figure 56 –** LCD interaction

Web Interface

The web interface will be implemented using the Arduino Ethernet shield. The shield comes with a Wiznet W5100 chip that will control all network traffic, TCP/IP stack implementation, obtaining mac addresses and setting the IP address for the A.T.T 4000. This shield will also give use the ability to store or web page on an external SD card and serve html pages off the SD card saving a tremendous amount of programming space on our microcontroller. Because the web interface must be extremely limited the user will only have read only access to the device though the webpage. The webserver will also tail the last 5 lines of the current experiments data output. The user will also be able to download current and pass experiment data stored on the data logging SD card through the web interface using the SdFatLs library.

USB to PC interface

The USB to PC interface will be the only full-featured interface available on the A.T.T 4000. It must be able to work on both Mac and Windows operating system, hide complexity of the system, communicate accurate information to the user, simple and user friendly and plug and play. We will create the GUI using Java Swing, with RxTx library to control hardware side of communication. This will allow our software to be portable between each operating system with.

In order the accomplish USB connectivity to our device we will use a USB to TTL UART Serial Adapter. When the user sends commands to the device from the GUI the commands will be passed through the USB connection of the PC and go to the microcontroller using a USB to TTL UART Serial Adapter. When a user interacts with any component of the GUI that triggers communication with the device, we will call the Serialize() method in the SerialCom class to convert the request into being sent into the correct format for the microcontroller to process, this will include running multiple commands to achieve a desired effect. The deserialize() method will convert information coming from the microcontroller into a user friendly format to be displayed when needed. Each method in the command class incorporates the appropriate procedure for running the desired function for example when command.start() is called our interface will check the input file for errors, calculate run time, set P,I,D constants, etc. Then send the proper produce to have the system begin the experiment. The Serialcom and command class can be seen in the diagram below.



Figure 57- Serial communication Class diagram

Files Classes and Data Structure

For data logging, setup, and running experiments a user must use files. We designed 3 different file class and data structures that will handle each type of processing. We choose a file structure instead of loading all information into ram because this will allow us to save programming space on the microcontroller by destroying the object after creation or read the information for the SD card if needed. All file will be stored on the SD card, the user can create a backup on his or her pc using the GUI. Class diagrams for these interfaces can been seen in figure 58 each class also includes its corresponding data structure.

The Setup file will contain the paths for the input file, output files, status, current position and finishing criteria. Also the user either through the GUI or manually must insert the value for the strain gauge, motor and load cell type. When the user pauses an experiment a backup of the current setup file we is created on the SD card and then the original file will is edited to include the a information dump of the current statusStruct of the experiment to allow the system to properly resume when needed.

The input file class will be used to control each cycle of the experiment. The user must supply a file in CSV format with the following variables for each cycle number, desired load, duration, and desired temperature. The input file can created using excel, notepad or within the GUI. The user will also be able to verify that there input file is correct by using the Checkfile() method of the InputFile class, this method will check each line of the input file for the correct syntax and data type of each cycle. This file also must accessible from the SD card, if the file is altered during a experiment the user must rerun the Checkfile() method before resuming. The data logger class and structure was discussed in the data logging section above.

Figure 61 - class diagrams of files structures



Experiment Flow Algorithms

Each test in the system flow section of our research will be inherited for the MainFlow class. The MainFlow class will include the data logger, motor control and PID, and analog signal class; MainFlow will be an abstract parent class for the following experiment classes, which are WaveTest, ConstantTest, and LinearTest. We will now discuss the design and algorithm of each class needed to run the 3 required by specification.

For the Linear test algorithm the user must specify both start and end cycle number, and the starting load, pressure increment. After this the microcontroller will implement the PID methods to reach the desired starting load then continue adding pressure according to the pressure increment value. The experiment will end if the load cell reaches zero or the program has reached the end cycle number.

The ConstantTest algorithm is the most basic of the 3, the user inputs a desired load and duration, and the microcontroller will us the PID controller to stabilize that load, while the data logger constantly log the data at 50 HZ. If the object does not break before the set duration is reached the experiment will end. The algorithm for the Waveform Test is figure 59 below; this test consists of constantly reading of the input file and writing to the SD card. For this test one should generate the input file using excel to generate the proper cycles, durations, and loads.

*…….*

*Check Input file*

*If not successful End*

*Initialize motor, PID*

*Log all analog device states*

*Create Output File for logging*

*For each cycle:*

*Read cycle from input file*

*Save desired load and duration*

*While duration > 0*

*Calculate error and P, I, D terms for PID algorithm*

*Adjust motor using PID algorithm*

*Log data, update status*

*If (load == desired load) increment duration*

*If test specimen is destroyed END*

*……….*

Figure 59 - pseudo codes for waveform test

**Build / Prototype**

To prototype the power source, we will take a standard discarded cell phone charger and strip it of its covering. The wires will be spliced and then pieced to a bread board, where we will place a resistor and a voltmeter to measure the output of the phone charger. This will be placed with different resistances to see the fluctuation in output voltage and current. This will be the basis for the power source.

The load cell circuits will be built onto a regular bread board for the initial prototyping. We will first build the 7812 regulator circuit and test it with a test load and the voltage across the load with a volt meter. The regulator circuit will include the 7812 regulator and two polarized capacitors. The capacitors are to limit and stop the voltage fluctuations and turn the output of the regulator into a complete DC voltage. The output of the regulator circuit goes into the load cell’s positive and negative excitation terminals to power. Once the load cell is powered, we will insert the signal leads into the voltmeter to test the output of the load cell. When there is no load on the cell, the output should be approximately zero. Then we will increase the load as done on the cell-specific sheet, and monitor the output as the load increases and figure if it follows the linear fit model in the test. If this is correct, then the load cell and regulator circuits work properly.

To prototype the strain gauge circuit we will use the 7812 regulator circuit as previously installed with the load cell. The set up will be the same with the 7812 regulator and the two polarizing resistors. The prototype strain gauge circuit will be set up in the following manner. The two resistors on the opposing side will be the same value and somewhere between 100 Ω and 1000 Ω. The resistor in series with a test strain gauge will be selected to match the initial resistance of the prototype strain gauge with popular values being 120 Ω, 350 Ω, and 1000 Ω. The prototype strain gauge circuit will output to a voltmeter that is connected at the two middle junctions on each side of the bridge. The voltmeter will measure the voltage across the two parts of the bridge. At equilibrium, when the strain gauge is under no stress, the voltmeter should output 0. When that is achieved the strain gauge will be correct.

PCB

We use eagle cad to layout or complete circuit for final production. Sparkfun and many other companies provide eagle libraries for most of their parts and for any costume part eagle cad has a pretty easy to use interface which we can use be used to build and configure parts needed.

We want to be able to change our part just in case of a malfunction or blow chip so the follow guide line will follow. All essential components will be able to be taken off the PCB and replaced if needed. This means all non-mechanical components such the LCD will use a mole connectors or DSUB connectors. DIP sockets will also be used for any major integrated circuit chips; giving us the ability to remove and reprogram the microcontroller if needed.

By specification all mechanical such parts such as the load cell, strain gauge, actuator and thermocouple must be able to be interchangeable. For this we will place multiple screw terminals on our PCB that will extend out of our project box, with label to let the user know what goes where. This is a common design we have seen in the mechanical engineering department’s research labs.

Once we lay out all our schematic design with interconnections we will use eagle cad’s schematic to PCB editor to place all our physical part on the PCB layout. Then use the auto route to route all our wires which takes anywhere from 30 seconds to 2 hours.

We will the purchase and have the PCB fabricated from pcbexpress.com which will cost $121 plus shipping and handle for 2 2-Layer +Silk-screen + Solder Mask.

Software Build/Prototype

GUI

We will use the eclipse IDE for programming and prototyping our GUI. Eclipse is open-source and come with a number of debugging tool and complete project management interfaces. Each component of the GUI will be prototyped and built separate keeping the design modular. Finally once we have all components working correctly separate we will then slowly integrate each component into the main GUI interface.

Serial Communication

For prototyping the serial communication interface we will use a DSUB 9 and a commercial USB to serial adapter, this will allow us to test all output from the microcontroller through the Arduino serial monitor or minicom from Linux. Once we completed the build and test of the FTDI RS232 interface we will switch to using an USB cable for further prototyping our device. Finally once we are able to communicate with the serial interface with our GUI we will switch to using it for till production.

Atmega328p programming

For writing code for the atmega328p we will use the Arduino language and IDE, which is a language similar to C/C++, based programming. Like the GUI, we will prototype each system separately and then slowly integrate them into the main program using header file. For getting code from the computer to the microcontroller we will use the 6 pins ICSP header w/ the SPI interface, an USBtinyISP programmer and avrdude, which is free utility, used for programming microcontrollers.

Micro SD

For prototyping micro SD card communication to the microcontroller we will use the SD adapter that comes with most micro SD card and can be found at Wal-Mart. We will then take apart and strip the wires from an old IDE cable. Finally we will solder leads to each pin on the micro SD adapter, and then use a breadboard to connect the microcontroller to the SD adapter interface. For production we will purchase and use an SD card breakout board from sparkfun.

Input and Output Signals Software

We will first interface a potentiometer with our microcontroller to properly calculate analog to digital conversion from a known source. Then after each electrical circuit has been prototyped on a breadboard, we will then proceed to testing analog to digital conversion using the microcontroller with each component separately and a voltage meter for correct comparison. Finally we will test each component and the conversion equation for voltage to their actual value for example voltage to pounds, or voltage to degrees.

For all outputs we will use a voltage meter to measure that we are sending the correct output for our motor.

**Testing**

The mechanical engineering department of the University of Central Florida will use the A.T.T 4000 test specimens from large array of companies. Experiments include putting the specimens under extreme pressure and data log for a significant period of time. We expect our testing efforts to give us a good indication of failing points, software malfunctions, and data logging speed, and actuator speeds and functionality. For our sponsor we expect our testing efforts to help us fine tone our system to their liking and verify each requirement has been filled.

GUI Event Table

Table 11 consists of how event will be handled by our GUI. How external Stimuli for the user running the experiment will affect internal data and states. Also what response will the user see when the active an event. Only a few cases are touched in the table but the give a good overview of all the possible causes while working with the Automatic Tensile Tester 4000. The user manual will contain a appendix dedicated to these type of events so the user can quick troubleshoot error if they may arise. All users should read this table to get the best understanding of how they should interact with the GUI and what goes on in the back end

|  |  |  |  |
| --- | --- | --- | --- |
| Event Name | External Stimuli | External Responses | Internal data and state |
| Application Opened | User opens application | Application GUI displays with main window. If experiment is currently running user directed to experiment status window. | System runs initialization script. System in Active state ready for user response |
| Initialize New Experiment | User selects option to start new experiment | User is directed to experiment configuration page, with require textbox inputs for configuration. | System is in stand by awaiting proper configuration data include experiment type, duration and other parameters. |
| Load / Confirm Experiment settings | User either submits initial experiment information or load old experiment configuration file | User is directed to main experiment status window /w Current status information available. | Query current experiment information. Goes back to Active state |
| Verify Input File | Use selects file verification or user try to start experiment | Application verification status bar appears | System checks each line of input file. Back to active state if no error, prompt user if error. |
| Run experiment | User select play button | Application experiment status bar appears. User prompt for error or to start experiment | System initialize and a test all components of the system. Including creation of all necessary files. Go back to Active state |
| Pause experiment | User select play button | User is directed main experiment status window. | System backups all current statuses and marks cycle position. Save all information to SD card. |
| Stop experiment | User select stop button or experiment is completed. | User is prompted with completion status then directed main experiment status window. | Verify all information is logged, save final status information. Return to active state |
| Application Closed | User closes application | Application prompted user to assure user want to exit | System is in Inactive state |

Table 11 - GUI Event Table

Test Environment

For our GUI we will run initial test in a Linux or Mac operating system, or microcontroller test environment will be an Arduino development board or a breadboard. For Many test we will also coordinate with the mechanical engineering lab. Developers and lab students and other user of the system will do test. The test environment will may not be considered as the actual environment our device will run in because of this we will simulated then predict outcomes of the device under a different environment. The main difference between testing and production environment is that we will keep the temperatures at room temperatures and also because or device is made to be portable we cannot predict every location our device will end up.

Stopping Criteria

For microcontroller software development we will use a waterfall method which call for use to continuously test while we development and also take a look back at or specification and update them as needed. If we the developers find no fatal errors during testing, we will the deliver the prototype to our sponsor for more testing and verification of specification completed.

The GUI we develop using extreme programming mixed with scrum method were testing development and planning will be tested for every widget we create. The reason for using XP and scrum put us on a plan, develop, test schedule each week for a different widget; completing 1 to 2 widget a week. We will stop testing at the end of each scrum period.

We will define the system as good enough to deliver, if the entire system hardware and software passes at least 2 hours experiment with planned interruption to test requirements periodically and 2 hours without user interaction. We believe if the system and run 2 hours it should be able to run 1 year with no problem.

Individual Test Cases

Test Objective: Wall Wart Output Regulation

Test Description: We want to make sure that the wall wart it outputting the correct voltage and current. The wires will be stripped and placed in a test bread board where the output voltage and current will be measured. After that, we will place increasingly large loads on the ends of the wall wart.

Test Conditions: The test will be performed at room temperature in a laboratory environment

Expected Results: The output voltage will stay constant until the current threshold is breached by a very small load.

Test Objective: 7805 Regulator Circuit Output Accuracy

Test Description: We want to make sure that the 7805 regulator circuit it outputting the correct voltage and not coming close to its maximum current. The stripped wires from the wall wart will be placed in a test bread board where the output voltage will be the inputs to the 7805 circuit. We will measure the initial output of the regulator. After that, we will place increasingly large loads on the ends of the regulator.

Test Conditions: The test will be performed at room temperature in a laboratory environment

Expected Results: The regulator circuit should output a constant voltage of 5 volts as long as the output current does not approach 1 Ampere

Test Objective: 7812 Regulator Circuit Output Accuracy

Test Description: We want to make sure that the 7805 regulator circuit it outputting the correct voltage and not coming close to its maximum current. The stripped wires from the wall wart will be placed in a test bread board where the output voltage will be the inputs to the 7805 circuit. We will measure the initial output of the regulator. After that, we will place increasingly large loads on the ends of the regulator.

Test Conditions: The test will be performed at room temperature in a laboratory environment

Expected Results: The regulator circuit should output a constant voltage of 12 volts as long as the output current does not approach 1 Ampere

Test Objective: Load Cell load accuracy

Test Description: This test is to determine the accuracy of the output voltages of the load cell and match them with the outputs displayed on the packaging of the load cell. This is also to test the wiring of the input voltage. We plan to use known weights of three to four known objects to measure the accuracy of the load cell to our specifications. One of the objects will be a text book placed on top of the load cell. The next object will be a stack of books placed on top of the cell. Next, one of the experimenters with a known weight will sit on top of a book to measure the maximum manual controlled load that can be placed on the cell. Finally, we will use a machine to apply a force of at least 1000 pounds on the load cell.

Test Conditions: This will be tested in stable laboratory conditions. Therefore this is a dry environment and room temperature test.

Expected Results: We expect our output voltage to be in the range from 0 to 24mV, showing reasonably quick response time on the voltage fluctuations.

Test Objective: Strain Gauge Accuracy

Test Description: Compare strain values with known values for the strain on an object. We will use a strain gauge with known values and compare our strain gauge circuit implementation. The output will be input into an equation derived

Test Conditions: The test will be conducted in a stable lab environment that is temperature and moisture controlled. The test will be on an actual specimen undergoing test conditions. There will be another strain gauge on the specimen, close to the area where our strain gauge will be located in case of poor spatial strain density.

Expected Results: We expect the voltage output of the circuit to be input back into the derived equation and get similar strain numbers to the ones recorded on the strain gauge

Test Objective: Communication with our microcontroller on a computer using the USB to serial interface.

Test Description: Send and receive the flowing set of characters [“ABCDEFGH”] to and from the atemga328p to if we can communication with the microcontroller first through a command line interface; then using our java serial class and finally test our GUI widget for serial communication that the user will interact with.

Test Conditions: Linux for command line test. Windows and Apple for GUI

Expected Results: Print to screen the following characters “ABCDEFGH”

Test Objective: Read the analog voltage outputs of each of our analog devices coming into our microcontroller using the analog to digital conversion.

Test Description: For each device we will hookup a voltmeter to the physical output pin. Then we will user the serial interface to read the calculated value from the microcontrollers corresponding ADC pin.

Test Conditions: Linux for command line test. Windows and Apple for GUI

Expected Results: The microcontroller should output the same voltages as the voltmeter plus or minus .5 volts

Test Objective: Measure temperature from thermocouple using microcontroller.

Test Description: We will repeat each thermocouple individual test with the thermocouple circuit connected to the microcontroller, and then use the LCD to output the converted voltage to temperature value in Fahrenheit.

Test Conditions: These test will be ran under room temperature, outside temperature, and by placing a finger on the thermocouple to measure body temperature.

Expected Results: 75 degrees room temperature, 98.6 degrees body temperature, and current temperature outside at the time

Test Objective: Measure load from load cell using microcontroller.

Test Description: We will repeat each load cell individual test with the load cell and its amplifying circuit connected to the microcontroller. Using the serial communication interface to read output

Test Conditions: Test ran using 1lb to 100lb blocks.

Expected Results: Correct load output reading to screen

Test Objective: Control motor using microcontroller.

Test Description: Using a breadboard circuit to hookup the motor/actuator, H-bridge and microcontroller. Then send an ‘f’ to move actuator forward ‘b’ to move it back.

Test Conditions: See Test Environment

Expected Results: actuator move either forward or back according to signal sent.

Test Objective: read and write data to a SD card using a microcontroller. This will simulate the all method needed for data logging to be achieved

Test Description: setup the microcontroller to run the following command when it receive either a ‘c’, ‘a’, ‘r’, or ‘d’ for the follow test create, append, read, and delete a multi-line test file

Test Conditions: See Test Environment

Expected Results: user should be prompted with an option to overwrite if file exist on create, append should write a line to the end of file, read should print file line by line to screen, delete should destroy the correct file without formatting of trashing the SD card.

Test Objective: Successfully control system with PID algorithm using actuator with load cell as feedback

Test Description: We place a 50-pound block on the load cell, and then insert the desired loads for +5 to +50 pounds into the GUI individually while watching the output for the correct stabilization.

Test Conditions: See Test Environment

Expected Results: We expect outputs from 55 to 100 pounds on the load cell, with a + or - %5 of accuracy

Test Objective: Maximum Clock Speed of microcontroller for data logging speed

Test Description: We will continuously adjust the clock speed of the microcontroller and data logging rate to find the optimal rate of data logging. Moving from standard 8 Mhz – 20Mhz clock rate with 1Hz/sec to 20 MHz/sec until the system fails.

Test Conditions: On microcontroller

Expected Results: Once the max rate is reach data should no longer be log correctly; we expect a rate between 10 and 15 MHz per sec

Test Objective: Complete system working for 2 hours consistently without being interrupted

Test Description: We will run all 3 specified test constant load, linear load, and variable load for 2 hours each, using data input files

Test Conditions: On completed mechanic system with everything hooked up. No computer connection

Expected Results: We expect the test to complete with no errors and proper data logging

Test Objective: Complete system working for 2 hours consistently without being interrupted. This well test unexpected power down modes, pause and resuming, connecting to and from pc while testing.

Test Description: We will run all 3 specified test constant load, linear load, and variable load for 2 hours each, using data input files. Periodically pausing and resuming the experiment from the computer and control box

Test Conditions: On completed mechanic system with everything hooked up. With computer connection

Expected Results: We expect the test to complete with no errors and proper data logging

Test Objective: Accuracy measure run temperature without microcontroller

Test Description: We will connect the thermocouple to the system in the mechanical engineering lab and see if we get the correct temperature reading

Test Conditions: Extreme Heat

Expected Results: We expect reading with values + - .05 volts

Test Objective: Accuracy calculate temperature without microcontroller

Test Description: repeat experiment above will calculating exact temperature readings in degrees Celsius and Fahrenheit

Test Conditions: Extreme Heat

Expected Results: We expect reading with values + - .5 degree

**REFERENCES**

Trent Jarvi, RXTX <http://rxtx.qbang.org/wiki/index.php/Main_Page>

Mikal Hart, “A New Software Serial Library for Arduino” <http://arduiniana.org/libraries/newsoftserial/>

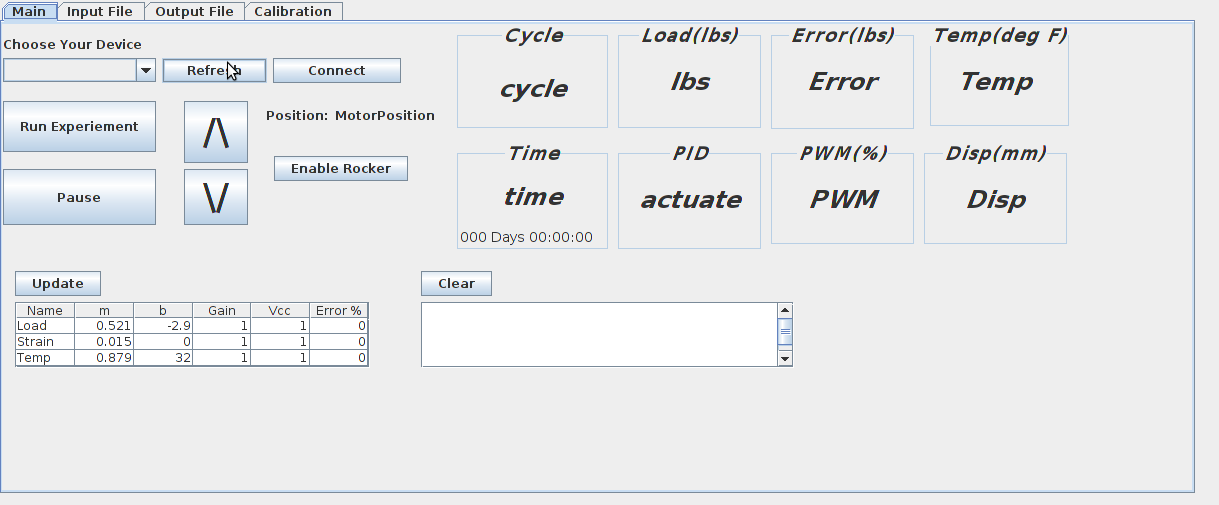
RepRap.org, Thermocouple Sensor 1.0 <http://reprap.org/wiki/Thermocouple_Sensor_1.0>

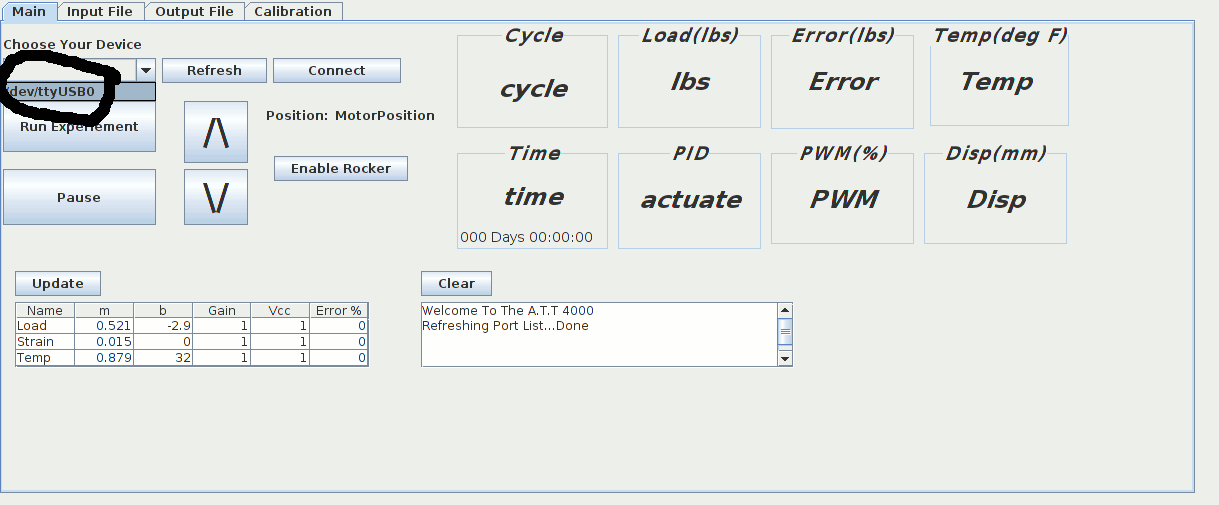
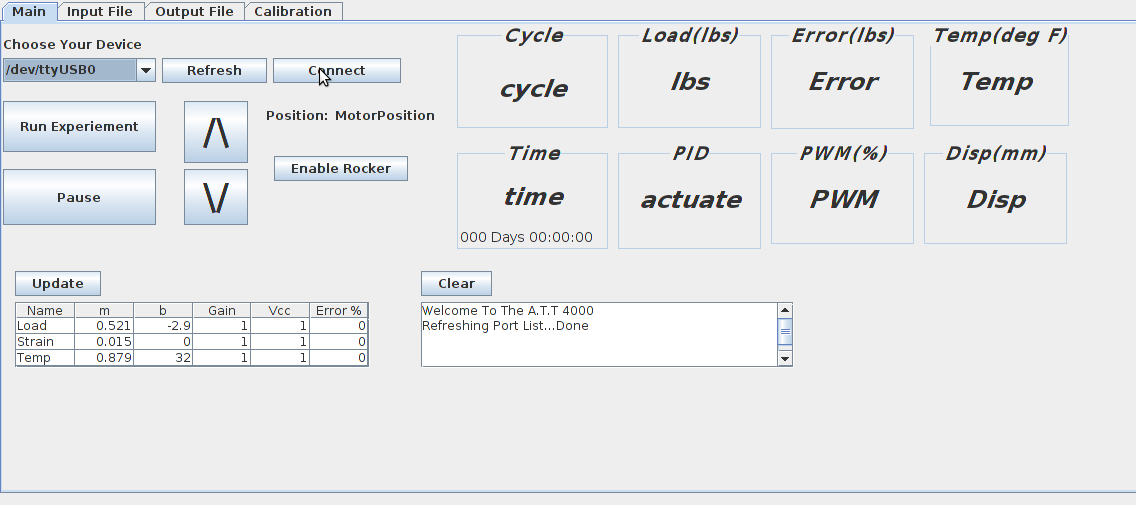
Society of Robots, PROGRAMMING - PID CONTROL <http://www.societyofrobots.com/programming_PID.shtml>

J. Sluka, A PID Controller for Lego Mindstorms Robots <http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html>

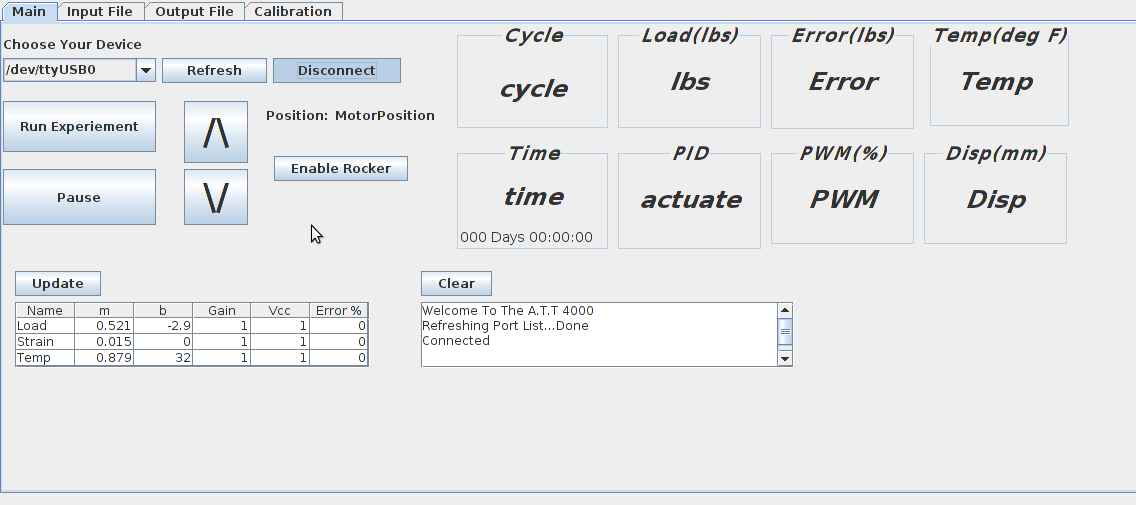
**Connecting to device:**

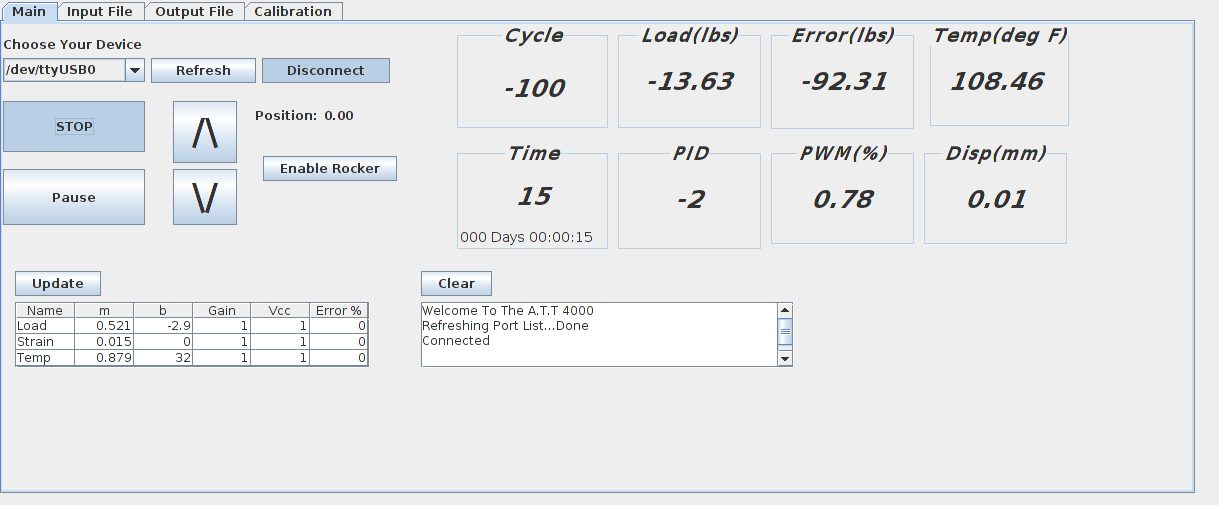
1. Hit refresh to get a list of all available devices connected to your computer

2. Select device you would like to connect to.

3. Click Connect

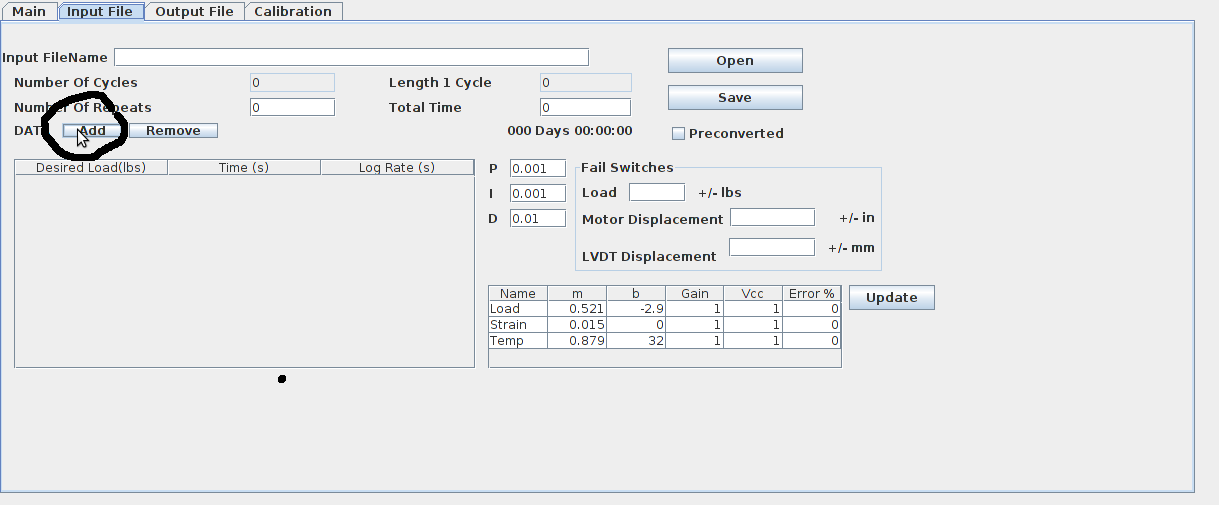
4. If an experiment is currently running on the device information will be streamed upon connecting to the device. If no experiment is running the user may use the up and down arrows to set the actuators position, enable the rocker switch to do the same.

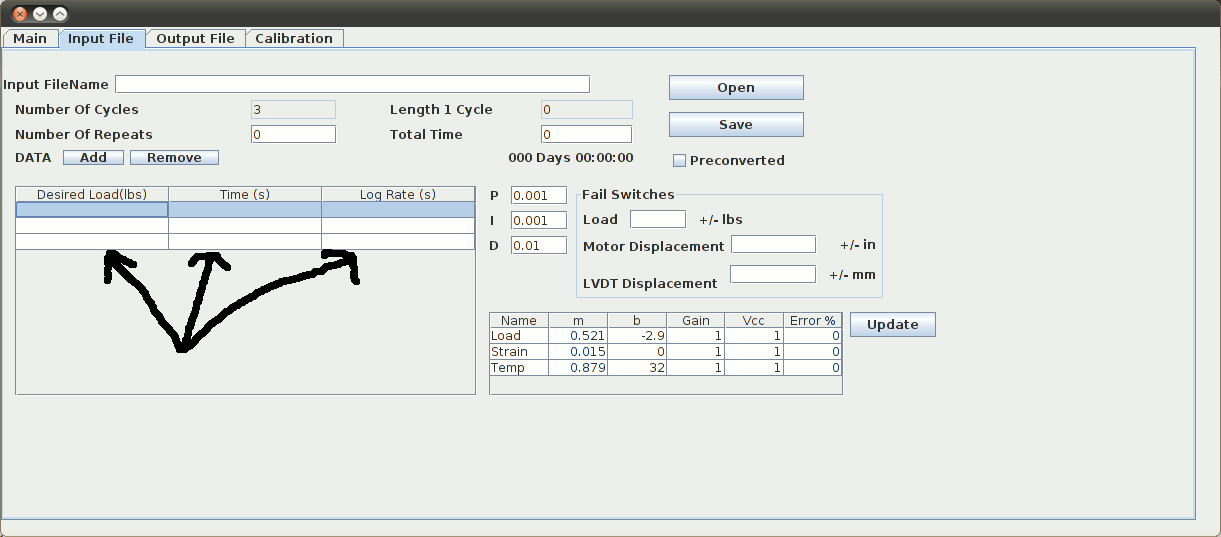


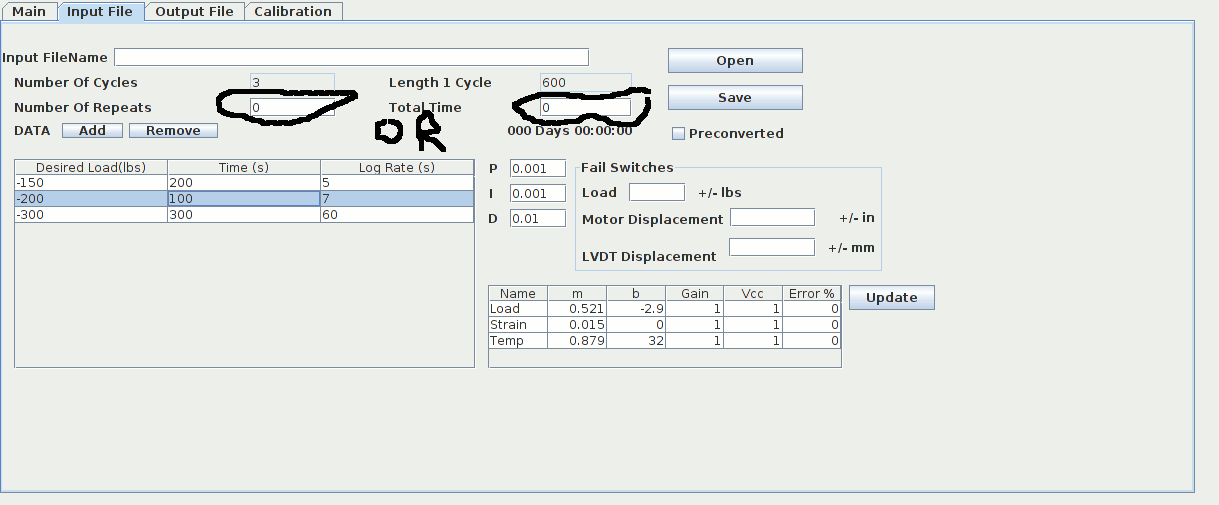


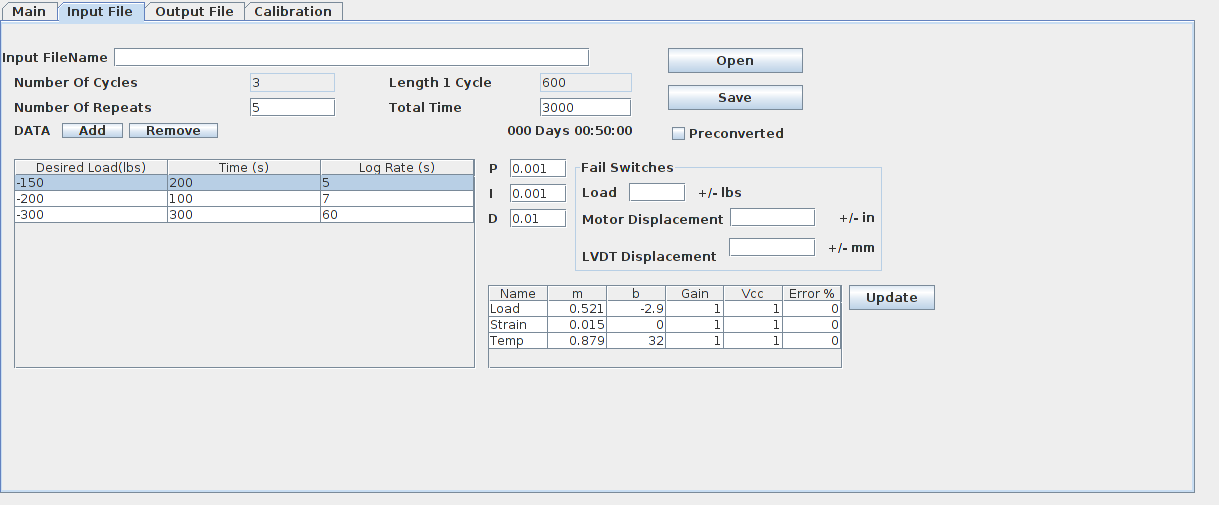
**Creating an Input File**

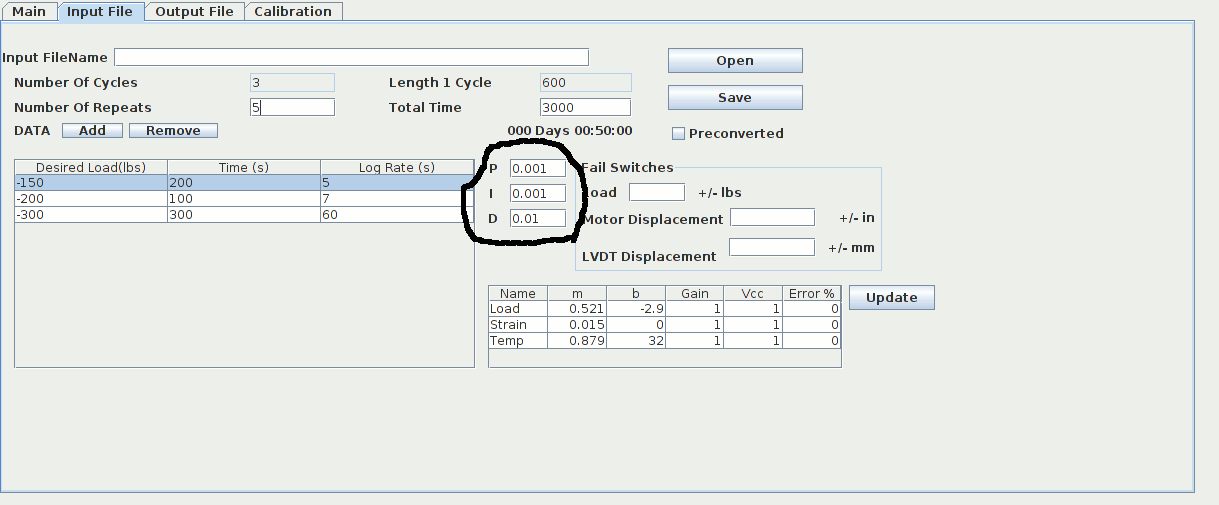
1. Add blank field to you data table

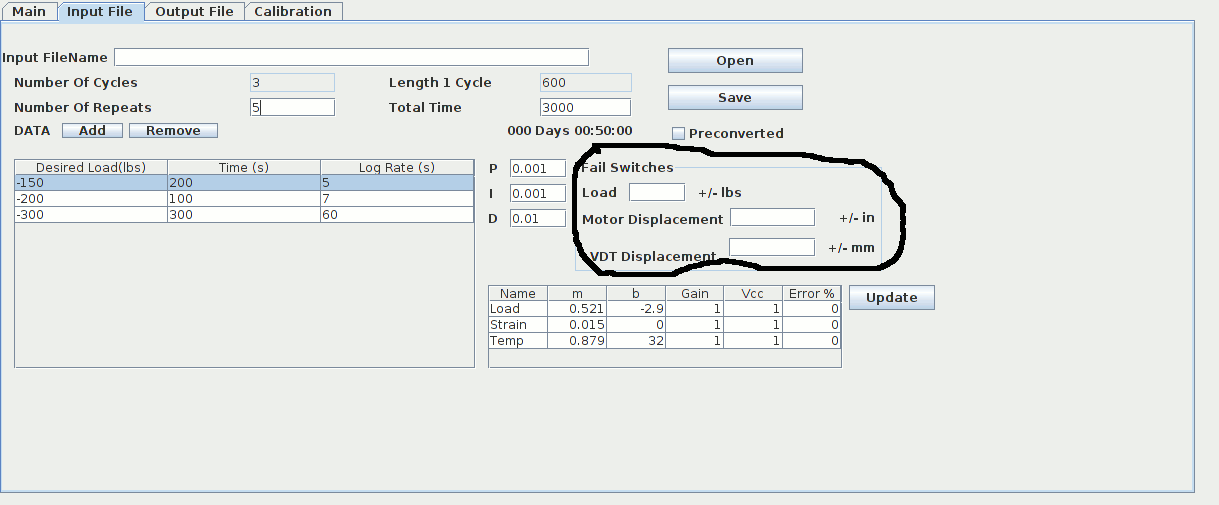
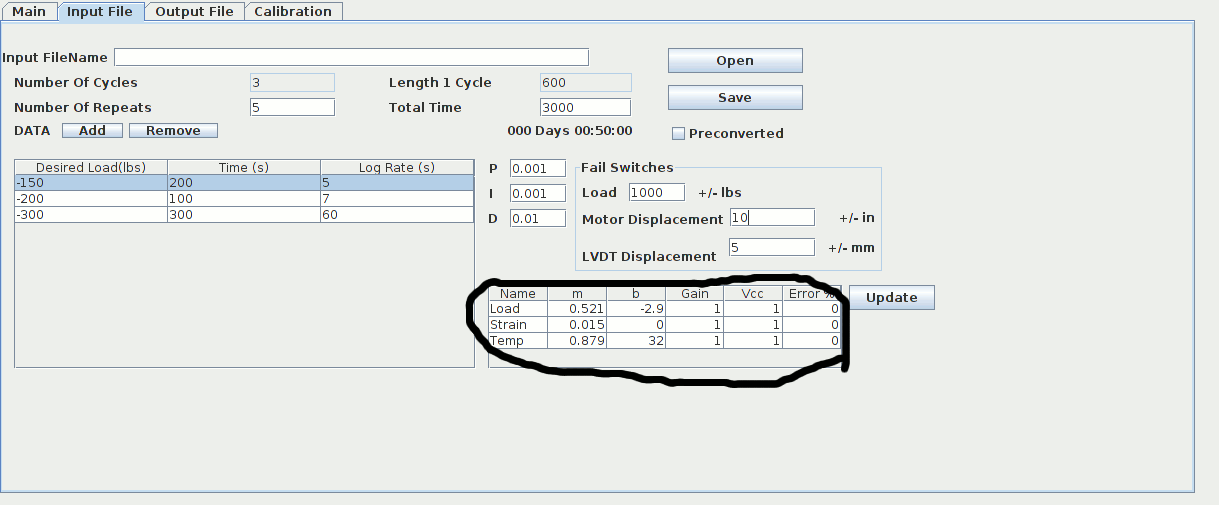
2. Fill in the desired load for each cycle. The time the cycle will run and the desired logging rate in seconds.

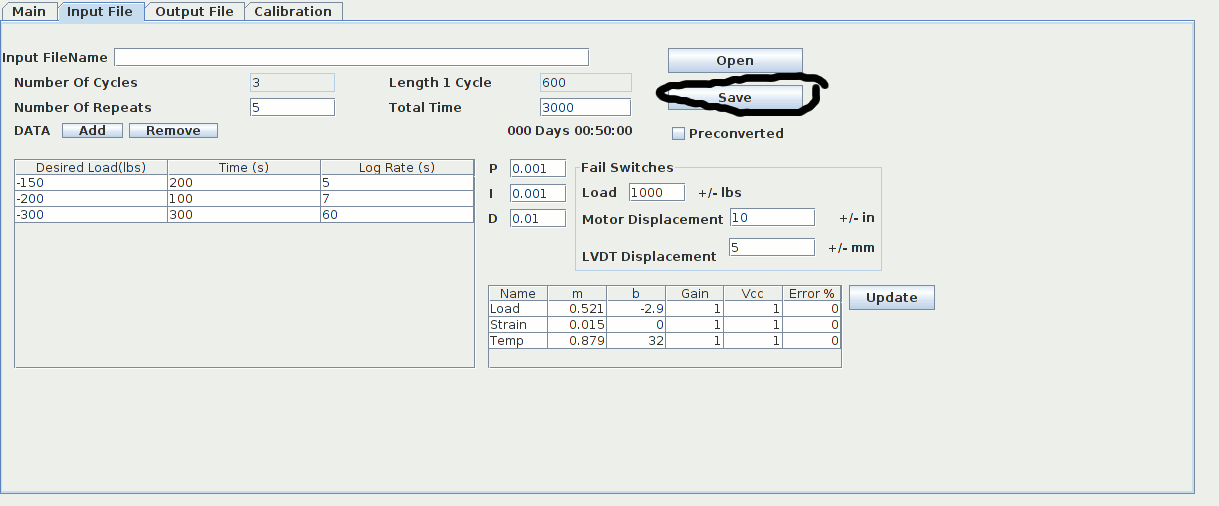
3. Enter the amount of time you would like to repeat the cycle set, or the total time you would like to run the experiment. These fields are dynamically connected.



4. Set PID values

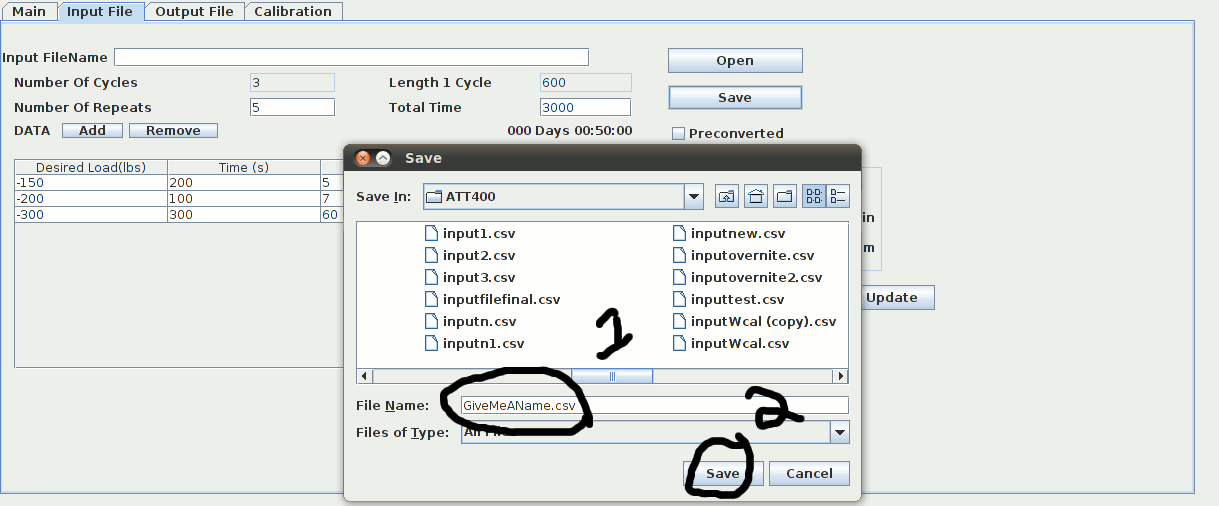
5. Set Fail Switches these switches will stop the experiment if reached

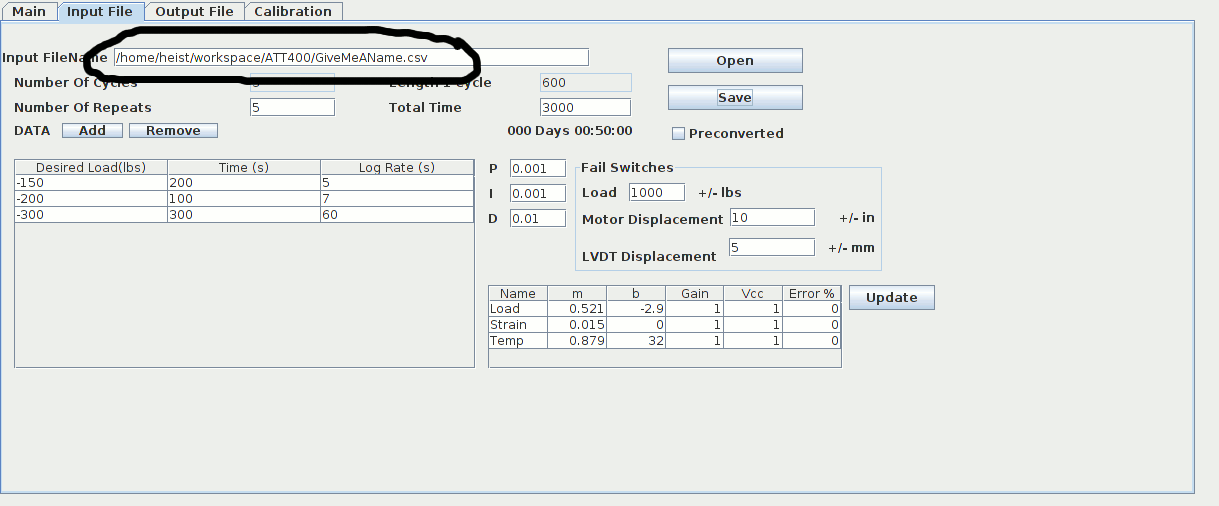
6. Set conversion factor values

7. Check all values and hit save

8. Name the input file. Then click save to save the input file in the desired location,

***Important the file must be called input.csv once transferred to the sd card***

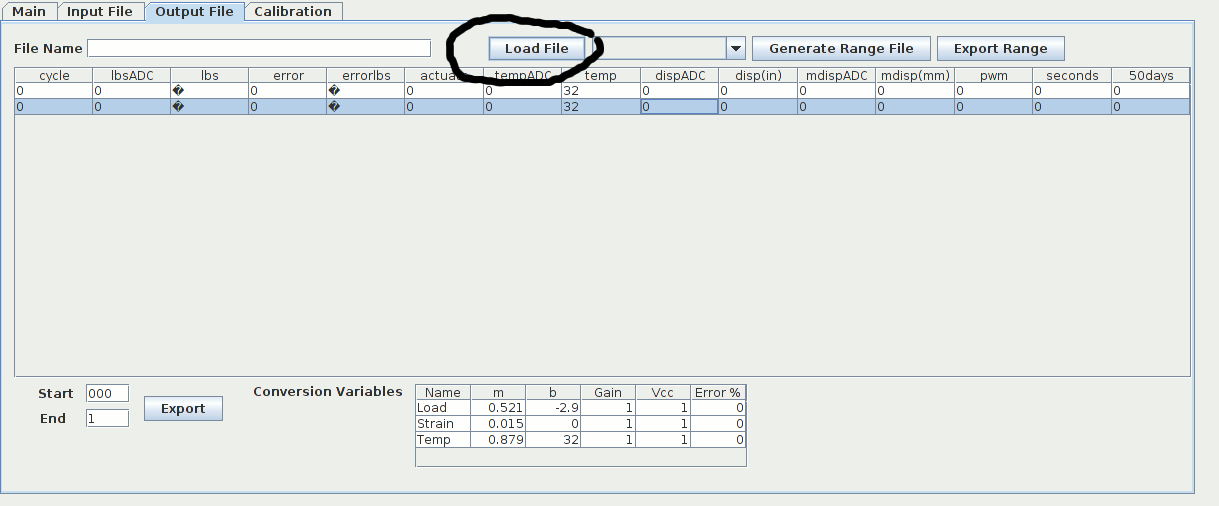
9. Check file name once saved



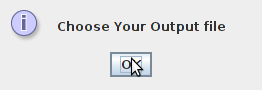
**Output File Interface**

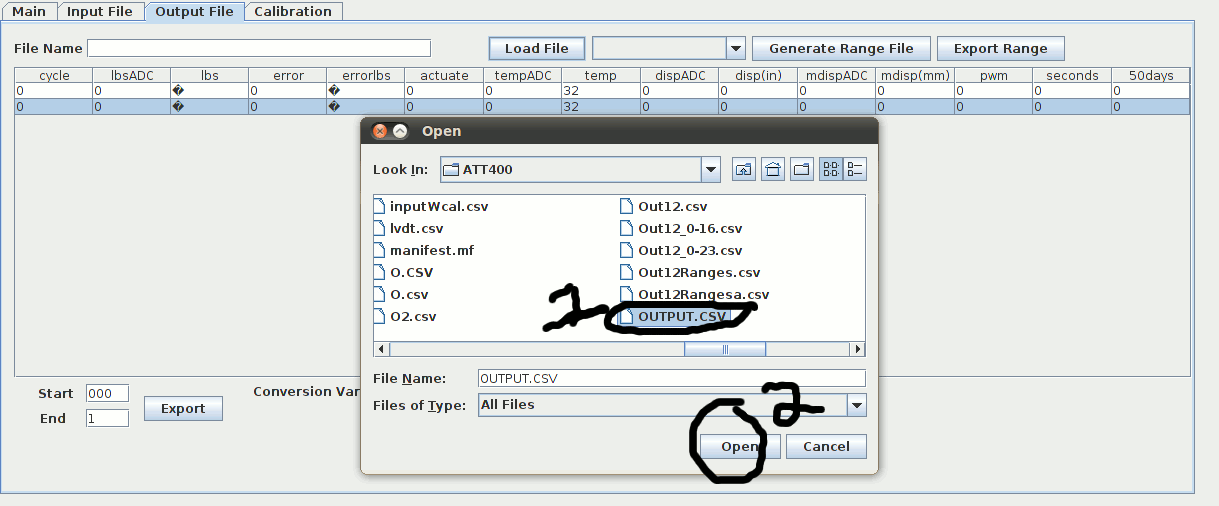
*A.* *Viewing File*

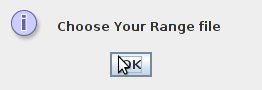
1. Click output field tab then click load File

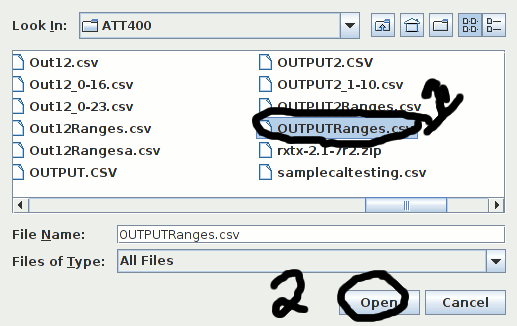


2. Click Ok

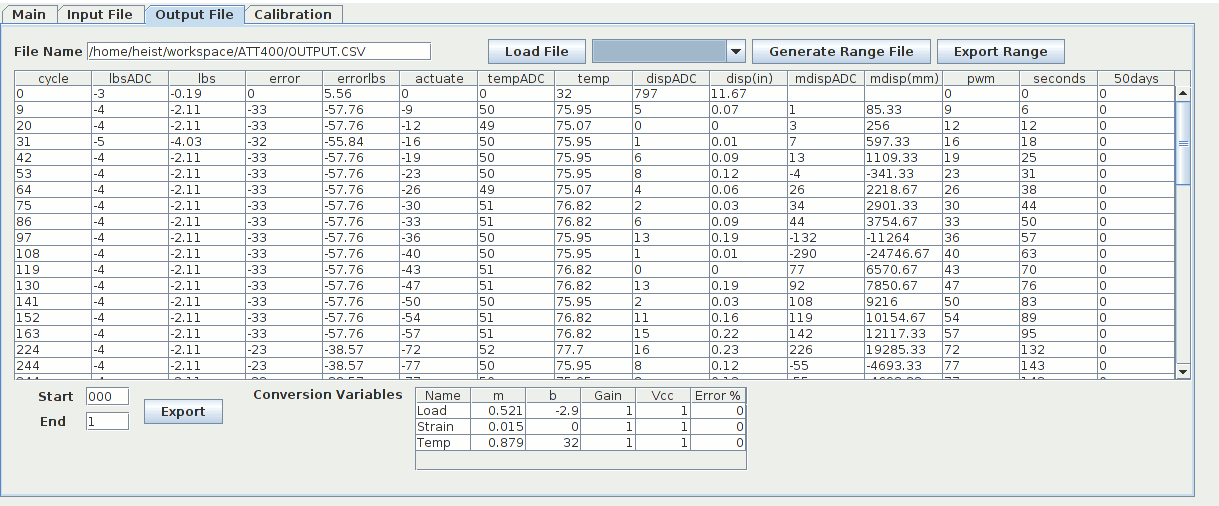
3. Select the file you would like to load. Then click open

4. Click ok



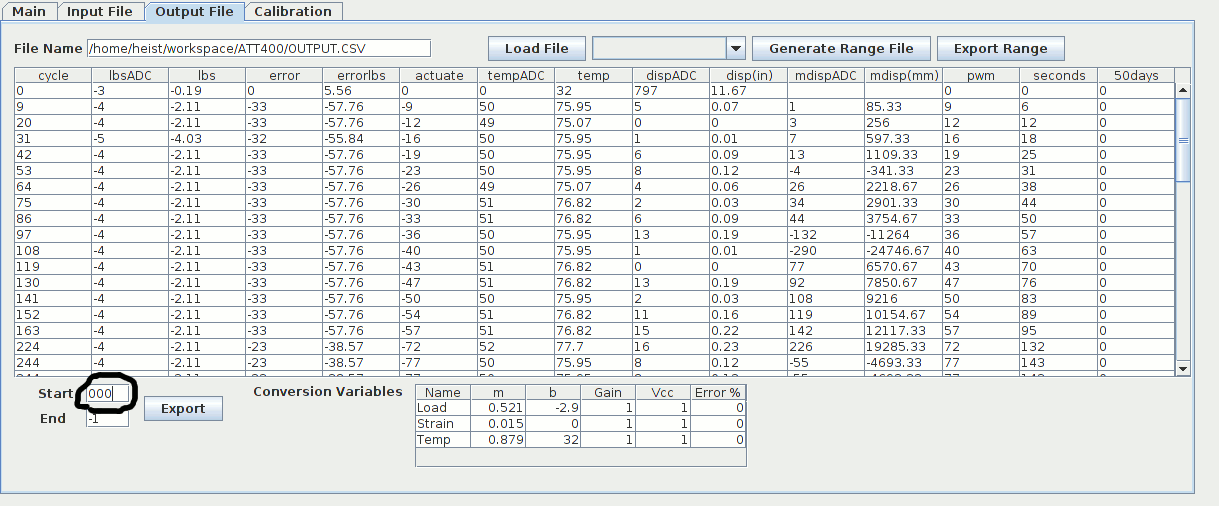
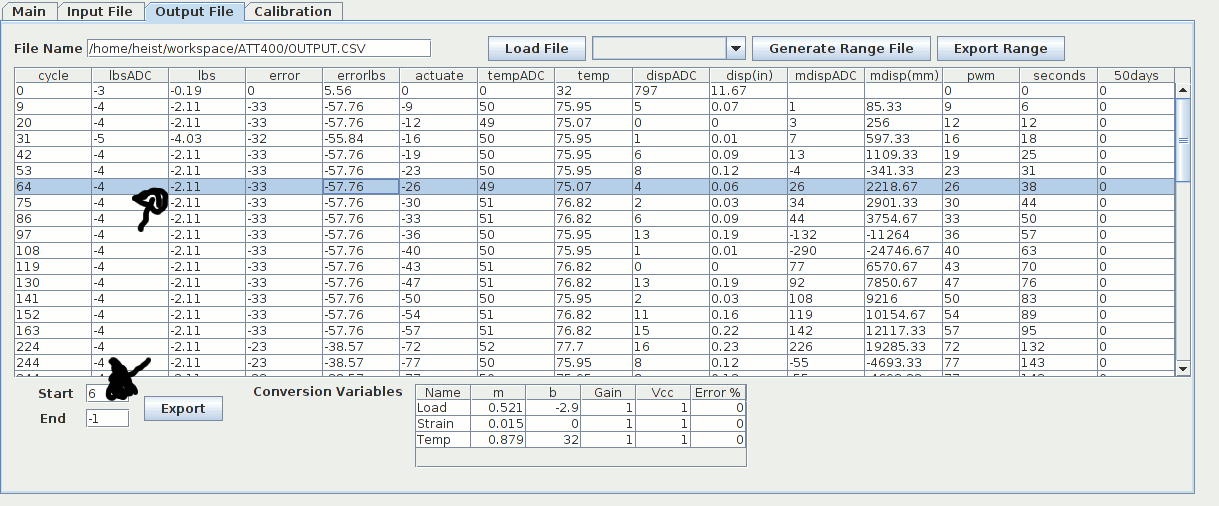
5. Locate and select the appropriate range file then click ok. If you do not have a range file click cancel.

6. File loaded

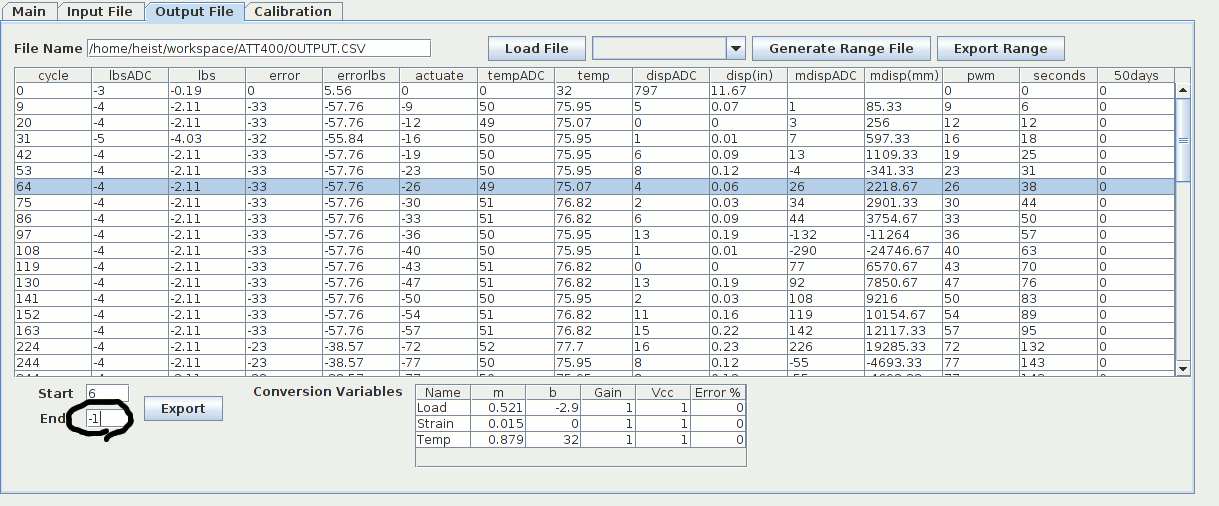


*B. Exporting*

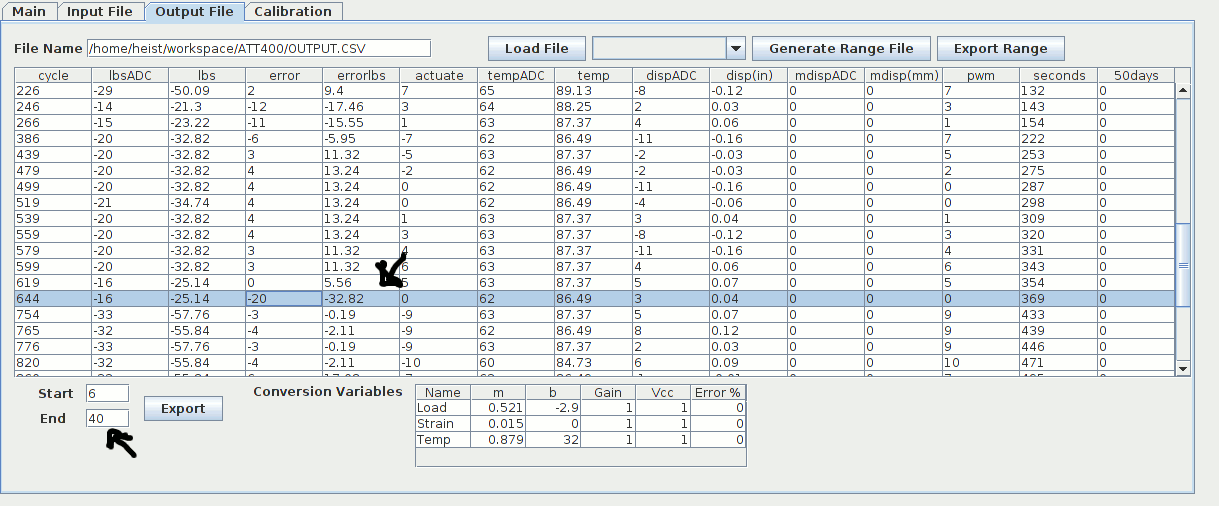
1. Click on the text field next to **start**

2. Click desired starting data field. The start text field should update accordingly

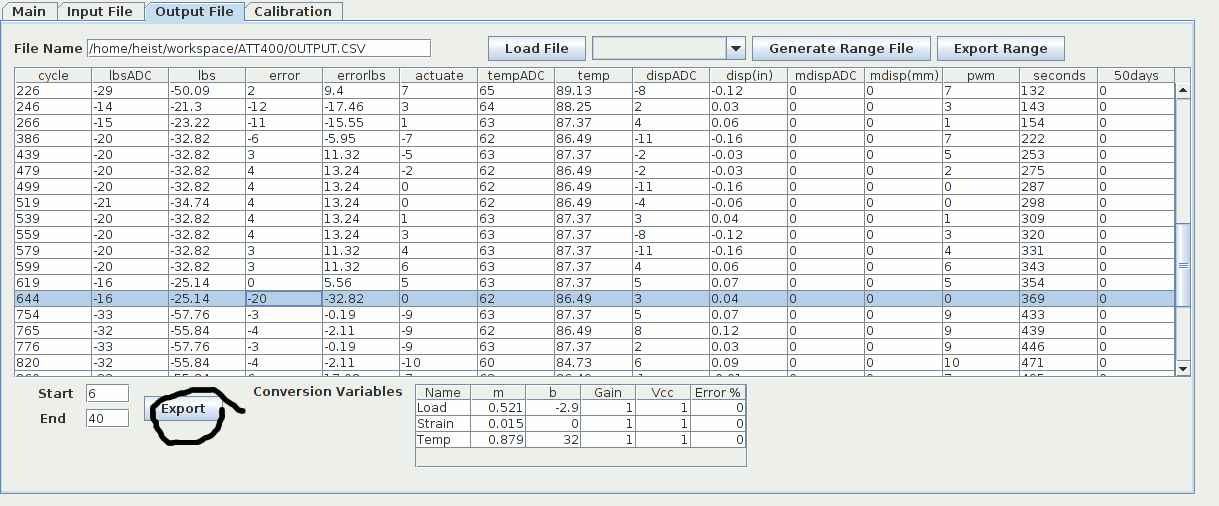
3. Click on the text field next to **end**



4. Click desired ending data field. The end text field should update accordingly.

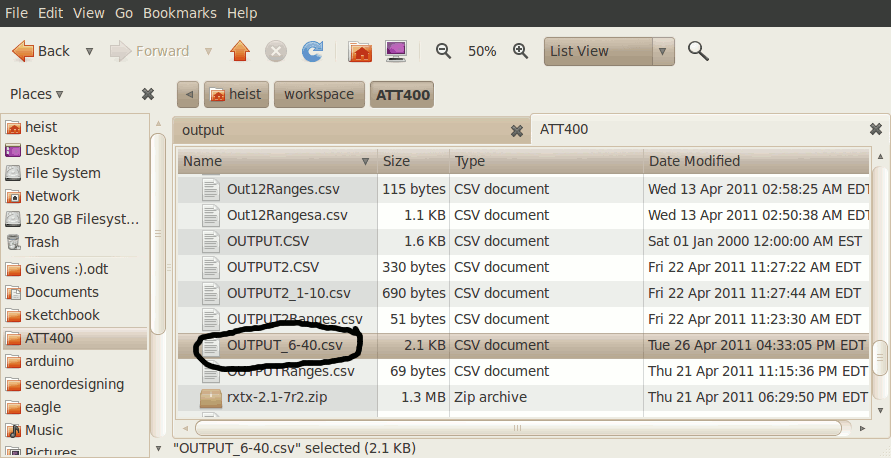


5. Click export



6. A file will be created in the current working folder with the format:

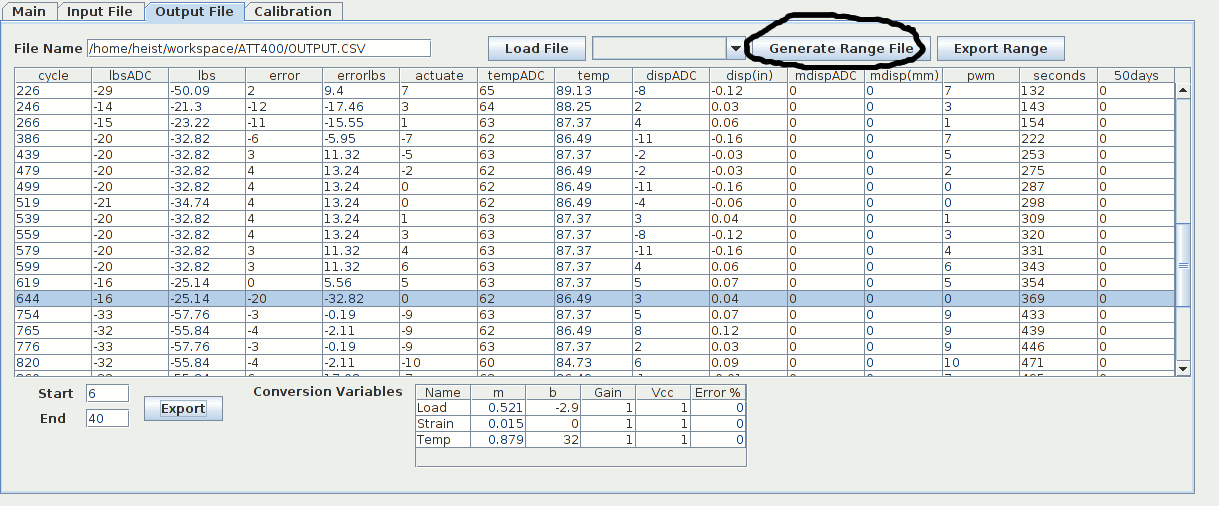
outputfilename\_**start**-**end**.csv

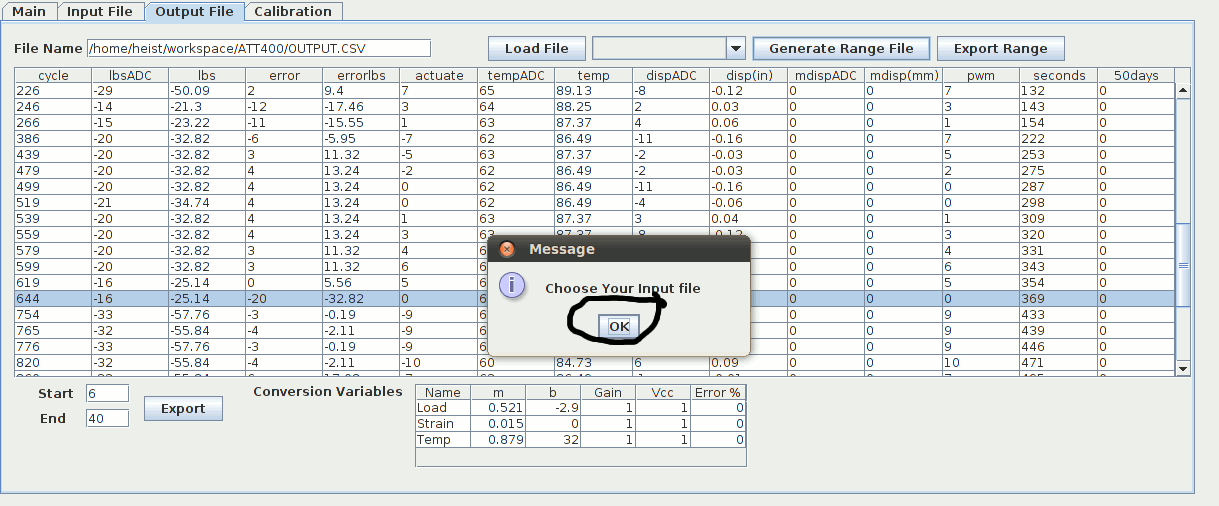


*C.Generating Range file*

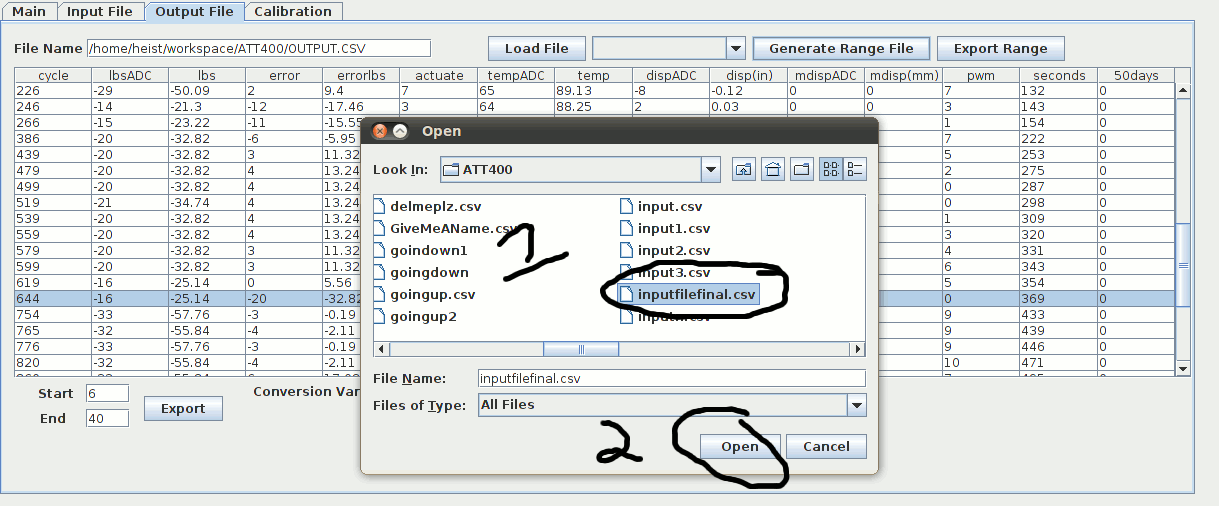
1. Follow Steps Viewing File Step and *hit cancel on step 5*

2. Click generate range line

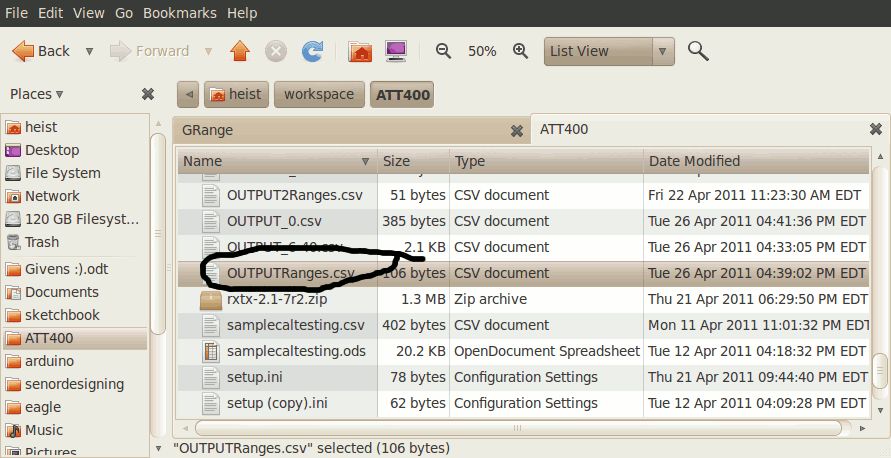
3. Click ok



4. Locate and select input file used in the experiment. Then press open

5. A file will be created in the current working folder with the format

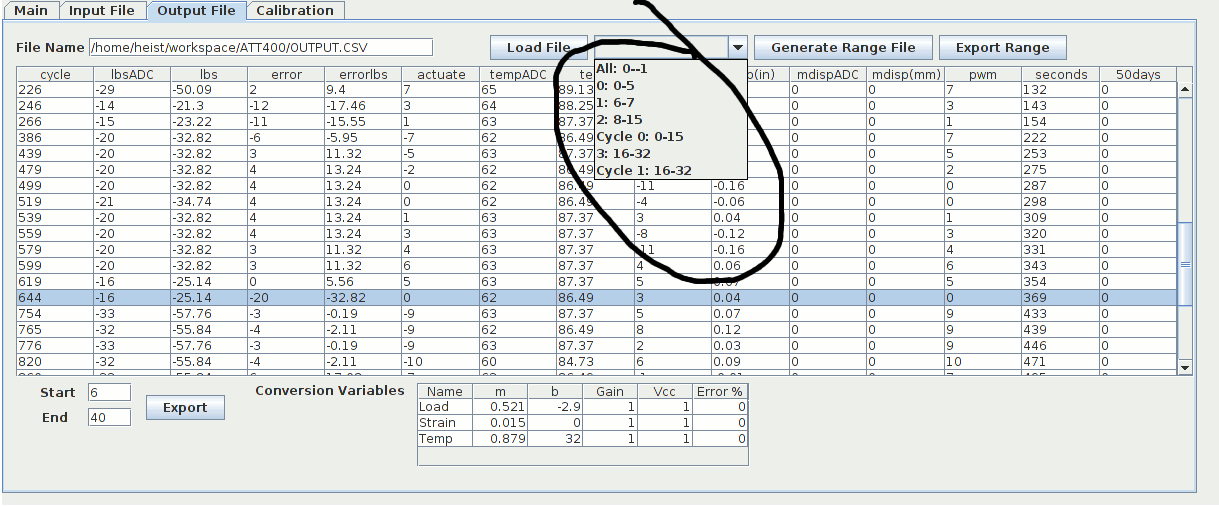
outputfilename**Ranges**.csv

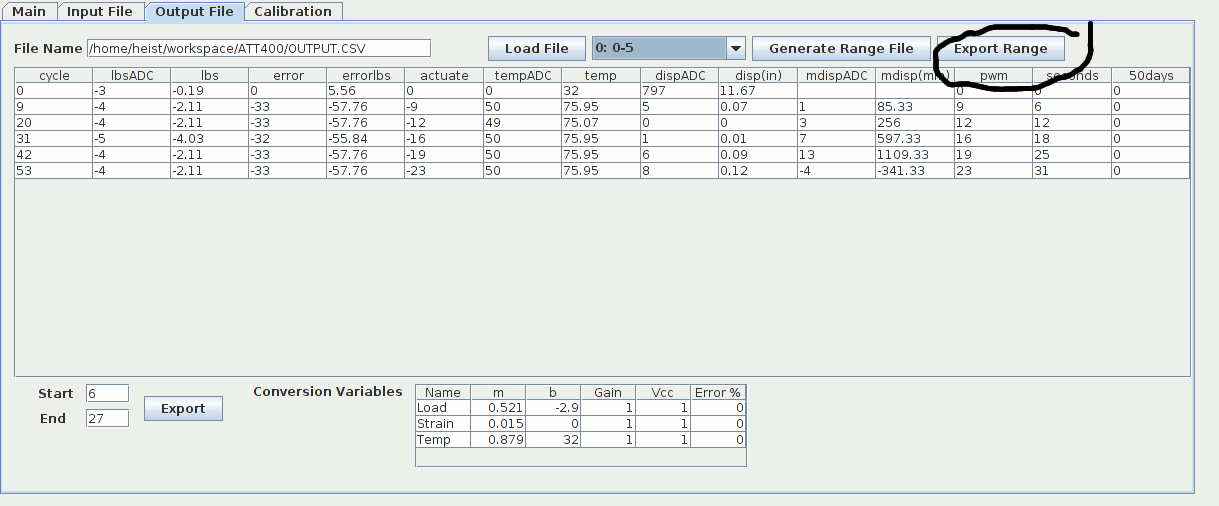


*D. Exporting Ranges*

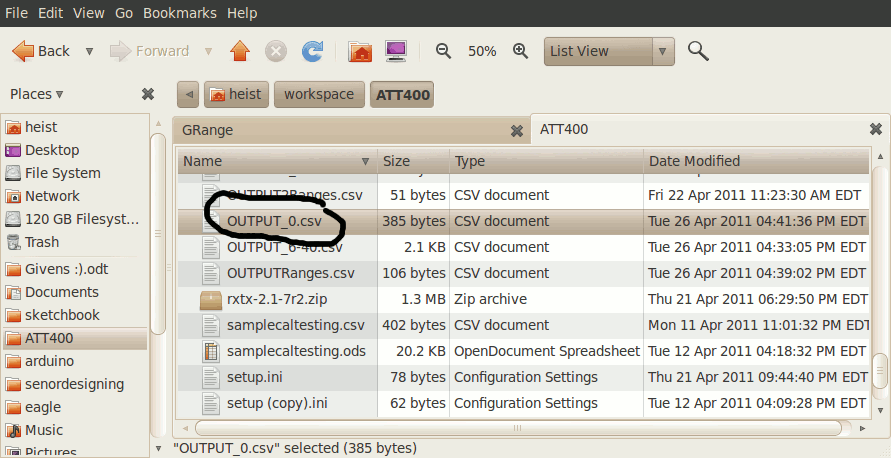
*1. Follow* ***Viewing File*** *Step*

*2.* Select a range from the drop down box

3. Click Export Range

4. A file will be created in the current working folder with the format

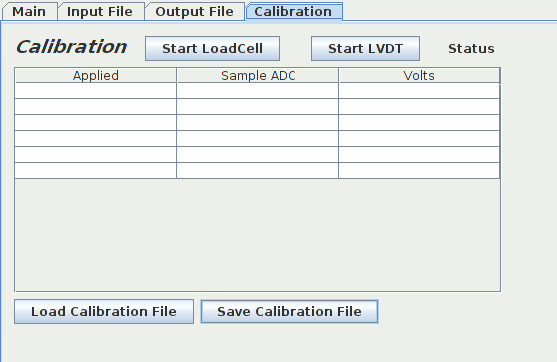
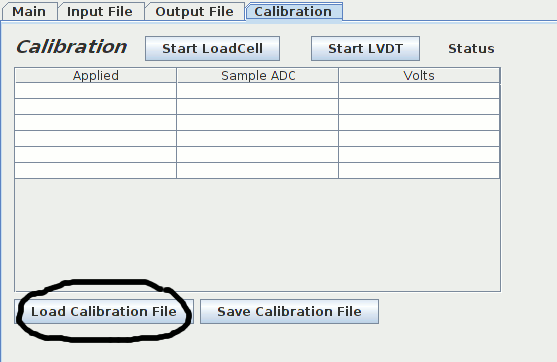
outputfilename\_**RangesName**.csv

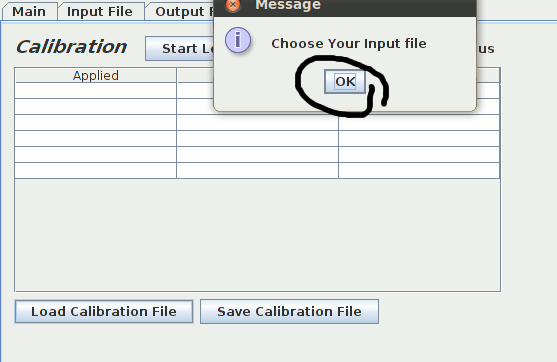


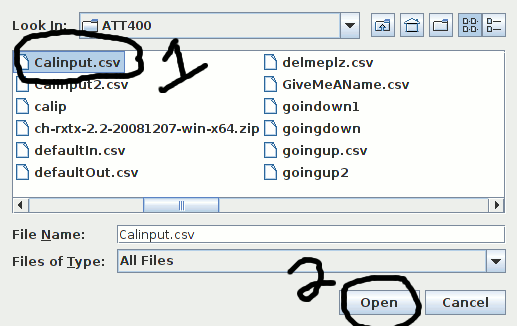
**Calibration File Interface**

1. Follow Connecting to device section to connect to a device

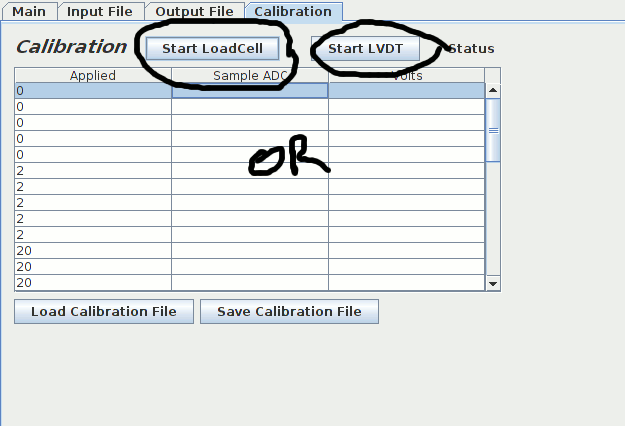
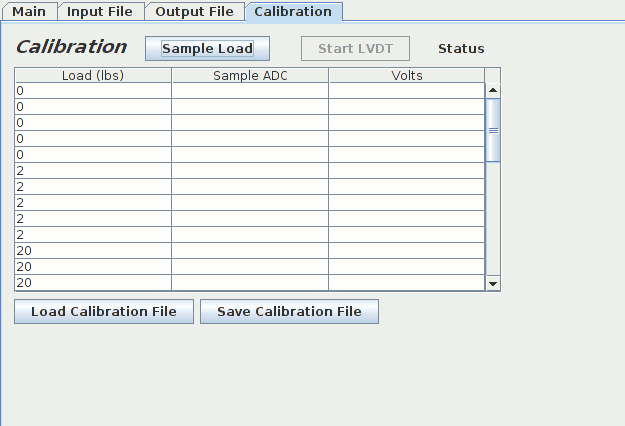
2. Click on calibration tab

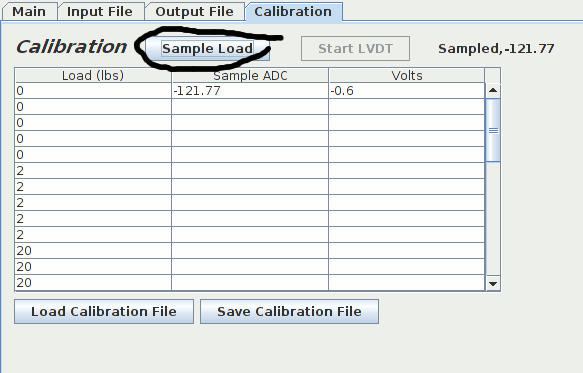
*3.* Click load calibration file

4. Press ok

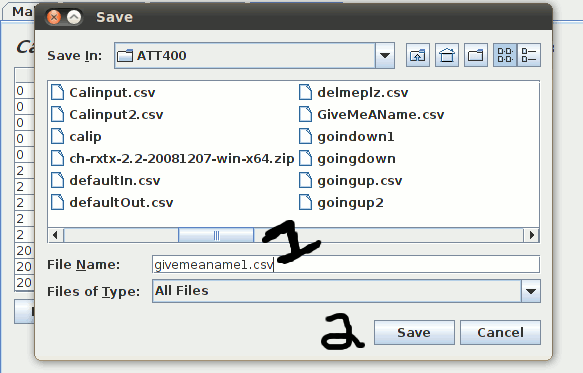
5. Select calibration file then hit ok

6. Select Either Start Load Cell to begin load cell calibration or start LVDT to begin LVDT calibration

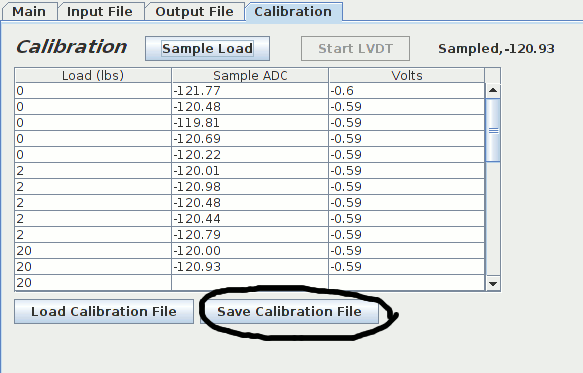
7. The fields should change appropriately

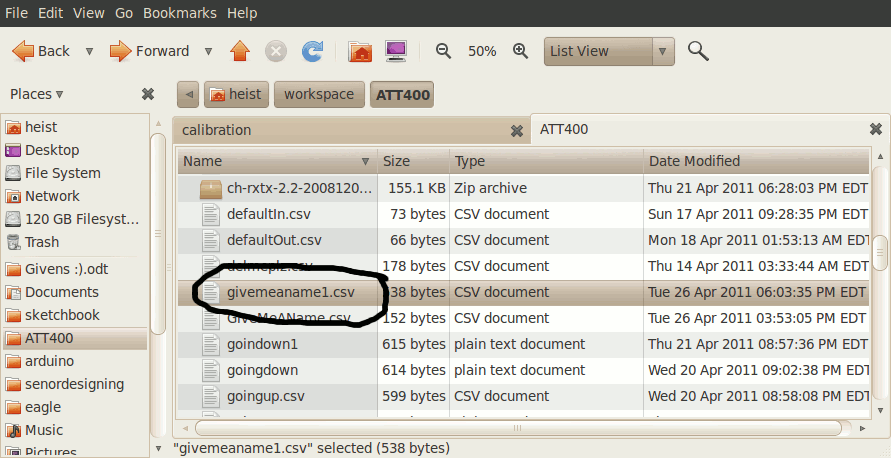
8. Begin sampling click the sample button once you would like to sample. You may click a field if you would like to start sampling from that location.

9. Once you are done sampling. Click save calibration file



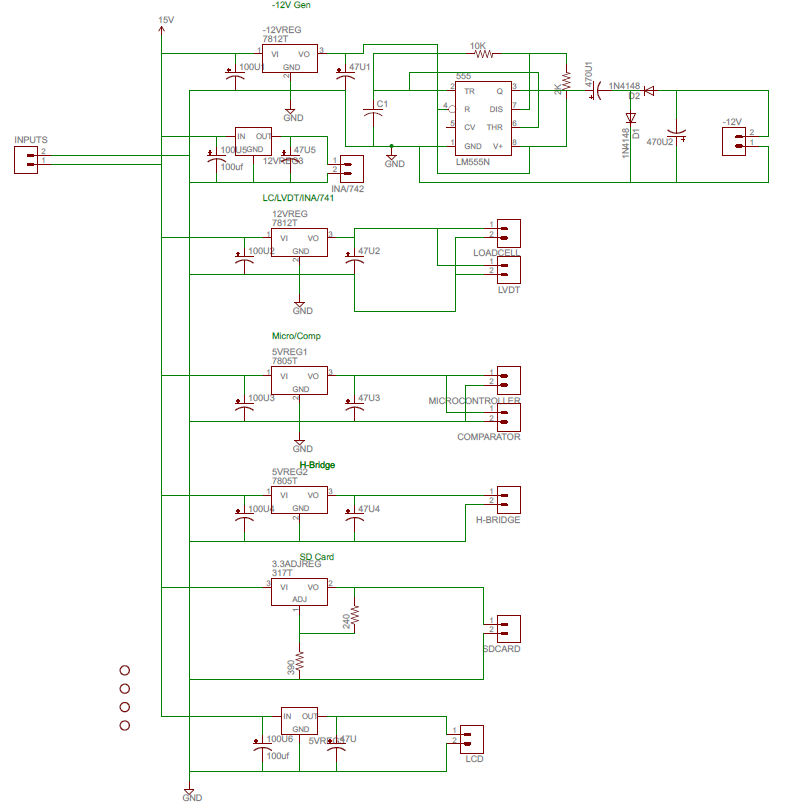
10. Name your file then click save



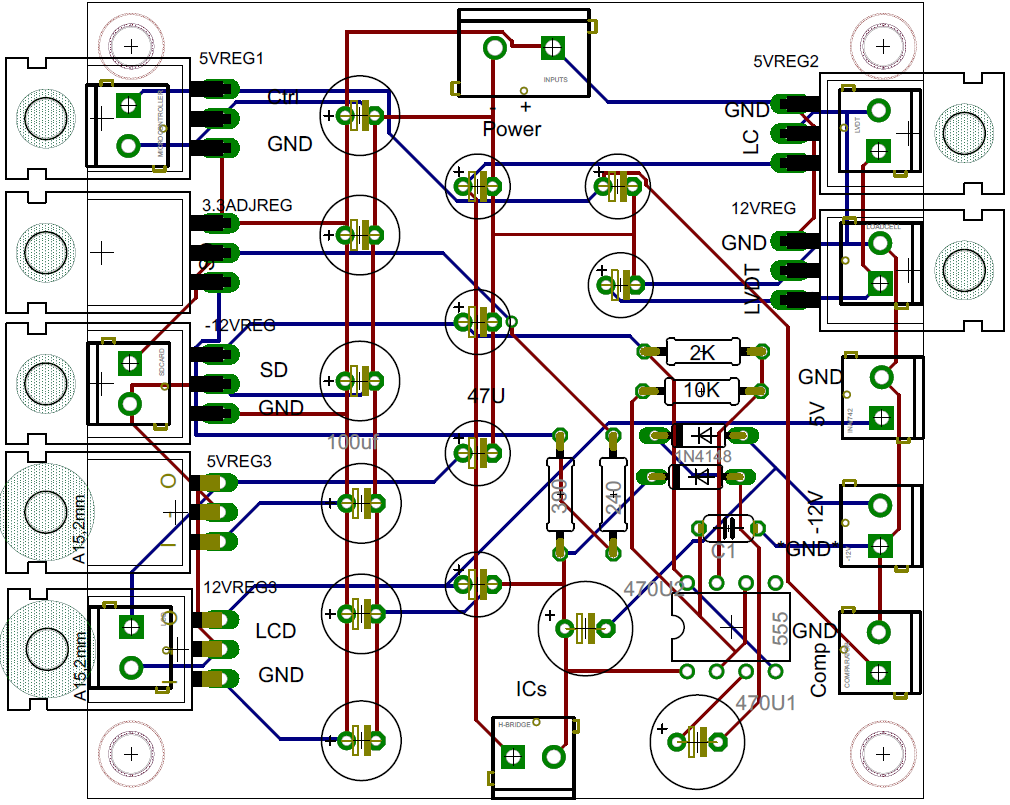
11. A file will be created in the current working folder

**Printed Circuit Boards**

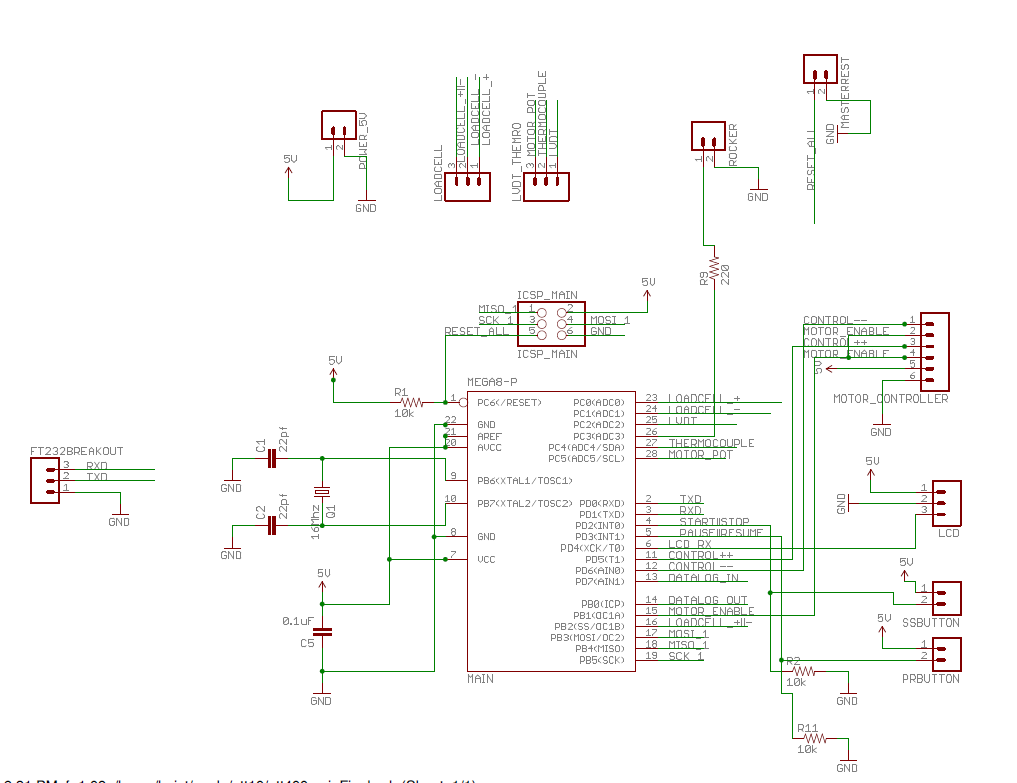
Analog Circuit Design

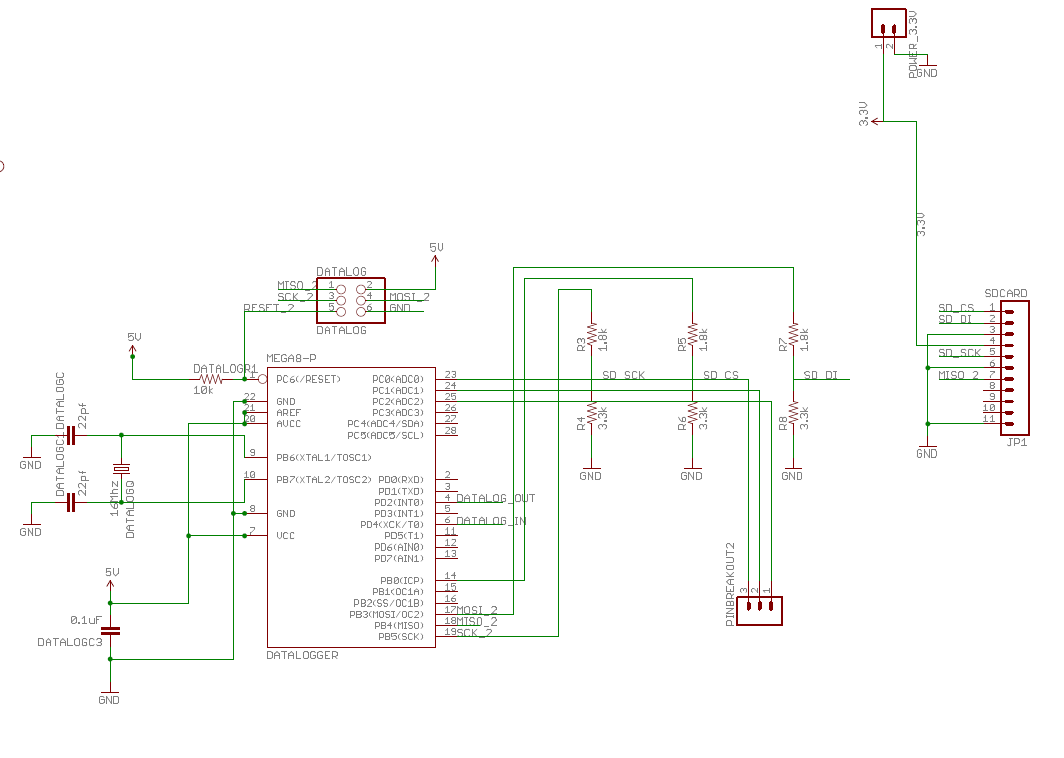


Analog PCB Layout

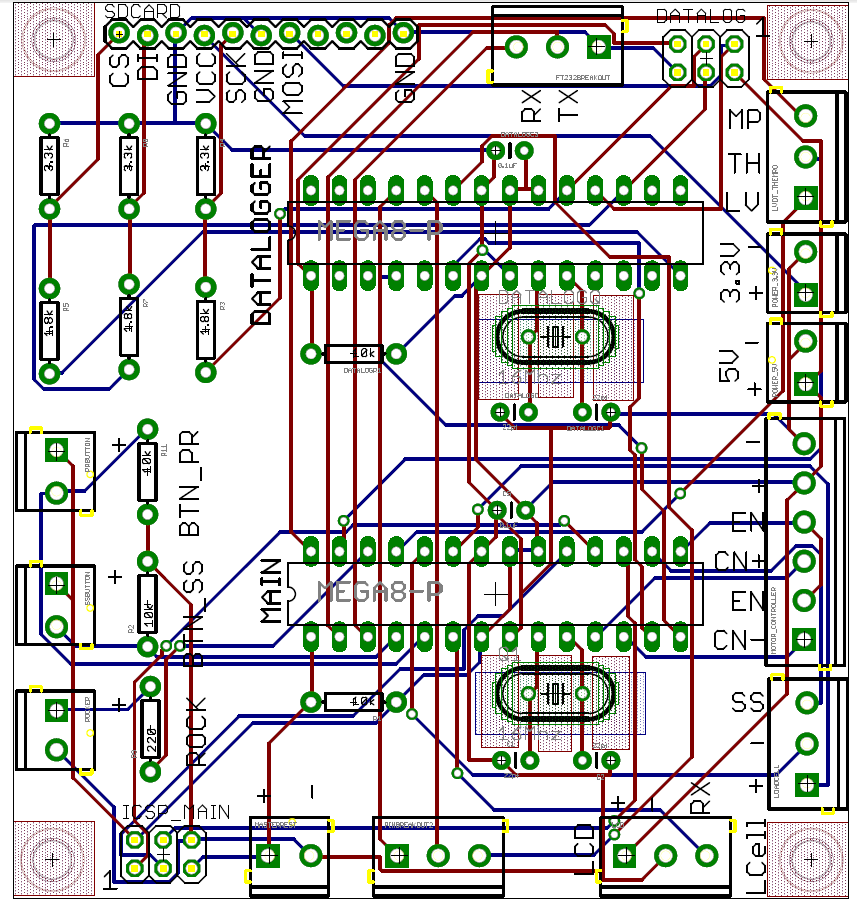


Logic Circuit Design





Logic PCB Layout



NON PCB Components

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Thermocouple** | | **LVDT** | | | | **Load Cell** | | | |
| IN+ | IN- | EX+ | EX- | Sig+ | Sig- | Ex+,Sense+ | Ex-,Sense- | Sig+ | Sig- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

**Table xx** - Sensor Inputs

**Thermocouple: LVDT: Load Cell:**

IN+ - YELLOW EX+ - RED EX+ - RED

IN- - RED EX- - BLUE EX- - BLACK

Sig+ - YELLOW Sense+ - ORANGE

Sig- - GREEN Sense- - BLUE

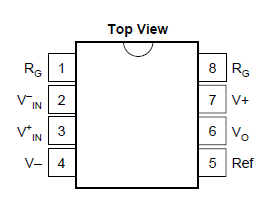
Sig+ - GREEN

Sig- - WHITE

**Table xx** – Sensor Wire Colors

Each block on the bottom of the table represents one screw terminal. Read as visualized from left to right, meaning the first screw terminal on the left belongs to the IN+ of the Thermocouple.

Load Cell



**Fig xx** – INA118 Pin Out for Load Cell Amplification

Rg1 - Rg8 ‒ 250Ω

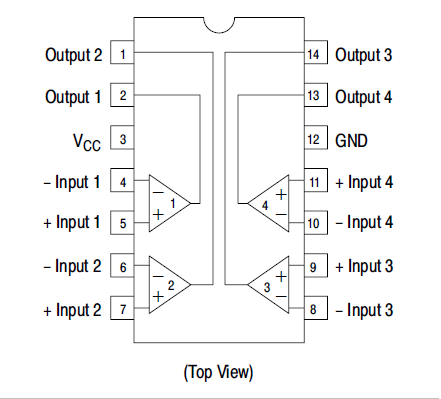
V- ‒ -12V

V+ ‒ 12V

Ref – Ground

V-in – EX/Sense-

V+in – EX/Sense+

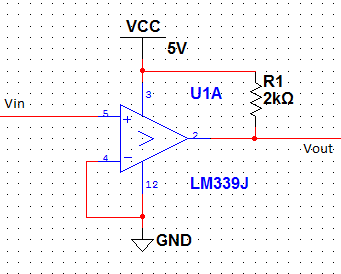


**Fig xx** – LM339 Comparator Pin out for Positive/Negative Selection

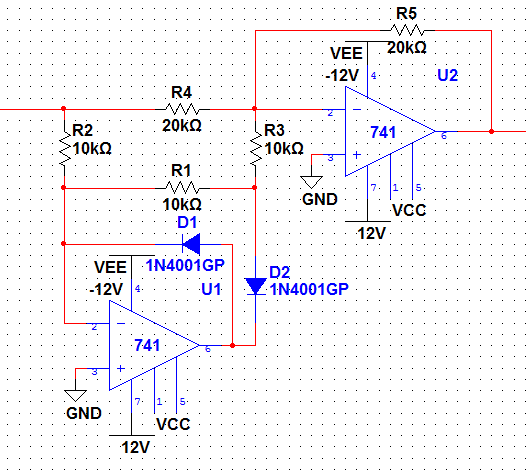
Vcc – 5V

All “-“ Inputs – Ground

GND – Ground

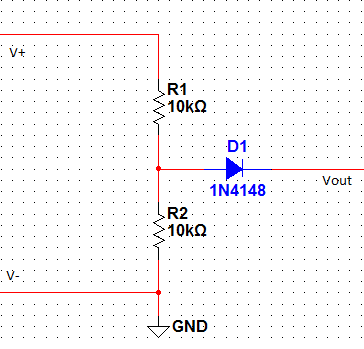


**Fig xx** – Selector Circuit with Comparators



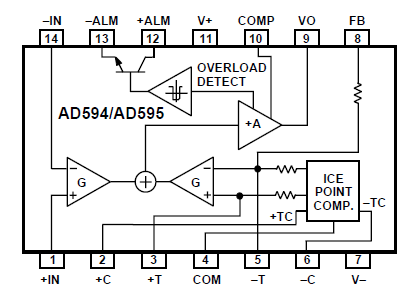
**Fig xx** – Full-Wave Rectifier with Operational Amplifiers

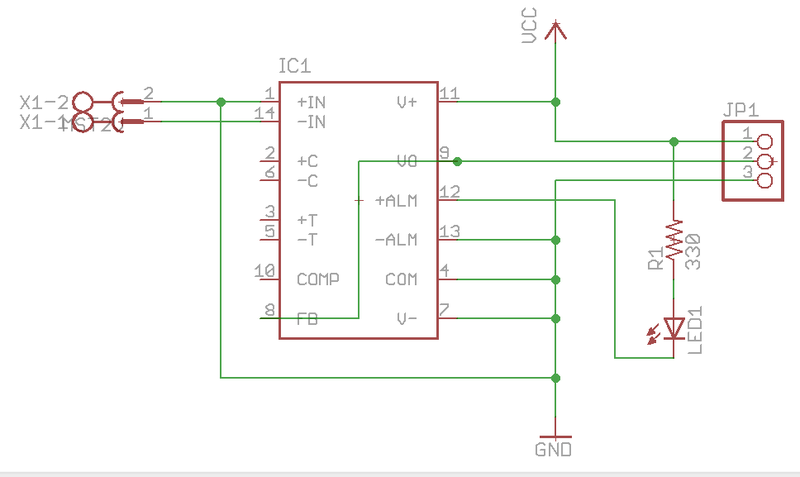
LVDT



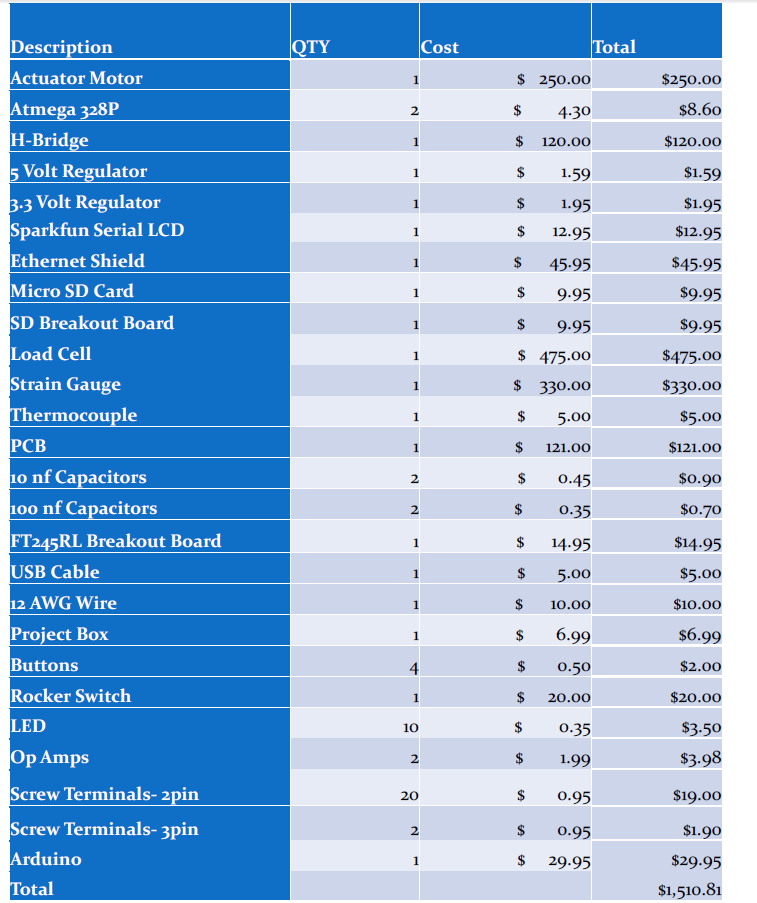
**Fig xx** – LVDT Voltage divider

Thermocouple



[](http://reprap.org/mediawiki/images/0/0a/Thermocouple_1.0_schematic.png)

**Budget**



**Group Responsibilities**

**Project MileStone**

|  |  |  |
| --- | --- | --- |
| 1/1/2011 | Finish Design |  |
| 1/7/2011 | Start Prototype |  |
| 1/13/2011 | PID algorithm version 1 | Start User Interface |
| 1/19/2011 | SD card library |  |
| 1/25/2011 | Lego Prototype Finished |  |
| 1/31/2011 | Finish CDR |  |
| 2/6/2011 | Breadboard LVDT |  |
| 2/12/2011 | Breadboard Load Cell |  |
| 2/18/2011 | Breadboard H-Bridge And motor |  |
| 2/24/2011 | Version 2 User Interface | Microcontroller Motor Control |
| 3/2/2011 | PCB preliminary design |  |
| 3/8/2011 | Power Systems version 1 | PID final algorithm |
| 3/14/2011 | PCB completed ready for review |  |
| 3/20/2011 | Power Systems version 2 |  |
| 3/26/2011 | PCB ordered |  |
| 4/1/2011 | Testing | PCB complete |
| 4/7/2011 | System Integration |  |
| 4/13/2011 | Testing |  |
| 4/19/2011 | Testing |  |
| 4/22/2011 | Presentation |  |

Project Summary and Conclusion

The overriding goals of this project were to create a strain/stress testing machine that not only functions properly, but can run without the assistance of a computer, be significantly less expensive than the models currently available, and have an easy to use interface. At the completion of this project, we believe that we have met all these goals substantially. Even with the general success of the project, we feel that there are many improvements that can be made, to increase the accuracy of all measurements, to expand the scope of the experiment, and to incorporate more tools.

To achieve greater accuracy of the measurements, we feel that we should use a stronger chip for the calculations. Currently, the smallest division in pounds is slightly less than 2. Over the range of 2000 total pounds of force, this is very little, but because we want to be as accurate as possible, we want this to be better. The same goes for the LVDT. Even though the curve is recognized as linear and the displacement is measured in millimeters, the movement of the specimen is usually in the realm of hundredths of millimeters. Increased accuracy in this range is a must.

With the circuits, to provide the most accurate measurements in an analog system, one should use precision resisters. The current resistors have tolerances of ±5%, which, at high resistances, can mean hundreds of Ohms of difference. So for systems like the full-wave rectifier and the voltage divider, these can change the output from its expected output value. Also more research can be done to determine if there are ways to eliminate the negative output from the LVDT without having the voltage-eating effect of a signal diode.

There also very small changes that can be made to make the system control not only one rig, but multiple rigs only limited by the amount of space on each SD card. Expansion of power supplies and power regulation PCB would allow for a hub/router type system to be developed.

To conclude we were task to build an inexpensive prototype strain/stress testing machine that not only functions properly, but can run without the assistance of a computer, be significantly less expensive than the models currently available, and have an easy to use interface. We completed this task giving the mechanical engineering department a base for newer version in the future. This product will save the university thousands of dollar and one day be the main teaching device in classrooms and labs.