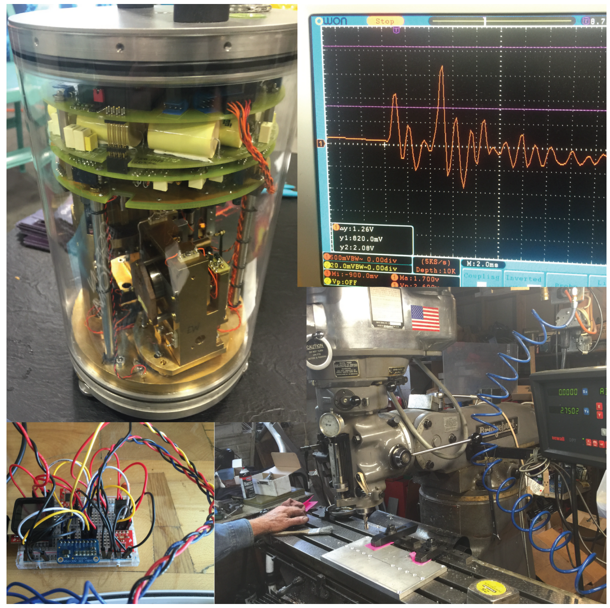
TGE Documentation

Release 0.1

Leeman, Marone

Jul 16, 2017

Contents



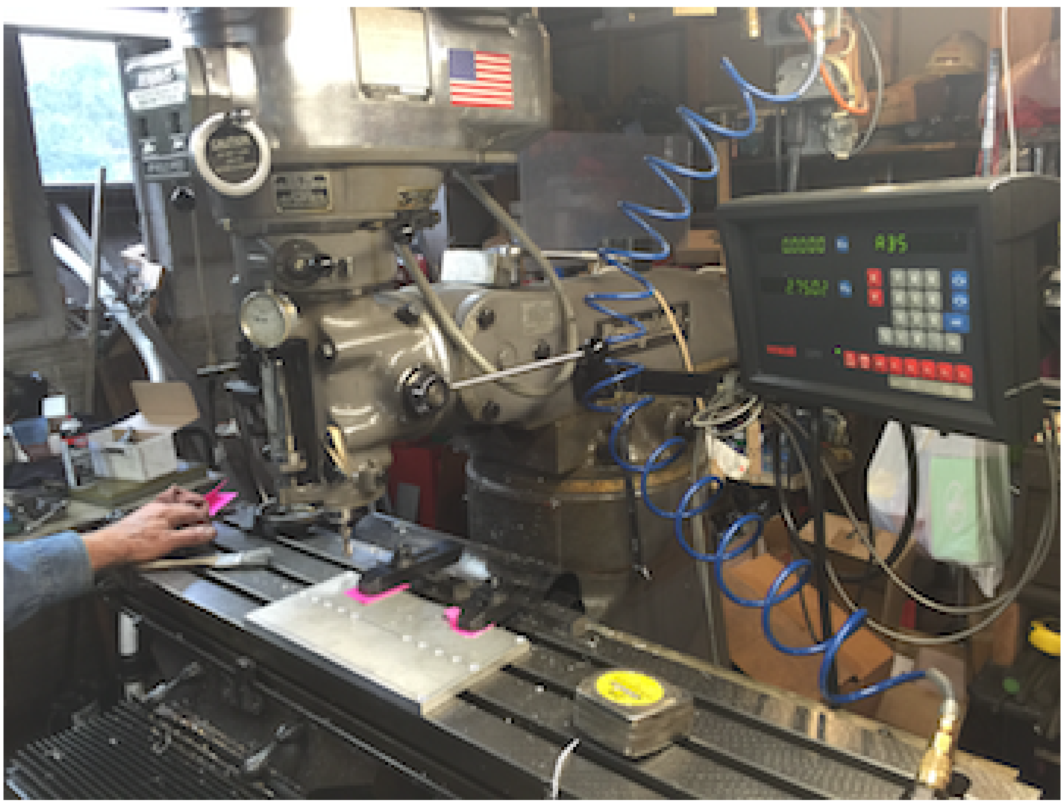
Techniques of Geoscientific Experimentation (TGE) is a resource designed to help scientists, hobbyists, and experimentalists learn how to develop their own experimental equipment. Topics include mechanical design, data acquisition, electronics, microcontrollers, CAD, pressure vessel design, and more.

This course material has been primarily developed by John Leeman and Chris Marone. The format and online format of the course are patterned after the excellent course structure developed by researchers and students at the University of British Columbia. Their courses are available on geosci.xyz. We encourage others to use the material with proper citation. Contributions are welcome and encouraged - the material improves though repeated use! Join the development on github.

CHAPTER 1

Contents:

Shop Tools and Practices



This course involves making things to accomplish scientific tasks. Making things necessitates the use of tools and mechanical/electrical workspaces. Learning which tools are used for certain tasks, how to use them safely, and how to care for the equipment are essential for success in the shop. In this section we introduce you to the basic tools that you will encounter and how to go home will all of the fingers and eyes you came to work with.

Contents:

Hand Tools

Combination Wrench Set (Wikipedia)

Hand tools are some of the most commonly used tools. Knowing the names of basic tools and how to use them will greatly speed up your work while also making it safer. A basic set of hand tools can be collected for a few hundred dollars and if cared for properly can provide a lifetime of service.

Contents:

Screwdrivers

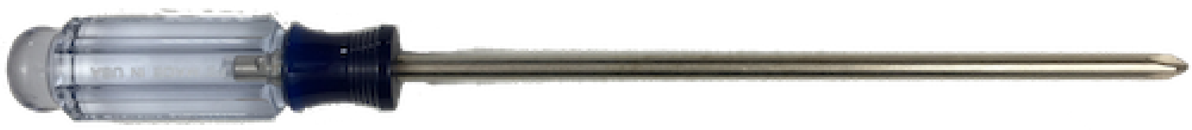
Screwdrivers are used to increase the torque you can apply to a screw by providing a large diameter and textured grip rigidly connected to a hardened shank with a driver ground into the end.

Types of Screwdrivers

Screwdrivers are available in a variety of shapes to allow access to screws in almost any possible orientation with many different clearances. Some screws may be buried deep in a recessed hole, requiring a long shank, while others may have almost no vertical clearance and require an offset screwdriver.

- **Normal** - The normal screwdriver has a shank whose length is roughly proportional to the size of the driver. These are the most commonly used screwdriver and should be the first thing you reach for.

- **Extended** - The long shank of the extended screwdriver lets you reach screws that are in deep pockets or across large enclosures.



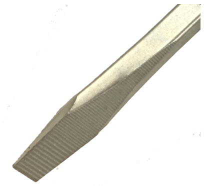
- **Stubby** - For screws that have less vertical clearance, such as those in recessed cavities or inside enclosures, a stubby screwdriver is often used. These short shanked tools often have large handles and are likely the second most common form factor.

- **Offset** - In some cases, screws have almost no vertical clearance. An offset screwdriver has a right angle bend and can be used to access such screws.
- **Ratcheting** - To increase how fast the user can turn the screw, some screwdrivers are equipped with a ratcheting mechanism. These are very useful, but should not be used for high-torque screws and they ratchet mechanism can become stripped.

Types of Drives

There are many types of drives, so many that it is really impossible to cover and not worth covering as you will commonly only encounter a few drives. Many drive types have been introduced by manufacturers to prevent tampering, others are used in certain niche applications only. A relatively comprehensive list of screw drives is available in a Wikipedia article.

- **Slot** - The first drive developed. It has been traditionally the most common because of its low cost of manufacture. Slotted screws are not common on anything that a power tool will be used on, as the blade slips out of the slot very easily. This could be dangerous to the part's finish and any nearby skin. You may hear this drive referred to as the common blade, standard, flat-blade, or flat-head drive.



- **Phillips** - This screw drive was dismissed by manufacturers in the 1930's, but has since become a very common drive. The cruciform nature of the phillips makes registering the driver into the screw easy and the blade cannot slip off the end of the head like common head screws. The Phillips does have a tendency to "cam out" at high torques. The bit will force itself up and out of the slots, possibly stripping. Other drives such as the PoziDriv were developed to combat this, but the Phillips is still commonly found in many applications. It is easy to confuse the Phillips with the very similar Fearson drive. Phillips drives have a blunt tip opposed to the sharp tip of the Fearson drive.

- **External** - External drives reverse the roles of the driver and screw head. The driver slips over a protruding pattern on the screw head, often square or hex. These provide a lower vertical clearance, but are much easier to tighten to high torques than common or Phillips drive screws.

- **Torx** - Torx is a trademarked name for a 6-point star shaped drive, generically referred to as the ISO10064 hexalobular internal drive. Torx provides a very good mechanical connection and can work in low vertical clearance situations.

- **Hex Socket (Allen)** - Similar to the external hex drive, the hex socket drive, commonly referred to as the Allen drive consists of a hexagonal pocket into the screw head and a hex shaped driver that fits into the "socket" head. This type of drive proves excellent mechanical connection and is often used in scientific equipment assembly.

Use

To use a screwdriver, first double check that you have the appropriate sized driver for the fastener you need to remove or install. Using the wrong sized tool will likely damage the fastener and make it difficult or impossible to remove. Insert the driver into the fastener and press straight down firmly. Turn the screwdriver clockwise to tighten or counter-clockwise to loosen a regular right-handed thread fastener. Occasionally you will encounter left-handed threads that are the opposite to this. As a rule of thumb, remember “righty tighty, lefty loosey.”

Screwdrivers are often used as pry bars, scrapers, or chisels, but they were **NOT** designed for these tasks. Improper use will damage the tip, bend the shank, and likely end in injury. A damaged screwdriver will then likely strip the screw it is used on next and create a cascading event of damaged hardware. Never use a chipped, bent, or otherwise mangled screwdriver.

Safety Precautions

The main safety hazards associated with screwdrivers are puncture wounds when the tool slips out of the screw head. Improper use of the driver, as a pry bar for example, could result in fracture and flying metal chips.

Files

Files are hardened tools that allow your to abrasively remove material from a workpiece. There are many different kinds of files, each suited to a specific job, but here we will cover the most common types of files commonly found in the shop or laboratory. Files are commonly used to perform fine sizing on a part, break sharp edges, or even produce odd shaped holes. Making a rectangular hole in a front panel for a switch or panel meter is an ideal example of where files would be used in the construction of laboratory equipment.

Types of Files

There is no standard or uniform rating system for files. They are generally manufactured in various shapes, roughnesses, and cuts. Common shapes for files include flat, round, half-round, triangular, knife edge, and square. The roughness of the file is classified as (from most aggressive to least): rough, middle, bastard, second cut, smooth, and dead smooth. Some Swiss files are number graded for roughness as well with lower numbers being more aggressive. Files can have grooves (teeth) cut in once angled direction (single-cut) or two directions (double-cut). To see all of these types of files explained, watch the video below.

When using a file, be sure to select the correct roughness for the job. Using a fine file of soft metal will load the file teeth with chips and quickly become ineffective and difficult to clean. Using a smooth file for roughing work will take much too long and excessively wear the file, while using a roughing file for finishing work will result in a poor surface finish and is likely to lead to the removal of excessive amounts of material.

Use

Using a file is as simple as firmly holding it from both ends and passing the cutting surface over the material with downward pressure. Do not file with excessive speed (about 1 pass/second is a good rule of thumb). Files should never be used without a handle - the tang of the file can be very dangerous. Before and after each use the file should be cleaned with a file-card and any stuck metal “pins” removed from the teeth. Files should be stored in such a way that their teeth are not scraping against each other and dulling all of the files in the drawer. For demonstrations of file use and care, see the video below.

Safety Precautions

When using a file, always make sure you are firmly gripping it and are not going to injure yourself when the file leaves the workpiece. Files must **always be used with a proper handle** or serious puncture wounds could result from the tang of the file. Other hazards include abrasions and cuts if the file slips and finds your hand, leg, etc. Always work on a firmly secured workpiece with eye protection.

Hammers

Hammers, sometimes casually called persuaders, are used to drive pieces together. Their form ranges from tiny hammers wielded by micro-scale hobbyists to hydraulic devices lifted over building pillars by cranes and operated by entire crews of people. For our discussion, we are going to talk about a few of the most common types of hammers you can expect to encounter.

Types of Hammers

- **Claw Hammer** - Commonly referred to as the carpenter's hammer, the claw hammer is generally used for nailing and pulling nails. The claw hammer is made of very hard steel and should never be struck against another hammer or metal object as the head may fracture and produce metal chips. The tapered slot or "claw" on the back of the hammer head is used to pry nails out using the extra leverage of the handle. An oversized version of the claw hammer is the framing hammer, commonly with a checkered hammer face to help keep each blow on the nail head.

Fig. 1.1: Claw Hammer (Wikipedia)

- **Ball-Peen Hammer** - There are many variations of the peening hammer (straight-peen, cross-peen, diagonal-peen), but the ball-peen or machinist's hammer is the most common. This hammer has a flat face used to strike chisels, punches, and other metal tooling and a hemispherical peening face used to work metal and round rivet heads. This is one of the most commonly used hammers when marking metal, prototyping, and working in the laboratory.
- **Brass Hammer** - Brass hammers are used in situations where sparks generated when striking an object could pose an explosion hazard. Such situations often arise when working with petroleum, natural gas, or other flammable materials. Brass hammers can also be used on objects that must not be damaged assuming that the brass is the softer material (though mallets are often used for this task as well).

Fig. 1.2: Ball-Peen Hammer (Wikipedia)

- **Dead Blow Hammer** - When working on precision work, the rebound of a hammer is often undesirable as the bounce could result in marring or damage of the part. On such delicate work, the very short duration over which the hammer's kinetic energy is delivered is also a problem. The dead blow hammer is generally hollow and filled with sand or metal shot (small metal balls). The shot lengthens the time of energy delivery and reduces rebound.

Fig. 1.3: Dead Blow Hammer (Wikipedia)

- **Mallet** - The mallet is a large faced hammer that is smaller than a sledge and is generally made from rubber, rawhide, or wood. Mallets deliver smaller amounts of energy with a soft and non-damaging face. They are generally used for small adjustments or in the assembly of delicate electromechanical devices such as motors.

Use

Using a hammer just takes practice and careful selection. Choosing the correct style and weight of the hammer can make a large difference to how successful your operation is. Raise the hammer and strike the object with the head of the hammer square to the surface being hit.

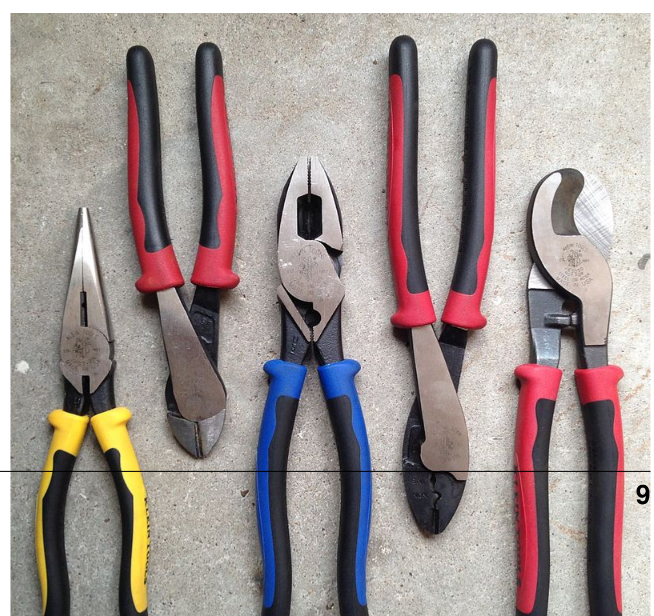
Fig. 1.4: Mallet (Wikipedia)

Safety Precautions

The largest safety hazard is crushing on fingers or toes. Being careful and practicing aiming the hammer blows will help reduce this risk, but even experienced carpenters and machinists will sometimes miss their mark. Using paper fixtures to hold nails safely away from your fingers when learning may be helpful. There is always the chance of the hammer head or tool being struck fracturing. If this occurs, flying metal fragments could result, so eye protection is essential.

Pliers

Pliers are used to bend, grab, compress, twist, and otherwise manipulate objects by amplifying your grip force with crossed levers. There are a blinding variety of pliers on the market in every imaginable shape and size. There are narrow pliers, tiny pliers, long pliers, curved pliers, and a thousand other speciality types. Here we will cover the basic types of pliers from which the others are derived. Again, there is little standardization on the size of each type of plier.



Types of Pliers

- **Slip Joint Pliers** - The slip-joint plier is possibly the most commonly used in the toolbox. It is a general purpose plier whose center joint can be offset to allow the user to grip a thicker object while still having the handles be spread a reasonable distance. Pliers that look identical, but have no adjustable jaw spacing are called combination pliers, but slip joint pliers are slightly more versatile.

Fig. 1.6: Slip Joint Pliers (Wikipedia)

- **Diagonal Cutters** - These pliers are designed to cut wire and other small gauge material. They do not shear the material like scissors, but instead their "V" shaped cutting surfaces indent and wedge the material apart. They are most commonly used to cut mechanic's wire (security wire) and other small materials in the shop.

Fig. 1.7: Diagonal Cutters (Wikipedia)

- **Lineman's Pliers** - Designed primarily for electricians, these pliers are used to cut, twist, and crimp wire. They have a large and heavy serrated gripping section at the nose of the plier, and cutting section in the craw (near the pivot), and an optional crimping area. These are used generally when mains wall power wiring is necessary. They are often certified to protect the user from a given voltage on a live wire.

Fig. 1.8: Lineman's Pliers (Wikipedia)

- **Needle-Nose Pliers** - The long and narrow nose of these pliers makes them ideal for reaching back into tight spaces to grab or twist wires, tubes, and other small items. Needle-nose provide excellent control and are used during electrical work and for small scale assembly. Many models have a cutter integrated into the craw of the plier. Many sizes of needle nose are made and keeping as many different sizes as possible in your tool box is a good idea.

Fig. 1.9: Needle-Nose Pliers (Wikipedia)

- **Locking "Vice-Grip" Pliers** - Locking pliers are used for jobs that need continued application of a very strong gripping force. These pliers have a clamping point that is adjusted by a thumbscrew. When gripped, if the break-over point is passed, the pliers will lock and maintain a strong grip. The grip is released by squeezing a small handle under the grip. The unlocking action can be sudden and violet, so use care to avoid injury when removing locking pliers.

Fig. 1.10: Locking Pliers (Wikipedia)

- **Tongue-and-groove "Channellock" pliers** - These pliers, sometimes referred to as "slip-joint" have offset jaws that can be set apart at several offsets. Similar to the combination/slip-joint plier, this means large objects can be easily gripped with small handle spreads. These pliers are often used when holding large nuts, bolts, or pipes.

Fig. 1.11: Slip-Joint Pliers (Wikipedia)

- **Ring Pliers** - Ring or "circlip" pliers have jaws that bend at a right angle to the plane of the handles and have two small circular pegs. These are used to compress and spread the diameter of retaining clips to install them into retaining grooves. There are different sizes for different sizes and styles of retaining clips. Do not use them for anything other than installing the clips as it will damage the pegs and make the tool useless.

Use

Pliers and squeezed to grip and cut or bend their target object. Locking pliers have some locking features that operate in various ways depending on the manufacturer.

Fig. 1.12: Ring Pliers (Wikipedia)

Safety Precautions

Pliers pose a large pinching hazard that can result in cuts and sub-dermal bleeding/blood-blisters. The most common injury is pinching a finger or finger webbing near the plier pivot when pliers slip off of their target object. When appropriate gloves can been worn. Eye protection should be worn, especially if any cutting operations will be taking place.

Saws

Hand saws are used to cut wood and metal down to size for a project. There is a huge variety of blades available with different numbers of teeth-per-inch, tooth rakes, and other characteristics that make each blade well suited for cutting certain materials. The saw terminology can be difficult to get around at first, but we want to introduce the basic saws that will be used to accomplish most prototyping work.

Types of Saws

- **Wood Saw** - These saws are generally relatively aggressive and are often used for quick cuts on boards and plywood. On very long cuts, binding can be a problem. It takes some practice to follow the marked cut line well.
- **Hacksaw** - Hacksaws are generally used to cut metal and plastic parts. For many lab applications, this will be the most common saw used. There are a variety of frames available, included small frameless blade holders for tight quarters work. Having multiple sizes of hacksaws and handles on hand is a must for any fabrication lab. The saw cuts on the push stroke, but is reversible.

Fig. 1.13: Hacksaw (Wikipedia)

- **Coping Saw** - The coping saw is used to cut shapes from wood. It has a small and very thin blade whose cutting direction can be pivoted with adjustments built into the handle. These are very useful for complex shapes, but care must be taken to not snap the blades.

Fig. 1.14: Coping Saw (Wikipedia)

- **Mitre Saw** - This saw is designed to work with a mitre box to make accurate 90 and 45 degree cuts. These cuts are useful when building frames, applying trim pieces, or anything else with a corner intersection that you desire to not have joined with a flat butting joint.

Use

Fig. 1.15: Mitre Saw (Wikipedia)

Align the saw with your mark and start with small and low pressure cutting strokes. Be sure to allow for the kerf (lost material) of the saw when aligning the cut. It is helpful to add a tick mark of the side of the marked line that you plan to cut. Support the workpiece, as a sagging cut section of material can bind the saw and make it difficult to use.

Safety Precautions

There is a significant cut and abrasion risk with saws. Be sure you know that the area below the cut is clear to avoid any accidental encounters with the blade. Also be sure that all fingers are clear and the workpiece is securely clamped before beginning to cut the material.

Squares

Squares are used to alight objects, mark straight lines, and check for proper fit or assembly of machines.

Types of Squares

- **Combination Square** - The combination square consists of a grooved ruler that accepts multiple different heads for different functionality. The most commonly used head has a 90 and 45 degree angle, along with a scribe and level used to check level/plumb of the part. There are also variable angle and “V” attachments used to measure arbitrary angles and the center of round dowels respectively.

Fig. 1.16: Combination Square (Wikipedia)

- **Try Square** - Named because it is often used to “try” a surface for squareness, the try square is most commonly used to test and mark wood pieces. It is commonly made of a wooden stock with riveted metal blade. The try square is very similar is use to the machinist’s square.

- **Sliding T-Bevel Square** - This square is often referred to as a "false square" as it is an adjustable blade square. The blade can be set to the desired angle using a protractor, or more commonly the T-bevel is used to transfer an angle by setting it on the original piece and then marking another piece.

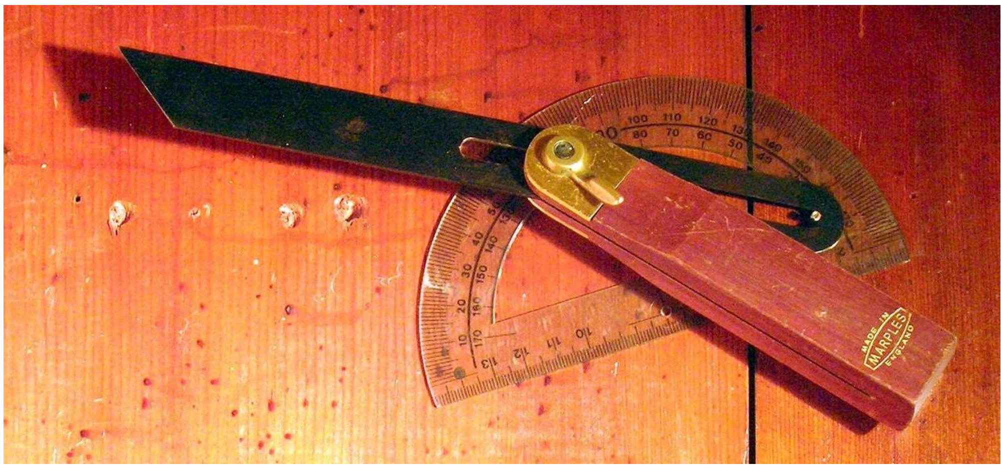


Fig. 1.18: T-Bevel Square (Wikipedia)

Fig. 1.17: Try Square (Wikipedia)

- **Machinist's Square** - The machinist's square is the metalworking version of the try square. They are manufactured to very high precision, generally in A and B grades with deviations of much less than a thousandth of an inch per inch of run. These squares are made of metal that is pinned and should be checked for accuracy periodically. The most common use of the machinists square is workpiece marking and tool alignment.

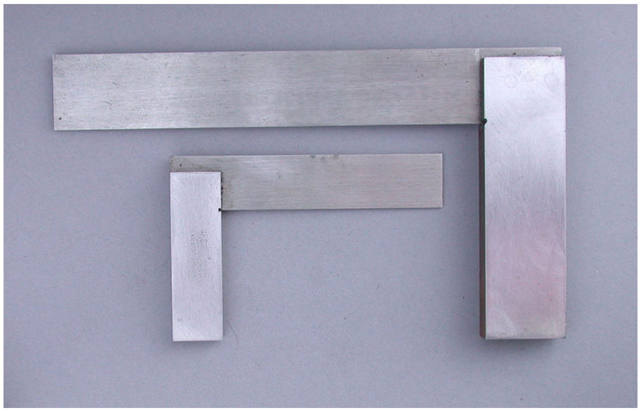


Fig. 1.19: Machinist's Square (Wikipedia)

- **Speed Square** - Invented in 1925, this is a very versatile square and one that is often reached for when making 90 or 45 degree marks. The square includes carpentry features to easily measure roof pitch and other common tasks. The speed square is available in 7" and 12" models made of metal or plastic. The Swanson tool co. (the original designers and manufacturers) offer a small book to accompany the squares describing their various uses.

Use

Squares are adjusted to the desired angle (if applicable) and aligned with the workpiece. They are then used as a guide, marking tool, or visual inspection tool. The video above demonstrates many of the common uses of squares in the shop. Squares that are dropped or experience extreme temperatures should always be checked before use as they could have been damaged.

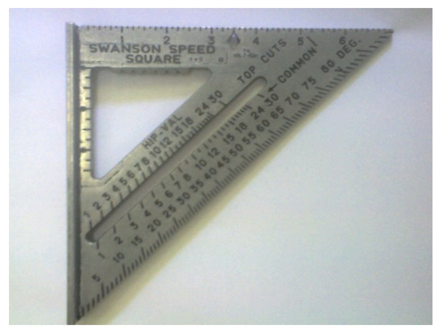


Fig. 1.20: Speed Square (Wikipedia)

Safety Precautions

Some squares may have sharp edges, wearing gloves is recommended with such squares. If the square is going to be used as a cutting guide, it must be firmly clamped, otherwise it could become a projectile.

Wrenches

Wrenches are used to amplify force through use of a lever to tighten of loosen nuts, bolts, pipes, and other threaded objects. There are many types (list), but here we will cover the non-speciality wrenches commonly used in prototyping, apparatus construction, and apparatus maintenance.

Types of Wrenches

- **Open End Wrench** - Open end wrenches fit around the faces of a nut or bolt head and are specified in the size of the opening in the jaw. You may also hear these called "open ended spanners". They are easy to rapidly apply and remove, but can slip off and caused rounded bolt corners and skinned knuckles. You must use an open end wrench when the top and bottom of the fittings are inaccessible, such as tube fittings.

Fig. 1.21: Open End Wrench (Wikipedia)

- **Box End Wrench** - These wrenches are completely closed, making them very secure on the nut/bolt being turned, but they can only be used when the top of the fitting can be accessed. When possible, use a box end wrench instead of the open ended. Box ended wrenches are commonly offset making their use easier in many situations.

Fig. 1.22: Box End Wrench (Wikipedia)

- **Combination Wrench** - The combination wrench has both an open and box ended wrench of the same size on the same tool. This is the most common wrench in the toolbox and the most versatile. When equipping your shop, these should be one of the first purchases in both imperial and metric sizes.

Fig. 1.23: Combination Wrench (Wikipedia)

- **Flare Nut Wrench** - The flare nut wrench solves the problem of tubing fitting accessibility faced by the box end wrench, while providing many of the same advantages. There is a slot in the end, allowing the wrench to pass over the tube, but still encircle the fitting to provide a positive purchase. The wrench is commonly much thicker than standard wrenches to distribute the tightening force over a larger surface area to reduce the risk of damage to pipe fittings, which are commonly made of soft metals.

Fig. 1.24: Flare Nut Wrench (Wikipedia)

- **Adjustable Wrench** - Often called the "Crescent wrench" after the brand name, this wrench is single handedly responsible for the destruction of more fasteners than any other tool in history. These wrenches have jaws offset from the handle about 15 degrees to help them work in close quarters. The jaw size can be adjusted with a worm gear drive to fit any nut size equally poorly. Tightening fasteners with adjustable wrenches requires careful adjustment to avoid the wrench slipping off the fastener, rounding the corners of the fastener, and making it difficult to tighten or remove. Only use adjustable wrenches when an appropriate fixed sized wrench is unavailable.

Fig. 1.25: Adjustable Wrench (Wikipedia)

- **Ratchet/Socket Wrench** - Ratchet wrenches have a one-way mechanism, allowing rapid tightening/loosening of fittings with a near full range of motion. They are available in multiple "drive sizes" that describe the size of the square shaft used to transfer torque from the wrench to the tooling. Generally sockets are used on the end to tighten or loosen nuts/bolts. At high torques the ratchet mechanism can fail, but these are generally the quickest way to work on assemblies with access to the tops of the fasteners.
- **Socket Accessories** - There are a number of accessories for socket wrenches. Typical accessories that should be in the lab box are extensions that allow deep reach into cavities with the sockets, universal joints that allow the wrench and socket to operate at an angle to each other, and crow-foot tools that allow access to nuts/blots with limited clearance and no top access.

Fig. 1.26: Ratchet Wrench (Wikipedia)

- **Breaker Bar** - This solid bar fits sockets of a fixed drive size and allows very tight torques to be applied without fear of stripping the ratcheting mechanism. These bars are often very large in diameter and long to provide extra mechanical advantage. When even more leverage is needed, users often slip an extra length of pipe over the handle to further extend it (a "cheater bar"). This practice is dangerous as it could case the drive to shear and all of the parts of the assembly to fall, possibly crushing hands or toes. The rapid release of energy can also cause head trauma if the user is in-line with the handle.

Fig. 1.27: Breaker Bar (Wikipedia)

- **Torque Wrench** - This ratcheting handle has a mechanism to measure torque exerted. Generally these wrenches have a set-point after which the wrench will break-over indicating that the desired torque has been reached. Torquing fasteners to the proper specification is important for many precision assemblies. Wrenches that range from inch-pounds to hundreds of foot-pounds are commonly found in lab tool boxes.

Fig. 1.28: Torque Wrench (Wikipedia)

- **Pipe Wrench** - These adjustable jaw wrenches strongly grip circular tubes when pulled towards the jaw opening, and release when pulled in the other direction. This makes it easy to rapidly tighten pipes. The teeth of the wrench generally mar the finish, but this is not considered as important as a tight connection for plumbing applications. Pipe wrenches are another often abused tool, used to tighten large nuts when the correct tooling is not available. Again, this is not recommended.

Use

Wrenches follow the same principle of "righty tighty, lefty loosey" as screwdrivers for right-handed threads. Socket wrenches have a lever to reverse their direction of operation.

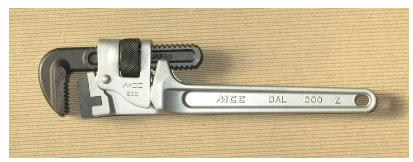


Fig. 1.29: Pipe Wrench (Wikipedia)

Safety Precautions

Pinching and crushing hazards are the largest safety issues when using wrenches. If the tool slips off of the fastener, it is easy to skin, pinch, or crush your hands. Gloves may be worn during operation, but an engaged and attentive user is the best safety precaution. **NEVER** use a wrench for anything but its intended purpose, doing so will likely result in damage to the tool and you. Using the wrong tool is also far more likely to strip the fastener, which should be avoided. As mentioned above, extension bars are also not a good idea as they often over stress the tool.

Machine Tools

CNC Workstation (Wikipedia)

Machine tools are a special category of power tools used to make precision parts and equipment. The main tools of the mill, lathe, and drill-press will be introduced, along with a few other useful, but less common tools. Learning to use these tools requires practice in the shop with an experienced supervisor to ensure your safety. Here we will introduce the machines, discuss basic use, and some safety precautions that should be observed when machining.

Contents:

Drill Press

Fig. 1.30: Drill Press (Wikipedia)

The drill press (pedestal drill) is a fixed drilling machine that is found in almost every shop. They are available in everything from bench-top sizes to floor-standing machines much taller than their operators. Drill presses are commonly sold based on their "swing", which is twice the distance from the drilling head to the support column (the machine's "throat"). The table of the drill press allows the operator to effectively clamp the work down and drill holes in a fixed orientation with much better positioning than possible with a hand drill. The levers used for plunging the tool into the work also provide some mechanical advantage to help increase the down-pressure on the bit.

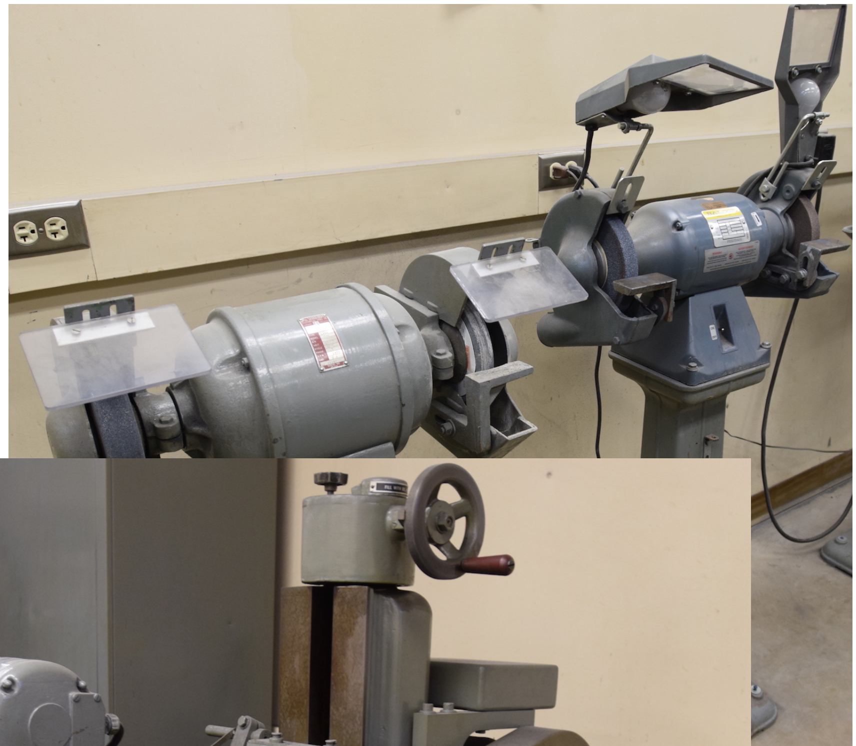
Most common drill presses are driven by a set of pulleys and belts in the top housing of the machine. These offer many different speeds for drilling holes of different sizes in different materials. There are machines with geared heads available that will not slip as belts sometimes do. They are most commonly found in heavy industrial settings where metal is being drilled and high-torque, low-speed operation is the key. Machines called radial arm dill presses have a head that can swing around the column, providing a very large working area and reducing the amount of re-clamping necessary to drill holes over large parts.

Use

Many videos focus on working with wood, but similar principles apply to metalwork as well. The video below is a good two-minute introduction to the drill press.

Safety Precautions

The main safety hazard with the drill press is the workpiece becoming caught in the drill and spinning. The best way to combat this is good clamping and careful selection of the drill bit and speed to match the material being drilled.



Grinders

Grinders are used to abrade away material with abrasive disks and brushes. The most common forms are the bench and angle grinders. For grinding surfaces to be very precise and smooth, there are also surface and cylindrical grinders that have grinding wheels held in precision stages and the part is moved via an automatic feed mechanism.

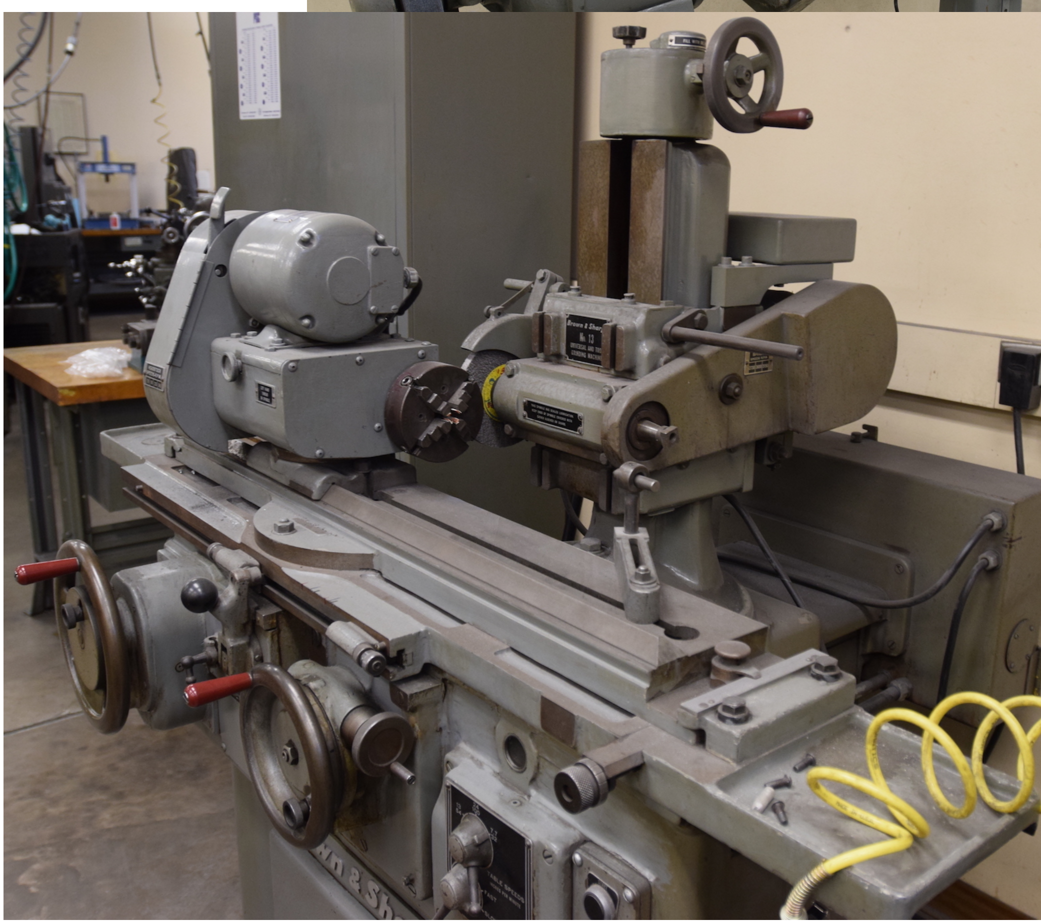


Fig. 1.32: Cylindrical Grinder (Ellis Dunklebarger)

Fig. 1.33: Surface Grinder (Ellis Dunklebarger)

Use

Grinder use is very easy, just be sure your hands cannot slip into the wheel in any way. Take care to keep the part cool and not use excessive pressure.

Safety Precautions

Protective eyewear, hearing, respiratory protection, and clothing should be worn. The grinder produces high temperature, high speed, flying bits of metal that you do not want in your eyes, lungs, or skin. **Never** stand in front of the grinding wheel, but off to the side. Grinding wheels are very brittle and will at some point fail, especially if there is a small defect due to an impact. The exploding wheel will hurl high velocity chunks that can easily penetrate the body.

Electrical Discharge Machining (EDM)

Fig. 1.34: Photo of an EDM (Ellis Dunklebarger)

One of the most precise and expensive machining techniques available is the electrical discharge machining (EDM) technique. A conductive part is mounted to a computer controlled table and submerged in a dielectric fluid (commonly oil or de-ionized water). An electrode is then brought very close to the part and a pulsed electrical power applied. The voltage between the electrode and part reaches the breakdown point of the dielectric and a spark occurs. This spark removes material from the workpiece and the electrode. After the spark is complete, fluid rushes in to sweep away waste material and cool the part. The workpiece is slowly eroded away until it is the desired shape. Electrodes are a significant cost of operating the machine. Systems that use a wire electrode pass it through the cutting area only once, then it is discarded.

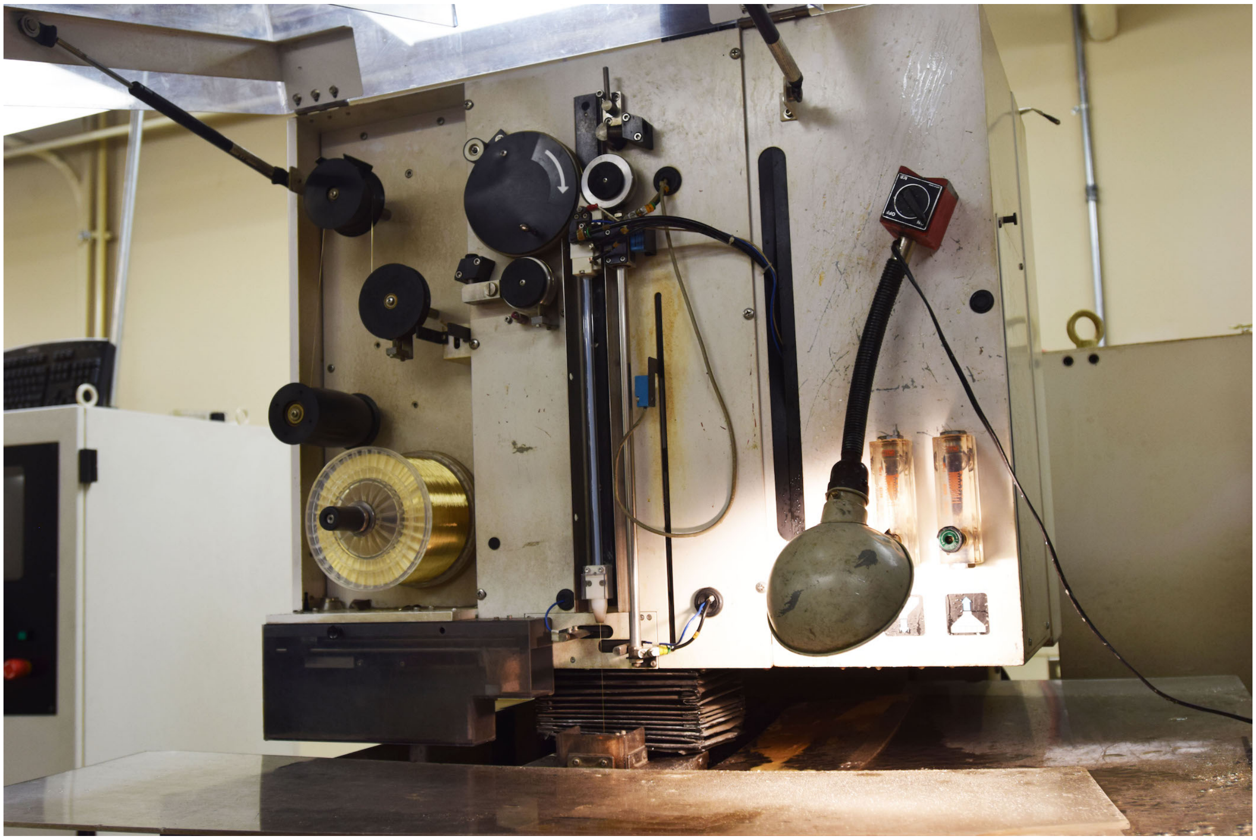


Fig. 1.35: Wire EDM (Ellis Dunklebarger)

The "kerf" of this process is very consistent around 0.001 in (0.03 mm). Therefore, machining accuracy of about 0.001 in (0.03 mm) is common, with an order of magnitude improvement possible for very high tolerance parts. The EDM surface finish is also very good.

It is also possible to build a basic EDM machine for use in the home or lab shop.

Safety Precautions

The main safety hazards with EDM are shock and slip. Avoid contact with the dielectric fluid or part when the machine is in operation. If the dielectric is spilled or is on your hands, make sure it is promptly cleaned as it is a slip and dropping hazard. Talk to your machine shop supervisor to learn any safety precautions specific to your machine and shop.

Lathe

The lathe is one of the most commonly used tools in the machine shop. It rotates the workpiece that is gripped in a chuck. The headstock that drives the chuck may be belt or gear driven, but is adjustable in speed to generate the correct surface speed for a given part size and operation. The material is cut by tools held stationary on a carriage that travels up and down the lathe bed. The lathe is the only machine tool that can completely build itself.

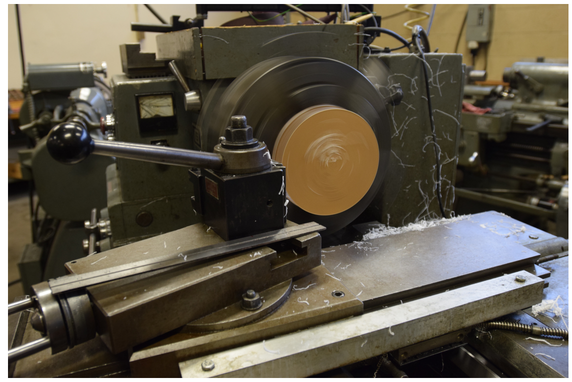


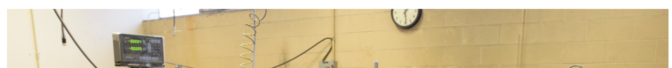
Fig. 1.36: Turning Acrylic (Ellis Dunklebarger)

Lathes are commonly used to turn down parts, bore holes, cut threads or grooves, facing parts, and cutting off parts. Learning each of these operations is relatively simple. The videos below provides a good overview of lathe operation.

Safety Precautions

The biggest safety hazard is becoming entangled in a rotating part and having body parts sheared off by the machine. Hair, clothing, and anything else hanging from the body present hazards. Also, never try to clean a rotating part (especially threads) with a cloth as it will become entangled and pull your finger/hand in. The other main hazard is a workpiece coming unclamped and flying out of the machine. Proper eye protection should also be worn.

Mill



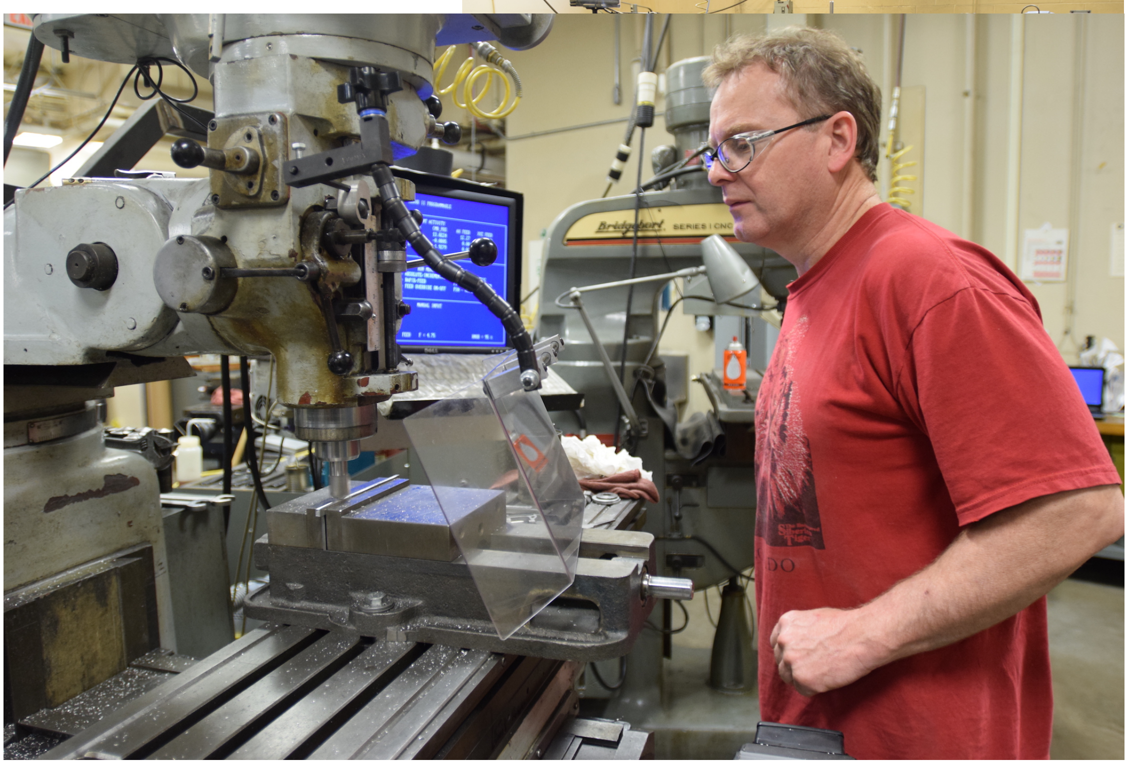


Fig. 1.38: Milling machine in operation (Ellis Dunklebarger)

The milling machine brings a rotating cutter into contact with a stationary and clamped workpiece that is help on a precision table. Milling can produce very flat and high precision parts with contours, holes, grooves, and other complex features. CNC milling machines have made it possible to turn a solid block of material into a complex shape in a matter of minutes to hours.

Milling machines are generally found in the vertical and horizontal types, referring to the axis of rotation of the tool. The vertical mill is most commonly found in shops, but well equipped shops have both types. Horizontal mills can be setup with ganged cutters to reduce the number of operations that must be performed on a piece during production.

Fig. 1.39: A CNC milling machine with tool changer (Ellis Dunklebarger)

Use

The mill, in principle, is very basic to use, but learning how to accurately operate the machine and its accessories takes some time and practice. The videos below are a good starter series to help you learn the basic operations.

Safety Precautions

The biggest safety hazard is being cut or entangled by the rotating tool. Hair, clothing, and anything else hanging from the body present hazards. Also, never try to clean chips away with your hand. Proper eye protection should also be worn.

Plasma Cutters

Plasma cutters use an electric current to create an accelerated jet of hot plasma. This heats and melts the metal, with a jet of air blowing away the molten metal. A plasma gas (often an Argon mix) is also consumed during the cutting operation. Plasma cutters produce a very clean cut and are very easy to use. CNC models are available that can cut very complex designs from large sheets of metal in minutes.

Use

The plasma cutter is almost a "point-and-shoot" activity, but there are a few things to know before you use one. The video below does a good job of showing how to perform a basic cut.

Safety Precautions

The main safety hazards are flying hot metal and UV exposure. Protective dark eyewear, gloves, and clothing should be worn. Be careful not to allow the cut parts to fall on your feet when cutting.

Water Jet

The water jet cutter, if you have access to one, is an incredibly useful and precise tool. It propels Garnet particles with a very high pressure jet of water to abrade through the workpiece. Multi-axis machines can even cut very complex 3D shapes. Water jet cutters make very clean cuts that are easily good to a 0.001" (0.03 mm) tolerance. These machines are exceedingly expensive to own and operate, so it is common to outsource water jet projects.

Fig. 1.40: Waterjet cutting a thick steel plate (Ellis Dunklebarger)

Use

Each machine operates slightly differently, but the basics of water jet operation are outlined by Dan Gelbart in the video below:

Some examples of using a water jet to create 2D and 3D parts:

Safety Precautions

The main hazard from a water jet cutter is instantly cutting off fingers if the jet is turned on while your fingers are near the nozzle. Water jets can easily cut through several inch thick steel plates, so humans pose no challenge.

Measurement Tools

Constructing laboratory equipment requires the use of accurate measurement tools to produce parts with tight tolerances. In this section of the course we will cover some of the basic measurement tools used in the laboratory and show

Fig. 1.41: A waterjet cutter in operation (Ellis Dunklebarger)

you how to read them accurately.

Contents:

Tapes and Rules

Constructing laboratory equipment requires the use of accurate measurement tools to produce parts with tight tolerances. In this section of the course we will cover some of the basic measurement tools used in the laboratory and show you how to read them accurately.

Tape Measure

The tape measure is used for rough measuring, cutting of stock, and often wooden and metal frame construction. It is a spring steel material that is rolled into a ruggedized carrier. The tape measure can be extended, locked, and used to measure easily down to the 16th of an inch. Tape measures have a few special features that allow easy measurement in a variety of construction scenarios. See video below for a few tips and tricks that you probably didn't know you could do with a tape measure.

Folding Rule

The folding rule is a predecessor to the tape measure, and often has about the same accuracy. Its advantage is that it is a rigid device, so it can be used in the situations where the measurement device needs to extend a significant length without bending, warping, or breaking over as tape measures are known to do.

Steel Rule

The steel rule is one of the most common measurement devices the laboratory and is easily read to a fraction of an inch. Its rigid construction with fine engraved measurement fiducials makes it a pocket essential when working in the machine shop or laboratory.

Micrometers

The micrometer is a precision measuring instrument with one fixed and one movable face. The movable face uses a screw mechanism that allows very accurate measurement of the distance between the faces. Variations of the micrometer measure inside, outside, and depth of a part feature. In addition to the dial caliper, the micrometer is a must for every laboratory toolkit. For very precise measurements, care should be taken to not allow body heat to expand the micrometer frame. The zero error of a micrometer should always be checked before any measurement is commenced, as dirt on the faces is enough to throw off the reading.

Calipers

Calibers are devices used to take a measurement of a physical object. There are a few different types of calipers that we will discuss. Dial calipers are the most commonly used instrument in the machine shop and laboratory, but inside and outside calipers also have their place in the tool chest.

Inside Calipers

Inside calipers are used to measure the inner diameter of a part, often a pipe or tube. They are commonly spring loaded with an adjustment nut and two outward facing measurement probes. The user adjusts the caliper until it just drags on the inside diameter that they wish to measure, then uses micrometer, steel rule, or other measuring device to measure the distance between the points of the caliper.

Outside Calipers

Outside calipers are used to measure the outer diameter of parts, such as tubes, pipes, or other semi-uniform parts. The user adjusts the points of the outside caliper until they drag on the surface to be measured. Another measurement device such as a steel rule can be used to measure the distance between the points of the caliper.

Dial Calipers

The dial caliper is the most commonly used measurement instrument in the machine shop or lab. It consists of a sliding carriage on a scale with hardened jaws at the end of the carriage and scale. The coarse measurements, commonly inches and tenths of inches are read off the main scale. A rack and pinion gear system drives a dial to allow measurement easily to the nearest thousandth of an inch. These calipers are also available in digital form. Both styles have a locking mechanism so the user can set a fixed distance and use the hardened points of the caliber as scribes to mark parts. These are easier to read than Vernier calipers, and have become much more common. The dial caliper can be used to measure inside and outside diameters, as well as depths in certain cases.

Vernier Calipers

The Vernier caliper is similar to the dial caliper except that it uses a sliding Vernier scale instead of the dial rack and pinion mechanism. Similar accuracy can be achieved, and Vernier calipers are often found in much larger sizes than dial calipers. As with the dial caliper, Vernier calipers can be used to measure inside and outside diameters.

Indicators

Indicators are used to very precisely align parts for machining and test operations. Their operation is often similar to dial indicators.

Test Indicator

The test indicator measures smaller deflections than the dial gauge. Instead of a pin retracting into the mechanism, the deflection of an arm is measured. These are often used when aligning milling machine heads or otherwise indicating pieces for machining operations.

Haimer Indicator

Haimer gauges are used in CNC machining operations to find the edges of a part in all three dimensions. These zero points are used by the software to locate the stock in the machine’s workspace before the machining operation can begin.

Edge Finder

Edge finders are used on the mill to locate the edge of the part and align it with the spindle. Often edge finders are operated by sight or feel, but provide remarkably accurate and reproducible edge locations.

Gauges

There are a variety of different gauges that can be used to measure and reference different angles, lengths, hole sizes, and other part features. Often a good gauge set is one of the most valuable tools when reverse engineering or prototyping parts. We will discuss a few of the most common gauge sets available, but looking through a machinists' catalog is one of the best ways to learn about the wide variety of gauges and measurement tools available.

Angle gauge

Angle gauges are used to compare inside and outside angles found on parts to a fixed set of reference angles. Common angles such as 60, 30 and 45 degrees are essential. Large sets have many different angles to choose from.

Pin Gauge

Pin gauges are precisely ground rods used to determine the diameter of a hole or bore in a part. Pin gauges come in sets ranging from very small to very large, depending on the needs of the machine shop. For laboratory use, a gauge set of up to 0.5" is commonly used.

Thread Gauge

Thread gauges are sets of mating teeth for threads of different pitches and sizes. The thread gauge is most commonly used to determine what type and pitch of thread is on an unknown part, or part that was broken and needs to be remade without a drawing. Standard and metric sets are available and commonly used in the laboratory.

Feeler Gauge

The feeler gauge is a set of shims with a precise thickness joined together for easy storage. The feeler gauge is often used to set the distance between two surfaces such as electrical contacts in relays and point systems. It can also be used to determine the clearance for items like spark gaps and plate gaps.

Hole Gauge

Hole gauges are expanding measurement tools that are compressed and inserted into a hole, then locked at its diameter and used as a transfer measurement. While not as accurate as pin gauges, they are probably the more common way to measure diameters of holes.

Telescoping Gauge

The telescoping gauge is used to measure distances inside the cavity of a part, such as a large bore or slot opening. The gauge is compressed, then inserted into the cavity and expanded. It is swept through the cavity to attain the correct dimension and used as a transfer measurement.

Bore Gauge

A bore gauge is used to measure the diameter of a hole/bore larger than those measured by common whole gauges. Uses often include cylinder measurement in combustion engines and hydraulic systems.

Center Gauge

Center gauges are used to check the alignment of a workpiece in the lathe and to grind threading tools on a tool grinder. Due to their appearance, they are often called fishtail gauges and consist of a 60 degree inside and outside angle on a material similar to a steel rule.

Depth Gauge

Depth gauges are used to measure the depth of a hole. They consist of a larger plate with a ground surface that sits across the top of the hole, and rod that extends down into the whole. They can have their own scale or be used as a transfer measurement. The tail end of dial calipers is often used as a depth gauge.

Height Gauge

The height gauge is a precision mechanism with a Vernier, digital, or dial scale to measure how far above the base a part extends. Height gauges are often used to mark parts or measure the height of tools in tool holders to be entered into a CNC machine control program.

Dial Gauge

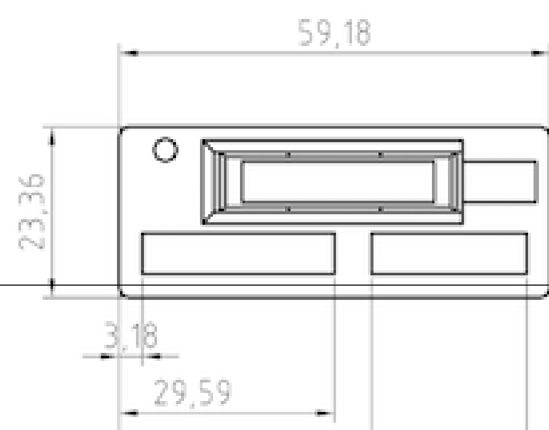
The dial gauge measures how much a pin is moved relative to the body of the gauge. These are often affixed in magnetic holders with flexible arms and used to ensure that round parts are centered in four jaw chucks and other machining operations. Dial gauges are also used to monitor relative movement of two parts during operations in the laboratory.

Reading Vernier Scales

The Vernier scale was invented in 1631 by a French mathematician. It is a secondary set of graduations on a measurement tool at a fixed fractional spacing of the main scale that allow a much more accurate reading of the instrument. Reading a Vernier scale takes some practice, but can be done very accurately. The Vernier is common in laboratory equipment and large machinists measuring tools, so being able to read it is an important skill for the experimentalist.

Mechanical

Mechanical design of parts and machines can be one of the most challenging aspects of apparatus design. Over designing results in wasted material and increased cost, under designing can result in dangerous failures that ruin your experiment and could cause serious harm. In this section we will learn the basic principles of sound mechanical design, how to draw your designs so that a machine shop can understand them, how modern 3D design work is done.



Contents:

Screws/Bolts

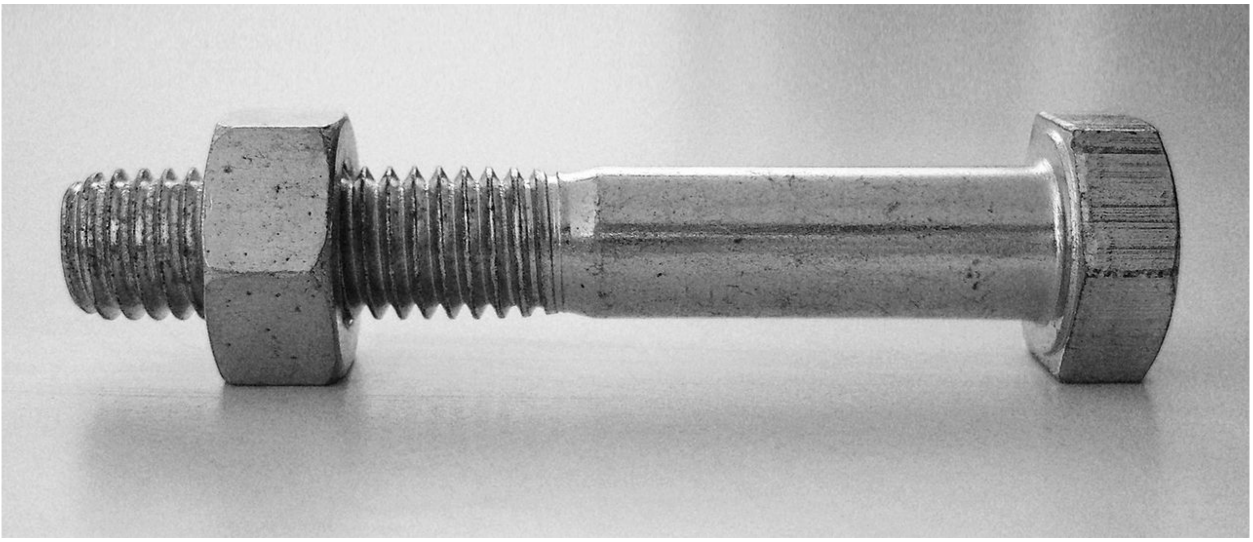


Fig. 1.42: Nut and Bolt (Image: Wikipedia)

Screws and bolts are used to hold together to parts by pressing them together and sometimes threading directly into one of the parts. The two terms are often used in an ambiguous way as the distinction is poorly defined. There is even a US Customs document "Distinguishing Bolts from Screws" Both are threaded fasteners, but distinctions are often made based on the size and use of the fastener. There are many types of screws listed on the Wikipedia screw page

Screws and bolts hold firmly against shear and tension on the parts. Some shoulder bolts or most long bolts have an unthreaded section to reduce the fretting of unthreaded sections of parts and/or allow a pivot point. Bolts are made in a variety of grades that represent different strengths. Markings on the head indicate the grade and can be looked up in a grade table

It is common to use screws and bolts with buts for parts with a clear-fit hole. Washers are used to distribute the load and reduce surface deformation. Locking washers prevent the assembly from loosening with vibration and use. Compounds such as Loctite are also often used to lock threaded fasteners.

Overview of Screws and Bolts:

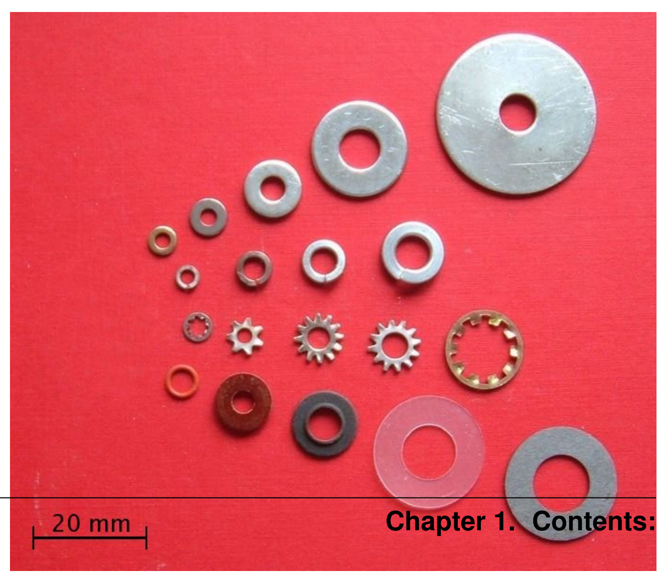


Fig. 1.43: Wahsers (Image: Wikipedia)

Nuts and Washers:

Thread Locking:

Threads

Threads are wrapped inclined planes that let us fasten parts together. There are many thread standards, the most common being the Unified Thread Standard coarse and fine variants (UNC and UNF). There are also metric thread standards and a smattering of speciality threads.

Threads are defined by their handedness (tightens in the right or left direction), their form (generally triangular), angle (generally sixty degrees), pitch (distance between thread highs), starts (where they can be engaged), diameters (major and minor), and lead (how much linear axial distance one turn of threads covers).

Consult a thread table or reference guide to learn the specifics of a given thread, but for laboratory work, we are generally working with UNC and UNF in the #2-#10 range, but metric is slowly becoming more common place in the US.

Threads can be cut on a lathe or CNC mill, but are most commonly cut in the laboratory with a tap for internal threads and a die for external threads.

Cutting internal threads:

Cutting external threads:

Unknown threads can be identified using a thread pitch gauge and calipers:

Rivets

Rivets are metal pins that come from the factory with a head on one end. Rivets are commonly found in aircraft, ship, and bridge construction. Rivets are commonly used in laboratory apparatus to make permanent joints between structural members or secure sheet metal to a frame. They are inserted through a hole in the two parts to be joined, and then the pin end is deformed or "bucked" until it forms a second head that permanently secures the fastener and parts. Rivets are best suited for shear load bearing applications and are fast and economical to install.

Hand setting of rivets:

Setting with a rivet gun:

A variety of rivets called pop or blind rivets can be used when only one side of the joint is accessible. They are hollow tube-type rivets with a shank running through their core. A special tool (air, electric, or hand) pulls on the shank after the rivet is inserted into the hole and deforms the back side of the rivet. The shank is then broken off and the joint is fast. Rivets can be easily removed by drilling off the head and punching the core out.

Using pop rivets:

Pins

Pins are used to hold two parts together, sometimes allowing there to be a pivot point, while other times holding the parts fast. There are several methods of pinning parts that we will cover here.

Spring Pins

Spring pins, also called roll pins, are self-retaining pins. They are under double shear since their middle section is experiencing shear forces in the opposite sense of those at the ends of the pins. Spring pins have a split along their length and are made to be driven into a hole slightly smaller than their outside diameter. The compression of the pin makes it secure it the hole with no other fasteners.

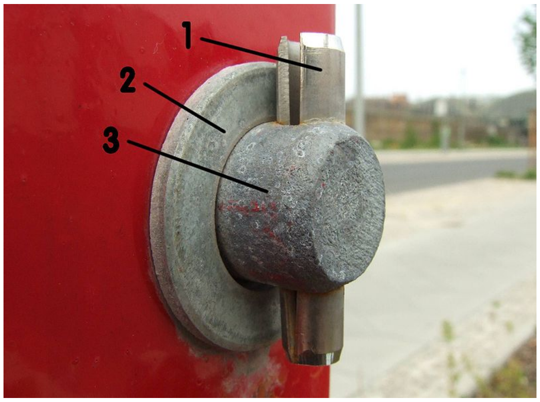


Fig. 1.44: A spring pin (Image: Wikipedia)

Dowel Pins

Dowel pins are primarily used to ensure proper alignment of parts as they are assembled when the loose alignment provided by bolts and screws in insufficient. The dowel pins can freely slide into a hole on a mating part or be slightly oversized so that they must be driven or pressed in. Such interference fits are a semi-permanent to permanent installation technique. There are a number of standards for dowel construction, including military, DIN 6325, and ISO 2338.

Taper Pins

Taper pins are very similar to dowel pins, except that they are slightly smaller on one end. Imperial sized pins generally have a 1:48 taper and metric pins a 1:50 taper. They are driven into place and are commonly used to hold interlocking shaft assemblies together.

Fig. 1.45: Steel dowel pins (Image: Wikipedia)

Split Pins

Split pins are slightly differently defined depending on your locale. In the US a split pin is a crimped wire that slides into a hole through a part (commonly a shaft) and over it like that shown in example C. What the rest of the world calls a split pin, the US knows as a cotter pin or cotter key. This fastener is placed through a hole and then its legs bent outward to secure it like that shown in example B.

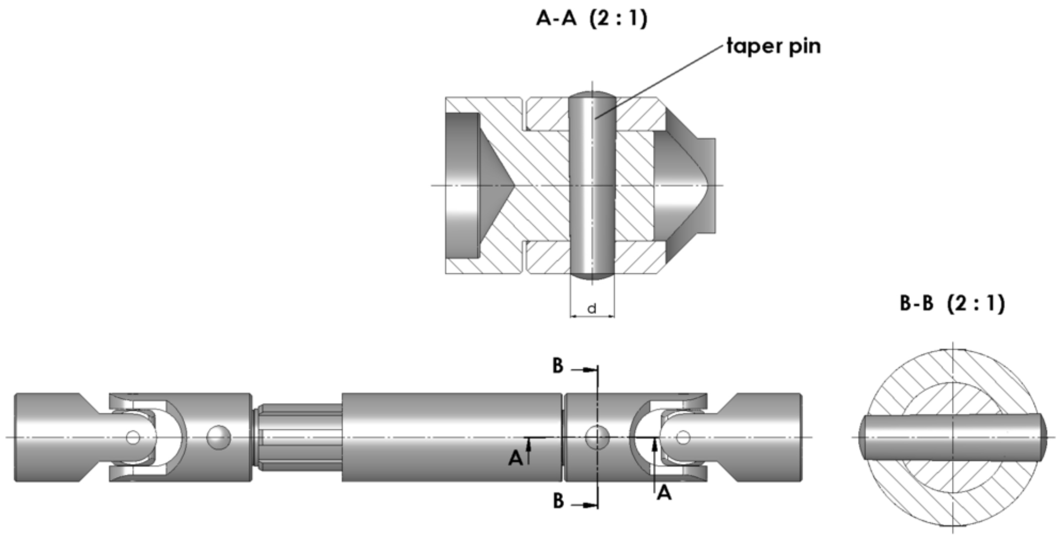


Fig. 1.46: Taper Pin (Image: Wikipedia)

Clevis Pins

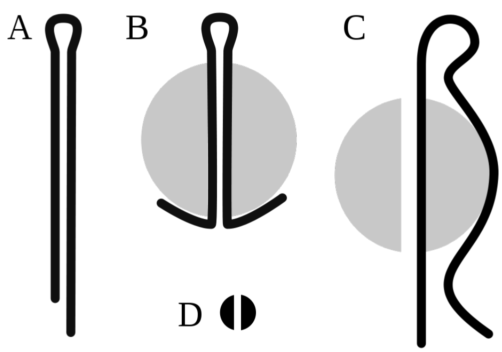


Fig. 1.47: Split Pins (Image: Wikipedia)

Clevis pins are used with a shackle and another securing pin (often a split pin) to provide lateral restraint, but allow rotation and movement in one direction. Clevis fastening systems are often found in rigging, aerospace, and other heavy duty control/actuation applications.

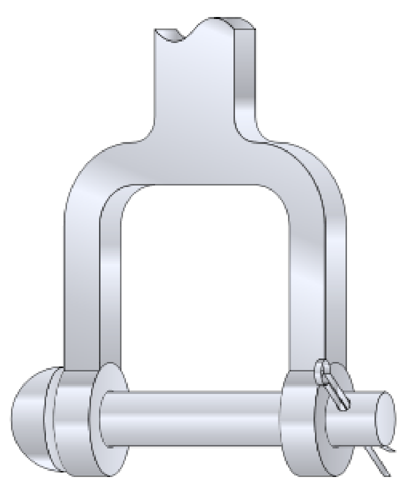


Fig. 1.48: Clevis fastener assembly (Image: Wikipedia)

Retaining Rings

Retaining rings are placed in grooves to hold parts onto a shaft or into another part such as a housing. There are a wide variety of speciality retaining rings for special applications, but the most common are axial and radial retaining rings. You may hear these called circlips or snap rings as well. There are special tools available to help with the installation of retaining rings.

Axial Rings

Fig. 1.49: Axial retaining rings (Image: Wikipedia)

These rings are installed into grooves on a shaft or in a housing by compressing or expanding them with special pliers and then slipping them over the end of the parts and releasing the pliers.

Radial Rings

Fig. 1.50: Radial retaining rings (Image: Wikipedia)

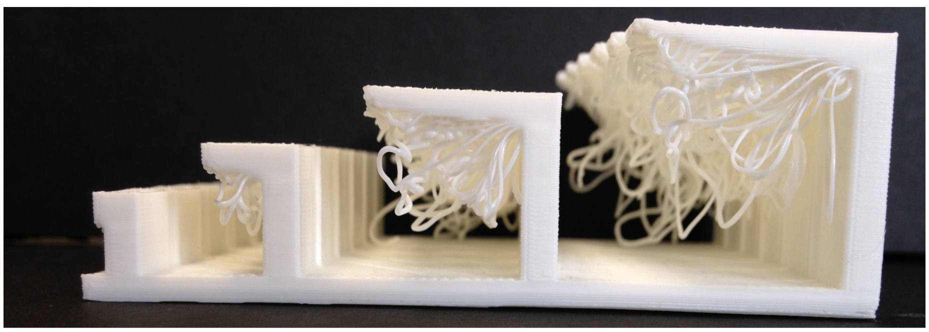
Radial rings are slipped over the side of a shaft into a machined groove.
There are special tools available for installation, but often a simple pair of locking pliers is used for installation and removal. A variant of these called "e-clips" is often found in assemblies as well.

Design for 3D Printing

When designing a piece of laboratory gear, there are many factors that have to be considered. This is even more true when the parts will be 3D printed as prototypes or as the final product. Due to the unique way that 3D printing produces parts, it imposes a special set of design constraints as well as a few design freedoms that may not be easily translated into a fully machined part. In this section we will go over some of these concerns and see how they are often addressed in the design and printing stages.

Overhangs

Any parts with overhang angles greater than 45 degrees from vertical should be supported or the print will most likely fail. External software generated supports may be added, or the designer may place temporary supports that are cleaned away after printing. It is also common to design permanent supports into the part that reduce any overhang angles to be less than 45 degrees. Using gussets and other features is a common way to accomplish this task. Printing a test piece is useful to get a feel for what your printer can do.



Failed Horizontal Overhangs

Fig. 1.51: (Image: utexas.edu)

Layer Stress

3-D printed models are the weakest along the boundary between layers. Any parts that are expected to experience stress should be oriented such that the maximum stress is perpendicular to the printing layers (commonly called the z-axis).

Wall Thickness

Too thick or too thin of a wall thickness on 3-D printed parts is a common failure mode and can increase the price of printing the part dramatically. Walls less than 1 mm thick are often weak, brittle, and print poorly. Walls thicker than several millimeters, are often unnecessary and are a waste of print time and material. Both of these increase the cost of the 3-D print significantly.

Bridges

Anytime the printer must print over open air is referred to as a bridging print. Bridging of more than a few centimeters often fails and results in sag of the print material. It is common for a 3-D print service bureau to be able to tell you what their maximum reliable bridging distance is. You can also do a bridging test print on your own 3-D printer to see what constraints you should follow.

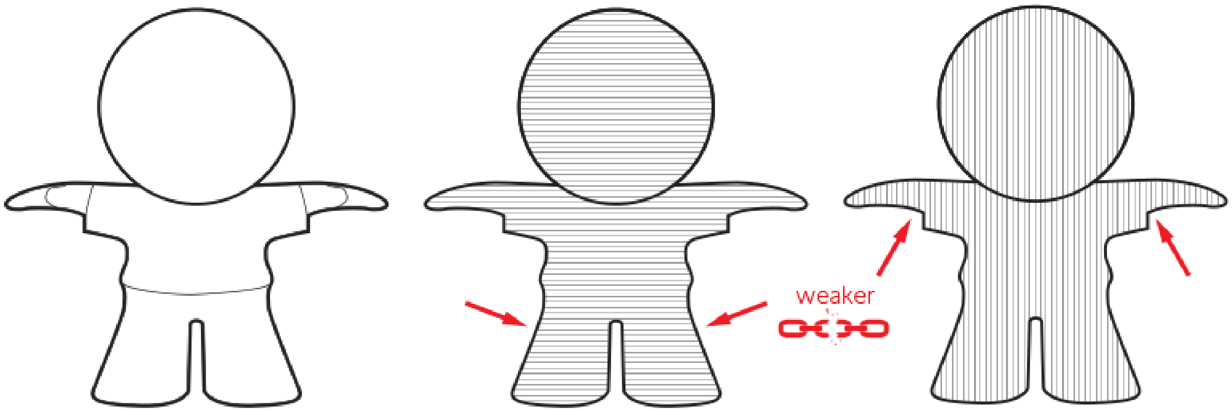


Fig. 1.52: Anisotropy of 3D print strength (Image: i.materialise.com)

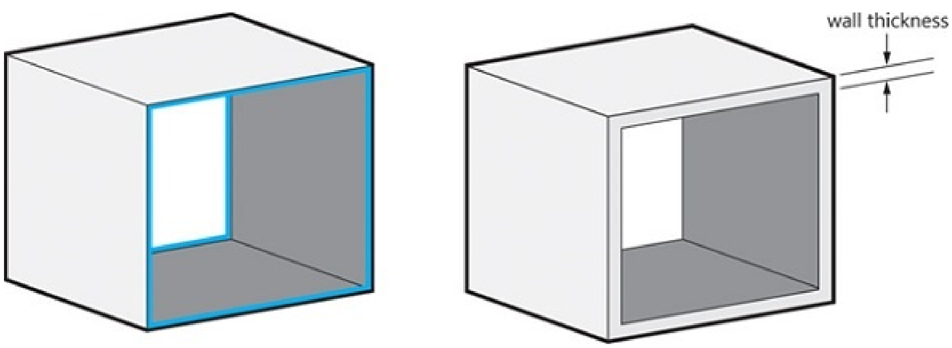


Fig. 1.53: 3D print wall thickness (Image: i.materialise.com)

Clockwise from top left
≤36mm – 0-0.5mm drooping, 36-60mm – 0.5-2mm drooping, ≥60mm – 2-5mm drooping

Fig. 1.54: ..‘(Image: utexas.edu) <https://innovationstation.utexas.edu/tip-design/>‘_

Part Clearance

When designing parts that are meant to fit together, some clearance must be allowed. To allow a free fit of parts, a clearance of 0.015-0.020” is often enough. When designing parts for a close fit, a tolerance of 0.005-0.010” is generally adequate.

Resolution

When exporting your model to a stereolithography file for printing, make sure an adequate resolution has been selected. Too low of a resolution will result in prints that are angular and do not represent the shape you desired. More complicated models will take more time to slice and process, but will be a better representation of your 3-D model.

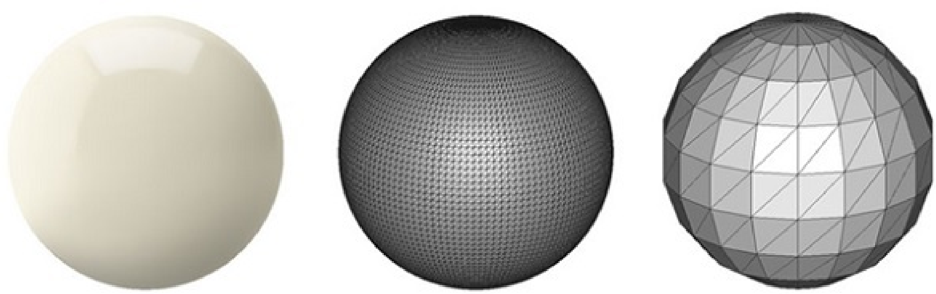


Fig. 1.55: Different file resolutions (Image: i.materialise.com)

Text

Adding text to a 3-D print is a great way to label front panels, personalize an object, or make the user experience more intuitive. Text can either be engraved with letters recessed into the part, or embossed with raised lettering. Using engraved text is highly recommended because embossed text often has features that are too small at reasonable font sizes and result in failed prints. Engraved text also doesn’t leave material sticking up from the top of the part free to catch things and possibly be sheared off.

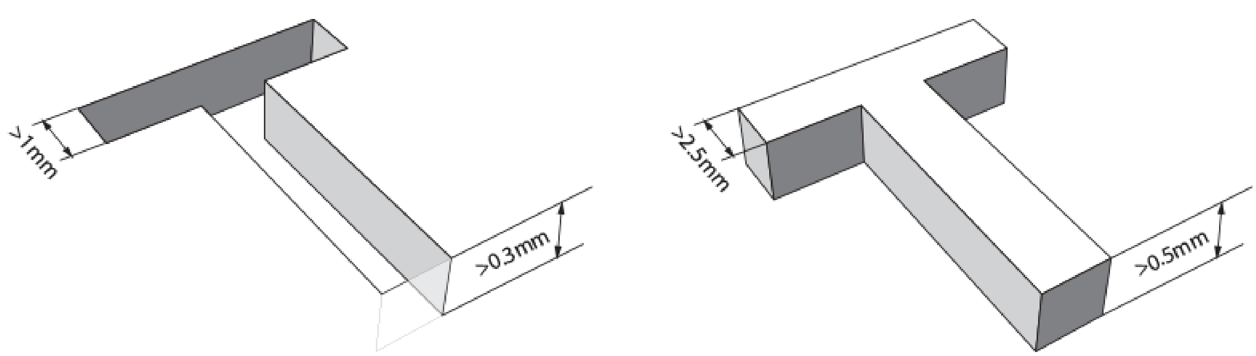


Fig. 1.56: Text engraved (left) and embossed (right) (Image: i.materialise.com)

Dimensional Accuracy

Every 3-D printer has an accuracy specification in XY and generally a separate specification in the Z. You should consider these accuracies and the thermal contraction of the plastic if your part requires precise dimensional accuracy. There are several empirical formulas that have been developed for holes oriented horizontally or vertically in 3-D printed parts. It is easy enough to print a calibration print on your printer and create your own equation for your printer. Designing parts with enough tolerance in mind is generally possible though as 3-D printed assemblies are not specified to nearly as tight tolerance as machined parts. Also consider printing the part in a different orientation to get the highest dimensional accuracy where required. You can even split up the part into multiple pieces and assemble them after printing.

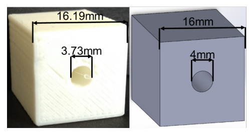


Fig. 1.57: Dimensional output of the printer (left) compared to the solid model (right) (Image: utexas.edu)

Model Verification

Your 3-D printed model could have errors when exported depending on your CAD package. Is important to be sure that the model is watertight and that no remaining internal geometry is left from doing Boolean operations on different shapes. Going through the sliced version of the model and looking for any incorrect internal geometries is highly recommended. Checking the generated mesh is easily accomplished with tools such as Netfabb or other printing and slicing software.

Internal Features

3-D printing allows you to design internal features and passageways that can be completely hidden from the outside, have bends internally, or embed other complex structures into the part. If your part will be machined by traditional processes at a later time, it is important to consider that these operations may not be easily achievable. Always design with the process that will generate the final part in mind.

Thin Walled Pressure Vessels

A thin walled pressure vessel is generally considered to be one whose walls are less than about 1/10 or 1/20 of the radius of the vessel. The formulas provided below are for reference and calculation, but before constructing a real pressure vessel you should check with an engineer. The information is provided for your reference. Remember that a catastrophic failure of a vessel could result in serious injury or death.

Longitudinal Stress

$\sigma_l = \frac{pd}{4t}$

Circumferential (Hoop) Stress

$\sigma_h = \frac{pd}{2t}$

Nomenclature

- σ_h = Hoop Stress
- σ_l = Longitudinal Stress
- p = Internal pressure
- d = Internal diameter of the tube
- t = Wall thickness

Thick Walled Pressure Vessels

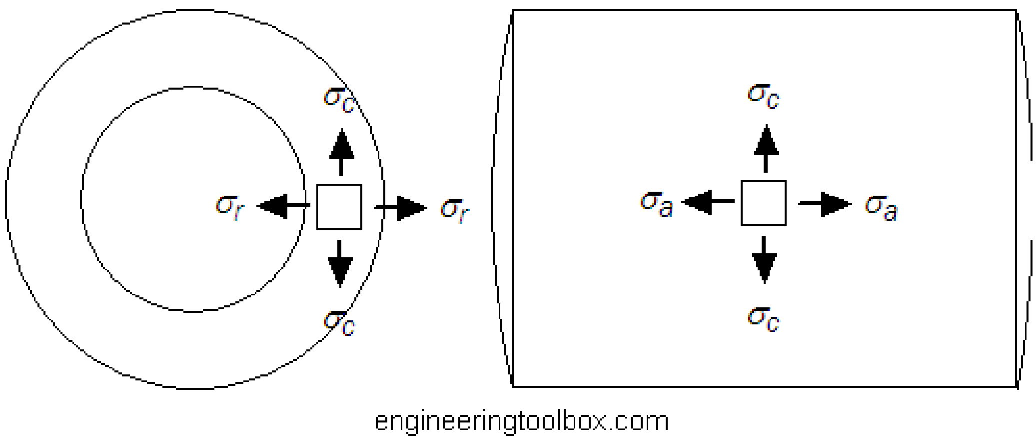


Fig. 1.58: Image: Engineering Toolbox.

A thick walled pressure vessel is generally considered to be one whose walls are greater than about 1/10 or 1/20 of the radius of the vessel. The formulas provided below are for reference and calculation, but before constructing a real pressure vessel you should check with an engineer. The information is provided for your reference. Remember that a catastrophic failure of a vessel could result in serious injury or death.

Axial Stress

$$\sigma_a = \frac{p_i r_i^2 - p_o r_o^2}{r_o^2 - r_i^2}$$

Circumferential (Hoop) Stress

$$\sigma_c = \frac{p_i r_i^2 - p_o r_o^2}{r_o^2 - r_i^2} - \frac{r_i^2 r_o^2 (p_o - p_i)}{r^2 (r_o^2 - r_i^2)}$$

Radial Stress

$$\sigma_r = \frac{p_i r_i^2 - p_o r_o^2}{r_o^2 - r_i^2} + \frac{r_i^2 r_o^2 (p_o - P_i)}{r^2 (r_o^2 - r_i^2)}$$

Nomenclature

- σ_a = Axial Stress
- σ_c = Circumgerential (Hoop) Stress
- σ_r = Radial Stress
- p_i = Internal Pressure
- p_o = External Pressure
- r_i = Internal Radius
- r_o = External Radius
- r = Radius to a point in the wall

Bending Forces

This page shows the way to calculate the bending force and displacement on a circular plate that is simply supported and loaded uniformly. This is a useful calculation to perform for end platens on internal pressure vessels.

Fig. 1.59: Image: Efunda.

Plate Rigidity

$D = \frac{Et^3}{12(1-\nu^2)}$

Displacement

$y_c = \frac{-qa^4}{64D}\frac{5+\nu}{1+\nu}$

Stress

$M_{\max} = \frac{qa^2}{16}(3+\nu)$

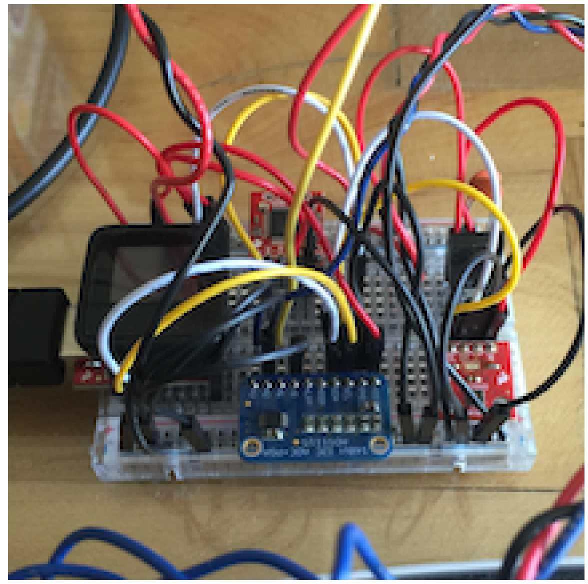
$\sigma_{\max} = M_{\max}\frac{6}{t^2}$

Nomenclature

- D = Plate rigidity
- q = Load/area
- a = Radius of plate
- t = Thickness
- E = Young's modulus
- ν = Poisson's ratio
- y = Displacement
- M = Moment
- σ = Stress

Electronics

Nearly every modern piece of laboratory equipment involves the use of some kind of electronics. In this section we will cover some of the basics that you need to be able to analyze, troubleshoot, and design electronic circuits.



Contents:

Getting Started with Arduino

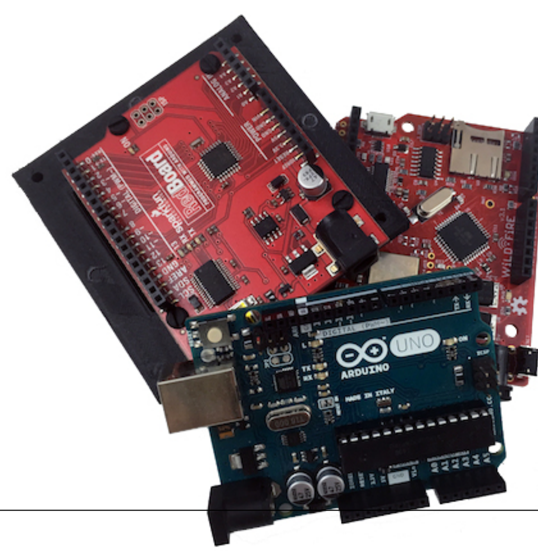
We'll be using the Arduino platform to explore basic electronics and microcontrollers. In this lesson we will learn the basics parts of the Arduino.

Arduino Basics

The Arduino is a small microcontroller that allows you to write firmware to read sensors, turn things on and off, and make decisions based on those inputs/outputs. You can think of the Arduino as a small computer, except it has no operating system. We write some code, translate it to instructions the Arduino can understand (compiling), and then we upload those instructions to the memory on the Arduino. While all of the details behind each of those steps are very interesting, they are a little beyond what we want to accomplish right now.

The Arduino Uno itself is based around an Atmel AVR microcontroller called the ATmega328P. There is actually no official designation of what AVR stands for, but it is generally thought to stand for **A**lf (Egil Bogen) and **V**egard (Wollan)'s **R**ISC processor. This is an 8-bit processor based on a modified Harvard Architecture and RISC design. Knowing what all of those words mean is not essential to getting started, but worth reading about for those of you that are curious. Arduino is a brand of products designed with the idea that non-engineers should be able to create projects that use digital electronics. The history of the company starts officially in 2005, with many changes, conflicts, and all of the normal things experienced by young technology companies. Arduino was quickly adopted by everyone from artists that wanted to make colorful and interactive displays to citizen scientists that wanted to measure their world. Now you see why we are using it for this course.

Learning how to program microcontrollers can be a daunting task. The datasheet for this very simple processor is a whopping 444 pages, but is the ultimate source of information. For our projects, we will be using the Arduino programming environment that makes the native functions of the processor much easier to use for beginners. It does take a bit of a mental shift if you are used to programming on personal computers, the Arduino has only 32kB of program memory space! Working in such resource constrained environments offers many opportunities to make low power systems, but does require some getting used to.



Take your Arduino out of the packaging and examine it closely. Identifying all of the parts is not too important right now, but let's take a tour of the capability of the development board and see where we will connect different peripherals.

Digital Input/Output

Sometimes called General Purpose Input/Output (GPIO) pins, there are 14 of these on the Uno. They can serve as an output providing a digital on (5V) or off (0V) to manipulate attached devices (think of it like a computer controlled light switch), or as an input that can read if the input is on or off. A few of these pins have special functions that you may need to be conscious of when designing a circuit around them:

- Pins 0-1 are for serial receive and transmit respectively. These are connected to a chip that interfaces your Arduino to the USB port and your computer. It's how you program the Arduino and read back data on the serial terminal. Connecting things to these can keep the Arduino from programming properly. Stay away from hooking to these until you're out of pins to use. Then you'll have to unhook them to upload new firmware.
- Pins 2-3 are capable of generating interrupts. This means they can watch their state, and create an event in your code when the state meets certain criteria. For example, you could use an interrupt based on if a pin value changes to count how many times a tipping bucket rain gauge has tipped, since each tip will cause the pin to change state.
- Pins 3,5,6,9,10,11 can generate a Pulse Width Modulated (PWM) waveform. This is used to create an analog voltage output, meaning a voltage that can be anywhere in the voltage range of the processor (0-5V). The PWM modules on the Arduino are 8-bit meaning there are 256 levels of possible votlage output.
- Pins 10-13 can interface to sensors that communicate using the Serial Peripheral Interface (SPI). This is a digital communication protocol often used to connect sensors such as pressure, temperature, and humidity to the microcontroller.
- Pin 13 has an attached light emitting diode (LED) on the circuit board. This is handy for troubleshooting as it requires no external parts.
- SCL/SDA are special purpose pins that communicate with sensors on the Inter-Integrated Circuit protocol (I2C). I2C only requires 2 data wires and is often used for sensors such as pressure, temperature, and humidity.

Analog Input

Analog inputs are used to measure a voltage. The Uno has 6 of them labeled A0-A5. The hardware that does the work of turning a voltage into a digital representation is called an analog-to-digital converter (ADC). In this case the ADC is built into the ATmega328P and has 10-bits of resolution. This means the voltage will be digitized to one of 1024 different values. By default the input range is 0-5 VDC, but that can be changed by providing a reference voltage to the AREF pin.

Power

The Uno has several power "rails" exposed. These can be used to power sensors and LEDs. If your project has significant power requirements, they may not be able to provide enough power, but they will provide plenty of power for the experiments we will be performing.

- Vin connects to the input voltage you provide if using the external power jack. You can also power the Arduino through this pin (7-12 VDC).
- GND connects to the system ground.
- 5V provides regulated 5 VDC power.
- 3.3V provides regulated 3.3 VDC power. Maximum current is 50 mA, so be careful!

- IOREF provides the operating voltage reference for inputs and outputs. This is useful when designing circuits to work with Arduinos of different operating voltages (5 or 3.3VDC).

Other Technical Specifications

Specification	Value
Recommended Input Voltage	7-12 VDC
DC current per I/O pin	20 mA
DC current for 3.3V power pin	50 mA
Flash memory	32 kB
SRAM	2 kB
EEPROM	1 kB
Clock Speed	16 MHz

Prototyping Techniques

When producing electronics for the laboratory, we often need to only make a few copies of our circuit, making manufacturing on a commercial scale unreasonable. There are a number of prototyping techniques that can be used to create circuits that are semi-permanent or permanent. We will cover the basics of each technique and its unique advantages/disadvantages.

Breadboard

The breadboard is the most versatile way to prototype electronics. A perforated plastic grid covers copper spring contacts. Components and wires can be pushed into the holes and the spring contacts make a good, but temporary electrical connection. Breadboards commonly contain power busses for convenience. Generally the vertical rows on each side of the center are electrically connected. Breadboards are useful for the initial prototyping of circuits when the design is being tested and iterated on often. This type of construction is also useful for temporary or demonstration circuit construction.

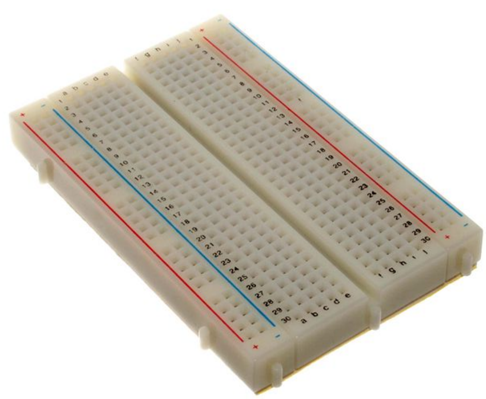


Fig. 1.60: A small breadboard (Image: Wikipedia)

There is some stray capacitance introduced into your circuit when using a breadboard. Long component legs and any flying jumper wires can be places for interference to be radiated/received. Breadboards are not suitable for radio frequency (RF) prototyping, but are often adequate for laboratory circuits. There are products on the market that are placed on top of the breadboard before the circuit is built and allow you to later remove the components and solder them to a permanent board. This is suitable for basic circuits with a careful initial circuit layout.

The video below demonstrates the capacitance effect of a breadboard:

Perf/Vero Board

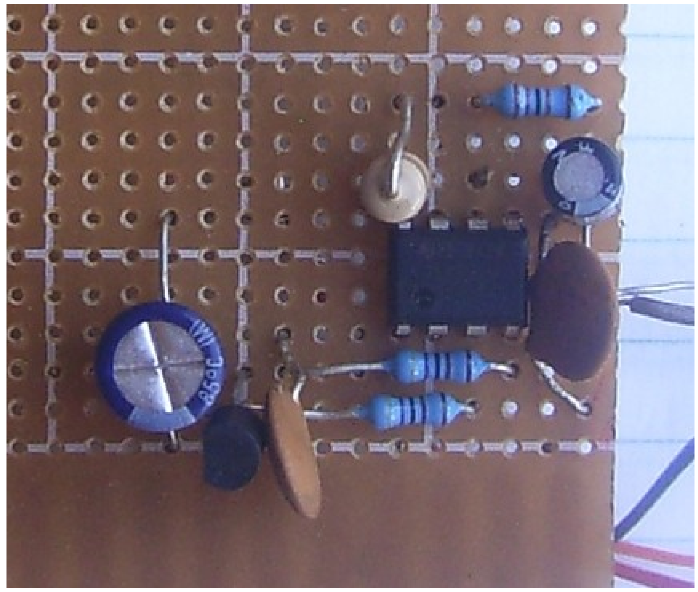


Fig. 1.61: A simple circuit constructed on perfboard (Image: Wikipedia)

Perf-board or vero-board can be used to create more permanent versions of circuits. These boards are made of an insulating material with a grid of holes drilled in it. Some designs have no copper pads or electrical connections, but many have pads to allow easy soldering and mechanical securing of the components. Different designs are available with connection patterns between rows and columns of holes to aid circuit construction with fewer jumpers.

This prototyping technique has many of the same issues as breadboard prototyping, but is a more permanent construction technique and one of the most popular. Some technicians prefer to use a tool to cut copper traces on the bottom of the boards to further increase the number of connections that can be made without jumpers. Layout of these prototypes requires some planning, but can make very nice low to mid-frequency circuits.

Manhattan

Manhattan prototyping is commonly used for high frequency and RF circuit prototyping and construction. A copper clad board is used as a ground plane and islands are cut or glued on to provide common connection points for components. Components are often wired in the air even. Keeping leads short is important! This construction can look messy, but when done well is an art. There are products available (like this kit) that provide glue on solder pads of various size and footprint, but often scrap bits of copper clad are used. Another way to create connection islands is with a hollow drill, such as this one.

This technique is named for the skyscraper/skyline look of the components standing up off the board. When integrated circuits are used, they are often glued to the board upside down with their pins facing upward. This is referred to as dead bug construction. This technique is very good at high frequencies, but can produce a slightly fragile circuit, so an enclosure is recommended.

Wire-Wrap

Wire-wrap uses special wire and a tool to strip, and firmly mechanically bond the wire to a square post with sharp corners. Wire wrapping is often used to prototype digital circuits in a semi-permanent way. Modifying wire wrap boards can become time consuming when many wires are connected to a single post or there are many overlapping wires. Special sockets for parts are required for this technique, but it can be a fast way to prototype and was often used in the design of early computers.

Wire wrapped designs are good for relatively high speed circuits, but not useable for modern computing speeds or often for low noise analog designs.

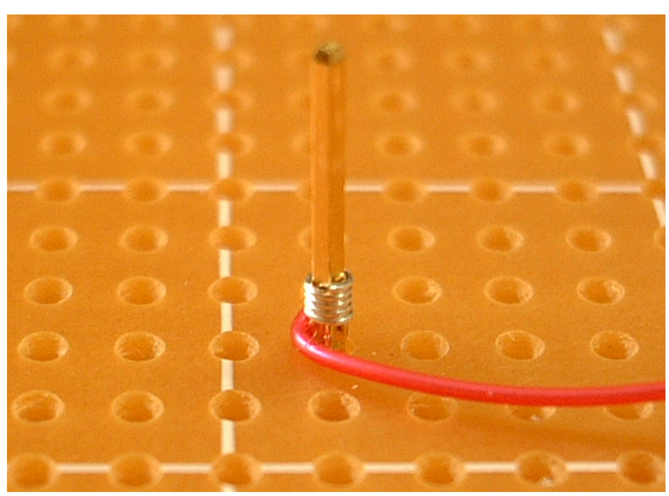


Fig. 1.62: A single wire-wrap connection (Image: Wikipedia)

Quick-turn PCB

In the last few years it has become economical to design a custom printed circuit board (PCB) and have it manufactured. Traditionally, you needed to produce a large number of boards to work with a manufacturer and the designs were made with cut tape patterns, called "taping out" a board. Now, computer design tools make laying out a board a relatively simple process. PCBs can be created with copper connections on a single side, on both sides, or even with many layers of copper sandwiched in between layers of insulators. Signals are passed between layers with holes that are copper plated called vias.

PCBs, when designed correctly, can be used for very high frequency and low noise designs. Laying out a circuit board takes considerable time, so that should be considered before setting out in this direction. The price with panel-sharing services such as OSH Park can be as low as $5/sq-in for a double sided board. Designs are generally made in a few days, but overnight manufacturing can often be arranged for a significant fee. Be sure to check the design rules for your manufacturer and set your CAD tool to enforce them to ensure a working board!

Examples

This video by Alan Wolke shows many excellent examples of the various electronics prototyping techniques and is packed full of great resources.

Soldering

Most of the semi-permanent to permanent prototyping techniques use soldering to make secure mechanical and electrical connections. Soldering uses a hot iron to heat the two wires/leads/pads and then flows a wire with a low melting point into the joint. Learning how to make clean and strong joints is essential to making reliable circuits. In this section we will cover the basic tools that are used in the lab and learn several basic soldering techniques.

Tools

- **Soldering Iron** - A good soldering iron must have an adjustable temperature control. Irons that just plug into the wall are nicknamed "fire starters" and should never be used - they are a good way to burn off PCB pads or wait for long times to heat large joints. Sticking with a reputable brand (Hakko, JRC) is recommended. You should purchase a large number of the iron tips for replacement and for different tasks. A needle point tip is often used by beginners, but is **NOT** an effective tip for most applications. A narrow chisel-like tip is recommended.
- **Solder** - Available in leaded and lead-free varieties, solder is a low melting point alloy that makes a solid mechanical and electrical connection. A multi-core rosin solder is recommended. Beginners usually find it useful to learn with leaded solder as it flows better, but be sure to wash your hands and reduce the chances of any lead contamination. For most electronics work a fine gauge solder is easiest to work with. Larger diameter solders are used with large power connections and large joints.
- **Flux** - Flux cleans the joint and allows you to make a good solder joint. There are many types of flux and many arguments for using each. For a beginner a good liquid no-clean flux in a flux pen is a good choice. There are thicker flux pastes that are also preferred by some as they can help adhere small components to the circuit board. While many solders have enough flux for new joints, it is essential to add flux for any kind of rework of a solder joint.
- **Tweezers/Pliers** - When soldering you often need to bend component leads and twist connections together. Small tweezers (curved and straight) are recommended, along with small flat nose pliers. These are all available from any electronics equipment provider. Wire strippers and diagonal cutters are also useful to keep on the bench. These tools should be dedicated to electronics and not used anywhere else in the lab!

Fig. 1.63: Hakko temperature controlled soldering iron (Image: Sparkfun)

Fig. 1.64: Solder (Image: Sparkfun)

Fig. 1.65: ESD Safe Tweezers (Image: Sparkfun)

- **Solder Braid** - When removing solder from a joint, solder braid can be used to wick the solder away from the joint. It is a cheap and effective desoldering method.

Fig. 1.66: Solder Braid (Image: Sparkfun)

- **Solder Vacuum** - When removing many through-hole components, it is useful to purchase a solder vacuum. These can be a separate tool used to suck up melted solder or a dedicated desoldering iron with a hollow tip and built in vacuum pump. Desoldering with these tools can be very fast.

Fig. 1.67: Solder Vacuum (Image: Sparkfun)

- **Third-Hand/Board Holder** - When soldering it is necessary that the circuit be firmly fixed and not moving around. Using a “third hand” device or other board holder/fixture can make many tasks much simpler and less frustrating. There are a range of products on the market including clip based tools, board vices, and others. A small vice, such as the Panavise Jr. and flexible hand like the SparkFun Third Hand are nice to have on the workbench.
- **Flush Cutters** - Use these to trim component leads off close to a PCB. Never use the flush cutters on any hard material as it will dent the jaws and ruin the tool. Only cut component leads and small gauge wire.

Fig. 1.68: Third Hand (Image: Sparkfun)

Fig. 1.69: Flush Cutters (Image: Sparkfun)

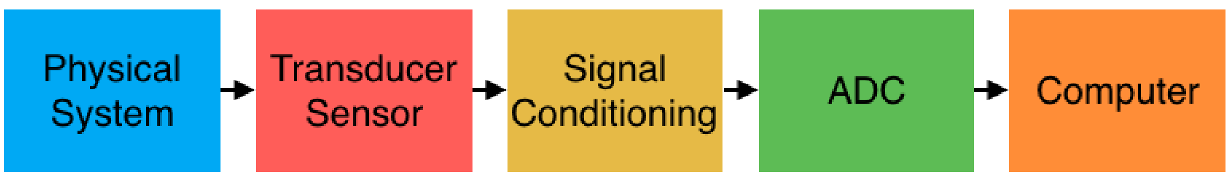
Technique

Soldering technique is difficult to describe with simple diagrams, but Dave Jones has made an excellent three part soldering tutorial that is well worth your time to watch. It is helpful to practice these techniques on some junk boards or components before working on your project.

Safety Precautions

Soldering involves a very hot iron and potentially dripping or flying hot metal and flying wire bits from trimming. Safety glasses should be worn at all times. Being aware of burn care techniques is a good idea incase you do accidentally burn yourself. Make sure the soldering iron is off when not in use as it is a potential fire hazard and dramatically decreases the life of the iron tip. Some find it helpful to use a timed outlet to make sure the iron is off after a fixed amount of time.

Data Acquisition



In most apparatus we need to record something about the physical system in a format to analyze later, commonly digital data. In this section we will examine how the data acquisition chain works and discuss each part and its role in making sure you get the best quality and most valid data. Doing data acquisition incorrectly can result in poor quality data, or data that is incorrectly sampled, often leading to mis-interpretation.

Physical System

The physical system is the actual thing we care about measuring and recording. It generally does not lend itself to direct recording. For example measuring the displacement of an object or the mass of an object is not natively a digital process. The most important thing when considering the physical system is considering how you can attach instrumentation without modifying the behavior of the system. It is also important to ensure that the system is functioning correctly and safely.

Transducer/Sensor

The transducer or sensor is the device that will turn the physical quantity (temperature, acceleration, mass, etc) into an electrical signal. Transducers transform one quantity into another through a well defined transfer function or calibration. Transducers often use the electrical quantities of voltage, resistance, or current to encode information about the physical system. We will go over many transducers in detail later in the course.

Signal Conditioning

Most transducers do not produce a signal that is immediately ready to be recorded. The act of modifying the signal to make it useful and recordable is known as signal conditioning. Common steps in signal conditioning include conversion to an easy to measure quantity (i.e. resistance to voltage), amplification (increasing the amplitude or output range of the signal), and filtering (cleaning up the signal). Each of these steps has a devoted section that we will discuss during the course.

Analog-to-Digital Conversion

Once the signal is in an easy to measure quantity (most commonly voltage) and in the ideal range for your hardware, it is time to convert the information to a digital quantity that we can store and analyze later. This process is known as analog-to-digital conversion (A2D or ADC). The process of converting an continuous quantity (an analog voltage that can have any value) to a digital representation that much have a discrete number of steps of values, results in a loss of some information. The resolution of the ADC is generally described in the number of bits used to represent the analog quantity, with higher bit ADCs having a finer resolution. The time resolution required should also be considered. Sampling must be a minimum of twice the maximum frequency of interest (the Nyquist criteria). In practice it is wise to always sample at least 5-10 times the maximum frequency of interest to avoid any aliasing issues and make filtering/decimation an easier process.

Computer/Recording

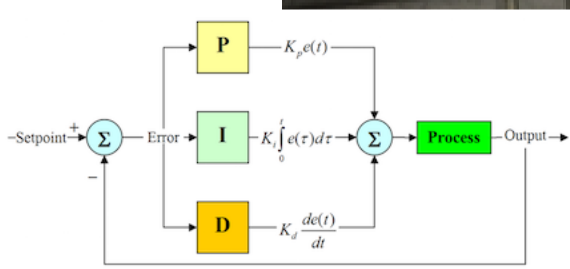
The ADC system is generally connected to a computer (embedded or otherwise) to store the data to a storage media. Often the incoming readings are visualized in real-time to help the equipment operator and/or used to control the system being monitored.

Transducers

Transducers are how we sense the physical world and turn those measurements into a quantity we can easily measure and record, such as an electrical voltage. In this section we will demonstrate a variety of transducers to measure strain, displacement, force, temperature, and other common physical quantities. We will also show how calibrations are performed so that the measurements can be converted to the physical units that we care about. There are many pitfalls and strengths associated with transducer we will discuss, learning these now can prevent hours of frustration, wasted funds, and incorrect results.

Contents:

Control Systems



Control systems are one of the most important parts of any apparatus. They take sensor readings and target values, then use actuators to try to drive the system towards the desired state. Control theory can become complex very quickly for systems with non-linear response functions or systems with strict response time requirements. In this section we will cover the fundamentals of control by examining the most common types of control systems. (Icon image: Wikipedia)

Contents:

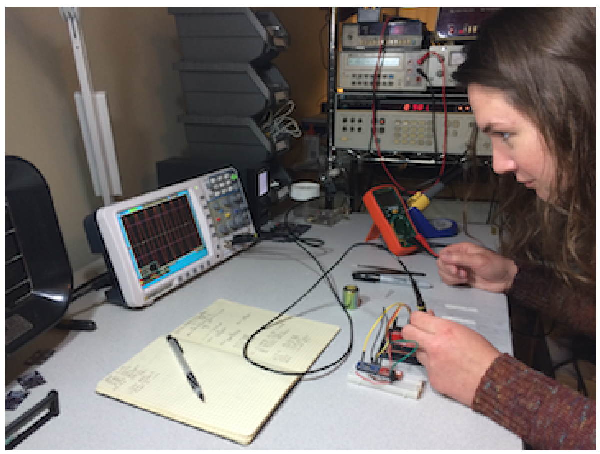
Lab Exercises

The only way to learn about building things is to build things! Follow along with these laboratory exercises to help you practice the new information you've learned along the way and build something useful.

Contents:

Blinky 1.0

This activity will get you familiar with the Arduino programming environment and show you how to blink an onboard LED with a simple program that we will write, compile, and upload.



Materials

- Arduino UNO
- USB Cable
- Computer (Mac, Linux, Windows)

Installing the Arduino IDE

We will use the Arduino interactive development environment (IDE) to write our code, compile, and upload it to the Arduino. This is a free tool that is actively maintained by the open-source community. It is available for Mac, Windows, and Linux machines from the Arduino Software page. You'll also need the drivers for the FTDI USB to Serial converter, available from the FTDI download page .

Download and install the software and FTDI drivers as you would any other application on your machine. Reboot after installation. Each project will need to be stored in a directory (folder) named the same thing as the project. It is recommended to store all of these projects in the "Arduino" directory, created in your documents folder after installing the IDE.

Your First Program

In the software world, we often write a "Hello, World!" program. In the hardware world, we often make "blinky", an application that blinks an LED. While that may not seem very exciting, remember that that LED could represent a pump, light, fan, linear actuator, or any number of other actuators you can control.

- Open the Arduino IDE and create a new sketch by selecting File > New File, if one was not created when you opened the application.

With the new file open, you will notice it is not empty. The skeleton of an Arduino has already been populated for you.

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

The skeleton has two functions, setup and loop. The setup function will run once when the Arduino is powered on or reset, then the loop function will run. Once the loop function reaches the end of its instructions, it starts again from the top. This function will run indefinitely for as along as the microcontroller has power. We will dive more into what functions mean later, but for now it is time to get something blinking!

Add the following code to your empty project and save the project as `my_first_blinky`.

```
void setup() {
  // initialize digital pin 13 as an output
  // remember pin 13 already has an LED connected!
  pinMode(13, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13, HIGH);   // turn the LED on
  delay(500);               // wait for 500 ms
  digitalWrite(13, LOW);    // turn the LED off
  delay(500);               // wait for 500 ms
}
```

Plug your Arduino into your computer. In the Tools > Board menu select Arduino/Genuino Uno. Then select the appropriate serial port from the Tools > Port menu. If you don't know which port to select, try unplugging the board, then note what options are in the menu. Plug the board back in and the new option in the menu will be the port the Arduino is on.

Next, press the arrow shaped "Upload" button at the top of the Arduino IDE. Your program will be compiled and uploaded to the Arduino in a few seconds. If you look at the Arduino, you should see an LED blinking. Congratulations! You did it!

This is a pretty simple program that is well commented, but we will quickly go over how it works. The first thing that happens is the `setup()` function runs. Everything with `//` before it is a comment and there for human readability only, the compiler throws all of that away before uploading the program to the Arduino. The only instruction is `pinMode(13, OUTPUT);`. Every statement in C (yes, you're really writing C/C++) needs to be completed with a semicolon. This particular command tells the microprocessor that we are going to be using pin 13 as an output. The pin is configured and will hold its status as an output until we change it. The command `digitalWrite` is used to change the output state of a digital pin. The syntax uses the pin number we want to modify and the state (high or low) that we wish to assign it. The `delay` command pauses the execution of the program for the given number of milliseconds (1 ms = 1/1000 second). In our case we turn the LED on by making pin 13 high, then waiting 0.5 seconds, then we turn the LED off by making the pin low, waiting another 0.5 seconds, then the whole thing repeats!

There you have it, your Arduino is setup and you have successfully controlled a small part of the world. You can keep exploring by checking out the examples in the File > Examples menu. Examples are one of the best ways to learn, and the included examples are thoroughly commented. Go through a few, even if you just read them, to get used to reading the code.

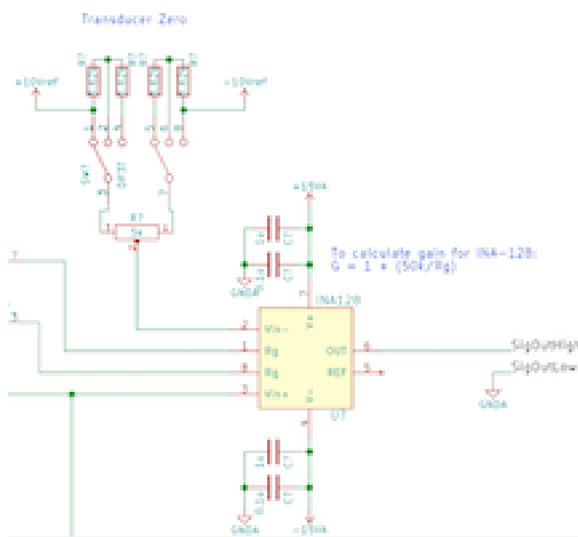
Grading Rubric

Description	Max Points
Activity Completed	10
Total	10

Project Brainstorm

The course project should help you solve a research problem you are facing while helping you actively learn many of the skills introduced in this course. The best way to learn is through trial and error - there is sure to be a lot of both during your project. We would like to help you define a project that is within the scope of the course and the semester time limit on the work. A good project will have several characteristics:

- Uses multiple disciplines we discuss (i.e. mechanical, electrical, software, etc.)
- Solves a problem with no commercial solution or no economical commercial solution.
- Will not cost thousands of dollars to build. Plan on \$50-\$100.
- Does not put you or others in the path of potential harm or danger.



Brainstorm 5 potential project ideas. Think about the problem and potential ways to approach it. Sketching ideas may help your brainstorming. Turn your project ideas in on paper or via email. Setup a meeting with the instructors to discuss your ideas.

Grading Rubric

Description	Max Points
Completed on time	5
Presented 5 ideas	25
Projects within guidelines	5
Total	35

Part Manufacturing Techniques

We've discussed the many ways that parts and assemblies are manufactured. Get a few different parts and objects from your everyday environment. Describe and discuss how they are made. Look for clues

like the surface finish of the part, what material it is made of, and where mechanical movement and wear happens.

Questions

- What techniques were used to create the part? Why?
- What fasteners were chosen? Why?
- What is the part made of? Why?

Grading Rubric

Description	Max Points
Participation in class	5
Total	5

Threading Activity

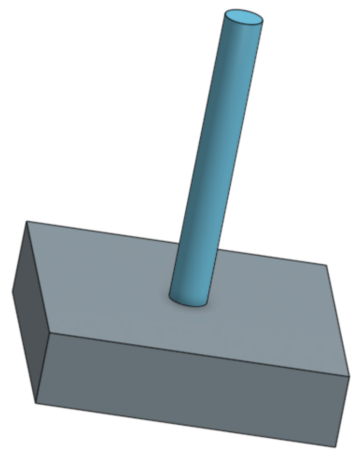
In this activity you will practice drilling and tapping, and well as threading with a die. Cutting threads is a technique used over and over again to provide a way to secure parts together in a firm, but removable way. Be sure to follow proper safety and personal protective equipment procedures!

Materials

- 6061 Aluminum plate (2”x2”x0.5” cut from McMaster 8975K74)
- 6061 Aluminum rod (0.25”x2.5” cut from McMaster 8974K22)
- 1/4”-20 Through-Hole Tap (McMaster 26955A43)
- 1/4”-20 Right Handed Die (McMaster 2576A451)
- #7 Twist Drill (McMaster 29045A727)
- #4 Centerdrill (McMaster 2915A14)
- Tap driver (McMaster 2546A23)
- Die handle (McMaster 25565A21)
- Deburring tool (McMaster 4289A35)

- Cutting lubricant (Crisco, Tap Magic, etc.)

Procedure



1. Layout and mark the center of the plate for drilling (with a rule or layout dye and calipers).
2. Using the center drill, make a pilot mark for the final drill. Do not drill through the material!
3. Drill the hole to its final diameter (#7). Drilling a smaller pilot hole first is recommended.
4. Deburr both sides of the hole.
5. Lubricate the hole and tap, then tap the hole being careful to get a straight start.
6. Mark the thread depth on the rod so that it will fully screw into the newly tapped hole, but not stick out of the back side of the plate.
7. Using the die and lubricant, thread the rod.
8. Assemble the parts, write your name, and turn it in!

Helpful Video

The video below shows roughly the process you will follow. If you have layout dye available, it is especially helpful.

Grading Rubric

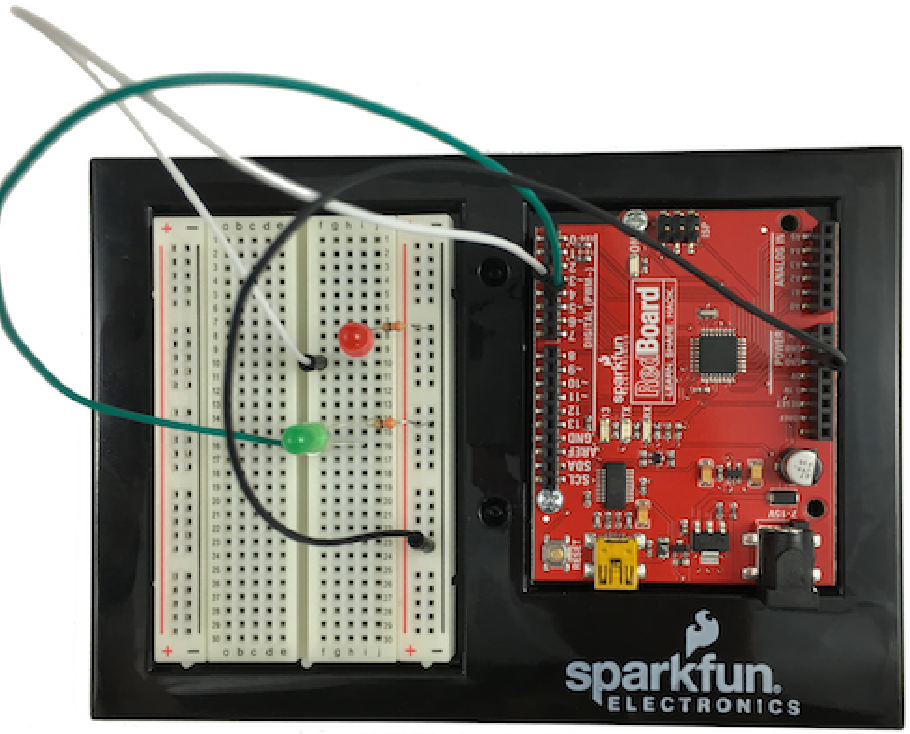
Description	Max Points
Turned in on time	5
Parts screw together	30
Centering of hole	10
Threads on dowel	10
Threads in plate	10
Total	65

Blinky 2.0

This activity will build on the *Blinky 1.0* activity we did using your new knowledge of electronics prototyping techniques. We will still blink LEDs, but we will use external LEDs on a breadboard.

Materials

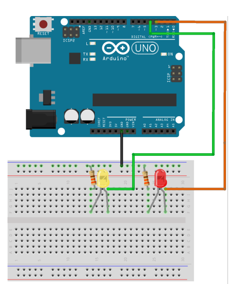
- Arduino UNO
- USB Cable
- Breadboard



- LEDs (two of your favorite colors)
- 330 Ω resistors (Orange-Orange-Brown)
- M/M Jumper wires
- Computer (Mac, Linux, Windows)

Procedure

- Hook up the LEDs, resistors, and Arduino using the jumper wires and breadboard as shown in the diagram below (click to enlarge).



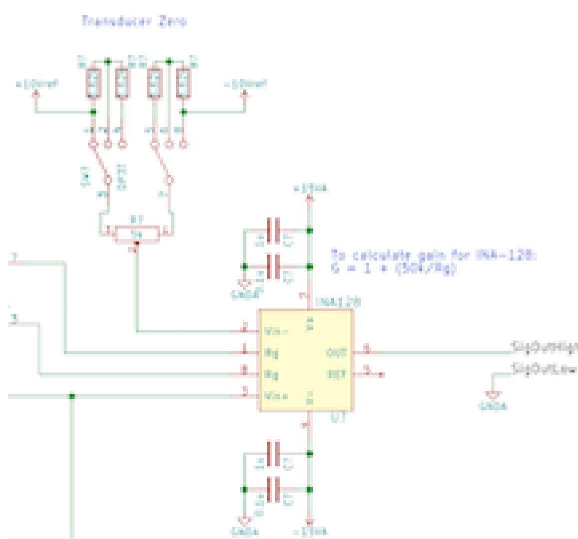
- LEDs are *polarized* components meaning they have a certain way they need to be in the circuit. On LEDs the short leg next to the flat edge is the ground (-) connection.
- You will need to bend the legs of the resistors to use them on the breadboard, you can do this with your hands or small pliers.

- Based on the code we used in the *Blinky 1.0* activity, write a new program that blinks the LEDs in an alternating pattern.
- Get creative. What patterns can you make? Can you add more LEDs or change the pin assignment for each LED?

Grading Rubric

Description	Max Points
Turned in on time	5
Code Compiles	10
LEDs blink in a pattern	10
Total	25

Project Proposal



After you have discussed your potential project ideas from the *Project Brainstorm* activity with the instructors, it is time to nail down exactly what your project will be and what the deliverables will be. Write a summary of one page or less that describes:

- What your project is.
- What it will do.
- What resources you will need to complete it.
- What parts you may need and a rough estimate of the cost.

This will serve as the specification document for your project goals at the end of the term.

Example

To help aid your writing, we've created a sample project proposal. Your proposal does not need to have this exact structure, but we have created it to clearly show what needs to be addressed in your document.

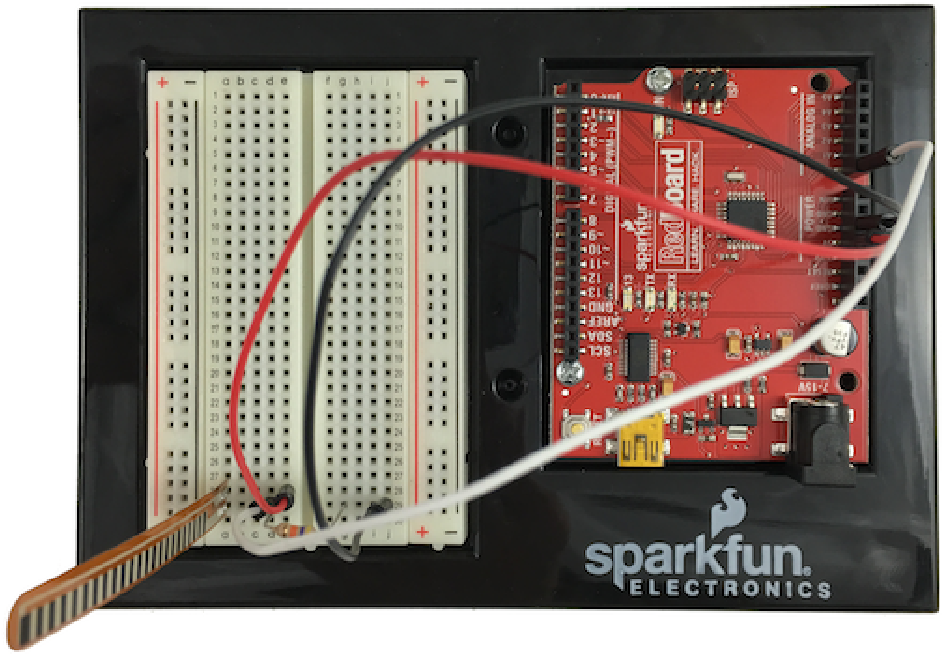
`Example Proposal`

Grading Rubric

Description	Max Points
Turned in on time	5
Describes Problem	10
Describes Solution	10
Describes Resources Needed	10
Has Project Budget	10
Total	45

Flex Sensor Data Acquisition

In this activity you will learn how to take a transducer, create a basic signal conditioning circuit, then record data from that system. We will be using the flex sensor from the Inventor's Kit and measure how far you have bent the sensor over time. This could represent the bending of a structure in the wind or the deformation of a system in laboratory. These sensors have also been used in virtual reality/game sensor systems and in the construction of robots. We will go through each step of the data acquisition process to build a system to record the approximate angle of bend of the sensor.



Materials

- Arduino UNO
- USB Cable
- Breadboard
- M/M Jumper wires
- Flex Sensor
- 47k Ω resistors (Yellow-Violet-Orange)
- Computer (Mac, Linux, Windows)

Sensor

The flex sensors in the activity are made by SpectraSymbol (checkout the datasheet). They are coated on one side with a polymer that has little conductive bits in it. When the sensor is flat (unbent), the characteristic resistance is about 30k Ω .

When you bend the sensor, the same number of conductive bits are stretched over a longer distance, making the resistance increase to about 70k Ω at a 90 degree bend. We can verify this with a multimeter.

Do **NOT** kink the sensor near the base or bend it in the wrong direction! This can damage it. It should only be bent as shown in the photo below.

The most recent shipment of sensors seem to be slightly different than the specifications used in the SparkFun tutorial materials. The unbent resistance is about 22k Ω and at ninety degrees bend it is about 48k Ω.

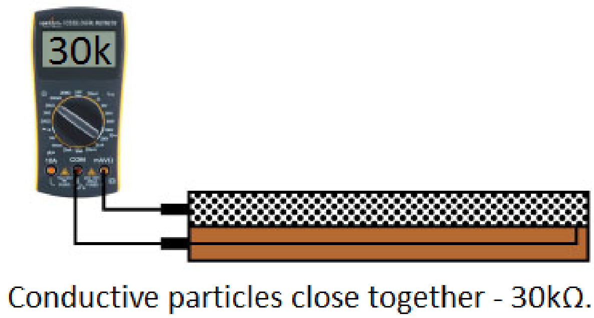


Fig. 1.71: Straight flex sensor (Image: Sparkfun)

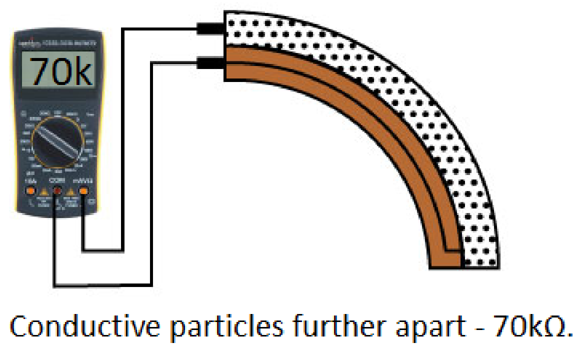


Fig. 1.72: Bent flex sensor (Image: Sparkfun)

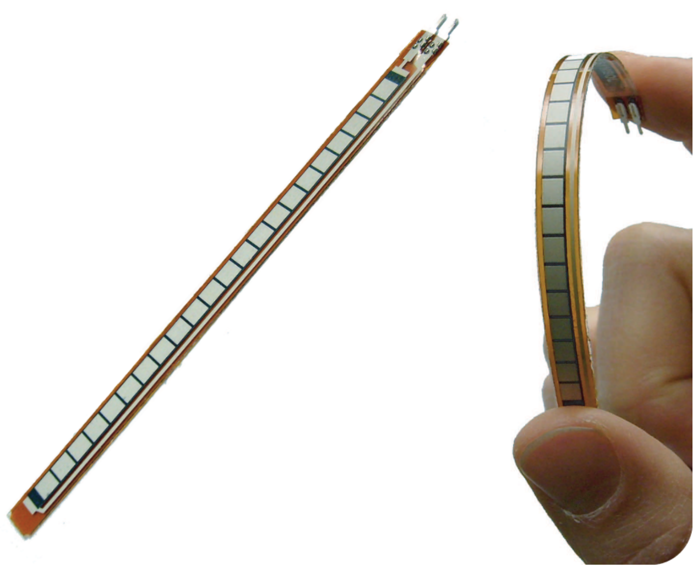


Fig. 1.73: Only bend the sensor in this direction (Image: Sparkfun)

Signal Conditioning

You have probably noticed that the Arduino doesn't have a port to plug in a sensor and measure its resistance. We need to condition the signal to be something that our acquisition hardware can handle. We are going to take a very simple approach here and use only a voltage divider circuit. This is a common technique that we will see again.

A voltage divider consists of two resistors in series. If we put a voltage V_{in} at the top of the series resistor string and ground the other end, the output at the junction of the two resistors will be:

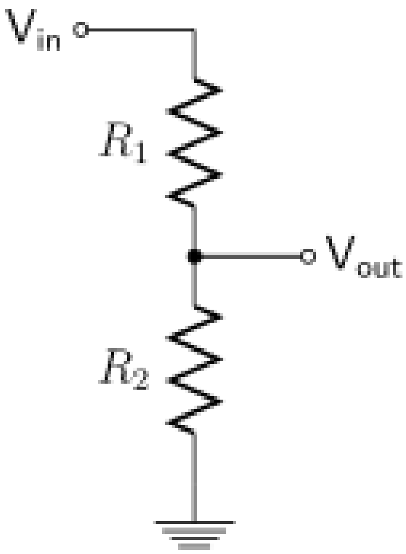


Fig. 1.74: Simple voltage divider (Image: Wikipedia)

$$V_{\text{out}} = V_{\text{in}} \frac{R_2}{R_1 + R_2}$$

We will implement the resistor divider by making R_1 in the circuit above be the flex sensor. As it flexes, the resistance will increase. This increase in R_1 will make the output voltage of the circuit go down towards ground. Ideally there would be a buffer circuit in between the voltage divider and the acquisition system, but for this particular setup and application it is not necessary. The voltage divider is designed to give us a wide range of output voltages, but some other sensors with smaller changes in resistance (such as strain gauges) would need amplification in addition to buffering. That's coming in another activity though. We can represent our system with the schematic below.

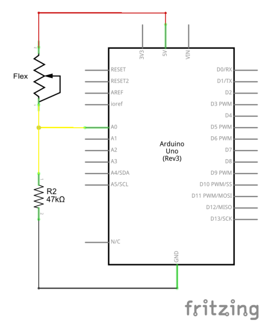


Fig. 1.75: Flex sensor schematic (Image: Sparkfun)

ADC/Firmware

Now that we have a conditioned signal that represents the physical thing we are trying to measure, it is time to turn those voltages into a digital representation that we can record. We will use the on-board analog-to-digital converter to read the voltage and convert it to a value the computer can record. The process behind this will be discussed in a future class.

The Arduino's ADC can read voltages from 0-5 VDC with a 10-bit resolution, meaning that we can resolve 1024 different values. That's plenty for what we are trying to do here. Let's have a look at the firmware that's going to run in this exercise.

```
/******************************************************************************
Modified from the SparkFun Flex Sensor Example Code
https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide#flex-sensor-overview
******************************************************************************/
```

```
const int FLEX_PIN = A0; // Pin connected to voltage divider output

// Measure the voltage at 5V and the actual resistance of your
// 47k resistor, and enter them below. This makes the angle
// calculation much more accurate.
const float VCC = 4.8; // Measured voltage of Arduino 5V line
const float R_DIV = 45900.0; // Measured resistance of 47k resistor

// Upload the code and try to determine an average value of
// resistance when the sensor is not bent, and when it is
// bent at 90 degrees. Enter those and reload the code for
// a more accurate angle estimate.
const float STRAIGHT_RESISTANCE = 22250.51; // resistance when straight
const float BEND_RESISTANCE = 48300.0; // resistance at 90 deg

void setup()
{
  Serial.begin(9600); // Startup the serial communications at 9600 baud
}

void loop()
{
  // Read the ADC
  int flexADC = analogRead(FLEX_PIN);

  // Calculate the voltage that the ADC read
  float flexV = flexADC * VCC / 1023.0;

  // Calculate the resistance of the flex sensor
  float flexR = R_DIV * (VCC / flexV - 1.0);


  // Use the calculated resistance to estimate the sensor's
  // bend angle my mapping the measured resistance onto the
  // known resistances at zero and ninety degrees of bend.
  float angle = map(flexR, STRAIGHT_RESISTANCE, BEND_RESISTANCE,
                   0, 90.0);

  // Send the results back to the computer formatted as a
  // comma delimited line.
  Serial.print(angle);
  Serial.print(",");
  Serial.println(flexR);

  delay(250); // Read the sensor at 4Hz.
}
```

The `setup()` function starts serial communication with the computer. In the main `loop()` function we read the ADC value with the `analogRead()` command and convert it to an actual voltage. Don't worry too much about that yet. We next convert that voltage to a resistance of the flex sensor by doing a bit of algebra on the voltage divider equation above to solve for R_1:

$$R_1 = R_2 \frac{V_{\text{in}} - V_{\text{out}}}{V_{\text{out}}}$$

which can be written a bit more nicely as:

$$R_1 = R_2 \left(\frac{V_{\text{in}}}{V_{\text{out}}} - 1 \right)$$

We then use the `map()` function which is a handy way to avoid doing the annoying math of scaling and calibration in this case. We assume the sensor is linear and map takes our no bend and bent resistance values and maps them to zero and ninety degrees. It then takes the measured resistance and estimates the bend angle based on those two end point calibration values. Checkout the documentation for the map function for the details.

Logging

Now that we have the Arduino reading the sensor and converting that digital value back into a meaningful physical unit, we need to record that data. Often in the lab this means writing custom software, but for our simple needs we can use the tools built into the Arduino IDE.

The serial monitor tool (magnifying glass icon) will show the serial traffic that the Arduino is sending back to us. You can copy and paste the data from there into a text editor and save it, but there is a limited row number history. You can pull this data into your favorite graphing tool of choice to make a plot.

If you go into the tools menu of the IDE, there is a Serial Plotter option. Clicking that will show a running graph of the serial data coming in, but it is rather limited. There is no time scale and there can only be one plot running. To use the serial plotter we need to change the serial output section of the code to output only a single angle value per line, no comma or resistance. After you get your calibration resistances (see below) you can modify the serial section to just be `Serial.println(angle);` (just commenting out the other lines is a good idea).

For a good summary of the current state of the serial plotter, checkout this blog post by Rheingold Heavy.

If you have problems with either of these methods, check that the baud rate of the terminal/plotter is set to 9600. You can also use an external serial monitor like CoolTerm that will log directly to a file and has many other bells and whistles.

Procedure

- Connect your flex sensor to the Arduino as shown in the diagram below.

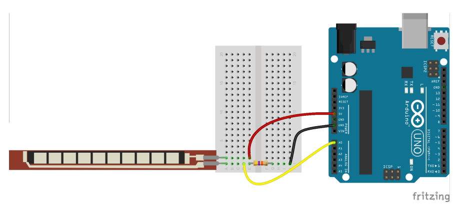


Fig. 1.76: Flex sensor hookup diagram (Image: Sparkfun)

- Plug the Arduino into the computer and upload the program listed above.
- Experiment with the plotter and sensor. Try changing the resistor in the voltage divider out for one of higher or lower value. How does the output voltage change? Does the range of output voltage change?

Deliverables

Turn in a plot of the bend of your sensor over time as you manipulate it. This can be a plot from the Arduino IDE or one you make in your favorite graphing software.

Grading Rubric

Description	Max Points
Turned in on time	5
Electronics complete	10
Graph of data sent in	10
Total	25

Building an XRD State Machine

X-ray diffraction is a commonly used laboratory technique to study the properties of crystals or determine the makeup of a bulk sample. The video above summarizes the operation of a diffractometer and shows one in operation. An X-ray diffractometer emits X-ray radiation towards the sample at a known angle θ. The radiation is elastically scattered off atoms in the crystalline lattice. The angle of reflection is equal to the angle of incidence. The reflected radiation is counted by a detector. By sweeping the emitter and detector through a range of angles. we will see peaks in the X-ray spectra characteristic of the material present. Since crystals have regular structures, at certain incidence angles, the reflections from two adjacent layers constructively interfere and create a larger radiation count. In this activity, you will design a simple state machine to control an X-ray diffractometer.

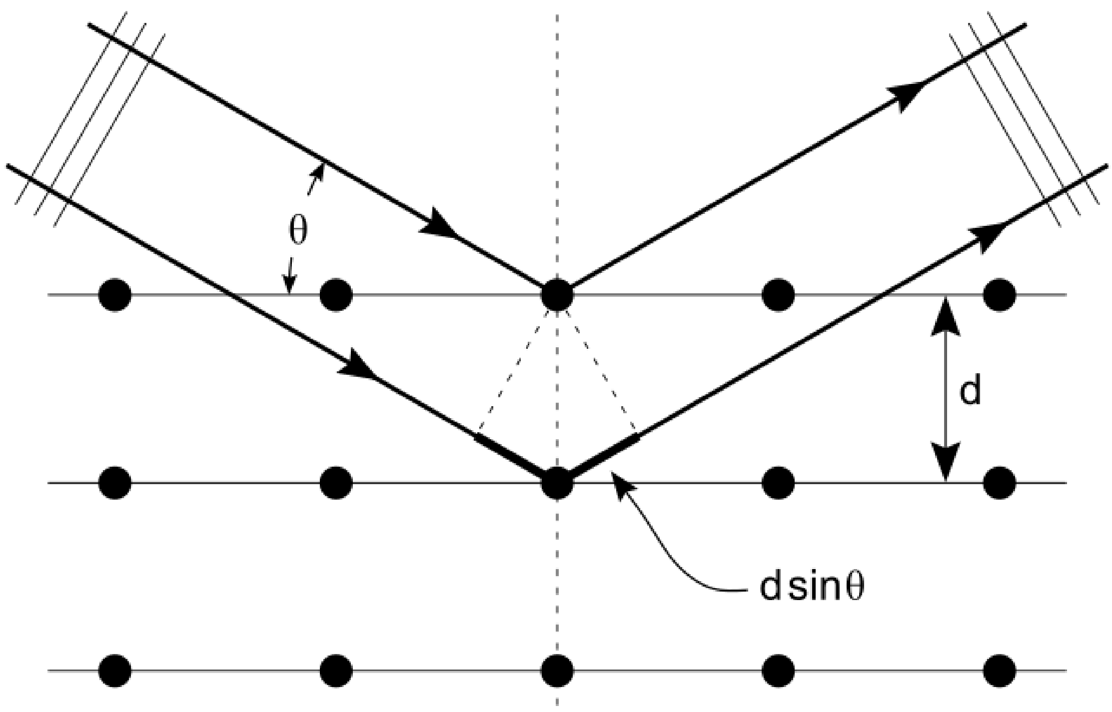


Fig. 1.77: Bragg Diffraction (Wikipedia)

We often use user stories to define how a machine should operate. The user story describes the process that the user or technician will go through to use the machine.

User Story

The user approaches the machine in the off condition. The main power is turned on and the machine starts with the X-ray source off in the X-ray shutter closed. The technician opens the safety doors and inserts a sample for analysis.

The doors are closed and the technician sets the scan parameters in the control software. The software specifies the minimum and maximum angles at which to scan, the step at which to go between these angles, and how long the radiation should be counted at each angle. Once the start scan button is clicked, the machine must verify that the safety doors are closed and the safety interlock switches are closed. The X-ray shutter is then opened and the X-ray source turned on. The machine moves to its minimum scanning angle and begins to count the radiation for the specified time. After this time, the machine moves by the specified step amount and counts radiation again for the same amount of time. This process is repeated until the maximum scan angle is reached. If at anytime during the operation of the machine the safety interlock switch is open, the machine must immediately turn off the X-ray source, close the X-ray shutter, and go to the shutdown state. When the scan completes successfully, the machine must turn off the X-ray source, close the X-ray shutter, and go to the shutdown state as well.

Assignment

Using what we've learned in class about state machines and state machine diagrams, draw a state machine diagram that describes the operation of the X-ray diffractometer. Remember to specify start and end points, error sources, and keep the number of states reasonable.

Grading Rubric

Description	Max Points
Turned in on time	5
Number of states	5
Logical flow of machine	20
Error sources included	5
Follows users story	20
Total	55

Arduino Stoplight

In this activity you will make a simple single stoplight controller with an Arduino UNO and some LEDs. You will become familiar with using the Arduino programming environment and learn how to use the General Purpose Input/Output (GPIO) pins

Fig. 1.78: Borrowed from Wikipedia.

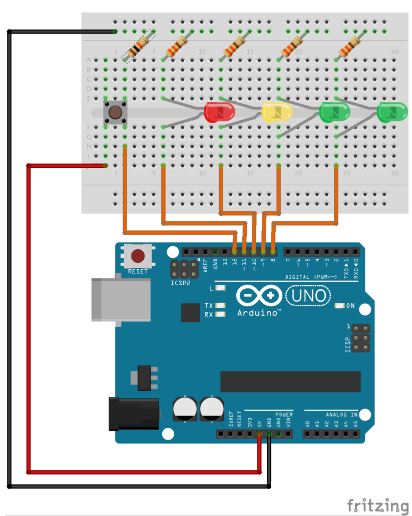
on the microcontroller. You will also practice using good software design technique by implementing well known design patterns and making maintainable code.

Materials

- Arduino UNO
- USB Cable
- LEDs (Red, Yellow, Green)
- 330 Ω resistors (Orange-Orange-Brown)
- 10k Ω resistor (Brown-Black-Orange)
- Push button (momentary-on type)
- Breadboard
- M/M Jumper wires
- Computer (Mac, Linux, Windows)

Procedure

1. Download and install the Arduino IDE
2. Connect the button, stop, caution, go, and left turn LEDs as shown in diagram.



3. Start the Arduino IDE. Open the Blink example from: *File* → *Examples* → *01.Basics* → *Blink*. Read the comments and make sure you understand how it works.

4. Connect your Arduino and hit the upload button. If it fails, check the board and port settings (in the *Tools* menu). Make sure the on-board LED is blinking to show a successful program upload.
5. Change the pin number in the blink example to that of one of your LEDs. Make sure that the LED on the breadboard blinks, if not, you need to check the connections. Do this for each of the 4 LEDs.
6. Draw a state machine diagram to meet the specifications of the attached requirements. Turn this in with the assignment!
7. Build the state machine in the Arduino IDE and test it on your stoplight. Your final code should be commented, compile and run, and meet the specifications. Be sure to use good coding practices! Your code will be tested/graded by an identical Arduino setup.

Requirements

- Begins in the red light state.
- Red light cycle lasts for 3 seconds.
- Yellow light cycle lasts for 1.5 seconds.
- Green light cycle lasts for 3 seconds.
- Works like a normal stoplight would, only one light on at a time and in the normal order (Red - Green - Yellow - Red).
- If a car was present in the left turn lane (simulated by holding down the push button) **before** the green light state, add a green left turn light for 2 seconds. If no car is present, repeat the cycle.
- Uses the **state machine implementation** with **functionalized code**. No interrupts allowed!

Example

The video below illustrates the proper operation of the stoplight.

Grading Rubric

Description	Max Points
Turned in on time	5
Code Compiles	20
Meets Project Requirements	40
Total	65

Digital Logic Operations

Given what you have learned about digital logic and how we write it, complete the truth table for the digital logic circuit below and write the equation representing the output using standard combinatorial notation.

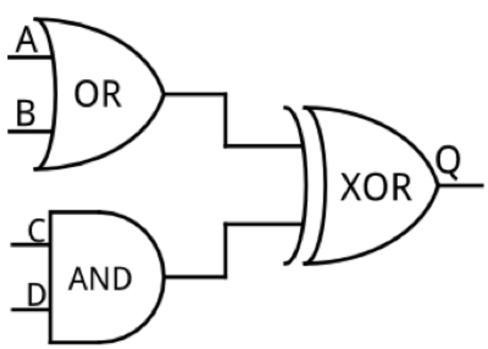
Digital Representation

Computers communicate with a binary (on/off) logic system. We can represent numbers in decimal (base 10), hexadecimal (base 16), or pure binary (base 2). All binary numbers are represented in **big endian** format.

1. Complete the table below with the other two forms of the given number. Assume that all values are **8-bit**. (2 pts. each, 20 total)


Blink | Arduino 1.6.9
Blink
Upload Button
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://www.arduino.cc
This example code is in the public domain.
modified 8 May 2014
by Scott Fitzgerald
*/
// the setup function runs once when you press reset or power the board
void setup() {
// initialize digital pin 13 as an output.
pinMode(13, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
Arduino/Genuino Uno on COM1

Activity: Fill out the truth table and write an expression for the output of this combinatorial circuit



		AB			
		00	01	10	11
CD	00				
	01				
	10				
	11				

Image: sparkfun.com

Decimal	Hexadecimal	Binary
		0110_0111
	0x0D	
124		
		1111_1111
256		
	0xDE	
		1010_1101
190		
		1010_1010
85		

2. What is the largest decimal value that can be represented by an unsigned 8 bit binary value? Please answer is base 10. What about 10, 16, and 24 bits? (8 pts.)

3. What values (two) in the table above would be easy to use as troubleshooting characters when looking at the signal on an oscilloscope or logic analyzer? Why? Consider the number and pattern of logic value transitions. What would be easy to spot? (3 pts.)

4. Complete the table below with the unsigned and two's complement value of the given binary number. *Answers should be base 10 integers*. (2 pts. each, 12 total)

Binary	Unsigned Value	Two's Complement
0111_1111		
1000_0001		
1111_1110		
0101_1010		
0000_0001		
1000_1100		

5. Assume a number is stored in the IEEE 754 single-precision floating point representation. The float is stored in memory as 0x40490E56. What is the floating point number in decimal form? (5 pts.)

6. Why should floating point calculations be avoided, especially in single-precision systems? (5 pts.)

7. How could you implement mathematical operations that involve floating point math on a resource constrained processor, like an Arduino, without using a floating point library? (3 pts.)

8. What is a common way that complex math like sin, cos, and log are implemented on resource constrained processors? (3 pts.)

Mechanical Drawing Activity

Now that you know how to make a mechanical drawing that the machine shop can read, it is time to practice. You will be provided (or need to find) a simple part and reverse engineer it into a mechanical drawing for the shop. This is a common practice when you need to make a copy of a part to replace a broken part or to create a slightly modified design of some part on a piece of equipment. Use the necessary measuring tools to create a fully dimensioned drawing of your part, on a one-to-one scale if possible. Don't forget to use any necessary center lines and to use good dimensioning style. Do not pick a difficult to draw cast part unless you really want a challenge! Brackets and simple lab jigs/fixtures are a good idea for your first try.

Mechanical Design Activity

We are going to design a simple part using the online CAD tool OnShape . There are a variety of other tools available for you to use, but OnShape is online so we don't need to worry about getting a package installed on everyone's machines. While designing our part, we will consider what we just learned about design for 3D printing and the best practices of modern parametric CAD.

We will be designing a holding system for a direct current displacement transducer (DCDT). This setup has a few parts and we will design as much as we can in class. You are encouraged to keep working on and finish the design on your own, but it is not a requirement. The completed parts from this design activity have been built and are currently in regular laboratory use.

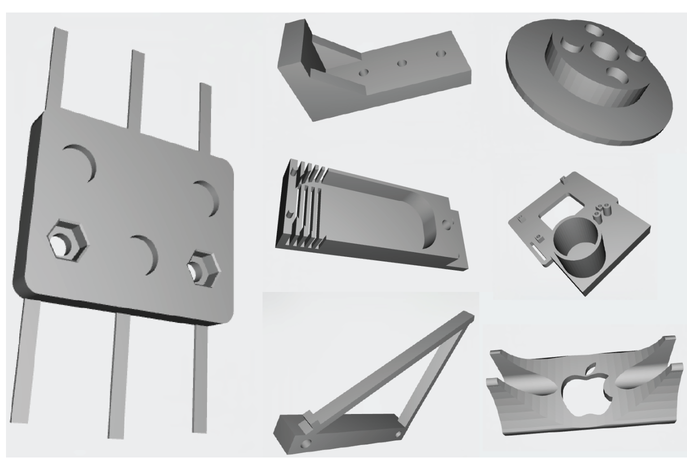
If you need reminders of how to do things in OnShape, be sure to checkout their website that has a

lot of great tutorials, webcasts, and other resources. You can also watch the lecture video in which we go through the design in class.

3D Printing Activity

In this activity you will design and print a 3D part. You may take an existing CAD file from a repository such as Thingiverse, GrabCAD, etc. and modify it to suit your purpose (must be a non-trivial change) or create your own design. You can make your design in your preferred CAD tool, a few great choices are OnShape, OpenSCAD, FreeCAD, AutoDeskInventor , and SketchUp These can be simple brackets and parts or very complex structures. Be sure to make something that actually can be 3D printed though - remember the design concerns we discussed in class. A few examples of CAD files designed for 3D printing are shown below.

Many universities, public libraries, and maker spaces have 3D printers that you can use for little or no cost. Penn State operates the Maker Commons as a part of the library that allows students to print. There are also commercial service bureaus that will make your print on professional grade machines and ship it to your door. Of these, Shapeways seems to have the largest variety of materials and services. Often service bureaus and library services can get very busy, so make sure you allow enough time for manufacturing and shipping!



Grading Rubric

Description	Max Points
Turned in on time	5
Part created or modified	20
Design for 3D printing	20
Total	45

Pressure Vessel Design Activity

Pressure vessels are everywhere in the experimental lab, from the air compressor to experimental material deformation and failure test chambers. In this activity you will perform the calculations we discussed in class to design a simple pressure vessel that could be used in the laboratory.

When doing your design, use good practice and document your assumptions, design decision reasoning, and clearly show all calculations. Calculations may be performed by hand or with a script. If you use a script, please clearly comment it and include it with your assignment. Python notebooks are ideally suited to this task as they allow you to combine documentation, photos, etc. with your code into a complete lab writeup, but you may use any tool you wish. We strongly encourage you to **NOT** use Microsoft Excel as it is an accounting and spreadsheet program, not an engineering program and can be very difficult to debug.

Background

Gas hydrates are a unique material in which molecule(s) of a gas, such as methane or carbon-dioxide are trapped inside a cage of water molecules. This solid looks similar to ice, but can have very different properties. Hydrates (or clathrates as they are sometimes called) are an active area of research and could be important for the storage of greenhouse gasses and present hazards to hydrocarbon drilling and exploration operations. Hydrates were even a large barrier to getting the Deepwater Horizon spill under control. When engineers attempted to capture the well's output with a metal cage termed the "top-hat solution", hydrates filled the structure and lifted it off of the well as they are less dense than water. Hydrates can be important for understanding submarine slope failure and geologic processes on other planets and moons.

Pressure Vessel Requirements

To investigate some of the properties of gas hydrates, we want to be able to make some in the laboratory. Since this is a new project, we have no facilities other than standard laboratory and break-room equipment. We need to design a pressure vessel that lets us produce some methane hydrate. We can then take the hydrate out of the vessel and light it on fire (in a controlled environment) to impress our funding agency and get funding for better equipment.

- Hydrate must be a stable phase when the pressure vessel is placed into the laboratory refrigerator, which is stable at 2 ° C.
- The safety factor must be at least 4 at the maximum expected operating pressure of the vessel.
- The vessel will be made of 6061-T4 Aluminum stock.
- Assume that the O-ring sealing is a solved problem for this vessel, no need to worry about designing the gland dimensions.
- The vessel will be capped with two flat plates that are bolted onto the end of the vessel.
- The vessel inner diameter should be 8 cm.
- The vessel inner length should be 10 cm.

Resources

- Information about Aluminum
- Periodic Table
- Bolt strength information

- Ideal Gas Law
- Methane Hydrate Information

Questions/Design Tasks

1. Knowing that the laboratory refrigerator is 2 ° C, what is the expected pressure of the phase boundary from gas + liquid water to gas hydrate? To be sure we are completely into the stable field and give our refrigerator a little room for error, add 50% to that. What is this pressure?

   (4 pts.)

2. Let's start by filling our vessel with 100 mL of liquid water, then sealing it up. What is the total volume of the vessel? What is the gas headspace of the vessel? (2 pts.)

3. Stoichiometrically, hydrate is $4CH_4 \cdot 23H_2O$. We want to make sure we can completely convert all of the water to hydrate, so we make sure that we have 6x as many water molecules as we have gas molecules. How many moles of water are in the vessel? How many moles of gas are required to complete the reaction? (4 pts.)

4. Given the headspace you calculated, how many moles of gas are required to keep the pressure at our target at 2 ° C? How much gas do we then need in the vessel in the beginning, before the reaction consumes any? (6 pts.)

5. When we charge the vessel with gas, we can assume that it is at room temperature. As the gas cools in the refrigerator, the pressure will be reduced. Given the amount of the gas you calculated above, what is the maximum pressure of the vessel when at room temperature (20 ° C) This will be your target design pressure. (4 pts.)

6. What is the yield strength, Young's modulus, and Poisson's ratio of 6061-T4 Aluminum? (3 pts.)

7. Calculate the wall thickness required to meet the maximum pressure and factor of safety requirements. Round the thickness up to the nearest millimeter. You can use a guess and check method, rewrite the equation, or work it out with your favorite numerical method. (6 pts.)

8. Plot the circumferential stress as a function of radius through the wall material. Where is this stress maximized? (6 pts.)

9. Calculate the axial stress on the pressure vessel. Is this greater or less than the circumferential stress? Does it mean you need to invoke any design changes? (6 pts.)

10. Calculate the maximum radial stress on the pressure vessel. Is this greater or less than the circumferential and axial stresses? Does it mean you need to invoke any design changes? (6 pts.)

11. Given the factor of safety we desire, what should the thickness of the end platens be? What will the maximum stress on the end cap be at this thickness? (6 pts.)

12. What is the maximum deflection of the end cap under the maximum pressure conditions (i.e. the working pressure * safety factor) (6 pts.)

13. Assume we are going to use 6 bolts to fix the platen to the end of the pressure vessel. How much stress will each bolt need to be able to support? Look at the strength of different bolt sizes and grades. What size of grade 8 hardware should be used to prevent fastener failure? (8 pts.)

14. What concerns do you have about this design? What things have we not considered or what assumptions that were made in the calculations may not be completely valid? (5 pts.)

15. What practical/manufacturing constraints could you apply to the design to make it cheaper (via reducing the cost of stock metal and/or the machining)? Can you do this without impacting the safety of the design? (5 pts.)

Arduino Voltmeter

A voltmeter is a handy tool to measure voltage at different places in the circuits you are building or troubleshooting. For this class, we didn't get voltmeters for everyone because we can turn your Arduino into an accurate enough voltmeter! In this section you will see how to create a four-channel single-ended or two channel differential voltmeter with the materials in your inventor's kits.

Materials

- Arduino UNO
- USB Cable
- Computer (Mac, Linux, Windows)
- Jumper Wires
- 16x2 LCD Display
- 10k Potentiometer
- Breadboard

Hookup

Follow the hookup diagram for the LCD screen below (click to enlarge):

Single Ended vs. Differential

You can operate this system in single ended or differential mode. Single ended means that all measurements are shown with respect to ground (0 volts). In the single ended mode, connect wires to pins A0, A1, A2, A3. Each of these can then be used to probe a point in the circuit and display its voltage above ground.

In differential mode, the difference between two of the inputs is displayed, in this setup we display A1-A0 and A3-A2. This allows us to measure the voltage difference across components in the circuit. You could of course do this math mentally, but having it displayed natively is often handy. Connecting A0 and A2 to ground would give you a single-ended like behavior on A1 and A3. Differential inputs are often used when measuring voltage drops across componets. They are also generally lower noise as the common mode noise is greatly reduced.

Useage Notes

It is very likely that the first time you power up this circuit you will not see anything on the screen. Adjust the potentiometer to change the contrast of the display until you can read the screen. If it is still not working, check that you have uploaded the sketch correctly and that all electrical connections are correct.

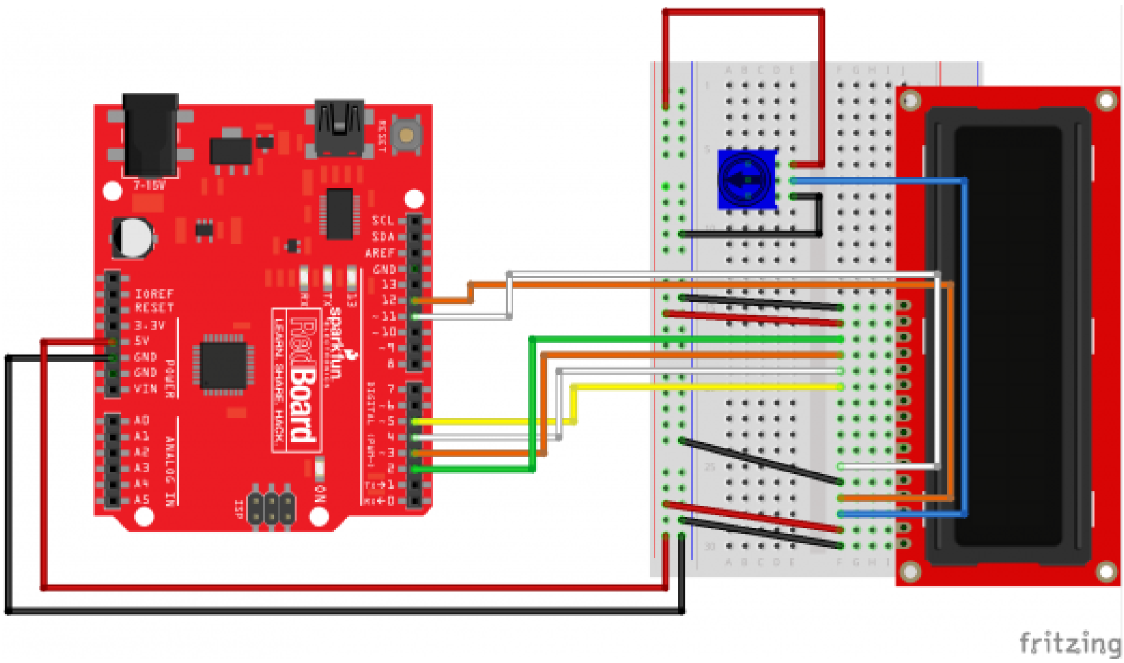


Fig. 1.80: Image: Sparkfun

Verify the correct operation of your volt meter by measuring the 5V and 3.3V ports on your Arduino.

Single-Ended Voltmeter Code

```
#include <LiquidCrystal.h>

#define NUM_TO_AVERAGE 500

LiquidCrystal lcd(12,11,5,4,3,2);

long a0_value, a1_value, a2_value, a3_value;
float a0_voltage, a1_voltage, a2_voltage, a3_voltage;

void setup()
{
lcd.begin(16, 2);
lcd.clear();

lcd.print("Arduino V Meter");
delay(2000);
lcd.clear();
}

void loop()
{

a0_value = 0;
a1_value = 0;
a2_value = 0;
a3_value = 0;
```

```
for(int i=0; i<NUM_TO_AVERAGE; i++){
  a0_value += analogRead(0);
  a1_value += analogRead(1);
  a2_value += analogRead(2);
  a3_value += analogRead(3);
}

a0_voltage = a0_value / NUM_TO_AVERAGE * (5.0/1023);
a1_voltage = a1_value / NUM_TO_AVERAGE  * (5.0/1023);
a2_voltage = a2_value / NUM_TO_AVERAGE  * (5.0/1023);
a3_voltage = a3_value / NUM_TO_AVERAGE  * (5.0/1023);

lcd.setCursor(0,0);
lcd.print("0:");
lcd.print(a0_voltage, 3);

lcd.setCursor(9,0);
lcd.print("1:");
lcd.print(a1_voltage, 3);

lcd.setCursor(0,1);
lcd.print("2:");
lcd.print(a2_voltage, 3);

lcd.setCursor(9,2);
lcd.print("3:");
lcd.print(a3_voltage, 3);

}
```

Differential Voltmeter Code

```
#include <LiquidCrystal.h>

#define NUM_TO_AVERAGE 500

LiquidCrystal lcd(12,11,5,4,3,2);

long a0_value, a1_value, a2_value, a3_value;
float a0_voltage, a1_voltage, a2_voltage, a3_voltage;

void setup()
{
  lcd.begin(16, 2);
  lcd.clear();

  lcd.print("Arduino V Meter");
  delay(2000);
  lcd.clear();
}

void loop()
{

  a0_value = 0;
  a1_value = 0;
```

```
  a2_value = 0;
  a3_value = 0;

  for(int i=0; i<NUM_TO_AVERAGE; i++){
    a0_value += analogRead(0);
    a1_value += analogRead(1);
    a2_value += analogRead(2);
    a3_value += analogRead(3);
  }

  a0_voltage = a0_value / NUM_TO_AVERAGE * (5.0/1023);
  a1_voltage = a1_value / NUM_TO_AVERAGE  * (5.0/1023);
  a2_voltage = a2_value / NUM_TO_AVERAGE  * (5.0/1023);
  a3_voltage = a3_value / NUM_TO_AVERAGE  * (5.0/1023);

  lcd.setCursor(0,0);
  lcd.print("A1-A0: ");
  lcd.print(a1_voltage - a0_voltage, 3);

  lcd.setCursor(0,1);
  lcd.print("A3-A2: ");
  lcd.print(a3_voltage - a2_voltage, 3);
}
```

Component Basics

In this activity you will gain experience with basic electric circuit analysis and working with sensors. Part of the activity will be done in teams of two and the other sections will be completed individually. Start on the Teamwork activities (items 1-8) and complete the individual work (items 9-12) afterward. Each student should turn in a complete assignment/report. Resistors A and B will be given out in class.

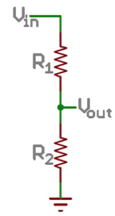
Materials

Tasks

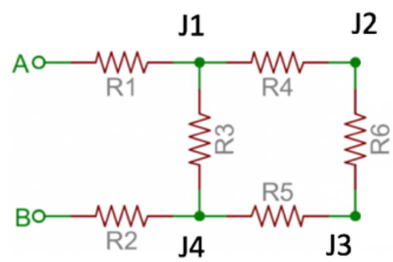
1. Each team will build a voltmeter with one Arduino board and use it to analyze circuits built on their second board. Start by completing the *Arduino Voltmeter* activity. Show us your group's working Arduino voltmeter in class or include a photo of it with your homework submission.

2. Using your Arduino voltmeter, determine the resistance of Resistors A & B, briefly describe your method. Include a sketch of your circuit if necessary.

3. Derive an equation for the Voltage at Vout in the sketch below. Build this circuit with R1=Resistor B and R2 as your 10k potentiometer. Measure the range of voltage at Vout for Vin = 1.67 V. Briefly describe your methods.



4. Now configure a circuit as shown below, where R1 is Resistor A, R3 is Resistor B, all other resistors are 330 Ω. Connect 5 V to A and ground at B. Label the currents in each leg of the circuit and write out the equations for the current loops. Use your equations to determine the voltage at each junction relative to ground. Finally, measure the voltage at each junction and comment on the comparison between your predictions and the measured values.



5. Now change R2 to the 10k potentiometer. What is the range of voltage you can measure at J3 by sweeping the potentiometer through its entire range?

6. Set up the wiring for a blinking LED and use your 10k potentiometer to control the blinking rate. Write the voltage to the serial monitor at the same rate that your LED is blinking. Start with this code fragment and produce a **fully commented** code to solve the problem.

```
int sensorPin = 0;     // Connect the pot to analog pin 0
int ledPin = 13;       // Connect the LED to digital pin 13

void setup() {
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);
}
```

```
void loop() {
    int sensorValue;

    sensorValue = analogRead(sensorPin);

    Serial.print("Pot. output value is: ");
    Serial.println(sensorValue);

    digitalWrite(ledPin, HIGH);
    delay(sensorValue);
    digitalWrite(ledPin, LOW);
    delay(sensorValue);
}
```

7. What happens when you vary the position of the pot? Measure the voltage with your Arduino Voltmeter and compare it to those values.

8. Convert the output that you measure from the pot. (sensorValue) to voltage. Include this in your code with a brief explanation of the conversion.

9. Build a thermometer using the TMP36 temperature sensor. You may find the datasheet helpful. Complete the wiring and write the code from scratch, without using any resources on-line or otherwise. This activity should be done individually. Your code should include detailed comments about all steps of the process. For an example of the expectations for your documentation, see the Ping code example in *File -> Examples -> Sensors -> Ping*.

10. What is the output voltage of the TMP36 at 25 ° C (room temperature)? Can you verify what the room temperature is with a separate thermometer for a more accurate assessment? Measure the output voltage yourself and comment on the comparison of your value with the manufacturer's specs.

11. What is the expected (and maximum) excitation voltage for the TMP36? Our Arduino has a standard 5V output. What would happen if you powered the TMP36 with 1 V rather than 5?

12. Set up a simple calibration with two known temperatures (room temperature and your skin temperature for example). Use the serial monitor to measure temperature as a function of output voltage from the TMP36. Use the serial plotter (found in the *Tools* menu) to plot temperature vs. time. Collect some data and plot temperature vs. output voltage. Comment on the comparison of your measurements with the manufacturer's specs.

Due: 10/20/16 A summary with plots, answers to all questions, sketches of your wiring diagrams, and your code (make sure to name your files appropriately).

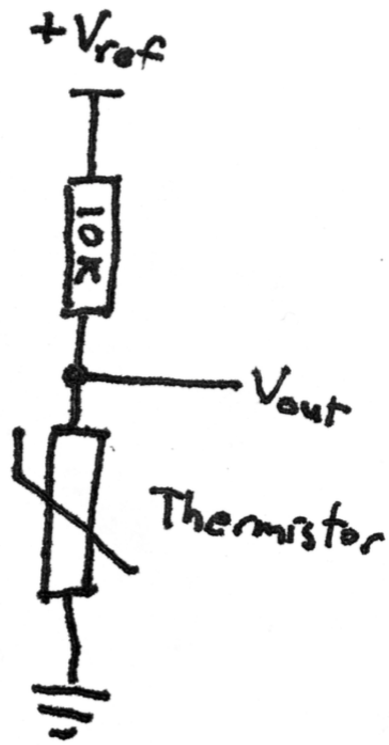
Thermistor Calculation Activity

In this activity you will learn how to calculate temperatures measured by a thermistor in a simple resistive divider circuit. The data were collected at around room temperature on a frictional shearing experiment. As the experiment was sheared, some heat was generated. Calculate the time-series of the temperature throughout the experiment. The data collection rate is one sample per second.

Questions

1. The thermistor part number B57861S103A39 was used in this experiment. Download the datasheet below and find the $B_{25/100}$ and R_{25} values for this part. What are they? (2 pts.)

2. Calculate the resistance of the thermistor for the time series data. The circuit was setup as shown below. Vref and Vthermistor are provided for each sampling time in the datafile. Is the mean above or below the R_{25} value? What does that tell you about the expected temperatures values? (4 pts.)



3. Calculate the temperature of the thermistor for the time series data. What are the maximum and minimum values? (4 pts.)

4. Make a plot of the time series of temperature. Assume that the data were collected at one sample per second. (2 pts.)

Data

`thermistor_data.txt`

`Datasheet`

Calibration Activity

In order to trust the measurements from our sensors and to get physical quantities in meaningful units (rarely are volts meaningful), we need good calibration data on the instrument. Ideally the response of the sensor is linear across its designed measurement range or the response is easily characterized. Minimal hysteresis effects are also desirable. In this activity you will perform a calibration on a transducer. This is a real exercise that we must perform on many transducers in the laboratory on an annual or semi-annual basis.

Tasks

1. Hookup the transducer to the power supply and voltmeter as shown below or as advised by the instructor. In this lab we will be using a direct current displacement transducer (DCDT) for the transducer and a vertical height gage as the transfer standard. Setup the height gage and DCDT to perform the calibration. Make sure that all components are aligned to reduce any error during the calibration. Take a photo of your setup and include it in the lab report. (4 pts.)

2. Measure the output of the power supply. In this case, the power supply should be set to +/- 15 V. What are the actual output values? (2 pts.)

3. What are the maximum and minimum voltages you would expect to be able to measure as outputs from this transducer? (2 pts.)

4. Move the transducer until you are at one end of the output range. Move the transducer core in 1 mm steps using the vertical height gage and record the output voltage at each step until you reach the other end of the output range. You could record the output continuously on a computer in the laboratory. In this case, at a random time, look up and write down the number displayed on the voltmeter at that instant. Repeat this three times and use the average value of those readings as the measurement at that displacement. Once you have reached the maximum range, do back down the range in the same 1 mm steps repeating your measurements. This will help us understand if there are any hysteresis effects in the transducer. Present your raw and averaged data in a data table. (10 pts.)

5. Make a plot of displacement vs. output voltage for the transducer. Is the output of the transducer linear over the entire range? Are there any hysteresis effects observed in the data? If so, what could have caused them? (5 pts.)

6. Fit a line to the linear portion of this data and explain the meaning of the slope and intercept of this line. What can we say about readings outside of the linear portion of the transducer's range? Would we use them as experimental observations? (5 pts.)

7. Apply your calibration factor to the transducer sample data to provide a relative displacement time-series. The data file contains two columns of data separated by commas. The first column is relative time in seconds, the second is the output of the transducer in volts Assume the transducer starts at "0 mm" displacement and produce a plot of the displacement time series. (6 pts.)

```
DCDT data file
```

8. What calibration biases could be introduced in our setup? (2 pts.)

9. What other factors could influence the output of the transducer that we are not considering in this calibration? (2 pts.)

Strain Gage Activity

In this activity you will learn how to apply strain gages to an object of your choice. It could be a rock, a ceramic mug, or anything else you think would have an interesting strain signal when squeezed, bent, or twisted by hand. We will build on this activity for several classes when learning how to condition, amplify, and record signals from transducers.

Materials

- Item to instrument (rock, metal plate, etc.)
- Strain gages (4x)
- Small gauge hookup wire
- Superglue
- Cellophane tape
- Acetone and cloth
- Square
- Marker
- Soldering iron/solder
- Tweezers

Procedure

1. Clean the surface you'll use with acetone. It the surface is very smooth, roughen it first with some emery cloth to help adhesion.
2. We will be creating a full bridge circuit using four gages. Layout a square pattern on your test object that lets you place a gage on each side of the square. Mark the center of each side as well. The needs to be really square and relatively accurately measured distances - use good technique!
3. Place the strain gauge right side up on a clean surface. Get a short piece of cellophane tape and place it over the gage, taking care to **not** cover the solder pad area of the gages.
4. Using the superglue and tweezers, place one gage centered on each side of the square you just laid out. Place a **small amount** of glue beneath each gage. Any glue that gets onto the contacts will make soldering much more difficult later! Stick the gage in place with the tape. In an instrumentation for research setting, using the manufacturers gage bonding chemicals is recommended.
5. After all of the glue has dried, hookup the strain gages in a full bridge configuration – pay attention the orientation of the gages when hooking them up or you could get no signal.
6. Hookup your bridge to a bi-polar power supply (two 9V batteries for example) and to a DMM set to read DC Voltage. The output under no-strain should be near zero volts. When you squeeze or deform your test object, you should observe a change in the output voltage. Be sure you are using the lowest range on the meter as this signal will likely be a few millivolts. We need to amplify the signal, but that is a future activity.

Deliverables

Show us your strain gaged object and the output voltage change when you stress your object.

PID Controllers

It can take awhile to gain some intuition about control systems, especially PID controllers. In this activity you will build a simple PID controller with your Arduino and adjust the different gain settings to see their effects. This lab is largely based on a lab written by Bret Comnes and A. La Rosa at Portland State University. Their original lab activity is available on their website or GitHub repo .

The control system you will build tries to maintain a constant light level on a photo resistor. You can think of this like your laptop screen that becomes brighter or dimmer based on the ambient light levels. This is a closed-loop system since we have a feedback path. An open-loop equivalent would be a traditional dimmer switch in a room. If the ambient light (sunlight) level changes, there is no feedback to change the room lights. You must walk over to the switch and adjust it manually (feed forward control).

In our system, an LED will shine directly into the photo resistor. The setpoint of the system is adjustable with a potentiometer on the board. The gains of the controller can be changed using the serial interface.

Materials

- Arduino UNO
- USB Cable
- LED
- 330 Ω resistor (Orange-Orange-Brown)
- 10k Ω resistor (Brown-Black-Orange)
- Photo resistor

- Breadboard
- M/M Jumper wires
- Computer (Mac, Linux, Windows)

Tasks

1. We could write all of the PID control code ourselves, but there is an Arduino library available with a very robust controller. Installing the library is easy. Download the library zip file from the GitHub repository here or clone into the repository if you are comfortable doing that. Unzip the folder and move it to the Arduino library folder. In most installations you will find this in your Documents folder: *~/Documents/Arduino/libraries*. If your folder has hyphens in the name, change them to underscores. If the Arduino IDE was open, restart it. You should find new examples for the PID library in *File* -> *Examples* -> *PID*. Have a look at the examples to get an idea of how the library works.

2. Using the parts in your kit, build the circuit we will use shown below. Take a photo of your circuit and attach it to the lab report. (2 pts.)

3. Read the following sketch and upload it to your Arduino.

```
// From https://github.com/bcomnes/315-lab-microcontroller/blob/master/code/pid_led_
↪set_serial/pid_led_set_serial.ino
#include <PID_v1.h>
const int photores = A0; // Photo resistor input
const int pot = A1; // Potentiometer input
const int led = 9; // LED output
double lightLevel; //variable that stores the incoming light level

// Tuning parameters
float Kp=0; //Initial Proportional Gain
float Ki=10; //Initial Integral Gain
float Kd=0;  //Initial Differential Gain

double Setpoint, Input, Output;  //These are just variables for storing values
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT); // This sets up our PDID␣
↪Loop
//Input is our PV
//Output is our u(t)
//Setpoint is our SP
const int sampleRate = 1; // Variable that determines how fast our PID loop runs

// Communication setup
const long serialPing = 500; //This determines how often we ping our loop
// Serial pingback interval in milliseconds
unsigned long now = 0; //This variable is used to keep track of time
// placehodler for current timestamp
unsigned long lastMessage = 0; //This keeps track of when our loop last spoke to␣
↪serial
// last message timestamp.

void setup(){
  lightLevel = analogRead(photores); //Read in light level
```

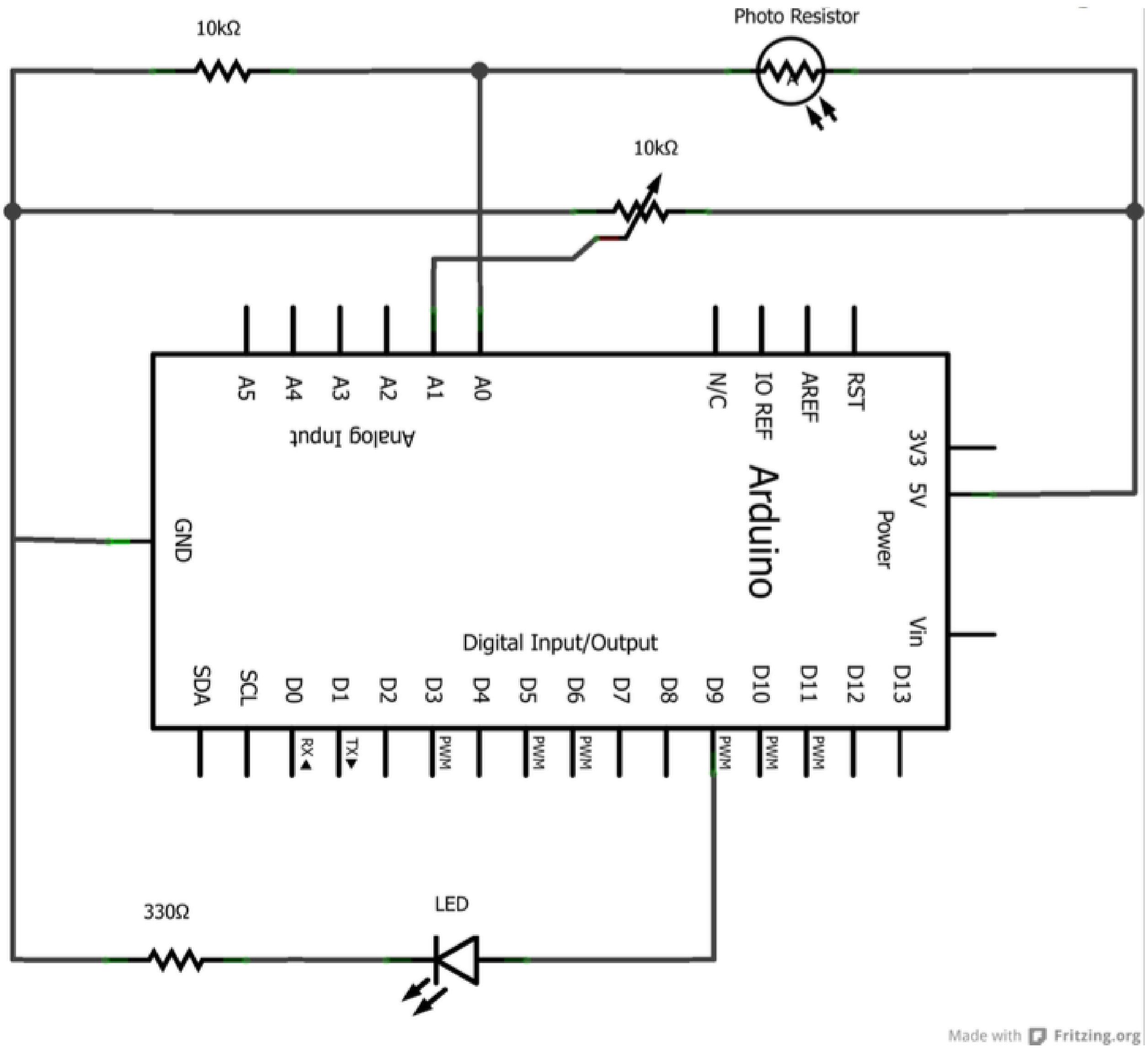


Fig. 1.81: Image: pdx.edu

```
  Input = map(lightLevel, 0, 1024, 0, 255); //Change read scale to analog out scale
  Setpoint = map(analogRead(pot), 0, 1024, 0, 255);  //get our setpoint from our pot
  Serial.begin(9600); //Start a serial session
  myPID.SetMode(AUTOMATIC);  //Turn on the PID loop
  mySetSampleTime(sampleRate); //Sets the sample rate

  Serial.println("Begin"); // Hello World!
  lastMessage = millis(); // timestamp
}

void loop(){
  Setpoint = map(analogRead(pot), 0, 1024, 0, 255); //Read our setpoint
  lightLevel = analogRead(photores); //Get the light level
  Input = map(lightLevel, 0, 900, 0, 255); //Map it to the right scale
  myPID.Compute();  //Run the PID loop
  analogWrite(led, Output);  //Write out the output from the PID loop to our LED pin

  now = millis(); //Keep track of time
  if(now - lastMessage > serialPing) {  //If its been long enough give us some info␣
↪on serial
    // this should execute less frequently
    // send a message back to the mother ship
    Serial.print("Setpoint = ");
    Serial.print(Setpoint);
    Serial.print(" Input = ");
    Serial.print(Input);
    Serial.print(" Output = ");
    Serial.print(Output);
    Serial.print("\n");
    if (Serial.available() > 0) { //If we sent the program a command deal with it
      for (int x = 0; x < 4; x++) {
        switch (x) {
          case 0:
            Kp = Serial.parseFloat();
            break;
          case 1:
            Ki = Serial.parseFloat();
            break;
          case 2:
            Kd = Serial.parseFloat();
            break;
          case 3:
            for (int y = Serial.available(); y == 0; y--) {
              Serial.read();  //Clear out any residual junk
            }
            break;
        }
      }
      Serial.print(" Kp,Ki,Kd = ");
      Serial.print(Kp);
      Serial.print(",");
      Serial.print(Ki);
      Serial.print(",");
      Serial.println(Kd);  //Let us know what we just received
      myPID.SetTunings(Kp, Ki, Kd); //Set the PID gain constants and start running
    }

    lastMessage = now;
```

```
    //update the time stamp.
  }

}
```

4. What are the initial values of the K_p, K_i, K_d gains? (3 pts.)

5. What happens as you change the set point of the system using the potentiometer? (2 pts.)

6. At a fixed set point, change the level of incoming light by shielding the setup with your hands and by increasing the light level using a flashlight. How fast does the system respond? (2 pts.)

7. Using the serial monitor, change the gain settings by sending three numbers separated by commas. For example to set $K_p = 2, K_i = 10, K_d = 0$ you would send `2,10,0`. Systematically vary the K_i setting with both K_p and K_d set to zero. Describe the effect this has. Why is it so? (4 pts.)

8. Set K_i to 10 and systematically increase K_p with K_d set to zero. Describe the effect this has. Why is this so? (4 pts.)

9. Set K_i to 10 and systematically increase K_d with K_p set to zero. Describe the effect this has. Why is this so? (4 pts.)

10. Does this system have any equivalent mass or inertial effects? (2 pts.)

11. What parameters seem to be the best for controlling the light level? Why do you think that is? (2 pts.)

Project Presentations

The course project is a large part of not only the grade in this course, but a way that you'll really internalize and learn the information we've covered. When your project is completed you will be required to present it to the class and to write a summary of what you did, why, and what you've learned. We would like to share your projects on the course website, so please inform us of any problems with that.

Oral Presentation

You will need to demonstrate your project to the class using the real hardware. Make a 10-15 minute presentation that covers what problem your project solves, what your design process was, any challenges you encountered, and the final results. You should conclude with a demonstration of your equipment. Remember to practice good public speaking skills! If you are looking for inspiration, checkout the book The Craft of Scientific Presentations: Critical Steps to Succeed and Critical Errors to Avoid .

Written Report

Documentation is the most important part of any project, it becomes more important as time passes. You will forget what you did last week, much less last year. Write a 2-4 page report that summarizes the problem you solved, challenges encountered, and the ultimate solution to the problem. This is your first line of defense against forgetting what you did and how you did it.

Online Bonus

We will add 10% to your written report grade as a bonus if you make the report online ready. This means you will need to format the report as a restructured text document, including the photos, links, etc. This is how the course website is generated and makes posting your report easier on the instructors and makes you learn some useful computing skills. The ReStructuredText Primer is a good resource to get started. Be sure to include a photo of yourself on the page. We will add an additional 5% to the written report grade if you submit your report as a pull request to the course repository.

Grading Rubric

Description	Max Points
Written Report	50
Oral Presentation	30
Project Works	20
Project fulfills the proposal	20
Total	120

Syllabus

GEOSC 597-003

Fall 2016

Tuesday, Thursday 12:05-1:20 PM, Deike 240

Instructors: Chris Marone, John Leeman

Offices: Deike 536, 439

E-mail: marone@psu.edu, jleeman@psu.edu

Course Description

Hands-on introduction to the principles of measurement, control and experimental design. Digital formats and representation, file structure, practical aspects of sensors. Introduction to transducers and digital representation of analog measurement. Design of reliable and effective measurements of physical variables including force, displacement, temperature, stress and strain. Analog to digital conversion, real-time data display, practical introduction to analog electronics including buffers and differential amplifiers, offsets and voltage dividers. Transducer calibration, instrument testing, and documentation. Basics of mechanical design for laboratory and field-based hardware, including communication with machine shops. Introduction to 3D printing and practical experience with 3D printers at Penn State.

Much of our work will be based on the Arduino platform via the Sparkfun Inventors Kit . Hands on classroom activities and class projects will use this kit. Students will develop approaches for signal measurement, experiment control, and data logging. A course research project will be completed with the inventor's kit.

Course Goals

1. To learn how to design and build basic mechanical and electrical devices for laboratory measurement work.
2. Gain hands-on experience using real equipment, sensors, and machines.

Course Materials

Text: We will be providing excerpts from a variety of texts. See the course website for a list of useful reference and learning materials.

Other Supplies: SparkFun Inventors Kit , Laptop

Laboratory Expectations

Each student is required to do his or her own lab. Discussion of the lab concepts amongst your classmates is encouraged. Late lab reports will not be accepted unless there are extenuating circumstances that have been brought to our attention and cleared prior to the deadline. If extenuating circumstances arise and have been cleared, the lab must be made up within a week of the missed class period.

Attendance

To learn from the lectures and activities you must be present. Attendance will count towards the class participation portion of the final grade.

If you need to miss class due to athletic or religious reasons please inform us as soon as possible, but by no later than one week prior to the missed class. We understand that sometimes you will need to miss a class due to illness or unavoidable circumstances (ex. a death in the family) and will not be able to provide the required one-week notice. Please let us know as soon as possible and arrangements can be made.

Grading

Exercises	40%
Final Project	50%
Class participation	10%

Grade Scale

A	92-100%
A-	89-91%
B+	86-88%
B	81-85%
B-	77-80%
C+	74-76%
C	65-73%
D	60-64%
F	<60%
X	Unsatisfactory (student did not participate)

Course Website

We will be using Canvas for posting grades and copyright class material (book chapter scans, etc.). The course material, topic summaries, labs, and other information is on the course website . Please check it regularly for announcements and to explore the embedded videos and content. Video recordings of the lecture will also be posted.

Accommodation of Students with Disabilities

The Office of Disability Services requests and maintains disability-related documents; certifies eligibility for services; determines academic adjustments, auxiliary aids, and/or services; and develops plans for the provision of academic adjustments, auxiliary aids, and/or services as mandated under Title II of the ADA Amendments Act (ADAAA) of 2008 and Section 504 of the Rehabilitation Act of 1973. A list of these services is provided .

Academic Misconduct

Cheating will not be tolerated under any circumstances. Cheating is unfair to your classmates and is an insult to curiosity and intellectual inquiry. In the context of this class, cheating includes using a clicker for another person, copying from another student's exam or assignments, plagiarism, and other activities. Students caught cheating will be reported to University officials and receive an automatic failing grade on the activity. General guidelines are covered here and in the Senate Policy 49-20 on Academic Integrity. This course follows the guidelines set out for the College of Earth and Mineral Sciences for offenses and appropriate punishments –please read these .

Plagiarism is cheating. This includes cutting and pasting material from sources on the web. All work and wording of assignments should be your own. Limited quotation of sources is acceptable and should be referenced accordingly. All written work will be subject to electronic plagiarism checking with Turn It In . Submitted computer code and other products will be automatically compared to ensure that two submissions are not the same.

Lectures

This class was taught during the Fall 2016 semester at Penn State in the department of geoscience. Here we have recordings of the lecture sessions and the lesson plans used for the course. This material should enable others to teach the course or take it online at any time.

Contents:

Course Ad

Geosc 597-3, 3 Credits Fall 2016

Tuesday & Thursday 12:05-1:20 PM, Deike 240

Introduction to the principles of experimental design, measurement, and control.

Content

Hands-on introduction to the principles of measurement, control and experimental design. Digital formats and representation, file structure, practical aspects of sensors. Introduction to transducers and digital representation of analog measurement. Design of reliable and effective measurements of physical variables including force, displacement, temperature, stress and strain. Analog to digital conversion, real-time data display, practical introduction to analog electronics including buffers and differential amplifiers, offsets and voltage dividers. Transducer calibration, instrument testing, and documentation. Basics of mechanical design for laboratory and field-based hardware, including communication with machine shops. Introduction to 3D printing and practical experience with 3D printers at Penn State.

Much of our work will be based on the Arduino platform via the Sparkfun Inventors Kit . Hands on classroom activities and class projects will use this kit. Students will develop approaches for signal measurement, experiment control, and data logging. A course research project will be completed with the inventor's kit.

Course Fee

$95 (for the Sparkfun Inventor's Kit)

Instructors

- Chris Marone
- John Leeman

Lectures 1/2 - Tools and Process Reading

Dates: 8/23/16, 8/25/16

Summary

The instructors and class are on travel to a conference. There will be no formal lecture or class session for these two dates.

Assignment

Read pages 1-39 in "Building Scientific Apparatus" by Moore, Davis, and Coplan. This part of the first chapter covers the basics of tooling and processes used when fabricating scientific apparatus. The basic properties of various materials are also covered. We will begin the lectures assuming that you have read these pages and will not be explaining their content. Bring any questions you have with you to class. **Due 8/30/16**

Lecture 3 - Course Introduction and Arduino Setup

Date: 8/30/16

Summary

In this lecture we will introduce the course and go over the requirements for the semester project. Examples of ideas for semester projects will be shown to help generate ideas for each student's project.

Class Plan

- Instructor introductions
- Go over course syllabus
- Introduce the idea of a semester long course project
- Show example projects
- Go over the Arduino hardware and the contents of the inventor's kit.

Activity

Have students setup their Arduino environment and complete the *Blinky 1.0* activity. Read the details about the Arduino on the *Getting Started with Arduino* page.

Assignment

Complete the *Project Brainstorm* activity. You should have 5 ideas for a project and have setup a meeting with the instructors by the due date. The earlier you begin on this, the better chance you have of completing a project that is useful to you in your research. **Due 9/6/16**

Media

`Lecture 3 Slides`

Instructor Notes

It is helpful to bring a Phillips screwdriver and set of diagonal cutters. The screwdriver is useful for helping students attach the Arduino to the base plate and the cutters are used to remove the zip-tie around the included jumper wires in the kit.

Lecture 4 - Tools and Shop Processes

Date: 9/1/16

Summary

This class will focus on the tools we use to build and maintain apparatus and how to use them safely. We will go over basic operations that can be done in the laboratory and machine shop.

Class Plan

- Ask students to go read/browse the hand and machine tools sections of the course webpage. We will not be going over much of that information, but it is all useful to know.
- Describe and show proper personal protective equipment (PPE) for various operations in the lab. Go over the shop/lab dress code.
- Introduce methods of part fabrication (machining, casting, sheet metal, welding, printing)
- Discuss the parts and purpose of a lathe
- Discuss the parts and purpose of a mill
- Discuss the parts and purpose of an EDM
- Discuss common materials used in equipment fabrication
- Show common fasteners (screws, bolts, rivets, pins, retaining rings)
- Demonstrate use of measurement tools (micrometer, steel rule, etc.)

Activity

Group students into groups of 2-3. Give each group a part design or physical part and ask them to complete the activity *Part Manufacturing Techniques*. Let them use the measurement tools to practice taking measurements of parts.

Assignment

Watch the threading video on the *Threading Activity* page. We will be doing it next lecture. This video will help you be prepared. Remember to wear shop appropriate clothing! **Due 9/6/16**

Media

`Lecture 4 Slides`

Lecture 5 - Machine Shop Field Trip and Lab

Date: 9/6/16

Summary

During this class we will visit an academic machine shop with many capabilities. The machinist will show the class the various machines and talk about how to design parts that are possible to manufacture and are cost efficient. After the field trip there is time for students to get their materials and work on the threading activity in the lab.

Class Plan

- Meet in the classroom and walk to the machine shop for a tour.
- Return to the lab to work on today's activity/assignment.

Activity

After returning from the machine shop tour, go to the lab. Students will receive the materials to complete the *Threading Activity* and have time to work on the assignment.

Assignment

Complete the *Threading Activity* and your parts ready to turn in during lecture. Contact us if you need assistance in the laboratory. **Due 9/8/16**

Media

No slides or video for this lecture - see the machine tools section of the course content for a summary of what was seen on the machine shop field trip.

Lecture 6 - Electronics Prototyping

Date: 9/8/16

Summary

In an effort to introduce students to a broad range of tools available for their projects, we will spend several lectures on digital and electronics topics before returning to mechanical design. This section is designed to be hands on as much as possible with the theory of electronics coming later when students can see how it will be applied. Prototyping electronics uses a wide variety of tools and construction techniques. In the class we will go over the prototyping techniques commonly used on the bench and in the lab.

Class Plan

- Talk about the various levels of permanence of prototypes (development, semi-permanent, permanent, production)
- Show breadboard and jumper construction technique
- Examples of Perfboard and point-to-point wiring (including strip/bus board)
- Show wire-wrap construction examples
- Talk about Copper board construction (skywire, deadbug, Manhattan)
- Show a custom PCB
- Cover basic soldering equipment and techniques

Activity

Using the breadboard and components in the inventor's kit, replicate the blink exercise using an external LED on a different digital output pin by following the *Blinky 2.0* activity.

Assignment

Complete the *Project Proposal* assignment. Think of this as writing the assignment for your final project. The idea and features must be settled now for you to be able to plan and complete the project on time. **Due 9/15/16**

Media

`Lecture 6 Slides`

Lecture 7 - Data Acquisition

Date: 9/13/16

Summary

In this lecture we will introduce principles of data acquisition, because to do science we need to collect data from our machines!

Class Plan

- Introduce the DAQ flow (physical system, transducer, signal conditioning, A2D, and computer)
- Give a high level overview of each block of the system
- Show common gotcha cases for data acquisition (aliasing, signal integrity, system loading)
- Show common systems used for DAQ (LabView/NI, LabJack, DATAQ, Omega, GDS)

Activity

Using your inventor's kit complete the *Flex Sensor Data Acquisition* activity.

Assignment

Complete the *Flex Sensor Data Acquisition* activity if you did not during class. Email us a plot (from the Arduino IDE, or one that you made) of your flex sensor flexing. **Due 9/15/16**

Media

`Lecture 7 Slides`

Instructor Notes

A few additional items need to be brought to class to accomplish the in-class activity:

- Multimeters to measure resistance of resistors and bend strip
- 47k resistors to use in the voltage divider

Lecture 8 - Microcontrollers and Design Patterns

Date: 9/15/16

Summary

There are many types of microcontrollers and peripherals. In this lecture we go over the basics of microcontroller interfaces and commonly found microcontrollers. We also discuss the software design patterns commonly found in software/firmware design.

Class Plan

- Go over each of the commonly found peripherals on a microcontroller (namely the Arduino the students have) such as I2C, SPI, PWM, and UART.
- Discuss what an interrupt is and why you shouldn't use them.
- Discuss system timers.
- Show what a function is and why we encapsulate and decouple code.
- Show common software architectures (polling, state machine, interrupt driven, producer-consumer)

Activity

Complete the *Building an XRD State Machine* activity.

Assignment

Complete the *Arduino Stoplight* activity. Email us your completed code. **Due 9/27/16**

Media

`Lecture 8 Slides`

Lecture 9 - Digital Logic and Representation

Date: 9/20/16

Summary

We have shown that almost all scientific data collection ends with a signal being digitized, operated on, and stored. In this class we will cover how data is represented digitally and basic digital logic operations.

Class Plan

- Show simple digital logic operations - AND, OR, XOR, NOR. (Do in class activity)
- Discuss representation of numbers/text as digital values with ASCII, Hex, Binary, and floating point representations.

Activity

In class, complete the *Digital Logic Operations* activity.

Assignment

Complete the *Digital Representation* homework activity. **Due 9/27/16**

Media

`Lecture 9 Slides`

Lecture 10 - Mechanical Drawing

Date: 9/22/16

Summary

We are going to switch gears back to mechanical design for about 2.5 weeks so students will have those skills for their projects. In this class we will show the principles of mechanical drawing so that you can communicate your part design to the machine shop or manufacturer. Good mechanical drawing makes the difference between you getting what you thought you designed and getting something totally different back from the shop.

Class Plan

- Introduce the purpose of mechanical drawing
- Show and demonstrate the basic drawing views (isometric, orthographic)
- Show proper dimensioning technique
- Show and demonstrate section and detail views

Activity

Complete the *Mechanical Drawing Activity* in class.

Assignment

Complete the drawing assignment if not completed during class. Parts must be returned with the drawing. Sign up for a free OnShape CAD account. **Due 9/27/16**

Media

`Lecture 10 Slides`

Lecture 11 - CAD and 3D Printing

Date: 9/27/16

Summary

Modern design often utilizes 3D CAD tools to help design assemblies with minimal unforeseen issues. In this lecture we will demonstrate the basic principles of 3D sketching/modeling and how to rapidly prototype those 3D designs with a 3D printer.

Class Plan

- Introduce the concept of sketches, extrusions, revolutions, and design intent
- Introduce the various forms of 3D printing and rapid prototyping (FDM, SLA, Laser Cutting, CNC machining)

Activity

Create a simple part or modify an existing part using the CAD platform of your choice (we recommend OnShape). Instructions are in the simple part design activity.

Assignment

Using the part you created in class, or another part, complete the *3D Printing Activity*. **Due 10/13/16**

Media

`Lecture 11 Slides`

Lecture 12 - Principles of Mechanical Design

Date: 9/29/16

Summary

Just like there are common software design patterns, there are best practices and tested solutions to many mechanical design problems. In this class we will cover some of the most common mechanical design principles and tools. With these tools you should have a good start on designing complex and useful parts and assemblies.

Class Plan

- Introduce degrees of freedom of a system
- Demonstrate basic techniques on kinematic design
- Show journal, rolling element, and linear bearings
- Introduce the concept of pre-loading
- Show torsion, tension, and compression springs
- Introduce flexures
- Show gears, chains, and belts and power transmission methods
- Discuss common materials used in laboratory and prototype fabrication

Activity

Look around online for materials and resources you will need for your project. Begin making a list of them to submit to us to order. Sign yourself up for catalogs from major suppliers.

Media

`Lecture 12 Slides`

Lecture 13 - Sealing

Date: 10/4/16

Summary

In almost every apparatus we are trying to keep different media separate. This could be sealing oil inside a hydraulic piston, sealing air inside a pressure vessel, or sealing in a vacuum chamber. In this class we will cover the most common sealing techniques used in laboratory equipment and how to safely implement them.

Class Plan

- Discuss the purposes of seals
- Introduce the IP rating system
- Cover gaskets, O-rings, D-rings, packed, and Bridgeman seals
- Discuss backup rings
- Discuss seal glands
- Show how varies threads seal

- Introduce vacuum fittings and seals
- Introduce basic epoxies

Activity

- Swagelok demo in the laboratory

Assignment

Make a parts list of what we need to buy for your project. **Due: 10/6/16**

Media

`Lecture 13 Slides`

Lecture 14 - Pressure Vessel Design

Date: 10/6/16

Summary

Pressure vessels are used to subject samples to in-situ conditions, to allow processes to happen in a vacuum, and to safely store/transfer energy. In this class we will discuss the safe operation of pressure vessels and how to design a pressure vessel that both withstands the required operating conditions and is not excessively over-engineered and wasteful of material.

Class Plan

- Introduce pressure vessels
- Show hazards and safety precautions
- Point to useful resources

Activity/Assignment

Complete the *Pressure Vessel Design Activity*. **Due 10/11/16**

Media

`Lecture 14 Slides`

Lecture 15 - Electronics Components and Schematics

Date: 10/11/16

Summary

A basic understanding of electronics is essential to work in the modern scientific lab. In this class we will cover the basic components: the resistor, inductor, capacitor, transistor, and diode. Students will also learn how to recognize components on a schematic diagram and read the connections between them.

Class Plan

- Introduce the basic concept of electricity
- Differentiate between conventional and electron current flow
- Introduce Ohm's Law
- Introduce Kirchhoff's Laws
- Show common schematic symbols and notation
- Discuss the resistor and it's properties
- Discuss the capacitor and it's properties
- Discuss the inductor and it's properties
- Discuss the diode and it's properties
- Discuss the transistor and it's properties

Activity

There are several in-class exercises included in the slides that students will perform to provide a formative assessment during the lecture.

Assignment

Pair up with someone and build a copy of the *Arduino Voltmeter* to use in completing the lab for Thursday. **Due: 10/13/16**

Media

`Lecture 15 Slides`

Lecture 16 - Lab: Basic Components and Circuit Analysis

Date: 10/13/16

Summary

Last class the basic electrical components were introduced. In this class we devote time to a laboratory activity that explores the properties of the basic components and introduces basic circuit analysis.

Class Plan

Complete the *Component Basics* activity during class time.

Assignment

Component Basics activity. **Due: 10/18/16**

Media

`Lecture 16 Slides`

No video for lab activity day.

Lecture 17 - Transducers

Date: 10/18/16

Summary

Transducers are how we can turn physical quantities into mechanical and electrical signals that we can more easily measure. In this class we will go over transducers to measure temperature, force, displacement, pressure, and more.

Class Plan

- Discuss the role of a transducer
- Show the difference between analog and digital transducers
- Discuss thermistors, thermocouples, and RTDs for temperature measurement
- Discuss DCDTs, LVDTs, and other displacement transducers
- Discuss strain gauges for force/strain measurement
- Discuss pressure transducers

Activity

Complete the *Thermistor Calculation Activity*. **Due 10/20/16**

Assignment

Find an object that you will instrument for the strain gauge activity. This could be a piece of rock, a coffee mug, or anything else you can think you. Remember that you must be able to mount the strain gauges to it, so a smooth surface is necessary. **Due: 10/25/16**

Media

`Lecture 17 Slides`

Lecture 18 - Calibration

Date: 10/20/16

Summary

To get a physical quantity from a transducer, we need to calibrate it. In this class we will cover calibrations and how to perform them. We will then go to the lab and perform a calibration.

Class Plan

- Discuss what a calibration is and why it is necessary
- Discuss standards and transfer standards
- Show good calibration practices

Activity

Complete the *Calibration Activity*. **Due 10/25/16**

Assignment

Finish the calibration activity if not completed in class and don't forget to bring your object to strain gage next class!

Media

`Lecture 18 Slides`

Lecture 19 - Strain Gauges

Date: 10/25/16

Summary

Last class we discussed transducers and calibration. This period we will build a strain gauge bridge to measure deformation of an object. This project will be built upon for the next several labs.

Class Plan

Complete the strain gauge activity during the class period using the facilities in the lab.

Assignment

Complete the *Strain Gage Activity*. Show us your strain gauged object working when hooked up to a multimeter. **Due: 10/27/16**

Media

Lab day - no lecture slides or video.

Lecture 20 - Amplifiers : Part 1

Date: 10/27/16

Summary

Many of the signals we want to measure when doing research are very small and need to be amplified. The most common tool in the engineer's tool belt is the operational amplifier. In this class we will introduce the ideal operational amplifier and show how to use it as a buffer, amplifier, and comparator.

Class Plan

- Discuss the need for amplifiers in the laboratory
- Introduce the ideal amplifier
- Introduce the concepts of gain and open-loop gain
- Show a voltage follower (buffer) circuit
- Comparator
- Inverting amplifier
- Non-inverting amplifier
- Differential amplifier

Activity

Complete the included formative assessments during the lecture.

Assignment

Complete the designing a differential amplifier activity. **Due: 11/1/16**

Media

`Lecture 20 Slides`

Lecture 21 - Amplifiers : Part 2

Date: 11/1/16

Summary

We will continue our discussion of amplifying small signals. We will go over the integrator, differentiator, and summing amplifier circuits. We will also show how to design a multivibrator circuit. Finally, we will introduce the instrumentation amplifier and go over some common problems encountered with using amplifiers in the laboratory.

Class Plan

- Integrator amplifiers
- Differentiator amplifiers
- Summing amplifiers
- Multivibrator design
- Instrumentation amplifiers

Activity

Complete the included formative assessments during the lecture.

Media

`Lecture 21 Slides`

Lecture 22 - Lab: Amplifiers

Date: 11/3/16

Summary

We have spent two class periods going over amplifiers – it is an important topic for an experimentalist! In this class you'll build an amplifier circuit of your own and use it to amplify the small signals produced in your strain gauge bridge circuit.

Class Plan

- Using parts provided by the instructors, construct an amplifier for your strain gauge object.

Assignment

Complete the amplifier activity. **Due: 11/8/16**

Media

Lab day - no lecture slides or video.

Lecture 23 - Signal Integrity and Input Impedance

Date: 11/8/16

Summary

Any time we amplify, process, or digitize an electrical signal there will be noise. One of the greatest challenge in research equipment is making sure that the signals remain as pristine and low noise as possible. In this class we will discuss how to keep your signals clean as well as how input impedance works.

Class Plan

- What is signal integrity?
- Crosstalk
- RFI
- Ringing
- Filtering
- Shielding
- Layout considerations
- Input impedance
- Impedance matching

Activity

- Show the effects of impedance mis-match with an oscilloscope, function generator, and termination resistor. Ask students to explain what we are seeing on the display.

Media

`Lecture 23 Slides`

Lecture 24 - ADC/DAC

Date: 11/10/16

Summary

We have already talked about the digital representation of voltage and even used an analog-to-digital converter to measure voltages with the Arduino. In this class we're going to dive into how analog-to-digital converters work and the advantages of different converter architectures. We will also introduce digital-to-analog conversion – generating a voltage from a digital representation.

Class Plan

- What is ADC and DAC?
- Charging a capacitor as an ADC
- Resistor divider/comparator example of ADC
- Flash ADC

- Successive approximation ADC
- Multi-slope ADC
- Sigma-Delta ADC
- PWM DAC
- R2R DAC

Media

Class did not meet on this day.

Lecture 25 - Filtering

Date: 11/15/16

Summary

Filters are used to isolate portions of a signal that we are interested in and to reduce unwanted noise. In this class we will introduce the basic types of filters (hardware and software) available and provide resources to learn more about each.

Class Plan

- Low pass filter
- High pass filter
- Bandpass filter
- Bandstop filter
- Passive filters
- Active filters
- FIR/IIR filters

Media

`Lecture 25 Notes`

Lecture 26 - Control Systems/Loops

Date: 11/17/16

Summary

We often need our apparatus to maintain some set of conditions. It could be controlling the temperature, pressure, humidity, of speed of some process. Taking a noisy real world signal and controlling an apparatus is a complex process. We will introduce the elements of PID control and show practical examples of control loops.

Class Plan

- Why do we need control systems
- Open-loop vs. Closed-loop control
- Logic control
- On-Off control
- Linear control
- Proportional control
- PID control

Media

`Lecture 26 Notes`

Lectures 27/28 - Projects

Date: 11/22/16, 11/24/16

Summary

There are no classes for the Thanksgiving holiday. Use this time to work on your semester projects. It is only about 2.5 weeks until the final project presentations! Make sure you are including the time required to have any parts 3D printed, machined, etc. The end of the semester is a busy time for the shops, so plan ahead!

Lecture 29 - PID Controllers

Date: 11/29/16

Summary

This class period is devoted to a lab exercise that allows students to explore the PID control loop. Students will use a PID controller implemented on their Arduino Uno to control a real system and tune the parameters of the control loop.

Class Plan

- Spend the entire period working on the PID controller activity.

Activity

Complete the *PID Controllers* activity. **Due: 12/1/16**

Media

Lecture slides and video will be posted here after the lecture is given.

Lecture 30 - Hydraulics/Pneumatics

Date: 12/1/16

Summary

Many pieces of laboratory apparatus use a hydraulic or pneumatic system to manipulate samples, apply load, or move parts of the machine. In this class we will introduce the basics of these systems including their mechanical advantage, operating principles, and hazards/maintenance.

Class Plan

- Why use hydraulics and pneumatic systems?
- Mechanical advantage
- Parts of a pneumatic system
- Parts of a hydraulic system
- Design considerations

Activity

Complete the calculating hydraulic force activity. **Due at the end of class**

Media

Lecture slides and video will be posted here after the lecture is given.

Lectures 31/32 - Project Presentations

Date: 12/6/16, 12/8/16

Summary

Each student will show their semester project in a 15-minute presentation. Presentations should cover the problem being solved, the design considerations, and the implementation of the solution. If a demonstration is possible, it is encouraged!

Class Plan

- Student presentations
- Complete course review

Media

Video will be posted here after the lecture is given.

Resources

Here we've collected links to help you find suppliers for parts, useful websites, YouTube channels, and other resources you will likely find interesting and useful. Please feel free to suggest additions through GitHub issues, pull requests, or an email to the instructors.

Contents:

Electronics Suppliers

- Adafruit
- All Electronics
- Digikey
- Mouser
- Newark/Element 14
- Pololu
- Servo City
- Sparkfun Electronics
- Wicked Device

Mechanical Suppliers

- Grainger
- McMaster-Carr
- MSC
- Servo City
- Side Cuts

YouTube Channels

- Adafruit
- Applied Science
- Chris Gammell (Contextual Electronics)
- EEVBlog
- Mike's Electric Stuff
- NYC CNC
- Sparkfun Electronics
- This Old Tony
- W2AEW

Books

Student Projects

Student projects will be shared here as reports, video, and slides.

Contents:

Automatic DCDT Calibration Device - Chas Bolton

Project goals

My semester project was aimed at improving our calibration techniques for the DCDTs. Currently, the technique for calibrating the DCDTs is done manually and is quite time consuming. In particular, to calibrate a DCDT one must mount the core to a vertical measuring device and clamp the outer part of the DCDT to a stationary ring stand. The user then moves the core in increments of 1mm or less throughout its entire voltage range. After one 1mm of displacement, the user must measure and record the voltage at this position. Again, this process will continue for the entire voltage range of the DCDT.

My goal was to improve the efficiency of this technique by creating a device that would automatically move the DCDT throughout its voltage range. Therefore, after the device is setup properly the only task for the user would be to simply record the voltage measurements. To carry out this task, I designed a holding bracket that would hold the DCDT and the bracket itself moves via a stepper motor that is controlled from a stepper motor driver and Arduino.

Design

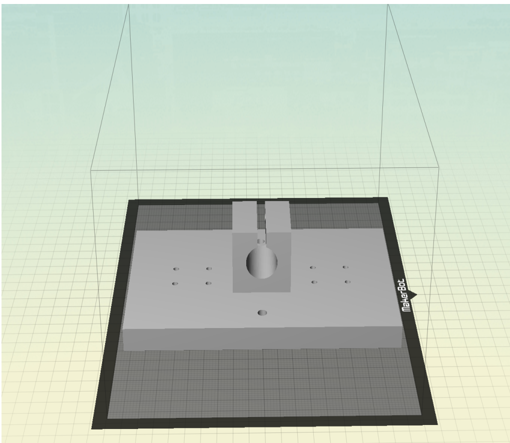
The first major design task was deciding how to hold and move the DCDT. To solve the issue of holding the DCDT, I came up with the idea of creating a holding bracket via 3D printing that would consist of a flat plate with a cylindrical holder mounted in the center of the plate that the DCDT could fit into.

To move the DCDT move up and down, I decided to mount the bracket assembly to two aluminum rods and allow the bracket to slide up and down the rods by a stepper motor, belt and pulley. To mount the rods to the DCDT bracket, I decided to use a set of linear ball bearings which would screw on to the bracket and would allow the bracket to move smoothly up and down the rods. The aluminum rods themselves would be mounted to a plywood frame.

The next step was deciding how to attach the belt to the bracket and how to mount the pulleys. I decided that the pulley could easily be mounted at the top of my frame via a U-bolt. However, once I started the assembly process I slightly modify this part. To mount the belt to the bracket, I decided to screw two I-bolts through the bracket and parallel with the cylindrical holder on the bracket. By using I-bolts, I could loop the belt through the I-bolts and tie them back into each other.

The final design task was deciding what material to build the frame with. I decided on using plywood, as this would be easy to work with and to mount things on. When deciding on the dimensions of the frame, I was primarily concerned with the height dimension due to the fact that the frame needed to be high enough to calibrate the DCDTs with long cores. A frame height of 2 feet was sufficient enough to accommodate this criterion.

Once I had an idea of how the project was going to work and what was needed to complete the project, I compiled a list of parts. The parts needed to complete my project consisted of stepper motor, a driver to control/drive the stepper motor, two aluminum rods, two linear ball bearings that the DCDT holder could mount to and slide up and down the rods, a belt, two pulleys, U-bolt, 2 I-bolts, and plywood.



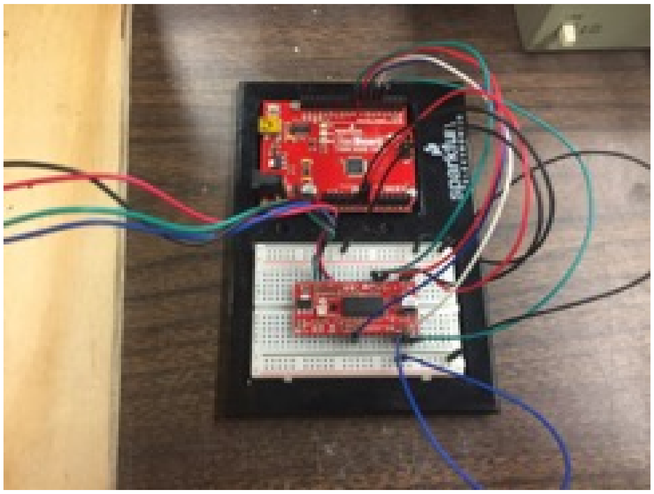
Assembly

The DCDT bracket was the first part that I sought out to design. At first, I considered having the machine shop design the bracket out of steel. However, I soon realized that this would be too expensive and unnecessary for simply moving a DCDT up and down. Therefore, I decided I would make my DCDT holding bracket via 3D printing. Before creating the bracket in On-Shape, I hand sketched all the components of the bracket along with their exact dimensions. Once this was completed I created the bracket in On-Shape and 3D printed the assembly.

The next object I built was the frame. To do this, I used a jigsaw to cut the correct dimensions of

the plywood and then assembled a simple frame consisting of two base plates and two side panels. The top and bottom pieces of plywood are used for mounting the stepper motor, pulley and aluminum rods. I mounted the aluminum rods in the center of the frame roughly 11 centimeters apart by drilling two holes through the top piece of plywood and two holes in the bottom piece of the plywood. The top holes are drilled all the way through the plywood for the aluminum rods to slide through, while the bottom holes are drilled only half way into the plywood so that the aluminum rods can be held securely in place.

The next task was to power my stepper motor and driver. I first mounted my driver with a set of headers to the breadboard of the Arduino and powered it up through an external power supply.



However, after doing this the voltages coming out the driver was inconsistent so I decided to solder the headers onto the driver. After soldering the headers on to the driver, the voltage readings come out of the driver were much more consistent. Once the driver seemed to be working properly, I modified an Arduino code such that I could move my motor in precise increments of 1mm and stop for 5 seconds and then move another 1 mm. After a total of 30 mm, the motor would then switch directions and move under the same conditions. To calculate the number of steps the motor must undergo for 1 mm of displacement, I used the following formula. Total number of steps per revolution/ total number of teeth on pulley times the pitch. Thus, for my assembly this equates to (400* 8)/2. The factor of 8 is due to the 1/8 micro stepping feature of the driver.

Once the motor was running properly and frame was built, I began working on connecting everything together to see if the project was actually going to work. I mounted my upper pulley to the top piece of plywood via a screw and two metal holding brackets. After the

belt and pulleys were properly assembled, I was able to run the first test to see if the project was actually going to work.

The first few trial tests turned out to be unsuccessful with making the bracket move. I began looking for issues with the design and noticed that the major problem was associated with the belt slipping on the pulley. My initial thought was that I was not supplying the motor with enough power. Therefore, I altered the power supply voltage and regulated the voltage coming from the potentiometer on the driver several times, but all to no avail. After assuming that it was not an electrical issue, I assumed that my problem could be a mechanical issue. Furthermore, I noticed that my belt was very loose and this might be the main source of error. After tightening the belt, the bracket began to move! However, it was still slipping and only moving in very small increments. Once I found this out I knew my problem was purely a mechanical issue. After thoroughly looking over everything on my frame, I noticed two other problems in addition to the loose belt. There was a slight misalignment between the two linear ball bearings that were mounted on the DCDT bracket and the top pulley mounted on the frame was not rotating smoothly and freely. Once I fixed these issues, the system started moving smoothly and the bracket was able to move up and down the aluminum rods in a continuous motion.

Currently, the automatic DCDT calibration device is able to move in smooth increments of 1mm, stop for 5 seconds and move another 1 mm. This process continues for 30 mm and the motor then switch directions. The idea of moving 1 mm is to ensure the voltage readings of the DCDT is linear with respect to displacement. The 5 second wait time is included to allow the user enough time to record the voltage measurements at a particular position. And the 30 mm, is around the average range of a typical DCDT. The idea of switching the motors direction is implemented make sure there is no hysteresis effect associated with the DCDT. All of these parameters can be changed by the user, and for a particular DCDT. For instance, not all DCDTs will need the full 30 mm range and this number can be increased or decreased. In addition, the user may need to increase the time it takes to read and record a voltage measurement and this can be done by simply changing the code to pause for 10 seconds.

Problems/Future work

The main problems that I encountered while working on this project were associated with the assembly process as mentioned above. If I were to continue working on this project to further its improvement there are a few modifications, I would make. First, I would redesign the upper pulley and belt system. As of right now, the belt has too much slack. I think the system would move much more smoothly if the belt was tighter. This could be done by implementing a spring the pulley/belt assembly. Also, I would consider reducing the size of the top and bottom portions of the frame and perhaps reducing the height of the side panels. As of right now, the frame is burdensome to move around and to work with.

Media

`Slides (PDF)`

`Slides (PPTX)`

Lab-scale Gas Monitor System - Long Fan

Problem

During the underground mining process (room and pillar system or long-wall mining system) within gas-containing coal seams, gas control with an effective ventilation system is vitally important for safety of workers and assessments. Ventilation system monitor plays a key role in giving an instant alarm of potential risks, such as the gas (methane) overrunning or blasting, and unstable ventilation supply caused by other potential disasters (roof falling, fire). Although methane and air velocity monitors has been developed and widely applied in underground coal mines, limited amounts of monitors are placed in limited locations due to its high cost and huge data acquisition in such a large scale. A scaling mining system model is built in lab and the ventilation system is built in to simulate the actual ventilation system. In order to monitor the air flow characteristics and the gas (methane) concentration distribution of this scaling ventilation system, lab-fit multifunction transducer is necessary to monitor the real-time air velocity and methane concentration at different positions of this system. Accordingly, when the monitored data exceeds a threshold we set, a light signal can be triggered as a feedback for disaster warning purpose. Thus, we'd like to be able to collect the real-time methane data in the scaling ventilation system so that it can be used as a technical support for a more efficient monitor distribution.

Solution

A methane sensor is going to be applied on methane concentration monitor. For methane concentration, when it exceeds 200 ppm, the signal light will be triggered to get people warned. The Arduino, LED and display in the SparkFun Kit and some analog components will be used. The recommend circuit for the methane monitor can be easily build based on the datasheet, and then, all the components will be hooked to the Arduion, as well as the LED. The data will be showed on the display.Two different color LEDs will shift when the methane concentration exceeds the value we set. A small USB battery pack is needed for power supply.

Resources

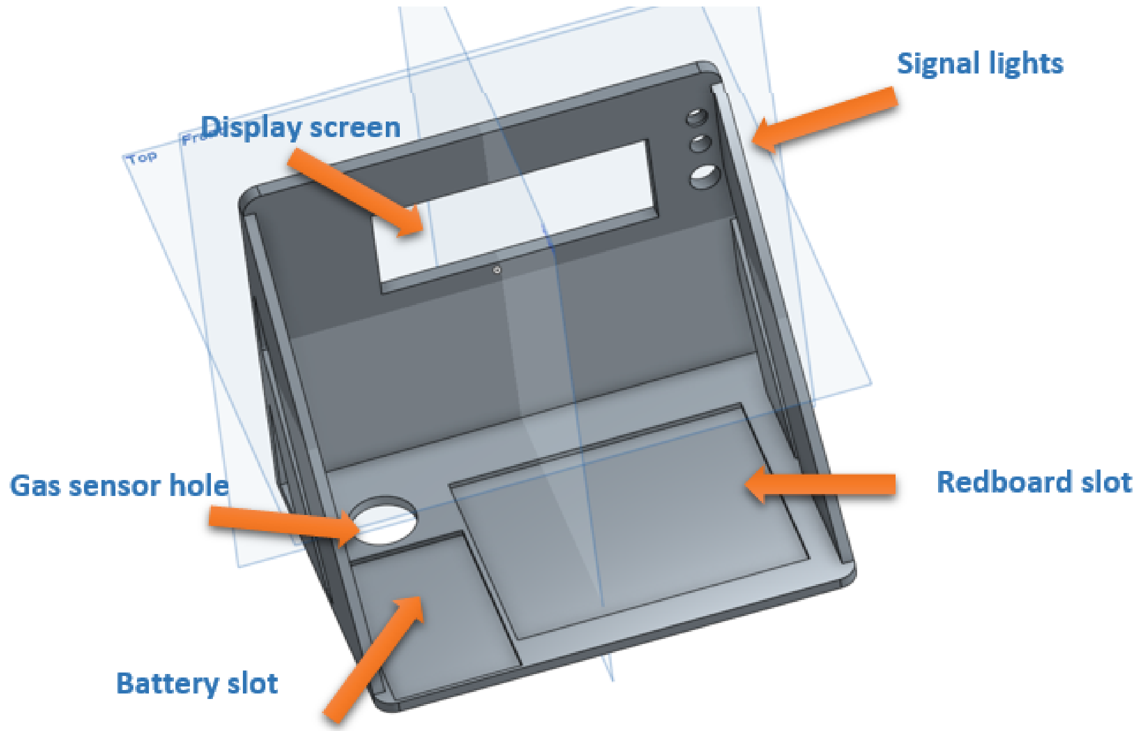
Soldering components may be needed and hand-tools in lab will be used for basic resembling and wire connection of sensor. No other machining or electronics resources will be necessary.

Items needed

Quantity	Item
1	Sparkfun Redboard
1	Breadboard
1/bunch	Jump wire
1	Methane sensor
1	Gas Sensor Breakout Board
1	USB Battery Pack
1	SparkFun USB LiPoly Charger
1	JST Right-Angle Connector
2	Display
1	Potentiometer
2	LED
2	10k Ω resistor
1	300 Ω resistor
1	3D Printed Bracket

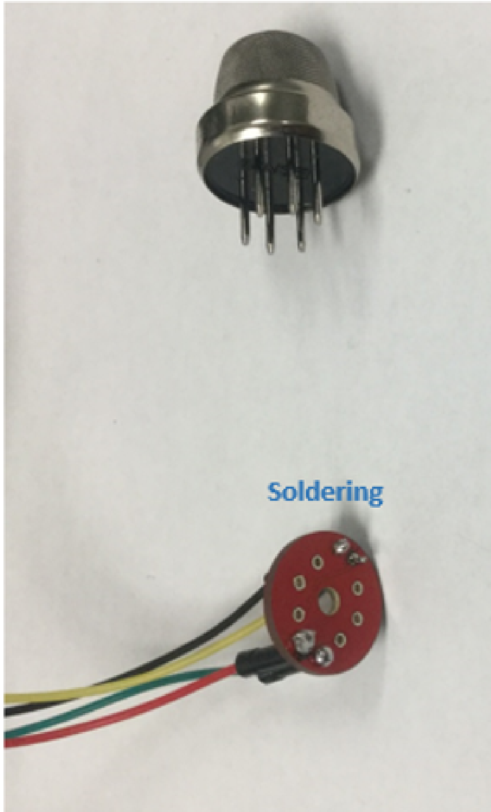
Assembling

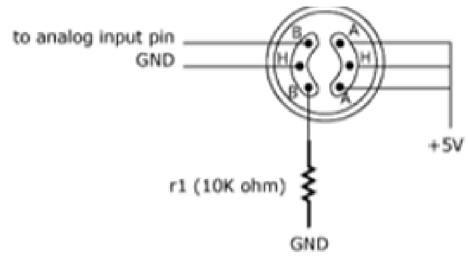
Since this gas monitor is going to be applied in lab, assembling is necessary to make it compact and easily built-in. A 3D printing bracket is designed to support all these components and to serve as a display and signal panel. Plus, the bottom of the bracket could be connected with the ventilation model through that gas sensor hole. 3D design of the bracket is shown below.

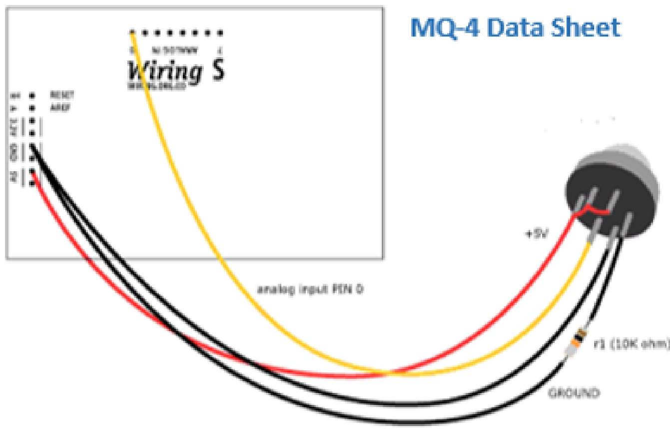


Gas Sensor wiring

A breakout board and a 10k Ω resistor are used to wire the gas sensor, the figure below shows the gas sensor and breakout board wiring.

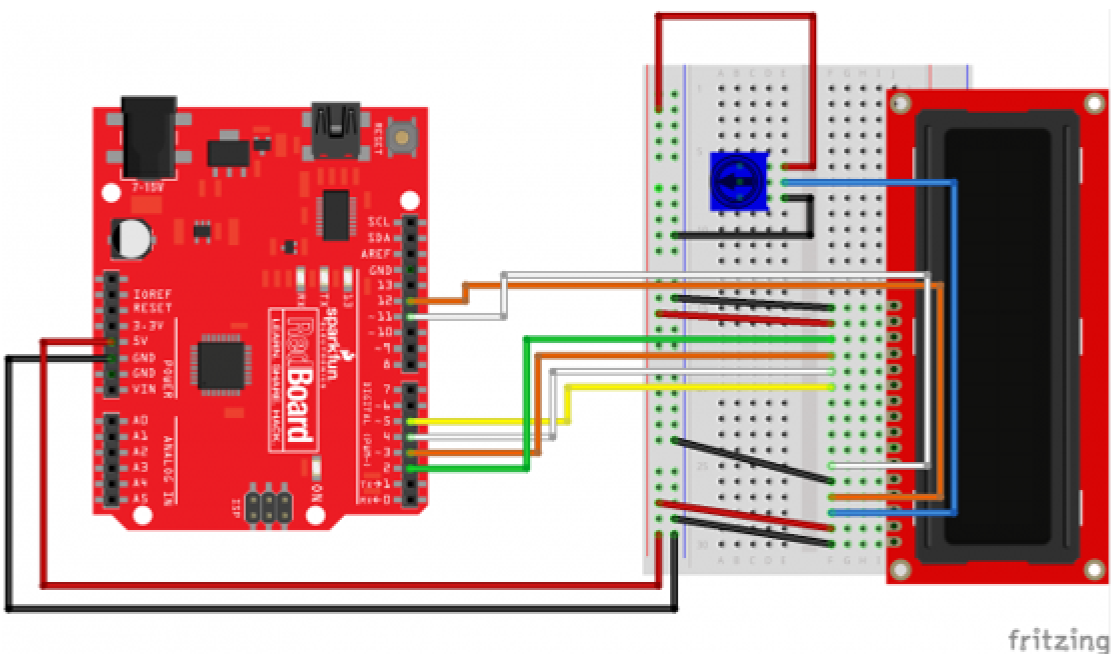




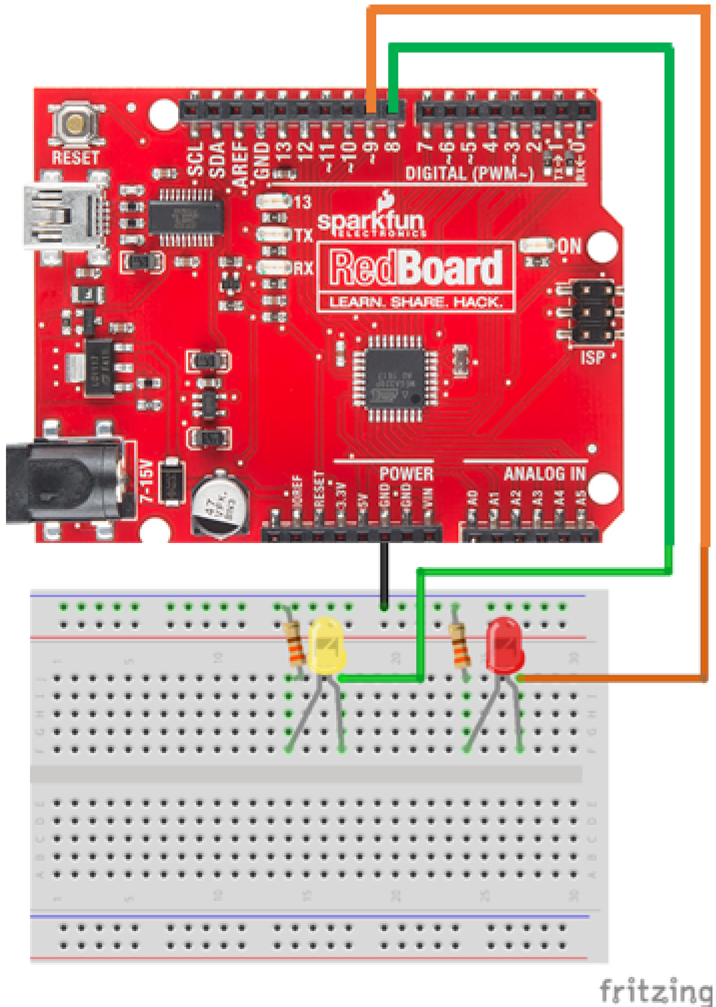


Led display wiring

A potentiometer is used to control the screen brightness. The wiring is shown below.

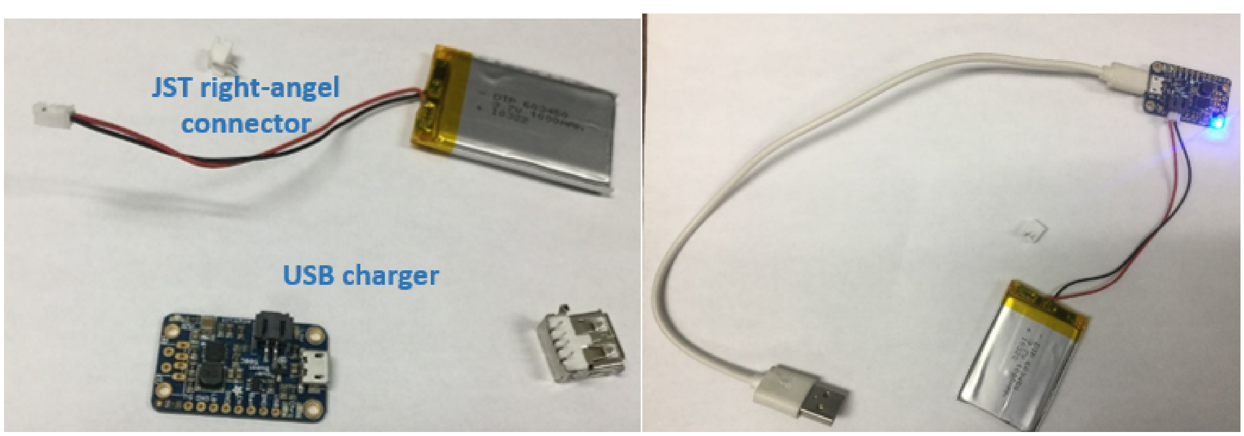


Signal lights wiring



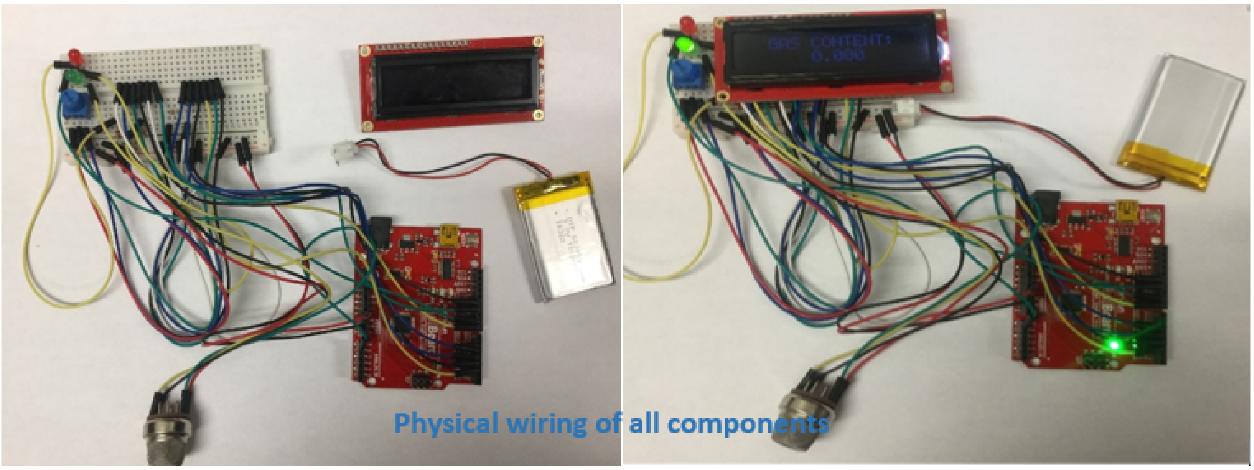
Battery wiring

100 mA, 3.7v Lipoly battery is used and a JST right-angel connector is used to connect the battery to the breadboard. Also, a USB Lipoly charger (the dark green component) can consistently supply power.

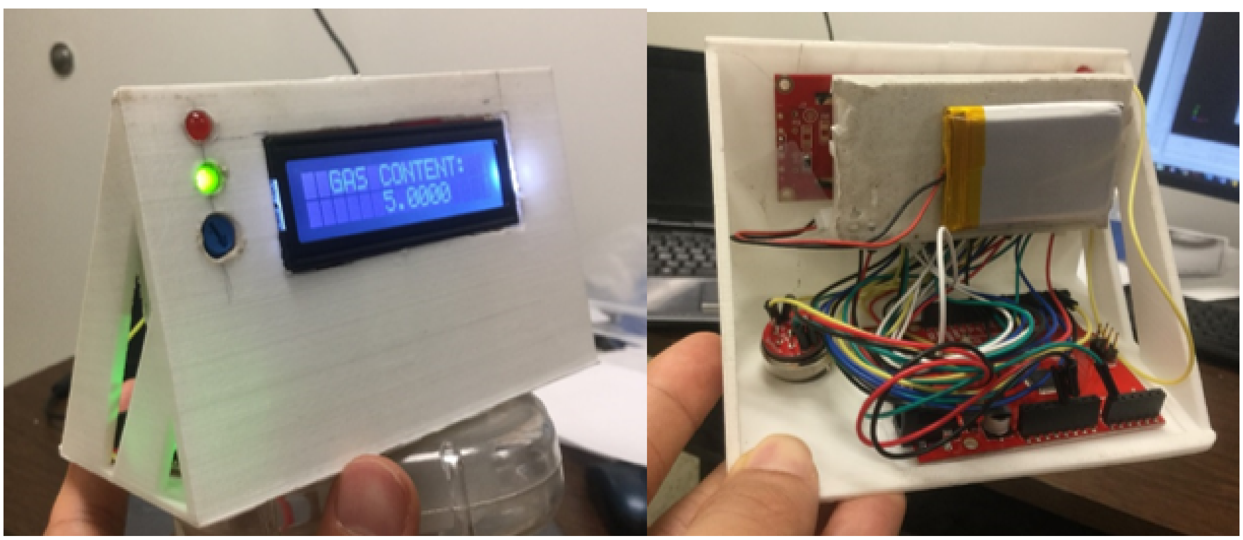


Testing

Hooking up all the components and test the performance of the display screen and gas sensor.



Final project



Challenges

1. The gas sensor is environmental sensitive. It seems when I do the test in different rooms, the performance will be a little different. When you keep your hands close to the sensor, the screen values will also change. I believe a more stable sensor will be selected in the future lab application.
2. From the data sheet, the gas sensor needs preheat 24hrs. It is kind of a long time and the sensor will get really hot. In the lab test, when I preheat the sensor for 3-4 hrs, most of time, it works stably. However, when I power on the system with this 3.7 v battery, it will take a really long time to preheat and we have to consider the duration of the battery although we have a battery charger.

Media

`Slides (PDF)`

`Slides (PPTX)`

Code

```
#include <LiquidCrystal.h>
#define NUM_TO_AVERAGE 500
LiquidCrystal lcd(12,11,5,4,3,2);
int sensorValue;
long a0_value, a1_value, a2_value, a3_value;
float a0_gascontent, a1_gascontent, a2_gascontent, a3_gascontent;
void setup()
{
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  Serial.begin(9600);

  lcd.begin(16,2);
  lcd.clear();

  lcd.print("   GAS SENSOR");
  delay(2000);
  lcd.clear();
}

void loop()
{
  sensorValue = analogRead(0);        // read analog input pin 0

              if (sensorValue < 200)      // if there is little or no gas detected display blinking green light
              {
                  digitalWrite(8, HIGH);
                  digitalWrite(9, LOW);
                  delay (500);
              }
               else
              {
               if (sensorValue >= 200)    // As the value of the gas measurement over 200, trigger the red light
                  {
                    digitalWrite(8, LOW);
                    digitalWrite(9, HIGH);
                    delay (500);
                  }
              }
  a0_value = 0;
  for(int i=0; i<NUM_TO_AVERAGE; i++){
    a0_value += analogRead(0);
  }

  a0_gascontent = a0_value / NUM_TO_AVERAGE;

  lcd.setCursor(0,0);
  lcd.print("  GAS CONTENT:");

  lcd.setCursor(0,1);
  lcd.print("      ");
  lcd.print(a0_gascontent, 3);

  Serial.print("#1 Gas Cont=");
```

```
Serial.print(a0_gascontent);
Serial.print(sensorValue);
Serial.print("#2 Gas Cont=");
Serial.print(a1_gascontent);
Serial.print("\n");
```

Syringe Pump for Injecting/Refilling Fluid - Yi Fang

Abstract

This report presents a process of identifying a realistic problem from daily experimental practice, designing the prototype and manufacturing the equipment for solving the realistic problem. A syringe pump was designed for long-term fluid injection/refill function. The syringe pump system is composed of structure elements, performing elements and control system. The structure elements are designed by free Cloud CAD suite Onshape and 3D printed by MakerBot at Penn State Library. The performing hardware (stepper motor, syringe and tubing) is purchased to perform injection/refill functions and the control system is designed using Arduino to control the hardware to inject or refill. The syringe pump prototype can be used to inject fluid and refill the fluid without unplugging the tubing and is suitable for cycled long-term injection/refill purpose.

1. Identification of Problem

1.1 Realistic Problem

Long-term fluid injection is vital for the success of in-house laboratory experiment. It is essential for pump to perform continuous injection. However, most pumps used in the lab experiments are commercial with limited volume (100 to 500 ml), in such a case, fluid injection may be disrupted because the tubing has to be unplugged from the running system and connected to reservoir for refilling (**Figure 1**).

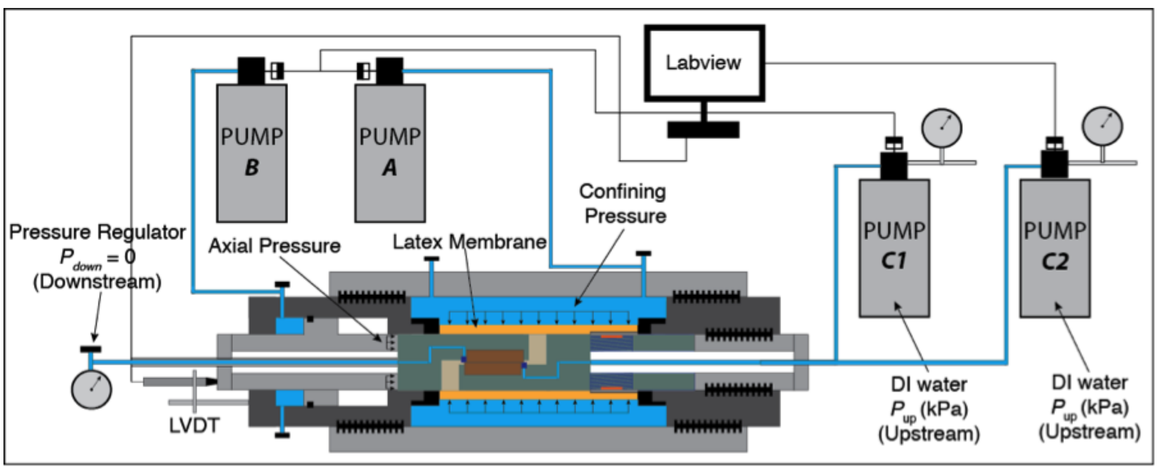


Figure 1. Schematic graph showing realistic problem: two pumps (C1 and C2) alternatively inject/refill the fluid.

1.2 Prototype and Simplification

The realistic problem can be solved by programming via Labview to accommodate two pumps alternatively injecting and refilling. To achieve the goal of this class - learn, explore and practice, a prototype of the realistic problem is developed to accomplish similar targets. Ideally, I would like to design a pump injection system composing of two syringe pumps that allow injection and refill acting simultaneously **(Figure 2a)**. However, to reduce the budget, I performed a necessary simplification in which the number of pump is reduced to one and the injection/refill switch is performed by accommodation of one-way valves **(Figure 2b)**.

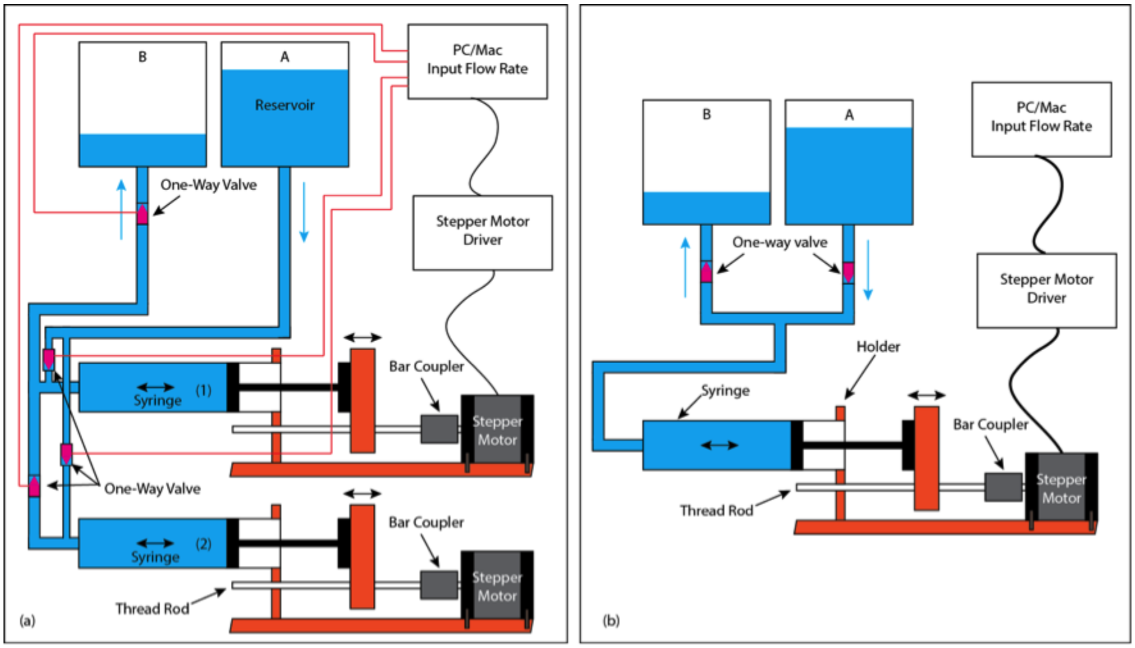


Figure 2. **(a)** Ideal version of syringe pump system as analogue to realistic problem **(b)** Simplified prototype of syringe pump system

2. Method

2.1 Pumping Hardware and Structure Elements

The pumping hardware is composed of a plastic syringe (20 ml), a stepper motor, a threaded rod with 4 nuts, a rod coupler, two one-way valves, one three way, a plastic tube and two plastic bottles, which are illustrated in **Figure 3** and the budget of each component is listed in **Table 1**. The hardware is then assembled with the structure elements (**Figure 4** and **5**). The structure elements were designed by an online open-source CAD (Onshape) and were 3D printed by MakerBot at Penn State Pattee Library.

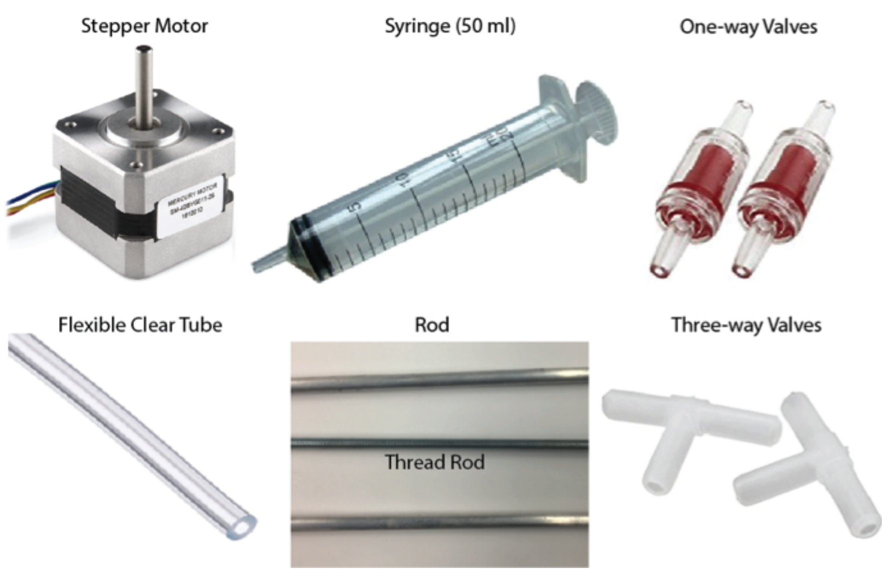


Figure 3. Major hardware component of syringe pump system

Table 1. Material and Budget

Quantity	Item	Cost
1	Syringe with tubing	$8.97
1	Stepper Motor	$14.95
1	Body holder	$0
1	Driving rod	$0
2	Supporting rod	$0
1	Two way tube	$5.67
1	Plastic tank	$5.0
2	Check valve	$24.16
1	Three way tube joint	$0
1	Wood block	$0
1	Rod Coupler	$6.0
1	Stepper Motor Driver	$14.0
1	Thread Rod	$6.0
1	Rod	$0.0
2	Nut	$0.0
Total		$84.75

Note: Those labeled as $0 are available in lab.

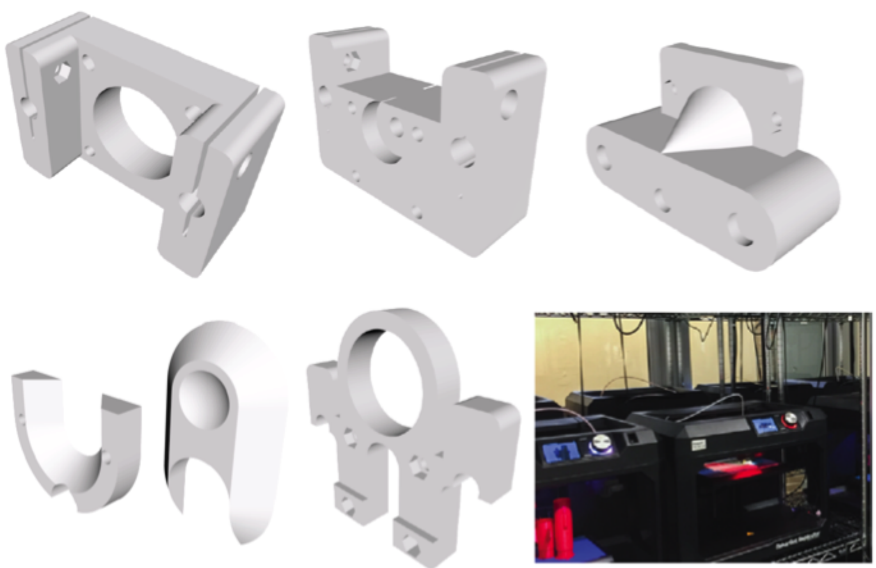


Figure 4. 3D geometry of structure elements

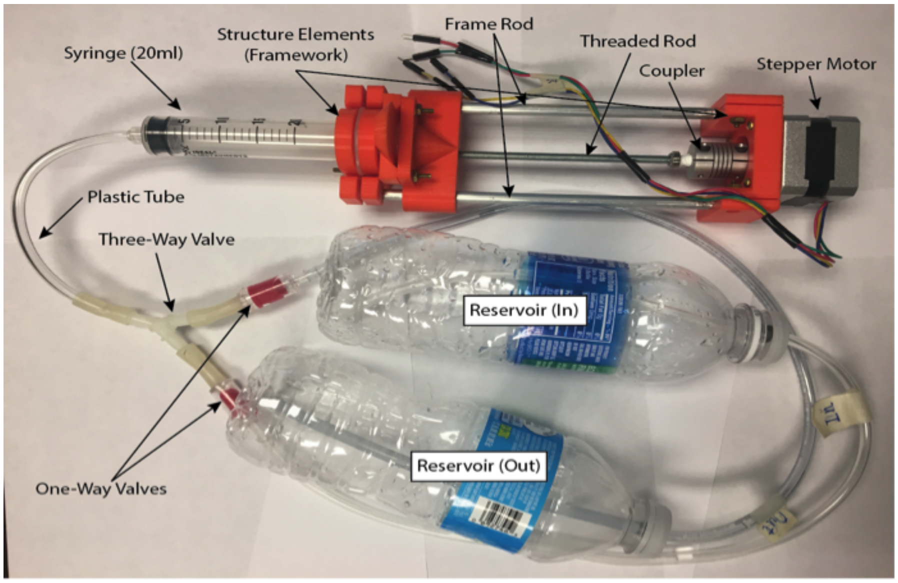


Figure 5. Assembled syringe pump system

2.2 Control System

The control system is composed of three parts: Arduino Redboard, Stepper Motor Driver and Arduino code (**Figure 6**). The Arduino code is first programmed and then uploaded to Arduino Redboad and Stepper Motor Driver. The commands are input via series monitor in Arduino software tools and are sent to Stepper Motor to perform the functions such as move forward or backward to control injection or refill of syringe. In this system, we developed four injecting flow rates (i.e., 0.585ml/s, 0.293ml/s, 0.146ml/s, and 0.074ml/s) based on the rotating rate of Stepper Motor (i.e., full, half, quarter, and 1/8th rotation rate). To send the command to the syringe pump, we use Arduino Series Monitor to input the desired injection/refill mode. **Figure 7** shows the interface of Arduino Series Monitor, in which 9 injection/refill modes are listed for reference. Particualarily, for long-term injection-refill process, mode 9 is selected.

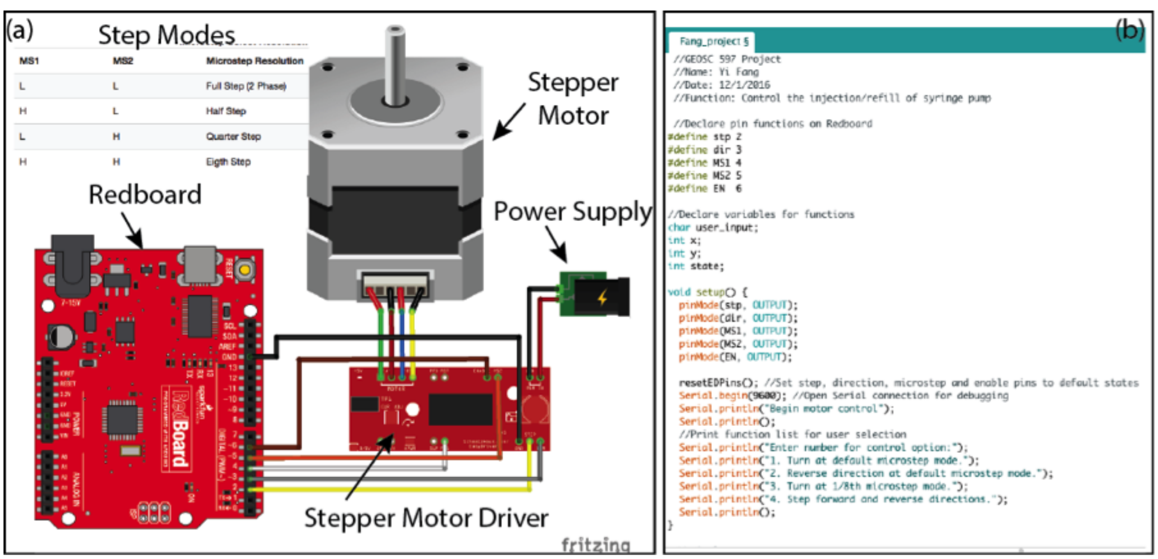


Figure 6. The Ardunio Redboard and Stepper Motor Driver together are used to send injection/refill command to the physical hardware (*i.e.,* Stepper Motor) to drive the syringe

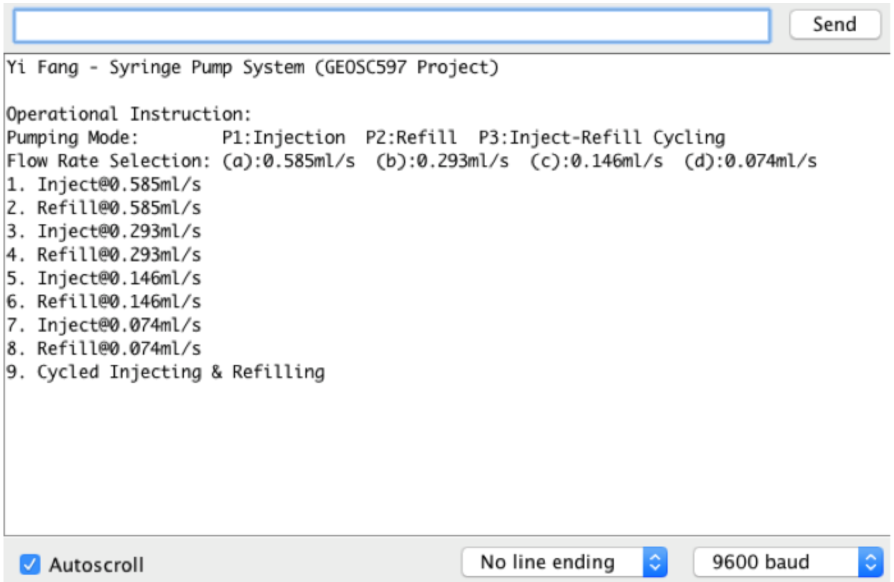


Figure 7. Interface of syringe pump control system

3. Challenges and Solutions

In developing the syringe pump system, following small challenging issues are encountered:

(1) Limited budget for solving realistic problem. To solve this problem, the simplest way is to simplify the problem. As shown in **Figure 2**, the realistic problem is simplified to an analogue version that still has the desired functions.

(2) Unmatched size between syringe and structure element: the problem was resulted from designing the structure element before syringe was purchased and delivered. This problem was easily solved by replacing the syringe with an appropriate size that matches the structure elements.

(3) Calculating and calibrating the flow rate: as no sensor is used to monitor the fluid volume, the only way to calculate the flow rate is by counting how much time it needs for syringe to pump out 20 ml water. In other words, this time is the same time that stepper motor needs to move forward a certain distance. After a series of tests, the precise flow rates are confirmed as illustrated in **Figure 7**.

Acknowledgement

I appreciate our lecture Mr. John Leeman for ordering the project hardware for me and for his great lectures and guide in the class. Thanks to Dr. Chris Marone for his kind help throughout the class.

Media

`Slides (PDF)`

`Slides (PPTX)`

`Code (ZIP)`

Arduino Alarm System - Gauanglong Hu

Problem

I had a very bad experience of burglary when I was a sophomore in undergraduate school about four years ago. I thought where I was living was a very safe place and I did not have a habit to lock the door. When I was reviewing for the exam in the morning about 2 am midnight, someone entered my room and tried to steal something. Fortunately, the person saw me awake and he fled immediately and nobody was hurt. This experience had haunted me for about two years and I could not fall asleep if I did not check whether the door was locked and with a baseball bat beside my bed. Moreover, there are some violence and crimes against specific ethic group. There was a song called "Meet the Flockers" by YG. The lyric is a standard operation procedure to motivate people how to rob a Chinese people. After the song was published, there have been some fatal home invasions targeted Chinese residential area. However, the most important thing is that I cannot possess a fire arm legally as a legal immigrant. Therefore, an alarm system will become helpful. It can generate high pitch sound to wake me up and also possibly scare burglars away. It can also mimic the light effect on police vehicle.

Methods

The alarm system was built using a distance sensor, a piezo buzzer, a blue LED, a green LED, a red LED, Sparkfun red board, and a bread board. The alarm system is shown in Figure 1 and the schematic shows the connection in Figure 2. The method to detect whether someone have broken in is to put the alarm system somewhere near the door and it will detect the distance between the distance between the door and the system itself. If the distance is smaller than the previously specified value in the code, the alarm will be triggered.

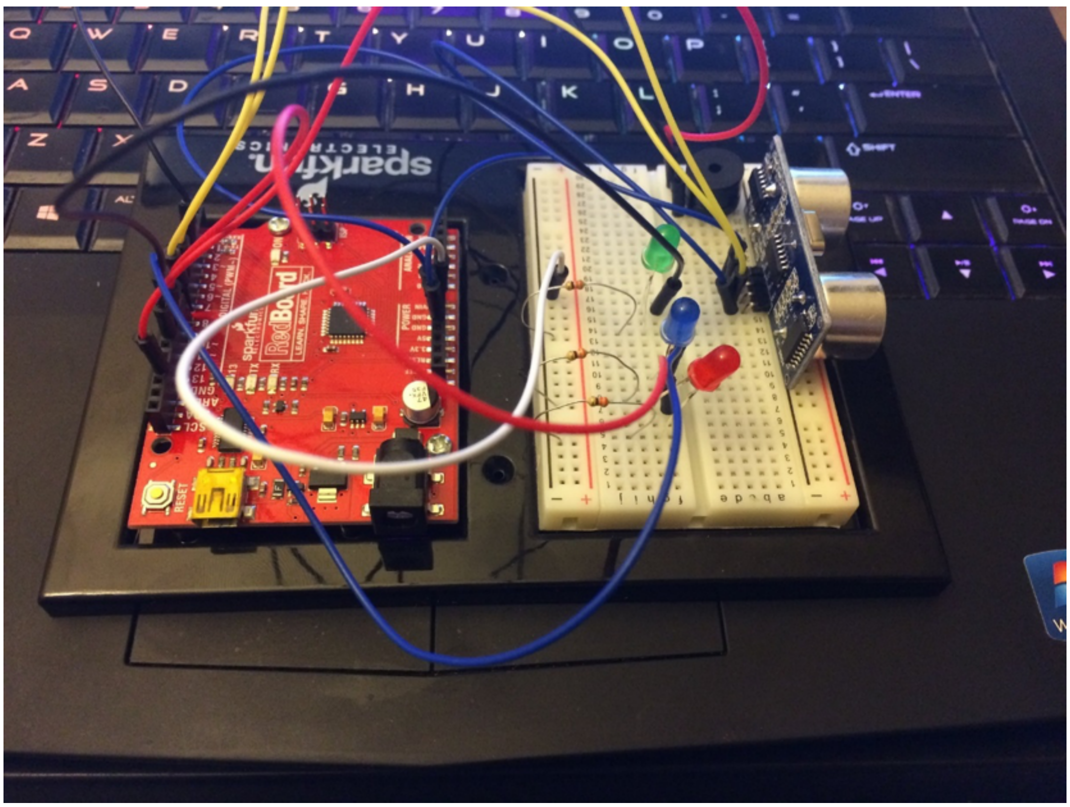


Fig. 1.82: Figure 1. Arduino alarm system

Challenge

The challenge I encountered in this project is that I did not know how does the distance sensor worked. As shown in Figure 3, the distance sensor is a SR04-Ultrasonic Distance Measurement Sensor and it has four different pins. The GND and Vcc are used to connect to the GND and 5-volts power supply portal on the Sparkfun red board. The "Trig" pin is used to command the sensor to send sonar signal to objects in front of it. The "Echo" pin is used to receive the signal reflected back from the objects and record the time consumed for the signal to come back after it was sent.

How it works

After the code is uploaded to the Sparkfun red board, the green light is on to indicate that the alarm system is armed. If

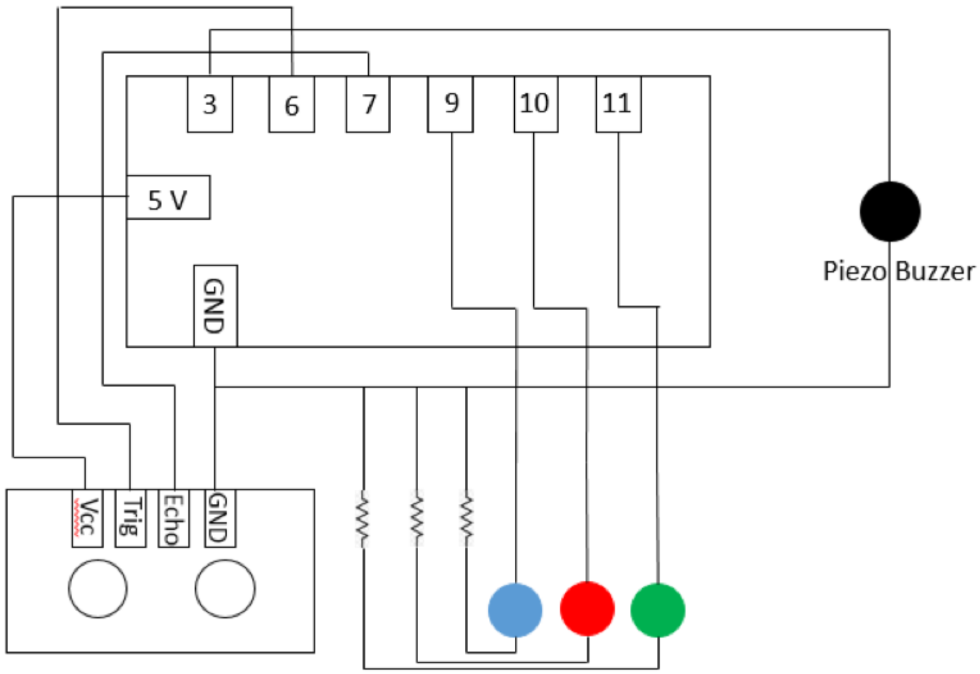


Fig. 1.83: Figure 2. Connection of Arduino alarm system

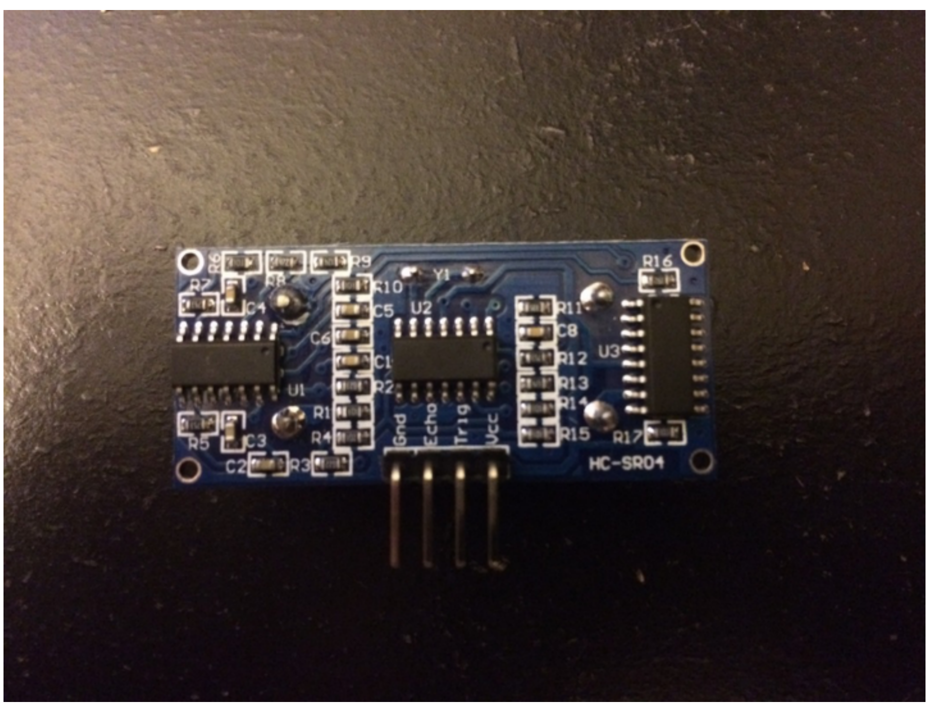


Fig. 1.84: Figure 3. SR04-Ultrasonic Distance Measurement Sensor

an object is put in front of it and the distance is smaller than the specified value in the code, the alarm will be triggered. The piezo buzzer will generate high pitch voice at a specified frequency and blue and green light will flash. The state machine diagram is shown in Figure 4.

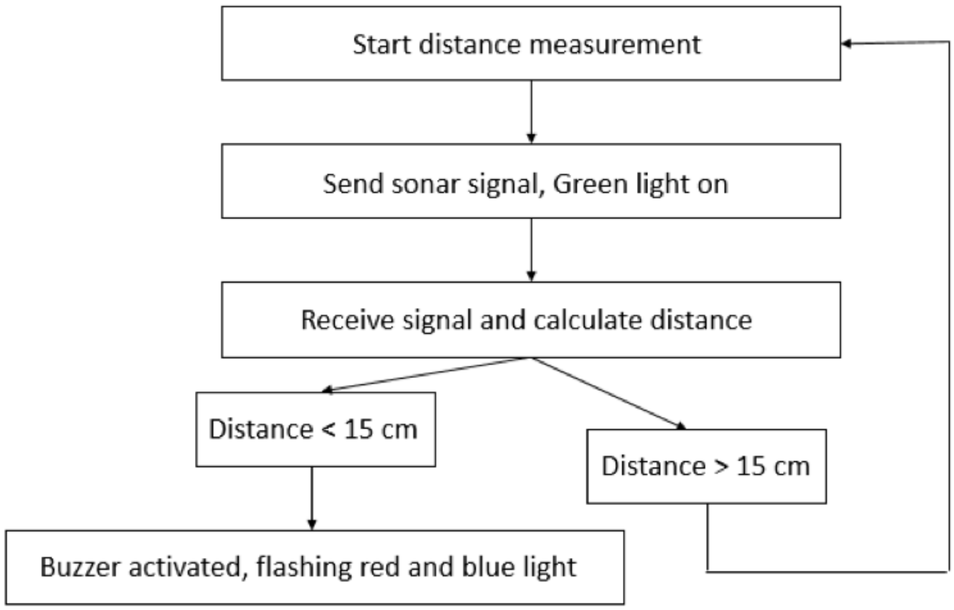


Fig. 1.85: Figure 4. State machine diagram of Arduino alarm system.

Media

`Slides (PDF)`

`Slides (PPTX)`

Sponge-quake Shear Box - Kyungjae Im

Problem Description

Earthquake is typically demonstrated by stick-slip motion in laboratory. Although experiments successfully describes substantial mechanism of earthquake, it has been not enough to reproduce natural earthquake since typical experiments only have single degree of freedom while natural earthquakes occurs on two dimensional fault plane with complexly distributed frictional properties. Increasing dimension of laboratory stick-slip would show a lot further information such as slip behavior on a plain with series of different frictional properties, earthquake nucleation, stress and slip purse propagation and so on. However, although there are several recent successful observations for laboratory induced multi-dimensional slip, the experiments are expensive and also requires huge care because rock is too stiff so deformations becomes too tiny to be measured and large dimension of rock mass and apparatus are required.

Proposed Solution and Requirements

To observe multi-dimensional slip behavior on distributed frictional property, a shear box with a soft shearing body is proposed for class project. First and most important requirement is to find such low stiffness but highly elastic material that represents an earth in earthquake. Initial candidates were (i) sponge, (ii) styroform and (iii) rubber. Several sponges were tested first and an appropriate candidate that is generate localized slip and stiff enough was selected.

To fulfill to goal of the proposal, following items are required to be further satisfied

- Uniform normal & shear application
- Unstable patch within stable surface
- Deformation measurement

Fig. 1A shows simplified mechanical framework of proposed shear box. It shows how boundary normal and shear stresses sould be applied on the shear body. Unstable patch will be placed on the bottom of the sponge and measurement will be made right on the top of the patch. Fig. 1B and 1C shows simulated normal and shear stress distributions after applying shown boundary forces. Resulted stress field are not perfectly uniform, but seems to be good enough at least near the unstable patch.

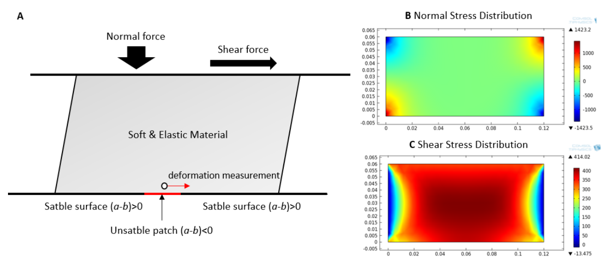


Fig. 1.86: Figure 1. A: Simplified mechanical framework of proposed shear box. B and C represents simulation results when boundary forces shown in A are applied.

Final Design

Several different mechanical design were proposed and configuration of Fig. 2 is selected as a final design. Overall apparatus is framed by aluminum U shaped bar. I bought a guide rail and two carriages to minimize frictional resistance when shearing the top of sponge. The carriage is bolted to a slider that can cover the top area of sponge. Slider bottom is covered by a rubber patch so that the shear force can be well delivered to sponge. Commercial linear actuator (Actuonix L12) is used to push slider. Bottom friction surface is made by rubber partly covered by kitchen wrap. I checked that kitchen wrap / sponge contact always shows stable sliding and rubber / sponge contact shows unstable stick-slip. Therefore size of unstable patch in this experiment is controlled by coverage of kitchen wrap and exposed sized of rubber.

For deformation measurement, Sparkfun flex sensor was initially considered. The sensor indeed can measure the displacement of sponge but resolution was poor. The sensor is changed to DCDT. Connecting the DCDT to separate 30V power supply connected to Arduino analog input (10bit, 1023 step) can yield displacement resolution as small as 5um which can be good enough to this prototype shear box. DCDT is connected to a pin that is firmly held at the bottom of the sponge.

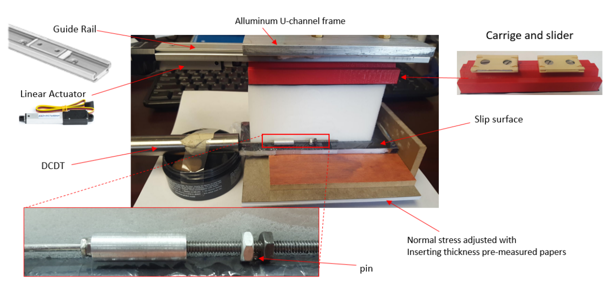


Fig. 1.87: Figure 2. Final design and major components of shear box.

Electric circuit

Two power sources are used: (i) Arduino 5V for powering linear actuator and (ii) separated 30V DC power to supply current to DCDT. Arduino breadboard and push switches are used to manage current direction to linear actuator in which way forward and backward operation can be controlled. DCDT output signal is connected to Arduino analog input and sent to computer to analyze.

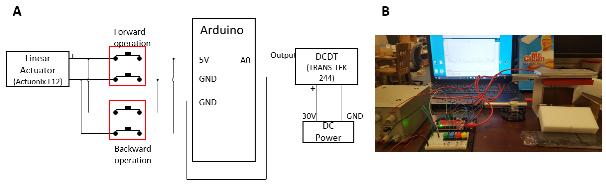


Fig. 1.88: Figure 3. Electric circuit diagram (A) and fully connected sponge-quake shear box.

Experiment set up and Results

12 different patch size with 8 normal stress are tested. Patch size starts from 5mm increased by 5mm until 60mm. Normal forces starts from 1.6N (0.5kPa, no paper) with increase by 2.74N (normal force increase with 10 papers) until 20.8N (6.4kPa, 70 papers inserted). Experiments on each patch size starts from no paper (1.6N), then 10 papers (4.3N). If slips are stable with 10 papers, I jumped to 50 papers (15.3N) and then 70 paper (20.8N). First stick-slip was detected at 40mm / 1.6N and then further experiment almost always shows stick-slip.

Fig. 4A and 4B show two different mode of sliding. A is stable displacement response of 25mm/4.3N and B is displacement response of 50mm/12.5N case. Difference can be clearly seen B shows clear stick-slip motion while A shows stable sliding. Displacement of each slip event shown in Fig. 4B is ~ 0.7mm which is far larger than the displacement that we typically observes in rock shear experiment. This shows that this sponge-quake can be utilized to indirect observation of earthquake slip behavior.

Plot C and D represent maximum slip velocity (C) and observed stability (D). Observation of slip velocity shows slip velocity increases with both patch size and normal force. Note that 0.005m/s is actuator velocity at no loading. So 0.005m/s represents stable sliding. Markers in Plot D denotes observed stability: square ? stable, triangle ? stable and unstable mixed and circle ? unstable. Interesting observation is that the stable-unstable criteria seems to be solely dependent on patch size. Theoretical and experimental analysis says the nucleation length is also dependent on normal stress. More experiments are required to see if this observation is repeatable.

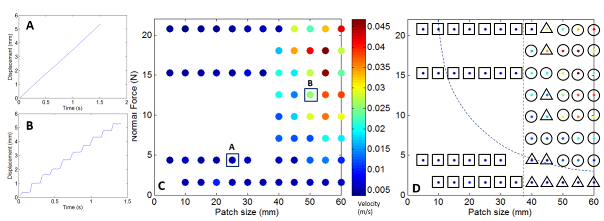


Fig. 1.89: Figure 4. Experimental result.

Problems and Further Upgrade Items

I expect this sponge-quake can reproduce substantial mechanism of natural earthquake in distributed frictional properties. Biggest problem of this prototype shear box is that sampling interval is too large (~20ms) at Arduino analog input which is not enough to observe pre-slip and accurate peak velocity in stick-slip motion. And also longer sponge is required to apply uniform normal and shear stress throughout the sponge body. Current boundary force application yield higher normal stress concentration on sponge front (fig 1) so normal stress decreases as the sponge slips. Further, to have better observation of natural earthquake behavior, several things can be upgraded such as multiple loading velocity, using gouge as a frictional surface and/or multiple point to measure slip propagation.

Media

`Slides (PDF)`

`Slides (PPTX)`

Strain Gauge Based Displacement Measurement - Ben Madara

Problem:

There are several displacement measurement methods. Currently, I utilize a linear variable differential transformer to measure the compaction and dilation of my experimental samples under load. This LVDT is housed inside of a high pressure vessel. LVDT is quite large for a crowded space as it requires two additional steel blocks to be fit into the set up. Additionally, the necessity of the steel mounting blocks means the measurement is not directly on the sample. This means the displacement measured is also dependent on any tilting of the mounting and loading blocks.

Solution:

I built a strain gauge based system to measure the horizontal displacement. The design was a prototype to be used outside of the pressure vessel that fits onto the 10x10 blocks in the biax. The larger size was for simplicity of electrical connection and the use of already owned strain gauges. The system works by measuring the change in resistance between four strain gauges laid out in a diamond shape by means of a Wheatstone bridge. The base material for the diamond shape bracket needs an elastic yield point higher than 30 MPa, so I used an aluminum alloy (McMaster-Carr 5052). The bracket frame was designed in Onshape. The design was first 3d printed to ensure sizing and functionality of the parts. Then a water just was used to cut the bracket pieces out of a ¼" aluminum plate. Once the pieces were in hand I was able to apply the four strain gauges, one to each of the aluminum bracket pieces. With the strain gauges super glued in place the connections were then soldered to complete a Wheatstone bridge. By using an external power source and multimeter the bracket was then bench tested. Upon a successful bench test the Wheatstone bridge was then connected to an amplifier circuit built on an Arduino Uno. The signal was amplified by the Texas Instruments INA128 amplifier. The Arduino was programmed to print and plot the voltage coming out of the Wheatstone bridge. The prototype successfully recorded strain and fit onto the 10x10 blocks for use in the biax. For future development of this project the first step would be to get higher quality strain gauges. Additionally, another amplifier and filtering of the data would improve the signal to noise ratio in the measurements.

Media

`Slides (PDF)`

`Slides (PPTX)`

Temp & soil moisture sensor for plants with web based data acquisition - Peter Miller

Problem description and solution

The purpose of this project was to create an Internet of Things (IoT) weather station that also incorporated a soil moisture-monitoring component. The device is intended to notify the user when the moisture content of the soil drops below a certain value. To notify the user that the moisture content of the soil is too low, 3 LED's, which correspond to the three soil moisture sensors, will turn on. The initial idea was to feed all of this data onto a server designed to handle IoT projects and then plot the data continuously. In practice I was able to record all of the data to a hard drive but was unsuccessful in streaming it to a web service. Because of this roadblock I have written a python module to read in and plot the recorded data with only a single input from the user. This project is also described in a state machine diagram (figure 1), which shows the workflow of the device.

Hardware and software solution

The atmospheric data is collected by a BME-280 breakout board, which was purchased from Sparkfun. The temperature is recorded in degrees C, relative humidity in %, altitude in meters, and atmospheric pressure in Pascal's'. This breakout board has the capacity to transmit data using both the SPI and I^2C protocols. For this project I used the I^2C protocol because space on my breadboard and the connecting wires were finite commodities. Since this board required 3.3V, a logic shifter was needed to step the output signal up so that the 5.0V SDA and SCL serial ports on the Arduino can read the data. The original Arduino code for the BME-280 was borrowed from Sparkfun and modified to support three soil moisture sensors and the LED warning lights.

To measure soil moisture I had three Sparkfun soil moisture sensors (SEN-13322). The moisture sensor functions as a variable resistor where the voltage between the two pads is a function of the conductivity of the soil. The conductivity of the soil increases as the volumetric water content, or total volume of water in the porous soil increases. When connected to the analog input of the Arduino, these sensors only throw values between 0 (perfectly dry) and 900 (100% saturated) instead of 0 - 1023. And since the maximum saturation of soil is roughly equal to the critical porosity of the soil or ~50%, I defined a linear saturation scale between 0 and 100%. At 100% the soil is effectively a colloidal material (mud). A perfectly dry soil ideally has an output voltage of zero because the constituent minerals are non-conductive. The lower limit that I placed on the moisture content of my soil was 250 or 27%. At or below this value the light will start flashing indicating that the plant needs water. See figure 2 for image of setup.

A simple calibration was performed in order to convert the serial output from the soil moisture sensor to a percent saturation. The two soil end member cases (colloidal sand and dry sand) were measured to determine the minimum and maximum output values. Since the relationship between the output values and the percent saturation was assumed to be linear, the ratio of the output value to the maximum value was assumed to be the percent saturation.

Finally, to visualize the data I wrote a python module that sorts and plots the data collected from the atmospheric and soil moisture sensors. The module works only on CSV data files and allows the presence of a header. The plots generated are saved as individual pdf documents to a specified directory. For more information please see the commented code.

Issues

The two data collection systems all functioned properly, however, the final goal was to make this data available online through IoT services like Thingspeak or data.sparkfun. I was unsuccessful in interfacing the ESP8266 WiFi module with one of the services largely due to an inability to send the module AT serial commands programmatically. The ESP8266 module allows a microcontroller to access the local WiFi network through standard TCP/IP protocols. The AT commands pre-loaded into the firmware allowed me to access IoT services through direct user input as serial commands, however automating these commands proved elusive.

Next Steps

This device both collected data continuously for an extended period of time and functioned as an alert device if the soil moisture dropped below the specified value. As a concept device this project worked. However, there are a few steps (listed below) needed to make this an automated IoT device.

1. Consolidate wiring and modules onto a smaller board so that it will fit into a reasonably small waterproof case. The soil moisture sensors are waterproof and so they are able to function in the environment.
2. Fix the program for WiF connectivity. This is close to working but the issue of programmatically sending serial AT commands needs to be fixed and/or circumvented.
3. Generate an automated alert system that will either text or email the user when the soil moisture reaches a dangerous level for the plant that is being monitored.

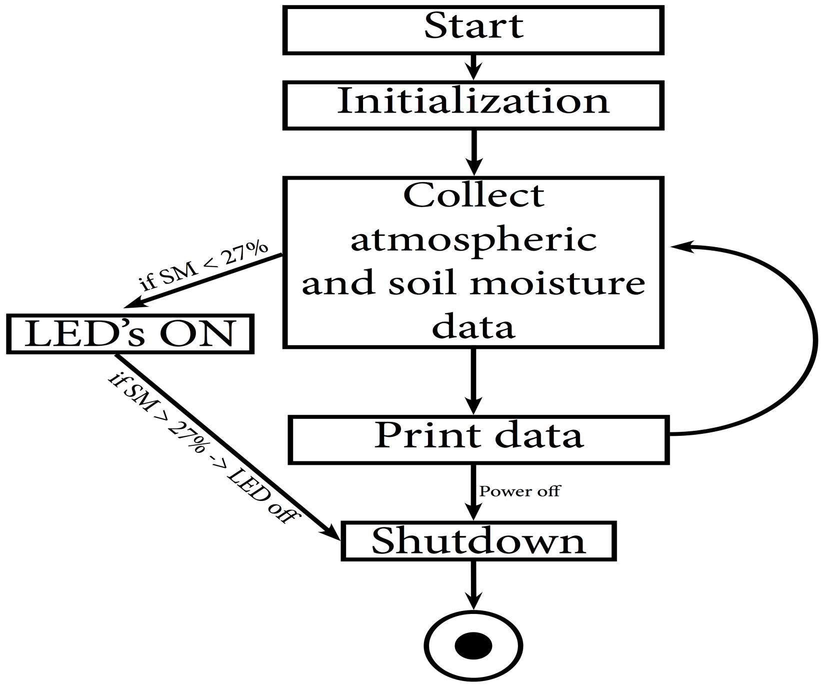


Figure 1: State machine diagram for soil moistures sensor. Once the initialization process takes place, the data is collected and the system checks the soil moisture. If it falls below the critical value of 27% then the LED attached to that sensor will turn on.

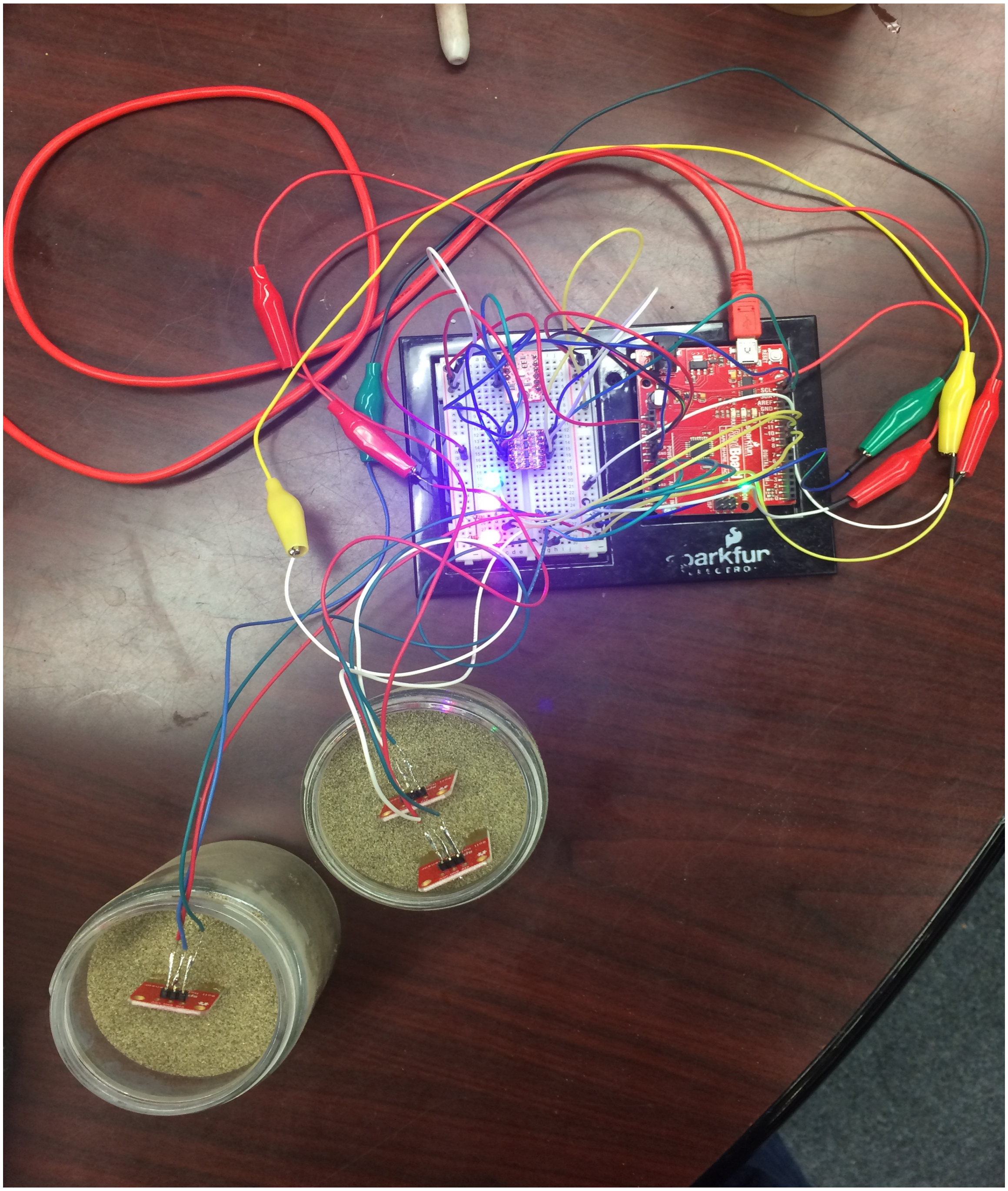


Figure 2: wiring setup between temperature sensors, soil moisture sensors, and the Arduino. The soil moisture sensors in this image are placed in the sand. The LED's in this image are both signifying the sand is below the critical water content of 27%.

Media

`Slides (PDF)`

`Slides (PPTX)`

`Code (ZIP)`

Shot Spinner - Kerry Ryan

Problem:

I wanted a way to entertain guests at parties and other social gatherings. What better way to impress your friends than with science AND beer!

Intended Solution:

My plan was to design an automatic beer-dispensing device. A step motor controlled platform that will rotate beer glasses underneath a spigot to be filled. Hall effect sensors will be used to identify when a glass is in position to be filled. The glasses will be filled from a large pitcher located in the center of the device.

Device Requirements:

1. Rotating table to put cup/container under spout
2. Sensor to identify when cup/container is present
3. Valve to control flow of liquid into container

Design Process: A lesson in resourcefulness, repurposing, and resilience

To start the design process I began by developing a state diagram to layout how I intend my device to work and to make sure that it would meet all of the requirements. Shown on the next page is my original state diagram.

State Diagram:

Part 1: Rotating Platform

I started off my project by trying to get the platform rotating. The idea was to use a stepper motor and a series of 3D printed gears to spin the platform. However, I had difficultly finding a lazy Susan that was appropriate for this project. Most of the ones I found online were made of glass and too heavy and/or large for this project. I found a smaller plastic one online. When the lazy Susan arrived I noticed that it was difficult to turn and did not roll smoothly on its bearings. The specifications of the item did not list how high off the surface the platform would be. These two factors made it difficult to set up the rotating platform the way I had planned.

Instead, I decided to try and mount the platform on top of the motor. The motor has a circular end which made it difficult to secure beneath the platform. Using a series of wooden rods and zip ties I was able to secure the platform on top of the motor. Some plumbers putty was used for additional stability.

Part 2: Operating the Motor

The next step was to set up the electronics for the motor. I used a 4 wire bipolar motor and easy driver from Sparkfund. Shown below is a hookup schematic from the Sparkfund website.

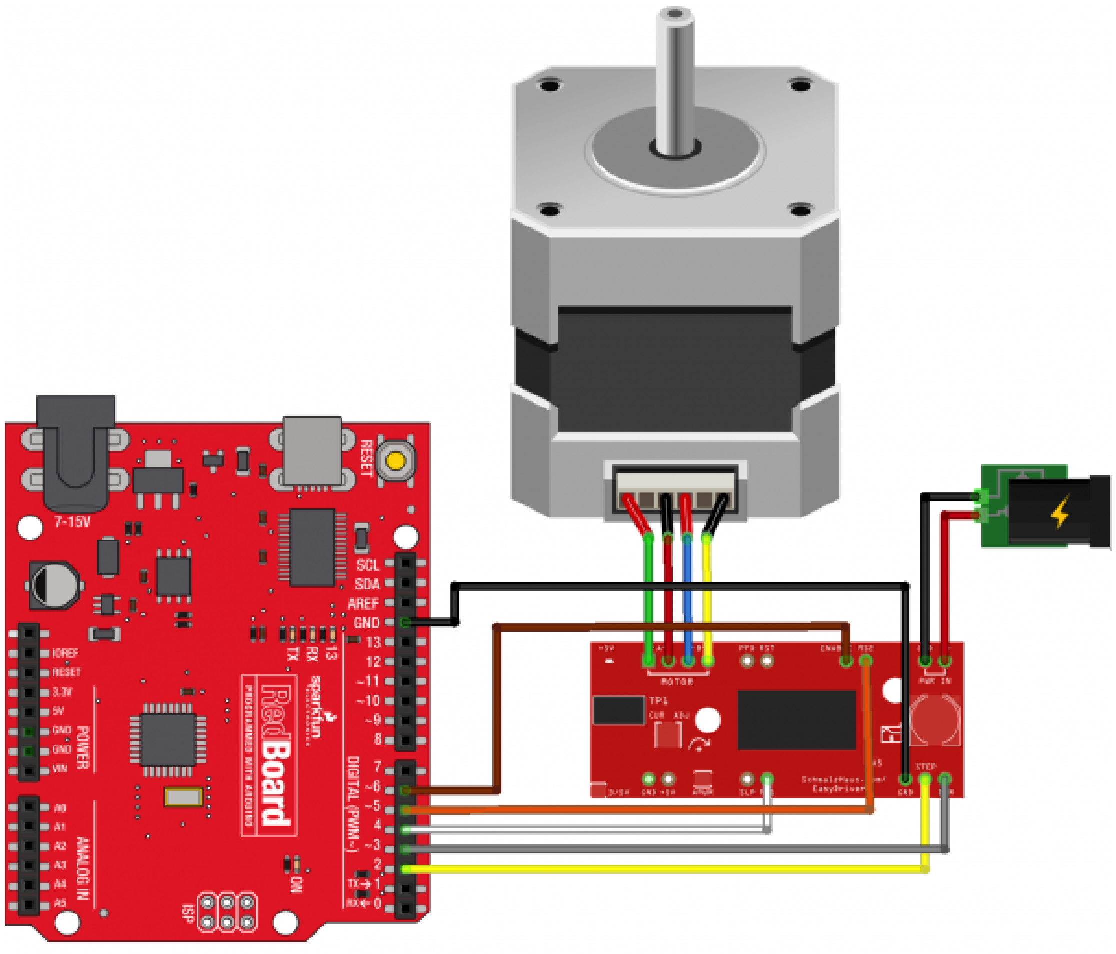
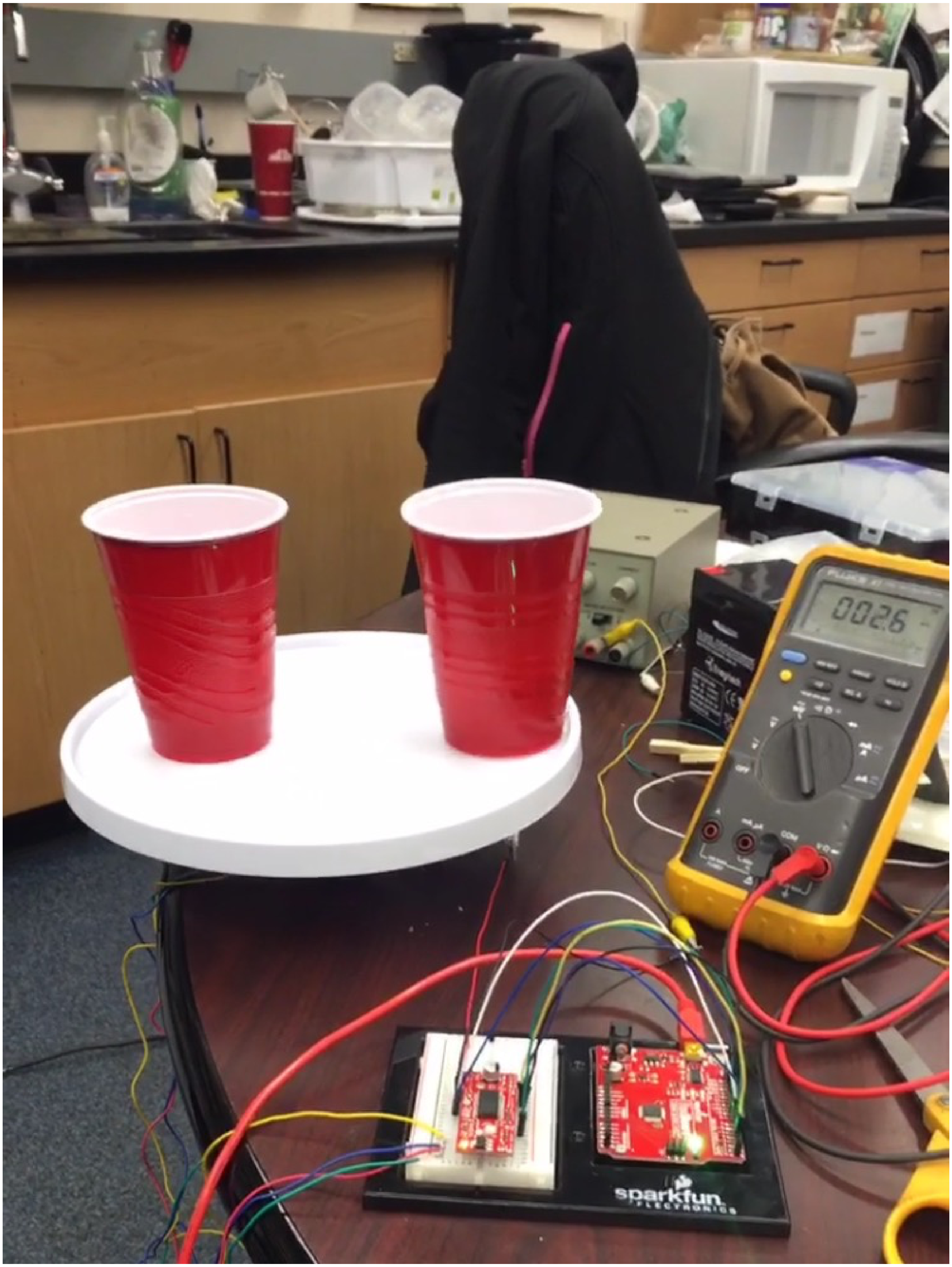


Figure 1: Schematic of stepper motor hookup. Details can be found on the Sparkfund website (https://learn.sparkfun.com/tutorials/easy-driver-hook-up-guide?_ga=1.133038150.938360751.1472576888).

Shown below is a picture of my actual set-up. The first time I attempted to setup the motor I was reading very low current that I believe was insufficient to power the motor. The setup required 6 solder connections and I believe that one of them may not have been very good. I took the setup apart and checked the connections. I re-soldered two of the connections and tried the setup again. Thankfully, this time it worked! The motor was successfully turning the platform.



However, at this point I came across my next obstacle. The platform was not very stable because of the way the motor was attached. Given this configuration it cannot support a lot of weight, i.e. cups of beer as it rotates. To remedy this, I decided using smaller plastic cupswith \ would be more feasible. With smaller cups and less liquid, the platform would be less likely to tip to the side.

Part 3: Hall Effect Sensors

The hall effect sensor is a transducer that changes its output voltage based upon the magnetic field. Specifically, I used a Melexis US1881 sensor from Sparkfund. This sensor is a latch sensor meaning that once the sensor detects a certain magnetic field it will stay in that state until it measures a magnratic field of the opposite polarity. To implement this in our design, the idea was to put magnets of alternating polarity on the platform so that when the magnetic field changed state the device would know that a new cup was ready to be filled.

The hall effect setup is shown below:

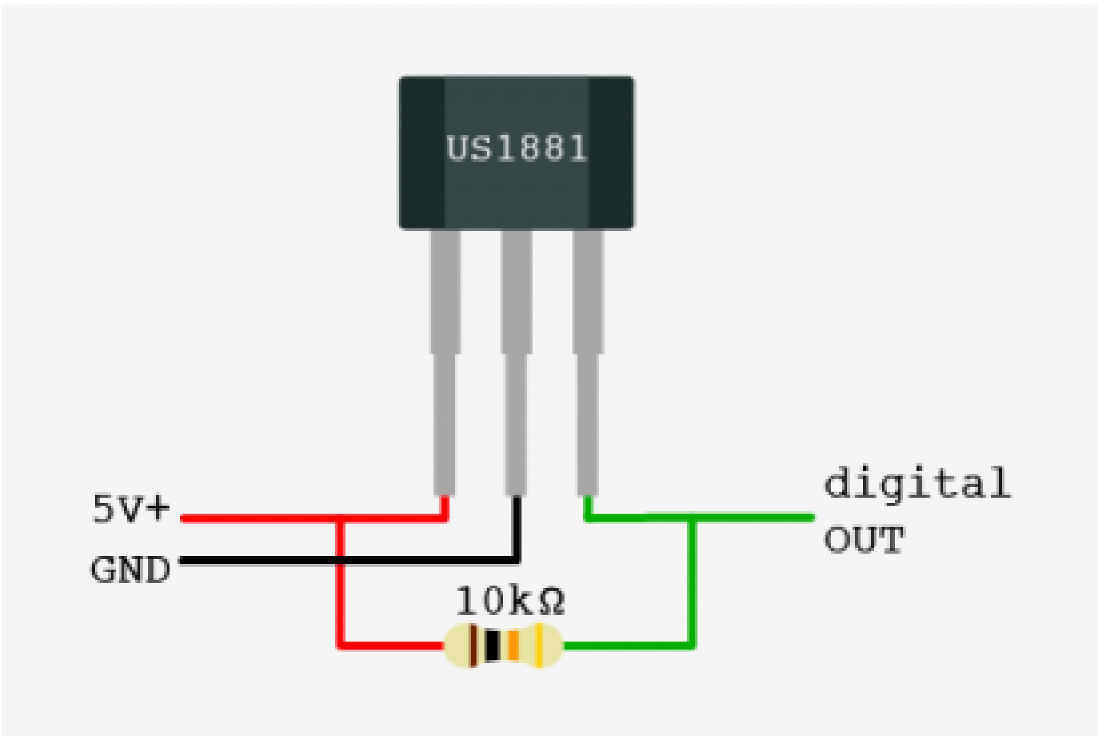


Figure 3: Hall effect sensor schematic.

http://bildr.org/2011/04/various-hall-effect-sensors/

Part 4: Dispensing Valve

To dispense the beer and/or any beverage of your choosing, I purchased a solenoid valve, food safe plastic tubing and a beer pitcher. Unfortunately, the valve I purchased was not appropriate for this project. The documentation on the website was not very clear to me and the valve required assembly parts that I did not purchase. Without this part, I had to rethink what my device could do. At this point, rather than making a beer dispenser I decided to make a shot spinner. A fun game that guests could play.

Part 5: Finalizing the shot spinner

The idea of the shot spinner is you place already filled small cups on the platform. Players sit around the platform. The game begins when you press a button located on the arduino board. The arduino randomly generates 2 numbers (which uses randomseed to start at a different place in the random number sequence each time) . The first number (0-100) determines if the platform roatates clockwise (0-50) or counterclockwise (51-100). The second number (1-2000) determines how many steps the motor takes i.e how far does it turn. When the platform stops the person with the shot drinks it. At this point you refill the cups and go again!

To make the shot spinner functional I programmed my Ardiuno as a state machine.

Initial State Check Button State Random Number Generator State Forward Rotate State or Backward Rotate State Check Button State

The process repeats this way. The default/shutdown case does not move the motor.

Future work: Along the way I encountered a number of problems that could be addressed in future modifications of this device. For one, investing in a quality rotating platform or designing a custom made platform for your purpose is important. It would be nice to have a lightweight but stable rotating platform that could support the full weight of pint glasses.

In addition, finding an appropriate valve to regulate flow would be desired. The automatic refilling could also be incorporated in to shot spinner.

In addition, this device can be repurposed for many useful tasks. Ever have trouble deciding what is for dinner? Write different ideas down and put them in the cups. Use the spinner device to pick for you! Or is there ever a certain task that needs to be done but no one is volunteering to do it? Put everyone's name is the spinner and push the button to make the final decision!

Code - Project

```
// Shot Spinner by Kerry Ryan
// Segments of code modified from Sparkfund Easy Driver Hook-up Guide

 //Declare pin functions on Redboard
#define stp 2
#define dir 3
#define MS1 4
#define MS2 5
#define EN  6

//Declare variables for functions
char user_input;
int x;
int y;
int state;
int randNumdir;
int randNumturns;
const int button=12;

// Identify Different States
  const int ButtonTrigger=0;
  const int RandomNum=1;
  const int ShotSpinForward=2;
  const int ShotSpinBackward=3;

void setup() {
  pinMode(stp, OUTPUT);
  pinMode(dir, OUTPUT);
  pinMode(MS1, OUTPUT);
  pinMode(MS2, OUTPUT);
  pinMode(EN, OUTPUT);
  pinMode(button,INPUT);
  resetEDPins(); //Set step, direction, microstep and enable pins to default states
  Serial.begin(9600); //Open Serial connection for debugging
  Serial.println("Press button to begin!");
  Serial.println();
```

```
  randomSeed(analogRead(0));
  digitalWrite(EN, LOW); //Pull enable pin low to allow motor control
}

void loop() {
  // Start of in Random number generator case
   static int state = ButtonTrigger;

    switch(state) {

      case ButtonTrigger:
      //Wait for Button Trigger Case
      if (digitalRead(button)==HIGH){
        state=RandomNum;
      }
      else{
        state=ButtonTrigger;
      }
        break;

      case RandomNum:
      //Random Number Generation
      randNumdir=random(0,100);
      randNumturns=random(1,2000);
      delay(100);

      if (randNumdir>50){
        state=ShotSpinForward;
      }
      else{
        state=ShotSpinBackward;
      }

      break;

      case ShotSpinForward:
      // Platform will rotate forward randNumturns steps
      Serial.println("Shots! Shots! Shots! ");
      digitalWrite(dir, LOW); //Pull direction pin low to move "forward"
      for(x= 1; x<randNumturns; x++)  //Loop the forward stepping enough times for␣
↪motion to be visible
      {
        digitalWrite(stp,HIGH); //Trigger one step forward
        delay(5);
        digitalWrite(stp,LOW); //Pull step pin low so it can be triggered again
        delay(5);
      }
      Serial.println("Refill! Then push button again!");
      state=ButtonTrigger;

        break;

      case ShotSpinBackward:
       Serial.println("Party! Party! Party!");
       digitalWrite(dir, HIGH); //Pull direction pin high to move in "reverse"
        for(x= 1; x<randNumturns; x++){  //Loop the stepping enough times for motion␣
↪to be visible
          digitalWrite(stp,HIGH); //Trigger one step
```

```
          delay(5);
          digitalWrite(stp,LOW); //Pull step pin low so it can be triggered again
          delay(5);
          }
          Serial.println("Refill! Then push button again!");
          state=ButtonTrigger;


      default:
        // Shutdown Case
        while(state>3){
          resetEDPins();
        }

    }
 }
//Reset Easy Driver pins to default states
void resetEDPins()
{
  digitalWrite(stp, LOW);
  digitalWrite(dir, LOW);
  digitalWrite(MS1, LOW);
  digitalWrite(MS2, LOW);
  digitalWrite(EN, HIGH);
}
```

Code - Hall Sensor

```
 const int magstate=7;
int sensorVal;
const int light=9;

void setup() {
  // put your setup code here, to run once:
pinMode(magstate, INPUT);
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
sensorVal=digitalRead(magstate);
Serial.println(sensorVal);
delay(500);
  if(sensorVal==1){
   digitalWrite(light, HIGH);
    }
else{
  digitalWrite(light, LOW);
}

}
```

Media

Slides (PDF)

`Slides (PPTX)`

Lightweight Precision Altimeter for UAV Survey of Ice - Emily Schwans

Problem Statement

Determining the location of the ice-air interface relative to the flight height of a UAV within the order of magnitude of accuracy needed for a high resolution, low frequency GPR survey is difficult to do using a simple DGPS altimeter, especially when working in a temperate area where there is likely to be a wet interface on the surface of the glacier that will cause a large amount of scatter of EM energy. Verification of interfaces within ice detected with GPR without digging pits or doing other ground-based work is also difficult. Hence, developing a low-cost, lightweight solution for accurate altimetry measurements is one of the first steps in designing a system for testing the feasibility of sounding en- and subglacial hydrologic systems in temperate glaciers using low frequency, drone- based GPR.

Design Considerations

While laser or ultrasonic rangefinders are more accurate than pressure-based altimeters, the former are expensive for far-ranges. Using a pressure sensor that resolves output to within fractions of a Pascal is accurate enough for this stage of design, and cost-effective in terms of the parts I needed. The hypothetical UAV-mounted GPR system would have to be light; therefore the altimeter has to be lightweight also. Additionally, the antenna on that system could not be heavy, either, so it would likely be necessary to use SAR techniques to increase the aperture of the antenna while keeping the desired low frequency and a small/light antenna. To do this, as precise of location data as possible are needed during flight time, including altitude measurements. This is why I chose a sensor that is accurate to within 30 cm (even down to 6 cm, according to the data sheet), weighs very little, and is compact. Another design consideration was the need to take readings at a rate suitable for a potential UAV-based GPR survey (Ruckamp et. al. 2011 cite 10Hz as their sampling rate for a UAS-based GPR survey). The sensor I used can output data at a comparable rate. The sensor is also effective down to -40 C, so it would be usable in a cold setting.

Implementation Process

Testing and prototyping was done with Arduino Redboard, requiring no soldering. While I did initially attempt to solder wires to the sensor so I could have it off the breadboard and do the rest of the circuitry with plenty of room to work with, I discovered I am not at all skilled in through-hole soldering. So, after ruining the first sensor I got and having to reorder another on my own, I decided to simply use the breadboard. This worked nicely, allowing me to rework my circuit as needed: adding buttons to switch modes was much easier. In hindsight, I would have liked to have learned how to solder better early on so that when this project rolled around, I'd be better prepared and could have produced a more compact, less wire-intensive prototype.

The sensor itself was quite simple to hook up using the breadboard and jumper cables. However, it required an input of 3.3V, and to communicate with it using 5V RedBoard, I had to add in-line resistors to the SCL and SDL connections, despite powering it directly with the 3.3V output on the Redboard. Initially, I simply had it plugged into the breadboard with jumper wires, but it kept slipping around and losing contact with the resistors, giving me a -999.99 reading for everything. So, I bent jumper wires and resistors to maintain contact. However, maintaining contact between the resistors and the breadboard for the button was a big issue while I was testing my code. In hindsight, getting a smaller board that I could physically solder components to would have been prudent, assuming I could have done clean soldering work.

Using a MacBook Air to upload the code was a big challenge: often, it would not recognize that my board was connected, so most of the time I was unable to upload the code without restarting my computer, and using the Serial Monitor or Plotter to get arrays of data was more or less impossible.

Additionally, I attempted to use a 7-segment LED display with pins on both sides, but was unable to get it to display what I wanted with enough detail to be useful. I found that the LCD display included in my inventor's kit was much easier to use once I figured out how to connect all the pins, and was much better to read data off of and to see when I switched between modes. It also didn't take up as much room on my breadboard having only one side with pins, which was nice since I had to also add a potentiometer to adjust the contrast on the LCD display so I could read it.

Had I known the LED would take up most of the breadboard, I would not have gotten that part. Also, I failed to take into account that this display would not have enough digits for readings in Pascals at the level of accuracy I desired.

For my code, I spent quite a while figuring out how to convert the reading from 20-bit to an actual altitude/pressure reading. This initial code was very long and took up a lot of memory on my RedBoard, which had been temperamental since the beginning, but even more so when the code took up more memory. Luckily, in going over the data sheets and specs online, I realized that there were libraries for my sensor out there that did the same thing I was trying to do, but in a much more simple way that took up less memory on my RedBoard when used. I looked into the source files for those libraries to make sure I wasn't just relying on a black box, but ultimately used these libraries instead of writing everything from scratch, because "reinventing the wheel", so to speak, is not time-effective, and using code that comes with a part you use seems reasonable as long as you understand how it works.

In general, I programmed the Arduino as a state machine; it continually takes readings in altitude mode until I press the button and switch states between either two altimetry modes (meters and feet above sea level) or one barometric mode, plus an initial state to configure the sensor, and a shutdown state, which it switches to when outside the operating temperature range. I could have also had additional states for changing between Pascals, mmHg, or millibars, but this was not part of my original design and theoretically I'd be using the data as an array anyhow, so could easily convert this post-acquisition.

I did not have the time to figure out how to hook up the battery to power the altimeter, but can do so in the future. I also could not figure out how to set the user-input sea level pressure, which is why the instrument is not very precise.

Instrument Testing

Accuracy in the altitude readings was tested by comparing altitude in multiple locations to those shown on topographic maps. Pressure readings were compared to local weather website measurements. Temperatures were compared to multiple thermostats. Ultimately, the temperature readings were accurate, as were the pressure readings, but the altitude drifted a lot. To get an idea of instrument drift, I carried the prototype up and down the stairwell, making note of any differences I saw in the top and bottom measurements over the course of carrying it up and down several times. At times, it was off by several meters.

Next Steps

A more permanent, compact design could be achieved using an Arduino Pro Mini ($9.95), which weighs less than 2 grams, has the same number of I/O pins, 2 extra analog pins compared to Redboard, and features an off-board USB connection. This would require more soldering work, but is necessary for a more refined prototype. Attaching the battery and the on/off switch, as well as adding a

Standby mode to conserve battery power while measurements are not being taken would be helpful. A chassis to protect the electronics from weather would also be a step in producing a more permanent product for field implementation. Additionally, programming the board in such a way so that it can either store measurements or transmit data via WiFi or Bluetooth to a computer or phone is necessary for field implementation, the latter of which is likely more realistic given the scope of a UAV-based survey and the amount of memory typically available.

The instrument doesn't "settle down" to the point where altitude is not constantly changing. This is a design flaw, and would have to be fixed if this were to be implemented for high-precision measurements while flying. This is due to the fact that the device is using pressure (which is varying) to calculate altitude. Having an insulated chassis could help with this.

Summary

All in all, I succeeded in designing a prototype for a high-accuracy pressure/altitude sensor that, with some tweaks and downsizing, could be used for UAV acquisition, albeit a laser range finder would be the best permanent solution (albeit expensive). I would need to fix the drift in the instrument and program a smaller, more permanent Arduino that could be attached to the sensor. I learned that soldering is not as easy as it looks, that circuits are complicated and I forgot a lot of what I learned about them in physics over the course of four years, and that, while it is easy to get lost in datasheets, really reading into the full capabilities of a given sensor is necessary to obtain the exact results you want.

Media

`Slides (PDF)`

`Code (ZIP)`

3D Laser Profilometer - Srisharan Shreedharan

The problem:

Pre- and post-shear analysis of rock samples in any rock mechanics lab would be well complemented by including information about the surface roughness of the sheared sample. This is commonly achieved by 3D photogrammetry or laser profilometry. My project builds a low cost 3D laser profilometer which may ultimately be used to scan the sheared surfaces of intact rock samples at Penn State's Rock Mechanics Lab. The scanner is a sub $30 prototype which will be improvised to incorporate higher resolution capabilities for use with smoother joint and fault surfaces.

Methods:

The product that I built is loosely based on two online instructables. The general principle behind the working of this profilometer involves the use of a line laser module and a webcam in tandem. The webcam 'looks' at the object that the line laser shines on and the code selects the 'red' pixels above a user specified threshold and saves them. Based on the distance between the webcam and line laser module, their distance from the object, and their respective geometric angles (as formed by the webcam,line laser and the object being scanned as vertices of a triangle), the code adjusts the point cloud to create a scan of the object.

I used the following resources for my project:

1. Sparkfun RedBoard and associated parts (eg. jumper cables, breadboard etc.)
2. 10k potentiometers x 2
3. Line laser module
4. Logitech webcam
5. NEMA 17 stepper motor (200 steps/revolution or 1.8º resolution)
6. TB6612 Stepper driver
7. 100 uF capacitor
8. Open source codes – Arduino 1.6.11 and Processing 3.2.1

Two broad mechanisms for the movement and position feedback are possible – seating the object to be scanned on a turnstile or moving the scanner (line laser and webcam) mechanism. I opted for the latter since it provides greater flexibility on the size and weight of the object to be scanned since it will neither be limited by turnstile size nor stepper motor torque.

To construct the mechanical side of the project, I took a plank of wood (33 cm x 5 cm x 2 cm) and named him Plank. I drilled a #20 pilot hole (11/64") through the center of Plank (16.5 cm, 2.5 cm) and chiseled a side of it so as to get a 'D' shaped hole to snugly hold the NEMA 17 stepper motor's 'D' shaped shaft. I mounted the line laser on one end of Plank and the webcam on the other end, with the RedBoard sitting at the center of my arrangement. Since the scanner had a weight imbalance as is, I added counterweights on the line laser side of Plank. The line laser was oriented perpendicular to Plank and the webcam makes a 20º-40:sup:*o* angle with it. A completely built prototype is shown in figure 1.

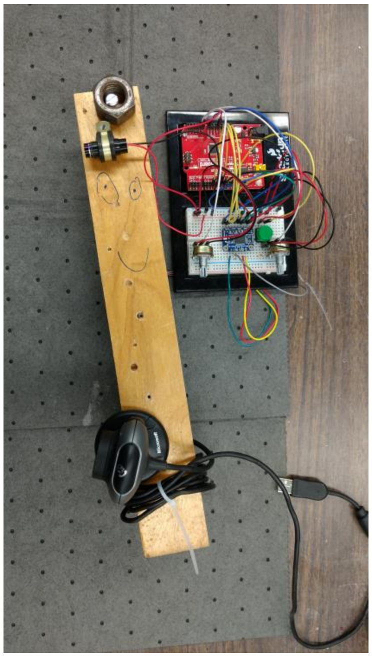


Figure 1: The completed mechanical build of the laser scanner

The electric build is relatively straightforward and figures 2 and 3 show the schematic and actual electric build I used in my project. The schematic shows an optional LCD screen that could have been used to see the speed and total angular movement of the device mapped to the two potentiometers, which I do not include in my project for lack of an I2C LCD. The schematic also shows the wiring for an A4988 stepper motor instead of the TB6612. I also include a button (not in schematic in figure 2) which restarts the device after each scan cycle if the user desires.

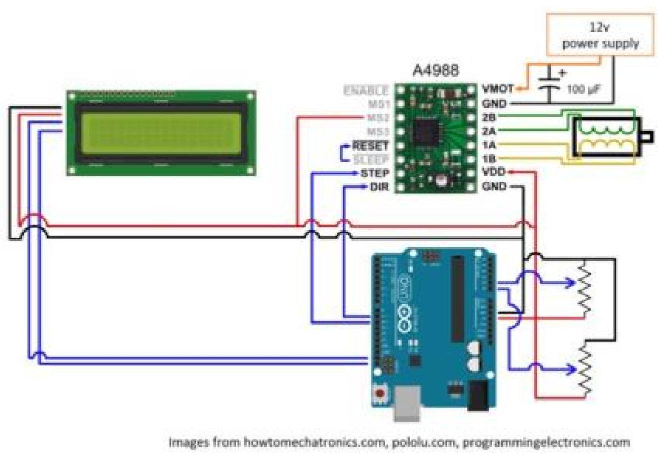


Figure 2: Schematic of the electric circuit

The Arduino code controls the physical aspects of the device. The device starts as soon as it is powered and steps from its starting point to a final angle (<360:sup:*o*) and a predefined speed, both of which are controlled by the two potentiometers. The scan occurs from 0 – final angle, at each step, after which the device steps back to its original position ‘0’. The processing of the scan occurs in a Processing sketch in tandem. The processing sketch A should be running during the scan sequence since it records the object’s reds as seen by the webcam and writes the coordinates into a file. Processing sketch B can be used after the scan to load the coordinates’ file, which it processes to provide a final point could of the scanned object. Sketch B requires the webcam angle relative to the laser angle (taken as 0 degrees) as an input.

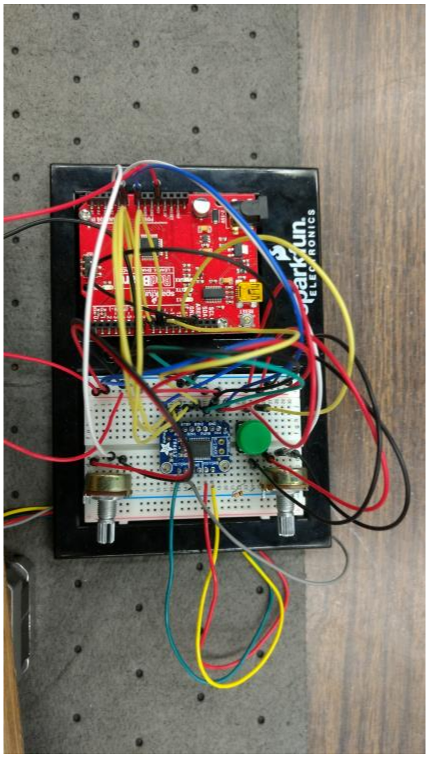


Figure 3: Actual wiring used in my project

Results:

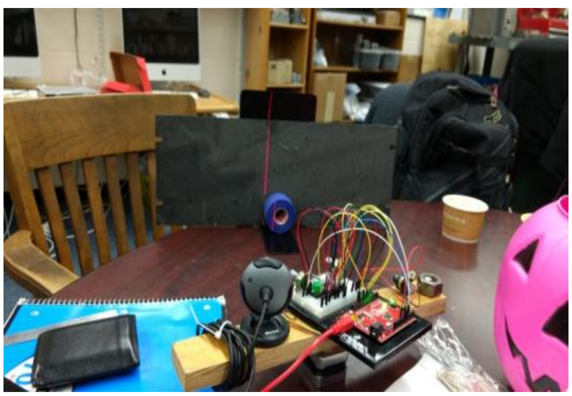


Figure 4: A laser scan of blue tape in progress

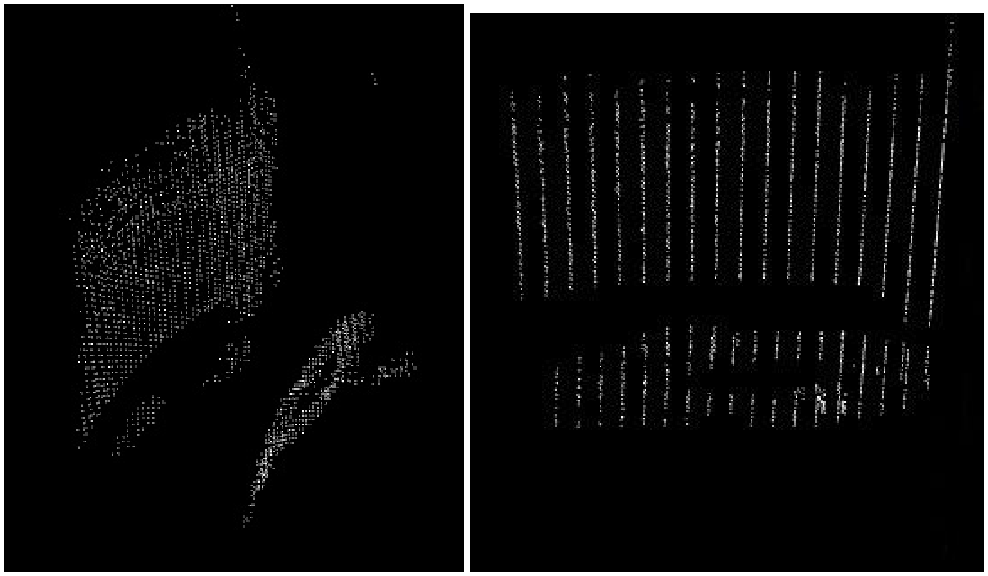


Figure 5: Two views of the scanned object.

Figure 4 shows a scan in progress, of a blue electrical tape, in the Rock Mechanics lab. Figure 5 shows the post-processed graphic render from Processing sketch B. The effects of distortion are apparent since the perfectly circular object appears to be elliptical. Features of the object itself, however, have been caught by the scan. The scan was carried out at a webcam angle of 15^{o} relative to the line laser. Ideally, the red threshold can be increased and the scanning can be carried out in a dark room or enclosure to get better scan results.

Challenges:

I encountered the following 'major' challenges during my project:

a) The webcam does not capture very high resolution images, nor is its color contrast very good. Hence, it was hard to differentiate the reds of the laser from the spectrum of pink/orange-brown. Running the scanner in a dark environment or one with sharply different colors, such as a green, blue, white or black background solves this problem.

b) I could not get microstepping (pulse width modulating the driver) to work. As a result my resolution was limited to 1.8^{o} resolution. This limits my ability to scan larger objects which I may not necessarily be able to bring very close to the laser.

c) Lens curvature and distortion was an aspect I did not anticipate (Figure 5). This requires additional corrections in Processing sketch B.

Future work:

As part of my future work, I plan to work on the challenges I encountered. Specifically, since I have a working proof of concept, I would like to apply for grants to get additional funds to build a working scanner with better resolution. Towards this end, I will master pulse width modulation and microstepping to obtain high resolution scans. I will also use a better camera and encase my setup in a dark box so that it is completely standalone for scanning small objects. I will also make my point cloud setup more user-friendly by using Python and creating a wireframe file that can be used with both 3D printers as well as for scientific analyses, by incorporating color gradients.

Instructables:

http://www.instructables.com/id/3D-Environment-Laser-Scanner-From-Scratch/

http://www.instructables.com/id/DIY-Arduino-3D-Laser-Scanner/

Media

Slides (PDF)

Slides (PPTX)

Code (ZIP)

Laboratory Temperature and Humidity Monitoring System - Robert Valdez

The problem

Small variations in both temperature and humidity have been known to disrupt and/or influence various laboratory-based parameters that are at the core of rock and sediment mechanics research. More specifically, we often observe 1) temperature-based fluctuations in pore fluid pressure during long-term (i.e. weeks to months) deformation and consolidation experiments and 2) humidity-based changes in frictional behavior during short-term (i.e. a few to several hours) unconfined shearing experiments. This project aims to continuously monitor the ambient conditions in our laboratory to help improve the quality of our data.

Instrumentation and setup

The main components required for this project are all readily available from SparkFun and are as follows:

1. Sparkfun RedBoard
2. Atmospheric Sensor
3. LCD screen
4. LED lights
5. 10k ohm resistors
6. 330 ohm resistors
7. 10k trimpot
8. push button
9. Jumper wires

All of the components listed above, excluding the atmospheric sensor, can be obtained by purchasing the sparkfun inventors kit.

The electrical build (Figure 1a) is rather straightforward, with all of the components being controlled by the RedBoard. The 16x2 LCD screen and the pushbutton shown in the schematic are required to display the various parameters (temperature, humidity, atmospheric pressure, and altitude) measured from the atmospheric sensor. The main parameters (i.e. temperature and humidity) are shown on the LCD screen as the default setting, but when the pushbutton is active the text on the screen changes to pressure and altitude. The primary function of the LED lights is to notify the user when the system is successfully running (solid green light) and to show the output rate of the atmospheric sensor (yellow blinking light). This system is then placed in an enclosure to allow for mobility and easy deployment into different environments (Figure 1b).

Results

An example of data collected from this setup is shown below (Figures 2-3).

To test the functionality of this system, I exposed the atmospheric sensor to different temperature and humidity conditions. This took the form of a common freezer and conventional oven. Not surprisingly, the temperature starts to decrease with time then increase once the ambient conditions were changed (i.e. from the freezer to the oven). The relative humidity shows a similar pattern, with low values in the freezer and higher values near the oven. The

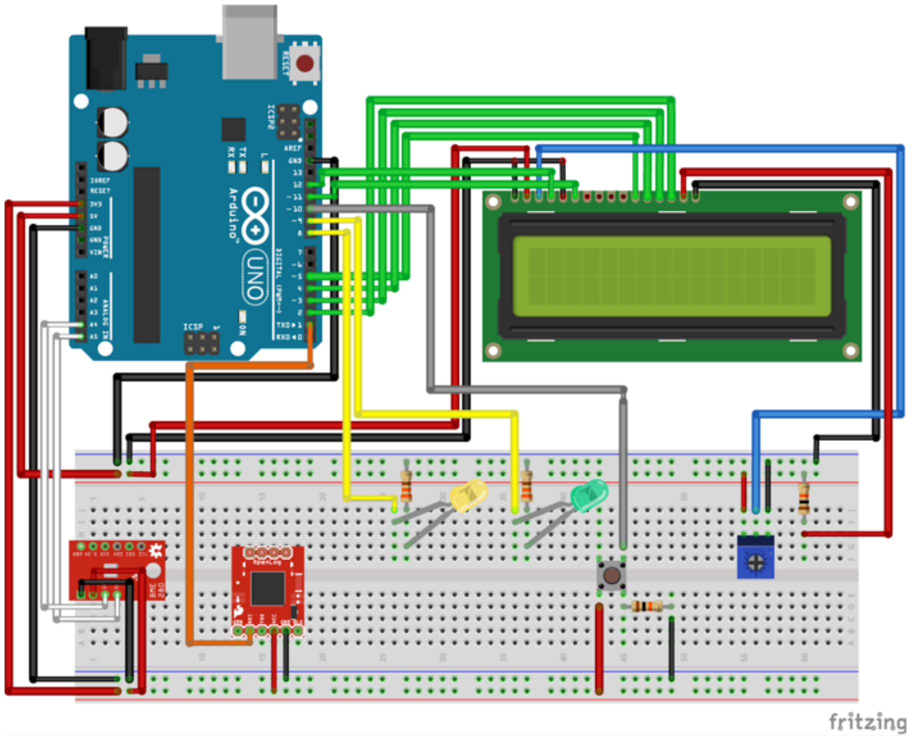


Fig. 1.90: Figure 1a - Schematic of the electrical circuit for this project.

Fig. 1.91: Figure 1b - Picture of the system being connected to a computer with a USB cord.

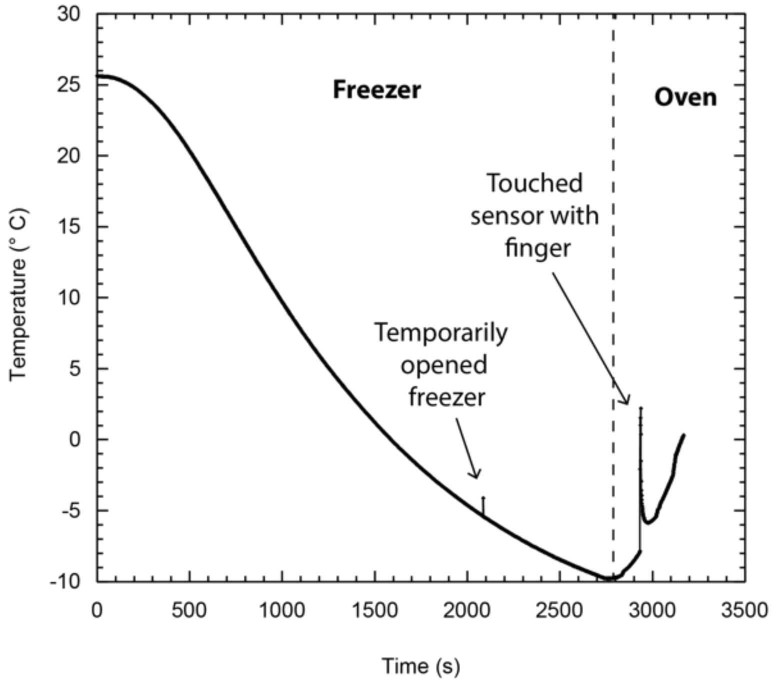


Fig. 1.92: Figure 2 - Temperature recording over 1 hr time window. The monitoring system was exposed to cold (freezer) followed by warm temperatures (oven).

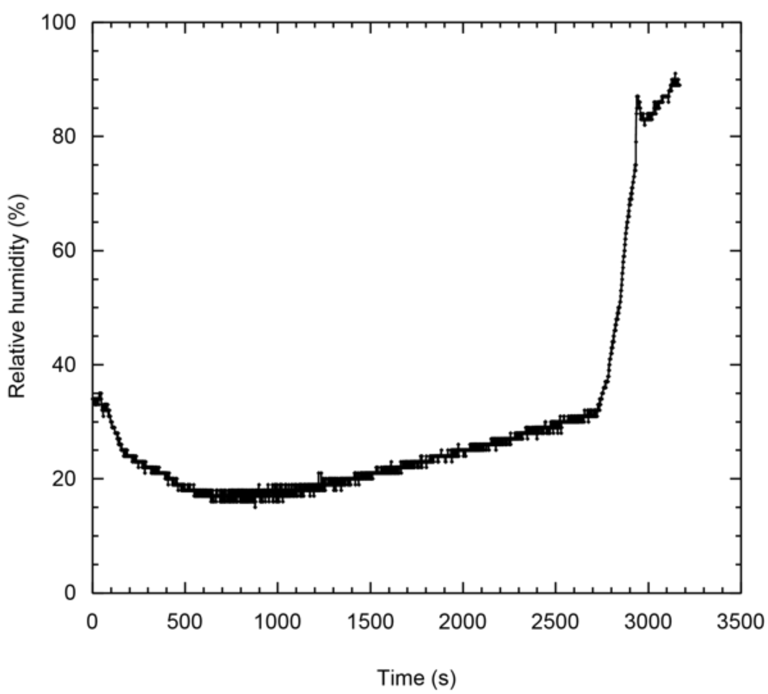


Fig. 1.93: Figure 3 - Relative humidity recording over the same 1 hr time window.

drastic increase in humidity could also be a result of condensation build up in the enclosure after exposing it higher temperatures.

Challenges

The challenges encountered in this project mainly revolved around 1) failure to implement other instruments into this monitoring system and 2) coding errors. I was not able to successfully implement the data-logging instrument into the system (i.e. https://www.sparkfun.com/products/13712), which made this system a "monitoring" rather than a "logging" system. However, I was able to actively record the output of the atmospheric sensor using the CoolTerm application. This application (http://freeware.the-meiers.org) allows the user to connect to the monitoring systems serial port and write its output to a text file. The data shown in Figures 2-3 were collected using this software while the systems was running. The coding errors are also linked to the data-logging instrument problem in that I was not able to successfully get these different components to sync with each other.

Future work

As part of my future work, I plan to address the challenges listed above. Specifically, I'd like to record data without actively using an outside application. Once the system is successfully recording data, I'd like to implement a wifi module so that this data is readily available online. This can be completed using the SparkFun Thing (https://www.sparkfun.com/products/13231), which allows the user to upload data to a server (i.e. https://data.sparkfun.com/). Once it is available online, I'll plot this data in real time so that others experimentalists at Penn State can view the temperature and humidity logs at their leisure. Lastly, I'd like to have this completely battery powered so that it is no longer tied to a computer.

Media

`Slides (PDF)`

`Slides (Keynote)`

Deformation-measuring Module - Chaoyi Wang

Abstract

A simple force measuring automation system is constructed to measure the deformation (strain) from a wooden cantilever bar structure. The system consists of two Arduino Uno boards with corresponding circuits, a servo motor connected to a wooden arm, and a cantilever-shaped load frame, a wooden bar is then pulled by the servo motor via the wooden bar through an elastic string. Specifically, one Arduino Uno board is programed to control the servo motor, the other Arduino Uno board is programed to register strain signals from the attached strain gages; the motion pattern of the servo motor can be either controlled by the swipe potentiometer or by pre-written scripts on computer; the strain gages are constructed with full-bridge configuration. As the servo motor pulling the wooden bar, the deformation and resultant force can be either observed from the LCD display on the Arduino Uno board or from the computer screen. This simple force-deformation measuring system is programed to be easily expanded and improved should there are more complicated applications.

Introduction

We measure all kinds of mechanical quantities in the rock mechanics lab, for instance, force, displacement, and acceleration, etc. Most of the measurement systems contain sensors and data acquisition systems. However, most systems are not self-sustainable, which means carrying out different test procedures automatically according to a pre-set scheme, giving out warning/shutting down the system automatically when necessary, and pushing notifications of test status to operators if the operator is not on site. Such automation is needed especially in long-duration experiments such as creep shear and slide-hold-slide.

Through this semester, we have learnt how to construct experimental apparatus from scratch, including building both hardware and software parts of the apparatus, signal conditioning, and data acquisition. In this project, a simple force measuring automation system is put together using the techniques learned from the class. The system consists of two major parts, hardware and software. The hardware part is composed of two Arduino Uno boards with preconnected circuits (one board controlling the servo motor, the other serve as a signal conditioner and data logger), a wooden load frame, a servo motor with a customized wooden arm, and a subject wooden bar which can be deformed by the servo motor. The software part is a piece of self-written program (LabVIEW), this program features functionalities of data displaying, conditioning, recording, and performing schematic loading sequences.

Combining the hardware and software part, this simple force-deformation measuring system is a prototype of more comprehensive-complicated experimental set ups used in rock mechanics laboratories. The system can be easily modified and expanded for additional functionalities.

Apparatus Configuration

Part 1. Hardware

1. Load frame

The load frame is constructed out of oak and pine strips purchased from Lowe's. The completed frame is shown in Figure 1.

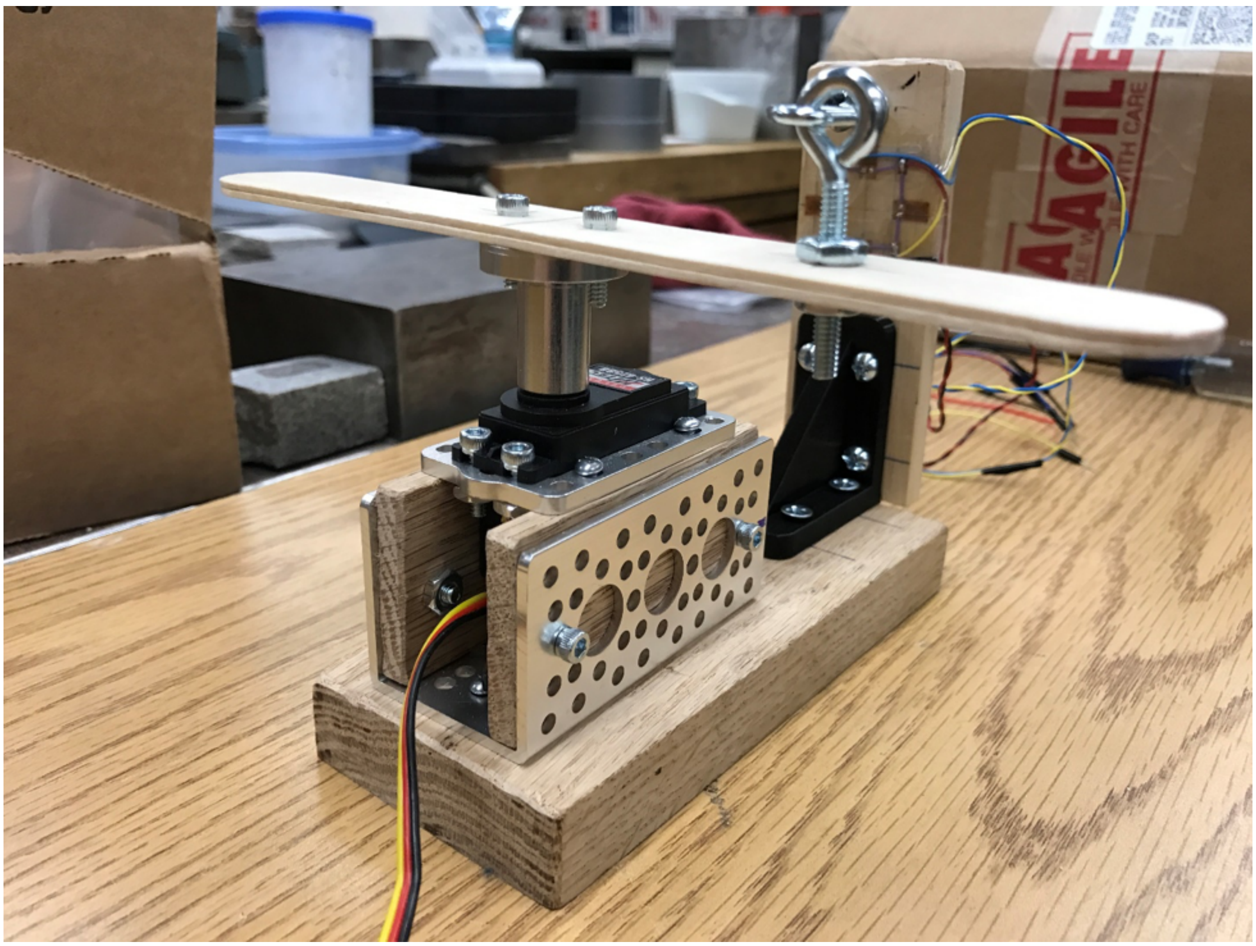


Fig. 1.94: Figure 1. Load frame: the material of the bottom piece is oak wood; an aluminum servo motor holder is fixed by two screws on the bottom piece; the servo motor is hosted by a bracket and sandwiched between two pine plates as shown; the whole sandwich structure is then attached on the servo holder (a larger bracket) via two bolts on each side.

There is a bar holder also fixed on the bottom piece, this triangular shaped bar holder is made by 3-D printing (Figure 2), the design is fairly simple, the holder consists of two plates sitting perpendicular to each other with a supporting rib connecting each other. Since this part is 3-D printed, therefore the constructing material is filler plastic. One may question the stiffness of this bar holder due to its material, however, as we are going to deform a thin pine wood bar, the rib could provide adequate stiffness and support in this specific application discussed here.

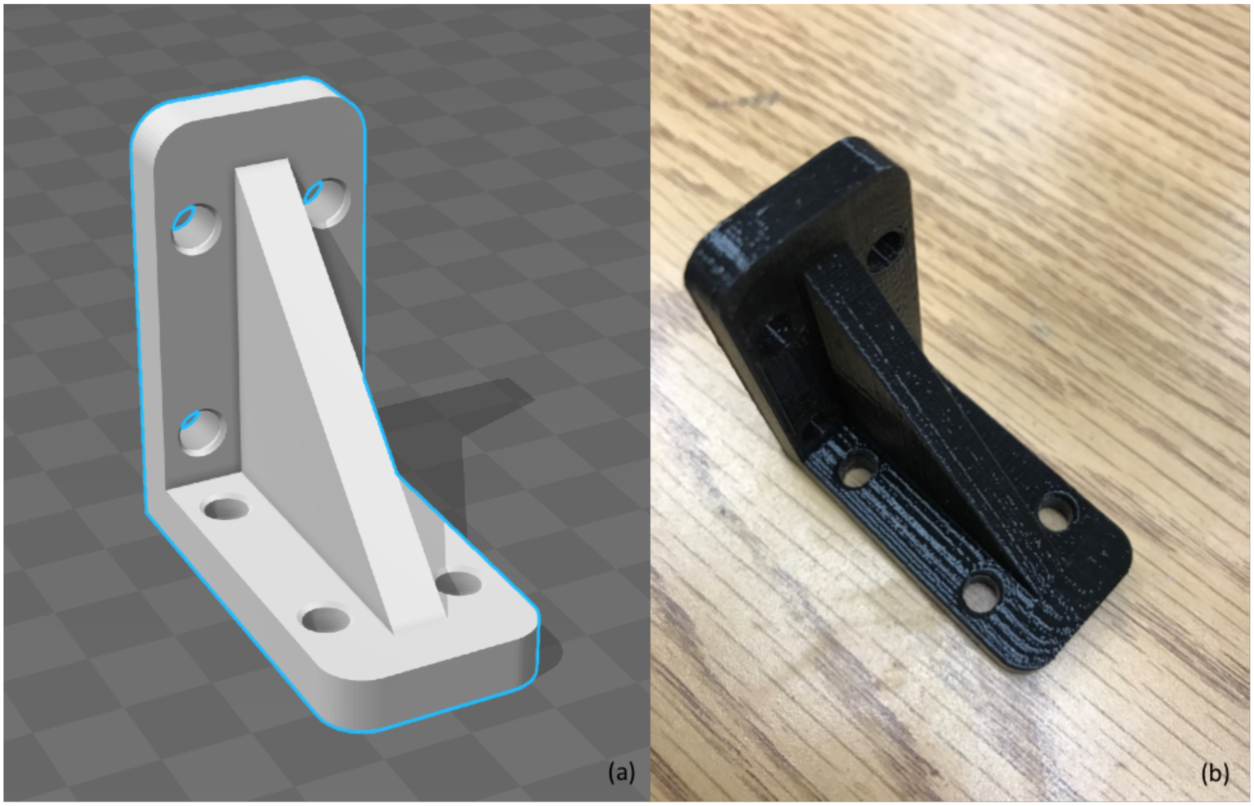


Fig. 1.95: Figure 2. 3-D printed bar holder: (a) Onshape model of the bar holder; (b) 3-D printed sample holder.

2. Wooden bar and strain gage configuration

The subject we are going to deform (bend) in this application is a pine wood bar (Figure 3). The dimension of the bar is NNxNNxNN. Four bolts are used to fix the wood bar to the bar holder though the configuration shown in Figure 3.

3. Servo motor

The model of the servo motor is Hitec HS-425BB (see the appendix for the link to the data sheet), the motor in assembly is shown in Figure 4. The maximum torque of this servo motor is 45.82/56.93 oz-in (4.8/6.0V).

4. Arduino Uno Board

There are two Arduino Uno boards used in this application, one Arduino board (AB1) is configured to control the servo motor and features a switch function from manual mode and programmed mode. To be specific, one can control the servo motor by slide the swipe potentiometer (voltage divider circuit) associated with the circuit, or control the servo motor by a set of pre-defined command scripts using serial communication. The other Arduino board (AB2) is configured as a signal amplifier and can register data to a personal computer. The two Arduino boards and the corresponding circuit is shown in Figure 5.

Part 2. Software

1. Data acquisition program

The data acquisition program “one_channel_serial_reader.vi” is written in LabVIEW 2015, the front control panel and the block diagram is shown in Figure 6 and Figure 7. This program can record data from the load frame to the local computer, and can read a script containing instructions for the servo motor to move.

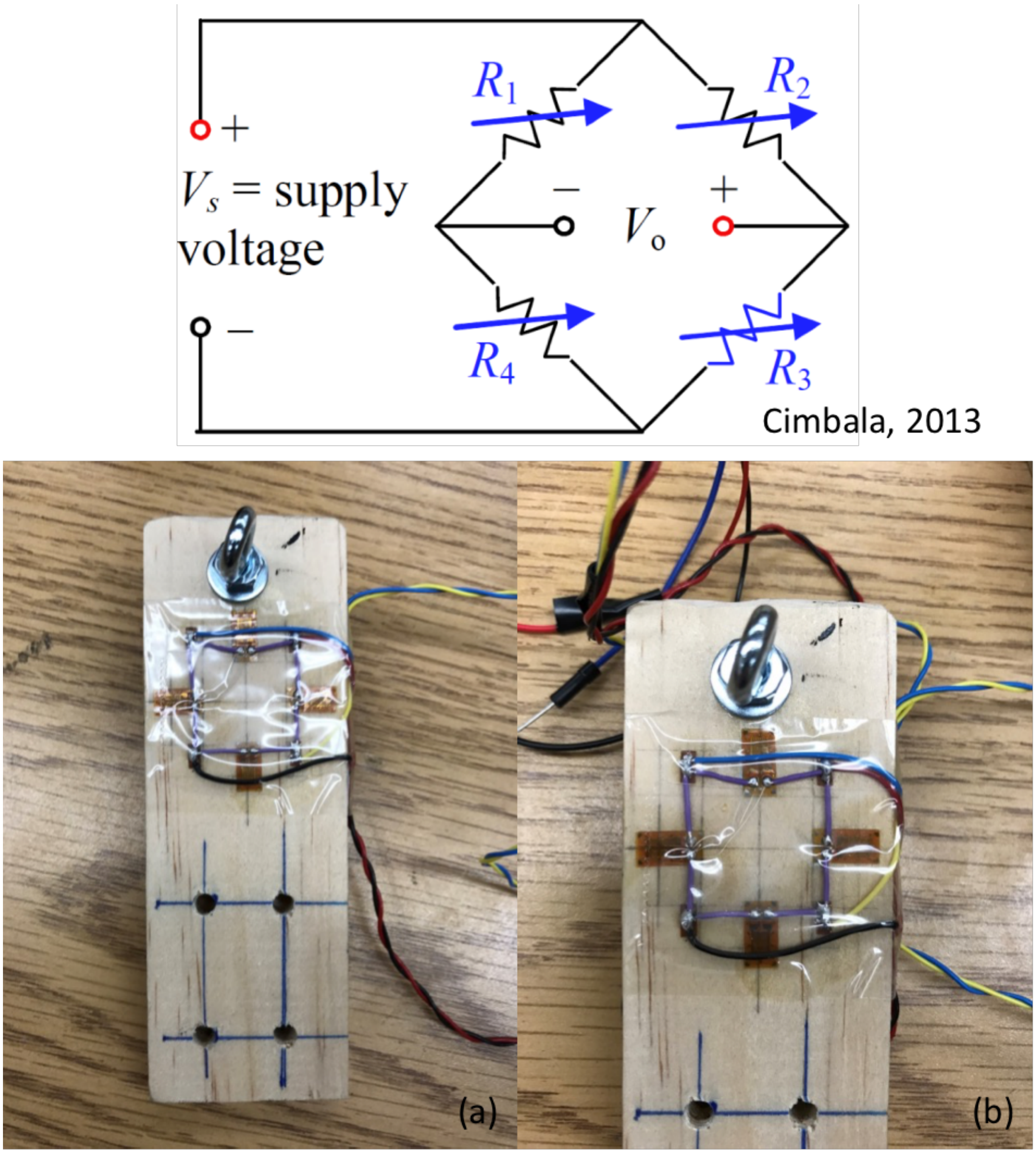


Fig. 1.96: Figure 3. Pine wood bar used in this application: full-bridge configured strain gages are attached on the upper portion of the wood bar; four bolts are used to fix the lower portion of the wood piece firmly on to the bar holder.

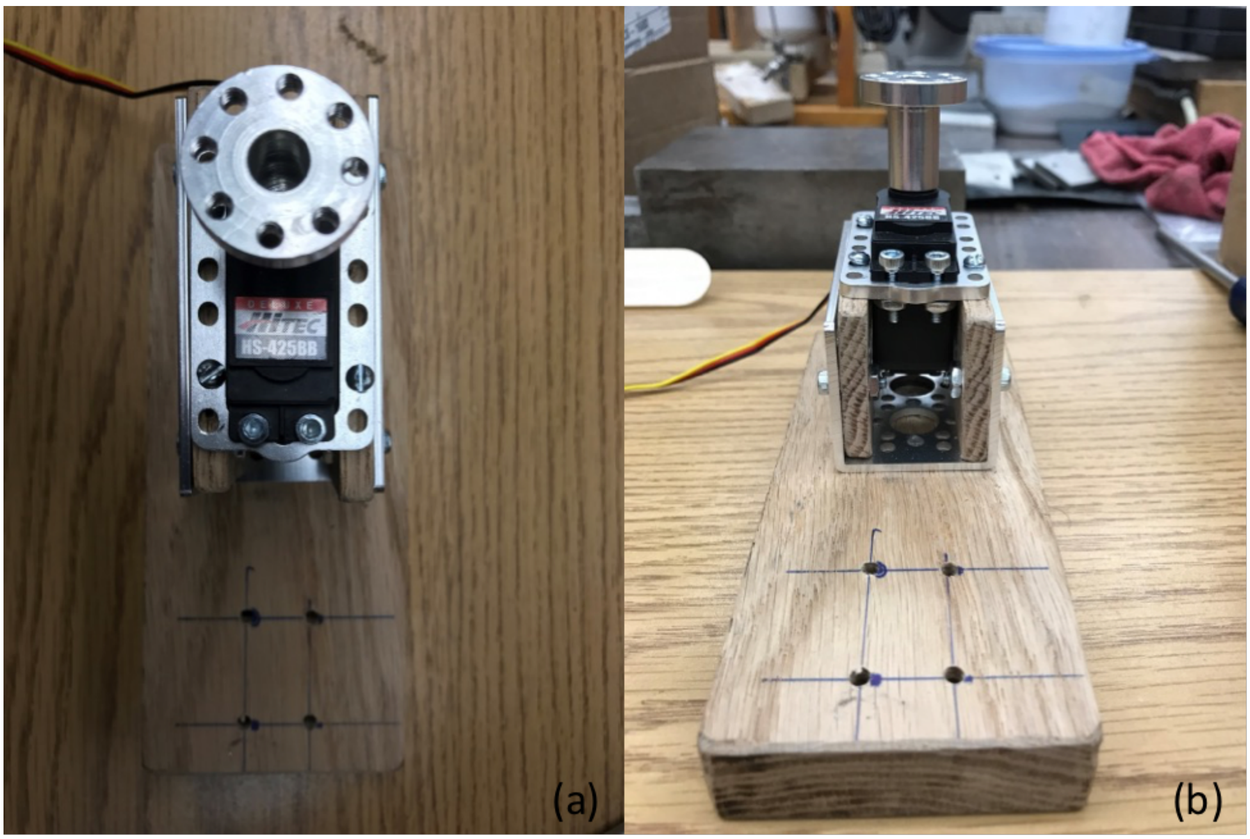


Fig. 1.97: Figure 4. Servo motor used in this project: (a) top view of the servo motor mounted in bracket; (b) side view of the servo motor mounted in the bracket.

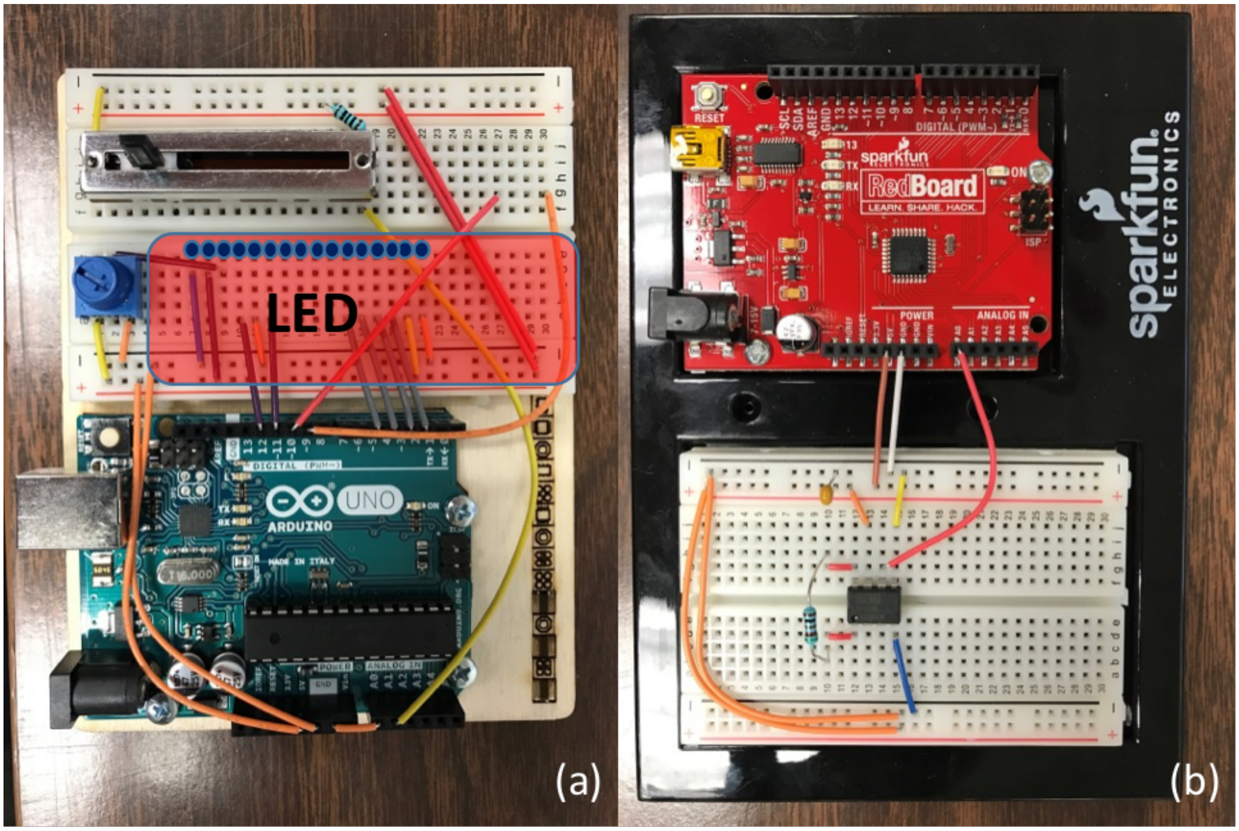


Fig. 1.98: Figure 5. Arduino boards and corresponding circuit diagrams: (a) AB1, this Arduino is used to control the servo motor in manual mode (use the swipe potentiometer) or in automatic mode; (b) AB2, this Arduino board is used to acquire analog signal, amplify the signal, convert the analog signal to digital bits, and register the digital data onto a local computer.

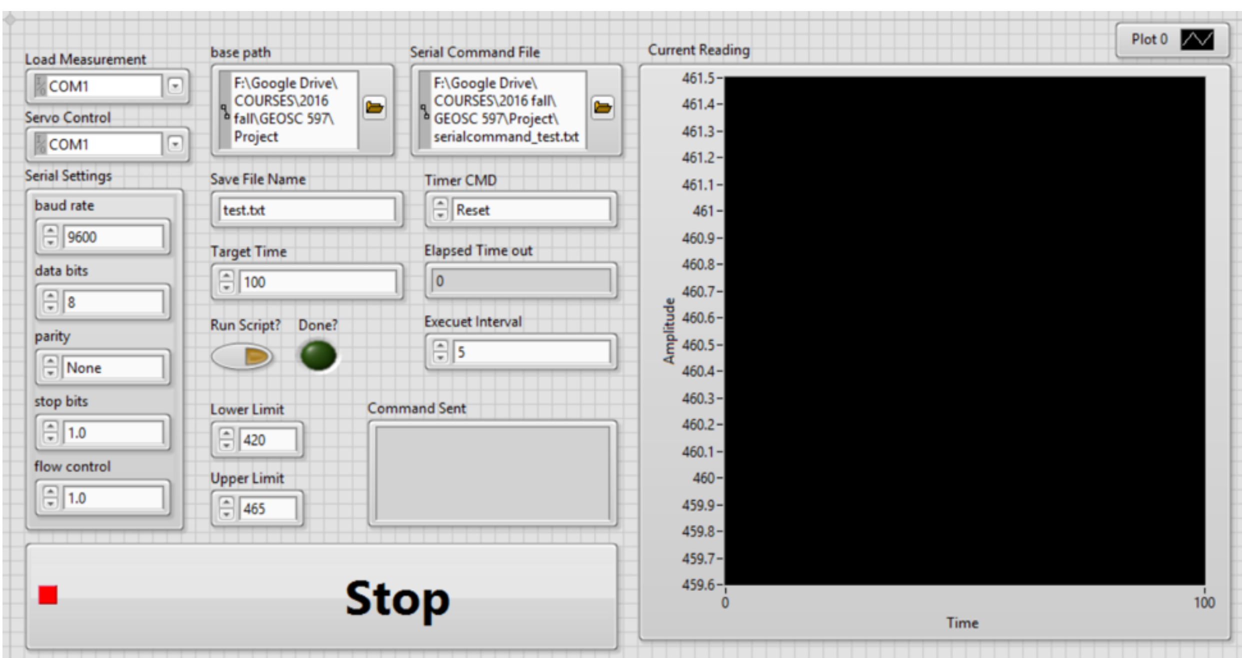


Fig. 1.99: Figure 6. Front control panel of "one_channel_serial_reader.vi".

2. Servo motor control module

The servo motor is controlled either manually or by reading a pre-written file containing instructions to the motor. In the manual mode, the servo motor will rotate according to how much voltage is divided by the swipe potentiometer. The potentiometer is set in a voltage divider circuit, as shown in Figure 8, and the range of voltage measured from the two ends of the potentiometer is approximately 0 to 4.95 V.

If automatic control mode is chosen, the program will execute instructions form the command file one at a time according to a user-defined execution interval.

Experimental Performance

Part 1. Loading manually by a swipe potentiometer

By sliding the swipe potentiometer, the Arduino board will convert the voltage measured from the voltage divider circuit and map the value to a range of 0 to 180 degrees. The position instruction is then send to the servo motor to execute. The rotation of the motor is continuous, since the resistant change in the swipe potentiometer is continuous. The LCD display will display the approximate angle the motor has rotated. Figure 8 shows an image of LCD display while the motor is rotating in manual mode.

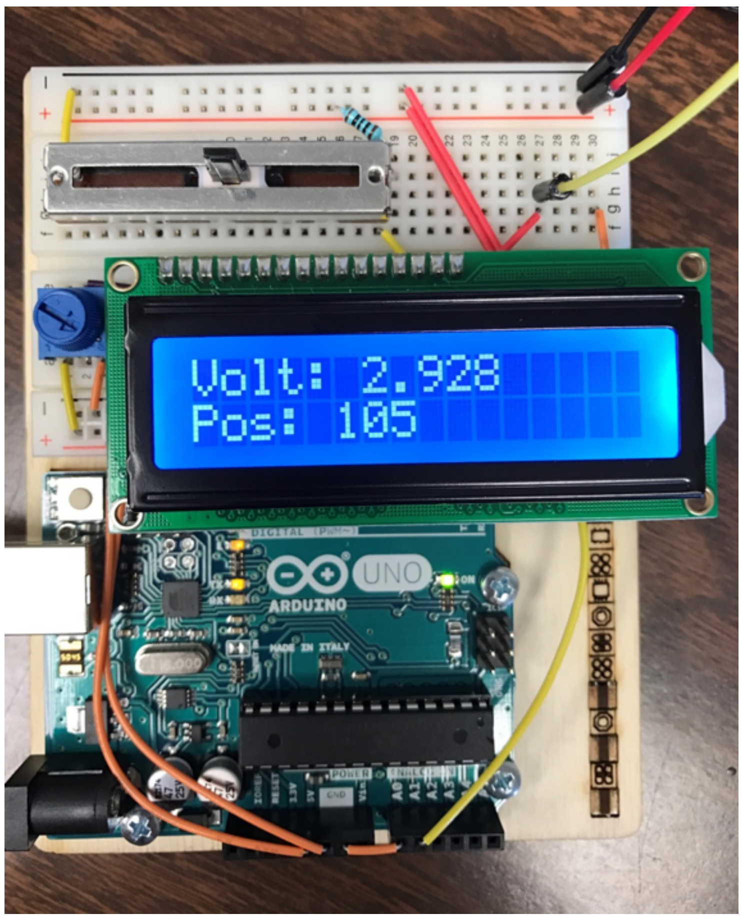


Fig. 1.101: Figure 8. Servo motor rotates in manual mode.

Part 2. Loading automatically by a pre-written script

When choosing the automatic mode, the “one_channel_serial_reader.vi” program will read a text file containing serial commands to the motor. The commands are composed of a simple absolute degree value followed by a semi-column as a delimiter. A sample command file is shown in Figure 9.

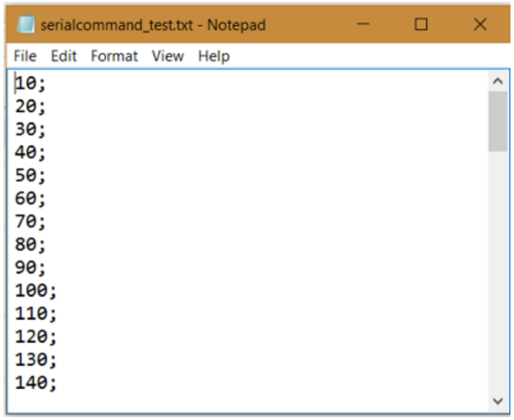


Fig. 1.102: Figure 9. Sample command file "serialcommand_test.txt": the file is composed of multiple lines of commands; each command contains a rotational degree value and a semi-column as a delimiter.

Part 3. Automatic shut down and overstress shut down threshold

The program can shut down by itself if certain criteria are met, such criteria are listed as following:

1. The scheduled experimental time has been exhausted.
2. The deformation reading exceeds either the pre-set upper limit or lower limit.
3. Should any internal program error occur.
4. The emergency stop button is pressed.

System Expandability

This system is a simple combination of hardware and software design in rock mechanics experimentation. The system can be easily modified or expanded either from software or hardware. The following example expansion scenarios are described:

1. Hardware: the servomotor can be replaced by an actuator or continuous motor in different applications.
2. Hardware: the wooden bar can be any material of interest such as rocks, polys, and metals. Under the assurance of higher stiffness of the sample holder.
3. Software: timing modules can be expanded to accommodate not only uniform time interval executions but non-uniform timing schemes.
4. Hardware and software: a cellular module can be added to the Arduino board 2, this cellular module can be programed to send out messages/emails to operator if the operator is not present onsite during automatic runs.

There are many more possibilities for this system to be expanded, however, due to the limited amount of resources, not all of them are described or addressed here.

Summary

In this specific application, a simple comprehensive automation system is built to accommodate the purpose of deformation measurements. The system consists of a hardware part and a software part, For the hardware part: a wooden arm driven by a servo motor pulls a wooden bar with an elastic string manually or automatically; the wooden bar is mounted on a 3-D printed bar-holder; full-bridge configured strain gages attached on the wooden bar are used to measure the induced deformation. For the software part: a LabVIEW program is developed to establish communication between the Arduino boards and a local computer; the program can register and record the deformation readings from the load measuring cell and control the servo motor by sending serial commands to the corresponding serial ports.

Acknowledgements

Hereby I want to express my sincere acknowledgements to Dr. Chris Marone and course instructor Dr. John Leeman for their patient instructions and inspiring lectures. And all people in the rock mechanics lab who tolerated me for doing wood work and making mess around.

Media

`Slides (PDF)`

`Code (ZIP)`

CHAPTER 2

Contributors:

- John Leeman, (@jrleeman)
- Chris Marone, (@cmarone)
- Martin Bentley, (@mtb-za)
- Kerry Ryan, (@kerryryan)