Earl E. Bakken
MEDICAL DEVICES CENTER

UNIVERSITY OF MINNESOTA

Prototyping Tutorial Handbook

MDC Equipment Specific Tutorials

Lathe
Milling Machine
Micro-Welder
Lab Oven
Soldering
Drill Press
Band Saw
Laser Cutter

Please note that you can also find these tutorials on the MDC Safety Training
<table>
<thead>
<tr>
<th>Tutorial</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laser Cutter Tutorial</td>
<td>5</td>
</tr>
<tr>
<td>CAD Tutorial</td>
<td>32</td>
</tr>
<tr>
<td>3D Printing Tutorial</td>
<td>61</td>
</tr>
<tr>
<td>Machining Tutorial: C-Clamp</td>
<td>87</td>
</tr>
<tr>
<td>Machining Tutorial: Screwdriver</td>
<td>114</td>
</tr>
<tr>
<td>Matlab Tutorial</td>
<td>133</td>
</tr>
<tr>
<td>3D Scanning Tutorial</td>
<td>146</td>
</tr>
<tr>
<td>Arduino Tutorial</td>
<td>157</td>
</tr>
</tbody>
</table>
Project Checklist

Laser Cutter
Computer Aided Design (CAD)
3D Printing
Machining: C-Clamp
Machining: Screwdriver
Matlab
3D Scanning
Arduino
Safety Reminders

Requirements for working in the Mechanical Prototyping Lab
- Safety Glasses (There are large ones available to put over your glasses)
- Long Pants/Skirt
- Closed toe shoes
- Long hair tied back
- If you are wearing any scarves, necklaces, or dangly bracelets, please remove these or tuck them into your clothing before working with moving equipment, especially drills.
- You may also need hearing protection, masks, and/or gloves depending on what you are doing.

Requirements for working in the Electrical Prototyping Lab
- If soldering
  - Safety glasses
  - Fume extractor
  - Long pants/skirt
  - Closed toe shoes
  - Tie long hair back
  - Also, remember to wash your hands after touching lead solder.
- If gluing
  - Safety Glasses
  - Gloves
  - Closed toe shoes
  - Long pants/skirt
  - Tie long hair back
  - UV protective glasses (if using UV cure system)
- In general, you do not need special equipment if you are using 3D printers or computers, however there are many hazards, like 3D printer hot ends, knives, and hot soldering irons to be aware of while in the electrical lab.

Requirements for working in the Biosafety labs
- You will need to do the additional DEHS training
- You may wish to get a hepatitis B vaccine (if you do not have one already)
  - This is mainly for working with human tissues. We very rarely work with human tissues here, and most of those come from donors who are screened. Nevertheless, this is a standard precaution.
- Lab Coat (we have some)
- Long pants/skirt
- Closed toe shoes
- Goggles
- Gloves
- Tie long hair back
Laser Cutter Tutorial

University of Minnesota
Driven to Discover®
Name Tag Prototyping Activity

Goal:
This project is meant to introduce you to the Bakken MDC laser cutter, while you introduce yourself to us by creating a nametag FOR USE in the Bakken MDC that expresses who you are in some way. You will be using CorelDrawX7 as well as the laser cutter for this assignment.

Requirements:
- Include your first name (last name optional)
- Include “Bakken MDC Intern” somewhere on the name tag
- Try to use at least 5 different pieces of equipment/software to make your name tag (eg.
  CorelDraw, laser cutter, hot glue gun, spray paint, sandpaper
- Beyond creating your name tag physically, it would be great practice for you to also create a model of your name tag in SolidWorks
  - Cool features to possibly use
    - Sketch a picture
      - [https://www.youtube.com/watch?v=bByetsIhmK8](https://www.youtube.com/watch?v=bByetsIhmK8)
      - [https://www.youtube.com/watch?v=XIIcbUJ0tyw](https://www.youtube.com/watch?v=XIIcbUJ0tyw)
    - Working with text
      - [https://www.youtube.com/watch?v=tVDHC5iVfGQ](https://www.youtube.com/watch?v=tVDHC5iVfGQ)

Assignment:
- Read through the laser cutter tutorial powerpoint.
- Find time to come in to the Bakken MDC to use the laser cutter and materials.
- If you are looking for ideas, help with a piece of equipment, or looking for a specific laser cutter housed at the Bakken MDC is older, and will also be very busy, you may also use the laser cutters in Anderson Labs (2134 Mechanical Engineering). The Anderson Labs are open from 10-6 M-F and you do not need special training to use the laser cutters. Close toe shoes are required in their space.
- Take your time with this tutorial and work through it carefully.
- When finished, obtain a magnet from lab staff.
- Wear your nametag whenever you are around the center!
- Please note that it is not a requirement to use the laser cutter, however this is a very good introduction project for the laser cutter and you may want to use it later.
- Your final deliverable will be a name tag with “Bakken MDC Intern” and your name somewhere on it. You must use 5 or more tools and exercise some creativity!
- Be creative and have fun!

Resources:
- You have the entire Bakken MDC facility at your disposal
- Any materials lying around are fair game, most are on the gray shelf in the back of the mechanical lab.
- The magnets, spray paint, acrylic, and wood will be in the black cabinet under the miter saw.
- There are a variety of spray paints in the cabinets below the fume hood.
  - If you are going to use spray paint, use it in the fume hood or outside. The mechanical lab itself is poorly ventilated and the spray paint smell will linger.
  - Use as much brown paper as you need to place under and behind your material, please ask lab supervisors.
- A very good resource about putting pictures on wood: http://support.epiloglaser.com/article/8205/50104/photo-processing-for-engraving-on-wood-coreldraw-x7
- If you would like to learn more about the laser cutter:
  - This is a great document created by The Art Institute of Chicago with a great overview of the Laser Cutter and the do's and don'ts.
- Some good videos or webpages to learn the basics of CorelDRAW:
  - www.lynda.umn.edu
    - CorelDRAW Essential Training
      - Focus on:
        - Importing files
        - Chapter 5, Working with objects
        - Chapter 9, How to duplicate and align multiple objects
        - Ch 11, Working with Color (NOTE! Laser cutter can only read RGB Colors, so make sure that if you are using colors they are RGB and not CMYK)
        - Ch. 12 - Working with Outline Types and Color (To vector cut lines, the line should be red, and must be Hairline in thickness)
        - Any other sections that you think you would find helpful to create your file
    - This is a much older version of CorelDRAW, but the quick trace tool still exists in Corel under the Bitmaps drop-down menu.

Examples of some stellar nametags:
Laser Cutter Tutorial

Bakken MDC
Safety Concerns

When working in the Mechanical Prototyping lab, you must:

- Wear safety glasses (provided, larger ones available to go over glasses)
- Wear long pants or a skirt
- Wear closed toe shoes
- Have long hair tied back
- Tuck in or remove scarves, necklaces, or dangly bracelets

You may also need (all of these available in the lab):

- Hearing protection
- Masks
- Gloves
Important Reminders

- Power on the fume extractor before operating the machine
- If there is a fire, stop the cut and increase the speed or lower the power
- Do not walk away from the laser as it is cutting
- You can increase the number of copies of an outline if the vector cut doesn’t go through (just keep the piece in the EXACT same spot)
- Do not leave the machine unattended during a cut
- Do a sample cut before cutting the actual design to ensure the settings are correct and so you don’t waste material
- If you are unsure about how to operate the machine, ask for help before using
1. Power on Laser Cutter

Power on the system using the switch on the bottom right side of the laser cutter.

Always turn off when you are done!
2. Power on Fume Extractor

Power on the fume extractor that is located on the Left side of the laser cutter. All you need to do is press the power button and make sure you hear it turn on. Always turn off when done!
3. Calibrate bed height using distance tool

Once the fume extractor and laser cutter are powered on and the system has initialized, press the Z button on the laser cutter. This will move the laser to the top left corner of the bed. Make sure you don’t place your acrylic or piece that you’re cutting on the table until after you’ve pressed the z button.
Caution!

When moving the bed to calibrate, **never** move the bed **up into the laser** or **all the way down**.
The end stop sensors are damaged and will let you do this.
DO NOT! It is very damaging to the laser cutter.
4. Adjust the height of the bed

Place the acrylic or wood piece inside the laser cutter on the bed under the laser. Use the up and down arrows to move the bed to the correct spot on the distance tool. (See picture on previous slide) The tip of the laser should be a small distance from your workpiece. Note that you can use the select button to move between the ones and tens place for how much the bed will move each time you press the up or down arrows.
5. Remove distance tool and close laser cutter

Now that you’ve calibrated the distance between the tip of the laser and the workpiece, it is time to design your nametag in CorelDrawX7 by opening the program and starting a new document.
6. Design name tag in CorelDrawX7

Create the outline of your nametag by using the shapes on the left toolbar or by importing .dwg file from a SolidWorks drawing. Keep in mind that the dimensions are shown as well as the units. Make sure you do not cut outside of where your part is located.
7. Change the outline to hairline thickness for vector (or through) type cuts

When using the laser cutter, you can raster (aka etch) or vector (aka through) cut. Vector cuts require hairline thickness. Once you have drawn your outline, make sure the object properties toolbar is showing by going to Object > Object Properties. You should see something like this, however, try to make your outline more unique than a simple rectangle.
8. Add text to your nametag

In order to add text, you’ll use the textbox tool on the left toolbar. It is represented by the letter “A”. Drag a textbox inside your nametag and type in the characters you want shown. Because the text will need to be raster (etched) cut rather than vector cut, change the text color to red.
9. Add an image to your nametag

First, paste the image you'd like to use into your project. Click on the image and press the “Trace Bitmap” button. Then, select “Outline Trace” and “High Quality Image”. Adjust the settings until the image appears the way you would like it. Make sure to check all of the checkboxes. Click OK.
10. Move image within nametag outline

Position any images and text inside the outline of your nametag to prepare for sending to the laser cutter. Click on the image and make it the same color as the text because we also want to raster cut (aka etch) the image onto the acrylic or wood.
11. Save your CorelDraw file and set up for the cut

Go to File > Save As and save your file with a unique name. Once you’ve saved your file, click File > Print. Then, select Preferences.
12. Adjust the laser speed and intensity for each color

Cycle through the “Pen Mode” setting for each color that is not present in your design and set it to “Skip”. For any cut that you want to go all the way through the workpiece, change the “Pen Mode” to “VECT”. I want to engrave the works and image into the part and then cut it out of a piece of acrylic. This is why I have selected black as VECT and red as RAST. Adjust the speed and power settings based on the type of material you are using. Always start very low with practice cuts on a small piece of acrylic to figure out the correct power and intensity. Once you’ve adjusted the speed and power and verified they’re where you want them, click OK.
Starting points for 1/8” Acrylic

For Vector cuts
- 100% Power, 3% Speed

For Raster cuts
- 100% Power, 50% speed
Power and Speed Settings for Acrylic Cutting and Engraving

**Laser Cutter Rules:**
- Never leave the laser cutter when it is running.
- Always open the cover, pause, or power down the laser cutter if you see any flames.
- Always use the fume extraction system when running the laser cutter.
- Never use PVC or other materials with chlorine constituents in the laser cutter. Always ask if you’re unsure about materials.
- Always ask a lab supervisor if you’re unsure of how to use the laser cutter.

**Laser Cutter Tips:**
- Always do a small test part to confirm the proper settings before laser cutting a large sheet to avoid wasting material.
- If the laser is not cutting as expected, let a lab supervisor know.
- Do not move your part from the laser cutter if you don’t think it cut all the way through with a vector cut. Just start the laser cutter again.
- If you’re starting fires with your settings, double the speed settings and “print” two copies instead.
- Tutorials and training resources are available on the computer desktop in a folder called “Laser Cutter Resources.”

<table>
<thead>
<tr>
<th>Vector Cuts (through cuts)</th>
<th>Raster Cuts (engraving)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness (in)</td>
<td>Speed (%)</td>
</tr>
<tr>
<td>1/16</td>
<td>9.8</td>
</tr>
<tr>
<td>1/8</td>
<td>3.6</td>
</tr>
<tr>
<td>3/16</td>
<td>1.8</td>
</tr>
<tr>
<td>1/4</td>
<td>1.1</td>
</tr>
<tr>
<td>1/2</td>
<td>0.4</td>
</tr>
</tbody>
</table>

\[ Y = 0.1318 \times X^{-1.574} \]

Use this equation or the above table to determine the required speed settings for a vector cut for a given sheet thickness. Always set the power at 100%.

\[ Y = \text{SPEED} \% \]
\[ X = \text{THICKNESS (in)} \]
\[ \text{Power} = 100 \% \]

\[ Y = 1.9107 \times X^{-1} \]

Use this equation or the above table to determine the required speed settings for a raster cut of a given depth. Always set the power at 100%.
13. Don’t damage the laser cutter!

It is important to always test the speed and power that you’re operating the laser at. If the speed is too low while the intensity is high, it can damage the laser or even start a fire. Alternatively, if the speed is too high and the intensity is too low, the laser will not cut all the way through the material. Try out different settings before you use a large piece of acrylic only to realize your settings were incorrect and you wasted an expensive piece of acrylic. You can cut out 0.5” diameter circles until you get the speed and intensity correct as a test.
Seriously, please don’t start a fire.

If there is a fire, stop the cut and adjust the settings. You can increase the speed and/or lower the % power. If the vector cut doesn’t go all the way through the material, you can increase the number of copies. Usually, small flames will die down quickly, especially because of the fume extractor. Get help immediately if they don’t stop a few seconds after you’ve stopped the laser.
14. Send your CorelDrawX7 file to the laser cutter

If you have an image, check the box “Print as Bitmap”. Then, click OK and then “Apply”. Now, your file has been sent to the laser cutter.
15. Find your file and start the laser

You should now see your file name on the display of the laser cutter. If you don’t, use the “PREV FILE” and “NEXT FILE” buttons to find it. Then, press the green start button. Watch your cut the entire way through to make sure no fires are started and that the laser isn’t being damaged. The “PAUSE” button on the laser cutter should stop the cut, but if it doesn’t, just power the laser cutter down.
It can be that simple!

There are many, many more ways to use the laser cutter to prototype and quickly manufacture 2D parts. Use online resources and ask other interns what they’re doing for their nametag. Be unique with your nametag. Have fun!
The University of Minnesota is an equal opportunity educator and employer.
Bakken MDC Internship Program

CAD Tutorial
Computer Aided Design (CAD) Assignment

Goal:
The goal of this project is to introduce you to Computer Aided Design through a Solidworks tutorial. At the end of this assignment, you will have a solid part file that can be machined or 3D printed.

Requirements:
- Be proactive and get started right away on this assignment.
- Read through this assignment all the way through before beginning (important!) so you know what resources are available to you for assistance.
- Figure out where you are going to be working on Solidworks. You're more than welcome to use the Bakken MDC workstations, but they will be very busy in the coming weeks. If you have your own computer (must be a windows machine) and you're a CSE student at the U of M, you can download a student copy of Solidworks here for your personal computer.
- A VPN version of solidworks through the CSE Labs apps portal is available to work from home as well: https://apps.umn.edu/vpn/index.html
- U of M students may also use the CSE Anderson Labs in Room 308 in the Mechanical Engineering building. If you are not a U of M student, you will need to use the computers in the Bakken MDC Electrical Lab.
- The tutorial will walk you through how to make a screwdriver. You will need to choose a different tool you might find in the Bakken MDC.

Assignment:
- Instead of doing two separate tutorials, we will all be doing the same tutorial, but at varying difficulties.
- This assignment requires you to find a tool and a caliper, and using the caliper, create a 3D rendering of that tool. If you are new to CAD, a simple tool with few curves and points is suggested. If you are comfortable with CAD and Solidworks, you can choose a more interesting tool.
- When you are finished, you will be putting your part(s) in a folder in Google Drive.
- Your tool may require you to make separate parts and ‘assemble’ them. If this is the case, please include all individual parts, as well as the assembly file in the Drive folder.
- The final deliverable of this tutorial will be a rendering of your CAD part that is not the tool shown in the tutorial, as well as a 3-5 second animation from Solidworks.
- Next week, these will be 3D printed, so the more detailed the file you make, the more functional of a tool you will have!

SolidWorks Resources
- YouTube is the best place to learn how to use CAD
- There are old tutorials from the Bakken MDC available as well.
- "SolidWorks 2018 Essential Training" tutorial from www.lynda.umn.edu For those of you who don’t know about lynda, it is a great tutorial site we have access to through the University of
Minnesota. Just follow the link above, sign in with your UMN credentials and search for “SolidWorks 2018 Essential Training.” This course covers a wide variety of topics organized in 2-10 minute video sections.

- Tutorials inside SolidWorks (Under Help->SolidWorks Tutorials)
- Google is amazing as well
Make a sketch before beginning in Solidworks

This is good practice and helps you understand the geometries of your part. Be sure to add a name, date, and page number to your drawing. Here’s an example:
Starting a new part

- When you open Solidworks to make a new part, you will go to File, New, Part, and OK
- Make sure you have the metric system on, shown in the bottom right corner. Click there to change. We will work in MMGS, which stands for millimeters, grams,
Using a Caliper: Zeroing

- When you get out a caliper, before you use it, you’ll want to make sure that when it is closed, it is at zero, you can do this by pressing the ‘Zero’ button.
- You’ll also want to make sure it is in mm, which can be changed by pressing the mm/in/F button. You will want to use a digital caliper, not a dial caliper, because most dial calipers are in hundredths of an inch.
3 Ways to use a Caliper

Calipers can be used to measure around an object, in a space, or a depth.

In this example, all of these measurements are 17.25 mm.
Getting Started

This example will feature a basic flathead screwdriver.

- As you can see from the picture, there are many small details on the handle. I will not include all of these, as they are not important for the function of the screwdriver, however if you are more advanced, these are not difficult features to add.

- You may not copy this tutorial exactly and turn it in. If you do choose a screwdriver as your tool, it is expected that you will add more or different features or create the file differently. You could choose to make one part of the piece metal and make a separate part for the handle and mate them. You could also choose a totally different tool.
Starting a new Sketch

1. You will go to the sketch tab of your menu
2. Select Sketch
3. Now, you will choose which plane to sketch in. For this, it does not matter a ton, so I chose the right plane.
4. Now, select ‘Corner Rectangle’. The flathead I am using has a square shaft, and this will be the easiest way to make it.
5. Draw a rectangle! Don’t worry about size, that’s next.
6. A new menu will pop up. You can use this to edit, or do this later. Press the green check when you are satisfied. Here, I decided not to change anything right away.
7. Now, select ‘Smart Dimension’. This screwdriver has a width of 6.30 mm and it turns out it is a square.
Where you should be at

- Here is a picture of the dimensioned square I have so far.
- The screwdriver I chose has filleted edges, so that is next.

8. Select the fillet tool and then select two perpendicular sides.
9. The fillet autopopulates to 10mm, which would make our square obsolete. You must change this. You could easily do some math to figure out the fillet, but in this case I am going to use my caliper to estimate the radius of the fillets at 1.25 mm. You can now fillet all your corners. I felt that 1.25 was too round, so I cut my radius back to 1mm. I was left with this:
Extruding

10. Now, we are done with our first sketch. I am going to select ‘Exit Sketch’ to go back to the home screen.

11. Next, you will want to extrude this to be the entire length of the metal part of your screwdriver. If you wanted to make separate parts that mated, you would include the length that is encased in the plastic. I will not be doing this. I am just measuring from the tip of the flathead to the base of the handle. In my case, 146.91 mm

12. Click back to the ‘Features’ Tab and select ‘Extruded Boss/Base’

13. We are doing a Blind extrude for 146.91mm.

14. Now, you should have a square stick, like so:
Adding the tapering

There are many ways to draw something in Solidworks, here I will be demonstrating how you could cut away at the piece to make an edge. You could also make it shorter and then add another piece that is tapered.

15. I’m going to start by making another sketch, using one of the sides as a plane.
16. I’m going to use the tools on the top of the screen to make this plane flat in my view.
17. I’m going to start by making two points centered on the end of one of the long sides which will become my flathead point.
18. Now, I’m going to add points where the tapering ends on the side, like so:

![Diagram of tapering points](image1.png)

19. And then, I’m going to connect the points with lines of infinite length:

![Diagram of connecting lines](image2.png)
20. To finish, you will want to close the shape by creating a triangle with the 2 point line tool.

It should look something like this.
Cutting the Taper

21. Now, you can exit the sketch.
22. From the ‘Features’ tab, select ‘Extruded Cut’ use the shapes you made as the ‘Selected Contours’ to cut out your tapers. This might take some messing around, but eventually it should work. You’ll have something like this:
Adding the Handle

23. Now we’re halfway there! Next thing to do is add a handle. We’ll start by making a drawing of a circle on the end of the piece that is not tapered. My circle will have a diameter of 26 mm. Here is my centered circle.
Extruding your Handle

24. Next, you can exit your sketch and extrude your handle. I extruded mine about 30mm.
25. Now, we’re going to make an offset plane for the next part of the handle.
26. Use the ‘Offset Surface tool to make a plane 5-10 mm past the base of your handle.
27. Next, use the ‘Planar Surface’ tool to make the surface offset a workable plane.
28. You’ll have something like this:
29. Now, we are going to use some of the fancier tools to draw on this new plane.
30. Start a new sketch on the plane.
31. Start with a circle the width of the handle. In my case, the width of the handle is a bit bigger than the base of the handle, about 30mm.
My handle has a pattern, so I’m going to add that.
Here’s my dimensioned handle pattern.
32. I then used the Trim tool to get rid of the extra bits of circle before extruding.
Almost there!

Now I have a ghost screwdriver that’s missing a bit in the middle like so,

We are now going to use the ‘Extruded Boss/Base’ tool to connect these two. You’ll go from the larger plane to the smaller plane and use a loft angle so that they match up.
And you’re done!
Making a Rendering

One of the last things to do is make a rendering, or a realistic picture of your tool.

- At the top, go to Tools>Add-In
- Add in PhotoView360
- Now, PhotoView360 should be at the top. Open a preview window
- Use the edit appearances tool to change the color, lights, and background to something unique that makes your tool look amazing.
- Use the ‘Save Preview Image’ tool in the preview window to save as a JPEG.
Changing the color of a part

- Go to the ‘Render Tools’ tab and select ‘Edit Appearance’
- Use the ‘Selected Geometry’ tool to either select faces or parts. You’ll want to do faces if you made your tool in one drawing, or parts if you made separate parts and put them together.
- Select your faces/parts and choose a color. You can also play around with the texture options as well.
Taking a video of your tool

The last deliverable is a 3-5 second video showing the tool you’ve created.

- To do so, you will go to View>Screen Capture>Record Video
- The preset settings work fine, click ok and begin to manipulate the object
- When you have a few seconds of movement, click the ‘Stop Recording’ button on the upper left side or the View>Screen Capture menu.
- You will need to submit this and your rendering to a folder in the working directory.
CAD Glossary

- Features Tab Explained

**Extruded Boss/Base**- The most basic tool to turn a 2D sketch into a 3D object. Simply takes the shape and drags it along a specified direction to create the solid shape.

**Revolved Boss/Base**- Takes a sketch and rotates it around an axis to create a 3D object. Useful when you want to create hollow objects.

**Swept Boss/Base**- Takes a sketch and drags it along a 3D path to create a solid object. The easiest way to make a spring.

**Lofted Boss/Base**- Connects two different sketches to create an object that represents a smooth transition between the two.

**Boundary Boss/Base**- Extends a sketch in a direction much like the Extruded Boss/Base tool, but will cut the object if it crosses a specified path.
**Extruded Cut**- Uses a sketch to remove parts of an object in a straight line. The exact opposite of the Extruded Boss/Base tool.

**Hole Wizard**- Useful for creating holes that will be used in complex mates.

**Revolved Cut**- Takes a sketch and rotates it around an axis to remove parts of an object.

**Swept Cut**- Takes a sketch and drags it along a 3D path to remove parts of an object.

**Lofted Cut**- Connects two different sketches to remove parts of an object that represents a smooth transition between the two.

**Boundary Cut**- Removes geometry from a sketch in a direction much like the Extruded Cut tool, but will stop cutting the object if it crosses a specified path.

**Fillet**- Rounds a harsh corner.

**Linear Pattern**- Multiplies entities of a sketch along a specified path.

**Rib**- Connects two faces of a 3D object with a wall.
**Draft**- Gives vertical or horizontal walls a slight slope. Usually used to facilitate injection molding.

**Shell**- Creates a zero-thickness wall around a specified object.

**Wrap**- Matches a flat surface to a curved face. Usually used to give an object text.

**Intersect**- Extends or cuts the surface geometry of two faces to force them to intersect smoothly.

**Mirror**- Replicates a specified object as if it were placed in front of a mirror. **Reference Geometry**- Creates a “fake” sketch that can be used as a parameter in other tools.
• Smart Dimensions
The Smart Dimension tool is a way to define and display the relationships and measurements that make up a sketch.

Defining length
Simply click on the line you want to define, then use the popup to specify its length. You can drag the location of the displayed measurement, as well as change its value, with left click.

Defining an Angle
Select the first line that will define the angle with left click, then without clicking anywhere else, select the other line with left click. The popup for this action should then change from length to the angle between the two line. This display acts the same as the length display.
The University of Minnesota is an equal opportunity educator and employer.
Bakken MDC Internship Program

3D Printing Tutorial

Earl E. Bakken
MEDICAL DEVICES CENTER

UNIVERSITY OF MINNESOTA
Driven to Discover®
3D Printing Assignment

Goal:
Learn and practice the logistics related to using and operating an FDM 3D printer to prototype
designs from CAD. The use of two different 3D printers are used to demonstrate the differences in
orientation, print speed, support settings, and infill density effect on the aesthetic quality, structural
integrity, and resolution/accuracy of the part. As always, please read through the entire assignment
before beginning your work.

Requirements:
- Note that it is always required that you wait for your print to start and the first few layers so
  you can be sure it will work. Things can go wrong and damage the 3D printers.
- Print two things on 3D Printers housed in the Bakken MDC with different settings that you,
yourself have drawn in solidworks.
- Block off at least one hour of your schedule to either come to the Bakken MDC to slice and
  start your print. It is recommended that you wait for your print to start so you can be sure it
  will work.
- Use 2 different 3D printers in the Electrical lab, instructions are included for using Cura, a
  slicing software used to convert your CAD drawing into gcode, which is read by our
  Monoprice Mini and CR-10 printers.

Assignment:
- Print at least one copy of your tool from the CAD tutorial
- Your second print can be from the same file but sliced differently or a different part with
different slicing settings.
- Submit image of 3D printed parts to '3D Print Submission Folder'
- Submit a document outlining what print settings were used, what was different, and how
  those differences impacted the part to the same folder as well.

Resources:
- 3D Printing with Cura slides
- Posters in the Electrical lab

NOTE: It is YOUR responsibility to prevent damage to the 3d printers! This includes making sure
the machine settings are properly set up. If there are any problems that occur or the printer is
making an erroneous sound, please pause the print, and ask a lab technician or supervisor to help.
3D Printing Tutorial
Bakken MDC
Setting up a new printer in Cura

Slicing:
Cura version 3.6.0
Can be downloaded [here](#)

Click on:
1. Add New Machine
2. Other
3. The 3D printer you will be running the print on
   - Creality CR-10
   - Monoprice Select Mini V1
Please verify that the machine specifications are as follows

If using a CR-10:

4. Width and depth to 300 (mm)
5. Max height to 400 (mm)
6. Check the box for Heated Bed

If using a Monoprice mini:

4. Width and depth to 120 (mm)
5. Max height to 120 (mm)
6. Check the box for Heated Bed
Using Cura to slice prints

Now that the printer is set up, use tools to drop in your .STL files and position appropriately

Most of the default settings are OK. There are a few things you may want to change:

• Profile: this is pretty self explanatory, if you are making a final prototype, you might want to do a more fine profile. A more coarse profile will print faster
• Support: If your part has overhangs or delicate geometry, check this box.
Using Cura to slice prints

You can refer to the Glossary at the end of the powerpoint to better understand these settings.

1. Layer height can be between 0.08 and 0.20 mm for CR-10 and 0.06 to 0.15 mm for the Mini, a lower layer height will have higher resolution.

2. Shell thickness can be between 0.8 and 1.6 mm for CR-10 and between 0.8 and 1.2 mm for the Minis in intervals of the nozzle size:
   - 0.8 mm - 2 shells, 1.2 mm - 3 shells, 1.6 mm - 4 shells

3. Infill can be changed if you like (Higher fill, stronger part; Lower fill, faster print)
Using Cura to slice prints

1. Select “Custom” to change 3D print settings

2. Slicing variations can be modified here.
   Set Nozzle Size as 0.4 (mm)
   Set Filament diameter as 1.75 (mm)
Using Cura to slice prints

- Bed temperature should be 55 for PLA (230, 80 for ABS, 230, 50 for NinjaFlex-Ask before printing NinjaFlex on the CR-10)

- Add support if desired (if print has part that overhangs >45°)

- Speed should NEVER go above 75
Using Cura to slice prints

- ALWAYS add a raft as the Build Plate Adhesion
- Set filament size to 1.75mm
- Save gcode to an SD card (Toolpath to SD) and insert card in printer. Use card that is in the printer that you would like to use- cards are printer-specific
Support

You have 3 options for support

- None: this is ok if you have no angles greater than 45 degrees and you know that nothing on your part will need to be supported.
- Touching Buildplate: This is the most commonly used one, it creates supports between anything that is overhanging and the build tray.
- Everywhere: This is rarely used because it takes so long, but for parts with hollow areas it might be a good fit. Support will be created between every part of the piece.

Once you have selected your type of support, you can click on the ellipses box and change the fill amount settings, which will modify the print time and density of the support.
Variations for assignment

Settings you may want to change for the deliverable:

- Layer height, print one with a significantly small layer height and another much larger
- Fill density, try printing one around 5% and another around 30%
- Support, try printing one without support, one with touching buildplate, or one with everywhere
- Brim vs Raft, which one adheres better and comes off more nicely?

The only setting you should not change is printing temperature, because that will cause a lot of filament clogs.
Treating the Buildtak

Most of our 3D Printers have Buildtak
- You can tell by the black surface on the print beds with a gray ‘Buildtak’ logo on the bottom right

- In order to get the best bed adhesion possible, please follow the steps below before starting a print:
Treating the Buildtak

1. Clear all filament from the bed after a print. Use scrapers to do this.

   - The scrapers are located on the back counter with the CR-10 printers. If you cannot find them please ask a lab supervisor
Treating the Buildtak

2. Using medium grit sandpaper, lightly sand back and forth across the entire build surface.

3. Turn the sandpaper 90 degrees and lightly sand back and forth across the entire build plate again.
Treating the Buildtak

4. Use a dry paper towel to brush off the debris. Only brush towards you, not back and forth.

5. Repeat with a paper towel lightly soaked in isopropyl alcohol, once again only brushing toward yourself.
Treating the Buildtak

6. Now you’re ready to print!

7. When finished with your print, please prepare the bed for the next user.
Printing:
Using a CR-10 Printer
Printing on a CR-10

1. Push down on the selector
2. Scroll to Print from SD Card
3. Search for your file
4. Select your file
5. Make sure to treat the bed by sanding and wiping with a paper towel to ensure bed adhesion during printing (previous section)
Printing:

Using a Monoprice Mini
Print on a Monoprice Mini

1. Select ‘Print’ from the home menu

1. Search for your file

1. Select your file

4. Make sure that the extruder and bed are heating up to the correct temperatures using the wheel.

NOTE: Do NOT touch the extruder while it is heating or printing. It is extremely hot and can cause burns.
Printing on a Monoprice Mini

5. Once raft is printed, you can use tape to hold down the corners if it is a large part (only if necessary)

6. Make sure to treat the bed by sanding and wiping with a paper towel to ensure adequate bed adhesion during printing (treating Buildtak section)
Reminders

- If the printer starts making a noise, press cancel or pause your print. Ask for help.

- Always return the 3d printing tools to the correct place and to let a lab supervisor know if something is missing or depleted.

- Do not let the buildtak touch other printers or objects.

- Do not re-adjust the buildtak by pushing it.
Glossary

**Maximum width/depth/height** - the maximum printing parameters of the 3D printer (specific to each type)

**Heated bed** - the buildtak itself is heated to improve print quality by preventing the plastic from warping while it is cooling on the buildplate

**Layer height** - the thickness of each layer of plastic laid down on your part. The smaller the height, the better the resolution

**Shell thickness** - setting in slicing software is used to define how many side-by-side solid lines of plastic will be used to line the walls of your model. General rule of thumb: it should be a multiple of the nozzle thickness, (i.e: if nozzle is 0.4mm so the shell thickness can be 0.8mm or 1.2mm etc)

**Enable Retraction** - Relieves pressure from melt zone to ensure filament is not flowing through nozzle when it is not supposed to

**Fill Density** - the denser the print the longer it takes, but produces a stronger part. Lower fill means a faster print

**Print Speed** - the faster the speed, the lower the quality of the print

**Printing Temperature** - the temperature needed to melt the plastic filament (usually 220 deg C for PLA). Varies for types of filament used
Glossary

**Bed Temperature** - The temperature needed to prevent the plastic from warping on the build plate. (Use 55°C for PLA)

**Support Type** - Many different types of supports to stabilize any large overhangs in a print. [Here](#) is a great resource for more information on the types and when to use/not use supports.

**Platform adhesion** - Rafts are commonly used to secure the part to the build plate. This can be removed once the part has been printed

**Filament Diameter** - The size of the filament fed into the extruder in mm

**Flow** - The percent of power used to extrude filament.

**Machine Nozzle Size** - The extruder’s heated tip diameter in mm
Bakken MDC Internship Program

Machining Tutorial
Machining Tutorial: C-Clamp

Goal:
This project is meant to familiarize you with the mechanical lab's machine shop equipment and the basics of machining. Please read through the entire assignment before beginning.

**NO MACHINING ALONE NOR AFTER HOURS WHEN LAB SUPERVISORS ARE NOT IN THE BAKKEN MDC!**

Requirements:
- Fabricate a C-clamp similar to the one in the instructions
- Use at least two different materials
- At least once, change out the drill chuck or milling collet on the mill, and change the height. This may mean you take out the current setup, change the height (up and back down), and replace the setup.
- **Ask for help if you’re unsure about something. Yes, this is a requirement.**
- Always use common sense in the lab. Think before you turn on any equipment and assess whether there are any risks with your set up. Please use other interns as resources to learn how to use equipment, but NEVER GUESS HOW TO USE THE EQUIPMENT!
- **THERE IS NO SUCH THING AS A STUPID QUESTION... Unnecessarily damaging equipment or hurting yourself because you didn’t want to ask a question is stupid.**

Assignment:
- Plan out the design of your C-Clamp. Be creative (i.e. we don't want everyone's C-Clamp to look the same)! Build the clamp in CAD (multiple parts put together in an assembly). Keep in mind that you will have to physically machine the clamp. Submit a rendering of your C-Clamp assembly to the C-Clamp CAD Submission folder.
  - Name the file 'Lastname_Clamp_Rendering'
- Create a solidworks drawing file and add each of your parts that comprise the C-Clamp to the drawing file. Submit a screenshot of your drawing file to the C-Clamp CAD Submission folder.
  - Name the file 'Lastname_Clamp_Drawing'
- Follow through the C-Clamp Tutorial paying very close attention to detail and asking any questions that you have. Safety first! Submit a photograph of your finished C-Clamp to the C-Clamp Picture Submission folder.
  - Name the file 'Lastname_Clamp_Photograph'
Resources:
- A set of slides has been compiled which gives step-by-step instructions with pictures. It is in this document's parent folder. Read through all the slides once before starting to get an overall idea of the project.
- The online Bakken MDC Safety Training has basic information about the machining equipment in the mechanical prototyping lab. Refer to this with basic questions if you have them.
- If you have not done the above safety training, YOU MUST before you start this project.
- You are welcome to use any scrap material you can find in the Bakken MDC. The materials you will need for this assignment will be in the bottom drawer of the black cabinet beneath the table that holds the miter and chop saw in the mechanical prototyping lab.
- Before you start working on this assignment, please ask lab staff for a walk through of the machine shop equipment. If you’re not sure what equipment does what, make sure you ask a lab supervisor. Don’t guess or try to figure it out on your own!
- Work with others on this assignment if possible. Teamwork makes everything safer.
Safety Concerns

When working in the Mechanical Prototyping lab, you must:
- Wear safety glasses (provided, larger ones available to go over glasses)
- Wear long pants or a skirt
- Wear closed toe shoes
- Have long hair tied back
- Tuck in or remove scarves, necklaces, or dangly bracelets

You may also need (all of these available in the lab):
- Hearing protection
- Masks
- Gloves
# Materials and Machines for making a C-Clamp

<table>
<thead>
<tr>
<th>Machines and Tools</th>
<th>Materials</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Band Saw</td>
<td>- Aluminum Plate (C body)</td>
</tr>
<tr>
<td>- Mill</td>
<td>- Plastic or aluminum round stock (clamping cleat)</td>
</tr>
<tr>
<td>- Drill Press</td>
<td>- Steel or aluminum round stock (threaded rod)</td>
</tr>
<tr>
<td>- Lathe</td>
<td>- Wood or plastic round stock (handle)</td>
</tr>
<tr>
<td>- Grinder</td>
<td>- Steel or aluminum rod (rivet pin)</td>
</tr>
<tr>
<td>- Hacksaw</td>
<td></td>
</tr>
<tr>
<td>- Dremel</td>
<td></td>
</tr>
<tr>
<td>- Tap and Die</td>
<td></td>
</tr>
<tr>
<td>- Vise</td>
<td></td>
</tr>
<tr>
<td>- Anvil</td>
<td></td>
</tr>
</tbody>
</table>
Cutting the aluminum to size

1. Use the bandsaw to cut off the desired length of plate for the “C” shaped clamp body
2. You are more than welcome to alter the size of your C-clamp (larger, not smaller) and the design, within reason.
3. When using the bandsaw, use stick lubricant on the blade and a few drops of lubricant on the metal.
4. Adjust the carriage height to be 1/8” above the piece you are working on. Having the carriage higher unnecessarily can cause the blade to warp.
5. Use gloves and a push stick because the metal can get hot
Squaring the sides

6. Use the mill to square the plate. Use the parallel clamp to tighten down the work safely. You'll need to do this on each cut edge.

7. Grab an end mill of appropriate size and fix it into the mill with a collet that holds it firmly.

8. Adjust the speed and height as necessary. Be sure to re-tighten the bolts. Note that the mill in the MDC is not the most accurate. You may use the mills in Anderson Labs and follow their safety procedures.
9. Before turning on the mill, and being sure not to hit the mill end on your work, bring the mill end down to your desired height and lock it in place (lever on the left)

10. Turn on the mill and use the X,Y table handles to move the work back and forth until the surface has been leveled.

11. Turn the work over and do the other side. You will now have 4 square sides.
Finishing the shape

12. Use a file to deburr the work as it may have sharp edges now.

13. Use a marker and pen along with a caliper to mark the shape of your desired “C” body. Take into consideration that while a thinner “C” may allow you to clamp slightly larger things, a thicker “C” will not flex or bend as much under pressure. Here, a width of 0.4” is used.

- This will be cut on the band saw, but with some foresight you can see the difficulty of cutting the back line. A 90 degree cut is not possible, so a hole will be drilled so the saw blade can turn.
Drilling the holes

14. Measure the width of the bandsaw (0.2395”) and find a drill bit with a slightly larger diameter (0.2795” shown).
15. Determine which corner you will need to drill. Both have been drilled here, however you can get away with one if you want to save some work. The diagram at bottom right shows a one-hole strategy. Obviously, the back cut is not at 90 degrees, this will be taken care of on the mill.

16. Don’t drill on the corner, but inside the corner. You don’t want to create a stress concentration, which will increase your clamps flexing under loads, and it’s chance of breaking.

17. Be sure not to drill into the vice clamp. Putting a piece of wood underneath the drill into, helps.
Cutting the shape

18. Using a push stick and gloves (located on top of tool chest) cut along your lines on the bandsaw. Use gloves! Cutting metal makes it very hot.

19. Rinsing the work under cold water after cutting helps to cool it off quickly.
Squaring your cuts

20. Now you want to square up the cuts you just made. Use a piece of wood or a set of parallels to clamp your “C” flat in the mill table vice. Go through the mill setup from earlier.

21. This time, you’ll be using the side of the mill bit to make the inside of the “C” just as you want it. Here, a little indentation on one side of the end of the “C” for clamping circular things more easily was added.

22. Use a file to remove any burrs or sharp edges.
Creating the threaded hole

23. Decide on a size for your threaded rod. If you’re using a bolt or other already-threaded rod, then use thread gauges to determine its thread.

24. You need to find a tap that matches a die so that the threaded rod matches the threaded hole in your “C”.

25. When using a tap or die to cut threads, use cutting fluid as a lubricant and only cut a full turn at once before turning backwards to break any chips. This will save you from breaking a tap off in a hole.
Drilling a hole for the tap

26. From online or a chart in the lab, find the proper size hole to drill for the tap size you will use.

27. If you are threading your own rod, make sure it is the diameter for the die you want to use.

28. Use the tap wrenches in the kit to turn the tap. Remember, 1 full turn forwards, ¼ back every time.

29. To turn a die, use a wrench that fits. Follow the same pattern.
Progress

Your C-clamp should now look something like this. As you can see, in this example a bolt was used for the threaded rod. In the next slide, you can see that the head of the bolt was cut off with a hacksaw.
Cutting off the head of the bolt

Wood helps to hold the threads of the bolt without smashing them. There is a good chance the bolt will get hot, so use a glove to handle.

30. File down any sharp parts.
Making the handle

31. Cut your handle material down to a reasonable length, leaving an extra inch or two for the lathe to grip while you form the handle part. Use a hand saw or bandsaw to make this cut.
32. Clamp your handle material in the chuck of the lathe. For the circular handle material, use the 3-jaw chuck. For square handle material, you’ll need to use the 4-jaw chuck.
33. Be sure to remove the chuck key before starting the lathe.
34. For addition help, please reference the lathe tutorial video

NOTE: We upgraded the lathe so the pictures may not match but the operational steps are similar
Using the lathe

34. With the lathe turning counterclockwise (top towards you), use the X, Y handles to move the cutter along the work to shape your handle.

35. The cutter should be at or slightly below the centerline of your work.
   - For a smooth taper, angle the cutter block so it cuts diagonally as you move it along its tracks.
Fitting the threaded rod to your handle

36. Next, you’ll need to center drill the handle for it to fit onto your threaded rod. With the lathe off, move the cutter back and away from the work. Remove your work and measure how deep you need to drill for the threaded handle. Use the same drill bit that you did for the hole in your “C”, as this hole will also be tapped.

37. Re-clamp your work into the chuck backwards. Use a center drill to mark it’s true center, then drill using your bit.

- If you’re reading ahead, you can do this step before shaping the handle, just be sure not to remove too much end material and expose the end of your center hole.
38. Switch to a cut-off tool and cut off the extra material from your handle.
- Here, the handle was flipped back around in the chuck before parting off. This is not necessary.
39. Clamp your newly formed handle in the bench vise and tap the hole. Cutting lubricant is not needed for plastic or wood.
Your handle should now fit onto your threaded rod.
Preparing the threaded rod for the clamp cleat

40. Use the bench grinder to carefully create parallel flat surfaces on one end of your threaded rod.
41. Then, select a small drill bit that is slightly larger than your rivet pin. Clamp your rod under the drill press and drill a through hole.
42. Small drill bits are easy to break. The more flat your drilling surface, the better off you’ll be. Use a nail or punch to create a small divot where you want to drill. Slowly apply pressure while watching the drill bit for wander. Make adjustments as needed.
Starting the cleat

43. First you’ll want to turn your stock down to the size you want (as shown below on the lathe).
44. Next, use either the milling machine or hand Dremel to create the flat sides and middle slot that will be needed to rivet the cleat to the threaded rod.
- Here, the milling machine was used for the sides and the hand Dremel with a cut-off grinding wheel was used to create the center slot.
Finishing the cleat

45. When you have finished the sides and slot, put your cleat back in the lathe and use a cut-off tool to cut the cleat off from the extra stock. Same as when the handle was cut off.

46. Do any finishing touches with a file or hand Dremel. You want the flat part of your threaded rod to fit in the center slot with a little wiggle room.

47. When it’s as good as you’d like it to be, clamp it under the drill press and drill straight through both sides. Be sure to use the same drill bit as you did for the threaded bar’s hole. Also make sure the position of this hole will line up with the hole on the threaded bar for the rivet to pass through.
Attaching the cleat

48. For the rivet pin, you’ll need a small length of scrap metal rod as shown. In this example, solid metal is used, however you may try using hollow metal tube. You’ll need about 1-1.5mm of length beyond the sides of the cleat.
49. When creating a mushroom head on the first end of your pin, the pin doesn’t need to be in place in the cleat, in fact, for the sake of the cleat, it’s better if it is not.

50. Hold the rivet pin normal to the anvil with pliers when forming the first head, and then move it into place through the cleat and hole in the threaded rod and secure the other side with the first side held flush to the anvil.

51. View the short video “Rivet Nail Heads” for help. You may use a nail or screw with threads removed as your rivet in order to have the first side already made.

- Remember to have the threaded rod already threaded through the “C” piece before you form the second head.
You’re finished!

Congratulations! You now have a working C-clamp that you made!
Bakken MDC Internship Program

Machining Activity: Screwdriver
Machining Activity: Screwdriver Assignment

Goal:
Machining is a very useful and important skill in engineering as well as life. This tutorial is meant to help you become more comfortable with machining after completing your C-Clamp and learning about some of the basics.

Requirements:
- This tutorial requires the use of these tools:
  - Bandsaw
  - Mill
  - Drill Press
  - Lathe
  - Grinder
  - Hacksaw
  - Dremel
  - Tap and Die
  - Vise
  - Anvil
  - Hammer
- Be sure you are familiar with the locations and use of all of these tools. Ask for help if you are unsure.
- You will need the following materials for this project:
  - Aluminum round stock
  - Steel round stock
  - PTFE round stock
  - Plastic or wood round stock
  - Screw
  - Nail

Assignment:
- If you have completed a C-Clamp in previous years, you MUST complete this assignment.
- If this is your first year as an intern, this is not required, but encouraged.
- Make a drawing with solidworks of the tool you plan on creating.
- Bring your drawing to the shop and make yourself a screwdriver to bring to the weekly meeting as a deliverable.

Resources:
- Ask lab staff for help with tools! Don’t use anything you aren’t comfortable with or familiar with before asking for assistance.
- The lab safety training has many videos and tips about how to use this equipment.
Machining Tutorial
Bakken MDC
Safety Concerns

When working in the Mechanical Prototyping lab, you must:

- Wear safety glasses (provided, larger ones available to go over glasses)
- Wear long pants or a skirt
- Wear closed toe shoes
- Have long hair tied back
- Tuck in or remove scarves, necklaces, or dangly bracelets

You may also need (all of these available in the lab):

- Hearing protection
- Masks
- Gloves
Materials and Machines for making a Screwdriver

**Machines and Tools**
- Band Saw
- Mill
- Drill Press
- Lathe
- Grinder
- Hacksaw
- Dremel
- Tap and Die
- Vise
- Anvil
- Hammer

**Materials**
- Aluminum round stock (handle)
- Steel round stock (shank)
- PTFE round stock (washer)
- Plastic or wood round stock (handle)
- Screw
- Nail
Make a Solidworks Model

- Find a screwdriver with a design that you like and take a picture of it with a ruler.
- Create a new part file in Solidworks. Create a sketch on one of the planes. While in the “Edit Sketch” mode, go to Tools -> Sketch Tools -> Sketch Picture... and insert the picture you took.
- Use the ruler to scale the sketch to the correct size. Place the front of the screwdriver handle on the origin and along an axis.
- Create a sketch of the profile of the screwdriver.
- Print out the profile of the screwdriver to scale to reference while you build it.
Making the Shank

1. Cut down the steel shank using a vise and a hacksaw. Grind down the end to a flathead.
Make the Aluminum Handle

2. Chuck up the aluminum rod into the lathe. Drill a hole on one end using a drill bit of the correct cutting tap size (see references) for the diameter of the Steel Pan Head Phillips Screw. You can figure out the tap size and pitch of the screw with the nut and bolt thread checker.
3. Set the drill point against the stock. The tailstock quill has a dial on it to indicate its travel distance. Drill about ¼” past the length of the screw to ensure you have enough room to place the screw.
Make the Aluminum Handle

4. Remove material off this end of the aluminum handle to fit into the wood handle later.
Make the Aluminum Handle

5. Use a bandsaw to cut the aluminum handle to its final length.

- When using the bandsaw, use stick lubricant on the blade and liquid lubricant on the metal.
- Adjust the carriage height to be 1/8” above the piece you are working on.
- Use gloves because the metal can get hot.
Make the Aluminum Handle

6. Finish off this end of the handle by tapping the hole with the correct thread size for the screw. Put cutting fluid on the tap.
   - To keep alignment, maintain pressure on the tap handle by holding the feed lever down. Spin the tap handle a full turn clockwise to create a thread, and then counterclockwise halfway to clear out chips.
Make the Aluminum Handle

7. Put the other end of the aluminum handle in the lathe and make the contour of the handle from your drawing. Adjust the angle of the tool post to create the contour.

8. Then, drill a centered hole in this end of the aluminum handle that will fit the diameter of the shank.
Attach the Shank to the Handle

9. Press the shank into the handle using the vise. Use the small drill press to drill a hole into the handle, through the shank, and out the other side. The hole should be a little smaller than the diameter of the nail.

- Hammer a nail through to connect the shank and handle.
- Use a dremel to cut off the nail and grind down the nail to make a flush surface.
Make the Wooden Handle

10. Use the bandsaw to cut a piece of wood to the write length. Drill a hole in the center of the wood that will fit the aluminum handle and give it room to turn.

11. Use the end mill to cut a flat side along the length of the wooden handle on both sides.
12. Take sandpaper and make a nice finish.
Make the Washer

13. Take a bar of polyethylene. On the lathe, drill the center of the washer to the correct diameter to fit over the aluminum rod. Part off a thin washer using the parting tool.
Assemble

14. Put it all together! Wow! You made it!
Tap/Drill Chart

Source:
http://www.biolettersample.info/tap-and-die-chart/
Matlab Tutorial

Cases of Influenza Over Time

Earl E. Bakken
MEDICAL DEVICES CENTER
University of Minnesota
Driven to Discover®
Intro to Matlab Assignment

Goal: Matlab is a very useful and common software in the engineering world. This is a very surface level introduction to demonstrate some of the features Matlab offers. By the end of this, you should be able to create programs and make graphs in Matlab.

Requirements:
- Matlab works through a command window, so this tutorial is written to be used step by step.
- Reading through the entire assignment before beginning will be helpful.
- Start by downloading the software on your personal computer through or planning to come to campus. The Bakken MDC computers have Matlab, and Anderson labs, Mechanical Engineering 308, has computers available as well.
- If you are having problems accessing Matlab, please see lab staff.
- At the end of the tutorial, you will assemble your new knowledge to put together a program that makes a graph.

Assignment:
- After downloading or accessing Matlab, have two windows open on your screen, one for Matlab and another for the Powerpoint.
- Follow along with the powerpoint in Matlab.
- Once you have finished the powerpoint, create your own script as described on the last few slides and take screenshots of your graph and add them to a document with your script. Include the inputs so that someone could copy your script, feed the inputs, and output the same graph as submitted. Upload the document to the folder in Drive.

Resources:
- Matlab is free for CSE and CFANS students at the following links:
  - https://wwws.cs.umn.edu/download_software/matlab
  - https://oit.cfans.umn.edu/student-resources/software
- For more information about Matlab, the Math Works index is very helpful: https://www.mathworks.com/help/matlab/index.html
Matlab Intro Tutorial
Bakken MDC
Getting Started

- Once you have activated your license, you can open the program. The first time you open it, it should look like this:

- The left window shows your directory, so the file you are working in and any files that might be in it.
- The middle window is where you will be executing commands.
- The right side ‘workspace’ will show any variables you have used and what they are currently assigned to.
1. Start by entering “x=3” in the command window and hit enter. After each of these commands, you will need to press enter when you are ready for output.

2. Next, “y=4;”
   - You will notice that ; suppresses the output. You’ll also notice that in your right side ‘workspace’ window, two variables have now popped up. Your screen should look like this now:

```
>> x=3
x =
3
>> y=4;
y =
4
```

3. Next, enter “z=y*x”
   - This will output z as a new variable assigned to the value 12.
Writing Scripts

4. Now, we will explore how to create programs, called scripts within Matlab.
5. Click on the ‘New’ button on the upper left hand corner.

6. Next, type the following program

   ```matlab
   function c = calc(a, b)
   c = a * b;
   ```

   - The first line asserts that there is a function which will equal C. When you call function C, you will need to input two variables.
   - The second line shows how the function should execute.

7. Now you need to save your function. Go to the drop down arrow below ‘Save’ and select ‘Save As’. Save your script as something other than function or untitled.
   - It will pop up in the left window, showing that it is saved in that folder.
Calling Scripts

8. Now you can call your function.
   - You will use whatever you saved the function as to call it. In this example, the
     function was saved as calc.m
   - In the command window, enter “calc(3,4)”
   - 12 should pop up and then on the right side, you should have a new variable called
     ‘ans’

9. You can also assign a variable to calc in the command window, so that the answer
    will be the new variable, for example, “AB=calc(1,2)” this would give you a
    variable AB with the value of 2 in your workspace.

10. When your workspace starts to fill up, or you want to reuse variables, you can
     right click and clear your workspace.
For Loops

- Another important function in Matlab is the for loop.
- A for loop will execute a function ‘for’ a set number of times.

11. Open a new script and enter the following:

```matlab
for a=(1:.5:5)
    b=(a+1);
    disp(b)
end
```

- This will output all of the values from 1 to 5 with steps of 0.5 and add 1 to each of them before outputting.

```plaintext
>> forlooppractice
    2
    2.5000
    3
    3.5000
    4
    4.5000
    5
    5.5000
    6
```
Graphing

- Graphing in Matlab is one of its most useful functions. Let’s start by graphing the cosine of x.

12. Enter “x=linspace(0,2*pi,100);”
13. Now, enter “y=cos(x);”

- This makes a matrix 1 row by 100 columns of evenly spaced numbers between 0 and 2 pi. More numbers will make your graph more smooth, fewer numbers will make it choppy.

- This creates another matrix of answers. Whatever type of variable you input, (ex, number or matrix) your answer will come out in that format. In this example, we inputted a matrix so we got a matrix out. In the earlier example, we inputted integers, so we got integers out.
14. Next, type “figure”
15. And then, “plot(x,y)”
Your graph should look like this:
16. Now, we will add labels.
17. Enter “title(‘Graph of Cosine’)”
18. Next, enter “xlabel(‘X’)” or whatever you would like to label it.
19. Then enter “ylabel(‘Y’)”
- Now, your graph should look like this:
Assignment

- With those basics down, your assignment is to create a script that inputs at least 2 variables and outputs a circular graph with creative labels. Here is an example:
Assignment

The dimensions of the circle may be whatever you desire. Please be sure to add a descriptive title to the graph, appropriate axis labels and a legend.
3D Scanning Tutorial

Goal:
Learn and practice the use of a 3D scanner and understand its best applications.

Requirements:
- Take a scan of an organically shaped object, something you would not want to draw in Solidworks. Body parts work very well for this.
- Take an hour to come to the Bakken MDC to use the scanner. You will want to make sure there is not an event in the Virtual Prototyping Lab, you may not use it when there is an event in there.
- You may want to partner up for this activity, you would still be expected to come up with your own part, however it is much faster to get the hang of it if you watch someone else try.
- If you are having trouble or have questions, please ask a lab supervisor for help, do not try to troubleshoot it yourself.
- Turn off and unplug the scanner once you are done. Be sure to keep the cords on the desk. They are notorious for getting caught on the arms of chairs.
- Save your completed STLs to a flash drive and edit them on another computer.
- Label your completed STLs with your first and last name and then add them to the 3D scanning project folder.
- Store the scanner away from the edge of the counter!

Assignment:
3D scan an object and create a:
- complete mesh from 3D scanning a part
- fixed mesh using an STL editing software like Meshmixer

Resources:
- Please reference the 3D scanner Google Calendar to see when it is available. To use the scanner, schedule a time on the calendar. Label the event with your full name and UMN x500.
- Reach out to lab staff for the computer password in the VR room and for any help you need with the scanner.
- For Meshmixer and mesh editing help, the Autodesk Meshmixer website will be helpful: http://www.meshmixer.com/
3D Scanning General Information

Uses of 3D scanning
- In many cases, you may find that drawing a part in CAD may be faster and more accurate than taking scans of it.

- The best applications of 3D Scanning are very unique parts with geometries that are difficult to measure, such as anatomy.

- Consider the opacity and shine of the material you are scanning. You will find that some materials will not produce scans because the light passes through it.
3D Scanning General Information

Fixing

- The initial scans you will make with the 3D Scanner will not look very nice. In order to get a complete mesh, you will need to make many scans and lay them together in Artec Studio.

- Artec Studio offers many fixing tools, however you may consider a fixing program like Meshmixer, which is a free STL editing tool from Autodesk.

- Your scans are not going to be perfect, so don’t be discouraged if they don’t look like exactly like the print. The main goal of this is to get you familiar with the technology. If you enjoy this and would like to get better at making scans, by all means please do so.
3D Scanning General Information

**Scanner Safety**
- The MDC has this scanner available for use because we believe that it is a valuable tool for medical device development. Please be considerate while using the scanner and make sure that it is not in danger at any time, making sure that it is not on the edge of a table, has plenty of cord slack, and unplugging it (by unscrewing the power cable) after each use.
Project 1: Scanning

In the VR Room, there are three 3D prints. This tutorial will walk you through how to scan and edit the first one, the bacteriophage, and then from there you will scan and fix the next two, the spinal section and the heart.
Project 1: Scanning

Starting
- You will place the center of the object on the center of the rotating tray.
- Next, open Artec Studio
- Select ‘Scan’ at the upper left hand corner
- Select ‘Preview’ and move the scanner around your object to get a feel for how far away you should be to properly scan the object.
  ■ Move the scanner back and forth and then up and down to get a full view of the object. The scanner will need to be a certain distance from the object. Don't worry about being shaky, the scanner accounts for human error.
Project 1: Scanning

- Next, select ‘Record’ and then ‘Stop’ once you have made a full pass of the object.
  - Use the tray to move the object around to get a full view.

- Continue taking scans (they will pop up in a list on the right hand side) until you have a good 8 or so.

- Be patient with the process and make sure the scanner is up to temperature.
Fixing in Artec Studio

- To compile all of your scans, make sure they are all selected on the list on the right.

- You will then go to ‘Tools’ and start at the top with Rough Serial Registration and then continue down the list smoothing out your files.

- When you are finished, you’ll go to file and then export your scans as an .STL file to a flash drive. Then, you will take your files to Meshmixer and work on fixing them as best you can there.
Project 2: STL Editing

- For this deliverable, you will want to download Meshmixer to your computer. It is a free software from Autodesk and should not take long to download. It can be found here: http://www.meshmixer.com/
  - Meshmixer is an intuitive software, use its tools to smooth out rough edges and fix open edges. You can patch holes or cut away extra in order to make your STL look as similar as possible to the starting print.
  - If you get stuck, try this website for help: http://www.mmmanual.com/

Completing the project:
- Label your completed STLS with your first and last name and then add them to the 3D scanning project folder.
Microcontroller Tutorial

Bakken MDC Intern Program
Preview

1. Microcontrollers
   a. What are Microcontrollers?
   b. Why Learn about them?
   c. Examples of Microcontrollers

2. Arduino
   a. What is an Arduino
   b. Code and Structure
   c. Arduino Parts
   d. Types of Arduinos

3. Mini-Project
   a. Project: Wheel-of-Life
   b. Why this project?
   c. Project Plan
   d. Preparation
Microcontrollers
What are Microcontrollers?

~ a small computer on a set of circuits

- They are used in consumer products like phones and automobiles.
- They build an interaction between users and devices.
- A.k.a “special purpose computers” because they fulfill one purpose well
Why Learn about them?

....because they are used in almost every device. Below are some examples.

Automobiles
I.e. Anti-Lock Brakes

Medical Devices
I.e. Pacemakers

Household Appliances
I.e. Refrigerators
Examples of Microcontrollers

- Raspberry Pi
- Arduino

(Which we’ll use in this tutorial)
Arduinos

*Addit. credit to: Dr. Salitermann
What is an Arduino?

~ a type of microcontroller that interacts with objects in the physical world

- Its programming language is in C/C++.
- Cross-platform: can work in Windows, Mac OS X, and Linux.
- Warning: the following slides may have a lot of detail. No worries, because we will guide you. Stay with me.
Once you download the Arduino IDE software and open the application, you will see a window similar to this.

This is a “sketch.” You will write code in the window to tell what you want your Arduino to do.
These are functions.

In programming language, a *function* is a set of procedures that could return values or make the object perform a certain task.
void setup()
~ sets up what Arduino pins to use and values it will INPUT/OUTPUT

void loop()
~ tells what actions the Arduino will do based on what was setup
Verify
~ checks code for errors when compiling it

Upload
~ compiles and uploads code/sketch to Arduino
Code and Structure (cont.)

New
~ creates a new sketch

Open
~ opens a new sketch

Save
~ saves the sketch
Serial monitor
~ displays data sent from Arduino

Message Board
~ displays if code was successfully uploaded, or if there were errors when compiled
Woah...
Looks complicated?
Let's break it down!
**Power Jack**
~ To be connected to an AC-DC adapter, or to power the Arduino.

**USB Connection**
~ Commonly used to upload and run code.
Arduino Parts (cont.)

- **Ground Digital Pin**
  - Sets reference voltage point to zero

- **Digital Pins**
  - Used to INPUT/OUTPUT discrete voltage values (0 or 1V) set by user
Analog Pins
~ Used to INPUT/OUTPUT continuous voltage values set by user (safe if less than 5V)

Ground & Vin Pin
~ Connects board to external power source
Arduino Parts (cont.)

Test LED
~ Can be programmed to flash lights

Power LED
~ Indicates whether Arduino was ON or OFF
Reset Button
~ Restarts the code uploaded on the Arduino

Micro-Controller Chip
~ the “data storage”; holds the code uploaded to the Arduino
Types of Arduinos

Arduino GEMMA
Arduino NANO
Arduino UNO

(which we’ll use in this tutorial)
Mini-Project
Project: Wheel-of-Life

A.k.a the Zoetrope => how it works
Why this project?

This project goes over feedback. Something common in both organisms and devices.

It may require your previous knowledge in tutorials such as laser-cutting, CAD, 3D printing, and machining.

Would include basic concepts about microcontrollers.
Project Plan

**Tutorial 1**
**INTRO**
(1-2 Days)

**Tutorial 2 + 3**
**CONCEPTS**
(2-4 Days)

**Tutorial 4**
**MINI-PROJECT**
(4-7 Days)

*WARNING*: The mini-project would NOT be a step-by-step tutorial. You would need some of the tutorials to solve it.

**If you feel like you have enough experience with Arduino (or want to skip intros), feel free to skip to Tutorials 3 and 4.**
1. **Download** the Arduino IDE
   a. Choose the one corresponding to your computer
   b. Or you could use the MDC’s.

2. **Review the parts list** from your starter kit to make sure you have everything.
   a. Report to the MDC supervisor if anything is missing.

3. Do this little-by-little.
   a. Because it will be easier than doing this in one sitting.
You must be thinking...
Wow, this is too much to handle.

It’s ok to feel that.

But, we will take you step-by-step. And it would be much easier than you think.
Credits:

Dr. Salitermann
~ for generously providing his notes about Arduinos.
~ please check out his website for more information about Arduinos: [http://www.tc.umn.edu/~drsteve/](http://www.tc.umn.edu/~drsteve/)

Joshua Cadavez
~ author of notes
Microcontroller Tutorial
Part 2
The Medical Device Center
University of Minnesota-Twin Cities
Preview

1. Quick Electronic Lessons
   a. Ohm’s Law
   b. Circuits

2. Starter Kit

3. Arduino Development
   a. Coding Language
   b. Circuit Schematics
   c. Algorithms
Warning

There are a lot of content in these slides. BUT, you don’t have memorize too much of the details.

Just use this as a reference when going through the project.
Ohm’s Law is the relationship between current (I), voltage (V), and resistance (R) within a circuit.

This is its formula: \( V = I \times R \)

**Voltage** = electrical pressure. Its units are volts (V).

**Current** = flow of electrical charge. Its units are in amps (A).

**Resistance** = opposition to the flow of electric current. Its units are in Ohms (Ω)
In this sense, a circuit is an electrical path between two or more points. To work, it needs a complete path from the source (power) to the point of least energy (ground).
A series circuit is a circuit that has components that come one after another.
Parallel Circuit

A parallel circuit is a circuit that has components that are side-by-side.
Arduino Starter Kit

Each kit has a parts list book.

It could also be found online.

If any part is missing, report it to the MDC Supervisor.
Breadboard

~ This board allows you to build electronic circuits.

This vertical strip is used for power and ground connections.

This middle row breaks the connection between both sides of the board.

Each row in the bus running horizontally is a conductive metal strip.
Resistors

This component resists the flow of electric energy and converts it into heat.

Resistors have a **color code** that notes its strength in reducing current flow.

### Resistor Color Code

<table>
<thead>
<tr>
<th>Color</th>
<th>1st Band</th>
<th>2nd Band</th>
<th>3rd Band</th>
<th>Multiplier</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x 1 Ω</td>
<td>+/- 1%</td>
</tr>
<tr>
<td>Brown</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>x 10 Ω</td>
<td>+/- 1%</td>
</tr>
<tr>
<td>Red</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>x 100 Ω</td>
<td>+/- 2%</td>
</tr>
<tr>
<td>Orange</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>x 1K Ω</td>
<td>+/- 5%</td>
</tr>
<tr>
<td>Yellow</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>x 10K Ω</td>
<td>+/- 1%</td>
</tr>
<tr>
<td>Green</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>x 100K Ω</td>
<td>+/- 1%</td>
</tr>
<tr>
<td>Blue</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>x 1M Ω</td>
<td>+/- 25%</td>
</tr>
<tr>
<td>Violet</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>x 10M Ω</td>
<td>+/- 1%</td>
</tr>
<tr>
<td>Grey</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>+/- .05%</td>
<td></td>
</tr>
<tr>
<td>White</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>+/- .05%</td>
<td></td>
</tr>
<tr>
<td>Gold</td>
<td>x .1 Ω</td>
<td></td>
<td></td>
<td>+/- 5%</td>
<td></td>
</tr>
<tr>
<td>Silver</td>
<td>x .01 Ω</td>
<td></td>
<td></td>
<td>+/- 10%</td>
<td></td>
</tr>
</tbody>
</table>
A.k.a “The Pot” and this component is a variable resistor with three pins.

Two pins are fixed to one end of a fixed resistor.

The middle pin moves across the resistor to divide it into two halves. This pin determines the difference in voltage.
Light Emitting Diode (LED)

This component converts electrical energy into light energy.

This is a polarized component, it allows electric flow in one direction.

The longer lead is the cathode (-) and the shorter lead is the anode (+).

The cathode is connected to ground while the anode is connected to a voltage source.
Switch

This electrical component interrupts the flow of electricity. When open, it breaks the circuit.

The ones we’ll be using in the tutorial are pushbuttons because they only close the circuit when pressure is applied.
Liquid Crystal Display (LCD)

This electrical component is an alphanumeric or graphic display.

Its liquid crystals come in different sizes, shapes, and styles.

Ours will have two rows with 16 characters each.
This electrical component converts electrical energy into mechanical energy to perform a certain function.

The coils in the wires inside the motor become magnetized when current flows through it.
Other Components

The USB cable connects the Arduino UNO to your personal computer. It also powers most of the projects.

These are header pins that helped connect components to the breadboard.

Used to connect a 9V battery to the power leads that can be plugged into the Arduino.
Arduino Development
Coding Language

Arduinos have their own set of language.

The following slides will go over some basic terms.
These are control statements:

- If, if-else, if-else-if
- For
- While
Coding Language (cont.)

These are control statements:

- **If, if-else, if-else-if**
  - It tells your program to execute a section “if” the condition is TRUE. “else” it would execute something else.
These are control statements:

- **For**
  - These statements specify the number of iterations to run a certain segment of code.

```plaintext
for i = first to last do statement
(* or just *)
for i = first..last do statement
```
These are control statements:

- **While**
  - It executes a certain code segment “while” it is true.

```java
while (true) {
    // your code goes here
}
```
Coding Language (cont.)

Arduino’s have their own punctuation notation.

Syntax

; Used to end a statement
{} Enclose statements, keep balanced
// Start comment until end of line
/* ... */ Multi-line comment
#define Assigning a value to a constant name
Follows C rules and no semicolon afterwards
Use const type variable = value (e.g. const float pi = 3.14) when able instead.
#include To include outside libraries
Coding Language (cont.)

These are operators.

The left column are for arithmetic operations whether you add, subtract, etc.

The right column are for boolean operations a.k.a determines TRUE or FALSE.

= assignment operator
+ addition
- subtraction
* multiplication
/ division
% modulo

&& and
|| or
! not

Prof. Steven S. Saliterman
These are more operators.

The left column are for making comparisons between values.

The right column are pointers. In other words, they point to where a value is stored.

**Comparison & Pointer Operators**

- `==` equal to
- `!=` not equal to
- `<` less than
- `>` greater than
- `<=` less than or equal to
- `>=` greater than or equal to

* dereference & reference
Coding Language (cont.)

These are more operators.

The left column are for making comparisons based on their bits.

The right column are operators that change the variable based on itself.

<table>
<thead>
<tr>
<th>Bitwise &amp; Compound Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>^</td>
</tr>
<tr>
<td>~</td>
</tr>
<tr>
<td>&lt;&lt;</td>
</tr>
<tr>
<td>&gt;&gt;</td>
</tr>
</tbody>
</table>

| | increment |
| | decrement |
| += | compound addition |
| -= | compound subtraction |
| *= | compound multiplication |
| /= | compound division |
| %= | compound modulo |
| &= | compound bitwise and |
| |= | compound bitwise or |

Prof. Steven S. Saliterman
Coding Language (cont.)

These are the types of data. They help classify the values a variable would store.

A general rule is that data could operate with other data that are the same type.

1. **boolean (8 bit)** - simple logical true/false (1 byte = 8 bits)
2. **byte (8 bit)** - unsigned number from 0-255
3. **char (8 bit)** - signed number from -128 to 127. The compiler will attempt to interpret this data type as a character in some circumstances, which may yield unexpected results
4. **unsigned char (8 bit)** - same as 'byte'; if this is what you're after, you should use 'byte' instead, for reasons of clarity
5. **word (16 bit)** - unsigned number from 0-65535 (1 word = 2 bytes)
6. **unsigned int (16 bit)** - the same as 'word'. Use 'word' instead for clarity and brevity.

Source: https://learn.sparkfun.com/tutorials/data-types-in-arduino

---

*Prof. Steven S. Salitserman*
Some more types.

The most common ones we will use are boolean and int.

7. int (16 bit) - signed number from -32768 to 32767. This is most commonly what you see used for general purpose variables in Arduino example code provided with the IDE.
8. unsigned long (32 bit) - unsigned number from 0-4,294,967,295. The most common usage of this is to store the result of the `millis()` function, which returns the number of milliseconds the current code has been running.
9. long (32 bit) - signed number from -2,147,483,648 to 2,147,483,647
10. float (32 bit) or double- signed number from -3.4028235E38 to 3.4028235E38. Floating point on the Arduino is not native; the compiler has to jump through hoops to make it work. If you can avoid it, you should.
To see examples, you may open them in your Arduino IDE as shown on the left.

I’ll go over one in the next example. But first...
Relax... take a deep breath...
Circuit Schematics

This is a circuit schematic. In other words, it is pictorial diagram of the components used in the circuit.

The components have standard symbolic representations. You may find an expansion of the list on p10 of your Arduino book or here.
Circuit Schematics (cont.)

This symbol represents the Arduino pin being used.

“A” and “D” represents analog and digital respectively.

The number represents the numbered pin being used.
This symbol represents the LED.

It is labelled what color it would display.

In this diagram, it’s Red with a wavelength of 633nm.
Circuit Schematics (cont.)

This symbol represents the resistor.

It is labelled with its value in Ohms (Ω).

For every additional resistor added, its labelled by R1, R2, R3, etc. for easy reference.
This symbol represents ground.

Ground is used as a reference point to measure the other voltage levels (V) in the circuit with Ground being 0V.
Algorithms & Key Terms

~ Before coding, programmers sketch out algorithms.

- Algorithms are a sequence of actions that may perform calculations, data processing, or automated reasoning.

- The diagram on the left is an example. It is an algorithm for code development.
Remember this circuit? This is a series circuit meaning that the current passing through the circuit is the same. To get it to work, it needs to be programmed.

Let’s set up an algorithm.

We want this circuit’s LED to blink ON and OFF.
This is how I would write it. You may have thought of something similar.

However, this algorithm needs to be translated into the Arduino's language.

On the following slides, I'll lead you on how to do so.
Initialize the variables:

Give an initial value to elements, or factors that would affect the circuit’s performance.

Here, we are using a Pin and giving it 9 because we will use the Arduino’s 9th pin to output energy.
Void setup():

Here we direct the Arduino’s digital pin (the 9th) to OUTPUT energy, or act as the power source.

When do you use OUTPUT or INPUT?
**HIGH** – Digital pins are set either **HIGH** or **LOW**

- When a pin is configured as an **INPUT** with `pinMode()`, and read with `digitalRead()`, the Arduino (Atmega) will report **HIGH** if:
  - a voltage greater than 3 volts is present at the pin (5V boards);
  - a voltage greater than 2 volts is present at the pin (3.3V boards);
- A pin may also be configured as an **INPUT** with `pinMode()`, and subsequently made **HIGH** with `digitalWrite()`.
  - This will enable the internal 20K pullup resistors, which will pull up the input pin to a **HIGH** reading unless it is pulled **LOW** by external circuitry.
Algorithms & Key Terms (cont.)

- When a pin is configured to `OUTPUT` with `pinMode()`, and set to `HIGH` with `digitalWrite()`, the pin is at:
  - 5 volts (5V boards);
  - 3.3 volts (3.3V boards);

- In this state it can `source` current, e.g. light an LED that is connected through a series resistor to ground.
Void loop():

HIGH and LOW determines the amount of volts the pin will output.

Here, **HIGH means 5V** and **LOW means 0V**.

So, when you upload this to your Arduino, the LED should blink.
This was only for LEDs... Let’s Practice with others!
Credits:

Dr. Salitermann

~ for generously providing his notes about Arduinos.

~ please check out his website for more information about Arduinos: http://www.tc.umn.edu/~drsteve/

Joshua Cadavez

~ author of notes
Preview

1. Practice
   a. LCDs
   b. DC Motors
   c. Sensors

*These tutorials are based off Adafruit’s.

This is not the deliverable but only for practice.
Practice
LCD
The LCD is a flat display that shows images depending on the information it receives.
The goal is to have the LCD print “I’ll keep learning” and also the number of seconds passing since the Arduino was reset.
You’ll need these following parts.

<table>
<thead>
<tr>
<th>Part</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCD Display (16x2 characters)</td>
<td>1</td>
</tr>
<tr>
<td>10 kΩ variable resistor (pot)</td>
<td>1</td>
</tr>
<tr>
<td>Half-size Breadboard</td>
<td>1</td>
</tr>
<tr>
<td>Arduino Uno R3</td>
<td>1</td>
</tr>
<tr>
<td>Jumper wire pack</td>
<td>1</td>
</tr>
</tbody>
</table>
This is the physical setup of the Arduino UNO. The potentiometer controls the LCD’s text resolution.
This code downloads the library to use the LCD.

```c
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup() {
  lcd.begin(16, 2);
  lcd.print("I'll keep learning");
}

void loop() {
  lcd.setCursor(0, 1);
  lcd.print(millis() / 1000);
}```
This code syncs the pins of the Arduino to the LCD's.

```cpp
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup() {
    lcd.begin(16, 2);
    lcd.print("I'll keep learning");
}

void loop() {
    lcd.setCursor(0, 1);
    lcd.print(millis() / 1000);
}
```
lcd.begin() tells how many columns and rows are there in the LCD.

lcd.print() what the LCD will display.
lcd.setCursor() sets the cursor position where the next text will appear, col 0 & row 1.

lcd.print() displays the number of seconds passed since the Arduino was reset.
DC Motors
Based on the magnetic field, rotary motors move based on the current flowing through it.
The goal is to have an Arduino that could control the DC Motor speed and direction it spins.
DC Motors

You’ll need these following parts.
DC Motors

The physical setup should look like the diagram on the right.
DC Motors

We initialize the variables.

The `enablePin` turns off the motor regardless of direction.

`in1Pin` and `in2Pin` determines the direction of the motor by directing current.

```
DC Motors

We initialize the variables.

The `enablePin` turns off the motor regardless of direction.

`in1Pin` and `in2Pin` determines the direction of the motor by directing current.

```

<table>
<thead>
<tr>
<th>In1</th>
<th>In2</th>
<th>Motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>GND</td>
<td>Stopped</td>
</tr>
<tr>
<td>5V</td>
<td>GND</td>
<td>Turns in Direction A</td>
</tr>
<tr>
<td>GND</td>
<td>5V</td>
<td>Turns in Direction B</td>
</tr>
<tr>
<td>5V</td>
<td>5V</td>
<td>Stopped</td>
</tr>
</tbody>
</table>
DC Motors

**switchPin** tells the motor to switch directions.

**potPin** determines the speed of the motor.

```
int enablePin = 11;
int in1Pin = 10;
int in2Pin = 9;
int switchPin = 7;
int potPin = 0;

void setup()
{
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
  pinMode(enablePin, OUTPUT);
  pinMode(switchPin, INPUT_PULLUP);
}

void loop()
{
  int speed = analogRead(potPin) / 4;
  boolean reverse = digitalRead(switchPin);
  setMotor(speed, reverse);
}

void setMotor(int speed, boolean reverse)
{
  analogWrite(enablePin, speed);
  digitalWrite(in1Pin, !reverse);
  digitalWrite(in2Pin, reverse);
}
```

<table>
<thead>
<tr>
<th>In1</th>
<th>In2</th>
<th>Motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>GND</td>
<td>Stopped</td>
</tr>
<tr>
<td>5V</td>
<td>GND</td>
<td>Turns in Direction A</td>
</tr>
<tr>
<td>GND</td>
<td>5V</td>
<td>Turns in Direction B</td>
</tr>
<tr>
<td>5V</td>
<td>5V</td>
<td>Stopped</td>
</tr>
</tbody>
</table>
The pins are defined by `void setup()`.

`INPUT_PULLUP` enables usage of the internal pullup resistors of the Arduino. They can be damaged if exposed above 5V.

No worries, you won’t need to use this for the project.
DC Motors

For every loop, the speed is determined by the potentiometer reading.

The direction is determined by the switchPin.

The setMotor() is not a preset Arduino function...

```
void setup()
{
    pinMode(in1Pin, OUTPUT);
    pinMode(in2Pin, OUTPUT);
    pinMode(enablePin, OUTPUT);
    pinMode(switchPin, INPUT_PULLUP);
}

void loop()
{
    int speed = analogRead(potPin) / 4;
    boolean reverse = digitalRead(switchPin);
    setMotor(speed, reverse);
}

void setMotor(int speed, boolean reverse)
{
    analogWrite(enablePin, speed);
    digitalWrite(in1Pin, ! reverse);
    digitalWrite(in2Pin, reverse);
}
```
Void setMotor():

analogWrite(enablePin, speed) affects the speed.

digitalWrite(in1Pin, ! reverse) and digitalWrite(in2Pin, reverse) are opposite values. One assigns a HIGH while the other assigns a LOW.

As you recall from the last Tutorial (#2), current flows from a HIGH to LOW voltage level.

So, assigning opposite values to these input pins would pass current through the DC motor.

The current direction affects the motor direction.
Sensors
There are different types of sensors. The ones we would use are photocells.

These sensors respond to light.
The goal is to have an Arduino LED that dims when a light is shined on the photocell.
# Sensors

You’ll need these following parts.

<table>
<thead>
<tr>
<th>Part</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>10KΩ Resistor</td>
<td></td>
</tr>
<tr>
<td>220Ω Resistor</td>
<td></td>
</tr>
<tr>
<td>Arduino Uno R3</td>
<td>1</td>
</tr>
<tr>
<td>Jumper wire pack</td>
<td>1</td>
</tr>
<tr>
<td>Half-size Breadboard</td>
<td>1</td>
</tr>
<tr>
<td>Photocell</td>
<td></td>
</tr>
</tbody>
</table>
Sensors

The physical setup should look like the diagram on the right.

The schematic is below.
Sensors

photocellPin connects to Arduino pin 0.

The readings from the photocell are stored in photocellReading.

LEDpin was connected to Arduino pin 11.

LEDbrightness will store the LED’s brightness.
In `setup()`, the serial monitor is initialized to gather information about the photocell.

The serial monitor prints the analog reading from the photocell.
Sensors

To view the information sent to the serial monitor, you could open it as shown in the right image.

It would only work if the Arduino is running the code.
As commented, the *photocellReading* adjusts to the darkness of the sensor. Since the LED only goes from 0 - 255, the reading is adjusted by the *map()* function. Then, it is written into the LED.

```
void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  int photocellReading = analogRead(pinh); // the analog reading from the sensor divider
  Serial.print("Analog reading = ");
  Serial.println(photocellReading);
  int LEDpin = 11;
  int LEDbrightness = 255;
  LED brightness = map(photocellReading, 0, 1023, 0, 255);
  analogWrite(LEDpin, LEDbrightness);
  delay(100);
}
```
Now you’re finished!

Time for the project!
Preview

1. What is Feedback?
2. Mini-Project Goal
   a. Objective
3. Description
   a. Requirements
   b. Schematic
   c. Hints
4. Uploading to Google Drive
What is Feedback?
Feedback

Feedback is a reaction from an input to a system.

In our project, light would trigger our project via a sensor.
Mini-Project Goal
Objective

To build a zoetrope whose speed is dependent on the light shining on a photoresistor.

The Zoetrope Arduino setup would be a variation of your kit’s Arduino Project Book on p103.

**Warning**: this is not a copy-and-paste project. There are some variations that you have to decide with your team on how to do it.
Feedback Cycle

When light shines on the Arduino, the motor would speed up as feedback:

- Input is light
- A = present motor speed
- B = increase in speed
- Output is motor speed
### Requirements:

<table>
<thead>
<tr>
<th>Use a <strong>photoresistor</strong> that affects motor speed.</th>
<th><strong>Optional:</strong> 3D print a base to stabilize the zoetrope described in your project book.</th>
<th><strong>Optional:</strong> you may lasercut materials to substitute some of the materials described in book.</th>
</tr>
</thead>
</table>

Use an **LCD** to display the zoetrope’s speed.

*Shouldn’t exceed 70.*
You could use serial monitor w/ potentiometer to find what the right speed is.

**Post code (or CAD) in Google Drive:**

But, you must,

- Comment your group mates’ names
- Comment what each part of your code means

**Post video of Zoetrope that must:**

- Show zoetrope start from OFF
- Flash light to speed up the zoetrope’s motor
- Turning off light will slow down zoetrope up to rest.
This is the basic setup of the zoetrope without the use of the photoresistor. This could be found in your kit’s Arduino Project Book on p103. Use it as a foundation before substituting the photoresistor.
Hints:

Look at the code in p106-109 of your project handbook to find out how to substitute the photoresistor. Which parts of the code relate to the motor? I uploaded this code unto the Drive.
Hints:

Here are the parts I used. You are not limited to these:

<table>
<thead>
<tr>
<th>Part</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCD Display (16x2 characters)</td>
<td>1</td>
</tr>
<tr>
<td>10 kΩ variable resistor (pot)</td>
<td>1</td>
</tr>
<tr>
<td>Half-size Breadboard</td>
<td>1</td>
</tr>
<tr>
<td>Arduino Uno R3</td>
<td>1</td>
</tr>
<tr>
<td>Jumper wire pack</td>
<td>1</td>
</tr>
<tr>
<td>1 KΩ Resistor</td>
<td>1</td>
</tr>
</tbody>
</table>
Hints:

Here are the parts I used. You are not limited to these:

- DC Motor
- 1 Photocell (Light Dependent Resistor)
- 9V Battery Snaps (w/ battery)
Hints:

A step forward, this diagram hints what could be substituted for the potentiometer from the diagram in the Arduino project book.
Hints:

The concepts you learned in Tutorial 3 could help you with this project.
Uploading to Google Drive
Instructions

1. In the MDC Directory Tutorial’s folder, you will find the folder for Microcontrollers.

2. Create a folder with your group name (based on the project you’re working on)

3. Upload these following files:
   a. Code
   b. Video of Arduino performing what’s required.