

**Engineering Notebook**

**Barnabas-Bot**

Build your own robot!

**16-SESSION**

**DATES:**

NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

COMPUTER ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SESSION 1 - INTRO TO ROBOTICS**

**OVERVIEW:** Today, we will learn about the basic parts of a robot (body, heart, brain and soul), and also about how we will work together to build, learn, teach and inspire through robotics!

**STEP 1:** The meaning of Barnabas

* Barnabas means the “Son of Encouragement “
* Barnabas was a community builder
* What does the word “encouragement” mean?
* What does the word “community” mean?

**STEP 2:** Understanding our goals.

* **BUILD**. What will we build?
* **LEARN**. What will we learn?
* **TEACH**. Who will teach the class?
* **INSPIRE**. What does it mean to inspire?

**STEP 3:** What is a robot?

* Draw your favorite robot.
* What do robots help us do?

**STEP 4:** What makes up a robot?

* Draw today’s robot example
* Write down the four main parts of a robot

**REFLECTION**

**What was your favorite thing about today’s class?**

**What was the hardest part of today’s class?**

**How can you help make our class a good community?**

**SESSION 2 – BUILDING OUR FIRST ROBOT**

**STEP 1:** Our first robot challenge

* We are going to work together to build a robot out of household items.
* Your team will be given a servo motor that spins half a circle (or 180 degrees).
* Your team will also be given a kit of parts.
* Your job will be to work together to build a robot that can hit a cup that is 8 inches away.
* Each kit of parts is different, so you’ll probably need to share parts.
* We have a common goal of all groups succeeding, so remember to work together!

**STEP 2:** Robot Review

* Write down the 4 main parts of a robot
* Fill in the blank!

The things in the bag and the servo is the robot’s \_\_\_\_\_\_\_\_\_

The Arduino controller is the robot’s \_\_\_\_\_\_\_\_\_\_\_

The battery is the robot’s \_\_\_\_\_\_\_\_\_\_\_\_

The program on the Arduino is the robot’s \_\_\_\_\_\_\_

**STEP 3:** Designing **your** first robot

* Look at your parts together and think about how you can build a robot to achieve the challenge. Write down your ideas, or draw what you think your robot will look like.

**STEP 4:** Designing **our** first robot

* Share your design with your team and come up with a design that you would like to build together as a team.

**STEP 5:** Build!

**ROBOT DESIGN REFLECTION**

**What did you learn about servo motors?**

**What did you learn about building your robot’s body?**

**Why is design important?**

**Did your final robot look exactly like your design? If it is different, explain how it is different.**

**TEAM REFLECTION**

**Did you work well as a team? Why or why not?**

**Did you work well with other teams? Why or why not?**

**What is encouragement?**

**What makes up a good community?**

**SESSION 3 - BUILDING OUR**

**ROBOT’S BODY**

**OVERVIEW:** Today, we will learn how to design your robot’s body using computer aided design. We will use these files to 3-D print your one-of-a-kind robot!

**STEP 1:** Getting Started

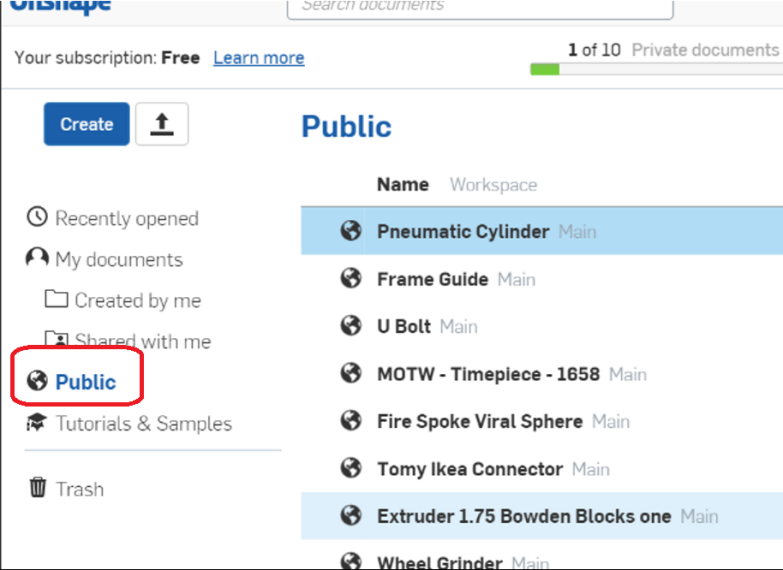
* Go to [www.onshape.com](http://www.onshape.com). If your teacher has provided you with an account, skip to STEP 3.

**STEP 2:** Create An Account

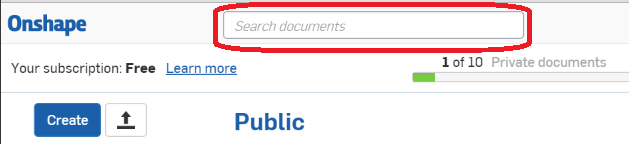
* Use your own email address

**STEP 3:** Search For Barnabas-Bot

* Login to your account and L-CLICK (left click) on “Public” on the left side of your screen.



In the search box on the top left, type in “BARNABAS-BOT 2.2” and search.

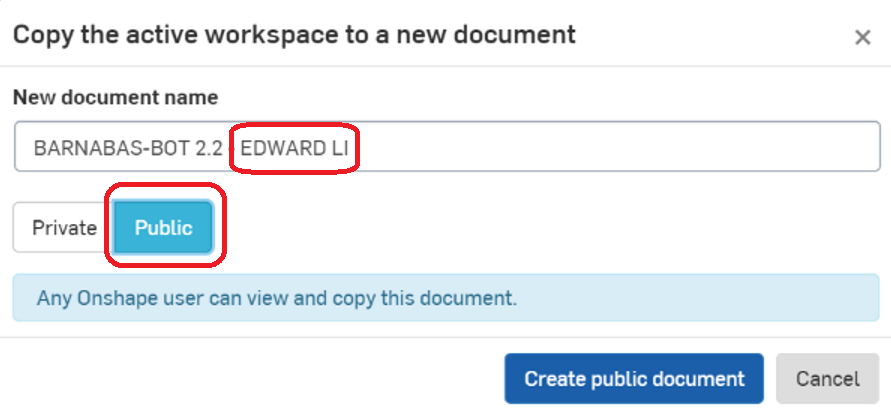
****

**STEP 4:** Create Your Own File

* If you don’t see a file with your name on it, L-CLICK on “BARNABAS-BOT 2.2”. Once it opens, L-CLICK on the icon show below,

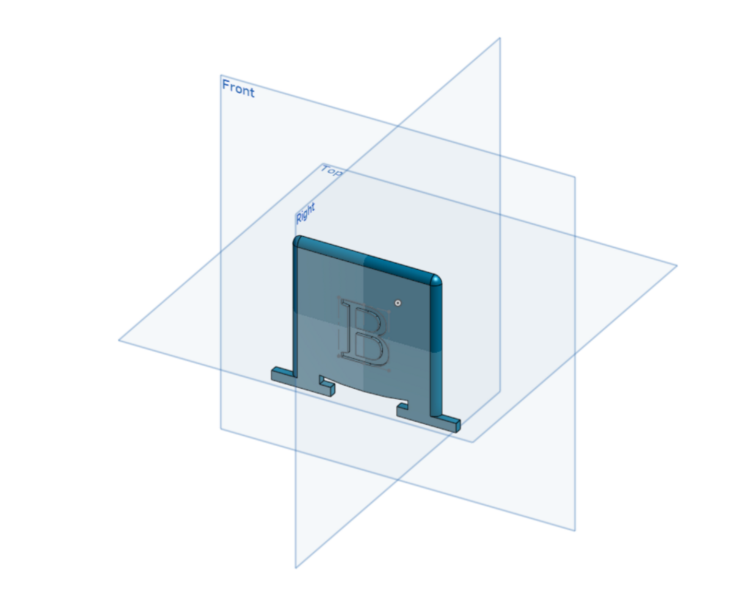


and L-CLICK on “Copy Workspace…”. Make the name: “BARNABAS-BOT 2.2 – [YOUR NAME]”. Also, make sure to select “Public”. L-CLICK “OK”.

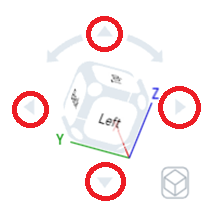


**STEP 5:** Viewing

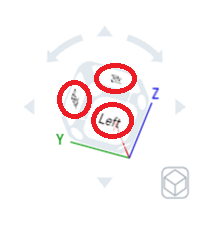
* Open up “robot body front plate”. You will find it in one of the tabs on the bottom of your screen. You should see the following shape.



* 1. **Rotating** **(Method 1)**: Click on arrows in the cube on the top left to rotate the face plate

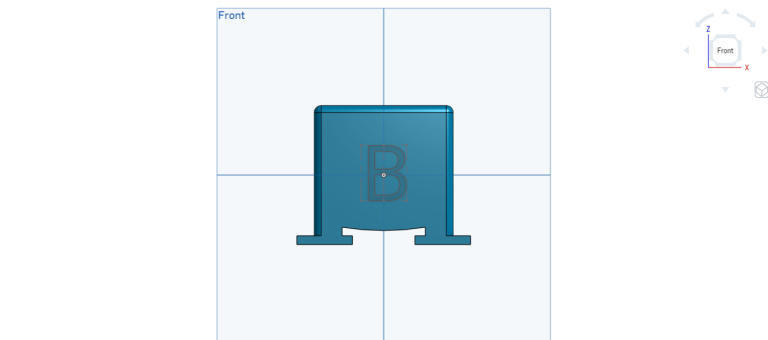


1. **Rotating** **(Method 2)**: Click on the “FRONT, BACK, LEFT, ETC.” On the cube on the top left.



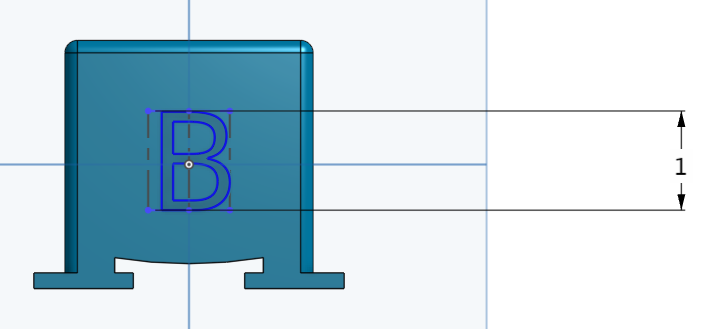
1. **Rotating** **(Method 2)**: R-CLICK on a part and move the mouse
2. **Translating (Method 1):** Hold down CTRL and R-CLICK (right click) on it to move it around
3. **Translating (Method 2):** Click down using the SCROLL button and move it around
4. **Zoom:** Move the scroll button up forward and backward

* Go to the “FRONT VIEW”
* Press “F” to ZOOM TO FIT”. It should look like:

****

**STEP 6:** Customizing The Front Plate

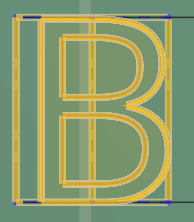
* Change the size of the text
  1. Double L-CLICK on the “B” until you see a “1” pop up on the right side of the screen



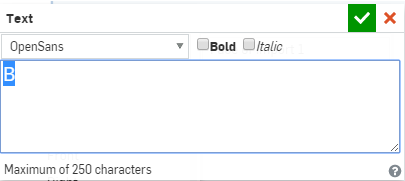
* 1. Change the “1” to “2”. The B should be larger now!
  2. Change the value to “0.5”. The B should be smaller now!
  3. Click on the green check mark to save your changes



* Move the location of the text
  1. R-CLICK on the “B” and L-CLICK on “Edit Sketch 1…”
  2. L-CLICK on one of the 4 blue corners (they are kinda small, you might need to zoom in) of the rectangle around the B and drag the blue rectangle up and down



* 1. Click on the green check mark to save your change.
* Change the text
  1. R-CLICK on the “B” and L-CLICK on “Edit Sketch 1…”
  2. R-CLICK on the “B” again and L-CLICK on “Edit Text”



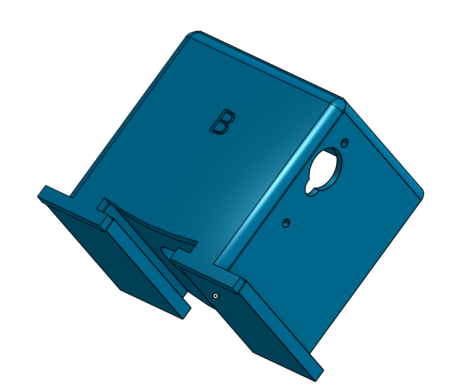
* 1. Now change the text to “C”. Click on the green check mark and see if it changes!
  2. Now change the text to “BARNABAS”. You’ll see that it doesn’t fit on the robot face. How can you make it fit? (Change the size of the text to 0.3 or less)
  3. Click on the green check mark and see if it changes!
  4. Now change the text to whatever you want! It can be a single letter, or many letters. Play with different sizes. Make sure that all of the text fit on the robot! Remember to click on the green check mark to save your changes!

**STEP 7:** Create An Assembly File

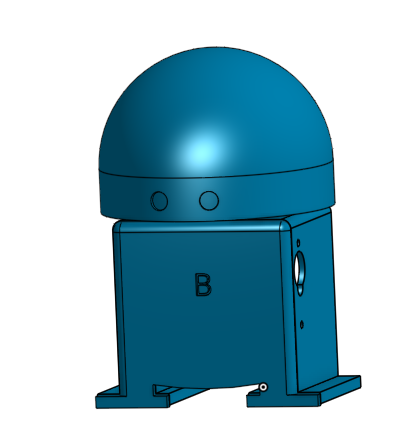
* Selecting your robot’s head
  1. Look at the following files:
     + “robot dome head”
     + “robot head mike1”
     + “robot head mike3”
  2. Select the head that you would like to use for your robot, and write it down
* Create an assembly file
  1. L-CLICK on the “+” sign at the bottom left of your screen
  2. L-CLICK on “Create Assembly”
  3. Your assembly file should automatically open up
* Inserting the body
  1. Insert your body by clicking on the insert button



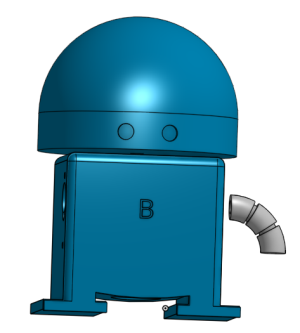
* 1. Now select the “robot body” file by selecting it in the list of files. Place it in your assembly file and click on the green check mark to save your work!
  2. Now go to the Front View
* Inserting your custom plate
  1. Now insert your “robot body front plate” file.
  2. Align the front plate with the body by left clicking on it and dragging it. You’ll need to rotate the views to get it to align correctly
  3. It should look something like:



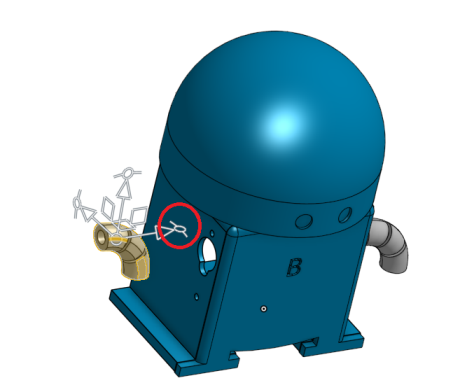
* Inserting your head
  1. Using the same process as before, insert the head that you want



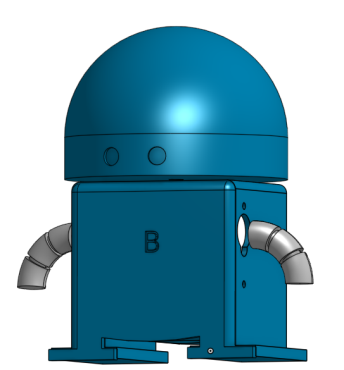
* Inserting your head
  1. Using the same process as before, insert the head that you want
* Insert your left arm and align it



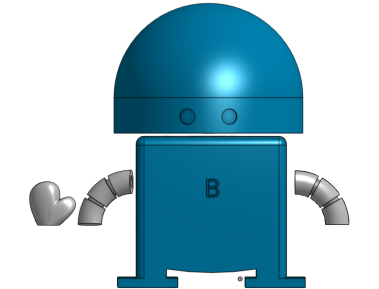
* Insert your right arm
  1. You’ll need to rotate it to align it correctly
  2. L-CLICK on the arm to show the rotation tool
  3. Click on the circle shown below and rotate it



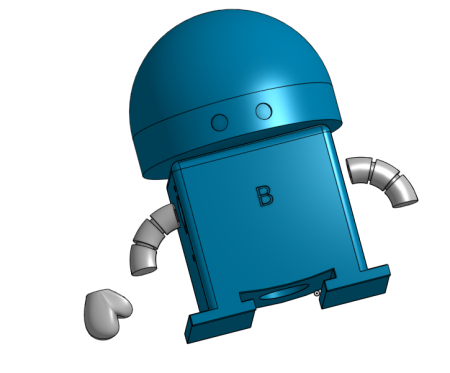
* 1. After it is rotated, align it correctly with the hole



* Insert your hand on the left side



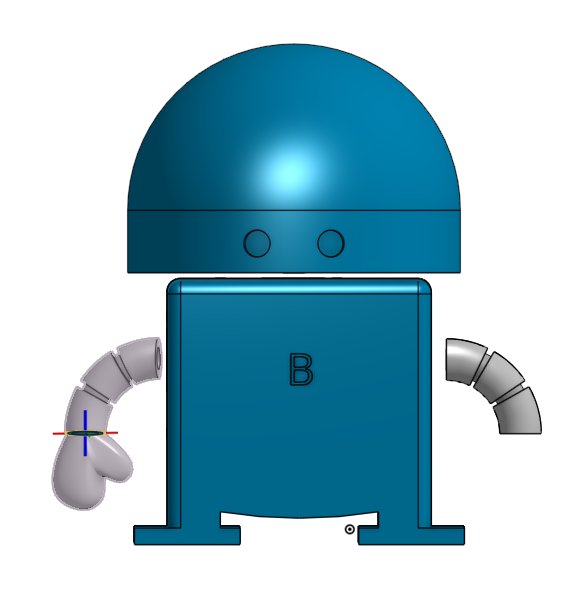
* 1. We’re going to learn a new tool, called the “Fasten Mate”. Using it, you can click on two faces, and it will automatically link them together
  2. Rotate your hand so that it is like this



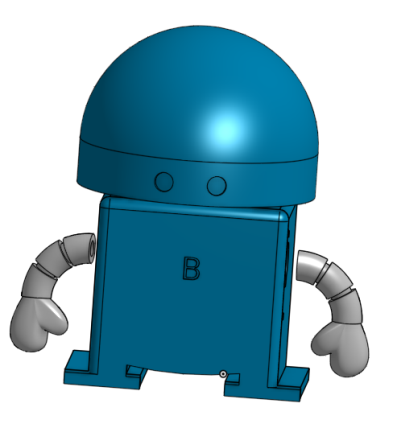
* 1. L-CLICK on the “Fasten Mate” icon



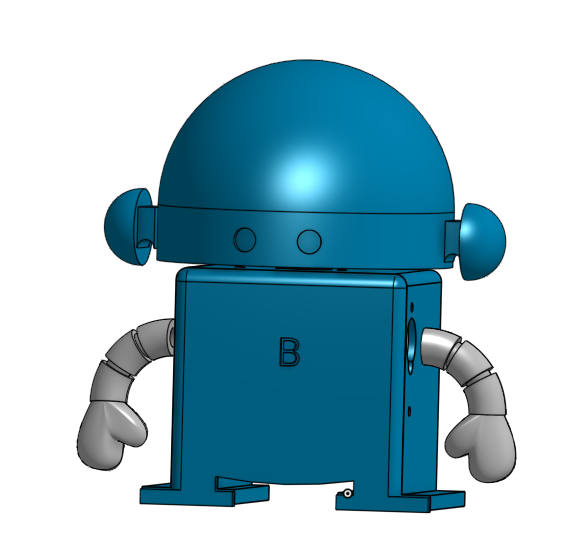
* 1. L-CLICK on the center of the bottom of the hand
  2. L-CLICK on the center of the bottom of the arm
  3. Now it should look like this!



* Now try doing the same with the other hand



* If you have time, try adding ears!



**REFLECTION**

**What does C.A.D stand for?**

**What do we do with a C.A.D. file?**

**What was the most difficult part of today’s class?**

**What was your favorite part about today’s class?**

**SESSION 4 – ADVANCED CAD**

**OVERVIEW:** Today, we will learn how to add shapes to our robot’s front plate.

**STEP 1:** Getting Started

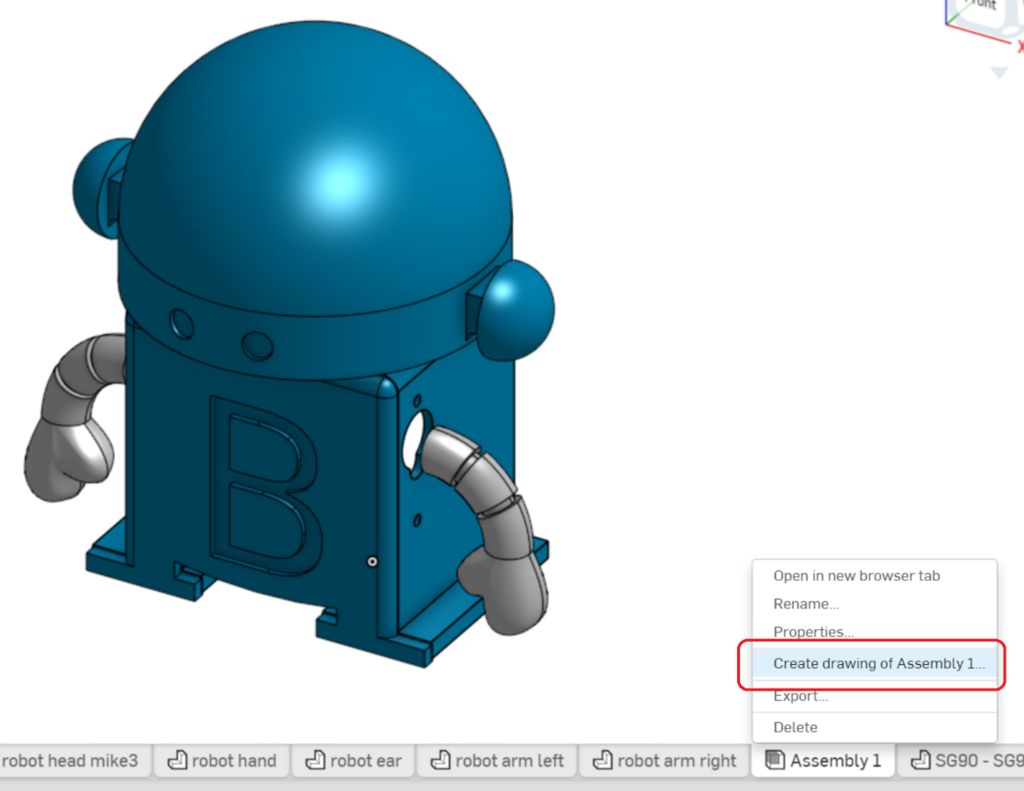
* Go to [www.onshape.com](http://www.onshape.com). Login to your account and open up your robot project.

**STEP 2:** Adding Shapes

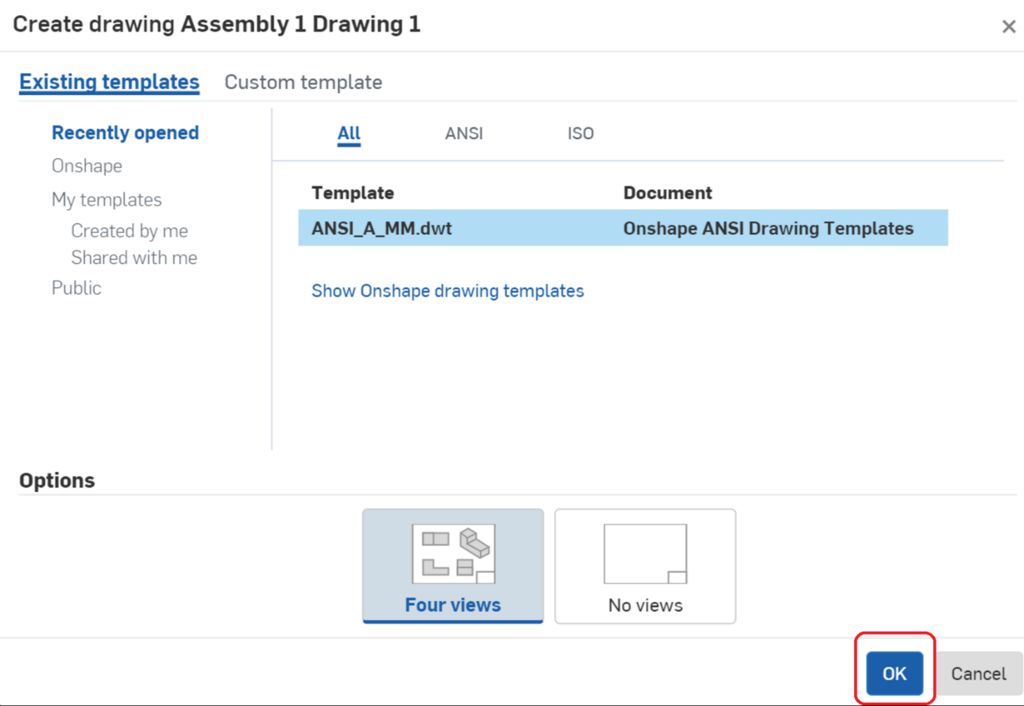
* Go to <https://app.schoology.com/page/635966221> and watch the video to learn how to add shapes to your robot’s front plate.

**STEP 3:** Create a Mechanical Drawing

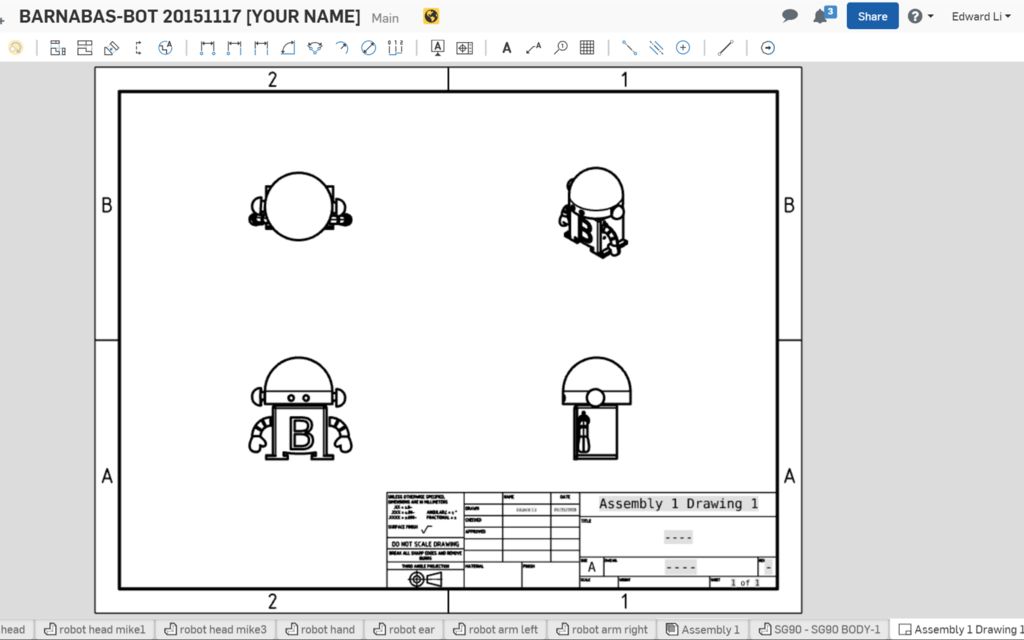
* R-CLICK on the assembly file tab that you just created
* L-CLICK on "Create drawing of ... "



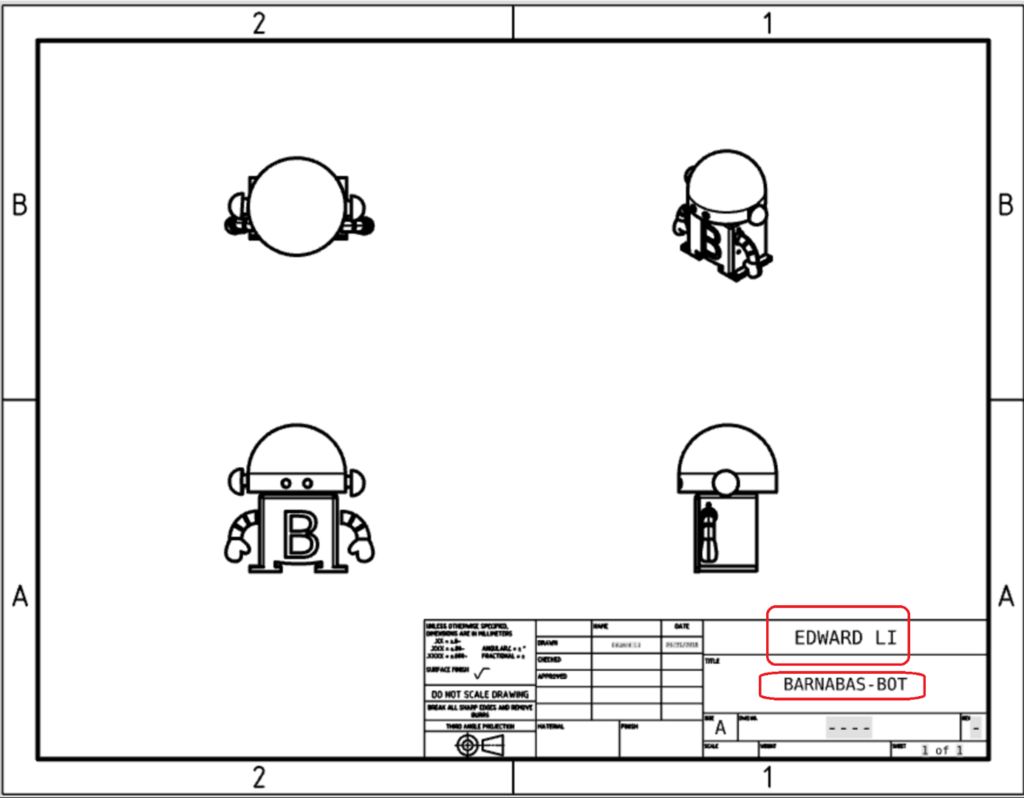
* A screen will pop up. Make sure that "Four views" is selected under "OPTIONS", and L-CLICK on "OK"



* A drawing will be automatically created.



* Double L-CLICK on the text boxes to edit the text. Write your name as well as your robot's name.



**SESSION 5 – 3-D PRINTING & DESIGN**

**OVERVIEW:** Today, we will learn about the world of 3-D printing!

**STEP 1:** Draw a diagram of a 3-D printer

**STEP 2:** Answer these questions.

* What type of file is sent over from the computer to the 3D printer?
* What is the name of 3-D printer part that heats up the plastic?
* Why do engineers use 3-D printers?

**STEP 3:** 3-D printing demonstration

* What did we have to do with the CAD file before we could start printing?
* What was the temperature of the extruder when it started to print?
* What object was printed out?
* How long did it take to print?
* What did you learn about 3-D printing that you didn’t know before?

**STEP 4:** Design Activity

* Describe your favorite type cell phone (apple, Samsung, or other).
* Design the ultimate cell phone.

**STEP 5:** Why do we design?

* What are the two things that we need to remember when we design things?

**STEP 6:** Designing your robot

* Who is your robot for?
* What is your robot’s name?
* What is his/her purpose?
* How do you plan to paint your robot?

**REFLECTION**

**What was the most difficult part of today’s class?**

**What was your favorite part about today’s class?**

**SESSION 6 – TIME TO PAINT!**

**OVERVIEW:** Today, we will paint your robot. We typically use acrylic paint. Make sure that you follow your teachers instructions on how to paint. Have fun!

**SESSION 7 – EXPLORING THE ROBOT’S HEART**

**OVERVIEW:** Today, we will be learning how the “heart” of our robot works. Remember, the heart refers to the battery of our robot, and it has to do with the area of electrical engineering. Have fun!

**STEP 1:** Robot review

* Make a simple sketch of your robot. Label the 4 main parts of a robot.
* Can you remember what type of engineers work on each part of the body?

**STEP 2:** Learning more about how your robot’s heart works.

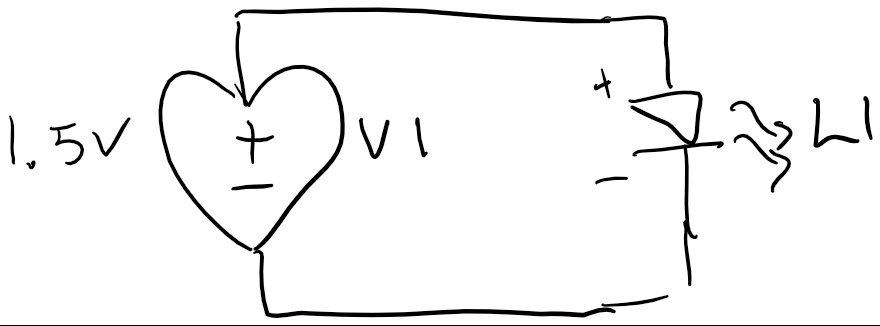
* **What is current?**
* **What is voltage?**
* **What is the role of a battery?**
* **What is the role of a wire?**
* **What is a circuit?**

**STEP 3:** More about circuits

* **What is a schematic?**
* **What is a closed-circuit?**
* **What is an open-circuit?**
* **What is a short-circuit?**
* **What is a resistor?**

**STEP 4:** Reading our first schematic

* Review the schematic below.



* 1. The heart (V1) is the battery
  2. The 1.5V refers to the voltage, which has to do with the power of the battery. The unit of voltage is Volt.
  3. The triangle-like symbol (L1) represents a LED (Light Emitting Diode)
  4. Both V1 and L1 are *components*.
  5. Notice that there are positive and negative sides on both components. You will need to pay attention to these when you start building the schematic, because they need to match up, just like in the schematic.
  6. Draw a path through the “circulation loop” on your schematic. It should start at the positive side of the heart and end on the negative side of the heart.

**STEP 5:** Building our first circuit

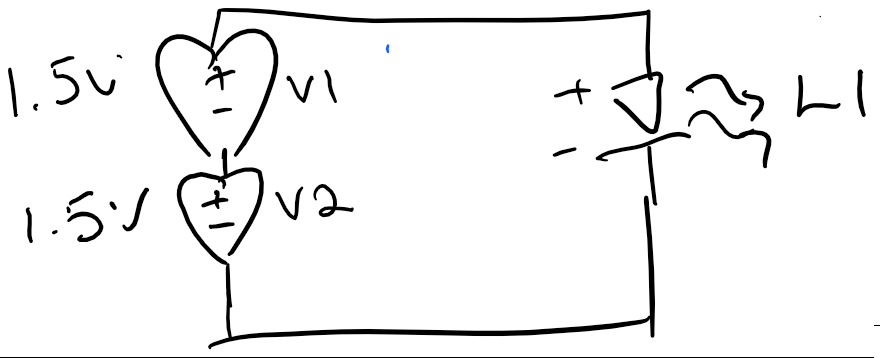
* Take out your AA battery and AA battery holder. Check the voltage on the battery. It should match the voltage label on your schematic.
* Place your AA battery into the battery holder. Make sure that it is in the right direction. The red wire should be connected to the positive side of the battery, and the black wire should be connected to negative side of the battery.
* Take out your LED. NOTE: the longer wire coming out of the LED is the positive side.
* Take some time to put your circuit together, and see if you can get your LED to turn on!

**STEP 6:** Did it work?

* Our LED needs 3 Volts to turn on. Take a look to see what voltage our battery puts out. You’ll notice that we don’t have enough!
* Now think about how we can get our battery to turn on.

**STEP 7:** Two batteries are better than one.

* One way to address our issue is to combine two batteries. Review the new schematic below. You’ll see that we now have two batteries, which will double our total voltage to the LED.



* Find a partner and build your new circuit based on the schematic. Remember to keep track of the positive and negative sides of all your components. The connections need to match your schematic!

**STEP 8:** Review questions

* What is the color of your LED once it is on?
* What is the total voltage that the LED now sees?
* What happens when the wire connections are loose?
* What happens when the wire connections are tight?

**STEP 9:** Bonus Challenges

* Try to make a circuit that uses 4.5V.
* Try to make a circuit that turns on two lights at the same time.
* Try to make a circuit that turns on three or more lights at the same time!

**REFLECTION**

**What was the most difficult part of today’s class?**

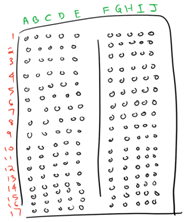
**What was your favorite part about today’s class?**

**SESSION 8 – INTRODUCTION TO BREAD-BOARDS**

**OVERVIEW:** Today, we will learn about breadboards. Breadboards helps us make circuits in a way where we can make strong connections without needing to hold the wires together.

**STEP 1:** Introducing the breadboard

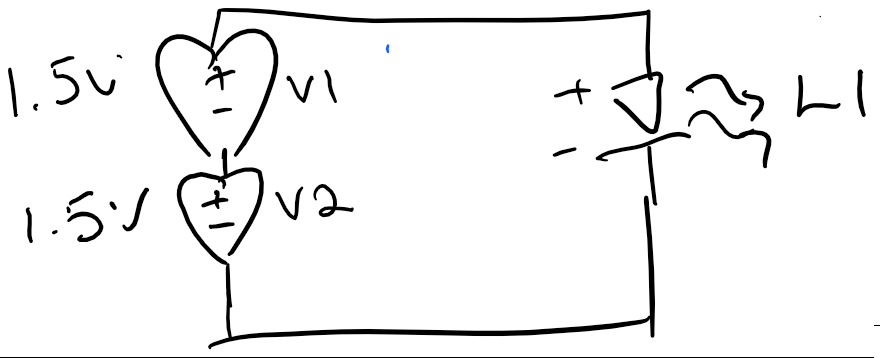
* Take a look at the picture of the breadboard below

****

* 1. Notice that each row has 10 holes in it. They are divided into sets of 5 (A-B-C-D-E, and F-G-H-I-J) by a column down the middle.
  2. How many rows are there?
  3. Each set of 5 holes has an individual wire underneath that connects the 5 holes together. To illustrate, draw the three lines below to show three of the wires.
     + Draw a line that connects A1, B1, C1, D1 and E1.
     + Draw a line that connects A2, B2, C2, D2 and E2
     + Draw a line that connects F1, G1, H1, I1 and J1
  4. Draw in the rest of the wires. How many total wires do you end up with?

**STEP 2:** Our first breadboard circuit

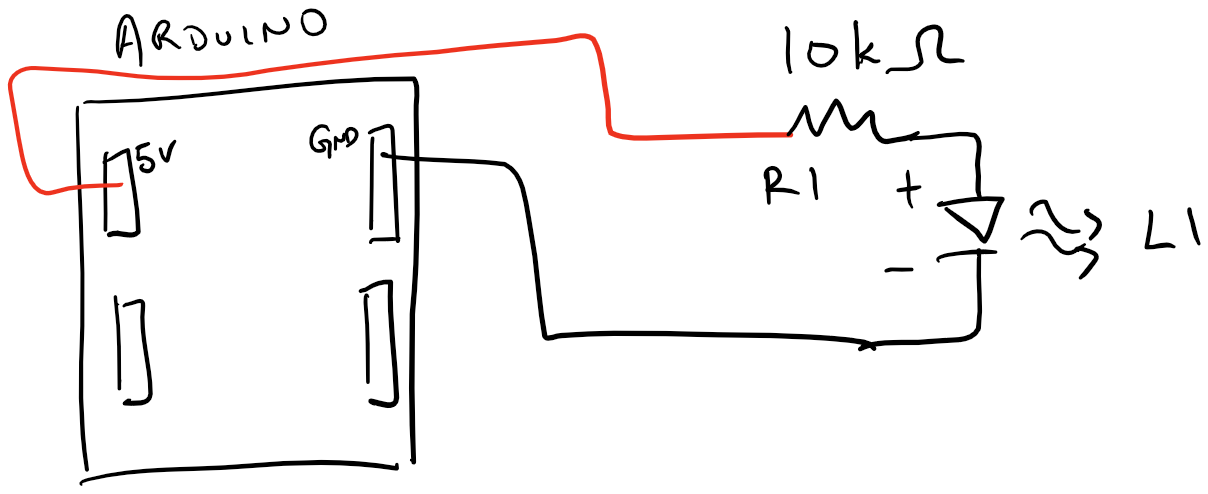
* Using what you’ve learned, now use the breadboard to make your 3V LED circuit.

****

* The goal here is to create a circuit that stays on without you needing to hold on to any wires. HINT: You can stick the AA battery holder wires and LED wires directly into the breadboard.

**STEP 3:** Increasing the voltage

* Now that we have learned how we use schematics to plan circuits, and how to actually build them, we will look at increasing the voltage, because our robot is going to need more than 1.5V to run!
* Review this new schematic



* 1. Note that Arduino is acting like the battery in this case (kind of – more on this later). The 5V is the positive and the GND is the negative.
  2. What is the voltage of this new circuit?

* 1. Notice R1. R1 is a resistor. The value is 10K Ohm, or 10,000 Ohm. Do you remember when we need to use resistors?
  2. On the schematic, trace the path where electricity will flow. It should start at 5V and end with GND
* Now build this circuit using your breadboard. HINT: You will need your 10K Ohm resistor (it has an orange stripe). You will also need two bread-board wires. Use a red and black one for this challenge.
  1. Once you are done, you’ll notice that the light does not turn on. This is because you need to provide a main power source. In this case, our 9V battery will be our main power source. The Arduino will then take the 9V and reduce it to 5V for our robot. This reduction in voltage is called “step-down” voltage regulation. Sometimes you will have a circuit that increases the source voltage, which is called “step-up” voltage regulation.
  2. Plug the 9V battery into the 9V battery holder, and stick it into the round power connector. Your light should now turn on!

**STEP 4:** Review questions

* The higher the resistor, the lower the flow of electricity. The lower the resistor, the higher the flow of electricity. If we change the resistance from 10K to 470 Ohm, what do you think will happen to the light?

1. Try replacing the 10K Ohm resistor with the 470 Ohm resistor. What happens to the light?

1. Put the 10K Ohm resistor back into your circuit.

* The flow of electricity is called ***current***. If you have a lot of flow, you have high current. If you have a little flow, you have low current. You can actually calculate this current using a formula, called Ohms Law. This is an important tool for engineers when they design circuits. The formula is:

**V (Voltage) = I (Current) x R (Resistance).**

The unit is Amps.

1. If the voltage is 5V, what is the current when you use the 10K Ohm resistor?
2. If the voltage is 5V, what is the current when you use the 470 Ohm resistor?

* Have you seen a light switch that you turn like a dial? Think about how you might build something like that based on what you have learned.

**SESSION 9 – THE BLINKING LIGHT CHALLENGE**

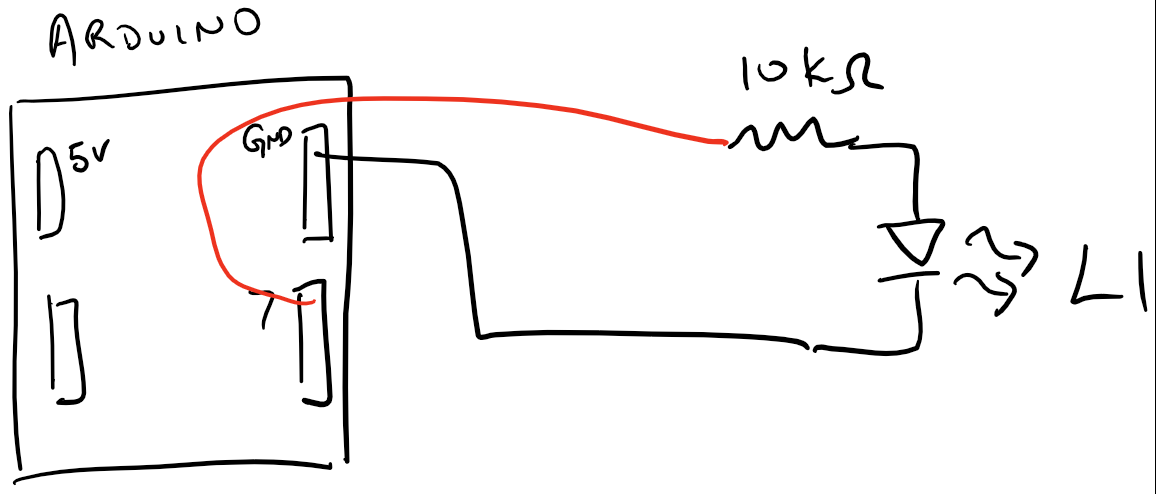
**OVERVIEW:** This week, we will be creating a light circuit for your robot.

**STEP 1:** What is a power indicator?

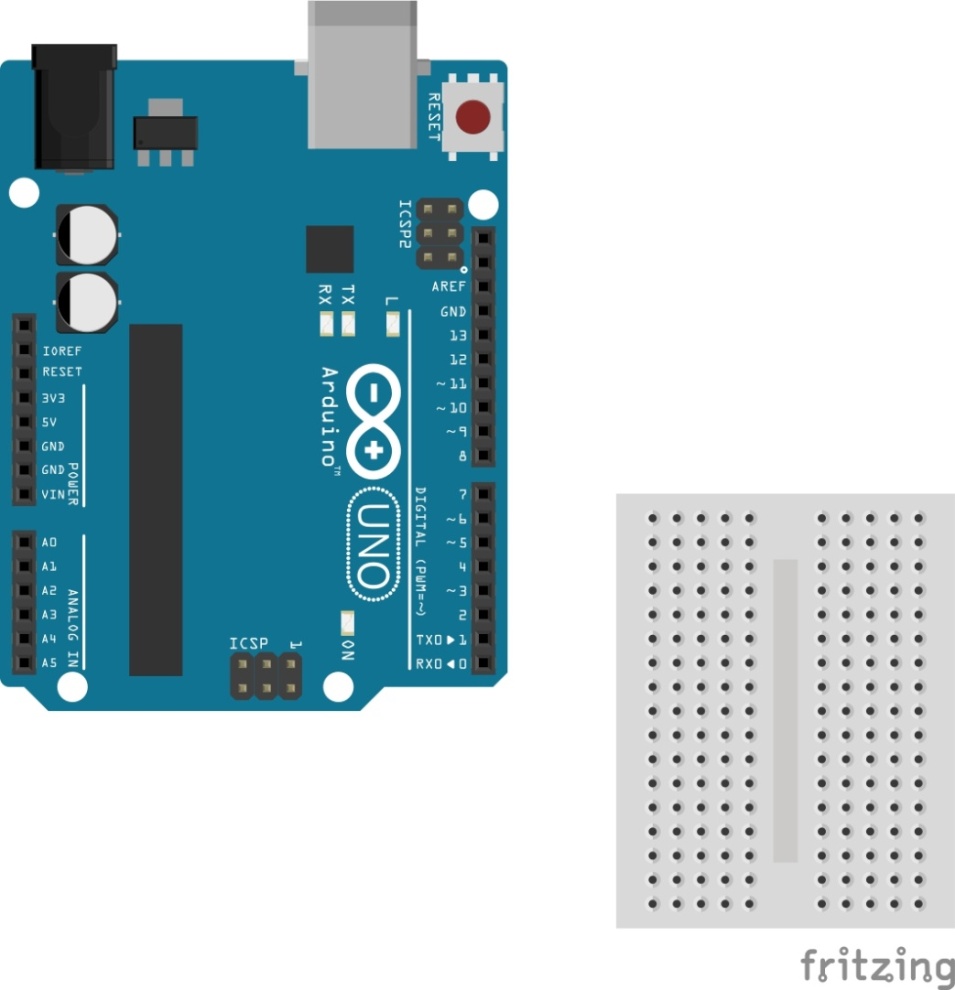
* What typically happens when you turn on electronics (cell phone, computer, tv, etc.)?
* Electronics use power indicators to let us know that electronics are on. Write down the two common components that are used to build power indicator circuits. Remember that a circuit is the “loop” that is formed that allows electricity to flow through your electronics. Every circuit needs a battery, or heart.

**STEP 2:** Building your circuit

* Build the following circuit based on the schematic below. Hints: You will need a resistor, LED and two jumper wires. Also, remember that there is a plus and minus on the LED. The minus of the LED should be connected to GND.



* Note that pin 7 is a digital output. This will allow you to turn your light on and off using robot’s brain (the Arduino)!
* How many components do you see in your schematic?
* When you are breadboarding, remember that two different sides of a component are always connected to DIFFERENT wires (or rows).
* Using the breadboard drawing below, let’s create a wiring diagram so that we know how to build our circuit.
  1. Draw in your components (LED, Resistor, Arduino). Remember that the LED has a plus and minus – the resistor doesn’t.
  2. Starting from the pin 7 on the Arduino, draw in the connections that will complete our circuit. Remember that you’ll need to end up with a loop so that electricity can flow!



1. Make sure that your drawing is correct. Once you have it confirmed, go ahead and build your circuit!

**STEP 4:** Connecting your Arduino

* You are now ready to connect your Arduino to your computer so that you can start programming!
* Using the USB cable, plug in the Arduino to your computer. Make sure to first unplug any other batteries that are connected to your Arduino when you do this.



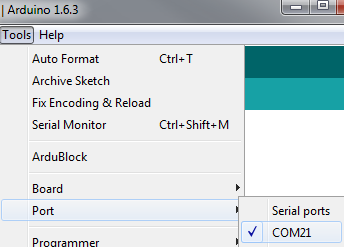
**STEP 5:** Opening the Software

* Open the Arduino software by clicking on the Arduino icon on your computer desktop.
* Wait for the software to open



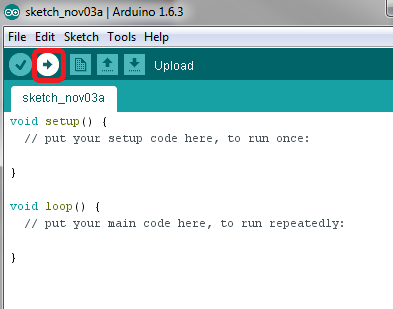
**STEP 6:** Selecting the COM Port

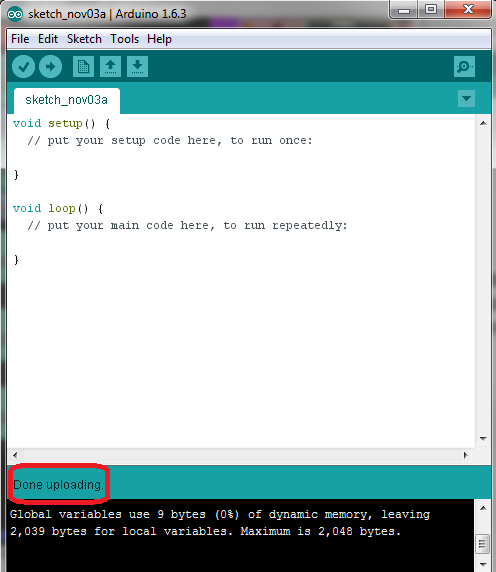
* Go to TOOLS->PORT, and click on the item that starts with the 3 letters, “COM”. It says “COM21” in the picture below, but yours may say a different number. That’s OK.



**STEP 7:** Uploading

* Click on the UPLOAD button and wait. It will take a few minutes. You will notice a green progress bar at the bottom right of your screen. If you see “DONE UPLOADING”, then you are good to go! If you see red on your screen, you selected the wrong “COM”, so go back to STEP 6, select a different “COM” port and try the upload again.

**

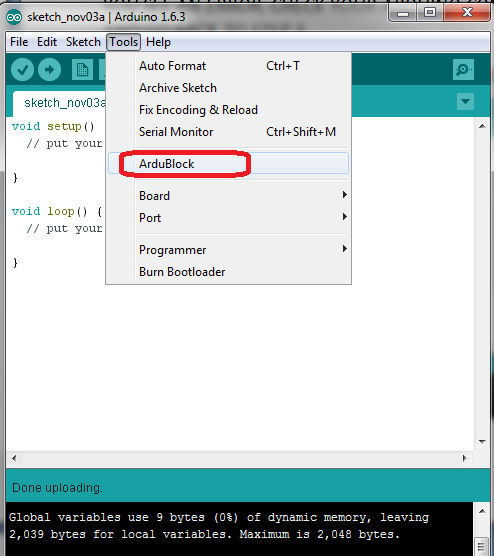


**STEP 7:** Uploading

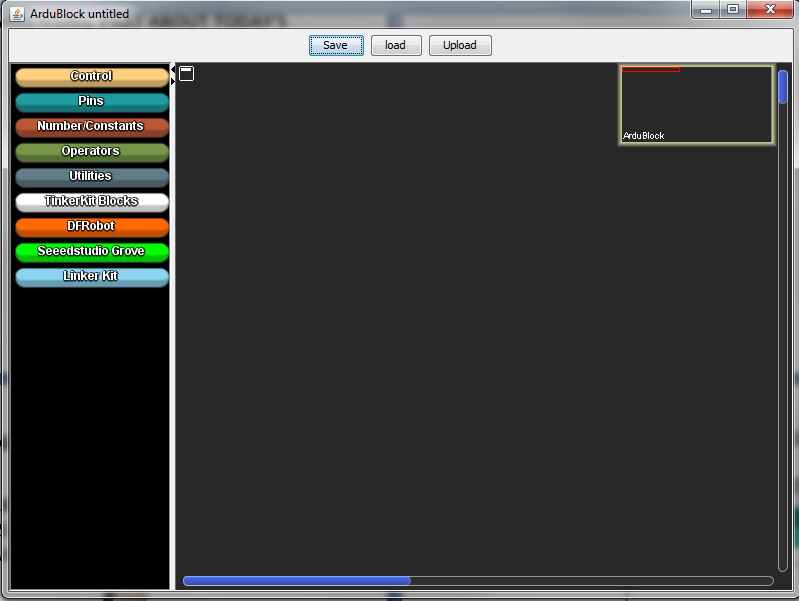
* Click on the UPLOAD button and wait. It will take a few minutes. You will notice a green progress bar at the bottom right of your screen. If you see “DONE UPLOADING”, then you are good to go! If you see red on your screen, you selected the wrong “COM”, so go back to STEP 6, select a different “COM” port and try the upload again.

**STEP 8:** Open ArduBlock

* You are now able to write a program to your Arduino! Now open ArduBlock by going to TOOLS->ARDUBLOCK.

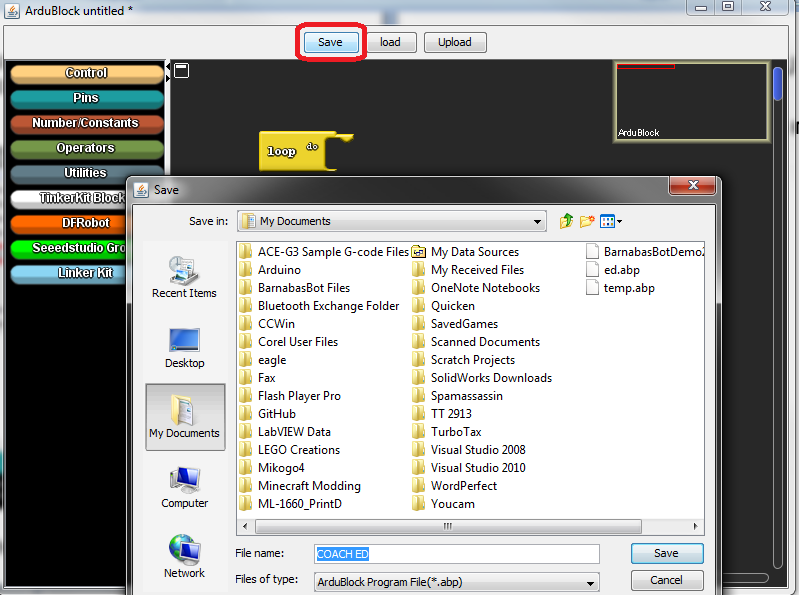
****

* The following screen will appear

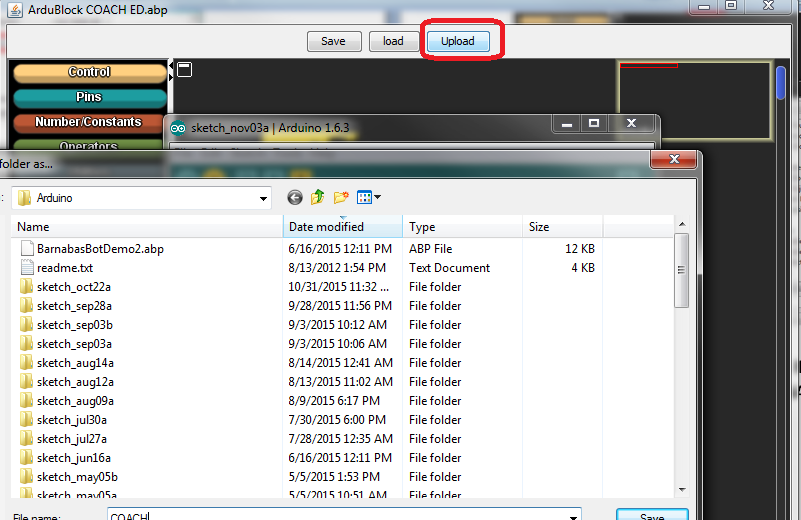
****

**STEP 9:** Writing Your First ArduBlock Program

* Go to [CONTROL], and drag in a [LOOP-DO] block. Click on [SAVE], type in your name, and click [SAVE] again. Note that every program needs a [LOOP-DO].

**

* Now click [UPLOAD], type in your name, and click [SAVE]. Your program will now upload!

**

**STEP 10:** Programming your LED

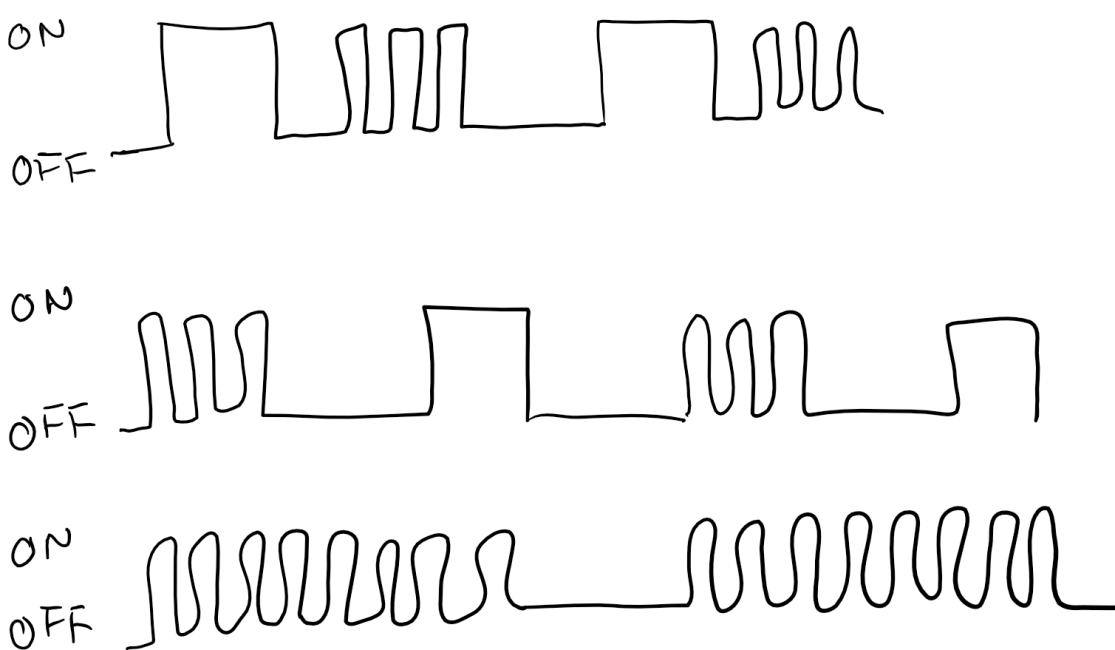
* Notice that we have a wire connected to digital output 7. Can you guess which programming block allows us to control digital output 7. HINT: look under [PINS]. Write down the name of the block.
* Notice that there is a #, and a HIGH/LOW. What do you think we should put as the pin #?
* Try writing a program to turn your LED on. Remember to click [SAVE] and then [UPLOAD]
* Try writing a program to turn your LED off. Remember to click [SAVE] and then [UPLOAD]

**STEP 11:** Make It Blink!

* Now you will need a new block. Go to [UTILITIES]->[delay milliseconds]. This block tells your program to wait a certain amount of time. Note that 1000 milliseconds is the same as 1 second.
* **CHALLENGE 1:** Try writing a program that turns your light on for 1 second, off for one second, and then repeats forever
* **CHALLENGE 2:** Write a program that turns your light on for ½ a second, off for ½ a second, and then repeats forever.
* **CHALLENGE 3:** Write a program that makes your light blink as fast as possible!
* **CHALLENGE 4:** Write a program that makes blinks 5 times fast, 5 times slow, and repeats forever. Hint: use the repeat block



* **CHALLENGE 5:** Write a program that can match these light patterns below.



* **CHALLENGE 6:** Create your own light patterns and challenge your classmates to program them!

**REVIEW**

**What pin number on the Arduino controls the light?**

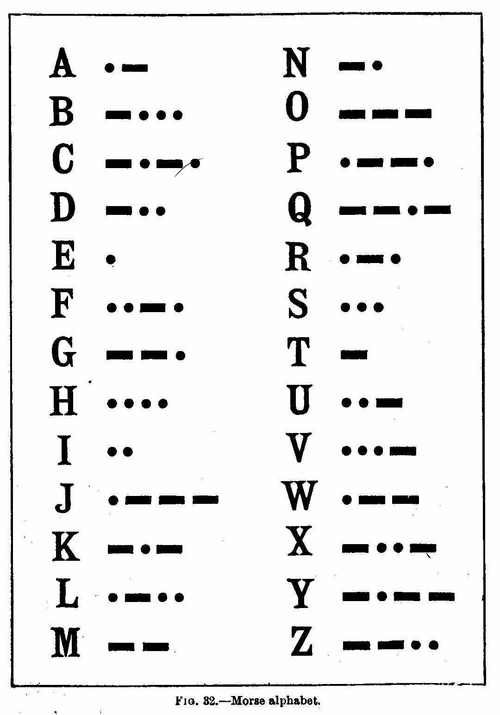
**What was the most difficult part of today’s class?**

**What was your favorite part about today’s class?**

**SESSION 10 – BLINKING IN CODE**

**STEP 1:** Morse Code

* Before there were cell phones, we used to use a telegraph to send information using short and long taps. In the past, people have also used light signals to make short and long blinks.
* You’ll need to following information to complete the challenge:
  1. Every time you see a dot, you want your light to turn on for 100 ms.
  2. Every time you see a dash, you want your light to turn on for 300 ms.
  3. In between a dash or dot within the same letter, turn the light off for 100 ms.
  4. When you write a new letter, turn the light off for 1000 ms.



**SECRET CODE CHALLENGE**

**Write down a word or short message that you would like your robot to say in Morse code.**

**Code your message and see if another classmate can read it out!**

**REFLECTION**

**What was the most difficult part of today’s class?**

**What was your favorite part about today’s class?**

**SESSION 11 – ASSEMBLING THE ROBOT**

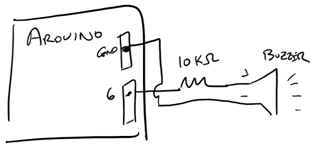
For directions, go to: [**http://tinyurl.com/hvn4ztp**](http://tinyurl.com/hvn4ztp)

**SESSION 12 - THE BUZZER**

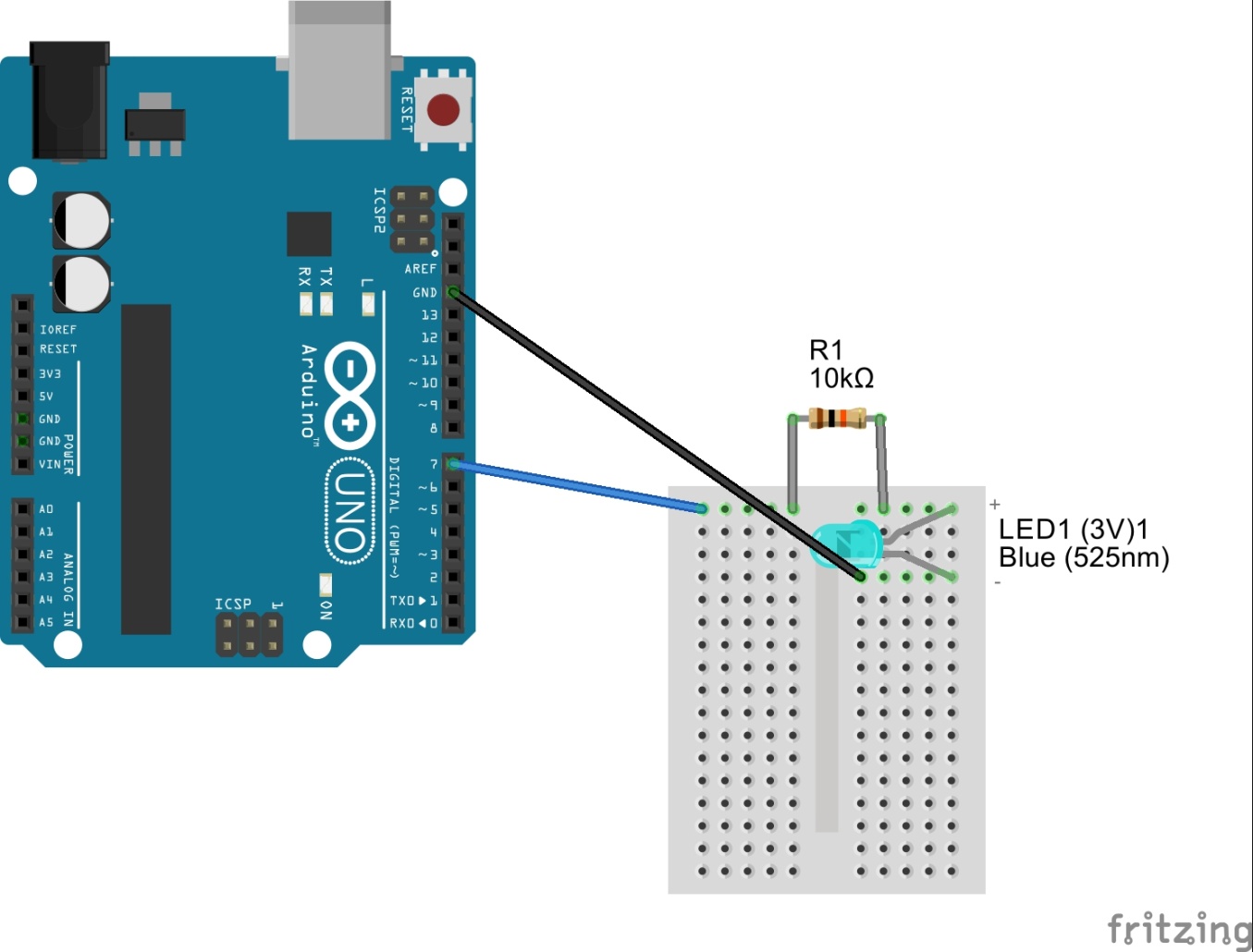
**OVERVIEW:** This week, you will be learning how to use a buzzer to play music!

**STEP 1:** Building your circuit

* We will not be taking our circuits apart from last week. Instead, we will be ADDING to it!
* Build the following circuit based on the schematic below. Hints: You will need a resistor, buzzer and two jumper wires. Can you guess which is the positive side of the buzzer?



* How many components do you see in your schematic?
* When you are breadboarding, remember that two different sides of a component are always connected to DIFFERENT wires (or rows).
* Using the breadboard drawing below, let’s create a wiring diagram so that we know how to build our circuit.
  1. Draw in your components (Buzzer, Resistor).
  2. Go ahead and draw in the necessary connections. However, this time, we don’t want to connect GND directly to the Arduino. We want to connect it on a row on the breadboard. See which row is already connected to GND!



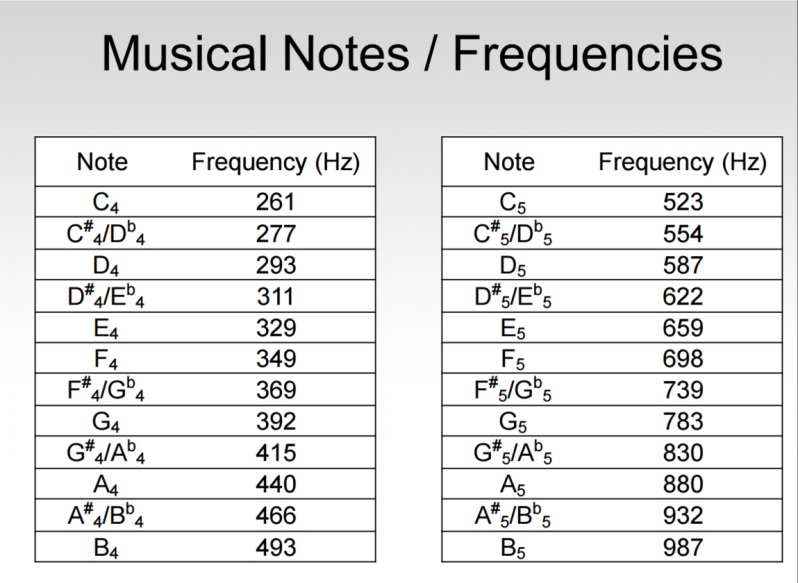
1. Ask your teacher to see if your drawing is correct. Once you have it confirmed, go ahead and build your circuit!

**STEP 2:** How does the buzzer work?

* Note that pin 6 is a PWM. A PWM is a signal that allows us to change the voltage that goes into the buzzer. As we change the voltage, the buzzer sound will change!
* Every sound that your buzzer makes creates a certain frequency. This frequency makes different notes – just like a piano. Can you think about how we might play music now?

**STEP 3:** Frequency Table

* Using the frequency table below, you will be able to play specific notes to make music!
* Note that you have low notes (left column) as well as high notes (right column).

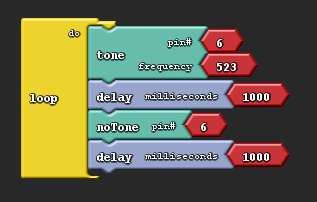
****

**STEP 4:** Music time!

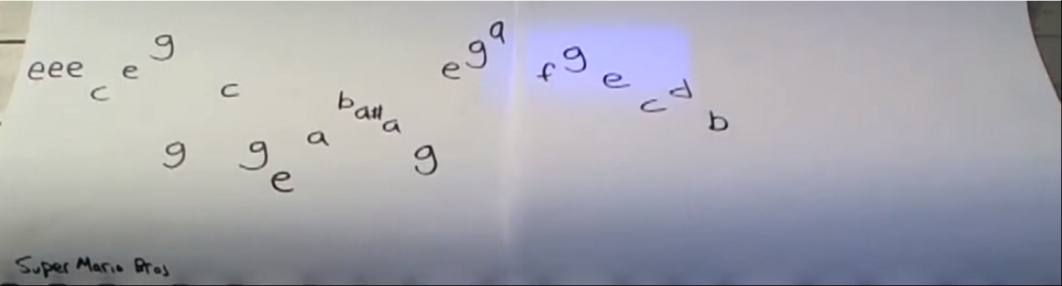
* Try to challenges below. Because we are using a 10K resistor, you’ll need to put your ear pretty close to the hear the sound.
* Connect your Arduino. If you forget how to start up ArduBlock, go back to the directions from the previous week
* Program blocks. This time, you’ll need a few more blocks:
  + PINS-> tone
    - Pin# should match your PWM pin
    - Frequency should match the frequency of the note that you are trying to play
  + PINS-> noTone
    - Pin# should match your PWM pin
    - Frequency should match the frequency of the note that you are trying to play
* Challenges
  + **CHALLENGE #0**: Play “C” forever!



* + **CHALLENGE #1:** PLAY “C” FOR 1 SEC, REST 1 SEC, AND REPEAT FOREVER



* + **CHALLENGE #2**: PLAY “C” FOR 500 MS, REST 500 MS AND REPEAT FOREVER
  + **CHALLENGE #3**: PLAY “B” FOR 500 MS, REST 500 MS AND REPEAT FOREVER
  + **CHALLENGE #4**: PLAY “B” FOR 1 SEC, PLAY “A” FOR 1 SEC, PLAY “G” FOR 1 SEC, REST FOR 1 SECOND AND REPEAT FOREVER. You just played Hot Cross Buns!
  + **CHALLENGE #5**: PLAY TWINKLE-TWINKLE(CC GG AA G FF EE DD C)
  + **CHALLENGE #6**: PLAY HAPPY BIRTHDAY(G G A G C B G G A G D C G G G E C C B A)
  + **CHALLENGE #7**: PLAY MARY HAD A LITTLE LAMB (B A G A BB B <> AA A <> BB B <> B A G A BBBB AAA BAG)
  + **CHALLENGE #8**: PLAY SUPER MARIO BROTHERS!
    - Note that there are both low notes and high notes on this one!

****

* + **CHALLENGE #9**: WORK WITH YOUR NEIGHBOR TO PLAY A SONG TOGETHER!
  + **CHALLENGE #10:** LOOK UP YOUR FAVORITE SONG AND PLAY IT
  + **CHALLENGE #11:** PLAY YOUR OWN SONG!

**REFLECTION**

**WHAT DID YOU LEARN TODAY THAT YOU DIDN’T KNOW BEFORE?**

**WHAT WAS YOUR FAVORITE PART ABOUT TODAY’S CLASS?**

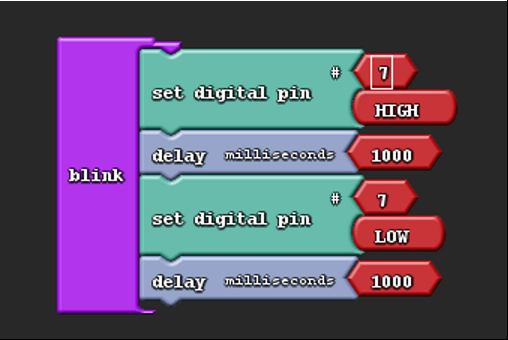
**WHAT WAS THE MOST CHALLENGING PART ABOUT TODAY’S CLASS?**

**SESSION 13 – Advanced Coding**

**OVERVIEW:** Today we will learn more tools that every software engineer needs to know!

**STEP 1:** Subroutines

* Sometimes you do the same things in your code. However, it gets tiring writing the same commands over and over again. This is why we have subroutines!
* Subroutines are little tasks that we can call at anytime.
* Create the following subroutine.



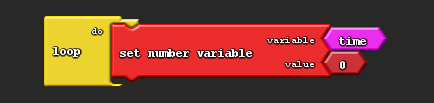
* Call the subroutine by putting the following into your LOOP-DO.



* **CHALLENGE #1:** Blinking Light Challenge
  1. Write a subroutine that turns on your light for 1 second.
  2. Write a subroutine that turns off your light for 1 second.
  3. Call both subroutines in your LOOP-DO.
* **CHALLENGE #2:** Lights and Music
  1. Write another subroutine that plays the song, Hot Cross Buns
  2. Using your subroutines, create a program that turns on the light while it starts to play Hot Cross Buns, and turns off the light when it is done.

**STEP 2:** Variables

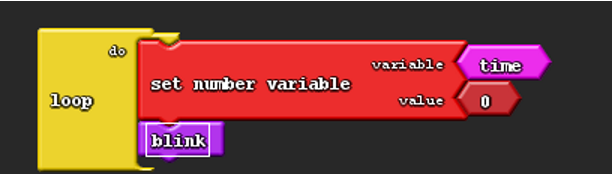
* We use variables whenever we want to change the value of something within our program.
* To use variables, we first need to create one. Let’s name our varable, “time”. We will set the initial value to 0.



* Let’s now use the variable in our blink subroutine.



* Think about what this code below is doing. Now try it and see what happens.



* **CHALLENGE #1:** Make the light blink slowly just by changing the “set number variable” block.
* **CHALLENGE #2:** Make the light blink very fast just by changing the “set number variable” block.
* **CHALLENGE #3:** Make a program that plays “HOT CROSS BUNS”, but this time use a variable in the delay block.
  1. Make the music play fast just by changing the “set number variable” block.
  2. Make the music play super fast just by changing the “set number variable” block.
* **CHALLENGE #4**: Using your subroutines, create a program that:
  1. Turns on the light while it starts to play Hot Cross Buns
  2. Turns off the light when it is done
  3. Repeat this 3 times slowly
  4. Repeat this 5 times fast
  5. Repeat the whole thing again.

**STEP 3:** Conditionals

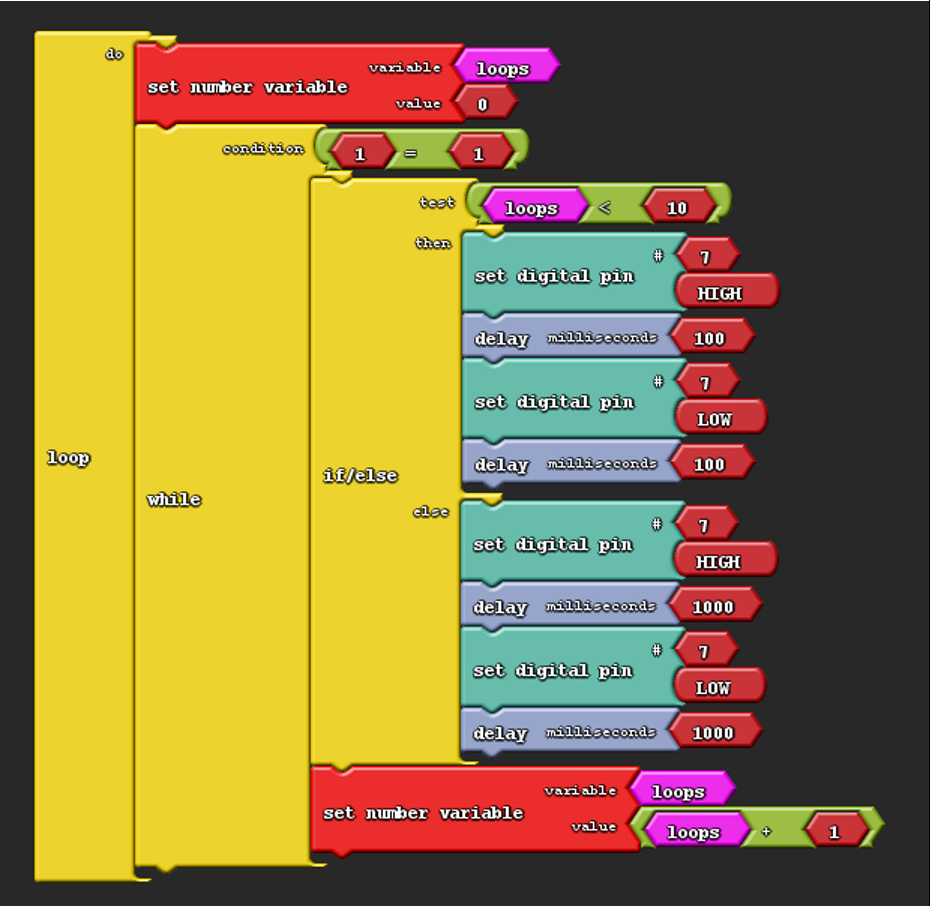
* Sometimes our program needs to test to see if something is true or not in order to know what to do next. This is when we use conditionals.
* The following program will turn the light on if mode = 0. Otherwise, it will turn the light off. Try the program and see what happens.

****

* Try out the program above, and see if you can get the behavior to change just by changing the value of the variable.
* **CHALLENGE #1:** Turn the light off just by changing the value of mode.
* **CHALLENGE #2:** Write a program that only plays Hot Cross Buns when the light is on. Remember to use Subroutines.

**STEP 4:** Math

* We sometimes use math to figure out what our program needs to do.
* The following program keeps track of how many “loops” the program makes and does something different based on that information.



* **CHALLENGE #1:** Can you guess what the program above does? Try running it, and see what happens. HINT: The WHILE block is like a LOOP DO that will run while its condition is true.
* **CHALLENGE #2:** Rewrite the program to use subroutines and variables to control the light
* **CHALLENGE #3:** Modify the program so that it resets the “loops” variable to 0 when “loops” is greater than 20.

**REFLECTION**

**WHAT DID YOU LEARN TODAY THAT YOU DIDN’T KNOW BEFORE?**

**WHAT WAS YOUR FAVORITE PART ABOUT TODAY’S CLASS?**

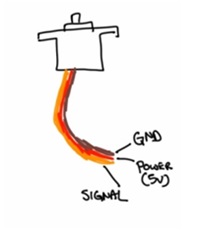
**WHAT WAS THE MOST CHALLENGING PART ABOUT TODAY’S CLASS?**

**SESSION 14 – SERVO MOTORS**

**OVERVIEW:** This week, we will be making your robot wave (or shake his/her head). ☺

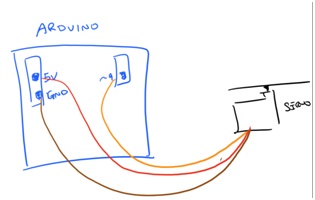
**STEP 1:** Introduction to servos

* The servo motor has an axel that rotates 180 degrees.  It only moves along that circular path.
* There are 3 wires that come out of the servo motor. (RED) powers the servo motor.  GND (BROWN) is ground, and the (ORANGE) wire tells it where to go.



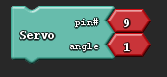
**STEP 2:** Wiring up your motor

* Pull the cables coming out of the servo motors through the hole on the bottom of your robot. There should be three in total.
* Look at the wiring diagram below. Based on what you’ve learned, add the necessary wires to connect your motor!



**STEP 3:** Programming motion

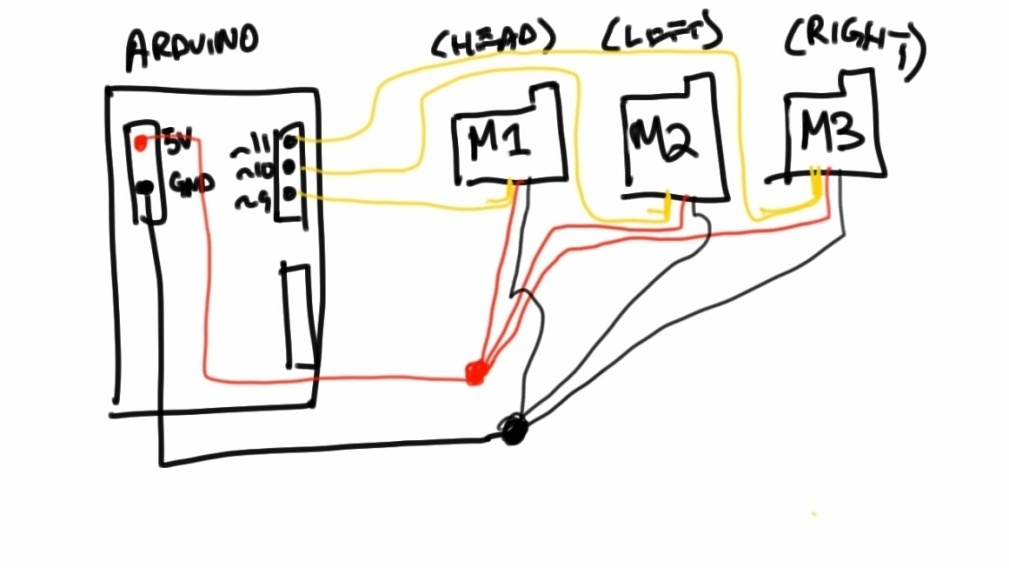
* Connect your Arduino. If you forget how to start up ArduBlock, go back to the directions from the previous week
* Use the following block to move your servo motor.

****

* + PINS -> Servo
    - Pin# should match your PWM pin
    - Angle accepts a value between 0 and 180 degrees
* Challenges
  + **CHALLENGE #0**: Move the motor to 1 degrees

****

* + **CHALLENGE #1:** Go to 90 degrees
  + **CHALLENGE #2:** Go to 0 degrees, wait 1 second, go to 90 degrees, wait 1 second, and repeat.
  + **CHALLENGE #3:** Make the motor move back and forth very fast.
  + **CHALLENGE #4:** Make the motor move back and forth very slowly.
  + **CHALLENGE #5:** Make your robot turn his light on while it is moving, but turn his light off while he is stopped
  + **CHALLENGE #6:** Make your robot light up, play music, and move all at the same time!
* **STEP 4:** Connecting the rest of the motors
  + Now connect the other two motors. See the wiring diagram below for all three motors. You might run out of GND and 5V connections on the Arduino, so think about how you can use the breadboard to make more GND and 5V connections.

****

* + **CHALLENGE #1:** Move your left arm back and forth.
  + **CHALLENGE #2:** Move your right arm back and forth.
  + **CHALLENGE #3:** Move your left and right arms back and forth at exactly the same time.
  + **CHALLENGE #4:** Move all your body parts back and forth at exactly the same time.

**REFLECTION**

**WHAT DID YOU LEARN TODAY THAT YOU DIDN’T KNOW BEFORE?**

**WHAT WAS YOUR FAVORITE PART ABOUT TODAY’S CLASS?**

**WHAT WAS THE MOST CHALLENGING PART ABOUT TODAY’S CLASS?**

**SESSION 15 – THE FINAL DESIGN**

**OVERVIEW:** Today we will be putting together the complete design (body, brain, heart, and soul) of your robot.

**STEP 1:** Your robot’s brain

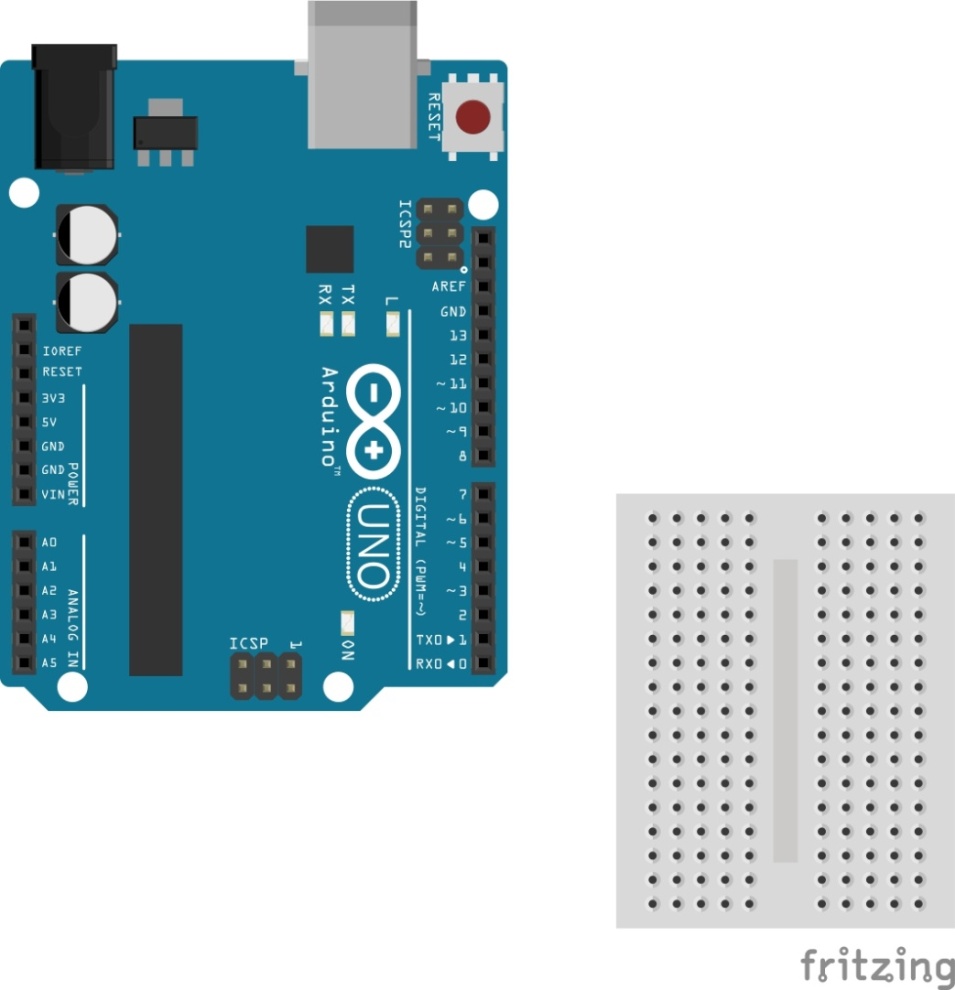
* What is your robot’s brain called?

**STEP 2:** Your robot’s body

* Draw your robot’s body.

**STEP 3:** Your robot’s heart system

* Draw the final wiring diagram for your robot’s heart and veins. Remember to add in all the components (3 x motors, 1 x LED, 1 x buzzer, 2 x resistors)



**STEP 4:** Your robot’s soul

* Plan out what your robot is going to do.
* Some ideas: Program your favorite song, make your robot dance to it, and have your robot say his or her name via the LED using Morse code!

**SESSION 16 – OPEN LAB**

**OVERVIEW:** Program away!

**STEP 1:** Presentation

* We will have short presentation to share what you made. Answer the questions below to prepare.
* What is your name and age?
* What is your robot’s name?
* Tell us the hardest part about designing your robot.
* Tell us the part that you enjoyed the most about building your robot.
* Tell us what you learned about yourself during this class.