1. Make sure you have all the parts of the kit: Arduino UNO R3, USB cable(A to B, just like a printer cable), Master Beta Shield, bullet vibrator with wires stripped(it should come stripped), 2 AA batteries in AA battery holder, a computer running any type of windows, mac OS, or linux
   1. Photo of uno in box
   2. Photo of uno out of box
   3. Photo of USB cable showing both ends clearly
   4. Photo of master beta shield board
   5. Photo of bullet vibrator with wires stripped
   6. Photo of AA batteries inside the holder with lid off
   7. Photo of AA battery holder with lid on, with on/off switch called out in a label on the picture
   8. Photo of computer, showing USB port with a label on photo
2. Go to the website [www.arduino.cc](http://www.arduino.cc) and click on download, or go directly to http://arduino.cc/en/Main/Software . Based on what operating system you have, click on the link on the right with the appropriate installer. Follow the instructions as they pop up to install the software. This website will be a great resource for you as you explore Arduino. There are many useful website out there, including [www.sparkfun.com](http://www.sparkfun.com) and [www.adafruit.com](http://www.adafruit.com) and hopefully ours([www.orgasmatronics.com/arduino [add](http://www.orgasmatronics.com/arduino [add) this?]), but the main site is always a key resource for figuring out software issues and what is out there in new hardware. Once you have the Arduino software installed, open it up so that you can see the main screen on your computer.
3. Take the Arduino UNO R3 out of its box and plug the USB cable into both your computer and the Arduino UNO. The wider and flatter connector connects to your computer and the more square shaped connector connects to the Arduino(just like a printer connection).
4. In the Arduino software, first click on “tools”, then go down to “board”, and make sure there is a check mark next to “Arduino UNO”. “Arduino UNO” should also appear next to the word “board” in the pulldown menu. If that is not the case, go to the menu of boards and select the “Arduino UNO” that that it is.
   1. Screenshot of this menu all the way out, with markings on figure showing where the checkmark goes for UNO
5. In the Arduino software, on the “file” menu, pull down to examples, then basics, then “blink”, then click on blink.
   1. Show screen shot here of how that menu opens up
6. Click on the arrow pointing to the right to upload and compile this program. This should then show some status reports for a few seconds and then say “uploaded to board”. If you see error messages(which are generally in an amber font color), you may want to unplug and replug the board, shut down and restart the arduino software, or change the serial port in the tools pulldown menu. Also see further troubleshooting guides on the Arduino site. If it does work, you should now see a blinking light on your board. Congratulations!! This is your first working Arduino program! You have now made software control the hardware!!!
7. Now go in the code for the blink program and in the code that is as follows:

void loop() {

digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(led, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

}

Delay() is a function that makes the arduino wait for a certain number of miliseconds(thousanths of a second). So 1000 miliseconds is 1 second, and this loop program flashes the on state (HIGH) for 1 second and the off state (LOW) for one second. Now edit it to be different wait times, say 100 and 100 instead of 1000 and 1000 or 2000 and 150, or whatever. This function(delay) is extremely important in all the things we will do here. The Arduino runs at a clock speed of 16 MHz, meaning it can do 16 *million* things in a second. In practical robotics and sex toy projects you probably never want to do more than about 10 things in a second. Maybe 20 or 50, but not more than 100! What do I mean by “things”? turning anything on, off, or whatever. So in order to make the Arduino do things at a human time scale we always need to be tweaking these delays, and miliseconds are the right unit because we generally want to wait less than 1 second but more than 1/1000th of a second. So now you have mastered control of the Arduino’s relationship with time, which basically makes you a time lord!

1. Now plug the vibrator wires into the spring clips on the top output line
   1. Show photograph showing the correct clip to use
   2. Show photograph zoomed in of the stripped wires with caption: note that the stripped parts must be in contact with the spring clips inside in order for electricity to flow.
   3. Show detail of a finger pushing down on the spring clip while a person’s other hand puts one of the vibrator wires into the hole of the spring clip. It takes some force to get the spring clip to open up, and you push down on the tab to make that happen. When the wire is as far in as it will go, release the tab, and it should be gripped by considerable friction in the spring clip.
   4. Photograph of both wires in, with hand tugging gently on the wire to make sure it is properly fixed in there. Caption: note that a gentle tug is a good way to check that the wires are properly situated.
2. Plug the AA battery pack into the input power supply
   1. Show a photograph of the battery pack, with labels showing where the switch is, and marking red as plus and black as minus.
   2. A photograph shows where the input is on the board for this power supply, and shows where it’s marked as being digital pin 5 output.
   3. Following the same procedure as for the vibrator, put the wires in the spring clips, making sure red and black go to the correct terminals. Show photo of correct orientation of wires.
   4. Open the program from the X. Treme tutorial fademotor1.ino, and upload it to the arduino board exactly as in the previous example.
   5. Make sure that if it is not already, the switch is in the “on” position(see above photo)
   6. You programmed a vibrator!! With power over a motor, you can now build ALL OF THE ROBOTS. But you won’t because you’ve seen Terminator 2 and know that will quickly lead to Judgment Day. So you proceed to the next step here and build the next sex toy project.
   7. Again, if you look at the code you see the use of the delay() function. Right now it’s 30 milliseconds, or 0.03 seconds, about 1/30th of a second. If you edit the line delay(30) to be, say, delay(15), it will be half the delay which makes it twice as fast. Try it! Note that like several computer languages, each line has to end in a semicolon or the whole thing will fail in any of various ways. So always keep that semicolon! Also, anything after a double “/” is a “comment”, meaning that is English for other humans to read and is not read by the machine. When you change stuff, comment it, that’s what real coders do and is one reason physicist code makes people sad (we tend to have horrific documentation and commenting). For more information on the Arduino’s language, see the reference page at <http://arduino.cc/en/Reference/HomePage> . It’s almost identical to the language called “Processing”, which is documented here: <https://processing.org>. If you want to build programs where both your computer *and* the arduino do something interesting, processing is a great place to start, but I’m going to skip that in this tutorial(partly because a lot of fun stuff with processing and vibrators can infringe on certain people’s troll-ass bullshit patents, but mostly because I’m lazy).
3. Open the file knob\_motor\_speed\_simple.ino in the Arduino software and upload it to the arduino. Again, make sure the switch on the AA battery pack is in the “on” position. Now turn the knob on the board(see photo showing knob). You have a speed controller! How does this work? The knob is what is called a potentiometer(<http://en.wikipedia.org/wiki/Potentiometer>), colloquially known as a “pot”, which changes the relative position of the middle connection to the two outer ones as you turn the knob. This changes the voltage that goes into an analog input, which is what the code reads(see the code comments for details). If you look at the comments in the code one thing to notice is that inputs are scaled to 1023 and outputs to 255. That is because 2 to the power of 10 is 1024 and 2 to the power of 8 is 256. So a 10 bit input scales to 1023(or 1111111111 in binary) and a 8 bit output scales to 255( or 11111111 in binary). This is not that important exactly but if you want to play with this code yourself you have to be aware of it because these scaling factors have to be carefully dealt with or you end up with crazy numbers.
4. Now open the program button\_simple.ino in the Arduino software, and upload it to the Arduino board. Again, be sure the AA battery pack is turned “on”. Now push the button(see photograph of button). Buzz, buzz!! This button is simply connecting a digital input line that is naturally in a zero voltage state with +5V, making it a digital “ON”, and the code detects that and turns the vibrator on.
5. Now open the program sawtooth\_knob.ino in the Arduino software and upload it to the board. This program is a simple “sawtooth wave”, so it ramps up, then abruptly drops to zero, then ramps up again, and so on, unlike the earlier triangle wave that ramps up then ramps back down at the same rate. Now turn the knob. You can change the rate! Unlike a lot of patterns, this might actually be useful/fun! But there’s more…
6. Open the program sawtooth\_knob\_button.ino and load it. This is the same as the previous program only when it reaches the max it just stays there until the button is pressed, which resets it. You can control the ramp time with the knob. Read the comments in the code to see how it works. Again, we’ve made an actual useful toy here! This is a sort of tease, where the user can get close to climax, then reset and repeat, ramping slower and slower.
7. More stuff! You can try the chaos code, and play with that, for endless mathematical/physics fun. Also a 9V battery and battery leads(not included) can let you make the thing self-contained without the USB. Duct tape that whole puppy together and you’ve got a crude but effective controller that you can actually use in the bedroom! For even more, see advanced tutorials on how to use the other features of this board. You can do a ton of things with it! You can control the x1 orgasmatron, various wand or bullet products, valves to control pouring of liquids, LED arrays, and can take input from force and pressure sensors! All of that is for more advanced tutorials however, and it makes sense to truncate this here and save that for future tutorials. The point of this kit is, after all, to introduce beginners. Those with lots of experience can figure out how to use the other features by looking at the Eagle files for the board.