# Dynamic Systems and Control Kit Setup

***Note: The part numbers referenced here can be found on the Parts List handout.***

0. Install Arduino Software

If the Arduino software is not already installed on your computer, go to the website https://www.arduino.cc/en/Main/Software and click on the Arduino IDE installation instructions link for your operating system. If you already have Arduino installed but the version is earlier than 1.6.5 (you can see your current version number at the top of any open Arduino sketch/script), you will need to re-download Arduino. We encourage you to use Windows or Mac OS X, since we cannot provide any support for Linux. ***Note*****:** If you are downloading Arduino for Windows, select either the "Windows Installer" or "Windows Zip File," not the "Windows App" option.

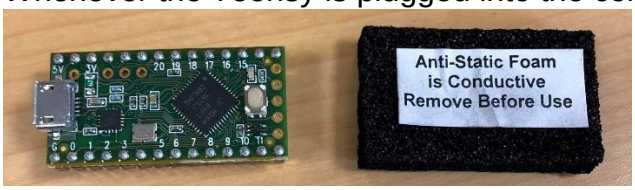
Supplementary Arduino Details:

For this course, we will be using the Teensy LC https://www.pjrc.com/teensy/teensyLC.html, a 32 bit microcontroller board. The Teensy can be programmed with Arduino software, so we will introduce it just as one would introduce the Arduino.

If you are new to Arduino, begin by reading these webpages:
What is Arduino?: https://www.arduino.cc/en/guide/introduction
Arduino Development Environment: http://arduino.cc/en/Guide/Environment

1. Configuring the Teensy

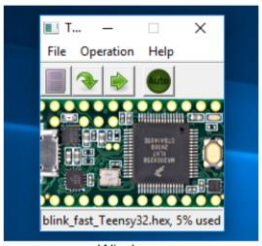
1) Remove the conductive foam on the back of the Teensy (Part #2, with pink wrapping). Whenever the Teensy is plugged into the computer, the foam should be detached.




2) Plug the Teensy into the computer using the USB cable. You should see the Teensy's LED blinking at medium speed.
3) Navigate to https://www.pjrc.com/teensy/loader.html and click on whichever OS you are using.

Macintosh

Linux (Ubuntu)

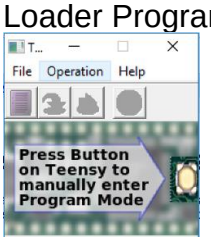


Windows

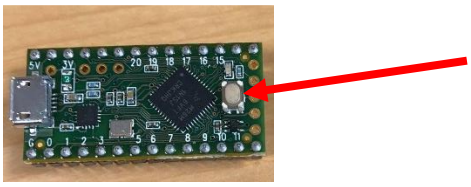
4) Next, you should see these links at the top of the page:

Teensy Loader Program
LED Blink, Both Slow & Fast

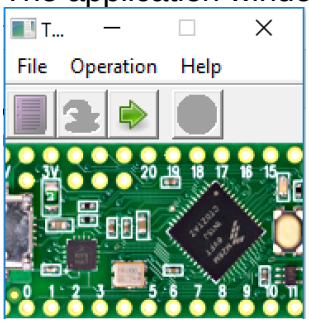
Click both of them. Unzip/extract the blink_both.zip file and remember the directory where the files are located. Open the teensy.exe file that downloaded when you clicked the Teensy Loader Program. You should see an application window that looks like the following:



5) Press File > Open HEX File. Open the file called “blink_fast_TeensyLC.hex”. Press the white button on the Teensy:



6) The application window should now look like this:



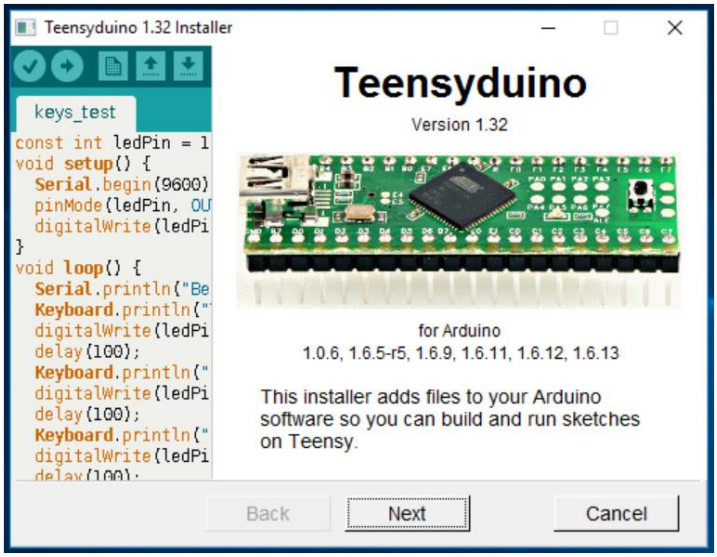

Press Operation > Program, and then Operation > Reboot. You should now see the Teensy’s LED blinking faster.

7) Navigate to https://www.pjrc.com/teensy/td_download.html and select the link that corresponds to your operating system:

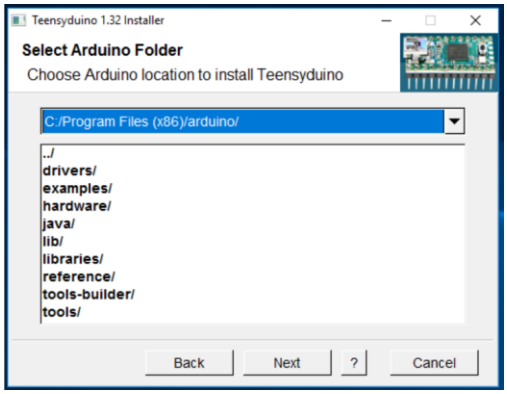
**Teensyduino Files:**

- Macintosh OS-X Installer (10.8 or later)
- Linux Installer (X86 32 bit)
- Linux Installer (X86 64 bit)
- Linux Installer (ARM 32 bit / Raspberry Pi)
- Linux Installer (AARCH64 / Jetson TX2)
- Windows XP / 7 / 8 / 10 Installer

Once you open the software after the download completes, you will see a screen that looks similar to this:

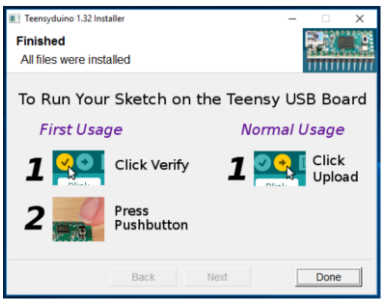



Click Next until you see a screen that looks like this:

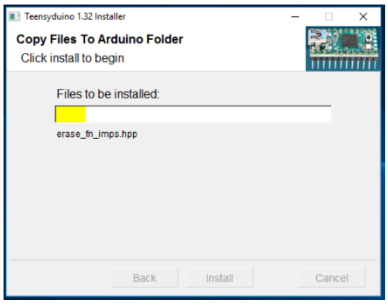



This is now one of the trickiest steps: you need to find the location of Arduino on your computer, but sometimes it is hidden within other folders and sometimes there are multiple Arduino directories on your computer. You will know that you have found the correct location when the "Next" button is clickable (as it is in the picture above). Until then, use trial-and-error to locate your Arduino location.

8) Once you have located Arduino, click Next, OK, and Install. If everything works well, you will see this screen:



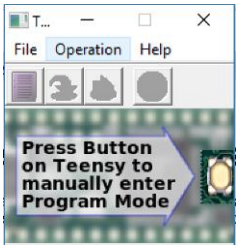

Often though, the installation stops short and notifies you that there is an Error on this screen:




If this happens, close your Arduino application if it is open. Return back to before where you pressed Install, ensure Arduino is closed, and then press Install again.

9) Once you have completed step 8, click "Done" and open up your Arduino environment.


10) In Arduino, go to File > Examples > 01.Basics > Blink. This is a small script that will cause the LED on the Teensy to blink at a desired speed. Go to Tools > Board and select Teensy LC. If that is not an option, close Arduino and re-open it, which should fix the issue. Next, go to Tools > Port. Sometimes there is a default option already selected. If not, select one of the available options. This default communication or the option you select will not be the communication protocol that we use in the future, but you typically need to select one of these options to upload your first script. After doing this, in later assignments it should recognize the Teensy and switch to the preferred communication (Serial communication, over a COM port).
11) Close the Teensy application window shown below:




12) Return to the Arduino environment and the Blink script. Press the Upload button:

13) Once the script is uploaded, the bottom of the screen should show “Done Uploading”. Your Teensy’s LED should be blinking at a medium rate. To confirm that you are able to upload an edited script to the Teensy, change the two delay(1000) lines to delay(100). This changes the delays from 1000ms to 100ms. Once it is done uploading, you should see that the LED is blinking much faster.

If you have previously programmed in C, Java, or many other programming languages, you will find that no additional tutorial is needed in order to program in the Arduino environment. For most assignments, we will provide you with majority of the code needed to complete the labs, and require you to make small, if any, modifications to the code. For any questions that you have regarding the code, we recommend that you use the Arduino Language Reference (http://arduino.cc/en/Reference/HomePage). However, if you are very new to programming or want a refresher, here is a 45-minute video that gives an introduction to programming with the Arduino: https://www.youtube.com/watch?v=yGzzRCoWxBE. Beyond this, you can find a number of web tutorials that will guide you through the design and implementation of Arduino programs (sketches). But don't be afraid to dive in and play with your Arduino code directly -- we highly recommend learning by doing.

## 2. Lab Kit Assembly and Testing

1. **Choose a motor mount (Part#12), a pendulum (Part#13), and a base (Part#14).**
2. **Pendulum Assembly**

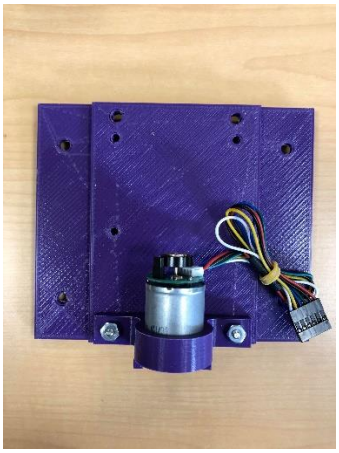
   Insert binder posts (Part#8) and screws (Part#9) into the large holes in the pendulum (Part#13). Before screwing in the bottom (the side farthest from the rectangular slot) screw, place two washers (Part#10) around the binder post.

   Place the torsional spring (Part#7) around the cylindrical extrusion of the pendulum and insert one of the tails of the spring into the receiving end of the wire.

3. **Structure Assembly**

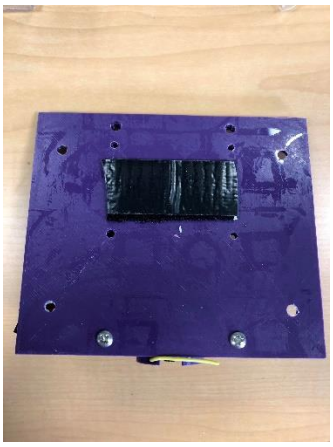
   Insert the motor (Part#1) into the hole in the motor mount (Part#12), roughly until the front face of the motor is touching the receiving end of the wire. This will require that you stretch/flex the motor mount a small amount (this is by design and ensures that the motor is tightly fixed to the mount).

   Use the large screws (Part#15) and large hex nuts (Part#16) to screw the motor mount into the base (Part#14), using the holes farthest from the rectangle of four smaller holes.



   Peel the adhesive tape off of one of the velcro strips (Part#11) and attach it to the bottom of the base (Part#14), ensuring that you do not cover any of the holes. Attach the two velcro strips together, but you

do not yet need to remove the adhesive tape on the back of the second velcro strip (you will need to do this when you eventually want to attach the base to a desk/table/surface).



## 4. Electrical Assembly

The Teensy has a default connection between the voltage provided by the USB and the 5V power line of the Teensy. However, we will be providing the 5V power directly from the PCB (Part #4 on the Parts List), and thus do not want the USB voltage and 5V to be connected (because they will not be at exactly the same voltage). The back of your Teensy should currently look like the Teensy shown in Figure 1. The connection between the USB voltage ("VUSB") and 5V is located in the center of the red circle in the figure.

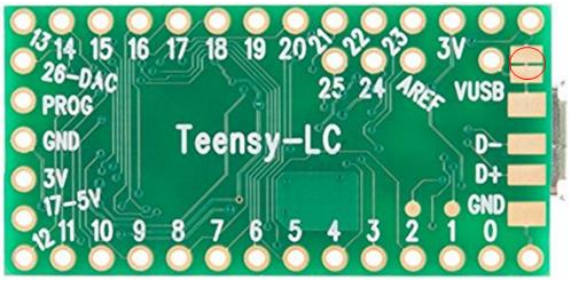



Figure 1: Original Teensy, with 5V and VUSB connection circled

Use an Exacto knife to cut this connection. Your final Teensy should look similar to that shown in Figure 2 below (noting the scratched area in the red circle).

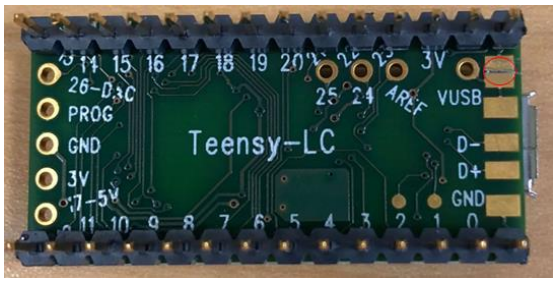



Figure 2: Teensy with VUSB and 5V connection separated

*Note: Because we are now disconnecting the computer's power from the board's, when you plug the Teensy into the computer via USB, it will neither power nor recognize the Teensy when the Teensy is not plugged in to the PCB and its power.*

Next, press the Teensy into the socket on the PCB, with the microUSB end on the side with the small semicircle symbol in the trace of the socket. Use four of each of parts 17, 18, and 22 (small screws, nuts, and spacers) to attach the PCB to the base. The header pins labeled "MOTOR_ENCODER_HEADER" should be closest to the motor. Plug the

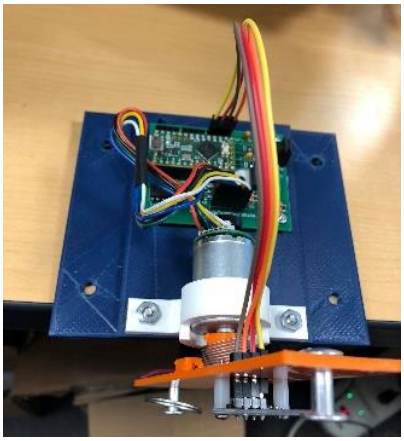
motor encoder pins into those header pins (if you are looking at the PCB with the motor encoder pins on the top of the board, from left to right the wires should be white, yellow, blue, green, red). At this point, the setup should look like the figure to the left (the white spacers, which cannot be seen here, are between the PCB and the base).

On the pendulum, use two each of parts 17, 18, and 21 to attach the IMU (part 3) to the pendulum. Plug the four wires through the rectangular slot and into the four header pins on the IMU. The pendulum should look like the picture shown below and left.

Next, plug in the other ends of the four wires into the PCB in the header pins labeled **IMU_HEADER**. ***The order that you plug these in matters and is not in the same order as on the IMU!*** **From left to right (when looking at the PCB with the text right-side-up) the wires should be in this order: GND, VCC, SCL, SDA.**

In order to attach the pendulum to the motor, press-fit the cylindrical extrusion on the back of the pendulum around the motor axle. Insert the second tail of the spring into the receiving end of the wire attached to the motor mount.

The final assembled kit should look like the picture shown below:



### 5. Testing the Kit

Download KitAssembly.ino from Canvas. Ensure that your kit is fully assembled and your pendulum is at rest (it is fine if it is at an angle due to the spring's natural position). Plug the wall adapter into your PCB and plug your Teensy into your computer and upload the code. If it fails to upload, ensure that the correct board ("Teensy LC") is listed in Tools > Board, and that there is a Port selected in Tools > Port. After it uploads, press the Serial Monitor (the magnifying glass button, or in Tools > Serial Monitor). This is outputting the readings from the gyroscope from the sensor on the pendulum (in degrees per second). Since this is the angular velocity at speed zero, this is the inherent offset value that we will want to subtract from the final angular velocity values. Approximate a very rough average of the values from the Serial Monitor. Return to the Arduino script and enter that number instead of zero into the line that says "double gyro_offset = 0 //CHANGE THIS". Press upload and open the Serial Monitor again: you should see that

the numbers are close to zero or roughly alternate between positive and negative.  If not, repeat these steps until it is close to zero or oscillates around zero.

Next, comment out the line that says “Serial.println(gz_deg_s,5)” and *un*comment the 5 lines farther below after the line that says “//UNCOMMENT THIS FOR PART 2:”. We now want to ensure that the angle reading from the motor encoder (which reads the orientation of the motor) is relatively close to the angle reading that we estimate from the angular velocity readings from the gyroscope. Upload the code and open the Serial Monitor again. You should see three columns: the first is the elapsed time in milliseconds, the second is the angle according to the encoder, and the third is the angle according to the gyroscope. Manually move the pendulum and ensure that: 1) both angle readings are close to each other (e.g. they should never be more than about 10 degrees different) and 2) the angle readings are reasonable (e.g. when you move the pendulum to horizontal, the angle is roughly +-90 degrees). Save your code.