

Instructions for setting up stepper motor:

1. Matlab setup	<p>Matlab R2011a, Data Acquisition toolbox and Instrument Control Toolbox. (Image acquisition toolbox is needed if you want to use it with the extended experiment of ping pong balls on stepper motor later).</p> <p>Do not forget to right click on Matlab and select "run it as an administrator".</p>
2. Assemble NI USB 6008 with the control box (Do not hookup to computer yet)	<p>NI USB 6008 has analog and digital channels. The stepper motor has two photocells. These are connected to the analog portion through the control box. The control box has clock and direction that are connected to the digital portion.</p> <p>You will need connection wires to connect the control box to ground and NI USB.</p> <p>USB6008 digital port 17 (P0.0) should be connected to direction control signal, port 18 (P0.1) should be connected to clock signal. USB-6008 analog port 8 (AI2) should be connected to 0 degree photocell and port 11 (AI3) should be connected to 90 degree photocell.</p>
3. NI-Software	<ol style="list-style-type: none"><li>a. Install the NI-DAQmx 9.3 version from <a href="#">this website</a> compatible with your OS. The software takes a bit long to install. Follow the directions as may pop-up during the installation. You will need to restart the computer to see the software installed. It should install NI Visa and a device monitor.</li><li>b. After you hookup the USB 6008 into the computer, the necessary driver should get installed and wait for a message to the effect "driver installed correctly" or "your hardware/usb is ready to use"</li><li>c. Check to see if your device monitor by NI or Ni Max software is able to see this device in the devices list. This means that NI can "listen" to your USB. You can use Self-Test and other tests available from Ni Max to make sure the USB has passed those tests.</li><li>d. Open a Matlab session and type <code>daqhwinfo('nidaq')</code> to see USB 6008 adapter is successfully installed. You should see the NI-US 6008</li><li>e. and a port in the output of this command.</li></ol>

<p>4. Software from NU</p>	<p>Store the files from NU in a working folder (where you will work). You may want to keep a safe copy of these elsewhere too so you can go back to the original.</p> <p>Setup_dio.m, cc.m, cw.m, onerot.m, readcell.m, move_a_while.m</p> <p>Two of these files might require changes to use the port of your USB. Specifically, in the files, setup_dio.m and readcell.m. There is a line in each of these two files that has a "dev1" or "Dev1". You/Students may need to change this to the port at which your USB is available. The port associated with your NI- USB can be found by typing <b>daqhwinfo('nidaq')</b> at the matlab command prompt. The output of this command will give you the port (such as "Dev3") at which the USB 6008 is available. Use this value in the readcell.m and setup_dio.m files. Save the changes. <i>Do not change anything else in these files.</i></p>
<p>5. Check or troubleshoot Matlab to USB 6008 connection</p>	<p>Type the following commands at the command prompt: daqhwinfo('nidaq')</p> <p>You should see NI-USB 6008 in the list. If this is not seen, type daqregister('nidaq') to make sure that Matlab is able to load the adapter.(For this you will need to run Matlab as administrator. Right click on Matlab icon and select "run as an administrator"). You may not want to give this option to students).</p> <p>On typing daqregister('nidaq') if you get the following error:</p> <pre> daqregister('nidaq') ans = Unable to load adaptor: C:\MATLABR2007b\toolbox\daq\daq\private\mw_nidaq.dll. Driver files or DLL for the specified adaptor could not be found or are not installed.</pre> <p>Then you will need to type in this command <i>again</i> daqregister('nidaq') after selecting to run it as an administrator. If the driver software has been loaded correctly, and you run as administrator, then you will get the following message:</p> <pre> mw_nidaq.dll was registered successfully.</pre>

<p>6. Setup</p>	<p>Create a dio object.  <b>dio = setup_dio;</b></p> <p>A dio object will be created and port information will be output.</p> <p>Check that photocells are working correctly:  <b>readcell(1)</b>  <b>readcell(2)</b></p> <p>The readcell output should not be negative. It will mean some problems with the photocell connections. Check the units carefully for loose wires, and control box for connections.</p> <p>Check that stepper motor can rotate:  <b>onerot</b></p> <p>The dial should turn either clockwise or counterclockwise.</p> <p>To turn counterclockwise, type (in that order):  cc;  onerot;</p> <p>To turn clockwise, type (in that order):  cw;  onerot;</p>
<p>7. Simple program</p>	<p>If your setup works correctly and the commands in #6 work correctly, you are ready for writing simple programs.</p> <p><b>Loops:</b></p> <p><u>For loops:</u>  For loops are used for repeatative tasks when you know the number of times you want to do those tasks.  For 100 loops, type:  for i= 1:100  onerot  end;</p> <p>You can change the number of loops the rotations will work.</p> <p><u>While loops:</u>  While loops are used for tasks to do until certain condition is met. You do not necessarily know how many times you will need to do the tasks until the condition is met.  While readcell(1)&lt;1.0</p>

Onerot  
End;

Use Ctrl+C to terminate the program for any reason.

Determine Step size:

Type in a large number of steps (such as 1000) and note down the total angle covered during those steps. You may have to find how many times (n) the flag on the dial crosses 0 or some particular angle and add any additional angle it may have gone.

Total angle/number of steps = step size

Experiment with different number of steps to see how many steps are needed to cover a 90 degree angle, a 180 degree angle and a 270 degree angle etc. You will need to use this with your ping-pong ball experiment.

$90/3.74 = \text{approx. } 25 \text{ steps}$

$180/3.74 \text{ approx. } 48 \text{ steps.}$

$270/3.74 \text{ approx. } 73 \text{ steps}$