**Lab 4: Using Pulse Width Modulation (Duty Cycle) to Control LED Light Level**

**Introduction to Lab 4: Presentation by Instructor**

Your instructor will introduce the hardware and software that will be used for this lab.

1. **Software Check for National Instruments myDAQ**
2. Form teams of 3 students at your recitation tables making sure that at least one team member downloaded the NI Driver file and MATLAB Update and passed the software/hardware check last week.
3. Ask your T.A. to give you one of the National Instruments myDAQs.
4. Software Re-Check: Connect the myDAQ to your computer. Type the following command in MATLAB:

>> clear all; d = daq.getDevices

If you get an output like this then you are good to go:

d =

ni: National Instruments NI myDAQ (Device ID: 'Dev1')

Analog input subsystem supports:

-2.0 to +2.0 Volts,-10 to +10 Volts ranges

Rates from 0.1 to 200000.0 scans/sec

2 channels ('ai0','ai1')

'Voltage' measurement type

**⁞**

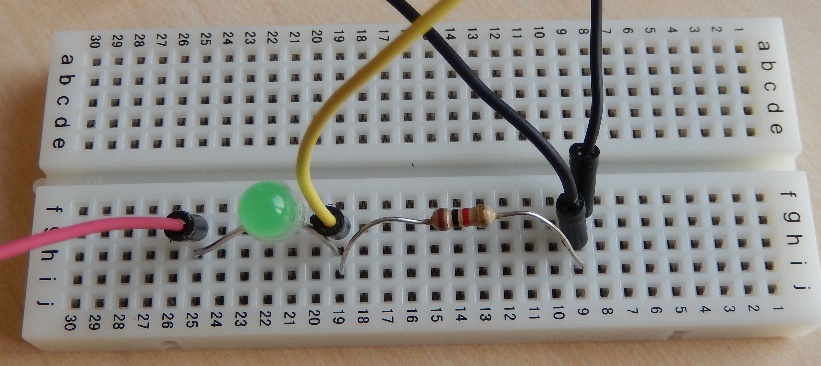
**If you get the message that No Data Acquisition Devices Are Available, then you have not installed the National Instruments Drivers file correctly.** Try reinstalling the National Instruments Drivers. Once the drivers are installed, try this MATLAB command once again:

>> clear all; d = daq.getDevices

1. **Wiring the Series Circuit**
2. Ask your T.A. for the following items:

* One breadboard
* One 1kΩ resistor (Color Code: Brown-Black-Red-Gold)
* One LED (any color)
* 1 red jumper wire
* 2 black jumper wires
* 1 yellow jumper wire

1. Insert the 1 kΩ resistor in the breadboard making sure to put each end in a separate row.
2. Insert shorter side of the LED (Cathode-) in the same row with one side of the resistor. Insert the longer side (ANODE+) into a row all by itself. ***Make sure the shorter lead of the LED (CATHODE) is connected to the resistor!***
3. Connect one end of the **RED** jumper wire to the side of the LED that is NOT connected to the resistor. Connect the other end of the RED jumper wire to the port on the NI myDAQ labeled AO0.
4. Connect one end of the **1s**t **BLACK** jumper wire to the side of the resistor NOT connected to the LED. Connect the other end of the black jumper wire to the port on the NI myDAQ labeled AGND.
5. Connect one end of the **YELLOW** jumper wire to the connection between the LED and the resistor. Connect the other end of the yellow jumper wire to the port on the NI myDAQ labeled AI0+.
6. Connect one end of the **2nd BLACK** jumper wire to the side of the resistor that is not connected to the LED. Connect the other end of the black jumper wire to the port on the NI myDAQ labeled AI0-.
7. If you have followed the directions, your breadboard should now look something like this:



1. **Send a Constant Voltage Signal and Test the Circuit**
2. The team member whose computer is being used should download the two m-files: LED\_PartC.m and LED\_PartD\_E.m from the metasite under Lab 4 and save them in his/her current folder.
3. Open the LED\_PartC file and take a quick look at the commands (**DO NOT EDIT**). These commands simply set up communication between MATLAB and the NI myDAQ.
4. Run the LED\_PartC file. After a minute or so you should get a message that the connection has been established.
5. The outputSingleScan command can be used to send a constant voltage signal to the circuit. At the COMMAND PROMPT (>>), type the following command to apply 10V to your circuit:

DAQ.outputSingleScan(10);

1. Look at your LED (from the top, not the side). It should now be ON. *If not, check all of your wiring. If that looks OK, make sure your LED isn’t backwards. Then try the command again. If it still doesn’t work, ask your TA for assistance!*
2. While watching the LED, run the following command to reduce the voltage across the circuit to 8V

DAQ.outputSingleScan(8);

**Does the LED look less bright at this voltage level?**

The LED looks just as bright at both levels.

1. While watching the LED, continue to step the voltage down by 1 V at a time and watch the effects on the light level in order to answer the following questions:

**At what voltage level does the LED start to appear less bright?**

The light appeared less bright at 4V.

**At what voltage level does the LED appear to be off?**

The Light appears to be off at 2V.

1. Close the LED\_PartC m-file.
2. **Controlling LED Light Level Using Pulse Width Modulation**
3. Open the LED\_PartD\_E m-file. **DO NOT EDIT LINES 1-25.** *These lines look much the same as the other file and simply set up communication between MATLAB and the myDAQ.*
4. Scroll down to Line 28. Read the comments in the file and fill in the required commands. **Stop adding code when you get to Line 48**. ***Don’t change any of the variable names.***
5. Save the file then run it. When prompted, enter the following values:

Frequency, f: 1

Duty Cycle, tau: 50

How Long, Tf : 5

**Observations?**

* The voltage oscillates between 0V and 10V
* The LED is flashing.

1. Run again for the following values:

Frequency, f: 10

Duty Cycle, tau: 50

How Long, Tf : 5

**Observations?**

* The LED flashes faster AKA oscillation of voltage is faster.

1. Run once more for the following values:

Frequency, f: 100

Duty Cycle, tau: 50

How Long, Tf : 5

**Observations? Explain.**

* Because the frequency is so high (means smaller period), you can no longer see the oscillation.

1. Now it’s time to vary that duty cycle. Using a frequency of 100 Hz, a duty cycle from the table, and 5 seconds for the total time, test each of the duty cycles shown in the table below and record your observations.

|  |  |
| --- | --- |
| **Duty Cycle** | **Observations?** |
| 25 | The light appears to be on for the whole time, even though it is not. It is dimmer because less light is leaving the bulb over the period. |
| 50 | The light appears to be on for the entire time as before, but it is again not. It looks brighter because more light is leaving the bulb. |
| 75 | The case from 50 to 75 is analogous to the case of 25 to 50. |
| 100 | The light appears to be on for the entire time, and it is because the duty cycle has it on for the whole interval. This is therefore also the brightest. |

**Does Pulse Width Modulation seem to be an effective way of controlling the light level?**

PWM is an effective way as it allows to control the level without a lot of hardware work-around.

1. **Measuring the Voltage Across the Resistor and Sending Data Back to MATLAB**

So far, you’ve only sent voltage out to the circuit using AO0 and AGRND (red and black wires). Now, we will use AI0+ and AI0- to measure the voltage across the resistor and send the data back to MATLAB (yellow and black wires).

Lines 53 and 56 in the LED\_PartD\_E m-file are:

DAQ.queueOutputData(AppliedVoltage');

[Vr,tmeas] = DAQ.startForeground;

The first line takes your square wave and sends it to the DAQ. The second line applies the square wave to the circuit and measures the voltage across the resistor. The voltage measurements are stored in the vector, Vr, and the times that measurements are taken are stored in the vector, tmeas.

1. Go to Line 60 in the LED\_PartD\_E m-file. Read the comments and fill in the required code to plot the measured and applied voltages.
2. Run the file using the following inputs then answer the following questions using the plot:

Frequency, f: 0.5

Duty Cycle, tau: 50

How Long? 5

**Using the data cursor tool, what is the voltage across the resistor when the Applied Voltage is 10 V?**

2.266V

**Using Ohm’s Law (V = IR) and the resistor value of 1 kOhm, what is the current flow through the circuit when the Applied Voltage is 10 V?**

I = V/R

I = 2.266/1000 = 2.266mA

**What is the voltage across the resistor when the Applied Voltage is 0 V?**

VR = 0V

**What is the current flow through the circuit when the Applied Voltage is 0 V?**

IR = 0A

1. Paste your plot and your script file commands in the spaces indicated below

**PASTE PLOT HERE: **

**PASTE SCRIPT HERE:**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lab 4: Pulse Width Modulation to Control LED Light Level

% The user will be asked to supply the desired period for the sin wave

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% house cleaning

clear; % clears out everything in workspace

close all; % closes all figures

clc; % clears out the command window

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set up communication to DAQ device

fprintf('Setting up connection to DAQ...\n');

devices\_found = daq.getDevices;

if isempty(devices\_found)

error('No devices connected: please connect a DAQ device and rerun.');

end

% Set up DAQ Connection to Device

DAQ\_Vendor = devices\_found.Vendor.ID; % determine type of device connected

DAQ\_ID = devices\_found.ID; % determine ID for connected device

DAQ = daq.createSession(DAQ\_Vendor); % create DAQ session

DAQ.addAnalogOutputChannel(DAQ\_ID,0,'Voltage'); % add output voltage channel to play waveform through filter

ch1 = DAQ.addAnalogInputChannel(DAQ\_ID,0,'Voltage'); % add input voltage channel to measure voltage

ch1.Range = [-10 10]; % set analog input range to -10 to 10V)

DAQ.Rate = 10000; % Set the DAQ sampling rate

DAQ.outputSingleScan(0); % ensure DAQ output is 0V

fprintf('Connection established...\n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Student Code Starts Here For Part D

% Complete the line of code below to ask the user to enter in the desired

% frequency of the square wave (input statement)

f = input('What frequency? ');

% Complete the line of code below to calculate the period, T, based on f:

T = 1/f;

% Write an fprintf statement to display the computed period to 2 decimal places

fprintf('The frequency is %0.2f\n',f)

% Complete the line of code below to prompt the user for a duty cycle

tau = input('What is the duty cycle? ');

% Complete the line of code to prompt the user for how long the square wave should last

Tf = input('How long should the square wave last? ');

% Complete the line of code below to create a vector of times, t.

% t should start at 0, increment by 1/DAQ.Rate, and end close to or at Tf

t = 0:1/DAQ.Rate:Tf;

% Complete the line of code below to create a square wave called AppliedVoltage

% using your vector, t, the desired frequency, f, and desired duty

% cycle, tau. Square wave should vary from 0V to 10V.

AppliedVoltage = 5\*square(2\*pi\*f\*t, tau) + 5;

% Student Code Ends Here FOR PART D

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% DO NOT EDIT LINES 49-59

% Send the voltage signal (square wave) to the NI myDAQ

AppliedVoltage(end) = 0;

DAQ.queueOutputData(AppliedVoltage');

% Make it go!!

% Collect the voltage across the resistor using s.startForeground

[Vr, tmeas] = DAQ.startForeground;

% Vr will be a vector of the measured resistor voltage

% timestamps is a vector of times at which the voltage was measured

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Student Code Starts Here For Part E

% Write a plot statement to plot two graphs together on the same plot (not subplot)

% Put tmeas on the x-axis and AppliedVoltage and Vr (resistor voltage) on

% the y-axis. Add legend, title, and axis labels.

% Add ylim([0 12]) to adjust the y-limits for a better graph

plot(tmeas,AppliedVoltage,'r-',tmeas,Vr,'b-');

title('\bfApplied Voltage and Resistor Voltage','FontSize',20);

xlabel('\bfTime (sec)','FontSize',14);

ylabel('\bfVoltage (V)','FontSize',14);

legend('Applied Voltage','Resistor Voltage');

ylim([0 12]);

% Student Code Ends Here for Part E

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

DAQ.outputSingleScan(0); % Turn LED Off

**To be turned in:**

* Everyone should submit the completed m-file, LED\_PartD\_E, and a copy of this lab report with the questions answered and the final plot. The team member who has this information should e-mail these documents to the other members of his/her team.
* To make things easier for the graders, please type the names of your team members in the submission box.