# Experiment 2: Simulink & Raspberry Pi

## Introduction

Controllers are built from mechanical and electrical building blocks. Mechanical controllers are usually implemented with shafts, gears, pulleys, lever arms etc. Electrical controllers, on the other hand, are subdivided into digital and analog controllers. Digital controllers are implemented with computers and digital circuitry, while analog controllers are implemented with operational amplifiers wired as summers, integrators and differentiators. One of the objectives of this lab is to teach the student how to use op-amps, capacitors and resistors to create these summers, integrators and differentiators.

## Lab 2: Pre-Lab

1. What is the Laplace transform of:

* An ideal integrator
* An ideal differentiator

1. Determine the equation, which relates blocks Constant (A) and Constant1 (B) in Figure 1‑1 with the output.
2. Calculate the transfer function Y/X for following models:

* The differentiator in Figure 1‑4
* The integrator in Figure 1‑6.

## Simulink Fundamentals

In this section of the lab, you will be introduced to Simulink.

### Summing Model

A summing block adds to signals together. You will be building the model shown in Figure 1‑1.

Directions:

* Simulink is graphical modeling and simulation that is included with MATLAB. So, first you need to open MATLAB by double clicking on the icon on the desktop.
* To create a new Simulink model, click on the New/Plus Sign button in the top left of the MATLAB window and select Simulink Model. You will get a blank version of the window in Figure 1‑1.
* To open the library of predefined Simulink blocks click the sixth button from the left (with four colored squares). The constants can be found in the Source category, the slider gains and sum can be found in the Math Operations category, and the display can be found in the Sinks category. Drag the blocks from the library and connect them in the configuration in Figure 1‑1.
* The values of each block can be changed by double clicking on the corresponding block. Change the range of the slider gains to a maximum of 10. See Figure 1‑2.
* Some settings of the model will also need to be changed. Click on the gear button (seventh from the left). The Configuration Parameters dialog box will open. Match the settings in the box to those in Figure 1‑3.
* Press the green play button to run the model. The display block will now show the result of the summation. Record this result in your data table in the Output Short column.
* Move each of the sliders to a new rational value and click play again. You will notice that there is rounding in the output value. Record this value in the Output Short column of your data table. Then double click on the display block, when the parameters appear change the format to long and click Ok. Record the new value in the display in the Output Long column of your data table. Repeat this step for three more combinations of values on the slider gain blocks.

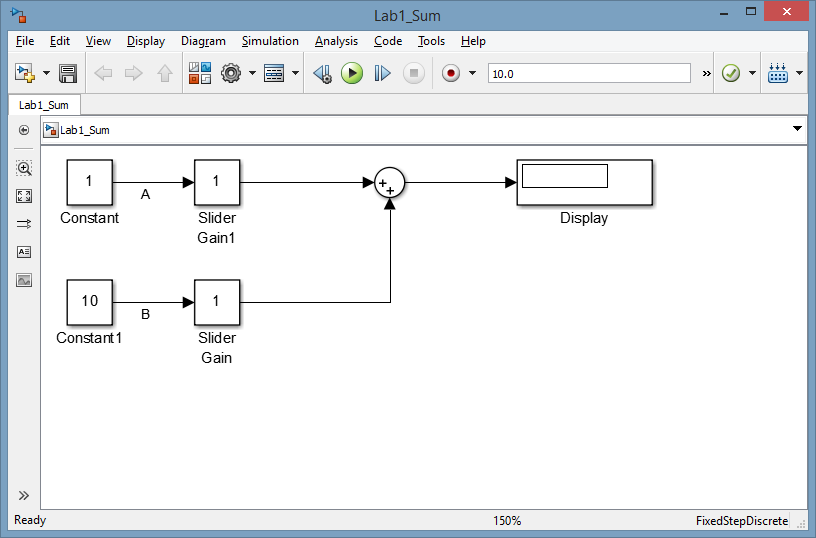


Figure 1‑1 Summing Model

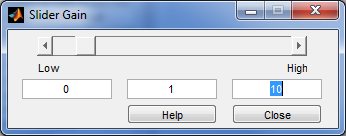


Figure 1‑2 Slider Gain

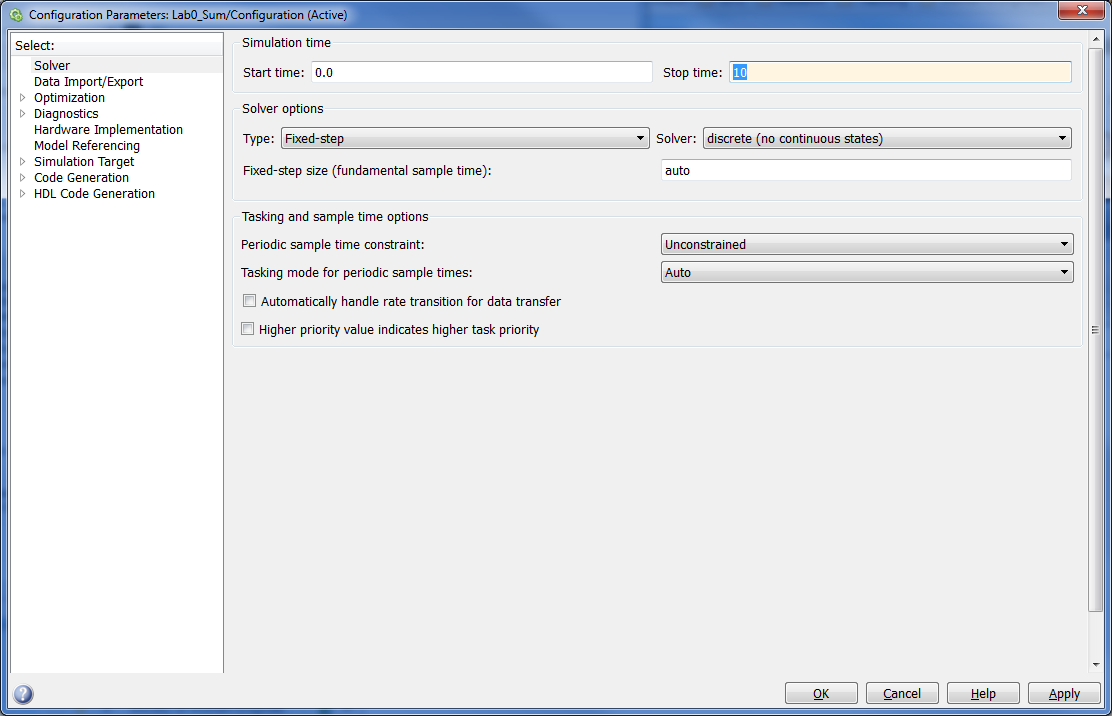


Figure 1‑3 Summing Model Configuration Parameters

### Differentiator Model

In this section you will model a differentiator as shown in Figure 1‑4.

Directions:

* Create a new Simulink model.
* Add a differentiator block to your model. It can be found in the Continuous category in the Simulink Library.
* We will use a special function generator to create our input signal. It can be found in the Lab1\_Starter.slx file in n:/labs/ge320/kit/exp1. Copy the block from this file into your new model.
* Double click on the function generator, choose the triangle wave form, then set the amplitude and frequency. Start at a frequency of 0.5 Hz and increment it in steps of 0.5 Hz, up to 2 Hz. Start with an amplitude of 1 Volt Peak-Peak and increment in steps of 0.5 Volts, up to 2.5 Volts Peak-Peak. For the last (fifth) measurement, set the frequency to 3 Hz and the amplitude to 6 Volts Peak-Peak.
* What is the transfer function of the system? What do you expect your output to be?
* The best way to 'see' the output will be using the scope. You should scope both the input and output of the differentiator using the two input channels in the scope. You can find the scope block in the Sinks category of the Simulink Library.
* Some settings of the model will also need to be changed. Click on the gear button (seventh from the left). The Configuration Parameters dialog box will open. Match the settings in the box to those in Figure 1‑5.
* Press the green play button to run the model. The display block will now show the result of the summation. Record the results in your data table.
* Since we built a differentiator, how would you expect the output amplitude to relate to the input slope? Do you notice any disagreement between the output amplitude and the input slope for a certain input? If yes, explain to your TA why.
* Take a total of 5 readings, record them on your data sheet and show the TA.
* *Change the function type to a square wave in the function generator block*. What happens? Why?

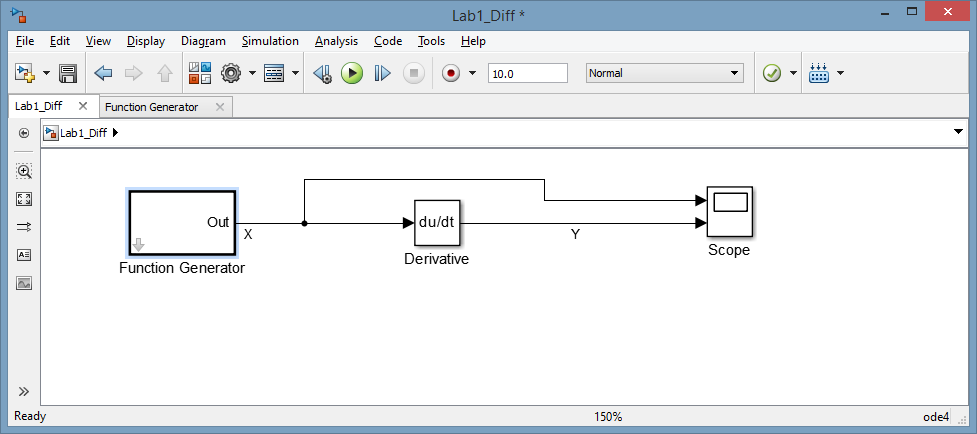


Figure 1‑4 Differentiator Model

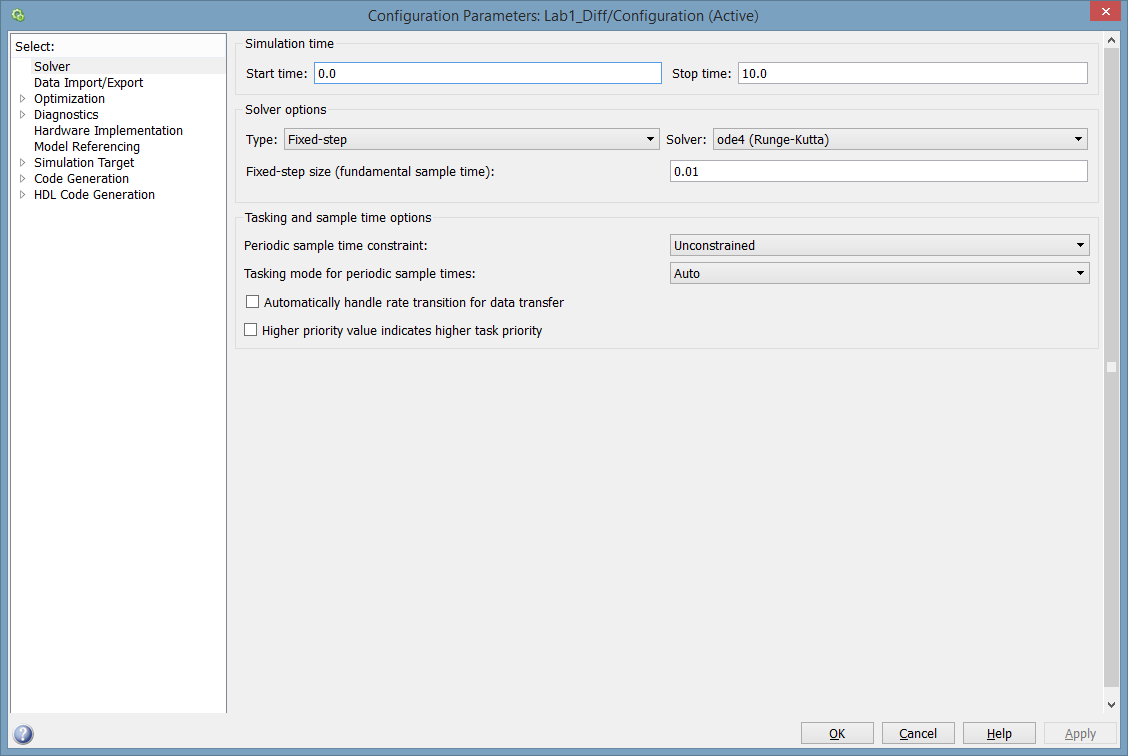


Figure 1‑5 Differentiator Model Configuration Parameters

### Integrator Model

Similar to the last two sections you will build the model as shown in Figure 1‑6.

Directions:

* Create a new Simulink model.
* Add an integrator block to your model. It can be found in the Continuous category in the Simulink Library.
* We will use the signal generator to create our input signal. It can be found in the Sources category of the Simulink Library.
* Double click on the signal generator choose a square wave form, then set the amplitude and frequency. Set the frequency to 1 Hz and keep it constant for this exercise. Start with an amplitude of 1 Volt Peak-Peak.
* What is the transfer function of the system? What do you expect your output to be?
* The best way to 'see' the output will be using the scope. You should scope both the input and output of the differentiator using the two input channels in the scope. You can find the scope block in the Sinks category of the Simulink Library.
* Some settings of the model will also need to be changed. Click on the gear button (seventh from the left). The Configuration Parameters dialog box will open. Match the settings in the box to those in Figure 1‑7.
* Press the green play button to run the model. The display block will now show the result of the summation. Record the results in your data table.
* Since we built an integrator, how would you expect the output slope to relate to the input amplitude? Do you notice any disagreement between the output amplitude and the input slope for a certain input? If yes, explain to your TA why.
* Change the amplitude of the square wave. Take a total of 5 readings, record them on your data sheet and show the TA.

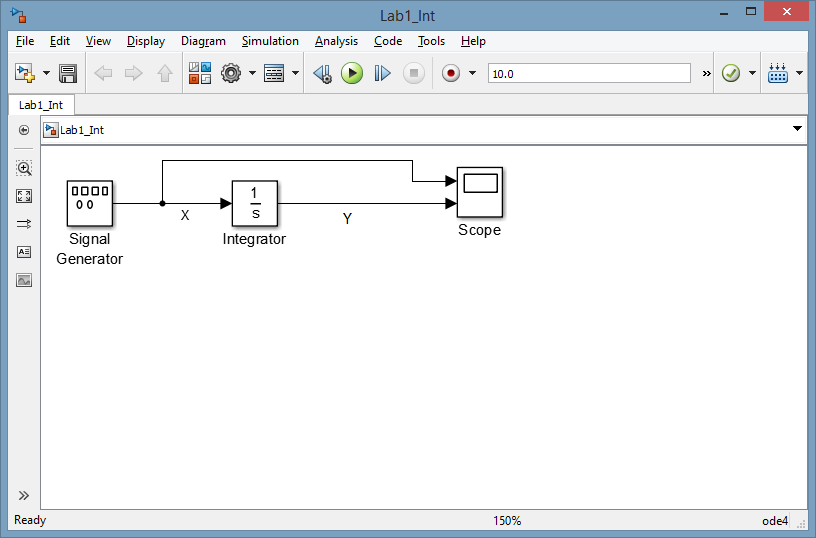


Figure 1‑6 Integrator Model

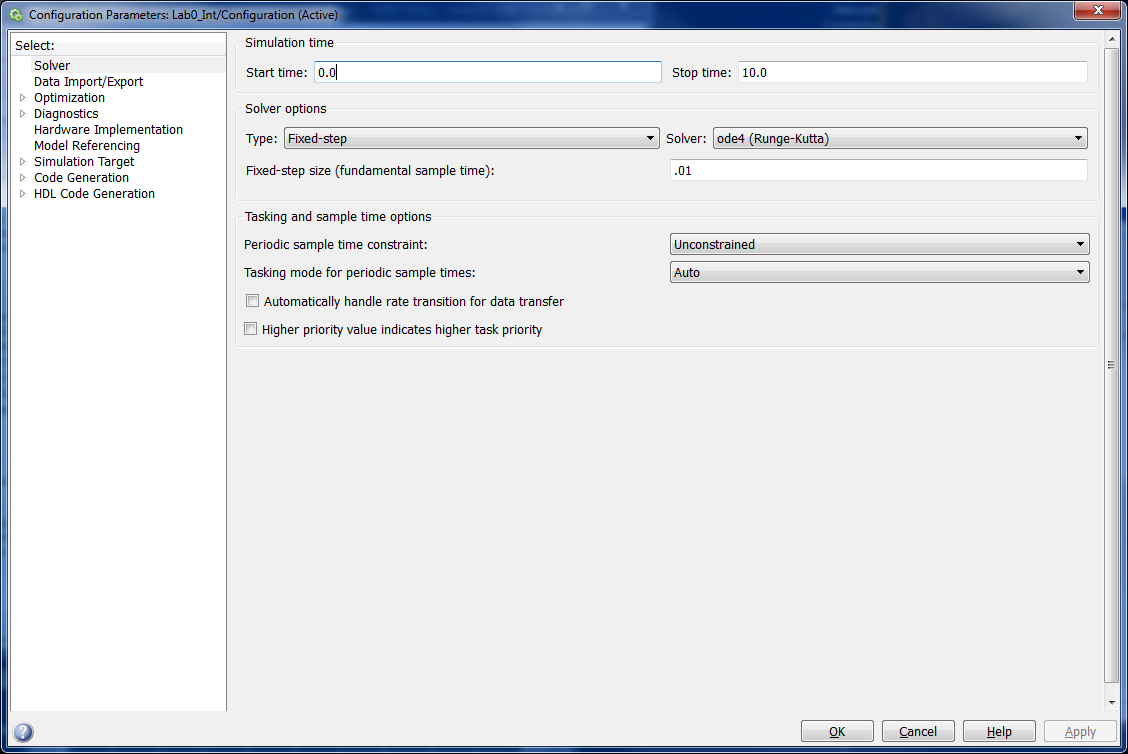


Figure 1‑7 Integrator Model Configuration Parameters

### Running a Model on the Raspberry Pi

In Lab 0 you ran a model on the Raspberry Pi that was already built. Now you will build your own model to run on the Raspberry Pi. The goal of this model will be to make the blue, green, and yellow LEDs on your circuit board blink.

Directions:

* In the MATLAB Command window, run the following command
  + mypi=raspi(‘ipaddress’,’pi’, ’raspberry’), where ipaddress refers to the actual IP address on the label affixed to the base of the kit.
* Create a new Simulink Model.
* Save the model in the same folder on the hard drive that you used in Lab 0. Name the file Lab1\_LED.
* In the Tools menu, click hover over the item Run on Target Hardware, then click Prepare to run…
* In the Target Hardware drop down menu select Raspberry Pi. Make sure your parameters match the values in Figure 1‑8, except the IP Address box should match the IP Address on the label affixed to your kit.

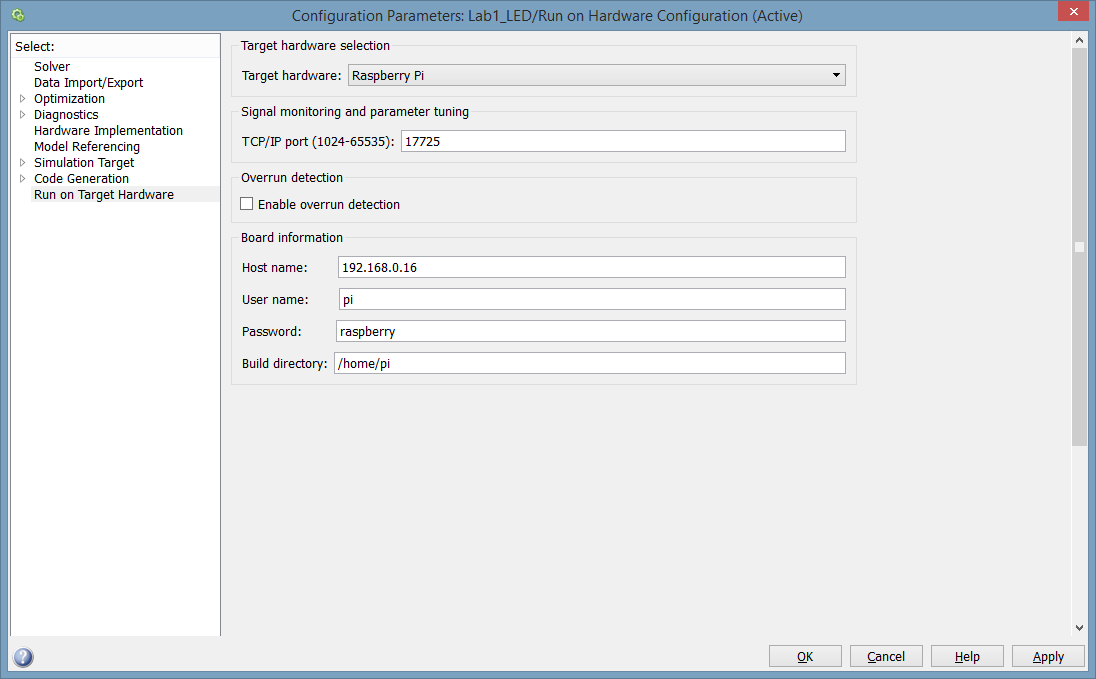


Figure 1‑8 LED Model Target Hardware Configuration Parameters

* Select the solver pane. Make sure your parameters match the values in Figure 1‑9.

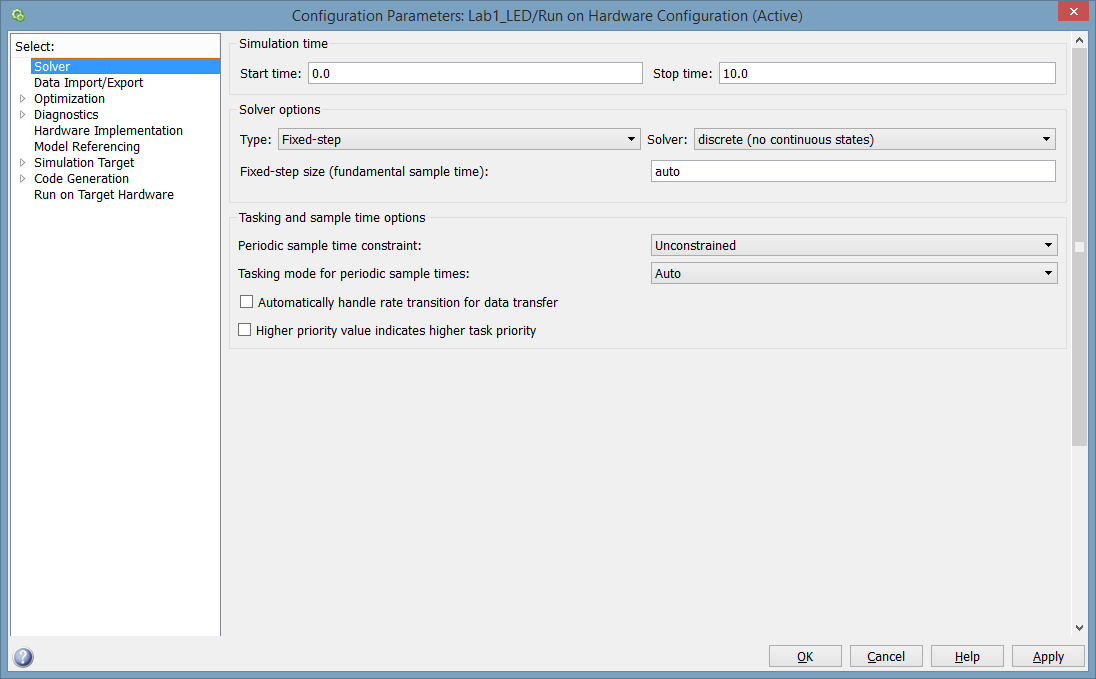


Figure 1‑9 LED Model Solver Configuration Parameters

* Click OK to close the configuration parameters.
* Next, you will create the model shown in Figure 1‑10.
* The Pulse Generator block can be found in the Sources category of the Simulink Library, Data Type Conversion blocks can be found in the Signal Attributes block and the Raspberry Pi GPIO Write blocks can be found in the Simulink Support Package for Raspberry Pi category.
* Each Pulse Generator should be set to have an amplitude and period of 1. The duty cycle of one generator should be 5%, the second one should be 50% and the last one should be 90%.
* For each GPIO Write block set the board to Raspberry Pi Model B Rev 2. The GPIO Number should be set to 23, 24, and 25 respectively.
* In the box next to the play and stop buttons set the simulation time to 30 seconds. In the drop down menu next to the time, set the simulation mode to External.
* Press the green play button to send the model to the Raspberry Pi to run.
* Observe the behavior of the LEDs on the breadboard of the kit.

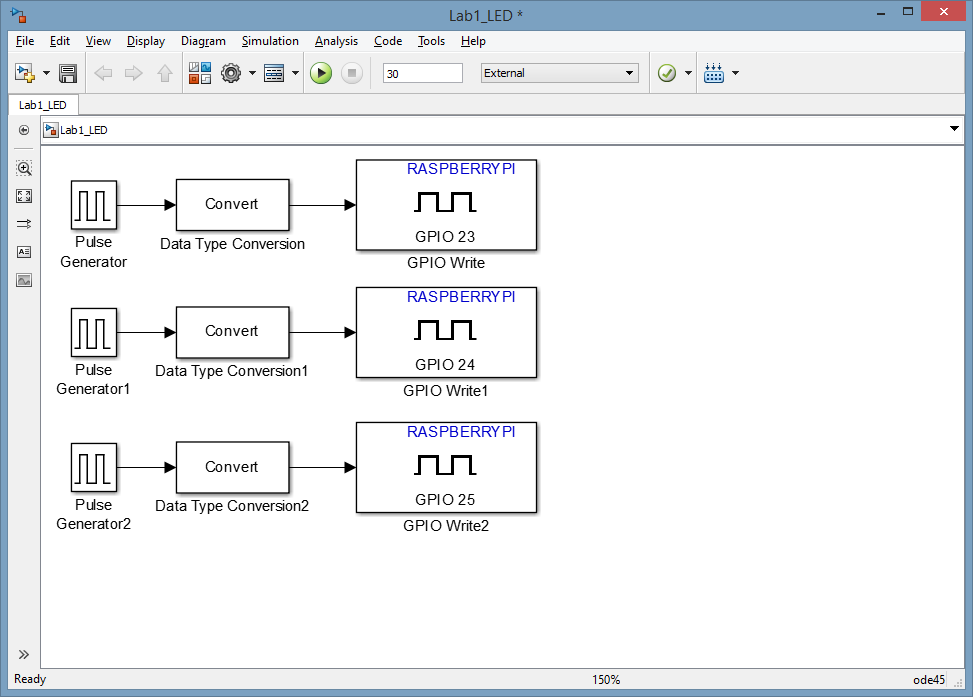


Figure 1‑10 LED Model

## Lab 2: Post-Lab

Include the answers to the following questions in your lab report.

1. What is the difference between the long and short number displays?
2. Do you think we modeled an ideal differentiator?
3. What happened when you input a square wave (instead of a triangular one) to the differentiator? Why?
4. What difference did the duty cycle make when the LEDs were hooked up to the Raspberry Pi outputs?

# Lab 2: Data Sheet

| Summing Model | | | |  | **Integrator**  **(Square Wave Input)** | |
| --- | --- | --- | --- | --- | --- | --- |
| **Input #1** | **Input #2** | **Output**  **Short** | **Output Long** |  | **Input**  **Amplitude** | **Output**  **Slope** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

| Differentiator (Triangular Wave Input) | | | |
| --- | --- | --- | --- |
| **Measured Input Frequency** | **Measured Input Voltage Peak-Peak** | **Input**  **Slope** | **Output**  **Amplitude** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |