Lab 3b: Blinking the LED by MingSeg

650:361 Introduction to Mechatronics

Team Members: Shivani Topiwala, Nancy Contreras, and Pamela Pajarillo

Questions:

1. How can you determine how fast you can run code in External mode?

In external mode, you can use the running clock in the Simulink diagram to see how fast it is processed. The trick about external mode is that you have to compare the running Simulink clock with a real world clock. If both clocks keep up with each other, the code is running in real time. If the Simulink clock is slower than the real world clock, that means that the program can not run the code in the allocated time.

1. How can you determine how fast your code can run in Normal mode?

In normal mode, you can use the running clock in the Simulink diagram to see how fast it is processed.

1. What is the difference between Normal mode and External mode?

External mode “allows bi-directional communication to and from the application board to the PC”. When using external mode, you can change the parameters of the code, which can not be done in normal mode. Normal mode can be used by selecting the “Deploy to Hardware” button. This mode just downloads your code and runs it. Normal mode does not interact with your code, so the memory it requires is less.

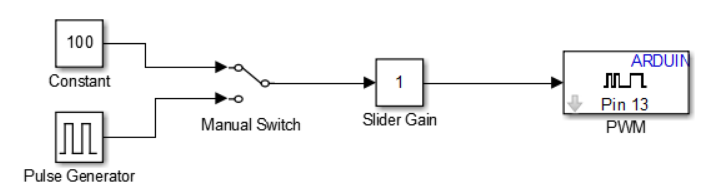
**Part 1)**

For this lab, we had to install certain software in order to use the Arduino. We first had to download the Rensselaer Arduino Support Package into Matlab. Next, we had to install the Arduino Simulink Support Package in order to use Simulink to control the Arduino.

**Part 2)**

For this section, our objective is to used Simulink to turn a LED light on the Arduino. In order to do this, we had to create a Simulink diagram (see Fig. 1.) with a PWM, slider gain, manual switch, constant, and pulse generator blocks.

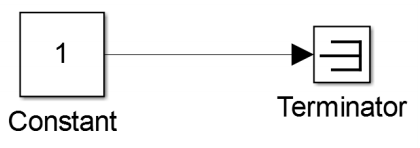
We first used External Mode to control the LED on the Arduino. The external mode allows us to change parameters while the system is running.

**Fig.1. Simulink Diagram using external mode by changing the brightness of an LED**

While the code was running, we observed the effects on the LED when changing the value of the Slider Gain, Manual Switch, and the Pulse Generator. When we turned the switch to constant, the light did not blink, but when we turned the switch to Pulse Generator, the LED started blinking. When we clicked on the Slider Gain while the code was running, the LED’s brightness changed. We also realized that when the code ran faster if it was deployed to hardware than external mode.

**Part 3)**

For this part, we evaluated the code execution speed. To determine how fast the code can be run on the target hardware, we used the “Enable overrun detection” in Simulink. We first built the following model:



While this code essentially does nothing, we use it to see how fast the hardware can execute this simple loop. We check the box “Enable overrun detection” and set the pin to 13. Then we go to “Run on Target Hardware” and click “Prepare to Run.” We use a sample time of 0.01 milliseconds. If the code cannot be executed in the time we specified, the LED on pin 13 will light up. The LED did end up lighting up which meant the microprocessor could not run the code every 0.00001 seconds. Therefore, we increased the sample time to 0.02 milliseconds and continued until the LED does not light up. We also built this code and tried to determine how fast that code ran:

We also played with different sample times to see how fast it would run. This concludes part 3. 