EECE 344 Digital Systems Design

Lab 3: SysTick for the Countdown Timer and PWM

1 Pre-Lab

1. Review the lecture notes about PLL, SysTick, and the DC motor.
2. Read the textbook Section 4.3 PLL, Section 4.4 SysTick, and Section 8.7 PWM.

2 Objective

The lab is intended to:

1. Learn the typical code structure for device driver in embedded C.
2. Learn and apply the flow control structures in C.
3. Build the interfacing circuit for the GPIO port with a DC motor.
4. Learn to configure the MCU's frequency by PLL.
5. Learn and apply **precise timing by SysTick**.

3 Requirements

Required parts: the Tiva LaunchPad for TM4C123, a 7-segment display, jumpers, ADALM 2000 (as the +5V power source and oscilloscope), a transistor, a diode, and a breadboard.

Work in pairs for this lab assignment.

4 System Requirements

Many chemical and manufacturing processes require timed or delayed steps in actuators. This lab tries to use a timer to control the delayed start of a motor. This section explains the system's behavior requirements. The system has two major external peripherals: a 7-segment display and a DC motor. The 7-segment display is to show the time information and the DC motor is the actuator. After the reset (start) of the system, the system will run a 10-second delay, shown on the 7-segment display. After the delay is passed, the DC motor will start to run.

4.1 Countdown Timer by PLL and SysTick

In this part, we will interface a 7-segment display to the Launchpad. The display is to show the time information. When your system works, it displays from 9, 8, …, down to 0, indicating that 10 seconds has passed. Therefore, each digit shows for 1 second.

Since we only use one 7-segment display in this timer, we do not use a transistor on the common cathode pin. You just need to directly ground the common cathode (pin 3). Use Port B for the interface with the 7-segment display, as explained in the lecture. You may proceed as follows.

1. Use the template provided on Canvas to start the design of your code. Start with the low-level "device drivers", the functions that configure the hardware and provide APIs to your main function. You may proceed with the following:
   (1) `int PLL_Init(void);` // Use PLL to configure the frequency to be **50 MHz**.

(2) `void SysTick_Wait(uint32_t delay);` // A utility function for SysTick to count down.

(3) `void SysTick_Wait200ms(uint32_t delay);` // This function will create multiples of 200 ms delay. The multiplication is specified by the parameter "delay". Therefore, if delay = 10, calling this function will create a delay of 200 ms * 10 = 2000 ms. Internally, this function calls `SysTick_Wait(uint32_t delay)` to assign SysTick with the appropriate counting down value to achieve the unit delay of 200 ms.

(4) In the `main` function, initialize the `unsigned char digitPattern[]` array, for use for the 7-segment display showing the count down timer information.

(5) In the `main` function, finish the initialization needed for the GPIO ports and pins.

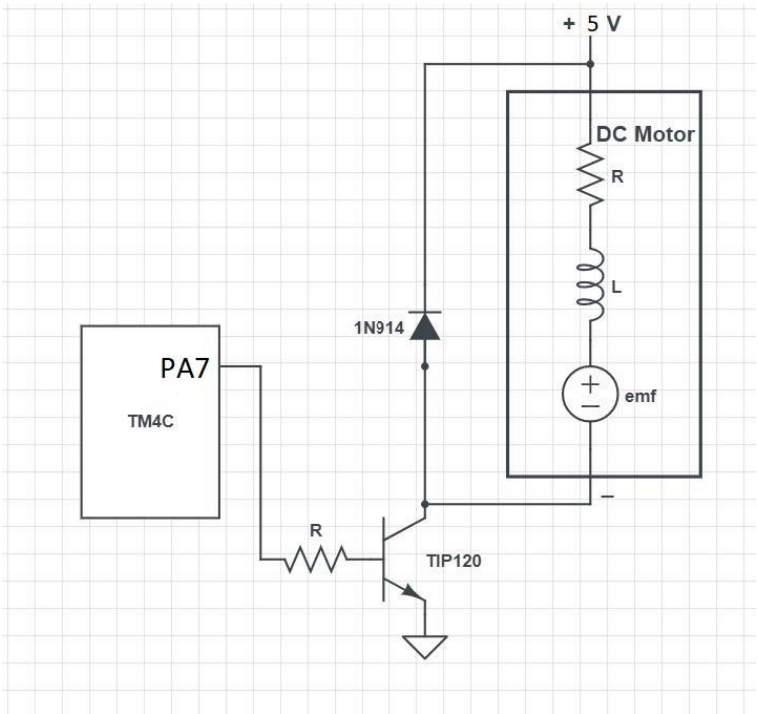
(6) In the `main` function, design and finish the code for showing the countdown time information on the 7-segment display. It should show each digit for **precisely** 1 second, from 9 down to 0.

2. Compile and debug your code to make it error-free. You may use the ADALM 2000 to help your debugging. Make sure your code has the correct logic behaviors.
3. Build your circuit, connecting the pins of the 7-segment and the MCU. Make sure they have the common ground.
4. Test the performance of your system.

Please keep the circuit and code for the next part. We will use it and add a DC motor to your system.

4.2 Interfacing with the DC Motor

Now let us add a DC motor to your circuit. After the countdown timer finishes, the DC motor will start to run. Interface the motor to the Lauchpad on Port A, as shown in the following figure.



The speed of the DC motor is controlled by an electronic circuit interface shown in the above figure with the GPIO PA7 generating a controlling signal. The controlling method is called "Pulse Width Modulation" or PWM (Section 8.7 of the textbook). The PWM is characterized by its period and the duty cycle.

In this lab, the period is 1 ms and the duty cycle (the time percentage that the signal should be logic High) is 80%. That is, the signal will be logic High for 80% of the period and logic Low for the remaining 20% period. Higher duty cycle will cause the motor to spin faster while lower duty cycle will decrease the motor's spinning speed. You may proceed with the following steps:

1. In the `main` function, after the countdown timer code:
   (1) Design code to output through PA for the DC motor;
   (2) Use the SysTick to achieve the desired 80% duty cycle within the 1 ms period.
2. Compile and debug your code to make it error-free. You may use the ADALM2000 to help your decide if you achieve the correct 80% duty cycle. Make sure your code has the correct logic behaviors.
3. Test the performance of your whole system.

5 Grading

The grading of your lab comprises the following components. Refer to the rubrics about the ranking of each component.

5.1 Performance of the Lab

Show the performance of your built system. Explain the interfacing circuit to your instructor. **Prove to your instructor that your duty cycle is 80%**.

5.2 Demonstration

When demonstrating the program, each student in the team will be asked a different question to demonstrate his/her understanding of the project at hand. You are expected to explain the design and implementation of your code if asked.

5.3 Deliverables

Please indicate who your lab partner is in your deliverables. Submit your deliverables on Canvas to “Lab 3: SysTick and PWM” under “Assignments” by the due time and date. Refer to the schedule shown on Canvas. For this lab, your deliverables are:

1. Provide a flowchart of the program you designed. Use software to create the chart, as any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.
2. Your interfacing circuit drawn by a professional software, such as PCB Artist (available at https://www.4pcb.com/free-pcb-design-software.html). Any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.

5.4 Code

Submit your code on Canvas to “Lab 3: SysTick and PWM” under “Assignments” by the due time and date. Your project folder should contain all C code, compressed into a zip file. Include comments and indentation in your code. Points will be deducted for sloppy code lacking documentation.

5.5 Lab Report

You (as an individual student, not a group) may choose to write a lab report for this lab if you have not fulfilled the requirement of writing three lab reports. If you choose to write a report for Lab 3, submit your report on Canvas “Assignments” by the due time and date. Follow the instructions in the section on "Lab Report Writing Guidelines" in “EECE 344 Digital Systems Design Lab Policy.” In the body of your report, make sure you include the following:

1. A flowchart of the program you designed. You may use the flowchart you prepared for Deliverables (5.3 above). Use software to create the chart, as any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.
2. Your interfacing circuit drawn by professional software. You may use the flowchart you prepared for Deliverables (5.3 above). Any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.
3. A photo of the circuit on the breadboard for the system.

Technical writing explaining your design and the flowcharts of your code. Describe any challenges in your experiment and how you solved the problems. Draw a conclusion about programming the digital I/O with the Launchpad.