ELEC 3300

LAB 4: TIMER AND PWM FUNCTIONS

A. OBJECTIVE:

1. To familiarize yourself with the MINI-V3 Development Board.
2. To understand the Timer function of STM32.
3. To understand the control of the servo motor

B. PRE-LAB ASSIGNMENT:

1. Study the information about MINI-V3 Development Board from the course website.
2. Study the information about Fire Debugger from the course website.
3. Study the tutorial information related to LAB4.
4. Study the SG90 Servo Motor Datasheet from the course website.

C. LAB SETUP DETAILS

1. Download the LAB4.zip from the course website and unzip it.
2. Open Keil. Go Project → Open Project... Navigate to the project file for this lab, the project file should be under …/LAB4/Project/RVMDK/LAB4.uvprojx.
3. Connect the Fire Debugger according to the information about Fire debugger. Make sure that the Green LED of the Fire Debugger is ON.

D. EXPERIMENT

In this LAB, there are 6 tasks.

Task 1 – Output SYSCLK via MCO.
Task 2 – Generate a student ID dependent PWM at TIM3_CH1, using SYSCLK as input source.
Task 3 – Generate a student ID dependent PWM at TIM4_CH1, using TIM3 as input source.
Task 4 – You need to change the optimization to see if you output a different wave.
Task 5 – Generate a 700Hz sound by modifying Task 3.
Task 6 – Control a SG90 servo motor by K1 and K2.

E. PROCEDURES

Task 1 – You need to complete the code under the procedure MCO_CONFIG to output the SYSCLK via MCO. Please refer to the Tutorial 4 for the details.

```
/*
 *  Task 1 – Output SYSCLK via MCO.
 */
```

Open system_stm32f10x.c change the clock from 72 MHz to 24 MHz, i.e.

```
#define SYSCLK_FREQ_24MHz  24000000        //uncomment this line
/* #define SYSCLK_FREQ_36MHz  36000000 */
/* #define SYSCLK_FREQ_48MHz  48000000 */
/* #define SYSCLK_FREQ_56MHz  56000000 */
//#define SYSCLK_FREQ_72MHz  72000000      //Comment this line
```

After that, run your program, you will be able to see a 24 MHz square wave on the CRO. You can change the system_stm32f10x.c to see if you can get the wave form of 36/48 MHz.

Checkpoint 1 : Show your TA that you can output the SYSCLK

Task 2 – Generate a student ID dependent PWM at TIM3_CH1, using SYSCLK as input source. Refer to the Tutorial 4 for the relation of the waveform to your student ID.

Before running your program, MAKE SURE you COMMENTED Task 1 and you need to change the system_stm32f10x.c back to 72MHz.

Please refer to Tutorial 4, connect the CH1 of your CRO to the PA.6. After that, run your program; check if you got the desired wave on the CRO.

Checkpoint 2 : Show your TA that you finished Task 2
Please clearly show to the TA the duty cycle and the frequency of your signal.

Task 3 – Generate a student ID dependent PWM at TIM4_CH1, using TIM3 as input source. Refer to the Tutorial 4 for the relation of the waveform to your student ID.

Please refer to Tutorial 4, connect the CH1 of your CRO to the PB.6. After that, run your program; check if you got the desired wave on the CRO.

Checkpoint 3 : Show your TA that you finished Task 3
Please clearly show to the TA the duty cycle and the frequency of your signal.

Task 4 – You need to change the optimization to see if you output a different wave. Your C++ Optimization should already being set to Level 0 (-O0) when you are doing Task 2 and Task 3.

Try to change your Optimization to Level 3 (-O3).

Compile your program and measure the waveform again, check the frequency of your waveform. Did the frequency of the waveform change?

Checkpoint 4 : Show your TA that you finished Task 4
Please clearly show to the TA you modified the optimization.

Task 5 – Generate a 700Hz sound by modifying Task 3.
Connect the output of PB.6 to onboard Buzzer.

Checkpoint 5 : Show your TA that you finished Task 5

Task 6 – Control a SG90 servo motor by K1 and K2.
Refer to Tutorial 4, and combining with your knowledge of LAB2, write a program to perform the following task.

At start, servo will stay at the middle
If K1 is pressed, servo turns to left by 30 degrees from the middle,
    when K1 is released it will stay at that position
If K2 is pressed, servo turns to right by 30 degrees from the middle,
    when K2 is released it will stay at that position
If both K1 AND K2 are pressed together, servo will stay at the middle.
    when both K1 and K2 are released it will still stay at middle

Checkpoint 6 : Show your TA that you finished Task 6