**Date Submitted: 10/9/2018 4:36 PM**

**Task 00: Execute provided code**

**Youtube Link: No Submission required for Task 00**

**------------------------------------------------------------------------------------**

**Task 01:** Change the PWM duty cycle to make the servo motor to do a loop of a complete sweep from 0 to 180 deg

Youtube Link: <https://www.youtube.com/watch?v=UcjvO7TYA0c>

This program is responsible for changing the PWM duty cycle to be able to perform a complete sweep rotating the Servo Motor from 0 to 180 degrees from its position.

**Modified Code:**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

**#include** "driverlib/debug.h"

**#include** "driverlib/pwm.h"

**#include** "driverlib/pin\_map.h"

**#include** "inc/hw\_gpio.h"

**#include** "driverlib/rom.h"

**#define** PWM\_FREQUENCY 50 //set frequency to 50 for 20ms period

**int** **main**(**void**)

{

**volatile** uint32\_t ui32Load;

**volatile** uint32\_t ui32PWMClock;

**volatile** uint8\_t ui8Adjust;

ui8Adjust = 75; //20us \* 75 = 1.5ms

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_5|SYSCTL\_USE\_PLL|SYSCTL\_OSC\_MAIN|SYSCTL\_XTAL\_16MHZ);

ROM\_SysCtlPWMClockSet(SYSCTL\_PWMDIV\_64);

ROM\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_PWM1);

ROM\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOD);

ROM\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

ROM\_GPIOPinTypePWM(GPIO\_PORTD\_BASE, GPIO\_PIN\_0);

ROM\_GPIOPinConfigure(GPIO\_PD0\_M1PWM0);

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = GPIO\_LOCK\_KEY; //unlock PortF

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_CR) |= 0x01;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = 0;

ROM\_GPIODirModeSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_4|GPIO\_PIN\_0, GPIO\_DIR\_MODE\_IN);

ROM\_GPIOPadConfigSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_4|GPIO\_PIN\_0, GPIO\_STRENGTH\_2MA, GPIO\_PIN\_TYPE\_STD\_WPU);

ui32PWMClock = **SysCtlClockGet**() / 64;

ui32Load = (ui32PWMClock / PWM\_FREQUENCY) - 1;

**PWMGenConfigure**(PWM1\_BASE, PWM\_GEN\_0, PWM\_GEN\_MODE\_DOWN);

**PWMGenPeriodSet**(PWM1\_BASE, PWM\_GEN\_0, ui32Load);

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_0, ui8Adjust \* ui32Load / 1000);

ROM\_PWMOutputState(PWM1\_BASE, PWM\_OUT\_0\_BIT, true);

ROM\_PWMGenEnable(PWM1\_BASE, PWM\_GEN\_0);

**while**(1)

{

**if**(ROM\_GPIOPinRead(GPIO\_PORTF\_BASE,GPIO\_PIN\_4)==0x00) //if SW1 is pressed

{

ui8Adjust = ui8Adjust + 5; //increment by 0.1ms ex. 20us \* 80 = 1.6ms

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_0, ui8Adjust \* ui32Load / 1000);

}

ROM\_SysCtlDelay(100000); //delay

}

}

**------------------------------------------------------------------------------------**

**Task 02:** Change PWM duty cycle from 10% to 90% to control the brightness of the LED at PF1.

Youtube Link: <https://www.youtube.com/watch?v=AMl04BY0srY>

This program is responsible for implementing a PWM cycle onto the onboard LED of the TIVA C board. Using PWM we can see the LED begin at 90% brightness and then decrease in brightness until it reaches 10%. Once it reaches 10% it will increase back to 90%. The LED that is chosen is using PF1 (Red LED).

**Modified Code:**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

**#include** "driverlib/debug.h"

**#include** "driverlib/pwm.h"

**#include** "driverlib/pin\_map.h"

**#include** "inc/hw\_gpio.h"

**#include** "driverlib/rom.h"

**int** **main**(**void**)

{

**int** i; //for loop temp variable

**SysCtlPWMClockSet**(SYSCTL\_PWMDIV\_1);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF); //enable GPIOF

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_PWM1); //enable PWM1

**GPIOPinConfigure**(GPIO\_PF1\_M1PWM5); //assign pin to PWM

**GPIOPinTypePWM**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3); //set pins as output

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = GPIO\_LOCK\_KEY; //unlock PortF

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_CR) |= 0x01;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = 0;

**PWMGenConfigure**(PWM1\_BASE, PWM\_GEN\_2, PWM\_GEN\_MODE\_DOWN);

**PWMGenConfigure**(PWM1\_BASE, PWM\_GEN\_3, PWM\_GEN\_MODE\_DOWN);

**PWMGenPeriodSet**(PWM1\_BASE, PWM\_GEN\_2, 400); //set period to 400

**PWMGenPeriodSet**(PWM1\_BASE, PWM\_GEN\_3, 400);

**PWMPulseWidthSet**(PWM1\_BASE, PWM\_OUT\_5, 300);

**PWMGenEnable**(PWM1\_BASE, PWM\_GEN\_2);

**PWMGenEnable**(PWM1\_BASE, PWM\_GEN\_3);

**PWMOutputState**(PWM1\_BASE, (PWM\_OUT\_5\_BIT | PWM\_OUT\_6\_BIT | PWM\_OUT\_7\_BIT), true);

//output PWM pins

**while**(1)

{

**for**(i = 40; i < 360 ; i++){ //increase from 10% (40) duty cycle to 90% (360)

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, i); //PF1

**SysCtlDelay**(100000);

}

**for**(i = 360; i > 40; i-- ){ //decrease from 90% (360) duty cycle to 10% (40)

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, i); //PF1

**SysCtlDelay**(100000);

}

}

}

**------------------------------------------------------------------------------------**

**Task 03:** Change PWM duty cycle from 90% to 10% to control the brightness of the all three LED at PF1, PF2, and PF3 using three nested “for loops”

Youtube Link: <https://www.youtube.com/watch?v=tH9wMp3tYMs>

This program is responsible for implementing PWM onto all three LEDs on the TIVA C board. To do this, nested loops will be implemented to be able to simultaneously change each LED. Each LED will change from 90% to 10% duty cycle. This will cause all three LEDs to start in a bright white and then end in a dull white light.

**Modified Code:**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

**#include** "driverlib/debug.h"

**#include** "driverlib/pwm.h"

**#include** "driverlib/pin\_map.h"

**#include** "inc/hw\_gpio.h"

**#include** "driverlib/rom.h"

**int** **main**(**void**)

{

**int** i,j,k; //set temporary values for nested for loops

**SysCtlPWMClockSet**(SYSCTL\_PWMDIV\_1);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF); //enable GPIOF

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_PWM1); //enable PWM1

**GPIOPinConfigure**(GPIO\_PF1\_M1PWM5); //assign PWM pins to Output LED pins

**GPIOPinConfigure**(GPIO\_PF2\_M1PWM6);

**GPIOPinConfigure**(GPIO\_PF3\_M1PWM7);

**GPIOPinTypePWM**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3); //assign pins to PWM

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = GPIO\_LOCK\_KEY; //unlcok PortF

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_CR) |= 0x01;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = 0;

**PWMGenConfigure**(PWM1\_BASE, PWM\_GEN\_2, PWM\_GEN\_MODE\_DOWN);

**PWMGenConfigure**(PWM1\_BASE, PWM\_GEN\_3, PWM\_GEN\_MODE\_DOWN);

**PWMGenPeriodSet**(PWM1\_BASE, PWM\_GEN\_2, 400); //set period to 400

**PWMGenPeriodSet**(PWM1\_BASE, PWM\_GEN\_3, 400);

**PWMPulseWidthSet**(PWM1\_BASE, PWM\_OUT\_5, 400); //initialize all LED output pins

**PWMPulseWidthSet**(PWM1\_BASE, PWM\_OUT\_6, 400);

**PWMPulseWidthSet**(PWM1\_BASE, PWM\_OUT\_7, 400);

**PWMGenEnable**(PWM1\_BASE, PWM\_GEN\_2);

**PWMGenEnable**(PWM1\_BASE, PWM\_GEN\_3);

**PWMOutputState**(PWM1\_BASE, (PWM\_OUT\_5\_BIT | PWM\_OUT\_6\_BIT | PWM\_OUT\_7\_BIT), true);

**while**(1)

{

**for**(i = 360; i > 40; i-- ){ //90 to 10%

**for**(j = 360; j > 40; j--){ //90% to 10%

**for**(k = 360; k > 40; k--){ //90%-10% nested for loops

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, i); //red

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_6, j); //blue

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_7, k); //green

**SysCtlDelay**(100000);

}

}

}

}

}