**Date Submitted: 10-10-2018**

**Task 00: Execute provided code (No submission required)**

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

**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://youtu.be/PCyn2lS2n_Q>

**Modified Code:**

//task01: Change the PWM duty cycle to make the servo motor to do a loop of a //complete sweep from 0 to 180 deg.

#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 55

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

**{**

volatile uint32\_t ui32Load**;**

volatile uint32\_t ui32PWMClock**;**

volatile uint8\_t ui8Adjust**;**

ui8Adjust **=** 83**;**

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**;**

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**)**

**{**

//range of servo: 165 degrees. Min: 18.2e-6\*ui8Adjust=0.8ms

//Max: 18.2e-6\*ui8Adjust=2.2ms

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

**{**

ui8Adjust**--;**

**if** **(**ui8Adjust **<** 44**)**

**{**

ui8Adjust **=** 44**;**

**}**

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

**}**

**if(**ROM\_GPIOPinRead**(**GPIO\_PORTF\_BASE**,**GPIO\_PIN\_0**)==**0x00**)**

**{**

ui8Adjust**++;**

**if** **(**ui8Adjust **>** 121**)**

**{**

ui8Adjust **=** 121**;**

**}**

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

**}**

ROM\_SysCtlDelay**(**10000**);**

**}**

**}**

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

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

Youtube Link: <https://youtu.be/tp4P3Sn9dDo>

**Modified Code:**

//task02: Change PWM duty cycle from 10% to 90% to control the brightness of the LED at PF1.

#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**)**

**{**

bool fadeOut**;**

int dutyCycle **=** 0**;**

SysCtlClockSet**(**

SYSCTL\_SYSDIV\_5 **|** SYSCTL\_USE\_PLL **|** SYSCTL\_XTAL\_16MHZ

**|** SYSCTL\_OSC\_MAIN**);** //40MHz clk

SysCtlPWMClockSet**(**SYSCTL\_PWMDIV\_1**);** //PWM match system clock

SysCtlPeripheralEnable**(**SYSCTL\_PERIPH\_GPIOF**);** //enable GPIO peripherals

SysCtlPeripheralEnable**(**SYSCTL\_PERIPH\_PWM1**);** //enable PWM peripheral

GPIOPinConfigure**(**GPIO\_PF1\_M1PWM5**);** //configure PF1(R) as PWM5

// GPIOPinConfigure(GPIO\_PF2\_M1PWM6); //configure PF2(B) as PWM6

// GPIOPinTypePWM(GPIO\_PORTF\_BASE, GPIO\_PIN\_2); //set PF2 pin type as PWM

GPIOPinTypePWM**(**GPIO\_PORTF\_BASE**,** GPIO\_PIN\_1**);** //set PF1 pin type as PWM

PWMGenConfigure**(**PWM1\_BASE**,** PWM\_GEN\_2**,**

PWM\_GEN\_MODE\_DOWN **|** PWM\_GEN\_MODE\_NO\_SYNC**);** //countdown non-synchronous mode

// PWMGenConfigure(PWM1\_BASE, PWM\_GEN\_3,

// PWM\_GEN\_MODE\_DOWN | PWM\_GEN\_MODE\_NO\_SYNC); //countdown non-synchronous mode

PWMGenPeriodSet**(**PWM1\_BASE**,** PWM\_GEN\_2**,** 100**);** //set period to 100

PWMPulseWidthSet**(**PWM1\_BASE**,** PWM\_OUT\_5**,** 10**);** //50% duty cycle

// PWMGenPeriodSet(PWM1\_BASE, PWM\_GEN\_3, 100); //set period to 100

// PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_6, 100); //75% duty cycle

// PWMGenEnable(PWM1\_BASE, PWM\_GEN\_3); //enable PWM generator 3 for PWM6/7

PWMGenEnable**(**PWM1\_BASE**,** PWM\_GEN\_2**);** //enable PWM generator 2 for PWM5

PWMOutputState**(**PWM1\_BASE**,** PWM\_OUT\_5\_BIT**,** true**);** //turn on LED with PWM

// PWMOutputState(PWM1\_BASE, PWM\_OUT\_6\_BIT, true); //turn on LED with PWM

**while(**1**)**

**{**

**if** **(!**fadeOut**){**

**if(**dutyCycle **<=** 90**){**

dutyCycle **=** dutyCycle **+** 5**;**

**if** **(**dutyCycle **>** 90**)**

fadeOut **=** true**;**

**}**

**}**

**else{**

dutyCycle **=** dutyCycle **-** 5**;**

**if(**dutyCycle **<=** 10**)**

fadeOut **=** false**;**

**}**

PWMPulseWidthSet**(**PWM1\_BASE**,** PWM\_OUT\_5**,** dutyCycle**);** //change duty cycle

SysCtlDelay**(**2000000**);**//delay

**}**

**}**

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

**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://youtu.be/UZs7KVPipQ8>

**Modified Code:**

//task03: 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

#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**)**

**{**

//start at 90% duty cycle

int i **=** 90 **;**

int j **=** 90 **;**

int k **=** 90 **;**

SysCtlClockSet**(**

SYSCTL\_SYSDIV\_5 **|** SYSCTL\_USE\_PLL **|** SYSCTL\_XTAL\_16MHZ

**|** SYSCTL\_OSC\_MAIN**);** //40MHz clk

SysCtlPWMClockSet**(**SYSCTL\_PWMDIV\_1**);** //PWM match system clock

SysCtlPeripheralEnable**(**SYSCTL\_PERIPH\_GPIOF**);** //enable GPIO peripherals

SysCtlPeripheralEnable**(**SYSCTL\_PERIPH\_PWM1**);** //enable PWM peripheral

GPIOPinConfigure**(**GPIO\_PF1\_M1PWM5**);** //configure PF1(R) as PWM5

GPIOPinConfigure**(**GPIO\_PF2\_M1PWM6**);** //configure PF2(B) as PWM6

GPIOPinConfigure**(**GPIO\_PF3\_M1PWM7**);** //configure PF2(G) as PWM7

//define as PWM pins

GPIOPinTypePWM**(**GPIO\_PORTF\_BASE**,** GPIO\_PIN\_1**);**

GPIOPinTypePWM**(**GPIO\_PORTF\_BASE**,** GPIO\_PIN\_2**);**

GPIOPinTypePWM**(**GPIO\_PORTF\_BASE**,** GPIO\_PIN\_3**);**

//countdown non-synchronous mode

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

**|** PWM\_GEN\_MODE\_NO\_SYNC**);**

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

**|** PWM\_GEN\_MODE\_NO\_SYNC**);**

PWMGenPeriodSet**(**PWM1\_BASE**,** PWM\_GEN\_2**,** 100**);** //set period to 100

PWMGenPeriodSet**(**PWM1\_BASE**,** PWM\_GEN\_3**,** 100**);** //set period to 100

PWMGenEnable**(**PWM1\_BASE**,** PWM\_GEN\_3**);** //enable PWM generator 3 for PWM6/7

PWMGenEnable**(**PWM1\_BASE**,** PWM\_GEN\_2**);** //enable PWM generator 2 for PWM5

PWMOutputState**(**PWM1\_BASE**,** PWM\_OUT\_5\_BIT**,** true**);** //turn on LED with PWM

PWMOutputState**(**PWM1\_BASE**,** PWM\_OUT\_6\_BIT**,** true**);** //turn on LED with PWM

PWMOutputState**(**PWM1\_BASE**,** PWM\_OUT\_7\_BIT**,** true**);** //turn on LED with PWM

**while(**1**)**

**{**

//cycle through colors and end with dim white

**for(;** i**>=**10**;** **--**i**)** **{**

**for(;** j**>=**10**;** **--**j**)** **{**

**for(;** k**>=**10**;** **--**k**)** **{**

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

SysCtlDelay**(**2000000**);**//delay

**}**

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

SysCtlDelay**(**2000000**);**//delay

**}**

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

SysCtlDelay**(**2000000**);**//delay

**}**

**}**

**}**