**10/9/2018:**

**Task 00: Execute provided code**

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

**Task 01:**

Youtube Link:

**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 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)**

**{**

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

**{**

**ui8Adjust--;**

**if (ui8Adjust < 56)**

**{**

**ui8Adjust = 56;**

**}**

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

**}**

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

**{**

**ui8Adjust++;**

**if (ui8Adjust > 111)**

**{**

**ui8Adjust = 111;**

**}**

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

**}**

**ROM\_SysCtlDelay(100000);**

**}**

**}**

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

**Task 02:**

Youtube Link: https://www.youtube.com/watch?v=zQjNKav-cpE

**Modified Schematic (if applicable):**

**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 55**

**int main(void)**

**{**

**volatile uint32\_t ui32Load;**

**volatile uint32\_t ui32PWMClock;**

**volatile uint8\_t ui8Adjust\_RED;**

**//set clock and PWM clock**

**SysCtlClockSet(SYSCTL\_SYSDIV\_5|SYSCTL\_USE\_PLL|SYSCTL\_OSC\_MAIN|SYSCTL\_XTAL\_16MHZ);**

**SysCtlPWMClockSet(SYSCTL\_PWMDIV\_64);**

**// enable PWM and Port F**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_PWM1);**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOD);**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);**

**// Enable GPIO pins**

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

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

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

**GPIOPinConfigure(GPIO\_PF1\_M1PWM5);**

**// enable switches**

**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;**

**GPIODirModeSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_4|GPIO\_PIN\_0, GPIO\_DIR\_MODE\_IN);**

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

**//calculate pwm period**

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

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

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

**PWMGenPeriodSet(PWM1\_BASE, PWM\_GEN\_2, ui32Load);**

**//set up and enable pwm**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, ui8Adjust\_RED \* ui32Load / 100);**

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

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

**while(1)**

**{**

**// adjust red persentage from 10 to 90%**

**for(ui8Adjust\_RED=10;ui8Adjust\_RED<90;ui8Adjust\_RED+=1) {**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, ui8Adjust\_RED \* ui32Load/100);**

**SysCtlDelay(100000);**

**}**

**}**

**}**

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

**Task 03:**

Youtube Link: https://www.youtube.com/watch?v=1toRstOa95I

**Modified Schematic (if applicable):**

**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 55**

**int main(void)**

**{**

**volatile uint32\_t ui32Load;**

**volatile uint32\_t ui32PWMClock;**

**volatile uint8\_t ui8Adjust\_RED;**

**volatile uint8\_t ui8Adjust\_BLUE;**

**volatile uint8\_t ui8Adjust\_GREEN;**

**// set up clock and pwm clock**

**SysCtlClockSet(SYSCTL\_SYSDIV\_5|SYSCTL\_USE\_PLL|SYSCTL\_OSC\_MAIN|SYSCTL\_XTAL\_16MHZ);**

**SysCtlPWMClockSet(SYSCTL\_PWMDIV\_64);**

**// enable pwm1 and port F**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_PWM1);**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOD);**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);**

**// Enable GPIO pins and PWMs**

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

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

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

**GPIOPinConfigure(GPIO\_PF1\_M1PWM5);**

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

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

**// enable switchs**

**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;**

**GPIODirModeSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_4|GPIO\_PIN\_0, GPIO\_DIR\_MODE\_IN);**

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

**// calculate pwm periods**

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

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

**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, ui32Load);**

**PWMGenPeriodSet(PWM1\_BASE, PWM\_GEN\_3, ui32Load);**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, ui8Adjust\_RED \* ui32Load / 100);**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_6, ui8Adjust\_RED \* ui32Load / 100);**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_7, ui8Adjust\_RED \* ui32Load / 100);**

**// enable pwms**

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

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

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

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

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

**while(1)**

**{**

**// go through combinations of colors**

**for(ui8Adjust\_RED=90;ui8Adjust\_RED>10;ui8Adjust\_RED-=10) {**

**for(ui8Adjust\_BLUE=90;ui8Adjust\_BLUE>10;ui8Adjust\_BLUE-=5) {**

**for(ui8Adjust\_GREEN=90;ui8Adjust\_GREEN>10;ui8Adjust\_GREEN-=1) {**

**//set pwm duties**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, ui8Adjust\_RED \* ui32Load/100);**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_6, ui8Adjust\_BLUE \* ui32Load/100);**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_7, ui8Adjust\_GREEN \* ui32Load/100);**

**// delay**

**SysCtlDelay(100000);**

**}**

**}**

**}**

**for(ui8Adjust\_GREEN=90;ui8Adjust\_GREEN>10;ui8Adjust\_GREEN-=10) {**

**for(ui8Adjust\_BLUE=90;ui8Adjust\_BLUE>10;ui8Adjust\_BLUE-=5) {**

**for(ui8Adjust\_RED=90;ui8Adjust\_RED>10;ui8Adjust\_RED-=1) {**

**// set pwm duties**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, ui8Adjust\_RED \* ui32Load/100);**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_6, ui8Adjust\_BLUE \* ui32Load/100);**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_7, ui8Adjust\_GREEN \* ui32Load/100);**

**// delay**

**SysCtlDelay(100000);**

**}**

**}**

**}**

**}**

**}**

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