#include <stdint.h>

#include <stdbool.h>

#include "inc/tm4c123gh6pm.h"

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "driverlib/sysctl.h"

#include "driverlib/interrupt.h"

#include "driverlib/gpio.h"

#include "driverlib/timer.h" // includes all necessary library and drivers to execute this program

int main(void)

{

uint32\_t ui32Period;

SysCtlClockSet(SYSCTL\_SYSDIV\_5|SYSCTL\_USE\_PLL|SYSCTL\_XTAL\_16MHZ|

SYSCTL\_OSC\_MAIN); //enabling system clock

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

GPIOPinTypeGPIOOutput(GPIO\_PORTF\_BASE, GPIO\_PIN\_3|GPIO\_PIN\_2|

GPIO\_PIN\_3); //enabling GPIO Pin 3 as Output

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER0);

TimerConfigure(TIMER0\_BASE, TIMER\_CFG\_PERIODIC);

ui32Period = (SysCtlClockGet() / 10) / .5; //configuring the period

TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32Period -1);

IntEnable(INT\_TIMER0A);

TimerIntEnable(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

IntMasterEnable();

TimerEnable(TIMER0\_BASE, TIMER\_A);

while(1)

{

}

}

void TIMER0A\_Handler(void) //Interrupt Service Routine

{

// Clear the timer interrupt

TimerIntClear(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

// Read the current state of the GPIO pin and+

// write back the opposite state

if(GPIOPinRead(GPIO\_PORTF\_BASE, GPIO\_PIN\_3))

{

GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 0); //writing 0 to GPIO PF3

}

else

{

GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_3, 8); //writing 8 in hex to the GPIO PF3

}

}