```
// SysTickInts.c
// Runs on LM3S1968
// Use the SysTick timer to request interrupts at a particular period.
// Daniel Valvano
// June 27, 2011

/* This example accompanies the book
   "Embedded Systems: Real Time Interfacing to the Arm Cortex M3",
   ISBN: 978-1463590154, Jonathan Valvano, copyright (c) 2011

   Program 5.12, section 5.7

 Copyright 2011 by Jonathan W. Valvano, valvano@mail.utexas.edu
    You may use, edit, run or distribute this file
    as long as the above copyright notice remains
 THIS SOFTWARE IS PROVIDED "AS IS".  NO WARRANTIES, WHETHER EXPRESS, IMPLIED
 OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF
 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE.
 VALVANO SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL,
 OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
 For more information about my classes, my research, and my books, see
 http://users.ece.utexas.edu/~valvano/
 */

// oscilloscope or LED connected to PD0 for period measurement
#include "hw_types.h"
#include "sysctl.h"
#include "lm3s1968.h"
#include "sound.h"
#include "dac.h"
// #include "lm3s1968.h"

#define NVIC_SYS_PRI3_R         (*((volatile unsigned long *)0xE000ED20))  // Sys. Handlers 12 to 15 Priority
#define NVIC_ST_CTRL_R          (*((volatile unsigned long *)0xE000E010))
#define NVIC_ST_RELOAD_R        (*((volatile unsigned long *)0xE000E014))
#define NVIC_ST_CURRENT_R       (*((volatile unsigned long *)0xE000E018))
#define NVIC_ST_CTRL_CLK_SRC    0x00000004  // Clock Source
#define NVIC_ST_CTRL_INTEN      0x00000002  // Interrupt enable
#define NVIC_ST_CTRL_ENABLE     0x00000001  // Counter mode
#define GPIO_PORTD_DIR_R        (*((volatile unsigned long *)0x40007400))
#define GPIO_PORTD_DEN_R        (*((volatile unsigned long *)0x4000751C))
#define SYSCTL_RCGC2_R          (*((volatile unsigned long *)0x400FE108))

void DisableInterrupts(void); // Disable interrupts
void EnableInterrupts(void);  // Enable interrupts
long StartCritical (void);    // previous I bit, disable interrupts
void EndCritical(long sr);    // restore I bit to previous value
void WaitForInterrupt(void);  // low power mode
#define GPIO_PORTG2             (*((volatile unsigned long *)0x40026010))

// **************SysTick_Init*********************
// Initialize Systick periodic interrupts
// Input: interrupt period
//        Units of period are 20ns
//        Maximum is 2^24-1
//        Minimum is determined by length of ISR
// Output: none
void SysTick_Init(unsigned long period){int nop;
  SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOG;
  nop = 0;
  nop = nop + 1;
  GPIO_PORTG_DIR_R &= 0x07;
  GPIO_PORTG_DIR_R |= 0x04;
  GPIO_PORTG_AFSEL_R |= 0xF8;
  GPIO_PORTG_PUR_R |= 0xF8;
  GPIO_PORTG_DEN_R |= 0xFF;
  NVIC_ST_CTRL_R = 0;         // disable SysTick during setup
  NVIC_ST_RELOAD_R = period-1;// reload value
  NVIC_ST_CURRENT_R = 0;      // any write to current clears it
  NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0x40000000; // priority 2
                              // enable SysTick with core clock and interrupts
  NVIC_ST_CTRL_R = NVIC_ST_CTRL_ENABLE+NVIC_ST_CTRL_CLK_SRC+NVIC_ST_CTRL_INTEN;
```

```
 73   }
 74
 75   void SysTick_Switch(unsigned int on) {
 76     if (on == 0) {
 77       NVIC_ST_CTRL_R = 0;
 78     } else {
 79       NVIC_ST_CTRL_R = NVIC_ST_CTRL_ENABLE+NVIC_ST_CTRL_CLK_SRC+NVIC_ST_CTRL_INTEN;
 80     }
 81   }
 82
 83
 84   void SysTick_Handler(void) {
 85     GPIO_PORTG2 ^= 0x04;        // toggle PD0
 86     DAC_Out(sinArray[index]);
 87     index+=1;
 88     if (index >= SAMPLE_RATE) {
 89       index = 0;
 90     }
 91   }
 92
 93   void SysTick_Wait(unsigned long delay) {
 94     volatile unsigned long elapsedTime;
 95     unsigned long startTime = NVIC_ST_CURRENT_R;
 96     do {
 97       elapsedTime = (startTime - NVIC_ST_CURRENT_R)&0x00FFFFFF;
 98     } while (elapsedTime <= delay);
 99   }
100
101
```