```c
// PeriodicSysTickInts.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 "inc/hw_types.h"
#include "driverlib/sysctl.h"
#include "SysTickInts.h"
#include "PLL.h"
#include "lm3s1968.h"
#include "DAC.h"
#include "Sound.h"
#include "Piano.h"
#include "math.h"

unsigned int guile_notes[] = {51, 51, 50, 0, 50, 51, 50, 51, 50, 51, 0, 50, 53, 0, 53, 51, 50, 46,
                              51, 51, 50, 0, 50, 51, 50, 51, 50, 51, 0, 50, 53, 0, 53, 51, 50, 46,
                              36, 36, 38, 39, 41, 43, 43, 41, 46, 44, 43, 44, 38, 39, 0, 46, 38, 41, 44, 46, 43, 0, 43, 41, 38,
                              36, 36, 38, 39, 41, 43, 43, 41, 46, 44, 43, 44, 38, 39, 0, 46, 38, 41, 44, 46, 43, 0, 43, 41, 38};
unsigned int guile_times[] = {1250, 625, 625, 625, 625, 5000, 1250, 625, 1250, 625, 625, 1250, 625, 625, 625, 1250, 1250, 1250,
                              1250, 625, 625, 625, 625, 5000, 1250, 625, 1250, 625, 625, 1250, 625, 625, 625, 1250, 1250, 1250,
                              5000, 1250, 1250, 625, 1875, 1875, 625, 1250, 2500, 1250, 625, 1875, 1875, 1875, 1250, 1250, 1250, 1250, 1875, 1875, 1250, 1250, 1250, 1250, 1250,
                              5000, 1250, 1250, 625, 1875, 1875, 625, 1250, 2500, 1250, 625, 1875, 1875, 1875, 1250, 1250, 1250, 1250, 1875, 1875, 1250, 1250, 1250, 1250, 1250};

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

int main(void){int i;
  // bus clock at 50 MHz
  //SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN);
  PLL_Init();
  SysTick_Init(50000);     // initialize SysTick timer
  EnableInterrupts();

  DAC_Init();
  Sound_Init();
  for (i = 0; i < sizeof(guile_notes)/sizeof(int); i++) {
    Sound_Play_Timing(guile_notes[i], guile_times[i]);
  }
  while(1){
    Piano_In();
    //WaitForInterrupt();
  }
}
```