```
// filename ******** Main.C **************
// LCD Display (HD44780) on Port H for the 9S12DP512
// Jonathan W. Valvano 9/18/09
// TCNT runs at 667ns,

//  This example accompanies the books
//   "Embedded Microcomputer Systems: Real Time Interfacing",
//        Thompson, copyright (c) 2006,
//   "Introduction to Embedded Systems: Interfacing to the Freescale 9S12",
//        Cengage Publishing 2009, ISBN-10: 049541137X | ISBN-13: 9780495411376

// Copyright 2009 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
// Purpose: test program for 4-bit LCD.C driver

/*
  size is 1*16
  if do not need to read busy, then you can tie R/W=ground
  ground = pin 1    Vss
  power  = pin 2    Vdd   +5V
  ground = pin 3    Vlc   grounded for highest contrast
  PH4    = pin 4    RS    (1 for data, 0 for control/status)
  PH5    = pin 5    R/W   (1 for read, 0 for write)
  PH6    = pin 6    E     (enable)
  PH3    = pin 14   DB7   (4-bit data)
  PH2    = pin 13   DB6
  PH1    = pin 12   DB5
  PH0    = pin 11   DB4
16 characters are configured as 2 rows of 8
addr  00 01 02 03 04 05 06 07 40 41 42 43 44 45 46 47
*/

#include <hidef.h>      /* common defines and macros */
#include <mc9s12dp512.h>     /* derivative information */
#pragma LINK_INFO DERIVATIVE "mc9s12dp512"

#include <stdio.h>
#include "LCD.H"
#include "PLL.H"
#include "fixed.h"

//--------------------TimerInit---------------
// initialize timer module to 0.667us(Boot Mode) TCNT clock
// inputs: none
// outputs: none
void TimerInit(void){
  TSCR1 = 0x80;   // Enable TCNT, 24MHz in both boo tand run modes
  TSCR2 = 0x04;   // divide by 16 TCNT prescale, TCNT at 667nsec
  PACTL = 0;      // timer prescale used for TCNT
/* Bottom three bits of TSCR2 (PR2,PR1,PR0) determine TCNT period
    divide  FastMode(24MHz)    Slow Mode (8MHz)
000   1     42ns  TOF  2.73ms  125ns TOF 8.192ms
001   2     84ns  TOF  5.46ms  250ns TOF 16.384ms
010   4    167ns  TOF  10.9ms  500ns TOF 32.768ms
011   8    333ns  TOF  21.8ms    1us TOF 65.536ms
100  16    667ns  TOF  43.7ms    2us TOF 131.072ms
101  32   1.33us  TOF  87.4ms            4us TOF 262.144ns
110  64   2.67us  TOF 174.8ms    8us TOF 524.288ms
111 128   5.33us  TOF 349.5ms   16us TOF 1.048576s */
// Be careful, TSCR1 and TSCR2 maybe set in other rituals
}
```

```
//---------------------mwait---------------------
// wait specified number of msec
// Input: number of msec to wait
// Output: none
// assumes TCNT timer is running at 667ns
void mwait(unsigned short msec){
unsigned short startTime;
  for(; msec>0; msec--){
    startTime = TCNT;
    while((TCNT-startTime) <= 1500){}
  }
}

//---------------------check---------------------
// if LCD broken toggle LED1 at 2Hz
// Input: last LCD status, 0 means bad
// Output: none
// Error: if status is zero, this function will not return
void check(short status){        // 0 if LCD is broken
  if(status ==0){
    for(;;) {
      PTP ^= 0x80;   // fast toggle LED
      mwait(250);    // 0.25 sec wait
    }
  }
}
//---------------------swait---------------------
// wait specified 2 sec, then clear LCD
// Input: none
// Output: none
// uses mswait and TCNT timer
void swait(void){
  PTP ^= 0x80;    // toggle LED0
  mwait(2000);    // 2 sec wait
  check(LCD_Clear());
}

//---------------------blank---------------------
// move cursor to second half of LCD display
// 32 spaces from address 08 to 40
// Input: none
// Output: none
void blank(void){
  check(LCD_OutString("                                "));
}

void main(void) {
  PLL_Init();       // set E clock to 24 MHz
  TimerInit();      // enable timer
  DDRP |= 0x80;     // PortP bit 7 is output to LED, used for debugging
  check(LCD_Open());
  check(LCD_Clear());
  for(;;) {         // Tests the three functions of Fixed.c
    LCD_OutString("Fixed_uD"); blank();
    LCD_OutString("ecOut2");
    swait();
    LCD_OutString("0     = "); blank();
    Fixed_uDecOut2(0);
    swait();
    LCD_OutString("1     = "); blank();
    Fixed_uDecOut2(1);
    swait();
    LCD_OutString("99    = "); blank();
```

```
Fixed_uDecOut2(99);
swait();
LCD_OutString("100   = "); blank();
Fixed_uDecOut2(100);
swait();
LCD_OutString("999   = "); blank();
Fixed_uDecOut2(999);
swait();
LCD_OutString("1000  = "); blank();
Fixed_uDecOut2(1000);
swait();
LCD_OutString("9999  = "); blank();
Fixed_uDecOut2(9999);
swait();
LCD_OutString("10000 = "); blank();
Fixed_uDecOut2(10000);
swait();
LCD_OutString("65534 = "); blank();
Fixed_uDecOut2(65534);
swait();
LCD_OutString("65535 = "); blank();
Fixed_uDecOut2(65535);
swait();

LCD_OutString("Fixed_sD"); blank();
LCD_OutString("ecOut3");
swait();
LCD_OutString("-32768 ="); blank();
Fixed_sDecOut3(-32768);
swait();
LCD_OutString("-10000 ="); blank();
Fixed_sDecOut3(-10000);
swait();
LCD_OutString("-9999  ="); blank();
Fixed_sDecOut3(-9999);
swait();
LCD_OutString("-999   ="); blank();
Fixed_sDecOut3(-999);
swait();
LCD_OutString("-1     ="); blank();
Fixed_sDecOut3(-1);
swait();
LCD_OutString("0      ="); blank();
Fixed_sDecOut3(0);
swait();
LCD_OutString("123    ="); blank();
Fixed_sDecOut3(123);
swait();
LCD_OutString("1234   ="); blank();
Fixed_sDecOut3(1234);
swait();
LCD_OutString("9999   ="); blank();
Fixed_sDecOut3(9999);
swait();
LCD_OutString("32767 = "); blank();
Fixed_sDecOut3(32767);
swait();

LCD_OutString("Fixed_uB"); blank();
LCD_OutString("inOut8");
swait();
LCD_OutString("0     = "); blank();
Fixed_uBinOut8(0);
```

```
    swait();
    LCD_OutString("2     = "); blank();
    Fixed_uBinOut8(2);
    swait();
    LCD_OutString("64    = "); blank();
    Fixed_uBinOut8(64);
    swait();
    LCD_OutString("100   = "); blank();
    Fixed_uBinOut8(100);
    swait();
    LCD_OutString("500   = "); blank();
    Fixed_uBinOut8(500);
    swait();
    LCD_OutString("512   = "); blank();
    Fixed_uBinOut8(512);
    swait();
    LCD_OutString("5000  = "); blank();
    Fixed_uBinOut8(5000);
    swait();
    LCD_OutString("30000 = "); blank();
    Fixed_uBinOut8(30000);
    swait();
    LCD_OutString("65534 = "); blank();
    Fixed_uBinOut8(65534);
    swait();
    LCD_OutString("65535 = "); blank();
    Fixed_uBinOut8(65535);
    swait();

  }

}
```