```
/*************************************************************
 Target MCU & clock speed: ATmega328P @ 1Mhz internal
 Name    : defines.h
 Author  : Insoo Kim (insoo@hotmail.com)
 Created : May 15, 2015
 Updated : May 16, 2015

 Description: Get system compile time & date and display on LCD 2*16
    Button toggling to turn on or off the backlight of LCD

 HEX size[Byte]:

 Ref:
    Donald Weiman    (weimandn@alfredstate.edu)
    http://web.alfredstate.edu/weimandn/programming/lcd/ATmega328/
      LCD_code_gcc_4d.html
 *************************************************************/

//Reference notes from the author that i referd at the begining
/*************************************************************************
    LCD-AVR-4d.c  - Use an HD44780U based LCD with an Atmel ATmega processor

    Copyright (C) 2013 Donald Weiman    (weimandn@alfredstate.edu)

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.
 */

/*************************************************************************
        File:    LCD-AVR-4d.c
        Date:    September 16, 2013

      Target:    ATmega328
    Compiler:    avr-gcc (AVR Studio 6)
      Author:    Donald Weiman

     Summary:    4-bit data interface, busy flag not implemented.
                 Any LCD pin can be connected to any available I/O port.
                 Includes a simple write string routine.
 */

/****************************** Program Notes ***************************

           This program uses a 4-bit data interface but does not use the
             busy flag to determine when the LCD controller is ready.  The
             LCD RW line (pin 5) is not connected to the uP and it must be
```

```
              connected to GND for the program to function.

            All time delays are longer than those specified in most datasheets
              in order to accommodate slower than normal LCD modules.  This
              requirement is well documented but almost always ignored.  The
              information is in a note at the bottom of the right hand
              (Execution Time) column of the instruction set.

  ************************************************************************

            The four data lines as well as the two control lines may be
              implemented on any available I/O pin of any port.  These are
              the connections used for this program:

                 -----------                    ----------
                | ATmega328 |                  |   LCD    |
                |           |                  |          |
                |        PD7|----------------->|D7        |
                |        PD6|----------------->|D6        |
                |        PD5|----------------->|D5        |
                |        PD4|----------------->|D4        |
                |           |                  |D3        |
                |           |                  |D2        |
                |           |                  |D1        |
                |           |                  |D0        |
                |           |                  |          |
                |        PD3|----------------->|E         |
                |           |         GND ---->|RW        |
                |        PD2|----------------->|RS        |
                 -----------                    ----------

  ************************************************************************/

//DS: Ch.8.2.1 default clock source is 1Mhz
#define F_CPU 1000000UL

#define MAXWORDCNT 10
#define LCD_MAXCOL 16
#define debug_PIN PORTB5

#define menuSelectInterval 4
#define adjustTimeInterval 8
#define halfSec 500 // 0.5 second checked by oscilloscope


#define DELAY_INST 40
// LCD interface (should agree with the diagram above)
//   make sure that the LCD RW pin is connected to GND
#define lcd_D7_port     PORTD                   // lcd D7 connection
#define lcd_D7_bit      PORTD7
#define lcd_D7_ddr      DDRD

#define lcd_D6_port     PORTD                   // lcd D6 connection
#define lcd_D6_bit      PORTD6
#define lcd_D6_ddr      DDRD
```

```
#define lcd_D5_port     PORTD                   // lcd D5 connection
#define lcd_D5_bit      PORTD5
#define lcd_D5_ddr      DDRD

#define lcd_D4_port     PORTD                   // lcd D4 connection
#define lcd_D4_bit      PORTD4
#define lcd_D4_ddr      DDRD

#define lcd_E_port      PORTD                   // lcd Enable pin
#define lcd_E_bit       PORTD3
#define lcd_E_ddr       DDRD

#define lcd_RS_port     PORTD                   // lcd Register Select pin
#define lcd_RS_bit      PORTD2
#define lcd_RS_ddr      DDRD

#define lcd_Backlight_port PORTB
#define lcd_Backlight_bit  PORTB0
#define lcd_Backlight_ddr  DDRB

#define tactile_Switch_port PINB
#define tactile_Switch_bit  PORTB4
#define tactile_Switch_ddr  DDRB

// LCD module information
#define lcd_LineOne     0x00                    // start of line 1
#define lcd_LineTwo     0x40                    // start of line 2
//#define   lcd_LineThree   0x14                  // start of line 3 (20x4)
//#define   lcd_lineFour    0x54                  // start of line 4 (20x4)
//#define   lcd_LineThree   0x10                  // start of line 3 (16x4)
//#define   lcd_lineFour    0x50                  // start of line 4 (16x4)

// LCD instructions
#define lcd_Clear           0b00000001          // replace all characters with ➥
  ASCII 'space'
#define lcd_Home            0b00000010          // return cursor to first position ➥
  on first line
#define lcd_EntryMode       0b00000110          // shift cursor from left to right ➥
  on read/write
#define lcd_DisplayOff      0b00001000          // turn display off
#define lcd_DisplayOn       0b00001100          // display on, cursor off, don't   ➥
  blink character
#define lcd_FunctionReset   0b00110000          // reset the LCD
#define lcd_FunctionSet4bit 0b00101000          // 4-bit data, 2-line display, 5 x ➥
  7 font
#define lcd_SetCursor       0b10000000          // set cursor position
```