

Program Source Code Requirements

There are requirements regarding content and format of program source files submitted to fulfill assignments in this class. It has been noted that the life cycle of software involves more than just coding and that coding may be a minor part of the overall time and effort spent on creating, maintaining, and supporting a software product. There are things that can be done when coding that help make software more supportable, maintainable, and in the context of course work easier for the instructor to help with and evaluate. Often the person who codes a program may not be the one that maintains it.

The following are requirements for this class:

- 1) Any source code submitted for an assignment or project in this class must have what we will call a header as illustrated on the next page. A program submitted without the header will not be graded and will receive a score of zero.
- 2) As shown in the example header, the name of the author, presumably you, will appear. By placing your name in the header you are declaring that the program is your creation and not copied from a classmate, the internet, or other resource. If the program is a product of group work then the names of all those involved must be listed in the header.
- 3) There may be times when pieces of code created by someone else (classmate, teacher, internet source, book, etc.) are placed in your program. The sections of this code must be clearly delineated and the author's name or the source identified with the section. Good delineation will include a long dashed line at the start of the section with author name adjacent and another long dashed line at the end of the section. (an example from the LCD routines is shown on the next page).

If you started with code from another author and modified it, state that the section was originally written by or obtained from _____ and then state modified by *your name*. In some cases it may be appropriate to state the nature of the modification. For example, I have provided routines to send data to the LCD display using specific ports on the MCU. If you decide to change the port(s) used then leave the original author as author but add a statement that ports were changed to xxx by *your name*.

- 4) Names used for variables, functions, etc. need to have clear meaning related to their function. That allows a person who didn't write the code to look it over and gain information about what the named item (variable, function, etc.) does in the context of the program (using simple variable names such as i, j, k, etc for loop indices is ok unless a more descriptive name would add understanding to reading the program code).
- 5) Required information: filename, author, version date, processor it runs on, compiler used, required libraries, hardware environment, information regarding what the program does.
- 6) Also, if header files (.h files) are modified then that fact must be documented as well.

Example header

```
// *****//
//      Example program to place characters on an LCD display      //
//                                                                    //
//      Filename:  lcd_4bit_demo.c                                  //
//      Author:    L.Aamodt                                        //
//      Version:   01/27/19 written                               //
//      Processor: NXP MKL25Z4                                    //
//      Compiler:  Keil uVision5                                  //
//      Library:   CMSIS core and device startup                 //
//                also needs lcd_lib_4bit.c                      //
//      Hardware:  NXP Freedom board connected to a 16x2 LCD display //
//      Software note: This program is a "bare metal" application since it //
//                    doesn't use an operating system.           //
//      Operation: Writes 11 alpha characters (F to P) to LCD display. //
//                    Blinks red LED on for 1 second, off for 1 second //
// *****//
```

Example function with delineation and author

```
/*-----*/
   Send data (one character, using ASCII code) to the LCD      L.Aamodt
*-----*/
void LCD_send_data(uint32_t data)
{
    PTE->PCOR = LCD_RW | LCD_EN;           // clear R/W and EN
    PTE->PSOR = LCD_RS;                     // set RS high
    PTB->PCOR = LCD_MASK;                   // clear output data bits to 0
    PTB->PSOR = (data & LCD_UPPER4_MASK)<<4; // output upper 4 bits of command
    pulse_the_LCD_enable();
    PTB->PCOR = LCD_MASK;                   // clear output data bits
    PTB->PSOR = (data & LCD_LOW4_MASK)<<8;  // output lower 4 bits
    pulse_the_LCD_enable();
}
/*-----*/
```