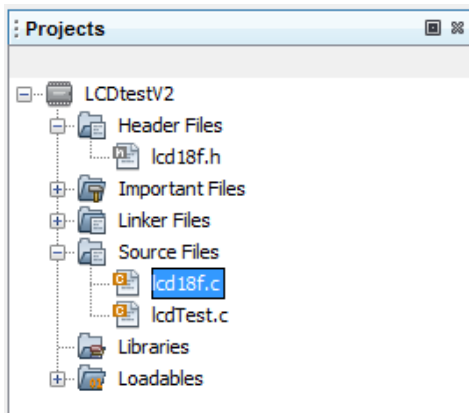


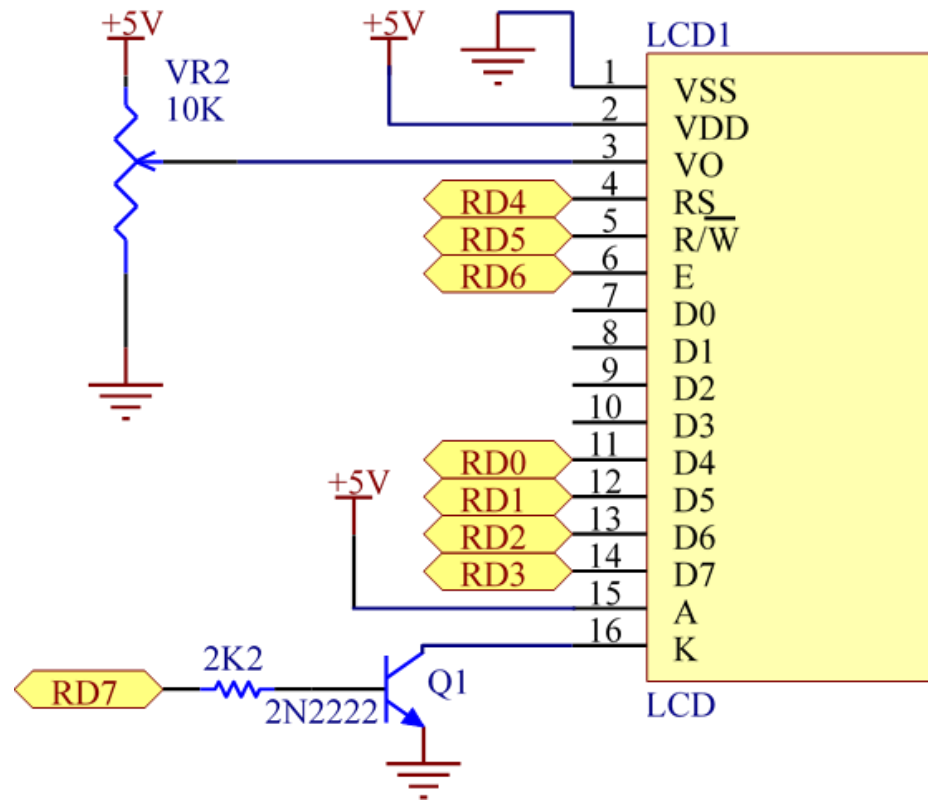
Lab 6 Write LCD Functions and Test the LCD

Write lcd18f.h and lcd18f.c files to add the LCD functions needed.

Write a test program called LCDtest.c to test all the functions of the lcd18f.c code in a creative way. At some point during the test your first name must be clearly displayed on the first line and your last name at the start of the second. Use must use a protection resistor in for each pin. Upload a picture of your wiring and commented code in a doc or pdf. Also upload a demo video to D2L.



Write the code in lcd18f.c to match the wiring in the following schematic:



CODE SECTIONS

“lcd18f.h”

```
/*
*****
* lcd18f.h
* ECET165 Embedded Micro-controllers
* Created: 13 Feb 2023
* Aaron Huinink
* Provides functionality for pic18f micro-controllers to connect to
an lcd on
* port d.
*****
*****/

#ifndef LCD18F_H
#define LCD18F_H

#ifdef __cplusplus
extern "C" {
#endif

#include <xc.h>
#define LCD_PORT PORTD
#define LCD_LAT LATD
#define LCD_TRIS TRISD
#define LCD_ANSEL ANSELD
#define LCD_WPU WPUD
#define LCD_K LATDbits.LATD7
#define LCD_EN LATDbits.LATD6
#define LCD_RW LATDbits.LATD5
#define LCD_RS LATDbits.LATD4
#define LCD_BF PORTDbits.RD3

#define LCD_STROBE LCD_EN = 1; __delay_us(1); LCD_EN = 0
#define LCD_PORTEN LCD_ANSEL = 0x0; LCD_TRIS = 0x0; LCD_WPU = 0xFF;
LCD_K = 1; __delay_us(1)
#define LCD_CLEAR LCDinstruct(0x01)
#define LCD_HOME LCDinstruct(0x02)
#define LCD_CURSORBLINK LCDinstruct(0x0F)

#ifndef _XTAL_FREQ
#define _XTAL_FREQ 6400000
#endif

// ===== FUNCTION PROTOTYPES
// =====

// ----- instruct ----- //
/*
* sends an instruction to the instruction register
*/
```

ECET 165
Embedded Microcontrollers

```
* ARGs: (instr<unsigned char> : the 2 byte instruction to be sent to
the lcd)
* RETURNS: [void]
*/
extern void LCDinstruct(unsigned char instr);

// ----- LCDinit ----- //
/*
* initializes an LCD display in 4 bit, 2 line display mode
* ARGs: (void)
* RETURNS: [void]
*/
extern void LCDinit();

// ----- LCDputc ----- //
/*
* prints a character to the lcd
* ARGs: (c<char> : the character to print to the lcd)
* RETURNS: [void]
*/
extern void LCDputc(char c);

// ----- LCDprint ----- //
/*
* prints a string of chars to the lcd
* ARGs: (*c<char> : the string to print to the lcd)
* RETURNS: [void]
*/
extern void LCDprint(char *c);

// ----- LCDgoto ----- //
/*
* goes to an address in DDRAM
* ARGs: (pos<unsigned char> : the position to go to in DDRAM)
* RETURNS: [address<unsigned char> : the current address in the
address counter]
*/
extern void LCDgoto(unsigned char pos);

// ----- LCDreturn ----- //
/*
* goes to next line of lcd display and clears it
* ARGs: (void)
* RETURNS: [void]
*/
extern unsigned char LCDreturn(unsigned char pos);

#ifdef __cplusplus
}
#endif

#endif /* LCD18F_H */
```

“lcd18f.c”

```
/*
*****
* lcd18f.c
* ECET165 Embedded Micro-controllers
* Created: 13 Feb 2023
* Aaron Huinink
* Provides functionality for pic18f micro-controllers to connect to
a 4bit,
* 2 line 5x8 font lcd on port d.
*****
*/

// ===== INCLUDES/DEFINES
=====//
#include <xc.h>
#include "lcd18f.h"

// ===== FUNCTION DEFS
=====//
void LCDinstruct(unsigned char instr){

    LCD_PORTEN; // enable LCD port

    // set rs to instruction register, write mode
    LCD_RW = 0;
    LCD_RS = 0;

    // send instruction
    LCD_LAT = (LCD_LAT & 0xF0) | (instr >> 4 & 0x0F); // send
MSnibble
    LCD_STROBE;
    __delay_us(1);
    LCD_LAT = (LCD_LAT & 0xF0) | (instr & 0x0F); // send LSnibble
    LCD_STROBE;
    __delay_ms(5); // allow busy flag to clear
};

void LCDinit(){
    __delay_ms(50); // wait for powerup

    LCD_PORTEN;

    LCD_RS = 0;
    LCD_RW = 0;

    __delay_us(1);

    // send the function set instruction 3 times
    LCD_LAT = (LCD_LAT & 0xF0) | 0x03;
```

```
LCD_STROBE;
__delay_ms(5);

LCD_LAT = (LCD_LAT & 0xF0) | 0x03;
LCD_STROBE;
__delay_ms(5);

LCD_LAT = (LCD_LAT & 0xF0) | 0x03;
LCD_STROBE;
__delay_ms(5);

//set 4 bit mode
LCD_LAT = (LCD_LAT & 0xF0) | 0x02;
LCD_STROBE;
__delay_ms(5);

// run setup functions
LCDinstruct(0x28); // set 4 bit mode, 2 line display, 5x8
font
LCDinstruct(0x08); // display off
LCDinstruct(0x0F); // display on, cursor on, blink on
LCDinstruct(0x01); // display clear
LCDinstruct(0x06); // entry mode set
};

void LCDputc(char c){
    LCD_PORTEN;
    LCD_RS = 1;
    LCD_RW = 0;

    // print character to lcd
    LCD_LAT = (LCD_LAT & 0xF0)|(c>>4); // send MSnibble
    __delay_us(1);
    LCD_STROBE;
    LCD_LAT = (LCD_LAT & 0xF0)|(c & 0x0F); // send LSnibble
    __delay_us(1);
    LCD_STROBE;
    __delay_ms(5); // allow busy flag to clear
}

void LCDprints(char *c){

    LCD_PORTEN; // enable the lcd port

    // set rs to data register, r/w to write mode
    LCD_RS = 1;
    LCD_RW = 0;

    __delay_us(1);

    unsigned char i = 0; // create an indexer variable

    while(c[i] != 0x00){ // while there are valid chars in the string

        // print character from c
```

```
        LCDputc(c[i]);

        i++;          // increase index by 1

    };
};

void LCDgoto(unsigned char pos){
    LCDinstruct(0x80+pos);
}

unsigned char LCDreturn(unsigned char pos){
    LCD_PORTEN;      // enable the LCD port

    unsigned char next_line = 0x40;      // next line begin address
    variable

    if (pos > 0x0F){
        next_line = 0x00;
    }
    LCDgoto(next_line);      // go to the beginning of the next line
    on the lcd

    for(unsigned char i = 0; i < 40; i++){
        LCDputc(' ');
    }

    LCDgoto(next_line);

    return next_line;
}
```

ECET 165 Embedded Microcontrollers

"lcdtest.c"

```
/*
*****
* lcdtest.c
* ECET165 Embedded Micro-controllers
* Created: 13 Feb 2023
* Aaron Huinink
* Demonstrates functionality of the lcd18f library for an LCD on
port d.
*****
*/

#include "lcd18f.h"
#include <xc.h>

// ===== PRAGMA CONFIG
// =====//
#pragma config WDTE = OFF           // Watchdog timer enable off
#pragma config FEXTOSC = OFF        // External oscillator off
#pragma config RSTOSC = HFINTOSC_64MHZ // Set reset oscillator to
high freq internal osc at 64MHz

// ===== MAIN LOOP
// =====//

void main(void) {

    LCDinit(); // initialize lcd

    // string variables to print
    char *intro = "My name is";
    char *name = "Aaron Huinink";

    LCDprints(intro); // print intro string
    LCDgoto(0x40); // go to next line
    LCDprints(name); // print my name
    while(1){}; // main loop
}
```


WIRING

