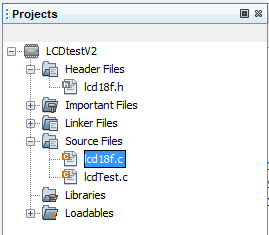
# Lab 6

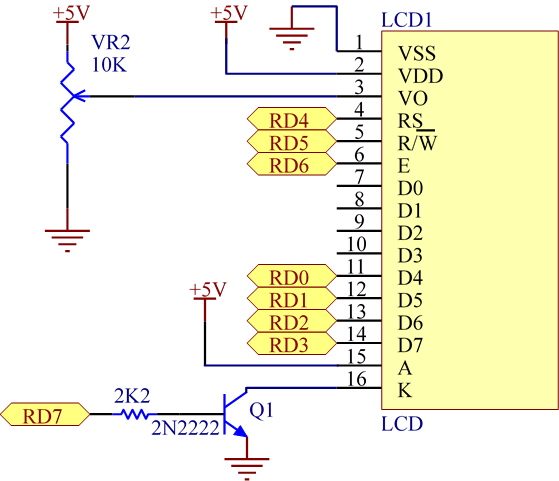
# Write LCD Functions and Test the LCD

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 64000000

#endif

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

// ----- instruct ----- //

/\*

\* sends an instruction to the instruction register

\* 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();

// ----- LCDprint ----- //

/\*

\* prints a character to the lcd

\* ARGS: (c<char> : the character to print to the lcd)

\* RETURNS: [void]

\*/

extern void LCDprintc(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 LCDprints(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 LCDprintc(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] != 0x0){ // while there are valid chars in the string

// print character from c

LCDprintc(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++){

LCDprintc(' ');

}

LCDgoto(next\_line);

return next\_line;

}

**“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

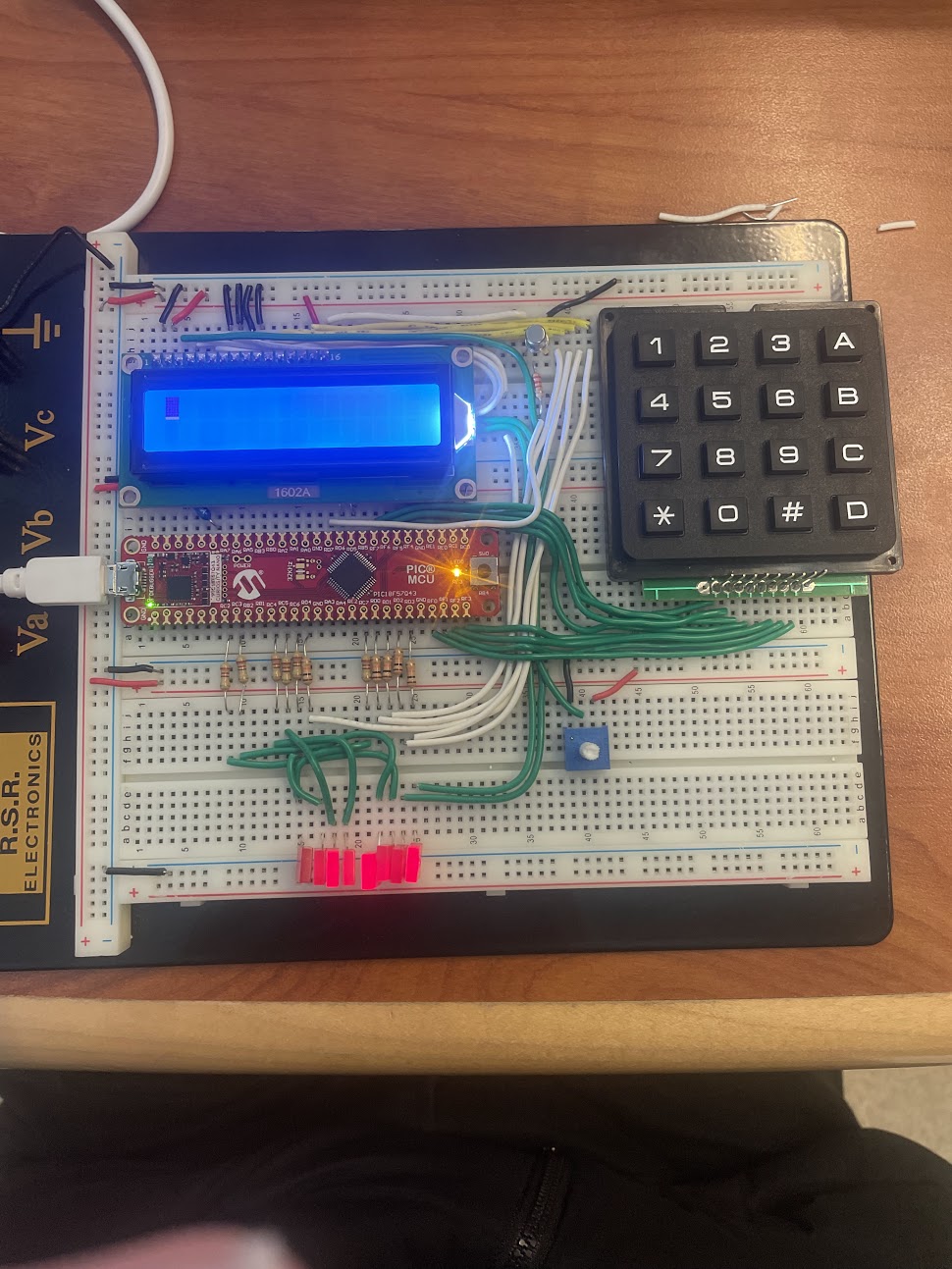
LCDgoto(0x40); // go to next line

LCDprints(name); // print my name

while(1){}; // main loop

}

**WIRING**

****