LCD, Timers and Serial IO

Due Date: Today +1 Week

In this lab experiment, the student will explore the LCD (liquid crystal display) and it's mode of operation using specific libraries. In addition, students will use additional feature of the serial library to transfer and receive data from and to PC.

Objectives

- ✓ Understand the mbed library import wizard.
- ✓ Understand LCD functionalities and libraries.
- ✓ Understand the mbed timer functions and the TextLCD library.

Liquid Crystal Display

Overview

The principle of an LCD is illustrated in Figure 1. The liquid crystal is an organic compound which responds to an applied electric field by changing the alignment of its molecules, and hence the light polarization that it introduces. A small quantity of liquid crystal is contained between two parallel glass plates. A suitable field can be applied if transparent electrodes are located on the glass surface. In conjunction with the external polarizing light filters, light is either blocked or transmitted by the display cell. The electrodes can be made in any pattern desired. These may include single digits or symbols, or the standard patterns of bargraph, seven-segment, dot matrix, starburst and so on. Alternatively, they may be extended to complex graphical displays, with addressable pixels.

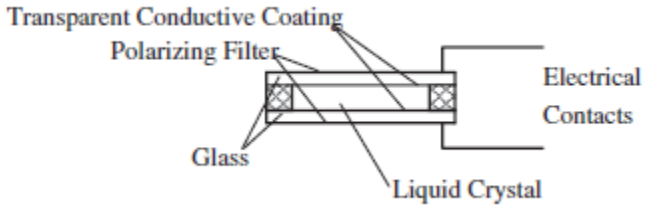

Figure 1 – Liquid Crystal Display LCD

The first such controller to gain widespread acceptance was the Hitachi HD44780. While this has been superseded by others, they have kept the interface and internal structure of the Hitachi device. It is important to know its main features, in order to design with it. The HD44780 contains an 80-byte random access memory (RAM) to hold the display data, and a read only memory (ROM) for generating the characters. It has a simple instruction set, including instructions for initialization, cursor control (moving, blanking, blinking) and clearing the display. Communication with the controller is made via an 8-bit data bus, three control lines and an enable/strobe line (E). These are itemized in Figure 2.

RS	Register Select: 0 = Instruction register 1 = Data Register
R/$\overline{W}$	Selects read or write
E	Synchronizes read and write operations
DB4 - DB7	Higher order bits of data bus; DB7 also used as Busy flag
DB0 - DB3	Lower order bits of data bus; not used for 4-bit operation

Figure 2 – LCD Control Lines

Mbed TextLCD Library

There is an mbed library which makes the use of an alphanumeric LCD much simpler and quicker to program. The mbed TextLCD library is also more advanced than the simple functions we have created; in particular, the Text LCD library performs the laborious LCD setup routine for us. The TextLCD definition also tells the LCD object which pins are used for which functions. The object definition is defined in the following manner:

TextLCD lcd(int rs, int e, int d0, int d1, int d2, int d3);

We need to ensure that our pins are defined in the same order. For our particular hardware setup (described in Figure 3) this will be:
TextLCD lcd(D8, D9, D4, D5, D6, D7);
In this experiment VCC module will be connected to 5V instead of 3.3V

Pin number	TextLCD pins	mbed pins
1	GND	0V
2	VCC	3.3V
3	VO	0V, via 1k resistor
4	RS	p15
5	RW	0V
6	E	p16
7	D0	not connected
8	D1	not connected
9	D2	not connected
10	D3	not connected
11	D4	p17
12	D5	p18
13	D6	p19
14	D7	p20

Figure 3 – LCD Pin Diagram

Lab Procedure

Task

- Experiment with an HD44780 16*2 LCD to display keyboard characters and text. The Nucleo board will randomly display a character, the user will need to input the displayed character.

Procedure

This experiment will utilize HD44780 16*2 LCD to display keyboard characters and text. The purpose of this experiment is to discover LCD functionalities, libraries and timers.

In this experiment the program will hold a character array formed of alpha-numeric entries. The Cortex M4 controller will randomly display an array character. Furthermore the user will have to input the displayed character from the keyboard as fast as possible to measure his input reflex.

The alpha numeric array will hold upper/lower case characters and numbers from 0 to 9.

Challenges:

1. Use external LEDs to display the user reflexes based on chosen time threshold.
2. Create a program with varying input wait speed. (External potentiometer).
3. Create a counter and an average algorithm for 10 rounds.
4. Create a scrolling message at the end of the game if the average speed is above a certain value.
5. Include a reset button (on board pushbutton switch) to restart the game.

Tips

- Digital Input ports, when declared, use a Pull – Up mode by default.
- Only the labels written in **blue/white** or **green/white** (i.e. PA_4, PB_5, A0, D14, LED1...) must be used in your code. The other labels are given as information (alternate-functions, power pins, ...).
- The mbed development environment holds a detailed list of available functions and classes.
- To import the TextLCD.h library right click on your project folder and choose import wizard.
- Use a switch – case statement to design a state machine architecture.