# mbed Lab 3: Do You See What I See?

CS 220

## Introduction

Using a host computer to display output from the KL25Z board is a bit of a drag. Who wants to carry a computer around with them just to find out what the current temperature is? Using a small LCD display panel to show output would be a far better solution. In this lab, you'll use an mbed Components library to conveniently access a two lines by 16 characters LCD display, the Newhaven Display NHD-0216HZ-FSW-FBW-33V3C. (That's a mouthful. From now on, I'll just refer to it as the LCD display.) The Components libraries are similar to the basic interfaces (DigitalOut, etc.) provided by the mbed system, but on a larger scale. These libraries amount to being small device drivers — pieces of software that interface complex I/O peripherals to a computer system.

## Lab Objectives

1. Use the mbed's TextLCD library to display text on an LCD display.
2. Display temperature readings from the TMP36 temperature sensor on an LCD display.
3. Gain confidence in using the mbed development system, and designing and wiring circuits.

## Lab Parts List

1. KL25Z board and Quick Reference Card
2. USB cable
3. Breadboard, with LCD display already mounted on it.
4. Your temperature-measuring circuit from the last lab: Analog Devices TMP36 analog temperature sensor, two 10.0$\mu$F capacitors, blue, and one 750$\Omega$ resistor, tan with colored stripes.
5. Plenty of wires.

## Lab Procedure

Read each of the following steps **completely** before acting on them.

1. You won't need to apply power to the KL25Z until later, so set the USB cable aside for now.
2. Visit the mbed development website and sign-in.
3. Let's start by trying out the "Hello World" program for the LCD display. Visit the page for the TextLCD's Hello World program.

4. Import the Hello World program. When the "Import Program" dialog box opens, check the update box so that you pick up the most recent versions of any library files. Open `main.cpp` and take a look at TextLCD's constructor — six pins with the functions `rs`, `e`, and `d4`–`d7`. `rs` is register select. The LCD's command register is selected when this pin is low, otherwise the data register is selected. `e` is the operation enable pin. It's similar to the load pin that you've used with sequential chips in Hack. The four remaining pins are used for transferring display data or commands.

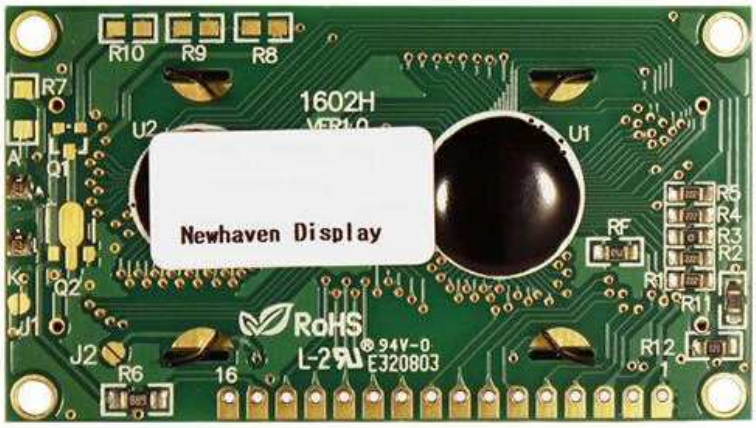
5. As you've done before, you'll need to replace the six pin names in the TextLCD's constructor with six KL25Z pin names and then use wires to connect these pins to the corresponding pins on the LCD display. Let's do that now; modify the constructor for the TextLCD component like so:

   ```
   TextLCD lcd(PTE30, PTE29, PTE23, PTE22, PTE21, PTE20); // rs, e, d4-d7
   ```

   This declares that the KL25Z's pin PTE30 will be used to control the LCD display's rs pin, KL25Z pin PTE29 will be used to control the LCD display's e pin, etc.

6. Let's continue with the LCD display's pins. Here's an image of the LCD display:


   There are 16 connection pins along the bottom edge of the display's circuit board. Pin 1 is the leftmost pin and pin 16 is the rightmost pin. How do we know this? These two pin numbers are printed on the bottom of the circuit board:



7. This table shows the function of each of the LCD display's pins. Note that the RS, E, and DB4–DB7 pins correspond directly to the pins of the same name on the mbed's TextLCD component.

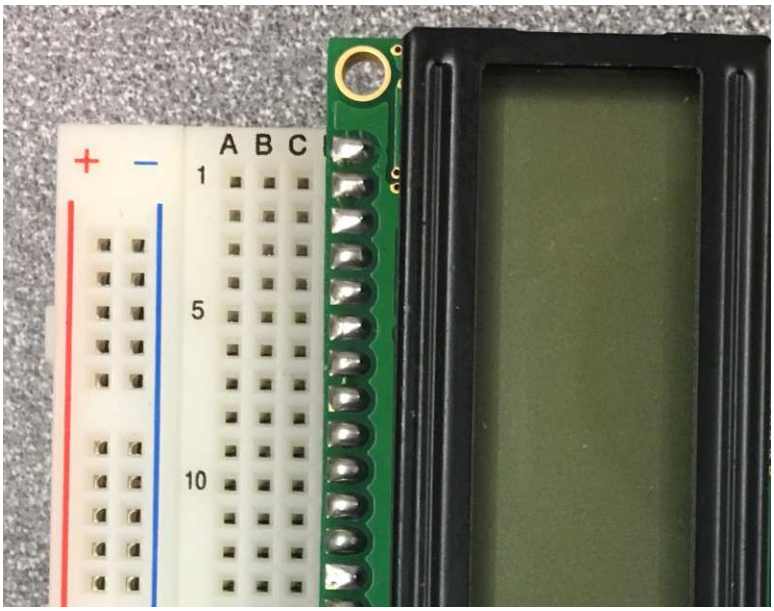
| **Pin** | **Function/Connection** |
|---|---|
| 1 | Circuit ground; connect to GND |
| 2 | Circuit power; connect to 3.3V |
| 3 | No connect |
| 4 | RS; select data register or command register |
| 5 | R/W; connect to GND because we only write data or a command to the display |
| 6 | E; enable command/data transfer |
| 7–10 | DB0–DB3; no connect |
| 11–14 | DB4–DB7 |
| 15 | Backlight power; connect to 3.3V |
| 16 | Backlight ground; connect to GND |

8. Notice that there are actually eight data pins. We're only using four. What gives? These displays have two data modes — eight-bit mode, in which an entire byte of display or command data is transferred in one clock cycle, or four-bit mode, in which a byte of display or command data is transferred in two clock cycles, with the two four-bit nibbles multiplexed over four of the data pins. We're using four-bit mode. Why two modes? I really can't think of a clear advantage of eight-bit mode (Maybe you can!), but an advantage of four-bit mode is that fewer connections need to be made (and therefore consumed) on the mbed board.

9. Make the following jumper wire connections from the KL25Z to the LCD display. Remember that LCD display pin 1 is the leftmost pin.

| **KL25Z Pin** | **LCD Display Pin & Function** |
|---|---|
| PTE30 | 4 (RS) |
| PTE29 | 6 (E) |
| PTE23 | 11 (DB4) |
| PTE22 | 12 (DB5) |
| PTE21 | 13 (DB6) |
| PTE20 | 14 (DB7) |

10. The LCD runs on 3.3V. You'll be making several power and ground connections. To make power and ground wiring easier, connect a `P3V3` pin from the KL25Z to the "`+`" bus column on the breadboard and a `GND` pin from the KL25Z to the "`-`" bus column on the breadboard. Refer to this figure:




All of the pins in each of these bus columns are electrically connected, just like the pins in the numbered rows in the middle of the breadboard. Use these bus columns for the following power and ground connections to LCD display pins.

- The following LCD display pins require a power connection, so they must be connected to the "`+`" bus strip:

  2, 15

- The following LCD display pins require a ground connection, so they must be connected to the "`-`" bus strip:

  1, 5, 16

11. Make the wiring connections.

12. Let's turn our attention back to the TextLCD Hello World program.

13. Using your USB cable, connect the KL25Z to your host computer, compile and download your program, and run your program. The LCD should display `Hello World!`.

14. Now, let's use the LCD to display the current temperature. Open your temperature sensor program from last week. Comment-out any lines referring to the Serial interface that we used last week. Import the TextLCD component into your temperature sensor program:

    (a) Visit this page for the TextLCD component's library code and click the `Import into Compiler` button.

    (b) This pop-up will appear:


Configure your settings so that they appear as here, except for the Target Path, which should be set to the name of your temperature sensor program.

15. Add the line

    ```
    #include "TextLCD.h"
    ```

    after the line

    ```
    #include "mbed.h"
    ```

    in your temperature sensor program.

16. Copy the declaration for the TextLCD object from your TextLCD Hello World program to your temperature sensor program.

17. Modify your temperature sensor program so that the current temperature is displayed on the LCD, updated once per second. The temperature display should be similar to

    The temperature is 72.45°F.

    You can use the character code `\337` in your `printf()` string to print the ° symbol.

18. To finish-up, disconnect both ends of the USB cable. You'll be using the LCD display in the next lab, so leave that circuit connected. You're finished with the temperature sensor circuit, so carefully take it apart, putting the wires back into the wires bag, and putting the temperature sensor, two capacitors, and resistor back into the static shielding bag. Then, carefully transfer the KL25Z board and the breadboard to the storage box, put everything else back into the storage box, and carefully close its lid.