

Implementing Liquid Crystal Displays and PIC communication

Application Note

Design Team 5

Tim Ross

April 11, 2010

Executive Summary

The purpose of this Application Note is to be a guide for interfacing an LCD with a PIC, how to program for outputting to the LCD, and how to wire the LCD to the PIC. It explains why the method of connection was chosen, in order to allow for versatility of LCD and PIC options.

Keywords: LCD, Menu, LCD Programming

1. Introduction

In all systems, communication of data to the user is very important. A system without this ability is of no use to a user. Liquid Crystal Displays, abbreviated LCD, are important for visually communicating information to an end user. They offer a simple output for the information. Often, LCDs are the primary source of interfacing data with the end user.

2. Objective

The objective of the Application Note is to provide a guide for interfacing the LCD with the PIC processor using examples. It will show an example of how to connect the LCD using USART to PIC. This Application Note will also give background information explaining how the components and provide simple software functions to make outputting to the LCD simpler.

3. Background:

The following links are to technical data about the Crystalfontz CFA631-YFB-KS LCD. This LCD will be used as an example LCD for designing menu layout and menus.

LCD Online Information: <http://www.crystalfontz.com/product/CFA632YFBKS.html>

LCD Data Sheet:

http://www.crystalfontz.com/products/632/datasheets/2026/CFA632_Data_Sheet_v2.1.pdf

PIC Data Sheet:

<http://ww1.microchip.com/downloads/en/DeviceDoc/39631a.pdf>

USART Information:

http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter

USART.H:

<http://www.koders.com/c/fid4913987042A7E1EB86016A7A0893873FC1246F29.aspx>

4. Implementation:

Step 1: Power the LCD

The CFA632-YFB-KS LCD is a 7 pin LCD display. For this example only 3 pins are required, $5+V(LCD)$, *GROUND*, and *DATA_IN*. These pins are shown in figure 1.

The first step is to power the LCD. As you can see from the Figure 1, *GROUND* is Pin 1, $5+V(LCD)$ is the 2nd pin. Typically, red wire is used for the +5V pin and black wire is used for wiring to ground. The $5+V(LED)$ for the Backlight LED can also be wired if desired on Pin 3.

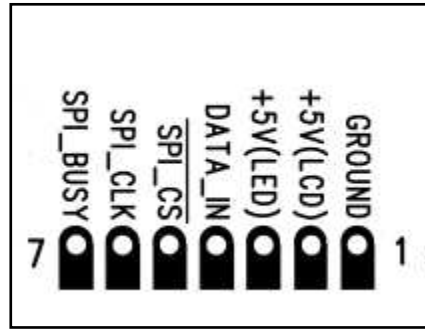


Figure 1: LCD Pin Layout

Step 2: Power the PIC

The PIC used for this example is the 40-pin PIC18F4420/4520 shown in Figure 2. The next step is to power the PIC. For the PIC to operate, V_{dd} should be at +5V and V_{ss} should be at ground. Again, red wire should be used for the +5V and black wire used for wiring to ground.

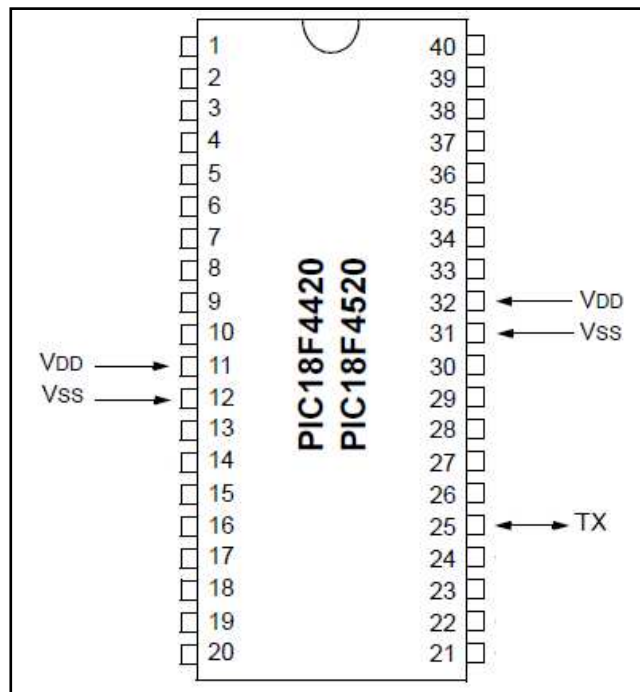


Figure 2: PIC Pin Layout

Step 3: Connect the PIC to the LCD

The next step is to set up the connecting wire for transmitting data from the PIC to the LCD. The output of the PIC's USART Transmit Pin will be wired to the LCD's DATA_IN pin. In the example, this means wire Pin 25 of the PIC to Pin 3 of the LCD. The LCD and PIC are now wired to communicate data and output data.

The reason the USART Transmit pin is connected to the LCD is because the USART component of the microcontroller is what will be used for transmitting data to the LCD.

The reason for using the USART component for transmitting is because the LCD takes serial information. The microprocessor can output data by using a port register, but using this method outputs the data simultaneously or in parallel. To transmit the data serially we must use the USART. The USART is an integrated circuit which takes bytes of data and transmits the individual bits in a sequential fashion. The output of the USART asynchronous transmit is Pin 25 of the PIC18F4420.

Step 4: Programming the PIC

A header file for USART communication is provided in the background section. Two USART.H functions that are used for outputting are **putcUSART(char Buffer)** and **BusyUSART()**. **putcUSART(char Buffer)** is a void function which outputs the character Buffer on the TX Pin. **BusyUSART()** is a wait function which prevents USART use until the hardware has completed writing the character to the pin.

Figure 3 shows two LCD writing functions, which use the USART.H functions to output to the LCD. The first function writes a character array to the LCD. The second function safely writes a single character to the LCD.

```
void lcd_write(char *buffer)
{
    int i = 0;
    while(BusyUSART());
    while(buffer[i] != '\0')
    {
        putcUSART(buffer[i]);
        i++;
        while(BusyUSART());
    }
}

void lcd_writechar(char buffer)
{
    while(BusyUSART());
    putcUSART(buffer);
}
```

Figure 3: LCD Writing functions

5. Conclusion:

In the end, it should be noted that this method is not the only way to interface an LCD and PIC. Many LCDs use parallel communication instead of serial communication. This guide is an example of connecting to a serial communicating Liquid Crystal Display. All reference sources are shown in the background section.