Model LR-DISPLAY

LCD Display Option
for the Model LR-F2812DAQ Data
Acquisition Daughtercard

Installation and Operating Guide

January 4, 2006

Document Number: 4-903-41

TABLE OF CONTENTS

I. Introduction

The Link Research model LR-DISPLAY is a 4 lines by 20 character, LCD display option for the model LR-F2812DAQ series of data acquisition daughtercards. The LCD display comes with a 6-inch flexible interface cable, to allow easy positioning of the display relative to the daughtercard. If this display module is purchased separately from the daughtercard, then a 14-pin header is also supplied to be installed on the daughtercard by the user.

The display module comes with a full set of software functions (API), which should allow the user to have the display operating in a very short time.

II. Hardware Installation

The LCD display module includes a 6 inch flexible ribbon cable. If the LCD display module was purchased separately from the DAQ daughtercard, then a 14-pin header is also supplied, which the user should solder to the daughtercard. This allows the display assembly to be easily attached and detached from the daughtercard. (If the display option is ordered at the same time that the daughtercard was purchased, this header will be installed at the factory.)

If user installation of the 14-pin header is required, the following procedure should be followed. The header must be installed on the daughtercard at position J5. The header has a short pin side, and a long pin side. The short pin side should be soldered to the board, as shown in figure 1. It is recommended that a water soluable solder, such as Kester type 331 be used for this purpose. Flux is also recommended. Kester type ZX2331 is recommended for this purpose. After soldering, the flux should be washed off using warm water. The board should be thoroughly dried prior to applying power to it.

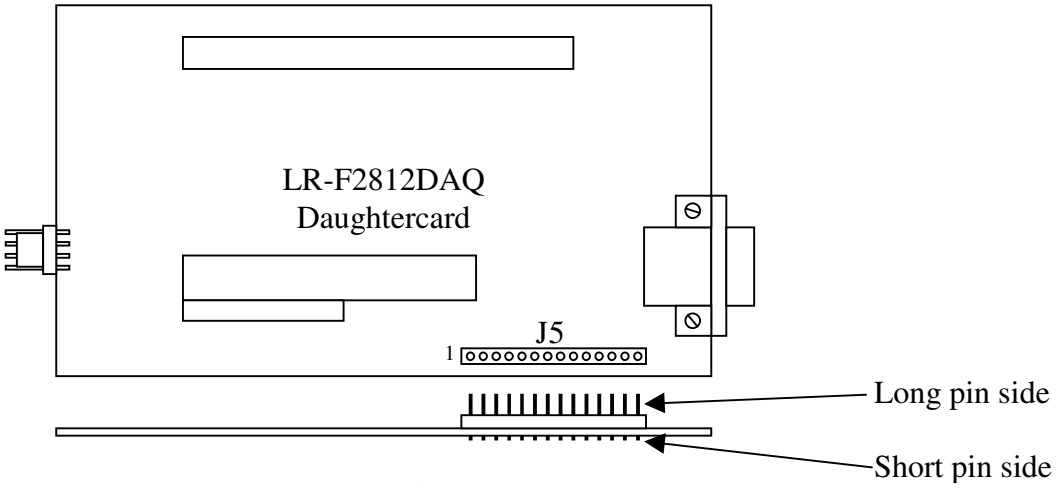

Figure 1

Once the header has been installed, the display assembly can be easily attached to the DAQ daughtercard. ***Note that the connector is NOT keyed, and care should be taken to orient the display as shown in figure 2.***

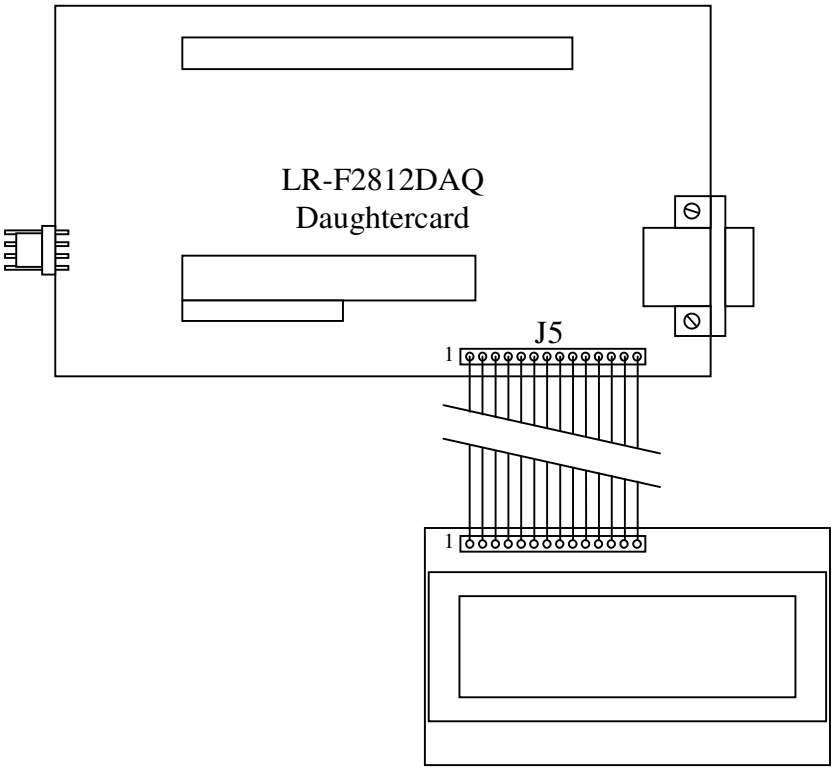

Figure 2

NOTE: The display module must be attached to the daughtercard with no power applied to the daughtercard. This is to avoid damage to the display module, the daughtercard, or both.

III. Hardware Description

1. LCD Display Interface

The LCD interface consists basically of 12 DSP GPIO lines, an octal bi-directional buffer, and connector J5. The interface is compatible with the Hitachi 44780 LCD controller IC. Connector J5 is pinned out in one of the common configurations – a single row of 14 pins. The connector pin-out is shown in the following table:

Pin	Signal	Description
1	GND	Module Ground
2	+5V	Module Power
3	Ve	Contrast voltage
4	RS	Register Select
5	R/W	Read/Write
6	E	Enable
7	D0	Parallel Data Bus
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	

.

IV. Software Description

1. General Information

The user accesses the LCD display through a high level API as described in the next section. All of the API functions are non-blocking except for **`init_LCD_display()`**. This feature eliminates the possibility of the relatively slow display interface from interfering with time critical DSP functions. The following section provides a detailed description of each function included in the API.

2. LCD API Functions

This section describes in detail, the application programming interface (API) used to access the LCD display module. These high level functions allow easy use of the display module. At the end of the section, a brief example is given showing what it takes to write Hello World on the LCD display.

```
void init_GPIO(void);
```

Description

This function should be called once before calling any other display function. The purpose of this function is to initialize specific GPIO pins on the TMS320F2812 device to provide compatibility with the display module.

Dependencies

None

Required include files:

None

```
void init_LCD_display(void);
```

Description

This function should be called once before calling any other display function. The purpose of this function is to initialize the 44780 compatible display module. This function includes software delays, and should not be used after time critical DSP operations have started.

Dependencies

Prior to calling this function, the function init_GPIO() must be called to configure the TMS320F2812 I/O pins associated with the display interface.

Required include files:

lcd.h

```
void lcd_write_line(int, int, unsigned char*);
```

Description

This function is used to display a character string starting at any position on the display. The first parameter specifies which line of the display is to be written. Valid values are 1,2,3, and 4. The second parameter specifies the starting position within the specified line. Valid values are 0 thru 19. The third parameter is a pointer to the NULL terminated character string to be written to the display.

Dependencies

Prior to calling this function, the function **`init_LCD_display()`** must be called to initialize the LCD display module.

Required include files:

lcd.h

`void lcd_write_line_center(int,unsigned char*);`

Description

This function is used to display a text string that is centered on a particular line of the display. The first parameter specifies which line of the display is to be written. Valid values are 1,2,3, and 4. The second parameter is a pointer to the NULL terminated character string to be written to the display.

Dependencies

Prior to calling this function, the function **`init_LCD_display(void)`** must be called to initialize the LCD display module.

Required include files:

lcd.h

`void print_byte(char, int, int);`

Description

This function is used to format and print a *signed byte* to the display. The first parameter is the byte to be displayed, the second parameter specifies the number of display positions, including sign, to be used for displaying the number, and the third parameter is a flag specifying whether or not leading zeros are to be suppressed. A value of zero causes leading zeros to be displayed, whereas a non-zero value causes leading zeros to be suppressed.

Dependencies

Prior to calling this function, the function **`init_LCD_display()`** must be called to initialize the LCD display module.

The function **`lcd_set_position()`** should be called prior to calling this function to specify the starting position for displaying the value.

Required include files:

lcd.h

`void print_int(int, int ,int);`

Description

This function is used to format and print a *signed integer* value to the display. The first parameter is the integer to be displayed, the second parameter specifies the number of display positions, including sign, to be used for displaying the number, and the third parameter is a flag specifying whether or not leading zeros are to be suppressed. A value of zero causes leading zeros to be displayed, whereas a non-zero value causes leading zeros to be suppressed.

Dependencies

Prior to calling this function, the function **`init_LCD_display()`** must be called to initialize the LCD display module.

The function **`lcd_set_position()`** should be called prior to calling this function to specify the starting position for displaying the value.

Required include files:

lcd.h

`void print_long(long, int, int);`

Description

This function is used to format and print a *signed long* value to the display. The first parameter is the long integer to be displayed, the second parameter specifies the number of display positions, including sign, to be used for displaying the number, and the third parameter is a flag specifying whether or not leading zeros are to be suppressed. A value of zero causes leading zeros to be displayed, whereas a non-zero value causes leading zeros to be suppressed.

Dependencies

Prior to calling this function, the function **`init_LCD_display()`** must be called to initialize the LCD display module.

The function **`lcd_set_position()`** should be called prior to calling this function to specify the starting position for displaying the value.

Required include files:

lcd.h

```
void lcd_write_data(unsigned char);
```

Description

This function is used to print a single ASCII character to the display. The one and only parameter specifies the ASCII character.

Dependencies

Prior to calling this function, the function **`init_LCD_display()`** must be called to initialize the LCD display module.

The function **`lcd_set_position()`** should be called prior to calling this function to specify the starting position for displaying the value.

Required include files:

lcd.h

```
void lcd_set_position(int, int);
```

Description

This function is used to set a starting position for subsequent text writing functions. The first parameter specifies the display line number (1 thru 4), and the second parameter specifies the position within the specified line (0 thru 19).

Dependencies

Prior to calling this function, the function **`init_LCD_display()`** must be called to initialize the LCD display module.

Required include files:

lcd.h

```
void service_lcd_display(void);
```

Description

This function is used in non-DSP/BIOS applications to perform low-level display handling functions. It should be called periodically either from the application's main loop, or from a background task.

In a DSP/BIOS application, the code within this function should be placed in a ***periodic*** function, and executed at a rate of about 1000 times per second.

Dependencies

Prior to calling this function, the function **`init_LCD_display()`** must be called to initialize the LCD display module.

Required include files:

lcd.h

3. Sample Application

The following example shows the ease at which a message can be sent to the LCD display module.

```
#include "DSP28_Device.h"
#include "DSP28_Globalprototypes.h"
#include "lcd_non_bios.h"
#include "string.h"

main()
{
    init_GPIO();      // Initialize all the GPIO for this project
    init_LCD_display();
    delay_msec(1000);

    lcd_write_line_center(1, (unsigned char*)"Hello");
    lcd_write_line_center(2, (unsigned char*)"World!");
    lcd_write_line_center(3, (unsigned char*)"This is Line 3");
    lcd_write_line_center(4, (unsigned char*)"This is Line 4");

    while(1)
    {
          // do something
          service_lcd_display();
    }     // end of while loop
}   // end of main
```