ANALOG DEVICES

AN-908
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

Interfacing the HD44780-Based Character LCD to an ADuC702x

MicroConverter ADuC702x Development System

INTRODUCTION

A vast array of LCD displays are currently available and, fortunately, many of these LCDs comply with the HD44780U standard. This standard refers to the LCD controller chip, which accepts data from the ADuC702x and communicates with the LCD screen. HD44780 standard LCD screens are available in numerous formats, the most popular of which are the 16 × 2 and 20 × 2 formats. This application note describes the commands to control the basic functions of the LCD.

INTERFACING WITH AN HD44780 LCD

The data bus that connects the HD44780 to the MicroConveter® can be 8 or 4 bits wide; in this application note, only the case of an 8-bit data bus is examined. In addition to the data bus, three control lines are required; thus, a total of 11 pins are required to interface the LCD to the MicroConverter.

The eight data lines that form the data bus, are referred to as DB0, and DB1 through DB7.

The three control lines are referred to as EN, RS and R/$\overline{W}$. Their functions are as follows:

Enable Line (EN)

This line indicates the start of a transmission of a data byte to the LCD controller. To indicate the start of transmission, this line is brought high; when transmission is complete, the EN line is brought low indicating that the transmission is complete.

Register Select Line (RS)

This line indicates to the LCD controller whether to treat the data byte as a command or as text data to be displayed on the screen. If the RS line is high, the data byte is treated as text to be displayed; if the RS line is low, the data byte is treated as a command.

Read/Write Line (R/$\overline{W}$)

When this line is low, the information on the data bus is written to the LCD controller. If this line is high, the LCD controller can be read. This is used to check the status of the LCD.

As shown in Figure 1, the eight data lines connect to Port 1 of the MicroConverter, and the three control lines connect to Port 0.5, Port 0.7, and Port 2.0.

The source code used to define this interface is given as follows:

```
unsigned char Init_MC08_LCD()
{
GP0DAT = 0xA0000000;   // P0.5 = RS, EN = P0.7
GP1DAT = 0xFF000000;   // P1.[7:0] = DB[0:7]
GP2DAT = 0x01000000;   // P2.0 = R/W

return 0x1;
}
```

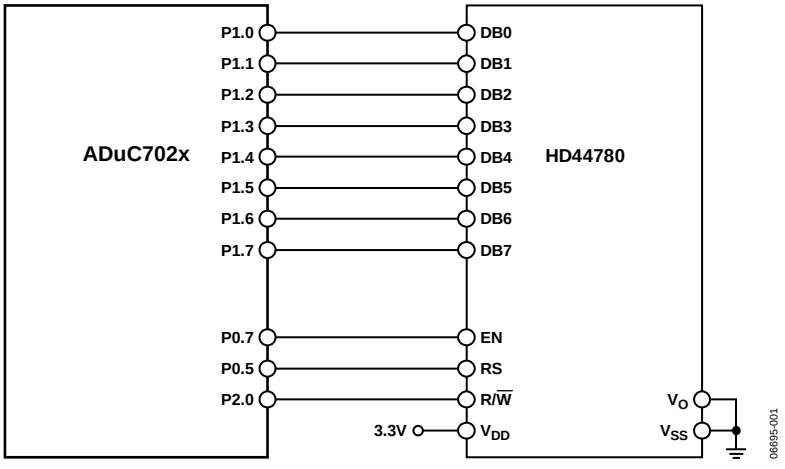

Figure 1. Connecting an HD44780 LCD to an ADuC702x

TABLE OF CONTENTS

Introduction 1
Interfacing with an HD44780 LCD 1
Configuring the LCD Screen 3
Clearing the LCD Screen 4
Setting the Entry Mode 5
Writing Text to the LCD Screen 6
Notes and References 7

CONFIGURING THE LCD SCREEN

To display text to the LCD screen, the LCD screen must be configured first. The configuration defines to the LCD controller the type of LCD screen that is being used, as well as the data bus format and font. The various commands available are listed in detail in any data sheet for an HD44780U-based LCD module.

In the example that follows, the LCD is configured to use an 8-bit data bus and display in a 5 × 7 dot character font. This configuration is achieved by sending 0x38 to the LCD controller through the Select_Function_Set function.

```
unsigned char Select_Function_Set(unsigned char ucFunctionSet)
{
     unsigned long ulFunction = 0;

     delay (10000);
     RdStatus = ReadStatus();
     ulFunction = ucFunctionSet;
     ulFunction = (ulFunction << 16);
     ulFunction |= 0xFF000000;

     GP0DAT = 0xA0000000;          //Clear RS pin(P0.5), clear E pin (P0.7) = R/W)
     GP2DAT = 0x01000000;          //Clear R/W pin   (P2.0 = R/W)
     delay (5);

     GP0SET = 0x800000;            // Set E high
     delay (5);                    // Allow min 800 ns setup time
     GP1DAT = ulFunction;          // Write to register
     delay (5);                    // allow hold time of 500 ns min
     GP0CLR = 0x800000;            // Set E low
     delay (5);

     return 0x1;
}
```

The ReadStatus function is used to read the busy signal output from the HD44780 module.

CLEARING THE LCD SCREEN

Before writing to the screen, it must first be cleared. To do this, use the function, Clear_Display_LCD, as outlined in the following:

```
unsigned char Clear_Display_LCD(unsigned char ucFunctionSet)
{
    unsigned long ulFunction = 0;

    delay (10000);
    RdStatus = ReadStatus();
    ulFunction = ucFunctionSet;
    ulFunction = (ulFunction << 16);
    ulFunction |= 0xFF000000;

    GP0DAT = 0xA0000000;          //Clear RS pin(P0.5), clear E pin (P0.7) = R/W)
    GP2DAT = 0x01000000;          //Clear R/W pin   (P2.0 = R/W)
    delay (5);

    GP0SET = 0x800000;            // Set E high
    delay (5);                    // Allow min 800 ns setup time
    GP1DAT = ulFunction;          // Write to register
    delay (5);                    // Allow hold time of 500 ns min
    GP0CLR = 0x800000;            // Set E low
    delay (2000);                 // Minimum clear time of 1.58 ms

    return 0x1;
}
```

SETTING THE ENTRY MODE

The entry mode must also be set. For this example, the increment is on and there is no shifting. This is achieved by sending 0x6 to the LCD controller through the Set_Entry_Mode function, as follows:

```
unsigned char Set_Entry_Mode(unsigned char ucFunctionSet)
{
     unsigned long ulFunction = 0;

delay (10000);
     RdStatus = ReadStatus();
     ulFunction = ucFunctionSet;
     ulFunction = (ulFunction << 16);
     ulFunction |= 0xFF000000;

     GP0DAT = 0xA0000000;          //Clear RS pin(P0.5), clear E pin (P0.7) = R/W)
     GP2DAT = 0x01000000;          //Clear R/W pin   (P2.0 = R/W)
     delay (5);

     GP0SET = 0x800000;            // Set E high
     delay (5);                    // Allow min 800 ns setup time
     GP1DAT = ulFunction;          // Write to register
     delay (5);                    // Allow hold time of 500 ns min
     GP0CLR = 0x800000;            // Set E low
     delay (5);

     return 0x1;
}
```

WRITING TEXT TO THE LCD SCREEN

In the sample program, the following text is written to the LCD screen:

```
Analog Devices
ADuC7020 LCDdemo
```

To do this, use the Wr_Data_LCD function, after the screen is cleared.

For example, to output the Character A, use its hexadecimal representation, 0x41. All other characters must be represented in their hexadecimal forms and sent to the function to be processed as follows:

```
unsigned char Wr_Data_LCD(unsigned char ucFunctionSet)
{
    unsigned long ulFunction = 0;

    delay (10000);
    RdStatus = ReadStatus();
    ulFunction = ucFunctionSet;
    ulFunction = (ulFunction << 16);
    ulFunction |= 0xFF000000;

    GP0DAT = 0xA0200000;          //Set RS pin(P0.5), clear E pin (P0.7) = R/W)
    GP2DAT = 0x01000000;          //Clear R/W pin   (P2.0 = R/W)
    delay (5);

    GP0SET = 0x800000;            // Set E high
    delay (5);                    // Allow min 800 ns setup time
    GP1DAT = ulFunction;          // Write to register
    delay (5);                    // Allow hold time of 500 ns min
    GP0CLR = 0x800000;            // Set E low
    delay (5);
    GP0CLR = 0x200000;            // Set RS low

    return 0x1;
}
```

NOTES AND REFERENCES

The dot matrix LCD screen used in implementation of all the processes described in this application note was the Samsung S6A0070 driver and controller. See the Samsung website for more information on their products.

For information on specific Analog Devices, Inc. products in the ADuC702x family, select from among the products listed in Table 1.

Table 1. Analog Devices ADuC702x Series

Part No.	No. GPIO Pins	No. ADC Channels	No. 12-Bit DAC Outputs	Temp. Range (°C)
ADuC7020	14	5	4	−40 to +105
ADuC7021	13	8	2	−40 to +85
ADuC7022	13	10	N/A	−40 to +85
ADuC7024	30	10	2	−40 to +105
ADuC7025	30	12	N/A	−40 to +105
ADuC7026	40	12	4	−40 to +125
ADuC7027	40	16	N/A	−40 to +125
ADuC7028	40	16	4	−40 to +125

NOTES

AN06695-0-6/07(0)