

基于HD44780的字符LCD与ADuC702x接口

MicroConverter ADUC702x开发系统

简介

目前市面上有各种各样的LCD显示屏，所幸的是其中许多LCD都符合HD44780U标准。此标准适用于那些接受ADuC702x的数据并与LCD屏幕实现通信的LCD控制器芯片。HD44780标准LCD屏幕可提供多种格式，最常用的是16 × 2和20 × 2格式。本应用笔记说明用于控制LCD基本功能的命令。

与HD44780 LCD接口

用于连接HD44780和MicroConveter®的数据总线可以是8位宽或4位宽，本应用笔记仅讨论8位数据总线。除数据总线外，还需要3条控制线，因此LCD与MicroConverter实现接口总共需要11个引脚。

构成数据总线的8条数据线称为DB0、DB1至DB7。

3条控制线称为EN、RS和R/W。其功能如下所述：

使能线(EN)

此线指示对LCD控制器的数据字节传输是否已开始。当此线被拉高时，指示传输开始。当传输完成时，EN线被拉低，指示传输已完成。

寄存器选择线(RS)

此线告知LCD控制器如何处理数据字节：是将数据字节作为命令还是作为要在屏幕上显示的文本数据。如果RS线为高电平，则将数据字节视为要显示的文本。如果RS线为低电平，则将数据字节视为命令。

读/写线(R/W)

当此线为低电平时，数据总线上的信息被写入LCD控制器。当此线为高电平时，可以读取LCD控制器，以便检查LCD的状态。

如图1所示，8条数据线连接到MicroConverter的端口1，3条控制线连接到端口0.5、端口0.7和端口2.0。

用于定义此接口的源代码如下：

```
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;
}
```

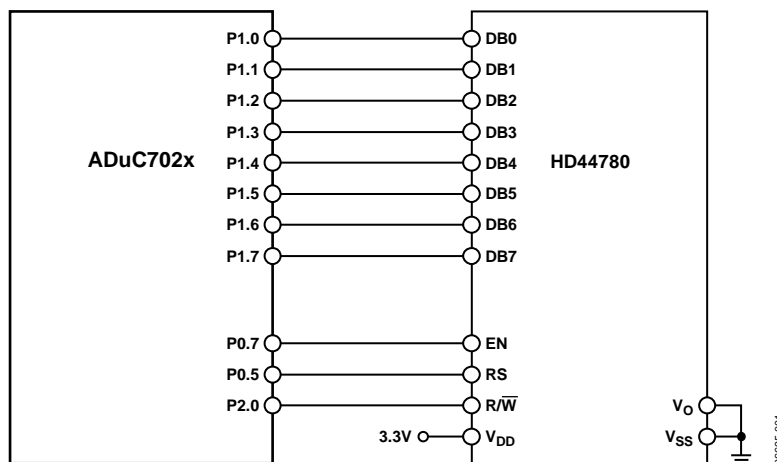


图 1. 连接HD44780 LCD和ADuC702x

目录

简介.....	1	设置输入模式.....	5
与HD44780 LCD接口.....	1	写入文本到LCD屏幕.....	6
配置LCD屏幕.....	3	注释和参考文献.....	7
清除LCD屏幕.....	4		

配置LCD屏幕

要将文本显示到LCD屏幕上，首先必须配置LCD屏幕。配置的目的是向LCD控制器明确所用的LCD屏幕类型、数据总线格式和字体。如需查看详细列出的各种可用命令，请参见HD44780U LCD模块的任何一份数据手册。

在下面的示例中，LCD被配置为使用8位数据总线并显示5 × 7点字符字体。这一配置是通过使用Select_Function_Set函数向LCD控制器发送0x38而实现的。

```
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/ $\bar{W}$ 
    GP2DAT = 0x01000000;           //Clear R/ $\bar{W}$  pin (P2.0 = R/ $\bar{W}$ )
    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;
}
```

ReadStatus函数用于从HD44780模块读取繁忙信号输出。

清除LCD屏幕

写入屏幕之前，必须清除屏幕。为此，请按照下面所示使用函数Clear_Display_LCD。

```
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/ $\bar{W}$ )
    GP2DAT = 0x01000000;           //Clear R/ $\bar{W}$  pin (P2.0 = R/ $\bar{W}$ )
    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;
}
```

设置输入模式

还必须设置输入模式。在下面的示例中，递增功能开启，无变换（Shift）功能。这是通过使用Set_Entry_Mode函数向LCD控制器发送0x6而实现的，如下所示：

```
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/ $\bar{W}$ )
    GP2DAT = 0x01000000;           //Clear R/ $\bar{W}$  pin (P2.0 = R/ $\bar{W}$ )
    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;
}
```

写入文本到LCD屏幕

在示例程序中，下列文本被写入LCD屏幕：

```
Analog Devices  
ADuC7020 LCDdemo
```

为此，请在清除屏幕后使用Wr_Data_LCD函数。

例如，若要输出字符A，应使用其十六进制表示形式0x41。所有其它字符都必须用其十六进制形式表示，并发送到函数进行处理，如下所示：

```
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/ $\bar{W}$ )  
    GP2DAT = 0x01000000;         //Clear R/ $\bar{W}$  pin (P2.0 = R/ $\bar{W}$ )  
    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;  
}
```

注释和参考文献

用于实现本应用笔记所述全部过程的点阵LCD屏幕为 Samsung S6A0070驱动器和控制器。有关其产品的更多信息，请访问Samsung网站。

有关ADI公司ADuC702x系列特定产品的信息，请从表1所列产品中选择相应的产品。

表1. ADI公司ADuC702x系列产品

产品型号	GPIO 引脚数	ADC 通道数	12位 DAC输出数	温度范围(°C)
ADuC7020	14	5	4	-40至+105
ADuC7021	13	8	2	-40至+85
ADuC7022	13	10	N/A	-40至+85
ADuC7024	30	10	2	-40至+105
ADuC7025	30	12	N/A	-40至+105
ADuC7026	40	12	4	-40至+125
ADuC7027	40	16	N/A	-40至+125
ADuC7028	40	16	4	-40至+125

注释