Keywords: DS2408, LCD

Jul 20, 2004

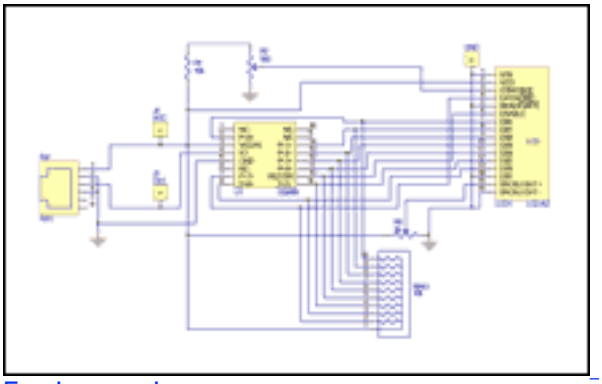
APPLICATION NOTE 3286

1-Wire 8-bit LCD Reference Design

This application note describes how a DS2408 1-Wire 8-Channel Addressable Switch can be used to display characters on an LCD. The code is written using the 1-Wire API for Java. A diagram shows how to hook the DS2408 to any LCD using a Hitachi-44780-based LCD controller chip.

General Overview

This application note demonstrates creating a 1-Wire® 8-bit LCD. The code to drive the DS2408 LCD display is also provided. The LCD used was the Data Vision Model NO: DV-16252-S1FBLY, however any LCD with this Hitachi 44780 based LCD controller chip can be used.



For Larger Image

Figure 1. 1-Wire LCD schematic.

Main Display Program

The code examples provided are written using the 1-Wire API for Java (http://www.ibutton.com/software/1wire/1wire_api.html). The main program, displayed in **Figure 2**, accesses the default adapter to talk to the DS2408 and initializes the LCD. Depending on the size of the LCD used the numCharsPerLine, secondStart and thirdStart may need to be updated. The fixed string '0123456789' is printed out to the LCD in the loop after the initialization is complete.

Figure 2. 1-Wire LCD main example code

```
import com.dalsemi.onewire.*;
import com.dalsemi.onewire.adapter.*;
import com.dalsemi.onewire.container.*;
import com.dalsemi.onewire.utils.*;


public class DisplayLCD
```

```
{
   static DSPortAdapter adapter;
   static byte[] address29 = new byte[]{0,0,0,0,0,0,0,0};
   static OneWireContainer29 owc29;

   public static void main(String[] args)
      throws Exception
   {
      adapter = OneWireAccessProvider.getDefaultAdapter();
      adapter.targetFamily(0x29);
      if(!adapter.findFirstDevice())
         return;
      owc29 = (OneWireContainer29)adapter.getFirstDeviceContainer();
      address29 = owc29.getAddress();

      System.out.println("Initing");
      doOverdrive();

      fixResetMode();

      initLCD();
      System.out.println("Done Initing");
      byte[] chars = "0123456789".getBytes();

      int i = 0;
      int numCharsPerLine = 20;
      int secondStart = numCharsPerLine - chars.length;
      int thirdStart = chars.length - secondStart;
      while(true)
      {
         System.out.print(".");
         returnHome();
         writeChars(chars, i, chars.length - i);
         if(i>thirdStart)
         {
            writeChars(chars, 0, chars.length);
            writeChars(chars, 0, i - thirdStart);
         }
         else
         {
            writeChars(chars, 0, secondStart + i);
         }
         i = (i+1)%chars.length;
      }
   }
}
```

Figure 3. Initializing LCD example code

```
   public static void initLCD()
   {
      setLatchOutput(0x30);
      waitMS(100);
      // init
      setLatchOutput(0x38);
      waitMS(10);

      // Enable Display, Cursor, and Blinking
      setLatchOutput(0x0F);
```

```
      // Entry-mode: auto-increment, no shift
      setLatchOutput(0x06);

      clearDisplay();
   }
```

Initializing LCD

In **Figure 3**, the code to initial the Data Vision Model NO: DV-16252-S1FBLY is shown. This will change depending on the model of LCD being used.

Setting RSTZ Pin Mode Control

In **Figure 4**, the RSTZ pin is being configured to the STRB output mode. This is connected to the LCD enable signal for writing.

Figure 4. Setting RSTZ pin example code

```
   public static void fixResetMode()
   {
      try
      {
         adapter.beginExclusive(true);
         byte[] register = owc29.readRegister();
         owc29.setResetMode(register, false);
         owc29.writeRegister(register);
      }
      catch(Exception e)
      {;}
      finally
      {
         adapter.endExclusive();
      }
   }
```

Display Characters on LCD

In **Figure 5**, the code to display the characters to the LCD is provided. The DS2408's P7 is being used to select whether text or commands are being sent. The method setLatchOutput is sending the data through P0-P6 of the DS2408.

Figure 5. Display characters on LCD example code

```
   public static void writeString(String message)
      throws Exception
   {
      byte[] chars = message.getBytes();
      writeChars(chars, 0, chars.length);
   }

   public static void writeChars(byte[] bytesToWrite, int offset, int length)
   {
      length += offset;
```

```
   for(int i=offset; i<length; i++)
   {
      if((bytesToWrite[i]&0x80)==0x080)
      {
         System.out.println("Unprintable character: " + (char)bytesToWrite[i]);
      }
      else
      {
         setLatchOutput(bytesToWrite[i]|0x80);
      }
   }
}

private static void setLatchOutput(int output)
{
   try
   {
      adapter.beginExclusive(true);
      adapter.assertSelect(address29);

      byte[] buffer = new byte[5];

      // channel access write command
      buffer[0] = 0x5A;
      buffer[1] = (byte)output;
      buffer[2] = (byte)(~output);
      buffer[3] = (byte)0x0FF;
      buffer[4] = (byte)0x0FF;

      adapter.dataBlock(buffer, 0, 5);
   }
   catch(Exception e)
   {;}
   finally
   {
      adapter.endExclusive();
   }
}
```

Clearing and Positioning

In **Figure 6**, the methods for clearing the LCD and positioning the cursor in the starting position are shown.

Figure 6. Clearing and positioning

```
public static void returnHome()
{
   setLatchOutput(0x02);
   waitMS(2);
}

public static void clearDisplay()
{
   // Clear Display
   setLatchOutput(0x01);
      waitMS(2);
}
```

Additional Resource Methods

In **Figure 7**, additional resource methods that were used are displayed.

Figure 7. Resource methods

```
public static void doOverdrive()
{
   try
   {
      adapter.reset();
      adapter.setSpeed(adapter.SPEED_REGULAR);
      adapter.reset();
      adapter.putByte(0x3C); //overdrive skip rom
      adapter.setSpeed(adapter.SPEED_OVERDRIVE);
   }
   catch(Exception e)
   {;}
}

public static void waitMS(long timeToWait)
{
   try
   {
      Thread.sleep(timeToWait);
   }
   catch(InterruptedException ie)
   {;}
}
```

Application Note 3286: http://www.maxim-ic.com/an3286

More Information
For technical questions and support: http://www.maxim-ic.com/support
For samples: http://www.maxim-ic.com/samples
Other questions and comments: http://www.maxim-ic.com/contact

Related Parts
DS2408: QuickView -- Full (PDF) Data Sheet -- Free Samples

AN3286, AN 3286, APP3286, Appnote3286, Appnote 3286