Keywords: DS8007,smart card,DS5002,DS5250,microcontroller,secure microcontroller,uC,multiprotocol,ISO 7816,EMV,Integrated Circuit Card, IC card,POS terminal,banking terminal,ATM,payment terminal,PIN pad,access control,pay tv,set top box,STB

APPLICATION NOTE 4036

Getting Started with the DS8007 Evaluation Kit

May 22, 2007

Abstract: This article describes the steps necessary to quickly begin using the evaluation (EV) kit for the DS8007, dual smart card interface. The article explains how to install and configure the software, configure the hardware, and create and load an application that can be executed by the on-board DS5002FP microcontroller. A simple "Hello World" example is provided, as well as a number of functions in C source code that will demonstrate use of the on-board LCD module.

Overview

The DS8007 evaluation (EV) kit provides a convenient, proven platform for evaluating the DS8007 dual smart card interface. The DS8007 is a mixed-signal peripheral that manages all the interface details between a microcontroller and two, independent smart cards.

The DS8007 EV kit contains a DS8007 and a DS5002FP secure microcontroller that can be programmed to provide a complete smart card interface. It also contains two smart card sockets: one the size of a standard credit/payment card and one the size of a Subscriber Identification Module (SIM) device. The DS8007 kit board is shown in **Figure 1**. This article describes how to get started quickly using this EV kit, and how to begin developing and running application code.

Figure 1. DS8007 EV kit board.

The DS8007 provides all electrical signals necessary to physically interface a microcontroller with two individual smart cards. The device also contains a dedicated internal sequencer that controls automatic card activation and deactivation, and an ISO UART for data communication. In addition, the DS8007 contains charge pumps and voltage regulators that allow it to operate from a 2.7V to 6.0V supply voltage and produce two independent smart card supply voltages, either of which can be 1.8V, 3.0V, or 5.0V. Communication with a microcontroller is provided by a standard parallel 8-bit bus in a multiplexed or a nonmultiplexed configuration.

Software Setup

For this application note, a Dallas Semiconductor Microcontroller Tool Kit (MTK) and Keil's PK51 Professional Developer's Tool Kit software should be installed and operational on your PC. Both software packages are included with the DS8007 EV kit. The following sections describe the steps for installing these applications. If either or both of these applications are already installed on your PC, the section(s) describing their installation can be ignored.

Installing the Microcontroller Tool Kit

1. Insert the DS8007 EV kit install CD into your computer's CD-ROM drive. The CD will autoboot and display the DS8007 EV kit welcome screen. If the CD-ROM does not autoboot, browse the CD root folder, and double-click on the index.html file.
2. Click the "Install MTK" button, and click the "Run" button to begin installation.
3. During the MTK installation, select the default settings. **Note:** security warnings may appear, depending on your web browser security settings and your version of Windows®. If this does occur, simply acknowledge the warning, and proceed with the installation.

Installing the Keil Professional Developer's Tool Kit

1. To install Keil's PK51 Professional Developer's Tool Kit, insert the Keil Microcontroller Development Tools CD (included with the DS8007 EV kit) into your computer's CD-ROM drive. The CD will autoboot and display the main menu. If the CD-ROM does not autoboot, browse the CD root folder, and double-click on the setup.exe file.
2. At the main menu, select "Install Evaluation Software," and then select the "C51 Compiler (Eval Tools)."
3. Follow the on-screen instructions to install the Keil tools on your PC.

Hardware Setup

Configuring the DS8007 EV kit board requires that you properly set the jumpers, connect its serial port to the PC, and connect the power supply. These steps are detailed below.

1. Connect jumpers 1 through 13 as shown in **Figure 2** and further described in **Table 1**.
2. Connect one end of the supplied serial cable to the board's DB-9 connector (J6) and the other end to the COM port of your PC.
3. Connect the 5V, regulated (±5%), 300mA, 2.5mm center positive, power supply included with the EV kit to the board's J7 power connector.

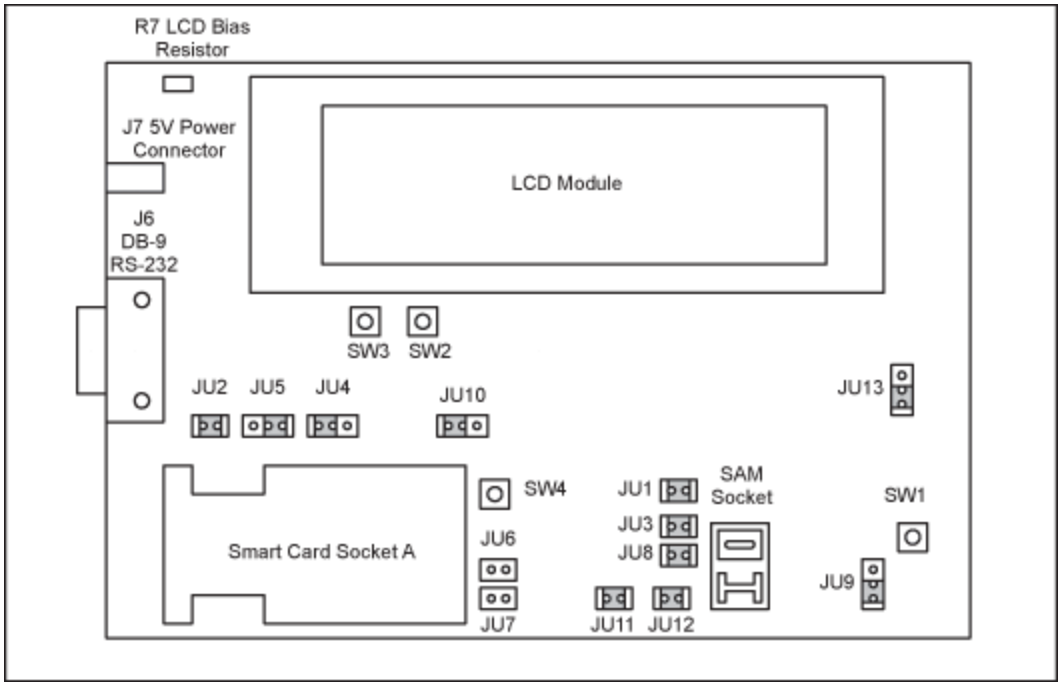

Figure 2. DS8007 board jumper locations.

Table 1. DS8007 Board Jumper Settings

Jumper	Installed	Description
JU1	Installed	Board's DVDD connected to DS8007's digital V_{DD}
JU2	Installed	DS8007's V_{CCA} connected to smart card socket 1, pin C1
JU3	Installed	Board's AVDD connected to DS8007's analog supply, V_{DDA}
JU4	Installed: Connect pins 1 and 2	Board's AVDD connected to smart card socket 1, pin S2
JU5	Installed: Connect pins 2 and 3	Board's AVDD connected to smart card socket 1 pin, S1 through 10K
JU6	Not Installed	
JU7	Not Installed	
JU8	Installed	DS8007's V_{CCB} connected to SAM socket, pin C1
JU9	Installed: Connect pins 2 and 3	DS8007's PRESB connected to GND (through 10K)
JU10	Installed: Connect pins 2 and 3	DS5002FP's nPROG connected to RS-232 DTR0/DSR0
JU11	Installed	Board's 5V supply connected to board's AVDD
JU12	Installed	Board's 5V supply connected to board's DVDD

JU13	Installed: Connect pins 2 and 3	DS8007's INTAUX pin connected to GND (through 10K)

Creating and Loading an Executable File

The next step is to create a .HEX file that can be downloaded into the board's memory and can then be executed by the DS5002FP. Creating this file involves configuring the Keil µVision® project file to create the proper loadable file, and then downloading it to the board's nonvolatile memory.

Creating the µVision Project

After the Keil C51 Development Tools are installed on the PC, start the µVision program by clicking on the icon created. We will now create a project and C source files for the demonstration program by following these steps:

1. Under the Project heading, select "New Project," and type DS8007-1 for the project name.
2. When the "Select Device for Target" appears, select the Dallas Semiconductor group and the DS5002FP from the device selection list.
3. Click the "Yes" button when the "Copy Standard 8051 Startup Code to Project Folder and Add File to Project" message appears.
4. Create the source file by clicking "File" then "New," and typing the following information. If you prefer downloading this file, it is available for download (ZIP, 11kB).

```
// File DS8007-1.C

#include <REG5000.H>                    // special function register
declarations
//                                      // for the DS5000/5002
#include <stdio.h>                      // prototype declarations for I/O
functions

// Main C function.  Program execution starts here.
void main (void) {

// Set up the serial port for 38400 baud at 14.7MHz.
        TMOD=0x21;                      // Timer 1: 8-bit auto-reload
from TH1, Timer 0: 16-bit
        TH1 = 0xFE;                     // Set timer reload value for
38,400 baud
        PCON|=0x80;                     // Turn on baud rate doubler
SMOD_0
        TCON=0x50;                      // Enable timers 0 and 1
        SCON=0x50;                      // 10-bit async, enabled
        TI=1;                           // Set TI to send first character
        TR1 = 1;                        // Start timer 1

// Start main program
        printf ("\nHello DS8007 World!\n\n");   // Output message
        while (1) ;                     // End program by looping here.
}
```

5. After this text has been entered, save the file by clicking "File" then "Save As" and typing in "DS8007-1.c" as the file name.
6. Add this file to the project list by right-clicking on Target 1 in the Project Workspace window, and then click the "Manage Components" option. Under the Project Components tab, click the "Add Files" button, and then enter the file name (DS8007-1.c) in the file name area. Click the "Add" button and then click the "Close" button. Click the "OK" button to close the Components window.

You will see that this file has been added to Source Group 1.

7. Configure the project options by right clicking on Target 1 in the Project Workspace window, and select the "Options for Target 'Target 1'." Click on the Target tab, and type 14.7 into the Xtal box. Select "Small: variables in DATA" for the Memory Model, select "Large: 64K program" for the Code ROM Size, and select "None" for the Operating system as shown in **Figure 3** below. All other selections in this window can be left in their default state.
8. On the Output tab, check "Create Executable" (if it is not already), and be sure that the "Create HEX File" box is checked. Select the HEX Format of HEX-80 from the dropdown selector box, as shown in **Figure 4**. All other selections in this window should remain in their default state. Click "OK" to close the Options for Target 'Target 1' window.
9. Save the Project file by clicking "File" and then "Save All."
10. Create the executable file by clicking the "Project" heading and select "Rebuild All Target Files." The build window at the bottom of the screen should indicate that there were 0 Error(s), 0 Warning(s). If not, locate the error(s) and make the necessary correction(s). Repeat this step until no errors are reported.

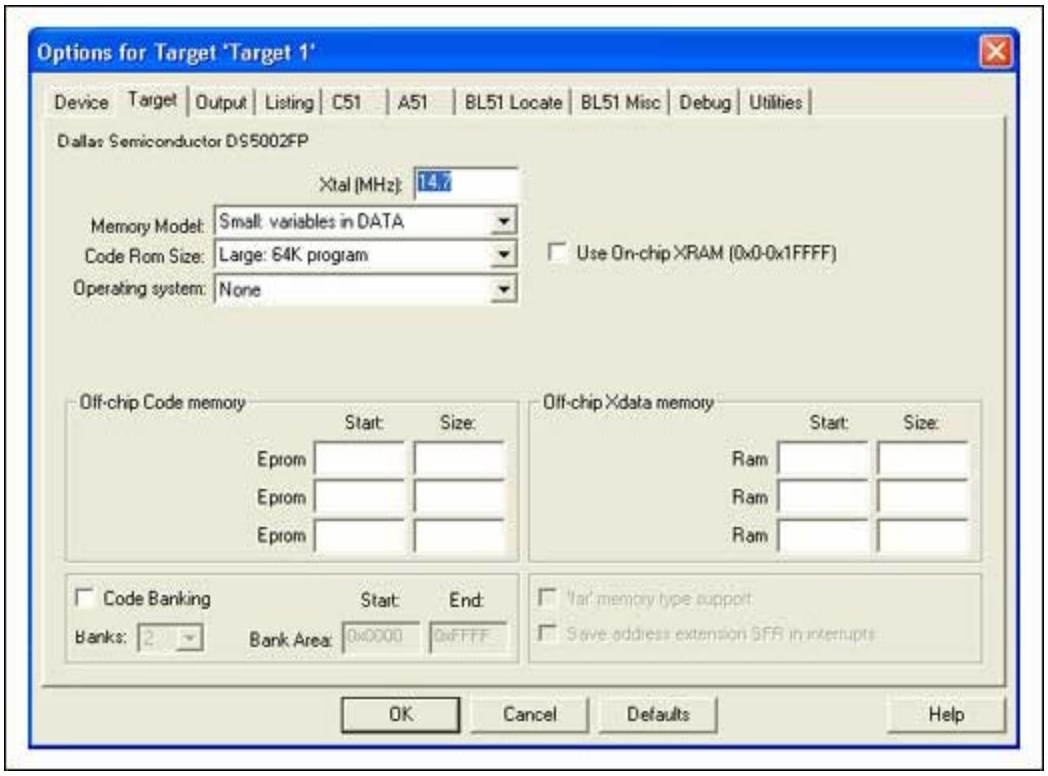

Figure 3. Project options target tab settings.

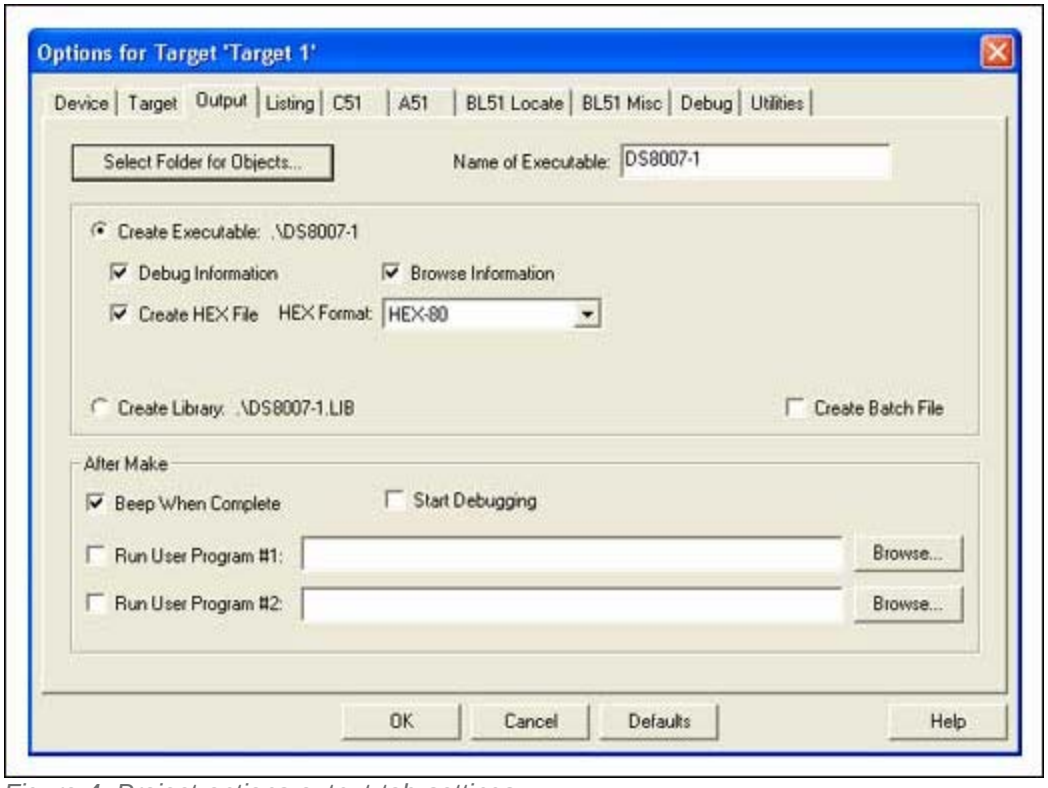

Figure 4. Project options output tab settings.

Loading the Executable File

After following the above steps to configure the hardware, the DS8007 board will be connected to your PC's COM port. The processor should now be configured so that it understands how to access the available on-board memory. This task is accomplished by writing the appropriate values to the processor's MCON and RPCTL registers, as described in the following steps.

1. Locate the MTK program on your PC and start it.
2. Select "DS5002FP" in the Select Device window (**Figure 5**) and click "OK."
3. Click "Options" and then click "Configure Serial Port." Select the Port, e.g., COM1, of the PC to which the DS8007 board is connected. Then select the "Speed 38400" from the drop-down menus, and click "OK."
4. Click "Target," and click "Open COM1 at 38400 baud."
5. Make sure that the board's serial port is connected to the PC and that 5V power has been applied. Click "Target" and "Connect to Loader." If successfully connected, the DS5002FP will display its Boot Loader Banner shown in **Figure 6**.
6. At the > prompt, type in the following commands to configure the MCON and RPCTL registers:

```
> W MCON 0A

> W RPCTL 01

> R

MCON:0A RPCTL:81 MSL:01       ← Microcontroller's response to read
register command
```

If you receive an "E: LOCKED" error message when entering a boot-loader command, the part has its security lock set. To unlock the part, simply enter the unlock command (U) at the loader prompt, >. This action will destroy the information stored in the processor's memory, but when the part is unlocked, the register commands above can be entered. After issuing the Read Register command (> R), the DS5002FP reports the status of registers MCON (0A) and RPCTL (81), as shown. The DS5002FP also shows the status of the MSL bit (see Secure Microcontroller User's Guide for details), which in this case is 01.

The DS8007 board is now ready to load the program file created with the Keil development tools.

1. Click "File" and then select "Load SRAM." Select the file DS8007-1.HEX and then click "OPEN." The DS5002FP processor will respond with:

   ```
   > Loading File ...\DS8007-1.hex
   ```

   and will indicate load progress by displaying a series of dots.
2. When finished, the processor will report "Load complete." and the program can be executed by clicking "Target" and then clicking "Disconnect from Loader."

Immediately after the bootstrap loader is disconnected, the processor begins running the program. The MTK screen will show the program's output, "Hello DS8007 World!" Success!

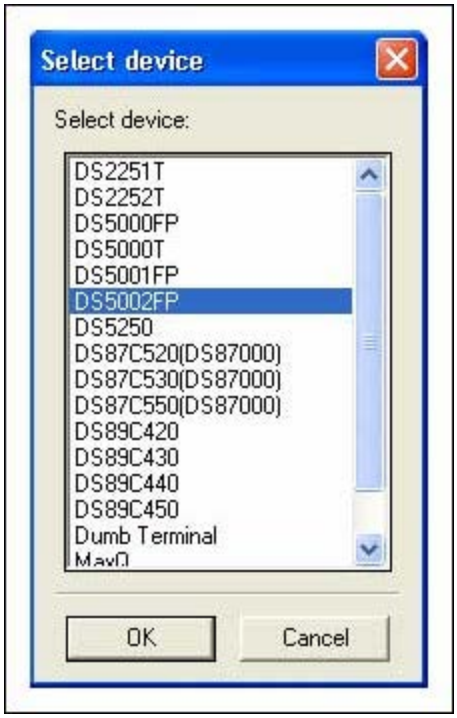

Figure 5. MTK device selection.

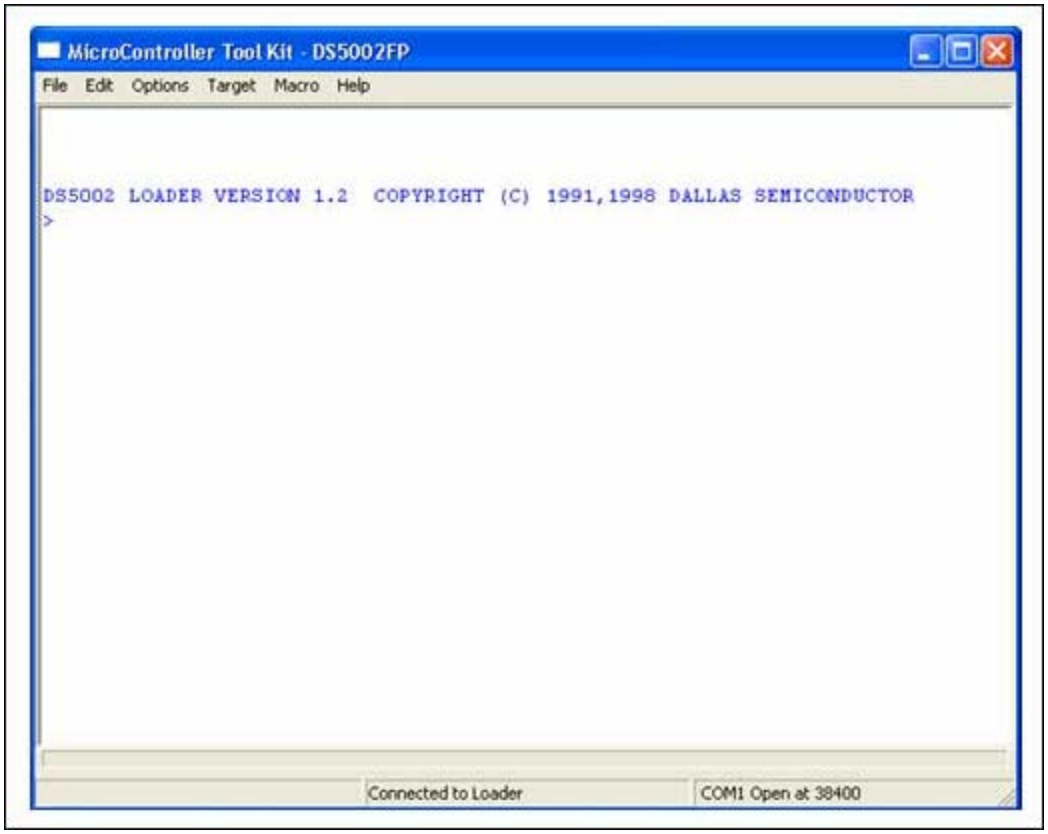

Figure 6. DS5002FP boot loader banner.

EV Kit LCD Module

To demonstrate one of the DS8007 EV kit's very useful features, we will now create an application program that will display a message on the 2-line, 20-character Liquid Crystal Display (LCD). Before beginning, however, we must set the LCD contrast adjustment so that it displays the message appropriately when the program is executed. First, apply power to the board; the LCD should be blank. Locate variable resistor R7 on the board (see Figure 2 above), and using a small screwdriver, adjust the resistor until individual 5 x 7 dot blocks appear in the character positions of the display. Slowly adjust R7 until the blocks just disappear. This sets the contrast of the LCD so that the characters will be visible, but spaces will not.

Follow steps 1 through 10 above (*Create the µVision Project*) to create a new project and executable file. Use DS8007-2 for the project file name and DS8007-2.c for the C source-code file name. The source code for this program is given below in **Appendix A**. If you prefer not to type in this information, the files are included in a .ZIP file located at the FTP web site mentioned above.

Once the HEX file has been created, follow the steps above (*Loading the Executable File*) to load it into the board's memory. After disconnecting from the loader, the LCD will display the two-line message.

Conclusion

The DS8007 EV kit provides a convenient, proven platform for evaluating the DS8007 Dual Smart Card

Interface. This kit and the Keil C development tools will simplify the development of your smart-card applications.

Appendix A. LCD Demonstration Code

```
// file DS8007-2.c
//

#include <REG5000.H>                        // special function register
declarations
//                                           // for the DS5000/5002 #include
<string.h>
#include "LCD_Funct.h"



        int tmp = 0;
        uint8_t LCD_Str[42];


void main(void)
{

        // Initialize LCD Module, and display 2-line message.
        LCD_Init();
        strcpy(LCD_Str, "DS8007 Dual");                 // Create first line
of LCD text
        tmp = LCD_Curpos(1,5);                          // Position cursor
line 1 char 5
        if (tmp == 0) LCD_WRStr(LCD_Str);               // Write string to
LCD

        strcpy(LCD_Str, "Smart Card Interface");        // Create 2nd line of
LCD text
        tmp = LCD_Curpos(2,1);                          // Position cursor on
line 2, char 1
        if (tmp == 0) LCD_WRStr(LCD_Str);               // Write string to
LCD

        tmp = LCD_Curpos(1,20);                         // Return cursor to
line 1 char 20

        while (1);                                       // Loop here to end
program
}


void LCD_Init()
{
        LCD_Delay();                    // Delay to ensure that LCD power-up
is complete

        LCD_WRCmd(FnctSet);             // Write FnctSet instruction
        LCD_WRCmd(DispCnt);             // Write Display Control instruction
        LCD_WRCmd(DispClear);           // Write Display Clear instruction
        LCD_WRCmd(EntryMode);           // Write Entry Mode instruction
}

// Write a string to the 2 x 20 LCD module
void LCD_WRStr(uint8_t LCD_Str[])
{
        int     i = 0;

        while ((LCD_Str[i] != '\0')     && (i < 20))
```

```
        {
                LCD_WRChr(LCD_Str[i]) ; // Write first 20 characters
                i++;
        }

        if  (LCD_Str[i] != '\0')
            LCD_WRCmd(Line2Ad);         // Set CGRAM address to first char
2nd line

        while ((LCD_Str[i] != '\0')     && (i < 40))
        {
                LCD_WRChr(LCD_Str[i]) ;
                i++;
        }
}

// Write a single character to the 2 x 20 LCD module
void LCD_WRChr(uint8_t LCD_Chr)
{
        LCD_Busy();                     // Wait until the LCD is not busy
        LCD_RS = 1;                     // Set RS high for character write
        LCD_Dat = LCD_Chr;              // Output character
        LCD_EN = 1;                     // Set enable high
        LCD_EN = 0;                     // Clear enable
        LCD_RS = 0;
        LCD_Dat = 0xFF;                 // Re-establish Port Pins as inputs
}

// Write a command to the 2 x 20 LCD module
void LCD_WRCmd(uint8_t LCD_Cmd)
{
        LCD_EN = 0;                     // Make sure that the LCD enable is
low
        LCD_RS = 0;                     // Set LCD register select and R/W
lines
        LCD_RW = 0;
        LCD_Busy();                     // Make sure that the display is not
busy

        LCD_Dat = LCD_Cmd;              // Output instruction
        LCD_EN = 1;                     // Set enable high
        LCD_EN = 0;                     // Clear enable
        LCD_Dat = 0xFF;                 // Re-establish Port Pins as inputs
}

// Set the cursor position to a specific location
int LCD_Curpos(uint8_t VPos, uint8_t HPos)
{
        int rtn;
        uint8_t Addr, Cmd;

        // Check input range 1..2 line, 1..20 characters per line
        if (((VPos == 0) || (VPos > 2)) || ((HPos == 0) || (HPos > 20)))
                rtn = -1;
        else
        {
                if(VPos == 2) Addr = (0x40 + (HPos - 1));
                else Addr = HPos - 1;
                rtn = 0;
                Cmd = Addr | 0x80;
                LCD_WRCmd(Cmd);
        }
        return(rtn);
}

// Test the LCD Module to see if it is Busy (loop until
// not busy)
```

```
void LCD_Busy()
{
        uint8_t LCD_Stat = LCD_Dat;     // Get byte containing status

        while (LCD_Stat & 0x80){        // Wait for busy flag to clear
                LCD_RW = 1;             // LCD RW needs to be high
                LCD_EN = 1;             // Strobe enable signal
                LCD_Stat = LCD_Dat;     // Read staus
                LCD_EN = 0;
                LCD_RW = 0;
        }
}

// Time delay for 2 x 20 LCD module (approximately 50ms)
void LCD_Delay()
{
        uint8_t i, j ;

        for (i=0;i!=100;i++)
        {
                for (j=0;j!=153;j++);
        }
}
```


Related Parts		
DS5002FP	Secure Microprocessor Chip	Free Samples
DS5250	High-Speed Secure Microcontroller	
DS8007	Multiprotocol Dual Smart Card Interface	Free Samples
DS8007-KIT	DS8007 EMV Evaluation Kit	

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 4036: http://www.maximintegrated.com/an4036
APPLICATION NOTE 4036, AN4036, AN 4036, APP4036, Appnote4036, Appnote 4036