Serial LCD

This section shows how to connect a serial LCD to the CSM-12C32 module and provide the C-codes for initializing the SCI port and for sending ASCII characters to the LCD.

The selected serial LCD is Parallax 4X20 serial LCD item code 27979. Pictures, ordering information and web link of the datasheet are shown below.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Part Number</th>
<th>Weblink for the part</th>
<th>Description</th>
<th>Unit Price</th>
</tr>
</thead>
</table>

The baud rate supported by the LCD are 2400, 9600, and 19200. To select the desired baud rate, operate the two switches on the back of the LCD as shown in the figure below.

Hardware interfacing to the Freescale 9S12C32 MCU
The wiring diagram of the serial LCD to the microcontroller are as follows:
Software development

The following instructions for using the serial LCD were taken from the datasheet.

**Displaying Text**
To display a character of text on the Serial LCD, simply send the ASCII code of that character to the Serial LCD over the serial port at the correct baud rate. When a character is received, the Serial LCD displays that character at the current cursor position and then moves the cursor one position to the right.

Once a full line of characters were sent to the LCD, the cursor automatically wraps around to the leftmost position of the second line. The text will wrap like this at the end of every line, with the end of the bottom line wrapping back around to the top line of the LCD.

**Moving the Cursor**
There are a few different ways to move the cursor on the Serial LCD display. After each character sent, the cursor automatically moves over one position. Along with this, there is a standard set of cursor move commands including Backspace, Carriage Return, and Line Feed. The Backspace/Left command (Dec 8) moves the cursor one place to the left. The Right command (Dec 9) moves the cursor one place to the right. These can be useful for moving the cursor around to overwrite existing text. These commands wrap to the next line of the display, if necessary.

The LineFeed command (Dec 10) moves the cursor to the next line of the display without changing the horizontal position of the cursor.

The Carriage Return command (Dec 13) also moves the cursor to the next line, but it moves the cursor to the leftmost position on that line as well.
The Form Feed command (Dec 12) clears the entire display and moves the cursor to the leftmost position on line 0, just like when you first turn on the display. You will need to pause for 5mS in your code after sending the Form Feed command to give the Serial LCD time to clear the display. Except for Form Feed, none of these move commands affects the characters on the display.

There are also direct move commands that can be used to move the cursor to any position on the display with a single command. Consult the datasheet for these commands.

**Controlling the Display**

The display-off command (Dec 21) turns off the display so that all of the characters disappear. The characters aren’t erased from the display, though, and one can even keep writing new characters to the display when it is turned off. A trick to make a lot of text show up all at once, even at a slow baud rate, is to turn off the display and then send all of your text. Then, when the display is turned on again, all of the text appears instantly. The display-on commands (DEC 22 to 25) turn the display back on and also control whether you want to display the cursor and/or make the cursor character blink. With models 27979, one can also control the backlight of the display. The backlight lights up the display so that it is easier to see in the dark. There are commands to turn the backlight on (Dec 17) and off (Dec 18).

**Command Set**

A sample of the command set is shown in the table below. Commands marked as N/A are invalid and are ignored. The lines of the LCD display are numbered starting from 0, with line 0 being the top line. The character positions on each line are numbered starting from 0, with position 0 being the leftmost position on the line.
The following program sends a greeting message to the serial LCD.

```c
#include <hidef.h>      /* common defines and macros */
#include <mc9s12c32.h>

#define SetUp  0x11, 0x16, 0x00 //backlight on, display on, cursor off, no blink
#define FormFeed  0x0C, 0x00 //Clear screen, move cursor to line 0 position 0
#define Line0 "EEL4744C "
#define Line1 "Microcontroller " //cursor to line 1 pos 1
#define Line2 "Applications " //cursor to line 2 pos 2
#define Line3 "Dr. Choi" //cursor to line 3 pos 3

#pragma LINK_INFO DERIVATIVE "mc9s12c32"
```
// Declares functions

void Init_sc0(void);
void putchar(char d);
void putstring(char *d);
void waitms(int ms); // for LCD, no need for Hyperterminal
void dly1ms(void);

#define TC1MS 2652

void main(void) {
  Init_sc0();
  putstring(SetUp);
  putstring(FormFeed);
  putstring(Line0);
  putstring(Line1);
  putstring(Line2);
  putstring(Line3);
}

// Initialize SCI
void Init_sc0(void) {
  SCIBDL=52;
  SCICR1=0x04;
  SCICR2=0x0C;
}

// Send a character to SCI
void putchar(char d) {
  while(!(SCISR1&0x80)) {};
  SCIDRL = d;
  // waitms(10); // delay needed for LCD display
  return;
}

// Send a string to SCI
void putstring(char *d) {
  int i = 0;
  while (d[i] != 0x00) {
    putchar(d[i]);
    i++;
  }
}

// 1 millisecond delay function
void dly1ms() {
  int i = TC1MS;
  while(i) {
    i--;
  }
  return;
}

// ms millisecond delay function
void waitms(int ms) {
  int i;
  for(i=ms;i>0;i--){
    dly1ms();
  }
}