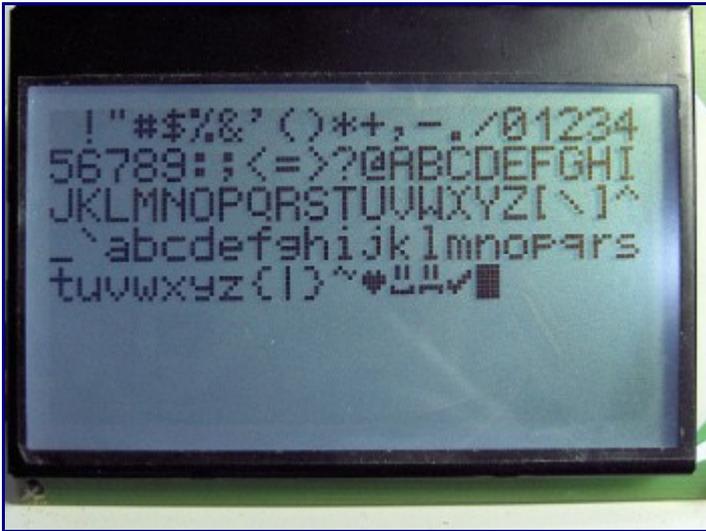


Graphic 128×64 LCD Working

2010-04-07 14:04:55 by Chris



The Electronic Access Module LCD is the NHD-C12864EZ-FSW-FTW-3V3P ([datasheet](#)) from Newhaven Display. I described its features in an [earlier post](#). This display is an unusual one compared to most of Newhaven's units in that it is a Chip-On-Glass LCD mounted on a PCB and has a simplex (one-way) serial interface.

Figure 1 shows the screen working in the EAM. It is displaying the printable ASCII characters (32-127) along with some extra characters that I created. Since taking this image I have created more characters and four small images to be used as notification icons.

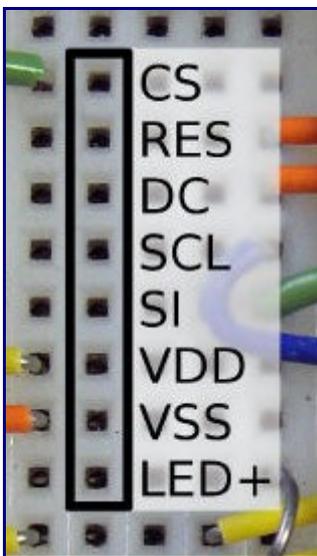


Figure 2 shows the LCD connector pinout (click thumbnail for complete view):

1. **CS** (active low) is the Chip Select pin and enables communication with the LCD. This pin needs to be set by the microcontroller during all LCD communication since SPI will also be used for communicating with the NIC submodule. The NIC is not connected yet though so for the time being I have connected it to ground.
2. **RES** (active low) is the Reset pin and is connected directly to PB3 on the AVR.
3. **DC** notifies the LCD of either incoming data (high) or command (low) communication. It is connected directly to Port B4 on the AVR.
4. **SCL** is the serial clock line and is connected directly to the SCL pin (Port B7) on the AVR.
5. **SI** is for serial data and is connected directly to the MOSI pin (Port B5) on the AVR.
6. **VDD** is the 3.3V supply for the display.
7. **VSS** is ground.

8. **LED+** is the 5V LED backlight supply. It is connected to 5V through a resistor to limit current to 18mA (max 25mA).

Here is the AVR C firmware that generates the above output: [eam_lcd_v0.1.zip](#)

Here is an excerpt from lcd.c that demonstrates sending commands and data to the LCD through the SPI bus:

```
void lcd_comm(unsigned char j)
{
    //delay(5);
    LCD_COMM_SEL;
    bit_clr(SPSR, 7); // Clear SPI interrupt flag
    //delay(5);
    SPDR = j; // Write byte out
    while(!(bit_get(SPSR,7))) { } // Wait to finish
    return;
}

void lcd_data(unsigned char j)
{
    //delay(5);
    LCD_DATA_SEL;
    bit_clr(SPSR, 7); // Clear SPI interrupt flag
    //delay(5);
    SPDR = j; // Write byte out
    while(!(bit_get(SPSR,7))) { } // Wait to finish
    return;
}

// Column: 0-127
void lcd_col(unsigned char column)
{
    // Byte H and L
    lcd_comm(0b00010000 | (column >> 4));
    lcd_comm(column & 0b00001111);
    return;
}

// Row (Actually, PAGE of 8 rows): 0-7
void lcd_row(unsigned char row)
{
    // Byte H and L
    lcd_comm(0b10110000 | row);
    return;
}

void lcd_init(void)
{
    // Set SPI control register appropriately (SPI interrupt enable, Master select)
    // I am using no prescaler (fastest speed) with no problems.
    SPCR = 0b01010000;
    // For testing use 0b01010011

    LCD_RESET_SET;
    // LCD initialization commands
    lcd_comm(0xA0); // ACD select (set RAM address: 0 normal)
    lcd_comm(0xAE); // Display OFF
    lcd_comm(0xC0); // Common Output Mode (COM) select (normal)
    lcd_comm(0xA2); // LCD bias voltage set (1/9 bias)
    lcd_comm(0x2F); // Power control set (internal power supply operating mode)
    lcd_comm(0x26); // V5 voltage regulator resistor ratio set
    lcd_comm(0x81); // Electronic volume mode set
    lcd_comm(0x22); // V5 voltage regulator resistor ratio set
    screen_clear(); //Clear LCD RAM
    lcd_comm(0xAF); // Display ON
}
```

