



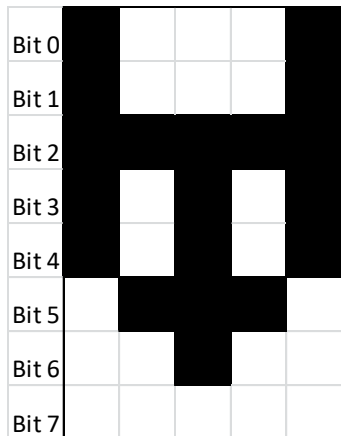
模块 7

活动：SPI 通信

活动：SPI 通信

问题1

定义你自己想显示的图像。使其高7像素，宽5像素。例如，德克萨斯大学奥斯汀分校的标志是UT。



该图像可以在字体表中定义为5个8-bit数字
{0x1f, 0x24, 0x7c, 0x24, 0x1f}

注意bit0位于顶部，注意bit7清零。请参阅JN_LCD.c文件中ASCII表的最后两行。将定义图像的5个8bit数字放入0x7F行，并通过输出进行测试
JN_LCD_OutChar(0x7F);

问题2

假设SPI时钟为12 MHz。假设每个字母图像的左边和右边都有一条8像素高，5像素宽的空白垂直线。换句话说，为了输出上述UT符号，软件必须输出这些数据。
{0x00, 0x1f, 0x24, 0x7c, 0x24, 0x1f, 0x00}

在液晶显示屏上绘制一个ASCII字符大约需要多长时间？

问题3

注意，在JN_LCD.c文件中，有一组在RAM缓冲区screen上操作的函数。这个函数是

JN_LCD_ClearBuffer	Clears the buffer (not the LCD)
JN_LCD_PrintBMP	Draw image into buffer
JN_LCD_ClrPxl	Clear pixel in buffer
JN_LCD_SetPxl	Set pixel in buffer
JN_LCD_DisplayBuffer	Displays buffer on LCD

使用驱动程序的方式是

- 1) 调用ClearBuffer
- 2) 根据需要调用PrintBMP ClrPxlSetPxl以形成图像
- 3) 调用DisplayBuffer更新显示

只要1-2-3序列发生的速度快于30次/秒，该显示在人眼看起来是连续的

Part a) 开发一个直线绘制到此缓冲区的函数。x坐标从0到47之间变化，y坐标从0到63之间变化。您可以调用任何现有的JN_LCD.c函数。

```
JN_LCD_Line(uint8_t x1, uint8_t y1,  
            uint8_t x2, uint8_t y2);
```

Part b) 开发一个将未填充的矩形绘制到此缓冲区中的函数。x坐标从0到47之间变化，y坐标从0到63之间变化。您可以调用任何现有的JN_LCD.c函数。这两个点是矩形的对角。

```
JN_LCD_Rect(uint8_t x1, uint8_t y1,  
            uint8_t x2, uint8_t y2);
```

Part c) 开发一个将填充的矩形绘制到此缓冲区中的函数。x坐标从0到47之间变化，y坐标从0到63之间变化。您可以调用任何现有的JN_LCD.c函数。这两个点是矩形的对角。

```
JN_LCD_RectFill(uint8_t x1, uint8_t y1,  
                uint8_t x2, uint8_t y2);
```