

模块 7

实验：SPI 通信



实验：SPI 通信

7.0 目标

本实验的目的是将 OLED 显示屏连接到微控制器并开发一组软件程序，以帮助调试机器人上的后续实验。

1. 您将 OLED 连接到微控制器。
2. 您将使用同步串行协议进行通信。
3. 您将 OLED 实现一组软件功能。

小知识：尽管 OLED 实际上并没有为解决机器人挑战所需的输入 - 计算 - 输出循环做出贡献，但是在竞技场中不受限制运行的机器人进行调试时，它将非常有用。

7.1 入门

7.1.1 从下面的软件工程起步

浏览下面这个工程：

Lab_SPI_OLED（视频中对应的程序）

7.1.2 参考资料

1.3inch_OLED.zip（data sheet for the OLED）

7.1.3 阅读材料

Volume 1 Sections 4.5, 7.6, 7.7, 8.3, and 8.4
Embedded Systems: Introduction to the MSP432 Microcontroller",
or
Volume 2 Sections 1.5, 3.4, and 7.5
Embedded Systems: Real-Time Interfacing to the MSP432 Microcontroller",

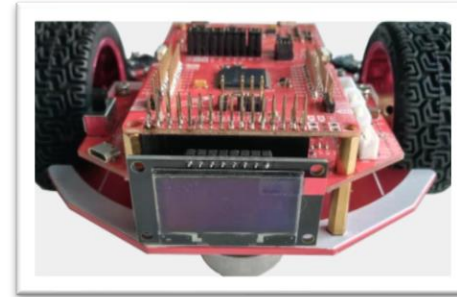
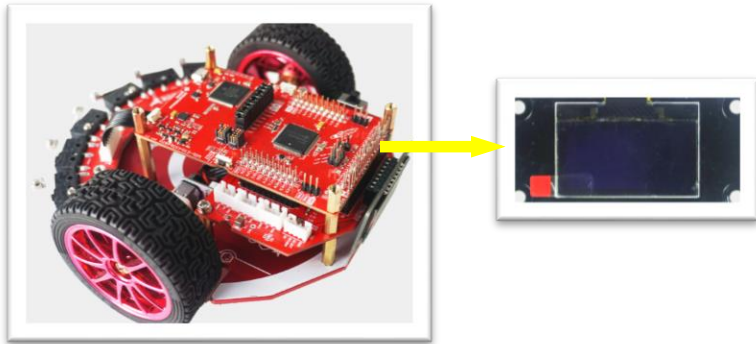


图 1.位于机器人后部的 JN_OLED 屏。

7.1.4 本实验所需组件

数量	组件描述	制造商	型号
1	MSP-EXP432P401R LaunchPad	TI	MSP-EXP432P401R
1	OLED display	JiangNiu	

7.1.5 所需实验设备

示波器（2 个至少 10MHz 采样的通道）
或逻辑分析仪（4 个至少 10MHz 采样的通道）

7.2 系统设计要求

本实验的总体目标是将 OLED 连接到微控制器和用它来输出字符和数字。
JN_OLED 是一款 128 像素宽，64 像素高的显示屏。每个字符由 5 像素宽×8 像素高的图像定义。您可以在 JN_LCD.c 启动文件中将字体表视为名为 ASCII 的常量数组。例如，字母'7'在第 114 行中定义为

```
{0x01, 0x71, 0x09, 0x05, 0x03}
```



实验：SPI 通信

该 40 位值在显示屏上创建图 1 所示的图像。0x01 是第一列，0x03 是最后一列，bit0 位于顶部。每个字符都有第 7 位清除，以便在行之间留出空格。函数 JN_LCDOutChar 将在每个字符后面放置一条空白垂直线和一个空白行。这意味着每个角色需要 7 x 8 像素区域才能打印。由于显示器为 128 宽 64 高，因此该字体大小允许每行 128/7，最多 18 个字符，并允许 64/8 = 8 行。

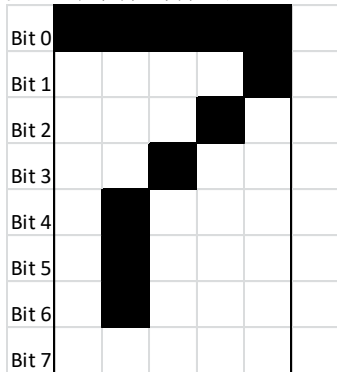


图 1. 创建数字 7 的像素图像，显示字符前后的空白垂直线。对于所有字符，Bit7 都是清除的。

在初级阶段，您需要编写一个向 OLED 发送 8-bit 命令的例程（**lcdcommandwrite**）和另一个向 OLED 发送 8-bit 数据的例程（**lcddatawrite**）。这些例程使用与 SPI 同步串行接口的忙等待同步。这些是私有函数，因此 JN_LCD.h 头文件中没有原型。

在高级阶段，您需要编写一个输出字符串的例程（**JN_LCD_OutString**）和另一个例程来输出无符号的 16 位十进制数字（**JN_LCD_OutUDec**）。

您可以在 inc 文件夹中找到 JN_LCD.h 和 JN_LCD.c 文件。这意味着在完成本实验后，您可以在其余实验中使用这些功能。

7.3 实验准备

您将使用 MSP432 LaunchPad 和 OLED 实现本实验。图 2 显示端口 9 可用于连接 OLED。但是，您可以使用任何支持同步串行端口（SPI 协议）的引脚。引脚 P9.7, P9.5 和 P9.4 配置为 SPI 模式。引脚 P9.3 和 P9.6 设置为常规数字输出。DC = 1 表示数据，DC = 0 表示命令。复位引脚用于初始化 OLED 硬件（RESET = 0，至少 1000ns 等待，RESET = 1）。

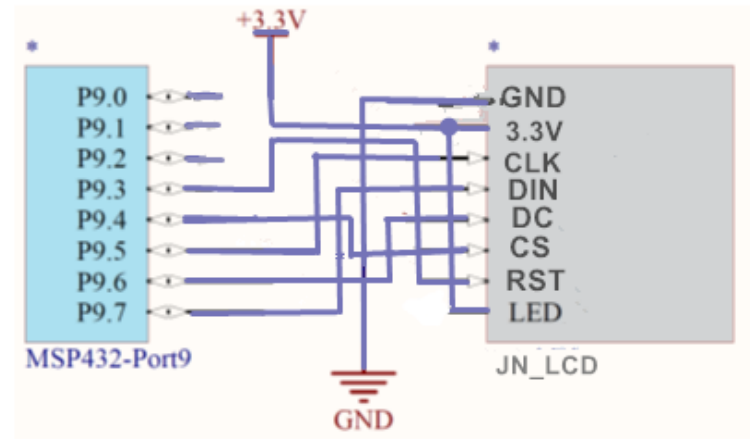


图 2. 端口 9 上的五个引脚可用于连接 OLED。

7.4 实验步骤

7.4.1 lcdcommandwrite

要发送一个 8 位命令，lcdcommandwrite 函数应执行以下四个步骤。

- 1) 等待 SPI 空闲（让前一帧结束），UCBUSY。
- 2) 设置命令 DC (0)。
- 3) 将 8-bit 命令写入 SPI 数据寄存器 (TXBUF)，启动 SPI。
- 4) 等待 SPI 空闲（让这个传输结束），UCBUSY。

在 JN_OLED 和 MSP432 数据手册中查找在程序中调用 lcdcommandwrite 时预期的波形。使用示波器或逻辑分析仪验证波形是否符合预期。您可以使用如下的简单程序来测试此初级功能。

```
void TestLcdCommandWrite (void) {
while (1) {
    lcdcommandwrite (0x21);
}
}
```



实验：SPI 通信

7.4.2lcddatawrite

要发送一个 8-bit 数据，lcddatawrite 函数应执行以下三个步骤。

- 1) 等待 SPI 空闲（让前一帧结束），UCBUSY。
- 2) 设置数据 DC（1）。
- 3) 将 8-bit 数据写入 SPI 数据寄存器（TXBUF），启动 SPI。

7.4.3JN_LCD_OutString

有一个叫做 JN_LCD_OutChar 的中级函数，可以向 OLED 输出一个字符。您将使用此函数实现一个名为 JN_LCD_OutString 的函数，该函数将一个字符串输出到 OLED。

7.4.4JN_LCD_OutUDec

同样，您将使用 JN_LCD_OutChar 实现一个名为 JN_LCD_OutUDec 的函数，该函数可向 OLED 输出一个 16 位无符号数字。此函数的一个规格是右对齐创建的图像恰好填充 5 个字符。此函数允许您在 OLED 上输出易于阅读的数字。

更具体地说，对于数字 0 到 9，您将输出 4 个空格和一个数字。对于数字 10 到 99，您将输出 3 个空格，后跟两个数字。对于数字 100 到 999，您将输出 2 个空格，后跟三个数字。对于数字 1000 到 9999，您将输出 1 个空格，然后输出 4 个数字。对于数字 10000 到 65535，您将输出所有五位数。

为了说明如何使用此功能，请考虑一下示例：OLED 在机器人运行期间不断更新调试数据，您可以在开始时执行一次此代码

```
JN_LCD_Set_Pos(1,2); // left, third row
JN_LCD_OutString("D= ");
JN_LCD_OutUDec(0);
JN_LCD_OutString(" mm");
```

然后，当您希望使用新的距离值更新 OLED 时，您只需输出新数值的 5 个字符。这种方法可以让数字改变时屏幕不会闪烁从而拥有优秀的显示效果。

```
JN_LCD_SetCursor(26,4); // spot for number
JN_LCD_OutUDec(myDistance);
```

使用调试配置文件来测量执行 JN_LCD_OutUDec 所需的时间。已知 12 MHz SPI 时钟频率是 12 MHz，解释为什么这种测量方法是合理的。

7.5 疑难解答

液晶显示屏不显示字符：

- 检查 LaunchPad 和 OLED 之间的所有连接。
- 确保 RESET 为高。
- Run 运行一个简单的主程序，使用相同的命令反复调用 lcdcommandwrite。使用逻辑分析仪或示波器验证从 LaunchPad 到 OLED 的所有五个信号都是正确的。数据手册的第 12.1 节描述了预期的 SPI 协议。

D1 灯不工作：

- 检查地线（GND）。
- 确认电源线（VCC）连接正确。

7.6 请思考

在本节中，我们列出了完成本实验后要思考的思考问题。这些问题旨在测试您对本实验中概念的理解。

- 该接口是串行的意味着什么？为什么串行很重要？
- 这个接口是同步的是什么意思？为什么同步重要？
- 这三个文件的目的是什么：JN_LCD.h, JN_LCD.c 和 Lab11_LCDmain.c？
- 执行 JN_LCD_OutUDec 需要多长时间？在 ISR 期间调用此功能是否合适，还是总是在主程序中执行 OLED 输出更好？

7.7 其它挑战

在本节中，我们列出了您可以执行的其它实验，以进一步探索此模块的概念。您可以扩展系统或提出完全不同的东西。例如，



实验：SPI 通信

- 为 16-bit 带符号数添加输出函数。
- 为带符号的 32-bit 数字添加输出函数。
- 连接不同的 LCD，例如 ST7735R。
- 创建一组功能，在 OLED 上绘制数据与时间的关系图。

7.8 接下来是哪些模块？

模块 1-11 介绍了微控制器的基础知识。下一组模块允许机器人具有更复杂的功能。

模块 3) 将电机的速度通过 OLED 屏进行显示。

模块 4) 将循线传感器检测到的黑白线的数值通过 OLED 屏显示。

模块 8) 连接红外距离传感器测量到墙壁的距离。

7.9 您应该已经学会

在本节中，我们将回顾您应该在本单元中学到的重要概念：

- 了解忙等待并明白为什么此接口使用了忙等待而不是中断。
- 同步串行协议：它的工作原理以及它的重要性。
- 最小侵入式调试的概念。