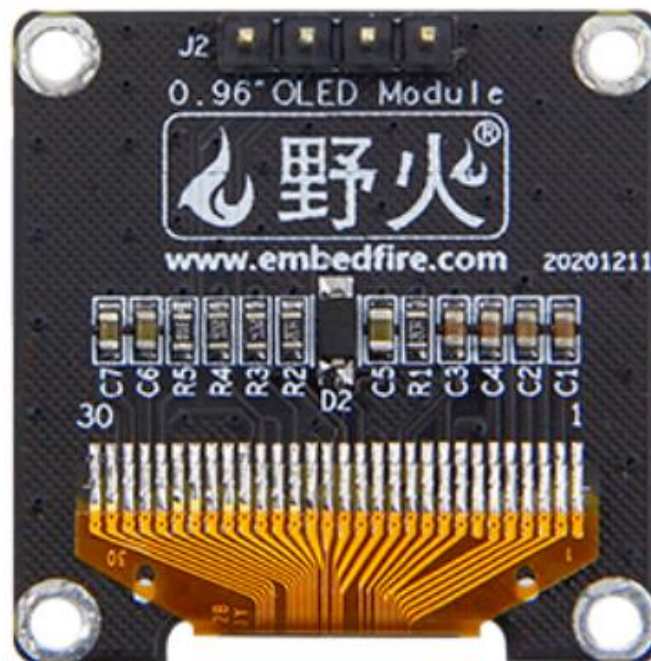


## 0.96 寸\_IIC\_OLED 模块



## 销售与服务联系

### 东莞野火科技有限公司

地址：东莞市大岭山镇石大路 2 号艺华综合办公大楼 301 1 2 3 4 楼

官网：<https://embedfire.com>

论坛：<http://www.firebbs.cn>

资料：<https://doc.embedfire.com>

天猫：<https://yehuosm.tmall.com>

京东：<https://yehuo.jd.com/>

邮箱：[embedfire@embedfire.com](mailto:embedfire@embedfire.com)

电话：0769-33894118

### 扫码获得更多精彩



野火百科



野火电子



野火天猫店



野火京东店



野火抖音号



野火视频号



野火B站号



野火小师妹

## 第一章 产品介绍

### 1.1 模块简介

0.96 寸 IIC OLED 模块，屏幕分辨率为 128\*64 像素，黄蓝双色显示，驱动芯片为 SSD1306，使用 IIC 接口，用于显示字符、图片等信息

### 1.2 参数特性

- ◆ 分辨率：128\*64 像素
- ◆ 显存大小（GDDRAM）：128\*64bit（128\*8Byte）SRAM
- ◆ 工作电压：3.3V
- ◆ 工作温度范围：-40° C ~ +85° C

## 第二章 使用说明

### 2.1 模块说明

（建议一边打开模块原理图并且一边对着芯片数据手册看）

J1 为 SSD1306 芯片的部分引出脚

VDD: IC 逻辑电源, 接 3V3

VCC: 面板驱动电源, 使用内部升压电路

NC/GND: 接地

BS[2:0]: 选择通信接口, BS0 和 BS2 接地, BS1 接 3V3, 使用 IIC 接口

CS: 片选, 低电平有效, 默认接地

RES: 复位, 低电平有效, 默认上拉

D/C: 在 IIC 模式下充当 SA0, 用于从机地址选择

R/W 和 E/RD: 在 IIC 模式下未使用, 该引脚需拉低接地

D0: 数据线 0, 在 IIC 模式下作为串行时钟线 SCL, 默认接上拉电阻

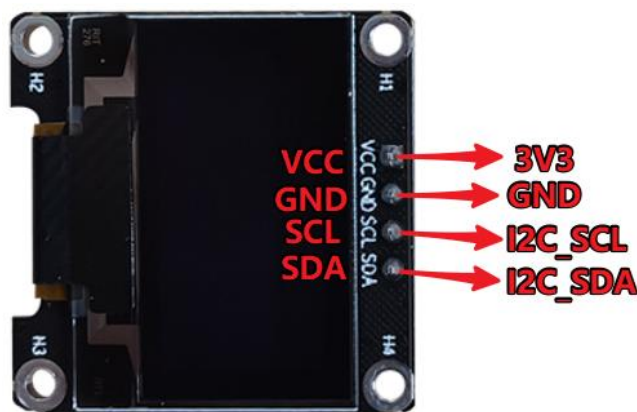
D1 和 D2: 数据线 1 和 2, 在 IIC 模式下需连接一起作为串行数据线 SDA, D1 做数据输入线 SDA<sub>in</sub>, D2 做数据输出线 SDA<sub>out</sub>, 默认接上拉电阻

D3 至 D7: 数据线 3~7, 未使用, 默认接地

IREF: 输出电流参考脚, 和 VSS 之间连接一个电阻, 以将 IREF 电流保持在 12.5 uA

OLED 的 IIC 地址有两个可选, 通过将 SA0 即 D/C 接入低电平或高电平, 地址可配置为“0111100”或“0111101”, 默认下拉, 使用“0111100”地址, 如果要更换 IIC 地址, 需要改 D/C 上拉到 3.3V

### 2.2 模块接口



编号	0.96 寸_IIC_OLED 模块	主控
1	VCC	3V3
2	GND	GND
3	SCL	I2C_SCL
4	SDA	I2C_SDA

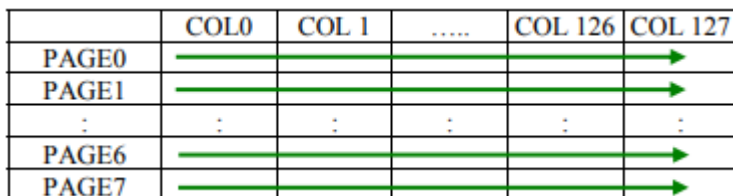
## 2.3 程序流程

SSD1306 芯片显存 GDDRAM 总共为 128\*64bit (128\*8Byte) 大小，对应屏幕的 128 (列)\*64 (行) 个像素点，SSD1306 将这些显存分成了 8 页，每页包含了 128 个字节

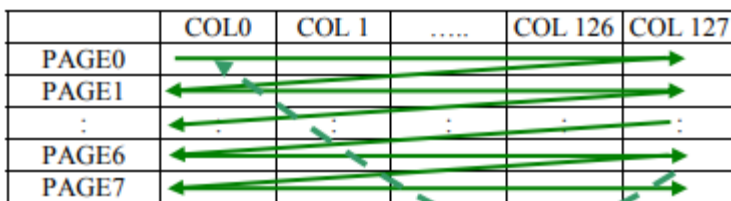
OLED: 128×64			0 列	1 列	2 列	.....	125 列	126 列	127 列
PAGE0	bit0	0 行							
	bit1	1 行							
	bit2	2 行							
	bit3	3 行							
	bit4	4 行							
	bit5	5 行							
	bit6	6 行							
	bit7	7 行							
PAGE1		8 到 15 行							
PAGE2		16 到 23 行							
PAGE3		24 到 31 行							
PAGE4		32 到 39 行							
PAGE5		40 到 47 行							
PAGE6		48 到 55 行							
PAGE7		56 到 63 行							

SSD1306 芯片提供三种内存寻址模式

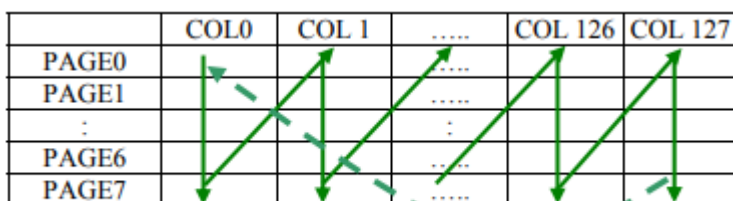
页寻址模式（默认）：写入数据，列地址指针自动增加 1，只能在当前页循环，不会自动换页，需要手动设置下一页，如 PAGE0 的 COL127 写完之后，地址指针会回到到 PAGE0 的 COL0



水平寻址模式：可以自动换页，如 PAGE0 的 COL127 写完之后，地址指针会移动到 PAGE1 的 COL0

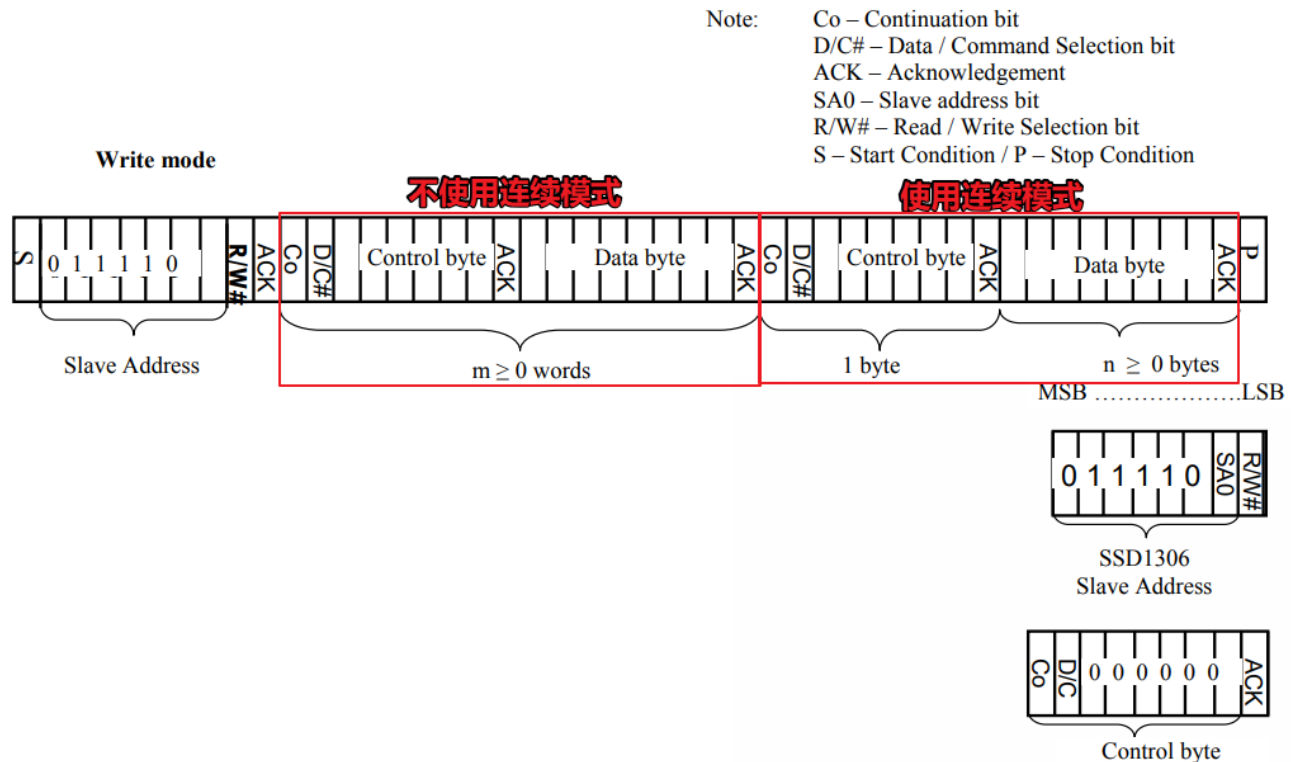


垂直寻址模式：写入数据，页地址指针自动增加 1，写完 1 列，自动换列，如 PAGE7 的 COL0 写完之后，地址指针会移动到 PAGE0 的 COL1



OLED 模块在串行模式下只能写入数据和命令，不能读取数据

写入数据或命令时序分析：如图所示是 IIC 协议数据总线上一帧数据的数据格式



1. 发送起始信号(S)
2. 确认 7 位从机地址和 1 位读写选择 (Slave Address+R/W#)
3. 从机应答信号 (ACK)
4. 发送控制字节 (Co+D/C#+ Control byte)

Co: 连续模式，置 1 表示每一个 Data byte 前都有一个 Control byte，置 0 则只有一次 Control byte，后面发送的都是 Data byte

D/C#: 数据指令选择位，置 1 表示写入的是数据，置 0 表示写入的是指令

Control byte: 没实际意义，填 0 即可

5. 从机应答信号 (ACK)
6. 发送数据或指令 (Data byte)

Data byte: 要发送的数据或指令，8 位，一个字节大小

7. 从机应答信号 (ACK)
8. 发送终止信号 (P)

下面以伪代码形式，举例实现逻辑

硬件 IIC 实现写入数据或命令（硬件 IIC 引脚自动完成时序）

STM32HAL 库

写指令：

```
{
    HAL_I2C_Mem_Write(IIC 句柄,
                      从机地址,
                      写入的地址确认是写指令,
                      写入的字节大小,
                      具体指令,
                      写入的字节数量,
                      等待时间)          调用 HAL 库函数实现写指令
}
```

写数据：

```
{
    HAL_I2C_Mem_Write(IIC 句柄,
                      从机地址,
                      写入的地址确认是写数据,
                      写入的字节大小,
                      具体数据,
                      写入的字节数量,
                      等待时间)          调用 HAL 库函数实现写数据
}
```

STM32 标准库

写指令：

```
{
    IIC_Write_Byte(确认是写指令,具体指令)    调用封装的发送一个字节函数
}
```

写数据：

```
{
    IIC_Write_Byte(确认是写数据,具体数据)    调用封装的发送一个字节函数
}
```

发送一个字节：调用标准库函数实现

```
{
    while (I2C_GetFlagStatus)                等待总线空闲
    I2C_GenerateSTART                          发送起始信号
}
```

while (!I2C_CheckEvent)	检测 EV5 事件并清除标志
I2C_Send7bitAddress	发送从机地址
while (!I2C_CheckEvent)	检测 EV6 事件并清除标志
I2C_SendData	发送写入的地址确认是写指令还是写数据
while (!I2C_CheckEvent)	检测 EV8 事件并清除标志
I2C_SendData	发送具体指令或数据
while (!I2C_CheckEvent)	检测 EV8 事件并清除标志
I2C_GenerateSTOP	发送终止信号
}	

软件 IIC 实现写入数据或命令（通过普通 GPIO 的拉高/拉低的动作来模拟 IIC 的硬件时序）

写指令：

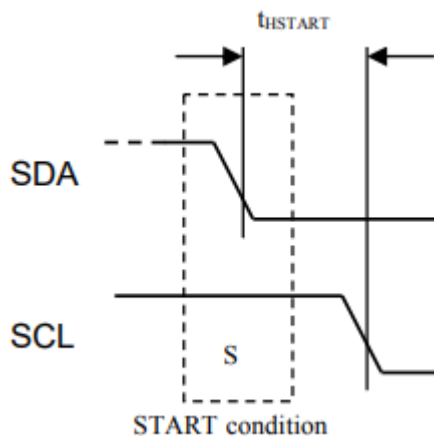
{	
IIC_Start	发送起始信号
IIC_Send_Byte(0xXX)	发送从机地址+读写位
while (IIC_Wait_ACK)	主机等待从机应答
IIC_Send_Byte(0xXX)	发送控制字节，确认是写指令
while (IIC_Wait_ACK)	主机等待从机应答
IIC_Send_Byte(0xXX)	发送指令
while (IIC_Wait_ACK)	主机等待从机应答
IIC_Stop	发送终止信号
}	

写数据：

{	
IIC_Start	发送起始信号
IIC_Send_Byte(0xXX)	发送从机地址+读写位
while (IIC_Wait_ACK)	主机等待从机应答
IIC_Send_Byte(0xXX)	发送控制字节，确认是写数据
while (IIC_Wait_ACK)	主机等待从机应答
IIC_Send_Byte(0xXX)	发送数据
while (IIC_Wait_ACK)	主机等待从机应答
IIC_Stop	发送终止信号
}	

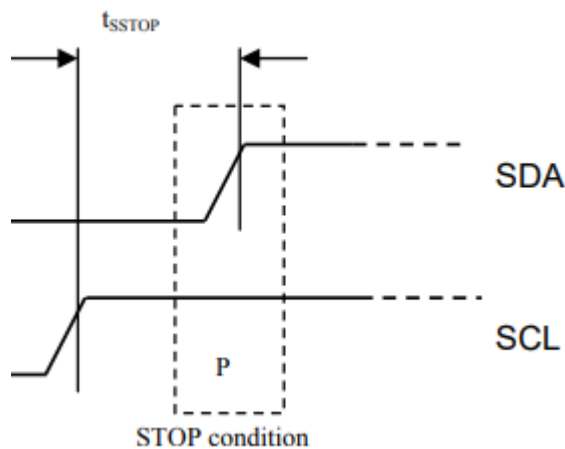


软件 IIC 的基本时序单元：起始信号、终止信号、发送一个字节、主机等待从机应答



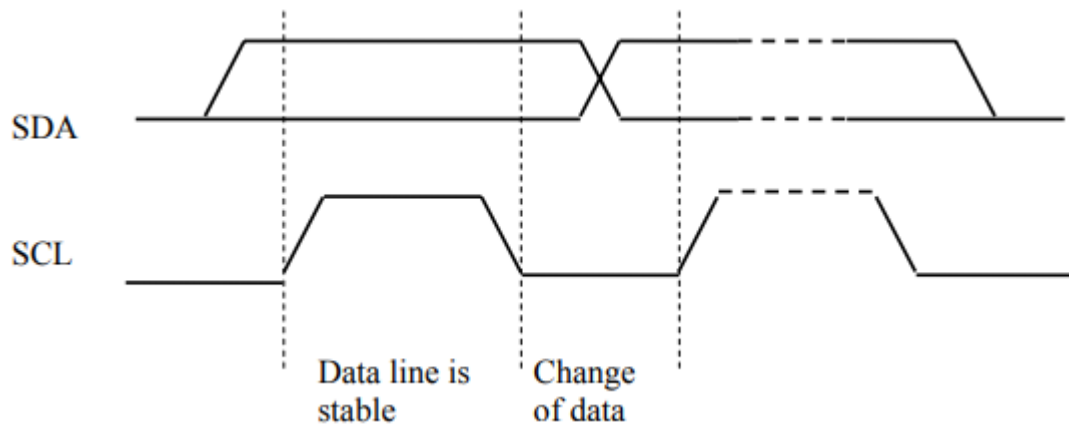
起始信号：SCL 高电平期间，SDA 从高电平切换到低电平(下降沿触发)

```
{
    IIC_SDA_1
    IIC_SCL_1           SCL 和 SDA 空闲时由上拉电阻拉至高电平
    IIC_SDA_0           SCL 高电平期间主机控制 SDA 拉低产生起始信号
    IIC_Delay            延时一段时间
    IIC_SCL_0            主机控制 SCL 拉低准备传输数据
}
```



终止信号：SCL 高电平期间，SDA 从低电平切换到高电平(上升沿触发)

```
{
    IIC_SDA_0
    IIC_SCL_0           SCL 和 SDA 拉低
    IIC_SCL_1           SCL 先拉高
    IIC_Delay            延时一段时间
    IIC_SDA_1           SCL 高电平期间主机释放 SDA 拉高产生终止信号
}
```

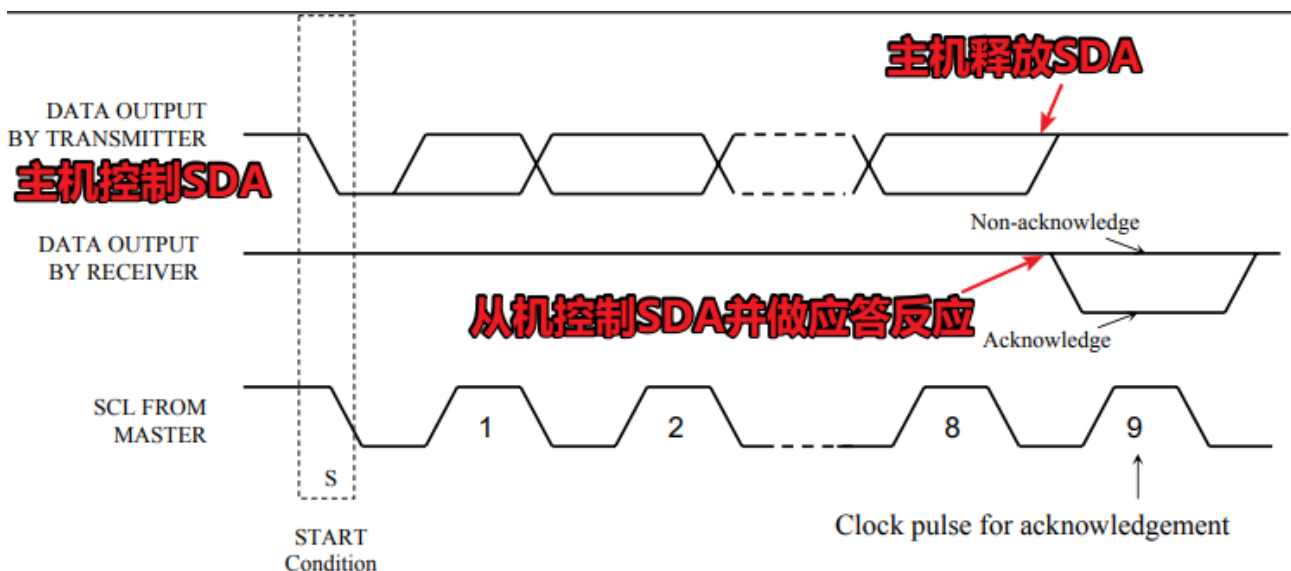


发送一个字节：SCL 低电平期间，主机将数据位依次放到 SDA 线上(高位先行)，然后拉高 SCL，从机将在 SCL 高电平期间读取数据位

```

{
    for (i = 0; i < 8; i++)                循环发送 8 次，每次发一位，高位先行
    {
        IIC_SDA_1 OR 0                    SDA 发送 0 或 1 数据
        IIC_SCL_1                          SCL 拉高，从机接收数据
        IIC_Delay                           延时等待从机接收数据
        IIC_SCL_0                          SCL 拉低，主机准备发送下一位数据
        IIC_Delay
    }
}

```



主机等待从机应答：主机在发送完一个字节后，在下一个时钟接收一位数据，判断从机是否应答 (主机在接收前，需要释放 SDA)

```

{
    IIC_SDA_1                            主机释放 SDA
    IIC_Delay
    IIC_SCL_1                            SCL 拉高
}

```

```
IIC_Delay
if (IIC_SDA_READ)           判断从机是否应答
    ACK = 1                 SDA 高电平没有应答
else
    ACK = 0                 SDA 低电平应答
IIC_SCL_0                   SCL 拉低
IIC_Delay
return ACK                  返回应答结果
}
```

用任意单片机确定使用 IIC 接口（硬件 IIC）或者任选 GPIO（软件 IIC）来驱动 OLED 模块

使用硬件 IIC：

1.初始化 MCU 对应的 GPIO 脚，SCL 和 SDA 配置为复用开漏输出模式

2.配置 IIC 结构体，初始化硬件 IIC

3.调用库函数编写 IIC 发送一个字节的函数

4.封装写指令和写数据函数

5.根据数据手册指令表，向屏幕发送指令完成屏幕初始化

6.通过写数据，控制屏幕像素点的亮灭

使用软件 IIC：

1.初始化 MCU 对应的 GPIO 脚，SCL 和 SDA 配置为开漏输出模式

2.编写时序基本单元：起始信号、终止信号、发送一个字节、主机等待从机应答（SCL 和 SDA 需要拉高/拉低的动作来模拟 IIC 的硬件时序）

3.封装写指令和写数据函数

4.根据数据手册指令表，向屏幕发送指令完成屏幕初始化

5.通过写数据，控制屏幕像素点的亮灭

更多详细内容可参考芯片数据手册和野火 STM32F1、F4 开发板教程 I2C 章节