电子工艺实训

OLED

一、I2C（IIC）简介

二、OLED显示屏的驱动和控制

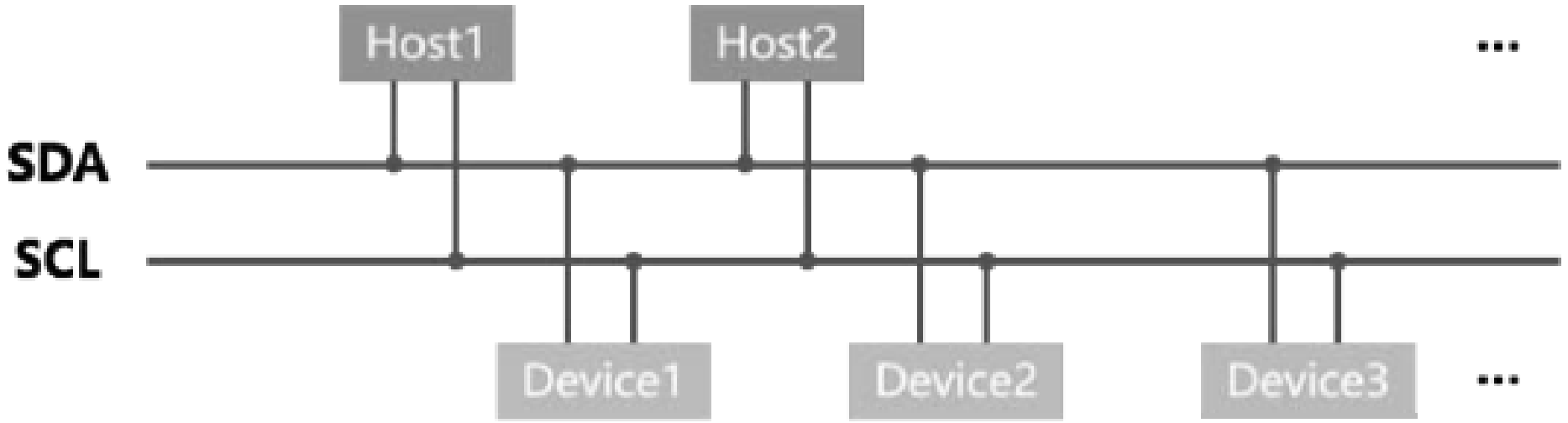
三、第三方OLED显示屏驱动库

四、OLED实训练习：OLED屏幕显示学号姓名

一、 I2C（IIC）简介

1、I2C含义

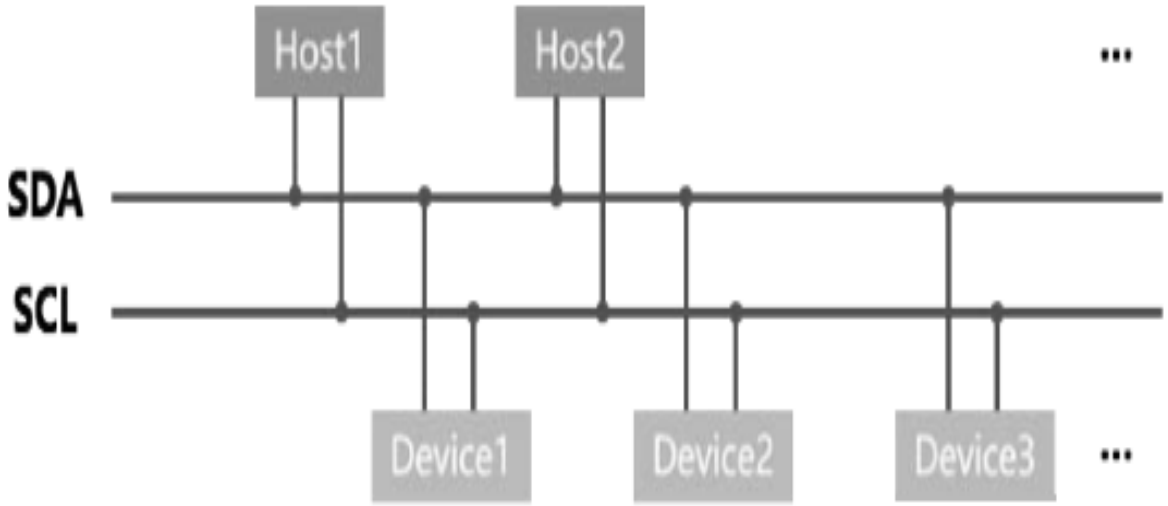
I2C 的全称是 Inter Integrated Circuit（内部集成电路），也可以简写成 IIC。



2、I2C总线

I2C 总线是由 Philips 公司开发的一种简单的、双向二线制的同步串行总线。它由SDA（串行数据线）和 SCL（串行时钟线）两根信号线构成。

I2C 以主从方式工作，通常会有一个或多个主设备（但在同一时刻只允许一个主设备占用 I2C 总线），一个或多个从设备。

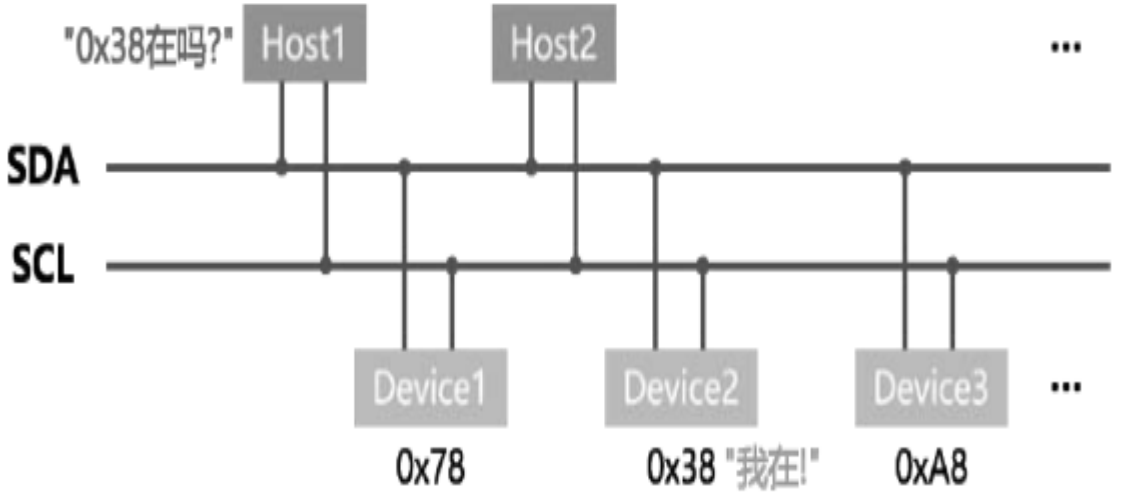


由于多个设备共用 SDA 和 SCL 两根线，所以它们之间的工作方式是串行的。

2、I2C总线

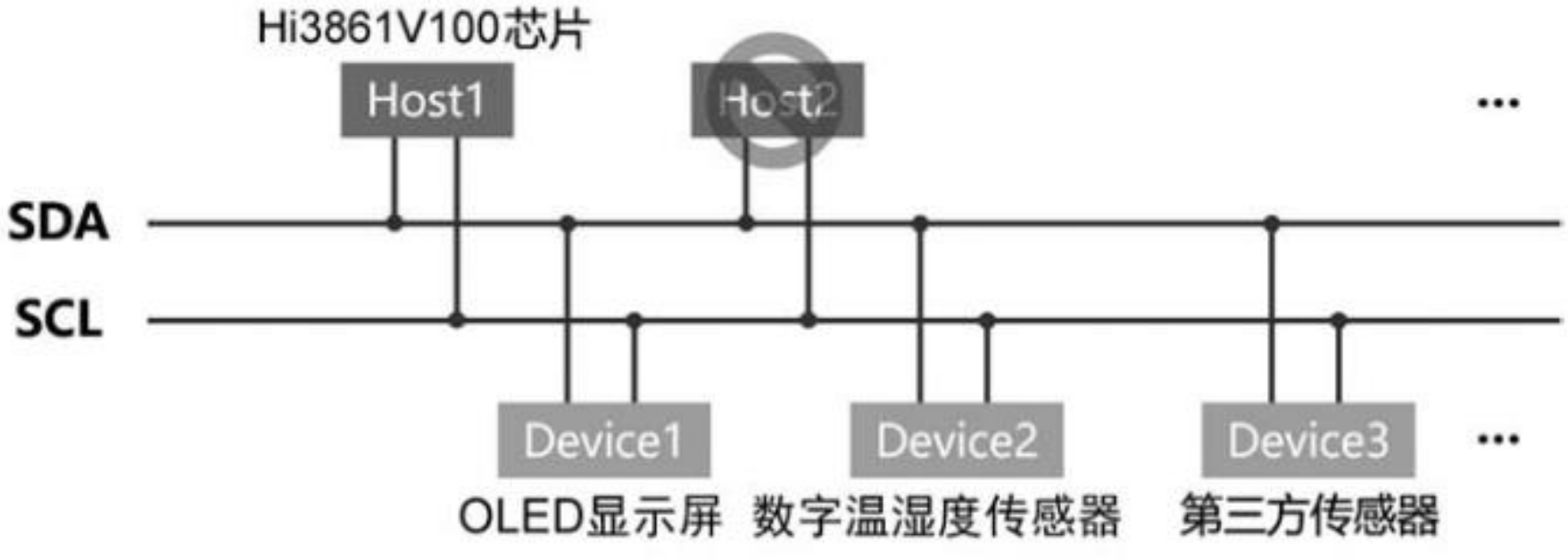
为了实现正确的数据传输， I2C 总线上的每个设备都拥有一个唯一的地址，用于区分彼此，我们把它叫设备地址。设备地址和 MAC 地址的作用是类似的，但是由于 I2C总线上的设备数量少，地址位会短得多，通常情况下 1 字节就够了。

当主设备需要与某个从设备通信时，要通过广播的方式将从设备地址写到总线上。如果某个从设备采用了此地址，将会发出应答信号，从而建立传输关系。I2C 总线是同步通信的，依靠的是时序 。



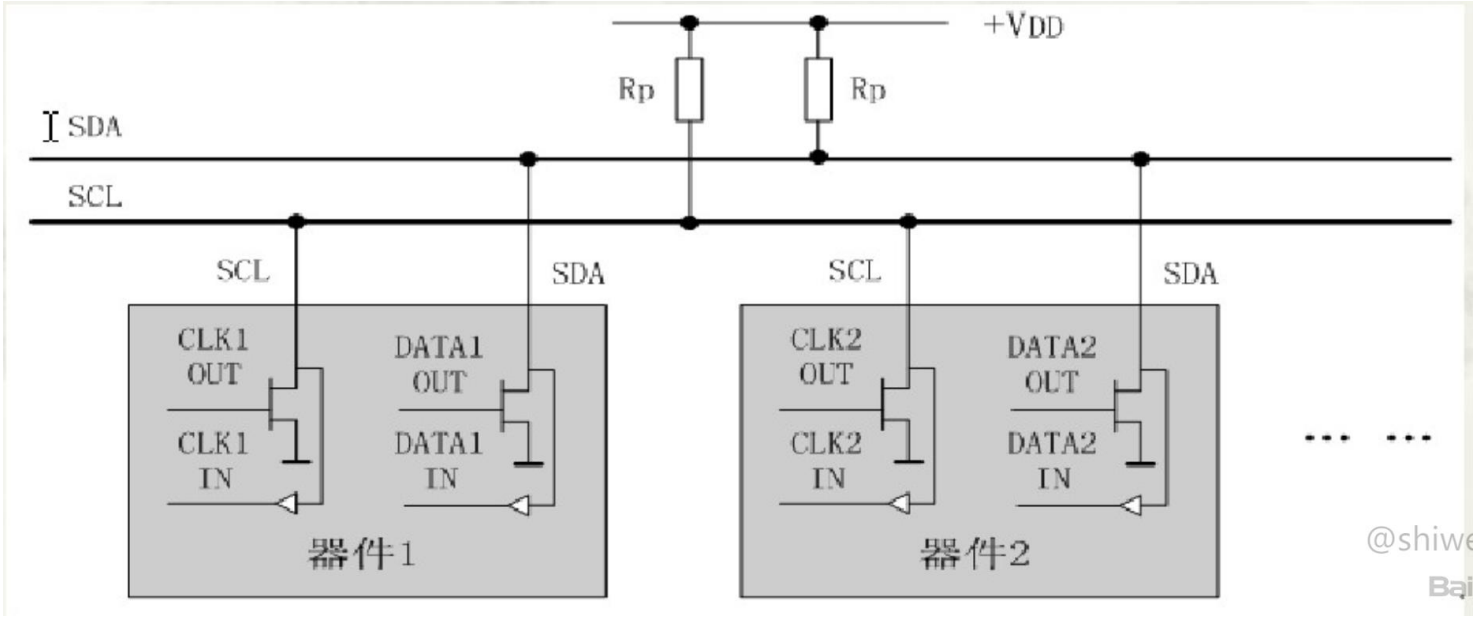
2、I2C总线

例如，在智能家居开发套件中，Hi3861V100 芯片是主设备，而 OLED 显示屏、数字温湿度传感器和其他第三方传感器则是从设备。



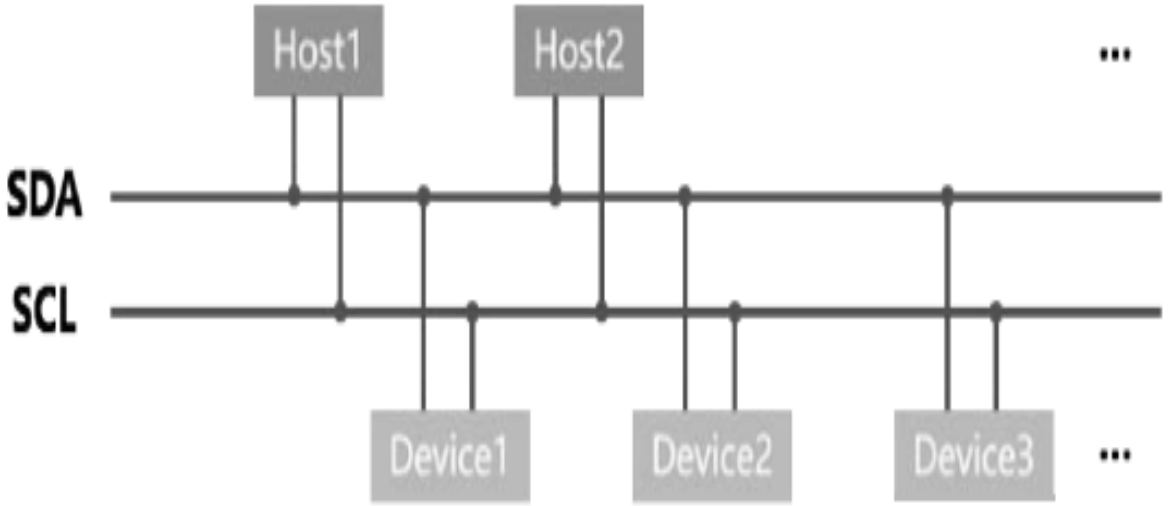
2、I2C总线

- SDA和SCL通过上拉电阻接正电源，
- 总线空闲时，SDA和SCL均为高电平
- 总线上任一器件输出低电平，总线电平变低（线与）



2、I2C总线

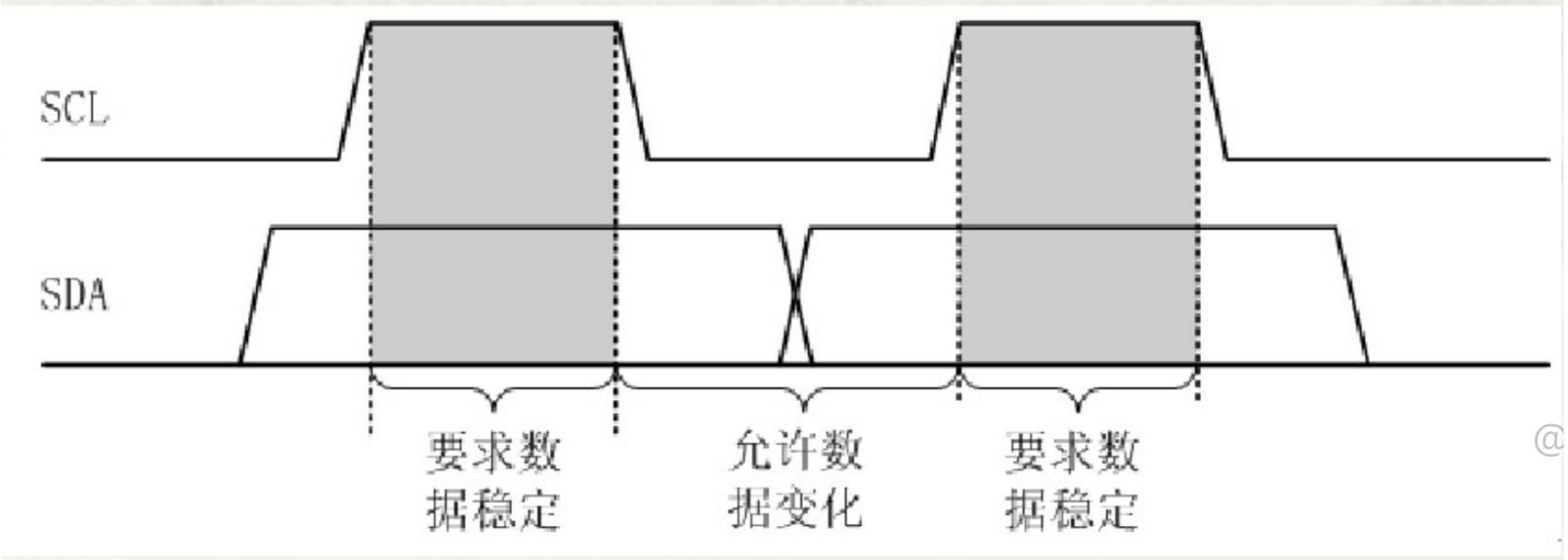
- 通信采用主从模式，一个主机，多个从机
- 若总线上的多个主机同时竞争总线，则需要通过仲裁决定谁控制总线
- 主机产生SCL线上的时钟信号
- 主机开始和结束一次通信



2、I2C总线

IIC总线的数据传送-数据位的有效性

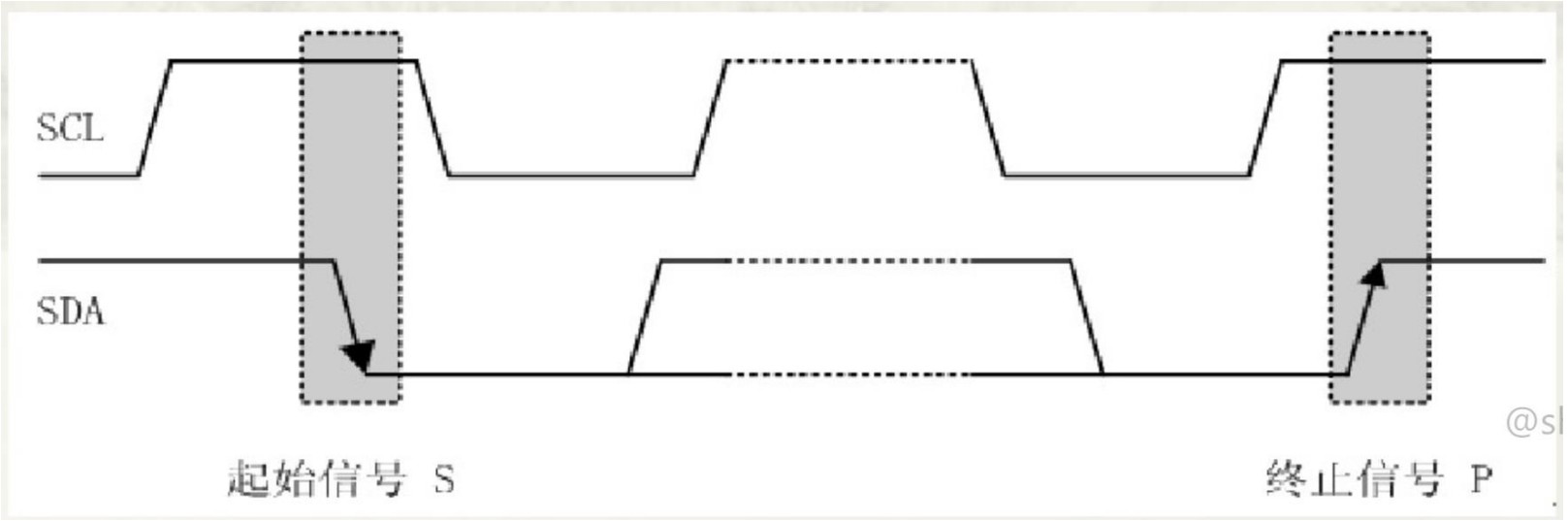
- 在时钟的高电平期间，SDA线上的数据必须保持稳定
- 在时钟的低电平期间，SDA线上的电平才允许变化



2、I2C总线

IIC总线的数据传送-起始和终止信号

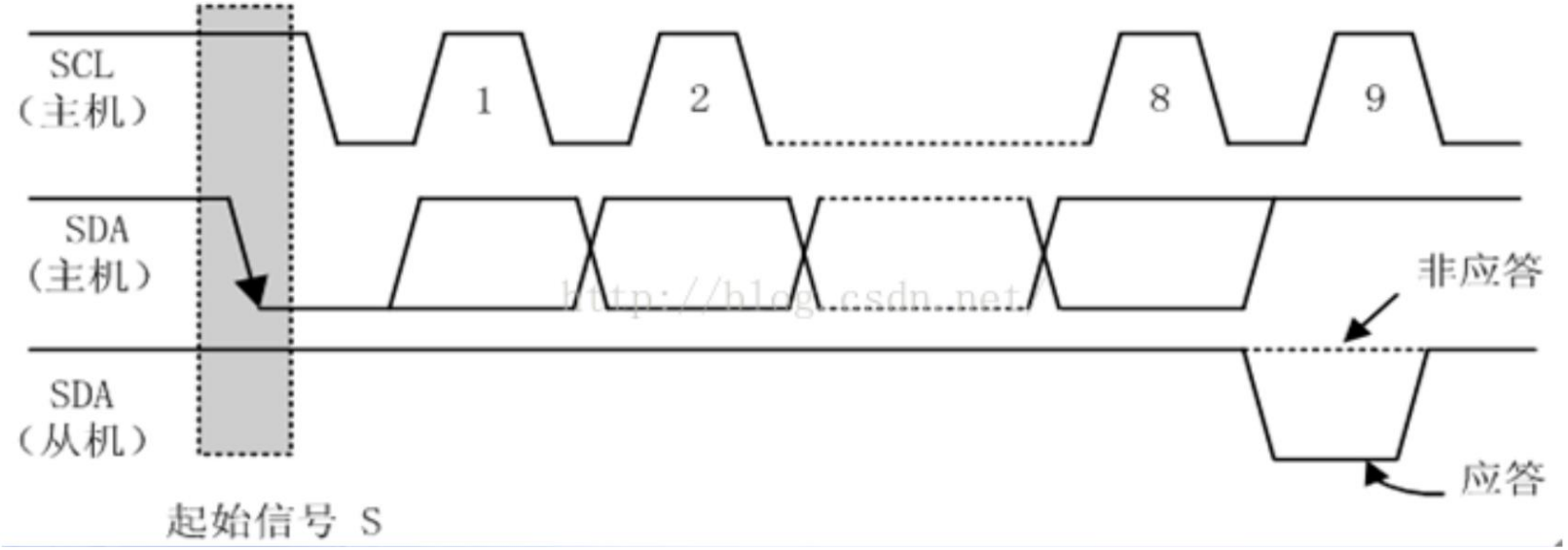
- 起始信号：SCL线为高电平期间，SDA线由高电平变为低电平
- 结束信号：SCL线为高电平期间，SDA线由低电平变为高电平
- 起始和终止信号都是由主机发出的



2、I2C总线

IIC总线的数据传送-应答

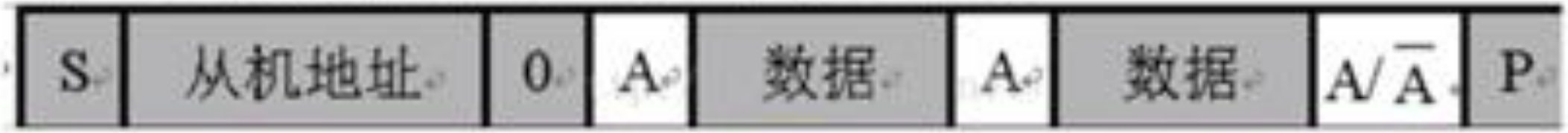
- 每当主机向从机发送完一个字节的数据，主机总是需要等待从机给出一个应答信号，以确认从机是否成功接收到了数据
- 应答出现在每一次主机完成8个数据位传输后紧跟着的时钟周期，低电平0表示应答，1表示非应答



2、I2C总线

IIC总线的数据传送-数据帧格式

- IIC总线上传送的数据信号是广义的，既包括地址信号，又包括真正的数据信号
- 在起始信号后必须传送一个从机的地址（7位），第8位是数据的传送方向位（R/T），用“0”表示主机发送数据（T），“1”表示主机接收数据（R）
- 每次数据传送总是由主机产生起始和结束信号。若主机希望继续占用总线进行新的数据传送，则可以不产生终止信号，马上再次发出起始信号对另一从机进行寻址。



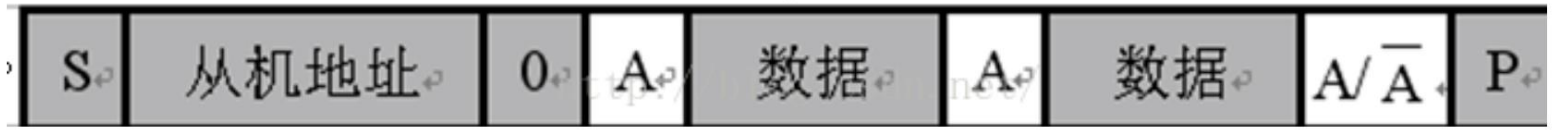
注：有阴影部分表示数据由主机向从机传送，无阴影部分则表示 数据由从机向主机传送。
A 表示应答(低电平)，A 非表示非应答（高电平）。S 表示起始信号，P 表示终止信号。

2、I2C总线

IIC总线的数据传送-数据帧格式

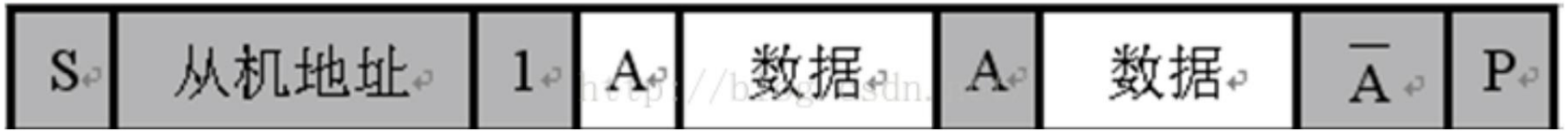
- 总线的一次数据传输过程中，可以有以下几种组合方式：

[1] 主机向从机发送数据，数据传送方向在整个传送过程中不变：



注：有阴影部分表示数据由主机向从机传送，无阴影部分则表示 数据由从机向主机传送。
A 表示应答(低电平)， A 非表示非应答（高电平）。 S 表示起始信号， P 表示终止信号。

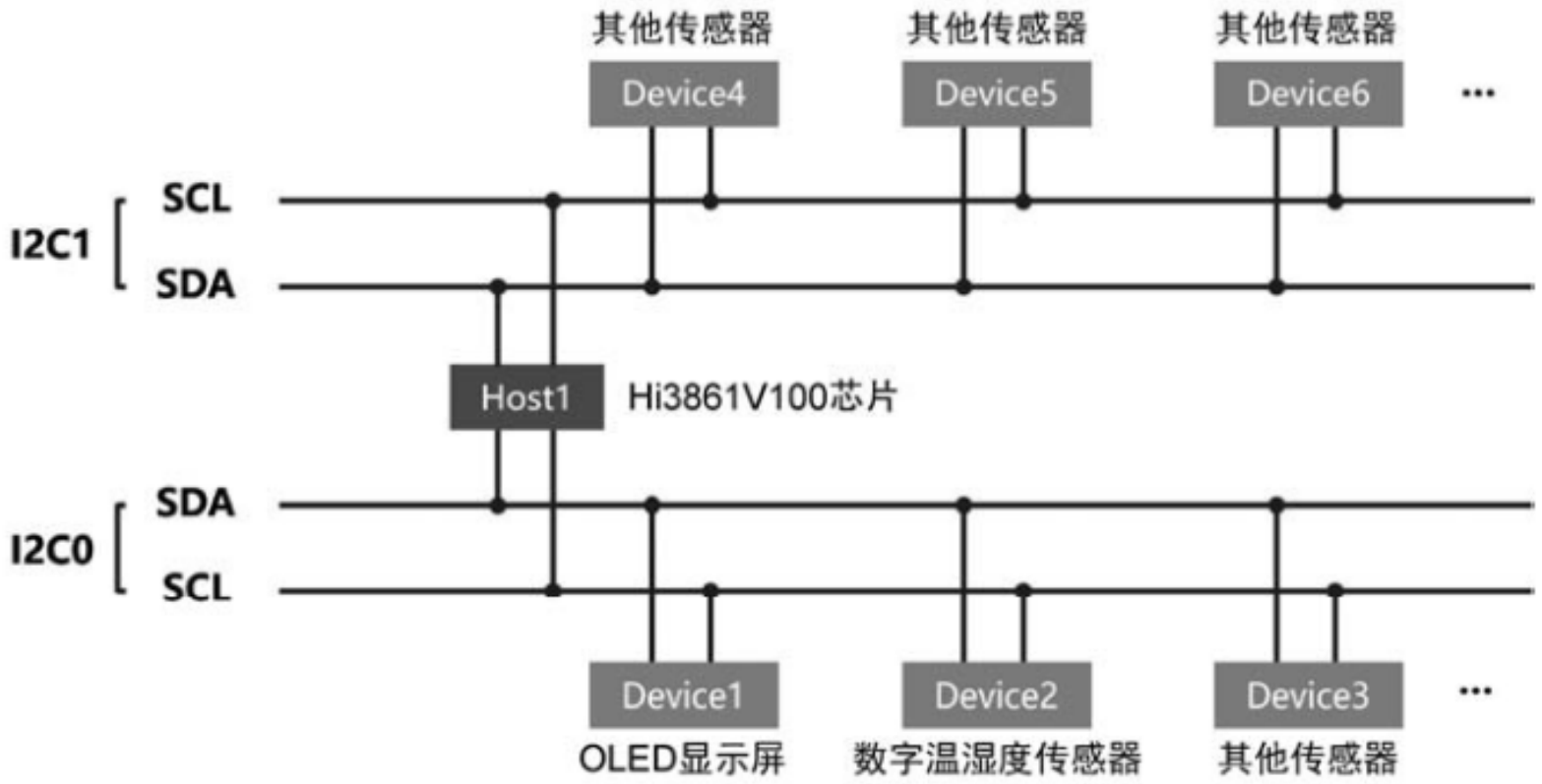
[2]主机在第一个字节后，立即从从机读数据:



[3]在传送过程中，当需要改变传送方向时，起始信号和从机地址都被重复产生一次，但两次读/写方向位正好反相:

3、Hi3861V100芯片的 I2C 引脚分布

Hi3861V100 芯片的硬件 I2C 总线有两个，分别是 I2C0 和 I2C1



3、Hi3861V100芯片的 I2C 引脚分布

由于 Hi3861V100 芯片的引脚数量有限，导致 I2C 功能没有独立的引脚，所以需要与 GPIO 功能进行引脚复用。对于同时具有 GPIO 和 I2C 这两种功能（或更多功能）的引脚来说，同一时刻只能使用其中的一种功能。

引脚编号	默认功能	I2C 引脚
2	GPIO-00	I2C1_SDA
3	GPIO-01	I2C1_SCL
5	GPIO-03	I2C1_SDA
6	GPIO-04	I2C1_SCL
27	GPIO-09	I2C0_SCL
28	GPIO-10	I2C0_SDA
31	GPIO-13	I2C0_SDA
32	GPIO-14	I2C0_SCL

4、相关 API 介绍

我们遵循 HAL+SDK 的接口使用策略，首先介绍 HAL 接口中的相关 API。接口位置在 base\iot_hardware\peripheral\interfaces\kits\iot_i2c.h 文件中

API 名称	说明
unsigned int IoTI2cInit(unsigned int id, unsigned int baudrate)	用指定的波特率初始化 I2C 控制器。参数 id 用于指定 I2C 总线 ID，参数 baudrate 用于指定波特率
unsigned int IoTI2cWrite(unsigned int id, unsigned short deviceAddr, const unsigned char *data, unsigned int dataLen)	将数据写入 I2C 设备中。参数 id 用于指定 I2C 总线 ID，参数 deviceAddr 用于指定 I2C 设备地址，参数 data 表示要写入的数据的指针，参数 dataLen 用于指定数据长度
unsigned int IoTI2cRead(unsigned int id, unsigned short deviceAddr, unsigned char *data, unsigned int dataLen)	从 I2C 设备中读取数据。参数 id 用于指定 I2C 总线 ID，参数 deviceAddr 用于指定 I2C 设备地址，参数 data 表示要读取的数据的指针，参数 dataLen 用于指定数据长度

4、相关 API 介绍

下面介绍海思 SDK 接口中的相关 API。请注意，海思 SDK 接口是备用接口，应当优先使用 HAL 接口。接口位置在 device\hisilicon\hispark_pegasus\sdk_liteos\include\hi_i2c.h 文件中

API 名称	说明
hi_i2c_init(hi_i2c_idx id, hi_u32 baudrate);	用指定的波特率初始化 I2C 控制器
hi_i2c_write(hi_i2c_idx id, hi_u16 device_addr, const hi_i2c_data *i2c_data);	将数据写入 I2C 设备中
hi_i2c_read(hi_i2c_idx id, hi_u16 device_addr, const hi_i2c_data *i2c_data);	从 I2C 设备中读取数据

二、 OLED显示屏的驱动和控制

1、OLED简介

- OLED 的全称是 Organic Light-Emitting Diode，即有机发光二极管。
- OLED 属于一种电流型的有机发光器件，有机半导体材料和发光材料在电场的驱动下通过载流子的注入和复合而导致其发光。 OLED 的发光强度与注入的电流成正比。
- OLED 有自发光、视角广、厚度薄、对比度高、清晰度高、构造简单、响应速度快、柔性好、使用温度范围广等特点。

1、OLED简介

OLED 的这些优点，使得它的应用领域十分广泛：

- 在商业领域中， OLED 可以用于POS 机、复印机、 ATM 机、广告屏等；
- 在消费类电子产品领域中， OLED 应用得最广泛的是智能手机，其次是笔记本、显示屏、电视、平板、数码相机、 VR 设备等；
- 在交通领域中， OLED 主要用于轮船和飞机的仪表、 GPS、可视电话、车载显示屏等；
- 在工业领域中， OLED 可以用于工控系统的显示屏、触控屏等；
- 在医疗领域中， OLED 可以用于医学诊断影像、手术监控屏幕等。

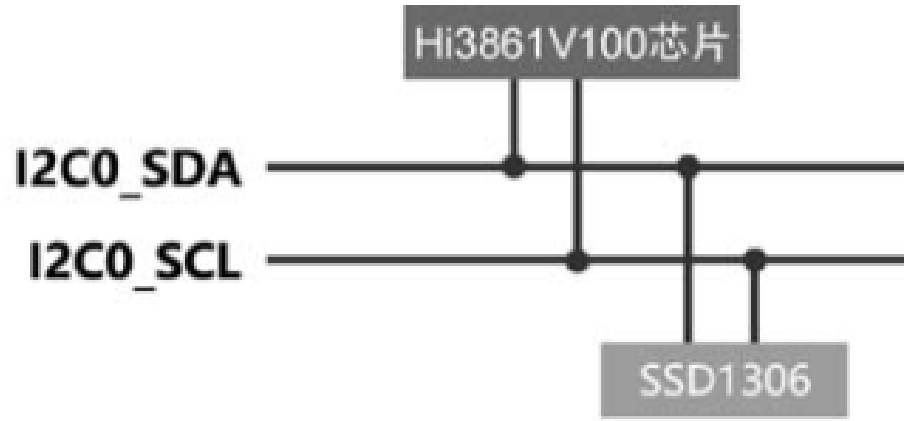
2、OLED显示屏板介绍

OLED 显示屏板的主要部件有以下 4 个：一个 0.96 英寸的 OLED 显示屏、一个 SSD1306 显示屏驱动芯片、两个按键。

- 0.96 英寸 OLED 显示屏。这块显示屏的屏幕分辨率为 128px× 64px，可以显示黑白两色。它的可视角度大于 160°，在正常使用时的功耗只有 0.06W。我们可以使用它显示文字、图形，实现简单的用户界面交互。

2、OLED显示屏板介绍

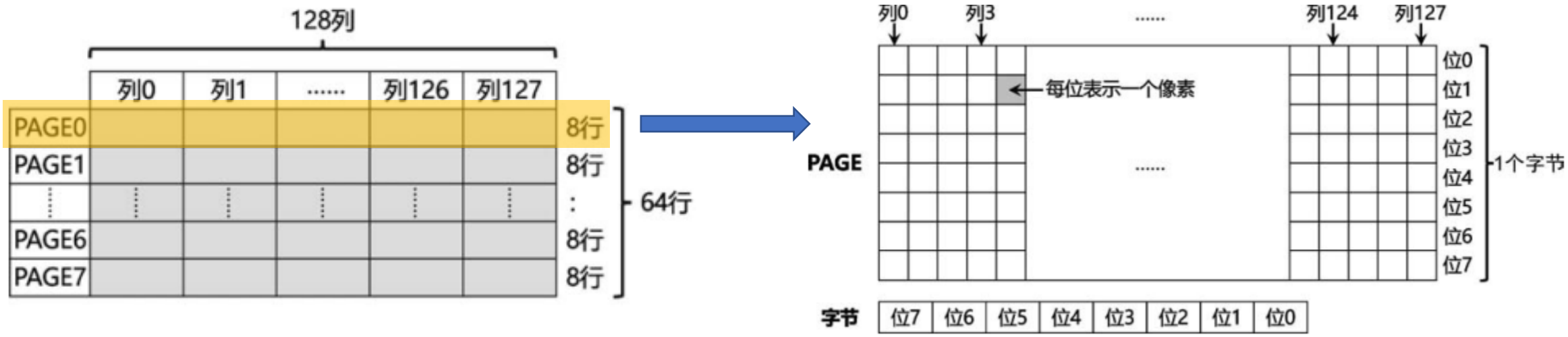
- SSD1306 显示屏驱动芯片。该驱动芯片采用 I2C 接口，连接到了 Hi3861V100 芯片的 I2C0 总线上，也就是 31 号和 32 号引脚上。 31 号引脚的复用关系为 GPIO-13 和 I2C0_SDA； 32 号引脚的复用关系为 GPIO-14 和 I2C0_SCL。
- 在 I2C0 总线上， SSD1306 显示屏驱动芯片的设备地址是 0x78。



2、OLED显示屏板介绍

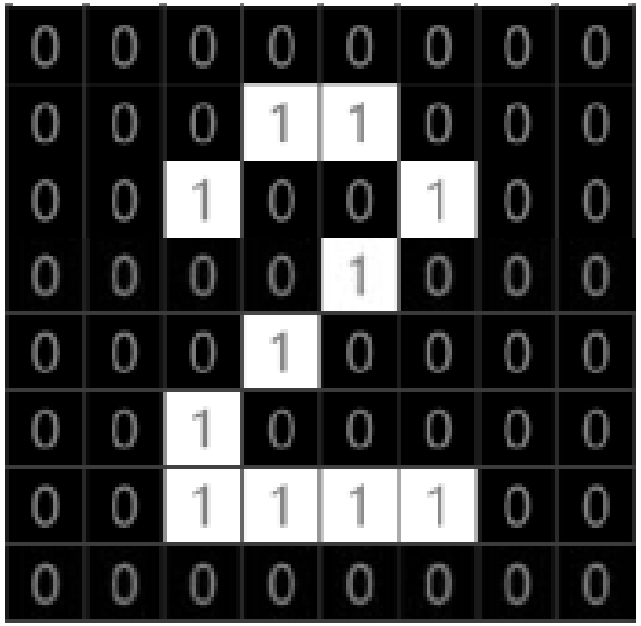
仔细阅读 OLED 显示屏板的原厂技术手册“ SSD1306.pdf”和“金马鼎 0.96 白色 30Pin.pdf”两个文件。OLED显示屏的驱动和控制一定要按照技术手册中的通信规则去使用它。

2、OLED显示屏板介绍



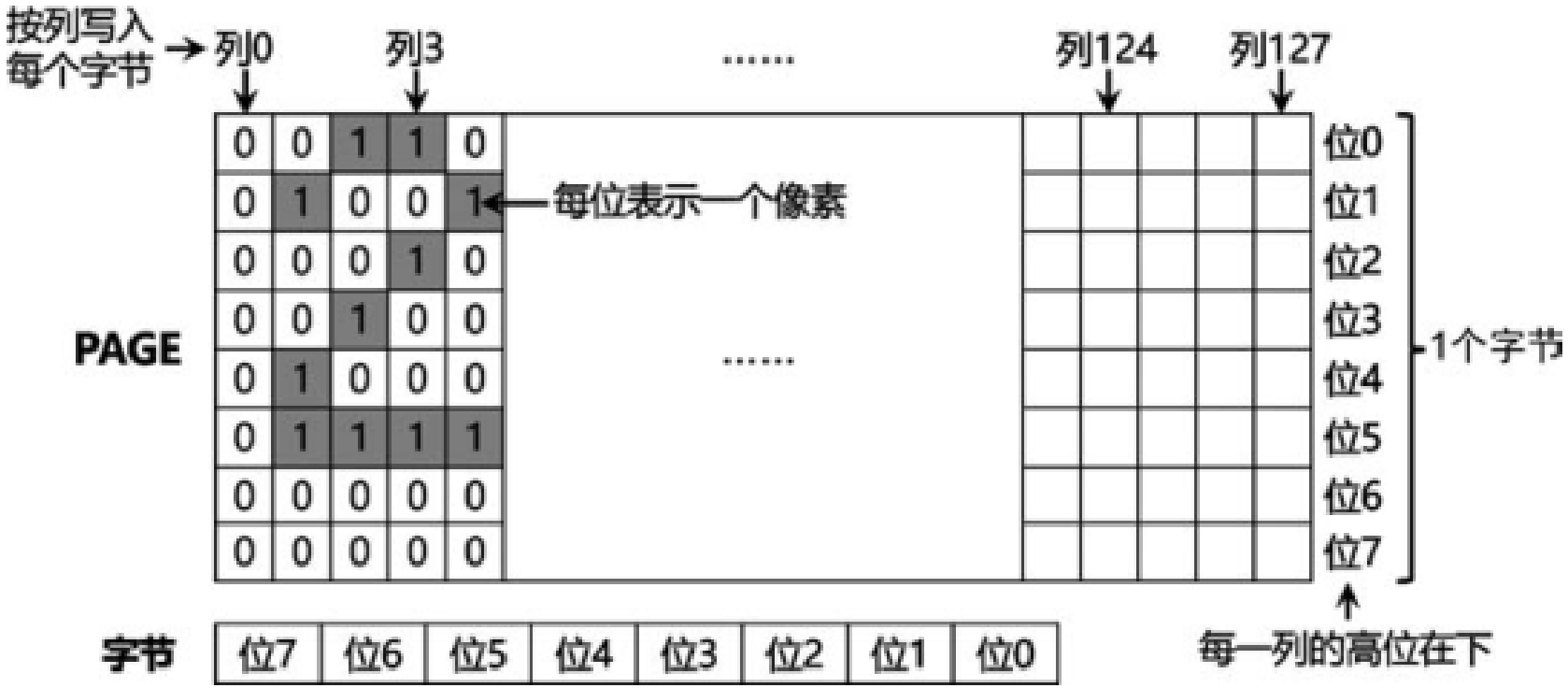
3、ASCII字符点阵显示

在屏幕上绘制 ASCII 字符需要用到点阵字体。点阵字体也叫位图字体，它的每个字形都用一组二维像素信息来表示。通俗地说，就是每个字的形状都用一张表来表示，表中的每个单元格都是一个像素。

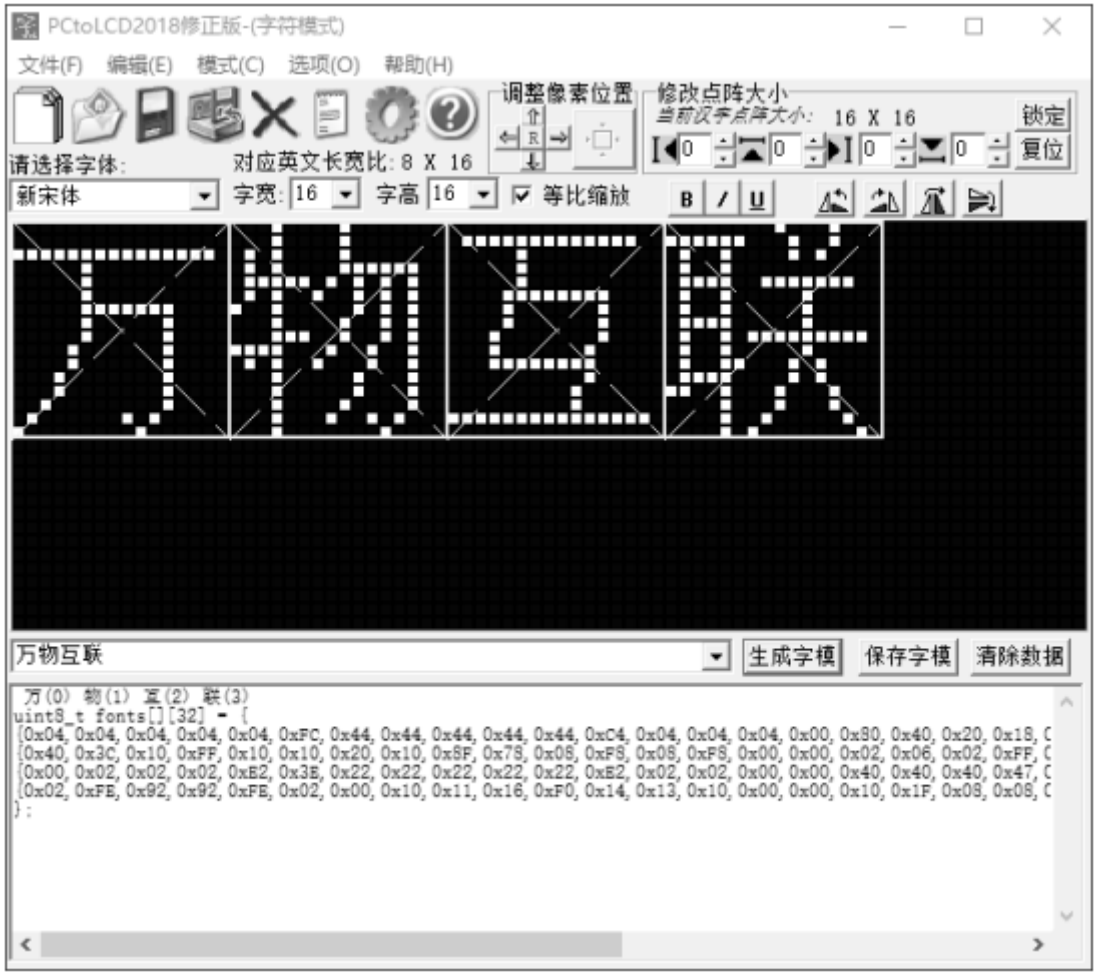


3、ASCII字符点阵显示

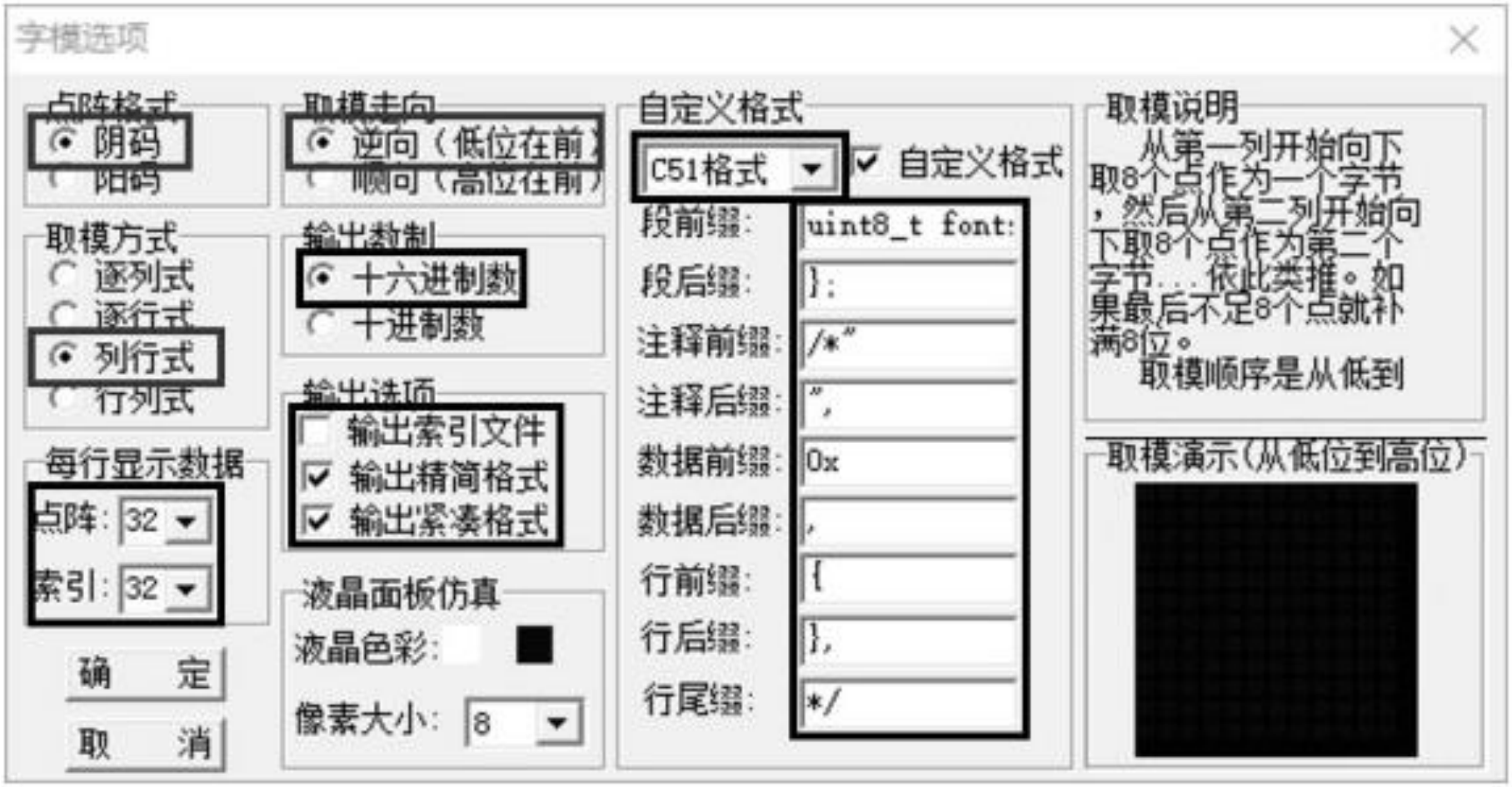
适合 SSD1306 显示屏驱动芯片的最佳取模方式为纵向 8 点下高位。



4、取模软件PCtoLCD



4、取模软件PCtoLCD



三、第三方OLED显示屏驱动库

1、驱动库简介

我们使用的第三方 OLED 显示屏驱动库是一个开源项目。

这个第三方驱动库的主要特性和功能如下：

（1）内置了 128bit× 64 bit 内存缓冲区，支持全屏刷新；

（2）优化了屏幕刷新速率，实测最大帧率可以达到 10FPS；

（3）使用 HAL API 和海思 SDK API；

（4）接口简捷，易于使用和移植；

（5）内置了测试程序，可以直接进行测试。

2、驱动库源码结构

注意有两层ssd1306目录，外层的ssd1306目录放在\src\applications\sample\wifi-iot\app中

测试程序、案例程序目录

从musl libc中抽取的sin和cos的实现，解决SDK中libm_flash.a报错问题

驱动源码目录

驱动编译脚本

驱动配置文件

ASCII字符集的字体库

字体库接口文件

驱动源码文件

驱动接口文件

驱动和测试程序编译脚本，废弃不用

用于将gif动图中的帧分离出来

用于将图片转为C数组，每个字节表示8个像素

python脚本依赖组件

3、驱动库API介绍

ssd1306.c 和 ssd1306.h 这两个文件是驱动库的核心文件。

API 名称	说明
void ssd1306_Init(void)	初始化 OLED 显示屏
void ssd1306_Fill(SSD1306_COLOR color)	以指定的颜色填充屏幕
void ssd1306_SetCursor(uint8_t x, uint8_t y)	定位光标
void ssd1306_UpdateScreen(void)	更新屏幕内容
char ssd1306_DrawChar(char ch, FontDef Font, SSD1306_COLOR color)	在屏幕缓冲区绘制 1 个字符
char ssd1306_DrawString(char* str, FontDef Font, SSD1306_COLOR color)	在屏幕缓冲区绘制字符串
void ssd1306_DrawPixel(uint8_t x, uint8_t y, SSD1306_COLOR color)	在屏幕缓冲区绘制 1 个像素
void ssd1306_DrawLine(uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, SSD1306_COLOR color)	画直线

3、驱动库API介绍

ssd1306.c 和 ssd1306.h 这两个文件是驱动库的核心文件。

void ssd1306_DrawPolyline(const SSD1306_VERTEX *par_vertex, uint16_t par_size, SSD1306_COLOR color)	绘制多段线
void ssd1306_DrawRectangle(uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, SSD1306_COLOR color)	绘制矩形
void ssd1306_DrawArc(uint8_t x, uint8_t y, uint8_t radius, uint16_t start_angle, uint16_t sweep, SSD1306_COLOR color)	绘制弧线
void ssd1306_DrawCircle(uint8_t par_x, uint8_t par_y, uint8_t par_r, SSD1306_COLOR color)	画圆
void ssd1306_DrawBitmap(const uint8_t* bitmap, uint32_t size)	绘制位图
void ssd1306_DrawRegion(uint8_t x, uint8_t y, uint8_t w, uint8_t h, const uint8_t* data, uint32_t size, uint32_t stride)	在指定区域绘制内容，按行绘制

4、驱动库接入方法

app/BUILD.gn

```
import("//build/lite/config/component/lite_component.gni")

lite_component("app") {
    features = [
        "ssd1306:app",
    ]
}
```

app/ssd1306/BUILD.gn

```
import("//build/lite/config/component/lite_component.gni")

lite_component("app") {
    features = [
        "ssd1306:oled_ssd1306",
        "examples:oled_test",
        "libm_port:libm_port"
    ]
}
```

app/ssd1306/ssd1306/BUILD.gn

```
static_library("oled_ssd1306") {
    sources = [
        "ssd1306.c",
        "ssd1306_fonts.c",
    ]

    include_dirs = [
        "//utils/native/lite/include",
        "//kernel/liteos_m/components/cmsis/2.0",
        "//base/iot_hardware/peripheral/interfaces/kits",
        "//device/soc/hisilicon/hi3861v100/hi3861_adapter/hals/communication/wifi_lite/wifiservice",
        "//device/soc/hisilicon/hi3861v100/hi3861_adapter/kal",
    ]
}
```

5、驱动自带的屏幕检测样例

app/ssd1306/examples/BUILD.gn

```
static_library("oled_test") {
    sources = [
        "ssd1306_demo.c",
        "ssd1306_tests.c",
    ]

    include_dirs = [
        "../ssd1306",
        "//utils/native/lite/include",
        "//kernel/liteos_m/components/cmsis/2.0",
        "//base/iot_hardware/peripheral/interfaces/kits",
        "//device/soc/hisilicon/hi3861v100/hi3861_adapter/hals/communication/wifi_lite/wifiservice",
        "//device/soc/hisilicon/hi3861v100/hi3861_adapter/kal",
    ]
}
```

5、驱动自带的屏幕检测样例

app/ssd1306/examples/ssd1306_demo.c	主程序
app/ssd1306/examples/ssd1306_tests.c	各种测试子程序

5、驱动自带的屏幕检测样例 app/ssd1306/examples/ssd1306_demo.c

```c
#include <stdio.h>
#include <unistd.h>

#include "ohos_init.h"          // OpenHarmony 初始化接口
#include "cmsis_os2.h"          // CMSIS-OS2 操作系统接口
#include "iot_gpio.h"           // GPIO 控制接口
#include "iot_pwm.h"            // PWM 控制接口（未使用）
#include "iot_i2c.h"            // I2C 控制接口
#include "iot_errno.h"          // 错误码定义

#include "ssd1306.h"            // SSD1306 OLED 驱动头文件
#include "ssd1306_tests.h"      // SSD1306 测试函数

#include "hi_io.h"              // 海思芯片 IO 控制
```

5、驱动自带的屏幕检测样例　app/ssd1306/examples/ssd1306_demo.c

```
void TestDrawChinese1(void)
```
显示16*16字符的样例

```
void TestDrawChinese2(void)
```
显示12*12字符的样例

```
void Ssd1306TestTask(void* arg)
```
主函数

5、驱动自带的屏幕检测样例　app/ssd1306/examples/ssd1306_tests.c

```
void ssd1306_TestBorder()       边框测试
void ssd1306_TestFonts()        字体大小显示测试
void ssd1306_TestFPS()          帧率测试
void ssd1306_TestLine()         直线测试
void ssd1306_TestRectangle()    矩形测试
void ssd1306_TestCircle()       圆形测试
void ssd1306_TestArc()          圆弧测试
void ssd1306_TestPolyline()     折线测试
void ssd1306_TestBitmap(void)   位图显示测试
```

四、OLED实训练习：OLED屏幕显示学号姓名