



OLED显示模块

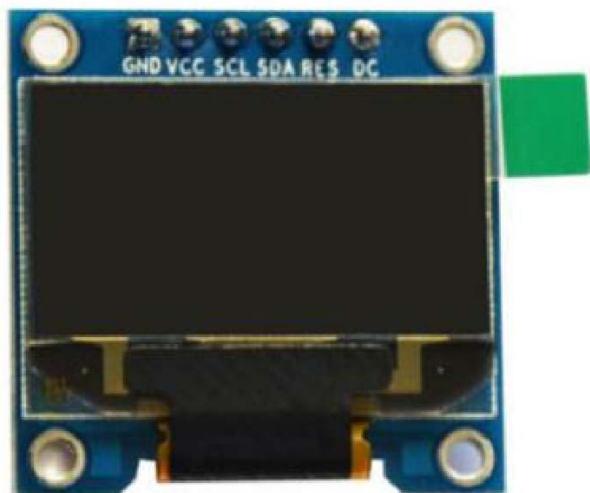


人机接口

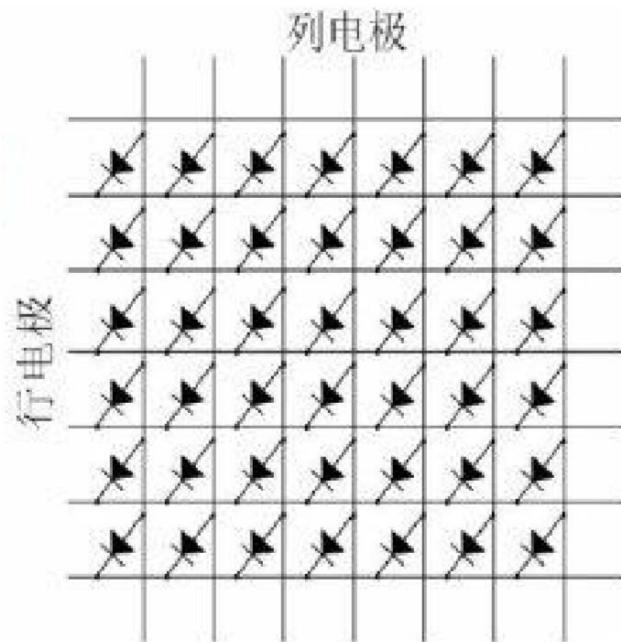
- 人机接口就是人与机器之间的沟通渠道，机器上的按键、电脑上的鼠标、显示器等都属于人机接口
- 输入设备
 - 人->机
 - 键盘、鼠标、触摸板、麦克风，手柄、遥控器
- 输出设备
 - 机->人
 - 显示器、蜂鸣器、LED。。。

OLED原理

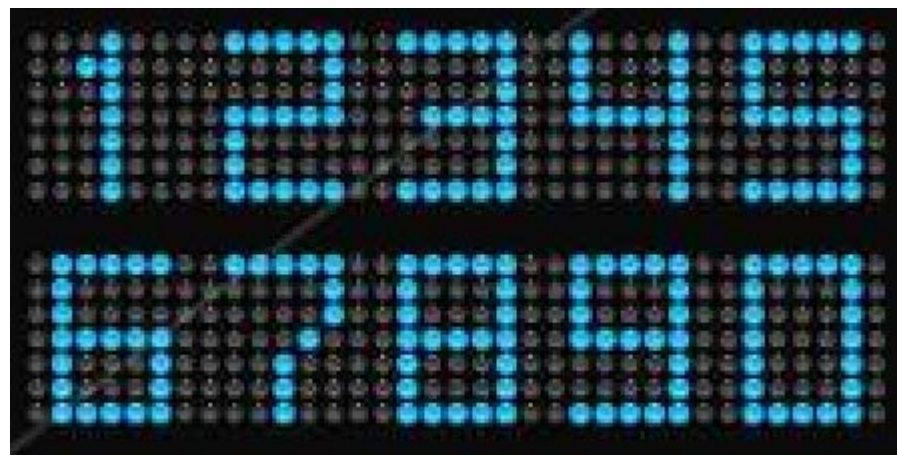
- 有机发光二极管（Organic Light-Emitting Diode, OLED）显示器的本质是由一个一个微小的LED发光管组成，这些LED排成阵列，通过对它们亮灭的控制，让它们呈现各种图案和文字



OLED 12864显示屏



OLED 显示屏原理示意



使用点阵的方式显示数字

OLED模块

- OLED模块已经封装在OLED.ino中。并在范例中完整给出，我们主要关注模块内函数的调用方法

OLED模块初始化。
几乎所有模块都要在setup()函数中初始化后才能正常使用

```
OLED_Init();
```

模块函数
在OLED上显示字符串

字符串首字符所在列

字符串首字符所在行

字符串常量
注意它外面包了一层双引号

```
OLED12864_ShowStr(0,0,"Hello World!");
```

字符串变量
需要先定义再使用

```
OLED12864_ShowStr(0,1,a);
```

字符串处理

- 字符串从本质上是一系列字符组成的数组，大部分时候我们直接把它看成一个变量

字符串初始化
char是字符类型
[15]代表这是一个
长度为15的数组

可以在定义字符串
变量的同时给字符串
赋初始值，赋初始
值的时候[]内可
以不写字符串长度

系统库函数
字符串格式化填充

```
char a[15];  
char b[]="Hello World!";  
sprintf(a,"LoopCnt:%d",loopCnt);
```

填充完毕后字符串
存放的变量

填充字符串的主体，
%d代表这个位置将填
入一个整数数字（可
以理解成填空题的空）

这是一个变量，它的值
将被填充到前面参数中
%d的位置

- 想一想，如果loopCnt变量等于7，这段程序运行完后a中的字符串是什么样子？

范例解析

static 说明这个变量是静态变量，它不会因为函数执行完毕而被系统清除，等下次这个函数再次运行时，它的值将会保持上次运行这个函数时的值

OLED模块初始化。几乎所有模块都要在setup()函数中初始化后才能正常使用

在定义一个变量时可以直接给它赋一个初始值

loopCnt后的++说明这里在读取loopCnt的值以后loopCnt内变量要+1
如果loopCnt在执行前的值为7，那这句执行后读取到的值为7，loopCnt内的值为8

- 任务：
 - 使用OLED显示环境亮度

```
void setup() {  
    OLED_Init(); //OLED初始化  
}  
  
void loop() {  
    static int loopCnt=0; //定义一个变量记录主循环的次，初始化0  
    OLED12864_ShowStr(0,0,"Hello World!"); //用OLED输出固定的字符串  
    char a[15]; //定义一个字符串变量（字符数组）  
    sprintf(a,"LoopCnt:%d",loopCnt++); //填充字符串变量  
    OLED12864_ShowStr(0,1,a); //用OLED输出字符串变量  
    delay(1000); //延时1000ms  
}
```