

# 树莓派开发

## 22 使用树莓派液晶屏 (arduino版)

# 1、1602液晶屏介绍

- 本次试验使用arduino直接驱动1602液晶显示文字
- 1602液晶在应用中非常广泛，最初的1602液晶使用的是HD44780控制器，现在各个厂家的1602模块基本上都是采用了与之兼容的IC，所以特性上基本都是一致的。

- 1602LCD主要技术参数：
- 显示容量为 $16 \times 2$ 个字符；
- 芯片工作电压为 $4.5 \sim 5.5V$ ；
- 工作电流为 $2.0mA$ （ $5.0V$ ）；
- 模块最佳工作电压为 $5.0V$ ；
- 字符尺寸为 $2.95 \times 4.35$   
（ $W \times H$ ）mm。



## 2、1602液晶屏的接口介绍

### □ 1602液晶接口引脚定义

编号	符号	引脚说明	编号	符号	引脚说明
1	VSS	电源地	9	D2	Date I/O
2	VDD	电源正极	10	D3	Date I/O
3	VL	液晶显示偏压信号	11	D4	Date I/O
4	RS	数据/命令选择端 (V/L)	12	D5	Date I/O
5	R/W	读/写选择端 (H/L)	13	D6	Date I/O
6	E	使能信号	14	D7	Date I/O
7	D0	Date I/O	15	BLA	背光源正极
8	D1	Date I/O	16	BLK	背光源负极

From [www.geek-workshop.com](http://www.geek-workshop.com)



From [www.geek-workshop.com](http://www.geek-workshop.com)

## 2、1602液晶屏接口介绍

### □ 接口说明：

1、两组电源：一组是模块的电源，一组是背光板的电源。一般均使用**5V**供电。使用**3.3V**供电也可以工作，就是暗一点。

2、**VL**是调节对比度的引脚，串联不大于**5KΩ**的电位器进行调节。实验使用**1KΩ**的电阻来设定对比度。可用连接分高电位与低电位接法，本次使用低电位接法，串联**1KΩ**电阻后接**GND**。

注意：不同液晶的对比度电阻是不同的，最好是接一个电位器进行测试，本次实验使用的**1KΩ**电阻在其他液晶上不一定正确。但不论阻值大小，仅影响亮度，没有风险。

3、**RS** 是很多液晶上都有的引脚，是命令/数据选择引脚。该脚电平为高时表示将进行数据操作；为低时表示进行命令操作。

## 2、1602液晶屏接口介绍

### □ 接口说明：

- 4、RW 也是很多液晶上都有的引脚，是读写选择端。该脚电平为高是要对液晶进行读操作；为低时表示进行写操作。
- 5、E同样很多液晶模块有此引脚，通常在总线上信号稳定后给一正脉冲通知把数据读走，在此脚为高电平的时候总线不允许变化。
- 6、D0—D7共8位双向并行总线，用来传送命令和数据。
- 7、BLA是背光源正极，BLK是背光源负极。

## 2、1602液晶屏接口介绍

□ 综合上述各脚的配合，1602液晶的基本操作分以下四种：

读状态	输入	RS=L, R/W=H, E=H	输出	D0~D7=状态字
写指令	输入	RS=L, R/W=L, D0~D7=指令码, E=高脉冲	输出	无
读数据	输入	RS=H, R/W=H, E=H	输出	D0~D7=数据
写数据	输入	RS=H, R/W=L, D0~D7=数据, E=高脉冲	输出	无

From [www.geek-workshop.com](http://www.geek-workshop.com)

- 本实验有用到写命令的（如清屏），也有用到写数据的
- 1602直接与arduino通信，根据产品手册描述，分8位连接法与4位连接法，一般采用4位连接法比较多。
- 本实验采用的也是4位连接法。



### 3、1602液晶屏的硬件连接

- 逻辑电路图：几乎所有接线图都是类似的，区别仅在使用哪4个数据线（4线方法），因为其他脚是不可选择的。

### 3、1602液晶屏的硬件连接

□ LCD的脚位连接说明：以下图为例，是从左到右第一脚开始数起：

1 Vss 接地 (0V)

2 Vdd 电源 (+5V)

3 Vo或与Vee - 对比度调整 (0-5V), 可接一颗1K的电阻或接一个5K的电位器

4 RS Register Select: 参数：{1: D0 – D7 设置为1时当作资料解释 0: D0 – D7 设置为0时当作指令解释}



5 R/W Read/Write

mode: 参数：{1: 从LCD读取资料 0: 写资料到LCD(一般情况下很少从LCD读取资料，把这个脚接地可省I/O脚位)}



### 3、1602液晶屏的硬件连接

□ 6 E Enable

7 D0 Bit 0 LSB

8 D1 Bit 1

9 D2 Bit 2

10 D3 Bit 3

11 D4 Bit 4

12 D5 Bit 5

13 D6 Bit 6

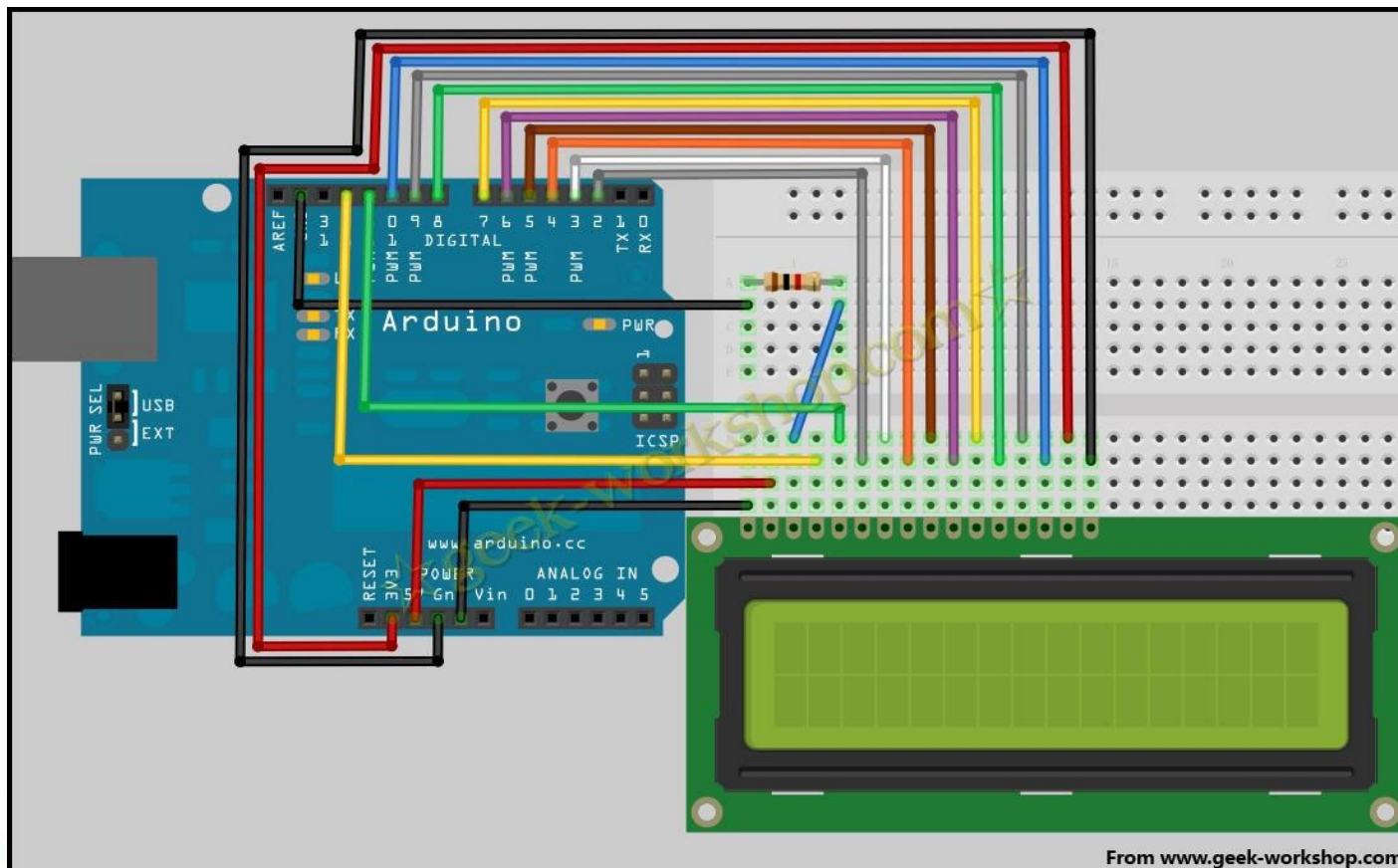
14 D7 Bit 7 MSB

15 A+ 背光(串联一个1K的电阻接地或串联一个5K的电位器调整背光)

16 K- 背光(GND)

### 3、1602液晶屏的硬件连接

□ 硬件连接方式如下图（这是比较好看的图）：



## 4、1602液晶屏的软件1

□ Arduino上的代码如下：

```
//LCD message board
#include <LiquidCrystal.h> //加载LCD库，这个库是Arduino的标准自带库
//LiquidCrystal(rs, rw, enable, d4, d5, d6, d7) //以LCD为对象的接口定义
LiquidCrystal lcd(2, 3, 4, 9, 10, 11, 12); //以Arduino为对象的接口定义
void setup()
{
  Serial.begin(9600); //使能串口
  lcd.begin(2, 16); //设置LCD显示的数目。16 X 2：16格2行。
  lcd.clear();
  lcd.setCursor(0,0); //使光标回到0,0位置
  lcd.print("hello,world!"); //显示hello,world!
  lcd.setCursor(0,1); //光标换行
}
```

## 4、1602液晶屏的软件1

### □ 注意代码中的接口定义方式

□ LiquidCrystal(rs, rw, enable, d4, d5, d6, d7) //此方式定义LCD接口，目前编译不支持。

□ LiquidCrystal lcd(2, 3, 4, 9, 10, 11, 12); //此方式定义Arduino接口

□ 其中：2、3、4、9、10、11、12是指arduino对应的d2、d3、d4、d9、d10、d11、d12口

□ 另一常用的接线方式有：

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

□ Arduino由于数据口比较多，所以可以选用上述两种不同的方式，而LCD则没有。

## 4、1602液晶屏的软件2

```
void loop()
{
    if (Serial.available()) //如果串口准备好
    {
        char ch = Serial.read(); //读串口
        Serial.write(ch); //向串口回送接收到的字符（调试用）
        if (ch == '#') //约定输入#为清屏
        {
            lcd.clear(); //清屏
        }
        else if (ch == '/')
        {
            lcd.setCursor(0,1); //约定输入/为换行
        }
        else
        {
            lcd.write(ch); //否则，显示输入字符
        }
    }
}
```

## 5、扩展作业

- 是否非常简单？这是因为使用了LCD库<LiquidCrystal.h>的原因。
- 将上节温度和湿度采样与本节液晶屏显示结合起来
- 在液晶屏上显示实时得到的温度和湿度



## 6、LiquidCrystal库简介

LiquidCrystal()——定义你的LCD的接口：各个引脚连接的I/O口编号，格式为

LiquidCrystal(rs, enable, d4, d5, d6, d7)

LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)

LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)

LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

begin()——定义LCD的长宽（n列×n行），格式lcd.begin(cols, rows)

clear()——清空LCD，格式lcd.clear()

home()——把光标移回左上角，即从头开始输出，格式lcd.home()

setCursor()——移动光标到特定位置，格式lcd.setCursor(col, row)

write()——在屏幕上显示内容（必须是一个变量，如” Serial.read()”），格式  
lcd.write(data)

print()——在屏幕上显示内容（字母、字符串，等等），格式lcd.print(data)

lcd.print(data, BASE)

cursor()——显示光标（一条下划线），格式lcd.cursor()

noCursor()——隐藏光标，格式lcd.noCursor()

blink()——闪烁光标，格式lcd.blink()

noBlink()——光标停止闪烁，格式lcd.noBlink()

display()——（在使用noDisplay()函数关闭显示后）打开显示（并恢复原来内容），格式lcd.display()

## 6、LiquidCrystal库简介

`noDisplay()`——关闭显示，但不会丢失原来显示的内容，格式为`lcd.noDisplay()`

`scrollDisplayLeft()`——把显示的内容向左滚动一格，格式`lcd.scrollDisplayLeft()`

`scrollDisplayRight()`——把显示的内容向右滚动一格，格式为  
`lcd.scrollDisplayRight()`

`autoscroll()`——打开自动滚动，这使每个新的字符出现后，原有的字符都移动一格：如果字符一开始从左到右（默认），那么就往左移动一格，否则就向右移动，格式`lcd.autoscroll()`

`noAutoscroll()`——关闭自动滚动，格式`lcd.noAutoscroll()`

`leftToRight()`——从左往右显示，也就是说显示的字符会从左往右排列（默认），但屏幕上已经有的字符不受影响，格式`lcd.leftToRight()`

`rightToLeft()`——从右往左显示，格式`lcd.rightToLeft()`

`createChar()`——自造字符，最多 $5 \times 8$ 像素，编号0-7，字符的每个像素显示与否由数组里的数（0-不显示，1-显示）决定，格式`lcd.createChar(num, data)`，有点难理解，可以看一个例子