

附录 1

1602LCD 主要技术参数:

显示容量为 16×2 个字符;

芯片工作电压为 $4.5 \sim 5.5\text{V}$;

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 2.95×4.35 ($W \times H$) mm。

1602 采用标准的 16 脚接口, 其中:

第 1 脚: VSS 为地电源

第 2 脚: VDD 接 5V 正电源

第 3 脚: VO 为液晶显示器对比度调整端, 接正电源时对比度最弱, 接地电源时对比度最高, 对比度过高时会产生“鬼影”, 使用时可以通过一个 10K 的电位器调整对比度

第 4 脚: RS 为寄存器选择, 高电平时选择数据寄存器、低电平时选择指令寄存器。

第 5 脚: R/W 为读写信号线, 高电平时进行读操作, 低电平时进行写操作。当 RS 和 RW 共同为低电平时可以写入指令或者显示地址, 当 RS 为低电平 RW 为高电平时可以读信号, 当 RS 为高电平 RW 为低电平时可以写入数据。

第 6 脚: E 端为使能端, 当 E 端由高电平跳变成低电平时, 液晶模块执行命令。

第 7~14 脚: D0~D7 为 8 位双向数据线。

第 15 脚: 背光电源正极

第 16 脚: 背光电源负极

LCD 可按 4 位模式连接, 如图 1 所示。

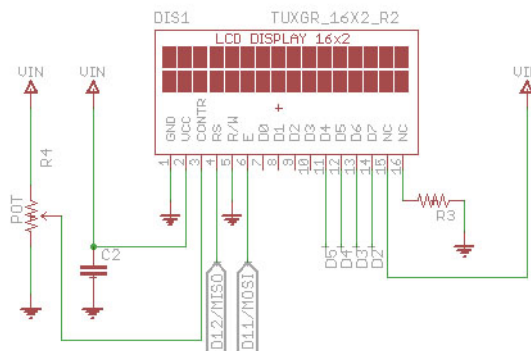


图 F.1 LCD 4 位连接模式

程序模版如下:

```
#include <LiquidCrystal.h>
LiquidCrystal iLCD (12, 11, 5, 4, 3, 2);
void setup ()
{
    iLCD.begin (16, 2 );
    // your code here...
}
void loop ()
{
    // your code here ...
}
```

在 Arduino 的安装目录下\libraries\LiquidCrystal 可以查看到函数的原型

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) //4 位连接模式

LiquidCrystal(rs, rw, enable, d4, d5, d6, d7) //4 位连接模式

LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7) //8 位连接模式

LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7) //8 位连接模式

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()

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()

附录 2

图 F.2 示出了当旋转编码器工作时输出 A 和 B 的信号是如何变化的。当顺时针旋转旋钮时，脉冲会变化，并且会在图中从左向右移动；当逆时针旋转旋钮时，脉冲会在图中从右向左移动。

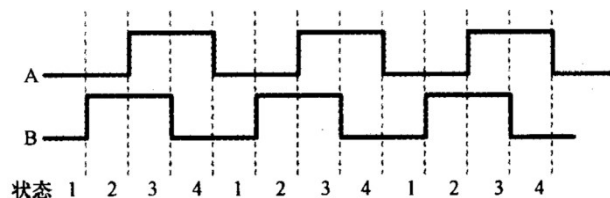


图 F.2 旋转编码器输出的脉冲波形

因此，如果 A 和 B 都是低电平，然后 B 变为高电平（从状态 1 到状态 2），这表明旋钮已经顺时针旋转；如果 A 低、B 高，然后 A 由低变高（从状态 2 到状态 3）也同样表明旋钮顺时针旋转；如果 A 高、B 低，然后 B 变高，已经从状态 4 到状态 3，表明旋钮逆时针旋转。

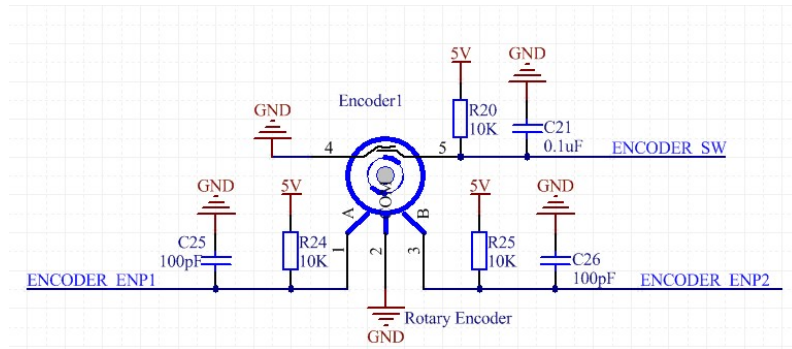


图 F.3 旋转编码器接线图

图 F.3 为 EC11 旋转编码器接线图，1、3 脚为 A、B 信号，4、5 脚为开关，2 脚接地。