

# Arduino "LiquidCrystal.h"(LCD) library

## 简介

通过使用LiquidCrystal库，您可以使用Arduino开发板控制基于Hitachi HD44780芯片组或兼容LiquidCrystal库的液晶显示器（LCD），如1602液晶显示器。该库可以以4线或8线模式工作（即除了rs，enable和rw控制线之外还可以使用4或8条数据线连接运行。

## 函数介绍

### 函数一览

```
1  1. LiquidCrystal();           //构造函数
2  2. begin();                   //指定显示屏尺寸
3  3. clear();                   //清屏并将光标置于左上角
4  4. home();                    //将光标置于左上角（不清屏）
5  5. setCursor();               //将光标置于指定位置
6  6. write();                   //（在光标处）显示一个字符
7  7. print();                   //显示字符串
8  8. cursor();                  //显示光标（就是一个下划线）
9  9. noCursor();                //不显示光标
10 10. blink();                  //光标闪烁（和8,9一起使用时不保证效果）
11 11. noBlink();                //光标不闪烁
12 12. noDisplay();              //关闭显示，但不会丢失内容
13 13. display();                //（使用noDisplay()后）恢复显示
14 14. scrollDisplayLeft();        //将显示的内容向左滚动一格
15 15. scrollDisplayRight();       //将显示的内容向右滚动一格
16 16. autoscroll();             //打开自动滚动（具体解释往下拉）
17 17. noAutoscroll();           //关闭自动滚动
18 18. leftToRight();            //从左向右显示内容（默认）
19 19. rightToLeft();            //从右向左显示内容
20 20. createChar();             //大奥义·字符创造之术（往下拉）
```

## LiquidCrystal

### 介绍

创建一个LiquidCrystal的实例对象，可以使用4线或者8线方式作为数据线(请注意,还需要指令线).当使用4线接法的时候，d0——d3不接线，RW引脚可接地而不用接在开发板的某个引脚上.如果这样接的话,就省略了在函数中的rw参数传入.

### 语法

```
1  /*四线接法*/
2  LiquidCrystal(rs, enable, d4, d5, d6, d7); //无rw(只进行写操作)
3  LiquidCrystal(rs, rw, enable, d4, d5, d6, d7); //有rw
4
5  /*八线接法*/
6  LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7); //无rw
7  LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7); //有rw
```

## 参数

- rs: rs连接的开发板引脚编号
- rw: rw连接的开发板引脚编号
- enable:enable连接的开发板引脚编号
- d0, d1, d2, d3, d4, d5, d6, d7: 连接的开发板引脚编号

## 返回值

无

# begin

## 介绍

自定义显示屏的显示参数（显示屏支持的列数和行数）。

根据LCD显示屏设备的不同需进行定义和调整

## 语法

```
1 | lcd.begin(cols, rows);
```

## 参数

- cols: 显示屏支持显示的列数(1602是16列)
- rows: 显示屏支持显示的行数(1602是2行)

## 返回值

无

# clear

## 介绍

清除LCD屏幕上内容,并将光标置于左上角(0,0)位置处。

## 语法

```
1 | lcd.clear();
```

## 参数

无

## 返回值

无

# home

## 介绍

将光标定位在屏幕左上方开始位置(0,0). 也就是说,在将后续文本显示到LCD时,将从该位置开始显示.

#如果要同时清除屏幕上的内容,请使用`clear()`函数代替.

## 语法

```
1 | lcd.home();
```

## 参数

无

## 返回值

无

## 例程

```
1 | #include<LiquidCrystal.h>
2 |
3 | // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
4 | const int rs=12,en=11,d4=5,d5=4,d6=3,d7=2;
5 | LiquidCrystal lcd(rs,en,d4,d5,d6,d7);
6 |
7 | void setup(){
8 |     // lcd初始化,同时设置lcd屏幕的列数和行数(宽和高)
9 |     lcd.begin(16,2);
10 | }
11 |
12 | void loop(){
13 |     // 设置光标位置并打印文字
14 |     lcd.setCursor(0, 0);
15 |     lcd.print("hello!");
16 |     lcd.setCursor(0, 1);
17 |     lcd.print("TaichiMaker");
18 |
19 |     delay(3000);
20 |     // 从(0,0)位置开始打印数字0~9
21 |     lcd.home();
22 |     for (int thisChar = 0; thisChar < 10; thisChar++) {
23 |         lcd.print(thisChar);
24 |         delay(500);
25 |     }
26 | }
```

# setCursor

## 介绍

定位LCD光标位置；也就是说，设置在LCD显示屏中随后要显示文本的位置。(可用在操作电脑时的鼠标位置来理解此函数)

## 语法

```
1 | lcd.setCursor(col, row);
```

## 参数

col: 光标所在的列 (第一列为0)

row: 行处定位光标 (0是第一行)

## 返回值

无

# write

## 介绍

写入数据到LCD屏幕

## 语法

```
1 | lcd.write(data);
```

## 参数

data: 想要在LCD显示的数据(in byte)

## 返回值

返回值类型:byte

返回写入的数据长度大小

# print

## 介绍

打印文字到LCD屏幕中

## 语法

```
1 | lcd.print(data);  
2 | lcd.print(data, BASE);
```

## 参数

- lcd:LiquidCrystal实例化对象
- data: 要在LCD显示的数据;参数类型 (char, byte, int, long, string)
- BASE (可选): 要打印到LCD的数据进制类型: 二进制 (base 2), 十进制 (base 10), 八进制 (base 8), 16进制 (base 16).

## 返回值

返回值类型:byte

print () 将返回写入的字节数

## cursor

### 介绍

显示光标(字符将在光标所在位置进行显示)

### 语法

```
1 | lcd.cursor();
```

### 参数

- lcd: LiquidCrystal实例化对象

### 返回值

无

## nocursor

### 介绍

隐藏光标

### 语法

```
1 | lcd.nocursor();
```

### 参数

- lcd: LiquidCrystal实例化对象

### 返回值

无

# blink

## 介绍

显示闪烁的LCD光标。如果与cursor（）函数结合使用，则结果将取决于您的LCD屏幕模块。

## 语法

```
1 lcd.blink();
```

## 参数

- lcd: LiquidCrystal实例化对象

## 返回值

无

## 例程

```
1 #include <LiquidCrystal.h>
2
3 // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
4 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
5 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
6
7 void setup()
8 {
9     // lcd初始化,同时设置lcd屏幕的列数和行数(宽和高)
10    lcd.begin(16, 2);
11    // 向LCD打印"hello, world!"
12    lcd.print("hello, world!");
13 }
14
15 void loop()
16 {
17     // 关闭光标闪烁
18    lcd.noBlink();
19    delay(3000);
20    // 打开光标闪烁
21    lcd.blink();
22    delay(3000);
23 }
```

# noblink

## 介绍

关闭光标闪烁。

## 语法

```
1 | lcd.noBlink();
```

## 参数

- lcd: LiquidCrystal实例化对象

## 返回值

无

## 例程

见blink

# display

## 介绍

如果您在程序中调用了 `noDisplay()` 隐藏LCD上显示内容后,可以调用本函数恢复显示文本和(光标).

## 语法

```
1 | lcd.display();
```

## 参数

- lcd: LiquidCrystal实例化对象

## 返回值

无

## 例程

```
1 | #include <LiquidCrystal.h>
2 |
3 | // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
4 | const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
5 | LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
6 |
7 | void setup()
8 | {
9 |     // lcd初始化,同时设置lcd屏幕的列数和行数(宽和高)
10 |    lcd.begin(16, 2);
11 |
12 |    // 向LCD打印"TaichiMaker"
13 |    lcd.setCursor(0, 0);
14 |    lcd.print("TaichiMaker");
15 |    lcd.setCursor(0, 1);
16 |    lcd.print("taichi-maker.com");
17 | }
18 |
19 | void loop()
```

```
20 {  
21 // 隐藏LCD的显示内容  
22 lcd.noDisplay();  
23 delay(1000);  
24 // 恢复显示文本内容  
25 lcd.display();  
26 delay(1000);  
27 }
```

## noblink

### 介绍

隐藏LCD显示屏显示内容，不会丢失当前显示在LCD上的文本内容。

### 语法

```
1 lcd.noDisplay();
```

### 参数

- lcd: LiquidCrystal实例化对象

### 返回值

无

### 例程

见display

## scrollDisplayLeft

### 介绍

将显示内容（文本和光标）向左滚动一格

### 语法

```
1 lcd.scrollDisplayLeft();
```

### 参数

- lcd: LiquidCrystal实例化对象

### 返回值

无



## 例程

```
1  #include <LiquidCrystal.h>
2
3  // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
4  const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
5  LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
6
7  void setup()
8  {
9      // lcd初始化,同时设置lcd屏幕的列数和行数(宽和高)
10     lcd.begin(16, 2);
11     // 向LCD打印"hello, world!"
12     lcd.print("hello, world!");
13 }
14
15 void loop()
16 {
17     //向左滚动13个位置 (即字符串长度)
18     //将其向屏幕外移动:
19     for (int positionCounter = 0; positionCounter < 13; positionCounter++)
20     {
21         // 向左滚动一格
22         lcd.scrollDisplayLeft();
23         //延时150毫秒
24         delay(500);
25     }
26     // 向右滚动29个位置 (字符串长度13+显示长度16)
27     // 向右滚动出屏幕
28     for (int positionCounter = 0; positionCounter < 29; positionCounter++)
29     {
30         //滚动一格单位
31         lcd.scrollDisplayRight();
32         // 些许延时
33         delay(500);
34     }
35     // 向左滚动16个位置 (显示长度)
36     // 移动回中心
37     for (int positionCounter = 0; positionCounter < 16; positionCounter++)
38     {
39         // 向左滚动一格
40         lcd.scrollDisplayLeft();
41         // 稍微延迟以更好的显示效果
42         delay(500);
43     }
44     // 在循环结束时延时
45     delay(1000);
46 }
```

# scrollDisplayRight

## 介绍

将显示内容（文本和光标）向左滚动一格

## 语法

```
1 | lcd.scrollDisplayRight();
```

## 参数

- lcd: LiquidCrystal实例化对象

## 返回值

无

## 例程

见scrollDisplayleft

# autoscroll

## 介绍

打开LCD的自动滚动。

这会使输出到显示器的每个字符将前一个字符推过一个空格。

如果当前文本方向是从左到右（默认），则显示时向左滚动；

否则，向右滚动。如果当前方向是从右到左，则显示屏向右滚动。

换句话说,当显示信息为单个字符时,会使得该字符总是输出在LCD上的同一个位置.

## 语法

```
1 | lcd.autoscroll();
```

## 参数

- lcd: LiquidCrystal实例化对象

## 返回值

无

## 例程

```
1 | #include <LiquidCrystal.h>
2 |
3 | // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
4 | const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
5 | LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
6 |
7 | void setup() {
8 |     // lcd初始化,同时设置lcd屏幕的列数和行数(宽和高)
```

```

9     lcd.begin(16, 2);
10 }
11
12 void loop() {
13     // 设置光标在(0,0)位置处
14     lcd.setCursor(0, 0);
15     // 使用for循环打印数字0~9
16     for (int thisChar = 0; thisChar < 10; thisChar++)
17     {
18         lcd.print(thisChar);
19         delay(500);
20     }
21
22     // 设置光标处于(16,1)位置处
23     lcd.setCursor(16, 1);
24     // 设置lcd屏幕自动滚动
25     lcd.autoscroll();
26     // 打印数字0~9
27     for (int thisChar = 0; thisChar < 10; thisChar++)
28     {
29         lcd.print(thisChar);
30         delay(500);
31     }
32     // 关闭自动滚动功能
33     lcd.noAutoscroll();
34
35     // 清屏并继续下一个循环
36     lcd.clear();
37 }

```

## noAutoscroll

### 介绍

与 `autoscroll` 函数相反,本函数功能为关闭LCD的自动滚动。

### 语法

```
1 lcd.noAutoscroll();
```

### 参数

- lcd: LiquidCrystal实例化对象

### 返回值

无

### 例程

见Autoscroll

# rightToLeft

## 介绍

设置文本写入LCD的方向为从右向左（默认是从左向右）。也就是说，后续字符将会由右至左写入并显示，但不影响先前的文本的显示。

## 语法

```
1 | lcd.rightToLeft();
```

## 参数

- lcd: LiquidCrystal实例化对象

## 返回值

无

## 例程

见示例4左右流动

# leftToRight

## 介绍

将写入LCD的文本的方向设置为从左到右（默认）。这意味着写入显示器的后续字符将从左到右显示，但不会影响先前输出的文本。

## 语法

```
1 | lcd.leftToRight();
```

## 参数

- lcd: LiquidCrystal实例化对象

## 返回值

无

## 例程

见示例4左右流动

# createChar

## 介绍

创建一个可在LCD上使用的自定义字符（字形）。最多支持8个5×8像素的字符（编号0至7）。每个自定义字符的外观由八个字节的数组指定，每行占用一个字符。所在的行最低由5个有效像素点决定.若要在屏幕显示自定义字符,请使用write()函数.(参数为字符的编号0-7)

## 语法

```
1 lcd.createChar(num, data);
```

## 参数

- lcd: LiquidCrystal实例化对象
- num: 所创建字符的编号(0-7)
- data: 字符的像素数据

## 返回值

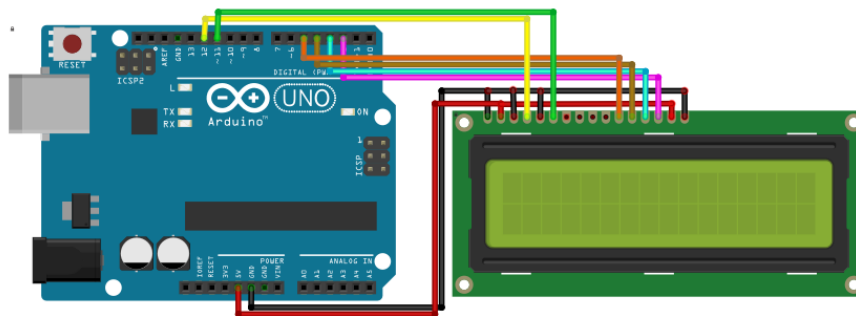
无

## 例程

```
1 #include <LiquidCrystal.h>
2
3 // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
4 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
5 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
6
7 // 定义一个自定义字符数组变量smiley[]
8 byte smiley[8] = {
9     B00000,
10    B10001,
11    B00000,
12    B00000,
13    B10001,
14    B01110,
15    B00000,
16 };
17
18 void setup() {
19     // 创建一个可在LCD上使用的自定义字符
20     lcd.createChar(0, smiley);
21     // 设置LCD显示尺寸
22     lcd.begin(16, 2);
23     // 使用write函数进行写入显示在LCD
24     lcd.write(byte(0));
25 }
26
27
28 void loop()
29 {}
```

# 例程 (LCD1602)

## Arduino使用LCD1602



VSS	-	GND
VDD	-	5V
V0	-	GND
RS	-	12
RW	-	GND
E	-	11
D4	-	5
D5	-	4
D6	-	3
D7	-	2
A	-	5V
K	-	GND

太极创客

(www.taichi-maker.com)

LCD602-UNO连线图

### 示例1 文字显示

使用print函数在LCD屏幕打印字符串信息以及当前设备运行的时间(秒)

```
1 #include<LiquidCrystal.h>
2
3 // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
4 const int rs=12,en=11,d4=5,d5=4,d6=3,d7=2;
5 LiquidCrystal lcd(rs,en,d4,d5,d6,d7);
6
7 void setup()
8 {
9     lcd.begin(16,2);
10    lcd.print("Hello world!");
11 }
12
13 void loop() {
14     // 设置光标位置在第0列,第1行
15     // 注意,行和列都是从0开始的
16     lcd.setCursor(0, 1);
17     //打印自开发板重置以来的秒数:
18     lcd.print(millis()/1000);
19 }
```

### 示例2 滚动显示

本示例将告诉您如何使用 `autoscroll()` 和 `noAutoscroll()` 函数向左或向右滚动显示的所有字符。

每次添加字母时, 通过调用 `autoscroll()` 函数会将所有文本向左移动一格

如果使用 `noAutoscroll()` 函数则会停止自动滚动

本示例在自动滚动关闭的条件下打印字符0到9, 然后将光标移到右下角, 打开自动滚动, 然后再次打印。

```
1 #include <LiquidCrystal.h>
2
```

```

3 // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
4 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
5 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
6
7 void setup() {
8     // lcd初始化,同时设置lcd屏幕的列数和行数(宽和高)
9     lcd.begin(16, 2);
10 }
11
12 void loop() {
13     // 设置光标在(0,0)位置处
14     lcd.setCursor(0, 0);
15     // 使用for循环打印数字0~9
16     for (int thisChar = 0; thisChar < 10; thisChar++) {
17         lcd.print(thisChar);
18         delay(500);
19     }
20 //<span style="color: #ffffff;">太极创客(www.taichi-maker.com)</span>
21     // 设置光标处于(16,1)位置处
22     lcd.setCursor(16, 1);
23     // 设置lcd屏幕自动滚动
24     lcd.autoscroll();
25     // 打印数字0~9
26     for (int thisChar = 0; thisChar < 10; thisChar++) {
27         lcd.print(thisChar);
28         delay(500);
29     }
30     // 关闭自动滚动功能
31     lcd.noAutoscroll();
32
33     // 清屏并继续下一个循环
34     lcd.clear();
35 }

```

### 示例3 串口输入到LCD

本示例将向您演示如何将串口的数据实时显示到LCD屏幕中,在此过程中,您可以在烧录程序后通过串口监视器发送数据并在LCD屏幕进行查看

```

1
2 #include <LiquidCrystal.h>
3
4 // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
5 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
6 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
7
8 void setup()
9 {
10     // lcd初始化,同时设置lcd屏幕的列数和行数(宽和高)
11     lcd.begin(16, 2);
12     // 初始化串口并设置波特率为9600
13     Serial.begin(9600);
14 }
15 //<span style="color: #ffffff;">太极创客(www.taichi-maker.com)</span>
16 void loop()
17 {
18     // 当串口收到数据时将数据写入到屏幕

```

```

19     if (Serial.available())
20     {
21         // 延时100毫秒以等待消息接收完整
22         delay(100);
23         // 清屏
24         lcd.clear();
25         // 读取串口缓冲区所以数据
26         while (Serial.available() > 0)
27         {
28             // 显示每一个字符到LCD显示屏
29             lcd.write(Serial.read());
30         }
31     }
32 }

```

#### 示例4 左右流动

本示例演示如何使用 `leftToRight()` 和 `rightToLeft()` 函数。这些函数控制文本从LCD流出的方式。

`rightToLeft()` 使文本从光标向左流动，就像显示是右对齐的一样。

`leftToRight()` 使文本从光标向右流动，就像显示是左对齐的一样。

本示例将打印字符 'a' 通过 l 从右到左，然后 m 通过 r 左到右，然后 s 通过 z 从右到左再次运行。

```

1  #include <LiquidCrystal.h>
2
3  // 创建lcd控制对象,并指定其引脚与Arduino控制板对应关系
4  const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
5  LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
6
7  // 定义一个'a'变量
8  int thisChar = 'a';
9
10 void setup()
11 {
12     // lcd初始化,同时设置lcd屏幕的列数和行数(宽和高)
13     lcd.begin(16, 2);
14     // 打开光标
15     lcd.cursor();
16 }
17
18 void loop()
19 {
20     //在'm'处转向
21     if (thisChar == 'm')
22     {
23         // go right for the next letter
24         lcd.rightToLeft();
25     }
26     // 在's'处再次反转
27     if (thisChar == 's')
28     {
29         // 向左走到下一个字母
30         lcd.leftToRight();
31     }

```



```
32 // 大于'z'则重置
33 if (thisChar > 'z')
34 {
35     // 回到(0,0)位置:
36     lcd.home();
37     //再次从a开始
38     thisChar = 'a';
39 }
40 // 打印字符
41 lcd.write(thisChar);
42 // 延时等待一秒
43 delay(1000);
44 // thisChar自增
45 thisChar++;
46 }
```

## 引用

---

[Arduino - LiquidCrystal库 - 太极创客 \(taichi-maker.com\)](http://taichi-maker.com/)

[Arduino - LiquidCrystal](#)