

第 3-3 讲：LCD1602 液晶显示模块的应用

1. 学习目的

1. 了解 LCD1602 显示模块涉及到的 3 种存储器：CGROM、CGRAM 和 DDRAM。
2. 掌握 LCD1602 显示模块的显示原理：DDRAM 和字符显示区域是如何对应的，写 DDRAM 时写入的数据和显示的字符如何关联起来的。
3. 根据 LCD1602 显示模块的读写时序和指令集，编写 LCD1602 显示模块的驱动程序。

2. LCD1602 液晶显示模块

2.1. LCD1602 简介

LCD1602 显示模块是一种专门用来显示字母、数字、符号等字符的点阵型液晶模块，因此，他也叫字符型液晶显示模块（以下简称 LCD1602）。

LCD1602 有 8 位数据总线 D0-D7，和 RS、R/W、EN 三个控制端口，工作电压为 5V。LCD1602 中的 1602 指的是液晶屏每行可以显示 16 个字符，共有 2 行，也就是 LCD1602 最多只能显示 $16 \times 2 = 32$ 个字符。

常用的 LCD1602 有蓝底色白字和黄绿底色黑字，如下图所示。

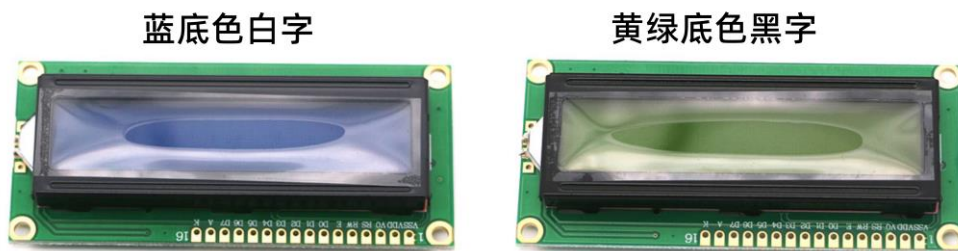


图 1：LCD1602 液晶显示模块

他们显示的效果如下图。

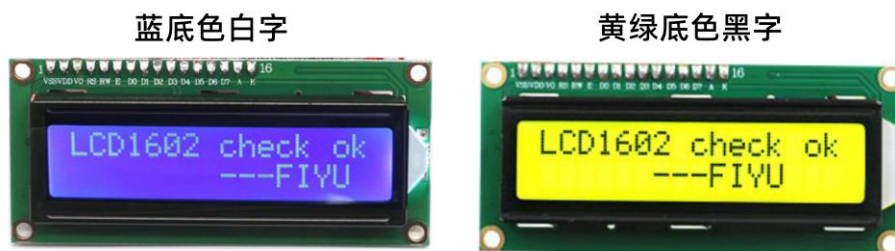


图 2：LCD1602 显示效果

LCD1602 模块主要特点如下：

- 1) 显示容量：16×2 个字符。
- 2) 芯片工作电压：+4.8 ~ +5.2V，最佳工作电压为 5.0V。
- 3) 工作电流（背光除外）1.7mA(max)。

- 4) 工作电流（背光） 24.0mA(max)。
- 5) 带有字符对比度调节和背光源。
- 6) 模块引脚数量：16 个。

LCD1602 模块引脚定义如下表所示。

表 1：LCD1602 液晶显示模块引脚定义

引脚编号	名称	功能描述
1	VSS	5V 电源负
2	VDD	5V 电源正
3	VO	LCD 驱动电压输入端
4	RS	指令/数据选择信号（低电平选择指令，高电平选择数据）
5	RW	读写选择信号（低电平选择写操作，高电平选择读操作）
6	E	使能信号
7	D0	8 位数据总线模式时为数据 0，4 位总线模式下 D0~D3 脚断开
8	D1	8 位数据总线模式时为数据 1，4 位总线模式下 D0~D3 脚断开
9	D2	8 位数据总线模式时为数据 2，4 位总线模式下 D0~D3 脚断开
10	D3	8 位数据总线模式时为数据 3，4 位总线模式下 D0~D3 脚断开
11	D4	8 位数据总线模式时为数据 4，4 位总线模式下 D0~D3 脚断开
12	D5	8 位数据总线模式时为数据 5，4 位总线模式下 D0~D3 脚断开
13	D6	8 位数据总线模式时为数据 6，4 位总线模式下 D0~D3 脚断开
14	D7	8 位数据总线模式时为数据 7，4 位总线模式下 D0~D3 脚断开
15	BLA	背光电源正极
16	BLK	背光电源负极

2.2. LCD1602 显示原理

学习 LCD1602 显示原理时，首先需要了解的是 LCD1602 片内的 3 种存储器，他们的作用和区别如下：

- 1) CGROM（字符发生器 ROM）：字模存储空间。液晶屏出厂时将一些标准的字符的字模编码固化到该空间，用户是无法修改的。
- 2) CGRAM（字符发生器 RAM）：他是用来扩展液晶屏可以显示的内容的，当我们要显示的字符在 CGROM 中没有，就可以自己设计字模编码，存储到该空间。通常，我们用不到 CGRAM。
- 3) DDRAM（显示数据寄存器）：我们在液晶屏上显示字符，主要就是和他打交道。显示字符时将字符编码写入到 DDRAM 中和 LCD1602 显示对应的地址，之后由液晶屏控制器根据字符编码从 CGROM 中找到对应的字符的字模并显示到液晶屏上。

那么，DDRAM 和 LCD1602 显示屏上的显示位置和显示内容是怎么对应起来的？换句话说，我知道 LCD1602 可以显示 2 行 16 列共 32 个字符，我也知道 DDRAM 中的字符编码

决定了显示内容，那么，如果我想在第 1 行的第 5 列显示数字“1”，怎么去实现显示？

看下面这张图，为了方便读者观察 LCD1602 的字符显示区域，特意调整了屏的对比度，可以很明显的看到屏上面分为 2 行 16 列共 32 个字符显示区域。



图 3：DDRAM 地址与 LCD 1602 显示屏上的显示位置的对应关系

DDRAM 的大小是 80 个字节，其中，地址 00H~0FH 依次对应 LCD1602 的第 1 行的第 1 列到第 16 列，地址 40H~4FH 依次对应 LCD1602 的第 2 行的第 1 列到第 16 列。可以看到，屏的每一个字符显示区域在 DDRAM 中都对应一个固定的地址，这个地址中写入的字符编码就决定了该字符显示区域显示的字符。

如果要在第 1 行的第 5 列显示数字“1”，只要向 DDRAM 的地址 04H 写入数字“1”的编码即可。同理，如果要在第 2 行的第 2 列显示字母“A”，只要向 DDRAM 的地址 41H 写入字母“A”的编码即可。

写入的字符编码怎么和显示的字模对应起来的？如让 LCD1602 其中的一个字符显示区域显示数字 1，应该向 DDRAM 对应的地址写入什么内容。

看下面的图片，下图是 LCD1602 的 CGROM 出厂时固化的字模。可以看到字符 1 对应的 D0~D7 高 4 位是 0011，低 4 位是 0001，组合后是 00110001b，对应的 16 进制为 0x31。而 0x31 正是字符 1 的 ASCII 码，对于 DDRAM 其他的字符，也是一样的，他们对应的 D0~D7 的数据也都是字符的 ASCII 码。

由此，我们知道，要在 LCD1602 的某个字符显示区域显示字符，只需将需要显示的字符的 ASCII 写入到 DDRAM 对应的地址即可。

Higher 4 bits Lower 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	1	P	`	P		-	夕	三	α	p
xxxx0001	(2)	!	1	A	Q	a	q	。	ア	チ	厶	ä	q
xxxx0010	(3)	"	2	B	R	b	r	「	イ	ツ	×	β	θ
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	ω
xxxx0100	(5)	\$	4	D	T	d	t	、	エ	ト	ト	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	1	σ	Ü
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	'	7	G	W	g	w	フ	キ	ヌ	ラ	g	π
xxxx1000	(1)	(8	H	X	h	x	イ	ク	ネ	リ	フ	̄
xxxx1001	(2))	9	I	Y	i	y	ウ	ケ	ル	ル	ˆ	y
xxxx1010	(3)	*	:	J	Z	j	z	エ	コ	ハ	レ	j	チ
xxxx1011	(4)	+	;	K	[k	[オ	サ	ヒ	ロ	*	斤
xxxx1100	(5)	,	<	L	¥	l	l	ャ	シ	フ	ワ	Φ	円
xxxx1101	(6)	-	=	M]	m	}	ユ	ズ	へ	ン	も	÷
xxxx1110	(7)	.	>	N	^	n	→	ヨ	セ	ホ	ゝ	ñ	
xxxx1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	°	ö	■

图 4: 字符码与字模的对应关系

了解了 LCD1602 的基本显示原理后, 接下来需要知道的是 LCD1602 的读写时序和指令集, 这些是和编程直接相关的。通过 LCD1602 的读写时序, 我们能知道如何通过硬件接口将输入发送给 LCD1602 或从 LCD1602 读取数据, 通过指令集, 我们能够知道, 让 LCD1602 实现不同的显示, 需要发送什么指令。

2.3. LCD1602 的读写时序

LCD1602 的基本操作分为四种：读状态、读数据、写命令和写数据。

- 1) 读状态：输入 RS=0, RW=1, E=高脉冲。输出：D0—D7 为状态字。
- 2) 读数据：输入 RS=1, RW=1, E=高脉冲。输出：D0—D7 为数据。

LCD1602 的读时序如下图所示。

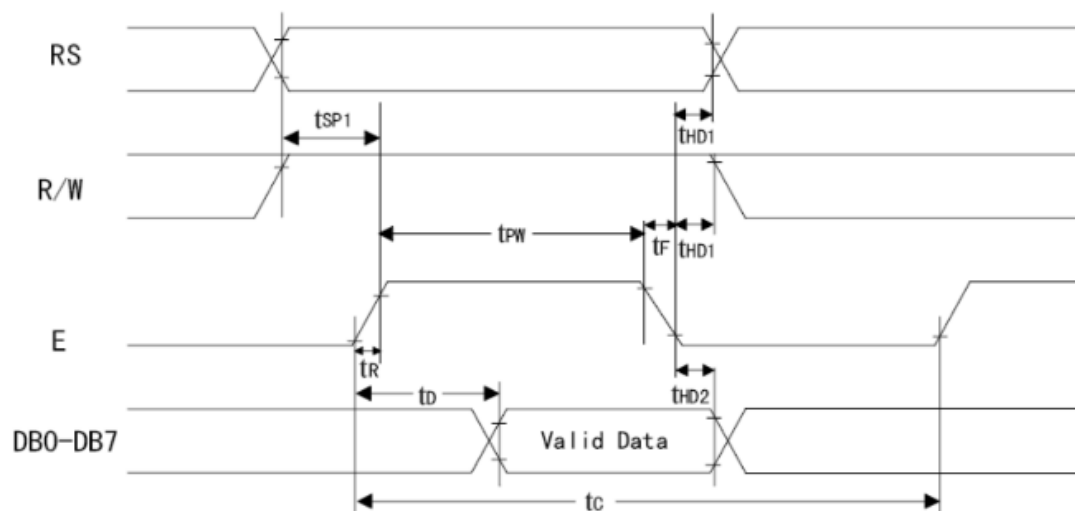


图 5: LCD1602 读时序

- 3) 写命令：输入 RS=0, RW=0, E=高脉冲。输出：无。
- 4) 写数据：输入 RS=1, RW=0, E=高脉冲。输出：无。

LCD1602 的写时序如下图所示。

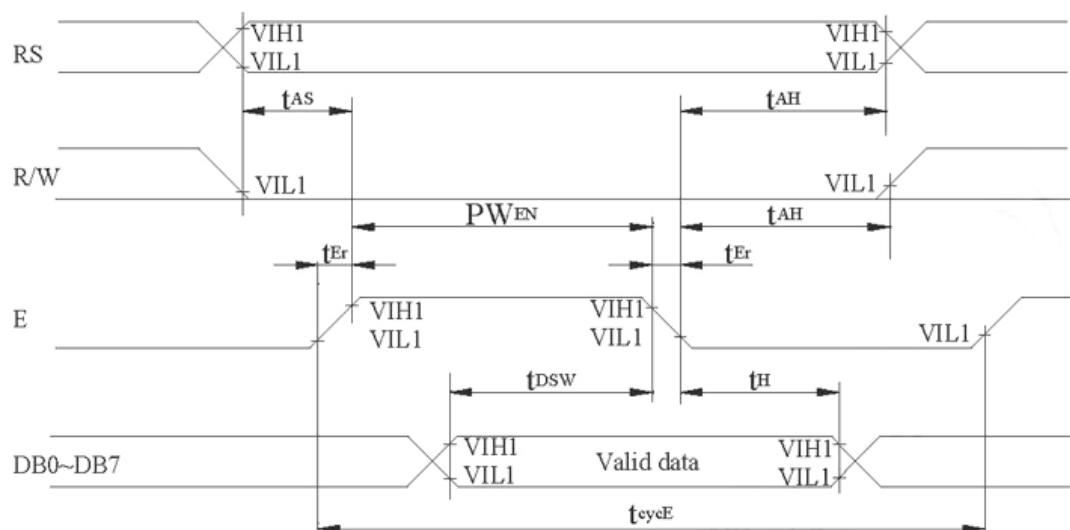


图 6: LCD1602 写时序

2.4. LCD1602 指令集

LCD1602 共有如下 11 条指令。

1. 清屏指令

■ 指令码：

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

- 功能：清显示，DDRAM 的全部空间填入空字符“20H”；地址计数器 AC=0；光标归位到 01H 位置（显示屏左上角）。

2. 光标归位指令

■ 指令码：

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

- 功能：地址计数器 AC=0；光标归位到 01H 位置（显示屏左上角）；DDRAM 内容保持不变。

3. 进入模式指令

■ 指令码：

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

- 功能：设置光标、画面移动方式。
- I/D=1：完成一个字符码传送后，光标右移，AC 自动加 1。
 - I/D=0：完成一个字符码传送后，光标左移，AC 自动减 1。
 - S=1：将全部显示向右（I/D=1）或向左（I/D=0）移位。
 - S=0：显示不移动。

4. 显示开/关控制指令

■ 指令码：

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

- 功能：控制显示器开/关、光标显示/关闭以及光标是否闪烁。
- D=1：开显示；D=0：关显示。关显示后，显示数据仍保存在 DDRAM 中。
 - C=1：显示光标；C=0：不显示光标。
 - B=1：光标闪烁；B=0：光标不闪烁。

5. 光标或显示移位指令

■ 指令码：

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

- 功能：设置整屏移位或光标移位。

S/C	R/L	说 明
0	0	光标向左移动, AC 自动减 1
0	1	光标向右移动, AC 自动加 1
1	0	光标和显示一起向左移动, AC 值不变
1	1	光标和显示一起向右移动, AC 值不变

6. 功能设置指令

■ 指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

■ 功能: 设置模块数据接口宽度和屏的显示方式。

- DL=1: 8 位数据总线 (D0~D7); DL=0: 8 位数据总线 (D4~D7, D0~D3 不使用)。
- N=1: 两行显示; N=0: 单行显示。
- F=1: 5×10 点阵加光标显示模式; F=0: 5×7 点阵加光标显示模式。

7. 设置 CGRAM 地址指令

■ 指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	ACG5	ACG4	ACG3	ACG2	ACG1	ACG0

■ 功能: 设定下一个要存入数据的 CGRAM 的地址。

8. 设置 DDRAM 地址指令

■ 指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

■ 功能: 设置 DDRAM 地址指针, 设置地址后即可通过写指令将欲显示的字符码写入该地址。

9. 读忙标志 BF 和 AC

■ 指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

■ 功能: 判忙和获取地址指针内容。

- BF=1: 模块正在进行内部操作, 无法接收任何外部指令和数据, 直到 BF=0 为止。
- AC6 ~ AC0: 地址计数器 AC 内的当前内容, 由于地址计数器 AC 为 CGROM、CGRAM 和 DDRAM 公用, 因此当前 AC 内容所指区域由前一条指令操作区域决定; 故只有 BF=0 时, 送到 DB7~DB0 的数据 AC6~AC0 才有效。

10. 向 DDRAM 或 CGRAM 写入数据

■ 指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

- 功能：将用户自定义字符的字模数据写到已经设置好的 CGRAM 的地址中，或者是将欲显示字符的字符码写到 DDRAM 中。

11. 从 CGRAM 或 DDRAM 中读数据

- 指令码：

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

- 功能：从地址计数器 AC 指定的 CGRAM 或者 DDRAM 单元中，读出数据 D7 ~ D0，在读数据之前，应先通过地址计数器 AC 正确指定读取单元的地址。

3. LCD1602 硬件电路设计

IK-64D4 开发板上设计了 LCD1602 显示模块接口，电路如下图所示。这里需要注意的是第 3 引脚（VO）为显示屏对比度调整端口，电路中通过一个 10K 的电位器调整对比度，VO 连接到电位器的抽头，通过旋转电位器旋钮即可改变电位器抽头电压，从而调整液晶屏对比度。

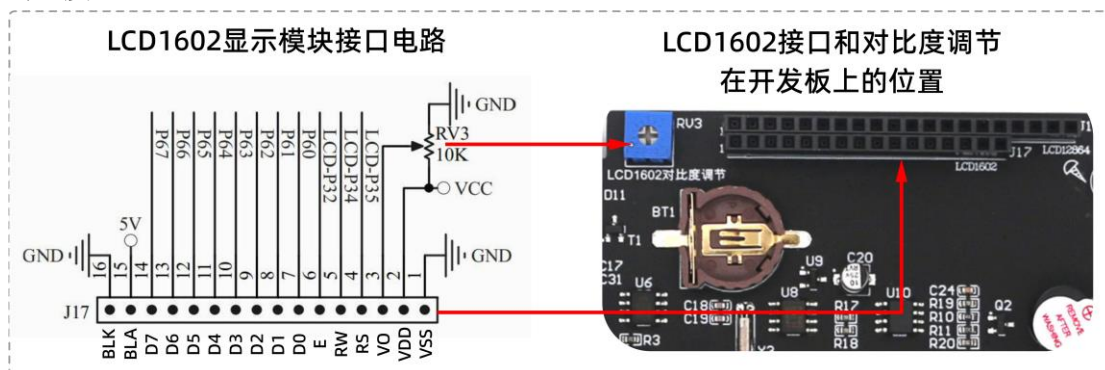


图 7：LCD1602 显示模块接口电路

LCD1602 显示模块占用 STC8A8K64D4 的引脚如下表：

表 2：LCD1602 引脚分配

LCD1602 引脚	信号名称	单片机	说明
1	VSS	连接到 5V 电源正	——
2	VDD	连接到 5V 电源负	——
3	VO	连接到电位器抽头	——
4	RS	P3.5	和 LCD12864 显示模块共用
5	RW	P3.4	和 LCD12864 显示模块共用
6	E	P3.2	和 LCD12864 显示模块共用
7	D0	P6.0	和 LCD12864 显示模块共用
8	D1	P6.1	和 LCD12864 显示模块共用

9	D2	P6.2	和 LCD12864 显示模块共用
10	D3	P6.3	和 LCD12864 显示模块共用
11	D4	P6.4	和 LCD12864 显示模块共用
12	D5	P6.5	和 LCD12864 显示模块共用
13	D6	P6.6	和 LCD12864 显示模块共用
14	D7	P6.7	和 LCD12864 显示模块共用
15	BLA	连接到 5V 电源正	——
16	BLK	连接到 5V 电源负	——

4. 软件设计

4.1. LCD1602 显示字符实验

✧ 注：本节的实验是在“实验 2-6-1：串口 1 数据收发实验”的基础上修改，本节对应的实验源码是：“实验 3-3-1：LCD1602 显示字符”。

4.1.1. 实验内容

LCD1602 模式：8 位数据总线（D0~D7）；两行显示；5×7 点阵显示模式。

上电启动后，LCD1602 第一行显示“LCD1602 check ok”，第二行显示“0123456789A BCDEF”。之后，每秒修改一次第 2 行第 10 个字符显示区域的显示内容：循环显示数字 0 到 9。

4.1.2. 代码编写

1. 新建一个名称为“lcd1602.c”的文件及其头文件“lcd1602.h”并保存到工程的“Source”文件夹，并将“lcd1602.c”加入到 Keil 工程中的“SOURCE”组。
2. 引用头文件

因为在“main.c”文件中使用了“lcd1602.c”文件中的函数，所以需要引用下面的头文件“lcd1602.h”。

代码清单：引用头文件

```
1. //引用 lcd1602 的头文件
2. #include "lcd1602.h"
```

3. 编写写指令和写数据的函数

根据前文描述的 LCD1602 显示模块的读写时序，编写的写命令函数和写数据函数代码清单如下。

代码清单：写命令

```
1. /*****
2. * 描 述：LCD1602 写命令函数
3. * 参 数：dat[in]：8 位的命令数据
4. * 返回值：无
5. *****/
```

```
6. void LCD1602_WriteCmd(u8 com)
7. {
8.     RS=0;           //选择指令寄存器
9.     RW=0;           //选择写操作
10.    P6=com;          //把命令送入到 LCD1602 屏 8 位数据口
11.    delay_ms(1);     //延时, LCD1602 准备接收数据
12.    //控制 LCD1602 使能引脚产生高电平到低电平的跳变, 液晶模块执行命令
13.    EN=1;            //使能引脚置高电平
14.    EN=0;            //使能引脚置低电平
15. }
```

代码清单：写数据

```
1. /*****
2.  * 描 述 : LCD1602 写数据函数
3.  * 参 数 : dat[in]: 8 位的数据
4.  * 返回值 : 无
5. *****/
6. void LCD1602_WriteData(u8 dat)
7. {
8.     RS=1;           //选择数据寄存器
9.     RW=0;           //选择写操作
10.    P6=dat;          //把命令送入到 LCD1602 屏 8 位数据口
11.    delay_ms(1);     //延时, LCD1602 准备接收数据
12.    //控制 LCD1602 使能引脚产生高电平到低电平的跳变, 液晶模块执行命令
13.    EN=1;            //使能引脚置高电平
14.    EN=0;            //使能引脚置低电平
15. }
```

4. LCD1602 显示模块初始化

LCD1602显示模块使用前先要初始化, 本例中配置LCD1602为8位数据总线(D0~D7)、两行显示、5×7点阵显示。配置后发送清屏指令清除LCD1602显示。

代码清单：LCD1602 显示模块初始化

```
1. /*****
2.  * 描 述 : LCD1602 初始化
3.  * 参 数 : 无
4.  * 返回值 : 无
5. *****/
6. void lcd1602_init(void)
7. {
8.     LCD1602_WriteCmd(0x38); //显示模式设置: 16*2 显示, 5*7 点阵, 8 位数据接口
9.     LCD1602_WriteCmd(0x0c); //开显示, 不显示光标, 光标不闪烁
10.    LCD1602_WriteCmd(0x06); //显示地址递增

```

```
11.     LCD1602_WriteCmd(0x01);           //清屏
12. }
```

4. LCD1602 在指定位置显示一个字符

LCD1602 在指定位置显示字符时先写入 DDRAM 地址，然后再向该地址写入数据，即：先用“设置 DDRAM 地址指令”设置地址，然后用“向 DDRAM 或 CGRAM 写入数据指令”将待显示的字符编写写入该地址。

代码清单：在指定位置显示一个字符

```
1.  /*****
2.  * 描 述：在指定的字符显示区域显示一个字符
3.  * 参 数：x[in]：行地址，范围是 0~15，对应第 1~第 16 个字符显示起始位置
4.  *       : y[in]：列地址，范围是 0~1，对应第 1、第 2 列
5.  *       : dat[in]：显示的数据
6.  * 返回值：无
7.  *****/
8. void LCD1602_DisOneChar(u8 x, u8 y, u8 dat)
9. {
10.     x &= 0x01;           //限制 X 不能大于 1，Y 不能大于 15
11.     y &= 0x0F;
12.     if (x) y |= 0x40;     //当要显示第二行时地址码+0x40;
13.     y |= 0x80;           //写入地址时第 7 位必须为 1
14.     LCD1602_WriteCmd(y); //发命令字
15.     LCD1602_WriteData(dat); //发数据
16. }
```

4. LCD1602 在显示字符串

LCD1602 显示字符串时，和显示单个字符一样，先写入 DDRAM 地址，然后再向该地址依次写入字符串中的字符。需要注意的是：字符串的长度不能超过 LCD1602 的字符显示区域能容纳的长度。

代码清单：在指定位置显示一个字符

```
1.  /*****
2.  * 描 述：LCD1602 从指定位置开始显示字符串，注意，字符串大小不能超过 LCD1602 显示区域
3.  * 参 数：x[in]：行地址，范围是 0~15，对应第 1~第 16 个字符显示起始位置
4.  *       : y[in]：列地址，范围是 0~1，对应第 1、第 2 列
5.  *       : str[in]：显示的字符串，字符串以/0 结尾
6.  * 返回值：无
7.  *****/
8. void LCD1602_DisString(u8 x, u8 y, u8 code *str)
9. {
10.     x &= 0x01; //限制 X 不能大于 1，Y 不能大于 15
11.     y &= 0x0F;
12. }
```

```

13.   if (x) y |= 0x40;    //当要显示第二行时地址码+0x40;
14.   y |= 0x80;          //写入地址时第 7 位必须为 1
15.   LCD1602_WriteCmd(y); //DDRAM 地址
16.   while(*str != '\0')
17.   {
18.       LCD1602_WriteData(*str);
19.       str++;
20.   }
21. }

```

5. 主函数

定义两个字符串，每个字符串包含 16 个字符，

分别存放将在 LCD1602 的两行上显示。注意，这里字符串数组用了“code”关键字修饰，因此，他们会被存放到代码区（Flash），他们是不能修改的，如果需要在程序中修改，删除“code”关键字即可。

代码清单：定义字符串数组

```

1.  u8 code table1[]="LCD1602 check ok";    //要显示的内容 1 放入数组 table1
2.  u8 code table2[]="0123456789ABCDEF";    //要显示的内容 2 放入数组 table2

```

主函数中，在进入主循环之前，在 LCD1602 第一行显示字符串数组“table1”中的内容“LCD1602 check ok”，第二行显示字符串数组“table2”中的内容“0123456789ABCDEF”。进入主循环后，每秒修改一次第 2 行第 10 个字符显示区域的显示内容：循环显示数字 0 到 9。

代码清单：主函数

```

1.  /*****
2.  功能描述: 主函数
3.  参    数: 无
4.  返 回 值: int 类型
5.  *****/
6.  int main(void)
7.  {
8.      u8 i = 0;
9.      P2M1 &= 0xBF;   P2M0 &= 0xBF;    //设置 P2.6 为准双向口（指示灯 D1）
10.     P3M1 &= 0xCB;   P3M0 |= 0x34;    //设置 P3.2,P3.4,P3.5 为强推挽输出（LCD1602 控制）
11.     P6M1 &= 0x00;   P6M0 &= 0x00;    //设置 P6.0~P6.7 为准双向口（LCD1602 数据）
12.     lcd1602_init();    //LCD1602 显示模块初始化
13.     LCD1602_DispString(0, 0, table1); //第 1 行显示 table1 数组中的内容
14.     LCD1602_DispString(1, 0, table2); //第 2 行显示 table1 数组中的内容
15.     while(1)
16.     {
17.         LCD1602_DispOneChar(1, 9, i+48); //加 48 是为了将数字转成对应的 ASCII 码
18.         i++;

```

4.1.3. 硬件连接

图 8: 硬件连接

4.1.4. 实验步骤

- #### 4.2. LCD1602 滚动显示实验

13 / 17

4.2.1. 实验内容

让“实验 3-3-1: LCD1602 显示字符”中显示的内容以向左滚动的方式显示。

4.2.2. 代码编写

LCD1602 滚动显示很简单，只需要发送“光标或显示移位指令”这一条指令即可实现滚动，该指令每发送一次，移位一个字符显示区域。

这里需要注意一下：DDRAM 实际大小是每行 40 个字节，其中前 16 个字节是和 LCD1602 的显示对应的，而滚动显示是按照 DDRAM 内容首尾衔接滚动的，所以在实验时会看到，滚动显示时中间有一部分是空的。如果要消除这部分空白，将 DDRAM 中第 17 到第 40 个地址都写入数据即可。

代码清单：LCD1602 滚动显示

```
1.  /*****  
2.  功能描述: 主函数  
3.  参    数: 无  
4.  返 回 值: int 类型  
5.  *****/  
6.  int main(void)  
7.  {  
8.      //省略了无关的内容  
9.  
10.     while(1)  
11.     {  
12.         LCD1602_WriteCmd(0x18);    //光标和显示一起向左移动，该指令每发送一次，移位一个字符显示区域  
13.         delay_ms(600);              //延时 600ms  
14.         led_toggle(LED_1);          //指示灯 D1 状态翻转，指示程序的运行  
15.     }  
16. }
```

4.2.3. 硬件连接

同实验 3-3-1.

4.2.4. 实验步骤

- 1) 解压“…\第 3 部分: 配套例程源码”目录下的压缩文件“实验 3-3-2: LCD1602 滚动显示”，将解压后得到的文件夹拷贝到合适的目录，如“D:\STC8”（这样做的目的是为了防止中文路径或者工程存放的路径过深导致打开工程出现问题）。
- 2) 双击“…\lcd1602_scroll\project”目录下的工程文件“lcd1602_scroll.uvproj”。
- 3) 点击编译按钮编译工程，编译成功后生成的 HEX 文件“lcd1602_scroll.hex”位于工程的“…\lcd1602_scroll\Project\Object”目录下。
- 4) 打开 STC-ISP 软件下载程序，下载使用内部 IRC 时钟，IRC 频率选择：24MHz。
- 5) 电脑上打开串口调试助手，选择开发板对应的串口号，将波特率设置为 9600bps。
- 6) 程序运行后，可以观察到 LCD1602 以 600ms 的间隔向左滚动显示。

4.3. LCD1602 显示 ADC 采样电压实验

✧ 注：本节的实验是在“实验 3-3-1：LCD1602 显示字符”的基础上修改，本节对应的实验源码是：“实验 3-3-3：LCD1602 显示 ADC 采样电压值”。

4.3.1. 实验内容

使用 ADC 模拟通道输入 14（即引脚 P0.6）采样电位器抽头电压，程序中每 1 秒执行一次电压采样，采样结果计算为电压值后在 LCD1602 显示模块上显示。

✧ 关于 ADC 应用相关的内容，读者可以参阅《第 2-11 讲：模数转换 ADC》，本节我们关注的是 LCD1602 如何实时显示 ADC 采样的电压值。

4.3.2. 代码编写

首先，定义一个大小为 5 个字节的用于存放显示内容的数组，由于数组中部分内容在显示电压的时候是无需改变的，因此，我们在定义的时候对数组进行初始化，这样，更新显示的时候，只需和电压相关的值就可以了。

代码清单：定义存放显示内容的数组

```
1. //该字符串在程序运行过程中无需改变，所以用 code 关键字修饰
2. u8 code table1[]="Voltage:";
3. //因为 ADC 采样的电压不会超过 2.5V，所以电压的整数部分 1 位，小数部分 2 位即可，再加上小数点和字母'V'，一共 5 个字节
4. u8 Disp_Buff[5] = {0, '.', 0, 0, 'V'};
```

主函数中，先执行相关的初始化，在进入主循环之前，在 LCD1602 先显示出程序运行中无需更改的部分。进入主循环后，每秒采样一次电位器抽头的电压，并将电压在 LCD1602 上显示出来，显示的时候保留 2 位有效数字。

代码清单：主函数

```
1. /*****
2. 功能描述：主函数
3. 参 数：无
4. 返 回 值：int 类型
5. *****/
6. int main(void)
7. {
8.     float adc_value;           //存放 ADC 计算后的电压值
9.     int adc_int,tmp;
10.
11.     P2M1 &= 0xBF;   P2M0 &= 0xBF;   //设置 P2.6 为准双向口（指示灯 D1）
12.     P3M1 &= 0xFE;   P3M0 &= 0xFE;   //设置 P3.0 为准双向口（串口 1 的 RxD）
13.     P3M1 &= 0xFD;   P3M0 |= 0x02;   //设置 P3.1 为推挽输出（串口 1 的 TxD）
14.     P3M1 &= 0xCB;   P3M0 |= 0x34;   //设置 P3.2,P3.4,P3.5 为强推挽输出（LCD1602 控制）
15.     P6M1 &= 0x00;   P6M0 &= 0x00;   //设置 P6.0~P6.7 为准双向口（LCD1602 数据）
16.     uart1_init();           //串口 1 初始化
```

15 / 17

```

17.  lcd1602_init();           //LCD1602 显示模块初始化
18.  adc_config();             //初始化 ADC
19.  EA = 1;                   //使能总中断
20.
21.  LCD1602_DisString(0, 0, table1); //LCD1602 显示无需改变的内容
22.  LCD1602_DisOneChar(0,10,Disp_Buff[1]); //显示小数点
23.  LCD1602_DisOneChar(0,13,Disp_Buff[4]); //显示字母 'V'
24.  while(1)
25.  {
26.      adc_start();           //启动 ADC 转换
27.      while(adc_completed() == false); //等待 ADC 转换完成
28.      adc_value = (2.5*get_adc_value())/4096; //读取 ADC 才采样值并转换为电压（单位 V）
29.      printf("voltage: %.2fV\r\n",adc_value); //串口打印 ADC 采样电压
30.      adc_int = (int)adc_value; //取整数部分
31.      Disp_Buff[0] = adc_int % 10 + 48; //整数转换为 ASCII 码
32.      tmp = (int)((adc_value-adc_int)*100+0.5); //保留 2 位小数，所以这里乘以 100,加 0.5 是四舍五入
33.      Disp_Buff[2] = tmp / 10 + 48; //第 1 位小数转换为 ASCII 码
34.      Disp_Buff[3] = tmp % 10 + 48; //第 2 位小数转换为 ASCII 码
35.      //LCD 显示计算后的电压值
36.      LCD1602_DisOneChar(0,9,Disp_Buff[0]);
37.      LCD1602_DisOneChar(0,11,Disp_Buff[2]);
38.      LCD1602_DisOneChar(0,12,Disp_Buff[3]);
39.
40.      delay_ms(1000); //延时，方便观察数据
41.      led_toggle(LED_1); //指示灯 D1 状态翻转，指示程序的运行
42.  }
43. }

```

4.3.3. 硬件连接

1. 安装 LCD1602 显示模块。
2. 按照下图短接 ADC 的跳线帽。



图 9：ADC 跳线帽短接

4.3.4. 实验步骤

- 1) 解压“…\第 3 部分：配套例程源码”目录下的压缩文件“实验 3-3-3: LCD1602 显示 ADC 采样电压值”，将解压后得到的文件夹拷贝到合适的目录，如“D\STC8”（这样做的目的是为了防止中文路径或者工程存放的路径过深导致打开工程出现问题）。
- 2) 双击“…\lcd1602_adc\project”目录下的工程文件“lcd1602_adc.uvproj”。
- 3) 点击编译按钮编译工程，编译成功后生成的 HEX 文件“lcd1602_adc.hex”位于工程的“…\lcd1602_adc\Project\Object”目录下。
- 4) 打开 STC-ISP 软件下载程序，下载使用内部 IRC 时钟，IRC 频率选择：24MHz。
- 5) 电脑上打开串口调试助手，选择开发板对应的串口号，将波特率设置为 9600bps。
- 6) 程序运行后，可以观察到 LCD1602 实时显示 ADC 测量的电位器抽头的电压，旋转电位器改变电位器抽头电压，可以观察到 LCD1602 上显示的电压随之改变。