第13章 STC单片机ADC原理及实现

何宾 2018.03

直流电压测量和1602 LCD显示 --实现目标

本设计将读取STC学习板上按下不同按键所得到的直流电压值,经过ADC转换器转换后,得到数字量的值,经过计算后,通过1602字符LCD屏显示得到的直流电压值。

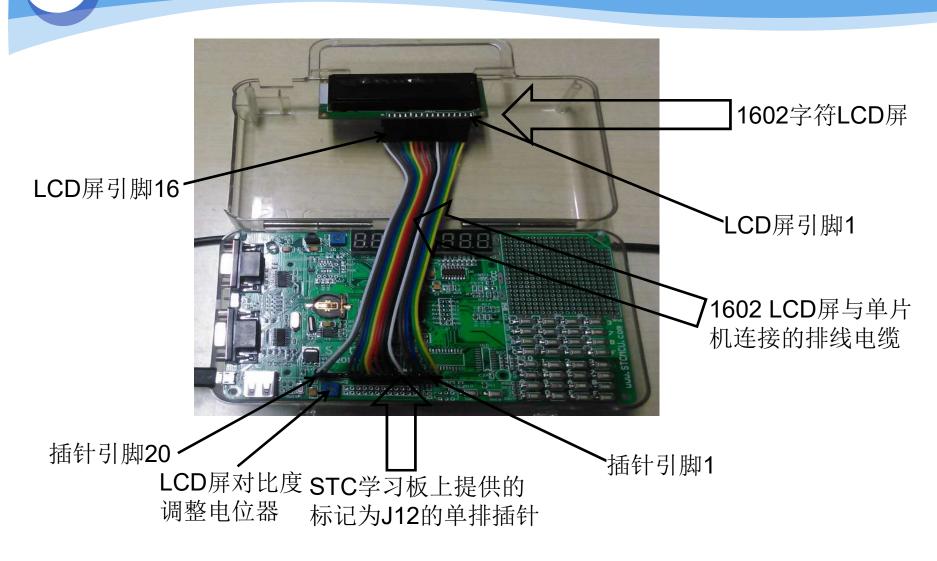
直流电压测量和1602 LCD显示 --硬件电路设计

在该设计中,1602字符屏通过排线电缆与STC学习板上的标记为J12的单排插座连接。

- 在图中标出了STC学习板上插针引脚1的位置和1602字符屏引脚1的位置。
- STC学习板上J12提供20个插针,可以直接与12864图形/字符 LCD进行连接,对于1602字符屏来说,不能直接进行连接。

注:它们的信号引脚定义如后表所示。

直流电压测量和1602 LCD显示 --硬件电路设计



直流电压测量和1602 LCD显示)

--硬件电路设计

STC学习板 J12插座引脚号	信号名字	与单片机引脚连接关系	1602LCD引脚号	信号名字	功能
1	GND	地	1	VSS	地
2	VCC	+5V电源	2	VCC	+5V电源
3	V0		3	V0	LCD驱动电压输入
4	RS	P2.5	4	RS	寄存器选择。RS=1,数据; RS=0,指令
5	R/W	P2.6	5	R/W	读写信号。R/W=1,读操作; R/W=0,写操作
6	Е	P2.7	6	Е	芯片使能信号
7	DB0	P0.0	7	DB0	8位数据总线信号
8	DB1	P0.1	8	DB1	
9	DB2	P0.2	9	DB2	
10	DB3	P0.3	10	DB3	
11	DB4	P0.4	11	DB4	
12	DB5	P0.5	12	DB5	
13	DB6	P0.6	13	DB6	
14	DB7	P0.7	14	DB7	
15	PSB	P2.4	15	LEDA	背光源正极,接+5.0V
16	N.C	P2.2	16	LEDK	背光源负极,接地
17	/RST	P2.3			
18	VOUT				
19	Α	背光源正极,接+5.0V			

直流电压测量和1602 LCD显示 --1602字符LCD原理

1602字符LCD指标

1602字符LCD主要技术参数

显示容量	16×2个字符,即:可以显示2行字符,每行可以显示16个字符
工作电压范围	4.5V~5.5V。推荐5.0V
工作电流	2.0mA@5V
屏幕尺寸	2.95×4.35mm (宽×高)

直流电压测量和1602 LCD显示 --1602字符LCD原理

1602字符LCD内部显存

- 1602液晶内部包含80个字节的显示RAM,存储需要发送的数据
- 第一行存储器地址范围0x00~0x27; 第二行存储器地址范围为 0x40~0x67。
 - 口 第一行存储器地址范围0x00~0x0F与1602字符LCD第一行位置对应。
 - 口 第二行存储器地址范围0x40~0x4F与1602字符LCD第二行位置对应。

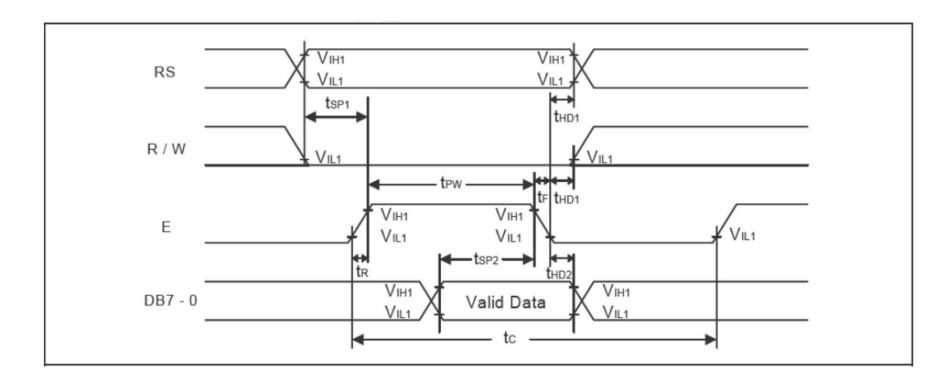
注:每行多出来的部分是为了显示移动字幕设置。

16×2字符LCD

(00	01	02	03	04	05	06	07	08	09	OA	ОВ	0C	OD	OE	0F	10	11	•••	27
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	•••	67

直流电压测量和1602 LCD显示--1602字符LCD读写时序

写操作时序

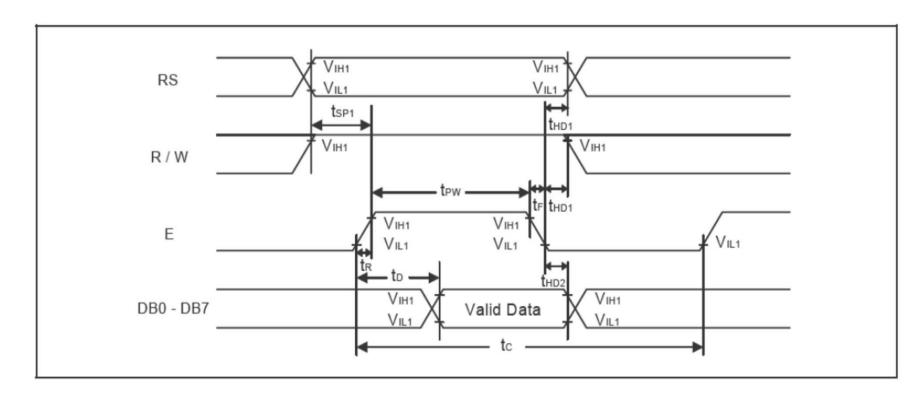


直流电压测量和1602 LCD显示 --1602字符LCD读写时序

- 首先,将R/W信号拉低。同时,给出RS信号,该信号为1或者0,用于区分数据和命令。
- 然后,将E信号拉高。当E信号拉高后,STC单片机将写入1602字符LCD的数据放在DB7~DB0数据线上。当数据有效一段时间后,首先将E信号拉低。然后,数据继续维持一段时间Tно2。这样,数据就写到1602字符LCD中。
- 最后,撤除/保持R/W信号。

直流电压测量和1602 LCD显示 --1602字符LCD读写时序

读操作时序



直流电压测量和1602 LCD显示 --1602字符LCD读写时序

- 首先,将R/W信号拉高。同时,给出RS信号,该信号为1或者0,用于区分数据和状态。
- 然后,将E信号拉高。当E信号拉高,并且延迟一段时间t。后, 1602字符LCD将数据放在DB7~DB0数据线上。当维持一段时 间tpw后,将E信号拉低。
- 最后,撤除/保持R/W信号。

直流电压测量和1602 LCD显示--1602字符LCD读写时序

将上面的读和写操作总结

读和写操作总结

RS	R/W	操作说明
0	0	写入指令寄存器 (清屏)
0	1	读BF(忙)标志,以及读取地址计数器的内容
1	0	写入数据寄存器(显示各字型等)
1	1	从数据寄存器读取数据

直流电压测量和1602 LCD显示 --1602字符LCD命令和数据

在STC单片机对1602字符LCD操作的过程中,会用到下

面的命令

1602字符LCD命令和数据

指令					指令	操作码	功能				
1日 文	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	切 肥
清屏	0	0	0	0	0	0	0	0	0	1	将"20H"写到DDRAM,将DDRAM地址从AC(地址计数器)设置到"00"
光标归 位	0	0	0	0	0	0	0	0	1	-	将DDRAM的地址设置为"00",光 标如果移动,则将光标返回到初 始的位置。DDRRAM的内容保持不 变
输入模 式设置	0	0	0	0	0	0	0	1	I/D	S	分配光标移动的方向,使能整个显示的移动。 I=0,递减模式。I=1,递增模; S=0,关闭整个移动。S=1,打开整个移动;

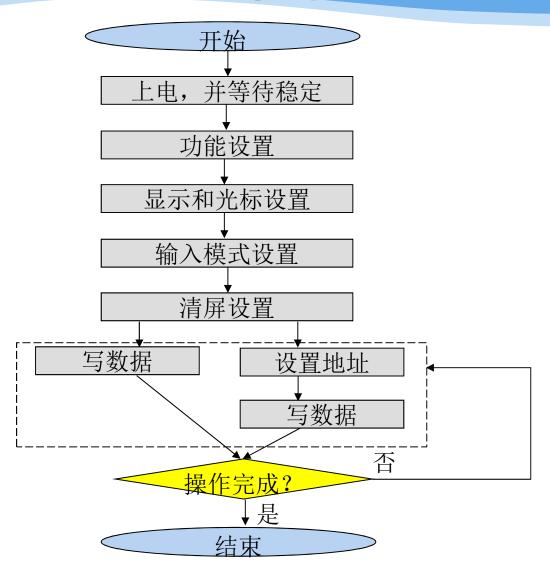
直流电压测量和1602 LCD显示 --1602字符LCD命令和数据

显示打开	0	0	0	0	0	0	1	D	С	В	设置显示(D), 光标(C)和光标闪烁(B)打开/关闭控制。 D=0,显示关闭; D=1,打开显示 C=0,关闭光标; C=1,打开光标 B=0,关闭闪烁; B=1,打开闪烁
光标或 者显示 移动	0	0	0	0	0	1	S/C	R/L	_	_	设置光标移动和显示移动的控制位,以及方向,不改变DDRAM数据S/C=0, R/L=0, 光标左移;S/C=0, R/L=1, 光标右移;S/C=1, R/L=0, 显示左移, 光标跟随显示移动S/C=1, R/L=1, 显示右移, 光标跟随显示移动

直流电压测量和1602 LCD显示 --1602字符LCD命令和数据

功能设置	0	0	0	0	1	DL	N	F	_	_	设置接口数据宽度,以及显示行的个数。 DL=1,8位宽度; DL=0,4位宽度 N=0,1行模式; N=1,2行模式 F=0,5×8字符字体; F=1,5×10字符字体
设置CGRAM 地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	在地址计数器中,设置CGRAM地址
设置DDRAM 地址	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	在计数器中,设置DDRAM地址
读忙标志 和地址计 数器	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	读BF标志,知道LCD屏内部是否正在操作。也可以读取地址计数器的内容
将数据写 到RAM	1	0	D7	D6	D5	D4	D3	D2	D1	DO	写数据到内部RAM (DDRAM/ CGRAM)
从RAM读数 据	1	1	D7	D6	D5	D4	D3	D2	D1	DO	从内部RAM(DDRAM/ CGRAM)读取数据

直流电压测量和1602 LCD显示 --1602字符LCD初始化和操作流程



直流电压测量和1602 LCD显示

--系统软件处理流程

ADC中断服务程序入口

软件清除ADC_CONTR寄存器 ADC_FLAG标志

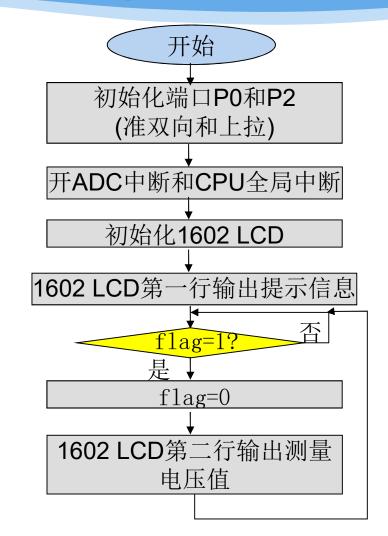
读取ADC_RES和ADC_RES1寄存器的内容,计算出数字量

(数字量×Vcc)/1024,得到对应的 浮点模拟电压值

flag=1

再次启动ADC

ADC中断服务程序结束



【例】采集分压网络的电压值在1602字符LCD上显示的C语言描述的例子。

led1602.h文件

```
//条件编译命令,如果没有定义_1602_
#ifndef 1602
                         //定义 1602
#define 1602
                         //包含reg51.h头文件
#include "reg51.h"
                         //包含intrins.h头文件
#include "intrins.h"
                         //定义LCD1602 RS为P2.5引脚
sbit LCD1602_RS=P2^5;
                         //定义LCD1602 RW为P2.6引脚
sbit LCD1602 RW=P2^6;
                         //定义LCD1602 E为P2.7引脚
sbit LCD1602 E =P2^7;
                         //定义LCD1602 DB为P0端口
sfr LCD1602 DB=0x80;
                         //定义P0端口P0M1寄存器地址0x93
sfr P0M1=0x93;
```

```
//定义P0端口P0M0寄存器地址0x94
sfr P0M0=0x94;
                                  //定义P2端口P2M1寄存器地址0x95
sfr P2M1=0x95;
                                  //定义P2端口P2M0寄存器地址0x96
sfr P2M0=0x96;
                                  //定义子函数lcdwait类型
void lcdwait();
                                  //定义子函数lcdwritecmd类型
void lcdwritecmd(unsigned char cmd);
                                  //定义子函数lcdwritedata类型
void lcdwritedata(unsigned char dat);
                                  //定义子函数lcdinit类型
void lcdinit();
void lcdsetcursor(unsigned char x, unsigned char y);
                                  //定义子函数lcdsetcursor类型
void lcdshowstr(unsigned char x, unsigned char y,unsigned char *str);
                                  //定义子函数lcdshowstr类型
                                  //条件预编译命令结束
#endif
```

直流电压测量和1602 LCD显示。

--具体实现过程

led1602.c文件

```
#include "led1602.h"
                              //包含led1602.h头文件
                              //声明lcdwait函数,用于读取BF标志
void lcdwait()
                              //读取前,先将P0端口设置为"1"
      LCD1602 DB=0xFF;
                              //空操作延迟
      _nop_();
                              //空操作延迟
      _nop_();
                              //空操作延迟
      _nop_();
                              //空操作延迟
      _nop_();
                              //将LCD1602的RS信号拉低
      LCD1602 RS=0;
                              //将LCD1602的RW信号拉高
      LCD1602_RW=1;
                              //将LCD1602的E信号拉高
      LCD1602 E=1;
                              //等待标志BF为低表示LCD1602空闲
      while(LCD1602 DB & 0x80);
                              //将LCD1602的E信号拉低
      LCD1602 E=0;
```

void lcdwritecmd(unsigned char cmd)

```
//声明lcdwritecmd函数,写命令到1602
                 //调用lcdwait函数
lcdwait();
                 //空操作延迟
_nop_();
                 //空操作延迟
_nop_();
                 //空操作延迟
_nop_();
                 //空操作延迟
_nop_();
LCD1602_RS=0;
                 //将LCD1602的RS信号拉低
                 //将LCD1602的RW信号拉低
LCD1602 RW=0;
                 //将命令控制码cmd放到P0端口
LCD1602 DB=cmd;
                 //将LCD1602的E信号拉高
LCD1602 E=1;
```

直流电压测量和1602 LCD显示)

--具体实现过程

void lcdwritedata(unsigned char dat) //声明lcdwritedata函数,写数据到1602

```
//调用lcdwait函数
lcdwait();
                 //空操作延迟
_nop_();
                 //空操作延迟
_nop_();
                 //空操作延迟
_nop_();
                 //空操作延迟
_nop_();
LCD1602_RS=1; //将LCD1602的RS信号拉高
LCD1602 RW=0; //将LCD1602的RW信号拉低
LCD1602 DB=dat; //将数据码cmd放到P0端口
                 //将LCD1602的E信号拉高
LCD1602 E=1;
                 //空操作延迟
_nop_();
                 //空操作延迟
_nop_();
                 //空操作延迟
_nop_();
                 //空操作延迟
_nop_();
LCD1602 E=0;
                 //将LCD1602的E信号拉低
```

```
      void lcdinit()
      //声明lcdinit子函数,用来初始化1602

      {
      lcdwritecmd(0x38);
      //发命令0x38,2行模式,5*8点阵,8位宽度

      lcdwritecmd(0x0c);
      //发命令0x0C,打开显示,关闭光标

      lcdwritecmd(0x06);
      //发命令0x06,文字不移动,地址自动加1

      lcdwritecmd(0x01);
      //发命令0x01,清屏
```

```
//声明lcdsetcursor函数,设置显示RAM的地址,x和y表示在1602的列和行参数
void lcdsetcursor(unsigned char x, unsigned char y)
                       //声明无符号char类型变量address
  unsigned char address;
                       //如果第一行
  if(y==0)
                       //存储器地址以0x00开始
     address=0x00+x;
                       //如果是第二行
  else
                       //存储器地址以0x40开始
     address=0x40+x;
  lcdwritecmd(address|0x80); //写存储器地址命令
```

```
void lcdshowstr(unsigned char x, unsigned char y,unsigned char *str)
//在液晶指定的x和y位置,显示字符
                      //设置显示RAM的地址
  lcdsetcursor(x,y);
  while((*str)!= '\0' ) //如果不是字符串的结尾,则继续
     lcdwritedata(*str); //发写数据命令, 在LCD上显示数据
                     //指针加1,指向下一个地址
     str++;
```

main.c文件

#include "reg51.h"

#include "stdio.h"

#include "led1602.h"

#define ADC POWER 0x80

#define ADC FLAG 0x10

#define ADC_START 0x08

#define ADC_SPEEDLL 0x00

#define ADC SPEEDL 0x20

#define ADC_SPEEDH 0x40

#define ADC_SPEEDHH 0x60

//定义ADC_POWER的值0x80

//定义ADC FLGA的值0x10

//定义ADC START的值0x08

//定义ADC SPEEDLL的值0x00

//定义ADC_SPEEDL的值0x20

//定义ADC_SPEEDH的值0x40

//定义ADC_SPEEDHH的值0x60

```
sfr AUXR = 0x8E;
                         //声明AUXR寄存器的地址0x8E
                         //声明ADC CONTR寄存器的地址0xBC
sfr ADC CONTR = 0xBC;
                         //声明ADC RES寄存器的地址0xBD
sfr ADC RES = 0xBD;
                         //声明ADC RESL寄存器的地址0xBE
sfr ADC RESL =0xBE;
                         //声明P1ASF寄存器的地址0x9D
sfr P1ASF = 0x9D;
                         //声明char类型变量ch
unsigned char ch=4;
                         //声明bit类型变量flag
bit flag=1;
                         //声明float类型变量voltage
float voltage=0;
                         //声明char类型数组tstr
unsigned char tstr[5];
                         //声明int类型变量tmp
unsigned int tmp=0;
```

```
//声明adc中断服务程序
void adc int() interrupt 5
      unsigned char i=0;
                                   //声明char类型变量i
      ADC CONTR &=!ADC FLAG; //将ADC FLAG标志清零
      tmp=(ADC_RES*4+ADC_RESL); //读取模拟信号对应的数字量
                                  //将数字量转换成模拟电压值
      voltage=(tmp*5.0)/1024;
                                  //将浮点数,转换成对应的电压值
      sprintf(tstr, "%1.4f", voltage);
                                  //将flag置1
      flag=1;
      ADC CONTR=ADC POWER | ADC SPEEDLL | ADC START | ch;
                                  //启动ADC
```

直流电压测量和1602 LCD显示

--具体实现过程

void main()

```
//声明int型变量i
unsigned int i;
                //通过P0M0和P0M1寄存器, 将P0口
P0M0=0;
                //定义为准双向,弱上拉
P0M1=0;
                //通过P2M0和P2M1寄存器, 将P2口
P2M0=0;
                //定义为准双向,弱上拉
P2M1=0;
                //将P1端口用于ADC输入
P1ASF=0xFF;
                 //将ADC RES寄存器清零
ADC RES=0;
                 //配置ADC CONTR寄存器
ADC CONTR=ADC POWER|ADC SPEEDLL | ADC START | ch;
for(i=0;i<10000;i++); //延迟一段时间
                 //CPU允许相应中断请求,允许ADC中断
IE=0xA0;
                //等待1602字符LCD稳定
lcdwait();
```

```
lcdinit();
                          //初始化1602字符LCD
lcdshowstr(0,0,"Measured Voltage is");
                             //在1602第一行开始打印信息
                            //在1602第二行第6列打印字符 'v'
lcdshowstr(6,1, "V" );
                             //无限循环
while(1)
                             //判断flag标志是否为1,
        if(flag==1)
                            //将flag置0
          flag=0;
          lcdshowstr(0,1,tstr); //在第二行, 打印电压对应的字符
```

下载和分析设计的步骤主要包括:

- ■打开STC-ISP软件,在该界面内,选择硬件选项。将"输入用户程序运行时的IRC频率"设置为6.000MHz。
- 单击下载/编程按钮,按前面的方法下载设计到STC单片机。
- ■观察1602字符屏上的输出结果。

