第13章 STC单片机ADC原理及实现

何宾

2018.03

温度的测量和串口显示
--实现目标

在前面的设计中，在转换成模拟电压值的时候是基于单片机的供电电压VCC，典型的+5V。如果单片机的外部供电电压发生变化，则转换出来的电压就一定存在误差，而且误差随着Vcc的变化而不确定。

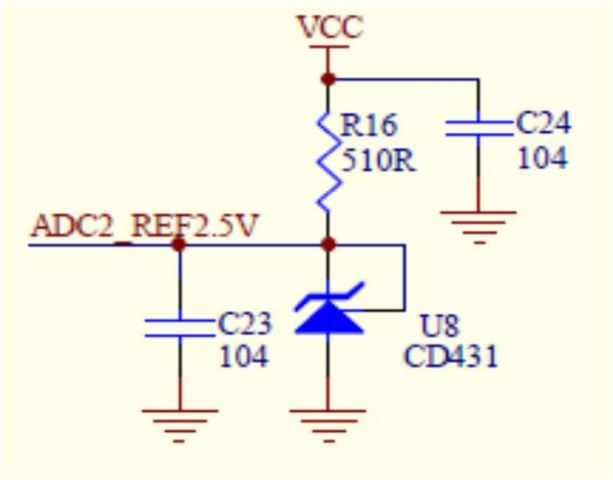
温度的测量和串口显示
--实现目标

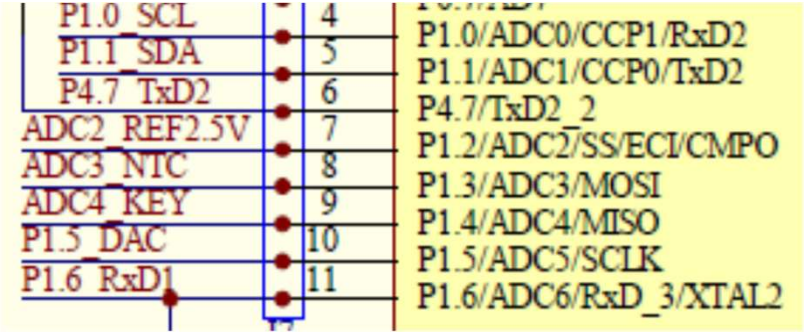
因此，如果能有一个稳定的电压源参考，然后基于此参考电源进行计算，所得到的被测量信号的模拟电压值误差只于STC15系列单片机ADC内部的转换误差，以及参考源的误差有关，和单片机的供电电压无关。这样，就很容易计算出输入模拟信号电压的相对误差和绝对误差。

- 在该设计中，通过基准电压源计算测量信号的结果，并通过串口1进行显示。

温度的测量和串口显示
--测量信号校准原理

在STC学习板上，提供了TL431基准参考电压源，该参考源默认输出+2.5V的参考信号。该信号连接到STC单片机的P1.2引脚上。





温度的测量和串口显示
--测量信号校准原理

TL431的技术指标主要包括：

- **在25℃时，误差为0.5%（B级）；误差为1%（A级）；误差为2%（标准级）；**
- **在0~70℃范围内，温漂为6mV；在 - 40~+85℃时，温漂为14mV。**

温度的测量和串口显示
--信号输入电路

在STC学习板上，提供了带有负温度系数NTC热敏电阻SDNT2012X103F3950FTF的信号输入电路。

- 该热敏电阻的负温度系数是指，即当温度升高的时候，热敏电阻值减少；
- 而当温度降低的时候，热敏电阻值增加。当在标称温度（25℃）时，热敏电阻的值为10KΩ。

温度的测量和串口显示
--信号输入电路

- 在温度T时的热敏电阻的值由下面的公式进行计算：

$$R_T = R_N \cdot \exp B(1/T - 1/T_N)$$

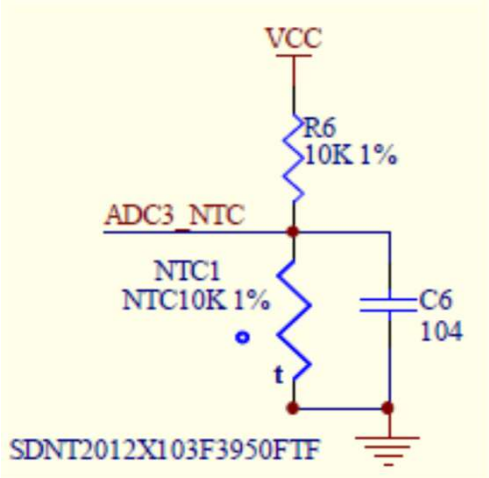
  - R_T：是指在温度 T（单位为开氏温度K）时的NTC热敏电阻阻值。
  - R_N：在额定温度 T_N（单位为开氏温度K）时的 NTC 热敏电阻阻值。
  - T：规定温度（单位为开氏温度K）。
  - B：NTC 热敏电阻的材料常数，又叫热敏指数。
  - exp：以自然数e 为底的指数（ e = 2.71828 …）。

温度的测量和串口显示
--信号输入电路

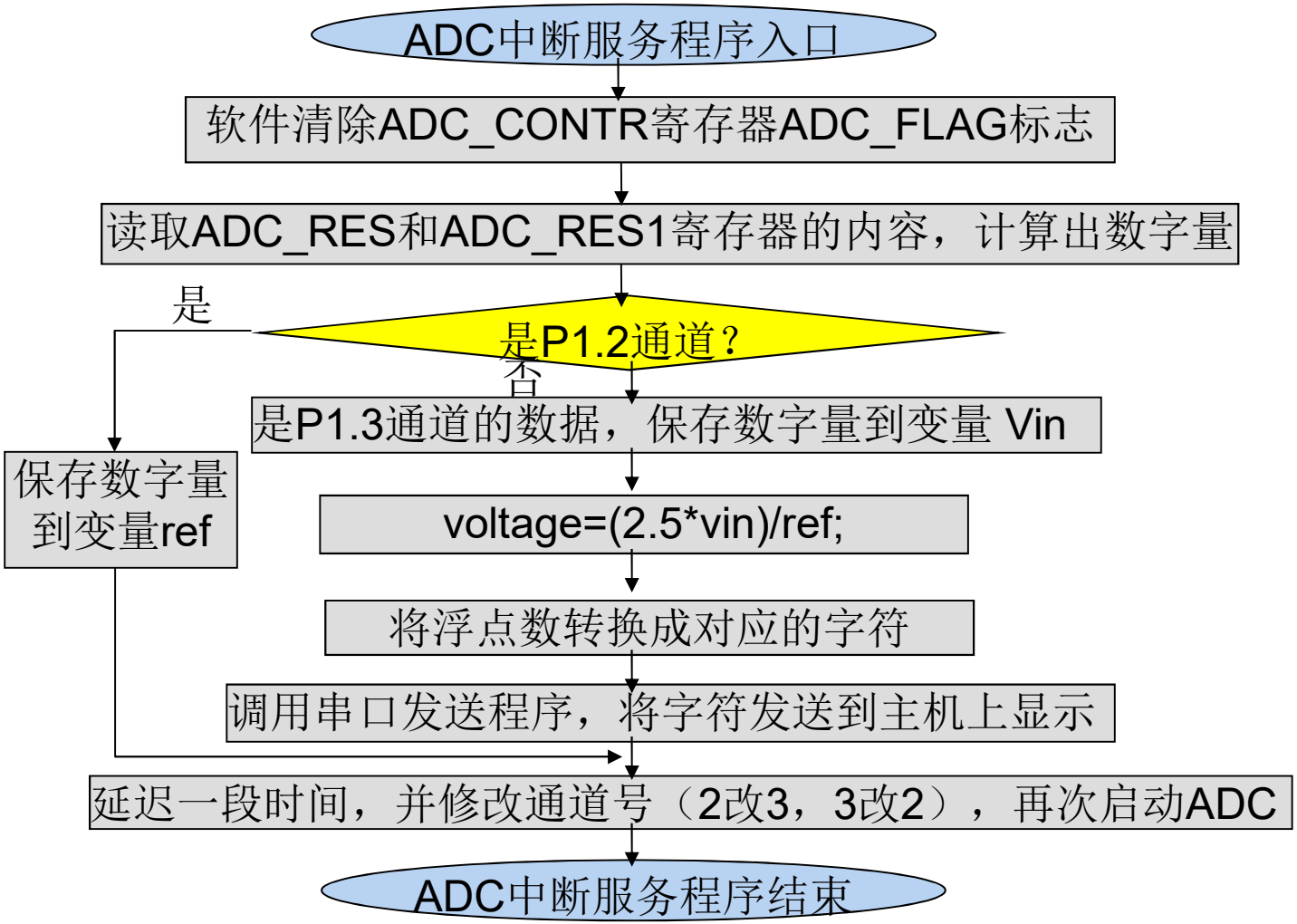
在图所示的电路中，热敏电阻作为分压网络的一部分与R6电阻连接在一起

- ADC_NTC网络连接到单片机P1.3引脚上。
  - ADC_NTC上的电压由下式确定：

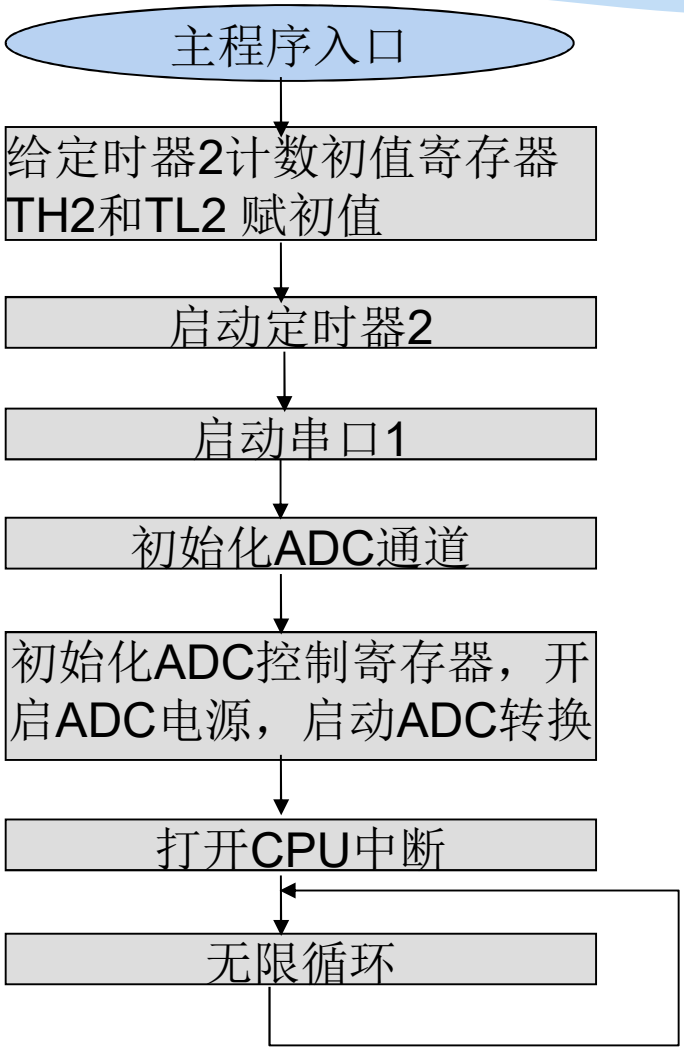
$$V_{ADC\_NTC}=(V_{CC}\times R_{NTC1})/(R_{NTC1}+R_6)$$



温度的测量和串口显示
--ADC中断程序处理流程



温度的测量和串口显示
--主程序处理流程



温度的测量和串口显示
--具体实现过程

【例】利用外部参考电压精确测量外部输入电压值C语言描述

```
#include "reg51.h"
#include "stdio.h"
#define OSC       18432000L        //声明单片机主时钟频率为18432000Hz
#define BAUD      9600             //声明单片机串口1通信波特率时钟
#define ADC_POWER    0x80          //声明ADC_POWER的值为0x80
#define ADC_FLAG   0x10            //声明ADC_FLAG的值为0x10
#define ADC_START   0x08           //声明ADC_START的值为0x08
#define ADC_SPEEDLL   0x00         //声明ADC_SPEEDLL的值为0x00
#define ADC_SPEEDL   0x20          //声明ADC_SPEEDH的值为0x20
#define ADC_SPEEDH   0x40          //声明ADC_SPEEDH的值为0x40
#define ADC_SPEEDHH  0x60          //声明ADC_SPEEDHH的值为0x60
```

温度的测量和串口显示
--具体实现过程

```
sfr T2H       =0xD6;          //声明T2H寄存器的地址为0xD6
sfr T2L       =0xD7;          //声明T2L寄存器的地址为0xD7
sfr AUXR      =0x8E;          //声明AUXR寄存器的地址为0x8E
sfr ADC_CONTR =0xBC;          //声明ADC_CONTR寄存器的地址为0xBC
sfr ADC_RES   =0xBD;          //声明ADC_RES寄存器的地址为0xBD
sfr ADC_RESL  =0xBE;          //声明ADC_RESL寄存器的地址为0xBE
sfr P1ASF     =0x9D;          //声明P1ASF寄存器的地址为0x9D
unsigned char ch=2;           //声明无符号char类型全局变量ch=2
float voltage=0;              //声明浮点类型全局变量voltage=0
unsigned char tstr[5];        //声明无符号char类型全局数组tstr
unsigned int ref=0,vin=0;     //声明无符号int类型全局变量ref和vin
```

温度的测量和串口显示
--具体实现过程

```
void SendData(unsigned char dat)          //声明串口发送子函数
{
        while(!TI);                       //如果TI不为1，正在发送数据，则等待
        TI=0;                             //TI标志清零
        SBUF=dat;                         //dat写入串口1发送寄存器SBUF中
}
```

温度的测量和串口显示
--具体实现过程

```
void adc_int() interrupt 5                    //声明ADC中断服务程序adc_int
{
        unsigned char i=0;                    //声明无符号char类型变量i
       unsigned long int j=0;                 //声明无符号long int类型变量j
       ADC_CONTR &=!ADC_FLAG;                 //清ADC_FLAG变量
       if(ch==2)                              //如果是参考电压源TL431通道
       {
           ref=(ADC_RES*4+ADC_RESL);
                                  //将参考电压所对应的数字量保存到变量ref中
       }
```

温度的测量和串口显示
--具体实现过程

```
else if(ch==3)                          //如果是热敏电阻分压输入通道
{
   vin=(ADC_RES*4+ADC_RESL);
                                        //将分压所对应的数字量保存到变量vin中
   voltage=(2.5*vin)/ref;               //计算分压的浮点电压值
   sprintf(tstr,“%1.4f”,voltage);   //转换成对应的字符串tstr
   SendData(‘\r’);                      //串口发送回车符
    SendData(‘\n’);                     //串口发送换行符
    for(i=0;i<5;i++)
     SendData(tstr[i]);                 //串口发送分压对应的ASCII字符
}
```

温度的测量和串口显示
--具体实现过程

```
if(ch==2)                          //如果当前通道是2
        ch=3;                      //则将通道号修改为3
  else if(ch==3)                   //如果当前通道是3
        ch=2;                      //则将通道号修改为2
  for(j=0;j<=80000;j++);           //延迟一段时间
  ADC_RES=0;                       //ADC_RES寄存器清零
  ADC_RESL=0;                      //ADC_RESL寄存器清零
  ADC_CONTR=ADC_POWER |ADC_SPEEDLL | ADC_START | ch;
}
```

温度的测量和串口显示
--具体实现过程

```
void main()
{       unsigned int i;
        SCON=0x5A;                          //串口1为8位可变波特率模式
        T2L=65536-OSC/4/BAUD;               //写定时器2低8位寄存器T2L
        T2H=(65536-OSC/4/BAUD)>>8;          //写定时器2高8位寄存器T2H
        AUXR=0x14;                          //定时器2不分频，启动定时器2
        AUXR|=0x01;                         //选择定时器2为串口1的波特率发生器
        P1ASF=0xFF;                         //P1端口作为模拟输入
        ADC_RES=0;                          //清ADC_RES寄存器
        ADC_CONTR=ADC_POWER|ADC_SPEEDLL | ADC_START | ch;
        for(i=0;i<10000;i++);                //延迟
        IE=0xA0;                            //CPU允许响应中断请求，使能ADC中断
        while(1);                           //无限循环
}
```

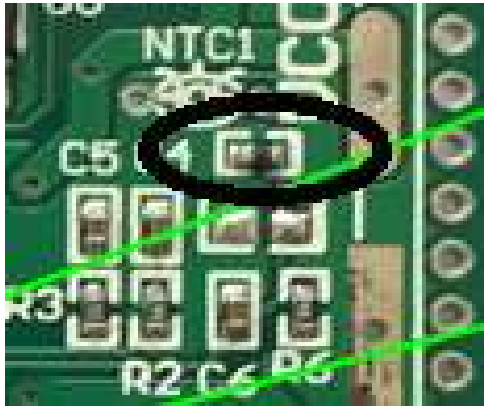
温度的测量和串口显示
--测试和验证

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18.432MHz。
- 单击下载/编程按钮，按前面的方法下载设计到STC单片机。
- 在STC-ISP软件右侧串口中，选择串口助手标签。在该标签串口界面下，按下面设置参数：
  - 串口：COM3（读者根据自己电脑识别出来的COM端口号进行设置）
  - 波特率：9600。
  - 校验位：无校验。
  - 停止位：1位。
- 单击打开串口按钮。

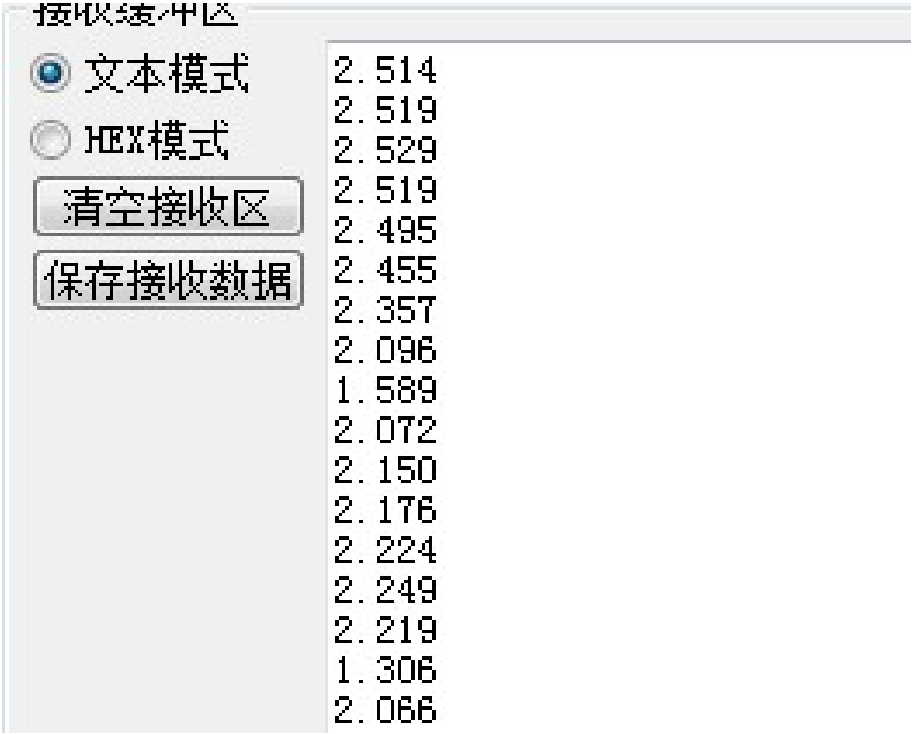
温度的测量和串口显示
--测试和验证

- 用电热吹风接近STC学习板上的热敏电阻，黑圈的位置。



温度的测量和串口显示
--测试和验证

- 观察串口的输出结果。
  - 当电烙铁瞬间接触热敏电阻时，其电压可以降低到1.306V。



温度的测量和串口显示
--测试和验证

- 将单片机的供电电压通过J12插座上的插针和导线连接到到P1.3插孔上，观察串口输出结果。
  - 从图中可以看出，单片机的供电电压非常稳定，值的变化范围为0.01V，即：10mV。