

TSic 系列高精度温度传感器

TSic 产品系列由集成芯片和已校准的温度传感器组成。内部集成了用于模拟或数字信号输出的信号转换器。它提供的测量精度从 $\pm 0.5^{\circ}\text{C}$ 到非常低的 $\pm 0.1^{\circ}\text{C}$ ，并且有（ 0.1°C ）到高至（ 0.034°C , 0.004°C ）的信号分辨率，以最佳化温度控制。使用 TSic 测量非常简单，它不仅提供出色的精度，而且具有长时间稳定性。由于他的低功耗（典型值为 $30\mu\text{A}$ ），也非常适合做数据记录器、数字温度计以及温度监控和测量应用。

TSic 具有以下优点：

更高的精度，精度可高达 $\pm 0.1^{\circ}\text{C}$ ，测量范围从 -50 到 150°C ，整个测量范围都保持较高的精度；

TSic 具有较高的性价比，IC 已经过校准测试，用户不需校准，具备更长的稳定性；

TSic 采用单总线输出方式，接口简单，用户可根据自身要求灵活应用；

TSic 系列包括标准 $0\sim 1\text{V}$

模拟输出电压和 11bit 数字信号输出方式，用户可根据自己的需求进行选择；

TSic 采用 DSP 技术，信号输出速率可以达到每 0.1s 输出一次；

TSic 工作电流非常低，功耗小，适用于移动设备；

宽电压操作（ $3.0\sim 5.5\text{V}$ ），可以工作于多种电源系统，使系统设计更加灵活；

TSic 系列温度传感器 IC 为低功耗特殊设计，用于建筑物内自动温度感知、测量装置、工业、医学/医药技术和移动应用。

TSic 采用与绝对温度成比例(PTAT)的高精度参考源，集成低功耗、精密 ADC 和片上 DSP 核，具有电可擦除存储器来精确地校正输出温度信号。

TSic 系列温度传感器全都经过测试和校正，在交付时就可获得绝对的测量准确性，不需要进一步的校正。使用 TSic 测量温度是非常简单的，它可以达到杰出的精度和长期的稳定性。

在 40°C 的范围内具有 $\pm 0.1^{\circ}\text{C}$ 的精度。

北京华夏日盛科技有限公司

Beijing HuaXiaRiSheng Technology CO., Ltd

电话: 010-88458650 传真: 010-88458449

<http://www.bjhxrs.com.cn>

TSic 506 特性

精度: $\pm 0.1^{\circ}\text{C}$, 在 5°C 到 45°C 范围;

分辨率: 0.034°C

测量范围: 10°C 到 60°C

信号输出, 每 0.1s 一次 (数字)

供电电压: 3.0V 到 5.5V, 高精度测量时供电电压范围 $V+ = 4.5\text{V}$ 到 5.5V

低电源电流, 在 25°C 、 $V+=3.3\text{V}$ 时供电电流的典型值为 $30\mu\text{A}$ 。

信号输出

输出信号的公式:

数字输出: $T = (\text{Digital_signal}/2047 * (\text{HTLT}) + \text{LT}) [^{\circ}\text{C}]$

模拟输出信号: $T = (\text{Sig[Volt]} * (\text{HTLT}) + \text{LT}) [^{\circ}\text{C}]$

比例输出信号: $T = ((\text{Sig[V]})/VDD[V]0.1)/0.8 * (\text{HTLT}) + \text{LT}) [^{\circ}\text{C}]$

输出例子

		测量范围: -10°C 到 60°C (14°F 到 140°F)		
Temp (°C)	Temp (°F)	数字值 (TSic 506)	模拟 0-1V [V] (TSic 501)	模拟比例 (TSic 503) 10%~90% ($V+ = 5.0\text{V}$)
-10 ¹	14	0x000	0.000	10.0% of $V+$ (e.g. 0.5V)
0	32	0x124	0.143	21.4% of $V+$ (e.g. 1.07V)
25	77	0x3FF	0.500	50.0% of $V+$ (e.g. 2.5V)
60 ²	140	0x7FF	1.000	90.0% of $V+$ (e.g. 4.5V)

¹ LT = -10, ² HT = 60 作为温度计算的标准值。

绝对最大额定参数

参数	MIN	MAX	单位
供电电压 ($V+$)	-0.3	6.0	V
在模拟 I/O 引脚上的电压 (V_{INA} , V_{OUTA})	-0.3	$V_{\text{DDA}}+0.3$	V
储存温度范围 (T_{stor})	-50	150	$^{\circ}\text{C}$

北京华夏日盛科技有限公司

Beijing HuaXiaRiSheng Technology COL., Ltd

电话: 010-88458650 传真: 010-88458449

<http://www.bjhxrs.com.cn>

运行条件

参数	MIN	TYP	MAX	单位
对地电源 ¹ 电压 (V ₊)	2.97	5.0	5.5	V
电源电流 (I _{V+}) 在 V ₊ = 3.3V, RT	25	30	60	μA
运行温度 ² 范围 (T _{amb})	-10		60	°C
输出负载电容 (C _L)			15	nF
V ₊ 和地 ³ 之间的外部电容	80	100	470	nF
信号和地 (或 V ₊) 之间的负载电阻	47			KΩ

¹ 电源电压 4.5V – 5.5V 最精确。2.97V – 4.5V 电源电压精度降低。

² 输出信号被限制在该温度范围内。

³ 尽可能接近 TSic V₊和地。

温度精度

参数	MIN	TYP	MAX	单位
非常高精度范围器件 5° 到 45°C				
5°C to 45°C	-0.1	+/-0.07	+0.1	°C ⁴
-10°C to 5°C	-0.2	+0.1	+0.2	°C
+45°C to +60°C	-0.2	+0.1	+0.2	°C
测量范围: -10°C 到 +60°C (±3°C)				

TSic 的 ZACwire 通信协议

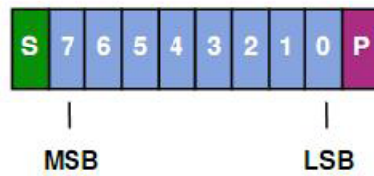
ZACwire 是一种单线双向通信协议。位编码类似于时钟信号嵌入数据信号中的曼彻斯特编码（信号的下降沿以固定周期产生）。这样协议对两个 IC 之间通信时波特率的差异就很不敏感。

在终端用户应用中，TSic 传送温度信息，另一个 IC（通常是 MCU）通过 ZACwire 读温度数据。

TSic 的温度传输包

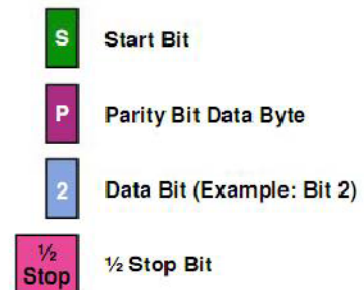
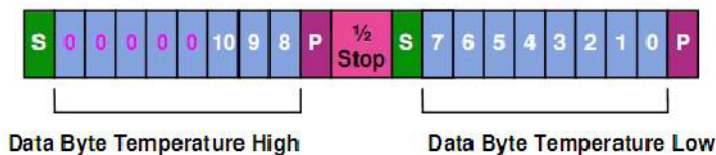
TSic 发送长度为 1 字节的包。这些包由 1 个起始位、8 个数据位和一个奇偶校验位组成。常用的波特率是 8kHz（125us 位宽）。信号通常是高，当传输开始，起始信号发生，紧接着是数据位（先高后低），包的结尾是偶校验位。

ZACwire™ Transmission Packet



TSic 提供 11 位分辨率的温度数据，11 位数据不能在单个的包中传递。一个来自 TSic 的完整的温度传输数据包由 2 个包组成。第一个包包含高 3 位温度信息，第二个包包含低 8 位温度信息。在第一个传输包的末端和第二个传输包的开始之间有一个宽度为二分之一信号位宽（Bit Window）的高信号（停止位）。

2 DATA Byte Packet – 11-Bit Temperature Output



位编码

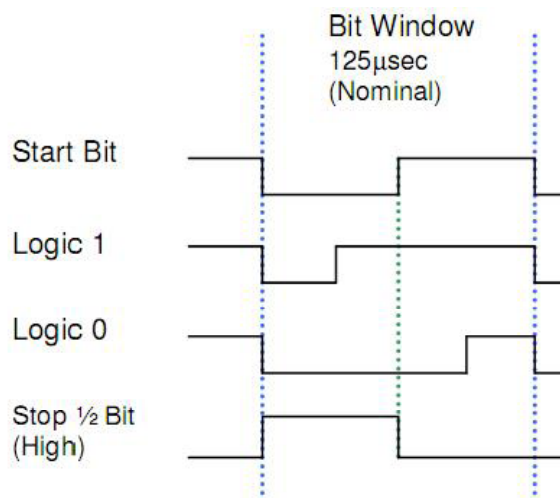
位格式是占空比编码：

起始位 => 50%占空比，用于设置闸门时间

逻辑 1 => 75%占空比

逻辑 0 => 25%占空比

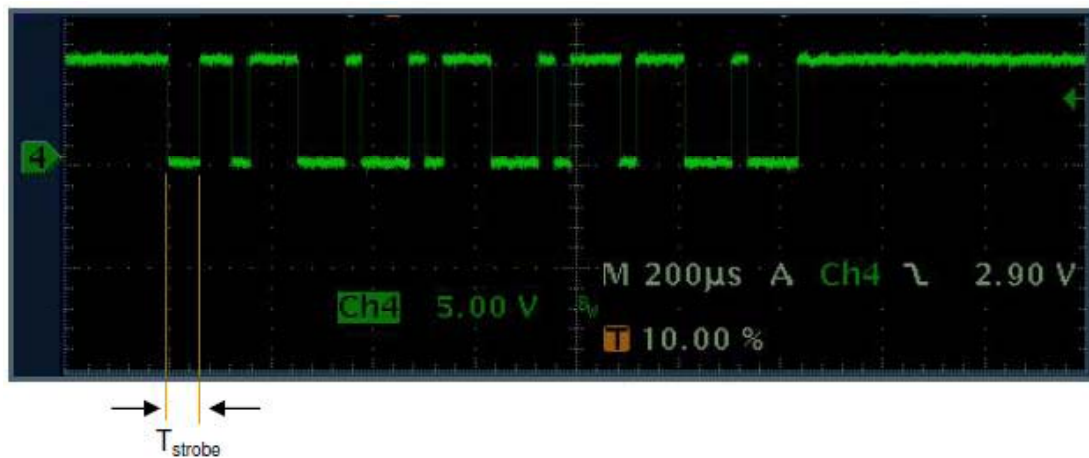
停止位 => 信号高电平，持续半位宽度时间。在包中的字节之间有一个半停止位时间。



北京华夏日盛科技有限公司

Beijing HuaXiaRiSheng Technology COL., Ltd
 电话: 010-88458650 传真: 010-88458449
<http://www.bjhxrs.com.cn>

一条 ZACwire 传输的示波器踪迹演示了这种为编码。下面的图显示了一个正在被传输的 96Hex 的信号包。因为 96Hex 是偶校验，校验位是 0。



怎样读包

测量起始位下降沿和上升沿之间的时间，该时间(T_{strobe})就是选通时间，其宽度为位宽度(Bit Window)的一半。在下一个下降沿，等待一个周期等于 T_{strobe} 的时间，然后采样 ZACwire 信号。在这段时间中，存在于信号中的数据就是正在被传输的位。因为每个位都以一个下降沿开始，所以每个传输的采样窗口都会复位。这意味着从起始位开始的比特流将不会发生误差，因此它可以带有象 RS232 这样的协议。建议，当捕获起始位时，ZACwire 信号的采样率至少为 16x 的正常波特率。因为正常波特率是 8kHz，当捕获 T_{strobe} 时要求最小 128kHz 的采样速率。

怎样用微控制器 (MCU) 读包

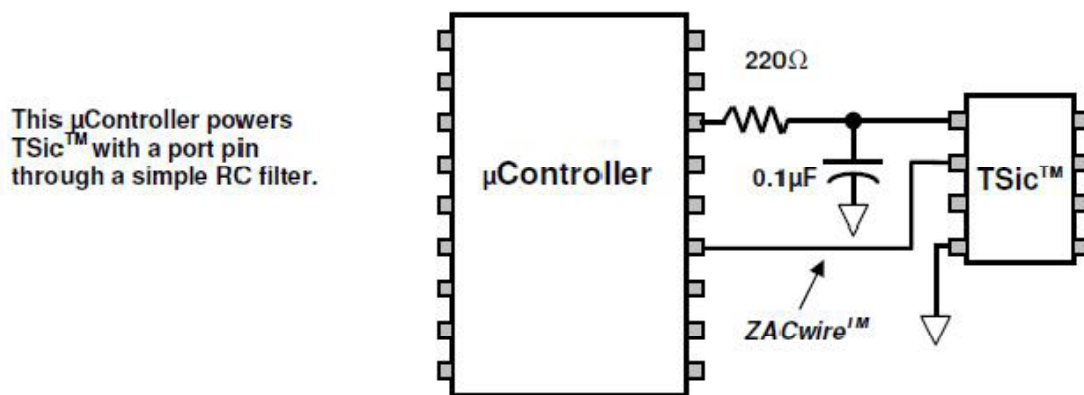
理想的情况是，当连接 ZACwire 信号到 MCU 的引脚时，能够在下降沿引发一个中断。当起始位的下降沿产生时，它引发 MCU 转向中断服务程序 ISR。该 ISR 进入一个计数循环，递增一个内存位置，直到看见 ZACwire 信号的上升沿，该计数值就是 T_{strobe} 。在获得 T_{strobe} 后，ISR 就可以简单地等待下面 9 个下降沿（8 个用于数据，1 个用于奇偶校验）。在每个下降沿之后，ISR 等待 T_{strobe} 期满然后采样下一个位。ZACwire 线要由一个强 CMOS 推挽驱动器来驱动。在一个噪声环境中，当 ZACwire 要驱动长时间(>2m)相互连接的 MCU 时，需要奇偶校验位。对于在没有噪声干扰环境中的系统，用户可以选择让 MCU 忽略奇偶校验位。

TSic 传递需要多长时间？

如果 TSic 正在被一个 ISR 读，MCU 为了读数据要中断多长时间？TSic 的更新速率可编程到 10Hz (0.1ms 响应时间)。进行一次温度读，ISR 需要大约 2.7ms，因此 MCU 花费大约 2.7%的时间读温度传输。

如果一个实时系统不能忍受 TSic 中断 MCU 时的解决方法

一些实时系统不能忍受 TSic 中断 MCU。在这种情况下，MCU 必须发起读温度操作，这可以用 MCU 的另一个引脚提供 VDD 到 TSic 来完成。TSic 在电源上电后大约 65ms 到 85ms 将传送它的第一个温度读数。当 MCU 要读温度时，首先用它的一个端口引脚供给 TSic 电源，大约 65ms 到 85ms 后它将收到一个温度传送。如果在这个时间内，一个更高周期的中断发生，MCU 可以简单地关闭 TSic 电源以确保 TSic 不会引起中断，或者出现更高周期的 ISR 在传送中间结束的情况。这种开关 TSic 的方法有额外的好处，就像掉电模式，可以将静态电流从通常的 45 μ A 减小到 0。TSic 是一种混合信号 IC，要使用低噪声 VDD 供电以提供最佳性能。通过 MCU 的一个引脚供电易受存在于 MCU 电源的数字噪声的影响，因此，用 MCU 的端口引脚供电时最好用一个简单的 RC 滤波器。见图 7。



8051 C 语言读 ZACwire 的例子

下面我们给出一个 TSic 506 的应用例子，使用其他型号的传感器时测量范围和计算公式需要调整。TSic 506 采用 T092 封装，以便于用冷连接工艺安装，热装配工艺会影响传感器已校准的精度。MCU 采用宏晶 STC11L16XE 8051 单片机，这是 3.3V 电源的芯片，便于锂电池供电，晶振选 11.0592MHz。按照 3.6 节说明，在 5V 供电时传感器可以提供最好精度，实际测试用 3.7V 锂电池供电对精度影响不大。我们在 4.4 节提到 ZACwire 线要由一个强 CMOS 推挽驱动器来驱动。我们选择 P3.7 连接 ZACwire，用 P3.6 为 TSic 506 供电。实验发现，按照图 7 用普通的 8051 端口为 TSic506 供

北京华夏日盛科技有限公司

Beijing HuaXiaRiSheng Technology CO., Ltd

电话: 010-88458650 传真: 010-88458449

<http://www.bjhxrs.com.cn>

电, 会造成 TSic 506 工作不稳定, STC11L16XE 是可以选择端口工作模式的, 我们将 P3.6 配置为强推挽输出就可以解决这个问题。ZACwire 线用普通的 8051 端口就可以。如果你不用中断, 也不在乎功耗, TSic 506 可以直接用电源供电, 但 RC 低通滤波器应该保留, 至少 0.1uF 的去耦电容是需要的。

8051 C 语言代码

下面的代码使用查询方式, 未使用中断。

```
#include "include\STC11.h"

typedef unsigned char UINT8; // 无符号 8 位整型变量

typedef unsigned short UINT16; // 无符号 16 位整型变量

// TSic 端口设置

sbit PWR_PIN = P3^6;

sbit SIG_PIN = P3^7;

UINT8 xdata tp_raw_diap[4] = {0x30, 0x31, 0x32, 0x33}; // 用于显示的原始温度数据

// tp_buf 存放 ASCII 格式温度数据, 用于 LCD 显示。

UINT8 xdata tp_buf[16] = {"012345678 "}; // 随便放几个数初始化

UINT8 code table[]={"0123456789ABCDEF"}; // 用于查找表显示 1602 十六进制数对应的字符

// 1602 端口设置

sbit rs=P1^0;

sbit rw=P1^1;

sbit en=P1^2;

/*****

* TSIC FUNCTION

*****/

void TSIC_ON() {PWR_PIN=1;}

void TSIC_OFF() {PWR_PIN=0;}

void TSIC_INIT()

{

PWR_PIN=0; /* power OFF*/
```

北京华夏日盛科技有限公司

Beijing HuaXiaRiSheng Technology COL., Ltd

电话: 010-88458650 传真: 010-88458449

<http://www.bjhxrs.com.cn>

```
SIG_PIN=1; /* signal */

}

/* 延迟 x*10us, 晶振为 11.0592MHz。*/

void delay_10us(UINT8 x) // uchar 8 位, 最大为 255

{

    int i, j;

    for(i=0; i<x; i++) {

        j = 6;

        while(--j);

    }

}

// 延迟 x ms, 晶振为 11.0592MHz。由 delay_ms 项目校准

void delay_ms(int x) // int 16 位

{

    int i, j;

    for(i=0; i<x; i++) {

        j = 785;

        while(--j);

    }

}

/* 写指令到 1602 */

void write_com(UINT8 com)

{

    delay_ms(5);

    rs = 0; //选择写指令

    rw = 0;

    P0 = com;

    en = 1; //en 先 1 后 0 为高脉冲

    delay_ms(5);
```

```
en = 0;

}

/* 写数据到 1602 */

void write_dat(UINT8 dat)

{

    delay_ms(5);

    rs = 1; //选择写数据

    rw = 0;

    P0 = dat;

    en = 1; //en 先 1 后 0 为高脉冲

    delay_ms(5);

    en = 0;

}

/* 对 1602 进行初始化，具体过程应严格参照产品说明书 */

void init_1602(void)

{

    rw = 0; //读/写 置低

    write_com(0x38); //设置 16x2 显示，5x7 点阵，8 位数据口

    write_com(0x08); //显示关闭

    write_com(0x01); //显示清屏，数据指针以及所有显示清零

    write_com(0x06); //读写数据，光标及数据指针自动加 1

    write_com(0x0c); //开显示，光标不显示

}

/* 显示字符串 */

void display(UINT8 * buf, UINT8 length)

{

    UINT8 i;

    for(i=0; i<length; i++) {

        write_dat(buf[i]); // 显示收到的数据
```

```
}  
  
}  
  
/* 取 TSic 温度 */  
  
UINT16 getTSicTemp (UINT16 *temp_value16)  
  
{  
  
    UINT16 temp_value1 = 0;  
  
    UINT16 temp_value2 = 0;  
  
    UINT8 i;  
  
    UINT16 Temperature;  
  
    UINT8 parity;  
  
    TSIC_ON(); // 打开 TSic  
  
    delay_10us(12); // 延时 120us, 等待稳定  
  
    // first data byte  
  
    while (SIG_PIN == 1); // wait until start bit starts  
  
    while (SIG_PIN == 0); // wait, TStrobe  
  
    // read 8 data bits and 1 parity bit  
  
    for (i = 0; i < 9; i++) {  
  
        while (SIG_PIN==1); // wait for falling edge  
  
        delay_10us(6);  
  
        if (SIG_PIN==1)  
  
            temp_value1 |= 1 << (8-i); // get the bit  
  
        else  
  
            while (SIG_PIN == 0); // wait until line comes high again  
  
    }  
  
    // second byte  
  
    while (SIG_PIN == 1); // wait until start bit starts  
  
    while (SIG_PIN == 0); // wait, TStrobe  
  
    // read 8 data bits and 1 parity bit  
  
    for (i = 0; i < 9; i++) {
```

```
while (SIG_PIN==1); // wait for falling edge

delay_10us(6);

if (SIG_PIN==1)

temp_value2 |= 1 << (8-i); // get the bit

else

while (SIG_PIN == 0); // wait until line comes high again

}

TSIC_OFF(); // switch TSic off

// check parity for byte 1

parity = 0;

for (i = 0; i < 9; i++)

if (temp_value1 & (1 << i))

parity++;

if (parity % 2)

return 0;

// check parity for byte 2

parity = 0;

for (i = 0; i < 9; i++)

if (temp_value2 & (1 << i))

parity++;

if (parity % 2)

return 0;

temp_value1 >>= 1; // delete parity bit

temp_value2 >>= 1; // delete parity bit

Temperature = (temp_value1 << 8) | temp_value2;

*temp_value16 = Temperature;

return 1; // parity is OK

}

/* 分离原始数据用于显示 */
```

```
void raw_data_bit(UINT16 temp_value)

{

char i,tmp;

for(i=0;i<4;i++) tp_raw_diap[i] = 0x20;

tmp = (temp_value & 0xf000) >> 12;

tp_raw_diap[0] = table[tmp];

tmp = (temp_value & 0x0f00) >> 8;

tp_raw_diap[1] = table[tmp];

tmp = (temp_value & 0x00f0) >> 4;

tp_raw_diap[2] = table[tmp];

tmp = temp_value & 0x000f;

tp_raw_diap[3] = table[tmp];

}

/* 分离各数位用于显示 */

void data_bit(int new_temp)

{

UINT8 tp_dig[6] = {"012345"}; // 存放十进制每位数字

if(new_temp<0) {

tp_dig[0]=0x2d; // 如果小于 0，显示负号;否则显示正号

new_temp = new_temp * (-1);

} else {

tp_dig[0]=0x2b;

}

// 分离各数位

tp_dig[1] = new_temp/10000; // 百位

tp_dig[2] = (new_temp%10000)/1000; // 十位

tp_dig[3] = (new_temp%1000)/100; // 个位

tp_dig[4] = (new_temp%100)/10; // 小数第一位

tp_dig[5] = new_temp%10; // 小数第二位
```

```
// 转换温度的每个位到 ASCII

tp_buf[0] = tp_dig[0]; // 正负号

tp_buf[1] = table[tp_dig[1]]; // 百位

tp_buf[2] = table[tp_dig[2]]; // 十位

tp_buf[3] = table[tp_dig[3]]; // 个位

tp_buf[4] = 0x2e; // 小数点

tp_buf[5] = table[tp_dig[4]]; // 小数点后第一位

tp_buf[6] = table[tp_dig[5]]; // 小数点后第二位

tp_buf[7] = 0xdf; // 显示"°"

tp_buf[8] = 0x43; // 显示"C"

}

/* 主函数 */

void main(void)

{

    UINT16 temp_value;

    float Temp_float;

    int t_value;

    P3M1 = 0x00; //0000 0000

    P3M0 = 0x40; //0100 0000 P3.6 强推挽输出(作为 TSic 506 的可控电源)，其他准双向 I/O 口。

    init_1602(); // 初始化 1602

    write_com(0x80); // 定位，在第一行显示。从第 0 个字符开始显示，自动移光标

    display(tp_raw_diap, 4); // 在第一行显示测试数据

    write_com(0xc0); // 定位，在第二行显示。从第 0 个字符开始显示，自动移光标

    display(tp_buf, 16); // 在第二行显示测试数据

    while(1) {

        TSIC_INIT(); // init the I/O pins used for the TSic

        TSIC_OFF(); // switch the TSic off until use

        if (getTSicTemp(&temp_value)) {
```

```
//temp_value=0x124; //用于测试负数

raw_data_bit(temp_value); // 分离原始数据用于显示

Temp_float = ((float)temp_value / 2047 * 70) -10; // conversion equation from
TSic's

t_value = (int)(Temp_float * 100);

data_bit(t_value); // 分离各数位用于显示

}

// 显示温度

write_com(0x80); // 定位, 在第一行显示原始温度数据, 从第 0 个字符开始
显示, 自动移光标

display(tp_raw_diap, 4); // 在第一行显示接收的原始温度数据

write_com(0xc0); // 定位, 在第二行显示温度, 从第 0 个字符开始显示, 自
动移光标

display(tp_buf, 16); // 在第二行显示接收的温度

delay_ms(500);

}

}
```

温度数据显示在 1602 液晶屏上, 第一行显示原始数据, 第二行显示计算后的温度。

程序运行结果见图 8。



关于延迟的校准

因为我们的程序与时间相关，延时要严格校准。可以用定时器，但太麻烦了，单任务时我通常用循环延时，下面是对 P2.0 引脚进行 10us 延时的 C 程序文件 delay_10us.c。

```
#include <STC11.h>

typedef unsigned char BYTE;

// 延迟 x*10us，晶振为 11.0592MHz

void delay_10us(BYTE x)

{

    int i,j;

    for(i=0;i<x;i++) {

        j = 3;

        while(--j);

    }

}

void main(void)

{

    while(1) {

        P2=0xfe;

        delay_10us(1);

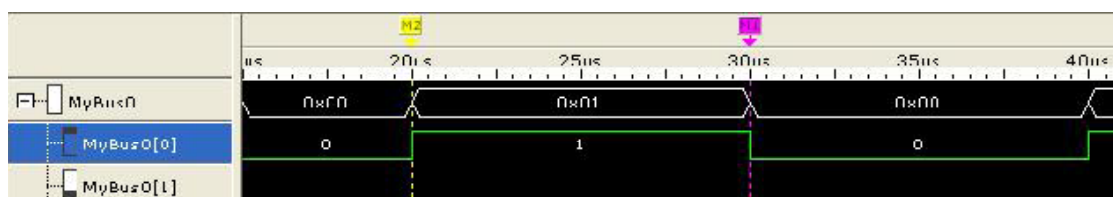
        P2=0xff;

        delay_10us(1);

    }

}
```

I/O 端口 P2.0 10us 延时的逻辑分析仪测试结果见图 9。



北京华夏日盛科技有限公司

Beijing HuaXiaRiSheng Technology CO., Ltd

电话: 010-88458650 传真: 010-88458449

<http://www.bjhxrs.com.cn>