

温度传感器NST1001应用指南

AN-12-0011

作者：Yuyun Yang



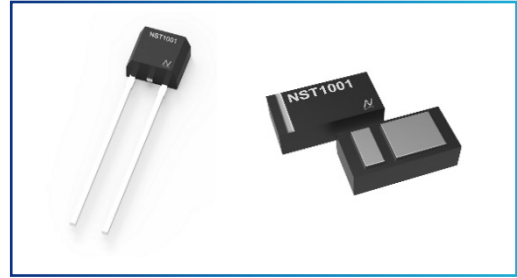
温度传感器NST1001应用指南

说明

NST1001是一款高精度双引脚数字脉冲输出型温度传感器，属于纳芯微电子D-NTC™系列。NST1001特有的脉冲计数型数字输出以及在宽温度范围内高精度的特性，使其可直接与MCU的GPIO连接使用，保障测量精度的同时，降低了MCU开销并减少了成本。NST1001的测温范围为-50°C到150°C，满足汽车、工业、家用电器等场合对温度监测的需求。NST1001仅有两个引脚，可轻松应用于双线数字温度探针或者直接替代NTC热敏电阻。相较于传统的NTC热敏电阻，NST1001的使用更简单、数字化、更精确，无需借助系统校准或软硬件补偿就可以实现全温区内的高精度温度检测。NST1001内置EMI滤波器，使得其可以用于存在较大干扰的设备中。由于NST1001工作电流极低，可以通过MCU的GPIO供电，因此也可以用于对功耗要求特别严格的无线物联网传感器节点中。封装形式方面，NST1001采用了标准的TO-92S和DFN2L封装。

功能特点

- 双引脚简化温度测量，无需额外器件
- 两脚连接，节约布线资源
- 宽温度范围：-50°C至150°C
- 高分辨率，可达0.0625°C
- 全温域内保持高精度：
 - 25°C ~ 45°C：±0.2°C（典型）
 - 20°C ~ 85°C：±0.5°C（最大）
 - 50°C ~ 150°C：±0.75°C（最大）
- 脉冲型数字输出，无需AD转换接口



- 单次温度转换时间50ms
- 转换电流仅30uA，零待机功耗
- 宽供电范围，1.65V ~ 5.5V
- 封装形式：
 - TO-92S-2L (4mm x 3mm)
 - DFN-2L (1.6mm x 0.8mm)

应用

- 数字输出温度探头
- 工业物联网
- 白电
- 冷链物流
- 电池管理
- 体温检测

温度传感器NST1001应用指南

目录

1. 管脚描述	2
2. 典型应用电路	3
2.1. 上拉电阻接法	3
2.2. 下拉电阻接法	4
2.3. R1与C1的选取	5
2.4. NST1001时序	6
3. 温度计算	7
4. 评估套件	8
4.1. 评估板硬件	8
4.1.1.评估板原理图	8
4.1.2.评估板PCB	8
4.1.3.评估板实物	9
4.2. 上位机	9
4.2.1.上位机介绍	9
4.2.2.串口驱动安装	9
4.2.3.评估板上位机软件使用	11
5. 参考例程	12
5.1. 外部中断	12
5.2. 外部中断计数，定时器中断读取温度值	14

温度传感器NST1001应用指南

1. 管脚描述

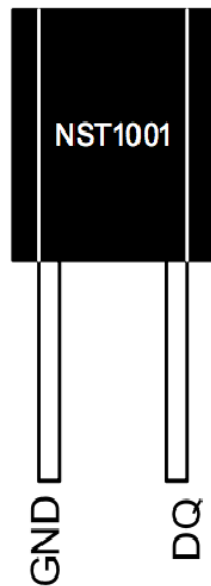
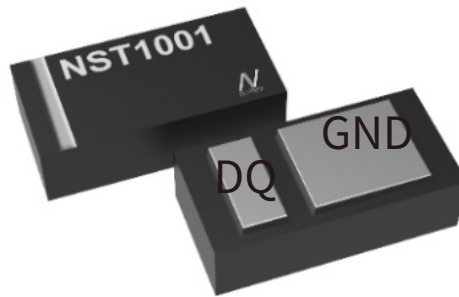


图1.1 NST1001封装图（上：DFN-2L，下：TO-92S-2L）

表1-1 NST1001管脚描述

管脚名称	I/O	描述
DQ	输入输出	供电及数据输出引脚
GND	GND	接地或下拉电阻到地

温度传感器NST1001应用指南

2. 典型应用电路

2.1. 上拉电阻接法

图2.1是NST1001的上拉电阻接法典型应用连接图。上拉电阻R1可以直接连到MCU的VDD，也可以通过一个单独的GPIO（图中为GPIO1）为NST1001供电，以便不用时关掉其供电来节省功耗。有些MCU的GPIO自带有可配置的上拉电阻，也可以替代外部电阻R1。在某些应用中为了提高对抗外部干扰的能力，可以在靠近NST1001处增加电容C1，C1的取值见章节2.3。图2.2是上拉电阻接法的典型DQ输出波形。

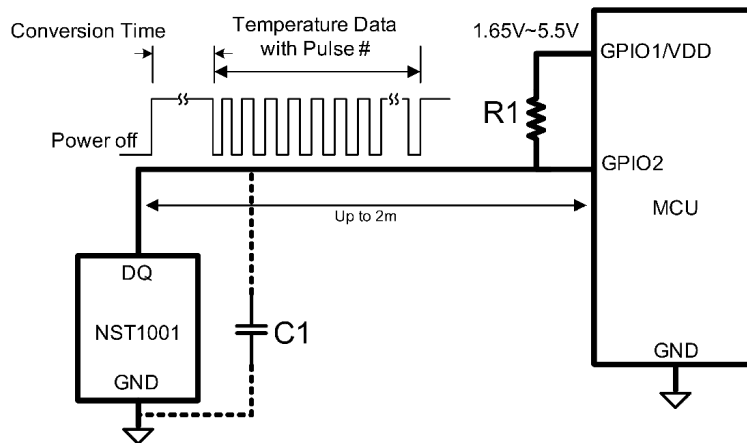


图2.1 NST1001上拉电阻连接应用电路示意图

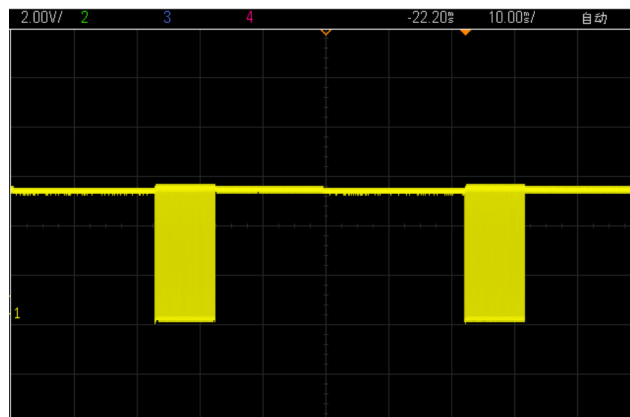


图2.2 NST1001上拉电阻DQ脉冲波形图

温度传感器NST1001应用指南

2.2. 下拉电阻接法

图2.3是NST1001的下拉电阻接法典型应用连接图，类似于常见的NTC温度采集方案。对应的输出波形见图2.4。

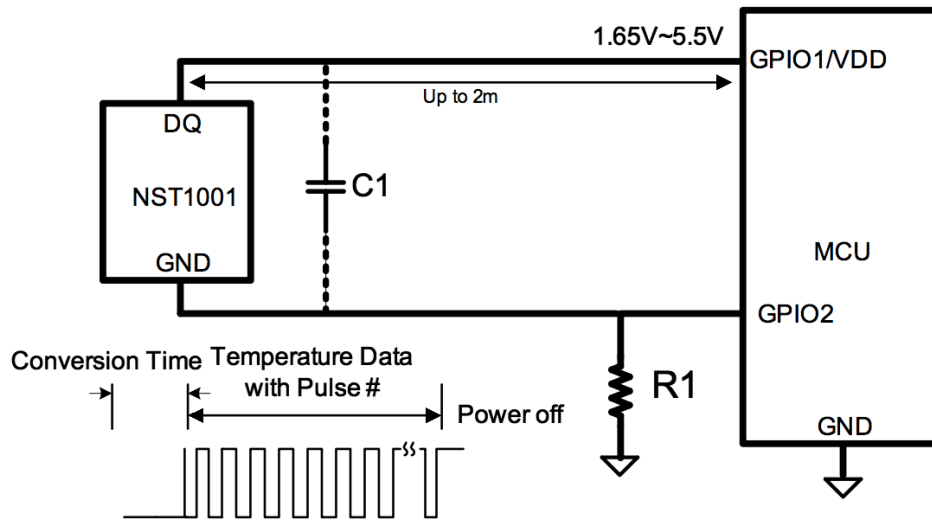


图2.3 NST1001下拉电阻连接应用电路示意图

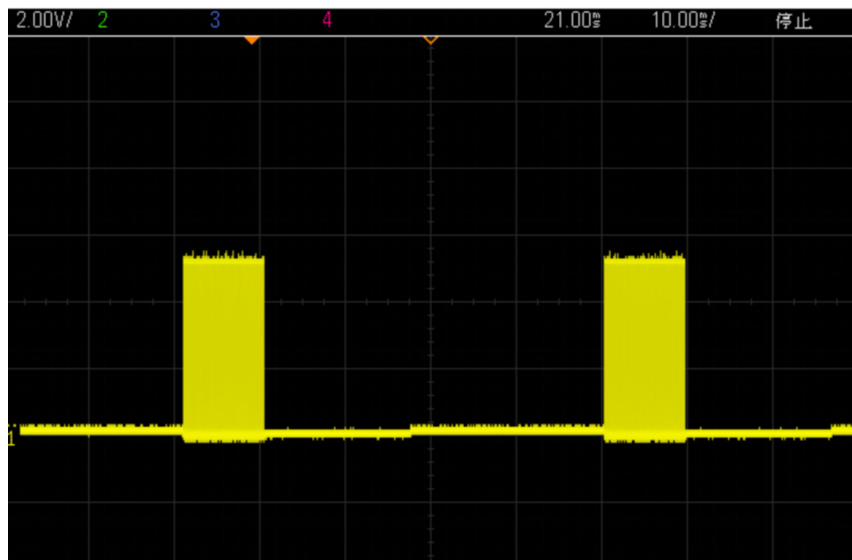


图2.4 NST1001下拉电阻DQ脉冲波形图

温度传感器NST1001应用指南

2.3. R1与C1的选取

上拉或下拉电阻R1可以选取500ohm到10kohm之间的值，具体的取值需要考虑最低工作电压、功耗和传输距离。由于NST1001进行温度转换时有最大可达45uA的电流，因此R1越小，在其上的压降就越小，给芯片的供电电压就越高。VDD的最小值可以用如下公式进行估算：

$$VDD > 1.45 + 45 * 10^{-6} * R1$$

另一方面，温度数据发送时的功耗随着电阻的减小而变大，因此为了最小化功耗，需要尽量采用更大的电阻。而在有些需要长距离传输的情况下，考虑到寄生电容对数据传输的影响，为了保证温度脉冲信号可以正常发送，电阻R1不能取得太大。

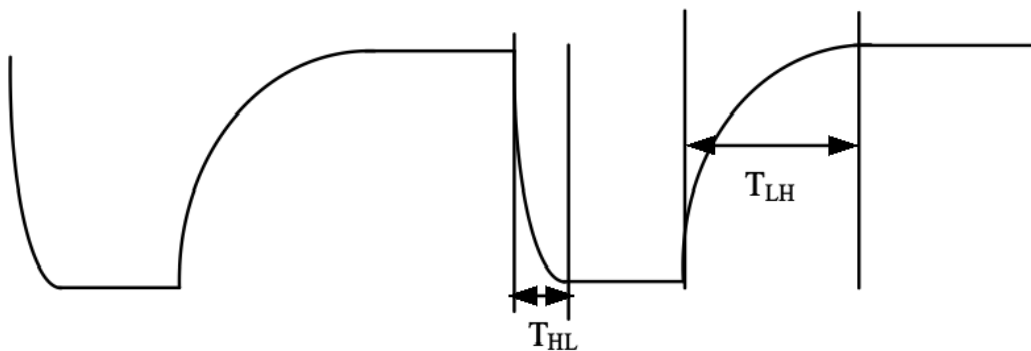


图2.5上拉模式DQ脉冲波形

如图2.5，在上拉模式中，由于DQ引脚开关电阻仅为50ohm左右，因此输出拉低速度一般较快，传输能力主要受限于输出由低变高时的速度。输出从低拉高到95%稳态值的时间TLH可以用如下公式进行计算：

$$T_{LH} = 3 * R1 * (C1 + Cpar)$$

其中，C1为外部滤波电容，Cpar为线束对地寄生电容。TLH需要小于DQ高脉冲的最短时间4us。假设R1为5.1kohm，不考虑Cpar，则C1不能大于261pF。

实际使用时，用户需要根据实际应用的情况，在最低工作电压、功耗和传输距离三个因素之间折中考虑。

温度传感器NST1001应用指南

2.4. NST1001时序

NST1001单次温度周期约50ms。NST1001上电后立即进入温度数据转换期，数据转换期典型值为24ms。之后进入脉冲通信期，输出温度脉冲数，最少1个脉冲，最多3201个脉冲。在单个脉冲周期内，总时长为8us，占空比为75%，即高电平持续时间约6us，低电平持续时间约2us。一次转换和输出脉冲的过程如图2.6所示。

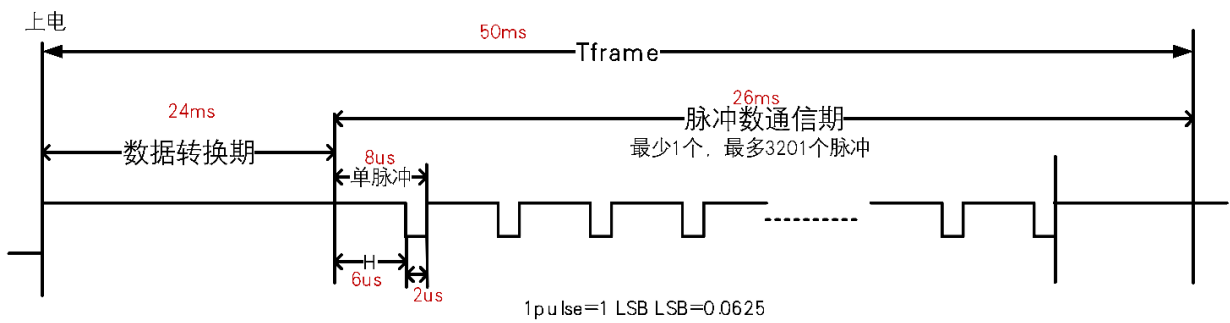


图2.6 NST1001一次转换和传输的时序图

NST1001周期地进行温度转换和温度数据发送，为方便部分客户使用单次温度转换模式，NST1001第二次温度转换后无数据输出，在第三次及以后数据传输恢复正常。

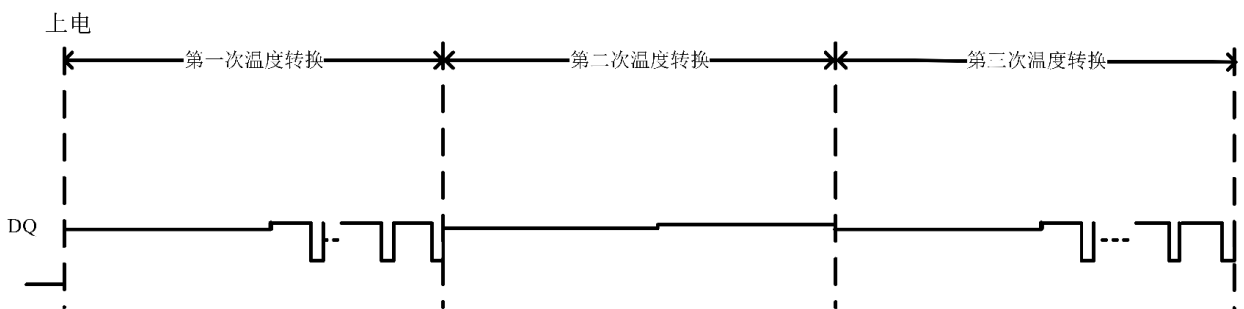


图2.7 NST1001多次温度转换和传输的时序图

温度传感器NST1001应用指南

3. 温度计算

温度计算方程如下所示：

$$Temp = Num * 0.0625^{\circ}C - 50.0625^{\circ}C$$

其中，Temp是温度值（-50°C~150°C），Num是脉冲数（1~3201个）。

表3-1 温度与脉冲数的对应关系表

脉冲数	温度
1	-50
161	-40
481	-20
801	0
1281	30
1601	50
2401	100
3201	150

温度传感器NST1001应用指南

4. 评估套件

4.1. 评估板硬件

4.1.1. 评估板原理图

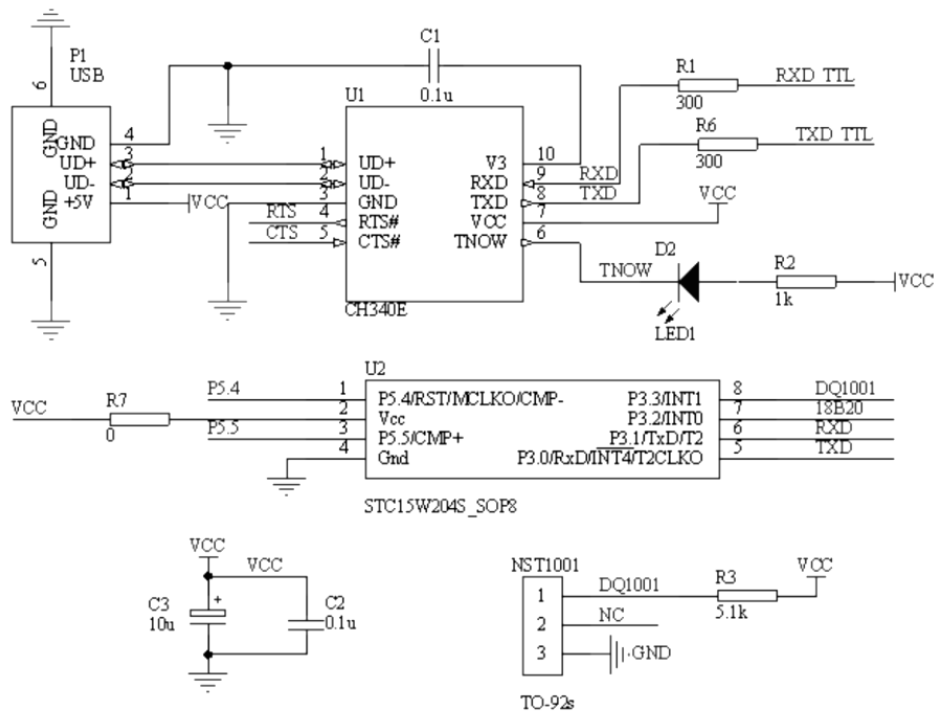


图4.1 NST1001评估板原理图

4.1.2. 评估板PCB

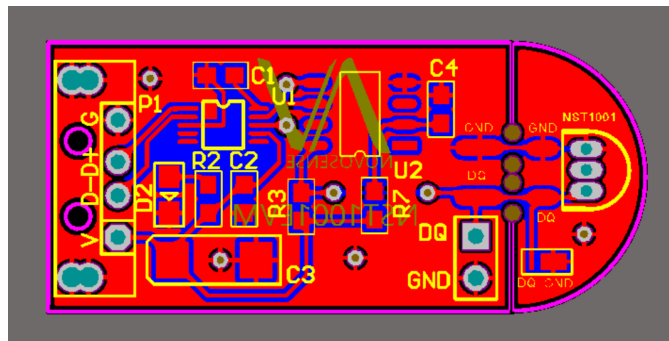


图4.2 NST1001评估板PCB图

温度传感器NST1001应用指南

4.1.3. 评估板实物

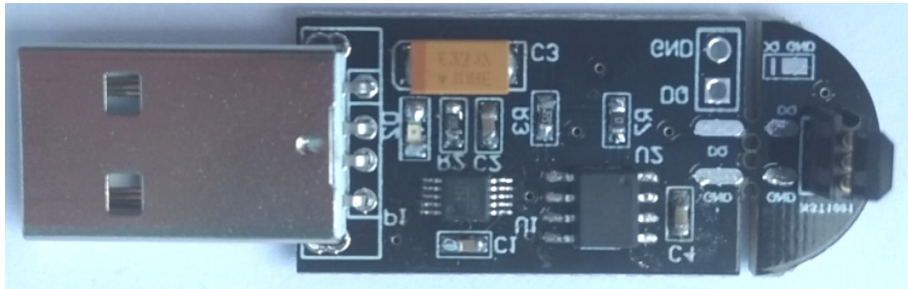


图4.3 NST1001评估板实物图

4.2. 上位机

4.2.1. 上位机介绍

上位机的文件夹包含下列文件：

- CH341SER.EXE
- NST1001评估板显示软件.exe
- SerialPoxy.dll
- ZedGraph.dll
- 软件使用说明.txt

4.2.2. 串口驱动安装

打开上位机文件夹，直接运行CH341SER.EXE可执行文件。弹出对话框如图4.4所示：

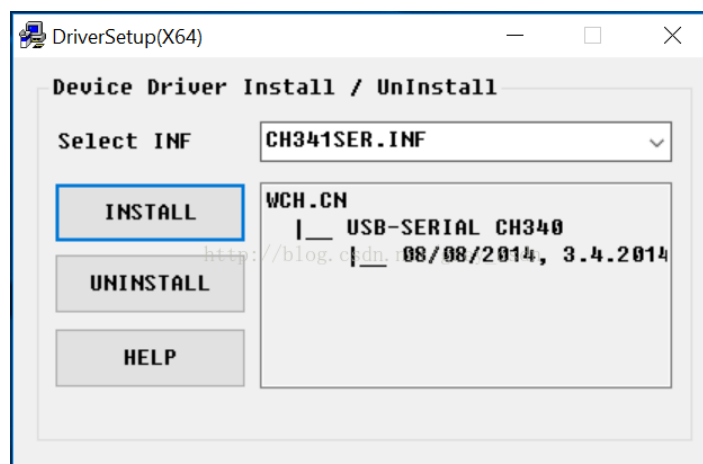


图4.4 CH341SER驱动安装

温度传感器NST1001应用指南

点击“INSTALL”，软件会自动安装好程序，安装结束后提示如图4.5所示：

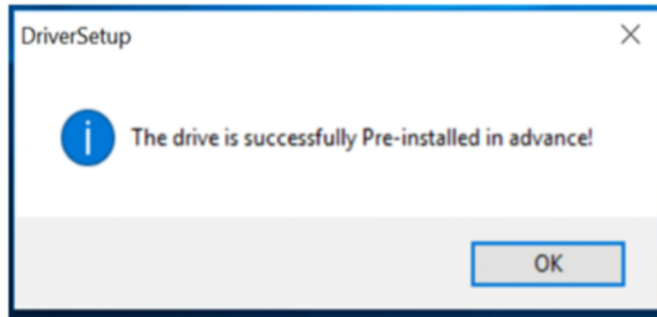


图4.5 CH341SER驱动安装成功

如图4.6所示，选择此电脑，右键单击，选择属性，进入设备管理器，选择端口（COM和LPT），可以查看到当前评估板所有COM口，如本机USB-SERIAL CH340 (COM25)，根据用户机显示选择正确的COM口。

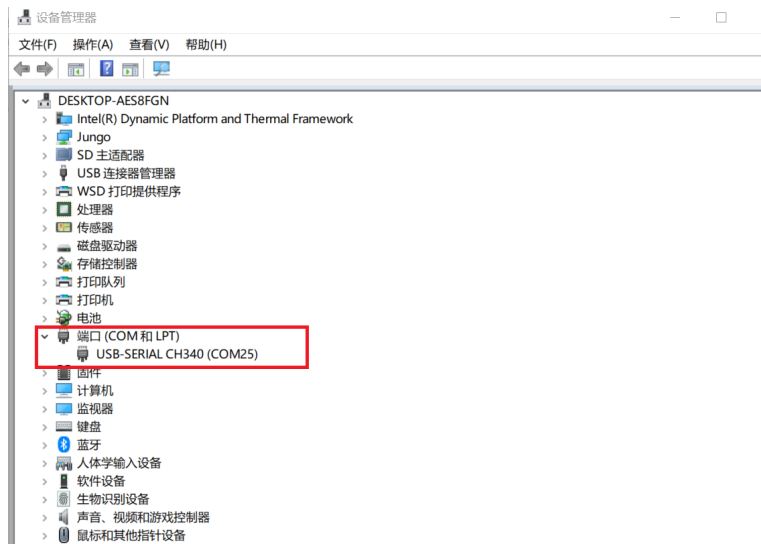


图4.6设备管理器查看评估板COM口

温度传感器NST1001应用指南

4.2.3. 评估板上位机软件使用

电脑USB口插入NST1001评估板硬件，打开上位机文件夹，运行可执行文件“NST1001评估板显示软件.exe”，如图4.7所示，在端口号下拉框选择“COM25”（此处需要根据用户电脑的具体COM口进行选择），波特率选择“9600”。



图4.7 NST1001上位机

点击“开始接收”，接收温度显示如图4.8所示。

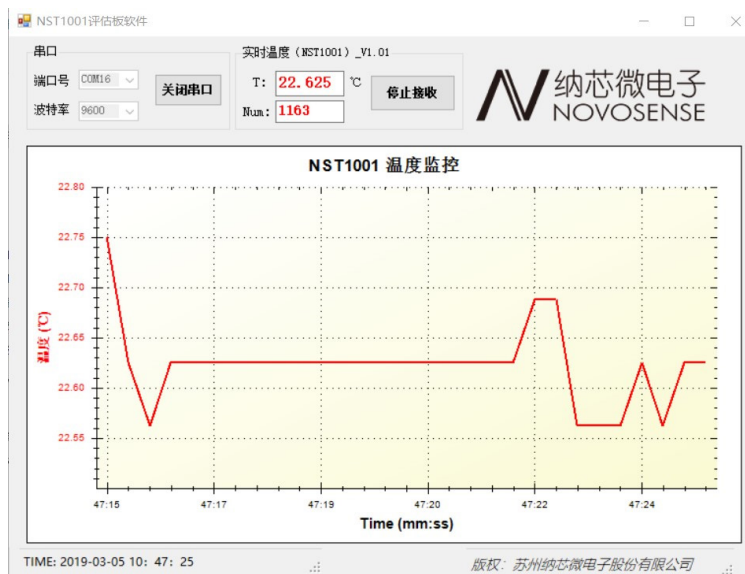


图4.8 NST1001上位机温度变化显示

温度传感器NST1001应用指南

5.参考例程

5.1.外部中断

辅助程序 1，定义用到的变量

```
unsigned int Count = 0;           //定义计数值变量COUNT
bit Flag = 0;                   //定义中断服务标志位C_FLAG;
sbit DQ = P3^3;                 //定义IO引脚
```

辅助程序 2，中断子函数

```
void exint1() interrupt 2
{
    _nop_();
    _nop_();
    if (DQ == 0)                 //误触发判断
    {
        Count++;                //计数+1
        Flag = 1;               //中断标志Flag
    }
}
```

辅助程序 3，延时子函数

```
void Delay2ms()
{
    for( i=0; i<xxxx; i++) {;} //时钟周期延时10ms，根据MCU指令周期设计具体的xxxx值
}
```

主程序

```
void main(void)                 //主程序 main
{
    unsigned int Data = 0;
    unsigned float Temp = 0;
    INT1 = 1;                    // 中断IO口初始状态为High
    IT1 = 1;                     // 外中断信号方式控制位，1：下降沿中断。单脉冲周期触发一次
    EX1 = 1;                     // 外部中断1允许控制位
    EA = 1;                      // 中断允许总控制位
    Count = 0;                   //计数值清零
```

温度传感器NST1001应用指南

```
while (1)
{
    if( Flag == 1)
    {
        Flag = 0;
        Delay2ms();
        if(!Flag)
        {
            EX1 = 0;           //关闭外部中断
            Data = Count;     //锁存计数值
            Count = 0;
            if(Data >= 801)   //正负温度判断
            {
                Temp = (Data - 801)*0.0625;
            } //正温度
            else
            {
                Temp = (801 - Data)*0.0625;
            } //负温度
            Flag = 0;
            EX1=1;           //开启外部中断
        } //END if(!Flag)
    } //END if(Flag == 1)
} //END while 循环
}
```

温度传感器NST1001应用指南

5.2.外部中断计数，定时器中断读取温度值

辅助程序 1，定义用到的变量

```
#define Temp_IN 2
long count = 0, DATA = 0;
char flag = 0;
long TEMP = 0;
```

辅助程序 2，中断子函数

```
void blink() //外部中断函数blink()
{
    if(flag == 1) //第一次外部中断
    {
        flag = 0; //清除第一次外部中断标志
        count++;
        FlexiTimer2::start(); //开启定时器
    }
    else
    {
        count++; //非第一次外部中断，均执行此行代码
    }
}

void event() //定时器中断函数
{
    pinMode(Temp_IN, INPUT); //关闭IO，NST1001 Closed
    FlexiTimer2::stop(); //关闭定时器中断
    DATA = count; //存储count 计数数值
    count = 0; //清零

    if(DATA >= 801) //正负温度判断
    {
        TEMP = (DATA - 801)*625;
    } //正温度
    else
    {
```


温度传感器NST1001应用指南

```
TEMP = (801 -DATA)*625;
} //负温度
}
辅助程序 3, 外部中断和定时器配置
void setup() //外部中断和定时器配置
{
    FlexiTimer2::set(35,1.0/1000,event); //定时器配置, 35mS中断一次
    attachInterrupt (0, blink, FALLING); //外部中断, 调用blink函数,下降沿触发
}
主程序
void loop()
{
    flag = 1; //标志置位
    pinMode( Temp_IN, INPUT_PULLUP); //IO PULLUP, NST1001上电开启
}

void main(void)
{
    setup();
    //需要采集温度时, 调用loop();函数, 50ms后, TEMP变量即为实测温度数值
    loop();
    /*
        用户其他执行代码
    */
}
```

温度传感器NST1001应用指南

修订历史

版本	描述	作者	日期
1.0	创建	Yuyun Yang	2023/08/24

销售联系方式: sales@novosns.com; 获取更多信息: www.novosns.com

重要声明

本文件中提供的信息不作为任何明示或暗示的担保或授权,包括但不限于对信息准确性、完整性,产品适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的陈述或保证。

客户应对其使用纳芯微的产品和应用自行负责,并确保应用的安全性。客户认可并同意:尽管任何应用的相关信息或支持仍可能由纳芯微提供,但将在产品及其产品应用中遵守纳芯微产品相关的有关法律、法规和相关要求。

本文件中提供的资源仅供经过技术培训的开发人员使用。纳芯微保留对所提供的产品和服务进行更正、修改、增强、改进或其他更改的权利。纳芯微仅授权客户将此资源用于开发所设计的整合了纳芯微产品的相关应用,不视为纳芯微以明示或暗示的方式授予任何知识产权许可。严禁为任何其他用途使用此资源,或对此资源进行未经授权的复制或展示。如因使用此资源而产生任何索赔、损害、成本、损失和债务等,纳芯微对此不承担任何责任。

有关应用、产品、技术的进一步信息,请与纳芯微电子联系(www.novosns.com)。

苏州纳芯微电子股份有限公司版权所有