Application Guide

NOVOSENSE

Temperature Sensor NST1002

AN-12-0041

Author: Jincheng Liu

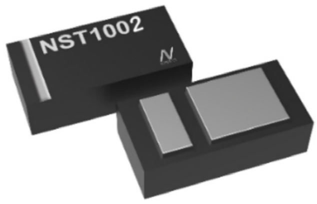
AN-12-0041

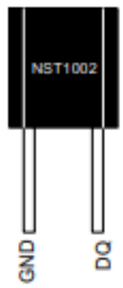
Temperature Sensor NST1002

Description

The NST1002 is a NOVOSENSE D-NTC® series digital temperature sensor compatible with One-Wire interfaces which makes it possible to directly connect to the GPIO of the MCU and save MCU resources to the greatest extent.

The NST1002 has a high accuracy and high resolution over temperature range of -50°C to 150°C. The on-chip 15-bit ADC offers resolutions down to 0.0078125°C. The devices offer a maximum accuracy of ±0.1°C from 0°C to 45°C without requiring calibration and are highly linear and do not require complex calculations or look-up tables. NST1002 suits automotive, industrial, home appliances and other applications for temperature monitoring, which can be easily used as a two wire digital temperature probe or as a direct replacement for NTC thermistors. NST1002 can also be used in wireless IoT sensor nodes with particularly stringent power requirements because of its extremely low operating current, which can be powered through the MCU's GPIO. The NST1002 is available in an DFN-2L.





Features

- Higher Resolution, 0.0078125°C (1 LSB)
- Date Conversion and Transmission: 32 ms/week
- Power Supply Operating Range: 1.7V to 5.5V
- Physical pin-to-pin Replacement of NTCs
- Conversion Current: 30uA (typical)
- Idle Current: 5.4μA (typical)
- Package Format
  DFN-2L (1.6mm × 0.8mm）
  TO-92S-2L(4.0mm x 3.0mm)

Applications

- Digital Output Wired Probes
- General System Thermal Management
- Computer Peripheral Thermal Protection
- Notebook Computers
- Industrial Internet of Things (IoT)
- Communications Infrastructure
- Power-system Monitors
- Thermal Protection
- Environmental Monitoring and HVAC
- Medical Devices

AN-12-0041

Temperature Sensor NST1002

NOVOSENSE

INDEX

1.Pin Configuration and Functions

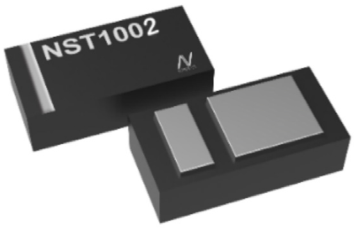


Figure 1.1 NST1002 Package Diagram (DFN-2L)

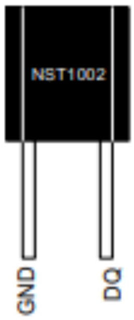


Figure 1.2 NST1002 Package Diagram (TO-92S-2L)

Pin Name	I/O	Descriptions
DQ	I/O	Supply and digital IO
GND	GND	GND (widely pad)

Table 1.1 NST1002 Pin Descriptions

2.Typical Application Circuits

The NST1002 can work in the parasitic power mode. When the bus is at a high level, power is supplied through One-Wire pull-up resistor. The high bus signal also charges the internal capacitors and then supplies power to the device when the bus is low. It should be noted that when the device is in the idle state, in order to read the temperature normally, the host must pull up the DQ pin and wait at least 30ms before initialization.

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

2.1.Single GPIO Application

As shown in Figure 2.1, the pinout DQ connect to GPIO and also connect to VDD with pull up resistor Rpu. The output pulse of the device can be read with a GPIO. There is only 1 GPIO needed in this application, saving the GPIO resource in the system. NST1002HA will pull down if DQ pull down the GPIO more than 5ms.

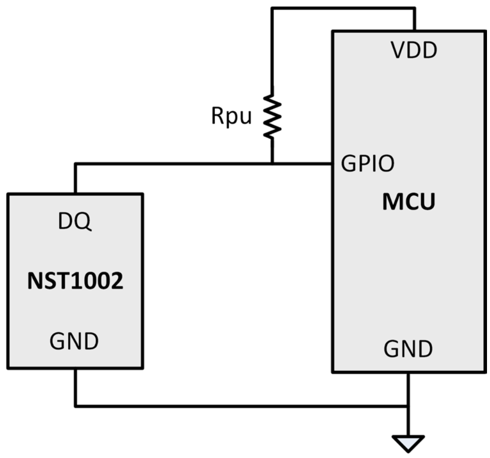


Figure 2.1 Single GPIO Application

Design parameter	Value
R_{pu}	1 KΩ~10KΩ
VDD	1.7V~5.5V
Microcontroller	General I/O

Table 2.1 Design Parameter

Note: the NST1002HA max Conversion current is 30μA (typical), and the min Operation voltage will be effected by pull up resistor Rpu. For example, the min Operation voltage is 1.7V while the R_{pu} =5KΩ.

2.2.No Power Consumption in Standby Mode Application

There are 2 GPIO needed in this application in order to achieve the no power consumption in standby mode. As shown in Figure 2.2, the DQ pin connected to GPIO2 and connects to GPIO1 with pull up resistor Rpu. The GPIO1 will set to high, and provide the power though the pull up resistor Rpu as VDD. The GPIO2 as One-Wire communication pin to get temperature data. If the temperature is calculated, and pulls down the GPIO1, there is no power consumption in standby mode.

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

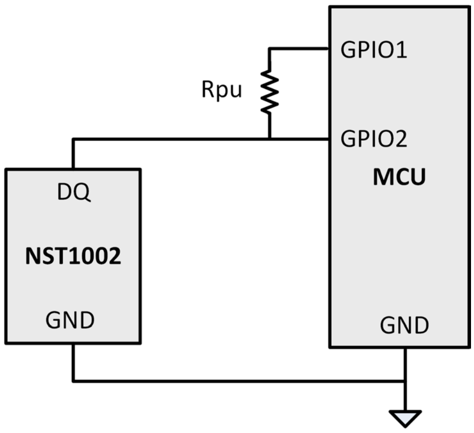


Figure 2.2 No Power Consumption in Standby Mode Application

2.3.Multi-point Temperature Acquisition

As shown in Figure2.3, all NST1002 nodes in this scheme share GPIO0 as the DQ count port and share the same pull-up resistor. The temperature node to be acquired is enabled by pulling one of GPIO1~GPIOn low, and the GPIO corresponding to the other unused nodes is set to high resistance state. Note that more than two of GPIO1~GPIOn cannot be pulled low at the same time.

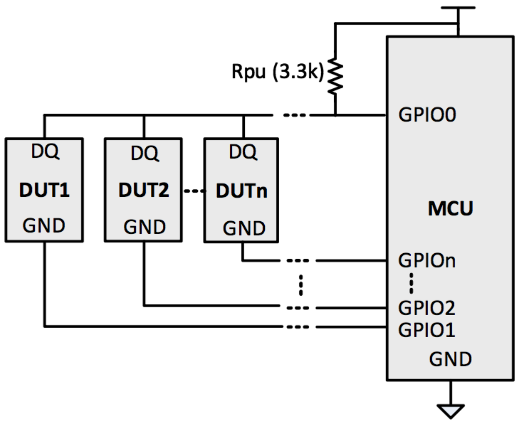


Figure 2.3 Multi-point Temperature Acquisition with NST1002

3.NST1002 Timing

Initialization:

The host (as MCU) sends a low pulse that lasts at least 200μs to initialize the device. After initialization, the device is ready for start temperature conversion. If the initialization is incorrect, such as the low pulse time is too short, the device will not perform temperature conversion and will remain idle state. It should be noted that when the device is in the idle state, in order to read the temperature normally, the host must pull up the DQ pin and wait at least 30ms before initialization.

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

Temperature conversion:

After the successful initialization of the device, it will enter in the temperature conversion stage. This process needs to ensure that the DQ pin is at a high level and lasts for 30ms (typical), and it will not be interrupted until finishing temperature conversion.

Temperature Done Pulse:

After the device completes the temperature conversion, it will send a low pulse lasting 17us (typical), this means that the temperature conversion is successful, meanwhile the host should configure the DQ pin as the input mode to read the Temperature Done Pulse, then prepare to read the data.

Read Out Data(Temperature data and CRC check data):

After completing the above series of operations, the host can read data bit by bit (including temperature data and CRC data). It should be noted that single bit data time(TL) and single bit period time (TB) during reading and pulling down DQ before reading each bit of data, and after data reading is completed, the device enters in the idle state. The detailed data format will be described in Section Evaluation kit .

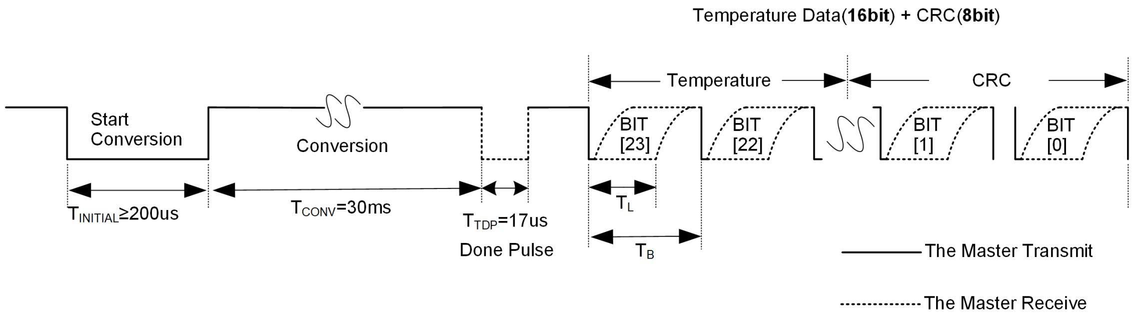


Figure 3.1 NST1002 Timing

4.Digital Temperature Data

The resolution of NST1002 is 15-bit, and 1LSB corresponding to 0.0078125°C. After the conversation, the Temperature data can be read from DQ pin with a total 24-bits data (16bits temperature data + 8bits CRC check data), and MSB firstly. The digital output from each temperature measurement conversion is stored in the Temperature register as 15-bit sign-extend complement format. The sign bit(S) indicate if the temperature is positive or negative: for positive number S=0 and for negative number S=1. Data format of temperature is listed in Table 4.1 and Table 4.2. Negative numbers are represented in binary complement format.

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

BIT	BIT23	BIT22	BIT21	BIT20	BIT19	BIT18	BIT17	BIT16	BIT15
Define	S	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Description	Sign	128	64	32	16	8	4	2	1
BIT	**BIT14**	**BIT13**	**BIT12**	**BIT11**	**BIT10**	**BIT9**	**BIT8**	**BIT7**	**BIT6**
Define	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	CC7	CC6
Description	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.0078125	CRC check	CRC check
BIT	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0**			
Define	CC5	CC4	CC3	CC2	CC1	CC0			
Description	CRC check	CRC check	CRC check	CRC check	CRC check	CRC check			

Note: The BIT0 to BIT7 is the CRC check bit.

Table 4.1. Temperature Data Format Description

The error detection scheme most effective at locating errors in a serial-data stream with a minimal amount of hardware is the CRC. The NST1002 uses the standard CRC model that size is 8bit, which is used to check the correctness of each bit of temperature data and the specific polynomial is shown in Equation 4-1:

$$CRC=X^8+X^5+X^4+1 \qquad (4\text{-}1)$$

Note： The original data and calculated data needs to be flipped. Table 4.2 Temperature Data Format (excluding CRC check bit)

Temperature	Digital Output	
(℃)	**Binary**	**HEX**
150.9921875	0100 1011 0111 1111	4B7F
127.9921875	0011 1111 1111 1111	3FFF
100	0011 0010 0000 0000	3200
25	0000 1100 1000 0000	0C80
0	0000 0000 0000 0000	0000
-0.1953125	1111 1111 1110 0111	FFE7
-25	1111 0011 1000 0000	F380
-50	1110 0111 0000 0000	E700

Table 4.2. Temperature Data Format (excluding CRC check bits)

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

5.Evaluation Kit

5.1.Evaluation Board Schematic Diagram

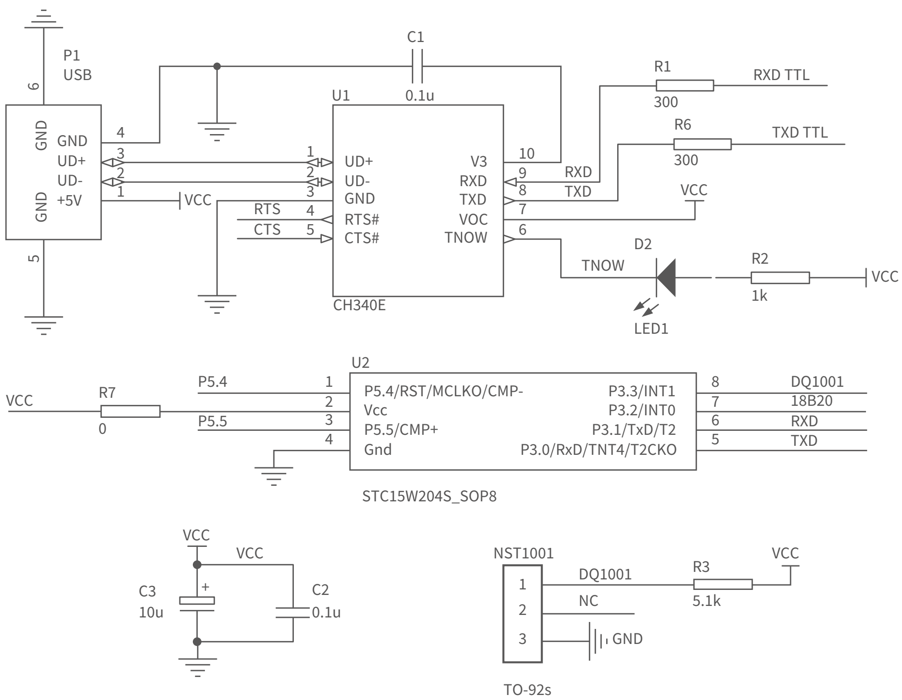


Figure 5.1 Schematic diagram of the NST1002 evaluation board

5.2.Evaluation Board PCBs

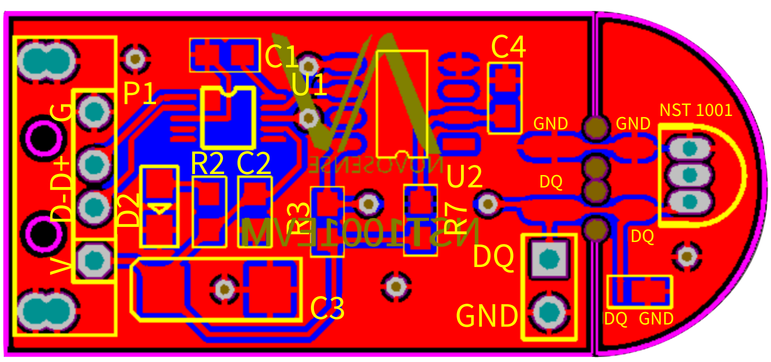


Figure 5.2 PCB Diagram of The NST1002 Evaluation Board

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

5.3.Evaluation Board Physical

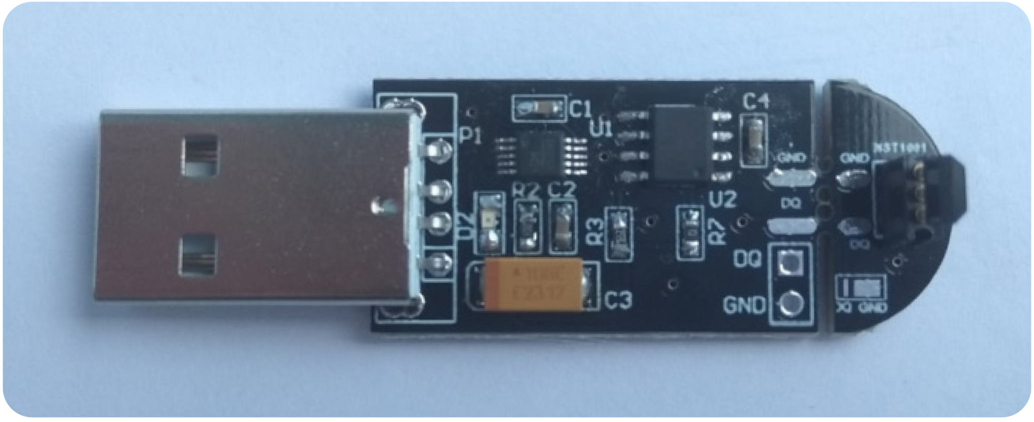


Figure 5.3 Physical Diagram of The NST1002 Evaluation Board

6.Upper Computer

6.1.Introduction to the Upper Computer

The folder in the OP contains the following files:

- CH341SER.EXE
- NST1002 Evaluation Board Display Software.exeSerialPoxy.dll
- ZedGraph.dll
- Software instructions.txt

6.2.Serial Driver Installation

Open the upper folder and run the CH341SER.EXE executable file directly. A pop-up dialog box is shown in Figure 6.1 below:

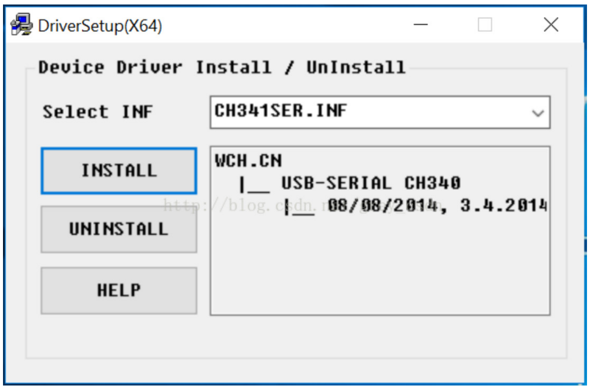


Figure 6.1 CH341SER Driver Installation

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

Click “INSTALL”, the software will automatically install the program, the end of the installation prompt as shown in Figure 6.2

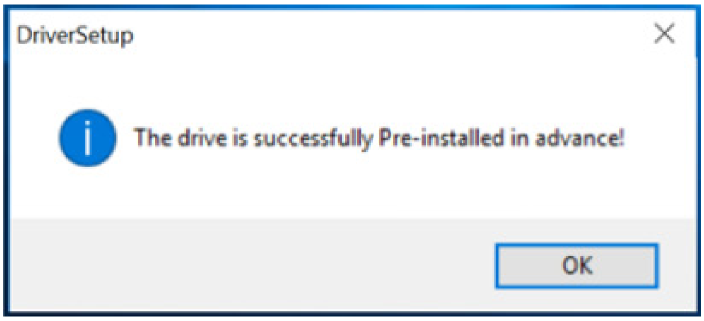


Figure 6.2 CH341SER Driver Installation Successful

As shown in Figure6.3, select this computer, right-click, select Properties, enter Device Manager, select Ports (COM and LPT), you can view all the COM ports of the current evaluation board, such as the local USB-SERIAL CH340 (COM25), and select the correct COM port according to the display of the user's machine.

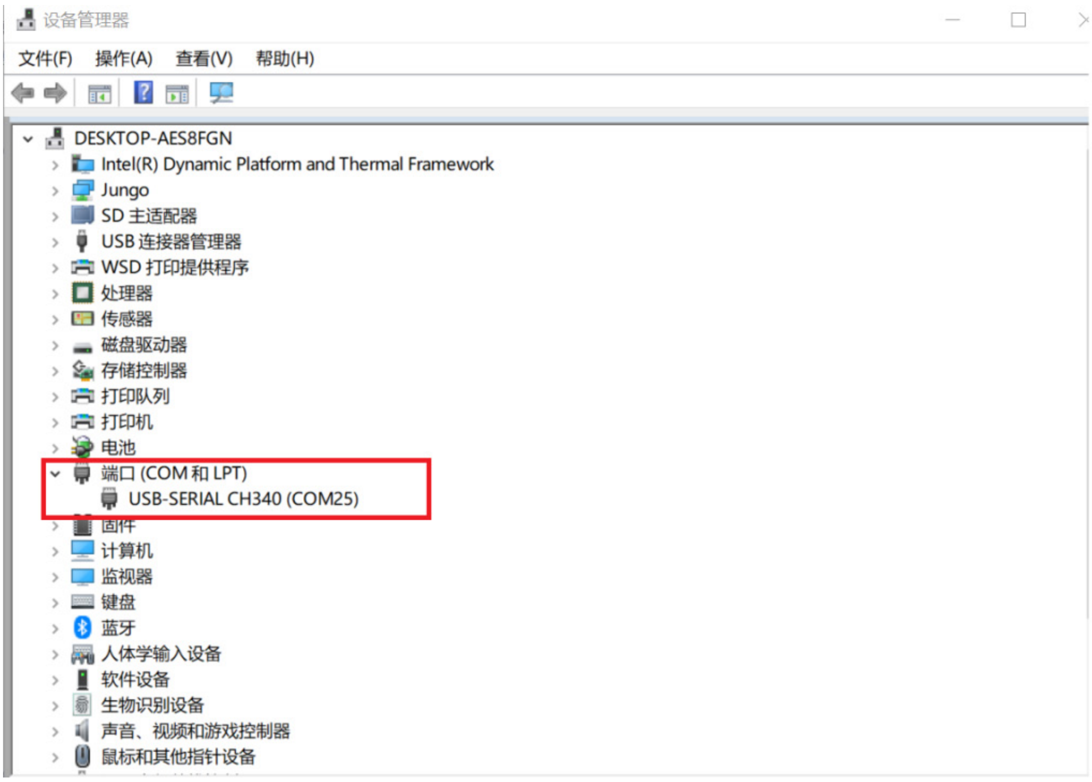


Figure 6.3 Device Manager Viewing Evaluation Board COM Ports

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

6.3.Evaluation Board Upper Computer Software Use

Insert the NST1002 evaluation board hardware into the USB port of the computer, open the host computer folder, run the executable file “NST1002 Evaluation Board Display Software.exe”, as shown in Figure 6.4 below, select “COM3” in the port number drop-down box (here you need to select according to the specific COM port of the user's computer), and select “9600” for the baud rate. (Here you need to select the COM port according to the specific COM port of the user's computer), and select “9600” as the baud rate.

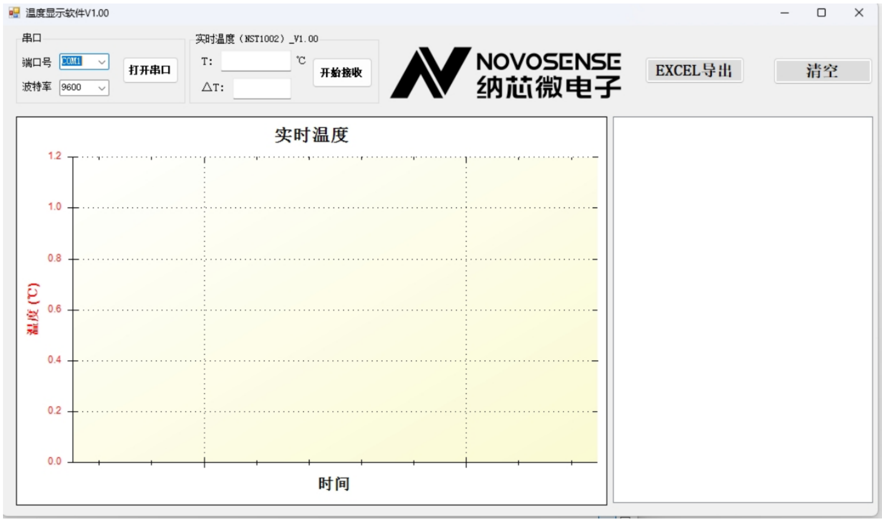


Figure 6.4 NST1002 Upper Unit

Click “Start Receiving” and the receiving temperature is displayed as shown in Figure 6.5 below.

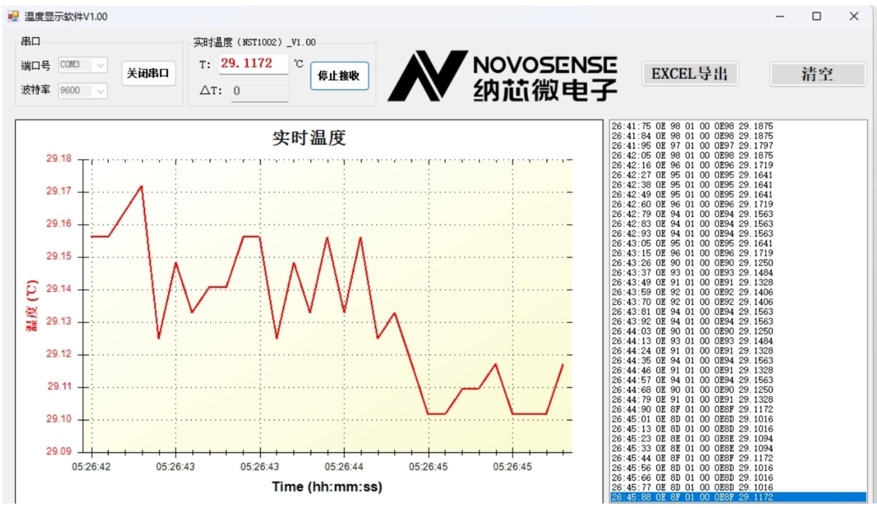


Figure 6.5 Temperature Change Display of NST1002 Upper Unit

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

NOVOSENSE

7.Reference Routines

```
#include <stdlib.h>
#define ERR_NC      -100
#define ERR_DONE    -101
#define ERR_DAT     -102
#define ERR_CRC     -103
//-----------------by usr  ----------------------
#define pin_write(x,y)  rt_pin_write(x,y)
#define pin_mode(x,y)   rt_pin_mode(x,y)
#define pin_read(x)     rt_pin_read(x)
#define DQ_PIN                  GET_PIN(A, 1)
#define delay_ms(x)             rt_thread_mdelay(x)
#define delay_us(x)             rt_hw_us_delay(x)
//---------------------------------------
static uint8_t reversal(uint8_t data)
{
  uint8_t _data = data;
  uint8_t i, retValue = 1;
  for (i = 0; i < 8; i++)
  {
    unsigned char temp = _data & 0x01;
    if (temp)
      retValue |= 0x01;
    else
      retValue &= ~0x01;
    if (i == 7)
      break;
    retValue <<= 1;
    _data >>= 1;
  }
  return retValue;
}
// crc8_maxim  input and output reversal
//x8+x5+x4+1    0x131
static uint8_t crc8_maxim(uint8_t *pdat, uint8_t len)
{
  uint8_t ret;
  uint8_t uCRC = 0x00; //CRC
  for (uint8_t num = 0; num < len; num++)
  {
```

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

```
        ret = reversal(*pdat++);
        uCRC = (ret) ^ uCRC; //
        for (uint8_t x = 0; x < 8; x++)
        {
            if (uCRC & 0x80)
            {
                uCRC = uCRC << 1; //
                uCRC = uCRC ^ 0x31; //
            }
            else   //
            {
                uCRC = uCRC << 1; //
            }
        }
    }
    ret = reversal(uCRC);
    return ret;
}
int nst1002_read_cal(uint16_t *pcal)
{
    int  retry = 0;
    __IO uint8_t crc;
    uint8_t i,j;
    uint8_t bit0 = 1;
    uint8_t data[3] = { 0, 0, 0 };
    uint8_t swap[2];
    uint8_t read[4];
//--------------power up------------------------
    pin_mode(DQ_PIN, PIN_MODE_OUTPUT_OD);
    pin_write(DQ_PIN, PIN_HIGH);
    delay_ms(15);
//--------------send start convert ------------------------------
    pin_write(DQ_PIN, PIN_LOW);        //DQ low
    delay_us(300);
    pin_write(DQ_PIN, PIN_HIGH);
                delay_us(10);
//-----------------check DQ pin status  should be high--------------------------------------
    pin_mode(DQ_PIN, PIN_MODE_INPUT);
    delay_us(100);
    if (pin_read(DQ_PIN) == 0)
    {
         *pcal = (uint16_t)(ERR_NC*128);
```

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

```
        return-1;
    }
    delay_ms(20);
//--------------wait for done pulse---------------
    do
    {
        bit0 = pin_read(DQ_PIN);
        delay_us(2);
        retry++;
        if (retry > 20000)// time out 40ms
        {
            *pcal = (uint16_t)(ERR_DONE*128);
            return -2;
        }
    } while (bit0);
    delay_us(10);
//--------------read 24bit data---------------
    pin_write(DQ_PIN, PIN_HIGH);
    pin_mode(DQ_PIN, PIN_MODE_OUTPUT_OD);
    delay_us(200);
    for (i = 0; i < 24; i++)
    {
        pin_write(DQ_PIN, PIN_LOW);        //Falling edge
        delay_us(1);                                        // Adjust according to the mcu clock cycle,It's not
necessary
                pin_mode(DQ_PIN, PIN_MODE_INPUT);
                // If the output bit is 0, the DQ will remain low for about 14us
                for(j=0;j<4;j++)              // It's not necessary to read it 4 timers , adjust according to the mcu
clock cycle
                {
                        read[j]= pin_read(DQ_PIN);
                }
                if(read[0]==read[1]) // It's not necessary
                    bit0 = read[1];
                else
                {
                    *pcal = (uint16_t)(ERR_DAT*128);
                    return -3;
                }
        data[i / 8] <<= 1;
        data[i / 8] |= bit0;
        pin_write(DQ_PIN, PIN_HIGH);        //DQ low
```

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

```
      pin_mode(DQ_PIN, PIN_MODE_OUTPUT_OD);
      delay_us(60);
    }
//------------------- NST1002 Big Endian,but mcu Little Endian
    swap[0] = data[1];
    swap[1] = data[0];
    *pcal = *((uint16_t *) swap);// output data  need swap
    crc = crc8_maxim(data, 2);
    if (crc == data[2])
      return 0;
    else
    {
      printf("data %02x %02x %02x  CRC %02x\r\n",data[0],data[1],data[2],crc);
       *pcal=ERR_CRC*128;
                        return-4;
    }
}
char err_code[4][20]={"NOT Connected","No DONE signal","DAT Recv Err","CRC Err"}
void read_temp(void)
{               int err,ei;
    double dtmp;
    int16_t cal;
    err=nst1002_read_cal((uint16_t *)&cal);
                if(err<0)
                {
                        ei=-1-err;
                 if(ei<4)
                   printf("NST1002 err %s\r\n",err_code[ei]);
                 return;
                }
  dtmp = cal / 128.0;
  printf("Temp %3.7f\r\n",dtmp);}
```

Copyright © NOVOSENSE Microelectronics Co.,Ltd 2023. All rights reserved.

8.Version Information

Revision	Description	Author	Date
V1.0	Initial Version	Jincheng Liu	2024/12/26

Sales Contact: sales@novosns.com; Further Information: www.novosns.com

IMPORTANT NOTICE

The information given in this document (the "Document") shall in no event be regarded as any warranty or authorization of, express or implied, including but not limited to accuracy, completeness, merchantability, fitness for a particular purpose or infringement of any third party's intellectual property rights.

Users of this Document shall be solely responsible for the use of NOVOSENSE's products and applications, and for the safety thereof. Users shall comply with all laws, regulations and requirements related to NOVOSENSE's products and applications, although information or support related to any application may still be provided by NOVOSENSE.

This Document is provided on an "AS IS" basis, and is intended only for skilled developers designing with NOVOSENSE' products. NOVOSENSE reserves the rights to make corrections, modifications, enhancements, improvements or other changes to the products and services provided without notice. NOVOSENSE authorizes users to use this Document exclusively for the development of relevant applications or systems designed to integrate NOVOSENSE's products. No license to any intellectual property rights of NOVOSENSE is granted by implication or otherwise. Using this Document for any other purpose, or any unauthorized reproduction or display of this Document is strictly prohibited. In no event shall NOVOSENSE be liable for any claims, damages, costs, losses or liabilities arising out of or in connection with this Document or the use of this Document.

For further information on applications, products and technologies, please contact NOVOSENSE (www.novosns.com).

Suzhou NOVOSENSE Microelectronics Co., Ltd