

7.温湿度传感器 DHT11

实验内容：

1. 掌握温湿度传感器使用
2. 掌握点对点通讯
3. 掌握 DHT11 移植方法

实现现象：

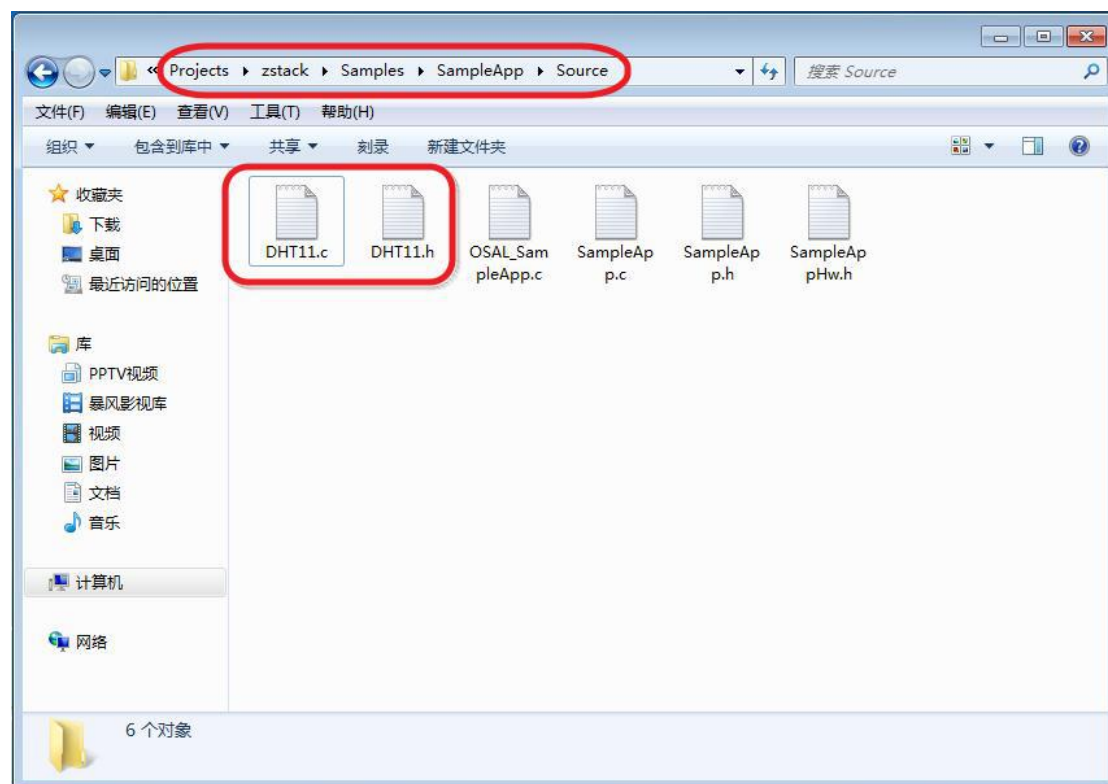
终端获得 DH T11 传感器的数据，无线传输给协调器；协调器再 通过串口发给电脑串口调试助手显示。协调 器、终端通过串口输出，LCD 也同步刷新。

实验详解：

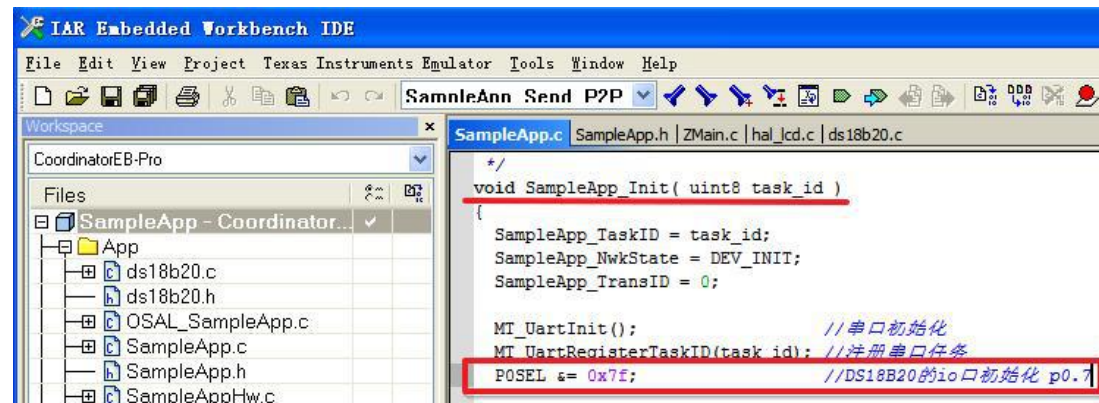
由于此实验和 DS18B20 共用一个 IO 口，所以移植起来更加容易。DHT11 带温度、湿度检测，而 DS18B20 则只带温度检测，不过精度相对要高点，一般应用只会选其中一种而已。程序大部分相同，本实验中只讲不同部分。

打开\\ZStack-CC2530-2.3.0-1.4.0\Projects\zstack\Samples\SampleApp\CC2530DB\SampleApp.eww 工程。

1. 我们将基础实验里面的 DHT11.c 和 DHT11.h 文件复制到 SampleApp\Source 文件夹下。



2. 在协议栈的 APP 目录树下点击右键--Add--添加 DHT11.c 和 DHT11.h 文件。并在 SampleApp.c 文件中包含 DHT11.h 头文件。
3. 初始化传感器引脚,和 ds18b20 共用一个 GPIO



4. 读取温度数据,这个是重点, 其它基本都相同, 只要看懂此段代码即可会使用 DH11 了.

```
void SampleApp_Send_P2P_Message( void )
```

```
{
```

```
    byte i, temp[3], humidity[3], strTemp[7];
```

```
    DHT11();           //获取温湿度
```

```
    //将温湿度的转换成字符串,供 LCD 显示
```

```
    temp[0] = wendu_shi+0x30;
```

```
    temp[1] = wendu_ge+0x30;
```

```
    temp[2] = '\0';
```

```
    humidity[0] = shidu_shi+0x30;
```

```
    humidity[1] = shidu_ge+0x30;
```

```
    humidity[2] = '\0';
```

```
    //将数据整合后方便发给协调器显示
```

```
    osal_memcpy(strTemp, temp, 2);
```

```
    osal_memcpy(&strTemp[2], " ", 2);
```

```
    osal_memcpy(&strTemp[4], humidity, 3);
```

```
    //获得的温湿度通过串口输出到电脑显示
```

```
    HalUARTWrite(0, "T&H:", 4);
```

```
    HalUARTWrite(0, strTemp, 6);
```

```
    HalUARTWrite(0, "\n", 1);
```

```
    //输出到 LCD 显示
```

```
    for(i=0; i<3; i++)
```

```
    {
```

```
        if(i==0)
```

```
        {
```

```
            LCD_P16x16Ch(i*16,4,i*16);
```

```
            LCD_P16x16Ch(i*16,6,(i+3)*16);
```

```

    }
    else
    {
        LCD_P16x16Ch(i*16,4,i*16);
        LCD_P16x16Ch(i*16,6,i*16);
    }
}
LCD_P8x16Str(44, 4, temp);
LCD_P8x16Str(44, 6, humidity);

if ( AF_DataRequest( &SampleApp_P2P_DstAddr, &SampleApp_epDesc,
                    SAMPLEAPP_P2P_CLUSTERID,
                    6,
                    strTemp,
                    &SampleApp_TransID,
                    AF_DISCV_ROUTE,
                    AF_DEFAULT_RADIUS ) == afStatus_SUCCESS )
{
}
else
{
    // Error occurred in request to send.
}
}

```

5. 接收数据

```

void SampleApp_MessageMSGCB( afIncomingMSGPacket_t *pkt )
{
    uint16 flashTime;
    switch ( pkt->clusterId )
    {
        case SAMPLEAPP_P2P_CLUSTERID:
            HalUARTWrite(0, "T&H:", 4);           //提示接收到数据
            HalUARTWrite(0, pkt->cmd.Data, pkt->cmd.DataLength); //输出接收到的数据
            HalUARTWrite(0, "\n", 1);              // 回车换行

            //输出到 LCD 显示
            osal_memcpy(temp, pkt->cmd.Data, 2);
            osal_memcpy(humidity, pkt->cmd.Data+4, 2);
            for(i=0; i<3; i++)
            {
                if(i==0)
                {
                    LCD_P16x16Ch(i*16,4,i*16);

```

```

        LCD_P16x16Ch(i*16,6,(i+3)*16);
    }
    else
    {
        LCD_P16x16Ch(i*16,4,i*16);
        LCD_P16x16Ch(i*16,6,i*16);
    }
}

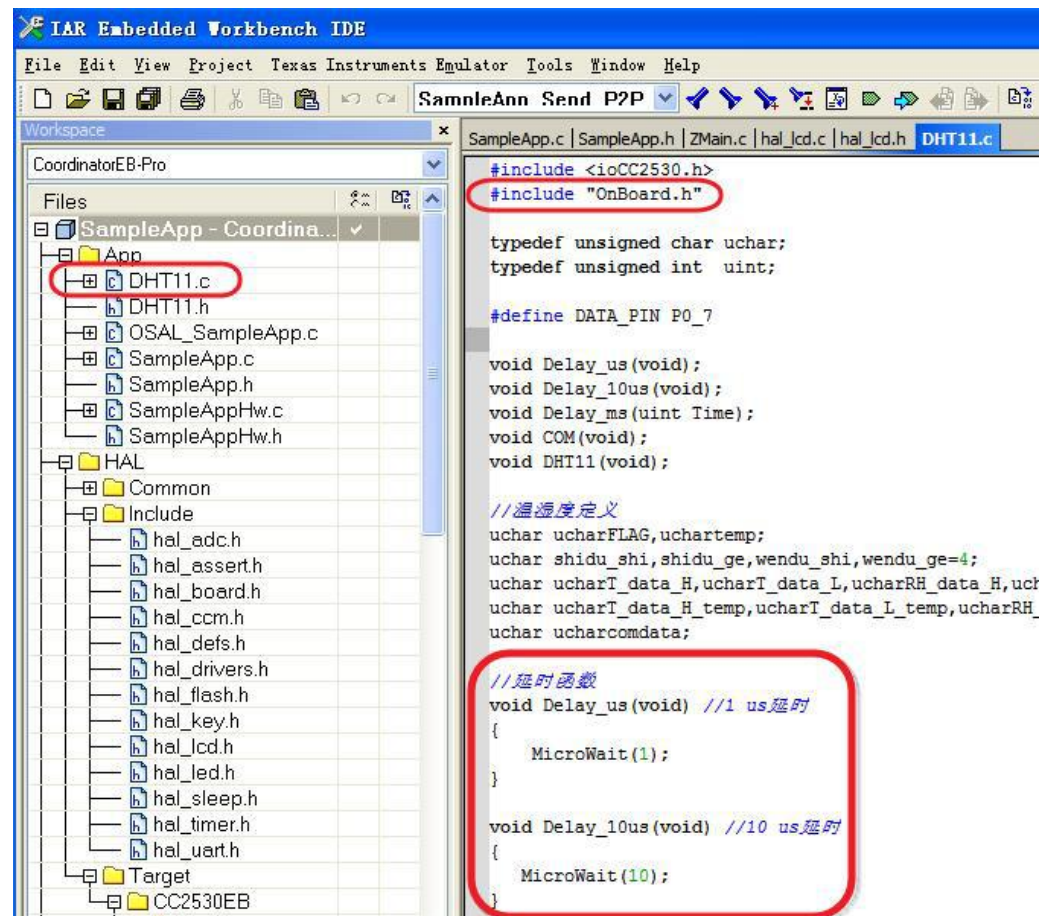
LCD_P8x16Str(44, 4, temp);
LCD_P8x16Str(44, 6, humidity);
break;

case SAMPLEAPP_PERIODIC_CLUSTERID:
    break;

case SAMPLEAPP_FLASH_CLUSTERID:
    flashTime = BUILD_UINT16(pkt->cmd.Data[1], pkt->cmd.Data[2] );
    HalLedBlink( HAL_LED_4, 4, 50, (flashTime / 4) );
    break;
}
}

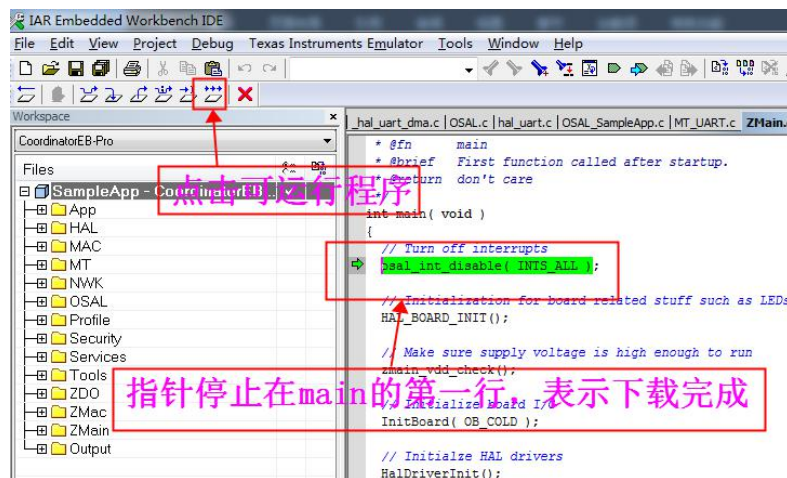
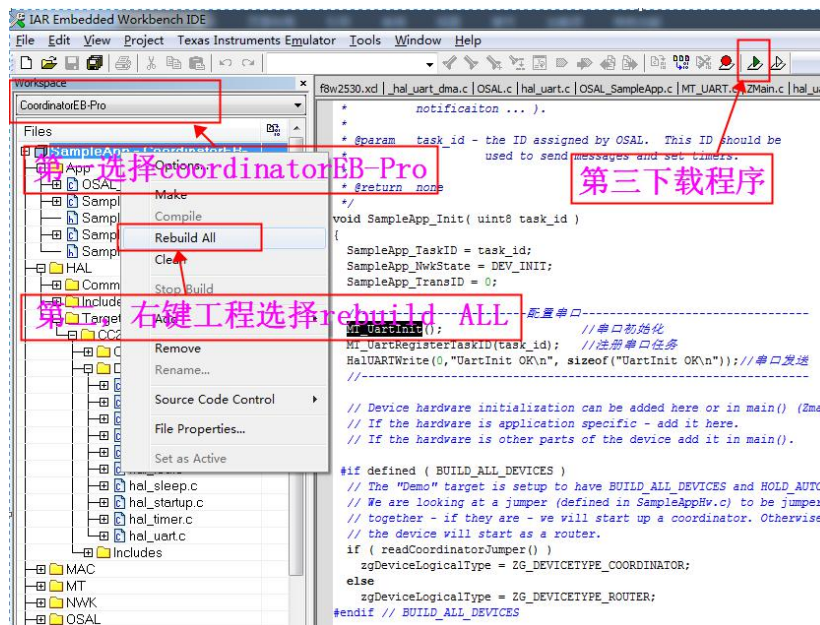
```

6. DH11.c 文件还需要修改一个地方。打开 文件将原来的延时函数改成协议栈自带的延时函数，保证时序 的正确。同时要包含 `#include "OnBoard.h"`。

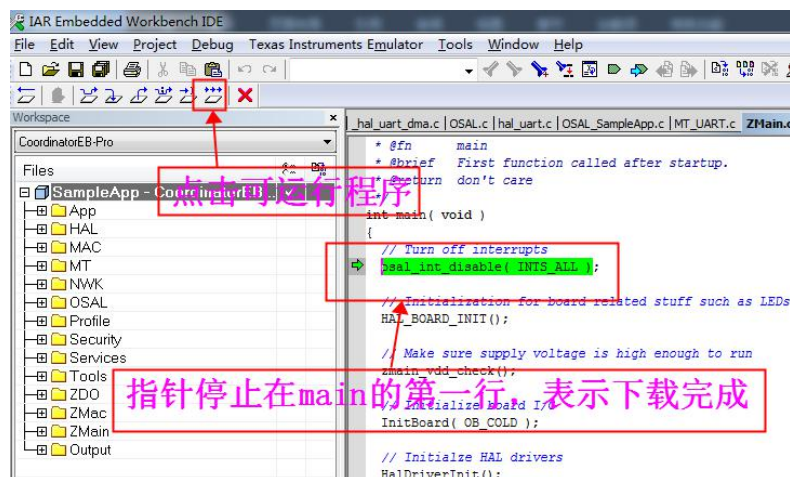
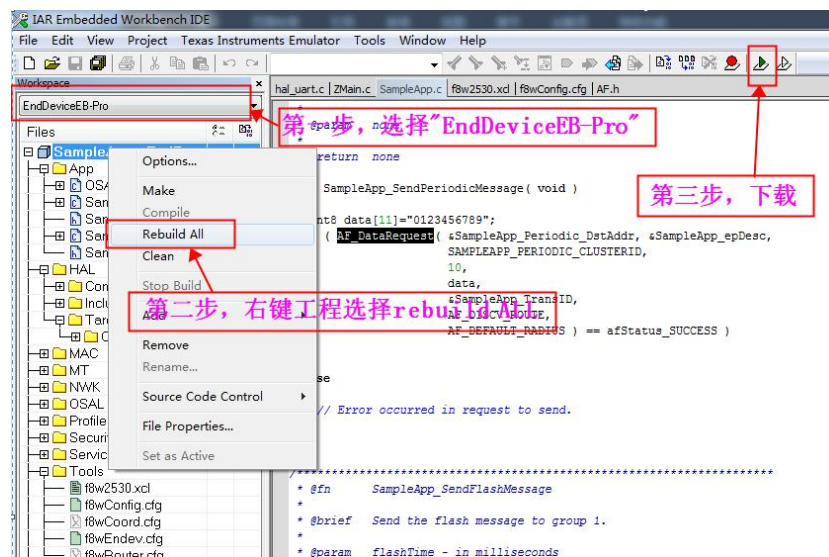


实验步骤

- 1) 选择 CoordinatorEB-Pro (作为协调器), 下载到开发板 A, 如:



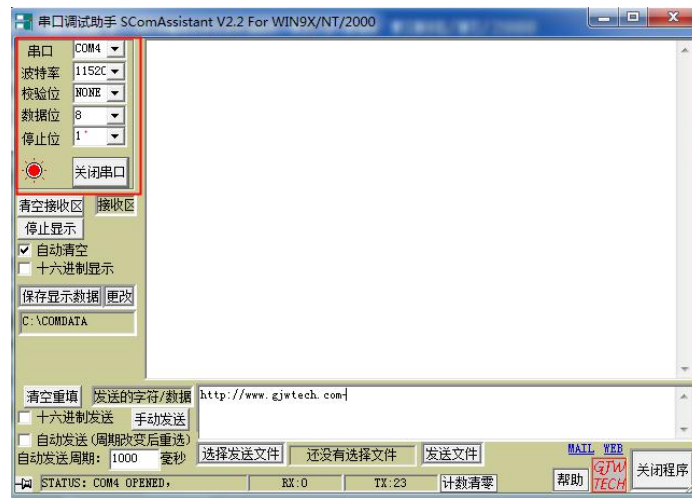
- 2) 选择 EndDeviceEB-Pro (终端), 下载到开发板 B, 如:



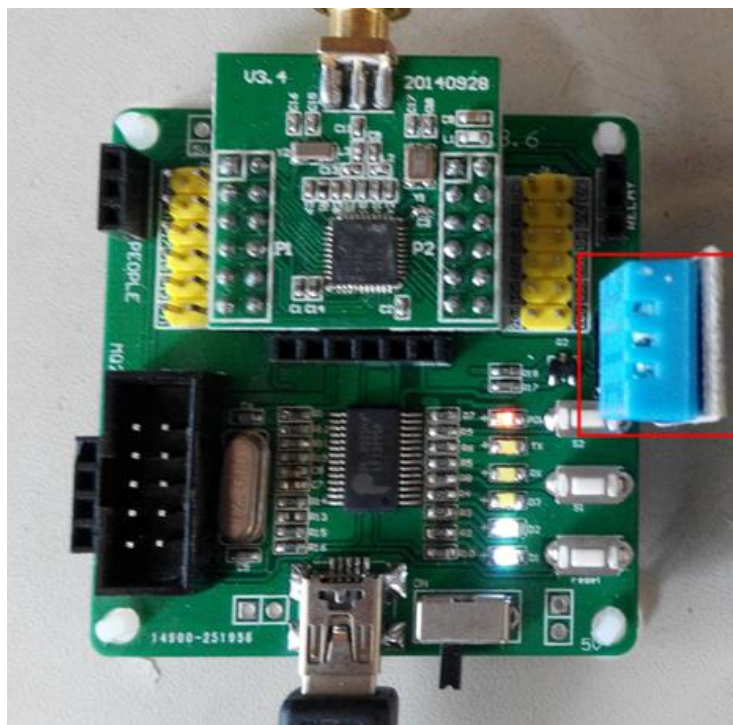
3) 开发板 A (协调器) 接上 LCD, 并上电, 查看生成的串口, 方法:



使用串口工具打开生成的串口 com4:



- 4) 给开发板 B（终端）接上 DHT11 传感器，并上电，当联网成功后 D3 会熄灭，如果联网不成功 D3 会闪烁。



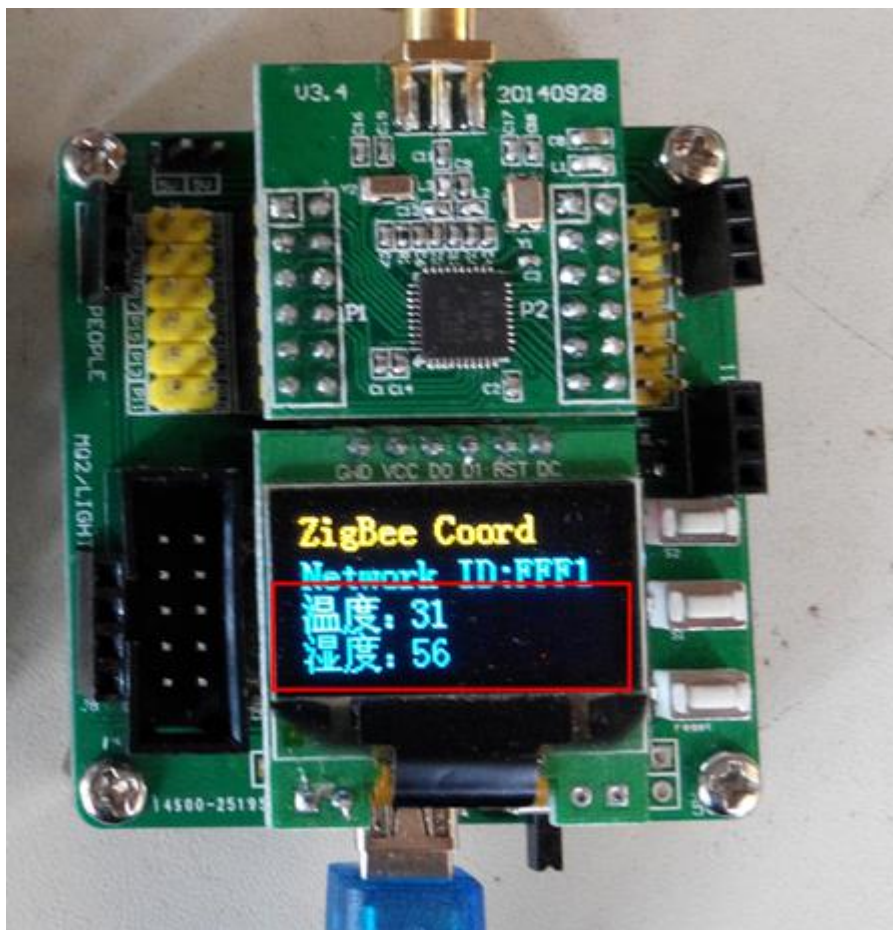
深圳市亿研电子有限公司

网址: <http://sz-yy.taobao.com/>

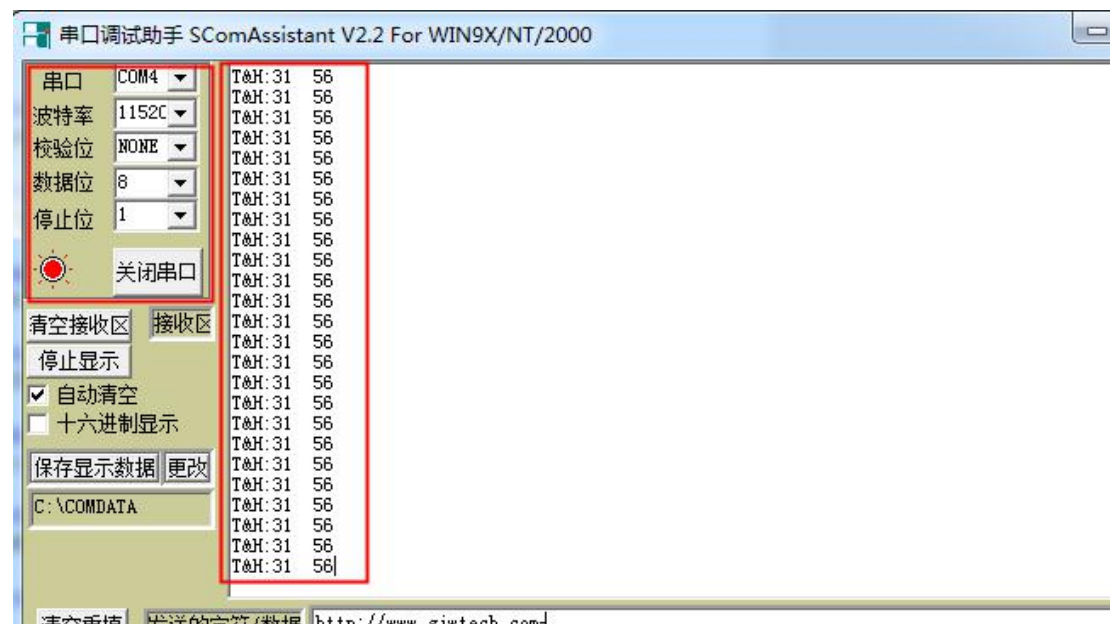
技术交流 QQ 群: 216388657 QQ:89339784

售后电话:18682359439

- 5) 观察协调器的 LCD 显示屏, 第三、第四行为温湿度显示。



- 6) 观察前面打开的串口 (协调器的串口)



- 7) 对着传感器吹气, 观察 LCD 和串口工具上的温度变化。

- 8) 也可以打开终端的串口, 同样有温度的输出。

- 9) LCD 接在终端上, 也可以看到 LCD 的上温度显示。

深圳市亿研电子有限公司

网址: <http://sz-yy.taobao.com/>

技术交流 QQ 群: 216388657 QQ:89339784

售后电话: 18682359439