



7.温湿度传感器 DHT11

1 . 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

2 . 实现现象

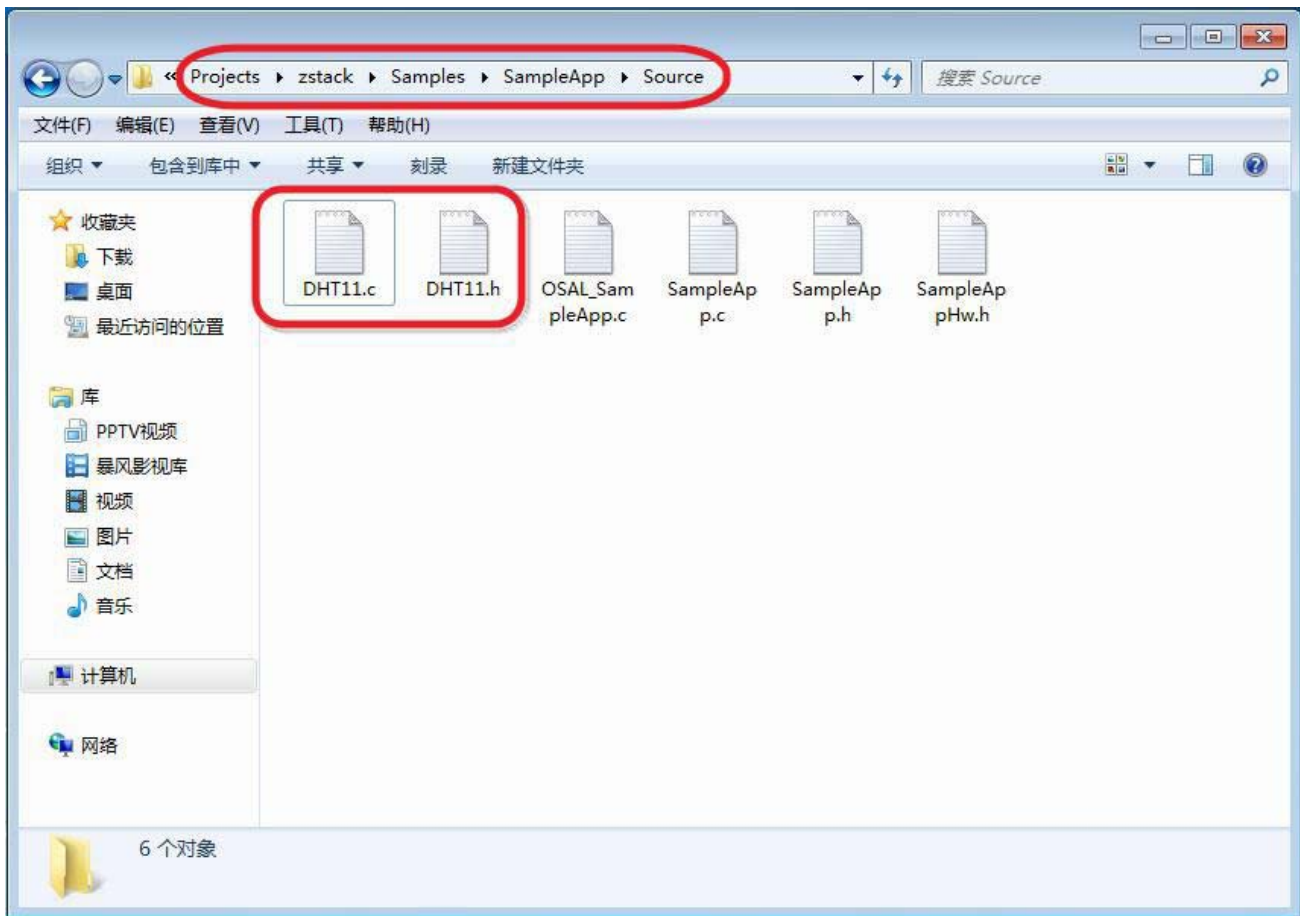
终端获得 DH T11 传感器的数据，无线传输给协调器；协调器再通过串口发给电脑串口调试助手显示。协调器、终端通过串口输出，LCD 也同步刷新。

3 . 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。

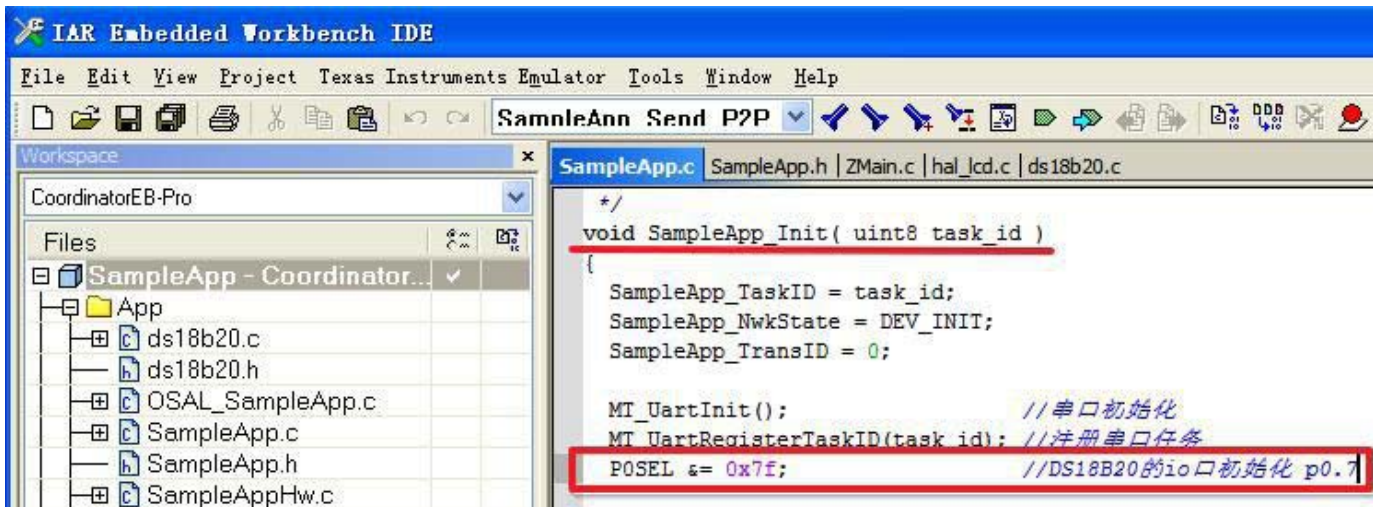
打开《3.高级篇-zigbee 协议栈应用与组网\7.温湿度传感器 DHT11\ZStack-CC2530-2.3.0-1.4.0\Projects\zstack\Samples\SampleApp\CC2530DB\SampleApp.eww》工程。

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++) //输出温度、湿度提示字符
```

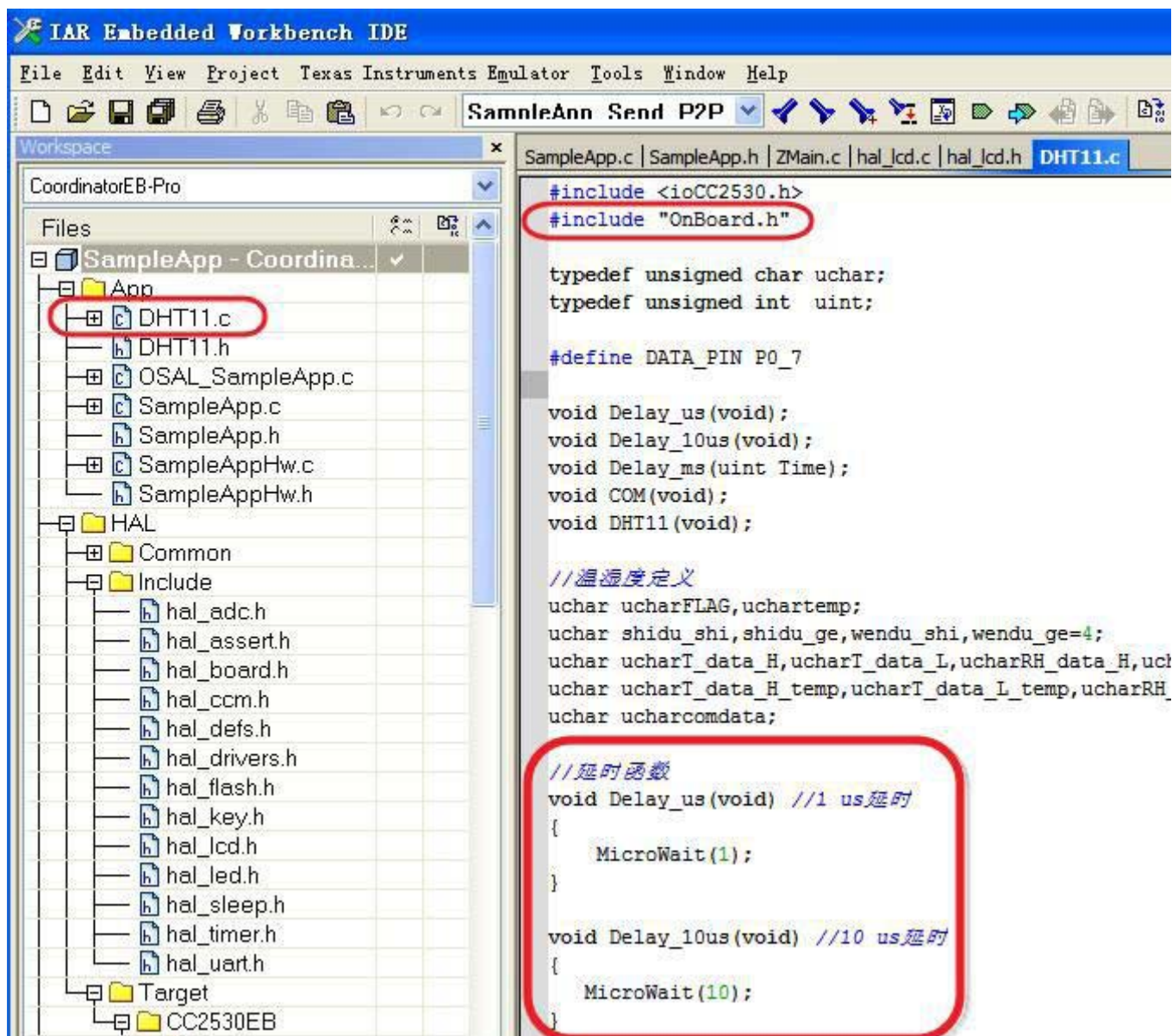
```
{  
    if(i==0)  
    {  
        LCD_P16x16Ch(i*16,4,i*16);  
        LCD_P16x16Ch(i*16,6,(i+3)*16);  
    }  
    else  
    {  
        LCD_P16x16Ch(i*16,4,i*16);  
        LCD_P16x16Ch(i*16,6,i*16);  
    }  
}  
LCD_P8x16Str(44, 4, temp); //LCD 显示温度值  
LCD_P8x16Str(44, 6, humidity); //LCD 显示湿度值  
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 )
```

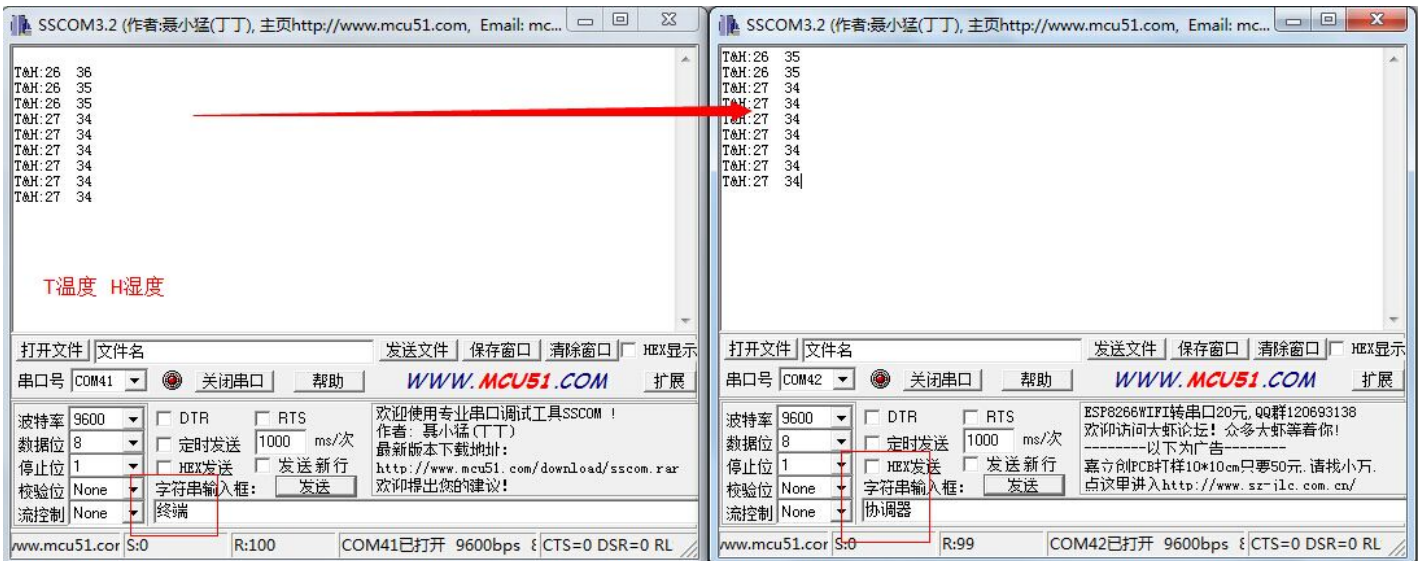
```
{
uint16 flashTime;
switch ( pkt->clusterId )
{
case SAMPLEAPP_P2P_CLUSTERID:
HalUARTWrite(0, "T&H:", 4); //提示接收到数据
HalUARTWrite(0, pkt->cmd.Data, pkt->cmd.DataLength); //输出接收到的数据
HalUARTWrite(0, "\n", 1); // 回车换行
break;
case SAMPLEAPP_PERIODIC_CLUSTERID:
break;
case SAMPLEAPP_FLASH_CLUSTERID:
flashTime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



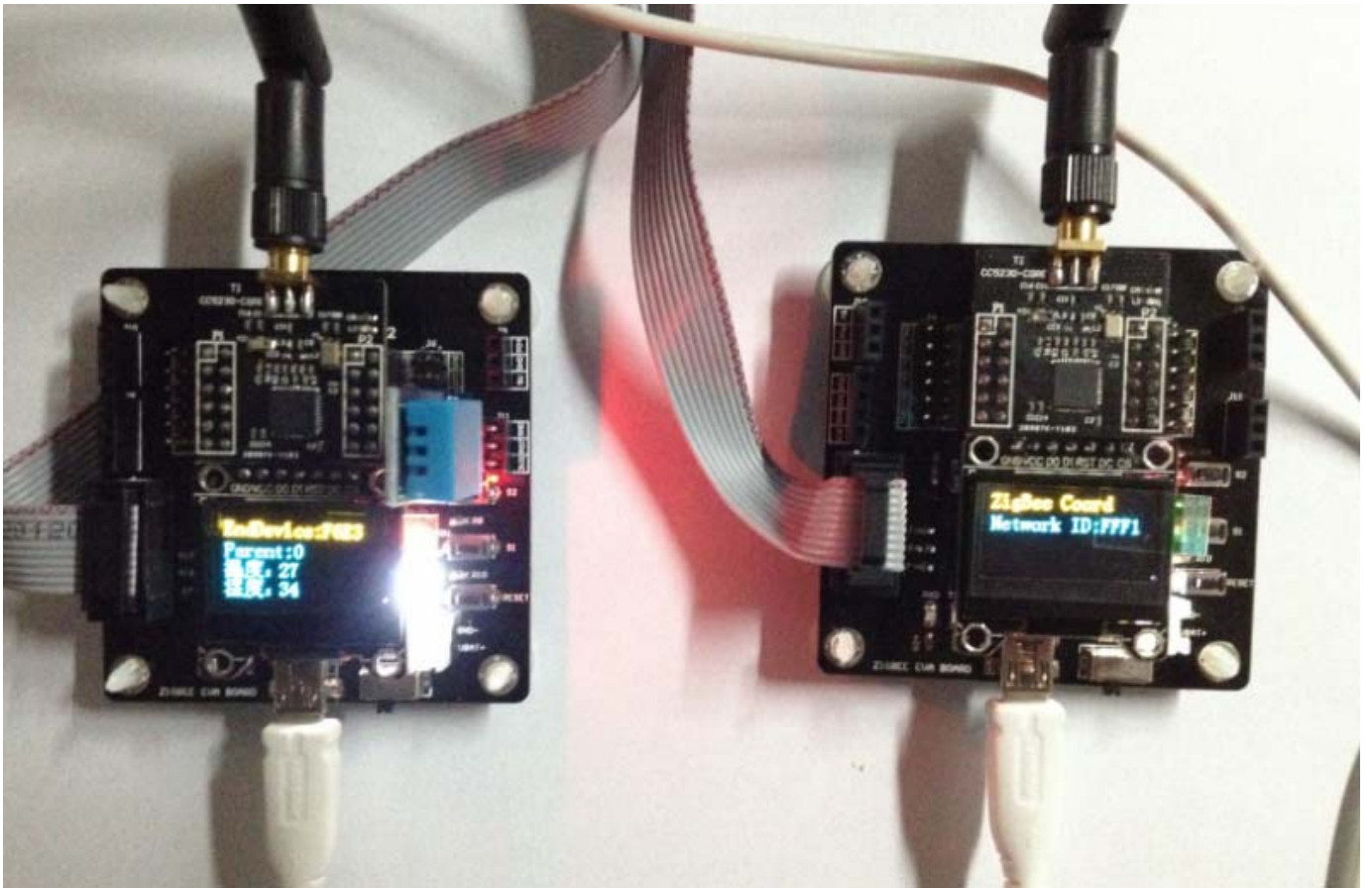
4. 实验步骤

- 1) 选择 CoordinatorEB-Pro, 下载到开发板 A ; 作为协调器 , 通过 USB 线跟电脑连接.
- 2) 选择 EndDeviceEB-Pro, 下载到开发板 B ; 作为终端设备无线发送数据给协调器, 也通过 USB 线跟电脑连接.
- 3) 给两块开发板上电 , 打开串口调试助手 , 设为:9600 8N1 并打开串口 , 串口请选择自己的端口号。终端连网成功后会向协调器发数据 , 实验结果如下图所示。



5. 实验结果





左边是终端 右边是协调器，终端采集温度，通过 ZigBee 发送到协调器，协调器经过串口，将数据发到电脑。