项目三　基础射频无线通信技术应用设计

【知识目标】

1．了解 Basic RF 工作机制；
2．熟悉无线发送和接收函数；
3．理解发送地址和接收地址、PAN_ID、RF_CHANNEL 等概念；
4．理解 CC2530_lib 库文件内各驱动文件的作用；
5．理解串口读写函数；
6．掌握各类典型传感器的工作原理。

【技能目标】

1．能够独立建立 Basic RF 工程；
2．会使用 CC2530 建立点对点的无线通信；
3．能够实现各类传感器信号采集功能；
4．能够实现基于 Basic RF 的信号采集与无线网络组建功能；
5．能够实现项目中多个设备组的工程配置；
6．初步了解项目文件管理方法。

【任务分解】

任务 3.1：Basic RF 无线控制 LED 灯
任务 3.2：Basic RF 无线串口通信
任务 3.3：开关量传感器采集系统
任务 3.4：模拟量传感器采集系统
任务 3.5：数字量传感器采集系统
任务 3.6：环境智能监测系统设计与应用

任务 3.1　Basic RF 无线控制 LED 灯

【任务描述】

以 Basic RF 无线点对点传输协议为基础，采用两个 ZigBee 模块作为遥控模块（发射模块）和被控对象模块（接收模块），按下发射模块上的 SW1 键，可以控制接收模块上的 LED1 灯的亮和灭，实现无线控制 LED 灯的功能。

【任务环境】

硬件：NEWLab 平台 2 套、ZigBee 节点板 2 块、CC2530 仿真器 1 组、PC 2 台。
软件：Windows 7/10，IAR 集成开发环境。

【必备知识点】

1．Basic RF 工作机制；
2．Basic RF 无线发送和接收函数；
3．Basic RF 发送地址和接收地址、PAN_ID、RF_CHANNEL 等概念。

3.1.1 Basic RF 工作原理

1．CC2530 Basic RF 工作机制

Basic RF 由 TI 公司提供，它包含了 IEEE 802.15.4 标准的数据包的收发功能但没有使用协议栈，Basic RF 仅让两个节点进行简单的通信，也就是说，Basic RF 仅包含 IEEE 802.15.4 标准的一小部分。其主要特点如下。

（1）不会自动加入协议。

（2）不会自动扫描其他节点，同时也没有组网指示灯。

（3）没有协议栈中的协调器、路由器或终端的区分，即各节点地位均相同。

（4）没有自动重发功能。

2．Basic RF 操作环节

Basic RF 操作依次包括启动、发送、接收三个环节。

1）*启动*

启动环节主要包括以下几项内容。

（1）初始化开发板的硬件外设和配置 I/O 端口。

（2）设置无线通信的网络 ID。

（3）设置无线通信的通信信道号。

（4）设置无线通信的接收和发送模块地址。

（5）若有必要，设置无线通信的网络安全加密等参数。

启动环节的以上内容的设置通过相关的数据结构体和相关函数来实现，涉及的结构体和函数如下。

（1）定义 basicRfCfg_t 数据结构体

basicRfCfg_t 数据结构体的定义在 basic_rf.h 文件中可以找到。数据结构体定义代码如下。

```
*******************************************************************************
typedef struct {
uint16 myAddr;              //本机地址，取值范围为 0x0000～0xffff，作为识别本模块的地址
uint16 panId;               //网络 ID，取值范围为 0x0000～0xffff，接收、发送模块此参数必须一致
uint8 channel;              //通信信道号，取值范围为 11～26，接收、发送模块此参数必须一致
uint8 ackRequest;           //应答信号
#ifdef SECURITY_CCM         //是否加密，预定义时取消了加密
uint8* securityKey;
uint8* securityNonce;
#endif
} basicRfCfg_t;
*******************************************************************************
```

（2）为 basicRfCfg_t 型结构体变量 basicRfConfig 填充部分参数

在 void main(void)函数中有如下 3 行代码，就是为 basicRfConfig 数据结构体部分变量赋值的。

```
*******************************************************************************
basicRfConfig.panId = PAN_ID;                   //宏定义：#define PAN_ID    0x2007
basicRfConfig.channel = RF_CHANNEL;             //宏定义：#define RF_CHANNEL     25
basicRfConfig.ackRequest = TRUE;                //宏定义：#define TRUE       1
*******************************************************************************
```

（3）调用 halBoardInit()函数

对硬件外设和 I/O 端口进行初始化，void halBoardInit()函数在 hal_board.c 文件中。

（4）调用 halRfInit()函数

此函数可打开射频模块，设置默认选项，允许自动确认和允许产生随机数。

2）发送

创建一个 buffer，把数据放入其中，调用 basicRfSendPacket()函数发送数据。在该工程中，light_switch.c 文件中的 appSwitch()函数是用来发送数据的。appSwitch()函数代码如下，请注意删除了液晶显示代码。

```
*******************************************************************************
1.  static void appSwitch()                          //开关功能函数
2.  {pTxData[0] = LIGHT_TOGGLE_CMD;                  //要发送的数据放到 buffer（即数组 pTxData）中
3.  basicRfConfig.myAddr = SWITCH_ADDR;              //本机地址
4.  if(basicRfInit(&basicRfConfig)==FAILED)          //初始化
5.  {     HAL_ASSERT(FALSE);          }
6.  basicRfReceiveOff();                             //关闭接收模式，节能
7.  while (TRUE)
8.    { if(halButtonPushed()==HAL_BUTTON_1)          //调用按键函数
9.        {
10. basicRfSendPacket(LIGHT_ADDR, pTxData, APP_PAYLOAD_LENGTH);      //调用发送函数
11. halIntOff();                                     //关中断
12. halMcuSetLowPowerMode(HAL_MCU_LPM_3);
13. halIntOn();                                      //开中断
14.     }
15. }
16. }
*******************************************************************************
```

说明：

（1）第 2 行，把要发送的数据 LIGHT_TOGGLE_CMD（宏定义该值为 1）放到 buffer 中，数组 pTxData 就是要发送的 buffer，即把要发送的数据存放到该数组中。

（2）第 3 行，为 basicRfCfg_t 型结构体变量 basicRfConfig.myAddr 赋值，宏定义 SWITCH_ADDR 为 0x2520，即发射模块的本机地址。

（3）第 4 行，调用 basicRfInit(&basicRfConfig)初始化函数，负责配置参数、设置中断等。在 basic_rf.c 文件中可以找到 uint8 basicRfInit(basicRfCfg_t* pRfConfig)。

（4）第 10 行，调用发送函数 basicRfSendPacket()，该函数带参数格式是：basicRfSendPacket (uint16 destAddr, uint8* pPayload, uint8 length)。

① destAddr 是发送的目标地址，实参是 LIGHT_ADDR，即接收模块的地址。

② pPayload 是指向发送缓冲区的地址，实参是 pTxData，该地址的内容是将要发送的数据。

③ length 是发送数据的长度，实参是 APP_PAYLOAD_LENGTH，单位是字节数。

3）接收

通过调用 basicRfPacketIsReady()函数来检查是否收到一个新的数据包，若有新数据，则调用 basicRfReceive()函数接收数据。在该工程中，light_switch.c 文件中的 appLight()函数是用来发送数据的。appLight()函数代码如下，请注意删除了液晶显示代码。

```
*******************************************************************************
1.  static void appLight()                              //LED 灯相关函数
2.  {    basicRfConfig.myAddr = LIGHT_ADDR;             //设定本模块地址
3.       if(basicRfInit(&basicRfConfig)==FAILED)        //初始化，方法与发送环节一样
4.     {   HAL_ASSERT(FALSE);    }
5.  basicRfReceiveOn();                                 //开启接收功能
6.       while (TRUE)
7.  {
8.            while(!basicRfPacketIsReady());          //检查是否有新数据，没有则一直等待
9.             if(basicRfReceive(pRxData, APP_PAYLOAD_LENGTH, NULL)>0)
10. { if(pRxData[0] == LIGHT_TOGGLE_CMD)                //判断接收到的内容是否正确
11. {    halLedToggle(1);   }                           //改变 LED1 灯的亮、灭状态
12.           }
13.      }
14. }
*******************************************************************************
```

说明：

（1）第 8 行，调用 basicRfPacketIsReady()函数来检查是否收到一个新数据包，若有新数据，则返回 TRUE。新数据包信息存放在 basicRfRxInfo_t 型结构体变量 rxi 中。

```
*******************************************************************************
1.  typedef struct { uint8   seqNumber;
2.       uint16 srcAddr;                //数据的来源地址，即发送模块的地址
3.       uint16 srcPanId;               //网络 ID
4.       int8 length;                   //新数据的长度
5.       uint8* pPayload;               //新数据包的存放地址
6.       uint8 ackRequest;
7.       int8 rssi;                     //信号强度
8.       volatile uint8 isReady;        //检查到新数据包的标志
9.       uint8 status;
10. } basicRfRxInfo_t;
*******************************************************************************
```

（2）第 9 行，调用 basicRfReceive(pRxData, APP_PAYLOAD_LENGTH, NULL)函数，把接收到的数据复制到 buffer 中，即 pRxData，注意与用来发送数据 buffer 的 pTxData 相区别。

```
*******************************************************************************
1.        uint8 basicRfReceive(uint8* pRxData, uint8 len, int16* pRssi)
2.        { halIntOff();                          //关中断
3.    //从 rxi.pPayload 中复制数据到 pRxData 中
4.    memcpy(pRxData, rxi.pPayload, min(rxi.length, len));
5.             if(pRssi != NULL) {
6.                  if(rxi.rssi < 128){
7.                  *pRssi = rxi.rssi - halRfGetRssiOffset();       }
8.            else{
9.                  *pRssi = (rxi.rssi - 256) - halRfGetRssiOffset();
10.              }
11.                                 }
12.          rxi.isReady = FALSE;               //取消新数据包标志
13.          halIntOn();                        //开中断
```

```
14.             return min(rxi.length, len);              //返回接收的字节数（最少的）
15.   }
      ********************************************************************************
```

从上述代码可知：接收到的新数据被复制到 pRxData 中。

说明：rssi 一般是用来说明无线信号强度的，英文是 received signal strength indication，它与模块的发送功率及天线的增益有关。

（3）第 10 行，判断接收到的内容是否与发送的数据一致。若一致，则改变 LED1 灯的亮、灭状态。

3.1.2　任务实训步骤

第 1 步，下载 CC2530 Basic RF 源文件。

登录 TI 官网，下载 CC2530 BasicRF.rar，解压后双击打开“\CC2530 BasicRF\CC2530 BasicRF\ide\srf05_cc2530\iar”文件夹中的“light_switch.eww”工程文件，如图 3-1 所示。

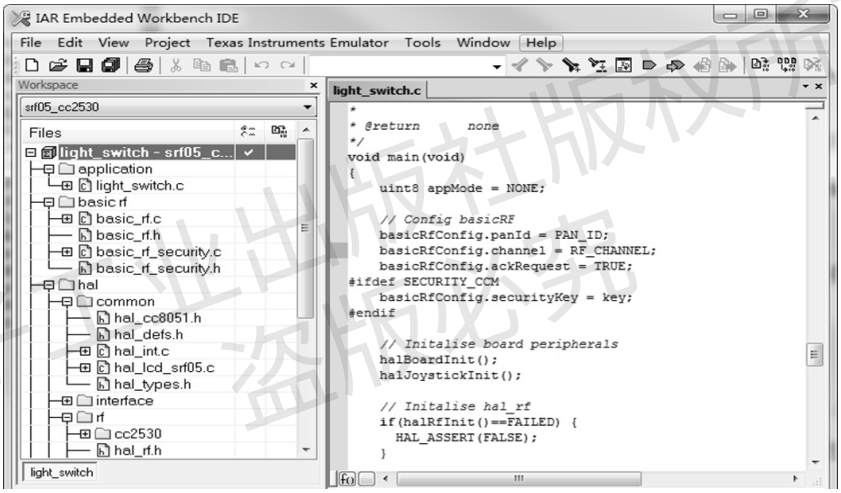


图 3-1 “light_switch.eww”工程文件

第 2 步，修改程序。

ZigBee 模块（网关节点）上有 2 个按键和 4 个 LED 灯，其中按键 SW1 和 SW2 分别由 P1.2 和 P1.6 控制，LED1～LED4 灯分别由 P1.0、P1.1、P1.3 和 P1.4 控制，如图 3-2 所示。这些接口与 TI 官网发布的开发平台有所差别，所以需要修改一下，操作方法如下。

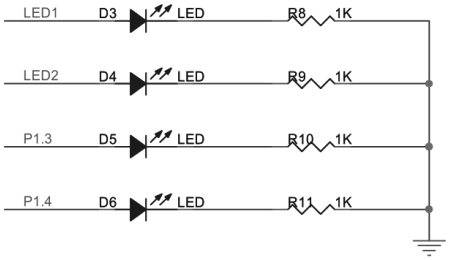


图 3-2　LED 灯与 P1 引脚连接图

（1）打开“hal_board.h”头文件，打开方法有以下两种。

① 展开工程左边“Workspace”栏中“light_switch.c”的“+”号，就可以在展开的文件

列表中找到“hal_board.h”头文件，双击该文件，就可以打开它。

② 在“light_switch.c”文件的开始部分代码中，可以找到“include <hal_board.h>”宏定义，右击该宏定义并选中“Open hal_board.h”命令，也可以打开该文件。

（2）在“hal_board.h”头文件中找到如下代码，并按照图 3-3 所示要求修改。

```
// LEDs
#define HAL_BOARD_IO_LED_1_PORT        1   // Green
#define HAL_BOARD_IO_LED_1_PIN         0
#define HAL_BOARD_IO_LED_2_PORT        1   // Red
#define HAL_BOARD_IO_LED_2_PIN         1
#define HAL_BOARD_IO_LED_3_PORT        1   // Yellow
#define HAL_BOARD_IO_LED_3_PIN         3
#define HAL_BOARD_IO_LED_4_PORT        1   // Orange
#define HAL_BOARD_IO_LED_4_PIN         4


// Buttons
#define HAL_BOARD_IO_BTN_1_PORT        1   // Button S1
#define HAL_BOARD_IO_BTN_1_PIN         2
```

图 3-3　代码的修改

说明：其中，HAL_BOARD_IO_LED_x_PORT 表示 x 端口（x 可以是 0、1、2…）；HAL_BOARD_IO_LED_y_PIN 表示 x 的 y 引脚（x 端口的第 y 个引脚，y 可以是 0～7）。

（3）修改“light_switch.c”文件中的 static void appSwitch()函数代码，把该函数中的“if(halJoystickPushed()){”代码注释掉，在其下一行添加“if(halButtonPushed()==HAL_BUTTON_1){ ”代码。

（4）注释掉部分代码，如图 3-4 所示。

```
    //Indicate that device is povered
    halLedSet(1);

    //Print Logo and splash screen on LCD
    //utilPrintLogo("Light Switch");

    //wait for user to press S1 to enter menu
    //while(halButtonPaushed()!= HAL_BUTTON_1);
    halMcuWaitMs(350);
    halLcdClear();

    //Set application role
    //appMode = appSelectMode();
    halLcdClear();
```

图 3-4　注释掉部分代码

第 3 步，编译、烧录程序。

修改程序后，进行编译，编译无误后分别发给发射模块和接收模块烧录程序。

（1）在“light_switch.c”文件的主函数中找到“uint8 appMode = NONE;”代码，并把它注释掉，在其下一行添加“uint8 appMode = SWITCH;”代码。编译程序，无误后下载到发射模块中。

（2）在“light_switch.c”文件的主函数中找到“uint8 appMode = SWITCH;”代码，将其修改为“uint8 appMode = LIGHT;”代码。编译程序，无误后下载到接收模块中。

第 4 步，测试程序功能。

每按一下发射模块中的 SW1 键，接收模块上的 LED1 灯的状态就会改变，即 LED1 灯亮、灭交替变化。把两个模块隔开 20m 以上的距离，再进行测试。

任务 3.2　Basic RF 无线串口通信

【任务描述】

以 Basic RF 无线点对点传输协议为基础，采用两个 ZigBee 模块（作为节点 1 和节点 2），这两个节点分别与两台计算机的串口连接。打开节点 1 和节点 2 对应计算机上的串口调试软件，相互收发信息，实现无线串口通信。

【任务环境】

硬件：NEWLab 平台 2 套、ZigBee 节点板 2 块、CC2530 仿真器 1 组、PC 2 台。

软件：Windows 7/10，IAR 集成开发环境，串口调试软件。

【必备知识点】

1．串口通信原理；

2．串口调试软件的使用。

3.2.1　串口通信原理

1．串口通信的定义

串行通信接口，简称串口（COM），出现在 1980 年前后，初始数据传输速率是 115～230kbit/s。串口出现初期，其功能是连接计算机外设，初期串口一般用来连接鼠标和外置 Modem 及老式摄像头和写字板等设备。串口也可以应用于两台计算机（或设备）之间的互联及数据传输。由于串口不支持热插拔及传输速率较低，部分新主板和大部分笔记本电脑已开始取消该接口。目前串口多用于工控和测量设备及部分通信设备中。

串口通信是指数据一位一位地顺序传送，其特点是通信线路简单，只要一对传输线就可以实现双向通信（可以直接利用电话线作为传输线），从而大大降低了成本，特别适用于远距离通信，但传送速度较慢。串口通信的特点：数据的传送按位顺序进行，最少只需一根传输线即可完成；成本低但传送速度慢。串口通信的距离可以从几米到几千米；根据信息的传送方向，串口通信可以分为单工、半双工和全双工三种。

尽管按位（bit）传送的串口通信比按字节（Byte）传送的并行通信速度慢，但是串口通信可以在使用一根线发送数据的同时用另一根线接收数据，很简单并且能够实现远距离通信。比如 IEEE 488 定义并行通信状态时，规定设备线总长不得超过 20m，并且任意两个设备间的长度不得超过 2m；而对于串口通信而言，其设备线长度可达 1200m。典型的，串口通信可用于 ASCII 码字符的传输。串口通信使用 3 根线完成，分别是地线、发送线、接收线。

2．串口的分类

按照运行模式，可将串口分为同步串口和异步串口两种。同步串口（Synchronous Serial Interface，SSI）是一种常用的工业通信接口。异步串口（Universal Asynchronous Receiver/Transmitter，UART），是指通用异步接收/发送器。UART 是一个将并行输入转为串行输出的芯片，通常集成在主板上。UART 包含 TTL 电平的串口和 RS-232 电平的串口。TTL 电平是 3.3V 的，而 RS-232 是负逻辑电平，它定义+5～+12V 为低电平，−12～−5V 为高电平。MDS2710、MDS SD4、EL805 等都是 RS-232 接口，EL806 有 TTL 接口。

CC2530 芯片共有 USART0 和 USART1 两个串口，它能够运行于异步模式或者同步模式。两个 USART 具有同样的功能，可以设置单独的 I/O 引脚。

在 UART 模式中，可以使用双线（包括 RXD、TXD）连接方式或四线（包括 RXD、TXD、RTS 和 CTS）连接方式，其中 RTS 和 CTS 引脚用于硬件流量控制。

对于每个 USARTx，都有控制和状态寄存器（UxCSR）、UART 控制寄存器（UxUCR）、通用控制寄存器（UxGCR）、接收/发送数据缓冲寄存器（UxDBUF）、波特率控制寄存器（UxBAUD）5 个寄存器。其中，x 是 USART 的编号，为 0 或者 1。本节内容涉及的串行通信接口相关寄存器如下。

（1）U0CSR：USART0 控制和状态寄存器，如表 3-1 所示。

表 3-1　U0CSR 相关寄存器

D7	D6	D5	D4	D3	D2	D1	D0
工作模式选择	接收器使能/禁用	SPI 主/从模式	帧错误检测状态	奇偶错误检测状态	字节接收状态	字节传送状态	接收/传送主动状态

D7 为工作模式选择，0 为 SPI 模式，1 为 USART 模式。

D6 为接收器使能/禁用，0 为接收器禁用，1 为接收器使能。

D5 为 SPI 主/从模式选择，0 为 SPI 主模式，1 为 SPI 从模式。

D4 为帧错误检测状态，0 为无错误，1 为出现帧错误。

D3 为奇偶错误检测状态，0 为无错误，1 为出现奇偶校验错误。

D2 为字节接收状态，0 为没有收到字节，1 为准备好接收字节。

D1 为字节传送状态，0 为字节没有被传送，1 为写到数据缓冲区中的字节已经被发送。

D0 为接收/传送主动状态，0 为 USART 空闲，1 为 USART 忙碌。

（2）U0GCR：USART0 通用控制寄存器，如表 3-2 所示。

表 3-2　U0GCR 相关寄存器

D7	D6	D5	D4～D0	U0BAUD
SPI 时钟极性	SPI 时钟相位	传送位顺序	波特率指数值	波特率控制小数部分

D7 为 SPI 时钟极性：0 为负时钟极性，1 为正时钟极性。

D6 为 SPI 时钟相位。

D5 为传送位顺序：0 为最低有效位先传送，1 为最高有效位先传送。

D4～D0 为波特率指数值：具体设置值见表 3-3。

U0BAUD 为波特率控制小数部分，见表 3-3。

表 3-3　32MHz 系统时钟常用的波特率设置值

波特率/bps	指　数　值	小 数 部 分
2400	6	59
4800	7	59
9600	8	59
14400	8	216
19200	9	59
28800	9	216

续表

波特率/bps	指　数　值	小 数 部 分
38400	10	59
57600	10	216
76800	11	59
115200	11	216
230400	12	216

3．串口通信的特征参数

串口通信最重要的参数是波特率、数据位、停止位和奇偶校验位。两个进行通信的端口，这些参数必须匹配。

（1）波特率

这是一个衡量符号传输速率的参数，表示每秒传送的符号的个数。例如波特率 300bps 表示每秒发送 300 个符号。当提到时钟频率时，就是指波特率。例如，如果协议需要波特率为 4800bps，那么时钟频率是 4800Hz，这意味着串口通信在数据线上的采样率为 4800Hz。通常电话线的波特率为 14400bps、28800bps 和 36600bps。波特率可以远远大于这些值，但是波特率和距离成反比。高波特率常常用于距离很近的仪器间的通信，典型的例子就是 GPIB 设备的通信。

（2）数据位

这是衡量通信中实际数据位的参数。计算机发送一个信息包，实际的数据不一定是 8 位的，标准的值是 5、6、7 和 8 位。如何设置取决于想传送的信息，比如，标准的 ASCII 码是 0～127（7 位），扩展的 ASCII 码是 0～255（8 位）。如果数据使用简单的文本（标准 ASCII 码），那么每个包使用 7 位数据。每个包是指一个字节，包括开始/停止位、数据位和奇偶校验位。由于实际数据位取决于通信协议的选取，“包”可指任何通信的情况。

（3）停止位

用于表示单个包的最后一位，典型的值为 1、1.5 和 2。由于数据是在传输线上定时的，并且每个设备都有自己的时钟，很可能在通信中两台设备间出现了小小的不同步。因此停止位不仅表示传输的结束，而且为计算机提供校正时钟同步的机会。适用于停止位的位数越多，不同时钟同步的容忍程度越大，但是数据传输也越慢。

（4）奇偶校验位

这是串口通信中一种简单的检错方式。共有 3 种检错方式：奇、偶、高和低，当然没有校验位也是可以的。对于奇和偶校验的情况，串口会设置校验位（数据位后面的一位），用一个值确保传输的数据有偶数个或者奇数个逻辑高位。例如，如果数据是 011，那么对于偶校验，校验位为 0，保证逻辑高的位数是偶数。如果是奇校验，校验位为 1，这样就有 3 个逻辑高位。高位和低位不是真正地检查数据，而是简单置位逻辑高或者逻辑低校验，这样使得接收设备能够知道一个位的状态，有机会判断是否有噪声干扰了通信或者传输、接收数据是否不同步。

需要补充的是，在数字信道中，比特率代表数字信号的传输速率，它用单位时间内传输的二进制代码的有效位（bit）数来表示，其单位为每秒比特数 bit/s（bps）、每秒千比特数（kbps）或每秒兆比特数（Mbps）（此处 k 和 M 分别表示 1000 和 1000000，而不是涉及计算机存储器容量时的 1024 和 1048576）。而波特率是指数据信号对载波的调制速率，它用单位时间内载波调制状态改变次数来表示。波特率与比特率的关系：比特率=波特率×单个调制状态对应的

二进制位数。显然，两相调制（单个调制状态对应 1 个二进制位）的比特率等于波特率；四相调制（单个调制状态对应 2 个二进制位）的比特率为波特率的 2 倍；八相调制（单个调制状态对应 3 个二进制位）的比特率为波特率的 3 倍，依次类推。

除此之外，不同协议标准的串口通信的传输距离不同，RS-232 用在近距离传输上，最大传输距离为 30m；RS-485 用在长距离传输上，最大传输距离为 1200m。

4．串口通信函数分析

本节任务在 Basic RF 无线通信功能基础上，结合串口通信，最终实现无线串口通信，程序中涉及串口数据发送与接收两个步骤。

串口数据发送：创建一个 buffer，把数据放入其中，然后调用 halUartWrite()函数发送数据。

串口数据接收：调用 RecvUartDate()函数来接收数据，并根据数据长度来判断是否收到数据。

3.2.2 任务实训步骤

第 1 步，新建工程和程序文件，添加头文件。

（1）复制库文件。将 CC2530_lib 文件夹复制到该任务的工程文件夹内，即“F:\ZigBee\任务 3.2 无线串口通信”（也可以放在其他文件夹内）。在该工程文件夹内新建一个 Project 文件夹，用于存放工程文件。

（2）新建工程。具体方法参照任务 2.3。在工程中新建 App、basicrf、board、common、utils 5 个组，把各文件夹中的“xx.c”文件添加到对应的文件夹中。

（3）新建程序文件。将其命名为“uartRF.c”，保存在“F:\ZigBee\任务 3.2 无线串口通信\Project”文件夹中，并将该文件添加到工程中的 App 文件夹中。

（4）为工程添加头文件。选择 IAR 菜单中的“Project”→“Options”命令，在弹出的对话框中选择“C/C++ Compiler”，然后选择“Preprocessor”选项卡，并在“Additional include directories”中输入头文件的路径，如图 3-5 所示，单击“OK”按钮。

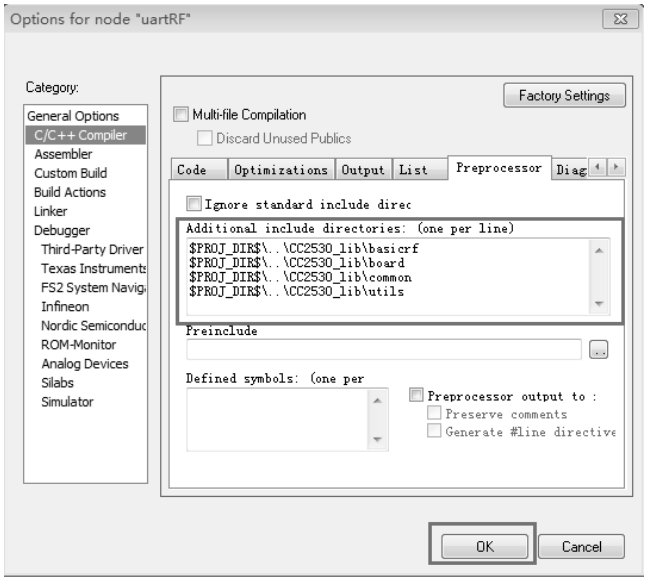


图 3-5　为工程添加头文件

注意：

图 3-5 中，“$PROJ_DIR$\”即当前工作的 Workspace 的目录。“…\”表示对应目录的上一层。

例如：“$TOOLKIT_DIR$\INC\”和“$TOOLKIT_DIR$\INC\CLIB\”，都表示当前工作的 Workspace 的目录。“$PROJ_DIR$\ ...\INC”表示用户的 Workspace 目录上一层的 INC 目录。

第 2 步，配置工程。

选择 IAR 菜单中的“Project”→“Options”命令，分别对“General Options”“Linker”“Debugger”三项进行配置。

（1）配置“General Options”。选中“Target”选项卡，在“Device”栏内选择“CC2530F256.i51”文件（路径为“C:\...\8051\config\devices\Texas Instruments”），如图 3-6 所示。

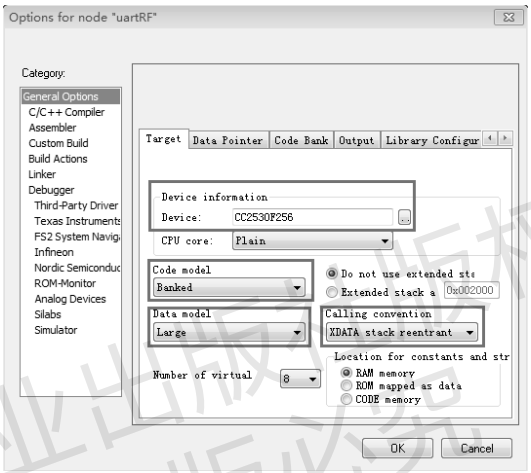


图 3-6　配置“General Options”

（2）配置“Linker”。选中“Config”选项卡，勾选“Override default”复选框，并在该栏内选择“lnk51ew_CC2530F256_banked.xcl”配置文件，其路径为“C:\...\8051\config\devices\Texas Instruments”。

（3）配置“Debugger”。选中“Setup”选项卡，在“Driver”栏内选择“Texas Instruments”；在“Device Description file”栏内，勾选“Override default”复选框，并在该栏内选择“io8051.ddf”配置文件，其路径为“C:\…\config\devices_generic”，如图 3-7 所示。

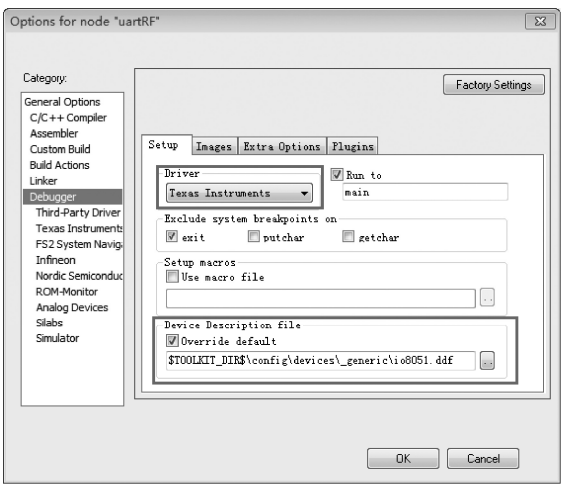


图 3-7　配置“Debugger”

第 3 步，编写程序。

由于程序很长，只能对关键部分进行分析，其他部分详见 uartRF.c 文件。

```
/*****************************点对点通信地址设置*********************************/
1.  #define RF_CHANNEL      20          //通信信道号 11～26
2.  #define PAN_ID          0x1379      //网络 ID
3.  //#define MY_ADDR       0x1234      //模块 A 的地址
4.  //#define SEND_ADDR     0x5678      //模块 A 发送模块 B 的地址
5.  #define MY_ADDR         0x5678      //模块 B 的地址
6.  #define SEND_ADDR       0x1234      //模块 B 发送模块 A 的地址
/*******************************************************************************/
/*****************************main**********************************************/
1.  void main(void)
2.  {uint16 len = 0;
3.  halBoardInit();                     //模块相关资源的初始化
4.  ConfigRf_Init();                    //无线收、发模块参数的配置初始化
5.  halLedSet(3);
6.  halLedSet(4);
7.  while(1)
8.  { len = RecvUartData();             //接收串口数据
9.  if(len > 0)
10. { halLedToggle(3);                  //绿灯取反，无线数据发送指示
11.                                     //把串口数据通过 ZigBee 发送出去
12. basicRfSendPacket(SEND_ADDR, uRxData，len);
13. }
14. if(basicRfPacketIsReady())          //查询有无收到无线数据
15. { halLedToggle(4);                  //红灯取反，无线数据接收指示
16.                                     //接收无线数据
17. len = basicRfReceive(pRxData, MAX_RECV_BUF_LEN, NULL);
18.                                     //将接收到的无线数据发送到串口
19. halUartWrite(pRxData，len);
20. }
21. }
22. }
/*****************************main end*****************************************/
```

第 4 步，烧录程序。

（1）为无线模块 A 下载程序。注释掉上述程序“点对点通信地址设置”的第 5 和第 6 行，重新编译程序无误后，下载到无线模块 A 中。

（2）为无线模块 B 下载程序。注释掉上述程序“点对点通信地址设置”的第 3 和第 4 行，重新编译程序无误后，下载到无线模块 B 中。

注意：如果有多组同学同时进行实训，每组间的 RF_CHANNEL 和 PAN_ID 值至少要有一个不同。如果多组间的 RF_CHANNEL 和 PAN_ID 值都一样，则会造成信号串扰。

第 5 步，运行程序。

（1）分别把节点 1 和节点 2 接到 PC 的串口，打开两个串口调试软件，把串口的波特率设置为 38400bps，再给两个模块上电。

（2）在两个串口调试软件上，发送不同的信息，并能显示对方发送的信息，如图 3-8、图 3-9 所示。

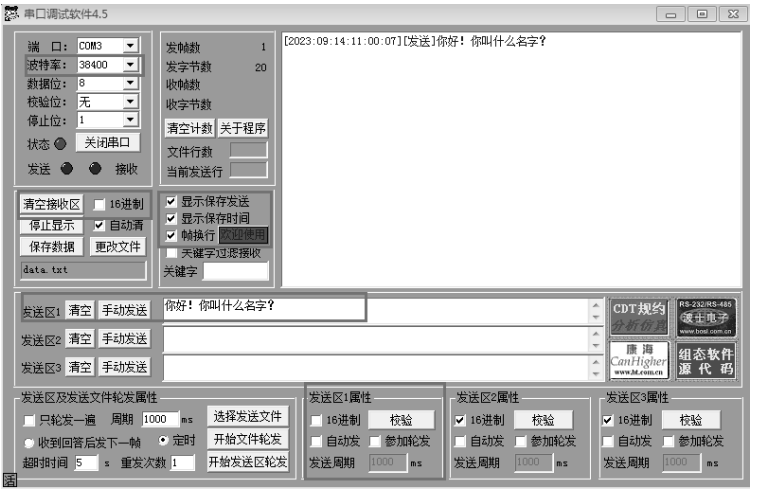


图 3-8　串口调试窗口 1

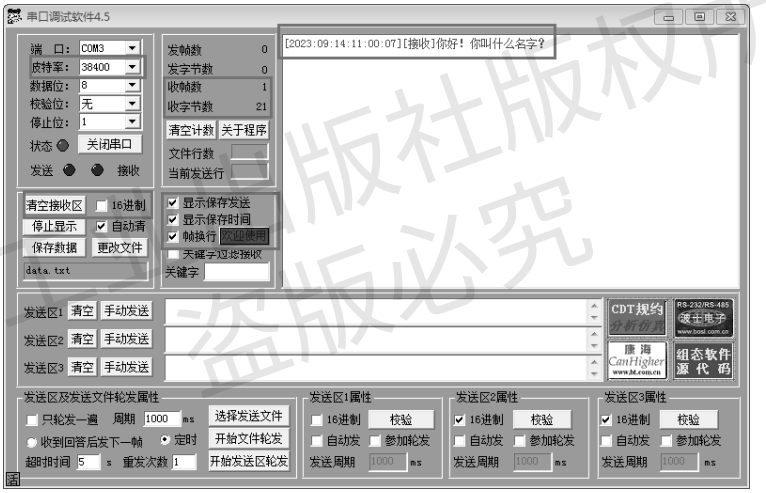


图 3-9　串口调试窗口 2

任务 3.3　开关量传感器采集系统

【任务描述】

采用声音传感器、红外传感器等，以及 ZigBee 模块组成一个开关量传感器采集系统。当声音传感器检测到声音时，系统会点亮 ZigBee 模块上的 LED1 灯，延时 2s 后，若没有再检测到声音，则熄灭 LED1 灯；当红外传感器检测到红外信号时，系统立即使 ZigBee 模块上的 LED2 灯点亮，反之则使 LED2 灯熄灭。

【任务环境】

硬件：NEWLab 平台 1 套、ZigBee 节点板 1 块、声音传感器 1 个、红外传感器 1 个、CC2530 仿真器 1 组、PC 1 台，信号线若干。

软件：Windows 7/10，IAR 集成开发环境。

【必备知识点】

1．传感器的技术原理；

2．红外传感器工作原理；

3．声音传感器工作原理。

3.3.1 传感器的技术原理

传感器是一种检测装置，能感受到被测量的信息，并能将感受到的信息，按一定规律变换成电信号或其他所需形式的信息输出，以满足信息的传输、处理、存储、显示、记录和控制等要求。它是实现自动检测和自动控制的首要环节，是物联网应用的信息来源。为了从外界获取信息，人必须借助于感觉器官，在研究自然现象和规律及生产活动中，人们自身的感觉器官的功能就远远不够了。为了适应这种情况，就需要用到传感器。因此可以说，传感器是人类五官的延长，又称为“电五官”。

国家标准 GB/T 7665—2005 对传感器的定义是：“能感受被测量并按照一定的规律转换成可用输出信号的器件或装置，通常由敏感元件和转换元件组成。”传感器在新韦式大词典中的定义为：“从一个系统接收功率，通常以另一种形式将功率送到第二个系统中的器件。”

新技术革命的到来，世界开始进入信息时代。在利用信息的过程中，首先要解决的就是获取准确可靠的信息，而传感器是获取自然和生产领域中信息的主要途径与手段。随着数字技术，特别是信息技术的飞速发展与普及，在现代控制、通信及检测等领域，为了提高系统的性能指标，对信号的处理广泛采用了数字计算机技术。由于系统的实际对象往往都是一些模拟量（如温度、压力、位移、图像等），要使计算机或数字仪表能识别、处理这些信号，必须首先将这些模拟信号转换成数字信号；而经计算机分析、处理后输出的数字量往往也需要将其转换为相应模拟信号才能为执行机构所接收。传感器技术是实现测试与自动控制的重要环节。在测试系统中，作为一次仪表定位，其主要特征是能准确传递和检测出某一形态的信息，并将其转换成另一形态的信息。这样，就需要一种能在模拟信号与数字信号之间起桥梁作用的电路——模/数和数/模转换器，即 A/D 和 D/A 转换器。

1．A/D 和 D/A 转换器的工作原理

将模拟信号转换成数字信号的电路，称为模/数转换器（简称 A/D 转换器或 ADC），将数字信号转换为模拟信号的电路称为数/模转换器（简称 D/A 转换器或 DAC）。A/D 转换器和 D/A 转换器已成为信息系统中不可缺少的接口电路。

A/D 转换包括采样、保持、量化和编码四个过程。在某些特定的时刻对这种模拟信号进行测量叫作采样，通常采样脉冲的宽度是很窄的，所以采样输出是断续的窄脉冲。要把一个采样输出信号数字化，需要将采样输出所得的瞬时模拟信号保持一段时间，这就是保持过程。量化是指将保持的抽样信号转换成离散的数字信号。编码是指将量化后的信号编码成二进制代码输出。这些过程有些是合并进行的，例如，采样和保持就利用一个电路连续完成，量化和编码也是在转换过程中同时实现的，且所用时间又是保持时间的一部分。

A/D 转换器中，模拟电压经电路转换后的 AD 值为：

$$\mathrm{AD}=\frac{U_{\mathrm{A}}}{V_{\mathrm{DD}}}\times 2^{n}=\frac{2^{n}}{V_{\mathrm{DD}}}\times U_{\mathrm{A}} \tag{3-1}$$

式中，AD 为转换电路输出的电压，U_{A} 为被测模拟电压，n 为采用 A/D 转换的精度位数，V_{DD} 为转换电路的供电电压。传感器实验模块中精度为 8 位，供电电压为 3.3V。

2．A/D 转换器的主要性能指标

在检测控制系统和科学实验中，需要对各种参数进行检测和控制，而要达到比较优良

的控制性能，要求传感器必须能够感测被测量的变化并且不失真地将其转换为相应的电量，这种要求主要取决于传感器的基本特性。传感器的基本特性分为静态特性和动态特性。

（1）传感器的静态特性

静态特性是指检测系统的输入为不随时间变化的恒定信号时，系统的输出与输入之间的关系，主要包括线性度、灵敏度、迟滞、重复性、漂移等。

① 线性度：指传感器输出量与输入量之间的实际关系曲线偏离拟合直线的程度。

② 灵敏度：灵敏度 S 是传感器静态特性的一个重要指标。其定义为输出量的增量Δy 与引起该增量的相应输入量增量Δx 之比。它表示单位输入量的变化所引起传感器输出量的变化，显然，灵敏度越大，传感器越灵敏。

③ 迟滞：传感器在输入量由小到大（正行程）及输入量由大到小（反行程）变化期间，其输入/输出特性曲线不重合的现象称为迟滞。也就是说，对于同样大小的输入信号，传感器的正反行程输出信号大小不相等，这个差值称为迟滞差值。

④ 重复性：指传感器在输入量按同一方向做全量程连续多次变化时，所得特性曲线不一致的程度。

⑤ 漂移：在输入量不变的情况下，传感器输出量会随着时间变化，此现象称为漂移。产生漂移的影响因素有两个：一是传感器自身结构参数，二是周围环境（如温度、湿度等）。最常见的漂移是温度漂移，即周围环境温度变化而引起输出量的变化，温度漂移主要表现为温度零点漂移和温度灵敏度漂移。温度漂移通常用传感器工作环境温度偏离标准环境温度（一般为 20℃）时的输出量的变化量与温度变化量之比来表示。

⑥ 测量范围：指传感器所能测量的最小输入量与最大输入量之间的范围。

⑦ 量程：指传感器测量范围的上限值与下限值的代数差。

⑧ 精度：指测量结果的可靠程度，是测量中各类误差的综合反映，测量误差越小，传感器的精度越高。传感器的精度用其量程范围内的最大基本误差与满量程输出之比的百分数表示，其基本误差是传感器在规定的正常工作条件下所具有的测量误差，由系统误差和随机误差两部分组成。工程中为简化传感器精度的表示方法，引入了精度等级的概念。精度等级以一系列标准百分比数值分挡表示，代表传感器测量的最大允许误差。如果传感器的工作条件偏离正常工作条件，还会带来附加误差，温度误差就是最主要的附加误差。

⑨ 分辨力和阈值：传感器测量输入量最小变化量的能力称为分辨力。对于某些传感器，如电位器式传感器，当输入量连续变化时，输出量只做阶梯变化，则分辨力就是输出量的每个“阶梯”所代表的输入量的大小。对于数字式仪表，分辨力就是仪表指示的最后一位数字所代表的值。当被测量的变化量小于分辨力时，数字式仪表的最后一位数字不变，仍指示原值。当分辨力以满量程输出的百分数表示时则称为分辨率。阈值是指能使传感器的输出端产生可测变化量的最小被测输入量，即零点附近的分辨力。有的传感器在零点附近有严重的非线性，形成所谓“死区”，则将死区的大小作为阈值；更多情况下，阈值主要取决于传感器噪声的大小，因而有的传感器只给出噪声电平。

⑩ 稳定性：稳定性表示传感器在一个较长的时间内保持其性能参数的能力。理想的情况是不论什么时候，传感器的特性参数都不随时间变化。但实际上，随着时间的推移，大多数传感器的特性会发生改变。这是因为敏感元件或构成传感器的部件，其特性会随时间发生变化，从而影响了传感器的稳定性。稳定性一般以室温条件下经过一段规定时间间隔后，传感器的输出与起始标定时的输出之间的差异来表示，称为稳定性误差。稳定性误差可用相对误差表示，也可用绝对误差来表示。

（2）传感器的动态特性

动态特性是指检测系统的输入为随时间变化的信号时，系统的输出与输入之间的关系。动态特性的性能指标主要有时域单位阶跃响应性能指标和频域频率特性性能指标。

3．传感器的组成

传感器一般由敏感元件、转换元件及基本转换电路三部分组成。

（1）敏感元件：直接感受被测物理量，并以确定关系输出另一物理量的元件（如弹性敏感元件将力、力矩转换为位移或应变输出）。

（2）转换元件：将敏感元件输出的非电量转换成电路参数（电阻、电感、电容）及电流或电压等电信号的元件。

（3）基本转换电路：将电信号转换成便于传输、处理的电量的电路。大多数传感器为开环系统，也有带反馈的闭环系统。

4．传感器的分类

从不同角度可将传感器分为不同类别，最常用的分类方法主要有以下三种。

（1）按用途分类

可分为力敏传感器、位置传感器、液位传感器、能耗传感器、速度传感器、加速度传感器、射线辐射传感器、热敏传感器等。

（2）按原理分类

可分为振动传感器、湿敏传感器、磁敏传感器、气敏传感器、真空度传感器、生物传感器等。

（3）按输出信号分类

模拟量传感器：将被测量的非电量转换成模拟电信号。

数字量传感器：将被测量的非电量转换（包括直接和间接转换）成数字输出信号。

开关量传感器：当一个被测量的信号达到某个特定的阈值时，传感器相应地输出一个设定的低电平或高电平信号。

本项目按输出信号将任务分为开关量采集、模拟量采集、数字量采集，对不同用途的传感器进行学习。

3.3.2 红外传感器工作原理

1．红外感应简介

普通人体会辐射波长在 10μm 左右的红外线，用专门设计的传感器可以针对性地检测这种红外线的存在与否，当人体辐射的红外线照射到传感器上时，因热释电效应将向外释放电荷，后续电路经检测处理后就能产生控制信号。

热释电效应与压电效应类似，是指由于温度的变化而引起晶体表面荷电的现象。热释电传感器是对温度敏感的传感器。它由陶瓷氧化物或压电晶体元件组成，在元件两个表面做成电极，在传感器监测范围内温度有ΔT 的变化时，热释电效应会使两个电极上产生电荷ΔQ，即在两个电极之间产生微弱的电压ΔV。由于输出阻抗极高，在传感器中有一个场效应管进行阻抗变换。热释电效应所产生的电荷ΔQ 会被空气中的离子所结合而消失，即当环境温度稳定不变时，ΔT=0，则传感器无输出。当人体进入检测区时，因人体温度与环境温度有差别，产生温度变化ΔT，则有ΔT 输出；若人体进入检测区后不动，则温度没有变化，传感器也就没有

输出了。所以这种传感器可检测人体或者动物的活动。实验证明，若传感器上不加光学透镜（也称菲涅尔透镜），其检测距离小于 2m；而加上光学透镜后，其检测距离可大于 7m。

红外传感器是利用红外线的物理性质来进行测量的传感器。红外线又称红外光，它具有反射、折射、散射、干涉、吸收等性质。任何物质，只要它本身具有一定的温度（高于绝对零度），都能辐射红外线。用红外传感器进行测量时不与被测物体直接接触，因而不存在摩擦，并且具有灵敏度高、反应快等优点。

红外传感器包括光学系统、检测元件和转换电路。光学系统按结构不同可分为透射式和反射式两类。检测元件按工作原理可分为热敏检测元件和光电检测元件。热敏检测元件应用最多的是热敏电阻。热敏电阻受到红外线辐射时温度升高，电阻发生变化（变化可能是变大也可能是变小，因为热敏电阻可分为正温度系数热敏电阻和负温度系数热敏电阻），通过转换电路转变成电信号输出。光电检测元件常用的是光敏元件，通常由硫化铅、硒化铅、砷化铟、砷化锑、碲镉汞三元合金、锗及硅等材料制成。

红外传感器常用于无接触温度测量、气体成分分析和无损探伤，在医学、军事、空间技术和环境工程等领域得到广泛应用。例如，通过红外传感器远距离测量人体表面温度获得的热像图，可以发现人体温度异常的部位，及时对疾病进行诊断治疗（热像仪）；利用人造卫星上的红外传感器对地球云层进行监视，可实现大范围的天气预报；采用红外传感器可检测飞机上正在运行的发动机的过热情况等；具有红外传感器的望远镜可用于军事行动，在林地战中探测密林中的敌人，在城市战中探测墙后面的敌人。以上例子均利用了红外传感器可以通过测量人体表面温度从而得知人体所在位置的原理。

本书中所用红外传感器为红外光电传感器。

2．红外光电传感器

红外光电传感器是通过把红外光强度的变化转换成电信号的变化来实现控制的。一般情况下，其由三部分构成：发光器、收光器和检测电路，如图 3-10 所示。

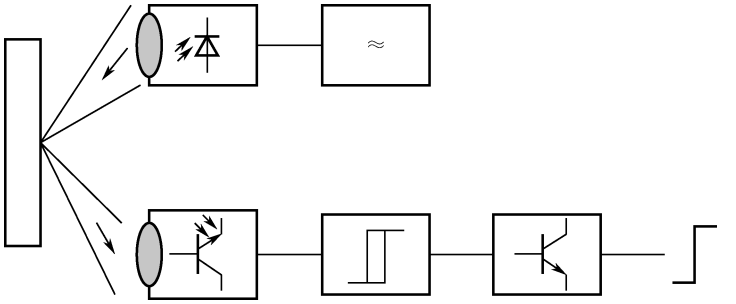


图 3-10　红外光电传感器

3．红外光电传感器的分类和工作方式

（1）槽形光电传感器：把一个发光器和一个收光器面对面地装在一个槽的两侧的是槽形光电传感器。发光器能发出红外光或可见光，在无阻情况下收光器能接收到光。但当被检测物体从槽中通过时，光被遮挡，槽形光电传感器便动作，输出一个开关控制信号，切断或接通负载电流，从而完成一次控制动作。槽形光电传感器的检测距离因为受整体结构的限制一般只有几厘米，如图 3-11 所示。

（2）对射型光电传感器：若把发光器和收光器分离开，就可使检测距离加大。由一个发光器和一个收光器组成的光电传感器就称为对射分离型光电传感器，简称对射型光电传感器。

它的检测距离可达几米乃至几十米，使用时把发光器和收光器分别装在被检测物体通过路径的两侧，被检测物体通过时阻挡光路，收光器就动作，输出一个开关控制信号，如图 3-12 所示。

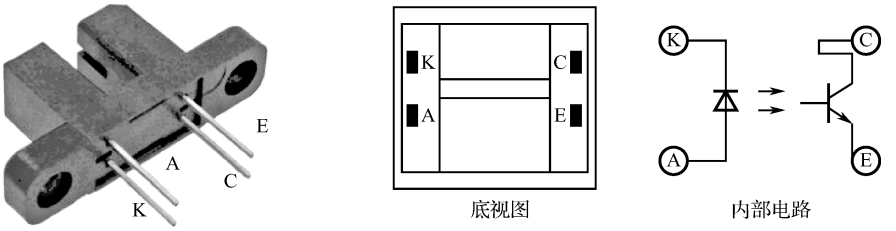


图 3-11　槽形光电传感器

图 3-12　对射型光电传感器

（3）反光板型光电传感器：把发光器和收光器装入同一个装置内，在它的前方装一块反光板，利用反射原理完成光电控制作用的传感器称为反光板型（或反射镜反射型）光电传感器。在正常情况下，发光器发出的光被反光板反射回来被收光器接收到；一旦光路被挡住，收光器接收不到光时，反光板型光电传感器就动作，输出一个开关控制信号，如图 3-13 所示。

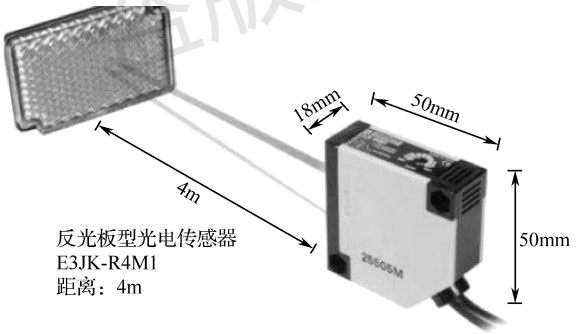


图 3-13　反光板型光电传感器

图 3-14　扩散反射型光电传感器

（4）扩散反射型光电传感器：它的检测头里也装有一个发光器和一个收光器，但前方没有反光板。在正常情况下，发光器发出的光收光器是接收不到的。当被检测物体通过时挡住了光路，并把部分光反射回来，收光器就接收到光信号，输出一个开关控制信号，如图 3-14 所示。

4．本实验使用的传感器介绍

（1）90°脚红外对管：由高功率红外发射二极管（蓝色管）和高灵敏度光电晶体管（黑色管）组成。电压：1.4～1.6V；电流：20mA；脚型：折弯脚 90°，如图 3-15 所示。

（2）槽形光耦：如图 3-16 所示。

图 3-15　90°脚红外对管

图 3-16　槽形光耦

槽形光耦参数指标如下。

① 槽宽：15mm
② 光圈宽度：1.5mm
③ 集电极—发射极最大电压：30V
④ 最大集电极电流：20mA
⑤ 正向电流：20mA
⑥ 安装风格：通孔
⑦ 最高工作温度：+85℃
⑧ 最低工作温度：−25℃
⑨ 封装：大批
⑩ 商标：Lite-On
⑪ 下降时间：4μs
⑫ 功率耗散：100mW
⑬ 上升时间：3μs
⑭ 感应距离：15mm
⑮ 感应方式：透射式，开槽

双孔座 LED 灯：用于指示灯连接，如图 3-17 所示。

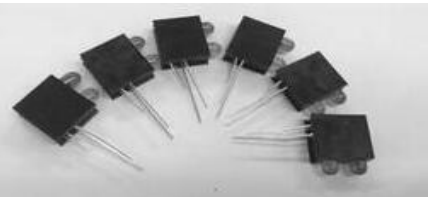


图 3-17　双孔座 LED 灯

（3）本实验所用红外传感模块组成如图 3-18 所示。

说明：

①、②：红外对射型传感器 LTH-301-32 及红外对射传感电路。

③、④：对射输出 1、2 接口 J5、J6，测量红外对射型传感器光敏晶体管输出的电压。

⑤、⑥：红外反射型传感器 ITR20001/T 及红外反射传感电路。

⑦、⑧：发射输出 1、2 接口 J2、J3，测量红外反射传感器光敏晶体管输出的电压，即比较器 1、2 正端（3 脚、5 脚）的输入电压。

⑨：反射 A/D 输出 1、2 接口 J10、J11，测量比较器 1、2 输出端（1 脚、7 脚）的电压。

⑩：接地 GND 接口 J4。

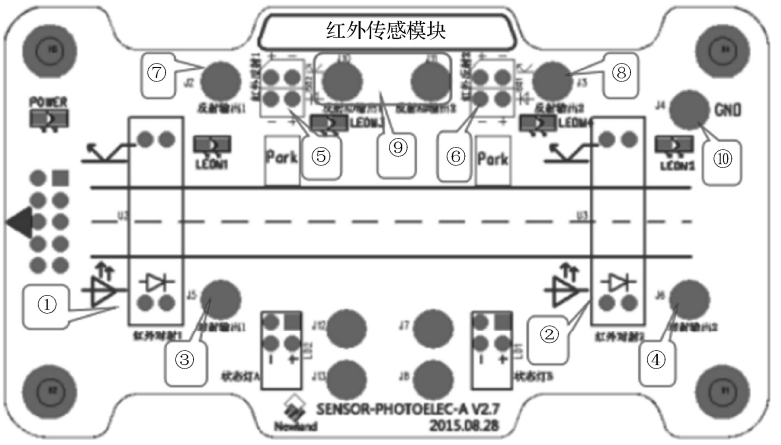


图 3-18　本实验所用红外传感模块组成

3.3.3　声音传感器工作原理

声音传感器相当于一个扬声器。它接收声波，显示声音的振动图像，但不能对噪声的强度进行测量。该传感器内置一个对声音敏感的驻极体电容式扬声器，声波使驻极体薄膜振动，导致电容变化，而产生与之对应的微小电压。这一电压随后被转化成 0～5V 的电压，经过 A/D 转换被数据采集器接收，并传送给计算机。

按照结构不同，可将声音传感器分为驻极体电容式声音传感器和压电驻极体声音传感器两类，下面分别进行简单介绍。

1. 驻极体电容式声音传感器

（1）驻极体电容式声音传感器的特点

驻极体电容式声音传感器分为振膜式驻极体电容式声音传感器和背极式驻极体电容式声音传感器。背极式驻极体电容式声音传感器的薄膜与驻极体材料能各自发挥其特长，因此性能比振膜式驻极体电容式声音传感器好。其结构如图 3-19 所示。

（2）驻极体电容式声音传感器的性能

不同型号的声音传感器，其响应指标也不同，市场上常见的几种类型的参数表如表 3-4 所示。

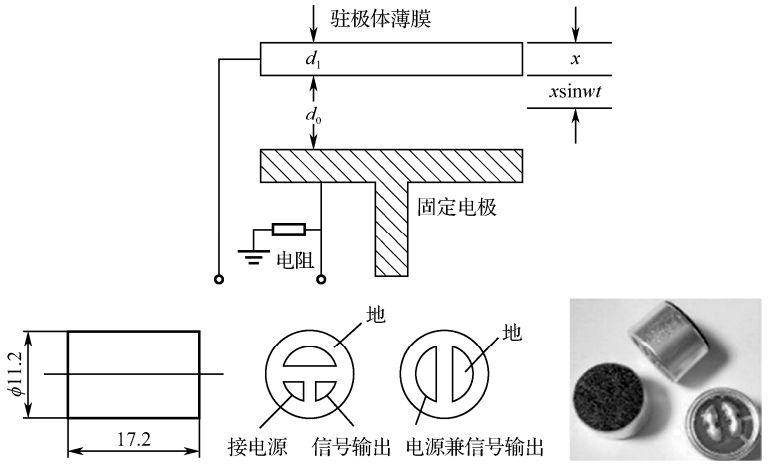


图 3-19　背极式驻极体电容式声音传感器结构

表 3-4　驻极体电容式声音传感器（电压输出型）参数表

型　　号	频率范围	灵敏度	响应类型	动态范围	外形尺寸
	±2db/Hz	mV・Pa^{-1}		db	直径/mm
CHZ-11	3～18k	50	自由场	12～146	23.77
CHZ-12	4～8k	50	声场	10～146	23.77
CHZ-11T	4～16k	100	自由场	5～100	20
CHZ-13	4～20k	50	自由场	15～146	12
CHZ-14A	4～20k	12.5	声场	15～146	12
HY205	2～18k	50	声场	40～160	12.7
4175	5～12.5k	50	自由场	16～132	2642
BF5032P	70～20000	5	自由场	20～135	49
CZ II-60	40～12000	100	自由场/声场	34	9.7

2．压电驻极体声音传感器

压电驻极体声音传感器利用压电效应进行声电/电声转换，其声电/电声转换器为一片 30～80μm 厚的多孔聚合物压电驻极体薄膜，相对电容式/动圈式结构复杂且对精度要求极高的零件配合设计，大大减小了电声器件的体积；同时，零件数目大为减少，可靠性得到保证，方便大规模生产。多孔聚合物压电驻极体薄膜能达到非常高的压电系数，比 PVDF 铁电聚合物及其共聚物的压电活性高 1 个量级；多孔聚合物压电驻极体薄膜的厚度可以做到很小，易于满足对几何尺寸的要求，且原料来源广泛，材料成本与加工制备均较压电陶瓷与铁电单晶材料容易许多。利用压电驻极体制成的声音传感器，可广泛应用于电声、水声、超声与医疗等领域。

压电驻极体声音传感器示意图及其结构图分别如图 3-20、图 3-21 所示。

图 3-20　压电驻极体声音传感器示意图

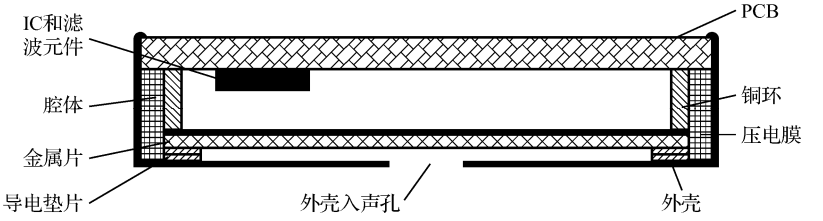


图 3-21　压电驻极体声音传感器结构图

3．本实验使用的声音传感模块组成

本实验使用的声音传感模块组成如图 3-22 所示。

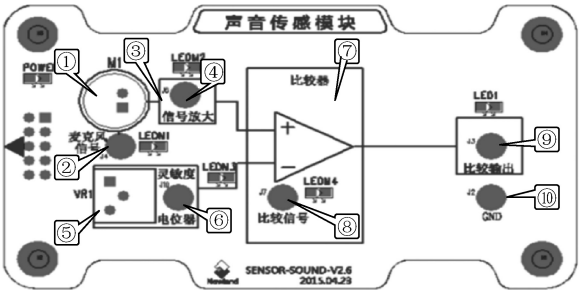


图 3-22　本实验使用的声音传感模块组成

说明：

①：MP9767P。

②：扬声器信号接口 J4，测试扬声器输出的音频信号。

③：信号放大电路。

④：信号放大接口 J6，测量音频信号经过放大后叠加在直流电平上的信号，即比较器 1 的负端输入电压。

⑤：灵敏度调节电位器。

⑥：灵敏度测试接口 J10，测试可调电阻可调端输出电压，即比较器 1 的正端输入电压。

⑦：比较器电路。

⑧：比较信号测试接口 J7，即比较器 1 的输出电压。

⑨：比较输出测试接口 J3，即比较器 2 的输出电压。

⑩：接地 GND 接口 J2。

3.3.4　任务实训步骤

第 1 步，搭建硬件环境，连接各模块。

开关量传感器采集系统按照如图 3-23 所示进行连线，具体步骤如下。

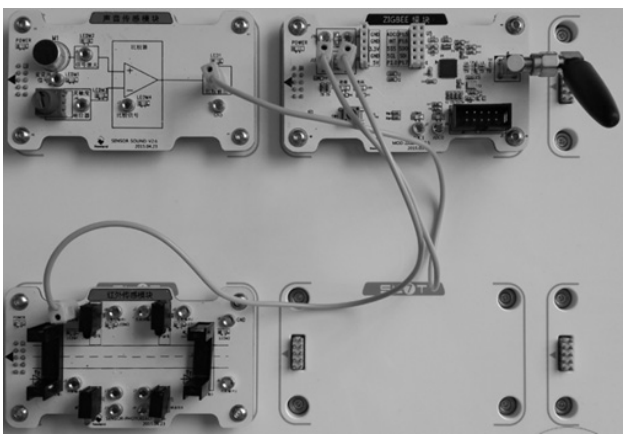


图 3-23　开关量传感器采集系统连线图

（1）将 ZigBee 模块、声音传感模块和红外传感模块置于 NEWLab 平台上。

（2）将红外传感模块的对射输出 1 接口（J5）与 ZigBee 模块的 IN1（J12/P1.4）接口相连。

（3）将声音传感模块的比较输出测试接口（J3）与 ZigBee 模块的 IN0（J13/P1.3）接口相连。

第 2 步，新建工程和程序文件，添加头文件。

新建工程的方法与过程参照本项目的任务 3.2，头文件应添加 basicrf、board、common、utils 4 组。

第 3 步，配置工程。

配置方法参照本项目的任务 3.2。

第 4 步，编写程序。

编写主程序，或将技术服务网站上相应的源代码添加到工程中。下面将主程序 Kaiguan_Sensor.c 中的关键代码分析如下。

（1）点对点通信地址设置

```
/**********************************点对点通信地址设置*****************************/
#define RF_CHANNEL      20          //通信信道号 11～26
#define PAN_ID          0x1379      //网络 ID
#define MY_ADDR         0x1234      //本机 1 号模块地址
#define SEND_ADDR       0x5678      //发送 1 号模块地址
/*********************************************************************************/
```

（2）main()函数

```
/*****************************************main*************************************/
void main(void)
{uint8 sensor_val;
halBoardInit();                     //模块相关资源的初始化
//   ConfigRf_Init();               //无线收、发模块参数的配置初始化
port1->port = 1;
port1->pin = 0x03;
port1->pin_bm = 0x08;
port1->dir = 0;
halDigioConfig(port1);
halDigioIntEnable(port1);
halDigioIntConnect(port1, port13Int);
while(1)
{ sensor_val=get_swsensor();        //读取开关量，即 P1.3 引脚状态
if(sensor_val)                      //红外传感模块
{
halLedSet(2);                       //点亮 LED2 灯
}
else
{
halLedClear(2);                     //熄灭 LED2 灯
}
if(SY_flag)                         //声音传感模块
{
SY_flag = 0x00;
halLedSet(1);                       //点亮 LED1 灯
halMcuWaitMs(30000);                //延时 30s
```

```
    halLedClear(1);                          //熄灭 LED1 灯
  }
  }
  }
  /*****************************************************************************/
```

第 5 步，下载程序、运行。

编译无误后，把程序下载到 ZigBee 模块中。

（1）将一个物体放到“红外对射 1”元件的槽中，发现 ZigBee 模块中的 LED2 灯立刻被点亮；当物体离开槽后，LED2 灯立刻熄灭。

（2）再拍手制造响声，ZigBee 模块中的 LED1 灯立刻亮起来，并且维持 2s 亮的状态，2s 后 LED1 灯自动熄灭。注意：可以调节电位器，设置触发阈值电压。

任务 3.4　模拟量传感器采集系统

【任务描述】

采用气体传感模块、温度/光照传感模块，以及 ZigBee 模块组成一个模拟量传感器采集系统。把带酒精的棉签靠近气体传感模块，使用手电筒照射温度/光照传感模块，当气体传感模块检测到不同浓度的气体时，温度/光照传感模块检测到不同强度的光照时，都会在计算机的串口调试软件上显示检测到的气体电压信息和光照电压信息。

【任务环境】

硬件：NEWLab 平台 1 套、ZigBee 节点板 3 块、气体传感模块 1 个、温度/光照传感模块 1 个、CC2530 仿真器 1 组、PC 1 台，信号线若干。

软件：Windows 7/10，IAR 集成开发环境，串口调试助手。

【必备知识点】

1．气体传感器工作原理；

2．光照传感器工作原理。

3.4.1　气体传感器工作原理

气体传感器是一种把气体中的特定成分检测出来，并把它转换为电信号的器件。它具有结构简单，使用方便，性能稳定、可靠，灵敏度高等诸多优点。按照结构特性，气体传感器一般可以分为以下几种：半导体型气体传感器、电化学型气体传感器、固体电解质气体传感器、接触燃烧式气体传感器、光化学型气体传感器、高分子气体传感器、红外吸收式气体传感器等。

本任务重点介绍半导体型气体传感器和红外吸收式气体传感器。

1．半导体型气体传感器

半导体型气体传感器的工作原理：传感器与气体相互作用产生表面吸附或反应，引起以载流子运动为特征的电导率或伏安特性或表面电位的变化，以此来检测特定气体的成分或者测量其浓度，并将其变换成电信号输出。

半导体型气体传感器又分为电阻式和非电阻式两种，不同材质制成的传感器可检测的气体类型也有所不同，如表 3-5 所示。

表 3-5　半导体型气体传感器的分类

	主要物理特性		工 作 温 度	典型被测气体
电阻式	电阻	表面控制型	室温～450℃	可燃性气体
		体控制型	700℃以上	酒精、氧气、其他可燃性气体
非电阻式	表面电位	氧化银	室温	硫醇
	二极管整流特性	铂/硫化镉、铂/氧化钛	室温～200℃	氢气、一氧化碳、酒精
	晶体管特性	铂栅 MOS 场效应管	150℃	氢气、硫化氢

（1）可燃气体/烟雾传感器

MQ-2 可燃气体/烟雾传感器使用二氧化锡半导体气敏材料，属于表面离子式 N 型半导体。当温度处于 200℃～300℃时，二氧化锡吸附空气中的氧，形成氧的负离子吸附，使半导体中的电子密度减少，从而使其电阻值增加。当其与烟雾接触时，如果晶粒间界处的势垒受到该烟雾的调制而变化，就会引起表面电导率的变化。利用这一点就可以获得这种烟雾存在的信息，烟雾浓度越大，电导率越大，输出电阻值越小。使用简单的电路即可将电导率的变化转换为与该气体浓度相对应的输出信号。

MQ-2 可燃气体/烟雾传感器对液化气、丙烷、氢气的灵敏度高，对天然气和其他可燃气体的检测也很理想。这种传感器可检测多种可燃性气体，是一款适合多种应用的低成本传感器。

（2）酒精传感器

使用酒精传感器可对被测人呼出气体进行检测。血液中的酒精含量越高，呼出气体中的酒精含量越高，检测到的信号越大。按照国际通用标准，呼出气体中的酒精含量是血液中酒精含量的 2100 倍，由此，根据检测到的呼出气体中的酒精含量就可以得出血液中的酒精含量。目前各国交通执法的快速血液酒精检测，均使用此方法。随着科技发展，酒精传感器的研制已经相当成熟，目前有电化学酒精传感器、半导体酒精传感器、催化燃烧酒精传感器等。

催化燃烧酒精传感器功耗大，漂移比较多，气体选择性差，所以不便制造为便携式仪器，目前很少使用催化燃烧酒精传感器。

电化学酒精传感器选择性好，稳定性好，功耗低，但造价高，同时应用成本高，一般使用在专业测试仪器上，如专业交警执法使用仪器、工业特殊作业场所检查仪器、工业检测仪器等。

半导体酒精传感器是近几年研究人员研制出来的新型酒精传感器，它的特点介于上述两者之间，有功耗低，稳定性好，响应速度快，生产成本相对较低，适合于大量生产的特点。因此使用半导体酒精传感器来制造司机个人用酒精检测仪成为首选。

酒精传感器一般有 3 个引脚，两侧的是加热电极，中间的是检测电极，从中间检测电极到任意两个加热电极间的电阻值都与酒精的浓度有关，因此检测这个电阻值就可以检测酒精的浓度。

由于检测电极与加热电极之间是电气连通的，受加热电极上电压的影响，需要从检测电极连接一个检测电阻到任意一个加热电极上，检测电极上的电压值即为传感器输出。

（3）空气质量传感器

空气质量传感器所使用的气敏材料是在清洁空气中电导率较低的二氧化锡（SnO_2）。当空气质量传感器所处环境中存在被污染气体时，其电导率随空气中被污染气体浓度的增大而增大。使用简单的电路即可将电导率的变化转换为与该气体浓度相对应的输出信号。

空气质量传感器（MQ-135）对氨气、硫化物、苯系气体的灵敏度高，对烟雾和其他有害气体的监测也很理想。这种传感器可检测多种有害气体，是一款适合多种应用的低成本传感器。

2．红外吸收式气体传感器

红外吸收式气体传感器的工作原理：一束红外光的强度在通过一个气体容器时会减小，而光强度损失是一定体积内活动气体分子数量的函数，用来表示气体浓度。不同气体的红外波长各不相同，部分典型气体的特征红外吸收波长如表 3-6 所示。

表 3-6　部分典型气体的特征红外吸收波长

气　体	特征红外吸收波长/μm	气　体	特征红外吸收波长/μm
CO	4.65	SO_2	7.3
CO_2	2.7，4.24，14.5	NH_3	2.3，2.8，6.1，9
CH_4	2.4，3.3，7.65	H_2S	7.6
NO	5.3	HCl	3.4
NO_2	6.13	HCN	3，6.25，16.6
N_2O	4.53	HBr	4

当强度为 I_0 的入射红外光穿过气体时，气体吸收自己特征频率红外光的能量，从而使出射光强度减弱为 I，即：

$$I = I_0 e^{(-\mu CL)} \tag{3-2}$$

式中，μ——气体吸收系数；

C——待测气体浓度；

L——光程长度。

这种红外吸收式气体传感器具有选择性好、不易受有害气体的影响而中毒或老化、响应速度快、稳定性好、防爆性好、信噪比高、使用寿命长、测量精度高、应用范围广等优点。

3．本实验使用的气体传感模块组成

本实验使用的气体传感模块组成如图 3-24 所示。

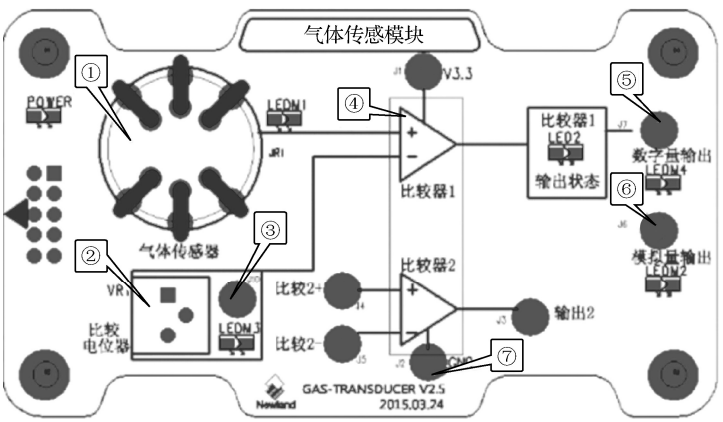


图 3-24　本实验使用的气体传感模块组成

说明：

①：MQ-2 可燃气体/烟雾传感器。

②：灵敏度调节电位器。

③：灵敏度测试接口 J10，测试有害气体浓度阈值电压，即比较器 1 负端（3 脚）电压。

④：比较器电路。

⑤：数字量输出接口 J7，测试比较器 1 输出电压。

⑥：模拟量输出接口 J6，测试气体传感器感应电压，即比较器 1 正端电压。

⑦：接地 GND 接口 J2。

3.4.2　光照传感器工作原理

光照传感器是将光通量转换为电量的一种传感器，它的理论基础是光电效应。

1．光电效应

光可以认为是由具有一定能量的粒子（一般称为光子）所组成的，而每个光子所具有的能量 E 与其频率大小成正比。光照射在物体表面上就可以看成物体受到一连串能量为 E 的光子轰击，而光电效应就是该物体吸收到能量为 E 的光子后产生的电效应。通常把光照射到物体表面后产生的光电效应分为三类。

（1）外光电效应。在光线作用下能使电子逸出物体表面的称为外光电效应。例如，光电管、光电倍增管等就是基于外光电效应制成的光电器件。

（2）内光电效应。在光线作用下能使物体电阻率发生改变的称为内光电效应，又称为光电导效应。例如，光敏电阻就是基于内光电效应制成的光电器件。

（3）半导体光生伏特效应。在光线作用下能使物体产生一定方向电动势的称为半导体光生伏特效应。例如，光电池、光敏晶体管就是基于半导体光生伏特效应制成的光电器件。

基于外光电效应制成的光电器件属于真空光电器件，基于内光电效应和半导体光生伏特效应制成的光电器件属于半导体光电器件。

2．光电器件的工作原理

（1）光电二极管

光电二极管（也称光敏二极管）和普通二极管相比虽然都属于单向导电的非线性半导体器件，但在结构上有其特殊的地方，光电二极管是基于半导体光生伏特效应的光电器件。光电二极管的符号如图 3-25 所示。光电二极管在电路中一般处于反向接入状态，即正极接电源负极，负极接电源正极，如图 3-26 所示。

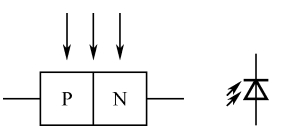


图 3-25　光电二极管的符号

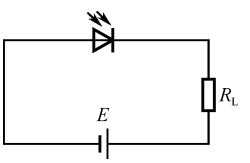


图 3-26　光电二极管在电路中的接法

在没有光照时，光电二极管的反向电阻很大，反向电流很微弱，称为暗电流。当有光照时，光子打在 PN 结附近，于是在 PN 结附近产生电子–空穴对，它们在 PN 结内部电场作用下做定向运动，形成光电流。光照越强，光电流越大。所以，在不受光照射时，光电二极管处于截止状态；受到光照射时，二极管处于导通状态。

（2）光电晶体管

光电晶体管（也称光敏晶体管）如图 3-27 和图 3-28 所示，它和普通晶体管相似，也有电流放大作用，只是它的集电极电流不只受基极电路和电流控制，也受光辐射的控制。通常基极不引出，有一些光电晶体管的基极有引出，用于温度补偿和附加控制等。当具有光敏特性的 PN 结受到光辐射时，形成光电流，光电流由基极进入发射极，从而在集电极回路中得到一个放大了相当于β倍的信号电流。不同材料制成的光电晶体管具有不同的光谱特性，与光电二极管相比，光电晶体管具有很大的光电流放大作用，以及更高的灵敏度。

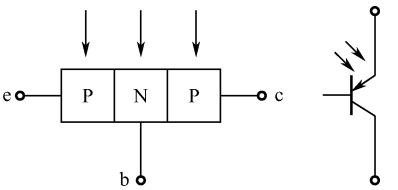


图 3-27　PNP 型光电晶体管

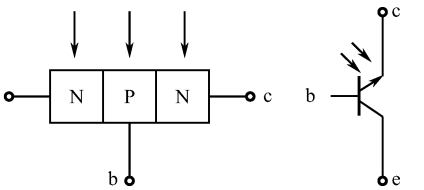
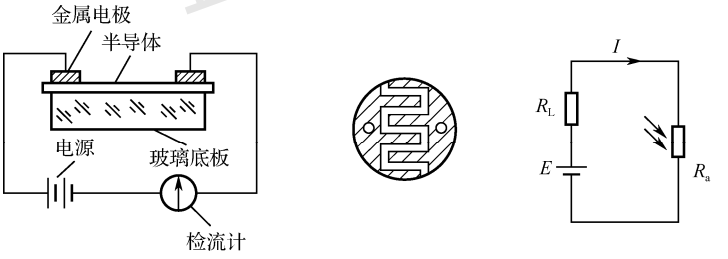


图 3-28　NPN 型光电晶体管

3．光敏电阻的工作原理

光敏电阻是利用半导体的光电效应制成的一种电阻值随入射光的变化而改变的电阻。入射光变强，电阻值减小；入射光变弱，电阻值增大。光敏电阻一般用于光的测量、光的控制和光电转换（将光的变化转换为电的变化）。常用的光敏电阻是硫化镉光敏电阻，它是由半导体材料制成的。光敏电阻的阻值随入射光线（可见光）的强弱变化而变化，在黑暗条件下，它的阻值（暗阻）可达 1～10MΩ；在强光（100lx）条件下，它的阻值（亮阻）仅有几百至数千欧姆。光敏电阻对光的敏感性（即光谱特性）与人眼对可见光（0.4～0.76μm）的响应很接近，只要人眼可感受的光，都会引起它的阻值变化。光敏电阻的结构、电极、接线图如图 3-29 所示。



（a）光敏电阻结构　　（b）光敏电阻电极　　（c）光敏电阻接线图

图 3-29　光敏电阻的结构、电极、接线图

4．本实验使用的温度/光照传感模块组成

本实验所用温度/光照传感模块中，光照传感器与温度传感器合为一体，如图 3-30 所示。

说明：

①：温度和光照传感器。

②：基准电压调节电位器。

③：比较器电路。

④：基准电压测试接口 J10，测试温度感应的阈值电压，即比较器 1 负端（3 脚）电压。

⑤：模拟量输出接口 J6，测试热敏电阻两端的电压，即比较器 1 正端（2 脚）电压。

⑥：数字量输出接口 J7，测试比较器 1 输出电压。

⑦：接地 GND 接口 J2。

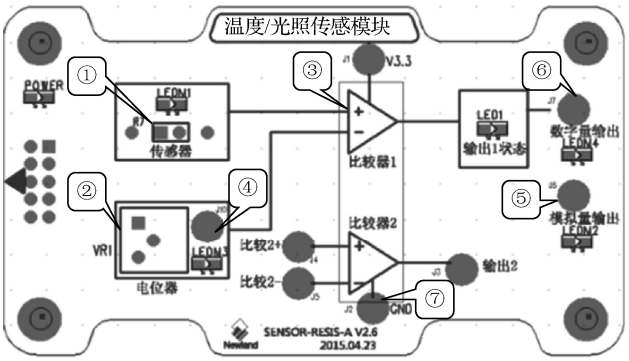


图 3-30　温度/光照传感模块组成

3.4.3　任务实训步骤

第 1 步，搭建硬件环境，连接各模块。

（1）组成光照传感器采集系统（温度/光照传感模块）

把 ZigBee 模块和温度/光照传感模块固定在 NEWLab 平台上，将温度/光照传感模块的模拟量输出接口与 ZigBee 模块的 ADC0（P0_0）接口连接起来。

（2）组成气体传感器采集系统（气体传感模块）

把 ZigBee 模块和气体传感模块固定在 NEWLab 平台上，将气体传感模块的模拟量输出接口与 ZigBee 模块的 ADC0 接口连接起来。

（3）组成模拟量集中采集系统（协调器模块）

将协调器模块通过串口线连接到 PC 串口或者通过 USB 转串口线连接到 PC，并给协调器通电。各模块连接效果如图 3-31 所示。

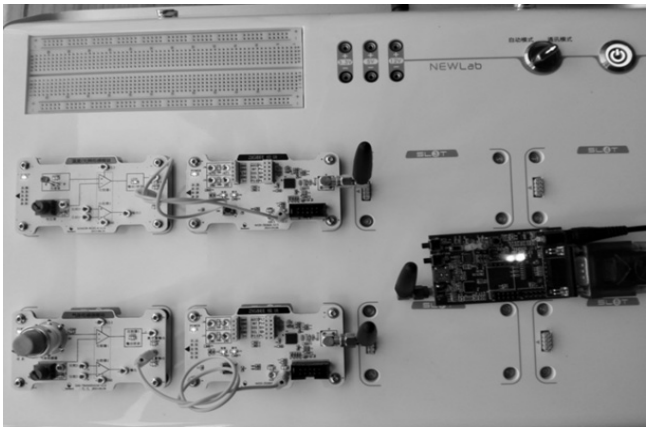


图 3-31　各模块连接效果

第 2 步，新建工程和程序文件。

新建工程的方法与过程参照本项目的任务 3.2。

第 3 步，编写程序。

编写主程序，或将技术服务网站上相应的源代码添加到工程中。下面将 sensor.c 和 collect.c 两个文件中的关键代码分别分析如下。

（1）sensor.c 中的 main()函数

```
/*************************************main*****************************************/
void main(void)
{ uint16 sensor_val;
uint16   len = 0;
halBoardInit();                          //模块相关资源的初始化
ConfigRf_Init();                         //无线收、发模块参数的配置初始化
halLedSet(1);
halLedSet(2);
Timer4_Init();                           //定时器初始化
Timer4_On();                             //打开定时器
while(1)
{    APP_SEND_DATA_FLAG = GetSendDataFlag();
if(APP_SEND_DATA_FLAG == 1)              //定时时间到
{    /*【传感器采集、处理】 开始*/
#if defined (GM_SENSOR)                  //光照传感器
sensor_val=get_adc();                    //取模拟电压
//把采集的数据转化成字符串，以便于在串口上显示、观察
printf_str(pTxData,"光照传感器电压：%d.%02dV\r\n",sensor_val/100,sensor_val%100);
#endif
#if defined (QT_SENSOR)                  //气体传感器
sensor_val=get_adc();                    //取模拟电压
//把采集的数据转化成字符串，以便于在串口上显示、观察
printf_str(pTxData,"气体传感器电压：%d.%02dV\r\n",sensor_val/100,sensor_val%100);
#endif
halLedToggle(3);                         // 绿灯取反，无线发送指示
//把数据通过 ZigBee 模块发送出去
basicRfSendPacket(SEND_ADDR, pTxData,strlen(pTxData ));
Timer4_On();  //打开定时
}  /*【传感器采集、处理】 结束*/
}
}
/*************************************main end ***********************************/
```

（2）collect.c 中的 main()函数

```
/*************************************main*****************************************/
void main(void)
{ uint16 len = 0;
halBoardInit();                          //模块相关资源的初始化
ConfigRf_Init();                         //无线收、发模块参数的配置初始化
halLedSet(1);
halLedSet(2);
while(1)
{ if(basicRfPacketIsReady())             //查询有无收到无线数据
{ halLedToggle(4);                       //红灯取反，无线接收指示
//接收无线数据
len = basicRfReceive(pRxData, MAX_RECV_BUF_LEN, NULL);
//把接收到的无线数据发送到串口
halUartWrite(pRxData,len);
```

```
14.  }
15.  }
16.  }
     /*************************************main end ************************************/
```

第 4 步，建立与配置模块。

1）建立与配置温度/光照传感模块

（1）建立模块

选择“Project”→“Edit Configurations”命令，弹出项目配置对话框，如图 3-32 所示，系统会检测出项目中存在的模块。

单击“New”按钮，在弹出的对话框中输入模块名称“gm_sensor”，基于“Deubg”模块进行配置，然后单击“OK”按钮完成模块的建立，如图 3-33 所示。在项目配置对话框中就可以自动检测到刚才建立的模块“gm_sensor”。

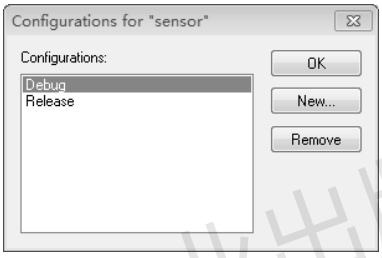


图 3-32　项目配置对话框

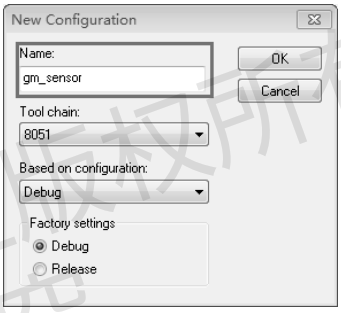


图 3-33　建立模块

（2）设置“Options”

为了给模块设置对应的条件编译参数，在此需要进行如下设置：在项目工作组中选择“gm_sensor”模块，单击鼠标右键选择“Options”，在弹出的对话框中选择“C/C++ Compiler”类别，在右边的窗口中“Preprocessor”选项卡中的“Defined symbols”栏中输入“GM_SENSOR”。具体设置如图 3-34 所示。

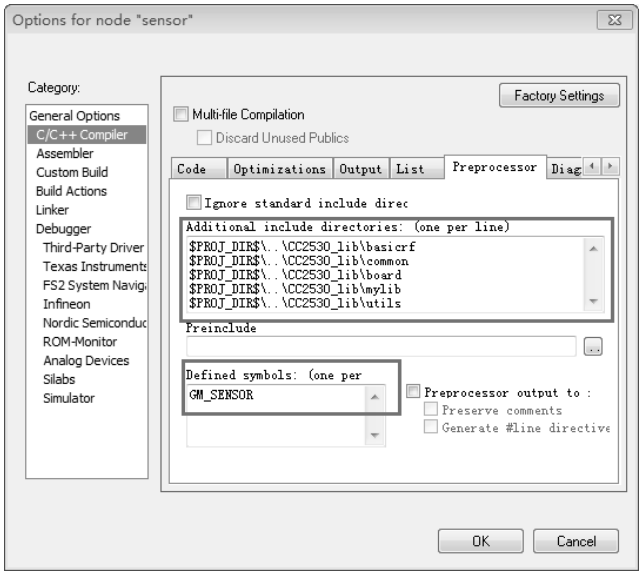


图 3-34　“Options”设置

2）建立与配置气体传感模块设备

操作步骤与建立温度/光照传感模块设备一样，只需要将模块设备名称与模块“Options”分别设置为“qt_sensor”与“QT_SENSOR”。

3）建立与配置协调器模块

此操作步骤与建立温度/光照传感模块一样，需要将模块名称设置为“collect”，并修改“Options”设置。

第 5 步，给各工作组下载程序。

（1）为温度/光照传感模块下载程序。

在 IAR 软件的“Workspace”栏内，选择“gm_sensor”模块，选中“collect.c”，单击鼠标右键，选择“Options”，在弹出的对话框中将“Exclude from build”复选框打“√”，然后单击“OK”按钮。重新编译程序无误后，给 NEWLab 平台上电，下载程序到 ZigBee 模块中。

（2）为气体传感模块下载程序。

在 IAR 软件的“Workspace”栏内，选择“qt_sensor”模块，选中“collect.c”，单击鼠标右键，选择“Options”，在弹出的对话框中将“Exclude from build”复选框打“√”，然后单击“OK”按钮。重新编译程序无误后，给 NEWLab 平台上电，下载程序到 ZigBee 模块中。

（3）为协调器模块下载程序。

在 IAR 软件的“Workspace”栏内，选择“collect”模块，选择“sensor.c”，单击鼠标右键，选择“Options”，在弹出的对话框中将“Exclude from build”复选框打“√”，然后单击“OK”按钮。重新编译程序无误后，将协调器模块通过串口线连接到 PC 串口或者通过 USB 转串口线连接到 PC，给协调器通电，下载程序到协调器模块中。

第 6 步，运行程序。

（1）将 NEWLab 平台的通信模块开关旋转到通信模式，给 NEWLab 平台上电。

（2）打开串口调试软件，把串口的波特率设置为 38400bps。根据光敏及气体浓度的不同，在 PC 的串口调试终端上显示不同的光照传感器与气体传感器电压信息。运行效果如图 3-35 所示。

图 3-35　串口调试终端运行效果

任务 3.5　数字量传感器采集系统

【任务描述】

采用温湿度传感模块和 ZigBee 模块组成一个数字量传感器采集系统，实现温湿度数据的采集和无线传输，并在 PC 串口上显示。

【任务环境】

硬件：NEWLab 平台 1 套、ZigBee 节点板 1 块、温湿度传感模块 1 个、CC2530 仿真器 1 组、PC 1 台，信号线若干。

软件：Windows 7/10，IAR 集成开发环境，串口调试助手。

【必备知识点】

1．数字量传感器技术；

2．温度传感器工作原理；

3．湿度传感器工作原理。

3.5.1　数字量传感器技术

数字量传感器是指将传统的模拟量传感器经过加装或改造 A/D 转换模块，使之输出信号为数字量（或数字编码）的传感器，主要包括放大器、A/D 转换器、微处理器（CPU）、存储器、通信接口、温度测试电路等。在微处理器应用变得越来越普遍的今天，全自动或半自动（通过人工指令进行高层次操作，自动处理低层次操作）系统可以包含更多智能化功能，能从环境中获得并处理更多不同的参数，尤其是 MEMS（微电机系统）技术，可使数字量传感器的体积非常小且能耗与成本也很低。以纳米碳管或其他纳米材料制成的纳米传感器同样具有巨大的潜力。

发展到今天，数字量传感器与传统的模拟量传感器相比，具有如下几个特点。

（1）先进的 A/D 转换技术和智能滤波算法，在满量程的情况下仍可保证输出的稳定。

（2）可行的数据存储技术，保证模块参数不会丢失。

（3）良好的电磁兼容性能。

（4）采用数字化误差补偿技术和高度集成化电子元件，实现线性度、阈值、温度漂移、蠕变等性能参数的综合补偿，消除了人为因素对补偿的影响，大大提高了综合精度和可靠性。

（5）输出一致性误差可以控制在 0.02%以内甚至更低，特性参数可完全相同，因而具有良好的互换性。

（6）采用 A/D 转换电路、数字化信号传输和数字滤波技术，抗干扰能力强，信号传输距离远，提高了稳定性。

（7）能自动采集数据并可预处理、存储和记忆，具有唯一标记，便于故障诊断。

（8）采用标准的数字通信接口，可直接连入计算机，也可与标准工业控制总线连接，方便灵活。

（9）数字量传感器是将 A/D、EPROM、DIE（还未封装的传感器芯片，属于裸片，大小介于 cell 和 chip 之间），封装在一块 PCB、金属块或陶瓷板上进行集成，通过各种温度、压力点的校准，计算出 DIE 的线性度，再利用 A/D 去补偿的方法加工而成的。

3.5.2 温度传感器工作原理

热电传感技术是利用转换元件电参数随温度变化的特征，对温度和与温度有关的参数进行检测的技术。利用热电传感技术制成的传感器称为温度传感器。其中，将温度变化转化为电阻变化的称为热电阻传感器，其中金属热电阻传感器简称为金属热电阻或热电阻，半导体热电阻传感器简称为热敏电阻；将温度变化转换为热电势变化的称为热电偶传感器。本书只介绍热敏电阻。

热敏电阻是一种阻值随温度变化的半导体。它的温度系数很大，比温差电偶和线绕电阻测温元件的灵敏度高几十倍，适用于测量微小的温度变化。热敏电阻体积小、热容量小、响应速度快，能在空隙和狭缝中测量。它的阻值高，测量结果受引线的影响小，可用于远距离测量。它的过载能力强，成本低。但热敏电阻的阻值与温度为非线性关系，所以它只能在较窄的范围内用于精确测量。热敏电阻在一些对精度要求不高的测量和控制装置中得到了广泛应用。

1．热敏电阻的结构形式

用热敏电阻制成的探头有珠状、棒杆状、片状和薄膜等形式，封装外壳多为用玻璃、镍和不锈钢管等做成的套管结构，如图 3-36 所示为热敏电阻的结构图，如图 3-37 所示为热敏电阻的实物图。

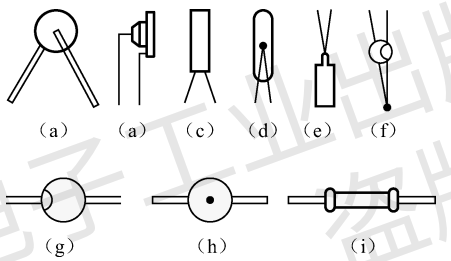


图 3-36　热敏电阻的结构图

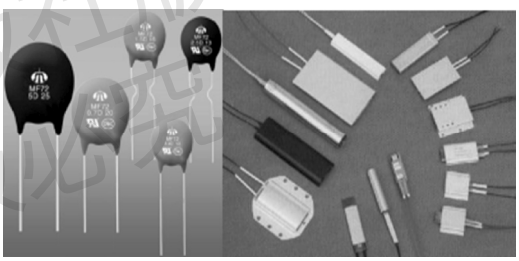


图 3-37　热敏电阻的实物图

2．热敏电阻的温度特性

热敏电阻的温度特性是指半导体材料的阻值随温度变化而变化的特性。热敏电阻按温度特性分为三类。

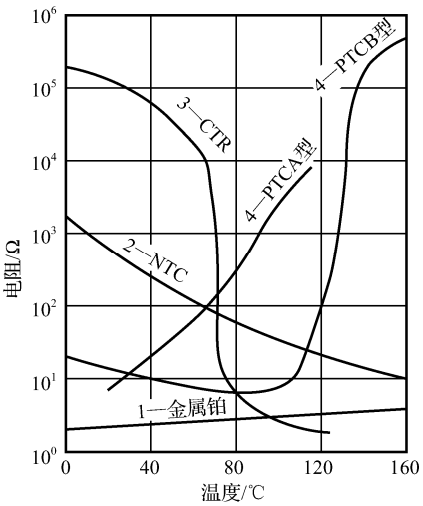


图 3-38　热敏电阻的温度特性曲线

（1）负温度系数（NTC）热敏电阻；

（2）正温度系数（PTC）热敏电阻；

（3）临界负温度系数（CTR）热敏电阻。

热敏电阻的温度特性曲线如图 3-38 所示。分析热敏电阻的温度特性曲线图可以得出下列结论。

（1）热敏电阻的温度系数值远远大于金属热电阻，所以其灵敏度很高。

（2）热敏电阻温度曲线非线性现象十分严重，所以其测量温度范围远小于金属热电阻。

（1）正温度系数（PTC）热敏电阻

PTC 是 Positive Temperature Coefficient 的缩写，意思是正的温度系数，泛指正温度系数很大的半导体材料或元器件。PTC 热敏电阻是一种典型的具有温度敏感性

的半导体电阻，超过一定的温度（居里温度）时，它的阻值随着温度的升高呈阶跃性增高。该材料是以 $BaTiO_3$ 或 $PbTiO_3$ 为主要成分的烧结体，其中掺入微量的 Nb、Ta、Bl、Sb、La 等氧化物进行原子价控制而使之半导体化，常将这种半导体化的 $BaTiO_3$ 等材料简称为半导（体）瓷。同时还可添加增大其正温度系数的 Mn、Fe、Cu、Cr 的氧化物和起其他作用的添加物，采用一般陶瓷工艺成形、高温烧结而使钛酸铂等及其固溶体半导体化，从而得到正特性的热敏电阻材料。其温度系数及居里温度随成分及烧结条件（尤其是冷却温度）不同而不同。

PTC 热敏电阻除用作加热元件，还能起到“开关”的作用（如图 3-38 中线 4 所示，PTCB 型热敏电阻的阻值随着温度的升高初始没有什么变化，在温度达到 100℃左右时突然快速增大），兼有敏感元件、加热器和开关三种功能，又称为“热敏开关”。电流通过元件后引起温度升高，即发热体的温度上升，当超过居里温度后，阻值增加，从而限制电流的增加，于是电流的下降又导致元件温度降低，阻值的减小又使电路电流增加，元件温度升高，周而复始，因此具有使温度保持在特定范围的功能，起到开关作用。利用这种阻温特性做成加热源，作为加热元件应用的有暖风器、烘衣柜、空调等，其还可对电器起到过热保护作用。

（2）负温度系数（NTC）热敏电阻

NTC（Negative Temperature Coefficient），意思是负的温度系数，泛指负温度系数很大的半导体材料或元器件。NTC 热敏电阻是一种典型的具有温度敏感性的半导体电阻，它的阻值随着温度的升高呈线性减小（如图 3-38 线 2 所示）。NTC 热敏电阻是以锰、钴、镍和铜等的金属氧化物为主要材料，采用陶瓷工艺制造而成的。这些金属氧化物材料都具有半导体性质，在导电方式上完全类似于锗、硅等半导体材料。温度低时，这些金属氧化物材料的载流子（电子和空穴）数目少，所以其阻值较高；随着温度的升高，载流子数目增加，阻值降低。

NTC 热敏电阻特性：NTC 热敏电阻的阻值随温度升高而迅速减小。

（3）临界负温度系数（CTR）热敏电阻

CTR 热敏电阻具有负电阻突变特性（如图 3-38 线 3 所示），在某一温度下，其阻值随温度的升高急剧减小，具有很大的负温度系数。其构成材料是钒、钡、锶、磷等元素的氧化物的混合烧结体，是半玻璃状的半导体，也称 CTR 热敏电阻为玻璃态热敏电阻，骤变温度随所添加氧化物的不同而不同。CTR 能够用作控温报警等。

3．本实验所用温度/光照传感模块组成

本实验所用温度/光照传感模块组成如图 3-39 所示。

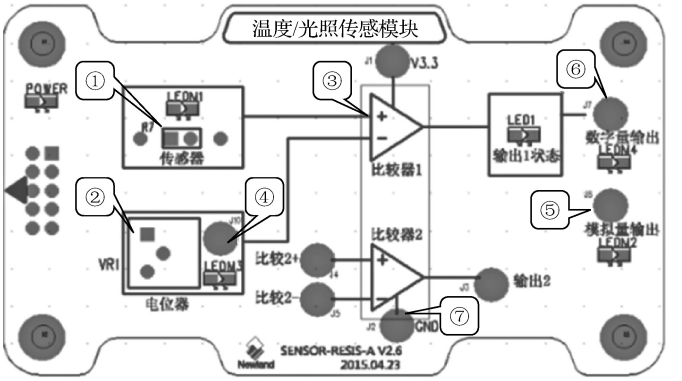


图 3-39　温度/光照传感模块组成

说明：

①：温度和光照传感器。

②：基准电压调节电位器。

③：比较器电路。

④：基准电压测试接口 J10，测试温度感应的阈值电压，即比较器 1 负端（3 脚）电压。

⑤：模拟量输出接口 J6，测试热敏电阻两端的电压，即比较器 1 正端（2 脚）电压。

⑥：数字量输出接口 J7，测试比较器 1 输出电压。

⑦：接地 GND 接口 J2。

3.5.3 湿度传感器工作原理

1．湿度传感器概述

湿度传感器，即能够感受外界湿度变化，并通过元器件材料的物理或化学性质变化，将湿度转化成有用信号的元器件。湿度检测较其他物理量的检测显得更加困难，首先，空气中水蒸气含量很少；其次，液态水会使一些高分子材料和电解质材料溶解，一部分水分子电离后与溶入水中的空气中的杂质结合成酸或碱，使湿敏材料受到不同程度的腐蚀和老化，从而丧失其原有的性质；第三，湿度信息的传递必须靠水对湿敏元器件的直接接触来完成，因此湿敏元器件只能直接暴露于待测环境中，不能密封。通常，对湿敏元器件有下列要求：在各种气体环境下稳定性好、响应时间短、寿命长、有互换性、耐污染和受温度影响小等。微型化、集成化及廉价是湿敏元器件的发展方向。

湿度是表示空气中水蒸气含量的物理量，常用绝对湿度、相对湿度、露点等表示。所谓绝对湿度就是单位体积空气内所含水蒸气的质量，也就是指空气中水蒸气的密度，一般用一立方米空气中所含水蒸气的克数表示，即 $h_a=m_v/v$，单位为 g/m^3。式中，m_v 为待测空气中水蒸气的质量，v 为待测空气的总体积。相对湿度是指空气中实际所含水蒸气的分压（p_w）和同温度下饱和水蒸气的分压（p_n）的百分比，即 $h_t=(p_w/p_n)t\times100\%RH$。通常，用%RH 表示相对湿度。当温度和压力变化时，因饱和水蒸气变化，空气中的水蒸气分压即使相同，其相对湿度也会发生变化。日常生活中所说的空气湿度，实际上就是指相对湿度。温度越高的空气，含水蒸气越多。若将空气冷却，即使其中所含水蒸气质量不变，相对湿度将逐渐增加，到某一个温度时，相对湿度达 100%，呈饱和状态，再冷却时，水蒸气的一部分凝聚生成露，把这个温度称为露点温度。即空气在气压不变时，为了使其所含水蒸气达到饱和状态所必须冷却到的温度称为露点温度。气温和露点温度的差越小，空气越接近饱和。

湿度的测量方式有几种，即采用伸缩式湿度计、干湿球湿度计、露点计和阻抗式湿度计等进行测量。伸缩式湿度计利用了毛发、纤维素等物质随湿度变化而伸缩的性质，以前多用于自动记录仪、空调的自动控制等，目前用于家庭设备的是把纤维素与厚度约为 50pm 的金属箔黏合在一起，卷成螺旋状的传感器，不需要进行温度补偿，但不能转换为电信号。阻抗式湿度计是根据湿敏电阻的阻抗值变化而求得湿度的一种湿度计，由于能简单地转换为电信号而被广泛采用。

2．湿度传感器的分类

湿敏元件是最简单的湿度传感器。湿敏元件主要分为湿敏电阻、湿敏电容两大类。

湿敏电阻的特点是在其基片上覆盖了一层用感湿材料制成的膜，当空气中的水蒸气吸附在感湿膜上时，其电阻率和电阻值都发生变化，利用这一特性即可测量湿度。湿敏电阻的种

类很多，如金属氧化物湿敏电阻、硅湿敏电阻、陶瓷湿敏电阻等。湿敏电阻的优点是灵敏度高，缺点是线性度和产品的互换性差。

湿敏电容一般是用高分子薄膜电容制成的，常用的高分子材料有聚苯乙烯、聚酰亚胺、酪酸醋酸纤维等。当环境湿度发生改变时，湿敏电容的介电常数发生变化，使其电容量也发生变化，其电容量变化量与相对湿度成正比。湿敏电容的主要优点是灵敏度高、产品互换性好、响应速度快、湿度的滞后量小、便于制造、容易实现小型化和集成化，其精度一般比湿敏电阻要低一些。

电子式湿敏传感器的准确度可达（2～3）%RH，比干湿球湿度计的测量精度高。湿敏元件的线性度及抗污染性差，在检测环境湿度时，湿敏元件要长期暴露在待测环境中，很容易被污染而影响其测量精度及长期稳定性。这方面没有用干湿球湿度计的测湿方法好。下面对各种湿度传感器进行简单的介绍。

（1）氯化锂湿度传感器

① 电阻式氯化锂湿度传感器。

第一个基于电阻-湿度特性原理的电阻式氯化锂湿度传感器是美国标准局的 F.W.Dunmore 研制出来的。这种传感器具有较高的精度，同时结构简单、成本低，适用于常温常湿的测控。

电阻式氯化锂湿度传感器的测量范围与湿敏层的氯化锂浓度及其他成分有关。单个元件的有效感湿范围一般在 20%RH 以内。例如 0.05%的浓度对应的感湿范围为（80～100）%RH，0.2%的浓度对应的感湿范围是（60～80）%RH。由此可见，要测量较宽的湿度范围时，必须把不同浓度的元件组合在一起使用。可用于全量程测量的湿度传感器组合的元件一般为 5 个，采用元件组合法的氯化锂湿度传感器测量范围通常为（15～100）%RH，国外有些产品称其测量范围可达（2～100）%RH。

② 露点式氯化锂湿度传感器。

露点式氯化锂湿度传感器是由美国的 Forboro 公司首先研制出来的，其后许多国家做了大量的研究工作。这种湿度传感器和上述电阻式氯化锂湿度传感器形式相似，但工作原理完全不同。简而言之，它是利用氯化锂饱和水溶液的饱和水汽压随温度变化而工作的。

（2）碳湿敏元件

碳湿敏元件是美国的 E.K.Carver 和 C.W.Breasefield 于 1942 年首先提出来的，与常用的毛发、肠衣和氯化锂等探空元件相比，碳湿敏元件具有响应速度快、重复性好、无冲蚀效应和滞后环窄等优点。我国气象部门于 20 世纪 70 年代初开展碳湿敏元件的研制，并取得了积极的成果，其测量不确定度不超过±5%RH，时间常数在正温时为 2～3s，滞差一般在 7%左右，比阻稳定性亦较好。

（3）氧化铝湿度传感器

氧化铝湿度传感器的突出优点是体积可以非常小（如用于探空仪的湿敏元件仅 90μm 厚、质量为 12mg），灵敏度高（测量下限达-110℃露点温度），响应速度快（响应时间一般为 0.3～3s），测量信号直接以电参量的形式输出，大大简化了数据处理程序。另外，它还适用于测量液体中的水分。如上特点正是工业和气象中的某些测量领域所需要的，因此它被认为是进行高空大气探测可供选择的几种合乎要求的传感器之一。近年来，这种传感器在工业中的低霜点测量领域开始崭露头角。

（4）陶瓷湿度传感器

在湿度测量领域，对于在低湿和高湿、低温和高温条件下的测量，目前为止仍然是一个薄弱环节，而其中又以高温条件下的湿度测量技术最为落后。一方面，通风干湿球湿度计几乎是在这个温度条件下唯一可以使用的产品，而该产品在实际使用中亦存在种种问题，无法

令人满意。另一方面，随着科学技术的进步，要求在高温下测量湿度的场合越来越多，如水泥生产、金属冶炼和食品加工等许多涉及工艺条件和质量控制的工业过程都需要进行湿度测量与控制。因此，自 20 世纪 60 年代起，许多国家开始研制适用于在高温条件下进行测量的湿度传感器。考虑到传感器的使用条件，人们很自然地把探索方向着眼于既具有吸水性又能耐高温的某些无机物上。实践已经证明，陶瓷元件不仅具有湿敏特性，而且可以作为感温元件和气敏元件，这些特性使它极有可能成为一种有发展前途的多功能传感器。

以上是应用较多的几种传感器，另外还有其他根据不同原理研制的湿度传感器，这里就不一一介绍了。

3．本实验所用温湿度传感模块组成

本实验所用的温湿度传感模块，主控器件采用瑞士 Sensirion 公司的 SHT10 单片数字温湿度集成 IC。该集成 IC 中包括一个电容式聚合体测湿组件和一个能隙式测温组件，并与一个 14 位的 A/D 转换器及串行接口电路在同一芯片上实现无缝连接，如图 3-40 所示。

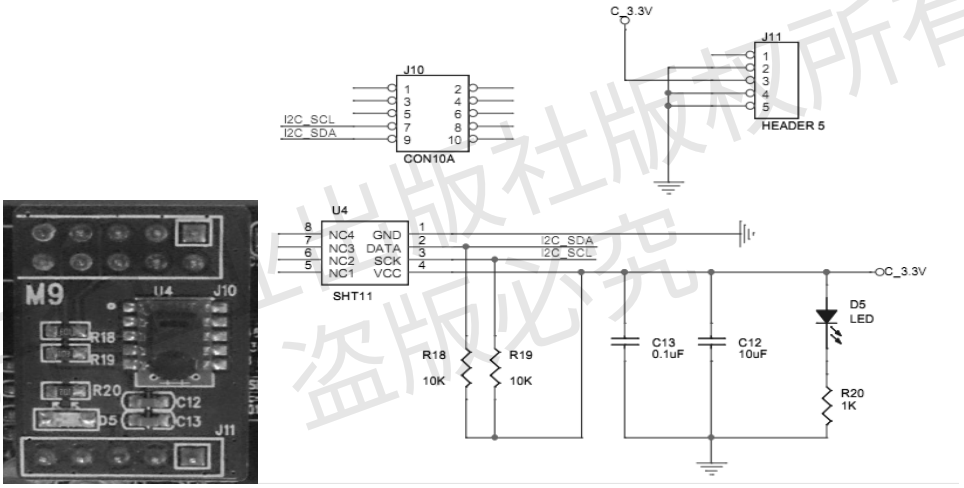


图 3-40　本实验所用的温湿度传感模块外形及电路连线图

3.5.4　任务实训步骤

第 1 步，新建工程、配置工程相关设置。

具体操作参照任务 3.2。

第 2 步，编写程序。

编写主程序，或将技术服务网站上相应的源代码添加进工程。下面将 sensor.c 和 collect.c 两个文件中的关键代码分别分析如下。

（1）sensor.c 中的 main()函数

```
/*********************************main*************************************************/
1.  void main(void)
2.  { uint16 sensor_val,sensor_tem;
3.  uint16 len = 0;
4.  halBoardInit();                          //模块相关资源的初始化
5.  ConfigRf_Init();                         //无线收、发模块参数的配置初始化
6.  halLedSet(1);
7.  halLedSet(2);
```

```
    Timer4_Init();                          //定时器初始化
    Timer4_On();                            //打开定时器
    while(1)
    { APP_SEND_DATA_FLAG = GetSendDataFlag();
    if(APP_SEND_DATA_FLAG == 1)             //定时时间到
    {    /*【传感器采集、处理】 开始*/
    #if defined (TEM_SENSOR)                //温湿度传感器
    call_sht11(&sensor_tem,&sensor_val);    //取温、湿度数据
    //把采集的数据转化成字符串，以便于在串口上显示、观察
    printf_str(pTxData,"温湿度传感器，温度：%d.%d, 湿度：%d.%d\r\n", sensor_tem/10,sensor_tem%10,
    sensor_val/10,sensor_val%10);
    #endif
    halLedToggle(3);                        //绿灯取反，无线发送指示
    //把采集的数据通过 ZigBee 模块发送出去
    basicRfSendPacket(SEND_ADDR, pTxData,strlen(pTxData ));
    Timer4_On();                            //打开定时器
    }  /*【传感器采集、处理】 结束*/
    }
    }
    /************************************main end ************************************/
```

（2）collect.c 中的 main()函数

```
    /************************************main start************************************/
    void main(void)
    { uint16 len = 0;
    halBoardInit();                         //模块相关资源的初始化
    ConfigRf_Init();                        //无线收、发模块参数的配置初始化
    halLedSet(1);
    halLedSet(2);
    while(1)
    { if(basicRfPacketIsReady())            //查询有无收到无线数据
    { halLedToggle(4);                      //红灯取反，无线接收指示
    //接收无线数据
    len = basicRfReceive(pRxData, MAX_RECV_BUF_LEN, NULL);
    //把接收到的无线数据发送到串口
    halUartWrite(pRxData,len);
    }
    }
    }
    /************************************main end ************************************/
```

第 3 步，建立模块设备。

参考任务 3.4 操作建立 tem_sensor 与 collect 模块。

第 4 步，模块连接及下载程序。

（1）温湿度传感模块

将温湿度传感模块固定在 NEWLab 平台上，选择“tem_sensor”模块，选择“collect.c”，单击鼠标右键，选择“Options”，在弹出的对话框中将“Exclude from build”复选框打“√”，然后单击“OK”按钮。重新编译程序无误后，给 NEWLab 平台上电，下载程序到温湿度传感模块中。

（2）协调器模块

选择“collect”模块，选择“sensor.c”，单击鼠标右键，选择“Options”，在弹出的对话框中将“Exclude from build”复选框打“√”，然后单击“OK”按钮。重新编译程序无误后，将协调器模块通过串口线连接到 PC 串口或者通过 USB 转串口线连接到 PC，给协调器通电，下载程序到协调器模块中。

温湿度传感模块与协调器模块连接图如图 3-41 所示。

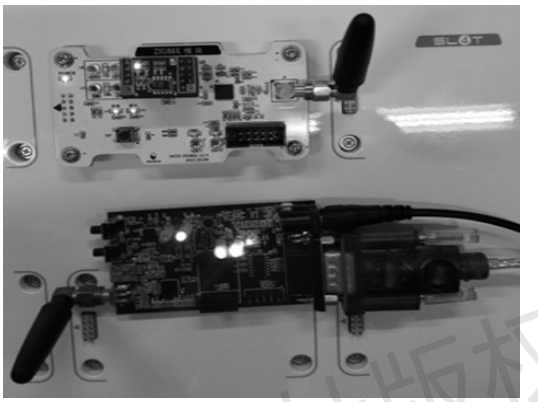


图 3-41　温湿度传感模块与协调器模块连接图

第 5 步，运行程序。

（1）将温湿度传感模块上电。

（2）打开串口调试软件，把串口的波特率设置为 38400bps。根据温、湿度的变化，在 PC 的串口调试终端上显示不同的温、湿度数据。

任务 3.6　环境智能监测系统设计与应用

【任务描述】

模拟现代智能农业养殖棚，要求能够实时监测棚内光的强度，气体浓度及棚内温、湿度，并能够监测是否有外来物（人或动物）侵入，最后将这些监测信息通过无线通信的方式传输至远程计算机。

【任务环境】

硬件：NEWLab 平台 1 套、ZigBee 节点板 5 块、温度/光照传感模块 1 个、温湿度传感模块 1 个、红外传感模块 1 个、气体传感模块 1 个、CC2530 仿真器 1 组、PC 1 台，信号线若干。

软件：Windows 7/10，IAR 集成开发环境，串口调试助手。

【必备知识点】

1．通信网络地址概述；

2．Basic RF 驱动文件介绍。

3.6.1　通信网络地址概述

Basic RF 点对点通信和其他网络通信一样，要求发送节点和接收节点必须在同一网络范围内，而约束网络范围的几个参数包括网络地址 PANID（简称网络 ID）、信道 CHANNEL、发送地址 SendAddr 和接收（本地）地址 MyAddr。

以任务 3.2 无线串口通信为例，在该任务中发送节点与接收节点两者的通信地址设置如下。

```
/*****************************点对点通信地址设置*****************************/
1.  #define RF_CHANNEL          20          //通信信道号 11～26
2.  #define PAN_ID              0x1379      //网络 ID
3.  #define MY_ADDR             0x1234      //本机模块地址，1 号模块
4.  #define SEND_ADDR           0x5678      //发送地址，1 号模块
5.  //#define MY_ADDR           0x5678      //本机模块地址，2 号模块
6.  //#define SEND_ADDR         0x1234      //发送地址，2 号模块
/*****************************************************************************/
```

1．信道

ZigBee 模块采用的是免执照的工业科学医疗（ISM）频段，所以使用了 3 个频段，分别为：868MHz（欧洲）、915MHz（美国）、2.4GHz（全球）。

因此，ZigBee 模块共定义了 27 个物理信道。其中，868MHz 附近频段定义了一个信道；915MHz 附近频段定义了 10 个信道，信道间隔为 2MHz；2.4GHz 附近频段定义了 16 个信道，信道间隔为 5MHz。具体信道分配如表 3-7 所示。

表 3-7　ZigBee 信道分配

信　道　号	中心频率/MHz	信道间隔/MHz	频率上限/MHz	频率下限/MHz
k=0	868.3		868.6	868
k=1,2,3,…,10	906+2×(k−1)	2	928	902
k=11,12,13,…,26	2401+5×(k−11)	5	2483.5	2400

理论上，在 868MHz 频段的物理层，数据传输速率为 20kb/s；在 915MHz 频段的物理层，数据传输速率为 40kb/s；在 2.4GHz 频段的物理层，数据传输速率为 250kb/s。实际上，除去信道竞争应答和重传等消耗，真正能被应用所利用的传输速率可能不足 100kb/s，并且余下的传输速率可能要被临近多个节点和同一个节点的应用瓜分。

注意：ZigBee 模块工作在 2.4GHz 频段时，与其他通信协议的信道有冲突。15、20、25、26 号信道与 WiFi 信道冲突较小，蓝牙基本不会与之冲突，无线电话尽量不与 ZigBee 模块同时使用。

2．PANID

PANID，全称是 Personal Area Network ID，一个网络只有一个 PANID，主要用于区分不同的网络，从而允许同一地区同时存在多个不同 PANID 的 ZigBee 网络。

Z-Stack 允许用两种方式配置 PANID，当 ZDAPP_CONFIG_PAN_ID 值不设置为 0xFFFF 时，那么设备建立或加入一个“最优”的网络，协调器可以随机获取一个 16 位的 PANID，建立一个网络，路由器或者终端节点可以加入任意一个自己设定信道上的网络，则不去关心 PANID。

注意：在不同地区或者同一地区不同的信道可以使用同一 PANID。

3．发送地址、接收地址

发送地址是指对方地址，接收地址是指本机地址。显然收、发双方地址对称才可以通信。

综上所述，在同一个系统内的各节点若要相互通信，要保证在同一个信道、PANID 下才可以连接成功。

3.6.2 Basic RF 驱动文件介绍

在基于 Basic RF 通信协议开发项目时，往往需要去技术服务网站上下载相关模块的驱动文件夹 CC2530_lib，这个文件夹包括相应模块的驱动和初始化等各类文件，如下所示。

```
|---common
|    |---hal_cc8051.h——MCU 输入/输出宏定义
|    |---hal_defs.h——通用定义
|    |---hal_mcu.c——MCU 函数库
|    |---hal_mcu.h——MCU 函数库的定义
|    |---hal_clock.c——时钟函数库
|    |---hal_clock.h——时钟函数库的定义
|    |---hal_digio.c——输入/输出中断函数库
|    |---hal_digio.h——输入/输出中断函数库的定义
|    |---hal_adc.c——ADC 函数库
|    |---hal_adc.h——ADC 函数库的定义
|    |---hal_int.c——中断函数库
|    |---hal_int.h——中断函数库的定义
|    |---hal_rf.c——无线函数库
|    |---hal_rf.h——无线函数库的定义
|    |---hal_rf_security.c——无线加密函数库
|    |---hal_rf_security.h——无线加密函数库的定义
|    |---hal_rf_util.c——无线通用函数库
|    |---hal_rf_util.h——无线通用函数库的定义
|    |---hal_timer_32k.c——32K 定时器函数库
|    \---hal_timer_32k.h——32K 定时器函数库的定义
|
|---basicrf
|    |---basic_rf.c——基本无线函数库
|    |---basic_rf.h——基本无线函数库的定义
|    |---basic_rf_security.c——基本无线加密函数库
|    \---basic_rf_security.h——基本无线加密函数库的定义
|
\---utils
|    |---util.c——工具函数库
|    \---util.h——工具函数库的定义
|
\---board
|    |---hal_board.c——ZigBee 模块上的资源初始化函数库
|    |---hal_board.h——ZigBee 模块上的资源初始化函数库的定义
|    |---hal_led.c——ZigBee 模块上关于 LED 的函数库
```

```
|    |---hal_led.h——ZigBee 模块上关于 LED 的函数库的定义
|
\--- module
|    |--- dma_ad590.c——模拟温度传感器函数库
|    |--- dma_ad590.h——模拟温度传感器函数库的定义
|    |--- dma_bma.c——重力传感器函数库
|    |--- dma_bma.h——重力传感器函数库的定义
|    |--- dma_dc.c——直流电机函数库
|    |--- dma_dc.h——直流电机函数库的定义
|    |--- dma_eeprom.c——EEPROM 函数库
|    |--- dma_eeprom.h——EEPROM 函数库的定义
|    |--- dma_imc.c——人体传感器函数库
|    |--- dma_imc.h——人体传感器函数库的定义
|    |--- dma_m4.c——光敏/光电传感器函数库
|    |--- dma_m4.h——光敏/光电传感器函数库的定义
|    |--- dma_tc72.c——数字温度传感器函数库
|    |--- dma_tc72.h——数字温度传感器函数库的定义
|    |--- dma_tgs.c——酒精传感器函数库
|    |--- dma_tgs.h——酒精传感器函数库的定义
|    |--- dma_sht.c——温湿度传感器函数库
|    |--- dma_sht.h——温湿度传感器函数库的定义
|    |--- dma_itg.c——陀螺仪传感器函数库
|    |--- dma_itg.h——陀螺仪传感器函数库的定义
|    |--- dma_kr.c——可燃气体传感器函数库
|    |--- dma_kr.h——可燃气体传感器函数库的定义
|    |--- dma_tgs2602.c——气体质量传感器函数库
|    |--- dma_tgs2602.h——气体质量传感器函数库的定义
```

3.6.3　任务实训步骤

第 1 步，搭建硬件环境，连接各模块。

在 NEWLab 平台上，连接各模块，如图 3-42 所示。

（1）红外传感模块的组成

把 ZigBee 模块和红外传感模块固定到 NEWLab 平台上，红外传感模块的对射输出 2 接口（J6）与 ZigBee 模块的 IN0 接口（J13/P1.3）相连。

（2）温度/光照传感模块的组成

把 ZigBee 模块和温度/光照传感模块固定到 NEWLab 平台上，温度/光照传感模块的模拟量输出接口（J6）与 ZigBee 模块的 ADC0 接口（J10/P1.0）相连。

（3）气体传感模块的组成

把 ZigBee 模块和气体传感模块固定到 NEWLab 平台上，气体传感模块的模拟量输出接口（J6）与 ZigBee 模块的 ADC0 接口（J10/P1.0）相连。

（4）温湿度传感模块的组成。

把温湿度传感模块插入 ZigBee 模块的 U5 接口。

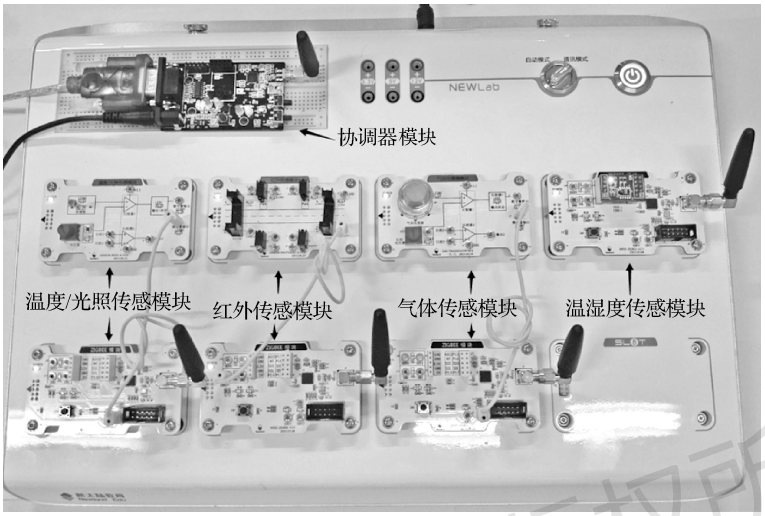


图 3-42　环境智能监测系统各模块

第 2 步，新建各传感模块工程、配置工程。

（1）选择“Project”→“Edit Configuration”命令，新建 gm_sensor、qt_sensor、hw_sensor、tem_sensor 和 collect 5 个工程。

（2）各传感模块工程的配置，参照任务 3.2 操作。

第 3 步，编写各传感模块程序。

各传感模块主程序 sensor.c 中的 main()函数如下。

```
/*********************************************main**********************************/
void main(void)
{ uint16 sensor_val,sensor_tem;
uint16   len = 0;
halBoardInit();                                  //模块相关资源的初始化
ConfigRf_Init();                                 //无线收、发模块参数的配置初始化
halLedSet(1);
halLedSet(2);
Timer4_Init();                                   //定时器初始化
Timer4_On();                                     //打开定时器
while(1)
{    APP_SEND_DATA_FLAG = GetSendDataFlag();
if(APP_SEND_DATA_FLAG == 1)                      //定时时间到
{    /*【传感器采集、处理】 开始*/
#if defined (GM_SENSOR)                          //光照传感器
sensor_val=get_adc();                            //取模拟电压
//把采集的数据转化成字符串，以便于在串口上显示、观察
printf_str(pTxData,"光照传感器电压：%d.%02dV\r\n",sensor_val/100,sensor_val%100);
#endif
#if defined (QT_SENSOR)                          //气体传感器
sensor_val=get_adc();                            //取模拟电压
//把采集的数据转化成字符串，以便于在串口上显示、观察
```

```
22.  printf_str(pTxData,"气体传感器电压：%d.%02dV\r\n",sensor_val/100,sensor_val%100);
23.  #endif
24.  #if defined (HW_SENSOR)                          //红外传感器
25.  sensor_val=get_hwsensor();                       //取红外传感器检测结果
26.  //把采集的数据转化成字符串，以便于在串口上显示观察
27.  if(sensor_val)
28.  { printf_str(pTxData,"红外传感器电压：%d\r\n",sensor_val); }
29.  else
30.  { printf_str(pTxData,"红外传感器电压：%d\r\n",sensor_val); }
31.  #endif
32.  #if defined (TEM_SENSOR)                         //温湿度传感器
33.  call_sht11(&sensor_tem,&sensor_val);             //取温、湿度数据
34.  //把采集的数据转化成字符串，以便于在串口上显示、观察
35.  printf_str(pTxData,"温湿度传感器，温度：%d.%d, 湿度：%d.%d\r\n",
36.  sensor_tem/10,sensor_tem%10,sensor_val/10,sensor_val%10);
37.  #endif
38.  halLedToggle(3);                                 //绿灯取反，无线数据发送指示
39.  //把采集的数据通过 ZigBee 模块发送出去
40.  basicRfSendPacket(SEND_ADDR, pTxData,strlen(pTxData ));
41.  Timer4_On();                                     //打开定时器
42.  }  /*【传感器采集、处理】 结束*/
43.  }
44.  }
     /*************************************main end ************************************/
```

第 4 步，编写协调器程序。

协调器节点主程序 collect.c 中的 main()函数如下。

```
     /*************************************main start************************************/
1.   void main(void)
2.   { uint16 len = 0;
3.   halBoardInit();                                  //模块相关资源的初始化
4.   ConfigRf_Init();                                 //无线收、发模块参数的配置初始化
5.   halLedSet(1);
6.   halLedSet(2);
7.   while(1)
8.   { if(basicRfPacketIsReady())                     //查询有无收到无线数据
9.   { halLedToggle(4);                               //红灯取反，无线数据接收指示
10.  //接收无线数据
11.  len = basicRfReceive(pRxData, MAX_RECV_BUF_LEN, NULL);
12.  //把接收到的无线数据发送到串口
13.  halUartWrite(pRxData,len);
14.  }
15.  }
16.  }
     /*************************************main end ************************************/
```

第 5 步，编译、烧录程序，测试系统功能。

（1）为传感器节点编译、烧录程序。

① 在“Workspace”栏中选择“gm_sensor”，然后在预定义栏中输入“GM_SENSOR”，

再编译程序，无误后烧录到该模块中。

② 在“Workspace”栏中选择“qt_sensor”，然后在预定义栏中输入“QT_SENSOR”，再编译程序，无误后烧录到该模块中。

③ 在“Workspace”栏中选择“hw_sensor”，然后在预定义栏中输入“HW_SENSOR”，再编译程序，无误后烧录到该模块中。

④ 在“Workspace”栏中选择“tem_sensor”，然后在预定义栏中输入“TEM_SENSOR”，再编译程序，无误后烧录到该模块中。

（2）为协调器编译、烧录程序。

（3）测试系统功能，运行效果如图 3-43 所示。

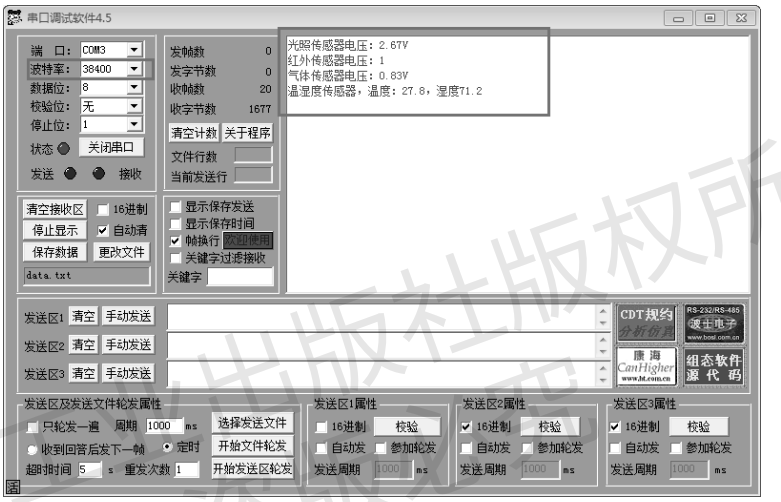


图 3-43　环境智能监测系统运行效果

【知识点小结】

1．Basic RF 操作依次包括启动、发送、接收三个环节，依靠各个函数完成相应的功能。

2．串口通信最重要的参数是波特率、数据位、停止位和奇偶校验位。两个进行通信的端口，这些参数必须匹配。

3．传感器是能感受被测量并按照一定的规律转换成可用输出信号的器件或装置，通常由敏感元件和转换元件组成。按输出信号，可将传感器分为开关量传感器、模拟量传感器和数字量传感器三类。

4．同一个系统内的各节点若要相互通信，要保证在同一个信道、网络地址下才可以连接成功。在 2.4GHz 频段下，信道号有效值范围为 11～26，网络地址有效取值范围为 0x0000～0xFFFF。

【拓展与思考】

1．在任务 3.1 中，改变设置，使两个程序中的 RF_CHANNEL 或 PAN_ID 不一致，观察结果；使一个程序中的 MY_ADDR 与另一个程序中的 SEND_ADDR 不相等，又会出现什么结果？

2．在实现任务 3.3 的基础上，增加霍尔传感模块、人体感应传感模块，运行后观察串口调试窗口显示的数据。

【强国实训拓展】

结合本项目所学技能，助力加快建设农业强国，扎实推动乡村产业、人才、文化、生态、组织振兴，实现农业现代化总体目标。试设计基于 ZigBee 无线组网通信的智能农业温室系统设计方案，实现温室温、湿度和光的强度监测，并实现智能换气、照明功能。