

### 3.13~3.14

2017年3月13日 17:05

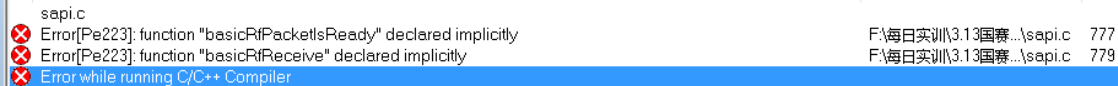
#### 1. 所遇问题

- 任务二编译问题。详见[2015国赛任务二编译错误解决方案](#)
- 用定时器时，写了定时器初始化函数，然后很机制的把定时器1模式选择写在触发的代码。但是由于没有在程序开头调用初始化函数，导致浪费了大概十分钟去找错误。
- 做题2时，用某块ZigBee模板按键去发送消息控制另外一块板子的灯，由于使用的IAR延时不准，于是消抖延时不够导致发送消息两次，我代码写着又是收到一次消息就将控制流水灯的标志位取反，这样就导致了只会流水灯只会运行一次。而且调试过程中，按键抖动是发现不了的，一直单步步进你会发现代码没有问题，这时候就要考虑消抖是否彻底了。
- 对按键使用中断时需要注意清除中断标志位，P1口需要清两次不然Bug也会来找你。
- 卸载SQL时，如果是64位的，注意在卸载程序那选择SQL 2016 64Bit卸载。
- 14号继续写题3，我在写sapi.c中写接收到数据处理方法出现如下图错误，这是没有找到函数声明导致。其中还有一个继电器的函数（set\_relay()）也找不到声明，于是我去找了继电器的函数所在位置（在sensor.c中）。只要找到了所在位置就好办了，检查头文件（sensor.h）中是否有声明“extern void set\_relay(uint8 status);”，有就直接在sapi.c中添加头文件引用（“#include "sensor.h" ”），无则加上那句声明。

然而下面还有两个函数没声明，但是我找了好久还没找到在哪里，我开始怀疑是不是这样写。于是我打算找个文件内搜索工具来找到函数位置，然后加上声明编译看看。于是我真的没有在题3文件里找到下面两个函数的声明，然后我在想消息处理那里是不是已经接收到了消息，直接判断就可以了。看看书摸索一下果然是，仿照书中代码如下即可读取到。

```
unsigned char buffer[3] = " ";
```

```
osal_memcpy(buffer,pMSGpkt->cmd.Data,3);
```



- 题3组网设置PanID和Channel时，必须全部修改完再调用参数去重启，不然会导致无法组网。
- 题3给.Net工具通过串口发送数据时，需注意必须按通讯协议发送该数组，且同时不发送其他数据。

#### 2. 方法

- 题3时，任务书并没有说清项目文件放在哪里，在题3目录下搜索任务书中提到的其中一个文件的名字“Coo”，即刻快速锁定项目文件路径（Projects \zstack\Samples \SensorDemo\CC2530DB）。
- 搜索“user code”，锁定写代码区域。
- 使用uart\_printf("\r\n哈哈哈哈哈\r\n"); 函数可输出中文至串口，便于调试。
- 在sapi.c中有以下代码，将uart\_printf注释去掉即可看到接收到的消息。

```
1 void SAPI_ReceiveDataIndication( uint16 source, uint16 command, uint16 len, uint8 *pData )
2 {
3     //uart_printf("\r\nreceive data:0x%x,0x%x,0x%x,0x%x",pData[0],pData[1],pData[2],pData[3]);
4     #if (LOG_TYPE==0) // 协调器
5         #if defined ( MT_SAPI_CB_FUNC )
6             /* First check if MT has subscribed for this callback. If so , pass it as
7              a event to MonitorTest and return control to calling function after that */
8             if ( SAPI_CB_CHECK( SPI_CB_SAPI_RCV_DATA_IND ) )
9             {
10                 zb_MTCallbackReceiveDataIndication( source, command, len, pData );
11             }
12         else
13             #endif //MT_SAPI_CB_FUNC
14             {
15             }
16         }
17     #endif //(LOG_TYPE==0)
18     zb_ReceiveDataIndication( source, command, len, pData);
19 }
```

#### 3. 代码参考

##### a. 题1

```
1 #include "ioCC2530.h"
2
3 #define D5 (P1_3)
4 #define SW1 (P1_2)
5
6 unsigned char flag = 0;
7
8 void delay(unsigned int time)
9 {
10     unsigned int i;
11     unsigned char j;
12     for(i=0;i<time;i++)
13         for(j=0;j<240;j++)
14         {
15             asm("NOP");
16             asm("NOP");
17             asm("NOP");
```

```

1#include "ioCC2530.h"
2
3#define D5 (P1_3)
4#define SW1 (P1_2)
5
6unsigned char flag = 0;
7
8void delay(unsigned int time)
9{
10    unsigned int i;
11    unsigned char j;
12    for(i=0;i<time;i++)
13        for(j=0;j<240;j++)
14        {
15            asm("NOP");
16            asm("NOP");
17            asm("NOP");
18        }
19}
20
21void initT1(void)
22{
23    T1CTL |= 0x0C;    //128
24    T1CC0L = 0x12;    //0.25s
25    T1CC0H = 0x7A;
26    T1IE = 1;        //使能定时器1中断
27    T1OVFIM = 1;     //使能定时器1溢出中断
28    EA = 1;          //使能总中断
29}
30
31#pragma vector = T1_VECTOR
32__interrupt void T1_INT(void)
33{
34    T1STAT &= ~0x20; //clear timer1 overflow sign
35    if(++flag == 4)
36    {
37        D5 = 0;
38        flag = 0;
39    }
40}
41
42void main(void)
43{
44    P1SEL &= ~0x0C;
45    P1DIR |= 0x08;
46    P1DIR &= ~0x04;
47
48    initT1();    //不要忘了初始化
49    D5 = 0;
50
51    while(1)
52    {
53        if(SW1 == 0)
54        {
55            delay(100);
56            if(SW1 == 0)
57            {
58                T1CTL |= 0x03; //采用正计数模式 开始工作
59                D5 = 1;
60                /*//延时还是有问题 写4000约2秒
61                D5 = 1;
62                delay(4000);
63                D5 = 0;*/
64            }
65        }
66    }
67}

```

b. 题2 (主节点和从节点编译时互换宏定义处的地址)

```

1#include "hal_defs.h"
2#include "hal_cc8051.h"
3#include "hal_int.h"
4#include "hal_mcu.h"
5#include "hal_board.h"
6#include "hal_led.h"

```

```

3 #include "hal_inc.h"
4 #include "hal_mcu.h"
5 #include "hal_board.h"
6 #include "hal_led.h"
7 #include "hal_rf.h"
8 #include "basic_rf.h"
9 #include "hal_uart.h"
10 #include "sensor_drv/sensor.h"
11 #include <stdio.h>
12 #include <string.h>
13 #include <stdarg.h>
14 /*****点对点通讯地址设置*****/
15 #define RF_CHANNEL 11 // 频道 11~26
16 #define PAN_ID 0x8006 // 网络id
17 #define MY_ADDR 0xAC3A // 本机模块地址
18 #define SEND_ADDR 0x1015 // 发送地址
19 /******/
20 static basicRfCfg_t basicRfConfig;
21 // 无线RF初始化
22 void ConfigRf_Init(void)
23 {
24     basicRfConfig.panId = PAN_ID;
25     basicRfConfig.channel = RF_CHANNEL;
26     basicRfConfig.myAddr = MY_ADDR;
27     basicRfConfig.ackRequest = TRUE;
28     while(basicRfInit(&basicRfConfig) == FAILED);
29     basicRfReceiveOn();
30 }
31
32 /*****MAIN*****/
33 #include "ioCC2530.h"
34 #define D5 (P1_3)
35 #define D6 (P1_4)
36 #define SW1 (P1_2)
37
38 uint8 recBuf[3] = " ";
39 uint8 sendBuf[3] = "New";
40
41 void delay(unsigned int time)
42 {
43     unsigned int i;
44     unsigned char j;
45     for(i=0;i<time;i++)
46         for(j=0;j<240;j++)
47         {
48             asm("NOP");
49             asm("NOP");
50             asm("NOP");
51         }
52 }
53
54 void initLED(void)
55 {
56     P1SEL &= ~0x1C;
57     P1DIR |= 0x18;
58     P1DIR &= ~0x04;
59
60     D5 = 0;
61     D6 = 0;
62 }
63
64 void lightLED(void)
65 {
66     D5 = 1;
67     delay(5000);
68     D6 = 1;
69     D5 = 0;
70     delay(5000);
71     D6 = 0;
72 }
73
74 void initP1Break(void)
75 {
76     IEN2 |= 0x10;
77     P1IEN |= 0x04;
78     PICTL |= 0x02;
79     EA = 1;

```

```

76 IEN2 |= 0x10;
77 P1IEN |= 0x04;
78 PICTL |= 0x02;
79 EA = 1;
80 }
81
82 #pragma vector = P1INT_VECTOR
83 __interrupt void P1_INT(void)
84 {
85     //不使用中断 消抖不彻底则会无法循环
86     if(P1IFG & 0x04)
87     {
88         basicRfSendPacket(SEND_ADDR, sendBuf, 3);
89         P1IFG &= ~0x04; //清两次标志位!!!
90     }
91     P1IF = 0; //清两次标志位!!!
92 }
93
94 void main(void)
95 {
96     halBoardInit(); //选手不得在此函数内添加代码
97     ConfigRf_Init(); //选手不得在此函数内添加代码
98
99     initLED();
100    initP1Break();
101
102    unsigned char isLight = 0;
103
104    while(1)
105    {
106        /* user code start */
107        if(isLight)
108            lightLED();
109
110        if(basicRfPacketIsReady())
111        {
112            if(basicRfReceive(recBuf, 3, NULL) == 3)
113            {
114                if(recBuf[0] == 'N' && recBuf[1] == 'e' && recBuf[2] == 'w')
115                {
116                    isLight = 1; //主节点不可被控制 即总是循环
117                    //isLight = ~isLight; //非主节点可被控制 即反转
118                }
119            }
120        }
121        /* user code end */
122    }
123 }

```

c. 题3

i. Coord1.c

```

652 /*****
653  * @fn          channel_panid_init
654  *
655  * @brief       对信道和PANID判断并设置
656  *
657  * @param       none
658  *
659  * @return      none
660  */
661 void ChannelPanidInit (void)
662 {
663     /* user code start */
664     uint8 channel = zb_Readchannel();
665     uint8 panid[2];
666     zb_Readpanid(panid);
667     if(channel != 11 || panid[0] != 0x06 || panid[1] != 0x80)
668     {
669         panid[0] = 0x06;
670         panid[1] = 0x80;
671         channel = 11;
672         zb_Writepanid(panid);
673         zb_Writechannel(channel);
674         zb_SystemReset();
675     }

```

```

672     zb_Writepanid(panid);
673     zb_Writechannel(channel);
674     zb_SystemReset();
675 }
676 /* user code end */
677 }
678 /*****

```

#### ii. Enddev1.c

```

601 static void sendDummyReport(void)
602 {
603     /* user code start */
604     uint8 status[3] = " ";
605     status[0] = 'E';
606     status[1] = '1';
607     status[2] = '0'; //默认为0
608     status[2] = get_rentihongwai() == 0x01?'1':'0'; //1 have 0 no
609     zb_SendZigbeeDatas(status,3); //datas[] lenth
610     /* user code end*/
611 }

```

```

637 /*****
638  * @fn          channel_panid_init
639  *
640  * @brief        对信道和PANID判断并设置
641  *
642  * @param        none
643  *
644  * @return        none
645  */
646 void ChannelPanidInit (void)
647 {
648     /* user code start */
649     uint8 channel = zb_Readchannel();
650     uint8 panid[2];
651     zb_Readpandid(panid);
652     if(channel != 11 || panid[0] != 0x06 || panid[1] != 0x80)
653     {
654         panid[0] = 0x06;
655         panid[1] = 0x80;
656         channel = 11;
657         zb_Writepanid(panid);
658         zb_Writechannel(channel);
659         zb_SystemReset();
660     }
661     /* user code end */
662 }

```

#### iii. Enddev2.c

```

637 /*****
638  * @fn          channel_panid_init
639  *
640  * @brief        对信道和PANID判断并设置
641  *
642  * @param        none
643  *
644  * @return        none
645  */
646 void ChannelPanidInit (void)
647 {
648     /* user code start */
649     uint8 channel = zb_Readchannel();
650     uint8 panid[2];
651     zb_Readpandid(panid);
652     if(channel != 11 || panid[0] != 0x06 || panid[1] != 0x80)
653     {
654         panid[0] = 0x06;
655         panid[1] = 0x80;
656         channel = 11;
657         zb_Writepanid(panid);
658         zb_Writechannel(channel);
659         zb_SystemReset();
660     }
661     /* user code end */
662 }

```

#### iv. sapi.c

任务书中还要求间隔5s内发送数据给串口，下面的代码没写，需完善。我没打算用定时器去写，索性利用它每两秒触发的事件去写一个标志位，每触发两次再延时1s给串口发数据。不过这样有点小问题：就是第一次确实是第5秒时发送了数据，但后来因为人体红外传感器并不延时所以后面都是每4秒发一次数据。

#### iv. sapi.c

任务书中还要求间隔5s内发送数据给串口，下面的代码没写，需完善。我没打算用定时器去写，索性利用它每两秒触发的事件去写一个标志位，每触发两次再延时1s给串口发数据。不过这样有点小问题，就是第一次确实是第5秒时发送了数据，但后来因为人体红外传感器并不延时所以后面都是每4秒发一次数据。

```

59 #include "osal_nv.h"
60 #include "NLMEDE.h"
61 #include "AF.h"
62 #include "OnBoard.h"
63 #include "nwk_util.h"
64 #include "ZDPProfile.h"
65 #include "ZDObject.h"
66 #include "hal_led.h"
67 #include "hal_key.h"
68 #include "sapi.h"
69 #include "MT_SAPI.h"
70 #include "DemoApp.h"
71 #include "UART_PRINT.h"
72
73 #include "sensor.h" //引用继电器声明
74
75 extern uint8 zgStartDelay;
76 extern uint8 zgSapiEndpoint;
77 extern void ChannelPanidInit (void);
78 /*****
79  * CONSTANTS
80  */

```

```

772 case AF_INCOMING_MSG_CMD:
773     pMSGpkt = (afIncomingMSGPacket_t *) pMsg;
774     SAPI_ReceiveDataIndication( pMSGpkt->srcAddr.addr.shortAddr, pMSGpkt->clusterId,
775                               pMSGpkt->cmd.DataLength, pMSGpkt->cmd.Data);
776     /* user code start */
777     unsigned char buffer[3] = " ";
778     osal_memcpy(buffer, pMSGpkt->cmd.Data, 3);
779     uint8 datas[12] = {0xFE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF};
780     if(buffer[0] == 'E' && buffer[1] == '1' && buffer[2] == '1'){
781         //uart_printf("\r\n有人 打开风扇\r\n");
782         set_relay(1); //开继电器
783         datas[1] = 0x01;
784         datas[2] = 0x00;
785         datas[3] = 0x01;
786         datas[4] = 0x00;
787         HalUARTWrite(HAL_UART_PORT_0, datas, 12); //按协议发送数据至串口
788     }
789     else if(buffer[0] == 'E' && buffer[1] == '1' && buffer[2] == '0')
790     {
791         //uart_printf("\r\n没人 关闭风扇\r\n");
792         set_relay(0); //关继电器
793         datas[1] = 0x00;
794         datas[2] = 0x00;
795         datas[3] = 0x00;
796         datas[4] = 0x00;
797         HalUARTWrite(HAL_UART_PORT_0, datas, 12);
798     }
799     /* user code end */
800     break;

```

## 4. 参考资料

- [http://www.swanlinux.net/2013/05/14/c\\_extern/](http://www.swanlinux.net/2013/05/14/c_extern/)
- 串口中文函数



UART\_PRI  
NT



UART\_PRI  
NT