

SYD8811 i2c 中断功能的使用

I2C 写中断的使用:

这里使用开发板上面的 OLED 来测试 I2C 写中断的功能,这里是一幅图的数据,通过 I2C 接口发送给屏幕就可以在屏幕上看到图片:

```
42 uint8_t raw_data[] =
43 {
44 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
45 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
46 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
47 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
48 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
49 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
50 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
51 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
52 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
53 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
54 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
55 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
56 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
57 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
58 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
59 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
60 0x00,0x80,0xC0,0xC0,0xE0,0xE0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xE0,
61 0xE0,0xC0,0x00,0x00,0x30,0xF0,0xF0,0xF0,0xF0,0xF0,0xC0,0x00,0x00,0x00,0x00,0x00,
62 0x00,0x80,0xE0,0xF0,0xF0,0xF0,0xF0,0xF0,0x70,0x10,0x00,0xE0,0xF0,0xF0,0xF0,0xF0,
63 0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xE0,0xE0,0xC0,0xC0,0x80,0x00,0x00,
64 0x00,0x00,0x00,0xE0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,
65 0xF0,0xF0,0xF0,0xF0,0xF0,0x00,0x00,0x00,0xC0,0xE0,0xE0,0xF0,0xF0,0xF0,0xF0,
66 0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0x00,0x00,0x00,0x00,0x00,0xF0,
67 0xF0,0xF0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xE0,0xF0,0xF0,0xF0,0xE0,
68 0x1E,0x7F,0xFF,0xFF,0xFF,0xFF,0xF1,0xF1,0xE0,0xE0,0xE0,0xE0,0xE1,0xC1,0xC1,0x83,
69 0x87,0x07,0x00,0x00,0x00,0x00,0x03,0x07,0x1F,0x7F,0xFF,0xFF,0xFC,0xF8,0xE0,0xF0,
70 0xFE,0xFF,0xFF,0x3F,0x0F,0x03,0x01,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
71 0x01,0x01,0x01,0x01,0x01,0x01,0x03,0x03,0x0F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xC0,
72 0x00,0x00,0x00,0x01,0x01,0x01,0x01,0x01,0x01,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x01,
73 0x01,0x01,0x01,0x01,0x01,0x00,0x00,0xF0,0xFF,0xFF,0xFF,0xFF,0xFF,0x03,0x83,0xE1,
74 0xE1,0xE1,0xE1,0xE1,0xE1,0xE1,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,
75 0xFF,0xFF,0xFF,0x80,0xC0,0xE0,0xF8,0xFC,0xFE,0x7F,0x3F,0x1F,0x07,0x03,0x01,0x00,
76 0xF0,0xE0,0xE0,0xC1,0xC1,0x83,0x83,0x83,0x87,0x87,0x87,0x87,0xCF,0xFF,0xFF,0xFF,
77 0xFF,0x7F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0xFF,0xFF,0xFF,0xFF,0xFF,
78 0xFF,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,
79 0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xF0,0xFF,0xFF,0xFF,0xFF,0x3F,0x01,
80 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,
81 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0x80,0xC3,0xC3,
82 0xC3,0xC3,0xC3,0xC3,0xC3,0xC3,0x80,0x00,0x00,0x00,0x00,0xFF,0xFF,
83 0xFF,0xFF,0xFF,0x01,0x03,0x07,0x1F,0x3F,0x7F,0xFE,0xFC,0xF8,0xF0,0xC0,0x80,0x00,
84 0x03,0x03,0x07,0x07,0x07,0x0F,0x0F,0x0F,0x0F,0x07,0x07,0x07,0x03,0x03,
85 0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x07,0x07,0x07,
86 0x07,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x07,0x07,0x07,0x07,0x07,
87 0x07,0x07,0x07,0x07,0x07,0x07,0x07,0x07,0x07,0x03,0x01,0x00,0x00,0x00,
88 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x0F,0x0F,0x0F,0x07,0x00,
89 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,
90 0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,0x00,0x00,0x00,0x00,0x00,0x0F,0x0F,
91 0x0F,0x0F,0x0F,0x00,0x00,0x00,0x00,0x00,0x01,0x03,0x07,0x0F,0x0F,0x0F,0x0F,
92 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
93 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
94 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
95 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
96 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
97 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
98 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
99 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
100 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
101 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
102 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
103 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
104 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
105 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
106 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
107 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
108 };
109
110
```

屏幕初始化完成并且打开了总中断后调用 oled_fill_image_irq 函数进行图片绘制:

```

2 uint8_t raw_data[];
3 int main()
4 {
5     __disable_irq();
6     dbg_init();
7     dbg_printf("SYD8811 I2C oled demo %s:%s\r\n", __DATE__, __TIME__);
8
9     GPIO_Set_Output( U32BIT(LED1));
10    GPIO_Pin_Set(U32BIT(LED1));
11
12    oled_init();
13
14    oled_printf(0,0,"SYD Inc.");
15    oled_printf(0,2,"SYD8811 EVB");
16    oled_printf(0,4,"oled demo");
17    oled_printf(0,6,"%s", __DATE__);
18    delay_ms(2000);
19    //oled_fill_image(raw_data);
20    //oled_fill_image_buff(raw_data);
21    enable_irq();
22    oled_fill_image_irq(raw_data);
23
24    while(1)
25    {
26        GPIO_Pin_Turn(U32BIT(LED1));
27        delay_ms(1000);
28    }
29 }

```

该函数源码如下:

```

1 }
2
3 /*****LCD全屏显示图片*****/
4 void oled_fill_image_irq(uint8_t bmp_dat[])
5 {
6     start_y=0;
7     num_32byte=0;
8     end_y=8;
9     p_image=bmp_dat;
10    num_32byte_all=X_WIDTH/32;
11    oled_wrcmd(0xb0+start_y);
12    oled_wrcmd(0x01);
13    oled_wrcmd(0x10);
14    dbg_printf("num_32byte:%x end_y:%x num_32byte_all:%x\r\n",num_32byte,end_y,num_32byte_all);
15    // enable IRQ
16    NVIC_EnableIRQ(I2C1_IRQn);
17    i2c_1_write_irq(I2C_ID_OLED, 0x01, 0x40, &bmp_dat[start_y*X_WIDTH+num_32byte*32], 32);
18 }

```

注意 I2C 模块只有 32 个 byte,所以这里要分批次发送.i2c_1_write_irq 函数源码如下:

```

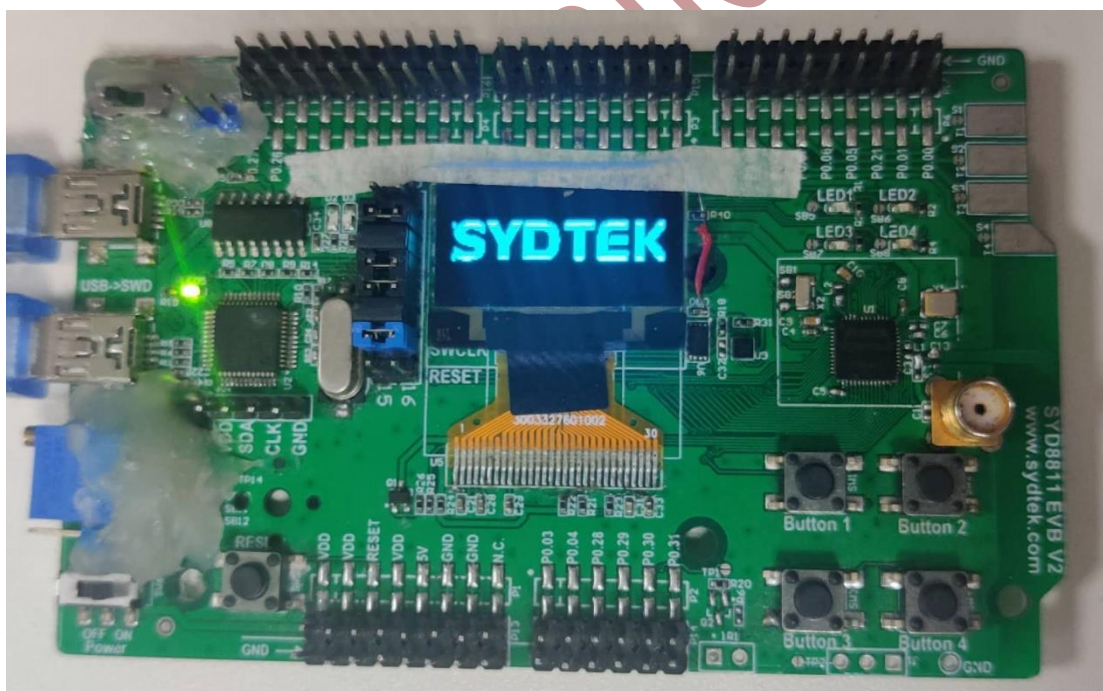
1 /*****I2C模块数据发送*****/
2 ErrorStatus i2c_1_write_irq(uint8_t id, uint8_t addr_len, uint16_t addr, uint8_t * buf, uint16_t sz)
3 {
4     ErrorStatus bret = SUCCESS;
5
6     I2C_1_CTRL->RSTB = 1;
7     I2C_1_CTRL->ID = id;
8     I2C_1_CTRL->ADDR_1 = addr_len;
9     I2C_1_CTRL->ADDRESS = addr;
10    I2C_1_CTRL->DATA_CNT = (sz - 1);
11
12    memcpy((uint8_t *)I2C_1_TXD->TX, buf ,sz);
13
14    I2C_1_CTRL->WRITE=1;
15
16    return bret;
17 }

```

当硬件把 I2C 的数据发送 sz 指定的数据的时候会产生中断,进入 I2C1_IRQHandler 函数,该函数具体会依次把未发送完的数据发送出去,源码如下(关于刷屏具体的逻辑向这里不再说明):

```
103 //*****  
104 功能: I2C口中断服务函数  
105 备注: 调用回调函数 为了方便直接放在这里了  
106 *****  
107 void I2C1_IRQHandler(void)  
108 {  
109     // check interrupt  
110     if(I2C1_CTRL->DONE_FLG==1)  
111     {  
112         I2C1_CTRL->DONE_FLG = 0;  
113  
114         num_32byte++;  
115         if(num_32byte<num_32byte_all)  
116         {  
117             i2c1_write_irq(I2C_ID_OLED, 0x01, 0x40, &p_image[start_y*X_WIDTH+num_32byte*32], 32);  
118             dbg_printf("num_32byte:%x\r\n",num_32byte);  
119         }  
120         else //一行结束  
121         {  
122             num_32byte=0;  
123             start_y++;  
124             if(start_y<end_y)  
125             {  
126                 // enable IRQ  
127                 NVIC_DisableIRQ(I2C1_IRQn);  
128                 oled_wrcmd(0xb0+start_y);  
129                 oled_wrcmd(0x01);  
130                 oled_wrcmd(0x10);  
131                 NVIC_EnableIRQ(I2C1_IRQn);  
132                 i2c1_write_irq(I2C_ID_OLED, 0x01, 0x40, &p_image[start_y*X_WIDTH+num_32byte*32], 32);  
133                 dbg_printf("start_y:%x\r\n",start_y);  
134             }  
135             else  
136             {  
137                 dbg_printf("image finish\r\n");  
138                 return;  
139             }  
140         }  
141     }  
142     if(I2C1_CTRL->ERR_FLG==1)  
143     {  
144         dbg_printf("ERR\r\n");  
145         I2C1_CTRL->ERR_FLG = 0;  
146     }  
147     //I2C1_CTRL->RSTB=0; //reset  
148 }  
149 //*****
```

下载代码后可以看到在开发板上正常刷新出图片来,如下:



SYDTEK

具体项目见 SDK 中的如下工程: Source Code\SYD8811_peripheral_misc\I2C_OLED12864_buff_irq

因为 I2C 的写和读中断源是一样的,所以这里不再举例读数据的中断例程!