

实验七 I^2C 总线

杨庆龙
1500012956

2018.5.9

1 实验目的

1. 了解 I^2C 总线的基本原理
2. 掌握使用C8051F020作为主设备进行 I^2C 通信的方法
3. 了解时钟芯片DS1307的使用方法

2 实验原理

2.1 I^2C 总线的基本原理

2.1.1 物理结构

I^2C 的总线标准有两条信号，其中SDA为双向数据信号，SCL为双向时钟信号。数据信号只在时钟信号为低时变化，在时钟上升沿采样。控制信号在时钟信号的为高时变化，下降沿表示起始位，上升沿表示停止位。

2.1.2 通信模式

I^2C 数据传输由主设备产生一个起始位开始，如何传递7个地址位，指定操作设备。接下来一位是读写位，0表示写入，1表示读入。接下来由从设备驱动，如果从设备地址与主设备地址相同，从设备将把SDA信号拉低，表示确认传输。

2.2 控制寄存器

- BUSY:11表示设备忙
- ENSMB:1表示设备激活
- STA:起始位控制，设置为1时，会发送起始位。
- STO:阶数为控制，设置为1时，会发送停止位。
- SI:中断标识
- AA:应答标识
- FTE:定时总线释放
- TOE:更短时间的定时总线释放

SMB0DAT寄存器用于存放即将发送的 I^2C 数据，软件只有在SI位有效的情况下才能访问这个寄存器。SMB0ADR寄存器用于存放设备的地址

Table 1: SMB控制寄存器

D7	D6	D5	D4	D3	D2	D1	D0
busy	ensmb	sta	sto	si	aa	fte	toe

3 实验内容

3.1 显示时钟

SMB的SDA和SCL与SPI的部分管脚共用，因此需要检测跳线是否接到了偏下的两个接线柱。

4 源码

```
1  #include <C8051F020.h>
2  #include <stdio.h>
3  #include "../includes/time.h"
4  #include "../includes/i2c.h"
5  #include "../includes/communicate.h"
6
7  void smbus_int(void) interrupt 7 {
8      bit FAIL = 0;
9      static unsigned char i;
10     switch(SMBOSTA) {
11         case SMB_START:
12         case SMB_RP_START:
13             SMBODAT = smb_buf[0];
14             STA = 0;
15             i = 0;
16             break;
17         case SMB_MTADDACK:
18             SMBODAT = smb_buf[1];
19             break;
20         case SMB_MTDBACK:
21             if(i < smb_len) {
22                 SMBODAT = smb_buf[i+2];
23                 i ++;
24             }
25             else {
26                 result = S_OVER;
27                 STO = 1;
28             }
29             break;
30         case SMB_MRADDACK:
31             if(smb_len == 1)
32                 AA = 0;
33             else
34                 AA = 1;
35             break;
36         case SMB_MRDBACK:
37             if(i < smb_len) {
38                 smb_buf[i+1] = SMBODAT;
39                 i ++;
40                 AA = 1;
41             }
42             if(i >= smb_len)
43                 AA = 0;
44             break;
45         case SMB_MRDBNACK:
46             smb_buf[i+1] = SMBODAT;
47             STO = 1;
48             AA = 1;
```

```

49     result = R_OVER;
50     break;
51     case SMB_MTADDNACK:
52     case SMB_MTDBNACK:
53     case SMB_MTARBLOST:
54     case SMB_MRADDNACK:
55     FAIL = 1;
56     break;
57     default:
58     FAIL = 1;
59 }
60 if(FAIL) {
61     SMBOCN &= -0x40;
62     SMBOCN |= 0x40;
63     STA = 0;
64     STO = 0;
65     AA = 0;
66     result = SMB_FAIL;
67     FAIL = 0;
68 }
69 SI = 0;
70 }
71
72 void main() {
73     int i;
74     int counter;
75     int flag;
76     unsigned char value;
77     unsigned char c;
78     WDTCN = 0xDE;
79     WDTCN = 0xAD;
80     sysclk_init();
81     i2c_port_init();
82     uart0_init();
83     TIO = 1;
84     i2c_init();
85     SI = 0;
86     EIE1 |= 0x02;
87     EA = 1;
88     printf("start!\r\n");
89     smb_transmit(0,0);
90     smb_receive(1);
91     if (smb_buf[1] & 0x80) {
92         unsigned char b = smb_buf[1];
93         smb_buf[2] = b & 0x7F;
94         smb_transmit(0,1);
95     }
96     i = 0;
97     counter = 0;
98     while(1) {
99         c = getchar();
100         switch(c) {
101             case 'y':
102                 scanf("%02bx",smb_buf + 2);
103                 smb_transmit(6,1);
104                 break;

```

```

105         case 'M':
106             scanf("%02bx",smb_buf + 2);
107             smb_transmit(5,1);
108             break;
109         case 'd':
110             scanf("%02bx",smb_buf + 2);
111             smb_transmit(4,1);
112             break;
113         case 'h':
114             scanf("%02bx",smb_buf + 2);
115             smb_transmit(2,1);
116             break;
117         case 'm':
118             scanf("%02bx",smb_buf + 2);
119             smb_transmit(1,1);
120             break;
121         case 's':
122             scanf("%02bx",smb_buf + 2);
123             smb_transmit(0,1);
124             break;
125         case 'p': flag = 1;
126     }
127     if(flag)
128         break;
129 }
130 while(1) {
131     smb_transmit(0,0);
132     smb_receive(8);
133     delay(100);
134     printf("%02bx-",smb_buf[7]);
135     printf("%02bx-",smb_buf[6]);
136     printf("%02bx□",smb_buf[5]);
137     //printf("%d%d:",smb_buf[5] >> 12,(smb_buf[5] >> 8) & 0xf);
138     printf("%02bx:",smb_buf[3]);
139     printf("%02bx:",smb_buf[2]);
140     printf("%02bx",smb_buf[1] & 0x7f);
141     printf("\r\n");
142 }
143 }

```