

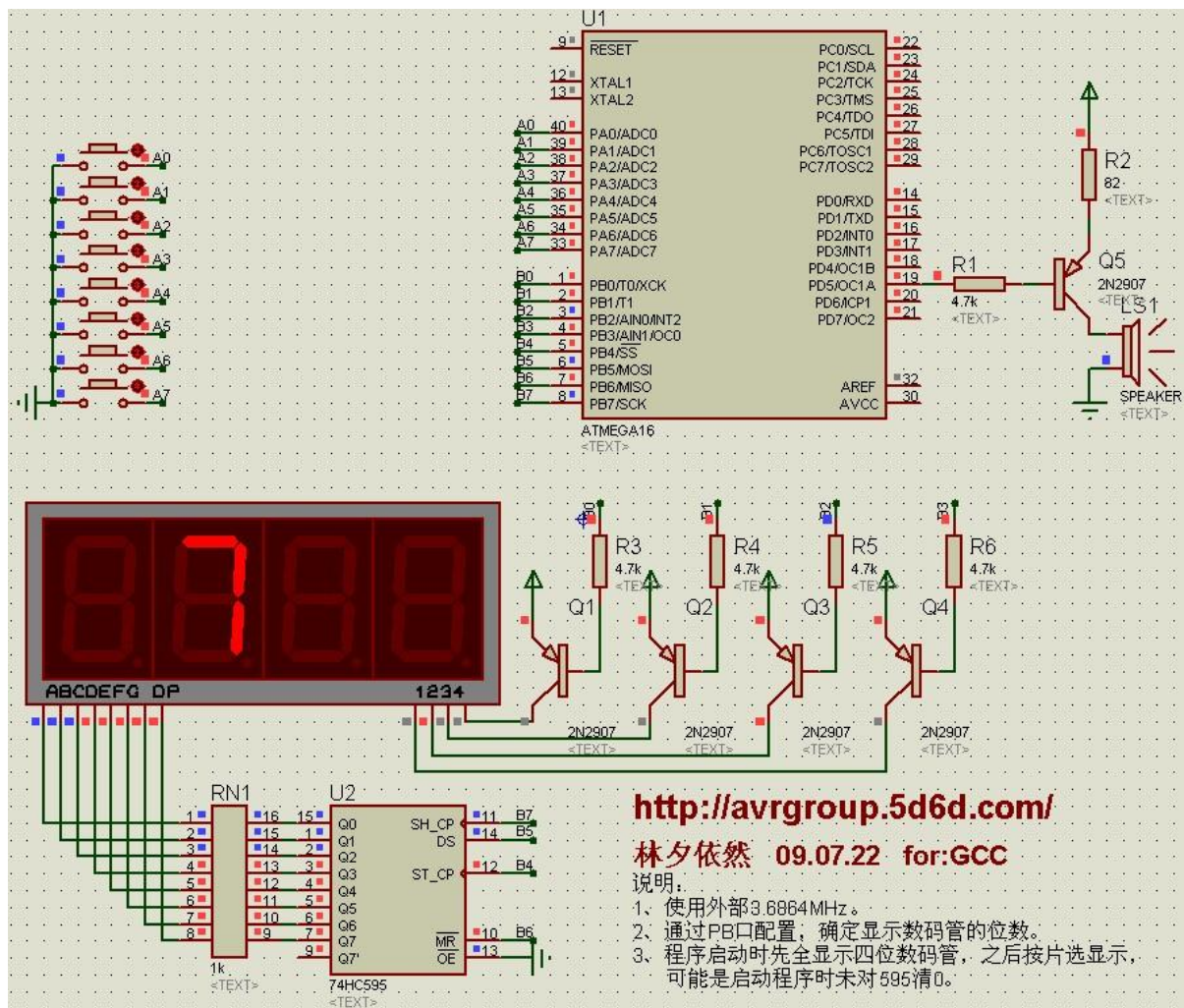
实验 12: SPI 实验 (数码管位显示控制) (GCC)

更多笔记: <http://bbs.armavr.com/>

一、程序结构



二、仿真效果



三、程序源码

1、main.c

```

1.  /*****
2.  Platform: AVR mega16 学习板 (www.iccavr.com)

```

- 3. Project : 实验十二: SPI 实验 (数码管位显示控制) (GCC)
- 4. Clock F : 3. 6864M
- 5. Software: Wi nAVR- 20090313+Proteus7. 4
- 6. Author : 林夕依然
- 7. Versi on : 08. 12. 10
- 8. Updata : 09. 02. 26 模块化
- 9. 09. 05. 04 增加 proteus 仿真模型, 仿真通过
- 10. 09. 07. 22 Wi nAVR- 20090313 调试通过
- 11. comments:
- 12. 1、以学习板 4 位数码管为硬件电路
- 13. 2、通过 PB 口配置, 确定显示数码管的位数
- 14. 3、进行此实验请插上 JP1 的所有 8 个短路块, JP6(SPI_EN) 短路块。
- 15. 4、参考: ... mcu\I CCAVR 学习板\旧版例程\M16 学习板实验程序\4- LED 数码管程序
- 16. *****/
- 17. #i ncl ude <avr/i o. h>
- 18. #i ncl ude <uti l /del ay. h>
- 19. #i ncl ude "spi . h"
- 20. #defi ne uchar unsigned char
- 21.
- 22. /*数码管 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 显示*/
- 23. uchar tab[]={0XC0, 0XF9, 0XA4, 0XB0, 0X99, 0X92, 0X82, 0XF8, 0X80, 0X90};
- 24.
- 25. /*-----*/
- 26. 函数名称: voi d HC595out(uchar i)
- 27. 函数功能: 74HC595 数据输出显示
- 28. 参 数:
- 29. 返 回 值: 无
- 30. -----*/
- 31. voi d HC595out(uchar i)
- 32. {
- 33. PORTB&=~(1<<4); //等待数据传输, 595 存储寄存器数据不变
- 34. SPI MasterTransmi t(i); //传输数据
- 35. PORTB|=(1<<4); //595 移位寄存器的数据进入数据存储寄存器, 并在数码管上显示
- 36. }
- 37.

```
38. int main(void)
39. {
40.     uchar i=0;
41.     DDRA =0X00; //上拉
42.     PORTA=0XFF;
43.     DDRB =0XFF; //低四位输出低电平，使四位数码管均作显示
44.     DDRC =0X00; //上拉
45.     PORTC=0XFF;
46.     DDRD =0X00; //上拉
47.     PORTD=0XFF;
48.
49.     SPImasterInit(); //SPI 初始化
50.     //PORTB=0XF7; //只有第 1 个数码管显示
51.     PORTB=0XFB; //只有第 2 个数码管显示
52.     //PORTB=0XFD; //只有第 3 个数码管显示
53.     //PORTB=0XFE; //只有第 4 个数码管显示
54.
55.     while(1)
56.     {
57.         i++;
58.         if(i>9)
59.         {
60.             i=0;
61.         }
62.         HC595out(tab[i]);
63.         _delay_ms(800);
64.     }
65. }
```

2、spi.c

```
1. #include <avr/i_o.h>
2. #define uchar unsigned char
3.
4. /*-----
```

```
5. 函数名称: void SPImasterInit(void)
6. 函数功能: SPI 初始化为主机
7. 参 数:
8. 返 回 值: 无
9. SPI 控制寄存器介绍
10. SPIE SPE DORD MSTR CPOL CPHA SPR1 SPRO
11. SPIE: 中断使能
12. SPE : SPI 使能
13. DORD: 数据次序 1 为 LSB 首先发送, 0 为 MSB 首先发送
14. MSTR: 主从选择 1 为主, 0 为从 MSTR 置位时选择主机模式, 否则为从机. 如果 MSTR 为"1",
15. SS 配置为输入, 且被拉低, 则 MSTR 被清零, 寄存器 SPSR 的 SPIF 置位. 用户必须重新
16. 设置 MSTR 进入主机模式。
17. CPOL: 时钟极性见 ATMEGA16L(中文) P130
18. CPHA: 时钟相位
19. SPR1, SPRO: SPI 时钟速率选择 1 与 0 SPR1 和 SPRO 对从机没有影响。
20. ----- */
21. void SPImasterInit(void) //SPI 初始化
22. {
23.   DDRB|=(1<<7)|(1<<5); //设置 SCK, MOSI 为输出, 其他为输入
24.   SPCR|=(1<<SPE)|(1<<MSTR)|(1<<SPR1)|(1<<SPRO); //使能 SPI, 主模式, Fck/128
25. }
26. /*-----
27. 函数名称: void SPIMasterTransmit(uchar TranData)
28. 函数功能: SPI 数据传输
29. 参 数:
30. 返 回 值: 无
31. ----- */
32. void SPIMasterTransmit(uchar TranData)
33. {
34.   SPDR=TranData; //数据放入 SPI 寄存器中
35.   while(!(SPSR&(1<<SPIF))); //等待数据传输完毕
36. }
```

3、spi.h

1. #define uchar unsigned char
2. void SPImasterInit(void);
3. void SPIMasterTransmit(uchar TranData);

四、完整项目文件下载

<http://bbs.armavr.com/thread-914-1-1.html>