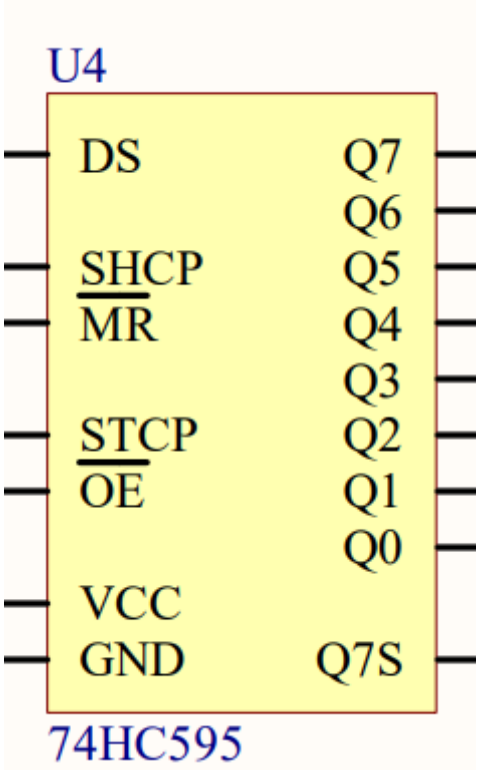
> 此代码参考自龙顺宇的《深入浅出STM8单片机入门、进阶与应用实例》118页

芯片介绍

芯片图



引脚

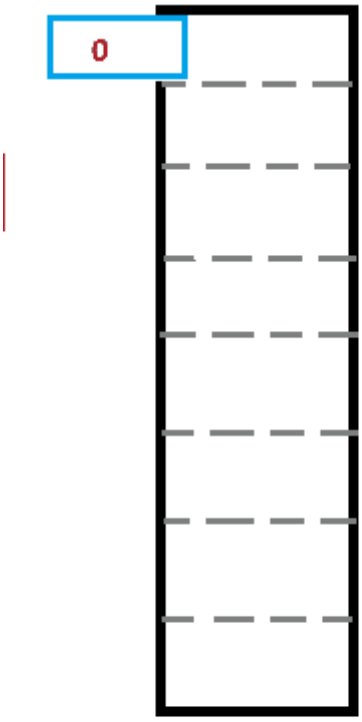
DS(SER、SDI)：串行数据输入引脚
OE：输出使能控制脚，它是低电才使能输出，所以接GND
RCK(STCP、RCLK)：存储寄存器时钟输入引脚。上升沿时，数据从移位寄存器转存到存储寄存器。
SCK(SHCP、SCLK)：移位寄存器时钟引脚，上升沿时，移位寄存器中的bit 数据整体后移，并接受新的bit（从SER输入）。
MR(SCLR)：低电平时，清空移位寄存器中已有的bit数据，一般不用，接 高电平即可。
DIO(Q7S、SDO)：串行数据出口引脚。当移位寄存器中的数据多于8bit时，会把已有的bit“挤出去”，就是从这里出去的。用于595的级联。
Q0~Q7：并行输出引脚

介绍

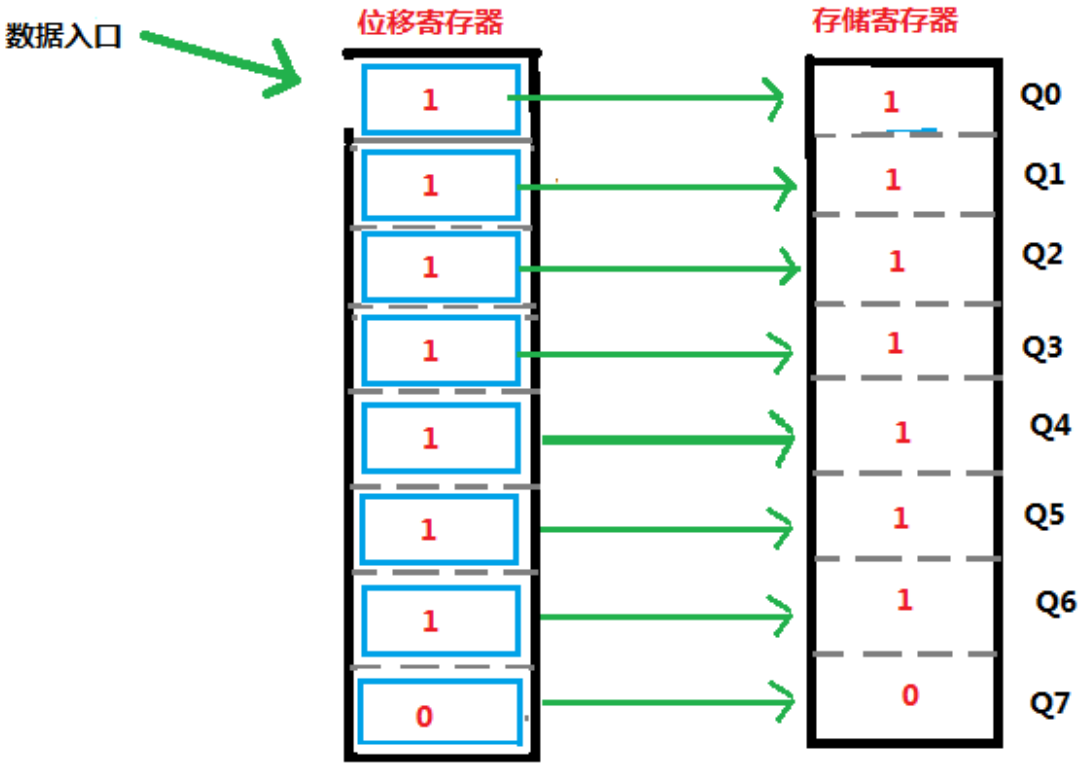
注意：第一个从SER送入的比特会从Q7出去。
具体流程如下所示：
假如，我们要将二进制数据0111 1111 输入到595的移位寄存器中，下面来上一张动态图，模拟了前2个位输入的情景。

- 数据从SER进入，每经过一次SCK上升沿，数据后移一位。所以我们按照高位在前的顺序输入。
- 0--->SER，数据送到SER：移位寄存器=**** ****
- SCK接收到上升沿信号，移位寄存器存储数据后移，并接收数据：移位寄存器=**** ***0
- 1--->SER，数据送到SER：移位寄存器=**** ***1
- SCK接收到上升沿信号，移位寄存器存储数据后移，并接收数据：移位寄存器=**** **01

- ……
- SCK接收到上升沿信号，移位寄存器存储数据后移，并接收数据：移位寄存器=0111 1111

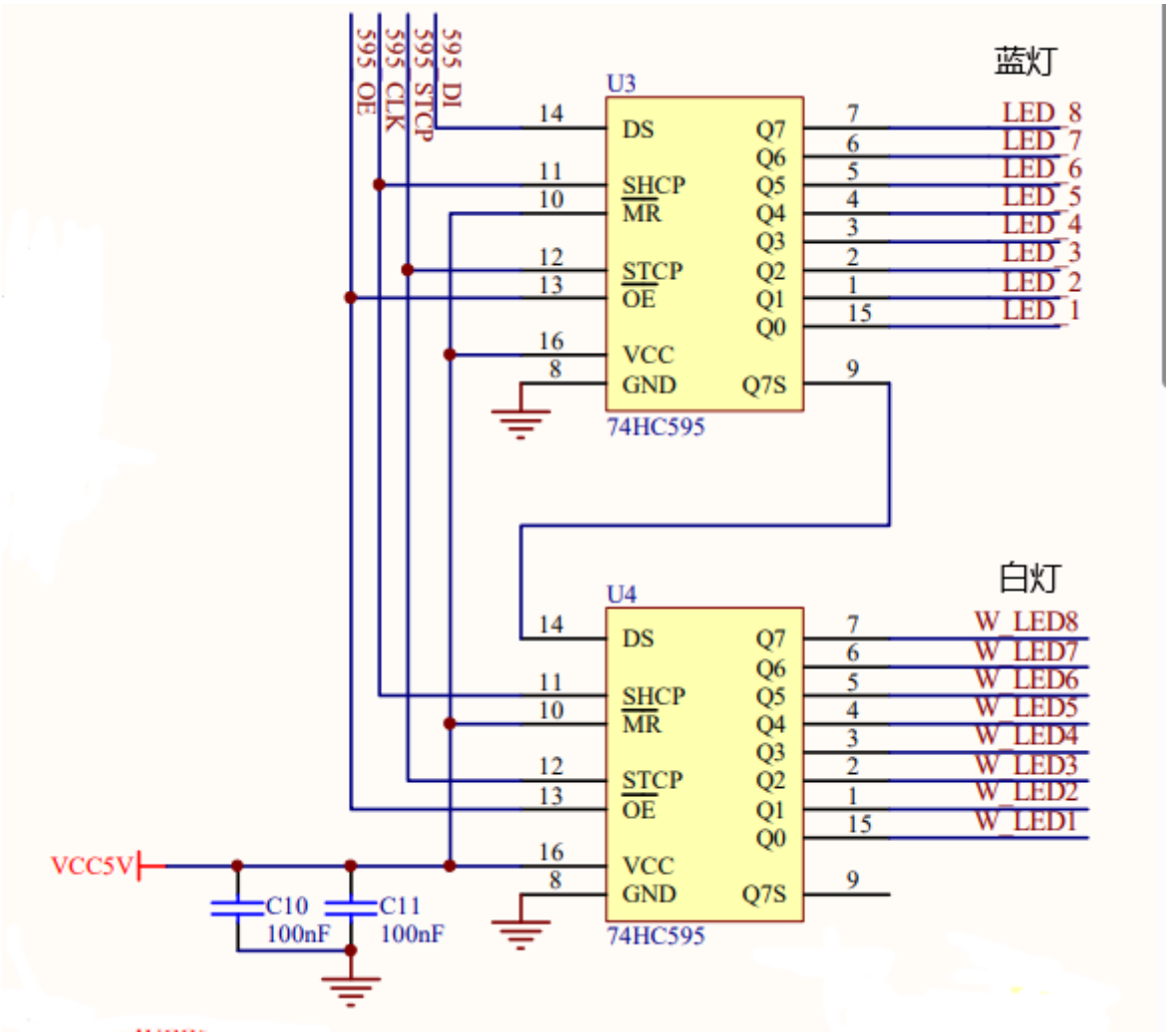


- RCK接收到上升沿信号，数据从移位寄存器挪动到存储寄存器：存储寄存器=0111 1111
  数据会一直从Q[7:0]输出，直到74HC95断电，或者存储寄存器数据被新数据覆盖为止。



74HC595级联

如图是两个74HC595级联



从上面我们已经知道，595会接收来自SER(DS)的数据，并随着SCK每次接收到上升沿数据，将移位寄存器数据后移并放入新数据。
那么，在我们将移位寄存器的8个位填满后，再往移位寄存器中塞一个会怎么样?
移位寄存器的最后一个位数据会被挤出去，从DIO（Q7S)输出的。
如果我们把第一个595的9脚连接到第二个的串行数据输入脚SER(DS)。

所以如果我们想要将U3输出0000 0001，U4输出1000 0000那么顺序应该是：

- 先发送1000 0000,再发送0000 0001，高位在前。
- 1--->U3的SER，数据送到U3的SER：U3的移位寄存器=**** ****

  (U3和U4,没有标注就是U3和U4的意思）SCK接收到上升沿信号，移位寄存器存储数据后移，并接收数据：
  - U3的移位寄存器=**** ***1
- 0--->U3的SER，数据送到U3的SER：U3的移位寄存器=**** ***1

  SCK接收到上升沿信号，移位寄存器存储数据后移，并接收数据：U3的移位寄存器=**** **10
- ……（1000 0000已经发送完成）
- 0--->U3的SER，数据送到U3的SER：U3的移位寄存器=1000 0000，U4的移位寄存器=**** ****

  SCK接收到上升沿信号，移位寄存器存储数据后移，并接收数据：
  - U3的移位寄存器=0000 0000，U4的移位寄存器=**** ***1
- 0--->U3的SER，数据送到U3的SER：U3的移位寄存器=0000 0000，U4的移位寄存器=**** ***1

SCK接收到上升沿信号，移位寄存器存储数据后移，并接收数据：

- U3的移位寄存器=0000 0000，U4的移位寄存器=**** **10

- ……（0000 0001已经发送完成，U3的移位寄存器=0000 0001，U4的移位寄存器=1000 0000）
- RCK接收到上升沿信号，数据从移位寄存器挪动到存储寄存器：

  U3的存储寄存器=0000 0001，U4的存储寄存器=1000 0000

代码

```
/*    添加包含芯片的头文件    */
#include<iostm8s105k6.h>
#include <stdint.h>
#include <stdio.h>

#define DIO PC_ODR_ODR5          //串行数据输入
#define RCLK PC_ODR_ODR4         //锁存控制信号
#define SCLK PC_ODR_ODR3         //时钟脉冲信号
#define u8 unsigned char
void led_init(){
  //OE使能74HC595 1101 1111
  PE_DDR|=0x20; //PE5
  PE_CR1|=0x20;
  PE_CR2|=0x20;
  PE_ODR &=0XDF;
  //初始化DI,RCLK,PCLK     PC3-5
  PC_DDR|=0x38;
  PC_CR1|=0x38;
  PC_CR2|=0x38;
  PC_ODR &=0XCF;
}

void delay(unsigned char ms)
{
  unsigned char i, j, k;

  i = 7;
  j = 1;
  k=1;
  do{
    do{
      do{
        while(--k);
      }while(--j);
    }while(--i);
  }while(--ms);
}

/*****************************************************************/
//单字节数据串行移位函数LED. ,Or(),有形参outdata用于传入实际数据
//无返回值
/*****************************************************************/
void LED_OUT(u8 outdata)
```

```
{
  u8 i;
  for(i=0;i<8;i++ )//循环8次
  {
    if (outdata & 0x80)//逐一取出最高位
      DIO= 1;//送出“1”
    else
      DIO= 0;//送出“0”
    outdata<<= 1 ;//执行左移-位操作
    //SCLK产生上升沿
    SCLK= 0;
    SCLK= 1;

  }
}

/*********************************************************************/
//点灯函数LED_Display(),有形参w_led,与b_led用于表示白灯与蓝灯的亮灭情况,无返回值
/*********************************************************************/

void LED_Display(u8 w_led,u8 b_led){
  LED_OUT(w_led);//白灯
  LED_OUT(b_led);//蓝灯
  //RCLK产生上升沿
  RCLK=0;
  RCLK= 1;
}

/*********************************************************************/
//点灯代码，实现流水灯，蓝灯轮流闪烁，然后白灯轮流闪烁
/*********************************************************************/
int main(void)
{
  u8 b_led=0x01;
  u8 w_led=0x01;
  u8 i=0;
  led_init();
 // LED_Display(0x00,0xff);
  while(1){
    for(i=0;i<6;i++){
      LED_Display(0xff,b_led<<i);//白灯低电平亮,蓝灯高电平亮
      delay(1);
    }

    for(i=0;i<6;i++){
      LED_Display(~(w_led<<i),0x00);//白灯低电平亮,蓝灯高电平亮
      delay(1);
    }


  }
}
```