数码管显示帮助文档

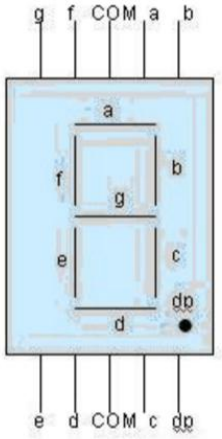
驱动芯片 74hc595

数码管用于显示各种数据贼方便，两个 595 芯片怎样只用了几个管脚，就驱动了八位数码管，在各类芯片中很有代表性，初学者确实很蒙比，但是看完就觉得贼简单。

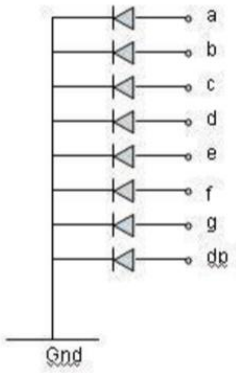
PS:想快速知道这玩意怎么用直接往后翻，，，可以不知道原理直接用【沧桑】

一． 数码管是什么鬼

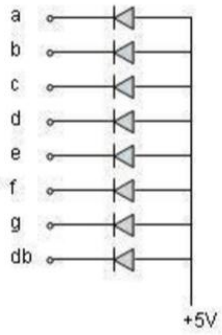
数码管可以理解为 8 个 led 小灯，上图



引脚图



共阴极



共阳极

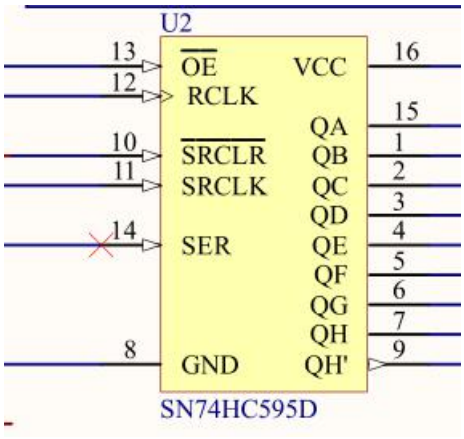
74HC595是硅结构的CMOS器件， 兼容低电压TTL电路，遵守JEDEC标准。 74HC595是具有8位移位寄存器和一个存储器，三态输出功能。如果两个时钟连在一起，则移位寄存器总是比存储寄存器早一个脉冲。 移位寄存器有一个串行移位输入（Ds），和一个串行输出（Q7'）,和一个异步的低电平复位，存储寄存器有一个并行8位的，具备三态的总线输出，当使能OE时（为低电平），存储寄存器的数据输出到总线。

。。

。。。。。。。。。。。。。。。

。。

。。还是上图。。



原理图这玩意就是电路图，没啥大区别

符号 引脚 描述

QA…QH 8 位并行数据输出 //这八个就是连到八个 led 上，控制显示字符的，是输出端

GND 第 8 脚 地 //接地端没啥好说的

QH’ 第 9 脚 串行数据输出 //这个引脚用来串联第二片 595，暂时不说

SRCLR 第 10 脚 主复位（低电平） //这个引脚低电平复位，所以我们永远拉高不复位

SRCLK 第 11 脚 移位寄存器时钟输入 //当这个引脚从 0->1 时，移位寄存器移位，重要

RCLK 第 12 脚 存储寄存器时钟输入 //当这个引脚从 0->1 时，更新储存寄存器，并且输出,重要

OE 第 13 脚 输出有效（低电平） //可以理解为使能端口，拉低时启用数码管输出

SER 第 14 脚 串行数据输入 //数据就是从这个地方进去

VCC 第 16 脚 电源 //电源没啥好说的

那么，你们懂了没，很好，没懂，，，，，

三．先瞅瞅原理

那我们继续，简单讲下原理，就是把一串数据一个一个输入到 595 里，然后转换成并行输出。

比如我要输出一个数字 2，就是让 a, b, g, e, d 亮，由于我们设计的是共阴极数码管，所以把 a, b, d, e, g 置高电平就是亮，高电平就是 1，用八位数据表示就是：11011010, 这个简单。

然后高潮就来了，我们为了节约管口，不能使用 8 个管口控制这八个数，我们省一省，用一个管口一个一个输入进去。这里要讲到一个寄存器。

寄存器可以理解为盒子，八位寄存器就是有八个小方格的盒子。

简陋了点，就像这样 0_0. . .
我们用到的第一个寄存器叫移位寄存器。SER14 这个数据输入管脚就是连着这个寄存器的最后一位。（不知道 SER14 是什么的往上翻翻）

然后我们先输入 11011010 的最高位，就是相应管脚把 SER14 置 1，就是像这样

							1

这个简单，输入一个数据后，我们需要。。。。。。。移位！

SRCLK11 这个引脚从 0->1 时，移位寄存器移位，我们就先置 1 再置 0（虽然是先 1 再 0，但是因为本来就是 0，所以是一个先上升再下降的过程），然后寄存器就像这样

						1	

然后再给 SER14 置 1，就变成这样

						1	1

1	1	0	1	1	0	1	0

1	1	0	1	1	0	1	0

1	1	0	1	1	0	1	0
1	1	0	1	1	0	1	0

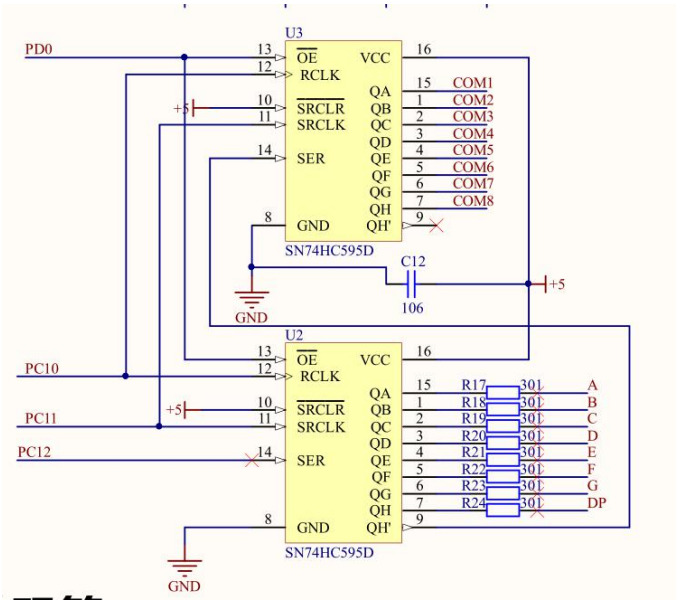
第一层的储存寄存器控制着输出，对应着 QA…QH 8 位并行数据输出 ，这样就能并行输出了
然后储存寄存器保持不变,移位寄存器再一位一位的接受新的数据,接受八个后再更新,,,,,,
如此这般

四．两个 595 串联的简单原理

你看一个 595 只能控制 QA…QH 8 位，要控制八个数码管难道要 8 个 595？。。。不存在的，哪来那么多地方焊 595，两个就够了。

我们再用一个 595 控制 8 位，当做开关控制 8 个 LED 的亮灭，选中哪一位，哪一位就亮，其他七位就灭，如果同时想输出八位，就是快速轮流切换八位，虽然同一时间里只有一个灯再亮，但是由于人眼的视觉暂留，会感觉是同时亮的。。。（别说你们听不懂）

上原理图



那就瞅瞅这个，SER14 是串行数据输入管脚，但是上方的那个 595 芯片的 SER14 脚连接的是下面 595 的 QH’9 管脚，看看前面写的管脚说明，QH’9 是用来串联两块 595 芯片的////////比如

SER14 串行给数据，给到 8 位就满了，如果给第 9 位的话，整体数据会在移位寄存器里再移动一格，也就是最前头一位会通过 QH’9 连到上面一片 595 的 SER14，然后进入它的移位寄存器，这样移位寄存器可视为有 16 位，等到 16 位都填满之后，再一起更新储存寄存器，然后实现 16 位并行输出。

感觉爬行文明我 bb 有点多了。。。。

五．程序的实现

程序主要是几个函数实现的，简单讲一讲（MMP 去年我们的学长就给我们一个 595 的破例程注释原理都没有都是抱着百度啃的今年为啥要给你们写这么多 emmm）

1. hc595init 函数

这就是初始化需要用的几个管脚，用来控制 595

```
void hc595_init(void) {
  GPIO_InitTypeDef GPIO_InitStructure;
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD, ENABLE);
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;          //管脚2
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  //管脚输出速度
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;  // GPIO设置为推挽输出
  GPIO_Init(GPIOD, &GPIO_InitStructure);

  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC, ENABLE);
  GPIO_InitStructure.GPIO_Pin =
      GPIO_Pin_10 | GPIO_Pin_11 | GPIO_Pin_12;       //管脚2
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  //管脚输出速度
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;  // GPIO设置为推挽输出
  GPIO_Init(GPIOC, &GPIO_InitStructure);
}
```

初始化了 PTD0， PTC10， PTC11，PTC12 这四个管脚

2. send_595(unsigned char dat)

这个函数就是通过 SER14 把数据串行输入的，再通过 SRCLK11 进行移位

```
void Send_595(unsigned char dat) {
  unsigned char i;
  OE(0);
  for (i = 0; i < 8; i++) {
    if (dat & 0x80) {
      SER(1);
    } else {
      SER(0);
    }
    SRCLK(1);
    SRCLK(0);
    dat <<= 1;
  }
}
```

从上往下看，这个需要 输入一个八位数据作为参数，比如上面的 11011010

OE（0）使能数码管输出

然后是一个八次的循环，也就是输入一位移动一位，dat&0x80,这个你们可以自己算一下，就是取 date 这个数据的最高位，如果是 1，就给 SER 置 1，是零就置 0

然后对 SRCLK 操作，先 1 再 0，使移位寄存器移位

然后左移 dat，使原来的第二位变成最高位，这样下一个循环的时候 dat&0x80 刷新

上面这个函数是整个 595 芯片的核心，看一遍理解不了就要多瞅几遍，百度百度。

3.displayscan(unsigned char r, unsigned char data)

这个是真正要用到的函数，用于显示刷新，r 操作的是位，就是个十百千的那个位，date 操作的是段，就是八个 led 中的哪一个，用于显示具体数字字母

上图

```
void DisplayScan(unsigned char r, unsigned char data) {
  Send_595(row[r]);
  Send_595(data);
  RCLK(1);
  RCLK(0);
}
```

这个函数里面调用了两个 send595 函数，也就是分两次输入一共 16 位数据，前八位控制原理图上面一个 595 芯片，后八位控制下面那个 595 芯片

然后再操作 RCLK 管口来更新储存寄存器，用于刷新显示

以下是你们要的智障教学

使用前先调用初始化函数

```
void main()
{

  hc595_init();        //数码管初始化
```

再调用 displayscan 函数

```
while(1)
{
    DisplayScan(row[1],data_table[1]);          // 显示数字
    Delay1s();
}
```

DisplayScan(row[x],data_table[y]);

的意思就是在第 x 位上显示 y

x 取 1-8

y 取 1-16（即 0-9 和 A-F）

然后再加个延时 delay1s();

emmmmmmmmmmmmmmmmmmmmm 这个不懂可以戳我，但你不知道我是谁。。