

# RTC实时时钟

RTC是个独立的定时器。RTC模块拥有一个连续计数的计数器，在相应的软件配置下，可以提供时钟日历的功能。修改计数器的值可以重新设置当前时间和日期 RTC还包含用于管理低功耗模式的自动唤醒单元。在断电情况下 RTC仍可以独立运行 只要芯片的备用电源一直供电,RTC上的时间会一直走。

两个 32 位寄存器包含二进制十进制格式 (BCD) 的秒、分钟、小时（12 或 24 小时制）、星期几、日期、月份和年份。此外，还可提供二进制格式的亚秒值。系统可以自动将月份的天数补偿为 28、29（闰年）、30 和 31 天。

上电复位后，所有RTC寄存器都会受到保护，以防止可能的非正常写访问。

无论器件状态如何（运行模式、低功耗模式或处于复位状态），只要电源电压保持在工作范围内，RTC使不会停止工作。

## RTC特性

RTC 单元的主要特性如下（参见图 222：RTC 框图）：

- 包含亚秒、秒、分钟、小时（12/24 小时制）、星期几、日期、月份和年份的日历。
- 软件可编程的夏令时补偿。
- 两个具有中断功能的可编程闹钟。可通过任意日历字段的组合驱动闹钟。
- 自动唤醒单元，可周期性地生成标志以触发自动唤醒中断。
- 参考时钟检测：可使用更加精确的第二时钟源（50 Hz 或 60 Hz）来提高日历的精确度。
- 利用亚秒级移位特性与外部时钟实现精确同步。
- 可屏蔽中断/事件：
  - 闹钟 A
  - 闹钟 B
  - 唤醒中断
  - 时间戳
  - 入侵检测
- 数字校准电路（周期性计数器调整）
  - 精度为 5 ppm
  - 精度为 0.95 ppm，在数秒钟的校准窗口中获得
- 用于事件保存的时间戳功能（1 个事件）
- 入侵检测：
  - 2 个带可配置过滤器和内部上拉的入侵事件
- 20 个备份寄存器（80 字节）。发生入侵检测事件时，将复位备份寄存器。

## RTC中断

| 中断名      | 中断类型号  |
|----------|--------|
| RTC闹钟中断  | EXTI17 |
| RTC唤醒中断  | EXTI22 |
| RTC入侵中断  | EXTI21 |
| RTC时间戳中断 | EXTI21 |

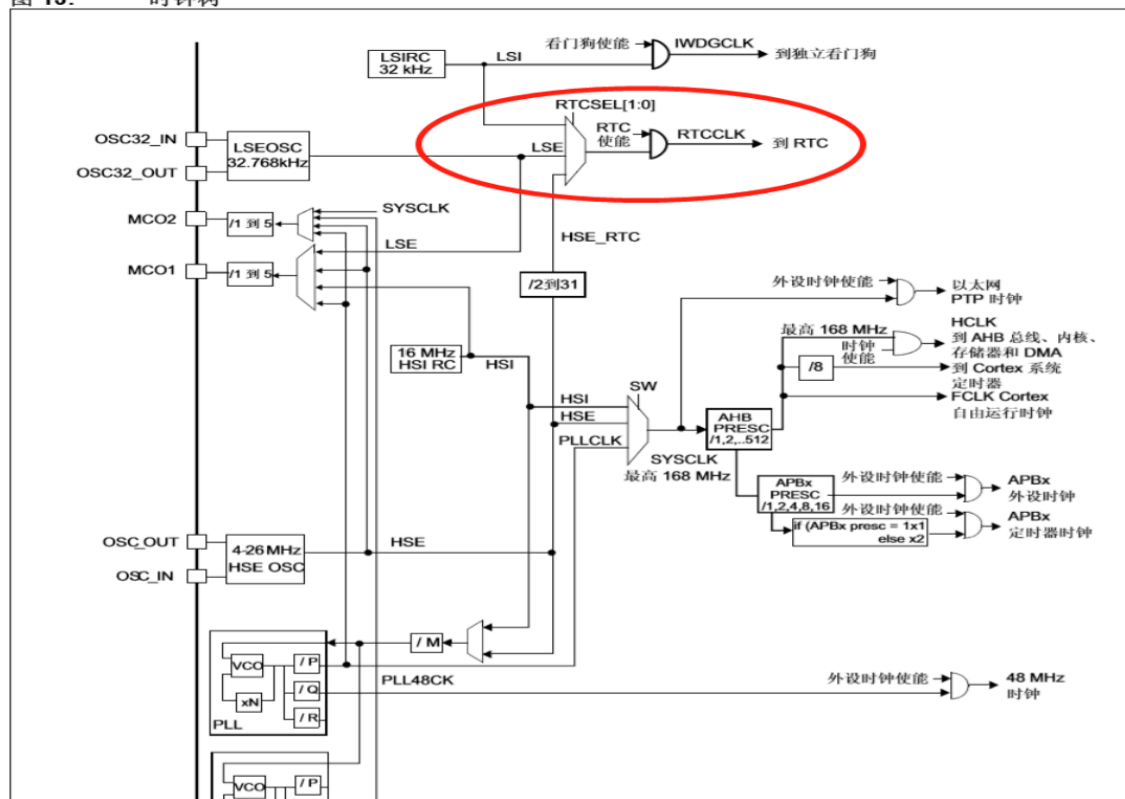
1. 将对应的类型中断号配置为中断模式并且使能
2. 配置NVIC中的（RTC\_Alarm IRQ、RTC\_WKUP IRQ、RTC\_STAMP IRQ）通道并且使能
3. 配置RTC以检测对应的RTC事件

## RTC时钟源

**RM0090**

## 复位和时钟控制 (RCC)

图 13. 时钟树



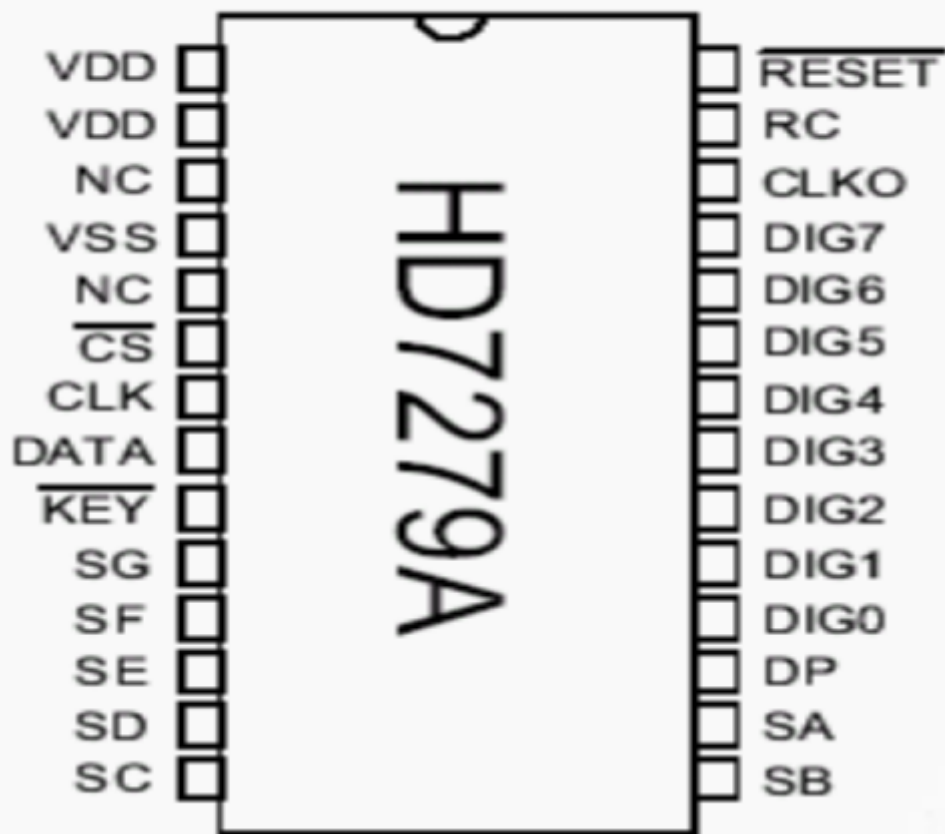
# HD7279芯片

HD7279A是一片具串行接口的芯片。它可完成LED显示、键盘接口的全部功能。

可同时驱动8位共阴式数码管的智能显示；还可连接多达64键的键盘矩阵

DIP/SOIC

图1



VDD: 正电源

CS: 片选

DATA: 串行数据输入/输出端

KEY: 按键有效输出端

SG-SA: 段g—段a驱动输出

DP: 小数点驱动输出

RC: RC振荡器连接端

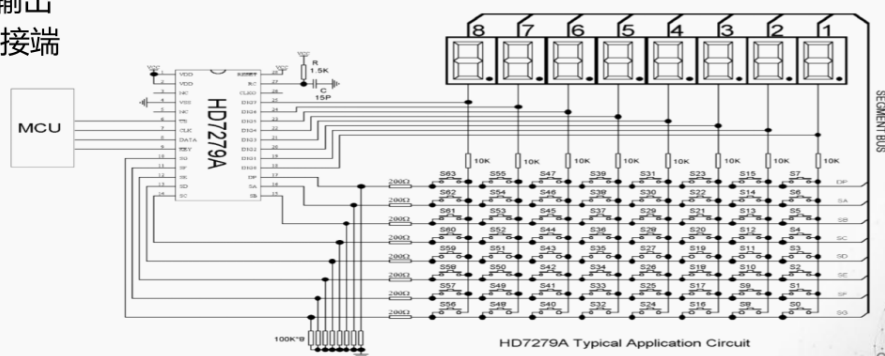
VSS: 地

CLK: 时钟输入端

CLKO: 振荡输出端

RES: 复位端

DIG0-7: 数位0-7驱动输出



HD7279A的控制指令分为两大类——纯指令和带有数据的指令。

## 纯指令

### 复位清楚指令A4H

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|
| 1  | 0  | 1  | 0  | 0  | 1  | 0  | 0  |

将所有显示清除、所有设置的字符的消隐、闪烁等也一起被清除。

## 带数据指令

### 下载数据按照方式0译码（80H - 87H）

高位：

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|
| 1  | 0  | 0  | 0  | 0  | a2 | a1 | a0 |

低位：

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|
| DP | X  | X  | X  | d3 | d2 | d1 | d0 |

其中：X 此处不译码；a2、a1、a0为位地址；d3、d2、d1、d0为显示数据

| a2 | a1 | a0 | 显示位 |
|----|----|----|-----|
| 0  | 0  | 0  | 1   |
| 0  | 0  | 1  | 2   |
| 0  | 1  | 0  | 3   |
| 0  | 1  | 1  | 4   |
| 1  | 0  | 0  | 5   |
| 1  | 0  | 1  | 6   |
| 1  | 1  | 0  | 7   |
| 1  | 1  | 1  | 8   |

| d0—d3为数据，收到此指令时，HD7279A按以下规则(译码方式0)进行译码，如下表： |    |    |    |    |        |
|--|----|----|----|----|--------|
| d3-d0(十六进制)                                  | d3 | d2 | d1 | d0 | 7段显示   |
| 00H  | 0  | 0  | 0  | 0  | 0      |
| 01H  | 0  | 0  | 0  | 1  | 1      |
| 02H  | 0  | 0  | 1  | 0  | 2      |
| 03H  | 0  | 0  | 1  | 1  | 3      |
| 04H  | 0  | 1  | 0  | 0  | 4      |
| 05H  | 0  | 1  | 0  | 1  | 5      |
| 06H  | 0  | 1  | 1  | 0  | 6      |
| 07H  | 0  | 1  | 1  | 1  | 7      |
| 08H  | 1  | 0  | 0  | 0  | 8      |
| 09H  | 1  | 0  | 0  | 1  | 9      |
| 0AH  | 1  | 0  | 1  | 0  | -      |
| 0BH  | 1  | 0  | 1  | 1  | E      |
| 0CH  | 1  | 1  | 0  | 0  | H      |
| 0DH  | 1  | 1  | 0  | 1  | L      |
| 0EH  | 1  | 1  | 1  | 0  | P      |
| 0FH  | 1  | 1  | 1  | 1  | 空(无显示) |

## 下载数据按照方式1译码（C8H - CFH）

高位：

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|
| 1  | 1  | 0  | 0  | 1  | a2 | a1 | a0 |

低位：

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|
| DP | X  | X  | X  | d3 | d2 | d1 | d0 |

其中：X 此处不译码；a2、a1、a0为位地址；d3、d2、d1、d0为显示数据

| a2 | a1 | a0 | 显示位 |
|----|----|----|-----|
| 0  | 0  | 0  | 1   |
| 0  | 0  | 1  | 2   |
| 0  | 1  | 0  | 3   |
| 0  | 1  | 1  | 4   |
| 1  | 0  | 0  | 5   |
| 1  | 0  | 1  | 6   |
| 1  | 1  | 0  | 7   |
| 1  | 1  | 1  | 8   |

此指令与上一条指令基本相同，所不同的是译码方式，该指令的译码按下表进行：

| d3-d0(十六进制) | d3 | d2 | d1 | d0 | 7段显示 |
|-------------|----|----|----|----|------|
| 00H         | 0  | 0  | 0  | 0  | 0    |
| 01H         | 0  | 0  | 0  | 1  | 1    |
| 02H         | 0  | 0  | 1  | 0  | 2    |
| 03H         | 0  | 0  | 1  | 1  | 3    |
| 04H         | 0  | 1  | 0  | 0  | 4    |
| 05H         | 0  | 1  | 0  | 1  | 5    |
| 06H         | 0  | 1  | 1  | 0  | 6    |
| 07H         | 0  | 1  | 1  | 1  | 7    |
| 08H         | 1  | 0  | 0  | 0  | 8    |
| 09H         | 1  | 0  | 0  | 1  | 9    |
| 0AH         | 1  | 0  | 1  | 0  | A    |
| 0BH         | 1  | 0  | 1  | 1  | b    |
| 0CH         | 1  | 1  | 0  | 0  | C    |
| 0DH         | 1  | 1  | 0  | 1  | d    |
| 0EH         | 1  | 1  | 1  | 0  | E    |
| 0FH         | 1  | 1  | 1  | 1  | F    |

## 下载数据不译码（90H - 97H）

高位：

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|
| 1  | 0  | 0  | 1  | 0  | a2 | a1 | a0 |

低位：

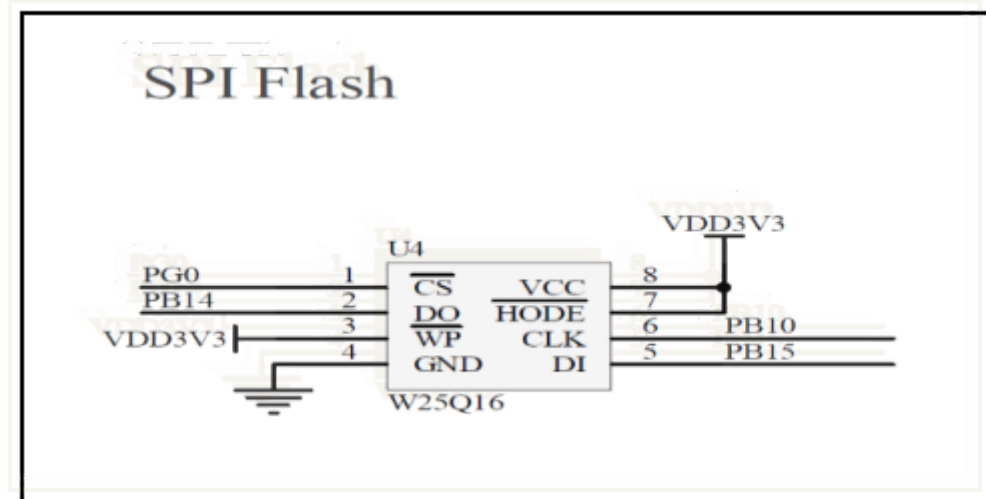
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|
| DP | A  | B  | C  | D  | E  | F  | G  |

其中，a2，a1，a0为位地址(参见‘下载数据且译码’指令)，A-G和DP为显示数据，分别对应7段LED数码管的各段。数码管各段的定义见下图。当相应的数据位为‘1’时，该段点亮，否则不亮。



## SPI接口

一种**高速的，全双工，同步**的通信总线。



Serial Peripheral interface，即串行外围设备接口。是Motorola首先在其MC68HCXX系列处理器上定义的。

在芯片的管脚上**只占用四根线**，具有节约芯片管脚、可为PCB节省空间等优点。

主要应用在 EEPROM、FLASH、实时时钟、AD转换器、数字信号处理器和数字信号解码器之间。

可以是8位数据传送，也可以是16位，可编程；既可以高位先行，也可以低位先行。

## W25Q16特点

1. SPI接口，容量为 16Mb ，也就是 2 M字节。
2. W25Q16将2M的容量分为32 个块（Block ）
3. 每个块（Block ）大小为64K 字节，有16个Sector
4. W25Q16的擦写周期多达10万次（读不损耗）
5. 具有 20 年的数据保存期限

在STM32F407的开发环境中，“变量”的值可以在“调试”时观察

## 与7279的区别

### 都是串行芯片

但7279：MCU通过软件，来模拟实现并变串、串变并的对HD7279的数据传输，采用半双工通信

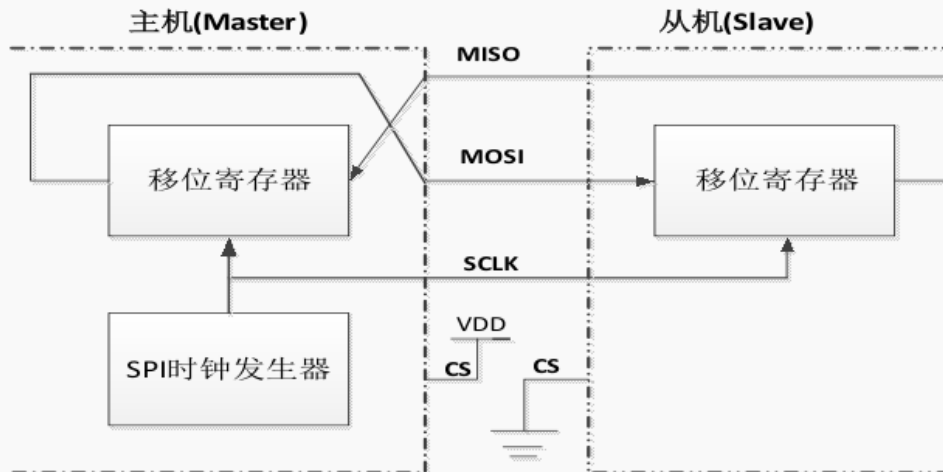
W25Q16：MCU通过硬件，来模拟实现并变串、串变并的对W25Q16的数据传输，可以采用全双工通信

## 命令识别码

| <b>INSTRUCTION<br/>NAME</b>               | <b>BYTE 1<br/>(CODE)</b> |
|---|--------------------------|
| Write Enable                              | <b>06h</b>               |
| Write Disable                             | <b>04h</b>               |
| Read Status Register-1                    | <b>05h</b>               |
| Read Status Register-2                    | <b>35h</b>               |
| Write Status Register                     | <b>01h</b>               |
| Page Program                              | <b>02h</b>               |
| Quad Page Program                         | <b>32h</b>               |
| Block Erase (64KB)                        | <b>D8h</b>               |
| Block Erase (32KB)                        | <b>52h</b>               |
| Sector Erase (4KB)                        | <b>20h</b>               |
| Chip Erase                                | <b>C7h/60h</b>           |
| Erase Suspend                             | <b>75h</b>               |
| Erase Resume                              | <b>7Ah</b>               |
| Power-down                                | <b>B9h</b>               |
| High Performance Mode                     | <b>A3h</b>               |
| Mode Bit Reset <sup>(4)</sup>             | <b>FFh</b>               |
| Release Power down or<br>HPM / Device ID  | <b>ABh</b>               |
| Manufacturer/<br>Device ID <sup>(6)</sup> | <b>90h</b>               |
| Read Unique ID <sup>(7)</sup>             | <b>4Bh</b>               |
| JEDEC ID                                  | <b>9Fh</b>               |



## SPI内部结构简明图



### SPI接口一般使用4条线通信：

MISO 主设备数据输入，从设备数据输出。

MOSI 主设备数据输出，从设备数据输入。

SCLK时钟信号，由主设备产生。

CS从设备片选信号，由主设备控制。

### 从器件选择 (NSS) 引脚管理

可以使用 SPI\_CR1 寄存器中的 SSM 位设置硬件或软件管理从器件选择。

- 软件管理 NSS (SSM = 1)

从器件选择信息在内部由 SPI\_CR1 寄存器中的 SSI 位的值驱动。外部 NSS 引脚空闲，可供其它应用使用。

- 硬件管理 NSS (SSM = 0)

根据 NSS 输出配置 (SPI\_CR1 寄存器中的 SSOE 位)，硬件管理 NSS 有两种模式。

- NSS 输出使能 (SSM = 0, SSOE = 1)

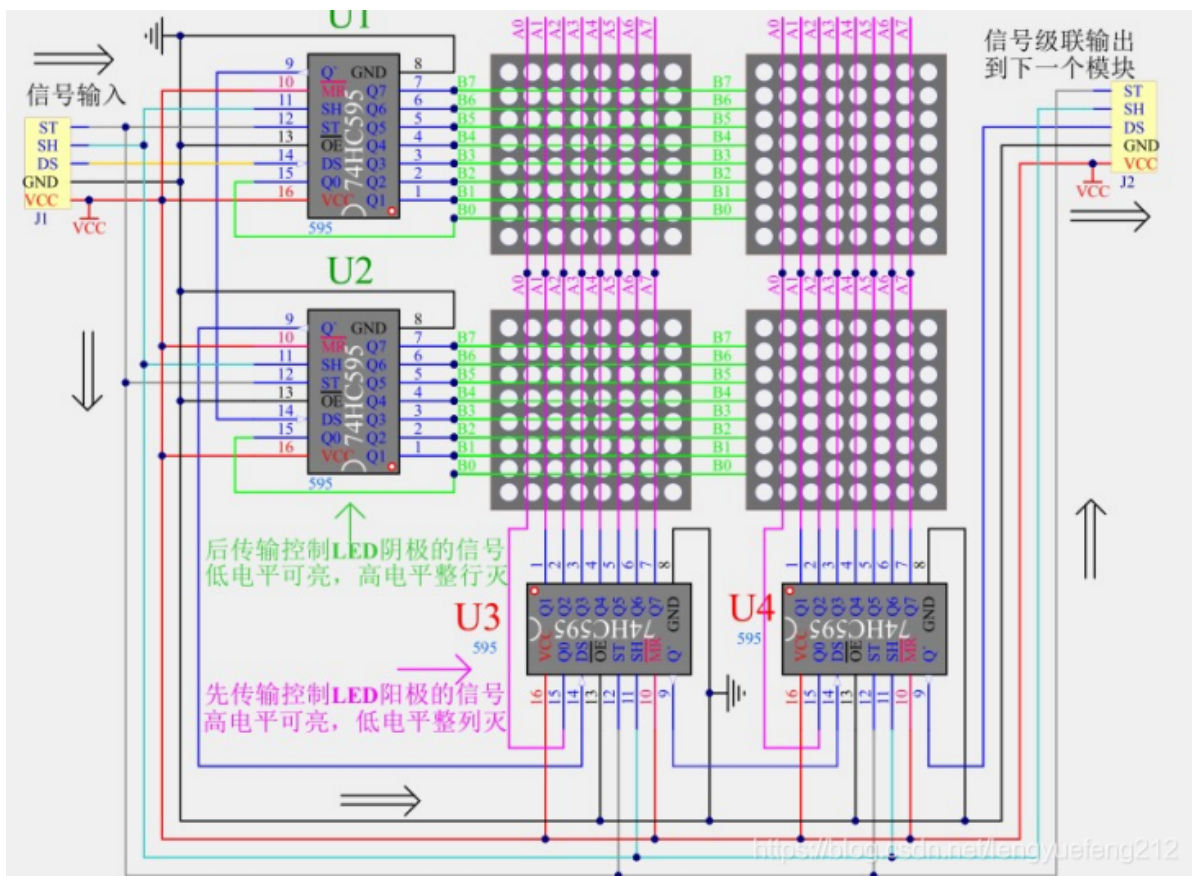
仅当器件在主模式下工作时才使用此配置。当主器件开始通信时，NSS 信号驱动为低电平，并保持到 SPI 被关闭为止。

- NSS 输出禁止 (SSM = 0, SSOE = 0)

对于在主模式下工作的器件，此配置允许多主模式功能。对于设置为从模式的器件，NSS 引脚用作传统 NSS 输入：在 NSS 为低电平时片选该从器件，在 NSS 为高电平时取消对它的片选。

## 点阵

16x16点阵用的是四块 CL-788BS (8\*8微型点阵) 和四块74HC595的8位串入并出的位移寄存器。



24X24LCD点阵：\

CL-788BS 共9片，\

74HC595 （行加一片，列加一片）共6片\

如图，这是16x16的接线图，将 CL-788BS 组成3x3矩阵，同时再加两片74HC595 对应相比16x16的多出的1行和1列。