

HT56R2x 使用 SPI 进行数据传输的用法与注意事项与 Software

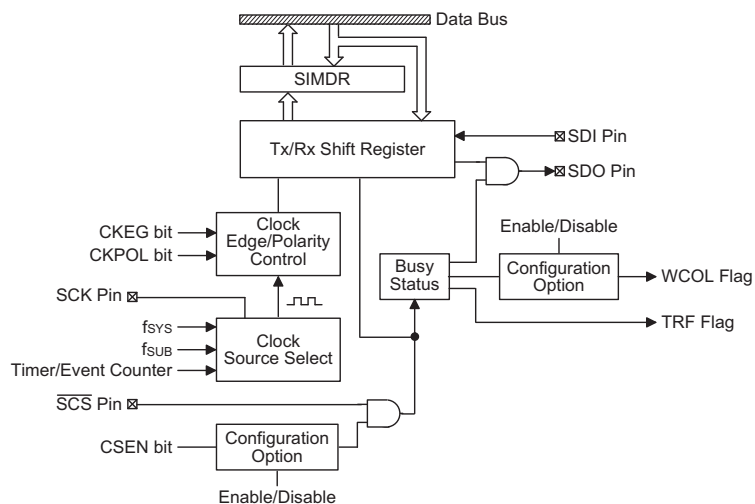
文件编码: HA0213S

简介

HT56R2x 内建有 Serial Interface Function, 其中包括了 SPI 和 I2C 这两种总线模式, 以及 SPI1 单独数据传输模式, SPI1 主要用于音频信号数据通信的传输, 由于 HT56R2x 系列的 SPI 与 SPI1 的通信方式和控制方式类似, 所以本文以 HT56R24 为母体, 主要介绍使用 SPI (SIM) 进行数据传输的方法和注意事项。

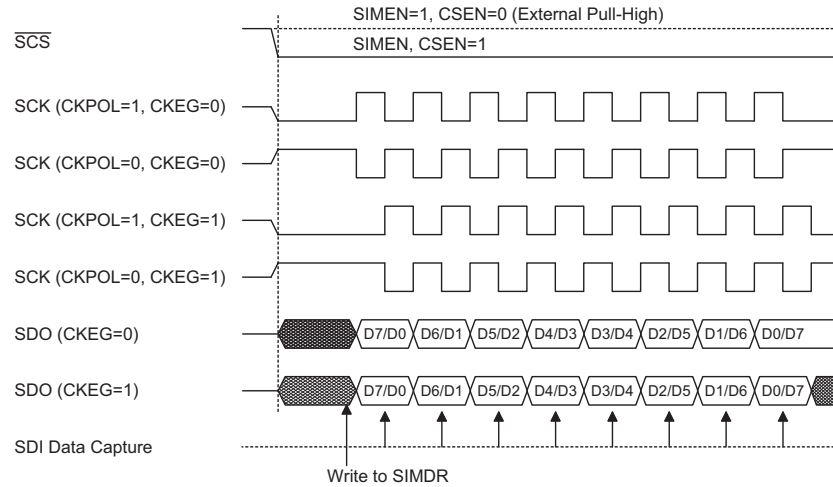
SPI Interface

SPI(Serial Peripheral Interface) 是一个全双工串行数据传输器, 最初由摩托罗拉设计, 其允许多种设备通过 SPI 总线进行相互通信。设备之间通过主/从技术, 只有主机能够发起数据的传递。一个简单的四线信号总线被用来进行所有的通信, 并且这些引脚与普通的 I/O 口共用引脚。

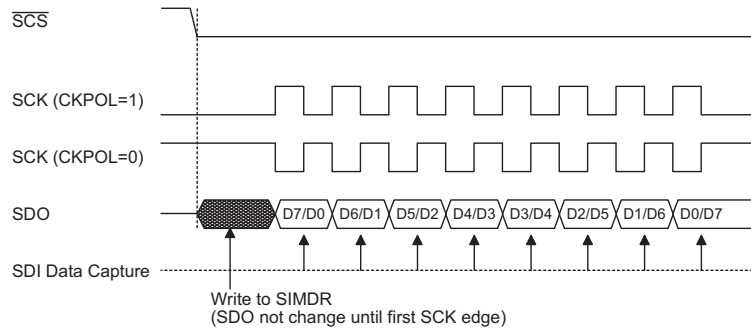


SPI 方框图

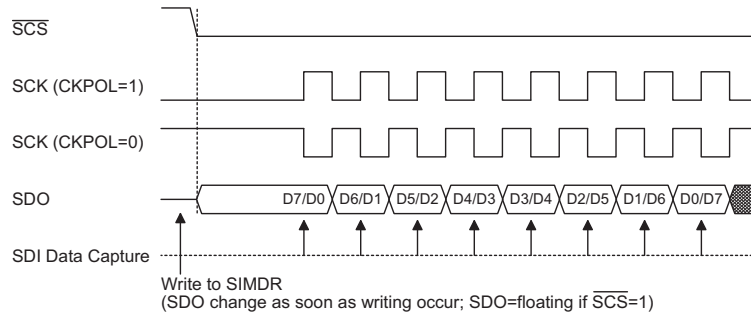
SPI 接口是一个全双工的串行数据连接，串行接口功能有 4 根基本信号线，包含 SDO (串行数据输出)、SDI (串行数据输入)、SCK (串行时钟) 和 \overline{SCS} (从组件选择)。注意的是，从机选择线的条件是由 SIMCTL2 控制寄存器内的 CSEN 位决定的。如果 CSEN 被置位， \overline{SCS} 线有效，但如果 CSEN 被清除为 0，那么 \overline{SCS} 线将处于浮空状态。下面的时序图描述了 SPI 总线的主模式和从模式下的时序协议。



SPI 主模式时序图



SPI 从模式时序图- CKEG=0

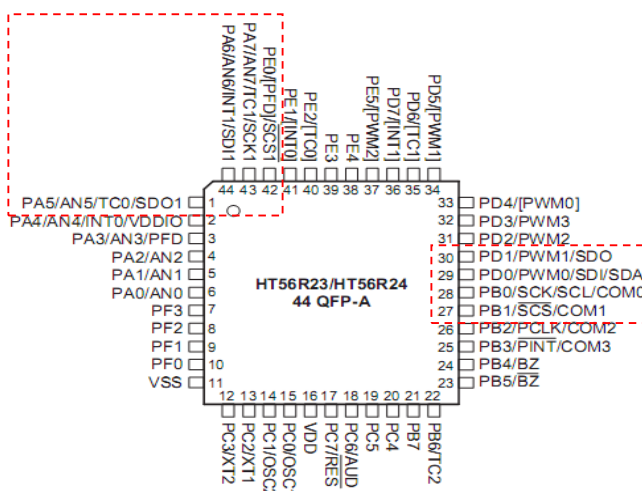


Note: For SPI slave mode, if SIMEN=1 and CSEN=0, SPI is always enabled and ignore the \overline{SCS} level.

SPI 从模式时序图 - CKEG=1

工作说明

如下图所示，SPI 的接口与 I/O 功能和 I2C 以及 COM 通道、PWM 等共用脚位，要打开 SPI 功能你必须首先将配置选项里的 SIM Function 选择为 Enable，WCOL 选择为 Enable，CSEN 选择为 Enable，然后设置好 SIMCTL0 和 SIMCTL2 寄存器值，如果在 IDLE Mode 中工作使能，则在 CLKMOD bit 4 SIMIDLE 设置为 1，则 SPI 通信工作模式在 IDLE Mode 使能，即为 Enable。而 CLKMOD bit 4 SIMIDLE 只有在 IDLE Mode 时提供给 SPI 接口 CLOCK 使用! (同时，SPI1 也支持在 IDLE Mode 下工作，也是由 CLKMOD bit 4 SIMIDLE 来决定，如果工作使能，则设置为 1，反之，设置为 0。)



HT56R24 中与 SPI 功能相关的一共有 3 个寄存器。它们是 SIMDR、SIMCTL0、SIMCTL2。SIMDR 寄存器被用来存储马上要传输或者刚接收到的数据。它是和 I2C 共用的，在 HT56R24 中，SPI 与 I2C 只能二选其一。要把数据写入 SPI 总线，数据必须要放入 SIMDR 寄存器才能被得到传输。相应的在数据从 SPI 总线接收到后，也只能从 SIMDR 寄存器中读出数据。

总之，任何通过 SPI 总线进行传送和接收的数据都必须通过相应接收或者发送数据寄存器。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SIMDR	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SIMDR 寄存器

数据传输工作模式

SIMCTL0 寄存器也是 SPI 和 I²C 共用的，它被用来打开或者关闭串行接口功能，设定 SPI 总线数据传输的时钟频率。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Label	SIM2	SIM1	SIM0	PCKEN	PCKPSC1	PCKPSC0	SIMEN	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

- SIMEN

这一个 Bit 在总体上控制 SPI 接口的开与关，当 SIMEN 位被清除为 0 时，将关闭掉 SPI 接口功能。SDI、SDO、SCK 与 SCS 将会处于浮空状态，此时，SPI 工作电流将变为一

个极小值。当这一位变为“1”时，SPI 功能将打开。当然，必须要先在配置选项中打开 SIM 功能，这一位才会有效。要注意到地方是，当 SIMEN 位从 0 变为 1 时，此时 SPI 的控制缓存器将会变为无法预测的值，因此，每当 SIMEN 位从 0 变为 1 时，应用程序必须要重新初始化 SPI 的控制缓存器。

- PCKEN、PCKPSC1、PCKPSC0

PCKEN、PCKPSC1、PCKPSC0 这三个缓存器用于控制外围时钟输出。此外围设备时钟输出引脚 PCLK 与 I/O 引脚 (PB.2) 共用，这个外围设备的时钟来源于定时/计数器 2 的 2 除频或内部系统时钟的除频。

PCKEN：这一个位在总体上控制外围时钟输出的开与关。

PCKPSC0, PCKPSC1：选择外围时钟输出的时钟来源，如下表：

PCKPSC [1:0]: Peripheral Clock Output Clock Source	
00:	f _{sys}
01:	f _{sys} /4
10:	f _{sys} /8
11:	Timer 2 output/2

- SIM0~SIM2

这几位用来设定 SIM 功能的操作模式，也就是选择 I²C 功能或者选择 SPI 功能。如果选择了 SPI 功能，还将选择主/从模式、主机的时钟频率，时钟可以选择为系统时钟，也可以来源于定时器。如果选择从模式，将不用选择时钟，它的时钟由它的主机提供。具体情况由下图所示：

```

SIM2, SIM1, SIM0: SIM Operating Mode Control
000: SPI master mode; SPI clock is fsys/4
001: SPI master mode; SPI clock is fsys/16
010: SPI master mode; SPI clock is fsys/64
011: SPI master mode; SPI clock is fTBC
100: SPI master mode; SPI clock is TM0 CCRP match frequency/2
101: SPI slave mode
110: I2C slave mode
111: Unused mode
    
```

SIMCTL2 缓存器是一个 SPI 专用的控制缓存器。

SIMCTL2 寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Label	—	—	CKPOL	CKEG	MLS	CSEN	WCOL	TRF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- TRF

传送/接收完成标志位，当传送或者接收完成后，它将自动设定为 1。需要软件清除。

- WCOL

主/从模式下，若正在发送数据或接收数据，写 SIMDR 缓存器将会设定位 WCOL，且写入数据被忽略。WCOL 功能可由配置选项打开或者关闭。WCOL 由硬件设定位，软件清除为 0。

- CSEN

串行接口片选功能打开/关闭。若 CSEN=1， \overline{SCS} 片选功能有效。主模式下，在 SCK 信号输出前输出 \overline{SCS} 片选信号；而在从模式下，接收到 SCS 信号前(后)，数据传输被关闭 (打开)。若 CSEN=0， \overline{SCS} 引脚处于浮空状态，片选功能失效，此时可在外部对 \overline{SCS} 引脚 Pull-High 以实现片选功能。CSEN 功能可由配置选项打开或者关闭。

- MLS

MSB 或 LSB 选择位。也就是传输时高位优先或者低位优先。

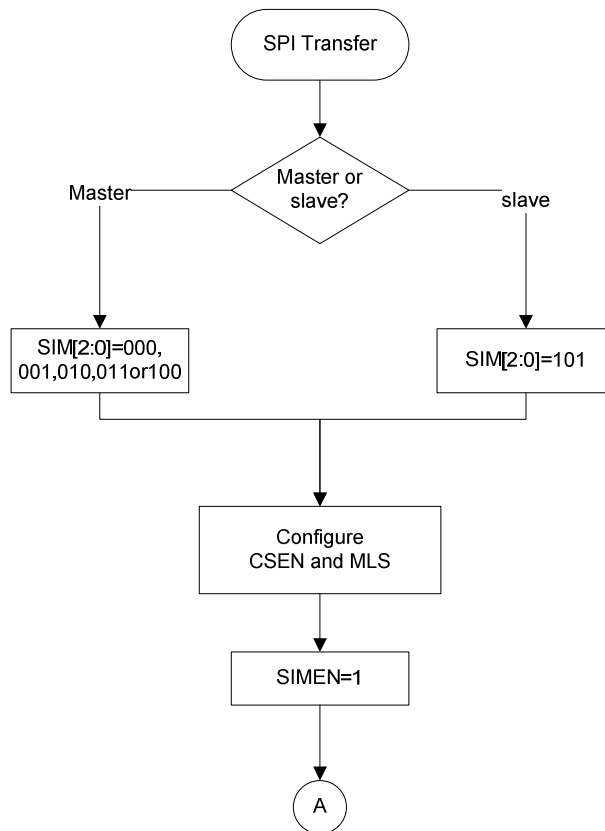
- CKEG 和 CKPOL

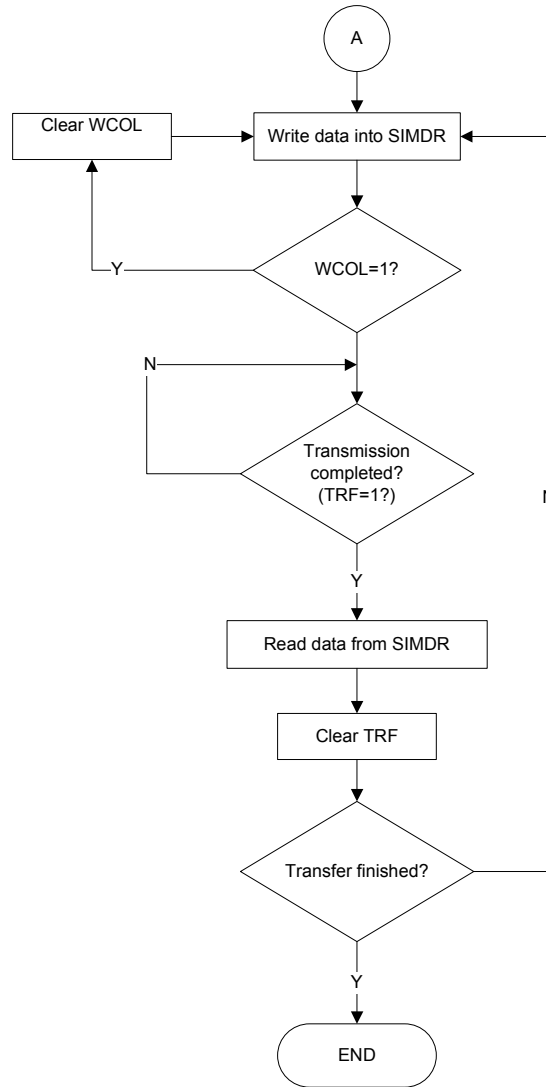
这两位在进行数据传输前必须进行设定，否则，一个错误的时钟沿会被产生。CKPOL 位决定时钟线的基本状态，当这一位为高位时，时钟线在不活动的状态下将为低位，反之为高位。CKEG 位决定时钟线有效时的时钟沿类型，它将依赖于 CKPOL 的值，下面是这 2 字节合起来的情况。

CKPOL	CKEG	SCKClock Signal
0	0	High Base Level Active Rising Edge
0	1	High Base Level Active Falling Edge
1	0	Low Base Level Active Falling Edge
1	1	Low Base Level Active Rising Edge

S/W 流程图

以 SPI 数据传输为例，以下为 SPI(SIM)程序的流程图 (SPI1 的数据传输方式流程图与之类似)，如下：





SPI 传输控制流程图

SPI 操作

在与 I²C 共用 SIMCTL0 缓存器里，在主/从机模式下，所有的通信都通过使用 SPI 总线得以实现。时序图显示了基本的总线操作。SPI 传输过程中，主机在输出 SCK 信号前，通过传输一个 $\overline{\text{SCS}}$ 信号选择从机，当 CSEN=0， $\overline{\text{SCS}}$ 引脚需外部 Pull-High，如 SPI Master Mode Timing 图所示。从模式在 $\overline{\text{SCS}}=1$ 时，SDO 为浮空， $\overline{\text{SCS}}=0$ 时，SDO 有效。从模式下，若 CSEN=0，不管 $\overline{\text{SCS}}$ 状态如何（为高电平或为低电平），只要 SIMEN=1，SPI 就处于有效状态。SIMCTL0 缓存器的 SIMEN 位必须置位，其可以将 SDI 引脚设成浮空状态并且将 SDO 引脚置高。在从模式下，SCK 引脚将处于浮空状态。如果 SIMEN 位清除为 0，那么总线将被禁止，并且 $\overline{\text{SCS}}$ 、SDI、SDO 和 SCK 将全都处于浮空状态。在主模式下，主机将始终产生时钟信号。在数据被写入 SIMDR 缓存器后，将启动时钟和数据的传递。在从模式下，数据的传递和接收将通过接收来自外部主机设备的时钟信号来启动。下面的步骤显示了在主/从模式下数据传递遵循的顺序。

主模式

- 步骤 1
程序初始化，设定 SIMCTL0 控制缓存器的 SIM0~SIM2 位来选择主模式和波特率。
- 步骤 2
设定 CSEN 并利用 MLS 来选择数据从高位还是低位开始，从机必须与主机保持一致。
- 步骤 3
设定 SIMCTL0 控制缓存器的 SIMEN 位来打开 SPI。
- 步骤 4
将欲传出资料写入 SIMDR，检查 WCOL: WCOL=1 → 发生冲突错误，并跳至步骤 4。
WCOL=0 → 跳至步骤 5。
- 步骤 5
检查 TRF 或等待 SPI 串行总线中断。
- 步骤 6
从 SIMDR 缓存器读取数据，放入 temp3 数据缓存器。
- 步骤 7
清除 TRF，返回步骤 4。

从模式

- 步骤 1
设定的 SIM0~SIM2 位为“101”来选择从模式。
- 步骤 2
设定 CSEN 并利用 MLS 来选择数据从高位还是低位开始，主机必须保持一致。
- 步骤 3
设定 SIMCTL0 控制缓存器的 SIMEN 位来打开 SPI 接口。
- 步骤 4
将资料写入 SIMDR，检查 WCOL: WCOL=1 → 发生冲突错误，并跳至步骤 4。
WCOL=0 → 跳至步骤 5。
- 步骤 5
检查 TRF 或等待 SPI 串行总线中断。
- 步骤 6
从 SIMDR 缓存器读取数据，放入 temp3 数据缓存器。
- 步骤 7
清除 TRF，返回步骤 4。

SPI 配置选项

一些配置选项必须通过设备的程序来设定后用作 SPI 接口功能。一个配置选项用来在 SIMCTL2 缓存器使能 WCOL 的操作和写入冲突位。另一个配置选项用来禁止或使能 SIMCTL0 缓存器中的 CSEN 位。如果配置选项禁止 CSEN 位，那么 CSEN 将不能用来影响 SPI 总线所有的控制。SIMCTL2 缓存器的 WCOL 位用来在数据传送中提示写入冲突错误。当数据传递操作中出现写 SIMDR 缓存器的现象时，WCOL 位会提示数据冲突并且防止继续进行写操作。WCOL 位将通过硬件被置位，但必须通过用户应用程序来清除为 0。WCOL 位的全部功能可以通过配置选项来禁止或打开。

对应的程序注意事项

在设备进入 IDLE 模式下，注意的是，数据的接收和传递将依赖于 FSYSON 位，它位于 WDTC 缓存器的第 7 位，当 FSYSON 位为“1”时，方可以在 IDLE 模式下继续进行数据的接收和传递。在数据传递和接收后，TRF 位用来产生中断 (SPI1 的程序注意事项与之类似)。

程序说明

本范例由主程序以及 SPI 的服务程序二部分组成，在主程序中，会对 MCU 进行初始化操作，主要是对 SPI 的相关缓存器进行初始化操作，关闭一些和 SPI 功能共用 I/O 的功能。

SPI 的服务程序 spi_Transmission 子程序中，会将数据缓存器 temp1 中的数据发送到 SPI 总线，会将从 SPI 总在线接收到的资料保存到 temp3 中，用户可以自行修改。如果用户只需要发送或者接收，就只需要注意 temp1 或者 temp3 的值。

程序中没有打开中断，如果要使用中断，请在检查 WOCL 为“0”后，不必再继续检查 TRF 位，打开相应中断使能位 ESIM，总中断使能位 EMI，等待中断产生。

配置选项中，要使用 SPI 总线，必须打开 SIM Function，WOCL 和 CSEN。其它选项由用户使用情况决定。附件 HT56R24(SPI SIM).zip 为 HT56R24 在 SPI 工作模式下与 HT66F40 的电路原理图。

程序范例

在 SPI(SIM)工作模式下的主/从机发送/接收程序:

```

;function:edit for SPI master
;MCU HT56R24
;option:
;OSC HIRC 4MHz
;SIM enable
;WCOL enable
;CSEN enable
;IO or resb IO
;WDT disable
;SPI1 disable
;SPI1 WCOL disable
;SPI1 CSEN disable
;pa5/6/7 VDD
;VDDIO disable
;pb4/5 IO
;pc6 IO
;others select by user
include HT56R24.inc
ds .section 'data'
temp1 db ?
temp2 db ?
temp3 db ?
temp4 db ?
temp5 db ?
temp6 db ?
temp7 db ?
temp8 db ?
cs .section 'code'
org000h
jmp main
org014h
reti
mmov macro h2,h1
mov a,h1
mov h2,a
endm
main:
call BK0
call BK2
call BK3
call BK4
call BK5 ;clr data of the bank0~5,except bank1
set pec2 ;set pe2 as input
set pepu2
set pec3 ;set pe3 as input
set pepu3
set pec4 ;set pe4 as input
set pepu4
clr emi
clr tmr1c
clr intc0
clr intc1
clr simdr
clr temp1
clr temp2
clr temp3

```

```

        clr     temp4
        clr     temp5
L0:sz    pe2           ;choose the transmit mode of SPI
        jmp    L1           ;pe2=1,slave
        jmp    L2           ;pe2=0,master,fsys/4
L1:clr   csen
        mov    a,0a0h
        jmp    L3
L2:set   csen
        mov    a,000h
L3:mov   simctl0,a
        clr    ckpol       ;high Base Level active rising edge
        clr    ckeg
        set    mis         ;data shift oder from msb to lsb
        set    simen       ;enable sim
        jmp    spi_Transmission
L:  sz    pe3
        jmp    Linput      ;pe3=1, pa port as input mode
        jmp    Loutput     ;pe3=0, pa port as output mode
Linput: ;set the pa port input mode one time in order not to inflict the transmission
        mov    a,temp4
        addm a,pcl
        jmp    Linput1
        jmp    Linput2
Linput1:
        set    pac         ;set pa port as input
        set    papu
Linput2:
        mmov   temp1,pa    ;mov pa to temp1 register
        mmov   temp4,1
        clr    temp5
        jmp    spi_Transmission
Loutput:
        mov    a,temp5
        addm a,pcl
        jmp    Loutput1
        jmp    Loutput2
Loutput1:
        clr    pac         ;set pa port as output
        set    pa
Loutput2:
        mmov   pa,temp3   ;mov the simdr data into temp3 and display by pa port
        mmov   temp5,1
        clr    temp4
        jmp    spi_Transmission
spi_Transmission: ;the transmission of SPI programme
        set    simen
        clr    trf
        clr    wcol
        mmov   simdr,temp1 ;mov the register data of temp1 into simdr,ready to transmit
        sz    wcol
        jmp    $-4
        snz   trf
        jmp    $-1
        mmov   temp3,simdr
        snz   pe4
        jmp    L0         ;pe4=0,return to judge the state of pe2,and judge the transmit slave or master mode
        jmp    L         ;pe4=1,return to judge the state of pe3,and judge the pdl input or output mode

```

备注:

- pe2, pe3:

- 00: 作为主机，输出，可以通过主机看到从机发送而被主机接收到的数据。
- 10: 作为主机，输入，可以通过从机看到主机发送而被从机接收到的数据。
- 01: 作为从机，输出，可以通过从机看到主机发送而被从机接收到的数据。
- 11: 作为从机，输入，可以通过主机看到从机发送而被主机接收到的数据。
- pe4:
 - 0: 重新检测 pe2 口高/低的状态，重新设定在 SPI 通信时的主/从模式。
 - 1: 进入检测 pe3 口高/低的状态，设定所 SIMDR 缓存器数据的存入/取出方式。

与 SPI 工作模式下相对应的从/主机接收/发送程序:

```

;function: edit for SPI master
;MCU: HT66F40
;option:
;OSC  $\Delta$  HIRC + 4M + LIRC 32K
;SIM enable
;WCOL enable
;CSEN enable
;/O or RESB  $\rightarrow$  RESB
;WDT Enable two instructions
;Others select by user
include HT66F40.INC
.listinclude
use_data.section 'data'
    data1 db      ?
    data2 db      ?
    data3 db      ?
    temp1 db      ?
    temp2 db      ?
maincode.section at 00 'code'
    org      00h
    jmp     main_start
mmov     macro h2,h1
mov      a,h1
mov      h2,a
endm
main_start:
    call    initial           ;initial the program
    set     pac
; the transmission of SPI programme
spi_server:
    set     csen              ;SPI SCS pin Control--enable
    set     mls               ;SPI Data shift order--MSB
    sz     pa.0
    jmp     L1
    jmp     L2
L1:mov    a,0a0h              ;pa.0=1 spi slave mode
    jmp     L3
L2:mov    a,020h              ;pa.0=0 spi master mode
L3:mov    simc0,a
    set     simen
    jmp     spi_Transmission
L: snz     pa.1
    jmp     pc_as_output      ;pa1=0,pc port as output
    jmp     pc_as_input;pa1=1,pc port as input
pc_as_output:
    mov     a,temp1
    addm a,pc1
    jmp     Loutput1
    jmp     Loutput2
Loutput1:

```

```

clr      pcc
clr      pc
Loutput2:
mmov    pc,data2
mmov    temp1,1
clr     temp2
jmp     spi_Transmission
pc_as_input:
mov     a,temp2
addm a,pc
jmp     Linput1
jmp     Linput2
Linput1:
set     pcc
Linput2:
mmov    data1,pc
mmov    temp2,1
clr     temp1
jmp     spi_Transmission
spi_Transmission:
clr     trf
clr     wcol
mov     a,data1
mov     simd,a           ;move data1 to simd register
sz     wcol             ;judge the mode of write whether or not conflict
jmp     $-4
snz    trf              ;judge the transmission finish
jmp     $-1
mmov    data2,simd     ;read data from simd
snz    pa2
jmp     spi_server ;pa2=0, return to judge the state of pa1 port
jmp     L              ;pa2=1, return to judge the state of pa0 port

```

备注:

- pa0、pa1:
 - 00: 作为主机，输出，可以通过主机看到从机发送而被主机接收到的数据。
 - 10: 作为主机，输入，可以通过从机看到主机发送而被从机接收到的数据。
 - 01: 作为从机，输出，可以通过从机看到主机发送而被从机接收到的数据。
 - 11: 作为从机，输入，可以通过主机看到从机发送而被主机接收到的数据。
- pa2:
 - 0: 重新检测 pa0 口高/低的状态，并重新设定在 SPI 通信时的主/从模式。
 - 1: 检测 pa1 口高/低的状态，设定所 SIMDR 缓存器数据的存入/输出方式。

总结

本文主要讲解了 HT56R24 的 SPI 操作与注意事项，由于 SPI1 的操作与注意事项类似与 SPI，所以用户若需要使用 SPI1，则可以对相应的缓存器进行设定，如下：

SPI(SIM)	SIMCTL0	SIMCTL2	SIMDR
SPI1(Voice)	SPICTL0	SPICTL1	SPIDR

在缓存器 SPICTL0 中只有设定主机、频率或从机，无设定 I²C 项。同时，在配置选项中应选择 SPI1 为 Enable、WCOL1 为 Enable、CSEN1 为 Enable。如果要使用 SPI1 中断，请在 WCOL1 为“0”后，不必再继续检查 TRF1 位，打开相应中断使能位 ESPI，总中断使能位 EMI，等待中断的产生。需要注意的是，SPI (SIM) 的中断发生地址为 014H，而 SPI1 (Voice) 的中断发生地址为 018H。其它选项均可参照上面表格对应的缓存器选项进行设定。用户可以在自己的程序中直接插入此 IP，可根据自己的需要按照上面程序说明自行修改。附件中附有 HT56R24 与 HT66F40 的 SPI(SIM)和 SPI1(Voice)的通信方式以及电路原理图。