



HT48 & HT46 MCU 用软件执行 I²C 总线的控制功能的方法

文件编码：HA0005s

简介：

I²C 总线包括了两条串行总线（时钟线 SCL 和数据线 SDA），通过这两条总线能实现多个芯片之间的通信。在互相连接的芯片中，至少有一个芯片作为总线控制器，而其它芯片则作为从控制器。在本应用说明中，介绍了用 Holtek 的八位 RISC 结构的单片机作为单总线控制器的软件实现的方法。在本文的示例中，采用了一片 EEPROM（型号 HT24LC02，2Kbit）作为从控制器参与测试。

电路说明：

HT24LC02 的 A0、A1、A2、VSS、WP 引脚接地，VCC 接+5V，SCL 接 PA3，SDA 接 PA2

使用说明：

例程中先向 eeprom 中写数据，写完后，再将 eeprom 中内容读出来，并将读出数据进行比较，若数据不相等程序跳到 fail_out 中；若相等，最后程序跳到 ok_end 中。

本说明中提供了一个源文件 OP_HT24.ASM 和一个包含文件 HT24.ASM。在应用时，要将 OP_HT24.ASM 文件添加到用户的 project 中，并修改 HT24.INC 文件中的变量设置，以建立 SCL/SDA 引脚来与用户的应用电路相匹配。

实例：

```
=====
; 文件名：op_ht24.asm
; 日期：2003/11/18
; MCU：HT48R30A-1
; EEPROM：HT24LC02
; Fsys：1MHz
=====
include ht24.asm

data .section 'data'
count db ?
fail db ?
success db ?
=====
main .section at 0 'code'
    clr count
    clr fail
    clr success

    clr pac
    clr pa

write_cycle:
```

HOLTEK HT48 & HT46 MCU 用软件执行 I²C 总线的控制功能的方法

```
mov    a, 0a0h                ;1010 0000
mov    deviceaddr, a
mov    a, count
mov    dataaddr, a
mov    data2, a

call   ht24_byte_write
call   write_delay            ;延时for write
siz    count                  ;256 x 8 = 2K
jmp    write_cycle

read_cycle:
mov    a, 0a0h                ;1010 0000
mov    deviceaddr, a
mov    a, count
mov    dataaddr, a

call   ht24_byte_read
mov    a, data2
xor    a, count
sz     acc
jmp    fail_out
siz    count
jmp    read_cycle

ok_end:
set    success
jmp    $

fail_out:
set    fail
jmp    $
```

在这个程序中，执行了两个文件HT24.INC(附件1)和HT24.ASM(附件2)。

附件 1: HT24.INC

```
;=====
; 文件名: HT24.INC
; 日期: 2003/11/18
;=====

;=====
; 可根据你的电路,修改引脚sda/sdac/scl/sclc的定义
;=====
```



HT48 & HT46 MCU 用软件执行 I²C 总线的控制功能的方法

```
#define port [12H]
#define portc [13H]
sda equ port.2 ;数据线
sdac equ portc.2 ;数据控制
scl equ port.3 ;时钟线
sclc equ portc.3 ;时钟控制

;=====
;以下部分不可修改
;=====
ifndef ht24_asm
extern deviceaddr :byte ;器件地址及操作码
extern dataaddr :byte ;页内地址
extern data2 :byte ;写入及读出的byte数据
extern array :byte ;写入及读出的page数据
extern num :byte ;page的大小
extern ht24_byte_write :near ;byte写
extern ht24_byte_read :near ;byte读
extern ht24_page_write :near ;page写
extern ht24_page_read :near ;page读
extern write_delay :near ;约5MS的延时
endif
;=====
;文件HT24.INC结束
;=====
附件 2: HT24.ASM
;=====
; 文件名 : ht24.asm
; 日期 : 2003/11/18
; ROM使用情况 : 94H
; RAM使用情况 : 16H
;=====
#define ht24_asm
include ht24.inc
include ht48r30a-1.inc

;宣告提供给外部程序使用的变量名称
public deviceaddr
public dataaddr
public data2
public array
public num
;宣告提供给外部程序使用的子程序名称
```



HT48 & HT46 MCU 用软件执行 I²C 总线的控制功能的方法

```
public ht24_byte_write
public ht24_byte_read
public ht24_page_write
public ht24_page_read
public write_delay
;=====
;数据段
ht24data .section 'data'
deviceaddr db ?
dataaddr db ?
data2 db ?
array db 16 dup (?)
num db ?
data1 db ? ;临时寄存器
movb db ? ;8bit计数
;=====
;程序段
ht48iiccode .section 'code'
;=====
; 描述：读取EEPROM中一个BYTE的数据
; 入口参数： deviceaddr:byte
; dataaddr :byte
; 出口参数： data2 :byte
; STACK使用： 1
;=====
ht24_byte_read proc
    call start ;送开始信号
    mov a, deviceaddr ;送device address & write op-code
    mov data1, a
    call byte_write
    mov a, dataaddr ;送data address
    mov data1, a
    call byte_write
    call start
    mov a, deviceaddr
    inc acc ;送device address & read op-code
    mov data1, a
    call byte_write
    mov a, offset data2
    mov mp0, a
    call byte_read ;读取指定地址处的data
    call stop ;送停止信号
    ret
```



HT48 & HT46 MCU 用软件执行 I²C 总线的控制功能的方法

```
ht24_byte_read   endp
;=====
; 描述：向EEPROM中写入一个BYTE数据
; 入口参数：  deviceaddr:byte
;             dataaddr  :byte
;             data2     :byte
; 出口参数：  无
; STACK使用：1
;=====
ht24_byte_write  proc
    call  start           ;送开始信号
    mov   a, deviceaddr   ;送device address & write op-code
    mov   data1, a
    call  byte_write
    mov   a,dataaddr      ;送data address
    mov   data1, a
    call  byte_write
    mov   a,data2         ;送数据
    mov   data1, a
    call  byte_write
    call  stop            ;送停止信号
    ret
ht24_byte_write  endp
;=====
; 描述：读取EEPROM中的一个PAGE的数据
; 入口参数：deviceaddr:byte
;             dataaddr  :byte
; 出口参数：array      :16 bytes
; STACK使用： 1
;=====
ht24_page_read   proc
    call  start           ;调用开始信号
    mov   a, deviceaddr
    mov   data1, a        ;页地址判断，送页地址及操作码
    call  byte_write
    mov   a, dataaddr
    mov   data1, a        ;送页内地址
    call  byte_write
    call  start
    mov   a, deviceaddr
    inc   acc
    mov   data1, a        ;页地址判断，送页地址及操作码
    call  byte_write
```



HT48 & HT46 MCU 用软件执行 I²C 总线的控制功能的方法

```
    mov    a, offset array ;将从EEPROM中读出来的数据存放到ARRAY数组中。
    mov    mp0, a
more:
    call   byte_read      ;调用读BYTE子程序
    sdz    num
    jmp    cmack
    jmp    retu
cmack:
    call   mack
    inc    mp0
    jmp    more
retu:
    call   stop           ;送停止信号
    ret
ht24_page_read  endp
;=====
; 描述：向EEPROM中写入一个PAGE的数据
; 入口参数：deviceaddr:byte
;           dataaddr  :byte
;           array     :16 bytes
; 出口参数：  无
; STACK使用：1
;=====
ht24_page_write proc
    call   start          ;送开始信号
    mov    a, deviceaddr
    mov    data1, a       ;送页地址和写操作命令
    call   byte_write
    mov    a, dataaddr
    mov    data1, a       ;送页内地址
    call   byte_write
    mov    a, offset array ;将存放在ARRAY数组中数据写入EEPROM。
    mov    mp0, a
amore:
    mov    a, r0          ;送数据至EEPROM
    mov    data1, a
    call   byte_write
    nop
    sdz    num
    jmp    amore
    call   stop          ;送停止信号
    ret
ht24_page_write  endp
```



HT48 & HT46 MCU 用软件执行 I²C 总线的控制功能的方法

```
=====
; 描述：写入数据后的延时,延时时间约为5MS(Fsys=4MHz)
; 入口参数： 无
; 出口参数： 无
; STACK使用： 无
=====
write_delay      proc
    set    data1
    mov    a,06h
    mov    movb,a
lpy1:
    sdz    data1
    jmp    lpy1
    sdz    movb
    jmp    lpy1
    ret
write_delay      endp
=====
;送信号0到EEPROM
=====
mack      proc
    clr    sdac
    clr    sda
    nop
    set    scl
    nop
    nop
    clr    scl
    nop
    set    sda
    ret
mack      endp
=====
;开始信号部分的子程序
=====
start      proc
    clr    sclc
    clr    sdac
    set    scl
    nop
    set    sda
    nop
    clr    sda
```



HT48 & HT46 MCU 用软件执行 I²C 总线的控制功能的方法

```
    nop
    clr    scl
    ret
start  endp
;=====
;结束信号部分的子程序
;=====
stop  proc
    clr    sdac
    clr    sda
    nop
    set    scl
    nop
    set    sda
    nop
    clr    sda
    clr    scl
    ret
stop  endp
;=====
;单字节写部分的子程序
;=====
byte_writeproc
    mov    a, 08h    ;串行输入八个bit的数据
    mov    movb, a
loop:
    clr    sdac
    clr    scl      ;写数据时, scl为high
    rl    data1
    mov    a, data1
    snz   acc.0    ;数据串行输入
    jmp   loop_1
    set   sda
    jmp  loop_2
loop_1:
    clr    sda
loop_2:
    nop
    set    scl
    nop
    nop
    sdz   movb
    jmp   loop
```



HT48 & HT46 MCU 用软件执行 I²C 总线的控制功能的方法

```
    clr    scl
    set    sdac
    nop
    set    scl    ;读数据时，scl为low
rep:
    sz     sda    ;检查EEPROM有没有送应答信号
    jmp    rep
    clr    scl
    ret
byte_writeendp
;=====
;单字节读部分的子程序
;=====
byte_read    proc
    mov    a, 08h    ;串行输出八个bit的数据
    mov    movb, a
    set    sdac
    nop
loops:
    set    scl    ;读数据时，scl为low
    rl     r0
    snz    sda    ;数据串行输出
    jmp    loops_1
    set    r0.0
    jmp    loops_2
loops_1:
    clr    r0.0
loops_2:
    clr    scl
    nop
    sdz    movb
    jmp    loops
    ret
byte_read    endp
;=====
;程序结束
;=====
```