

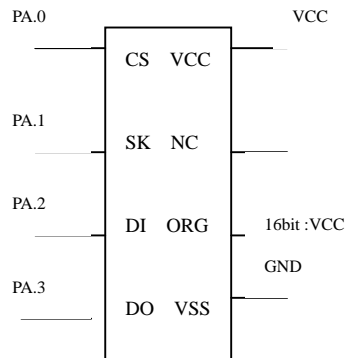
HT48 & HT46 MCU 与 HT93LC46 EEPROM 的通信

文件编码：HA0003s

介绍：

HT93LC46EEPROM 是 Holtek 制造的 1K 位系列的 EEPROM (电可擦除只读存储器), 一般它用于微控制器的固定数据的存储。在本文中, 我们将以 Holtek 公司 8 位微控制器为例, 介绍该芯片常用的操作功能代码。用户只需把代码加到程序中, 并且在使用 HT93LC46 之前将引脚 CS/SK/DI/DO 连接即可。

电路图：



功能实现：

本文是以 HT48R30A-1 控制 HT93LC46 为例的, ORG 接 VCC 即容量为 64X16bits。在本文中用到一汇编源文件 OP16_93LC46.ASM。在 HT - IDE 开发环境下, 在使用所提供的接口函数前按下述的操作步骤做：

- 步骤 1：把 OP16_93LC46.ASM 加到项目下 (用[Project/Edit]指令)
- 步骤 2：根据你的电路, 修改 HT93LC46.ASM 文件连接 CS/SK/DI/DO 引脚
- 步骤 3：包括 HT93LC46.ASM 的源文件调用接口函数
- 注意：在使用这些函数前要适当地设置输入/输出口的模式

生成一个目标控制器为 HT48R30A-1 的项目 (Project), 然后加入 OP16_93LC46.ASM 文件到这个项目中。

在这个例子中, 首先使用 HT93LC46_EWEN 函数使 HT93LC46 写置能; 然后, 使用 HT93LC46_WRITE 函数向存储器写入相应的地址; 最后, 使用 HT93LC46_READ 指令将刚才写入的数据读出并检查写入的数据是否正确。

程序如下：

```

;=====
; 文件名：op16_93lc46.asm
; 日期：2003/11/21
; MCU：HT48R30A-1
; EEPROM：HT93LC46, 64x16bits
;=====

```



HT48 & HT46 MCU 与 HT93LC46 EEPROM 的通信

```
include ht48r30a-1.inc
include ht93lc46.asm ;见附件 1

data .section 'data'
temp db ?

main .section at 0 'code'
start:
    clr csc
    clr skc
    clr dic
    set doc

;-----
    call ht93lc46_ewen ;ht93lc46 写置能
    nop
    mov a,3fh ;ORG 脚接 VCC , 64x16
    mov temp,a
write:
    mov a, temp
    mov dataaddr, a
    mov data2, a
    cpl acc
    mov data3, a
    call ht93lc46_write ;ht93lc46 写
    sdz temp
    jmp write
    call ht93lc46_ewds ;ht93lc46 写除能

;-----
; 开始读数据并比较
    mov a,3fh ;ORG 脚接 VCC , 64x16
    mov temp,a
read:
    clr data2
    clr data3
    mov a, temp
    mov dataaddr, a
    call ht93lc46_read ;ht93lc46 读

    mov a, temp ;比较数据
    xor a, data2
    snz z
    jmp fail
    cpla temp
```

```

xor    a, data3
snz    z
jmp    fail
sdz    temp
jmp    read

;-----
call   ht93lc46_ewen      ;ht93lc46 写置能
nop
call   ht93lc46_eral     ;ht93lc46 擦除全部空间
nop
mov    a, 3fh            ;ORG 脚接 VCC , 64x16
mov    temp, a

read_1:
clr    data3
clr    data2
mov    a, temp
mov    dataaddr, a
call   ht93lc46_read     ;ht93lc46 读

mov    a, 0ffh          ;比较数据
xor    a, data2
snz    z
jmp    fail
mov    a, 0ffh
xor    a, data3
snz    z
jmp    fail
sdz    temp
jmp    read_1

;-----
mov    a, 04bh
mov    data2, a
mov    a, 0c3h
mov    data3, a
call   ht93lc46_wral     ;ht93lc46 写全部空间
nop
mov    a, 3fh            ;ORG 脚接 VCC , 64x16
mov    temp, a

read_2:
clr    data2
clr    data3
mov    a, temp
mov    dataaddr, a

```

```

call    ht93lc46_read          ;ht93lc46 读

mov     a, 04bh                ;比较数据
xor     a, data2
snz     z
jmp     fail
mov     a, 0c3h
xor     a, data3
snz     z
jmp     fail
sdz     temp
jmp     read_2

;-----
mov     a, 3fh                 ;ORG 脚接 VCC , 64x16
mov     temp, a
write_1:
mov     a, temp
mov     dataaddr, a
call    ht93lc46_erase        ;ht93lc46 擦除指定地址
sdz     temp
jmp     write_1
nop
mov     a, 3fh                 ;ORG 脚接 VCC , 64x16
mov     temp, a
read_3:
clr     data3
clr     data2
mov     a, temp
mov     dataaddr, a
call    ht93lc46_read        ;ht93lc46 读

mov     a, 0ffh                ;比较数据
xor     a, data2
snz     z
jmp     fail
mov     a, 0ffh
xor     a, data3
snz     z
jmp     fail
sdz     temp
jmp     read_3

jmp     $                      ;操作成功

```



```
fail:
    jmp    $                ;操作失败
;-----
;程序 op16_93lc46.asm 结束
;-----
```

以下为附件 1 (HT93LC46.ASM), 源文件调用接口函数, 包含了 HT93LC46 的七条指令。

```
=====
; 文件名:ht93lc46.ASM
; 日期:2003/11/21
; ROM 使用情况:9DH
; RAM 使用情况:07H
=====
#define    ht93lc46_asm

;操作码
oc_read    equ    10000000b
oc_erase   equ    11000000b
oc_write   equ    01000000b
oc_ewen    equ    00110000b
oc_ewds    equ    00000000b
oc_eral    equ    00100000b
oc_wral    equ    00010000b

;=====
;如果 ORG 脚接 VCC, 则屏蔽 bit8, 64x16
;如果 ORG 脚接 VSS, 则屏蔽 bit16, 128x8
;=====
#define    bit16
;#define    bit8
;=====
;可根据你的电路修改各引脚 CS/SK/DI/DO 的定义
;=====

port      equ [12H]
portc     equ [13H]
#define    sk port.1
#define    skc portc.1
#define    di port.2
#define    dic portc.2
#define    do port.3
#define    doc portc.3
#define    cs port.0
#define    csc portc.0
;=====
```

```

;以下部分不可修改
;=====
ifndef ht93lc46_asm
extern dataaddr    :byte
extern data3      :byte
extern data2      :byte
extern ht93lc46_write    :near
extern ht93lc46_read    :near
extern ht93lc46_ewen    :near
extern ht93lc46_ewds    :near
extern ht93lc46_eral    :near
extern ht93lc46_wral    :near
extern ht93lc46_erase   :near
endif
;=====
;宣告提供给外部程序使用的变量
public dataaddr      ;数据的页内地址
public data3         ;存取的数据的高八位
public data2         ;存取的数据的低八位
;宣告提供给外部程序使用的子程序
public ht93lc46_eral ;
public ht93lc46_wral ;
public ht93lc46_ewen ;
public ht93lc46_ewds ;
public ht93lc46_write ;
public ht93lc46_read ;
public ht93lc46_erase ;
;=====
;数据段
ht93lc46data .section 'data'
dataaddr db ? ;操作地址
data3 db ? ;操作数据寄存器,高八位
data2 db ? ;操作数据寄存器
data1 db ? ;进行移位的数据寄存器
movb db ? ;循环移位次数寄存器
reg db ? ;时延数据寄存器
reg1 db ? ;时延数据寄存器
;=====
;代码段
ht93lc46code .section 'code'
;=====
; READ--读取数据
; 描述:从 EEPROM 的指定地址处读取数据

```

```

; 入口参数 dataaddr:byte 指定地址
; 出口参数 :data2 :byte 读取的数据的低八位
;          data3 :byte 读取的数据的高八位
; 堆栈使用:    1
;=====
ht93lc46_read    proc
    call    ht93_start        ;开始信号
    mov     a, oc_read
    mov     data1, a
    mov     a,2                ;写入 2 位 op-code 代码
    call    wbit
    mov     a, dataaddr
    mov     data1, a
    rl     data1
    ifdef   bit8
        mov     a,7            ;写入 7 位 dataaddr
    endif
    ifdef   bit16
        rl     data1
        mov     a,6            ;写入 6 位 dataaddr
    endif
    call    wbit
    nop
    call    rbit
    ifdef   bit16
        mov     a, data2
        mov     data3, a
        call    rbit
    endif
    clr     cs
    ret
ht93lc46_read    endp
;=====
; WRITE--写入数据
; 描述: 往 EEPROM 的指定地址处写入数据
; 入口参数 :dataaddr:byte 指定地址
;          data2 :byte 写入的数据的低八位
;          data3 :byte 写入的数据的高八位
; 出口参数:    无
; 堆栈使用:    1
;=====
ht93lc46_write   proc
    call    ht93_start

```

```

mov    a, oc_write
mov    data1, a
mov    a, 2                ;写入 2 位 op-code 代码
call   wbit
mov    a, dataaddr
mov    data1, a
rl     data1
ifdef  bit8
    mov    a, 7            ;写入 7 位 dataaddr
endif
ifdef  bit16
    rl     data1
    mov    a, 6            ;写入 6 位 dataaddr
endif
call   wbit
nop
ifdef  bit16
    mov    a, data3
    mov    data1, a
    mov    a, 8
    call   wbit
endif
mov    a, data2
mov    data1, a
mov    a, 8
call   wbit
clr    cs
call   delay
call   mverify
clr    cs
ret

ht93lc46_write      endp
;=====
; ERASE--擦除数据
; 描述：往 EEPROM 指定地址处写入数据 1
; 入口参数：    dataaddr:byte 指定地址
; 出口参数：    无
; 堆栈使用：    1
;=====
ht93lc46_erase      proc
    call   ht93_start
    mov    a, oc_erase
    mov    data1, a

```

```

mov    a,2                ;写入 2 位 op-code 代码
call   wbit
mov    a, dataaddr
mov    data1, a
rl     data1
ifdef  bit8
    mov a,7                ;写入 7 位 dataaddr
endif
ifdef  bit16
    rl  data1
    mov a,6                ;写入 6 位 dataaddr
endif
call   wbit
clr    cs
call   delay
call   mverify
clr    cs
ret

ht93lc46_erase    endp
;=====
; EWDS--写除能
; 描述：写除能，使 EEPROM 不能进行写入的操作
; 入口参数：    无
; 出口参数：    无
; 堆栈使用：    1
;=====
ht93lc46_ewds    proc
    call ht93_start
    mov  a, oc_ewds
    mov  data1, a
    mov  a,8                ;写入 8 位 op-code 代码
    call wbit
    ifdef bit8
        clr  sk
        set  sk
        nop
    endif
    clr  sk
    clr  cs
    ret
ht93lc46_ewds    endp
;=====
; EWEN--写置能

```

; 描述：写置能，使 EEPROM 能进行写入的操作

; 入口参数： 无

; 出口参数： 无

; 堆栈使用： 1

=====

ht93lc46_ewen proc

call ht93_start

mov a, oc_ewen

mov data1, a

mov a,8 ;写入 8 位 op-code 代码

call wbit

ifdef bit8

clr sk

set sk

nop

endif

clr sk

clr cs

ret

ht93lc46_ewen endp

=====

; ERAL--擦除所有地址

; 描述：将 EEPROM 中所有地址处写入数据 1

; 入口参数： 无

; 出口参数： 无

; 堆栈使用： 1

=====

ht93lc46_eral proc

call ht93_start

mov a, oc_eral

mov data1, a

mov a,8 ;写入 8 位 op-code 代码

call wbit

ifdef bit8

clr sk

set sk

nop

endif

clr sk

clr cs

call delay

call mverify

clr cs

```

ret
ht93lc46_eral    endp
;=====
; WRAL--写所有地址
; 描述：将数据写入所有 EEPROM 空间
; 入口参数：data2  :byte 写入的数据的低八位
;           data3  :byte 写入的数据的高八位
; 出口参数：  无
; 堆栈使用：   1
;=====
ht93lc46_wral    proc
    call  ht93_start
    mov   a, oc_wral
    mov   data1, a
    mov   a,8                ;写入 8 位 op-code 代码
    call  wbit
    ifdef bit8
        clr   sk
        set   sk
        nop
    endif
    clr   sk
    ifdef bit16
        mov   a, data3
        mov   data1, a
        mov   a,8
        call  wbit
    endif
    mov   a, data2
    mov   data1, a
    mov   a,8
    call  wbit
    clr   cs
    call  delay
    call  mverify
    clr   cs
    ret
ht93lc46_wral    endp
;=====
; 开始信号
;=====
ht93_start    proc
    set   cs

```

```

    clr    sk
    set    di
    nop
    nop
    set    sk
    nop
    clr    sk
    ret
ht93_start    endp
;=====
; 写 n bit 数据子程序, n 由 acc 决定
;=====
wbit        proc
    mov    movb, a
loop1:
    clr    sk
    rl     data1
    snz    data1.0
    jmp    loop1_1
    set    di
    jmp    loop1_2
loop1_1:
    clr    di
loop1_2:
    nop
    set    sk
    nop
    sdz    movb
    jmp    loop1
    clr    sk
    ret
wbit        endp
;=====
;读 8 bit 数据子程序
;=====
rbit        proc
    mov    a, 08h
    mov    movb, a
loop_r:
    rl     data2
    set    sk
    nop
    snz    do

```

```

        jmp     loops_0
        set     data2.0
        jmp     loops_1
loops_0:
        clr     data2.0
loops_1:
        clr     sk
        sdz     movb
        jmp     loop_r
        ret
rbit      endp

;=====
;检测 DO 是否有 HIGH 信号，即操作是否完成
;=====
mverify   proc
        set     cs
        nop
        nop
        nop
check:
        snz     do
        jmp     check
        ret
mverify   endp

;=====
delay     proc
        set     reg1
        mov     a, 06h
        mov     reg, a
lpy:
        sdz     reg1
        jmp     lpy
        sdz     reg
        jmp     lpy
        ret
delay     endp

;=====
;程序 ht93lc46.ASM 结束
;=====

```