- **产品名称**

MK7XXXX系列芯片

- **标题**

如何使用MK7XXXX系列芯片读写 24CXX

- **简介**

MKT的MK7XXXX系列IC是利用片内提供的资源，用软件模拟的方法来进行数据存储器扩展的。24CXX系列是获得广泛应用的两线制E^2 PROM，它的工作时序（如图 1）完全符合I^2C接口协议的规范。

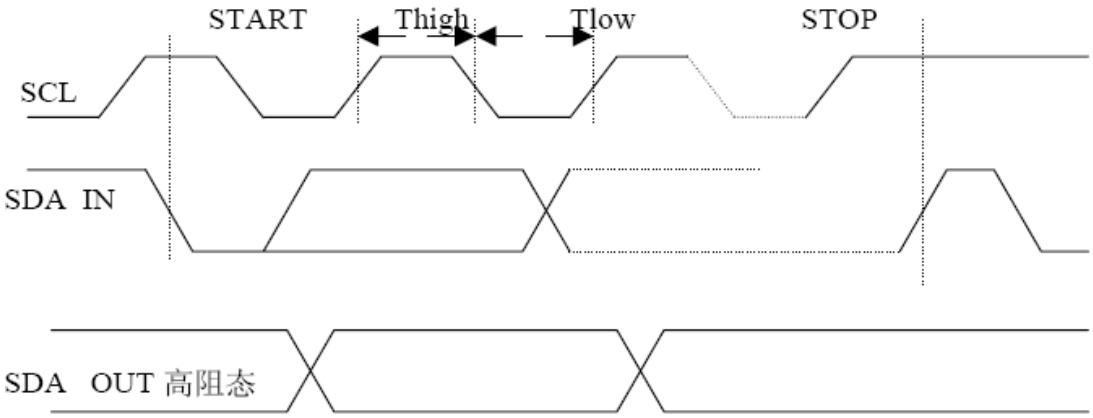

图 1　　24CXX 的工作时序图

SDA 和 SCL 两总线闲置时总为高电平。SCL 为高电平期间数据的改变表示“开始”和“停止”两种状态。当 SCL 处于高电平时，SDA 由高变低表示一个开始状态，反之由低变高则表示一个停止状态（参考图 2）。其中，“开始状态”必须在其它操作进行之前执行，而停止状态则中止所有操作。除了以上两种操作状态外，24CXX 与外界的通讯还需要另外一种状态，即“确认”（ACK）状态。总线上的任何数据接收设备必须将 SDA 总线置于低电平，以确认它成功地收到每个字节（所有的地址和数据都是以 8 位码串行输入或输出的），该确认是在每个字节之后第 9 个时钟周期发生（见图 3）。24CXX 也通过在收到每个地址或数据码之后置 SDA 为低电平的方式予以确认。为了正确访问 24CXX，外部数据传送设备必须在发出“开始”状态之后，随即给出一个地址码。该码被称为器件寻址码，该码高 4bit 为“1010”，这在 24CXX 系列中都是一样的，接下来 3bit 依次为 A2、A1 和 A0，他们与芯片各自的输入地址（与引脚硬件连接有关）相对应，未做硬件连接的引脚所对应的位用于页面寻址。最后一位是读写操作选择位，该位为 0 时激发写操作，为 1 时激发读操作。

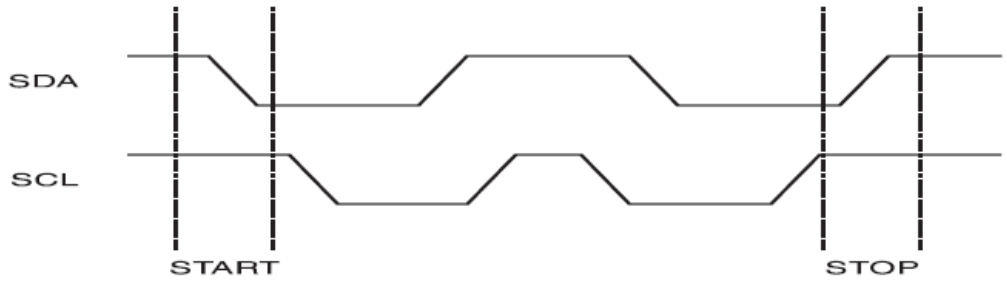

图 2　　START 和 STOP 的时序定义

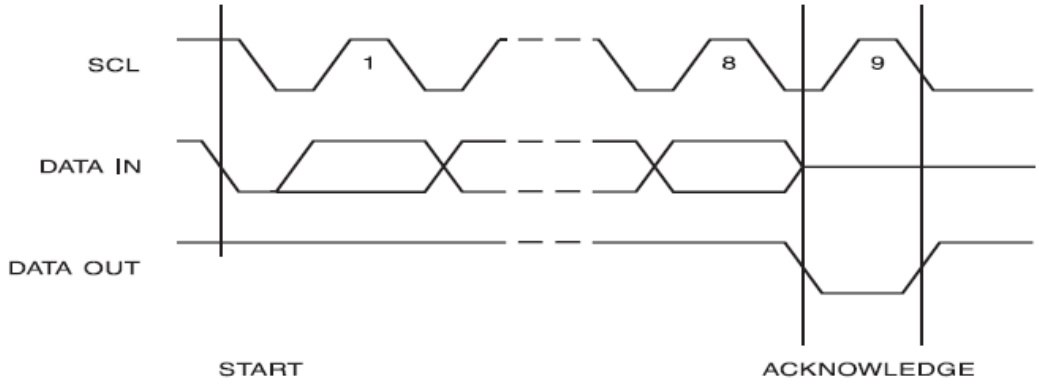

图 3　　ACK 应答信号的时序定义

24CXX 写操作分为写字节和写页面两种方式，而读操作则分为立即地址读取、随机地址读取和顺序地址读取三种。

下面是一个 MK7A11P 读写 24C32A 的实例，其线路图见图 4。主要功能是如下：

1. 将 MK7A11P 的 0X31～0X37 里面的数据分别写入到 24C32A 里面，然后进入 SLEEP。
2. 将写入到 24C32A 的数据分别读出，并存入 MK7A11P 的 0X38～0X3E 里面。
3. 根据按键的次序分别调出 MK7A11P 的 0X38～0X3E 里面的数据，并送显示，如按第 1 次显示 0X38 里面的数据，_ _ _ _，按第 7 次显示 0X3E 里面的数据，按第 8 次又开始显示 0X38 里面的数据，以后依次循环。并且 RAM 里面的内容与 LED 的对应关系如下表：

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
不显示	LED7	LED6	LED5	LED4	LED3	LED2	LED1

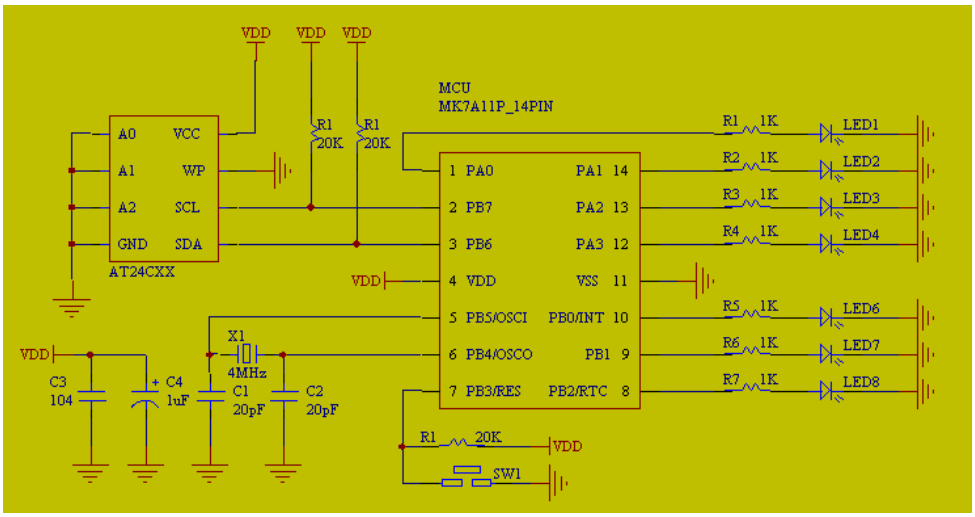

图 4 MK7A11P 读写 24C32A

● DEMO 程序

➢ 汇编程序文档

```
;-------------------------------------------------------------------------------------------------------------
#include    "mk7a11p_hw.inc"                ;编译该文档需包含"mk7a11p_hw.inc"文件
;-------------------------------------------------------------------------------------------------------------
;芯片型号（mk7a11p）
;-------------------------------------------------------------------------------------------------------------
;配置寄存器设置说明（CONFIG）
;1----------FOSC=NS                        ;LS,NS,HS,RC
;2----------INRC=OFF                       ;ON,OFF
;3----------CPT=OFF                        ;ON,OFF
;4----------WDTE=Disable                   ;Enable,Disable
;5----------LV=Low Vol Reset ON            ;Low Vol Reset ON,Low Vol Reset OFF
;6----------RESET=...reset...              ;...input...,...reset...
;-------------------------------------------------------------------------------------------------------------
rxbuf           equ         0x20        ;接收缓冲寄存器
txbuf           equ         0x21        ;发送缓冲寄存器
count           equ         0x22        ;字节计数寄存器
dat_r           equ         0x23        ;数据缓存
dis_r           equ         0x24        ;按键次数
;-------------------------------------------------------------------------------------------------------------
data1           equ         0x31
data2           equ         0x32
```

```
data3           equ             0x33
data4           equ             0x34
data5           equ             0x35
data6           equ             0x36
data7           equ             0x37
;-------------------------------------------------------------------------------------------------------------
#define         scl             7               ;时钟
#define         sda             6               ;数据
;-------------------------------------------------------------------------------------------------------------
                org             0x3ff           ;mk7a11p 的复位向量地址定义
                lgoto           main            ;跳转到主程序入口
;-------------------------------------------------------------------------------------------------------------
                org             0x100           ;主程序入口地址定义
main
                ;PortA 端口方向及状态设定
                movla           b'11110000'
                iodir           porta
                clr             porta
                clr             pa_pdm
                ;-----------------------------------------------------------------------------------------------
                ;PortB 端口方向及状态设定
                movla           b'00000000'
                iodir           portb
                movam           pb_pup
                clr             portb
                clr             pb_pod
                clr             pb_pdm
                clr             wake_up
                ;-----------------------------------------------------------------------------------------------
                ;中断禁止
                clr             irqm
                clr             irqf
                ;-----------------------------------------------------------------------------------------------
                ;配置 TMR0,预分频比为 1:8
                ;TMR0 初始值为 250
                ;movla          b'11000010'
                ;select
                ;movla          .250
                ;movam          tmr0
                ;-----------------------------------------------------------------------------------------------
```

```
                ;向 0x31-0x37RAM 里面送数据
                movla           0x71
                movam           data1
                movla           0x62
                movam           data2
                movla           0x53
                movam           data3
                movla           0x44
                movam           data4
                movla           0x35
                movam           data5
                movla           0x26
                movam           data6
                movla           0x17
                movam           data7
                ;-----------------------------------------------------------------------------------------
                clr             0x38
                clr             0x39
                clr             0x3a
                clr             0x3b
                clr             0x3c
                clr             0x3d
                clr             0x3e
                ;-----------------------------------------------------------------------------------------
                btss            status,pd
                lgoto           read_data
;-----------------------------------------------------------------------------------------------------
write_data
                ;将 0x31-0x37RAM 里面的数据读出分别
                ;写到 24C32A 的 0011h-0017h 地址里面
                ;-----------------------------------------------------------------------------------------
                lcall           condstart       ;开始
                ;-----------------------------------------------------------------------------------------
                movla           0xa0
                movam           txbuf
                lcall           tx8bit          ;发送写命令
                ;-----------------------------------------------------------------------------------------
                movla           0x00
                movam           txbuf
                lcall           tx8bit          ;发送写地址 1
```

```
                ;-------------------------------------------------------------------------------------------------------
                movla       0x10
                movam       txbuf
                lcall       tx8bit          ;发送写地址 2
                ;-------------------------------------------------------------------------------------------------------
                movla       0x31
                movam       fsr             ;fsr 指向 0x31
write_loop
                mov         indf,a
                movam       txbuf
                lcall       tx8bit          ;发送数据
                ;-------------------------------------------------------------------------------------------------------
                inc         fsr,m           ;fsr 指向下一寄存器
                ;-------------------------------------------------------------------------------------------------------
                mov         fsr,a
                andla       b'00111111'
                xorla       0x38
                btss        status,z
                lgoto       write_loop      ;7 个数据未发完继续发送
                ;-------------------------------------------------------------------------------------------------------
                lcall       condstop        ;结束
                lgoto       into_sleep
;-----------------------------------------------------------------------------------------------------------------------
read_data
                ;分别读出 24C32A 的 0011h-0017h 地址里面的数据
                ;-------------------------------------------------------------------------------------------------------
                lcall       condstart       ;开始
                ;-------------------------------------------------------------------------------------------------------
                movla       0xa0
                movam       txbuf
                lcall       tx8bit          ;发送写命令
                ;-------------------------------------------------------------------------------------------------------
                movla       0x00
                movam       txbuf
                lcall       tx8bit          ;发送写地址 1
                ;-------------------------------------------------------------------------------------------------------
                movla       0x10
                movam       txbuf
                lcall       tx8bit          ;发送写地址 2
                ;-------------------------------------------------------------------------------------------------------
```

```
                ;-------------------------------------------------------------------------------------------------
                lcall           condstart       ;开始
                ;-------------------------------------------------------------------------------------------------
                movla           0xa1
                movam           txbuf
                lcall           tx8bit          ;发送读命令
                ;-------------------------------------------------------------------------------------------------
                movla           0x38
                movam           fsr
read_loop
                lcall           rx8bit          ;读数据
                ;-------------------------------------------------------------------------------------------------
                mov             rxbuf,a
                movam           indf            ;将读出的数据保存给
                                                ;fsr 当前指向的寄存器
                ;-------------------------------------------------------------------------------------------------
                inc             fsr,m           ;fsr 指向下一寄存器
                ;-------------------------------------------------------------------------------------------------
                mov             fsr,a
                andla           b'00111111'
                xorla           0x3f
                btsc            status,z
                lgoto           read_end        ;最后一个数据不用应答
                ;-------------------------------------------------------------------------------------------------
rxack
                bc              portb,sda       ;数据线拉低
                nop
                nop
                nop
                nop
                nop
                bs              portb,scl       ;发送应答信号
                nop
                nop
                nop
                nop
                bc              portb,scl       ;应答信号发送完成将时钟线拉低
                lgoto           read_loop
                ;-------------------------------------------------------------------------------------------------
read_end
```

```
                movla       b'00000000'
                iodir       portb           ;scl 和 sda 设为输出
                nop
                bs          portb,sda
                bc          portb,scl
                nop
                nop
                nop
                nop
                nop
                nop
                nop
                bs          portb,scl
                lcall       condstop
                ;-------------------------------------------------------------------------------------------------
                movla       0x38
                movam       fsr             ;读完数据后将 fsr 指向 0x38
;-----------------------------------------------------------------------------------------------------------
display
                mov         dis_r,a
                add         fsr,m           ;将 fsr 指向指定的寄存器
                mov         indf,a
                movam       dat_r           ;将指定寄存器里面的数据
                                            ;输送给缓存寄存器
                ;-------------------------------------------------------------------------------------------------
                movla       b'00001111'
                and         dat_r,a
                movam       porta           ;将缓存寄存器里面的低 4bit
                                            ;数据输送给 PortA 显示
                ;-------------------------------------------------------------------------------------------------
                rr          dat_r,m
                rr          dat_r,m
                rr          dat_r,m
                rr          dat_r,m
                movla       b'00000111'
                and         dat_r,a
                movam       portb           ;将缓存寄存器里面的高 3bit
                                            ;数据输送给 PortB 显示
                ;-------------------------------------------------------------------------------------------------
                inc         dis_r,m
```

```
        movla   .7
        sub     dis_r,a
        btsc    status,c
        clr     dis_r
;-------------------------------------------------------------------------------------------------------------------
into_sleep
        nop
        nop
        nop
        sleep
        nop
        nop
        nop
;=====================================================================
;以下是子程序
condstart                               ;I2C 接口启动信号
        movla   b'00000000'
        iodir   portb           ;scl 和 sda 设为输出
        nop
        bs      portb,sda
        nop
        bs      portb,scl
        nop
        nop
        nop
        nop
        bc      portb,sda
        nop
        nop
        nop
        nop
        bc      portb,scl
        ret
;-------------------------------------------------------------------------------------------------------------------
condstop
        movla   b'00000000'
        iodir   portb           ;scl 和 sda 设为输出
        nop
        bs      portb,scl
        nop
```

```
                bc          portb,sda
                nop
                nop
                nop
                bs          portb,sda
                nop
                nop
                nop
                ret
;-------------------------------------------------------------------------------------------------------------------
rx8bit
                movla       b'01000000'
                iodir       portb           ;scl 设为输出,sda 设为输入
                clr         rxbuf
                movla       .8
                movam       count
rxlp
                bc          portb,scl
                nop
                nop
                nop
                bs          portb,scl
                nop
                nop
                nop
                bc          status,c
                btsc        portb,sda
                bs          status,c
                rl          rxbuf,m
                decsz       count,m
                lgoto       rxlp
                bc          portb,scl       ;时钟线拉低
                movla       b'00000000'
                iodir       portb           ;scl 和 sda 设为输出
                ret
;-------------------------------------------------------------------------------------------------------------------
tx8bit
                movla       b'00000000'
                iodir       portb           ;scl 和 sda 设为输出
                movla       .8
```

```
                movam       count
        txlp
                rl          txbuf,m
                btsc        status,c
                bs          portb,sda       ;设定数据 1（时钟线为低时设定数据）
                btss        status,c
                bc          portb,sda       ;设定数据 0（时钟线为低时设定数据）
                nop
                nop
                nop
                bs          portb,scl       ;发送一位数据（上升沿有效）
                nop
                nop
                nop
                bc          portb,scl       ;时钟线拉低
                decsz       count,m
                lgoto       txlp
                ;-----------------------------------------------------------------------------------------------
                movla       b'01000000'
                iodir       portb           ;scl 设为输出,sda 设为输入
                nop
        txack
                bc          portb,scl       ;将时钟线拉低
                nop
                nop
                nop
                bs          portb,scl       ;给一上升沿脉冲
                nop
                nop
                nop
                btsc        portb,sda
                lgoto       txack           ;未接收到响音信号，继续
                bc          portb,scl       ;接收到响应信号后将时钟线拉低
                ret
        ;-----------------------------------------------------------------------------------------------
                end
```

- mk7a11p_hw.inc 文档

```
;---------Define special register(Define SFR) ----------------------------------------------------------
indf            equ             0x00
tmr0            equ             0x01
```

```
pc              equ             0x02
status          equ             0x03
fsr             equ             0x04
porta           equ             0x05            ;porta(0-3)
portb           equ             0x06            ;portb(0-7)
;-----------------------------------------------------------------------------------------------------------
irqm            equ             0x09
irqf            equ             0x0a
;-----------------------------------------------------------------------------------------------------------
pa_pdm          equ             0x0b
pb_pup          equ             0x0c
pb_pdm          equ             0x0d
pb_pod          equ             0x0e
wake_up         equ             0x0f
;--------Define [status    Register] special bit-----------------------------------------------------------
c               equ             0
dc              equ             1
z               equ             2
pd              equ             3
to              equ             4
;--------Define [irqm      Register] special bit-----------------------------------------------------------
tm0m            equ             0
extm            equ             1
intm            equ             7
;--------Define [irqf      Register] special bit-----------------------------------------------------------
tm0f            equ             0
extf            equ             1
;--------Define [pa_pdm    Register] special bit-----------------------------------------------------------
da0             equ             0
da1             equ             1
da2             equ             2
da3             equ             3
;--------Define [pb_pup    Register] special bit-----------------------------------------------------------
ub0             equ             0
ub1             equ             1
ub2             equ             2

ub4             equ             4
ub5             equ             5
ub6             equ             6
```

```
ub7             equ         7
;--------Define [pb_pdm     Register] special bit-------------------------------------------------------------
db0             equ         0
db1             equ         1
db2             equ         2

inte            equ         6
rtce            equ         7
;--------Define [pb_pod     Register] special bit-------------------------------------------------------------
ob0             equ         0
ob1             equ         1
ob2             equ         2

ob4             equ         4
ob5             equ         5
ob6             equ         6
ob7             equ         7
;--------Define [wake_up    Register] special bit-------------------------------------------------------------
en0             equ         0
en1             equ         1
en2             equ         2
en3             equ         3
en4             equ         4
en5             equ         5
en6             equ         6
en7             equ         7
;-----------------------------------------------------------------------------------------------------------------
```