模拟 I²C 总线汇编程序软件包

一、概述

为了非常方便地对 I²C 从器件进行快速的、正确的读写操作,我们为此而设计出虚拟 I²C 总线操作平台软件包。本软件包是主方式下的虚拟 I²C 总线软件包,只要用户给子程序提供几个主要的参数,即可轻松地完成任何 I²C 总线外围器件的应用程序设计。

I²C 总线是 PHILIPS 公司推出的芯片间串行数据传输总线，2 根线(SDA、SCL)即可实现完善的全双工同步数据传送，能够十分方便地地构成多机系统和外围器件扩展系统。I²C 器件是把 I²C 的协议植入器件的 I/O 接口，使用时器件直接挂到 I²C 总线上,这一特点给用户在设计应用系统带来了极大的便利。I²C 器件无须片选信号，是否选中是由主器件发出的 I²C 从地址决定的，而 I²C 器件的从地址是由 I²C 总线委员会实行统一发配。我们推出的 I²C 总线的操作平台软件包，只要你给出器件从地址[，子地址（注：PCF8574 无子地址）]，即可进行字节读，字节写，多字节读，多字节写，能够非常方便地使用 I²C 器件，无须你介入底层的 I²C 操作协议。

二、编程序软件包

此软件包是用在单主 I²C 总线上，硬件接口是 SDA，SCL，使用 MCU 的 I/O 口来模拟 SDA/SCL 总线。设计有/无子地址的子程序是根据 I²C 器件的特点，目的在于将地址和数据彻底分开。软件包的接口界面为：

IRDBYTE	（无子地址）读单字节数据	（现行地址读）
IWRBYTE	（无子地址）写单字节数据	（现行地址写）
IRDNBYTE	（有子地址）读 N 字节数据	
IWRNBYTE	（有子地址）写 N 字节数据	

说明：现行地址读/写即专指无子地址的器件，不给定子地址的读/写操作。

软件包占用内部资源：R0、R1、R2、R3、ACC、Cy。

使用前须定义变量：SLA 器件从地址，SUBA 器件子地址，NUMBYTE 读/写的字节数 ，位变量 ACK。

使用前须定义常量：SDA 、SCL 总线位，MTD 发送数据缓冲区首址，MRD 接收数据缓冲区首址。

```
;*********************************************************************
;VI2C_ASM.ASM
;I²C 软件包的底层子程序，使用前要定义好 SCL 和 SDA。在标准 80C51 模式
;（12 Clock）下,对主频要求是不高于 12MHz(1 个机器周期 1us);若 Fosc>12MHz
;则要增加相应的 NOP 指令数。在使用本软件包时，请在你的程序的末尾加入
;$INCLUDE (VI2C_ASM.ASM)即可。
;*********************************************************************
;启动 I²C 总线子程序
START:
    SETB    SDA
    NOP
    SETB    SCL                 ;起始条件建立时间大于 4.7us
    NOP
    NOP
    NOP
    NOP
```

```
        NOP
        CLR     SDA
        NOP                     ;起始条件锁定时大于 4us
        NOP
        NOP
        NOP
        NOP
        NOP
        CLR     SCL             ;钳住总线，准备发数据
        NOP
        RET

    ;结束总线子程序
    STOP:
        CLR     SDA
        NOP
        SETB    SCL             ;发送结束条件的时钟信号
        NOP                     ;结束总线时间大于 4us
        NOP
        NOP
        NOP
        NOP
        SETB    SDA             ;结束总线
        NOP                     ;保证一个终止信号和起始信号的空闲时间大于 4.7us
        NOP
        NOP
        NOP
        RET

;发送应答信号子程序
MACK:
        CLR     SDA             ;将 SDA 置 0
        NOP
        NOP
        SETB    SCL
        NOP                     ;保持数据时间，即 SCL 为高时间大于 4.7us
        NOP
        NOP
        NOP
        NOP
        CLR     SCL
        NOP
        NOP
        RET
```

```
;发送非应答信号
MNACK:
      SETB    SDA                 ;将 SDA 置 1
      NOP
      NOP
      SETB    SCL
      NOP
      NOP                         ;保持数据时间，即 SCL 为高时间大于 4.7us
      NOP
      NOP
      NOP
      CLR     SCL
      NOP
      NOP
      RET

; 检查应答位子程序
; 返回值，ACK=1 时表示有应答
CACK:
      SETB    SDA
      NOP
      NOP
      SETB    SCL
      CLR     ACK
      NOP
      NOP
      MOV     C,SDA
      JC      CEND
      SETB    ACK                 ;判断应答位
CEND:
      NOP
      CLR     SCL
      NOP
      RET

;发送字节子程序
;字节数据放入 ACC
;每发送一字节要调用一次 CACK 子程序，取应答位
WRBYTE:
      MOV     R0,#08H
WLP:
      RLC     A                   ;取数据位
      JC      WR1
      SJMP    WR0                 ;判断数据位
```

```
WLP1:
        DJNZ    R0,WLP
        NOP
        RET
WR1:
        SETB    SDA                     ;发送 1
        NOP
        SETB    SCL
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        CLR     SCL
        SJMP    WLP1
WR0:
        CLR     SDA                     ;发送 0
        NOP
        SETB    SCL
        NOP
        NOP
        NOP
        NOP
        NOP
        CLR     SCL
        SJMP    WLP1

;读取字节子程序
;读出的值在 ACC
;每取一字节要发送一个应答/非应答信号
RDBYTE:
        MOV     R0,#08H
RLP:
        SETB    SDA
        NOP
        SETB    SCL             ;时钟线为高，接收数据位
        NOP
        NOP
        MOV     C,SDA           ;读取数据位
        MOV     A,R2
        CLR     SCL             ;将 SCL 拉低，时间大于 4.7us
        RLC     A               ;进行数据位的处理
        MOV     R2,A
        NOP
```

```
        NOP
        NOP
        DJNZ    R0,RLP              ;未够 8 位，再来一次
        RET

; 无子地址器件写字节数据
; 入口参数: 数据为 ACC、器件从地址 SLA
; 占用:  A、R0、CY
IWRBYTE:
      PUSH    ACC
IWBLOOP:
      LCALL   START               ;起动总线
        MOV     A,SLA
        LCALL   WRBYTE              ;发送器件从地址
        LCALL   CACK
        JNB     ACK,RETWRB          ;无应答则跳转
        POP     ACC                 ;写数据
        LCALL   WRBYTE
        LCALL   CACK
        LCALL   STOP
        RET
RETWRB:
      POP     ACC
        LCALL   STOP
        RET

;无子地址器件读字节数据
;入口参数: 器件从地址 SLA
;出口参数: 数据为 ACC
;占用 A 、R0、R2 、CY
IRDBYTE:
      LCALL   START
      MOV     A,SLA               ;发送器件从地址
        INC     A
        LCALL   WRBYTE
        LCALL   CACK
        JNB     ACK,RETRDB
        LCALL   RDBYTE                  ;进行读字节操作
        LCALL  MNACK                    ;发送非应信号
RETRDB:
      LCALL  STOP                 ;结束总线
      RET

;向器件指定子地址写 N 个数据
```

```
;入口参数:  器件从地址 SLA、器件子地址 SUBA 、发送数据缓冲区 MTD、发送字节数 NUMBYTE
; 占用：  A 、R0 、R1 、R3 、CY
IWRNBYTE:
      MOV     A,NUMBYTE
      MOV     R3,A
      LCALL   START                ;起动总线
      MOV     A,SLA
      LCALL   WRBYTE               ;发送器件从地址
      LCALL   CACK
      JNB     ACK,RETWRN           ;无应答则退出
      MOV     A,SUBA               ;指定子地址
      LCALL   WRBYTE
      LCALL   CACK
      MOV     R1,#MTD
WRDA:
      MOV     A,@R1
      LCALL   WRBYTE               ;开始写入数据
      LCALL   CACK
      JNB     ACK,IWRNBYTE
      INC     R1
      DJNZ    R3,WRDA              ;判断写完没有
RETWRN:
      LCALL   STOP
      RET

;向器件指定子地址读取 N 个数据
;入口参数:  器件从地址 SLA、器件子地址 SUBA、接收字节数 NUMBYTE
;出口参数:  接收数据缓冲区 MTD
;占用：A、 R0、 R1、 R2、 R3、 CY
IRDNBYTE:
      MOV     R3,NUMBYTE
      LCALL   START
      MOV     A,SLA
      LCALL   WRBYTE               ;发送器件从地址
      LCALL   CACK
      JNB     ACK,RETRDN
      MOV     A,SUBA               ;指定子地址
      LCALL   WRBYTE
      LCALL   CACK
      LCALL   START                ;重新起动总线
      MOV     A,SLA
      INC     A                        ;准备进行读操作
      LCALL   WRBYTE
      LCALL   CACK
```

```
        JNB     ACK,IRDNBYTE
        MOV     R1,#MRD
RDN1:
        LCALL   RDBYTE                  ;读操作开始
        MOV     @R1,A
        DJNZ    R3,SACK
        LCALL   MNACK                   ;最后一字节发非应答位
RETRDN:
        LCALL   STOP                    ;并结束总线
        RET
SACK:
        LCALL   MACK
        INC     R1
        SJMP    RDN1
```

三、应用举例

```
;为软件包定义变量
ACK             BIT     10H             ;应答标志位
SLA             DATA    50H             ;器件从地址
SUBA            DATA    51H             ;器件子地址
NUMBYTE         DATA    52H             ;读/写的字节数
;使用前定义常量
SDA             EQU     P1.3            ; I2C 总线定义
SCL             EQU     P1.2
MTD             EQU     30H             ;发送数据缓冲区首址 (缓冲区 30H－3FH)
MRD             EQU     40H             ;接收数据缓冲区首址 （缓冲区 40－4FH）
;定义器件地址
CSI24WCXX       EQU     0A0H

                ORG     0000H
                AJMP    MAIN

                ORG     0080H
MAIN:           MOV     R4,#0F0H        ;延时，等待其它芯片复位好
                DJNZ    R4,$
;对 24WCXX 指定单元进行写操作,指定的子地址放入 SUBA，数据依次放入 MTD 缓冲区
WR24WCXX:
                MOV     SLA,#CSI24WCXX
                MOV     SUBA,#30H       ;指定存储地址
                MOV     NUMBYTE,#01H    ;写入一字节数据
                MOV     MTD,#58H        ;写入的数据放入 MTD 缓冲区
                LCALL   IWRNBYTE
                SJMP    $               ;操作结束
               ;
```

```
        $INCLUDE (VI2C_ASM.ASM)   ;包含 VIIC 软件包
        ;
END
```