

FT6XF2XX

Application note

目录

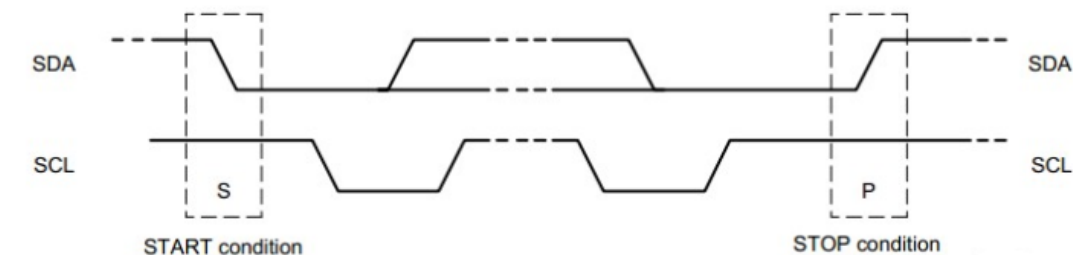
1	IIC 应用说明.....	3
2	应用范例.....	3

FT6XF2XX IIC 应用

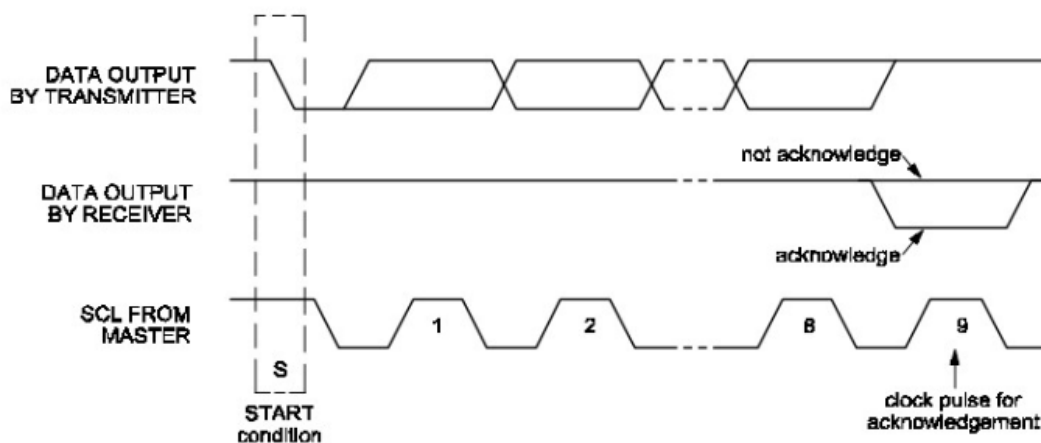
1 IIC 应用说明

I2C总线是一种串行数据总线，共二根信号线：双向的数据线SDA和时钟线SCL。

IIC协议：两条线可以挂多个设备。IIC设备里有个固化的地址，只有在两条线上传输的值等于（IIC设备）的地址时，才作出响应。



起始和停止条件



I²C 总线的响应

本说明以IC FT60F011A与存储芯片24C02为示范。

本程序中数据线SDA与SCL所对应的IO引脚：

```
#define IIC_SCL    RA4
```

```
#define IIC_SDA    RA2
```

2 应用范例

```

//*****
/* 文件名： Test_62F21X_IIC.asm
* 功能：   Test_62F21X_IIC 功能演示
* IC：    FT62F21X SOP8
* 晶振：   16M/4T
* 说明：   此演示程序位 62F21X_IIC 的演示程序.

```

```

*           该程序读取(24C02)0x12 地址的值,取反后存入 0x13 地址
*           FT62F21X SOP8
*           -----
* DemoPortOut -----|1(PA4)                (PA3)8 |-----NC
* NC-----|2(TKCAP)                (PA0)7 |-----NC
* NC-----|3(VDD)                (PA1)6 |-----NC
* NC-----|4(VSS)                (PA2)5  |-----DemoPortIn
*           -----
*/

```

```

//=====
#include <FT62F21X.INC>;
;=====
;RAM DEFINE
TEMP EQU 0X40
TEMP1 EQU 0X41
TEMP2 EQU 0X42
W_TMP EQU 0X43
S_TMP EQU 0X44
READPIN EQU 0X45
IICDATTEMP EQU 0X46
IICADRTEMP EQU 0X47
IICDATA EQU 0X48
IICADDR EQU 0X49
buff EQU 0X4a
#define f_ack buff, 0
count EQU 0X4b
;=====
;USER DEFINE
;=====
#define IIC_SCL PORTA,4
#define IIC_SDA PORTA,2
;=====
;PROGRAM START
;=====
ORG 0000H
LJUMP RESTART
ORG 0004H
LJUMP interrupt
;=====
;SYSTEM START
;=====
RESTART:
BANKSEL PORTA
LCALL INITIAL

```

```

BANKSEL    IICADDR
LDWI       0x12
STR        IICADDR
LCALL      IIC_READ

```

```

BANKSEL    IICDATA
LDWI       0X13
STR        IICADDR
COMR       IICDATA,R    ;取反
LCALL      IIC_WRITE    ;取反过后存到 0x13 地址

```

MAIN:

```

NOP
LJUMP      MAIN

```

```

;*****
;
;      中断程序
;*****

```

interrupt:

```

STR        W_TMP
SWAPR      STATUS,W
STR        S_TMP
BCR        STATUS,RP0

```

INT_RET:

```

SWAPR      S_TMP,0
STR        STATUS
SWAPR      W_TMP,1
SWAPR      W_TMP,0
RETI

```

```

;=====
;SYSTEM INITIAL
;=====

```

INITIAL:

```

BANKSEL    PORTA
LDWI       B'00000000'
STR        PORTA
BANKSEL    TRISA
LDWI       B'00001000' ;PA3 输入
STR        TRISA      ;SET IO Direction
LDWI       B'00000000'
STR        WPUA
LDWI       B'00000000'

```

```

    STR        OPTION_REG    ;SET OPTION
    LDWI       B'01110000'
    STR        OSCCON        ;SET OSCCON
    BANKSEL    INTCON
    LDWI       B'00000000'
    STR        INTCON

CLEAR_RAM:
    LDWI       40H
    STR        FSR

CLEAR_RAM_LOOP:
    CLRR       INDF
    INCR       FSR,F
    LDWI       80H
    XORWR      FSR,W
    BTSS       STATUS,Z
    LJUMP      CLEAR_RAM_LOOP
    RET

SDA_OUT:
    BANKSEL    TRISA
    LDWI       H'00'        ;PA2  输出
    ANDWR      TRISA,1      ;SET IO Direction
    BANKSEL    INTCON
    RET

SDA_IN:
    BANKSEL    TRISA
    LDWI       B'00000100'   ;PA2  输入
    IORWR      TRISA,1      ;SET IO Direction
    BANKSEL    INTCON
    RET

=====
;DELAY_5US 16MHZ/4T
=====
DELAY_5US:
    LDWI       H'05'
    STR        TEMP

DELAY_5USLOOP:
    CLRWDI
    DECRSZ     TEMP,F
    LJUMP      DELAY_5USLOOP
    RET

```

```

.....
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
.....
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
IIC_Start:
    LCALL  SDA_OUT      //SDA 线输出
    BSR    IIC_SDA
    BSR    IIC_SCL
    LCALL  DELAY_5US
    LCALL  DELAY_5US
    BCR    IIC_SDA      //START:when CLK is high,DATA change form high to low
    LCALL  DELAY_5US
    LCALL  DELAY_5US
    BCR    IIC_SCL      //钳住 I2C 总线，准备发送或接收数据
    RET

```

```

.....
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
.....
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
IIC_Stop:
    LCALL  SDA_OUT      //SDA 线输出
    BCR     IIC_SCL
    BCR     IIC_SDA      //STOP:when CLK is high DATA change form low to high
    LCALL  DELAY_5US
    LCALL  DELAY_5US
    BSR     IIC_SCL
    BSR     IIC_SDA      //发送 I2C 总线结束信号
    LCALL  DELAY_5US
    LCALL  DELAY_5US
    RET

```

```

;等待应答信号到来
;接收应答失败 f_ack=1
;接收应答成功 f_ack=0

```

```
IIC_Wait_Ack:
    CLRR    count
    BCR     f_ack
    LCALL   SDA_IN           //SDA 设置为输入
    BSR     IIC_SDA
    LCALL   DELAY_5US
    BSR     IIC_SCL
    LCALL   DELAY_5US
IIC_Wait_Ack_loop:
```

```

BTSS    IIC_SDA
LJUMP   IIC_Wait_Ack_right
INCR    count,1
LDWI    0xfa
SUBWR   count,0
BTSS    STATUS,0
LJUMP   IIC_Wait_Ack_loop
CLRR    count                ;等待超时
LCALL   IIC_Stop
BSR     f_ack
RET
IIC_Wait_Ack_right:
BCR     IIC_SCL                ;时钟输出 0
RET

```

```

;产生 ACK 应答

```

```

IIC_Ack:
BCR     IIC_SCL
LCALL   SDA_OUT
BCR     IIC_SDA
LCALL   DELAY_5US
BSR     IIC_SCL
LCALL   DELAY_5US
BCR     IIC_SCL
RET

```

```

;不产生 ACK 应答

```

```

IIC_NAck:
BCR     IIC_SCL
LCALL   SDA_OUT
BSR     IIC_SDA
LCALL   DELAY_5US
BSR     IIC_SCL
LCALL   DELAY_5US
BCR     IIC_SCL
RET

```

```

;IIC 发送一个字节

```

```

IIC_Send_Byte:
CLRR    count

```



```

LCALL SDA_OUT
BCR IIC_SCL ;拉低时钟开始数据传输
IIC_Send_Byte_loop:
LDWI 0x08
SUBWR count,0
BTSC STATUS,0
RET
INCR count,1
BTSS IICDATTEMP,7
LJUMP $+3
BSR IIC_SDA
LJUMP $+2
BCR IIC_SDA

BCR STATUS,0
RLR IICDATTEMP,1
LCALL DELAY_5US
BSR IIC_SCL
LCALL DELAY_5US
BCR IIC_SCL
LCALL DELAY_5US
LJUMP IIC_Send_Byte_loop

```

读 1 个字节, ack=1 时, 发送 ACK, ack=0, 发送 nACK

IIC Read Byte:

```
CLRR    count
CLRR    IICDATTEMP
LCALL   SDA_IN      ;SDA 设置为输入
```

IIC Read Byte loop:

LDWI	0x08
SUBWR	count,0
BTSC	STATUS,0
LJUMP	IIC_Read_Byte_end
INCR	count,1
BCR	IIC_SCL
LCALL	DELAY_5US
BSR	IIC_SCL
BCR	STATUS,0
RLR	IICDATTEMP,1
BTSC	IIC_SDA
INCR	IICDATTEMP,1

```

LCALL DELAY_5US
LJUMP IIC_Read_Byte_loop

```

```

IIC_Read_Byte_end:
    LCALL IIC_Ack
    RET

```

```

/*-----
* 函数名: IIC_READ
* 功能:   IIC 读出制定位置的数据
*         读出 IICADDR 存储器里面的数据 IICDATTEMP
*-----*/

```

```

IIC_READ:
    LCALL IIC_Start
    LDWI    0xa0
    STR     IICDATTEMP
    LCALL IIC_Send_Byte
    LCALL IIC_Wait_Ack
    BTSC    f_ack
    LJUMP IIC_READ ;填要读的数据地址
    LDR     IICADDR,0
    STR     IICDATTEMP
    LCALL IIC_Send_Byte
    LCALL IIC_Wait_Ack
    BTSC    f_ack
    LJUMP IIC_READ
    LCALL IIC_Start();
    LDWI    0xa1
    STR     IICDATTEMP
    LCALL IIC_Send_Byte
    LCALL IIC_Wait_Ack
    BTSC    f_ack
    LJUMP IIC_READ
    LCALL IIC_Read_Byte
    LCALL IIC_Stop
    LDR     IICDATTEMP,0
    STR     IICDATA
    RET

```

```

/*-----
* 函数名: IIC_WRITE
* 功能:   IIC 把数据 data 写入制定的位置 IICADDR
*-----*/

```

```

IIC_WRITE:
    LCALL IIC_Start

```

```
LDWI    0xa0
STR      IICDATTEMP
LCALL   IIC_Send_Byte
LCALL   IIC_Wait_Ack
BTSC    f_ack
LJUMP   IIC_WRITE
LDR      IICADDR,0
STR      IICDATTEMP
LCALL   IIC_Send_Byte
LCALL   IIC_Wait_Ack
BTSC    f_ack
LJUMP   IIC_WRITE
LDR      IICDATA,0
STR      IICDATTEMP
LCALL   IIC_Send_Byte
LCALL   IIC_Wait_Ack
BTSC    f_ack
LJUMP   IIC_WRITE
LCALL   IIC_Stop
RET
```

END

Fremont Micro Devices (SZ) Limited

#5-8, 10/F, Changhong Building, Ke-Ji Nan 12 Road, Nanshan District, Shenzhen, Guangdong 518057

Tel: (86 755) 86117811

Fax: (86 755) 86117810

Fremont Micro Devices (Hong Kong) Limited

#16, 16/F, Blk B, Veristrong Industrial Centre, 34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong

Tel: (852) 27811186

Fax: (852) 27811144

Fremont Micro Devices (USA), Inc.

42982 Osgood Road Fremont, CA 94539

Tel: (1-510) 668-1321

Fax: (1-510) 226-9918

Web Site: <http://www.fremontmicro.com/>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices, Incorporated (BVI) assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices, Incorporated (BVI). Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices, Incorporated (BVI) products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices, Incorporated (BVI). The FMD logo is a registered trademark of Fremont Micro Devices, Incorporated (BVI). All other names are the property of their respective own.