FT61F02X

IIC Application note

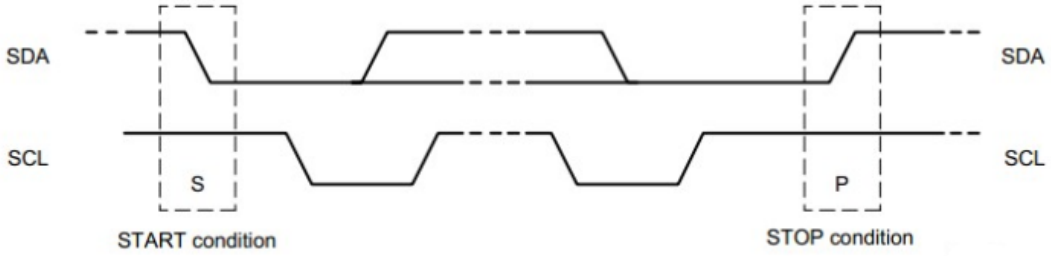
Table of contents

FT61F02x IICapplication

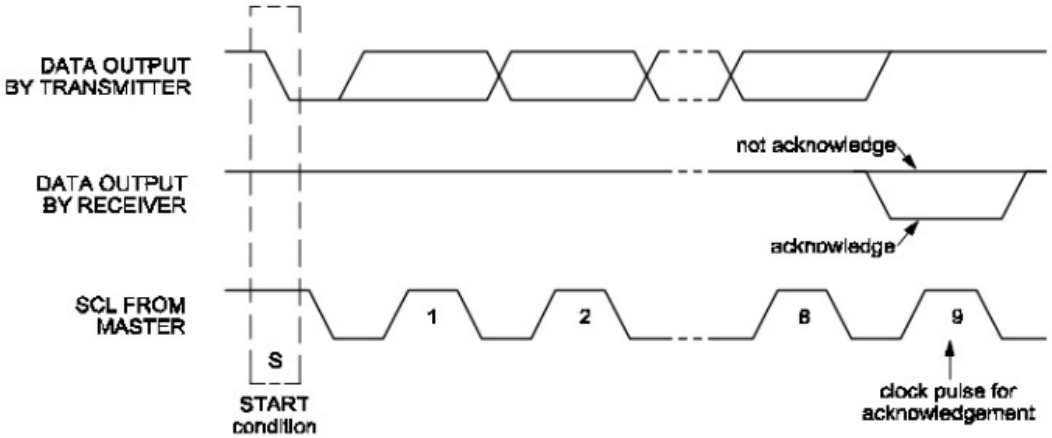
1. IICapplication notes

I2CThe bus is a serial data bus with two signal lines: bidirectional data linesSDAand clock lineSCL.

IICProtocol: Multiple devices can be connected to two lines.IICThere is a solidified address in the device, only the value transmitted on the two lines is equal to (IIC device) before responding.



起始和停止条件



I^2C 总线的响应

This note starts withIC FT61F023 SOP16with memory chip24C02For demonstration.

Data cable for this programSDAandSCLCorrespondingIOPin:

```
#define    IIC_SCL      PORTA,4
#define    IIC_SDA      PORTA,2
```

2.Application example

```
//*************************************************
********** /*file name:TEST_61F02x_IIC.c
* Function:     FT61F02x-IICDemo
*IC:            FT61F023 SOP16
* Crystal:      16M/4T
* illustrate:   This demo program bit61F02x_IICdemo program for . The program reads (24C02)0x12The
*               value of the address, reversed and stored0x13address
*
*                   FT61F023 SOP16
*                    ------------------------
* VDD----------|1(VDD)       (VSS)16|----------GND
* NC-----------|2(PA7)        (PA0)15|-----------NC
* NC-----------|3(PA6)        (PA1)14|-----------NC
* NC-----------|4(PA5)        (PA2)13|-----IIC_SDA
* NC-----------|5(PC3)        (PA3)12|-----------NC
* NC-----------|6(PC2)        (PC0)11|-----------NC
* IIC_SCL-----|7(PA4)         (PC1)10|-----------NC
* NC-----------|8(PC5)        (PC4)09|-----------NC
*                    ------------------------
*/
//************************************************* ***********
# include "SYSCFG.h"
//************************Macro definition ***************************
#define unchar unsigned char

#define   IIC_SCL PA4
#define   IIC_SDAPA2

#define   SDA_OUT     TRISA2 = 0
#define   SDA_IN      TRISA2 = 1

unchar IICReadData;
/*---------------------------------------------- -
* Function name:POWER_INITIAL
* Function:   Power-on system initialization
* enter:    none
* output:   none
-----------------------------------------------
---------*/ void POWER_INITIAL (void)
{
    OSCCON = 0B01110001;          //IRCF=111=16MHz/4T=4MHz,0.25μs //Temporarily
    INTCON = 0;                   disable all interrupts
    PORTA = 0B00000000;
```

```
    TRISA = 0B00000000;          //PAinput Output0-output1-enter
    PORTC = 0B00000000;
    TRISC = 0B00000000;          //PCinput Output0-output1-enter //PAPort pull-
    WPUA = 0B00000000;           up control1-pull up0-close pull //PCPort pull-up
    WPUC = 0B00000000;           control1-pull up0-close pull

    ANSEL = 0B00000000;
    OPTION = 0B00001000;         //Bit3=1, WDT MODE, PS=000=WDT RATE 1:1
    MSCKCON = 0B00000000;
    //Bit6->0,prohibitPA4,PC5Regulated output
    //Bit5->0,TIMER2the clock isFosc //Bit4->0,
    prohibitLVR
    CMCON0 = 0B00000111;         //turn off the comparator,Cxfor numbersIOmouth
}
/*---------------------------------------------- -
* Function name:Delay Us
* Function:    Short delay function --16M-4T--probably fast1%about.
* enter:       TimeDelay time length Delay time lengthTime*2μs none
* output:
  ----------------------------------------------
-------- * / void DelayUs(unsigned char Time)
{
    unsigned char a;
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}
/*---------------------------------------------- -
* Function name:IIC_Start
* Function:    produceIICstart signal
* enter:       none
* output:      none
  ----------------------------------------------------
----- */ void IIC_Start(void)
{
    SDA_OUT;                     //SDAline output
    IIC_SDA=1;
    IIC_SCL=1;
    DelayUs(10);
    IIC_SDA=0;                   //Start: whenSCLWhen high, theSDAfrom high to low
    DelayUs(10);
    IIC_SCL=0;                   //viseI2Cbus, ready to send or receive data
    DelayUs(10);
```

```
}
/*---------------------------------------------- -
* Function name:IIC_Stop
* Function:    produceIICstop signal
* enter:    none
* output:    none
  ------------------------------------------------
  ----- */ void IIC_Stop(void)
{
    SDA_OUT;                    //SDAline output
    IIC_SCL=0;
    IIC_SDA=0;                  //terminated: whenSCLWhen high, theSDAfrom low to high
    DelayUs(10);
    IIC_SCL=1;
    DelayUs(10);
    IIC_SDA=1;                  //sendI2Cbus end signal
    DelayUs(10);
}
/*---------------------------------------------- -
* Function name:IIC_Wait_Ack
* Function:    Wait for the answer signal to come
* enter:    none
* output:    return value:1, failed to receive the response
*                          0, the response is received successfully
  ------------------------------------------------
--------- */ unsigned char IIC_Wait_Ack(void)
{
    unsigned char ucErrTime=0;
    SDA_IN;                     //SDAset as input
    IIC_SDA=1;
    DelayUs(5);
    IIC_SCL=1;
    DelayUs(5);
    while(IIC_SDA)
    {
        ucErrTime++;
        if(ucErrTime>250)       //wait timeout
        {
            IIC_Stop();
            return 1;
        }
    }
    IIC_SCL=0;                  //clock output0
    return 0;
```

```
}
/*------------------------------------------------ -
* Function name:IIC_Ack
* Function:    produceACKanswer
* enter:    none
* output:    none
  ------------------------------------------------
----- */ void IIC_Ack(void)
{
    IIC_SCL=0;
    SDA_OUT;                                //SDAline output
    IIC_SDA=0;
    DelayUs(5);
    IIC_SCL=1;
    DelayUs(5);
    IIC_SCL=0;
}
/*------------------------------------------------ -
* Function name:IIC_NAck
* Function:    does not produceACKanswer
* enter:    none
* output:    none
  ------------------------------------------------
----- */ void IIC_NAck(void)
{
    IIC_SCL=0;
    SDA_OUT;                                //SDAline output
    IIC_SDA=1;
    DelayUs(5);
    IIC_SCL=1;
    DelayUs(5);
    IIC_SCL=0;
}
/*------------------------------------------------ -
* Function name:IIC_Send_Byte
* Function:    IICsend a byte
* enter:    Write a single byte of data to sendtxd
* output:    none
  ------------------------------------------------
--------- */ void IIC_Send_Byte(unsigned char txd) {

    unsigned char t;
    SDA_OUT;                        //SDAline output
    IIC_SCL=0;                      //Pull clock low to start data transfer
```

```
    for(t=0;t<8;t++)
    {
        if((txd&0x80)>>7)
            IIC_SDA=1;
        else
            IIC_SDA=0;
        txd<<=1;
        DelayUs(5);
        IIC_SCL=1;
        DelayUs(5);
        IIC_SCL=0;
        DelayUs(5);
    }
}
/*---------------------------------------------- -
* Function name:IIC_Read_Byte
* Function:   IICread a byte
* enter:    none
* output:   Read the data in the memory and returnreceive
 ----------------------------------------------
--------- */ unsigned char IIC_Read_Byte(void)
{
    unsigned char i,receive=0;
    SDA_IN;                         //SDAset as input
    for(i=0;i<8;i++ )
    {
        IIC_SCL=0;
        DelayUs(5);
        IIC_SCL=1;
        receive<<=1;
        if(IIC_SDA)receive++;
        DelayUs(5);
    }
    IIC_NAck();                     //sendnACK

    return receive;
}
/*---------------------------------------------- -
* Function name:IIC_READ
* Function:   IICRead the data at the specified location
* enter:    address
* output:   read outaddressdata in memoryiicdata
 ----------------------------------------------------
- */ unsigned char IIC_READ(unsigned char address)
```

```
{
    unsigned char iicdata = 0;
    IIC_READ_Begin:
    IIC_Start();
    IIC_Send_Byte(0xa0);

    if(IIC_Wait_Ack())goto IIC_READ_Begin;
    IIC_Send_Byte(address);                 //Fill in the data address to be read

    if(IIC_Wait_Ack())goto IIC_READ_Begin;
    IIC_Start();
    IIC_Send_Byte(0xa1);

    if(IIC_Wait_Ack())goto IIC_READ_Begin;
    iicdata=IIC_Read_Byte();
    IIC_Stop();

    return iicdata;
}
 /*---------------------------------------------- -
* Function name:IIC_WRITE
* Function:    IICput the datadatawrite to specified location
* enter:    address address,data
* output:    none
 ------------------------------------------------------------*/
void IIC_WRITE(unsigned char address, unsigned char data) {

    IIC_WRITE_Begin:
    IIC_Start();
    IIC_Send_Byte(0xa0);
    if(IIC_Wait_Ack())goto IIC_WRITE_Begin;

    IIC_Send_Byte(address);
    if(IIC_Wait_Ack())goto IIC_WRITE_Begin;

    IIC_Send_Byte(data);
    if(IIC_Wait_Ack())goto IIC_WRITE_Begin;

    IIC_Stop();
}
/*---------------------------------------------- -
* Function name:main
* Function:    main function
* enter:    none
```

```
* output: none
  ------------------------------------------------------------
*/ void main()
{
    POWER_INITIAL();                    //system initialization
    IICReadData = IIC_READ(0x12); //read0x12addressEEPROMvalue
    IIC_WRITE(0x13,~IICReadData); //Negate the write address0x13 while(1)

    {
        NOP();
    }
}
```

contact information

Fremont Micro Devices Corporation

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

Tel: (+86 755) 8611 7811
Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Limited

16, 16/F, Block B, Veristong Industrial Centre, 34−36 Au Pui
Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186
Fax: (+852) 2781 1144

http://www.fremontmicro.com