FT62F08X

Application note

目录

FT62F08x EEPROM 应用

1.1. 与 EEPROM 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
INTCON	0x0B	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
EECON1	0x195	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	0000 x000
EECON2	0x196	EEPROM 写控制寄存器 2								---- ----
EECON3	0x198	—							DRDEN	---- ---0
EEADRH	0x192	—	EEPROM 地址高 7 位							-000 0000
EEADRL	0x191	EEPROM 地址低 8 位								0000 0000
EEDATL	0x193	EEPROM 数据低 8 位								xxxx xxxx
EEDATH	0x194	—	—	EEPROM 数据高 6 位						--xx xxxx

1.1.1. EEDAT 寄存器，地址 0x193, 0x194

EEDATL, SFR 地址 0x193

Bit	7	6	5	4	3	2	1	0
Name	EEDAT[7:0]							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

EEDATH, SFR 地址 0x194

Bit	7	6	5	4	3	2	1	0
Name	—	—	EEDAT[13:8]					
Reset	—	—	x	x	x	x	x	x
Type	RO-0	RO-0	RW	RW	RW	RW	RW	RW

Bit	Name	Function
15:14	N/A	保留位，读 0
13:0	EEDAT	EEPROM/FLASH 读写数据寄存器 在写周期（约 2ms）内，该寄存器不可写

1.1.2. EEADR 寄存器，地址 0x191, 0x192

EEADRL, SFR 地址 0x191

Bit	7	6	5	4	3	2	1	0
Name	EEADR[7:0]							

Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

EEADRH, SFR 地址 0x192

Bit	7	6	5	4	3	2	1	0
Name	—	EEADR[14:8]						
Reset	—	0	0	0	0	0	0	0
Type	RO-0	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
15	N/A	保留位，读 0
14:0	EEADR	EEPROM/FLASH 读写地址寄存器 在写周期（约 2ms）内，该寄存器不可写 **注意：用该寄存器访问程序存储器时，地址范围必须位于 0~0x1FFF，否则无法完成读写访问**

1.1.3. EECON1 寄存器，地址 0x195

Bit	7	6	5	4	3	2	1	0
Name	EEPGD	CFGS	—	FERAE	WRERR	WREN	WR	RD
Reset	0	0	0	0	x	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW-1	RW-1

Bit	Name	Function
7	EEPGD	FLASH/数据 EEPROM 存储器选择位 1 = 访问 FLASH 0 = 访问数据 EEPROM 存储器
6	CFGS	FLASH/数据 EEPROM 或配置寄存器选择位 1 = 访问配置寄存器，读访问 0 = 访问 FLASH 或数据 EEPROM 存储器
5	Reserved	**保留位，不要向该位写 1**
4	FERAE	FLASH 擦除使能位 当 CFGS=0 且 EEPGD=1（FLASH）： 1 = 在下一条 WR 命令执行擦除操作（擦除完成后由硬件清零） 0 = 在下一条 WR 命令执行写操作 如果 EEPGD=0 且 CFGS=0（访问数据 EEPROM）： **该位不起作用**，下一条 WR 命令将启动一个擦除周期和一个写周期
3	WRERR	EEPROM 错误标志位 1 = 此状态表示试图进行不当的编程或擦除序列 或在写周期过程中被意外终止时，该位自动置 1 终止的事件可以是除 POR 之外的任何复位

		0 = 编程或擦除操作正常完成
2	WREN	编程/擦除使能位 1 = 允许执行编程/擦除操作 0 = 禁止编程/擦除 FLASH 和数据 EEPROM **在写周期内，禁止对该寄存器位写**
1	WR	FLASH/EEPROM 写控制位 1 = 启动 FLASH 或数据 EEPROM 编程/擦除操作 该操作是自定时的，且该位在操作完成时由硬件清零 **用软件只能将 WR 位置 1，但不能清零** 0 = 对闪存或数据 EEPROM 的编程/擦除操作已完成，当前不在进行中
0	RD	FLASH/数据 EEPROM 读控制位 1 = 启动对 FLASH 或数据 EEPROM 的读操作。读操作只占用一个周期 RD 由硬件清零，用软件只能将 RD 位置 1，但不能清零 0 = 不启动 FLASH 或数据 EEPROM 读操作

1.1.4. EECON2 寄存器，地址 0x196

Bit	7	6	5	4	3	2	1	0
Name	EEPROM 控制寄存器 2							
Reset	x	x	x	x	x	x	x	x
Type	WO	WO	WO	WO	WO	WO	WO	WO

Bit	Name	Function
7:0	EECON2	FLASH/数据 EEPROM 写操作解锁控制寄存器 要解锁写操作，在对 EECON1 寄存器的 WR 置位前，必须先写 55h，随后是 Aah。写入该寄存器的值用于解锁写操作。对这些写操作有特殊的时序要求，必须是在连续的指令周期完成

1.1.5. EECON3 寄存器，地址 0x198

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	DRDEN
Reset	—	—	—	—	—	—	—	0
Type	RO-0	RO-0	RO-0	RO-0	RO-0	RO-0	RO-0	RW

Bit	Name	Function
7:1	N/A	保留位，读 0
0	DRDEN	数据 EEPROM 读使能 1：允许软件读数据 EEPROM(DROM)，置 1 后至少要等 0.2us 才能发起 DROM 读 0：禁止软件读数据 EEPROM(DROM)

2 应用范例

```
//*************************************************************
/*   文件名：TEST_62F08x_EEPROM.c
*    功能：    FT62F08x-EEPROM 功能演示
*    IC:      FT62F088 LQFP32
*    内部：     16M
*    empno: 500
*    说明：          此演示程序为 62F08x_EEPROM 的演示程序.
*              把 0x55 写入地址 0x13,再读出该值

*
*    参考原理图  TEST_62F08x_sch.pdf
*/

//*************************************************************
#include "SYSCFG.h"
//*************************************************************
//宏定义*******************************************************
#define   unchar          unsigned char
#define   unint           unsigned int
#define   unlong          unsigned long

volatile   unchar EEReadData;

/*-----------------------------------------------
 *   函数名：  interrupt ISR
 *   功能：    中断处理，包括定时器 0 中断和外部中断
 *   输入：    无
 *   输出：    无
 -----------------------------------------------*/
void interrupt ISR(void)                    //PIC_HI-TECH 使用
{

}
/*-----------------------------------------------
 *   函数名：POWER_INITIAL
 *   功能：    上电系统初始化
 *   输入：    无
 *   输出：    无
 -----------------------------------------------*/
void POWER_INITIAL (void)
{
```

```
    OSCCON = 0B01110001;     //WDT 32KHZ IRCF=111=16MHZ/2
                             //Bit0=1,系统时钟为内部振荡器
                             //Bit0=0,时钟源由 FOSC<2：0>决定即编译选项时选择
    INTCON = 0;              //暂禁止所有中断

    PORTA = 0B00000000;
    TRISA = 0B11111111;      //PA 输入输出  0-输出  1-输入
    PORTB = 0B00000000;
    TRISB = 0B11110111;      //PB 输入输出  0-输出  1-输入
    PORTC = 0B00000000;
    TRISC = 0B11111111;      //PC 输入输出  0-输出  1-输入
    PORTD = 0B00000000;
    TRISD = 0B11111111;      //PD 输入输出  0-输出  1-输入

    WPUA = 0B00000000;       //PA 端口上拉控制  1-开上拉  0-关上拉
    WPUB = 0B00000000;       //PB 端口上拉控制  1-开上拉  0-关上拉
    WPUC = 0B00001000;       //PC 端口上拉控制  1-开上拉  0-关上拉
    WPUD = 0B00000000;       //PD 端口上拉控制  1-开上拉  0-关上拉

    WPDA = 0B00000000;       //PA 端口上拉控制  1-开下拉  0-关下拉
    WPDB = 0B00000000;       //PB 端口上拉控制  1-开下拉  0-关下拉
    WPDC = 0B00000000;       //PC 端口上拉控制  1-开下拉  0-关下拉
    WPDD = 0B00000000;       //PD 端口上拉控制  1-开下拉  0-关下拉

    PSRC0  = 0B11111111;     //PORTA,PORTB 源电流设置最大
    //BIT7~BIT6:PORTB[7:4]源电流能力控制,BIT5~BIT4:PORTB[3:0]源电流能力控制
    //BIT3~BIT2:PORTA[7:4]源电流能力控制,BIT1~BIT0:PORTA[3:0]源电流能力控制

    PSRC1  = 0B11111111;     //PORTC,PORTD 源电流设置最大
    //BIT7~BIT6:PORTD[7:4]源电流能力控制,BIT5~BIT4:PORTD[3:0]源电流能力控制
    //BIT3~BIT2:PORTC[7:4]源电流能力控制,BIT1~BIT0:PORTC[3:0]源电流能力控制

    PSINK0 = 0B11111111;     //PORTA 灌电流设置最大  0:最小，1:最大
    PSINK1 = 0B11111111;     //PORTB 灌电流设置最大  0:最小，1:最大
    PSINK2 = 0B11111111; //PORTC 灌电流设置最大  0:最小，1:最大
    PSINK3 = 0B11111111; //PORTD 灌电流设置最大  0:最小，1:最大

    ANSELA = 0B00000000;     //全为数字管脚
}
/*---------------------------------------------------
 *  函数名称：EEPROMread
 *  功能：     读 EEPROM 数据
 *  输入参数：EEAddr  需读取数据的地址
 *  返回参数；ReEEPROMread  对应地址读出的数据
```

```
 ----------------------------------------------------*/
unchar EEPROMread(unchar EEAddr)
{
    unchar ReEEPROMread;

    DRDEN=1;
    NOP();
    NOP();
    EEADRL = EEAddr;

    CFGS =0;
    EEPGD=0;
    RD = 1;

    ReEEPROMread = EEDATL;
    DRDEN=0;
    return ReEEPROMread;
}
/*----------------------------------------------------
 *  函数名称：EEPROMwrite
 *  功能：     写数据到 EEPROM
 *  输入参数：EEAddr  需要写入数据的地址
 *            Data  需要写入的数据
 *  返回参数：无
 ----------------------------------------------------*/
void EEPROMwrite(unchar EEAddr,unchar Data)
{
    GIE = 0;                    //写数据必须关闭中断
    while(GIE);                 //等待 GIE 为 0
    EEADRL = EEAddr;                //EEPROM 的地址
    EEDATL = Data;                  //EEPROM 的写数据

    CFGS =0;                        //访问 EEPROM 存储器
    EEPGD=0;                         //

    WREN=1;                          //写使能

    EEIF = 0;
    EECON2=0x55;
    EECON2=0xAA;

    WR = 1;                         //置位 WR 启动编程
    while(WR);                  //等待 EE 写入完成
    WREN=0;
```

```
    GIE = 1;
}

/*---------------------------------------------
 *  函数名:  main
 *  功能:    主函数
 *  输入:    无
 *  输出:    无
 ---------------------------------------------*/
void main(void)
{
    POWER_INITIAL();                    //系统初始化
    EEPROMwrite(0xff,0xaa);
    EEPROMwrite(0xff,0xaa);             //在未使用到的随意一个地址写两次 0xaa

    EEPROMwrite(0x13,0x55);         //0x55 写入地址 0x13

    EEReadData = EEPROMread(0x13);   //读取 0x13 地址 EEPROM 值

    while(1)
    {
        NOP();
    }
}
```

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/