FT61F02X

EEPROM Application note

Table of contents

FT61F02x EEPROMapplication

1.dataEEPROM (DATA EEPROM)

FT61F02xOn-chip is integrated with256 x 8-bitnon-volatileDATA EEPROMstorage area and is independent of the main program area. Typical erase and write cycles for this data storage area are up to100million times. Read/write access can be performed by instructions, and the unit that can be read or written each time is1indivualbyte (8−bit), without page mode (page mode). Erase/program is hardware self-timed without software polling to save limited code space. Therefore, write operations can run in the background without affectingCPUExecute other commands, even enterSLEEPstate.

read operation requires2instruction clock cycle, and the time required for the write operation is$T_{WRITE-EEPROM}$(Enable Auto Erase as2 ~ 4 ms , and auto-erase off for0.7 ~ 1.3ms). The chip has a built-in charge pump, so there is no need to provide an external high voltage toEEPROMarea for erasing and programming. The corresponding interrupt flag bit will be set when the write operation is completeEEIF.

Sequential reads are not supported (sequential READ)or sequential writes (sequential WRITE), so each read/write must update the corresponding address.

if only$V_{DD} \geq V_{POR}$, CPUavailable at8 MHz / 2Toperating at high speeds, even at high temperatures as low as1.5Vabout. while writing DATA EEPROMThe required voltage ($V_{DD-WRITE}$)higher Industrial and Automotive1lowest grade$V_{DD-WRITE}$respectively1.9Vand2.2V. readDATA EEPROMThere is no such minimum voltage limit (see$V_{DD-READ}$).

1.1. DATA EEPROMSummary of related registers

name	state	register	address	reset value
EEDAT	DATA EEPROMdata	EEDAT[7:0]	0x9A	RW−0000 0000
EEADR	DATA EEPROMaddress	EEADR[7:0]	0x9B	RW−0000 0000
WREN3	DATA EEPROMwrite enable (bit 3) 111 =Enable, reset to000 = (other) closure	EECON1[5]	0x9C	RW−0
WREN2	DATA EEPROMwrite enable (bit 2)	EECON1[4]		RW−0
WRERR	DATA EEPROMwrite error flag 1 =Abort (happensMCLRorWDTreset) 0 = Completed normally	EECON1[3]		RW−x
WREN1	DATA EEPROMwrite enable (bit 1)	EECON1[2]		RW−0
RD	DATA EEPROMread control bit 1 = Yes (Keep4indivualSysClkperiod, then =0) 0 = No	EECON1[0]		RW−0
WR	DATA EEPROMwrite control bit 1 =Initiates a write or a write is in progress (resets to0) 0 = Finish	EECON2[0]	0x9D	RW−0

surface1-1 EEPROMRelated User Control Registers

name	state		register	address	reset value
GIE	global interrupt	1 =Enable (PEIE, EEIEBe applicable) 0 =global shutdown (wake up is not affected)	INTCON[7]	0x0B 0x8B 0x10B	RW−0
PEIE	Total Peripheral Interrupt	1 =Enable (EEIEBe applicable) 0 =closure (no wakeup)	INTCON[6]		RW−0
EEIE	EEPROMwrite complete interrupt	1 =Enable 0 =closure (no wakeup)	PIE1[7]	0x8C	RW−0
EEIF	EEPROMwrite complete interrupt flag bit	1 = Yes (latch) 0 =no	PIR1[7]	0x0C	RW−0

surface1-2 EEPROMInterrupt Enable and Status Bits

1.2. initialization

dataEEPROMThe following initialization operations must be performed before use (whether it is reading or writing):EEPROMA cell is written twice 0xAA, subsequent programs do not operate on this unit. like:

```
SYSTEM_INIT:
    ...
    ...
    LDWI 0x55
    STR EEPROM_ADDR
    LDWI 0xAA
    STR EEPROM_DATA
    LCALL EEPROM_write
    LCALL EEPROM_write
    ...
```

1.3. WriteDATA EEPROM

1. set up"GIE = 0";

2. judgeGIE,if"GIE = 1",then repeat steps (1);

3. PastEEADRwrite target address;

4. PastEEDATwrite target data;

5. set up"WREN3, WREN2, WREN1" = "1, 1, 1", and maintain this setting throughout the programming process;

6. must be set immediately"WR = 1"to start writing (otherwise it will abort);

7. Programming complete (see programming time$T_{WRITE-EEPROM}$)back,"WR"and"WREN3, WREN2, WREN1"will be automatically cleared 0;

Sample program:

```
BCR INTCON, GIE
BTSC INTCON, GIE
LJUMP $-2
BANKSEL EEADR
LDWI 55H
STR EEADR          ;address is0x55
STR EEDAT          ;The data is0x55
LDWI 34H
STR EECON1         ; WREN3/2/1at the same time1 ;
BSR EECON2, 0      start write
BSR INTCON, GIE    ;GIEplace1
```

Note:

1.While programming is in progress, theData EEPROMPerforming a read operation will result in an incorrect read result.

2.If programming is complete before,WREN3, WREN2orWREN1Any one is cleared0, Need to be cleared before next programmingEEIFFlag bit.

1.4. readDATA EEPROM

write target addressEEADRregister, then initiate a read("RD = 1").2After instruction clock cycles, theEEPROMdata is written EEDAT register, so the read instruction must be followed by aNOPinstruction.EEDATThe register will hold this value until the next read or write operation.

readDATA EEPROMA sample program is as follows:

```
BANKSEL EEADR
LDWI dest_addr
STR EEADR
BSR EECON1, RD
NOP                ;read wait
LDR EEDAT, W       ;At this point, data can be read by the instruction
```

1.5. Auto Erase Function

Write data to bytes (byte)The process includes2Step: Erase the byte first, then program the byte. The erase operation will byte allbitsrubbed into "1", while the programming operation will selectively place individualbitswritten as "0"he chip has built-in automatic erase function, That is, the erase operation will be performed automatically before programming.

multiple programmingFFThe data actually erases the corresponding byte multiple times. However, multiple programming is notFFThe data is actually programmed only once for the corresponding byte, since each programming is preceded by an automatic erase.

2.Application example

```
/*file name:TEST_61F02x_EE.C
* Function:       FT61F02x-internalEEDemo
*IC:              FT61F023 SOP16
* Crystal:       16M /2T
* illustrate:     This demo program is61F02x EERROMA demo program for . The program reads
*                 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|---------NC
* NC----------|5(PC3)          (PA3)12|---------NC
* NC----------|6(PC2)          (PC0)11|---------NC
* NC----------|7(PA4)          (PC1)10|---------NC
* NC----------|8(PC5)          (PC4)09|---------NC
*                   ------------------------------
*/
//************************************************** ***********
# include "SYSCFG.h"
//************************Macro definition ****************************
#define unchar          unsigned char

unchar EEReadData;
/*---------------------------------------------- -
* Function name:POWER_INITIAL
* Function:    Power-on system initialization
* enter:    none
* output:   none
 ------------------------------------------------
--------- */ void POWER_INITIAL (void)
{
    OSCCON = 0B01110001;                    //IRCF=111=16MHz/2=8MHz,0.125μs//Temporarily
    INTCON = 0;                             disable all interrupts
    PORTA = 0B00000000;
    TRISA = 0B00000000;                     //PAall output
    PORTC = 0B00000000;
    TRISC = 0B00000000;                     //PCall output
    WPUA = 0;                               //ban allPApull up
    WPUC = 0;                               //ban allPCpull up

    OPTION = 0B00001000;              //Bit3=1 WDT MODE, PS=000=1:1 WDT RATE
```

```
    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:EEPROM read
* Function:   readEEPROMdata
* enter:      EE AddrThe address of the data to be read
* output:     ReEEPROM readThe data read from the corresponding address
 ------------------------------------------------
--------- * / unchar EEPROM read(unchar EEAddr)
{
    unchar ReEEPROM read;

    EEADR = EEAddr;
    RD = 1;
    ReEEPROMread = EEDAT;          //EEPROMread dataReEEPROMread = EEDATA;
    return ReEEPROMread;
}

/*---------------------------------------------- ----
* Function name:EEPROMwrite
* Function:   write data toEEPROM
* enter:      EE AddrThe address where data needs to be written
*             Datadata to be written
* output:     none
-----------------------------------------------------------
--- * / void EEPROMwrite(unchar EEAddr,unchar Data) {

    GIE =  0;                      //Write data must turn off the interrupt
    while(GIE);                    //waitGIEfor0
    EEADR = EEAddr;                //EEPROMthe address of
    EEDAT = Data;                  //EEPROMwrite data     EEDATA = Data;
    EEIF = 0;
    EECON1 = 0x34;                 //PositionWREN1, WREN2, WREN3three variables. //Position
    WR = 1;                        WRstart programming
    while(WR);                     //waitEEwrite complete
    GIE =  1;
}

/*---------------------------------------------- -
* Function name:main
```

```
* Function:    main function
* enter:    none
* output:    none
  ------------------------------------------------------------
*/ void main()
{
    POWER_INITIAL();                    //system initialization
    EEPROMwrite(0xff,0xaa);
    EEPROMwrite(0xff,0xaa);             //Write twice at any unused address0xAA

    EEReadData = EEPROMread(0x12);      //read0x12addressEEPROMvalue //Negate
    EEPROMwrite(0x13,~EEReadData);      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