

# HT67Fx0 Data EEPROM 的使用方法与注意事项

文件编码: HA0254S

## 简介

HT67Fx0 系列都有内部 EEPROM Data Memory 功能, EEPROM 数据寄存器容量为 64×8~256×8, 用户可以根据不同应用需求选择合适 EEPROM 大小的 MCU。由于映射方式与程序存储器和数据存储器不同, 因此不能像其它类型的存储器一样寻址。使用 Bank 0 中的一个地址和数据寄存器以及 Bank 1 中的一个控制寄存器, 可以实现 EEPROM 的字节读写操作。本文以 HT67F50 为母体, 说明 EEPROM Data Memory 的使用方法。

## 工作原理

HT67Fx0 系列有三个寄存器控制内部 EEPROM 数据存储器的操作。地址寄存器 EEA、数据寄存器 EED 及控制寄存器 EEC。其中 EEA 与 EED 位于 BANK0 中, 它们能像其它特殊功能寄存器一样直接被访问。而 EEC 位于 BANK1 中, 不能被直接访问, 仅能通过 MP1 和 IAR1 进行间接读取或写入。由于 EEC 控制寄存器位于 Bank 1 中的[40H], 在 EEC 寄存器上的任何操作被执行前, MP1 必须先设为[40H], BP 被设为[01H]。以下为 EEPROM 读取操作或写入操作的步骤:

- 设定所需要读取/写入数据的地址值 (EEA 寄存器的值)。
- 设定所需要写入数据的值 (EED 寄存器的值)。
- 设定 BP.0 位, 选中数据存储区 BANK1。
- 设定 MP1 为[40H], 指针指向 EEPROM 的控制寄存器 EEC。
- 使能读取/写入模式 (设定 RDEN/WREN 位)。
- 开始读取/写入周期 (设定 RD/WR 位)。
- 等待读取/写入周期结束。
- 读取/写入周期结束后, 系统会自动清 RD 或 WR 为“0”, 如果是写入周期结束, 这时其中断请求标志位 DEF 会设定为“1” (HT67F50 目前读取操作没有中断)。数据在其它读取或写入操作执行前将一直保留在 EED 寄存器中。应用程序可以通过查询 RD 或 WR 位以确定数据是否有效地被读取或写入。

## 程序说明

以下做了两个范例以详细说明 EEPROM 的相关读取/写入操作,其中范例 1 是以查寻相关读取/写入标志位的方式完成对 EEPROM 的读取/写入,主要说明如何设定地址寄存器 EEA、数据寄存器 EED 和控制寄存器 EEC 来实现 EEPROM 的读取/写入;范例 2 主要说明如何设定 EEPROM 相关中断寄存器来实现写入数据周期结束后进入中断服务程序,以中断的方式完成对 EEPROM 的写入。

### 范例<1>: 以查寻标志位的方式进行读取/写入EEPROM

; 程序名: EEPROM 读取/写入 (eeprom\_rw.asm)  
 ; 目的: 将数据写入 EEPROM 一个字节后再读取出来相比较  
 ; EEPROM 中断: Disable  
 ; Option: WDT disable  
 ; 程序清单:

```

INCLUDE HT67F50.INC
;-----
DATA .SECTION 'DATA'
EEPROM_ADRES DB ? ;user defined address
EEPROM_DATA DB ? ;user defined data
;-----
CODE .SECTION 'CODE'
    ORG 00H
    JMP MAIN
    ORG 04H ; External Interrupt 0 entry address
    RETI
    ORG 08H ; External Interrupt 1 entry address
    RETI
    ORG 0CH ; Comparator 0 Interrupt entry address
    RETI
    ORG 010H ; Comparator 1 Interrupt entry address
    RETI
    ORG 014H ; Multi-function Interrupt 0 entry address
    RETI
    ORG 018H ; Multi-function Interrupt 1 entry address
    RETI
    ORG 01CH ; A/D Converter Interrupt entry address
    RETI
    ORG 020H ; Multi-function Interrupt 2 entry address
    RETI
    ORG 024H ; Time base 0 Interrupt entry address
    RETI
    ORG 028H ; Time base 1 Interrupt entry address
    RETI
    ORG 02CH ; Multi-function Interrupt 3 entry address RETI
;-----
    ORG 30H
MAIN:
    CLR PAC ; PA set as output mode
    SET PAPU
    MOV A,0AH
    MOV WDTC,A ; close WDT
    CLR EMI ; Close Global interrupt
    SET HLCLK ; System clock selection for HXT
    CLR ACERL
    CLR CPOC
    
```

```

        CLR CPIC                ; Set as the I/O mode
;-----
;EEPROM WRITE OPERATION
;-----
WRITE_EEPROM:
    MOV A,10H
    MOV EEPROM_ADRES,A        ; Set write data address
    MOV A,055H
    MOV EEPROM_DATA,A        ; Set write data value
;-----
    MO  A,EEPROM_ADRES
    MOV EEA,A
    MOV A,EEPROM_DATA
    MOV EED,A
    SET BP,0                  ; setup Bank Pointer
    MOV A,40H                 ; setup memory pointer MP1
    MOV MP1,A                 ; MP1 points to EEC register
    SET IAR1.3                ; set WREN bit, enable write operations
    SET IAR1.2                ; start Write Cycle - set WR bit
WRITE_WAIT:
    SZ  IAR1.2                ; check for write cycle end
    JMP WRITE_WAIT
;-----
    CLR IAR1                  ; disable EEPROM read/write
    CLR BP
;-----
;EEPROM READ OPERATION
;-----
READ_EEPROM:
    MOV A,10H
    MOV EEPROM_ADRES,A        ; Set write data address
;-----
    MOV A,EEPROM_ADRES
    MOV EEA,A
    SET BP,0                  ; setup Bank Pointer
    MOV A,40H                 ; setup memory pointer MP1
    MOV MP1,A                 ; MP1 points to EEC register
    SET IAR1.1                ; set RDEN bit, enable read operations
    SET IAR1.0                ; start Read Cycle - set RD bit
READ_WAIT:
    SZ  IAR1.0                ; check for read cycle end
    JMP READ_WAIT
    MOV A,EED                  ; move read data to register
;-----
    XOR A,EEPROM_DATA        ; Read out value compared to the value written
    SZ  ACC                    ; Determine the consistency of read data and write data
    JMP FAIL
    MOV A,01H
    XORM A,PA                  ; If the same then PA0 output waveform
    JMP WRITE_EEPROM
;-----
FAIL:
    MOV A,02H
    XORM A,PA                  ; If the inconsistency then PA1 output waveform
    JMP FAIL

```

### 程序说明

本范例主要介绍如何通过查询标志，以及正确设定相关寄存器来实现对 EEPROM 的读取/写入操作，在范例上只是简单的循环写入和读取 EEPROM 的一个字节的内容，其执行过程是对 EEPROM 的[10H]写入 55H 后，再从 EEPROM 的[10H]读取出来与写入的值相比较，如果正确即 PA0 输出波形，反之 PA1 输出波形。通过示波器观察 2 个 I/O 口的输出就可以判断对 EEPROM 的读取/写入操作是否正常。

用户使用时，只要调用下面读取/写入操作的子程序模块（完成一个指定地址的读取和写入），即可完成单字节的读取/写入操作，如需要读取/写入多个字节的内容时，只需在调用此子程序的同时修改所需要写入和读取时的地址变量 EEPROM\_ADRES 和数据变量 EEPROM\_DATA 就可以进行多字节操作。读取/写入操作子程序如下：

读取操作模块：

```

READ_EEPROM_LOOP:
    MOV A,EEPROM_ADRES
    MOV EEA,A
    SET  BP,0           ;setup Bank Pointer
    MOV A,40H          ;setup memory pointer MP1
    MOV MP1,A          ;MP1 points to EEC register
    SET  IAR1.1        ;set RDEN bit, enable read operations
    SET  IAR1.0        ;start Read Cycle - set RD bit
READ_WAIT:
    SZ   IAR1.0        ;check for read cycle end
    JMP  READ_WAIT
    CLR  IAR            ;disable EEPROM read/write
    CLR  BP
    RET
    
```

写操作模块：

```

WRITE_EEPROM_LOOP:
    MOV A,EEPROM_ADRES
    MOV EEA,A
    MOV A,EEPROM_DATA
    MOV EED,A
    SET  BP,0           ;setup Bank Pointer
    MOV A,40H          ;setup memory pointer MP1
    MOV MP1,A          ;MP1 points to EEC register
    SET  IAR1.3        ;set WREN bit, enable write operations
    SET  IAR1.2        ;start Write Cycle - set WR bit
WRITE_WAIT:
    SZ   IAR1.2        ;check for write cycle end
    JMP  WRITE_WAIT
    CLR  IAR1          ;disable EEPROM read/write
    CLR  BP
    RET
    
```

**范例< 2>: 以中断的方式完成对EEPROM的写入操作**

- ; 程序名: EEPROM 写入模式中断 (eeprom\_r\_int.asm)
- ; 目的: 完成对 EEPROM 写入操作后进入中断服务程序。
- ; EEPROM 中断: Enable
- ; Option: WDT Disable
- ; 程序清单:

```

INCLUDE HT67F50.INC
;-----
DATA .SECTION 'DATA'
EEPROM_ADRES DB ? ;user defined address
EEPROM_DATA DB ? ;user defined data
EEPROM_INT_FLAG DBIT ;Flag of interrupt
;-----
CODE .SECTION 'CODE'
ORG 00H
JMP MAIN
ORG 04H ;External interrupt 0 entry address
RETI
ORG 08H ;External interrupt 1 entry address
RETI
ORG 0CH ;Comparator 0 Interrupt entry address
RETI
ORG 010H ;Comparator 1 Interrupt entry address
RETI
ORG 014H ;Multi-function Interrupt 0 entry address
RETI
ORG 018H ;Multi-function Interrupt 1 entry address
RETI
ORG 01CH ;A/D Interrupt entry address
RETI
ORG 020H ;Multi-function Interrupt 2 entry address
RETI
ORG 024H ;Time base 0 Interrupt entry address
RETI
ORG ;Time base 1 Interrupt entry address
RETI
ORG 02CH ;Multi-function Interrupt 3 entry address
JMP EEPROM_INTERRUPT
;-----
ORG 30H
MAIN:
CLR PAC ;PA set as output mode
SET PAPU
SET HLCLK ;select system clock : HXT
CLR CMP0C ;set as I/O mode
CLR CMP1C ;set as I/O mode
CLR ACERL ;set as I/O mode
MOV A,01H
MOV INTC0,A ;enable EMI
MOV A,08H
MOV INTC2,A ;Enable Multi-Function Interrupt
MOV A,02H
MOV MF13,A ;Enable EEPROM Interrupt
CLR EEPROM_INT_FLAG ;Initial interrupt flag
;-----
;EEPROM WRITE OPERATION
;-----
WRITE_EEPROM:

```

```

MOV A,20H
MOV EEPROM_ADRES,A      ; Set the address of data to be written
MOV A,055H
MOV EEPROM_DATA,A       ; Set the value of data to be written
;-----
WRITE_EEPROM_LOOP:
MOV A,EEPROM_ADRES      ; user defined address
MOV EEA,A
MOV A,EEPROM_DATA       ; user defined data
MOV EED,A
SET  BP,0                ; setup Bank Pointer
MOV  A,40H               ; setup memory pointer MP1
MOV  MP1,A              ; MP1 points to EEC register
SET  IAR1.3             ; set WREN bit, enable write operations
SET  IAR1.2             ; start Write Cycle - set WR bit
WRITE_WAIT:
SNZ  EEPROM_INT_FLAG    ; check for write cycle end
JMP  WRITE_WAIT
CLR  IAR1                ; disable EEPROM read/write
CLR  BP
CLR  EEPROM_INT_FLAG
JMP  WRITE_EEPROM
;-----
;EEPROM Interrupt Service Route
;-----
ORG 0A0H
EEPROM_INTERRUPT:
SET  EEPROM_INT_FLAG
MOV  A,01H
XORM A,PA                ; PA0 output waveform
CLR  MFI3.5              ; clr DEF
RETI

```

### 程序说明

本范例程序主要说明如何设定 EEPROM 中断有关的寄存器，在对 EEPROM 的一个写入周期结束后产生中断请求，并能进入中断服务程序。与 EEPROM 中断有关的寄存器有三个：总中断使能位 EMI[INTC0.0]、多功能中断使能位 MF3E[INTC2.3]和 EEPROM 中断使能位 DEE [MFI3.1]。要想能够进入 EEPROM 中断，这三个中断使能位必须先设定为“1”。一旦 EEPROM 一个写入周期结束后，中断请求标志 DEF 会设定为“1”，在有关 EEPROM 中断都使能的情况下即进入中断服务程序。在中断产生后，多功能中断请求标志 MF3F 位会自动清除为“0”，而 EEPROM 中断请求标志 DEF 位则不会自动清除为“0”，需要用软件手动清除为“0”。本范例的执行过程是循环对 EEPROM 其中的一个地址[20H]写入 55H 的数据，并在数据写入周期结束后进入中断服务程序，在中断服务程序中设定一个标志，用于判断每写完一次数据后是否有进入中断服务程序，若有进入中断则 PA0 有波形输出。

## 注意事项

要执行对 EEPROM 的读取/写入操作时,必须按照先将 RDEN/WREN 设定为高,再将 RD/WR 设定为高的顺序,才能开始读取/写入周期。

在同一条指令中 WREN、WR、RDEN 和 RD 不能同时设定为“1”。WR 和 RD 不能同时设定为“1”。

由于控制寄存器 EEC 位于 BANK1 中,不能被直接访问,仅能通过 MP1 和 IAR1 进行间接读取或写入,在 EEC 寄存器上的任何操作被执行前,BP 必须先被设为[01H],MP1 必须先设为[40H]。

中断产生后,EEPROM 中断请求标志 DEF 位不会自动清除为“0”,这时需要软件来手动清除为“0”,以便下一次的中断响应。

在正常程序操作中确保控制寄存器 EEC 中的写入使能位被清除,这样能防止不正确的写入操作。

## 版本记录

### 版本: V1.10

修改人员: 马灵

修改日期: 2011 年 12 月 20 日

修改内容:

- 文中修改了“EEA、EED、WDTC、MFI3、ACERL、CP0C、CP1C”寄存器及“DEF、DEE、MF3E”位。
- 程序附文件中修改了“EEA、EED、WDTC、MFI3、ACERL、CP0C、CP1C”寄存器及“DEF、DEE、MF3E”位。