



FT61F02x Application Note

[Version 0.05]

milk.chen

1. 文档更改历史

日期	作者	版本	内容
2015-7-15	Milk.Chen	0.01	初版
2017-8-21	Milk.Chen	0.04	页脚加入日期 添加“寄存器 ADRESH/L 读写问题” 添加“EECPAS 寄存器定义” 添加“关于 ADC 的采样时间要求”
2018-2-5	Milk.Chen	0.05	修改“关于 ADC 的采样时间要求”中注意事项

2. 详细说明

2.1 复位时 IO 处于高阻态

- 适用芯片型号:

所有版本

- 问题描述:

任何复位后, GPIO 都处于输入状态且内部上拉关闭, 外部电路看到的 IO 为高阻态。

- 解决方法:

应用方案应该要评估该特性会不会对电路其它部分造成影响, 如果需要可在关键 IO 外挂上拉或者下拉。

2.2 PA 端口 (电平) 变化中断标志位的清除

- 适用芯片型号:

所有版本

- 问题描述:

端口变化中断的清除在数据手册的 14.2 节提到:

a) 读取 PORTA;

b) 清除 PAIF;

软件开发需要用到端口 (电平) 变化中断的, 清除 PAIF 标志位必须按照以上步骤执行。

其中 a) 步骤是清除不匹配条件, 因为不匹配条件一直存在, PAIF 就不能被软件清除。

- 解决方法:

另外, 初始化端口变化中断时建议按以下顺序操作:

a) 设置 TRISA, 把相关 PA 口设置为输入口;

b) 读 PORTA;

c) 清 PAIF;

d) 设置 IOCA 寄存器;

e) 设置 RAIE 位;

2.3 Data EEPROM 编程

2.3.1 编程步骤

- A. 把 INTCON 的 GIE 位清 0;
- B. 判断 GIE 是否为 1, 是则重复 A 步骤, 否则可以进行下一步;
- C. 往 EEADR 写入目标地址;
- D. 往 EEDAT 写入目标数据;
- E. 把位 WREN3/WREN2/WREN1 全部置 1;

-
- F. 把位 WR 置 1 (EECON2.0, 此后 WR 会维持高);
 - G. 写过程不能改变 WREN3/2/1 的值, 否则编程终止;
 - H. 等大概 2ms 之后编程自动完成, WR 自动清 0, WREN3、WREN2、WREN1 清 0;

注意:

- a) 以上步骤的 E、F 两步必须是连续的两条指令周期完成, 不能错开, 否则编程操作不会启动;
- b) F 步骤之后, 根据应用需要, 可以打开 GIE 全局中断使能位。

2.3.2 编程建议

在使用 EEPROM 之前需要对 EEPROM 进行初始化操作, 在未使用到的 EEPROM 地址写入两次“0xAA”, 后续程序不要操作到此地址数据。如:

SYSTEM_INI:

```
.....
.....
LDWI  0x55
STR   EEPROM_ADDR
LDWI  0xAA
STR   EEPROM_DATA
LCALL EEPROM_Write    ;往 0x55 地址写入数据 0xAA
LCALL EEPROM_Write    ;写两次
.....
.....
RET
```

2.3.3 检查 WRERR 位

如果在 EEPROM 写过程中发生了外部复位、WDT 溢出复位、LVR 复位或者非法指令复位, 标志位 EECON1.WRERR 会被置 1。利用它, 软件可以知道在前一次 EEPROM 写过程中有没有发生异常情况, 进而采取相应处理措施。

2.3.4 检查写完成

EEPROM 编程时间大概为 2ms 左右, 这过程可以通过检查 EECON2.WR 或者 PIR1.EEIF 位来确定写结束。

如果应用允许, 还可以启动编程后进入 SLEEP 状态, 然后等待写完成的唤醒, 需要把 EEIE 和 PEIE 位置 1。

2.3.5 加入写校验步骤

根据应用情况, 将写入 EEPROM 的实际值和要写入的目标值做核对是一个很好的编程习惯,

即写完成后，软件读一下 EEPROM 与写目标值对比，相等则说明写成功，否则写失败，软件可采取重写策略。

2.3.6 使能 LVR/LVD 模块

在发生欠压（VDD 低于 MCU 最低工作电压时。最低工作电压跟工作频率相关，如只有在 VDD>2.7V 时，MCU 才能跑 16M，这时建议 LVR 设置 2.8V）而 MCU 没进入复位的状态下，可能会发生偶发性的写入。可通过使能 LVR 或者外部复位电路让器件处于复位，以确保在超出正常工作电压范围时，不发生数据 EEPROM 的写操作。另外，软件还有另外一个选择，LVD。当判断到有低电压标志时，不要执行写 EEPROM 的流程。

2.4 提高内部快时钟稳定性和降低工作电流

应用程序遵守以下建议能提高内部 16M 快时钟的稳定性，同时带来另一个好处是能降低工作电流：

2.4.1 循环体内少用 NOP 指令

如果非要使用 NOP 指令，在应用允许的情况下，执行 NOP 的循环之前把工作寄存器 W 清 0。

2.4.2 跳转指令 LJUMP 避开 0xFF 地址

使用 LJUMP label_xxx 指令时，在程序空间足够且应用允许的情况下，尽量避开 label_xxx 地址低 8 位为全 1 的情况，诸如 0xFF 或 0x1FF 等，可考虑把 label_xxx 地址放在 0x100/0x200，如：

```
LJUMP label_xxx
ORG 0x100
label_xxx:
```

2.5 LVR 选项

跟 FT60F0XX 不一样，FT61F02X 的 LVR 烧录选项有以下几种，在烧录程序应注意它们的区别。

- 使能低电压复位
- LVR 由 MSCKCON 的 SLVREN 决定
- MCU 正常模式时开启 LVR，睡眠模式时关闭 LVR，跟 SLVREN 位无关
- 禁止低电压复位

2.6 16M/2T 下数据 EEPROM 无法编程

- 适用芯片型号：
C 版之前的芯片
- 问题描述：
芯片配置为 2T 模式，且 OSCCON.IRCF 配置为 111 时，即 16MHz/2T，程序无法对 EEPROM 编程。
- 解决方法：
C 版之后（包括 C 版在内），该问题已经得到解决。

对于 C 版之前的芯片，如果应用需要用到 EEPROM 编程，建议不要使用最高速度 16M/2T，可以使用 16M/4T，或者 8M/2T 或者其它任何系统时钟。

2.7 16M/2T 下寄存器 OSCCON 无法改写

- 适用芯片型号：
C 版之前的芯片
- 问题描述：
芯片配置为 2T 模式，且 OSCCON 配置为 111 时，即 16MHz/2T，程序无法改写 OSCCON。
- 解决方法：
C 版之后（包括 C 版在内），该问题已经得到解决。

对于 C 版之前的芯片，如果应用需要动态改写 OSCCON 寄存器，建议不要使用最高速度 16M/2T，可以使用 16M/4T，或者 8M/2T 或者其它任何系统时钟。

2.8 寄存器 ADRESH/L 读写问题

- 适用芯片型号：
C 版之前的芯片
- 问题描述：
软件对 ADRESH/L 不可写。
- 解决方法：
C 版之后（包括 C 版在内），该问题已经得到解决。

对于 C 版之前的芯片，软件只需把这两个寄存器当作只读类型。

2.9 ECCPAS 寄存器定义

- 适用芯片型号：

C 版之前的芯片

- 问题描述:

ECCPAS 寄存器位 PSSAC/PSSBD 描述有改动:

3:2	PSSAC[1:0]	P1A 和 P1C 引脚关闭状态控制位 00 = 驱动引脚 P1A 和 P1C 为 0 01 = 驱动引脚 P1A 和 P1C 为 1 对于 1x 值, 有以下区分: 1x = P1A 和 P1C 引脚为三态 (C 版之后, 包括 C 版) 1x = P1A 和 P1B 引脚为三态 (C 版之前)
1:0	PSSBD[1:0]	P1B 和 P1D 引脚关闭状态控制位 00 = 驱动引脚 P1B 和 P1D 为 0 01 = 驱动引脚 P1B 和 P1D 为 1 对于 1x 值, 有以下区分: 1x = P1B 和 P1D 引脚为三态 (C 版之后, 包括 C 版) 1x = P1C 和 P1D 引脚为三态 (C 版之前)

- 解决方法:

应用程序应注意 C 版之前和 C 版之后的区别。

2.10 关于 PA5 的使用

- 适用芯片型号:

G 版和 H 版

- 问题描述:

在 MCU 上电的 10ms 内, PA5 不能被外部拉低, 否则会导致上电失败, MCU 不工作。

- 解决方法:

I 版之后的芯片会修复该问题。

对于 G、H 版, 应用上有以下限制:

1. 不能用 PA5 驱动阻性负载, 如果非得驱动阻性负载, 则该负载到地的电阻不能少于 200k 欧姆;
2. 不能用来驱动共阴极 LED 灯或者 LED 数码管, 但可以驱动共阳极 LED;
3. 可以驱动 CMOS 输入的其它 IC 管脚;

2.11 关于 ADC 的采样时间要求

- 适用芯片型号:

所有版本

- 问题描述:

主要是强调数据手册第 12.2.6“A/D 转换步骤”。

- 解决方法:

以下是使用 ADC 进行模数转换的步骤示例：

1. 配置端口：
 - 禁止引脚输出驱动器（见 TRIS 寄存器）
 - 将引脚配置为模拟
2. 配置 ADC 模块：
 - 选择 ADC 转换时钟
 - 配置参考电压
 - 选择 ADC 输入通道
 - 选择转换结果的格式
 - 打开 ADC 模块
3. 配置 ADC 中断（可选）：
 - 将 ADC 中断标志清零
 - 允许 ADC 中断
 - 允许外设中断
 - 允许全局中断
4. 等待所需稳定时间 $T_{ST}^{(1)}$ ；
5. 等待所需的采集时间 $T_{ACQ}^{(2)}$ ；
6. 将 GO/DONE 置 1 启动转换；
7. 通过以下情况之一等待 ADC 转换完成：
 - 查询 GO/DONE 位
 - 等待 ADC 中断（允许中断时）
8. 读取 ADC 结果；
9. 将 ADC 中断标志清零（在允许了中断的情况下这一步是必需的）。

以下是一段示例代码：

```
BANKSEL ADCON1 ;  
LDWI B'01110000' ;ADC Frc clock  
STR ADCON1 ;  
BANKSEL TRISA ;  
BSR TRISA,0 ;Set RA0 to input  
BANKSEL ANSEL ;  
BSR ANSEL,0 ;Set RA0 to analog  
BANKSEL ADCON0 ;  
LDWI B'10000001' ;Right justify,  
STR ADCON0 ; Vdd Vref, AN0, On  
LCALL StableTime ; ADC stable time  
LCALL SampleTime ;Acquisiton delay  
BSR ADCON0,GO ;Start conversion  
BTSC ADCON0,GO ;Is conversion done?  
LJUMP $-1 ;No, test again  
BANKSEL ADRESH ;  
LDR ADRESH,W ;Read upper 2 bits  
STR RESULTHI ;store in GPR space  
BANKSEL ADRESL ;
```


LDR ADRESL,W ;Read lower 8 bits
STR RESULTLO ;Store in GPR space

注意:

1. T_{ST} 时间是 ADC 的稳定时间, 当使用内部参考时, ADC 首次启动还需要考虑参考电压的稳定时间 T_{VRINT} , 等待时间应取两者的较大者, 即 $\max(T_{VRINT}, T_{ST})$;
2. 见图 1, ADC 转换时序;
3. 切换通道后必须等待足够长的 T_{ACQ} 时间, 即上面示例中的 SampleTime 必须满足 T_{ACQ} 时间要求 (具体见数据手册中表格 19.12), 否则 ADC 精度、线性度将不能保证;

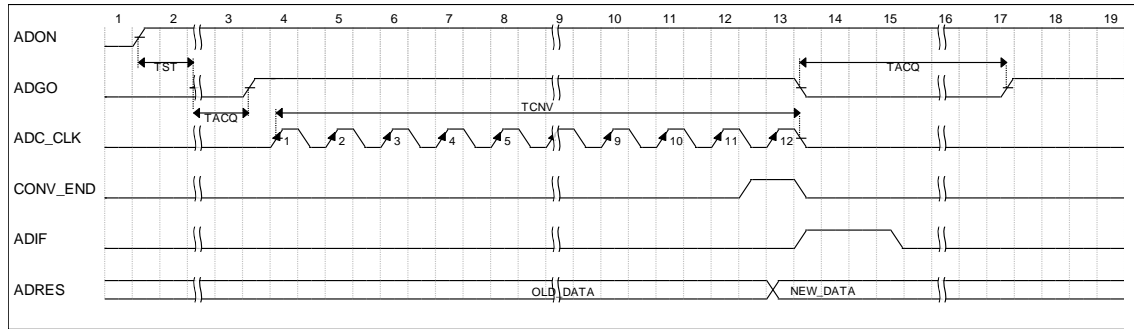


图 1 ADC 转换时序