



十速

TM52系列IC

系统时钟最高限值和
快、慢系统时钟切换注意事项

Application Note

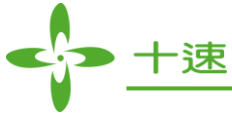
Rev V1.0

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.



修改记录

| 版本 | 日期 | 描述 |
|------|-----------|----|
| V1.0 | May, 2014 | 新颁 |



目录

| | |
|--|---|
| PRODUCT NAME..... | 4 |
| TM52 系列 IC 系统时钟最高限值和快、慢系统时钟切换注意事项..... | 4 |
| 一、 IC 系统时钟(CPUClk)频率最高值为 6MHz..... | 4 |
| 二、 TM52 系列 IC 快、慢系统时钟(CPUClk)切换注意事项..... | 5 |

PRODUCT NAME

TM52 系列 IC

TM52系列IC系统时钟最高限值和快、慢系统时钟切换注意事项

本 AP Note 针对现有 TM52 系列 IC: TM52F5284、TM52F5288、TM52M5254、TM52M5258 等的系统时钟(CPUClk)频率最高值和快、慢系统时钟间切换注意事项进行阐述。

一、 IC 系统时钟(CPUClk)频率最高值为 6MHz

1. IC 系统时钟(CPUClk)频率 > 7.2MHz 时, 会出现 Flash/MTP ROM 读取错误的情况; < 6.2MHz 时则不会读错 Flash/MTP ROM。所以 IC 系统时钟(CPUClk)频率最高限定为 6MHz。
2. 降低系统时钟(CPUClk)频率有助于提高 IC 的 EFT 抗干扰能力, 此频率越低效果越明显。如: CPUClk 取内部快速 RC 四分频(FIRC/4)要比取内部快速 RC 两分频(FIRC/2)抗 EFT 能力强。
3. IC 内部快速 RC(FIRC)中心频率为 7.3728MHz, 如使用此频率作为系统时钟(CPUClk), 因上述 1.条之缘故, 必须进行分频, 以保证分频后的频率不超过 6MHz。如果使用快速晶体(FXT)作为系统时钟(CPUClk),因上述 1、2.条之缘故, 最好对 FXT 频率进行分频, 保证分频后的频率不超过 6MHz 后提供给 MCU。这样有助于提高 EFT 抗干扰能力。如: 如果系统时钟(CPUClk)拟采用 4MHz: A.采用 4MHz crystal(FXT = 4MHz)不进行分频作为 CPUClk; B.采用 8MHz crystal(FXT = 8MHz)进行两分频后作为 CPUClk。B.种方案较 A.种方案有较佳的抗 EFT 能力, 建议采用 B 种方案。

二、 TM52 系列 IC 快、慢系统时钟(CPUClk)切换注意事项

IC 上电后系统时钟(CPUClk)默认为慢速内部 RC(SIRC),分频比为 1: 1(约 80KHz),程序如果需要将系统时钟(CPUClk)由 SIRC 切换成快速时钟(FIRC 或 FXT),需按如下步骤操作, 否则可能会导致程序运行错误:

1. 快/慢时钟切换 IRC 或者 XT(晶振)时必须在对立的(慢/快)时钟模式下进行。例如: 慢时钟从 SIRC(慢速 IRC)切换到 SXT(慢速晶振)模式时必须在系统时钟为快时钟的情况下进行。同理快时钟的模式切换必须在系统时钟为慢时钟的情况下进行。
2. 上电后将系统时钟(CPUClk)由 SIRC 切换成快速时钟(FIRC 或 FXT)时: 需先设定时钟分频比, 以保证 FIRC 或 FXT 分频后的频率不超过 6MHz, 设定完分频比后至少要延时 8 个 NOP 指令以上; 之后才将 FIRC 或 FXT 设定为系统时钟(CPUClk), 时钟切换后, 同样至少要延时 8 个 NOP 指令以上, 然后才执行后面的程序, 例程如下:

```
CLKCON=0X22; //b0010_0010 先对时钟 2 分频,(默认为 0x23=b0000_0011:1 分频)
_nop_();          //分频后至少要延时 8 个 NOP
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
SELFCK=1;        //切换 FIRC 或 FXT 为系统时钟 CPUClk)
_nop_();          //切换后至少要延时 8 个 NOP
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
System_int();
```

CLKCON 寄存器说明如下:

| | | Bit Name | Bit # | R/W | Rst | Description |
|----|--------|----------|-------|-----|-----|--|
| D8 | CLKCON | SCKTYPE | 7 | R/W | 0 | Slow clock Type. 0=SRC, 1=SXT. This bit can be changed only in Fast mode (SELFCK=1). |
| | | FCKTYPE | 6 | R/W | 0 | Fast clock Type. 0=FRC, 1=FXT. This bit can be changed only in Slow mode (SELFCK=0). |
| | | KICKSXT | 5 | R/W | 1 | Kick Slow Crystal |
| | | STPPCK | 4 | R/W | 0 | Set 1 to stop UART/Timer0/Timer1/Timer2/ADC Clock in Idle mode for current reducing. If it is set, only Timer3 and pin interrupts are alive in Idle Mode. |
| | | STPFCK | 3 | R/W | 0 | Set 1 to stop Fast clock for power saving in Slow mode. This bit can be changed only in Slow mode |
| | | SELFCK | 2 | R/W | 0 | Select Fast clock as System clock source. 1=Fast mode, 0=Slow mode. |
| | | CLKPSC | 1:0 | R/W | 0x3 | System clock Prescaler, 00: System clock is Fast/Slow clock divided by 16 01: System clock is Fast/Slow clock divided by 4 10: System clock is Fast/Slow clock divided by 2 11: System clock is Fast/Slow clock divided by 1 |