BJ8P161A
FAQ（汇集）

如果您在使用上还有其它问题请与我们联系：tbs002@bjxmcu.com

目 录

Q1：电气特性里的 VIL/VIH 的具体含义？ 4

Q2：MCU POWER ON 时，VDD 的上升时间有何限制？ 4

Q3：MCU INPUT PIN 若用于侦测 110V 或 220/230 V AC 讯号的过零点(ZERO CROSSING)功能时，外部应串接多大的电阻才不会造成 IC 损坏？ 4

Q4：仿真器内部的 VCC 可提供多大的电流？I/O 口最大的输出电流是多少？ 4

Q5：在睡眠模式下，芯片输入口应如何设置以获得最低电流？ 4

Q6：在程序中应如何设置才能增加 MCU 进出 GREEN MODE 的稳定性？ 5

Q7：芯片从低速模式进入睡眠模式，将芯片唤醒后，系统将工作于哪种模式？ 6

Q8：芯片为何无法进入 SLEEP MODE？ 6

Q9：如何正确设置进入低速模式？ 6

Q10：MCU 在 SLEEP MODE 功耗一般为多大？ 7

Q11：MCU 规格中提到的“1T”是什么含义？ 7

Q11：系统板采用 RC 振荡器，如何解决仿真和实际芯片工作的频率误差问题？ 7

Q12：应用 RC 振荡时频率应注意哪些问题？ 7

Q13：可以由目标板提供系统时钟吗？ 8

Q14：内部 ILRC 的频率为何? 8

Q15：当希望程式执行速率较高（FCPU 为 16M 或 8M）且实际系统将工作在杂汛较严重的场合，NOISE_FILTER 打开与否影响大不大？ 8

Q16：IHRC_RTC 模式使用注意事项？ 8

Q17：RAM 中资料最低保持电压是多少？是否有一个大概的范围可供参考？ 9

Q18：怎样选择 CODE OPTION 各选项？ 9

Q19：用 1M 晶振，CODE OPTION 应怎么选？ 10

Q20： BJ8P161A 怎么使用定义滚动码？ 11

Q21： BJ8P161A 的烧录文件是否需要转码？ 11

Q22： BJ8P161A 在普通模式/低速模式下，T0 时钟选择 “RTC” T0 应该产生 0.5S 中断，但实际中断时间计时不对？ 12

Q23： BJ8P161A 在绿色模式下，由 RTC 使能 T0 0.5S 唤醒一次？唤醒计时不准？ 12

Q1：电气特性里的 ViL/ViH 的具体含义？

◆ ViL——在一个 input mode 的 I/O 口上，输入电压从 0 向 VDD 变化，当 MCU 读到的

值从 0 变化的 1 时，此时对应的输入电压即为 ViL，即输入低电平。

◆ ViH——在一个 input mode 的 I/O 口上，输入电压从 VDD 向 0 变化，当 MCU 读到的值从 1 变化的 0 时，此时对应的输入电压即为 ViH，即输入高电平。

※ 注：当输入电压介于 ViL 和 ViH 之间时，例如 ViL/ViH 为 0.3VDD/0.7VDD，当输入电压为 0.5VDD 时，MCU 读到的电平将是不确定的，用户应注意避免这种情况。

Q2：MCU Power On 时，VDD 的上升时间有何限制？

MCU Power On 时为了使 Reset 可以完全成功，建议 Power On VDD 上升时间在 20ms 以内。当电池逐渐没电，电池内阻增大，造成 VDD 上升缓慢会影响 Reset 成功。 参考 Datasheet 的 VDD 上升速率不能小于 0.05v/ms。

Q3：MCU input pin 若用于侦测 110V 或 220/230 V AC 讯号的过零点(Zero Crossing)功能时，外部应串接多大的电阻才不会造成 IC 损坏？

由 AC LINE 直接连接到 IC 的 IO Port，主要须考虑的是来自 AC LINE 的浪涌电压及高频杂讯可能对 IC IO Port 造成的过流及过压破坏。 在 110 V 应用下，建议串接一个 2M 欧姆电阻。在 220/230 V 应用下，建议串接 4M~5M 欧姆。若使用 SMD 零件，由于其耐压值只有 200V，故必须串接 2 个 2M~2.5M 电阻。

Q4：仿真器内部的 VCC 可提供多大的电流？I/O 口最大的输出电流是多少？

仿真器内部的 VCC 可提供电流大概为 600mA。所有的 I/O 输出电流总和在 300Ma 左右。

Q5：在睡眠模式下，芯片输入口应如何设置以获得最低电流？

1. 睡眠模式下，输入口应设置为上拉状态（input pull-up）；不能设为浮动状态（input floating）。

2. 建议客户可透过以下两种方法实现：

(1). BJ 系列芯片大部份 I/O 都有内建可编程上拉电阻（programmable pull-up resistor），客户可透过程序设置内部上拉电阻使能（input pull-up resistor enable）。

例如：使能 P1 口上拉（enable P1 pull-up resistor）

```
mov A, #0FFh
mov P1UR, A     ;P1 上拉电阻缓存器 P1UR 设置为 1，使能上拉
```

(2). 对内部无上拉功能的输入口（例如：与 RST 共享的输入口），建议外接一个电阻（20Kohm）上拉到 VDD 。 另外，用户有时询问该如何设置不用的 IO 口以省电。其实，输入还是输出，是一样的，只要保持 IO 口有固定的状态，不要悬浮，即可减少漏电流的产生；当然输出口需要根据外部电路来确定状态。

Q6：在程序中应如何设置才能增加 MCU 进出 GREEN MODE 的稳定性？

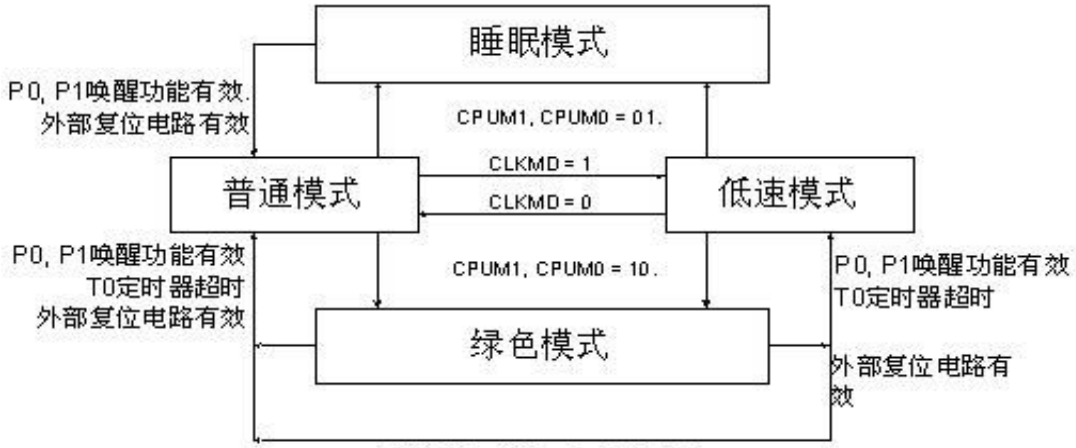
为了增加 MCU 进出 GREEN MODE 的稳定性，在进行模式切换时，必须使用 IDE 自带的宏指令来操作（在 M2IDE_V115 或以后的版本中已附带这部分宏指令，路径为 C:\Sonix\M2IDE_V115\use_inc2 下，相应母体的 inc 文件内），其它模式间的切换也必须使用对应的宏指令来完成。

关于各种模式操作的宏名称, 请参考下表:

宏名称	宏长度	说明
@SleepMode	1-word	系统进入 Sleep Mode (Power Down).
@GreenMode	3-word	系统进入 Green Mode, 包含系统同步处理程序.
@SlowMode	2-word	系统进入 Slow Mode, 同时停止高速振荡源.
@NormalMode	5-word	系统由 Slow Mode 切换至 Normal Mode, 程序包含启动高速振荡器, 高速振荡器 warm-up 程序与系统切换至 Normal Mode 控制程序.

Q7：芯片从低速模式进入睡眠模式，将芯片唤醒后，系统将工作于哪种模式？

下图是 MCU 的工作模式切换图。



系统模式切换示意图

从上图可以看到，不管系统是从低速模式还是从普通模式进入睡眠模式，当系统被唤醒后，它都会进入普通模式。

另外，系统从 Green Mode 被唤醒后，将返回原模式（普通模式或低速模式） 。

Q8：芯片为何无法进入 Sleep Mode？

需要进入 Sleep Mode，可在程序中设置：B0BSET CPUM0。如果无法进入 Sleep Mode，请检查以下三种情况：

1. 检查 watchdog 的设置，当选择 Always on 选项时，系统将无法进入 sleep 模式；
2. 检查 Port0 是否变化的电平信号输入，用户需要根据实际情况决定 Port0 是否上拉；
3. 检查是否在程序中设置 Port1 具有唤醒功能，如果有设置此功能，还需检测 Port1 是否有变化的电平信号输入。

以上三种情况都可使芯片退出 Sleep Mode，因此用户需确保外来信号的正确性，而且作为有唤醒功能的口都要上拉。

Q9：如何正确设置进入低速模式？

设置进入低速模式可分为两种情况：不停掉外部高速振荡器或停掉外部高速振荡器。

1 用户若对功耗要求不是很高，可以不停掉外部高速振荡器。

程式设置如： B0BSET FCLKMD

2 停掉外部高速振荡器以减小功耗。

B0BSET FCLKMD

B0BSET FSTPHX

以上指令执行顺序不能颠倒，若客户在没有设置进入低速模式前停掉外部高速振荡器，由于

此时程式还处于普通模式状态，振荡器一旦停振，系统将不再执行，从而导致出错。
详细资讯可参考 datasheet 中系统模式切换和系统低速时钟章节。

Q10：MCU 在 Sleep Mode 功耗一般为多大？

MCU 进入 Sleep Mode 时功耗将降至最小，一般不会超过 2uA。在-40℃~85℃时，其 Sleep Mode 功耗为 10~21uA。具体可参看 datasheet 电气特性章节

Q11：MCU 规格中提到的“1T”是什么含义？

“1T”是含义是 MCU 的运行时钟最高速度可为 Fcpu=Fosc，因为 BJ MCU 的绝大部分指令都是单时钟周期，所以系统执行一条指令的最短时间也为一个振荡周期。绝大部分芯片都支持“1T”，但是在有些情况下无法做到“1T”，比如在 Code Option 中选择 Noise_Filter Enable 时，MCU 的运行时钟最高速度只可选为 Fcpu=Fosc/4，即“4T”。

Q11：系统板采用 RC 振荡器，如何解决仿真和实际芯片工作的频率误差问题？

建议仿真器仿真采用晶振，实际芯片工作采用 RC 振荡电路的方式来开发。首先根据芯片内部工作频率计算采用外部晶振和 RC 振荡器的大概参数，仿真时仿真器上插入晶振，仿真通过后重新编译程序，将 High_Clk 选项修改为 RC，芯片烧录后调整目标板上 RC 振荡器的组件参数，直至符合要求。为了便于调整 RC 的组件参数，可以在开发时利用一闲置 I/O 口输出一固定频率方波，如此使用示波器观察波形调整 RC 的参数，也便于生产时的调试工作。如果没有闲置 I/O 口可使用，可编写一简单的测试程序，烧录芯片进行调整。

Q12：应用 RC 振荡时频率应注意哪些问题？

RC 振荡频率主要受下述条件影响：
1. RC 振荡的工作电压，不同的工作电压会影响 RC 振荡的工作频率。在工作频率较高时，应注意系统的工作电压，工作电压太低，可能出现系统工作的不稳定，特别是电池供电系统；
2. 工作环境温度，不同的环境温度也会影响 RC 的振荡频率；
3. 外部的干扰源，不同的 RC 组合，会有不同的抗干扰性能，PCB 布板也可以改善 RC 振荡的稳定性，C 端接地点应接 MCU 系统内部地，不应该直接接外部电路地。

Q13：可以由目标板提供系统时钟吗？

不能，进行仿真时默认为仿真器内部的晶振，目标板上的晶振不起作用。

Q14：内部 ILRC 的频率为何?

内部 ILRC 振荡器频率受工作电压影响较大，VDD=5V 时，约 32KHz，VDD=3V 时，约 16KHz。

Q15：当希望程式执行速率较高（Fcpu 为 16M 或 8M）且实际系统将工作在杂汛较严重的场合，Noise_Filter 打开与否影响大不大？

当客户的系统工作在杂汛较严重的场合，建议其将杂汛滤波器打开，即 Noise_Filter 使能（选择 Enable），这样可以有效滤除或减少外部环境带来的干扰。当 Code Option 中的 Noise_Filter 使能时，Fcpu 选项将自动遮罩掉 Fosc/1 和 Fosc/2 选项。因而即使外挂 16M 晶振，若 Noise_Filter 使能，Fcpu 最大也只能跑 4M。
用户应综合考虑，尽量降低执行速率。若程式对执行速率的确要求较高，可以考虑采用其他方法来降低干扰，如添加滤波电容等。

Q16：IHRC_RTC 模式使用注意事项？

1. 在 IHRC_RTC 模式下，使能 RTC 功能（B0BSET FT0TB），进入 GREEN 模式后，系统会自动关闭内部高速振荡（IHRC），用户不需要设置 FSTPHX；如果设置 FSTPHX（B0BSET FSTPHX），系统会关闭外部 32K 振荡，造成系统无法由 T0 定时唤醒；在 IHRC_RTC 模式下，不使能 RTC 功能（B0BCLR FT0TB），进入 GREEN 模式后，系统不会自动关闭内部高速振荡，此时设置 FSTPHX（B0BSET FSTPHX），系统会关闭内部高速振荡和外部 32K 振荡。
2. 在其它模式（非 IHRC _RTC 模式）下，进入 GREEN 模式后，系统都不会自动关闭内部高速或外部振荡，需要通过设置 FSTPHX 控制位来完成。

Q17：RAM 中资料最低保持电压是多少？是否有一个大概的范围可供参考？

RAM 中资料最低保持电压因 MCU 型号而异，用户可以参考相应 datasheet 电气特性章

节 RAM Data Retention voltage 中资料。详细资讯请参考 AN040：RAM 掉电记忆功能的实现。

Q18：怎样选择 Code Option 各选项？

不同的 chip 宣告和不同版本的编译软件，Code Operation 有不同的选项可供选择。其选项含义如下：

◆ Watchdog：

Always on—看门狗定时器一直开启；

Enable—看门狗定时器在 normal 和 slow 模式下开启，在 green 和 sleep 模式下停止；

Disable—看门狗定时器关闭。

※ 注：当选择 Always on 选项时，系统将无法进入 sleep 模式。

◆ Reset_Pin：

Pxx—选择内部复位，同时该引脚将作为单向输入口 Pxx 使用；

Reset—选择外部复位；

※ 注：当选择内部复位时，Pxx 口为单向输入口，且无内部上拉电阻。

◆ High_Clk：

IHRC_16M—芯片工作振荡源采用内部 16M 高速 RC 振荡电路；

Ext_RC—芯片工作振荡源采用外部 RC 振荡电路；

32K_X'tal—芯片工作振荡源采用外部低频率晶振（例如 32.768KHz）；

4M_X'tal—芯片工作振荡源采用外部标准石英或陶瓷振荡器（一般在 2M ～ 10MHz）；

12M_X'tal—芯片工作振荡源采用外部高速石英或陶瓷振荡器（一般在 10MHz ～ 16MHz）。

※ 注：IHRC_16M 选项只有在内部集成了高速 RC 振荡电路的 IC 型号中才会出现，当选择此项时，XIN/XOUT 两个引脚将作为一般 I/O 使用。

◆ Fcpu：

Fosc/1—指令周期 ＝1 个时钟周期；

Fosc/2—指令周期 ＝2 个时钟周期；

Fosc/4—指令周期 ＝4 个时钟周期；

Fosc/8—指令周期 ＝8 个时钟周期；

Fosc/16—指令周期 = 16 个时钟周期；

※ 注：当在 Code Option 中选择 Noise_Filter Enable 或 IHRC_16M 时，Fcpu 选项里的 Fosc/1 和 Fosc/2 两项将被自动屏蔽。

◆ Security：

enable—程序代码加密；

disable—程序代码不加密。

◆ Noise_Filter：

enable—打开噪声滤波功能。

disable—关闭噪声滤波功能。

※ 注：当开启噪声滤波功能后，会提高芯片的抗干扰能力，同时 Fcpu 选项里的 Fosc/1 和 Fosc/2 两项将被自动屏蔽。

◆ LVD：

LVD_L—VDD 低于 2.0V 时，LVD 复位系统。

LVD_M—VDD 低于 2.0V 时，LVD 复位系统，LVD 的 24-bit PFLAG 寄存器作为 2.4V 低电压监测器。

LVD_H—VDD 低于 2.4V 时，LVD 复位系统，LVD 的 36-bit PFLAG 寄存器作为 3.6V 低电压监测器。

Q19：用 1M 晶振，Code Option 应怎么选？

当所选外部振荡器频率小于 1MHz 时，可选择 32K_X ′ tal 选项，大于 1MHz 且小于 10MHz 可选择 4M _X ′ tal 选项， 10MHz 以上可选择 12M _X ′ tal 选项。

Code Option 中 High_Clk 各选项的选择只与仿真时振荡器的驱动能力有关，在实际运行时则以所挂晶振值为准。因而，使用 1MHz 晶振时，可选择 32K_X ′ tal 选项，也可选择 4M _X ′ tal 选项。

Q20： BJ8P161A 怎么使用定义滚动码？

```
例：    ORG             0X11    ; 0X11 代表滚动码首地址，且必须为奇数地址
    .ROLLING_CODE   2       ; 2 代表 2 个地址位做滚动码 滚动妈最大值为：0XFFFF 0XFFFF
```

Q21： BJ8P161A 的烧录文件是否需要转码？

用编程软件编译出的烧录文件为 *.sn8，必须要把 *.sn8 文件利用转码工具转换成 *.cs 文件才能利用 BJX 烧录器烧录。

Q22： BJ8P161A 在普通模式/低速模式下，T0 时钟选择 “RTC” T0 应该产生 0.5S 中断，但实际中断时间计时不对？

1、在 RTC 模式下，必须延迟 1/2 RTC 时钟源（32768Hz）之后再对 T0IRQ 作清零动作，否则 RTC 间隔时间可能出错。即从程序响应 T0 中断开始到 T0IRQ 再次被清零大约需要 16us。

2、在 RTC 模式下，T0 间隔时间固定为 0.5s，T0C 计数范围也固定为 256，中断服务程序中不能对 T0C 进行清零。

所以在普通模式/低速模式下，当 T0 时钟选择为 “RTC”后，开启 T0 中断，如果程序同时开启了 TC0 中断,可能对 T0 中断有影响，T0 中断 0.5S 时间会出错，建议在不要开 T0 中断，仅使能 T0 定时器，T0 时钟选择为 RTC，在主程序中查询 T0C 寄存器是否有溢出，有溢出就计一次数。可参考 RTC 例程序中“MAIN_RTC_ON”程序段落

Q23： BJ8P161A　在绿色模式下，由 RTC 使能 T0 0.5S 唤醒一次？唤醒计时不准？

在绿色模式下，由 RTC 使能 T0　0.5S 唤醒功能后，必须延迟 1/2 RTC 时钟源（32768Hz）之后再对 T0IRQ 标志清零动作，否则 RTC　0.5S 间隔时间可能出错。即从查询到 T0 中断标志 T0IRQ 为 1 开始到 T0IRQ 被清零大约需要 16us。参考 RTC 例程序中“MAIN_GREEN_RTC”程序段落

RTC 例程序：

```
    ORG         0
    JMP         REST                ;跳到第一次上电复位后程序
    ……
    ……
    ORG         10
REST:
    MOV         A,#07Fh             ;Initial stack pointer and
    B0MOV       STKP,A              ;disable global interrupt
    CLR         PFLAG               ;pflag = x,x,x,x,x,c,dc,z
    MOV         A,#00h              ;Initial system mode
    B0MOV       OSCM,A
    MOV         A, #0x5A
    B0MOV       WDTR, A             ;Clear watchdog timer
    CALL        ClrRAM              ;Clear RAM
    ……
    ……                              ;用户程序
MAIN_RTC_ON:
    CALL        T0_INIT             ;T0 INT
    B0BSET      FGIE                ;Enable global interrupt
    JMP         MAIN_LOOP

T0_INIT:
    B0BCLR      FT0IEN              ;清 T0 中断使能
    B0BCLR      FT0ENB              ;禁止 T0 中断请求
    CLR         T0C                 ;只在上电第一次清零，程序运行后不能再清零。
    B0BCLR      FT0IRQ              ;清 T0 中断标志
    B0BSET      FT0ENB              ;开启 T0
    B0BSET      FT0TB               ;使能 RTC
    B0BSET      F_PWRON             ;置第一次上电标志
T0_INIT_END:
    RET
; 在普通模式下，如果用 RTC 实时时钟 0.5S 功能，在主程序查询 T0C 寄存器是否有溢出，有溢出说明 0.5S
```

时间到了。

```
Main_LOOP:
    MOV         A, #0x5A
    B0MOV       WDTR, A             ;Clear watchdog timer
    MOV         A,T0C
    B0BTS1      FZ
    JMP         MAIN_RTC_TIME_RET1
    B0BTS0      F_PWRON             ;第一次上电不处理
    JMP         MAIN_RTC_TIME_RET
    B0BTS0      F_RTC
    JMP         MAIN_RTC_TIME_RET
    B0BSET      F_RTC
    M0V         A,#00000001B        ;用 P2.0 输出 0.5S 方波，测试 0.5S 准确性
    XOR         P2,a
MAIN_RTC_TIME_RET:
    JMP         MAIN_LOOP
MAIN_RTC_TIME_RET1:
    B0BCLR      F_RTC
    B0BCLR      F_PWRON
    JMP         MAIN_LOOP


MAIN_GREEN_RTC:
    B0BCLR      FTC0IEN             ;清 TC0 中断使能
    B0BCLR      FTC0ENB             ;禁止 TC0
    B0BCLR      FT0IEN              ; T0 不响应中断。
    B0BCLR      FT0IRQ              ; 清除 T0 中断请求。
    B0BSET      FT0ENB              ; 使能 T0 定时器。
    B0BSET      FT0TB               ; 使能 RTC。
MAIN_GREEN_DEAL:                    ; 进入绿色模式。
    B0BCLR      FCPUM0              ; CPUMx = 10。
    B0BSET      FCPUM1
    MOV         A, #0x5A
    B0MOV       WDTR, A             ;Clear watchdog timer
;在绿色模式利用 RTC0.5S 唤醒一次，需要要查询 T0 中断标志，并且要等待 16US 才能清除中断标志，否则计时出错
    B0BTS1      FT0IRQ
    JMP         MAIN_GREEN_DEAL
    JMP         $+1
    JMP         $+1
…….
…….
```

```
JMP     $+1
JMP     $+1
B0BCLR  FT0IRQ              ;等待大于 16US 后，才清中断标志
MOV     A,#00000010B        ;用 P2.1 输出 0.5S 方波，测试 0.5S 准确性
XOR     P2,A
JMP     MAIN_GREEN_DEAL
```

文件更新记录

序号	版本	更改单号	更改前内容	更改后内容
1	A/0	首次发布		