

TKS 系列仿真器 FAQ(1~30)

一、通用问题

1. 为什么我的仿真器通信不正常？
2. 我在仿真之后要把程序烧录到芯片中去，应该怎样生成 HEX 文件？
3. 仿真时如何使用我的用户板上的复位电路？
4. 仿真时如何使用我的用户板上的电源？
5. 我在仿真时如何使用外部晶振？
6. 为什么有时候读取片外 RAM 会出错？
7. 我的仿真器如何进行升级？
8. 在调试串口中，当我中断当前的程序运行时，再次运行无法进行串口正常的接收和发送。
9. 我在汇编语言中使用 ORG 伪指令定义了起始地址，并开始了一段程序代码的编写。然后又用 ORG 开始了另一段程序的编写，调试程序时出现紊乱，这是什么原因？
10. 我使用同一台仿真器，使用 DIP40 仿真头仿真很顺利，但使用 PLCC44 仿真头却经常出现程序运行不正常的现象。
11. 我用 74LS373 锁存低 8 位地址，但是仿真时经常地址锁存错误，请问如何解决？
12. 我在仿真调试一块用户板时，仿真有时很顺利，但有时完全不正确，请问一般是什么原因造成的？
13. 为什么我使用仿真器内部提供的电源仿真正常，而使用用户电源则不正常？

二、TKStudio 环境

14. 在 TKStudio 集成开发环境下编译时弹出以下窗口，是怎么回事？
15. 为什么在 TKStudio 中的变量观察窗口观察不到汇编变量？
16. 我在 TKStudio 开发环境中进行硬件调试时，运行速度很慢。

三、Keil IDE (μVision2) 环境

17. 如何查看当前仿真器内部的仿真芯片的型号？
18. 在 Keil C51 环境中如何正确安装驱动程序？
19. 为什么屏幕的数据刷新速度慢？
20. 我用 TKStudio 和 Keil 的 μVision2 环境调试中发现出很多莫名其妙的现象，程序指针乱飞，重新编译也不能解决问题。
21. 我在仿真中使用 TKS 仿真器进行单步操作时，为何仿真器进行连续运行状态，监控/运行指示灯连续闪烁？
22. 我在调试 C 语言调试时，发现某些语句在调试中被忽略了。
23. 在 Keil IDE (μVision2) 环境中，为什么对一个汇编程序进行编译时总是编译出错？
24. 为什么在 Keil IDE (μVision2) 环境中编辑中文时会出现乱码现象？

四、TKS-764

25. 使用 TKS-764 时，芯片的配置字应该怎么设置？

五、TKS-932

26. 请问 TKS-932 仿真器可以仿真 P89LPC935 吗？

27. TKS-932 仿真器有没有带 28 脚 DIP 插座的仿真头？

六、TKS 仿真器 S 系列

28. 我在使用 TKS-52S 的精密时间功能调试某些 C 语言语句时，发现显示的当前时间偏小，而总的时间显示正确。

七、TKS 仿真器 B 系列

29. 请教 B 型仿真器的 XDATA 空间 RAM 的几种定义。

30. 怎么观察 TKS-668B 的代码运行时间？

一、通用问题

1. Q: 为什么我的仿真器通信不正常?

A: 通信不正常的原因有很多, 通常是用户没有正确设置导致的。

- 1) 仿真器没有电源。
- 2) 仿真器同 PC 的通讯电缆连接不良。
- 3) 串口已经被其它设备占用, 例如您使用了串口调试助手, 而该程序又与仿真器使用同一个串口。
- 4) 仿真器是否有正确复位。仿真器复位时其黄色 LED 和绿色 LED 会交替闪烁几次, 如果你没有看到 LED 出现闪烁, 你需要按仿真器上的复位键或者重新上电, 以使仿真器复位。
- 5) 没有安装驱动。如果您使用 Keil C51 集成开发环境, 则需要手工安装驱动程序。每一款 TKS 仿真器都有其相应的驱动程序, 请按照仿真器随机光盘中的说明进行正确安装。
- 6) 在程序调试环境错误的选择了仿真器硬件驱动程序。你可以在 TKStudio IDE 环境的“工程->配置目标工程->调试”中 (Keil 环境则是在“Project->Option for target->Debug”中) 选中“使用仿真器”, 并在其下拉列表中选择相应的仿真器。
- 7) 串口的设置不正常。这可在 TKStudio IDE 环境的“工程->配置目标工程->调试->设置”中 (Keil 环境则是在“Project->Option for target->Debug->Settings”中) 进行设置。一是要正确选择串口, 你的仿真器连接到 PC 的哪个串口, 你就需要在设置选项中选择相应的 COM 口 (可以使用“Search”搜索)。

2. Q: 我在仿真之后要把程序烧录到芯片中去, 应该怎样生成 HEX 文件?

A: 在完成程序的编写, 并硬件仿真通过之后, 就可以把程序烧录到芯片中去, 在目标板上实际运行。烧录时需要十六进制文件 (即扩展名为“HEX”的文件)。如何产生 HEX 文件呢? 可以使用如下方法:

1) TKStudio 集成开发环境: 在“工程->配置目标工程->输出”对话框中, 选择“创建应用文件”, 然后点击“选择”按钮, 以设置 HEX 文件存放的文件夹, 并在“输出文件名”输入框中输入你要产生的 HEX 文件名。确定设置后重新编译, 即可产生 HEX 文件。烧录 HEX 文件到单片机, 可以使用周立功公司的 EasyPRO 系列通用编程器、ExpertPRO 系列芯片编程或者专用编程器进行编程。

2) Keil C51 环境: 在“Project->Option for target->Output”对话框中, 选择“Create HEX File”, 然后点击“Select Folder for Objects”按钮, 以设置 HEX 文件存放的文件夹, 并在“Name of Executable”输入框中输入你要产生的 HEX 文件名。确定设置后重新编译, 即可产生 HEX 文件。

3. Q: 仿真时如何使用我的用户板上的复位电路?

A: 可以在集成开发环境中进行配置。

1) TKStudio 环境: 在“工程->配置目标工程->调试->设置”对话框中选择“Use Ext.RST in”或者“Use Ext Reset”。

2) Keil C51 环境: 在“Project->Option for target->Debug->Settings”对话框中选择“Use Ext.RST in”或者“Use Ext Reset”。

注意: 在TKS-900仿真器中, 该选择项只在UCFG1的RPE开启后才有意义, RPE关闭用户将无法启动该选项。

4. Q: 仿真时如何使用我的用户板上的电源?

A: 首先, 要在集成开发环境中选择使用外部电源。

1) TKStudio 环境: 在“工程->配置目标工程->调试->设置”对话框中选择“Use External Power”。

2) Keil C51 环境: 在“Project->Option for target->Debug->Settings”对话框中选择“Use External Power”。

经过上述设置, 就可以使用用户板上的电源了。另外, 对于 TKS-61、TKS-52 等型号的单片机, 还有一个需要注意的地方, 那就是如果在使用外部电源的同时使用外部晶振, 则要将仿真头上的“OSC PWR”

短接到“Ext”一侧。

5. Q: 我在仿真时如何使用外部晶振?

A: 首先, 要在集成开发环境中选择使用外部晶振。

1) TKStudio 环境: 在“工程->配置目标工程->调试->设置”对话框中选择“External OSC”或者“TestBoard OSC”。

2) Keil C51 环境: 在“Project->Option for target->Debug->Settings”对话框中选择“External OSC”或者“TestBoard OSC”。

经过上述的设置, 就可以使用外部晶振了。外部晶振还有两种选择: 仿真头上的晶振或者用户板上的晶振。不同型号的仿真器, 其两种外部晶振的选择方法略有不同:

1) TKS-764、TKS-764B 等型号的仿真器: 仿真头上的晶振插座是直接为用户板上的振荡器引脚相连的, 所以仿真头和用户板上同时只能有一个晶振, 即如果使用用户板上的晶振, 就要把仿真头上的晶振取出来, 反之亦然。

2) TKS-61、TKS-52 等型号的仿真器: 其仿真头上的晶振插座与用户板上的振荡器引脚并不直接相连, 而是通过两个跳线器(靠近仿真头上的晶振插座)来选择的。把两个跳线器短接到“OnBoard OSC”一侧, 选择仿真头上的晶振; 把两个跳线器短接到“Target OSC”一侧, 则选择用户板上的晶振。还要注意的是如果使用用户板上的电源, 则无论选择仿真头上的晶振还是用户板上的晶振, 都要把“OSC PWR”跳线器短接到“Ext”一侧; 如果使用仿真器电源, 则把这个跳线器短接到“Int”一侧。

6. Q: 为什么有时候读取片外 RAM 会出错?

A: 可能的原因有几个:

1) 建议 P0 口加上拉电阻, 阻值 5K~10K。

2) 仿真频率过高, 建议在读取片外 RAM 时, 采用 12MHz 以下频率仿真。

3) 没有选择使用外部总线。对于单片机的 P0 口和 P2 口, 由于 TKS 系列中一些仿真器不同时仿真总线方式和 I/O 口方式, 所以在读取片外 RAM 时要先选择仿真总线方式。方法如下:

TKStudio 环境: 在“工程->配置目标工程->调试->设置”对话框中选择“Use Bus”。

Keil C51 环境: 在“Project->Option for target->Debug->Settings”对话框中选择“Use Bus”。

对于 TKS B 系列仿真器, 则可以根据需要选择“Use XBus Only”或“Use All Bus”, 具体使用方法参见用户使用手册。

7. Q: 我的仿真器如何进行升级?

A: 请定期留意我们的主页上的升级信息, 或者直接联系周立功公司的销售部门进行升级。升级联系电话: 020-38730916。

8. Q: 在调试串口中, 当我中断当前的程序运行时, 再次运行无法进行串口正常的接收和发送。

A: 这是仿真中出现的一种非常常见的现象, 当串口正处于发送/接收状态时, 如果暂停程序运行, 串口将从发送/接收状态中强行退出。由于串口的运行状态遭到破坏, 再次运行程序可能造成串口功能无法恢复的现象。另外, 单步运行 MOV SBUF,A 指令也会出现这种现象。

解决的办法: 用户应当尽量保证不要在串口发送/接收期间中断运行。

9. Q: 我在汇编语言中使用 ORG 伪指令定义了起始地址, 并开始了一段程序代码的编写。然后又用 ORG 开始了另一段程序的编写, 调试程序时出现紊乱, 这是什么原因?

A: 这种情况一般是由于二次 ORG 定义的代码段出现重叠造成程序编译出错。由于其它原因, Keil 的编译器无出错提示, 请检查, 在第二次使用 ORG 时, 程序代码已经超出 ORG 定义的代码地址范围。

10. Q: 我使用同一台仿真器，使用 DIP40 仿真头仿真很顺利，但使用 PLCC44 仿真头却经常出现程序运行不正常的现象。

A: 这种问题的出现是由于 PLCC 仿真头同用户板的 PLCC 插座接触不良造成的，尤其是 PLCC 插座弹性较差或用户插入过 PLCC44 芯片后尤为明显。用户可检查 PLCC44 仿真头与插座是否有污迹，或用镊子等工具将 PLCC44 插座的簧片重新整形调整。

11. Q: 我用 74LS373 锁存低 8 位地址，但是仿真时经常地址锁存错误，请问如何解决？

A: 这是由于 74LS373 阈值电压过低而引起的，由于使用仿真器肯定会引起额外的干扰，因此使用 74HC373 等阈值电压高的 373 芯片能有效的解决这个问题。

12. Q: 我在仿真调试一块用户板时，仿真有时很顺利，但有时完全不正确，请问一般是什么原因造成的？

A: 这种现象原因很多，用户应仔细检验用户板的硬件连接以及程序的稳定性。但是另一个问题应引起用户的注意，这就是仿真器以及用户板的上电顺序。

一般情况下，应先将仿真器上电，待自检结束后，再将仿真头插入用户板，然后将用户板上电。如果这种上电顺序不能解决问题，用户可将仿真器和用户板分别上电结束，然后再将二者带电连接。

13. Q: 为什么我使用仿真器内部提供的电源仿正常，而使用用户电源则不正常？

A: 用户应检查用户板上电源到仿真器的通路是否有连接不正常的现象，另外还应检查用户板的电源质量，包括电源电压的范围是否在仿真器允许的范围，电源能否提供足够的电流。

根据我们的经验，用户使用开关电源经常会出现这种现象，原因是开关电源输出电压不稳定，特别是经常出现高电平脉冲，引起仿真器内部保护电路启动，引起仿真器工作不正常，甚至损坏仿真器。用户应使用质量更高的开关电源，或直接使用稳压电源。

二、TKStudio 环境

14. Q: 在 TKStudio 集成开发环境下编译时弹出以下窗口，是怎么回事？



图 1 编译提示

A: 这是因为您没有安装 Keil C51 软件。TKStudio IDE 自身不带编译器，所以在使用之前请先安装 Keil C51。我们的随机光盘中有评估版的 Keil C51 软件，也可以联系我们公司购买无代码限制的正版软件。

15. Q: 为什么在 TKStudio 中的变量观察窗口观察不到汇编变量？

A: 在 TKStudio 集成开发环境中，变量观察窗口会区分大小写，以更好地支持 C 语言。因此，在观察汇编变量时，用户应使用大写。

16. Q: 我在 TKStudio 开发环境中进行硬件调试时，运行速度很慢。

A: 这是因为用户在硬件调试执行单步运行时，每运行一次，相应的窗口都会刷新一次。因此，为了加快运行速度，用户可关闭不需要更新的窗口。

三、Keil IDE (μ Vision2) 环境

17. Q: 如何查看当前仿真器内部的仿真芯片的型号?

A: 多数 TKS 仿真器可以更换内部的仿真芯片以适应用户不同的要求。由于性能更高的仿真芯片出现, TKS 可能变化出厂时仿真器内部缺省芯片的型号。用户可以自行查询当前仿真器内部的仿真芯片的设置, 方法是在 Keil 51 环境下进入硬件仿真环境, 在菜单“Peripherals”中点击“About TKS”查看内部仿真芯片的型号。

18. Q: 在 Keil C51 环境中如何正确安装驱动程序?

A: TKS 系列仿真器可以实现与 Keil C51 集成开发环境的无缝连接, 只要安装相应的驱动就可以了。如果使用仿真器随机光盘中的 Keil 安装程序, 驱动程序可以自动安装。

如果没有使用随机光盘中的 Keil 安装程序, 则相应的驱动程序要手工安装。驱动程序位于光盘的 Driver 目录下。以装 TKS-HOOKS 系列仿真器为例, 手工安装驱动程序的步骤如下:

1) 首先, 将随机提供的驱动程序拷贝到 Keil 的安装目录 C51\bin 下。例如, 如果您的 Keil 的安装目录为 C:\Keil, 则将 TKS_DEB.DLL 拷贝到 C:\Keil\C51\bin 目录下。

2) 打开 C:\Keil 目录下的 Tools.ini 文件, 在几个分类中找到[C51], 加入下列描述:

```
TDRV3=C:\Keil\C51\bin\TKS_DEB.DLL ("TKS Debugger")
```

其中TDRV3是驱动DLL的序号, 如果您前面已经安装了多个驱动DLL以致占用了TDRV3, 则将TKS的驱动程序序号向后顺延, 例如TDRV5。

19. Q: 为什么屏幕的数据刷新速度慢?

A: 可能与Cache Option缓存区域的设置有关。使用存储器缓冲区域, 这样在一般的操作中仿真软件不用频繁的读取仿真器中的内容, 而且可以大大加快仿真速度, 此时用户将该区域选项全部选中。使用缓冲的缺点是屏幕的数据刷新慢, 但是在单步或运行后所有在屏幕上显示的信息将全部刷新一次, 如果用户觉得刷新速度慢, 建议可以只选择“Cache Code”选项。

20. Q: 我用 TKStudio 和 Keil 的 μ Vision2 环境调试中发现出很多莫名其妙的现象, 程序指针乱飞, 重新编译也不能解决问题。

A: 此情况是由中文的兼容性引起的。Keil 提供的汇编编译器/C 语言编译器/连接器不支持中文路径, 因此建立的工程路径不要包含中文路径。

21. Q: 我在仿真中使用 TKS 仿真器进行单步操作时, 为何仿真器进行连续运行状态, 监控/运行指示灯连续闪烁?

A: 这种现象发生于使用 Keil 公司的 μ V2 仿真环境, 产生的原因是 Keil 对运行状态的解释出现问题。例如, 如果用户调试中出现 for(i=0;i<1000;i++)的程序行时, Keil 试图用连续单步完成该程序行的调试, 由于连续单步需要运行几千次, 将耗费很多的运行时间, 造成监控/运行指示灯长时间闪烁。在 C 语言调试中, 从汇编初始化程序到 main 函数也会出现这种现象。

解决的办法是: 用户可以在该语句中设置断点, 或使用“运行到”功能, 一旦出现这种现象, 用户可以点击“stop”按钮停止运行。用户也可以使用 TKStudio 进行调试, 该软件不存在这个问题。

22. Q: 我在调试 C 语言调试时, 发现某些语句在调试中被忽略了。

A: 这是由于 C 编译器的优化等级过高造成的。用户可以降低优化等级或改变语言的写法。

在 Keil μ V2 中优化等级的设置方法为: 选择“Project->Option for target->C51”, 然后点击“Code Optimization”中“evel”右边的下拉菜单, 进行相应优先等级的选择。

23. Q: 在 Keil IDE (µVision2) 环境中, 为什么对一个汇编程序进行编译时总是编译出错?

A: 对程序编译出错, 其可能的原因有很多:

- 1) Keil 的版本太低, 建议用户重新安装版本比较高的 Keil 软件。
- 2) 如果没有用到外部 ROM, 不要将 “Project->Option for target->Target” 中的 “Code Banking” 选中。
- 3) 如果用户以前习惯于使用其他的编译环境编写汇编程序, 则在改用 Keil µVision2 环境时要注意在 Keil 中编写汇编程序的一些差别, 以免出现语法错误, 导致编译出错。另外提醒用户, 在使用 TKS 仿真器时, 只能使用 Keil C51 或 TKStudio 集成开发环境对硬件进行仿真。

24. Q: 为什么在 Keil IDE (µVision2) 环境中编辑中文时会出现乱码现象?

A: 这是因为在 Keil µV2 环境中, 1 个中文占据 2 个字符的位置, 用户在修改和删除中文字符时要特别注意。用户还可以以文本方式打开相关程序文件, 对其中的中文字符进行编辑并保存, 然后仍可以在 Keil 环境中打开并可对其进行修改或编译, 这样可以避开乱码的产生。另外, 用户也可以选择 TKStudio 集成开发环境编写程序, 在该环境下不存在上述问题。

四、TKS-764

25. Q: 使用 TKS-764 时, 芯片的配置字应该怎么设置?

A: 如果在 Keil 环境中使用 TKS-764 时, 没有专门的驱动程序, 仿真时使用 “Keil Monitor-51 Driver” 驱动, 因此不能对芯片的配置字 (UCFG1 和 UCFG2) 进行设置。这就需要在源程序中添加如下代码:

```
/*  
ORG 0FD00H  
DB 7BH  
ORG 0FD02H  
DB 0FFH ;设置配置字,使能 A/D 转换等  
*/
```

如果是 C 语言编程, 则可以另外新建一个 CONFIG.ASM 文件, 并把该文件添加到工程中。然后在 CONFIG.ASM 文件中添加如下代码:

```
/*  
CSEG AT 0FD00H  
DB 7BH  
CSEG AT 0FD02H  
DB 0FFH ;设置配置字,使能 A/D 转换等  
END  
*/
```

五、TKS-932

26. Q: 请问 TKS-932 仿真器可以仿真 P89LPC935 吗?

A: 可以仿真, 但是有些限制, 只能仿真 P89LPC932 和 P89LPC935 共同的特性和功能模块。例如, TKS-932 仿真器不能仿真 P89LPC935 的 A/D 和 D/A 功能。不过, TKS-932 可以升级到完全支持 P89LPC935。

27. Q: TKS-932 仿真器有没有带 28 脚 DIP 插座的仿真头?

A: 由于 P89LPC932 等芯片没有 DIP 封装, 所以 TKS-932 仿真器在仿真 P89LPC932、P89LPC935 等 28 脚芯片时, 只提供带 28 脚 PLCC 插座的仿真头。如果要使用 28 脚的 DIP 插座, 则您可以单独购买选配的 TB900-DIP28 仿真头。

六、TKS 仿真器 S 系列

28. Q: 我在使用 TKS-52S 的精密时间功能调试某些 C 语言语句时, 发现显示的当前时间偏小, 而总的的时间显示正确。

A: 这种现象发生于 Keil 公司的 $\mu V2$ 调试环境中, 这是由于 $\mu V2$ 解释某些 C 语句出现偏差, 使用了多个单步操作合成该 C 语句。因此, 当前运行时间显示为最后一个单步的时间, 引起数值偏小, 而总的运行时间经过累加后仍然正常。

用户可以使用断点和全速运行的方法解决, 或使用 TKStudio 调试环境解决这个问题。

七、TKS 仿真器 B 系列

29. Q: 请教 B 型仿真器的 XDATA 空间 RAM 的几种定义。

A: 片内 XRAM: 指仿真器中仿真芯片内部 XRAM 数据空间; 片外 RAM: 指用户目标板上扩展的 SRAM 数据空间; 仿真器自带 RAM: 指仿真器内部自带的 64K SRAM 数据空间。

在用户对仿真器的“Xdata Memory Map”进行选择时, 需要注意的是, 添加的影像是仿真器自带 RAM 数据空间, 没有涉及到的空间地址则影像到仿真器外部。

当用户选择“Use All Bus”时, 即使选择“64K Internal RAM”, 当对 P0、P2 口进行读写操作时, 仍然会有读写信号产生。当用户选择“Use No Bus”时, 对 P0、P2 口进行读写操作时, 不会有读写信号产生, 但是对已定义的仿真器自带 RAM 数据空间操作时, 有读写信号产生。当用户使用仿真芯片内部 XRAM 数据空间时, 无论选择“Use All Bus”或“Use No Bus”都不会产生读写信号, 这就是片内 XRAM 与仿真器自带 RAM 的区别。

30. Q: 怎么观察 TKS-668B 的代码运行时间?

A: B 系列新增了一个极为先进的运行时间的显示功能, 显示的内容分为总运行时间/当前操作的运行时间/总运行时钟数/当前操作的运行时钟数, 在主寄存器窗口可以观察到。

- 总运行时间 Tsum: 为仿真系统从上电或复位后到当前状态经历的有效运行总时间。在监控状态并不会使该时间增大, 只有有效的运行操作, 例如单步/全速运行才能使之增加。

- 当前运行时间 Tcur: 记录了前一次有效运行操作经历的时间。例如, 运行一个单步经历了 1us, 则该时间显示为 1us; 再运行一个单步经历了 2us, 则该时间显示为 2us。与总运行时间的不断累加不同, 该时间是一个时间差, 便于用户观察本次操作经历的时间。

- 总运行时钟数 Nsum: 为仿真系统从上电或复位后到当前状态经历的振荡周期数。在监控状态并不会使该时间增大, 只有有效的运行操作, 例如单步/全速运行才能使之增加。

- 当前运行时钟数 Ncur: 记录了前一次有效运行操作经历的振荡周期数。例如, 运行一个单步经历了 12Clk 则显示为 12; 再运行一个单步经历了 12Clk, 则该时间显示为 12Clk。与 Nsum 的不断累加不同, 该时间是一个差值, 便于用户观察本次操作经历的时钟数。

注意事项:

Tsum/Tcur 显示的数值跟用户选择的时钟有关系。如果用户选择仿真器内部提供的时钟, 仿真器能完全

准确的显示Tsum/Tcur。如果用户选择外部时钟（External OSC），仿真器需要知道用户从外部输入时钟的频率。如果用户填写的时钟频率不准确，Tsum/Tcur显示的数值将是错误的。用户在选择External OSC后，需要在晶振频率设置栏中填入频率数值，注意是以MHz为单位的。

Nsum/Ncur显示的是运行中经过的时钟周期数，显示的数值是独立的，即使在选择外部时钟时不依赖于用户设置的时钟频率。Ncur在观察当前的时钟模式非常有用。例如，在仿真87C52X2芯片时，该芯片可以动态地改写当前的时钟模式6/12Clk。如果当前的时钟模式为12Clk，则执行一个NOP操作后Ncur=12；如果当前的时钟模式为6Clk，则执行一个NOP操作后Ncur=6。这种功能在使用有双时钟模式的仿真芯片，如P89C668/P87C591/P89CRD2/P89C6X等，可以非常方便的看出当前的时钟模式。

复位后Tsum/Tcur/Nsum/Ncur的数值全部为0，且不能改动。