

关于 AVR 单片机熔丝位的设置和拯救方法大全

AVR 单片机熔丝位的设置和详细的拯救方法

编辑作者: ZRP 2007-10-1 于深圳

熔丝位是 ATMEL 公司 AVR 单片机比较独到的特征。在每一种型号的 AVR 单片机内部都有一些特定含义的熔丝位,其特性表现为多次擦写的 E²PROM。用户通过配置(编程)这些熔丝位,可以固定地设置 AVR 的一些特性,参数以及 I/O 配置等,当然也包括对片内运行代码的锁定(加密)。

用户使用并行编程方式、ISP 编程方式、JTAG 编程方式都可以对 AVR 的熔丝位进行配置,但不同的编程工具软件提供对熔丝位的配置方式(指人机界面)也是不同的。有的是通过直接填写熔丝位位值(如: CVAVR、PonyProg2000 和 SLISP 等),有的是通过列出表格选择(如 AVR STUDIO、BASCOS-AVR)。前者程序界面比较简单,但是需要用户在仔细查询操作,会引起一些意想不到的后果,如造成芯片无法正常运行,无法再次定入 ISP 编程模式等。建议用户对 AVR 的熔丝位进行配置时,选择用户表格选择方式界面的编程软件,如 BASCOM-AVR。不过版主使用的是前者 PonyProg2000。

对 AVR 熔丝位的配置操作是比较细致的工作,用户往往忽视其重要性,或感到不易掌握。看到这么多的人对 AVR 的熔丝位不会使用和误操作,结合本人的使用实践,给出以下方面的意见和参考。

下面给出对 AVR 熔丝位的配置操作时的一些要点和需要注意的相关事项以及相应的拯救方法。

(一) 设置方法

1.1.1 正确配置 AVR 熔丝位

对 AVR 熔丝位的配置是比较细致的工作,用户往往忽视其重要性,或感到不易掌握。下面给出对 AVR 熔丝位的配置操作时的一些要点和需要注意的相关事项。有关 ATmega128 熔丝位的具体定义和功能请查看本书相关章节,在附录中将给出一个完整的汇总表。

(1) 在 AVR 的器件手册中,对熔丝位使用已编程(Programmed)和未编程(Unprogrammed)定义熔丝位的状态,“Unprogrammed”表示熔丝状态为“1”(禁止);“Programmed”表示熔丝状态为“0”(允许)。因此,配置熔丝位的过程实际上是“配置熔丝位成为未编程状态“1”或成为已编程状态“0””。

(2) 在使用通过选择打钩“√”方式确定熔丝位状态值的编程工具软件时,请首先仔细阅读软件的使用说明,弄清楚“√”表示设置熔丝位状态为“0”还是为“1”。

(3) 使用 CVAVR 中的编程下载程序时应特别注意,由于 CVAVR 编程下载界面初始打开时,大部分熔丝位的初始状态定义为“1”,因此不要使用其编程菜单选项中的“all”选项。此时的“all”选项会以熔丝位的初始状态定义来配置芯片的熔丝位,而实际上其往往并不是用户所需要的配置结果。如果要使用“all”选项,应先使用“read->fuse bits”读取芯片中熔丝位实际状态后,再使用“all”选项。

(4) 新的 AVR 芯片在使用前,应首先查看它熔丝位的配置情况,再根据实际需要,进行熔丝位的配置,并将各个熔丝位的状态记录备案。

(5) AVR 芯片加密以后仅仅是不能读取芯片内部 Flash 和 E2PROM 中的数据,熔丝位的状态仍然可以读取但不能修改配置。芯片擦除命令是将 Flash 和 E2PROM 中的数据清除,并同时两位定位状态配置成“11”,处于无锁定状态。但芯片擦除命令并不改变其它熔丝位的状态。

(6) 正确的操作程序是:在芯片无锁定状态下,下载运行代码和数据,配置相关的熔丝位,最后配置芯片的锁定位。芯片被锁定后,如果发现熔丝位配置不对,必须使用芯片擦除命令,清除芯片中的数据,并解除锁定。然后重新下载运行代码和数据,修改配置相关的熔丝位,最后再次配置芯片的锁定位。

(7) 使用 ISP 串行方式下载编程时,应配置 SPIEN 熔丝位为“0”。芯片出厂时 SPIEN 位的状态默认为“0”,表示允许 ISP 串行方式下载数据。只有该位处于编程状态“0”,才可以通过 AVR 的 SPI 口进行 ISP 下载,如果该位被配置为未编程“1”后,ISP 串行方式下载数据立即被禁止,此时只能通过并行方式或 JTAG 编程方式才能将 SPIEN 的状态重新设置为“0”,开放 ISP。通常情况下,应保持 SPIEN 的状态为“0”,允许 ISP 编程不会影响其引脚的 I/O 功能,只要在硬件电路设计时,注意 ISP 接口与其并接的器件进行必要的隔离,如使用串接电阻或断路跳线等。

(8) 当你的系统中,不使用 JTAG 接口下载编程或实时在线仿真调试,且 JTAG 接口的引脚需要作为 I/O 口使用时,必须设置熔丝位 JTAGEN 的状态为“1”。芯片出厂时 JTAGEN 的状态默认为“0”,表示允许 JTAG 接口, JTAG

的外部引脚不能作为 I/O 口使用。当 JTAGEN 的状态设置为“1”后，JTAG 接口立即被禁止，此时只能通过并行方式或 ISP 编程方式才能将 JTAG 重新设置为“0”，开放 JTAG。

(9) 一般情况下不要设置熔丝位把 RESET 引脚定义成 I/O 使用（如设置 ATmega8 熔丝位 RSTDISBL 的状态为“0”），这样会造成 ISP 的下载编程无法进行，因为在进入 ISP 方式编程时前，需要将 RESET 引脚拉低，使芯片先进入复位状态。

(10) 使用内部有 RC 振荡器的 AVR 芯片时，要**特别注意熔丝位 CKSEL 的配置**。一般情况下，芯片出厂时 CKSEL 位的状态默认为使用内部 1MHz 的 RC 振荡器作为系统的时钟源。如果你使用了外部振荡器作为系统的时钟源时，不要忘记首先正确配置 CKSEL 熔丝位，否则你整个系统的定时都会出现问题。而当在你的设计中没有使用外部振荡器（或某种特定的振荡源）作为系统的时钟源时，千万不要误操作或错误的把 CKSEL 熔丝位配置成使用外部振荡器（或其它不同类型的振荡源）。一旦这种情况产生，使用 ISP 编程方式则无法对芯片操作了（因为 ISP 方式需要芯片的系统时钟工作并产生定时控制信号），芯片看上去“坏了”。此时只有使用取下芯片使用**并行编程**方式，或使用 **JTAG** 方式（如果 JTAG 为允许时且目标板上留有 JTAG 接口）来解救了。另一种解救的方式是：尝试在芯片的晶体引脚上**临时人为的叠加上不同类型的振荡时钟信号**，一旦 ISP 可以对芯片操作，立即将 CKSEL 配置成使用内部 1MHz 的 RC 振荡器作为系统的时钟源，然后再根据实际情况重新正确配置 CKSEL。

(11) 使用支持 IAP 的 AVR 芯片时，如果你不使用 BOOTLOADER 功能，注意不要把熔丝位 BOOTRST 设置为“0”状态，它会使芯片在上电时不是从 Flash 的 0x0000 处开始执行程序。芯片出厂时 BOOTRST 位的状态默认为“1”。关于 BOOTRST 的配置以及 BOOTLOADER 程序的设计与 IAP 的应用请参考本章相关内容。

1.1.2 ATmega128 中重要熔丝位的配置

上一小节介绍了配置 AVR 熔丝位的要点和注意事项，本小节把在一般情况下使用 ATmega128 时，几个重要的熔丝位配置情况进行说明。

(1) 熔丝位 M103C。M103C 的配置将设定 ATmega128 是以 ATmega103 兼容方式工作运行还是以 ATmega128 本身的方式工作运行。ATmega128 在出厂时 M103C 默认状态为“0”，即默认以 ATmega103 兼容方式工作。当用户系统设计使芯片以 ATmega128 方式工作时，应首先将 M103C 的状态配置为“1”。

(2) CLKSEL0..3。CLKSEL0、CLKSEL1、CLKSEL2、CLKSEL3 用于选择系统的时钟源。有五种不同类型的时钟源可供选择（每种类型还有细的划分）。芯片出厂时的默认情况为 CLKSEL3..0 和 SUT1..0 分别是“0001”和“10”。即使用内部 1MHz RC 振荡器，使用最长的启动延时。这保证了无论外部振荡电路是否工作，都可以进行最初的 ISP 下载。对于 CLKSEL3..0 熔丝位的改写需要十分慎重，因为一旦改写错误，会造成芯片无法启动，见上一小节第 10 点说明。

(3) JTAGEN。如果不使用 JTAG 接口，应将 JTAGEN 的状态设置为“1”，即禁止 JTAG，JTAG 引脚用于 I/O 口。

(4) SPIEN。SPI 方式下载数据和程序允许，默认状态为允许“0”。一般保留其状态。

(5) WDTON。看门狗的定时器始终开启。WDTON 默认为“1”，即禁止看门狗的定时器始终开启。如果该位设置为“0”后，看门狗的定时器就会始终打开，不能被内部程序控制了，这是为了防止当程序跑飞时，未知代码通过写寄存器将看门狗定时器关断而设计的（尽管关断看门狗定时器需要特殊的方式，但它保证了更高的可靠性）。

(6) EESAVE。执行擦除命令时是否保留 E2PROM 中的内容，默认状态为“1”，表示 E2PROM 中的内容同 Flash 中的内容一同擦除。如果该位设置为“0”，对程序进行下载前的擦除命令只会对 FLASH 代码区有效，而对 E2PROM 区无效。这对于希望在系统更新程序时，需要保留 E2PROM 中数据的情况下是十分有用的。

(7) BOOTRST。决定芯片上电启动时，第一条执行指令的地址。默认状态为“1”，表示启动时从 0x0000 开始执行。如果 BOOTRST 设置为“0”，则启动时从 BOOTLOADER 区的起始地址处开始执行程序。BOOTLOADER 区的大小由 BOOTSZ1 和 BOOTSZ0 决定，因此其首地址也随之变化。(8) BOOTSZ1 和 BOOTSZ0：这两位确定了 BOOTLOADER 区的大小以及其起始的首地址。默认的状态为“00”，表示 BOOTLOADER 区为 4096 字，起始首地址为 0xF000。

(9) 推荐用户使用 ISP 方式配置熔丝位。配置工具选用 BASCOM-AVR（网上下载试用版，它对 ISP 下载无限制），和 STK200/STK300 兼容的下载电缆（见第四章内容）。

注：不同 AVR 的熔丝也不同，使用前必须仔细查看芯片手册。

要重视手册学习，不仅是掌握如何使用，也是从根本上认识和掌握原理和结构。对于硬件工程师来将，数据手册是真正的“经书”，其它都是“修炼经验”。不熟读“经书”，你无法修炼成“仙”的。这也是《M128》、《M8》的目的之一！

（二）AVR 熔丝位拯救方法详细攻略

1.1.3 AVR 芯片的 ISP 全攻略

当然你在配置熔丝位时要小心从事，防止芯片锁死。有不少网友屡屡遭此不幸。其实你在熔丝位配置时，只要方法正确，一般是不会出错的。如果当你改动了 AVR 的熔丝位配置，重新加电后，想再用 ISP 下载，提示：“进入编程模式失败”等，极有可能是你搞错了熔丝位，导致芯片不知道使用何种外部主频而无法正常工作(仅限于使用内部 RC 振荡的情况)。不过，不用太担心，其实拯救的办法还是有很多的，操作起来也是很容易的，具体解决方法为：

1. 进入 JTAG 模式修改 SPIEN 熔丝位

需要使用 JTAGICE 调试器，不过一般情况下调试模式也进不去，JTAGEN(值为 1)锁死了。

2. 并行编程器恢复 需要你要有 USB 或 COM1 口制作的并行 HVPROG 编程器，STK200/STK300

3. 通过外加有源晶振 需要你要有外部有源振荡晶体（石英或晶振 1-4M）这个方法最简单，速度最快（强烈推荐）

4. 使用通用编程器将芯片恢复到出厂状态 这个方法，需要你要有市场上的通用编程器，可以修改三个熔丝位寄存器数据。

5. 寄回给芯片服务商，让他们帮忙将芯片恢复 这是最省事，但是最费时间，最无可奈何的方法。

先介绍一下基本恢复理论

ISP(In System Programmability) 在系统编程，简称为 串行下载

IAP(In Application Programing) 在应用编程，BootLoader 也是类似的意思

1. ISP 虽然利用了 SPI 接口(例外:M64/M128 为 UASRT0 接口，Tiny13 等没有 SPI 接口)的引脚，但只在复位时起作用，而且下载完成后合格的下载器会自动断开端口的连接，对正常工作时没有影响的（在产品应用中，下载器一定是不会一直粘在上面的）。

2. 虽然高压并行下载能修复任何熔丝位，但对于贴片封装来说是很不现实的，所以添加 ISP10 标准插座接口后就是最常用的 ISP 下载方式了

3. 虽然 IAP 是一种新的升级方法，但 IAP 程序本身也是要先用高压并行下载或 ISP 来烧进芯片里面才行

4. Tiny13 等少管脚 AVR 芯片因为管脚实在太少了，有 ISP,但没有[高压并行编程]而特制了[高压串行编程]

所以，产品上一般都留有 ISP10 标准接口插座，或更省位置的 ISP6 标准插座----留 6 个焊盘就行了

ISP 的工作前提

1. 芯片没有物理损坏 2. 芯片的 SPIEN 熔丝位=0 使能 ISP 功能

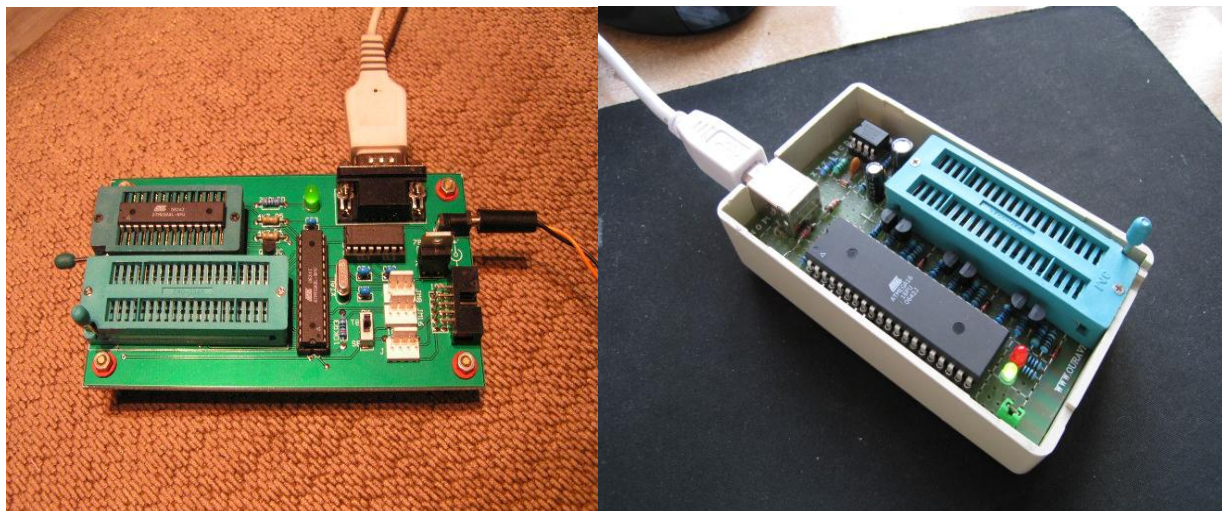
3. 芯片的 RSTDISBL 熔丝位=1 RESET 引脚有效 (假如芯片有这个熔丝位)

4. 线路正常--接错线? 短路? 5. ISP 下载器正常-----特别要考虑 连线的接触不良问题

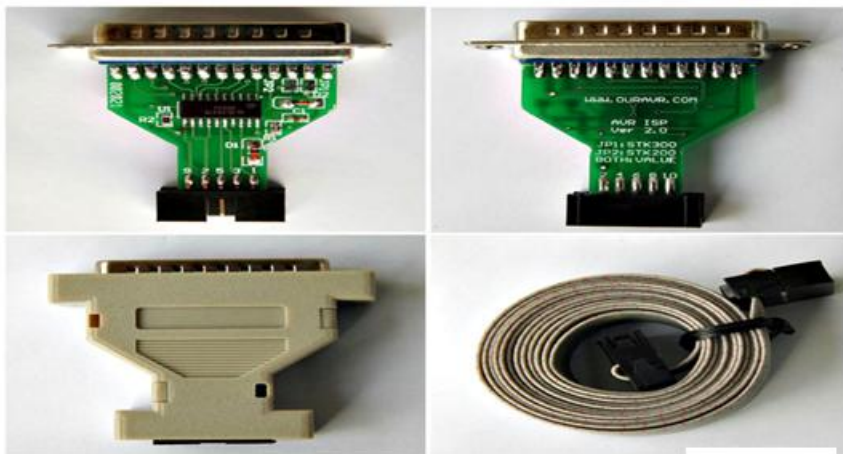
6. 电源正常

(一) 并行编程器恢复，最早的编程方法，功能最强大，通常需要 12V~24V 的高压，以示区别，下面称为 高压并行编程。制作方式也有：

1.USB 或 COM1 口制作的并行 HVPROG 编程器,软件例如 USBASP 的 ISP 上位机软件 AVR_fighter.exe(zhb2000) 和 USB AVRISP (ZHIFENG)，但需要 AVR MCU 连接较多的引脚（一般 20 脚）；



2. 并行编程器 STK200/STK300, 串行编程器软件 PonyProg2000. 但需要电脑留有 LPT 打印机接口, AVR MCU 留有 ISP6-10 标准烧写口。



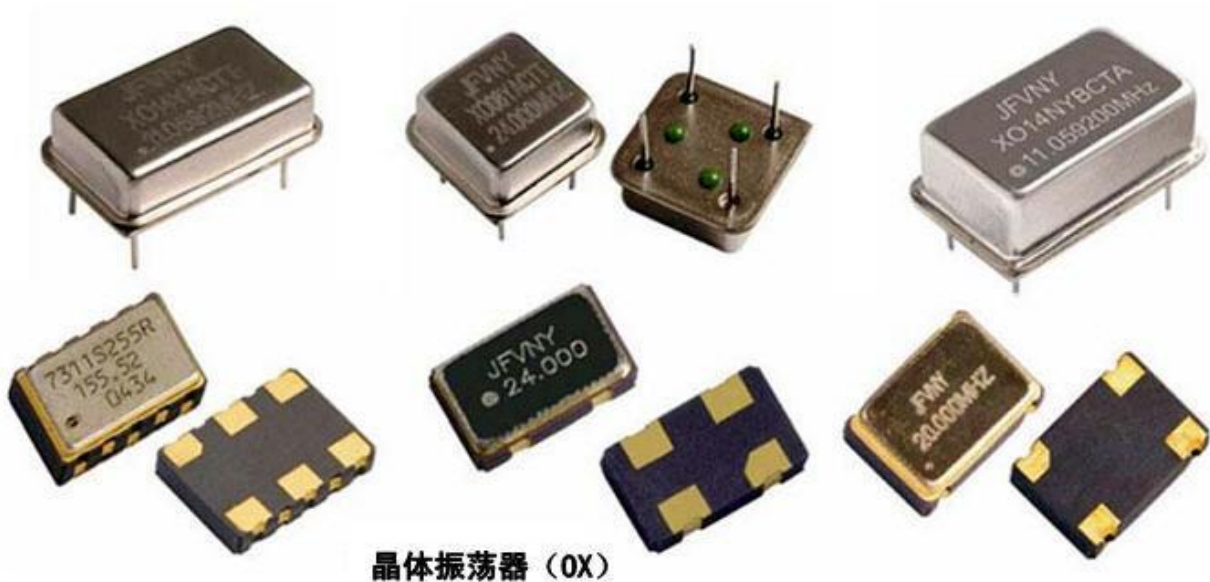
1. 以我们常用的小马 PonyProg2000 串行编程器下载软件为例,在 Command 菜单中选 Security and Configuration Bits, (或者点 Security and Configuration Bits 按钮) 在接着出现的熔丝位配置窗口的下面, 有几个按钮, 千万不要乱按。正确的方法是先按“Read”按钮, 读出原来芯片中熔丝位的配置, 然后再以此为基础, 按你的需要进行局部修改, 确定无误后再按“Write”按钮, 就 OK 了。一般不会出现锁死的问题。如果万一由于操作不当, 出现锁死也不必惊慌。

2. 除了用并行编程器恢复的方法以外, 对于没有并行编程器的朋友, 还可以用外接有源振荡器的方法解锁, 方法是用一个 1MHz 的振荡信号接到 ATmega16L 的 13 脚 XTAL1, 接通单片机的电源, 正常的话你会看到原来单片机中的程序已经可以正常运行了。说明外接振荡器已经使单片机恢复工作了。这下来就好办了, 仍然用 PonyProg2000, 点 Security and Configuration Bits 按钮, 进入熔丝位配置窗口, 这时你会看到所有的熔丝位都没打√, 连灰色的 SPIEN 前面那个模糊不清的√也没了, 不过没关系 Don't worry, 你只要按照下图中 ATmega16L 出厂时熔丝位的设置打√, 然后按“Write”按钮, 如果屏幕轻轻的闪一下, OK, 恭喜你已经好了, 再按“Read”按钮, 你会看到除了你打√的地方以外, SPIEN 前面那个模糊不清的√也出来了。锁死的芯片已经解锁了。



要注意的是外接的信号幅度不可太大，以免损毁芯片，我开始用一个 51 单片机的震荡信号接过来，结果不行，可能是驱动能力不够。后来换了一个信号发生器输出的 1MHz 的信号，就成功了。也可以用 NE555 芯片搭个振荡器做信号源。

(二)外加有源晶振



晶体振荡器 (0X)

运行时钟 ISP 时钟(必须低于运行时钟的 1/4)

4096Hz <1024Hz //很变态的用法，外接 32.768KHz 晶体+CKDIV8 ,不过 AVRISP 还是提供了 603Hz 这个速度了 //另一简易解决办法是 下载时在 32.768KHz 晶体并联一个 1MHz 晶体，双龙的下载线就配有一个 8MHz 的石英晶体

32768Hz <8192Hz

128KHz < 32KHz //内部 RC128KHz

1.0MHz <250KHz //默认值(包括 8MHz+CKDIV8)，所以 AVRISP 的 ISP 速度多为 230KHz

8.0MHz <2000KHz

16.0MHz <4000KHz

运行时钟不等于震荡器的频率，因为部分 AVR 芯片有系统时钟预分频器，可以对震荡器进行 1~256 分频

CKDIV8 熔丝位决定 CLKPS 位的初始值。

若 CKDIV8 未编程, CLKPS 位复位为“0000”; 若 CKDIV8 已编程, CLKPS 位复位为“0011”, 给出启动时分频因子为 8
AVRISP 可提供的 ISP 时钟 921.6KHz, 230.4KHz, 57.6KHz, 28.8KHz, 4.0KHz, 603Hz

STK500 可提供的 ISP 时钟 1.845MHz, 460.8KHz, 115.2KHz, 57.6KHz, 4.0KHz, 1206Hz

时钟设定 ISP 方案 内部 RC 选择合适的 ISP 速度

外部 RC 接上合适的电阻和电容, 选择合适的 ISP 速度。-----补救: 外部时钟源接到 XTAL1

外部 RC 根本就没有什么意义, 频率精度/稳定度不高, 成本也没有降低, 所以新的 AVR 芯片已经没有这个选项了。

各位网友要注意的是错误设定后补救方法

外部晶体 接上合适的晶体, 选择合适的 ISP 速度。 -----补救: 外部时钟源接到 XTAL1

外部时钟 接上合适的时钟源, 选择合适的 ISP 速度。 -----补救: 外部时钟源接到 XTAL1

外部时钟源可以是 外部(4MHz)有源晶体输出, 其他 MCU 的 XTAL2 脚, 各种方波振荡电路(NE555)输出等

大部分 AVR 芯片的 ISP 端口是 SCK,MOSI,MISO,RESET

而 M64/M128 的 ISP 端口是 SCK, PDI, PDO,RESET

而且 M64/M128 出厂默认兼容 M103----熔丝位 M103C=0, 很多新特性不能使用, 程序也可能不能正常运行

----因为 C 编译器通常默认自动把 SP 指向 SRAM 的末端, M103=0x0FFFH, M64/M128=0x10FFH, 必然出错!

AVR 的所有熔丝位均是:

1 未编程, 多为不起作用的意思。

0 编程, 多为 起作用的意思。

基于可编程工艺的都是这样:

PROM/EEPROM/FLASH 都是出厂时和擦除后变为全 1(0xFF)的,要编程才能变成 0。

反过来就是了, 跟 CE/OE/INT 都是[低电平有效]一样, 都是很常见。

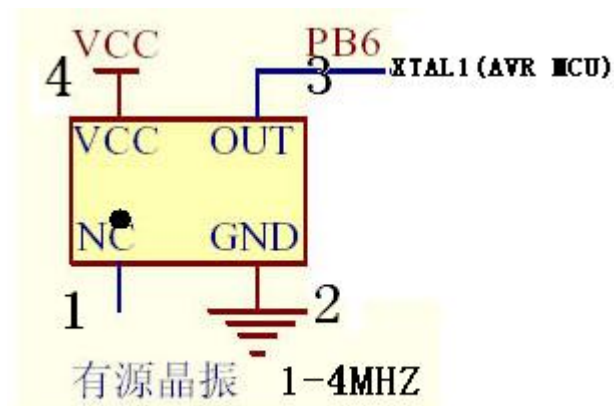
在 ISP 模式下永远不能访问(修改)SPIEN 位, 这是 AVR 芯片的硬件保护

有独立 RESET 脚的 M16/M32/M64/M128 等, 在 ISP 模式下根本就就不会令 ISP 无效, 无论如何修改熔丝位, 都能恢复正常。M8/M48/M88/M168/Tiny 系列有 RSTDISBL 熔丝位可以令导致 RESET 失效而令 ISP 无法工作外, 其他情况都能恢复正常。

一般来说, 只要满足 ISP 的工作前提, 再把 XTAL1 接到一个 4MHz 有源晶体的输出, 基本是万试万灵的。

通过外加有源晶振的办法, 让其恢复, 这个方法最可行。它可以恢复大部分熔丝位搞错的芯片。(重点推荐)

接法如下:



打点的口子对应 1 号脚 依次逆时针

1 NC 2 GND 3 CLK 4 VCC

1 脚不接, 2 脚接地, 3 脚接输出, 4 脚接电源。

恢复方法:

接上上图的有源晶振，重新通电，可以看到芯片又重新正常工作了，这时就能用 ISP 或 JTAG 下载线修改错误的熔丝位了。修改完成后，断电，将有源晶振拆走，看看是否已经恢复正常。

还有一个办法，如果没有有源晶振的话可以用**其他工作正常的单片机的时钟**作为外部晶振，只要将工作正常的单片机的 XTAL2 脚连接熔丝设置错误的单片机的 XTAL1 引脚即可。像我使用 AVR910 下载线的可直接把 AT90S2313 的时钟输出连到被设置错的芯片就可恢复了，很方便。

不要忘记，并行高压编程的时钟信号也是从 XTAL1 导入方波信号的。

如果有源晶振的方法不行(除了 $ISPEN=0, RSTDISBL=0$ 情况外)，恐怕高压编程也未必能奏效。

其他一些功能操作对 ISP 熔丝位的影响

1. JTAG 的影响(M16,M32,M128 等):

JTAG 能访问 SPIEN 和 JTAGEN,要是不小心同时改成 $SPIEN=1, JTAGEN=1$ ，将会导致 MCU 锁死，需要高压并行编程才能恢复。

2. DebugWIRE 的影响:(M48,M88,M168,T2313 等,数据手册里面的资料不是很详细)

由于 DebugWIRE 使用 RESET 脚来通讯，所以跟 ISP 有所冲突

可以通过 ISP 或并行高压编程来使能 DebugWIRE 功能[即 $DWEN=0$]，使能 DebugWIRE 功能后,ISP 功能失效。

可以通过 DebugWIRE 来关闭 DebugWIRE 功能[即 $DWEN=1$]，关闭 DebugWIRE 功能后,如果 $RSTDISBL=1, SPIEN=0$ ，ISP 功能有效。

比较特殊的是 DebugWIRE 调试中，断点的使用会降低 Flash 数据记忆时间 DebugWIRE 调试用的器件不能发给最终客户。

3.JTAG MKII 同时具备 JTAG/DeubgWIRE/ISP 三种功能，可以轻松实现 DebugWIRE/ISP 的切换。

(软件需要升级到 1.09 版以后 即对应 AVRstudio 4.12 以后版本)

http://218.16.124.196/bbs/bbs_content.jsp?bbs_sn=551225&bbs_page_no=1&sub_kind_id=1205&bbs_id=1000>最新版本 JTAG MK2 使用说明中文 pdf (20051125)

设计使用 debugWIRE 的系统时，必须进行下面的检查：

- $dW/(RESET)$ 的上拉电阻不得小于 $10k\Omega$ 。debugWIRE 并不需要上拉电阻
- 将 RESET 引脚与 VCC 直接连接将无法工作
- 使用 debugWIRE 时必须断开与 RESET 引脚连接的电容
- 必须断开所有的外部复位源

End