



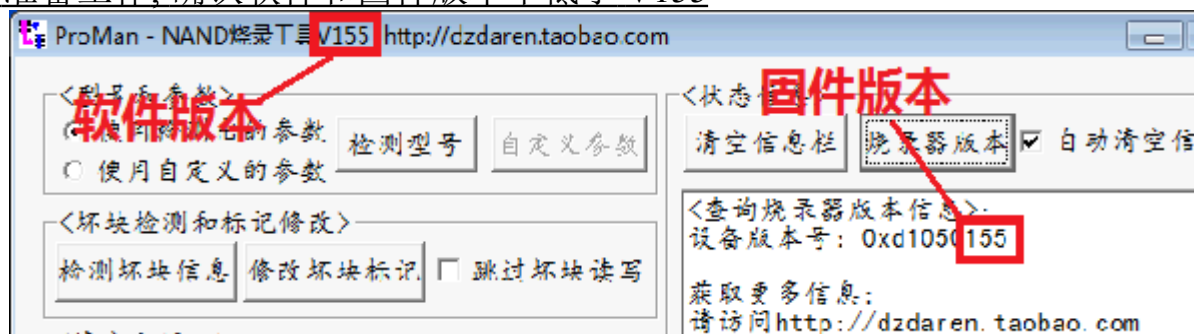
### 前言:

近期总结下来发现,好几位朋友是在利用 ProMan 烧录器从**源片**(也叫母片)把数据提取出来,然后写入到**新片**(也叫空片,目标片)中去,接着焊接新片到板子上去运行。现在把先前的一些经验总结成本文,尽量说细说清楚,希望能为大家带来一些方便。

### 主要内容:

0. 准备工作
  1. 第一步,读取整个源片的数据
  2. 第二步,没有坏块的新片的处理方法
  3. 第三步,有坏块的新片的处理方法之一
  4. 第四步,有坏块的新片的处理方法之二
  5. 附录:坏块的一些基础知识

### 0. 准备工作,确认软件和固件版本不低于 V155



## 1. 第一步，读取整个源片的数据

用户读取整个源片的数据，并且备份到电脑上，特别注意**冗余区数据也要一起读出来**。整片读取的时候，起始地址和结束地址的设置如下：

16M 字节容量的芯片是 0x0 - 0xffffffff

32M 字节容量的芯片是 0x0 - 0x1fffffff

64M 字节容量的芯片是 0x0 - 0x3fffffff

128M 字节容量的芯片是 0x0 - 0x7fffffff

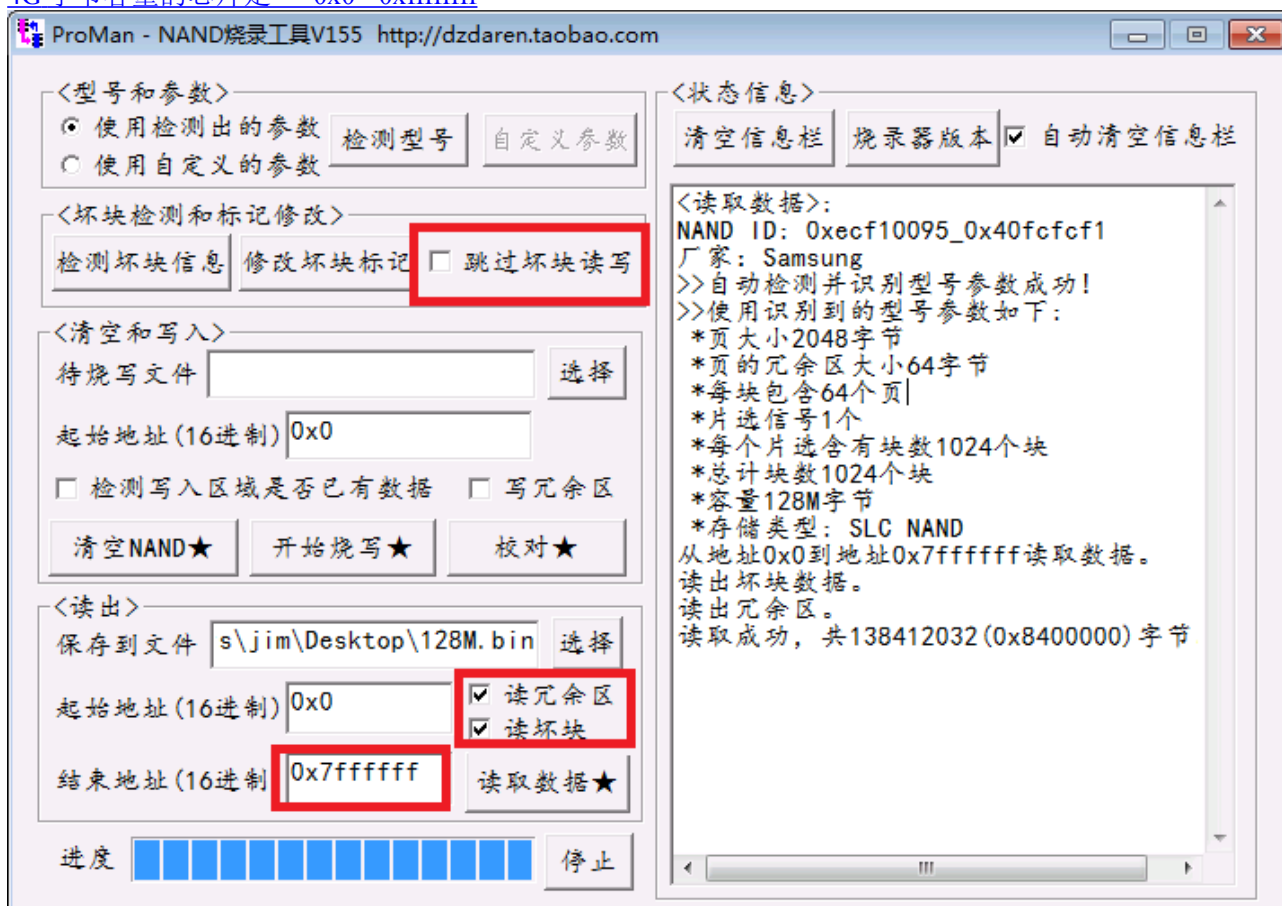
256M 字节容量的芯片是 0x0 - 0xffffffff

512M 字节容量的芯片是 0x0 - 0x1ffffffff

1G 字节容量的芯片是 0x0 - 0x3ffffffff

2G 字节容量的芯片是 0x0 - 0x7ffffffff

4G 字节容量的芯片是 0x0 - 0xffffffff



上面是读 128M 字节芯片 K9F1G08U0C-PCB0 的例子。注意**勾上**[读坏块]，**勾上**[读冗余区]和**不要勾上**[跳过坏块读写]，然后开始读，并把数据保存为 **128M.bin**(也可以取其它名字)。

因为带冗余区一起读，每页 2048 字节多了 64 个字节冗余区数据，所以实际读出的文件有 138412032 字节，比 128M 字节大一些。此处建议用户读 2 次，然后用文件比较工具(推荐 Beyond Compare)比较一下，确认提取到的数据是正确的。具体请参考<<手册 5 – BeyondCompare 和 Winhex 的使用说明.pdf>>。

2. 第二步，检查新片的坏块信息。如果有坏块，请直接跳到第三步。如果没有坏块，执行到第二步就可以上机焊接。下面是第二步的具体操作过程。

如果新片没有坏块，就直接把从源片读到的 128M.bin 烧录到新片中去。起始地址选择 0x0，**勾上**[写冗余区]，**不要勾上**[跳过坏块读写]。烧写并且校对成功之后，就可以**焊接上机**。



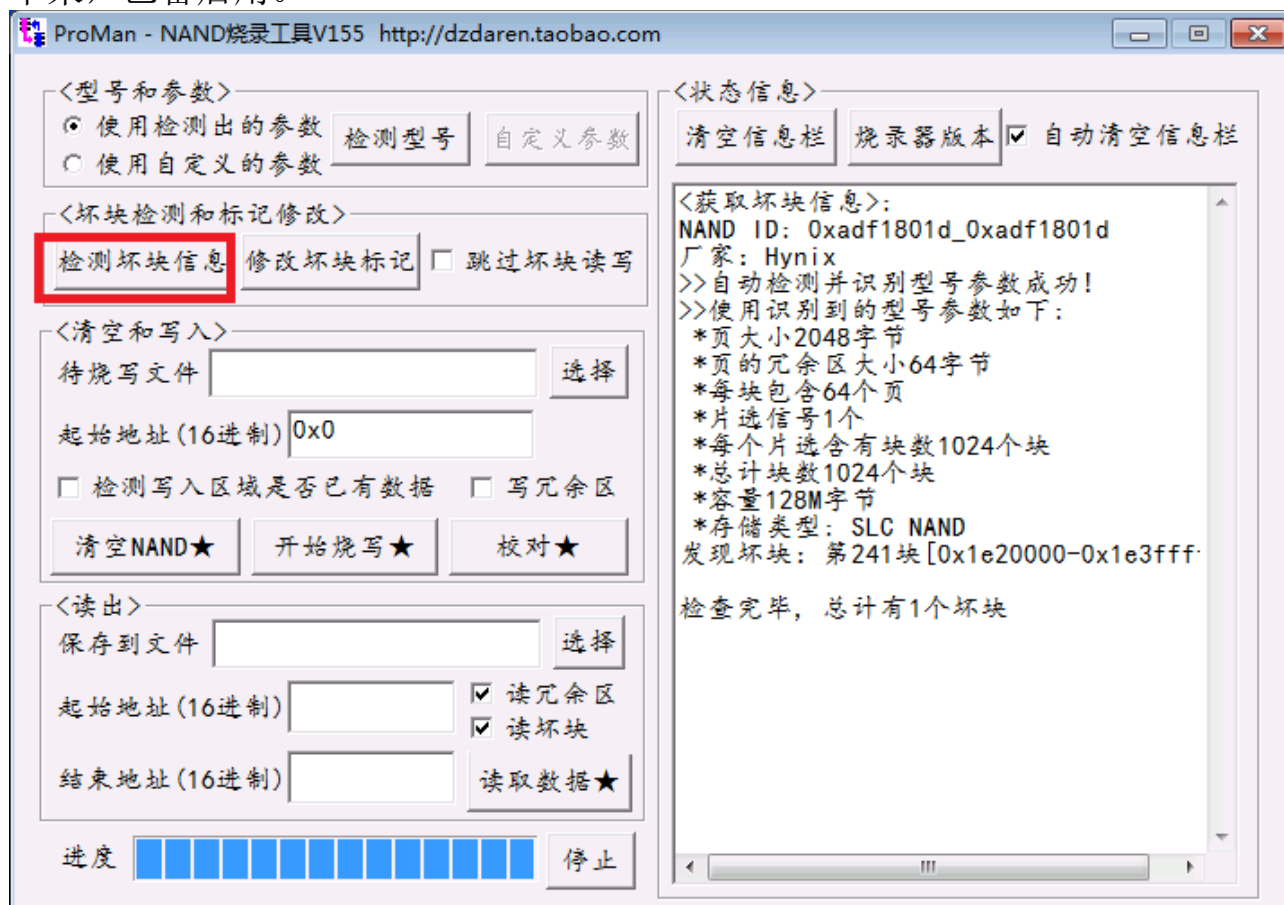
以下 3 点也请注意下：

- 1. 有的芯片的坏块并不是真的坏块，可以尝试下用[修改坏块标记]里面的[恢复为好块]功能来恢复一下。看看能不能恢复为好块。
- 2. **新片和源片的厂家和型号尽量保证是一样的**。这样复制出来的芯片，除了数据之外，ID 也完全一样，实现了完全的物理代替。在找不到同一型号的芯片的时候，才建议考虑其他的兼容型号。
- 3. 目前我们实测的结果，Samsung, Hynix 等大厂的 128MBytes 芯片因为不量产了，所以有坏块的可能性较大，反而 ESMT 家的 128MBytes 芯片 **F59L1G81A** 很少有坏块，所以在替代的时候大家可以考虑这颗芯片。

另外，Micron 的类似型号是量产的，新片的坏块非常少，推荐给大家，128MBytes 的 MT29F1G08ABAEAWP:E，和 256MBytes 的 MT29F2G08ABAEAWP:E。

### 3. 第三步，新片确认有坏块的情况下的分析处理办法之一。

以下图为例，新片是 HY27UF081G2A-TPCB，它与三星的源片兼容，它有一个坏块，是第 241 块，地址范围为[0x1e20000-0x1e3fff]。我们就把这个信息保存下来，以备后用。



此时就是要分析源片数据 **128M.bin** 在新片的坏块(第 241 块)地址范围内的信息是否是关键信息。如果源片在该块内的数据都是 0xFF，那么说明源片在该块内的存储空间并没有实际使用，因此还是可以直接烧录。

用户有两种办法找到第 241 块内的数据。

a. 第一种方法是算出第 241 块的偏移地址是 241 x (页大小 2048+冗余区大小 64) x 块内页数 64 = 32575488, 换算为 16 进制是 0x1F11000 处。然后用 WinHex 直接打开 128M.bin 的文件，并且走到 0x1F11000 处观察数据是否全部为 0xFF。

带冗余区数据的块大小计算公式：(页大小 + 页的冗余区大小) x 每块包含的页数

带冗余区数据的块的偏移地址计算公式：块号 x (页大小 + 页的冗余区大小) x 每块包含的页数

b. 第二种方法是用编程器直接读出源片第 241 块的数据，地址范围为 [0x1e20000-0x1e3fff]，**勾上**[读坏块]，**勾上**[读冗余区]和**不要勾上**[跳过坏块读

写], 然后开始读。然后再用 WinHex 打开读到的数据进行观察。

如果观察下来, 在新片的坏块地址范围内并不需要存放实际的数据, 那么将 128M.bin 烧录到芯片并且校对通过之后, 就可以焊接上机。

#### 4. 第四步, 在新片坏块处需要存放代码和数据。

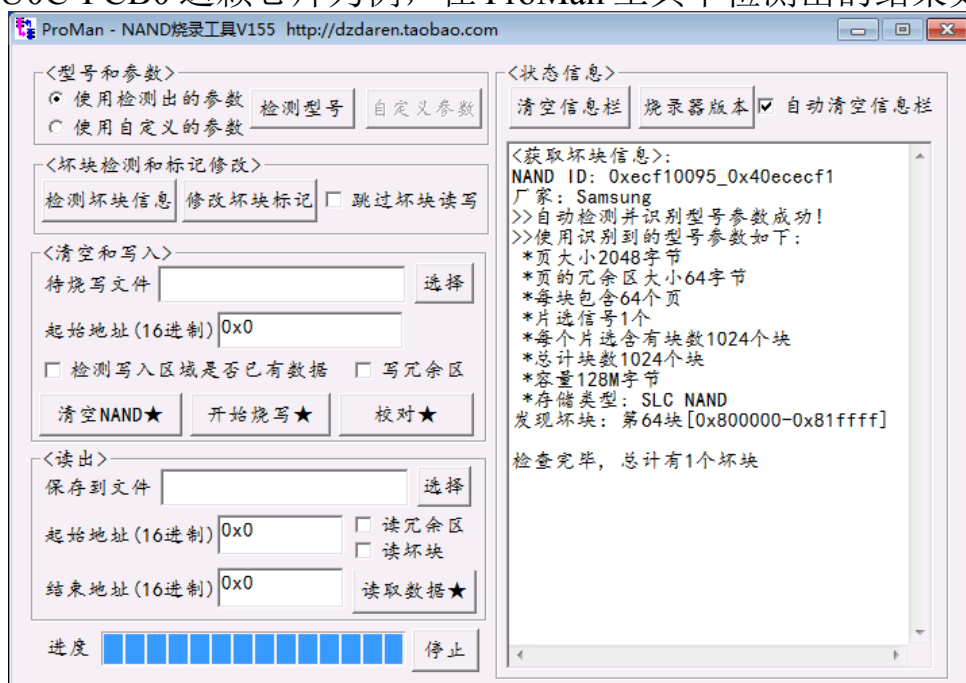
目前暂时还没有好的解决办法, 只能选择避开这一问题, 建议用户改用没有坏块的新片或者是坏块所在地址范围内不需要存放关键数据的新片。

有位朋友曾打了个很恰当的比方: 利用编程器烧录的方式, 有点类似于 Ghost 安装操作系统, 一旦碰到硬盘的坏道处如果需要存放数据的话, 就需要技巧来处理。将来, ProMan 烧录器会尝试针对不同的文件系统, 从软件角度分析并且跳过坏块, 此方法类似于从光盘安装操作系统, 就算有坏块也能跳过。这种方法目前还在开发中, 总之希望能为维修工作带来点方便多尽一份力。(感谢!)

#### 5. 附录:

最后介绍一下坏块的基础知识。

NAND 芯片的存储空间被划分成了一系列的块, 通常一个 NAND 芯片会由几百到几千个块组成, 块大小从 16KBytes 到 4MBytes 都是常见的。以 Samsung 的 K9F1G08U0C-PCB0 这颗芯片为例, 在 ProMan 工具下检测出的结果如下图。



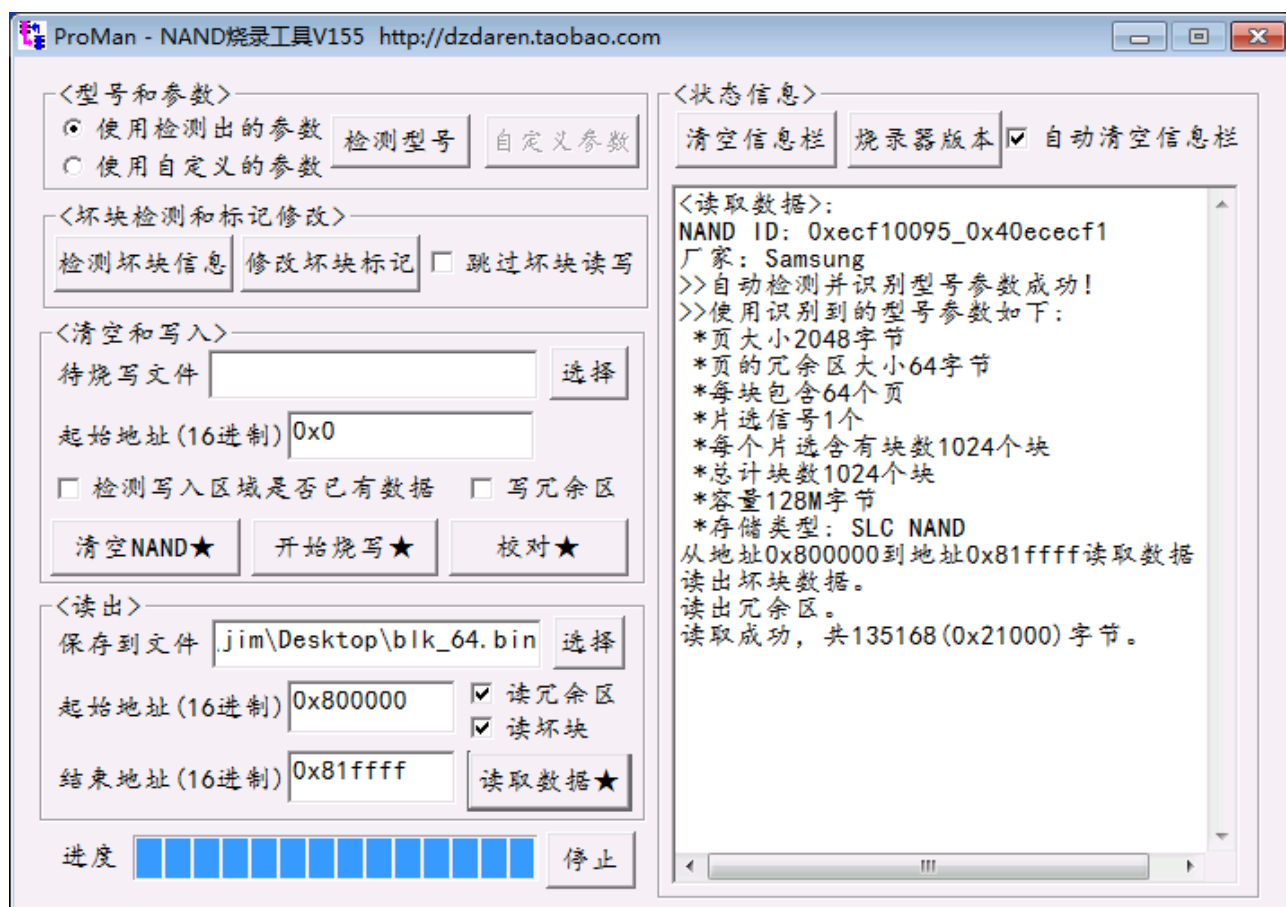
图中可见, K9F1G08U0C-PCB0 由 1024 个块组成, 每个块的大小为  $64 \times 2048 = 128 \text{KBytes}$ , 并且每个块还带有  $64 \times 64 = 4096 \text{Bytes}$  的冗余区。另外, 我们可以发现, 这 1024 个块中的第 64 块被标记为坏块, 地址范围是  $[0x800000-$



0x81ffff]。

一个块被标成坏块的原因有 2 种: (1) 该块无法正确写入数据。如果把该块标成坏块, 以后读写数据的时候通过判断坏块标志就能避开这个坏块, 转而去使用其它好块, 从而避免数据存取的错误。这种情况下, 该块里面的数据是不可靠的。(2) 软件故意把块标成坏块。这可能是为了完成某些特殊的用途, 所以故意把块标成坏块。这种情况下, 虽然表面上看起来是坏块, 但是实际上对这个块的读写和其它的好块是没有差别的。数据依然可读可写, 不会错误。

对于从 NAND 源片提取数据的编程器用户, 因为不知道源片软件开发者的意图, 所以即使是被标为坏块的块里面的数据也要读取出来, 因为这些块有可能是软件故意标成坏块的。是否真的坏块, 只能通过读取源片的坏块内的数据来分析。通常, 如果坏块内的数据都是 0xFF 或者都是 0x00 等, 那里面数据没有用处; 反过来, 如果是一些不规则的数据, 那么有可能这是有用的数据或者代码, 应该把这个块里面的数据当作好块数据处理。总之, 不论坏块读出来的数据是什么, 都要把它们复制到新片里面去。只要保证源片和新片完全一致, 就能开机成功。



ProMan 编程器从 V155 版软件开始，已经支持从坏块读出数据了。只要勾上[读坏块]和取消[跳过坏块读写]，就能把坏块的数据读出来。还是以上面的 K9F1G08U0C-PCB0 为例，我们把数据读出来。因为第 64 个坏块的地址范围是 [0x800000-0x81fff]，所以我们可以读取[0x800000-0x81fff]范围内的数据来分析。因为是带冗余区读，每一页多了冗余区的 64 个 Bytes，整个地址范围多了 0x1000 个 Bytes 的数据，所以最终读了 0x21000 个 Bytes。一旦读完，用户就可以用 WinHex 编辑工具来检查这个文件了。

<本文结束 2014-06-18>