

FC/NES Maker (VK5) 使用说明

The Viking

维京猎人

V1.0.2.1

为了让各路英雄发挥本软件最强性能，本说明将分成三档来解说，各位英雄可以循序渐进。

第一档：“小白学员”。如果你就是一枚小白，对 NES 的制作一无所知，只一心想亲手打造一个 NES，那么你就找对方向了。另外，如果你已经是一枚高手，想用用我这个软件，那么也请先看看这一档的内容。因为不同档的内容不会重复的。显浅的内容将在这一档解说，要是错过了，说不定在浅坑栽筋斗。

第二档：“制作达人”。如果你已经能够用我这软件成功制作出 NES 了，要想自定义图库、更改 Mapper、使用中断、自定义项目模版，那么就从这儿继续吧。本档要求 C 与汇编混合编程。

第三档：“绝世高手”。如果你认为 CC65 都不能满足了，想要连 CC65 都给换掉，我这里也可以提供一个方案。但 V1.0 的方案不太方便，而 V2.0 的方案就好方便的了。

本软件的预览：（可以在两个窗口间切换）



图 1

第一档：“小白学员”

打开"FC/NES Maker"第一件事，就是打开一个工程。

如果从来没有用过它，那就要新建一个工程。

一、操作方法。

(a) 新建工程：

点击，打开主菜单。选择新建工程(N)。于是出现这样的窗口。

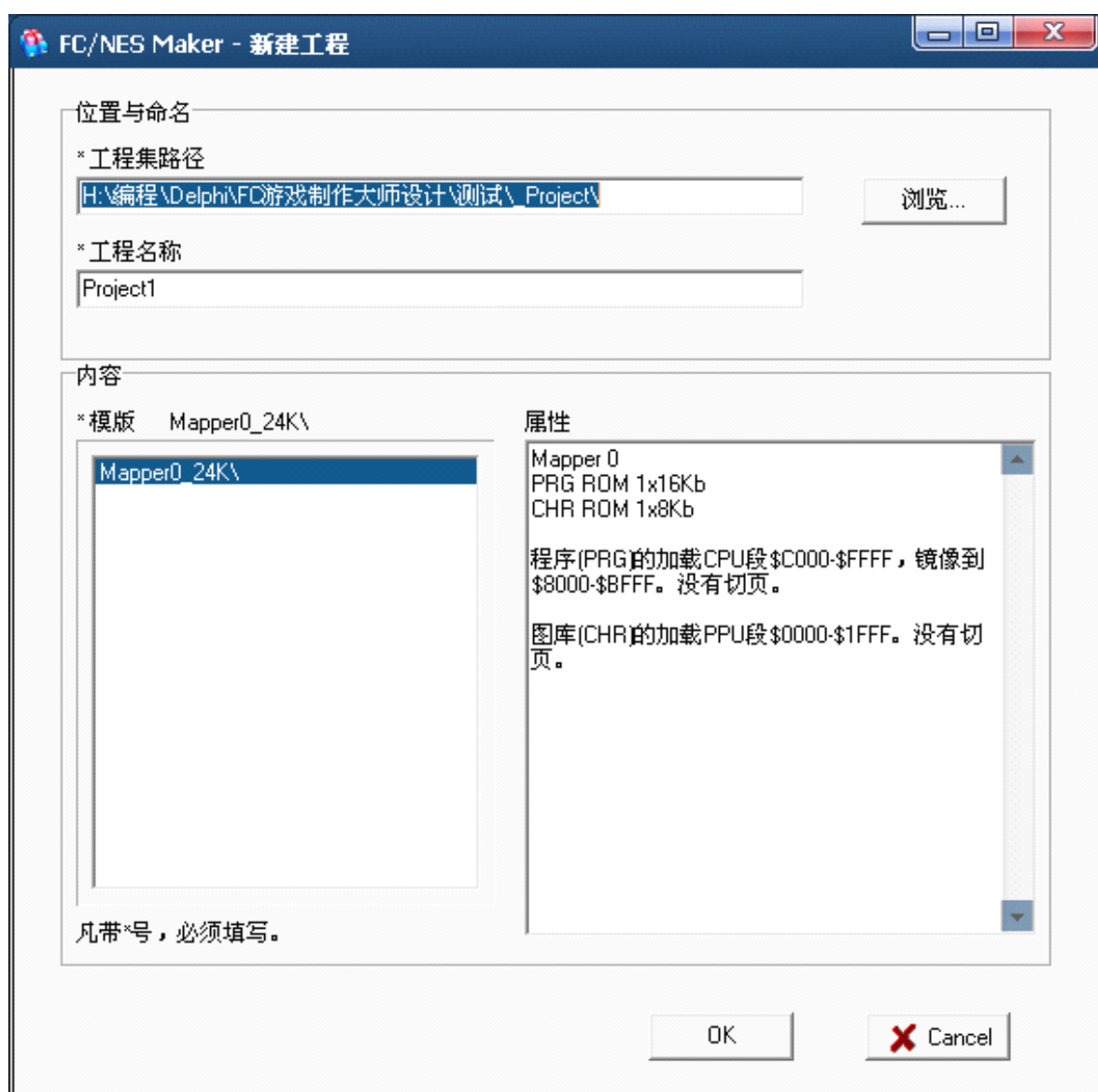


图 2

工程集路径，是说这个工程在硬盘的哪个位置。工程集：就是你说你所建的多个工程都应放在一个根路径下管理。本软件提供一个默认的工程集路径。你只要按“浏览”就可以选定

另一个早已建好的根路径。

每个工程将会在工程集路径下，独立自建一个子路径（子文件夹），存放工程的文件。该子文件夹与工程同名。同时工程的主文件也是工程同名，后缀名为.vk5。

工程名就不要与现有的工程同名了，否则软件会问你是否覆盖。每个相同的文件都问一次。

下面要选一个模版，小白就选 Mapper0_24K 好了。先学这个吧。属性栏是解说模版的各项属性。

最后按"OK"。想取消新建，则按"Cancel"或关闭窗口。我想英文小白也能明白这两个单词吧。狂汗了。.....

如果你是纯小白，对于我上面的话，没有一个是字是不认识的，只是合起来，就没有一句是明白的。那么，我说你做，不用问为什么。

操作：

按默认的工程集路径，打开“我的电脑”查看有什么文件夹（我想就是空白的），于是在工程名称上填一个名称，中文英文都行。只要不与文件夹重名就行。在模版选 Mapper0_24K，果断按下 OK。

那么，至此，你已经建好一个新项目了。

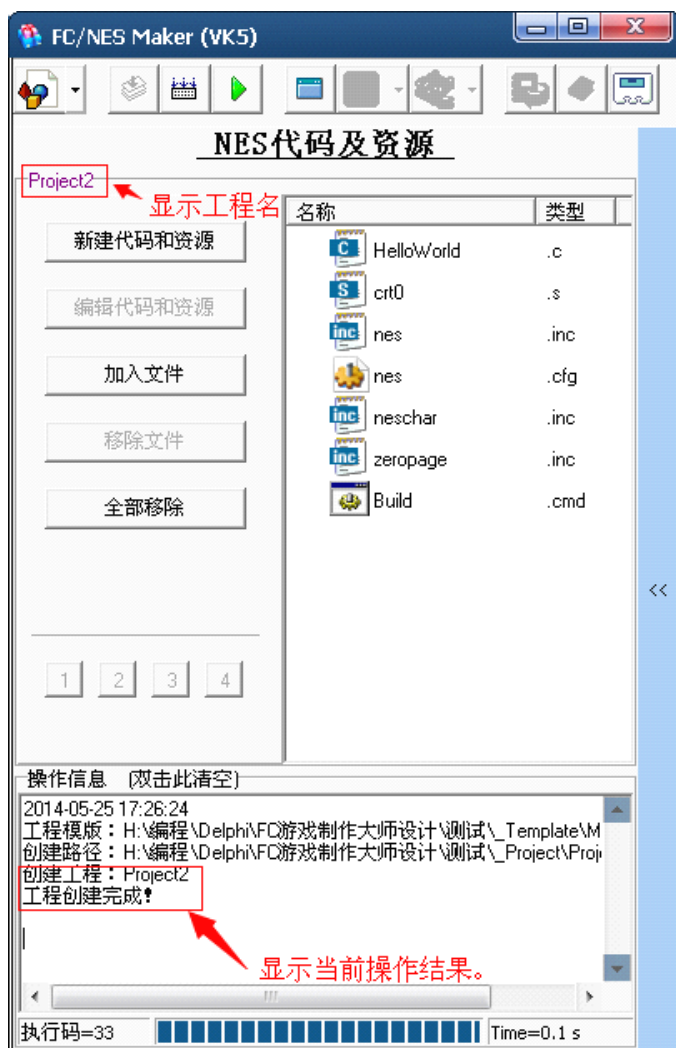


图 3

上图，右侧栏里面的是文件项，即工程中的文件，这里只存放需要编辑的文件即可，不想编辑、不用编辑或不能编辑的文件，都可以不用添加到文件项来。

目前工程文件项不支持子文件夹，就是说，即使在工程文件夹里建了个子文件夹，我们也不能将那子文件夹的文件“直接”添加到文件项里来。

除了工程根文件夹以外的文件，添加到工程中，都是通过拷贝来实现。（包括它的子文件夹）。

但是，它的编译是通过批处理文件来实现的，所以编译可以支持子文件夹。你也能在工程中建立子文件夹来保存其它文件。只是它们不在工程管理中。

关于新建工程，就说到这里。你可能要追问我，为什么模版只有 1 个，太少了。呃！程序刚写好，没空写模版了。到了“制作达人”档，我会介绍如何自定义模版。你就可以自己建模版。或者那时我和别的达人们已经写出了一些模版了。你只要拷贝到“_Template”文件夹中就能用了。大家不要急，我们一步一步来。




(b) 打开工程：


 点击 ，打开主菜单。选择 ，选择硬盘上一个后缀名.vk5 的文件，打开它。截图就跟图 3 一样，只是信息不同，不用多说。

本软件会默认打开程序根路径，默认的工程文件夹是“_Project”，打开它就能找到，你在默认工程集路径下所建的工程。如果你在别的地方建的工程，那么你得自己打开路径了。

(c) 关闭工程：

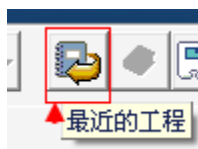
一般，工程操作结束，直接退出或关闭窗口就行。工程是自动自动保存的。关闭工程，是为了打开另一个工程。

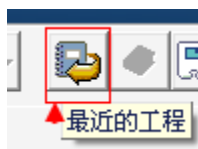

 点击 ，打开主菜单。选择 。

(d) 打开最近的工程：


 点击 ，打开主菜单。选择 

这很有用，因为经常专注于一个工程，一段长时间就只会使用一个工程。那么每次都用它就好了。



另一个更方便的操作：，为了不会误操作，这个按钮在打开工程以后就失效的了。

一个小技巧：如果先前曾打开一个工程，在不关闭工程的情况下，从主菜单，可以打开上次的工程。

当经常使用 A 工程，偶然想查看 B 工程，然后想快速回到 A 工程，可以用这个技巧。

(e) 信息

打开一个工程之后，就可以菜单点击“信息”，查看和修改文件头信息。对于小白，这个参数只能动动“镜像”，别的不要动了。镜像的作为与 NES 密切相关，我以后再讲。而其它信息则在“制作达人”上面讲。

(f) 设置环境变量

我们已经不使用环境变量，假如你安装了 CC65，你可以选择留用这个 CC65，或者卸载。假如你是小白，那么使用旧的 CC65 或我提供的 CC65，那完全没有区别。假如你是达人或高手，想改造 CC65，那么保留环境变量则只有原本安装的 CC65 才有效。（即环境变量指向的程序才有效。）

另外就是高手要想改造我的这个程序，使用另外一些程序，需要环境变量，可以开启环境变量功能。我将在第三档进行说明。

另外，就是试用我的 1.0.0.0 版的网友，（这个版本使用了环境变量），如果想简单处理环境变量的残留，可以找找我，或我远程给你解决。

下面说说“NES 代码及资源”页上面的按钮。

(h) 新建代码和资源



图 4

这个不用多说，就是选一个类型（选什么上面的解说的），然后定义好文件名，确认点 OK。取消点 Cancel。除了解说，底栏的信息留给“绝世高手”使用。小白不用关注。

(i) **编辑代码和资源**，点先一个文件，再点这个按钮就会用相应的软件打开。如果没有对应软件，就是没效果。双击这个文件，也能打开它。

(j) **加入文件**，从硬盘或 U 盘，光盘选一个文件，拷贝到工程中来。

(l) **移除文件**，点先一个文件，再点这个按钮。移除是指从工程文件项中去除，但文件还保存在工程文件夹中。再想不要，得手动删除。

(o) **全部移除**，就是字面意思，清空工程文件项，但文件还在工程文件夹中。

重要的时刻到了，我们点击右侧浅蓝色竖条，可以开启 NES 库文件的大门。小白到这边只是看看罢了，小白不要动这边的文件。看过后，点左侧的浅蓝色竖条，回到“NES 代码及资源”。



选中一个.c 文件，就可以激活它，可以将.c 文件生成.s 文件（即汇编文件）。



每个工程，必须有 Build.cmd 文件，bat 后缀不行，我不想加代码。这个文件可以不在文件项中，但一定要在工程文件夹中。点这个按钮，就会自动运行 Build.cmd，由它来生成 NES。我的模版中包含这个文件，小白不用太担心。



检测与工程同名的 nes 是否存在，如果有，则用模拟器打开。

(s) 命令行窗口

这个只为方便，自动打开路径为工程路径的命令行窗口。调试批处理文件时用得到。

右侧窗口“管理库文件”，暂时不提，这里是小白档，那留到“制作达人”才讲。小白不要删除模块。假如不小心删了，立马退出，再到“_Template\Mapper0_24K”文件夹拷一个 nes.lib 到你的工程中，恢复。

先写到这儿。后面继写小白的注意事项，和生成第一个 NES 的步骤。

二、生成第一个 NES

- 1 新建一个工程，模版就选这个 Mapper0_24K。



- 2 然后点击，出现编译窗口，按任意键（或关闭窗口）。



- 3 再按，OK。你的第一个 NES 运行了。

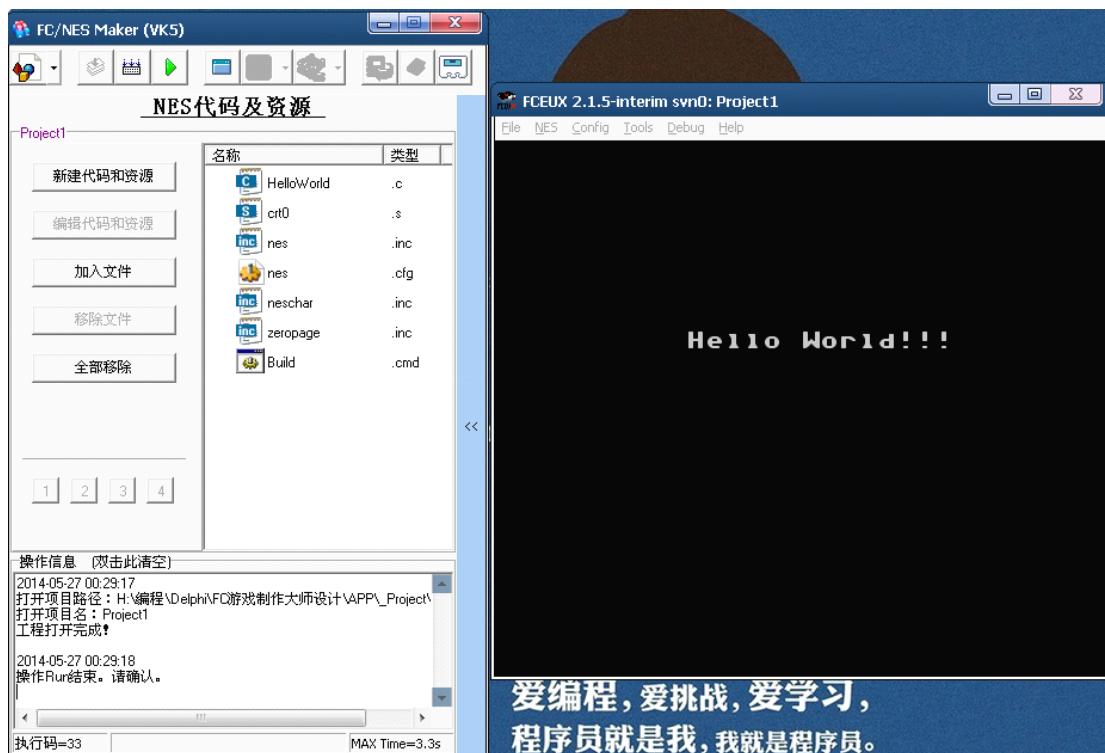


图 5

三、小白的修养。

作为小白也不能是纯小白。要有一技所长才行。

小白至少要认识二进制，16 进制，还有相关运算。这部分不作讲解。应自己找书学习。

为了让小白顺利升级为达人，让小白也享受创作游戏带来的幸福感和成就快乐。

创作 FC 游戏的基本要求：

- A 学习 NES 文件的结构。附卡带的知识。
- B 学习 FC 硬件组成结构。
- C 学习 FC 内部运行流程。
- D 学习基本 C 语言语法。
- E 学习基础游戏运作流程。
- F 学习 6502 汇编的寻址和指令。

以上 6 点，我来一一简单解说。我的解说只能看成是入门的引子。看过之后，要从相关资料和书籍入手学习。以练带学，以学推进。

学会了哪些，记得用小提勾记好，再向没学好的进发。

- 怎么才算学好呢？那是有层次的。你可能只学会了一个层次，却不知下一层次在哪儿。人人都是这样的。
- 写游戏是一个很好的检验。能写到什么程度，就是学到什么层次了。
- 学习总是迂回前进的，不是一样学完再学下一样，也不是一条路学到尾的。一定要东学学西学学、学学停停练练。
- 进步快不得，有顿悟才会有进步。

下面说的这些,是基础中的基础，你也许还不知道如何运用，但至少可以有助你理解我们后面的课程。

(A) NES 文件的结构

我们平时玩 FC 模拟器，运行的"ROM"就是*.nes 文件，这就是我们要讲的。

NES 文件是更准确的名称，"ROM"只是我们常叫的。ROM 的专业意思是只读寄存器。它不存在什么结构。

◆ 各组成部分。

NES 文件是怎么来的？它是来自于 FC/NES 游戏卡带。卡带上有两部分独立的储存体（分程序和图形库，两部分），还有一些逻辑电路。这些都被转换成一个 NES 文件。（这个过程，叫 DUMP）

对应卡带上的程序储存体，NES 文件上面叫 PRG ROM。

对应卡带上的图形库储存体，NES 文件上面叫 CHR ROM。

对应逻辑电路，NES 文件上面叫文件头。

这样再先后次序连结成 NES 文件。次序由低位到高位：文件头 + PRG ROM + CHR ROM

(传说有些 NES 文件里多了一个 TRAINER，当文件头的 TRAINER 值为 YES，则 NES 文件的结构为：文件头 + TRAINER + PRG ROM + CHR ROM 注：TRAINER 的长度为固定的 512。我对 TRAINER 的认识太少了，我还不清楚它的作用。)

下面我们不管 TRAINER，我所见的 NES 文件基本没有这个。那我们只研究没 TRAINER 的情况。

有些 ROM 是没有 CHR ROM 的，即文件头上 CHR ROM 值为 0。那就说这个卡带有 CHR RAM。以后再说这个。这是好东西呀。

◆ 组织与分布。

我们来说说 NES 文件的低位和高位是什么。这意思是指 NES 文件中的地址值，小就是低位，大就是高位。

用 16 进制编辑软件，打开一个 NES 文件，可以看见，前 16 字节就是文件头。这是文件头的固定值。紧接着是 PRG ROM，再紧接着是 CHR ROM。见图 6。

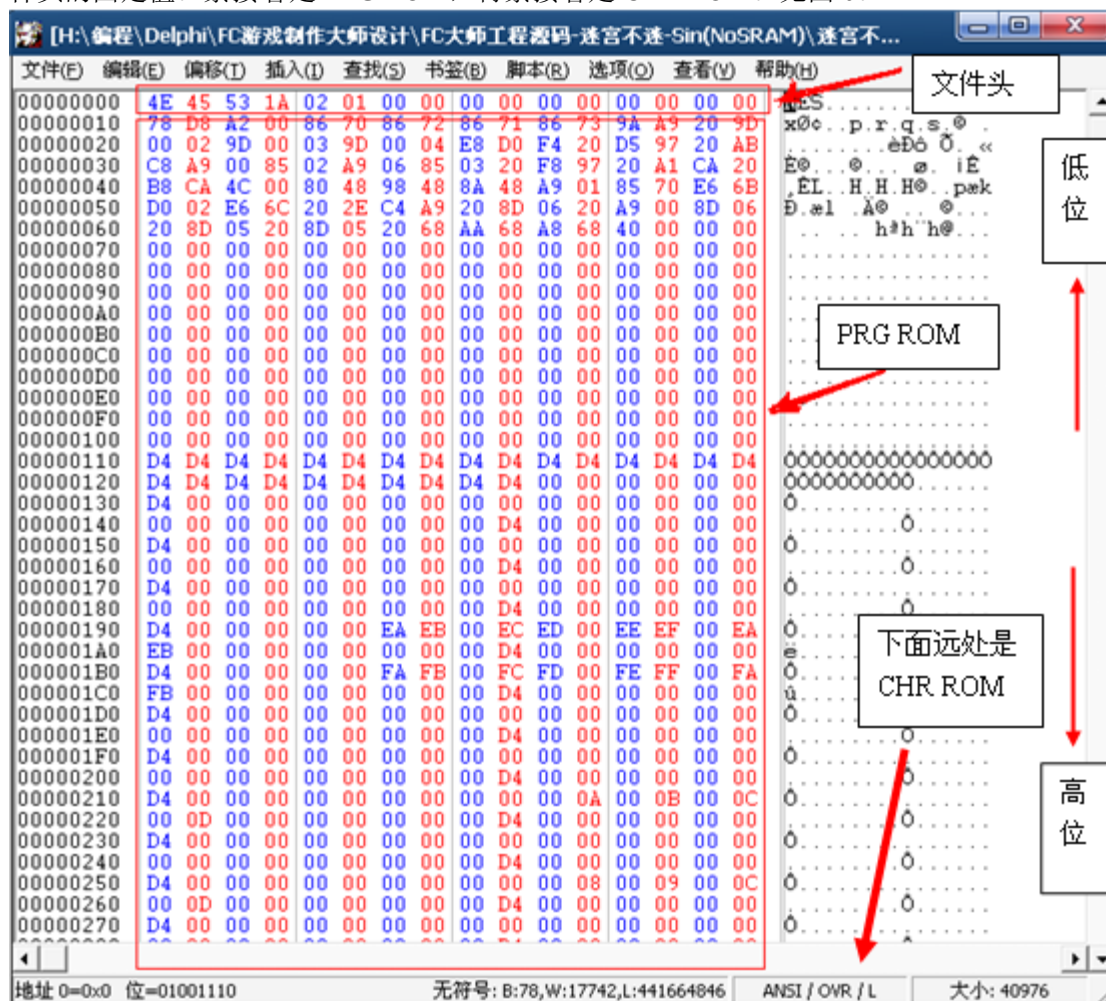


图 6

◆ 关于文件头。

文件头对应逻辑电路，记录逻辑电路的特征，并用序号(Mapper)记录不同的功能电路。
(文件头也记录 PRG ROM 和 CHR ROM 的数量。)

(还有一件有趣的事：一个二进制文件开头的部分用于记录各种信息，这个开头部分叫文件头；而 C 语言的包含文件*.h 叫头文件。嘿嘿名字接近，不要混淆。)

用我的 NESInfo.exe，可以打开 NES 文件，看看文件头的信息。(按"高级..", 就看见"打开"按钮。)见图 7

Mapper、镜像和 4 屏幕就指代了卡带上的逻辑电路。(卡带上的加密电路 GAL 已被 DUMP 人员破解，无视它吧。)这 3 个参数具体作用，后面再说。

S-RAM，指卡带上有一段 RAM (即：易失式可读可写寄存器)，这 RAM 在 NES 文件里不作记录。如果某 NES 文件上这个 S-RAM 值为 YES，那么运行过之后，同一文件夹中产生*.sav 文件。Nes 和 sav 文件一起就能延续这个游戏。S-RAM 用法，后面再说。

(我拆过一些电池记忆的卡带，里面的确是装 SRAM。**SRAM** 意思是静态-易失式可读可写寄存器。网上称为**电池记忆**，这有些不准确。卡带上安装 SRAM 是为了数据可写入，还能保存一些日子，**SRAM 断电则掉失数据**，所以卡带上专为 SRAM 增设了纽扣电池。网上的叫法，因此而来。嘿嘿，原来 S-RAM 有可能不是 SRAM 而是 SaveRAM。即：保存用的 RAM。)

(RAM 是一个统称：易失式可读可写寄存器，基本的分法有动态和静态两种，现在出现更多的种类。动态 RAM 即 DRAM, PPU 芯片里头有它的身影，我们 PC 机的内存就是 DRAM。动态 RAM 只能保存信息几毫秒，它需要不停刷新。但成本低。)

TRAINER 在上页讲过了，不重复。



图 7

我们多看看不同模拟器，看它们能不能打开 NES 文件的文件头。

VirtuaNES 图 8



FC_DEBUG (拿 VirtuaNES 改造的，题标没改。) 图 9



Jnes (纳尼，Jnes 关于各参数的叫法好像更加正规。) 图 10



任天堂红白机模拟器（一打开就大叫“小霸王其乐无穷”） 图 11



我来说明一下，

PRG ROM = PRG Banks

CHR ROM = CHR Banks

镜像 = MIRROR = Mirroring

4 屏幕 = 4 块显存 = EXVARM

S-RAM = 电池记忆

VirtuaNES 上面说的“效验”是哪个位呢，还的各种 CRC 在哪儿呢？我的资料上没说。。。

（CRC 就是效验值。）

了解到这儿就行。大家已经被这种炮轰式教学，累垮了。休息一下把。

下面的教学更花脑筋，我们要开始计算了。

◆ 我们最后来说，如何计算三部分，各自在 NES 文件里面的地址。

文件头，长度是固定的 16 Byte，它固定占据 \$0000 到 \$000F

这个例子中，PRG ROM 有 32KB = 32*1024 Byte，我们计算 CHR ROM 的开头地址。

拿 Windows 附带的计算器，按“查看-科学型”，算算 $32*1024=32768$ ，再按十六进制，得到 8000，再加上 10（这时是在 16 进制的模式下的，\$10 即文件头的长度。）

结果 8010

那么，PRG ROM 占据 \$0010 到 \$800F。从 \$8010 到最后都归 CHR ROM 的了。



图 12

如果文件头中 CHR ROM=0，那就不用算了，除了文件头就是 PRG ROM 的了。

(B) FC 硬件组成结构

FC 的结构就像一台电脑，一台简化版的 PC；它的 CPU，有总线，有 RAM，有 ROM，外部设备有 PPU、APU、手柄。难怪它可以做成电脑学习机，FC 这名字就是家用电脑的意思。更多科普问度娘。

我说几样特别的，作为对资料上的补充：

1. APU，声音合成器。

FC 用的 CPU 是特别订制的，里面包含一个 APU（即多媒体过程单元），其实就是一个声音合成器，其中还包含一个声音解压器。也就是说 FC 的声音（效果音和背景音乐）是合成出来。NES 文件里面不可能包含 wav 或 mp3。FC 的所谓真人发声，就是用声音解压，其压缩采样率极低，解压的模式与 MP3 大为不同。这个 APU 的工作参数比较多，暂时不讲它。

2. PPU 的总线。

PPU 有自己独立的总线，自己的 RAM 和 ROM，它们分别叫 VRAM 和 VROM。PPU 还有一个内部的 SPR-RAM（就是上面讲过的 DRAM），这是专门保存“精灵”的。

VRAM，只有对应 2 个（背景）屏幕的数量，但 FC 有 4 个屏幕的地址空间，VRAM 没有安装的地址就成了镜像，即另外 2 个屏的地址就是镜像性质。镜像屏幕也可以用于显示的。还常常利用这一特性。（**凡说到 VRAM 或 VROM 地址都是指 PPU 的独立地址系统。要与 CPU 的地址系统分开看。FC 里头有 2 套完整地址系统。**）

对应 1 屏幕的 VRAM，由一个命名表和一个属性表组成。

从 \$2000 开始的屏，它的命名表：\$2000-\$23BF，属性表：\$23C0-\$23FF

从 \$2400 开始的屏，它的命名表：\$2400-\$27BF，属性表：\$27C0-\$27FF

从 \$2800 开始的屏，它的命名表：\$2800-\$2BBF，属性表：\$2BC0-\$2BFF

从 \$2C00 开始的屏，它的命名表：\$2C00-\$2FBF，属性表：\$2FC0-\$2FFF

可以认为"\$2000 屏"和"\$2400 屏"（这个简称希望大家能理解）是原本机内带的。但镜像也同样能操作，不过就如同操作本像。即写入 \$2800 跟写入 \$2000 是一样的，它们同时有反应。

那么每两个屏幕有时是左右对称，有时是上下对称。你可以在模拟器通过查看背景来看到它们。它们的镜像方式，是可以通过不同的接法得到的。这个连接线是在卡带上的，所以文件头记录了它。

还有就是所谓的 4 屏幕，或 ExVARM（扩展 VRAM），简单易明，就是多加 2 个屏幕的 VRAM，让 4 个屏幕都的地址都有 RAM，都可以使用了。那么扩展的 2 个屏幕的 VRAM 哪本的？就是卡带上安装的，所以文件头也记录下来。如果是 4 屏幕，那么"\$2800 屏"和"\$2C00 屏"就不是镜像了，他们可以独立使用。

VRAM 还有多出来的 16 个字节，那就是记录色彩的：\$3F00-\$3F1F。

VROM 就是文件头说的 CHR ROM 的低位 8K。因为 VROM 是指连接 PPU 的那 8KB 空间（即：\$0000-\$1FFF），但是 CHR ROM 可以不止 8KB，我们有一个叫法，叫切页。即通过逻辑门的线路变化，令 CHRROM 上超出 8K 的部分与 8K 内的部分作切换。（全部切换或部分切换，看对应 Mapper 的功能）那么 PPU 就能用上了。切页只存在于 Mapper0 以外的 Mapper 号。（又多一个名词，Mapper0 就是最原始的，不具备任何切页和别的特殊功能的卡带电路。）

再说说一种情况，就是 CHR ROM=0，这么说，就是这 8K：\$0000-\$1FFF，在卡带上安装成 RAM，是静态 RAM 了。那么可以叫它是 VRAM 或 CHR RAM。这个 CHR RAM，就没有安装切页功能的了，因为 RAM 好贵，没有厂家这样设计的。卡带 DIY 大师可以给 CHR RAM 设计个切页功能，至少在卡带上，的确可以运行。但我不知道，真的这样做 DUMP 之后会成为怎样的，模拟器认不认之类的。

那么 CHR RAM，怎么使用？CHR 被保存在 PRG ROM 了，可以是压缩的形式，也是为了防盗 CHR。CPU 可以通过 PPU 将 CHR 传到 CHR RAM，那么图形就能被 PPU 使用了。通过 PRG ROM 的切页，也可以实现 CHR 切换了！只是速度慢了。而 CHR 还可以在 CPU 方面重组后再传送。这样 CHR 就灵活了。（你想问我，学习机的输入法是不是用这术，我也不知道呢。）

SPR-RAM，（即：精灵 RAM）没有具体地址，因为与 PPU 的系统也是分离的。实体是包含在 PPU 芯片里面。可以通过 CPU 的\$2003 和\$2004 单个读写，地址为\$00-\$FF，也可以通过 CPU 的\$4014 批量写入。这样理解就好。

3. CPU 的总线。

CPU 的总线连接着 RAM 与 ROM，从 CPU 和总线的角度看，外设是 APU、PPU、手柄、电光枪、键盘等。外设都是通过端口访问的。

CPU 是 8 位的，总线是 16 位的。即 CPU 一次读入 1Byte，地址空间\$0000-\$FFFF（即 64KB）。RAM 只有 2K（从\$0000 到\$07FF）

在\$2000 到\$2008 是一段寄存器。主要用于 PPU。

在\$4000 到\$401F 又是一段寄存器。主要是用于 APU 和手柄，还有一个是传送精灵指令。

如果 S-RAM=YES，在\$6000-\$7FFF 是 S-RAM。这个说过了。

如果 trainer=YES，\$7000-\$71FF 有一个 ROM。我想，trainer 和 S-RAM 并存的机会好低。真的在一起则 S-RAM 会不会少一半空间。矛盾。不钻这个牛角尖。

\$8000-\$BFFF，我给一个名，叫切页 Bank，因为它多用于切页。

\$C000-\$FFFF，我给的名，叫常驻 Bank。因为较少用于切页。这是我定义的，非官方。

注：我们说说 Mapper0，有多少种容量。

1. 16K，即 PRG ROM=16K；CHR ROM=0，PRG ROM 是加载到\$C000-\$FFFF
2. 24K，即 PRG ROM=16K；CHR ROM=8K，PRG ROM 是加载到\$C000-\$FFFF
3. 32K，即 PRG ROM=32K；CHR ROM=0，PRG ROM 是加载到\$8000-\$FFFF
4. 40K，即 PRG ROM=16K；CHR ROM=8K，PRG ROM 是加载到\$8000-\$FFFF

最后贴个我自己想出来的 FC 结构，也是非官方。

《任天堂产品系统文件》、《任天堂红白机（NES）文档》

《任天堂游戏编程解密》、《电脑游戏机硬件与编程特技_清浙版(FBASIC).pdf》

这些书有详细解说。

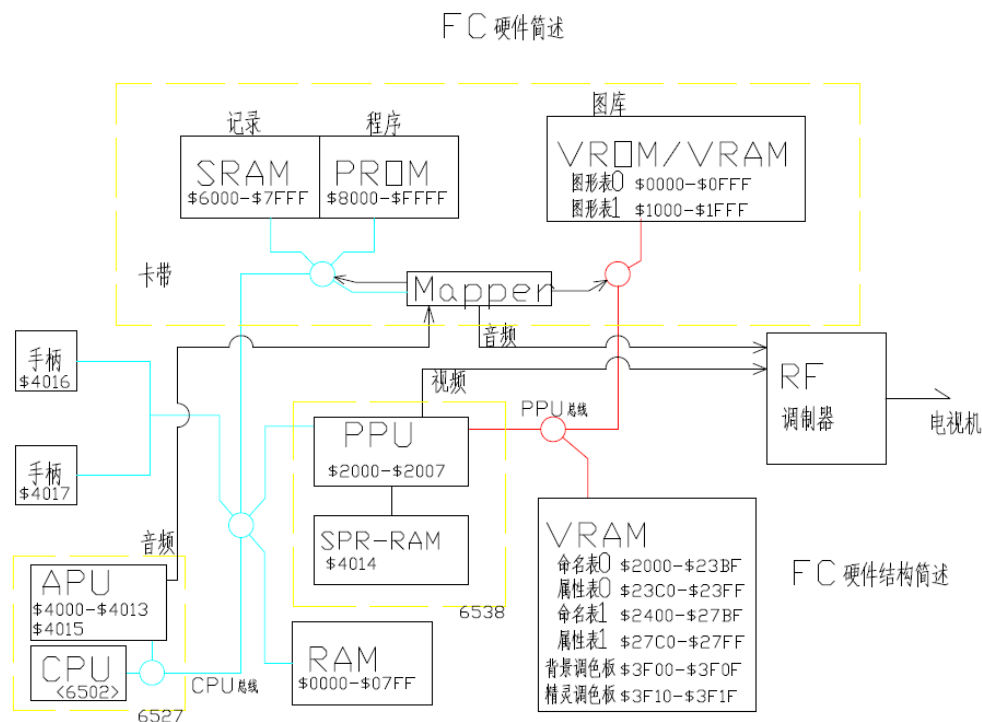


图 13

(C) FC 内部运行流程

上电，就自动复位，CPU 从\$FFFC-\$FFFD 读取一个地址。并作为启动地址。

下面这是从模拟器的内存视窗，我们找\$FFFC 和\$FFFD，找到 00 80。这两个数前一个是低位，后一个是高位，那么组合成一个地址是\$8000。这个地址值在复位后，加载到 CPU 的 PC 寄存器，则从这个地址开始运行。这个不一定是固定\$8000，可以是\$8000 到\$FFF7 的任一个地址。这\$FFFC-\$FFFD，我们叫它做"RESET 向量"。

ADDR	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	0123456789ABC
FF60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FF70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FF80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FF90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FFA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FFB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FFC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FFD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FFE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FFF0	00	00	00	00	5C	80	5C	80	5C	80	35	80	00	80	5C	80 ¥. ¥. ¥. 5...

图 14

运行的时候，可以通过 CPU 指令禁止和允许 IRQ 中断在响应，IRQ 中断信号从卡带而来。有的 Mapper 可以让 IRQ 在每条扫描线结束时启动。关于 IRQ，信息少，不确定，不会用最好禁了。

我们重点说说 NMI。NMI 比 IRQ 高级，NMI 不可禁。但没有书籍正面的说出优先级，即在 IRQ 被允许的情况下，当 NMI 中断运行过程中，IRQ 被触发，会怎样？或 IRQ 中断运

行过程中，NMI 被触发会怎样。书上都没说。我暂时保留。

NMI 是连接着 PPU，由 PPU 发信号触发 NMI 中断。CPU 虽然不能通过自身指令禁止 NMI 中断的响应，但是因为 FC 的设计，可以通过 \$2000 的 D7 位，要求 PPU 是否发来中断信号。从而达到控制 NMI 的效果。

若响应 IRQ 中断，则 CPU 将 PC 寄存器进栈，读取 \$FFFE-\$FFFF 的地址值，并加载到 PC 寄存器，则从这个新地址开始运行。直到 RTI 指令，才将中断前的地址值出栈，并继续运行。

若响应 NMI 中断，则 CPU 将 PC 寄存器进栈，读取 \$FFFA-\$FFFB 的地址值，并加载到 PC 寄存器，则从这个新地址开始运行。直到 RTI 指令，才将中断前的地址值出栈，并继续运行。

不管是 IRQ 或是 NMI，都没有自动完成保护现场的工作，要求在代码中完成保护现场和恢复现场的工作。

FC 的 CPU 工作没有 end 的，所以代码也不能写 END，只能是进入循环。这个概念在以后的程序设计中，非常有用。

以上是 FC 的 CPU 工作流程。具体游戏的流程，要看代码的编写，但以上的内容，不管代码如何编写都会按规则工作的。因为这是硬件定义的。上面的部分比较深，好多名词小白可能没听过，但一些 6502 老手一看就明了。这也好难去深入浅出。能看多少算多少。以后领悟了，回来再看。或是互相学习，互助进步。这部分是基础，但要到混合编程时才能真正用上。这里能明白一点的话，在别处能够明白好多的。

(D) 基本 C 语言语法

据说 C 语言有 C89 标准，又有 C99 标准，目前我还不知道 CC65 是按哪一个标准，还是都不是很严谨。

我们先推荐几本书，小白先学点语法，慢慢研究。

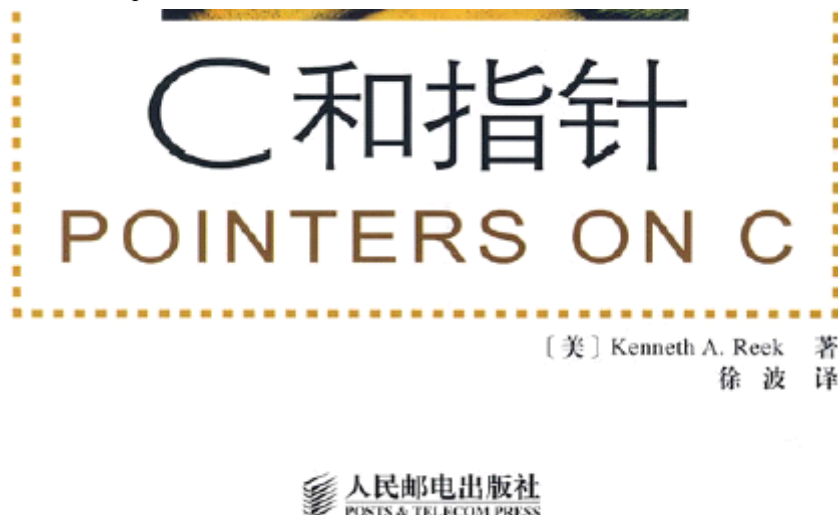
《C 程序设计语言_第二版·新版.pdf》机械工业出版社



ISBN 7-111-12806-0/TP - 2869

这本书重基础

《C 和指针.pdf》



提高，应学习指针，在混合编程，一定要用指针。

《C_Primer_Plus（第五版）中文版.pdf》

C Primer Plus（第五版）中文版

[美] Stephen Prata 著

云巅工作室 译

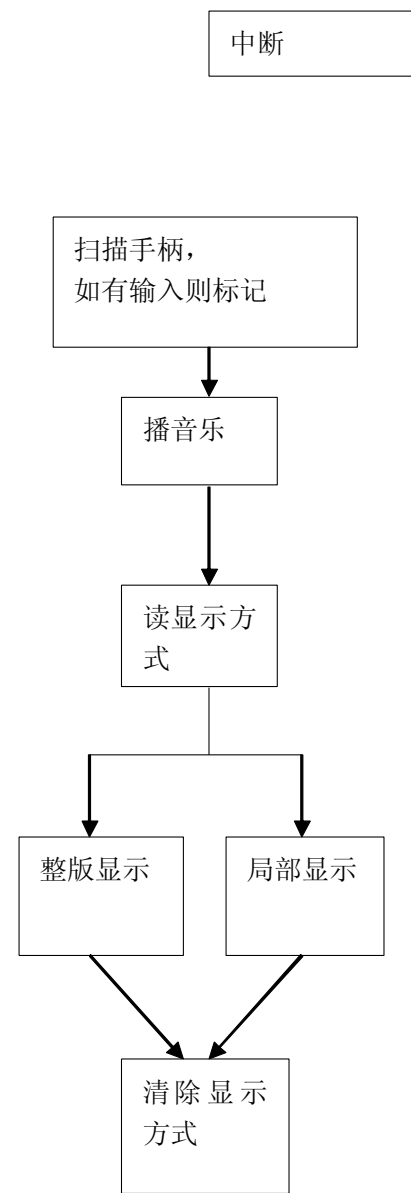
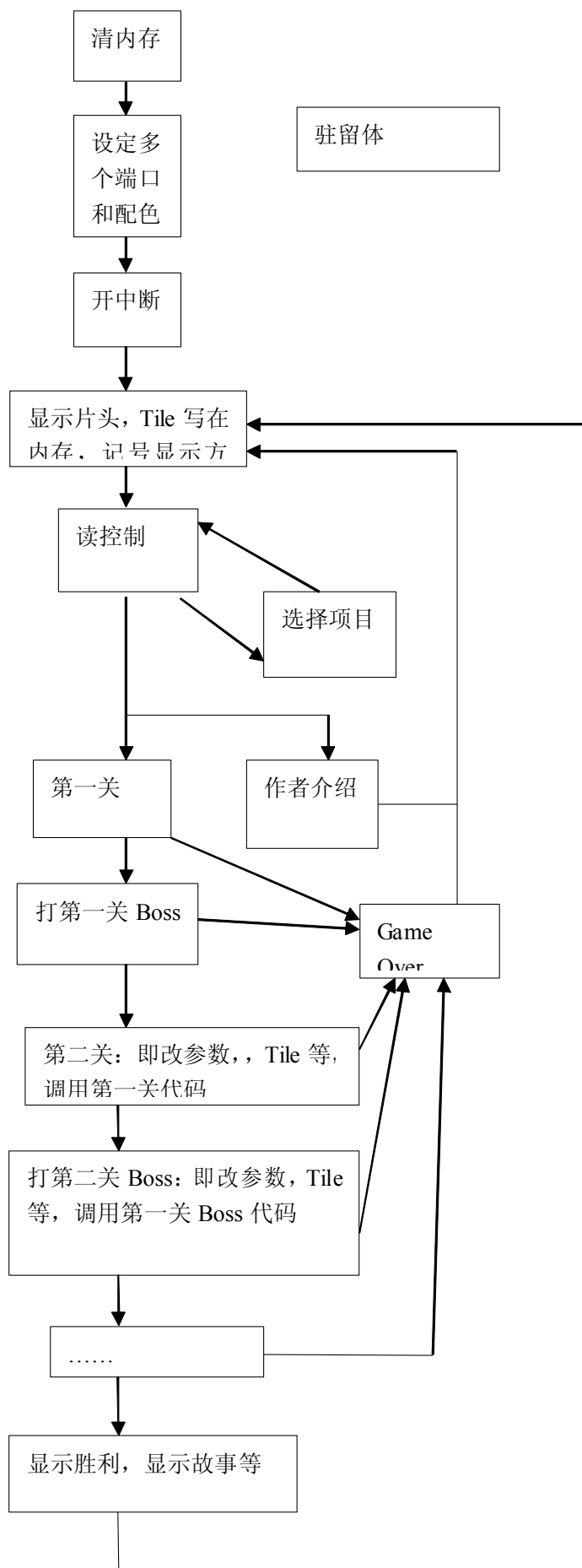
这本是备选的。

（E）基础游戏运作流程。

应以服务-客户的方式来写这个程序。一些基础信号，例如手柄、音乐、PPU 缓冲应当在 NMI 中断完成。一些特别效果用 IRQ+NMI 实现，但不能太多代码。几能是十来行。主程序与中断交流，要设一些公共变量。中断是服务方，主程序是客户方。

这个又有些难明了，我又做成图形方式，大家慢慢看。

如果有多个关，每一关的背景、数据都有不同，可以几关放在一页，用切页来控制。主程序做成不变，只要每关的数据不同。切页一般是 \$8000-\$BFFF，驻留是 \$C000-\$FFFF。中断的代码也应写在驻留的部分。



手柄操作：驻留体测到有输入标记，则按手柄值操作，完了则清除输入标记。

显示：设有显示方式标记。（分“整版”和“局部”）

1 整版显示：画面 Tile 先写入内存。在中断内全部写入 PPU。

2 局部显示：以这个格式“数量，低位地址，高位地址，数据，数据，数据”记录。在

(F) **6502 汇编的寻址和指令。**

这个也是看书好了。

特别的说的，是寻址是汇编的灵魂。不同的 CPU 有自己的汇编。同一系列的 CPU 有兼容的汇编指令，个别之间也有小区别。6502 汇编是指 6502 系列的 CPU 的兼容汇编。

《6502 编程大奥秘》前 3 章适合 FC，后面是文曲星独有的。这书写得简单明了，概括。非常好的入门书。

《6502 微处理机及其应用.PDF》上面的指令功能，与机械码，还有运行时间，是绝对正确的。别的书真有错漏。

《学习机 6502 汇编语言-有标签.pdf》好书，正正规规的书。还的子程序解说。

小白档先说到这儿，完。

其它从一些小工程，里面附解说，一个例子进行解说。

修订：

关于：“我们最后来说说，如何计算三部分，各自在 NES 文件里面的地址。”

原文：

◆ 我们最后来说说，如何计算三部分，各自在 NES 文件里面的地址。

。。。

拿 Windows 附带的计算器，按"查看-科学型"，算算 $32*1024=32768$ ，再按十六进制，得到 8000，再加上 10（这时是在 16 进制的模式下的，**\$10 即文件头的长度+1。**）

结果 8010

修改：

◆ 我们最后来说说，如何计算三部分，各自在 NES 文件里面的地址。

。。。

拿 Windows 附带的计算器，按"查看-科学型"，算算 $32*1024=32768$ ，再按十六进制，得到 8000，再加上 10（这时是在 16 进制的模式下的，**\$10 即文件头的长度。**）

结果 8010