

1. 编程器概述

1.1 编程器性能介绍

精心打造的廉价的专业型编程器，高密度贴片工艺制作，统一的操作界面，使用方便，功能完整，应用程序运行可靠，代码超小，执行速度更快，支持中英双语。可运行的操作系统：

WINXP WIN2003 WIN2008 VISTA WIN7 WIN8 WIN10 32 位或 64 位操作系统

- 编程速度大幅提升.完全适合小批量生产，支持一台电脑拖 4 个编程器。
- 宽范围的芯片支持，初始版本支持芯片 15000+，支持各种 24 25 26 27 28 29 37 39 49 50 系列串并行存储芯片，单片机支 51 系列，ATMEL AVR ATMEGA AT90 系列，MICROCHIP PIC10 PIC12 PIC16 PIC18 系列单片机，GAL 可编程逻辑器件编程，SRAM 测试，54/74 及 CMOS4000 系列集成电路综合功能测试。
支持 NAND Flash 芯片，功能完整全面，容量支持至 8Gbits。
25 系列 Nor Flash 容量支持到 512Mbits。
- 引脚接触检查功能，不良接触引脚能精确定位
- 超低功耗：整个编程器全部采用超低功耗芯片，采用 USB 接口供电，第一台真正做到所有芯片编程不用外接电源的综合编程器。便携性能超强：体积全球最小的一个，外型尺寸 10CM*6CM*2.5CM，只有驾驶证大小。
- 可编程 VCC 从 1.8V 到 6.5V，VPP 从 1.8V 到 18V 的芯片
- 芯片序列号二次开发功能：编程器内置了常用的多种自动编号功能。并可使用自定义算法，通过 DLL 动态库实现任意的序列号。该功能可以使你在批量生产时，每个芯片都产生一个唯一的标识信息。通过自定义算法还可对该标识进行加密，从而更有效保护你的产品权益。
- 万能串行编程接口 ICSP（使同时支持高压并行编程与串行低压编程）。注：ICSP 现支持 24 系列 25 系列 93 系列串行集成电路，ATMEL89S51、52，AVR ATMEGAxxx 全系列及 MICROCHIP PIC10Fxxx 12Fxxx 16Fxxx 18Fxxx 全系列，及新茂 SYNCMOS SM59Dxx SM59Rxx 全系列芯片进行在线串行编程。
- 性能超强的 54/74F/LS/HC CMOS4000 系列集成电路综合功能测试：真正可测试常见逻辑集成电路的第一个开发型编程器，测试可定位到逻辑门电路的错误，测试集成电路的任意可能出现的输入组合。

1.2 编程器芯片支持范围

- 各种 26 27 28 29 37 39 49 50 系列并行 ROM、EPROM、EEPROM
速度快，支持全面。同时支持 TSOP32 TSOP40 TSOP48 的芯片到 64Mbits，将加入对 TSOP56 引脚的支持。
（TL866II 编程电压 VPP 只支持到最高 18V，对早期的 27C 系列 ROM，用 21V 25V VPP 电压的芯片不再支持写入，但可以读取，如果用到，可用同型号的编程电压 VPP 为 13V 或 12V 的芯片替代，例：27C32 有 21V 13V 芯片购买时注意编程电压）
- 24 25 35 45 85 93 95 系列串行 EEPROM
串行系列芯片支持全面，支持芯片容量 25 系列 8 16PIN 到 512Mbits（64M 字节）

- NAND FLASH 芯片支持到 8Gbits
- 单片机系列，上百个种型号的 51 系列单片机
- AVR 单片机，包刮 ATMEGA 系列 ATtiny 系列 AT90XXX 系列单片机，所有的 AVR ATMEGAxxx 芯片同时支持用 ICSP 接口进行 SPI 串行下载编程。支持 AVR 单片机用 RC 校正字节进行一对一软加密。
- MICROCHIP PIC10 PIC12 PIC16 PIC18 系列单片机：
PIC 系列 300 多个型号，900 个封装以上的芯片支持，目前支持 PIC 芯片最全的综合编程器之一。对各种型号的 PIC 单片，严格按厂家手册要求，进行最优化的编程，编程速度优秀。极大部分芯片同时支持 ISCP 在线编程及通过编程插座进行编程。
- GAL 可编程逻辑器件编程
- SRAM 测试
支持常见 24 61 62 DS12 系列 SRAM 测试，分别可以以数据线测试、地址线测试、内存单元增量测试及单元测试。
- 性能超强的 54/74F/LS/HC CMOS4000 系列集成电路综合功能测试：
真正可测试常见逻辑集成电路的第一个开发型编程器，测试可定位到逻辑门电路的错误，测试时，测试集成电路的任意可能出现的输入组合。

2. 软件安装方法

2.1.1 下载并安装应用软件

64 位操作系统安装时，以管理员身份运行 EXE 安装包。

从 <http://www.xgecu.com> 下载 Xgpro_setup.exe 文件，双击该文件进入安装程序界面。如下：（安装文件内包含了本说明书的内容） 点击[安装]开始安装应用程序，弹出如下画面：

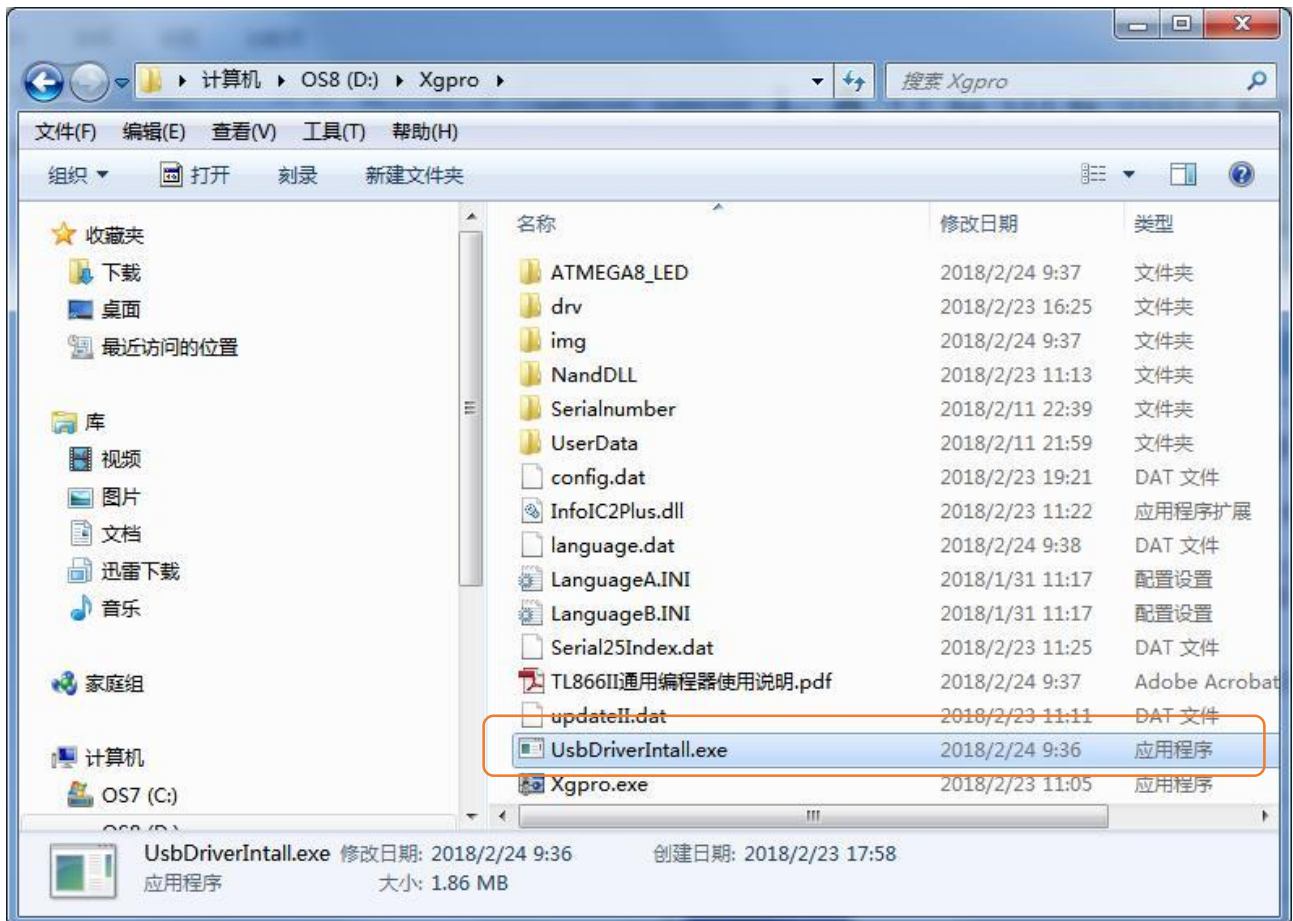
注：在 WIN7 64 位操作系统安装时，以鼠标右键点击 XgproVxxx_setup.exe 文件，选择以管理员身份运行。 在这里输入你想安装的的目录。点击安装开始应用程序的安装。



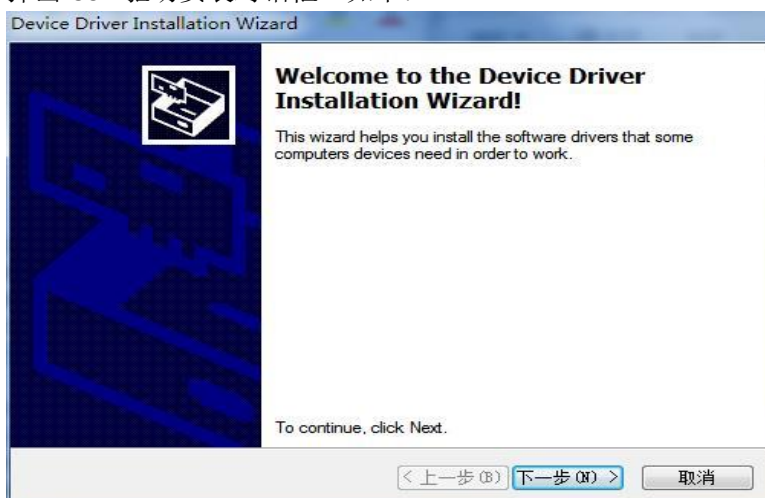
点击安装，直到完成。

2.1.2 安装 USB 驱动

在安装目录下（D:\xgpro）找到 UsbDriverInstall.exe 文件，双击运行



弹出 USB 驱动安装对话框。如下：



下一步：



USB 驱动安装完成。

2.1.3 进行设备安装

完成以上安装后，这时插入编程器到 USB 口，操作系统将发现硬件，如下：



选择[自动安装软件]，点击下一步，windows 将自动安装，直到成功完成安装。

（注意：在这过程中，驱动程序会更改注册表，如果装了防火墙软件，请放行本软件，windows 还会提示驱动程序未经数字签证，确认后继续安装）。

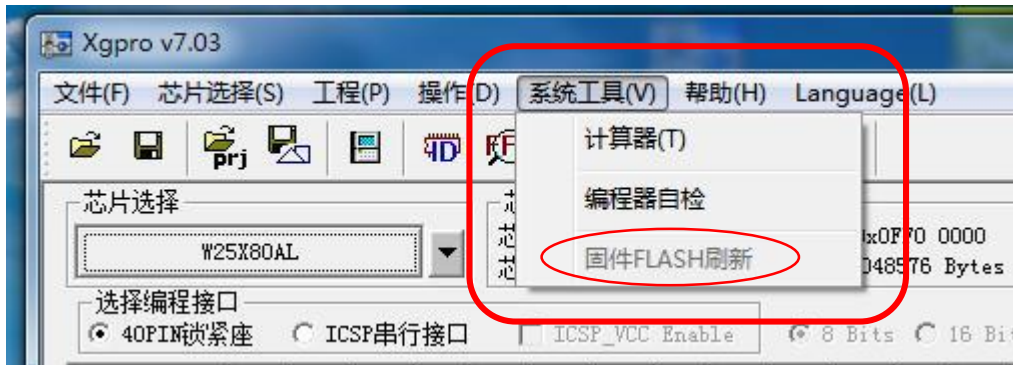
如果自动安装不成功，也可以选择<从列表或指定位置安装>，USB 驱动文件在安装目录（D:\xgpro\drv）



下。安装完成，打开桌面的上运行编程器软件。即可使用。

2.1.4 固件 FLASH 刷新

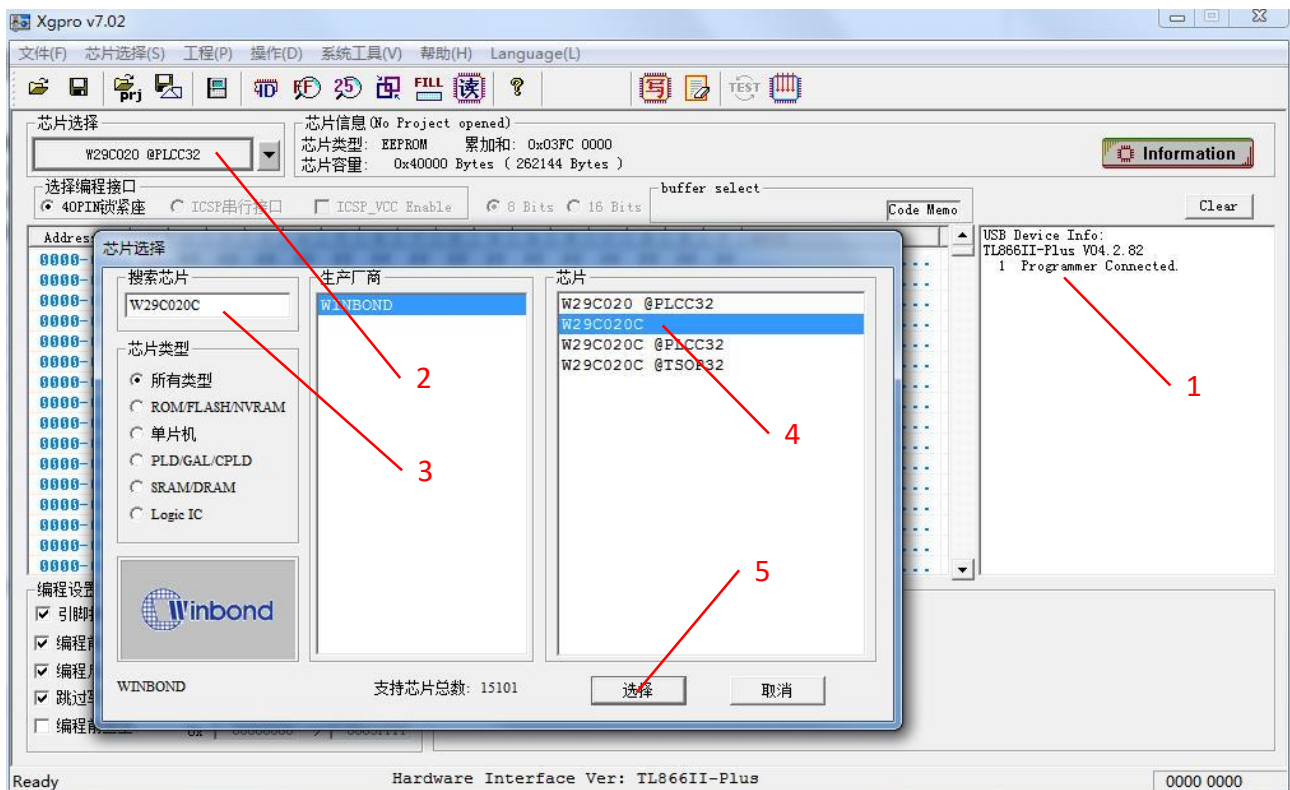
首次使用或升级时可能需要进行固件刷新，在软件的主菜单，进行一次刷新操作，如下图：



3. 快速入门

3.1 硬件准备并选择正确的芯片型号

在您确保正确安装了通用编程器应用软件，以及 USB 驱动程序之后，连接好 USB 通讯线。此时编程器的红色电源指示灯亮指示连接正常(点亮状态)，黄色指示灯熄灭等待编程（如果黄色指示灯一值闪烁，表示 USB 驱动安装错误，或没有安装 USB 编程器设备驱动），即可进入下一步。

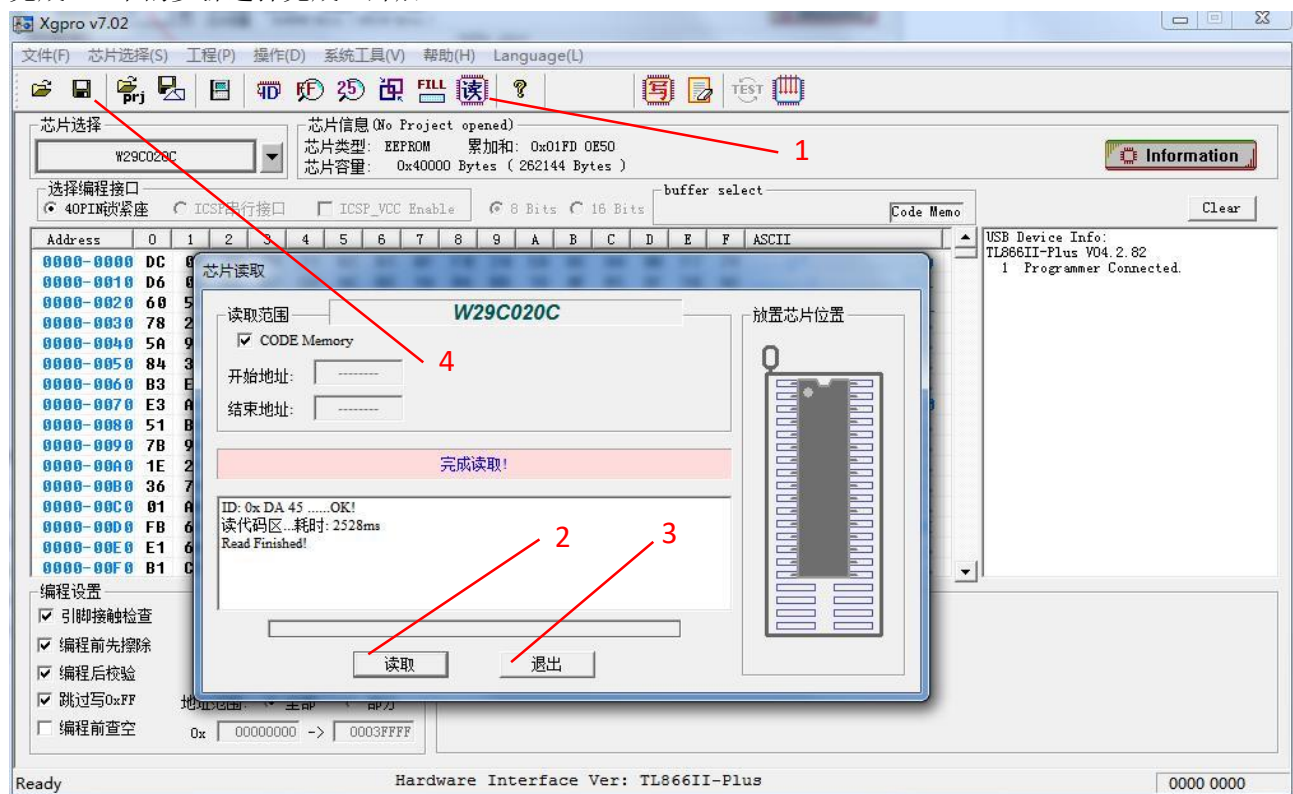


1、 1 Programmer Connected 表示编程器已经连接上

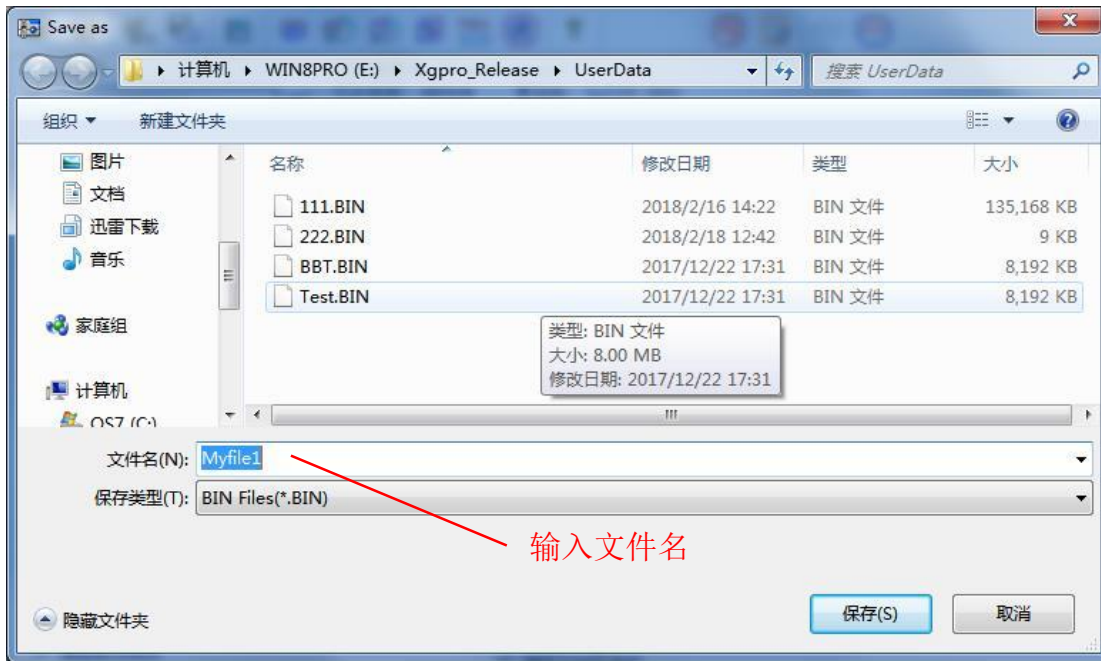
- 2、 点击芯片选择这个按钮，弹出上图芯片选择对话框，
- 3、 在搜索芯片处，输入要读写的芯片型号（本例 W29C020C）
- 4、 点击正确的型号 注：@PLCC32 @TSOP32 指芯片封装 无@标注指 DIP 封装
- 5、 选择确定

3.2 读取芯片内容并保存文件

完成 3.1 节的步骤选择完成芯片后

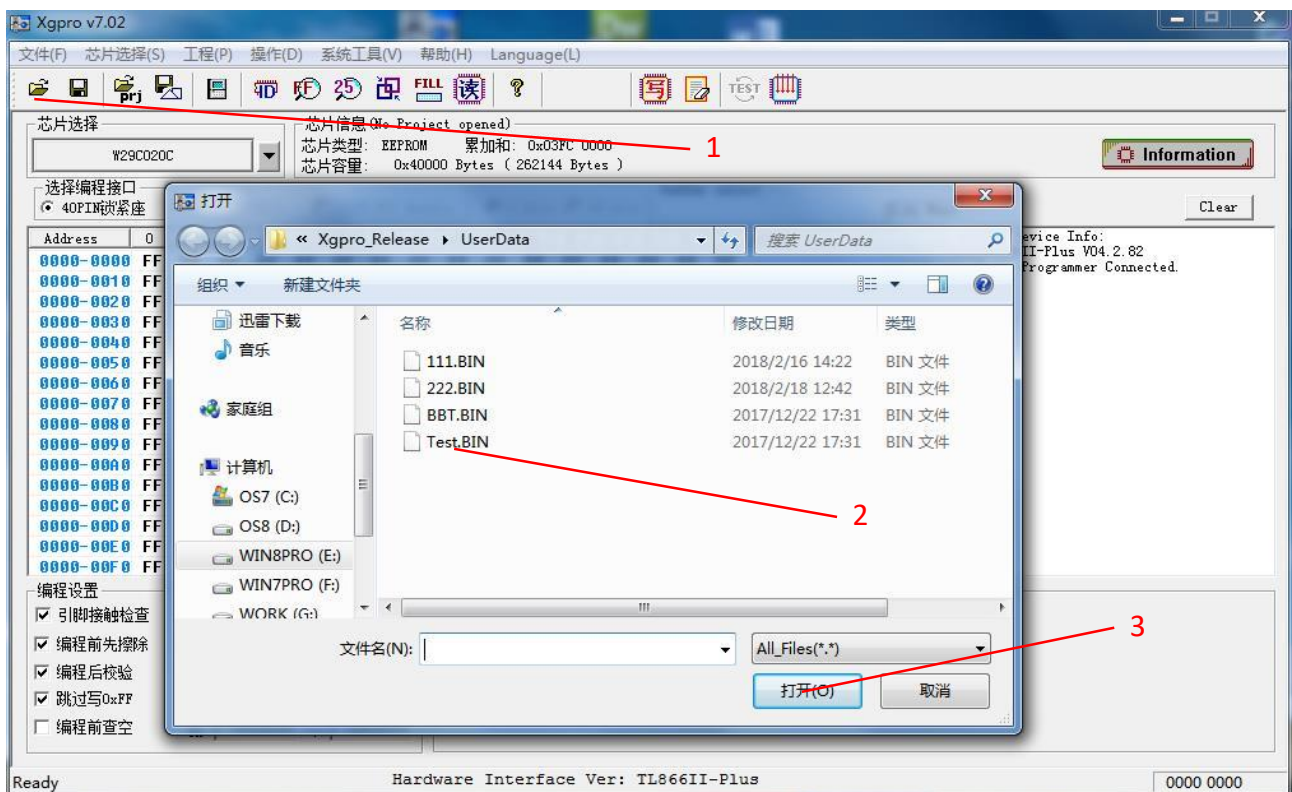


- 1、 点击上图工具栏中的【读】按钮，弹出读对话框。
- 2、 在读对话框中，点击【读取】按钮。
- 3、 读取完成后，， 点击【退出】按钮。
- 4、 点击保存文件 （也可在菜单<文件>中选择<保存文件>），如下图输入文件名后，保存文件。



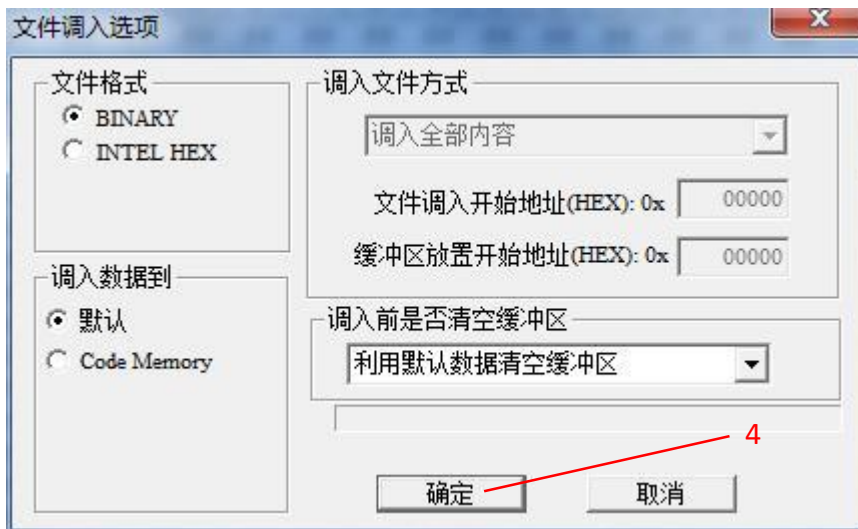
3.3 将文件内容烧录到芯片

完成 3.1 节的步骤选择完成芯片后

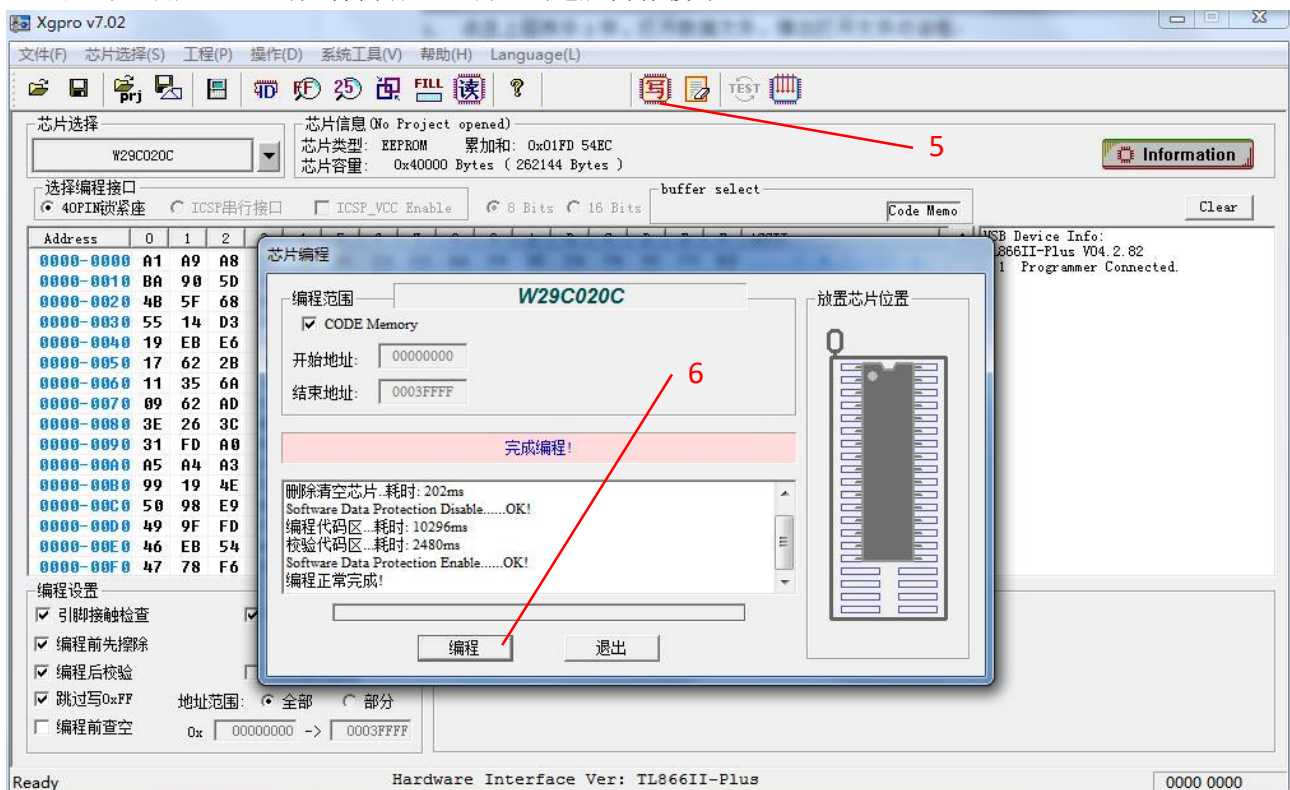


1. 点击上图所示 1 中，打开数据文件，弹出打开文件对话框
2. 选择要打开的数据文件（Test.BIN）

3. 点击【打开】，弹出以下调入文件选项对话框，按默认设置，点击【确定】即可。



4. 点击【确定】，此时文件内容已经调入到电脑内存缓冲区。



5. 点击【写】，弹出芯片编程对话框。

6. 点击【编程】，开始对芯片进行烧录，烧录时没有出现错误提示，即烧录完成。

7. 如果需烧录更多芯片，换上新的芯片后，直接点击【编程】开始新的烧录。

4. 编程器功能详解

4.1 主菜单总汇

菜单命令总汇	
文件菜单	打开、保存、缓存操作功能、退出
芯片选择菜单	按菜单列表选择被编程芯片、25 自动识别
工程菜单	打开工程、保存工程、工程另存为、关闭工程、工程属性、工程密码更改
操作菜单	读取芯片内容、芯片 ID 设别、校验芯片、编程芯片、擦除芯片、查空、自动编号设定、测试, NAND 坏块检查
系统工具菜单	计算器、编程器升级检查、编程器自检
帮助菜单	帮助文件、关于
语言菜单	选择语言
编辑缓冲区	在缓存中查找对应的 16 进制值或 ASCII 码字符串，数据，拷贝、复制、粘贴、填充、部分另存为 TXT 文件

4.1.1 文件菜单

文件菜单项内容如下：



打开文件:

装入 16 进制文件或二进制文件到指定的缓冲区，GAL 芯片时为装载 JED 格式的文件。
NAND FLASH 操作时，打开文件，调入到 BBT 缓冲区。

保存到文件

将当前缓存保存到 16 进制格式的 HEX 文件或 2 进制格式的 BIN 文件，GAL 芯片时保存为 JED 格式文件。
NAND FLASH 操作时，保存文件，是保存 BBT 缓冲区。

注意:对于 MICROCHIP 的单片机，装入 HEX 文件时将可装入 MiroChip IDE 开发环境生成 HEX 文件到全部缓冲区，保存时可保存所有缓冲到 HEX 文件，并完全兼容 MICROCHIP IDE 开发环境。

对于 Bin 文件的保存与打开，默认情况下是 CodeMemory 内容+ Data Memory 内容（如果存在）
对于 MICROCHIP 的单片机，Bin 同时存储了配置信息（按 Microchip 地址按排）

复制、粘贴、块另存为、块定义、块填充、清空当前缓存、清空所有缓存

这些菜单功能可使用快捷键或右键菜单调出。

详见文件编辑功能--->编辑缓冲区

4.1.2 芯片选择菜单



点击<查找选择芯片>



25 Flash 识别，是对 8 脚，16 脚的 25 系列 Nor Flash 进行自动识别，注意：因为很多芯片 ID 相同，所有结果有多个，你需要根据识别的结果，在其中选一个准确的型号。



首先选择芯片的引脚数量，点击<检测>开始识别

4.1.3 工程菜单

工程文件是为了更好的维护芯片的烧录而建立的概念。它是一个保存当前所有缓冲区数据及工作环境设置的文件，包括当前器件的信息，所有操作选项的设置，自动编号的有关设置。其方便日常量产使用。工程可设置密码保护，一定程度上可保护产品程序的安全。

工程菜单项目如下：



保存工程

在应用程序中设置好烧录芯片的所需参数后,选择【工程】--【保存工程】,即建立了一个工程。并在窗口的信区显示工程信息。在【保存工程】点击后，输入工程名称并点击确定后弹出下面工程对话框：



- 1、密码可为空，则当前工程不设置密码保护。
- 2、工程如果设置为保护模式，密码必须输入。设为工程在保护模式后，工程的有关数据将不可更改。并且编程器有部分功能被禁用。

打开工程

打开现存的工程,选择【工程】--【打开工程】,在打开窗口选择工程名称并点击确定。如果工程设置了密码,弹出以下对话框。(如果没有设置密码工程则直接打开)



注：对于 TL866 编程器的老版本工程文件，在此处也可以直接打开调入。

工程另存为：

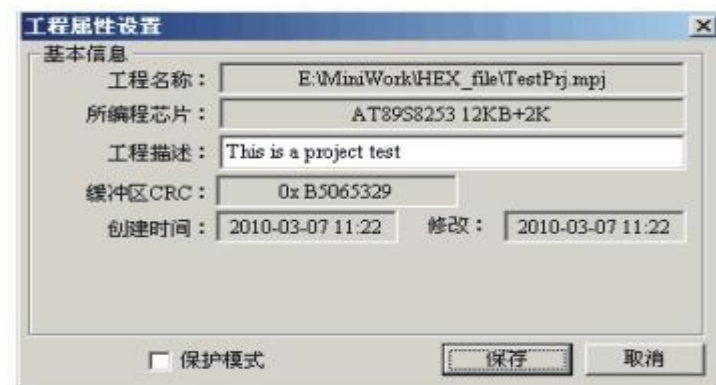
把当前工程，另存为另一个工程名。操作参考【保存工程】。

关闭当前工程

关闭打开的工程。切换到普通的编程模式，关闭工程并清空当前的缓冲区数据。

工程属性

选择【工程】--【工程属性】,弹出属性设置对话框。工程属性对话框可更改工程描述及工程保护模式。保存时需输入工程密码。



更改工程密码

重新设置工程密码



提示:

- 1、确切的工程描述可区分大量的工程。工程模式一般用与批量生产。
- 2、工程文件已经过一定的加密处理，保护好工程密码可防止工程文件复制。一定程度上可保护数据。
- 3、工程文件保存或打开时都含 32 位 CRC 校验信息，可防止存储文件损坏等其他不确定因素。

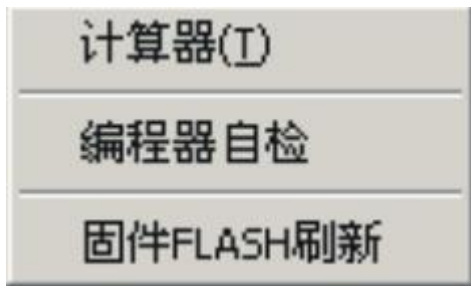
4.1.4 操作菜单

本菜单功能是对芯片的编程操作，具体内容请看 4.2 节 **编程器操作功能**
菜单项如下图：



4.1.5 系统工具菜单

工具菜单项目如下：



计算器： 这是 Windows 自带的一个计算器，切换到科学型模式时对各种格式的计算很方便。



编程器自检： 对本编程器硬件进行自检操作。

点击后弹出如下对话框：



注意: 编程器开始自检前, 请移除 40PIN 插座上的 IC 及 ICSP 连接线, 自检时编程器会向全部引脚施加 VPP、VCC、GND 进行测试, 并进行内部短路过电流测试 (同时在一个引脚上施加 VPP VCC 及 GND), 如果插入 IC 可能损坏插在上面的 IC。测试完成后, 在对话框内显示各个供电的测试情况。

固件 FLASH 刷新：

对编程器硬件固化程序进行刷新升级。

当升级新版本的应用程序后, 如需要对 FLASH 内程序进行升级刷新, 应用程序在进入时会提示进行固件刷

新，如果不需要刷新，则此刷新功能禁用。点击<固件 FLASH 刷新>弹出如下对话框：



点击<开始刷新>按钮进行刷新，刷新总时间大约 15 秒。点击<取消>可退出刷新。

刷新时注意：

刷新时不要使用外置 USB 集线器，直接接入电脑 USB 口，外置集线器有部分无法进行复位升级。关闭其他所有程序，并且保证计算机电源可靠！

4.1.6 编辑缓冲区

编程器软件可以的缓冲区的内容进行修改，常用的功能有：

- 缓冲区块定义
- 缓冲区块填充
- 复制、粘贴
- 块另存为
- 清空缓冲
- 查找、查找下一个
- 定位到地址

缓冲区块定义

块定义有两种方式：第一种方法是：在缓冲区内按鼠标左键，移动鼠标，选择一块数据。图如下：



第二种方法是：在【主菜单】或鼠标【右键菜单】选择【块定义】，或直接按快捷键“CTRL+B”弹出如下对话框，输入开始地址及结束地址，按确定键就完成块定义。

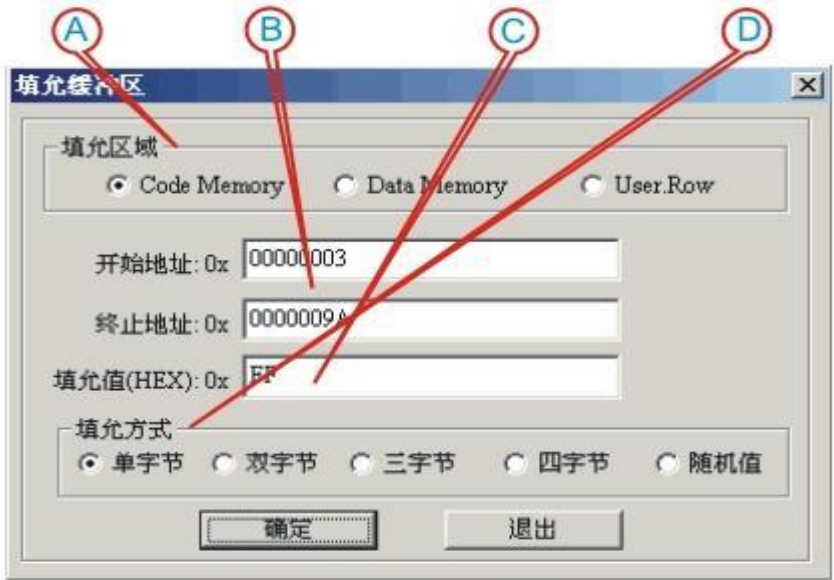


对定义的数据块你可进行后续的块操作：块复制、块填充、块另存为 TXT 文件

缓冲区块填充

填充缓冲区是指，将程序代码缓冲区中某段连续地址的数据全部替换为一固定的值。

弹出鼠标【右键菜单】选择【块填充】，弹出如下对话框



- A. 缓冲区选择栏，默认值为当前窗口显示的缓冲区。
- B. 要填充区块的开始地址和结束地址，默认值为：如定义了块，则为定义数据块的开始到结束。如果没有定义块，则为当前缓冲区的全区域大小。
- C. 填充值，最长允许 4 字节。如果填充方式为随机，则该值无关。

D. 填充方式，默认方式：如果芯片按 8 位方式编程，则为默认单字节方式，如果芯片是按 16 位方式编程的则为双字节方式，你可随意改变你想要的填充方式。

复制

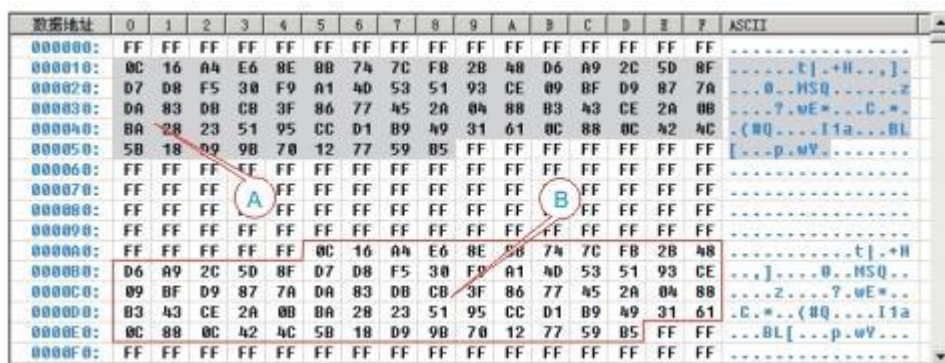
定义了数据块后，你可弹出鼠标【右键菜单】选择【复制】，这时块的内容已复制到 WINDOWS 的剪贴板内。也可直接按 CTRL+C 键，完成复制。

粘贴

功能：将复制到剪贴板的内容填充到缓冲区内。

操作：将光标移动到你要填充的缓冲区的开始地址后。按 CTRL+V 键。就完成粘贴。

将 A 区域的数据复制到 B 区域的效果



数据地址	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
000000:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
000010:	0C	16	A4	E6	8E	8B	74	7C	FB	28	48	06	A9	2C	5D	8F	...t .H...]
000020:	07	08	F5	30	F9	A1	4D	53	51	93	CE	09	8F	09	87	7A	...0..MSQ...z
000030:	0A	83	08	CB	3F	86	77	45	2A	04	88	83	43	CE	2A	0B	...?.wE*...C.*
000040:	8A	28	23	51	95	CC	01	B9	49	31	61	0C	88	0C	42	4C	... (BQ...I1a...BL
000050:	5B	18	09	9B	70	12	77	59	B5	FF	FF	FF	FF	FF	FF	FF	[...p.wV...
000060:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
000070:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
000080:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
000090:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0000A0:	FF	FF	FF	FF	FF	FF	0C	16	A4	E6	8E	8B	74	7C	FB	28	48
0000B0:	06	A9	2C	5D	8F	07	08	F5	30	F9	A1	4D	53	51	93	CE	...].0..MSQ..
0000C0:	09	8F	09	87	7A	0A	83	08	CB	3F	86	77	45	2A	04	88	...z...?.wE*..
0000D0:	03	43	CE	2A	0B	8A	28	23	51	95	CC	01	B9	49	31	61	...C.*..(BQ...I1a
0000E0:	0C	88	0C	42	4C	5B	18	09	9B	70	12	77	59	B5	FF	FF	...BL[...p.wV...
0000F0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	

注意：为了加快粘贴的速度，剪贴板的数据格式为本编程器专用的二进制格式，所以其他 WINDOW 应用程序不能直接使用本程序复制到剪贴板的数据，如果要将数据块复制给其他的应用程序使用，请使用【块另存为】TXT 文件功能。

块另存为

该功能把你定义的数据块输出为 TXT 文本文件，方便设计人员编程时应用，文本格式自动按 16 位或 8 位方式排列。操作方法：定义缓冲区块后，点击鼠标右键，弹出右键菜单，选择【块另存为】

清空缓冲

清空当前缓冲： 功能：清空应用程序当前窗口显示的缓冲区。

清空所有缓冲： 功能：把芯片的所有缓冲区以默认值清空。

查找、查找下一个

功能：查找当前缓冲区中的 HEX 或 ASCII 码字符串。 操作：按 CTRL+F 或点击右前键菜单内的【查找】项，弹出如下对话框：



在输入框中输入要查找的 HEX 值或 ASCII 码字符串，最后点击对话框中的"查找"按钮。当在程序代码缓冲区中有与此匹配的十六进制数据或字符串时，光标将定位至第一个与所查找内容匹配的内容的地址，按 F3 键继续查找下一个。直至缓冲区全部搜索完毕。

定位到地址

功能：光标移动到当前缓冲区指定的地址。

操作：按快捷键 CTRL+G，或按鼠标右键弹出菜单选择【地址定位】项后，弹出如下对话框：



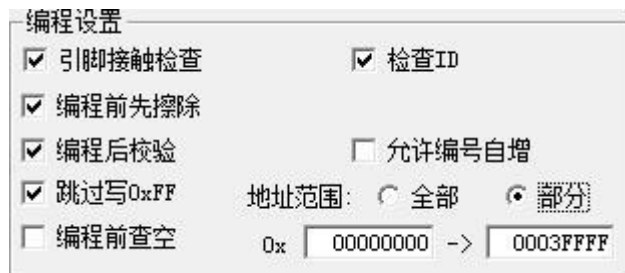
输入地址，按确定按钮，光标将移动到地址处

4.2 编程器操作功能

编程器操作功能详解	
选择芯片	按菜单列表选择被编程芯片
读取芯片	读入芯片的有关内容到 PC 缓存 （注： NAND 直接读入到文件）
编程芯片	把缓冲区（NAND 文件内容）的内容固化到芯片内
擦除芯片	完全擦除芯片的内容
查空芯片	检查芯片当前状态是否为未编程状态。
校验芯片	校检芯片与缓冲区的内容是否相同，可标记不同处校验。
设置芯片配置字	主要是设定保护位、熔丝位，加密位等等。
设置芯片编号	在芯片内增加序列号
编程器自检	检测编程器各个编程端口功能是否正常
ICSP 编程	ICSP 在线串行编程功能接口、及其使用方法
RAM 测试	测试常见的静态 RAM 芯片
数字逻辑电路测试	测试 54 74 系列、CMOS4000/MC14xxx 系列数字集成电路

4.2.1 编程设置

编程设置项目如下图，量产操作时的选项，一般按默认设置即可，无需进行更改。



4.2.1.1 引脚接触检查

在对芯片进行操作前，软件自动对芯片引脚是否接触良好，进行检查，该功能极大多数情况下能避免引脚接触不良引起的编程或读取错误。

引脚接触不良的极端情况：

当引脚氧化特别严重时，个别引脚可能不是完全断开，（例相当于一个 3-10K 的接触电阻），这种情况下无法判断是否接触良好，编程时可能出现错误，所以对旧芯片，应仔细处理引脚。

（在稀盐酸中浸泡 30 秒，取出后用水清洗后晾干，能达到最佳效果）

对极少部分芯片有可能出现个别引脚，引脚接触检查无法通过，这时在确保引脚接触良好的情况下临时取消这个选项。对芯片进行读取或写入操作前将不进行引脚检查。

4.2.1.2 编程前先擦除

在烧录编程操作前，首先进行芯片内容全片擦除。

有的芯片没有擦除功能，此选项会禁用，如：大部分 27Cxxx 芯片为一次性 OTP ROM 或为 UVRAM。是不可擦除或没有电擦除功能的。24 系列 25 系列 EEPROM 93 系列芯片,没有专门的擦除命令时，此功能也禁用，代表这些芯片不用擦除，可直接改写数据。（如果用户一定想擦空芯片内容，就是将 0xFF 写入芯片的全部存储空间）

4.2.1.3 编程后校验

正常情况此项必选中，否则编程完成后，数据可能是错误的。

4.2.1.4 跳过写 0xFF

对为 FF 的数据块将跳过写操作，能大大提高编程效率。此选项选中后，校验时也将跳过

4.2.1.5 检查 ID

很多芯片内部有一个芯片的识别标志（Identification），内容一般有 2 个或 2 个以上字节组成，第一个字节一般是制造商代码（Manufacture ID），后面紧跟的是 芯片类型或芯片容量。芯片不同 ID 会不同，这个选项的功能是防止不正确的芯片放入。该选项是在对芯片读取或写入前，对芯片 ID 进行检查，如果正确继续，如果 ID 错误中止，该功能是可选的。默认是使能选项

同类芯片，厂家不同可能编程方法是一样的，对没有在列表中支持的芯片，可以选择其他厂商同容量同类芯片进行烧录。因为 ID 会不同，烧录前，取消 ID 检查即可

4.2.1.6 地址范围

可以设置编程芯片的地址范围，量产时能提高效率。

4.2.1.7 编程前查空

除 27 系列 UVROM，此项一般没有必要选中，

4.2.1.8 允许编号自增

参见 4.2.9 节 设置芯片编号

4.2.2 选择芯片

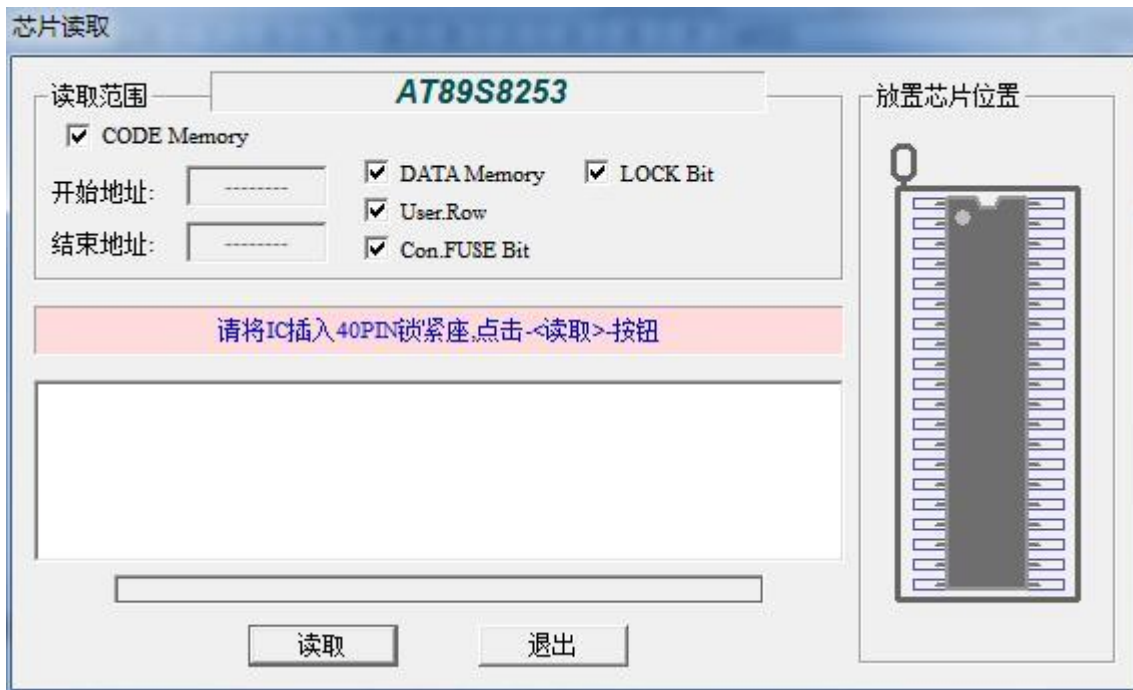
选择要被编程的芯片，在软件的左上角点击<选择 IC 按钮>



选择时注意芯片封装必须对上。部分芯片封装不同，引脚顺序可能不同。对同型号各种封装的 IC 如果没有在菜单中列出各种封装，对于非 DIP 封装的芯片则只要对照 IC 产品说明手册，直接连接对应引脚就可编程。

4.2.3 读取芯片

读取芯片是指将芯片中的数据读出至缓冲区中。不同芯片的读取窗口可能不同，下面以读取 AT89S8253 为例，说明如何读取芯片，点击工具栏中的【读】按钮，弹出下面对话框



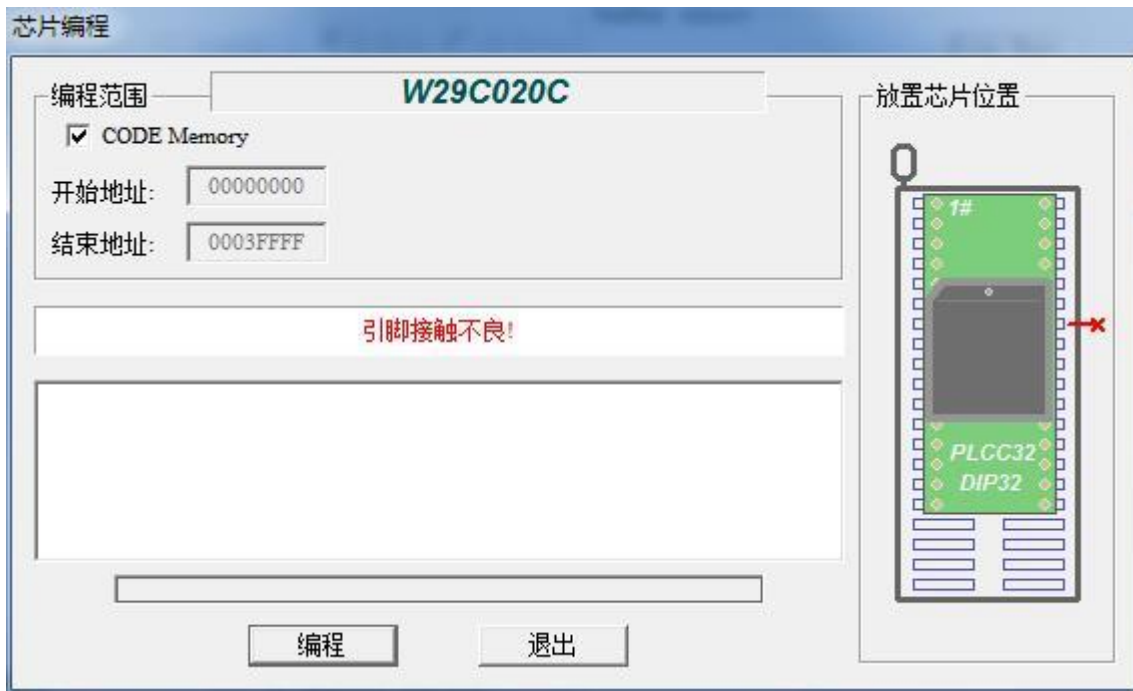
首先对【读取范围】进行设置,其中包括【CODE】【DATA】【FUSE】【LOCK】，开始地址及结束地址读取时无效。按【读取】按钮开始读芯片数据，注意：如果芯片被加密则读出值全为 0xFF。读取完成后，在信息栏内，显示各个缓冲的读取时间。

注意：读取操作一般不会报错,若想验证读取的数据是否正确,则使用校验功能验证,具体操作方法请查看校验说明，本编程器对芯片的操作前默认都会首先检查芯片的 ID 值。

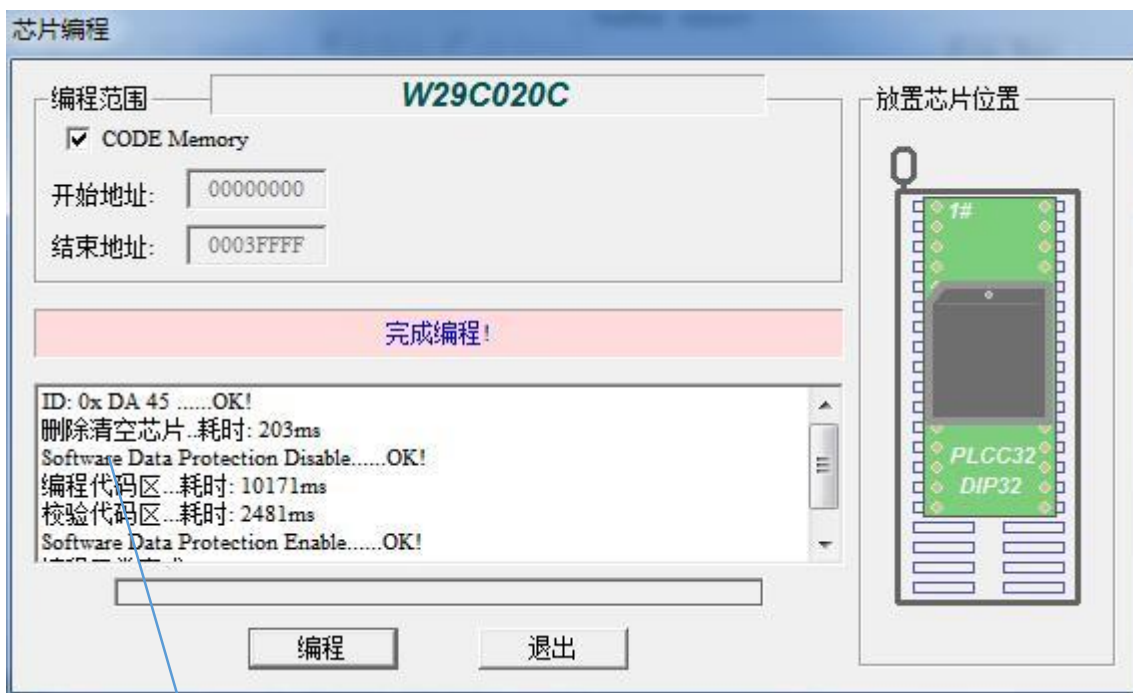
4.2.4 编程芯片

点击工具栏中的<写>按钮，编程对话框如下

以 W29C020 为例，如果引脚有接触不良，如下显示不良的引脚状态



正常编程如下图所示

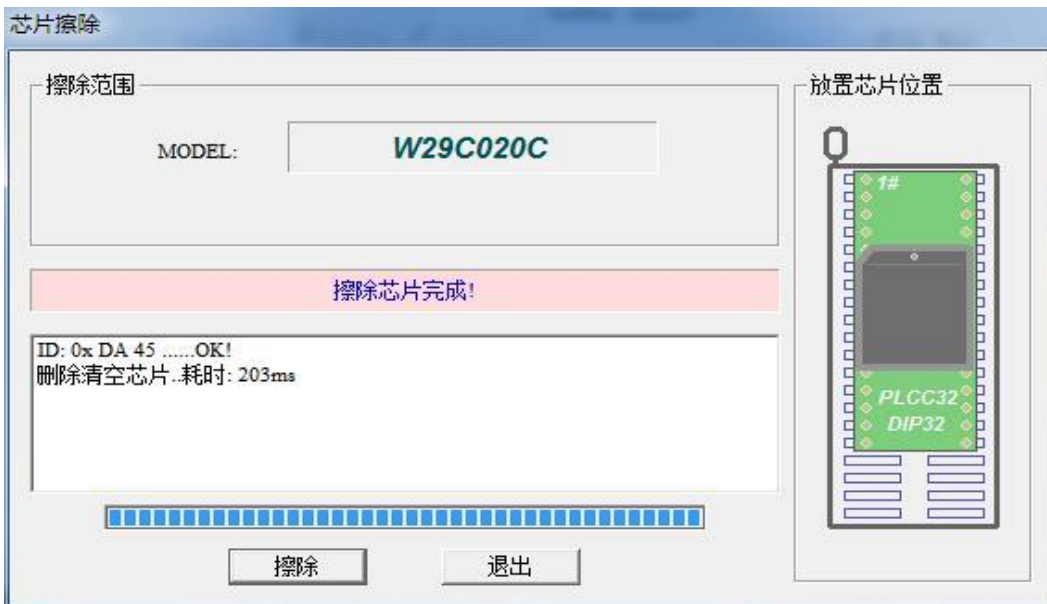


信息框内显示所有操作的过程及
所消耗的时间

大部分芯片，本编程器在编程过程中进行自动校验，如果编程过程中发现错误会立刻终止编程。如果用户选择了"芯片编号自增"选项，则在芯片编程完后，缓冲区中的芯片编号将按照设定自动增加。为了保证编程数据的正确，对于编程后校验，建议用户总是执行。

4.2.5 擦除芯片

对可电擦除的芯片点击工具栏中的【擦除】按钮，弹出下面对话框：



默认情况擦除前检查芯片 ID，对很少部分单片机，有的芯片加密后不可读取芯片 ID，这时应在主界面中取消 ID 检查，否则 ID 检查错误。点击对话框中的【擦除】按钮，擦除完成后，显示擦除的时间。

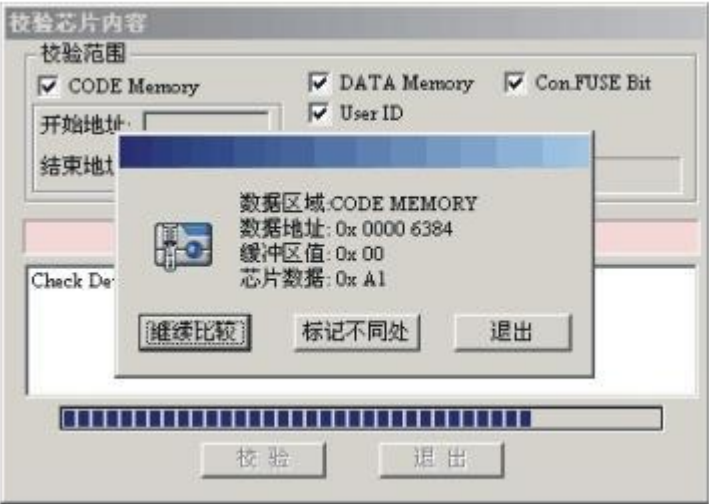
4.2.6 查空芯片

默认情况擦除前检查芯片 ID，查空前可先选择芯片要查空的区域，默认为全部。点击对话框中的【查空】按钮，开始查空操作后，完成显示查空各个区域所化的时间。如果查空其间，发现芯片非空，则终止查空，并显示非空区域的地址及存储值。

4.2.7 校验芯片

校验前可先选择芯片要校验的区域，默认为全部。点击对话框中的【校验】按钮，开始校验操作后，如全部数据相同，完成后显示校验各个区域所消耗的时间。

如果校验期间，发现芯片数据不同，则弹出如下对话框：



点击【继续比较】： 将继续比较如又有不同则会弹出同样的对话框提示。

点击【 标记不同处】： 将比较完芯片的全部内容，在比较过程中如发现不同也不会弹出提示，直到完成后，提示有多少处不同，并在缓冲区内以红色字体显示不同的单元（NAND 除外），如下图所示。

点击【退出】： 将直接退出，不进行后面的比较。



4.2.8 设置芯片配置字

点击芯片缓冲区选择栏中的【配置】按钮，弹出下面界面(芯片型号 PIC18F4550):

选择编程接口

☒ 40PIN锁紧座
☐ ICSP串行接口
☐ ICSP_VCC Enable

buffer select

☒ 8 Bits
☐ 16 Bits

Code MemoData Memo配置

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0X300000:	-	-	☒ USBDIV	☒ CPUDIV1	☒ CPUDIV0	☒ PLLDIV2	☒ PLLDIV1	☒ PLLDIV0
0X300001:	☒ IESO	☒ FCMEN	-	-	☒ FOSC3	☐ FOSC2	☒ FOSC1	☐ FOSC0
0X300002:	-	-	☒ VREGEN	☐ BORV1	☐ BORV0	☐ BOREN1	☐ BOREN0	☐ /PWRTEN
0X300003:	-	-	-	☐ WDTPS3	☐ WDTPS2	☐ WDTPS1	☐ WDTPS0	☐ WDTEN
0X300004:	-	-	-	-	-	-	-	-
0X300005:	☐ MCLRE	-	-	-	-	☒ LPT1OSC	☐ PBADEN	☐ CCP2MX
0X300006:	☐ /DEBUG	☒ XINST	☒ ICPRT	-	-	☐ LVP	-	☐ STVERN
0X300007:	-	-	-	-	-	-	-	-
0X300008:	-	-	-	-	☐ CP3	☐ CP2	☐ CP1	☐ CP0
0X300009:	☐ CPD	☐ CPB	-	-	-	-	-	-
0X30000A:	-	-	-	-	☐ WRT3	☐ WRT2	☐ WRT1	☐ WRT0
0X30000B:	☐ WRID	☐ WRIB	☐ WRTC	-	-	-	-	-
0X30000C:	-	-	-	-	☐ EBTR3	☐ EBTR2	☐ EBTR1	☐ EBTR0
0X30000D:	-	☐ EBTRB	-	-	-	-	-	-

编程设置

☒ 引脚接触检查
☒ 检查ID

☒ 编程前先擦除
☐ 允许编号自增

☒ 编程后校验

☒ 跳过0xFF

☐ 编程前查空

地址范围:
☒ 全部
☐ 部分

0x
->

芯片配置信息

CONFIG: (0x300000-0x30000D)

00 05 1F 1F 00 83 85
00 0F C0 0F E0 0F 40
Note:Checked = 0

USERID: (0x200000-0x200007)

FF FF FF FF FF FF FF FF

USERID:

下侧红圈内配置信息的内容的值与配置界面对应：点击配置界面，配置信息内的值自动改变。

不同芯片的配置信息不同，所以用户首先查看芯片的数据手册,并按照具体的需要来正确的配置芯片。

芯片配置字，需理解配置字中各个位的作用

4.2.9 设置芯片编号

芯片编号设定功能是，在编程时给芯片自动加入一个编号到芯片某个存储区域内，可方便用于芯片出厂时的设置编程。加入时间或唯一序列号等等，本编程器有多种编号算法可供使用，能附合各种用户的要求。点击<主菜单>---<操作>---<自动编号设定>可弹出如下对话框：

芯片序列号设置

自增算法选择

☒ 默认增量
☐ 时间算法
☐ 随机序列号
☐ 用户自定义

编号存放区块

☒ CODE Memory
☐ DATA Memory
☐ USER_ROW
☐ USERID

算法设置

编号始地址:0x

编号长度:

编号初始值:

增量步长(10):

编码模式

☒ HEX
☐ ASCII

增量模式

☒ 16进制
☐ 10进制

☐ 从高位地址开始递增

记录文件:

注 编号记录到文件,空为不记录

☒ 允许自动编号

测试

保存设置

取消

首先选择自动编号算法，在对话框内有四种算法可供选择，下面对各种编号算法的作分别说明。

海口鑫工电子有限公司
<http://www.xgecu.com>

25

4.2.9.1 默认增量算法

该算法的设置界面就是上图所示，该算法功能是在设定的存储区域开始地址的地方，放入一个设定的长度的编号，编号每次按步长增加。设定好参数后，你可按<测试>按钮进行测试，查看各种参数设定下的变化规律。

编码的初值及步长等可在对话框内设定。

设定好参数并测试正确无误后，设定记录文件，如果为空则记录文件无效，编程时不会把数据记录到文件中，如果设定了记录文件，则每次编程后，都把你设置的编号添加到记录文件的最后。

最后，选中<允许自动编号>按钮，并<保存设置>。

4.2.9.2 时间算法

时间算法对话框界面如下图：在 0x2000 处，放入了一个时间数据



时间编码格式有四种，分别是：

- (1) 日期+时间 编号总长：16 字节
编程时在指定的位置添加日期及时间，总长是 16bytes.
- (2) 日期+5 个字节的增量 编号总长：16 字节
其中增量部分内容可设定初值及步长。
- (3) 日期+时间+8 个字节的增量 编号总长：24 字节
其中增量部分内容可设定初值及步长。
- (4) 日期+时间+16 字节随机编号 编号总长：32 字节

其中 16 字节是随机产生的字节，可生成产品的唯一序列号。这种方式下，记录文件必须设置，让产生的编码记录在文件中。否则你自己也不知道产品的编号是什么。

后面的操作步骤同上《默认增量算法》

4.2.9.3 随机序列号算法

该算法产生一个指定长度的随机号，如下图：在地址 0x2000 开始处，放入了 16 个字节的随机数据



4.2.9.4 用户自定义算法

用户自定义算法有很强的灵活性，你可以设定任意的编号算法，详细的使用方法请看目录..\Serialnumber\source_dll 的中 VC++源程序中的注释说明。界面如下：



AVR 单片机的 RC 校正字节加密方法，完全利用此地调用 DLL 动态库实现，
..\TAMEGA_LED 目录内含所有源程序及测试方法。

制作好 DLL 文件后，你只要把它复制到应用程序目录下的..\Serialnumber 的子目录就可。设置时就在用户算法文件列表中，选择你制作的算法文件。后面的操作步骤与其他算法相同。

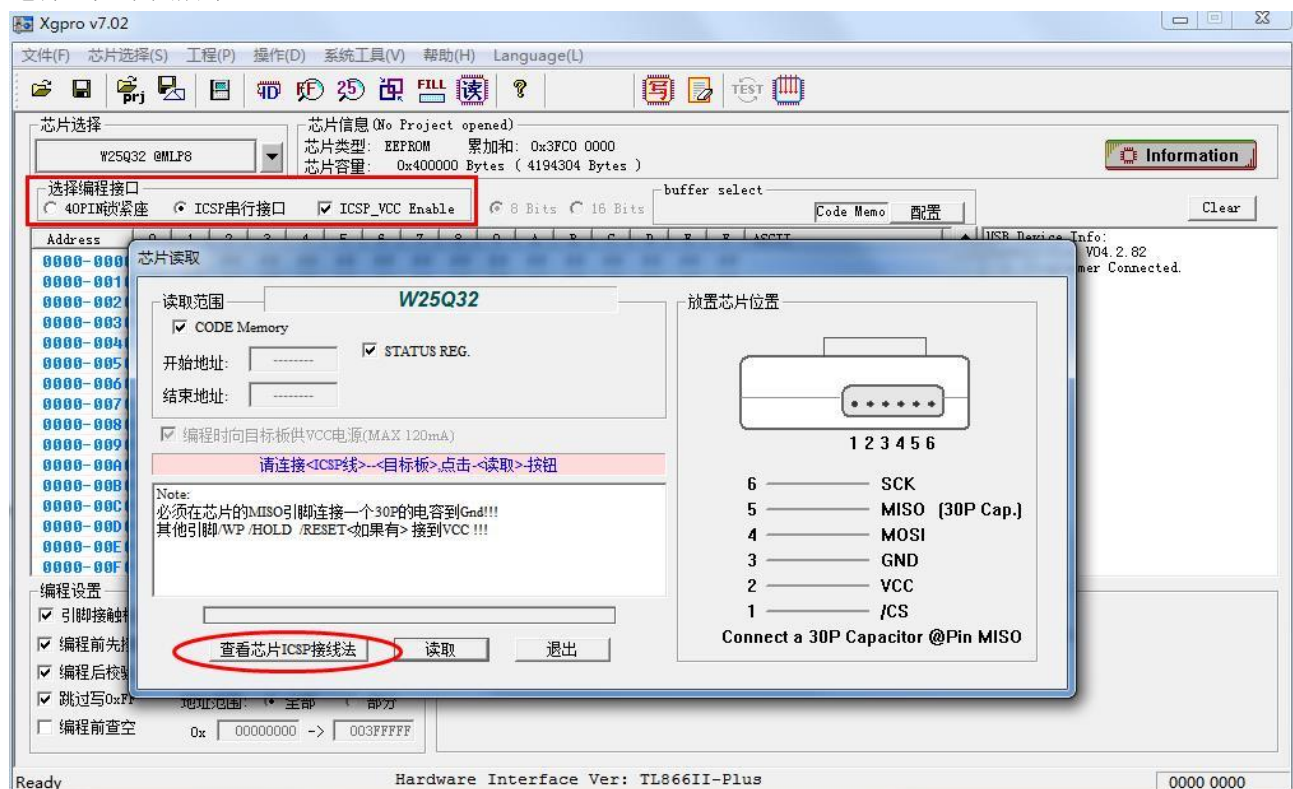
重要提示:

- 1、 设置好以上内容后，编程器时在编程操作时，并不会自动添加你设定的编号到芯片中，要自动添加编号你还必须在主界面中选中<芯片编号自增>。
- 2、 主界面中能选中<芯片编号自增>的条件是，必须首选设置好编号算法，并在算法内设定<允许自编号>。
- 3、 在主界面中选中<芯片编号自增>后，如果你又重新设定编号算法，则主界面中<芯片编号自增>按钮必须重新选中。
- 4、 编程芯片型号变换后，必须重新设定编号设定算法。
- 5、 在工程模式下，该算法保存在工程文件中，可直接打开工程文件，并自动调入自动编号算法。所以批量生产时一般在工程模式下使用，简单方便。

4.2.10 ICSP 编程

对可串行在线编程的芯片，本编程器可通过独特的 ICSP 接口对目标板上的芯片进行编程，编程前先在主界面的<编程接口选择栏>，选择 ICSP 口，此时【ICSP_VCC_ENABLE】变为允许状态，默认【ICSP_VCC_ENABLE】选中，表示编程向目标板提供 VCC 电源。注意，编程器 VCC 电源最大能提供的电流为 120ma,如果目标板需要更大的电源，请用目标板自供电源。

如果不需要编程器提供电源，可不选中【ICSP_VCC_ENABLE】按钮，这样编程时的 VCC 引脚上将没有 VCC 电源。如下图所示:



这样下一步操作，读取、擦除、校验、编程等操作时，将从 ICSP 口进行操作，操作时 40PIN 万能座上不可同时插入芯片，点击工具栏上的【写】按钮，弹出如下对话框：（注意读取、擦除、校验等操作的对话框和这个写操作一样）。右侧显示 ICSP 口接线示意图，更详细的接线图可点击【查看芯片 ICSP 接线法】

ICSP 支持的芯片有:

- 1、 24 系列 25 系列 93 系列，
- 2、 ATME89S51、52, AVR ATMEGA 全系列, 注：并行时为高压编程，ISP 时为低压编程
- 3、 MICROCHIP PIC10Fxxx 12Fxxx 16Fxxx 18Fxxx 全系列，

4、新茂 SYNCMOS SM59Dxx SM59Rxx 全系列芯片

4.2.11 多机编程

软件支持最多 4 台编程器同时编程，（NAND FLASH 暂不到持多机编程）。

USB 扩展请选择使用带电源的优质 USB 集线器，

进行多机编程界面如下：多机编程时**自动编号**功能可正常使用。



多机编程时可使用快捷键 **【Z】【X】【C】【V】**，启动相应的编程器。

4.2.12 RAM 测试

选择对应型号的 RAM 芯片后，点击工具栏中的**【测试】**按钮，弹出下面对话框：



RAM 测试有 4 种方法，通过这四种方法的测试，基本上可测试 RAM 的单元功能是否正常。一般编程器只对 RAM 单元读写是不可测试出 RAM 的好坏的。例如：如果两条相邻的地址线短路，你只是向 RAM 单元简单地读写数据，是不可能检查出问题的。所以只是有一种方法测试是不完整的测试。

插入芯片，点击【测试】按钮，可进行 RAM 的测试。

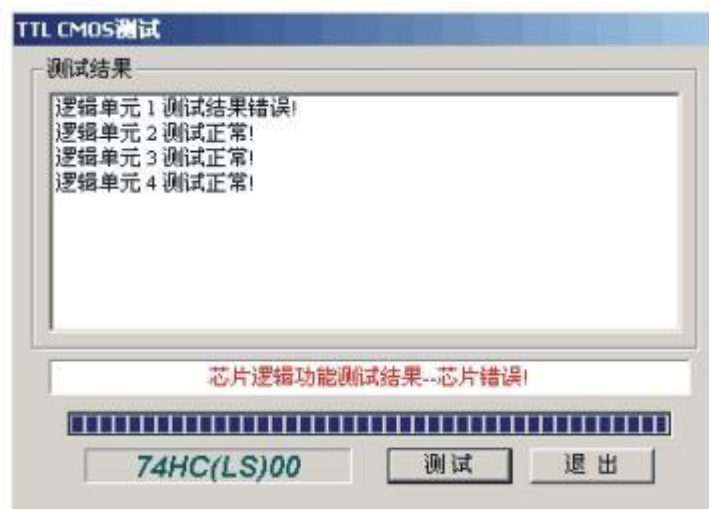
4.2.13 数字逻辑电路测试

本编程器可能是真真可以测试常用 74 系列 CMOS4000 系列集成电路的第一台开发型综合功能的编程器，测试时可以定位到某个门电路的故障。

选择对应型号的数字逻辑集成芯片后，点击工具栏中的【测试】按钮，弹出下面对话框：



以上是对 74HC00 的测试结果，正常情况下显示各个逻辑单元测试是否正常，信息框内显示经多种输入组合测试。如果集成电路某个门电路错误，例把 74HC00 的 PIN1 弯曲断开，测试结果显示如下：提示芯片错误。



根据芯片不同，逻辑单元个数不同。逻辑单元编号同集成电路手册。

5. NAND Flash 烧录说明

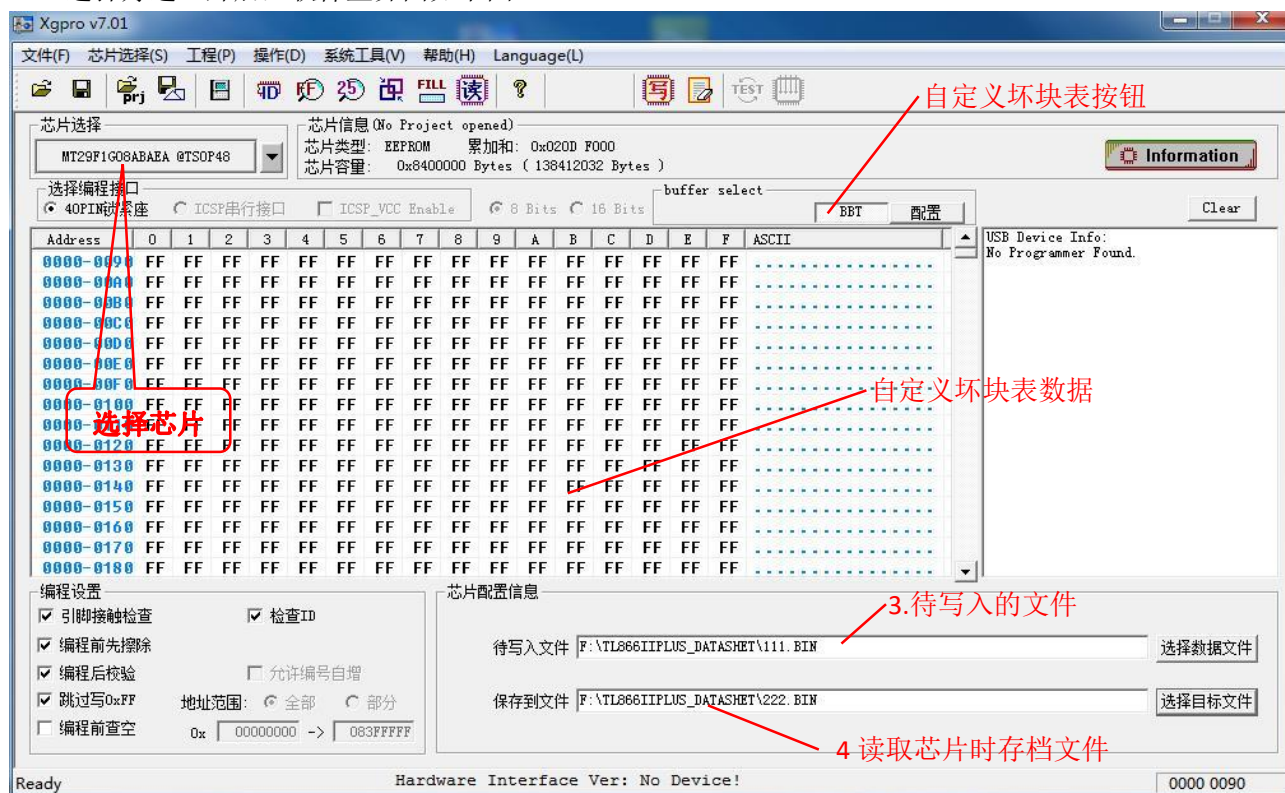
5.1 NAND FLASH 的特点:

- A. NOR Flash 全随机访问内存映射和专用接口(如 EPROM)地址和数据行。而 NAND 闪存没有地址专线。它是由通过 8/16 位宽接口总线发送命令,地址和数据到内部寄存器,这样就为许多主控提供了更灵活的配置方式, NAND flash 结构, 强调降低每比特的成本, 更高的性能, 并且象磁盘一样可以通过接口轻松升级, NAND 成本更低, 容量更大。
- B. Nand Flash 允许有坏块存在。由于 NAND 生产工艺的原因, 出厂芯片中会随机出现坏块。坏块在出厂时已经被初始化, 并在特殊区域中标记为不可用, 在使用过程中如果出现坏块, 也需要进行标记。
- C. 易出现位反转。NAND FLASH 更易出现位反转的现象, 如果位反转出现在关键文件上, 会导致系统挂机。所以在使用 NAND FLASH 的同时, 需使用 ECC/EDC 等算法确保可靠性。
- D. 存在 Spare 区。正因为 NAND FLASH 有着上面的两项特殊的地方, Spare 区就扮演作存放坏块标志, ECC 值以及芯片信息和文件信息的作用。
- E. NAND FLASH 必须进行坏块处理

NAND FLASH 的坏块处理方式有很多, 不同的公司或者系统提供商都会选择不同的坏块处理方法, 来满足产品开发的需要。TL866 编程器定义了常用的三种方法可供用户选择使用。并且接合用户自定义 ECC 算法实现灵活应用。

5.2 烧录主界面：

选择好芯片后，软件主界面如下图 1-1：



1、 点击此按钮<选择芯片>

2、 自定义<BBT>坏块表（Bad Block Table）按钮

TL866II 编程器，设定了一个可选的 BLOCK 大小的自定义表，这个表一般可作为专用的坏块表 BBT 使用，在烧写完文件内容的最后，紧接着可以将此表内容写入到用户在配置界面设定的芯片任意位置中。但 BBT 写入的块索引号参数设定需保证地址块不重叠。

- (1) 表中的数据可以在 DLL 自定义函数中，跟据芯片的实际烧录结果改变数据。
- (2) 表中可以装入初始值：可在文件菜单中打开文件，数据预先装入 BBT 表。
- (3) 如果在烧写配置中没有选定 Enable BBT Programming to Block 相关选项,则此表被忽略。

3、 读取芯片前：需要首先设定好存档的文件名

编程芯片时：要设定要写入到芯片的源数据文件名。

注意：NAND 编程时，数据直接从设定的<待写入文件>写入到芯片中。读取时是将芯片的内容直接保存到文件中。

这操作与其他芯片不同，其他非 NAND 芯片操作：编程时，是将缓冲区数据（内存中）写入到芯片。读取时是将芯片数据读入到电脑缓冲区。保存文件时是将缓存中的数据保存到磁盘文件中。打开文件是指将磁盘文件数据调入到电脑内存中。

在 NAND 中，<保存文件>与<打开文件>是对 BBT 表的内存缓冲区的操作。

本文档以 MT29F1G08ABAEA 为例，阐述了 TL866II 编程器对 NAND FLASH 的烧录方法

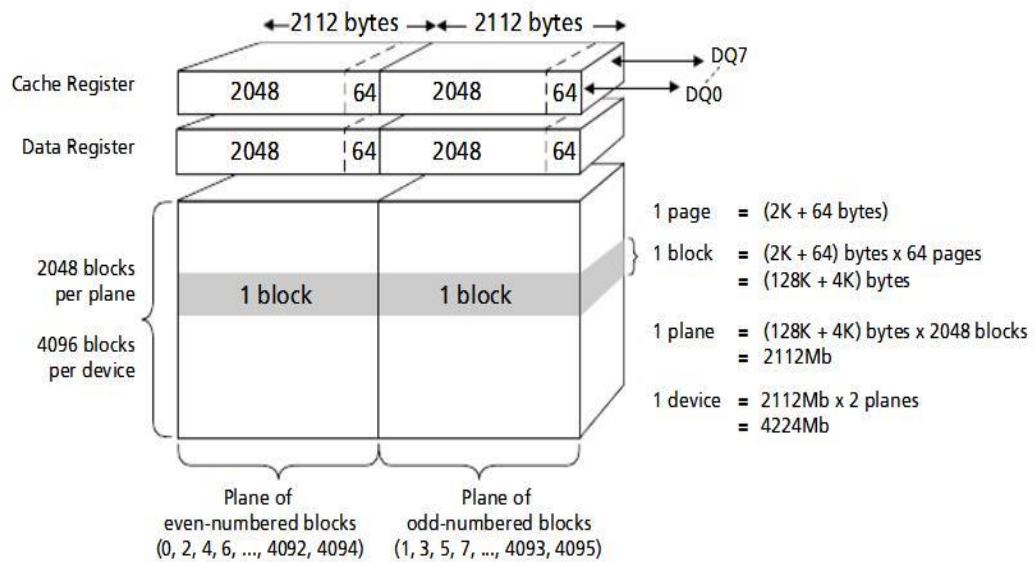
MT29F1G08 有 1024 个块(Block)组成, TL866II 编程器是以块为单位进行操作的, 索引号 0-1023 块。

每个块有 64 个页 (page), 对每个块的处理过程中是分别按页顺序写入芯片中的。

每个页大小: 2048 bytes (Page size) + 64 bytes (Spare size) = 2112 bytes

芯片的总容量为: $2112 \times 64 \text{ (页)} \times 1024 \text{ (块 Block)} = 138,412,032 \text{ bytes}$.

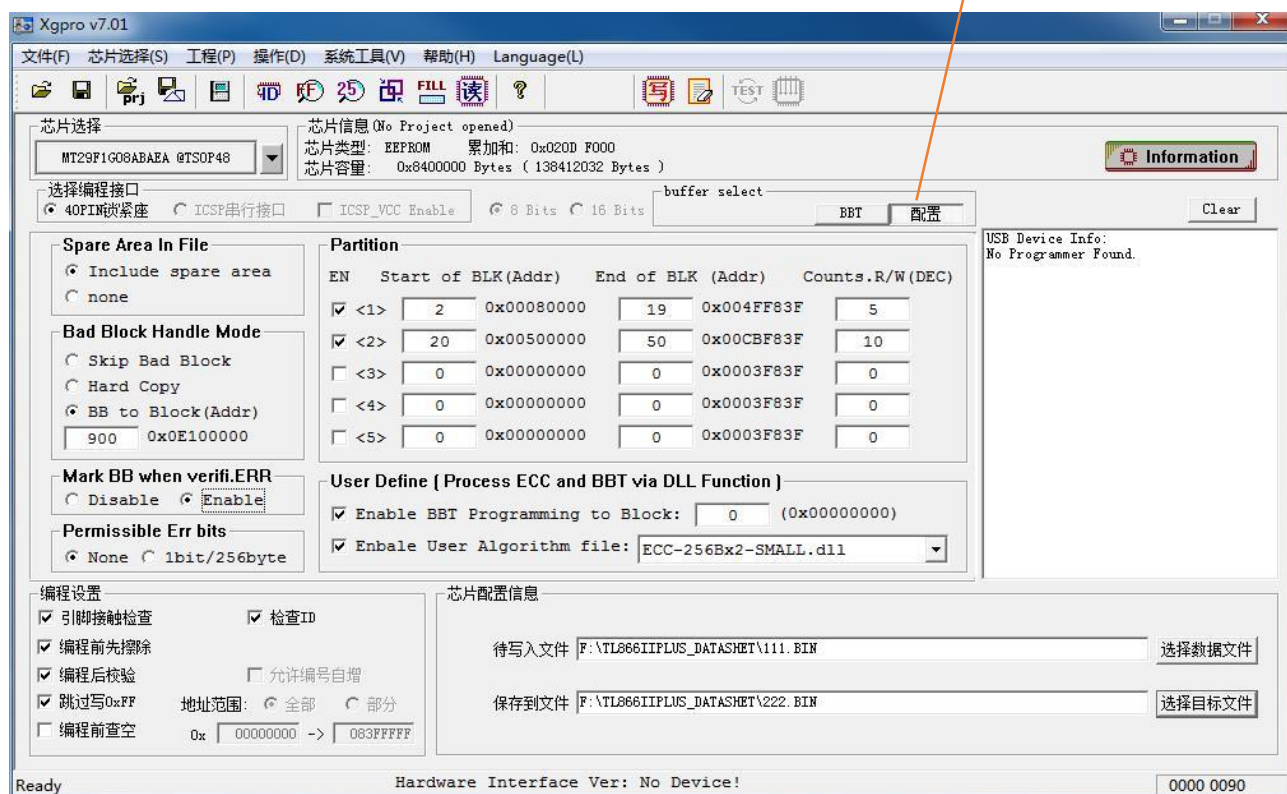
Figure 9: Array Organization - MT29F4G08 (x8)



5.3 配置界面

如下图，点击 < 配置 > 按钮进入烧录配置界面。
准确配置烧写参数是芯片能成功写入的关键。

图 1-2:



5.4 配置说明

Spare Area In file

(文件中数据包含冗余区数据)

由于 NAND FLASH 的特殊性质，用户文件中的数据跟据实际情况是否含有 Spare Area（冗余区）的数据。如果不包含：

在读芯片内容时，保存的文件中将不包刮 Spare Area（冗余区 64 字节）的数据。

在编程（写入）芯片时，编程器只从文件中读取 PageSize(1024)字节的数据，后面的（SpareSize）64 字节数据将默认为空数据(0xFF)。这种情况，一般需要使用用户自定义算法文件，将冗余区的数据按用户要求正确填入。

None 文件中数据不包含冗余区数据

Bad Block Handle Mode (坏块处理模式)

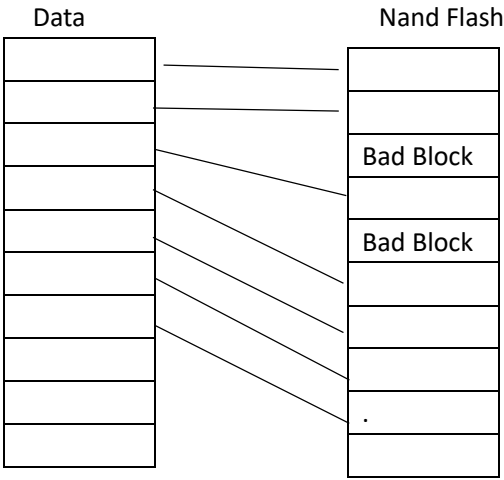
坏块处理方法有三种：

Skip Bad Block （跳过坏块）：

在读写芯片过程中如果发现坏块，TL866II 编程器将直接跳过坏块（Block）.直接对下一块数据进行读或

写。写入方式如下图

表 1-1



Hard Copy（硬拷贝）：不管块是否有坏块标记，强制进行数据读取或写入，如果在写入过程中校验出错，将停止编程。这种方法一般为技术性维修时，更换存储芯片时的使用方法。

BB to Block：发现坏块写入到特定的块开始处，这样的好处是让系统在逻辑上感觉存储器依然是连续的区域，一般与分区（Partition）配合使用。在后面实例配置中详细说明。编辑框中输入数字（例:900）是发现坏块时，实际写入到 FLASH 中的替换块索引（地址）

Mark BB When Verify Error:

当编程数据后，发生校验错误时，自动将这块（BLOCK）标记为坏块，并将数据写入到下一个块（**Skip Bad Block**），或将数据写入到特定的坏块数据存放处（**BB to Block** 替代地址）

如果不使能（disable），校验错误时将终止编程。

Permissible ERR bits (允错位)

None 不允许有错误发生

1bits/256bytes：每 256 个字节允许 1 个位错误。在读取 NAND FLASH 芯片内容时，有可能发生位翻转，当校验时如果检测到一个位错误（连续的 256 个字节中），将不视为错误。但 spare area 中的数据中仍然是不允许出现任何位错误。

Partition (分区)

将文件内容分区块要求写入到 NAND FLASH。或按分区要求读入到文件。

本软件允许将文件内容最多设置为 5 个分区，分别写入。

EN 选项：选择是否允许该分区。

Start of BLK：分区的开始块索引号

End of BLK：分区的结束块索引号

Counts of R/W：编程软件对该分区实际的读或写的块数

分区参数设置要求：

- A. 所有的块索引号，不可大于 NAND FLASH 实际含有的块数。
 - B. $\text{Start of BLK} \leq \text{End of Block}$, $\text{Counts of R/W} \leq (\text{End of Block s} - \text{Start of BLK} + 1)$ 分区时实际写入的块数量（Counts of R/W），最好小于分区大小几个块。使其有一定的余量
 - C. 下一个分区的 **Start of BLK** 应大于上一个分区的 **End of Block**
- 如果使用 **BB To Block** 选项，则 **Bad Block** 替换地址不应该包含在任何一个分区中，如果使用 **BBT** 表，则 **BBT** 表的写入位置也应不包含在任何一个分区中。

Enable BBT Programming to Block xx

BBT 表的内数据编程到 xx BLOCK

使能本操作选项，在全部分区（或全部文件数据）写入到 FLASH 中后，编程器最后将 BBT 表内容，写入到指定的 xx 块中。

Enable User Algorithm file

使用用户自定义算法文件

在自定义算法文件中，用户可以对块中的任何数据进行编程前的数据处理或自动序列号计算，及进行 ECC 算法等等。也能跟据编程过程中坏块的跳过情况进行 BBT 表的数据自动填入，供用户能证应用程序正确读入内容。

具体操作方法，请参考 DLL 动态函数的原代码中的注释。

5.5 配置实例 1(MT29F1G08)

- 1) 使用（Partition）分区表对 NAND FLASH 进行文件写入操作
- 2) 坏块处理模式：Skip bad block
- 3) 文件数据：include spare area （一般原数据文件需要处理好 Spare area 的内容）
- 4) 使用 2 个分区：
第 1 个分区：从 0#-199# 总共 200 个 BLOCK，实际写入数据长度为 10 个 BLOCK
第 2 个分区：从 1000#--1023# 总共 24 个 BLOCK，实际写入数据长度为 5 个 BLOCK
写入的数据文件为：Test.BIN (注意：NAND 编程时，只能使用二进制格式的文件)

通过以上设置，就可进行编程操作，操作的结果示意见表 1-2

如果文件长度不足，则结束编程。

如果文件长度超过需要写入的 BLOCK（15 blocks）数据长度，则写完设定的 BLOCK 数量后结束编程。

配置界面如下图所示：图 1-3：

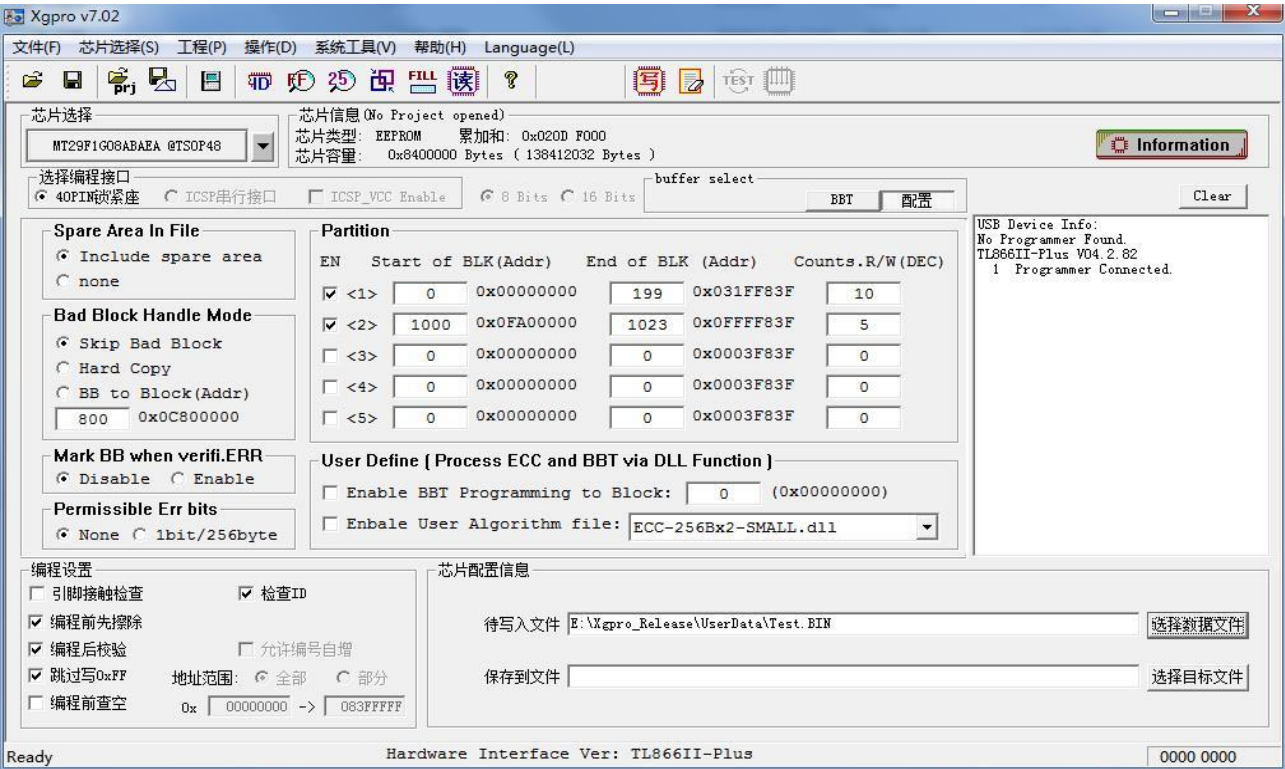
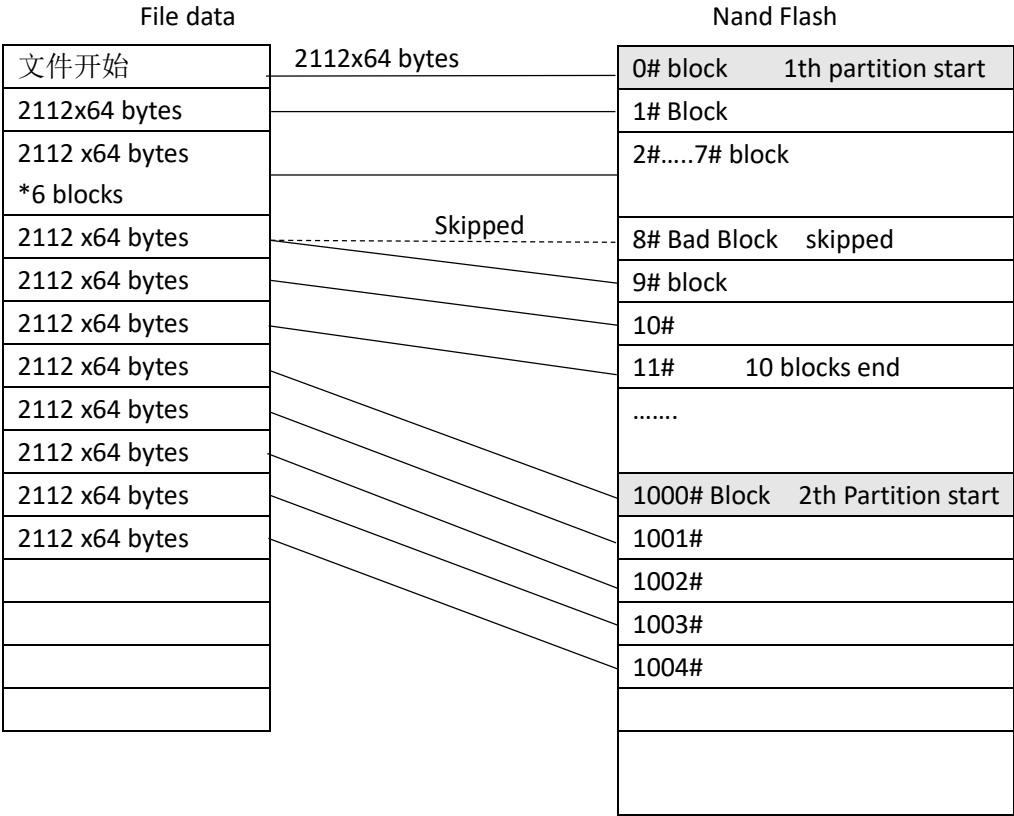


表 1-2：烧录过程示意



注意：
在这配置实例下，读取或校验芯片时，也是按分区进行的

- 1、读取分区 1 的 10 个 BLOCK 保存到文件
- 2、读取分区 2 的 5 个 BLOCK 保存到文件
- 3、读取完成。

通过灵活地设置分区，你可以单独读取或写入几个或一个 BLOCK 的数据，

5.6 配置实例 2(K9F1208UOC)

以 K9F1208UOC 为例 芯片容量：(512+16)*32*4096 块
K9F1208UOC:
Page Size: 512 bytes
Spare size: 16 bytes
Pages Per Block : 32 pages
Block Size : 4096

- 1) 使用 (Partition) 分区表对 NAND FLASH 进入文件写入操作
- 2) 坏块处理模式: BB to Block 4000, 发现坏块, 写到块 4000#开始位置
- 3) 文件数据: not include spare area (在 DLL 中处理 Spare area 的内容)
- 4) 使用 2 个分区,
第 1 个分区: 从 0#-1000# 总共 1001 个 BLOCK, 实际写入数据长度为 100 个 BLOCK
第 2 个分区: 从 2000#--3999# 总共 1000 个 BLOCK, 实际写入数据长度为 3 个 BLOCK
- 5) 写入的数据文件为: Test.BIN (注意: NAND 编程时, 只能使用二进制格式的文件)
- 6) 使能 BBT Programming to Block 0. BBT 表的内容定入到 0# Block.
- 7) 使能 User Algorithm file ,使能算法文件, ECC_256Bx2-SMALL.dll,(该 DLL 实例中只是计算 ECC 值, 放入 spare area, 用户可以跟据实际需要重新修改 DLL 函数, 修改 BBT 表内容)
- 8) 烧录过程示意如表 1-3 所示。

图 1-4:

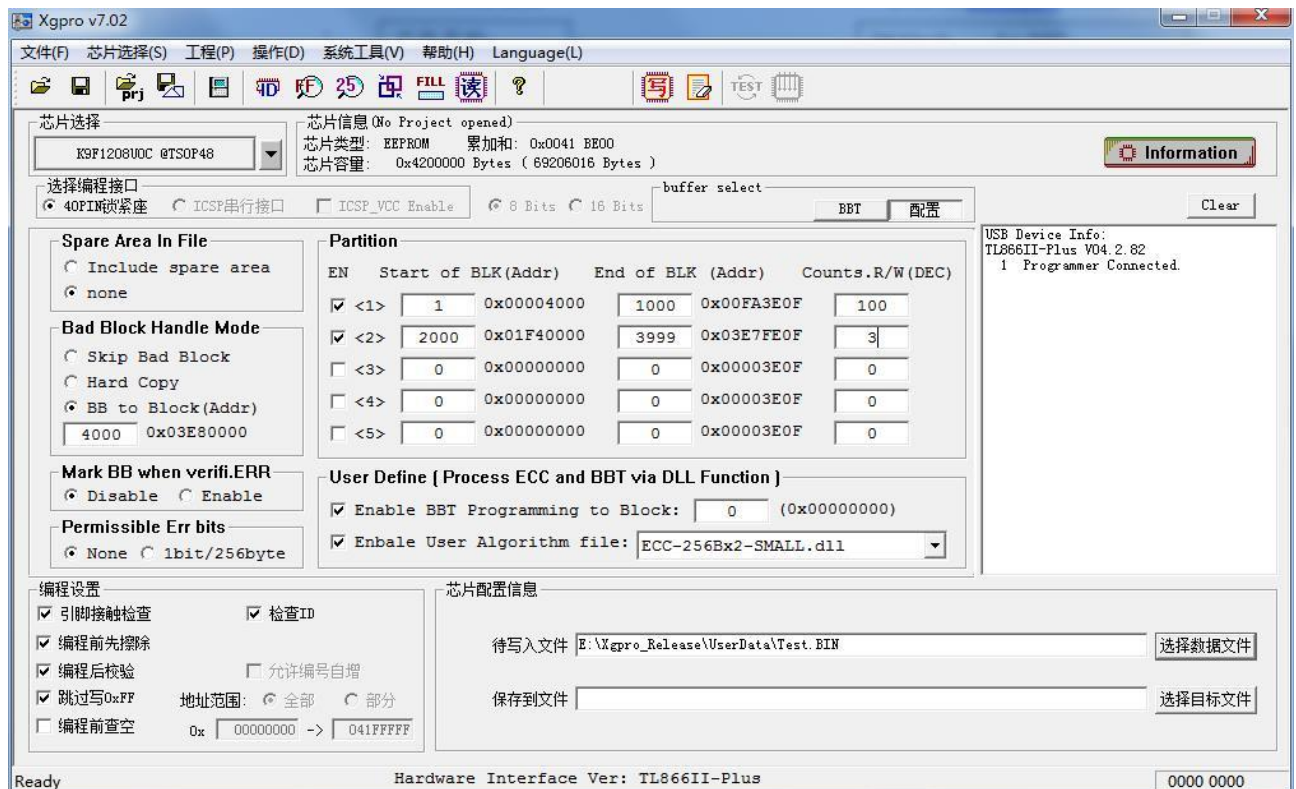
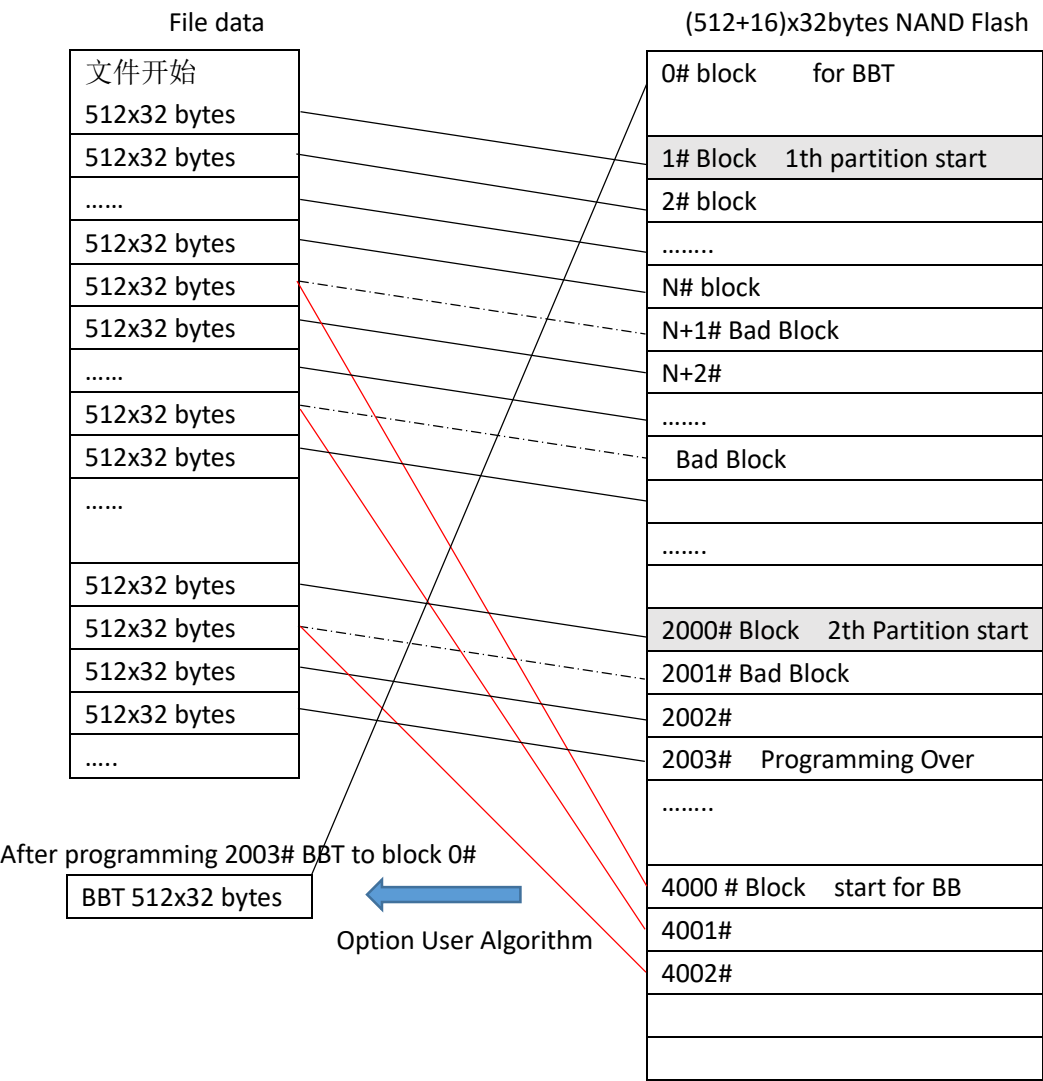


表 1-3：烧录过程示意



注：16 bytes Spare Area Data get from DLL function or all are 0xFF

K9F1208U0C:

Page Size: 512 bytes

Spare size: 16 bytes

Pages Per Block : 32 pages

Block Size : 4096

5.7 保存工程文件

所有的配置参数，及 BBT 表内容在设置完成后，可以“保存工程”，当再次使用时，只需要打开该工程文件后可直接烧写，适用于批量生产。

6. NAND Flash 芯片复制方法

6.1 注意事项

在维修设备的过程中，我们经常会需要从一个母片中读出数据后，复制到的一个新的芯片中，然后焊到电路板上运行，为保证芯片能正常运行，用户需要注意以下几点：

- 1) 复制芯片的配置：用 TL866II 编程器软件的默认设置。

如下图选项，必须使用 **Hard Copy** 文件中数据必须包含：**include Spare area**，其他选项全部默认不选用。按这模式配置读取时，编程器软件能读取芯片的全部内容（包刮坏块内容）到文件。

- 2) 有部分 NAND FLASH 芯片有唯一序列号 unique ID，嵌入式系统应用时有可能会读取芯片的 unique ID，并在程序中进行加密运算，因为 unique ID 是芯片生产厂出厂时就已经设定好的，无法改变，也无法复制。在这种情况下你即使正确复制了芯片的全部内容（包刮后面的介绍的 OTP Area 内容），更换新芯片后也是无法正常运行的。

例：MT29F4G08ABA 有 16+16 字节的 unique ID 号，只能读取，不能改写。

应用电路是否使用这种方法进行加密，需用户跟据芯片的数据手册确定芯片有没有 **unique ID**。如果有，并且应用时使用了 **Unique ID** 加密，则不可以更换新芯片，除非你能修改嵌入式系统的软件。

- 3) OTP Area（OTP 一次性编程的存储区）

OTP Area 只是在部分 Nand Flash 芯片上存在，具体有没有请查看相关的芯片数据手册

例：MT29F4G08ABA 有 30 个页的（30Pages*2112 字节）的 OTP 数据，TL866II 暂没有加入对一些 NAND FLASH 芯片的 OTP Area 进行编程及读取操作，如果应用中使用了 OTP Area，则暂不能复制这部分数据。（如果有需要，可以联系我们）

6.2 新片复制步骤

6.2.1 选择文件与设置

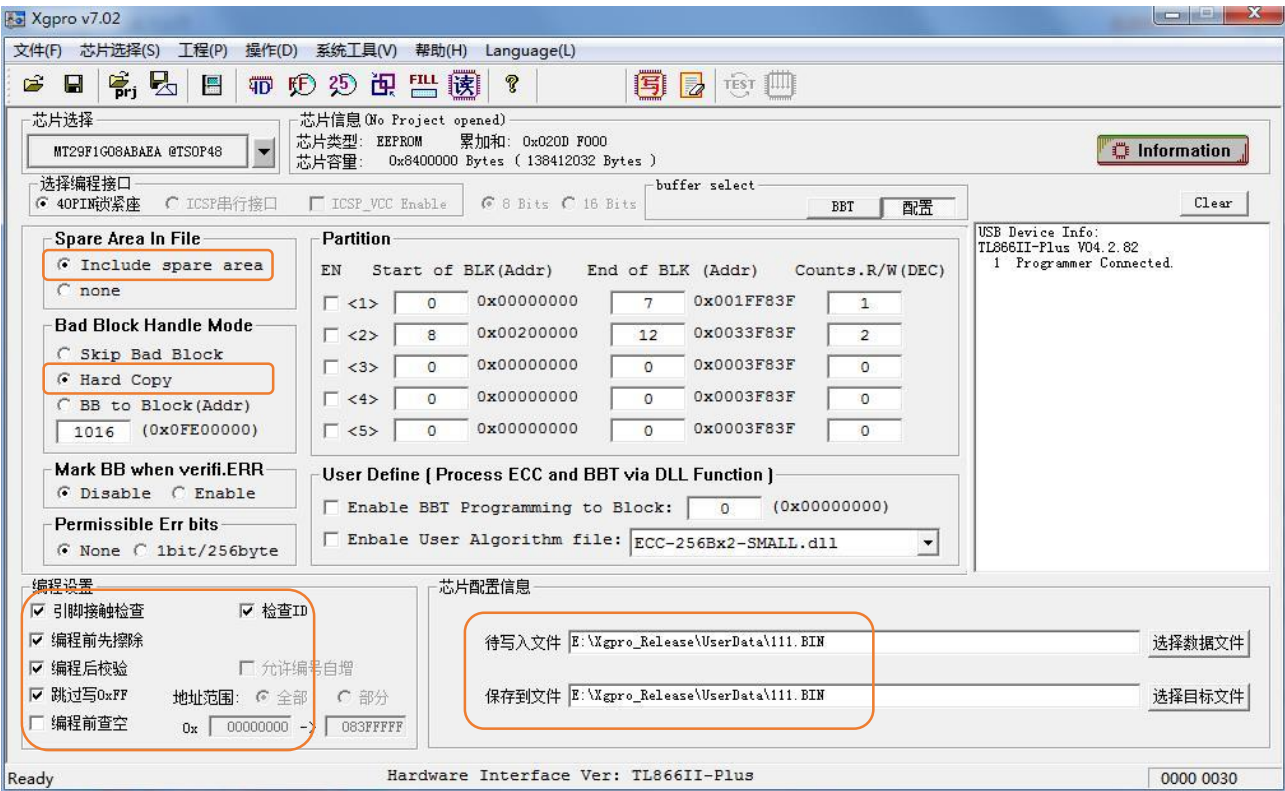
选择或输入文件名，<保存到文件>与<待写入文件>为相同一个文件名

Spare Area In File: Include Spare Area

Bad Block Handle Mode: Hard Copy

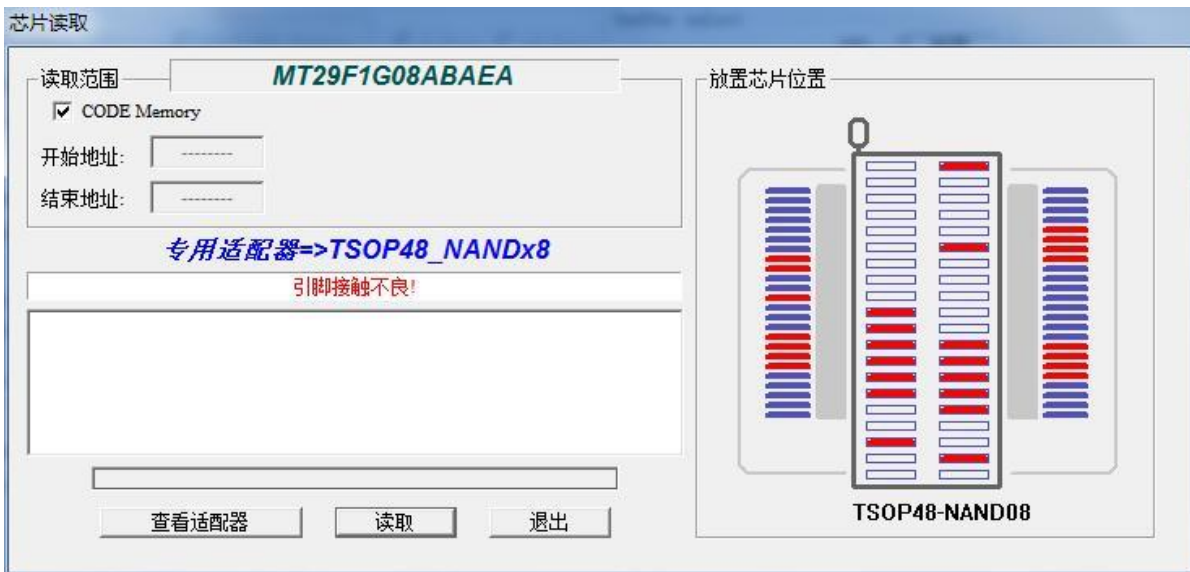
其他选项全部为默认设置。配置如下图 1-5 所示。

图 1-5:



6.2.2 读取芯片

击工具栏上的<读>按钮，用 NAND08 专用适配器，放好芯片。弹出如下对话框



注意：本编程有引脚接触检查功能，如果有使用到的引脚接触不良，会有接触不良提示，对个别芯片的少量引脚有可能不能使用引脚检查，在确保接触良好的情况下可以暂时取消引脚检查功能（在主界面左下角编程设置选项中）。

正常读取时，如下图所示：



直到完成读取。在读取完成后，可以在主菜单的<操作>选项中点击<数据校验>，进行一次数据校验，确保读入数据的正确性，（校验速度有可能大大快于读取时的速度是正常现象，因为 TL866II 编程器软件对空的数据块会在校验时自动跳过）

6.2.3 编程新芯片

更换新的芯片，点击工具栏上的<写>按钮，弹出如下界面：

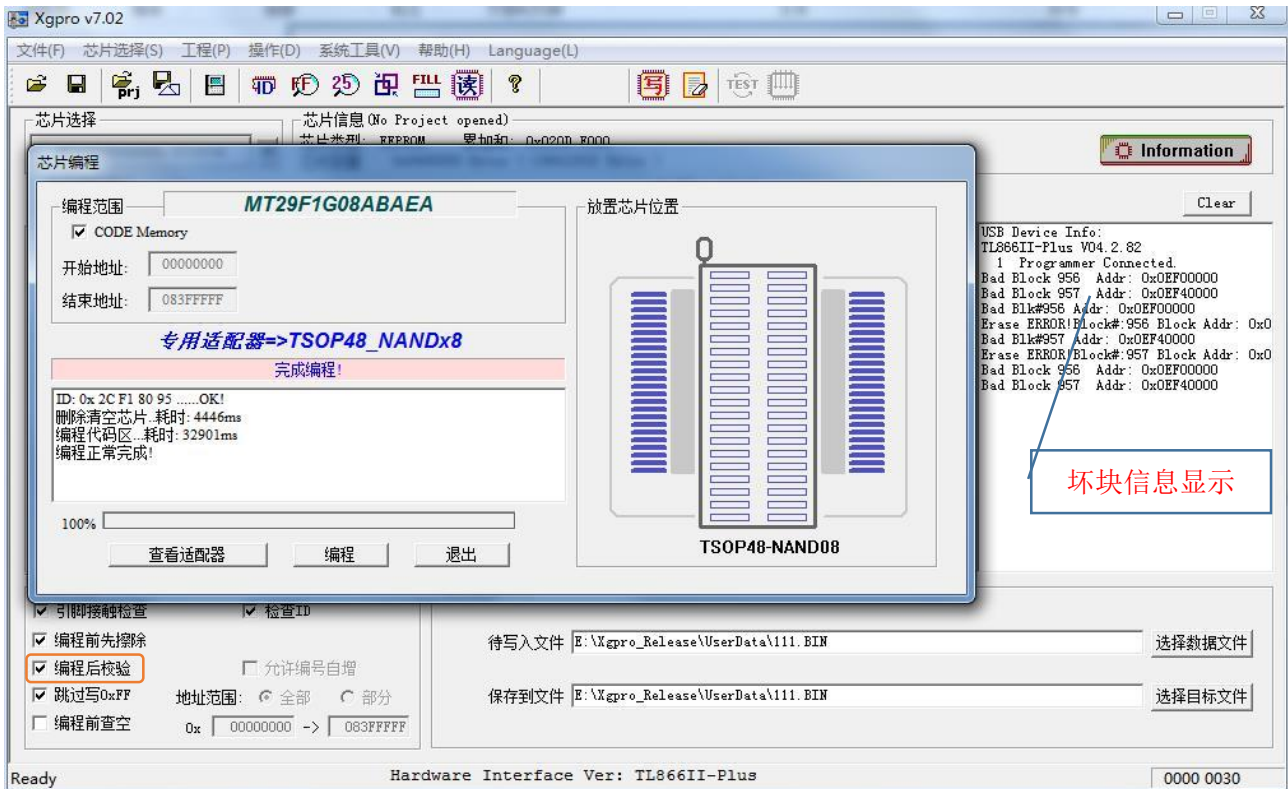


注：

- 烧写芯片的速度有可能比读取内容快，不用奇怪，因为在烧写时，软件自动对空的区域内容跳过不写。
- 在读取、删除、烧写过程中，芯片如果有坏块，会在主界面的右侧信息框内会出现坏块提示，但不影响对芯片的读写。
- 在复制前不需要有意去检查源芯片及新芯片的坏块情况，烧写时只要校验正确，没有出现错误中止。复制的新芯片就可用。新芯片中如果有坏块，在复制时软件发现此处为空数据，本软件能自动跳过。如果新芯片有坏块，源片又有数据需要写入这个坏块的位置，复制时就无法写入到此坏块处，则编程终止，请换用另一个新芯片。

因为维修技术员，一般并不能知道芯片数据结构如何、坏块处理方式如何。所以只能直接硬复制，TL866II 编程器软件，编程时对源片的空数据在写入新芯片时按跳过处理，这样避免了新片有坏块时，硬拷贝不能写入的问题

- 在 NAND Flash 烧写时编程与校验是按块进行的，只要在左下角<编程设置>中选中了<编程后校验>选项，编程完成后的同时检验也完成，不需要再次校验。



6.3 文件地址的计算方法

分二种情况计算：

- 1) 文件包含 Spare Area(冗余区)
数据在文件中的位置 = $nBlock(块号) \times (Page\ Size + Spare\ Size) \times 每块的页数$
- 2) 文件不含 Spare Area(冗余区)
数据在文件中的位置 = $nBlock(块号) \times Page\ Size \times 每块的页数$

7. 附录

7.1 免责条款

- (1) 未经我们正式授权的任何组织或个人修理、操作或改动过的产品，不在包修责任内。
- (2) 产品中锁紧座及其它人为的损坏，不在包修责任内。
- (3) 因为硬件故障或软件缺陷造成的连带扩展责任。

特别申明

由于编程器，支持的芯片种类多，软件系统复杂，错误或缺陷是难免的。在批量生产时，特别时对 **OTP** 一次性编程芯片，你必须确认编程正确无误后，才能进行批量生产。本公司不对因为硬件故障或软件缺陷造成的连带扩展损失负责。如发现问题或对我们的产品有好的宝贵意见，请联系我们。

7.2 保修条例

在正常使用范围内，从购买产品之日起一年内，若产品出现故障可免费保修。

7.3 联系方式

随着软件更新，本说明书内容仅供参考，以产品应用软件为准。
你可在 <http://www.xgecu.com/en> 下载最新的应用软件。

名称：海口鑫工电子有限公司

地址：海南省海口市丘海大道西 80 号 景润大酒店四层 邮编：570311

EMAIL: 532007590@QQ.com

TEL: 0898-68681816

7.4 常见问题

1. 设备不可用或错误

通常是 USB 驱动没有安装，或驱动被杀毒软件禁止，此时 USB 编程器上的黄灯快速闪动。

解决方法：参考 2.1.2 节，重装 USB 设备驱动。

2. 升级固件刷新时出现错误，无法刷新

通常情况为使用 USB 集线器的问题，刷新时不要使用外接 HUB，设备直接插到电脑的 USB 口上，进行升级刷新。如果还不能解决问题，可以在另一台电脑进行刷新操作。

3. 引脚接触不良

情况一：旧芯片，经常会有引脚接触不良提示，仔细处理引脚。

情况二：如果使用适配器，用万用表测量适配器引脚是否正常。

情况三：有个别芯片的个别引脚可能不支持引脚接触检查，这种情况可在主界面左下角，选项临时取消引脚检查功能。

4. 检测到 ID 错误

很多芯片内部有一个芯片的识别标志（Identification），内容一般有 2 个或 2 个以上字节组成，第一个字节一般是制造商代码（Manufacture ID），后面紧跟的是 芯片类型或芯片容量。芯片不同 ID 会不同，这个选项的功能是防止不正确的芯片放入。该选项是在对芯片读取或写入前，对芯片 ID 进行检查，如果正确继续，如果 ID 错误中止，该功能是可选的。默认是使能选项

同类芯片，厂家不同可能编程方法是一样的，对没有在列表中支持的芯片，可以选择其他厂商同容量同类芯片进行烧录。因为 ID 会不同，烧录前，取消 ID 检查即可

对很少部分单片机，有的芯片加密后不可读取芯片 ID，这时应在主界面中取消 ID 检查，否则 ID 检查错误。

5. 关于 27C 系列 VPP 电压为 21V 或 25V 的芯片编程

TL866II 编程电压 VPP 只支持到最高 18V，对早期的 27C 系列 ROM，用 21V 25V VPP 电压的芯片不再支持写入，但可以读取，如果用到，可用同型号的编程电压 VPP 为 13V 或 12V 的芯片替代，例：27C32 有 21V 13V 芯片购买时注意编程电压。芯片可直接替换。

不要使用外部电源提升编程电压进行强制编程，当芯片损坏的情况下，21V/25V 高压有可能回窜至编程器内部，将会损坏编程器硬件。