

## 引言

STM32 ST-LINK Utility软件通过ST-LINK和ST-LINK/V2工具加快开发环境中STM32微控制器系列的系统内编程。

本用户手册介绍了STM32 ST-LINK Utility的软件功能。在使用STM32 ST-LINK Utility时，建议下载STM8和STM32微控制器的ST-LINK在线调试器/编程器用户手册（UM0627）或STM8和STM32微控制器的ST-LINK/V2在线调试器/编程器用户手册（UM1075），它们提供了ST-LINK工具的更多相关信息。

**注：** STM32 ST-LINK Utility软件的产品编号为STSW-LINK004。

# 目录

<b>1</b>	<b>入门指南</b>	<b>6</b>
1.1	系统要求	6
1.2	硬件要求	6
1.3	安装STM32 ST-LINK Utility	6
1.4	卸载STM32 ST-LINK Utility	7
<b>2</b>	<b>STM32 ST-LINK Utility用户界面</b>	<b>8</b>
2.1	主窗口	8
2.2	菜单栏	9
2.2.1	文件菜单	9
2.2.2	编辑菜单	10
2.2.3	查看菜单	10
2.2.4	目标菜单	11
2.2.5	ST-LINK菜单	13
2.2.6	外部加载程序菜单	13
2.2.7	Help菜单	16
<b>3</b>	<b>STM32 ST-LINK Utility的功能</b>	<b>17</b>
3.1	器件信息	17
3.2	设置	18
3.3	存储器显示和修改	19
3.4	Flash存储器擦除	21
3.5	器件编程	21
3.6	选项字节配置	24
3.7	MCU内核功能	29
3.8	自动模式功能	30
3.9	为外部存储器开发自定义加载程序	31
3.9.1	Loader_Src.c文件	31
3.9.2	Dev_Inf.c文件	32
3.10	通过SWO查看器查看Printf	34
<b>4</b>	<b>STM32 ST-LINK Utility命令行接口（CLI）</b>	<b>35</b>
4.1	命令行的使用	35

---

4.1.1	连接和存储器操作命令 .....	35
4.1.2	内核命令 .....	37
4.1.3	Flash命令 .....	38
4.1.4	其他命令 .....	39
4.1.5	选项字节命令 .....	40
4.1.6	外部存储器命令25 .....	46
4.1.7	ST-LINK_CLI返回代码 .....	47
<b>5</b>	<b>STM32 ST-LINK Utility外部加载程序开发 .....</b>	<b>48</b>
5.1	Loader_Src.c文件 .....	48
5.2	Dev_Inf.c文件 .....	49
<b>6</b>	<b>版本历史 .....</b>	<b>50</b>

表格索引

表1. STM32F04x的nBoot1配置 ..... 44

表2. STM32F0和STM32F3的nBoot1配置 ..... 44

表3. ST-LINK\_CLI返回代码 ..... 47

表4. 文档版本历史 ..... 50

表5. 中文文档版本历史 ..... 53



## 图片索引

图1.	STM32 ST-LINK Utility用户界面主窗口.....	8
图2.	菜单栏.....	9
图3.	文件菜单.....	9
图4.	编辑菜单.....	10
图5.	查看菜单.....	10
图6.	目标菜单.....	11
图7.	ST-LINK菜单.....	13
图8.	外部加载程序菜单.....	13
图9.	外部加载程序窗口.....	14
图10.	外部加载程序子菜单.....	14
图11.	外部存储器网格.....	15
图12.	Help菜单.....	16
图13.	主用户界面中的器件信息区.....	17
图14.	“设置”对话框.....	18
图15.	STM32 ST-LINK Utility用户界面.....	20
图16.	Flash存储器映射对话框.....	21
图17.	打开文件对话框.....	22
图18.	器件编程对话框（编程）.....	23
图19.	选项字节对话框.....	27
图20.	读/写保护模式.....	28
图21.	MCU内核面板对话框.....	29
图22.	自动模式.....	30
图23.	串行线查看器窗口（SWV）.....	34

# 1 入门指南

本章介绍安装STM32 ST-LINK Utility软件的要求和步骤。

## 1.1 系统要求

STM32 ST-LINK Utility对PC配置的最低要求如下：

- 具有USB端口和Intel® Pentium®处理器的PC，运行以下Microsoft®操作系统之一的32位版本：
  - Windows® XP
  - Windows® 7
  - Windows® 10
- 256M字节RAM
- 有30 Mb硬盘空间可用

## 1.2 硬件要求

STM32 ST-LINK Utility可与以下器件一起使用：

- STM32F0、STM32F1、STM32F2、STM32F3、STM32F4、STM32L4、STM32L1、STM32L0和STM32W系列
- ST-LINK或ST-LINK/V2或ST-LINK/V2-ISOL在线调试器/编程器探头

*注：* 在本文档中，ST-LINK/V2指ST-LINK/V2和ST-LINK/V2-ISOL，它们的功能相同。

## 1.3 安装STM32 ST-LINK Utility

按照以下步骤和屏幕上的说明安装STM32 ST-LINK Utility：

1. 从ST网站下载STM32 ST-LINK Utility软件压缩文件。
2. 将.zip文件的内容解压缩至临时目录。
3. 双击解压缩后的可执行文件`setup.exe`开始安装，并按照屏幕上的提示在开发环境中安装STM32 ST-LINK Utility。实用工具的文档位于安装STM32 ST-LINK Utility的子目录\Docs中。

*注：* 如果已经安装了STM32 ST-LINK Utility软件的更早版本，则按照第 1.4节提供的卸载说明卸载该版本，然后再安装新版本。

## 1.4 卸载STM32 ST-LINK Utility

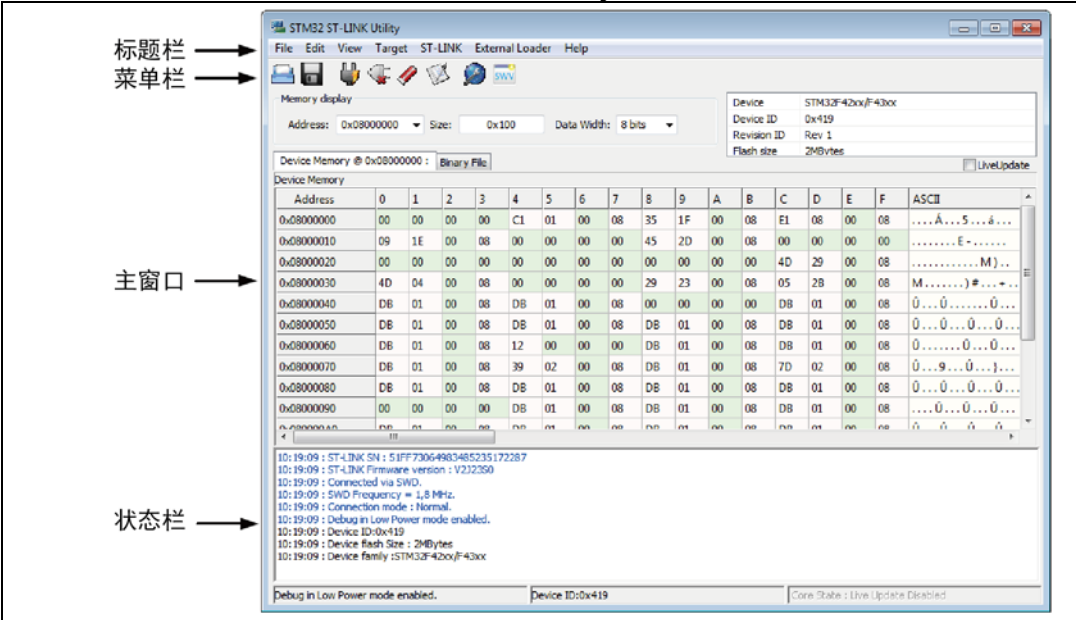
按照以下步骤卸载STM32 ST-LINK Utility：

1. 选择**开始 | 设置 | 控制面板**。
2. 双击**添加或删除程序**。
3. 选择**STM32 ST-LINK Utility**。
4. 点击**删除**按钮。

## 2 STM32 ST-LINK Utility用户界面

### 2.1 主窗口

图1. STM32 ST-LINK Utility用户界面主窗口



主窗口由三个区和三栏组成，如 图 1 所示：

- 存储器显示区
- 器件信息区
- 存储器内容区
  - 选中**实时更新**复选框可实时更新存储器数据（第 3.3 节中描述了此功能的更多细节）
- 标题栏：当前菜单的名称
- 菜单栏：使用菜单栏访问STM32 ST-LINK Utility的以下功能：
  - 文件菜单
  - 编辑菜单
  - 查看菜单
  - 目标菜单
  - 帮助菜单
- 第 2.2 节中描述了这些菜单的更多细节。
- 状态窗口：状态栏显示：
  - 连接状态和调试接口
  - 器件ID
  - 内核状态（仅在“实时更新”功能激活并选中了存储器网格时激活）

STM32 ST-LINK Utility用户界面还提供额外的表单和描述性的错误弹出消息。



2.2 菜单栏

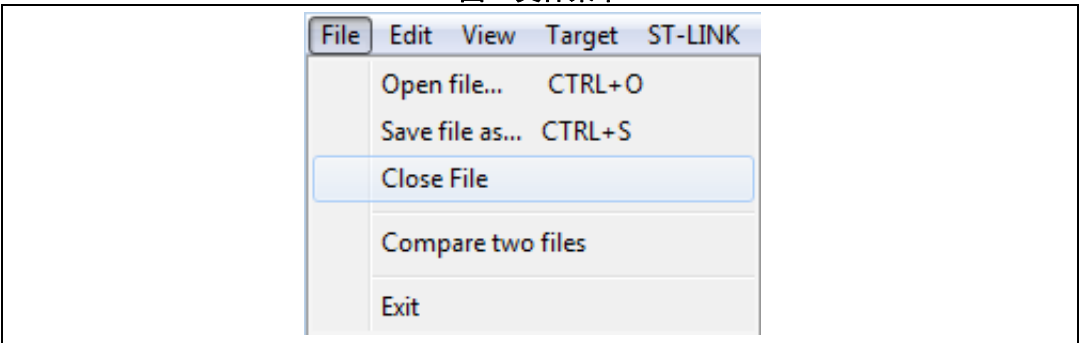
菜单栏（图 2）允许用户探索STM32 ST-LINK Utility软件的功能。

图2. 菜单栏



2.2.1 文件菜单

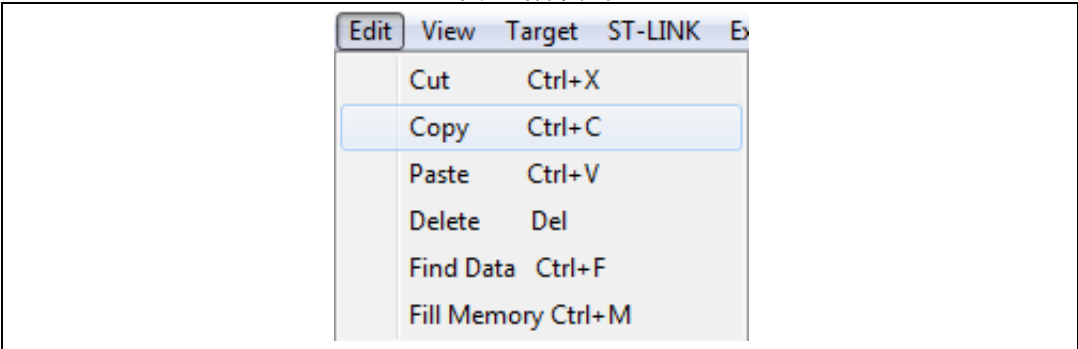
图3. 文件菜单



- |          |   |
|----------|---|
| 打开文件...  | 打开二进制、Intel十六进制或Motorola S-record文件。                                  |
| 文件另存为... | 将存储器面板上的内容保存为二进制、Intel十六进制或Motorola S-record文件。                       |
| 关闭文件     | 关闭加载的文件。  |
| 比较两个文件   | 比较两个二进制、十六进制或srec文件。不同之处以红色显示在文件面板上。如果文件某部分的地址范围在另一个文件中不可用，这部分将显示为紫色。 |
| 退出       | 关闭STM32 ST-LINK Utility程序。  |

2.2.2 编辑菜单

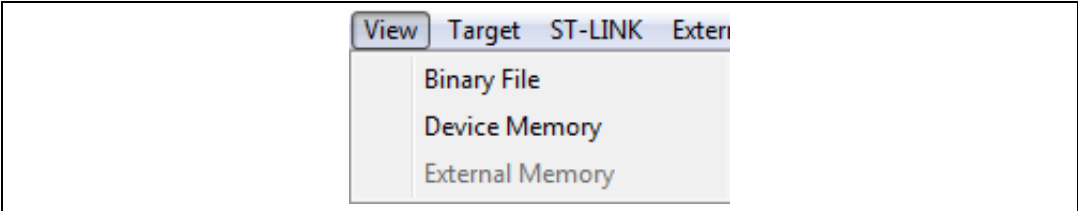
图4. 编辑菜单



- 剪切** 剪切在文件或存储器网格中选中的单元格。
- 复制** 复制在文件或存储器网格中选中的单元格。
- 粘贴** 粘贴位于文件或存储器网格中选中位置的已复制单元格。
- 删除** 删除在文件或存储器网格中选中的单元格。
- 查找数据** 在文件或存储器网格中查找二进制或十六进制格式的数据。
- 填充存储器** 使用选定的数据从选定的地址开始按SIZE填充文件或存储器网格。

2.2.3 查看菜单

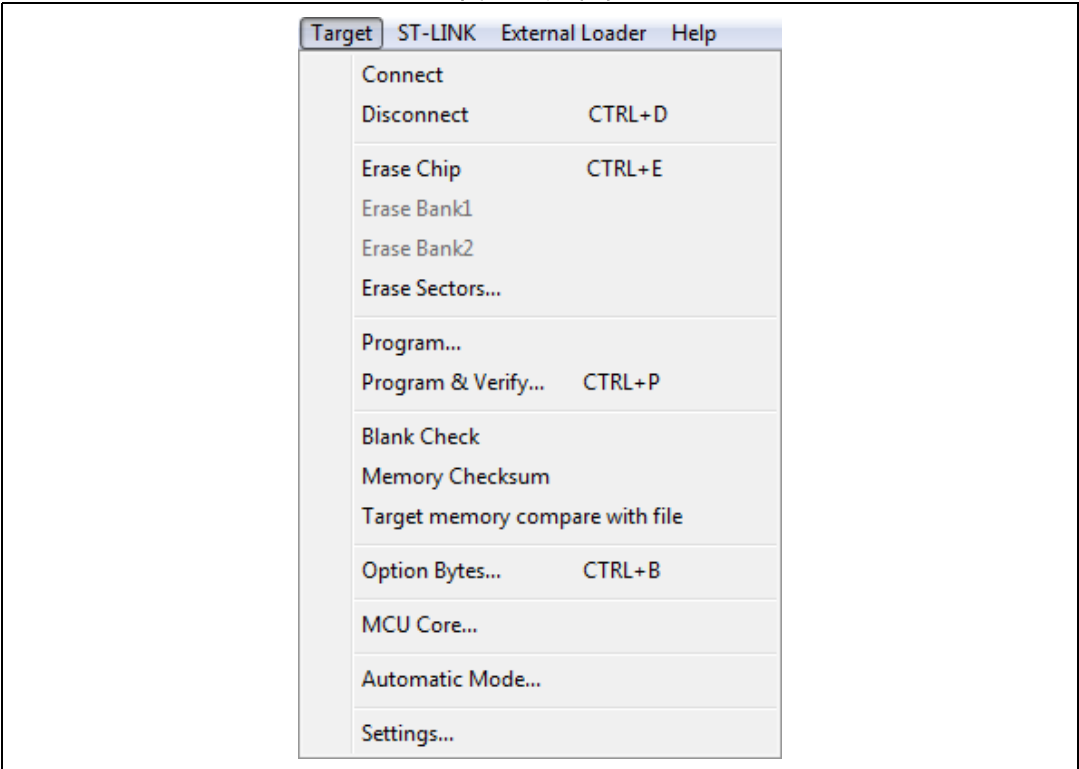
图5. 查看菜单



- 二进制文件** 显示加载的二进制文件的内容。
- 器件存储器** 显示器件存储器的内容。
- 外部存储器** 显示外部存储器的内容。

2.2.4 目标菜单

图6. 目标菜单



连接	连接目标器件并在器件信息区显示“器件类型”、“器件ID”和“Flash存储器大小”。
断开	断开与目标器件的连接。
擦除芯片	执行Flash存储器批量擦除，然后在存储器面板上显示Flash存储器的内容。
擦除存储区1	擦除Flash存储器的存储区1。仅当连接到XL容量器件时，才使能该菜单。
擦除存储区2	擦除Flash存储器的存储区2。仅当连接到XL容量器件时，才使能该菜单。
擦除扇区...	使用擦除扇区对话框窗口选择要擦除的扇区（参见 <a href="#">第 3.4节：Flash存储器擦除</a> 了解详细信息）。
编程...	将二进制、Intel十六进制或Motorola S-record文件加载到器件存储器（Flash或RAM）中。为此，选中一个二进制、Intel十六进制或Motorola S-record文件，在编程对话框窗口中输入起始地址（文件在器件中的放置位置），然后点击编程按钮（参见 <a href="#">第 3.5节：器件编程</a> ）。
编程和验证...	将二进制、Intel十六进制或Motorola S-record文件加载到器件存储器（Flash或RAM）中，然后执行编程数据的验证。
空白检查	确认STM32 Flash存储器为空白状态。如果Flash存储器不是空白状态，将在提示消息中突出显示第一个包含数据的地址。
存储器校验和	计算指定存储区的校验和值，该存储区由主窗口的存储器显示区中的地址和大小字段来定义。基于算术和算法按位计算校验和。结果截断为32位字。校验和值显示在日志窗口。

比较器件存储器和文件	将MCU器件存储器的内容与二进制、十六进制或srec文件进行比较。不同之处 以红色显示在文件面板上。
选项字节...	打开选项字节对话框窗口（参见第 3.6 节：选项字节配置了解详细信息）。
MCU内核...	打开“MCU内核”对话框窗口（参见第 3.7 节：MCU内核功能了解详细信息）。
自动模式...	打开“自动模式”对话框窗口（参见第 3.8 节：自动模式功能了解详细信息）。
设置...	<p>“设置”对话框允许用户选择一个ST-LINK探头并定义其连接设置。</p> <p>ST-LINK探头列表包含连接到计算机的所有探头的序列号。</p> <p>如果在显示“设置”对话框时插入或拔出某些ST-LINK探头，使用“刷新”按钮可更新ST-LINK探头列表。当用户选中一个探头时，将显示固件版本和连接的目标（取决于连接设置）。此后，用户可以选择调试接口（JTAG或SWD），并选择要连接的访问端口（当器件包含多个访问端口时）。</p> <p>还可以选择复位类型：</p> <ul style="list-style-type: none"><li>* 使用“复位状态下连接”选项可在执行任何指令之前连接到目标。这在很多情况下是很有用的，例如当目标包含了禁用JTAG/SWD引脚的代码时。</li><li>使用“热插拔”选项可在不停机或复位的情况下连接到目标。这对于在应用运行时更新RAM地址或IP寄存器非常有用。</li></ul> <p>当通过ST-LINK/V2连接目标时，“供电电压”组合框显示目标电压。</p> <p>当使用ST-LINK连接STM32F2或STM32F4器件时，“供电电压”组合框允许用户选择能够正确完成Flash存储器编程的目标供电电压。</p> <p>使用“在低功耗模式下使能调试”选项可连接处于低功耗模式的器件。</p> <p>如果任何连接设置发生变化，对话框会尝试标识具有新的连接设置的目标。</p>

注：

“复位状态下连接”选项只对ST-LINK/V2的SWD模式可用。

对于JTAG模式，自ST-LINK/V2固件版本V2J15Sx起提供“复位状态下连接”。

JTAG连接器（引脚15）的RESET引脚应连接到器件复位引脚。

“热插拔”选项在SWD模式下可用。

当用户断开与目标的连接时，低功耗模式禁用。

对于JTAG模式，自ST-LINK固件版本V2J15Sx起提供“热插拔”。

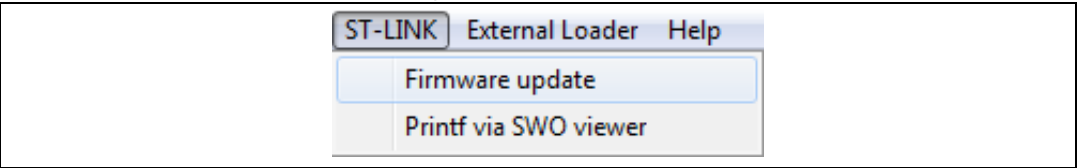
要在多探头的环境下使用的ST-LINK固件版本应为：

- V1J13S0或更高ST-LINK固件版本。
- V2J21S4或更高ST-LINK/V2固件版本。
- V2J21M5或更高ST-LINK/V2-1固件版本。

当另一应用使用ST-LINK/V2或ST-LINK/V2-1探头时，将不显示该探头的序列号且其不能在ST-LINK Utility的当前实例中使用。

### 2.2.5 ST-LINK菜单

图7. ST-LINK菜单



固件更新

显示ST-LINK和ST-LINK/V2固件的版本并更新至最新版本：

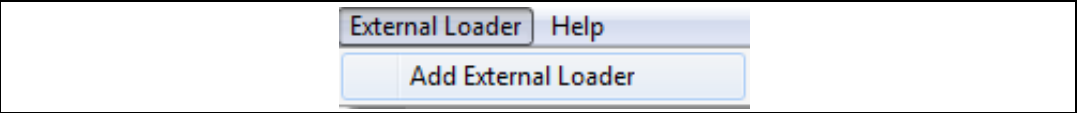
ST-LINK: V1J13S0  
ST-LINK/V2: V2J21S4  
ST-Link/V2-1: V2J21M5

通过SWO查看器发送Printf

显示通过SWO从目标发送的printf数据（参见 [第 3.10 节：通过SWO查看器查看Printf](#) 了解详细信息）。

### 2.2.6 外部加载程序菜单

图8. 外部加载程序菜单



STM32 ST-LINK Utility包含**添加外部加载程序**子菜单，它允许用户选择将被ST-LINK Utility用来读取、编程或擦除外部存储器的外部加载程序。

必须将外部加载程序添加到ST-LINK utility目录下的ExternalLoader目录中（参见 [第 3.9 节：为外部存储器开发自定义加载程序](#) 了解关于如何创建自定义加载程序的详细信息）。

如果在**外部加载程序**对话框中选中了外部加载程序（参见 [图 9](#)），将为每个选中的外部加载程序显示一个新的子菜单。

这些子菜单提供相应外部加载程序提供的所有功能（编程、扇区擦除等）（参见 [图 10](#)）。

图9. 外部加载程序窗口

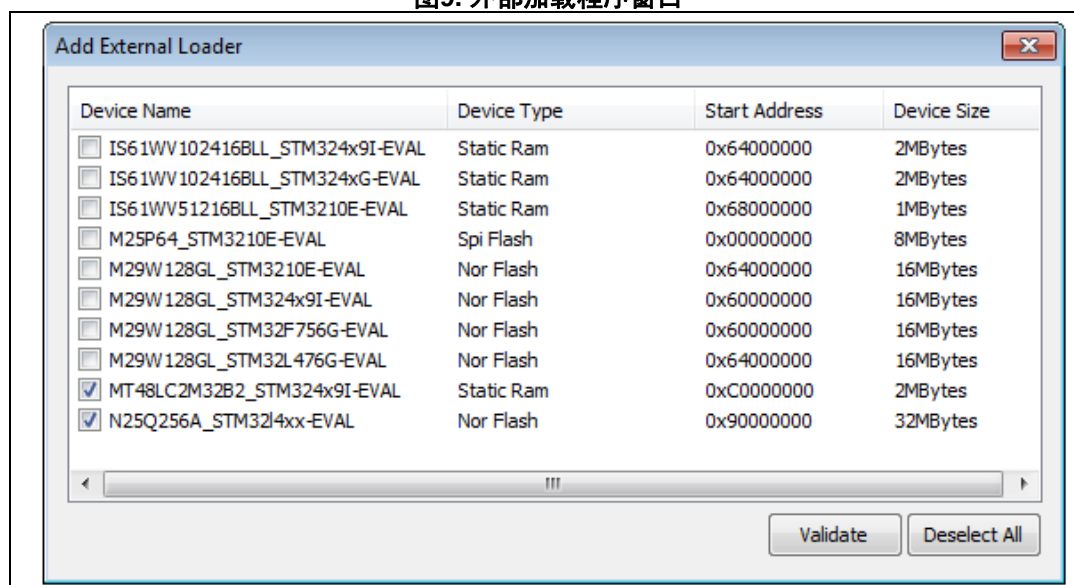
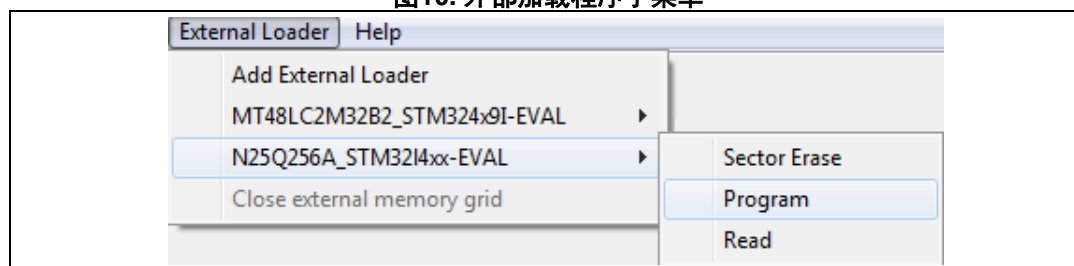


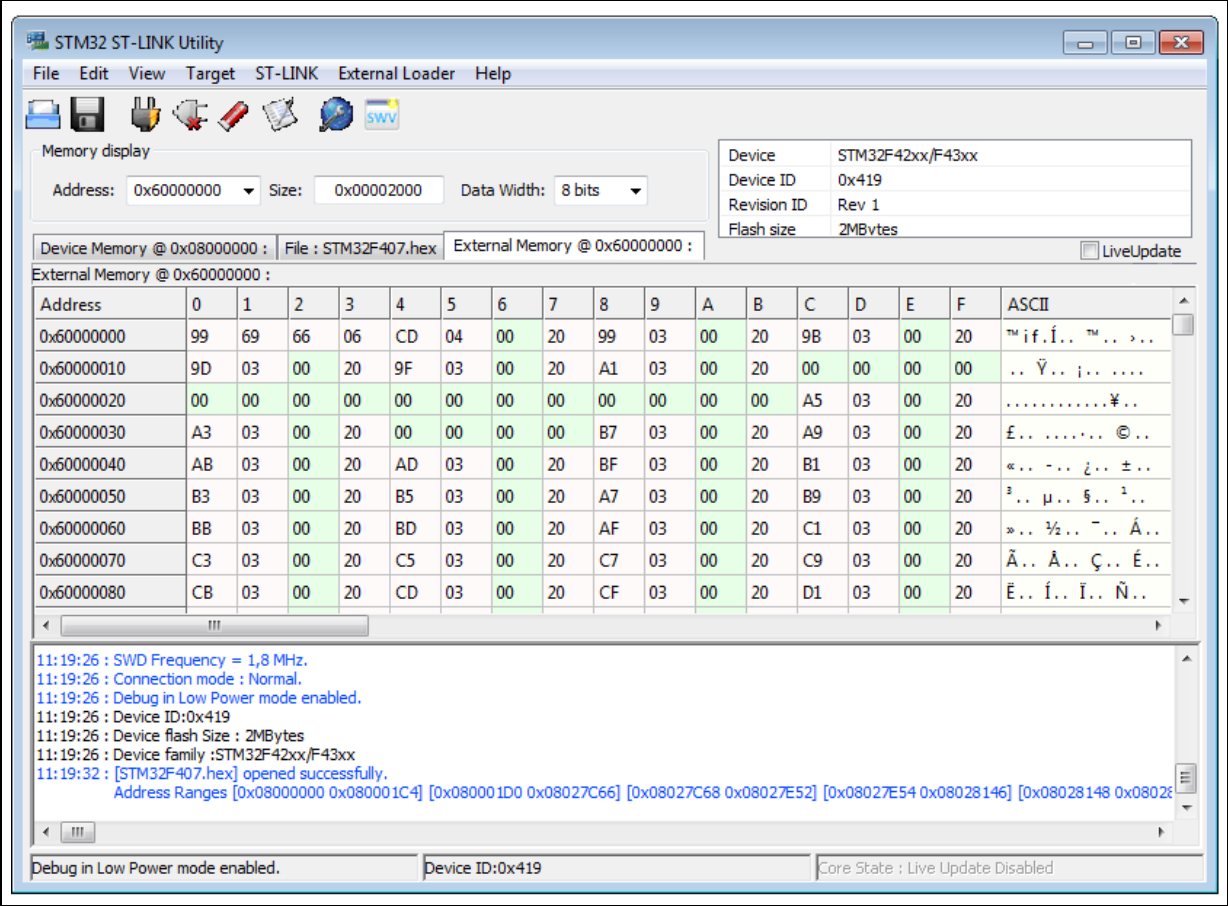
图10. 外部加载程序子菜单



外部存储器的内容显示在**外部存储器**网格中（[图 11](#)）。

选择**关闭外部存储器网格**子菜单可关闭**外部存储器**网格窗口。

图11. 外部存储器网格

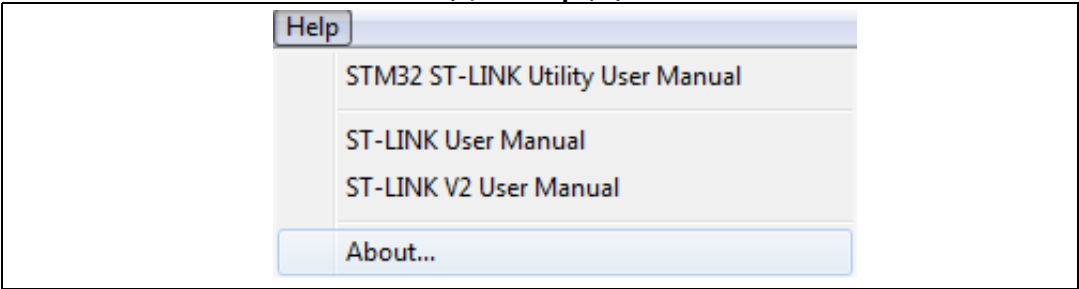


注： 一次只能选择10个外部加载程序。

如果已使用**添加外部加载程序**子菜单添加了相应的外部加载程序，则可通过**器件存储器网格**自动访问通过FSMC连接到STM32的外部存储器。

2.2.7 Help菜单

图12. Help菜单



STM32 ST-LINK Utility用户手册	打开STM32 ST-LINK Utility用户手册。
ST-LINK用户手册	打开ST-LINK用户手册。
ST-LINK/V2用户手册	打开ST-LINK/V2用户手册。
关于...	显示STM32 ST-LINK Utility软件版本和版权信息。



### 3 STM32 ST-LINK Utility的功能

本章将详细描述如何使用STM32 ST-LINK Utility的功能：

- [器件信息](#)
- [设置](#)
- [存储器显示和修改](#)
- [Flash存储器擦除](#)
- [器件编程](#)
- [选项字节配置](#)
- [MCU内核功能](#)
- [自动模式功能](#)

#### 3.1 器件信息

器件信息区显示如 [图 13](#)所示的信息。

图13. 主用户界面中的器件信息区

Device	STM32F40xx/F41xx
Device ID	0x413
Revision ID	Rev Z
Flash size	1MBytes

- 器件：

连接的STM32器件所属的系列。每种器件类型包含许多具有不同特性（例如Flash存储器大小、RAM大小和外设）的器件。
- 器件ID：

外部PPB存储器映射中的MCU器件识别码。
- 版本ID：

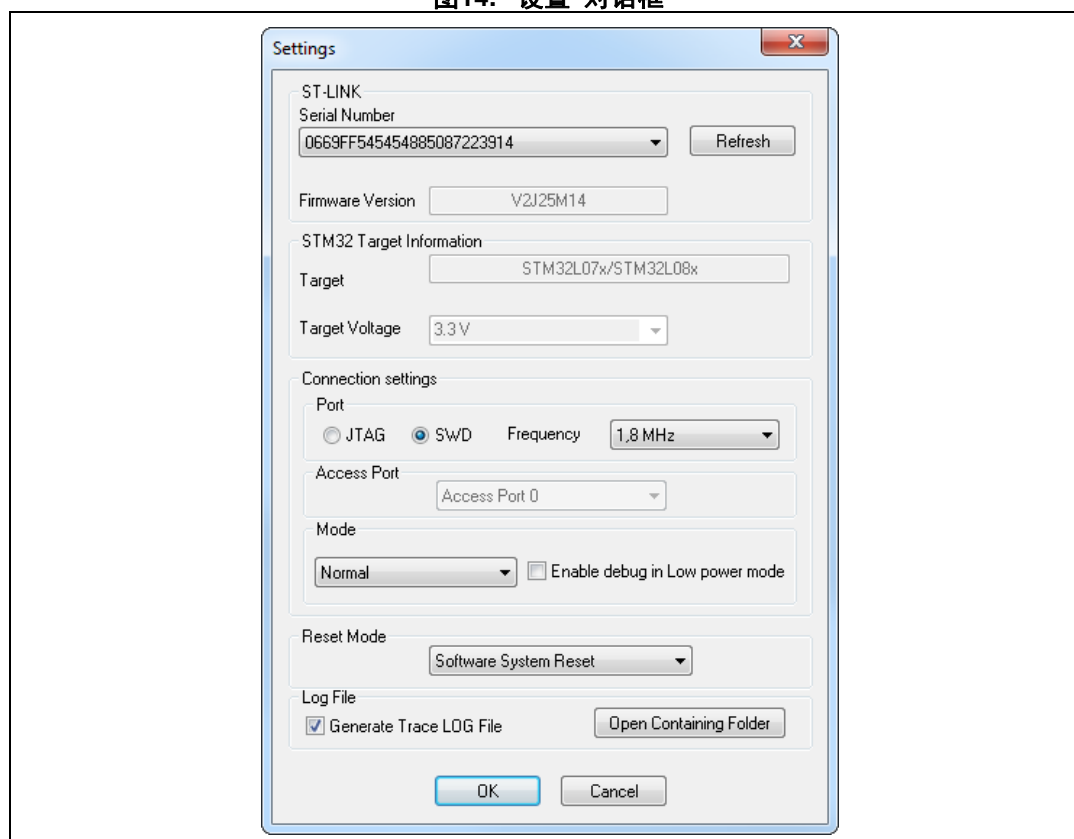
连接的MCU器件的版本ID。
- Flash大小：

片上Flash存储器的大小。

## 3.2 设置

图 14所示的“设置”对话框显示关于连接的ST-LINK探头和STM32目标的有用信息，并用于配置连接设置。

图14. “设置”对话框



用户可以根据探头序列号或连接的目标（显示在STM32目标信息区）来选择一个已连接的ST-LINK探头进行使用。

当使用ST-LINK/V2或ST-LINK/V2-ISOL时，将测量目标电压并显示在STM32目标信息区。

可用的连接设置：

- 端口：JTAG或SWD
- 访问端口（当器件包含多个访问端口时）
- 频率
- 模式：
  - Normal  
使用“Normal”连接模式时，目标是休眠的，随后挂起。使用“Reset Mode”选项来选择复位的方式
  - 复位状态下连接  
“复位状态下连接”选项能够在执行指令之前使用复位向量捕获连接到目标。这在很多情况下是很有用的，例如当目标包含了禁用JTAG/SWD引脚的代码时。
  - 热插拔  
使用“热插拔”选项可在不停机或复位的情况下连接到目标。这对于在应用运行时更新RAM地址或IP寄存器非常有用。
- 在低功耗模式下使能/禁用调试
- 使能/禁用跟踪日志文件生成

可用的复位模式：

- 软件系统复位
- 硬件复位
- 内核复位

*注：在选择“复位状态下连接”模式时，会自动选择硬件复位模式。在进行选项字节编程时，会在操作结束时发出复位请求。此类复位将单独处理且不受这些选项的影响。*

### 3.3 存储器显示和修改

除了器件信息区，主窗口还包含其他两个区：

- 存储器显示
- 存储器数据

**存储器显示：**该区包含三个编辑框：

**地址：**                    用户要读取的存储区起始地址。

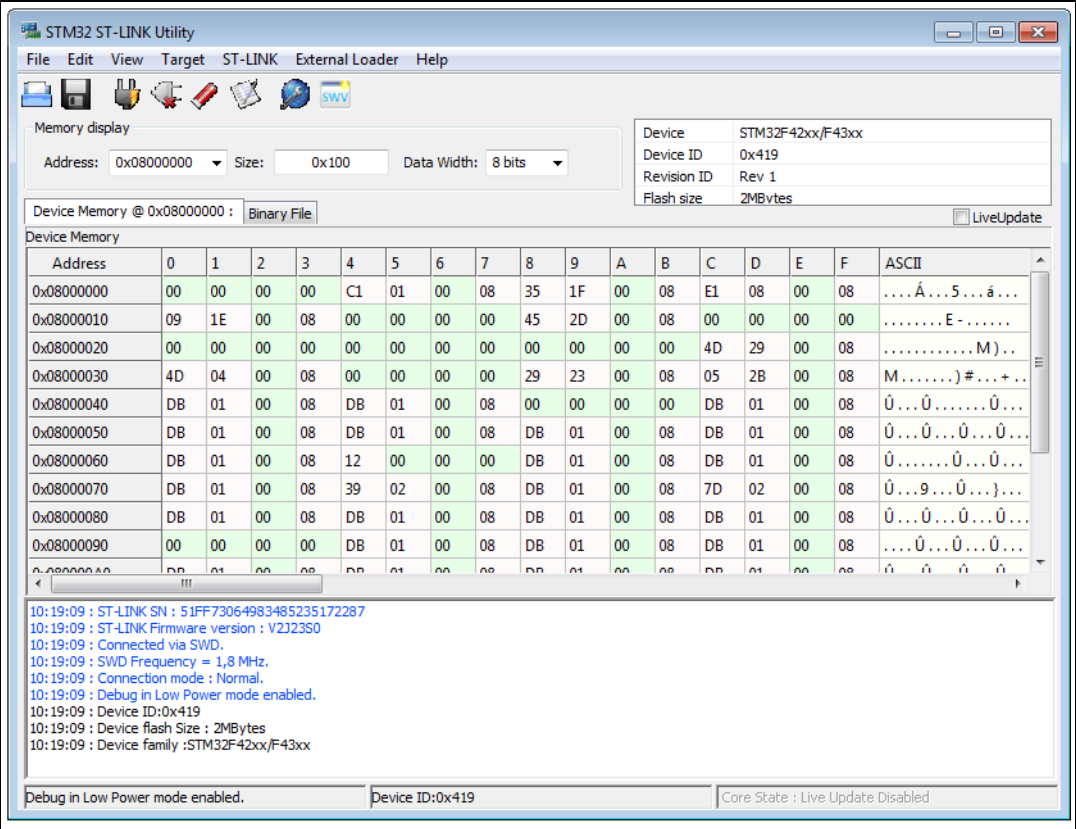
**大小：**                    要读取的数据量。

**数据宽度：**              显示的数据的宽度（8位、16位或32位）。

**存储器数据：**该区域显示从文件读取的数据或连接的器件的存储器内容。在下载前，用户可以修改文件内容。

- 要使用该区域显示二进制、Intel十六进制或Motorola S-record文件的内容，请转至**文件 | 打开文件...**
- 要使用该区域读取和显示所连接器件的存储器内容，请在 **存储器显示**区输入存储区起始地址、数据大小和数据宽度，然后按下**Enter**键。
- 在读取数据后，用户还可以修改每个值，方法是双击相应的单元格，如图 15所示。用户还可以使用菜单**文件 | 文件另存为...**，将器件存储器内容保存到二进制、Intel十六进制或Motorola S-record文件中。
- 在使用“实时更新”功能时，器件存储器网格将实时更新，修改过的数据显示为红色。

图15. STM32 ST-LINK Utility用户界面



**注：**当存储器数据区显示器件存储器内容时，任何修改都将自动应用于芯片。用户可以修改用户Flash存储器、RAM存储器和外设寄存器。  
对于STM32F2和STM32F4器件，可以直接从存储器数据区修改OTP区。

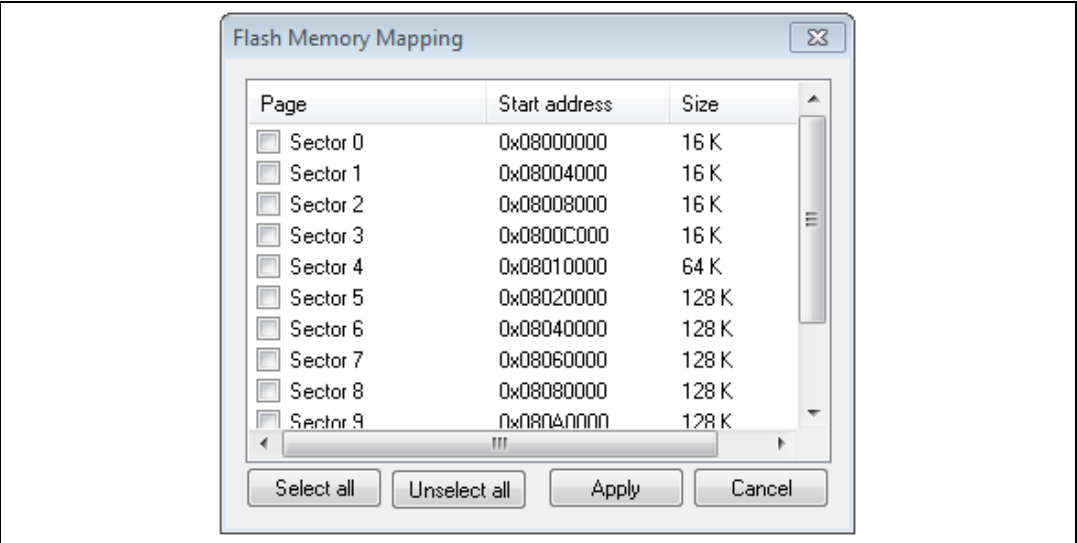


### 3.4 Flash存储器擦除

有两种类型的Flash存储器擦除：

- **Flash批量擦除：**擦除所连接器件的所有Flash存储器扇区。点击菜单**目标 | 擦除芯片**执行擦除。
- **Flash扇区擦除：**擦除选中的Flash存储器扇区。要选择扇区，请转至**目标 | 擦除扇区...**，系统随即显示**Flash存储器映射**对话框，用户可以从这里选择要擦除的扇区，如图16所示。
  - 使用**全选**按钮可选中所有Flash存储器页面。
  - 使用**取消全选**按钮可取消选中所有已选中页面。
  - 使用**取消**按钮可放弃擦除操作，即使已选中某些页面。
  - 使用**应用**按钮可擦除所有选中页面。

图16. Flash存储器映射对话框



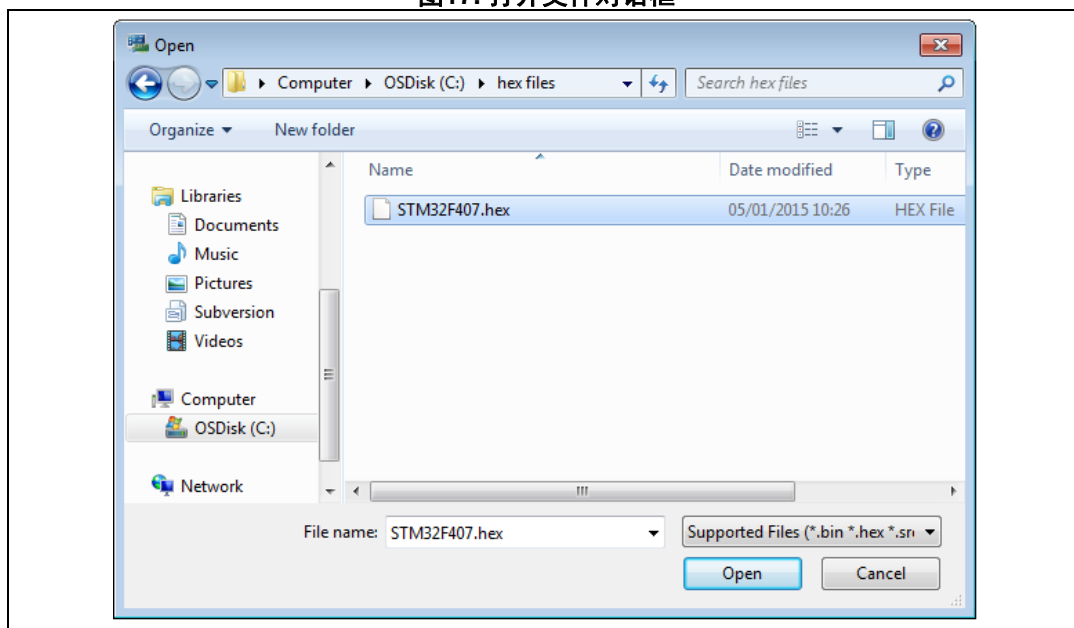
**注：**要擦除超低功耗STM32L1器件的Flash数据存储器扇区，请选中列表末尾的数据存储器复选框并点击**应用**。

### 3.5 器件编程

按照以下步骤，STM32 ST-LINK Utility可将二进制、十六进制或srec文件下载到Flash或RAM中：

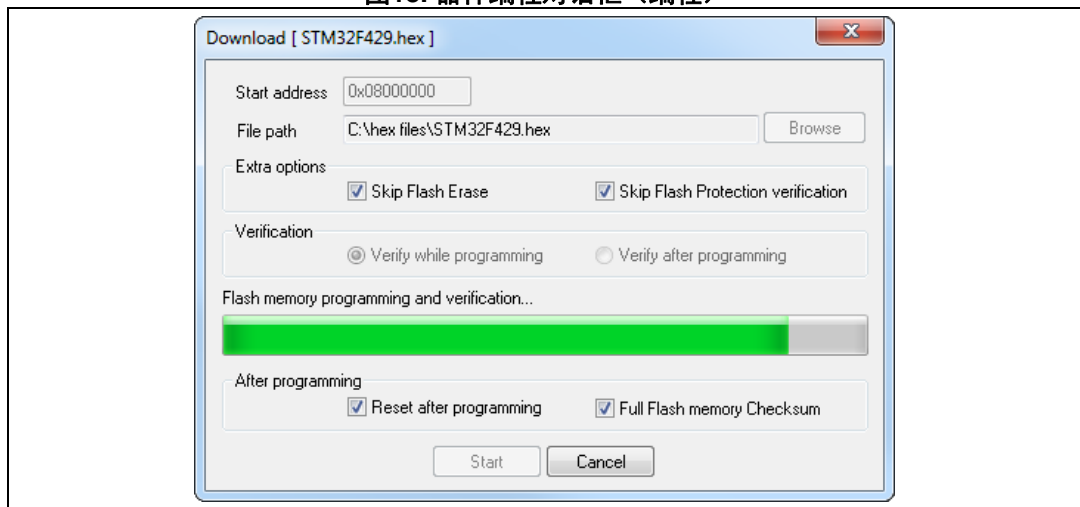
1. 点击**目标 | 编程...**（若用户想要 验证写入的数据，则点击**目标 | 编程和验证...**）打开**打开 文件**对话框，如图17所示。如果二进制文件已打开，则转至步骤3。

图17. 打开文件对话框



2. 选中二进制、Intel十六进制或Motorola S-record文件并点击**打开**按钮。
3. 指定开始编程的地址，如 [图 18](#)所示：它可以是Flash或RAM地址。

图18. 器件编程对话框（编程）



4. 如果器件已擦除，则选中“跳过Flash擦除”选项跳过Flash擦除操作。
5. 如果器件无保护，则选中“跳过Flash保护验证”选项跳过Flash保护验证。
6. 通过选中两个单选按钮之一选择验证方法：
  - a) 在编程时验证：快速片上验证法，可比较编程缓冲区内容（文件的一部分）与Flash存储器内容。
  - b) 在编程后验证：慢速但可靠的验证法，可在编程操作结束后读取所有已编程存储区，并将其与文件内容进行比较。
7. 最后，点击“开始”按钮开始编程：
  - a) 如果选中**编程后复位**复选框，将发出MCU复位请求。
  - b) 如果选中**完整Flash存储器校验**和复选框，将在编程操作后计算完整Flash存储器的校验和，并显示在日志窗口中。
8. 通过选中两个单选按钮之一选择验证方法：
  - a) 在编程时验证：快速片上验证法，可比较编程缓冲区内容（文件的一部分）与Flash存储器内容。
  - b) 在编程后验证：慢速但可靠的验证法，可在编程操作结束后读取所有已编程存储区，并将其与文件内容进行比较。
9. 最后，点击“开始”按钮开始编程：
  - a) 如果在第一步中选择了**目标 | 编程和验证...**，将在编程操作期间完成检查。
  - b) 如果选中**编程后复位**复选框，将发出MCU复位请求。

注：1 根据MCU供电电压，STM32F2和STM32F4系列支持不同的编程模式。当使用ST-LINK时，应在**目标 | 设置**菜单中指定MCU供电电压，以便能够以正确的模式进行器件编程。当使用ST-LINK/V2时，会自动检测供电电压。

如果器件受读保护，读保护会被禁用。如果Flash存储器页面受写保护，编程期间会禁用写保护，之后会再恢复。

注：2 用户可以对包含不同目标存储位置（内部Flash存储器、外部Flash存储器、选项字节等）的多个片段的十六进制/Srec文件进行编程。

如果将“读出保护”设定为2级（调试和从SRAM/系统内存自举功能禁用），将显示用于确认的消息框，以避免意外的芯片保护操作。

注：3 额外选项专用于无保护和已擦除器件上的编程操作。

## 3.6 选项字节配置

STM32 ST-LINK Utility可通过**选项字节**对话框（如 [图 19](#)所示）（通过**目标 | 选项字节...**访问）配置所有选项字节。

**选项字节**对话框包含以下几个部分：

- **读出保护**

修改Flash存储器的读保护状态。

STM32F0、STM32F2、STM32F3、STM32F4、STM32L4和STM32L1器件有以下读保护级别：

- 级别 0：无读保护
- 级别1：存储器读保护使能
- 级别2：存储器读保护使能且所有调试功能禁用。

对于其他器件，只能使能或禁用读保护。

- **BOR级别**

欠压复位级别。该列表包含激活/释放欠压复位的供电电压阈值。该选项仅对STM32L1、STM32L4、STM32F2、STM32F4和STM32F7器件可用。

对于STM32L4器件，有5个可编程VBOR阈值可供选择：

- BOR 级别 0：复位电压阈值为约1.7 V
- BOR 级别 1：复位电压阈值为约2.0 V
- BOR 级别 2：复位电压阈值为约2.2 V
- BOR 级别 3：复位电压阈值为约2.5 V
- BOR 级别 4：复位电压阈值为约2.8 V

对于超低功耗器件，有5个可编程VBOR阈值可供选择：

- BOR 级别 1：1.69 V 到 1.8 V 电压范围的复位阈值级别
- BOR 级别 2：1.94 V 到 2.1 V 电压范围的复位阈值级别
- BOR 级别 3：2.3 V 到 2.49 V 电压范围的复位阈值级别
- BOR 级别 4：2.54 V 到 2.74 V 电压范围的复位阈值级别
- BOR 级别 5：2.77 V 到 3.0 V 电压范围的复位阈值级别



对于STM32F2和STM32F4器件，有4个可编程VBOR阈值可供选择：

- BOR 级别 3：供电电压范围为2.70至3.60 V
- BOR 级别 2：供电电压范围为2.40至2.70 V
- BOR 级别 1：供电电压范围为2.10至2.40 V
- BOR 关闭：供电电压范围为1.62至2.10 V

- **用户配置选项字节**

- WDG\_SW：如选中，通过软件使能看门狗。否则，在上电时自动使能看门狗。
- IWDG\_STOP：如未选中，独立看门狗计数器在停止模式下冻结。如选中，该计数器在停止模式下处于激活状态。
- IWDG\_STBY：如未选中，独立看门狗计数器在待机模式下冻结。如选中，该计数器在待机模式下处于激活状态。
- WWDG\_SW：如选中，通过硬件选项位使能窗口看门狗。
- SRAM2\_RST<sup>(a)</sup>：此位允许用户使能在系统复位时擦除SRAM2。如选中，在发生系统复位时不擦除SRAM2。如未选中，在发生系统复位时擦除SRAM2。
- SRAM\_PE<sup>(a)</sup>：此位允许用户使能SRAM2硬件奇偶校验。如选中，将禁用SRAM2奇偶校验。
- DUALBANK<sup>(b)</sup>：如选中，512/256K双存储区Flash具有连续地址。
- DB1M<sup>(c)</sup>：1-Mb Flash存储器上的双存储区。
- PCROP\_RDP<sup>(a)</sup>：如选中，当RDP级别从级别1降至级别0时，将擦除PCROP区（完全批量擦除）。
- nRST\_SHDW<sup>(a)</sup>：如选中，将不生成复位。如未选中，将在进入关断模式时生成复位。
- nRST\_STOP：如未选中，将在进入待机模式时生成复位（1.8 V域掉电）。如选中，在进入待机模式时不生成复位。
- nRST\_STDBY：如未选中，将在进入停止模式时生成复位（所有时钟停止）。如选中，在进入停止模式时不生成复位。
- nBFB2：如未选中且自举引脚置位为让器件在启动时从用户Flash自举，则器件将从Flash存储区2自举；否则，将从Flash存储区1自举。该选项仅在连接至包含两个Flash存储区的器件时使能。
- nBoot1<sup>(d)</sup>：与BOOT0引脚一起，选择自举模式：
  - nBoot1选中/未选中且BOOT0 = 0 => 从主Flash存储区自举；
  - nBoot1选中且BOOT0 = 1 => 从系统存储区自举；
  - nBoot1未选中且BOOT0 = 1 => 从嵌入式SRAM自举。
- VDDA\_Monitor<sup>(d)</sup>：选择对VDDA电源的模拟监控：  
如选中，将使能VDDA电源监控器；否则，将禁用VDDA电源监控器。
- nSRAM\_Parity<sup>(d)</sup>：此位允许用户使能SRAM硬件奇偶校验。  
如选中，将禁用SRAM奇偶校验；否则，将使能SRAM奇偶校验。
- SDADC12\_VDD\_Monitor<sup>(e)</sup>：

a. 仅对STM32L4器件可用。

b. 仅对支持双存储区模式的STM32L4器件可用。

c. 仅在STM32F42x/STM32F43x 1-Mb器件上可用。

d. 仅对STM32F0和STM32F3器件可用。

如选中，将使能SDADC12\_VDD电源监控器；否则，将禁用SDADC12\_VDD电源监控器。

- nBoot0\_SW\_Cfg<sup>(a)</sup>：此位允许用户完全禁用BOOT0硬件引脚并使用用户选项位11（nBoot0）。

如选中，BOOT0引脚将绑定至GPIO引脚（对于LQFP32和更小封装为PB8，对于QFN32和更大封装为PF11）。

- **自举地址选项字节：**

对于支持BOOT\_ADDx的器件，它允许从选项字节BOOT\_ADDx定义的基址自举。

BOOT\_ADDx[15:0]对应地址[29:11]。

对于支持BOOT\_ADD0和BOOT\_ADD1的器件，则取决于BOOT0引脚：

- 如果BOOT0 = 0，从选项字节BOOT\_ADD0定义的基址自举。
- 如果BOOT0 = 1，从选项字节BOOT\_ADD1定义的基址自举。

用户可以输入自举地址或BOOT\_ADDx选项字节值。

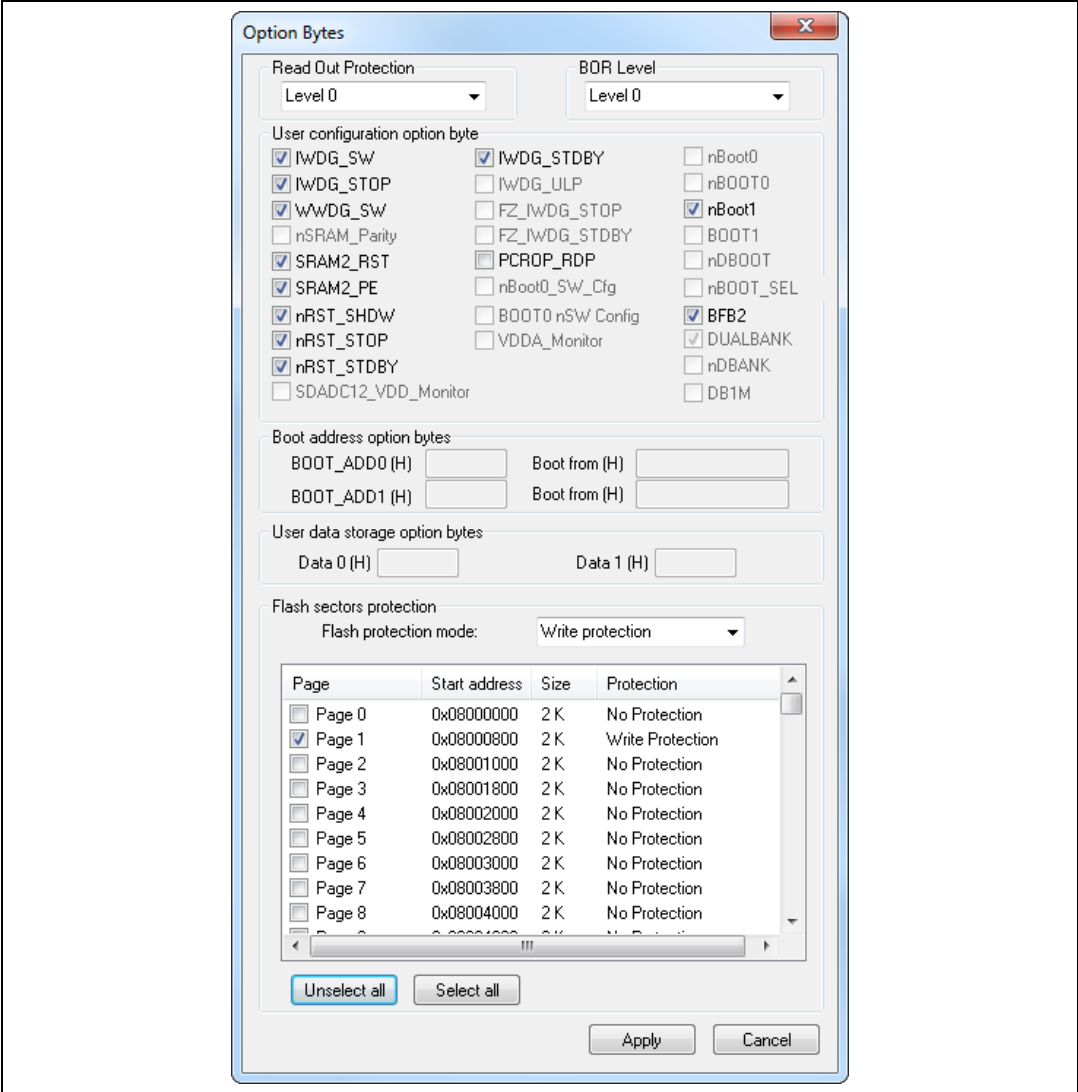
- **用户数据存储选项字节：**包含两个用于用户存储的字节。这两个选项字节对STM32F0、STM32F2、STM32F3、STM32F4和STM32L1器件不可用。
- **Flash扇区保护：**根据连接的器件，按定义的页数将Flash扇区分组。在这里，用户可以修改每个Flash扇区的写保护。
- 对于支持PCROP功能的器件，可以使能/禁用每个扇区的读保护。“Flash保护模式”允许用户选择读或写保护。

---

e. 仅STM32F37x器件提供。

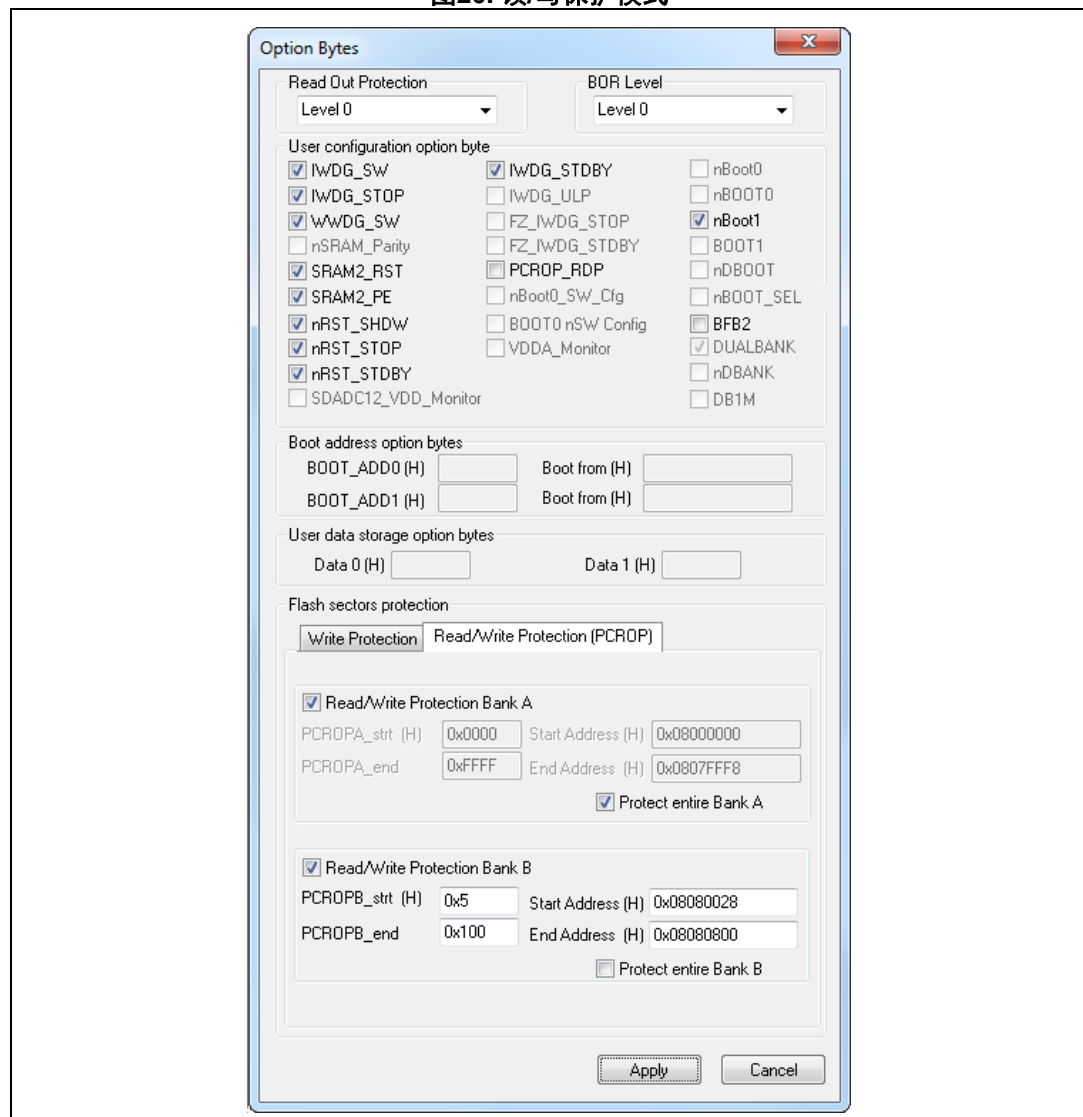
a. 仅STM32F04x器件提供。

图19. 选项字节对话框



对于STM32L4器件，通过两个地址定义PCROP区，如 图 20所示：

图20. 读/写保护模式



- 读/写保护存储区A：如选中，PCROPA\_STRT、PCROPA\_END、起始地址（H）和结束地址（H）字段可编辑，用户可以输入PCROP STRT/END字段或起始/结束地址。
- 保护整个存储区A：如选中，对整个存储区A进行pcrop保护。
- PCROPA\_strt：存储区A中受保护区的PCROP起始字段。
- 起始地址：由PCROPA\_strt字段定义的起始地址。
- PCROPA\_end：存储区A中受保护区的PCROP结束字段。
- 结束地址：由PCROPA\_end字段定义的结束地址。
- 读/写保护存储区B：如选中，PCROPB\_STRT、PCROPB\_END、起始地址（H）和结束地址（H）字段可编辑，用户可以输入PCROP STRT/END字段或起始/结束地址。

- 保护整个存储区B：如选中，对整个存储区B进行pcrop保护。
- PCROPB\_strt：存储区B中受保护区的PCROP起始字段。
- 起始地址：由PCROPB\_strt字段定义的起始地址。
- PCROPB\_end：存储区B中受保护区的PCROP结束字段。
- 结束地址：由PCROPB\_end字段定义的结束地址。

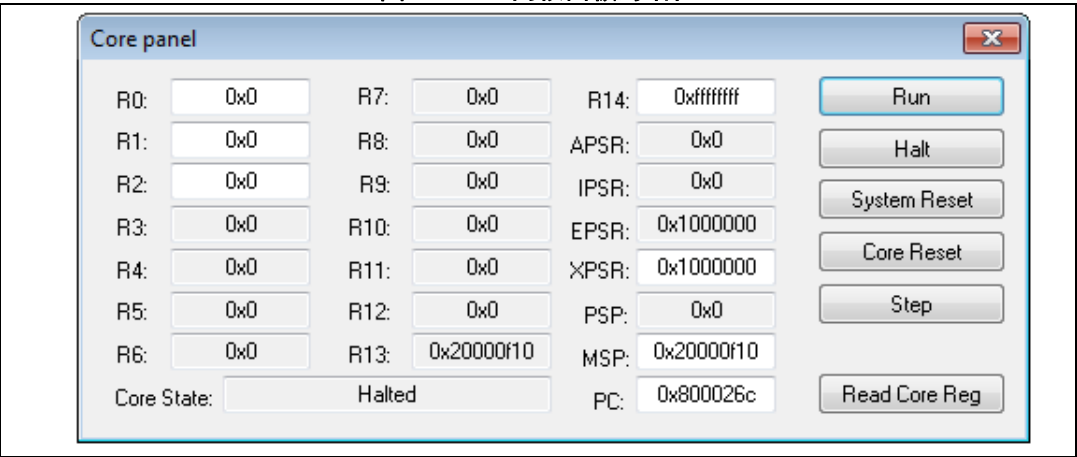
更多信息，请参阅 [www.st.com](http://www.st.com) 网站上Flash编程手册和参考手册中的选项字节部分。

### 3.7 MCU内核功能

图 21所示的**内核面板**对话框显示ARM®Cortex®-M3内核寄存器值。它还允许用户使用右侧的按钮对MCU执行以下操作：

- **运行**：运行内核。
- **停止**：停止内核。
- **系统复位**：发送系统复位请求。
- **内核复位**：复位内核。
- **步骤**：执行一步内核指令。
- **读取内核寄存器**：更新内核寄存器值。

图21. MCU内核面板对话框



注：PC和MSP寄存器均可以从该面板进行修改。

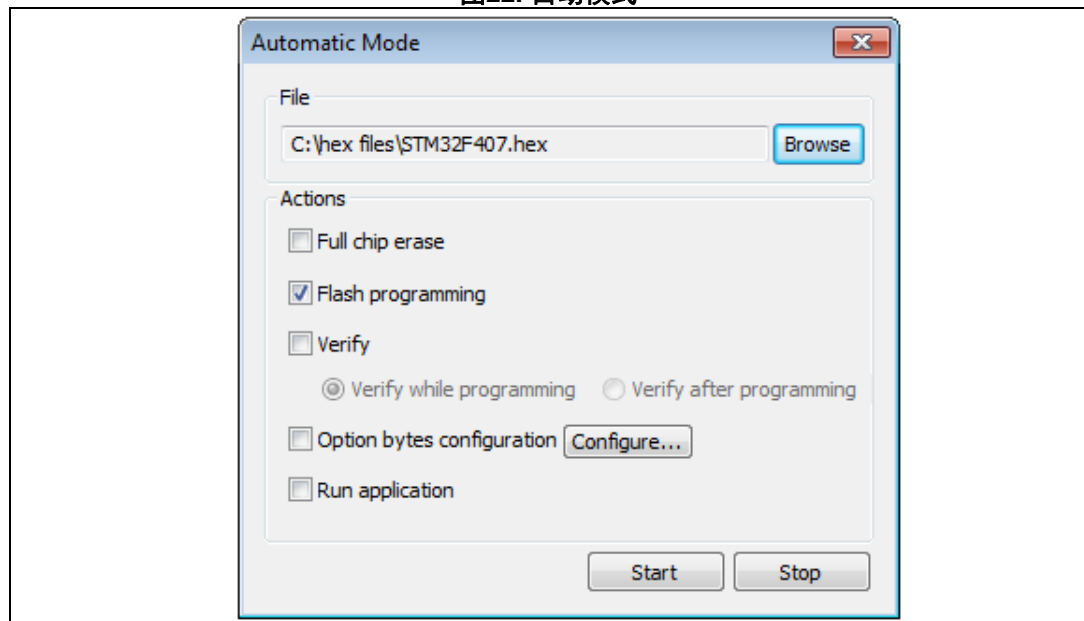
### 3.8 自动模式功能

图 22 所示的**自动模式**对话框允许用户对环路中的STM32器件进行编程和配置。它还允许用户对STM32器件执行以下操作：

- 全片擦除
- Flash编程
- 验证：
  - 在编程时验证
  - 在编程后验证
- 选项字节配置
- 运行应用

点击“开始”按钮对连接的STM32器件执行选中的操作，等到断开当前器件并连接新器件后重复相同操作。

图22. 自动模式



**注：**当STM32 Flash存储器受读出保护时，如果用户取消选中Flash编程操作，将自动取消保护。

当某些或全部STM32 Flash存储器受写保护时，如果用户取消选中Flash编程操作，将自动取消保护并在编程操作结束后恢复保护。

应建立与器件的连接，以便能够使用配置按钮选择选项字节配置。

连接的器件应衍生自同一STM32系列，并且必须全部以相同模式（JTAG或SWD）连接。

如果计算机连接了一个以上的ST-LINK探头，则不能使用自动模式。系统会显示一个对话框，阻止并要求用户只保留一个连接的ST-LINK探头以便继续使用该模式。

在开始自动模式之前，如果选项字节配置已选中，则必须使用“配置...”按钮配置选项字节。

在首次为特定器件ID配置选项字节时，将从连接的器件加载初始值。

如果已连接器件与配置选项字节时连接的器件的器件ID不相同，则必须在自动模式开始前使用“配置...”按钮配置选项字节。

## 3.9 为外部存储器开发自定义加载程序

基于 *ExternalLoader* 目录下的示例，用户可以为给定的外部存储器开发自定义加载程序。

这些示例适用于三种工具链：MDK-ARM™、EWARM和TrueSTUDIO®。自定义加载程序的开发可以使用上述三个工具链之一来执行，能够保持相同的编译器/链接器配置，如示例中所示。

要创建一个新的外部存储器加载程序，请按照以下步骤操作：

1. 使用外部存储器相关的正确信息，来更新 *Dev\_Inf.c* 文件的 *StorageInfo* 结构中的设备信息。
2. 在 *Loader\_Src.c* 文件中重写相应的函数代码。
3. 更改输出文件名。

注：一些函数是强制性的，不能省略（参见 *Loader\_Src.c* 文件中的函数说明）。

不应修改链接文件（linker files）或分散链接描述文件（scatter files）。

在构建外部加载程序项目之后，会生成一个ELF文件。ELF文件的扩展名取决于所用工具链（对于Keil为.axf，对于EWARM为.out，以及对于TrueSTUDIO或任何基于gcc的工具链为.elf）。

必须将ELF文件的扩展名更改为“.stldr”，且必须将该文件复制到 **ExternalLoader** 目录下。

### 3.9.1 Loader\_Src.c文件

基于特定IP为内存开发外部加载程序需要以下函数：

- **Init函数**  
Init函数定义将外部存储器连接到设备所用的GPIO引脚，并初始化所用IP的时钟。  
如果成功则返回1，失败则返回0。  
`int Init (void)`
- **Write函数**  
Write函数将一块RAM范围中的缓冲区数据写入到指定的地址上去。  
如果成功则返回1，失败则返回0。  
`int Write (uint32_t Address, uint32_t Size, uint8_t* buffer)`
- **SectorErase函数**  
SectorErase函数擦除指定扇区的存储器。  
如果成功则返回1，失败则返回0。

注：该函数在SRAM存储器中不能使用。

```
int SectorErase (uint32_t StartAddress, uint32_t EndAddress)
```

其中，“StartAddress” = 要擦除的第一个扇区的地址，  
“EndAddress” = 要擦除的最后一个扇区的地址。

必须在外部加载程序中定义上述函数。工具用其来擦除和编程外部存储器。

例如，如果用户从外部加载程序菜单中单击程序按钮，该工具将执行以下操作：

- 自动调用Init函数来初始化接口（QSPI、FMC……）和闪存
- 调用SectorErase()来擦除所需的闪存扇区
- 调用Write()函数来编程存储器。

除了这些函数，我们还可以定义以下函数：

- Read函数  
Read函数用来读取指定范围的存储器，并将读取的数据返回到RAM里的缓冲区中。  
如果成功则返回1，失败则返回0。

```
int Read (uint32_t Address, uint32_t Size, uint16_t* buffer)
```

其中“Address” = 读取操作起始地址，“Size”= 读取操作的大小，“buffer”= 指向读取后的数据的指针。

**注：**对于QSPI/OSPI（Quad-SPI/Octo-SPI）存储器，可以在Init函数中定义存储器映射模式；这种情况下，Read函数无用。

- Verify函数  
选择“verify while programming”模式时会调用Verify函数。该函数检查编程的存储器是否与RAM中定义的缓冲区保持一致。它返回一个uint64，定义如下：

```
checksum<<32 + AddressFirstError
```

其中“AddressFirstError”为第一次失配的地址，“checksum”所编程缓冲区的校验和值

```
uint64_t Verify (uint32_t FlashAddr, uint32_t RAMBufferAddr, uint32_t Size)
```

- MassErase函数  
MassErase函数擦除整个存储器。  
如果成功则返回1，失败则返回0。

```
int MassErase (void)
```

- 校验和函数

所有上述函数在成功操作的情况下返回1，在失败的情况下返回0。

### 3.9.2 Dev\_Inf.c文件

该文件定义了StorageInfo结构。该结构定义的信息类型示例如下所示：

```
#if defined (__ICCARM__)
__root struct StorageInfo const StorageInfo = {
#else
```



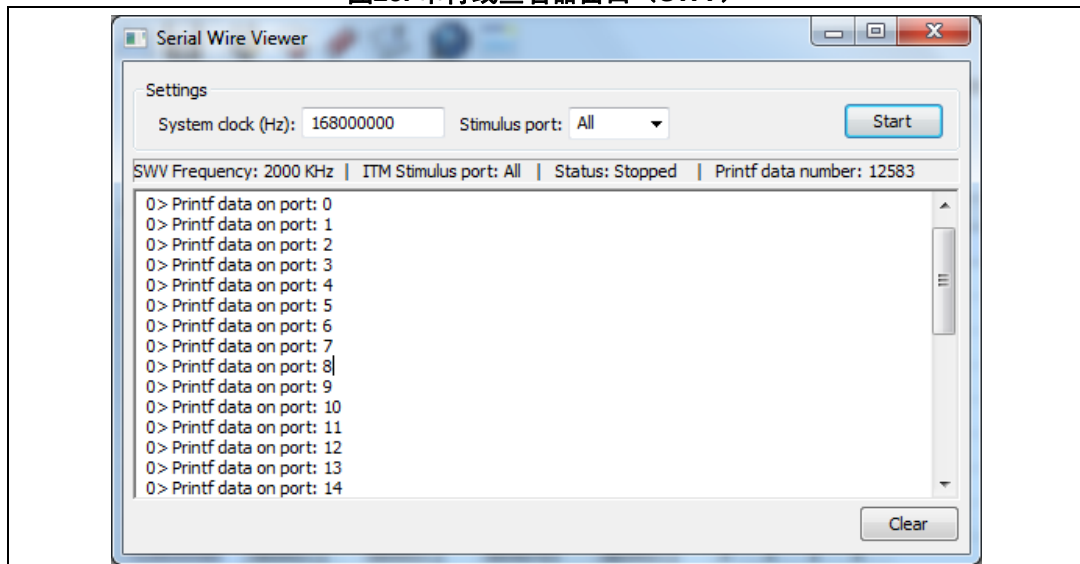
```
struct StorageInfo const StorageInfo = {
#ifdef
    "External_Loader_Name", // Device Name + version number
    MCU_FLASH, // Device Type
    0x08000000, // 器件起始地址
    0x00100000, // Device Size in Bytes (1MBytes/8Mbits)
    0x00004000, // Programming Page Size 16KBytes
    0xFF, // Initial Content of Erased Memory
// 指定扇区的大小和地址（查看下面的示例）
    0x00000004, 0x00004000, // Sector Num : 4 ,Sector Size: 16KBytes
    0x00000001, 0x00010000, // Sector Num : 1 ,Sector Size: 64KBytes
    0x00000007, 0x00020000, // Sector Num : 7 ,Sector Size: 128KBytes
    0x00000000, 0x00000000,
};
```

### 3.10 通过SWO查看器查看Printf

通过SWO查看器从目标板上的SWO脚位输出打印信息（Printf），通过Printf，可以在程序运行中显示一些有用的信息。

在开始接收SWO数据之前，为了让工具正确地配置ST-LINK，用户必须指定准确的目标系统时钟频率，以及正确的SWO频率的目标。“激励端口”组合框允许用户选择给定ITM刺激端口（从端口0至31）或从所有ITM激励端口同步接收数据。

图23. 串行线查看器窗口（SWV）



SWV信息栏显示关于当前SWV传输的有用信息，例如SWO频率（从系统时钟频率推导得出）和接收的数据量（以字节为单位）。

**注：** 由于ST-LINK硬件缓冲区的容量有限，传输期间可能会丢失一些SWV字节。

## 4 STM32 ST-LINK Utility命令行接口 (CLI)

### 4.1 命令行的使用

以下各节介绍如何由命令行来使用STM32 ST-LINK Utility。

ST-LINK Utility命令行接口位于以下地址：

`[Install_Directory]\STM32 ST-LINK utility\ST-LINK utility\ST-LINK_CLI.exe`

#### 4.1.1 连接和存储器操作命令

**说明：**选择JTAG或SWD通信协议。默认使用JTAG协议。

**语法：** `-c [ID=<id>/SN=<sn>] [JTAG/SWD] [FREQ=<frequency>] [UR/HOTPLUG] [LPM]`

[ID=<id>]: 当有多个探头连接到主机时，要使用的ST-LINK[0..9]的ID。

[SN=<sn>]: 选择的ST-LINK探头的序列号。

[UR]: 在复位状态下连接到目标。

[HOTPLUG]: 在不停机或复位的情况下连接到目标。

[FREQ=<frequency>]: JTAG或SWD协议的频率，以KHz为单位（将频率值提高到允许的频率值）。

SWD频率值：

4000KHz、1800KHz、900KHz、480KHz、240KHz、125KHz、100KHz、50KHz、25KHz、15KHz和5KHz。SWD协议的默认频率值为4000KHz。

JTAG频率值：

9000KHz、4500KHz、2250KHz、1125KHz、562KHz、281KHz、140KHz。

JTAG协议的默认频率值为9000KHz。

[LPM]: 在低功耗模式下激活调试

示例1: `-c ID=1 SWD UR LPM JTAG freq=1000`

示例2: `-c SN=55FF6C064882485358622187 SWD UR LPM`

**注：** 已弃用[SWCLK=<f>]和[JTAG=<f>]选项，改用[FREQ=<frequency>]选项。  
[SWCLK=<f>]: SWD协议的频率[0..10]

0 = 4.0 MHz（未指定时的默认值）

1 = 005 KHz

2 = 015 KHz

3 = 025 KHz

4 = 050 KHz

5 = 100 KHz

6 = 125 KHz

7 = 240 KHz

8 = 480 KHz

9 = 0.9 MHz

10 = 1.8 MHz

[JTAGCLK=<f>]: JTAG协议的频率[0..6]

0 = 9.0 MHz (未指定时的默认值)

1 = 140 KHz

2 = 281 KHz

3 = 562 KHz

4 = 1125 KHz

5 = 2250 KHz

6 = 4500 KHz

V2J24xx或更高ST-LINK/V2固件版本支持[JTAG=<f>]。

注: 如未指定[ID=<id>]和[SN=<sn>], 将选择ID=0的第一个ST-LINK。按ID或序列号选择ST-LINK的情况适用于:

- V1J13S0或更高ST-LINK固件版本
- V2J21S4或更高ST-LINK/V2固件版本
- V2J21M5或更高ST-LINK/V2-1固件版本

[UR]只对ST-LINK/V2的SWD模式可用。

当用户断开与目标的连接时, [LPM]模式禁用。

对于JTAG模式, 自ST-LINK固件版本V2J15Sx起提供“复位状态下连接”。

JTAG连接器(引脚15)的RESET引脚应连接到器件复位引脚。

[HOTPLUG]在SWD模式下可用。

对于JTAG模式, 自ST-LINK固件版本V2J15Sx起提供热插拔连接。

## -List

说明: 列出连接到计算机的每个ST-LINK探头的相应固件版本和唯一的序列号(SN)。

注: 为了获得正确的SN, ST-LINK固件版本应为:

- V1J13S0或更高ST-LINK固件版本。
- V2J21S4或更高ST-LINK/V2固件版本。
- V2J21M5或更高ST-LINK/V2-1固件版本。

当另一应用使用ST-LINK/V2或ST-LINK/V2-1探头时, 将不显示该探头的序列号且其不能在ST-LINK Utility的当前实例中使用。

## -r8

说明: 读取<NumBytes>存储器。

语法: -r8 <Address> <NumBytes>

示例: -r8 0x20000000 0x100

**-w8**

说明：向指定存储器地址写入8位数据。

语法：-w8 <Address> <data>

示例：-w8 0x20000000 0xAA

注：-w8支持对Flash存储器、OTP、SRAM和R/W寄存器的写入。

**-w32**

说明：向指定存储器地址写入32位数据。

语法：-w32 <Address> <data>

示例：-w32 0x08000000 0xAABBCCDD

注：-w32支持对Flash存储器、OTP、SRAM和R/W寄存器的写入。

#### 4.1.2 内核命令

**-Rst**

说明：复位系统。

语法：-Rst

**-HardRst**

说明：硬件复位。

语法：-HardRst

注：-HardRst命令仅对ST-LINK/V2可用。JTAG连接器（引脚15）的RESET引脚应连接到器件复位引脚。

**-Run**

说明：按照用户应用的定义设置程序计数器和堆栈指针，并执行运行操作。

语法：-Run [<Address>]

示例：-run 0x08003000

**-Halt**

说明：停止内核。

语法：-Halt

**-Step**

说明：执行步骤内核指令。

语法：-Step

**-SetBP**

说明：在特定地址设置软件或硬件断点。如未指定地址，则使用0x08000000。

语法：-SetBP [<Address>]

示例：-SetBP 0x08003000

**-ClrBP**

说明：清除所有硬件断点（如果有）。

语法：-ClrBP

**-CoreReg**

说明：读取内核寄存器。

语法：-CoreReg

**-SCore**

说明：检测内核状态。

语法：-SCore

### 4.1.3 Flash命令

**-ME**

说明：执行全片擦除操作。

语法：-ME

**-SE**

说明：擦除Flash扇区。

语法：-SE <Start\_Sector> [<End\_Sector>]

示例：-SE 0 => 擦除扇区0

-SE 2 12 => 擦除扇区2至扇区12

\* 对于STM32L系列，以下命令用于擦除数据EEPROM：

-SE ed1 => 擦除地址为0x08080000的数据EEPROM

-SE ed2 => 擦除地址为0x08081800的数据EEPROM

### -P

**说明：**将二进制、Intel十六进制或Motorola S-record文件加载到器件存储器中，不进行验证。对于十六进制和srec格式，地址十分重要。

**语法：**-P <File\_Path> [<Address>]

示例：-P C:\file.srec

-P C:\file.bin 0x08002000

-P C:\file.hex

**注：**根据STM32供电电压，STM32F2和STM32F4系列支持不同的编程模式。当使用ST-LINK/V2时，会自动检测供电电压。这样可以选择正确的编程模式。当使用ST-LINK时，默认选择32位编程模式。

如果器件受读保护，读保护会被禁用。如果Flash存储器页面受写保护，编程期间会禁用写保护，之后会再恢复。

### -V

**说明：**验证编程操作已成功执行。

**语法：**-V [while\_programming/after\_programming]

示例：-P \*C:\file.srec\* -V "after\_programming"

**注：**如果没有提供参数，将执行编程时验证法。

## 4.1.4 其他命令

### -CmpFile

**说明：**将二进制、Intel十六进制或Motorola S-record文件与器件存储器内容进行比较，并显示第一个不同值的地址。

**语法：**-CmpFile <File\_Path> [<Address>]

示例1：-CmpFile "c:\\application.bin" 0x08000000

示例2：-CmpFile "c:\\application.hex"

用户还可以将文件内容与外部存储器进行比较。应通过-EL命令指定外部存储器加载程序的路径。

示例1：-CmpFile "c:\\application.bin" 0x64000000 -EL "c:\\Custom-Flash-Loader.stldr"

### -Cksum

**说明：**计算给定文件或指定存储区的校验和值。使用的算法是简单的按位求和算法。结果截断为32位字。

**语法：**-Cksum <File\_Path>

**-Cksum** <Address> <Size>

示例1: -Cksum "C:\File.hex"

示例2: -Cksum 0x08000000 0x200

示例3: -Cksum 0x90000000 0x200 -EL "C:\Custom\_Flash\_Loader.stldr"

### **-Dump**

**说明:** 读取目标存储器内容并保存到文件中。

**语法:** -Dump<Address> <Memory\_Size> <File\_Path>

### **-Log**

**说明:** 使能跟踪日志文件生成。

生成的日志文件位于%userprofile%\STMicroelectronics\ST-LINK utility目录下。

### **-NoPrompt**

**说明:** 禁用用户确认提示（例如，在文件中设定RDP级别2）。

### **-Q**

**说明:** 使能静默模式。不显示进度条。

### **-TVolt**

**说明:** 显示目标电压。

## **4.1.5 选项字节命令**

### **-rOB**

**说明:** 显示所有选项字节。

**语法:** -rOB

### **-OB**

**说明:** 配置选项字节。该命令:

- 将读保护级别设为级别0（无保护）
- 将IWDG\_SW选项设为“1”（通过软件使能看门狗）
- 将nRST\_STOP选项设为“0”（在进入待机模式时生成复位）
- 设置Data0选项字节
- 设置Data1选项字节

**语法:** -OB [RDP=<Level>][BOR\_LEV=<Level>][IWDG\_SW=<Value>]



```
[nRST_STOP=<Value>][nRST_STDBY=<Value>][nBFB2=<Value>]  
[nBoot1=<Value>][nSRAM_Parity=<Value>][Data0=<Value>]  
[SPRMOD=<Value>][Data1=<Value>][WRP=<Value>][WRP2=<Value>]  
[WRP3=<Value>][WRP4=<Value>]  
[BOOT_ADD0=<Value>]  
[BOOT_ADD1=<Value>]
```

示例: -OB RDP=0 IWDG\_SW=1 nRST\_STOP=0 Data0=0xAA Data1=0xBC

### 选项字节命令参数说明

#### RDP=<Level>:

RDP=<Level>设置Flash存储器读保护级别。

<Level>可以是以下级别之一:

- 0: 保护禁用
- 1: 保护使能
- 2: 保护使能 (调试和从SRAM自举功能禁用)

注: 级别2仅对STM32F0、STM32F2、STM32F3、STM32F4和STM32L1系列可用。

#### BOR\_LEV=<Level>:

BOR\_LEV设置欠压复位阈值电压。

对于STM32L4系列:

- 0: 复位电压阈值为约1.7 V
- 1: 复位电压阈值为约2.0 V
- 2: 复位电压阈值为约2.2 V
- 3: 复位电压阈值为约2.5 V
- 4: 复位电压阈值为约2.8 V

对于STM32L1系列:

- 0: BOR关闭, 电压范围1.45至1.55 V
- 1: 电压范围1.69至1.8 V
- 2: 电压范围1.94至2.1 V
- 3: 电压范围2.3至2.49 V
- 4: 电压范围2.54至2.74 V
- 5: 电压范围2.77至3.0 V

对于STM32F2和STM32F4系列:

- 0: BOR关闭, 电压范围1.8至2.10 V
- 1: 电压范围2.10至2.40 V
- 2: 电压范围2.40至2.70 V
- 3: 电压范围2.70至3.60 V

**IWDG\_SW=<Value>:**

<Value>应为0或1:

- 0: 硬件独立看门狗
- 1: 软件独立看门狗

**nRST\_STOP=<Value>:**

<Value>应为0或1:

- 0: 在CPU进入停止模式时生成复位
- 1: 不产生复位

**nRST\_STDBY=<Value>:**

<Value>应为0或1:

- 0: 在CPU进入待机模式时生成复位
- 1: 不产生复位

**PCROP\_RDP=<Value>:**

<Value>应为0或1:

- 0: 当RDP级别从级别1降至级别0时，不擦除PCROP区。
- 1: 当RDP级别从级别1降至级别0时，擦除PCROP区（完全批量擦除）。

**PCROPA\_STRT=<Value>:**

设置存储区A的PCROP起始区  
请参阅相应器件的参考手册<Value>

**PCROPA\_END**

设置存储区A的PCROP结束区  
请参阅相应器件的参考手册<Value>

**PCROPB\_STRT**

设置存储区B的PCROP起始区  
请参阅相应器件的参考手册<Value>

**PCROPB\_END**

设置存储区B的PCROP结束区  
请参阅相应器件的参考手册<Value>

**VDDA=<Value>:**

<Value>应为0或1:

- 0: VDDA电源监控器禁用
- 1: VDDA电源监控器使能

**DUALBANK=<Value>:**

<Value>应为0或1:

0: 单存储区Flash。

1: 双存储区Flash。

**BFB2=<Value>:**

双存储区自举

<Value>应为0或1:

0: 双存储区自举禁用。

1: 双存储区自举使能。

**SRAM2\_RST=<Value>:**

在系统复位时擦除SRAM2

<Value>应为0或1:

0: 在发生系统复位时擦除SRAM2。

1: 在发生系统复位时不擦除SRAM2。

**SRAM2\_PE=<Value>:**

SRAM2奇偶校验使能

<Value>应为0或1:

0: SRAM2奇偶校验使能。

1: SRAM2奇偶校验禁用。

**nBFB2=<Value>:**

<Value>应为0或1:

0: 当自举引脚置位为从用户Flash位置自举时, 从Flash存储区2自举 (默认)

;

1: 当自举引脚置位为从用户Flash位置自举时, 从Flash存储区1自举 (默认)。

*注: nBFB2只对包含两个Flash存储区的器件可用。*

**nBoot0\_SW\_Cfg=<Value>:**

仅STM32F04x

<Value>应为0或1:

0: 它允许用户完全禁用BOOT0硬件引脚并使用用户选项位11 (nBoot0)。

1: BOOT0引脚将绑定至GPIO引脚 (对于LQFP32和更小封装为PB8, 对于QFN32和更大封装为PF11)。

**nBoot0=<Value>:**

仅适用于STM32F04x且仅当nBoot0\_SW\_Cfg置位时。

<Value>应为0或1:

选择自举模式连同nBoot1 (参见下面的表 1)。

**nBoot1=<Value>:**

仅适用于STM32F0和STM32F3系列

<Value>应为0或1:

表1. STM32F04x的nBoot1配置

nBoot1	nBoot0	BOOT0引脚	nBoot0_SW_Cfg	Flash存储器为空	自举模式
X	X	0	1	否	主 Flash
X	X	0	1	是	系统存储区
0	X	1	1	X	内部 SRAM
1	X	1	1	X	系统存储区
X	1	X	0	X	主 Flash
0	0	X	0	X	内部 SRAM
1	0	X	0	X	系统存储区

表2. STM32F0和STM32F3的nBoot1配置

nBoot1	BOOT0	自举模式
X	0	主 Flash
0	1	内部 SRAM
1	1	系统存储区

**nSRAM\_Parity=<Value>:**

此位允许用户使能SRAM硬件奇偶校验。

<Value>应为0或1。

注: nSRAM\_Parity只对STM32F0和STM32F3系列可用。

**SDADC12\_VDD=<Value>:**

选择对SDADC12\_VDD电源的模拟监控（与Bgap 1.2V电压进行比较）。

<Value>应为0或1。

注: SDADC12\_VDD只对STM32F37x器件可用。

**Data0=<Value>:**

Data0设置Data0选项字节。

<Value>应为[0..0xFF]。

注: 对STM32F0、STM32F2、STM32F3、STM32F4和STM32L1器件不可用。



**Data1=<Value>:**

Data1 设置 Data1 选项字节。

<Value> 应为 [0..0xFF]。

*注:* 对 STM32F0、STM32F2、STM32F3、STM32F4 和 STM32L1 系列不可用。

**SPRMOD = <Value>:**

选择 WPRi 位的保护模式:

<Value> 应为 0 或 1。

0: WPRi 位用于扇区 i 的写保护 (默认)。

1: WPRi 位用于扇区 i (扇区) 的 PCRoP 保护 (读保护)。

*注:* 仅对支持 PCRoP 功能的器件可用。

**WPRMOD = <Value>:**

选择 WPRi 位的保护模式:

<Value> 应为 0 或 1。

0: WPRi 位用于扇区 i 的写保护 (默认)。

1: WPRi 位用于扇区 i (扇区) 的 PCRoP 保护 (读保护)。

*注:* 仅对支持 PCRoP 功能的 STM32L0 器件可用。

**WRP = <Value>:**

WRP 使能/禁用 MCU Flash 扇区的写保护。

根据连接的器件, 每一位使能/禁用一个或更多扇区的写保护。

对于 STM32L1 器件, WRP[i] = 0 表示 Flash 扇区受保护。

对于其他器件, WRP[i] = 1 表示 Flash 扇区受保护。

该命令足以用来使能/禁用所有 Flash 扇区保护, 但 STM32L1 中等以上容量和高容量器件除外, 这些器件可能必须使用 WRP2、WRP3 和 WRP4 命令。

对于 STM32F4 系列, WRP 的每一位使能/禁用一个扇区的写保护。

<Value> 应为 [0..0xFFFFFFFF]

*注:* 对于支持 PCRoP 功能的器件, 当 SPRMOD = 1 时, WRP 控制 MCU Flash 扇区的读保护。

**WRP2 = <Value>:**

WRP2 只可由 STM32L1 中等以上容量、高容量和高以上容量器件用来使能/禁用从页面 512 至 1023 的 Flash 扇区的保护。

<Value> 应为 [0..0xFFFFFFFF]。

注： 对于支持PCROP功能的器件，当SPRMOD = 1时，WRP控制MCU Flash扇区的读保护。

**WRP3=<Value>:**

WRP3只可由STM32L1高容量和超高容量器件用来使能/禁用从页面1024至1535的Flash扇区的保护。

<Value>应为[0..0xFFFFFFFF]

**WRP4=<Value>:**

WRP4只可由STM32L1超高容量器件用来使能/禁用从扇区1536至2047的Flash扇区的保护。

<Value>应为[0..0xFFFFFFFF]

**BOOT\_ADD0=<Value>:**

当BOOT0 = 0时，自举地址使能。

<Value>应为[0..0xFFFF]。

BOOT\_ADD0[15:0]对应地址[29:14]

**BOOT\_ADD1=<Value>:**

当BOOT0 = 1时，自举地址使能。

<Value>应为[0..0xFFFF]。

BOOT\_ADD1[15:0]对应地址[29:14]

**DB1M =<Value>:**

1-MB Flash存储器上的双存储区：

<Value>应为0或1。

注： 1 可在支持双区交换的STM32F42x/STM32F43x 1-Mb器件上使用。

注： 2 上述所有参数应采用十六进制格式。  
关于详细信息，请参阅[www.st.com](http://www.st.com)上相应器件的Flash编程手册中的选项字节部分。

## 4.1.6 外部存储器命令25

**-EL**

说明：为外部存储器操作选择自定义Flash存储器加载程序。

语法：-EL [<loader\_File\_Path>]

示例：-P c:\application.hex -EL c:\Custom-Flash-Loader.stldr

4.1.7 ST-LINK\_CLI返回代码

在执行ST-LINK\_CLI命令时，如发生错误，返回代码（Errorlevel）将大于0。  
下面的表 3汇总了ST-LINK\_CLI返回代码：

表3. ST-LINK\_CLI返回代码

返回代码	指令	误差
1	全部	命令参数错误。
2	全部	连接问题。
3	全部	命令对连接的目标不可用。
4	-w8, -w32	向指定存储器地址写入数据时发生错误。
5	-r8, r32	无法从指定存储器地址读存储器。
6	-rst, -HardRst	无法复位MCU。
7	-Run	未能运行应用。
8	-halt	未能停止内核。
9	-STEP	未能执行单指令步骤。
10	-SetBP	未能设置/清除断点。
11	-ME, -SE	无法擦除一个或多个Flash扇区。
12	-P, -V	Flash编程/验证错误。
13	-OB	选项字节编程错误。
14	-w8, w32, -r32, -P, -V, -ME, -SE	存储器加载程序故障（内部Flash或外部存储器）

## 5 STM32 ST-LINK Utility外部加载程序开发

在外部加载程序项目中，有两个基本文件：Loader\_Src.c和Dev\_Inf.c。

### 5.1 Loader\_Src.c文件

基于特定IP为内存开发外部加载程序需要下述函数。请注意，必须在外部加载程序中定义以下函数。

- **Init函数**  
Init函数定义用于连接外部存储器的GPIO，初始化所用IP的时钟，并定义使用的GPIO。

```
int Init (void)
```

- **Write函数**  
Write函数对使用RAM范围内的地址定义的缓冲区进行编程。

```
int Write (uint32_t Address, uint32_t Size, uint8_t* buffer)
```

- **SectorErase函数（Flash存储器）**  
SectorErase函数擦除由起始地址和结束地址定义的存储器扇区。

**注：** 该函数在SRAM存储器中不能使用。

```
int SectorErase (uint32_t StartAddress, uint32_t EndAddress)
```

其中，“StartAddress” = 要擦除的第一个扇区的地址，  
“EndAddress” = 要擦除的最后一个扇区的地址。

下面是可以定义的其他函数：

- **Read函数**  
该函数用来读取指定范围的存储器，并将读取的数据返回到RAM里的缓冲区中

```
int Read (uint32_t Address, uint32_t Size, uint16_t* buffer)
```

其中，“Address” = 读取操作的起始地址，“Size” = 读取操作的大小，“buffer” = 指向读取后的数据的指针。

**注：** 对于QSPI/OSPI（Quad-SPI/Octo-SPI）存储器，可以在Init函数中定义存储器映射模式；这种情况下，Read函数无用。

- **Verify函数**  
选择“verify while programming”模式时会调用该函数。该函数检查编程的存储器是否与RAM中定义的缓冲区保持一致。它返回一个uint64，定义如下：

```
checksum<<32 + AddressFirstError
```

其中“AddressFirstError”为第一次失配的地址，“Checksum”为编程缓冲区的校验和值。

```
uint64_t Verify (uint32_t FlashAddr, uint32_t RAMBufferAddr, uint32_t Size)
```

- **MassErase函数**  
该函数擦除整个存储器。



int MassErase (void)

- 校验和函数  
Checksum函数计算已编程的存储器校验和。使用的算法是简单的按位求和算法。结果截断为32位字。使用在ST-LINK Utility中打开的文件计算校验和值，是一种更快的验证编程操作的方法。  
如果成功则返回1，失败则返回0。

## 5.2 Dev\_Inf.c文件

该文件定义了StorageInfo结构。该结构定义的信息类型示例如下所示：

```
#if defined (__ICCARM__)
__root struct StorageInfo const StorageInfo = {
#else
struct StorageInfo const StorageInfo = {
#endif
    "External_Loader_Name", // Device Name + version number
    MCU_FLASH, // Device Type
    0x08000000, // Device Start Address
    0x00100000, // Device Size in Bytes (1MBytes/8Mbits)
    0x00004000, // Programming Page Size 16KBytes
    0xFF, // Initial Content of Erased Memory
// 指定扇区的大小和地址（查看下面的示例）
    0x00000004, 0x00004000, // Sector Num : 4 ,Sector Size: 16KBytes
    0x00000001, 0x00010000, // Sector Num : 1 ,Sector Size: 64KBytes
    0x00000007, 0x00020000, // Sector Num : 7 ,Sector Size: 128KBytes
    0x00000000, 0x00000000,
```

6 版本历史

表4. 文档版本历史

日期	版本	变更
2010年1月22日	1	初始版本。
2010年2月12日	2	修改了图1、2、3、4、5、6和7。增加了SWD支持信息。
2010年5月20日	3	在 <a href="#">第 2.2.4节</a> 和 <a href="#">第 3.6节</a> 中增加了XL容量器件的支持信息。
2010年8月27日	4	增加了STM32L1的支持信息。
2011年2月3日	5	十六进制、srec格式支持。 命令行接口支持。 修改了名称和所有图形。 增加了 <a href="#">第 3.8节：自动模式功能</a>
2011年8月1日	6	在 <a href="#">第 1.2节：硬件要求</a> 中增加了ST-LINK/V2的支持信息， 并在 <a href="#">第 3.5节：器件编程</a> 和 <a href="#">第 4.1.3节：Flash命令</a> 中增加了STM32的不同编程模式的支持信息。 在 <a href="#">第 3.1节：器件信息</a> 中增加了MCU版本ID显示。
2011年10月18日	7	在整个文档中增加了STM32W和STM32F4的支持信息。 在 <a href="#">第 2.2.4节：目标菜单</a> 中增加了“复位状态下连接”选项的支持信息。 替换了 <a href="#">图 21：MCU内核面板对话框</a> 。 更新了 <a href="#">第 3.3节：存储器显示和修改</a> 中通过GUI命令修改Flash和 <a href="#">第 4.1.1节：连接和存储器操作命令</a> 中通过CLI命令修改Flash的相关信息。 在 <a href="#">第 4.1.2节：内核命令</a> 中增加了CLI模式下的 <a href="#">-HardRst</a> 命令。 在 <a href="#">选项字节命令参数说明</a> 中增加了用于STM32L1高容量器件的WRP2和WRP3。
2012年5月11日	8	增加了STM32F0的支持信息。 增加了 <a href="#">表1：适用工具</a> 。 替换了 <a href="#">图 1</a> 、 <a href="#">图 15</a> 、 <a href="#">图 18</a> 、 <a href="#">图 19</a> 和 <a href="#">图 22</a> 。 在 <a href="#">第 2.2.4节：目标菜单</a> 中增加了关于JTAG模式的注释。 在 <a href="#">选项字节命令参数说明</a> 中增加了nBoot1、VDDA和nSRAM_Parity命令。 更新了 <a href="#">第 2.2.4节：目标菜单</a> 中的“复位状态下连接”选项。 更新了 <a href="#">第 4.1.1节：连接和存储器操作命令</a> 中的-c命令并增加了-Q命令。 增加了 <a href="#">第 4.1.7节：ST-LINK_CLI返回代码</a> 。



表4. 文档版本历史（续）

日期	版本	变更
2012 年 10 月 05 日	9	<p>增加了STM32F050、STM32F3和STM32L中等以上容量器件的支持信息。</p> <p>在 <a href="#">第 2.2.1 节：文件菜单</a> 中增加了“比较两个文件”并替换了 <a href="#">图 3</a>。</p> <p>替换了 <a href="#">图 6</a>。</p> <p>在 <a href="#">第 3.5 节：器件编程</a> 中增加了“编程后复位选项”，并将 <a href="#">图 17</a> 替换为 <a href="#">图 13</a>。</p> <p>在 <a href="#">第 4.1.5 节：选项字节命令</a> 中增加了“SPRMOD”命令。</p> <p>更新了 <a href="#">第 3.6 节：选项字节配置</a> 中的Flash扇区保护，并替换了 <a href="#">图 19</a>。</p> <p>替换了 <a href="#">第 3.7 节：MCU内核功能</a> 中的 <a href="#">图 21</a>。</p> <p>更新了 <a href="#">第 4.1.5 节：选项字节命令</a> 中的“WRPx”命令。</p> <p>纠正了打印错误。</p>
2013年1月11 日	10	<p>更新了 <a href="#">第 1.1 节：系统要求</a></p> <p>更新了 <a href="#">第 2.1 节：主窗口</a>，包括 <a href="#">图 1：STM32 ST-LINK Utility 用户界面主窗口</a>（实时更新复选框和编辑菜单）</p> <p>更新了 <a href="#">图 3</a></p> <p>增加了 <a href="#">第 2.2.2 节：编辑菜单</a>，包括 <a href="#">图 4：编辑菜单</a></p> <p>更新了 <a href="#">图 5</a></p> <p>更新了 <a href="#">第 2.2.4 节：目标菜单</a>，包括 <a href="#">图 6</a></p> <p>更新了 <a href="#">第 2.2.5 节：ST-LINK 菜单</a>，包括 <a href="#">图 7</a></p> <p>更新了 <a href="#">图 12</a></p> <p>更新了 <a href="#">第 3.3 节：存储器显示和修改</a>，包括 <a href="#">图 15</a></p> <p>更新了 <a href="#">第 4.1 节：命令行的使用</a>（热插拔）</p> <p>在“-OB”命令的语法中增加了“[SPRMOD=&lt;Value&gt;]”，参见 <a href="#">第 4.1.5 节</a></p> <p>通篇用“nBFB2”替换了“BFB2”</p>
2013年4月30 日	11	<p>更新了 <a href="#">图 1：STM32 ST-LINK Utility 用户界面主窗口</a> 和 <a href="#">图 3：文件菜单</a>。</p> <p>在 <a href="#">第 2.2.3 节：查看菜单</a> 中增加了外部存储器。</p> <p>在 <a href="#">第 2.2.5 节：ST-LINK 菜单</a> 中增加了printf数据。</p> <p>增加了 <a href="#">第 2.2.6 节：外部加载程序菜单</a>。</p> <p>更新了 <a href="#">图 12：Help 菜单</a>、<a href="#">图 15：STM32 ST-LINK Utility 用户界面</a>、<a href="#">图 17：打开文件对话框</a>、<a href="#">图 19：选项字节对话框</a>、<a href="#">图 21：MCU内核面板对话框</a> 和 <a href="#">图 22：自动模式</a>。</p> <p>增加了 <a href="#">第 3.9 节：为外部存储器开发自定义加载程序</a> 和 <a href="#">第 3.10 节：通过SWO查看器查看Printf</a>。</p> <p>在 <a href="#">第 4.1.3 节：Flash 命令</a> 中增加了 CmpFile。</p> <p>增加了 <a href="#">第 4.1.6 节：外部存储器命令25</a>。</p> <p>更新了尾页的免责声明。</p>
2013 年 7 月 10 日	12	<p>更新了 <a href="#">第 2.2.4 节：目标菜单</a> 中关于“设置”的说明。</p> <p>删除了“适用工具”表，并更新了封面上的产品编号。</p>
2013年11月 04 日	13	<p>更新了 <a href="#">图 9：外部加载程序窗口</a>、<a href="#">图 18：器件编程对话框（编程）</a> 和 <a href="#">图 22：自动模式</a>。</p> <p>更新了 <a href="#">第 3.5 节：器件编程</a> 中的第(4)点并增加了第(5)点，更新了 <a href="#">第 3.8 节：自动模式功能</a> 中的第(3)点。</p> <p>更新了以下命令：-V、WRP=&lt;Value&gt;、WRP2=&lt;Value&gt; 和 WRP3=&lt;Value&gt;，并增加了 WRP4=&lt;Value&gt;。</p>

表4. 文档版本历史（续）

日期	版本	变更
2013年12月16日	14	更新了第 1.3节：安装STM32 ST-LINK Utility说明。
2014年2月13日	15	更新了Chapter 4.1.5: 选项字节命令中的语法OB，增加了nSRAM_Parity=<Value>: 和nBoot0_SW_Cfg=<Value>:，更新了nBoot1=<Value>:，更新了WRP=<Value>:，增加了WRP4=<Value>:
2014年5月16日	16	增加了ST32L0系列的支持信息。 更新了第 2.2.4节：目标菜单、第 3.8节：自动模式功能、第 4.1.1节：连接和存储器操作命令和第 2.2.5节：ST-LINK菜单。
2014年10月13日	17	更新了第 3.6节：选项字节配置中的用户配置选项字节并添加了自举地址选项字节。 更新了第 4.1.5节：选项字节命令。 更新了图 19：选项字节对话框。
2015年2月11日	18	增加了第 3.2节：设置和第 4.1.4节：其他命令。 更新了第 2.2节：菜单栏、第 3.5节：器件编程、第 4.1.1节：连接和存储器操作命令和第 4.1.3节：Flash命令。 更新图片1、3、4、5、6、7、8、9、10、11、12、13、15、16、17、18、21和22。 增加了图 2：菜单栏和图 14：“设置”对话框。
2015年7月23日	19	更新了第 1.2节：硬件要求。 更新了第 3.6节：选项字节配置： - 读出保护和BOR级别分段。 - 更新了图 19：选项字节对话框。 - 增加了图 20：读/写保护模式。 - 增加了STM32L4器件的相关段落。 更新了第 3.8节：自动模式功能，完善了图 21：MCU内核面板对话框下方的注释。 更新了第 4.1.5节：选项字节命令中关于STM32L4系列的内容。
2015年11月10日	20	更新了第 2.2.4节：目标菜单、第 3.2节：设置、第 4.1.4节：其他命令。
2016年4月12日	21	更新了第 2.2.4节：目标菜单，第 3.2节：设置，第 3.5节：器件编程，图 14：“设置”对话框，图 18：器件编程对话框（编程），图 20：读/写保护模式，第 4.1.1节：连接和存储器操作命令。

表4. 文档版本历史（续）

日期	版本	变更
2016年8月9日	22	更新了第 1.1 节：系统要求和第 4.1.2 节：内核命令。
2017年7月13日	23	更新了 <ul style="list-style-type: none"> <li>– 第 2.2.4 节：目标菜单中的存储器校验和</li> <li>– 第 3.2 节：设置</li> <li>– 第 3.5 节：器件编程</li> <li>– 第 3.9 节：为外部存储器开发自定义加载程序</li> <li>– 第 3.9.1 节：Loader_Src.c 文件</li> <li>– 第 3.9.2 节：Dev_Inf.c 文件</li> <li>– Section : -Cksum on page 39</li> <li>– Table 3: ST-LINK_CLI 返回代码</li> </ul> 增加了第 5 节：STM32 ST-LINK Utility 外部加载程序开发

表5. 中文文档版本历史

日期	版本	变更
2019年1月3日	1	中文初始版本。

**重要通知 - 请仔细阅读**

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2019 STMicroelectronics - 保留所有权利