



HT32 系列在系统编程 / 在应用编程用户手册

版本 :V1.00 日期 :2011-05-30

www.holtek.com

目录

1 简介 5

 关于该文档 5

 HT32 Flash 编程器简介 5

 ISP/IAP 简介 6

2 使用 HT32 Flash 编程器 7

 安装 HT32 Flash 编程器 7

 HT32 Flash 编程器快速启动 11

 功能介绍 14

 连接 14

 图像选择 15

 保护设定 15

 Flash 操作 16

 处理状态和结果信息 17

 自动连接和自动编程 18

 器件状态 19

 读 Flash 存储器 20

 工程操作 21

 为 HT32 系列单片机开发 Flash 编程器 22

 简介 22

 动态数据链接库 API 描述 24

 入门实例 31

3 使用 ISP 和 IAP 固件 33

 HT32 Flash 命令协议 33

 协议描述 33

 命令格式 34

 命令描述 35

 使用 ISP 启动加载器 39

 IAP 范例程序代码 40

 简介 40

 IAP 文本模式范例程序代码 43

 IAP UI 模式范例程序代码 46

 更新 IAP 固件 48

图列表

图 1. ISP/IAP 编程工具系统架构	5
图 2. ISP/IAP 的存储配置	6
图 3. HT32 Flash 编程器安装 – 欢迎页面	7
图 4. HT32 Flash 编程器安装 – 选择安装路径	8
图 5. HT32 Flash 编程器安装 – 准备安装页面	9
图 6. HT32 Flash 编程器安装 – 完成	10
图 7. 启动 HT32 Flash 编程器	11
图 8. HT32 Flash 编程器主屏	11
图 9. 连接状态	12
图 10. 运行成功屏	12
图 11. 读 Flash 存储器	13
图 12. 连接和相关设定	14
图 13. 图像选择	15
图 14. 保护设定	15
图 15. Flash 操作相关设定	16
图 16. 处理状态和结果	17
图 17. 自动连接和自动编程	18
图 18. 芯片状态	19
图 19. 读 Flash 存储器	20
图 20. 工程文件操作	21
图 21. VC++6.0 ISPCmd.dll 程序代码	31
图 22. 配置 COM 口和波特率	31
图 23. Holtek Flash 编程器协议	33
图 24. IAP 范例程序代码架构	40
图 25. IAP 模式启动流程	41
图 26. IAP 范例程序代码文件夹	42
图 27. 应用文本菜单	45
图 28. IAP 文本模式文本菜单	45
图 29. RAM 执行 IAP 文本模式文本菜单	48

表格列表

表 1. 处理状态和结果信息	17
表 3. API 的返回代码	23
表 4. ISP_SetupSerialPortConnection 函数	24
表 5. ISP_CloseSerialPortConnect 函数	24
表 6. ISP_ErasePage 函数	25
表 7. ISP_EraseMass 函数	25
表 8. ISP_BlankCheck 函数	26
表 9. ISP_WriteProgramB 函数	26
表 10. ISP_CRCCheck 函数	27
表 11. ISP_ReadData 函数	27
表 12. ISP_Execute 函数	28
表 13. ISP_Exit 函数	28
表 14. ISP_Reset 函数	28
表 15. ISP_IsConnectedToDevice 函数	29
表 16. ISP_GetTransProgress 函数	29
表 17. ISP_GetInformation 函数	29
表 18. ISP_GetInformation 函数的 nCode 定义	30
表 19. 命令描述	34
表 20. 获取信息命令 (i)	35
表 21. 获取信息命令的类型定义	35
表 22. 整体 / 页擦除命令 (e)	36
表 23. 空白检查命令 (b)	36
表 24. 编程 / 验证命令 (w)	37
表 25. CRC 检查 (c)	37
表 26. 执行, 退出和复位命令 (g)	38
表 27. 读存储器命令 (r)	38
表 28. 文本模式源文件	43
表 29. IAP UI 模式源文件	46

1 简介

关于该文档

该文档描述了如何使用 HT32 系列单片机的在系统编程 ISP 和在应用编程 IAP 工具。主要有两部分内容，基于 PC 的 HT32 Flash 编程器用户界面和 ISP/IAP 固件。在以下各章节对 HT32 Flash 编程器用户界面和 ISP/IAP 固件的简单介绍有助于对其架构的初步了解。该章节将描述如何使用 Flash 编程器和相关的开发信息包括命令协议，ISP 启动加载器和 IAP 范例程序代码。

HT32 Flash 编程器简介

编程器的主要任务是根据用户的应用进行代码 / 数据编程和 Flash 保护设置。另外，目标器件的现状比如器件名，Flash 大小，代码 / 数据信息和保护设置等可以进行查询并显示在图形用户界面。Flash 编程器通过 UART 接口与 ISP/IAP 固件进行通信，然后利用预定义的协议 / 命令提供 Flash 操作。用户可通过 GUI 选择 Flash 图像，调整 Flash 的保护设置，分配合适的 Flash 操作。经过几个简单的步骤，图像就可以自动导入目标器件。

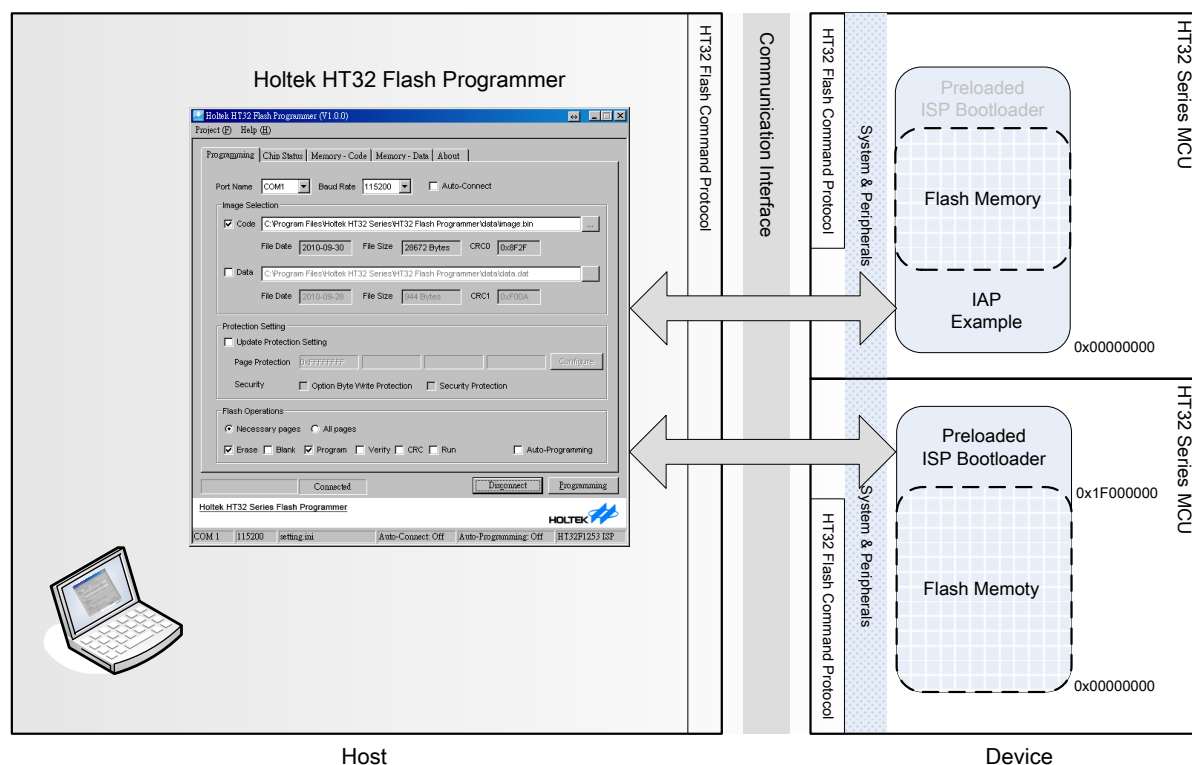


图 1. ISP/IAP 编程工具系统架构

ISP/IAP 简介

对于具有有限通信接口和嵌入 Flash 系统的最终产品来说，应用固件的更新能力是一个重要的考虑因素。因此需要软件控制的编程方案通过预定义的通信接口来擦除和编程 Flash 存储器。在 ISP 提供了可与根据 Holtek HT32 Flash 命令协议更新 Flash 存储器的 PC 软件进行通信的预装载 ISP 启动加载器。

在 IAP 中也有同样的功能，除此之外，IAP 还可以在应用程序运行期间实现编程过程。IAP 程序，像正常的应用程序一样，需利用其他方法比如 ISP，USB 调试器或者 Flash 烧录编入 Flash 存储器。如下图虚线部分所示就是可通过 ISP/IAP 更新的 Flash 存储器。

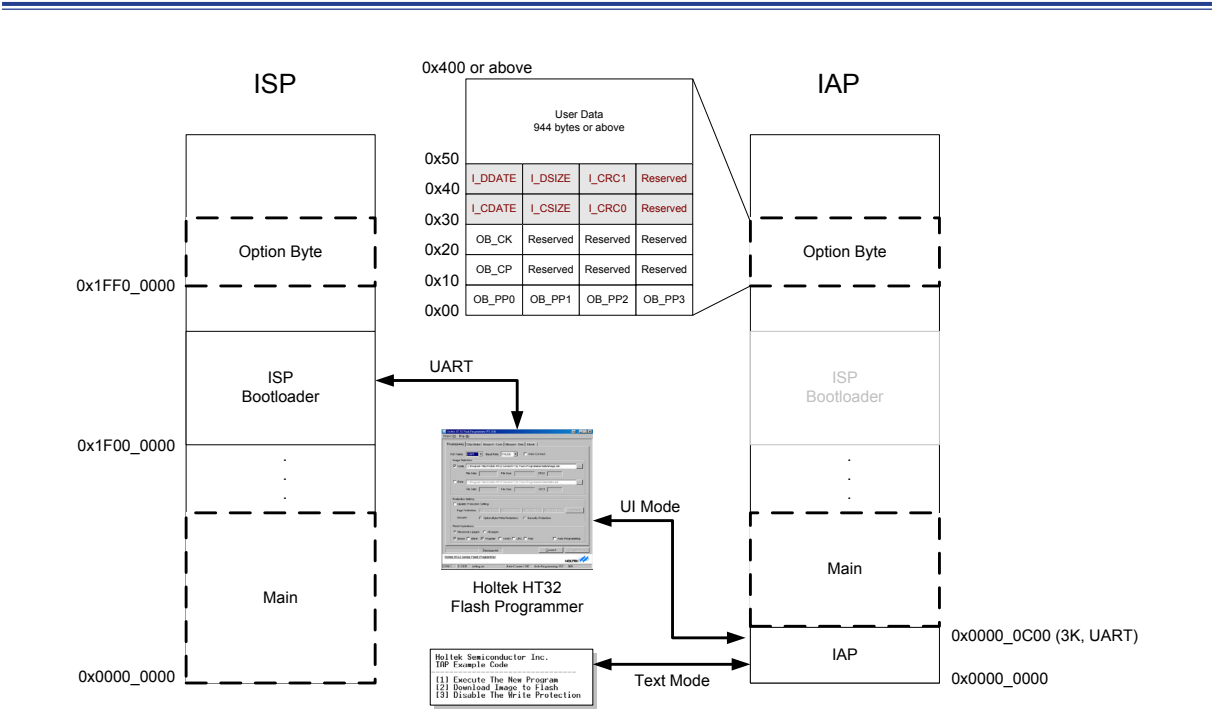


图 2. ISP/IAP 的存储配置

2 使用 HT32 Flash 编程器

该章节将描述如何安装 HT32 Flash 编程器, 下载图片, 配置字节选项和如何使用 Holtek 窗口动态连接库文件为 HT32 系列单片机开发用户 Flash 编程器。

安装 HT32 Flash 编程器

建议使用的系统要求:

- 运行在 Microsoft® Windows® X, Vista, 或者 Windows® 7 环境下的主机
- 建议使用有 1 GB RAM 和 20 MB 可用空间的硬盘
- 彩显解析度为 XGA (1024x768) 或者较高分辨率的显示器
- 鼠标或其他指示设备
- 可选 CD-ROM 驱动

以下步骤描述了如何安装 HT32 Flash 编程器。

步骤 1: 从 Holtek 官方网或者 Holtek 提供的 CD-ROM 中获得 HT32 Flash 编程器的最新版本。所需的文件诸如“HT32_Flash_Programmer_Vnnn.exe”, 其中“nnn”代表版本号。

步骤 2: 双击“HT32_Flash_Programmer_Vnnn.exe”和点击如下所示的对话框中的“Next”继续。

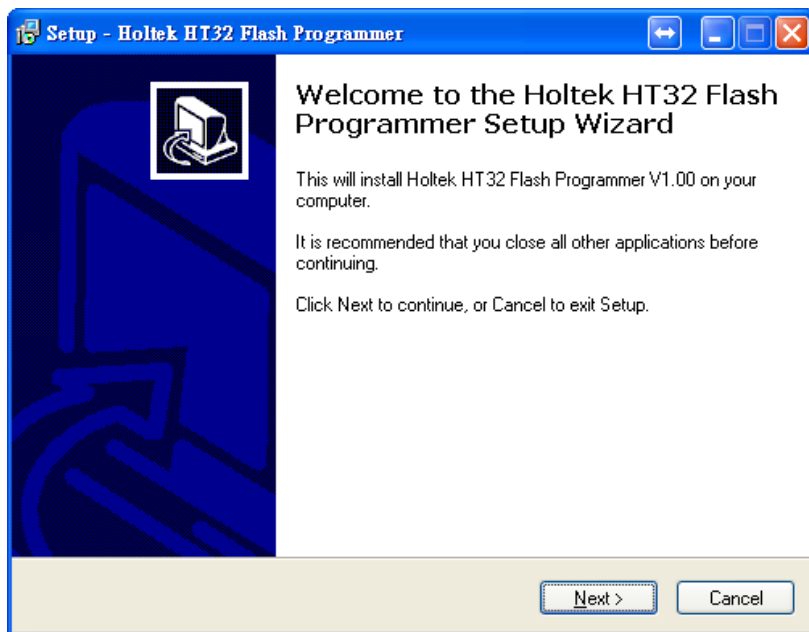


图 3. HT32 Flash 编程器安装 – 欢迎页面

步骤 3: 默认的安装路径是“C:\Program Files\Holtek HT32 Series\HT32 Flash Programmer”。
选择“Browse”改变安装路径。选择路径后, 点击“Next”继续安装。

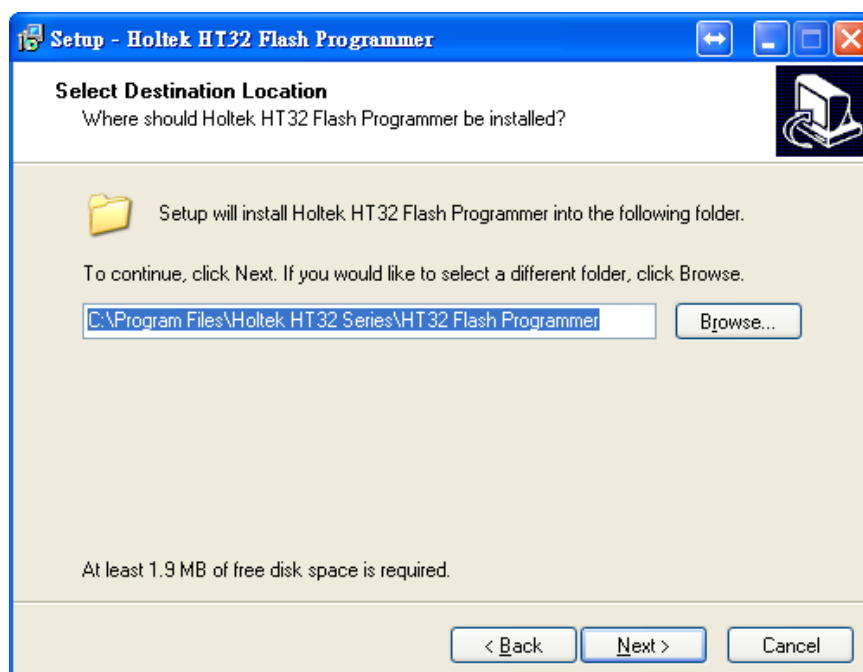


图 4. HT32 Flash 编程器安装 – 选择安装路径

步骤 4: 弹出“ready to install”界面时点击“Install”开始安装。

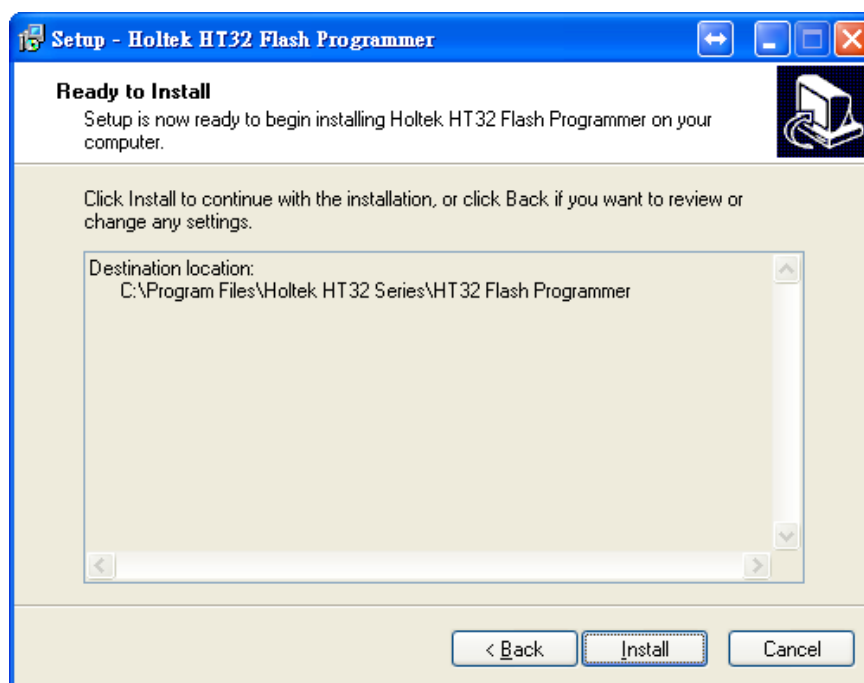


图 5. HT32 Flash 编程器安装 – 准备安装页面

步骤 5: 安装完成, 弹出如下所示的完成界面。选择是否浏览发布说明还是启动 HT32 Flash 编程器。点击“Finish”结束安装。

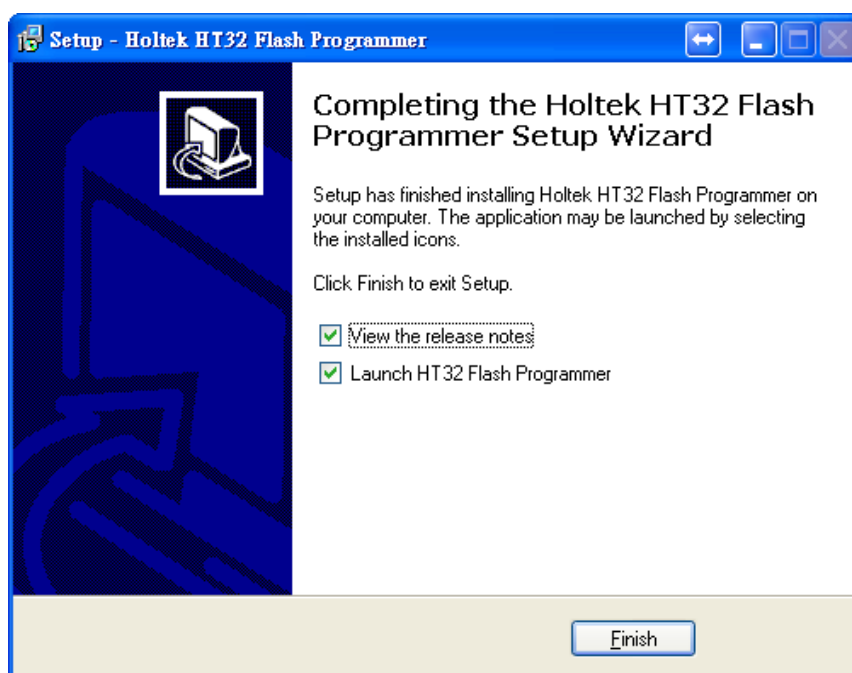


图 6. HT32 Flash 编程器安装 – 完成

HT32 Flash 编程器快速启动

启动 Flash 编程器，所需元件如下所示：

- HT32 系列单片机目标板
- 运行在 Microsoft® Windows® XP, Vista, 或者 Windows® 7 环境的主机
- 如上所述完整的 HT32 Flash 编程器安装

步骤 1: 将目标板连接到带有 RS232 接口的 PC 机的 COM 口。

步骤 2: 确保 BOOT0 和 BOOT1 引脚做如下配置：

ISP: BOOT1 = 0, BOOT0 = 0 或者 BOOT1 = 0, BOOT0 = 1

参考各个器件的用户手册

IAP: BOOT1 = 1, BOOT0 = 0

步骤 3: 双击“Start → All Programs → Holtek HT32 Series → Holtek HT32 Flash Programmer → HT32 Flash Programmer”进行编程。

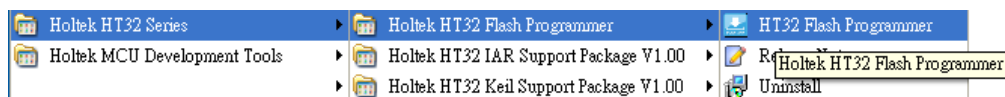


图 7. 启动 HT32 Flash 编程器

步骤 4: 点击“Project → New Project”重新存储默认值设置。若需要，改变 COM 口。确认波特率是 115200。

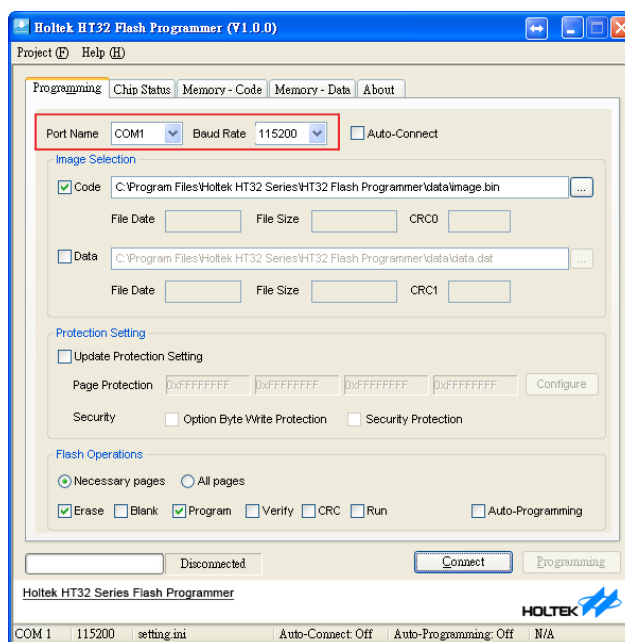


图 8. HT32 Flash 编程器主屏

步骤 5: 打开电源按下“Connect”进行连接。若连接成功, 会出现一个“Connected”的提示状态。器件名如下所示显示。按下“Programming”开始下载。

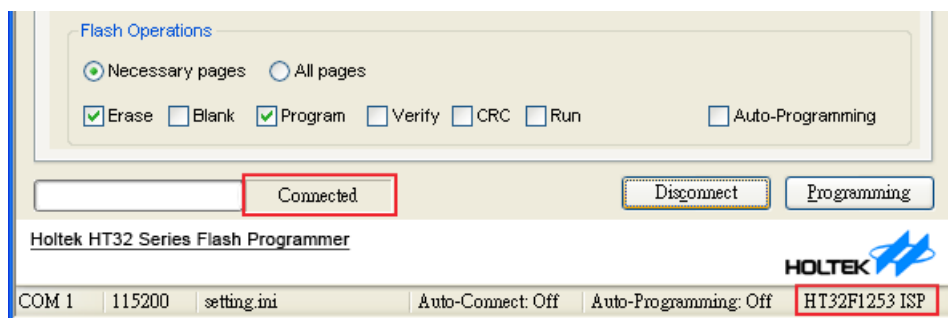


图 9. 连接状态

步骤 6: 若运行成功, 会弹出如下所示“Operations success”的提示。

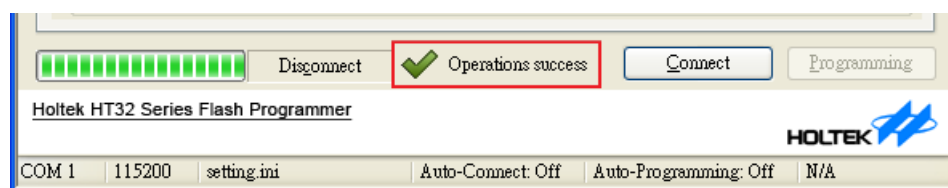


图 10. 运行成功屏

步骤 7: 按下“Connect”再次连接和改变选项到“Memory-Code”页。按下“Read”从 Flash 读取数据。要是选择默认值，如下所示一个空数据“image.bin”会编入 Flash 存储器进行测试。

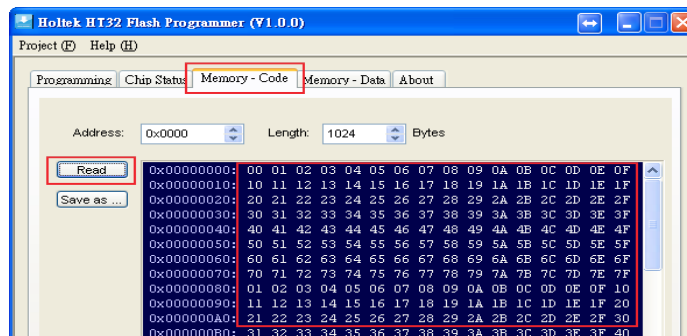


图 11. 读 Flash 存储器

此时，HT32 Flash 编程器已可用。欲知更多详情，请参考以下章节。

功能介绍

在这一章节详细描述了 HT32 Flash 编程器工作的各个功能。

连接

HT32 Flash 编程器通过 RS232 接口与目标器件进行通信。相关的编程器设置如下所示。在确定 COM 口和波特率的设置正确后选择“Connect”进行所需连接。如何使用自动连接功能会在下一章节进行探讨。

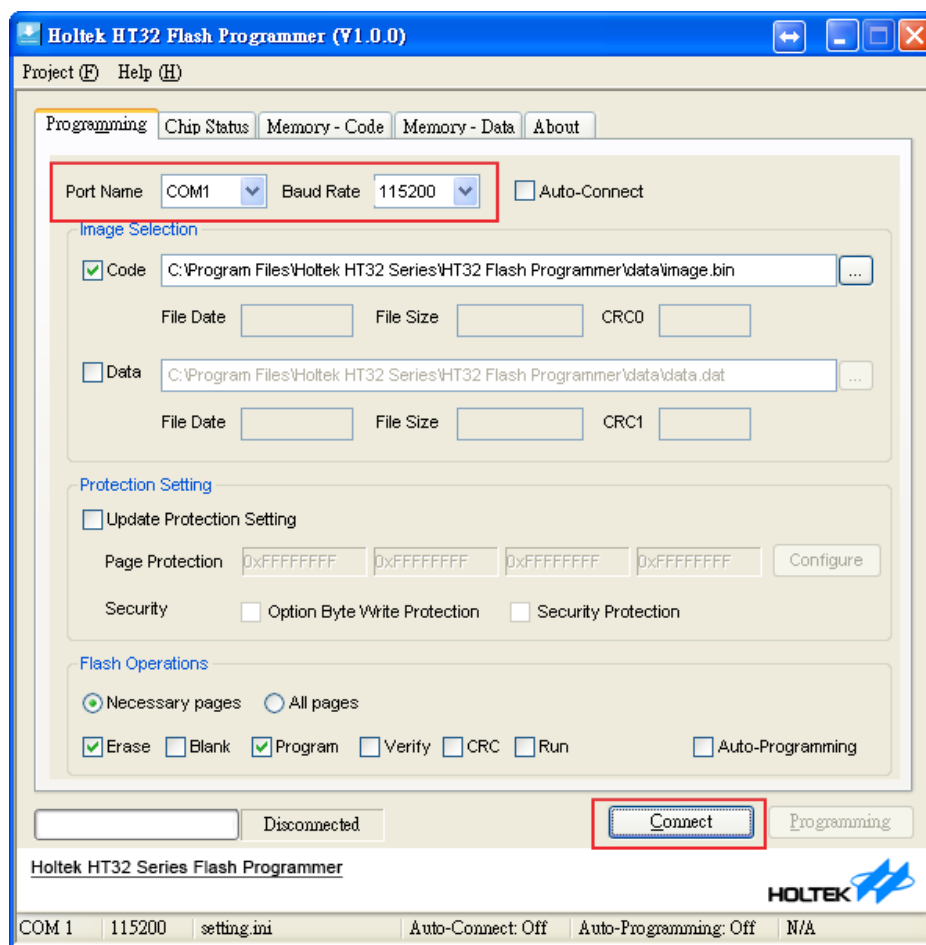


图 12. 连接和相关设定

图像选择

选择“...”，文件浏览按钮选择图像文件里支持二进制和十六进制的代码或者数据。若目标已连接，将更新文件日期、大小和 CRC 值的相关信息。仅当检查完代码或数据复选框后，Flash 操作会被应用到代码或数据部分。

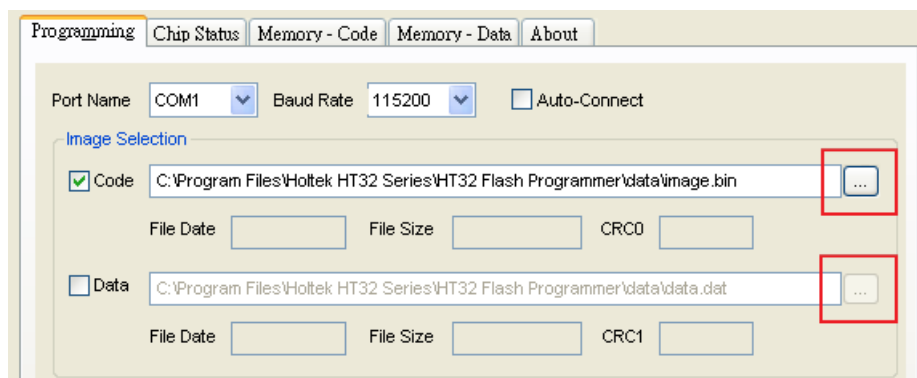


图 13. 图像选择

保护设定

选择“Configure”如下所示设置保护值。设置窗口左边描述了实际可选字节的值，当复选框状态值改变的时候，可动态更新。使用单个“All”文件选择特定的端口。点击“OK”可保留相关值。注意，在 IAP 模式可选字节写入和安全保护功能无效。

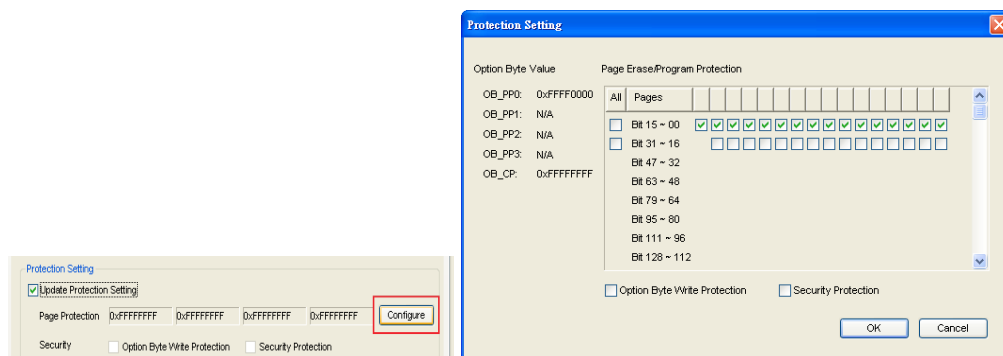


图 14. 保护设定

Flash 操作

Flash 操作的主要对象是代码、数据和保护设置。选择相应的复选框，执行不同的操作，例如，Flash 操作只能部分更新。

“Necessary pages”和“All pages”用来选择擦除器件和进行空白检查。若选择“Necessary pages”，只影响指定的 Flash 页。该页的目标范围由图像大小决定。若选择“All pages”，所有的 Flash 存储器页可被擦除或者进行空白检查。该功能可用于擦除所有的器件内容。如图所示是 Flash 操作的相关设定。

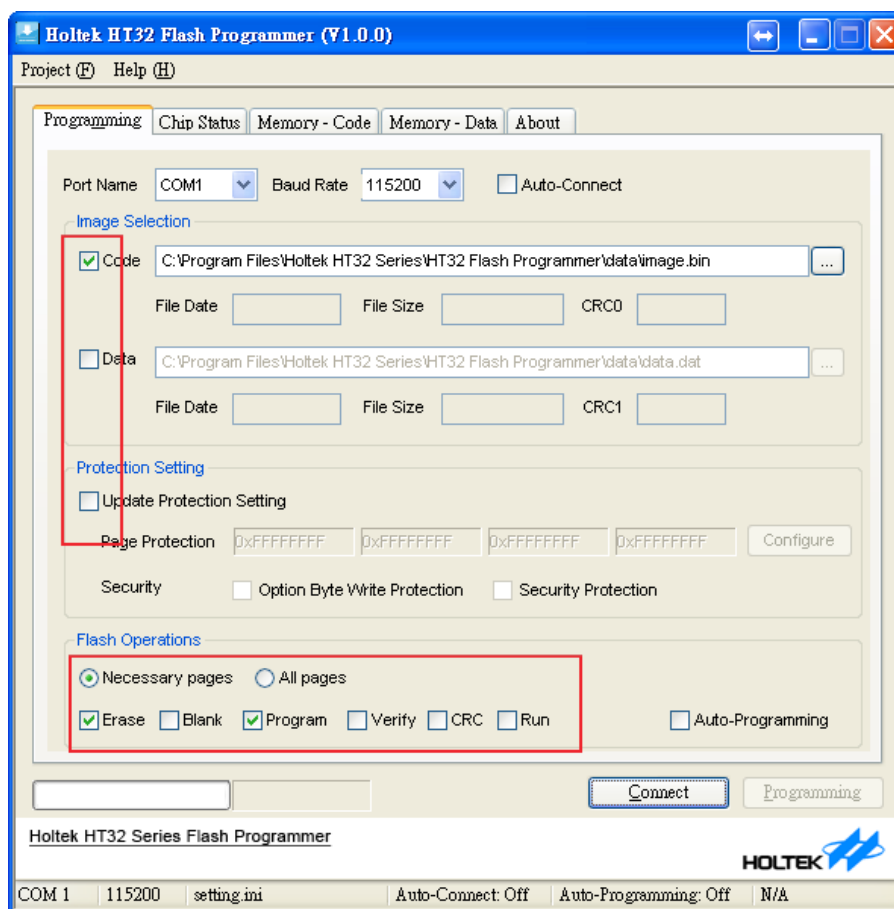


图 15. Flash 操作相关设定

处理状态和结果信息

处理状态和结果有助于理解运行情况。相关信息如下图所示。所有信息描述如表 1 所示。

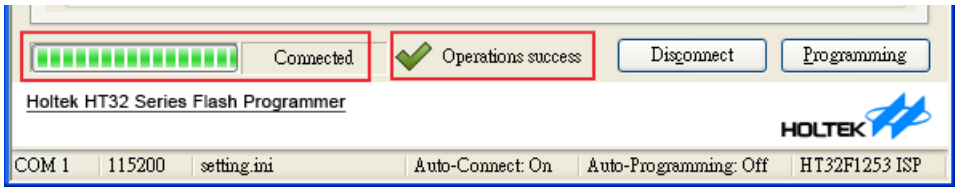


图 16. 处理状态和结果

表 1. 处理状态和结果信息

处理状态	结果信息	描述
未连接	—	编程器处于空闲模式。COM 口关闭。
在连接	—	编程器尝试连接到设备。
已连接	—	设备已连接。
擦除	—	擦除正在运行。
空白检查	—	空白检查正在运行。
编程	—	编程正在运行。
验证	—	验证正在运行。
CRC 检错	—	CRC 检错正在运行。
未连接，已连接或在连接（根据自动 - 连接和运行的设定）	操作成功	所有操作成功。
	擦除失败	擦除失败。
	空白检查失败	空白检查失败。
	编程失败	编程失败。
	验证失败	验证失败。
	CRC 检错失败	CRC 检错失败。

自动连接和自动编程

HT32 Flash 编程器提供有自动连接和自动编程功能。下图中描述了如何使用这两种功能。当自动连接功能使能时，若其他任何预置操作已完成，无论是启动后或者再次连接完，都会自动连接到 HT32 Flash 编程器。若器件已连接，HT32 Flash 编程器自动开始 Flash 操作。自动连接和自动编程的功能有利于减少大容量器件编程期间的手动操作时间。

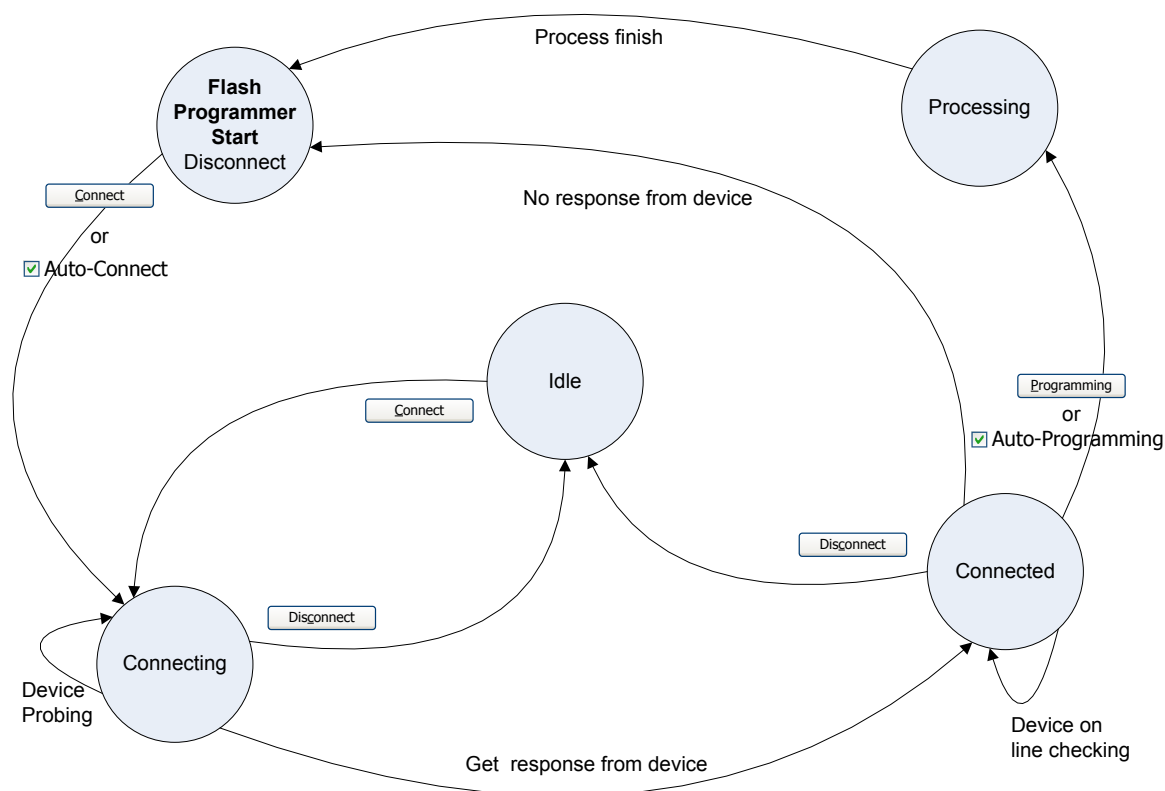


图 17. 自动连接和自动编程

器件状态

连接好编程器，HT32 Flash 编程器可查询器件名、Flash 大小和连接模式的相关信息并用状态表显示。若已编程，图像信息比如日期、大小和 CRC 值将会显示在右上角，同时显示当前保护设定值。

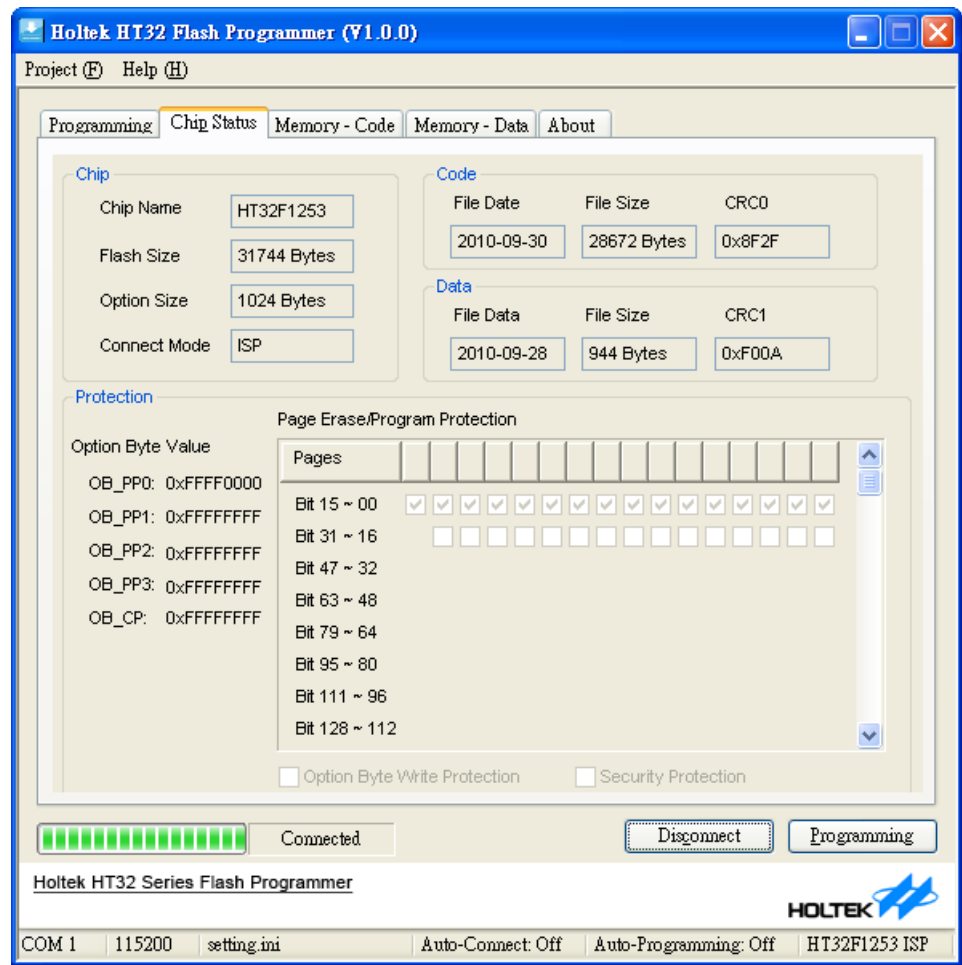


图 18. 芯片状态

使用
HT32 Flash
编程器

读 Flash 存储器

“Memory-Code”选项和“Memory-Data”选项为调试提供了“memory display”和“save as”功能。连接好编程器，进入起始地址和长度，选择“Read”执行读操作。HT32 Flash 编程器 UI 将会收到存储器数据并将其显示在屏幕上。右击存储器窗口如下所示，改变显示宽度为字节，半字或全字。特定的内容点击“Save as”用二进制格式存储以备后用。

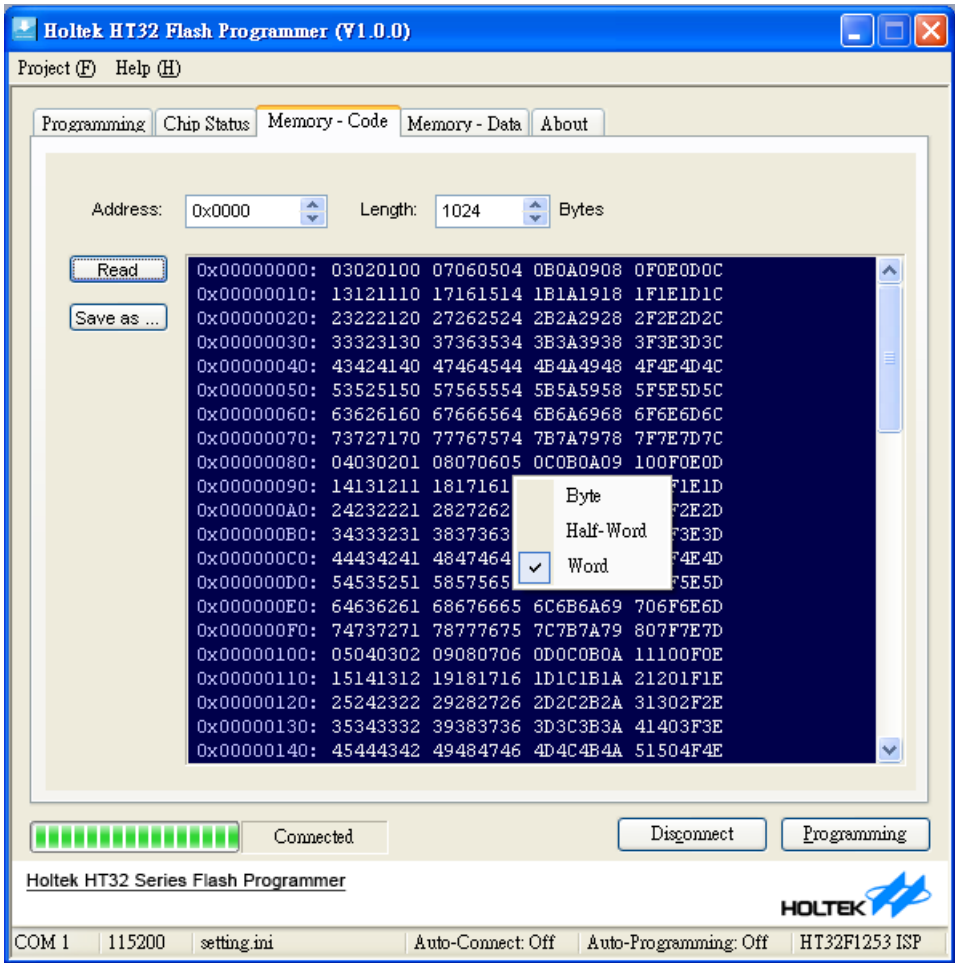


图 19. 读 Flash 存储器

使用
HT32 Flash
编程器

工程操作

下图描述了工程操作。利用“create”，“open”，“save”和“save as”等菜单进行 HT32 Flash 编程器的设定。编程 Flash 时，下载文档“setting.ini”作为默认文档以减少设置时间。若文件有更改，“Save Project”会把值存储到“setting.ini”文件。如果选择“Project”菜单中的“New Project”，所有的控制被复位到默认值。该默认值存储在 HT32 Flash 编程器安装路径的“default.ini”文件。“Save Project As”将会把设定值存储到另一个特定的文件，而“Open Project”将会打开所选的文件作为当前设定。欲知详情，请参考下图。

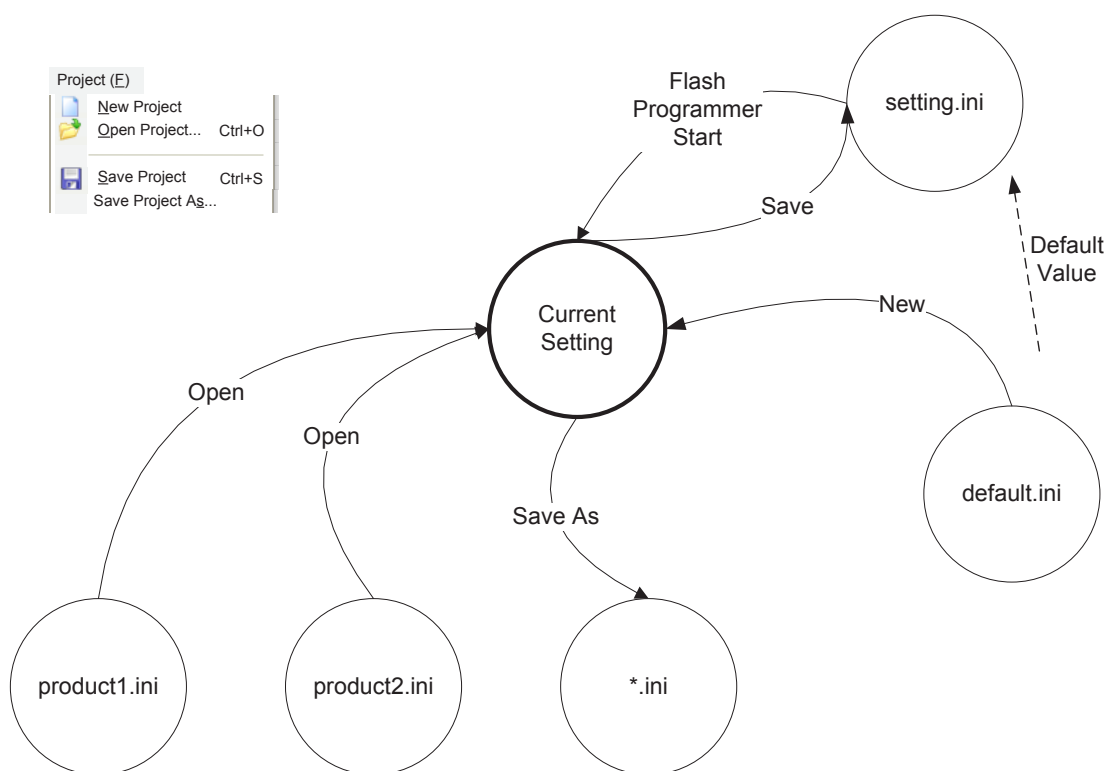


图 20. 工程文件操作

为 HT32 系列单片机开发 Flash 编程器

简介

Holtek 为 Flash 编程提供了动态连接库“ISPCmd.dll”，执行 HT32 Flash 的命令协议和通信 APIs 与 ISP/IAP 固件的通信。和 HT32 Flash 编程器一样，软件编程器可以添加 DLL 库到它们自己的工程并使用 API 功能对器件进行编程。下列两个表格中是 API 功能清单和“ISPCmd.dll”的返回代码。各个功能的详细信息将在以下章节进行探讨。

表 2. 动态数据链接库 API 窗口

类型	API 名称	描述
连接	ISP_SetupSerialPortConnection	连接到编程器
	ISP_CloseSerialPortConnect	关闭编程器连接
控制	ISP_ErasePage	页擦除
	ISP_EraseMass	整体擦除
	ISP_BlankCheck	空白检查
	ISP_WriteProgramB	编程或验证
	ISP_CRCCheck	Flash 存储器 CRC 检错
	ISP_ReadData	从设备读存储器
	ISP_Execute	执行用户编程
	ISP_Exit	退出命令模式
	ISP_Reset	复位编程器
状态	ISP_IsConnectedToDevice	检查连接
	ISP_GetTransProgress	获取编程或验证的进度
	ISP_GetInformation	获取设备信息

表 3. API 的返回代码

返回代码	值
ISP_CMD_SUCCESS	0x00
ISP_CMD_EXECUTE_FAILED	0x01
ISP_CMD_EXECUTE_BADCMD	0x02
ISP_ECHO_ERROR	0x03
ISP_BAD_FW_REPLY_DATA	0x04
ISP_SECURITY_LOCK	0x05
ISP_WRITE_PROTECT	0x06
ISP_BAD_PROGRAMMING	0x07
ISP_BAD_PROGRAMMING_NO_FIND_ACK	0x08
ISP_BAD_PROGRAMMING_TIMEOUT	0x09
ISP_BAD_PROGRAMMING_OPEN_FILE_NG	0x0a
ISP_BAD_UNABLE_EXITXMODEM	0x0b
ISP_SERIAL_PORT_OPEN_FAILED	0x0c
ISP_SERIAL_PORT_NOT_CONNECTED	0x0d
ISP_SERIAL_PORT_UNABLE_WRITE	0x0e
ISP_SERIAL_PORT_UNABLE_READ	0x0f
ISP_DEVICE_NOT_FOUND	0x10
ISP_RESET_ERROR	0x11
ISP_INVALID_PARAMETER	0x12
ISP_FUNCTION_NOT_SUPPORTED	0x13
ISP_SYSTEM_ERROR_CODE	0x14
ISP_UNKNOWN_ERR	0xff

动态数据链接库 API 描述

■ ISP_SetupSerialPortConnection

表 4. ISP_SetupSerialPortConnection 函数

API	INT ISP_SetupSerialPortConnection(INT nPort, DWORD baud_rate);
描述	打开特定的串口连接到编程器。
参数	nPort: 特定串口 Baud_rate: 特定波特率
返回	ISP_CMD_OK: 操作成功 ISP_SERIAL_PORT_OPEN_FAILED: 无法打开串口
范例	<pre>If (!ISP_SetupSerialPortConnection (1, 115200)) { // Unable to connect the serial port... ISP_CloseSerialPortConnect (); }</pre>

■ ISP_CloseSerialPortConnect

表 5. ISP_CloseSerialPortConnect 函数

API	VOID ISP_CloseSerialPortConnect(VOID);
描述	关闭串口。
参数	无
返回	无
范例	<pre>If (!ISP_SetupSerialPortConnection (1, 115200)) { // Unable to connect the serial port... ISP_CloseSerialPortConnect (); }</pre>

使用 HT32 Flash 编程器

■ ISP_ErasePage

表 6. ISP_ErasePage 函数

API	INT ISP_ErasePage(DWORD dwStartAddr, DWORD dwEndAddr);
描述	根据参数 dwStartAddr 和 dwEndAddr 进行页擦除。
参数	dwStartAddr:Flash 存储器起始地址 dwEndAddr:Flash 存储器终止地址
返回	ISP_CMD_OK: 操作成功 ISP_INVALID_PARAMETER: 无效参数 ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口 ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口 ISP_SERIAL_PORT_UNALBE_WRITE: 无法写 COM 口 ISP_CMD_EXECUTE_FAILED: 操作失败
范例	ISP_ErasePage(0xc00, 0xffff); // 从 0xc00 到 0xffff 擦除

■ ISP_EraseMass

表 7. ISP_EraseMass 函数

API	INT ISP_EraseMass(VOID);
描述	在 Flash 存储器进行整体擦除。
参数	无
返回	ISP_CMD_OK: 操作成功 ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口 ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口 ISP_SERIAL_PORT_UNALBE_WRITE: 无法写 COM 口 ISP_CMD_EXECUTE_FAILED: 操作失败
范例	ISP_EraseMass(); // 开始整体擦除

■ ISP_BlankCheck

表 8. ISP_BlankCheck 函数

API	INT ISP_BlankCheck(DWORD dwStartAddr, DWORD dwEndAddr);
描述	根据参数 dwStartAddr 和 dwEndAddr，在特定范围内执行空白检查。
参数	dwStartAddr: Flash 存储器起始地址 dwEndAddr: Flash 存储器终止地址
返回	ISP_CMD_OK: 操作成功 ISP_INVALID_PARAMETER: 无效参数 ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口 ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口 ISP_SERIAL_PORT_UNABLE_WRITE: 无法写 COM 口 ISP_CMD_EXECUTE_FAILED: 操作失败
范例	ISP_BlankCheck(0xc00, 0xfff); // 在 0xc00~0xfff 进行空白检查

■ ISP_WriteProgramB

表 9. ISP_WriteProgramB 函数

API	INT ISP_WriteProgramB(DWORD dwStartAddress, DWORD dwEndAddress, BYTE mode, BYTE * pbValue);
描述	开始编程或验证的 Xmodem 协议。
参数	dwStartAddr: Flash 存储器起始地址 dwEndAddr: Flash 存储器终止地址 mode: 1 为编程，0 为验证。 pbValue: 写入数据的缓冲指针。
返回	ISP_CMD_OK: 操作成功 ISP_INVALID_PARAMETER: 无效参数 ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口 ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口 ISP_SERIAL_PORT_UNABLE_WRITE: 无法写 COM 口 ISP_BAD_PROGRAMMING_NO_FIND_ACK: 无法搜寻到确认符 ISP_BAD_PROGRAMMING_TIMEOUT: 超出编程时间 ISP_BAD_UNABLE_EXITXMODEM: Xmodem 协议出错 ISP_BAD_PROGRAMMING: 编程出错
范例	Byte buffer[16]; buffer[0] = 0x55; ... ISP_WriteProgramB(0xc00, 0xc0f, 1, buffer); // 写入主 Flash 地址 0xc0016 字节

■ ISP_CRCCheck

表 10. ISP_CRCCheck 函数

API	INT ISP_CRCCheck(DWORD dwStartAddr, DWORD dwEndAddr, BYTE *pbValue);
描述	Flash 存储器 CRC 检错。
参数	dwStartAddr: Flash 存储器起始地址
	dwEndAddr: Flash 存储器终止地址
	pbValue: CRC 结果的缓冲指针
返回	ISP_CMD_OK: 操作成功
	ISP_INVALID_PARAMETER: 无效参数
	ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口
	ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口
	ISP_SERIAL_PORT_UNABLE_WRITE: 无法写 COM 口
	ISP_CMD_EXECUTE_FAILED: 操作失败
范例	BYTE bValue[2]; ISP_CRCCheck(0xc00, 0xffff, &bValue);

■ ISP_ReadData

表 11. ISP_ReadData 函数

API	INT ISP_ReadData(DWORD dwStartAddress, DWORD dwEndAddr, BYTE* pbValue);
描述	从设备读取 Flash 存储器数据。
参数	dwStartAddr: Flash 存储器始地址
	dwEndAddr: Flash 存储器终止地址
	pbValue: 读取数据的缓冲指针。
返回	ISP_CMD_OK: 操作成功
	ISP_INVALID_PARAMETER: 无效参数
	ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口
	ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口
	ISP_SERIAL_PORT_UNABLE_WRITE: 无法写 COM 口
	ISP_SECURITY_LOCK: 安全锁使能
范例	ISP_BAD_FW_REPLY_DATA: 固件返回值无效
	Byte buffer[16]; ISP_ReadData(0xc00, 0xc0f, buffer); // 从主 Flash 地址 0xc00 读取 16 字节

■ ISP_Execute

表 12. ISP_Execute 函数

API	INT ISP_Execute(VOID);
参数	无
返回	ISP_CMD_OK: 操作成功
	ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口
	ISP_DEVICE_NOT_FOUND: 搜寻不到设备
	ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口
	ISP_SERIAL_PORT_UNABLE_WRITE: 无法写 COM 口
	ISP_CMD_EXECUTE_FAILED: 操作失败
范例	ISP_Execute();

■ ISP_Exit

表 13. ISP_Exit 函数

API	INT ISP_Exit(VOID);
参数	无
返回	ISP_CMD_OK: 操作成功
	ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口
	ISP_DEVICE_NOT_FOUND: 搜寻不到设备
	ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口
	ISP_SERIAL_PORT_UNABLE_WRITE: 无法写 COM 口
范例	ISP_Exit();

■ ISP_Reset

表 14. ISP_Reset 函数

API	INT ISP_Reset(VOID);
描述	复位设备。
参数	无
返回	ISP_CMD_OK: 操作成功
	ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口
	ISP_DEVICE_NOT_FOUND: 搜寻不到设备
	ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口
	ISP_SERIAL_PORT_UNABLE_WRITE: 无法写 COM 口
	ISP_RESET_ERROR: 复位出错
范例	ISP_Reset();

■ ISP_IsConnectedToDevice

表 15. ISP_IsConnectedToDevice 函数

API	BOOL ISP_IsConnectedToDevice(VOID);
描述	检查设备连接状态。
参数	无
返回	TRUE(1): 设备已连接 FALSE(0): 无法连接设备
范例	<pre>if (ISP_IsConnectedToDevice ()) { // Find the device... } else { // Not find the device... }</pre>

■ ISP_GetTransProgress

表 16. ISP_GetTransProgress 函数

API	DWORD ISP_GetTransProgress(VOID);
描述	返回编程或验证的百分比进度。
参数	无
返回	百分比进度 : 从 0 到 100
范例	<pre>DOWRD dwCurrProgress; dwCurrProgress = ISP_GetTransProgress ();</pre>

■ ISP_GetInformation

表 17. ISP_GetInformation 函数

API	INT ISP_GetInformation(DWORD nCode, BYTE *pbValue, int *pLen);
描述	根据参数 nCode 从设备获取信息。
参数	nCode: 信息选择器。欲知详情，请参考表 18。 pbValue: 保存结果的缓冲器指针 pLen: 返回信息字节长度
返回	ISP_CMD_OK: 操作成功 ISP_SERIAL_PORT_NOT_CONNECTED: 无法连接 COM 口 ISP_DEVICE_NOT_FOUND: 搜寻不到设备 ISP_SERIAL_PORT_UNABLE_READ: 无法读 COM 口 ISP_SERIAL_PORT_UNALBE_WRITE: 无法写 COM 口 ISP_CMD_EXECUTE_FAILED: 操作失败
范例	<pre>Byte buffer[255]; INT nLen; ISP_GetInformation(0, buffer, &nLen); // 获取芯片名</pre>

表 18. ISP_GetInformation 函数的 nCode 定义

nCode	描述	范例
0	获取芯片名	HT32F1253O HT32F1251O
1	获取 Flash 大小	31744O 8192O
2	获取选项大小	1024O
3	获取 OB_PP 位数	31O 8O
4	获取固件版本	ISPV1.0.0O IAPV1.0.0O
5	获取固件日期	2011-03-11O
6	获取 ISP/IAP 起始地址	0O 4096O 8192O
其他	不支持	F

入门实例

使用“ISPCmd.dll”提供的“ISPCmdSimpleDemo.zip” VC++ 6.0 的程序代码。在 HT32 Flash 编程器安装路径“C:\Program Files\Holtek HT32 Series\HT32 Flash Programmer\ VC++6.0_Example\”可找到该程序代码。该程序代码展示了诸如获取信息, 读 / 写数据, 擦除数据等 DLL 函数。以下步骤展示了如何开始该程序实例。

步骤 1: 解压“ISPCmdSimpleDemo.zip”。

步骤 2: 编程器上电, 用线连接到 PC 机。确保编程器可连接。

步骤 3: 双击“Output\ISPCmdSimpleDemo.exe”, 弹出如下图所示的范例程序用户接口。

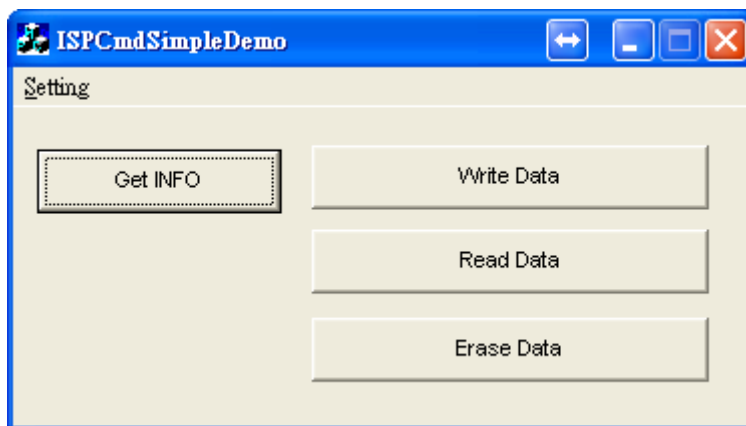


图 21. VC++6.0 ISPCmd.dll 程序代码

步骤 4: 选择“Setting”如下图所示配置 COM 口和波特率。

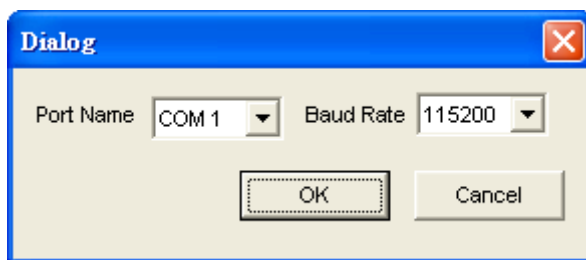


图 22. 配置 COM 口和波特率

步骤 5: 按下屏幕上的按键测试相关功能。

该范例程序代码是使用微软 VC++ 6.0 设计的。开发一个 Flash 编程器, 可从该范例程序代码开始或者按照以下的步骤建立一个新工程。

步骤 1: 启动微软 VC++ 6.0, 通过 MFC AppWizard 建立一个新的工程。

步骤 2: 从“ISPCmdSimpleDemo.zip”复制“Common.h”, “ISPCmd.dll”, “ISPCmd.lib” 和 “ISPCmdExport.h” 到工程文件夹。

步骤 3: 把“ISPCmd.lib” 和 “ISPCmdExport.h” 包含到工程里。

步骤 4: 欲使用 DLL 函数, 请参考其他章节对 API 的描述。

3 使用 ISP 和 IAP 固件

该章节将描述了 HT32 Flash 命令协议的详情及命令格式。介绍了如何使用 ISP 启动加载器，在此章节的最后小结讨论了 IAP 的范例程序代码。

HT32 Flash 命令协议

协议描述

HT32 Flash 编程器协议是基于 ASCII, 用质询 - 响应来更新固件的协议。它支持的命令设置可达 7 条: 获取信息, 擦除, 空白检查, 编程 / 验证, CRC 检错, 执行和读存储器。协议的架构如下图所示。该协议由三部分组成: 参数命令, 返回值和确认。主机发送一条命令给器件, 从 1 字节命令字符开始, 然后是可选参数, 最后发送终止字符 (CE, 0x0D)。注意, 设备接收到的每个字节将会被直接返送回主机检错。一旦设备接收到命令终止字符, 将回复 2 字节的确认字符 (ACK 和 0x0D 0x0A) 以示命令已成功接收。设备开始分析该命令 / 参数和根据所需的 Flash 操作进行处理。处理完, 设备发送可选信息, 1 字节的状态和 2 字节的确认字符到主机。最后, 设备发送一个提示字符如 “ISP>” 或 “IAP>” 提示操作已完成, 准备接收下一个命令。主机会检查状态字符以确认操作是否成功。各个符号的格式如下图所示。

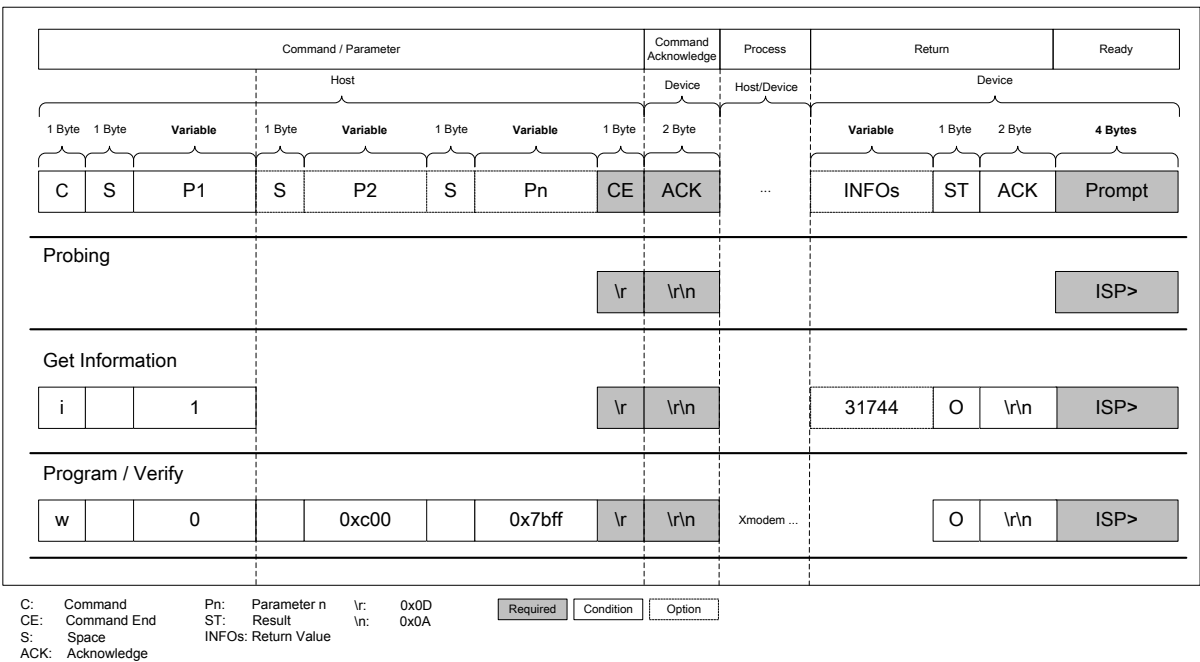


图 23. Holtek Flash 编程器协议

命令格式

下表是 HT32 Flash 命令协议所支持的命令。下面将探讨每条命令的详细内容。

表 19. 命令描述

描述	命令 C	参数	返回
		S P1 S P2	INFOs ST
获取信息	i	[TYPE]	[INFO1] [INFO2]... [INFOn][ST]
擦除	e	[TYPE] [SADDR] [EADDR]	[ST]
空白检查	b	[SADDR] [EADDR]	[ST]
编程或验证	w	[TYPE] [SADDR] [EADDR]	[ST]
CRC 检错	c	[SADDR] [EADDR]	[CRC][ST]
执行，退出或复位	g	[TYPE]	None
读存储	r	[R1] [SADDR] [EADDR]	[Byte stream][ST]

□: 空间 (0x20)
注意: 命令 [SADDR] 和 [EADDR] 必须用小写字体。

使用 ISP 和 IAP 固件

命令描述

■ 获取信息 (i)

表 20. 获取信息命令 (i)

描述	根据参数 [TYPE] 从设备获取信息。		
格式	命令： i □ [TYPE][CE] 返回： [INFO][ST]		
	内容	描述	
参数	[TYPE]	信息类型。欲知详情，参考表 21。	
返回	[INFO]	返回信息。欲知详情，参考表 21。	
	[ST]	结果 O: 获取信息成功 F: 不支持 [TYPE]	
范例	描述	主机	设备
	获取芯片名	i □ 0	HT32F1253O
	获取 Flash 大小	i □ 1	31744O

表 21. 获取信息命令的类型定义

类型	描述	返回	范例	单元
0	获取芯片名	[INFO][ST]	HT32F1253O HT32F1251O	文本 (ASCII)
1	获取 Flash 大小	[INFO][ST]	31744O 8192O	字节 (ASCII)
2	获取选项大小	[INFO][ST]	1024O	字节 (ASCII)
3	获取 OB_PP 位数	[INFO][ST]	31O 8O	位数 (ASCII)
4	获取固件版本	[INFO][ST]	ISPV1.0.0O IAPV1.0.0O	文本 (ASCII)
5	获取固件日期	[INFO][ST]	2011-03-11O	文本 (ASCII)
6	获取 ISP/IAP 起始地址	[INFO][ST]	0O 4096O 8192O	字节 (ASCII)
其他	不支持	[ST]	F	状态

■ 整体 / 页擦除 (e)

表 22. 整体 / 页擦除命令 (e)

描述	根据参数 [TYPE] 进行整体 / 页擦除。		
格式	命令： e □ [TYPE] □ [SADDR] □ [EADDR] 返回： [ST]		
	内容	描述	
参数	[TYPE]	0: 整体擦除 (注 1) 1: 页擦除	
	[SADDR]	起始地址 (注 2)。范围： 0x0 ~ [EADDR]	
	[EADDR]	终止地址 (注 2)。范围： [SADDR]~ 芯片最大地址	
返回	[ST]	结果 O: 擦除成功 F: 擦除失败 - 地址溢出或擦除类型不支持	
范例	描述	主机	设备
	整体擦除	e □ 0	O
	页擦除	e □ 1 □ 0x0 □ 0x17ff	O

注意： 1. IAP 模式只支持页擦除命令。当接收到整体擦除的命令时，IAP 固件返回 [ST] = F。
2. [SADDR] 和 [EADDR] 只在页擦除模式有效，即 [TYPE] = 1。

■ 空白检查 (b)

表 23. 空白检查命令 (b)

描述	根据参数 [SADDR] 和 [EADDR] 在特定范围内进行空白检查。		
格式	命令： b □ [SADDR] □ [EADDR] 返回： [ST]		
	内容	描述	
参数	[SADDR]	起始地址。范围： 0x0 ~ [EADDR]	
	[EADDR]	终止地址。范围： [SADDR]~ 芯片最大地址	
返回	[ST]	结果 O: 空白检查成功 F: 空白检查失败 - 某些 Flash 内容不是 0xFFFFFFFF	
范例	描述	主机	设备
	空白检查	b □ 0x0 □ 0xffff	O

■ 编程 / 验证 (w)

表 24. 编程 / 验证命令 (w)

描述	执行 Xmodem 协议的编程或验证。		
格式	命令： w □ [TYPE] □ [SADDR] □ [EADDR] 返回： [ST]		
	内容	描述	
参数	[TYPE]	编程或验证模式 0: 验证模式 1: 编程模式	
	[SADDR]	编程或验证的起始地址。范围： 0x0 ~ [EADDR]	
	[EADDR]	界内检查的终止地址。范围： [SADDR] ~ 芯片最大地址	
返回	[ST]	结果 O: 编程 / 验证成功 F: F 编程 / 验证失败 超出地址范围 /Flash 出错 /Xmodem CRC 出错等	
范例	描述	主机	设备
	编程	w □ 0x1 □ 0x0 □ 0x1fff	O
	验证	w □ 0x0 □ 0x0 □ 0x1fff	O

■ CRC 检查 (c)

表 25. CRC 检查 (c)

描述	根据参数 [SADDR] 和 [EADDR] 执行 CRC 检错。		
格式	命令： c □ [SADDR] □ [EADDR] 返回： [CRC][ST]		
	内容	描述	
参数	[SADDR]	CRC 运算的起始地址。范围： 0x0~[EADDR]	
返回	[EADDR]	CRC 运算的终止地址。范围： [SADDR]~ 芯片最大地址	
范例	[CRC]	16-bit CRC 值	
	[ST]	结果 O: CRC 运算成功	
	描述	主机	设备
	CRC 检错	c □ 0x0 □ 0x1fff	[ByteLByteH]O

■ 执行 / 退出 / 复位 (g)

表 26. 执行, 退出和复位命令 (g)

描述	开始执行用户应用, 退出 ISP/IAP 固件命令模式或者器件复位。		
格式	命令: g □ [TYPE] 返回: [ST] (Optional)		
	内容	描述	
参数	[TYPE]	执行, 退出或者复位 0: 执行 1: 退出 2: 复位	
返回	[ST]	执行结果 F: 当 [TYPE] = 0 不存在用户应用	
范例	描述	主机	器件
	执行	g □ 0	若失败为 F
	退出	g □ 1	无
	复位	g □ 2	无

■ 读存储器 (r)

表 27. 读存储器命令 (r)

描述	读芯片存储器内容		
格式	命令: r □ [R1] □ [SADDR] □ [EADDR] 返回: [Byte stream][ST]		
	内容	描述	
参数	[R1]	保留。为 0。	
	[SADDR]	起始地址。范围: 0x0 ~ [EADDR]	
	[EADDR]	终止地址。范围: [SADDR] ~ 芯片最大地址	
返回	[ST]	结果 O: 读取成功 F: 读取失败 - 安全锁使能或者地址溢出	
范例	描述	主机	器件
	读主 Flash	r □ 0 □ 0x0 □ 0x1fff	[Byte stream]O
	读可选字节	r □ 0 □ 0x1ff00000 □ 0x1ff00030	[Byte stream]O

使用 ISP 启动加载器

ISP 启动加载器是一个预装载程序, 利用 UART 接口, 通过 HT32 Flash 命令协议与 HT32 Flash 编程器进行通信。启动 ISP 启动加载器, 必须注意以下各项:

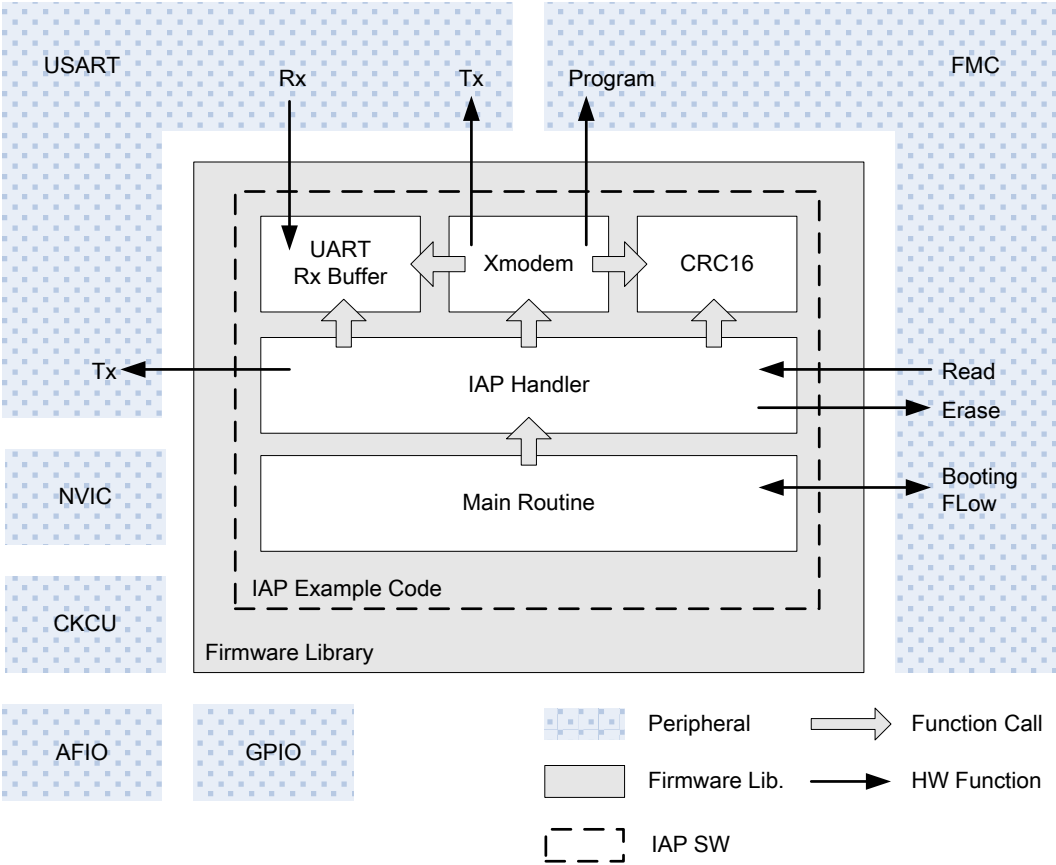
- 确保在目标板上的启动设置为 Bootloader。根据器件需设定为 “BOOT0 = 0, BOOT1 = 0” 或者 “BOOT0 = 0, BOOT1 = 1”。欲知详情, 请参考各个器件的用户手册。
- 连接目标板到带有 RS232 口的 PC 机的 COM 口。
- 启动 COM 口程序比如 HyperTerminal 或者 HT32 Flash 编程器。
- 确保 UART 已做如下设置:
 - 波特率: 115200bps
 - 数据: 8 位
 - 奇偶校验: 无
 - 终止位: 1 位
 - 流程控制: 无
- 打开设备电源。若连接无误, 在 HyperTerminal 或在 HT32 Flash 编程器的一个 “Connected” 信息框会有一个诸如 “ISP>” 的字符提示。

由于 ISP 启动加载器位于一个独立区域, 它可以擦除 / 编程所有的 Flash 存储器件。同时也支持整体擦除器件和安全保护功能。若最后的产品有 UART 接口和启动设置, 和 IAP 固件一样, ISP 启动加载器也可用于更新应用。

IAP 范例程序代码

简介

该章节描述了 IAP 范例程序代码的文本模式和 UI 模式。图 24 是 IAP 范例程序代码的架构。IAP 程序主要是“**IAP Handler**”模块，负责解析命令，并与其他模块共同工作，以完成所有的操作。“**UART Buffer**”模块利用 RAM 空间和 Rx 中断驱动程序保存主机发送的数据。利用 RAM 缓冲和驱动中断，可同时接收数据和 Flash 编程以减少耗时。“**IAP Handler**”不仅接收命令还有缓冲器发送的 Xmodem 协议包，利用 Flash 存储控制器把数据编入 Flash 存储器。另外，“**CRC16**”模块计算 Xmodem 协议和 Flash 校验和的 CRC 值。

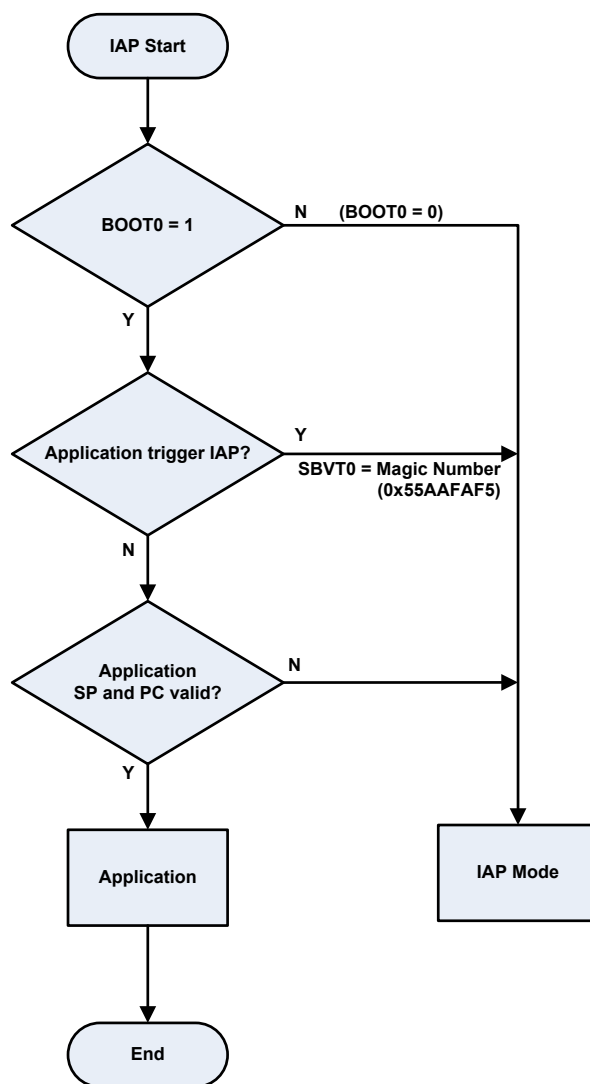


Architecture of IAP Example Code

图 24. IAP 范例程序代码架构

“Main Routine”主要处理系统的初始化,例如时钟, I/O 和中断。通过 BOOT0 决定是用户应用还是 IAP 模式需要初始化。下图是 IAP 范例程序代码的启动流程。

- 当 IO (BOOT0) 设置为 0 时, 进入 IAP 模式。
- 当应用触发时, 进入 IAP 模式。(应用设置 SBVT0 为 “0x55AAFAF5” 和触发复位)
- 当应用图像不存在时, 进入 IAP 模式。即检查应用向量格式。
- 否则, 开始执行应用编程。



Note: Application can set SBVT0 as 0x55AAFAF5 and trigger a reset to start IAP mode. SBVT registers only reset by Power-on-reset.

图 25. IAP 模式启动流程

IAP 范例程序代码是 HT32 系列单片机固件库的一部分。从 Holtek 官方网站或者 Holtek 提供的 CD-ROM 下载 HT32 系列单片机固件库的最新版本。如图 26 所示, IAP 范例程序代码包括四个文件夹。

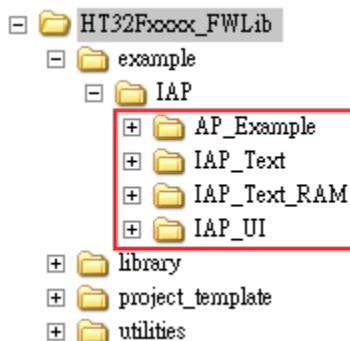


图 26. IAP 范例程序代码文件夹

“AP_Example” 描述了如何设置使用 IAP 的 RO 地址。在 IAP 应用中, RO 地址一般默认为 0xC00。首个 3k 字节是保留给 IAP 的。“IAP_Text” 和 “IAP_UI” 提供有使用 UART 接口的 IAP 模式的实例。IAP 文本模式和 IAP UI 模式的相关内容在下一节将有详细介绍。另外, “IAP_Text_RAM” 为 IAP 固件的自动更新提供代码。

IAP 文本模式范例程序代码

IAP 文本模式提供有一个文本菜单, 通过 UART 下载用户的图像到主 Flash。该程代码可在 HT32 系列单片机的开发板上运行。如下表所示是 IAP 文本模式的范例程序代码源文件。

表 28. 文本模式源文件

路径	文件名	描述
IAP_Text/Src_AP/	main.c	应用主程序
	ht32fxxxx_it.c / ht32fxxxx_it.h	应用中断处理程序
	ht32fxxxx_conf.h	固件库配置文件
	system_ht32fxxxx.c	系统配置文件
IAP_Text/Src_IAP/	iap_main.c	IAP 主程序
	iap_handler.c / iap_handler.h	控制更新过程的 IAP 处理程序
	iap_xmodem.c / iap_xmodem.h	Xmodem 传输协议
	iap_buffer.c / iap_buffer.h	UART 缓冲功能
	iap_crc16.c / iap_crc16.h	Xmodem 和 Flash 检验和的 CRC 运算法则
	iap_ht32fxxxx_it.c / iap_ht32fxxxx_it.h	IAP 中断处理程序
	ht32fxxxx_conf.h	IAP 固件库配置文件
	iap_system_ht32fxxxx.c	IAP 系统配置文件
IAP_Text/ EWARMv5.4/	Project.eww	EWARM 工程文件
	iap_startup_ht32fxxxx.s	EWARM IAP 启动文件
	startup_ht32fxxxx.s	EWARM 应用启动文件
	iap_ht32fxxxx_flash.icf	EWARM IAP 脚本文件
	ht32fxxxx_flash.icf	EWARM 应用脚本文件
IAP_Text/MDK_ARM/	Project.uvmpw	MDK-ARM 多项目工程文件
	iap_startup_ht32fxxxx.s	MDK-ARM IAP 启动文件
	startup_ht32fxxxx.s	MDK-ARM 应用启动文件
	iap.s	包含 IAP 的二进制应用 ASM 文件
	iap.lin	MDK-ARM 链接文件
	iap.ini	调试初始化文件

该范例是用一个多工程处理 IAP 编程和用户应用程序。多工程有助于同时建立,下载和调试 IAP 和应用程序。以下步骤描述了如何通过建立一个多工程建立,下载和调试 IAP 和应用程序。

■ 对于 Keil uVision

建立: 双击“MDK_ARM/Project.uvmpw”打开多项目文件。选择工具栏上的“Batch Build”或者单击“Project → Batch Build”打开“Batch Build”窗口。选择“Build”或者“Rebuild”建立 IAP 和应用程序。该编程工具的输出文件如下所示:

MDK_ARM/Obj/IAP_AP_HT32Fxxxx.hex	(十六进制 IAP 和应用程序文件)
MDK_ARM/Obj/IAP_AP_HT32Fxxxx.axf.bin/AP	(二进制应用程序文件)

下载: 在“Project”窗口右击名为“Project_IAP”的工程,然后点击“Set as Active Project”。按下“LOAD”下载 IAP 和应用程序到 Flash 存储器。

调试: 确定正在运行的工程是“Project_AP”。按“Ctrl+F5”键进入调试模式。“iap.ini”会下载 IAP 的调试符号。由于两者的图像已经下载到 uVision,要跟踪 IAP 和应用程序的转接。

■ 对于 IAR EWARM

建立: 双击“EWARMv5.4/Project.eww”打开多项目文件。按下“F8”或者选择“Project → Batch Build”打开“Batch Build”窗口。选择“Make”或者“Rebuild All”建立 IAP 和应用程序。该编程工具的输出文件如下所示:

EWARMv5.4/HT32Fxxxx/Exe/HT32Fxxxx_IAP_AP.bin	(二进制 IAP 和 应用程序文件)
EWARMv5.4/HT32Fxxxx/Exe/HT32Fxxxx_AP.bin	(二进制应用程序文件)

下载: 点击“Workspace”窗口下拉菜单确定正在运行的工程是“Project_IAP-HT32Fxxxx”,然后选择“Project → Download → Download active application”下载 IAP 和应用程序到 Flash 存储器。

调试: 确定正在运行的工程是“Project_AP - HT32Fxxxx”。然后按下键“Ctrl+D”键进入调试。由于目前的工程设置,只能跟踪到应用程序。

范例程序代码下载到目标板，打开 COM 口应用比如 PC 的 HyperTerminal。如下所示更改 UART 的设置：

- 波特率: 115200 bps
- 数据: 8 位
- 奇偶校验: 无
- 终止位: 1 位
- 流程控制: 无

连接目标板到带有 RS232 口的 PC 机，复位目标板。若 BOOT0 和 BOOT1 都设置为 1，通过 UART，初始化用户应用并弹出一个如下图所示的文本菜单。选择键“1”~“3”执行相关功能。

```
Application
-----
[1] Start IAP mode
[2] Hello World
[3] Enable Write Protection
```

图 27. 应用文本菜单

若 BOOT0=0, BOOT1=1 或者从上面的应用菜单选择键“1”，IAP 模式初始化并弹出一个如下图所示的文本菜单。从屏幕上选择所需的功能。例如，选择键“2”下载图像到 Flash。该范例根据 Xmodem 协议发送无数字符“C”以示已准备好从 PC 机接收数据。开始 Xmodem 传输，选择二进制文件编到应用目标板。

```
Holtek Semiconductor Inc.
IAP Example Code
-----
[1] Execute The New Program
[2] Download Image to Flash
[3] Disable The Write Protection
```

图 28. IAP 文本模式文本菜单

IAP UI 模式范例程序代码

IAP UI 模式连接到某个用户接口，通过 UART 下载用户的图像到主 Flash。只要符合 HT32 Flash 命令协议，该接口可以是基于窗口的 HT32 Flash 编程器或者任何用户应用程序。欲知更多 HT32Flash 命令协议详情，请参考 3.1 章节。IAP UI 模式范例源文件如下表所示。

表 29. IAP UI 模式源文件

路径	文件名	描述
IAP_UI/Src_AP/	main.c	应用主程序
	ht32fxxxx_it.c / ht32fxxxx_it.h	应用中断处理程序
	ht32fxxxx_conf.h	固件库配置文件
	system_ht32fxxxx.c	系统配置文件
IAP_UI/Src_IAP/	iap_main.c	IAP 主程序
	iap_handler.c / iap_handler.h	控制更新过程的 IAP 处理程序
	iap_xmodem.c / iap_xmodem.h	Xmodem 传输协议
	iap_buffer.c / iap_buffer.h	UART 缓冲功能
	iap_crc16.c / iap_crc16.h	Xmodem 和 Flash 检验和的 CRC 运算法则
	iap_ht32fxxxx_it.c / iap_ht32fxxxx_it.h	IAP 中断处理程序
	ht32fxxxx_conf.h	IAP 固件库配置文件
	iap_system_ht32fxxxx.c	IAP 系统配置文件
IAP_UI/EWARMv5.4/	Project.eww	EWARM 工程文件
	iap_startup_ht32fxxxx.s	EWARM IAP 启动
	startup_ht32fxxxx.s	EWARM 应用启动文件
	iap_ht32fxxxx_flash.icf	EWARM IAP 脚本文件
	ht32fxxxx_flash.icf	EWARM 应用脚本文件
IAP_UI/MDK_ARM/	Project.uvmpw	MDK-ARM 多项目工程文件
	iap_startup_ht32fxxxx.s	MDK-ARM IAP 启动文件
	startup_ht32fxxxx.s	MDK-ARM 应用启动文件
	iap.s	包含 IAP 的二进制应用 ASM 文件
	iap.lin	MDK-ARM 链接文件
	iap_ht32fxxxx.ini	调试初始化文件

另外，该范例是用多工程处理 IAP 编程和用户应用程序。一个多工程有助于同时建立，下载和调试 IAP 和应用程序。以下步骤描述了如何通过多工程设置建立，下载和调试 IAP 和应用程序。

■ 对于 Keil uVision

建立: 双击“MDK_ARM/Project.uvmpw”打开多项目文件。选择工具栏上的“Batch Build”或者单击“Project → Batch Build”打开“Batch Build”窗口。选择目标板上的图标文件或者点击窗口。点击“Project_AP”和“Project_IAP”中所需的器件例如名为“HT32Fxxxx”。选择“Build”或者“Rebuild”建立 IAP 和应用程序。该编程工具的输出文件如下所示:

MDK_ARM/Obj/IAP_AP_HT32Fxxxx.hex (十六进制 IAP 和应用程序文件)
MDK_ARM/Obj/IAP_AP_HT32Fxxxx.axf.bin/AP (二进制应用文件)

下载: 在“Project”窗口右击名为“Project_IAP”的工程, 然后点击“Set as Active Project”。按下“LOAD”下载 IAP 和应用程序到 Flash 存储器。

调试: 确定正在运行的工程是“Project_AP”。按“Ctrl+F5”键进入调试模式。“iap.ini”会下载 IAP 的调试符号。由于两者的图像已经下载到 uVision, 要跟踪 IAP 和应用程序的转接。

■ 对于 IAR EWARM

建立: 双击“EWARMv5.4/Project.eww”打开多项目文件。按下“F8”或者选择“Project → Batch Build”打开“Batch Build”窗口。选择“Make”或者“Rebuild All”建立 IAP 和应用程序。该编程工具的输出文件如下所示:

EWARMv5.4/HT32Fxxxx/Exe/HT32Fxxxx_AP.bin (二进制 IAP 和 应用文件)
EWARMv5.4/HT32Fxxxx/Exe/HT32Fxxxx_AP.bin (二进制应用程序文件)

下载: 点击“Workspace”窗口的下拉菜单确定正在运行的工程是“Project_IAP-HT32Fxxxx”, 然后选择“Project → Download → Download active application”下载 IAP 和应用程序到 Flash 存储器。

调试: 确定正在运行的工程是“Project_AP - HT32Fxxxx”。然后按下键“Ctrl+D”键进入调试模式。

和 IAP 文本模式一样, 可改变启动设定来启动应用程序或者 IAP 模式。设计 IAP UI 模式连接到 HT32 Flash 编程器或者用户的某个与 HT32 Flash 命令协议兼容的 Flash 编程器。如何使用 HT32 Flash 编程器请参考相关篇章。

更新 IAP 固件

对于嵌入式 Flash 存储系统，擦除器件或者编程不能在同一页应用。由于 IAP 范例程序代码是在主 Flash 的起始地址，所以通过一般的 IAP 下载，它是无法自动更新的。在这种情况下，如果硬要擦除器件或者自动编程，IAP 固件将会消失。为了编程或者更新 IAP 固件，需要额外的编程方案比如 ISP, USB 调试器或者 Flash 编程器。欲知更多 USB 调试器或者 Flash 编程器的详情请参考下列文件。

- 快速启动手册: HoltekHT32 系列单片机的 Keil MDK-ARM 快速启动
- 快速启动手册: HoltekHT32 系列单片机的 IAR EWARM 快速启动
- 32 位单片机用户手册 Holtek 编程环境 HOPE3000

另外，范例程序，“IAP_Text_RAM”会在提供有自动更新 IAP 固件的代替方案的 RAM 中执行。该程序利用链接脚本详细描述下载视图和执行图像中 RO 代码的视图。RO 代码是位于主 Flash 的起始位置，在 IAP 程序开始之前通过 C 库复制到 RAM。因为在执行期间 IAP 程序是位于 RAM，所以它可以在主 Flash 中擦除和再烧入整个 IAP 图像。注意，在下载过程中，目标板不能掉电或者复位，否则下载会失败，目标板将无法再从主 Flash 启动。在上述情形下，必须使用 ISP, ICP 或者 Flash 编程器下载 IAP 图像。

请参考“IAP_Text”部分建立和下载“IAP_Text_RAM”范例程序代码到目标板。如下图所示，范例程序将通过 UART 显示一个文本菜单。例如，按下“3”键将启动 Xmodem 和更新 IAP 固件

```
Holtek Semiconductor Inc.  
IAP Example Code  
-----  
[1] Execute The New Program  
[2] Download Image to Flash  
[3] Update IAP Firmware  
[4] Disable The Write Protection
```

图 29. RAM 执行 IAP 文本模式文本菜单

盛群半导体股份有限公司(总公司)
新竹市科学工业园区研新二路 3 号
电话 : 886-3-563-1999
传真 : 886-3-563-1189
网站 : www.holtek.com.tw

盛群半导体股份有限公司(台北业务处)
台北市南港区园区街 3 之 2 号 4 楼之 2
电话 : 886-2-2655-7070
传真 : 886-2-2655-7373
传真 : 886-2-2655-7383 (International sales hotline)

盛扬半导体有限公司(深圳业务处)
深圳市南山区科技园科技中三路与高新中二道交汇处生产力大楼 A 单元五楼 518057
电话 : 0755-8616-9908,8616-9308
传真 : 0755-8616-9722

Holtek Semiconductor(USA), Inc. (北美业务处)
46712 Fremont Blvd., Fremont, CA 94538
电话 : 510-252-9880

传真 : 510-252-9885
网站 : www.holtek.com

Copyright © 2011 by HOLTEK SEMICONDUCTOR INC.

使用指南中所出现的信息在出版当时相信是正确的, 然而盛群对于说明书的使用不负任何责任。文中提到的应用目的仅仅是用来做说明, 盛群不保证或表示这些没有进一步修改的应用将是适当的, 也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。盛群产品不授权使用于救生、维生器件或系统中做为关键器件。盛群拥有不事先通知而修改产品的权利, 对于最新的信息, 请参考我们的网址 <http://www.holtek.com.tw>。