



## STM32F2x7 通过以太网实现在应用中编程（IAP）

---

### 前言

本应用笔记的目标读者为使用 STM32F2x7 微控制器的开发者。它提供了如何使用 STM32F2x7 以太网接口实现在应用中编程（IAP）的解决方案。

在 LwIP TCP/IP 协议栈上提供了两种可行的解决方案：

- 使用 TFTP（简单文件传输协议）的 IAP
- 使用 HTTP（超文本传输协议）的 IAP

# 目录

<b>1</b>	<b>IAP 概述</b> .....	<b>5</b>
1.1	工作原理 .....	5
1.2	使用 MCU 以太网接口实现 IAP .....	5
1.3	在 STM32F2x7 上通过以太网实现 IAP .....	6
1.3.1	使用 TFTP 实现 IAP 的方法 .....	6
1.3.2	使用 HTTP 实现 IAP 的方法 .....	6
<b>2</b>	<b>使用 TFTP 实现 IAP</b> .....	<b>7</b>
2.1	TFTP 概述 .....	7
2.2	在 STM32F2x7 上使用 TFTP 实现 IAP .....	7
2.3	使用软件 .....	9
<b>3</b>	<b>使用 HTTP 实现 IAP</b> .....	<b>10</b>
3.1	HTTP 文件上传概述 .....	10
3.2	在 STM32F2x7 上使用 HTTP 实现 IAP .....	12
3.3	使用软件 .....	14
3.4	已知限制 .....	14
3.4.1	添加到二进制文件的额外字节 .....	14
<b>4</b>	<b>环境</b> .....	<b>15</b>
4.1	MAC 和 IP 地址设置 .....	15
4.2	STM32xG-EVAL 板上的跳线设置 .....	15
4.3	软件文件结构 .....	16
4.4	代码量测量 .....	16
4.5	构建 IAP 映像 .....	17
<b>5</b>	<b>版本历史</b> .....	<b>18</b>

## 表格索引

表 1.	TFTP 操作码 .....	7
表 2.	跳线配置 .....	15
表 3.	文件组织 .....	16
表 4.	代码量 vs 配置选项 .....	16
表 5.	文档修订历史 .....	18

# 图片索引

图 1.	IAP 操作流程.....	5
图 2.	TFTP 包.....	7
图 3.	使用 TFTP 实现 IAP 的流程图.....	8
图 4.	TFTPD32 对话框.....	9
图 5.	文件上传 HTML 格式的浏览器视图.....	10
图 6.	IE8 HTTP 头格式.....	11
图 7.	Mozilla Firefox HTTP 头格式.....	11
图 8.	登录网页.....	12
图 9.	文件上传完成页面.....	12
图 10.	使用 HTTP 实现 IAP 的流程图.....	13

# 1 IAP 概述

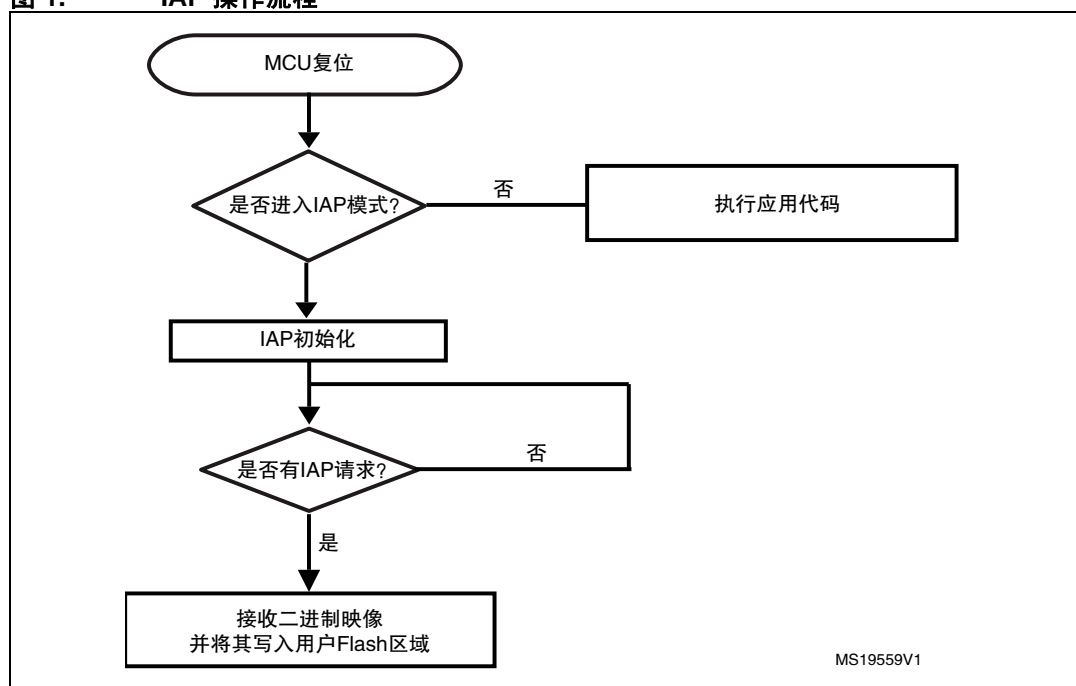
## 1.1 工作原理

在应用中编程（IAP）是一种在现场通过 MCU 的通信接口（例如 UART,USB,CAN 和以太网等）进行固件升级的方式。

当启动微控制器时，您可以选择让它进入 IAP 模式以执行 IAP 代码，或者进入正常模式来开始执行应用代码。IAP 代码和应用程序代码都在微控制器的嵌入式 Flash 中。通常 IAP 代码被存储在 MCU Flash 的前几页，用户应用代码占据剩余的 Flash 区域。

图 1 显示了 IAP 操作流程：

图 1. IAP 操作流程



## 1.2 使用 MCU 以太网接口实现 IAP

当以太网可用时，它通常是在嵌入式应用中实现 IAP 功能的更好的接口。其优点是：

- 高速通信接口（10/100 Mbit/s）
- 通过网络（LAN 或者 WAN）进行远程编程
- TCP/IP 协议栈上的标准应用协议（例如 FTP、TFTP、HTTP）可用于实现 IAP

## 1.3 在 STM32F2x7 上通过以太网实现 IAP

本应用笔记说明了两种使用以太网通信外在 STM32F2x7 上实现 IAP 的解决方案：

- 使用 TFTP（简单文件传输协议）的 IAP
- 使用 HTTP（超文本传输协议）的 IAP

两种解决方案都运行在 LwIP 协议栈 (v1.3.2) 上，它是 TCP/IP 协议套件的一种轻量级实现。

### 1.3.1 使用 TFTP 实现 IAP 的方法

使用 TFTP 实现 IAP 的方法广泛应用于需要固件升级能力的嵌入式应用（比如在嵌入式 Linux bootloader 中）。

TFTP 是工作在 UDP 传输层上的简单文件传输协议。它主要应用在 LAN 环境中。它基于客户端 / 服务器体系结构，其中客户端向文件服务器请求文件传输（读或写操作）。

在这个例子中，服务器只处理来自 PC TFTP 客户端的写请求，所以在 LwIP 栈上实现了一个简单的 TFTP 服务器。

### 1.3.2 使用 HTTP 实现 IAP 的方法

使用 HTTP 协议进行固件升级没有使用 TFTP 常见，但是当需要通过互联网进行远程编程时，它是一种有用的解决方案。在这种情况下，需要 TCP 传输协议来确保最优操作。

工作在 TCP 上的 HTTP，提供了一种用 HTML 形式从 web 客户端（Mozilla Firefox 或 Microsoft Internet Explorer）发送二进制文件的方法，称作 HTTP 文件上传（RFC 1867）。

本文档的下一部分给出了关于这两种 IAP 方法实现的详细信息并解释了如何使用这个软件。

## 2 使用 TFTP 实现 IAP

### 2.1 TFTP 概述

TFTP 是工作在 UDP 传输层上的简单文件传输协议。文件传输是从 TFTP 客户端发起的，TFTP 客户端向 TFTP 服务器发送读或写请求。当服务器响应该请求后，文件数据传输开始。数据按固定大小的数据块发送（如 512 字节的数据块）。

必须在每个传输的数据块被接收端响应后，才能发送下一个数据块。回应机制通过随每个数据块同时发送的数据块编号来实现。数据块小于固定块大小表示文件传输结束。

图 2 说明了各类 TFTP 包的格式：

图 2. TFTP 包

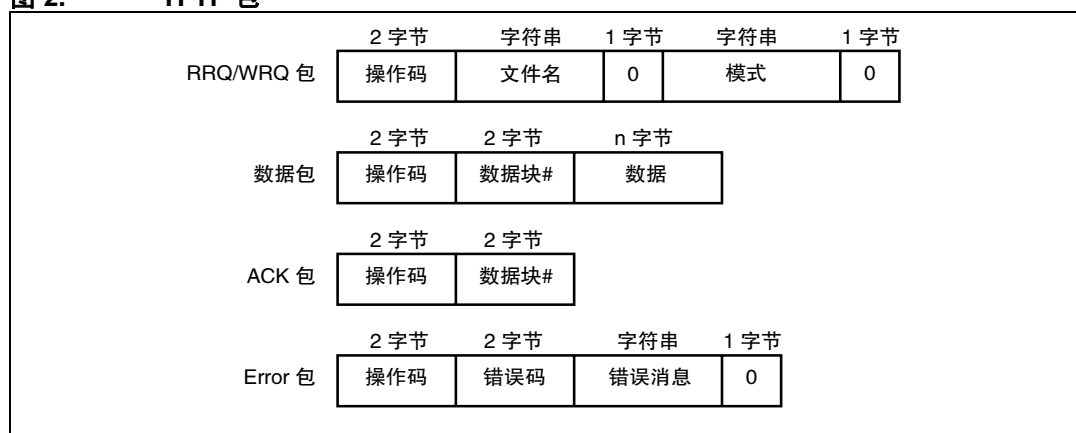


表 1 列出了 TFTP 操作码。

表 1. TFTP 操作码

操作码	操作
0x1	读请求 (RRQ)
0x2	写请求 (WRQ)
0x3	数据
0x4	响应 (ACK)
0x5	错误

### 2.2 在 STM32F2x7 上使用 TFTP 实现 IAP

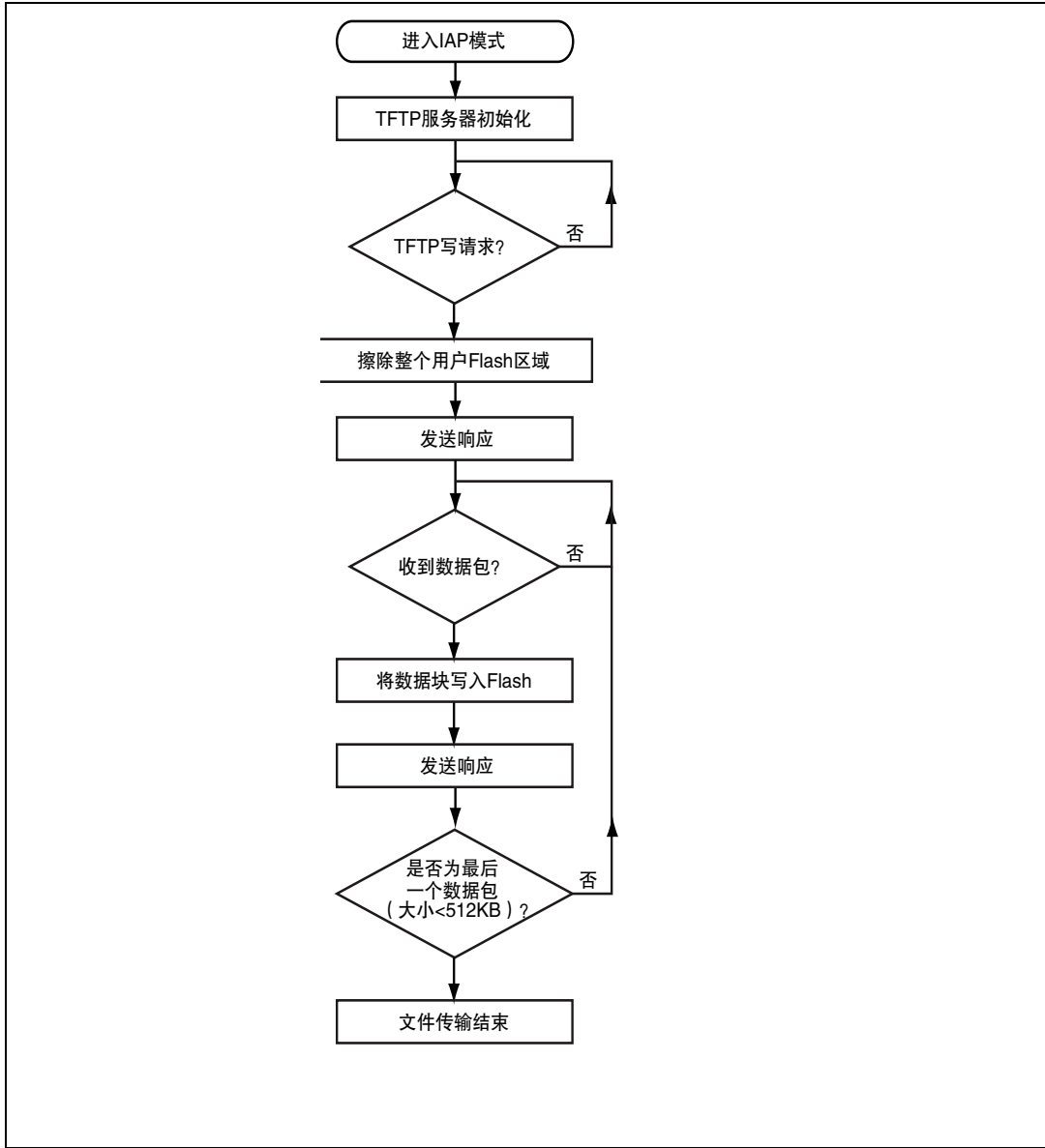
该 IAP 实现包括一个 LwIP TCP/IP 栈上的 TFTP 服务器。该服务器响应从远程 TFTP 客户端 (PC) 接收的 WRITE 请求。TFTP READ 请求被忽略。

服务器将接收到的数据块写入 MCU Flash（在用户 Flash 区域），而不是将接收的文件写入文件系统。

注： 在本实现中，数据块大小固定为 512 字节。

图 3 给出了使用 TFTP 实现 IAP 操作的过程。

图 3. 使用 TFTP 实现 IAP 的流程图





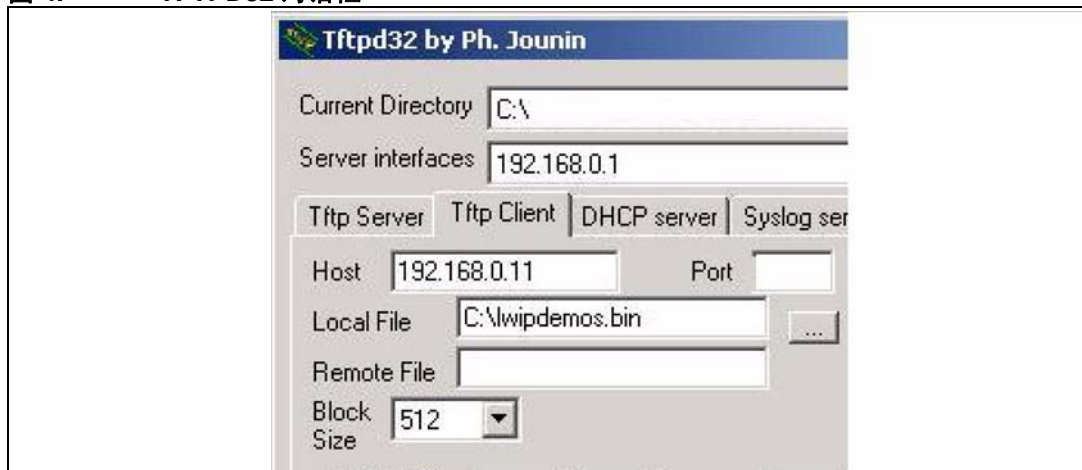
## 2.3 使用软件

为了通过 TFTP 对 IAP 进行测试，遵循如下步骤：

1. 在 STM322xG-EVAL 板上确保正确的跳线设置（参见表 2）。
2. 在 main.h 文件中，取消注释选项“USE\_IAP\_TFTP”。根据需求，您还可以取消注释 / 注释其它选项，比如“USE\_DHCP”或“USE\_LCD”。
3. 重新编译软件。使用生成的 map 文件，确保 IAP 代码区域（从地址 0x0 开始）和开始于 USER\_FLASH\_FIRST\_PAGE\_ADDRESS（在 main.h 中定义）的用户 Flash 区域之间没有重叠。
4. 将程序下载到 STM32 Flash 中运行它。
5. 按住 Key 按钮的同时，按下并释放 Reset 键，进入 IAP 模式。
6. 若在 main.h 文件中定义了“USE\_LCD”，LCD 屏幕上会显示消息，指示已经进入 IAP 模式。如果采用了 DHCP（main.h 文件中定义了 USE\_DHCP），LCD 屏幕上也会显示一条消息，指示 DHCP IP 地址分配的成功或失败。
7. 分配完 IP 地址后（动态或静态地址），用户可启动 IAP 进程。
8. 在 PC 上，打开 TFTP 客户端（例如，TFTPD32）并配置 TFTP 服务器地址（TFTPD32 中的主机地址）。
9. 选择一个二进制映像下载到 STM32 Flash 中（/project/binary 文件夹中提供了两个二进制映像作为例子）。
10. 在 TFTPD32 应用程序上点击“Put”按钮，启动一个文件写请求。
11. 如果定义了 USE\_LCD，IAP 操作的进度会显示在 LCD 上。
12. 在 IAP 操作结束时，可以复位评估板来运行您刚才在 STM32 Flash 中编写的程序。

**注：** 在 USE\_LCD 使能的情况下，如果出现连接问题，则 LCD 屏幕上会显示错误消息，指示连接失败。

**图 4. TFTPD32 对话框**



### 3 使用 HTTP 实现 IAP

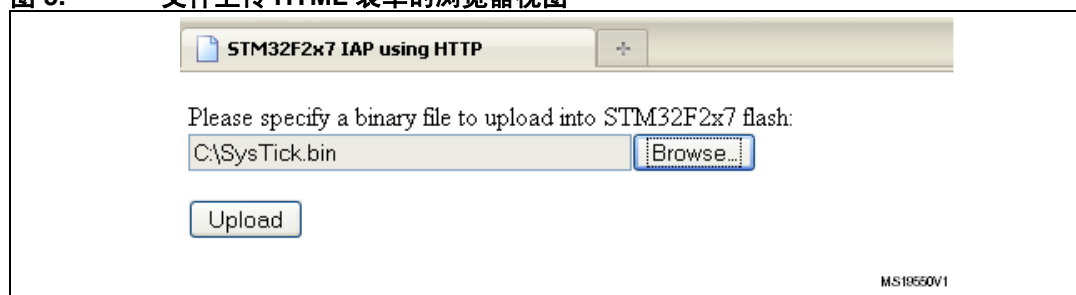
#### 3.1 HTTP 文件上传概述

RFC1867 中定义了使用 HTTP 进行文件上传。该上传文件方法基于 HTML 表单。通常用 HTML POST 方法代替 GET 来发送原始二进制数据。

下面是一个实现基于表单的文件上传的 HTML 代码示例：

```
<form action ="/upload.cgi" enctype="multipart/form-data" method="post">  
  <p>Please specify a binary file to upload into STM32F2x7 flash:  
  <br>  
  <input type="file" name="datafile" size="40">  
</p>  
<div>  
  <input type="submit" value="Upload">  
</div></form>
```

图 5. 文件上传 HTML 表单的浏览器视图



浏览选择一个二进制文件上传，然后按下上传按钮发送它。

根据文件大小，数据以连续的 TCP 段形式发送给 Web 服务器。

注：发送文件数据前，Web 客户端发送包括文件名、内容长度等信息的 HTTP 头数据，其中一些信息必须由 Web 服务器进行解析。

Web 客户端并不总是拥有相同的 HTTP 头格式。图 6 显示了 POST 请求的 Internet Explorer HTTP 头格式。图 7 显示了 Mozilla Firefox HTTP 头格式。

http Web 服务器必须能够处理这些不同格式。

图 6. IE8 HTTP 头格式

```

Hypertext Transfer Protocol
  POST /upload.cgi HTTP/1.1\r\n
  Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, appl
  Referer: http://192.168.0.12/checklogin.cgi\r\n
  Accept-Language: en-us\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; Trident/4.0; .NET CLR 1.1.432;
  Content-Type: multipart/form-data; boundary=-----7db28061402a2\r\n
  Accept-Encoding: gzip, deflate\r\n
  Host: 192.168.0.12\r\n
  Content-Length: 1965\r\n
  [Content length: 1965]
  Connection: Keep-Alive\r\n
  Cache-Control: no-cache\r\n
  \r\n
MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----
[Type: multipart/form-data]
First boundary: -----7db28061402a2\r\n
  Encapsulated multipart part: (application/octet-stream)
    Content-Disposition: form-data; name="datafile"; filename="STM322xG_EVAL_SysTick.bin"\r\n
    Content-Type: application/octet-stream\r\n\r\n
  Media Type
Last boundary: \r\n-----7db28061402a2--\r\n

```

图 7. Mozilla Firefox HTTP 头格式

```

Hypertext Transfer Protocol
  POST /upload.cgi HTTP/1.1\r\n
  Host: 192.168.0.13\r\n
  User-Agent: Mozilla/5.0 (windows; U; windows NT 5.1; en-US; rv:1.9.2.3) Gecko/20100401 Firef
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
  Accept-Language: en-us,en;q=0.5\r\n
  Accept-Encoding: gzip,deflate\r\n
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
  Keep-Alive: 115\r\n
  Connection: keep-alive\r\n
  Referer: http://192.168.0.13/checklogin.cgi\r\n
  Content-Type: multipart/form-data; boundary=-----114782935826962\r\n
  Content-Length: 1969\r\n
  [Content length: 1969]
  \r\n
MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----
[Type: multipart/form-data]
First boundary: -----114782935826962\r\n
  Encapsulated multipart part: (application/octet-stream)
    Content-Disposition: form-data; name="datafile"; filename="STM322xG_EVAL_SysTick.bin"\r\n
    Content-Type: application/octet-stream\r\n\r\n
  Media Type
Last boundary: \r\n-----114782935826962--\r\n

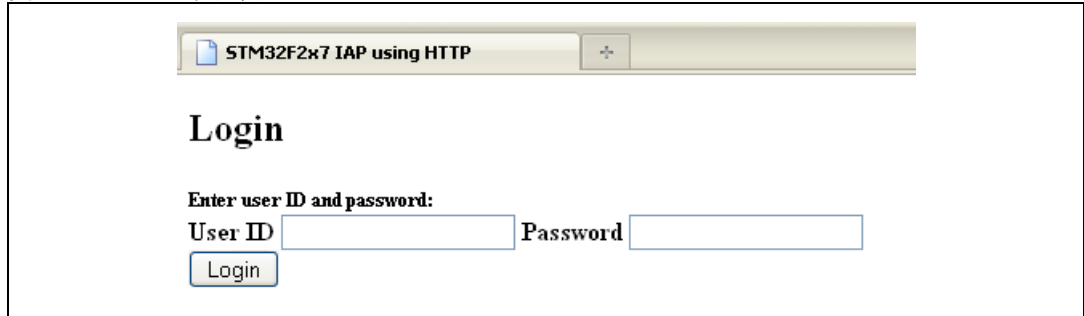
```

### 3.2 在 STM32F2x7 上使用 HTTP 实现 IAP

该 IAP 实现包括一个基于 LwIP 栈的 HTTP Web 服务器。

当在浏览器上输入 STM32 IP 地址时，将显示登录网页如 [图 8](#) 所示。此登录网页限制只有授权用户可以进行 IAP 文件上传。

**图 8. 登录网页**



输入正确的用户 ID 和密码（在 main.h 文件中预定义的）并点击登录按钮。然后载入一个文件上传页面（参见 [图 5](#)）。

- 注：
- 1 默认用户 ID 为：“user”，密码为“stm32”。
  - 2 如果用户 ID 或密码不正确，登录网页将会重新加载。

成功登录后，浏览选择要加载进 STM32 Flash 的二进制文件。

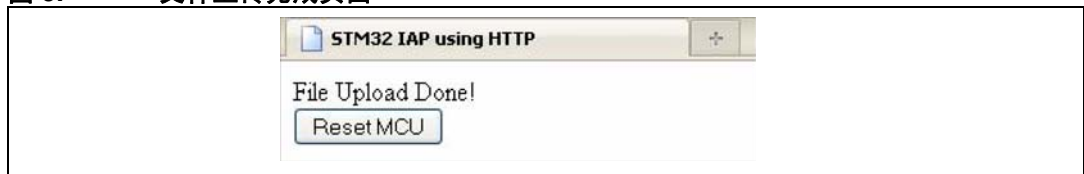
注： 确保该二进制文件的大小不超出 STM32 用户 Flash 区域的总大小。

点击上传按钮（参见 [图 5](#)）时，就向服务器发送了一个 POST 请求。此时服务器开始擦除所有的用户 Flash 区域并等待二进制文件的原始数据。然后接收到的数据将会被写入用户 Flash 区域。

请注意，要接收的数据总长度是从传输开始时发送的 HTTP 头数据中提取出来的。

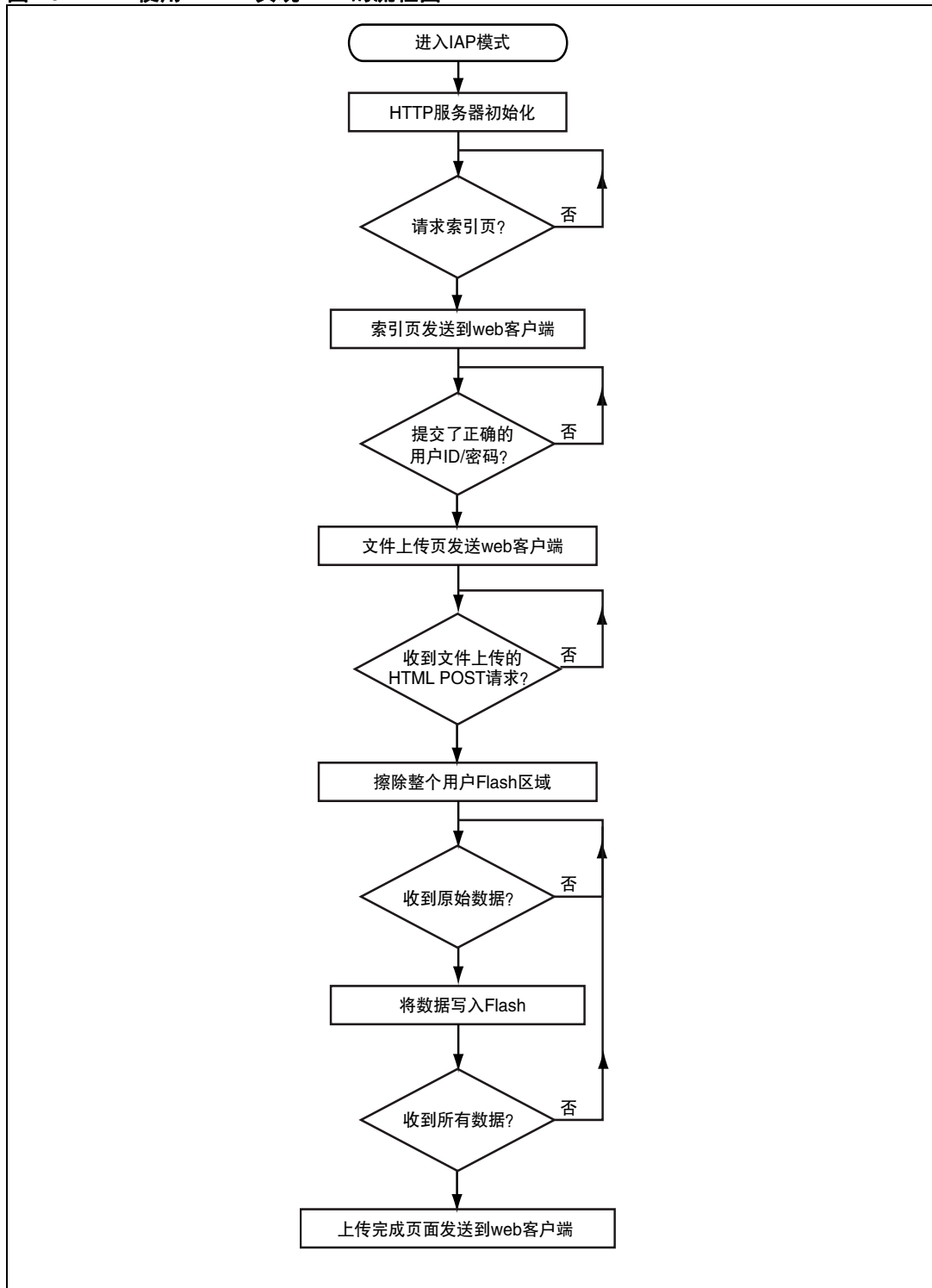
在 IAP 操作结束时，一个网页指示 IAP 操作成功，网页上显示了一个可以复位 MCU 的按钮。

**图 9. 文件上传完成页面**



[图 10](#) 总结了使用 HTTP 实现 IAP 的方法。

图 10. 使用 HTTP 实现 IAP 的流程图



## 3.3 使用软件

为了通过 TFTP 对 IAP 进行测试，请遵循如下步骤

1. 在 STM322xG-EVAL 板上确保正确的跳线设置（参见表 2）。
2. 在 main.h 文件中，取消选项“USE\_IAP\_HTTP”的注释，并且根据需求您还可以取消注释 / 注释其它选项，比如“USE\_DHCP”或“USE\_LCD”。
3. 重新编译软件。使用生成的 map 文件，确保 IAP 代码区域（从地址 0x0 开始）和开始于 USER\_FLASH\_FIRST\_PAGE\_ADDRESS（在 main.h 中定义）的用户 Flash 区域之间没有重叠。
4. 将程序下载到 STM32Flash 中并运行它。
5. 按住 Key 按钮的同时，按下并释放 Reset 键，进入 IAP 模式。
6. 若在 main.h 文件中定义了“USE\_LCD”，LCD 屏幕上会显示消息，指示已经进入 IAP 模式。同样在使用 DHCP（main.h 文件中定义了 USE\_DHCP）的情况下，LCD 屏幕上也会显示一条消息，指示 DHCP IP 地址分配的成功或失败。
7. 分配完 IP 地址后（动态或静态地址），用户可启动 IAP 进程。
8. 打开一个 web 客户端（Mozilla Firefox 或 Microsoft Internet Explorer）并输入 STM32 IP 地址。
9. 将会显示一个登录网页。在 UserID 字段中输入“user”并在 Password 字段中输入“stm32”，然后按 Login 按钮。
10. 然后会加载 fileupload.html 网页。选择一个要加载进 STM32 Flash 的二进制映像，然后按 Upload 按钮开始 IAP 进程。
11. 如果定义了 USE\_LCD，IAP 操作的进度会显示在 LCD 上。
12. 在 IAP 操作结束时，会加载一个新的网页，指示文件上传操作成功。
13. 可以按“RESET MCU”按钮复位 MCU 来运行您刚才在 STM32 Flash 中编写的程序。

- 注：
- 1 在 USE\_LCD 使能的情况下，如果出现连接问题，则 LCD 屏幕上会显示错误消息，指示连接失败。
  - 2 该软件已用下列 Web 客户端测试：Microsoft Internet Explorer 8 和 Mozilla Firefox 3.6。

## 3.4 已知限制

### 3.4.1 添加到二进制文件的额外字节

网络浏览器（Microsoft Internet Explorer 或 Mozilla Firefox）会将一个随机边界标签（根据 RFC 1521，不长于 72 字节）添加到上传的二进制文件末尾。在当前的 IAP 软件版本中，如果有足够的空间，该边界标签不会被移除而是存储在 Flash 中。如果空间不足，额外字节不写入 Flash 中，并不返回错误。

## 4 环境

### 4.1 MAC 和 IP 地址设置

*main.h* 文件中定义了 MAC 和 IP 地址。

默认 MAC 地址固定为：00:00:00:00:00:02。

IP 可以被设置为静态地址，也可以设置为由 DHCP 服务器分配的动态地址。默认静态 IP 地址设为：192.168.0.10

您可以通过在 *main.h* 文件中使能 USE\_DHCP 来选择 DHCP 模式。

请注意，如果选择通过 DHCP 配置 IP 地址，但应用程序无法在它已经连接到的网络上发现 DHCP 服务器，则 IP 地址会自动设为静态地址（192.168.0.10）。

### 4.2 STM322xG-EVAL 板上的跳线设置

若需运行软件，在 STM322xG-EVAL 板。

在 project\inc 文件夹下的 *main.h* 文件中选择 MII 或 RMII 配置。

例如，选择 RMII 模式：

```
//#define MII_MODE
#define RMII_MODE
```

对于 MII 模式，PHY 时钟取自外部晶振；如果 *main.h* 文件中定义了 MII\_MODE 和 PHY\_CLOCK\_MCO，则时钟由 STM32 经由 MCO 管脚提供。

- 注：
- 1 在 RMII 模式中，如 STM32F20x & STM32F21x 勘误手册（ES0005）第 2.6.5 节所述，PLL 限制导致无法使用 MCO 将 50 MHz 时钟输出到 PHY。在这种情况下，需要从外部提供 50MHz 时钟，在评估板 CN3 下方的 U3 处，焊接一个 50 MHz 振荡器（参考 SM7745HEV-50.0M 或同类器件）并且将 JP5 跳线移除。该振荡器不随板提供。若需更详细信息，请参见 STM3220G-EVAL 评估板用户手册 UM1057。
  - 2 在本文档中，“STM322xG-EVAL 板”指的是 STM3220G-EVAL 和 STM3221G-EVAL 板。

表 2. 跳线配置

跳线编号	MI I 模式配置	RMII 模式配置
JP5	1-2: 由外部晶振提供 25 MHz 时钟 2-3: 由 PA8 处脚（MCO）提供的 25 MHz 时钟	不适用
JP6	2-3: 启用 MII 接口模式	1-2: 启用 RMII 接口模式
JP8	不连接: MII 接口模式被选中	连接: RMII 接口模式被选中

### 4.3 软件文件结构

表 3 介绍了项目源文件：

表 3. 文件结构

文件名	说明
main.c	主应用文件
main.h	主配置文件
httpserver.c/.h	HTTP 服务器实现
tftpserver.c/.h	TFTP 服务器实现
flash_if.c/.h	高级闪存访问函数
netconf.c/.h	高级以太网接口函数
stm32f2x7_eth_bsp.c/.h	STM32F2x7 以太网硬件配置
stm32f2xx_it.c/.h	中断处理程序
fsdata.c	HTML 文件作为 ROM 文件系统
lwipopts.h	LwIP 配置选项

注：表中未列出所用的标准固件库和 LwIP 栈的文件。

### 4.4 代码量测量

表 4 给出了根据 main.h 文件中不同的配置选项进行的代码量测量。

表 4. 代码量 vs 配置选项

代码量（字节）	USE_IAP_TFTP	USE_IAP_HTTP	USE_LCD	USE_DHCP
13752	X			
21736	X		X	
27264	X		X	X
25408		X		
32512		X	X	
38440		X	X	X
39928	X	X	X	X

注：该软件使用 IAR EWARM v6.10 编译，对代码量进行了高度优化。



## 4.5 构建 IAP 映像

为了构建 IAP 映像（将用 IAP 软件进行加载），请确保：

1. 被编译 / 链接的软件必须从用户 Flash 区域起始地址开始运行（该地址应与 main.h 文件中的 USER\_FLASH\_FIRST\_PAGE\_ADDRESS 定义的地址相同）。
2. 向量表起始地址配置为用户 Flash 区域的起始地址。

向量表起始地址偏移有两种配置方法：

- a) 在应用代码中，使用 misc.h/c 驱动中的“NVIC\_SetVectorTable”函数将向量表重新定位到应用程序的起始地址。

例如，将向量表起始地址位置设置为 0x08010000:

```
NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x10000);
```

- b) 通过修改 system\_stm32f2xx.c 文件中定义的常量“VECT\_TAB\_OFFSET”。

例如，将向量表起始地址位置设置为 0x08010000:

```
#define VECT_TAB_OFFSET 0x10000
```

3. 编译后的软件大小不超出总的用户 Flash 区域。

*使用 STM32F10xxx 的 USART 实现在应用中编程*（AN3374）中的软件示例，提供了一个预先配置的项目可以创建用 IAP 加载的应用程序。

## 5 版本历史

表 5. 文档版本历史

日期	版本	变更
2011 年 5 月 27 日	1	初始版本。
2011 年 10 月 19 日	2	更新了 <a href="#">第 4.2 章节: STM322xG-EVAL 板上的跳线设置</a> 和 <a href="#">表 2: 跳线配置</a> 。

**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。

© 2015 STMicroelectronics - 保留所有权利