



UM1709

사용자 매뉴얼

## STM32Cube 이더넷 IAP 예

### 소개

STM32Cube™ 이니셔티브는 STMicroelectronics가 개발 노력, 시간 및 비용을 줄여 개발자의 삶을 편리하게 하기 위해 시작되었습니다. STM32Cube는 STM32 포트폴리오를 포괄합니다.

STM32Cube 버전 1.x에는 다음이 포함됩니다.

- STM32CubeMX는 다음을 생성할 수 있는 그래픽 소프트웨어 구성 도구입니다.  
그래픽 마법사를 사용한 C 초기화 코드.
- 시리즈별로 제공되는 포괄적인 임베디드 소프트웨어 플랫폼(예:  
STM32F4 시리즈용 STM32CubeF4)
  - STM32 추상화 계층 임베디드 소프트웨어인 STM32Cube HAL은 다음을 보장합니다.  
STM32 포트폴리오 전반에 걸쳐 이식성 극대화
  - RTOS, USB, STMTouch와 같은 일관된 미들웨어 구성 요소 세트  
FatFS 및 그래픽
  - 모든 임베디드 소프트웨어 유틸리티에는 전체 예제 세트가 제공됩니다.

IAP(In-Application 프로그래밍)은 동일한 플래시에서 코드를 실행하는 동안 플래시 메모리를 프로그래밍하는 방법입니다. 고속 통신 프로토콜을 사용하여 애플리케이션 코드를 로드할 수 있는 가능성을 제공합니다.

이 사용 설명서는 STM32 마이크로컨트롤러에서 STM32Cube 펌웨어를 사용하는 개발자를 위해 작성되었습니다. 이더넷 통신을 사용하여 IAP(In-Application 프로그래밍)을 구현하는 방법에 대한 전체 설명을 제공합니다.

LwIP TCP/IP 스택 위에 두 가지 가능한 솔루션이 제공됩니다.

- TFTP(Trivial File Transfer Protocol)를 사용한 IAP
- HTTP(Hypertext Transfer Protocol)를 사용한 IAP

메모:

이 문서는 이더넷 주변 장치를 갖춘 모든 STM32 시리즈에 적용됩니다. 그러나 단순화를 위해 STM32F4xx 마이크로컨트롤러와 STM32CubeF4가 참조 플랫폼으로 사용됩니다.

STM32F107xx, STM32F2x7xx 및 STM32F7xx와 같이 이더넷 연결을 제공하는 다른 시리즈에도 동일한 설명, 파일 이름 및 스크린샷이 적용됩니다. STM32 시리즈의 이더넷 IAP 예시 구현에 대해 자세히 알아보려면 관련 STM32Cube 펌웨어 패키지에 제공된 설명서를 참조하세요.



## 내용물

1	IAP 개요 .....	5
1.1	작동 이론. ....	5
1.2	MCU 이더넷 인터페이스를 사용하는 IAP. ....	6
1.3	이더넷을 통한 IAP 구현. ....	6
1.3.1	TFTP를 이용한 IAP 방식 ..	6
1.3.2	HTTP를 사용하는 IAP 방식입니다. ..	6
2	TFTP를 사용하는 IAP.....	7
2.1	TFTP 개요. ....	7
2.2	STM32F4xx용 TFTP를 사용하여 IAP 구현... ..	8
2.3	펌웨어 사용. ....	9
삼	HTTP를 사용하는 IAP.....	10
3.1	HTTP 파일 업로드 개요. ....	10
3.2	STM32F4xx용 HTTP를 사용하여 IAP 구현.. ..	10
3.3	펌웨어 사용. ....	13
4	환경 .....	14
4.1	애플리케이션 설정. ....	14
4.1.1	PHY 인터페이스 구성. ....	14
4.1.2	MAC 및 IP 주소 설정. ....	14
4.2	평가 보드 설정. ....	14
4.3	펌웨어 파일 구성. ....	14
4.4	IAP용 이미지 구축. ....	15
5	결론 .....	16
부록 A	FAQ.....	17
A.1	정적 또는 동적(DHCP) IP 주소 할당 중에서 선택하는 방법	17
A.2	이더넷 케이블 연결이 끊어졌을 때 애플리케이션이 어떻게 작동하는지..	17
A.3	다른 하드웨어에 애플리케이션을 이식하는 방법. ....	17
6	개정 내역 .....	18

테이블 목록

표 1. TFTP 연산 코드 . . . . . 7

표 2. 파일 구성 . . . . . 14

표 3. 문서 개정 내역 . . . . . 18

# 그림 목록

그림 1. IAP 작업 흐름 .	..5
그림 2. TFTP 패킷.	..7
그림 3. TFTP를 이용한 IAP 흐름도 .	..8
그림 4. TFTP32 대화 상자. .	..9
그림 5. 파일 업로드 HTML 양식의 브라우저 보기.	..10
그림 6. 로그인 웹 페이지.	..11
그림 7. 파일 업로드가 완료되었습니다.	..11
그림 8. HTTP를 사용하는 IAP의 흐름도입니다. .	..12

## 1 IAP 개요

### 1.1 작동 이론

IAP(In-Application 프로그래밍)는 UART, USB, CAN 및 이더넷과 같은 MCU 통신 인터페이스를 사용하여 현장에서 펌웨어를 업그레이드하는 수단입니다.

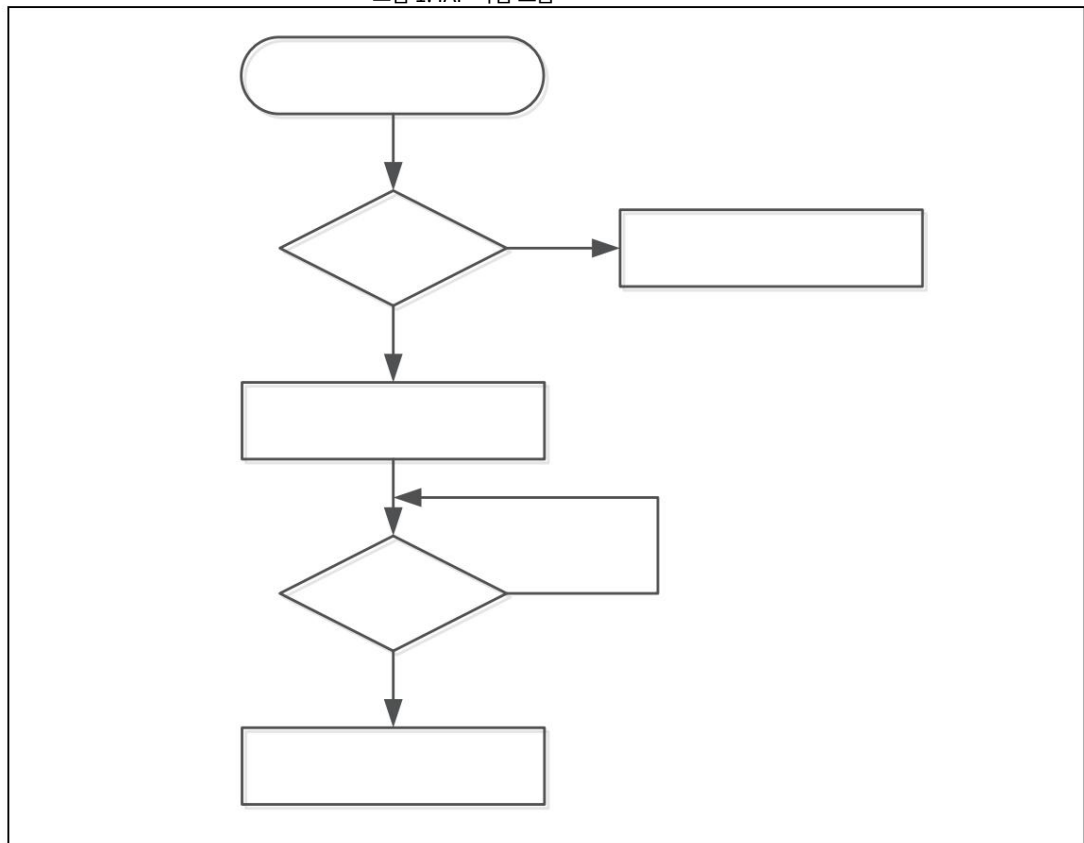
마이크로컨트롤러를 부팅할 때 다음 중 하나에 배치하도록 선택할 수 있습니다.

- IAP 코드를 실행하기 위한 IAP 모드.
- 응용 프로그램 코드를 실행하기 위한 일반 모드입니다.

IAP 코드와 애플리케이션 코드는 모두 마이크로컨트롤러의 내장 플래시 메모리에 있습니다. IAP 코드는 일반적으로 MCU 플래시의 첫 번째 페이지에 저장되며 사용자 애플리케이션 코드는 나머지 플래시 영역을 차지합니다.

그림 1은 IAP 작업 흐름을 보여줍니다.

그림 1. IAP 작업 흐름



## 1.2 MCU 이더넷 인터페이스를 사용하는 IAP

사용 가능한 경우 이더넷은 임베디드 애플리케이션에서 IAP 기능을 구현하기 위해 선호되는 인터페이스인 경우가 많습니다. 장점은 다음과 같습니다.

- 고속 통신 인터페이스(10/100Mbit/s)
- 네트워크(LAN 또는 WAN)를 통한 원격 프로그래밍
- TCP/IP 위에 FTP, TFTP, HTTP와 같은 표준화된 애플리케이션 프로토콜 IAP 구현에 사용할 수 있는 스택

## 1.3 이더넷을 통한 IAP 구현

이 사용자 매뉴얼은 이더넷 통신 주변기기를 사용하여 STM32F4xx용 IAP를 구현하는 두 가지 솔루션을 설명합니다.

- TFTP(Trivial File Transfer Protocol)를 사용한 IAP
- HTTP(Hypertext Transfer Protocol)를 사용한 IAP

두 솔루션 모두 TCP/IP 프로토콜 제품군의 경량 구현인 LwIP 스택 위에서 실행됩니다.

### 1.3.1 TFTP를 이용한 IAP 방식

TFTP를 사용하는 IAP 방법은 펌웨어 업그레이드 기능이 필요한 임베디드 애플리케이션(예: 임베디드 Linux 부트로더)에서 널리 사용됩니다.

TFTP는 UDP 전송 계층 위에서 작동하는 간단한 파일 전송 프로토콜입니다. LAN 환경에서 사용하기 위한 것입니다. 이는 클라이언트가 파일 서버에 파일 전송(읽기 또는 쓰기 작업)을 요청하는 클라이언트/서버 아키텍처를 기반으로 합니다.

이 경우 서버는 PC TFTP 클라이언트의 쓰기 요청만 처리하므로 LwIP 스택 위에 간단한 TFTP 서버가 구현됩니다.

### 1.3.2 HTTP를 이용한 IAP 방식

HTTP 프로토콜을 사용한 펌웨어 업그레이드는 TFTP보다 덜 일반적이지만 인터넷을 통한 원격 프로그래밍이 필요할 때 유용한 솔루션이 될 수 있습니다. 이 경우 최적의 작동을 보장하려면 TCP 전송 프로토콜이 필요합니다.

HTTP는 TCP 위에서 작동하며 HTML 양식을 사용하여 웹 클라이언트(Mozilla Firefox 또는 Microsoft Internet Explorer)에서 이진 파일을 보내는 방법을 제공합니다. 이를 HTTP 파일 업로드(RFC 1867)라고 합니다.

이 문서의 다음 섹션에서는 두 IAP 방법의 구현에 대한 세부 정보와 소프트웨어 사용 방법에 대한 설명을 제공합니다.

## 2 TFTP를 사용한 IAP

### 2.1 TFTP 개요

TFTP는 UDP 위에서 작동하는 간단한 파일 전송 프로토콜입니다. 파일 전송은 TFTP 서버에 읽기 또는 쓰기 요청을 보내는 TFTP 클라이언트에서 시작됩니다. 서버가 요청을 승인하면 파일 데이터 전송이 시작됩니다. 데이터는 고정된 크기의 블록(예: 512 바이트 블록)으로 전송됩니다.

전송된 각 데이터 블록은 다음 블록이 전송되기 전에 수신자가 승인해야 합니다. 승인 메커니즘은 각 데이터 블록과 함께 전송된 블록 번호를 기반으로 합니다. 고정된 블록 크기보다 작은 데이터 블록은 파일 전송이 종료되었음을 나타냅니다.

그림 2에서는 다양한 TFTP 패킷의 형식을 설명합니다.

그림 2. TFTP 패킷

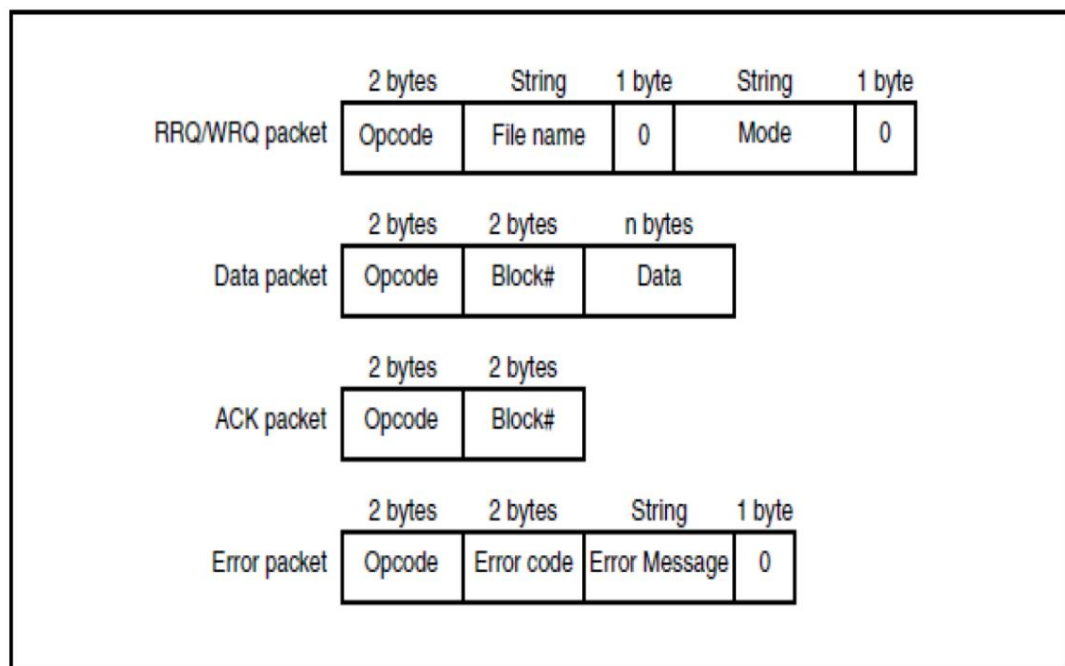


표 1에는 TFTP opcode가 나열되어 있습니다.

표 1. TFTP 연산 코드

연산코드	작업
0x1	읽기 요청(RRQ)
0x2	쓰기 요청(WRQ)
0x3	데이터
0x4	승인(ACK)
0x5	오류

## 2.2 STM32F4xx용 TFTP를 사용하여 IAP 구현

이 IAP 구현은 LwIP TCP/IP 스택 위에 TFTP 서버로 구성됩니다.

이 서버는 원격 TFTP 클라이언트(PC)로부터 받은 파일 쓰기 요청에 응답합니다.

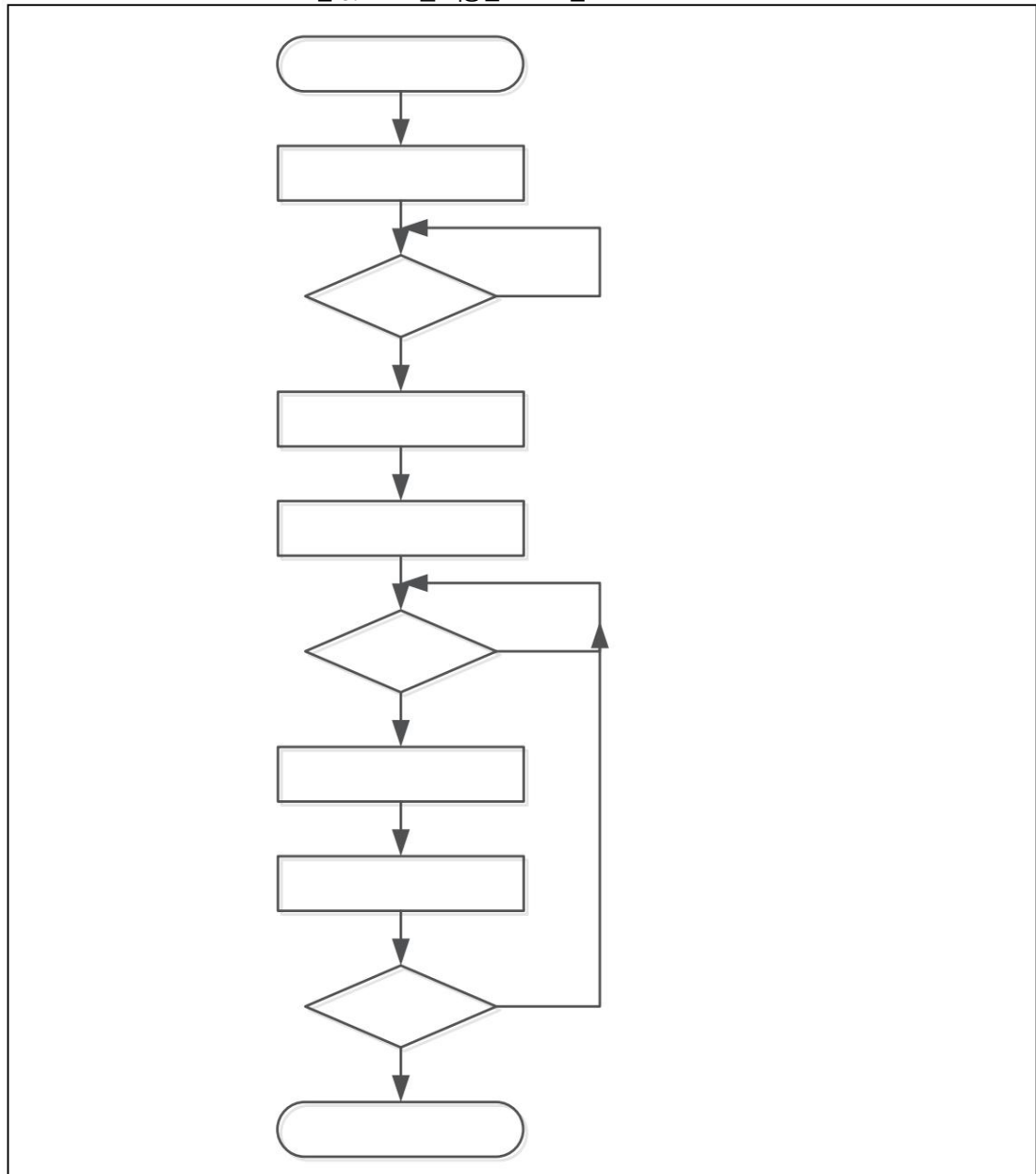
TFTP 읽기 요청은 무시됩니다.

일반적으로 TFTP가 사용되는 파일 시스템에 수신된 파일을 쓰는 대신 서버는 수신된 데이터 블록을 MCU 플래시(사용자 플래시 영역)에 씁니다.

참고: 이 구현에서는 데이터 블록 크기가 512바이트로 고정됩니다.

[그림 3](#)은 TFTP를 사용한 IAP 작업의 개요를 제공합니다.

그림 3. TFTP를 이용한 IAP 흐름도



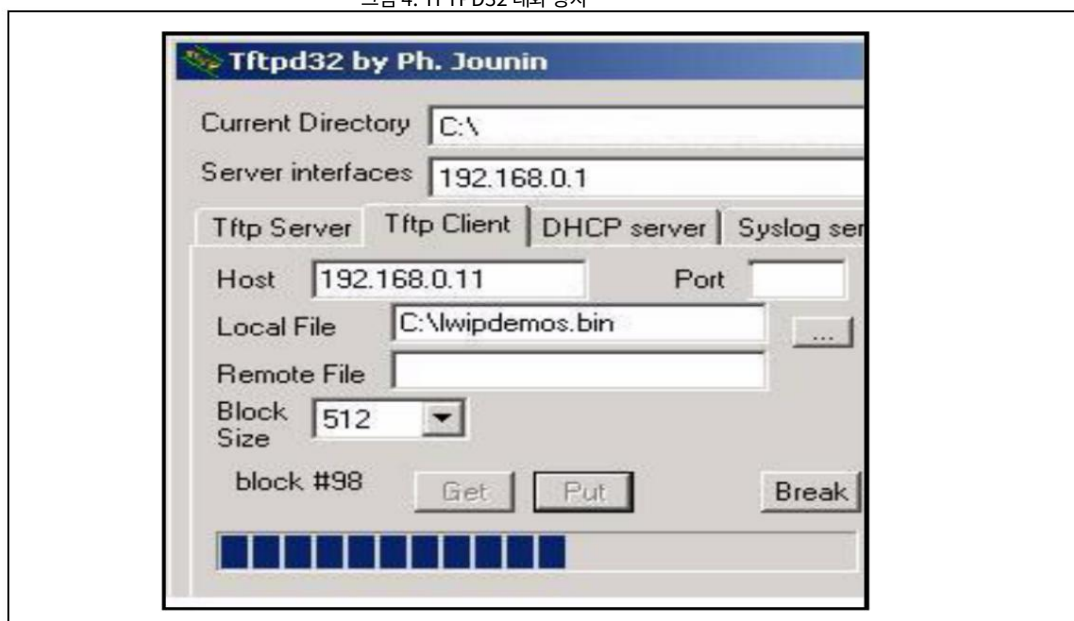


## 2.3 펌웨어 사용

TFTP를 통해 IAP를 테스트하려면 다음 단계를 따르세요.

1. 평가 보드의 점퍼 설정이 올바르게 설정되어 있는지 확인하십시오(참조: [섹션 4.2](#))
2. main.h 파일에서 #define USE\_IAP\_TFTP 옵션의 주석 처리를 제거합니다. 또한 필요에 따라 #define USE\_DHCP 또는 #define USE\_LCD와 같은 다른 옵션의 주석 처리를 해제/주석 처리할 수 있습니다.
3. 펌웨어를 다시 컴파일하십시오. 생성된 맵 파일을 사용하여 IAP 코드 영역(0x0 주소에서 시작)과 main.h에 정의된 USER\_FLASH\_FIRST\_PAGE\_ADDRESS 주소에서 시작하는 사용자 플래시 영역이 겹치지 않는지 확인합니다.
4. STM32F4xx 플래시의 펌웨어를 프로그래밍하고 실행합니다.
5. IAP 모드로 들어가려면 키 버튼을 누른 상태에서 재설정 버튼을 눌렀다가 놓습니다.
6. USE\_LCD가 main.h 파일에 정의되어 있으면 LCD 화면에 IAP 모드가 시작되었음을 나타내는 메시지가 표시됩니다. 또한 DHCP를 사용하는 경우(main.h에 정의된 USE\_DHCP) DHCP IP 주소 할당의 성공 여부를 나타내는 메시지가 LCD 화면에 표시됩니다.
7. IP 주소 할당(정적 또는 동적 주소) 후 사용자는 IP 주소를 시작할 수 있습니다.  
IAP 프로세스.
8. PC 측에서 TFTP 클라이언트(예: TFTP32)를 열고 TFTP를 구성합니다.  
서버 주소(TFTP32의 호스트 주소).
9. STM32F4xx 플래시에 로드할 바이너리 이미지를 찾습니다(바이너리 이미지는 /project/binary 폴더에 예제로 제공됩니다).
10. TFTP32 유틸리티에서 Put 버튼을 클릭하여 파일 쓰기 요청을 시작합니다.
11. LCD가 활성화되면 IAP 작업 진행 상황이 LCD에 표시됩니다.
12. IAP 작업이 끝나면 평가 보드를 재설정하고 애플리케이션을 실행할 수 있습니다.  
STM32F4xx 플래시에서 방금 프로그래밍한 것입니다.

그림 4. TFTP32 대화 상자



## 삼 HTTP를 사용하는 IAP

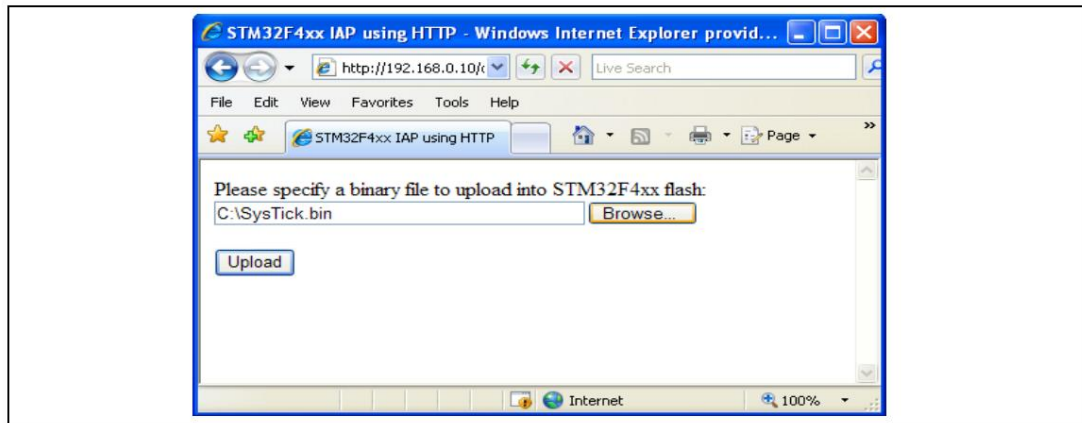
### 3.1 HTTP 파일 업로드 개요

HTTP를 사용한 파일 업로드는 RFC1867에 정의되어 있습니다. 이 파일 업로드 방법은 HTML 양식을 기반으로 합니다. 원시 바이너리 데이터를 전송하려면 GET 대신 HTML POST 메시지가 사용됩니다.

다음은 양식 기반 파일 업로드를 구현하기 위한 HTML 코드의 예입니다.

```
<form action="/upload.cgi" enctype="multipart/form-data" method="post">
  <p>STM32F4xx 플래시에 업로드할 바이너리 파일을 지정하세요.
  <br>
  <input type="file" name="datafile" size="40">
</p>
<div>
  <input type="submit" value="업로드">
</div></form>
```

그림 5. 파일 업로드 HTML 양식의 브라우저 보기



찾아보기 버튼을 눌러 업로드할 바이너리 파일을 선택한 다음 업로드 버튼을 눌러 전송하세요.

파일 크기에 따라 데이터는 연속적인 TCP 세그먼트로 웹 서버로 전송됩니다.

메모:

파일 데이터를 보내기 전에 웹 클라이언트는 파일 이름, 콘텐츠 길이 등의 정보가 포함된 HTTP 헤더 데이터를 보냅니다. 이 중 일부는 웹 서버에서 구문 분석해야 합니다.

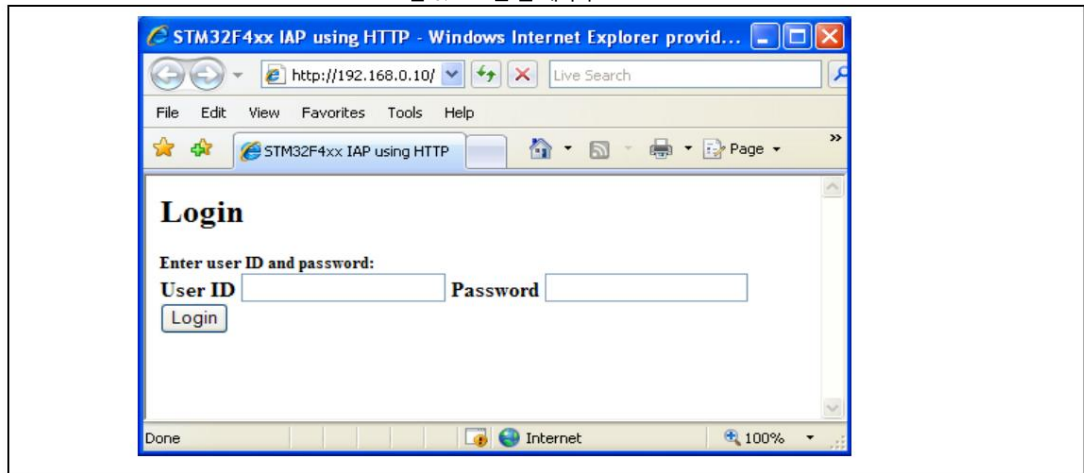
웹 클라이언트는 항상 동일한 HTTP 헤더 형식을 갖지 않습니다. http 웹 서버는 이러한 차이점을 처리해야 합니다.

### 3.2 STM32F4xx용 HTTP를 사용하여 IAP 구현

이 IAP 구현은 LwIP 스택 위에 HTTP 웹 서버로 구성됩니다.

브라우저에 STM32 IP 주소를 입력하면 로그인 웹 페이지가 표시됩니다 (그림 6). 이 로그인 웹 페이지는 IAP 파일 업로드에 대한 액세스를 승인된 사용자로 제한합니다.

그림 6. 로그인 웹 페이지



올바른 사용자 ID와 비밀번호 (main.h 파일에 미리 정의됨)를 입력하고 로그인을 클릭하세요.  
단추. 그러면 파일 업로드 웹 페이지가 로드됩니다(그림 5 참조).

메모:

1. 기본 사용자 ID는 user이고 비밀번호는 stm32입니다.
2. 사용자 ID나 비밀번호가 올바르지 않은 경우 로그인 웹페이지가 다시 로드됩니다. 로그인에 성공한 후 STM32 플래시에 로드할 바이너리 파일을 찾아 선택하세요.
3. 바이너리 파일 크기가 STM32 사용자 플래시의 전체 크기를 초과하지 않는지 확인하십시오.  
영역.
4. 업로드 버튼(그림 5 참조)을 클릭하면 POST 요청이 서버로 전송됩니다.  
이때 서버는 모든 사용자 플래시 영역을 삭제하기 시작하고 바이너리 파일 원시 데이터를 기다립니다. 수신된 데이터는 사용자 플래시 영역에 기록됩니다.
5. 수신할 데이터의 전체 길이는 전송 시작 시 전송된 HTTP 헤더 데이터에서 추출됩니다.
6. IAP 작업이 끝나면 웹 페이지에 IAP 작업 성공 여부가 표시되며,  
MCU를 재설정할 수 있는 버튼이 표시됩니다.

그림 7. 파일 업로드 완료

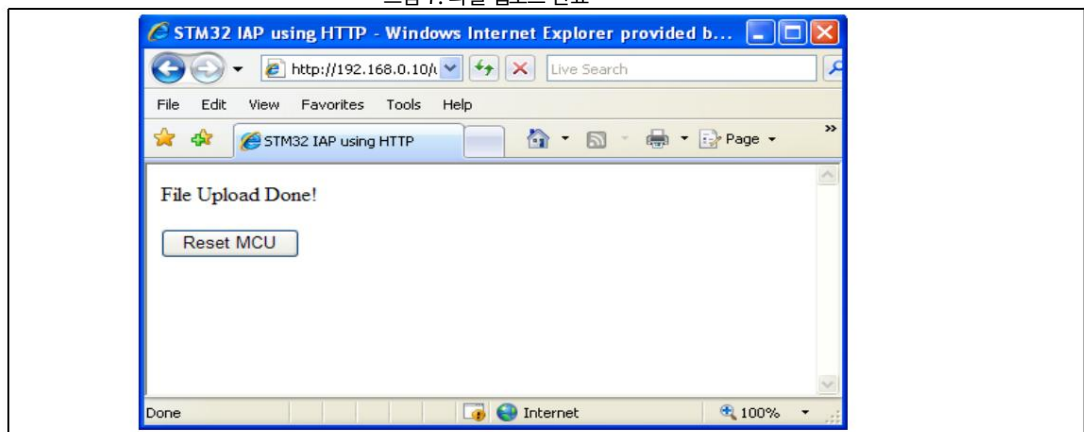
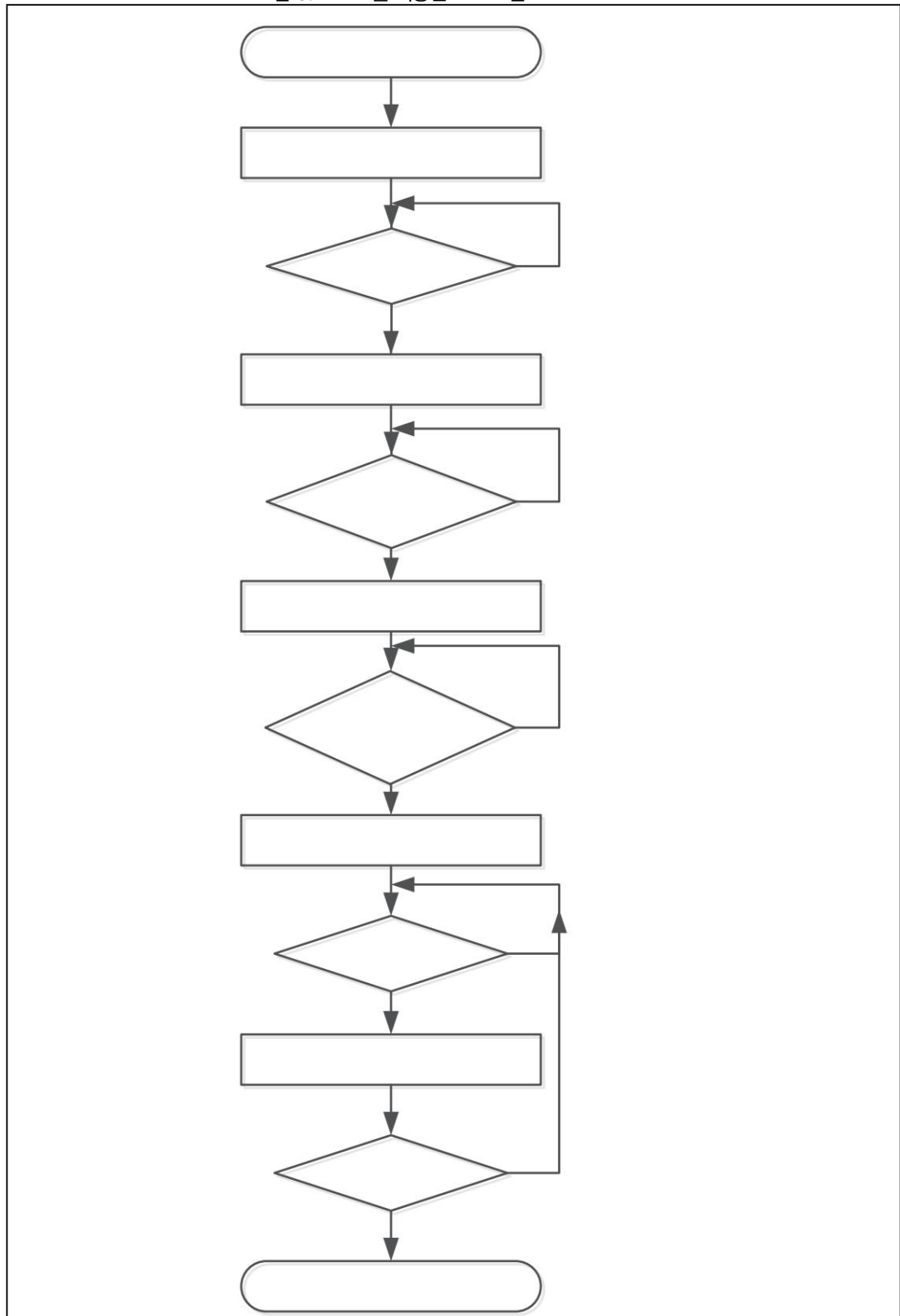


그림 8은 HTTP를 이용한 IAP 방식을 요약한 것이다.

그림 8. HTTP를 이용한 IAP 흐름도



### 3.3 펌웨어 사용

HTTP를 사용하여 IAP를 테스트하려면 다음 단계를 따르세요.

1. 평가 보드의 점퍼가 올바르게 설정되었는지 확인하십시오( [섹션 4.2 참조](#)).
2. main.h 파일에서 USE\_IAP\_HTTP 옵션의 주석 처리를 제거하고 필요에 따라 USE\_DHCP 또는 USE\_LCD와 같은 다른 옵션의 주석 처리/주석을 제거할 수도 있습니다.
3. 펌웨어를 다시 컴파일하십시오. 생성된 맵 파일을 사용하여 IAP 코드 영역(0x0 주소에서 시작)과 USER\_FLASH\_FIRST\_PAGE\_ADDRESS(main.h에 정의됨) 주소에서 시작하는 사용자 플래시 영역이 겹치지 않는지 확인합니다.
4. STM32F4xx 플래시에 펌웨어를 프로그래밍하고 실행합니다.
5. IAP 모드로 들어가려면 키 버튼을 누른 상태에서 재설정 버튼을 눌렀다가 놓습니다. 놓았습니다.
6. USE\_LCD가 main.h 파일에 정의되어 있으면 LCD 화면에 IAP 모드가 시작되었음을 나타내는 메시지가 표시됩니다. 또한 DHCP(main.h에 정의된 USE\_DHCP)를 사용하는 경우에도 DHCP IP 주소 할당의 성공 여부를 나타내는 메시지가 LCD 화면에 표시됩니다.
7. IP 주소 할당(정적 또는 동적 주소) 후 사용자는 IP 주소를 시작할 수 있습니다. IAP 프로세스.
8. 웹 클라이언트(Mozilla Firefox 또는 Microsoft Internet Explorer)를 열고 STM32를 입력합니다. IP 주소.
9. 로그인 웹 페이지가 표시됩니다. 사용자 ID 필드에 "user"를 입력하고 비밀번호에 필드에 "stm32"를 입력한 다음 로그인 버튼을 누릅니다.
10. 그러면 fileupload.html 웹 페이지가 로드됩니다. STM32 플래시에 로드할 바이너리 이미지를 찾은 다음 업로드 버튼을 눌러 IAP 프로세스를 시작하세요.
11. LCD가 활성화되면 IAP 작업 진행 상황이 LCD에 표시됩니다.
12. IAP 작업이 끝나면 성공을 나타내는 새로운 웹 페이지가 로드됩니다. 파일 업로드 작업.
13. RESET MCU 버튼을 눌러 MCU를 재설정하고 애플리케이션을 실행하세요. STM32F4xx 플래시에 프로그래밍되었습니다.

메모:

1. LCD가 활성화된 상태에서 연결 문제가 있는 경우 연결 실패를 나타내는 오류 메시지가 LCD 화면에 표시됩니다.
2. 소프트웨어는 Microsoft Internet Explorer 8 및 Mozilla Firefox 24 웹 클라이언트에서 테스트되었습니다.

## 4 환경

### 4.1 애플리케이션 설정

#### 4.1.1 PHY 인터페이스 구성

이더넷 주변 장치는 외부 PHY와 인터페이스되어 물리 계층 통신을 제공합니다. PHY 레지스터 정의 및 정의는 HAL 구성 파일 "stm32f4xx\_hal\_conf.h" 아래에 있습니다.

PHY는 MII와 RMII의 두 가지 모드로 작동합니다. 필요한 모드를 선택하려면 사용자는 이더넷 주변 장치를 초기화할 때 "Init" 구조의 "MediaInterface" 매개변수를 채워야 합니다.

메모: 지원되는 보드에서 사용 가능한 PHY 인터페이스 모드에 대해 자세히 알아보려면 장치 이더넷 IAP 예제에 제공된 추가 정보 파일을 참조하세요.

#### 4.1.2 MAC 및 IP 주소 설정

기본 MAC 주소는 00:00:00:00:00:02로 설정됩니다. 이 주소를 변경하려면 stm32f4xx\_hal\_conf.h 파일에 정의된 6바이트를 수정하십시오.

기본 IP 주소는 192.168.0.10으로 설정됩니다. 이 주소를 변경하려면 main.h 파일에 정의된 6바이트를 수정하십시오.

### 4.2 평가 보드 설정

이더넷 IAP 예제를 실행하기 전에 해당 Readme 파일을 읽고 보드 점퍼를 구성하여 올바른 작동을 보장하는 방법을 알아보세요.

### 4.3 펌웨어 파일 구성

이더넷 IAP 예제 소스는

Projects\STM324xx\_EVAL\Applications\LwIP\LwIP\_IAP\에서 사용할 수 있습니다. 여기서 STM324xx\_EVAL은 STM32F4xx EVAL 보드(예: STM32F407/417 라인의 경우 STM324xG-EVAL)를 나타냅니다. [표 2](#) 예제 소스 파일에 대해 설명합니다.

표 2. 파일 구성

파일 이름	설명
main.c	주요 애플리케이션 파일
메인.h	기본 구성 파일
httpserver.c/.h	HTTP 서버 구현
tftpserver.c/.h	TFTP 서버 구현
flash_if.c/.h	높은 수준의 플래시 액세스 기능
stm32f4xx_it.c/.h	인터럽트 핸들러
fsdata.c	ROM 파일 시스템으로서의 HTML 파일

표 2. 파일 구성

파일 이름	설명
lwipopts.h	LwIP 구성 옵션
이더넷if.c/.h	LwIP와 이더넷 드라이버 간의 인터페이스
stm32f4xx_hal_conf	HAL 구성 파일

메모: 이 표에는 STM32Cube HAL 및 BSP 라이브러리와 LwIP 스택에서 사용되는 파일이 표시되지 않습니다.

## 4.4 IAP용 이미지 구축

IAP(IAP 소프트웨어를 사용하여 로드)용 이미지를 빌드하려면 다음을 확인하세요.

1. 펌웨어는 사용자 플래시 영역의 시작 주소(main.h의 USER\_FLASH\_FIRST\_PAGE\_ADDRESS에서 정의한 주소와 동일해야 함)부터 실행되도록 컴파일 및 링크됩니다.
2. 벡터 테이블 시작 주소는 사용자 플래시 영역의 시작 주소로 구성됩니다.  
벡터 테이블 베이스 오프셋은 system\_stm32f4xx.c 파일에 정의된 상수 VECT\_TAB\_OFFSET 값을 수정하여 구성됩니다. 예를 들어, 벡터 테이블 기본 위치를 0x08020000으로 설정하려면: #define VECT\_TAB\_OFFSET 0x20000
3. 컴파일된 소프트웨어 크기는 전체 사용자 플래시 영역을 초과하지 않습니다.

## 5 결론

이 사용자 매뉴얼의 목적은 STM32F4xx 마이크로컨트롤러용 STM32Cube HAL 드라이버를 사용하는 이더넷 IAP(In-Application 프로그래밍)를 설명하는 것입니다.

HTTP 및 TFTP 프로토콜을 지원하기 위해 두 가지 솔루션이 제공됩니다. 둘 다 TCP/IP 통신을 위한 미들웨어 구성 요소인 LwIP 스택을 기반으로 합니다.



## 부록 A FAQ

### A.1 정적 또는 동적(DHCP) IP 주소 할당 중에서 선택하는 방법

"main.h"에 있는 매크로 `#define USE_DHCP`가 주석 처리되면 고정 IP 주소가 STM32 마이크로컨트롤러에 할당됩니다(기본적으로 192.168.0.10, 이 값은 "main.h" 파일에서 수정 가능).

`#define USE_DHCP` 매크로의 주석 처리가 제거되면 DHCP 프로토콜이 활성화되고 STM32는 DHCP 클라이언트로 작동합니다.

### A.2 이더넷 케이블 연결이 끊어졌을 때 애플리케이션이 작동하는 방식

케이블이 분리되면 이더넷 주변 장치는 전송 및 수신 트래픽을 모두 중지하고 네트워크 인터페이스도 설정됩니다. LCD 컨트롤러를 사용하는 경우 케이블이 연결되지 않았음을 사용자에게 알리는 메시지가 표시됩니다. 그렇지 않으면 평가 보드의 빨간색 LED가 켜집니다.

사용자가 케이블을 다시 연결하면 이더넷 트래픽이 재개되고 네트워크 인터페이스가 설정됩니다. LCD 컨트롤러를 사용하는 경우 정적 또는 동적 할당을 통해 사용자에게 새 IP 주소를 알리는 메시지가 표시됩니다. 그렇지 않으면 평가 보드의 노란색 LED가 켜집니다.

### A.3 다른 하드웨어에 애플리케이션을 이식하는 방법

다른 하드웨어 플랫폼을 사용하는 경우 이더넷 주변 장치에 대한 `HAL_ETH_MspInit()` 함수에 GPIO 구성을 확인해야 하며, 애플리케이션에 더 많은 PPP 주변 장치가 필요한 경우 `HAL_PPP_MspInit()` 또는 `HAL_MspInit()`도 확인해야 합니다.

## 6 개정 이력

표 3. 문서 개정 내역

날짜	개정	변경 사항
2014년 3월 28일	1	초판
2015년 2월 5일	2	업데이트된 <a href="#">섹션: 소개</a> 및 <a href="#">섹션 1: IAP 개요</a>
2015년 5월 27일	삼	<p><a href="#">섹션: 소개</a> 가 업데이트되어 STM32Cube 개요 섹션과 병합되었습니다.</p> <p>RMII 모드와 관련된 참고 사항을 제거하고 <a href="#">섹션 4.1.1: PHY 인터페이스 구성</a>에 지원되는 보드에 관한 참고 사항을 추가했습니다.</p> <p><a href="#">섹션 4.2: 평가 보드 설정</a> 에서 전용 평가 보드 설정 섹션을 제거했습니다 .</p> <p>업데이트된 <a href="#">섹션 4.3: 펌웨어 파일 구성</a> 소개.</p>

중요 공지 - 주의 깊게 읽어주세요

STMicroelectronics NV 및 그 자회사("ST")는 통지 없이 언제든지 ST 제품 및/또는 이 문서를 변경, 수정, 향상, 수정 및 개선할 수 있는 권리를 보유합니다. 구매자는 주문하기 전에 ST 제품에 대한 최신 관련 정보를 얻어야 합니다. ST 제품은 주문 승인 당시의 ST 판매 약관에 따라 판매됩니다.

ST 제품의 선택, 선택 및 사용에 대한 책임은 전적으로 구매자에게 있으며, ST는 구매자 제품의 애플리케이션 지원이나 설계에 대해 어떠한 책임도 지지 않습니다.

ST는 지적 재산권에 대해 명시적이든 묵시적이든 어떠한 라이선스도 부여하지 않습니다.

여기에 명시된 정보와 다른 조항을 적용하여 ST 제품을 재판매할 경우 해당 제품에 대해 ST가 부여한 보증이 무효화됩니다.

ST 및 ST 로고는 ST의 상표입니다. 기타 모든 제품 또는 서비스 이름은 해당 소유자의 자산입니다.

이 문서의 정보는 이 문서의 이전 버전에서 이전에 제공된 정보를 대체하고 대체합니다.

© 2015 STMicroelectronics - 판권 소유

