

## 一起学习 CC3200 系列教程之 HttpServer 前序

阿汤哥

序:

能力有限，难免有错，有问题请联系我，

QQ1519256298 [hytga@163.com](mailto:hytga@163.com)

Pdf 下载 <http://pan.baidu.com/s/1hqiWB56>

现在介绍 HttpServer，本来应该是 TCP 或者是 UDP 开始讲的，但是考虑到 HTTPServer 的现象比较容易看到，所以就先讲讲，考虑到对者一块也不怎么了解，所以就先来一个前序吧！

首先，介绍 PDF，这个是官方的，被我重新命名了下，叫 CC3200 网络处理器系统\_编程指导.pdf,里面主要就是讲 CC3200 的网络部分，有兴趣的自己到我的网盘下载。

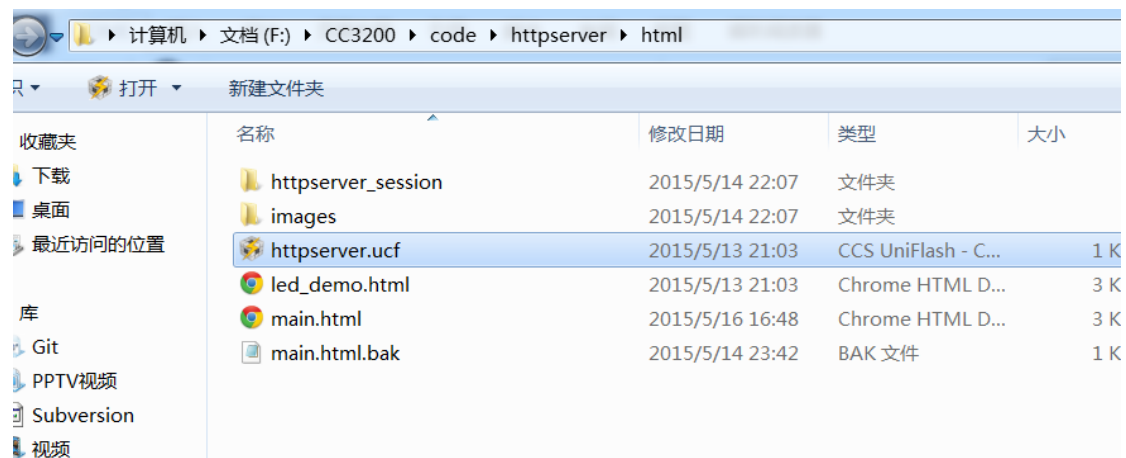
首先需要对 CC3200 进行烧录

- 1、 Format 格式化
- 2、 service pack updata 更新固件
- 3、 program 烧录文件

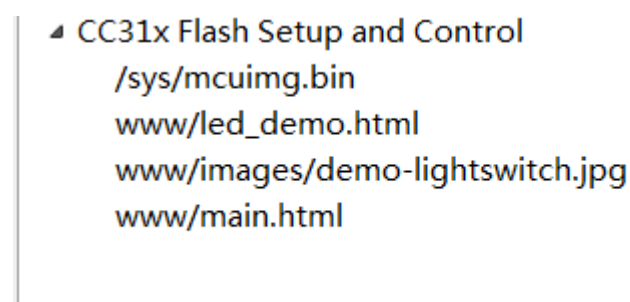
ti 提供了一个文件系统，uniflash 工具把这些文件烧录到外挂的 flash 中，当然我们是不太清楚文件是怎么放的，也没必要知道。

怎么烧录文件？

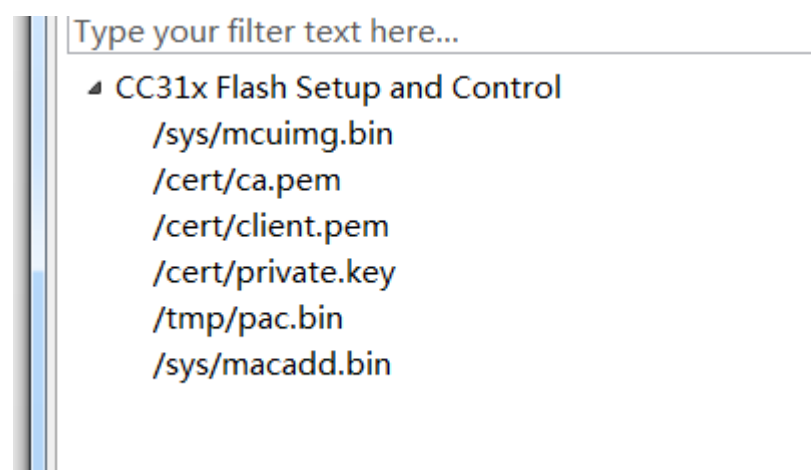
利用 **httpServer** 例程自带的一个后缀为 **ucf** 文件，如下图



用 **uniflash** 打开它。这是一个配置文件，打开后 **uniflash** 的界面是这样的

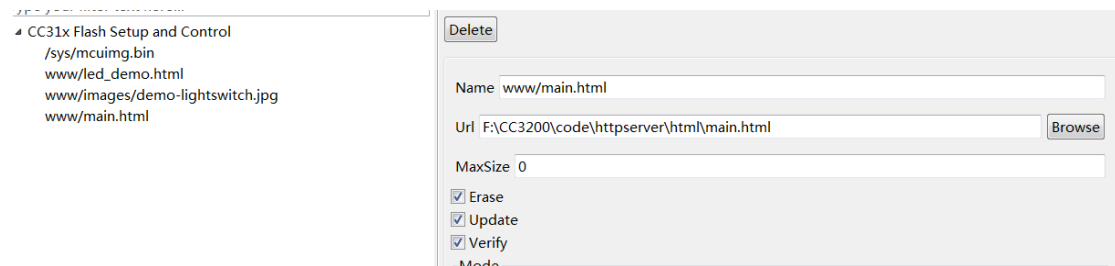


如果我们是打开一个全新的配置文件，**uniflash** 的界面是这样的



看到了熟悉的文件路径没有？这些文件都是需要用 **uniflash** 去烧录的。

## 举例：main.html



一般我们都是选择上图的选项。其中 **URL** 指向的文件的路径，**name** 是很重要的，就是一个存放在 **flash** 的文件名。

一般作为 **httpserver** 我们关心的是

- 1、 网页
- 2、 数据的上传和下传

因此在这一部分我们就讨论这个两个方面的问题，其他的留到以后再说

### 1、 网页

**CC3200 网络处理器系统\_编程指导.pdf** 有以下内容：

The SimpleLink server communicates with the serial flash file system, which hosts the web page files. The files are saved in the serial flash under their own filename. Filenames can include the full path to achieve a directory structure-like behavior. For security purposes, the ability of the web server to access the file system is limited to the following root folders:

- (a) **www/**
- (b) **www/safe/**

**Important:** One of these two root folders should be prefixed to the filename when downloading files to the file system.

大概意思就是要把网页放在这两个路径下。这个路径已经被固定住了。

当我们在浏览器中输入一个网址时，就会跳到网站的主页面：

这个主页面就是 **main.html** 或者是 **index.html**,这个文件名也是固定死了。我是找不到有什么地方可以修改的这个。。。

我曾经用 **uniflash** 把这个 **main.html** 擦除掉，但是还是有主页面存在，

所以猜测，里面有默认的 **main.html** 存在。

## 2、数据的传输

数据的传输有两种方式 **get** 和 **post**，**get** 主要用于获取数据，**post** 主要用于上传数据，数据的传输需要用户自己实现。

如何使用 **httpserver**：

- 1、 提供了 **HTML** 网页数据
- 2、 开启 **httpserver**
- 3、 编写一个回调函数：用于处理 **get** 和 **post**

其中 1 部分前面已经介绍过了，用 **uniflash** 烧录

第 2 部分，用一个函数就搞定了

**//Start Internal HTTP Server**

```
lRetVal = sl_NetAppStart(SL_NET_APP_HTTP_SERVER_ID);
```

第 3 部分：比较麻烦，需要用户在 **main** 文件定义以下函数

**//http** 的回调函数:在这一部分主要处理的是 **post** 和 **get**

```
Void                               SimpleLinkHttpServerCallback(SlHttpServerEvent_t  
*pSlHttpServerEvent,  
  
SlHttpServerResponse_t  
*pSlHttpServerResponse)
```

为什么我们只要在这里定义一个这样的函数就能处理 **get** 和 **post**?

其实这个涉及到一个头文件，这个用户是需要根据应用去改的，这个叫 **user.H**

我截了一张图

```
/*
\brief      A callback for HTTP server events.
            Possible events are:
            SL_NETAPP_HTTPGETTOKENVALUE - NWP requests to get the value of a specific token
            SL_NETAPP_HTTPPOSTTOKENVALUE - NWP post to the host a new value for a specific token

\param      pServerEvent - Contains the relevant event information (SL_NETAPP_HTTPGETTOKENVALUE or SL_NETAPP_HTTPPOSTTOKENVALUE)
\param      pServerResponse - Should be filled by the user with the relevant response information (i.e SL_NETAPP_HTTPSETTOKENVALUE as a res
\sa|
\note      belongs to \ref porting_sec
\warning
*/
#define sl_HttpServerCallback    SimpleLinkHttpServerCallback
```

看懂了吧，就是这个东西。

我们前面有说过 **M4** 和 **NWP** 的关系，**NWP** 就像是 **M4** 的一个外设，有一个中断，当 **NWP** 发生了事件，这个中断就触发了，接下来就运行 **httpserver** 的回调函数 **SimpleLinkHttpServerCallback**。

所以我们就要这个函数里去处理 **get** 和 **post** 了。

如果对 **get** 和 **post** 不懂的同学可以 **google** 下。

我提供了我在 **main** 函数的注释，有需要自己到我的网盘下载

文件名是 **HTTP\_SERVER\_main.c**，注释也不是很多，希望对你有帮助！

最后啰嗦下怎么去使用 **httpsever** 例程，

- 1、 用跳帽把 **P58** 和 **VCC** 连起来，选择的是 **ap** 模式，设置 **sop【2:0】=0b100**，编译代码
- 2、 用 **uniflash** 进行一下操作

选择串口号

导入配置文件 **httpserver.ucf**，里面有一些选项，你可以根据自己的 **HTML** 和 **bin** 文件路径去选择，或者是直接默认掉

**format, server pack updata, grogram。**

烧录成功后

设置 **sop【2: 0】=0b000**，复位芯片，用手机连上 **CC3200** 的 **wifi**。

浏览器输入 **http: //mysimplelink.Net**

结束。。。

备注：我的电脑一直连不上 **CC3200** 的 **WIFI**，不知道是什么原因？难道是 **wifi** 没有设置密码？可怜的只能用手机。。。