基于 TI-RTOS 的互联网固件升级参考设计

TIVA 应用

Holly Gu

摘要

随着嵌入式设备功能的多样化和现场应用的复杂化，系统固件更新变得越来越频繁，远程升级产品的需求日益迫切，如何便捷地完成设计并简化用户操作成为关注的焦点。本文基于 TI 的开发板，实现了基于互联网的远程自动升级解决方案。软件系统采用 TI-RTOS（TI-Real Time Operation System）实时操作系统，通过 HTTP（HyperText Transfer Protocol）GET 功能实现互联网远程升级。设计完整，使用方便，易于扩展。

目录

1 系统简介 2
1.1 微处理器及系统平台简介 2
1.2 远程升级 4
2 软件系统实现 4
2.1 HTTP 固件下载 5
2.1.1 GET 请求下载 5
2.1.2 固件文件处理 7
2.2 实时系统 TI-RTOS 的构建 8
2.2.1 TI RTOS 系统配置 8
2.2.2 网络配置 9
2.2.3 文件系统配置 10
2.2.4 多任务配置 10
2.3 本地升级 11
3 演示安装 12
3.1 硬件配置 12
3.2 软件配置 12
4 演示运行 13
5 软件及其资源 14
5.1 Bootloader 工程 14
5.2 TI-RTOS 工程 15
6 引用 17

图例

图 1. 开发板功能结构图 3
图 2. 开发板展示图 3
图 3. 互联网远程升级系统功能示意图 4
图 4. 版本信息判定软件流程图 6
图 5. 固件文件下载软件流程图 6
图 6. 固件文件处理软件流程图 7
图 7. TI-RTOS 实时系统的功能模块图 8
图 8. TI-RTOS 实时系统配置图 9
图 9. TI-RTOS 实时系统网络功能配置图 10
图 10. 互联网升级任务软件流程图 11
图 11. Bootloader 更新程序软件流程图 12
图 12. HTTP 服务器网页示图 13
图 13. 系统正常运行串口打印信息图 14
图 14. Bootloader 功能代码列表图 15
图 15. 互联网数据下载功能代码列表图 15

简介

本设计使用 TI 的嵌入式处理器 TM4C1294 实现了一种基于互联网的电子设备远程升级方案。待下载的程序固件存放在远端网站的后台服务器中，固件信息呈现在网页上。电子设备可通过接入互联网自动验证及下载最新的程序固件并完成本地程序更新。整个系统在 EK-TM4C1294XL 开发板上采用 TI-RTOS 实时操作系统实现。易于使用，便于修改，并为用户实现更多更复杂的定制功能提供可能。

1 系统简介

本设计采用 Tiva™ C 系列 LaunchPad 评估板作为硬件开发平台实现设备远程自升级。

1.1 微处理器及系统平台简介

EK-TM4C1294XL 开发板的核心器件采用 TM4C1294NCPDT 高性能微处理器为主控单元。该处理器基于 ARM Cortex-M4 内核，工作频率可达 120MHz，内含 1MB Flash 和 256KB SRAM 存储空间，该处理器的一大亮点是完整集成以太网 MAC（Media Access Control）和 PHY（Physical Layer Interface），使得用户的网络应用变得十分便利。该系列还集成有并口，可外扩内存，并自带高速 USB2.0（Universal Serial Bus）数字接口，集成有数个低速和中速串行接口，12 位 ADC（Analog-to-digital converter）以及显示控制器，可充分满足从人机接口要求较高的工业通信设备的应用需求，其在智能电气、智能物联等领域也有不俗的表现。

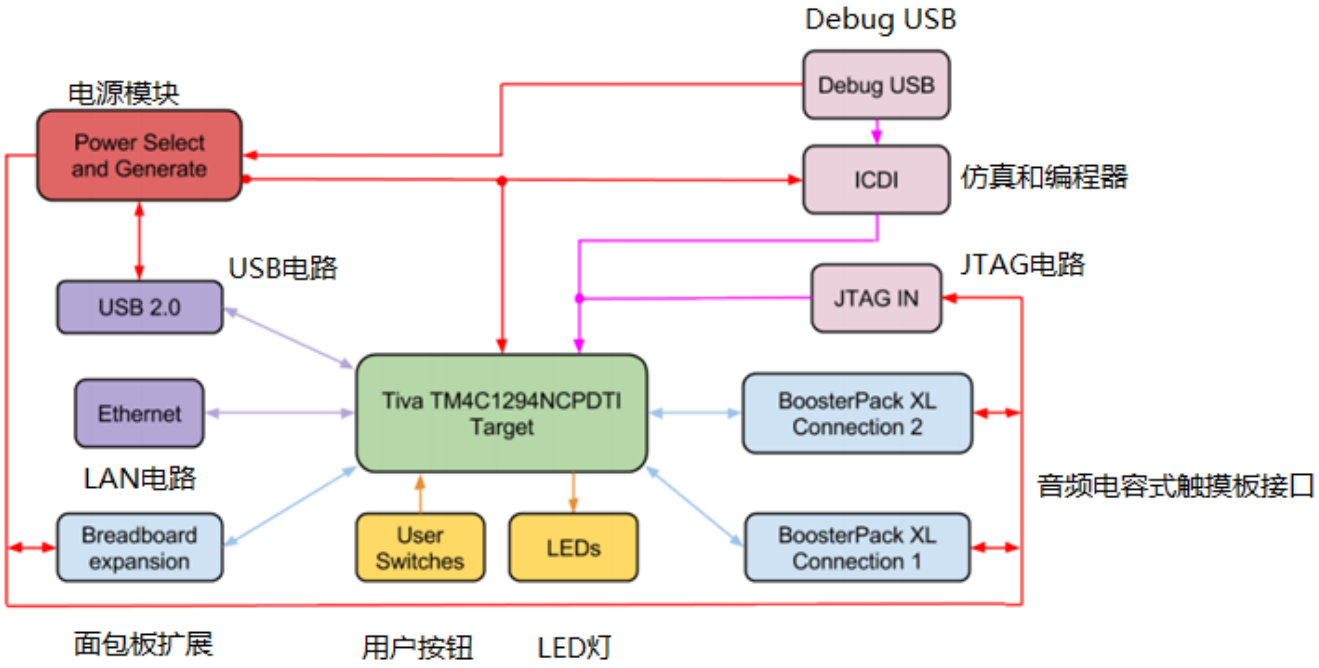

图 1.　　开发板功能结构图

Tiva™ C 系列 TM4C1294 连接 LaunchPad 评估板是用于基于 ARM® Cortex-M4 的微控制器的低成本评估平台。此 LaunchPad 评估板完全通过 TM4C1294NCPDT 微控制器引出 10/100 M 以太网网络接口、USB 2.0 接口、休眠模块、运动控制脉宽调制以及大量同步串行接口。

本设计中通过网络接口和 USB 接口实现主要硬件功能。

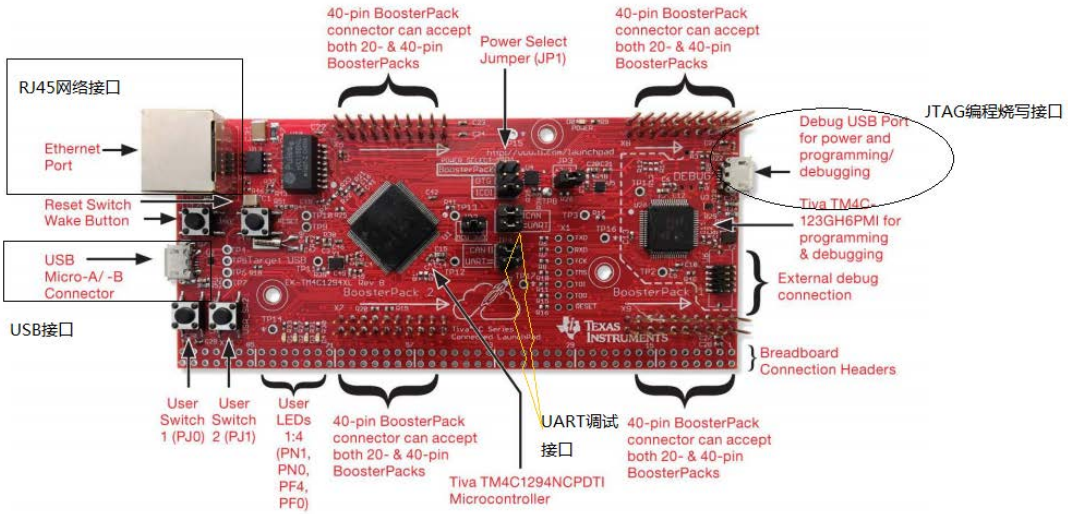

图 2.　　开发板展示图

本设计中用户可通过 RJ45 网络插口接入外部网络，使设备具有互联网访问能力；USB（Universal Serial Bus，通用串行总线）接口用于外扩 U 盘物理存储；JTAG（Joint Test Action Group，联合测试行为组织）接口用于在线调试或程序烧写。考虑到基于实时操作系统进行多任务软件调试的复杂性，本设计特别引入 UART 辅助调试。由于该硬件平台属于现有资源，此处不再过多赘述，更多的开发资料可以参阅 TI 官网上的 EK-TM4C1294XL Design Files [2] 。

1.2 远程升级

常见的嵌入式设备程序升级有两大类：本地升级和远程升级。本地升级是指所需更新的固件被存放在本地，需人工在现场对设备进行固件升级，这种方式较为常见，不需要预设联网功能，开发简单，但是操作繁琐，维护成本高，容易误操作。远程升级是指设备可以通过某种手段远程进行固件更新，这种方式操作简单，可以有效避免人员过多介入，升级过程也更加安全可靠。常见的远程升级一般采用有线或者无线通信的方式，由用户通过上位机远程操作完成升级。这种方式较本地升级更加智能，但仍需人工介入。为了使得升级更加智能便捷，本文设计实现了一种互联网远程自升级方案。设备会主动进行新固件的检查，下载及更新，全程可不需要人工干预。

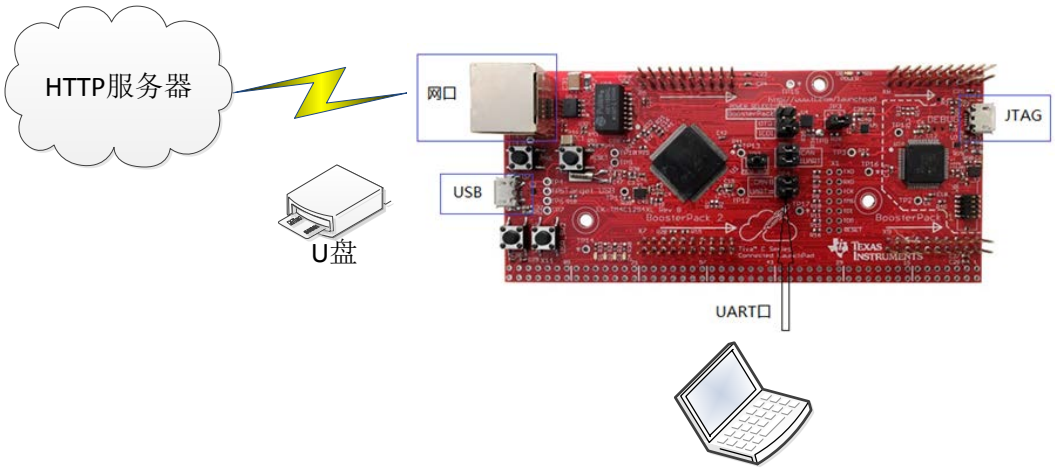

图 3. 互联网远程升级系统功能示意图

整个系统的功能示意图如图 2 所示，在本互联网升级方案中，本地设备被设计成一个 HTTP 客户端，通过 HTTP GET 的方式从远程的 HTTP 服务器中获取设定好的升级固件，从而完成无人值守的本地程序更新。

2 软件系统实现

系统软件主要使用 TI 的集成开发环境 CCS（Code Composer Studio）作为开发调试环境。TI 官方提供了 SDK-TIVAWARE 和实时操作系统 TI-RTOS，包含大量软件代码示例，详尽的技术开发文档，资源丰富，可大大加速用户开发进度。

本设计中软件共分为三大部分：HTTP 固件下载，TI-RTOS 任务调度，本地升级。设备正常上电运行后，每一次网络生效都会进行一次升级文件的检查和下载，并将新的固件存放在 USB 中。待设备重启后，自动完成固件升级。整个升级过程由 TI-RTOS 实时操作系统管理。升级完成后，下载的固件将被自动删除。

2.1 HTTP 固件下载

HTTP 超文本传输协议是互联网上应用最为广泛的网络协议之一，它是客户端浏览器或其他程序与 Web 服务器之间的应用层通信协议，在 Internet 上 Web 服务器里存放的都是超文本信息，客户端需要通过 HTTP 协议传输所要访问的超文本信息。HTTP 协议中包含的命令和传输信息，不仅可用于 Web 访问，也可以用于其他因特网或内联网应用系统之间的通信，从而实现各类应用资源超媒体访问的集成。人们熟悉的所有 WWW 方式的数据传输都必须遵守这个标准。

完整的 HTTP 固件下载可以分为两个部分：HTTP 请求下载和数据处理存储。

2.1.1 GET 请求下载

HTTP 是一个客户端和服务器端请求和应答的标准。通常，由 HTTP 客户端发起一个请求，建立一个到服务器指定端口（默认是 80 端口）的连接。HTTP 服务器则在那个端口监听客户端发送过来的请求。一旦收到请求，服务器（向客户端）发回一个状态行，比如"HTTP/1.1 200 OK"，和（响应的）消息，消息的消息体可能是请求的文件、错误消息、或者其它一些信息。

HTTP/1.1 协议中共定义了八种方法来表明 Request-URI 指定的资源的不同操作方式：

（1）OPTIONS--返回服务器针对特定资源所支持的 HTTP 请求方法。

（2）HEAD--向服务器索要与 GET 请求相一致的响应，只不过响应体将不会被返回。

（3）GET--向特定的资源发出请求。

（4）POST--向指定资源提交数据进行处理请求。

（5）PUT--向指定资源位置上传其最新内容。

（6）DELETE--删除指定资源。

（7）TRACE--回显服务器收到的请求。

（8）CONNECT--HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器。

本文采用常见的 GET 方法获取服务器端存放的数据。该网络功能已经集成进了 TI-RTOS 的 Network Services 功能中。通过调用相应的 API 函数可以实现客户端的功能。本设计并不进行加密处理，整个数据的处理过程均采用明文，供用户自行修改。GET 下载分为两部分：查询版本信息和固件文件下载。

版本信息判定是客户端检查有没有需要更新的固件，其工作流程图如 2 所示：

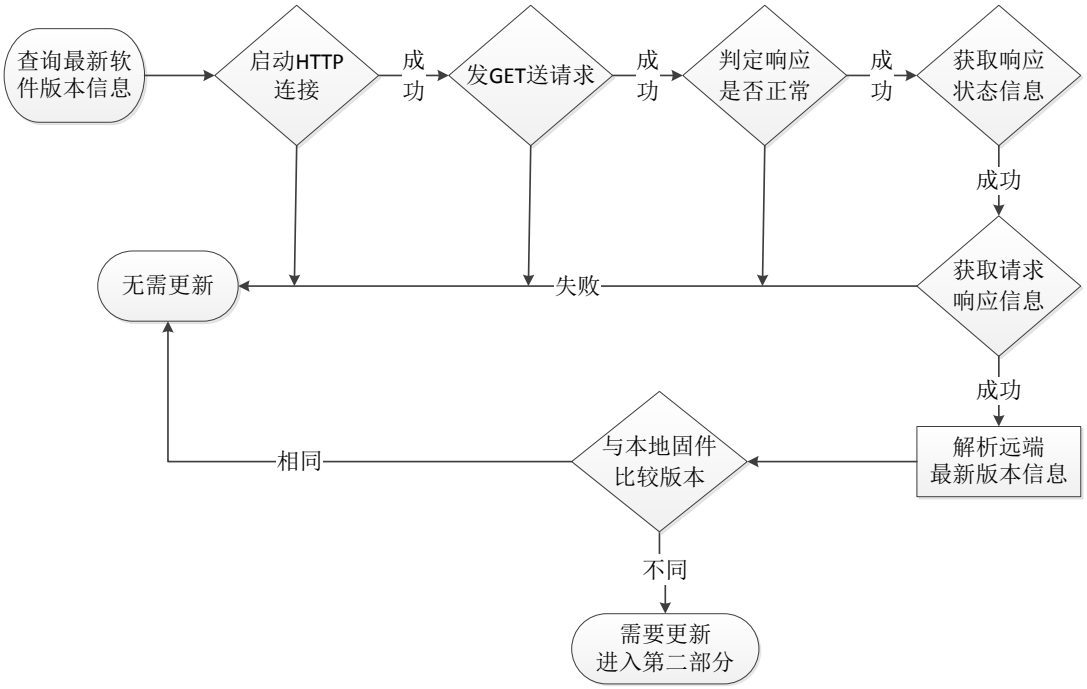

图 4.　　版本信息判定软件流程图

第二部分是固件文件下载，本地设备从服务器端下载所需更新的固件文件数据，流程图如图 3 所示：

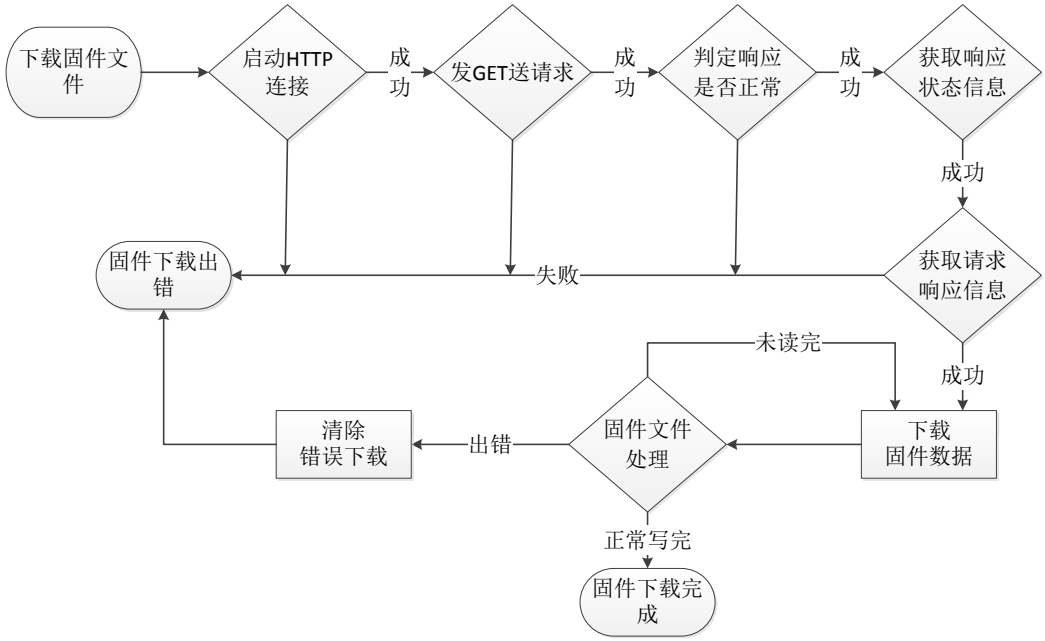

图 5.　　固件文件下载软件流程图

版本信息判定和固件文件下载都使用 HTTP GET 请求获取数据，只是信息的有效部分含义不同。总的来说，客户端首先从服务器端获取版本信息，如果网页中信息表明固件版本与设备当前版本不同，则判定需要升级。接着就可以获取固件文件数据。数据下载正常，才可以进入到随后的 U 盘处理过程。

2.1.2 固件文件处理

通过上节的介绍，已经能通过 HTTP 实现固件文件下载。为了实现自升级，还需将下载的固件保存到设备本地的存储空间内。本节主要进行固件文件的处理和存储。

固件文件并不一定一次下载完成，如果是首次下载，则调用文件系统创建一个临时的本地升级文件 HTTPUPFW.BIN，否则，对已创建的临时文件进行续写。数据集齐后，创建一个可被用于升级的文件 FIRMWARE.BIN，中途一旦发生异常，则删除该临时文件 HTTPUPFW.BIN，放弃本次升级操作。固件文件处理软件流程图如图 4 所示。

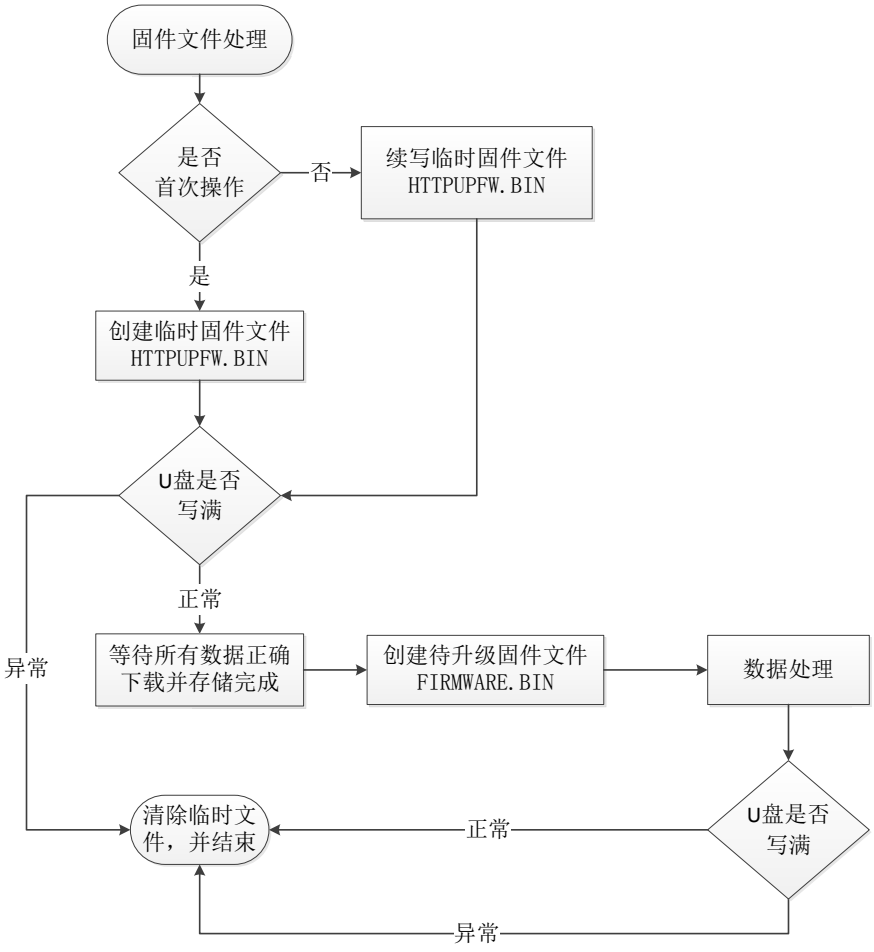

图 6. 固件文件处理软件流程图

本设计中，正确获取临时固件文件 HTTPUPFW.BIN 后，会创建真正用于升级的固件文件 FIRMWARE.BIN。本文这样做的目的是留出一定的接口给用户修改，以适应用户有特殊场景需求。例如从服务器上获取的升级文件是经过加密的，可在此处进行解密处理。除此之外，本设计中使用的存储设备是 USB 存储器，当然也可以是 SD 卡等其他设备。

2.2 实时系统 TI-RTOS 的构建

TI-RTOS 是由 TI 推出面向嵌入式处理器平台、基于抢占式多线程内核的完整的实时操作系统，其功能模块框图如图 5 所示。

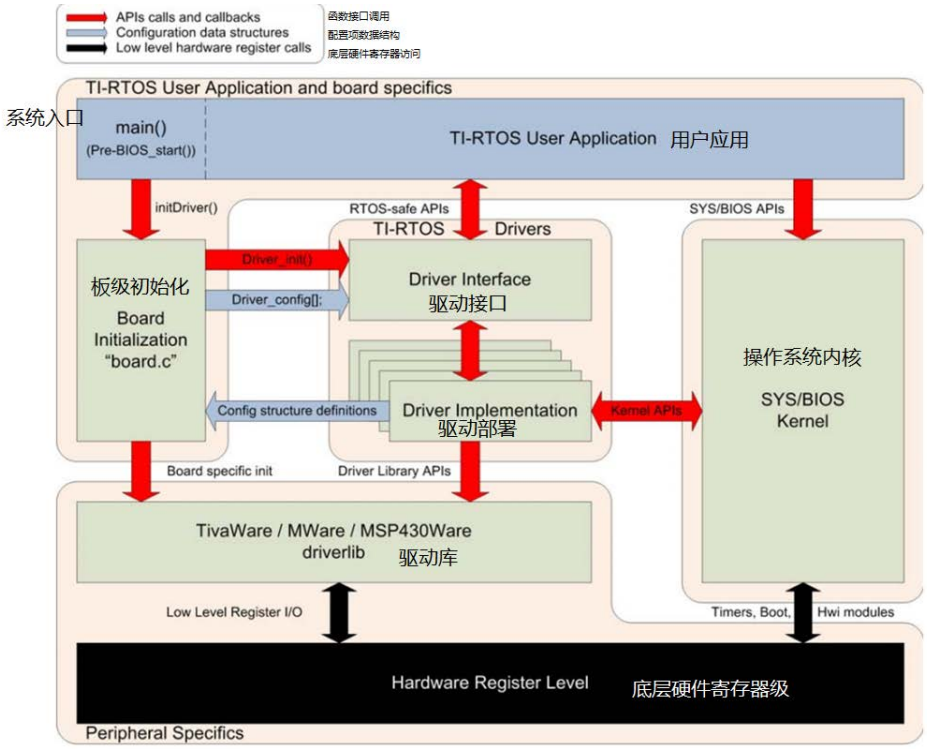

图 7. TI-RTOS 实时系统的功能模块图

该实时操作系统包含各种 RTOS 组件包，例如普及型 SYS/BIOS 实时内核和网络开发套件 TCP/IP 协议栈，设计人员无需编写和维护诸如调度工具、协议栈以及低级驱动器等复杂的系统软件程序，可将更多的时间和精力放在自己差异化的应用开发上。同时，TI 提供了完整的开发软件和文档，为 TI 合作的伙伴带来一种免费的、无专利限制的广泛应用型平台。

2.2.1 TI RTOS 系统配置

在 CCS 中，集成有图形化的用户界面可供用户直接对 TI-RTOS 进行静态配置，为开发提供方便。如图 6 所示。

TI-RTOS 模块大致可分为 5 大部分：调试功能模块、通信功能模块、内核模块、文件系统模块、驱动模块。调试功能模块是指使用 CCS 直接调试 RTOS 应用时的调试方法，本例启用 Logging；通信功能模块是指有线或无线通信支持，本例启用有线网络通信。文件系统模块使用 FatFS，可以支持 USB 存储设备和 SD 卡等存储设备。内核模块和驱动模块是必选模块，其中内核模块可以通过图形化设定配置参数，本例使用默认参数；设备驱动需要在代码中自行添加或删除特定功能。

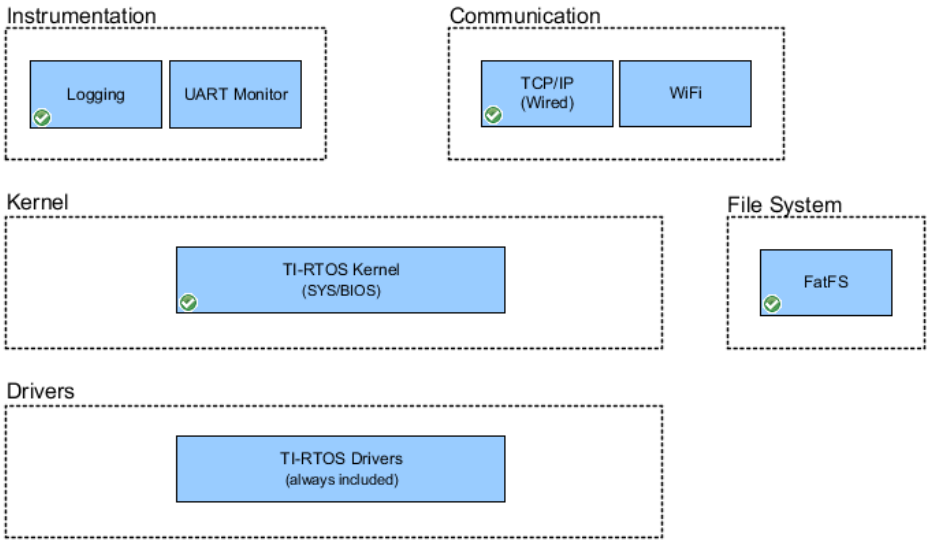

图 8.　　TI-RTOS 实时系统配置图

2.2.2　网络配置

本设计中为了使用 HTTP 功能，将操作系统中的网络传输服务选定为 TCP（Transmission Control Protocol）传输。数据链路层，网络互联层，传输层和应用层的图形化设置如图 7 所示。

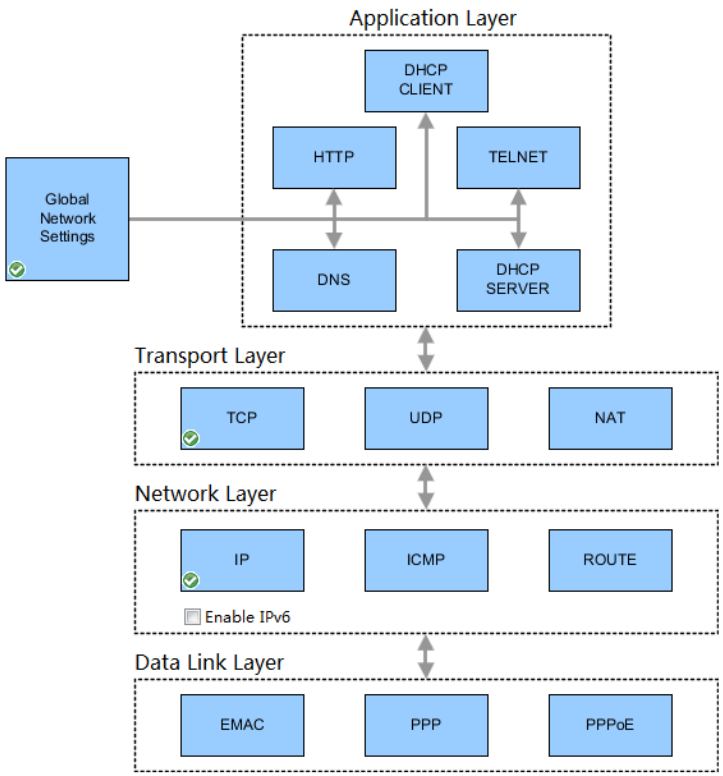

图 9. TI-RTOS 实时系统网络功能配置图

尽管本设计直接简单设定 IP（Internet Protocol）参数，即手动配置本地 IP 和网关，但是用户可以设定自动获取 IP 并开启 DNS（Domain Name System）服务。

2.2.3 文件系统配置

文件系统是操作系统在存储设备上组织文件的方法。它负责为用户建立，写入、读出、修改、转储文件，控制文件存取，当用户不再使用时撤销文件等。TI-RTOS 提供 FatFS 文件系统来进行文件访问。FatFS 是一个免费的第三方 FAT 文件系统，专为嵌入式系统设计，能够方便开发人员操作 SD 卡及 USB 等存储设备。本设计就是用该文件系统进行 U 盘文件的数据处理。

如图 6 所示，当有外置存储时，可以选配文件系统功能，例如前文 2.1.2 中固件文件的处理实际上就是调用了 FatFS 文件系统。本设计中使用了 U 盘做大数据存储扩展，用户也可以改用 SD 卡做数据存储，无论使用哪一种存储机制，都需通过文件系统进行访问。

2.2.4 多任务配置

TI-RTOS 操作系统支持多任务实时调度，可以采用动态或静态的方式进行任务创建和参数设置。本设计中主要实现了两个用户级任务，一个是串口调试任务，另一个是互联网升级任务，未创建其它子任务。图 8 展示了实时系统任务特别是互联网升级任务在操作系统中的正常运行流程。

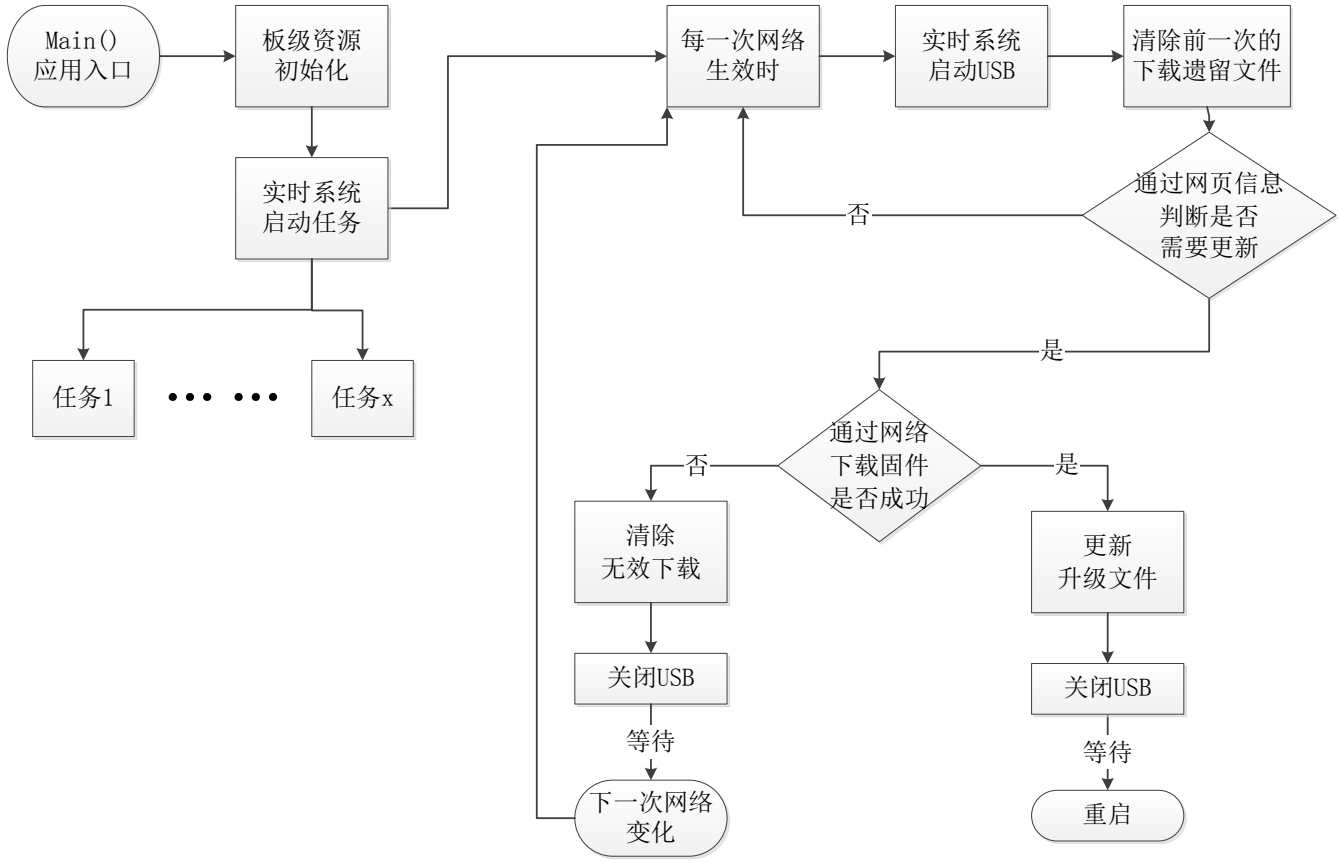

图 10.　　互联网升级任务软件流程图

2.3　本地升级

本设计中的本地升级是指通过 Bootloader 读取 U 盘中的程序固件，更新到 MCU 中的片内程序空间。Bootloader 中 Flash 升级过程的逻辑控制如图 9 所示：

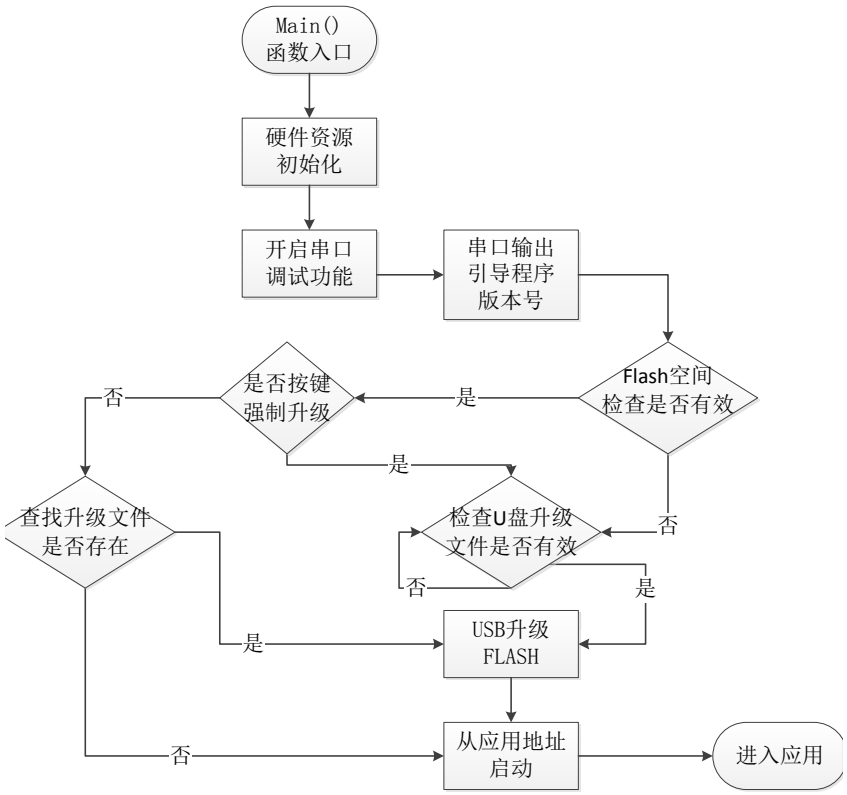

图 11. Bootloader 更新程序软件流程图

TI 官方的 TivaWare™ Boot Loader 示例代码针对 USB 升级提供了封装好的 API 接口，可以直接使用。除此之外，TI 还提供了针对 SD 卡、SPI 存储器等方式的程序升级接口。

3 演示安装

3.1 硬件配置

本设计使用一台 PC 电脑作为 HTTP 服务器和调试台，通过网线连接目标开发板，实现 HTTP 测试。开发板接有 16G U 盘一个作为外部存储设备。PC 机通过 JTAG 烧写调试开发板，并可以通过串口监视系统运行情况。

3.2 软件配置

由于本设计中的可升级固件镜像包含实时系统和自定义应用两部分，将由 Bootloader 引导启动，即作为二次加载程序，无法直接通过 CCS 在线调试，所以加入 UART 模块实现简单的串口打印来监视系统运行情况，方便开发人员进行功能测试。开发板使用串口 0，整个开发调试运行在 windows 环境下。

PC 机上需要运行 HTTP Server 服务，并提供 Web 网页资源。HTTP 服务使用第三方软件 everything 完成服务端架设，本文自行设计了网页页面及其数据存放。远端服务需要先行启动，本设计中服务器地址为 192.168.8.1，服务端口选用 8888。如所示：

图 12.　HTTP 服务器网页示图

网页上显示最新的固件版本号，例如图中的 2.0.0，以及最新固件地址，例如图中 HTTPUPFW.BIN 的超链接。版本信息及最新发布的固件升级文件可供用户查询和下载。

4　演示运行

开发板的初始运行版本设置为 1.0.0 版本固件，服务端存放 2.0.0 版本固件。开发者可以通过串口调试台看到系统工作状态。从服务器端正常下载固件文件后，U 盘中会重新存入临时文件 HTTPUPFW.BIN 和待升级固件 FIRMWARE.BIN。系统重启后，会删除过时的 HTTPUPFW.BIN 文件。

本文的打印功能设计非常简单，打印内容没有做特殊处理，打印信息都很简短。串口作为功能打印窗口，启动阶段和系统运行阶段的串口参数设置相同。参数如下：波特率为 115200bps；8 位数据位；无校验位；1 位停止位；无流控。此类串口助手软件工具非常多而且常见，此处不做赘述。

系统运行参数设置和显示结果图 11 所示：

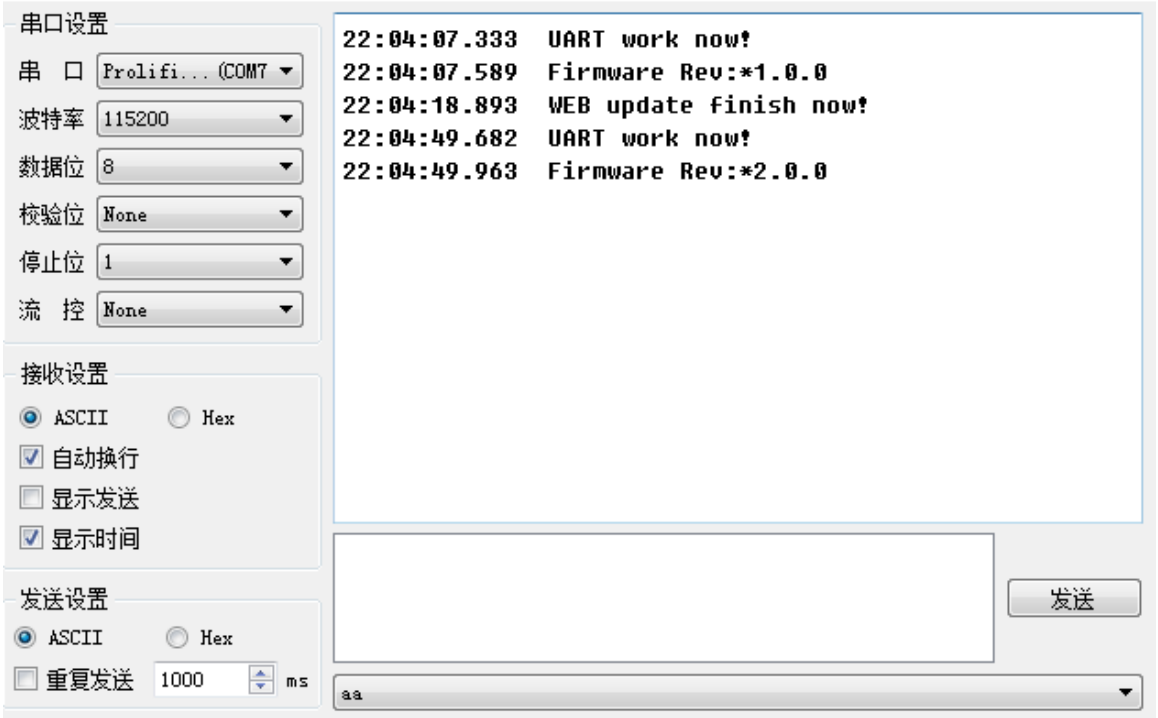

图 13. 系统正常运行串口打印信息图

刚上电时，串口会打印信息，显示系统已经开始工作，随后打印当前软件版本号 1.0.0。一旦网络有效，程序将执行远程升级任务。远程升级后，按键重启开发板，本地设备会运行新的固件，串口打印出新的软件版本号为 2.0.0，测试结果表明系统固件已被更新且运行正常。

5 软件及其资源

软件代码分为两个工程：Bootloader 工程和 TI-RTOS 工程。整个开发环境依赖两个 TI 发布的开发包，本设计使用版本如下：

“SW-TM4C-2.1.1.71.exe”--TivaWare_C_Series-2.1.1.71

“tirtos_tivac_setupwin32_2_14_00_10.exe”--tirtos_tivac_2_14_00_10

如需获取软件代码，请访问网址： 。

5.1 Bootloader 工程

该工程用于引导、升级 TI-RTOS 系统及其应用程序。 由于该代码是启动代码，所以可以通过 CCS6 进行在线调试和烧写，代码列表如下图所示：

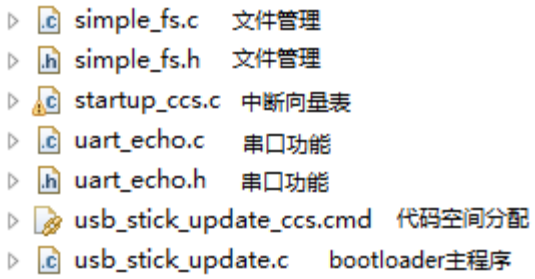

图 14. Bootloader 功能代码列表图

用户可以通过 startup_ccs.c 添加自定义的中断及其中断处理函数。

usb_stick_update.c 通过宏定义指明了系统应用启动的地址，用户可修改：

#define APP_START_ADDRESS 0x8000 此地址必须与 TI-RTOS 工程中的地址相同。

usb_stick_update.c 代码文件中的“int main(void)”是 Bootloader 工程的入口函数，在调用用户定义的功能函数之前应当对所需设备进行初始化，需要进行中断处理的功能，应在 startup_ccs.c 中进行中断处理函数的定义和注册。

5.2 TI-RTOS 工程

该工程用于生成包含应用程序的 RTOS 二进制文件 FIRMWARE.BIN。由于该程序被 Bootloader 引导执行，所以不可以直接通过 ccs6 进行在线调试，代码列表如下图所示：

图 15. 互联网数据下载功能代码列表图

EK_TM4C1294XL.cmd 文件中定义了本应用的启动地址

#define APP_BASE 0x00008000 此地址必须与 Bootloader 工程中的地址相同。

usbfile.c 实现了 USB 文件系统中的文件操作，用户可以根据需要自定义文件名和版本号。

```
static const char downFileUsb[] = "fat:"STR(USB_DRIVE_NUM)":HTTPUPFW.BIN";  //从网页下载的更新文件.
static const char FWUsb[] = "fat:"STR(USB_DRIVE_NUM)":FIRMWARE.BIN";        //用于升级的程序文件.
char fwNowRev[9] = "2.0.0";   //本应用软件版本号.
```

usbfile.c 中的 int upFW(void)函数完成了 HTTPUPFW.BIN 文件到 FIRMWARE.BIN 文件的转换，用户可以自定义此处功能，如特定的解密/解压功能。

webfile.c 实现了通过 HTTP GET 请求获取更新文件，用户可以根据自身需要修改宏定义：

```
#define IP    "192.168.8.1"                    //远端 IP 地址
#define PORT  8888                             //端口号
#define HOSTNAME "192.168.8.1:8888"            //主机地址字符串
#define REQUEST_URI_INDEX "/index.html"    //web 主页
#define REQUEST_URI_BIN "/index_files/HTTPUPFW.BIN"      //待下载文件路径
```

需要注意的是，由于本程序被引导启动，所以需要对操作系统的特点文件也进行相应地址修改，Debug/configPkg/linker.cmd 中必须修改相应中断向量表的偏移地址，详细如下所示：

```
SECTIONS
{
   .bootVecs:  type = DSECT
   .vecs: load > 0x20000000
   .resetVecs: load > 0x00008000  //此地址应与 EK_TM4C1294XL.cmd 中的 APP_BASE 一致

   xdc.meta: type = COPY
   xdc.noload: type = COPY
}
```

除此之外，httpget.cfg 用来对 TI-RTOS 实时操作系统的各种功能及其参数进行图形化配置或文本编辑配置。本设计参考如下，用户可自行修改：

```
/* ================ Driver configuration ================ */
var TIRTOS = xdc.useModule('ti.tirtos.TIRTOS');
TIRTOS.useEMAC = true;                         //开启网络驱动
TIRTOS.useGPIO = true;                         //开启 GPIO 驱动
TIRTOS.useUART = true;                         //开启 UART 驱动
TIRTOS.useUSBMSCHFatFs = true;                 //开启文件系统
TIRTOS.libType = TIRTOS.LibType_Instrumented;
Semaphore.supportsEvents = true;

/* ================ HTTP configuration ================ */
var Http = xdc.useModule('ti.net.http.Http');      //使用 HTTP 功能
Http.networkStack = Http.NDK;
Ip.autoIp = false;                             //关闭自动 IP 获取
Ip.address = "192.168.8.2";                    //设定本地 IP 地址
Ip.mask = "255.255.255.0";                     //设定子网掩码
Ip.gatewayIpAddr = "192.168.8.1";              //设定网关地址

/* ================ Task configuration ================ */
var task0Params = new Task.Params();
task0Params.instance.name = "echo";
task0Params.stackSize = 2048;                  //设定任务栈大小
Program.global.echo = Task.create("&echoFxn", task0Params);      //启动串口任务
```

6 引用

1. Tiva™ TM4C1294NCPDT Microcontroller DATA SHEET (SPMS386)
2. EK-TM4C1294XL Rev D Design Files (EAGLE) (SPMR241)
3. Tiva™ C Series TM4C1294 Connected LaunchPad Evaluation Kit EK-TM4C1294XL User's Guide (*SPMU365B*)
4. TI-RTOS 2.14 for TivaC Getting Started Guide（SPRUHU5C）
5. TI-RTOS 2.14 User’s Guide（SPRUHD4J）
6. SYS/BIOS (TI-RTOS Kernel) v6.41 User's Guide（SPRUEX3O）
7. TI Network Developer's Kit (NDK) v2.24 User's Guide（SPRU523I）
8. USER’S GUIDE TivaWare™ Boot Loader（SW-BOOTLDR-UG-1.0）
9. USER’S GUIDE TivaWare™ Peripheral Driver Library（SW-TM4C-DRL-UG-2.1.1.71）
10. ROM USER’S GUIDE Tiva™ C Series TM4C129x（SPMU363）

重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改， 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内，且 TI 认为 有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定，否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应 用相关的风险，客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予 的直接或隐含权限作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息，不能构成从 TI 获得使用这些产品或服 务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可，或是 TI 的专利权或其它 知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分，仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况 下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时，如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分，则会失去相关 TI 组件 或服务的所有明示或暗示授权，且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意，尽管任何应用相关信息或支持仍可能由 TI 提供，但他们将独力负责满足与其产品及在其应用中使用 TI 产品 相关的所有法律、法规和安全相关要求。客户声明并同意，他们具备制定与实施安全措施所需的全部专业技术和知识，可预见 故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因 在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中，为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特 有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此，此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III（或类似的生命攸关医疗设备）的授权许可，除非各方授权官员已经达成了专门管控此类使 用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同 意，对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用，其风险由客户单独承担，并且由客户独 力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品，这些产品主要用于汽车。在任何情况下，因使用非指定产品而无法达到 ISO/TS16949 要求，TI不承担任何责任。

	产品		应用
数字音频	www.ti.com.cn/audio	通信与电信	www.ti.com.cn/telecom
放大器和线性器件	www.ti.com.cn/amplifiers	计算机及周边	www.ti.com.cn/computer
数据转换器	www.ti.com.cn/dataconverters	消费电子	www.ti.com/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com/energy
DSP - 数字信号处理器	www.ti.com.cn/dsp	工业应用	www.ti.com.cn/industrial
时钟和计时器	www.ti.com.cn/clockandtimers	医疗电子	www.ti.com.cn/medical
接口	www.ti.com.cn/interface	安防应用	www.ti.com.cn/security
逻辑	www.ti.com.cn/logic	汽车电子	www.ti.com.cn/automotive
电源管理	www.ti.com.cn/power	视频和影像	www.ti.com.cn/video
微控制器 (MCU)	www.ti.com.cn/microcontrollers		
RFID 系统	www.ti.com.cn/rfidsys		
OMAP应用处理器	www.ti.com/omap		
无线连通性	www.ti.com.cn/wirelessconnectivity	德州仪器在线技术支持社区	www.deyisupport.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265