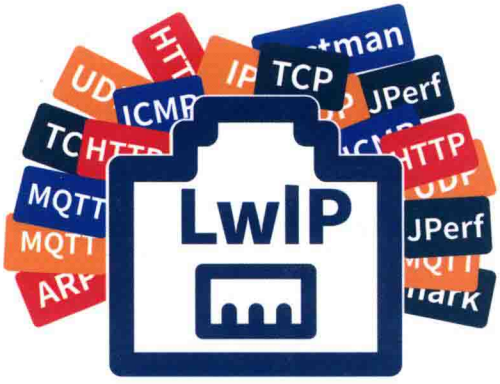
HZ BOOKS
华章科技

深入剖析LwIP中的各种协议和应用。

以LwIP源码为核心，结合经典的云-管-端物联网应用实例，由浅入深，讲解LwIP技术和应用开发。配套野火STM32 M4/M7系列开发板，提供完整源代码，极具操作性。

野火

野火嵌入式系列



0110.0110.0110.0110

LwIP应用开发实战指南

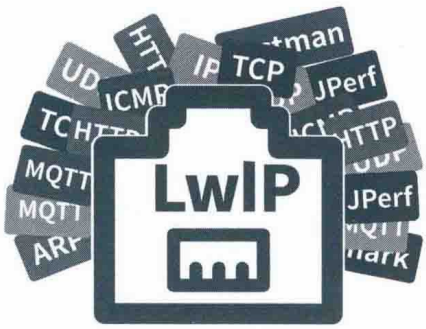
基于STM32

刘火良 杨森 编著

机械工业出版社
China Machine Press

野火

野火嵌入式系列



0110.0110.0110.0110

LwIP应用开发实战指南

基于STM32

刘火良 杨森 编著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

LwIP 应用开发实战指南：基于 STM32 / 刘火良，杨森编著．—北京：机械工业出版社，2019.9
（电子与嵌入式系统设计丛书）

ISBN 978-7-111-63582-6

I. L…　II. ①刘…　②杨…　III. 微控制器－系统开发－指南　IV. TP332.3-62

中国版本图书馆 CIP 数据核字（2019）第 188830 号

LwIP 应用开发实战指南：基于 STM32

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：赵亮宇　　责任校对：李秋荣
印　　刷：三河市宏图印务有限公司　　版　　次：2019 年 9 月第 1 版第 1 次印刷
开　　本：186mm×240mm　1/16　　印　　张：28.75
书　　号：ISBN 978-7-111-63582-6　　定　　价：119.00 元

客服电话：（010）88361066　88379833　68326294　　投稿热线：（010）88379604
华章网站：www.hzbook.com　　读者信箱：hzit@hzbook.com

作者简介

刘火良　野火电子创始人，嵌入式软件工程师，长期从事STM32软硬件开发，网络教程“零死角玩转STM32”的作者，《STM32库开发实战指南》的合著者。

杨森　野火电子合伙人，嵌入式软件工程师，专注于STM32和周边软件开发。合著有《STM32库开发实战指南》，已累计印刷十余次，广受STM32开发者的喜爱。

阅读建议

全书内容循序渐进，不断迭代，前一章是后一章的基础，尽量不要跳跃性阅读，从理解原理到源码的实现是个很重要也很枯燥的过程，一定要耐心学习。此外，在学习的时候务必做到两点：一是不能一味地看书，要把代码和书结合起来学习，一边看书一边调试代码；二是在学完每一章之后，将配套例程重写一遍，做到举一反三，真正理解。

本书电子版和配套的源码均可到野火论坛（www.fireBBS.cn）下载，欢迎各位读者积极参与交流分享。

前　言

如何学习本书

本书围绕 LwIP 2.1.2 版本源码讲解 TCP/IP 网络协议栈的基本知识，带领读者进入网络的世界。无论你是学生、嵌入式开发者还是物联网开发者，都可以从本书中学习到网络的相关知识，了解网络协议栈的处理思想。

本书将深入分析网络协议栈的原理与实现过程，涉及 ARP、IP、ICMP、TCP、UDP、HTTP、MQTT 等协议，还将深入讲解 LwIP 中内存管理、pbuf 数据包、网卡接口管理的原理与实现，并详细介绍 LwIP 的移植过程，读者可以将其移植到无操作系统 / 有操作系统的环境中使用。

除此之外，本书还通过理论和实践相结合，使用 LwIP 接入当前较大的几个云平台，如阿里云、百度云、OneNET，以使读者熟练掌握接入方法。在本书中，读者可以熟练掌握 LwIP 中 Netconn API 与 Socket API 的使用方式，并且掌握多种网络调试工具，如 Wireshark、Postman 以及 MQTT.fx 等的使用方法。

全书内容循序渐进，不断迭代，建议读者在学习的时候做到两点：一是不能一味看书，要把代码和理论结合起来学习，一边看书，一边调试代码。通过执行每一个程序，检验程序的执行流程和执行效果与自己想的是否一致，不断总结经验；二是在每学完一章之后，必须将配套的例程重写一遍（切记不要复制，即使是一个符号），举一反三，确保真正理解。

推荐阅读

- LwIP 官方源代码
- 《STM32 库开发实战指南》（已由机械工业出版社出版多个版本）
- 《FreeRTOS 内核实现与应用开发实战指南：基于 STM32》（已由机械工业出版社出版，ISBN 978-7-111-61825-6）

本书的技术论坛

如果在学习过程中遇到问题，可以到野火电子论坛（www.firebbs.cn）发帖交流，开源共享，共同进步。

鉴于水平有限，本书难免有错漏之处，热心的读者也可把勘误发送到论坛以便我们加以改进。祝你学习愉快，LwIP 的世界，野火与你同行！

目　录

第 1 章
网络协议概述

1.1 常用网络协议

互联网为人类社会带来巨大变革，几乎改变了人们生活的方方面面。互联网通信的本质是数字通信，任何数字通信都离不开通信协议，通信设备只有按照约定的、统一的方式去封装和解析信息，才能实现通信。互联网通信所要遵守的众多协议，被统称为 TCP/IP。

TCP/IP 是一个协议族，包含众多协议。对于网络应用开发人员，可能听到得更多的是其中的应用层协议，比如 HTTP（Hyper Text Transfer Protocol，超文本传输协议）、FTP（File Transfer Protocol，文件传输协议）、MQTT（Message Queuing Telemetry Transport，消息队列遥测传输）等。

HTTP 的应用最为广泛。比如大家日常使用计算机时的一个常规操作：打开计算机，再打开浏览器，输入网址，最后按下 Enter 键，这一刻你就开启了 HTTP 通信。HTTP 工作于 < 客户端 - 服务端 > 架构之上（服务端也称为服务器端，除特别说明外，本书出现的“服务端”即为“服务器端”），浏览器作为 HTTP 客户端，通过 URL 向 HTTP 服务端（即 Web 服务器）发送所有请求。Web 服务器根据接收到的请求向客户端发送响应信息。借助这种浏览器和服务器之间的 HTTP 通信，我们能够足不出户地获得来自世界各地的信息。另外，网页不仅仅是大型服务器的专利，在物联网风潮盛行的今天，许多随处可见的小型设备（如空调、冰箱、插座、路由器等）都内嵌网页，在物理链路畅通的情况下，用户可以用手机、平板电脑上的浏览器随时随地监控这些设备。

FTP 是工作在应用层的网络协议，使得主机间可以共享文件，用于在两台设备之间传输文件（双向传输）。它也是一个客户端 - 服务端框架系统。用户可以通过一个支持 FTP 的客户端程序，连接到远程主机上的 FTP 服务端程序，通过客户端程序向服务端程序发出命令，服务端程序执行用户所发出的命令，并将执行的结果返回客户端。FTP 除了支持基本的文件上传 / 下载功能外，还支持目录操作、权限设置、身份验证机制，许多网盘的文件传输功能都是基于 FTP 实现的。

在物联网发展的初期，物联网场景中的设备使用何种应用层协议进行通信一直是备受争议的话题。很多开发人员习惯了网页开发模式，于是经常选择 HTTP 作为通信方式。使用 HTTP 有以下不利因素：HTTP 是一种同步协议，设备需要等待服务器的响应才可以进行下

一步的工作，然而在设备数量多、网络不可靠的场景下，实现同步通信很困难；HTTP 是单向的，设备只能主动向服务器发出数据，无法被动接收来自网络的数据，这不适用于实时控制的场合；HTTP 是有许多帧头和规则的重量级协议，在设备中实现需要耗费大量的系统资源。基于上述形势，MQTT 和 COAP 等轻量级、异步的通信协议便得到了物联网设备开发商的青睐，尤其是 MQTT。

MQTT 协议是 IBM 公司于 1990 年设计并推出的一款通信协议，于 2014 年正式成为一个 OASIS 开放标准。近年来，MQTT 的应用呈现爆炸性的增长势头，大有一统物联网的趋势。另外，MQTT 在物联网以外的其他领域也得到了广泛的应用，比如许多公司在制作手机 APP 时，会使用 MQTT 来实现消息推送、即时聊天等功能。

嵌入式设备接入互联网的需求越来越大，有以下几点原因：

1）近些年，各种带网络接入功能的 MCU、SoC 层出不穷，开源轻量的 TCP/IP 协议栈日趋成熟和完善，云平台的市场越来越繁荣，这些因素大大降低了嵌入式设备的入网成本，也为许多资源受限的低端设备接入互联网提供了可能。

2）“物联网 +”的风潮日渐盛行，设备能够被远程监控，这一点已经成为许多产品的技术要求。

3）人们对于设备“智能性”的需求越来越高，当今热门的大数据、图像处理、语音识别、机器学习等功能都可以集成在云端，成为云平台能提供的服务。终端设备大多是计算、存储能力有限的设备，这些设备如果想要获取“智能”，最便捷的办法就是接入云平台，利用各项云服务。

互联网的基础就是 TCP/IP。TCP/IP 是一个非常复杂的协议族，即使我们能把它的设计思想和实现原理都解释清楚，你也不一定有时间和精力去学习它，所以本书的写作重点不在于对 TCP/IP 的解读，而在于对它的应用。另外，TCP/IP 的复杂性也决定了它使用起来没那么简单，即使我们只关注应用开发，也依然需要对它的许多概念和设计思想有所了解，才能编写出正确、高效、健壮的应用程序。

希望能借此书，让嵌入式开发工程师以浓厚的兴趣和清晰的视野，搭上物联网发展的快车。

1.2 网络协议的分层模型

TCP/IP 是众多网络协议的集合，包括 ARP、IP、ICMP、UDP、TCP、DNS、HTTP、FTP、MQTT 等。这些协议按照功能，可以划分为几个不同的层次，如图 1-1 所示。我们在 1.1 节中介绍的 HTTP、FTP、MQTT 隶属于应用层。那么 TCP/IP 为什么需要分层，分层的依据又是什么呢？

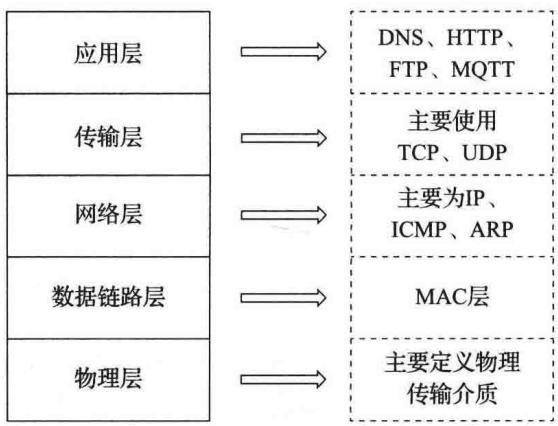


图 1-1 TCP/IP 的分层

TCP/IP 协议栈中不同协议所完成的功能是不一样的，某些协议的实现要依赖于其他协议，依据这种依赖关系，可以将协议栈分层。在图 1-1 中，低层协议为相邻的上层协议提供服务，是上层协议得以实现的基础。

其中，物理层（PHY）规定了传输信号所需要的物理电平、介质特征。数据链路层（MAC）规定了数据帧能被网卡接收的条件，最常见的方式是利用网卡的 MAC 地址，发送方会在欲发送的数据帧的首部加上接收方网卡的 MAC 地址信息，接收方只有监听到属于自己的 MAC 地址信息后，才会去接收并处理该数据。每台网络设备都应该有自己的网络地址，网络层规定了主机的网络地址该如何定义，以及如何在网络地址和 MAC 地址之间进行映射，即 ARP，还实现了数据包在主机之间的传递，而一台主机内部可能运行着多个网络程序。传输层可以区分数据包是属于哪一个应用程序的，可以说传输层实现了数据包端到端的传递。另外，数据包在传输过程中可能会出现丢包、乱序和重复的现象，网络层并没有提供应对这些错误的机制，而传输层可以解决这些问题，如 TCP。应用层以下的各层完成了数据的传递，应用层则决定了你如何应用和处理这些数据。之所以会有许多应用层协议，是因为互联网中传递的数据种类很多、差异很大、应用场景十分多样。

1.3　协议层报文间的封装与拆封

本书的后面章节会对 TCP/IP 协议栈中的每层协议进行分析和讲解。在这里，我们以图 1-2 简单解释一下在数据的发送和接收过程中，TCP/IP 都做了什么。

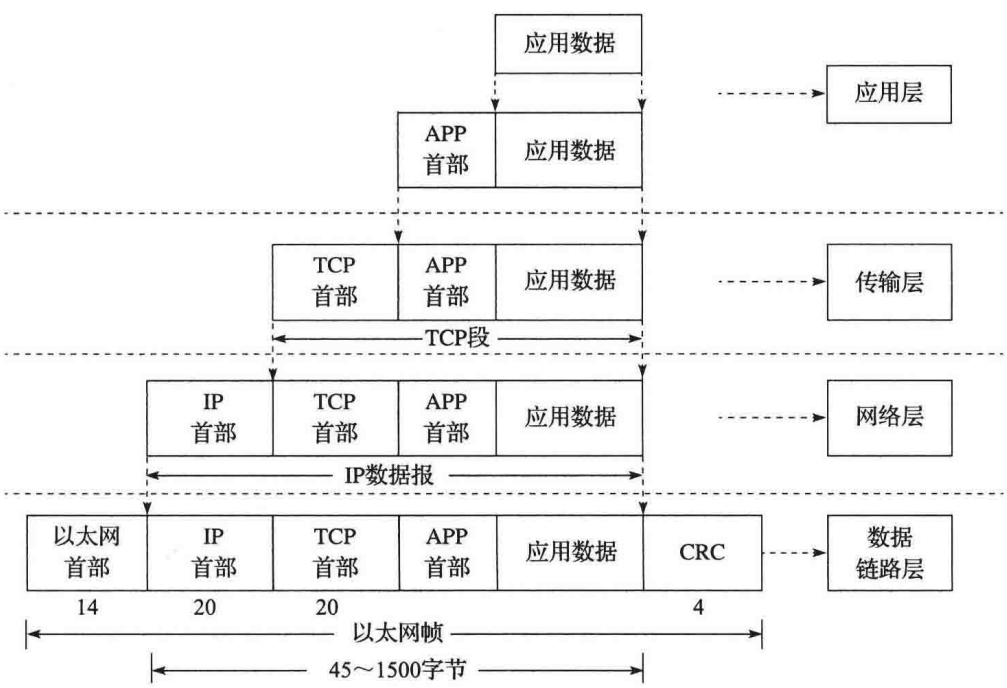


图 1-2　TCP/IP 协议栈各层的报文封装与拆封