# ZigBee 无线传感器网络的设计及应用

## 中文摘要

无线传感器网络是当前国际上备受关注的、由多学科高度交叉的新兴前沿研究热点。无线传感器网络综合了传感器技术、无线通讯技术和计算机技术等，具有信息采集、传输和处理的能力。低成本、低功耗、应用简单的 ZigBee 协议的诞生为无线传感器网络及大量基于微控制的应用提供了互联互通的国际标准，也为这些应用及相关产业的发展提供了有力的契机。

目前，国内主要以 ZigBee 技术的应用研究为主，尚没有对外公布的协议栈。多数应用都是以 Freescale 或 Microchip 公司所提供的开发套件为基础平台，也有少数有自己的硬件平台，但软件上仍然是在 Freescale 或 Microchip 公司所提供的底层协议 API 接口基础之上来开发实现的。

本文首先介绍了无线传感器网络和 ZigBee 技术的相关基础知识，然后在现有的 ZigBee 硬件方案中选择了 Freescale 公司提供的解决方案：MC9S08GB60 和 MC13192，并以此方案为背景设计开发了 MT-ZigBee 硬件平台。接着在深入分析 ZigBee 协议规范的基础上，对 ZigBee 协议物理层、MAC 层和网络层功能的设计与实现作了详细介绍。作为对 MT-ZigBee 硬件平台和协议栈可行性的测试与验证，论文的最后以农业大棚为实际的应用对象，组织了一个较为简单的应用实例，验证了 MT-ZigBee 硬件平台和协议栈的可用性。

本文所设计实现的 MT-ZigBee 硬件平台与简化的 ZigBee 协议栈，对于 ZigBee 技术和无线传感器网络的应用研究具有一定的参考价值和实际意义，为 ZigBee 无线技术在工业、农业、家庭建筑和环境监测等方面的进一步应用提供了相关的软硬件基础平台，同时也为对 ZigBee 协议本身的研究与改进提供了相应的工作基础。

关键词：无线传感器网络，WSN，ZigBee，IEEE 802.15.4，MC13192

作者：刘 辉

指导老师：王宜怀

# Design and Application of ZigBee Wireless Sensor Network

## ABSTRACT

Wireless sensor network which is the cross of many disciplines has become the new researching hotspot focused by many countries. It has the abilities of information collection, transmission and processing as it includes various technologies, such as sensor, wireless communication and computation. The appearance of the ZigBee protocol with low cost, low power consumption and simple application offers the international standard to wireless sensor network as well as a mass of applications based on micro-control. At the same time it offers the chance to these applications and the development of related industries

At present, domestic researchers concentrate their attentions on the ZigBee's applications, and there is no public protocol stack. Most applications are based on the development platform offered by Freescale or Microchip company. However, a few of applications have their own hardware, but the software is still based on the API interface provided by Freescale or Microchip.

This paper firstly introduces the related knowledge about the wireless sensor network and ZigBee technology, then gives a ZigBee's hardware scheme offered by Freescale and a ZigBee's hardware platform developed on the scheme. After reading and analyzing the ZigBee protocol, this paper describes the design of the PHY, MAC and NWK layers in detail. At last, it validates the feasibility of the hardware platform called MT-ZigBee and the protocol stack by the practical application of the agriculture glasshouse.

This research provides a valuable reference to the application of the ZigBee and wireless sensor network. The hardware and stack offer the basic platform to the applications of ZigBee on industry, agriculture, family, environmental monitor and so on, and also supply the basis for the study of the ZigBee protocol.

**Key words:** wireless senor network, WSN, ZigBee, IEEE 802.15.4, MC13192

Written by Liu Hui
Supervised by Wang Yihuai

# 苏州大学学位论文独创性声明及使用授权的声明

## 学位论文独创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含其他个人或集体已经发表或撰写过的研究成果，也不含为获得苏州大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

研究生签名：刘将 日 期：2007.5.31

## 学位论文使用授权声明

苏州大学、中国科学技术信息研究所、国家图书馆、清华大学论文合作部、中国社科院文献信息情报中心有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权苏州大学学位办办理。

研究生签名：刘将 日 期：2007.5.31

导师签名：[illegible] 日 期：2007.5.31

# 第一章 绪论

作为将对二十一世纪产生巨大影响的技术之一，无线传感器网络(wireless sensor network，WSN)是近几年来国内外研究的热点，无线传感器网络引起了世界上许多国家军界、学术界和工业界的高度重视[1][2]，其应用前景十分广阔。目前，在无线传感器网络中，短距离、低成本、低功耗的 ZigBee 技术是无线通信应用的首选技术之一。

本章首先介绍了无线传感器网络和 ZigBee 技术的相关背景，随后给出本文的工作内容、课题意义及论文结构。

## 1.1 无线传感器网络简介

微电子技术、计算技术和无线通信等技术的深入研究推动了低功耗、多功能传感器的快速发展，使其在微小体积内能够集成信息采集、数据处理和无线通信等多种功能[3]。

在概念上，无线传感器网络是指由大量无处不在的，具有通信与计算能力的微小传感器节点密集分布在监控区域内而构成的根据环境自主完成指定任务的自治测控网络系统[4][5]。

### 1.1.1 无线传感器网络的特点

WSN 是一个由几十到上万个节点组成的、采用无线通信方式的、动态组网的多跳对等网络，其自身有显著的特征[6][7]。

(1) 节点数量大，但通信距离、数据传输量有限

为了对一个区域执行监测任务，往往有成千上万个传感器节点空投到该区域，节点数量很大，但节点与节点之间的直接通信距离一般在几十米范围内，并且通信的数据量有限，这样在通信速率上要求不高，一般在 1~100kbps 即可满足要求[8]。

(2) 传感器节点体积小、成本低、计算能力有限[9]

为了实际应用的方便，WSN 节点在硬件体积上一般设计的比较小，便于实施安装；WSN 网络的节点数量庞大，因而在单节点上成本不可以太高；由于成本的控制，传感器节点的计算能力、程序空间和内存空间比普通的计算机要弱很多，这一点决定了在系统设计中，协议层次不能太复杂。

(3) 网络的动态拓扑、多跳路由[10]

无线传感器网络是一个动态的网络。一个节点可能会因为电池能量耗尽或其他故障从运行网络中退出；也可能由于工作的需要而被添加到网络中。这些都会使网络的拓扑结构随时发生变化，因此网络应该具有动态拓扑组织功能。由于网络中节点通信距离有限，如果希望与其射频覆盖范围之外的节点进行通信，则需要通过中间节点进行路由。

(4) 以数据为中心

在无线传感器网络中，人们只关心某个区域的某个观测指标的值,而不会去关心具体某个节点的观测数据。以数据为中心的特点要求传感器网络的设计必须以感知数据管理和处理为中心。

### 1.1.2 无线传感器网络的发展历程

无线技术和传感器技术的不断进步是无线传感器网络发展的直接推动力。无线传感器网络的发展是由传感器网络开始的，从年代上，传感器网络大致可以分为四代[11]，如图 1-1 所示。

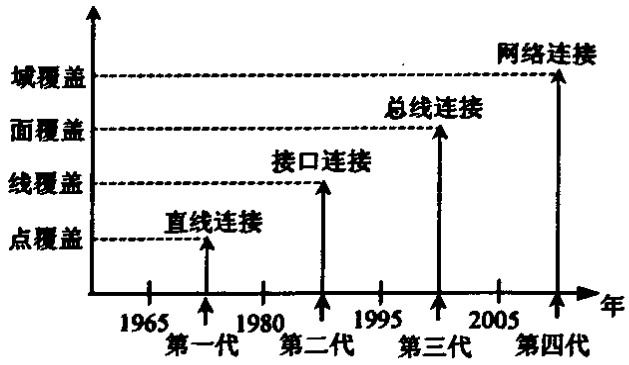


图 1-1 传感器网络的发展历程

早在上世纪 70 年代，就出现了将传感器采用点对点的通信方式，这样所连接的传感器便构成传感器网络的雏形，我们可以把它归之为第一代传感器网络。

20 世纪 80 年代，随着相关学科的不断发展和进步，传感器网络同时具有了获取多种信息的综合处理能力，并可通过串行或并行等接口与控制器相关联，组成了有信息综合和处理能力的传感器网络，这是第二代传感器网络。

从上世纪末开始，现场总线技术开始应用于传感器网络，人们用其组建智能化传感器网络，大量多功能传感器被运用，第三代传感器网络逐渐形成[12]。

第四代传感器网络采用大量的具有多功能多信息获取能力的传感器，在通信方式上采用自组织无线接入，从而构成了现代意义的无线传感器网络。无线传感器网络起始于 20 世纪 90 年代末期，最早用于军事上战场信息的收集。最早的代表性论述出现在 1999 年，题为“传感器走向无线时代”[11]。随后在美国的移动计算和网络国际会议上，提出无线传感器网络是下世纪面临的一个发展机遇。2002 年 10 月 24 日，美国英特尔公司发布了“基于微型传感器网络的新型计算发展规划”[13]。在美国自然科

学基金委员会(natural science foundation,NSF)的推动下，加州大学伯克利分校、麻省理工学院、康奈尔大学、加州大学洛杉矶分校等大学研究了无线传感器网络的基础理论和关键技术[14]。2003 年，美国《技术评论》杂志在论述未来新兴十大技术时，无线传感器网络被列为第一项未来新兴技术。同年，美国《商业周刊》未来技术专版，在论述四大新技术时，无线传感器网络也被列入其中。美国《今日防务》杂志更认为无线传感器网络的应用和发展，将引起一场划时代的军事技术革命和未来战争的变革。2004 年 IEEE Spectrum 杂志发表一期专集：传感器的国度，论述了无线传感器网络的发展和可能的广泛应用[11]。

在国内，现代意义的无线传感器网络及其应用研究首次正式出现在 1999 年中国科学院《知识创新工程试点领域方向研究》的“信息与自动化领域研究报告”中，并作为该领域提出的 5 个重大项目之一。随着知识创新工程试点工作的深入，2001 年中国科学院依托上海微系统所成立了微系统研究与发展中心，中心在无线传感器网络方向陆续部署了若干重大研究项目和方向性项目。2005 年初，中国科学院召开了关于无线传感器网络技术的研讨会，共商无线传感器网络下一步的工作。《国家中长期科学和技术发展规划纲要(2006-2020)》[15]在支持的重点领域及其优先主题“信息产业及现代服务业”中列入了“传感器网络及智能信息处理”，并在前沿技术中重点支持“自组织传感器网络技术”。我国国家自然科学基金 2005 年将网络传感器中的基础理论和关键技术列入计划[16]，2006 年国家自然科学基金将水下移动传感器网络的关键技术列为研究重点[17]。

## 1.2 ZigBee 技术简介

目前在 WSN 的无线通信方面可以采用的主要有 ZigBee、篮牙、WiFi 和红外等技术[18]。其中红外技术的实现和操作相对简单，成本低廉，但红外光线直线传输、易受遮挡，可移动性差，只支持点对点视距连接，无法灵活地构建网络[19]；篮牙技术是工作在 2.4GHz 频段的无线技术，目前在计算机外设方面应用较广泛，但由于其协议本身较复杂、开发成本高、节点功耗大等缺点，从而限制了其在工农业方面的进一步推广[20]；WiFi 技术的通信速率为 11Mbps，通信距离为 50~100 米，适合于多媒体的应用，但其本身实现成本高，功耗大，安全性能低，从而在 WSN 中应用较少；ZigBee 技术以其经济、可靠、高效等优点在 WSN 中有着广泛的应用前景。

ZigBee 技术是一种短距离、低复杂度、低功耗、低数据速率、低成本的双向无线通信技术或无线网络技术，是一组基于 IEEE 802.15.4 无线标准研制开发的有关组

网、安全和应用软件方面的通信技术[21][22]。ZigBee 的名字来源于蜂群使用的赖以生存和发展的通信方式，蜜蜂通过跳 ZigZag 形状的舞蹈来通知发现的新食物源的位置、距离和方向等信息，ZigBee 技术就是以此作为新一代无线通讯技术的名称[23][24]。

### 1.2.1 ZigBee 技术的特点

ZigBee 技术主要具有低速率、低功耗、低成本、时延短和高安全性等特点[25][26]。

低速率，ZigBee 技术的数据传输速率只有 20～250kb/s(2.4GHz)，40kb/s(915MHz)和 20kb/s(868MHz)的原始数据吞吐率，满足低速率传输数据的应用需求。

低功耗，ZigBee 节点在低功耗待机模式下，两节普通 5 号电池可使用 6～24 个月，免去了充电或者频繁更换电池的麻烦。

低成本，因为 ZigBee 数据传输速率低，协议简单，所以大大降低了开发成本，并且 ZigBee 协议免收专利费用。ZigBee 网络节点的硬件成本一般可控制在 1000 元以下。

时延短，ZigBee 的响应速度较快，一般从睡眠状态唤醒到工作状态只需要 15ms，节点连接进入网络只需要 30ms，进一步节约了能源。

安全性高，ZigBee 提供了三级安全模式。使用接入控制清单，防止非法获取数据以及采用高级加密标准(ASE-128)的对称密码，以灵活地确定其安全性。

### 1.2.2 ZigBee 技术的发展历程

随着通信技术的迅速发展，人们提出了在自身附近几米范围内通信的要求，这样就出现了个人区域网络（personal area network，PAN）和无线个人区域网络（wireless personal area network，WPAN）的概念。WPAN 网络为近距离范围内的设备建立无线连接，把几米范围内的多个设备通过无线方式连接在一起，使它们可以相互通信甚至接入 LAN 或 Internet。虽然蓝牙技术的出现缓解了 WPAN 范围内无线通信的压力，但由于其协议复杂、实现成本高，使其在进一步的推广中受阻。在这种情况下，协议简单、实现成本低的 ZigBee 技术应运而生。

2000 年 12 月，IEEE(美国电子和电气工程师协会)成立了 IEEE 802.15.4 工作组，致力于开发一种可应用在固定、便携或移动设备上的，低成本、低功耗和低速率的无线连接技术[27]。

2001 年 8 月，美国 HONEYWELL 等公司发起成立了 ZigBee 联盟，他们提出的 ZigBee 规范被确认为 IEEE 802.15.4 标准。

2002年，摩托罗拉、飞利浦和三菱等企业加盟ZigBee联盟，2006年，中国的华为公司也加入了该联盟。目前该联盟已有180多个成员企业，包括终端产品商、软件供应商和系统集成商。

2003年，IEEE 802.15.4标准获得通过，并在2004年12月推出了ZigBee技术规范1.0版本，正式对外公布。

2006年12月份，ZigBee联盟正式推出ZigBee的升级规范－ZigBee 2006[28]，也称为“增强型”ZigBee。

## 1.3 本文主要工作、课题意义及论文结构

无线传感器网络和ZigBee技术具有十分广阔的应用前景，已经成为国内外一个公认的新兴前沿热点研究领域，目前，国内大部分高校和科研单位已经开始了该领域的研究工作。本文将基于ZigBee技术的无线传感器网络作为研究对象具有重要的科研意义和实用价值。

### 1.3.1 本文主要工作

本文的最终目标是选择设计一个合适的ZigBee硬件平台，设计实现ZigBee技术相关的底层协议栈，为上层用户提供实用的接口，并通过一个具体的应用实例验证硬件平台和协议栈的可用性。因此本文的主要工作内容包括：

(1) 协议的分析、理解

① 收集IEEE 802.15.4和ZigBee v1.0协议规范；

② 仔细阅读、分析各层协议，充分理解其内容。

(2) MT-ZigBee硬件平台设计

① 平台方案选择，根据实验室现有的工作基础，在论文初期的可选方案中选择了Freescale公司所提供的ZigBee硬件平台方案；

② 根据主控芯片和射频芯片的相关参考资料，设计硬件原理图，绘制PCB电路板，并交由厂家生产；

③ 焊接、测试，完成MT-ZigBee硬件平台的搭建工作。

(3) ZigBee协议栈的设计实现

① 在底层驱动之上，实现信道设置、信道能量检测和物理数据包的收发等物理层功能；

② 以物理层为基础，有所裁减的设计实现了MAC帧的组织、信道冲突避免和

MAC 帧的收发等 MAC 层功能；

③ 进一步实现网络的组建、加入、退出和网络报文的路由以及网络报文的收发等网络层功能。

(4) 总体验证测试

在 MT-ZigBee 硬件平台和实现的协议栈基础上，选择农业大棚作为实际的应用对象，进一步验证软硬件的可用性。

### 1.3.2 课题意义

近几年来，采用 ZigBee 通信技术的无线传感器网络在国内有着较广泛的研究，但大部分的研究工作都不设计实现 ZigBee 底层协议栈和 2.4GHz 的硬件平台，目前在国内尚没有对外公布的 ZigBee 协议栈。据了解多数的研究工作大致可分为两类。

一类是对协议中某一点的具体深入研究，如对网络路由算法的研究与改进。这部分的研究工作是在已有的软硬件开发平台基础上进行的，研究人员对底层协议和硬件平台部分不需要去花费时间和精力。这些已有的软硬件平台基本上都是由 Freescale 或 Microchip 公司所提供的，国内也有少数公司供应开发平台，但软件上仍然是封装在国外公司提供的底层协议栈 API 接口之上的，并没有自己的协议栈，同时这样的开发套件也是价格不菲的。

另外一类则是 ZigBee 的应用研究。该类研究工作也有部分搭建自己的硬件平台，但软件上则是基于 Freescale 或 Microchip 公司所提供的 API 接口之上来应用开发的。

本文的主要工作就是在 Freescale 公司所提供的硬件方案基础上构建了 MT-ZigBee 硬件平台，同时有所裁减的设计实现了 ZigBee 的底层协议栈。本文的工作可以为 ZigBee 技术和无线传感器网络的进一步推广和研究工作提供相应的软硬件基础平台，同时也希望课题的研究可以为相关的研究与应用工作提供一定的参考价值。

### 1.3.3 论文结构

全文共分为八章，各章的内容安排如下：

第一章介绍了无线传感器网络和 ZigBee 技术的发展概况和特点。在此基础上，给出了毕业设计的核心内容、课题意义及论文结构；

第二章对 ZigBee 技术给出总体介绍，并简述了 ZigBee 协议的框架及分层结构，同时概述了各层的主要功能；

第三章讲述了在论文初期可选的硬件方案中选择了 Freescale 公司的硬件平台方案，并为进一步实现 ZigBee 协议栈搭建 MT-ZigBee 硬件开发平台；

第四章在所搭建硬件平台和相关的底层驱动程序基础上，设计实现 IEEE 802.15.4 物理层协议的主要功能；

第五章具体讲述了 IEEE 802.15.4 MAC 层相关基础知识，并在程序上有所裁减的设计实现了 MAC 层主要功能；

第六章首先介绍 ZigBee 网络层的主要内容，在此基础上设计实现对网络层协议功能的支持；

第七章以农业大棚的模拟应用为例，详细阐述了网络组建、加入、路由和网络地址分配等主要网络层功能；

第八章对本文工作进行了总结，并提出了一些后继工作。

# 第二章 ZigBee标准协议架构

在具体实现硬件平台和ZigBee各层协议之前，需要对ZigBee协议的整体架构有一个总体的认识。本章首先介绍了ZigBee协议的分层结构和ZigBee网络的拓扑结构，并在此基础上概括了物理层、MAC层、网络层和应用层的主要功能。

## 2.1 ZigBee协议总体结构

ZigBee技术是一种具有统一技术标准的短距离、低速率的无线通信技术，其物理层和媒体访问控制层协议由IEEE 802.15.4标准来规定，网络层和应用层协议则是由ZigBee技术联盟制定的[29][30]。

### 2.1.1 ZigBee协议的分层结构

ZigBee标准采用分层结构，每一层为上层提供一系列特殊的服务：数据实体提供数据传输服务；管理实体则提供所有其他的服务。所有的服务实体都通过服务接入点(service access point，SAP)为上层提供接口，每个SAP都支持一定数量的服务原语来实现所需的功能[31]。ZigBee标准的分层架构是在OSI七层模型的基础上根据市场和应用的实际需要定义的。其中IEEE 802.15.4-2003标准定义了底层协议：物理层(Physical Layer，PHY)和媒体访问控制层(Medium Access Control Sub-Layer，MAC)。ZigBee联盟在此基础上定义了网络层(Network Layer，NWK)，应用层(Application Layer，APL)架构。

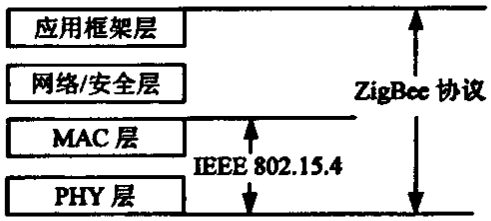

图2-1 ZigBee协议的分层结构

ZigBee协议的体系结构如图2-1所示[32]。其中PHY层主要功能包括启动和关闭无线收发器，信道能量检测，链路质量检测，信道选择，空闲信道评估(CCA)，以及通过物理信道对数据包进行发送和接收等；MAC层主要实现信标管理，信道接入，时隙管理，发送与接收帧结构数据，提供合适的安全机制等；网络安全层主要用于ZigBee网络的组网连接、数据管理和网络安全等；应用层主要为ZigBee技术的实际应用提供一些应用框架模型。

## 2.1.2 ZigBee 网络拓扑结构

在 ZigBee 网络中，根据设备所具有的通信能力，可以分为全功能设备（full-function device，FFD）和精简功能设备（reduced-function device，RFD）。FFD 之间以及 FFD 和 RFD 之间都可以相互通信；但 RFD 只能与 FFD 通信，而不能与其他 RFD 通信[33]。RFD 主要用于简单的控制应用，传输的数据量较少，对传输资源和通信资源占用不多，可以采用相对廉价的实现方案，在网络结构中一般作为通信终端。FFD 则需要功能相对较强的 MCU，一般在网络结构中拥有网络控制和管理的功能。

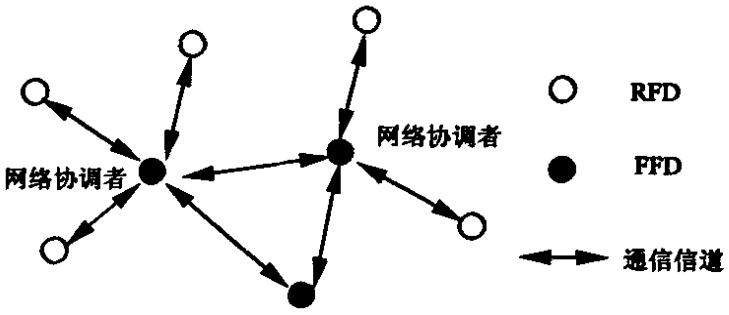


图 2-2 ZigBee 网络组件和拓扑关系

ZigBee 网络中，有一个称为 PAN 网络协调者（PAN coordinator）的 FFD 设备，它是网络的中心节点。PAN 网络协调者除了直接参与应用以外，还要负责其他网络成员的身份管理、链路状态信息的管理以及分组转发等功能。图 2-2 是 ZigBee 网络的一个例子，给出了网络中各种设备的类型以及它们在网络中所处的地位。

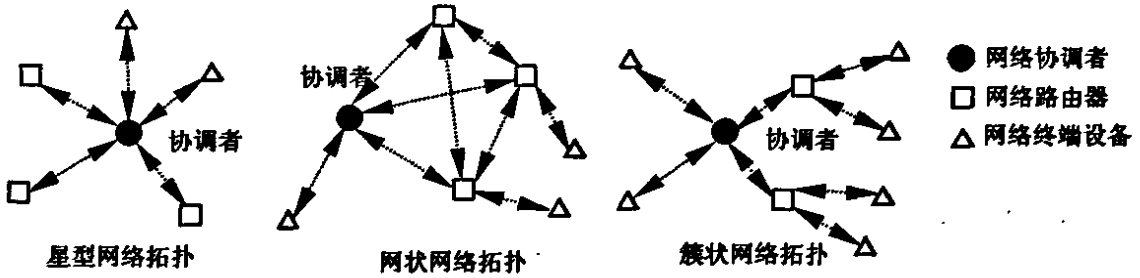


图 2-3 ZigBee 网络拓扑结构

从网络拓扑的角度来看，ZigBee 设备主要有 3 种角色：网络协调者、网络路由器和网络终端设备。其中网络协调者主要负责网络的建立，以及网络的相关配置；路由器主要负责找寻、建立以及修复网络报文的路由信息，并负责转发网络报文；网络终端具有加入、退出网络的功能，并可以接收和发送网络报文，但终端设备不允许路由转发报文。通常协调者和路由器节点一般由 FFD 功能设备构成，终端设备由 RFD 设备组成[34]。

ZigBee 网络根据应用的需要可以组织成星型网络、网状网络和簇状网络 3 种拓扑结构[35]，如图 2-3 所示。在星型结构中，所有的设备都与中心设备—PAN 网络协调者通信，实际上在这种简单的网络结构中路由器是没有路由作用的。在这种网络结构中，网络协调者一般使用电力系统供电，而其他设备采用电池供电。星型网络适合家庭自动化、个人计算机外设以及个人健康护理等小范围的室内应用；与星型网络不同，网状网络(Mesh)只要彼此在对方的无线辐射范围内，任何两个 FFD 设备之间都能直接通信，在 Mesh 网络中每一个 FFD 设备都可以认为是网络路由器，都可以实现对网络报文的路由转发功能，Mesh 网在构建时比较复杂，节点所要维护的信息较多；对于簇状网络实际上可以看做是一个复杂的星型网络，一个扩展的星型拓扑或是由多个简单的星型网络组成的拓扑结构，在簇状网络中，网络协调者、路由器和终端设备的功能清晰，相对于 Mesh 网络，构建簇状网络比较简单，所需的资源相对较少，并且可以实现网络的路由转发功能，从而也扩大了网络的通信范围[36]。

## 2.2 ZigBee 标准各层的主要功能

下面对 IEEE 802.15.4 的 PHY、MAC 层和 ZigBee 的网络层、应用层分别加以介绍，给出各层的主要功能。

### 2.2.1 物理层

物理层定义了物理无线信道和与 MAC 层之间的接口，提供物理层数据服务和物理层管理服务。物理层数据服务是从无线物理信道上收发数据，物理层管理服务维护一个由物理层相关数据组成的数据库。

(1) 无线信道的分配

IEEE 802.15.4 规范的物理层定义了三个载波频段用于收发数据：868MHz~868.6MHz、902MHz~928MHz 和 2400MHz~2483.5MHz[32]。在这三个频段上发送数据使用的速率、信号处理过程以及调制方式等方面都存在着一定的差异，其中 2400MHz 频段的数据传输速率为 250kbps，915MHz、868MHz 分别为 40kbps 和 20kbps。

IEEE 802.15.4 规范定义了 27 个物理信道，信道编号从 0 到 26，每个具体的信道对应着一个中心频率，这 27 个物理信道覆盖了以上 3 个不同的频段。不同的频段所对应的宽度不同，标准规定 868MHz 频段定义了 1 个信道(0 号信道)；915MHz

频段定义了10个信道(1～10号信道)；2400MHz频段定义了16个信道(11～26号信道)。这些信道的中心频率定义如下[32]：

F=868.3MHz k=0

F=906+2(k-1)MHz k=1,2,...,10

F=2405+5(k-11)MHz k=11,12,...,26

其中k为信道编号，F为信道对应的中心频率。

通常ZigBee硬件设备不能同时兼容2个工作频段，在选择时，应符合当地无线电管理委员会的规定。由于868MHz~868.6MHz频段主要用于欧洲，902MHz~928MHz频段用于北美，而2400MHz~2483.5MHz频段可以用于全球，因此在中国所采用的都是2400MHz的工作频段[37]。

(2) PHY层主要功能

物理层功能相对简单，主要是在硬件驱动程序的基础上，实现数据传输和物理信道的管理。数据传输包括数据的发送和接收；管理服务包括信道能量监测（energy detct，ED），链接质量指示（link quality indication，LQI）和空闲信道评估（clear channel assessment，CCA）等，其模型如图2-4所示。其中RF-SAP是由驱动程序提供的接口，而PD-SAP是PHY层提供给MAC层的数据服务接口，PLME-SAP是PHY层给MAC层提供的管理服务接口。

物理层服务包括以下五个方面[27][32]：

① 激活和休眠射频收发器；

② 信道能量检测（ED）；

③ 检测接收数据包的链路质量指示（LQI）；

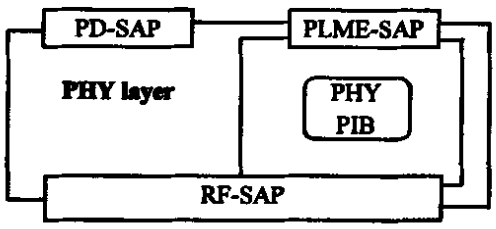


图2-4 PHY层模型

④ 空闲信道评估（CCA）；

⑤ 收发物理数据包。

信道能量检测为上层提供信道选择的依据，主要是测量目标信道中接收信号的功率强度。该检测本身不进行解码操作，检测结果为有效信号功率和噪声信号功率之和。

链路质量指示为上层服务提供接收数据时无线信号的强度和质量信息，它要对检测信号进行解码，生成一个信噪比指标。

空闲信道评估判断信道是否空闲。IEEE 802.15.4定义了三种空闲信道评估模式：第一种简单判断信道的信号能量，当信号能量低于某一门限值就认为信道空闲；

第二种通过判断无线信号的特征，该特征包含两个方面，即扩频信号特征和载波频率；第三种是前两种模式的综合，同时检测信号强度和信号特征，给出信道是否空闲的判断。

### 2.2.2 MAC 层

在 IEEE 802 系列标准中，OSI 参考模型的数据链路层进一步划分为 MAC 和 LLC 两个子层。MAC 子层使用物理层提供的服务实现设备之间的数据帧传输，就本质而言，实际上是建筑在物理层之上的为原始信息(比特或字节)的交互而建立的语法、句法逻辑实体结构；而 LLC 在 MAC 子层的基础上，在设备间提供面向连接和非连接的服务[26]。其中 IEEE 802.15.4 仅划分了 MAC 层。

MAC 层提供两种服务：MAC 层数据服务和 MAC 层管理服务。前者保证 MAC 协议数据单元在物理层数据服务中的正确收发，而后者从事 MAC 层的管理活动，并维护一个信息数据库。

IEEE 802.15.4 定义的 MAC 层协议，提供数据传输服务（MCPS）和管理服务（MLME），其逻辑模型如图 2-5 所示，其中 PD-SAP 是 PHY 层提供给 MAC 的数据服务接口，PLME-SAP 是 PHY 层给 MAC 层提供的管理服务接口， MLME-SAP 是由 MAC 层提供给网络层的管理服务接口，MCPS-SAP 是 MAC 层提供给网络层的数据服务接口。MAC 层的数据传输服务主要是实现 MAC 数据帧的传输；MAC 层的管理服务主要有信道的访问，PAN 的开始和维护，节点加入和退出 PAN、设备间的同步实现、传输事务管理等。

MAC 层的主要功能包括如下几个方面[27][32]：

① 网络协调者产生并发送信标帧（beacon）；

② 设备与信标同步；

③ 支持 PAN 链路的建立与断开；

④ 为设备的安全性提供支持；

⑤ 信道接入方式采用免冲突载波检测多路访问(CSMA-CA)机制；

⑥ 处理和维护保护时隙(GTS)机制；

⑦ 在两个对等的 MAC 实体之间提供一个可靠的通信链路。

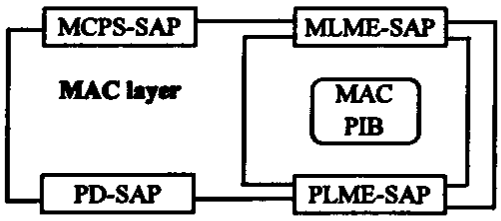


图 2-5 MAC 层模型

这里只给出 MAC 层功能的简介，具体内容将在 MAC 层程序实现中详细讲述。

### 2.2.3 网络层

网络层(NWK)是位于 MAC 层与应用层(APL)之间的一个协议层。网络层的任务是通过正确操作 MAC 层提供的功能来向应用层提供合适的服务接口。为了与应用层交互，网络层逻辑上包含两个服务实体：数据服务实体（NLDE）和管理服务实体（NLME）。

ZigBee 规范定义的 NWK 层协议，提供数据传输服务（NLDE）和管理服务（NLME），其逻辑模型如图 2-6 所示。其中 NLDE-SAP 是 NWK 层提供给 APL 层的数据服务接口，用于将 APL 层提供的数据打包成网络层协议数据单元，并将其传输给相应节点的 NWK 层；或者将接收到的 NWK 层协议数据单元进行解包，并将解包后得到的数据传送给本节点的 APL 层。也就是说 NLDE-SAP 实现两个 APL 层之间的数据传输；NLME-SAP 是 NWK 层给 APL 层提供的管理服务接口；MCPS-SAP 是由 MAC 层提供给 NWK 层的数据服务接口；MLME-SAP 是 MAC 层提供给 NWK 层的管理服务接口。

ZigBee 网络层管理服务的主要功能有以下 4 个方面[27][32]：

① 构建一个新网络；

② 设备加入已存在的网络；

③ 已加入网络的设备从网络中退出；

④ 网络报文的路由。

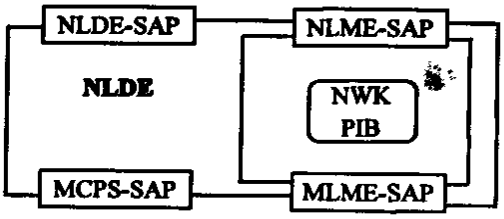


图 2-6 NWK 层模型

### 2.2.4 应用层

ZigBee 协议的应用层主要是在网络层接口程序基础之上，针对于具体的应用实例设计相关的应用协议。

在本文的最后，以农业大棚中参数的采集与控制为应用对象，给出一个较简单的应用实例。

## 2.3 本章小结

本章主要介绍了 ZigBee 协议的整体结构及各层主要功能。

参照 OSI 参考模型，ZigBee 协议采用标准的分层结构，共分为物理层、MAC 层、网络层和应用层，其中物理层和 MAC 层是由 IEEE 802.15.4 规范来定义的。在网络拓扑结构中，ZigBee 网络主要可采用星型、网状和簇状三种结构，各种网络的

特点及应用在本章中也有所讲述。IEEE 802.15.4 物理层规定 ZigBee 技术主要可以使用 2400MHz、915MHz 和 868MHz 三个频段，根据无线电管理规定，我国所采用的是 2400MHz 的频段。本章最后概述了物理层、MAC 层、网络层和应用层的主要功能及各层所提供的服务。

通过本章的介绍，对 ZigBee 协议框架有一个整体的认识，为下面各层的具体设计实行奠定基础。

# 第三章 MT-ZigBee硬件平台的设计

在设计开发 ZigBee 协议之前必须有相应硬件平台的支撑，本章主要介绍MT-ZigBee硬件平台的设计实现。首先介绍了目前现有的ZigBee硬件方案以及论文所选方案的主控制器和物理层射频芯片；接着详细介绍了硬件平台主要模块的具体设计，包括电源模块、主控制器支撑电路、主控制器与射频芯片的接口以及射频芯片的通信等模块；最后给出MT-ZigBee的硬件测试。

## 3.1 硬件选型

硬件选型在嵌入式产品的设计中是一个重要的环节，将直接影响着后续的开发设计流程。

### 3.1.1 ZigBee硬件方案

在ZigBee技术联盟中，Freescale、Ti、Chipcon、Philips等公司都是ZigBee标准制订的先驱。在射频收发芯片方面，主要有Freescale公司的MC13192、MC13193和Chipcon公司的CC2420、CC2430所提供的两大解决方案。下面简单介绍这两种可选的硬件开发方案。

挪威半导体公司Chipcon目前已成功被Ti公司收购，Chipcon推出的CC2430[38]是全球首颗符合ZigBee技术标准的2.4GHz射频芯片，它沿用了CC2420的架构。CC2430兼容IEEE 802.15.4标准，具有8051核的单片机，其在单芯片上集成了ZigBee RF前端、存储器和微控制器。另外，CC2430内部还包含了模数转换器(ADC)、定时器、AES-128协处理器、看门狗、32kHz晶振时钟、上电复位电路、掉电检测电路以及21个可编程I/O接口。CC2430的使用十分方便，只需要极少的外围电路支持。它所提供的 ZigBee 硬件方案是真正的系统级芯片解决方案，在消费电子、家庭与建筑自动化、汽车、工控系统等方面具有广泛的应用前景。

Freescale 公司针对 ZigBee 技术推出了完整的硬件解决方案，其中主要包括MC13192、MC13193射频(radio frequence，RF)收发芯片；与RF端相配套的低功耗HCS08核MCU；相关的传感器等。MC13192、MC13193是符合IEEE 802.15.4标准的射频数据调制解调器，它工作在2.4GHz频段下，与MCU通过标准的4线SPI接口通信，采用16个射频通道，数据速率为250kbps。与HCS08核MCU配套使用，

可提供经济高效的 ZigBee 硬件平台方案。此外，Freescale 公司还提供了详细的芯片手册、参考设计、布线设计等文档说明，这同时也可以加快硬件平台的搭建。在这里需要说明的是，Freescale 公司于 2006 年底推出了整合 S08 核和射频模块的单片 MC1321x[39]，但在论文设计的初期尚未推出，因而论文没有采用这种方案。

在上述两种方案的选择上，本文主要从对芯片及其开发环境的熟悉程度等方面进行选择。由于实验室对 Freescale 系列的 MCU 有着长期的研究，对 Freescale 公司的 8 位、16 位和 32 位系列 MCU 的烧写、启动以及内部的功能模块都比较熟悉，基于上述的考虑，论文选择了 Freescale 公司所提供的 ZigBee 硬件平台方案：MC9S08GB60 与 MC13192。

### 3.1.2 MC9S08GB60 微控制器

MC9S08GB60[40](以下简称 GB60)是一款 Freescale 公司 S08 系列的 8 位 MCU[41]。HCS08 核，最高总线频率可达 40MHz；增加了 16 位指令，能灵活方便的访问 16 位 HX 寄存器，同时支持 1 个 WAIT 和 3 个 STOP 模式，对低功耗模式提供更全面的支持，在 40MHz 的工作频率下，其功率消耗不到 1mW，而且该微控制器具有多种省电模式供选择；它内部具有 64KB 的 FLASH 和 4KB 的 RAM 存储空间；除了具有丰富的片上存储资源和多种省电模式以外，模数转换模块具有 8 路 10 位的 A/D 通道，2MHz 的采样频率；内部集成了 1 个 SPI 模块，适合与 MC13192 的通信；2 个 SCI 模块，方便与 PC 通信；具有背景调试模块，能利用单线对 HCS08 核的系列 MCU 进行方便地写入和调试，加快开发的速度并大大降低了调试的难度。

### 3.1.3 MC13192—ZigBee 物理层芯片

MC13192[42]是 Freescale 公司于 2005 年推出的工作在 2.4GHz 频率下短距离，低功率，工业、科学和医疗(ISM)的无线数据收发器。它包含基于 IEEE 802.15.4 标准设计的物理层结构，选择一款合适的 MCU 后，可提供一种性价比极高的 2.4GHz 频率短距离数据传输的无线方案。MC13192 与 MCU 的接口简单，只需四线的 SPI，一个 IRQ 中断请求线和三个控制线。SPI 用于 MC13192 和 MCU 进行双向的数据通信，MCU 对 MC13192 的配置和控制命令同样也是通过 SPI 传输的。当 MC13192 的工作状态发生变化时，MC13192 将通过 IRQ 管脚通知 MCU，并由 MCU 作相应的仲裁处理。

(1) MC13192 的基本性能特点[42][43]

① MC13192符合IEEE 802.15.4标准，它的工作频率是2.405～2.480GHz，数据传输速率为250kbps，采用O-QPSK调试方式；

② 功能丰富的双向2.4GHz收发器带有一个数据调制解调器，可以在ZigBee技术应用中使用，它还具有一个优化的数字核心，有助于降低MCU处理功率，缩短执行周期；

③ 内部集成了4个24位精度的定时比较器，使其可以和性能较低、价格低廉的MCU配合使用以降低成本；

④ 丰富的维护服务中断更加易于MCU的编程与调试；

⑤ 标准的4线SPI接口，方便与MCU的通信；

⑥ 芯片集成了连接质量指示与电源检测功能，该功能可以为ZigBee网络的组网及维护提供必要的信息；

⑦ 可编程的时钟模块，输出时钟可直接供主控MCU使用；

⑧ 可编程的输出功率，额定输出为0dBm，也可以通过编程提高最大输出到4dBm（建议使用额定输出，因为要符合无线电管制规范[44]）；

⑨ 三种省电模式可供选择，超低功率：芯片采用2.7V供电，接收状态耗电37mA，发射状态耗电30mA。

(2) MC13192的内部结构[42]

MC13192的内部结构如图3-1所示。芯片主要由模拟接收和发射部分、数字调制解调部分、片内频率合成器、电源管理部分以及与MCU接口部分组成。其中与MCU接口部分主要包括了SPI通信接口、7根GPIO口、4个24位定时器和大小为125字节载荷的内部缓冲RAM，这些资源是在设计MC13192的驱动程序时必须熟悉掌握的。

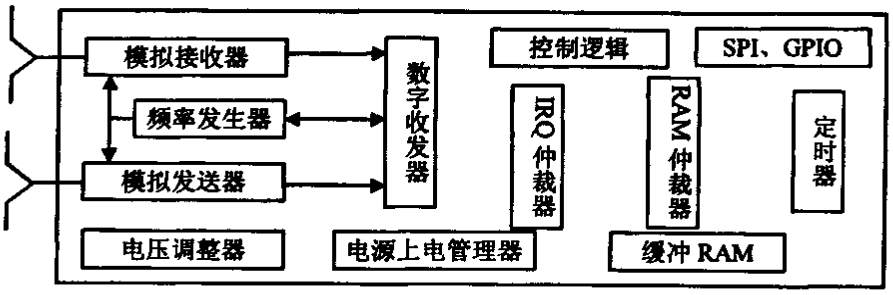


图3-1 MC13192结构框图

## 3.2 MT-ZigBee硬件平台设计

考虑到论文最后的具体应用，在MT-ZigBee硬件平台的搭建时，就预留了相关

的传感器接口、液晶和键盘等功能模块。因此，本小节所介绍的硬件设计就是后面具体应用的硬件基础，在应用章节将不再讲述具体的硬件工作了。

### 3.2.1 设计框图

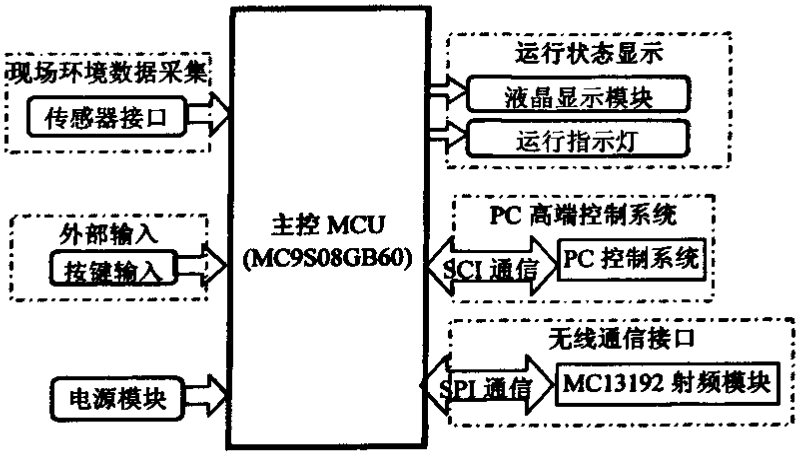


图 3-2 系统结构框图

MT-ZigBee 的系统结构如图 3-2 所示。其中主控 MCU 及其支撑模块和电源模块是整个硬件系统中最基本的部分，GB60 负责整个系统的控制，电源模块负责所有模块的电源供给；MC13192 射频模块是 2.4GHz 无线射频的基础，它与 GB60 之间主要通过 SPI 接口进行通信；运行状态显示包括有液晶显示和运行指示灯，这主要是用于显示当前系统的运行状态，同时也可以为硬件模块的检测和程序的调试带来方便；MT-ZigBee 可以通过串行通信接口与 PC 机的控制系统进行通信，这样就可以直接通过 PC 机来对 ZigBee 网络进行操作；考虑到论文最后具体的应用，硬件系统中还加入了现场环境数据的采集接口和按键的输入，为农业大棚的模拟应用提供相应的接口。

对于按键输入、SCI 串行通信模块、液晶及运行指示灯模块设计比较简单，不再叙述。下面重点介绍电源模块、MCU 支撑模块、GB60 与 MC13192 接口电路和 MC13192 无线射频通信模块的硬件设计。

### 3.2.2 电源模块

无线传感器网络主要用于采集现场数据，再进行相应控制。设备均安放在采集现场，考虑到便于携带、安装，供电电源采用 1 节 9V 的干电池。在硬件电路上电源分为两路：一路是单独供给主控芯片 GB60 的电源，另一路是供给 LCD、MC13192、SCI、按键和测试小灯等所有外围模块的电源。具体电源部分电路如图 3-3 所示。

主控芯片电源在任何情况下都是存在的，这样保证任何情况下GB60 都是工作的；外围模块电源是受到主控芯片控制的，GB60 通过 MOS 管来控制外围模块电源：当系统正常工作时，GB60 允许外围模块电源上电；当系统进入低功耗状态时，GB60 切断外围模块电源，这样整个系统只有主控芯片有供电，主控芯片再进入低功耗模式(Stop Mode)，这样就更好的实现了整个系统的低功耗。注意，在切断外围模块电源时，不能直接用一般的三极管，这样进入低功耗状态后外围模块仍然有较大的电流消耗，应该使用电流截止性能好的 MOS 管(如：SI2301)来实现。

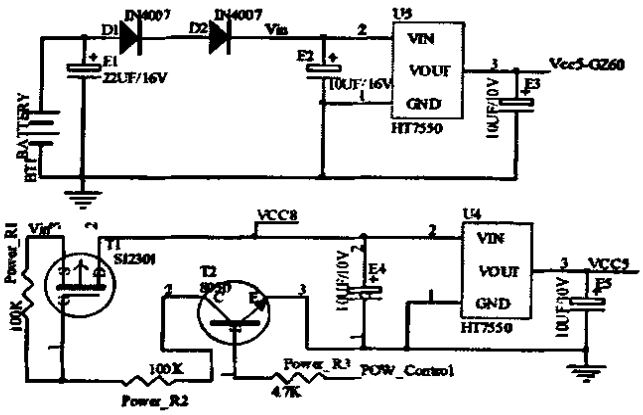

图 3-3 电源模块电路图

## 3.2.3 GB60 支撑电路

嵌入式 MCU 在硬件设计上一般都需要一定的支撑电路来使得 MCU 能够稳定的工作。作为 S08 系列的 GB60 所需要的支撑电路相对比较少，一般只需要晶振电路就可以使得 GB60 稳定工作，考虑到 MC13192 能够直接对外提供晶振时钟，所以可以直接采用 MC13192 的输出时钟作为 GB60 的时钟，因而在这一块设计时忽略晶振电路的设计，简化硬件设计，同时也降低成本。

## 3.2.4 MC13192 与 GB60 的接口电路

MC13192 与 GB60 的接口电路如图 3-4 所示。主要有 9 个接口连接：4 根 SPI 通信接口、IRQ 中断接口、3 根 MC13192 的控制口和 MC13192 时钟输出引脚。其中对于 4 线 SPI，根据参考手册指出，当作为 SPI 主机方式，同时 SPI 状态与控制寄存器的模式错误标志(MODF)有效并置为 1 时，$\overline{SS}$ 引脚可单独作为 I/O 口使用[42]。在本设计中 GB60 为 SPI 主机方，将$\overline{SS}$直接作为输出口使用，用以控制 MC13192 的 CE 使

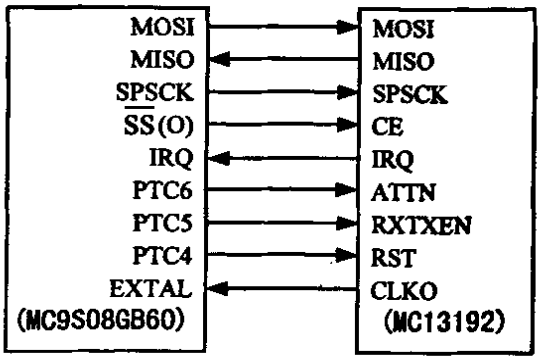

图 3-4 GB60 与 MC13192 接口

能信号。

GB60对MC13192上的寄存器、片上RAM读取和写入时都是通过标准的4线SPI接口来实现的。通信时，MC13192只能作为从机，因此对于MCU而言，MOSI线是发送数据线，而MISO线是接收数据线，SPI的同步时钟由GB60在SPSCK管脚上给出，连接到MC13192的SPICLK上。

MC13192的IRQ管脚连接到GB60的IRQ管脚上，MC13192上产生的所有中断事件直接反映给GB60。当GB60接收到来自MC13192的外部中断时，还要查询其中断标志寄存器，来判断产生的中断事件，并作出相应的处理。

在GB60对MC13192的3根控制口中，ATTN管脚用于MCU将MC13192从低功耗模式下唤醒，而RXTXEN管脚则用来使能MC13192的收发器。在通常情况，为了降低功耗，射频芯片的收发器都是关闭的，只有在发送和接收数据的时候才使能有效，这样能大大降低射频芯片的功耗。当射频芯片工作异常的时候，MCU也可以通过RST管脚来硬件复位MC13192。

MC13192的时钟输出引脚CLKO直接与GB60的EXTAL引脚相连接，从而使得GB60不再需要外部晶振电路的支持，直接采用来自MC13192的时钟源即可。该时钟源是可编程的，能够提供8种不同的时钟频率：16MHz、8MHz、4MHz、2MHz、1MHz、62.5KHz、32.768KHz和16.393KHz，本文中采用的是16MHz的频率。

### 3.2.5 MC13192无线射频通信模块

射频电路的设计是硬件设计中最为复杂的部分。这一部分对PCB的材质、电阻电容的精度、电路的走线等等都有很高的要求，其参数选择的好坏也直接影响到射频电路的质量。

射频电路的设计是在实验室原有工作基础之上，参考Freescale、Microchip等公司给出的设计样例[45][46]来进行设计开发的。本小节简要介绍MC13192的支撑电路和天线电路的设计。对于元器件的实际参数需要根据具体硬件电路进行调节，这是一个较为繁琐的过程。

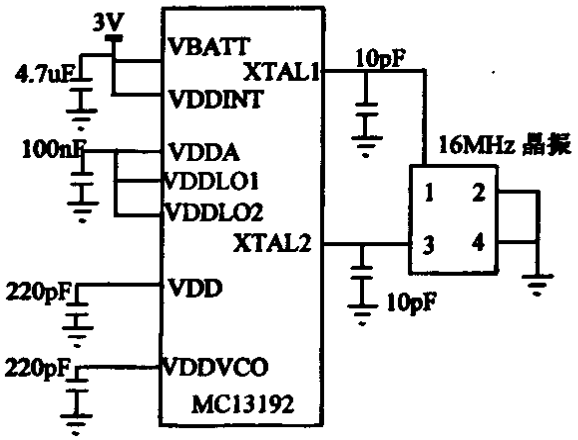


图3-5 MC13192支撑电路

(1) MC13192支撑电路

MC13192的支撑电路包括电源电

路，滤波电路和晶振电路，其逻辑连接如图 3-5[26]。VBATT 和 VDDINT 是电源输入引脚，MC13192 的正常工作电压为 2.0-3.6V，必须接一个 4.7uF 的稳压电容。VDDA，VDDLO1 和 VDDLO2 为经过整流的模拟电压，须旁接一个 100nF 的滤波电容。VDD 为经过内部整流的数字电压，旁接一个 220pF 的滤波电容。VDDVCO 为 VCO(voltage controlled oscillator)电路供电，同样必须旁接一个 220pF 的电容。XTAL1 和 XTAL2 外接 16MHz 的专用于 2.4GHz 射频电路的晶振，其旁路电容为 10pF。

(2) 天线电路的设计

用于 2.4GHz 射频电路的天线有 3 种类型：外接直立天线、PCB 天线和片式天线。外接直立天线的性能最好，但体积过大，只能用于对体积无要求的场合；片式天线采用集成电路来实现，性能一般，而且很难根据实际调整性能；PCB 天线具有体积优势，但是对设计和 PCB 布线要求高，在无线传感器网络的硬件平台上应用最多。

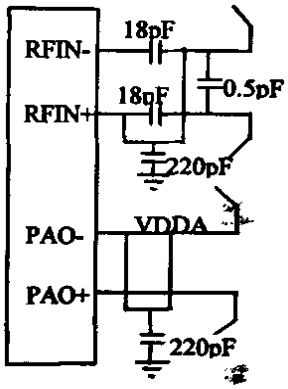

图 3-6 天线逻辑电路

图 3-6 为天线电路的原理图。RFIN-和 RFIN+为接收通道，2 个 18pF 的电容过滤掉高频干扰信号，而 0.5pF 的电容能防止共扼干扰。PAO-和 PAO+为发送通道，这两个管脚和 VDDA 连在一起，给发送通道提供必要的能量。

综合各方面因素考虑，最终制作的 MC13192 射频模块的 PCB 如图 3-7 所示。

## 3.3 MT-ZigBee 的硬件测试及心得

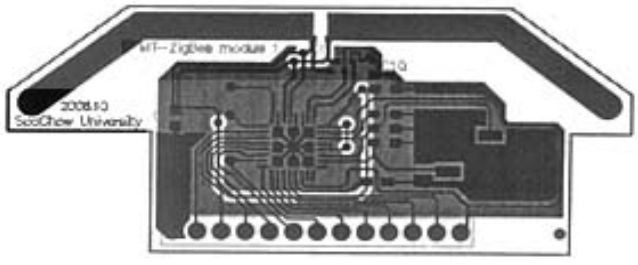
图 3-7 射频模块 PCB 图

在完成硬件电路设计后，必须对各模块的硬件电路进行测试，以保证硬件电路的可靠性。

MT-ZigBee 主要模块的基本测试流程如下：

(1) 电源模块测试

在空的 PCB 电路板上首先将电源模块的相关元器件焊接好，上电后直接利用万用表测量电源的输出点，看是否得到要求的电压值，以保证其他模块能正常工作。

(2) 微控制器部分测试

当电源模块工作正常后，就需要测试 GB60 是否正常工作。对于 MCU 的测试

主要就是通过 BDM 烧写器与 GB60 通信，看是否能进行正常的擦除与写入操作。若无法正常工作，则首先就应该仔细核对 MCU 支撑电路及电阻、电容的值是否正确，尤其是在晶振电路部分(本设计没有晶振电路)。GB60 含有 4MHz 的内部时钟源，且外围电路很少，所以比较容易调试通过。

芯片的焊接也是不能忽视的环节，在 GB60 的焊接时就遇到过这样的问题，对于这种 QFP 封装的密脚芯片在焊接时需要使用助焊剂来帮助芯片的放置与焊接。当时使用的是实验室剩留的助焊剂，但在焊接好之后发现 GB60 工作总是不稳定，BDM 烧写器经常找不到芯片，再查找了很长时间以后才发现是由于助焊剂的问题导致的：助焊剂时间过久，失效了，从而造成了芯片管脚短路。在重新使用新的助焊剂焊接后，芯片工作稳定。

(3) MC13192 模块测试

对于 MC13192 射频模块的测试，主要就是通过读写其内部寄存器和缓冲区来进行测试的，这部分的内容将在物理层的底层驱动程序部分讲述。

(4) 其他外围模块测试

其他外围模块的测试顺序没有特定的要求。串行通信(SCI)就是通过 PC 实现基本的收发；测试小灯模块，通过 MCU 将相应的 I/O 口置不同的值，看是否能点亮对应的小灯；液晶 LCD 模块也是同样如此，看是否在液晶上显示指定的字符。当模块测试不通过时，则首先应该通过测试小灯测试各模块相应的 I/O 口是否焊接正常。

硬件的测试是一个复杂的过程，但也还是有一定方法可循的。对于整块硬件电路，在首次调试时千万不要将所有元器件都焊接上，应该按模块分别焊接、调试，并逐模块调试通过后再联合起来一起调试。当然各模块也需要按照一定的顺序进行调试，如第一个模块肯定是电源模块。硬件的设计也是一个很需要耐心和细心的工作，在原理图设计时应该按模块设计，同时在原理图中可以用虚线框标出各个不同的模块，这样在以后查找时就很方便了；PCB 电路板的绘制一定要格外细心，按模块将相关的元器件尽量放在一起，特别是滤波电容，离的远了，就起不到滤波的作用；另外在硬件 PCB 设计的最后阶段时需要预留出一定的测试点，以便以后调试测量使用；最后就是高频电路部分的设计，这部分的设计对于元件的摆放位置、线路的走向等等都是有要求的，需要较丰富的设计经验，本文这部分的设计主要是在实验室原有的基础上实现的。

## 3.4 本章小结

本章详细介绍了 MT-ZigBee 的相关设计。目前常用的 ZigBee 硬件方案主要由 Freescale 和 Chipcon 两家公司提供，在介绍了这两种方案之后，根据对芯片和其开发流程的熟悉程度，本文选择了 Freescale 公司的 GB60 和 MC13192；在简要介绍了这两种芯片的基本特性之后，接着通过具体的分析，给出了 MT-ZigBee 的系统框架，并在此基础上阐述了电源电路、MC13192 与 GB60 接口电路和 MC13192 射频模块的详细设计；本章的最后给出了 MT-ZigBee 的硬件测试以及在硬件设计和测试过程中的一些心得体会。

# 第四章 物理层协议分析与实现

本章的目的就是设计实现 IEEE 802.15.4 物理层协议的主要功能。MC13192 是符合 IEEE 802.15.4 标准的物理层芯片，物理层程序的设计是直接面向硬件的，因而协议的实现必须有相关的底层驱动作为基础。

本章首先介绍了物理层协议与底层驱动之间的联系；接着讲述了主要的硬件驱动和 MC13192 驱动部分的设计，其中包括了 MC13192 片上寄存器与缓冲区的读写、MC13192 输出时钟的设置、运行状态的切换以及 IRQ 中断的处理等；在此基础上详细阐述了物理信道的设置、物理数据包的收发和信道能量的检测等物理层主要功能的设计实现。在本章的最后给出了整个物理层实现的主要函数列表和物理层的相关测试。

## 4.1 物理层协议与底层驱动的联系

在嵌入式软件设计中，需要将与硬件相关的驱动程序和与硬件无关的程序分离开，这样在软件设计中，便于程序的调试运行，同时也有助于与其他设计者之间的交流沟通和软件接口的清晰化。图 4-1 给出了物理层功能函数与底层驱动之间的关系。

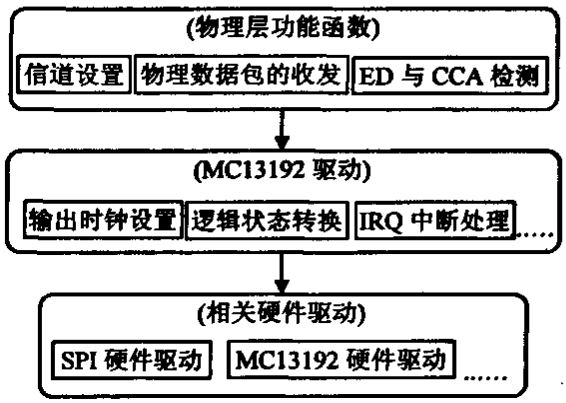


图 4-1 物理层函数与底层驱动的关系

硬件驱动部分直接建立在硬件基础之上，是直接驱动各硬件模块工作的基本函数。与物理层协议相关的主要有 SPI 硬件驱动和 MC13192 硬件驱动两部分。

在硬件驱动之上，MC13192 驱动部分主要指 MC13192 内部相关模块的功能函数。与 MC13192 硬件驱动不同：MC13192 硬件驱动是指 MCU 通过 SPI 接口来驱动 MC13192 正常工作，它仅包含 MC13192 内部寄存器和缓冲区的读写操作；而 MC13192 的驱动则是指其功能模块的程序实现，主要包括输出时钟的设置、逻辑状态的转换和 IRQ 中断的处理等。

最上层则是物理层功能函数的设计了，主要包括物理信道的设置、物理数据包的接收与发送、信道能量检测(ED)与空闲信道检测(CCA)。

## 4.2 相关底层硬件驱动

与ZigBee物理层相关的底层硬件驱动主要包含了SPI硬件驱动和MC13192硬件驱动两部分。对于SPI驱动，由于GB60内部含有SPI模块，这部分的驱动实现简单，下面仅介绍MC13192硬件驱动的实现。

MC13192硬件驱动主要就是GB60通过SPI通信接口实现对MC13192片上寄存器、收发缓冲区的读取和写入。

MCU通过SPI与MC13192之间的通信有两种方式[42]：一种是单字读写模式；另一种是包模式，也称为巡回读写模式。这两种方式在MC13192硬件驱动中均有应用：对MC13192片上寄存器的读写采用单字读写模式，对缓冲区的读写则采用包模式。

(1) MC13192片上寄存器的读写

MC13192有48个片上寄存器[42]，寄存器的数据宽度为16位，其中寄存器的地址范围为0x00~0x2F。对于寄存器的读写操作采用的是单字SPI读写模式。

对于单字读写方式，在数据格式上一次实际数据操作长度是24位：8位的头信息加16位的实际操作字数据。在前8位的头信息中，按位顺序第1位为读写标志位，表示本次操作是读操作还是写操作，0表示写操作，1表示读操作；第2位固定为0；后6位表示本次操作的寄存器地址；接下来的16位按实际字数据的高低字节排列。单字读写操作的时序如图4-2所示。

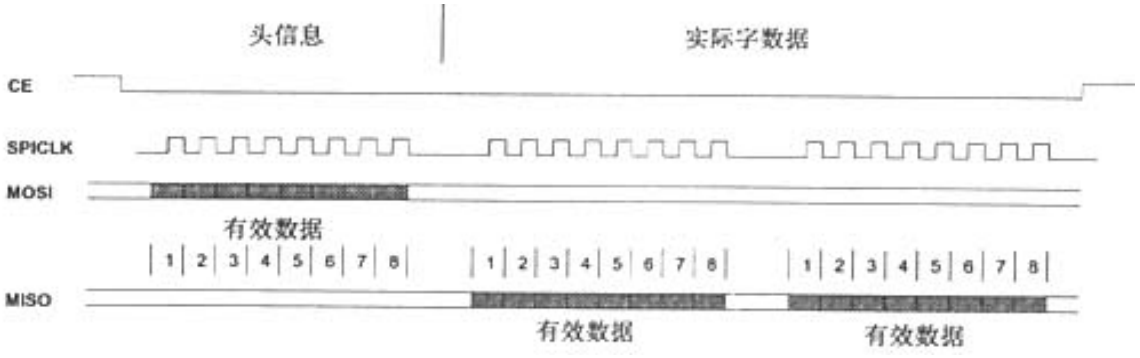


图4-2 单字读写操作时序

在程序设计上，寄存器的读写操作通过SPIReadWord和SPIWriteWord两个函数，这两个函数就是按照上面的时序来实现的。

```
/*SPIReadWord:MCU 通过 SPI 从 MC13192 读取 1 个字----------*
*功  能:MCU 通过 SPI 从 MC13192 读取 1 个字                *
*参  数:u8Addr:要读取 MC13192 片内寄存器的地址             *
*返  回:读出的 1 个字数据                                  *
*-----------------------------------------------------------*/
UINT16 SPIReadWord(UINT8 u8Addr)
```

```
/*SPIWriteWord:MCU 通过 SPI 向 MC13192 写入 1 个字-----------------------------*
*功  能:MCU 通过 SPI 向 MC13192 写入 1 个字                                    *
*参  数:u8Addr:要写入 MC13192 片内寄存器的地址                                 *
*       u16Content:要写入 1 个字内容                                           *
*返  回:无                                                                     *
*-------------------------------------------------------------------------*/
void SPIWriteWord(UINT8 u8Addr, UINT16 u16Content)
```

(2) MC13192 片上缓冲区的读写

MC13192 片内含有最大可容纳 125 字节载荷的收发缓冲区。对于收发缓冲区的读写采用的是 SPI 包读写模式，包读写模式中第 1 个字节的含义与单字读写模式一致：第 1 位为读写标志位，表示本次操作是读操作还是写操作；第 2 位固定为 0；后 6 位表示本次操作的寄存器地址，同时该地址为巡回读写提供一个起始地址；后面按字顺序为实际操作的数据。

缓冲区的读写操作分别通过 ReadRXRAM 和 WriteTXRAM 两个函数实现。

```
/*ReadRXRAM:MCU 从 MC13192 的 RX RAM 中读取数据块-----------------------------*
*功  能:MCU 从 MC13192 的 RX RAM 中读取数据块                                  *
*参  数:psRxPkt:读出数据块的指针                                               *
*返  回:操作结果：成功 !0；失败 0                                              *
*-------------------------------------------------------------------------*/
UINT8 ReadRXRAM(PHYPacket *phyRxPacket)
```

```
/*WriteTXRAM:MCU 向 MC13192 的 TX RAM 中写入数据块-----------------------------*
*功  能:MCU 向 MC13192 的 TX RAM 中写入数据块                                  *
*参  数:psTxPkt:要写入数据块的指针                                             *
*返  回:操作结果：成功 1；失败 0                                               *
*-------------------------------------------------------------------------*/
UINT8 WriteTXRAM(PHYPacket *phyTxPacket)
```

MC13192 规定接收缓冲映射在寄存器 RX_Pkt_RAM(0x01)中，它的读取流程如图 4-3 所示[42]。首先读取 RX_Status(0x2D)的 0~6 位，为接收到字符数据的长度，并根据读取的值计算循环访问的次数 N；接着使能 MCU 与 MC13192 通信；再将第 1 个头信息发送给 MC13192，其中为读操作，寄存器地址为 0x01；依次循环 N 次从 RX_Pkt_RAM 中读取接收到的数据；最后停止 MCU 与 MC13192 通信。

发送缓冲映射在寄存器 TX_Pkt_RAM(0x02)中，缓存的写操作如图 4-4 所示，其基本的执行流程与读操作相似，不再讲述。

## 4.3 MC13192 驱动

MC13192 的驱动部分主要有输出时钟的设置、MC13192 运行状态的转换和 IRQ 中断处理等功能模块。

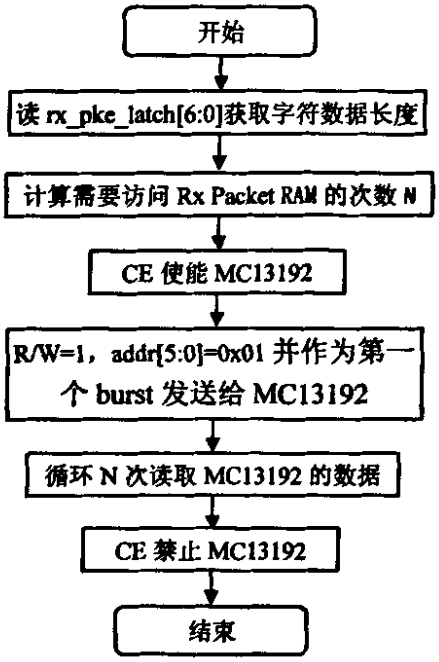


图 4-3 缓冲读操作流程

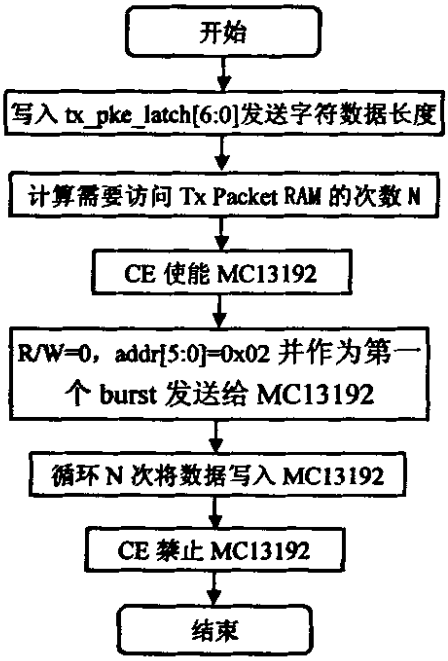


图 4-4 缓冲写操作流程

### 4.3.1 MC13192 输出时钟的设置

在 ZigBee 硬件平台的设计时，GB60 不采用外部晶振的方式，而是直接使用 MC13192 的 CLKO 引脚所提供的时钟源来作为 GB60 的系统时钟。

MC13192 的 CLKO 引脚输出时钟频率可以为 16MHz、8MHz、4MHz、2MHz、1MHz、62.5KHz、32.768KHz 和 16.393KHz 这 8 种不同的频率[43]。PLMESetMC13192ClockRate 函数就是实现 CLKO 输出时钟的选择。在该函数中通过对 MC13192 时钟寄存器的设置来选择时钟频率，默认输出为 32.768KHz，论文中选择的是 16MHz 的时钟频率。

```
/*PLMESetMC13192ClockRate:设置 CLOK 的输出频率----------------------------*
*功  能:本地 MAC 层设置 CLOK 的输出频率                                    *
*参  数:u8Freq:需要设置的频率所对应的数字                                  *
*       0  ---  16Mhz                                                     *
*       1  ---  8Mhz                                                      *
*       2  ---  4Mhz                                                      *
*       3  ---  2Mhz                                                      *
*       4  ---  1Mhz                                                      *
*       5  ---  62.5KHz                                                   *
*       6  ---  32.786KHz default                                         *
*       7  ---  16.393Khz                                                 *
*返  回:结果参数:成功 1；失败 0                                           *
*-------------------------------------------------------------------------*/
UINT8 PLMESetMC13192ClockRate(UINT8 u8Freq)
```

在此需要注意的问题是 GB60 在设置 MC13192 输出时钟之前是采用什么时钟源的？也就是说在获得 MC13192 的输出时钟之前，GB60 必须能够以某一频率来稳定地工作，这样才能够与 MC13192 通过 SPI 进行通信，并进一步初始化配置 MC13192。实际上 GB60 内部具有一个 4MHz 的时钟源，这样就解决了这个问题，在未获得 MC13192 的输出时钟之前，GB60 使用内部 4MHz 的时钟作为系统时钟，在配置好 MC13192 后，再采用 MC13192 的 CLKO 的输出时钟作为系统时钟。

如果不采用 MC13192 的输出时钟，仅使用 GB60 内部的 4MHz 时钟也是可以正常的。但内部时钟源频率固定为 4MHz，而 MC13192 输出时钟可以提供高达 16MHz 的晶振频率。为提高通信、数据处理的速度，从而采用 MC13192 的输出时钟作为 GB60 的系统时钟源。

### 4.3.2 MC13192 的运行状态

MC13192 有多种运行状态或工作模式，可分成两种类型：一类是活动模式，另一类是低功耗模式。具体可分为 7 种：Off 模式、Hibernate 模式、Doze 模式、Idle 模式、Rx 模式、Tx 模和 CCA\ED 模式[42]。各种模式对硬件资源的需求不尽相同，模式之间的转换条件及转换时间也不同，具体的工作模式参见表 4-1。

表 4-1 MC13192 的工作模式

| 模式 | 晶振 | CLKO | SPI | RAM 数据是否丢失 | 输出为三态 | 转换时间 |
|---|---|---|---|---|---|---|
| Off | x | x | x | 是 | √ | 25ms→Idle |
| Hibernate | x | x | x | 否 | x | 20ms→Idle |
| Doze | √ | ≤1M | x | 否 | x | 300+1/CLKOus→Idle |
| Idle | √ | √ | √ | 否 | x | |
| Receive(Rx) | √ | x | √ | 否 | x | 144us←Idle |
| Transmit(Tx) | √ | x | √ | 否 | x | 144us←Idle |
| CCA/ED | √ | x | √ | 否 | x | 144us←Idle |

低功耗模式主要有 Off 模式、Hibernate 模式和 Doze 模式。Off 模式的功耗最低，硬件复位后，直接进入 Off 模式；Hibernate 模式下的功耗仅次于 Off 模式，这种模式下硬件状态与 Off 模式下的状态基本上相同，唯一不同的是 RAM 区的数据可以被保留；Doze 模式是一个附加的低功耗模式，用来协同事件定时器的工作，此模式下大部分的硬件模块都不工作，仅参考时钟和事件定时器仍然工作，且内部 RAM 区数据和 SPI 的配置仍被保留。

活动模式有 Idle 模式、RX 和 TX 模式、CCA/ED 模式。Idle 模式是退出任何一

个其他模式后的默认工作模式，同样也是进入另外一个工作模式的必经状态；RX、TX 模式为接收、发送数据所处的工作状态，即当 MC13192 正在接收物理数据包时处于 RX 模式，当 MC13192 在发送数据包时处于 TX 模式；CCA/ED 模式为空闲信道评估/能量检测模式，该模式可用来测量当前选定信道的能量值。

以上 7 种工作模式的转换条件如图 4-5 所示。

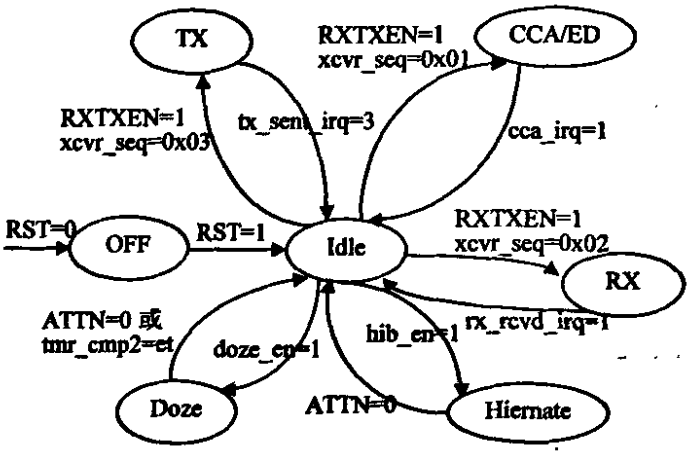


图 4-5 模式转换图

其中 Idle 模式是退出任何一个其他模式后的默认工作模式，是进入另外一个工作模式的必经状态，也就是说任何两个非 Idle 模式 A、B，A 转换成 B，则 A 必须先转换成 Idle 模式，再由 Idle 模式转换成 B，即 A—>Idle—>B。

某一时刻 MC13192 只可能处于一个状态下，从图 4-5 可以看出，MC13192 通常处于 Idle 状态，这样当需要切换到下一状态时，只要转换条件满足就可以了。如当前 MC13192 处于 Idle 状态，且用户有数据需要发送，则 MC13192 需要先转入 TX 状态。按照转换条件，将 MC13192 的 RXTXEN 管脚置高，即允许其收发；接着将 Control_A(0x06)寄存器的 xcvr_seq[1:0]设置为 3，即发送状态，则 MC13192 进入 TX 模式。在数据发送完之后，MC13192 同样需要再次切换到 Idle 状态，为下一次状态转换做准备。

### 4.3.3 IRQ 中断处理

MC13192 通过唯一的中断输出引脚—IRQ 引脚来反映当前 MC13192 的状态变化。在前面的硬件设计中可以看出，本文将该中断引脚直接与 MCU 的 IRQ 脚连接，这样一旦 MC13192 产生中断就会触发 MCU 的 IRQ 中断。因而对 MC13192 的 IRQ 中断处理是在 GB60 的 IRQ 中断处理函数中完成的。

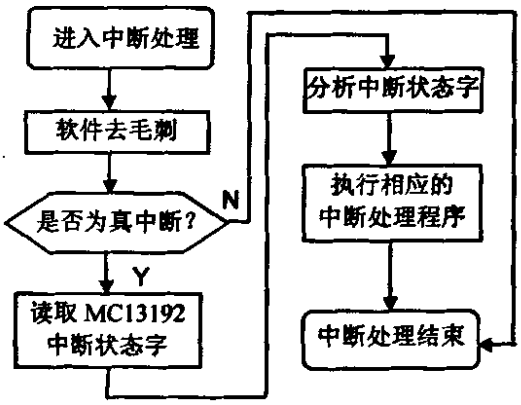


图 4-6 MC13192 中断处理流程

图 4-6 描述了实际的中断处理过程。在进行实际中断处理之前，必须进行软件

去毛刺过程，防止产生误中断。此外，MC13192 还具有中断屏蔽功能，只有那些没有被屏蔽的中断才能进行实际的中断处理。根据 MC13192 IRQ_Status 寄存器的内容，产生的中断主要有接收完成中断、发送完成中断、定时器比较中断、ED 检测中断等，在该中断处理函数中需要分别予以处理，对应每一种中断实际上就是置相应的状态标志，来反映 MC13192 当前所处的状态。

## 4.4 物理层功能实现

物理层实现的主要功能包括有物理信道的设置、物理数据包的收发和物理信道的检测三个部分。

### 4.4.1 物理信道设置

MC13192 是 2.4GHz 频段的射频芯片，IEEE 802.15.4 规定在物理信道上 2.4GHz 频段上对应 16 个物理信道，其中信道号为 11~26，这在第二章物理层信道分配中有所讲解。

在本文中通过 PLMESetChannelRequest 函数来选择相应的物理信道，其中入口参数只支持 11~26，对于其他的入口参数，函数将错误退出。

```
/*PLMESetChannelRequest:设置物理信道----------------------------------------*
*功   能:本地 PHY 层设置物理信道                                            *
*参   数:u8Channel:需要设置的信道号(0~26)                                    *
*返   回:TRUE：成功；FALSE:失败                                             *
*备   注:2.4GHz 对应信道号为 11~26                                          *
*-------------------------------------------------------------------------*/
Bool PLMESetChannelRequest(UINT8 u8Channel)
```

在入口参数为 11 至 26 号信道后，根据输入参数对 MC13192 的 LO1_Int_Div 和 LO1_Num 寄存器进行相应设置，选择 MC13192 的射频频率。

### 4.4.2 物理数据包的收发

PHY 层数据单元(PHY protocol data unit，PPDU)的格式如表 4-2 所示[32]。

表 4-2 PPDU 结构

| 4字节 | 1字节 | 1字节 | | 变量 |
|---|---|---|---|---|
| 前同步码 | 帧定界符 | 包长度(7bit) | 预留位(1bit) | PSDU |
| 同步包头 | | 物理层包头 | | 物理层载荷 |

其中同步包头可以使得接收设备锁定在比特流上，并且与比特流保持同步；物理层包头主要包含数据包的长度信息；物理层服务数据单元(PHY service data unit，

PSDU)，也称物理层载荷，装载的是从 MAC 层接收来的帧信息。MC13192 是支持 IEEE 802.15.4 的物理层芯片，因此 PPDU 的数据包是不需要用户进行组装和分解的，用户只需要写入和读取 PSDU 即可，物理层数据包的组装与分解是由 MC13192 硬件完成的。

(1) 发送 PHY 层载荷

物理层载荷的发送比较简单。首先检查 PSDU 载荷长度，不允许超过 125 字节，这主要是对应 MC13192 片上缓冲 RAM 的大小。根据状态转换条件可知，要发送物理数据包，当前 MC13192 的运行状态必须为 Idle 状态，因而若当前不处于 Idle 状态，表示 MC13192 正在运行其他任务，则无法进行发送任务；若为 Idle 状态，将要发送的 PSDU 载荷装入 MC13192 的 RAM 缓冲，再将 MC13192 的运行状态转换为发送状态，MC13192 将在预先设定好频率的无线信道将整个物理数据包(PPDU)发送出去。最后等待发送完毕所产生的 IRQ 中断，将当前状态置回为 Idle 状态。这样就完成了物理物理载荷(PSDU)的发送过程。

```
/*PDDataRequest:发送数据请求函数-----------------------------------------*
*功   能:本地 MAC 层调用该原语发送数据                                    *
*参   数:psPacket:发送数据的结构体(包括长度和首地址)                      *
*       gu8RTxMode,全局变量,MC13192 所处的模式                           *
*返   回:PHY 发送数据后的逻辑的状态                                       *
*-----------------------------------------------------------------------*/
PHY_Logic_Status PDDataRequest(PHYPacket *phyTXPacket)
```

(2) 接收 PHY 层载荷

当 MC13192 的收发器已被打开，若此时相同频率的无线信道上存在数据的传输，且在其射频范围以内，则 MC13192 会接收该物理数据包。当接收数据完成，MC13192 自动去除该物理数据包的同步包头和物理层包头，并将剩下的物理载荷域放入片上 RAM 缓冲中，同时将载荷的长度写入 MC13192 的接收数据包长度寄存器(0x2D)；接着 MC13192 会产生接收中断，并通过 GB60 的 IRQ 中断通知主控 MCU。以上的工作都是由硬件来完成的，对于接收程序则比较简单，只须根据接收数据包长度寄存器的值从缓冲 RAM 中读出相应长度的数据即可，并作为 MAC 帧交给 MAC 层处理。

### 4.4.3 ED 与 CCA 检测

MC13192 芯片支持 IEEE 802.15.4 规定的信道能量检测(ED)和空闲信道评估(CCA)的功能。ED 检测的是当前指定信道的能量值；而 CCA 则是评估当前指定信道是否空闲，但 CCA 评估与 ED 检测的实现机制基本一致，只是在检测到能量值

的基础上，进一步将该值与预先设定的门限值比较，根据比较的结果判断当前信道是否空闲。

```
/*PLMECCARequest:CCA 信道检测-------------------------------------------*
*功   能:CCA 信道检测                                                   *
*参   数:无                                                             *
*返   回:TURE：当前信道空闲；FALSE：当前信道忙                          *
*-----------------------------------------------------------------------*/
Bool PLMECCARequest(void)
```

CCA 检测功能主要用于信道的有效访问。在 MAC 层的信道访问中采用冲突避免(CSMA-CA)的访问机制，在发送 MAC 帧之前首先需要使用 CCA 检测信道，若指定信道空闲才允许 MAC 帧的进一步发送。

表 4-3 物理层函数列表

| 文件名 | 函数编号 | 函数头 | 函数功能简述 |
|---|---|---|---|
| MC13192.c | 1 | void MC13192Init(void) | MC13192的初始化工作 |
| | 2 | void SPIWriteWord(UINT8 u8Addr, UINT16 u16Content) | MCU通过SPI向MC13192写入一个字 |
| | 3 | UINT16 SPIReadWord(UINT8 u8Addr) | MCU通过SPI从MC13192读取其一个字 |
| | 4 | Bool WriteTXRAM(PHYPacket phyTxPacket) | MCU向MC13192的TX RAM中写入数据块phyTxPacket |
| | 5 | Bool ReadRXRAM(PHYPacket *phyRxPacket) | MCU从MC13192的RX RAM中读取数据包 |
| Zig_PHY.c | 1 | Bool PHYInit(UINT8 ch,UINT8 CLKout) | 物理层初始化工作 |
| | 2 | PHY_Logic_Status PDDataRequest( PHYPacket psPacket) | 发送物理数据包请求函数 |
| | 3 | void PDDataIndication(UINT8 len) | 接收到数据指示原语 |
| | 4 | PHY_Logic_Status PLMESetTrxStateRequest( PHY_Logic_Status u8State) | 设置PHY层的逻辑运行模式 |
| | 5 | Bool PLMESetChannelRequest(UINT8 u8Channel) | 设置物理信道，其中u8Channel有效范围为11~26 |
| | 6 | Bool PLMESetMC13192ClockRate(UINT8 u8Freq) | 设置CLOK的输出频率 |
| | 7 | Bool PLMESetMC13192TmrPrescale(UINT8 freq) | 设置MC13192定时器的分辨率 |
| | 8 | UINT32 PLMEGetTimeRequest(void) | 获取MC13192 Timer当前时间值 |
| | 9 | Bool PLMESetTimeRequest( UINT32 u32RequestedTime) | 设置MC13192 Timer当前时间值 |
| | 10 | Bool PLMEEnableMC13192Cmpx(UINT8 ch, UINT32 u32RequestedTime) | 使能MC13192的比较器x |
| | 11 | Bool PLMEDisableMC13192Cmpx(UINT8 ch) | 禁止MC13192的比较器x |
| | 12 | Bool PLMECCARequest(void) | CCA信道检测 |
| | 13 | Bool PLMEEDRequest(UINT8* Energy) | 信道能量检测 |

IEEE 802.15.4 规定了 3 种 CCA 模式[32]，前文已有叙述。本文的 CCA 检测采

用第一种模式，即通过判断信道的信号能量，当信号能量值低于某一门限值时就认为该信道是空闲的，具体的实现函数是 PLMECCARequest。

## 4.5 物理层函数列表

物理层协议的函数设计主要包含底层驱动和物理层功能函数两部分，分别对应在 MC13192.c 和 Zig_PHY.c 两个文件中。表 4-3 列出了这两部分的主要函数头及简要的说明。

由于篇幅有限，在此仅给出函数头及函数功能简述，具体的函数代码在此不再给出。

## 4.6 物理层测试及心得

对于物理层所实现的功能函数在设计过程中是逐步测试通过的，本小节给出了物理层功能的综合测试，测试界面如图 4-7 所示。

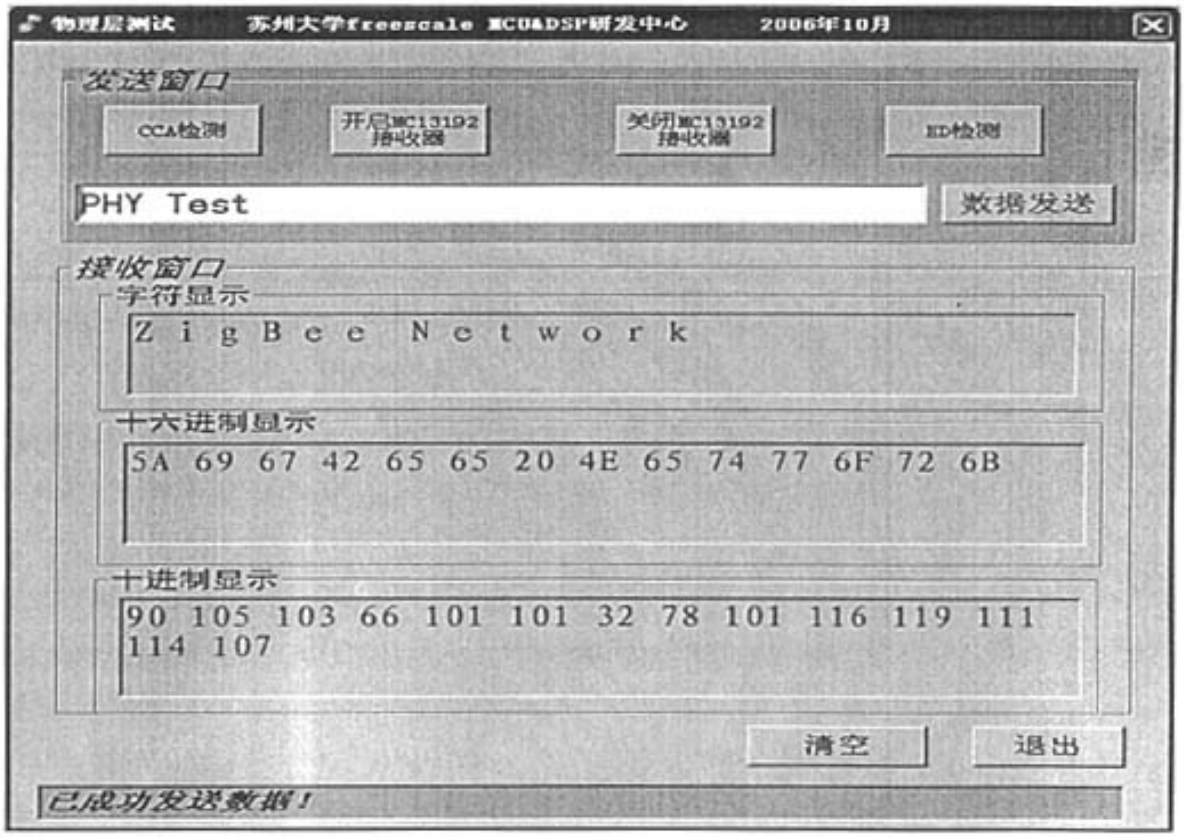


图 4-7 物理层测试界面

在物理层的综合测试中主要测试了 MC13192 收发器的打开与关闭、物理数据

包的发送与接收、ED 检测和 CCA 检测等功能。对于 MC13192 收发器的打开与关闭在实现上很简单，只要将 MC13192 的 RXTXEN 管脚置高电平或低电平即可；数据包的发送与接收是由函数 PDDataRequest 与 PDDataIndication 来实现的；ED 检测、CCA 检测分别由函数 PLMEEDRequest 和 PLMECCARequest 完成的。

物理层整体的工作相对简单，主要的工作可以说是实现底层的驱动程序，只要底层驱动稳定工作，则对于物理层的功能实现就不是什么难事了。同样，物理层的测试工作也并不繁琐，如物理数据包的发送与接收：两个 ZigBee 节点选择相同的信道后，一个节点负责接收由另外一个节点发送来的数据，并显示在 PC 方即可。发送物理数据包的物理载荷部分也没有格式的要求，不需要去分析有效信息的比特流，而在 MAC 层的测试部分将会对相关 MAC 帧的比特流进行分析，是很复杂的。

## 4.7 本章小结

本章主要介绍了 IEEE 802.15.4 物理层协议及其主要功能的实现。物理层程序直接面向底层硬件，因而首先介绍了相关的硬件驱动和 MC13192 驱动部分。其中 MC13192 的硬件驱动部分介绍了 MC13192 的单字读写和巡回读写两种通信模式，并在这两种通信模式基础上实现了对 MC13192 寄存器和片上缓冲 RAM 的访问；MC13192 的驱动部分主要包括 MC13192 输出时钟的选择设置、运行状态的切换和 MC13192 产生 IRQ 中断的处理流程。输出时钟部分详细讲述了主控 MCU 如何使用 MC13192 的输出时钟以及为什么不采用内部时钟的原因；在运行状态的切换中介绍了 MC13192 的 7 种运行状态及相互切换的条件；IRQ 中断处理介绍了其基本的处理流程。

以此底层驱动为基础进一步设计实现了物理层的主要功能，包括 2.4GHz 频段 16 个物理信道的选择；物理数据包的接收、发送；能量检测和空闲信道评估。在本章的最后详细列出了物理层的主要功能函数，并给出了物理层的基本测试。

# 第五章 MAC 层协议分析与实现

本章介绍了 MAC 层协议的主要功能及其函数实现。IEEE 802.15.4 标准规定 MAC 层主要负责物理层无线信道的访问接入，即在物理层的接口函数基础上实现信道的有效访问。

本章首先介绍了 MAC 层的几个重要概念和 MAC 层的两种数据传输模型；接着详细阐述了 MAC 层主要功能的实现，包括各种 MAC 帧结构的封装、信道访问的冲突避免、设备间同步机制的实现、MAC 帧的收发；然后在给出了整个 MAC 层实现的主要函数列表的基础上，根据 IEEE 802.15.4 MAC 层的描述列出了论文所做的主要裁减工作；最后给出 MAC 层的相关测试工作。

## 5.1 MAC 层的几个重要概念

IEEE 802.15.4 规范中，信标帧(Beacon)、超帧(Superframe)等是几个出现频度很高的词。在具体阐述论文的 MAC 层工作之前，首先简要介绍一下常用的几个重要概念。

### 5.1.1 信标帧

IEEE 802.15.4 规定 MAC 帧可分为信标帧、数据帧、应答帧和命令帧四种，其中信标帧只可由 PAN 协调者或路由器负责组织、发送。信标帧主要包含了协调者或路由器自身的基本信息。对于由协调者组织的信标帧还包含了超帧的起始时隙以及网络其他设备是否在协调者中存有数据等信息。信标帧的作用主要是在子设备加入 PAN 时能让子设备获取父设备的基本信息，其中主要包括父设备的网络标识、网络地址和 MAC 地址。在支持超帧结构的 PAN 中，信标帧作为超帧的起始标识，也可用于实现设备间的同步。

### 5.1.2 超帧

与信标帧、数据帧、应答帧和命令帧不同，超帧只是一种对时间进行管理的机制，不是数据信息的组织结构，因而并不是一个实际意义上的帧，只是一种时间上的概念。在支持超帧结构的 PAN 中，可利用超帧对数据的通信时间进行严格的划分。超帧的结构是由 PAN 协调者来规定的。

在IEEE 802.15.4中，可以选用以超帧为周期来组织低速率无线个域网(low rate wireless person area network，LR-WPAN)网络内设备间的通信。每个超帧都以网络协调器发出信标帧为起始，在这个信标帧中包含了超帧将持续的时间以及对这段时间的分配等信息。网络中其他设备接收到超帧开始时的信标帧后，就可以根据其中的内容安排自己的任务，例如进入休眠状态直到这个超帧结束或等待接收数据。

在支持超帧结构的 PAN 中，可利用超帧结构来有效的控制无线信道的访问。超帧将通信时间划分为活跃(Active)和不活跃(Inactive)两个部分。在不活跃期间内，PAN 网络中的设备不会相互通信，从而可以进入休眠状态以节省能量。具体的超帧结构如图 5-1[32]所示。

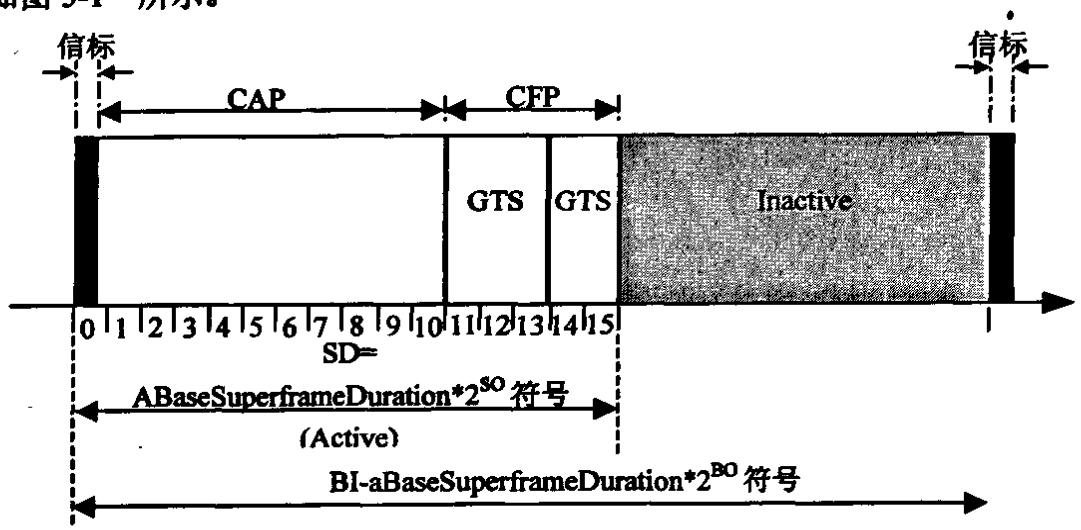


图 5-1 超帧结构

超帧结构的时间长度主要由属性变量 macBeaconOrder 和 macSuperframeOrder 的值来描述。macBeaconOrder(BO)，信标序号描述了协调者发送信标帧的间隔(beacon interval,BI)，即整个超帧的时间长度(包括活跃部分和不活跃部分)；macSuperframeOrder(SO)用以描述超帧活跃部分的时间长度(superframe duration,SD)，其中信标帧的发送是包含在活跃部分中的。

超帧的活跃期划分为三个阶段：信标帧发送阶段、竞争访问阶段（contention access period，CAP）和非竞争访问阶段（contention-free access period，CFP）。超帧的活跃期又可被划分为 16 个等长的时隙，每个时隙的长度、竞争访问期包含的时隙数等参数，都是由协调者决定的，并通过信标帧广播到整个网络。

信标帧之后，紧接着是竞争访问阶段，竞争期以超帧起始时隙为边界，在非竞争期开始之前结束。如果非竞争期长度为 0，则竞争期就一直到超帧结束处。除确认帧和紧跟在数据请求命令确认帧之后的任何数据帧，在竞争期内，发送的所有帧都使用带时隙的 CSMA-CA 机制访问信道。MAC 层命令帧总是在竞争期内发送的。

从竞争期结束到活跃期结束，这段时间为非竞争期。若 PAN 协调者已经分配了保护时隙，则保护时隙应位于非竞争期之内，并占用连续的时隙。在非竞争期，数据的传输不使用 CSMA-CA 机制访问信道，设备均是在指定的时隙上发送数据，从而避免了冲突访问。

### 5.1.3 保护时隙

在支持超帧结构的 PAN 中，对于超帧中的非竞争期间(CFP)，在时间的分配上采用保护时隙(guaranteed time slot，GTS)的方式。这一期间，协调者根据网络设备的 GTS 请求来进行时隙的分配，但在整个非竞争期间最多只能为 7 个设备分配 GTS，即最多只包含 7 个 GTS，其中每一个 GTS 至少占用 1 个时隙。

## 5.2 MAC 层的数据传输模型

ZigBee 网络中主要有 2 种数据传输方式[32]：信标使能通信(beacon-enabled)和非信标使能通信(non beacon-enabled)。

### 5.2.1 信标使能通信

在信标使能的网络中，网络协调者定时广播信标帧，PAN 网络中的设备都通过协调者发送的信标帧来进行同步。图 5-2 示意了信标使能的数据传输方式。

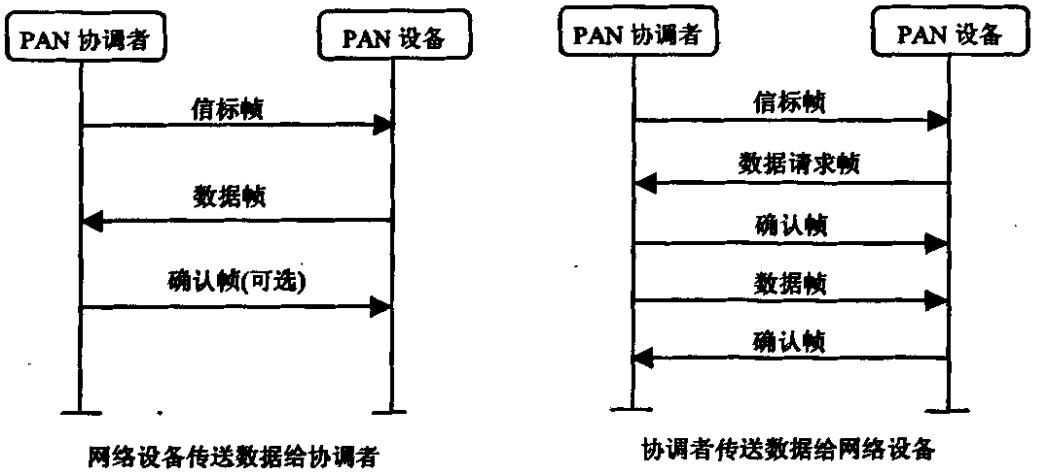


图 5-2 信标使能的数据传输模型

当设备要发送数据给协调者时，它首先侦听网络中的信标帧。如果接收到信标帧，表示与协调者实现了同步，它就使用 CSMA-CA 机制选择一个合适的时机把数据发送给协调者，协调者成功接收到数据后，可以选择回送一个确认帧表示成功收

到了该数据。

当协调者要向某一网络设备发送数据时，就在下一个信标帧中说明协调者拥有属于该设备的数据正在等待发送。目的设备在周期性的侦听过程中会接收到这个信标帧，从而得知有属于自己的数据保存在协调者中。这时该设备就向协调者发送数据请求帧，协调者接收到该帧后回送一个确认帧，然后开始发送所保存的数据。设备成功接收到数据后，再回送一个确认帧，协调者收到后确认目的设备已成功接收了数据，则将保存的数据移除。

### 5.2.2 非信标使能通信

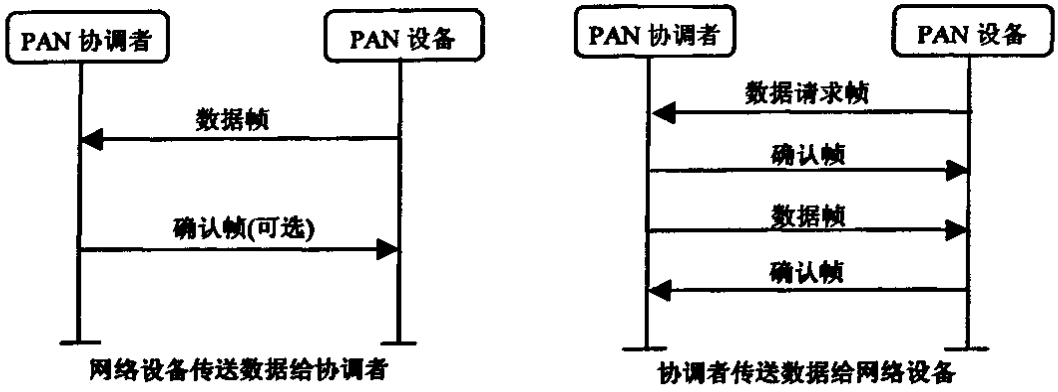


图 5-3 非信标使能的数据传输模型

对于采用非信标使能的网络，网络协调者不定时发送信标帧。每当设备要发送数据时，它首先等待一段随机长的时间，然后开始检测信道状态，若信道空闲，该设备开始发送数据；若信道忙，则设备需要重复等待一段时间后并再次检测信道，直到能够发送数据为止。为简化实现协议，本设计采用非信标使能的数据传输方式。图 5-3 为非信标使能的数据传输模型。

当网络设备要向协调者发送数据时，不再需要定位信标帧，直接采用 CSMA-CA 机制选择发送时机将数据发送出去。协调者成功接收后，可根据需要回送确认帧表示成功接收。

协调者有其他网络设备的数据时将数据保存，等待网络设备的数据请求；当接收到设备的数据请求命令帧后，协调者回送确认帧后，紧接着发送该设备的数据；设备再次给协调者回送确认帧表示当前保存的数据可以移除了。

## 5.3 MAC 帧结构与组织实现

帧的封装实际上就是按照各种类型的帧结构，通过程序来组织实现相应类型的

帧。下面具体介绍各种类型帧的结构，并在此基础上给出部分实现函数头说明。

### 5.3.1 MAC帧的基本结构

MAC层的帧类型主要包括信标帧、数据帧、应答帧和命令帧四种。每种MAC帧都包括MAC帧头(MHR)、MAC帧载荷(MAC payload)和MAC帧尾(MFR)三个基本组成部分，如表5-1所示[32]。其中对于各类帧，帧头和帧尾的结构都是相同的。

表5-1 MAC层帧结构

| 2字节 | 1字节 | 0/2字节 | 0/2/8字节 | 0/2字节 | 0/2/8字节 | 可变长度 | 2字节 |
|---|---|---|---|---|---|---|---|
| 帧控制信息 | 帧序列号 | 目的设备PAN标识符 | 目的地址 | 源设备PAN标识符 | 源地址 | 帧载荷 | 帧校验 |
| | | 地址信息 | | | | | |
| MAC帧头 | | | | | | MAC帧载荷 | MAC帧尾 |

(1) 帧控制信息

帧控制信息域为2字节长，包括帧类型的定义、地址子域和其他控制标志。帧控制信息域的结构如表5-2所示。

表5-2 帧控制信息域结构

| 位0-2 | 3 | 4 | 5 | 6 | 7-9 | 10-11 | 12-13 | 14-15 |
|---|---|---|---|---|---|---|---|---|
| 帧类型 | 安全允许 | 帧未处理 | 确认请求 | 内部PAN | 保留 | 目的地址模式 | 保留 | 源地址模式 |

帧类型，用来标识不同类型的帧。以$b_2b_1b_0$为顺序，000表示信标帧；001表示数据帧；010表示确认帧；011表示命令帧；100~111保留。

安全允许，若MAC层没有对该帧进行加密保护，则该位为0；若该位为1，则表示该帧已通过相应的加密算法进行了加密处理，在解析该帧之前应通过相应的解密算法来进行解密处理。

帧未处理，如果发送设备在当前帧发送完之后还有帧要发送给同一个接收方，则该位应置为1；否则为0。

确认请求，表示当接收到数据帧或MAC命令帧时，接收方是否需要发送确认帧，该位为1，表示需要确认。

内部PAN，指定将该MAC帧是在个域网内传输的，还是传输到其他个域网。

目的地址模式和源地址模式，表示发送方和接收方在该帧中采用的地址模式。以$b_1b_0$为顺序，00表示地址域不存在；01保留；10表示使用16位的网络地址；11表示使用64位的MAC地址。

(2) 帧序列号

帧序列号域为1字节长，是MAC帧唯一的序列标识符。对于信标帧该域为信标序号(beacon sequence number,BSN)；对于数据帧、确认帧或命令帧，该域为数据

序号(data sequence number,DSN)。每生成一帧，该域依次加1。

(3) PAN标识符

目的设备或源设备PAN标识符域为2字节长，描述了接收或发送设备所在PAN的唯一标识。当目的PAN标识符域为0xFFFF时表示该帧为广播帧。

(4) 地址域

目的或源地址域为2字节或8字节长，其长度由帧控制子域的地址模式的值决定。8字节的长地址为设备所固有的MAC地址，2字节的短地址是在设备加入PAN时由协调者所分配的网络地址。当目的地址域为0xFFFF时表示该帧为广播帧，所有侦听信道的设备均可以接收到该帧。

(5) 帧载荷

帧载荷域的长度是可变的，不同的类型帧该域包含不同的内容。这一部分参考具体类型的帧内容。

(6) 帧效验

帧效验域(frame check squence,FCS)为2字节长，包含了16位的ITU-T CRC码。该域是由MAC帧头和MAC帧载荷两部分的位序列经过CRC16计算得到的[47]。

### 5.3.2 信标帧

信标帧的帧头域按上节所讲述的格式进行相应的设置，如帧类型子域设为Beacon类型(000)；若信标帧使用安全机制，则安全允许位为1。值得注意的是，在地址子域中仅包含源地址域，目的PAN标识符和目的地址域都为空。

信标帧的载荷域结构如表5-3所示[32]。

表5-3 信标帧载荷域结构

| 字节 | 可变长度 | | | 可变长度 | | 可变长度 |
|---|---|---|---|---|---|---|
| 超帧描述符 | 1字节 | 0/1字节 | 可变长度 | 1字节 | 可变长度 | 信标载荷 |
| | 保护时隙描述符 | 保护时隙方向掩码 | 保护时隙列表 | 未处理地址描述 | 未处理地址列表 | |
| | 保护时隙域 | | | 未处理地址域 | | |

超帧描述符子域为2字节长，主要定义了超帧的基本结构。其具体内容主要包括了信标的序列号、超帧的序列号、最后竞争期的时隙、是否为协调者的标志和是否允许连接的标志等信息。

保护时隙描述域为1字节长，定义了GTS的基本信息，包括GTS描述符计数和GTS允许标志。GTS描述符计数，指定GTS列表域中GTS描述符的个数；GTS允许标志，表示PAN协调者是否接收GTS的分配请求。

保护时隙方向掩码域为 1 字节长，该域中只有前 7 位有效，每一位对应 1 个 GTS(因为最多只有 7 个 GTS，所以只需要 7 位)。该域主要描述 GTS 的方向(接收还是发送)，当某位为 1 时表示对应的 GTS 为接收 GTS，否则为发送 GTS。

保护时隙列表包括所有 GTS 描述符。GTS 描述符主要包括该 GTS 所属设备的网络地址、GTS 的起始时隙和 GTS 的长度(即连续占用的时隙数)等信息。

未处理地址描述域为 1 字节长，指定了在协调者中存在未处理数据所对应设备的个数，其中包括 16 位网络地址和 64 位 MAC 地址两种地址方式的所有设备数。

未处理地址列表域的长度主要由未处理地址描述域中未处理网络地址数和未处理 MAC 地址数共同决定，但协调者中最多只包含 7 个设备的未处理数据，即未处理网络地址数与未处理 MAC 地址数的和不可以大于 7。该地址列表包含了当前需要与协调者传输未处理或等待消息的设备的地址列表，其中所有的网络地址信息均排列在 MAC 地址之前，对于广播地址 0xFFFF 不包括在地址列表中。

信标载荷域为一个可变序列，其内容来源于网络层的数据单元。信标载荷域的目的是为了顺带一定的上层信息，但一般情况下，该域为空。

### 5.3.3 数据帧

数据帧的结构比较简单，与 MAC 帧基本结构是一致的，帧类型子域设为数据帧类型(001)，其余字段，如安全允许位、目的和源地址模式、目的和源地址域都根据具体的需求进行相应的设置。帧载荷域是由上层传来的有效信息，主要指网络层传来的网络报文。

### 5.3.4 应答帧

应答帧的结构是几种帧类型中最简单的，在 MAC 帧的基本结构基础上进一步简化。

整个应答帧只有 5 字节长，包括 2 字节的帧控制信息域、1 字节的帧序列号和 2 字节的帧校验码。帧控制信息域中的帧类型子域为应答帧类型(010)；安全允许位根据要求设置；其余字段，如目的和源地址模式、目的和源地址域均设置为 0。应答帧不存在目的和源地址域；帧序列号是根据收到帧的序列号来设置的；应答帧没有载荷域；帧校验码是计算所得到的 16 位 CRC 校验码。

### 5.3.5 命令帧

命令帧在MAC层协议中是非常重要的，通过各种命令帧实现网络的连接、断开、信道的有效访问等功能。命令帧结构也是几种帧类型中最复杂的。

IEEE 802.15.4协议规定：MAC命令帧载荷域的第1个字节为命令类型域。其中MAC命令帧共有9类：0x01为连接请求命令；0x02为连接响应命令；0x03为断开连接通知命令；0x04为数据请求命令；0x05为PAN ID冲突通知命令；0x06为孤立点通知命令；0x07为信标请求命令；0x08为协调者重新调整命令；0x09为GTS请求命令；其余保留，不作使用。

根据实际需求，为降低协议实现的复杂度，在MAC命令帧实现时，进行了一定的裁减，只实现了连接请求、连接响应、断开连接通知、数据请求、PAN ID冲突通知和信标请求这6种类型的命令帧。

这6种命令帧在格式上差异并不是很大，下面就以连接请求命令为例介绍命令帧的具体结构。

ZigBee设备通过连接请求命令请求与其他具有连接能力的ZigBee设备建立连接。连接请求命令帧的格式如表5-4所示。

表5-4 连接请求命令帧结构

| 17/23字节 | 1字节 | 1字节 | 2字节 |
|---|---|---|---|
| 帧头域信息 | 命令帧标识符 | 性能信息 | 帧校验 |

帧头域与MAC帧基本结构一致，其中关于目的设备的信息与要连接的对象一致；IEEE 802.15.4规定连接命令的源地址采用MAC地址模式，即源地址模式域应设置为3；源PAN ID域按规定设置为0xFFFF。

命令帧标识符为连接请求命令，即0x01。

性能信息字节主要包括了发起连接请求命令设备的某些属性。其中第0位表示该设备是否具有能够成为PAN协调者的能力；第1位表示设备类型，是完整功能设备还是简化功能设备；第2位表示采用交流电源标志；第3位表示本设备在空闲时是否打开接收机；4、5位保留；第6位表示设备是否采用安全加密算法来对帧处理；第7位表示设备与目的设备建立连接后是否希望被分配一个16位的网络地址，当该位为0时，则目的设备将特殊的网络地址0xFFFE分配给该设备。

### 5.3.6 MAC帧的组织实现

对于上述的MAC帧在程序上的组织实现实际上就是提供必要的入口参数，按照各种类型帧的结构逐个字段的组织连接。下面以实现信标帧的函数头为例简要说明帧的组织，对于其他类型帧的组织在流程上基本相同。

```
/*MLMEMakeBeaconFrame:生成 Beacon 帧-----------------------------------------*
*功  能:生成 Beacon 帧                                                      *
*参  数:1: IsSecurity,该帧是否加密                                          *
*       2: Dest,目标设备的基本信息                                          *
*       3: macFinalCAPSlot:超帧结构中 CAP 阶段的最后 Slot                   *
*       4: GTSField: Beacon 帧 GTS 域                                       *
*       5: PendingAddr,Beacon 帧中 Pending Address 域                       *
*       6: BeaconPayload,Beacon 帧中接收上层来的数据域                      *
*       7: macData,用于返回生成的 Beacon 帧                                 *
*返  回:帧长度                                                              *
*备  注:1:16 位的 CRC 校验码在发送时按实际发送的位序列生成                  *
*       2:由于 macData 在该函数内使用 malloc 分配空间,所以在主调函数要进行释放 *
*--------------------------------------------------------------------------*/
UINT8 MLMEMakeBeaconFrame(Bool IsSecurity,UINT8 macFinalCAPSlot,GTSFieldStruct
    GTSField,PendingAddrStruct PendingAddr,MACPayloadStruct BeaconPayload,
    UINT8 **macData)
```

通过对信标帧结构的研究，信标帧生成函数主要有 7 个入口参数：该信标帧是否要加密、目的设备信息、超帧中 CAP 的长度、GTS 数据、未处理地址数据、网络层来的 Beacon 载荷和存放生成的信标帧的缓冲 macData。实际上该函数还需要本设备的相关信息，这些参数是通过全局变量来获取的。在取得上述必要的参数后剩下的工作就是按照信标帧的结构逐段组织。最终生成的信标帧存放在 macData 中来返回，供进一步的使用。生成帧的长度作为函数返回值。

需要注意的是在函数实现中，16 位的 CRC 校验码在此函数中并没有生成，而是在发送帧时计算得到的，并附加在帧尾；另外由于帧的长度是不可预知的，因而用于存放帧的 macData 采用二次指针的形式，在本函数体内来动态分配，这样在调用该函数结束后，主调函数需要释放 macData 的空间。

## 5.4 信道访问

在 ZigBee 无线网络中主要通过冲突避免载波检测多路访问(CSMA-CA)算法来实现物理信道的有效访问。另外在支持超帧结构的 PAN 中，采用超帧结构来进一步协调节点间对物理信道的访问。但在本设计中为简化设计复杂度，信道访问不支持超帧机制。

由于在无线传输中无法实现冲突检测，因而 ZigBee 网络采用与 802.11 相同的 CSMA-CA 算法来实现物理信道的访问。节点在发送数据前，为避免冲突，即使在信道空闲情况下，也会推迟一段时间才发送数据。

在竞争期内发送数据或 MAC 命令帧之前，将使用 CSMA-CA 算法来进行信道访问。如果 PAN 支持超帧结构，将使用时隙 CSMA-CA 算法，若不支持超帧，则使用非时隙 CSMA-CA 算法来发送帧。在时隙 CSMA-CA 算法中，PAN 中每个设备

退避周期的边界都与 PAN 协调者超帧时隙的边界一致，即每个设备第一个退避周期的起始与信标帧传送起始的时间一致。而非时隙 CSMA-CA 中，设备的退避周期与 PAN 中任何其他设备的退避周期是没有关系的。CSMA-CA 算法的基本流程如图 5-4 所示[32]。

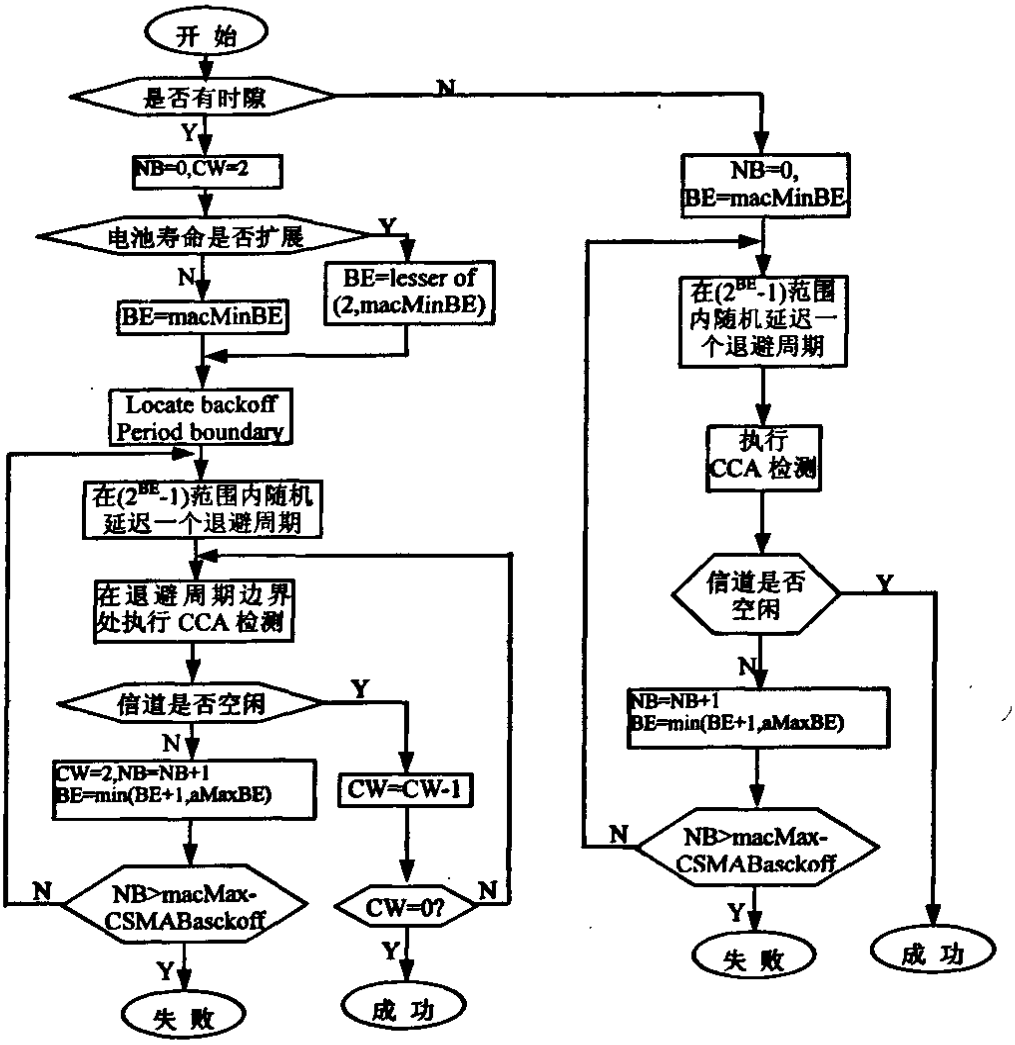


图 5-4 CSMA-CA 算法流程

每个设备为发送任务保存 3 个变量：NB、CW 和 BE。NB 是在执行当前发送任务时，CSMA-CA 算法所需要进行退避的次数，在每次执行新的发送之前，这个值初始化为 0；CW 是竞争窗口长度，在传送开始之前确定实现信道活动空闲前退避的次数，在每次传送开始之前，该值初始化为 2，并且在每次信道访问为忙的时候复位为 2，CW 变量仅用于时隙 CSMA-CA；BE 是退避指数，与设备在试图访问信道之前，需要等待的退避周期有关，在非时隙系统中，或 macBattLifeExt 设置为 FALSE 的时隙系统中，BE 初始化为 2 和 macMinBE 中的较小值。

## 5.5 设备与协调者间的同步机制

在 IEEE 802.15.4 协议中，设备与协调者之间同步的实现主要有两种方式：一种是利用信标帧来实现同步；另外就是上拉数据方式的同步机制。

### 5.5.1 信标同步机制

在支持超帧的网络中，所有设备均通过获得超帧的起始信标帧来实现与协调者的同步。

在支持超帧的网络中，协调者会定时广播信标帧。网络设备为与协调者同步，打开接收器，在一定的时间内跟踪信标帧。当接收到信标帧后，设备验证该信标帧是否来自它所在网络的协调者；如果接收到的信标帧的源地址与其所在 PAN 的协调者不一致，则丢弃该信标；若信标帧有效，则检查信标帧内是否有自己的数据，若有，则在下一个超帧指定的时隙内与协调者通信。这样就实现了设备与网络协调者之间的同步。

### 5.5.2 上拉数据方式

在不支持超帧的网络中，设备主要通过向协调者上拉数据的方式来实现同步。

设备通过数据请求命令帧来向协调者上拉数据。当设备发送完数据请求命令帧后，将在一段时间内轮询协调者，等待从协调者来的数据；若协调者存有该设备的数据，在接收到数据请求命令帧后，将发送该设备的数据。

在本设计中，因为不支持超帧，所以采用该方式实现同步机制。

## 5.6 MAC 帧的发送与接收

MAC 帧的发送相对比较简单。有数据要发送时，按照所属的帧类型进行组织，实际上帧的组织是由上层协议，即网络层来调用的；组织好后按非时隙 CSMA-CA 机制进行信道的检测；当信道可用时，调用物理层的发送程序将该帧作为物理载荷发送出去即可。

在物理信道上，设备会接收到相同频率的所有物理层数据。因此，在 MAC 层中，要对接收到的 MAC 帧进行处理，过滤掉不是自己的和不符合要求的 MAC 帧。下面主要介绍对收到帧所做的过滤处理。

设备的物理层接收到数据后传递给 MAC 层，MAC 层对收到的 MAC 帧进行有

效性检查，执行流程如图5-5所示。

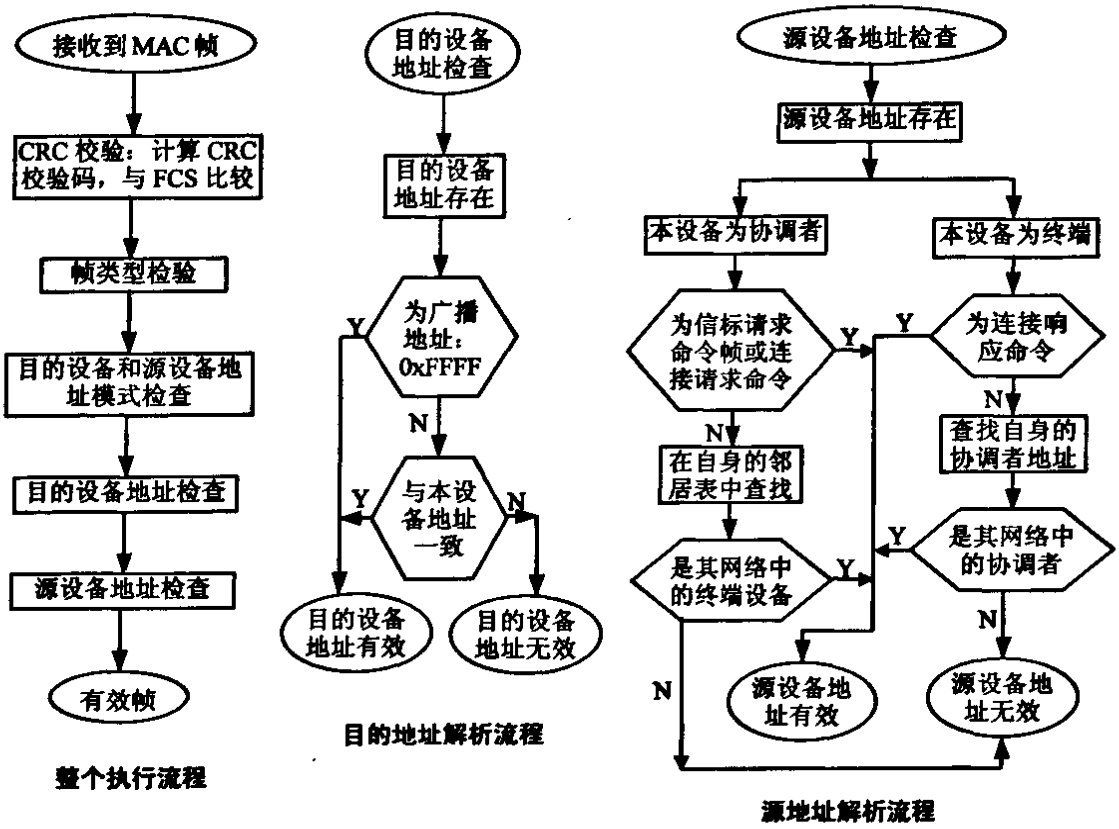


图5-5 MAC帧有效性检查流程

接收到MAC帧后，首先计算除FCS帧尾域以外的所有字段的CRC校验码，将计算得到的16位CRC校验码与FCS比较，看是否相等，若不相等，则帧在传输过程中发生了错误，丢弃该帧；相等，再检查帧的类型域、目的设备和源设备的地址模式是否符合要求；均符合要求再检查目的设备地址和源设备地址，若检查都有效，则该帧是一个有效的MAC帧。

对于目的设备地址的检查较简单，若目的设备地址域是广播方式，即0xFFFF，则不需要检查；否则目的设备的地址一定要与本设备自身的地址一致才有效。

源设备地址的检查要视设备类型的不同而不同。当设备为协调者时，对于终端设备的信标请求和连接请求命令不做地址检查，这主要是考虑到在终端设备加入网络之前，协调者并不知道终端设备的信息，因而在信标请求和网络连接请求时不做源设备地址检查；当不是这两种命令帧时，源设备必定已在本PAN中，则该终端设备地址信息存储在邻居表中，查找邻居表，看是否有效。当设备为终端设备时，若是连接响应命令帧，则不检查源设备地址信息，因为在连接响应前，设备并不知道网络协调者的地址信息；对其他类型帧，终端设备只接收其网络协调者的数据。

检查 MAC 帧合法后，则根据帧的不同类型进行不同处理：若为命令帧，根据命令类型直接调用相应的命令处理函数对接收帧进行处理；若该帧为数据帧，则直接将帧头、帧尾去除，将数据载荷域作为网络报文传递给网络层，交由网络层处理。

表 5-5 MAC 层函数列表

| 函数类型 | 函数编号 | 函数头 | 函数功能简述 |
|---|---|---|---|
| 内部功能函数 | 1 | Bool UnSlotCSMACA(UINT8 macMinBE, UINT8 macMaxCSMABackoffs) | 无时隙的CSMA/CA算法实现 |
| | 2 | UINT16 CRCCaculate(UINT8 *ReData,UINT8 Len) | 根据数据按位计算CRC校验码 |
| | 3 | UINT8 IsMACFrameAvail(PHYPacket MACFrame) | 对接收到的帧进行有效性校验 |
| | 4 | UINT8 MCPSCmdProcess(void) | MAC层命令帧处理函数 |
| | 5 | UINT8 AssociateReqCmdProcess(void) | 对连接请求命令帧的处理 |
| | 6 | UINT16 AllocateNWKAddr(void) | 分配一个可用的16位网络地址 |
| | 7 | UINT8 MCPSDataProcess(void) | MAC层数据帧处理函数 |
| | 8 | UINT8 DisAssocaiteNoticeCmdProcess(void) | 对断开连接命令帧的处理 |
| 外部功能函数 | 1 | void MACInit(void) | MAC层初始化工作 |
| | 2 | Bool MLMEEnableRx(void) | MAC层接收允许 |
| | 3 | Bool MLMEDisEnableRTX(void) | MAC层接收禁止 |
| | 4 | UINT8 MLMESendMACFrame(UINT8 len, UINT8 *SendData) | 通过PHY层发送MAC帧 |
| | 5 | UINT8 MLMEMakeBeaconFrame(Bool IsSecurity, UINT8 macFinalCAPSlot,GTSFieldStruct GTSField,PendingAddrStruct PendingAddr, MACPayloadStruct BeaconPayload,UINT8 **macData) | 生成信标帧 |
| | 6 | UINT8 MLMEMakeDataFrame(Bool IsSecurity, Bool IsFramePending,Bool IsACK,Node_info Dest, MACPayloadStruct DataPayload, UINT8 **macData) | 生成数据帧 |
| | 7 | UINT8 MLMEMakeACKFrame(Bool IsFramePending, UINT8 ACKSN,UINT8 **macData) | 生成应答帧 |
| | 8 | UINT8 MLMEMakeAssociateReqCmdFrame( Bool IsSecurity,Bool IsFramePending, Node_info Dest,UINT8 **macData) | 生成连接请求命令帧 |
| | 9 | UINT8 MLMEMakeAssociateResponseCmdFrame( Bool IsSecurity,Bool IsFramePending, Node_info Dest, UINT16 nwkDevAddr, UINT8 state,UINT8 **macData) | 生成连接响应命令帧 |
| | 10 | UINT8 MLMEMakeDisAssociateNoticeCmdFrame( Bool IsSecurity,Node_info Dest, UINT8 Reason,UINT8 **macData) | 生成断开连接通知命令帧 |
| | 11 | void MLMEFrameIndication(PHYPacket MACFrame) | 物理层向MAC层报告接收到帧 |
| | 12 | UINT8 MLMEFrameProcess(void) | 处理MAC帧 |
| | 13 | UINT8 AssociateResponseCmdProcess( UINT16 nwkFatherAddress) | 对连接应答命令帧的处理 |

## 5.7 MAC层函数列表

MAC层功能设计实现的函数都包含在Zig_MAC.c文件中。在本层的程序设计上将内部调用函数和供给上层调用的函数清晰的分开，这样可以更好地实现与上层函数的关联。MAC层主要功能函数列表如表5-5所示

其中内部函数主要负责内部功能的实现，如CSMA/CA冲突避免机制、CRC码的计算与验证、16位网络地址的计算等功能，这些功能的实现对于上层函数是不需要了解的；而外部功能函数则是直接供上层调用的，如各种类型帧结构的组织、帧的收发以及对各种的帧的相关处理等。

## 5.8 MAC层的主要裁减工作

参照MAC层所实现的函数列表，并根据IEEE 802.15.4规范，表5-6给出了本文所做的主要裁减内容。由于不支持超帧，因而同时也不支持保护时隙；在信道访问机制上采用的是无时隙的CSMA_CA机制；为简化协议实现，在设备同步机制中采用上拉数据的同步方式；信道扫描采用ED信道扫描方式；同时裁减的协议栈不支持PAN ID冲突检测和MAC层的安全机制。

表5-6 MAC层主要裁减内容

| 802.15.4规范主要实现功能 | 具有功能 | 本文实现 | 802.15.4规范主要实现功能 | 具有功能 | 本文实现 |
|---|---|---|---|---|---|
| 超帧 | 超帧结构的实现 | × | 保护时隙机制 | 时隙的分配与管理 | × |
| 各种类型的帧 | 信标帧 | √ | 信道访问机制 | 无时隙的CSMA_CA机制 | √ |
| | 数据帧 | √ | | 时隙CSMA_CA机制 | × |
| | 应答帧 | √ | 设备同步机制 | 信标同步机制 | × |
| | 连接请求命令帧 | √ | | 非信标同步机制 | √ |
| | 连接响应命令帧 | √ | 信道扫描 | ED信道扫描 | √ |
| | 断开连接通知命令帧 | √ | | 主动信道扫描 | × |
| | 数据请求命令帧 | √ | | 被动信道扫描 | × |
| | PAN ID冲突通知命令帧 | √ | | 孤立信道扫描 | × |
| | 孤立点通知命令帧 | × | PAN ID冲突检测 | PAN ID冲突检测 | × |
| | 信标请求命令帧 | √ | 帧安全机制 | 安全模式加密机制的实现 | × |
| | 协调者重新调整命令帧 | × | | 非安全模式 | √ |
| | GTS请求命令帧 | × | | | |

## 5.9 MAC层测试及心得

对于MAC层的测试实际上应该需要一个MAC层分析仪。目前市场上也有几家公司推出了相应的分析仪，如深圳英蓓特信息技术有限公司就开发了MAC层的

分析仪，并配有 PC 方的软件，可以自动分析出所捕捉到的各种帧格式信息。但由于价格偏高，本课题并没有使用分析仪进行 MAC 层的设计与测试。

本文对 MAC 层的测试，首先构造网络层的调用命令，然后调用 MAC 层功能函数来组织相应帧结构；MAC 层最终通过物理层，将生成物理数据包在指定信道上发送出去；另外一个设备在相同物理信道上接收该物理数据包，并通过串行通信方式将接收到的物理载荷域，即 MAC 帧信息发送到 PC 方显示，注意该设备在软件上只含有物理层协议，它只负责显示发送设备所组织的帧信息，并不交给上层处理；然后通过显示的位流信息按照相应帧格式仔细分析，核对组织的结构是否正确。

以设备加入网络为例，该功能是网络层所拥有的，但真正的核心内容是由 MAC 层来实现的。首先构造加入网络的函数，在该函数内调用 MAC 层的连接请求命令生成函数，MAC 层内部最终将连接请求命令帧通过物理层发送出去；另外一个物理层设备将接收到的帧信息显示在 PC 方，其中包括了完整的帧头、帧尾和 MAC 有效载荷信息，再对照连接请求命令帧的结构仔细分析，检查结构是否正确。

论文所采用的 MAC 层测试方法并不是很合理，利用获得的位流来分析结构的正确与否很容易出错，并且也不易判断。希望可以在进一步的工作中开发出 MAC 层的截帧工具，帮助开发人员分析帧结构的正确与否。

合理设计各种类型帧的数据结构是程序设计中一个较重要的环节，合理的数据结构将给程序开发带来很大的方便，附录 A 给出了论文中主要数据结构的详细设计。嵌入式软件的调试工作与 PC 架构的程序开发调试也有所不同，最终生成的二进制文件需要下载到特定的 MCU 中，本文中也就是 GB60，对于程序的跟踪控制则不太方便。在实际的开发中，通常采用某一种与 PC 的通信接口，在函数代码中逐段加入通信语句，在程序运行时输出想要观察的变量，看是否是所希望的值，这样来逐步定位到问题的所在。

## 5.10 本章小结

本章在具体设计 MAC 层功能之前，介绍了信标帧、超帧和保护时隙等几个常用的概念；接着讲述了协调者与设备之间在信标使能传输模型和非信标使能传输模型下是如何通信的，本文采用的是信标使能传输方式；接着在详细介绍了 MAC 帧基本结构的基础上，阐述了论文所实现的各种帧的结构，并以信标帧为例简单介绍了在程序上的组织实现；多数无线通信的信道访问采用的都是 CSMA_CA 机制，ZigBee 技术也同样如此，文中具体介绍了 CSMA_CA 算法的实现逻辑；在设备的

同步机制上，本文采用的是上拉数据方式的同步机制；MAC 帧的发送相对简单，将由上层发送来的数据装载到有效载荷中，再按相应格式组织并发送出去；对于 MAC 帧的接收，则需要过滤那些无效的帧，本章详细阐述了帧接收的检查流程；本章在列出了论文所实现 MAC 层主要函数的同时，通过与 IEEE 802.15.4 对比，也归纳了本文在实现 MAC 层上所做的裁减工作；最后，给出了对 MAC 层的测试与一些心得体会。

# 第六章　网络层协议分析与实现

在 ZigBee 协议中，网络层主要负责新建网络、加入网络、退出网络和网络报文的路由传输等功能。本章在介绍网络报文结构的基础上，重点阐述网络层的这些主要功能。

## 6.1　网络报文结构

在 ZigBee 协议中，网络报文包括数据报文和命令报文两种类型。下面重点介绍网络报文的结构，对于报文结构的组织实现，与 MAC 帧实现的构思基本一致，就是按照报文结构逐段组织，就不再讲述了。

### 6.1.1 网络报文的基本结构

如表 6-1 所示，网络报文主要包括控制信息域、网络路由域和报文载荷域[33]。

表 6-1 网络报文结构

| 2字节 | 2字节 | 2字节 | 0/1字节 | 0/1字节 | 可变长度 |
|---|---|---|---|---|---|
| 报文控制信息 | 目的设备网络地址 | 源设备网络地址 | 广播半径 | 广播序列号 | 报文载荷 |
| | 路由域 | | | | |
| 网络报文头域 | | | | | 网络层有效载荷 |

(1) 报文控制信息

报文控制信息域长 2 字节，包括报文类型的定义、协议版本及其他控制标志。控制信息域的具体结构如表 6-2 所示。

表 6-2 报文控制信息域结构

| 位0~1 | 2~5 | 6 | 7~8 | 9 | 10~15 |
|---|---|---|---|---|---|
| 报文类型 | 协议版本 | 发现路由 | 保留 | 安全 | 保留 |

报文类型，用来标识不同类型的报文。在 ZigBee 网络中只存在数据报文和命令报文两种类型。以 $b_1b_0$ 为顺序，00 表示数据报文；01 表示命令报文；10 与 11 保留不作使用。

协议版本，该域反映了当前协议所使用的版本号。目前 ZigBee 协议的版本号为 0x01。

发现路由，表示该报文是否支持网络路由。该位为 0 表示不支持网络路由；为 1 表示支持网络路由。

安全，表示网络层是否采取了报文的安全处理。

(2) 目的设备网络地址

该域为 2 字节长，表示该网络报文要发送的目的设备。当该报文为网络广播报文时，该域值为 0xFFFF。

(3) 源设备网络地址

2 字节的源设备网络地址域给出了发送该报文的设备的网络地址。

(4) 广播半径

当报文的目的网络地址为广播地址(0xFFFF)时，广播半径域存在。该域表示广播报文传输的范围，每个接收设备接收一次该报文，则将该域减 1，当减到 0 后，该报文不再允许传输。

(5) 广播序列号

当广播半径域存在时，广播序列号存在。该域规定了广播报文的序列号，每传送一个新的广播报文，该序列号加 1。

(6) 报文载荷

报文载荷域的长度是可变的，针对不同类型的报文，该域的内容是不同的。

### 6.1.2 数据报文

数据报文的结构很简单，与网络报文的基本结构是一致的，报文载荷域存放的就是由上层传输来的有效数据。

### 6.1.3 命令报文

ZigBee 协议规定网络层命令报文有 3 种：路由请求命令、路由应答命令和路由错误命令。网络命令报文的网络报文头域与网络报文的基本结构一致，不同的就是在载荷域中标识不同命令信息。本文实现了路由请求命令和路由应答命令两种。

(1) 路由请求命令

路由请求命令是用于网络设备向其 POS 范围内的其他设备请求目的设备的网络路由信息，以便建立到达目的设备的路由信息。路由请求命令报文的报文载荷域格式如表 6-3 所示。

ZigBee 协议规定路由请求命令中目的地址为广播地址，因而在网络报文头域中，目的设备地址域为 0xFFFF。

命令类型标识给出了该命令报文的类型，对

表 6-3 路由请求命令报文结构

| 1 字节 | 1 字节 | 1 字节 | 2 字节 | 1 字节 |
|---|---|---|---|---|
| 命令类型标识 | 命令选择 | 路由请求标识符 | 目的地址 | 路由开销 |
| 报文载荷域 | | | | |

于路由请求命令，该域为 0x01。

命令选择字节中只有第 7 位有效，为路由维护位。当该域为 1 时，表示需维护当前路由信息。

路由请求标识符为 1 字节的路由请求序列号，每发送一次路由请求命令，该值加 1。

2 字节的目的地址域为路由请求命令的目的设备的网络地址。

路由开销字节用来记录路由传输的开销。

(2) 路由应答命令

路由请求命令的目的设备使用路由应答命令报文来对源设备做出响应。路由应答命令报文的报文载荷域结构如表 6-4 所示。

表 6-4 路由应答命令报文结构

| 1 字节 | 1 字节 | 1 字节 | 2 字节 | 2 字节 | 1 字节 |
|---|---|---|---|---|---|
| 命令类型标识 | 命令选择 | 路由请求标识符 | 源地址 | 响应地址 | 路由开销 |
| 报文载荷域 | | | | | |

命令类型标识字节标识该命令报文的类型，此处为 0x02，表示路由应答命令。

其余字段的含义与路由请求命令基本相同。

(3) 路由错误命令

当网络设备无法向前传送数据报文时，使用路由错误命令来通知该报文的源设备：报文在传送的过程中出现了错误。路由错误命令报文的报文载荷域结构如表 6-5 所示。

表 6-5 路由错误命令报文结构

| 1 字节 | 1 字节 | 2 字节 |
|---|---|---|
| 命令类型标识 | 错误代码 | 目的地址 |
| 报文载荷域 | | |

命令类型标识字节为 0x03，为路由错误命令。

错误代码用来表示出错的类型，协议规定 0x00 为无有效路由；0x01 为树状链路失败；0x02 为非树状链路失败；0x03 为低电池电压；0x04 为无路由能力；其余值保留。

目的地址域为路由错误命令报文要传送的目的设备的地址信息。

## 6.2 网络新建

新建网络的功能只能由具有 ZigBee 协调者能力的设备来实现，将其自身初始化为一个新的 ZigBee 网络协调者。

在接收到上层新建一个网络的调用后，该设备首先在所指定的信道上扫描。如果上层已经确定了网络标识 PAN ID，那么网络层将确保所给定的 PAN ID 不会与所

选择信道的现有网络 PAN ID 参数产生冲突。若发现存在冲突，那么，如果有可能则从给定的信道中选择另外的一个信道，在这个信道中，所给定的 PAN ID 不与信道中的其他网络冲突；如果选择不到合适的信道，则网络层发送错误标志。如果上层未指定 PAN ID，那么网络层在所选定的信道中选择与任何已存在网络不会冲突的 PAN ID 作为新建网络的标识。

一旦合适的信道和个域网标识 PAN ID 确定后，网络层将选择 0x0000 作为 16 位的网络地址，并通知 MAC 层，设置为网络地址。这样在指定信道上的网络就新建好了，此时该网络中只有协调者自身一个节点。

## 6.3 设备加入网络

设备加入网络功能就是通过与已加入网络的协调者或路由器设备建立连接来实现的。当设备与某一网络协调者或路由器连接后，将形成父子关系，前者为子设备，后者为父设备。

设备与网络连接的方式有两种：一种是子设备主动与指定的 PAN 进行连接；另一种是子设备预先指定的父设备主动先将子设备加入到 PAN 中，而后子设备再通过孤立点方式加入到 PAN 中。在本设计中采用第一种方案，子设备主动加入网络。

### 6.3.1 子设备主动加入网络

当子设备接收到加入网络命令后，如果子设备已经同网络连接，则返回出错标志，否则尝试连接其 POS 范围内的网络协调者或路由器。具体加入 PAN 的流程如图 6-1 所示。

首先，子设备要获取其 POS 范围内具有允许连接能力的网络协调者或路由器的地址信息。因而首先发送信标请求命令，其 POS 范围内具有允许连接能力的网络协调者或路由器接收到该命令后都会发送各自的信标帧；在这里采取时间最短的策略，子设备将加入最先接收到的信标所对应的 PAN 中。但若一段时间未接收到信标，则退出。

在获取了要加入的父设备信息后，子设备向父设备发送连接请求命令。父设备接收到连接命令后，检查当前资源是否能够再接收设备加入 PAN 中。若资源满足后，父设备将存储子设备地址，并为子设备分配 16 位的网络地址，同时生成连接响应命令帧，向连接请求子设备发送有未处理数据的 Ack 应答帧；若资源不满足，

则直接发送无未处理数据的 Ack 应答帧。

子设备在一段时间内等待接收来自父设备的 Ack 应答帧，接收到后判断父设备是否有本设备的未处理数据，若无或在指定时间内未接收到 Ack 应答帧则退出。

当子设备判断父设备有未处理数据，则向父设备发送数据请求命令；父设备接收到该命令后，发送缓存的连接响应命令帧；子设备接收到后，更新其设备网络地址、PAN ID、父设备地址信息等参数。此时子设备就完成了加入 PAN 的整个过程。

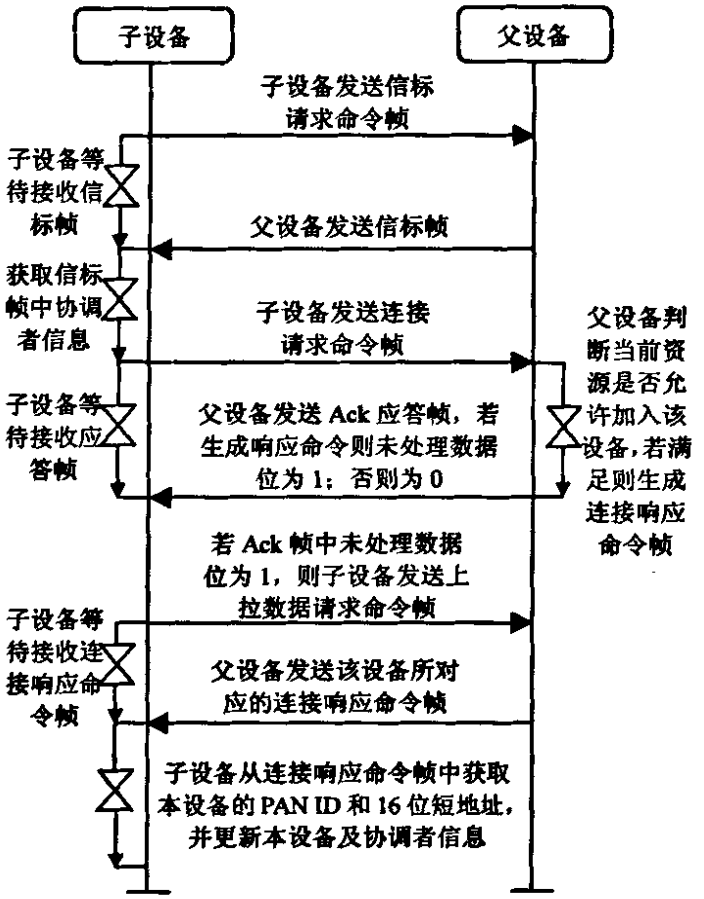


图 6-1 子设备连接的基本流程

## 6.3.2 父设备主动将子设备加入网络

ZigBee 网络层允许网络协调者或路由器利用直接请求的方法，将另一个设备加入到自身的网络中来。

当作为网络协调者或路由器的父设备接收到上层的这一功能命令后，父设备试图将指定的子设备加入到自己的邻居表中。若父设备中已经存在该子设备的地址信息，则不需要做其他操作；否则需要在邻居表中创建新的入口，并存放子设备的地址信息。这样就可以表示父设备已经将子设备加入到网络中了。注意在这一过程中，父设备与子设备间并没有数据交互。

但此时子设备本身并不知道已经加入到网络中，子设备将通过孤立点命令请求来与父设备进行交互以完成加入网络的全部过程。

## 6.3.3 子设备网络地址分配机制

在 ZigBee 网络层中，采用的是分布式地址分配机制[32][48]，即为每一个父设备分配一个有限的网络地址段。因而网络协调者决定了整个网络的最大子设备数。

每一个设备都有一个连接深度，即表示该设备到网络协调者的最小跳数，其中协调者自身的连接深度为 0，其直接子设备为 1。协调者也决定了网络的最大连接深度，用 nwkMaxDepth(Lm)变量表示，父设备拥有最大子设备数 nwkMaxChildren(Cm)，最大路由器设备数 nwkMaxRouters(Rm)。通过这几个网络参数即可计算父设备所能分配子区段地址的数量(Cskip)：

$$Cskip(d)=\begin{cases}1+Cm*(Lm-d-1) & \text{当Rm=1时}\\ (1+Cm-Cm*Rm^{Lm-d-1})/(1-Rm) & \text{当Rm>1时}\end{cases}$$

其中 d 为当前父设备在网络中的深度。若一个设备的 Cskip(d)小于或等于 0，则表示该设备没有接收子设备连接的能力。

利用 Cskip(d)作为偏移，向子设备分配有效的网络地址：

$$\begin{cases}A_1=Aparent+1\\ A_n=A_{n-1}+Cskip(d-1)\end{cases}$$

Aparent 为父设备的网络地址，$A_1$ 为第一个子设备分配的地址，即在父设备地址的基础上加 1；其后的子设备网络地址均在前一个设备的基础上向后偏移 Cskip(d—1)，这样就可以实现网络地址的有效分配了。

## 6.4 设备退出网络

对于已连接网络的设备主要有两种从网络中断开连接的方式：子设备自身主动要求断开连接；父设备要求某一子设备从网络中断开连接。

子设备自身主动要求断开连接。子设备首先检查自身是否已经加入网络，并且父设备是否与要断开连接的对象相同；接着子设备组织断开连接请求命令帧，并发送给父设备；注意按照 ZigBee 协议规定，子设备在发送了断开连接请求命令后，无论父设备是否做出断开连接响应，子设备均将其父设备信息清空，表示子设备已经从网络中退出；当父设备成功接收到断开连接请求命令时，在其邻居表中检查是否存在该子设备，若存在则将该子设备从邻居表中移除。

父设备要求某一子设备从网络中断开连接。父设备首先检查要断开连接的对象是否在其邻居表中，若在则生成断开连接请求命令帧，并发送给指定子设备；与子设备主动要求断开连接一样，无论父设备是否收到子设备的应答，都将该子设备从邻居表中移除；当子设备成功接收到断开连接命令后，将父设备信息清空。

## 6.5 网络路由

路由功能就是将网络报文从源穿过网络传递给目的的行为。无线传感器网络的路由设计是比较复杂的，并且不同的网络适用的网络路由方法也各不相同，这里主要讲述无线传感器网络的路由特点和常用的路由协议，论文中网络路由的具体实现将在后面的应用实例中详细阐述。

### 6.5.1 无线传感器网络的路由特点

与传统网络的路由协议相比，无线传感器网络的路由协议具有以下特点[49]：

(1) 能量优先

传统路由协议在选择最优路径时，不考虑节点的能量消耗问题。而无线传感器网络的节点能量有限，延长整个网络的生存期成为无线传感网络路由协议的重要目标。

(2) 基于局部拓扑信息

无线传感器网络为了节省通信能量，通常采用多跳的通信模式，而有限的存储资源和计算资源，使得节点不能存储大量的路由信息，不能进行太复杂的路由计算。

(3) 以数据为中心

传统的路由协议通常以地址作为节点的标识和路由的依据，而无线传感器网络中大量节点随机部署，所关注的是监测区域的数据，不依赖于地址标识。传感器网络包含多个传感器节点到少数协调器节点的数据流，按照对感知数据的需求、数据通信模式和流向等，以数据为中心形成消息的转发路径。

(4) 应用相关

传感器网络的应用环境千差万别，数据通信模式也不尽相同，没有一个路由机制适合所有的应用，这是传感器网络应用相关性的一个体现。设计者需要针对每一个具体应用的需求，设计与之相适应的特定的路由机制。

### 6.5.2 无线传感器网络的常用路由协议

目前研究人员已经提出了多种路由算法，这些路由协议可以大致分为四类：洪泛式路由协议、层次式路由协议、以数据为中心的路由协议以及基于位置信息的路由协议[50][51][52]。

洪泛式路由协议中具有代表性的是 Flooding 与 Gossiping。它们是传感器网络

中应用最早、最简单的路由协议，不需要维护网络拓扑结构。Flooding 是一种古老的传统洪泛式路由技术，接收到消息的节点向它的所有邻居节点广播接收到的数据，如此反复，直到数据达到目的节点或者达到数据报的最大跳数而被终止。

层次式路由协议将所有的节点分为若干簇，每个簇选举一个首领，由首领实现数据融合，达到节约功耗的目的。簇的划分依据是节点现有的电量和它与首领的距离。

在以数据为中心的路由协议中常用的有 SPIN 和定向扩散两种。SPIN 通过协商机制来解决泛洪算法中的“内爆”和“重叠”问题。传感器节点仅广播采集数据的描述信息，当有相应的请求时，才有目的地发送数据信息；定向扩散模型是 Estrin 等人专门为传感器网络设计的，与已有的路由算法有着截然不同的实现机制。

基于位置信息的路由协议是充分考虑了能源有效性的基于位置的路由协议，它比其他的基于位置的路由协议能更好的应用于无线传感器网络之中。

## 6.6 网络报文的发送与接收

(1) 网络报文的发送

网络报文的发送相对比较简单：首先，根据不同的报文类型封装相应的网络报文，其中报文的源地址为本设备网络地址，目的地址为最终的目的设备网络地址；再调用 MAC 层数据帧的封装函数组织 MAC 数据帧，其中数据帧的源地址为本设备地址，而目的地址是通过相应的路由算法计算出来的下一跳节点的地址信息；帧组织好后，将该数据帧发送给下一跳节点，由下一跳节点来负责接收、转发。

(2) 网络报文的接收

MAC 层成功接收到数据帧后，将去除帧头、帧尾域，剩下数据载荷域作为网络报文传递给网络层。网络层接收报文的处理流程如图 6-2 所示。

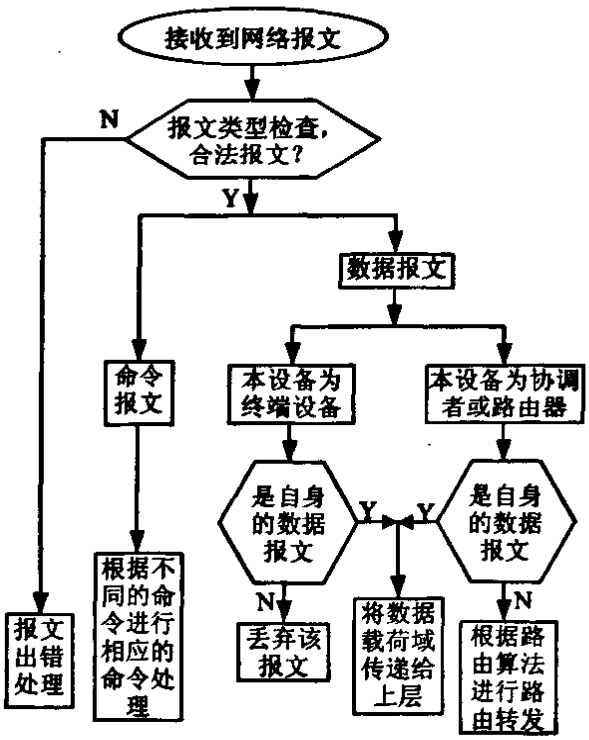


图 6-2 接收报文处理流程

当网络层成功接收到网络报文后，首先根据报文头域检查是否为合法的网络报

文，主要检查报文类型是否合法：若不是数据报文或 3 种类型的命令报文，则对该报文进行报文出错处理。

当为合法的命令报文时，取得具体的命令类型，并根据不同的命令进行相应的命令处理、应答。

当为合法的数据报文时，由于网络终端设备没有路由转发功能，因此设备类型不同，处理也不尽相同。对于自身是网络终端设备时，根据报文的目的地址域检查该数据报文是否属于自己的，若是，则将报文载荷域传递给上层，供上层处理；否则，丢弃该数据报文。对于具有路由转发能力的网络协调者或路由器设备，若不是自己的数据报文，根据报文的目的地址和路由算法计算下一跳节点的地址信息，再调用 MAC 层程序接口将该数据报文转发给下一跳节点；若是自己的数据报文，同样将报文载荷域传递给上层协议，由上层来进一步处理。

表 6-6 网络层函数列表

| 函数类型 | 函数编号 | 函数头 | 函数功能简述 |
|---|---|---|---|
| 内部功能函数 | 1 | UINT8 NLDECmdPKTProcess(void) | 网络层命令报文处理函数 |
| | 2 | UINT8 NLDEDataPKTProcess(void) | 网络层数据报文处理函数 |
| 外部功能函数 | 1 | void NWKInit(void) | 网络层初始化工作 |
| | 2 | Bool NLMEEnableRx(void) | 网络层接收允许 |
| | 3 | Bool NLMEDisEnableRTX(void) | 网络层接收禁止 |
| | 4 | UINT8 NLMESendPKTReq(UINT8 DestAddrMode, UINT8 SourceAddrMode, UINT16 DestNWKAddr, UINT8 PKTLen,UINT8 *nwkPKT) | 网络层子设备请求连接协调者 |
| | 5 | UINT8 NLMEMakeDataPKT(Bool IsSecurity, UINT16 nwkDestAddr, UINT8 nwkPayloadLen, UINT8 *nwkPayload,UINT8 **nwkPKT) | 生成数据报文 |
| | 6 | UINT8 NLMEMakeRouteReqPKT(Bool IsSecurity, UINT8 **nwkPKT) | 生成路由请求命令报文 |
| | 7 | UINT8 NLMEMakeRouteAckPKT(Bool IsSecurity, UINT16 nwkDestAddr,UINT8 **nwkPKT) | 生成路由应答命令报文 |
| | 8 | Bool CheckInNeighborTable(Node_info Dest,Bool DelFlag, UINT8 *index) | 在邻居表中查找目的设备 |
| | 9 | UINT8 NLMEStartRoute(UINT16 *nwkAssociateAddr) | 网络层开始发起路由请求 |
| | 10 | UINT8 NLMEPKTProcess(void) | 网络报文的处理 |
| | 11 | UINT8 NLMEJoinPAN(void) | 网络设备加入PAN |
| | 12 | UINT8 NLMEExitPAN(UINT16 nwkDestAddr) | 请求退出网络 |
| | 13 | UINT8 NLMESendData(UINT16 nwkDestAddr, UINT8 dataLen,UINT8 *data) | 网络层发送数据报文 |

## 6.7 网络层函数列表

网络层功能设计实现的函数都包含在 Zig_NWK.c 文件中。与 MAC 层相同，在本层的程序设计上同样将内部函数和外部函数分开设计。网络层主要功能函数列表如表 6-6 所示

其中内部功能函数比较简单，主要负责网络命令报文和数据报文的处理；外部功能函数主要包括了各种网络报文的生成、网络的加入与退出、网络报文的发送等功能函数。

## 6.8 网络层主要裁减内容

本文所设计的网络为簇状网络拓扑结构，这一点在下一章中会有所体现；网络报文的种类较少，对数据报文、路由请求命令报文和路由应答命令报文都是支持的；在设备加入网络时，父设备采用的是分段的地址分配机制；网络新建功能是有协调者来负责实现的；对于设备加入网络，论文中只支持子设备主动加入网络，没有实现父设备主动将子设备加入网络的方式；设备退出网络的两种方式都是支持的；对于网络路由部分，有针对性的设计了一跳路由功能。具体内容如表 6-7 所示。

表 6-7 网络层主要裁减内容

| 网络层主要实现功能 | 具有功能 | 本文实现 | 网络层主要实现功能 | 具有功能 | 本文实现 |
|---|---|---|---|---|---|
| 网络拓扑 | 星型网络 | × | 网络新建 | 协调者建立网络 | √ |
| | 网状网络 | × | 设备加入网络 | 子设备主动加入网络 | √ |
| | 簇状网络 | √ | | 父设备主动将子设备加入网络 | × |
| 网络报文 | 数据报文 | √ | 设备退出网络 | 子设备主动退出网络 | √ |
| | 路由请求命令报文 | √ | | 父设备要求子设备退出网络 | √ |
| | 路由应答命令报文 | √ | 网络路由 | 一跳路由 | √ |
| | 路由错误命令报文 | × | | Adhoc多跳的实现 | × |
| 网络地址分配 | 分段地址的分配 | √ | | 路由维护及修复 | × |

## 6.9 网络层测试

在网络层的测试中，对于各种网络报文的组织函数在测试方法上，与 MAC 层基本一致，通过另外一个 MAC 层设备接收 MAC 帧，并将帧的有效数据显示出来，再对照报文格式研究分析。而对于网络层功能的测试，如网络的新建、设备加入与退出网络、网络路由等功能将在下一章具体的应用实例中给出详细的测试。

## 6.10 本章小结

本章在介绍网络报文结构的基础之上，重点讲述了网络新建、设备加入网络、设备从网络中退出和网络报文的路由等网络层功能。

网络的新建功能是由协调者负责完成的，协调者上电初始化后，将自己的网络地址赋为 0，以作为网络的初始节点；设备加入网络主要有子设备主动加入网络和父设备将子设备加入网络两种方式，本文采用的是第一种方式；同样，设备退出网络也存在两种方式：子设备要求退出网络和父设备要求子设备退出网络，本设计中同时支持这两种方式；本章在网络路由的介绍中，仅给出了无线传感器网络中常用的网络路由协议及其特点，没有具体介绍本文的路由设计，这部分内容在下一章中具体讲述；最后，详细列出了网络层的主要功能函数和网络层所做的主要裁减工作。

实际上，网络层功能的核心内容仍然是依靠 MAC 层来实现的，即各项功能的主要实现函数仍然是调用 MAC 层接口来组织的。另外在设计实现机制上，网络报文的组织和报文的发送、接收与 MAC 帧的组织和帧的发送、接收也比较类似。

# 第七章 WSN在农业大棚中的模拟应用

前面章节重点介绍了ZigBee协议的物理层、MAC层和网络层主要功能的设计实现以及MT-ZigBee硬件平台的搭建。本章将在此硬件平台和所实现的底层协议栈之上构建一个具体的应用实例—WSN 在农业大棚中的应用，来验证硬件平台和协议栈的可行性。重点讲述实际应用中网络的组建、加入、退出和网络路由等功能。

农业大棚(温室)的测量控制是目前实施农业中一个备受关注的问题。对农业大棚的测控主要是大棚内温度、湿度和光照度等参数的检测与控制[53]。由于现场布线的不方便、现场节点的位置不固定，采用无线通信方式来传输现场采集的数据与控制信息，更适合现场使用。

由于大棚内监控的参数较多、且复杂，本章仅在实验室内模拟实现组网、部分参数的采集和控制。

## 7.1 系统功能

WSN 在农业大棚中应用的主要功能就是实现对大棚内相关现场参数的采集和调节控制。在农业大棚中使用的传感器品种较多，按其检测参数分类，主要有以下几种：

温度和湿度。作物的生长与温度和湿度有密切关系，塑料大棚的控制参数中，温度与湿度的检测、控制是主要参数之一；

土壤干燥度。作物生长需要水分，在实施农业中，灌溉如何做到既不影响作物生长又不浪费水资源是至关重要的。土壤干燥度的检测，需要用干燥度传感器。目前采用较广泛的干燥度传感器是由负压传感器与陶瓷过滤管组成的；

二氧化碳浓度。农作物生长发育离不开光合作用，而光合作用又与二氧化碳有关，所以控制二氧化碳的浓度，有利于作物的生长发育；

光照度。实施农业中，采用栽培管理自动化系统，采用光传感器来检测和控制光照强度，使作物可以得到均匀一致的光照；

土壤养分。土壤养分依赖于施肥，合理施肥不仅可以提高作物产量，而且可以避免过量施肥而造成不必要的损失。土壤养分的测定包括土壤有机质、pH值、氮、磷、钾以及交换性钙和镁的检测。土壤养分测定广泛采用离子、生物传感器。

## 7.2 系统构成与网络拓扑

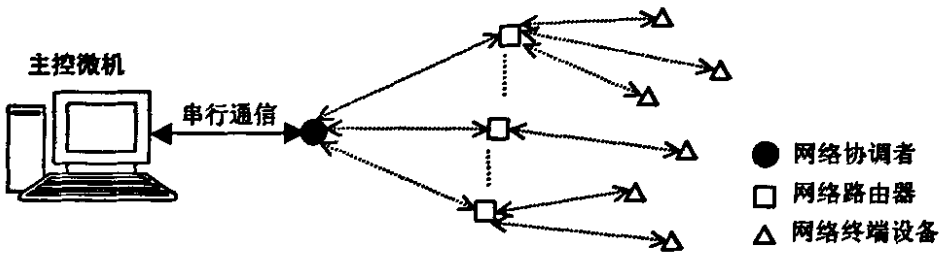


图 7-1 系统构成框图

系统构成和特定的网络拓扑结构如图 7-1 所示。根据农业大棚中数据传输距离较短的特点，整个网络拓扑采用一个 3 级的树型结构。在该应用中 ZigBee 节点分为标准的 3 类设备：网络协调者、网络路由器和网络终端设备。其中所有的网络节点在硬件上是完全一样的，并安装了用于检测环境温度等参数的传感器。

网络协调者负责整个网络的构建、传输、控制和数据的汇集，并且可以通过串行通信的方式与 PC 进行通信，PC 通过协议的接口命令可以与整个网络中每一个节点进行通信，进行环境数据的采集和控制。

网络路由器节点的目的主要是为了增加网络通信距离。ZigBee 物理信道的通信距离一般在 50 米左右，考虑农业大棚的实际情况，网络只设置了一级路由器，这样整个网络总共有 3 层，通信距离可以覆盖以协调者为中心方圆 200 米，这样足够满足农业大棚的实际需求。

网络终端节点在软件上最简单，没有路由的功能，只负责接收数据和发送数据。

## 7.3 网络新建

网络新建的功能比较简单，与网络层协议栈所实现的内容没有什么区别。网络协调者在选择好相应的 PAN ID 后，将 0 作为自身的网络地址，并初始化新网络，此时新建的网络中只有协调者一个节点。

## 7.4 设备加入网络和退出网络

设备加入网络、退出网络的基本流程与网络层协议栈所实现的内容是一致的。对于网络地址是根据上述的网络结构来进行分配的，采用地址分块的分配方法。

当网络的根节点建立以后，第 2 层的网络路由器和第 3 层的终端节点便可加入

网络中，但他们加入是有顺序的，协调者不接收终端设备的连接请求。网络的加入操作是由按键的硬件中断来触发的。

当路由器按下加入网络的按键后，路由器广播连接请求命令，在其POS范围内的协调者接收到该命令后检查当前资源是否满足，若满足则允许该路由器的网络连接，回应以连接响应命令，并将该路由器的节点信息加入到邻居表中；路由器接收到连接响应命令后将协调者信息记录为其父节点，此时连接成功。对于路由器网络地址的分配是由协调者来决定的：对于第2层路由器的网络地址，以0x1000为增量，为每一个加入的路由器分配分块的网络地址，即网络地址分别为0x1000、0x2000等。但需要注意的是路由器可以从网络中退出，例如当前路由器地址已经分配到了0x8000，在下一路由器加入网络前，有其他的路由器退出了网络，如其地址为0x3000。则当下一路由器加入网络时，协调者选择一个最小的、可用的网络地址来进行分配，即0x3000，而不是0x9000。

当终端设备按下加入网络按键后，终端设备广播连接请求命令，在其POS范围内的路由器将做出连接响应应答。路由器加入网络时，协调者只有一个，而终端设备加入网络时，在终端设备POS范围内的路由器很可能不只一个，即所有接收到该连接请求命令的路由器都将做出连接响应，这里采用时间最优的方法：即终端设备最先接收到哪个路由器的连接响应，就与该路由器进行网络连接。对于终端设备网络地址的分配与路由器的分配方案基本相同，在路由器网络地址的基础上，以增量1来进行分配的。

网络的退出也是由与网络加入相同的按键来触发的。当已加入网络的设备再次按下该按键后执行退出网络操作。其执行的流程在网络层协议实现中已详细讲述了。

## 7.5 报文的网络路由

在本应用所构建的网络拓扑中，网络中每个节点都可以与任何其他节点进行通信。

对于协调者要发送报文给某一设备，根据目的设备的地址计算其所在的子树，接着调用MAC层数据发送接口将报文发送给该子树的路由器节点，由路由器实现转发。

终端设备要发送报文则很简单，不管目的设备地址如何，调用MAC层接口将报文发送给其父亲路由器，由路由器来路由转发。

路由器具有路由转发功能，稍微复杂一些：首先查看目的网络地址是否在其邻居表中，若在则直接发送给该子设备；否则将该报文发送给其父设备—网络协调者，交由协调者来实现路由。

如图 7-2 所示的特定网络结构，下面以节点 3 要给节点 5 发送网络报文为例介绍在网络中是如何实现通信的。

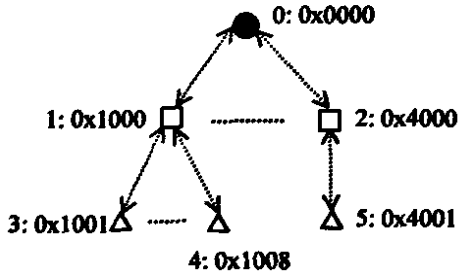


图 7-2 网络特例

节点 3 要发送报文给节点 5，则目的网络地址为 0x4001，源网络地址为 0x1001。首先节点 3 将该报文封装成发送给其父节点—路由器 1(0x1000)的 MAC 数据帧，并发送；路由器 1 接收到该报文后，检查到目的地址为 0x4001，不在其邻居表中，即不在该子树中，将该网络报文按目的设备为节点 0(0x0000)的地址封装成 MAC 数据帧，并发送出去；协调者 0 接收到该报文后，根据网络地址 0x4001 计算出所在子树路由器的网络地址为 0x4000，再次将该报文按目的地址为 0x4000 封装成 MAC 数据帧，发送给节点 2；节点 2 接收到后，发现报文目的地址在其邻居表中，则直接将该报文发送给节点 5。这样便完成了整个报文的路由传输过程。

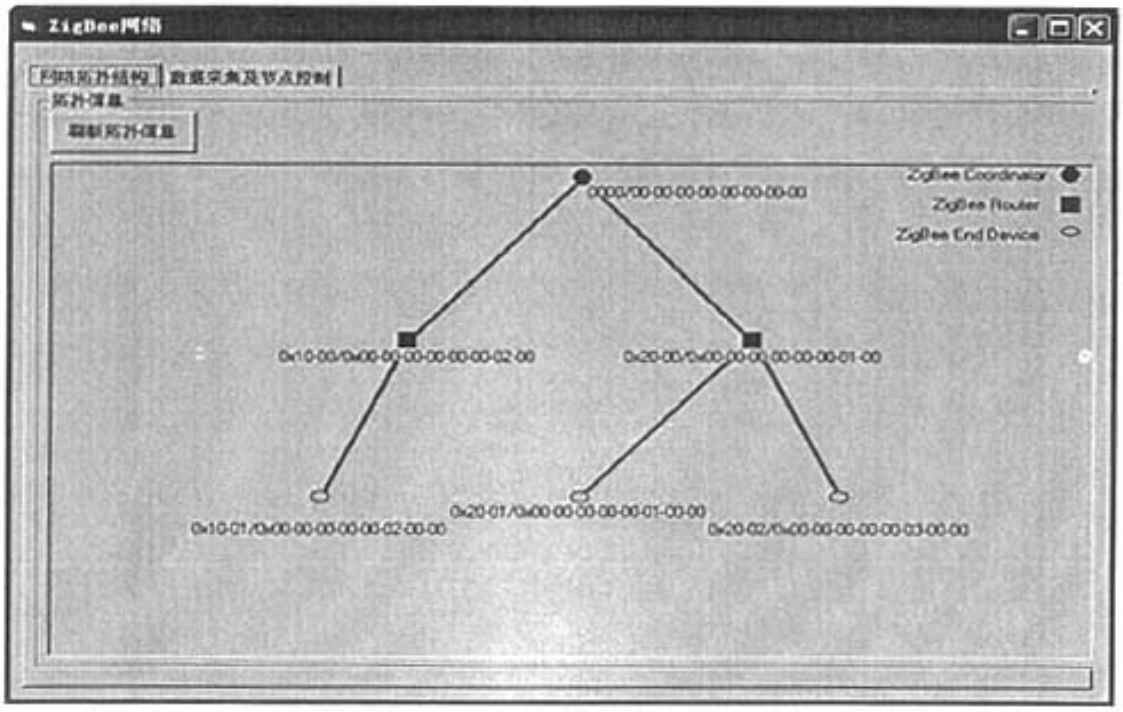


图 7-3 网络拓扑结构

## 7.6 模拟运行情况

本文的验证实例是在实验室内模拟实现的。下面就实验室内6个节点的测试情况给出简单的介绍。

由于节点数量有限，在实验室内模拟测试时只验证了6个节点的情况。这6个节点分别是1个网络协调者、2个网络路由器和3个网络终端设备。首先协调者新建网络，接着路由器依次加入网络中，最后网络终端设备再与其最近的路由器进行连接，这样便完成了整个网络的组建。其中网络协调者与PC机通过串口连接，PC端程序可以通过协调者了解到整个ZigBee网络中每个节点的情况，并实现参数的采集与控制。

为便于直观的显示出网络结构，在PC端程序中绘出了整个网络的拓扑结构，如图7-3所示。最上层为网络协调者；中间层的2个设备为网络路由器；下层的3个设备表示网络终端设备，其中第1个终端设备是与第1个网络路由器相连接的，而后2个终端设备是与第2个路由器相连接。每个设备附近显示的是该设备的网络地址和MAC地址，图7-3中第2个路由器的地址为“0x20-00/0x00-00-00-00-00-00-01-00”，其中“0x20-00”表示其网络地址，后面8个字节表示其固有的MAC地址。按照上面网络地址的分配，这6个节点的网络地址依次为0x0000、0x1000、0x2000、0x1001、0x2001、0x2002。

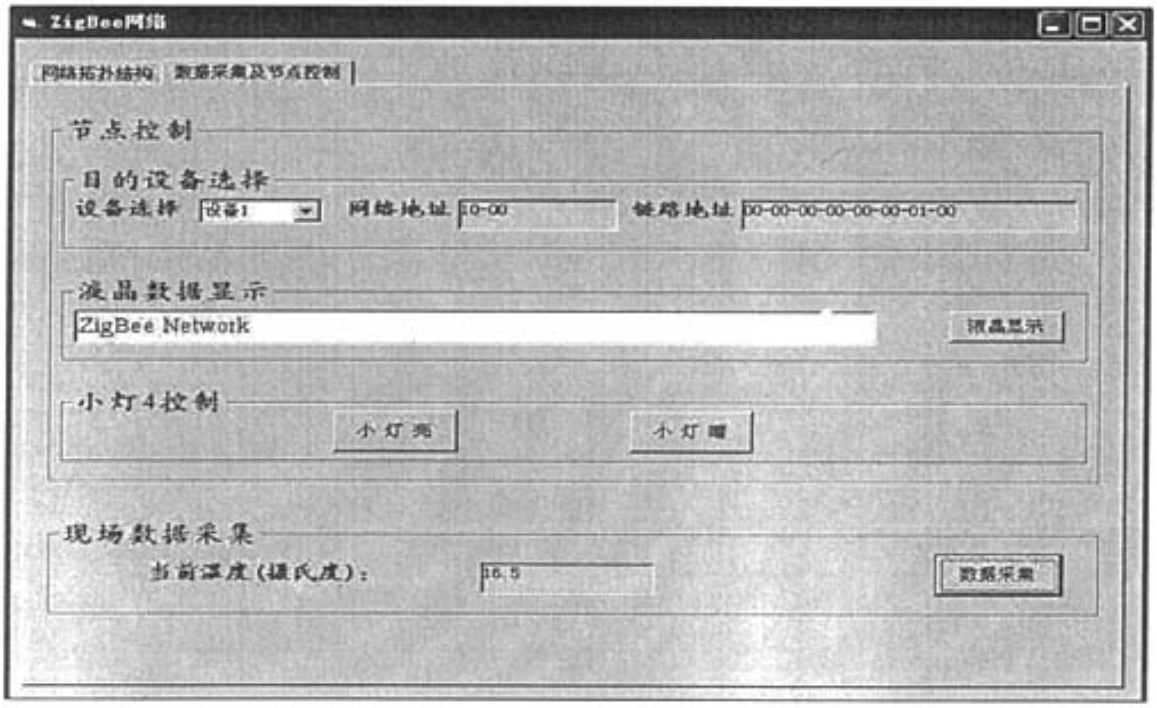


图7-4 参数采集与节点控制

参数的采集与节点的控制如图 7-4 所示。通过目的设备的选择来确定要采集和控制的对象，其中目的设备为整个网络中的任何一个节点。在节点的硬件设计中预留了液晶显示模块和运行指示灯，因此在节点的控制中就采用这两个模块来测试对节点的控制。点击“液晶显示”按钮将在选择节点的液晶屏上显示文本框中的内容，其中要显示的内容是通过网络协调者发送到目的节点的，若是目的节点是网络终端节点则还需要网络路由器接收转发到目的设备上。小灯的亮暗控制在 ZigBee 网络中的发送流程与液晶显示是基本一致的。

对于现场数据的采集，由于时间和条件的限制，仅模拟采集了现场温度一个参数。其中用于检测温度的传感器采用负温度系数热敏电阻 MF58，其特点是耐热、体积小、稳定、可靠性高。

## 7.7 本章小结

本章在已搭建的硬件平台和底层协议基础之上，针对农业大棚的实际需要设计了 3 层的树状网络结构，并给出了网络组建、网络地址分配及报文路由的具体过程。最后在实验室现有的条件下模拟实现 6 个节点的网络组建、节点参数采集和控制。

本章的测试实例就是在底层协议基础上构建简单的应用层命令接口。由于节点数目、传感器类型有限，因此在测试中仅提供了 6 个节点，且采集的数据类型也只有温度一个参数。虽然本模拟实验中网络通信没有发现问题，但由于节点较少，网络规模小，网络数据量不大，这样实验室内的模拟测试是不够充分的。在实际现场应用中还需要对大数据量交互的网络进行充分的测试。

# 第八章 总结与展望

## 8.1 总结

ZigBee技术和无线传感器网络的研究，是当今新兴前沿的研究热点，目前在国内刚刚起步，其应用前景十分广泛。

本文在深入研究IEEE 802.15.4和ZigBee协议的基础上，通过自身设计的硬件开发平台，有所简化的实现了物理层、MAC层和网络层协议栈，并在论文的最后给出了一个简单的验证实例，从而验证硬件平台和协议栈的可行性。

本文的主要工作总结如下：

(1) 收集IEEE 802.15.4、ZigBee协议规范及相关资料，仔细阅读、理解，并深入研究其内容。了解ZigBee技术的基础知识和网络通信体系结构中的分层原理，同时充分掌握各层的主要功能及实现的技术要领。

(2) 选用Freescale公司提供的ZigBee开发方案：MC9S08GB60和MC13192。向Freescale公司申请ZigBee物理层芯片－MC13192，设计实现MT-ZigBee硬件平台，并交由厂家生产后焊接调试通过。

(3) 在此硬件平台之上，根据协议规范详细设计实现了物理层、MAC层和网络层协议栈，当然这也是本文的工作重心。

物理层主要负责物理信道上数据包的发送与接受、信道检测等关于物理信道的功能，其功能的实现主要是在底层驱动程序的基础上近一步设计完成的。

MAC层主要负责各种类型帧的组织、信道的冲突避免访问、设备间的同步和MAC帧的发送、接受等功能。MAC层的程序设计可以说是实现整个协议栈的主要内容，包括网络层大部分功能的核心内容实际上也是由MAC层来实现的。

网络层完成网络报文的组织封装，实现网络的拓扑，网络报文的路由，网络的新建，设备加入和退出网络等功能。

(4) 以农业大棚为应用对象，组建3层树型网络结构，并在实验室内模拟测试硬件平台和协议栈的可行性。

通过本文的设计，熟悉了ZigBee的关键技术要领、无线通信的分层原理及整体架构。通过对MT-ZigBee硬件平台的设计，掌握了一定的高频电路设计的相关知识，同时也积累了一些硬件设计、焊接和调试等方面的经验。在ZigBee协议栈的设计中，掌握了软件设中层次划分和合理封装函数的基本要领，深刻体会到在嵌入

式软件设计中，对底层硬件驱动程序的封装要与上层的功能函数分割开，这样将更加便于函数的调用与问题的检查。

## 8.2 展望

由于时间和现有条件的限制，本文在设计实现过程中存在不少不尽完善的地方，下面给出几点在后续设计中需要进一步改进和完善的工作：

在硬件平台方案的选择上，Freescale 公司目前已提供了 HCS08 核与 2.4GHz 射频前端相结合的一体化芯片 MC1321x，其中提供了 16KB、32KB 和 60KB 不同大小 Flash 的选择。在进一步的设计中，可以直接采用这种方案，这样可简化硬件设计的工作量，同时也降低了开发成本。

在便携式产品中，功耗是一个非常重要的技术指标。由于本文的工作重点是协议栈的实现，在功耗方面没有太多的关注，仅仅在硬件设计上为实现低功耗提供了电路基础，但功耗的降低必须是软件与硬件的结合才能实现的，因而在后续工作中低功耗的实现将是一个工作重点。

由于时间的有限，对协议规范的理解有所局限，同时所实现的底层协议栈也是从便于设计的角度出发，例如为简化设计内容本文不支持超帧结构；在设备的同步机制方面，IEEE 802.15.4 提供了信标同步和上拉数据两种方式，为简化程序的设计，本文选择上拉数据的方式。同时在对某些数据结构的设计和协议功能分层的设计方面也存在不合理的地方，因此对协议栈的改进工作还有较大的余地。

最后需要改进的地方就是整体的测试工作。由于节点数目有限，本文最后的验证实例对协议的测试工作不是很充分。在产品应用中，必须做到非常详细充分的测试工作，以检查出系统中隐蔽的问题。因而详尽的测试实验是后续工作中必不可少的环节。

ZigBee 技术以其低成本、低功耗等优势在无线传感器网络中有着十分广阔的应用前景，可以相信，随着研究的不断深入，以 ZigBee 为通信技术的无线传感器网络将会广泛融入我们的生活。

# 参考文献

[1] 崔莉，鞠海玲，苗勇等.无线传感器网络研究进展[J].计算机研究与发展，2005，42(01):163-174.

[2] DAVID E C,WEI Hong.Wireless sensor networks[J].communications of ACM,2004,47(06):30-33.

[3] Ren FY,Huang HN,Lin C.Wireless sensor networks[J].Journal of Software,2003,14(07):1282-1291.

[4] Zhao F,Guibas LJ.Wireless Sensor Networks:An Information Processing Approach[M].Morgan Kaufmann,2004.

[5] 于海斌,曾鹏等编著.智能无线传感器网络系统[M].北京:科学出版社,2006.

[6] 曹红萍，蒋云良，缪强.室内无线传感器网络及其应用[J].计算机应用研究,2006,09:209-212.

[7] 孙利民,李建中,陈渝等编著.无线传感器网络[M].北京:清华大学出版社,2005.

[8] 莫宵雁.无线传感器网络分簇式路由协议的研究与设计[D].浙江大学,2006.

[9] Warneke B,Last M,Liebowitz B,Pister K S J.Smart dust:Communicating with a cubicmillimeter computer[J].IEEE Computer Magazine,2001,34(01):44-51.

[10] Akyldiz IF,Su W,Sankarasubramaniam Y,Cayirci E.Wireless sensor networks:A survey[J].Computer Networks,Mar.,2002,38(04):393-422.

[11] 孙亭，杨永田，李立宏.无线传感器网络技术发展现状[J].电子技术应用,2005,06: 1-5.

[12] Bonner P,Gehrke J,Seshadri P.Querying the physical worked[J].IEEE Personal Communiciation,2000,07(05):10-15.

[13] 李建中，李金宝，石胜飞.传感器网络及其数据管理的概念、问题与进展[J].软件学报,2003,14(10):1717-1727.

[14] Polastre J,Szewcayk R,Mainwarint A.Analysis of wireless sensor networks for habitat monitoring[J].Wireless Sensor Networks，2004,399-423.

[15] 国家中长期科学和技术发展规划纲要(2006-2020)[Z]，http://www.nsfc.gov.cn/nsfc/cen/ghgy/,2006.02.16.

[16] 徐琳,刘志勇,刘克.国家自然科学基金委员会信息科学部计算机科学处 2005年度基金申请与资助概况[J].软件学报,2005,16(11):2021-2028.

[17] 信息科学部.2006 年度国家自然科学基金项目指南：重点项目[J].自然科学进展,2006,01:71-71.

[18] Hamid Gharavi,Srikanta P.Kumar.Special issues on sensor networks and

applications[J]. Proceeding of the IEEE, 2003, 91(08): 1151-1153.

[19] 彭木根, 姜涌, 王文博. 无线数字家庭网络泛在接入技术[J]. 中兴通讯技术, 2006, 12(04): 41-46.

[20] 王锐华, 益晓新, 于全. ZigBee 与 Bluetooth 的比较及共存分析[J]. 测控技术, 2005, 24(06): 50-56.

[21] Gislason Drew, Gillman Tim. ZigBee wireless sensor networks—ZigBee is an emerging wireless protocol designed for low-cost, high-reliability sensor networks[J]. Software Tools for the Professional Programmer, 2004, 29: 40-42.

[22] Fukui K, Fukunaga S, Tanimoto K. ZigBee technology for low-cost and low-power radio communication systems[J]. Journal Institute of Electronics Information and Communication Engineers, 2005, 88(01): 40-45.

[23] Evans-Pughe, C. Bzzzz zzz(Zigbee wireless standard) [J]. IEEE Review, 2003, 49(03): 28-31.

[24] ZigBee Alliance. ZigBee alliance finalizes specification[EB/OL]. http://www.zigbee.org. 2004.

[25] 王东, 张金荣, 魏延等. 利用 ZigBee 技术构建无线传感器网络[J]. 重庆大学学报(自然科学版), 2006, 29(08): 95-98.

[26] 蒋建辉. ZigBee 网络的设计与实现[D]. 苏州大学, 2006.

[27] ZigBee 联盟. ZigBee 技术引领无线数字新生活[J]. 电脑知识与技术, 2006, 27: 29-34.

[28] Zigbee Alliance. Network Specification, Version 2006-Zigbee Document 053474r13 , December 1th, 2006.

[29] EGAN D. The Emergence of ZigBee in Building Automation and Industrial Control[J]. Computing & Control Engineering Journal, 2005, 16(02): 14-19.

[30] 蒋挺, 赵成林编著. 紫蜂技术及其应用[M]. 北京: 北京邮电大学出版社, 2006.

[31] 贺文. 基于 IEEE 802.15.4/ZigBee 的无线传感器网络研究[D]. 浙江大学, 2006.

[32] IEEE 802.15.4, Part 15.4: Wireless Medium Access Control(MAC)and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs) , October, 2003.

[33] Zigbee Alliance. Network Specification, Version 1.00-Zigbee Document 02130r10 , December 14th, 2004.

[34] 凌志浩等. ZigBee 无线通信技术及其应用研究[J]. 华东理工大学学报(自然科学版), 2006, 32(07): 801-805.

[35] Mohammad Upal Mahfuz, Kazi M Ahmed. A review of micro-nanoscale wireless

sensor networks for environmental protectionProspects and challenges[J]. Science and Technology of Advanced Materials, 2005, 06: 302-306.

[36] 赵志峰, 郑少仁. Ad hoc 网络体系结构研究[J]. 电信科学, 2001, 01: 14-17.

[37] GB/T 13622-92, 中华人民共和国无线电频率划分规定(S).

[38] Chipcon AS SmartRF. CC2430 Preliminary Datasheet rev 1.01[EB/OL]. http://www.chipcon.com/, 2005..

[39] MC1321xRM Rev1.1.2006. Freescale Semiconductor, Inc[EB/OL] http://www.freescale.com/, 2006.

[40] Data Sheet MC9S08GB60 Rev. 2.3.2004. Freescale Semiconductor, Inc[EB/OL]. http://www.freescale.com/, 2004.

[41] 王宜怀, 刘晓升. 嵌入式应用技术基础教程[M]. 北京: 清华大学出版社, 2005.

[42] MC13192RM Rev1.3. Freescale Semiconductor, Inc[EB/OL]. http://www.freescale.com/, 2005.

[43] 耿欣, 沙斐, 李云志. 无线个域网的传感器应用设计[J]. 北京交通大学学报, 2005, 05(29): 64-67.

[44] 信息产业部文件178号, 微功率(短距离)无线电设备管理暂行规定[Z], 1998.

[45] Compact Integrated Antennas Rev1.2. Freescale Semiconductor, Inc[EB/OL]. http://www.freescale.com/, 2005.

[46] PCB Layout Guidelines for the MC1319x Rev0.1. Freescale Semiconductor, Inc [EB/OL]. http://www.freescale.com/, 2005.

[47] James, F. Kurose, Keith, W. Ross. Computer Networking: A Top-down Approach Featuring the Internet[M]. Pearson Education, 2001.

[48] 朱向庆, 王建明. ZigBee协议网络层的研究与实现[J]. 电子技术应用, 2006, 01: 129-132.

[49] 安琦. 无线传感器网络能量有效性路由协议研究[D]. 西北工业大学, 2006.

[50] Heinzelman WR, Kulik J, Balakrishnan H. Adaptive protocols for information dissemination in wireless sensor networks[J]. Proc MobiCom. Seattle: ACM Press, 1999. 174-185.

[51] 张晓薇. 无线传感网络数据合并算法的设计与实现[D]. 北京大学, 2005.

[52] 赵进. 无线传感器网络路由协议研究与实现[D]. 南京航空航天大学, 2005.

[53] 文汉云, 吴修德. 基于无线通信技术的传感系统在实施农业中的应用[J]. 自动化与仪器仪表, 2005, 01: 25-31.

# 附录 A ZigBee 协议设计主要数据结构

## 1 物理层主要数据结构

```
//1、接收、发送数据包结构体定义
typedef struct{
    UINT8 u8DataLength;     //数据包长度，对应字节数
    UINT8 *u8Data;          //数据包内容首地址
}PHYPacket;

//2、MC13192 对应 7 种运行状态
typedef enum{
    Off_Mode=0x00,          //Off 模式
    Hibernate_Mode,         //Hibernatef 模式
    Doze_Mode,              //Doze 模式
    Idle_Mode,              //Idle 模式
    RX_Mode,                //Receive 模式
    TX_Mode,                //Transmit 模式
    CCA_Mode,               //CCA/ED 模式
}PHY_Logic_Status;
```

## 2 MAC 层主要数据结构

```
//1、MAC_PIB，MAC 层属性结构定义
typedef struct{
    Bool   macBattLifeExt;        //电池寿命扩展标志
    UINT8  macAddMode;            //节点的地址模式(TRUE:可以使用短地址)
    UINT8  macAddress[8];         //节点自身 64 位扩展的 IEEE 地址
    UINT16 nwkAddress;            //节点自身由协调者分配的 16 位网络地址
    UINT16 macPANID;              //所在 PAN 的 16 位 PAN ID
    UINT16 nwkFatherAddress;      //父节点的 16 位网络地址
    UINT8  macFatherAddress[8];   //父节点 64 位扩展的 IEEE 地址
    UINT8  macDSN;                //发送数据或 MAC 命令帧的序列号
  #ifndef IsEndDev
  //协调者或路由器独有属性
    Bool   macAssociationPermit; //协调器或路由器当前是否允许连接标志
    UINT8 macBeaconOrder;        //信标帧间隔(0~15)
    UINT8 macBSN;                //发送信标帧的序列号
    UINT8 macSuperframeOrder;    //用以表示超帧活跃部分的长度(包括信标帧)
  #endif
}MAC_PIB;


//2、MAC 帧头中 2 字节帧控制字段的结构定义(2B)
```

```
typedef union{
    UINT16 W16;
    struct{
        //LSB, 低字节定义
        union{
            UINT8 B8;
            struct{
                UINT8 FrameType       :3;  //0~2, 3 位的帧类型
                UINT8 SecurityEnabled :1;  //3, 安全允许位
                UINT8 FramePending    :1;  //4, 帧未处理标记位
                UINT8 ACKRequest      :1;  //5, 确认请求位
                UINT8 IntraPAN        :1;  //6, 内部 PAN 标记位
                UINT8 Reserved        :1;  //7, 该位保留
            }bits;
        }LSB;
        //MSB, 高字节定义
        union{
            UINT8 B8;
            struct{
                UINT8 Reserved1       :2;  //8~9, 保留
                UINT8 DstAddrMode     :2;  //10~11, 目的地址模式
                UINT8 Reserved2       :2;  //12~13, 保留
                UINT8 SrcAddrMode     :2;  //14~15, 源地址模式
            }bits;
        }MSB;
    }MLSB;
}MAC_Frame_Control;

//3、Beacon Frame 中 2 字节 Superframe specification 的结构定义(2B)
typedef union
{
    UINT16 W16;
    struct{
        //LSB, 低字节定义
        union{
            UINT8 B8;
            struct{
                UINT8 BeaconOrder          :4;  //0~3, 信标序号
                UINT8 SuperframeOrder      :4;  //4~7, 超帧序号
            }bits;
        }LSB;
        //MSB, 高字节定义
        union{
            UINT8 B8;
            struct{
                UINT8 FinalCAPSlot         :4;  //8~11, 最终竞争时隙
                UINT8 BatteryLifeExtension :1;  //12, 电池寿命扩展位
```

```
                UINT8 Reserved              :1;  //13，保留
                UINT8 IsPANCoord            :1;  //14，PAN协调者标志位
                UINT8 IsAssocPermit         :1;  //15，连接允许位
            }bits;
        }MSB;
    }MLSB ;
}Superframe_Specification;

//4、MAC层接收到上层来的数据包结构体
typedef struct {
    UINT8 u8DataLen;        //数据包的长度，字节数
    UINT8 *Data;            //数据包存放首地址
}MACPayloadStruct;

//5、MAC帧结构
typedef struct{
  Bool   FrameFlag;             //TRUE：帧有效；FALSE：帧无效
  MAC_Frame_Control FremeCon;   //2字节的帧控制域
  UINT8  FrameSN;               //1字节的帧序列号
  UINT8  Len;   //帧其他字节数(只包括地址域和帧载荷2部分),帧总长度=Len+5
  UINT8 AddrANDLoad[aMaxPHYPacketSize-5];//地址域和帧载荷2部分的有效数据
}MACFrameStruct;
```

## 3 网络层主要数据结构

```
//1、邻居表项结构定义
typedef struct{
    UINT16 nwkAddr;          //路由器或子设备对应网络地址
    UINT8  macAddr[8];       //路由器或子设备对应MAC地址
}nwkNeigborEntryStruct;

//2、网络层属性定义
typedef struct{
    UINT8 DevType;           //设备类型
    #ifndef IsEndDev
    //邻居表部分的维护
        UINT8  nwkChildrenNum; //本设备所拥有的子设备数
        //协调者或路由器中的邻居表
        nwkNeigborEntryStruct nwkNeigborTalbe[nwkcMaxChildNum];
    #endif
}NWK_PIB;

//3、网络报文控制域结构定义
typedef union{
    UINT16 W16;
    Struct{
        //LSB，低字节定义
```

```
        union{
            UINT8 B8;
            struct{
                UINT8 PKTType          :2;  //0~1，数据报文类型
                UINT8 ProtocolVersion  :3;  //2~5，协议版本
                UINT8 DiscoverRoute    :1;  //6，发现路由位
                UINT8 Reserved1        :1;  //7，保留
            }bits;
        }LSB;
        //MSB，高字节定义
        union{
            UINT8 B8;
            struct{
                UINT8 Reserved2        :1;  //8，保留
                UINT8 Security         :1;  //9，安全位
                UINT8 Reserved3        :6;  //10~15，保留
            }bits;
        }MSB;
    }MLSB ;
}NWKPKT_Control_Struct;

//4、网络层接收到下层来的数据包结构体
typedef struct {
    Bool    PKTFlag;                          //网络报文有效标志
    NWKPKT_Control_Struct PKTControl;         //网络报文控制域
    UINT16  nwkDestAddr;                      //目的地址
    UINT16  nwkSourceAddr;                    //源地址
    UINT8   nwkPayLoadLen;                    //网络载荷长度
    UINT8   nwkPayLoad[nwkcMaxPayloadSize];   //网络载荷域
}NWKPKTStruct;
```

# 附录 B 程序文件列表

| 目录 | 文件 | 文件简要描述 | 代码行数 |
|---|---|---|---|
| MC13192 | MC13192.c | 主要包括了 MC13192 硬件配置函数 | 317 |
| | MC13192.h | MC13192.c 文件的头文件 | |
| ZigBee | Zig_PHY.c | 包括了实现的 IEEE 802.15.4 物理层所有函数 | 668 |
| | Zig_PHY.h | Zig_PHY.c 文件的头文件 | |
| | Zig_MAC.c | 包括了实现的 IEEE 802.15.4 MAC 层所有函数 | 1699 |
| | Zig_MAC.h | Zig_MAC.c 文件的头文件 | |
| | Zig_NWK.c | 包括了实现的 ZigBee 的网络层所有函数 | 964 |
| | Zig_NWK.h | Zig_NWK.c 文件的头文件 | |
| MC9S08GB60 | MCU_hw_Config.c | 主要为 GB60 初始化程序，其中包括了 MCU 对 MC13192 的初始化工作 | 193 |
| | MCU_hw_Config.h | MCU_hw_Config.c 文件的头文件 | |
| | SCI.c | GB60 的串行通信模块程序 | 152 |
| | SCI.h | SCI.c 文件的头文件 | |
| | SPI.c | GB60 的 SPI 通信模块程序 | 114 |
| | SPI.h | SPI.c 文件的头文件 | |
| | LCD.c | GB60 对 MT-ZigBee 的液晶模块程序 | 91 |
| | LCD.h | LCD.c 文件的头文件 | |
| | Key&LED.c | GB60 对 MT-ZigBee 的键盘和指示灯模块程序 | 101 |
| | Key&LED.h | Key&LED.c 文件的头文件 | |
| | vector.c | GB60 的中断入口函数 | 258 |
| 主目录 | Main.c | 主函数入口 | 475 |
| | includes.h | 主要头文件 | |
| | Pub_def.h | 设计中的类型定义 | |
| 统计 | 总代码： 5032 行 | 最终目标二进制代码： 14.2Kb | |

# 攻读学位期间公开发表的论文及参与的鉴定项目

[1] 刘辉,王宜怀等.一种应用在宠物身份识别中的RFID阅读起的开发.计算机应用与软件，2007.12

[2] 作者参与了2006年第一届“飞思卡尔”杯智能车邀请赛，于2006年8月在清华大学顺利完成比赛，并获优胜奖

[3] 作者参与了《16位IDE的开发》项目，该项目已于2005年11月通过校专家组评审

[4] 作者参与了《应用于动物识别的特种电子标签及相关读写器中试》项目，国家科技部技术创新基金项目(04C2262232200503)，该项目已于2007年4月通过评审

# 致 谢

研究生生活即将结束，三年的苏大学习使我获益匪浅。

在这里我首先要感谢的自然是我的导师王宜怀教授。王老师，您渊博的知识、开阔的视野和敏锐的思维，尤其是您严谨求实的治学态度，无论在专业领域还是日常生活中都给了我深深的启迪，教导我怎样去为人、做事。

感谢陆晓峰老师和刘晓升老师，在这三年的实验室生活和相关的项目开发中，你们都给了我无私的帮助，让我在专业技术领域学到了好多。

三年的生活中，相处最多的自然是实验室的兄弟姐妹。融洽的实验室生活是我这三年中最值得回忆的细节，真有幸成为这个大家庭中的一员，谢谢你们给予我各方面的帮助。

感谢我的父母，养育之恩，无以回报，你们永远健康快乐是我最大的心愿。

从入学，到这毕业之际，在这三年中有太多的师长、同学、朋友给了我无言的帮助，在这里一并表达我真挚的谢意！