论文题目　基于 CC2530 的社区无线抄表系统设计

工 程 领 域　电子与通信工程

指 导 教 师　毛玉明　　教　授

作 者 姓 名　李　翔

学　　号　201050101021

分类号　　密级

UDC [注1]

学　位　论　文

基于 CC2530 的社区无线抄表系统

（题名和副题名）

李 翔

（作者姓名）

指导教师姓名　毛玉明　教　授

电子科技大学　成　都

许淮武　高　工

中国工程物理研究院通信部　绵　阳

（职务、职称、学位、单位名称及地址）

申请专业学位级别　硕士　专业学位类别　工 程 硕 士

工程领域名称　电子与通信工程

提交论文日期　2012.6.1　论文答辩日期　2012.6.2

学位授予单位和日期　电 子 科 技 大 学

答辩委员会主席　黄 炜

评阅人　廖丹　邱一佳

2012 年 6 月 1 日

注 1：注明《国际十进分类法 UDC》的类号

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

签名：_____ 日期：　　年　　月　　日

论 文 使 用 授 权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

签名：_____　　导师签名：_____

日期：　　年　　月　　日

摘　要

随着电子和通信技术的发展，传统的人工抄表方式已经不能满足社会发展以及阶梯抄表的需求。对于人工抄表的方式，存在人力和物力的浪费。在步入 21 世纪的今天，各行各业都不断向智能化的发展。因此，智能抄表的系统正在不断地被人们提出来。Zigbee 是近几年来发展的一种非常热门的无线通信技术，它低功耗，低速率，低成本，它主要用于各种短距离，低功耗，低速率的各种仪器设备间的传输数据。基于此，本文提出了利用 Zigbee 技术在无线自动抄表领域的应用，通过使用 TI 公司最新的 CC2530 芯片，结合 ST 公司的 STPM01 功率测量芯片，可以达到测量准确，在小区内智能化抄表的目的。

本文所设计的无线抄表系统分为信号采集、数据传输、控制中心三部分，利用意法半导体公司生产的 STMP10 芯片作为前端电能表的数据采集设备，同时利用 TI 公司的 CC2530 芯片作为无线传输模块，上位机则是社区物业或电力主管部门所使用的控制中心。本文分为六个部分，前两部分分别对课题的背景以及笔者对 ZigBee 技术的特点的学习进行知识总结，对目前常用的几类无线传感技术与 ZigBee 技术进行详细对比，详细介绍了本设计中所使用到的协议栈，网络拓扑等知识，并对 ZigBee 技术的各协议栈架构以及 STMP01 芯片的技术特点采用文字加图片的方式进行了大量的论述。其次，论文阐述了系统的设计思路，同时对系统所需软硬件系统做出分析，第三，论文分别针对系统中的硬件以及软件系统的关键部分做出详细论述，详细介绍了电源电路已经传感电路的设计思路和设计方法，分析了 CC2530 与 STPM01 芯片间的通信过程已经数据采集。最后对系统的结论及相关测试结果做出适当论述，并指出目前系统存在的不足，以及系统后续的研究内容。

设计经过实验证明，在设计所模拟的采集环境内，能较为准确的测量出电能表在某一时刻的读数，将来可以在此基础上对系统添加分时读表的功能，即可满足阶梯抄表的需要，本设计具有一定的实用价值，为未来社区无线抄表的普及提供了较好的思路。

关键词：ZigBee，STPM01，无线传感技术，CC2530，无线抄表系统

ABSTRACT

The development of electronic and communication technologies, the traditional manual meter reading cann’t longer meet the needs of social development and the ladder meter reading. For a manual meter reading, the waste of human and material resources. Into the 21st century, businesses are constantly to the intelligent development. Therefore, the smart meter system is to be raised. Zigbee is a very popular in recent years the development of wireless communication technology, low power, low-rate, low cost, it is mainly used for transmission between a variety of short-range, low power, low-rate variety of equipment the data. Based on this, Zigbee technology in the field of wireless automatic meter reading applications, by using TI's CC2530 chip, combined with ST's STPM01 power measurement chip, and can achieve the measurement accuracy in the district of intelligent meter reading purposes .

The wireless meter reading system designed in this paper is divided into signal acquisition, data transmission, the control center of three parts, the use of the STMicroelectronics production STMP10 chip as a front-end energy meter data acquisition equipment, while taking advantage of TI's CC2530 chip as a wireless transmission module , PC is a community property or the power used by the department in charge of the control center. The paper is divided into six parts, the first two parts summarize knowledge on the background of the subject and author of the characteristics of ZigBee technology learning, a detailed comparison of the most commonly used class of wireless sensor technology and ZigBee technology, described in detail in this design knowledge of the protocol stack, network topology, the protocol stack architecture and STMP01 of technical characteristics of the chip and ZigBee technology used to the way text and picture a lot of exposition. Secondly, the paper described the system design and analysis of the system hardware and software systems required to make, papers, respectively, for a key part of the system hardware and software systems to make a detailed discussion details the power circuit has been transmitted sense circuit design ideas and design methods, analysis of the communication process between the CC2530 and STPM01 chip has data acquisition. Finally, conclusions and test results to make

appropriate discourse, and pointed out the shortcomings of the current system, and the systematic follow-up research.

The design has been proved through experiments in the design of analog acquisition environment can be more accurately measure the power meter readings at a particular moment, can add timeshare meter reading on this basis, to meet the ladder copy the needs of the table, this design has a certain practical value, provide a better idea for the future popularity of wireless meter reading in the community.

Key words: ZigBee,STMP01,wireless sensor technology,CC2530,wireless meter reading system

目 录

第一章 引言

1.1 课题来源及相关背景

1.1.1 课题来源

自从人类发现了电，并将水、电、气能能源引入到平常百姓的生活中后，各种各样的仪表数量也发生了飞跃的发展，随着人们生活质量的提高，传统的手工抄表，人工录入等读表记录方式渐渐不能满足人们现有的对读表方式的需求，也难以满足阶梯电价的计费要求。由于时常忘记抄表，而造成的因阶梯价格多交电费的情况时有发生。伴随着信息技术的发展，智能家居、工业智能化已经自动化技术的发展，人们越来越渴望能够实现室内电表的智能抄收。

常用的电子电表分为两种：自轮式和液晶显示式。相对于机械式仪表，它可以提供更低的功耗，更高的精度，出色的可靠性。同时，电子电表在恶劣的工作环境中的鲁棒性范围更广。电子式电能表也有许多新的功能，如多速率的计算和通信能力，支持预付费防盗系统功能。随着社会的发展，用电量增加，电子电表的市场将更加广阔。

电子式电表的改革也引发了相应的芯片式电表生产厂家的发展，并带动了市场的进步，但也加剧了终端芯片制造商在电能计量芯片领域的竞争，由于国内标准制定的滞后等因素。各种电能表在激烈的竞争中，而没有哪个电能表可以发挥主导作用。因此，在电力部门在表库有多种时代和不同的制造商和各类电能表[1]。

随着电子、电气和技术产品的广阔应用，人们在享受这些产品的同时，往往也受到来自各种电缆、数据线的束缚。这些电缆、数据线线虽然肩负着不同的功能，却也有着以下缺点：首先限制着运营商的手脚；其次造成电器用品的浪费；第三废电器用品也造成环境的污染。第四，电器老化和不当布局等原因，也埋下了不小的安全隐患。此外，大型有线网络的铺设是容易受到地形，地貌，巨大的前期投资，后期难以维护，容易产生浪费和重复建设等问题的局限。鉴于上述考虑，以及对未来技术的发展思考，越来越多使用无线接入技术的设备逐步取代和淘汰各种家用电气，工业设备和电线，电缆，电器产品。让用户真正享受无线操作无束缚和便捷的服务。无线接入技术，也是 IT 业的发展趋势[2]。

随着无线技术的发展，蓝牙、Wi-Fi 等技术逐渐被应用在无线网络的使用过程当中，而 ZigBee 技术也是在这几年兴起的一类低速率无线传感技术，它具有功耗及成本较低而且容易安装的特点。该项技术的出现对当前布线复杂、抄表效率低的传统抄表系统来说，无疑是一种很好的解决方案。

1.1.2 国内外研究概况

早在上世纪 80 年代，发达国家便开始使用无线技术来收集信息和实验，如在美国的芝加哥地区，韩国和其他国家，特别是欧洲。然而这些努力都由于无线定位和通信技术有限公司是不够成熟，并没有得到很好的效果。直到九十年代，伴随着无线通信技术和 GPS 技术在这一领域的普及，这方面的国际工作也逐渐丰富起来。

例如，英特尔公司和加州大学伯克利分校，所领导的“dust”技术研究的工作。他们已经成功地创建一个全能的传感器，而其只有瓶盖大小，却可以实现计算、测试和通信等功能。

在日本，日立 YRP 网络研究所在 2004 年开发出世界上最小的无线传感器网络终端，虽然该网络需要使用有源无线终端，但却能够搭载多种传感器以测量温度，亮度，红外线，加速等。这种网络终端正可运用于智能家居，无线抄表等领域[37]。

我国于 1999 年开始做这方面的研究，这与国际上的相关国家的启动时间是一致的，在中国科学院当年的信息与自动化领域的相关研究报告中有着明确的体现从 2004 年起在北京进行了大规模外场演示，其中部分成果已使用在实际工程系统中。同时，国内许多的高校也在积极做出这方面的基础研究和应用研究。国内数所院校也纷纷开展了有关无线传感网络的基础研究当中。与此同时，国内的许多企业也加入到这个应用行列中来[3]。

1.1.3 国内智能抄表系统的现状

在我国已经主要应用的远程无线自动电抄表系统中，按底层通讯方式的不同分为以下三种：

1.1.3.1 通过电力载波通信的方式来实现远程电抄表

这种通信方式不需要铺设通信线缆，可以节约资金投入，同时也具有维护量小的特点，但目前电力载波通讯有许多不足之处。在我国许多低压配电网中，其电力负荷变化大且没有明显的规律性，导致干扰十分严重，因此，将扩频通信技

术应用在自动电抄表系统，目前在我国并未取得好的效果。

1.1.3.2 通过 RS485 有线通讯网实现远程电抄表

这种通信网具有抗干扰能力强，传输距离可达 1000 米以上，数据速率高等特点，但由于 RS485 通讯网是有线网络，则需要专门铺设相应的传输设备，这就需要额外增加投资和费用的开支，因此，也不能满足一些网络借点较多用户的需要，在我国，这种应用方式目前主要应用在少量高档生活小区[4]。

1.1.3.3 通过 GPRS 无线通信网实现远程电抄表

GPRS 是通用无线分组业务的英文简称，GPRS 作为一种已被广泛使用的的无线传输系统网络，其特点主要表现在：传输率大，频率利用率高，数据传输有可靠性保证等方面，因此可以作为一种远程自动抄表的技术方案。但是其设备的价格较高，而且功耗也较大。由于 GPRS 无线自动抄表系统自投入使用开始，用户就需要向运营商支付流量费用。再加上其安装、维护和使用成本来看，其使用成本远远高于人工抄表。因此，目前并没能被广泛应用[4]。

1.1.4 研究前景

ZigBee 技术具有网络自组织功能，可以自动路由，组成多跳网络。国内外已经开发出一些基于 ZigBee 的无线抄表系统,但是距离实际应用还是有一定的距离，有些是 ZigBee 与 RS232、电力线、GPRS/GSM 等方式结合实现部分无线抄表功能[5]。本文通过使用 TI 公司最新的 CC2530 芯片，结合 ST 公司的 STPM01 芯片，提出了一套全网络全功能的 ZigBee 无线抄表系统,现在一个区域内所有用户的无线抄表，满足目前小区形式的主流居住环境，又节约了小区各用户的 GPRS 通信成本或布线及维护成本，对传统人工抄表模式做出了根本上的改变。

1.2 本文的主要工作和组织结构

1.2.1 论文完成的主要工作

本文的主要工作是研究基于 IEEE802.15.4/ZigBee2007 协议的无线传感网络，设计一种采用 Ti 公司的 CC530 的无线传感网络，同时为了避免 CC2530 直接采集用电数据精度不高，算法难度大的缺陷。在测量用电数据的模块中我们采用了 ST 公司的 STPM01 功率测量芯片，可以达到无线传输和测量精度较高的目的。

在本系统中，设计在社区多个家庭中设置测试点，各个测试点将所探测到的电量数据信息进行初步的处理和融合，接着发送到控制中心，数据传送的过程是通过使用 CC2530 传输模块，将各个测试点的数据互传，并以接力的方式传送到主控中心，主控中心在收到传输模块传来的数据信息后，通过有线连接传送给用户。无线传感器网络与其他传统的网络相比有一些独有的特点。

1.2.2 论文结构

结合课题研究期间的工作，论文划分为七章撰写，具体如下：

第一章：主要介绍了课题的背景，已经目前国内外对无线传感技术的研究和应用情况。同时，对无线传感技术的研究前景以及方向做了一定的阐述。同时简单介绍了本文的知识结构。

第二章：主要介绍 ZigBee 协议栈的知识，同时列举出几类目前比较常用的无线传感技术，并对其优缺点进行深入学习后，做出总结。将几种常用的无线传感技术与 ZigBee 技术的对比，提出该技术在本设计中的优势，通过对 ZigBee 协议层和协议栈的知识的学习，做出了大量的介绍。

第三章：对比几种可以通过 ZigBee 技术实现本设计的方法进行对比，提出本设计的可行性.提出本文的总体设计思路，以及主要技术路线，各模块的主要功能。

第四章：主要介绍设计所使用的硬件系统，并画出设计中所采用的电路图。

第五章：主要介绍了软件系统的设计思路，以及设计中所用到的一些关键烧录程序。

第六章：详细介绍了实验结论及相关实验数据。

第七章：是对全文的总结以及对未来发展方向的展望。本章总结设计的设计过程，指出了设计的不足之处，并对未来的研究方向提出了设想。

第二章 IEEE802.15.4/ZigBee 通信标准

2.1 ZigBee“紫蜂”

ZigBee 是一种基于 IEEE802.15.4 协议的一种无线网通信技术，其适用范围包括一些通信数据量，数据传输速率较低，且分布的范围有一定局限性的环境当中，但数据的安全可靠性有一定要求，由于成本和功耗极低，同时安装比较容易和使用场合比较广泛的原因。“HomeRF Lite”或者是“FireFly”是 ZigBee 的另一个称谓，近距离无线连接是其主要的应用领域。ZigBee 组成很多个微型传感器网络，传感器可以通过 ZigBee 自己的无线电协议进行通信。ZigBee 节点间的通信效率非常的高，相邻的传感设备之间只需要很低的能量消耗，就可以无线接力的方式将数据在相关传感设备之间进行数据传输。通过一定的通信协议，ZigBee 的数据最终可以送给计算机进行数据交换。

IEEE802.15.4 是电气电子工程师协会由所确定的一种面向对象的低速无线网络标准，该协议的物理层和介质访问层是由电气电子工程师协会由所所定义。IEEE 802.15.4 是一个低功耗协议，IEEE 802.15.4 目标是为在个人操作空间（personal operating space, POS）内相互通信的无线通信设备提供通信标准。而这个个人操作空间则是指一般是指在用户周围 10 米左右的空间范围，在这个范围内用户可以是固定的，也可以是移动的。因为其范围小，所以需要的发射功率就小，众所周知发射功率随着距离的增长呈指数级增长，而发射功率是整个无线节点耗电量最大的，发射功率较低无疑是从根本上决定了这是一个低功耗的协议。MAC 层控制着节点接入无线信道的方法，发起网络，关联等一些核心操作都是 MAC 层完成的，而且正是因为有个 IEEE 802.15.4 的 MAC 层，才更决定了这是一个低功耗协议。ZigBee 联盟定义了 ZigBee 协议栈的网络层及应用层。在工业监控、家庭监控、安全系统、传感器网络的无线通信等技术的应用中，ZigBee 被认为是最合适的一种技术[5]。

2.2 典型短距离无线通信网络技术

随着数字通信和计算机技术的发展，科研工作者们提出了大量短距离无线通

信的要求。而短距离无线通信和长距离无线通信有着多样的区别，其主要特征表现在以下几点：

a) 无线发射功率为几μW到100μW；
b) 通信距离范围是几cm到几百m；
c) 主要应用于室内；
d) 使用全向天线和线路板天线；
e) 不用申请无线频道；
f) 高频操作；
g) 电磁供电的无线发射器和无线接收器。

无线网络可以利用RF（无线电射频）或IR（红外线）等无线传输媒体与技术构成的通信网络系统。由于取消了诸如双绞线、光纤、同轴电缆等有线介质。使得网络用户真正达到了“信息无处”不在的理想境界。

目前，无线局域网（Wi-Fi）、超宽带通信（UWB）、近场通信（NFC）、蓝牙（Bluetooth）、红外线数据通信IrDA和ZigBee等五种短距离无线网络技术正在成为业界谈论的热点。

2.2.1 Wi-Fi（IEEE 802.11）

Wi-Fi(wireless fidelity)即IEEE 802.11x，其最初规范是在1997年被提出的。是一个建立在IEEE 802.11标准的无线局域网络（WLAN）基础上的设备。由于两套系统之间有很多相关之处，很多人也把Wi-Fi称作IEEE 802.11。Wi-Fi可以实现几兆到几十兆的无线局域网接入。Wi-Fi部署区网（LAN）可让客户端设备无需使用电线，通常可以降低部署网络和扩充网络的成本。许多空间不能敷设电缆，比如室外空间以及一些名胜古迹，但是这些环境却能通过使用无线局域网来进行改善。现在几乎所以的数笔记本电脑制造商在产品出厂时就已经内建了无线网络的装置。随着Wi-Fi的价位的持续下跌，使该技术在经济网络选择中被广泛运用，特别是在企业办公中。 不同竞争品牌的接入点和客户端的网络使用接口具互操作性的一个基本服务水平。“Wi-Fi”不同于移动电话，它指定了一套全球统一的标准:，任何Wi-Fi标准设备可以在世界上任何地方无差异的工作。 Wi-Fi已在22万个以上的公共热点和几千万户的家庭,公司及世界各地的大学校园中使用。但是Wi-Fi却需要靠牺牲大量的功耗来换取其足够的带宽，造成了使用该技术的相关硬件设备也需要为此消耗大量电量，因此，这也限制了其在功能简单的低功

耗应用等的应用及推广。

2.2.2 超带宽通信 UWB

UWB（Ultra Wide Band，超宽带）是一种能够穿透墙壁等障碍物的无线传感技术，它可以极低功率在短距离内进行高速的数据传输。UWB 技术可以调整信号频宽和发射功率的基础是根据数据传输的距离和数据量的大小，UWB 技术主要用于无线数据传输上，但目前已逐渐发展到雷达、跟踪定位等领域。军方很早便已开始使用 UWB 技术。UWB 主要用于实现无线电定位系统，以及位置和距离的精确测量上，如在汽车碰撞定位检测器；同时，由于 UWB 使用的频带带宽高，并且能够穿透障碍物，所以 UWB 也被使用在一些雷达应用上，如警察解救人质时所需的穿墙成像系统等。UWB 技术的功耗很低，其功耗量级为微瓦级。但是，UWB 芯片组的功耗大约是毫瓦量级比其技术本身的功耗要大很多。与其他传统的无线通信技术相比，UWB 技术适合于便携式应用，因为 UWB 的传输速率快和通信距离很短，平均发射功率低，难以满足在社区抄表这类环境中的传输距离的需要[6]。

2.2.3 近距离无线通讯技术 NFC

NFC 是 Near Field Communication 缩写，即近距离无线通讯技术。飞利浦公司和索尼公司共同开发了该项技术。近距离无线通讯技术被广泛运用在移动设备、消费类电子产品、PC 等设备间，它不需要设备间进行实质的接触便可以完成识别和互联等功能。近距离无线通讯技术所提供的解决方案是简单、触控式的，能够实现消费者简单直观地信息交换、以及内容的访问和服务。

近距离无线通讯技术是一种在一块单芯片中将接触读卡器、非接触卡和点对点（Peer-to-Peer）功能整合的技术，以实现用户多样化的数字体验。NFC 技术平台的接口对外开放，可以对无线网络进行快速、主动数据设置，同时，NFC 也是虚拟的数据连接器，可以服务于现有 GSM 网络、蓝牙和无线 802.11 设备。

NFC 具有价格低廉、简单易用等特点，这使得其潜力在无线传感应用领域显得极为突出，NFC 通过组合芯片、天线和软件等，能够在几厘米范围内实现各种设备的通信，而费用仅为 20-30 元。但由于 NFC 的数据传输速率较低，仅为 212Kbps，不适合需要较高带宽的应用[7]。

2.2.4 蓝牙

蓝牙技术是一种短距离通信技术，它可以传输音频信号，其工作频率为2.4GHz，采用 FHSS 扩频方式。爱立信公司在 1994 年开始对蓝牙技术进行研究。其主要研究领域是实现蓝牙技术在移动电话和其他配件间进行低成本、低功耗无线通信。现在蓝牙可以用在不同的设备之间以进行无线连接，比如连接手机或个人终端与计算机实现通信，连接计算机和外围设备等。现在在两个都使用了蓝牙技术的设备之间几乎都可以进行传输数据。但是蓝牙在 2.4GHz 的会受到电波干扰却一直难以解决，特别是在和无线局域网一起使用时会互相干扰。和 zigbee 低功耗不同，蓝牙的功耗一般都比较大，由于蓝牙在睡眠状况下所消耗的电流已经激活延迟，一般的电池能够使用 2~4 个月[8]。

2.2.5 红外通信技术

数字红外通信技术是利用红外线实现点与点之间通信的无线传感技术。目前红外线数据通信的传输速率已从最初的几百 kbit/s 逐步发展到了目前的几 Mbit/s。目前，能够支持该技术的软件及硬件技术都比较成熟，已被广泛使用在小型的移动设备上。红外线在点对点通信有其特有的优势。但是也正是红外线数据通信只能实现点对点的通信，而且也存在有视距角的问题，所以很难应用于工业网络上[3]。

2.2.6 短距离无线通信协议比较

本方案选择 ZigBee 技术作为系统的主要传输技术，其原因如下：

a) 省电：两节五号电池可使用 6~24 个月，避免因电量不足等原因，需要相关维护人员随时对系统更换电池，降低维护的工作量；
b) 可靠及自组网：可以避免在发送数据时发生竞争与冲突；通过 ZigBee 在整个传输网络中的自动路由，确保信息传输的可靠。但也能保证数据传输的准确性。
c) 时延短：使用技术手段优化了对时延的敏感，有效的降低了通信时延以及休眠激活时延；
d) 网络容量大：可支持节点达 65000 个，可以满足各种规模社区的使用；
e) 高保密性：64 位的编号并且支持 AES-128 加密[7]。

综上特点，本人将 ZigBee 与其他几个比较常用的无线传感技术绘制出相应表

格进行对比，见表 2-1。

表 2-1 ZigBee 与其他几种通信技术的比较

名称	ZigBee	蓝牙	Wi-Fi	GPRS/GSM CDMA
应用范围	监测控制	电缆替代品	WebE-mail，图像	广泛，声音数据
工作频段	869/915M/2.4GHz	2.4GHz	2.4GHz	900、1800MHz
系统资源 KB	4-32	250	1024	16384
电池寿命（天）	100-1000	1-7	0.5-5	1-7
网络大小	255/65535	7	32	1
带宽（KB/S）	20-250	720	11000	64-128
传输距离	1米-100米	1米-10米	1米-100米	1000米
优点	可靠，低功耗，便宜	方便，便宜	灵活	覆盖面广

2.3 ZigBee2007 协议分析

2.3.1 ZigBee2007 协议概述

ZigBee 的物理层以及媒体介质层所使用的协议标准 IEEE802.15.4 协议标准，而 ZigBee 技术联盟制定了网络层的协议，用户可以根据自己的应用需要对应用层进行开发应用。因此 ZigBee 技术能够为用户提供机动、灵活的组网方式，图 3-1 所显示的为 Zigbee2007 协议结构。

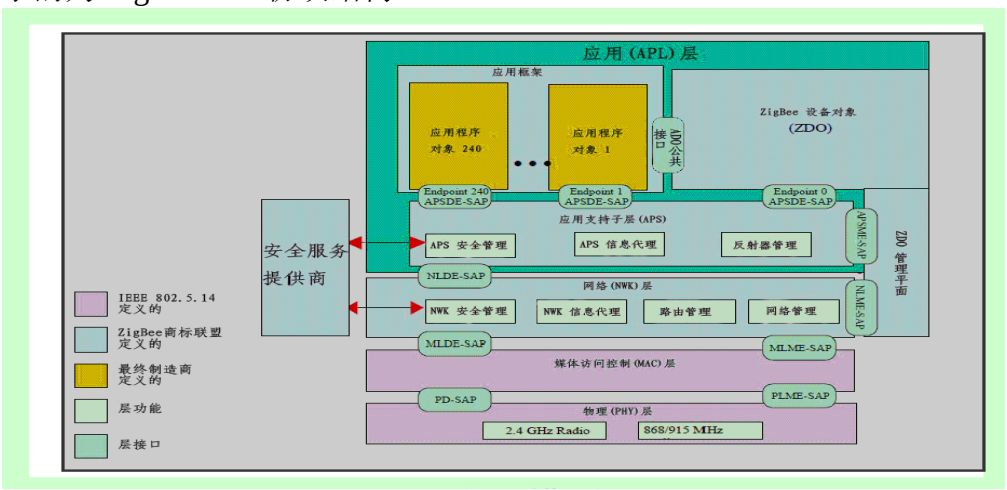


图 3-1 Zigbee 协议结构

2.3.1.1 IEEE802.15.4 的物理层和介质访问控制层协议

物理层所定义的通信频率有三种，分别是 868MHZ,915MHZ 和 2.4GHZ，三种频段所使用的传输速率以及调制解调方式不同，2.4GHz 频段，是免费频段，包含 16 个信道，是全球通用免申请频段；其传输速率可达 250kbps。915MHz 包含 10 个信道，传输速率可达 40kbps；而 868MHz 频段的信道只有 1 个，传输速率为 20kbps[9]。

媒体介质层定义了网络内的各种帧格式，通过使用 16bit 短地址和 64bit 长地址实现：这里面的每个物理节点都是在出厂时就设定好了的，同时这也是一种全球唯一地址结构，而这些节点的标记工作则是通过长地址的标记来完成的；每个加入到网络节点被分配的 16bit 地址就是短地址，它是通过网络协调器分配的，网络协调器的主要功能就是负责启动无线网络的节点。网络内的节点的寻址方式可以通过使用网络标识加短地址或长地址实现。数据帧，命令帧，ACK 确认帧和信标帧是网络间传输三种帧结构[10]。

IEEE802.15.4 中也详细描述了如何通过在物理层加密数据以启用安全模式的无线网络。

2.3.1.2 ZigBee 标准协议制定了网络层及应用层

在应用层内又提供了应用支持子层和 ZigBee 设备对象应用框架中也加入了用户自定义的应用对象[11]，如图 3-2。

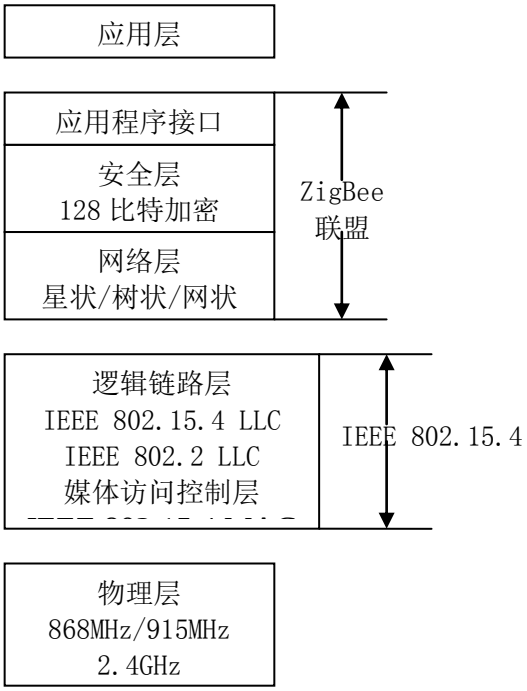


图 3-2 ZigBee 协议栈

2.3.2 物理层规范

物理层需要对信道和数据传输之间进行管理，而这种管理方式的实现，是建立在硬件驱动程序的基础上。数据的发送和接受构成了数据的传输，物理层的管理内容体现在：链路质量、空闲信道的评估、信道能力监测等，如图 3-3。物理层中的 PD-SAP 所提供的接口则是由物理层提供给介质访问层的数据服务接口，而物理层管理实体是物理层给介质访问层提供的管理服务接口。物理层主要完成：休眠无线收发设备、激活、连接质量指示、对当前频道进行能量检测、频道选择、数据的发送及接受等。

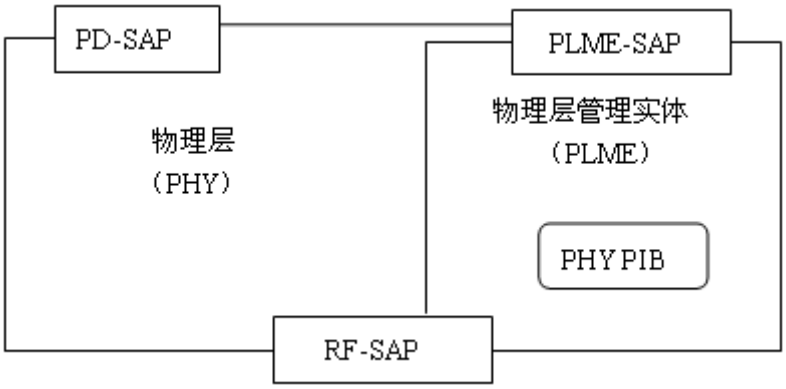


图 3-3 PHY 模型

在物理层的有关参数中，有四个重要参数：

a) 传输能量约为 1 毫瓦；
b) 传输中心频率的兼容性；
c) 接收器的感度；
d) 接收信号强度只是的测量（RSSI）。

2.3.3 PanID 和通道（Channel）选择

ZigBee 协议规范规定，每个唯一的网络需要通过一个 14 位的个域网标志符（PAN ID）来进行标识。Z-Stack 可以用两种方式由用户自己选择其 PAN ID，当 ZDAPP_CONFIG_PAN_ID 值设置不为 0xFFFF 时，那么设备建立或加入网络的 PAN ID 由 ZDAPP_CONFIG_PAN_ID 指定；如果设置 ZDAPP_CONFIG_PAN_ID 为 0xFFFF；则设备会进入一个它在网络中搜索到最优质的网络。在 2.4G 频段上，IEEE 802.15.4/ZigBee 规范规定了 16 个频道。用户可以通过选择 DEFAULT_CHANLIST 不同的值可以选择不同的频道，协议默认频道为 0xB 及 0x00000800。

2.3.4 网络层规范

2.3.4.1 ZigBee 设备的类型和结构

在网络层中，ZigBee 定义了 3 种角色：第一个是协调器，负责网络的建立及位置的分配，同时存储关于网络的信息，包括作为认证中心的和作为安全密钥的储藏所；第二个是路由器，主要负责找寻信息包、建立并修复数据包的路径，并负责转送数据包；第三个是终端，终端通过智能选择加入网络，并收发信息，只是不具备转发信息和路由的功能。通常，全功能装置（FFD）实现协调器和路由器，而终端由简化功能装置（RFD）实现。在组网方式上，ZigBee 主要采用大约三类组网方式如图 3-6 所示：

第一种网络结构是星型网，它主要包括协调器和最多 65535 个终端；

第二种是为簇状型网，他是由扩展的星型网组成；

第三种为网状网，它是一种新型无线网络技术。在这种网状网络中，所有无线设备节点的功能可以一样，都可以同时作为路由器和发射点，所有在网络中设备节点都能够进行数据的发送和接收信号，这些无线节点间可以相互之间做对等的数据交流[12]。

此结构的好处是：它可以使数据自动选择到一个通信流量较小的邻近节点进行传输，而避免了 AP 由于流量过大而导致拥塞的问题。数据包能够对网络情况进行自适应，继续路由到与之最近的下一个节点进行传输，直到到达它的最终目的地为止。

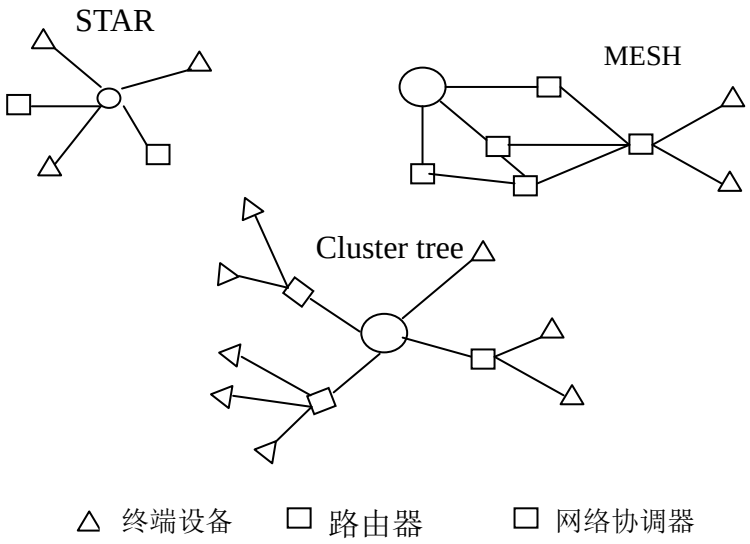


图 3-6 ZigBee 网络拓扑结构

2.3.4.2 ZigBee 网络管理

这个功能将执行 ZigBee 协调器、ZigBee 路由器或者 ZigBee 终端设备逻辑设备类型根据已确定的配置设置，通过程序应用或者在安装期间。如果设备类型是一个 ZigBee 协调器或者 Zigbee 终端设备，这个功能将提供选择一个存在的 PAN 来加入和如果网络通信断开执行允许设备重新加入的程序的能力。如果设备类型是 ZigBee 协调器或者是 Zigbee 路由器，这个功能将提供为一个新的 PAN 建立选择一个未用的信道。注意在没有一个设备是预先指定为协调器的情况下，配置一个网络是可能的，这时，第一个全功能设备（FFD）被确定为 ZigBee 协调器的角色。网络管理做如下处理：允许为网络信道列表的规定扫描程序。缺省值是规定在已选择的操作联合的所有信道的使用。管理网络扫描程序来确定邻居网络和它们协调器和路由器的一致性[13]。允许一个信道的选择来启动一个 PAN（ZigBee 协调器）或者一个存在的 PAN 的选择来连接（ZigBee 路由器或者 Zigbee 终端设备）。支持孤点和扩展的程序来重新连接网络，包括支持可携带的内部 PAN。也许支持直接连接。对于 ZigBee 协调器和 ZigBee 路由器，直接连接的一个本地版本被支持来使能设备通过孤点或者重新连接流程来加入网络[14]。

2.3.4.3 ZigBee 绑定管理

绑定管理执行下列任务：

首先绑定表建立标识值。要建立的标识值确定了一些了计划的申请，并在设备安装中定义配置参数。通过实体处理绑定请求以确定绑定表增加或者减少。ZigBee 设备 Profile 支持绑定和解绑定命令（见 2.4 节）。同时 ZigBee 协调器，可以绑定终端，而这种绑定是允许以按钮按压或其他手动菜单为基础的绑定。

当我们需要对操作进行绑定时，那这个端点就必须向协调器发送绑定请求，协调器就会在有限的时间间隔内对接收到的绑定邀请后，在这两个不同的端点之间建立绑定表以形成逻辑链路。所以我没说两个端点在进行绑定后的两者之间所进行的消息传送这个过程应该属于消息的间接传送。其中一个端点首先会将信息发送到 ZigBee 协调器中，ZigBee 协调器在接收到消息后会通过查找绑定表，对这两个端点与协调器之间形成绑定。

2.3.4.4 节点管理

对于 Zigbee 协调器和路由器，节点管理功能执行以下步骤：

（1）允许遥控操作命令来执行网络发现

（2）提供遥控操作命令来重新获得路由表

（3）提供遥控操作命令来重新获得绑定表

（4）提供一个遥控操作命令来使一个设备离开网络或者是命令另一个设备离开网络

（5）提供一个遥控操作命令来重新获得 LQI，是为这个遥远的设备的邻居获得的。

（6）允许源设备向一个初始化绑定表高速缓冲寄存器登记的能力来保持他们自己绑定表

（7）允许配置工具把一个设备换成另一个设备，这个设备是在所有的绑定表入口中，这个入口涉及到他。

（8）允许初始化绑定表高速缓冲寄存器备份和恢复个人绑定入口或者入口绑定表或者保持他们自己绑定表的源设备的表

（9）提供一个遥控操作命令来允许或者禁止连接一个特殊的路由器；或者通常允许或者禁止通过信托中心连接[15]。

2.3.5 应用层规范

在 ZigBee 的协议里，应用层由用户应用程序、设备配置层以及应用支持子层（ZPS）等几部分组成。

用户应用程序的主要功能包括：

（1）依据设备的服务及需要对相应的设备实现配对,；

（2）在实现配对的绑定设备间完成消息的传送；

（3）完成 16 位网络地址与 64 位 IEEE 地址间的地址映射；

（4）对重要数据进行分段、重组以保证数据在传输过程中的可靠性。

设备配置层的主要功能包括：

（1）实现网络内部的角色的定义；

（2）寻找网络设备以及分配网络设备所提供的应用服务；

（3）初始化设备的同时对响应绑定请求；

（4）建立绑定表，以实现网络设备间的联系，同时保证数据的安全。

应用层的 ZPS 子层所提供的服务由 APSDE(应用支持层一下数据实体)及 APSME（应用支持层管理实体）两个实体提供。而其通过设备配置层和厂商所定义的应用对象来实现网络层和应用层所提供的接口服务。

2.4 ZigBee 的组网

2.4.1 ZigBee 的网络地址的分配

ZigBee 的网络地址分配是通过使用分布式寻址的方式来实现的。这样可以保证在中网络地址在整个网络中的地址唯一性，为了保证某个特定的数据包可以准确的发到目的设备。同时，该寻址算法本身的分布特性可以保证设备所接收到的网络地址来源只能通过与他的上一级设备通讯来实现。而无需对整个网络内的地址进行通信分配，这样便可以提升网络的可测量性。

2.4.2 ZigBees 的寻址

应用程序发送数据给网络中的特定设备所使用的函数是 AF_DataRequest()函数。数据包将要对目标设备发送给一个 afAddrType_t 类型的参数。数而据包可以被发送设备使用单点传送，多点传送或者广播传送发送，这三种发送方式的地址模式参数是不同的[16]。

2.4.2.1 单点传送

单点传送是标准寻址模式，通过向网络设备发送数据使其将 afAddrMode 参数设置为 Addr16Bit，数据包包含了发送给目的设备的数据和目的 ZigBee 设备的网络地址。

2.4.2.2 间接传送

使用此模式是因为应用程序不知道数据包的目的设备的网络地址。此时设置 STPM01_DstAddr.addrMode = AddrNotPresent ;目的设备通过在发送设备的栈绑定表内查找到其网络地址并使用该目标地址。这样，数据包就会被被处理成个标准的单点传送数据包。但是如果在绑定表中找到多个设备，那么就需要向这些设备都发送一个数据包的拷贝[16]。

2.4.2.3 广播传送

当网络中的一个设备需要向 ZigBee 网络中的所有设备发送数据包时，则需要通过广播方式实现数据的传送，此时 STPM01_DstAddr.addrMode =AddrBroadcast。

2.4.3 路由协议

ZigBee 所执行的路由协议是基于 AODV（Ad hoc On demand Distance Vector）的路由协议。ZigBee 路由协议支持环境中的移动节点，连接失败和丢包功能。其过程大致为：当路由器接收到一个 P2P 的信息包时，那么从他的应用或者从其他设备,NWK 层将继续向前依照下面的进程。如果目的节点是与路由器相邻的节点(包括它的子设备）之一，那么这个信息包就会直接传输到目的设备[17]。另外一种方式就是，路由器通过检查它的路由表格，并检查相应的信息包内的目的条目。如果在路由表格中有正在使用的路由路线可以到达这个目的设备，这个信息包将依照路由条目被转播到下一跳节点地址并储存[18]。如果没有活跃的条目被发现，那么路由就会发现被启动并且该信息就会被缓存直到过程完成。同时，ZigBee 终端设备不能执行任何路由功能。当一个终端设备想发送一个信息包到任何设备都要先把数据发送到其上一级设备，然后在由其上一级设备进行路由操作。类似的，任何设备想发送信息包到终端设备，都将发起一个路由发现操作，当然该操作都由终端设备的上一级设备响应[19]。

2.4.4 ZigBee 路由过程

ZigBee 路由包括了路由请求、路由响应、路由维护和路由终结四个过程[20]。

2.4.4.1 路由请求

路由通过请求以及信息包响应被发现[21]。一个源设备通过发送一个广播路由请求（RREQ）信息到它的邻设备请求一个路由以获得一个目的地址[22]。当某个节点接收到一个广播路由的请求信息时，它将依次转播这个广播路由的请求信息。但是在做这个之前，它需要更新广播路由的请求信息的消耗域，这个节点通过增加连接消耗以获得最后的连接。这样，广播路由的请求信息将携带向前传输的所有的连接消耗。这个重复过程直到这个目的设备收到广播路由请求信息为止。广播路由请求的一些复制一般会经过许多条不一样的路径多次到达目的设备。该目的设备选择最好的广播路由请求信息并发送一个路由答复（RREP）返回到源设备。

2.4.4.2 路由响应

路由答复会选择沿着唯一一条的相反的路径以返回到最初的路由路由请求节

点[23]。作为路由答复信息传播回源节点，在传输过程中遇到的转播节点会自动更新他们的路由表格，并指出相应的路由路线到目的设备[24]。一旦一个路由被创建，数据包能被发送。当节点传输过程中出现丢失（发送数据包时，它不能接收一个 MAC 应答 ACK），这个节点会通过发送一个路由答复到所有可能接收它路由答复的节点，使该路由无效。在接收一个 RREQ，RREP 或 RERR 之上，这些节点都将更新他们的路由表格。

2.4.4.3 路由维护

在前面我们提过，网状网络可以为网络提供路由维护和自动修复。中间节点保持沿着连接传输失效的路径。当某个连接被确定被证明无法进行传输了，那么其反向传输的节点就会为所有那些连接的路由路线启动路由修复。这些工作通过路由启动重新发送，为了路由下一次数据包接收。如果路由重新发现以后，路由还是不能启动，或者由于某些原因照成连接失败了，那么，就会对这个数据包的源设备发送一个路由错误（RERR）信息，然后重新启动新的路由发现[25]。总之，不管是使用哪一种方式，都可以实现这条路由的重新自建立。

2.4.4.4 路由终结

路由建立的过程就是对路由表格进行维护的过程。假设在某一个时间段内都没沿着路由路线发送的数据包，那么这条路由就会被做出标记。终止该路由而不是删除他。因此没有被删除直到它完全需要时。自动路由终结时间能被配置“在 f8wconfig.cfg″文件中”。设置 ROUTE_EXPIRY_TIME 参数为终结时间（秒）。设置 0 为了关闭路由终结。

2.4.5 ZigBee Profile

ZigBee 网络中两个设备之间通过将 profile 进行统一来实现通信。具体说来，假设规范一个 profile(可以理解成一套规定)，这个 profile 用来规范智能家居领域的相关产品都要满足那些要求，那么 home automation public profile 就规定了智能家居都要做什么。Profile 可以分为私人的和公共的两个等级。所谓私人的就是应用者定义一个自己的 profile，称为 provite profile[26]。公共的为 zigbee 联盟已经规已经规定的一些 profile，比如 home automation，smartenergy,building automation 等，一个 public profile 也规定了 profile

的 ID，比如智能家居就规定是 0x104。协议栈本身也有一个 profile,就是 Zigbee Device Profile,也就是 ZDP 了，这里规范了一个 zigbee 节点都要具备那些功能。Profile 的典型例子是智能家居[27]。智能家居 profil 允许一系列设备类型交换特定的控制消息来构造一个无线智能家居应用。这些设备被设计成很好的交换已知信息来实现这些控制，灯的开与关就是就是智能家居应用的一个例子。

ZigBee 中的一个枚举量参数就是 Profile 标识符。每一个 Profile 标识符定义簇标识符以及设备描述的一个联合的枚举量。比如说，对 Profile 标识符“1”，存在一些被 16 位值描述的设备描述(就是说在每一个 Profile 中可能有 65536 个设备描述）和一些被 16 位值描述的簇标识符（就是说在每一个 Profile 中可能有 65536 个标识符）。每一个簇标识符也支持一些被 16 位值描述的属性。例如，每一个 Profile 标识符最多有 65536 格簇标识符且每一个这样的标识符最多又可以包含 65536 个属性。而程序员的工作就是定义和分配设备描述、簇标识符以及为它们分配 Profile 标识符里的属性。注意设备描述、簇标识符和属性标识符的定义必须很小心的采用以保证简单描述的有效建立和当交换消息时单一化处理。
一个 ZigBee 设备可以包含多个 profile，这些 profile 是由在这些 profile 定义的各种簇标识符提供，同时，这些 profile 能够维持多样的设备描述。在设备里使用一个分层寻址定义的能力如下：

（1）设备：设备是由有网络地址和唯一的 IEEE 的单个无线电来维持的。

（2）端点：对不同的应用程序都进行了描述，而单个无线电都可以满足这些应用的维持。设备的 profile 是所有 ZigBee 设备必须使用的；其中端点 0xff 是用来寻址所有活动的端点（广播端点），且端点 0xf1-0xfe 会被保留。一个单独的物理 ZigBee 无线电能维持最多 540 个应用程序，在端点 0x01-0xf0 所表现出来的应用的决定是如何建立设备端点配置应用程序和端点广播信息。期望设备描述枚举在终端里的使用或者为其他辅助的绑定设备提供设备能力的额外描述[28]。

ZigBee设备能被建立并表示为一个带有为标准而写的单独的端点应用程序，公开的ZigBee profile标识符“XX”。如果生产商想配置一个ZigBee设备支持的标准profile“XX”，且提供给开发者特殊的扩展名，那么这些扩展名就可能会被广播在一个孤立的端点。维持标准的profile标识符“XX”，但生产时没有开发者扩展名的设备将仅仅广播维持单独的profile标识符“XX”，且不能使用卖主扩展名响应或者建立消息[29]。

在先前的例子中，我们介绍了如何使用通用标准建立一个设备，这个标准公布ZigBee的profile标识符“XX”,它包含了标准的profile的初始版本。如果ZigBee

联盟会通过更新这个标准profile来建立新的特性和加法，修订本将组合成一个新的标准profile，这个新的标准profile有一个新的profile标识符（即“XY”）[30]。有profile标识符“XX”的设备应域新设备兼容，这新的设备对于profile标识符“XX”和profile标识符“XY”有新设备advertised维持。以这种方式，新设备使用profile标识符“XX”与旧设备通信，然而，也可以使用profile标识符“XY”与旧设备通信在相同的应用程序里。在ZigBee中的服务发现特性激活网络中的设备来确定维持级别[31]。

zigbee联盟在协议栈之外又增加了一部分操作cluster的函数，那就是zigbee cluster library,(ZCL)，这里边已经以源代码的形式提供了操作联盟规范的那些public profile下的函数，其主要功能包括一些command的response,indicate以及confirm等，还有读写attribute的一些操作函数。所以在理解了ZCL的工作机制基础上，通过调用ZCL的函数实际上会让应用程序设计变得简单。假设我们要控制一个LED，有一个远程节点（发命令控制led），一个本地节点（接受命令并真正的让led 亮起来)，那么如果引入ZCL的概念，你可以设置这个操作led 的事情是一个cluster，其下包含三个命令，一个是open命令，一个是close命令[32]，一个是read attribute命令，灯还有一个attribute，那就是当前的status，远程节点可以用ZCL的函数发open和close命令，也可以随时发一个read attibute命令读取本地节点led 的状态。这么做的好处是不需要再自己设计一个规定，而是直接调用ZCL即可实现，这对于command和attribute数量很少的应用不见得有多大好处，但是当command和attribute数量很多的时候，引入ZCL会让事情变得简单。

2.4.6 设备和服务发现

此功能将支持在一个单独的PAN中发现的设备和服务。ZigBee协调器，ZigBee路由器和ZigBee终端设备的类型，此功能将作如下处理：

a) 在每一使用休眠的ZigBee终端设备、ZigBee路由器（或ZigBee协调器）的网络，必须被设计作为如它们的节点描述符描述的Primary Discovery Cache Devices。这些Primary Cache Devices 是它们自己可发现的，且提供服务器服务来上载和存储代表休眠的ZigBee终端设备的发现信息。另外Primary Cache Devices响应代表休眠Zigbee]终端设备的发现请求。每一个Primary Discovery Cache Device是ZigBee路由器或者ZigBee协调器。

b) 对于被Config_Node_Power,设备和服务发现指示想要休眠的ZigBee终端设

备将管理被ZigBee终端设备选择的Primary Discovery Cache设备上的网络地址、IEEE地址、活动节点、简单描述符、节点描述符和电源描述符的上载和存储来允许在这些休眠设备上的设备和服务发现操作[33]。

c) 对于被设计作为Primary Discovery Cache Device的ZigBee协调器和ZigBee路由器，这个功能将代表休眠ZigBee终端设备响应发现请求，这些终端设备已经注册和上载了它们的发现信息[34]。

d) 对于所有的ZigBee设备、设备和服务发现将支持设备和从其他设备过过来的服务发现请求，且允许从其他本地的应用对象过来的请求的产生。注意设备和服务发现服务是由Primary Discovery Cache设备代表其他ZigBee终端设备提供的。万一Primary Discovery Cache Device是请求的目标，那么NWKAddrOfInterest或者Interest域的设备将被请求和/或响应填满来区分从设备来的请求的目标，这个设备是发现的目标[35]。

第三章 无线抄表系统的整体设计方案

3.1 无线抄表系统功率计量方案

3.1.1 设计方案比选

通过前期调研，以及资料收集，我们总结了以下三种方案，并对以下三种方案的设计进行了比较：

a) TI 方案 MSP430FE427 加 ZigBee 无线模块；

b) 专用计量芯片 STPM01 加 CC2530；

c) CC2530 自身的 12 位 ADC 测量方案。

这三种方案的简单比较如表 4.1：

表 3-1 几种方案优缺点的比较

方案	描述	优点	缺点
MSP430FE427+专用无线发射模块	MSP430FE427 集成了 MSP430 单片机和 ESP430 电量计量专用模块。	1.测量精度高；2.因为集成了 MSP430,所以功能强大（如LCD显示等);3.ESP430 集成了信号处理和计算功能；4.编程较简单，可直接读取测量数据。	价格贵。
STPM01+CC2530	STPM01 是电量计量专用芯片，可以将所测数据传送给外部 CC2530。	1.测量精度高；2.应用简单；3.集成了信号处理和计算功能；4.编程简单，外部 MCU 可直接读取测量数据。	价格较贵。
CC2530	直接用 CC2530 自带的 12 位 ADC 来测量数据。	价格便宜。	1.精度较低；2.应用较复杂；3.需要较复杂的程序来处理信号和计算测量参数；

根据上表比较，考虑以合理的价格取得较高的计量精确度，最终采用了 CC2530 加 STPM01 的设计方案[36]。

3.1.2 设计方案的可行性

整个方案对我们来说，首先是为了避免传统布线的麻烦，所以选用了 ZigBee 技术，而选用 CC2530 主要考虑 TI 在技术支持方面的优势，因此，我们可以在网络很容易的收集到相关的资料，很多国内的制作 ZigBee 开发板的公司都采用了 TI 的方案，同时，TI 的有专门的协议栈 Z-stack，方面实际应用[37]。

在本设计方案中，我们假设一个社区有 20 栋楼，每栋楼 9 层，每层楼有 3 个住户，总共有 540 户，每户有一个智能电表和一个无线抄表的 ZigBee 节点，共需 540 套前端采集节点，正如上文中提到的 ZigBee 网络可容纳 65，000 个设备，那么 540 个设备在这个网络可以同时启用，由于抄表系统工作数据量少、数据突发的间隔时间通常以小时计，所以在整个网络有数百个节点情况下，也不易发生通信拥塞情况。同时 ZigBee 具有自组网功能，自动调频技术，在这样的网络中，可以减少和其他工作在 2.4GHz 无线技术的干扰。

STPM01 是意法半导体公司的产品，内部集成了相应的算法，在应用中我们主要根据他们公司提供的一些典型应用电路做相应的修改就可以完成我们对应的硬件设计。再根据 STPM01 特定的 SPI 时序来和 CC2530 通信。实现起来简单易行。

3.2 系统整体方案的设计

电表设计用于 220V/50Hz 电网，能测量的基本电流为 10A，最大电流 30A。电压由电阻分压器进行采样，而电流由电流互感器和锰铜分流器采样，采样信号送给测量芯片 STPM01，STPM01 通过对信号进行模数转换后转换为能够被 STPM01 的 DSP 处理数字信号，内部 DSP 经过相位校正，能够计算出测量点的功率、电流、电压等数据，STPM01 能够自动选择电流较大的通道计算功率。这些数据最后通过 SPI 接口传送给 CC2530。该系统的整体设计方案如图 3-1 所示。根据 ZigBee2007 的协议栈，ZigBee 支持 5 级的深度，也就是 6 个层级，协调器为第一级，下一级可以是路由器或者终端，再下一级也可以是路由器或者终端，但是只有协调器或者路由器可以有子节点，终端没有子节点，路由器还有父节点。从图中我们可以看到，上位机负责数据的处理，通过 COM 口和通信模块进行通信，通信模块位于 ZigBee

的网络中，是 ZigBee 的协调器，控制着网络中的重要数据，包括密钥等等，它负责启动和组建网络。通信模块下面可以有 6 个具有路由功能的 ZigBee 设备子设备，或者说是 20 个 ZigBee 设备（包括 14 个终端设备和 6 个路由设备）。在第二层级的每个路由模块（测量节点）下面也可以连接 6 个路由设备（测量节点）或者 14 个终端设备（测量节点）。后面的第三层级、第四层级、第五层级、第六层级的设备都可以是路由模块或者终端设备。这些设备是否具有测量功率的功能，需要根据实际情况来安装。这些设备如果用于测量用户的用电数据的同时，还需要进行 j 中继功能的，我们需要在此用户的家中安装具有路由功能的测量节点，如果安装在用户家中单纯用于测量用电数据的，那么我们只需要安装一个终端测量节点。由于 ZigBee 工作在 2.4GHz 的频段，波长比较短，传输距离和穿透能力都比较弱，所以在某些节点距离比较远的时候需要加入一些只有路由功能的 ZigBee 模块。整个网络开始运行后是一个树行结构。ZigBee 网络中的每个设备会在运行后根据网络的运行情况，一定的路由算法，在将用电数据传输回通信模块的时候会自动选择最佳的路径。也就是在某一时刻 A 设备将用户的用电数据传回通信模块，需要经过具有路由功能的 B、C、D 才传回通信模块，但是假如网络中有些数据发生了改变，A 设备的数据传回通信模块可能只经过了 B、C、F。ZigBee 数据交换的时候通过 ZigBee 信道双向传输。测量节点就是由 ZigBee 芯片 CC2530 和 STPM01 功率测量模块组成。通信模块和只具有路由模块的 ZigBee 设备主要是 CC2530 芯片。而我们只需要在应用端口的设定时，对相应的端口赋予不同的定义，则可以实现通信功能和路由功能的变换。

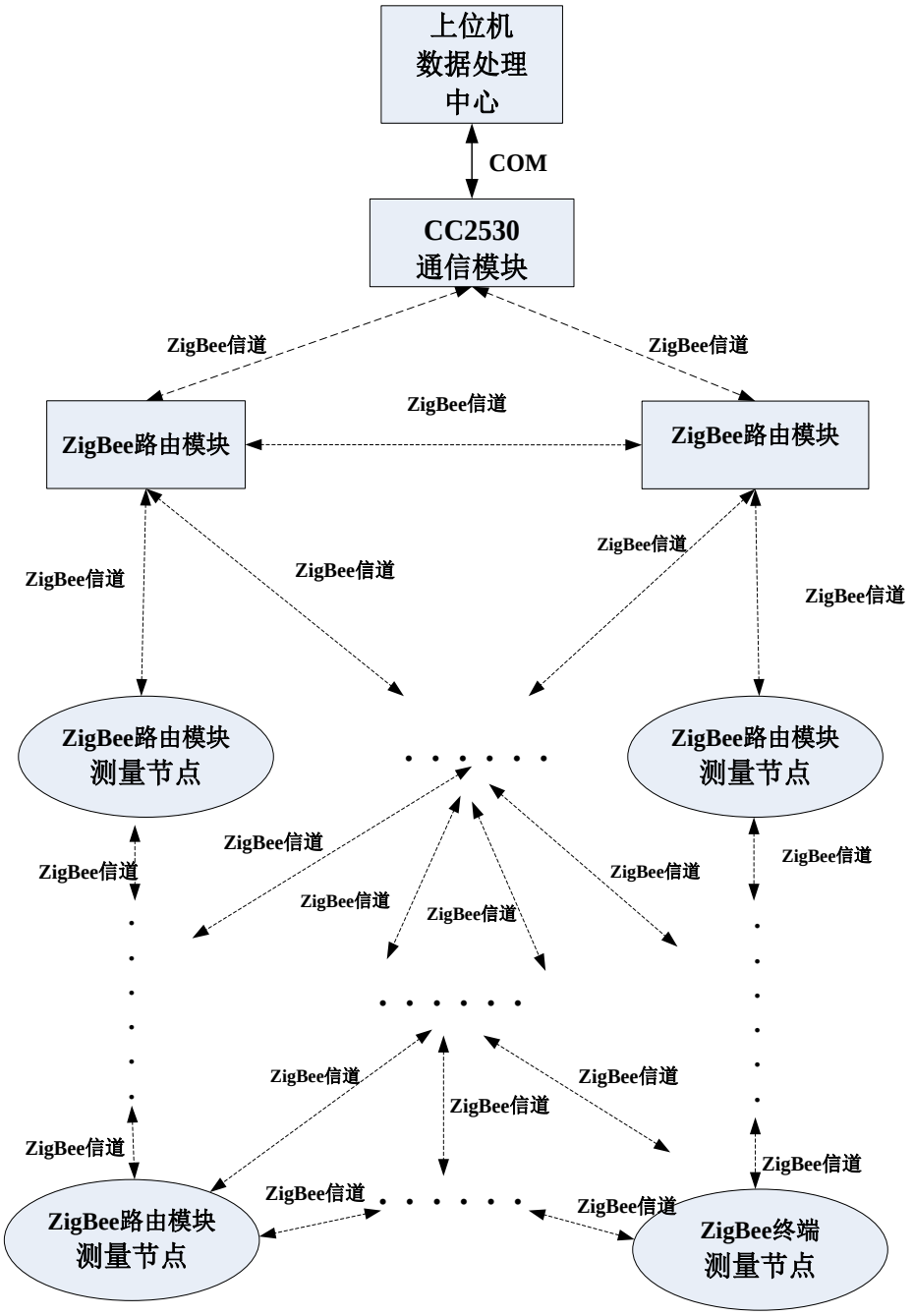


图 3-1 系统总体设计方案

3.3 系统功能描述

由图 3-1 可知，该自动抄表系统分为三大部分：上位机显示和控制、通信模块、以及电网功率数据测量节点。上位机主要负责实时监测节点的工作状态，当

测量节点工作不正常时显示报警信息给用户，同时显示相应测量点的实时功率。通信模块主要负责和上位机通信，根据上位机发来的命令，经过处理后发送命令给相应的测量节点，同时接收测量节点传回的数据，并经过初步处理后传给上位机处理。测量节点的主要工作是测量相应用户的用电数据，并在接收到相应的命令后传输数据给通信模块。

为了使整个系统能正常工作，提高 ZigBee 网络传输数据的速率，我们采用了网络地址的通信方式,其工作流程如图 3-2 所示：

a) 把每个模块的 IEEE 地址上位机的相应数据库中，在整个网络开始工作后，上位机调用数据库中的数据,获取包括通信模块在内所有 CC2530 模块的网络地址，把 IEEE 地址和网络地址的对应关系更新到上位机的数据库中；若是某个节点没有加入网络，则不能获取其节点的网络地址，上位机就报警；并在后面的工作中重新尝试获取未加入网络的节点，如果其网络地址获取成功，则取消对应节点的未获取网络地址的报警。

b) 上位机发送第一个测量节点的网络地址给通信模块，通信模块根据网络地址发送命令给网络地址所对应测量节点的 CC2530 芯片。

c) 每个节点加入网络后，CC2530 就会开始初始化用户的用电数据，每隔一段时间就会根据相应的 SPI 时序来读取 STPM01 芯片所测量到的用电数据，当接收到通信模块发送过来的请求返回该节点用电数据后，便可以立即将数据发送回通信模块；

d) 通信模块再通过相应的协议把此数据发送给上位机，上位机经过处理和存储后显示在显示器上。

e) 上位机处理完第一个测量节点后，就开始根据一定的方式获取第二个节点的数据了。如此反复的操作不断获得这个小区内所有测量节点的数据。如果上位机连续发送五次都没有没有收集到相应测量节点的数据的话，那么显示屏上会显示出相应的报警信号并告知用户修理更换。当相应的模块被更换好后，管理中心就需要向上位机的数据库传输新的CC2530模块的64位IEEE地址并进行更新保存，并通过 ZDP_NwkAddrReq()命令重新获取新模块的网络地址，这样相应不正常工作的节点又可以恢复正常的工作。

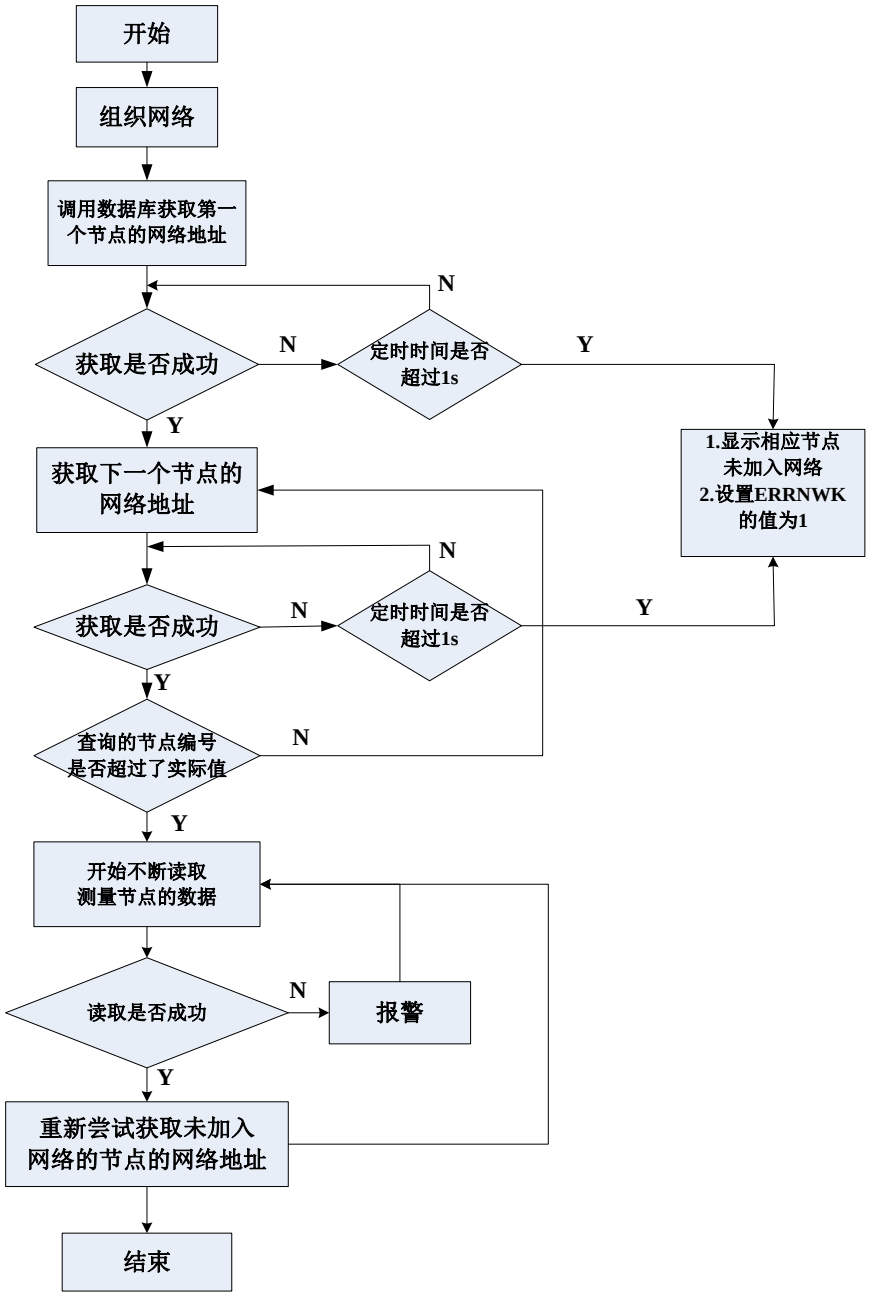


图 3-2 系统工作流程图

之所以采用上位机为主动的方式，主要是为了避免多个测量节点同时发送数据给通信节点信息的阻塞或丢失。测量节点如果距离通信节点较远，返回给通信节点的数据会经过一些具有路由功能的测量节点的中继。如果有些测量节点不需要进行路由功能，那么此处的测量节点只使用了一些具备终端功能的测量节点。图 3-3 是设计方案布局的一个缩影。ZigBee 网络代表了很多的测量节点、通信模块、和只有路由功能的点。从图中我们可以看出，测量点 B 和测量点 C 具体 ZigBee

网络比较远，不能加入此网络，所以需要在他们距离合适的点上安装一个只有路由功能的 ZigBee 模块。测量点 A 是通过 B 节点才能把数据发送到网络中的其他节点，就是他的数据需要经过中继才能发给通信模块，C 节点的用电数据也是要经过中继才能将数据发送给通信模块。所以对于测量点 A、测量点 B、测量点 C 根据功能分析，他们分别是终端、路由、终端的 zigBee 模块。

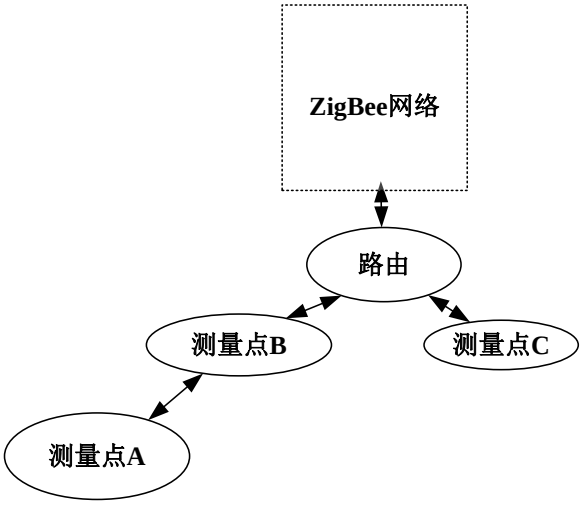


图 3-3 测试示意图

3.4 本章小结

本章首先对几个适合我们设计的方案进行对比，从测量精度、实现性和价格上等综合考虑之下采用了 CC2530 和 STPM01 的方案设计。接下来介绍本方案的整体规格，能够测量的电网的参数，并结合第二章 ZigBee 协议谈论了系统的整体架构以及工作流程。

第四章 硬件电路的设计

4.1 关键芯片的简介

4.1.1 CC2530 的简介

CC2530 是德州仪器公司推出主要用于 Zigbee 应用的一款芯片。它不仅成本非常低,而且能建立强大的网络节点。CC2530 包含了的 RF 收发器、增强型 8051CPU，8KBRAM，可编程闪存和多种其他强大的功能。CC2530 根据闪存的大小划分为四种不同的版本：CC2530F256/128/64/32,分别具有 256/128/64/32KB 的闪存。CC2530 拥有不同的工作模式，使他也满足超低功耗系统的要求。工作模式之间的转换时间短更确保了低能源消耗[37]。

模块框图如图 4-1 所示，模块大致分为三种类型：CPU 和内存相关的模块；时钟、外设和电源管理相关的模块以及无线电相关的模块。CPU 内核是一个单周期的 8051 兼容内核。它有三个不同的存储器访问总线，一单周期访问 SFR、DATA 和主 SRAM。它还包括一个调试接口和一个 18 输入的扩展中断。中断控制器提供了 18 个中断源，分为六个中断组，每组与四个中断优先级相关。当设备从空闲模式回到活动模式，也会发出一个中断服务请求。一些中断还可以从睡眠模式唤醒设备。内存仲裁器位于系统中心，因为它通过 SFR 总线，把 CPU 和 DMA 控制器和物理存储器和所有外设连接在一起。内存仲裁器有四个存取访问点，访问每一个可以映射到三个物理存储器之一；8-KB SRAM 映射到 DATA 存储空间和 XDATA 存储空间的一部分。8-KB SRAM 是一个超低功耗的 SRAM，当数字部分掉电时能否保留自建的内容，这对于超低功耗应用时一个很要的功能。闪存块为设备提供了内电路可编程的非易失性程序存储器，映射到 CODE 和 XDATA 存储空间。除了保存程序代码和常量，非易失性程序存储器允许应用程序保存必须保留的数据，这样在设备重新启动之后可以使用这些数据。数据内核和外设由一个 1.8V 低差稳压器供电。另外 CC2530 包括一个电源管理功能，可以实现使用不同供电模式的长电池寿命的低功耗应用运行。CC2530 包括了许多不同的外设，如 IO 控制器，定时器，SPI 功能模块等，允许应用程序开发者开发先进的应用。

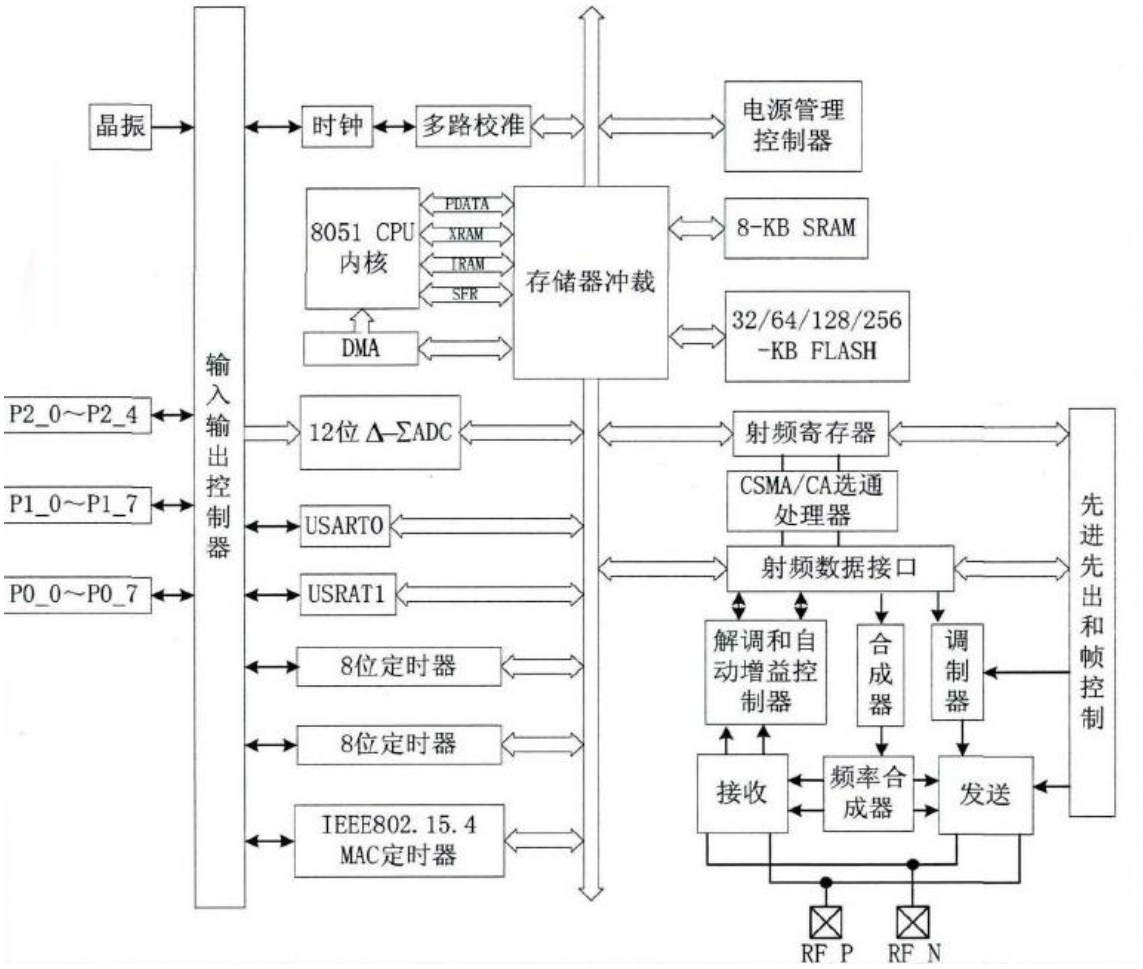


图 4-1 CC2530 结构框图

CC2530 设备内置了一个 IEEE 802.15.4 兼容无线收发器，这就是 RF 内核控制模拟无线模块。同时，CC2530 还为 MCU 和无线设备之间提供了一个接口，这样就可以实现确定发出命令、读取状态、自动操作和确定无线设备时间的顺序。无线设备还包括了一个数据包过滤和地址识别模块。

CC2530 中的 SPI 模式分为 4 线接口或 3 线接口与外部系统通信。接口包含引脚 MOSI、MISO、SCK 和 SS_N。当寄存器写入 UxBUF 后，SPI 主模式就开始传送字节，SCK 串行时钟使用波特率发生器产生，发送寄存器提供将字节输出到 MOSI 引脚。同时接收寄存器从输入引脚 MISO 收取字节。当传送开始 UxCSR_ACTIVE 位变高，而传送结束后，UxCSR_ACTIVE 位变低。当传送结束时，UxCSR_TX_BYTE 位设置为 1。SCK 的极性是由 UxCSR_CPOL 位选择，它的相位由 UxGCR_CPHA 位选择。字节传送的顺序由 UxGCR_ORDER 位选择。传送结束时，收到的数据字节由 UxBUF 提供读取。在单元就绪接受另一个自己用来发送时，产生发送中断。因为 UxBUF 是

双缓冲，这个操作正好在发送开始时就发送了。对于 DMA 传输这是自动处理的。还要注意发送中断和接收中断的区别，因为发送中断会提前大约 8 位周期到达接收中断[17]。

4.1.2 STPM01 简介

STPM01 是意法半导体公司的一款高精度高和可靠性专门用于测量用电数据的专用 IC，它包含两个电流通道和一个电压通道，可以测量无功功率、有功功率、视在功率、电网频率、电流和电压等。它的电流传感器可以选用多种电路设计方式，一般可以用电流互感器、锰铜分流器或者 ROGOWSKI 线圈。STPM01 主要模拟和数字两部分组成，包括、A/D 转换器、前值放大器、、调压器、带隙电压基准、DSP 和 SPI 接口等。模拟部分总的输出送给 DSP 单元，DSP 处理和计算出消耗的电能、参考的和实际的电压和电流值。DSP 计算出的结果转换为了数字量，通过 SPI 接口和单片机通信。STPM01 提供基准电压和电流，由于电压是动态变化的，这个存储的基准电压值是 11 位的分辨率，而电流时 16 位的分辨率[38]。

STPM01 配置了用于存储校表数据信息的 56 位 OTP 存贮器。STPM01 的调压器可以向内部模拟和数字电路供电，同时也能向外部提供两种电压电源[38]。

STPM01 有一个电压输入通道和两个的电流输入通道，电压通道的增益为 4，电压通道的输入误差为± 0.3 V.。电流通道提供一个单输入复用方式的前置放大器。这个前置放大器输出连接到一个可编程的放大器，放大倍数可以选择为 2、4、6、8。所以电流通道总的放大倍数为 8、16、24、32。增益的部分被写到增益寄存器，同时电流的两个通道的放大倍数可以不相同[39]。

STPM01 内含 2 个有功功率，即 0 类和 1 类有功功率。0 类基波有功功率，1 类有功功率寄存器是含有谐波的总有功功率。因此利用这两个寄存器可以计算出谐波功率含量。除测量和配置寄存器外片内还设有状态寄存器和模式寄存器两类[39]。

STPM01 可以作为单片机外围设备，也可以独立工作。当 STPM01 独立工作时，STPM01 芯片能够直接驱动一些外围设备,同时 STPM01 的 LED 管脚输出相应的脉冲。在此模式下,SCL/NLC 输出有无负载指示信号，SDA/TD 输出窃电指示信号，SYN 输出电能是否反向指示信号[39]。

在作为单片机外围设备模式时，单片机作为主机，STPM01 作为从机。两者可以通过 SPI 接口通信，单片机能够通过 SPI 通信方式读取 STPM01 的内部的测量数据、配置和状态信息，它还能够更改 STPM01 的配置位，实时改变 STPM01 的工作

模式。修改 APL 寄存器的值，可以对包括 MOP、MON 等管脚的作用进行修改[39]。其机构框图如图 4-2 所示。

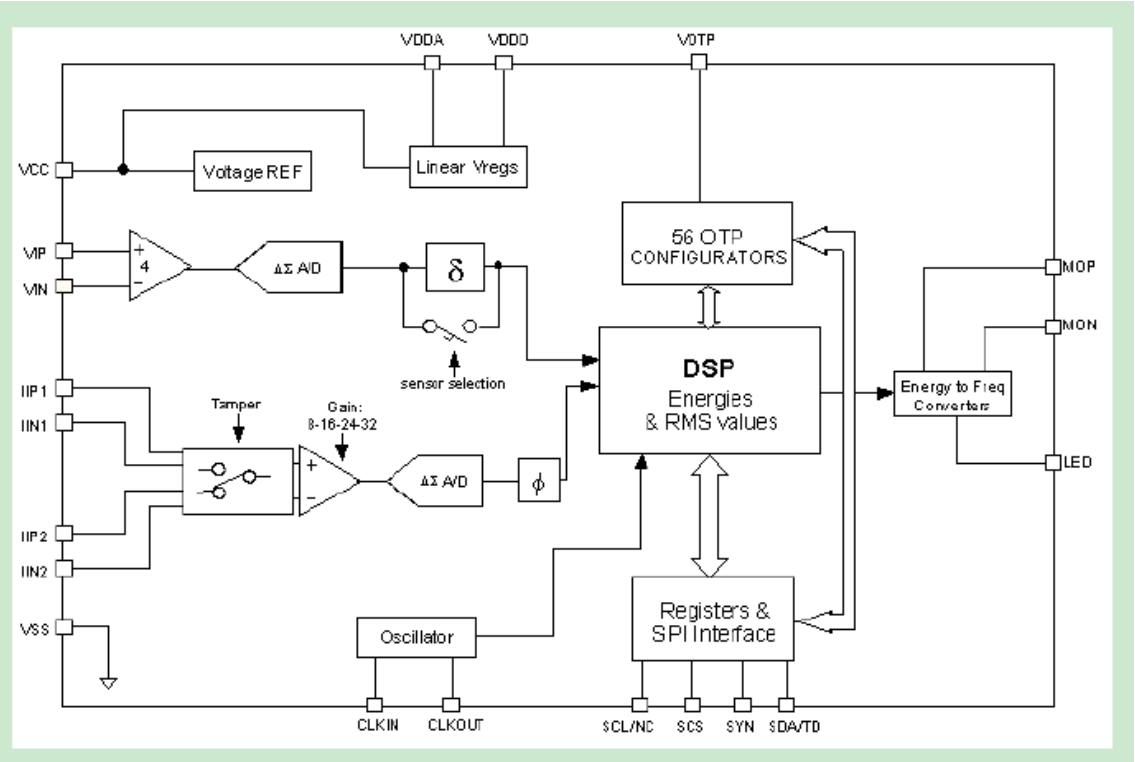


图 4-2 STPM01 结构框图

4.2 系统的设计

CC2530 与 STPM01 通信方式采用 SPI 的通信方式，在 CC2530 我们不采用中断的方式，而采用了轮询的方式，一个方面是为了减少 CPU 的资源，另一方面我们采用轮询的方式已经满足我们的设计要求。

4.2.1 STPM01 与 CC2530 通信接口的设计

下图所示为 STMP01 与 CC2530 通信接口连接示意图：

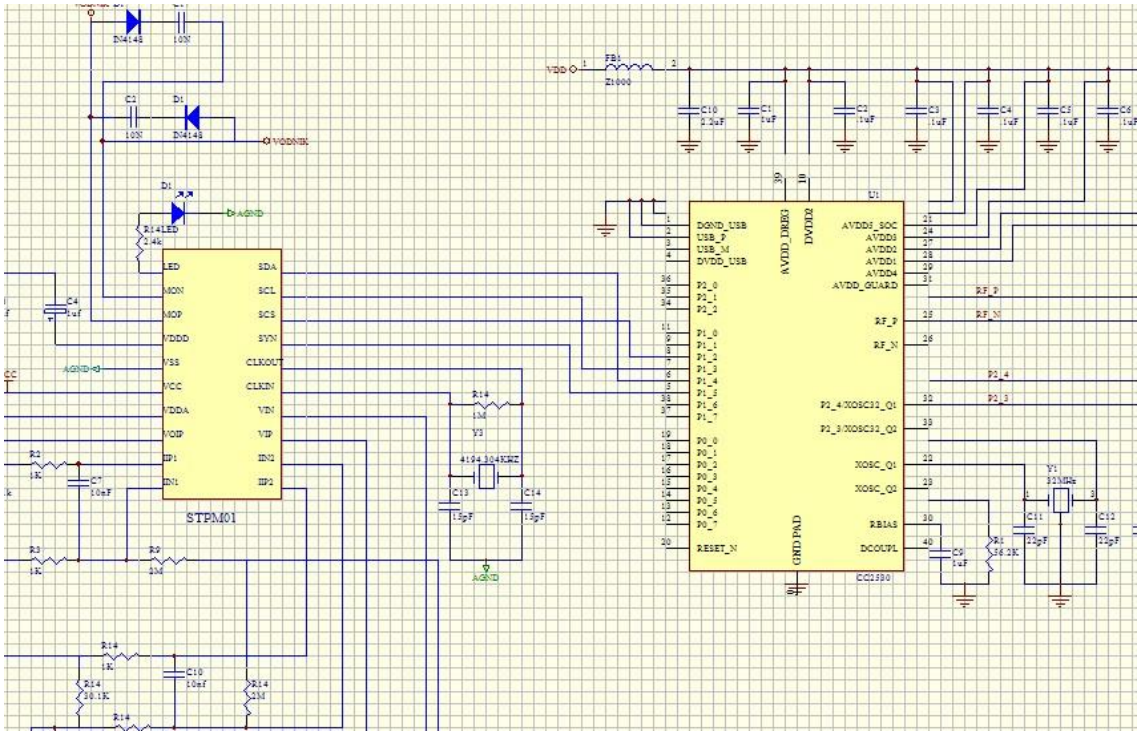


图 4-3 CC2530 与 STPM01 的连接图

STPM01 与外部通信主要通过两线制的 SPI 方式，它可以让 CC2530 执行下面的操作：

1、给 STPM01 发送远程复位请求

2、读取数据中高 8 个四字节的数据记录

3、临时设置或者清除系统的信号或者永久的设置 OTP，

其中，SPI 通讯使用了 STPM01 中的下面四个接口

1、SCS：当设置它为低电平的时候，它使能一些 SPI 操作，

2、SYS：它控制 SDATD 输出

3、SCLNLC：SPI 通信的同步时钟信号输入管脚，

4、DATD 在 SYN 为高电平的时候，它为空闲的状态，否则，它使能串口数据输入。

SPI 的主控器为 CC2530，这意味着 SCS 和 SYN 通过一些通用 IO 口驱动，CC2530 到 STPM01 连接通过下面的方式连接

MISO 和 SDATD 连接

SCK 和 SCLNLC 连接

SS 和 SCS 连接

通用 IO（此设计中采用了 P0_1 脚）管脚连接到 SYN

4.2.2 CC2530 对应电路的设计

由于 CC2530 只是起到了和 STPM01 通信以及无线传输信号的作用，所以 CC2530 射频天线的设计至关重要。根据相应的资料，天线我们采用倒 F 式的。射频天线主要设置的几个参数是阻抗、Vswk、带宽、RL 参数，TI 给出了典型的值，阻抗我们一般取 50 欧姆，Vswk 小于等于 1.5，带宽大于 100M，RL 小于负 100DB。我们需要根具这些设置我们的天线参数。由于 Zigbee 的终端、路由器、协调器的主要是软件部分的不同，所以这三种功能的模块我们都采用了相同的电路设计，再根据实际的需要，通过下载相应的应用软件。CC2530 典型的部分外围应用电路如图 4-4 所示，从图中我们看出 CC2530 只需要极少的外部元件。

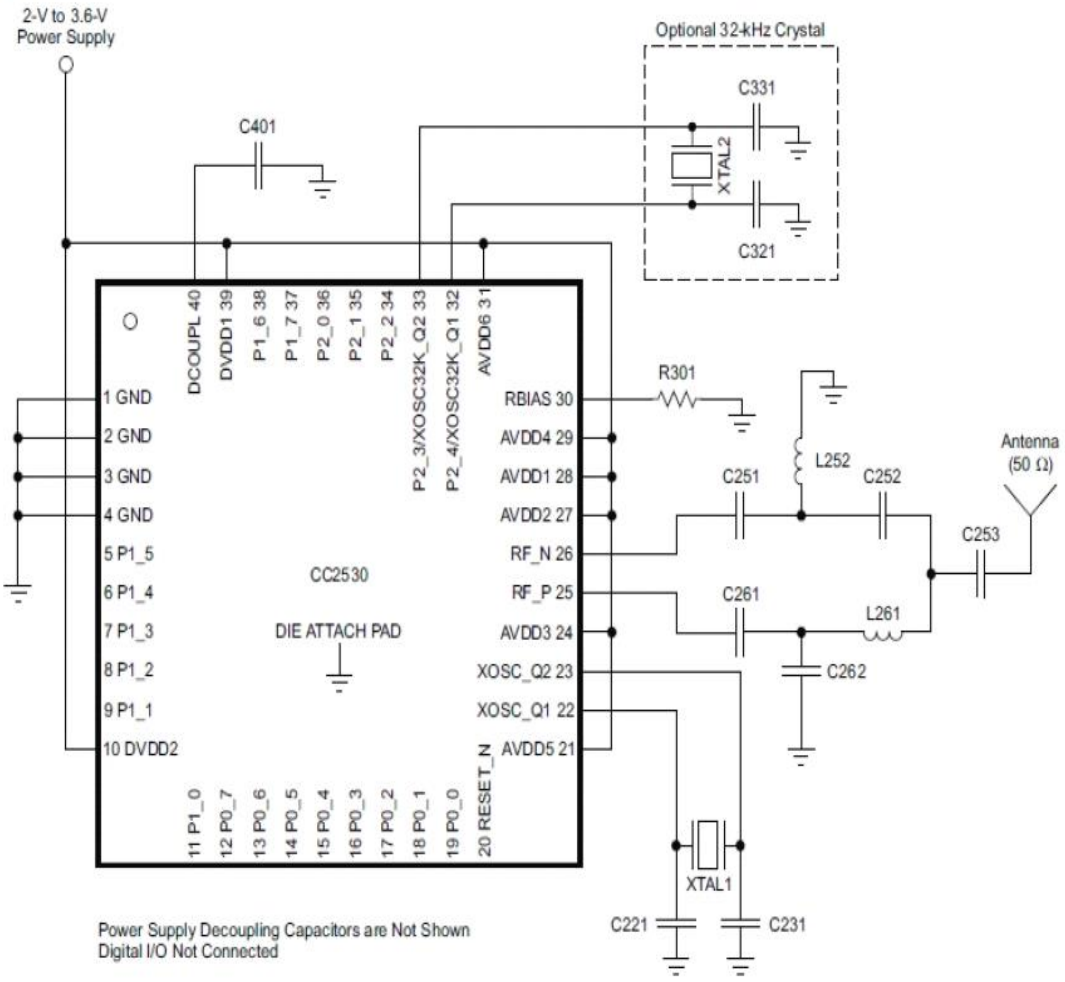


图 4-4 CC2530 部分外围应用电路

4.2.3 电源电路的设计

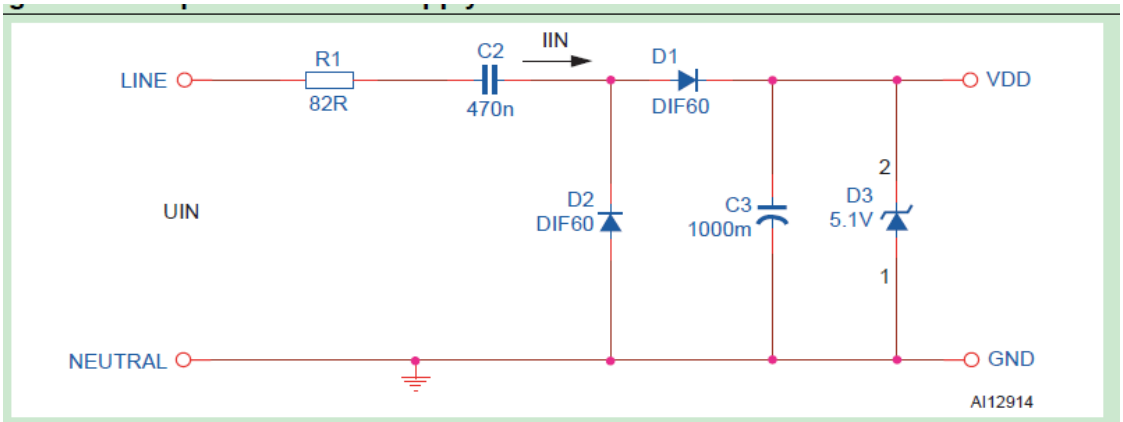


图 4-5 电源电路设计

由于直接测量的是用电情况，而 CC2530 和 STPM01 的以及外围对应的一些这两电路所需使用的电量相对用户的用电量可以忽略不计，因此我们采用电网的电压供电。虽然一般的设计方式是需要一个变压器和一个整流电路，同时我们也会用到开关电源的方案，但是，这两个方案不仅贵还需要占用大量的 PCB 空间，所以我们提供了一个便宜而且实用的方案。其原理图如上图 4-5 所示，这个输入电流 IIN 通过电阻 R1 和电容 D2 的容抗进行限制，其相关的表达式为

$$IIN = \frac{VIN(RMS)}{Xc2 + R1} \tag{4-1}$$

Xc2 为电容 C2 的容抗。R1 可以限制电流的作用，但是它也消耗能量。通过增加一个便宜的半波整流器，在正本周的时候电流允许输出能量。$V_{IN(RMS)}$ 为半波 AC 波形电压的有效值，他的表达式如下所示：

$$V_{IN(RMS)} = \frac{1}{2} \times \frac{V_{PEAK} - V_Z}{\sqrt{2}} \tag{4-2}$$

V_{PEAK} 为峰值电压

V_Z 为电压通过 D1 和 D3 后的降压值

Xc2 为电容的阻抗，他的表达式

$$X_{C2}=\frac{1}{2\pi fC_2} \tag{4-3}$$

通过等量代换，可以得到下面的表达式

$$I_{IN}=\frac{V_{PEAK}-V_Z}{2\sqrt{2}(X_{C2}+R_1)}=\frac{\sqrt{2}V_{mains}-V_Z}{2\sqrt{2}(X_{C2}+R_1)} \tag{4-4}$$

假如每个 2 极管的压降是 0.7V，那么总的压降是

$$V_Z=V_{D1}+V_{D3}=5+0.7\times 2=6.4V \tag{4-5}$$

以上根据是假设我们的参数为：电压 220V，F=50HZ，VZ=6.4V，则计算 I_7 将是 IIN=15.7mA

由于我们真正使用的电压是 3.3V 的，所以还需要加分压电阻和 430 稳压管电路，如遇特别情况，比如用电的电流比较大，则建议重新设计电路。

4.2.4 STPM01 外围电路的设计

下面的框图为 STPM01 外围电路的的框图

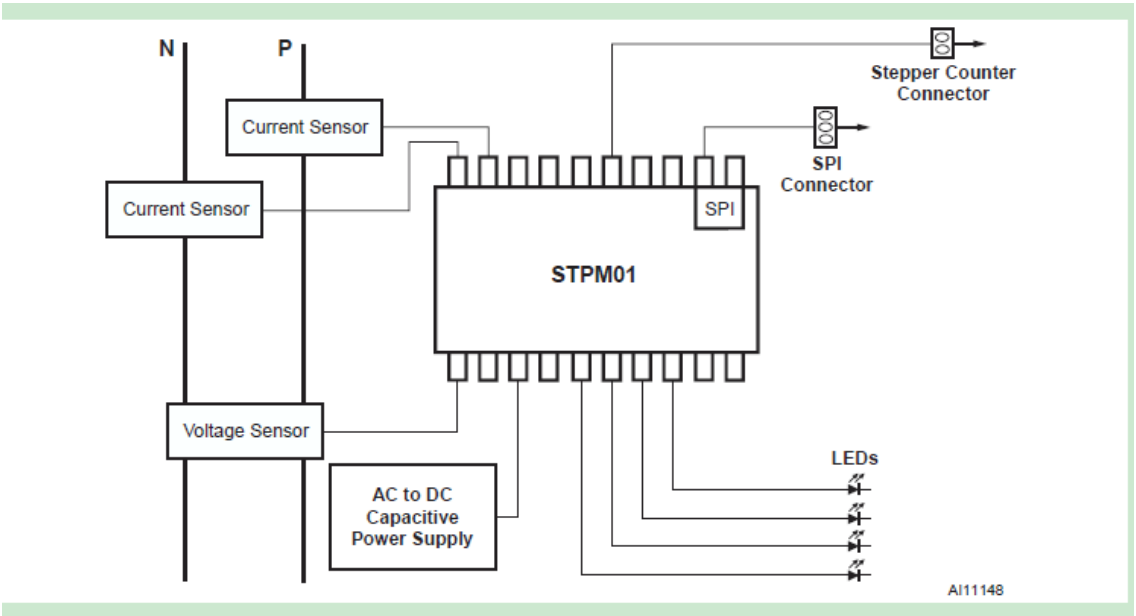


图 4-6 STPM01 外围电路框图

4.2.4.1 电流传感器电路

电流传感器电路包括了两个外部电路传感器电路，包括了他们分别放在了初级通道的和次级通道上。初级通道使用一个电流变压器和主电流联系起来，这个负

载电阻用于处理 Vin1 和 Vip1 的电压，这个低通滤波器用于滤除高频和电压下降时的少量干扰，初级线圈传感器电路如图 4-7 所示

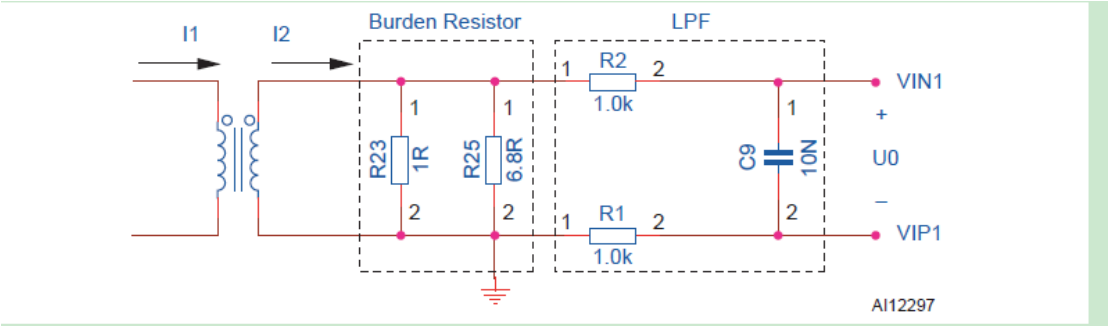


图 4-7 线圈传感器电路图

根据数学了电路理论我们可以得到初级电流的计算如下

等式 1

$$I_2 = \frac{N_1}{N_2} \times I_1 \tag{4-6}$$

等式 2

$$U_0 = U_A = I_2 \times \frac{R_{23} \times R_{25}}{R_{23} + R_{25}} = \frac{N_1}{N_2} \times I_1 \times \frac{R_{23} \times R_{25}}{R_{23} + R_{25}} \tag{4-7}$$

假设主电流的峰值为 I_{1PEAK}，那么初级电流的分值为 I_{2PEAK}

由式 4-6 得

$$\frac{I_{1PEAK}}{I_{2PEAK}} = \frac{N_2}{N_1} = \frac{2000}{1} \tag{4-8}$$

由式 4-8 可得，

$$I_{2PEAK} = \frac{I_{1PEAK}}{2000} = 3mA \tag{4-9}$$

由以上各式可以得出

$$U_{0PEAK} = U_{APEAK} = I_{2PEAK} \times \frac{R_{23} \times R_{25}}{R_{23} + R_{25}} = 2.6mV \tag{4-10}$$

这个输入电压 V_{in1} 和 V_{ip1} 的最大不同由可编程增益放大器来确定，这个应用的目的是使用 8 的倍数作为增益的值，所以 $V_{0PEAK} = 0.15V$

由此可以得出等式 4-11

$$V_{0PEAK} = V_{APEAK} = 0.15V \tag{4-11}$$

所以

$$I_{2PEAK} = U_{APEAK} \times \frac{R_{23} + R_{25}}{R_{23} \times R_{25}} = 172mA \tag{4-12}$$

$$I_{1PEAK} = 2000 \times I_{2PEAK} = 344A \tag{4-13}$$

$$I_{1RMS} = \frac{I_{1PEAK}}{\sqrt{2}} = 243A \tag{4-14}$$

这个初级电流传感器电路连接关键步骤如下：

在这个模式中火线必须连接到管脚 F，一般的，这条线也连接到火线电流丝中，然而，在产生和验证阶段，这些线也需要连接到其他的一些线电压源。

在这个模式中零线电压线必须连接到管脚 N，这条线也连接到了电流先的零线。火线电流线放置必须通过电流传感器 TR1 的孔，同理，零线的电流线也放置必须通过 TR2 的孔。

次级电流传感器使用分流电阻结构，420uW 分流是用于传感器的动态范围的最大值，然而，有一些重要的考虑当我们选择分流电阻的结构应用于功率测量时，分流电阻的功率消耗必须最小，我们的设计中的最大额定电流是 20A，所以的最大功率消耗在分流电阻的值为

$$(20A)^2 \times 420\mu\Omega = 168mW \tag{式 4-15}$$

在分流电阻太高的消耗可能不能满足设计的需求虽然这个分流电阻有铝合金制造，但是当温度达到很高的时候也会出现很明显的误差，这个分流电阻也需要耐的住短路的考验，所以我们需要尽量的减少分流电阻的值。

4.2.4.2 电压传感器电路

STPM01 采用一般的电阻分压作为电压的输入检测，这个 783kΩ电阻被分为了 3 个串联的 261kΩ的电阻，这样能保证瞬时的高电压通过时不会击穿这个电阻，这些三个电阻也减少横跨电阻的可能性，因此减少弧出现的可能性。其对应的电路如图 4-8 所示：

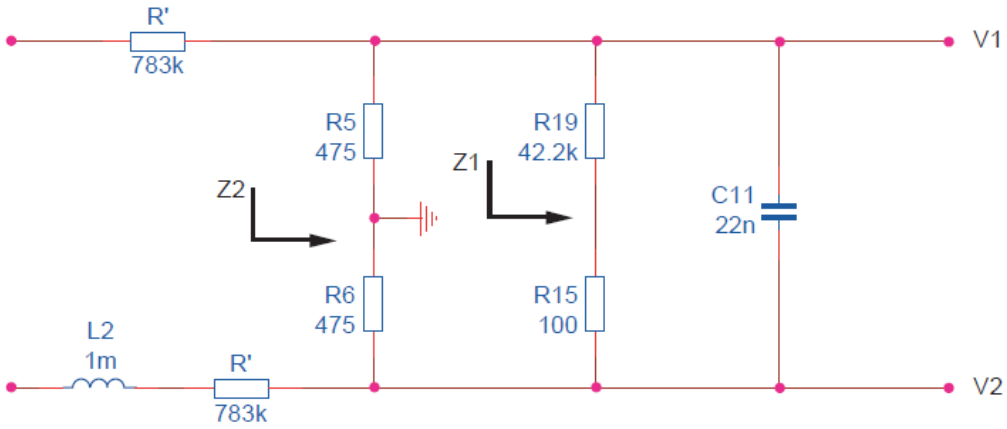


图 4-8 电压传感器电路

4.2.5 串行通信接口的设计

CC2530 串口输出的是 TTL 电平，PC 机输出的是 232 电平，为了能使他们之间能够通信，所以需要在他们之间加入一个转换芯片。我们采用 TI 公司的 SN65C3243。其电路图如 4-9 所示：

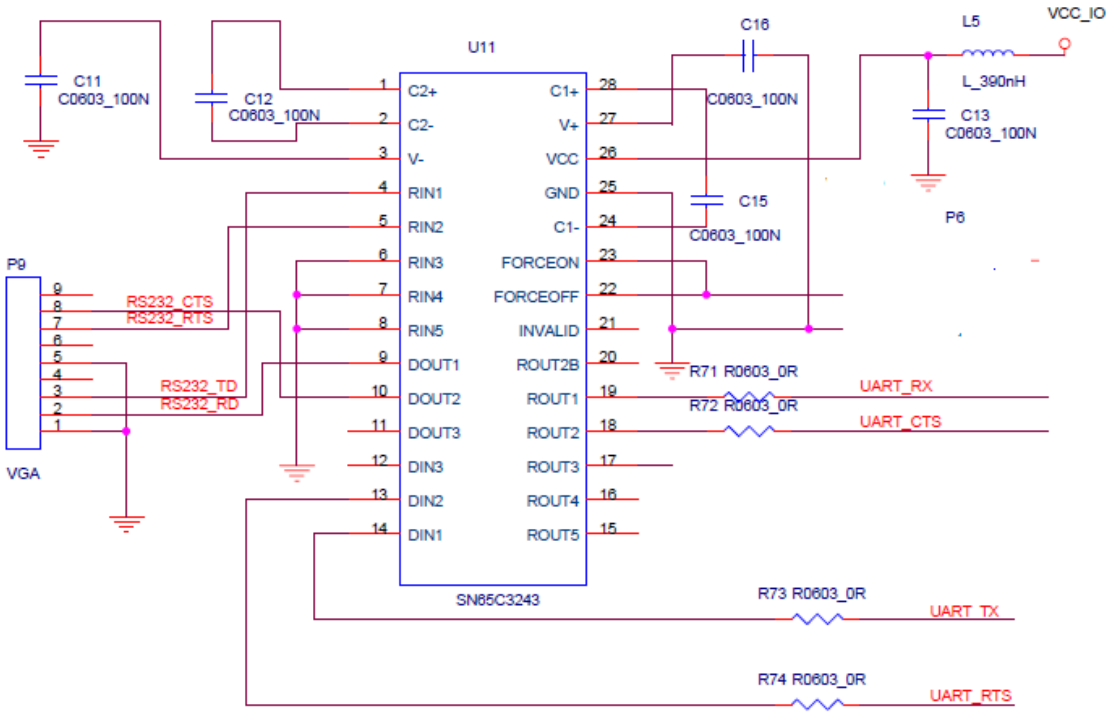


图 4-9 串行通信电路图

4.3 系统的结构框图

4.3.1 通信模块的结构框图

通信模块需要和上位机通信，所以他应该包括电平转换芯片，同时他需要和其他网络通信，所以它需要天线模块，整个通信模块的结构框图如图 4-10

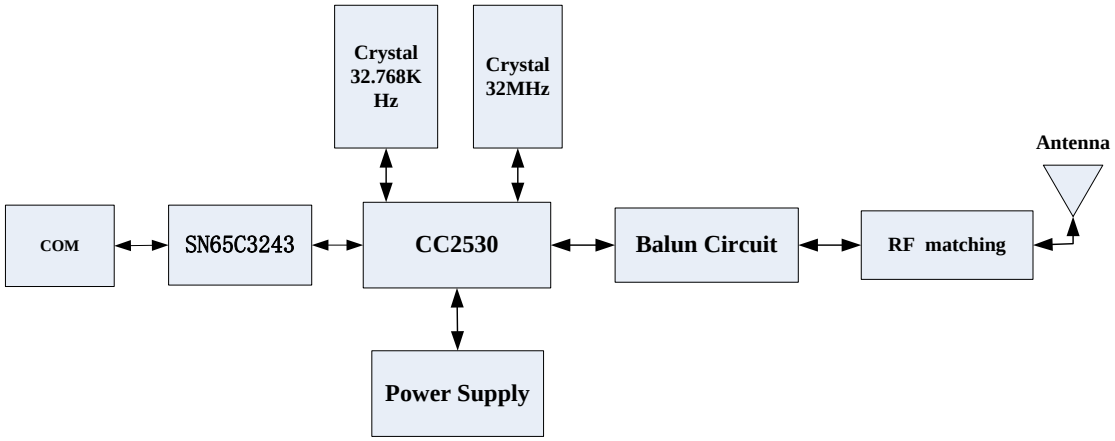
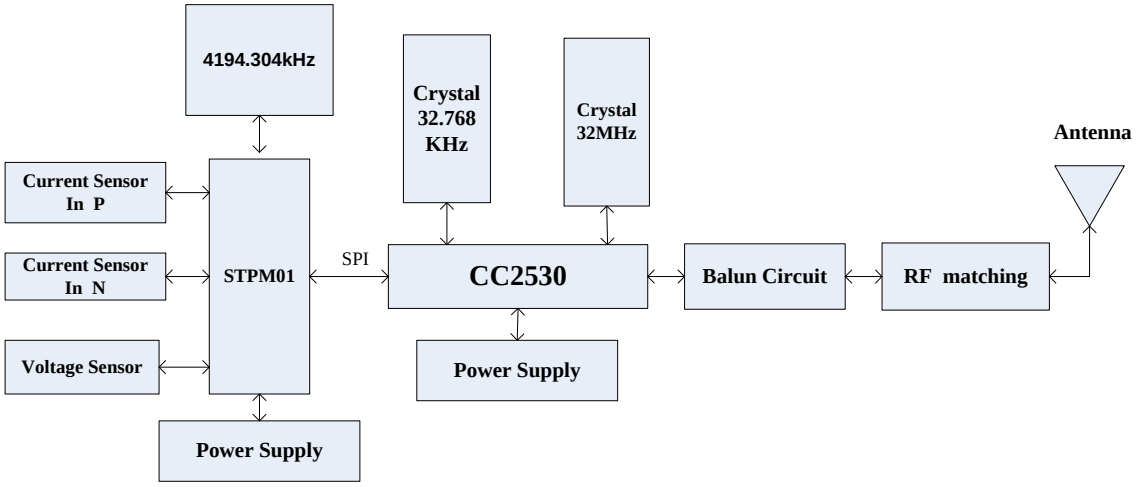


图 4-10 通信模块的结构框图

4.3.2 测量节点的结构框图

测量节点需要读取用户的用电数据，在设计中我们用到了意法半导体的 STPM01 芯片，此芯片业包括了一些外围电路。同时我们采用了 ZigBee 技术，所以也用到了 CC2530 芯片，其硬件结构框图如 4-11 图所示。



4-11 测量节点结构框图

4.4 硬件设计总结

本章开始主要介绍了硬件设计主要用到两款的芯片，包括用于无线发射的CC2530 和测量用电数据的 STPM01。同时使用了 TI 提供的 CC2530 外围设计的经典电路。天线部分在设计中采用的是到 F 型的。由于无线发射的性能的好坏，和 PCB 的厚度，以及阻抗匹配有很大的关系，一般实际中需要一系列的 LC 来调试，使阻抗匹配的参数接近 50 欧姆。对于测试用电数据的地方我们介绍的比较详细。由于测量的是大于 STPM01 的安全电压范围的电压，所以测量的电压和电流（实际测量中也会转换为测量电压）时需要将电压和电流经过一定的衰减送到 STPM01 的引脚上。这些衰减的比例在我们读到的测量数据中需要经过一定的处理。STPM01 的测量数据需要经过 SPI 协议送给 CC2530 芯片，因此也给出了 CC2530 和 STPM01 的连接图。最后作为总结，画出了通信模块（协调器）的结构框图和测量节点的结构框图。

第五章 软件的设计

5.1 Zigbee 软件架构的介绍

TI 的 ZigBee 解决方案作 ZigBee 联盟的长期会员，具有经 Golden Unit 认证 ZigBee 兼容平台，TI 是 ZigBee 解决方案的领先供应商。TI 提供完整的硬件和软件 ZigBee 兼容平台。与其他将其 ZigBee 栈开发外包出去的硬件供应商不同，TI 拥有自己内部专门的软件工程团队，负责最新版本的 ZigBee 堆栈和应用配置文件测试。完整的硬件和软件 Zigbee 兼容平台，经 ZigBee 联盟批准的测试机构认证免费的 IEEE 802.15.4MAC 软件和最高业内水平(Golden Unit status)的 Z-Stack，基于本次设计所使用的是 TI 的 CC2530 芯片，所以我们在 TI 公司为 CC2530 芯片编写好的协议栈下修改设计的程序。

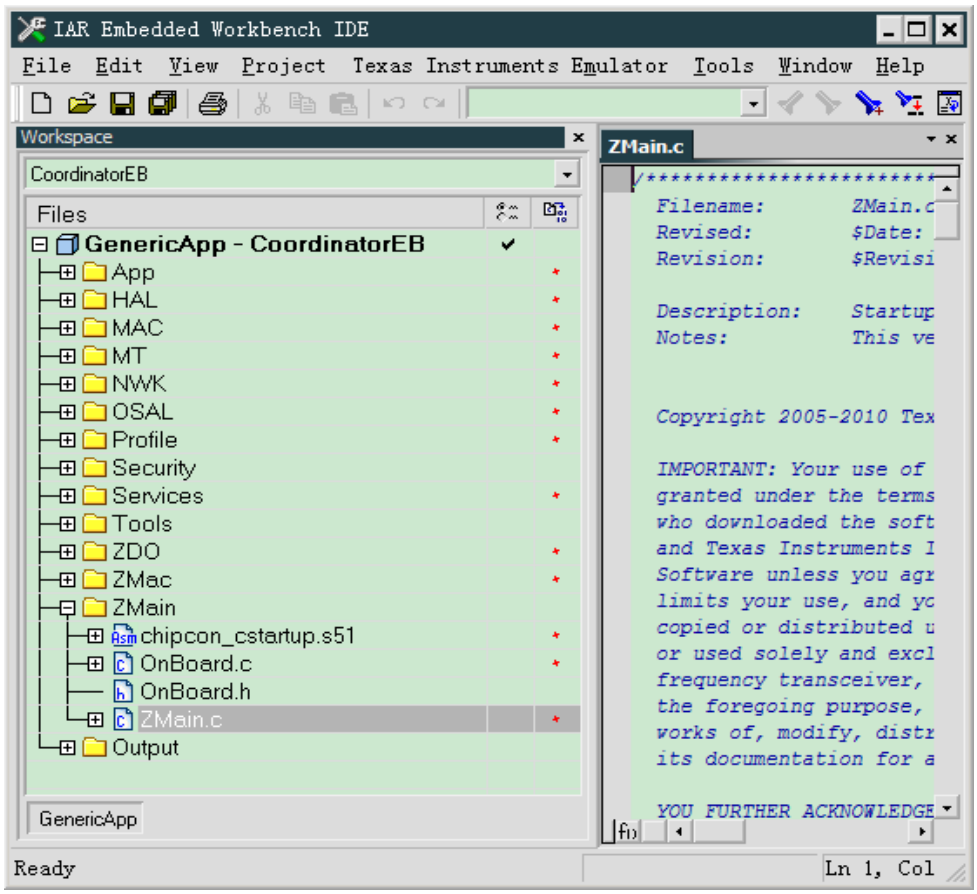


图 5-1 Zigbee 软件架构

本次设计采用了 TI 最新的 ZIgbee 协议栈 ZStack-CC2530-2.5.0，为了提供后续工作的参考，此处介绍一下开发平台各部分的作用。APP 是应用层的目录，这是用户创建各种不同工程的区域，在这个目录中包含了应用层的内容和这个项目的主要内容，在协议栈里面一般是以操作系统的任务实现的。其中，HAL 是硬件层目录，这里面包括着一些与硬件有关的配置和操作函数；MAC 层目录，包含 MAC 层配置参数文件及 MACLIB 库的函数接口文件；MT 目录，包含基于 AF 层的调试函数文件，主要包括串口等通信函数；NWK 网络层目录，包含网络层的配置参数文件及 MAC 的 LIB 库函数接口文件；Profile 目录包含 AF 层处理函数文件；OSAL 目录是协议栈的操作系统程序；Service 地址处理函数目录，包含着地址模式的定义及地址处理函数文档；Security 安全层目录，包含安全层的处理函数；Tools 就是工程配置目录，这里面主要是协议栈栈的配置文档；ZDO 层目录，包括层处理函数文档；ZMain 为主函数目录，包括入口函数及硬件配置文件；ZMac 目录包括 MAC 层参数配置及 MAC 层 LIB 库函数回调处理函数；Output 输出文件目录，这是 EW8051 IDE 自动生成的文件。要将每个功能模块的类型进行选择，可以改变 Workspace 的值。当选择 Workspace 的选项为 CoordinatorEB 时，进行编译和下载，可以使此功能模块的类型是协调器，当选择 Workspace 的选项为 RouterEB 时，进行编译和下载，可以使此功能模块的类型为路由器。同理，当选择 Workspace 的选项为 EndDeviceEB 时，进行编译和下载，可以使此功能模块的类型为终端[41]。

5.2 编译选项

在将 ZigBee 的代码写好后，编译选项的选择也是一个很关键的点，如果编译选项选择不对则有可能达不到我们预计的功能。点击 IAR 的 Project 中的 Option，出现如图 5.2 的界面，我们点击 C/C++ Compiler 选项，看到 defined symbols 上，这里就是我们需要修改的编译选项，一般的编译选项里面包函 NWK_AUTO_POLL、ZTOOL_P1、MT_TASK、MT_SYS_FUNC、MT_ZDO_FUNC，假如我们的应用中用到了 LCD，则使用中需要加入 LCD_SUPPORTED，如果不加入此编译选项，则 LCD 就会不能正常工作，而在正常的使用过程中如果我们的模块没有此功能，将这些编译选项去掉，则能减少 CC2530 的功耗。对于我们使用的上面几个编译选项我们现在做一个简单的介绍。ZTOOL_P1 是说明我们使用串口通信使用的 P1 通道，MT_TASK 这个编译选项使得设备可以与 Z-Tool PC 应用程序通信，并可以使用 debug_str() 函数输出最低限度的调试信息。删除它可以节省一些代码和 RAM 的使用。MT_ZDO_FUNC 和

MT_SYS_FUNC 是编译选项是 ZDO 命令可以和 Z-Tool 一起使用。

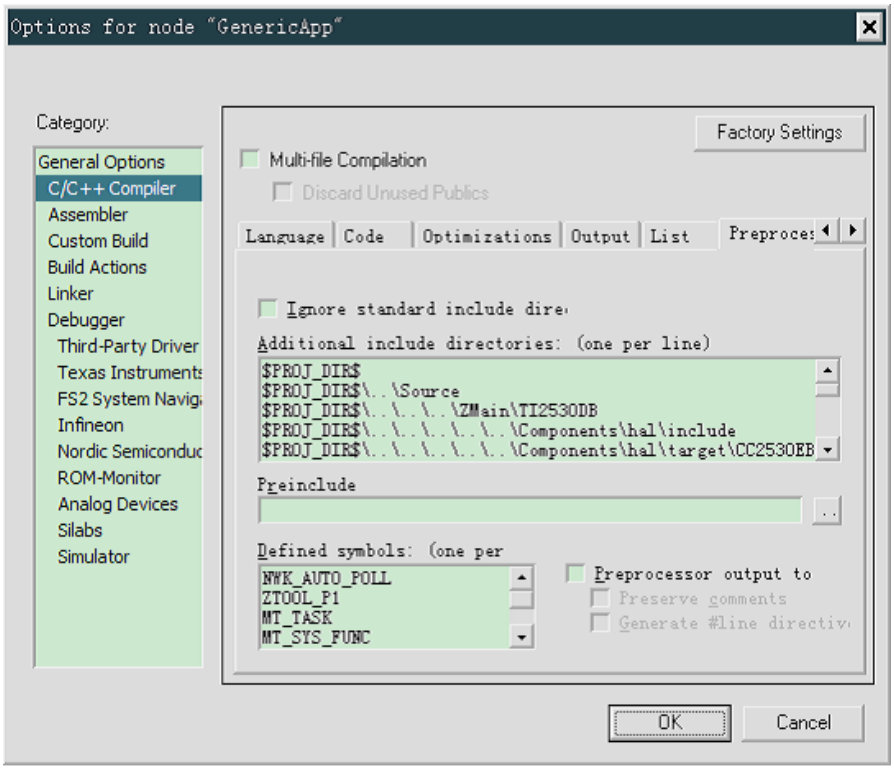


图 5-2 编译选项界面

5.3 网络地址请求和响应的解析

本系统中利用的是网路地址进行点对点的通信，这样可以提高数据传输的速率，本地节点首先需要在应用层中注册 NWK_addr_rsp 信息，然后调用 ZDP_NwkAddrReq()请求远程节点的网络地址。远程节点接受到该请求信息（该信息从属于 AF_DATA_CONFIRM_CMD），则根据 Cluster ID 选择处理函数，本例中使用 zdpProcessAddrReq()来处理网络地址请求。当处理完之后，通过调用参数 fillAndSend()将响应信息发送至本地节点。由于在应用层中注册了该响应信息，因此调用 SAPI_ProcessZDOMsgs()来处理响应信息）其详细流程图如图 5-3 所示[19]。

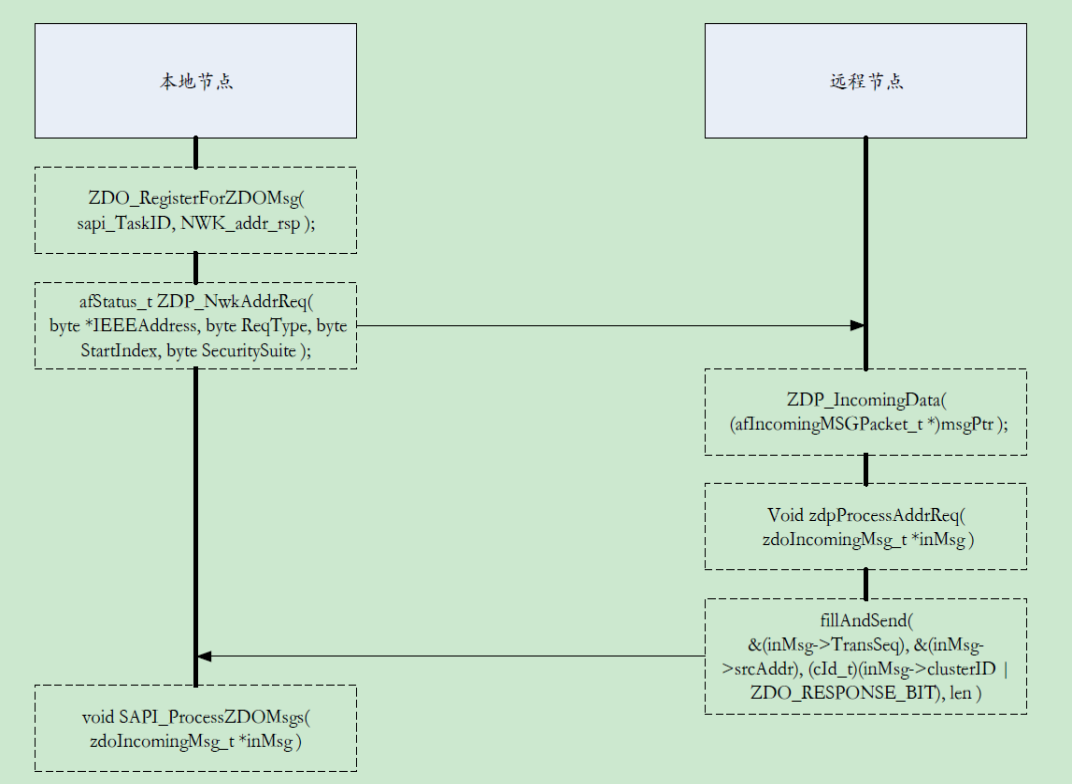


图 5-3 网络地址请求和响应

5.4 ZigBee 系统框架解析

ZigBee 协议栈中的每一层都会执行很多很多原语操作，因此对于整个协议栈来说，就会要执行很多并发操作。协议栈中的每一层都设计了一个事件处理函数，用来处理与这一层操作相关的各种事件。将这些事件处理函数看成是与协议栈每一层相对应的任务，由 ZigBee 协议栈中调度程序 OSAL 来进行管理。这样，对于协议栈来说，无论何时发生了何种事件，我们都可以通过对协议栈相应层的任务的调度，也就是事件处理函数来进行处理。这样，整个协议栈便会按照时间顺序有条不紊的运行。

要使用 ZigBee 系统，我们需要对他的系统有一定的了解，TI 的 Z-Stack 软件是一种轮询的小型操作系统，他的入口函数和普通的单片机是一样为 main 函数，它位于 ZMain 目录下的 Zmain.c 文件中，它的相关程序代码如下所示：

```
Int main(void)
{
  Osal_int_disable(INTS_ALL);
```

```
  HAL_BOARD_INIT();
  Zmain_vdd_check();
  InitBoard(OB_COLD);
  HalDriverInit();
  Osal_nv_init(NULL);
  ZMacInit();
  Zmain_ext_addr();
  Zmain_cert_init();
  ZgInit();
  AfInit();
  Osal_init_system();
  Osal_int_enable(INTS_ALL);
  InitBoard( OB_READY );
  Zmain_dev_info();
  Zmain_lcd_init();
  WatchDogEnable( WDTIMX );
  Osal_start_system();
  Return 0;
}
```

系统开始运行后，各个任务在系统的协调下工作。

系统在 for()循环下进行无线循环，循环中有两个关键的主循环数组，从图 5-5 中我们可以看出来，*tasksEvents 与*tasksArr，tasksEvents 这个数组存放的是从序号为 0 到 tasksCnt 的数组，每个任务在本次循环中是否要被运行，需要运行的任务其值非 0，否则为 0。而 tasksArr 数组则存放了对应每个任务的入口地址，只有在 tasksEvents 中记录的需要。运行的任务，在本次循环中才会被调用到。

正如代码所示的的，main 函数做了很多事情其流程图如图 5-4 所示

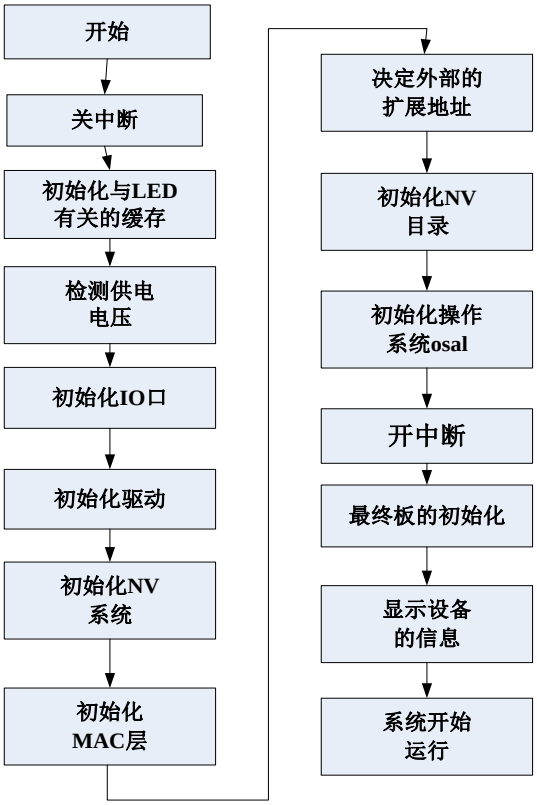


图 5-4 系统流程图

我们知道，每个操作系统我们知道都有一个“节拍”－tick，就像每一个“活人”都有心跳一样。Zigbee 的系统 OSAL 的心跳是 1ms。当然这个速度是可以设置的，在 osal_timer_activate 函数中开启了系统节拍，用 TICK_TIME 来定义其速度：

#define TICK_TIME 1000

这个 1000 是 micro-sec（微秒），而不是 milli－sec（毫秒）

CC2530 有 4 个定时/计数器，其中 timer4 用来做系统计时。在上述 osal_timer_activate 函数中，开启了系统计时，并将 timer4 的初始设为 TICK_TIME（1000），这样 timer4 就开始了从 1000 开始的减计数，减到 0 以后寄存器 TIMIF 会产生一个溢出标志。

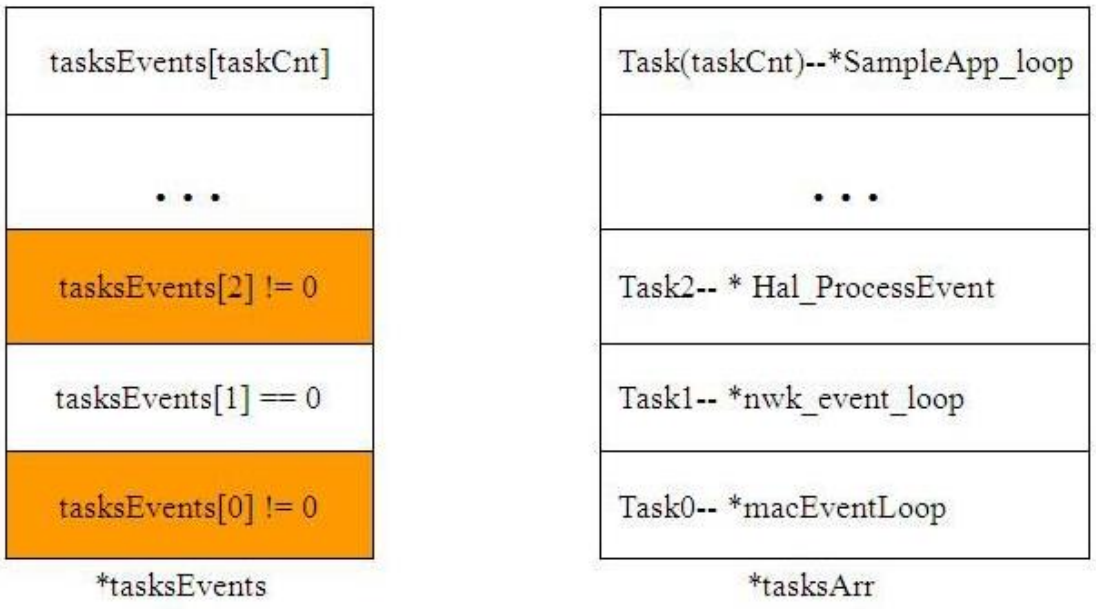


图 5-5 主循环函数处理机制

在主函数里的“Hal_ProcessPoll();这个函数里调用了 HalTimerTick，这个函数就是专门来检查是否有硬件定时器溢出的，如果有的话会调用 halTimerSendCallBack 这个函数，对溢出事件做处理。

在主函数里的“Hal_ProcessPoll(); 这个函数里调用了 HalTimerTick，这个函数就是专门来检查是否有硬件定时器溢出的，如果有的话会调用 halTimerSendCallBack()这个函数，对溢出事件做处理。

在 halTimerSendCallBack()这个函数里面调用了“callBackFunc”函数，也就是说每个定时器溢出后都有一个 callBackFunc 函数，在 HalTimerConfig 这个函数中，它可以对每个定时器进行定义。定义的时候是在 InitBoard，即板子上电初始化的时候就做了这个定义的。

我们看到 timer4 的 callBackFunc 函数是 Onboard_TimerCallBack，最终指向 osalTimerUpdate。

从上面的分析中我们知道它是每 1ms 被调用一次的，这样它就为应用程序提供了一个 ms 计时器，应用程序所用的定时往往以 ms 为单位足够了，这样的话就不用另外再占用硬件计时器了，毕竟只有 4 个。同时这个函数还提供了一个系统时钟－osal_systemClock。

5.5 应用程序的初始化

在系统上电和初始化任务时会加入任务函数：

STPM01_TaskID = task_id;

提醒的这个系统通过这个函数的参数分配任务到这个任务 ID，这个任务 ID 为应用程序设置一个定时器来设置相应的事件，或者发送一个系统消息到任务 ID 本身，这种操作方式是为了将大块的进程分成较小的一些进程。这样做可以使系统执行连续的时段而不会使系统花费很多时间在单一的时段中。当一个任务将一个大的工作进程分为每一个小的时段来执行的时候，这个任务正有效的影响着系统任务的分配。

STPM01_NwkState = DEV_INIT;

这个函数用于维持本地设备网络状态的副本，网络状态在上电的时候为不连接或者设备初始化，系统的任务在上电后将会获取 ZDO_STATE_CHANGE 的默认状态信息，所以我们需要对这个设备的网络信息进行初始化，一旦获得一个新的网络状态，这个任务将获得 ZDO_STATE_CHANGE 消息，需要注意的是当一个设备编译的时候使用了 NV_RESTORE 选项和连接到一个网络，这个 ZDO_STATE_CHANGE 消息将会在上电后立刻获得，且不需要通过空间传输，因为这个网络连接状态已经被存储在非易失性存储器中。

```
STPM01_Addr.addrMode = Addr16Bit;
STPM01_Addr.endPoint = 0;
STPM01_Addr.addr.shortAddr = 0x00;
```

这个默认地址的初始化是为了信息发送到相应的目的地址，Addr16Bit 为单点传送方式。

```
STPM01_epDesc.endPoint = STPM01_ENDPOINT;
STPM01_epDesc.task_id = &GenericApp_TaskID;
STPM01_epDesc.simpleDesc =;
SimpleDescriptionFormat_t *)& STPM01_SimpleDesc;
STPM01_epDesc.latencyReq = noLatencyReqs;
```

上面的这些编码是为了目标进行初始化，他使 AF 层知道如何将信息包送到断点，它将会通过发送一个系统的 SYS_EVENT_MSG 消息到这个任务。

5.6 应用程序的处理过程

在 Z-Stack 中，每个应用任务都通过 SampleApp_ProcessEvent()函数来处理任务中的事件。一旦 STPM01App_TaskID 任务的某个 OSAL 事件发生，那么就可以通过调用 SampleApp_ProcessEvent()函数来处理。在 SampleApp_ProcessEvent()中有一个事件处理循环。每当 OSAL 事件发生的时候，事件处理函数从 OSAL 任务处理循环将依次调用,应用程序的任务事件处理程序的参数是一个 16 位的位掩码，函数的调用可以设置一个或多个位，大多数时候只有一个关键事件被调用，而且几乎 SYS_EVENT_MSG 总是作为最高优先事项。

```
if ( events & SYS_EVENT_MSG )
  {
    MSGpkt                    =                    (afIncomingMSGPacket_t
*)osal_msg_receive( GenericApp_TaskID );
    while ( MSGpkt )
    {
      switch ( MSGpkt->hdr.event )
      {
        case AF_INCOMING_MSG_CMD:
          STPM01_MessageMSGCB( MSGpkt );
          break;
      }…
```

当这个信息到达网络目的，接收应用程序对象将通过 SYS_EVENT_MSG 进入到 AF_INCOMING_MSG_CMD 消息。然后进入到 STPM01_MessageMSGCB(MSGpkt)函数;去处理别的网络节点发送给此节点的数据。

5.7 Zigbee 串口通信方式的设置

CC2530 和上位机之间的串口通信采用的是 RS-232 标准。RS232 是一种异步通信方式，由于受到驱动器的电容负载的限制，传输距离相对都比较短，它一般用在 20m 以内的通信。RS232 通信方式即可以支持点对点的通信方式，也支持一点对多点的通信方式。RS232 通信方式传输数据可以选择从最小的每秒 50 波特到

115200 波特进行选择。 一般单片机之间的电平如果是相同的，可以直接按照 A 机的发送端口 B 机的接收端口相连，A 机的接收端口与 B 机的发送端口相连，然后将他们之间的地线相连接既可以通信。有些设备还硬件流控制，所以这些设备的串行通信有四根线，分别为接收端口 RXT、发送端口 TXD、可选 RTS 和 CTS。如果两个设备之间的电平不相同，需要经过一个电平转换芯片（如 max232）将他们的电平转换后才可以实现串行通信。CC2530 的串行通信就是一个很典型应用的例子。

CC2530 串行通信模式提供异步串行接口，可以与上位机通信。在此模式中，接口可以使用 2 线或者 4 线的方式，UART 模式提供全双工传送，接受器的位同步对发送功能没有影响。UART 字节的格式是一个起始位，8 个数据位、一个奇偶校验位（此为也可以作为第九位数据）、1 个或者两个停止位。数据传送过程中是以字节为单位的。如果寄存器 UxUCR 中的第九位和奇偶校验位设置为 1，就会产生奇偶校验和检测使能。

当运行在 UART 模式是，内部的波特率发送器设值 UART 波特率，由 UxGCR。BAUD_E[4：0]定义波特率，该波特率用于 UART 传送，也用于 SPI 传送的串行时钟速率。波特率由下式给出：

$$\text{波特率} = \frac{(256 + BAUD\_M) \times 2^{BAUD\_E}}{2^{28}} \times F \qquad \text{(式 5-1)}$$

式中：F 是系统时钟频率。

当 BAUD_E 等于 16 且 BAUD_M 等于 0 时，UART 模式的最大波特率是 f/16[17]。

要使用 Zigbee 的串口，首先需要对初始化部分的修改，具体的修改是在源程序 MT_UART.C 的 MT_UartInit()函数中，我们需要设置相应的参数，这些参数包括设置波特率、是否允许硬件流控、接收或者发送数据的缓冲区等。波特率的参数默认为 38400，允许或者禁止硬件流控决定我们采用 2 线的 UART 通信或者 4 线的 UART。在此函数中如果配置了 uartConfig.callBackFunc = NULL;则到 void MT_UartProcessZToolData (uint8 port, uint8 event)函数中根据实际的需要进行相应的配置。这个函数实际就是通信的协议，具体说来就是把串口发来的数据包进行打包，校验，生成一个消息，发给处理数据包的任务。发过来的串口数据具有以下格式：

0xFE-数据帧头；

DataLength-Datapayload 的数据长度，以字节计，低字节在前；

CMO-命令低字节；

CM1-命令高字节;(ZTOOL 软件就是通过发送一系列命令给 MT 实现和协议栈交互);

Data payload-数据帧具体的数据，这个长度是可变的，但是要和 DataLength 一致;

FCS-校验和。

设置完这些后我们需要到应用层的初始化函数加入串口的初始化函数 MT_UartInit(); 注册 MT_UartRegisterTaskID (GenericApp_TaskID);这样一个处理串口的任务。在应用的主函数中加入处理串口的簇和处理函数。

5.7.1 CC2530 和 STPM01 的通信

SPI 接口是一种串行外围接口，它是一种全双工的高速通信方式，可以实现串行同步数据传输，数据传输速度比异步的串行传输要快，传输的速率最高可以达到几 Mbps。由于 SPI 协议比较成熟，所以很多外围的一些设备都支持 SPI 通信方式。支持 SPI 接口的设备一般都两个工作模式可以供选择：主工作模式或者从工作模式。和同步串行相同，有些外围设备在 SPI 的发送和接收可以同时支持查询或者中断方式来实现。这种通信方式有使用三线制的或者四线制的，有的 SPI 接口没有主机输出或者从机输入线 MOSI，有些 SPI 接口还带了中断信号线。SPI 接口一般包括了主机输入/从机输出数据线、串行时钟线、主机输出/从机输入数据线和低电平有效的从机选择线。

STPM01 和外部通信时使用的是 SPI 的通信方式，STPM01 做为从机。根据通信的协定，可以分为远程复位的时序，读数据时序，写命令时序。我们要实现这些功能需要根据 STPM01 特定的编程方式实现这些功能。

5.7.2 远程复位

远程复位请求的时序如图 5-6 所示，这个时序的时间需要保持最少 30ms。这个内部复位信号的被定义为 RRR。和 POR 不同得是，此信号在模拟模式下不会有 30ms 的延迟重新启动，在数字模式下不会有 120ms 的延迟重新新启动[18]。

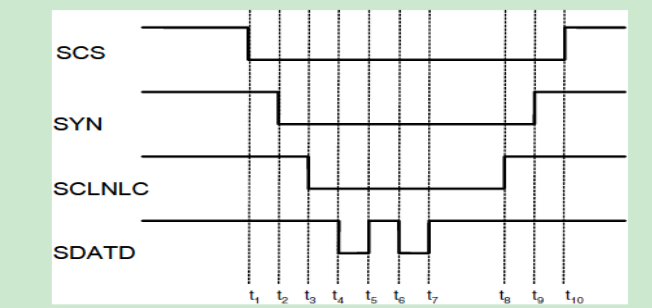


图 5-6 远程复位时序

5.7.3 读数据记录

当应用中有单片机的时候可以实现数据记录的读取，从而可以读取 STPM01 的测量值和所有系统信号。这个读取的信号的时序需要保持 30ms，读取信号包括了两个阶段，我们定为锁存和移位。锁存用于将采样结果送到发送锁存器，这个发送锁存器使数据很容易地搬移到 SPI 接口。这个过程需要将 SYN 激活当 SCS 为空闲模式的时候，SYN 脉冲的长度必须大于两个测量时钟周期。数据的移位发送在 SCS 为激活状态的时候。首先读到的数据位为数据记录的低有效位，第四个位为最有用的位。每个位包含了 8 个字节[18]。其时序图如图 6-7 所示。

这个系统从 STPM01 读取数据时应该校验每个数据记录的完整性，如果校验失败，将会进行再次的读取。但条件是在移位操作允许的情况下。否则一个新的数据将会所存到发送到发送锁存器从而导致数据的丢失。

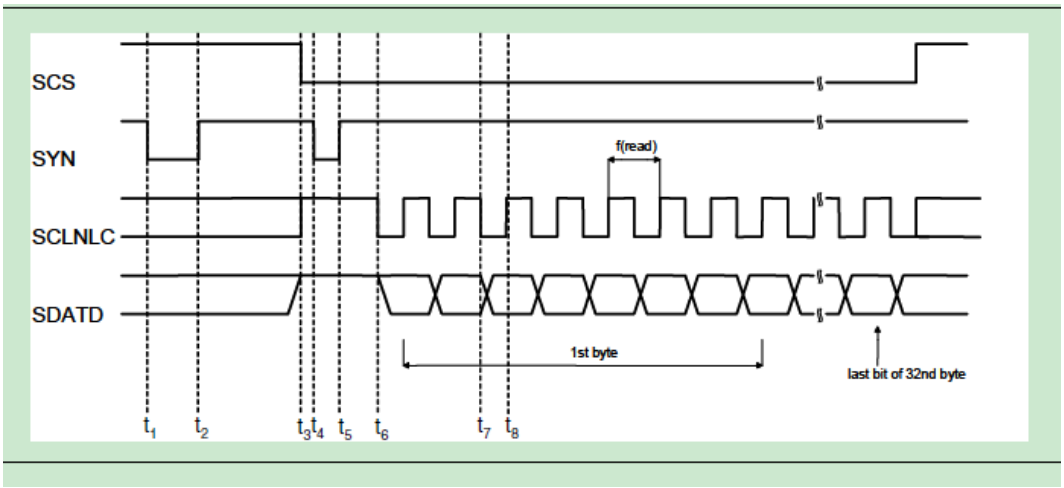


图 5-7 读数据的时序图

5.7.4 写时序

STPM01 在 DFP 数据记录中包含 8 种模式信号，其中的三个用于内部测试的目的，而另外的 5 个用于改变一些操作。在 STPM01 不支持或者 POR 发生的时候这些模式信号无效。这些模式信号可以通过正常的写时序操作写到芯片内部。因此，根据相应的模式信号的定义，我们可以通过清除 RD 来清除所有的系统信号。要实现这种方式可以通过两个方式，第一个方式是产生 POR 信号，但是这种方式将会清除和复位整个设备。另外一个种方式是设置 TSTD 位，这种设置方式将会在 SCS 到达空闲状态后。然后 TSTD 将会清除所有的系统信号，这些系统信号包含了 TSTD 本身，不过它不能复位整个设备[18]。表 5-1 为模式信号描述表。

表 5-1 为模式信号描述表

编号	信号名	位值	状态	二进制命令
0	BANK	0/1	用于 RC 启动过程。	0111000x
1	PUMP	0/1	0 为 MOP 和 MON 一般操作模式；1 为 MOP 和 MON 提供驱动信号用于充电中的 DC-DC 转换。	0111001x 1111001x
2	保留			
3	保留			
4	CSEL	0/1	0 为选择电流通道 1；1 为选择电流通道 2。	0111100x 1111100x
5	RD	0/1	0 为这 56 配置位由 OTP 相应位开始；1 为这 56 配置位由屏蔽锁存位开始	0111101x 1111101x
6	WE	0/1	0 为任何写操作的配置位在屏蔽锁存器进行；1 为写操作在 OTP 和屏蔽锁存器进行	0111110x 1111110x
7	Precharge	1	交换 32 位读取的数据值。	1111111x

每个写使能位包括了配置和模式位，他们都有自己的 6 位地址。配置位的 6 位地址为对应自身的 10 进制值，模式位的地址指示了相应的信号段。为了改变一些锁存位的状态，CC2530 需要通过 SPI 方式发送一些数据到 STPM01 的数据位。这些数据总共有八位，包含了位于最高位的 1 位数据位、6 位目的地址位和位于最低位

的保留位。例如，我们需要发送配置值 47 给此测量芯片，我们需要先将十进制的 47 转换为 6 位 2 进制的值。这个位命令组成就会变成：1 为的数据位、6 位的地址位和 1 位的描述位（0 或者 1）从而这个最终的 2 进制值为 01011111（0x5f）或者 01011110（0x5e）。写使能和模式信号对应的时序如图 5-8 所示。

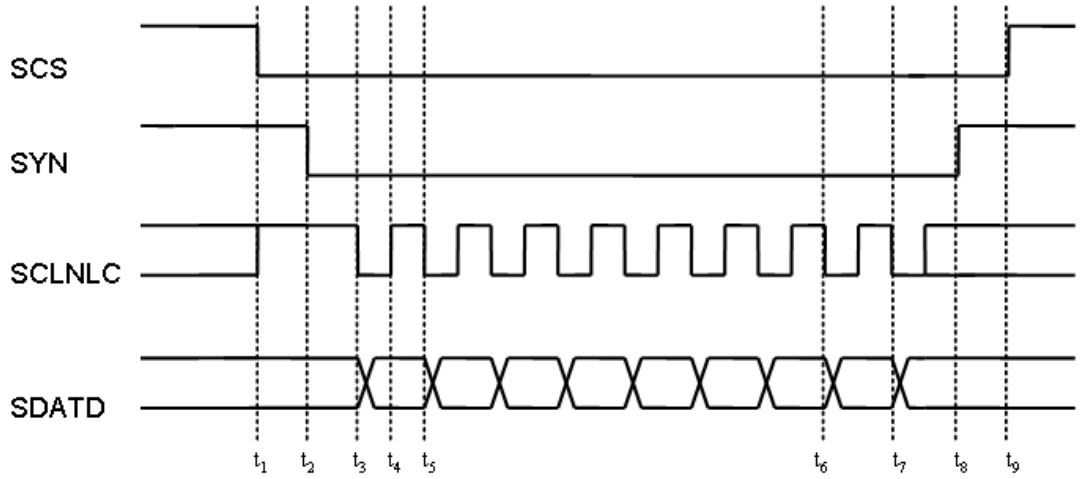


图 5-8 写时序

5.7.5 通信的过程

由于测量节点中有 STPM01，所以主要讨论测量节点，我们需要优先设置一个主任务，osal_start_timerEx(STPM01_TaskID, start_EVT, 1000);也就是 1s 中和 STPM01 通信一次。采样次数相对与上位机交换的时间相对比较小些。整个系统中测量节点和 STPM01 的 SPI 通信与测量节点和上位机间接交换数据是相对独立的。

每次任务发生后，CC2530 就会与 STPM01 进行 SPI 通信了，我们需要在任务函数中设置相应的事情，这些事情包括有更新功功率定值中的余数和整数，产生一个输出脉冲给 LED，调用 12 个子任务，更新子任务的指针，加入一些特定的任务等。子任务包括:

1.STPM01 值的锁存；

2 读取 STPM01 寄存器的值；

3 重复读取 STPM01 寄存器的值；

4 检验寄存器的校验码和比较两次读取的结果是否一致；

5.解码 STPM01 寄存器的值是否正确；

6 STPM01 的状态值是否正常；

7.如果读取两次寄存器的值一致计算有功功率的值和更新旧的有功功率的值；
8.计算有效电压的平均值；
9.计算有效电流的平均值；
10.更新无功功率的余数和整数部分；
11.加入读到的两次值不一致定量值为 0；
12 检测是否有通信模块的请求发送功率数据.如果有发送给通信模块、否则保存到数组中暂时保存起来。如果测试节点接收到了上位机发来的请求发送用电数据命令，则将这段时间的用电数据发送给上位机，由上位机根据一定的规则解析后显示在屏幕上。

5.8 通信模块的处理过程

通信模块是整个 zigbee 系统的核心，它是具有协调器功能的模块，起到了启动和控制网络的作用，它不仅负责和上位机通信，而且和测量节点通信，是测量节点和上位机交换数据的桥梁。从 5-6 流程图我们可以看出，整个通信模块是这样工作的：由于它是协调器，所以上电后进行相应的初始化工作，并建立一个稳定的 zigbee 网络，每当上位机通过串口发送数据给上位机时，通信模块就会触发中断，并进入中断函数处理上位机发来的数据，这些数据包括了要测量节点的网络地址和相应控制测量节点的命令，数据处理完成后更新相应的标志位为 1，告诉系统发送相应的数据到网络，然后系统开始定时，等待接收网络中返回的数据。如果在规定的时间内有数据返回，通信模块就将相应的用电数据返回给上位机。否则继续等待上位机发来的下一次命令。

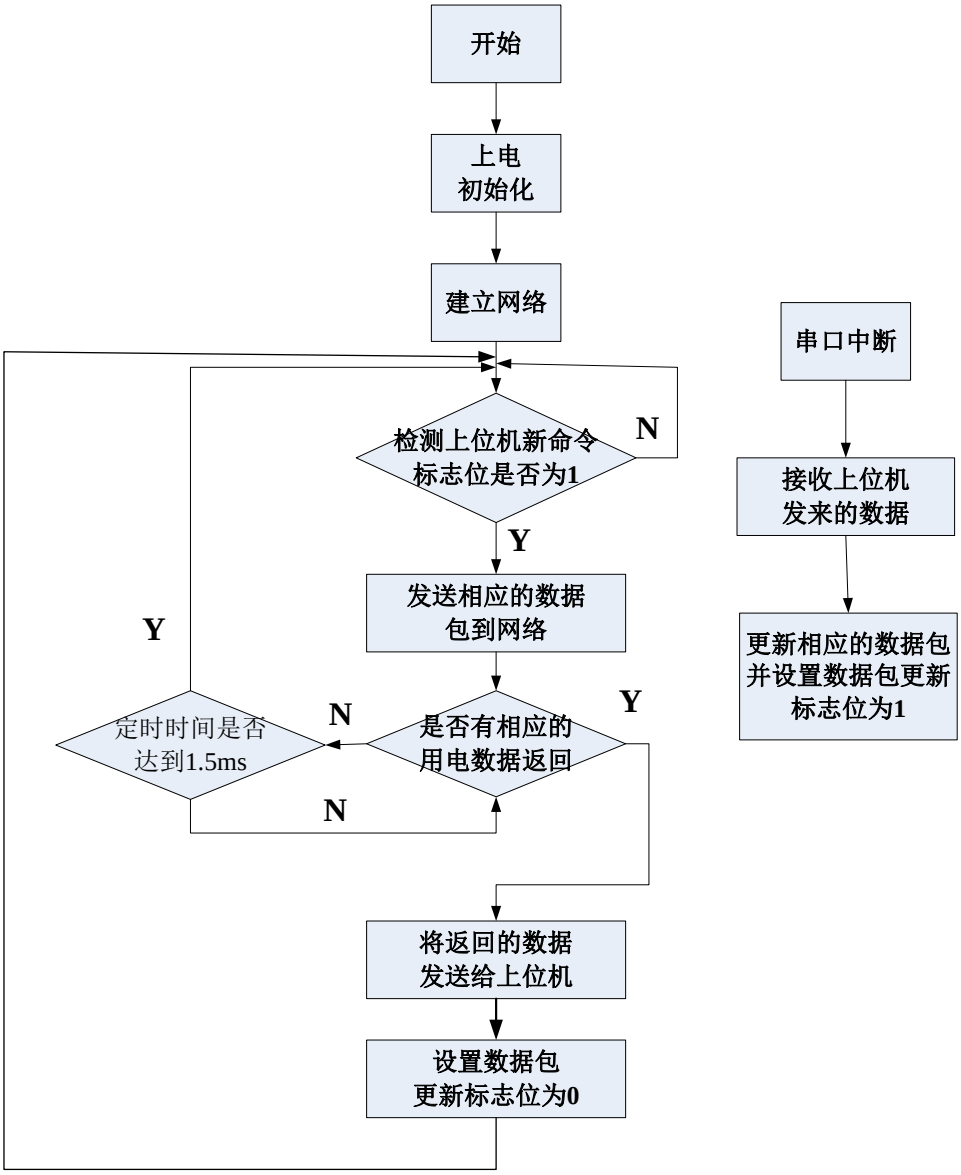


图 5-9 通信模块的流程图

5.9 测量模块的处理过程

测量模块主要通过 SPI 通信模式和 STPM01 读取用户的用电数据，并且将用户的用电数据发送给通信模块，并最终显示在电脑显示屏上。测量模块上电后先进行初始化，然后加入到 ZigBee 网络中，将用户的用户数据清零后开始读取用户的用电数据，如果通信模块发送了请求命令，要求将用户数据送回，则将用户的用电数据发送给通信模块。其大概的工作流程图如图 5-10 所示

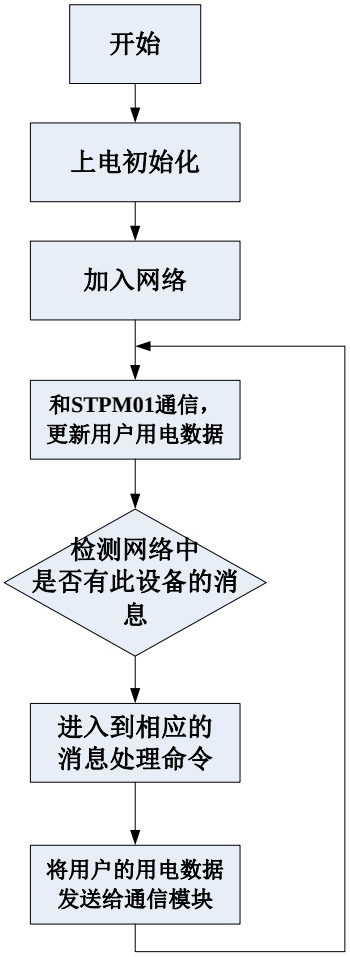


图 5-10 测量模块的流程图

5.10 上位机软件介绍

5.10.1 使用的编程语言

Visual Basic.NET 是从 Visual Basic 语言演变而来，是一种为高效的生成类型安全和面向对象的应用程序而设计的语言。Visual Basic 允许开发人员开发面向 Windows、Web 和移动设备的程序。与所有面向 Microsoft .NET Framework 语言一样，使用 Visual Basic 编写的程序都具有安全性和语言互操作性方面优点。VB.net 是微软最新平台技术,是.netframeworkSDK 的一种语言。VB. net 和 VC#.net 在功能上没有区别。编译以后生成的可执行文件被称为 Assembly，即程序集。。VB. Net 具有很多特点：真正成为面向对象以及支持继承性的语言； 窗体设计器

支持可视化继承，并且包含了许多新的特性，比如自动改变窗体大小、资源本地化支持、数据类工具内在支持 XML 数据；

直接建立在.NET 的框架结构上，因此开发人员可以充分利用所有.NET 平台特性，也可以与其他的.NET 语言交互；为 Windows 应用程序提供了 XCOPY 部署，开发者不再需要为 DLL 的版本问题担忧。编译环境我们采用了 Microsoft Visual Studio 2010。

5.10.2 Microsoft Visual Studio 2010 简介

Visual Studio 是微软公司推出的开发环境。是目前最流行的 Windows 平台的应用程序开发环境。Visual Studio 2010 版本于 2010 年 4 月 12 日推出，其集成开发环境的界面被重新设计和组织，变得更加简单明。Visual Studio 2010 同时带来 NET Framework 4.0、Microsoft Visual Studio 2010 CTP，并且支持开发面向 Windows 7 的应用程序。除了 Microsoft SQL Server，它还支持 IBM DB2 和 Oracle 数据库。它支持包括了 VB.NET、Visual C#、Visual C++、Visual F#在内的多种编程语言，我们可以使用它建立网站，编写应用程序等。

5.10.3 程序设计原理

要实现上位机通过串口和下位机通信实现有两个途径：其一就是使用 MSCOMM 控件，此控件提供很多方便操作的属性和方法，利用它们可以很方便地实现你的目的；另外一种途径就是使用 Windows API，微软为开发者提供了很多相关的串行操作的编程接口，使用这些接口，可以很完成更为强大的功能。无论使用哪一种途径，其基本的操作步骤大致一样的：首先初始化串口，比如端口号，波特率等属性，然后打开端口，通过接受缓冲区读上行数据，通过发送缓冲区来写下行数据。最后通过事件驱动来反映数据的到达与发送过程，另外在通讯过程中的错误的产生也可以通过 CommEvent 属性来管理。使用 MSCOMM 控件主要是通过事件来处理串行口的交互，即当数据到达时，控件的 OnComm 的事件就会来捕获或处理这些通讯事件。而 OnComm 事件也可以用来捕获和处理通讯错误。在实际应用中，一个 MSCOMM 控件就对应一个串行口，所以如果要处理多个串行口的话，必须有相应数量的控件与之对应。

第六章 系统测试条件和测试结果

在本系统中，其测试布局如图 6-1 所示，测试点 A 测试一台台式电脑的用电数据，测试点 B 测试一台打印机的用电数据，测试点 C 测试一台笔记本电脑的用电数据，测试点 D 测试一台负载仪的用电数据，测试点 E 测试一台示波器的用电数据。由于我们使用的是将天线布局在 PCB 板上，同时将 CC2530 的发射功率调为一般的级别，所以他们之间的通信距离不是很远，这样也是方便我们进行一些实验。

系统上电后，整个系统开始进行了初始化，上位机显示“系统正在初始化，请耐心等待”。然后我们将通信模块和测试节点都安装好后几乎同时上电，过一会上位机开始连接网络成功，此时我们按下秒表。过了一些时间后，上位机开始显示一号节点的用电数据，间隔一定时间后更新编号为 2 测试节点的用电数据，不断地循环下去。当五个节点的数据全部都更新完成 1000 次后，我们按下了秒表，用时 540s，所以测算出循环一次花费的时间为 540ms。平均下来一个节点花费的时间就是 1ms，所以对于有 540 个节点的小区系统来说我们需要 540s 更新一次数据。用电数据一般都是按月进行结算的，采用 540s 的数据更新也是为了实时监测用户的用电数据，所以整个设计基本满足我们的系统需求。

为了对 ZigBee 协议的了解和对我们整个方案进行不断地完善，我们模拟了实际使用过程中可能出现的情况。主要测试的是网络中的测试节点不能正常工作上位机能否及时报警，同时是否会影响其他测试节点正常工作，同时测试了 ZigBee 是否有中继功能。我们将 A 点的测试节点取下来，上位机间隔一定时间后报警，但不影响其他测试点用电数据的采集。此过程我们程序的运行过程为上位机发送采集 A 节点的数据时，发送第一次收不到 A 节点发回的数据，接着上位机继续发送一次接受 A 节点的数据，如果第五次收不到 A 节点的数据，则上位机认为此节点工作不正常。由于报警后上位机会发送继续采集其他节点的命令，所以上位机除了不断显示对应报警信号外，还能继续接收其他测试节点的数据。之后我们把测试点 A、B、C、D 的测试节点断电，发现虽然测量节点 E 可以在未断电的情况下，上位机显示包括 E 的所有测试节点的报警信息。这时候我们把 D 节点加上，则在上位机既可以读取测试节点 D 节点的用电数据，还能读取 E 节点的用电数据。之所以这样是由于 ZigBee 网络的传输距离并不长，其实 E 节点传输的数据是经过了

其他节点传输到通信模块，并最终送给上位机显示的。在 D 节点加入后，D 节点除了和 STPM01 进行 SPI 通讯，把用电数据传给上位机外，还充当了 E 节点的路由功能。

表 6-1 为某时刻相应测试点的用电数据。测试时除了需要注意用电的安全之外，还需要注意我们仪器的测量范围，电压需要低于 240V，电流需要低于 40A。

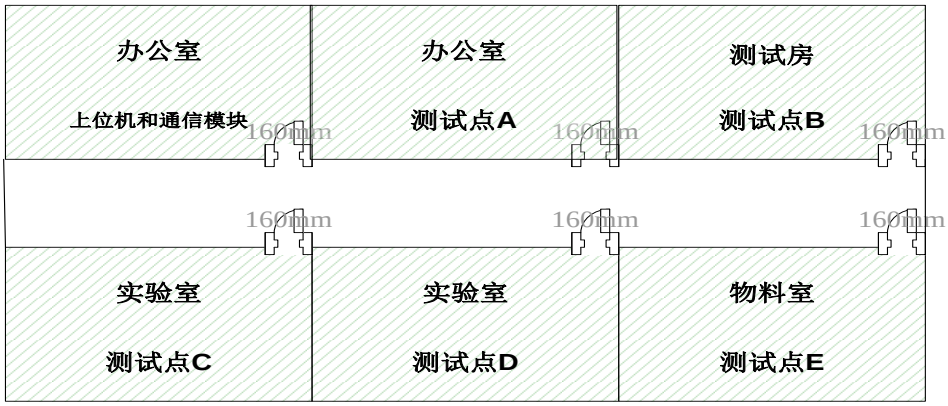


图 6-1 测试条件的平面图

图 6-2 某时刻相应测试点的用电数据。

以下是通信模块和测量模块的两个实物图，通信模块主要包括了一个 CC2530

芯片和一个和 SN65C3243 芯片。测量节点主要有互感器，STPM01，cc2530 芯片，它主要负责采集用户的用电数据，并通过 ZigBee 网路发送给通信模块，并最终显示在上位机上。

图 6-3 通信模块的实物图

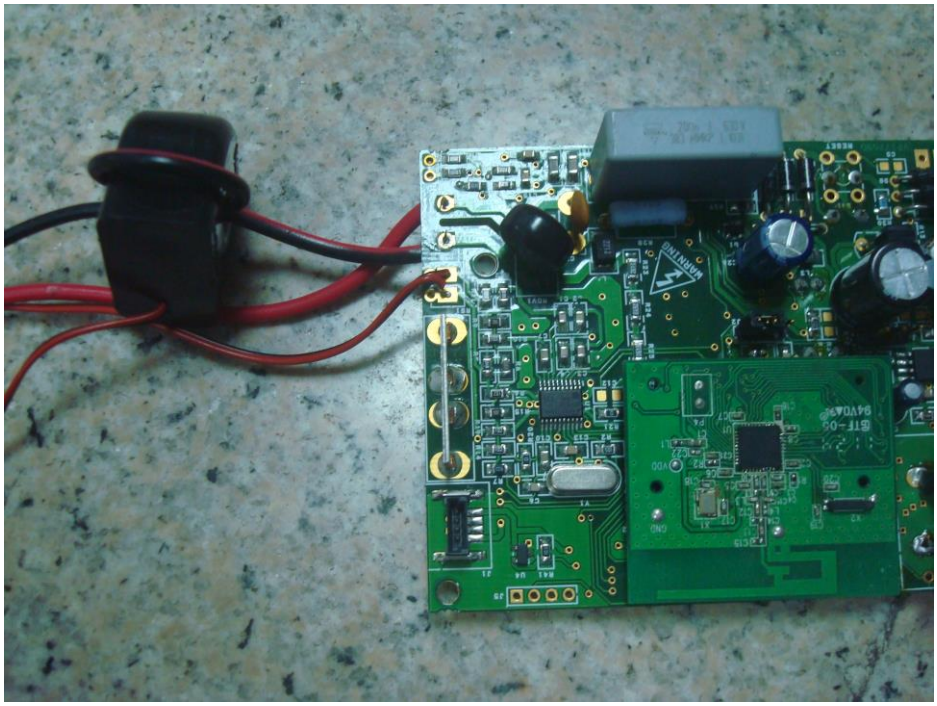


图 6-4 测量节点的实物图

第七章 总结与展望

本文所研究的社区智能无线抄表技术是基于TI公司最新推出的CC2530无线传感芯片，以及意法半导体公司的推出的第一款电能计量专用芯片STPM01作为研究对象，所研究的最新一代无线抄表技术。通过使用ZigBee技术可以有效的降低成本以实现无线传感，以及无线收发等功能。

系统由三部分组成，第一部分是采用STMP01芯片作为下位机，以实现社区住户内的电量，以及相关数据的采集工作；第二部分是采用CC2530无线传感芯片所组成的设计无线传感网络，进行数据的传输、转发等工作；第三部分则可根据需要配置PC机，作为上位机，以实现数据的接收及显示工作。

通过完成这次毕业设计，我查阅了大量的资料，学到了许多平时没有接触过的知识，极大的丰富了我在无线传感技术领域的认识。并且在这段时间内，学习很很多硬件知识，也了解了许多与本系统相关的芯片，最后，能选择STPM01芯片作为系统数据的采集芯片，也是因为这款芯片编程相对简单，精度相对较高。

由于个人能力有限，在软硬件系统的设计上，都有一些不足之处：

(1) 由于条件的限制，这个硬件部分的抗干扰能力没有做到最优，PCB的布局没有做好。

(2) 采用了模拟环境代替实际的工作环境，但是无法预测实际环境下可能出现网络阻塞或者连接不正常的问题。并对出现的问题进行优化。

(3) 由于国内ZigBee的发展现状，我们使用的协议还很难做到通用性。

(4) 还需进一步完善软件和硬件的设计。

虽然还有很多问题有待解决，但是能够基本实现基于CC2530的无线抄表系统的功能，而且可以较为准确的读取用户的用电数据。我认为，在不久的将来，为无线抄表毕定会取代有线抄表，并会在人们的生活中得到广泛的应用。在后续的学习和工作中，我要尽可能的提高自身水平，并不断完善设计功能，为社会的发展做出更大的贡献。

致　谢

从2010年底开始做这篇论文的开题报告起,我就开始了为论文编写准备资料，

在学习和实验当中虽然遇到了许多的困难，但是我的校内导师毛玉明老师和校外导师许淮武，以及同学们的鼓励和帮助。这些都给了我前进的动力。特别要感谢我的导师毛玉明,毛老师作为我无线传感技术的启蒙老师,引导我进入ZigBee技术世界，悉心指导我作项目开发。毛老师对工作认真负责的作风，是我学习的榜样，完成这篇论文后，我感觉到，我不仅学到了很多技术知识，同时也学到了许多做人的道理。

我也感谢研究生三年期间所有指导过我的老师，感谢他们对我无私的教诲和帮助。

感谢刘华同学对我的帮助和指点。他给我提供了许多资料，并对我的程序提出了非常宝贵的意见。对于我一个对编码技术的初学者来说，他们的帮助，使我对编码知识有了较为系统的认识和学习。我能顺利的完成论文与他的帮助是分不开的。

感谢我研究生三年的同窗，是你们让我感受了大学的美好。和你们共同走过的这研究生三年，我感到无比的快乐。

每一份收获，都凝结了很多的汗水。在论文即将完成之际，我感到非常激动，从论文的开题，方案的设计、到论文的拟写的这段时间里，导师、同学和朋友们给了我太多的帮助，可以说少了他们的帮助，完成的过程会变得非常地艰辛。在这里致以我最真诚的感谢!

在以后的人生道路上，我将铭记我的老师和朋友们，他们是我人生中无比珍贵的财富。

参考文献

[1] 《ZigBee 技术实践教程-基于 CC2430/31 的无线传感器网络解决方案》.

[2] 张少虎,《基于 ZigBee 的自动抄表系统的设计》[硕士学位论文].西安科技大学，2011.

[3] 金福根，肖民等，居民电能表集中抄表系统存在问题分析[J]，电力需求侧管理.2007.9（5）；67-68.

[4] 任晓晖，宋丽娜,《ZigBee 技术在远程无线自动电抄表系统中的应用前景》，黑龙江农业工程职业学院学报，2008 年 9 月第 3 期.

[5] 潘家根.无线传感器网络通信机制与节能的研究[硕士学位论文].电子科技大学，2007.

[6] 陈斌.无线“第三者”—UWB.计算机世界,2004/03/08

[7] 别坤.NFC:近距离无线通讯的新宠.互联网周刊,2011-05-20

[8] 金纯，蒋小宇，罗祖秋.ZigBee 于蓝牙的分析与比较[J].标准与技术追踪，2006.6：17-20.

[9] 宓霖，基于 ZigBee 技术的无线数据收发器 MAC 层协议研究与硬件设计[硕士学位论文].东南大学，2005.

[10] Institute of Electrical and Electronic Engineers, Draft Standard for Part 15.4;Wireless Medium Access Control Layer(MAC) and Physical Layer(PHY) specifications for Low Rate Wire lee Personal Area Networks (LR-WPANs),DRAFT P802.15.4/D18,Feb2003,55~179.

[11] 蒋挺，赵成林等.紫蜂技术极其应用（IEEE 802.15.4）[M].北京邮电大学出版社，2006.

[12] Yimin Liu,A Two-hop Energy-efficient Mesh Protocol for Wireless Sensor Networ,the Degree of Master of Applied Science,Ottawa,Ontario,Canada,Carleton University,2004.

[13] Sing Yiu Cheung, Sinem Colery, Baris Dundar, Traffic Measurement and Vehicle Classification with a Single Magnetic Sensor, California, and PATH working paper, 2004:20 ~ 28.

[14] 金春嫣，基于 ZigBee 和红外检测的停车位监控系统的研究[硕士学位论文].东南大学，2010.

[15] 刘钊，全网全功能 Zigbee 无线抄表系统的设计与实现[硕士学位论文]，北京邮电大学，2011.

[16] ZigBee 数据传输实验例程手册,郑州新双恒信息技术有限公司,2010.

[17] CC253X 用户指南（中）,郑州新双恒信息技术有限公司,2009.

[18] STPM01 DATASHEET, ST, 2011.

[19] ZigBee2007 协议栈 API 函数使用说明,锋硕电子科技有限公司,2010.

[20] 杨晓华.无线城市的研究与设计[硕士学位论文]. 上海交通大学,2008-12-01

[21] 欧杰峰,基于 IEEE802.15.4 的无线传感器网络组网研究[硕士学位论文].浙江大学,2006

[22] 贺文.基于 IEEE802.15.4/ZigBee 的无线传感器网络研究[硕士学位论文].浙江大学，2006

[23] Jianbin Jiao,Qixiang Ye，Qingming Huang，A configurable method formulti-style license plate recongnition[J]，Pattern Recognition[J]，Volume 42，lssue 3，March 2009，pp358-369

[24] 李文仲，段朝玉等.ZigBee2006 无线网络与无线定位实战[M].北京航空航天大学出版社，2008

[25] 窦振中.单片机外围器件实用手册[M].北京航空航天大学出版社，2000

[26] 张宁，王越，王东，基于精简协议栈的 ZigBee 网络节点研究[J].单片机与嵌入式系统应用，2009.2

[27] 高守玮,吴灿阳.ZigBee 技术实践教程:基于 CC2430/31 的无线传感器网络解决方案[M].北京航空航天大学出版社，2009

[28] 李志明，自动抄表系统的研究[D].浙江大学，2005：1-8，21.

[29] SK-CC2530 ZDK ZigBee 2007/PRO 无线开放套件用户指南，湘潭斯凯电子科技有限公司出版

[30] SK-CC2530 ZDK ZigBee 2007/PRO 无线开放套件实验指导书(下)，湘潭斯凯电子科技有限公司出版

[31] 任丰原，黄海宁，林闯.无线传感器网络[J]软件学报，2003，14(7)；1281-1291

[32] 田真,无线传感器网络中的差错控制技术研究[D].山东大学,2009

[33] 孙利民，李建中，陈渝，朱红松.无线传感器网络[M].北京.清华大学出版社.

[34] 别坤.NFC:近距离无线通讯的新宠.互联网周刊,2011-05-20

[36] 迁华斐.基于 Zigbee 和 ARM 的智能家居系统的设计.东北石油大学硕士论文,2011-03-26

[37] 李生辰.无线传感器网络在智能家居系统中的应用.辽宁工学院硕士论文. 2007-03-01

[38] 刘江.单相防窃电电能表. 哈尔滨智通科技有限公司

[39] 梁海涛.青铜峡农电防窃电管理系统研究.重庆大学硕士论文. 2007-10-01

[40] 薛利娟.基于 ZigBee 技术的无线自组织的研究与设计.西安电子科技大学硕士论文

.2011-01-01

[41] 基于 ZigBee 和 Meter-Bus 技术的自动抄表系统的研究及实现[硕士学位论文].云南大学，2011.