

**本科毕业设计（论文）**



|  |  |
| --- | --- |
| **题目** | **楼宇火灾无线监测报警** |
| **系统的设计** | |

|  |  |  |  |
| --- | --- | --- | --- |
| **学生姓名** | **任思霖** | **学　号** | **201431030690** |
| **教学院系** | **机电工程学院** | | |
| **专业年级** | **测控技术与仪器2014级** | | |
| **指导教师** | **罗宾** | **职　称** | **高级实验师** |
| **单　　位** | **西南石油大学** | | |
| **辅导教师** |  | **职 称** |  |
| **单　　位** |  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **完成日期** | **2018** | **年** | **5** | **月** | **6** | **日** |

Southwest Petroleum University

Graduation Thesis



**Design of Building Fire Wireless Monitoring and Alarm System**

**Grade: 2014**

**Name: Ren SiLin**

**Speciality: Measurement Control**

**Technology and Instruments**

**Instructor: Luo Bin**

**School of Mechanical and Electrical Engineering**

**2018-5**

# 摘要

在社会高速发展的今天，建筑的外观与质量已经得到了很大的提升，与此同时，由于建筑的密集度增大与电器设备的规模化运用，建筑物本身所受到的威胁也越来越大，这导致增加了火灾发生的概率。

本设计中提出了一种能够监测温度，一氧化碳（CO）浓度与可燃性气体烟雾浓度的火灾监测系统。该设计结合了无线传感器网络技术，应用ZigBee技术来完成无线数据传输，采用ZigBee的最佳片上解决方案CC2530作为终端（采集数据）与协调器（收发数据）。

硬件系统包含三个部分： 集成传感器采集模块、ZigBee传输模块与主控模块；软件系统运行在PC端上，实时读取串口数据，对传感器采集的数据进行融合处理，融合算法是基于遗传算法优化的BP神经网络算法与模糊逻辑进行多方面的决策判断。经过数据融合算法处理后的决策值有效度大于传统设备，极大程度上避免了传统设备因传感器单方面的数据输入造成精度不高与误报产生的情况出现。

本文在理论的基础上做了相应的结果论证。试验数据表明，通过传感器数据融合算法后的数据精确度比传统设备数据精度高，且能够准确分辨阴火和明火火灾，确定了系统的研究方向，且系统在实验中工作稳定，数据可靠，满足火灾报警的需求。

**关键词：**ZigBee；火灾探测；数据融合；神经网络算法；

# Abstract

With the development of society, the appearance and quality of buildings have been greatly improved. At the same time, due to the increase in the density of buildings and the large-scale use of electrical equipment, the threat to the buildings themselves will increase. And it shows that the probability of fire is increasing.

In this design, a fire monitoring system capable of monitoring temperature, carbon monoxide (CO) concentration and flammable gas smoke concentration is proposed. The design incorporates wireless sensor network technology, uses ZigBee protocol for wireless data transmission, and uses ZigBee's best on-chip solution CC2530 as a terminal (acquisition data) and coordinator (transceiver data).

The hardware system consists of three parts: integrated sensor acquisition module, ZigBee transceiver module and master module; the software system runs on the PC, reads the serial data in real time, and carries out fusion processing of the sensor data. The fusion algorithm is a BP neural network algorithm optimized based on genetic algorithms. After the fuzzy rules, various decision-making decisions are made. The decision value processed by the data fusion algorithm is more effective than the traditional device, Therefore, the situation that the accuracy of the conventional device is not high and the false alarm is generated which caused by the unilateral data input of the sensor is avoided.

This article makes the corresponding result demonstration on the basis of the theory. According to the experimental results, the accuracy of data after sensor data fusion algorithm is higher than that of traditional equipment data, and can accurately distinguish the fires of the overcast and the open flame. The research direction of the system is determined, and the system is stable in the experiment, the data is reliable, and the fire alarm is meet demand.

**Keywords:** ZigBee; fire detection; data fusion; neural network algorithm;

**目录**

[摘要 I](#_Toc513740612)

[Abstract II](#_Toc513740613)

[第1章 绪论 1](#_Toc513740614)

[1.1 论文的研究背景及意义 1](#_Toc513740615)

[1.2 国内外研究现状 2](#_Toc513740616)

[1.2.1 国内研究现状 2](#_Toc513740617)

[1.2.2 国外研究现状 3](#_Toc513740618)

[1.3 论文的主要研究内容 3](#_Toc513740619)

[第2章 无线传感器网络技术及ZigBee简介 5](#_Toc513740620)

[2.1 无线传感器网络技术 5](#_Toc513740621)

[2.1.1 无线传感器网络体系结构 5](#_Toc513740622)

[2.1.2 无线传感器网络的特点 6](#_Toc513740623)

[2.2 ZigBee简介 7](#_Toc513740624)

[2.2.1 ZigBee技术特点 7](#_Toc513740625)

[2.2.2 ZigBee协议栈架构 8](#_Toc513740626)

[2.2.3 ZigBee的网络拓扑 12](#_Toc513740627)

[2.2.4 ZigBee与其他无线标准的技术特点比较 14](#_Toc513740628)

[2.3 本章小结 14](#_Toc513740629)

[第3章 火灾报警系统的数据处理方法 16](#_Toc513740630)

[3.1 相关算法介绍 16](#_Toc513740631)

[3.1.1 遗传算法 16](#_Toc513740632)

[3.1.2 神经网络算法 19](#_Toc513740633)

[3.1.3 模糊逻辑算法 24](#_Toc513740634)

[3.2 信息融合技术 24](#_Toc513740635)

[3.3 信息融合技术在火灾报警系统中的应用 25](#_Toc513740636)

[3.3.1 数据层融合器 26](#_Toc513740637)

[3.3.2 特征层融合器 27](#_Toc513740638)

[3.3.3 决策层融合器 30](#_Toc513740639)

[3.4 本章小结 31](#_Toc513740640)

[第4章 火灾报警系统的总体方案研究 32](#_Toc513740641)

[4.1 系统设计原则 32](#_Toc513740642)

[4.2 选型分析 32](#_Toc513740643)

[4.2.1 处理器选型 32](#_Toc513740644)

[4.2.2 节点传感器选型 34](#_Toc513740645)

[4.3 系统总体方案分析 37](#_Toc513740646)

[4.3.1 系统硬件总体架构 37](#_Toc513740647)

[4.3.2 系统软件总体架构 38](#_Toc513740648)

[4.4 本章小结 39](#_Toc513740649)

[第5章 火灾报警系统的硬件设计 40](#_Toc513740650)

[5.1 总体硬件设计思路 40](#_Toc513740651)

[5.2 CC2530主控制器硬件电路设计 40](#_Toc513740652)

[5.2.1 CC2530最小系统 40](#_Toc513740653)

[5.2.2 电源及复位电路 40](#_Toc513740654)

[5.2.3 系统时钟电路 40](#_Toc513740655)

[5.2.4 串口通讯电路 40](#_Toc513740656)

[5.3 传感器节点硬件电路设计 40](#_Toc513740657)

[5.3.1 传感器节点 40](#_Toc513740658)

[5.3.2 AD转换电路 40](#_Toc513740659)

[5.4本章小结 40](#_Toc513740660)

[第6章 火灾报警系统的软件设计 41](#_Toc513740661)

[6.1 软件开发平台 41](#_Toc513740662)

[6.1.1 IAR嵌入式软件开发平台 41](#_Toc513740663)

[6.1.2 ZStack协议栈开发平台 41](#_Toc513740664)

[6.1.3 Microsoft Visual Studio软件开发平台 42](#_Toc513740665)

[6.2 节点软件设计 42](#_Toc513740666)

[6.2.1 终端数据采集节点的软件设计 42](#_Toc513740667)

[6.2.2 路由节点的软件设计 45](#_Toc513740668)

[6.2.3 协调器节点的软件设计 45](#_Toc513740669)

[6.3 监控PC端软件设计 47](#_Toc513740670)

[6.3.1 串口通讯程序 48](#_Toc513740671)

[6.3.2 阈值检测算法程序 49](#_Toc513740672)

[6.3.3 遗传算法程序 50](#_Toc513740673)

[6.3.4 BP神经网络算法程序 51](#_Toc513740674)

[6.3.5 模糊逻辑算法程序 53](#_Toc513740675)

[6.4 软件整体工作流程 55](#_Toc513740676)

[6.5 本章小结 56](#_Toc513740677)

[第7章 模拟与实验结果分析 57](#_Toc513740678)

[7.1 实验数据采集结果 57](#_Toc513740679)

[7.2 实验数据分析 57](#_Toc513740680)

[7.3 本章小结 57](#_Toc513740681)

[第8章 结论及展望 58](#_Toc513740682)

[8.1 总结 58](#_Toc513740683)

[8.2 展望 58](#_Toc513740684)

[致谢 60](#_Toc513740685)

[参考文献 61](#_Toc513740686)

# 第1章 绪论

## 1.1 论文的研究背景及意义

进入21世纪之后，全世界每年因火灾事故造成直接死亡人数约为7万人，造成的经济损失更是不计其数。火灾是不可避免的自然灾害之一，火灾会对人民群众的生命、财产、健康、安全等造成巨大的损失。

2017年1月至10月，国内共接报火灾约21万起，伤亡人数共计1700余人，经核对，火灾所造成的直接财产损失约为26亿元。据统计，发生较多火灾地点为居民住宅区，约占火灾总数的43.5%，死亡人数为821人次，占死亡人总数的77.1%；另有446人次受伤，占受伤人总数的65.7%。在统计数据中能发现规律，在越发达的国家，因火灾造成的死亡人口与经济损失相对较低，而在发展中国家相对较高。

在火灾发生的时候，一般达不到第一时间采取对应的解决办法的要求，不能防患于未然，往往是等到火灾火势逐渐到达控制不了的地步才进行报警，在此期间消防人员出动也需要一定的时间，一旦火势达到已经不能控制的局面，尤其是发生在楼宇的情况下，事故后果将会十分严重，也会造成非常大的经济损失。如何在火灾将要发生的时候、火灾火势在自控范围内的时候及时发现火灾的存在，并采取相应的措施防止火灾进一步扩大，将火灾扼杀在摇篮中，这种预警方式就显得格外重要了。

就目前市面上的大多数预警系统所面临的问题主要在两个方面体现：一是所采用的是有线方式传输进行连接的，这种预警方式由于需要大范围的安装通信材料，安装较为麻烦，成本高，工程量大等缺点，所以并没有得到广泛的应用，因此研究无线楼宇火灾预警系统对人们的生活有着不可小觑的作用；二是传统的火灾报警器几乎没有对多个传感器进行数据融合算法的处理，导致精度达不到要求，且会产生一定的误报情况，解决其精度，减少误报率是一个成熟的火灾报警器必须具有。

本文针对于这一痛点，本文详细得介绍了如何对火灾探测器应用ZigBee技术和对所采集的数据进行数据融合进行了深入探讨。

## 1.2 国内外研究现状

### 1.2.1 国内研究现状

近年来，国内的火灾预警系统应用技术取得了相应的成就，不尽人意的是，在实际运用过程中，由于技术发展较为晚，导致在通讯、精度等一系列问题上有一定的欠缺，例如：可靠性差而产生的误报；火灾探测器的算法单一导致的精度不高，也在火灾智能报警与善后问题的技术上水平较为落后，主要体现在以下几种情况：

（1）标准规范不统一[1]。

国内已经针对各个地区，针对不同的情况已经颁布了一些消防规范和标准，但是由于不同的设备公司针对自己的产品并没有一个统一的设计标准，导致相同的设备可能换一个环境不能达到相应的效果，有时候则是需要在指定的公司进行相应的设备安装，这样使设备的兼容性大大降低，造成此类情况的主要原因还是由于标准不统一。

（2）报警系统的误报率和漏报率较高[1]。

此处的问题主要是归结于传感器的单方面输入，比如感温火灾探测器，烟雾探测器，这种仅仅是单纯的阈值判断来对火灾进行报警，并不能排除偶然的室内温度过高、由粉尘引起的烟雾浓度增高、无烟雾火灾（例如酒精）的情况，导致火灾的误报率极高，智能程度不高；且因为传统的火灾探测器的灵敏度并不是出于一个最优的状态，过低或过高分别会引起探测器的漏报或者误报。

（3）报警系统的适用范围小[1]。

由于我国的火灾探测器行业起步较晚与相应的政策规范颁布不全面，导致火灾探测器的安装范围仅仅局限于大型商场，人群密集的地方，忽略了人们的起居住所，也是最易发生火灾的地方。

（4）火灾报警系统的构件连接需要改进[1]。

目前我国的大部分火灾探测器都采用的传统总线制连线方式，这种传统的布线方式存在很多缺陷，例如线与线之间存在干扰导致抗干扰能力差、导线发热存在精度不高等，这一系列情况都会导致火灾探测器的可靠程度低，在另一方面上，传统的布线方式耗费的人力物力将会很大，繁琐的布线方式也会对环境要求特别高。如果采用无线传输方式，就不需要考虑以上的问题，且在经济层面上也有了很大的盖申，最主要的是提升了火灾探测器的精度。

（5）火灾报警系统的数据处理有待提高。

在国内，仅只有极个别的火灾探测器在智能化设计上做了相应的文章。总体来说，国内的大部分系统存在的问题在于监测参数少，软件开发不成熟，存在较多BUG，算法不够精确等，传统的火灾探测器并未涉及多传感器数据融合处理，导致判断单一，偶然性大，系统的可靠程度不高。

### 1.2.2 国外研究现状

国外的火灾警报产品相对于国内来说起步较早，技术已经相当成熟了，在很多发达国家已经是实现了具有一系列的火灾防范措施，从火灾预防、报警、扑救，善后等一系列的事项都考虑的想到周到。

例如在美、法、英、日等发达国家，城市火灾自动报警系统方面的钻研有丰硕的阅历。不仅在技术上有很高的成就，并且也投入正常使用多年，也很少存在误报、漏报、迟报等技术问题，这样能将现场损失降到最低。

且在发达国家中，对于智能化火灾报警系统管理方面很规范，相关政府也会成立专门的机构，来维护系统的正常运行，通信机制正常等工作，来保证在发生火灾时不会因为系统的过失造成相应的损失。

## 1.3 论文的主要研究内容

本文的章节分布及主要内容。

第1章是绪论，着重介绍了本文研究的背景以及研究的意义，从国内外火灾探测器的研究发展现状做了简要的分析，更加应征了本文的研究意义，并对本文的架构做了一个简单的梳理。

第2章主要介绍了无线传感器网络相关理论知识，然后简要引出了ZigBee无线通讯技术，为后续的系统总体方案确定打下理论基础。

第3章主要介绍了数据融合的处理方法，着重介绍了遗传算法（Genetic Algorithm）、BP神经网络算法（BP Neural Network Algorithm）与模糊逻辑算法（Fuzzy Logic Algorithm），接着对本系统的数据融合器进行了系统总体方案的规划和提出，为后续的软件设计打下理论基础。

第4章确定了无线楼宇火灾报警系统的总体设计方案，确定硬件与软件的总体设计方案。

第5章主要介绍了系统的硬件方案与连接形式，主要包括控制器（终端与协调器），传感器模块，串口通讯等。

第6章主要确定软件设计的整体方案，重点研究了串口数据的接收，传感器采集的信息融合算法的编程。

第7章对硬件系统进行仿真，经过制造实验火对火灾探测器的终端与协调器的数据传输进行了验证，且验证了经过BP神经网络算法机器学习后的数据处理。

第8章是总结与展望，对本文所研究的系统进行了总结，并对系统提出了一些改善的建议，与总结了可以优化升级扩展的功能。

# 第2章 无线传感器网络技术及ZigBee简介

在第一章中，对传统无线火灾探测器的有线传输方式在实际应用中的弊端做了简要的分析。为了在实际火灾情景中，提高系统的可靠程度，避免造成不必要的后果与财产损失，已有科学依据提出了无线传感器网络的应用。这一技术的提出，将无线传感器网络技术应用到火灾探测系统中，能够很好的解决传统系统的痛点，这样不仅避免了现场布线的繁琐工序，也大大节省了人力财力，且现场监测的节点个数几乎不受限制。而对于一个庞大的无线传感器网络，需要构建一个能够承载这个网络节点的传输协议， ZigBee技术很好的满足了此类需求。

本章将对无线传感器网络技术与ZigBee技术做具体的分析。

## 2.1 无线传感器网络技术

无线传感器网络（Wireless Sensor Network, WSN），它是由大量部署在现场区域内的传感器构成的，传感器之间依靠无线通信的方式，以自组织和多跳网络的方式进行工作，以协作方式来感知、采集、传输和处理网络覆盖区域内监测对象的信息，因此无线通信的方式有效地避免了传统设备的布线工序，并且能够全面覆盖监测区域，准确的采集现场数据并传输。

### 2.1.1 无线传感器网络体系结构

WSN的主要架构由三类节点组成，它们分别是传感器终端节点、路由节点和协调器（主控）节点。由散布在现场监测区域内的大批传感器终端节点采集现场所监测的数据，经过路由节点（中间节点），在终端节点与协调器节点之间构建通信桥梁，负责两类节点的通讯，从而扩大了监测范围与网络规模，而协调器则是为所有的终端节点加入其网络时调配一个独有的网络地址，以方便接受各个网络所监测的数据而不会造成数据紊乱的情况发生。

图2.1为无线传感器网络的典型结构[1]。



图2.1 无线传感器网络的典型结构

Fig.2.1 Wireless Sensor Networks

### 2.1.2 无线传感器网络的特点

WSN具有如下几点特点：

1．传感器节点分布密集，规模庞大。

散布在火灾现场监测区域内的传感器采集节点可部署的规模是可调控的，根据现场环境与精度需求，可以无限扩大规模，从而来保障整个系统的工作时不漏掉任何一个监测点。

2．硬件设计简单。

由于传感器节点网络仅仅需要采集数据，所以运用电路十分简单，且数据处理在一般的MCU芯片的运行速率不能够满足要求，因此监控算法软件一般运行在PC机上。所以传感器节点做的事情极为精简。

3．自组织网络与多跳路由。

移动自组织(Ad Hoc)网络是一个多跳的临时自治系统。美国早在1968建立的ALOHA网络和1973年底提出的分组无线电（Packet Radio）网络，而Ad Hoc的原型是就是以上两组网络。通过自己构建的网络，可在没有基站或者是不需要依赖基站等的情况下进行终端设施之间的彼此通讯。由于各个终端节点间的通讯距离可能只有数百米，达不到真正规模庞大的网络需求，此时多跳网络能够很好的解决，通过中间节点的转发至下一个节点，完成整个网络的数据转发。

## 2.2 ZigBee简介

本系统采用的是短距离无线通讯技术[3]，所谓的短距离无线通信技术定义为收发双方在传输距离较短（通常定义在几十米）的情况下，经过无线进行数据传输的通信技术。

ZigBee技术主要用于短距离且功耗要求较低的设施之间进行数据传输，并对一些典型的有周期性的数据或者是间歇性采集数据等传输也同样适用。

ZigBee网络的设计初衷就是为解决短距离传输网络的构建。ZigBee的体积小，低功耗，成本低廉，且ZigBee可扩展到65535个无线数据传输模块，此网络对于火灾探测系统可以说是十分庞大的一个网络。依据ZigBee的此特点，能够很好得作为无线传感器网络的传输协议。

### 2.2.1 ZigBee技术特点

ZigBee技术具备以下几种特点。

1．低功耗。

由于ZigBee的工作模式定义为休眠模式，再加上ZigBee的发射功率低至1毫瓦，因此ZigBee设备的低功耗是有目共睹的。

2．低成本。

ZigBee的传输协议不含版税，生产成本低，因此ZigBee模块的市场价格已为广大公众所接受。

3．低延时。

ZigBee的模式切换（休眠模式与工作模式间的切换、设备搜素与连接等）时间非常短，通常在60ms的时间内即可完成设备的搜索，激活，接入。因此ZigBee技术已经满足并兼任很多无线控制应用。

4．网络庞大。

一个星形的ZigBee网络可以容纳254个终端和一个协调器。且依据ZigBee技术的16位短地址定义，一个ZigBee理论上最多能够管理的网络节点个数多至65535，而且不仅局限于此，还可以根据64位IEEE地址进行扩展，因而ZigBee可容纳的网络是相当庞大的[3]。

5．可靠程度高。

ZigBee通过预定的策略可以很好地避开发送数据的冲突和竞争。MAC层定义的协议为每次数据收发传输都需要等待对方的确认信息，才能执行后续操作。如果传输过程失败可以重新进行传输。

6．安全性能高。

ZigBee提供了基于循环冗余校验（CRC）的报文完整性检查特性。它支持身份验证，并且采用AES-128加密算法，每个应用程序均可以灵活地确定其安全属性。

### 2.2.2 ZigBee协议栈架构

ZigBee协议的创建初期时使用的是七层协议，但由于初衷是为了构建出低功耗、高速率的结构，后来将其结构简化为四层协议，分别为物理层(PHY)、媒体访问控制层(MAC)、网络层(NWK)及应用层(APL)。Zigbee的协议栈架构如图2.2所示。



图2.2 ZigBee协议栈架构

Fig.2.2 The architecture of ZigBee protocol stack

图2.3为开放系统互连参考模型 (Open System Interconnect 简称OSI）与ZigBee协议栈的对比。



图2.3 OSI模型与ZigBee协议栈结构

Fig.2.3 OSI Model and ZigBee protocol stack structure diagram

ZigBee协议栈的每层都有自身独有的构造模型，每层所负责的主要任务也各不相同，下面对四层结构进行简要的介绍。

1．物理层(PHY)

物理层是整个协议层中最基本的部分，它指定了所使用的频带，以及诸如编码、调制、扩频和频率调制等无线传输技术；物理层维护了一个自身的数据库，管理从物理信道上所收发的数据。如图2.4所示为物理层结构模型。



图2.4 物理层结构模型

Fig.2.4 The PHY layer structure diagram

2．媒体访问控制层(MAC)

MAC层的主要作用定义了无线通信通道的访问控制机制，即它规定每个设备依次根据所建立的规则使用信道，从而不至于产生访问冲突的情况。在网络中能够协调多个设备进行信道资源访问。

IEEE 803.4.4的MAC层提供了信道接入（CSMA-CA和TDMA）、本地网络建立维护和同步、安全、可靠通信等[2]。

MAC层同样维护了一个属于该层相关的数据库（主要是子层协议的相关状态）。



图2.5 MAC层结构模型

Fig.2.5 The MAC layer structure diagram

3．网络层(NWK)

ZigBee最主要的部分在于网络层，核心功能是路由、寻址、网络的形成与维护。

网络层主要负责网络的管理，例如网络发现与网络形成、加入网络和离开网络、节点的增加与删除。

网络层支持多种路由算法，兼容多跳路由，可以实现星型（Star）拓扑结构、树型（Cluster-Tree）拓扑结构和网状（Mesh）拓扑结构等不同的网络拓扑结构。（网络拓扑结构在下一小节中详细讲解）。



图2.6 网络层结构模型

Fig.2.6 The NTK layer structure diagram

4．应用层(APL)

除了用户定义的应用对象之外，主要还包括三个部分：一是应用支持子层（APS）；二是ZigBee设备对象（ZDO）；三是应用框架等部分。

应用支持子层包括数据实体（APSDE）和管理实体（APME），用于向上层提供数据传输和管理配置服务。应用支持子层将应用程序框架与网络层隔离，并为应用程序的操作提供支持。

ZDO的任务包括：协议栈初始化（如APS和NWK层的初始化和安全服务）；以及通过ZDO执行设备发现、服务发现、安全管理、网络管理和绑定管理的应用。



图2.7 应用支持子层结构模型

Fig.2.7 The APL layer structure diagram

### 2.2.3 ZigBee的网络拓扑

在ZigBee网络中，根据设备完成的任务区分实际上是由两种设备组成的：一种是全功能设备（Full Function Device, FFD），另一种则是精简功能设备（Reduce Function Device, RFD）。对于两种设备的介绍如下。

1．全功能设备。

该设备领有完全的通信协议，在网络中能扮演三种角色，分别是协调器（Coordinator）节点、路由器（Router）节点与普通（Device）节点。

2．精简功能设备。

设计初衷为实现最简单的功能，拥有最简单的通讯协议，在网络中仅仅只能扮演普通节点的角色。

FFD与RFD设备功能如图2.8所示。

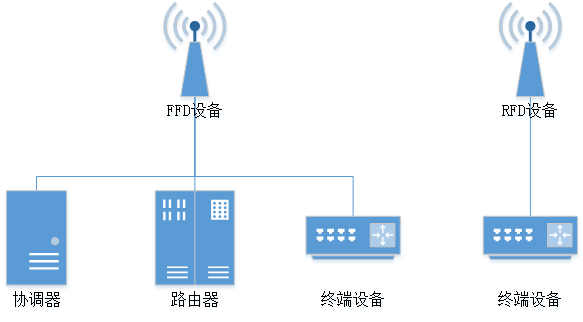


图2.8 FFD与RFD的功能

Fig.2.8 The function of FFD and RFD

一个完全的ZigBee网络中包含三类节点：协调器、路由器和终端设备。而网络中的协调器只能是全功能设备。在个人局域网（Personal Area Network, PAN）中必须领有至少一个协调器，这是建立一个完整的网络的基础，而其在网络中的作用负责PAN网络的初始化和地址分配，管理网络连接等。

在网络结构中，每个终端设施都会被调配一个独有的64位IEEE地址用来直接询问PAN从而进行通信，当协调器加入网络后，每个终端设施经过64位IEEE地址连接，在连接成功之后，在系统分配PAN标识符之后，协调器分配给设备的16位短地址可用于此时的通信[1]。

以下介绍几种常见的网络拓扑结构，图2.9为星型（Star）拓扑结构，图2.10为树型（Cluster-Tree）拓扑结构，图2.11为网状（Mesh）拓扑结构。

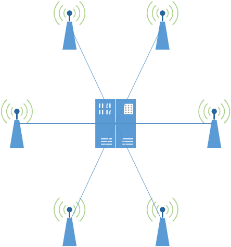


图2.9 星型结构

Fig.2.9 Star Topology

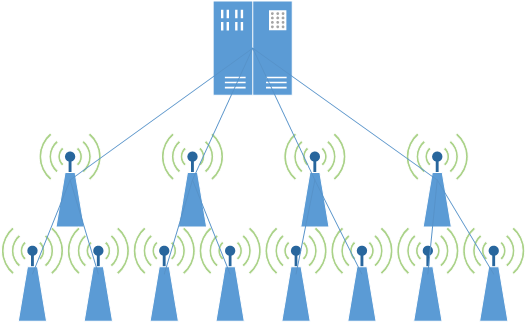
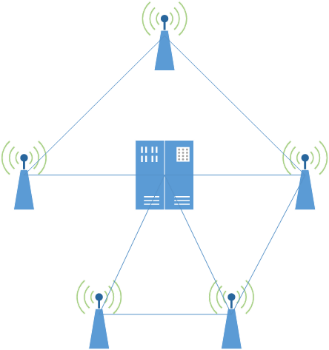
 

图2.10 树形结构 图2.11 网状结构

Fig.2.10 Cluster-Tree Topology Fig.2.11 Mesh Topology

由图我们可以清晰看出，星型结构包括一个协调器（主）与若干个路由节点或终端节点（从），该结构只能通过主设备访问从设备，而从设备间不能直接访问通信，只能通过访问主设备再访问其需要的从设备，这种网络成为单跳网络。

而经过不同的初始化操作可以实现不同的网络结构，如树形结构和网状结构，这两种网络结构为多跳网络。其与星型结构相同的是都由主设备与从设备组成，不同的是在于，后两种结构可以通过从设备访问从设备，树形结构与网状结构的架构虽然比星型结构复杂，但是通讯机制灵活，网络稳定性高。

### 2.2.4 ZigBee与其他无线标准的技术特点比较

在无线通信的领域，不仅仅只有ZigBee通信技术，还有其他很多无线通信技术，关于任何一种的通讯技术都有其不同的适用范畴及领域。表2.1为ZigBee与其他无线技术的比较，根据结果表明，ZigBee技术较其他技术更适用与本系统。

表2.1 ZigBee与其他无线标准的比较

Table 2.1 ZigBee compared with several other wireless standards

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 规范 | 工作频段 | 速率（Kbps） | 功耗（mW） | 连接设备数 | 主要用途 |
| ZiigBee | 868915MHz,  2.4GHz | 20,40,250 | 1~3mW | 216~264 | 家庭网络，传感器网络 |
| 红外 | 820hm | 0.521,4,16 | 4~8 mW | 2 | 可见范围内数据传输 |
| HomeRF | 2.4GHZ | 1,2 | 100 mW | 127 | 家庭无线局域网 |
| 蓝牙 | 2.4GHz | 1,2,3 | 1~100 mW | 7 | 个人网络 |
| RFID | 5.8GHz | 0.212 | 无需供电 | 2 | 超市，物流管理 |

## 2.3 本章小结

本章主要通过无线传感器网络的应用来引出ZigBee技术，此技术在后文为系统的总体方案设计提供了相应的理论基础。

首先对无线传感器网络的原理进行了分析，根据其庞大的网络节点需求，根据分析，选用了ZigBee技术作为本系统的通信协议。接着重点剖析ZigBee技术的协议栈与其网络拓扑结构，依据其应用范畴与其他无线技术的性能比较，最终决定ZigBee技术的各方面功能较其他技术更加适用于本系统。

# 第3章 火灾报警系统的数据处理方法

针对于传统的单传感器火灾探测器来讲，传感器的数据处理过于单一，而对于一个火灾现场来说，并不仅仅局限于阈值的“与”或“非”的问题，而且需要经过多方面的参考。由于在实际生活中，非火灾信号和电磁噪声可能也会导致传感器有一定的输出值，在特定的情况下，这些输出可能大于预设阈值，导致系统产生误报，此时多传感器数据融合就显得额外重要。在人工智能技术发展下，将人工智能技术运用到火灾探测器中是一个必然趋势，机器学习方法能使系统更加稳定，将传感器数据进行融合，是系统精度更高，鲁棒性强。本章将从算法介绍与火灾报警系统的数据融合的应用进行相关的介绍。

## 3.1 相关算法介绍

此部分主要介绍遗传算法、神经网络算法和模糊逻辑算法，三个算法将作为火灾报警系统数据融合器设计的理论基础。

### 3.1.1 遗传算法

3.1.1.1 遗传算法简介

遗传算法（Genetic Algorithm）也是一种进化算法。这是一个近似模拟达尔文的生物进化理论的“自然选择和适者生存”过程的计算模型，它模拟自然演化过程并搜寻最优解。

利用“袋鼠跳”的例子可以很好解释遗传算法。假设在珠穆朗玛峰下生活着一群袋鼠，由于山下的猎人捕杀行为，导致袋鼠被破往山上迁移，种群中强大的个体迁移速度比其余较弱的个体快，也就避免了暂时的被猎杀，逐渐弱势群体被淘汰，经过种群繁殖，基因变异，优良基因的传承，依据“生存竞争，适者生存”规律，种群最终必然是一个最优种群。

3.1.1.2 遗传算法运行步骤及组成

在遗传算法中，由4个部分组成：种群（Population）、个体（Individual）、基因（Gene）、染色（Chromosome）。

在遗传算法中，有着相同的道理，根据“生存竞争，适者生存”法则，环境适应性高的个体在种群中参与育种的机会更多，后代的后代将是最佳的解决方案。相反，适应度低的个体后代就会越来越少，在其中渐渐的被种群淘汰。个体在繁衍后代时，会发生交叉( Crossover ) ，变异 ( Mutation ) ，在此过程中，适应度( Fitness )低的个体会被淘汰，而适应度高的个体会越来越多。所谓适应度，实则为针对项目所描述的问题，根据既定表达式所计算出的。

遗传算法的算法核心思想是通过随机值产生函数初始化一个种群，对每一个个体的特征值进行编码（编码形式有二进制编码、实数编码、符号编码等），通过选择（通常为轮盘赌法）来完成“天择”，被选择的个体将会在种群中繁衍后代，完成基因的交叉或变异操作来提升个体的适应度，当种群进化迭代次数达到预定阈值时，算法终止，并在种群中寻找适应度最大的个体。

遗传算法的基本运行步骤如图3.1所示。



图3.1 遗传算法的基本运行步骤

Fig.3.1 Basic operation steps of genetic algorithm

遗传算法主要包括以下几个核心算法子模块：遗传编解码、适应度计算、遗传操作算子[5]。

1．遗传编码。

遗传编码是遗传算法进行中的前提。遗传算法工作时，首先会将特征值编码成可用于后续遗传操作的形式，在进行遗传操作结束后，也有与之相对应的解码函数来完成特征值的还原，以便于适应度的计算。迄今为止，编码形式有很多，例如二进制编码、实数编码、符号编码等。但二进制编码是最常用的编码形式，可直接被机器操作，理解简单。

2．适应度计算。

为了满足优胜劣汰的要求，会对每个个体进行适应度的计算用以评判是否可以继续存在于种群之中。

3．选择算子。

常用的选择算子有以下几种：排序选择、比例选择、轮盘赌法等。本系统采用轮盘赌法进行种群的一个选择保留。进行此操作的目的是模拟自然选择的过程，看似随机，实则必定会决定一个发展方向。此处选择算子有很多，常见的的轮盘赌法算法如图3.2所示，轮盘转起来，指针指向的区域则会被选中。



图3.2 选择算子的工作流程

Fig.3.2 Select operator workflow

4．交叉算子。

交叉操作是模拟生物界中繁衍时的基因重组的过程，将父代的两部分基因进行交换重组，产生新的个体，以继承父类的优良基因。交叉操作包括单点交叉、多点交叉、算数交叉等多种交叉形式[5]。

遗传操作交叉算子的工作流程如图3.3所示。



图3.3 交叉算子的工作流程

Fig.3.3 Crossover operator workflow

5．变异算子。

变异算子是模拟生物进化中基因突变的过程。在遗传算法中，当某个个体的某段特定的二进制编码位发生改变时，个体的适应度也会随之发生改变，从而可以在变异的过程中，保留优良个体的基因，从而使种群最终接近于最优解。

遗传操作变异算子的工作流程如图3.4所示。



图3.4 变异算子的工作流程

Fig.3.4 Workflow of mutation operator

### 3.1.2 神经网络算法

3.1.1.1 人工神经元模型

众所周知，人的大脑可以用来处理身体部位发送的复杂信息，根据经验和逻辑来推测行为并作出相应决策，并能对各种信息进行分类归纳，这就是生物神经上常说的“自学习”过程[6]。

如果我们将人脑的这种“自学习”过程运用到实际生活中，让机器自己进行学习和探索，模拟人脑建立对数据的信息处理的网络，于是便引出了人工神经网络。人工神经网络属于一种自我学习的系统[6]，其单个人工神经元模型如图3.5所示，通过机器自学习过程，建立一个满足要求的算法，函数，或者是一种规则。机器学习的过程其实很简单，是根据输入值经过前馈输出值与期望输出值进行对比然后进行反馈不断修正网络的阈值和权重，是一个自适应的过程，当前馈输出与期望输出之间的误差达到既定的要求时，则机器学习任务完成，人工神经网络建立完毕。



图3.5 人工神经元模型

Fig.3.5 Artificial neuron model

由图3.5可以得出，该单个人工神经元模型不仅只有一个输入信号，n个输入信号经过与权值的加权以及激活函数的处理，与阈值的矫正，最终输出决策值。该神经元各部分的关系如公式3-1和公式3-2所示。

 （3-1）

 （3-2）

 —— 输入信号；

 —— 处理单元阈值；

 —— 输入层i到j的权值。

f(x)为激活函数。一般情况下激活函数采用Sigmoid函数。Sigmoid函数公式如3-3所示。

 （3-3）

对于人工神经网络算法，人们已经提出十余种模型，每种模型都要其适合的场景。目前应用最为广泛的神经网络模型为前向神经网络模型，数据经过前向网络，最终达到输出层，在输入层与输出层直接可以有很多隐藏层，这样就构成了一个由输入层、隐藏层与输出层组成的一个复杂的神经网络。BP神经网络算法是前向神经网络算法的典型应用实例，以下着重介绍BP神经网络算法。

3.1.1.2 BP神经网络算法

BP（BackProgation）神经网络算法属于前向神经网络的代表[7]，其组成结构如图3.6所示。BP神经网络不同于简单神经网络的是其输入学习样本后，将得到的输出值与期望值进行比较可以得到相应的误差值，根据误差值可以推算出影响误差产生的原因，主要是权值与阈值导致的，经过反馈处理来不断的修正误差与权值，直到输出结果满足预期的要求则终止学习。



图3.6 BP神经网络算法结构图

Fig.3.6 BP neural network algorithm structure diagram

BP神经网络算法同前向神经网络算法一样，拥有三层结果，其中输入层与输出层仅有一层，且节点个数根据训练样本和输出期望而定，而隐藏层数量与隐藏层节点数不限，根据用户的需要自己设定。理论上来讲，如果训练数据与隐藏层的层数与节点数足够大，系统能够模拟出任何方程。一般来讲，节点数可以根据以下经验公式来进行隐藏层节点数的计算：

 （3-4）

 —— 隐藏层节点数；

 —— 输入层节点数；

 —— 输出层节点数；

 —— 1~10之间的整数。

当隐藏层节点数确定之后，整个网络基本上就确定。以下介绍BP神经网络算法的工作流程。

当输入的样本数据通过公式3-1的计算与3-2的激活，输入层的输出数据作为隐藏层的值数据，然后将隐藏层的值同样经过公式3-1的计算与3-2的激活，将隐藏层的输出值作为输出层的值。

此时仅仅完成的前向输出工作，BP神经网络训练方法其独特之处在于会通过输出层的值与期望值之间的误差对阈值与权值进行一个修正，最终使系统的误差在许可的范围之内。

根据误差值（Error）反向传送进行修正步骤如下。

首先计算输出层的输出与期望值之间的误差：

 （3-5）

 —— 输出层误差值；

 —— 输出层实际值；

 —— 输出层期望值。

其次计算隐藏层的误差：

 （3-6）

 —— 隐藏层误差值；

 —— 输出层误差值；

 —— 输出层实际值。

误差值计算完成之后，需要对网络各层之间的权重和节点固有阈值进行更新：

 （3-7）

 （3-8）

 （3-9）

 （3-10）

 —— i节点到j节点（输入层到隐藏层或隐藏层到输出层）的权重更新值；

 —— 当前节点的阈值更新值；

 —— 神经网络学习速率；

 —— 当前节点的误差值；

 —— 输出层的实际输出值；

经过以上步骤反复经过训练，最终达到实际输出与期望值输出的误差在预设误差内即可完成网络训练，并确定网络各个参数（权值与阈值）。

图3.7为BP神经网络算法流程图。



图3.7 BP神经网络算法流程图

Fig.3.7 BP neural network algorithm flow chart

### 3.1.3 模糊逻辑算法

模糊逻辑是建立在多值逻辑基础上，模拟人脑思维的一个模式，来解决一个需要多方位判断的模糊规则，实行模糊综合判断。模糊逻辑处理流程如图3.8所示.



图3.8 模糊化流程

Fig.3.8 Folw of fuzzy

模糊逻辑算法需要对输入的特征信息进行一个量化，确定他们的论域。在无特殊情况下，论域U上的模糊集分为七级，分别是负大（NB）、负中（NM）、负小（NS）、零（ZD）、正小（PS）、正中（PM）、正大（PB）。根据隶属度函数（例如三角隶属度函数，正太隶属度函数等）确定每一个特征信息属于哪一个层次。

判断特征值所属论域后，则可建立控制规则表。举例说明，当A成立，B成立，则C成立；或者是当A是PB，B是NB时，C不成立。规则的表现形式根据用户的需求，可以有多条，数目由输入和量化等级决定。

当特征值经过隶属度函数处理之后，最终都会根据既定的表来查找对应的决策值。

## 3.2 信息融合技术

信息融合（Data Fusion）技术也称为多传感器信息融合技术或数据融合技术[3]，是一个对从单个和多个信息源获取的数据和信息进行关联、相关和综合，以获得精确的位置和身份估计，以及对态势和威胁及其重要程度进行全面及时评估的信息处理过程。在本系统中，根据集成传感器模块所采集到的不同的数据信息，以既定的准则加以剖析，将多种数据最终融合在一起，输出一个最终的决策值，这样克服了传统单传感器的缺点，使系统更优越，性能更高。

传感器数据融合技术关系如图3.9所示，可以分为三层：数据层融合、特征层融合与决策层融合。



图3.9 传感器数据融合技术层次关系

Fig.3.9 Sensor Data Fusion Technology Hierarchy

1．数据层融合。

该层数据处理一般在传感器所采集到的原始数据层。主要负责的任务是对数据进行预处理，主要的数学方法是：加权平均法、卡尔曼滤波、贝叶斯估计、参数估计法等，与信号处理有一定的相似性。数据层在信息融合技术层次里的等级属于最低级，该层任务是尽可能多得保存现场真实数据，方便后面两层的数据处理。

2．特征层融合。

对现场所采集的原始数据进行一定的预处理，提取各个数据的特征后，针对这些特征进行分析，由此可见，相比数据层融合，该层的层次高于数据层，同时特征层也显得更加抽象。特征层融合分为两大类：一是特征层目标状态数据融合；二是特征层目标特性融合。前者主要用于多传感器目标跟踪领路，融合系统首先对传感器数据进行预处理以完成数据校准，然后实现参数相关的状态向量估计[3]。后者主要是特征层联合识别，融合方法是模式识别的相应技术[3]。特征层融合后的数据较原始数据相比进行了压缩，同时保留原始数据的有用信息，便于处理时的提出，特征层融合后的数据，更有利决策层融合数据的需要。

3．决策层融合。

决策层融合经过前两层融合所提取的特征向量值，根据既定准则和决策的可信度做出最终的判断。决策层是最高的层次，结合特征层的数据输出，对问题做出最终判断。

## 3.3 信息融合技术在火灾报警系统中的应用

根据国内目前火灾探测器的发展情况，，传统的火灾探测器对现场的数据采集处理十分简单，没有复杂的处理逻辑，导致系统的可靠程度不高，缺陷较为明显。本系统所研究的是一个高性能，高可靠性，高精度的火灾探测器，因此需要建立一个基于信息融合的火灾报警系统。

据实验结果表明，大部分的现场火灾发生时，温度、湿度、O2浓度、CO浓度、烟雾浓度等均会发生变法，但在变化过程中，温度，CO浓度与烟雾浓度的变化速率显著，因此这三个参数是最能反映火灾发生的检测对象，依据信息融合理论基础，本系统的火灾探测融合过程依旧采用三层结果，分别是数据层融合、特征层融合和决策层融合，设计了如图3.10所示复合火灾探测器。

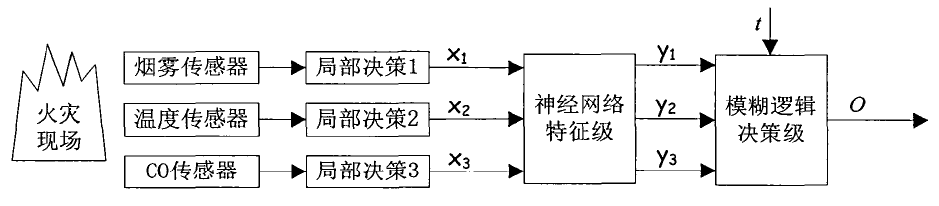


图3.10 复合火灾探测器

Fig.3.10 Composite fire detector

以下对本火灾报警系统的数据融合器进行分层次介绍。

### 3.3.1 数据层融合器

数据层融合发生在集成传感器，如图3.10所示为局部决策。传感器采集现场数据后，经过简单的判断处理，初步确定是否有火灾发生的可能性，如果有发生火灾的可能性，则将数据保留，进行下一步处理，反之，系统将继续采集工作不做任何处理。

本系统在数据层融合器中主要运用到的处理方法为阈值检测算法。火灾探测器从现场所采集到的原始数据并不是直接用于后期处理判断，在当前层会将数据进行预处理。阈值检测法为最为直观的预处理算法，若阈值多次超过预设值时，则判定火灾有可能发生，然后将该组数据传入下一层进行进一步融合。

阈值检测算法如公式3-11与公式3-12所示：

 （3-11）

—— 单位阶跃函数；

—— 第i个传感器的阈值；

—— 第i个传感器的第k次采集到的数据；

—— 第i个传感器第k次的决策结果。

 （3-12）

m —— 为采集次数。

当（）任意一个输出为1时，说明火灾探测器的复合传感器模块中所采集的三个信号（温度，CO浓度，烟雾浓度）有超过预设阈值，此时输出结果是可能发生火灾，此时保留数据，向下一层传入进行决策判定。

### 3.3.2 特征层融合器

本系统采用基于遗传算法优化的BP神经网络算法作为火灾报警系统的特征层融合器。遗传算法优化的BP神经网络算法流程图如图3.11所示。

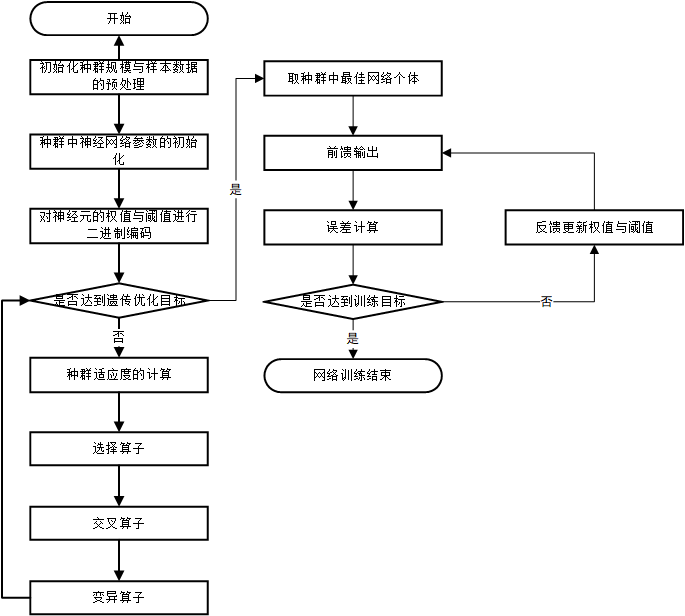


图3.11 基于遗传算法的BP神经网络算法流程图

Fig.3.11 BP neural network algorithm flow chart based on genetic algorithm

将经过遗传算法选择后的最优网络作为训练网络的初始值，将数据层融合器的火灾特征参数进行归一化处理。如图3.10所示为神经网络特征级。分别用来表示温度传感器输出参数，烟雾传感器输出参数与CO传感器输出参数。三类参数输入到特征层后，经过训练后的网络进行数据的计算和处理，判断火灾处于明火，阴火还是非火灾状态，其结果分别用来表示。特征层融合器系统结构如图3.12所示。

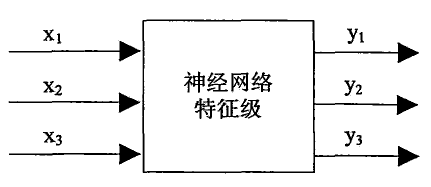


图3.12 特征层融合器系统结

Fig.3.12 Feature Layer Fusion System Node

根据3.1.2节所介绍的BP神经网络算法，本系统针对此算法确定了系统网络的设定。由于隐藏层的节点数对系统的性能影响较大，如果网络过于庞大，导致系统的收敛速度过慢，学习时间过长；如果网络过于简单，可能会导致系统受干扰比较大，使系统的精度不高。经过多次实验和理论依据，本系统设定为1层隐藏层，隐藏层的节点数为7个。火灾报警系统的BP神经网络如图3.13所示。



图3.13 火灾报警系统的BP神经网络

Fig.3.13 BP Neural network of fire alarm system

对于不同环境的火灾有不同输输入数据，火灾的数据量就十分庞大，此时只针对重要的情况进行数据训练，其他情况我们并不关心。本系统数据融合训练数据选择中国明火SH4和标准阴燃火SH1的相关数据。标准数据如表3-1所示。

表3-1 BP神经网络算法训练数据

Table.6.1 BP Neural network algorithm training data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 温度 | 烟雾 | CO气体 | 非火灾概率 | 阴燃火概率 | 明火概率 |
| 1 | 1.000 | 0.200 | .0900 | 0.000 | 0.050 | 0.950 |
| 2 | 0.900 | 0.230 | 1.000 | 0.000 | 0.050 | 0.950 |
| 3 | 1.000 | 0.220 | 0.850 | 0.020 | 0.080 | 0.900 |
| 4 | 0.900 | 0.220 | 1.000 | 0.020 | 0.080 | 0.900 |
| 5 | 0.875 | 0.180 | 0.350 | 0.050 | 0.100 | 0.850 |
| 6 | 0.950 | 0.200 | 0.750 | 0.030 | 0.100 | 0.850 |
| 7 | 0.900 | 0.130 | 0.200 | 0.100 | 0.100 | 0.800 |
| 8 | 0.925 | 0.180 | 0.250 | 0.100 | 0.100 | 0.800 |
| 9 | 0.600 | 1.000 | 0.750 | 0.700 | 0.100 | 0.200 |
| 10 | 1.000 | 1.000 | 0.750 | 0.700 | 0.100 | 0.200 |
| 11 | 0.450 | 0.150 | 0.500 | 0.050 | 0.800 | 0.150 |
| 12 | 0.450 | 0.150 | 0.700 | 0.050 | 0.850 | 0.100 |
| 13 | 0.450 | 0.180 | 0.750 | 0.050 | 0.850 | 0.100 |
| 14 | 0.600 | 0.700 | 0.450 | 0.800 | 0.100 | 0.100 |
| 15 | 1.000 | 0.420 | 0.700 | 0.850 | 0.050 | 0.100 |
| 16 | 0.500 | 0.250 | 0.700 | 0.020 | 0.900 | 0.080 |
| 17 | 0.525 | 0.500 | 0.750 | 0.020 | 0.900 | 0.080 |
| 18 | 0.550 | 0.830 | 0.750 | 0.000 | 0.950 | 0.050 |
| 19 | 0.375 | 0.370 | 0.450 | 0.850 | 0.100 | 0.050 |
| 20 | 0.625 | 0.200 | 0.500 | 0.850 | 0.100 | 0.050 |

在输入训练数据准备完毕之后，则可以根据3.1.2节所介绍的BP神经网络的工作流程进行网络的一个机器学习的过程，最终启动属于本火灾报警系统的神经网络（具体软件实现在第六章介绍）。

### 3.3.3 决策层融合器

决策层融合器主要是应用模糊逻辑技术，在本章第3.1.3节已经对模糊逻辑技术的理论做了简要的介绍，本小节针对其在火灾报警系统中的应用再次做一个详细的介绍。

将明火概率与阴燃火概率进行量化，确认其所属论域与模糊等级。非火灾概率不是本系统所关心的概率，至此，我们需要考虑系统的智能化程度，增添一个控制规则为危险信号持续时间，同样对时间信号进行量化。

图3.14所示为决策层融合器结构。



图3.14 决策层融合器结构

Fig.3.14 Decision layer fuser structure

y1 —— 明火概率；

y2 —— 阴燃火概率；

T —— 危险信号持续时间；

u —— 决策层输出结果。

在决策层，主要任务是将三个输入数据利用正态分布隶属度函数进行模糊化成为PB、PM、PS三个等级（由于概率不可能为负，零可以近似为极小值）。正态分布隶属度函数如公式3-13所示。

 （3-13）

式中b根据经验要求取值0.2，a根据模糊规则中PS、PB、PM分别取值0,0.5,1；

本系统我们做一个特殊处理，将火灾持续时间模糊化为两个等级（PB与PS），且阴火火灾概率与明火火灾概率不可能同为PB或者PM与PB同时存在，因此，火灾模糊逻辑规则如表3.2所示。

表3.2 模糊推理规则

Table.3.2 Fuzzy Inference Rules

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | T | u |
| PS | PS | PS | PS |
| PS | PM | PS | PS |
| PS | PB | PS | PS |
| PS | PS | PB | PS |
| PS | PM | PB | PB |
| PS | PB | PB | PB |
| PM | PS | PS | PS |
| PM | PM | PS | PB |
| PM | PS | PB | PB |
| PM | PM | PB | PB |
| PB | PS | PS | PS |
| PB | PS | PB | PB |

## 3.4 本章小结

本章首先着重介绍了三大算法，为信息融合技术在火灾报警系统中的应用打下了理论基础；其次简要介绍信息融合技术；根据系统的要求与前面的理论基础，最后对系统的数据融合器进行了相关设计。依据信息融合技术的理论，本系统的火灾数据融合器一样分为数据层融合器、特征层融合器和决策层融合器三个部分。针对每层完成的任务各不相同，数据层也是最低级的一层，通过阈值算法对采集的原始数据进行预处理与预判断；特征层则经过神经网络算法来自行运算，输出相应的决策值；决策层依赖模糊逻辑算法与相应的模糊规则对明火，阴火还是非火灾概率进行最终的决策判断，输出最终的火灾情况。

本章利用数据融合的处理算法，将传感器数据进行融合，能够解决传统设备的不足之处，改善系统的性能和可靠性。

# 第4章 火灾报警系统的总体方案研究

## 4.1 系统设计原则

从系统的应用场景出发，火灾报警系统要求时效性，可靠性高，稳定强度高，抗干扰能力强。

系统的硬件在设计过程中，需要对每一个传感器进行相应的选型，选型依据是传感器的精度和适用场景，同时成本因素也应考虑进来。为后期系统的优化升级与发展，系统需要预留一些扩展的I/O口供设备的升级。

系统的软件设计则是分模块设计，将每个模块结构化，然后统一联调，既有编程化思想，提升开发效率，也对代码复用有一定的帮助。

## 4.2 选型分析

### 4.2.1 处理器选型

4.2.1.1处理器选型

目前市面上有很多ZigBee技术的无线芯片与SOC产品，如何选用最适合本系统的芯片则显得额外重要。如表4.1所示为常见ZigBee芯片的性能比较。

表4.1 常见ZigBee芯片性能比较

Table.4.1 Common ZigBee chip performance comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 各芯片型号 | JN5148 | CC2530 | MC13192 | EM260 |
| 工作频率（Hz） | 2.4G | 2.4G | 2.4G | 2.4G |
| 无线速率（Kbit/s） | 250 | 250 | 250 | 250 |
| 发射功率（dBm） | +2.5 | +4.5 | +3.6 | +3 |
| 接收灵敏度（dBm） | -97 | -97 | -92 | -97 |
| 最大发射电流（mA） | 15 | 35 | 35 | 37.5 |
| 最大接收电流（mA） | 18 | 24 | 42 | 41.5 |
| 休眠电流（uA） | 0.2 | 1 | 1 | 1 |
| 工作电压范围（V） | 2.0~3.6 | 2.0~3.6 | 2.0~3.4 | 2.1~3.6 |
| 硬件开放度 | 不开放 | 开放 | 开放 | 开放 |

除了要满足火灾数据采集的各方面要求，还需要从芯片的可靠性，性价比与系统的开放程度多方面入手，综合考虑最优方案[7]。

经过对表4.1中各芯片的综合对比，且考虑性价比最高的选择，本系统采用TI（Texas Instruments, 德州仪器）CC2530芯片作为本系统的处理器。

4.2.1.2 CC2530芯片简介

如图4.1为CC2530芯片的外围电路图。

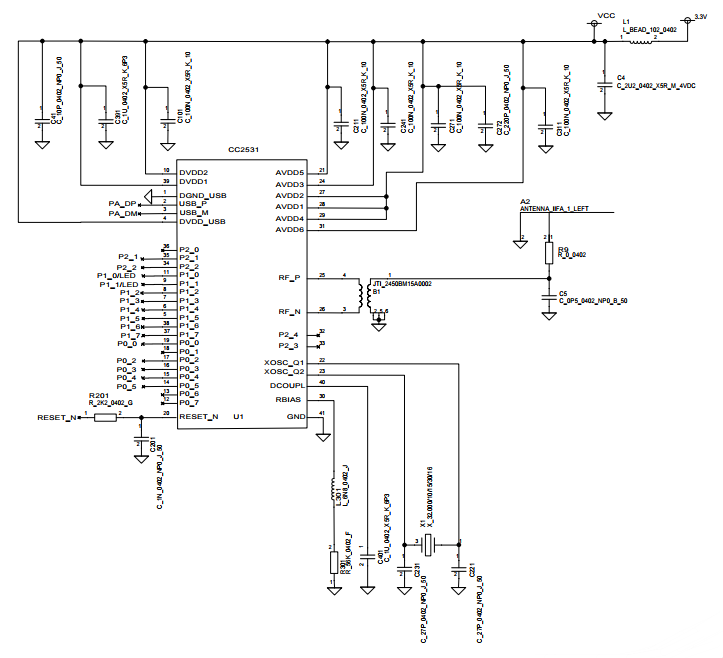


图4.1 CC2530芯片外围电路图

Fig.4.1 CC2530 chip peripheral circuit diagram

在CC2530芯片内部详细参数与系统特性如图4.2所示。



系统组成

系统特性

图4.2 CC2530芯片内部详细参数

Fig.4.2 CC2530 chip internal detailed parameters

此款芯片很好的兼容了TI公司的Z-Stack协议栈，与开发工具能很好的结合起来，使得用户在开发过程中能更加方便。

### 4.2.2 节点传感器选型

众所周知，对于一个优良的测量系统，传感器的性能往往决定了系统的性能，因此，对于传感器的选型是尤为重要的。在前面第三章中所介绍的数据融合处理方法提到，系统采用温度、烟雾浓度与CO浓度监测对象。下面分别对三个参数所用到的传感器进行分析。

1．温度传感器。

本系统的温度传感器采用DS18B20。DS18B20具有体积小、抗干扰能力强和精度高等特点，且为数字信号输出，不需要相应的的AD转换器既可以直接输入到微处理器内。

DS18B20的测温范围为－55℃～+125℃，且测温误差在1℃左右，完全满足火灾测量的温度需求。另该传感器的接线方式简单，具有独特的单线接口方式，在与微处理器连接时仅需要一条口线即可实现微处理器与DS18B20的双向通讯，且测量结果不需要经后续电路的处理，直接为9~12位数字量方式串行传送。

DS18B20引脚图4.3所示。



图4.3 DS18B20引脚图

Fig.4.3 DS18B20 Pin Diagram

2．烟雾传感器。

本系统使用的烟雾传感器为MQ-2气敏传感器。

该器件的阻值与气体浓度之间一般呈非线性关系，但在浓度域的气体检测时可近似认为是线性的。因此，在一定的浓度范围内的检测是有效的，它不适合于仪表之类精确浓度的测定，但是完全可以满足本系统对烟雾浓度采集的要求。

MQ-2型传感器对天然气、液化石油气等烟雾有很高的灵敏度，尤其对烷类烟雾更为敏感。其工作原理主要是源于其内部组成材料为二氧化锡半导体气敏材料，当火灾发生时，空气中的烟雾与二氧化锡结合会导致电阻值发生改变，使输出的模拟信号发生变化，从而可以监测空气中烟雾浓度。

MQ-2烟雾传感器的特性如图4.4所示。

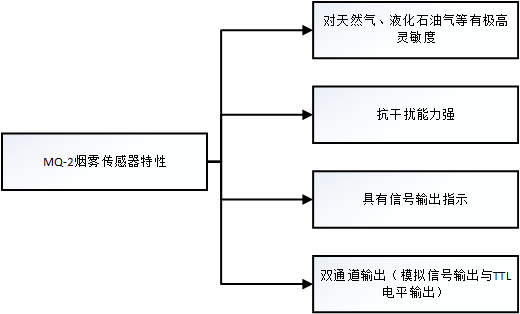


图4.4 MQ-2烟雾传感器的特性

Fig.4.4 MQ-2 Smoke Sensor Features

MQ-2烟雾传感器应用电路图如图4.5所示。

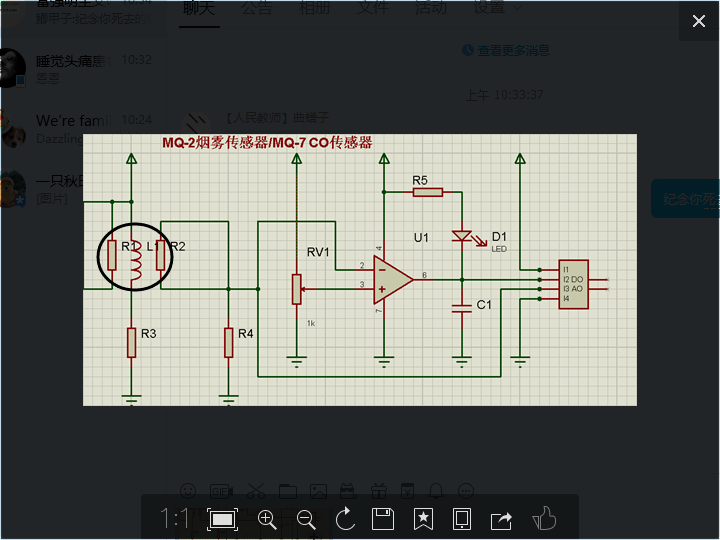


图4.5 MQ-2烟雾传感器应用电路图

Fig.4.5 MQ-2 smoke sensor application circuit diagram

3．CO传感器。

本系统选用的CO传感器为MQ系列的MQ-7 CO传感器，此传感器的工作模式如同上所述的MQ-2烟雾传感器，传感器的电导率随空气中一氧化碳气体浓度增加而增大，从而随着空气中的CO浓度升高会导致模拟信号的升高，因此可以简单的监测空气中的CO浓度变化。且MQ-7的应用电路图与MQ-2类似，如图4.5所示。

图4.6为MQ-7的相关特性。

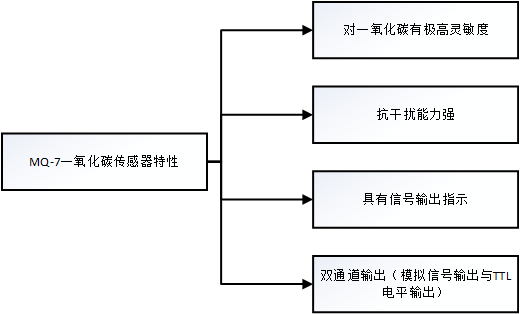


图4.6 MQ-7烟雾传感器的特性

Fig.4.6 MQ-7 Smoke Sensor Features

## 4.3 系统总体方案分析

### 4.3.1 系统硬件总体架构

楼宇火灾报警系统的硬件总体架构包含五个部分：火灾探测复合传感器模块、发射无线通信节点、路由器、接收无线通信节点与主控制器。图4.7为系统硬件总体架构。



图4.7 系统硬件总体架构

Fig.4.7 System hardware architecture

根据4.2.2的传感器节点选型，探测器节点拥有三个传感器，分别是DS18B20、MQ-2与MQ-7，利用三个传感器来采集现场的温度，可燃性气体浓度与CO浓度三个监测参数对象。本系统中的DS18B20为数字信号，可以直接输入使用，而MQ-2与MQ-7则需要进行AD转换将物理信号转换为电信号再进行后续处理。当探测器节点采集数据完成之后，依靠ZigBee技术将数据发送到主控制器进行监测软件上的数据融合处理。

无线传感器网络与ZigBee技术的应用是实现本系统的基础，没有数据交互也就不存在监控的概念，整个网络负责数据的收发与通信，ZigBee技术负责管理整个庞大的网络节点，是系统有序的进行工作。整个硬件系统架构缺一不可，而探测器又如同整个系统的感觉器官，也是最基础的。

### 4.3.2 系统软件总体架构

本系统最重要的通信网络是利用ZigBee技术来构成的，其主要包括三个部分：终端节点（探测器节点）、路由器节点与主协调器节点三部分构成。无线网络由协调器负责建立，然后路由器节点和终端节点通过扫描通信范围内的可用信道向协调器申请加入网络，在无特殊情况下，协调器同意网络加入后为其两个子网络分配一个独有的短通信地址，从而建立组网[1]。

各节点的编程任务如下：

终端节点负责对信号进行编码，对编码后的字符合法性也需要进行判断；同时终端节点负责数据的传输。

协调器不仅要负责网络的形成，且还需要对终端节点数据的收发进行负责。

本课题的CC2530硬件编程使用IAR嵌入式软件开发平台作为开发环境。无线传感器网络的初始化程序需要使用TI提供的协议栈ZStack-CC2530来进行开发。

通讯方式建立完成之后，我们需要对CC2530的串口通讯进行编程，也就是如何在协调器与PC端建立相应的通讯机制。

PC端接收到数据之后，在PC端主要是做数据融合处理，首先需要对神经网络进行初始化，本课题在第三章中介绍了如何初始化一个基于遗传算法优化的BP神经网络算法，此网络相当于整个PC端软件的心脏，实时对采集的数据进行一个运算。数据融合过程首先是通过数据层融合器，利用阈值算法来确认是否可能是发生火灾，如果被确定疑似火灾数据则进行下一步的特征层融合，特征层融合器通过服务器启动时初始化的BP神经网络，经过神经网络的前馈运算后，最终送到决策层融合器进行模糊逻辑算法的一个决策值输出，最终判断是否为火灾发生。

软件编程要求具有模块化思想，各部分有各部分的特有的功能。图4.8为系统软件设计的总体架构。

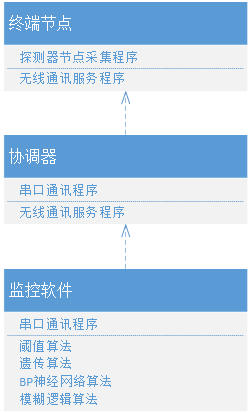


图4.8 系统软件总体架构

Fig.4.8 System Software Architecture

## 4.4 本章小结

本章首先是确定了系统设计的原则，然后根据系统的要求，结合实际情况，对硬件进行选型，分别确定了系统的主控制器使用TI公司最新的ZigBee片上解决方法CC2530，探测器节点使用DS18B20、MQ-2与MQ-7分别来监测温度、烟雾浓度与CO浓度。在本章的最后，提出了系统的硬件与软件总体架构，为下一章的具体设计做好准备。

# 第5章 火灾报警系统的硬件设计

## 5.1 总体硬件设计思路

## 5.2 CC2530主控制器硬件电路设计

### 5.2.1 CC2530最小系统

### 5.2.2 电源及复位电路

### 5.2.3 系统时钟电路

### 5.2.4 串口通讯电路

## 5.3 传感器节点硬件电路设计

### 5.3.1 传感器节点

### 5.3.2 AD转换电路

## 5.4本章小结

# 第6章 火灾报警系统的软件设计

## 6.1 软件开发平台

### 6.1.1 IAR嵌入式软件开发平台

IAR嵌入式软件开发平台是由瑞典的IAR Systems公司专门为ARM嵌入式微处理器开发的集成开发环境。此开发软件能够方便用户对底层代码的开发。

IAR嵌入式开发软件上手速度快，简洁易懂。本系统主要是用IAR嵌入式软件开发平台实现对CC2530的功能开发。

### 6.1.2 ZStack协议栈开发平台

要是有ZigBee网络，需要有协议栈的支撑来实现系统功能。协议栈的工作模式为层与层通过APS接口发送或者接收操作请求。在协议栈中包含一个调度程序，即OSAL（操作系统抽象层），ZStack协议栈就是在OSAL上运行的[1]。

OSAL是一个具有多任务处理、事件轮询、消息处理机制的操作系统。当需要时操作时系统被唤醒，处理事件，其余情况下系统处于休眠状态。OSAL系统还具有消息队列处理机制，将事件按照紧急程度的执行顺序排列，然后从高到低执行事件操作。

如图6.1为项目中的ZStack结构目录。如图所示，在一个ZStack目录下，有一些是出厂时已经根据硬件设备设计好的，我们并不需要关心，拿本系统举例，例如MAC/NWK/OSAL等是厂家并不希望我们去更改的，我们在这里只需要关注的是App层与HAL层。App层主要是增加本系统的应用的具体程序，而HAL层是主要对硬件配置相关做具体的修改。

在系统启动时，会对系统所有的东西进行初始化，为后续网络建立做准备工作。然后系统跑起来后，就是OSAL的调度程序开始工作，实质上它是一个死循环，即轮询式查找状态，主要发现除PHY层以外所有层中发生的事件任务，事件的执行顺序为从下到上的的MAC层直到应用层。

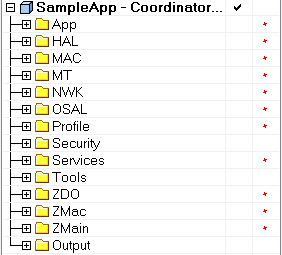


图6.1 系统软件总体架构

Fig.6.1 System Software Architecture

### 6.1.3 Microsoft Visual Studio软件开发平台

Microsoft Visual Studio（简称VS）是美国微软公司的开发工具包系列产品。

VS是一个基本完整的开发工具集，它包括了整个软件生命周期中所需要的大部分工具，如UML工具、代码管控工具、集成开发环境(IDE)等等。所写的目标代码适用于微软支持的所有平台，包括Microsoft Windows、Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework和Microsoft Silverlight 及Windows Phone。

Visual Studio是目前最流行的Windows平台应用程序的集成开发环境。

## 6.2 节点软件设计

### 6.2.1 终端数据采集节点的软件设计

根据功能需求与节点采集软件的主要功能，该软件设计流程如下：

（1）软件在上电时开始初始化所有功能，然后循环扫描可用信道加入网络，等待加入网络成功响应。

（2）当终端节点成功加入组网后，就开始执行自己所负责的工作，即开始进行数据采集工作，将所采集的信号传递至路由器，再由路由器转发至协调器。

终端数据采集节点软件流程图如图6.2所示。

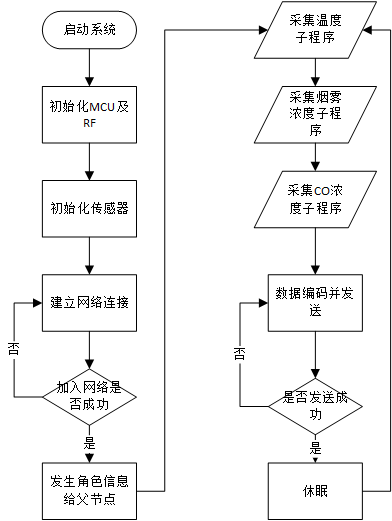


图6.2 终端数据采集节点软件流程图

Fig.6.2 Terminal data acquisition node software flow chart

终端采集节点程序代码如下：

1. 温度采集程序。

float floatReadDs18B20(void)

{

unsigned char V1,V2;

unsigned int temp;

float fValue;

Ds18b20Initial();

Ds18b20Write(0xcc);

Ds18b20Write(0x44);

Ds18b20Initial();

Ds18b20Write(0xcc);

Ds18b20Write(0xbe);

V1 = Ds18b20Read();

V2 = Ds18b20Read();

temp=V2\*0xFF+V1;

fValue = temp\*0.0625;

return fValue;

}

2．烟雾浓度采集及AD转换程序。

uint16 readCGData(void)

{

uint16 reading = 0;

ADCCFG |= 0x40;

ADCCON3 = 0x86;

while (!(ADCCON1 & 0x80));

ADCCFG &= (0x40 ^ 0xFF);

reading = ADCL;

reading |= (int16)(ADCH << 8);

reading >>= 8;

return (reading);

}

3．CO浓度采集及AD转换程序。

uint16 readCOData(void)

{

uint16 reading = 0;

ADCCFG |= 0x40;

ADCCON3 = 0x85;

while (!(ADCCON1 & 0x80));

ADCCFG &= (0x40 ^ 0xFF);

reading = ADCL;

reading |= (int16)(ADCH << 8);

reading >>= 8;

return (reading);

}

### 6.2.2 路由节点的软件设计

路由器作为中间节点负责的任务是将终端采集节点的数据转发给协调器。路由节点的软件流程如图6.3所示。

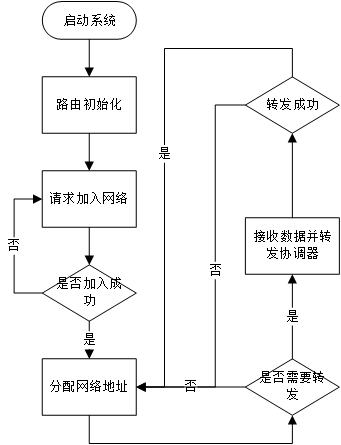


图6.3 路由节点软件流程

Fig.6.3 Routing node software flow

### 6.2.3 协调器节点的软件设计

协调器不仅是完成整个组网的建立与通信，还需要完成数据的收发。接受来自终端的数据且将接收到的数据通过串口转发给PC端。

协调器主要工作流程如下：

（1）系统上电后即初始化，然后进行网络的组网和建立，筛选出可用信道，接受设备的申请需求，分配网络地址。

（2）网络建立完成后，接受转发的数据信息，将数据通过串口发送给PC机。

（3）重复步骤（2）。

此时需要对协调器进行编程，由于ZigBee协议栈提供的应用层接口方便用户使用。本系统的传输方式采用点播通讯，在ZStack协议栈中已经提供相应的接口SampleApp\_Send\_P2P\_Message，我们只需要加入任务处理需求与应用程序代码即可。

协调器软件工作流程如图6.4所示。

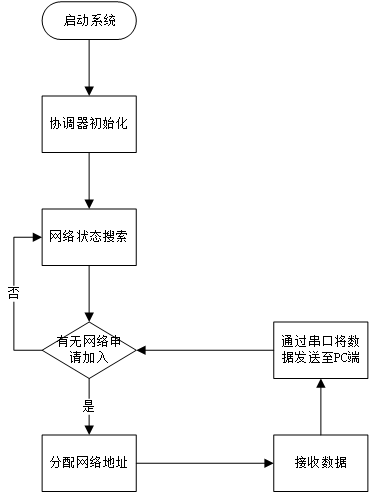


图6.4 协调器软件工作流程

Fig.6.4 Coordinator software workflow

通讯程序代码如下：

void SampleApp\_Send\_P2P\_Message( void )

{

char precdictValue[30]; //待传输数据

uint16 CGData;

uint16 COData;

float temperature;

temperature = floatReadDs18B20(); //DS18B20数据子模块

CGData = readCGData(); //烟雾浓度读取子模块

COData = readCOData(); //CO浓度读取子模块

sprintf(precdictValue, "TM%.02f|CG%d|CO%d", temperature, CGData, COData); //数据编码，方便PC端程序使用

if ( AF\_DataRequest( &SampleApp\_P2P\_DstAddr, &SampleApp\_epDesc,

SAMPLEAPP\_P2P\_CLUSTERID,

strlen(precdictValue),

precdictValue,

&SampleApp\_TransID,

AF\_DISCV\_ROUTE,

AF\_DEFAULT\_RADIUS ) == afStatus\_SUCCESS )

{

} //数据发送

}

当通讯类型为SAMPLEAPP\_P2P\_CLUSTERID时，通过HalUARTWrite(0, pkt->cmd.Data, pkt->cmd.DataLength)来完成串口数据的发送。

## 6.3 监控PC端软件设计

监控PC端软件主要是通过采集到串口传输的数据来进行数据融合，最终输出一个决策值。本系统软件设计思路考虑到多线程思想，为网络机器学习的过程专门开辟一个新线程，同时也为串口数据读取开辟一个新的线程，主线程执行其他扩展操作。整个代码的功能块主要划分为5个部分：串口通讯程序、阈值检测算法、遗传算法程序、BP神经网络算法程序、模糊逻辑算法程序。

由于考虑到论文的精简与突出重点，本章只介绍比较简单的代码块，比较复杂的代码块用流程图、伪代码的形式表现，具体每个模块代码详见附录。

### 6.3.1 串口通讯程序

如图6.5所示为本系统串口设计类图。



图6.5 串口通信类图

Fig.6.5 Serial port communications class diagram

串口通讯程序是使用第三方开源库QT库中<QSerialPort>与<QSerialPortInfo>来完成的，该串口通讯程序主要是完成从协调器串口读取数据与解码工作，然后发送至神经网络线程中进行决策输出。

串口初始化流程与串口通讯程序流程分别如图6.6和6.7所示：

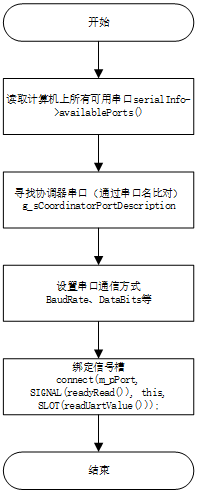
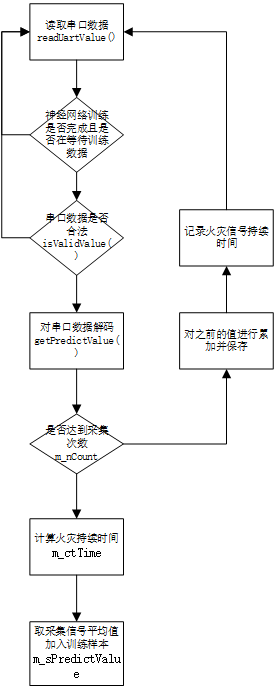
 

图6.6 串口初始化流程 图6.7 串口通讯程序流程

Fig.6.6 Serial initialize flow Fig.6.7 Serial port initialization process

### 6.3.2 阈值检测算法程序

此处的阈值检测算法为数据层融合器，仅仅只是过滤掉非火灾信号。

算法代码如下：

……

if (ALARM\_THRESHOLD < temp.inputValue.at(0) || ALARM\_THRESHOLD < temp.inputValue.at(1) || ALARM\_THRESHOLD < temp.inputValue.at(2))

{

……

}

if (ALARM\_THRESHOLD > temp.inputValue.at(0) && ALARM\_THRESHO\

LD > temp.inputValue.at(1) && ALARM\_THRESHOLD > temp.inputValue.at(2))

{

……

}

……

### 6.3.3 遗传算法程序

如图6.8所示为本系统遗传算法类的设计图。



图6.8 遗传算法类图

Fig.6.8 Genetic algorithm class diagram

由于遗传算法的流程图已在3.1.1节图3.1中介绍，此处不再介绍遗传算法的流程图。

针对本系统的遗传算法，设定初始化遗传代数GA\_GENERATIONS\_NUMB-ER为10000代；初始种群规模GA\_POPULATION\_SCALE为100代。

遗传算法的代码实现流程如下：

for (int i = 0; i < GA\_GENERATIONS\_NUMBER; i++)

{

/\* Genetic algorithm \*/

population->selection(); //选择算子

population->cross(); //交叉算子

population->mutate(); //变异算子

}

在遗传迭代次数完成后，需要经过以下两个步骤：

population->setBestIndividual();

population->getBestIndividual();

此时选出的最佳的个体的阈值和各层之间的权值作为BP神经网络算法的初始化阈值和权值。

在这里，特别说明遗传算子的编解码形式采用宏定义，以提高系统运算的效率。编解码的宏定义如下：

#define GA\_ENCODE\_STYLE(x) ((x) \* (GA\_RANGE - 1)) / (GA\_MAX\_WE\

IGHT - GA\_MIN\_WEIGHT)

#define GA\_DECODE\_STYLE(x) ((double)(x) \* (double)(GA\_MAX\_WEIGHT - GA\_MIN\_WEIGHT) / (double)(GA\_RANGE - 1))

### 6.3.4 BP神经网络算法程序

如图6.9所示为本系统BP神经网络类图。



图6.9 BP神经网络类图

Fig.6.9 BP neural network class diagram

由于BP神经网络算法的流程图已在3.1.2节图3.7中介绍，此处不再介绍BP神经网络算法的流程图。

在BP神经网络算法中，核心代码主要是网络训练代码块与网络预测代码块。

1．网络训练代码块。

void BPNeuralNetworks::training(samples learnValueList, double rateThreshold\

Value, double errorThresholdValue, bool bothMatch)

{

……

while (1)

{

if (bothMatch)

{

…… //退出条件

}

else

{

…… //退出条件

}

……

//Start training

for (int iter = 0; iter < learnValueCount; iter++)

{

setInput(learnValueList[iter].inputValue);

setOutput(learnValueList[iter].outputValue);

sendForwardPropagation(); //前馈计算

sendBackPropagation(); //反馈计算

}

function1(); //更新输入层权值与阈值

function2(); //更新隐藏层权值与阈值

function3(); //更新输出层阈值

……

}

}

2．网络预测代码块。

void BPNeuralNetworks::predict(samples& testValueList)

{

……

for (int it = 0; it < testValueCount; it++)

{

//前馈计算

}

}

### 6.3.5 模糊逻辑算法程序

图6.10为模糊逻辑算法的类图。



图6.10 模糊逻辑算法类图

Fig.6.10 Fuzzy logic algorithm class diagram

模糊推理规则如第3.3.3节中表3.2节所示，在实例化FuzzyReasoning时，会对模糊规则进行录入，核心代码如下：

void FuzzyReasoning::setMembershipRule(const double fuzzyRuleList[]\

[FUZZY\_RULE\_COLUMN])

{

for (int i = 0; i < FUZZY\_RULE\_ROW; i++)

{

FuzzyRule fuzzySingleRule;

fuzzySingleRule.isFireRate = fuzzyRuleList[i][0];

fuzzySingleRule.likeFireRate = fuzzyRuleList[i][1];

fuzzySingleRule.time = fuzzyRuleList[i][2];

fuzzySingleRule.finalStatus = fuzzyRuleList[i][3];

m\_oFuzzyRules.push\_back(fuzzySingleRule);

}

}

当接收到一组数据的时候会对数据的隶属度进行判断，然后再查所录入的表信息，最终输出决策值。

模糊逻辑算法流程图如图6.11所示。

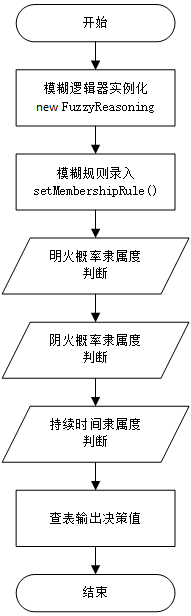


图6.11 模糊逻辑算法流程图

Fig.6.11 Fuzzy logic algorithm flow chart

## 6.4 软件整体工作流程

本部分介绍整个监测软件的工作流程，将所有模块组合起来了解会更清晰整个系统的工作模式。软件整体工作流程如图6.12所示。

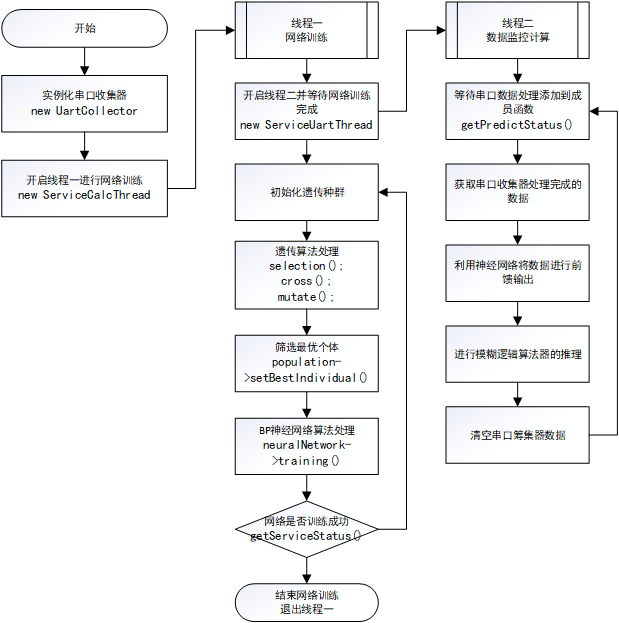


图6.12 软件整体工作流程

Fig.6.12 Software folw chart

## 6.5 本章小结

本章从两个部分介绍了整个系统的软件部分：一是节点软件的设计；二是监控PC端软件的设计。总的来说，节点软件的核心部分为协调器端的编程，需要确定通信方式，本系统采用ZStack中已有的点对点通信模式，而监控软件的核心在于如何训练出神经网络，本节都对这些部分进行一个详细的介绍。

最后对系统做了一个总结，将所有软件流程化，是系统的工作流程更加清晰。

# 第7章 模拟与实验结果分析

## 7.1 实验数据采集结果

## 7.2 实验数据分析

## 7.3 本章小结

# 第8章 结论及展望

## 8.1 总结

本系统的设计抓住了传统设备的痛点，把无线传感器网络和ZigBee技术运用到了无线火灾报警系统中，并且利用数据融合算法来对传感器探测节点的数据进行融合。该系统既克服了传统设备的布线工序的复杂，也解决了因单传感器导致的误报与漏报情况的产生。本论文的具体完成内容如下。

1．分析了国内外火灾报警系统的研究现状。

2．将无线传感器网络与ZigBee技术运用到火灾报警系统中。

3．深入剖析了本系统的数据融合算法，然后针对于本系统来建立了基于遗传算法优化的BP神经网络算法的机器学习的过程，同时也确定了模糊逻辑的判断规则。

4．确定了系统的总体设计方案，包括处理器与传感器选型，软硬件设计架构。采用CC2530作为系统的主控制器芯片，传感器节点采用DS18B20、MQ-2、MQ-7来监测现场参数。软件采用模块化编程思想，确定了软件的设计思路。

5．完成了系统硬件的设计。包括CC2530主控电路的设计与传感器采集节点与AD转换电路的设计。

6．采用ZStack协议栈详细设计了CC2530的软件 ，给出了各模块的流程图与伪代码。再利用Visual Studio 2015，使用C++语言对监控系统的上位机软件进行了编写。

7．对系统进行了联调。模拟火灾情景，收集现场实验数据与结果。测试结果表明。系统运行可靠，达到预期目标。

## 8.2 展望

由于时间与个人水平有限。本文目前只完成了在实验阶段的测试，并没有能力实际运用到楼宇火灾的情况中，且系统的网络算法有时运行时间较长，训练数据规模较小，导致特殊情况的数据不精确，今后系统需要从以下几点去改进或者是升级系统。

1．数据融合。

对于神经网络算法来讲，需要提升该系统的训练数据规模，改进神经网络算法学习率的自适应度。对于遗传算法需要寻找一个最优的适应度表达公式，寻找最优的变异或交叉的算法。

2．报警功能完善。

本系统暂时没有增添报警电路，后续升级可以根据系统输出的标识符（0或1）来驱动报警电路，实现应该比较简单。

3．报警善后处理。

要真正做到智能化，可以根据火灾现场的火势大小（可以采用摄像头，利用图像算法来识别火势大小，这样精度更高）采取不同的善后处理措施。

# 致谢

到此，本系统的设计工作与论文撰写工作也即将完成。这篇文章是对我大学四年生活的一个总结，也将我大学四年的所有知识运用在一起完成了这个系统的研究。

回首四年的生活，欢声笑语、酸甜苦辣的校园生活丰富了我的人生。在大学学到了很多，课本上的知识，以及课本上学不到的一些待人处事的道理。这次论文的撰写也是对我大学生活的一个总结，是对西南石油大学的留念。

首先，感谢西南石油大学对我四年的栽培，四年的收获，是我一辈子的财富，四年的时光是我人生中美丽的回忆。

其次，感谢我的指导老师——罗宾老师。从进入西南石油大学，所有的实验课都是罗老师一直在带我们，罗老师的授课方式我十分喜欢。这次罗老师带我毕业设计更增加了我对罗老师的感激之心，罗老师对我的毕设任务十分负责。一日为师终身为父，很珍惜这学期以来能和老师接触的时间这么多，感谢罗老师一直对我的指导关心和督促。

其次，感谢陪伴我大学四年的舍友、同学们。感谢你们给我大学四年的生活增添了很多色彩，大学因你们而精彩。

最后，感谢所有的答辩老师，谢谢你们能够耐心得对我论文进行评审，感谢老师们的悉心指教，感谢所有老师复出的辛勤劳动。

# 参考文献

1. 李吉宗. 基于ZigBee的智能楼宇火灾报警系统研究[D]. 上海:华东理工大学, 2014. 5-7
2. 白云龙. 基于ZigBee的无线消防报警系统设计[D]. 辽宁:辽宁工业大学, 2016. 5-7
3. 颜学义. 基于ZigBee的智能火灾报警系统设计 [D]. 《国防科学技术大学》, 2008. 6-10
4. 陈霄. DNA遗传算法及应用研究[D]. 浙江:《浙江大学》, 2010. 2-4
5. 满丽欣. SDN网络下基于BP神经网络算法的负载均衡研究[D]. 吉林:《吉林大学》, 2017. 21-26
6. 赵利民. 一种改进的BP神经网络算法及其在集中供热系统中的应用 [D]. 吉林:《吉林大学》, 2012. 5-11
7. 周创创. 基于无线传感网络的楼宇火灾监测系统研究 [D]. 西安:《西安建筑科技大学》, 2016. 15-22