

团队编号： XJTU2017TF275



第十七届“腾飞杯”课外学术科技作品竞赛

参赛作品

作品名称： IMCS: 基于物联网技术的智能监控系统

类别：

- ☐ 自然科学类学术论文
- ☐ 哲学社会科学类社会调查报告和学术论文
- ☒ 科技发明制作 A 类
- ☐ 科技发明制作 B 类

论文题目：IMCS: 基于物联网技术的智能监控系统

学科专业：自动化控制科学与技术

申请人：杨涛，蒋旻阳，曾浩伦，涂志俊，王子路

指导教师：任鹏举 副教授

摘 要

物联网是新一代信息技术的重要组成部分，也是“信息化”时代的重要发展阶段。其英文名称是：“Internet of things (IoT)”。顾名思义，物联网就是物物相连的互联网。这有两层意思：其一，物联网的核心和基础仍然是互联网，是在互联网基础上的延伸和扩展的网络；其二，其用户端延伸和扩展到了任何物品与物品之间，进行信息交换和通信，也就是物物相息。物联网通过智能感知、识别技术与普适计算等通信感知技术，广泛应用于网络的融合中，也因此被称为继计算机、互联网之后世界信息产业发展的第三次浪潮。

本文基于创新和实用两大主题，构建了一套通用的基于物联网技术的智能监控系统，提出了通用的 IMCS (Intelligent Monitor and Control System based on IoT technique)：基于物联网技术的智能监控系统的方案和设计实现，并以中铁十八局新建京张铁路降水自动化监控系统作为本系统的典型应用和测试验证。本系统的整体设计与开发主要包括两大方面：嵌入式系统（包括软/硬件开发）和系统软件开发。硬件开发主要包括：根据应用需求设计电路原理图与印制电路板（PCB），基于 ARM 内核的 STM32 嵌入式编程、基于 Zigbee 技术的无线通讯传输。软件部分主要包括：基于 GPRS 移动网络的数据通讯接口开发、服务器端开发、数据库设计和 web 开发。STM32 编程是嵌入式系统开发部分的核心，主要功能包括节点工作控制和系统通讯两部分。前者主要为：完成 IoT 节点网络自动配置、记录节点的工作状态和根据传感器与服务器远程信号控制执行单元，系统通讯主要包括：ZigBee 数据收发、UART 串口通讯、响应通讯指令中断等，多层数据的通讯；而系统软件部分的核心是数据库——MySQL，它的作用是管理和存储硬件部分的 DTU (Data Transmit Unit) 通过 GPRS 网络传递到服务器上的 IoT 节点的实时工作状态以及历史统计信息，并与 Web 端进行交互，实现数据在网页和移动端的显示和与用户的交互。

为了获得更好的用户体验，我们的 web 开发分为 PC 端和移动端，以满足不同用

户的需求和适应不同设备分辨率的差别。并且计划了安卓移动端 App 的开发，以满足一个 App 监控多个系统的便利。

该智能监控系统具有方案完整、实用性高、可移植性好、应用前景广泛的特性，对于不同的实际需求可以通过增加外设（如摄像头、各式传感器等）实现不同的数据采集功能，并且在很多需要自动化控制或智能监控的场景都能直接应用，如：智能灌溉、智能消防、养老院老人监护等。

关 键 词： 物联网；电路设计；STM32；数据库；web 开发

目 录

| | |
|-------------------------|----|
| 1 绪论 | 1 |
| 1.1 “互联网+”浪潮 | 1 |
| 1.2 物联网的发展趋势 | 2 |
| 1.3 国内外物联网应用现状 | 3 |
| 1.4 本文的主要研究内容 | 3 |
| 1.5 本文的主要章节安排: | 4 |
| 2 系统总体设计 | 5 |
| 2.1 网络结构 | 5 |
| 2.2 节点结构 | 6 |
| 2.3 数据传输 | 7 |
| 3 电路系统设计 | 9 |
| 3.1 主控芯片 | 10 |
| 3.2 电源模块 | 10 |
| 3.3 无线模块 | 11 |
| 3.4 拨码开关 | 11 |
| 3.5 有线通讯 | 12 |
| 3.6 执行单元控制输出 | 12 |
| 4 嵌入式软件开发 | 13 |
| 4.1 程序的整体设计 | 13 |
| 4.2 程序的具体实现过程 | 14 |
| 4.3 实验过程中遇到的困难 | 17 |
| 4.4 解决方案 | 17 |
| 5 服务器开发及数据交互接口设计 | 18 |
| 5.1 TCP/IP 服务器 | 18 |
| 5.2 与 DTU 数据交互 | 22 |
| 5.3 与数据库交互 | 23 |
| 5.4 与 Web 后端交互 | 23 |
| 5.5 遇到的问题 | 24 |
| 5.6 解决方案 | 24 |
| 6 数据库开发 | 25 |
| 6.1 数据库结构设计 | 25 |
| 6.2 数据库数据的导入和实时更新 | 26 |
| 7 web 开发 | 27 |

| | |
|---------------------|----|
| 7.1 网站的构建 | 27 |
| 7.2 网页后端开发 | 29 |
| 7.3 网页美工设计 | 29 |
| 7.4 网页前端开发 | 29 |
| 7.5 网页前端与后端交互 | 30 |
| 8 移动端 App 开发 | 31 |
| 9 系统评测与后期改进 | 32 |
| 9.1 系统评测 | 32 |
| 9.2 改进 | 32 |
| 10 实际应用与推广 | 33 |
| 10.1 实际应用 | 33 |
| 10.2 项目推广 | 35 |
| 参考文献 | 37 |
| 附 录 | 39 |

1 绪论

20 世纪下半叶，以互联网计算机为代表的第三次工业革命迅速席卷全球，世界进入了信息化时代。而 21 世纪提出的物联网也被称为继计算机、互联网之后世界信息产业发展的第三次浪潮。物联网将是下一个推动世界高速发展的“重要生产力”，是继通信网之后的另一个万亿级市场。而在物联网的潮流中，应用创新是物联网发展的核心，以用户体验为核心的创新 2.0 是物联网发展的灵魂。

1.1 “互联网+”浪潮

“互联网+”是创新 2.0 下的互联网发展新形态、新业态，是知识社会创新 2.0 推动下的互联网形态演进及其催生的经济社会发展新形态。“互联网+”是互联网思维的进一步实践成果，它代表一种先进的生产力，推动经济形态不断的发生演变。从而带动社会经济实体的生命力，为改革、创新、发展提供广阔的网络平台。

通俗来说，“互联网+”就是“互联网+各个传统行业”，但这并不是简单的两者相加，而是利用信息通信技术以及互联网平台，让互联网与传统行业进行深度融合，创造新的发展生态。它代表一种新的社会形态，即充分发挥互联网在社会资源配置中的优化和集成作用，将互联网的创新成果深度融合于经济、社会各领域之中，提升全社会的创新力和生产力，形成更广泛的以互联网为基础设施和实现工具的经济发展新形态。

“互联网+”行动计划将促进产业升级。首先，“互联网+”行动计划能够直接创造出新兴产业，促进实体经济持续发展。互联网+行业能催生出无数的新兴行业。比如，互联网+金融激活并提升了传统金融，创造出包括移动支付、第三方支付、众筹、P2P 网贷等模式的互联网金融，使用户可以在足不出户的情况下满足金融需求。其次，“互联网+”行动计划可以促进传统产业变革。“互联网+”令现代制造业管理更加柔性化，更加精益制造，更能满足市场需求。最后，“互联网+”行动计划将帮助传统产业提升。互联网+商务=电商，互联网与商务相结合，利用互联网平台的长尾效应，在满足个性化需求的同时创造出了规模经济效益。

“互联网+”行动计划将重点促进以云计算、物联网、大数据为代表的新一代信息技术与现代制造业、生产性服务业等的融合创新，发展壮大新兴业态，打造新的产业增长点，为大众创业、万众创新提供环境，为产业智能化提供支撑，增强新的经济发展动力，促进国民经济提质增效升级。

1.2 物联网的发展趋势

作为一个新兴产业,物联网从诞生到广泛应用需要经历四个阶段:

第一阶段为设想阶段,这个时期为产业发展的最初期。

第二阶段为技术研发阶段,美国、欧盟等都在投入大量资源深入研究物联网。我国在 1999 年就启动了物联网核心技术无线传感器网络的研究,国家投入大规模资金用于物联网研发。

第三阶段为实验阶段,在技术研发的水平达到一定程度时,就可以进行小范围的试用和检测,这是从理论走向实践的一步。目前一些企业已开始尝试物联网商用,如 IBM 公司的“智慧地球”。在商用领域我国相关企业也开始了探索,在 2009 中国国际信息通信展览会上,中国移动、电信和联通三大电信运营商开始尝试物联网业务,比如中国移动的手机钱包和手机购电业务。该业务也可以应用于超市、餐厅等小额支付场合;中国联通的无线环保检测平台通过 3G 网络,可实时对水表、灌溉、水文、水质等动态数据进行监测,又可对空气质量、碳排放和噪音进行监测。

第四阶段为全面推广阶段,也是投入资金最大的时期。同时,一旦大规模商用,大量基础设施的建设和终端产品的全面推广必将推动电信、信息存储处理、IT 服务整体解决方案等众多相关市场的发展。在物联网普及以后,用于动物、植物和机器、物品的传感器与电子标签及配套的接口装置的数量将远远超过手机的数量。物联网的推广将会成为推进经济发展的重要领域,美国权威咨询机构 Forester 预测,到 2020 年,世界上物物互联的业务,跟人与人通信的业务相比,将达到 30 比 1,因此,“物联网”被称为是下一个万亿级的通信业务。

1.3 国内外物联网应用现状

1.3.1 国内现状

早在“十二五”时期，我国在物联网发展政策环境、技术研发、标准研制、产业培育以及行业应用方面取得了显著成绩，物联网应用推广进入实质阶段，示范效应明显，已成为推动经济社会智能化和可持续发展的重要力量。国家“十三五”规划纲要明确提出“发展物联网开环应用”，将致力于加强通用协议和标准的研究，推动物联网不同行业不同领域应用间的互联互通、资源共享和应用协同。目前，我国物联网正广泛应用于电力、交通、工业、医疗、水利、安防等领域，并形成了包含芯片和元器件、设备、软件、系统集成、电信运营、物联网服务在内的较为完善的产业链体系。另外，在空间上初步形成环渤海、长三角、泛珠三角以及中西部地区四大区域集聚发展的格局。

1.3.2 国外现状

国际金融危机爆发后，美、欧、日、韩等主要发达国家纷纷把发展物联网等新兴产业作为应对危机和占领未来竞争制高点的重要举措，制定出台战略规划和扶持政策，全球范围内物联网核心技术持续发展，标准和产业体系逐步建立，初步形成了传感器与无线射频识别（RFID）等感知制造业，网络设备与通信模块、机器到机器（M2M）终端与运营服务以及基础设施服务、软件与集成服务等产业链，2011 年全球物联网产业规模超过 1345 亿美元。发达国家凭借信息技术和社会信息化方面的优势，在物联网应用及产业发展上具有较强竞争力。

1.4 本文的主要研究内容

本文的主要研究内容是基于物联网技术的智能监控系统的设计与开发。我国的物联网应用前景广阔，但总的来说普及率不高，很多场景都具有“物联网+”的潜在应用可能性。本系统设计的目的是基于数据库实现在 Web 端和移动 App 上实时显示 IoT 各节点的工作情况和远程监控以及历史信息记录的完整系统方案。研究的主要内容如下：

1. 嵌入式电路设计：主要是根据应用功能需求设计电路原理图和 PCB；
2. 嵌入式软件开发：基于 ARM 内核的 STM32 芯片的嵌入式编程；

3. 数据交互接口技术和服务器开发：实现 IoT 节点的 Zigbee 和 GPRS 数据收发，以及与服务器的数据传输；
4. 数据库开发：将节点工作信息管理和存储在一个具有明确结构的数据处理系统中，便于生成历史数据，方便工业化管理以及 web 端进行数据交互；
5. Web 开发，主要包括网页前端开发和后端开发，实现数据的更为直观的显示和更好的交互效果，改善用户体验和提高管理效率。

1.5 本文的主要章节安排：

第一章是绪论部分，主要论述该项目的选题意义、应用背景、国内外研究现状分析和本文的主要研究内容。

第二章主要是系统的总体设计方案，分别从网络结构、节点结构和数据传输三个大的方面阐述了整个智能监控系统的总体设计思路。

第三章开始是系统各部分详细的设计方案，第三章主要是电路系统的设计，分别从主控芯片、电源模块、无线模块等六个部分详细阐述了电路系统的设计方案。

第四章是嵌入式软件开发，该章节主要从基于 ARM 内核的 STM32 单片机软件编程的总体设计、程序具体实现过程和遇到的问题及解决的方案几个方面阐述了系统的嵌入式软件开发部分的设计思路。

第五章是服务器开发及数据交互接口设计，该章节主要从 TCP/IP 服务器开发、服务器与 DTU 交互、与数据库交互等几个方面阐述了服务器开发及数据交互接口设计的设计思路。

第六章是数据库的开发，该章节主要讲解了数据库的结构设计和数据导入数据库及数据的及时更新设计的具体方案。

第七章是 web 开发，其分为网站的构建、网页前后端设计及其交互等几个部分。

第八章移动端 App 的开发，主要阐述了移动端开发的流程及整体设计思路。

第九章是系统测评和后台改进，主要是从一个比较客观的角度评价了该系统的总体设计方案的先进性和提出了后续可以改进的不足之处。

第十章是实际应用和推广，该章节重点阐述了作为本智能检测系统的典型应用和测试验证的中铁十八局新建京张铁路降水自动化监控系统的实际设计思路和测试进度，并且展示了该智能监控系统的广大的应用前景。

2 系统总体设计

本项目是设计基于物联网技术的智能监控系统。首先需要考虑的是网络系统的设计，其次是单个节点执行单元的设计，从而提高用户体验，使得数据更加利于用户分析与观察，增加系统功能的同时提高系统可移植性。优化的网络结构，与数据采集方式可以增强系统稳定性，以及便于开发或使用人员对突发事件的处理，合理的节点执行单元的设计可以使得系统可移植与功能更加丰富。

2.1 网络结构

网络结构的设计主要考虑两个方面，第一个是用户接口和管理，其次是节点执行单元的设计。

2.1.1 面向用户设计

用户方面，为便于用户实时方便快捷的获取系统信息，以及对系统进行相应控制，我们使用互联网作为用户获取信息的渠道，App 考虑到用户能够及时方便快捷的获取系统信息，我们采用网页 web 的 PC 端以及移动端设计，为增强系统稳定性，防止由于突发事件使得用户不能获取到相应信息，本项目采取移动端 App 作为用户获取系统信息的备用方案。

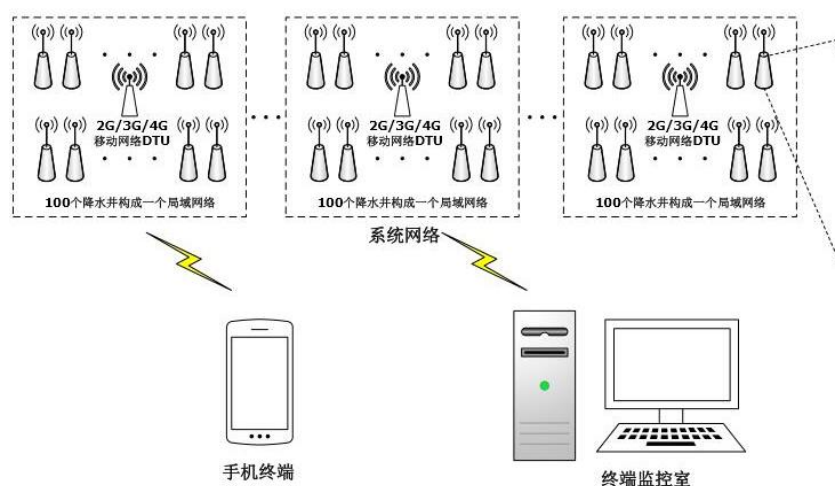


图 2-1 网络结构示意图

2.1.2 节点设计

单个节点执行单元方面，由于节点执行单元需要先构建一个底层网络，将一定数量的节点组成该系统的网络，由于传输协议不同，以及便于开发人员进行进一步的数据处理工作，我们需要该系统的二级网络，连接用户与一级网络，并在二级网络的云端进行相应的数据处理，然后将处理后的数据发送到用户，考虑到现在较为成熟的 IoT 传输协议为 Zigbee，为增强系统的可靠性，我们采用其作为该系统网络的一级网络的传输协议，增加节点的宽带利用率以及防止数据丢包，我们采用网状拓扑结构，即一个网络一个中心节点，其余均设置为路由节点。如上图所示，当节点收到非中心节点的信息时，会将数据立即转发出去，并且通过定时轮询方式（云端逐一与每个网络的每个节点进行一次握手和数据通讯）。考虑到系统稳定性，以及便于用户通过互联网获取数据，我们采用 GPRS 移动互联网协议作为二级网络的数据通讯，将一级网络的数据进一步传输到互联网云端，进行数据处理和管理。

2.2 节点结构

2.2.1 节点设计：

为了便于节点在较为恶劣的环境下工作，我们进行了相关元器件的选型，首先考虑到，功能的复用性，以及系统的可移植性。，我们选择 STM32F10x 系列芯片作为控制核心单元，采用拨码开关进行不同共工作模式的选择与相关参数配置，并且节点可以重配置与重编程以便添加与删除相关功能和工作方式的更改。，源部分的设计，考虑到驱动大型电器工作，我们选择从 220V 中取电，将 220V 转换为直流 3.3V 和 5V，供 ARM 及外围电路使用。为便于 Zigbee 传输数据，增加数据可传输的距离，采用胶棒天线作为单个节点的数据收发天线，并通过 UART（通用异步收发传输器）进行 Zigbee 与 ARM 的数据交互，考虑到地形复杂的原因，单个节点我们设置了两个工作模式，一个是 Zigbee 工作模式另一个是以太网工作模式，以便在 Zigbee 无信号的区域采用以太网进行数据传输。

2.2.2 软件设计：

为便于项目的移植性，STM32 工程的设计为系统文件，和外部功能文件，系统文件主要功能为进行一级网络的搭建，系统的基本功能，例如配置和初始化 Zigbee 模块

等其他基本外围电路，外部功能文件包括驱动外围控制电路的程序和相关数据采集程序，若需要添加和更改相关功能，采集更多种类的传感信息，可以通过更改或者添加相关功能文件实现，不同的功能采用不同的相关程序文件夹，以便于更好的移植和扩展系统功能。

2.3 数据传输

2.3.1 数据传输格式

底层网络层面：

我们采用如下图所示的数据传输格式进行相关数据的传输，最先两位是识别节点和网络的必要数据。

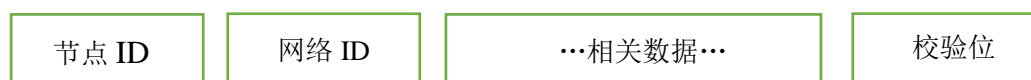


图 2-2 网络数据传输格式

中间网络层面：

为便于数据的进一步处理，采用与底层网络相同的数据格式

互联网层面：

由于大量的数据需要从数据库中提取并传给用户，我们采用 JSON 数组的数据传输方式即为：

```
{["net_id":"*****"; "node_id":"*****"; "year_data":"*****"]}
```

2.3.2 数据处理方式：

为减少数据处理的时间，以便达到数据的实时采集和显示，采用 php 调用数据库的引擎进行相关的数据处理，例如计算周平均值，累计值，详细处理方法需要结合相应用途，少量的简易数据处理在客户端进行。

2.3.3 数据传输过程

如下图所示，数据传输过程为：

正向收集数据回路：每个网络的单个节点传输数据到相邻节点，依次类推最后通过离中心最近的节点传输到 DTU(数据传输单元：由一个 Zigbee 与一个 GPRS 所组成)，中心 Zigbee 收到信息后将所得信息通过串口传输到 GPRS，由 GPRS 将数据传输到最近的基站，通过基站进入以太网连接到系统的服务器，将所得数据汇总到服务器，服

务器获得数据后，将数据存入数据库，通过数据库数据处理之后，若用户端产生 Http 请求，服务器将相关网页文件返回给用户，接着通过 JSON 数组将数据发送到用户端，用户端将数据呈现在网页上。

逆向控制回路：当用户对单个节点或者某片节点产生操作时，通过网页向服务器发送 Http 请求，后台产生相关控制指令的 txt 文件生成，服务器读取此文件并向相关网络的 DTU 发送与此相关的指令，当数据到达基站后，通过 GPRS 网络传输到 DTU，DTU 传输到中心 Zigbee，又由中心的 Zigbee 传输到相关节点 Zigbee 中，最后 Zigbee 将数据命令发送至 ARM 核心控制单元，对指令进行解析和判断之后，执行相关的功能和操作。

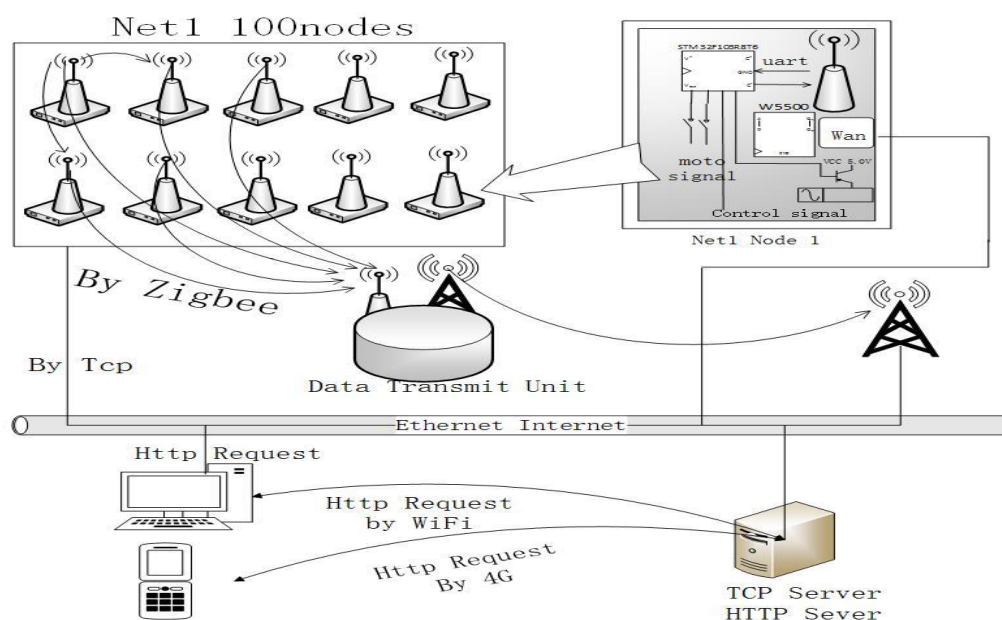
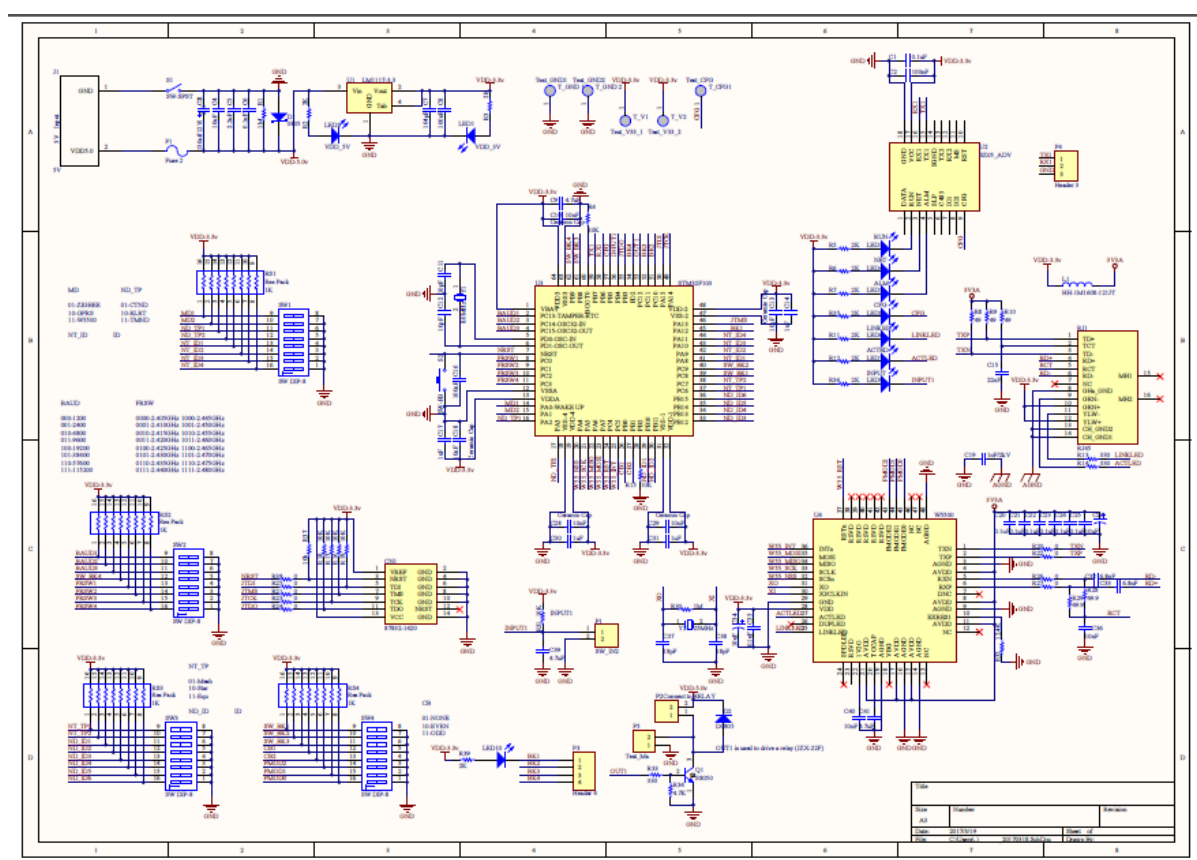


图 2-3 系统结构示意图

3 电路系统设计

本项目的电路总体设计如下，包括主控芯片 ARM 处理器（STM32RBT6LQFP64），电源模块，低压差线性稳压器（LM1117），Zigbee 无线模块（SZ05_ADV），用户配置拨码开关四个，以太网控制芯片（W5500），网口以及若干连接传感器和执行单元的接口电路。



3.3 无线模块

无线模块采用了 SZ05_ADV，其可以利用 ZigBee 快速组网并通信，抗干扰能力强，串口应用灵活。可选择多种工作模式，如发送模式有广播发送及目标地址发送，节点类型有中心节点，路由节点，终端节点等。拥有 16 个信道，可设置 65535 个网络 ID，网络容量大。因此，我们选择此模块作为无线收发装置。

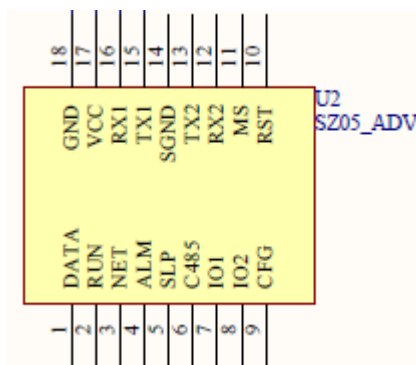


图 3-5 Zigbee 无线模块

3.4 拨码开关

拨码开关的作用是提高系统使用的灵活性与可复用性，在快速切换节点当前的工作模式，通过拨码可以调整无线模块的发送频率，组网方式，节点类型，网络节点 ID，节点 ID 等配置信息。当节点网络被重新部署，拓扑结构发生变化时，可以通过拨码开关配置来进行快速部署。

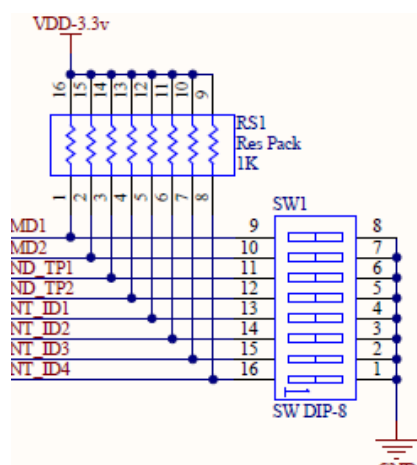


图 3-6 拨码开关电路图示例

3.5 有线通讯

下图中的芯片为 W5500，负责控制以太网的工作，当本项目采用网口直接连接有线网络，可以同样实现基于有线网络远程控制功能。

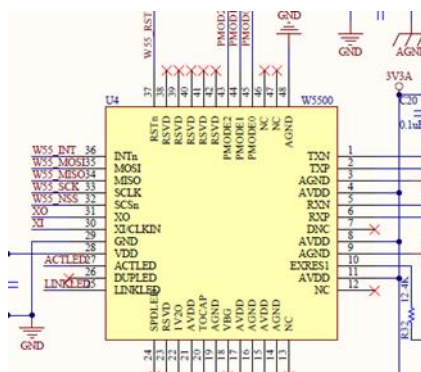


图 3-7 有线通讯模块 W5500

3.6 执行单元控制输出

IO 输出采用了三极管的开关和电流放大的性质：当主控芯片接收到开启电机的信号后，便会拉高 OUT1 的电平，从而使电路导通，吸合继电器线圈，开启电机。反之，则关闭电机。P5 作为测试孔，当电路出现问题无法启动电机时，可以通过测试 P5 来诊断电机或者继电器的故障。

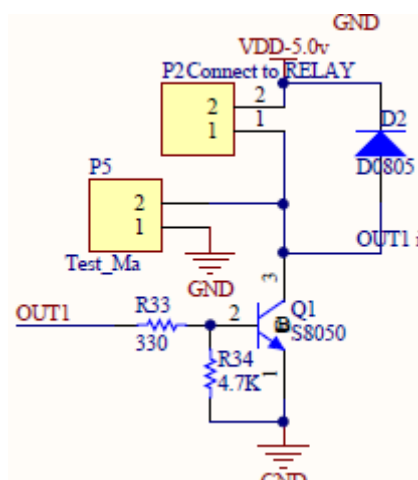


图 3-8 执行单元控制输出电路

4 嵌入式软件开发

嵌入式 ARM 软件设计是该检测系统的终端部分,单个节点的控制信息收集主要由该部分进行操作,同时作为系统的执行单元,它的作用是驱动设备工作,使其按照一定的逻辑进行工作,通过与 ZigBee 的数据传输,达到上传数据,识别指令并执行的作用。在该项目中,需要配置多个参数,以及检测节点工作状态、记录节点工作时间和控制电机,经过进一步计算和设计,我们需要 64 引脚的,flash 为 128Kb,有多个定时器计数器的工业级芯片,因此这里我们选用意法半导体公司生产的 STM32F103RBT6 芯片,因其运用广泛,功能强大,价格便宜。

4.1 程序的整体设计

由于本项目中每个节点是实时工作的,并且功能较为复杂,所以在编写嵌入式程序之前,我们需要对嵌入式程序进行整体设计,大致将程序分为了以下几个模块:

- 1) 初始化模块:初始化 IO 及其所对应的时钟总线,进行系统时钟树的分频
- 2) 读取工作模式模块:通过读取拨码开关的电平,读取节点工作模式
- 3) ZigBee 工作模式模块:收发数据由 ZigBee 进行操作
- 4) W5500 工作模式模块:由于无线工作时信号强度有差异,因此将 W5500 用作备用的模块。

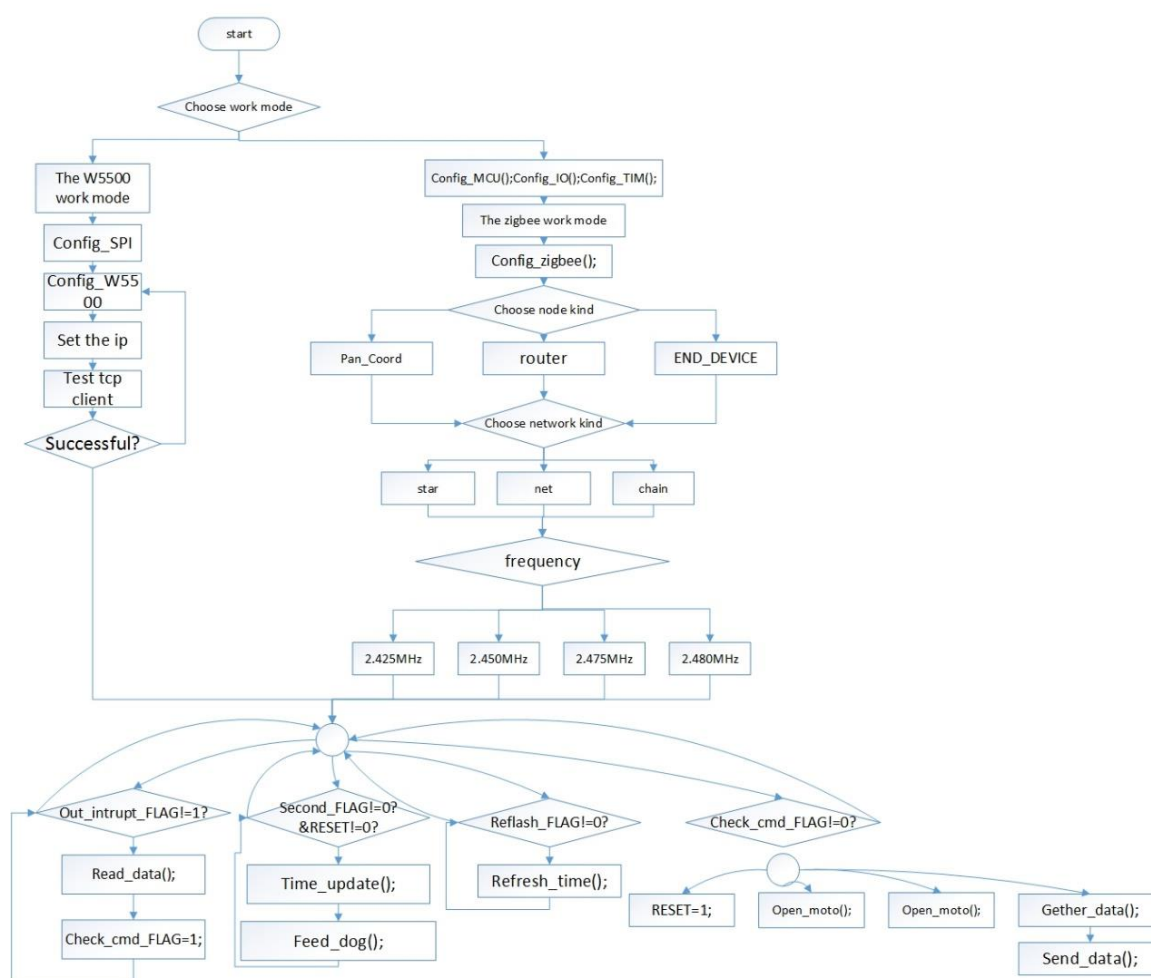


图 4-1 程序整体流程

4.1.1 ZigBee 工作模式模块可分成以下几个子模块：

- 1) ZigBee 参数读取模块：读取拨码开关的电平读取配置参数；
- 2) ZigBee 参数配置并初始化模块：进行配置 ZigBee 并进行初始化；
- 3) 初始化定时器计数器，内部中断，外部中断；
- 4) 初始化节点参数模块：存储每个节点的 NetID, NodeID 以及传输波特率；
- 5) 正常工作模块：接收指令收发数据，控制电机等。

4.2 程序的具体实现过程

1. 初始化模块：初始化延时函数，并且使能所用引脚的时钟将控制引脚设置为推挽输出，将读取拨码开关引脚设置为下拉输入，将 ARM 连接 ZigBee 的 RX 与 TX 引脚分别设置为推挽输出与下拉输入，初始化时钟工作频率为 72MHz，初始化计时器

使得计时器 1s 产生一个内部中断，计算公式为： $t = \frac{1s}{boardrate \times 8}$

2. 初始化外部中断，设置中断优先级，考虑到数据接收的时间较短，因此设计外部中断的优先级高于内部中断，则两个中断互不影响。在此项目中外部中断先抢占主优先级 1，再抢占次优先级 1，内部中断抢占主优先级 1，在抢占次优先级 2。

3. 读取参数模块：依次读取每一个拨码开关的电平，当开关闭合时读取到的是低电平，开关闭合时，读取到低电平，每个拨码开关对应着二进制参数的一位，一个字节为八位，通过将读取的值进行位操作，即将该数据的第 i 位左移 i 个单位，将移位得到的结果用或操作连接，再与上位宽，起到截止数据的作用。读取的数据为以下：

- 1) 网络 ID
- 2) 节点 ID
- 3) 网络类型
- 4) 节点类型
- 5) 发送模式
- 6) 通讯波特率
- 7) 校验位
- 8) 无线频点

4. ZigBee 工作模式：根据 Zigbee 模块的使用要求，输出 3s 低电平信号到 ZigBee 的 config 引脚，进行 ZigBee 配置参数，将上述读取到的参数发送至 ZigBee，初始化 ZigBee 的工作模式。

5. UART 串口通讯的实现：

UART(通用异步收发传输器)串口通讯原理：本项目采用全双工的工作模式，全双工工作模式如图所示：

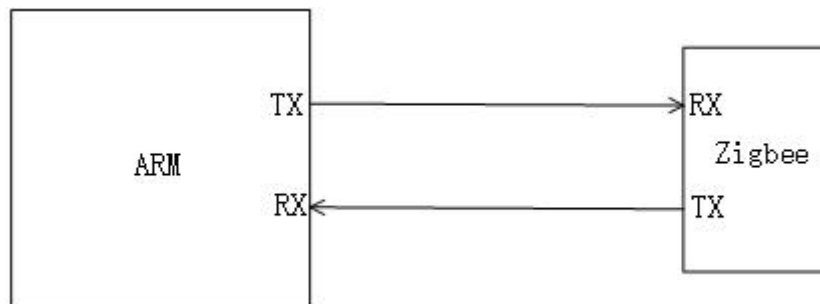


图 4-2 全双工工作模式

数据收发的具体格式为:

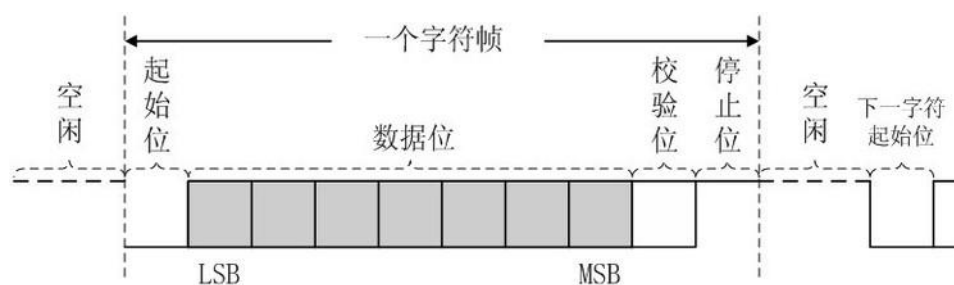


图 4-3 数据收发的格式

起始位诱发外部中断，进入接收数据接收模式，根据波特率进行周期采样，计算公式为:

$$t = \frac{1s}{\text{boardrate} * 8}$$

延时 t ，周期性进行采样，利用位操作和或操作将 8 个比特数据换算成 16 进制代码，尽量使得采样点落在数据位的中心位置，防止在下降沿或者上升沿时进行采样减少误码的概率，每次进入数据接收模式需要关闭外部中断，防止由于进入中断函数造成误码，当收完数据之后关闭外部中断。

当需要进行发送数据时，将数据进行右移操作，并与 0x01 即将每一位数据分开，若结果为 1，则发送持续时间为 t 的高电平信号，反之，则发送持续时间为 t 的低电平信号，直至顺序发送完 8 个比特位。

6. 正常工作模式的实现:

外部中断服务函数作用:更改相应的 flag 进入所对应的程序块中。

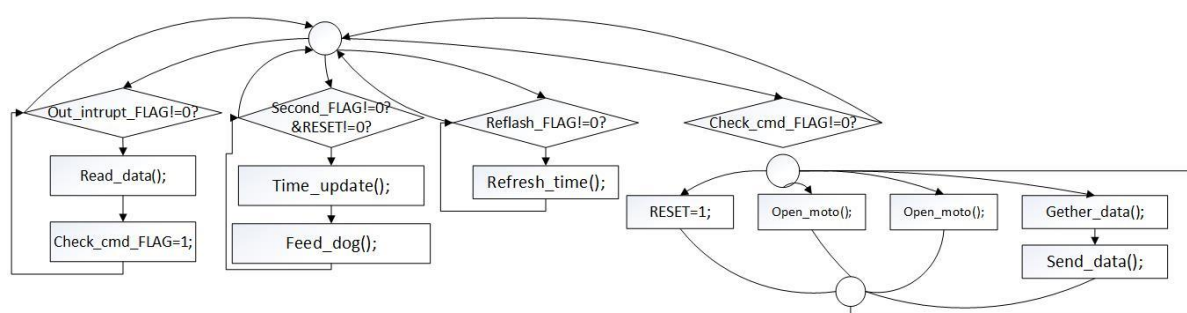


图 4-4 流程图

外部中断产生时程序进入到外部中断的中断服务函数中去，将 Out_input_Flag 设置为 1，进入数据接收程序块，关闭外部中断开始读取由 ZigBee 发送过来的数据，读取完成之后，打开外部中断，当接收的数据大于 3 个字节时，将 Check_cmd_Flag 设置

为 1。进入读取指令模式中如图，先核对节点 ID 和网络 ID，全部一致则证明该指令为当前节点的控制命令，读取指令，并按照指令执行相应的操作。

定时器中断服务函数作用:更改 `second_Flag`,使得程序进入到更新时间的程序块中更新存储的系统时间同时读取电机接触器电平，判断电机是否工作，若工作，则同时开始更新电机开启时间，电机开启时间每天进行依次清零。

看门狗电路作用：为增强系统鲁棒性，如果程序跑飞，看门狗会由于没有清零而产生溢出，溢出会导致产生芯片的重启。因此我们也可以利用看门狗的这一性质进行重启芯片，从而提高系统的可靠性。

4.3 实验过程中遇到的困难

1. 波特率理论计算与实际由于数据类型不同需要基于理论计算并多次实验确定，但数据收发过程不可见。无法知道是否收发成功，采样周期时间的长短不同。
2. 实验过程中产生了由于假溢出而产生的 bug，但由于是硬件调试，难以判别。
3. 实验过程中节点正常运行一段时间后停止发送接收，不知道是数组越界所致，还是由于程序运行错误。

4.4 解决方案

1. 采用示波器，观察数据收发波形，判断参数设置是否正确，然后利用 TTL 转 USB 模块将串口通讯转为 USB 通讯，与电脑进行数据收发，判断数据收发是否成功。
2. 程序运行过程中输出一些信息到 PC 端，根据信息判断程序运行情况。通过第二种方案实验发现是由于数组错位所致，程序中固定数据接收在数组中的位置。

5 服务器开发及数据交互接口设计

为了使系统各个部分的数据能够以相对应的正确的格式在系统各个部分流畅地传递和被调用,我们考虑用 Python 编写 TCP Server 用以与终端建立连接接收和发送数据,并且编写相对应的数据交互接口用以处理数据并转化为对应的格式以待发送。

5.1 TCP/IP 服务器

TCP 是面向连接的通信协议,通过三次握手建立连接,通讯完成时要拆除连接,由于 TCP 是面向连接的所以只能用于端到端的通讯。

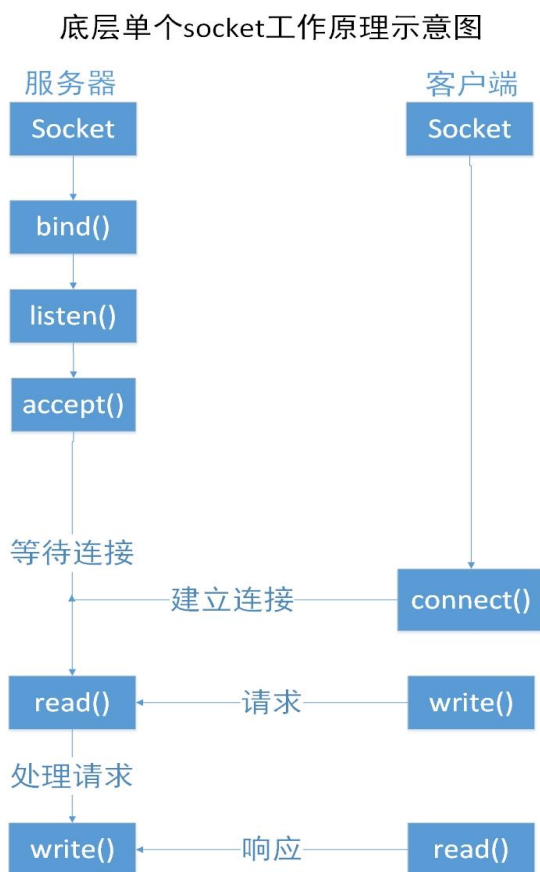


图 5-1 单个 TCP 连接示意图

5.1.1 TCP Server 基本工作原理

- 首先服务器端有一个监听线程，不断监听来自客户端的连接。
- 当一个客户端连接到监听线程后，便建立了一个新的连接。监听线程将新建立的连接放入连接池进行管理，然后继续监听新来的连接。
- 线程池中有多个服务线程，每个线程都监听一个任务队列，一个建立的连接对应一个服务任务，当服务线程发现有新的任务的时候，便使用此连接向客户端提供服务。
- 一个 Socket Server 所能够提供的连接数可配置，如果超过配置的个数则拒绝新的连接。
- 当服务线程完成服务的时候，客户端关闭连接，服务线程关闭连接，空闲并等待处理新的任务。连接池的监控线程清除其中关闭的连接对象，从而可以建立新的连接。

TCP Server 整体布局示意图

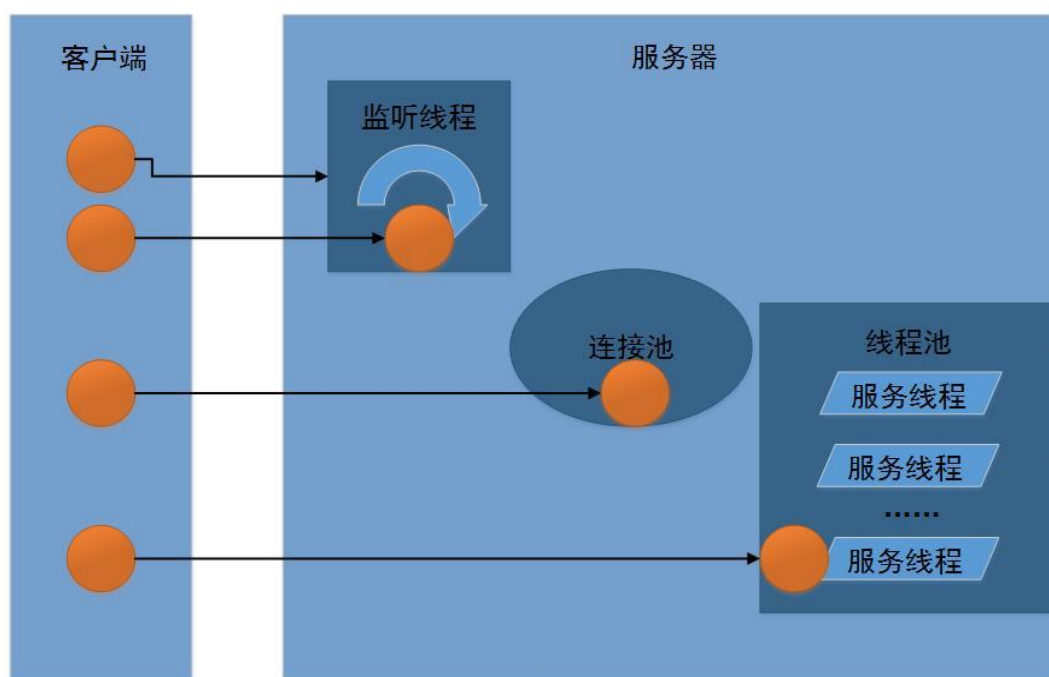


图 5-2 TCP Server 整体布局示意图

5.1.2 具体实现过程

- 实现过程:

依赖库: 在具体代码实现是我们采用的是 Python 中的 `socketserver` 库。`SocketServer` 简化了网络服务器的编写。它有 4 个类: `TCPServer`, `UDPServer`, `UnixStreamServer`, `UnixDatagramServer`。这 4 个类是同步进行处理的, 另外通过 `ForkingMixIn` 和 `ThreadingMixIn` 类来支持异步和多线程

连接释放阶段“死锁”现象示意图

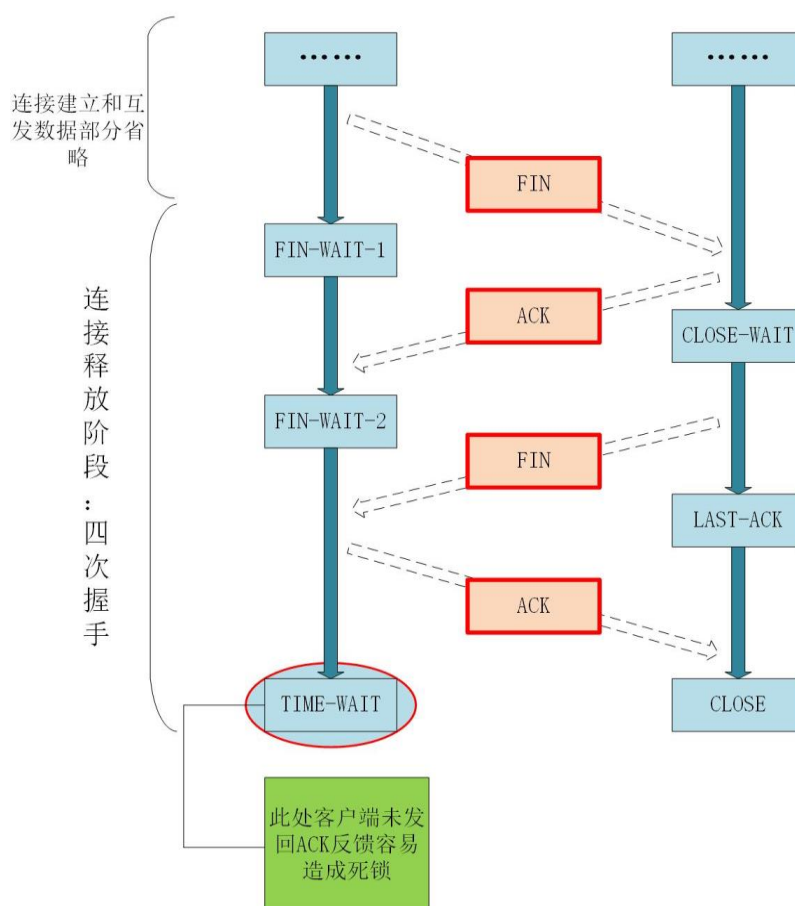


图 5-3 连接释放阶段“死锁”示意图

- 原因:

- 其对 `socket` 进行了良好的封装, 便于我们快速的定制和实现一个 `TCP Server`。
- 在 `TCP Server` 与 `Client` 断开连接时必然要进行下图所示“四次挥手”相互确认后方能确实地断开, 然而, 这很可能因为网络阻塞等突发状况造成“死锁”, 大大降低了 `TCP Server` 的稳定性, 而用 `socketserver` 库我们可以很方便地解决这一问题而不触及底层。

- 增加功能模块：

- ◆ 获取客户端的 IP 和 port

通过在 `ThreadedTCPRequestHandler` 类中改写 `setup` 函数，从而在 TCP 建立连接后打印「<IP>:<port> is connect!」信息出来并获取客户端的 IP 地址和端口信息。

同时将获取到的客户端（DTU）IP 和 port 记录进一个列表并根据其发送的心跳包信息进行标记其 `net_id`，从而在处理数据完成后可以以此将数据发送给相应 DTU 连接下的 ZigBee 节点。

- ◆ 保持 TCP 长连接和自动重连

在 `ThreadedTCPRequestHandler` 类中的 `handle` 函数中添加 `while` 循环，同时修改代码为：先判断缓冲区是否有数据，有数据才进行响应。

由于使用花生壳进行端口映射，考虑到其存在的不稳定性，即软件掉线的情况，在 `while` 循环中添加一旦 TCP Server 掉线立即保持尝试重连状态，直到重新启动。

- ◆ 自动获取本机 IP

考虑到是动态分配 IP，出于自动化的考虑，添加了读取本机的 IPv4 地址的脚本，读取后传递给 TCP Server 作为其 IP。

- ◆ 多线程

改写 `ThreadingMixIn` 类以实现了核心的线程化功能，用于与服务器类进行混合(mix-in)。

5.2 与 DTU 数据交互

1) 目的:

对与 DTU 连接的各 5 个子网下的 100 ZigBee 节点进行周期性的轮询，轮训周期参数化，根据 ZigBee 的覆盖范围等信息计算同时经过测试后将从上一节点到下一节点的周期定为 3 秒。

2) 工作过程:

1. 连接及记录

建立 TCP Server 后，DTU 与其连接同时发送心跳包，即 DTU 所对应网络 id 号，此时 TCP Server 记录下其 IP 和 port 和网络 id 号。

2. 轮询

a) 初始情况

开始对各个 ZigBee 节点进行轮询，轮询 request 发送格式为：
[node_id, net_id, command], 如 "01 01 ff"。如果收到反馈，其格式应该为 [node_id, net_id, work_status, second, minute, hour], 其中后三个参数为工作时长，其值是以十六进制发送，相关转换在另一格式转换函数中进行。

b) 错误处理

如果在设定周期（即 3 秒）内没有得到反馈或者由于网络阻塞等原因出现数据截断或重叠，则开始重发，重发三次为仍未得到反馈对下一个节点进行轮询

3) 数据处理

a) 正常工作

轮询得到及时反馈则说明正常，则在收到的反馈数据后加入标签 flag（为 0）后导入数据库，标签为使数据库便于判断和存储不同数据。
导入格式为： [node_id, net_id, work_status, second, minute, hour, flag]

b) 故障

出现 3 次重发为得到反馈或仍有数据截断，则判断此节点为故障，

同时将工作时长全部置为 0，添加故障 flag（为 1）后导入数据库。

导入格式为：[node_id, net_id, work_status, 0, 0, 0, flag]

4) 具体实现：

由于存在周期性检测时间和检测是否收到反馈两个独立的过程，故考虑采用 Python 中的 `threading` 模块加入多线程从而省去诸多烦杂的标签，同时考虑到占用冲突的问题，两个线程传递不同的标签，在主线程中以时间标签为优先进行判断。

5.3 与数据库交互

1) 实现功能：

- a) 节点反馈信息导入
- b) 定期下载数据库数据形成每天，每周，每月，每年的 Excel 数据表以供查看分析。

2) 具体实现

- c) 导入，通过 `mysql` 库这一数据库与 Python 的 API 接口连接数据库，进行插入操作。
- d) 下载，用 Python `sys` 库中的 `datetime` 模块读取系统时间，设定下载间隔，后持续检测时间，当到设定时间，通过 `mysql` 库连接数据库，下载相应数据创建并存入 Excel 表格。

5.4 与 Web 后端交互

1) 实现功能：

远程操控

2) 具体实现

用户用终端（电脑，手机等）访问网页时，点击重启或开启/关闭等按钮，Web 后端会在本地创建一个 `txt` 文件并写入相应的操作信息，格式为：
[node_id,net_id,command],（如“01 01 cc”），而服务器本地的 Python 脚本会持续检测此文件中是否写入内容，若写入则会转换格式后立即发送给相应子网的节

点，其优先级高于时间检测。

5.5 遇到的问题

- 1) 开发最初，数据处理与数据收发为分裂的两个方面，数据收发用“NetAssist”软件作为 TCP 服务器将收到的数据和待发送的数据分别存入不同的 txt 文件，再经由 Python 脚本在本地实时判断修改。
 - (a) 此软件存在不稳定性，工作一定时间会断开而停止工作，且未提供接口可调试性差
 - (b) 两个过程分裂容易造成时间延误
- 2) 数据处理中存在检测时间，检测反馈，检测远程控制信息三个相对独立过程，采用单线程造成了程序的复杂，不稳定和不易修改

5.6 解决方案

- 1) 自主编写 TCP 服务器，即可使两个方面合并。
- 2) 采用多线程，并用不同标签设定优先级，以避免占用冲突的问题。

6 数据库开发

数据库开发是该智能监控系统的一个核心部分，它的作用是将硬件部分的 DTU (Database Transmit Unit) 传输到服务器的数据管理和存储在一个具有明确结构的数据处理系统中，并与 web 端交互，将其以特定的形式显示在网页的各个界面中。在该项目中，我们使用的是 Oracle 公司旗下的关系型数据库管理系统--MySQL，因为其开源、免费、稳定性好、使用广泛。

6.1 数据库结构设计

最初的设计中数据库的结构有三种选择：多个数据库对应多个表格、一个数据库对应多个表格、一个数据库对应一个表格。考虑到数据库管理的方便性和数据搜索查询的即时性，和经过多次修改调试后的数据库结构稳定性，我们最终确立的是一个数据库 (receiver) 对应一个表格(control)的结构，并根据实际情况修改数据库的搜索引擎和搜索频率较高的元素项添加索引以再次提高数据搜索的速度。

- 1) 元素：① node_id (电机的节点 ID)；② net_id；③ (电机的网络 ID) motor_time (电机当天工作时长)；④ motor_status (电机的即时工作状态)；⑤ flag (电机是否当天出现过故障)；⑥ cur_date (电机工作的日期)。
- 2) 引擎：MyISAM；不采用常见的 InnoDB 引擎的原因是 InnoDB 引擎响应速度较慢，而在本项目中，要求数据库能够对查询请求快速响应。故在不大幅增加服务器性能的前提下，采用 MyISAM 引擎能够使请求快速得到响应。
- 3) 添加索引项：motor_status;motor_time;flag：此两键均为较常用的查询键，将其设为主键并添加索引可减少响应时间。

| # | 名字 | 类型 | 排序规则 | 属性 | 空 | 默认 | 额外 | 操作 |
|---|--------------|-------------|------|----|---|----|----|--|
| 1 | node_id | tinyint(11) | | 否 | 无 | | | 修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT) |
| 2 | net_id | tinyint(11) | | 否 | 无 | | | 修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT) |
| 3 | motor_status | tinyint(11) | | 否 | 无 | | | 修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT) |
| 4 | motor_time | time | | 否 | 无 | | | 修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT) |
| 5 | flag | tinyint(11) | | 否 | 无 | | | 修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT) |
| 6 | cur_date | date | | 否 | 无 | | | 修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT) |

全选 选中项: 浏览 修改 删除 主键 唯一 索引 全文搜索

图 6-1 数据库结构

6.2 数据库数据的导入和实时更新

数据库中电机信息数据的导入是利用 Python 程序对 TCP 服务器接收到硬件部分发来的数据包解析后即时插入到数据库中的，并且根据 5 个网络共 500 个节点大约每隔 10 分钟轮巡一次的设定，数据库当天电机数据每十分钟实时更新一次。

| node_id | net_id | motor_status | motor_time | flag | cur_date |
|---------|--------|--------------|------------|------|------------|
| 1 | 1 | 1 | 14:20:05 | 1 | 2016-01-01 |
| 2 | 1 | 0 | 16:46:31 | 0 | 2016-01-01 |
| 3 | 1 | 2 | 19:25:17 | 0 | 2016-01-01 |
| 4 | 1 | 1 | 19:58:56 | 1 | 2016-01-01 |
| 5 | 1 | 1 | 20:30:23 | 1 | 2016-01-01 |
| 6 | 1 | 0 | 22:01:21 | 0 | 2016-01-01 |
| 7 | 1 | 1 | 15:58:32 | 1 | 2016-01-01 |
| 8 | 1 | 0 | 14:27:14 | 1 | 2016-01-01 |
| 9 | 1 | 0 | 22:09:13 | 1 | 2016-01-01 |
| 10 | 1 | 1 | 18:59:29 | 0 | 2016-01-01 |
| 11 | 1 | 0 | 16:51:25 | 0 | 2016-01-01 |
| 12 | 1 | 2 | 21:12:09 | 1 | 2016-01-01 |
| 13 | 1 | 2 | 15:38:22 | 0 | 2016-01-01 |
| 14 | 1 | 1 | 23:12:46 | 0 | 2016-01-01 |
| 15 | 1 | 0 | 16:17:11 | 0 | 2016-01-01 |
| 16 | 1 | 2 | 17:11:33 | 0 | 2016-01-01 |
| 17 | 1 | 2 | 21:54:05 | 1 | 2016-01-01 |
| 18 | 1 | 0 | 21:48:31 | 0 | 2016-01-01 |
| 19 | 1 | 0 | 22:25:09 | 1 | 2016-01-01 |
| 20 | 1 | 2 | 21:45:22 | 0 | 2016-01-01 |

图 6-2 数据库数据显示

7 web 开发

物联网的核心和基础是互联网，是在互联网基础上的延伸和扩展的网络，所以在互联网+的大时代潮流下，为了更好地增加用户的体验度，硬件的实际应用需要与互联网更加紧密地结合起来。在我们的项目中，web 的开发主要包括网站的构建、网页后端的开发、网页美工设计和网页前端开发。

7.1 网站的构建

7.1.1 建站的目标

以本次我们的实际应用（中铁十八局新建京张铁路降水自动化监控系统）为例，建站的目的是为了实现一种更加直观的数据显示方式和更加方便的实时监控方式，给用户一种更直观的体验，而 PC 端和移动端的分开设计也是为了适应不同用户和不同设备显示的需求。

7.1.2 规划域名

由于该网站只限专业的管理人员使用，暂时不具普遍性，且为了开发的方便、节省开支，我们在本次的实际应用中并没有真正的去申请一个具有实际意义的域名，而是利用一个动态域名解析软件，将一个电信宽带的动态 IP 包装成一个静态域名，以实现网站的上传和访问。

7.1.3 网站策划

该网站主要是用于配合硬件部分数据的可视化和对硬件的控制，所以以本次的实际应用为例，我们分四个部分来规划整个网站：节点信息、网络信息、实时监测和全局视图。

① 数据的可视化中：节点信息和网络信息页面分别展示单个节点和单独的整个网络（100 个节点）的所有电机的工作信息统计信息，并且加上节点选择、网络选择、近期查询、时间查询、前一周、前一个月和前一年的电机工作信息汇总（折线图、柱状图可以相互转换）的功能；而全局视图页面则是能实时刷新显示最近

10 分钟内电机所处的工作状态（每隔十分钟所有节点全部轮巡一次）。

② 对硬件的控制：实时监控页面有对单个节点的控制功能，包括远程操控电机的关闭和开启，电机的 RESET（复位），显示该电机最近时间内故障次数。

7.1.4 硬件平台搭建

① 服务器：联想 think station;

② 现场大型显示屏：监控室采用 12 块显示屏拼接而成的大显示屏（1920x1080）。



图 7-1 服务器 Think Station

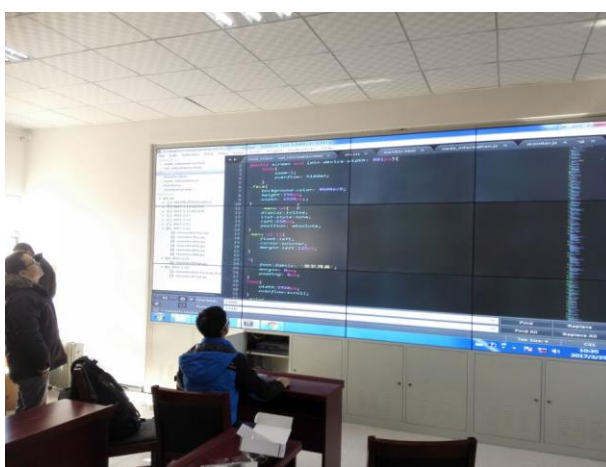


图 7-2 监控室和 12 块合成显示屏

7.1.5 选择软件实现平台

① 动态域名解析软件：花生壳

② 网页开发调试软件：Chrome 浏览器

③ 网站开发编辑软件：Dreamweaver

④ 数据库软件：MySQL 5.6

⑤ 数据库可视化管理工具：PHPMysqlAdmin

⑥ PHP 网页建站工具组合包：Appserv

7.2 网页后端开发

7.2.1 网页后端与前端交互

网页后端开发主要运用了 PHP 编程语言进行开发，与网页前端的交互主要运用了 Ajax 网页异步更新技术，通过接收前端的请求（解析字符串），在数据库中搜索所需要的信息，然后将数据打包成 JSON 数组的形式传递给前端。

7.2.2 网页后端与 MySQL 数据库交互

网页后端与数据库的交互主要运用了 PHP 和 MySQL 脚本语言。网页后端收到了前端的请求后，即在 PHP 文件中调用 MySQL 脚本语言在数据库中查找前端所需要的数据信息并以数组的形式返回给后端。

7.3 网页美工设计

网页的美工设计包括 PC 端网页设计和移动端网页设计两个部分，可适应 PC 端和手机移动端浏览器显示屏的不同分辨率，两者设计的功能完全相同。

由于网页的功能主要是数据的呈现，所以本次网页设计采用的是简洁、商务的设计风格，以蓝色调的图片为背景，基本采用黑色微软雅黑字体，按钮采取悬浮式，数据部分采取折线图（和柱状图，可转换）和饼图的方式呈现，全局视图的各个电机即时工作状态采用不同颜色的小圆点展示，使得用户能有较好的直观体验。

7.4 网页前端开发

7.4.1 网页静态开发

- 1) 编程开发语言：HTML、CSS
- 2) HTML 主要用于设计整个网页的内容和布局，而 CSS 主要用于对各个部分进行修饰和排版，如字体、线宽、颜色，大小，间距.....

7.4.2 网页动态开发

- ① 编程开发语言：JavaScript
- ② 在网页的动态开发中 JavaScript 主要用于给按钮等事件赋予动态效果，如悬浮动态，点击事件.....

7.5 网页前端与后端交互

- 1) 技术: Ajax 网页异步更新技术
- 2) 传统的网页(不使用 AJAX)如果需要更新内容,必须重载整个网页页面网页后端开发,而 Ajax 技术则通过在后台与服务器进行少量数据交换,可以使网页实现异步更新,其中前端的请求以字符串形式发到后端,后端的数据则以 JSON 数组的形式传回前端,实现前后端的交互。



图 7-3 PC 版登录界面



图 7-4 PC 版节点信息界面

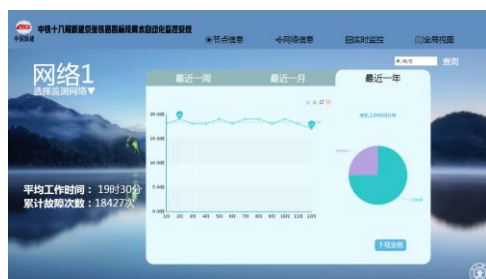


图 7-5 PC 版网络信息界面



图 7-6 PC 版实时监控界面



图 7-7 PC 版全局视图界面



图 7-8 移动版全局视图界面

8 移动端 App 开发

随着智能手机在大众的日常生活中使用越来越频繁，移动端 App 开发也是一个互联网+潮流中的一个比较突出的趋势。且用户对移动端 App 的使用比移动端网页具有更好的体验，于是我们同时计划了安卓移动端 App 的开发。而本项目中 App 的开发包括需求分析、管理平台开户、App 框架输出、App 内容架构组织、UI 设计及制作、App 程序测试调优、App 应用上线发布。而且 App 功能并非单一，可将多个不同的物联网应用安置其中，在未来万物联网的时代保证用户一个 App 即可掌控一切信息，监控所有的系统。目前，App 仍在开发中，预计 4 月初设计完成。

9 系统评测与后期改进

9.1 系统评测

该项目（基于物联网技术的智能监控系统）的设计包括主要嵌入式开发和系统软件两个部分，具有很强的可移植性，能根据不同的用户需求即时改进设计并应用。其中硬件的设计可很方便地增加外拓，例如摄像头、传感器、通讯设备……不断地强化其功能；而软件开发中的数据库设计、网站开发、App 开发体现了当下最潮流的互联网+的思想—用户体验至上，整个系统的设计也体现了我们的两大设计初衷--创新和实用。

9.2 改进

- a) 硬件的设计中电源体积较大，可采用同功率的体积较小的电源（价格较贵）代替；
- b) 服务器和硬件部分的 DTU 数据交互部分较为复杂，可利用 C#或 GUI 进行图形化界面设计，将其封装起来，以提高系统的稳定性；
- c) 网页前端数据的显示还未实现完全意义上的动态即时刷新显示；
- d) 网站和数据传输的安全性较低。

10 实际应用与推广

10.1 实际应用

10.1.1 项目简介

京张高铁东花园隧道段，由于地下水位较高（近官厅水库），为了保障施工建设的顺利进行，拟实施水位控制及远程监控的信息自动化解决方案。

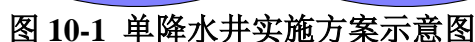
10.1.2 项目需求

该段隧道长度约 5 公里，隧道开挖宽度约 50-75 米，基坑两侧设置间隔 6 米的双排降水井，共计约 3000 降水井。降水井设计深度 25-30 米，孔径 600 毫米，井管外径 400 毫米，内径 300 毫米。每个降水井配置电箱一台，水泵一台，自动控制系统一套。水泵的工作启停状态根据降水井内水位的相对位置进行自动控制，即：当降水井内液面水位超过警戒线高度时，打开电源控制开关，启动水泵，将液体抽出，排放在泻流水渠，待降水井内液面水位低于安全线高度时，关断电源控制开关、关闭水泵。实现无人值守的自动化水位控制，监控室可实时的监控全部降水井的工作状况。

除水泵工作状态根据水位进行自动化启停外，还应当对于水泵的工作状态进行实时监测，并将水位与警戒线和安全线的相对信息、水泵启停工作状态、水泵故障等信息传送到监控室，维护人员可根据实时监控信息及时的精确维护。

10.1.3 项目技术方案

根据项目实际情况以及需求，定制技术方案如下：



降水井内安置液位传感器，进行实时液面监测，根据监测结果与预设的警戒和安水位进行对比，从而通过水泵的电源开关完成水泵的工作启停控制。在水泵工作过程中，需实时监测液面的变化。当水泵发生故障时，还可以将该信息及时反映到远程终端，形成警示信号，从而提醒工作人员进行人为介入，排除故障。单降水井传感器采用直流 3.3V 供电，需从水泵供电系统通过交流/直流转换进行取电。

采用电缆浮球液位开关，使用磁力运作，无机械连接件，运作简单可靠，性价比高。电缆式浮球开关，利用重力与浮力的原理设计而成。主要包括浮漂体，设置在浮漂体内的大容量微型开关和能将开关处于通，断状态的驱动机构，以及与开关相连的三芯电缆。当浮球在液体浮力的作用下随液位的上升或下降到与水平呈一定角度时，浮球体内的驱动机构——驱动大容量微动开关，从而输出开（ON）或关（OFF）的信号，共报警提示或远程控制使用。

将液位传感器收集的信息,通过电线输送到嵌入式微控制电路板,电路板嵌入式

MCU 负责根据数值的比较进行水泵开关电源的控制，并通过支持工业级无线网络协议 (Zigbee) 的模块，将单降水井的水泵工作状态和水位信息通过在降水井顶部安置的微型天线发送到网络中枢节点。

嵌入式微控制器电路板采用直流 3.3V 供电，从水泵供电系统通过交流-直流电源转换模块(380VAC-220VAC)，完成电压转换，之后通过直流电源开关转化为 24VDC 和 5VDC，之后经过电源模块转换 5VDC->3.3VDC 进行取电。

d) 网络方案

系统网络的方案充分考虑到该项目的示范性效应以及未来可扩展、可移植、可升级的实际需求，定制项目具体方案如下：

e) 网络拓扑结构

由于隧道跨越距离较长（5km），节点数目众多（3000），呈并行双带状分布，因此，拟采用 2 级网络的方案设计：第一级网络规模支持 100 节点（降水井），第二级网络支持 5 节点的网络方案。每个节点（降水井）对应唯一地址 ID，系统网络系统软件可以根据节点 ID 定位到具体的降水井位置，以及降水井的工作状态。为系统的正常监控和后期维护提供信息。

二级网络通过构建 2G/3G/4G 移动网络进行组织，节点需配置 SIM 卡，并开通数据流量功能，实时传输网络状态数据到监控室的服务器。

监控室服务器端系统软件包含数据库软件，可以进行日、月、年水位记录，水泵电机的工作时间等数字化信息记录。

f) 远程监控方案（后期升级方案）：

利用 2G/4G 移动互联网，设计完成手机控制端 APP，可以通过该软件实时远程监控各降水井的状态，以及进行相应的控制操作。

10.2 项目推广

本作品作为一款基于物联网技术的智能监控系统，拥有很大的推广应用空间。在当今高速网络通讯时代，智能监控技术可以大面积应用于恶劣偏远环境或无人监管环境当中，通过物联技术实现对各种信号的采集与汇总，最终通过有线或无线网络实现远距离监控，保证高效监控性的条件下同时兼顾数据采集的准确性，保证在突发情况下可以使控制端迅速作出反应。

- ① 智能安防：实时监测摄像头镜头的画面若为静态，则不拍摄，若有动态情况出现才拍摄并存储资料，以减少电量消耗和空间存储。
- ② 智能农业灌溉：通过特定传感器智能感应农作物的干旱情况并在农作物需要水分的时候自动灌溉，可减少人工灌溉的成本且利于农作物的成长。
- ③ 智能路灯：通过声学传感器、光学传感器或其他传感设备感知附近人的情况，没人或天亮的情况下自动处于熄灭状态，在感应到附近有人且天较阴暗的情况下自动亮灯，能减少对电力的浪费。
- ④ 智能交通：智能感知交通情况，在非常规情况下实现红绿灯的自动转换以减少交通堵塞。
- ⑤ 除此之外该系统在工业监测、环境监测、情报搜集.....均能有所应用。

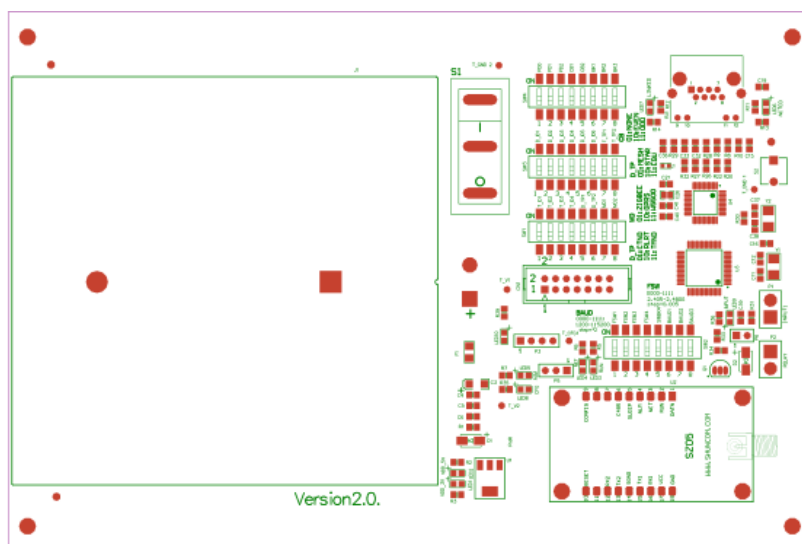
参考文献

- [1] 刘火良, 杨森等. STM32 库开发实战指南[M]. 北京:机械工业出版社, 2013.
- [2] 刘军, 张洋, 严汉宇, 左忠凯. 原子教你 STM32 寄存器版[M]. 北京:北京航空航天大学出版社, 2015.
- [3] 李刚. 疯狂 Ajax 讲义 jQuery/Ext JS/Prototype/DWR 企业应用开发实战[M]. 北京:电子工业出版社, 2013.
- [4] Jin, Yang;Lei, Song;Adam, koeppe. LTE Field Performance for IoT Applications[J]. Vehicular Technology Conference(VCT-Fall), 2016, (84): 1-5
- [5] Petter, Maki;Sampsa, Rauti;Shohreh, Hosseinzadeh. Interface Diversification in IoT Operating Systems[J]. Utility and Cloud Computing(UCC), 2016, (9): 304-309
- [6] Narmada, Alaparathi;P, Sudhakara, Rao. Adaptation layer towards integration of Zigbee and IP stacks[J]. Automatic Control and Dynamic Optimization Techniques(ICACDOT), 2016, (10): 212-215
- [7] Yong-Wei, Li;Ying, Pang. Based on the Zigbee greenhouse grey trend prediction control[J]. Machine Learning and Cybernetics(ICMLC), 2016, (89): 794-798
- [8] huabiao, Li;Chong, Li;Kai, Nan;Zhishi, Zhu. A high efficient Application communication solution for GPRS-DTU[J]. Advanced Computer Control(ICACC), 2011, (3): 7-11
- [9] Lin, Lihua;, Xie, Xiufeng;Tang, Shancheng;Zhao, Anxin. The Application and research of embedded web server technology in smart grid DTU[J]. Signal Processing, Communications and Computing(ICSPCC), 2015, (10): 1-5
- [10] STM32F103CB_Mainstream Performance line, ARM Cortex-M3 MCU.[S] <http://www.st.com>
- [11] SZ05-ADV 无线数传模块用户手册.[S] <http://shuncom.com>
- [12] 顺舟 GPRS DTU 用户使用手册.[S] <http://shuncom.com>
- [13] JZX-22F 小型电磁继电器.[S] <http://shuncom.com>
- [14] NC1 交流接触器.[S] <http://shuncom.com>
- [15] NKY1 开关电源.[S] <http://shuncom.com>
- [16] TI_LM1117DT-3.3-datasheet.[S] <http://www.ti.com>
- [17] 施瓦茨等. 高性能 MySQL[M]. 王小东等. 北京: 电子工业出版社, 2010.1.
- [18] 姜承尧. MySQL 技术内幕-SQL 编程[M]. 北京:机械工业出版社, 2012.
- [19] 周润景等. Altium Designer 原理图与 PCB 设计[M].北京:电子工业出版社, 2015.
- [20] 郑亮. 嵌入式系统开发与实践——基于 STM32F10x 系列[M].北京:北京航空航天大学出版社, 2015.
- [21] 朱正东等. 数字逻辑与数字系统[M]. 北京:电子工业出版社, 2015.
- [22] 杨栓科等. 模拟电子技术基础[M]. 北京:高等教育出版社, 2010.
- [23] 明日科技. PHP 从入门到精通[M]. 北京: 清华大学出版社, 2016.

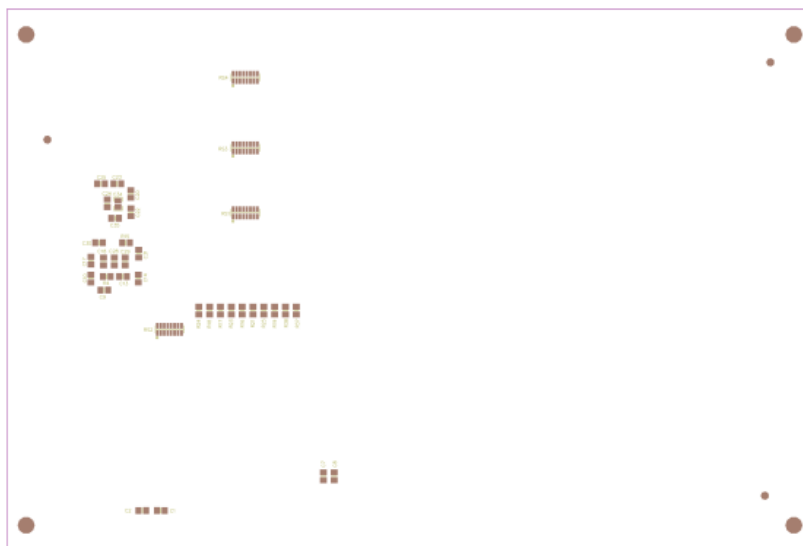
- [24] Luke welling,Laura Thomson. PHP and MySQL Web Development[M]. Beijing: China Machine Press,2009.
- [25] 软件开发技术联盟. PHP+MySQL 开发实战[M]. 北京:清华大学出版社, 2013.
- [26] 张工厂. PHP+MySQL 动态网站开发从入门到精通[M]. 北京:清华大学出版社, 2017.
- [27] 杜江. PHP 与 MySQL 高性能应用开发 [M]. 北京:机械工业出版社, 2016.
- [28] [美]朱莉, 梅洛尼 (Julie, Meloni). HTML、CSS 和 JavaScript 入门经典[M]. 北京:人民邮电出版社, 2015.
- [29] 刘西杰, 张婷. HTML CSS JavaScript 网页制作从入门到精通[M]. 北京:人民邮电出版社, 2016.
- [30] 赵锋. 网页设计与制作——HTML5+CSS+JavaScript[M]. 北京:清华大学出版社, 2013.
- [31] 孙良军, 胡秀娥. HTML+CSS+JavaScript 网页设计与布局实用教程[M]. 北京:清华大学出版社, 2011.
- [32] 孙波翔, 李斌, 李晗. Python 核心编程 [M] .北京: 人民邮电出版社, 2016.
- [33] 诸豪文. Python 网络编程[M]. 北京: 人民邮电出版社, 2016.
- [34] (挪)海特兰德. Python 基础教程[M]. 北京:人民邮电出版社, 2014.
- [35] [日]BePROUD 股份有限公司. Python 项目开发实战[M]. 北京:人民邮电出版社, 2016.

附 录

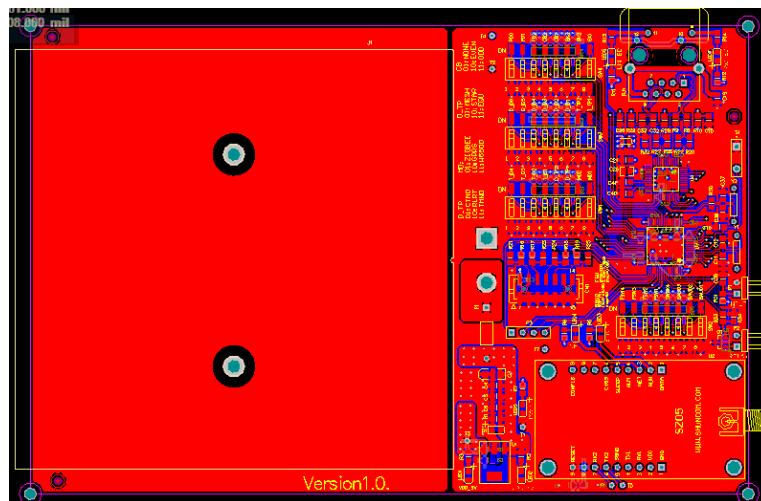
附录 1 PCB 设计图



附图 1-1 PCB 正面



附图 1-2 PCB 反面

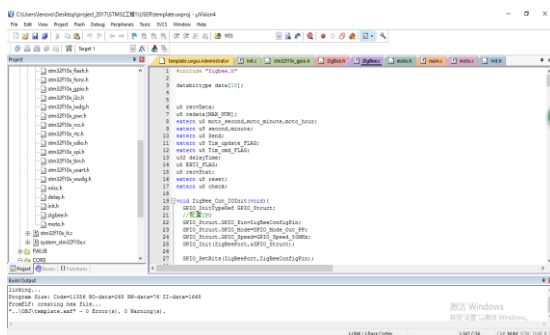


附图 1-3 PCB 电路图

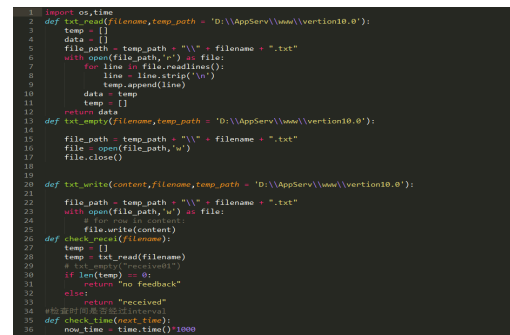


附图 1-4 电路板照片

附录 2 源代码截图



附图 2-1 STM32ARM 编程代码

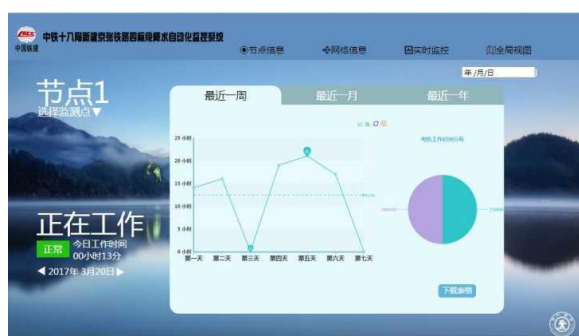


附图 2-2 数据交互接口开发 Python 代码

[illegible]

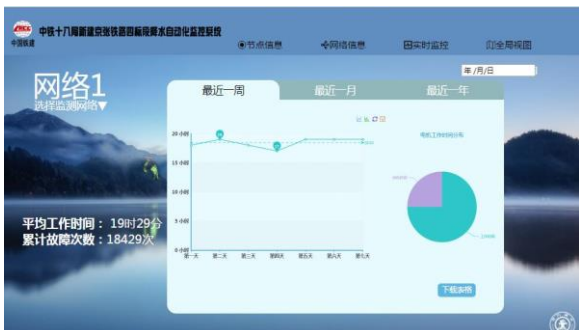
附图 2-4 网页后端开发 PHP 代码

附图 3-1 五个网络 20 个节点测试



附图 3-2 web 端测试

附图 4-1 五个网络 100 个节点测试



附图 4-2 web 端测试

附录 5 实验室调试现场



附图 5-1 实验室工作现场 1



附图 5-2 实验室工作现场 2

附录 6 中铁十八局调试现场



附图 6-1 电机内部结构



附图 6-2 电机组装现场



附图 6-3 现场测试 1



附图 6-4 现场测试 2