分类号 编 号

U D C 密 级



**本科生毕业设计（论文）**

**题 目： 工业物联网数据管理信息系统**

**与终端硬件设计**

**姓 名： 马思清**

**学 号： 11712610**

**系 别： 电子与电气工程系**

**专 业： 信息工程**

**指导教师： 虞亚军**

2021 年 月 日

**诚信承诺书**

1.本人郑重承诺所呈交的毕业设计（论文），是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。

2.除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。

3.本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。

4.在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

作者签名：

年 月 日

工业物联网数据管理信息系统与终端硬件设计

马思清

（电子与电气工程系 指导教师：虞亚军）

[摘要]： 本设计是关于工业企业物联网系统的一套完整搭建方案，涉及设备数据采集，信号处理，信息分析以及信息的可视化应用，涵盖了物联网信息收集、处理、展示 全流程。本方案的具体内容包括一个数据采集硬件设备，一个联接管理软件系统，与一个用户端应用程序。数据采集硬件按照工业标准设计，可采集工业设备的各种数据，实现了多种模拟量与传感器信号的接入，并搭载最新的窄带物联网（NB-IoT）技术，有着良好的无线通信性能。联接管理系统是一个运行在云服务器上的程序，收集各个硬件终端通过互联网上报的消息数据，对其中的信号进行预处理和分析，转储提取出的信息，并监控各个硬件终端的工作状态。用户端应用界面基于WEB技术设计，实现了多种方式的数据可视化展示，并允许用户异地远程监控设备工作情况。整套系统结合了电子技术，信息处理技术和计算机技术，能一次性部署到位，解决了本地设备数据难以联网所形成的信息孤岛问题，实现了工业设备的云监控和工业数据的云储存，解决了因不同厂商的硬件设备、云服务、可视化平台标准不一，难以协同开发的问题，有很好的实用价值和发展前景。

[关键词]：物联网; 智能硬件; 云服务

**目录**

**1.工业物联网系统介绍.............................页码**

1.1 工作原理和模式................................页码

1.2 国内外发展现状................................页码

1.3 产品特点和创新................................页码

**2.系统全局设计.............................页码**

2.1需求分析..................................页码

2.1.1 使用场景与环境................................页码

2.1.2 产品功能及组成................................页码

2.1.3 成本及安全性................................页码

2.1.4 施工与部署特性................................页码

2.2设计方案..................................页码

1.3.1 信息采集................................页码

1.3.2 云端通信和管理................................页码

1.3.3 系统工作模式................................页码

**2.智能硬件终端设计.............................页码**

2.1全局设计..................................页码

2.1.1性能指标................................页码

2.1.2工作模式................................页码

2.2 STM32L431主控模块设计..................................页码

2.2.1初始化配置................................页码

2.2.1.1芯片引脚配置................................页码

2.2.1.2芯片时钟配置................................页码

2.2.2 外部接口................................页码

2.2.2.1 UART配置................................页码

2.2.2.2 SPI配置................................页码

2.3 通信模块设计..................................页码

2.3.1 NB-IoT无线通信................................页码

2.3.1.1 NB-IoT技术简介................................页码

2.3.1.2中移动M5311模组...........................页码

2.3.1.3模组驱动电路设计...........................页码

2.3.2 RS485有线通信................................页码

2.3.2.1 Modbus协议简介..................................页码

2.3.2.2 RS485转换电路设计..................................页码

2.4传感器接入电路..................................页码

2.4.1多通道模拟信号采样电路................................页码

2.4.2.高精度高速信号采样电路.............................页码

2.5外围电路设计..................................页码

2.5.1 电源电路................................页码

2.5.2隔离和防护电路................................页码

2.6固件代码和PCB图纸..................................页码

**3.联接管理系统.............................页码**

3.1全局设计..................................页码

3.1.1 系统工作模式................................页码

3.1.2 软件组成架构................................页码

3.1.3 开发平台与框架...........................页码

3.2设备接入..................................页码

3.2.1.边缘节点接入................................页码

3.2.2 云平台物联网接口接入................................页码

3.2.3云平台直接接入................................页码

3.3数据分析和处理..................................页码

3.3.1数据分析组件................................页码

3.3.2算法定制化开发................................页码

3.4数据转移和储存..................................页码

3.4.1边缘设备缓存................................页码

3.4.2数据共享平台................................页码

3.4.3云端储存................................页码

3.5部署与功能拓展..................................页码

3.5.1云端部署方案................................页码

3.5.2组件升级方案................................页码

3.6 源代码及配置参数..................................页码

**4.客户端应用软件.............................页码**

4.1全局设计..................................页码

4.1.1 系统工作模式................................页码

4.1.2 软件组成架构................................页码

4.1.3 开发平台与框架...........................页码

4.2应用后端开发..................................页码

4.2.1 基于node.js技术的后端开发................................页码

4.2.2 数据流控制................................页码

4.3前端界面开发..................................页码

4.3.1 基于Express的前端开发................................页码

4.3.2 前端交互设计................................页码

4.4代码维护与功能拓展..................................页码

4.5 源代码与配置参数..................................页码

**5.实用案例.............................页码**

2.2.1三级标题(宋体四号)................................页码

2.2.2三级标题(宋体四号)................................页码

2.2.3三级标题(宋体四号)................................页码

2.3三级标题(宋体四号)..................................页码

**参考文献(宋体四号，加粗)...............................页码**

**附录(宋体四号，加粗)...................................页码**

**致谢(宋体四号，加粗)...................................页码**

1. 工业物联网系统

1.1 工业物联网系统介绍

随着电子信息和互联网技术的不断发展，越来越多家居用品或设备器材都集成了物联网功能，可以实现智能化感知、识别和管理。工业物联网系统是物联网技术的一个典型应用，这类系统通过收集和监控各个子设备的信息，可以实现规模化管理，实现末端生产设备的精确监控和控制，同时汇聚海量的设备数据，打破各个生产设备间的数据壁垒。利用工业物联网的强大功能，企业可以有效优化生产过程，高效监管生产设备，获取更多有价值的生产信息，实现精细化管理。工业物联网系统基于信息化浪潮所带来的数字化生产模式，融合先进的通信技术以及更强大的云端算力，在生产数字化基础上实现对通信链路及数据的高效利用，能大大提升生产线的自动化和智能化水平。本设计旨在实现一个具有代表性的全栈工业物联网系统，其内容包括前期数据获取和采集，中段数据分发与解析，以及末端的数据可视化，包含软件和硬件设备。同时，该系统软硬件高度整合，易于部署和施工，拥有良好的拓展性能。

1.1.1工作原理和模式

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.1.2国内外发展现状

物联网系统的设计部署缺乏统一的模式和标准，对于中小企业来说，依靠自身能力很难整合软硬件资源，来设计并搭建自己的物联网系统。一些大型的制造业企业，通过自研的方式为自己的工厂设计了配套的物联网软硬件系统，如美的公司的MeiCloud系统等，但这些系统都是高度定制且不完全开放，很难适用于形态多样的中小微制造企业。一些国内的云服务提供商如华为云、阿里云推出了有关物联网数据储存和物联网应用设计的云服务应用，但没有提供与之配套的硬件设备，在企业缺乏专业信息技术人才的情况下，很难将其与现有的硬件设备进行整合。市场缺乏一种囊括端设备数据采集、设备间通信、云端接入、管理软件和数据应用软件的物联网全栈解决方案。

1.1.3产品特点和创新

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.2 设计需求分析

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.2.1使用场景与环境

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.2.2产品功能及组成

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.2.3成本及安全性

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.2.4施工与部署特性

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.3 全局设计方案

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.3.1信息采集

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.3.2云端通信和管理

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1.3.3系统工作模式

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1. 智能硬件终端设计

2.1 全局设计

智能硬件终端用于采集生产过程中产生的各种数据，轻便可靠，易于大批量安装。设备实际上是一种无线联网的数据采集模块，可以收集多种类型的传感器信号，同时支持人工录入数据。该设备可以以无线方式接入互联网，也可使用串行接口接入厂区现有的有线网络。

2.1.1性能指标

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

**表1 主要设计指标**

|  |  |
| --- | --- |
| 项目 | 设计要求 |
| 体积限制 | PCB面积<0.01m2, 外壳< |
| 成本限制 | 20$ |
| 处理器 | STM32L431RCT |
| 传感器接入方式 | 电流、电压、RS485串行接口 |
| 电流模拟量输入 | 量程4-20mA， 分辨率0.01mA，8通道 |
| 电压模拟量输入 | 量程0-1V，分辨率0.01V，8通道 |
| 高精度采样接口 | Vmax=5V，24bit，1.5MHz |
| 人工输入方式 | 二进制码流 |
| 有线通信 | RS232，RS485（Modbus） |
| 无线通信 | 窄带物联网（NB-IoT） |
| 安全性 | IP67级防护，防雷击电路，防输入过载电路 |
| 拓展模块 | 标准M53接口， OLED显示屏 |

2.1.2工作模式

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.2 STM32L431主控模块设计

主控模块由中央处理器和外围接口电路组成，负责协调各个子模块的工作，收集各个子模块上传的信息和数据，并在内部对其进行整合和处理。主控模块需要实现稳定的不间断工作，对中央处理器和外围元器件的综合性能有着一定的要求。此处选用STM32L431RCT作为主控模块的中央处理器，以满足高计算性能、高稳定性以及低功耗的设计要求。以下是对这款芯片以及主控模块设计的详细介绍。

STM32系列微处理器是目前世界上最为流行的ARM架构微处理器，由意法半导体公司研发，专门面向高性能、低成本、低功耗的嵌入式应用场景。STM32L4x1系列属于基本型超低功耗STM32处理器，在满足设计要求的前提下在功耗、性能以及成本之间作了很好的平衡。下表为该芯片的基本属性。

**表1 STM32L431RCT低功耗微处理器性能参数表**

|  |  |
| --- | --- |
| 参数 | 值 |
| 型号 | STM32L431RCT |
| 封装形式 | LQFP |
| 引脚数 | 64 |
| 尺寸 | 12mil×12mil×1.6mil |

数据来源：STM32L431xx Datasheet - production data意法半导体

2.2.1初始化配置

STM32系列微处理器的初始化配置主要是针对芯片的硬件属性，包括芯片引脚配置、时钟配置、外设接口配置等。这些初始化参数以配置文件的形式保存，在编译时以编译参数的形式提供给编译器。开发者可以直接编写C代码配置文件，但需要对MCU硬件结构和原理有着很深入的理解。此外，开发者也可以通过一些集成化的初始化生成软件来实现STM32的初始化，如ST公司官方推出的STM32Cube MX代码生成工具。

STM32CubeMX是ST公司针对STM32系列微处理器开发的一款初始化代码生成工具，它集成了多个软件平台，包括STM32Cube HAL集成库以及TCP/IP,USB,RTOS等通信中间件，可以支持STM32全系列芯片的开发。STM32CubeMX拥有一个图形化的操作界面用于配置MCU的初始参数，并自动生成其所对应的初始化C代码及工程文件，开发者可以在生成的工程目录下调用其生成的库函数直接进行二次开发，而不用关心底层的实现。

需要配置的初始化项目根据功能可以分为几类，分别是针对MCU工作形态的全局配置，外设驱动的配置以及通信接口的配置。以下是针对本产品的MCU初始化配置参数说明。

2.2.1.1芯片引脚配置

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

|  |  |  |
| --- | --- | --- |
| 名称 | 标识 | 功能 |
| PA9 | USART1\_TX | 外部Modbus通信 |
| PA10 | USART1\_RX | 外部Modbus通信 |
| PC1 | LPUART1\_TX | NB-IoT模组通信 |
| PC0 | LPUART1\_RX | NB-IoT模组通信 |

**表1 主控芯片引脚功能分配表**

2.2.1.2芯片时钟配置

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.2.2外部接口

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.2.2.1 UART配置

UASRT1用于外部Modbus协议通信，由于Modbus报文长度未知，此处使用HAL库DMA+空闲中断实现串口不定长度数据的接收。此处使用STM32CubeMx软件根据如下初始化配置的参数生成初始化代码。

|  |  |  |  |
| --- | --- | --- | --- |
| 配置项 | USART1 | USART2 | LPUART1 |
| 工作模式 | 异步串行 | 异步串行 | 异步串行 |
| 波特率 | 115200 | 115200 | 115200 |
| 字长 | 8bit | 8bit | 8bit |
| 校验位 | 无校验 | 无校验 | 无校验 |
| 停止位 | 1 | 1 | 1 |
| 串口中断 | 开启 | 关闭 | 关闭 |
| DMA接收 | 开启 | 关闭 | 关闭 |
| 中断优先级 | 1 |  |  |

**表1 USART1配置项目**

2.2.2.2 SPI配置

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.3 通信模块设计

2.3.1 NB-IoT无线通信

2.2.1.1 NB-IoT技术简介

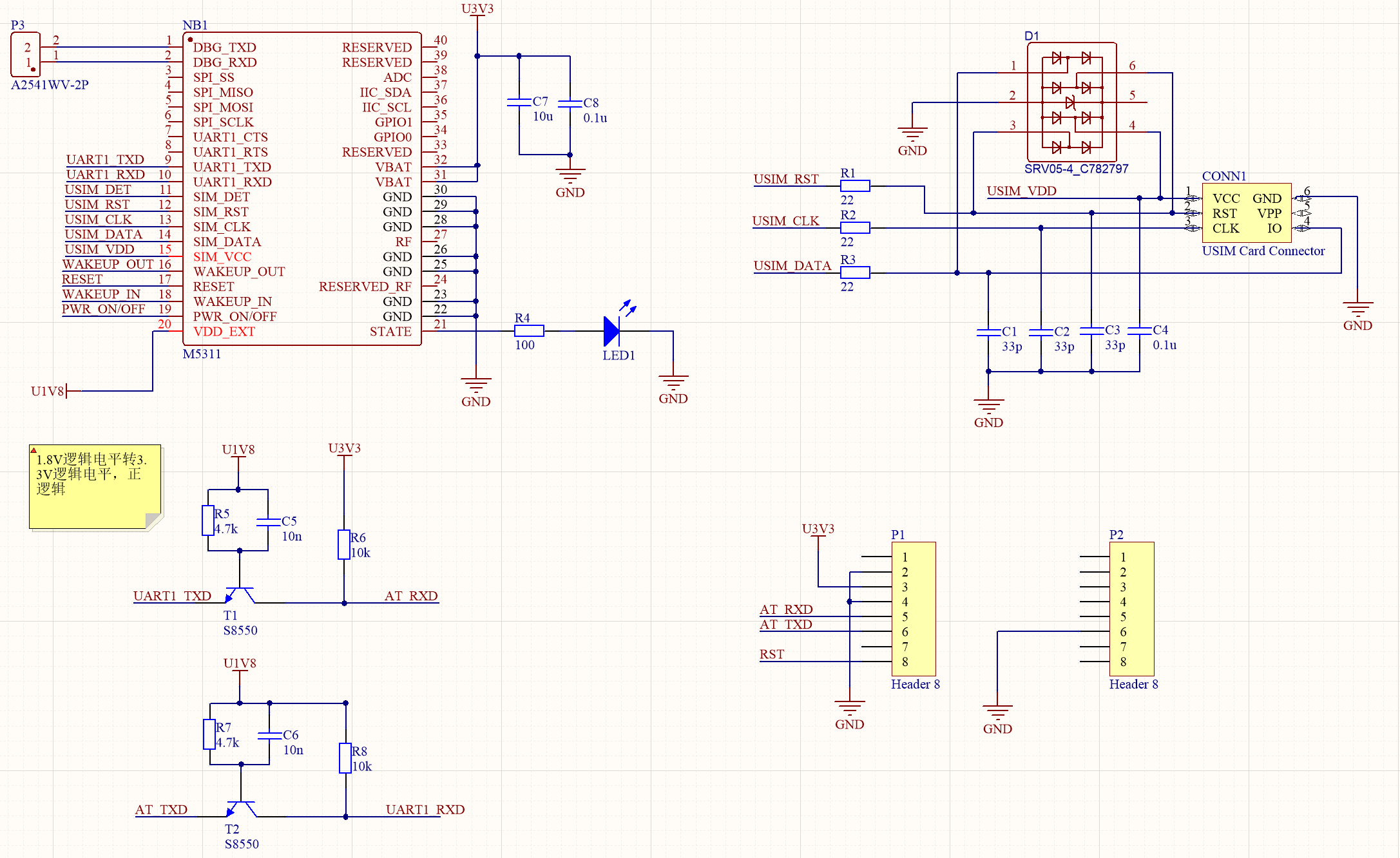
中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.3.1.2 中移动M5311模组

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.3.1.3 模组驱动电路设计

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。



**图1 XX图**（中文黑体，英文为Times New Roman五号加粗，居中）

* + 1. RS485有线通信

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.3.2.1 Modbus协议简介

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.3.2.2 RS485转换电路设计

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.4 传感器接入电路

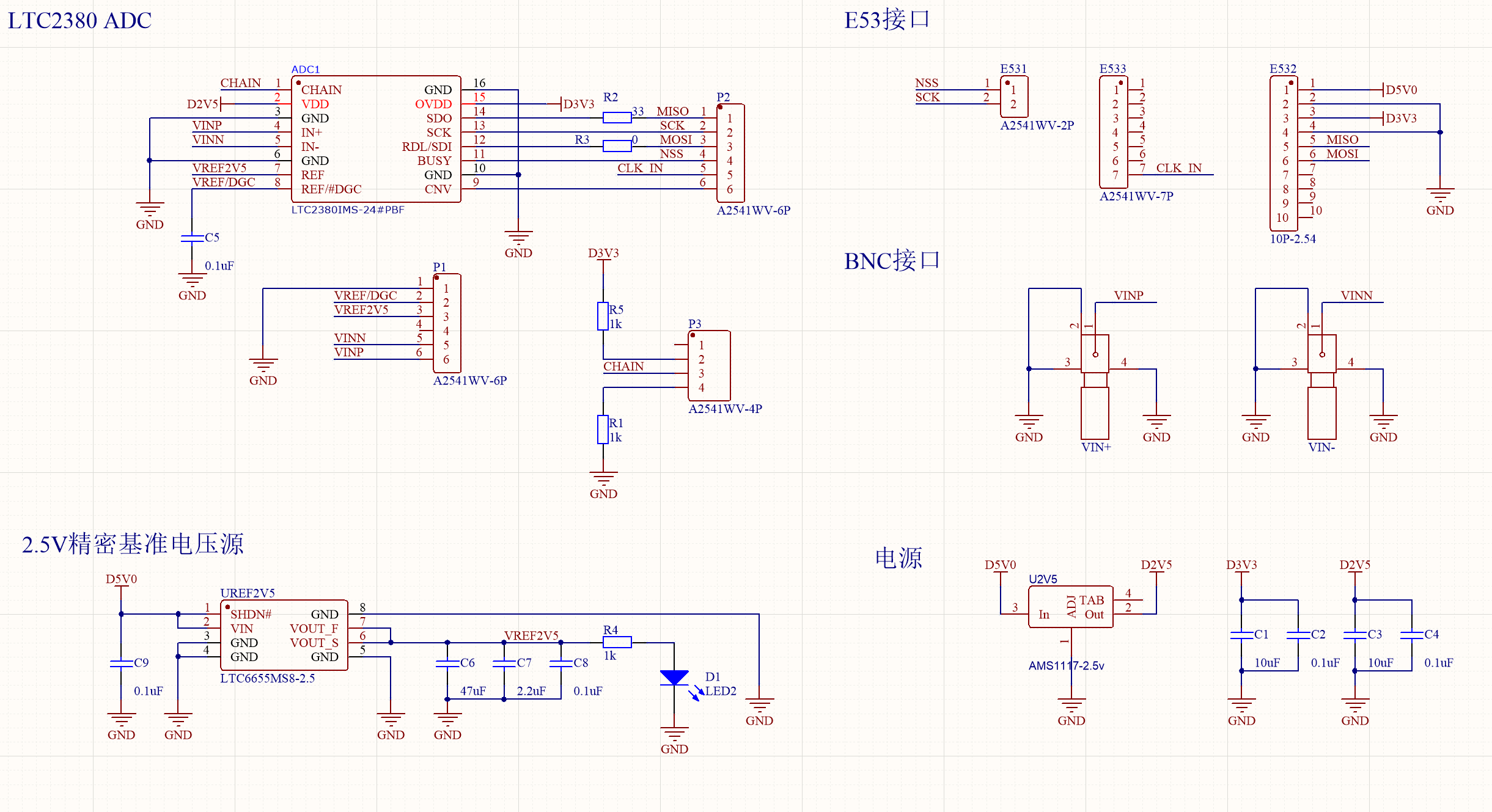
中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.4.1 多通道模拟信号采样电路

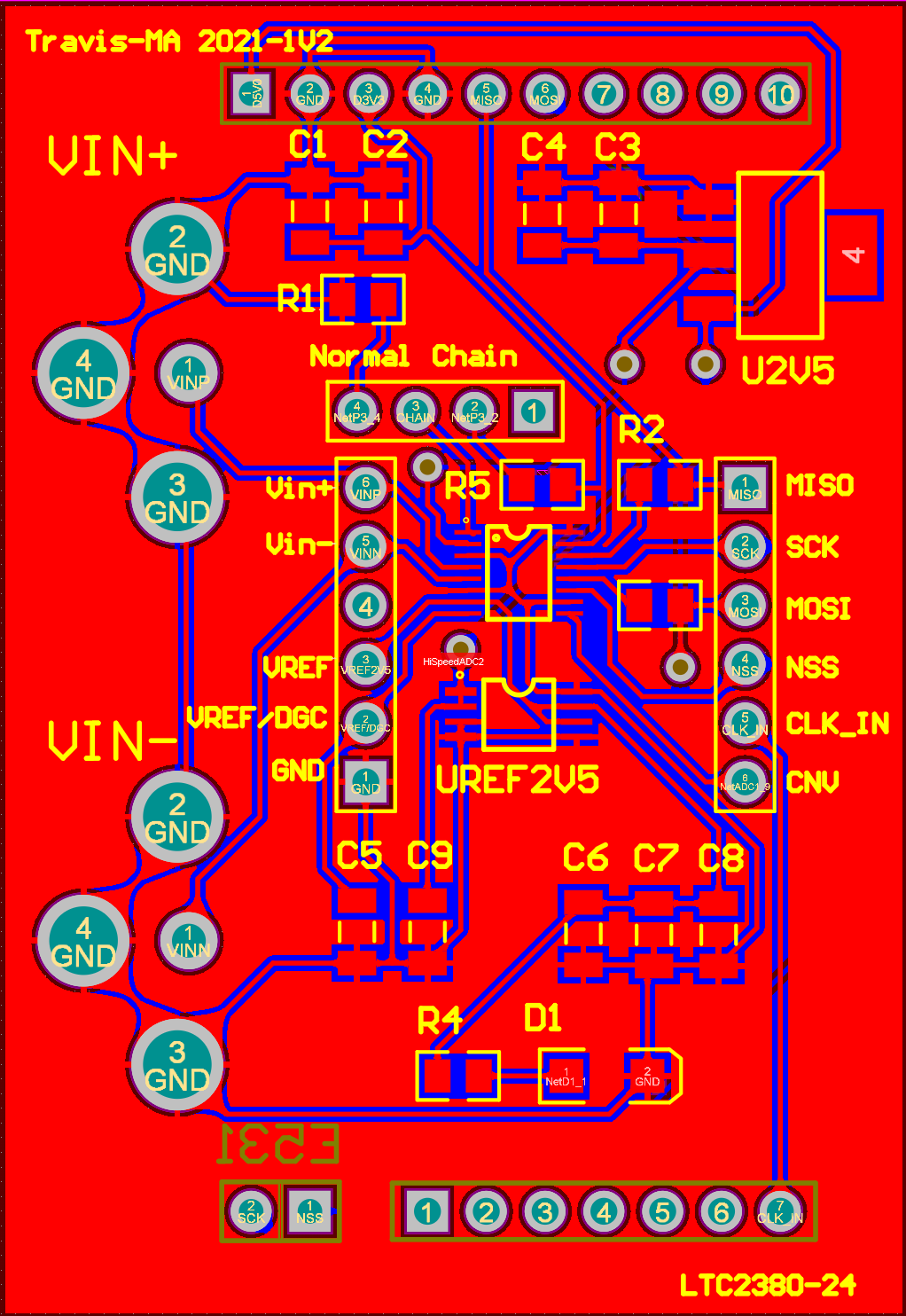
中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.4.2.高精度高速信号采样电路

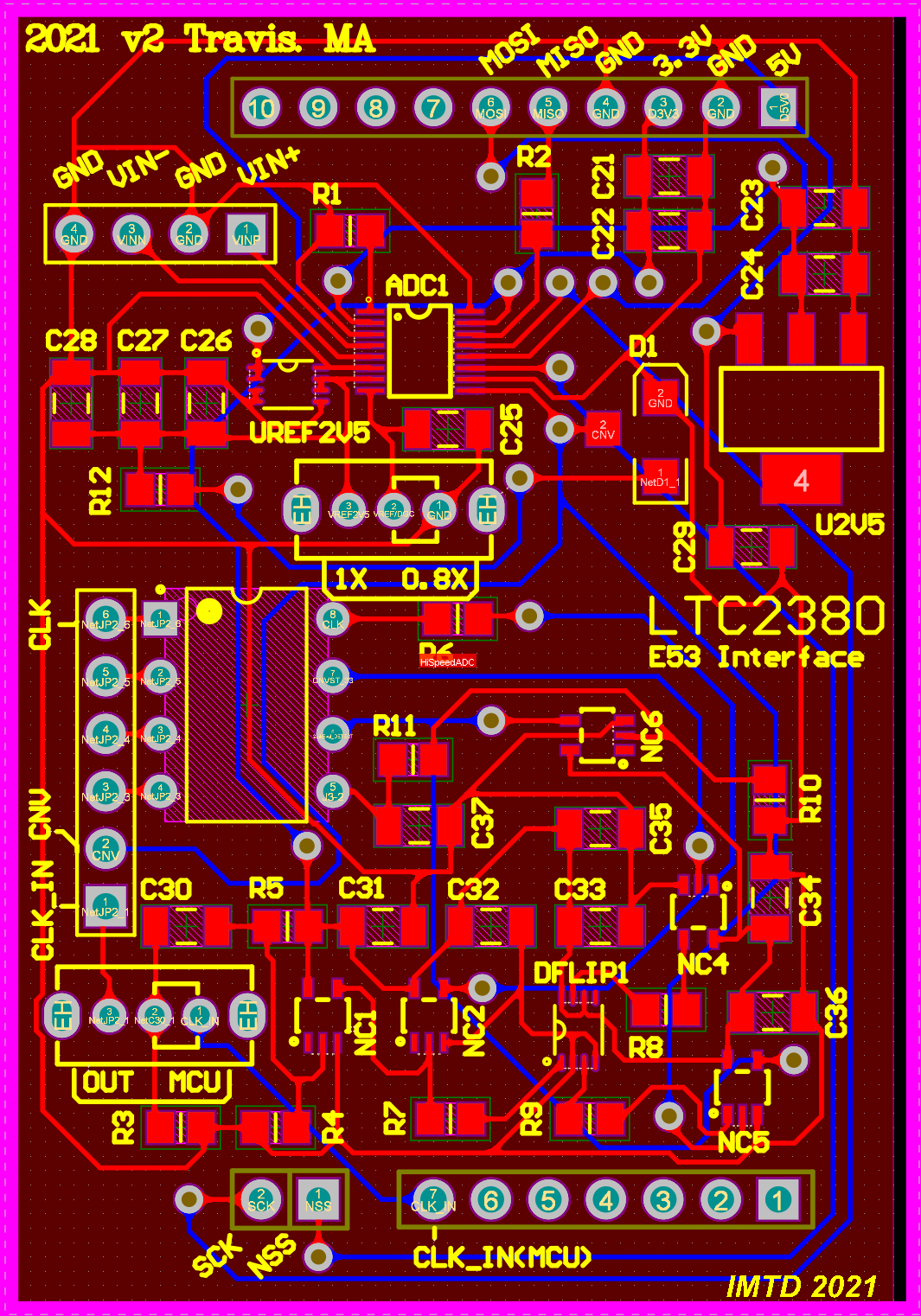
中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。



**图1 XX图**（中文黑体，英文为Times New Roman五号加粗，居中）



**图1 XX图**（中文黑体，英文为Times New Roman五号加粗，居中）

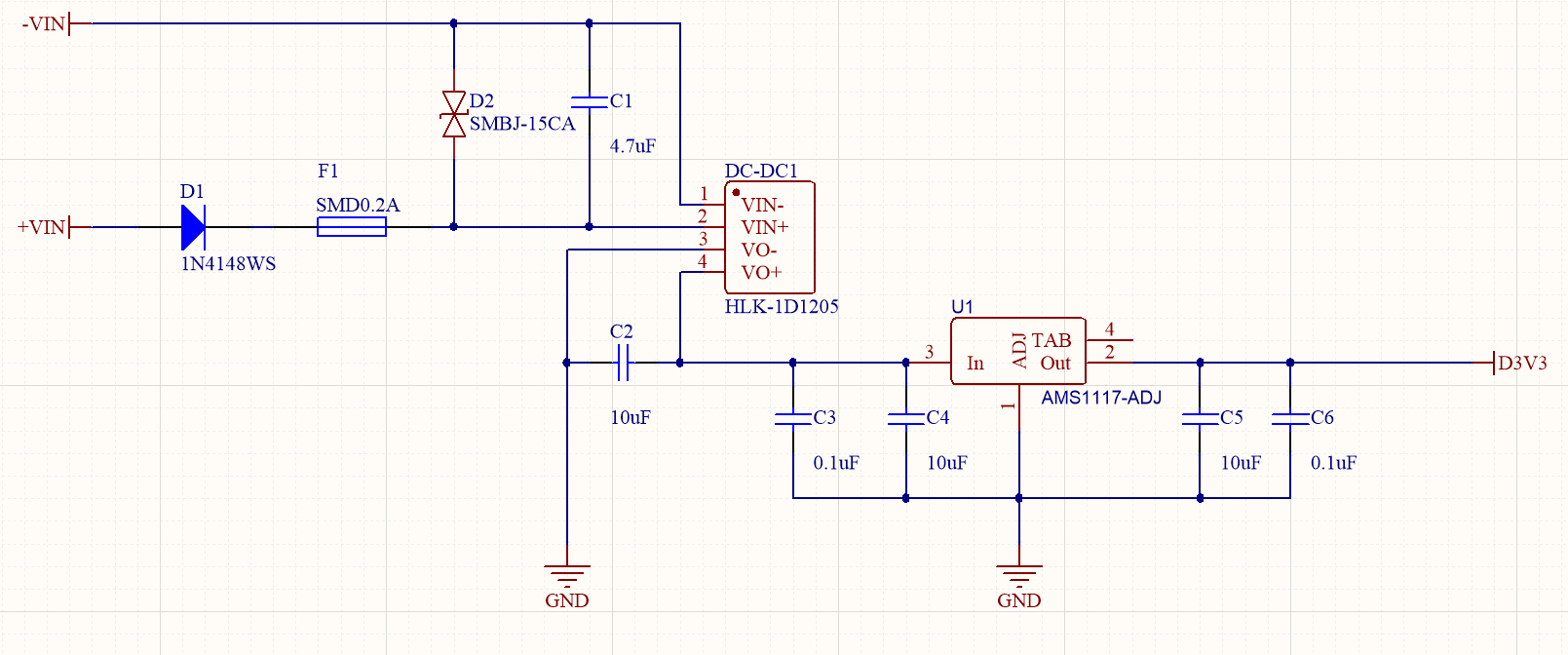


2.5 外围电路

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.5.1 电源电路

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。



**图1 XX图**（中文黑体，英文为Times New Roman五号加粗，居中）

2.5.2.隔离与防护电路

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

2.6 固件代码和PCB图纸

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

1. 联接管理系统

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.1 全局设计

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.1.1系统工作模式

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.1.2软件组成架构

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.1.3开发平台与框架

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.2 设备接入

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.2.1边缘节点接入

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.2.2云平台物联网接口接入

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.2.3云平台直接接入

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.3 数据分析和处理

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.3.1数据分析组件

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.3.2算法定制化开发

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.4 数据转移和储存

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.4.1边缘设备缓存

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.4.2数据共享平台

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.4.3云端储存

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.5 部署与功能拓展

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.5.1云端部署方案

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.5.2组件升级方案

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

3.6 源代码及配置参数

1. 客户端应用软件

客户端应用软件基于物联网核心应用场景设计，将后台储存的物联网数据可视化，并提供控制接口供用户和设备管理人员使用。本客户端应用软件根据典型的工业物联网应用场景设计了多种类型的设备状态监控界面，可显示实时数据和历史数据曲线，同时可在此基础上开发多种数据分析界面。软件基于WEB技术实现，在浏览器上运行，满足了PC、手机、PAD多种终端的使用需求。

4.1 全局设计

4.1.1系统工作模式

软件基于WEB技术开发，支持多平台运行，拥有独立的后端（服务器端）和前端（客户端）进程，后端进程运行于远程服务器，负责与数据库对接，并响应前端进程的请求。前端进程运行于客户浏览器端，负责在本地渲染网页效果，并将后端发回的数据布设在网站界面中。这种工作模式的好处是，后端程序运行于较高性能的远程服务器，拥有更强的计算能力和更高的网络链路速度，可以更快速的从远程数据库中读取数据并进行预处理。预处理过程包括数据包的重组，压缩等，尽量减少数据中的冗余部分。前端程序仅需编写界面的交互逻辑，而不用考虑大流量数据的传输和处理，仅需接收后端程序缓存完成的数据包即可，减轻了浏览器的工作负担，同时简化了程序代码，使其能更快的加载和运行。

4.1.2软件架构和运行逻辑

整个软件系统拥有两个主进程，分别运行于服务器和客户端浏览器，每个主进程中包含若干子线程，共同维护软件的运作。服务器进程需要拥有对接联接管理系统的上行接口，此处通过调用与联接管理系统共享信息的华为云OBS对象储存服务API来实现，这种方案的优势将在下文描述。服务器进程也需要对客户端进程发送的不同HTTP请求进行监听，并返回对应的数据包，所以服务器进程也需要对来自上行接口的数据进行重新整理和压缩，降低数据包大小以节省网络带宽。我们使用json格式的字符串作为服务器进程的通用数据传输格式，即向上行API获取的数据和向客户端发送的数据都以json格式编码。总体上来说，服务器进程可以分为三个子线程，线程1按照一定的周期向上行API调取原始数据，调取到的原始数据以文件形式保存在本地。线程2按照一定的周期对原始数据进行重新整理和压缩，这些处理过后的数据也以文件形式保存在本地，在客户端请求到来时作为HTTP的响应内容发送给客户端。线程3负责监听网络中的HTTP请求，并返回数据包至客户端。三个线程以异步形式运行，线程3开始运行的时间是一个随机值，取决于客户端何时返回请求。线程1和2的工作实际上是一个连续过程，但此处将其分为两个独立线程，原因是当线程1因为数据量过大或网络故障发生阻塞，线程2仍可以保证在一个运行周期过后返回最新数据的处理结果。

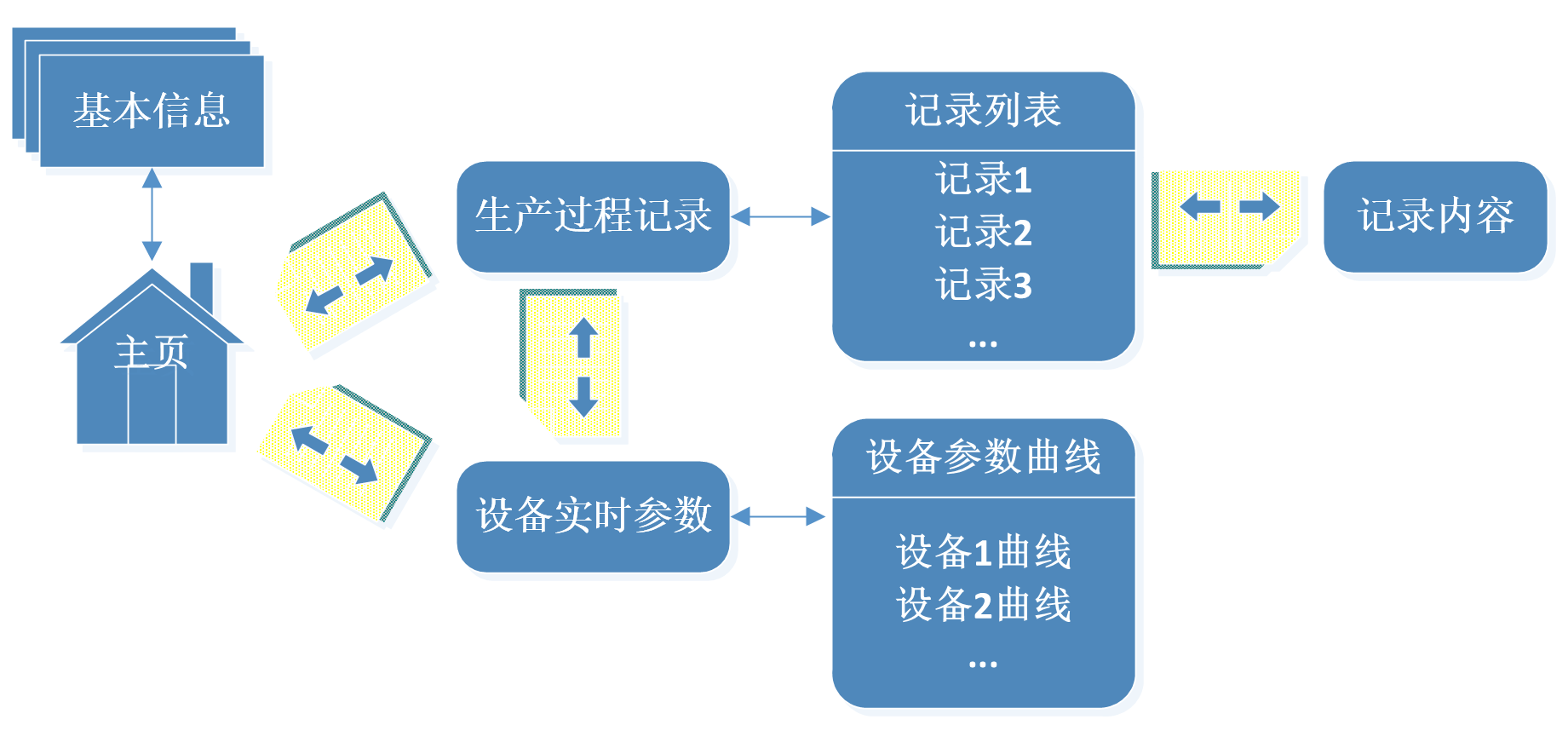
客户端进程的执行文件在网页加载初期被发送到用户浏览器上，同时传送的还有网页的静态文件，包括html文档，css文件，图片等。之后用户浏览器会运行客户端进程，解析网页静态文件，并根据用户的不同操作向服务器发送不同的HTTP请求。客户端进程中同样拥有异步运行的独立线程，如在网页加载初期，渲染网页静态文档和向服务器请求原始数据由两个分立的线程负责，这种架构保证了网站运行的流畅性。

4.1.3开发平台与框架

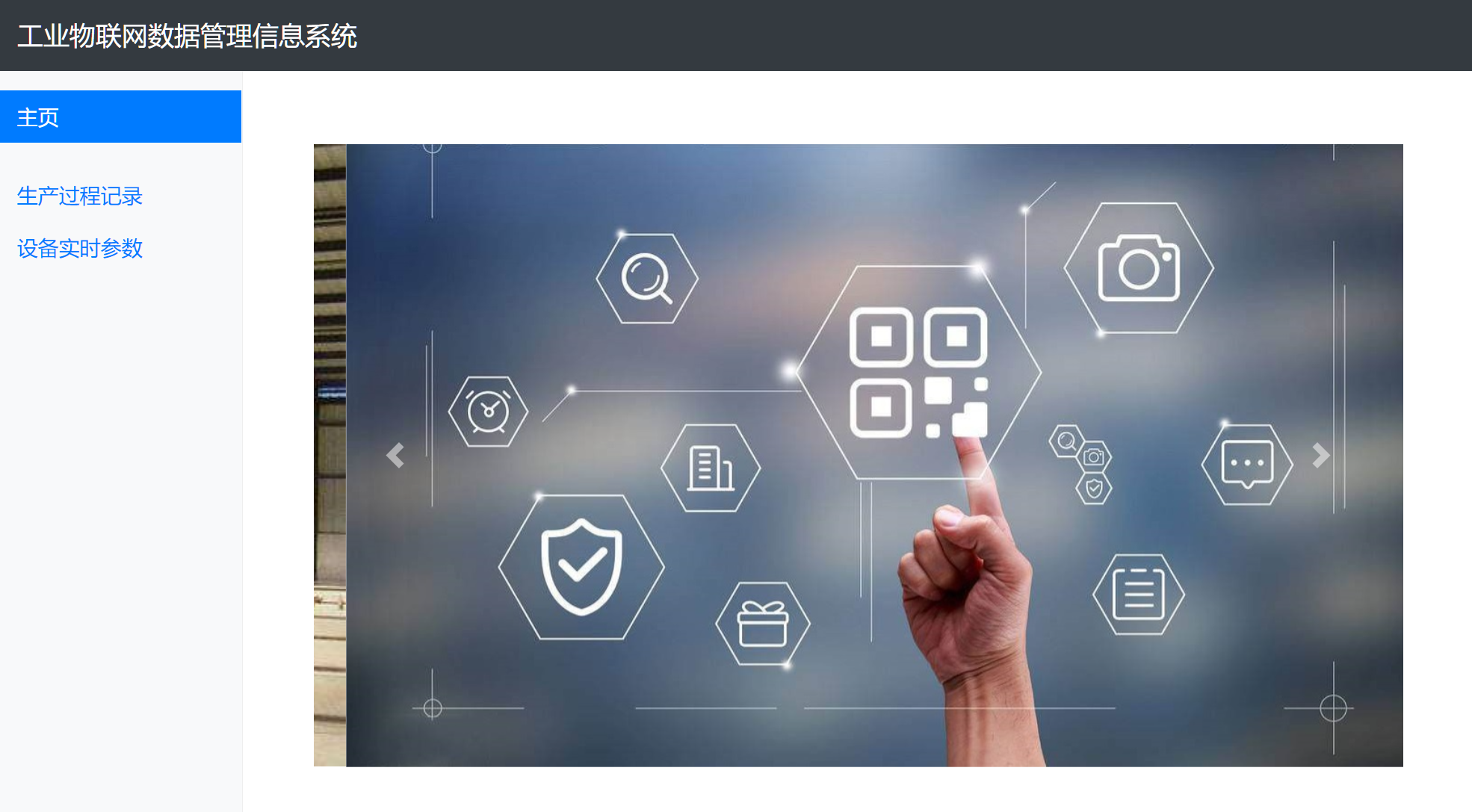
随着软件工程技术的发展，越来越多的软件通过不同的开发平台和框架实现，开发平台和框架是对一类形式的软件的抽象，它规定了一种文件结构形式或代码样式，使得在此基础上开发软件更加快速方便，同时拥有更好的性能。同样，物联网系统的客户端应用软件同样基于一些开发平台和框架实现：服务器端与客户端都基于node.js框架开发，node.js是一个开源与跨平台的 JavaScript 运行环境，可以不依赖浏览器运行JavaScript程序，这样一来，服务器端和客户端都可以用JavaScript一种语言来编写，通过标准的node.js接口通信，大大简化了系统的复杂程度。此外，客户端也使用了express和bootstrap框架用于网页交互的设计，实现了高性能的网页渲染和生动的交互效果。此外，客户端的数据可视化图表使用了由百度公司开源的 ECharts 技术，基于此技术，可以以低成本实现惊艳的数据可视化效果。

4.2 应用功能设计

工业设备物联网客户端拥有基本的设备参数监控功能与数据分析结果查看功能。在网页设计中，我们将这两个主要功能分为两个界面，分别是“设备实时参数”和“生产过程记录”。下图是网站的总体页面设计图。



其中“主页”页面可用于放置介绍企业基本信息的静态图片或文字（图xx），同时带有通向“设备实时参数”和“生产过程记录”页面的链接。“设备实时参数”页面（图xx）中放置有显示设备参数曲线的图表组件，可以自行编辑和缩放，并带有基本的统计功能。“生产过程记录”页面（图xx）根据相应规则将时序的生产数据分为多个记录集，以列表形式排列在页面上，通过点击子记录按钮，可以进入“记录内容”页面（图xx）查看该记录的详细信息，包括相应设备参数曲线，以及系统计算得出的各种数据分析结果。





4.2.1设备监控功能

WEB应用的后端进程运行在网络服务器上，通过监听特定端口传回的客户端HTTP报文，来处理客户端的请求，之后再通过HTTP协议回复客户端。运行在服务器端的后端进程可能需要在同一时间处理多个客户端发来的请求，这就意味着需要在同一时间处

4.2.1分析记录查看功能

WEB应用的后端进程运行在网络服务器上，通过监听特定端口传回的客户端HTTP报文，来处理客户端的请求，之后再通过HTTP协议回复客户端。运行在服务器端的后端进程可能需要在同一时间处理多个客户端发来的请求，这就意味着需要在同一时间处理

4.2 应用实现方案

4.2.1基于node.js技术的网站开发

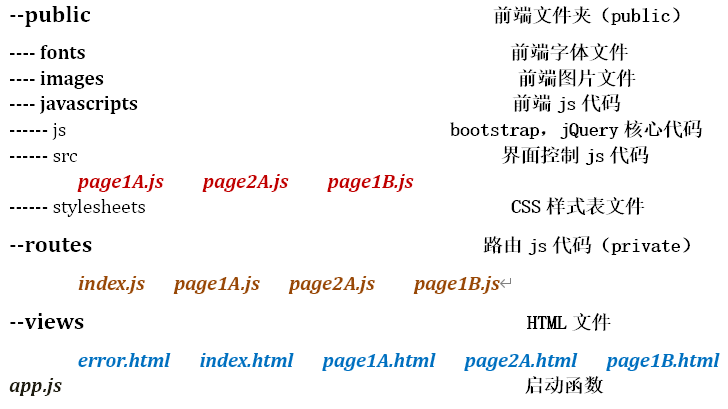
WEB应用的后端进程运行在网络服务器上，通过监听特定端口传回的客户端HTTP报文，来处理客户端的请求，之后再通过HTTP协议回复客户端。运行在服务器端的后端进程可能需要在同一时间处理多个客户端发来的请求，这就意味着需要在同一时间处理若干个并发连接。此外，后端进程应该在处理请求的空闲时间缓存来自上级数据库的数据，避免在客户端提出请求时再进行数据库读取，从而延长响应时间。以上问题对后端进程的使用性能都有着很大的影响，需要在设计时得到妥善解决。

传统的后端程序代码使用java或python等高级语言编写，开发者需要自行实现用于处理异步并发响应的多线程算法，无论是使用阻塞线程或引入管理线程来处理并发事件，实现起来都较为繁琐，受限于开发者有限的编程水平，这类算法很容易出错且难以达到很好的性能。

为了克服以上难题，我们选择Node.js技术开发后端应用。Node.js是一个开源与跨平台的JavaScript运行环境【】，拥有独立的Chrome V8引擎，使其可以不依赖浏览器独立运行JavaScript程序。由于JavaScript是一门专门为网络编程创建的编程语言，使用JavaScript开发服务器端程序有着天然的优势，很多专为网络编程设计的语法特性和函数可以直接使用，而无需像其他高级语言那样需要调用特定的插件和库。此外，相比传统开发方式，Node.js拥有更加强大的异步并发处理能力，同时支持各种最新的网络编程技术。Node.js异步运行的特点贯穿于整个编程过程中，如大量的使用回调函数，使得它再执行I/O操作（网络读取、访问数据库或文件系统）时，会在响应返回时恢复操作，而不是阻塞线程使得CPU循环等待【】。

4.2.2程序架构

本客户端软件按照标准的Node.js网页应用架构设计，代码文件包括运行在客户浏览器上的界面脚本，以及运行于服务器端的路由控制脚本，服务器程序由一个独立的启动脚本启动，下图显示了本软件的代码文件结构：



其中字段index代表“主页”，page1A代表“设备实时参数”页面，page2A代表“生产过程记录”页面，page1B代表“记录内容”页面。

4.2.2后端程序工作模式

后端程序运行于远程服务器，主要负责处理前端发回的各种响应请求，不同请求使用不同的URL来标记，后端主程序通过识别不同的请求URL会交给路由处理来转发给相应的路由控制器（图xx棕褐色代码文件）。在后端一共有四个路由控制器程序，分别处理主页、生产记录、实时参数、记录内容页面发回的不同URL，每一个控制器程序内包括数据仓库读取组件、数据格式编排组件等与对应页面功能有关的算法组件。下表列出了网站URL的路由规则。

**表1 网站路由表**

|  |  |  |
| --- | --- | --- |
| URL | 目标页面 | 请求描述 |
| /index | index | 返回主页 |
| /page1A | page1A | 进入设备实时参数页面 |
| /page2A | page2A | NB-IoT模组通信 |
| PC0 | LPUART1\_RX | NB-IoT模组通信 |

4.3.2前端程序工作模式

界面脚本代码文件与每一个界面html文档对应，负责控制界面中各个图表和组件，以及界面效果加载等，这部分代码在网页加载时被发送到客户机上，随后在客户端浏览器运行，这就要求严格控制代码文件大小并简化内部算法，以适应不同性能的浏览器。我们通过将耗时耗力的数据规整和表格渲染任务被安排在服务器端完成，减轻浏览器的工作压力，但这种方式可能会导致页面加载时间延长，因为浏览器必须等待服务器返回预处理好的结果才能进行画面下一步的渲染，为此，我们设计了一个分段数据传输算法来提升页面加载速度。在页面加载的不同阶段，客户端算法与服务器端算法将会进行多次双向通信，按照数据量大小分批次发送数据文件，客户端进程可在数据传输时同时进行界面渲染，这样大大提升了页面的整体加载速度。

4.3.1基于Express的前端开发

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

4.4 代码维护与功能拓展

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

4.5 源代码与配置参数

中文为宋体小四号字，段落首行缩进2字符，行距1.5倍。

表序在表题左方，不加标点，中间空一格，标题末尾不加标点。全文表格可统一编序，也可按章节编序，表序须连续。

表的示例如下：

**表1 XX表**（中文黑体，英文为Times New Roman五号加粗，居中）

|  |  |  |
| --- | --- | --- |
| XX（宋体五号） | XX | XXX  表格内容中文为宋体五号，英文为Times New Roman五号 |
| 1 | 11 | 111 |
| 2 | 22 | 222 |
| …… | …… | …… |

数据来源：......( 注于表下方，宋体五号，相对表格左下角缩进2个汉字。)

脚注（也可在论文篇末作尾注），字号小五，中文宋体英文Times New Roman。

图的示例如下：

图片

**图1 XX图**（中文黑体，英文为Times New Roman五号加粗，居中）

图序和图题居于图的下方正中，图序须连续。可用全文统一或按章节编序，但无论用哪种方式，应和表格、公式的方式统一。

结束语：（可选项，**中文黑体，英文Times New Roman，三号**）

......（正文内容格式：中文为宋体，英文为Times New Roman，均为小四号字，段落首行缩进2字符，行距1.5倍，下同。）

**参考文献**（黑体三号，另起一页）

[1] 作者．文献名[M]．出版地：出版者，出版年：起止页码（整体引用时不注）．(图书文献适用)

[2] 作者．文献名[J] ．刊名，年，卷（期）：起止页码．（期刊文献适用）

[3] 作者．文献名[N]．报纸名，出版日期（版次）．（报纸文献适用）

[4] 标准编号，标准名称[S]．（标准、法规文献适用）

[5] 作者．文献名[文献类型标识/载体类型标识]．出版地：出版者，出版年：起止页码（当整体引用时不注）．(载体类型标识为 “DK”、“MT”和“CD”，分别对应磁盘、磁带和光盘电子文献适用）

[6] 作者．文献名[文献类型标识/ OL]．（发表或更新日期）．[引用日期]．电子文献网址．(在线电子文献适用）

中文用宋体，英文用Times New Roman，均为五号字体。

附录（黑体三号字，为可选项，另起一页）

附录A

附A1

......（内容格式：中文为宋体，英文为Times New Roman，均为小四号字，段落首行缩进2字符，行距1.5倍。）

附录B

附录C

致谢（黑体三号字，为可选项，另起一页）

......（内容格式：中文为宋体，英文为Times New Roman，均为小四号字，段落首行缩进2字符，行距1.5倍。）