AIGC检测 · 全文报告单

NO:CNKIAIGC2025FJ_202505104733283　　检测时间：2025-05-25 20:11:50

篇名：基于单片机的多功能安全帽设计

作者：曾宇鹏

单位：

文件名：1281258_曾宇鹏_基于单片机的多功能安全帽设计.doc

全文检测结果

AI特征值：11.1%

AI特征字符数：3598

总字符数：32293

- AI特征显著（计入AI特征字符数）
- AI特征疑似（未计入AI特征字符数）
- 未标识部分

AIGC片段分布图

前部20%	中部60%	后部20%
AI特征值：16.9%	AI特征值：12.9%	AI特征值：0.0%
AI特征字符数：1092	AI特征字符数：2506	AI特征字符数：0

0　　32293

AI特征显著　AI特征疑似　未标识部分

分段检测结果

序号	AI特征值	AI特征字符数 / 章节(部分)字符数	章节(部分)名称
1	0.0%	0 / 1517	中英文摘要等
2	34.5%	666 / 1932	第1章引言
3	8.2%	690 / 8410	第2章系统需求分析与技术论证
4	17.4%	2017 / 11566	第3章系统设计与实现_第1部分

5	6.8%	225 / 3299	第3章系统设计与实现_第2部分
6	0.0%	0 / 1598	第4章系统测试与验证
7	0.0%	0 / 3971	第5章总结与展望

1. 中英文摘要等

AI特征值：0.0%　　AI特征字符数 / 章节(部分)字符数：0 / 1517

片段指标列表

序号	片段名称	字符数	AI特征	
1	片段1	312	疑似	20.6%

原文内容

20.6%(312)

摘要

本设计基于STM32单片机，实现了矿工安全帽的环境监测、GPS定位、照明管理和远程管理等功能的集成。系统采用DHT11温湿度传感器与MQ-2烟雾传感器，对矿井内环境参数进行实时的采集；硬件上集成LED灯组提供手动照明和远程控制照明；并且集成了讲接口以支持简单语音通话；通过Wi-Fi模块与MQTT协议实现数据上传与远程控制。现场模拟测试表明，定位精度可达约3m，环境异常时可立即触发本地报警与远程提示，通信可靠性约95 %，系统功耗低，满足矿井的基本应用需求。该方案重点在于功能覆盖全面与极高的稳定性，为矿井安全管理提供了一种简便可行的低成本解决方案。

关键词：多功能安全帽；单片机；智能监测；实时定位；远程管理

Abstract

This design is based on STM32 microcontroller, which realizes the integration of environmental monitoring, GPS positioning, lighting management and remote management of miners' safety helmets. The system uses DHT11 temperature and humidity sensor and MQ-2 smoke sensor to collect environmental parameters in the mine in real time. Integrated LED light clusters on the hardware provide manual lighting and remote control lighting; And the integrated speaking interface to support simple voice calls; Data upload and remote control are realized through the Wi-Fi module and MQTT protocol. The field simulation test shows that the

positioning accuracy can reach about 3m, the local alarm and remote prompt can be triggered immediately when the environment is abnormal, the communication reliability is about 95%, and the system power consumption is low, which meets the basic application needs of the mine. The focus of this solution is on comprehensive functional coverage and high stability, providing a simple and feasible low-cost solution for mine safety management.

Keywords: Multifunctional Safety Helmet; Microcontroller; Environmental Monitoring; Basic Positioning; Remote Management

2. 第1章引言

AI特征值：34.5%　　AI特征字符数 / 章节(部分)字符数：666 / 1932

片段指标列表

序号	片段名称	字符数	AI特征		
2	片段1	204	显著		10.6%
3	片段2	462	显著		23.9%

原文内容

第1章引言

1.1 研究背景和意义

随着矿业的不断发展，矿井作业环境愈发复杂，矿井作业在安全方面存在的问题也逐渐地被凸显出来，矿工们在安全保障方面所面临的问题慢慢地成为了整个社会所以关注的焦点[1]。煤矿井下的危险因素多种多样，这当中包括瓦斯爆炸、煤尘爆炸、坍塌等，这些因素直接威胁到矿工的生命安全。尽管传统的矿工安全帽具有保护矿工在危险工作环境下头部免受物理外力直接打击、以及在矿下照明的功能，但在实际运用当中，还存在一些问题，如功能比较单一、各种监控综合性不强、系统实施成本高、信息整合能力弱、井下人员与设备的跟踪定位能力差、煤矿生产各要素的统合性考虑不够[2]。

因此，智能矿工安全帽的设计遂成为提升矿井安全性的一个重要的研究方向。本研究旨在设计一种集成多功能的智能矿工安全帽，通过集成定位、语音通信、环境监测、自动报警，远程管理等智能化技术，以此来提高矿井安全生产管理水平，并确最大程度保矿工的生命安全。该系统能够做到实时监控矿工的工作环境，同时还能在发生突发事件的时候马上报警，并且为矿井管理人员提供实际的后台管理

10.6%(204)

手段,以期为矿工的人身安全和矿业的发展提供有效助力。

1.2 国内外研究现状

1.2.1 国外研究现状

国外进行的研究工作，主要是聚焦于智能化技术和矿工安全帽进行深度整合的这个方面，以提升作业效率以及救援响应能力这两个方面的表现。例如加拿大NSS公司与Cambrian公司两者合作去开发出来的一款AR安全帽产品，它搭载了微软HoloLens 2，能够去实时地显示出来钻孔定位方面的数据信息，从而能够将爆破作业中所产生的误差给减少大约15%的幅度，并通过AR标注功能实现地面专家进行特定的远程指导维修，作业效率提升将近80%[3]。国外在平台标准化有相当深入的研究，如霍尼韦尔智能安全帽联动无人机与救援机器人，塌方事故中自动发送生命体征与坐标数据，救援响应时间缩短了30%。南非的Expert Mining Solutions公司和四所大学他们合作去研发出来的一套名为“Life”的穿戴式系统，通过借助脑电波传感器这类的设备，来对矿工的疲劳状态情况进行监测的动作，其监测的准确率方面的表现能够达到95%的水平，与此同时，并且结合IMU传感器，来识别出一些具有高风险性的身体姿势，并推送一些用以矫正姿势的建议信息[4]。在标准化这个方面的工作上，像是EN397以及ANSI Z89.1这些标准规范，就对智能安全帽产品所涉及到的通信协议以及数据加密标准这些内容去进行了明确的规定动作，从而能够去确保在跨国作业的矿山之间实现数据信息方面的互联互通。

1.2.2 国内研究现状

国内矿工安全帽的智能化研究以5G通信技术与多模态传感器融合为核心。是拿5G通信技术与多模态传感器加以融合的技术手段来作为其核心内容的，它是通过借助于去集成像是北斗/GPS这类能够达到厘米级别的定位模块，高精度的气体传感器，与此同时，还包括了生命体征监测方面的模块如心率、血氧监测，从而能够去实现对井下作业人员进行动态追踪的动作，并且对环境当中潜藏的风险进行实时预警方面的功能。在数据驱动决策方面，徐工集团开发的数字孪生系统可预判设备故障，通过分析安全帽采集的振动、温度等数据，优化维护周期，使采矿设备故障率降低30%[4]。此外，模块化设计推动技术普惠，例如安徽庐江某铁矿通过可拆卸卡扣将传统安全帽升级为智能终端，支持远程操控无人铲运机，井下作业效率提升50%并联动通风系统自动排险，事故响应时间缩短至30秒内此同时[5]。

然而，国内的研究多集中在单一功能或某些特定模块的设计上，缺乏整体系统设计的研究。例如，许多研究侧重于传感器的选择与数据采集，但缺少对系统集成和数据处理的综合考虑。近年来，有些企业设计了数字化安全监管平台，提出通过APP进行远程监控与管理，进一步推动了智能安全帽的应用。然而，如何在复杂的矿井环境中实现高效、稳定的多功能集成，仍是国内研究面临的主要挑战。

1.3 论文主要内容概述

本论文共分为五章，具体安排如下：

第1章引言：阐述研究背景、意义及国内外研究现状，明确研究目标和论文结构。第2章系统需求分析与技术论证：分析系统功能需求，评估技术可行性，为后续设计提供依据。

第3章系统设计与实现：详细介绍系统总体架构、硬件电路设计、软件架构设计及各功能模块的实现方案。

第4章系统测试与验证：描述系统测试过程，通过功能测试和性能测试验证系统效果。

第5章结论与展望，总结了论文的研究成果，并对未来的研究方向进行展望。

通过以上安排，本文旨在系统展示智能矿工安全帽的设计与实现过程，及其在提升煤矿安全中的应用价值。

3. 第2章系统需求分析与技术论证

AI特征值：8.2%　　AI特征字符数 / 章节(部分)字符数：690 / 8410

片段指标列表

序号	片段名称	字符数	AI特征		
4	片段1	291	疑似		3.5%
5	片段2	426	显著		5.1%
6	片段3	264	显著		3.1%
7	片段4	205	疑似		2.4%

原文内容

第2章系统需求分析与技术论证

2.1 系统需求分析

矿井中作业环境非常复杂且危险，早年的安全设备已无法满足如今矿井对于安全管理的需求。因此，能够适应当下新矿井作业需求的多功能安全帽就显得无比重要。该系统希望通过集成多种技术，来提升矿工的安全性和工作效率。

在矿井中实现精准定位的是设计当中的核心需求之一。矿工的实时位置必须可以准确显示出来，定位误差最好控制在3米以内，这样矿工的安全才能够得到保障。通过集成u-blox的NEO-6M GPS模块，系统可以确保矿工的地理位置通过现场OLED显示屏和远程APP进行实时精确查看，管理人员随时掌握矿工的动态以及矿工确保自己得安全。

3.5%(291)

环境监测也是一个重要得需求，矿井内的温湿度和有害气体的变化，可能会带

来安全隐患。为了及时发现危险并去采取应对措施，安全帽必须配备温度、湿度和烟雾传感器，可以实时监控矿井的环境状况。一旦环境参数超出了安全范围，系统就可以自动触发报警，蜂鸣器会报警响起，警告矿下作业人员及时撤离，同时APP会及时推送警报信息给矿井管理人员，确保后方应急力量可以快速响应。

语音通信系统是另一个重要的功能。矿井内噪音往往很大，并且信号一般也不好，普通的通信设备可能很难保障通话的清晰和稳定。因此，安全帽需要集成对讲模组，以此来确保矿工与指挥中心之间的语音通信质量，特别是在紧急情况之下，可以实现清晰、实时的交流，及时求助后台，保障自身的安全。

照明功能在矿井当中同样不可忽视。矿井的光照条件通常比较差，矿工在低光环境下需要额外的安全保障。安全帽需要集成高亮度LED灯，支持手动和APP远程控制的开关功能，保障矿工在任何情况下都能有足够的照明，在遇到危险时可以由远程的人员来开启以查看具体的矿下情况，做出分析，以此来确保其安全操作。

远程管理平台则用于矿井后台管理人员实时监控矿工的工作状态以及环境数据。该平台可以通过云服务器与安全帽进行数据交互，支持管理人员远程查看数据，进行设备配置，及时调整设备参数或者去处理紧急情况。系统的可靠性以及通信稳定性也必须经过优化，确保在矿井恶劣环境当中长时间运行，系统总体需求分析见表2.1。

表2.1 系统需求分析

功能需求描述

环境监测使用DHT11温湿度传感器和MQ-2烟雾传感器，能够实时获取温度、湿度和烟雾值。

自动报警和远程预警功能工作人员遇到紧急情况能够自动报警，管理员能通过APP远程推送预警告知。

安全照明功能集成高亮度LED灯，支持手动和APP远程控制，保证矿工在低光环境下的安全。

语音通信功能集成对讲模组，实现语音通信，保证紧急情况下能够联系到后天人员。

远程数据通信设备端的环境监测数据可以上传至云平台。

后台管理平台远程管理平台可以实时接收云端推送的报警，以及远程配置同步到云端和设备。

2.2 技术可行性评估

为了确保多功能智能安全帽系统能够满足矿井复杂环境下的安全作用与智能需求，接下来将定位技术、环境监测技术、语音通信技术和数据通信技术等多个方面来进行技术可行性评估。本次所要进行的可行性评估工作，是基于先前已完成的系统需求分析这个基础之上的，采用专业的架构和低成本、低功耗的方案，使得系统的专业性与可实现性这两方面得到保证。

2.2.1 定位技术可行性

定位系统在矿工安全帽当中承担着着极其重要的角色，可以确保在矿井环境中实时追踪并获取矿工的地理位置信息。井上设备常见的定位方式为全球导航卫星系统，井下设备的定位以无线定位技术为主，当前常采用的定位技术包括超带宽、射频识别、无线局域网等[6]。对于矿井作业这个特殊的场景，我们需要综合考虑不同的定位技术的可行性。

1）GNSS技术

GNSS，也就是我们常说的全球导航卫星系统，是一种基于卫星的导航技术，其原理涉及卫星的分布、信号传输、接收与计算以及多系统融合。GNSS系统由一组轨道卫星和地面控制站组成，卫星分布在地球轨道上，以确保全球覆盖，也就是说，它能够通过接收多颗卫星所发出来的信号，进而计算出接收器自身所在的一个精确位置[7]。通常来说，GNSS想要实现三维定位的话，往往需要接收到至少四颗卫星的信号才可以。不过，它在定位精度上常常会受到卫星信号的质量、大气环境状况以及接收环境等诸多因素所呈现出来的一些影响。

然而，要是把GNSS技术运用到矿井环境当中，那它就会面临一些非常显著的技术挑战了。由于卫星信号强度的限制，无法直接应用到井下。如此一来便使得GNSS在地下矿井里面基本上是处于一种失效的状态，仅仅只能够在地面以及井口这些区域拿来提供一个相对来说比较可靠的定位服务[8]。这种局限性就让GNSS没有办法完全满足矿井作业那种需要全覆盖的定位需求了。尽管GNSS在地下环境当中的应用效果是比较有限的，但它在矿井入口以及地面区域的定位需求当中，却具备了比较高的适用性，能够为矿井作业提供一个可靠的辅助定位支持。

目前来看，基于STM32微控制器所搭建出来的GNSS模块集成方案，在技术上是显得比较成熟的，并且在成本控制方面也显得比较合理，与此同时，它还能够充分地去利用多系统卫星信号，进而提升定位的稳定性。

2）超带宽无线技术

UWB技术运用极短脉冲信号来进行通信以及测距，通过测量信号的到达时间差或者到达时间来实现高精度定位。UWB超带宽技术是人员定位系统中公认的精确度最高的无线定位技术， 也是应用最广的无线定位技术，定位精度最高达0.1 m，最低精度也不超过0.3 m，满同时还具备良好的穿透能力以及抗干扰性能[9]。超宽带技术作为新兴的室内高精度定位解决方案，它具备纳秒级时间分辨能力和厘米级定位精度，非常适合在复杂环境当中进行精确的人员定位，可以满足工业应用的定位需求。UWB技术的强抗多径干扰能力，让它在矿井这种多反射环境中表现优异，能够提供稳定且可靠的定位服务。然而，UWB系统的部署需要在矿井内建设完整的基站网络，这当中会涉及大量的基础设施投入以及复杂的网络规划。要是考虑到矿井环境的特殊性以及安全要求，基站的安装、维护以及升级都面临着较大的技术和成本挑战，这个在一定程度上就限制了UWB技术在大规模矿井应用当中的可行性。

5.1%(426)

3）RFID技术

RFID 定位系统是一种利用射频技术进行物体定位和跟踪的非接触类系统，基本组件包括天线和标签。凭借标签体积小、成本低等优点，RFID定位技术被广泛应用于室内环境[10]。这个技术分为低频、高频以及超高频三种工作频段，不同的频段在传输距离以及穿透能力方面都有其各自的特性。RFID系统主要依靠接收信号强度指示来估算距离，因此定位精度相对来说有限。RFID技术在矿井环境中主要适宜于区域性的人员识别以及粗略定位，它的技术原理决定了定位精度会相对有限。虽然RFID系统的部署成本较低，技术也相对成熟，但是它的定位精度很难满足矿井安全对精确位置信息的严格要求。RFID技术容易受到金属环境的信号屏蔽影响，在矿井这种金属设备密集的环境中，信号传输的稳定性存在不确定性。除此之外，RFID的有效识别距离有限，需要密集地去布置读写器才能得以实现较好的覆盖效果，这同时也就增加了系统的复杂性以及维护成本。

综合分析各种定位技术的特点以及限制，选用GPS定位方案，在当前技术条件下以及需求上是拥有较高可行性的，既能够去控制系统成本，又能够在一定程度上满足矿井安全的基本定位需求。

2.2.2 环境监测技术可行性

矿井环境中，对于温湿度以及有害气体等多项监测需求，接下来我们逐一分析每种环境监测需要用到的技术以及可行性

1）温度和湿度监测

在矿井中，温湿度作为衡量矿井安全的重要指标，对工作人员的身体健康以及人身安全起着重要的影响。目前来看温湿度的监测主要使用的是固定设备，并不能随处设置进行全域监测，而矿工可能会出现偶现情况，所以可以将温湿度传感器集成到安全帽上，对温湿度传感器的选取中，考虑到传统的数据采集需要A/D转换电路，从而给电路设计带来很大麻烦[11]。对此本文选择DHT11温湿度传感器因为它拥有低功耗特性，并且在测量精度方面非常准确，最重要的是对于本文所使用到的架构来看，使用DHT11只需要很少的外设资源。所以说，运用它来开展矿井温湿度的监测工作，是比较适宜的而且，其成本也比较低，。因此，借助它低成本以及高性价比的特性，DHT11可以说非常适合矿井环境中那些常规的温湿度监测需求。

2）有害气体监测

煤矿井下作业环境复杂，存在着H2 S、CO、NO、C12 等多种有害气体，这些有害气体一旦浓度超标，将严重威胁工人健康，并可能引发爆炸或窒息等重大安全事故[12]。当前，煤矿井下有害气体监控主要有手持检测设备和有线监测系统，显然这两种方式都极为不便。因此将监测的设备集成到安全帽里是最便捷，最高效的方法。在煤矿井下有害气体监控系统中，传感器的选择至关重要，直接影响数据采集准确性和系统可靠性。本研究采用MQ-2气体传感器能够去检测像甲烷、一氧化碳等多种可燃气体，并且响应时间很快，因此适合矿井中有害气体的检测。借助STM32的

ADC接口，我们可以读取其模拟信号，在经过标定之后，就可以将其转换为PPM值。该传感器性能稳定，并且被广泛地运用于工业气体监测的方面，所以非常适合矿井环境中的气体安全监测工作。

综合来看，运用DHT11与MQ-2传感器来进行组合，这个方案能够全面地监测矿井当中的温湿度以及有害气体。它拥有低功耗、调试简单，以及技术成熟等特性，所以说，这项环境监测技术的可行性是非常高的。

2.2.3 语音通信技术可行性

在矿井这种复杂的地下环境当中，要去选择一种合适的语音通信技术来加以应用，是非常重要且值得思考的，并且是需要去综合性地考虑到像是信号覆盖的情况、网络依赖的程度 、安全合规方面的要求以及部署成本的高低等等这些方面的因素。下面将会针对三种主流的技术方案，来分别进行简要介绍以及可行性评估的动作。

1) 射频模拟对讲技术

射频数字对讲是当前社会中一种十分重要的无线通信工具，在各类手机无法使用的应急通信当中，它发挥着不容忽视的巨大作用。其采用FM调制，将麦克风采集的模拟的语音信号直接调制到射频载波上面，通过专用频段点对点或组播传输，完全不依赖基站或IP网络，仅需基础硬件即可实现语音通话。再通过专用频段点对点或者以组播方式进行无线传输。此方案完全不依赖运营商方面的基站设施或者是IP网络资源，只需要一些基础硬件，就能够去完成语音通话的功能实现。与此同时，它还具备有像动态的信道选择这样的机制，如此一来便能够使其抗干扰能力这方面的特性得到提升。

在矿井这样的特殊作业环境当中，这种技术方案是能够在数百米乃至数公里范围的巷道内部，来保持一种稳定可靠的通信状态的，从而满足到矿工在像是日常巡检作业、调度对讲沟通以及紧急情况下的呼救等等这些方面的实际需求。与此同时，矿井这样的环境下，安全帽里面的组件小型化是当中最为主要的一个特点，尤其是要集成在安全帽这样随声佩戴的装备当中，其内部各种元器件的密度将会很高，相互之间也容易发生干扰。如果处理不当，导致其在工作中产生了严重的电磁干扰，就会影响到整个电路系统[13]。为了方便佩戴和使用以及产品的生产，数字对讲相较于其它的方案整体上是更加适合矿井场景的。

2) IP 网络语音通信技术

IP 网络语音传输，即VoIP的原理是通过语音的压缩算法对语音数据编码进行压缩处理，然后把这些语音数据按 TCP/IP 标准进行打包，经过 IP 网络把数据包送至接收地，再把这些语音数据包串起来，经过解压处理后，恢复成原来的语音信号，从而达到由互联网传送语音的目的[14]。然而，就矿井深部区域的信号覆盖而言，其所呈现出来的难度可以说是非常之大的，这就需要在每一个工作面以及那些辅助巷道之类的地方去部署上多个AP、微基站，又或者是光纤中继设备，与此同时

，还需要把网络冗余以及切片隔离这方面的工作给做好，这样做的目的，就是为了防止网络节点那里出现单点故障这样的情况。像是网络抖动、时延以及丢包这些现象之类的，它们都会直接影响到语音通话所呈现出来的质量效果；通话质量受到网络好坏的影响。并且在停电时候无法使用，这在真正遇到问题时候不能通话有很大的风险的。同时，VoIP 模块功耗相对较高，设备在通话时电流峰值可达数百毫安，对安全帽内的电池容量和散热设计提出更严格的要求，综合来看这个方案并不适合。

3）卫星语音通信技术

卫星通信具有覆盖范围广、实时性强、抗干扰的能力强等特点，可在无地面网络覆盖区域或地面常规通信基础设施遭到破坏的情况下提供独立的应急通信服务[15]。其通过 LEO或 GEO卫星链路，提供几乎全球无缝语音覆盖，不需要依赖任何地面铺设的网络基础设施，就能够去轻松完成远距离的通话任务，这种特性使得它尤其适用于像是在应急救援行动当中，或者是在人员与外界失去联络这种极端状况之下所进行的跨区域通信联络等场景。但矿井井下环境因遮蔽天空，无法很好的与卫星保持视距链路，必须在井口或地面搭建高增益天线与中继站，然后通过光纤或者无线微波回传至井下的节点，再由井下终端回传至安全帽，系统集成复杂度和建设投入大幅提升。并且，由于卫星链路传输时延长，数据传输的实时性受到一定影响，同时，卫星通信环境的复杂性也可能导致数据传输的稳定性下降[16]，这对矿工的生命安全有着极大的影响。卫星通信终端本身存在的尺寸规格方面通常是相对比较大的，其运行时的功耗表现以及对电池续航能力的需求相应也会更高一些；与此同时，卫星通话的资费标准也是比较贵的，不适合矿井大规模、日常对讲进行使用，一般作为应急备份方案，将其集成到小的安全帽里面更是不太合适，但是其可以在矿井外部设置一个卫星终端应急备用。

经过对覆盖范围、通信稳定性、系统成本、部署复杂度和安全合规性等多维度比较与权衡，最终选择射频数字对讲技术。该方案在保证高抗干扰性、低延迟及安全认证的前提下，可利用现成的对讲模块快速集成到安全帽中，满足矿井内日常通信和应急呼叫需求，同时具备较低的设备采购和维护成本。

2.2.4 数据通信技术可行性

在矿井多功能安全帽的通信技术选择中，4G无线通信技术、Wi-Fi和LoRa各有其特点，为了准确的确定方案，需要对这几种技术进行仔细的可行性分析。

1）4G无线通信技术

第四代移动通信技术，简称4G。它是集3G与WLAN于一体，并能够快速传输数据、高质量音频、视频和图像等。4G能够以100Mbps以上的速度下载，此外，4G可以部署在DSL和有线电视调制解调器没有覆盖的地方[17]。4G技术具备较强的抗干扰能力和稳定性以及较高的数据传输率与容量，适用于煤矿这种特殊环境，能够满足煤矿对大量数据传输的需求。同时，具有较强的系统兼容性和扩展性，便于与其他系统

3.1%(264)

进行集成和互联。从实际情况来看，4G技术在煤矿无线通信系统中的应用已经比较成熟，目前正朝着技术融合方向转型[18]。

但是，对于矿井中矿工安全帽这个设备情况就不同了，4G模块自身在功耗方面的特性是比较高的，这就需要去进行一种复杂的电源设计工作，才能来支持它长时间去运行下去 。如此一来便会去增加安全帽本身的重量以及成本会导致线路布局严重，进而会影响到佩戴时候的舒适感觉。除此以外，4G模块本身以及相关的流量费用，其成本也是比较贵，并且在深入矿井的一些区域，信号的穿透能力也会受到一定的限制，这就需要去进行额外的基站或者是中继设备的部署工作，从而会进一步地去提升建设和维护方面的成本支出。对于多功能安全帽，4G的高功耗、高成本和部署复杂性使其在长期使用和大规模部署中的可行性较低，尤其不适合对便携性和成本敏感的场景。

2) Wi-Fi通信技术

Wi-Fi通信技术是近年来应用较多的矿用无线通信技术之一，与其他矿用无线通信技术相比，具有如下几个特点：组网灵活、融合性好、兼容性高、通信速率多样性、布网成本低等特点[19]。它工作在2.4GHz和5GHz频段，能够支持高速的数据传输和低延迟通信，广泛应用于各种场景中，尤其是在设备连接性强、数据量大的应用环境当中。对于矿井多功能安全帽的通信需求，Wi-Fi技术具有显著的优势，尤其是在已有Wi-Fi网络覆盖的矿井环境中。Wi-Fi通过部署中继器或Mesh网络可灵活扩展信号覆盖，适应矿井复杂地形和多障碍环境，确保信号稳定性和可靠性。如今有很多经济高效的Wi-Fi模块，如乐鑫的ESP8266或ESP32，具有低功耗、体积小并且易于集成等方面的特性，非常适合被拿来嵌入到安全帽的内部去，如此一来便能够显著地去降低设备在设计方面的难度以及整体的成本投入。因此在性能、成本和部署便捷性方面具有高可行性，是矿井复杂环境中的理想选择。

3) LoRa通信技术

LoRa(Long Range,超长距低功耗数据传输技术）是LPWA(Low Power Wide Area,低功耗广域)技术的代表，为物联网的低速率、低功耗、远距离、多连接应用而设计，在地面已成功应用于远程抄表、资产跟踪、智能停车、智慧社区、智慧农业等领域[20]。但是对矿井这种情况来说，将 LoRa移植在矿井中较小的矿工安全帽中，需要解决射频信号在巷道中传播时产生的路径损耗和多径衰落的问题，这2个问题使得通讯距离大打折扣，想要完全实现矿山长距离地下与地面的无线连接，还需要克服环境因素的影响[21]。对于矿井多功能安全帽的应用，LoRa技术在低频数据传输、设备状态监测和定位信息传输等场景中确实具有一定的优势，特别是在远距离、低带宽需求的情况下，能够有效延长设备的使用时间并减少电池更换的频率。然而，由于其数据速率低和响应时间较长，LoRa并不适合需要实时性较高的任务，如语音通信和实时数据传输以及命令接收，因此其在矿井多功能安全帽的全面应用中存在一定局限性。

综上所述，Wi-Fi通信技术在矿井中多功能安全帽的数据通信中，能够提供高效、稳定且低成本的解决方案，能够满足复杂环境下的实时数据传输的需求，是在矿井这样的特殊场景下可行性较高的技术选择。

2.2.5 远程控制APP技术可行性

远程控制 APP 是矿工安全帽系统当中一个重要的组成部分，旨在为矿井管理人员提供实时监控与控制矿工安全帽的能力。以下是几种 APP 开发技术的一个简要对比。

2.4%(205)

1) 原生开发技术

原生开发就是针对像Android、iOS这些不同的操作系统来选用它们官方所推荐的编程语言以及工具集去进行独立的应用构建。这种方式的核心优势就在于它能够去最大限度地挖掘以及利用设备的硬件资源还有操作系统的特性，这样一来，就可以去实现最佳的运行性能以及最流畅的用户体验了。不过，原生开发的主要不足之处就在于它成本比较高，而且开发周期也比较长。由于需要为不同的平台去分别组建开发团队并且编写独立的代码库，也就是在各自的移动操作系统上进行原生应用的开发，会带来昂贵的开发费用、耗费大量人力、增加修改维护难度，若仅仅对部分功能进行细微修改，也不能避免在各个操作系统上单独进行版本改动[22]。此外，原生开发需要专门的技术人员来开发和维护不同平台的代码，这对于小型团队或预算有限的项目来说，可能是一项沉重的负担。

2) 跨平台开发框架

跨平台框架为在多个操作系统上去高效部署应用提供了一个途径，在众多为高效多系统应用部署提供途径的跨平台框架当中，像React Native、Flutter等，Apache Cordova凭借着它的Web技术核心，成为了一个很关键的范例。它的核心价值就在于能够借助统一的代码库来显著缩短开发周期、降低成本，并且简化维护工作。其中的Cordova就是一款开源的移动开发框架，可用标准的 Web 技术进行跨平台移动应用程序的开发，在 Web 页面也可以使用 Cordova 丰富的插件调用原生代码，获取设备相关信息、调取手机摄像头等，Cordova 具有以下优点：跨平台，开发一个应用程序，可以在不同平台使用包括 Android，IOS；开发效率高，迭代更新容易；开发出来的 App 很小[23]。所以对于那些追求快速上市、想去移植现有Web应用，或者是在预算和时间有限，并且对极致原生性能要求不高的项目当中，Cordova就展现出了卓越的成本效益以及开发速度。它那庞大的社区以及丰富的插件生态也进一步增强了对原生设备功能的访问能力。

综上所述，Apache Cordova凭借着它低成本、快速开发的特性，特别适合于矿井环境当中对远程控制APP的快速部署需求，它的技术可行性很高。

4. 第3章系统设计与实现_第1部分

AI特征值：17.4%　　AI特征字符数 / 章节(部分)字符数：2017 / 11566

片段指标列表

序号	片段名称	字符数	AI特征	
8	片段1	553	显著	4.8%
9	片段2	1464	显著	12.7%

原文内容

第3章系统设计与实现

本章将会阐述多功能智能矿工安全帽的系统设计以及实现方案，这当中涵盖了总体的设计目标、系统架构、硬件模块的选型与电路设计，还有嵌入式软件以及远程管理 APP 的开发过程。这项设计是以 STM32 单片机为核心，去集成定位、环境监测、通信、照明和报警这些功能，并且借助物联网技术来实现远程监控，以此来满足矿井复杂环境下的安全需求。

3.1 总体设计

3.1.1 设计目标

智能矿工安全帽集成了精确定位、环境监测、照明、语音通信和远程管理这些功能，能够全面提升矿工在矿井复杂环境当中的安全保障以及工作效率。系统可以实现矿工位置的实时精确定位，误差控制在3米以内，定位信息会同步显示在本地 OLED 屏幕和远程 APP 上。温湿度传感器以及烟雾传感器会实时监测环境参数，异常的时候会触发本地声光报警，并且推送至远程平台。高亮度 LED 照明支持手动和远程控制，可以确保矿工在低光环境下的操作安全。对讲模块保障了矿工与指挥中心之间的语音通信，紧急情况下能够实现实时沟通。 Wi-Fi 模块和 MQTT 协议可以实现数据实时上传以及远程指令下发，系统还具备良好的扩展性，未来可以接入更多的传感器模块，从而增强安全保障。智能矿工安全帽为矿井安全管理提供了一种高效、可靠、智能化的解决方案。

4.8%(553)

3.1.2 系统架构设计

系统分成了智能安全帽终端、无线通信网络和远程管理平台即APP三部分。安全帽终端以 STM32F103C8T6 单片机主控制器， 硬件上集成了GPS 定位、温湿度、烟雾传感器、 LED 照明、蜂鸣器报警和 Wi-Fi 通信模块以及语音对讲模组，通过 MQTT 协议与 EMQX 服务器进行交互，实现收发数据。远程 APP 将会订阅 EMQX 服务器的 Topic，获取安全帽终端的实时数据用以展示再UI界面上并且管理员可以在APP上下发控制指令。语音通信则通过在MCU的控制下使用对讲模组来实现的。硬件模块通过 UART、 I^2C、 GPIO 等接口与单片机连接，软件采用了模块化设计，分层

解耦的架构使得程序更加的高效灵活。图3.1为系统架构图。

图3.1 系统架构图

3.2 硬件电路设计

硬件电路为主要分为五个部分，分别是单片机最小系统、电源模块、环境监测模块、通信模块以及语音对讲电路的设计。

3.2.1 单片机最小系统电路

为了确保MCU能够正确的与周围硬件电路进行通信，我们需要保证单片机的最小系统是完整的，这就包括了复位电路、boot启动电路以及晶振电路，还有SWD程序下载接口。具体来说利用LDO将USB的电压降至3.3V给整个电路供电。晶振电路是用来给整个数字电路信号的。复位电路使用官方推荐的上拉电阻方法并且并联电容进行滤波去耦，单片机的最小系统电路设计见图3.2。

图3.2 单片机最小系统

3.2.2 电源电路

系统通过USB-Type-C接口输入5V电源，使用AMS1117-3.3V LDO将电压降至3.3V，给MCU、环境监测传感器、通信模块供电。LDO输入端并联10μF和100nf电容滤除高频噪声，输出端并联10 μF和0.1 μF电容稳定电压，电路设计见图3.3。

图3.3 电源模块设计

3.2.3 传感器电路

1) 温湿度传感器电路

温湿度传感器模块选用的是广州奥松电子有限公司生产的DHT11 数字温湿度传感器。这款传感器在消费级应用当中是非常普遍的，这主要是因为它拥有简易性以及较高的性价比。

DHT11 的数据引脚会连接到STM32 的 GPIO（PA8）上，并且会配置为开漏输出，在外部会接入一个4.7 kΩ的上拉电阻并连接到3.3V。单片机会通过单总线协议来读取40位的数据，它的采样周期大约是2秒，主要是为了降低功耗，但其实这个采样频率已经完全符合矿井环境监测的需求，也足以保持矿工安全的实时监测。

图3.4 DHT11温湿度传感器

2) 烟雾传感器电路

烟雾传感器选用的是 MQ-2 气体传感器，他对液化气、甲醇的灵敏度比较高，同时也可以检测甲烷以及其他的可燃蒸汽。

MQ-2 的模拟输出引脚AOUT会直接连接到STM32 的 ADC 输入引脚 PA1 上的。通过配置STM32 内置的 12 位 ADC 去对这个电压进行采样，就能够得到 0 到 4095 的数字量，它会对应 0 到 3.3V的电压范围。加热丝引脚需要稳定供电，这里我们设计为连接到 3V 电源VCC，热丝的引脚是被连接到3V电源VCC上面的，如此能够确保传感器可以正常地来工作，因此需要先加以预热一段时间，才能够让输出变得稳定起来。

图3.5 MQ-2烟雾传感器

3) GPS电路

我选用的是u-blox 公司出品的 NEO-6M ，他与STM32微控制器的连接相对直接。模块的VCC引脚连接至系统提供的稳定的3.3V电源，GND引脚接地。核心的串行通信通过模块的TXD引脚连接至STM32的一个UART接口的RXD引脚，模块的RXD引脚连接至STM32对应UART接口的TXD引脚，在系统启动之后，MCU会去配置GPS模块，使其能够正常工作，获取当前自身的经纬度数据并通过串口发送给微控制器。

该设计利用了NEO-6M所具备的高灵敏度以及稳定的GPS引擎，如此一来便能够满足矿工安全帽所需要的定位精度。电路连接起来也很简单，功耗也比较适中，通过UART通信从而确保了实时、可靠地来传输位置数据。

图3.6 GPS模块原理图

3.2.4 数据通信电路

无线数据传输选用的是乐鑫公司出品的 ESP8266 Wi-Fi SoC，并经过安信可封装的ESP-01S模组无线数据传输模块的电路设计是ESP-01S模块的TXD引脚是连接到STM32的UART2接口PA2（RXD）上面的，而RXD引脚，则是被连接到PA3（TXD）上面，便能够实现串口数据的收发，从而去获取网络数据。模块的VCC引脚是被接入到3.3V电源，GND引脚则是直接接地，如此一来便能够确保稳定方面的供电。STM32常常会通过AT指令集来控制ESP-01S，去执行Wi-Fi网络扫描、连接SSID，又或是建立TCP/UDP连接以及数据收发之类的操作。

该设计是利用ESP8266所具备的高度集成以及内置的TCP/IP协议栈，从而来加以简化网络通信的，通过借助UART通信，从而确保了实时、可靠地来交互数据，如此一来便能够满足物联网应用那种低成本以及高效率的要求。

图3.7 ESP-01S电路原理图

3.2.5 语音对讲电路

该语音对讲模组采用的是模拟对讲技术，需要给它装上对应的麦克风和扬声器。将模组的CH1和CH2引脚接到单片机的两个GPIO引脚上，实现动态切换频道，CALL使用按键接地，这样保证按下时能够播放出拨打铃声，PTT引脚则是接MCU的外部中断引脚，这样按下按键之后就可以实现语音通话。

图3.8 语音对讲模组

图3.9 语音对讲电路设计

3.3 单片机程序设计

3.3.1 设计概述

单片机程序设计作为整个智能矿工安全帽系统的核心环节，它负责对各个硬件模块的工作进行协调，功能方面囊括了传感器数据采集、GPS定位解析、MQTT通信、以及报警控制与显示等。其整体设计是基于STM32F103C8T6微控制器来开展的，选用了C语言来进行开发，Keil MDK作为开发环境。这个程序选用的是中断驱动以及定时

器轮询机制，以此来确保系统可以高效地运行并具备实时性。图3.2所示的便是这个程序的流程图。

图3.8 程序流程图

3.3.2 资源分配

为了有效的管理mcu资源，不造成浪费，需要将mcu的外设资源进行仔细划分，表3.1是资源分配表。

表3.1 资源分配表

硬件模块外设资源引脚用途

GPS USART3 PB10 (TX), PB11 (RX) 定位

ESP8266 USART2 PA2 (TX), PA3 (RX) WIFI通信

DHT11 GPIOA PA8

温湿度检测

MQ-2 ADC1

PA1 烟雾浓度检测

OLED IIC PA5 (SCL), PA7 (SDA) 屏幕显示

LED GPIOA PA4 安全照明

KEY GPIOB PB0 开关照明

BUZZER GPIOA PA0,PB9 报警功能

3.3.3 功能实现

1) 烟雾浓度检测

烟雾浓度的监测工作是依赖于MQ-2传感器来开展的，该传感器所输出的模拟信号会借助STM32的ADC接口，也就是PA1，来进行采集。单片机当中的ADC模块负责把这个模拟信号转换为数字信号，并且会凭借其内置的定时器，即TIM2，去进行周期性的采集工作。每一次数据采集操作完成之后，STM32便会对当前的烟雾浓度同预先设定的阈值去进行比较。要是检测到的浓度超出了标准，那么STM32就会马上驱动蜂鸣器，也就是PA0，以及LED灯，也就是PA4，来发出对应的警报信号。其对应的关键代码如下所示：

```
/*--> [1.3] Smoke sensor reading */
smoke_adc_value = Get_ADC_Value(ADC_Channel_1, 100);
smoke_vol = (float)smoke_adc_value * (3.3f / 4096.0f);
RS = (5.0f - smoke_vol) / smoke_vol * 0.5f;
smoke_ppm = powf(11.5428f * R0 / RS, 0.6549f) * 100.0f;
// printf("Smoke: %.2f PPM\r\n", smoke_ppm);
采用中位值平均滤波算法：
u16 Get_ADC_Value(u8 ch, u8 times)
{
```

12.7%(1464)

```
    u32 temp_val = 0;
    u8 t;
    u16 adc_buf[255]; // Buffer for ADC values, ensure times <= 255
    u16 max_val, min_val;
    // Ensure at least 3 samples for valid filtering
    if(times < 3) times = 3;
    // Configure ADC channel, sequence, and sampling time
    ADC_RegularChannelConfig(ADC1, ch, 1, ADC_SampleTime_239Cycles5);
    // Collect ADC samples
    for(t = 0; t < times; t++)
    {
    ADC_SoftwareStartConvCmd(ADC1, ENABLE); // Start ADC conversion
    while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC)); // Wait for conversion
to complete
    adc_buf[t] = ADC_GetConversionValue(ADC1); // Store ADC value
    ADC_ClearFlag(ADC1, ADC_FLAG_EOC); // Clear end of conversion flag
    delay_ms(5); // Small delay for stability
    }
    // Find maximum and minimum values
    max_val = adc_buf[0];
    min_val = adc_buf[0];
    temp_val = adc_buf[0];
    for(t = 1; t < times; t++)
    {
    if(adc_buf[t] > max_val) max_val = adc_buf[t];
    if(adc_buf[t] < min_val) min_val = adc_buf[t];
    temp_val += adc_buf[t];
    }
    // Calculate average excluding max and min values
    temp_val = temp_val - max_val - min_val;
    return temp_val / (times - 2); // Return average of remaining values
    }
```

2) 温湿度检测

对于温湿度的检测这个方面，系统是选用了DHT11传感器来进行的。凭借单总线协议，STM32的GPIO口，也就是PA8，会同DHT11去进行通信，从而读取到温湿度数据。在数据采集的整个过程当中，需要去准确地对单片机的引脚时序进行控制，以此

来确保信号能够被正确地接收到。单片机会借助定时器来实现定时的数据采集工作，并且会把所读取到的温湿度值同设定好的安全阈值进行一番比较。要是所得到的数据超过了预设的这个设定值，单片机便会凭借GPIO口来触发蜂鸣器以及LED灯，去进行报警的动作。其核心代码如下所示：

```
//从DHT11读取一次数据
//temp:温度值(范围:0~50°)
//humi:湿度值(范围:20%~90%)
//返回值：0,正常;1,读取失败
u8 DHT11_Read_Data(u8 *humiH,u8 *humiL,u8 *tempH,u8 *tempL)
{
u8 buf[5];
u8 i;
DHT11_Rst();
if(DHT11_Check()==0)
{
for(i=0;i<5;i++)//读取40位数据
{
buf[i]=DHT11_Read_Byte();
}
if((buf[0]+buf[1]+buf[2]+buf[3])==buf[4])
{
*humiH=buf[0]; //坑啊原子哥，说明书明明是湿度在前温度在后
*humiL=buf[1];
*tempH=buf[2];
*tempL=buf[3];
}
}else return 1;
return 0;
}
//初始化DHT11的IO口 DQ 同时检测DHT11的存在
//返回1:不存在
//返回0:存在
u8 DHT11_Init(void)
{
GPIO_InitTypeDef GPIO_InitStructure;
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE); //使能PA端口时
```

钟

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8; //PA0端口配置
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure); //初始化IO口
GPIO_SetBits(GPIOA,GPIO_Pin_8); //PA0 输出高
DHT11_Rst(); //复位DHT11
return DHT11_Check();//等待DHT11的回应
}
```

1) 自动报警

系统的自动报警功能是基于定时器，也就是TIM2的周期性中断来开展的。在每一个定时器周期当中，单片机方面会去读取报警标志位，要是这个标志位被置位了，那么就会去执行报警的操作，也就是说蜂鸣器会响起，并且在主循环里面把对应的消息发送到服务器上。其对应的关键代码如下所示：

```
void TIM3_IRQHandler(void) // TIM3 interrupt handler
{
if (TIM_GetITStatus(TIM3, TIM_IT_Update) != RESET) // Check if the specified TIM interrupt occurred: TIM update interrupt
{
TIM_ClearITPendingBit(TIM3, TIM_IT_Update); // Clear TIMx interrupt pending bit: TIM update interrupt
if (alarmFlag) // If alarmFlag is set, toggle LED and BEEP
{
// LED1 = !LED1;
// BEEP = !BEEP;
LED1 = 0;
BEEP = 1;
}
else
{
LED1 = 1;
BEEP = 0;
}
}
}
```

2) WIFI通信

Wi-Fi通信模块，即ESP8266，是借由USART2接口，也就是PA2、PA3，同STM32单片机实现连接的，它选用的是AT指令集来进行配置以及通信。凭借UART接口，单片机能够把传感器数据以JSON格式传输至远程平台，并且还会去接收远程下发的命令。在数据上传以及控制指令的接收这个方面，STM32会借助定时器去开展数据传输周期的管理工作，与此同时，还会凭借中断去处理所接收到的命令。这样一来，就使得系统可以在低功耗的状态之下稳定地去运行，并且还能够在有需要的时候快速地去响应外部的指令。其对应的关键代码如下所示：

```
void ESP8266_Init(void)
{
GPIO_InitTypeDef GPIO_Initure;
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
// ESP8266 reset pin configuration
GPIO_Initure.GPIO_Pin = GPIO_Pin_8; // GPIOB8 - Reset pin
GPIO_Initure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_Initure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOB, &GPIO_Initure);
GPIO_WriteBit(GPIOB, GPIO_Pin_8, Bit_RESET);
delay_ms(250);
GPIO_WriteBit(GPIOB, GPIO_Pin_8, Bit_SET);
delay_ms(500);
ESP8266_Clear();
while(ESP8266_SendCmd("AT\r\n", "OK"))
delay_ms(500);
ESP8266_SendCmd("AT+RST\r\n", "");
delay_ms(500);
ESP8266_SendCmd("AT+CIPCLOSE\r\n", "");
delay_ms(500);
while(ESP8266_SendCmd("AT+CWMODE=1\r\n", "OK"))
delay_ms(500);
while(ESP8266_SendCmd("AT+CWDHCP=1,1\r\n", "OK"))
delay_ms(500);
while(ESP8266_SendCmd(ESP8266_WIFI_INFO, "GOT IP"))
delay_ms(500);
while(ESP8266_SendCmd(ESP8266_ONENET_INFO, "CONNECT"))
delay_ms(500);
ESP8266_INIT_OK = 1;}
```

3）GPS定位

在GPS定位模块方面，STM32是借由USART3接口，也就是PB10、PB11，同NEO-6M GPS模块实现连接。STM32会凭借NMEA协议去进行GPS模块所返回的定位数据的解析工作，从而来实时地获取到矿工的经纬度信息。GPS数据的处理工作是由STM32的UART接口以及中断机制来共同完成的。在每一次进行数据采集的时候，STM32会把经过解析后的GPS数据通过OLED显示屏，也就是PA5以及PA7，进行实时的显示操作，并且把它上传到远程管理平台。

3.4 远程控制APP设计

3.4.1 设计概述

远程管理APP，它在MineSafe系统里面担当一个移动端工具的角色，其目的在于为矿井安全去进行实时的监控以及设备方面的管理工作。管理员可以运用这个APP来进行查看像温度、湿度、以及烟雾等等的环境数据，也可以对头盔设备进行控制，以操作灯光的开关或者触发预警，同时还可以进行一个阈值的设置，来应对那些异常情况的发生。这个应用的开发工作是基于HTML5、CSS3以及JavaScript来开展的，并且它选用了Apache Cordova来进行打包工作，使其可以成为一个Android应用，这样它就能够适宜于在移动设备上加以使用。在数据通信这个方面，它是借助MQTT协议来实现的，如此便可以确保信息能够得到实时的传递。本小节内容将会介绍一下这个APP的架构设计、它所拥有的功能实现以及消息格式。为了能够将这些目标都加以实现，是选用了HTML5以及JavaScript来作为前端方面的核心技术，同时也会结合运用MQTT.js库来进行实时通信的处理，而Apache Cordova则会被运用在移动端的打包工作当中。这样的一种技术组合方式，它不仅降低了开发方面的成本，同时也对整个系统的可维护性以及扩展性进行了提升。

3.4.2 功能实现

在远程管理APP当中主要包含了如下的四个核心功能模块部分：用户认证、环境实时监测、设备控制以及状态日志。接下来，我将会逐一地去阐述它们的具体实现方法。

1）管理员认证

管理员认证这个功能的实现，是依靠登录页面来完成的，仅有获得授权的用户才被允许去访问该系统。登录界面会选用HTML表单，当中会包含用户名以及密码的输入区域，而CSS Flexbox布局则会确保这些元素能够居中来进行显示。JavaScript中的attemptLogin()函数会去进行验证，看输入的内容是不是预设值（具体来说，用户名和密码都被设定为“chan”）。要是验证顺利通过之后，那么登录页面便会被隐藏起来，之后主应用界面则会进行显示，并且还会去启动MQTT的连接过程，管理员认证的界面可以参考图3.9所示。

图3.9 管理员登陆界面

2）环境数据监测

环境数据监测主要是要去负责实时获取并展示智能头盔传感器所采集的环境数据，这当中囊括了温度、湿度、烟雾浓度、照明状态、地理位置以及危险状态。系统会通过MQTT协议去订阅“helmet/status”这个主题，从而接收JSON格式的状态消息，并且会调用updateMonitoringData(data)这个函数去解析并更新那些对应的界面元素；布局方面会采用CSS Grid，每项数据都会以卡片的形式来呈现，要是危险状态为真的时候，相应的卡片就会自动闪烁并且显示顶部的警告条。它主要是通过调用了编写的updateMonitoringData函数，以此来实现对各个数据的获取以及更新工作。其中，地理位置数据会借助convertToDMS函数把十进制度数转换为度分秒格式，同时也会结合半球标识来显示为中文标签（比如“东经”），并且会以两行的格式来呈现出来。警告条在危险状态触发的时候就会显示出来，管理员可以点击它来关闭。其核心逻辑就是：

```
client.on('message', (topic, message) => {
if (topic !== 'helmet/status') return;
try {
updateMonitoringData(JSON.parse(message.toString()));
} catch (e) {
console.error('解析状态数据失败:', e);
}
});
function updateMonitoringData({ danger, temperature, humidity, smoke } = {}) {
// 危险状态
if (danger !== undefined) {
dangerValue.textContent = danger ? '是' : '否';
dangerItem.classList.toggle('blink', danger);
monitorPanel.classList.toggle('warning-panel', danger);
alertBar.style.display = danger ? 'block' : 'none';
if (danger) addLogEntry('危险状态触发');
}
// 环境数据
if (temperature !== undefined) tempValue.textContent = temperature;
if (humidity !== undefined) humidityValue.textContent = humidity;
if (smoke !== undefined) smokeValue.textContent = smoke;
}
```

最终效果见图3.10：

图3.10 环境监测界面

3) 设备控制

设备控制这个功能，主要是运用远程命令来对智能头盔的照明系统和预警功能进行实时的管理工作，并且还支持管理员进行自定义阈值方面的设置。这个模块它会借助 MQTT 协议来和头盔终端进行通信，向 "helmet/cmd" 这个主题去发布 JSON 这种格式的控制指令，以此来实现对光源开关、预警触发以及像是温度、湿度和烟雾这些阈值进行灵活的控制。接下来的内容就是它详细的设计说明。

(1) 照明开关

当用户在 APP 上面点击了“开灯”或者“关灯”这样的按钮之后，系统便会去生成相应的指令，比如说 { "light_switch":"on" } 或者 { "light_switch":"off" }，并且把这个指令发布到 “helmet/cmd” 这个主题上面去。终端在接收到这个指令之后，会马上对 LED 灯组的状态进行切换，同时会把这个操作所产生的结果，通过状态上报的方式，反馈给后台系统。

实现代码：

```
/**
* 照明开关
* @param {'on'|'off'} state
*/
function setLight(state) {
const cmd = { light_switch: state };
client.publish(CMD_TOPIC, JSON.stringify(cmd), err => {
if (err) {
console.error('灯光控制命令发送失败:', err);
alert('灯光控制命令发送失败');
} else {
console.log(`Light ${state.toUpperCase()} command sent`, cmd);
addLogEntry(`灯光已${state === 'on' ? '开启' : '关闭'}`);
}
});
}
```

(2) 远程预警

在矿井当中出现了某些突发的危险状况的时候，那么管理员就可以在 APP 里面去发起远程预警的操作，也就是去发送 { "remote_warning":"activate" } 这样的指令。安全帽在接收到这个指令之后，它就会去触发蜂鸣器以及指示灯来进行报警，并且会把 “danger” 这个字段的数值设置成为 true，然后上报到管理平台那里，这样就能方便指挥中心方面去作出迅速的响应。而在需要解除预警的这个时候，那么就会去发送 { "remote_warning":"deactivate" } 这个指令。

5. 第3章系统设计与实现_第2部分

AI特征值：6.8%　　AI特征字符数 / 章节(部分)字符数：225 / 3299

片段指标列表

序号	片段名称	字符数	AI特征	
10	片段1	225	显著	6.8%

原文内容

```
/**
* 远程预警触发 / 取消
* @param {'activate'|'deactivate'} action
*/
function triggerWarning(action) {
const cmd = { remote_warning: action };
client.publish(CMD_TOPIC, JSON.stringify(cmd), err => {
if (err) {
console.error('预警命令发送失败:', err);
alert('预警命令发送失败');
} else {
console.log(`Warning ${action} command sent`, cmd);
addLogEntry(`远程预警已${action === 'activate' ? '启动' : '取消'}`);
}
});
}
```

(3) 阈值调整

为了能够灵活地去应对环境当中的一些变化，APP 这个应用它支持对温度、湿度以及烟雾浓度这些阈值去进行动态的修改工作。在修改界面里面输入了新的阈值之后，系统就会分别地去生成像是 { "temperature_threshold":30.0 }、{ "humidity_threshold":80.0 } 或者 { "smoke_threshold":2000.0 } 这一类的指令。终端在接收到这些指令之后，就会去更新它本地所存储的阈值参数，后续的传感器在进行数据采集的时候，就会依据这些新的参数来判断是否需要去触发报警。

关键代码：

```
// 温度阈值
```

```
function sendTempThreshold() {
const v = parseFloat(document.getElementById('tempThreshold').value);
if (isNaN(v)) { alert('请输入有效温度阈值'); return; }
sendThreshold({ temperature_threshold: v }, '温度');
}
// 湿度阈值
function sendHumidityThreshold() {
const v =
parseFloat(document.getElementById('humidityThreshold').value);
if (isNaN(v)) { alert('请输入有效湿度阈值'); return; }
sendThreshold({ humidity_threshold: v }, '湿度');
}
// 烟雾阈值
function sendSmokeThreshold() {
const v =
parseFloat(document.getElementById('smokeThreshold').value);
if (isNaN(v)) { alert('请输入有效烟雾阈值'); return; }
sendThreshold({ smoke_threshold: v }, '烟雾');
}
```

最终效果见图3.11：

图3.11 管理员控制面板界面

4) 状态日志

状态日志这个模块，它的主要作用是用来记录系统当中的一些事件，比如像是危险情况的触发、灯光的开关操作以及阈值的更新等等。这些日志条目会按照时间的倒序方式来进行显示，并且还运用了 CSS 样式来对日志的显示效果进行优化，以此来确保每一个条目都能够清晰地展示出来，同时也支持进行滚动的查看操作。它是凭借 addLogEntry 这个函数来动态地添加到界面当中的：

6.8%(225)

关键代码：

```
function addLogEntry(message) {
const now = new Date();
const logEntry = document.createElement('div');
logEntry.className = 'log-entry';
logEntry.textContent = `${now.toLocaleString()} - ${message}`;
logContainer.prepend(logEntry);
logRecords.push({ time: now, message });
```

}

日志效果如图3.12：

图3.12 日志界面

5) 通信消息格式

在通信协议方面，选用的 MQTT 这种协议，相关的消息会以 JSON 这种数据格式来进行传输，并且可以把它们划分为上行以及下行这两个主要的类别。

(1) 上行消息格式

上行消息就是由安全帽这一端发布到名为 “helmet/status” 的这个主题上面的，其内容主要囊括了环境方面的数据信息、安全帽自身的配置情况，以及它的当前状态。

具体来说，这些字段会包括环境数据，也就是指温度、湿度和烟雾这些信息，还包含了地理坐标，这个坐标是由经度、纬度以及所在的半球所共同组成的，另外还有设备的状态，比如说照明情况、是否处于危险状态，以及相关的阈值设定。

{
"temperature": 25.3,
"humidity": 60.5,
"smoke": 150.7,
"latitude": 39.9042,
"latitudeHem": "N",
"longitude": 116.4074,
"longitudeHem": "E",
"light": "off",
"danger": false,
"temperatureThreshold": 30.0,
"humidityThreshold": 80.0,
"smokeThreshold": 2000.0
}

(2) 下行消息格式

下行消息则是由 APP 这个应用发布到名为 “helmet/cmd” 的主题上面去的，它主要是被当作控制方面的指令来使用：

灯光控制：

{"light_switch": "on"} 或 {"light_switch": "off"}

远程预警：

{"remote_warning": "activate"} 或 {"remote_warning": "deactivate"}

阈值设置：

{"temperature_threshold": 30.0}、{"humidity_threshold": 80.0}、

{"smoke_threshold": 2000.0}

3.4.3 打包与部署

在着手进行移动端的部署工作之前，需要预先去进行 Node.js、Cordova CLI、JDK以及 Android SDK 的安装和配置工作，并且要把它们的可执行文件路径添加到系统的环境变量当中去。在确认了这些命令行工具都可以正常地加以运用之后，就需要去新建一个项目目录，并且借助 Cordova 来对项目的骨架进行初始化的操作。当进入到项目的根目录之后，接下来就需要去添加 Android 平台方面的支持。在这个时候，Cordova 这个工具就会和 Android SDK 进行通信，去下载所必需的 Gradle 插件以及平台相关的资源，并且还会去生成 Android 的原生工程结构，这样做能够方便后续进行资源整合的工作。需要把所有的前端资源，比如说像 HTML、CSS 以及 JavaScript 这些文件，都放置到 www 这个目录里面去，同时把 index.html 这个文件当作是应用的入口点来使用。还需要在 config.xml 这个文件当中去进行应用名称、版本号以及一个尺寸为 512×512 像素的 PNG 格式图标的配置工作，当 Cordova 在进行打包操作的时候，它会自动地去读取这些配置信息，并且把它们嵌入到最终生成的 APK 文件当中。最终打包好的APP如图3.13：

图3.13 打包好的APP

6. 第4章系统测试与验证

AI特征值：0.0%　　AI特征字符数 / 章节(部分)字符数：0 / 1598

片段指标列表

序号	片段名称	字符数	AI特征

原文内容

第4章系统测试与验证

本设计基于STM32F103C8T6单片机，设计了一款智能矿工安全帽，集成了实时定位、语音通信、照明功能、环境监测及自动报警与远程控制app五大功能模块。系统集成图如下：

图 4.1 系统集成图

4.1 实时定位测试

（1） 速度

GPS的获取地理位置的效率受到环境影响较大，在窗口测试需要8分钟。在世外空旷地方测试只需要4分钟左右即可获取地理信息。

（2） 精度

使用GPS获取的地理位置信息见图4.2：

图 4.2 MCU获取的地理位置信息

电脑高精度定位获取的位置信息见图4.3：

图 4.3 电脑定位

4.2 语音对讲测试

对语音对讲功能进行通话清晰度、信号延迟与抗干扰能力，确保通信畅通可靠。在距离50米的情况下能够清晰的语音通话，稍有延迟，对讲语音质量较为普通。

图 4.4 对讲功能测试

4.3 照明功能测试

1) 端侧按键控制LED照明开关控制测试

图4.5为未按下按键时，LED为关闭状态，图4.6为按键按下之后LED亮起。

图 4.5 按键为按下时

图 4.6 按键按下

经过反复100次测试的测试结果见表4.1。

表 4.1 端侧按键测试

按键按下次数 LED翻转次数

100次 100次

结论：使用端侧按键控制LED测试通过。

2) APP远程控制测试

图4.7为未按下按键时，此时可以看到设备上传上来的LED为关闭状态。图4.8为按键按下之后LED亮起后设备回传的LED状态为开启。

图 4.7 app未开启照明

图 4.8 点击开启之后APP接收到的最新状态

经过反复100次测试的测试结果见表4.2。

表 4.2 APP开关照明测试

按键按下次数 LED翻转次数

100次 95次

结论：APP因网络波动、阻塞等问题出现偶尔丢包，在正常范围内，测试通过。

4.4 环境监测及自动报警测试

1) 设备侧环境监测及地理位置信息与APP侧收到的数据对比

图 4.9 设备端环境参数值

图 4.10 APP接收到的环境参数值

经过接近两小时的密集观察，APP的上的值均是由设备端上传，测试通过。

2) 自动报警测试

使用酒精喷洒烟雾传感器，设备侧立马报警，约0.5秒后，数据上传至APP触发APP警告动画。

图 4.11 未喷洒酒精状态

使用酒精之后，烟雾传感器检测值超过阈值，触发报警，蜂鸣器响起，并上传报警逻辑到APP，见图。

图 4.12 烟雾报警

经过反复100次测试的测试结果见表4.2。

表 4.3 自动报警功能测试

触发异常环境参数次数 APP报警接收

50次 50次

由于环境异常不是瞬时操作会场持续一段时间，也就是说状态会反复上传多次，基本不存在丢包现象，自动报警功能测试通过。

4.5 远程控制APP测试

由于APP控制照明已经在4.3中测试，下面测试APP配置环境参数功能。

当前环境：温度31.1度，湿度51 %RH。分别设置温度和湿度阈值进行测试。

1）温度阈值设置

设置温度阈值为10度，蜂鸣器立马响起，大约1秒后设备状态上传，触发APP报警动画效果。

图 4.13 设置温度阈值

图 4.14 APP触发报警动画

表 4.4温度阈值功能测试

设置阈值次数设备回传阈值成功次数

50次 46次

偶尔出现丢包，成功率在90%以上，温度阈值设置功能测试通过。

2）湿度阈值设置

设置湿度阈值为50，蜂鸣器立马响起，大约1秒后设备状态上传，湿度阈值上传至APP，并且触发APP报警动画效果。

图 4.15 设置50湿度阈值

图 4. 更新后的湿度阈值

表 4.5温度阈值功能测试

设置阈值次数设备回传阈值成功次数

50次 46次

偶尔出现丢包，成功率在90%以上，湿度阈值设置功能测试通过。

知网AIGC检测服务

7. 第5章总结与展望

AI特征值：0.0%　　AI特征字符数 / 章节(部分)字符数：0 / 3971

片段指标列表

序号	片段名称	字符数	AI特征

原文内容

第5章总结与展望

5.1 主要工作与创新点

这个项目我总计耗时将近4个月。主要工作是搭了个多功能智能安全帽，集成了GPS定位、温湿度和烟雾监测、LED照明和APP的远程管理。硬件上，我把STM32、NEO-6M、DHT11、MQ-2、ESP8266和OLED整合成一个紧凑系统，利用Altium Designer进行PCB布线集成到一块板子，并进行打板。在此期间也是系统的学习了一下AD。软件上，我利用MCU实现了传感器驱动、数据处理和WIFI通信，APP基于Cordova框架利用HTML，CSS,JS开发，界面友好，功能齐全。测试验证了系统稳定性和实用性，定位精度、通信可靠性和响应速度都达到设计目标。

该项目创新点有如下四个。第一，使用了滤波算法对环境监测数据进行了处理，使得采集的数据更加的准取。第二，基于Cordova框架，以HTML、CSS、JavaScript开发APP，将实时数据监测、远程控制和参数配置同一平台，操作直观、一键切换，满足矿井安全管理需求。第三，利用Altium Designer对整个系统进行了圆形布局，贴合安全帽的外观，占用空间极小。第四，系统集成度高，成本非常低，且功能十分完善，利于使用。

5.2 后续研究工作展望

系统还有不少提升空间。硬件上，我想把电路小型化，换更高效的芯片和传感器，减小体积，戴着更舒服。电源得加上锂电池和充电管理，使得续航增加，并且打算使用更先进的算法和资源优化来实现低功耗。定位方面，矿下GPS信号差，我计划加UWB或RFID，解决室内定位问题。传感器可以再加CO和CH4检测，使得监测更全面[19]。

软件上，APP功能还能丰富，比如实时视频和风险预测。语音通信质量还可以提升。并且重要一点是安全帽的安全性也得考虑，使用加密算法对后天管理链路进行加密，使得通信信道绝对的安全。稳定性测试还得加强，尤其在高温高湿环境下跑更长时间，找出潜在问题。

长远看，我希望这套系统不只用在煤矿，建筑工地、隧道施工这些高危场景也能用。功能上可以扩展，比如加个心率监测，反馈工人的健康状态，以及能跟大数据平台对接，利用大模型分析历史数据，预测风险。总之，这只是个起点，未来还有很多可能性等着去实现。

致谢

我要真心感谢我的导师孟振亚老师。他在毕业设计中给了我很多指导，从选题到方案设计，再到硬件调试和论文写作，孟老师的专业建议总能让我找到方向。遇到技术难题时，他耐心帮我分析，还鼓励我大胆尝试，让我学到不少东西。他的严谨和关怀让我很敬佩，也点燃了我对物联网的热情。

同时，感谢一起奋斗的同学们。你们在代码调试和硬件搭建上帮了大忙，给我出了很多主意，分享了不少点子。有了你们的帮助和支持，项目才顺利完成。

最后感谢我在完成整个毕业设计过程中参考的各类文章的各位专家，你们的文章给了我很多启迪和思考，收获颇丰。

附录A 源代码

```
#include "main.h"
// Global variables
volatile u8 alarmFlag = 0;
volatile u8 alarm_is_free = 10;
u8 temperatureH, temperatureL, humidityH, humidityL;
float smoke_ppm;
extern nmea_msg gpsx;
// Sensor threshold settings
volatile u8 temperatureThreshold = 30;
volatile u8 humidityThreshold = 80;
volatile float smokeThreshold = 2000.0f;
// Main function
int main(void)
{
unsigned short tick = 0;
unsigned char *rxData = NULL;
HW_Init(); // Initialize hardware
Net_Init(); // Initialize network
TIM2_Int_Init(4999, 7199);
TIM3_Int_Init(2499, 7199);
BEEP = 0;
delay_ms(250);
BEEP = 1;
// Main loop
while (1)
{
if (tick % 40 == 0)
```

```
{
Sensor_Process();
}
Comm_Handler(&tick);
rxData = ESP8266_GetIPD(3);
if (rxData != NULL)
{
OneNet_RevPro(rxData);
}
delay_ms(10);
}
}
// Hardware initialization
void HW_Init(void)
{
Usart1_Init(115200);
delay_init();
OLED_Init();
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_0);
LED_Init();
KEY_Init();
DHT11_Init();
BH1750_Init();
ADCx_Init();
Usart2_Init(115200);
ESP8266_Init();
GPIO_Config_Init();
USART_Config_Init();
NVIC_Configuration();
}
// Sensor data processing
void Sensor_Process(void)
{
static float RS, R0 = 6.64f; // Reference resistance for MQ2 sensor
static uint16_t smoke_adc_value;
static float smoke_vol;
```

```
    GpsDataRead(); // Update GPS data
    DHT11_Read_Data(&humidityH, &humidityL, &temperatureH,
&temperatureL);
    // Read smoke sensor data
    smoke_adc_value = Get_ADC_Value(ADC_Channel_1, 100);
    smoke_vol = (float)smoke_adc_value * (3.3f / 4096.0f);
    RS = (5.0f - smoke_vol) / smoke_vol * 0.5f;
    smoke_ppm = powf(11.5428f * R0 / RS, 0.6549f) * 100.0f;
    // Alarm logic
    if (10 == alarm_is_free)
    {
    alarmFlag = ((humidityH >= humidityThreshold) || (temperatureH >=
temperatureThreshold) || (smoke_ppm >= smokeThreshold));
    }
    if (alarm_is_free < 10)
    {
    alarm_is_free++;
    }
    }
    // Communication handler (publish data)
    void Comm_Handler(unsigned short *tick)
    {
    if (++(*tick) >= 119)
    {
    char PUB_BUF[256];
    float lon = gpsx.longitude / 100000.0f;
    float lat = gpsx.latitude / 100000.0f;
    sprintf(PUB_BUF,
    "{"
    "\"temperature\":%d.%d,"
    "\"humidity\":%d.%d,"
    "\"smoke\":%.2f,"
    "\"latitude\":%.5f,"
    "\"latitudeHem\":\"%c\","
    "\"longitude\":%.5f,"
    "\"longitudeHem\":\"%c\","
```

```
"\"light\":\"%s\","
"\"danger\":%s,"
"\"temperatureThreshold\":%d,"
"\"humidityThreshold\":%d,"
"\"smokeThreshold\":%.2f"
"}",
temperatureH, temperatureL,
humidityH, humidityL,
smoke_ppm,
lat, gpsx.nshemi,
lon, gpsx.ewhemi,
(GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_4) == 1) ? "on" : "off",
(alarmFlag == 1) ? "true" : "false",
temperatureThreshold, humidityThreshold, smokeThreshold
);
OneNet_Publish(devPubTopic, PUB_BUF);
*tick = 0;
ESP8266_Clear();
}
}
```

附录B 原理图

说明:

1、支持中、英文内容检测;

2、AI特征值=AI特征字符数/总字符数;

3、红色代表AI特征显著部分，计入AI特征字符数;

4、棕色代表AI特征疑似部分，未计入AI特征字符数;

5、检测结果仅供参考，最终判定是否存在学术不端行为时，需结合人工复核、机构审查以及具体学术政策的综合应用进行审慎判断。

关注微信公众号