

毕业设计说明书（论文）中文摘要

摘要

传统的定位系统，如北斗和 GPS，依靠部署在地球轨道上的卫星群来提供精确的地理位置数据。这些系统在室外环境下能够提供广泛的覆盖范围和相对较高的定位精度，然而在室内定位会出现精度不足的问题。为了解决这一问题，本课题探索使用基于接收信号强度的室内可见光定位系统来提高室内定位的精度和可靠性。这种系统主要依赖于室内常见的 LED 照明设备，通过 LED 发出的光信号来进行定位。

本文提出的室内可见光定位系统的主要优势在于其不仅拥有较高的精度，而且可以依赖简单的基础设施。与传统的无线电频率（RF）技术相比，可见光定位技术在室内环境中不易受到电磁干扰，能够提供更为精确和稳定的定位结果。此外，由于 LED 灯具在现代建筑中普遍存在，这种系统的部署成本较低，且易于维护和升级。

关键词：可见光室内定位；STM32；多边定位算法；多传感器融合；

毕业设计说明书（论文）外文摘要

Abstract

Traditional positioning systems, such as BeiDou and GPS, rely on constellations of satellites deployed in Earth's orbit to provide precise geographic location data. These systems offer extensive coverage and relatively high positioning accuracy in outdoor environments. However, they face accuracy issues when used for indoor positioning. To address this problem, this study explores the use of an indoor visible light positioning system based on received signal strength to enhance the precision and reliability of indoor positioning. This system primarily relies on common indoor LED lighting fixtures, utilizing the light signals emitted by LEDs for positioning.

The main advantage of the indoor visible light positioning system proposed in this paper is its high accuracy, which can be achieved with simple infrastructure. Compared to traditional radio frequency (RF) technology, visible light positioning technology is less susceptible to electromagnetic interference in indoor environments, providing more accurate and stable positioning results. Additionally, since LED lights are widely used in modern buildings, the deployment cost of this system is relatively low, and it is easy to maintain and upgrade.

Keywords: Visible Light Indoor Positioning; STM32; Trilateration Algorithm; Multi-Sensor Fusion;

目 录

前 言	1
第一章 绪论	2
1.1 常见的室内定位方案.....	2
1.2 可见光室内定位简介.....	3
1.3 关于本课题.....	4
第二章 开发环境与主要技术	5
2.1 开发平台.....	5
2.2 主要技术.....	6
第三章 系统需求分析	10
3.1 定位算法.....	10
3.2 定位系统具体需求.....	10
第四章 系统设计与实现	12
4.1 整体框架.....	12
4.2 模块的选择.....	15
4.3 模拟数据的生成.....	20
4.4 接发端双向通信的实现.....	20
4.5 单片机矩阵运算的实现.....	23
4.6 OLED 屏幕显示的实现	24
4.7 光强分布数据的获取与处理.....	25
4.8 接收端与 LED 距离算法的实现.....	27
4.9 多边定位算法的实现.....	29
4.10 实物搭建.....	33
第五章 实物测试	35
5.1 程序烧录.....	35
5.2 实地测试.....	36
5.3 BUG 发现与测试总结	39
第六章 结束语	41

6.1 毕业设计的难点与创新.....	41
6.2 毕业设计的收获.....	42
参考文献	43
在校期间发表的研究成果及获奖	44
1 攻读学士学位期间的研究成果.....	44
2 荣获奖励.....	44
致 谢.....	46

前 言

在当今快速发展的社会中，随着人们生活和工作环境的不断演变，尤其是商场、医院及办公室等室内场所的扩建，人们对精准有效的室内定位技术的需求变得日益迫切。尽管传统的室内定位技术如 Wi-Fi 和蓝牙已被广泛采用，但它们仍然存在精确度不足和环境依赖性强的局限性。面对这些挑战，基于光通信的可见光室内定位技术（VLP）因其高精度、低成本以及易于部署的优势，正逐渐成为学术研究和工业应用的新焦点。

本论文以“基于 RSS 的室内可见光定位关键技术研究——可见光定位系统研发”为题，旨在探索通过技术创新来提升室内定位的精度与实用性。在选题依据和初步调研阶段，系统地回顾了国内外关于室内定位的先进研究和技术应用，广泛收集并分析了各种室内定位技术的数据和案例，尤其是聚焦于可见光定位技术的潜力与挑战。

本研究在南京工程学院计算机工程学院蔡玮教授和高级软件工程师许震的指导下进行。他们丰富的专业知识和实战经验为本课题的理论探索与实验实施提供了有力的支撑。此外，本项目亦得到了众多同学与技术人员的积极参与和协助，他们的共同努力使得研究工作得以顺利进行。

第一章 绪论

本章主要对常见的室内定位方案进行介绍，并对当前新兴的可见光室内定位方案从原理上进行简单的介绍。

1.1 常见的室内定位方案

随着人们生活水平的提高和城市化进程的加快，人们对室内导航和定位服务的需求也不断增加。在商场、机场、医院、办公楼等室内环境中，传统的 GPS 技术可能失效，因此需要更精确的室内定位服务来帮助人们准确找到目的地。当前室内定位已经有了多样的形式，常见的室内定位方案包括视觉定位、红外定位、Wi-Fi 定位、蓝牙定位和超声波定位等^[01]。

1.1.1 视觉定位

视觉定位是一种利用摄像头或深度相机获取环境信息的技术，通过图像处理和计算机视觉技术识别环境中的特征或目标物体，从而确定自身位置的方法。

1.1.2 红外定位

红外定位是一种利用红外线信号来确定目标位置的定位技术。它通过发送和接收红外线信号来实现对目标位置的测量和定位。红外定位系统通常包括红外传感器、发射器和接收器等组件。红外定位的原理是利用红外线信号在空气中的传播特性。

1.1.3 Wi-Fi 定位

Wi-Fi 定位是一种利用 Wi-Fi 信号来确定移动设备位置的技术。它基于 Wi-Fi 信号的强度和到达时间等信息，通过计算设备与已知 Wi-Fi 接入点之间的距离，以及多个 Wi-Fi 接入点之间的相对位置关系，来确定移动设备的位置。

1.1.4 蓝牙定位

蓝牙定位是一种利用蓝牙信号来确定移动设备位置的技术。它基于蓝牙信标或蓝牙网关发送的信号，通过移动设备接收这些信号，并计算其与已知信标或网关之间的距离和方向，从而确定设备的位置。

1.1.5 超声波定位

超声波定位是一种利用超声波信号来确定目标位置的定位技术。它通过发送超声波信号并测量信号的传播时间，从而计算目标与发送器之间的距离，进而确定目标的位置。

1.2 可见光室内定位简介

随着可见光通信技术的发展和應用，围绕可见光同样发展出新兴的室内定位方案。主流的可见光定位方案主要有^[02]：

1. AOA 到达角度定位法；
2. RSS 三边定位；
3. TOA 到达时间定位法；
4. TDOA 到达时间差定位法。

LED 可以通过直接发送位置信号给接收端实现定位，从原理来讲相对来说最容易实现，对软硬件性能需求也小，但精度低。想要提高精度，则可以利用接收端光电检测装置检测 LED 光源强度、角度信息，由于检测的信息不同，方法也可以大致分为以上几种。接下来将详细地讨论每种定位算法，介绍它们之间的区别。

1.2.1 AOA 到达角度定位法

AOA (Angle of Arrival) 到达角度定位法是一种基于接收器接收到信号的到达角度来确定移动设备位置的定位方法。它通过测量信号到达接收器的角度，从而确定信号源与接收器之间的方向，进而确定移动设备的位置^[03]。

1.2.2 RSS 三边定位

RSS 三边定位是一种基于接收信号强度 (Received Signal Strength, RSS) 的三边定位方法。在这种方法中，移动设备通过测量其与三个或以上已知位置的信号源之间的信号强度来确定自身的位置。

1.2.3 TOA 到达时间定位法

TOA (Time of Arrival) 到达时间定位法是一种利用信号到达接收器的时间来确定移动设备位置的定位方法。它通过测量信号从发送器到达接收器的时间差，然后

利用信号的传播速度来计算移动设备与发送器之间的距离，进而确定设备的位置。

1.2.4 TDOA 到达时间差定位法

TDOA (Time Difference of Arrival) 到达时间差定位法是一种利用信号到达不同接收器之间的时间差来确定移动设备位置的定位方法。它通过测量信号从发送器到达不同接收器的时间差，然后利用信号的传播速度来计算移动设备与发送器之间的距离，进而确定设备的位置。

1.3 关于本课题

本课题将通过软硬件设计一个基于 RSS 的可见光定位系统。该系统在三边定位的基础上拓展为多边定位算法，并增加了算法的健壮性。在系统使用方面，采用实时更新的策略，每收到发送端信号后就会更新一次坐标，避免了死等坐标信号，坐标由常见的二维坐标，通过 TOF 测距仪和 MPU6050 的多传感器融合，提升为三维坐标^[04]。

第二章 开发环境与主要技术

本章主要对涉及到的开发平台进行介绍，并对该课题开发过程中涉及到的硬件技术进行详细介绍。

2.1 开发平台

本课题的开发平台为：

操作系统：Windows 11

程序语言：C、Python

IDE 工具：Keil uVision5、Clion 、Pycharm

本课题是软硬件结合的项目，涉及到嵌入式系统开发和软件编程两个方面。在硬件部分，使用了微控制器和多种传感器进行数据采集和控制；而在软件部分，使用了 C 语言和 Python 语言进行程序开发，以及使用了多个开发工具进行代码编写、调试和仿真。

2.1.1 Keil uVision5

Keil uVision5 是一个集成的开发环境，提供了统一的界面，包括代码编辑器、编译器、调试器和仿真器等工具，方便开发者进行软件开发、调试和测试。它支持多种编程语言，如 C、C++ 和汇编语言，同时也支持多种微控制器架构，如 ARM Cortex-M、ARM7 和 ARM9 等。

2.1.2 Clion

Clion 是一款由 JetBrains 公司开发的专为 C 和 C++ 语言开发而设计的集成开发环境（IDE）。它不仅提供了强大的代码编辑器和智能代码完成功能，还包括代码导航、静态代码分析、调试器和版本控制集成等丰富功能，使开发者能够更高效地编写、调试和维护 C 和 C++ 项目。

2.1.3 Pycharm

PyCharm 是由 JetBrains 开发的专业 Python 集成开发环境（IDE），旨在提高 Python 开发者的生产力。它拥有强大的功能，包括智能代码编辑器、代码导航、代

码完成、调试器、单元测试工具、版本控制集成等。PyCharm 支持多种 Python 开发框架和库，如 Django、Flask、Pyramid 等，同时也支持 HTML、CSS、JavaScript 等前端语言的开发。其用户界面友好、功能丰富，是众多 Python 开发者的首选开发工具之一。

2.2 主要技术

2.2.1 嵌入式系统开发技术概述

嵌入式系统开发是指针对特定应用场景设计、开发和部署嵌入式硬件和软件系统的过程。这些系统通常集成在更大的设备或产品中，具有特定的功能和限制。嵌入式系统开发涉及硬件设计、嵌入式软件开发、系统集成和测试等多个方面^[05]。

在嵌入式系统开发的硬件设计阶段，开发者需要精心挑选适用于特定应用场景的微处理器、传感器、通信接口等硬件组件，并进行电路图设计和 PCB 布局。这一过程涉及对功耗、性能、成本和尺寸等多方面因素的综合考量，以确保设计能够充分满足系统的功能和性能需求。在选择硬件组件时，需要权衡各项指标，比如选择性能强大但功耗较高的处理器，还是选择功耗低但性能稍逊的处理器，以及根据系统的实际需求选择合适的传感器类型和通信接口。

在嵌入式软件开发方面，开发者需编写嵌入式程序，通常选用 C、C++ 等低级语言。此外，他们还需要了解特定硬件平台的驱动程序编程，以及实时操作系统（RTOS）或裸机编程技术。在软件开发过程中，主要目标是实现系统所需的功能，并通过优化资源利用率和性能来提高系统的效率^[06]。

系统集成是将硬件和软件组件有机地整合到一起，以确保它们能够协同工作、相互交互。这一过程可能涉及编写设备驱动程序，以确保硬件能够正确地与软件进行通信和控制。此外，还需要进行编译和链接嵌入式软件，以生成可在目标硬件平台上运行的可执行程序。在整合过程中，还需要进行系统调试，以确保各个组件能够正确地协作和执行其功能。最后，执行集成测试是非常重要的一步，以验证整个系统的功能和性能是否符合设计要求，并发现潜在的问题和缺陷。通过系统集成，可以确保嵌入式系统在实际应用中能够稳定、可靠地运行。

2.2.2 最小二乘法

最小二乘法（Least Squares Method）是一种常用的数学优化技术，用于拟合数据和估计参数。它的主要思想是通过最小化观测数据与理论模型之间的残差平方和来确定模型参数，从而找到最适合数据的模型。下面是基本步骤和原理：

1. 定义模型：其目的是确定一个理论模型来描述数据之间的关系。这个模型可以是线性的、非线性的、单变量的或者多变量的，具体取决于数据的特性和需要解决的问题。线性模型通常形式简单，适用于线性关系较为明显的情况，而非线性模型则更能描述数据之间的复杂关系。单变量模型涉及到一个自变量和一个因变量之间的关系，而多变量模型则可能涉及多个自变量和一个因变量之间的关系，以及多个因变量之间的关系。选择合适的模型是至关重要的，它直接影响到最小二乘法的有效性和结果的准确性^[07]。

2. 确定目标函数：确定目标函数是最小二乘法中的关键步骤之一。目标函数通常定义为观测数据与理论模型之间的残差平方和。残差是观测数据与模型预测值之间的差异，其平方用于消除正负差异带来的影响，并强调差异的大小。

3. 最小化目标函数：通过对目标函数进行求导，可以得到参数的估计值。这一步骤通常是在寻找目标函数的最小值或最大值时进行的，通过求导可以找到目标函数的极值点，从而确定参数的最佳估计值。这些算法通过不断迭代，调整参数值，使目标函数逐渐收敛到最小值，从而实现对参数的优化^[08]。

2.2.3 无线通信协议

无线通信协议可以根据其应用领域、工作频段、通信范围、功耗、数据速率以及通信模式等方面进行分类。下面根据应用领域进行分类并做详细介绍。

1. 个人通信领域

个人通信领域的无线通信是指用于个人设备之间的短距离通信和数据传输的无线技术和协议。这些技术和协议主要用于连接个人设备，如智能手机、耳机、智能手表、电脑等，实现数据交换、语音通话、音频传输等功能。以下是个人通信领域常见的无线通信技术和协议：

蓝牙（Bluetooth）： 蓝牙是一种低功耗的短距离无线通信技术。它可以连接手机、耳机、音箱、智能手表等个人设备，实现音频传输、数据同步、远程控制等功能。

Wi-Fi（无线局域网）： Wi-Fi 是一种高速的无线局域网技术，也常用于个人通信领域。通过 Wi-Fi 连接，个人设备可以接入互联网，进行在线聊天、视频通话、浏

览网页等活动。

2. 物联网（IoT）领域

物联网（IoT）领域的无线通信是指用于连接物联网设备之间的无线通信技术和协议。这些无线通信技术和协议在物联网应用中起着至关重要的作用，实现了设备之间的互联互通，数据的传输和交换，以及远程监控和控制等功能。以下是物联网领域常见的无线通信技术和协议：

Wi-Fi（无线局域网）：Wi-Fi 是一种常用的无线局域网技术，提供了高速的数据传输和较大的覆盖范围。在物联网应用中，Wi-Fi 通常用于连接家庭、办公室或工厂内的设备，实现数据的传输和远程控制。

Zigbee：Zigbee 是一种低功耗、短距离的无线通信协议，适用于物联网设备之间的低速率数据传输。在物联网中，Zigbee 通常用于构建大规模的传感器网络，实现智能家居、智慧城市等应用^[09]。

3. 移动通信领域

移动通信领域的无线通信是指用于移动设备之间的无线通信技术和协议，主要用于移动电话、移动数据和移动互联网等应用。以下是移动通信领域常见的无线通信技术和协议：

GSM（Global System for Mobile Communications）：GSM 是一种数字移动通信标准，广泛应用于全球范围内的移动电话系统。它使用时分多址（TDMA）技术进行信道分配，支持语音通信、短信和数据传输等功能。

5G（Fifth Generation）：5G 是第五代移动通信技术，具有更高的数据传输速率、更低的延迟和更大的网络容量，支持超高清视频、虚拟现实、物联网等应用。5G 采用了新的技术架构和频谱资源分配方式，包括毫米波、多用户多输入多输出（MU-MIMO）和网络切片等。

4. 卫星通信领域

卫星通信领域的无线通信是指利用人造卫星作为中继站，在地面、海洋或空中等区域提供覆盖范围广阔的通信服务。这种通信方式通过卫星与地面站或移动终端之间的无线连接，实现了长距离、大范围的通信传输。以下是卫星通信领域的一些关键特点和应用：

覆盖范围广泛：卫星通信可以覆盖地球上大部分地区，包括偏远地区、海洋、高空等无法覆盖的区域。这种覆盖范围的广泛性使得卫星通信成为连接人类社会各个

角落的重要通信手段。

长距离通信：卫星通信可以实现长距离的通信传输，克服了传统地面通信设施无法跨越大洋或障碍物的限制。这种特点使得卫星通信在远程地区、海上航行、航空航天等领域具有重要意义^[10]。

2.2.4 传感器技术

传感器是一种将物理量转换为可电信号或其他可记录的信号的装置，通常用于测量和监测环境中的各种参数。传感器技术在各种领域中都有广泛的应用，包括工业自动化、医疗保健、汽车行业、消费电子等^[11]。以下是传感器技术的一些关键方面：

1. 工作原理

传感器通过感知环境中的物理量（如温度、湿度、压力、光照等）并将其转换为电信号或其他可记录的信号。不同类型的传感器采用不同的工作原理，包括电阻、电容、电感、压阻、光电、超声波等。

2. 分类

传感器可以根据测量的物理量分类。常见的传感器类型包括温度传感器、湿度传感器、压力传感器、光敏传感器、加速度传感器、陀螺仪传感器、磁力传感器等。传感器还可以根据其工作原理进行分类，如电阻传感器、压阻传感器、光电传感器等。

第三章 系统需求分析

本章主要讨论了定位算法及其在实际应用中的扩展和具体需求。

3.1 定位算法

在原本的三边定位算法上进行拓展，以适应多 LED 的情况。在实际应用中，使用更多的 LED 可以提高定位的准确性和鲁棒性。因此，需要对现有算法进行拓展，使其能够适应多 LED 的情况。同时，算法应能处理特殊情况，如传入的 LED 数量小于 3 或大于 9，以及出现奇异矩阵的情况。

3.1.1 LED 数量不匹配

如果传入的 LED 数量少于 3 个，算法应能检测到并做处理，提醒用户至少需要 3 个 LED 进行有效定位，或者用替代的算法。其次，如果传入的 LED 数量超过 9 个，算法应忽略多余的 LED。

3.1.2 奇异矩阵处理

在实际计算过程中，可能会遇到奇异矩阵的情况，算法需要具备识别和处理这些情况的能力，具体方法包括使用伪逆矩阵来代替不可逆矩阵，或通过调整初始条件和迭代方法来重新计算，以确保计算的顺利进行和结果的稳定性。这样可以避免因奇异矩阵导致的计算失败或不准确，从而保证算法在各种情况下的鲁棒性和可靠性。

3.2 定位系统具体需求

3.2.1 实现无线通信

要求接收端和发射端能够实现高效的无线通信，即在发射端发送信号后，接收端能够迅速接收到该信号并作出及时响应，同时将处理后的数据回传给发射端，以确保通信过程的实时性和可靠性，从而满足系统对于快速和准确数据传输的需求。

3.2.2 数据采集

要求应有设备能够便捷的采集 LED 光强分布情况，选择适合的光敏传感器或光强传感器，能够准确地测量 LED 光源的光强分布情况。传感器应具有良好的灵敏度

和分辨率，以确保对光强变化的敏感度和准确度。

3.2.3 采集后数据处理

开发相应的数据处理和分析算法，对采集到的 LED 光强数据进行处理和分析。对数据滤波、数据校正，通过计算多个采样点的平均值来平滑数据，减小随机噪声的影响。以获得清晰的光强分布图像和数据统计信息。

3.2.4 发射端控制

发射端需要设计适当的 LED 驱动电路，以提供 LED 所需的电流和电压。LED 驱动电路通常包括电流限制器、电压稳定器等组件，以确保 LED 的稳定工作并延长其寿命。使用 STM32 生成控制信号，控制 LED 的开启和关闭。通过编程实现控制信号的生成和输出。

3.2.5 接收端数据采集

接收端需要搭载多个传感器，以便全面获取环境和运动数据，包括用于测量光照强度的光敏传感器，用于测定距地面高度的超声波或激光测距传感器，以及用于捕捉物体加速度和角速度的加速度计和陀螺仪。通过集成这些传感器，系统能够实现对光照条件、空间位置和运动状态的精确监控和分析，从而提供更全面和细致的数据支持。

3.2.6 接收端数据处理

采用多传感器融合算法，对从各个传感器获得的数据进行综合处理，确保能够准确还原出真实的光照强度，同时，通过结合加速度计和陀螺仪的数据，对接收端的倾斜情况进行实时校正，从而准确计算出接收端相对于地面的垂直高度，确保所有测量数据的高精度和可靠性。

第四章 系统设计与实现

本章主要对定位系统的整体框架进行设计，并实现光强分布的检测、接收端发射端双向通信、矩阵运算、LED 控制、OLED 显示等。

4.1 整体框架

4.1.1 项目的整体框架

为了灵活地应对需求的变化，项目选择使用标准库进行项目开发。标准库提供了丰富的通用功能和数据结构，提供了快速开发、稳定可靠的基础。通过利用标准库的成熟功能，能够更加高效地满足不断变化的需求，同时降低了项目的风险和开发成本。

标准库是指在软件开发中经过广泛使用和认可的、由编程语言的官方或第三方提供的一组标准化的函数、类、数据结构和模块等集合。这些函数和类通常被设计用来解决常见的问题，如字符串处理、文件操作、数据容器等。

4.1.2 项目流程图

光强检测设备在工作时，会不断采集光强数据，并判断是否正常，如果数据正常则采集下一组，不正常则跳过本组数据，工作流程如图 4-1 所示。

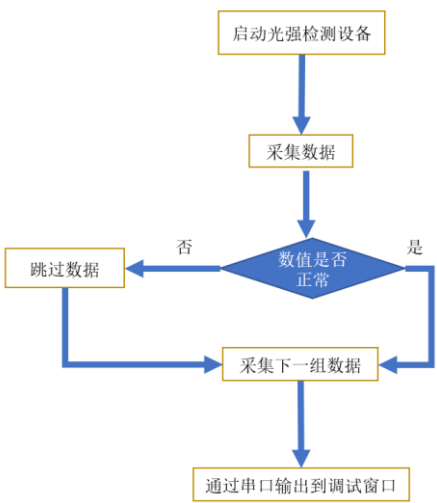


图4-1 光强检测设备工作流程图

接收端在工作时，会不断记录环境光强数据以及数据对应的 LED 坐标，根据记

录数据计算定位坐标并实时显示到 OLED 屏幕上，工作流程如图 4-2 所示。

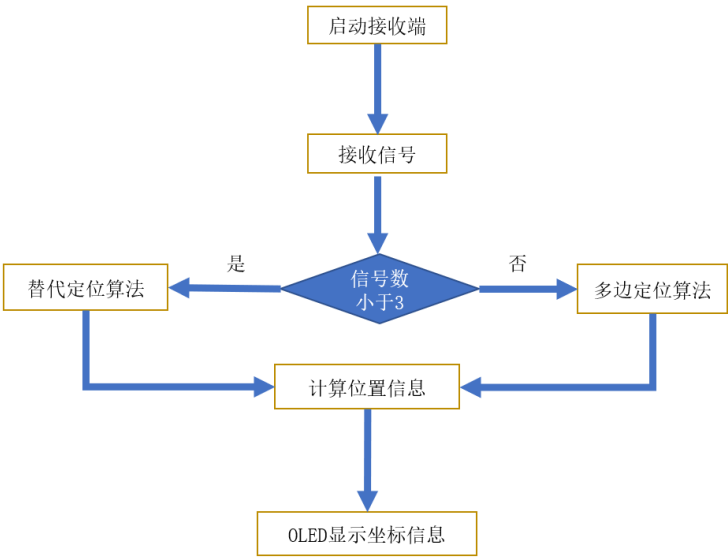


图4-2 接收端工作流程图

发射端不断循环点亮 LED，并发送 LED 坐标，工作流程如图 4-3 所示。

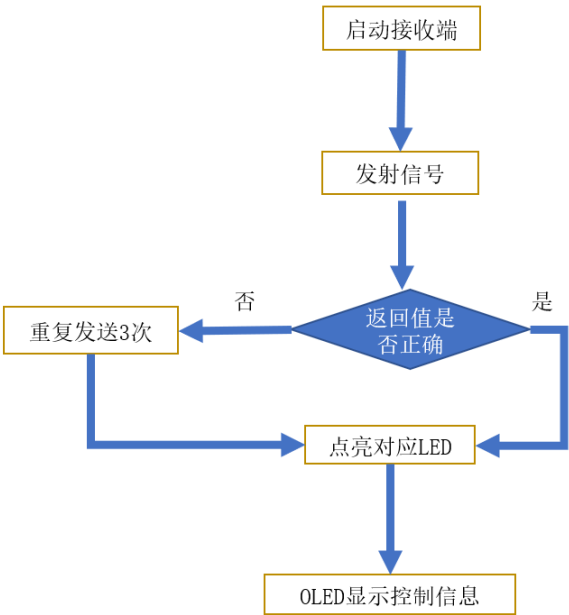


图4-3 发射端工作流程图

4. 1. 3 文件结构

本项目的发射端文件主要包括标准库文件，如 OLED.c 和 led.h。控制代码主体放在了 main.c 文件里。层次结构如图 4-4 所示

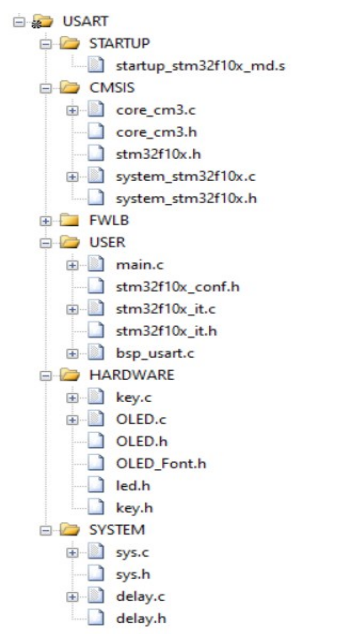


图4-4 发射端文件层次结构图

接收端文件主要包括标准库文件，如 MPU6050.c、bsp_adc.c 和 OLED.c。处理各个传感器接收到的数据的代码主体放在了 main.c 文件里。层次结构如图 4-5 所示

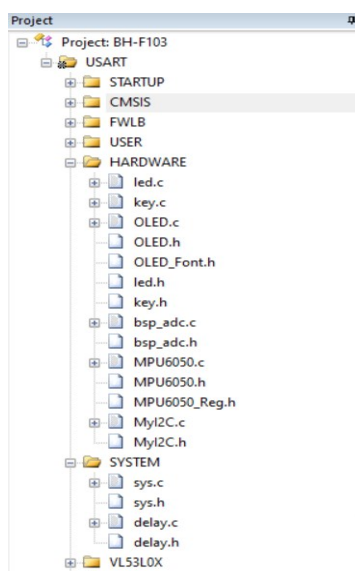


图4-5 接收端文件层次结构图

光强分布检测设备主要涉及到光敏电阻和 TOF 激光传感器，两个模块的标准库函数代码放在了文件夹 bsp 中，文件层次结构如图 4-6 所示

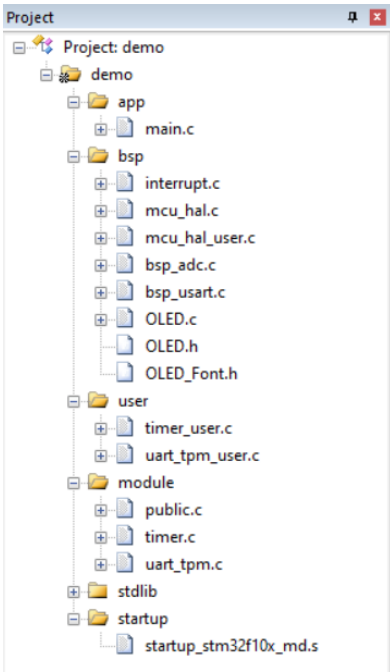


图4-6 光强分布检测设备文件层次结构图

4.2 模块的选择

整个定位系统是由软硬件组成，在硬件这块，对于模块的选择尤为重要，需要考虑到性能、价格等多方面的内容。不同设备之间选用的模块也不完全相同，目的也不完全一致，因此下面分设备详细介绍各个模块的选择。

4.2.1 光强分布检测设备

该设备主要涉及五个部分，模块示意图如图 4-7 所示

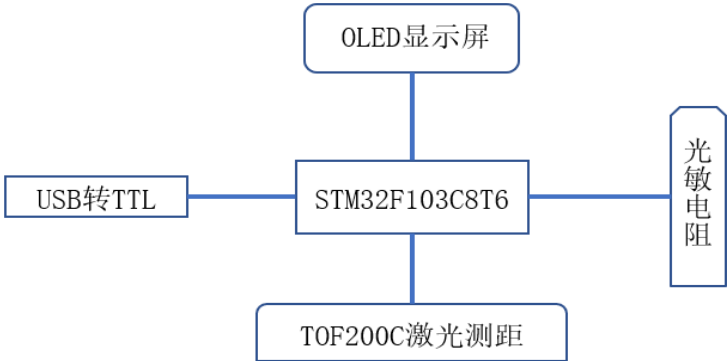


图4-7 光强分布检测设备模块示意图

所涉及模块主要是 STM32F103C8T6、TOF200C 激光测距模块和光敏电阻模块。

STM32F103C8T6 微控制器：微控制器是整个设备的核心，负责控制各个模块的工作，收集数据并进行处理。

TOF200C 激光测距模块：这个模块使用了飞行时间（TOF）技术来测量物体与模块之间的距离。它会向目标发送激光脉冲，然后通过测量激光从发送到接收的时间来计算目标物体的距离。

光敏电阻模块：这个模块用于检测环境光的强度。光敏电阻的电阻值随着光照强度的变化而变化，因此可以利用这一特性来测量光的强度。

USB 转 TTL 模块：这个模块用于将 STM32 微控制器的串行通信（TTL 级别）转换成 USB 信号，以便与计算机进行连接。

OLED 显示屏：这个显示屏用于显示设备采集到的数据或者状态信息，提供用户友好的界面。



图4-8 USB转TTL

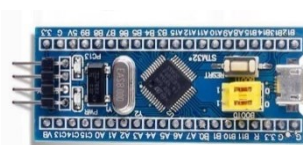


图4-9 STM32F103C8T6



图4-10光敏电阻



图4-11 TOF200C激光测距



图4-12 OLED显示屏

4.2.2 发射端设备

该设备分为两大部分，单片机除了需要控制直连的模块，还需要通过驱动板控制多个LED，如图4-13所示。

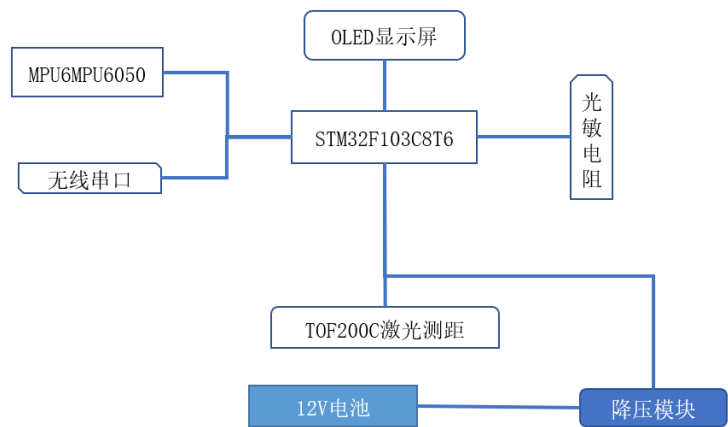


图4-13 发射端模块示意图

所涉及的模块主要是 STM32F103C8T6、OLED 显示屏、无线串口模块以及若干 LED 组成。

STM32F103C8T6 微控制器：作为设备的核心，STM32 微控制器负责控制各个模块的工作。它通过编程逻辑来实现接收到的数字编号与 LED 的对应关系，以及向 LED 发送控制信号。

OLED 显示屏：OLED 显示屏用于显示发送的数字编号。STM32 微控制器通过串行通信 I2C 与 OLED 显示屏通信，将接收到的数字发送给显示屏进行显示。

无线串口模块：无线串口模块用于与外部通信，接收外部发送的数字编号。它能够通过无线方式接收到来自另一端的数字编号，并将其传输给 STM32 微控制器。

LED：LED 组成了设备的输出部分，根据接收到的数字编号来进行控制。每个 LED 对应一个数字编号，当接收到对应的数字时，点亮相应的 LED，并熄灭其他所有 LED。

12V 转 5V 的降压模块：这个模块负责将输入的 12V 电压转换为稳定的 5V 电压，以供给 STM32 微控制器、OLED 显示屏和无线串口模块使用。



图4-14 无线串口

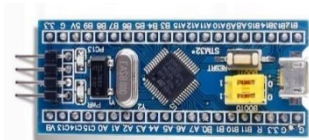


图4-15 STM32F103C8T6



图4-16 OLED显示屏



图4-17 LED



图4-18 降压模块

4.2.3 接收端设备

接收端涉及模块较多，因功能需要，MPU6050 需要放在 TOF200C 的正上方，线路连接示意图如图 4-19 所示

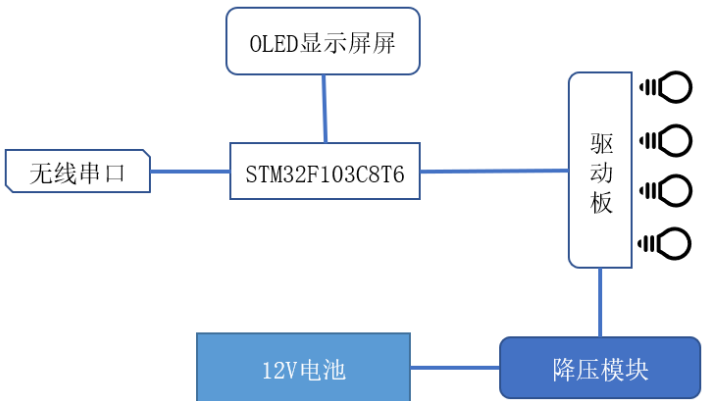


图4-19 接收端模块示意图

STM32F103C8T6 微控制器：作为设备的核心，STM32 微控制器负责整个系统的控制和数据处理。它通过编程逻辑来实现对其他模块的控制和数据处理，包括接收和处理来自无线串口模块的 LED 编号信息，控制 OLED 显示屏显示信息，以及对 TOF200C 激光测距模块和 MPU6050 进行数据采集和处理。

OLED 显示屏：OLED 显示屏用于显示接收到的数据信息，包括来自发射端的 LED 编号信息和其他系统状态信息。

MPU6050 模块：MPU6050 是一款集成了三轴陀螺仪和三轴加速度计的惯性测量单元（IMU）模块。它用于检测接收端的倾斜或者姿态变化，以及设备的运动状态。

TOF200C 激光测距模块：这个模块用于测量垂直地面的高度，用于高度测量的应用。

光敏电阻模块：光敏电阻模块用于检测环境光的强度，可能与其他传感器共同用于环境监测或者调节光照等。STM32 微控制器通过模拟输入引脚与光敏电阻模块连接，获取光强数据，用于相应的应用。

无线串口模块：无线串口模块用于与发射端通信，接收来自发射端发送的 LED 编号信息。

12V 电池和降压模块：12V 电池为整个系统提供电源，而降压模块则将 12V 电压降低为适合其他模块工作的稳定电压，通常是 5V。



图4-19 无线串口

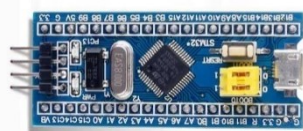


图4-20 STM32F103C8T6



图4-21 OLED显示屏



图4-22 光敏电阻

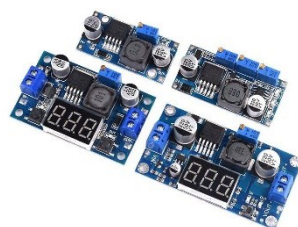


图4-23 降压模块



图4-24 MPU6050



图4-25 TOF200C激光测距



图4-26 12V电池

4.3 模拟数据的生成

在实验的初期阶段，需要生成大量的数据以满足模拟实验的需要，因此需要一个自动化实现计算节点与 LED 之间距离的程序，输出 LED 的坐标以及节点到各个 LED 的距离。

代码如下：

```
struct Point {double x, y, z;};  
// 计算 LED 坐标和 Node 到各个 LED 的距离  
double calculateDistance(struct Point p1, struct Point p2) {  
return sqrt(pow(p2.x - p1.x, 2) + pow(p2.y - p1.y, 2) + pow(p2.z - p1.z, 2));}  
void calculateDistances(struct Point inputPoint, struct Point *leds, int  
numLeds) {  
    double *distances = malloc(numLeds * sizeof(double));  
    for (int i = 0; i < numLeds; ++i) {  
        distances[i] = calculateDistance(inputPoint, leds[i]);  
    }  
    // 按照LED坐标和Node到各个LED的距离输出  
    printf("LED positions and distances:\n");  
    for (int i = 0; i < numLeds; ++i) {  
        printf("{%.2f, %.2f, %.2f},\n", leds[i].x, leds[i].y, distances[i]);  
    }  
    free(distances);}
```

定义结构体 Point，结构体 Point 包含三个成员变量 x、y 和 z，分别表示三维空间中的坐标。calculateDistance 函数接收两个 Point 结构体作为参数，并计算它们之间的欧氏距离。这个函数接收一个输入点 inputPoint 和 LED 的坐标数组 leds[] 以及 LED 的数量 numLeds 作为参数。在函数内部，动态分配一个长度为 numLeds 的双精度浮点型数组 distances[] 来存储节点到各个 LED 的距离。使用 calculateDistance 函数计算节点与每个 LED 的距离，并将结果存储在 distances[] 数组中。

4.4 接发端双向通信的实现

双向通信使得发射端和接收端可以相互发送和接收数据，实现了信息的交互与反馈。可以实时地从接收端获取处理后的数据，并在发射端进行相应的处理或调整。

通过双向通信，可以实现更加灵活和智能的控制和调节。

4.4.1 接收端

接收端使用中断处理和无限循环来处理从 USART 接收的数据。

代码如下：

```
while (1) {
    if (rx_index > 0) {
        ledIndex = rx_buffer[0] - '0'; // 将字符转换为整数索引
        ledIndex=ledIndex-1;
        if (ledIndex < 0 || ledIndex >= MAX_LEDS) {
            continue; // 如果接收到无效索引，跳过
        }
        else {
            received_str[0] = rx_buffer[0];
            received_str[1] = '\0';
            //在OLED上显示接收的字符串
            OLED_ShowString(1, 5, received_str);}
    }

    adc_sum = 0;
    rx_index = 0;
}

#define RX_BUFFER_SIZE 128 // 可以根据需要调整缓冲区大小
volatile uint8_t rx_buffer[RX_BUFFER_SIZE];
volatile uint16_t rx_index = 0; // 指向下一个空闲位置
// 串口中断服务函数
void DEBUG_USART_IRQHandler(void)
{
    uint8_t ucTemp;
    if (USART_GetITStatus(DEBUG_USARTx, USART_IT_RXNE) != RESET)
    {
        ucTemp = USART_ReceiveData(DEBUG_USARTx); // 接收数据
        if (rx_index < RX_BUFFER_SIZE) {
            rx_buffer[rx_index++] = ucTemp; // 存储数据到缓冲区
        }
    }
}
```

如果 rx_index 大于 0，说明缓冲区 rx_buffer 中有未处理的数据。ledIndex = rx_buffer[0] - '0'：将接收到的第一个字节转换为整数。检查计算后的索引是否在有效范围内。如果不在，则通过 continue 语句跳过当前循环迭代。如果索引有效，

则将接收的字符存储到 received_str 数组中，并在 OLED 上显示这个字符。

4.4.2 发射端

发射端使用串行通信与接收端交互，轮流发送 LED 编号信息，发送后等待回应，如果返回值正确则发送下一组数据，无返回值或返回值错误，则重试 4 次，如果仍错误，则发送下一组数据。

代码如下：

```
while (1)
{ if (rx_index > 0 && rx_buffer[0] == current_number)
{
    delay_ms(200);
    received_str[0] = rx_buffer[0];
    // 将接收到的字符赋给第一个位置
    received_str[1] = '\0';           // 设置空终止符
    // 在OLED上显示转换后的字符串
    OLED_ShowString(2, 10, received_str); // 更新显示接收到的数字
    rx_index = 0;
    // 重置接收索引
    // 准备发送下一个数字
    if (current_number < MAX_LED_NUM)
    { current_number++; }
    else
    {
        current_number = '1'; // 重置为1，循环发送
        display_str[0] = current_number; // 更新显示字符串
        display_str[1] = '\0';
        OLED_ShowString(3, 10, display_str); // 显示下一个要发送的数字
        // 发送下一个数字
        USART_SendData(DEBUG_USARTx, current_number);
        control_LED(current_number);
        retry_count = 0; // 重置重试计数器
    }
    // 如果重试次数达到最大值，则发送下一个数字
    if (retry_count >= MAX_RETRY_COUNT)
    {
        if (current_number < MAX_LED_NUM)
        {
            current_number++;
        }
        else { current_number = '1';

```

```
// 重置为1，循环发送 }
    USART_SendData(DEBUG_USARTx, current_number); // 发送下一个数字
    control_LED(current_number);
    retry_count = 0; // 重置重试计数器 }

else
{ // 增加重试计数器
    retry_count++;
    // 重试发送当前数字
    USART_SendData(DEBUG_USARTx, current_number); }
```

4.5 单片机矩阵运算的实现

在 STM32 微控制器上实现矩阵运算可以通过多种方式进行。在本项目中是引入 arm 内核的 DSP 库文件（含矩阵计算库），如图 4-27 所示。

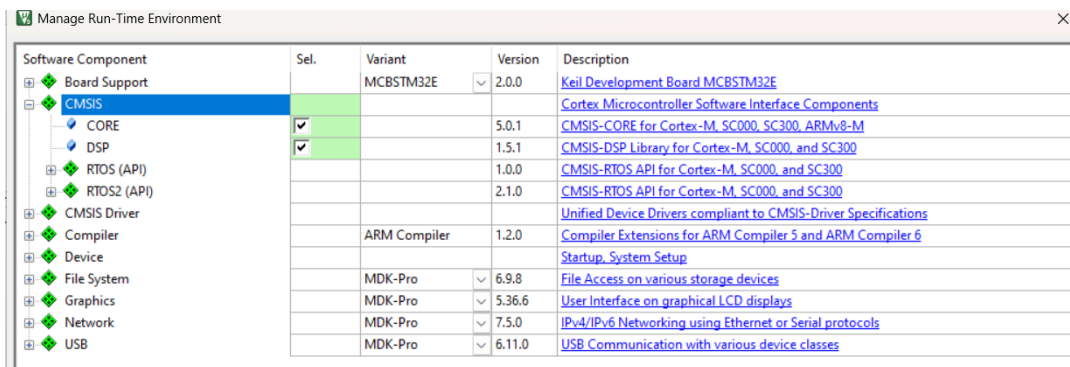


图4-27 引入库文件图

引入成功后还需要在” option of target” 中” C/C++” 中的 define 栏添加 “, ARM_MATH_CM3” 以声明开启 arm_math 库，如图 4-28。

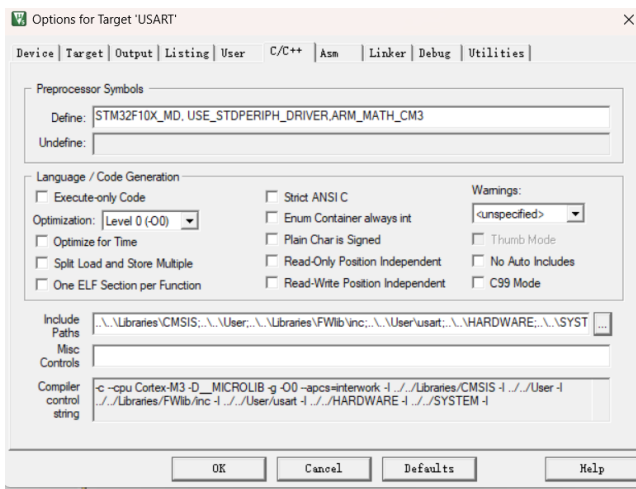


图4-28 申明开启arm_math库

4.6 OLED 屏幕显示的实现

在嵌入式系统中，利用 OLED 屏幕显示矩阵数据可以直观地呈现各种信息，如数值矩阵、状态信息或图形数据。为了方便调试算法，将每一个矩阵运算结果都是直接显示到 OLED 上，这种实时可视化反馈不仅加速了开发过程，还提高了代码的准确性和效率。

因涉及显示多种类型的结果，因此需要设计不同的显示方式，当显示矩阵时，代码如下：

```
void display_matrix(const char* name, arm_matrix_instance_f32* matrix) {
    char buffer[20];
    uint8_t line = 1;
    uint8_t column = 1;
    int i;
    int j;
    OLED_ShowString(line++, 1, (char *)name);
    for ( i = 0; i < matrix->numRows; i++) {
        for ( j = 0; j < matrix->numCols; j++) {
            snprintf(buffer, sizeof(buffer), "%5.2f", matrix->pData[i *
matrix->numCols + j]);
            OLED_ShowString(line, column, buffer);
            column += 5;
            if (column > 16) {
                column = 1;
                line++;
            }
        }
        line++;
        column = 1;
        if (line > 4) {
            line = 2;
        }
    }
}
```

在接收端定义了一个名为 display_matrix 的函数，它在 OLED 屏幕上显示一个浮点数矩阵的内容。它主要使用一个基于 ARM DSP 库的 arm_matrix_instance_f32 矩阵结构。函数接收一个矩阵名称以及矩阵实例的指针作为参数，并通过 OLED 显示它的内容。在调用函数时，传入矩阵名称，用于在屏幕上显示作为标题。参数 matrix

是指向 `arm_matrix_instance_f32` 结构体的指针，用于访问矩阵数据。变量初始化：
`buffer`：用于存储格式化的矩阵值，以字符串的形式在屏幕上显示。

4.7 光强分布数据的获取与处理

通过光强分布检测设备，如图 4-29。

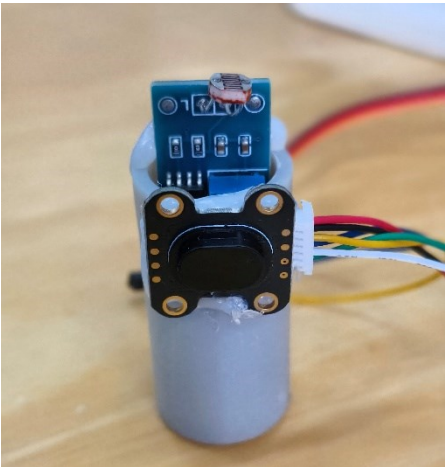


图4-29 光强分布检测设备

获得数据如图 4-30。

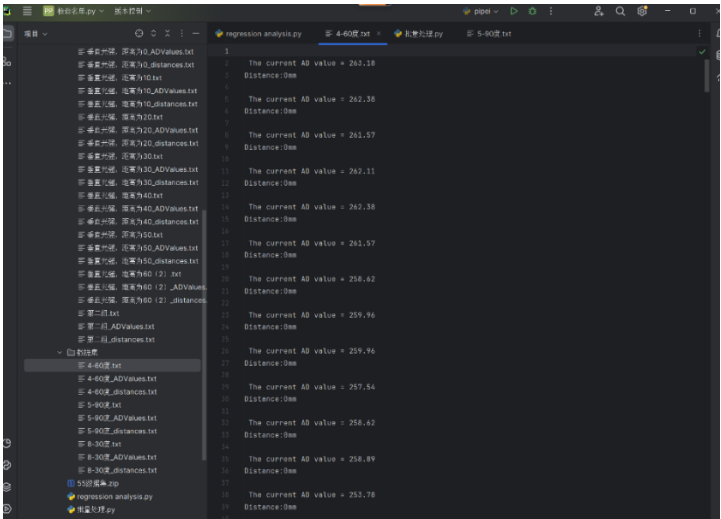


图4-30 光强分布数据图

需要将文本数据处理成对应的映射关系，代码如下：

```
# 创建字典来存储数据
distance_ad_values = {}
# 用来保存当前正在处理的距离
current_distance = None
```

```

last_distance = -1 # 假设距离不会小于0
import re
# 打开文件并逐行读取
with open('E:\PythonProjects\小工具\光强分布测试数据结果\数据集\5-90度.txt',
'r') as file:
    for line in file:
        line = line.strip()
        if line: # 确保行不为空
            if 'Distance:' in line:
                # 更新当前距离
                distance_str = line.strip().replace('Distance:',
                '').strip().replace('mm', '')
                if re.match(r'^-?\d+(\.\d+)?$', distance_str):
                    # 检查字符串是否表示数字
                    distance = float(distance_str)
                if distance >= last_distance: # 只有当距离大于或等于最后一个距离时才更新
                    current_distance = f"{distance:.2f}"
                    last_distance = distance
                    # 确保字典中有这个距离的键
                    if current_distance not in distance_ad_values:
                        distance_ad_values[current_distance] = []
                else:
                    print(f"Discarded invalid distance data: {distance_str}")
            elif 'The current AD value = ' in line:
                # 分割得到AD值
                parts = line.split('The current AD value = ')
                if len(parts) == 2 and current_distance is not None:
                    ad_value_str = parts[1].strip()
                if re.match(r'^-?\d+(\.\d+)?$', ad_value_str): # 检查字符串是否表示数字
                    ad_value = float(ad_value_str)
                    # 添加AD值到当前距离的列表
                    distance_ad_values[current_distance].append(ad_value)
                else:
                    print(f"Discarded invalid AD value data:
                    {ad_value_str}")
# 处理距离和AD值，计算平均值
for distance, values in distance_ad_values.items():
    if values: # 确保列表中有值
        if all(isinstance(value, (int, float)) for value in values):
            # 检查值是否都是数字
            average_value = sum(values) / len(values)
            # 将平均AD值更新到原始字典中

```

```

        distance_ad_values[distance] = average_value
    else:
        print(f"Invalid values for distance {distance}: {values}")
# 输出距离和对应的平均AD值数组的C语言格式
print("// 距离和对应的平均AD值数组")
print("const float distances[] = {", end="")
print(", ".join([distance.replace('mm', '') for distance in
distance_ad_values.keys()]), "); // 单位: mm")
print("const float average_ad_values[] = {", end="")
print(", ".join([f"{value:.2f}" for value in distance_ad_values.values() if
isinstance(value, (int, float))]),
"); // 对应的平均AD值")

```

处理完后输出数据示意如下：

```

// 距离和对应的平均AD值数组
const float
distances[] = {1.00, 2.00, 4.00, 6.00, 10.00, 21.00, 22.00, 26.00, 29.00,
33.00, 46.00, 48.00, 50.00, 52.00, 55.00, 570.00, 571.00, 574.00, 579.00, 581.00,
589.00, 602.00, 608.00}; // 单位: mm
const float
average_ad_values[] = {347.11, 345.59, 346.03, 346.17, 345.23, 346.70,
347.24, 346.97, 347.24, 348.58, 347.78, 348.58, 2653.86, 2684.20, 2699.24, 2712.40,
2960.54, 3079.79, 3090.26, 3105.03, 3101.00}; // 对应的平均AD值

```

4.8 接收端与 LED 距离算法的实现

为了实现多边定位算法，必须准确计算接收端与每个 LED 之间的直线距离。这种算法的基本原理是通过测量接收端和多个 LED 之间的距离，并结合每个 LED 的已知位置，来确定接收端在空间中的具体位置。

首先，需要在空间中确定每个 LED 的坐标。每个 LED 的坐标位置是固定且预先已知的，通常在系统初始化或部署阶段就已经确定。LED 坐标初始化代码如下：

```

//具体LED坐标
float32_t led_positions[4][2] = {
    {12.5, 12.5},
    {-12.5, 12.5},
    {-12.5, -12.5},
    {12.5, -12.5}};

```

通过之前测算得到的光强距离分布数据，可以将其转变成接收端距离 LED 投影

点的距离 x ，在 LED 高度 d 已知的情况下，通过公式可以计算得出，接收端和 LED 之间的距离，计算公式如下：

$$d = \sqrt{x^2 + h^2}$$

现在需要解决的如何求得 x ，基本思路是通过光敏电阻得到当前的光强，然后根据先前处理好的映射关系求得 x 。

代码如下：

```
int findClosest(float target, const float arr[], int n) {
    // 二分搜索查找最接近的元素索引
    int left = 0, right = n - 1;
    // 如果目标值小于数组的第一个元素，返回第一个元素的索引
    if (target <= arr[0]) return 0;
    // 如果目标值大于数组的最后一个元素，返回最后一个元素的索引
    if (target >= arr[n - 1]) return n - 1;
    while (left <= right) {
        int mid = left + (right - left) / 2;
        // 如果目标值在当前元素和下一个元素之间，返回当前元素的索引
        if (arr[mid] <= target && target <= arr[mid + 1]) {
            return (target - arr[mid] >= arr[mid + 1] - target) ? (mid + 1) :
                mid;
        }
        // 如果目标值小于当前元素，向左搜索
        if (arr[mid] > target) {
            right = mid - 1;
        }
        // 否则，向右搜索
        else {
            left = mid + 1;
        }
    }
    return -1;
}

for (i = 0; i < 3; ++i) {
    ADC_ConvertedValueLocal = (float)ADC_ConvertedValue / 4096 * 3.3;
    adc_sum += ADC_ConvertedValueLocal;
    // 多次取值
    delay_ms(50);
}
//计算平均值
adc_avg = (adc_sum / 3)*1000;
closest_index = findClosest(adc_avg, average_ad_values,
```



```

sizeof(average_ad_values) / sizeof(float));
closest_distance = (distances[closest_index]/10);
distance = sqrt(pow(H, 2) + pow(closest_distance, 2));

```

这段代码主要用于对 ADC 转换结果进行处理，并根据处理后的平均值来获取最接近的距离值。在一个循环中，首先进行了三次 ADC 转换，并将转换得到的值存储在 ADC_ConvertedValue 中。每次转换后，将转换得到的值转换为电压值，然后将其存储在 ADC_ConvertedValueLocal 中。将每次转换得到的电压值相加，并存储 adc_sum 中。每次转换之间还有一个 100 毫秒的延迟。接下来是一个用于查找最接近值的函数 findClosest。

4.9 多边定位算法的实现

4.9.1 理论推导

RSS 三边定位法即根据接收信号的强度来实现定位。定位过程中，通过设备测量三个不同参考点的信号强度，依据物理学模型计算出三个距离值，再通过几何关系求解目标节点的位置。

如果任意 LED 被遮挡则会导致定位失效，采用多个 LED 则可以有效避免意外遮挡，人员物体移动带来的定位影响。基于此，对实验拓扑图重新设计如图 4-31。

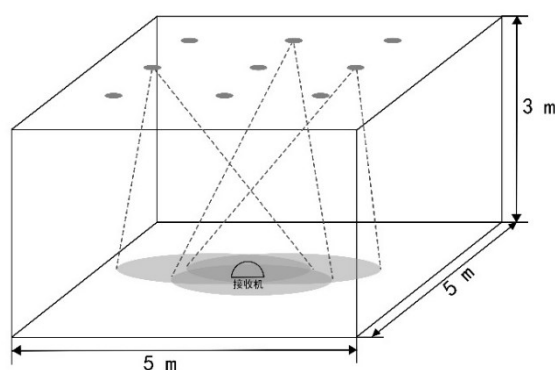


图4-31 实验拓扑图

已知多个 LED 的坐标分别为 $(x_1, y_1, z_1), (x_2, y_2, z_2) \dots (x_n, y_n, z_n)$ ，待定位节点 Node 的坐标为 (x, y, z) 。通过强度估计法测得 Node 点到 LED 的距离分别为 $d_1, d_2 \dots d_n$ ，则可联立方程组：

$$\begin{aligned}
 d_1^2 &= (x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 \\
 d_2^2 &= (x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2 \\
 &\vdots \\
 d_n^2 &= (x_n - x)^2 + (y_n - y)^2 + (z_n - z)^2
 \end{aligned}$$

针对 $z_1 = z_2 \dots z_n = H$ (H 为房屋高度), $z=0$ 情况, 此时, Node 坐标 (x, y) 可以通过求解下面两个线性方程获得。将其整理成矩阵形式:

$$VX = Q$$

其中

$$V = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ \vdots & \vdots \\ x_n - x_1 & y_n - y_1 \end{bmatrix}, Q = \frac{1}{2} \begin{bmatrix} (d_1^2 - d_2^2) + (x_2^2 + y_2^2) - (x_1^2 + y_1^2) \\ \vdots \\ (d_1^2 - d_n^2) + (x_n^2 + y_n^2) - (x_1^2 + y_1^2) \end{bmatrix}$$

根据上式定义残差值 r

$$r = Q - VX$$

残差的平方

$$f(X) = r^2 = (Q - VX)^2 = (Q - VX)^T(Q - VX)$$

对 $f(X)$ 求导并令其为零

$$\frac{df(X)}{d(X)} = 2V^T VX - 2VQ = 0$$

可以得到 X 的解

$$X = (V^T V)^{-1} V^T Q$$

在以上计算过程中, 只要三个 LED 不共线, 方程组即存在唯一解。

4.9.2 对于算法的编程实现

定义一个用于矩阵乘法的 C 语言函数, 函数名为 `Matrix multiply(Matrix mat1, Matrix mat2)`。该函数接收两个矩阵作为参数, 分别为 `mat1` 和 `mat2`, 并返回这两个矩阵的乘积。

代码如下:

```
Matrix multiply(Matrix mat1, Matrix mat2) {
    if (mat1.cols != mat2.rows) {
        fprintf(stderr, "Error: Matrix dimensions do not match for
            multiplication.\n");
        exit(EXIT_FAILURE);
    }
    Matrix result = createMatrix(mat1.rows, mat2.cols);
    for (int i = 0; i < result.rows; i++) {
        for (int j = 0; j < result.cols; j++) {
            result.data[i][j] = 0.0;
        }
    }
    for (int i = 0; i < mat1.rows; i++) {
```

```
        for (int j = 0; j < mat2.cols; j++) {
            for (int k = 0; k < mat1.cols; k++) {
                result.data[i][j] += mat1.data[i][k] * mat2.data[k][j];
            }
        }
    }
    return result;
}
```

定义一个名为 `multiplyWithRegularization` 的函数，这是一个进行矩阵乘法并允许添加正则化项的函数。该函数接收两个矩阵 `mat1` 和 `mat2`，以及一个双精度浮点数 `lambda` 作为正则化系数。函数的目的是计算 `mat1` 和 `mat2` 的乘积，并在结果矩阵的对角线上添加正则化项。

代码如下：

```
// 矩阵乘法，允许加入正则化项
Matrix multiplyWithRegularization(Matrix mat1, Matrix mat2, double lambda) {
    Matrix result = createMatrix(mat1.rows, mat2.cols);
    for (int i = 0; i < result.rows; i++) {
        for (int j = 0; j < result.cols; j++) {
            result.data[i][j] = 0.0;
            for (int k = 0; k < mat1.cols; k++) {
                result.data[i][j] += mat1.data[i][k] * mat2.data[k][j];
            }
        }
    }
    // 在对角线上添加正则化项
    if (mat1.rows == mat2.cols) { // 仅当结果是方阵时添加
        for (int i = 0; i < result.rows; i++) {
            result.data[i][i] += lambda;
        }
    }
    return result;
}
```

定义函数名为 `inverse` 的函数，用于计算给定矩阵的逆，如果存在的话。该函数使用增广矩阵和高斯-约当消元法来实现。函数首先检查输入的矩阵 `mat` 是否为方阵（即行数等于列数）。这是矩阵可逆的必要条件之一。如果不是方阵，则打印错误信息，并返回一个空的矩阵结构体。为了避免在消元过程中出现数值问题，函数通过查找当前列绝对值最大的行，并将其与当前行交换。

代码如下：

```
// 使用增广矩阵和行操作求矩阵逆，使用高斯-约当消元法
Matrix inverse(Matrix mat) {
    int n = mat.rows;
    if (n != mat.cols) {
        fprintf(stderr, "Error: Only square matrices can be inverted.\n");
        Matrix empty = {0, 0, NULL};
        return empty;
    }

    Matrix aug = createMatrix(n, 2 * n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            aug.data[i][j] = mat.data[i][j];
            aug.data[i][j + n] = (i == j) ? 1.0 : 0.0;
        }
    }

    for (int i = 0; i < n; i++) {
        double maxVal = fabs(aug.data[i][i]);
        int maxRow = i;
        for (int k = i + 1; k < n; k++) {
            if (fabs(aug.data[k][i]) > maxVal) {
                maxVal = fabs(aug.data[k][i]);
                maxRow = k;
            }
        }
        if (maxRow != i) {
            for (int j = 0; j < 2 * n; j++) {
                double temp = aug.data[i][j];
                aug.data[i][j] = aug.data[maxRow][j];
                aug.data[maxRow][j] = temp;
            }
        }
        double divisor = aug.data[i][i];
        if (fabs(divisor) < 1e-6) {
            fprintf(stderr, "Error: Matrix is singular, cannot be
            inverted.\n");
            freeMatrix(&aug);
            Matrix empty = {0, 0, NULL};
            return empty;
        }
    }
}
```

```
        for (int j = 0; j < 2 * n; j++) {
            aug.data[i][j] /= divisor;
        }
        for (int k = 0; k < n; k++) {
            if (k != i) {
                double factor = aug.data[k][i];
                for (int j = 0; j < 2 * n; j++) {
                    aug.data[k][j] -= factor * aug.data[i][j];
                }
            }
        }
    }
    Matrix inv = createMatrix(n, n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            inv.data[i][j] = aug.data[i][j + n];
        }
    }
    freeMatrix(&aug);
    return inv;
}
```

4.10 实物搭建

在有了前面论述内容的基础后，可以着手于最终实物的搭建。一共分为 3 个部分，第一部分是实验场景，用来做实验，第二部分是接收端，需要将前面所提到的模块整合调试，第三部分是发射端，控制 LED 和发送信号。

4.10.1 实验环境搭建

为了构建一个 50cm x 50cm x 50cm 的立方体实验环境，首先将 8 根水管切割成合适的长度，每根长度为 50cm。接着，使用四个三通连接器，每个连接器用于立方体的一个角，以连接三根相交的水管，形成立方体的框架。这样，每个水管的一端插入三通连接器中心孔，而另一端则延伸至下一个三通连接器。通过这种方式，可以确保所有水管均匀分布，立方体结构稳固且对称。最后，检查所有连接点确保没有松动，以保证实验环境的稳定性和安全性。最终成品如图 4-32。



图4-32 实验环境

4.10.2 发射端模块整合

将无线串口模块、OLED 模块、电路板与微控制器连接，如图 4-33。



图4-33 发射端实物图

4.10.3 接收端模块整合

将 STM32F103C8T6, OLED 显示屏, MPU6050, TOF200C 激光测距模块, 光敏电阻, 无线串口模块组合，如图 4-34。

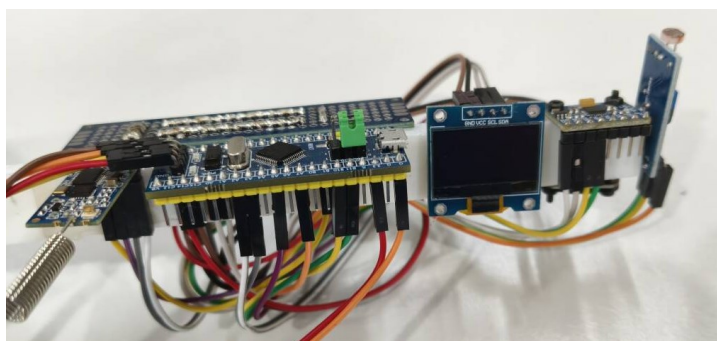


图4-34 接收端实物图

第五章 实物测试

5.1 程序烧录

连接目标设备：将 STM32 开发板通过 USB 线连接到计算机上，确保设备被正确识别并与 Keil μ Vision 5 软件连接。如图 5-1。

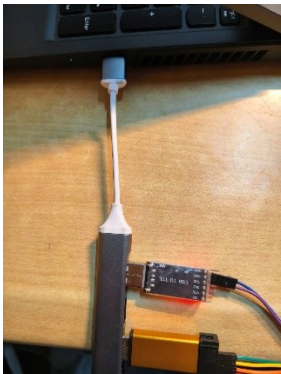


图5-1 连接开发板

编译工程：在 Keil μ Vision 5 中打开工程文件（.uvprojx），然后选择 Translate 选项编译工程。如图 5-2。

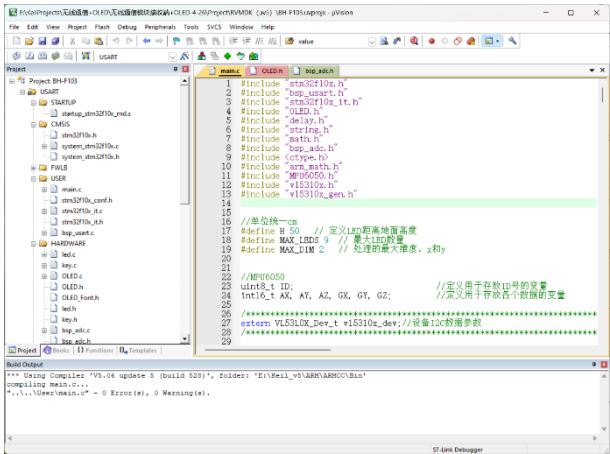


图5-2 编译工程文件

构建工程：在 Keil μ Vision 5 中打开工程文件（.uvprojx），然后选择 Build 选项来构建工程。生成 .bin 文件。如图 5-3。

$$Mean\ Error = \frac{1}{n} \sum_{k=0}^n d_i$$

第一次测量数据结果如下表 5-1 所示。

表 5-1 第一次数据

真实坐标		估计坐标		距离误差
x	y	x	y	单位 cm
0	0	0.412	0.41	0.58
第一象限				
真实坐标		估计坐标		距离误差
4	4	3.25	3.25	1.06
8	8	6.24	6.24	2.49
12	12	7.068	7.06	6.98
16	16	8.548	8.54	10.54
20	20	8.228	8.221	16.65
第二象限				
真实坐标		估计坐标		距离误差
-8	8	-6.66	5.346	2.97
-12	12	-8.47	8.336	5.09
-16	16	-10.46	10.28	7.96
-20	20	-11.96	7.542	14.83
第三象限				
真实坐标		估计坐标		距离误差
-8	-8	-6.729	-5.42	2.88
-12	-12	-8.679	-5.96	6.89
-16	-16	-8.219	-4.96	13.51
-20	-20	-6.689	-3.34	21.32
第四象限				
真实坐标		估计坐标		距离误差
8	-8	7.45	-6.066	2.01
12	-12	8.31	-8.076	5.39
16	-16	5.35	-6.65	14.17
20	-20	4.28	-6.68	20.60

第二次测量数据结果如下表 5-2 所示。

表 5-2 第二次数据

真实坐标		估计坐标		距离误差
x	y	x	y	单位 cm
0	0	-0.39	0.83	0.92
第一象限				
真实坐标		估计坐标		距离误差
8	8	6.24	6.24	2.49
12	12	5.8	7.02	7.95
16	16	8.08	8.08	11.20
20	20	7.3	8.64	17.04
第二象限				
真实坐标		估计坐标		距离误差
-8	8	-6.66	5.34	2.98
-12	12	-9.33	7.47	5.26
-16	16	-10.78	9.06	8.68
-20	20	-11.96	7.54	14.83
第三象限				
真实坐标		估计坐标		距离误差
-8	-8	-6.72	-6.42	2.03
-12	-12	-8.67	-5.95	6.91
-16	-16	-7.7	-6.4	12.69
-20	-20	-4.76	-3.42	22.52
第四象限				
真实坐标		估计坐标		距离误差
8	-8	6.83	-5.57	2.70
12	-12	7.73	-7.15	6.46
16	-16	5.13	-7.29	13.93
20	-20	3.66	-5.95	21.55

第三次测量数据结果如表 5-3 所示。

表 5-3 第三次数据

真实坐标		估计坐标		距离误差
x	y	x	y	单位 cm
0	0	-0.39	0.83	0.92
第一象限				
真实坐标		估计坐标		距离误差
8	8	7.13	7.13	1.23
12	12	8.36	7.52	5.77
16	16	8.54	8.54	10.55
20	20	9.15	9.15	15.34
第二象限				
真实坐标		估计坐标		距离误差
-8	8	-6.669	5.34	2.97
-12	12	-9.83	8.47	4.14
-16	16	-10.29	9.55	8.61
-20	20	-11.45	8.05	14.69
第三象限				
真实坐标		估计坐标		距离误差
-8	-8	-7.69	-6.05	1.97
-12	-12	-8.37	-5.65	7.31
-16	-16	-6.81	-5.43	14.01
-20	-20	-5.52	-3.87	21.68
第四象限				
真实坐标		估计坐标		距离误差
8	-8	7.39	-6.47	1.65
12	-12	8.31	-8.07	5.39
16	-16	6.84	-8.04	12.14
20	-20	5.17	-6.17	20.28

5.3 问题发现与测试总结

因项目涉及广泛，在做的过程中出现了很多的问题，所有发现的主要问题如下：

- （1）OLED 屏幕显示混乱（2）矩阵取逆异常（3）TOF 激光模块与光敏电阻模块融合

时中断函数异常（4）接收端倾斜时对垂直高度的矫正。

对于第一个问题，经排查发现是对行和列的处理有问题。在 OLED 显示时设置行和列都从 1 开始，每个字符设置固定像素宽度。同时为确保矩阵数据不会溢出显示器的尺寸，控制显示尺寸长度最大为 16 像素，行数最多四行。通过以上修改成功解决问题。

对于第二个问题，经过反复测试模拟数据最终定位到问题为奇异矩阵。在矩阵乘法和逆矩阵计算中加入了错误检查，并采用高斯-约当消元法来计算矩阵的逆，使用增广矩阵和行简化技术，确保了计算的精确性和稳定性，成功解决问题。

对于第三个问题，经排查发现是两个模块都使用的 IIC 通信协议导致冲突，将 TOF200C 激光模块改成软件实现 IIC 通信后即可解决问题。

对于第四个问题，在融合 MPU6050 和 TOF200C 激光模块实现高度纠正时，需要分情况考虑，如果只考虑一个角度的倾斜，从 MPU6050 获取加速度（AX，AY，AZ）和角速度（GX，GY，GZ）从 ToF 传感器获取距离测量。通过以下公式即可矫正高度。

$$\theta = \arctan\left(\frac{AX}{\sqrt{AY^2 + AZ^2}}\right)$$

$$\text{垂直高度} = \text{ToF 测量的距离} \times \cos(\theta)$$

当物体沿着多个轴倾斜时，需要使用三维空间中的向量来计算物体相对于地面的真实倾斜角度。原理上使用 MPU6050 的三轴加速度数据（AX，AY，AZ）来确定物体的姿态，并据此校正 ToF 激光传感器的距离读数。但在实际操作中，需要进行正确的加速度数据滤波和姿态解算算法的选择，以确保准确地计算出物体的倾斜角度。这涉及到对姿态解算算法的深入理解和调试，以及对硬件和软件方面的优化。由于个人水平有限，一时未能解决该问题，后续将继续深入研究，克服这一难关。

第六章 结束语

通过一个学期的不懈努力和持续探索，我终于圆满地完成了本次毕业设计课题。经过多次试验与调整，我成功开发出了一套可见光定位系统，其定位精度超出预期，能够非常有效地满足用户对室内定位的需求。

6.1 毕业设计的难点与创新

本设计的最大难点在于其需要广泛的知识面和技能，因为它涉及到了编程、电子工程、机械设计等多个领域。从 C 语言编程，到复杂的电路设计与电焊，再到使用 STM32 微控制器进行设备控制，每一步都要求具备深入的专业知识和实际操作能力。此外，项目还包括了精确的建模和 3D 打印技术，这些都是完成设计所必须的关键技术。本设计的特色主要有以下几点：

1. 在传统三边定位算法的基础上，拓展为多边定位算法，并增加了算法的健壮性，在实地实验中，出现不足 3 个 LED 情况，做了优化，采用备用算法，取获取到的 LED 坐标平均值来作为测量坐标。在大于 4 个 LED 的情况，取光强最大的 4 个 LED，进一步提升算法的准确性。对于奇异矩阵，采用矩阵奇异值分解，使得算法能够适用于更普遍的情况。

2. 在计算接收端和发射端距离方面，从用朗伯模型，改成用 TOF200C 激光测距仪和光敏电阻模块，两者结合实地测量光强分布情况，简化了建模过程，数据获取更加建议可靠。在获取测量的数据后，采用自动化数据处理，节省人工处理校对时间，提高了系统效率。

3. 在系统使用方面，采用实时更新的策略，每收到发送端信号后就会更新一次坐标，避免了死等坐标信号，坐标由常见的二维坐标，通过 TOF 测距仪和 MPU6050 的多传感器融合，提升为三维坐标。

4. 在信号交互方面，接收端和发射端采用相互确认的信息传输方式，发射端发送完信号后，接收端在接收后，立即返回接收到的数据，发射端确认返回的信息无误后，开始发起下一轮信息，如果长时间获取不到正确的信息，将自动跳到下一轮。

6.2 毕业设计的收获

这段时间的工程开发中，我对基于 RSS 的可见光定位系统的理解和认识有了显著的提升。通过不断的实验和调整，我逐渐掌握了系统的核心技术和关键参数。这一过程中，我遇到了许多挑战，如信号干扰问题、环境光的影响以及硬件的限制，但这些困难最终都被逐一克服，极大地提高了我的解决问题的能力 and 创新思维。此外，团队的合作也是我收获颇丰的一部分，通过与同学们的共同努力和知识交流，我们优化了系统设计，提高了定位的精确性和稳定性。这一切经历都极大地丰富了我的专业技能，也为我的学生生涯增添了宝贵的经验。

参考文献

- [01]. Ibrahim M. Abou-Shehada, Abdullah F. AlMuallim, et al. Accurate Indoor Visible Light Positioning Using a Modified Pathloss Model With Sparse Fingerprints [J]. Lightwave Technol, 2021, 39:6487-6497 .
- [02]. Huang Y, Pan YH, Yang R, Bao LH, Meng L, Luo HL, Cai YQ, Liu GD, Zhao WJ, Zhou Z, Wu LM, Universal mechanical exfoliation of large-area 2D crystals[J]. Nat Commun. 2020, 11(1):2453.
- [03]. 郭涛. 室内不规则场景 VLC 信道建模与定位的研究[D]. 内蒙古科技大学, 2022.
- [04]. 温倩, 张谦述, 张冲. 基于 LED 室内可见光的边界定位精度研究[J]. 激光杂志, 2023, 44(07):205-211.
- [05]. 孟祥艳, 张欣, 张峰等. 融合迁移特征学习的生物启发网络用于可见光成像室内位置感知方法[J]. 中国激光, 2023, 50(10):176-183.
- [06]. 王加安, 顾偲雯, 张思琦. 基于改进粒子群算法的室内可见光定位研究[J]. 半导体光电, 2023, 44(05):729-735.
- [07]. 朱亚丽. 基于深度神经网络的可见光通信室内定位[J]. 光学技术, 2023, 49(04):452-458.
- [08]. 陈俊波. 基于朗伯模型优化的可见光室内定位神经网络算法研究[D]. 西安工业大学, 2023.
- [09]. B. Hussain, Y. Wang, R. Chen, H. C. Cheng and C. P. Yue. LiDR: Visible-Light-Communication-Assisted Dead Reckoning for Accurate Indoor Localization[J]. IEEE Internet of Things Journal, 2022, 9(17):15742-15755.
- [10]. 徐心怡. 双灯可见光定位的时频结合实现方案[D]. 南京邮电大学, 2023.
- [11]. 王玲燕. BES-RBF 神经网络支持下的可见光定位系统[J]. 测绘通报, 2023, (10):91-97.

在校期间发表的研究成果及获奖

1 攻读学士学位期间的研究成果

1.1 公开发明专利

[01]. 张乐泉, 蔡玮, 马国阳等. 单关节仿生水下可见光通信无人潜航器、系统及控制方法, CN202210480568.2[P], 2022.

1.2 负责或参与课题

[01]. 马国阳, 卢凯昂, 陆九州等. 基于 Android 的虚拟宠物养成, TB202207044, 南京工程学院本科生科技创新训练项目, 2022.

[02]. 马国阳, 卢凯昂, 竹心雨等. 基于 RSS 的可见光定位关键技术研究, TB202307008, 南京工程学院本科生科技创新训练项目, 2023.

[03]. 马国阳, 张心怡, 尹力等. 基于 RSS 的可见光定位关键技术研究, 202311276022Z, 江苏省大学生创新创业训练计划项目, 2024.

2 荣获奖励

[01]. 2023 年获得江苏省第十八届大学生课外学术科技作品竞赛一等奖

[02]. 2022 年获得第十二届“挑战杯”江苏省大学生创业计划竞赛省级银奖

[03]. 2023 年获得第十四届江苏省蓝桥杯三等奖

[04]. 2023 年获得南京工程学院第二十二届校内大学生课外学术科技作品竞赛特等奖

[05]. 2022 年获得南京工程学院“互联网+”大学生创新创业大赛竞赛 一等奖

[06]. 2023 年获得南京工程学院“互联网+”大学生创新创业大赛竞赛 二等奖

[07]. 2022 年获得南京工程院校成果展二等奖

[08]. 2022 年获得天印杯初赛二等奖

[09]. 2020/2021 学年第一学期, 获得院级三等奖学金

[10]. 2021/2022 学年第一学期, 获得校级二等奖学金

[11]. 2021/2022 学年第二学期, 获得校级一等奖学金

- [12]. 2021/2022 学年，获得校优秀学生干部
- [13]. 2022/2023 学年，获得国家励志奖学金
- [14]. 2022/2023 学年第一学期，获得校级二等奖学金
- [15]. 2022/2023 学年第二学期，获得院级三等奖学金
- [16]. 2024 学年获得校级二等奖学金
- [17]. 2024 学年获校优秀学生干部

致 谢

在我的毕业设计即将画上圆满句点之际，我感到有必要并且非常想要真诚地表达我的感激之情给我的指导老师——蔡玮老师。从项目一开始的构思规划，到研究的深入，再到论文的撰写与最终的完成，蔡老师的细致指导和不懈支持一直是我研究旅程中的坚强后盾。在面对学术和技术的挑战时，蔡老师总能提供精准的指导和宝贵的建议，帮助我克服一个又一个难关。此外，蔡老师在精神上对我也有极大的鼓励，总是能在我感到迷茫或疲惫时，给予我力量和动力继续前行。这一路上获得的每一个成就，都有蔡老师的努力和付出在其中。我深感荣幸能在这样一位杰出导师的指导下完成我的学业，蔡老师的教诲和榜样将永远铭记在我心。

在本次毕业设计过程中，我也得到了很多同学的帮助，他们的支持对我完成设计起到了关键的作用。特别要感谢朱泰亨同学，他在我遇到编程难题时帮我调试代码，耐心地解决了许多技术问题。徐子安同学对我的电路设计提供了巨大帮助，他的专业知识和创造性思维使我们能够设计出高效稳定的电路。竹心雨同学在实验测试的过程中给予我极大的帮助，她的细心和专注保证了实验数据的准确性。李凤婷同学则在模型制作方面展现了她的才能，她不仅帮我建立了精确的 3D 模型，还亲自操作 3D 打印机，确保最终产品的质量。张心怡同学认真帮我整理好文档，为专利和论文付出大量时间和心血。张乐泉学长帮我分析问题原因，主动问我需要哪些帮助。每一位同学的贡献都是不可或缺的，我对他们的帮助和友情感激不尽。

致谢的最后，感谢我的舍友李智、刘文豪、刘一辉、陆九州、卢凯昂，陪我一起度过了四年美好的大学生活。长路漫漫，砥砺前行！