创新实验课

课题总结报告

课题名称：基于树莓派的智能移动“充电桩”小车

课题名称（英文）：Intelligent mobile“charging station”car based On Raspberry Pi

课题组长：闫康 2022210566

课题组成员：苏珈磊 2022210560 孙常鑫 2022210581 薛皓林 2022210571

报告时间： 2024 年 10 月 7 日

目录

1. 课题研究的选题背景和意义

1.1 选题背景

一、中央和政府方面

(1)项目归属:

本项目植根于无人驾驶这一充满未来感的领域，无人驾驶技术正如同一颗璀璨的星辰，逐渐在科技的夜空中绽放其夺目的光芒。随着技术的不断进步和创新，无人驾驶行业正受到前所未有的关注和重视。它不仅代表着智能交通的未来，更是智慧城市建设的重要组成部分，其影响力正逐步渗透到社会的每一个角落。

无人驾驶技术的发展，如同一场革命，正在重新定义我们对出行方式的认知。它不仅能够提高道路安全性，减少交通事故，还能提升交通效率，缓解城市拥堵。此外，无人驾驶技术的应用，还将为物流、配送、公共交通等领域带来颠覆性的变革，极大地提高服务效率和质量。

随着政策的扶持和市场的推动，无人驾驶行业正迎来一个全新的黄金发展期。从技术研发到商业应用，从法规制定到标准建立，无人驾驶正在逐步走向成熟。未来，无人驾驶技术有望成为推动社会进步的重要力量，引领我们走向一个更加智能、便捷、绿色的交通新时代。

(2)技术：无人驾驶技术逐渐成为中央和地方政府关注的焦点，各方纷纷出台了一系列精心设计的政策与措施，旨在为这一充满潜力的领域注入强劲动力，推动其迅猛发展。 从技术研发、人才培养到市场准入、安全监管，每一个环节都得到了细致的考量和周到的安排。政策的实施，不仅为无人驾驶技术的发展提供了坚实的基础，更为整个行业的发展指明了方向。

(3)行业：

中国无人驾驶行业正沐浴在政策的春风中，各级政府的重视如同阳光雨露，滋养着这一领域的茁壮成长。国家不仅将其上升至战略高度，更是以一系列政策措施，为无人驾驶技术的创新与应用铺就了金光大道。从中央到地方，政策的暖风频吹，不仅为无人驾驶技术的研发和测试提供了肥沃的土壤，更为其商业化、规模化发展注入了强劲动力。随着技术的不断突破和政策的持续优化，无人驾驶的商业化之路正逐渐从梦想照进现实，展现出无限广阔的发展前景。

在政策的推动下，无人驾驶技术的应用场景不断扩大，包括智能网联汽车“车路云一体化”应用试点、智能网联汽车准入和上路通行试点等 。同时，无人驾驶汽车的市场规模也在持续扩大，预计 2024 年全球无人驾驶汽车行业市场规模将达到 575.3 亿元，中国无人驾驶市场规模预计将达到 3832 亿元 。 随着技术的进步和政策的支持，无人驾驶汽车的商业化之路正逐渐从梦想照进现实，展现出无限广阔的发展前景。未来，无人驾驶技术有望为城市出行带来革命性变化，通过智能调度系统优化道路交通流量，提高道路使用效率，并可能对城市规划和基础设施进行调整 。此外，中国消费者对自动驾驶的接受度高，电动车的快速发

展和自动驾驶出租车的试运营有助于加深消费者对自动驾驶的认知 。在自动驾驶的产业链中，中国已初步形成了完整的本土产业链，包括核心算法、激光雷达、AI 芯片等领域均有具备全球竞争潜力的本土企业 。随着应用场景的不断丰富，中国的相关创业团队在全球竞争中有望获得更多的发展机会。

中国无人驾驶行业最新政策汇总一览

发布时间	政策名称	发布单位	主要内容
2024年1月	《关于开展智能网联汽车“车路云一体化”应用试点工作的通知》	公安部、自然资源部等五部门	鼓励试点城市内新销售具备L2级及以上自动驾驶功能的量产车辆搭载C-V2X车载终端，支持车载终端与城市级平台互联互通，选取部分公交线路（含BRT），实现全线交通设施联网识别和自动驾驶模式运行。
2023年12月	《自动驾驶汽车运输安全服务指南（试行）》	交通运输部	自动驾驶运输经营者应建立健全运输安全保障体系，在正式运营前要制定自动驾驶汽车运输安全保障方案，明确自动驾驶汽车的设计运行条件、人员配备情况、运营安全风险清单、分级管控措施、突发情况应对措施等。
2023年11月	《关于开展智能网联汽车准入和上路通行试点工作的通知》	公安部 、交通运输部等四部门	通过开展试点工作，引导智能网联汽车生产企业和使用主体加强能力建设，在保障安全的前提下，促进智能网联汽车产品的功能、性能提升和产业生态的迭代优化，推动智能网联汽车产业高质量发展。
2023年9月	《公路工程设施支持自动驾驶技术指南》	交通运输部	公路工程设施支持自动驾驶的建设应评估既有交通安全设施、服务设施、管理设施等的功能与性能，当符合本指南的技术规定时，应考虑融合利用。
2023年7月	《国家车联网产业标准体系建设指南（智能网联汽车）（2023版）》	工业和信息化部	到2025年，系统形成能够支撑组合驾驶辅助和自动驾驶通用功能的智能网联汽车标准体系。
2022年8月	《关于支持建设新一代人工智能示范应用场景的通知》	科技部	针对自动驾驶从特定道路向常规道路进一步拓展需求，运用车端与路端传感器融合的高准确环境感知与超视距信息共享、车路云一体化的协同决策与控制等关键技术，开展交叉路口、环岛、匝道等复杂行车条件下自动驾驶场景示范应用，推动高速公路无人物流、高级别自动驾驶汽车、智能网联公交车、自主代客泊车等场景发展。
2022年8月	《关于做好智能网联汽车高精度地图应用试点有关工作的通知》	自然资源部	在北京、上海、广州、深圳、杭州、重庆 6 个城市开展智能网联汽车高精度地图应用试点。
2022年2月	《关于试行汽车安全沙盒监管制度的通告》	市场监管总局、应急部等五部门	作为传统监管方式的有益补充，汽车安全沙盒监管变被动监管为主动监管，有利于更早地将前沿技术引发的质量安全问题纳入监管范围，提高应急处置能力，防范和化解重大风险，保护消费者合法权益，同时有利于鼓励企业技术创新，倡导最佳安全设计实践。
2022年1月	《“十四五”现代综合交通运输体系发展规划》	国务院	加强交通运输领域前瞻性、战略性技术研究储备，加强智能网联汽车、自动驾驶、车路协同、船舶自主航行、船岸协同等领域技术研发，开展高速磁悬浮技术研究论证。
2022年1月	《“十四五”数字经济发展规划》	国务院	推动智能计算中心有序发展，打造智能算力、通用算法和开发平台一体化的新型智能基础设施，面向政务服务、智慧城市、智能制造、自动驾驶、语言智能等重点新兴领域，提供体系化的人工智能服务。
2022年1月	《“十四五”现代流通体系建设规划》	国家发展改革委	加大北斗卫星导航系统推广，提高车路协同信息服务能力，探索发展自动驾驶货运服务。
2021年8月	《关于科技创新驱动加快建设交通强国的意见》	交通运输部、科技部	开发新一代智能交通系统，促进自动驾驶、智能航运等加快应用，突破综合交通网运营服务、危险货物管控等关键技术。
2021年7月	《智能网联汽车道路测试与示范应用管理规范（试行）》	交通运输部、工业和信息化部、公安部	推动汽车智能化、网联化技术应用和产业发展，规范智能网联汽车自动驾驶功能测试与示范应用。
2021年2月	《国家综合立体交通网规划纲要》	国务院	加强智能化载运工具和关键专用装备研发，推进智能网联汽车（智能汽车、自动驾驶、车路协同）、智能化通用航空器应用。
2021年1月	《交通运输部关于服务构建新发展格局的指导意见》	交通运输部	推进自动驾驶、智能航运、高速磁悬浮技术研发与试点示范工作，推进无人机基地智慧寄递网络、地下物流配送系统、交通运输天地一体化信息网、综合交通大数据中心、重点科研平台建设。

制图：中商情报网(www.askci.com)

图表 1 中国无人驾驶行业政策支持 截至 2024 年

二、高校方面

(1)高校教育领域正日益重视无人驾驶技术的研究与教学，聚焦于技术创新、

政策研究、产业趋势、人才培养、产学研合作、伦理社会影响以及国际比较研究，旨在通过跨学科合作和产学研结合，培养专业人才，推动无人驾驶技术的学术研究和实际应用，以应对未来交通领域的挑战和机遇。

(2) 截至 2024 年，中国有多所高校在无人驾驶领域开展了深入的研究和教学工作，以下是部分高校的例子：

高校名称	相关研究
清华大学	清华大学在无人驾驶领域的研究处于国内领先地位，拥有智能出行研究所等团队，专注于智能车辆理论与设计、智能驾驶感知决策与规划技术等方面的研究 。学校的车辆学院完成了国内首套全栈式端到端自动驾驶系统的开放道路测试，这项研究涵盖了从感知到控制的全链路环节，为高级别自动驾驶系统的落地应用奠定了基础 。此外，清华大学车辆与运载学院的教授们在智能网联汽车领域取得了显著成果，发表了多篇高水平论文，并获得了多项国家技术发明奖和科技进步奖 。学院还与国内外领军汽车企业和出行厂商开展商业化合作，推动无人驾驶技术的进一步发展 。
北京大学	北京大学智能车辆与移动机器人实验室在智能交通和网联技术领域取得显著成果，发表多篇论文并承担国家级项目。实验室由谢昆青教授领导，开展复杂时空建模、数据分析等研究，并与企业合作研究北斗车联网系统。同时，智能机器人与无人驾驶联合实验室聚焦汽车制造创新应用。教学上，开设无人驾驶相关课程，培养专业人才。
北京理工大学	北京理工大学的汽车研究所在智能车辆自动操控技术研究方面取得了显著的科研成果。该研究所是北京理工大学车辆传动国家重点实验室的重要组成部分，拥有一支由教授和副教授领导的团队，在智能交通与网联技术、智能车辆理论与设计、智能驾驶感知决策与规划技术、运动规划与控制技术、智能线控底盘技术等领域进行深入研究。研究所的团队成员在智能车辆领域成果丰富，曾研制中国第一辆无人驾驶车辆，并在 1999 年获得国家科技进步三等奖。近年来，围绕国家重大需求，在智能车辆自动操控技术领域取得了系列突破并得到系列应用，2019 年牵头获得国家科技进步二等奖，并获得了数十项部级科技奖励，为国防和国民经济发展作出了重要贡献。 此外，汽车研究所还拥有多个重要的科学研究平台，包括智能无人系统技术国家级重点实验室、无人车技术工业和信息化部重点实验室、教育部仿生机器人重点实验室、地面无人机动武器平台国防科技创新团队。 在教学方面，研究所开设了“无人驾驶车辆理论与设计”、“智能车辆理论与应用”等课程，并建设了相关的慕课平台，供学生和专业人士学习，旨在培

	养学生在智能汽车领域的专业知识和实践能力。
北京航空航天大学  	北京航空航天大学交通科学与工程学院在智能车辆与多智能体协同控制领域具有显著研究实力。学院拥有强大的科研团队，专注于智能交通系统、车联网信息安全等关键技术，取得了多项研究成果。设有智能交通研究中心和车路一体智能交通全国重点实验室，与行业龙头企业合作，推动智能交通技术发展。
上海交通大学  	上海交通大学的智能汽车研究所和智能车实验室在智能驾驶电动汽车的可靠性与环境适应性方面取得了显著成就。该研究所依托控制科学与工程和机械工程两个国家一级重点学科，以及汽车电子控制技术国家工程实验室等，开展与智能车辆相关的定位、导航、感知、控制等理论研究和应用研究 。智能车实验室在智能驾驶电动汽车领域取得了一系列成果，包括在动态干扰环境下园区无人驾驶可靠定位和场景理解方法的研究与验证、面向室内自主代客泊车的场端环境感知方法研究、基于虚实迁移的智能车辆端对端学习方法研究等 。 此外，上海交通大学还成立了智能网联电动汽车创新中心，该中心主动对接国家在智能网联汽车领域的重大需求，聚焦汽车“新四化”进程，即智能化、电动化、网联化、共享化，致力于构建智能网联汽车研究领域具有国际“领跑者”地位的学术高地 。在智能网联汽车与智能运维、氢燃料电池与系统、动力锂电池系统与应用、低碳车用动力系统等方向，上海交通大学也取得了显著成果，与行业内重点企业组建了创新技术研究联合体，推动了智能网联汽车全产业链的发展 。
北京邮电大学  	北京邮电大学在无人驾驶领域的研究涵盖了智能车辆的自动操控技术、车路云一体化、三维视觉智能感知等多个方面。朱孔林副教授在“车路云一体化”的无人驾驶方面进行了深入研究，探讨了无人驾驶的背景、原理、应用和发展趋势。徐士彪教授则专注于无人驾驶系统的三维视觉智能感知，包括三维场景地图重建、空间视觉定位及语义目标识别等。 此外，北京邮电大学还开展了智能网联电动汽车创新中心的研究工作，致力于构建智能网联汽车研究领域的学术高地，推动汽车“新四化”进程，即智能化、电动化、网联化、共享化。 在智能车辆与多智能体协同控制领域，北京邮电大学的研究团队取得了一系列成果，包括在动态干扰环境下园区无人驾驶可靠定位和场景理解方法的研究与验证、面向室内自主代客泊车的场端环境感知方法研究、基于虚实迁移的智能车辆端对端学习方法研究等。 北京邮电大学的研究团队还参与了多项国家级科研项目，如国家重点研发计划、国家自然科学基金等，并在国际高水平期刊和学术会议上发表了多篇论文，为无人驾驶技术的发展做出了重要贡献。

图表 2 中国高校在无人驾驶领域开展的研究和教学工作高校示例

1.2 选题意义

先来介绍选题调研:我们利用谷歌浏览器(Chrome)搜索根据我们的项目提出的有关问题,“人们对于手机电量下降存在很大程度上的焦虑?”以及利用了语言大模型进行调研,我们选取的大模型是 Kimichat 大模型,并且通过大模型可以返回得到相关的网站并进行网站内容的相关总结。同时也参考了同学们的意见,我们也通过微信 QQ 等聊天程序或者亲身询问同学们的意见,我们也选取了两个具有代表性的意见,具体的调研结果我们放在了立项报告的第一大点项目背景及意义当中。

再来介绍功能性的意义:

① 公共场所的充电宝服务

在商场或者公园等公共场所,智能小车可以提供移动充电宝服务,方便用户及时地为自己的电子设备进行充电,并且我们的小车具有定位服务,在商场由于人流量巨大,商铺排位复杂,公园中树木数量多,遮挡部分路线,导致固定的充电桩难以寻找,小车的定位服务便可以有效解决这一问题。

图表 3 位于商场和公园中的固定充电桩(品牌:怪兽充电)

② 校园内的充电服务:

在大学校园内,智能小车可以在图书馆,食堂,教师,宿舍区,操场等地点为同学们提供移动充电服务,由于大学人员较为密集,并且充电的资源是有限的,“充电桩”小车的引入可以帮助学生解决忘记带充电宝,充电器或者找不到充电插座的问题。

③ 户外活动或者紧急情况:

在户外活动比如露营,徒步等场景中,小车可以提供紧急的电源供应,为参与者提供充电服务。在紧急情况下,比如自然灾害或者是突发情况,小车可以快速部署,为受影响的人们提供必要的技术支持。

④ 为无人驾驶的发展提供技术支持

无人驾驶技术作为当下的热点,对于抢占科技前沿技术阵地,践行科技强国的国家号召具有重要意义。

我们的智能导航小车依靠 GPS 获取实时的位置信息,根据 GPS 路径信息点

完成自主巡航，对于无人驾驶技术的发展有一定的参考价值。

下面介绍一些拓展的应用场景，小车项目在商业推广和广告领域，能够通过搭载广告屏幕或宣传材料，在提供充电服务的同时进行品牌宣传。在安全监控方面，利用 GPS 和监控摄像头，智能小车可以在特定区域进行巡逻，监控安全状况。在旅游景点，它不仅能为游客提供充电服务，还能通过内置导航和信息屏提供旅游信息，提升游客体验。在企业园区和科技园区，智能小车作为移动充电站，可以提高员工的工作效率和便利性。而在住宅小区，它为居民提供便捷的充电服务，尤其是在休闲和散步时，增加了居住的舒适度和便利性。这些应用场景展示了智能小车在提供基本充电服务的同时，还能根据不同环境和需求提供额外的便利和安全保障。

2. 课题研究的主要内容与特色

2.1 选题研究的主要内容

在本次研究中，我们主要探讨了利用树莓派 4B 作为核心控制单元，结合 Donkey Car 框架，通过微信小程序实现用户交互，以及利用 OneNET 平台进行数据传输的智能送宝小车项目。研究重点包括 GPS 导航与定位技术的细节问题，并针对如何设计一个基于 GPS 自主导航实现巡航的难点提出了解决方案。通过深入分析和实验验证，我们旨在为智能送宝小车提供一种高效、可靠的自主导航方法，以满足其在复杂室外环境中的定位和巡航需求。

我们的小车主要功能为导航，能够实现驶向用户并返回充电柱的功能。具体内容如下：

① 驶向用户

当用户打开微信小程序，系统将自动获取 Donkey Car 的当前状态。如果小车正在执行任务，用户将被告知需要进入等待队列；若小车处于待命状态，用户可以点击“叫车”按钮。此时，微信小程序会计算 Donkey Car 与用户之间的经纬度差异，生成一条导航路径，并通过 OneNET 平台将这条路径信息发送给 Donkey Car。接到叫车指令后，Donkey Car 将利用 GPS 定位获取实时的地理位置，依据路径跟随算法，自动导航至用户所在位置。

② 返回充电柱

通过叫车功能，智能导航小车到达用户后，用户可以取下充电宝。随后，用户可以点击“巡航”按钮，微信小程序会以起点和充电桩（预设固定点位）的经纬度为参数，调用驾车路径规划 API 生成导航路径信息，并通过 OneNET 将其传送至 Donkey Car。小车收到导航指令后，会依赖 GPS 获取实时位置经纬度，基于路径跟随算法，自主根据 GPS 路径信息返回充电桩。

主要研究问题：

在实现我们所需要的功能的过程中，需要解决的问题有：

1. GPS 定位问题。GPS 的精确度应尽量缩小，否则会对实验结果造成较大影响。经过我们的测试，当 GPS 的精确度大于 2mCEP（即当小车移动距离小于

2m 时，GPS 很难判断出小车的位置发生了移动）时，无论怎么优化算法，也没有拌饭实现一个较好的跟随效果。因此，我们需要考虑如何解决 GPS 精度不准确的问题，并尝试各种解决方法。为了解决这一问题，我们采用了 RTK 技术来减少大气误差对 GPS 定位的影响，并利用差分定位技术显著提高了定位的精确度。通过这一改进，我们的智能导航小车能够在微信小程序的辅助下，更准确地执行用户的叫车指令，并根据实时 GPS 路径信息，顺畅地导航至用户所在地点，进而为用户提供更加可靠的服务体验。

2. 如何使用微信小程序将不同的功能连接到一起，并解决使用过程中出现的问题。我们面对的挑战包括确保地图组件准确显示位置、实时展示小车状态信息、管理多用户等待队列以及计算小车与用户之间的距离和预计等待时间等。

3. 实现微信小程序与树莓派的通信。通过 OneNET 平台，我们构建了一个稳定的数据传输通道，使 Donkey Car 能够不间断地上传其 GPS 位置数据。同时，用户可以通过微信小程序实时接收这些数据，并激活小车的多种功能，如导航至用户位置、自动返回充电桩，以及多用户冲突问题等问题。

4. 如何确保微信小程序与 Donkey Car 之间的数据交互既实时又准确。考虑到自主导航系统对 GPS 数据的高精确度需求，以及这些数据随时间变化可能产生较大差异，如何设计一个低延迟且稳定的通信系统，以确保小车能够及时接收和处理来自微信小程序的用户指令和实时位置信息至关重要。

5. 如何设计一个高效的状态机，确保树莓派驱动的 Donkey Car 小车能够在包括用户交互在内的多种情境中平稳过渡。面对用户可能同时发送的多条无标识指令，状态机必须能够准确地解析和执行每一条指令，以实现小车在不同操作模式间的无缝切换。

2.2 选题研究特色

- 小车根据 GPS 导航路径信息移动，不依赖于其他的视觉算法。用户只需在微信小程序中选定起点和终点，小程序便会自动调用 API 来规划最优驾驶路线。规划好的路线信息随后被传输至树莓派。树莓派负责实时追踪 Donkey Car 的 GPS 坐标，并依据路线信息智能地控制小车的行驶方向。这种导航方式赋予了小车强大的适应力，使其能够在未曾涉足的道路上也能自如地完成导航任务，确保了高度的通用性和灵活性。
- 将自动驾驶技术应用于北邮操场等日常环境，结合近年来的技术进步，可以为高校带来实际应用价值，实现智能导航小车在校园中的高效运作。
- 通过微信小程序轻松控制智能导航小车，无需下载任何额外应用，降低了使用门槛，简化了操作流程。小程序提供实时地图显示功能，帮助用户清晰地了解自己的当前位置，提升了用户体验。

3. 课题的设计思路、总体架构

3.1 设计思路

总体分为两部分，一部分为硬件，即 Donkey Car 主体加外围电路，另一部分为软件，即微信小程序，硬件与软件之间通过 OneNET 平台传输数据实现通信。

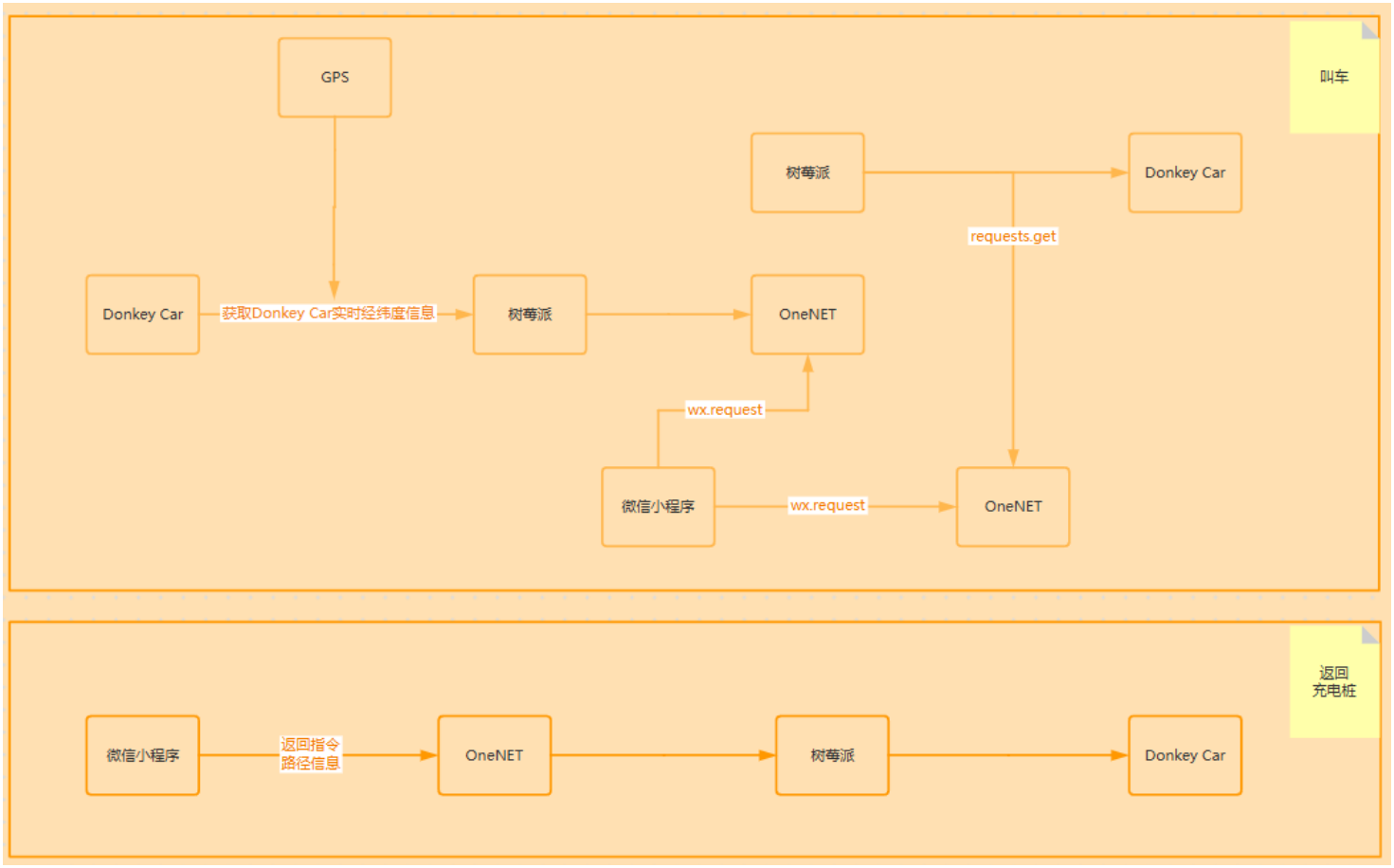
硬件配置方面，我们选择了 Donkey Car 小车作为移动平台，并以树莓派作为核心控制器。外围电路包括 PCA9685 和 L76X 两个关键组件。树莓派通过 I2C 协议与 PCA9685 通信，发送指令让 PCA9685 生成 PWM 信号来驱动小车前进。同时，树莓派通过串口与 L76X 通信，以获取并解析 GPS 数据，从而实时追踪小车的地理位置。此外，树莓派还利用 requests 库发起网络请求，与 OneNET 平台进行数据交换，实现远程控制和数据同步。硬件部分启动后第一步则将树莓派经纬度即小车的位置信息，发送至 OneNET 平台。

微信小程序在接收到 OneNET 平台提供的 Donkey Car 的经纬度信息后，结合用户授权的位置数据，调用驾车路径规划 API 来计算出一条从当前小车位置到用户位置的路线。计算出的路径信息随后被发送到树莓派，树莓派根据这些信息控制小车驶向用户。当小车到达用户位置，用户取下充电宝并点击“巡航”按钮后，微信小程序会再次调用 API 生成一条从当前位置到充电桩的路线，并将该路线信息发送至 OneNET 平台。Donkey Car 接收到新的路径信息后，便能引导用户前往充电桩。

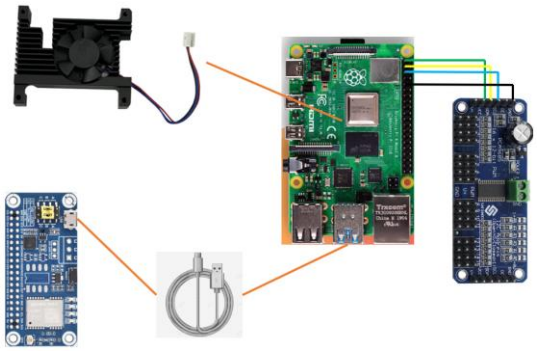
3.2 总体架构

总体架构图如图表 4 所示，总体接线电路图如图表 5 所示。

智能导航小车系统由微信小程序、Donkey Car 硬件以及数据交互平台 OneNET 三部分组成。用户可以通过微信小程序查看小车当前的状态信息。如果小车正在使用中，用户可以通过预约按钮加入等待队列，小程序将实时更新小车的状态；若小车空闲，用户可以通过点击“叫车”按钮召唤小车。一旦 Donkey Car 启动，它会利用 GPS 模块和 RTK 技术提升位置数据的精确度，并将位置信息实时上传至 OneNET 平台，微信小程序则通过网络请求获取这些经纬度信息。用户呼叫小车时，小程序会利用小车和用户的位置信息调用驾车规划 API 生成路线，并将路线信息发送给小车，使其前往用户所在地。用户点击“巡航”按钮后，小程序会根据小车的当前位置和充电桩的位置信息再次调用 API 生成返回路线，引导小车自动返回充电桩。



图表 4 系统总体架构图



图表 5 系统总体接线电路图

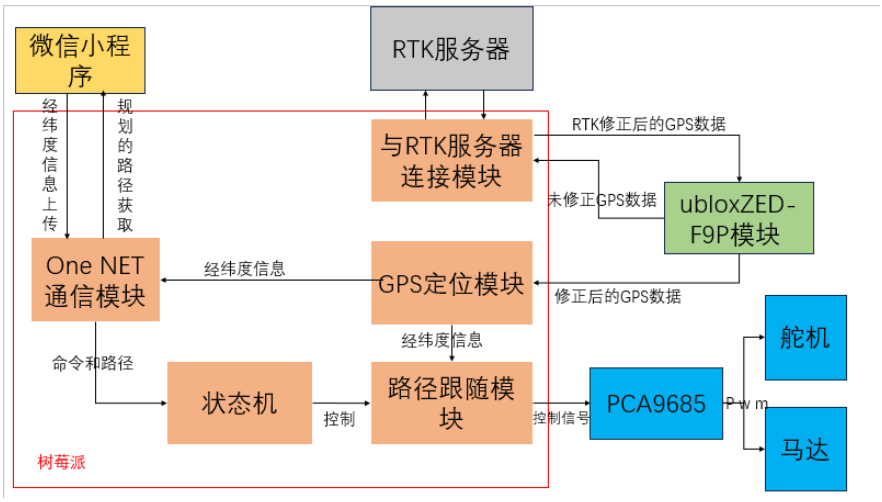
4. 各模块详细设计、实现及调试说明与分析（包括电路图和代码流程图）

4.1 Donkey Car

4.1.1 总体架构

总体架构见下图所示。Donkey Car 总体分为四个部分，分别为树莓派、 GPS 模块 **ublox ZED-F9P** 、PCA9685，RTK 服务器。其中树莓派作为主控，负责全

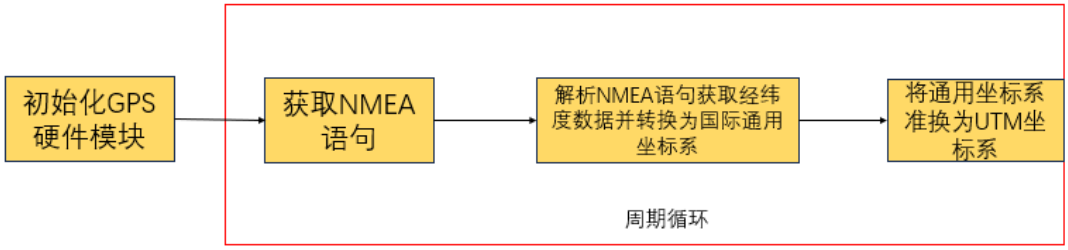
部工作的调度，包括与 OneNET 云服务器通信，GPS 定位、路径遵循，以及与 RTK 服务器连接等。



图表 6　Donkey Car 总体架构

4.1.2 GPS 定位模块

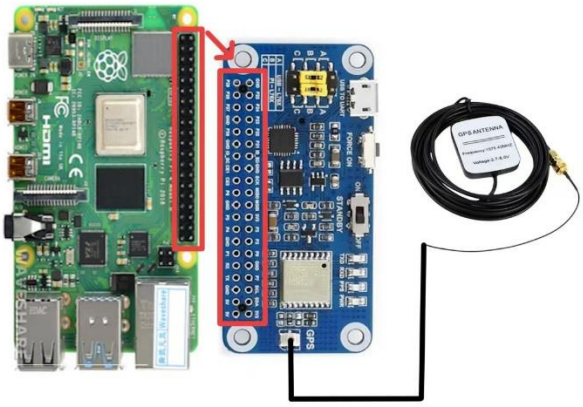
1. 总体流程



图表 7　GPS 定位总体流程

总体流程如图所示。首先对 GPS 的硬件进行初始化，例如波特率设置串口选择等，之后读取原始 nmea 语句并进行解析，得到一组数据，通过算法将这组数据映射为 UTM 坐标，此坐标即为我们想要得到的数据。

电路图如图表 8 所示。



图表 8　GPS 接线电路图

2. 代码实现

（1）首先初始化 GPS 硬件模块，设置波特率为 115200，由于 **ublox ZED-F9P** 启动时波特率默认为 115200（已经在连接 RTK 的文档处实现了相关配置，这个波特率可以自行更改），要以增加数据传输效率，减少延迟带来的误差。

```
def __init__(self, port:str='/dev/ttyACM0',
baudrate:int=115200, bits:int=8, parity:str='N',
stop_bits:int=1, charset:str='ascii', timeout:float=0.1):
    self.port = port
    self.baudrate = baudrate
    self.bits = bits
    self.parity = parity
    self.stop_bits = stop_bits
    self.charset = charset
    self.timeout = timeout
    self.ser = None
```

图表 9 GPS 初始化先关配置代码实现

（2）获取 NMEA 语句并保存为数组

```
def _readline(self) -> str:
    """
    Read a line from the serial port in a threadsafe manner
    returns line if read and None if no line was read
    """
    if self.lock.acquire(blocking=False):
        try:
            # TODO: Serial.in_waiting _always_ returns 0 in
            Macintosh
            if dk_platform.is_mac() or (self.serial.buffered()
            > 0):
                success, buffer = self.serial.readln()
                if success:
                    return buffer
        finally:
            self.lock.release()
    return None
```

图表 10 GPS 获取 NMEA 语句代码实现

（3）解析 NMEA 语句获取经纬度数据并转换为国际通用坐标系 UTM 坐标

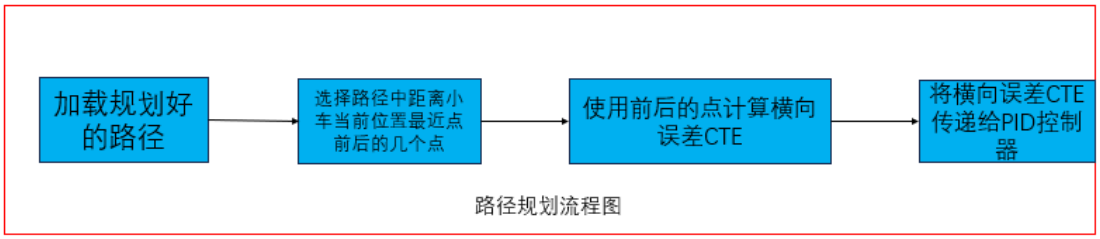
```
if nmea_parts[2] == 'V':
    # V = Warning, most likely, there are no satellites in
    view...
    logger.info("GPS receiver warning; position not valid.
    Ignore invalid position.")
else:
    #
    # Convert the textual nmea position into degrees
    #
    longitude = nmea_to_degrees(nmea_parts[5], nmea_parts
    [6])
    latitude = nmea_to_degrees(nmea_parts[3], nmea_parts[4])
    if debug:
        if msg.longitude != longitude:
            print(f"Longitude mismatch {msg.longitude} !=
            {longitude}")
        if msg.latitude != latitude:
            print(f"Latitude mismatch {msg.latitude} !=
            {latitude}")
```

```
def nmea_to_degrees(gps_str, direction):
    if not gps_str or gps_str == "0":
        return 0
    parts = gps_str.split(".")
    degrees_str = parts[0][:-2]        # results in zero to 3 digits
    minutes_str = parts[0][-2:]        # always results in 2 digits
    if 2 == len(parts):
        minutes_str += "." + parts[1]  # combine whole and
    degrees = 0.0
    if len(degrees_str) > 0:
        degrees = float(degrees_str)
    minutes = 0.0
    if len(minutes_str) > 0:
        minutes = float(minutes_str) / 60
    return (degrees + minutes) * (-1 if direction in ['W', 'S']
    else 1)
```

图表 11　GPS 的 NMEA 语句转换为国际通用坐标系代码实现代码

4.1.3 路径遵循模块

1、 总体流程



图表 12 路径遵循流程图

如图表 12 所示，该模块保存了从 OneNET 获取的路径点，并获取 GPS 定位模块输出的 UTM 坐标，遍历所有路径点计算与当前坐标的距离，并从中找出与当前坐标最近的点。

（1）加载规划好的路径

```
if os.path.exists(cfg.PATH_FILENAME) and path.load(cfg.
PATH_FILENAME):
    print("The path was loaded was loaded from ", cfg.
    PATH_FILENAME)
```

```
def load(self, filename: str) -> bool:
    path = pathlib.Path(filename)
    if path.is_file():
        with open(filename, "r") as infile:
            self.path = []
            for line in infile:
                xy = [float(i.strip()) for i in line.strip().
                split(sep=",")]
                self.path.append((xy[0], xy[1]))
                self.throttles.append(xy[2])
        return True
    else:
        logging.warning(f"File '{filename}' does not exist")
        return False
```

图表 13 路径遵循代码加载路径实现

（2）寻找最近的点，并且从最近的点前后各取几个点

寻找最近的点：

```
def nearest_pt(self, path, x, y, from_pt=0, num_pts=None):
    from_pt = from_pt if from_pt is not None else 0
    num_pts = num_pts if num_pts is not None else len(path)
    num_pts = min(num_pts, len(path))
    if num_pts < 0:
        logging.error("num_pts must not be negative.")
        return None, None, None

    min_pt = None
    min_dist = None
    min_index = None


    if min_dist is None or d < min_dist:
        min_pt = p
        min_dist = d
        min_index = i
    return min_pt, min_index, min_dist
```

在最近前后各找几个点：

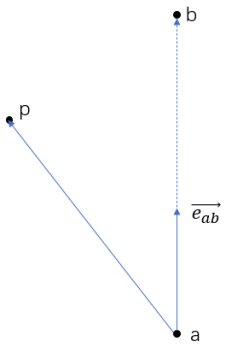
```
def nearest_waypoints(self, path, x, y, look_ahead=1,

    if path is None or len(path) < 2:
        logging.error("path is none; cannot calculate nearest points")
        return None, None

    if look_ahead < 0:
        logging.error("look_ahead must be a non-negative number")
        return None, None
    if look_behind < 0:
        logging.error("look_behind must be a non-negative number")
        return None, None
    if (look_ahead + look_behind) > len(path):
        logging.error("the path is not long enough to supply the waypoints")
        return None, None

    _pt, i, _distance = self.nearest_pt(path, x, y, from_pt, num_pts)

    # get  start of segment
    a = (i + len(path) - look_behind) % len(path)

    # get the end of the segment
    b = (i + look_ahead) % len(path)

    return a, i, b
```

图表 14 路径遵循代码实现

（3）通过这几个点计算横向误差 CTE

计算方法如下：

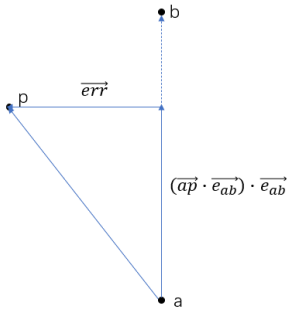
首先，创建了一个向量 ap，如图



图表 15 向量 ap

然后将向量 ap 与归一化向量 ab 点乘，得到向量 ap 在 ab 方向的投影大小，再将归一化向量 ab 乘上这个投影大小，并与向量 ap 相减得到从线段 ab 指向 p

点的向量 err，其模值即为点 p 到向量 ab 的距离，如图



图表 16 示意图

最后将向量 ab 与向量 err 叉积得出误差的方向，设置符号 sign 为正负 1，与向量 err 的模值相乘得出误差。

相关实现代码

```
cte = 0.
i = from_pt

a, b, i = self.nearest_track(path, x, y,
                             look_ahead=self.look_ahead, look_behind=self.look_behind,
                             from_pt=from_pt, num_pts=self.num_pts)

if a and b:
    logging.info(f"nearest: ({a[0]}, {a[1]}) to ({x}, {y})")
    a_v = Vec3(a[0], 0., a[1])
    b_v = Vec3(b[0], 0., b[1])
    p_v = Vec3(x, 0., y)
    line = Line3D(a_v, b_v)
    err = line.vector_to(p_v)
    sign = 1.0
    cp = line.dir.cross(err.normalized())
    if cp.y > 0.0 :
        sign = -1.0
    cte = err.mag() * sign
else:
    logging.info(f"no nearest point to ({x},{y}))")
return cte, i
```

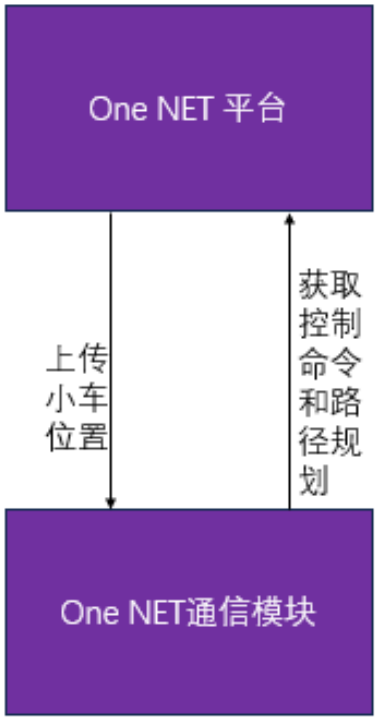
图表 17 路径遵循代码实现

（5）最后 cte 传入 PID 控制器，通过 PID 控制器输出转向值

```
class PID_Pilot(object):

    def __init__(
            self,
            pid: PIDController,
            throttle: float,
            use_constant_throttle: bool = False,
            min_throttle: float = None) -> None:
        self.pid = pid
        self.throttle = throttle
        self.use_constant_throttle = use_constant_throttle
        self.variable_speed_multiplier = 1.0
        self.min_throttle = min_throttle if min_throttle is not None else throttle

    def run(self, cte: float, throttles: list, closest_pt_idx: int) -> tuple:
        steer = self.pid.r
        un(cte)
        if self.use_constant_throttle or throttles is None or closest_pt_idx is None:
            throttle = self.throttle
        elif throttles[closest_pt_idx] * self.variable_speed_multiplier < self.min_throttle:
            throttle = self.min_throttle
        else:
            throttle = throttles[closest_pt_idx] * self.variable_speed_multiplier
        logging.info(f"CTE: {cte} steer: {steer} throttle: {throttle}")
        return steer, throttle
```

图表 2 代码实现

4.1.4 OneNET 通信模块



图表 3　One NET 模块

（1）如图 20 所示小车上传自身位置 到 One NET 平台

```
def upload_gps_to_onenet(self):

    if self.gps_latest_position is not None:
      try:
            # 获取最新的GPS位置
            latest_position = self.gps_latest_position.get_latest_position()
              print("chenggong45646")
            if latest_position is not None:
                # 假设 latest_position 是一个元组 (easting, northing)
                easting, northing = latest_position[1:]
                  print([easting,northing])
                # 构建数据
                values = {
                    'datastreams': [
                        {
                            'id': 'GPS',
                            'datapoints': [
                                {
                                    'value': {
                                        'latitude': easting,  # 这里假设 easting 代表纬度信息
                                        'longitude': northing  # 这里假设 northing 代表经度信息
                                    }
                                }
                            ]
                        }

            # 上传数据
            jdata=json.dumps(values).encode("utf-8")
            response = requests.post(self.upload_url, headers=self.headers, data=jdata)
            print("Upload response:",response.content)
        else:
            print("No GPS position available.")
    except requests.RequestException as e:
        print(f"上传出错: {e}")
```

图表 4 代码实现

（2）小车从 OneNET 获取小程序规划好的路径点位和是否切换路径的命令。

```
def poll(self):
    try:
        response = self.session.get(self.download_url, headers=self.headers)
        response.raise_for_status()
        params = response.json()
          print(params)
        a=params['data']
          print(params)

        if 'error' not in a:
            for datastream in a:
                if datastream['id'] == 'Location':
                    self.xunhangline_data=datastream['current_value']['road_message']
                    self.signal = datastream['current_value']['signal']  # 获取最新的信号值
                elif datastream['id'] == 'stop':
                    self.stop_signal =datastream['current_value']['road_message'][0]['signal']
                elif datastream['id'] == 'xunhangline':
                    self.chargeline_data=datastream['current_value']['road_message']

            # 检查经纬度数组是否同步更新，然后转换为UTM坐标
            self.coordinates = [(float(coord['longitude']), float(coord['latitude'])) for coord in self.
            xunhangline_data]
              print(self.coordinates)
```

图表 21 代码实现

（3）将获取的路径点写入路径文件中去

```
def write_to_csv(self,filename,data,throttle_value):
    with open(filename,'w',newline='') as csvfile:
        writer=csv.writer(csvfile)
        print("write success")
        for coord in data:
            formatted_throttle = f"{throttle_value:.10e}"
            writer.writerow([coord[0],coord[1],formatted_throttle])
```

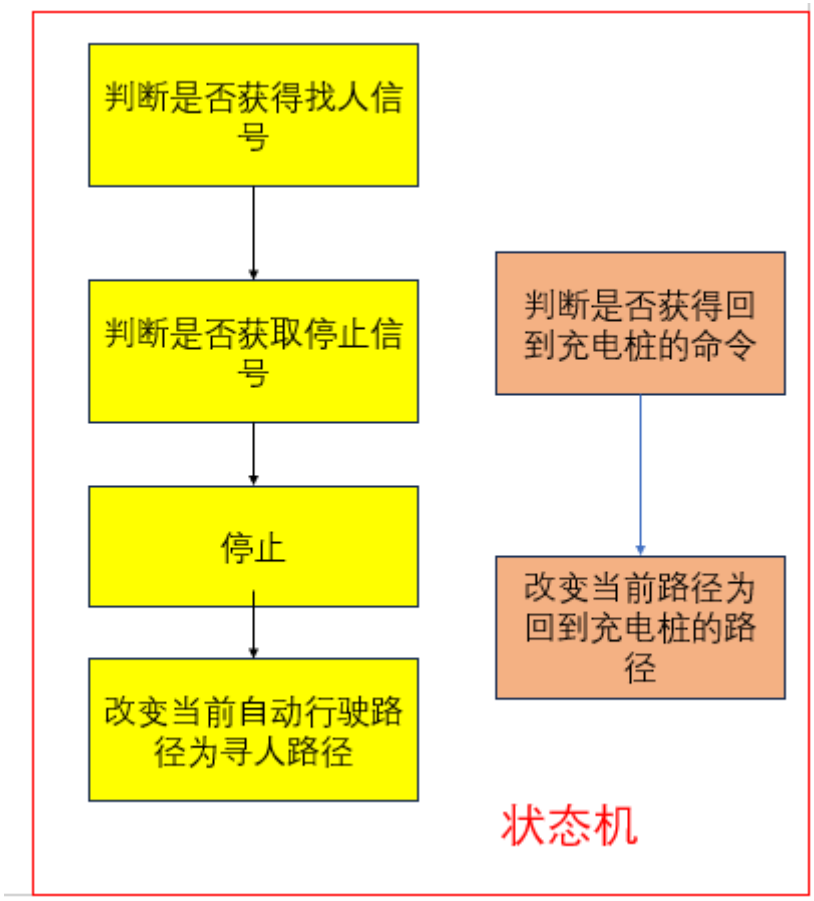
图表 22 代码实现

（4）上传与获取数据的函数都写在了循环里面，为了持续调用。

```
def update(self):
    while self.running:
        self.poll()
        self.upload_gps_to_onenet()
        time.sleep(self.poll_delay)
```

图表 23 代码实现

4.1.5 状态机



图表 24 状态机流程图

如图表 所示，状态机控制小车行驶的开始和结束。主要是通过获取 One NET 上面的命令来进行切换路径，从而使得小车可以变换不同的路径。

（1）小车是否寻找小程序用户的行为判断，如果是，就切换当前路径为找人路径’onenet.csv’

```
if self.signal == 1 and self.last_signal == 0:

        self.write_to_csv('onenet.csv',self.coordinates,self.throttle_value)
        self.ctr.button_down_trigger_map['start']()
        self.path.load('onenet.csv')
        self.ctr.button_down_trigger_map['start']()
        print("load onenet.csv")
        self.last_signal = self.signal
```

图表 25

（2）小车是否停止的行为判断：

```
if self.stop_signal ==0 and self.stop_last_signal ==1:

        self.ctr.button_down_trigger_map['start']()
        self.ctr.emergency_stop()
        self.stop_last_signal = self.stop_signal
        print("E-stop")
        self.charge_last_signal=0
```

图表 26

（3）小车是否寻找小程序充电桩的行为判断，如果是，就切换当前路径为找人路径’chongdainzhuang.csv’

```
if self.stop_signal ==1 and self.stop_last_signal ==0:

        self.write_to_csv('chongdianzhuang.csv',self.chargeline_data_utm,self.throttle_value)

        self.path.load('chongdianzhuang.csv')
        print("chongdianzhuang.csv")
        self.ctr.button_down_trigger_map['start']()

        self.stop_last_signal = self.stop_signal
```

图表 27

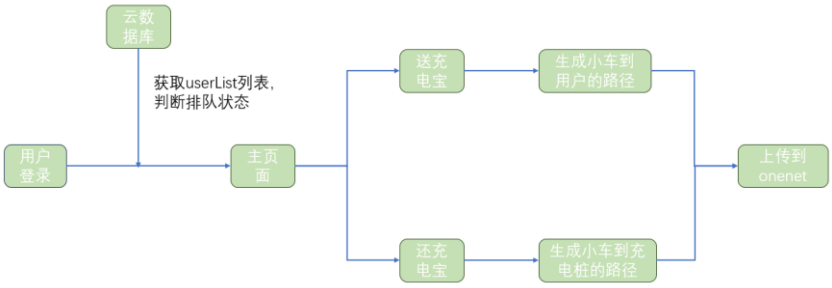
4.1.6 树莓派与 RTK 服务器数据交互

这一部分另附 pdf 文件《GPS 连接 RTK 说明》

4.2 微信小程序

4.2.1 原理

微信小程序架构图如图表 28 所示。微信小程序模块主要包括“叫车”和“换车”两个部分。



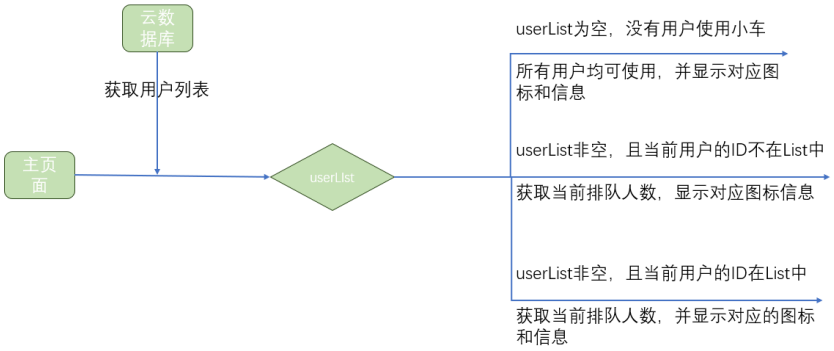
图表 28 微信小程序架构图

页面展示部分可以展开如图表 30。

主页面可以分为两种状态，分别展示不同的图案和文字，来提示用户小车当前的使用状态。

首先，当用户进入微信小程序后，小程序会直接出发 onLoad 生命周期函数，在生命周期函数中，我们直接调用云函数获取当前用户的_openid，并访问数据库获取当前小车的用户列表 userList。如果用户列表 userList 为空，则当前小车为空闲状态，用户可以直接叫车，小程序页面会展示小车到用户的距离和预计到达的时间，同时会展示对应的图标；如果用户列表 userList 非空且用户的_openid 不在用户列表 userList 中，说明此时小车正送宝的过程中，而当前用户没有预约小车，这时会直接获取用户列表 userList 的长度，并在主页显示预约按钮，当前排

队人数以及对应的图标；如果用户列表 userList 非空，用户的_openid 在用户列表 userList 中，即用户预约过小车，这时主页面会显示小车预约排队人数和取消预约按钮，并显示对应的图标。图 13 为上述逻辑的流程图，图 14 为小车不同状态的页面展示



图表 29 微信小程序页面展示部分

图表 30 小车不同状态的不同主页面

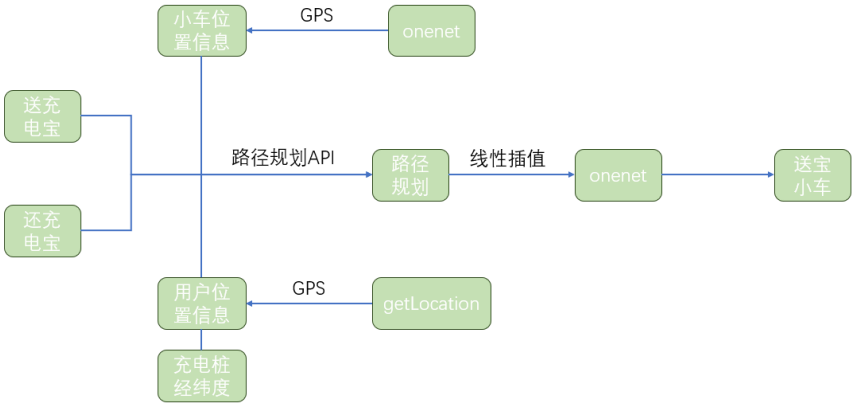
送还工作流程图如图标 32 所示。

送充电宝流程：用户进入微信小程序，系统会直接获取用户当前的位置信息。Donkey Car 通过 GPS 获取小车的经纬度信息后实时上传到 OneNET 平台，同时系统会以 1 秒的频率从 OneNET 平台实时获取小车的位置信息 ，同时刷新地图中的小车的坐标。用户点击“送还”按钮后，微信小程序将跳转到子页面， 用户在子页面点击出发按钮后，系统将会以 Donkey Car 的经纬度为起点，用户的经纬度为终点，调用线性插值算法生成每半米一个坐标点的路径信息，并将其和一个 bool 信号一起到 OneNET 平台，同时在地图上画出小车到用户的路径。小车会实时读取 OneNET 平台上的信息，当 bool 信号发生变化时，小车会加载小

图表 31 小程序子页面

车找用户的路径信息，从巡航状态改变为送宝状态，沿着生成路径到达用户的所在地。

还充电宝流程为：小车到达用户的身边后，用户按下“停止”按钮暂停小车，用户归还充电宝后，用户点击“巡航”按钮重新启动小车，小程序会获取当前小车所在位置的经纬度信息并把它作为起点，同时把实现确定好的充电桩的经纬度位置信息作为路径的终点，调用线性插值算法生成小车到充电桩的路径信息，并把生成好的路径信息上传到 OneNET 平台。小车在通过 OneNET 平台接受到启动信号后，会读取小车到充电桩的路径信息，并最终驶向充电桩所在的位置。



图表 32 微信小程序工作流程图

4.2.2 代码

一、页面展示

```
'/设定小车初始状态
:heckCarStatus(e) {
 wx.showLoading({
   title: '信息获取中…',
 })
 wx.cloud.database().collection('waitCarList').where({
    _openid: 'oJPfn4o841lCQyL4Ixd45YqqRDQI'
  })
  .get()
  .then(res => {
    this.data.userList = res.data[0].userList
    if (!this.data.userList.length) {
      this.setData({
        car_can_use: 1,
        identification: 0
      })
      wx.hideLoading()
    } else {
      const signal = this.data.userList.findIndex(e => e === this.data.openId)
      if (signal == -1) {
        this.setData({
          car_can_use: 0,
          identification: 1,
          needWaitPeople: this.data.userList.length
        })
        wx.hideLoading()
      } else if (signal == 0) {
        this.setData({
          car_can_use: 0,
          identification: 1
        })

      } else {
        var index = this.data.userList.findIndex(e => e === this.data.openId)
        this.setData({
          car_can_use: 0,
          identification: 2,
          needWaitPeople: index
        })
        wx.hideLoading()
      }
      wx.hideLoading()
    }
```

图表 33 页面展示相关代码

二、送充电宝

```
songbao(e){
  var _this = this
  var distance = Math.round(this.haversineDistance(parseFloat(this.data.car_lat_wgs), parseFloat(this.data.car_lon_wgs), parseFloat(this.data.
  you_lat_wgs), parseFloat(this.data.you_lon_wgs)))*2
  console.log('distance:',distance)
  // 生成路径经纬度信息点
  var route_list = this.interpolateLinear(parseFloat(this.data.car_lat_wgs), parseFloat(this.data.car_lon_wgs), parseFloat(this.data.you_lat_wgs
  , parseFloat(this.data.you_lon_wgs), distance)
  // console.log(route_list)
  // console.log(this.data.you_lat_wgs,this.data.you_lon_wgs)
  // console.log(this.data.car_lat_wgs,this.data.car_lon_wgs)
  //上传用的WGS坐标
  var pl_wgs = []
  // 将解压后的坐标放入点串数组pl中
```

```
//上传用的WGS坐标
var pl_wgs = []
// 将解压后的坐标放入点串数组pl中
for (var i = 0; i < route_list.length; i += 1) {
        pl_wgs.push({
          latitude: route_list[i][0],
          longitude: route_list[i][1]
        })
      }
      pl_wgs.push({
        latitude: parseFloat(this.data.you_lat_wgs),
        longitude: parseFloat(this.data.you_lon_wgs)
      })
      console.log('pl_wgs:',pl_wgs)
      this.data.datapoint_xunhang["datastreams"][0]["datapoints"][0]["value"]["road_message"] = pl_wgs
      this.data.datapoint_xunhang["datastreams"][0]["datapoints"][0]["value"]["signal"] = 1
```

```
      //画图用的火星坐标
        var pl = [
          {
            latitude: parseFloat(this.data.car_lat),
            longitude: parseFloat(this.data.car_lon)
          }
        ]
      // 将解压后的坐标放入点串数组pl中
      for (var i = 0; i < route_list.length; i += 1) {
        const baiduInstance = new mapTransfrom();
        baiduInstance.Google_Coordinates(route_list[i][0],route_list[i][1])
              pl.push({
                latitude: baiduInstance.Lat_Goodle,
                longitude: baiduInstance.Lon_Goodle
              })
            }
            pl.push({
              latitude: parseFloat(this.data.you_lat),
```

```
            longitude: parseFloat(this.data.you_lon)
          })
        // console.log('pl:',pl)
        console.log('小车已出发')
        this.uplonlaufrompi();
        this.setData({
          // 将路线的起点设置为地图中心点
          latitude: pl[0].latitude,
          longitude: pl[0].longitude,
          //  waitTime : res.data.result.routes[0].duration,
            // 绘制路线
            polyline: [{
              points: pl,
              color: '#54dde7',
              width: 10,
              borderColor: '#2f693c',
              arrowLine: true,
              borderWidth: 2
```

图表 54 送宝函数代码

三、还充电宝

```
//小车回充电桩
backtochongdianzxhuang(e){
  this.getcarstation()
  console.log(this.data.car_lat_wgs,this.data.car_lon_wgs)
  console.log(this.data.zhuang_lat_wgs,this.data.zhuang_lon_wgs)
  var distance = Math.round(this.haversineDistance(parseFloat(this.data.car_lat_wgs), parseFloat(this.data
  zhuang_lat_wgs), parseFloat(this.data.zhuang_lon_wgs)))*2
console.log('distance:',distance)
// 生成路径经纬度信息点
var route_list = this.interpolateLinear(parseFloat(this.data.car_lat_wgs), parseFloat(this.data.car_lon_wg
zhuang_lat_wgs), parseFloat(this.data.zhuang_lon_wgs), distance)
  // console.log(route_list)
  // console.log(this.data.zhuang_lat_wgs,this.data.zhuang_lon_wgs)
  // console.log(this.data.car_lat_wgs,this.data.car_lon_wgs)
  //上传用的WGS坐标
  console.log('小车位置: ',this.data.car_lat_wgs,this.data.car_lon_wgs)
```

```
var pl_wgs = [
{
  latitude: parseFloat(this.data.car_lat_wgs),
  longitude: parseFloat(this.data.car_lon_wgs)
}
]
// 将解压后的坐标放入点串数组pl中
for (var i = 0; i < route_list.length; i += 1) {
    pl_wgs.push({
      latitude: route_list[i][0],
      longitude: route_list[i][1]
    })
  }
  pl_wgs.push({
    latitude: parseFloat(this.data.zhuang_lat_wgs),
    longitude: parseFloat(this.data.zhuang_lon_wgs)
  })

    this.data.datapoint_xunhang["datastreams"][0]["datapoints"][0]["value"]["road_message"] = pl_wgs
    this.uploadxunhang()
    console.log('小车回到充电桩')
},
```

图表 35 导航相关代码

四、路径规划线性插值

```
// 线性插值函数
interpolateLinear(lat1, lon1, lat2, lon2, numPoints) {
  const points = [];
  // 在每个插值点之间进行线性插值计算
  for (let i = 1; i <= numPoints; i++) {
    const fraction = i / (numPoints + 1); // 插值比例

    // 线性插值计算纬度和经度
    const lat = lat1 + fraction * (lat2 - lat1);
    const lon = lon1 + fraction * (lon2 - lon1);

    points.push([lat, lon]); // 将插值点添加到结果数组中
  }
  return points;
},
```

```
// 计算两个经纬度点之间的距离（单位：米）
haversineDistance(lat1, lon1, lat2, lon2) {
  const R = 6371000; // 地球半径，单位为米
  const toRadians = (degree) => degree * Math.PI / 180;

  const dLat = toRadians(lat2 - lat1);
  const dLon = toRadians(lon2 - lon1);
  const lat1Rad = toRadians(lat1);
  const lat2Rad = toRadians(lat2);

  const a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
            Math.cos(lat1Rad) * Math.cos(lat2Rad) *
            Math.sin(dLon / 2) * Math.sin(dLon / 2);

  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  return R * c; // 返回两个坐标点之间的距离，单位为米
},
```

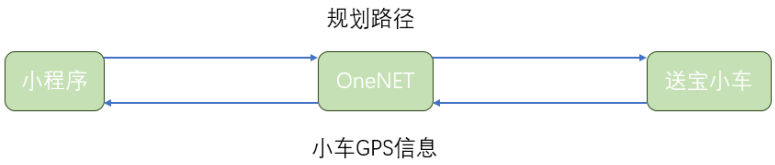
图表 36 路径规划代码

4.3 OneNET 模块

4.3.1 概述

当小车路径信息生成后，微信小程序通过 wx.request 发送网络请求，将路径信息发送至 OneNET，树莓派通过 requests.get 网络请求获取导航路径信息。

Donkey Car 获取自己当前的经纬度信息后，通过 requests.post 网络请求将自己的经纬度信息上传至 OneNET，微信小程序通过 wx.request 发起网络请求获取 Donkey Car 的实时经纬度信息。



图表 37 OneNET 架构图

4.3.2 代码

微信小程序获取 Donkey Car 的经纬度信息代码如图表 38 所示

```
//获取小车实时的经纬度信息
getcarstation(e) {
  var that = this
  wx.request({
    url: 'https://api.heclouds.com/devices/1100048518/datapoints',
    header: {
        "api-key": "d4Rs4uU=tusmD3Dmh3KI1OcPvzg=" //自己的api-key
    },
    success: res => {
      // console.log(res)
      const endInstance = new mapTransfrom();
        endInstance.Google_Coordinates(res.data.data.datastreams[2].datapoints[0].value.longitude,res.data.data.datastreams[2].datapoints[0].
        value.latitude)
      this.setData({
        car_lat_wgs: res.data.data.datastreams[2].datapoints[0].value.longitude,
        car_lon_wgs: res.data.data.datastreams[2].datapoints[0].value.latitude,
        car_lat: endInstance.Lat_Goodle,
        car_lon: endInstance.Lon_Goodle
```

图表 38 获取经纬度信息

微信小程序上传导航路径信息如图表 39 所示

```
//上传小车巡航线信息
uploadxunhang(e){
  wx.request({
    url: 'https://api.heclouds.com/devices/1100048518/datapoints',
    header: {
       "api-key": "d4Rs4uU=tusmD3Dmh3KI1OcPvzg=" //自己的api-key
    },
    method: 'POST',
    data: this.data.datapoint_xunhang,
    success: res => {
    },
    fail: err => {
    }
  })
},
//上传小车控制信息
uploadgo(e) {
  var pl = []
  // 将解压后的坐标放入点串数组pl中
   pl.push({
                signal: 1,
              })
 this.data.datapoint_stop["datastreams"][0]["datapoints"][0]["value"]["road_message"] = pl
  wx.request({
    url: 'https://api.heclouds.com/devices/1100048518/datapoints',
    header: {
      "api-key": "d4Rs4uU=tusmD3Dmh3KI1OcPvzg=" //自己的api-key
    },
    method: 'POST',
    data: this.data.datapoint_stop,
    success: res => {
      wx.showToast({
        title: '小车开始巡航',
        icon: 'success'
```

图表 39 微信小程序上传路径信息

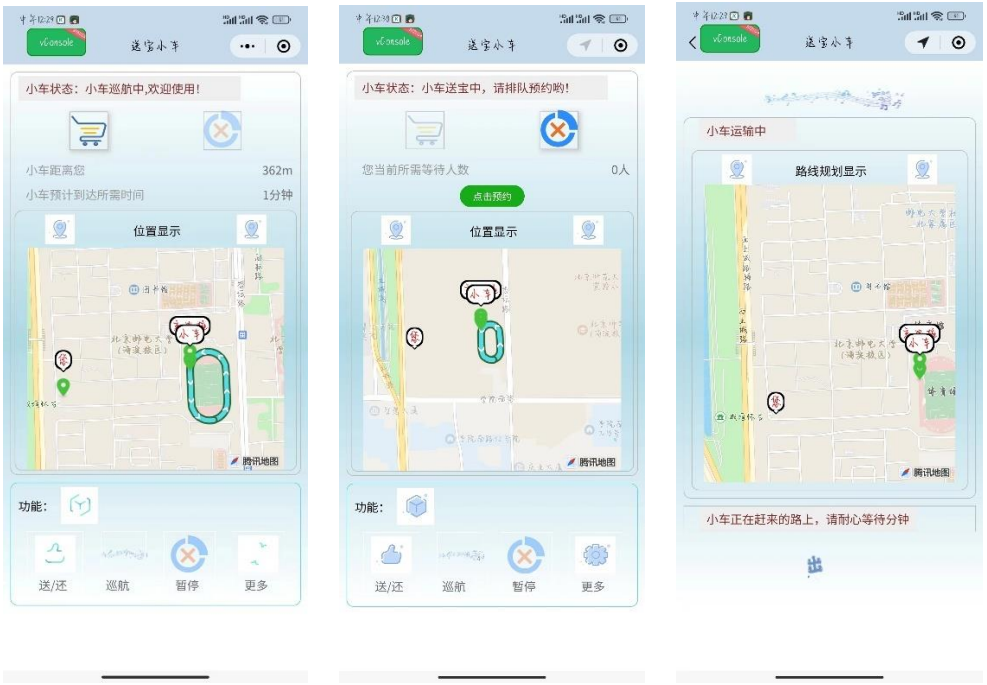
5. 课题成果及其性能分析（包括已完成的功能，主要实验数据、实验方法等）

5.1 已完成的功能

一、微信小程序

① UI 设计

我们的用户界面（UI）是基于小程序构建的，旨在优化用户交互体验。在设计过程中，我们精心考虑了界面的交互性、功能性和美观性，确保能够全面满足用户的使用需求。



首先使用原型设计用户使用界面，之后使用 h5+CSS 实现 UI。步步推进，加入不同动效元素，以蓝色为底色，完成了 UI 的美化，使得微信小程序的界面变得好看，简约，操作间单便捷。

③ 获取用户位置信息

当用户首次进入微信小程序时调用 wx.getLocation()接口获取用户的初始位置，之后使用坐标转换算法 Google_Coordinates(res.latitude, res.longitude)将原始 WGS 数据格式转换为火星坐标系下的经纬度格式，使得其与 map 组件中所要求的数据格式相同，防止出现位置偏移，以实现用户位置的精确展示。

④ 路径生成

目前演示只需要直线规划路径，结合路径规划 API 可以实现折线规划

一是，当用户点击“出发”按钮，微信小程序能够根据小车所在地与用户所在地的经纬度信息生成他们之间的路径，并将路径通过 OneNET 发送至树莓派，使得树莓派控制 Donkey Car 前往用户处；

二是，当用户还充电宝后，点击“巡航”按钮，微信小程序能够根据用户输入的起点与终点生成他们之间的路径，并将路径通过 OneNET 发送至树莓派，使得树莓派控制 Donkey Car 回到充电桩的位置充电后开始下一次巡航 。

④ 多用户串行使用

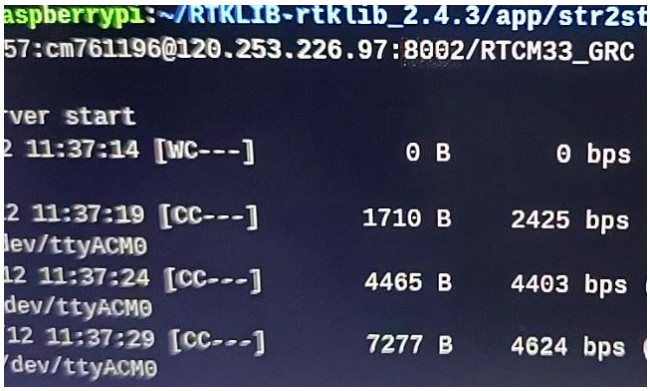
微信小程序通过数据库中的 userList 队列进行识别，能够实现多个用户的串行使用。

当小车处于闲置状态时，userList 队列为空，当前用户可以马上使用。当小车处于使用状态时，userList 队列不为空且不包含当前用户，可以进行预约。

若 userList 队列不为空且包含当前用户，则当前用户已经进行预约，等待排队，小车会按照队列的顺序依次使用。

二、Donkey Car

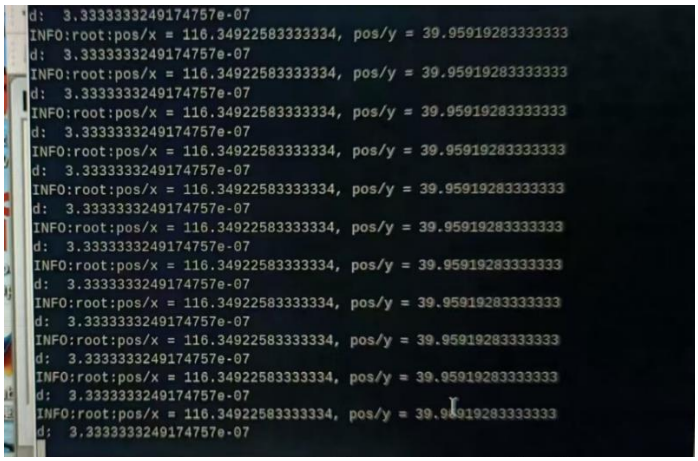
1、GPS 成功连接到 RTK 服务器，通过树莓派进行数据交互，从而达到厘米级别的定位精度



图表 40 GPS 与 RTK 进行数据传递

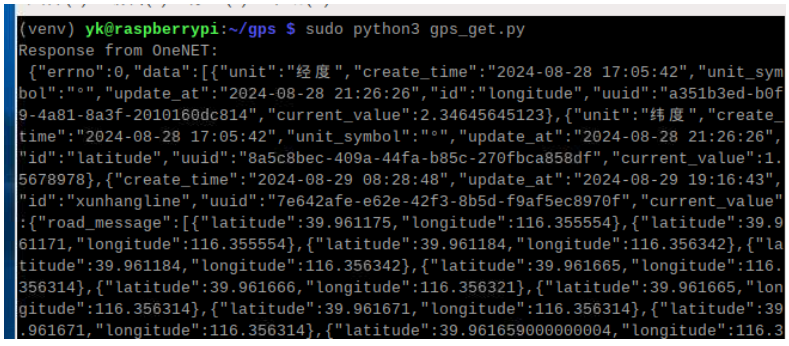
2、实时获取 GPS 位置信息并且静态漂移误差不大于 5cm

GPS 模块实现了以 115200 的波特率获取 NEMA 语句并解析成经纬度，进一步转换成 NEMA 坐标，在空旷区域能够实现比较好的效果。



图表 41 获取 GPS 位置信息

3、 将小车 GPS 坐标发送到 One NET 以供微信小程序读取，同时将 UTM 坐标传递到其它模块以使小车能实时按照坐标进行循迹，同时可以获取 One NET 上面的数据。



图表 42 小车 GPS 位置上传至 One NET，并且获取数据

4、小车加载规划好的路径实现任意路径的循迹功能。

加载规划好的路径，是一个 csv 文件

```
116.35378085714285,39.95996857142857,3.0000000000e-01
116.35377904761904,39.959964642857145,3.0000000000e-01
116.35377723809523,39.959960714285714,3.0000000000e-01
116.35377542857142,39.95995678571428,3.0000000000e-01
116.35377361904762,39.95995285714285,3.0000000000e-01
116.3537718095238,39.95994892857143,3.0000000000e-01
116.35377,39.959945,3.0000000000e-01
```

代码实现路径跟随

```
INFO:root:nearest: (60.73363462317502, 70.87076207995415) to (0, 0)
INFO:root:CTE: 0.5139374065468201 steer: -0.010278748130936401 throttle: 0.25
INFO:root:pos/x = 0, pos/y = 0
INFO:root:nearest: (60.73363462317502, 70.87076207995415) to (0, 0)
INFO:root:CTE: 0.5139374065468201 steer: -0.010278748130936401 throttle: 0.25
INFO:root:pos/x = 0, pos/y = 0
INFO:root:nearest: (60.73363462317502, 70.87076207995415) to (0, 0)
INFO:root:CTE: 0.5139374065468201 steer: -0.010278748130936401 throttle: 0.25
INFO:root:pos/x = 0, pos/y = 0
INFO:root:nearest: (60.73363462317502, 70.87076207995415) to (0, 0)
INFO:root:CTE: 0.5139374065468201 steer: -0.010278748130936401 throttle: 0.25
INFO:root:pos/x = 0, pos/y = 0
```

实际效果见视频

图表 43 小车实现路径跟随

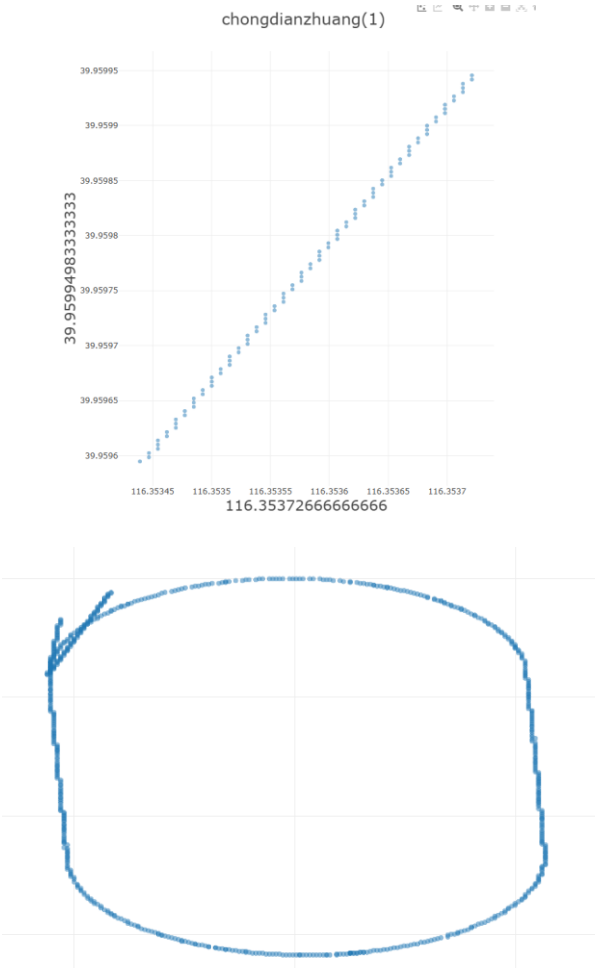
5.2 实验数据

一、GPS 静止时的数据，误差 3cm

```
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
INFO:root:pos/x = 116.3492258333334, pos/y = 39.95919283333333
d:  3.3333333249174757e-07
```

图表 44 GPS 静止时的数据

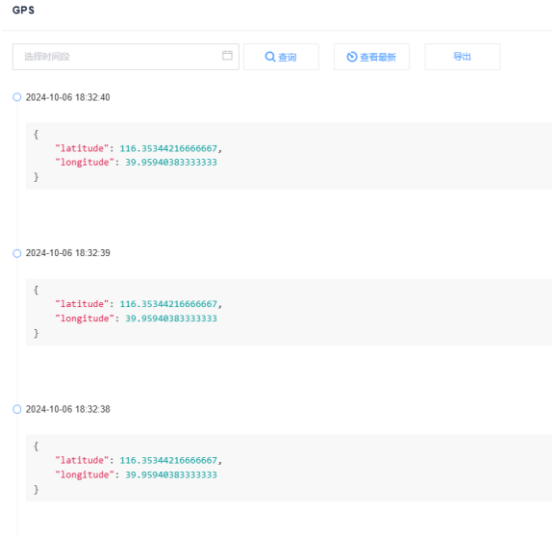
二、GPS 移动时的寻人/充电桩的路径规划



图表 45 小车的 GPS 的路径规划

三、OneNET

① OneNET 上小车的经纬度数据，保持实时更新

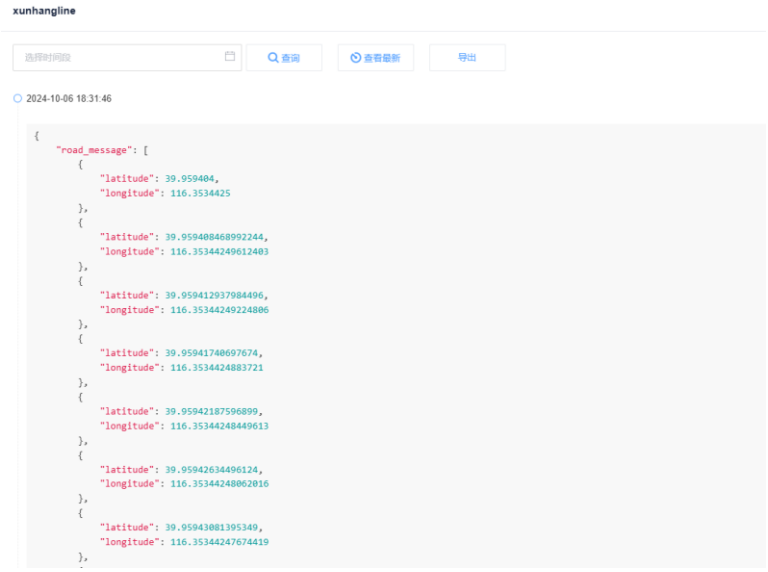


图表 46 树莓派 IP 与经纬度

① 微信小程序端生成的路径数据并上传到 OneNET

图表 47 小车送宝路径数据

② 小程序生成会充电桩的路径信息



图表 48 小车返回充电桩

5.3 实验方法

第一步，系统功能确定。对课题内容进行市场调查、查找文献，确定用户需求，再根据用户的需求进一步确定系统的功能。

第二步，模块的设计与实现。根据系统的功能划分为不同的模块，交由不同的同学负责，对各自的内容进行学习，完成各自的模块内容。

第三步，系统构建。将不同的模块代码整合，并修改冲突的部分，去掉冗余的内容，保证能够运行。

第四步，整体优化。继续实验，不断修改参数和代码，以达到预期的效果。

6. 课题成员分工与合作情况说明（包括各成员工作内容和贡献率，组间、组内协作与分享说明）

6.1 工作内容

小组成员	成员分工
闫康	负责编写和调试控制小车的软件代码，完成了路径跟随算法、优化 PID 参数等工作；与硬件组装负责人合作，确保软件能够正确控制硬件，并进行代码集成和优化
薛皓林	负责设计和开发微信小程序的用户界面，确保用户能够通过小程序实时监控小车状态，并进行远程控制；与软件开发负责人合作，确保小程序能够与小车的控制系统无缝对接
苏珈磊	负责树莓派 4B 小车的硬件组装，包括电路连接、传感器安装和电机配置，以及测试各个硬件组件的功能；与系统测试负责人合作，参与系统整体的调试与优化和文档撰写
孙常鑫	负责整体系统的测试工作，包括功能测试、性能测试和稳定性测试，以及编写项目文档；协助软件开发负责人进行工作，完成了路径跟随算法参数的调整工作

6.2 组内协作

① 线下交流

由于我们小组成员是一间宿舍的，因此并没有使用腾讯文档等线上工具。每个周末，我们宿舍的团队成员都会安排时间进行面对面的交流会议。在这些会议中，我们会详细回顾每个人在项目中取得的进展和完成的工作。我们会将每个人的更新和成果进行整合，并与我们最初的计划和目标进行对比分析。根据复盘的结果，我们识别进度上的差距，讨论存在的问题，并基于当前的实际情况对接下来的工作计划进行调整。

② 实地测试

我们的项目依赖于实地测试小车的性能和功能，因此我们制定了一个计划，每周至少进行三次实地测试。在这些测试中，至少需要两名成员参与，以确保测试的顺利进行和安全。在实地测试的同时，其他小组成员则留在室内，专注于完成分配给他们的其他特定任务，如算法优化等工作。

当遇到技术上的挑战或难题时，我们整个小组会一起出动，共同参与到室外测试中。我们会对测试过程进行详细的记录，包括测试条件、遇到的问题以及任何观察到的现象，以便于归来后进行复盘，并指定下一次的调试策略。

6.3 组间分享

我们的小组在专注于自身项目的同时，也积极与其他小组进行交流和讨论。这种跨团队的沟通让我们能够共同探讨在项目中遇到的共性问题，比如与跟跑小

车组讨论 GPS 模块的精度问题。我们会一起研究影响 GPS 定位精度的因素，并探索减少这些误差的方法。

此外，我们还会讨论路径规划和跟随算法的实现，分享各自的思路和进展，从而相互学习、借鉴对方的长处。通过这种开放式的合作，我们找到了许多不足，并有针对性的优化我们的算法设计，提升小车性能和可靠性。

图表 49　小组技术人员操场调试进行中

7. 课题所用器材列表及说明（包括课程提供的器材及自行购买的器材）

序号	名称	型号	数量	单位	金额（元）
1	树莓派	4B	1	个	464
2	Donkeycar 小车(附手柄)	HSP94186 有刷	1	个	1718
3	电机驱动芯片 PCA	PCA9685	1	个	14.50
4	GPS 模块和 GPS 天线	L76X GPS HAT	1	个	367
5	XiaoMi 移动充电宝(给树莓派供电)	20000mAh	1	个	66.9
6	树莓派 4B 散热器铝合金散热风扇	Raspberry Pi4B PWM	1	个	19.9
7	微信小程序云服务	基础版	3	月	59.9
8	HSP 1/16 1/18 车模原装电池 7.2V 1100 毫安镍氢电组 小田宫插头(配套充电器)	7.2V 1100 毫安镍氢电组	2	个	60
9	闪迪内存卡手机 32g/64g/128g/高速 tf 行车记录仪存储卡 sd switch 卡	32G	1	个	29.9
10	创乐博树莓派 10 寸高清显	10 寸	1	个	260

	示屏(附 HDML 线)				

8. 课题成果链接（包括视频、代码）

演示视频：https://windows-dance-tb6.craft.me/0Zn6d4NZNNE4RJ

Donkey Car 代码：https://pan.baidu.com/s/1lGn6J6a_kHCJiIqoJQD94g(rhrm)

微信小程序代码：https://github.com/jiang-you-si/-.git

9. 参加本课程的收获、体会及对课程的建议

9.1 参加本课程的收获、体会

1. 独立解决问题的能力：在这次课程的历练中，我们逐渐领悟到，在遭遇未知挑战时，如何独立地剖析问题、逐层深入，直至找到解决之道，而非一味寻求他人的援手。这种自我探索与解决问题的能力，如同宝藏一般珍贵，它赋予了我们面对未来挑战时的自信与力量。

2. 时间管理和多任务处理：在课程的持续推进中，我们得以跨越小学期与秋季学期的界限，深入领悟了时间管理的艺术。面对纷至沓来的多项任务与层层叠加的压力，我们学会了如何巧妙地安排每一分每一秒，使得时间不再是束缚，而是成就的催化剂。这一过程，不仅锻炼了我们的时间规划能力，更让我们在处理多重任务时游刃有余，显著提升了我们的效率与执行力。

3. 知识储备和自主学习能力：在课题的挑战中，我们如同探险者一般，不断挖掘知识的宝藏。这些挑战激发了我们探索未知的热情，促使我们积极学习新知，从而丰富了我们的知识储备。在这个过程中，我们不仅学会了如何高效地搜集资料，更学会了如何甄别信息的真伪与价值，这让我们的自主学习能力得到了显著提升。我们像是在知识的海洋中航行，不断吸收养分，使自己变得更加充实和强大。

4. 团队合作和沟通：在团队合作的舞台上，我们深刻体会到了协作的力量。我们认识到，即使是最微小的疏忽，也可能引发连锁反应，导致严重的后果。通过与团队成员的深入交流，我们学会了如何在思想的火花碰撞中，寻找到更加卓越的解决方案。这种沟通不仅促进了问题的解决，更让我们学会了倾听、尊重他人的观点，以及如何在集体智慧中找到灵感的源泉。即使我们在项目的完成方面，对于方案选择等小组成员间存在过分歧和非议，但是最后我们都会向着最终的目标去前进，最后才能使我们的项目走向成功。

5. 耐心和持之以恒：在挑战的征途上，我们学会了以耐心为盾，持之以恒为剑。面对重重困难，我们不再急躁，而是以冷静的心态去迎接每一个挑战，理解到解决问题并非一蹴而就，而是一个充满探索与发现的旅程。每一次的失败，都是对正确答案的一次接近，每一次的尝试，都是对成功的一次积累。在本项目的实现过程中，我们每个人都会用进展报告来记录相关开发中遇到的问题，并且规划好次日或者下一周的工作，我们一路上解决了选题确认问题----树莓派连接稳定问题----GPS 硬件问题----小车调试问题----RTK 问题确保 GPS 硬件精度----小程序和 OneNet 服务器连接问题----并且小组四人在风雨室外操场上进行了数百次的实验调试，我们坚信，只要坚持不懈，终将迎来胜利的曙光。

6. 嵌入式开发的挑战：在嵌入式开发的挑战中，我们认识到了它与软件编程相比的复杂性。它要求我们不仅要关注软件逻辑，还要深入到硬件、接口和外部环境的每一个细节。我们学会了如何调试和定位问题，这些技能对我们未来的开发工作来说，无疑是宝贵的财富。通过这些经历，我们不仅提升了技术能力，更锻炼了解决问题的思维方式，为未来的技术挑战做好了准备。比如我们在实验的过程中遇到过 PCA9685 与树莓派连接不稳定的问题导致小车无法驱动和转向，以及室外天气的不同，环境的不同对于 GPS 数据的影响等一系列外部因素造成的问题。

7.代码理解和框架分析：在代码理解和框架分析的探索之旅中，我们不仅深入研究了 Donkey Car 的代码和库，逐步揭开了其工作原理的神秘面纱，还勇敢地迈出了创新的步伐。我们在此基础上进行了创新性的修改，不仅优化了现有功能，还引入了新的模块——GPS 模块和小程序模块，以满足我们项目的独特需求。 引入 GPS 模块，我们学会了如何精确地获取和处理位置数据，这对于项目的定位和导航功能至关重要。我们掌握了如何集成硬件设备，以及如何确保软件与硬件之间的无缝协作。 同时，通过开发小程序模块，我们拓宽了项目的交互性和可访问性。这不仅提高了用户体验，还让我们学会了如何设计和实现用户友好的界面，以及如何将复杂的功能简化为易于操作的小程序。 这些新增的模块，以及我们在这个过程中积累的宝贵经验，不仅增强了我们对代码的理解和框架分析的能力，还让我们掌握了调试代码和解决 bug 的技巧。这些技能的积累，无疑为我们未来在编程领域的探索和创新奠定了坚实的基础，让我们在技术的道路上更加自信地前行。

8.创新思维的训练:我们打破了传统意义上此类项目的思考，利用创新思维构建出来了新的项目，这体现出了当代北邮大学生的创新思维。创新实验课程鼓励我们思考问题的解决新方法，创造具有价值的新的项目，培养了我们的创新思维，让我们在面对问题时能够跳出传统的框架，寻找新的解决方案。

9.2 对课程的建议

1.时间方面:对于我们项目来说，时间还是比较紧张的，直到我们答辩的前一天整个项目才算是一个完整的闭环结束，并且当初设想的很多东西无法完成，例如避障模块，语音模块，视觉模块等等，并且只能针对于室外的项目，建议下一届可以适当增加两周时间，到期中考试之前，我觉得时间刚刚好，因为我们还要面临其他课程的压力，其他课程单独也很大，还有诸如电磁场实验，通信系统实验等一系列课程实验等待完成。

2.选题问题:建议老师可以讲解一下学长学姐们的往届选题，紧张情况，各种不同类型的选题难度等，毕竟我们的创新实验是一个四学分的大课，占用的时间很多，老师应给出足够的时间去思考同时要给予同学们足够的建议。同意建议老师简介一下选题时，这个方向的选题可能会遇到什么重大的问题，有没有踩坑点等等，至于一些技术问题，例如本学期讲的神经网路，Donkeycar 等，可以缩减学时，我们人为学生们的自主探索会更加的重要，毕竟不能让小组的选题变成一个无底洞，遇到的问题根本无法解决或者远超出本科生的能力。同时很多同学对于这种东西没有什么经验，这个和电赛还不一样，没有确切的选题来供同学们选择，希望老师能够给出一些选题的建议，哪怕一个方向的选题限制几组可以选择，避免重复，因为将自己的创新想法付之于商业应用的实际场景还是非常有难度的，毕竟在当今这个时代，基本上各行各业都已具备一定基础，在创新是一件十分困难的事情，耗时费力。

3. 硬件设施问题:在我们对其他小组的项目的对比加上我们自己的项目分析，我们不难发现，大多数小组的选题都是选择是控制类的项目，这就对于硬件的要求精度非常高，因此建议学校及时的对一些热门硬件，比如 Donkeycar，电池，树莓派 4B 开发板等硬件及时换新，确保没有问题，比如我们在一开始利用旧车进行实验时，旧车的车轮和舵机转向杆发生脱落，走直线也无法走，希望能够及时淘汰过于陈旧的硬件，避免因为不必要的硬件问题而造成宝贵时间的浪费。

10. 参考文献

一、微信小程序

微信开放文档：https://developers.weixin.qq.com/miniprogram/dev/framework/

Vant Weapp：https://vant-contrib.gitee.io/vant-weapp/#/home

异步 Promise：https://www.runoob.com/js/js-promise.html

微信小程序交流社区：https://developers.weixin.qq.com/community/

二、Donkey Car

树莓派实验室：https://shumeipai.nxez.com/

Donkey Car 文档：http://docs.Donkey Car.com/

关于树莓派 4B 两个 i2c 通道的理解：https://zhuanlan.zhihu.com/p/361521936

地图坐标系转换：https://tool.lu/coordinate

树莓派 ACT LED 指示灯闪烁模式代表的状态：

https://shumeipai.nxez.com/2021/05/26/raspberry-pi-act-led-error-patterns.html

DonkeyCar 自动驾驶: https://medium.com/@johnas.io/building-and-training-donkeycar

四、OneNET 服务器

MQTT 协议：https://open.iot.10086.cn/doc/v5/develop/detail/588

MQ 消息队列：https://open.iot.10086.cn/productservice/msgMq/

基站定位：https://open.iot.10086.cn/doc/v5/fuse/detail/1088

WiFi 定位：https://open.iot.10086.cn/doc/v5/fuse/detail/1090

上传数据点：https://open.iot.10086.cn/doc/v5/develop/detail/593

查询数据流：https://open.iot.10086.cn/doc/v5/develop/detail/588

OneNET 学院：https://open.iot.10086.cn/college/home

五、路径规划算法

路径规划 API 文档:https://lbs.qq.com/service/webService/webServiceGuide/webServiceRoute

基于 Raspberry Pi 的 Traccar GPS 跟踪系统:

https://pimylifeup.com/raspberry-pi-gps-tracker/

路径规划 API：https://lbs.qq.com/service/webService/webServiceGuide/webServiceRoute

六、其他资料

L76X GPS HAT：https://www.waveshare.net/wiki/L76X_GPS_HAT

该 GPS 支持的输出格式（百度百科）：https://baike.baidu.com/item/NMEA-0183/1810482

该 GPS 的格式解析（知乎，阅读方便）：https://zhuanlan.zhihu.com/p/591884197

CMP10A 资料：https://github.com/Staok/IMU-study

Raspberry Pi GPS 模块教程：https://learn.adafruit.com/adafruit-ultimate-gps-on-the-raspberry-pi

附件：课题执行过程中的主要问题及解决方法

1. **Q：GPS 定位，静态漂移问题如何解决才能获得最佳的效果？**

   A：RTK 实时动态差分技术，通过基准站（中国移动）和流动站的配合使用，可以提供厘米级别的定位精度，有效减少静态漂移，实现效果极佳，在整个校园内只要天线无明显覆盖，均是 cm 级定位。

2. **Q：树莓派与 RTK 服务器连接不稳定问题如何解决？**

   A:。确保树莓派的网络连接稳定。如果使用 Wi-Fi 连接不稳定，可以考虑使用有线连接或者更换无线网卡增强信号，更重要的是发往服务器的数据格式需要做一个修正，不然服务器是无法接受到传输的数据的，这些均在《GPS 与 RTK 连接说明》中有完整介绍。

3. **Q：小车实现路径跟随时走‘S’形曲线应该如何解决？**

   A：PID 控制器的参数设置十分重要，P 参数，和 D 参数最为重要。这两个参数的调节需要用户耐心进行实验。推荐使用二分法。

4. **Q：程序运行出现串口冲突问题解决方案是什么？**

   A： PCA9635 与 GPS 模块占用了同一个串口，所以会出现串口冲突的报错问题，所以需要将 GPS 模块与树莓派连接的串口定为 USB 串口，因为 GPS 模块还额外需要一个 USB 串口与树莓派连接而进行 RTK 数据矫正，所以 GPS 模块上面需要两个 USB 串口（一般只有一个），因此需要额外购买一个串口转换器。

5. **Q：小车寻找用户失败是怎么回事？**

   A：用户手机的 GPS 定位也有可能出现漂移问题，因此用户手机的 GPS 定位精度有一定要求（m 级即可）。

6. **Q：小车返回充电桩出现失败的问题如何解决？**

   A：小车返回充电桩出现失败的主要问题是，小程序从 One NET 获取小车实时位置出现延迟的问题，因此路径规划不准确，因此在规划路径前需要等待 5 秒左右。

7. **Q：小车在用户控制停车后，卡死不动如何解决？**

   A：手柄和小程序不能同时控制，卡死或者加载不出来路径是由于同时控制导致的。

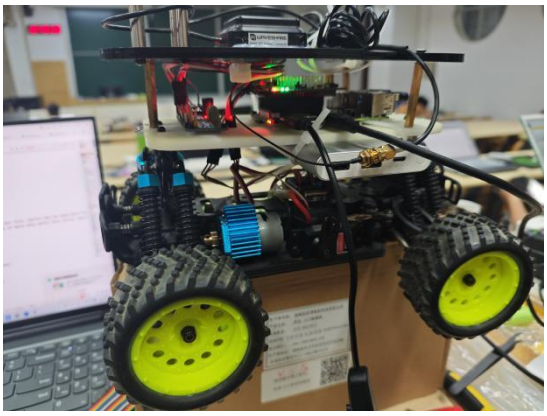
8. **Q：小车无法跑直线的原因探究与思考？**

   A：我们主要遇到的还是硬件的问题，之前使用的旧的小车，曾经的轴承和连接杆掉下来过，这对我们小车的平衡和方向感之间造成了很大程度上的影响，因此我们更新了小车的硬件，进而最后解决了这个问题，也许是暂时性解决，也许是偶然性很大，但是至少测试时是成功的，丝滑的行驶。

9. **Q：安装硬件时，我们没有 3D 打印的架子了，如何解决，硬件布局如何分配？**

   A：利用打电赛时的遗产，利用一块轮趣科技的小车 R3 底盘，并利用电转转

孔成功将小车加高，并利用适配的螺丝固定。将整个小车分为三层，小车整体的结构层次分为三层，最底层是四个充气橡胶轮胎，以及舵机和电调等控制驱动设备，和车载供电锂电池，用于给小车的电调和舵机供电；中层是主控板树莓派 4B 和舵机驱动板 PCA9685，以及 GPS 扩展板，中间一层是主控层，相当于核心，并且有散热系统；最顶层是最高处的 GPS 天线，小车模块和层次都十分地清晰。小车的材料多为塑料或者是金属材质，轮子是采用常见的橡胶轮子。车身所有外设都是用热熔胶或者是螺丝进行了固定。



图表 50　小车实物图

10. **Q：手柄默认按键并不足以实现项目功能，怎么办，是不是还要分配，如何分配按键？**

A：我们觉得类似管脚分配一样，需要我们重新定义。在处理复杂的代码系统时，如果对整体框架缺乏深入的理解，那么解决特定问题的挑战性会显著提高。这种情况尤其在系统功能设计不完善，或者用户界面上的按键与实际功能不一致时表现得更为突出。尽管面临这些困难，我们通过持续的探索和分析，最终找到了控制按键功能分配的关键函数。深入研究这个函数后，我们发现了一个值得注意的现象：在 Xbox 游戏控制器的设计中，有一些按键并未在系统的功能框架中得到定义，这些按键被系统识别为“未知”（unknown）状态。这意味着这些按键在默认情况下是没有特定功能的。为了充分利用这些按键，我们采取了一个策略：通过编程手段，将这些“未知”按键的物理地址与用户希望实现的具体功能进行关联。这样，我们就能够为这些原本未被定义的按键赋予新的、有用的功能，从而增强了控制器的可用性和用户体验。这个过程涉及到对系统进行定制化的修改，以适应用户的特定需求和偏好。

11. **Q：微雪树莓派 GPS 拓展板和天线连接的接口发生断裂，导致拓展板不可用。**

A：我们手中自己没有多余的 GPS 套件，只能在 8 月 29 日寻找老师跟换套件，我们下次准备使用清理的可清除的热熔胶来固定拓展板，尽可能的避免再次出现这样的问题，因为这个接口连接设计的实在是太不牢固了，并且 GPS 套件也是我们项目的核心组成部分，尽可能的不要出现硬件损坏的问题，这也算是给其他用到 GPS 套件同学的一个警醒。

图表 51　GPS 扩展板和天线间的接口发生断裂