



机电控制课程设计实验报告

小组： 第 7 组

课题： 基于树莓派的人物追踪小车

组长： 葛易谕

组员： 黄宸睿

李澍凡

刘启源

Project Report:基于树莓派的人物追踪小车

设计目的

树莓派是能够实现 PC 所有功能的微型电脑，以其低能耗、移动便携性、GPIO 等特性在学习计算机编程等方面有着广泛的使用。并且，树莓派能够完成比单片机更复杂的任务管理与调度，基于树莓派的智能小车也能够实现更多复杂的功能，并且因为开发语言不仅限于 C 语言，能够搭载 TensorFlow 进行深度学习相关功能的实现。树莓派智能小车上手难度不高，对于初次接触树莓派的同学经过学习也能够进行相关的拓展应用。

我们本次课程设计的题目是**基于树莓派的人物追踪小车**，并且在此基础上增加了超声波避障、红外线定位等功能的实现。

主要功能定位有：

1. 使用超声波和红外传感器实现的障碍物检测与回避；
2. 将树莓派摄像头拍摄到的视频流传到 PC 端，并在 PC 端查看；
3. 基于视觉使小车能够沿车道线行驶；
4. 实现基于视觉的人物识别，并能够根据目标的移动自动跟踪。

本次课程设计主要实现的几项功能都已经有了较为成熟稳定的算法，本次设计的目的也并非追求创新，因为小组中有未深入学习过相关知识的成员，故我们想要通过树莓派小车实现的大多是较为基础的功能，也能够让每个小组成员都能够参与项目的设计与完成。

通过这次课程设计，我们希望能够让小组成员更深入地学习理解树莓派、单片机等相关知识，增强动手能力，培养团队意识，加强代码的理解与编写能力，能够通过这次课程学到更多有用的东西。

原理说明

本项目是基于树莓派的智能小车，主要实现的功能包括：自动避障、实时图像传输、视觉车道循迹、目标检测以及人物追踪，接下来将对各项功能的原理进行程序架构的说明。

A 自动避障

自动避障功能的实现是基于超声波传感器和红外避障传感器，使小车在运行过程中不会撞上障碍物。超声波传感器的控制口发一个 10us 以上的高电平，就可以在接收口等待高电平输出。一有输出就可以开定时器计时，当此口变为低电平时就可以读定时器的值，此时就为此次测距的时间，方可根据

声速算出距离。红外避障传感器传回 0 表示前方有障碍物，传回 1 表示前方无障碍物。主程序为 `main_obstacle_avoidance.py`，超声波传感器测出小车距离前方障碍物的距离，两

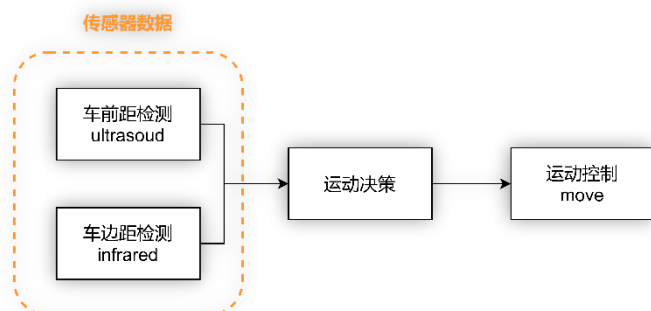


Figure 1 自动避障程序架构图

边的红外避障传感器测出两边是否有障碍物，根据测量结果进行运动决策和电机控制。

B 实时图像传输

该功能是将树莓派摄像头拍摄到的视频流传到 PC 端，并在 PC 端查看。目的是为了便于摄像头姿态的调整和图像处理算法的调试。我们使用 UDP 传输协议进行图像传输。具体实现主要分为发送端和接收端两部分：

发送端： (camera.py VideoTransimssion)

- 图像编码 (cv2.imencode)
- 校验数据发送 (数据长度作为校验)
- 编码数据发送 (socket.sendall)

接收端： (pc_receiver.py)

- 接收校验数据 (4 字节数据)
- 接收图像编码 (校验数据后的第一个数据包)
- 简单校验 (校验数据 == 编码数据长度)
- 图像解码 (cv2.imdecode)

其中，发送端在树莓派端运行，接收端在 PC 端运行。二者同时运行。

C 视觉车道循迹

基于视觉，使小车沿车道线行驶。环境要求为白色的地板，黑色 (深色) 的车道线。主程序为 main_lane_tracking.py

车道线检测

- 图像二值化，提取车道线；
- 提取车道线的内侧点：在图像的特定行，从中间向两侧检索，检测到 0 像素点及为车道线点。图像左边边检测到的点即为左车道线点，右半边检测到的点即为右车道线点。(为防止因各种原因车道线部分缺失，我们选择图像 4 个行提取 4 组车道线内侧点)。

运动决策

- 如果两侧车道线都能检测到，则直行。
- 如果只能检测到一侧车道线，则有三种情况：
 1. 该侧车道线靠近图像边缘，则还可以继续直行；
 2. 该侧车道线靠近图像中央，则急需转弯，原地旋转；
 3. 该侧车道线位置适中，则缓慢转弯，在前进中转弯。
- 如果两侧车道线都检测不到，则维持之前的动作。

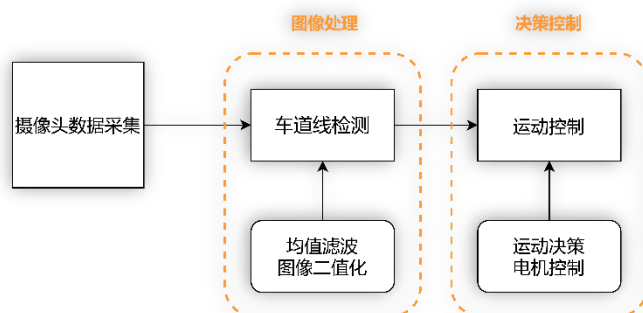


Figure 2 视觉车道循迹程序架构图

运动控制 根据决策结果，控制小车电机输出。

D 人像追踪

该功能主要实现的是识别并定位摄像头图像中的人物，并且对人物进行定距离追踪。主程序为 `main_object_detection.py`，其调用了 TensorFlow Object Detection API，使用了训练好的 SSDLite 目标检测模型，在树莓派端进行目标检测。我们使用的 SSDLite 模型

主要优点是运行速度快、占用内存小，适合在树莓派端进行运算。接着利用 `visualization_utils` 坐标提取，检测得到人体在屏幕坐标系下的坐标，根据图像坐标数据决策车辆的转向移动。

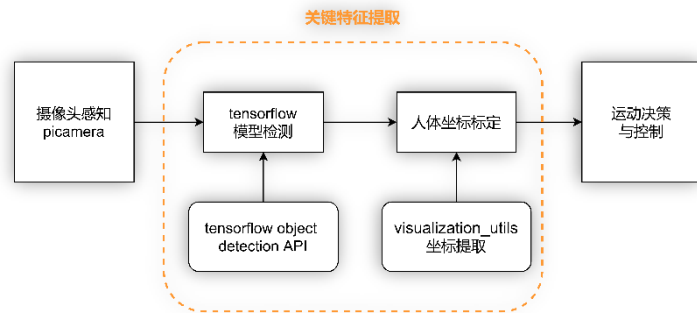


Figure 3 人像追踪程序架构图

电路图

小车使用的硬件接线情况如 Figure 4 所示。图中最大的灰底虚线框部分是一块底板，它承担着锂电池向除树莓派外的硬件的供电、各硬件模块与树莓派之间的信号传递。

值得注意的是，该底板同样具备直接向树莓派供电的线路，但是实际测试中我们发现由于电池容量限制，为了保护树莓派免于由于低电压的非正常关机，使用了一只充电宝单独承担向树莓派的供电工作。

小车摄像头模块使用的是树莓派官配 Pi-camera。

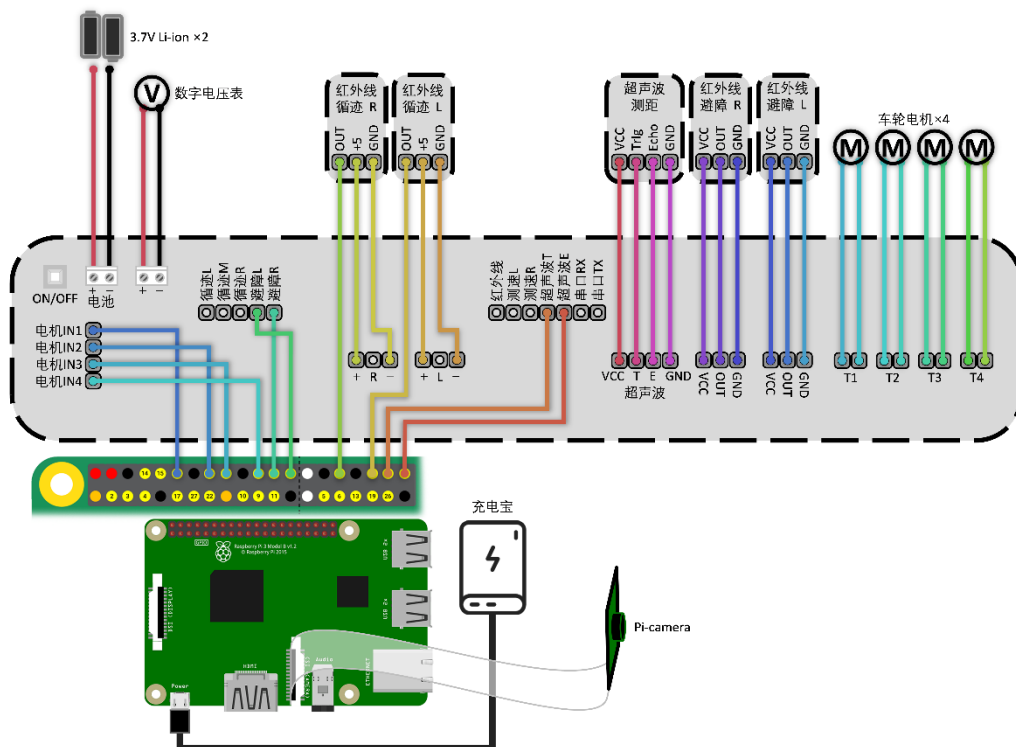


Figure 4 电路示意图

多角度实物照片

超声波测距

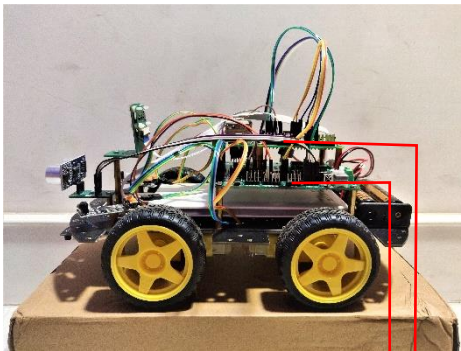
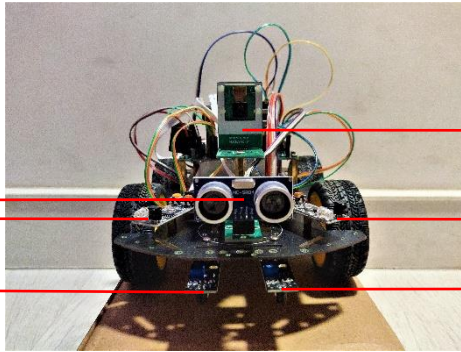
右侧红外避障

右侧红外循迹

Pi-camera

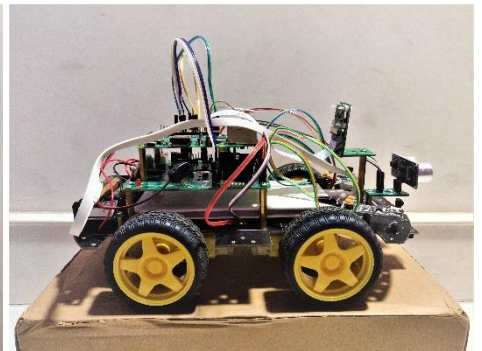
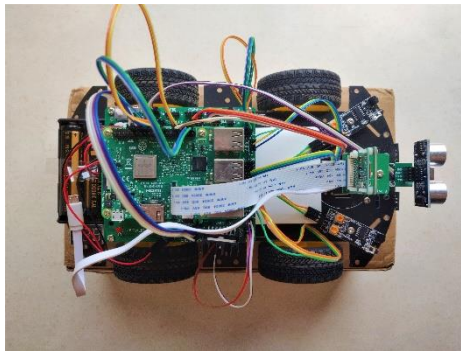
左侧红外避障

左侧红外循迹

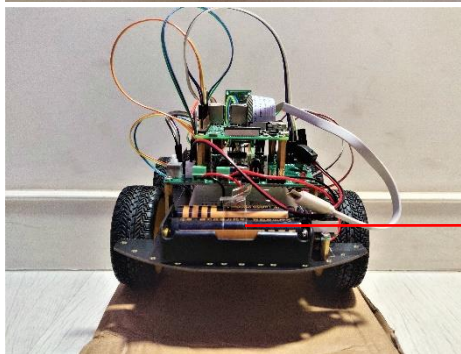


底板

树莓派



锂电池盒



程序架构

主功能包

main_lane_tracking.py

main_object_detection.py

main_obstacle_avoidance.py

功能模块

ultrasound.py 超声波检测

infrared.py 红外线检测

track.py 赛道检测

move.py 舵机移动

pc_receiver.py 实时图象传输

object_detection

目标检测

visualization_utils 图形检测可视化

.....

在参考项目基础上的改进点

1. 增设人像追踪功能：

- 利用原有的 TensorFlow Object Detection API，通过修改 API 文件获得人体在屏幕坐标系下的坐标，进而根据图像坐标数据决策车辆的转向移动。
- 针对转向过程中舵机转速问题，添加软件定时器，限定舵机通电时间，较好的限定了舵机的转向速度，间接地限定了小车转向的角度。
- 根据实际调试优化了舵机转动参数，使跟踪得到了一定的提升。

2. 电源管理：

树莓派使用独立电源供电，避免了电机快速耗电带来的电压过低异常关机的问题。

3. 红外避障模块的优化：

原套件安装图纸以及参考项目中的避障模块安装方式容易导致在临界障碍物逼近时陷入决策死循环：即小车进一步能检测到障碍物，做出相应的后退响应后又检测不到该障碍物，导致重新做出前进响应。这种情况在障碍物边长较大时基本不会发生，但在遇到较小边长的障碍物时则很容易陷入死循环。我们对其的改进是通过重新安排了红外避障模块的安装角度和位置以及响应的灵敏度调试实现的。

以上的改进点中，1 是课题所明确框定的开发方向，2&3 则是在调试过程中为方便功能实现而做出的改进。

组员分工情况

Table 1 简要概述了本学期课程设计过程中小组各成员的工作。由于疫情影响组员无法集合，因此对硬件直接上手操作只能由一人完成，综合考虑后硬件由组长葛易谕处理，小组通过腾讯会议的对接形式进行开发和调试。

Table 1 组员分工情况概述表

葛易谕	35%	<ol style="list-style-type: none"> 1. 树莓派机器视觉与机器学习框架的安装 2. 硬件接线与初步调试 3. 程序开发过程中的测试运行 4. 维护 Github 项目以及每周项目文档
黄宸睿	25%	<ol style="list-style-type: none"> 1. 项目程序“架构”的布局 2. 软件部分调参 3. 与负责硬件的同学会议对接“云”调试
李澍凡	20%	<ol style="list-style-type: none"> 1. 选题前期的文献检索 2. 对参考项目的代码的学习与分析 3. 小车程序的调试和硬件的组装 4. 协助准备答辩材料
刘启源	20%	<ol style="list-style-type: none"> 1. 选题前期的文献检索 2. 对参考项目的代码的学习与分析 3. 参与了对小车功能实现中存在问题的讨论 4. 协助准备答辩材料