

学 号: 201614420112



华北理工大学  
NORTH CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY

# 毕业设计说明书

GRADUATE DESIGN

设计题目: 基于 Arduino 倒车雷达的设计与实现

学生姓名: 刘佳玮

专业班级: 16 计科 1 班

学 院: 人工智能学院

指导教师: 曾屹 高级实验师

2020 年 06 月 13 日



## 摘 要

本文以实现汽车倒车安全性和便捷性作为设计背景，设计出一种基于超声波测距且含报警系统的倒车雷达系统。系统将测量倒车时与障碍物的距离并反馈给用户，并在 PC 端上位机实时显示倒车状况，最后与报警系统相结合，以增加驾驶者在倒车停车时候的安全性以及便捷性。本设计选用 Arduino 作为控制器，依照超声波测距理论，选用超声波传感器为测距工具，通过 PC 端上位机发送控制指令操作车辆驾驶。当与障碍物的距离小于安全距离时，系统将采取紧急制动。最后通过模拟测试，发现系统不仅能实现在水平正方向 5cm-100cm 的距离较准确测距，对于车体后方的形状规则的障碍物有一定的提醒作用。本设计具有成本低，集成度高等优点，对于辅助驾驶员倒车有很好的帮助作用，因此具有一定的理论价值和实用价值。

**关键词** Arduino；倒车雷达；超声波测距

## Abstract

In this paper, a reversing radar system based on ultrasonic ranging and alarm system is designed to realize the safety and convenience of automobile reversing. The system will measure the distance between reversing and obstacles and feed back to the user, and display the reversing condition in real time on the PC upper computer. Finally, it will be combined with the alarm system to increase the safety and convenience of the driver when reversing and parking. Arduino is selected as the controller in this design. According to the ultrasonic ranging theory, the ultrasonic sensor is selected as the ranging tool, and the PC upper computer sends control instructions to operate the vehicle driving. When the distance from the obstacle is less than the safe distance, the system will take emergency braking. Finally, through simulation test, it was found that the system could not only achieve relatively accurate ranging from 5cm-100cm in the positive horizontal direction, but also serve as a reminder to obstacles with regular shapes behind the vehicle body. This design has the advantages of low cost and high integration, which is very helpful to assist the driver to reverse, so it has certain theoretical value and practical value.

**Keywords** Arduino; parking sensor; ultrasonic ranging

# 目 录

摘 要.....	I
ABSTRACT.....	II
1 绪论.....	1
1.1 ARDUINO 的发展与使用情况阐述.....	1
1.1.1 Arduino 现状研究.....	1
1.1.2 Arduino 的可扩展性研究.....	1
1.1.3 外设的可利用性研究.....	2
1.2 倒车雷达的现状研究.....	2
1.2.1 驾驶倒车现状.....	2
1.2.2 倒车雷达现状.....	2
1.3.本课题的来源.....	3
2 倒车雷达系统设计.....	4
2.1 硬件结构.....	4
2.1.1 超声波模块.....	4
2.1.2 直流电机模块.....	6
2.1.3 电机驱动模块.....	7
2.1.4 蓝牙控制模块.....	8
2.1.5 供电系统.....	9
2.2 微机接口.....	10
3 倒车雷达软件设计.....	11
3.1 系统功能模块的划分.....	11
3.2 系统运行流程图.....	12
3.3 系统详细设计.....	13
3.3.1 车辆行驶功能设计.....	13
3.3.2 车辆减速功能设计.....	14
3.3.3 感知功能设计.....	15
3.3.4 通信功能设计.....	16
3.4 PC 端软件功能实现.....	17
3.4.1 通信功能设计.....	19
3.4.2 倒车雷达实时动画制作.....	19
4 倒车雷达的安装与调试.....	22
4.1 电路连接图.....	22
4.2 倒车雷达实体图.....	23
4.3 PC 端上位机运行图.....	24
5 结 论.....	26
6 谢 辞.....	27
7 参考文献.....	28



# 第1章 绪论

## 1.1 ARDUINO 的发展与使用情况阐述

### 1.1.1 Arduino 现状研究

Arduino 诞生于 2005 年，是一款以 AVR（Advanced Virtual RISC）单片机为核心控制器的单片机应用开发板和软件 Arduino IDE（Integrated Development Environment），是一种便捷灵活、方便上手的开源电子原型平台。Arduino 开发人员提供的简洁的 Arduino IDE 是一款十分方便用于写代码、编译、调试和下载的上位机软件，可以在 Windows、Macintosh OS X、Linux 三大主流操作系统上运行，所以 Arduino 可以实现跨平台开发。同时，Arduino 开发人员开发了简单的函数和众多的应用库，因不用直接操作寄存器，所以使得没有很好的单片机基础的人员也可以使用 Arduino 完成快速开发。

由于硬件电路图开源、编写程序语句简洁、编程调试环境便捷、库函数丰富的原因，原型开发平台 Arduino 被广大设计者应用于快速完成原型开发。因为 Arduino 的种种优势，越来越多的软件开发者使用 Arduino 进入硬件、物联网等开发领域来开发相关的项目、产品。

### 1.1.2 Arduino 的可扩展性研究

Arduino 采用的 Creative Commons 许可证，不需要付版权费用即可在原开发板上完成重新设计与二次开发<sup>[1]</sup>。此外 AVR 处理器采用了 RISC（Reduced Instruction Set Computer）技术，这一系列的处理器都是集成到单个芯片的独立的计算机。对于任何一种小型控制或监视应用来说 AVR 处理器都是理想的选择。它们包含有一组内置的片上外设，还可以在片外扩充附加功能，所以价格相对低廉划算。

Arduino 具有丰富的接口，可以利用多种通信机制对周边 I/O 设备进行编程<sup>[2]</sup>进而完成逻辑控制，且很多常用的 I/O 设备都已经带有库文件或者样例程序，在此基础上进行简单的修改，即可编写出比较复杂的程序，完成功能多样化的作品。如可以快速使用 Arduino 与 Adobe Flash，Processing，MaxMSP，SuperCollider 等软件结合，做出互动作品，也可以使用现有的电子元件例如：开关、传感器、其他控制器件、发光二极管、步进马达、其他输出装置，开发出更多令人赞叹的作品。考虑到 Arduino 具有低成本可易开发的优点，Arduino 会有广阔的市场和受用人群。

### 1.1.3 外设的可利用性研究

在电子技术飞速发展的今天，积累了众多性能优良、功能众多的模块，且配备了丰富的相关文档。而 Arduino 可以很好的和这些模块完成交互，如温湿度检测、距离判断、光检测和烟雾检测等，不必为了适配 Arduino 开发板而去重新开发模块。因为重新开发已有的子模块只会增加多余工作量完成重复的工作，提升了开发成本。而 Arduino 可以很好的利用已有模块，借助前人的智慧快速完成产品原型的开发<sup>[3]</sup>。因此系统开发的重点转变为软件开发，即在 Arduino 的控制下，各个模块之间的逻辑控制、功能实现与数据交互。

且系统的稳定运行离不开软件的设计，而本次设计的软件开发采用了 C++ 语言完成软件系统的开发，具有丰富的库函数和面向对象编程的优势，可以简洁快速的完成程序开发工作；采用 Python 语言借助 Qt 接口可以完成 PC 端上位机的开发。此外，选用的模块具有丰富的文档，合理利用文档便可掌握如何使用模块及发挥功能，可完成 Arduino 控制小车与衔接模块的功能。

## 1.2 倒车雷达的现状研究

### 1.2.1 驾驶倒车现状

随着司机和车辆数量的迅速增长，道路变得越来越拥挤，停车条件也随之越来越困难。司机开车的难度更是增加很多，尤其是在特定的环境里，如黑夜、道路拥挤、大雾和地面环境恶劣的场所。在车内的司机不能及时和全面的了解汽车外部面临的各种情况，尽管所有的车辆都配备了后视镜装备，但并不能完全解决这方面的问题。逆向停车是一个复杂的系统，倒车成功不仅取决于司机成熟的驾驶经验和技能，更取决于司机判断的灵敏度。但倒车雷达大大降低了驾驶员的体力和脑力消耗，逆向停车的司机，只需要坐在驾驶座上，就可以了解车辆周围的环境，为在视线受阻、地面环境恶劣的情况下倒车停车提供了很大的安全性，从而提高行车安全，降低司机很多安全隐患。

### 1.2.2 倒车雷达现状

倒车雷达发展至今经历了五个阶段，第一阶段为开始倒车时以大分贝喇叭提示行人注意避让；第二阶段的产品使用蜂鸣器进行警报提示；当与障碍物距离很近时发出警报；第三阶段的产品借助数码管显示与障碍物的距离；第四阶段的产品借助液晶屏显示与障碍物的距离，第五阶段<sup>[4]</sup>的产品借助超声测距，提高了对障碍物距离判断的准确性。而现阶段将车载娱乐系统和第四阶段的液晶屏幕、第



五阶段的超声测距相结合，汽车厂商为了使车主在倒车时能够更加安全、方便，于是在车载娱乐系统上集成了由倒车雷达、静态倒车辅助线、动态倒车辅助线、倒车影像、全景影像等功能组成的倒车辅助系统，但价格较为昂贵。至此，可以分析出倒车雷达是否稳定的主要因素取决于视频和声波两方面。

### 1.3. 本课题的来源

本课题来源于生活实际，用在辅助驾驶中以解决倒车过程中存在的安全隐患。如今，汽车的数目日渐增多，人们在日常生活中倒车停车的次数也是越来越多。因此，当倒车停车的时候，人们对汽车倒车停车系统的安全性和便捷性要求也逐步增加。现在的小汽车一般都配备有倒车影像，但有的车辆工作环境恶劣，使用倒车影像并不现实。即便是有倒车影像，由于影像镜头一般采取广角设计，画面会发生扭曲<sup>[5]</sup>。对于缺乏距离感的人来说，想要安全倒车依然有困难，这时配备一款超声波倒车雷达就显得很有必要。而新司机驾驶技术令人堪忧，尤其是在停车过程中，不能及时有效地处理避障情况，往往损害到汽车本身，甚至危及其他人的安全。汽车倒车时需要检测驾驶员背后的驾驶区域，为驾驶员提供有效的提示，从而避免汽车的损坏和危险以及其他车辆的安全状况。

## 第 2 章 倒车雷达系统设计

### 2.1 硬件结构

在实现倒车雷达系统的过程中，硬件系统将作为软件正确运行的载体。软件的成功运行必然离不开 Arduino 与各类传感器的连接、逻辑控制和分工合作。如选择直流电机带动轮胎的旋转来模拟实际车辆轮胎的转动，选用蓝牙模块传输数据作为驾驶员发送指令数据和 Arduino 传送距离信息的媒介等。本次设计的倒车雷达硬件系统结构框图如图 2.1 所示：

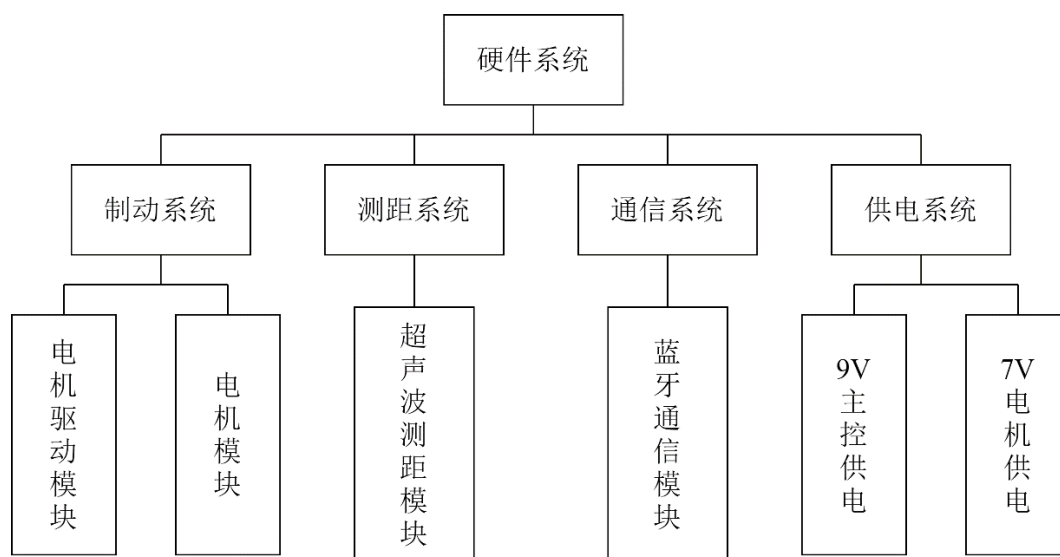


图 2.1 倒车雷达系统硬件框架图

在图 2.1 所示的硬件系统结构框架图中，清晰的展示了倒车雷达硬件系统的硬件逻辑和结构：（1）制动系统由电机模块和电机驱动模块组成，负责驱动车辆完成前进、后退和转向等功能；（2）测距系统由超声波模块组成，负责实现障碍物的检测和测量与障碍物距离的功能；（3）通信系统由蓝牙模块组成，负责数据的收发功能，即接受 PC 端上位机发来的控制指令数据，将超声波的测量的距离数据发送回 PC 端；（4）供电系统由 7V 和 9V 两种电池组成，因为 Arduino 的输出电压无法驱动电机，所以制动系统需要单独的供电系统。因此供电系统分为两部分，分别为 9V 电池负责给主控芯片供电，和两组电压 7V 的电池负责给电机模块供电。

#### 2.1.1 超声波模块

为使用超声波测量与障碍物的距离，选用 HC-SR04 超声波模块，本模块性能稳定，测度距离精确，能和国外的 SRF05，SRF02 等超声波测距模块相媲美，

可应用于距离测量、机器人控制和防盗装置等。

HC-SR04 具有两个超声波传输口,分别为 T 和 R。对 Trig 管脚赋予高电平,即为逻辑数字 1,并保持 10us 以上再赋予低电平,即为逻辑数字 0。此时会触发 HC-SR04 的开始信号,并开始如下的工作流程:由其中的 T 声波传输口发送 8 个 40khz 的方波;在发送完毕后赋予 Echo 引脚高电平 1,此时另一个声波传输口 R 开始接收 T 口发送的超声波,在接收结束后自动赋予 Echo 引脚低电平 0。此时计算 Echo 引脚保持高电平的时间,而计算结果代表了 T 口发送的这段超声波在当前位置和障碍物之间的往返时间。因为是往返时间,所以计算与障碍物的距离时需要减半来变成单程的时间,再将单程时间乘以声速即 340m/s,就会得到从当前位置到对面障碍物的距离。HC-SR04 的工作原理示意图如图 2.2 所示。

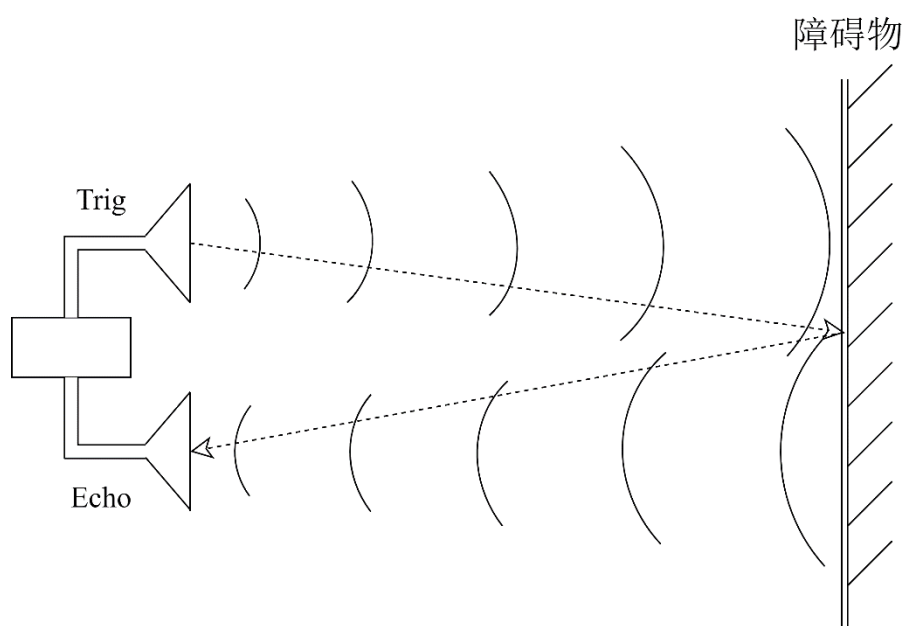


图 2.2 超声波工作原理示意图

在了解完硬件的逻辑后,需要了解硬件的工作参数才能对实物进行连接和控制,得到的 HC-SR04 超声波模块工作参数如表 2-1 所示:

**表 2-1 超声波模块工作参数表**

工作参数	参数阈值
使用电压	5V
工作电流	15 mA
感应角度	不大于 15°
探测距离	2cm 到 450cm
工作频率	40Hz

在了解完 HC-SR04 的工作原理与工作参数后,也很容易得到 Arduino 与 HC-SR04 的控制逻辑与连接方式。HC-SR04 共四个引脚,分别为: VCC, GND, Trig

和 Echo。将 HC-SR04 的 VCC 引脚连接到 Arduino 的+5V 引脚，以此来实现 Arduino 给 HC-SR04 供电的功能；将 HC-SR04 的 Trig 引脚连接到 Arduino 的 8 到 13 号的任意引脚作为信号的出口，拉高对应引脚的电平即可触发 HC-SR04 的开始信号；将 HC-SR04 的 Echo 引脚连接到 Arduino 的 8 到 13 号任意引脚作为信号的入口，检测对应引脚的电平的变化即可计算超声波的传输时间；最后将 HC-SR04 的 GND 引脚连接到 Arduino 的 GND 引脚即可，完成电路连接的通路。

而在超声波完成一次测距后，必须要延时一秒左右才能开启下一次的测距功能，因为程序的执行速度和 HC-SR04 的实体硬件执行速度并不在一个数量级。且程序执行的速度要远快于硬件的执行速度，因此在程序执行完毕后需要等待硬件执行完毕，否则在没有延时函数的条件下 HC-SR04 将无法正常工作。

### 2.1.2 直流电机模块

直流电机是一种将直流电能转换为机械能的转动装置。为模拟车辆的运行，选用直流电机带动轮胎完成车辆前进、后退和转向等功能。

直流电动机的基本构造包括电枢、场磁铁、集电环和电刷。其中电枢是将软铁芯缠绕多圈铜线线圈并可以绕轴心转动的装置；场磁铁用于产生磁场，通常由强力永久磁铁或电磁铁组成；集电环是随线圈转动，可供改变电流方向的变向器，当集电环每转动 180 度时，线圈上的电流方向就会改变一次；电刷通常使用碳制成，将集电环和固定位置的电刷接触，则电机便接通电源开始旋转。直流电机结构示意图如图 2.3 所示：

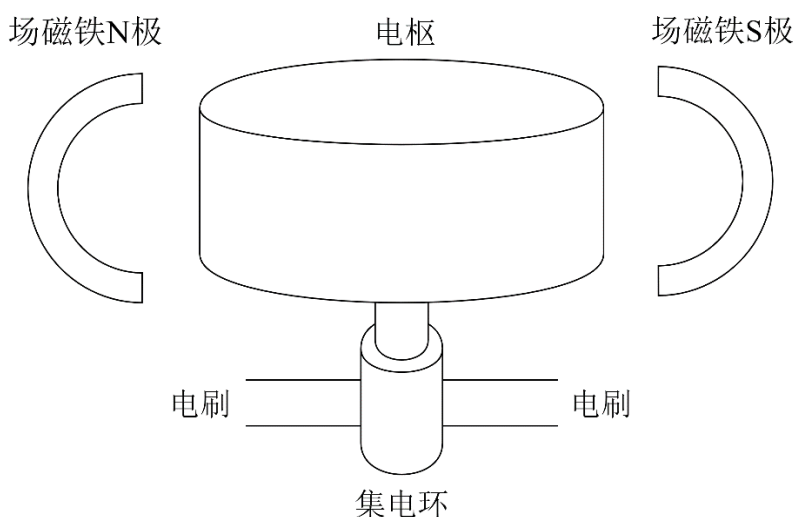


图 2.3 直流电机结构示意图

当使用直流电对电刷供电后，电枢周围产生磁场，带动转子的左侧被推离左侧的磁铁，并被吸引到右侧的磁铁，从而产生转动。转子依靠惯性继续转动，当转子旋转角度为 180 度的倍数时，电流变换器将线圈的电流方向逆转，线圈所产

生的磁场亦同时逆转,使这一过程得以重复,进而达到旋转的功能<sup>[6]</sup>。之后将电机连接轮胎,从而带动轮胎的旋转,实现车辆的制动功能。

本次设计选用的电机模块为 HC02-48,电机的工作参数如表 2-2 所示:

**表 2-2: 直流电机工作参数表**

工作参数	参数阈值
工作电压	3V 到 6V
工作电流	150mA
减速比	48:1
3V 空载转/分钟	125
扭力	0.8kg · cm

据 HC02-48 的工作参数表明,在最小驱动电压 3V 的情况下,车速能满足对车辆完成前进和后退等控制要求。在根据工作参数中的扭力<sup>[7]</sup>进行计算,若此时直接连接半径为 3cm 尺寸的轮胎,则经由车轮所发挥的推进力量约为 0.27kg,则两个电极发挥的推进力量约为 0.54kg。因此,可以根据车身的重量预估所需电极的数量,在车身重量不超过 0.54kg 的情况下,选用两个电极即可。

但是 Arduino 不能直接驱动 HC02-48 电机实现以上功能,因为驱动电机时需要提供较大电流。而 Arduino 的核心功能在于高低电平的控制而不是电流的输出。Arduino 的 IO 口输出电流在 40mA 左右,因此无法驱动最低电流要求为 150mA 的电机。所以,只能采用外接驱动芯片来放大驱动的方式。至此,可以推断出驱动芯片的最低条件,即驱动芯片的输出电压和输出电流满足电机启动的最小需求。

### 2.1.3 电机驱动模块

经过不断的比对和筛选驱动芯片的参数,最终选用的电机驱动模块为 L298N 模块。其工作参数如表 2-3 所示:

**表 2-3: L298N 工作参数表**

工作参数	参数阈值
逻辑部分输入电压	6.5V 到 12 V
驱动部分输入电压	4.8V 到 35V
逻辑部分工作电流	小于等于 36mA
驱动部分工作电流	小于等于 2A
控制信号输入电平	2.3V 到 5V
控制信号输入电平	-0.3V 到 1.5V
工作温度	-25℃到 130℃

L298N 有单独的供电引脚,此外还有四个入口控制引脚,四个出口控制引脚,

可同时控制两个电极。在表 2-3 中，可以了解到逻辑部分的入口输入电压最低需求是 6.5V，而 Arduino 的供电电压为 5V，因此需要增加额外的电池来完成对 L298N 模块逻辑部分供电的功能，而使用 Arduino 自身引脚的电压值即可为 L298N 的驱动部分提供电压输入。此外，在分析 L298N 的工作参数列表时，可以发现 L298N 能够驱动任何额定电压在 4.8~35V 和电流小于 2A 范围内的直流电机，而 HC02-48 电机的工作电压在 3-6V，电流也小于 2A。所以 L298N 能够完好的驱动 HC02-48 电机，因此可以在硬件层面完成对电机的驱动。

将 L298N 的四个入口引脚 IN1，IN2，IN3 和 IN4 连接至 Arduino 的引脚，通过引脚的输出电平来实现控制电机转动的目的。而 L298N 模块的四个出口引脚 OUT1,OUT2,OUT3 和 OUT4 均标明了正负极，此时将 Out1，Out2 连接一个电机，Out3，Out4 出口引脚连接另外的电机。通过对 4 个入口引脚的电平赋值来更改出口引脚的电平输出，进而控制电机的正反转，完成前进、后退和转向的功能。此外可以通过总开关 ENA，ENB 两个使能端控制两个电机能否转动，并且可以通过 PWM 形式控制对使能端的电压输入，达到车辆加速减速的目的。使能端，IN1，IN2，IN3 和 IN4 的赋值情况与车辆运行情况的真值表如表 2-4 所示：

表 2-4：车轮转动真值表

直流电机状态	ENA/ENB	IN1/IN2	IN3/IN4
停止	0	X	X
制动	1	0	0
正转	1	1	0
反转	1	0	1
制动	1	1	1

而电机的转动需要分别给电机的两个电刷赋予高低两个不同的电平，以实现电机的旋转。若不借助驱动芯片，一个电机占用 Arduino 的两个 IO 口，两个电机占用 4 个 IO 口。而一个 L298N 驱动芯片需要 4 个 IO 口的接入实现对两个电机的控制，因此尽管额外接入了 L298N 模块，但并没有造成 IO 口资源的浪费。

## 2.1.4 蓝牙控制模块

为测试倒车雷达的功能是否稳定，需要合理的模拟车辆运作。本次设计选择的方式为通过 PC 端上位机发送指令来控制车辆的前进、后退和转向功能，以此来模拟驾驶员驾驶车辆的行为。而 Arduino 端需要准确的读取 PC 端发送的数据，为此选用 HC-05 蓝牙模块实现数据读取功能。

HC-05 模块的主要引脚为：KEY 使能端、VCC 供电口、GND 地线、RXD 读取数据的入口和 TXD 发送数据的出口，并且主要通过 RXD、TXD 两引脚完成

通信功能。无线传输的接收灵敏度决定了设备能否良好的接收数据，提高信号的接收灵敏度可使无线产品具有更强地捕获弱信号的能力。随着传输距离的增加，接收信号变弱，高灵敏度的无线产品仍可以接收数据，维持稳定的连接，大幅提高传输距离。市面上普通产品的接收灵敏度一般为-85dBm，而 HC-05 蓝牙模块在 2Mbps 的条件下工作时，工作灵敏度为-85dBm，即 HC-05 能保证在短距离内以较低的误码率传输数据。

HC-05 蓝牙模块的工作参数如表 2-5 所示：

**表 2-5：HC-05 工作参数表**

工作参数	参数阈值
波特率	9600
通信距离	10 米
工作电压	3.2-6V
工作频段	2.4G
空中速率	2Mbps
工作电流	40mA
接收灵敏度	-85dBm

将 HC-05 蓝牙模块的 RXD 端连接 Arduino 的 TXD 端，将 HC-05 蓝牙模块的 TXD 端连接 Arduino 的 RXD 端。如此连接，会使得 HC-05 蓝牙模块的数据输出口与 Arduino 的数据读入口相连接，数据读入口与 Arduino 的数据输出口相连接，构成通信的回路，这也是顺利完成两者通信的重要前提。但必须要注意的是，在给 Arduino 烧录程序时需要断开 Arduino 与 HC-05 的 RXD 和 TXD 连接，否则 Arduino 的 TXD 和 RXD 两引脚处于连接状态而非悬空状态时，Arduino 处于通信状态，程序无法通过串口烧录至 Arduino 内。

### 2.1.5 供电系统

倒车雷达系统的供电系统分为两部分，第一部分为 Arduino 供电系统，第二部分为 L298N 电机供电系统。

对于 Arduino 供电系统，选择了电源接口为 Arduino 供电。通过此方法为 Arduino 开发板供电时，直流电源电压的范围应为 9V~12V，此时仅需要一个 9V 电池和转接线即可。使用低于 9V 的电源电压可能导致 Arduino 工作不稳定，使用高于 12V 电源电压存在着毁坏 Arduino 开发板的风险。

对于 L298N 供电系统，选择的是 3.7V 18650 可充电锂电池，将两节此类电池串联在一个电池盒内即可。电池盒的要求必须是串联，而不能是并联，通过串联做到提升电压的效果。将电池盒自带的红色线与 L298N 的+12V 连接，将黑色

线与 L298N 的 GND 连接即可构成电路的回路，实现电机部分供电的功能。

## 2.2 微机接口

微机接口是指计算机与其以外的设备进行通讯时的连接方式，具体分为硬件接口和软件接口。其中的硬件接口已经完成：选用 HC-SR04 超声波模块实现测距功能；选用 HC-05 蓝牙模块实现通信功能；选用 L298N 电机驱动模块实现制动功能。虽然给出了相关硬件的工作原理，但也要分析软件和硬件之间的工作流程。即了解软件接口后，才能为实现打车雷达系统的软件设计做好基础。而最基本的软件接口为 `pinMode` 定义引脚模式和 `digitalWrite` 控制引脚的输出。

`pinMode` 通过操作对应引脚的 DDR（Data Direction Register）将引脚配置为输入模式或输出模式，其中 0 为输入模式，1 表示输出模式。而 Arduino 含有三组 DDR，分别为 DDRB, DDRC 和 DDRD。而 `digitalWrite` 的工作原理和 `pinMode` 类似，通过设置 PORTB, PORTC 和 PORTD 三组寄存器中的 bit 位来定义对应引脚的输出电平，1 为高电平，0 为低电平。这三组寄存器和控制引脚的对应关系如表 2-6 所示：

**表 2-6：寄存器与控制引脚的对应关系**

Bit 位	DDRB/PORTB 引脚	DDRC/PORTC 引脚	DDRD/PORTD 引脚
0	D8	A0	D0
1	D9	A1	D1
2	D10	A2	D2
3	D11	A3	D3
4	D12	A4	D4
5	D13	A5	D5
6	--	--	D6
7	--	--	D7

当配置引脚为输入模式时，引脚将处于高阻抗状态，此时引脚会串联一个 100 兆欧的电阻。高阻抗中的输入电阻近似无穷大，所以可以通过捕获微小的电流变化来获取外界的感知，而且在一定程度上也增加了芯片的抗电压冲击能力。通过 `pinMode` 配置为 OUTPUT 的引脚处于低阻抗状态，这意味着该引脚可以为其他电路提供大量电流，此时 Arduino 引脚可以向其他设备/电路提供（提供正电流）或吸收（提供负电流）高达 40 mA 的电流。例如，此电流足以点亮 LED 或运行许多传感器。但 Arduino 的输出电流却不足以运行大多数继电器，螺线管或电动机。



## 第3章 倒车雷达软件设计

### 3.1 系统功能模块的划分

将倒车雷达全部的功能划分为几个功能模块，解决专一的问题有利于提升开发效率。因此将本次倒车雷达系统的功能划分为两个大类，第一类为硬件层的软件设计，第二类为PC端的软件设计。第一类包括三个模块：感知功能模块、通信功能模块和控制功能模块。其中，感知功能由超声波模块构成，负责探测与外部障碍物的距离，用于给驾驶员提供辅助信息；通信功能模块由蓝牙模块和PC端上位机组成，由PC端上位机借助串口发送控制指令的数据到蓝牙模块，Arduino读取蓝牙的数据进而控制车辆完成相应指令；控制功能模块由Arduino构成，负责系统整体的运行，根据PC上位机端发送的指令进入倒车模式、控制车辆运行、紧急制动和退出倒车模式。第二类包括两个模块：显示功能模块和通信功能模块，显示功能模块负责PC端上位机实时显示倒车影像，数据通信功能模块负责发送控制指令和读取Arduino端发送的距离数据。综上所述，倒车雷达系统软件层的功能模块结构框图如图3.1所示：

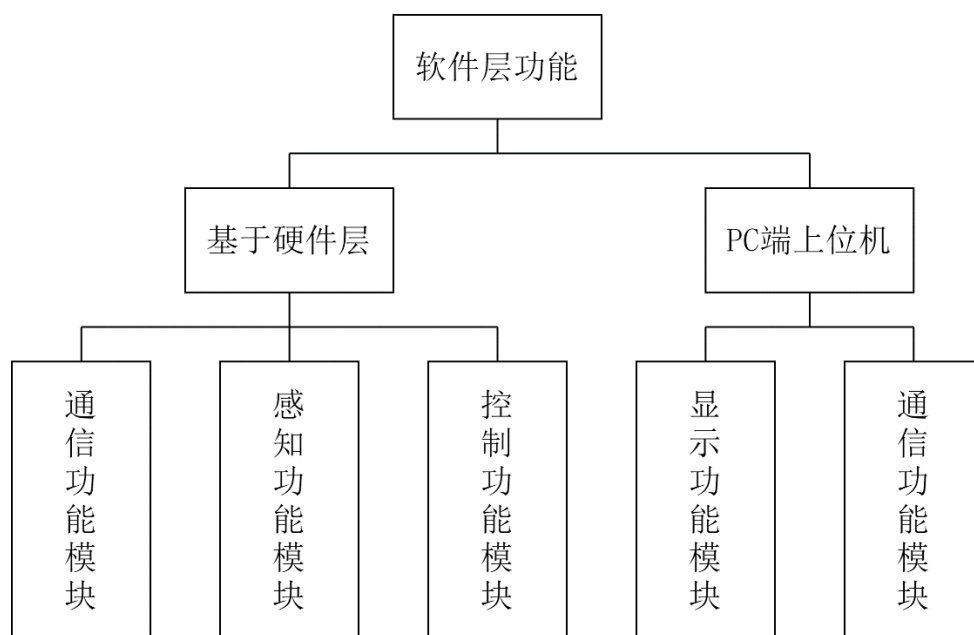


图 3.1 软件层功能模块结构框图

在驾驶用户开启倒车功能后，感知功能模块开始运转。将车辆后侧距离障碍物的距离数据通过通信功能模块反馈给用户，对用户的倒车起到一定辅助作用。当距离感知模块探测到与障碍物的距离小于安全距离时，车辆将采取紧急制动的措施而并非继续执行驾驶员发出的控制指令。而停车的另一情况为倒车结束，此时车尾与障碍物距离过近，驾驶员可确认停止倒车，此时倒车正式结束。

### 3.2 系统运行流程图

在倒车开始时，由超声波测量距离并给予驾驶员提示，而在测量与障碍物的距离过紧到达危险阈值后，由系统强行制动保证车辆和人身安全<sup>[8]</sup>。而驾驶员意识到倒车结束后，可手动解除倒车，退出倒车模式。待到下次倒车开始时，驾驶员可以选择是否开启倒车雷达。在驾驶员正常驾驶的情况下，或者驾驶员不开启倒车雷达的情况下，系统将关闭倒车雷达的超声波模块和制动中断等功能。一方面，降低系统功耗，节约资源保护环境；另一方面，在正常驾驶的过程中，难免遇到堵车等情形，而不关闭倒车系统的话，可能会造成车辆的意外停车现象<sup>[9]</sup>，带来潜在的交通隐患。因此只有确认进入倒车模式时才开启超声测距功能，而在倒车结束后需要关闭超声测距功能。根据对系统功能的分析，得到车辆紧急制动时的运行流程图，如图 3.2 所示。

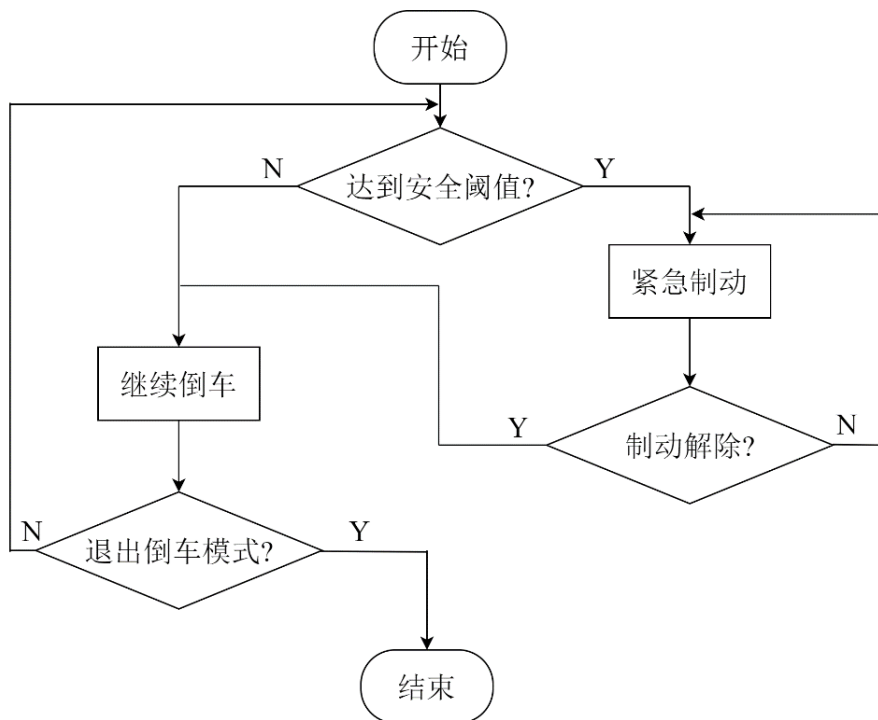


图 3.2 系统制动流程图

此外需要格外注意的是，倒车时的车辆行驶速度会远远低于正常驾驶的速度以保证倒车安全。因此在进入倒车模式时会利用 PWM (Pulse width modulation) 技术将车辆的行驶速度调制到低档；与此同时，尽管车辆在低速移动，但当车尾距离障碍物越近则认为越危险，当达到某危险阈值时，系统应立刻紧急制动，停止此时的倒车行为。

### 3.3 系统详细设计

#### 3.3.1 车辆行驶功能设计

倒车雷达系统中最基础的功能为驾驶员驾驶车辆进行前进、后退、转向等车辆移动功能。因此，首先需要完成 Arduino 控制直流电机实现正反转的设计。从物理角度出发，当左右两侧的两个车轮同时向前旋转时，车辆会前进；当两个车轮同时向后旋转时，车辆会后退；可以推理出在车辆向右转向时，左侧车轮需要向前移动，右侧车轮需要向后移动，其中右转向的示意图如图 3.3 所示。

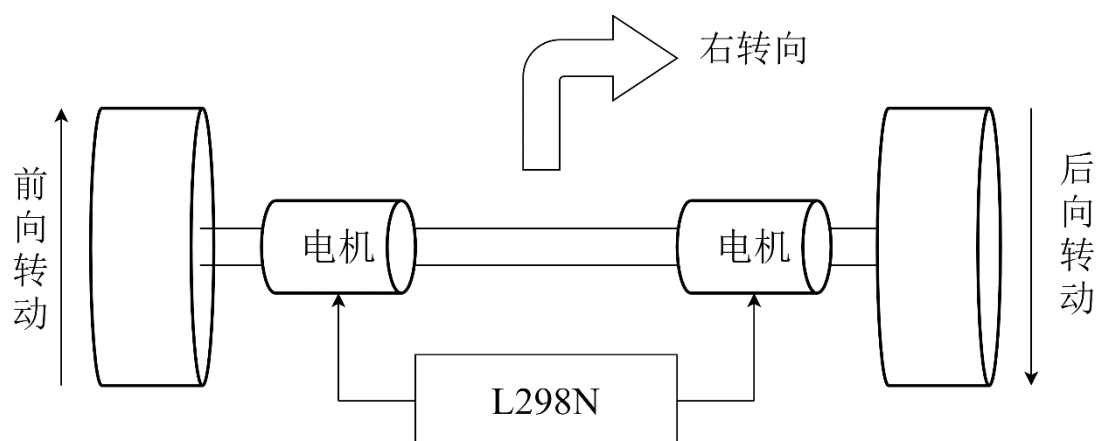


图 3.3 右转向示意图

同理，当左侧车轮向后转动，右侧车轮向前转动时，此时车身将向左转向。而在硬件层面，实现车轮的前向转动与后向转动的方式为更改电机两端电刷的正负电压即可。最终对应到软件层面，更改车轮前后向转动的方式为更改 L298N 的出口引脚的电平。将 Arduino 的四个引脚 5, 6, 9, 10 接入 L298N 的四个入口引脚，并将 5, 6, 9 和 10 这四个 IO 口定义为输出模式，即对引脚赋值高低电平后输后到 L298N 模块，由 L298N 模块的入口引脚读入电平，解析指令并执行相应动作。对于两个电机 M1 和 M2，如 L298N 的出口引脚 Out1 和 Out2 分别接入 M1 电机的正极和负极，另外的出口引脚 Out3 和 Out4 接入 M2 电机的正极和负极。则实现右转的伪代码如下：

```
pinMode(5, OUTPUT); //初始化各 IO,模式为 OUTPUT 输出模式
pinMode(6, OUTPUT); pinMode(9, OUTPUT); pinMode(10, OUTPUT);
digitalWrite(input1, LOW); // 左轮前转动 digitalWrite(input2, HIGH);
digitalWrite(input3, HIGH); //右轮后转 digitalWrite(input4, LOW); delay(1000);
```

在伪代码中，可以明确看到先由 Arduino 的库函数 pinMode 声明引脚的逻辑为输出模式，即可以向外部的 L298N 发送数据。再由库函数 digitalWrite 向引脚发送高低电平，其中 LOW 的含义为低电平，HIGH 的含义为高电平，而电流会

由高电平一侧流向低电平一侧，以此来带动电机旋转。而 L298N 将输入引脚的高低电平信号进行解读，并在输出引脚进行输出对应的逻辑，则电机便可以按照 L298N 模块的输出进行转动，以此来带动车辆的运转。

### 3.3.2 车辆减速功能设计

而在倒车过程中，最重要的为减速慢行。因此需要设置控制车辆减速慢行的功能模块。而电机旋转的快慢与直流电源提供的电压值大小有关，由于计算机不能输出模拟电压，且 L298N 的供电口只能输出固定大小的电压值，因此需要使用脉宽调制技术来降低 L298N 的输出电压。

脉宽调制技术是最常用的调速的工具，脉冲宽度调制（PWM）是一种对模拟信号电平进行数字编码的方法<sup>[10]</sup>。PWM 的控制方式是对逆变电路开关器件的通断进行控制，使用更改方波的占空比的调制方法使输出端得到一系列幅值相等的脉冲，用这些脉冲来代替正弦波或所需要的波形，达到对一个具体模拟信号的电平进行编码的目的。即在输出波形的一个周期中按一定的规则对各脉冲的宽度进行调制，即可改变逆变电路输出电压的大小，也可改变输出频率。但 PWM 信号仍然是数字的，因为在给定的任何时刻，满幅值的直流供电只可能是最大值（供电状态）或者为 0（电源关闭）。

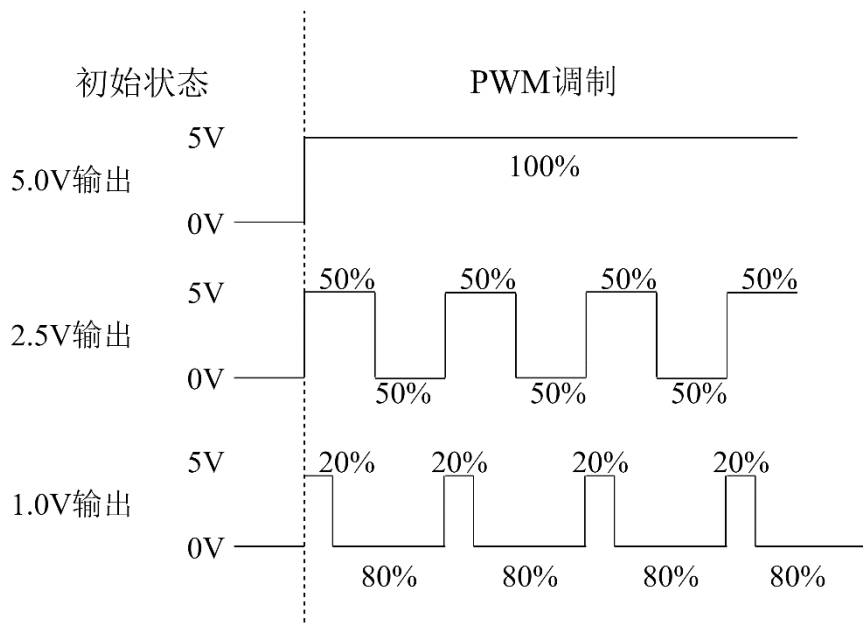


图 3.4 不同输出电压占空比示意图

不同电压的方波占空比示意图如图 3.4 所示。在软件设计中使用 PWM 技术时，仅了解需要输出电压的大小即可设置数值。以高电平输出 5V 为例，在输出 5V 的电压时，只需一直输出高电平即可；输出 2V 的电压时，需要在一个周期内（一个高电平和一个低电平为一个周期）保证有 20% 时间输出高电平，80% 时间

输出低电平即可；同理，在输出 2.5V 的电压时，仅需要在一个周期内 50% 时间输出高电平，50% 时间输出低电平即可。

而 Arduino 提供了 5 个 PWM 调制引脚口，分别为 3，5，6，10 和 11 这五个引脚口。在得到对 PWM 的客观认知后，同样可以借助 Arduino 的库函数简易实现电压的调制输出。首先将标号为 5 和 6 的口作为输出口，伪代码如下所示：

```
int leftPWM = 5; int rightPWM = 6; // 定义 PWM 口
pinMode(leftPWM, OUTPUT); pinMode(rightPWM, OUTPUT); // 指定模式
```

而借助 Arduino 的 analogWrite() 函数便可以实现 PWM 的输出。函数的第一个参数需要传入要调制的引脚，第二个参数为占空比，取值范围为 [0, 255]。0 对应 OFF，255 对应 ON，则数值 204 表示 80% 的占空比，即在 5V 为最高输出电压的背景下输出 4V 的电压时，伪代码如下所示：

```
analogWrite(leftPWM, 204); analogWrite(rightPWM, 204); // PWM 输出
```

### 3.3.3 感知功能设计

感知功能是倒车雷达距离探测的基础，由超声波模块组成。感知功能负责在倒车过程中探测车辆与后方障碍物的距离，并将距离数据反馈给驾驶员。

为了提升安全系数，将距离声明为 float 类型的变量，把距离精度提升到小数级别，而在反馈距离时，仅需要选择保留小数点后一位的精度即可。在上一章节的硬件设计中了解到 HC-SR04 的工作模式为：将 Trig 赋予高电平 10us 以上后将 Trig 赋予低电平，之后赋予 Echo 高电平并开始计时，Echo 接收到反馈声波，变为低电平时结束计时。因 Trig 引脚向外发送数据所以定义为输出模式，Echo 引脚接受数据定义为接收模式。通过时间计算与障碍物的距离，伪代码如下所示：

```
pinMode(trigPin, OUTPUT);           // 定义 Trig 为输出模式
pinMode(echoPin, INPUT);            // 定义 Echo 为输入模式
digitalWrite(trigPin, LOW);          // 初始化 Trig 引脚
delayMicroseconds(5);               // 延时 5 微秒
digitalWrite(trigPin, HIGH);         // 拉高 Trig 电平
delayMicroseconds(10);              // 持续 10ms
digitalWrite(trigPin, LOW);          // 拉底 Trig 电平
duration = pulseIn(echoPin, HIGH);   // 记录 Echo 端接受声波的时间
cm = (duration/2) / 29.1;            // 将时间转换为距离
```

### 3.3.4 通信功能设计

通信功能是实现 Arduino 和 PC 端上位机交互的重要枢纽，其主要由蓝牙模块实现。在模拟驾驶员驾驶车辆时，选取的方式为 PC 端上位机连接蓝牙，通过串口发送指令来控制车辆的移动。在将蓝牙模块与 Arduino 按第二章的方案正确连接后，即可完成通信功能，其示意图如图 3.5 所示：

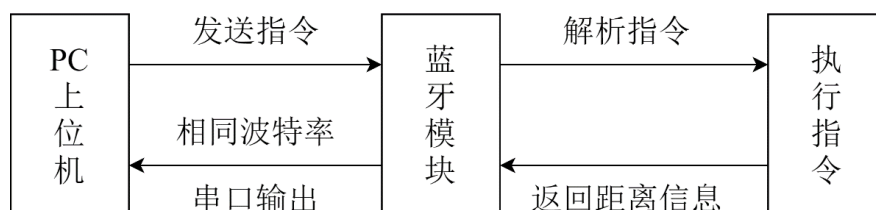


图 3.5 通信功能示意图

在图 3.5 中，可以清晰的观察到蓝牙模块在处于数据通信的交通枢纽，负责 PC 上位机端与车辆的数据通信。而 PC 端发送的指令集如表 3-1 所示：

表 3-1：发送端指令集

指令内容	指令类型	含义
“0”	字符型	车辆前进
“1”	字符型	车辆后退
“2”	字符型	车辆停止
“3”	字符型	车辆左转
“4”	字符型	车辆右转
“5”	字符型	解除倒车模式
“6”	字符型	开启倒车模式

在车辆的正常驾驶中，车辆需要不断获取 PC 端上位机发送的指令，进而开始下一步行动。因此应将蓝牙的通信功能作为主函数中的主循环部分，通过调用 available()方法时刻接收 PC 上位机端发出的指令。以 loop 函数为主循环，并检测蓝牙端是否有数据传输，伪代码如下所示：

```

void loop()                // loop 是 Arduino 执行函数的主循环
while(Serial.available())  // 判断 PC 端是否发送数据
    char c=Serial.read();   // 当 PC 端有发送的数据时，读取
    
```

而在这个过程中，通信的网络协议可保证数据以透明形式在发送端与接收端进行。以 PC 上位机串口发送数据为例，由应用层规定传输数据的格式，经由传输层形成 UDP（User Datagram Protocol）或 TCP（Transmission Control Protocol）报文经过端口传输至网络层，因 PC 连接好 Arduino 端的蓝牙后可捕获蓝牙模块的 MAC(Media Access Control Address)地址且默认在同一子网内，通过 CSMACD



（Carrier-sense multiple access with collision detection）广播协议<sup>[11]</sup>将数据帧通过数据链路层进行发送，且数据帧内含有检验信息，而后数据帧通过 2.4G 频段的物理层传播至接收端的数据链路层。

接收端由数据链路层解析数据并传播至网络层。首先解析物理层传播的数据，将电信号转换为具体数值，且仅 MAC 地址相匹配才能接受发送的数据帧。经传输层到达应用层，而后处理应用层的数据。其中，数据链路层、网络层、传输层具有完备的数据检验方式和差错处理机制。如在网络层、数据链路层会在数据报文或数据帧中加入校检数据，接收端可根据校检数据判断数据是否接受正确。此外，数据链路层的 CSMACD 协议用于保证广播通信时 PC 端发送的数据被指定接收端接收，而传输层的慢启动、拥塞控制和流量控制<sup>[12]</sup>等算法可保证数据以合理的速度进行传输，防止传输过快或过慢导致的网络拥堵或资源过剩现象。完整的计算机网络通信模型如图 3.6 所示：

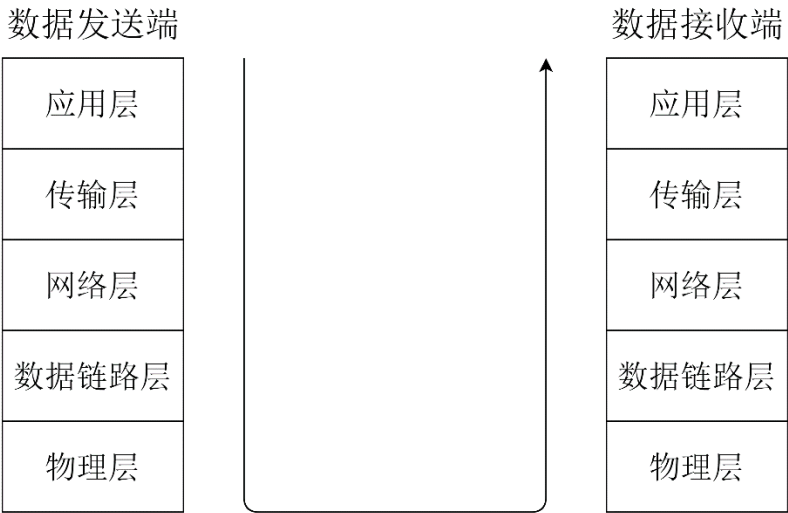


图 3.6 计算机网络通信模型示意图

将 HC-SR04 探测的距离信息的反馈给驾驶员是仍然需要通信功能模块的支持。此时数据的发送端蓝牙模块与 PC 端上位机需要设置同样的波特率。在计算机通信中，数据在物理层的传输均为电信号，而携带数据信息的电信号单元为码元，一个码元表示一个脉冲信号，波特率表示的为每秒钟传送的码元符号的个数。若发射端和接收端的波特率不同，两者会工作在不同的带宽中，且接收端将无法正确的对发送端的波形进行解调，将无法收到数据。为适配 HC-05 蓝牙模块，将接收端的波特率约定为 9600。

### 3.4 PC 端软件功能实现

在完成硬件对应的软件功能设计后，下一步的主要目标为实现 PC 端上位机相关功能，而 PC 端上位机的功能结构框图如图 3.7 所示。

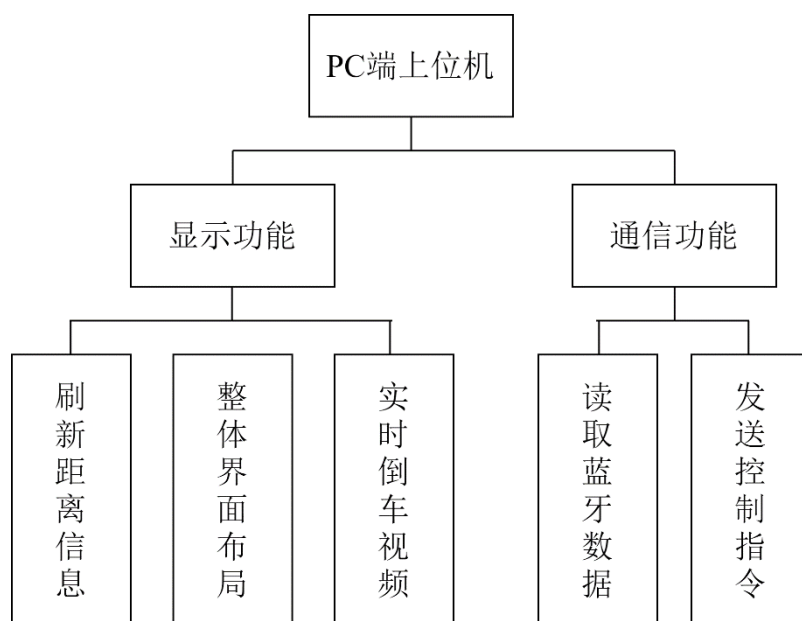


图 3.7 PC 端上位机软件功能结构框图

PC 端上位机主要负责控制车辆的运行和数据的接收与发送，选用的软件开发工具为 Qt，并绘制相关界面。在图 3.7 中，简洁的展示了 PC 上位机两大功能。分别为界面显示功能和数据的通信功能。界面显示功能主要负责维护窗口与控件的显示和正常运行，数据通信功能负责读取串口输入数据或从串口发送数据作为输出。且在开启倒车模式后，数据通信功能模块将一直开启，不断的进行数据的读取和发送，即读取距离信息与发送控制指令，直到退出倒车模式。

考虑到 Python 的多进程并不共享变量，若使用多进程操作距离信息等变量会存在诸多不便。而 Python 的多线程自带 GIL（global interpreter lock），这限制了 Python 的线程的使用。虽然理论上创建了多线程，但解释器执行代码时，即任何 Python 线程执行前，必须先获得 GIL 锁，然后每执行 100 条字节码，解释器就自动释放 GIL 锁，使得让其他的线程有机会被执行。GIL 全局锁把所有线程的执行代码都加上了限制锁。所以，多线程在 Python 中只能交替执行，即 100 个线程跑在 100 核 CPU 上，也只能用到 1 个核。至此，我们可以了解到 GIL 最大的问题就是 Python 的多线程程序并不能利用多核 CPU 的优势。但是，GIL 只会影响到那些严重依赖 CPU 的程序（如计算型程序）。而本次设计的程序大部分只涉及到 I/O，如距离信息、控制信息等数据交互程序，那么此时使用多线程就很合适，因为 I/O 程序的大部分时间都在等待而非计算。

而在 Qt 中，主线程为界面显示线程，此时若在显示线程中执行通信功能的循环，则程序会由显示线程的循环进入通信的循环且无法退出，此时显示线程无法正常运行，导致界面没有响应而卡死。因此，实现 PC 端上位机的两个功能需要创建两个线程：主线程和通信线程。即利用多线程技术，使得主线程负责界面的显示，通信线程负责数据的接收与发送，互不影响，防止只有一个线程带来的



卡死现象。因多线程可以共享同一数据变量，所以，主线程可以获取通信线程读取的距离数据，并对倒车影像中的车辆位置和形态进行调整。

### 3.4.1 通信功能设计

借助 Python 的串口工具库 serial 来实现 PC 端的数据发送和读取功能。而 serial 的发送与读取的数据均为字节类型，因此在数据发送时，需要将发送的数据转为字节类型；在接收 Arduino 发送的数据时，需要将字节类型的数据转为字符串等能处理的高级对象。

在通信之前，需要连接 PC 端与 Arduino 端的蓝牙设备，并通过对应的串口才能进行通信，其中 Windows 为 COM 口，而 Linux 为/dev/ttyUSB。在打开串口之后，设定相同的波特率，即可完成通信链路的搭建。而对于 serial 工具库，可将其实例化为一个对象，通过调用对象的发送（write）、接受（read）两个方法来实现数据的发送与读取功能。本次设计使用的 Serial 方法和属性如表 3-2 所示：

表 3-2：使用的串口方法和对应逻辑

方法与属性	描述
port='COM6'	指定对应的 COM 口
baudrate=9600	指定对应的波特率
readline()	读取以“\n”转义符结尾的数据
isOpen()	判断串口是否打开
write()	发送数据

### 3.4.2 倒车雷达实时动画制作

为了方便驾驶员在倒车期间对倒车情况的掌控，所以在 PC 端上位机加入实时的倒车影像来辅助驾驶员的倒车行为。首先创建容纳车辆实体的场景，并将其视为倒车时容纳车辆的场所，如：车库。之后，考虑到市面上绝大部分车辆的身形为矩形，所以为做到倒车雷达更广泛的应用，因此统一车身的形状，创建矩形实例来代表实际的车辆。对于 PC 端上位机的实时刷新倒车情况而言，共分为三种情况：第一种情况为前进、后退等直线移动；第二种为左转和右转等转向移动；第三种为在转向结束后，车辆按照转动的角度继续前进或后退。但无论哪种情况，均需要控制 PC 端上位机中的代表车辆的矩形实体完成对应的前进、后退的动画。

而实现动画的方案为根据通信功能所反馈与障碍物的距离信息计算矩形的坐标，根据发送的旋转指令计算矩形的旋转角度。将矩形从当前坐标移动到新坐标即可，以此来实现实时倒车动画的功能，其示意图如图 3.8 所示。

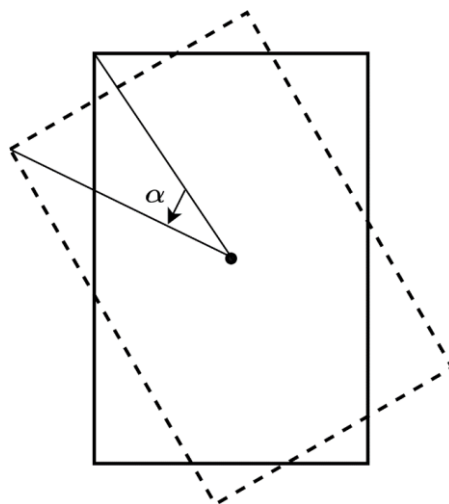


图 3.8 车身左旋示意图

对于车辆的左转与右转这种情况，PC 端上位机中的影像也需要实时的更新，此时需要根据旋转的角度来刷新车辆的位置。当实际的车辆左转向时，软件中的矩形实体也应该实时的旋转对应的角度。在图 3.8 中，当旋转角度为 $\alpha$ 时，由实线边框旋转至虚线边框。同理，可以根据左转向的方法得到右转向的方法。当车辆左转或右转后会产生一定的角度，假设旋转的角度仍然为 $\alpha$ 。如果此时再次进行倒车操作，那么车辆的运行轨迹应按照对应的旋转角度继续向后运行，如图 3.9 所示。在图 3.9 中，车辆由虚线位置移动到实线位置。设车辆由 A 点移动到 B 点的距离为 $d$ ，且开始移动时 A 点的坐标为 $(x, y)$ ，很容易得到 B 点的横坐标为 $x + dsin\alpha$ ，B 点的纵坐标为 $y + dcos\alpha$ 。在得到移动前后具体的坐标值之后，便可以很容易的实时动画中的移动功能，仅需将物体从旧坐标点移动到新坐标点即可。

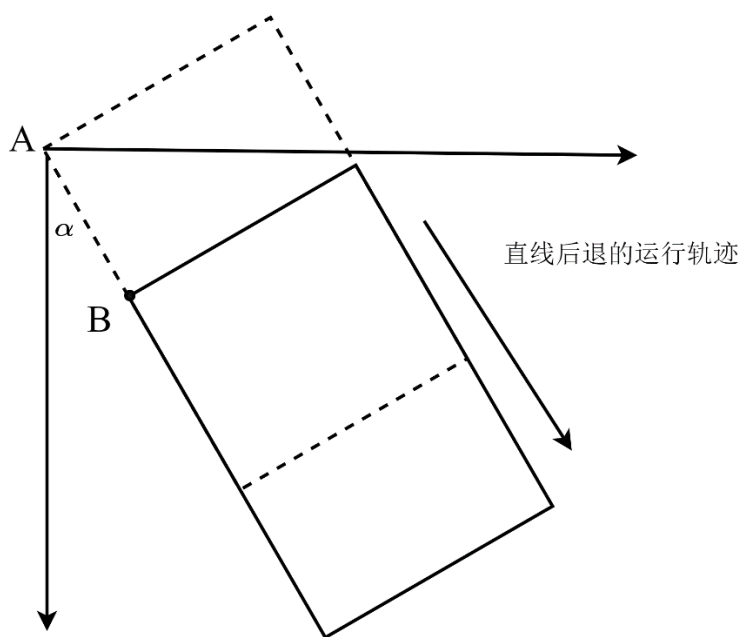


图 3.9 车身左旋后退示意图

此外，一方面而为了能让倒车雷达的影响更好的辅助驾驶员对倒车信息有所

了解，另一方面考虑到 PC 端上位机中实际场景大小的因素，设置当车辆与障碍物的距离大于 20cm 时，PC 端上位机中的矩形实体并不移动；当车辆与障碍物的距离小于 20cm 时，PC 端上位机中的矩形实体开始移动；考虑到实际距离和 PC 端上位机中的像素点对应关系，我们设定：将车辆与障碍物的距离的 10 倍作为场景移动时需要移动的像素点距离。

### 3.5 系统运行流程图

在分别介绍倒车雷达的硬件系统和软件系统完毕后，便可以得到倒车雷达系统运行的流程图如图 3.10 所示。

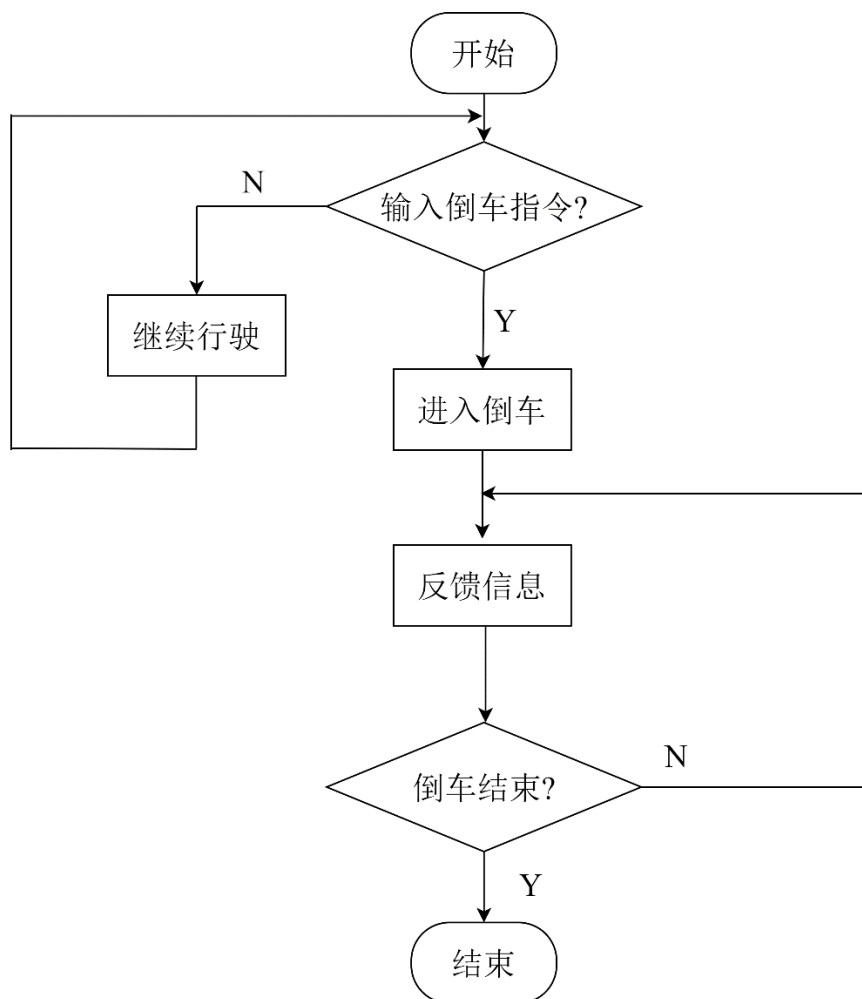


图 3.10 倒车雷达系统运行流程图

## 第 4 章 倒车雷达的安装与调试

在详细设计完各个模块的功能后，将各模块的功能加以组合运用，便能实现最终的倒车雷达系统。因此将实物与理论相结合，制作实际的倒车雷达系统。

### 4.1 电路连接图

对于测距模块,Arduino 与 HC-SR04 超声波模块的连接示意图如图 4.1 所示。

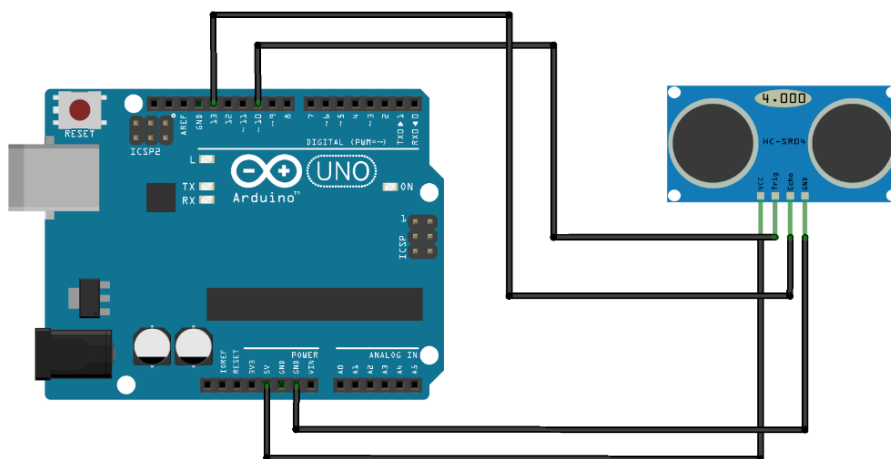


图 4.1 Arduino 与 HC-SR04 连接示意图

对于制动模块,根据 L298N 与 Arduino 的连接方式,进而得到了 Arduino 通过 L298N 驱动电机模块的连接的方式,示意图如图 4.2 所示。

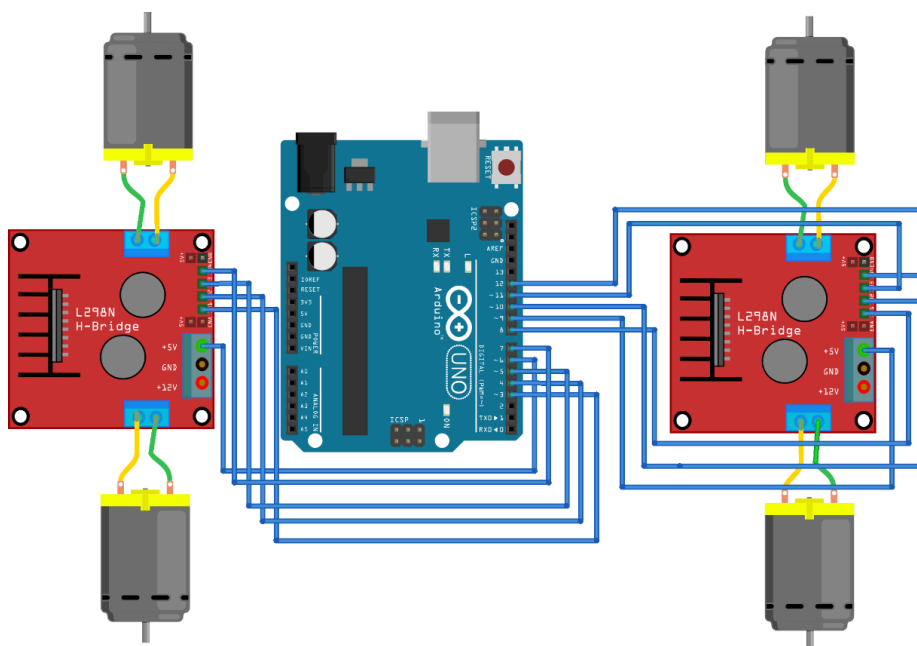


图 4.2 Arduino 驱动电机模块电路原理图

对于通信模块，Arduino 和 HC-05 蓝牙模块的连接示意图如图 4.3 所示。

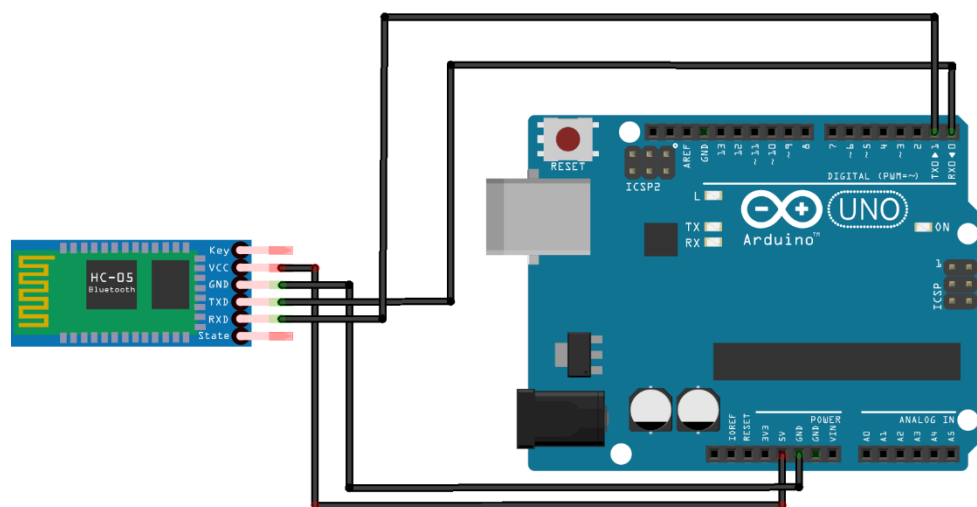


图 4.3 Arduino 与 HC-05 连接示意图

## 4.2 倒车雷达实体图

最后，按照 4.1 章节中的电路连接图，按照不同的功能模块对实体期间进行连接。为了便于展示最终的硬件实物，将实物的俯视图、侧视图进行展示。其中，侧视图如图 4.4 所示，俯视图如图 4.5 所示。

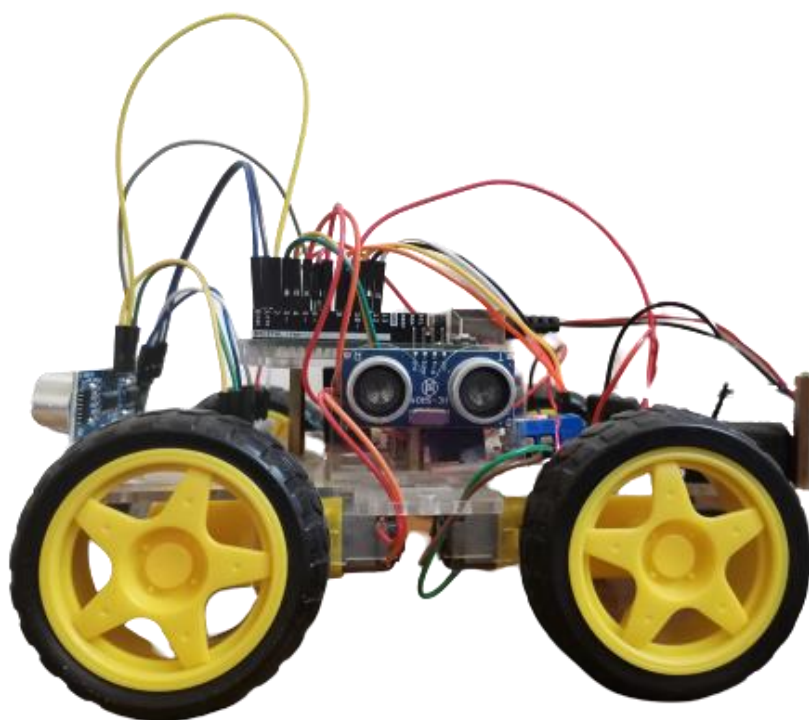


图 4.4 实体倒车雷达系统侧视图

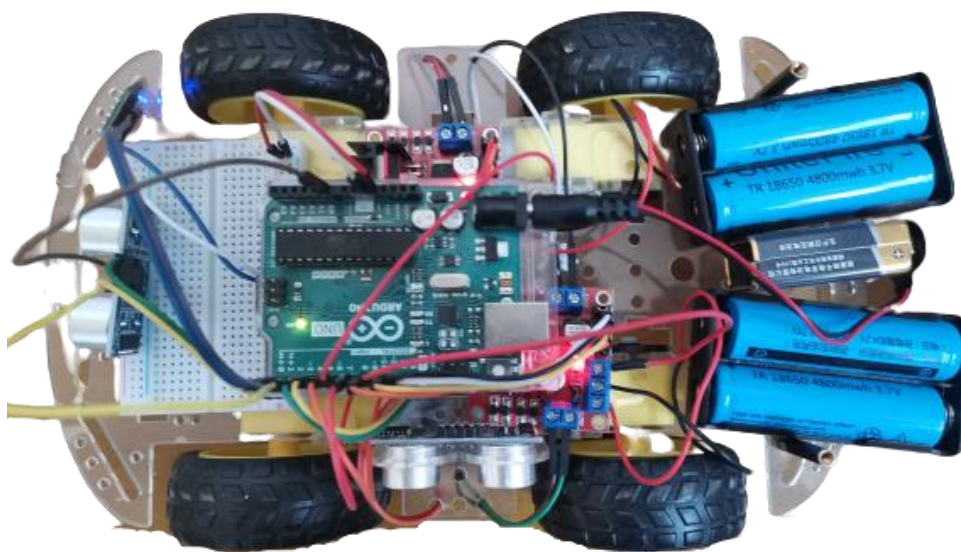


图 4.5 实体倒车雷达系统俯视图

### 4.3 PC 端上位机运行图

一个易于操作和美观的 PC 端上位机能更加方便驾驶员对车辆的控制，本着简易、美观且功能强大的目的，开发出的 PC 端上位机运行图如图 4.6 所示。

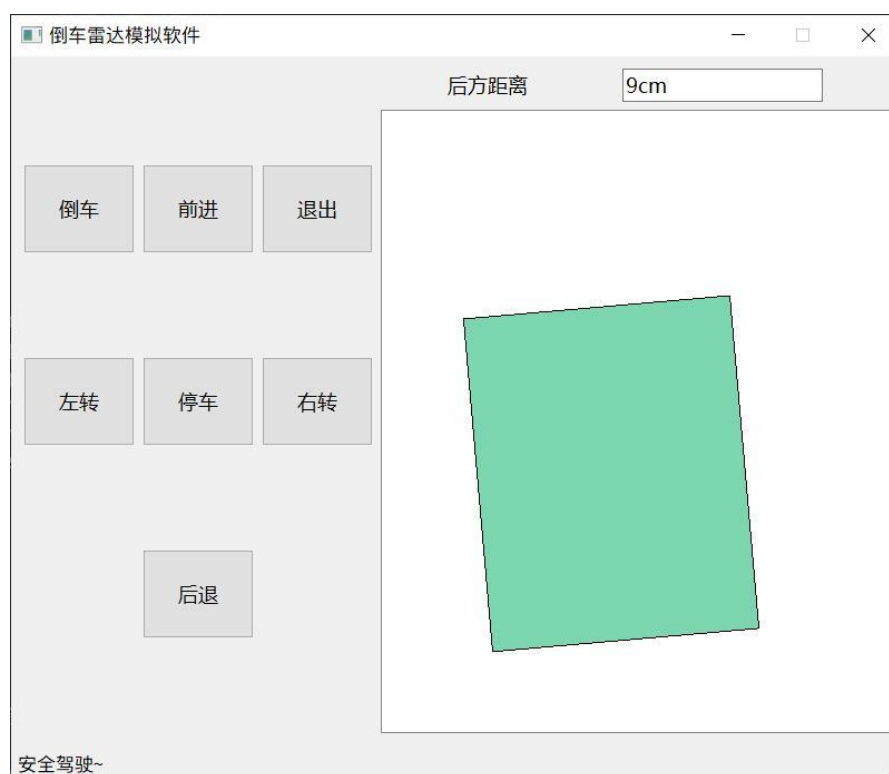


图 4.6 PC 端上位机运行示意图

如图 4.6 所示，软件的左侧为控制端按钮，倒车按钮负责控制车辆进入倒车模式；退出按钮负责控制车辆退出倒车模式；其余的前进、停止、后退、左转和右转按钮负责控制车辆的行进。软件的右侧负责实时显示倒车过程中的影像，上方通过不可编辑的文本框显示与后方障碍物的距离，使驾驶员对距离障碍物的信息更加直观，而不可编辑的文本框可以防止对距离数据的误操作；下方通过绿色的矩形来代表实际的车辆，抽象的表示更利于驾驶员掌握车辆的情况。在图 4.6 中，此时的车辆进行了左转动作，且与后方障碍物的距离为 9cm。

## 结 论

对系统的运行进行模拟实验，发现具有良好的性能。能够准确的接收 PC 上位机端发送的指令并按指令执行。在倒车期间能及时减速，精准的测量各个方位的障碍物，且遇到危险情况时能及时制动。

本次设计并实现了一种车辆的倒车雷达系统。在整个过程中，首先研究超声测距的原理和相关技术，在此基础上根据倒车操作时的具体需要而选定了具体功能与模块，包括数据传输功能、PWM调速功能和制动报警功能，最终设计出了基于Arduino控制系统的整体设计方案。系统采用了HC-05蓝牙模块用于接收与发送数据，该装置不但可以显示数字还可以显示字符，能够满足系统发送数据与接收数据的需求，以给驾驶员更多的倒车提示。为了提高系统的安全性，系统还加入了PWM调速功能，用于降低车辆倒车时的车速，且不断检测与障碍物的距离，能及时触发报警系统采取紧急制动，更加有利于保障人身安全。

通过对倒车雷达系统的研究和设计，本课题主要完成的工作有：（1）根据系统需求提出了系统的整体设计方案，同时确定了相关模块的选取与使用方法。

（2）完成了整个雷达系统的硬件设计。系统以Arduino为控制核心，控制超声波传感器测距模块，可以得到与障碍物的安全距离；通过HC-05蓝牙模块进行数据通信，保证了距离信息的可反馈与控制指令的有效性；通过电机驱动模块与电机模块共同保证了车辆的正常运行。（3）采用C++语言对系统的软件部分进行了编写，包括系统运行主程序、超声波测距程序、PC上位机接收与发送数据程序、车辆调速程序和报警制动程序，以及一些相关子程序，采用Python语言编写了PC端上位机，负责指令的发送与倒车影像的实时显示。

但本系统仍然存在可扩展的方向，包括：（1）测距精度。超声波的传播速度与温度、湿度具有一定相关度，但本系统没有考虑到温度和湿度的因素，因此整个系统没有达到理想的精度。为了使系统可以在任何温度、湿度下都能够准确的测量，可以考虑在外围电路增加温湿度传感器模块，然后控制芯片根据实测的温湿度修正声波的传输速度，从而增加系统测距的精度。（2）电磁辐射干扰。倒车雷达系统的运行环境是相对比较复杂的，如外界的电磁波可能会对系统的信号接收与发送造成不良的影响。所以在进行硬件搭建和软件设计时可以考虑采取适当的抗干扰措施，以提高系统的工作可靠性。

以上为倒车雷达系统的总结。在设计该系统的过程中参考了市场上已经存在的倒车雷达系统的特点。虽然整个系统存在很多的不足，但是作为相对简洁并且成本比较低的车辆雷达系统，还是具有一定实用价值的。



## 谢 辞

本此的设计是在我的导师曾屹的悉心指导和严格要求下完成的。曾屹老师在电路设计、硬件仿真和软件设计等方面造诣颇深，在我设计的期间给予了许多意义深远的指导。同时，他将理论知识和生产实际的背景相结合，对我的设计项目提出了宝贵的建议和意见，使我在项目设计中有更多的收获。曾屹老师丰富的专业知识、资深的实验经验和求实创新的工作作风深深的影响着我。在此，谨向曾屹老师致以我最崇高的敬意和真挚的感谢！

感谢我的家人和朋友对我生活上的关心，学习和工作的支持，这些使得我能够安心的完成我的设计工作。

最后，对在我的学习和成长道路上给予帮助的所有老师和朋友们表示深深地感谢，对评阅该论文的所有专家表示最崇高的敬意和真挚的感谢！

## 参考文献

- [1] 卓静.Arduino在大学生创新项目中的应用探索[J].江苏第二师范学院学报, 2016, 32 (06): 64-66.
- [2] 熊慧, 邱博文, 刘近贞.开源平台Arduino硬件生态扩充研究[J].实验室研究与探索, 2019, 38 (06): 103-106.
- [3] 史记征, 梁晶, 崔俊.基于Arduino的《传感器技术及应用》课程教学改革探索[J].教育现代化, 2019, 6 (A5): 98-99+115.
- [4] 汤传国.基于超声波测距的倒车雷达系统研究[D].长安大学, 2015.
- [5] 潘康福.基于超声波测距的倒车防撞报警系统研究[D].南京邮电大学, 2018.
- [6] 朱静.有刷直流电机旋转工作噪声分析及研究[J].无线互联科技, 2019, 16(16): 149-150.
- [7] 许元洪.某汽车扭力梁的实验分析及其结构优化[D].广西科技大学, 2019.
- [8] 刘贵如, 周鸣争, 王陆林, 王海.城市工况下最小安全车距控制模型和避撞算法[J].汽车工程, 2016, 38 (10): 1200-1205+1176.
- [9] 魏金丽.BRT安全适应性车速建模与分析[J].科学技术与工程, 2016, 16(29): 126-131.
- [10] 边春元, 段鹏飞, 肖鸿权, 李晓霞, 张争强.一种用于无刷直流电机回馈制动的PWM调制方式[J].中国电机工程学报, 2019, 39 (17): 5247-5256+5305.
- [11] 黄旭方, 王旭阳, 孙鑫, 陈冰雪.有效减小时延的自适应p-坚持CSMA协议研究[J].计算机工程与应用, 2017, 53 (12): 99-104.
- [12] 杨华甫, 倪子伟.基于TCP的流量控制和拥塞控制分析[J].计算机系统应用, 2005 (10): 50-53.