



西安交通大学
XI'AN JIAOTONG UNIVERSITY

电子设计课程报告

基于 Micropython 的多功能小车设计

组 员 1: 电气 64 班 李荆晖 2160400103

组 员 2: 电气 64 班 陈钱牛 2160400096

指导老师: 电气 学院 张 璞

完成日期: 2019 年 5 月



基于 Micropython 的多功能小车设计

李荆晖\陈钱牛 2019 年 5 月

摘 要

基于循迹和避障的载具技术在现代工业、物流业、零售业等领域得到了广泛的应用，不断催生出新的技术和产品。本项目旨在设计基于 STM32+Micropython 平台的多功能小车以模拟大型无人驾驶载具的远程操控和自动工作状态，探索无人驾驶技术的初级形态。该项目作品具备全方位的调速运动、超声波避障、红外循迹、手机和通讯盒的双蓝牙遥控等功能。本作品的特色在于在单片机上实现了面向对象的开发过程，设计了单超声波探头的不停止车的超声波避障和便携式语音蓝牙通讯机。本文详细地论述了作品的工作原理和软硬件设计方案，末尾简要记录测试结果和设计过程中遇到的问题与解决办法。该方案经过反复调试和改进，功能完善，最终呈现为具有基本的无人驾驶能力，符合设计预期。

关键词：红外循迹；超声波避障；蓝牙传输；无人驾驶；Micropython 开发

目录

1	前言.....	1
2	系统工作原理.....	3
2.1	电机驱动原理	3
2.2	超声波测距原理	4
2.3	舵机控制原理	5
2.4	红外传感和循迹原理	6
2.5	蓝牙传输原理	7
2.6	LD3320 语音识别原理	10
2.7	Micropython 简介与固件载入原理.....	11
3	软件设计分析.....	13
3.1	概述	13
3.2	蓝牙输入指令的控制流程	15
3.3	小车的前进、后退、左转和右转	17
3.4	避障功能	18
3.5	寻迹功能	19
4	硬件设计.....	21
4.1	主要芯片简要介绍	22
4.2	硬件电路设计	23
4.3	产品的外形和结构设计	27
5	作品测试与展示.....	29
5.1	参数调试	29
5.2	功能测试	29
5.3	测试总结	31
6	总结部分.....	32
6.1	阶段性进展情况	32

6.2	过程中遇到的困难和解决办法.....	32
7	课程心得体会	35
7.1	陈钱牛部分.....	35
7.2	李荆晖部分.....	37
	参考文献.....	39
	附录一 原理图	41
	附录二 PCB 图.....	43
	附录三 项目进度完成表	45
	附录四 使用说明	47

1 前言

基于循迹和避障的载具技术在国民生产生活的许多领域都有或都可以有广泛的应用。在现代工业、物流业、零售业等领域中，基于循迹和避障的运输载具使得大批量货物的精准、高效、安全运输成为了可能。在家用生活领域，搭载了避障系统的智能家居设备能够感知周围的物理空间，从而表现得更加智慧和人性化。同样，在电动汽车领域，无人驾驶技术大量应用了图像识别、激光识别等避障技术，在未来几十年内具有大好的发展前景和市场空间。

本项目旨在通过使用单片机实现小车的电机控制，进而实现避障、循迹、语音控制等功能，以模拟大型无人驾驶载具，以较低的成本探索无人载具的应用雏形。尽管这些技术已经得到了极大的发展，我们所做的工作也可以帮助我们学生以设计师的身份更加真切深入地认识我们生活中或未来可能将出现的类似的智能化设备。

根据电子设计的一般性流程，本项目设计目标总体可以分为三个阶段，**第一阶段**是创造和试验阶段。这一阶段采用稳压模块、电机驱动模块等功能模块组件搭建小车，编程实现小车的驱动、遥控、语音控制、避障、循迹等功能，实现作品雏形。**第二阶段**是硬件电路定制阶段。这一阶段将定制专属于产品的硬件电路，将之前使用的部分功能模块以芯片的形式集成到定制的 PCB 板上，做到外观简洁，功能高效；**第三阶段**是后期完善阶段。由于硬件平台的更换，尤其是电机驱动模块的更换，导致参数都需要重新调试。这一阶段主要是调试程序和定制产品最终展现形式。本文在论述的过程中对第一阶段中使用到但是在第二阶段被替换的功能模块不再赘述。

本文第二章将论述系统主要部分的工作原理和作品系统在理论上的可行性；第三章将论述系统的功能在软件上的设计方案。这一章我们将结合作品系统的整体和各部分的结构框图展开；第四章将论述系统的硬件设计，包括硬件电路设计和物理结构外观设计；第五章将测试并展示成品；第六章将总结设计过程，包括汇报我们的阶段性进展情况和叙述我们设计过程中遇到的问题。

需要指出的是，本项目基于搭载 Micropython 固件的 STM32F405RG 芯片。Micropython 是于 2013 年发起的适用于 ARM 体系结构微控制器的 Python3 解释器。选用 Micropython 的好处在于：烧写和调试方便高效；**面向对象**的编程代码可读性和可修改性大大提高；不必过多钻研底层驱动，更加专注于功能的实现；具有较为全面的官方文档，方便熟悉 Python 的非硬件专业人士学习使用。其缺点在于占用了部分的片内 ROM 资源；不能直接和底层打交道；无法直接使用 PC 上的大量优质包；适配单片机种类有

限（暂时）；正因为采用了 **Micropython**，我们的程序设计一大特点就是面向对象，我们的程序中定义并使用了大量的类及其继承关系。

项目相关的资料已经上传到 Github: https://github.com/zjcqn/car_project。

2 系统工作原理

本作品的功能框图如图 2-1 所示。作品拥有两种控制方式，分别为手机蓝牙控制和蓝牙模块间的连接控制。手机上，本作品利用蓝牙通讯助手 APP 定制了一个专用的控制界面，能够实现一键式控制小车；而语音控制方面，本作品则自制了一个通讯机，自带语音模块和电源开关，操作方便。小车拥有多项功能，分别为前进、后退、左右转、刹车、避障行驶和循迹行驶。本作品系统的主要工作流程是：手机或者语音通讯机通过蓝牙向小车发送指令，小车再根据相应的指令开始、停止或切换运动模式。

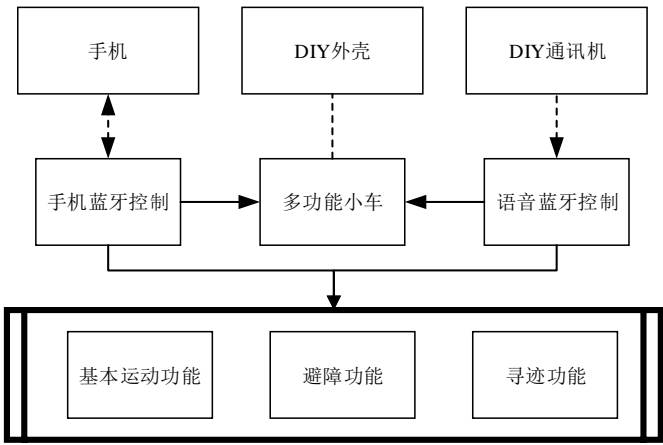


图 2-1 作品功能框图

2.1 电机驱动原理

小车的电机驱动需要提供较大的电流，本作品采用的是 L298N 双 H 桥芯片来控制小车电动机的转动，其形状和引脚定义如图 2-2 所示。

芯片内部使用了一个 H 桥来实现对电机正反转的控制，电机的 H 桥的框如图 2-3 所示（来自 L298N datasheet 的 BLOCK DIAGRAM）。以左边方框图为例解释电机驱动原理当 IN1 为高电平，IN2 为低电平且 ENABLEA 有高电平进入，通过与门，三极管①与三极管④被导通，而三极管②与三极管③被关断，则驱动流通过三极管①进入 OUT1 口，通过电机后再从 OUT2 口流出，最后通过三极管④流入地端，实现了电机的正转。同理，电机反转的原理与正转相同。

PWM 波控制电机转速的原理也可以通过此图来解释。ENABLE A 输入一定占空比的 PWM 波，则当 PWM 波为高电平时，相应二极管导通，电流驱动电机转动，而当 PWM 波为低电平时，与门的控制使所有三极管全部关断，使得电机没有驱动电流，电机减速，

于是，在高频率的 PWM 波的作用下，就可以使电机得到一定的速度，而且，电机转动速度的大小与 PWM 的占空比成线性关系。

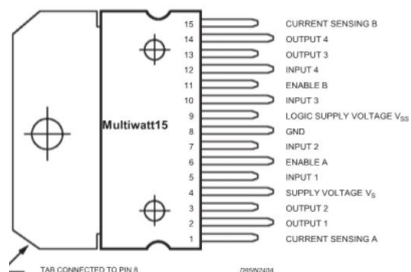


图 2-2 L298N 引脚图

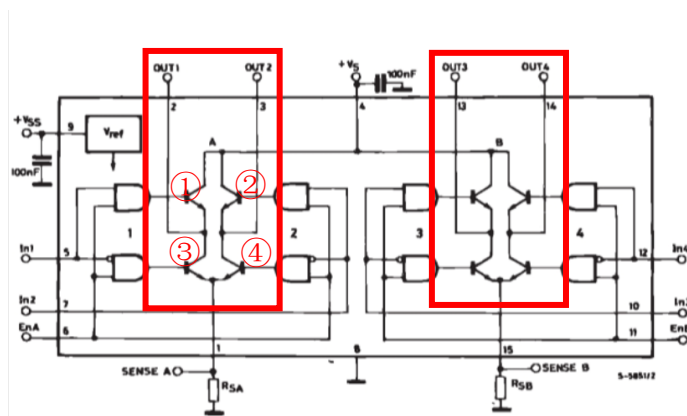


图 2-3 L298N 内部双 H 桥结构图

一个 L298N 模块就可以提供两个输出即 OUTPUT1 与 OUTPUT2 作为输出 A, OUTPUT3 与 OUTPUT4 作为输出 B，其中输出 A 给到小车左边的两个轮子，输出 B 给到右边的两个轮子，这样可以使左边两轮的速度保持一致，也使右边两轮的速度保持一致，既减少了模块使用的数量，又使得轮组运行更加稳定。其控制的框图如表 2-1 所示。

表 2-1 L298N 的控制的指令

直流电机	旋转方式	IN1	IN2	IN3	IN4	ENA	ENB
电动机 1	正转	高	低	—	—	PWM 信号输入	—
	反转	低	高	—	—		
	停止	低	低	—	—		
电动机 2	正转	—	—	高	低	—	PWM 信号输入
	反转	—	—	低	高		
	停止	—	—	低	低		

2.2 超声波测距原理

2.2.1 超声波发生原理

压电式超声波发生器实际上是利用压电晶体的谐振来工作的。超声波发生器内部结构如图 2-4 所示。它有两个压电晶片和一个共振板。当它的两极外加脉冲信号，其频率

等于压电晶片的固有振荡频率时，压电晶片将会发生共振，并带动共振板振动，便产生超声波。反之，如果两电极间未外加电压，当共振板接收到超声波时，将压迫压电晶片作振动，将机械能转换为电信号，这时它就成为超声波接收器了。

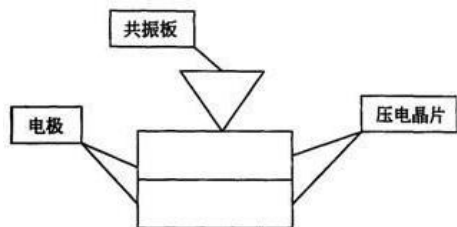


图 2-4 超声波发生器的结构



图 2-5 HC-SR04 测距模块实物图

2.2.2 超声波测距原理

时间差测距法。超声波发射器向某一方向发射超声波，在发射时刻的同时开始计时，超声波在空气中传播，途中碰到障碍物就立即返回来，超声波接收器收到反射波就立即停止计时。超声波在空气中的传播速度为 340m/s，根据计时器记录的时间 t ，就可以计算出发射点距障碍物的距离(s)。

2.2.3 超声波模块使用原理

超声波测距用到了超声波测距模块，其外形如图 2-5 所示。

其四个引脚分别为 VCC、Trig、Echo、GND。VCC 接电源正极，GND 接地。Trig 是触发端，给这个端口提供一个持续 10us 的高电平，就可以使模块的一个出声口发出超声波，超声波碰到障碍物返回后，会被另外一个声口接收，接收以后，Echo 会出现一段高电平，其时间是超声波发出到被接收的这段时间，读取这段高电平的时间 T_0 ，乘以声速再除以 2，就可以得到前方障碍物的距离了。计算公式为：

$$L = \frac{340 \times T_0}{2}$$

2.3 舵机控制原理

标准的微型伺服马达有三条控制线，分别为：电源、地及控制。电源线与地线用于提供内部的直流马达及控制线路所需的能源，电压通常介于 4V-6V 之间，该电源应尽可能与处理系统的电源隔离(因为伺服马达会产生噪音)。甚至小伺服马达在重负载时也会拉低放大器的电压，所以整个系统的电源供应的比例必须合理。控制线输入一个周期性

的正向脉冲信号，这个周期性脉冲信号的高电平时间通常在 1ms-2ms 之间 1.5ms 对应为 0° 。图 2-7 表示出一个典型的 20ms 周期性脉冲的正脉冲宽度与微型伺服马达的输出位置的关系。



图 2-6 舵机实物图

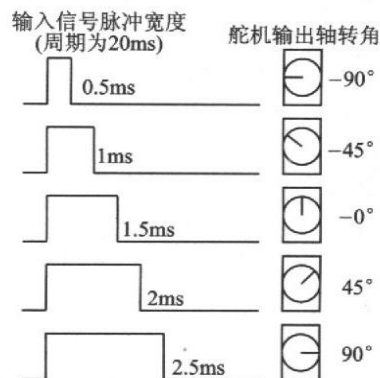


图 2-7 舵机驱动脉冲与伺服马达输出的关系图

2.4 红外传感和循迹原理

一组红外传感器的红外光电管有两种，一种是无色透明的 LED，此为发射管，它通电后能够产生人眼不可见红外光，另一部分为黑色的接收部分，它内部的电阻会随着接收到红外光的多少而变化。

无论是一体式还是分离式的红外传感器，其检测原理都相同，由于黑色吸光，当红外发射管照射在黑色物体上时反射回来的光就较少，接收管接收到的红外光就较少，表现为电阻大，通过外接电路就可以读出检测的状态；同理，当照射在白色表面时发射的红外线就比较多，表现为接收管的电阻较小，此时通过外接电路就可以读出另外一种状态，如用电平的高低来描述上面两种现象就会出现高低电平之分，也就是会出现所谓的 0 和 1 两种状态，此时再将此送到单片机的 I/O 口，单片机就可以判断是黑白路面，进而完成白底循黑迹的功能。

小车的寻迹用到了一个四路寻迹模块，但本作品只用到了其中的两路，还有两路是空闲的。其外形图 2-8

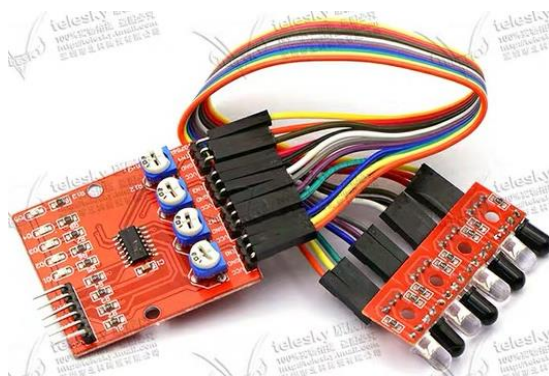


图 2-8 红外模块实物图

其主模块上有 6 个接口，分别是 VCC、GND、IN1、IN2、IN3 和 IN4，VCC 接电源、GND 接地、IN_x 是第 x 路寻迹传感器的输入。寻迹传感器的工作原理是红外发射管发射红外线，如果反射面是白色的，则红外线会反射回来被红外接收管接收，那么，就会在对应的 IN 口上产生高电平；如果反射面是黑色的，则红外线会被吸收而不会反射回来，那么，就会在对应的 IN 口上产生低电平。利用这个原理，就可以在白色的底板上循着黑色的轨迹行走了。

2.5 蓝牙传输原理

2.5.1 蓝牙的主与从

蓝牙技术规定每一对设备之间进行蓝牙通讯时，必须一个为主角色，另一为从角色，才能进行通信，通信时，必须由主端进行查找，发起配对，建链成功后，双方即可收发数据。理论上，一个蓝牙主端设备，可同时与 7 个蓝牙从端设备进行通讯。一个具备蓝牙通讯功能的设备，可以在两个角色间切换，平时工作在从模式，等待其它主设备来连接，需要时，转换为主模式，向其它设备发起呼叫。一个蓝牙设备以主模式发起呼叫时，需要知道对方的蓝牙地址，配对密码等信息，配对完成后，可直接发起呼叫。

2.5.2 蓝牙的呼叫过程

蓝牙主端设备发起呼叫，首先是查找，找出周围处于可被查找的蓝牙设备。主端设备找到从端蓝牙设备后，与从端蓝牙设备进行配对，此时需要输入从端设备的 PIN 码，也有设备不需要输入 PIN 码。配对完成后，从端蓝牙设备会记录主端设备的信任信息，此时主端即可向从端设备发起呼叫，已配对的设备在下次呼叫时，不再需要重新配对。已配对的设备，做为从端的蓝牙耳机也可以发起建链请求，但做数据通讯的蓝牙模块一般不发起呼叫。链路建立成功后，主从两端之间即可进行双向的数据或语音通讯。

在通信状态下，主端和从端设备都可以发起断链，断开蓝牙链路。

2.5.3 蓝牙一对一的串口数据传输应用

蓝牙数据传输应用中，一对一串口数据通讯是最常见的应用之一，蓝牙设备在出厂前即提前设好两个蓝牙设备之间的配对信息，主端预存有从端设备的 PIN 码、地址等，两端设备加电即自动建链，透明串口传输，无需外围电路干预。一对一应用中从端设备可以设为两种类型，一是静默状态，即只能与指定的主端通信，不被别的蓝牙设备查找；二是开发状态，既可被指定主端查找，也可以被别的蓝牙设备查找建链。

2.5.4 蓝牙模块与手机通讯

要和手机通讯，必须要车载一个蓝牙模块，我们使用了 HC-02 蓝牙模块，其外形如图 2-9 所示。



图 2-9 蓝牙模块实物图

蓝牙模块与 PYboard 间的通讯是遵循 UART 协议通讯协议的，所以蓝牙模块上的 RXD 与 TXD 要分别和 PYboard 上的 TXD 与 RXD 相连，当然也要给蓝牙模块接上电源和接地线，示意框图如图 2-10 所示。

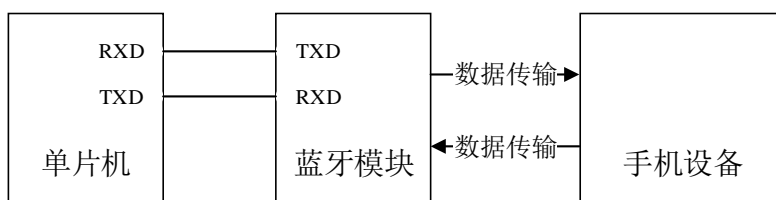


图 2-10 手机-单片机蓝牙通讯示意框图

接下来，就需要用手机蓝牙连接到这块蓝牙模块，默认蓝牙名称是 HC-02，默认配对码是 1234。当然，你也可以进入 AT 模式来改动蓝牙模块的配对码和蓝牙名称。具体操作这里不展开说，我会在后面的语音通讯中详细讨论。连接上蓝牙后，还需要手机下载一个蓝牙通讯助手来辅助通讯，通过手机终端发送不同的指令，单片机读取后再执行相应的操作，这样就完成了手机蓝牙控制小车进行行动。

2.5.5 蓝牙模块与蓝牙模块通讯

就本实验而言，本作品选用 LD3320 芯片来作为语音识别的核心，使用蓝牙模块来与车载蓝牙传输信息，这样就需要去实现蓝牙模块间的主从通讯其原理示意框图如图 2-11 所示。

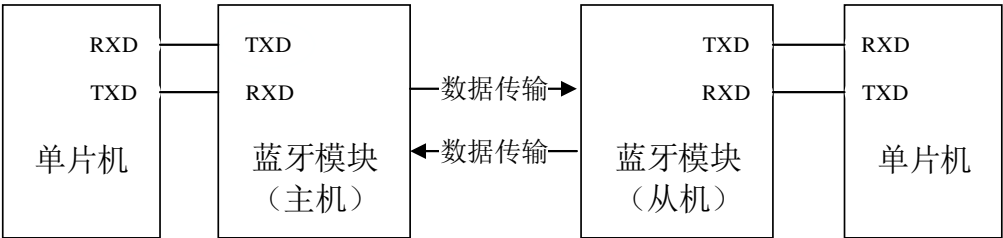


图 2-11 蓝牙模块间通讯示意框图

与手机蓝牙通讯不同的是，这次要实现的是蓝牙模块和蓝牙模块间的通讯，这就要用蓝牙主机与从机通讯的方式来实现通讯了，具体配对过程如图 2-12 所示。

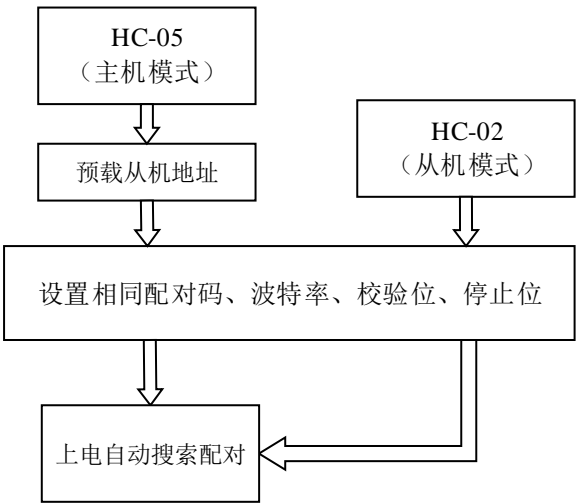


图 2-12 蓝牙模块间主从配对示意图

想要做到主机与从机的通讯，首先要确定谁是主机，谁是从机，举例来说，我们选择 HC-05 作为主机，需要先进入 AT 模式使其设置成为主机状态，具体操作是，将 HC-05 与电脑相连，即通过 UART 与电脑通讯，使用电脑的串口调试助手与 HC-05 通讯，设置好 COM 口、波特率（默认是 9600）、数据位（默认是 8）、校验位（默认是 NONE）和停止位（默认是 1），就可以开始传输相应的 AT 命令。用同样的操作设置另一块蓝牙模块作为从机。蓝牙间的通讯要实现的效果是，两块蓝牙上电以后，可以自动的连接在一起，这和手机搜索并连接蓝牙芯片的操作是不一样的，所以主机必须采用指定蓝牙地址的模式，而指定的蓝牙地址即为从机蓝牙芯片的蓝牙地址。同样，可以通过 AT 模式来获取从机蓝牙芯片的蓝牙地址，这样就建立了两块蓝牙芯片间的通讯，就可以做到语

音识别模块和车载单片机间的通讯了，也就做到了通讯机控制小车运动的效果。表 2-2 为设置过程中需要用到的 AT 操作码。

表 2-2 AT 操作码

序号	作用	AT 指令
1	获取模块蓝牙地址	AT+ADDR?
2	设置蓝牙配对码	AT+PSWD=XXXX（四个数字）
3	设置模块角色	AT+ROLE=X（0 表示从机，1 表示主机）
4	设置串口参数	AT+UART=XXXX, X, X 三个参数依次为： 波特率（本实验使用的是 9600） 停止位（本实验使用的是 1） 校验位（本实验使用的是 0，即 NONE）
5	设置连接模式 （只有主机才能设置 连接模式）	AT+CMODE=X X=0:指定蓝牙地址连接模式 X=1:任意蓝牙地址连接模式 X=2:回环角色
6	设置绑定蓝牙地址	AT+BIND=<蓝牙地址>

2.6 LD3320 语音识别原理

LD3320 语音识别芯片采用的就是 ASR 技术，就是由 LD3320 和单片机(或嵌入式系统)组成的语音识别系统。本文中选用的 MCU 是 STC10L08XE 单片机。

用户只需要把识别的关键词语以字符串的形式传送进芯片，即可以在下次识别中立即生效。简单地通过设置芯片的寄存器，把诸如“你好”这样的识别关键词语的内容动态地传入芯片中，芯片就可以识别所设定的关键词语了。每个关键词语可以是单字、词组、短句或者任何的中文发音的组合，同时也不需要用户作任何的录音训练。

LD3320 有两种用户使用模式，即触发识别模式和循环识别模式。用户可以通过编程，设置两种不同的用户使用模式。本作品主要使用循环识别模式。

循环识别模式：系统的主控 MCU 反复启动识别过程。如果没有人说话就没有识别结果，则每次识别过程的定时到时后再启动一个识别过程；如果有识别结果，则根据识别作相应处理后(比如播放某个声音作为回答)再启动一个识别过程。其识别过程如图 2-13 所示。

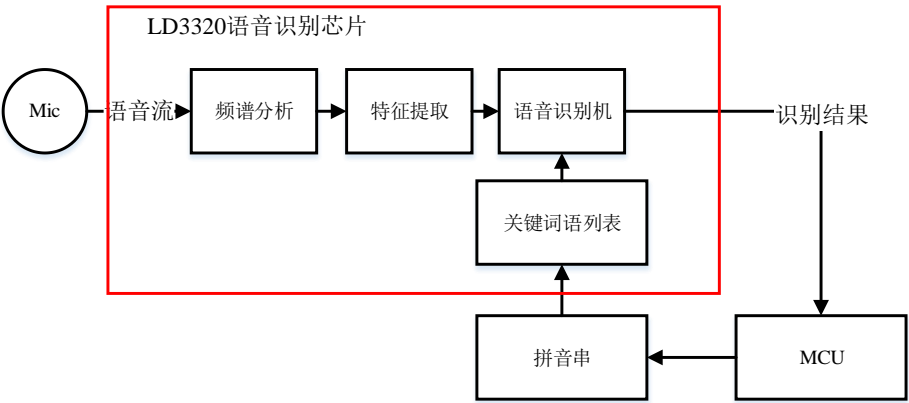


图 2-13 LD3320 模块识别原理

2.7Micropython 简介与固件载入原理

Micropython 是在 STM32 等 MCU 芯片上运行的 Python 解释器。相当于对 MCU 的初次开发与封装,用户使用 Micropython 开发自己的项目相当于是进行二次开发,可以。在 Micropython 中我们可以使用 Python 的基本功能和基本类,对于一些基础的类似于 PWM 生成, UART 收发功能,都有预定义的类可供直接调用,在搭载 Micropython 的平台上,用户完全可以实现面向对象的开发流程。搭载 Micropython 后的 STM32 通过 USB 串口连接电脑后以磁盘的形式出现,用户可以将编辑完的脚本直接存入此磁盘,待芯片读写完成就可以在线或离线运行。用户还可以通过 REPL（交互式解析器,就像 Matlab 的命令行）来进行学习和在线调试,只需要安装类似于 Putty 的 SSH 远程控制工具访问对应 COM 端口,非常方便。

需要使用 Micropython 的前提是需要要在 STM32 中预载如 Micropython 的固件,本项目使用的是 Pyboard V1.10 的版本。以下介绍 Micropython 固件强制更新（DFU）的方式。STM32 具有三种启动方式,分别是 从 Flash 启动、从 RAM 启动、从系统存储器启动,三种启动方式的存储介质均是芯片内置的,需要通过匹配 BOOT0 和 BOOT1 引脚设计,匹配规则如下表 2-3 所示。系统内置的 Flash 中的内容掉电不丢失,从 Flash 启动为正常的工作模式;SRAM 内的内容掉电丢失,但是读写速度快可以用于调试程序;系统存储器是芯片内部一块特定的区域,芯片出厂时在这个区域预置了一段 Bootloader,用户无法修改。使用系统存储器启动即进入了 DFU Mode（强制更新模式）,此时可以配合计算机上的 DFU 文件烧入 Micropython 的固件。完成固件更新后将启动模式改回从 Flash 启动即可正常使用。此时使用 USB 口连接单片机和计算机后,计算机上会显示出 现新磁盘,说明固件安装正常。

表 2-3 STM32 启动方式配置表

启动模式选择引脚		启动模式
BOOT1	BOOT0	
X	0	FLASH
0	1	System Memory
1	1	SRAM

3 软件设计分析

3.1 概述

小车的设计大致可以分为**软件设计**和**硬件设计**两大块，软件设计部分主要是围绕各个功能之间的逻辑关系以及各个功能的实现流程展开，硬件设计包括硬件电路设计，小车结构和外形设计、通讯机结构外形设计等。本章讨论软件设计分析内容，下一章讨论硬件部分。**程序源代码可以参考附录四使用说明部分连接 PCB 板在线查看。**

本项目基于 Micopython 的 STM32，它的开发语言是 Python。因此本项目最大的特点是面向对象的开发，设计过程中设计并使用了大量的类（class）及其嵌套继承关系。各个类之间的结构关系图如图 3-1 所示。

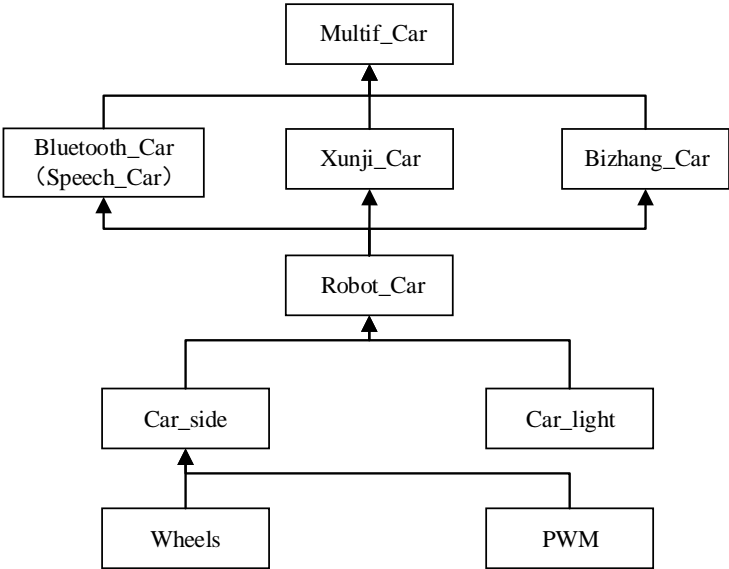


图 3-1 小车各类（class）之间的的结构关系示意图

最底层的三个类是轮子类、PWM 类和车灯类。**轮子类**是控制轮子的各种运动方式的类，它的主要功能是输出高低电平给驱动电机芯片的 IN1~IN4 口，实现电动机正转、反转以及停止旋转的选择。**PWM 类**则直接调用了 Micropython 中预定义的 Timer 类，能够实现输出 PWM 波给电机驱动芯片的使能端 ENA 与 ENB，实现的功能是控制轮子的转速。**车灯类**有刹车时亮起的红灯、前进时亮起的绿灯和转弯时亮起的黄灯。车灯类实现的功能是，配合之后更复杂的子类实现对应小车的运动状态，亮起相应的指示灯。

由于本作品保持一侧的车轮速度相等，所以需要分别控制两侧车轮的运动状态，因此我们定义了一个类来控制一侧的车轮的运动。通过继承 PWM 类和轮子类，系统得到

了**车侧类**。此类实现的功能，一是控制一侧的车轮运动状态相同，二是控制车轮的速度变化。并且对 PWM 的占空比进行限制，防止占空比超过 100%使得系统混乱，也防止占空比过低是的电动机停转带来了的发热问题。

通过集成车侧类和车灯类，系统得到了基础的**机器车类**，它通过定义了指令字典，控制了小车的各项运动功能，是其他功能的基础。此类实现的功能有：

1. 前后：分为快速、中速和慢速三挡；
2. 转弯：有左右转之分，转弯幅度有转小弯，转中弯、转大弯和原地转弯四挡；
3. 刹车：退出当前模式并立即停止。

之所以分挡位是因为实际使用中并不需要无级调速，分为经过调试的挡位更加合理。

进而通过继承机器车类，并进一步设计开发，得了蓝牙车类、循迹车类、避障车类。

蓝牙车类的主要功能是实现作品板载蓝牙与手机蓝牙或通讯机蓝牙的通讯，实现小车的驱动控制（单独的蓝牙车类只具有驱动控制，无法做到循迹和避障）。此类还包含指令字典和信息检测匹配的方法，使信息更加简洁准确的传入并接收。原本还设计了语音车类，专门用于接受通讯机蓝牙信息，但是因为其功能和蓝牙车类非常类似，因此整合。

循迹车类的主要功能是实现小车的循迹运动功能，此类包含了循迹模块与单片机信息传递的方法和循迹运动的方法。

避障车类的主要功能是实现小车的避障运动功能，此类包含了超声波测距的方法和舵机旋转的方法以及避障运动的方法。

继承上述三大类，就可以得到**多功能车类**。多功能车类实现的功能是以手机蓝牙或者通讯机蓝牙作为控制，控制小车的前进、后退、左转、右转、刹车、循迹行走和避障行走。其实现方式是重写机器车类中的指令字典，加入了循迹和避障的指令。

以下按照主要功能模块的实现流程来叙述开发思路。

3.2 蓝牙输入指令的控制流程

蓝牙通讯和信号处理的软件流程图如图 3-2 所示。为了更好地将蓝牙指令与具体命令相关联，首先需要设置一个对应关系。字典是 Python 中的一个的容器类，它的作用是将关键字和内容对应起来。这里我们用字典是将接收的蓝牙信息和对应的操作功能用字典对应起来。由于蓝牙传输信息过程中可能存在干扰和错误，指令应该尽可能短，过长的指令可能会影响系统的实时性和导致信号接收错误。例如传输过来的信息时“6”，通过字典的对应，就可以找到“6”对应的功能是循迹模式。表 3-1 列举了本项目中应用到的蓝牙指令与对应命令的对应关系。

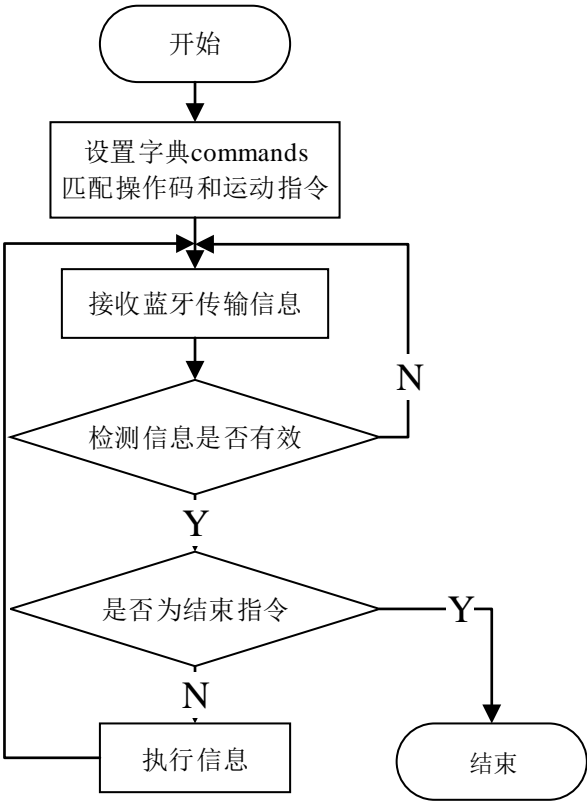


图 3-2 蓝牙接收和指令处理流程图

表 3-1 接收蓝牙指令信息与命令对应表

指令信息	对应命令
0	刹车或退出当前模式
11~13	前进 1~3 挡
21~23	后退 1~3 挡
31~34	左转弯 1~4 挡
41~44	右转弯 1~4 挡
5	进入避障模式
6	进入循迹模式

接收到传输过来的数据后，程序首先需要对信息进行检测查错，以排除因信息接收不完全或接收错误信息导致无法匹配字典关键词的情况发生。先判断接收到的字符串是否是数字，再判断这数字是否符合指令集中的关键词。如果都符合则执行指令并保持该正确的指令直到下一次接收到正确的指令；否则放弃信息，等待接受下一次蓝牙信息。图 3-3 为指令信号检测的流程图。

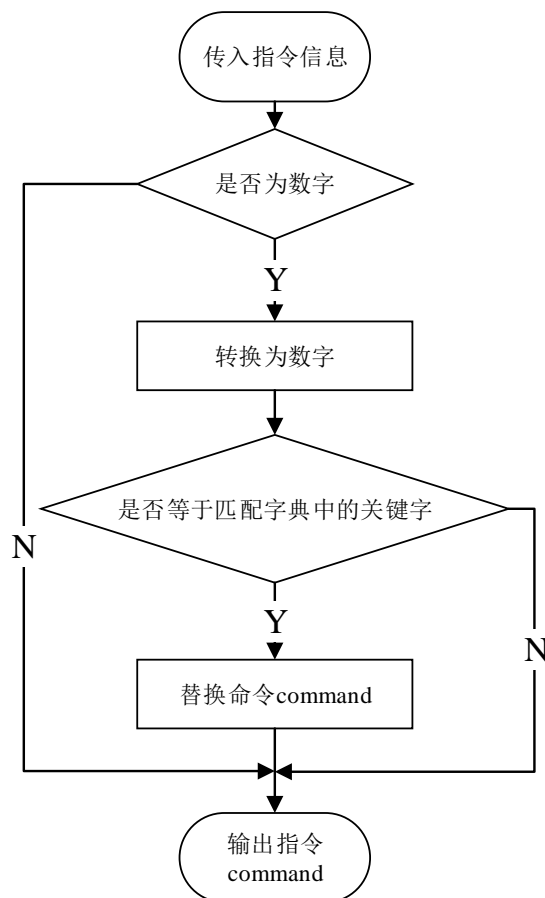


图 3-3 指令信号检测的流程图

3.3 小车的前进、后退、左转和右转

小车的前进和后退以及控制其速度的原理在电机驱动原理已经讲述过了，软件程序可以实现小车的三种档位速度前进、三种档位速度后退和四种档位速度转弯。其实现方式如表 3-2 所示，由于小车电机并不完全一样，其中的占空比关系并不准确，只是用于体现我们的设计思路，具体数值需要测试，见第五章。

表 3-2 小车运动方式与两侧电机 PWM 占空比的关系

	左侧车轮		右侧车轮	
一档前进	前进	80%	前进	80%
二档前进		90%		90%
三挡前进		100%		100%
一档后退	后退	80%	后退	80%
二档后退	后退	90%	后退	90%
三档后退		100%		100%
一档左转	前进	90%	前进	100%
二档左转		80%		100%
三档左转		70%		100%
四挡左转	后退	80%		80%
一档右转	前进	100%	前进	90%
二档右转		100%		80%
三档右转		100%		70%
四挡右转		80%	后退	80%

3.4 避障功能

小车自动避障是一个复杂的过程，我们的要求是使用一个可以转动地超声波实现小车在运动过程中不停止运动的避障。需要使用的外设模块有超声波测距模块和舵机旋转模块，以及连接超声波和舵机的云台。舵机与超声波模块配合使用，就可以测量多个方向上的障碍物距离，来做出向对应的避障操作。其避障的流程图如图 3-4 所示。

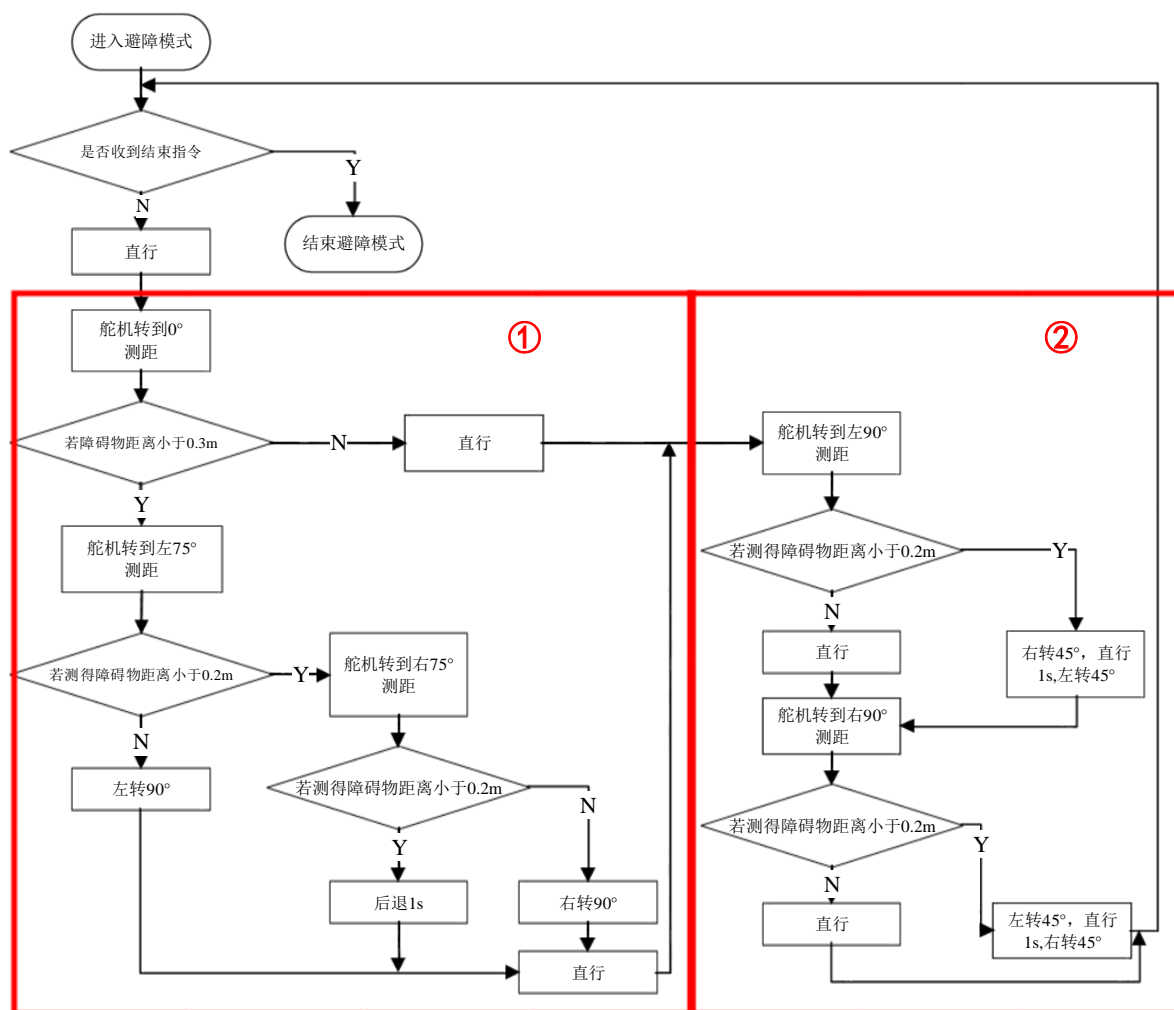


图 3-4 避障流程图

本作品能够完成的避障操作分为两种，第一种是避开正前方的障碍物，一种是远离左右侧面墙体。图 3-4 中①部分所示的是避开正前方的障碍物的操作流程图。图 3-4 中②部分所示的是避开左右墙体的操作流程图。小车避障运动的一次循环就是，正面避障—>左侧避障—>右侧避障。

3.5 寻迹功能

寻迹的原理流程图如图 3-5 所示。我们把红外寻迹探头放在车头的两侧。设计思路是，当左侧红外寻迹探头检测到黑色纸带时，左转 50ms，然后继续检测，直到左侧的红外探头检测不到黑色纸带为止。对于右边的纸带也是同理。关于延时 50ms 的解释是，如果没有延时，因为红外检测会相当灵敏，在检测到一点黑色或者白色时，会立马做出调整，灵敏度十分高，导致寻迹时的效果较差，小车易出现原地抽搐的现象，故这里设置 50ms 延时来降低灵敏度。

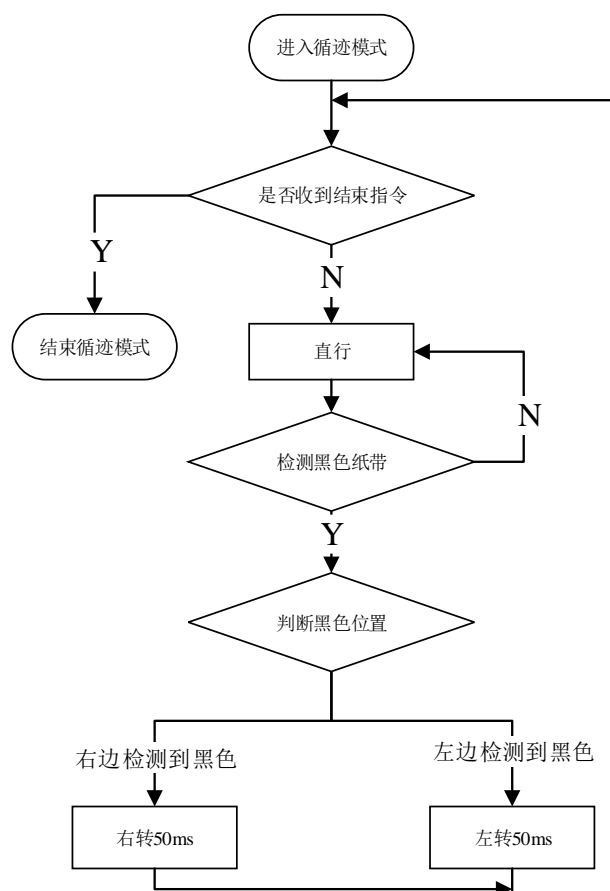


图 3-5 循迹流程图

4 硬件设计

硬件设计主要是指电气设计和产品的外形结构设计。硬件设计的目标是将产品在电气结构和机械结构上实现集成化，尽可能做到简洁、美观和高效。本项目的硬件设计的特点在于：

- 1) PCB 板面积小（5.7*6.9mm），结构紧凑；
- 2) PCB 板上同时具有三个电压等级的总线；
- 3) PCB 板集成了稳压、蓝牙和电机驱动芯片，减少集成模块使用，减少接线；
- 4) 合理 DIY 通讯盒和小车外壳，起到了集成、稳固、保护和美化的作用。

硬件电路设计的结构框图如图 4-1 所示。PCB 板主要包括单片机及其最小系统、稳压、电机驱动以及外部设备的接口。外部设备主要包括红外探头、超声波模块、舵机模块和直流电机。最右侧是语音通讯盒的内部结构框图，主要包含语音识别模块和蓝牙模块。

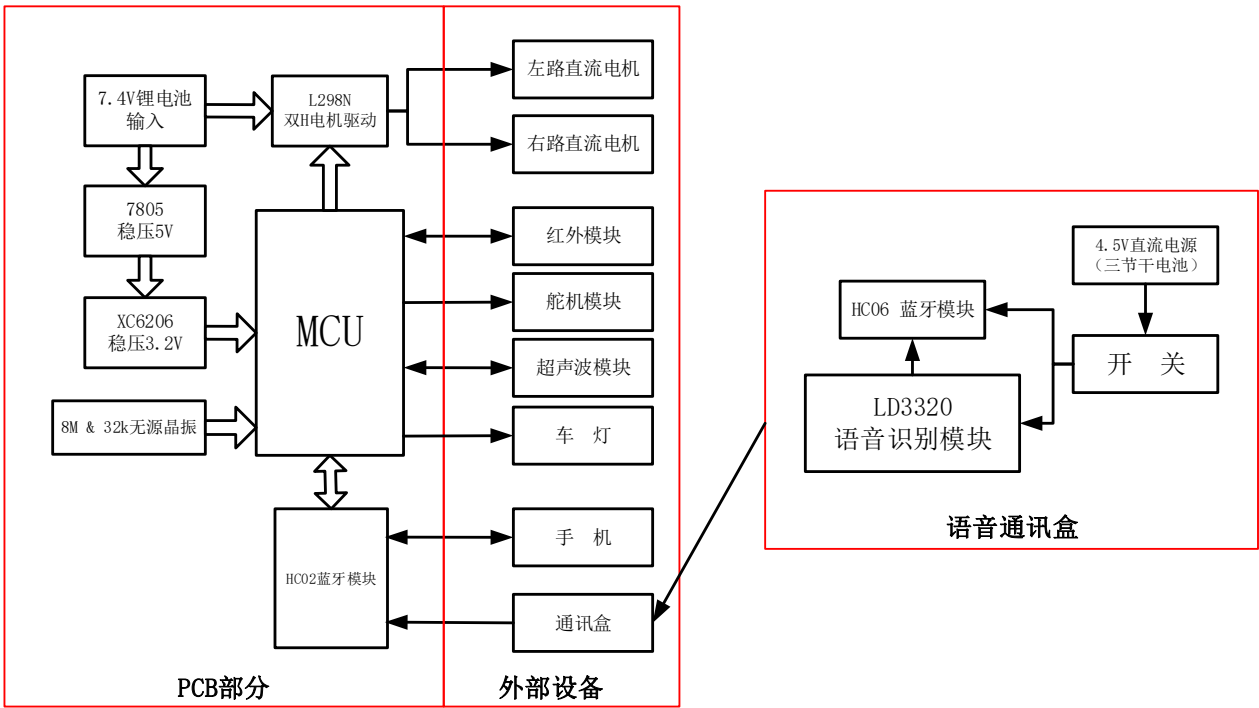


图 4-1 硬件结构框图

4.1 主要芯片简要介绍

本小节就本次 PCB 设计用到的主要芯片简要介绍。介绍内容主要围绕本次设计使用到的展开，不过多叙述，若读者需要详细了解芯片内容可以查阅数据手册。

4.1.1 STM32F405RGT6 (LQFP-64)

作为本项目的主控芯片，STM32F405RGT6 最大支持 168M 时钟频率（本项目使用了 8M 外部晶振，倍频后为最大频率为 72M），192KB 数据 RAM，1MB Flash 程序存储器，拥有 51 个完整的独立 IO 口，拥有多个定时\计数器和 12 位 ADC 通道。

4.1.2 L298N 电机驱动芯片 (Multiwatt 15 封装)

L298N 是一种双 H 桥电机驱动芯片，其中每个 H 桥可以提供 2A 的电流，功率部分 (电机驱动电压) 的供电电压范围是 2.5-48V，逻辑部分 5V 供电，接受高电平范围为 2.3V~VSS，低电平范围为 -0.3~1.5V。一般情况下，功率部分的电压应大于 6V。

图 4-2 是 L298N 芯片的框图（来自 datasheet 的 BLOCK DIAGRAM）。其内部是两个全桥电路。使用 PWM 波控制 EN 口的占空比可以实现两路电机的四象限运行，可以实现电机的调速。控制原理已在原理部分阐述，这里不再赘述。

集成电路中总的可以分成两个电压等级，控制电压和驱动电压，实现电机与单片机的电气隔离。控制电压是输入的逻辑信号的电压，控制三极管的导通和关断，与 STM32 直接连接。STM32 输出电压考虑到芯片输入电压范围内，无需接上拉电阻。驱动电压是 VS，直接供给到三极管 C、E 两端，与 STM32 隔离。

电路中的电容参数按数据手册推荐使用。SENSE 端未使用，接地。

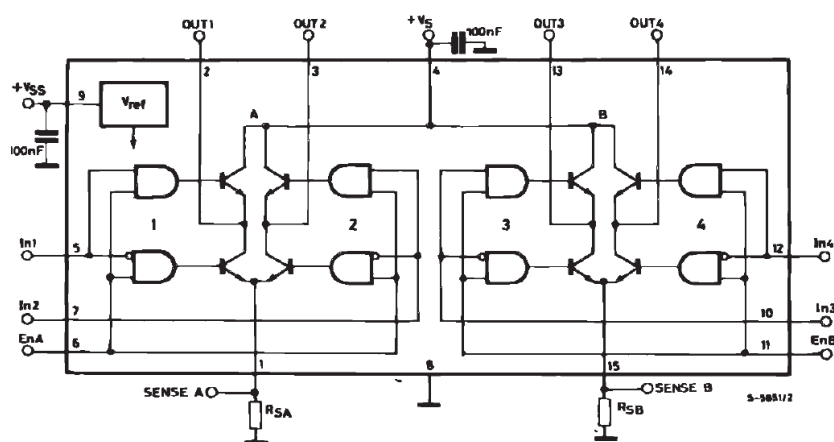


图 4-2 L298 内部结构框图

4.1.3 XC6206P332MR 3.2V 稳压芯片（SOT23 封装）

XC6206 系列是一款高精度、低功耗的电压调整芯片，在输出电流较大的情况下输出压差也能很小。其最大输出为 200mA，足够驱动 STM32 使用。电流芯片框图和典型应用（来自 datasheet）如图 4-3 和图 4-4 所示。芯片内部包括一个电流限制电路，一个驱动三极管和一个高精度参考电压源和一个误差校准电路。根据典型电路我们只需要提供电源（大于输出电压小于 7V）和参考地就能输出 3.2V 电压供单片机工作使用。

4.1.4 7805 5V 稳压芯片（TO-220 封装）

7805 芯片是常见的 5V 输出的三端稳压芯片，TO-220 封装的三个引脚分别对应输入电压、输出电压和参考地。其最大输入电压为 35V，最大输出电流为 1.5A。该芯片内部具有过流、过热及调整管的保护电路等。本项目采用的典型电路如图 4-5 所示。

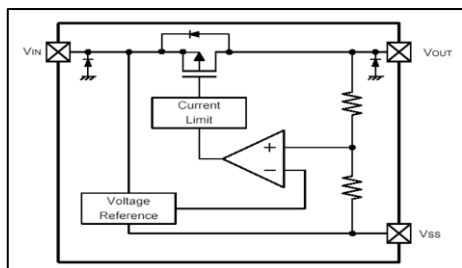


图 4-3 XC6206 内部结构框图

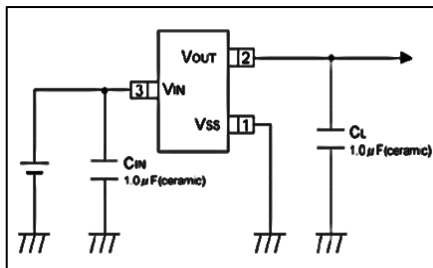


图 4-4 XC6206 典型电路拓扑

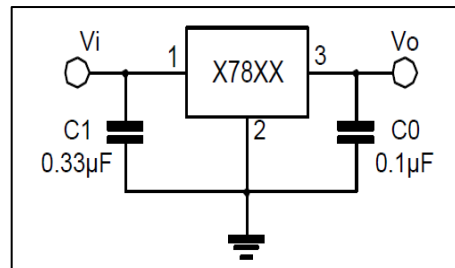


图 4-5 7805 典型电路拓扑

4.2 硬件电路设计

PCB 电路设计是一个从零开始的过程，由于对硬件电路、集成 IC 以及设计流程的不熟悉或者疏忽，难免出现问题，本项目的 PCB 设计也是经过了多次的迭代。改进后的完整 PCB 原理图和布线图见附录一、二。以下分块叙述硬件电路设计细节。

4.2.1 供电电路

本项目采用的供电系统较为复杂，电压转换的大致结构如下图 4-6 所示。PCB 板上可以提供 7.4V、5V、3.2V 电压。其中 7.4V 为外接锂电池供给 L298N 驱动模块驱动电机的电压；5V 为外设驱动电压，可以有 L298N 转换得到，也可以由 USB 接口得到；3.2V 为 STM32 单片机所需的电压，由 5V 电压经 XC6206P332 芯片稳压变换得到。在每一级电压变换处和接入 STM32 处均并联滤波电容或电感，相应电路和参数值均参考已有案例与 Datasheet 得到，以保证其可行性。

STM32 需要多路供电，一共需要 4 路数字电源，1 路模拟电源，2 路数字地和

1 路模拟地。由于没有用到模拟信号，因此数字口和模拟口统一用数字电源和数字地，如图 4-7 所示。

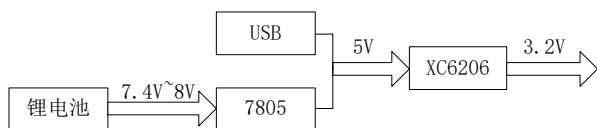


图 4-6 电压转换结构图

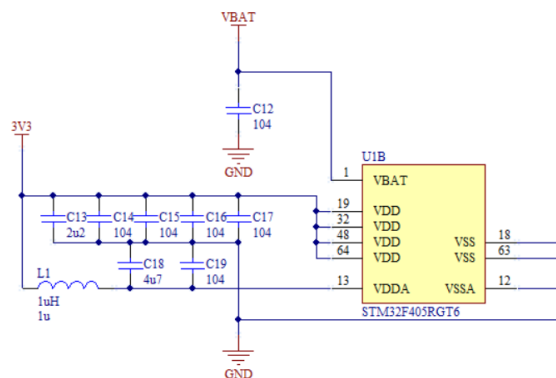


图 4-7 STM32 供电电路

4.2.2 晶振电路

本项目采用两个外接晶振为 STM32 提供时钟频率，分别是 8MHz 四脚无源高速晶振为系统提供主时钟和 32.768KHz 无源低速晶振为系统内 RTC（Real-Time Clock）提供精准时钟。

4.2.3 启动配置电路

STM32 有 3 种启动模式，通过配置 BOOT 引脚选择。BOOT 引脚配置电路如下图 4-8 所示。默认启动为从 Flash 启动（00、10），因此默认 BOOT 引脚都置低。但是当我们给单片机安装或者升级 Micropython 的 DFU 固件的时候需要使用到从系统存储器启动（01），因此预留 BOOT 配置电路。（这里预留 BOOT0 引脚没有起到作用）

4.2.4 下载电路（PC 端供电电路）

本项目 PCB 板的唯一下载渠道是 USB 下载，选用的 USB 接口为小巧的 MicroUSB，与 PCB 小尺寸相匹配，接法参考相关典型电路，如图 4-9 所示。VUSB 经肖特基二极管用于保护 USB 口，然后为 PCB 板提供 5V 电压。

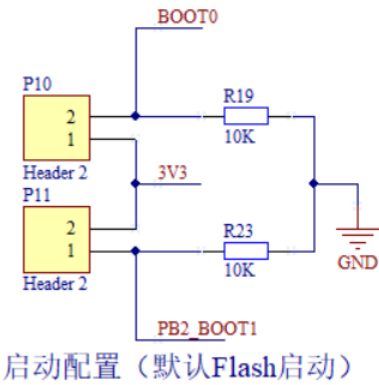


图 4-8 BOOT 配置电路

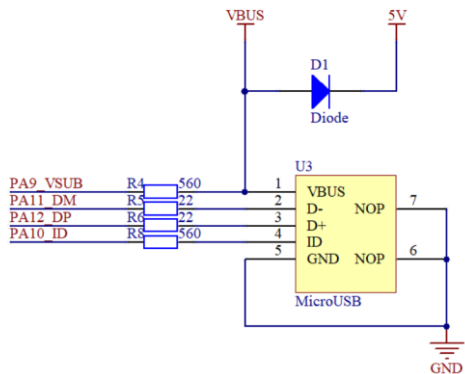


图 4-9 USB 串口电路

4.2.5 电机驱动电路

电机驱动电路如图 4-10 所示，电路的主体是 L298N 芯片，该芯片可以驱动电机工作，驱动方法见原理部分。由于集成在 PCB 板上，芯片只需要留出 OUT1~OUT4 四个接口连接电机即可，非常方便简洁，体现了集成化的优点。

值得一提的是，在后来的调试中发现本电路设计存在较为严重的缺陷。电机驱动电路应该在输出的四个 OUT 口上各加两个续流二极管用于泄放反电动势。由于缺乏设计经验，在设计 PCB 的过程中没有考虑到这一问题，导致电机停转或者减速的时候 L298N 急剧发烫。后来通过飞线外接 8 个 1N4007 大功率管补救解决。外接电路实物如图 4-11 所示。

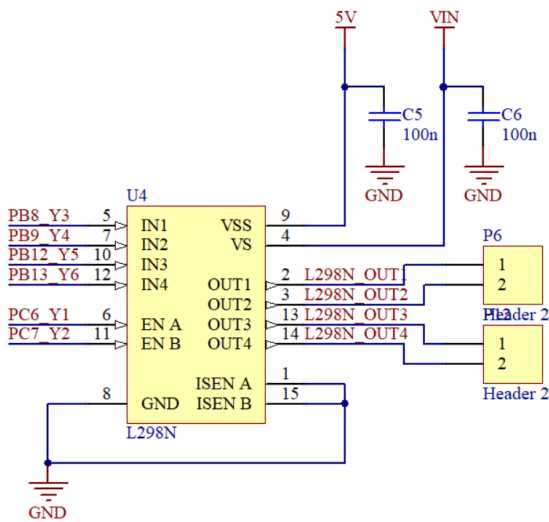


图 4-10 L298N 电机驱动电路

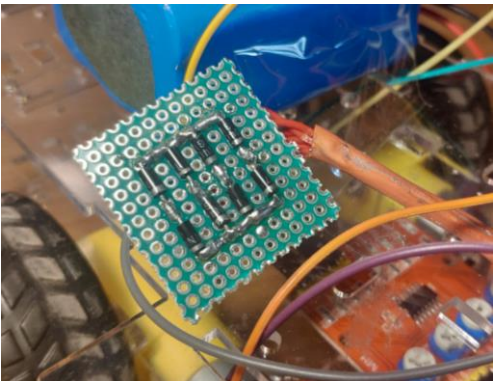


图 4-11 飞线外接续流二极管实物图

4.2.6 PCB 信号灯和车灯电路

PCB 信号灯和车灯具有相同的功能但是却是两个不关联的电路,图 4-12 是信号灯电路,图 4-13 是车灯电路。设计两个电路的原因在于车灯需要用导线从 PCB 板上焊孔外接到小车上使用并固定在小车上,在调试的时候非常不方便,在 PCB 上集成相同的信号灯可以方便我们调试使用。在车灯电路中的串联电阻需要根据选用的车灯调整。

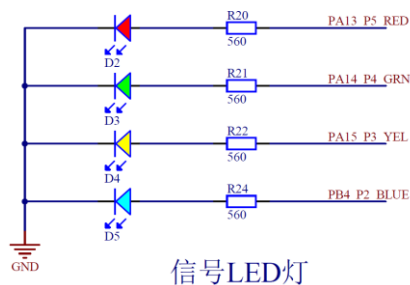


图 4-12 信号灯电路

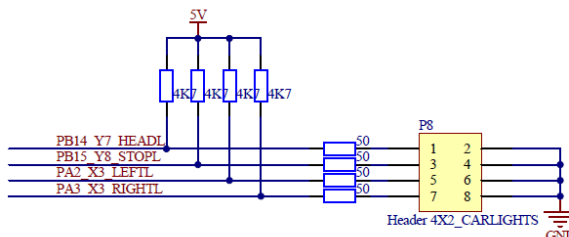


图 4-13 车灯电路

4.2.7 蓝牙电路、循迹电路、避障电路等外设接口

蓝牙电路、循迹电路、避障电路都是采用排插或焊接将集成器件与 PCB 板连接,本项目的 PCB 板上分别流出三块区域与之匹配。分别如下图 4-14、图 4-15、图 4-16 所示。

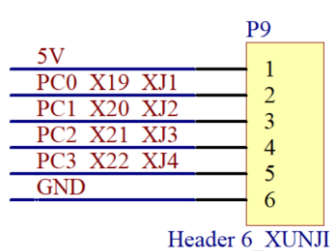


图 4-14 循迹插口

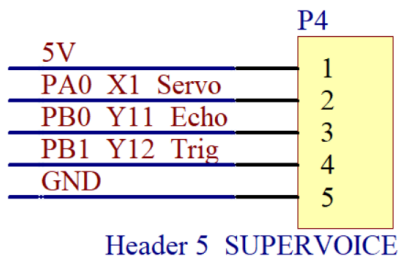


图 4-15 避障插口

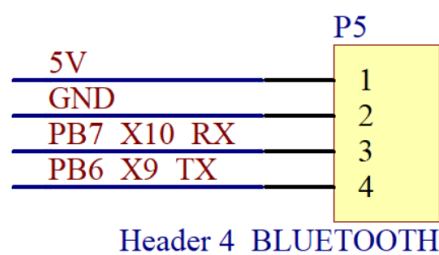


图 4-16 蓝牙通讯插口

4.2.8 PCB 布线设计

因为采用的是 64 脚的 STM32,整板基本统一采用表贴元件,排布紧凑,总面积小(5.8cm*7cm)。布线过程中遵循先布局再布线,先最小系统后外围器件最后布电源的原则。所有标贴器件均放置在 PCB 板的 Top Layler,布局位置考虑到板子在实际小车中的位置设置。考虑到人工焊接,所有阻容器件均采用 0805 封装(刚开始使用 0603 封装,不容易焊接),器件之间预留足够的焊接操作空间。PCB 布线过程简单而复杂,没有太多可以叙述,完整 PCB 图见附录二,最终焊接完成的电路板实物图如图 4-17 所示。

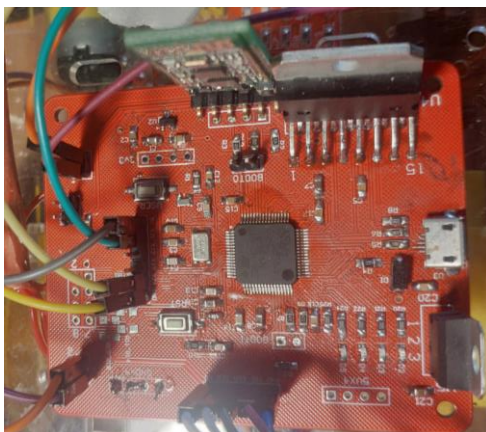


图 4-17 电路板实物图

4.3 产品的外形和结构设计

4.3.1 小车物理结构设计

项目采用双层亚克力板加六枚铜轴连接构成基本小车的骨架，可分为顶层、中层和底层。底层用于固定 4 个直流电机和轮子和循迹感应探头，中层用于放置循迹模块，顶层用于放置大容量锂电池，主 PCB 板和舵机超声波模块。小车物理结构如下图 4-18 所示。

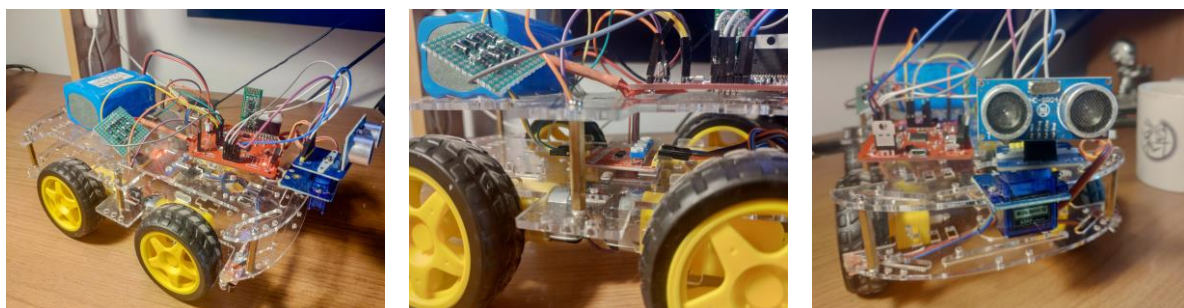


图 4-18 作品物理结构图

4.3.2 小车外壳设计

小车外壳设计的总体思路是在原有小车物理框架的基础上，使用板纸和胶枪制作出车型可拆卸外壳，在板纸外侧精细化处理，加一层材料，最后再外层喷漆或彩绘。

考虑到小车头部装有超声波探测器，尾部装有大容量电池，外型采用头尾分离式设计（两厢车外观）。头部包括“引擎盖”以前部分，需要将超声波探测器露出来，并能够保证其正常转动工作，同时具有保护超声波探测器的功能，即其尺寸需要覆盖超声波转动范围的包络线。尾部包括“前挡风玻璃”以后部分，这一部分在水平维度上需要和头部契

合，使用三角形卡口结构使其相互牵制以起到固定的作用；在垂直维度上需要以锂电池作为支撑，并用磁铁或其他手段保证其垂直方向上的稳定。板纸制作成果如下图 4-19 所示。



图 4-19 外壳实物图

4.3.3 通讯盒设计

通讯和总共包括蓝牙模块、语音模块、开关和电池模块。通过板纸和胶枪制作出长方体外壳，将以上器件固定并裁剪人机交互的窗口，以知道各模块是否正常工作。在盒子的侧面固定开关按键，控制电池的通断。盒子的门采用磁吸式的结构，方便开合。通讯盒设计如下图 4-20

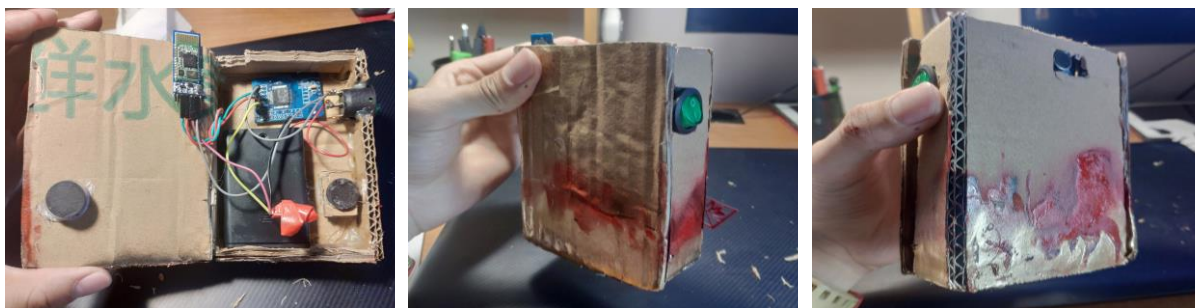


图 4-20 通讯盒实物图

5 作品测试与展示

5.1 参数调试

小车电机驱动参数需要通过实测配置，本小节将展示调试方法和配置结果。注意本章测试环境均为较光滑瓷砖地面，水泥地、沥青地可能会因摩擦系数不同而产生不同结果。

5.1.1 直线运动调试

直线分为前进和后退，各设三档调速（无极调速也可以实现）要求是将小车置于直线中间，让小车速度较快的轮子的控制 PWM 占空比分别保持在 100，90，80，改变较慢轮子运动。小车保持 2 米以内不驶出直线为合适。结果记录如表 5-1：

表 5-1 前进直线运动调试结果（单位：%）

	快速左	快速右	中速左	中速右	慢速左	慢速右
前进	100	93	90	86	80	75
后退	100	95	90	87	80	80

5.1.2 转弯测试

转弯分为前左转和前右转，依据转弯半径从小到大各设四档，最大挡位为原地转弯。四档速度均为一次满速，调整另一侧。测试过程中以地面瓷砖为参考，尽量追求左右侧旋转速度一致。结果记录如表 5-2。

表 5-2 转弯调试结果（单位：%）

	I	II	III	IV
左转弯（右 100%）	95	90	85	-80
右转弯（左 100%）	86	83	79	-80

5.2 功能测试

5.2.1 蓝牙发送正确率率测试

通过手机分别在 2、5、10 米远处分别测试运动指令 5 次，记录结果如表 5-3 所示。由于蓝牙性能出乎意料的好，在测试地走廊（约 60 米）两个尽头测试都能准确识别，

因此加设一列 60 米。蓝牙性能完全满足要求，无需测出其上限。

表 5-3 蓝牙发送正确率测试结果

	2 米	5 米	10 米	60 米
正确识别次数（次）	5	4	5	5
正确率（%）	100	80	100	100

5.2.2 语音识别收发正确率测试

任选 3 人各通过语音通讯盒发送运动指令 10 次，记录结果于表 5-4。

表 5-4 语音识别正确率测试结果

实验人员	正确次数（次）	正确率（%）
A	10	100
B	9	90
C	9	90

5.2.3 驱动测试

测试直线行驶（2 米内）以及左右转弯趋势是否大致相等。记录结果于表 5-5 至表 5-8。

表 5-5 小车前进测试结果（各测 3 次）

	快速	中速	慢速
正确次数（次）	3	3	3

表 5-6 小车后退测试结果（各测 3 次）

	快速	中速	慢速
正确次数（次）	3	3	3

表 5-7 小车左转弯测试结果（各测 3 次）

	I	II	III	IV
正确次数（次）	3	3	3	3

表 5-8 小车右转弯测试结果（各测 3 次）

	I	II	III	IV
正确次数（次）	3	3	3	3

5.2.4 避障测试

避障环境为连续 3 个直角弯和长直走廊。预期小车能够躲避 3 个连续直角弯并在长直走廊中自动避障并朝一个方向前进。测试各进行 3 次，记录结果于表 5-9。

表 5-9 避障测试结果（各测 3 次）

	连续直角弯	长直走廊
正确次数（次）	3	3

5.2.5 循迹测试

循迹的要求是能够在给定的白底黑色跑道上连续跑 10 圈，中断一次该圈计为错误，从下一圈重新开始。测试三次，记录结果于表 5-10。

表 5-10 循迹测试结果（测 3 次，每次 10 圈）

测试次数	正确圈数
I	10
II	10
III	10

5.3测试总结

小车的测试结果非常理想，各项指标达标。尤其是蓝牙芯片接受范围，出乎意料地好，数据手册标称 10 米接受范围，实际测试在走廊里 60 米都能准确识别。但是因为我们的 小车采用开环控制，且轮子之间存在摩擦等因素，如果改变实验环境可能会得到不一样的测试结果。展示视频附在同目录文件夹内，供读者播放观看。

6 总结部分

根据上一章展示，作品在 Micropython 平台实现了驱动、循迹、避障以及语音操控等功能，并最终集成化、定制化的姿态呈现，达到了项目的最初设想。通过这次项目设计，我们对无人控制的载具有了进一步的理解。尽管我们的作品还无法应用与实际生产过程中，我们较为深入地认识到了避障和循迹中的难点和技术点，对电机驱动、蓝牙无线传输也有了进一步的理解。最重要的是，通过本项目，我们第一次真正意义上接触到了集成电路设计的全部流程，大大提高了实践能力和设计能力。

以下分别叙从阶段性进展情况和过程中遇到的困难进行总结。

6.1 阶段性进展情况总结

阶段性进展情况详见附录三工作进度表。纵观整个过程，我们的进度很有特色——“软”快“重”慢。在软件上，得益于 Micropython 的简便性，当然和参与者能力也有关系，我们进展飞快。从收到小车到实现小车驱动和蓝牙控制我们只用了一周时间。但是在硬件上，我们遇到了重重阻碍。因为板子参考文档不足，封装错误，焊接技巧不够，对芯片不够熟悉，对板子集成度要求高等原因，我们反复重新绘图，重新投板，重新购买器件，将进度延误了 6 周。不过所幸的是最终较为成功地完成了最初的设想。在延误的过程中我们实现了播报盒和作品外壳的制作。在这 6 周的过程中也并未一无所获，我们对硬件电路的理解大为深入，焊接技巧得到了提高，发现并解决问题的能力也得到了锻炼，有失有得，不失为一种经历。

6.2 过程中遇到的困难和处理办法

6.2.1 小车的左右侧电机转速不对称

电机本身和输出驱动模块的两路输出电压存在差异性，四个电机的转速很难保持做到一样。因此，在开环的系统中小车没有办法无差地执行指令，比如说直行和右转弯 90°。对于这个问题，有如下三种方案。

1. 使用能够锁定目标的闭环系统，比如摄像头识别目标，获取车身和目标的距离和夹角做出调整；
2. 使用步进电机或测速码盘，保持电机的速度尽可能一致和较精确可调；
3. 提前测量小车左右侧速度在各自不同占空比条件下的速度关系，通过测量转弯

延时来控制转弯角度。

相比之下，第二种方案相对较为合理的，既方便也有较好的效果。我们也确实采购了测速码盘和红外计数器，但是结果并不如预期，同侧前后轮的速度是不一样的，所以一方面我们需要用到四个计数器，需要用到 12 个引脚（每个码盘 3 个），安装上也存在困难（和我们的车架子不兼容），另一方面，即便测出了四个轮子的速度，同样也没有办法知道前后轮是哪个轮子打滑，应该采用哪个速度来进行控制。

因此，尽管我们也知道应该尽量避免延时和烦杂的人工校准，我们最终选择了第三种方案。实测结果显示，就我们的需求而言，较为粗糙的开环控制已经能够满足需求了。

6.2.2 超声波测距的角度限制导致无法做到动态跟踪障碍目标

超声波测距只能测量和超声波模块板成 30° 范围内的平面的距离。也就是说超声波模块将无法准确测量测量方向上的斜面和自己之间的距离。在小车的前进过程中，小车和障碍物的相对位置是不断变化的，超声波能正对障碍物的情况下，障碍物接收到超声波的面不可能一直是平行于超声波模块的平面。这就意味着，我们将无法做到动态测量和任意形状任意形式的障碍物的距离。

作品设想超声波能够动态跟踪障碍物，但是遗憾的是，根据以上分析，这个想法无法实现，而且这个限制是无法避免的。我们只能降低要求，事先规定大致的障碍物的种类针对性的去设计控制转向角度的避障算法。

6.2.3 超声波避障时的角度选择问题

刚开始设计超声波避障时的设计思路是，舵机每旋转 1° ，超声波测距测量一次，并且设定一个危险距离，当超声波测量到的障碍物距离小于安全距离时，进行相应的避障操作。即可以简单理解为若检测到障碍物在右边，就进行左转操作，若障碍物在右边，就进行右转操作。关于障碍物在左还是右的问题，可以通过舵机反馈回来的角度进行判断。

但是，再按照这样的思路设计好程序后，小车却没有按我们想象的方式运作，舵机在疯狂的抽搐式摆动，却没有任何避障操作，连上电脑测试舵机的角度反馈，发现舵机根本没有做到 -75° 到 75° 的旋转（舵机的旋转极限），而是在 -50° 到 40° 左右一度一度的旋转。

资料表明，舵机由一个角度旋转到另外一个角度需要时间，且旋转时间比机器时钟要长的多，如若在舵机旋转期间不加延时函数，舵机做不到一度一度的旋转，而是会不听指令抽搐式转动。但是在每个转动周期内加了一个约 50ms 的延时以后，系统的实

时性大幅度降低，因为舵机旋转一周有 150 个角度值，导致前方出现障碍物了，但是舵机仍没有转到相应的位置，使得避障失败。

在权衡了测量角度的个数和实时性后，我们选择了有代表性的三个角度进行障碍物检测，也就是正前方、左前方和右前方的障碍物检测。能够避开大部分障碍物，对于一些例如狭窄的角落这样不常见的地形，小车不能做出很好的避障操作。

6.2.4 循迹模块过于灵敏导致循迹失误过大的问题

在设计之初，我们将红外传感器安装在车的两侧，我们的循迹思路是如果左侧的传感器检测到黑线，则小车左转；若右侧的传感器检测到黑色，则小车右转。

但是，到赛道上测试时发现，小车会在运动过程中经常原地抽搐无法前进，观察发现，小车抽搐最容易出线的地方就是转弯处，也就是黑线密集且变化较明显的地方。在这个地方小车每时每刻都在检验是否有黑线，导致其不断的选择运动方式，也就会出现抽搐的现象。

为了降低红外检测的灵敏度，我们给红外循迹循环里加了一定时间的延时，这个延时时间不宜过长，过长会影响系统的实时性；也不宜过短，过短起不到调整效果，最后选择 50ms 作为降灵敏度延时。

6.2.5 PCB 制作过程中没有为电机加续流二极管

由于没有设计经验，设计电路的时候没有在直流电机的两端接续流二极管，导致一旦电机停转或者减速芯片就会急剧发烫，甚至烫坏了一片 L298N 这样的大功率驱动芯片。后来仔细研究电路和查阅网上的资料和数据手册，辗转多次，最终猜想并确定是续流二极管的问题。在张璞老师的建议下，我们采用了洞洞板+1N4007 管飞线外接的方式为四个输出分别接上了 2 个续流二极管，解决了这个问题。

6.2.6 焊接困难

这一问题主要是由于操作者技术问题导致的，加上板子面积小，STM32 引脚密集，贴片器件尺寸小等因素，在焊接上确实花费了不少时间精力。最终通过不断焊接调高熟练度得以解决。

7 课程心得体会

7.1 陈钱牛部分

这一学期印象最深刻的就是这辆小车，虽然最终也说不上完美，但也称得上圆满。取得最终的成功是在意料之中，甚至在一开始觉得这对我们俩来说是轻而易举的，但是过程中遇到的困难却着实在意料之外。以下说说过程中遇到的困难、收获以及对课程的意见建议。

之所以说一开始觉得简单是因为我觉得在软件控制上小车其实没有很多太深的东西，我们没有用到很深的算法，也没有用到闭环 PID 控制等等。我们只需要把控制逻辑搞明白了程序自然就写出来了，大不了多调试多改进，而且网上也有很多类似的成功案例（事实上我们并没有参考任何一个案例）。所以在这个过程中我会对我们的代码要求比较高，要求我们必须严格按照面向对象的开发方式来设计，每一个环节都要做好封装，做好注释和说明，同时在细节上也有很多考量。所以如果去看我们的代码的话，会觉得我们的程序其实设计的会比一般的小车要复杂一些，但是代码的结构很清晰，可读性高。完成软件设计的雏形其实我们只用了两周不到的时间，实现了小车的蓝牙控制驱动和避障雏形（避障的思路后来不断调整过）。

真正头大的部分是硬件设计，硬件设计其实我们很早就开始了，我记得在其他组还在学 51 的时候我们就已经在学画原理图了。我们第一次投板也比较早。但是我们的硬件在 14 周才算完成，从第一次投板到完成中间历经了 6 周的时间，完全超乎预期。原理图、PCB 图改了 n 遍（有点强迫症），PCB 投了有 4 次，主控芯片买了 6 个全部用完。这其中的原因主要有以下几点：自我要求高、原理图的规划不够到位、封装对应错误、IC 芯片不熟悉、焊接技术差、粗心大意等。每一次投板我都有一种“这回画的肯定还有问题”的感觉，简直是噩梦。尽管最后一版成功了，但还是存在问题：续流二极管没有设计，所幸最后还是通过强大的飞线外接解决了。

课程的收获很大（付出的时间精力更大，一般 2.5 学分的课哪能让我周内周末都在忙活）。主要的收获有以下 4 点：

1. 熟悉面向对象的开发方式，这一点不管是在电子设计还是在其他很多领域都很有用；
2. 第一次接触并独立完成电子系统开发的全过程，尤其是硬件电路设计；

3. 焊接技巧得到了提高，对 IC 芯片有了新的理解，动手能力大幅度提高；
4. 写报告的能力进一步得到强化，对流程图、框图的绘制更加的规范。

关于意见和建议。该课程可以说是宁改娣、金印彬老师上的数电实验的升级版，同样很好地锻炼了电类方向学生的综合能力。几位带课老师付出的辛勤努力都发挥了巨大的价值，直观地来说在很多方面老师们都已经做到让同学们心悦诚服地程度了。但是我还是有几点建议，反映一些问题（不一定是发生在我们组身上的）。

1. 关于报销的问题。做电子设计一开始一般都是需要买一些模块来学习和使用的，尤其是做 stm32 的话.但是在后面制作 PCB 的时候可能就集成到 PCB 里面了，这样就导致原来的器件闲置了，或者这些模块买来之后发现不会用或者综合考虑后最后没有采用。另外，做 PCB 的时候，需要采购器件，由于缺乏经验，我们很容易就买多了或者买错了。这些部分我觉得学生会花不少冤枉钱，我建议老师们在往后地课程中应该尽早就采购问题专门讲，并在过程中多注意这个问题。其实我们组是花了很多钱，也花了不少冤枉钱，最后肯定有一部分是无法报销的。我觉得老师们应该在这钱方面应该更加重视。
2. 搭档工作分配问题。不是说我们组。我身边有部分组，一个成员很积极，什么事情都一个人在做，另一个人虽然也在做，但是很不积极。然后最后答辩的时候，积极的那位也不好意思说自己干了大部分的活。这种现象确实存在，而且在某些组里还很严重。尤其是考研的和保研的分到一块的时候。甚至有些组做相同的题目的时候整个组不干，借鉴别人组的情况都有。我也不知怎么解决，可能一旦合作就会有人躺这是自然规律吧。
3. 实验室的镊子...每次焊接室去第一件事都是找镊子，以后应该多做一些规定吧。

7.2 李荆晖部分

历时 13 周左右的电子设计课程结束了，我们的作品完成了我们的预期，整个制作过程艰辛与快乐并行，困难时的低落与解决问题时的狂喜充斥着整个过程，可以用 24 字概括此次设计：始于困难选择，过程悲喜交加，搭档强大可靠，终于喜悦成就。

始于困难选择，本作品在早起预想设计时，有许多挑战性的设想和选择，例如，我们最初并没有打算使用 LD3320 的语音识别板，而是选择去了解算法，用算法的方式实现语音识别的功能，的确，我们也进行了约两周的算法学习过程，但是以无果告终，在学业的压力和时间的压力下，我们选择放弃用算法实现语音识别，而是使用专门的外设实现语音识别，虽然很遗憾没有实现语音识别的算法，但是节约出来的时间也给我们更多的时间来丰富小车的其他内容。当然，也有一些正确的选择，我们使用的是基于 Micropython 的开发板，因为陈钱牛同学对 Python 有所涉猎，并且使用起来得心应手，推荐本次作品使用 Micropython，但对我来说，Python 零基础、从零开始学 Python、将 Python 应用于单片机开发都是遥不可及的难题。不过，我选择相信陈钱牛同学的选择，花了大约两周的时间速学 python，事实也证明，我们的选择十分正确，micropython 强大的对象库以及便捷的编程环境为我们的设计带来了许多方便。一学期来，不仅完成了期望中的作品，也学习了 Python，一举两得，甚是欢喜。

过程悲喜交加，一帆风顺的设计过程将是索然无味的，只有经历过遇到困难时的低落，才知道解决困难后的成就感是多么的令人愉悦。本次设计中，我们遇到的困难大大小小数不胜数。因为舵机的偏转角问题以及避障思路问题，在避障的设计中卡了好久，由于我们的设计思路比较少见，网上也没有多少资料可供参考，全凭反复的尝试，舍弃了许多想法，每一次舍弃都是心力交瘁的选择，但随后又会有新想法不断冒出，这些想法就像黑暗中的曙光指引我们最终得到了现在这样的避障设计。当然，也有许多比较顺利的设计，例如循迹的设计，就是我在某个周六的晚上睡不着，爬起来设计，仅花了不到一小时就设计出来的循迹程序，运行的效果十分理想，虽然没有太多的波澜，但是轻而易举的成功也的确给了我不少成就。

搭档强大可靠，本次设计我的搭档陈钱牛同学，能力十分强大可靠，目光长远，做事严谨，一丝不苟，实乃真大腿！陈钱牛同学负责并完成了本项目的大部分工作，对本次设计所做工作十分多。我们共同完成的工作有很多，也是因为他的帮助，使得这次设计变的十分有趣，而不是当成任务去完成。

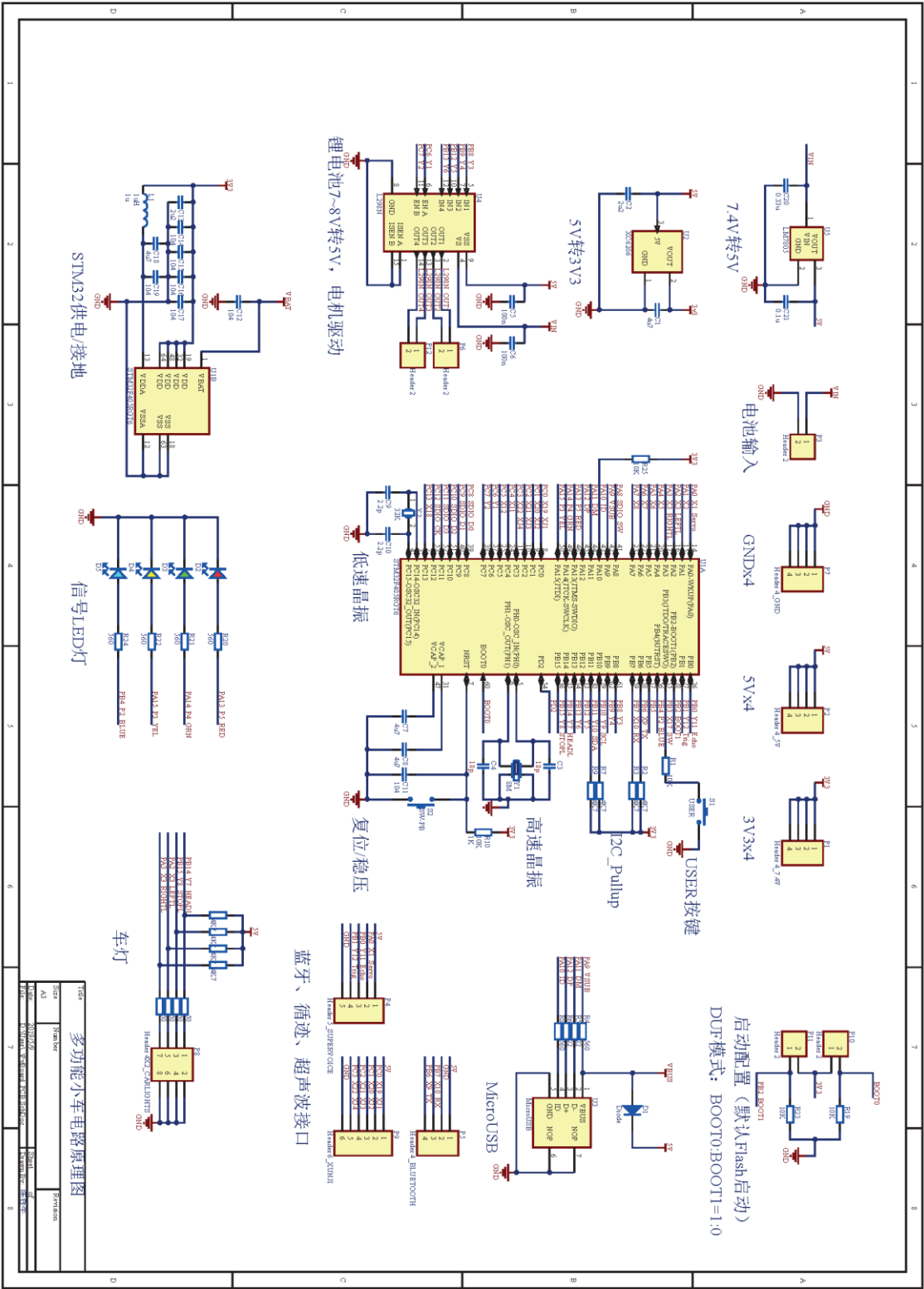
终于喜悦成就，也就是结果十分满意了，我们最后的结果让我十分满意，一辆多功能的小车，可以完成手机控制，可以语音控制，可以完成各种姿势的行走，还可以避障

和循迹行走，这些在开始时想都不敢想的功能，在结束时都一一实现了，喜悦与成就感十分足，另外，也学习了一门新的编程语言，更是被称为未来语言的 Python，也是不小的收获。总的来说，这次设计我十分满意，这次课程我也收获不少，感谢学校的经济支持，感谢老师的指导，更要感谢搭档陈钱牛同学的帮助！

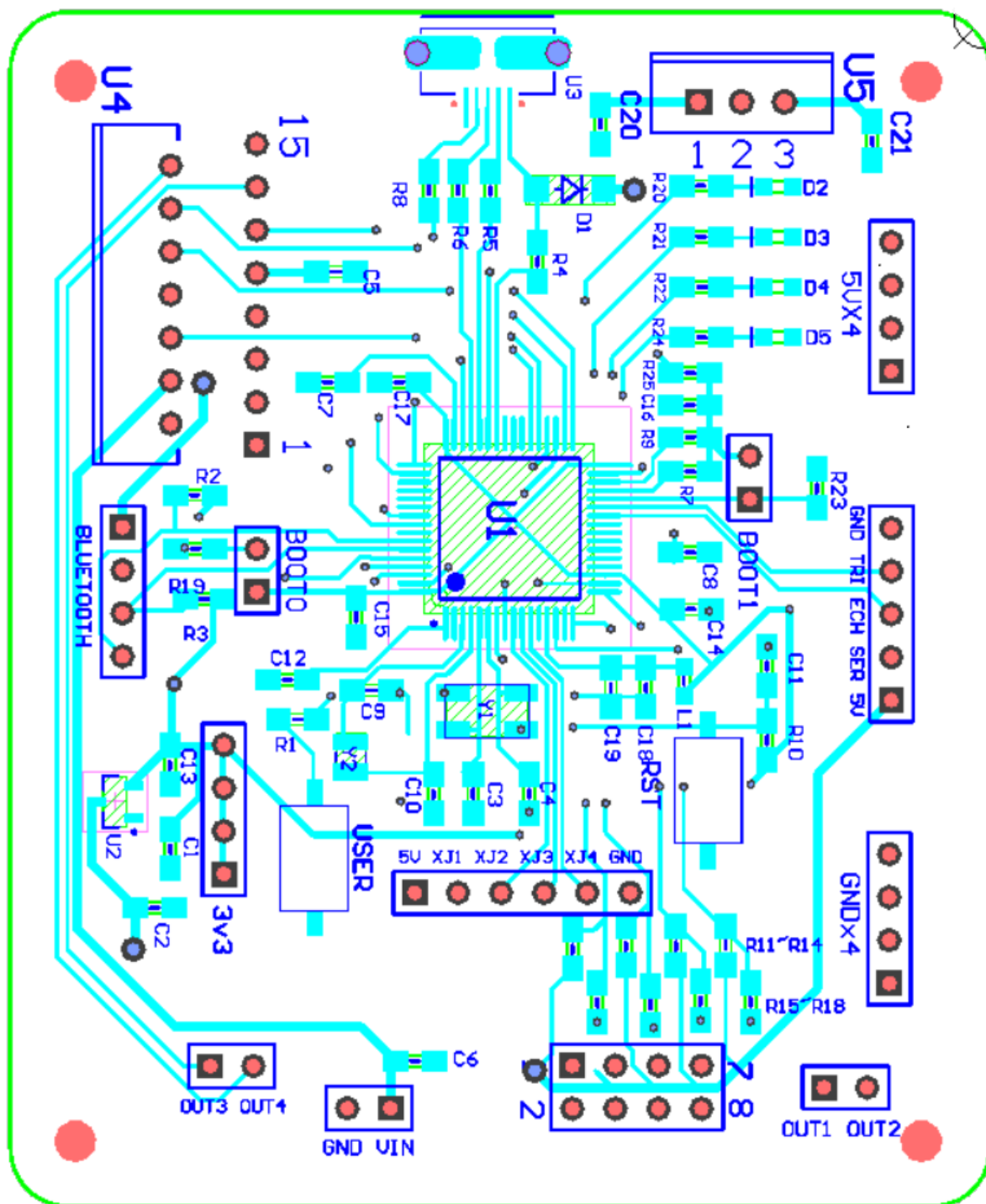
参考文献

- [1]张璞.电路板设计入门[M].西安交通大学电工电子教学实验中心,2019.
- [2]邵子扬.Micropython 入门指南[M].电子工业出版社,2018.
- [3] Donald Norris. Python 微控制器编程从零开始使用 MicroPython,清华大学出版社,2019.
- [4]蒋冬清,王一舒.LD3320 语音芯片与单片机的通讯分析[J].科技创新与应用,2016.
- [5]孙李生.一种基于 51 单片机的超声波测距设计[J].魅力中国,2018.
- [6]张怡,于佳,杨铭, 等.基于 STM32 的智能坦克车系统研究[J].电子设计工程,2018.

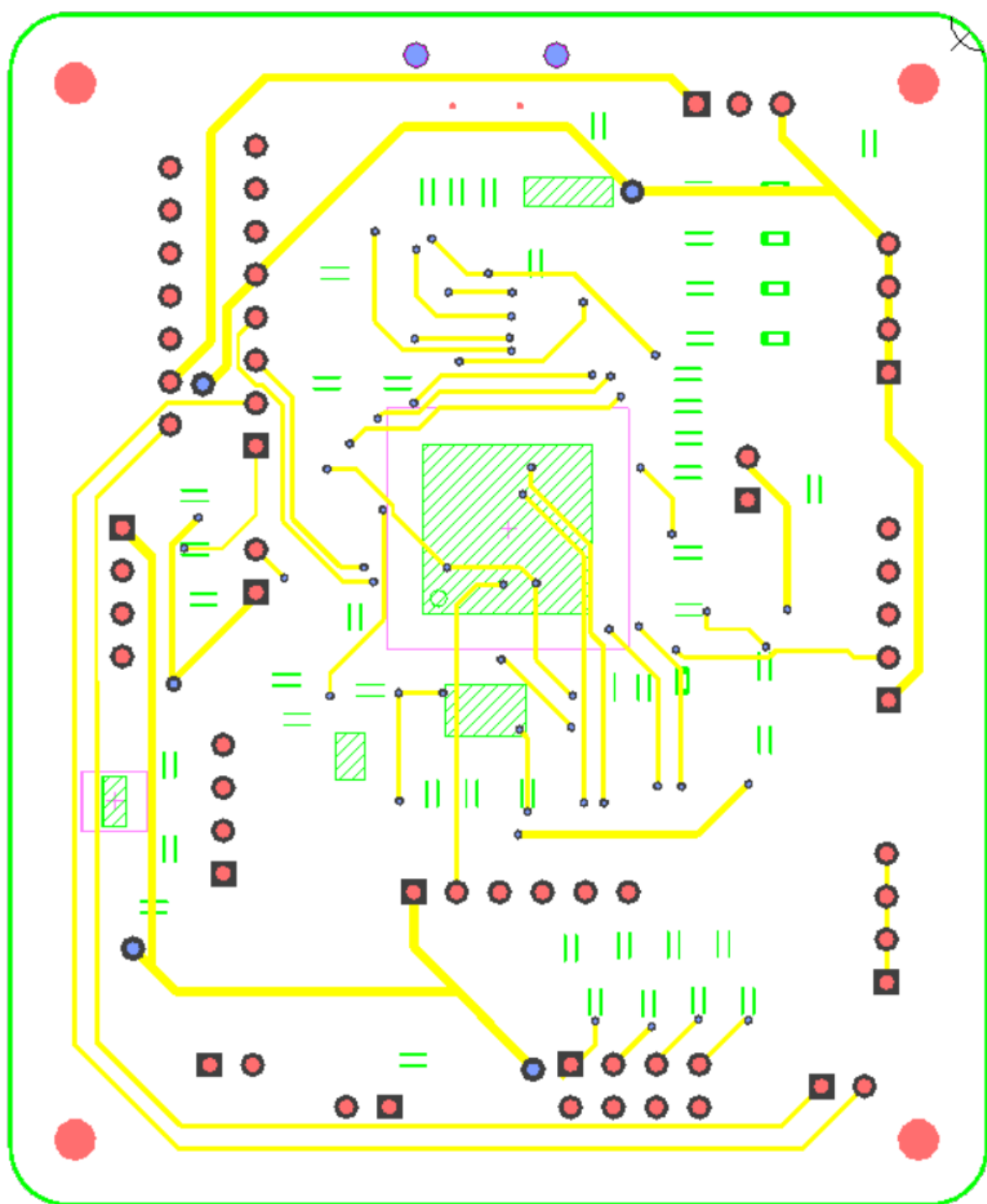
附录一 原理图



附录二 PCB 图



PCB 顶层



PCB 底层

附录三 项目进度完成表

计划	计划时间	完成时间	未完成原因
采购并组装小车基本框架	第三周	第三周	-
实现小车机械功能并封装			
实现 UART 串口（蓝牙）控制小车运动			
完善小车并整合自动避障功能和语音控制	第四周	第四周	-
实现使用语音识别模块蓝牙控制小车	第五周	第五周	-
开始学习 PCB 制版（AD 软件）			
初步完成原理图绘制	第六周	第六周	-
实现循迹功能	无计划	第六周	-
完成 stm32f405 原理图绘制	第七周	第七周	-
制 PCB 版（先准备好，等老师安排）	第八周	第十四周	反复制版焊板
DIY 语音播报盒	无计划	第九周	-
DIY 小车外壳模型	无计划	第九周	-
调试完善、增添屏幕显示功能	第九周	-	后取消计划
调试完善、增添语音播报功能	第十周	-	后取消计划
调试完善，起草报告	第十一周	第十四周	尚未完成制版
调试完善，完成报告	第十二周	-	-

附录四 使用说明

本作品具备遥控、红外循迹和超声波避障功能。可以通过手机蓝牙或语音通讯盒连接蓝牙操控。

功能说明

其中超声波避障需要测试地形开阔，障碍物形式不过于复杂，如走廊或预置的较为垂直于小车运动方向的障碍面。红外循迹需要在具有白底黑色轨迹带的地面上使用，并提前将小车位置调整至轨迹带位于两侧红外探头中间。

蓝牙连接控制




1. 手机控制 小车通电（将电池插到小车对应的 VIN GND 口），PCB 板上的蓝牙指示灯闪烁进入配对模式。手机搜索名为“ljh946”的蓝牙，配对码为 1234。成功连接后可以收到“I am ready”的反馈信号。输入下表左半部分指令可以控制小车。
2. 通讯盒控制 小车和通讯盒都上电，检查两者蓝牙指示灯均不闪烁。使用较为清晰的语音向通讯盒说“输入指令”。如果通讯盒红色指示灯亮，说明识别成功，否则再次尝试。红灯亮后，说出下表右半部分所列命令即可控制小车运动。

手机输入指令信息	对应命令	通讯盒输入指令信息	对应命令
0	刹车；退出当前模式	刹车	刹车；退出当前模式
11~13	前进 1~3 挡	前进	前进 3 挡
21~23	后退 1~3 挡	后退	后退 3 挡
31~34	左转弯四挡	左转弯	大角度左转弯
41~44	右转弯四挡	右转弯	大角度右转弯
5	避障模式	避障模式	进入避障模式
6	循迹模式	循迹模式	进入循迹模式

程序源代码查看和修改

使用 MicroUSB 数据线连接 PCB 板和计算机，计算机上会弹出名为“PYBFLASH”的磁盘，大小约为 95.0KB。打开磁盘既可以看到如下图所示两个文件和一个文件夹。boot.py 中可以查看芯片的启动信息，包括可以修改其启动文件的名称和其他参数，一般默认不修改。main.py 为主函数。ClassLib 文件夹内包含了程序设计中使用到的 12 个类，可以查看和修改其定义。需要注意的是，每次需改完成保存需要等待开发板红色指示灯

灭后才能进行下一步操作，否则如果在读写过程中操作很可能会出错，需要重置操作，数据全部初始化。

 ClassLib	2019/4/9 17:44	文件夹	
 boot.py	2015/1/1 0:00	PY 文件	1 KB
 main.py	2019/4/10 8:25	PY 文件	1 KB

REPL（交互式解析器）实时调试程序

REPL 是 Micropython 的一个具有特色的部分，其作用是你无需编写玩程序再进行运行而是可以输入一条指令让 MCU 立即执行。同样我们可以通过 REPL 实时观测 MCU 的状态，来进行程序调试。它也相当于一个 MCU 端的串口通讯助手，我们可以实时查看到 MCU 接收到的串口信号。使用方法是使用 Putty 等 SSH 远程控制工具，找到对应的 COM 端口，即可连接访问。

其他关于本项目的相关的资料已经上传到 Github

https://github.com/zjcqn/car_project