

浙江工业大学

计算机工程实践报告



题目 智能寻物小车

成员姓名 黄启明、邹蓬麦子、张毅涛、张栩宁

班 级 软外 2103

教师评语：

评语及成绩：（此部分老师填写）

不跨到下一页。先对全组进行整体评价（含作品主要功能、亮点和不足等），然后对各个成员的表现分别进行评价。

团队分工与贡献度说明

成员姓名	主要分工	总贡献度 (%)	备注
黄启明	1. 完成了环境搭建； 2. 完成了对项目进度的统筹安排； 3. 完成了系统的整体的构思； 4. 完成了系统通信模块的设计与调试； 5. 完成了视频录制； 6. 完成了答辩 PPT 背景意义和系统介绍部分的设计； 完成了实验报告的感悟部分。	25.5%	组长
邹莲麦子	1. 完成了环境搭建； 2. 完成了系统的图像识别方面的构思； 3. 完成了系统图像识别模块的设计与调试； 4. 完成了视频录制、剪辑与配字幕； 5. 完成了答辩 PPT 系统介绍部分的设计； 完成了实验报告的特色与亮点、感悟部分。	24.8%	组员
张毅涛	1. 完成了环境搭建； 2. 完成了系统的语音识别方面的构思； 3. 完成了系统基本路线功能、语音识别模块/部分的设计/调试； 4. 完成了视频录制； 5. 完成了答辩 PPT 总结与展望部分的设计； 完成了实验报告的第总结与展望、感悟部分。	24.9%	组员
张栩宁	1. 完成了环境搭建； 2. 完成了流程图设计； 3. 完成了系统的寻物路径策略方面的构思； 4. 完成了系统基本路线功能、语音识别模块/部分的设计/调试； 5. 完成了视频录制； 6. 完成了实验报告的系统简介部分。	24.8%	组员

一、系统简介、主要功能、特色与亮点

1.1 系统简介

语音识别自动寻物系统，用于在一定空间内，在使用者语言指令的控制下，小车能够从语言中提取关键词，从而利用计算机视觉，去寻找目标寻找物。

(1) 应用场景。

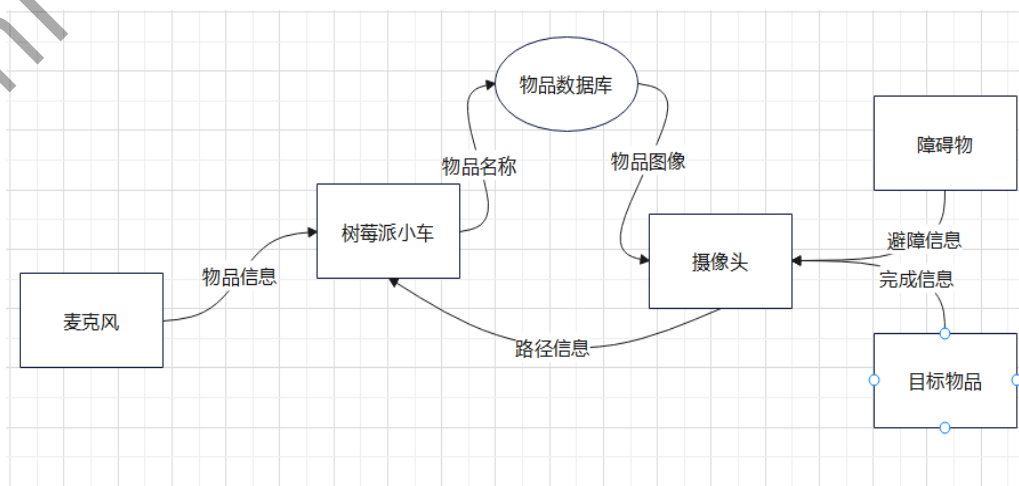
物流仓库：在物流仓库中，可以使用该功能来自动搜寻和捕捉需要移动的货物，并将其运输到指定的位置，以此来减少人力物力，减轻资金和劳动力负担。

智能家居：在智能家居中，可以使用该功能来寻找遗失的物品，如手机、钱包等，并且可以向家居的其他成员传递信息，以此来简化我们日常生活中的一些琐事，节省时间。

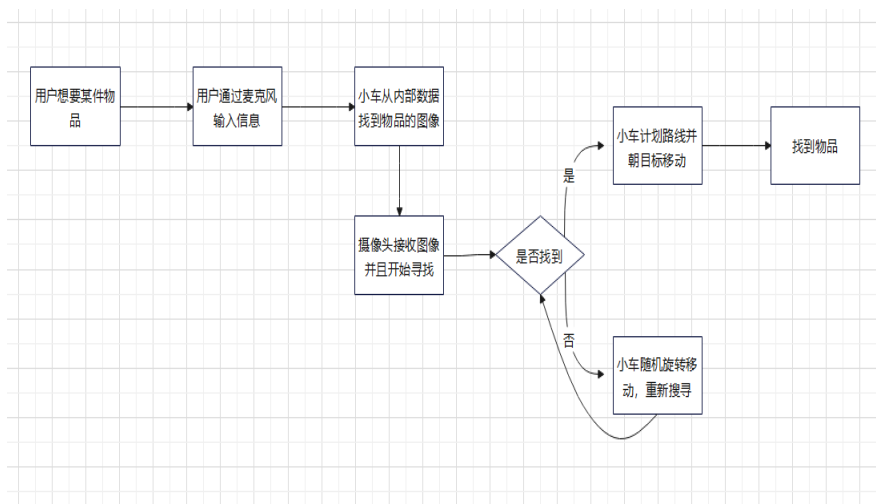
安防系统：在安防系统中，可以使用该功能来自动搜寻和捕捉入侵者，增加安防系统中物力，从而提高区域的安全性。

(2) 业务逻辑。

1. 小车启动，并准备接收语音输入指令。
2. 用户使用语音输入指令，提供要寻找的物体的名称。
3. 小车接收语音输入，并将提供的物体名称作为目标物体。
4. 小车开始进行目标图像生成。它在嵌入的物体图像库中搜索与目标物体名称对应的图像。
5. 小车使用搭载的摄像头进行目标检测。它捕获当前环境的图像，并使用图像识别技术来识别特定的物体。
6. 小车分析目标检测结果，判断是否检测到了目标物体。
7. 如果目标物体检测成功，小车开始自主导航。它根据目标物体的位置信息，计算出前进的方向和距离。
8. 小车开始朝向目标物体前进，并在导航过程中持续进行目标检测，以确保正确地朝向目标。



数据流传递图



业务逻辑图

1.2 系统主要功能

- (1) 能够识别出用户的语音输入内容并回答，并且提取一个关键词并保存。
- (2) 系统可以通过物体的名称提取出该物体的图像。
- (3) 小车的寻找是对当前环境巡视一圈，类似于人在没有线索时找东西的过程中扫视周围环境。

1.3 特色与亮点

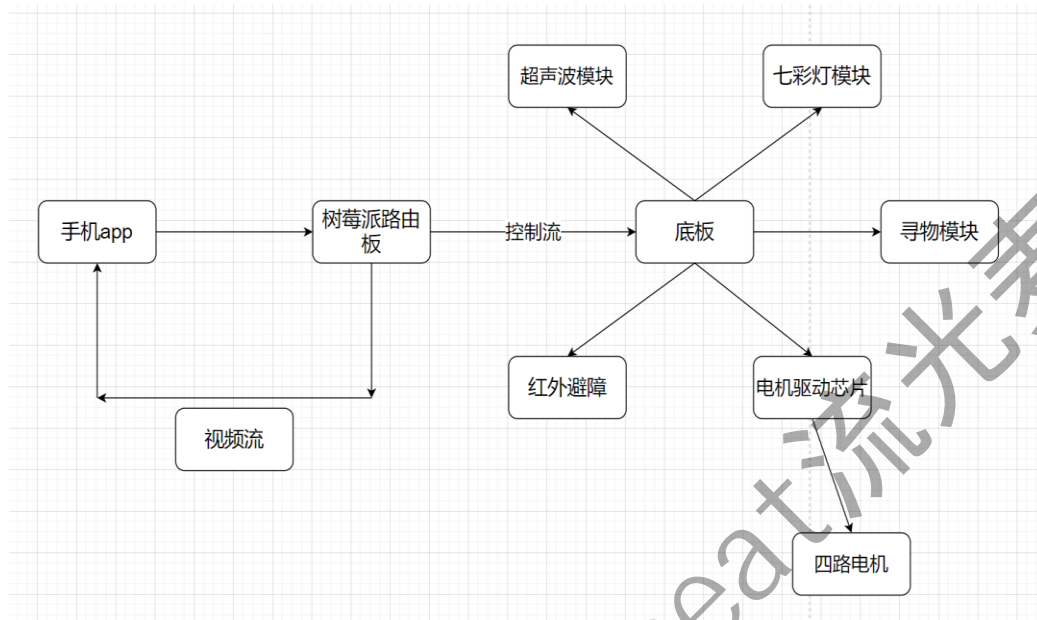
- (1) 语音输入控制：用户可以通过语音指令与小车进行交互，提供需要寻找的物体名称。
- (2) 目标生成：小车通过搭载的摄像头每 2 秒获取一张图像，并使用嵌入的 YOLOv3 模型对图像进行目标检测，从中找到与目标寻找物体相对应的物体名称。
- (3) 目标检测：小车的摄像头使用先进的图像识别技术，能够准确地识别出特定的物体，并根据训练好的模型和算法进行目标检测。
- (4) 自主导航：小车根据目标检测结果实现自主导航功能。它可以根据检测到的目标物体确定自己的位置和朝向，并使用路径规划算法决定如何前进以到达目标物体所在位置。
- (5) 导航过程中，小车会实时更新目标检测结果，确保始终朝向正确的目标物体。如果目标物体位置发生变化，小车将相应地调整自身导航路径，以确保准确抵达目标。

二、系统硬件设计与实现

2.1 系统硬件总体结构

树莓派 wifi 智能小车是利用 PC 或者手机作为控制端，通过手机连接树莓派路由器以获取 wifi 信号，树莓派的底板通过排线与树莓派主控板连接起来，树莓派的底板上就连接相应的外设。从而实现

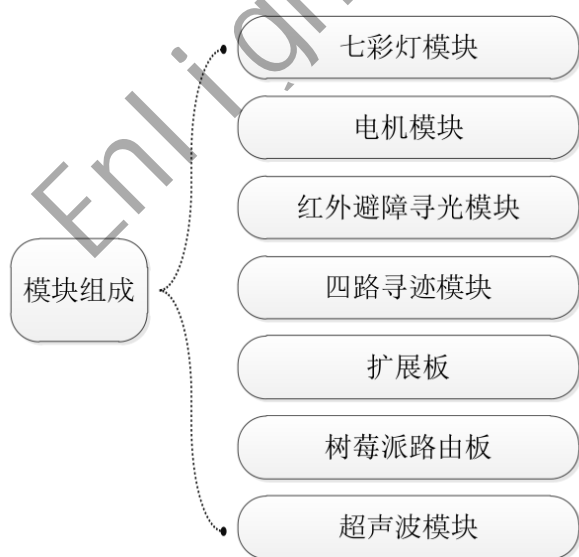
手机和小车的无线连接。我们通过手机上的 android 软件以 wifi 作为信息载体。通过 socket 套接字的 TCP 协议进行通信，将指令发送给树莓派路由器，树莓派通过解析相关的手机发送过来的指令，并设置相应的 GPIO 口，以实现相应的避障，跟随，巡线，寻光，灭火，点灯等功能。同时，我们树莓派路由器上连接上了 usb 摄像头，获取摄像头所采集的图像信息数据实时的通过 wifi 发送给我们的上位机或者手机，最后我们就可以通过手机控制小车和完成远程监控。



2.2 系统核心模块

树莓派 wifi 智能小车主要由七彩灯模块、电机模块、超声波模块、红外避障寻光模块、四路寻迹模块、底板和树莓派路由板构成，如图 2 所示。

下面分模块介绍我们的智能小车所具备的结构及模块功能。



①七彩灯模块

图 3 为七彩灯模块示意图。七彩灯通过电平的变化七彩灯模块可产生各种颜色（共可产生 255*255*255 种颜色）。七彩灯原理图如下图 4 所示。

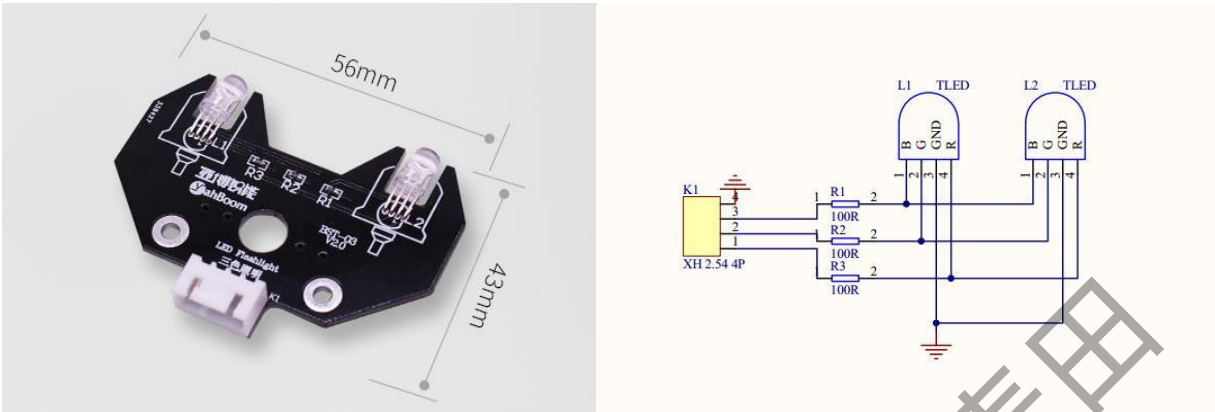


图 3 七彩灯示意图

图 4 七彩灯原理图

②电机模块

对于 4 路直流减速电机的控制采用的是 TB6612FNG 驱动芯片来驱动电机。通过控制驱动芯片的 AIN1,AIN2,BIN1,BIN2,PWMA,PWMB 的电平高低来控制电机的正转，反转，停止。

- STBY 置 0:电机全部停止;置 1:使能驱动模块,通过 AIN1/AIN2/BIN1/BIN2 控制正反转。
- VM 接 15V 以内电源;VCC 接 5V,GND 接电源负极。

TB6612FNG 双驱动芯片如图 5 所示，其芯片管脚如图 6 所示。



图 5 TB6612FNG 双驱动芯片

图 6 双驱动芯片管脚

驱动说明如图 7 所示。

驱动1路

PWMA 接单片机的PWM口

真值表:

AIN1	0	0	1
AIN2	0	1	0

停止 正转 反转

AO1 接1电机的正极

AO2 接1电机的负极

驱动2路

PWMB 接单片机的PWM口

真值表:

BIN1	0	0	1
BIN2	0	1	0

停止 正转 反转

BO1 接2电机的正极

BO2 接2电机的负极

图 7 驱动说明示意图

③超声波模块

超声波模块是利用超声波特性检测距离的传感器。其带有两个超声波探头，分别用作发射和接收超声波。其测量的范围是 3-450cm。图 8 为组装小车所需零件，由超声波传感器、配对数据线、固定铜柱及螺丝组成。

图 9 为超声波模块实体及其管脚示意图。图 10 为其引脚说明。



图 8 超声波模块零件图



图 9 超声波模块实体及其管脚示意图

引脚名称	说 明
Vcc	电源 5 V
Trig	触发引脚
Echo	回馈引脚
Gnd	地

图 10 超声波引脚说明图

测距功能触发后，模块将自动发出 8 个 40kHz 的超声波脉冲，并自动检测是否有信号返回，这一步由模块内部自动完成。一旦检测到有回波信号则 ECHO 引脚会输出高电平。高电平持续的时间就是超声波从发射到返回的时间。此时可以使用时间函数计算出 ECHO 引脚高电平的持续时间，可计算出距被测物体的实际距离。取声速 340m/s，此时即可计算出实际的距离：

$$\text{距离} = \text{高电平时间} \times \text{声速}(340\text{M/S}) / 2$$

④红外避障寻光模块

红外避障原理：该传感器具有一对红外发射管和接收管，发射管在一定的范围内发射一定频率的红外线。如果没有障碍物，发射出去的红外线，会因传播的距离越来越远而逐渐减弱，最后消失。若有障碍物，红外线反射回来后会被接收管所接收。通过检测相关引脚的电平状态，让小车做出相应的动作。

寻光原理：光敏电阻是利用半导体的光电导效应制成的一种电阻值随入射光的强弱而改变的电阻器，又称为光电导探测器；入射光强，电阻减小，入射光弱，电阻增大。通过光敏电阻连接到主控板上的引脚的电平变化来决定是否有光，从而让小车做出相应的动作，来达到寻光的效果。

图 11 为红外传感器组装零件，由红外线传感器、配对数据线、固定铜柱及螺丝组成。



图 11 红外传感器零件图

图 12 为红外避障寻光模块。



图 12 红外避障寻光模块图

图 14 四路寻迹模块图

⑤舵机原理

若要控制舵机，控制信号由接收机的通道进入信号调制芯片，获得直流的偏置电压。它内部有一

个基准电路，产生周期为 20ms，宽度为 1.5ms 的基准信号，将获得的直流偏置电压与电位器的电压比较，获得电压差输出。最后电压差的正负输出到电机驱动芯片决定电机的正反转。当舵机转速一定时，通过级联减速此轮带动电位器旋转，使得电压差为 0，电机停止转动。

各模块连接：

每个硬件都有其对应的接口，注意将每一个硬件都合理妥当的接入小车主板。

如下图 15 为小车模块连线图。下图 16 为 TB6612FNG 驱动芯片物理接线图。

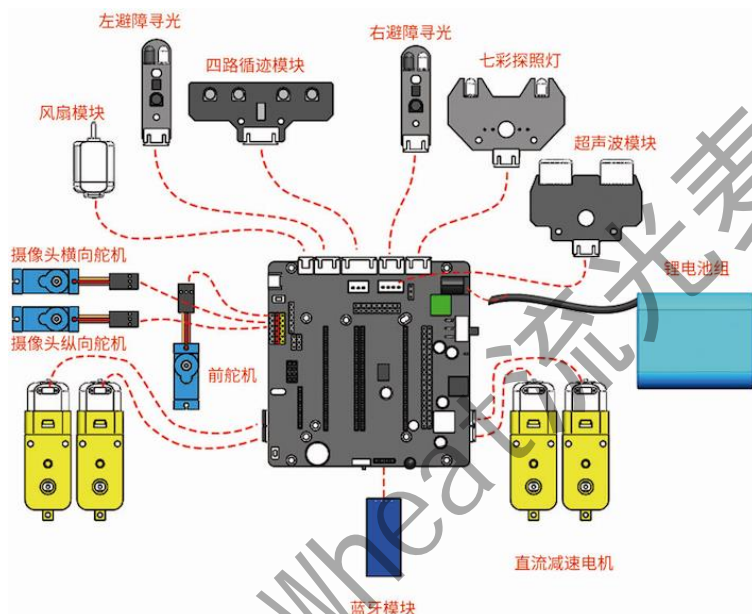


图 15 模块连线图

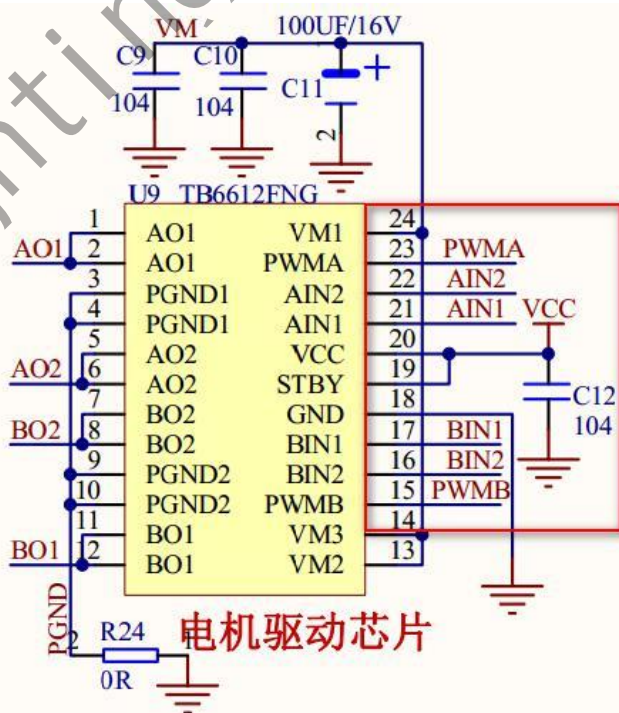


图 16 TB6612FNG 驱动芯片物理接线图

实验所用的 4B 小车主板结构、接口如图 17。

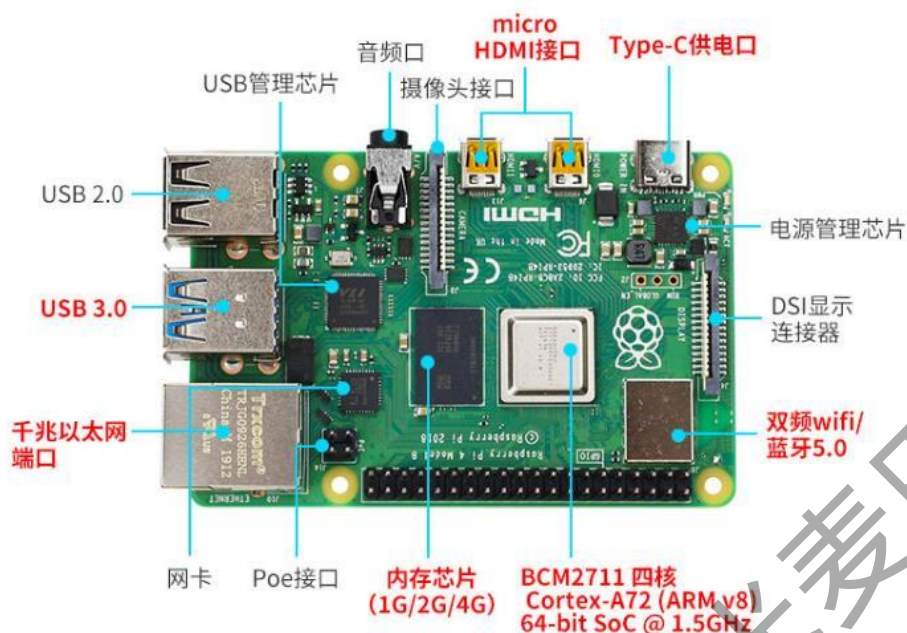


图 17 4B 小车基本结构图

- USB 端口:它们用于连接鼠标和键盘。还可以连接其他组件,如 USB 驱动器。
- 以太网端口 :用来连接树莓派到网络与电缆。树莓派还可以通过无线局域网连接到网络。
- RCA av 端口音频输出:可以在这里连接耳机或扬声器。
- HDMI 接口:这是连接显示器(或投影仪)的地方,您使用它来显示树莓派的输出。如果你的显示器有扬声器,也可以用它们来听声音。
- 按钮:连接到树莓派。
- 摄像头接口:用于连接摄像头。
- LED 系统指示灯:指示树莓派各模块运行情况。
- DSI 显示连接器:用于连接显示屏。
- 双频 802.11 ac 无线局域网蓝牙 4.2:用于蓝牙连接。
- 以太网(Poe) 供电:用于以太网(Poe) 供电。
- RAM:用于存储树莓派运行时的数据。
- USB 电源接口:用于 USB 供电。

根据相应的图示和相应的插口,完成整车的组装。



图 19 实物连接 1 图

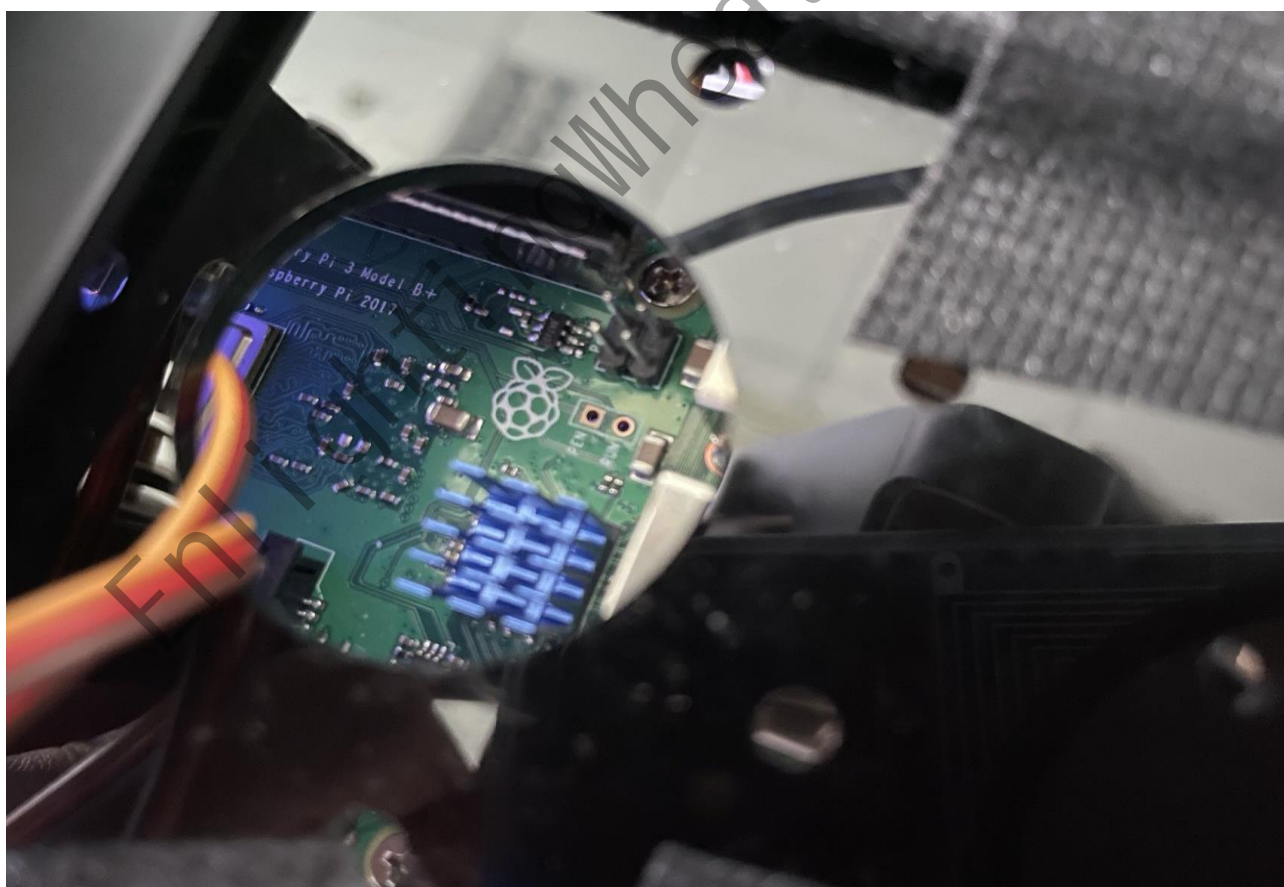


图 20 实物连接 2 图

⑥摄像机模块

该树莓派小车采用的是 4WD 二自由度摄像头，组件包括一个摄像头和两个舵机，并将该摄像

头固定在一个二层平台上，如图 21：

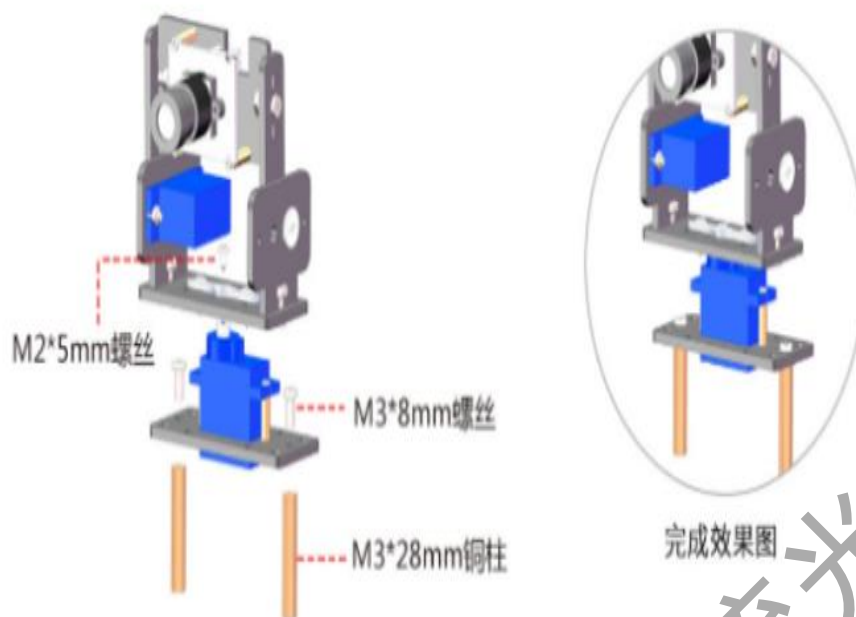


图 20

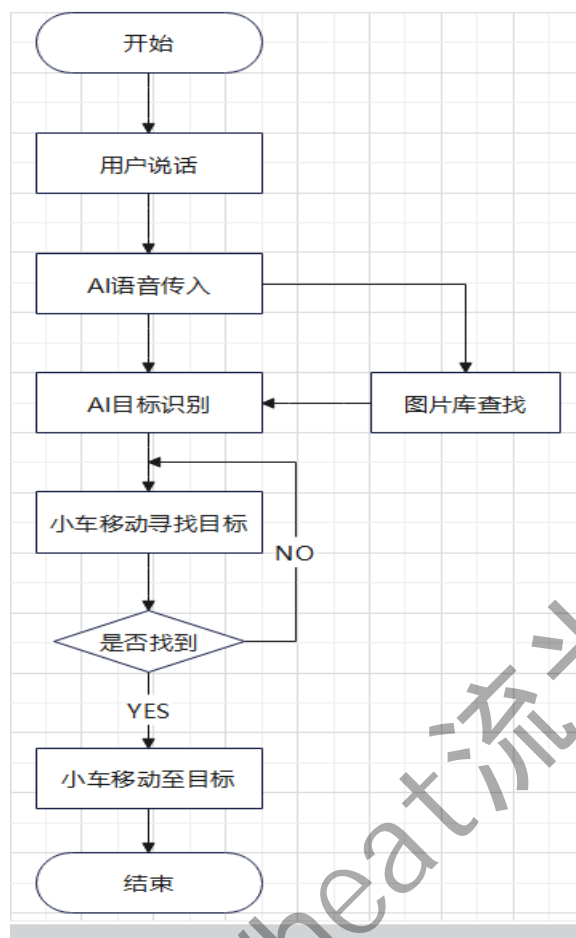
该摄像头通过部分的代码可以实现由舵机和电机的直接控制。

三、系统软件设计

3.1 系统软件总体设计思路

程序刚开始，用户通过麦克风说出想要小车寻找的目标物体，语音被传入至 AI 语音程序系统，该程序拥有 Chatgpt 接口和关键字提取模块，因此程序会将想要寻找的目标物体以文本的形式传输给 AI 目标识别程序部分，之后用 YOLOv3 模型识别小车摄像头所拍摄图片中的物体与目标物体进行比对，并将图像传输给小车，小车识别了图像后开始控制摄像头和舵机进行随机搜寻，若捕捉到目标物体的图像，则启动移动装置前往目标物体，若没有捕捉到目标物体的图像，则继续进行随机搜寻，直到捕捉到目标物体的图像为止，最后小车移动至目标物体，程序结束。

程序流程图如下：



程序流程图

3.2 系统核心功能实现

(1) 语音识别功能

语音识别模块能够识别出用户的语音输入内容并回答，并且提取一个关键词并保存。

语音识别功能的实现通过语音输入与用户进行交互，并使用 OpenAI GPT-3 模型生成相应的回复。同时，它还使用 SpeechRecognition 库将语音转换为文本，使用 gTTS 库将文本转换为语音，并使用 pygame 库播放生成的语音。通过这些功能，实现了一个简单的语音交互人工智能助手。

(2) 图像识别功能

图像识别模块可以通过物体的图像提取出该物体的名称。

采用 yolo3_darknet53_coco 模型，给定的图像进行目标检测。它首先加载并预处理图像，然后通过模型获取目标的类别 ID、置信度分数和边界框坐标。然后，它读取一个文本文件中的目标名称，并与检测到的目标进行匹配。如果匹配成功，它返回与目标相关的信息；如果未找到匹配的目标，它返回一个无关的目标的信息。如果出现异常（比如图像损坏），它返回 0。

(3) 实时通信功能

小车与计算机有实时通信，根据计算机的实时通信改变路线。

树莓派小车段与计算机端实现图片通信的功能实现是将指定的图像文件通过 socket 连接发送到服务端。它通过 OpenCV 读取摄像头捕获的帧，保存为图像文件。然后，它使用 socket 对象将文件头信息和文件数据发送到指定的服务器地址和端口。通过这个过程，实现了文件上传的功能。

(4) 路线规划功能

小车的寻找是对当前环境巡视一圈，类似于人在没有线索时找东西的过程中扫视周围环境。同时，路线策略模块的功能实现是根据目标在图像中的位置和大小，控制小车的运动。根据目标在图像中的位置和大小，进行不同的运动操作，使得小车能够根据目标的位置调整自己的位置和朝向目标。其中，使用了一些条件判断语句来确定需要执行的运动操作，如前进、后退、左移、右移等，以及根据目标在图像中的位置和大小计算运动的时间。最后，通过调用相应的移动函数实现小车的运动。

3.2.1 语音模块

语音输入与识别的功能使用 OpenAI 的 GPT-3 模型、pygame、speech_recognition 和 gtts 等库来实现语音识别、语音合成和与 GPT-3 模型进行交互的功能。代码通过监听用户的语音输入，提取关键词并翻译成英文，然后根据关键词生成相应的回复或执行特定操作，最后将回复转换为语音进行播放。这样实现了一个简单的语音交互助手，结合了语音识别、自然语言处理和语音合成等技术。

<https://highlightcode.com/>

```
1. # -*- coding: utf-8 -*-
2. import openai
3. import pygame
4. import os
5. import speech_recognition as sr
6. from gtts import gTTS
7. import time
8.
9. # Replace with your GPT-3 API key
10. openai.api_key = "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"
11.
12. def generate_gpt3_response(prompt):
```



```
13.     response = openai.Completion.create(  
14.         engine="text-davinci-002",  
15.         prompt=prompt,  
16.         max_tokens=100,  
17.         n=1,  
18.         stop=None,  
19.         temperature=0.5,  
20.     )  
21.  
22.     return response.choices[0].text.strip()#GPT-3 生成的第一个响应文本的字符串返回，并去除开头和结尾的空白字符。  
23.  
24. def speak(text, language="zh-cn"):  
25.     tts = gTTS(text=text, lang=language)  
26.     tts.save("response.mp3")  
27.  
28.     pygame.mixer.init()#初始化pygame.mixer 模块  
29.     pygame.mixer.music.load("response.mp3")#方法加载response.mp3 音频文件  
30.     pygame.mixer.music.play()#播放该音频文件  
31.  
32.     while pygame.mixer.music.get_busy():  
33.         pygame.time.Clock().tick(10)#当音频正在播放时，  
           pygame.mixer.music.get_busy()方法将返回True，while 循环会不断调用tick()方法等待音频播放完毕。  
34.  
35.     # 等待 3 秒，确保文件不再被占用  
36.     time.sleep(3)  
37.  
38.     os.remove("response.mp3")  
39.  
40. def recognize_speech(retries=3):  
41.     recognizer = sr.Recognizer()  
42.     for _ in range(retries):#默认尝试3次麦克风识别  
43.         with sr.Microphone() as source:#打开麦克风
```

```
44.         print("请说话...")
45.         audio = recognizer.listen(source)#Recognizer 对象监听来自麦克
           风的语音输入，并将其存储在 audio 变量中以备后续的语音识别
46.
47.         try:
48.             text = recognizer.recognize_google(audio, language="zh-CN")
           #将存储在 audio 变量中的语音转换为文本
49.             print(f"你说的是: {text}")
50.             return text
51.         except sr.UnknownValueError:
52.             print("Google Speech Recognition 无法识别语音")
53.         except sr.RequestError as e:
54.             print(f"无法从 Google 语音识别服务请求结果; {e}")
55.
56.     return None
57.
58. def aispeak():
59.     while True:
60.         spoken_command = recognize_speech()
61.
62.         if spoken_command is not None:
63.             # Get a keyword from the spoken command and translate it to
           English
64.             keyword_prompt = f"从 \"{spoken_command}\" 中提取一个关键词并
           翻译成英文"
65.             keyword = generate_gpt3_response(keyword_prompt)
66.             print(f"Keyword: {keyword}")
67.
68.             if keyword.lower() == "joke":
69.                 joke_prompt = f"请用中文给我讲一个笑话"
70.                 joke = generate_gpt3_response(joke_prompt)
71.                 speak(joke)
72.             elif keyword.lower() == "song":
73.                 if "猜猜" in spoken_command:
```

```
74.             guess_prompt = f"猜猜我最喜欢的歌曲"
75.             guess = generate_gpt3_response(guess_prompt)
76.             speak(guess)
77.         else:
78.             song_prompt = f"请告诉我今年的十大热门歌曲"
79.             top_song = generate_gpt3_response(song_prompt)
80.             speak(top_song)
81.         elif "小车" in spoken_command:
82.             response_prompt = f"请根据 \"{spoken_command}\" 生成一个
            合适的回复"
83.             response = generate_gpt3_response(response_prompt)
84.             speak(response)
85.
86.         else:
87.             if "相关" in spoken_command:
88.                 content_prompt = f"请根据关键词 \"{keyword}\" 生成一段
                有关的内容"
89.                 content = generate_gpt3_response(content_prompt)
90.                 speak(content)
91.             else:
92.                 modified_response = f"我在找 {keyword} 快让我看看 "
93.                 speak(modified_response)
94.                 return keyword
95.         # 暂停 5 秒
96.         time.sleep(5)
97.
98. if __name__ == "__main__":
99.     result=aispeak()
100.    print(f"Returned: {result}")
```

3.2.2 图像识别模块

图像识别模块实现了一个目标检测功能。它首先加载预训练的 YOLOv3 模型，并将待检测的图像进行预处理。然后，通过模型对图像进行推理，得到目标的类别 ID、置信度分数和边界框信息。接着，从指定的文件中读取目标的名称。代码会根据目标名称在检测结果中寻找匹配的目标，并返回该目标

的类别 ID、置信度分数和边界框信息。如果未找到匹配目标，代码会返回默认类别 ID、置信度分数和边界框信息。如果在图像预处理或目标检测过程中出现异常，代码会捕获异常并输出相应的错误信息。同时对于一个物体的识别，不是返回多个识别结果，而是返回多个结果，比较得出相似度得分最高的一个结果。

```
1. def scan(dir):
2.     net = model_zoo.get_model('yolo3_darknet53_coco', pretrained=True,
        root='D:\\AI_project\\gluoncv\\model')
3.     try:
4.         x, img = data.transforms.presets.yolo.load_test(dir, short=512)
5.         print('Shape of pre-processed image:', x.shape)
6.         class_IDs, scores, bounding_boxes = net(x)
7.         # print('class:', class_IDs[0][0])
8.         # print('scores:', scores[0][0])
9.         # print('bounding_box:', bounding_boxes[0][0])
10.        with open("D:/Desktop/animal_follower/PC/target.txt", "r", encoding="utf-8") as f:
11.            target=f.read()
12.            print("我在找"+target)
13.            targetIDs=selectID(class_IDs)
14.            for key in targetIDs:
15.                if net.classes[targetIDs[key]]==target:
16.                    print("我看到了",target)
17.                    return class_IDs[0][key],scores[0][key],bounding_boxes[0][key]
18.            # plt.show()
19.            print("没有找到目标")
20.            return class_IDs[0][50],scores[0][50],bounding_boxes[0][50]
21.        except:
22.            print("刚刚那张图片损坏了")
23.        return 0
```

3.2.3 通信模块

小车将图片传给 PC:

首先创建一个 TCP/IP 类型的 socket 对象,并通过该 socket 对象连接到指定的服务器地址和端口。然后,它使用 OpenCV 打开一个视频流,并读取视频的帧。接下来,代码将视频帧保存为图像文件。之后,代码进入一个循环,将图像文件的路径、文件头信息和文件数据通过 socket 连接发送给服务器。循环中的每一次迭代,代码将读取文件数据并发送,直到文件数据全部发送完成。最后,关闭与服务器的连接。这样实现了将指定的图像文件通过 socket 连接上传到服务器的功能。

```
1. import socket
2. import os
3. import sys
4. import struct
5.
6. def upload():
7.     try:
8.         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9.         #192.168.199.1 和 8088 分别为服务端 (pc) 的 IP 地址和网络端口
10.        s.connect(('192.168.50.107', 8086))
11.    except socket.error as msg:
12.        print(msg)
13.        print(sys.exit(1))
14.
15.    video = cv2.VideoCapture("http://192.168.50.1:8080/?action=stream")
16.    cv2.namedWindow("img",cv2.WINDOW_NORMAL)
17.    ret,frame = video.read()
18.    time.sleep(2)
19.    if(frame is None):
20.        print("2")
21.    cv2.imwrite("/home/pi/Desktop/gluoncv/image.png", frame)
22.
23.    while True:
24.        #filepath 是要被发送图片的路径
25.        filepath = '/home/pi/Desktop/gluoncv/image.png'
```

```
26.         fhead = struct.pack(b'128s1', bytes(os.path.basename(filepath),
        encoding='utf-8'), os.stat(filepath).st_size)
27.         s.send(fhead)
28.         print('client filepath: {0}'.format(filepath))
29.
30.         fp = open(filepath, 'rb')
31.         while 1:
32.             data = fp.read(1024)
33.             if not data:
34.                 print('{0} 发送成功...'.format(filepath))
35.                 break
36.             s.send(data)
37.             s.close()
38.             break
39. returnn 0
```

PC 将有关识别结果的 txt 文件传给小车：

首先创建一个 TCP/IP 类型的 socket 对象，并通过该 socket 对象连接到指定的服务器地址和端口。然后，代码进入一个循环，将指定文本文件的路径、文件头信息和文件数据通过 socket 连接发送给服务器。循环中的每一次迭代，代码将读取文件数据并发送，直到文件数据全部发送完成。最后，关闭与服务器的连接。这样实现了将指定的文本文件通过 socket 连接发送给服务器的功能。

```
1. def send_txt():
2.     try:
3.         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4.         # 192.168.199.1 和 8088 分别为服务端 (pc) 的 IP 地址和网络端口
5.         s.connect(('192.168.50.1', 8003))
6.     except socket.error as msg:
7.         print(msg)
8.         print(sys.exit(1))
9.
10.    while True:
11.        # filepath 是要被发送图片的路径
12.        filepath = r'D:\AI_project\gluoncv\result.txt'
```



```
13.         fhead = struct.pack(b'128s1', bytes(os.path.basename(filepath),
           encoding='utf-8'), os.stat(filepath).st_size)
14.         s.send(fhead)
15.         print('client filepath: {0}'.format(filepath))
16.
17.         fp = open(filepath, 'rb')
18.         while 1:
19.             data = fp.read(1024)
20.             if not data:
21.                 print('{0} 发送成功...'.format(filepath))
22.                 break
23.             s.send(data)
24.         s.close()
25.         break
```

3.2.4 目标物体的寻找跟随

小车路径策略的代码实现了根据目标的位置和得分进行移动控制的逻辑。首先，代码将输入的参数转换为浮点型以便后续的计算。然后，根据目标的得分进行判断和操作。

如果得分低于 0.01（接近 0），说明找不到目标，需要随机移动直到找到目标。在这种情况下，代码调用 `right_move(t)` 函数执行向右移动的操作，能够缓慢转 360 度在视野中找到目标。

如果得分为-2，表示特殊情况，代码调用 `run_stop_put()` 函数执行相应的操作。

对于其他得分情况，代码会根据目标的边界框位置进行判断和操作。首先，根据边界框的高度判断图像显示是否完整，如果不完整，则需要后退一部分距离。然后，根据边界框的水平位置判断目标是偏左还是偏右。如果目标偏左，代码会计算目标偏离中心点的距离，并根据距离调整移动时间，然后执行向右移动的操作。如果目标偏右，代码会做类似的计算和操作，执行向左移动的操作。如果目标的水平位置在合适的范围内，代码执行其他操作，如停止运动或向前移动。

总体来说，小车路径策略会根据目标的得分和位置，以及预设的条件和移动策略，控制机器人小车的移动行为，使其根据目标的相对位置进行相应的移动操作。

```
1. def move(s, x1, y1, x2, y2, t):
2.     s = float(s)
3.     x1 = float(x1)
4.     y1 = float(y1)
5.     x2 = float(x2)
```

```
6.     y2 = float(y2)
7.     t = float(t)
8.     if s >= -1 and s<0.01: # 得分过低, 找不到目标, 需要随机移动, 直到找到东
    西
9.         right_move(t)
10.    elif s== -2:
11.        run_stop_put()
12.    else:
13.        if y2 - y1 > 250:
14.            if x1 > 640 or x2 > 640 or x1 < 0 or x2 < 0 or y1 > 480 or
    y2 > 480 or y1 < 0 or y2 < 0:
15.                # 图像显示不全, 需要后退
16.                back(t / 2)
17.                print("back")
18.            else:
19.                if (x1 + x2) / 2 > 400: # 目标偏左
20.                    x=((x1+x2)/2)-340
21.                    t=x/680*t
22.                    right_move(t)
23.                    print("close_right_move")
24.                if (x1 + x2) / 2 < 240: # 目标偏右
25.                    x=(340-((x1+x2)/2))
26.                    t=x/680*t
27.                    left_move(t)
28.                    print("far_left_move")
29.                if 240 <= (x1 + x2) / 2 <= 400: # 位置合适, 不动, 接着拍
    摄即可# 找到了目标, 但是距离太远, 需要接近目标
30.                    print("位置合适,ok")
31.                    run_stop_put()
32.            else:
33.                if y1 > 480 or y2 > 480 or y1 < 0 or y2 < 0:
34.                    # 图像显示不全, 需要后退
35.                    back(t / 2)
36.                    print("back")
```

```
37.         else:
38.             if (x1 + x2) / 2 > 400 or x1<20 or x2<20:# 目标偏左
39.                 t=(x2-x1)/680*2*t
40.                 right_move(t)
41.                 print("right_move")
42.             if (x1 + x2) / 2 < 240 or x2>640 or x1>640: # 目标偏右
43.                 t=(x2-x1)/680*2*t
44.                 left_move(t)
45.                 print("left_move")
46.             if 240 <= (x1 + x2) / 2 <= 400:# 位置合适找到了目标,但是
                距离太远, 需要接近目标
47.                 forward(t)
48.                 print("forward")
```

四、系统执行结果与调试说明

4.1 系统执行结果

```
我看到了 banana
client filepath: D:\AI_project\gluoncv\result.txt
D:\AI_project\gluoncv\result.txt 发送成功...
连接开启, 等待传图...
成功连接上 ('192.168.50.1', 41338)
Shape of pre-processed image: (1, 3, 512, 683)
我在找banana
我看到了 banana
client filepath: D:\AI_project\gluoncv\result.txt
D:\AI_project\gluoncv\result.txt 发送成功...
连接开启, 等待传图...
成功连接上 ('192.168.50.1', 41342)
Shape of pre-processed image: (1, 3, 512, 683)
我在找banana
我看到了 banana
client filepath: D:\AI_project\gluoncv\result.txt
D:\AI_project\gluoncv\result.txt 发送成功...
连接开启, 等待传图...
成功连接上 ('192.168.50.1', 41346)
Shape of pre-processed image: (1, 3, 512, 683)
我在找banana
我看到了 banana
client filepath: D:\AI_project\gluoncv\result.txt
D:\AI_project\gluoncv\result.txt 发送成功...
连接开启, 等待传图...
成功连接上 ('192.168.50.1', 41350)
Shape of pre-processed image: (1, 3, 512, 683)
我在找banana
我看到了 banana
client filepath: D:\AI_project\gluoncv\result.txt
D:\AI_project\gluoncv\result.txt 发送成功...
连接开启, 等待传图...
成功连接上 ('192.168.50.1', 41354)
Shape of pre-processed image: (1, 3, 512, 683)
我在找banana
我看到了 banana
client filepath: D:\AI_project\gluoncv\result.txt
D:\AI_project\gluoncv\result.txt 发送成功...
连接开启, 等待传图...
成功连接上 ('192.168.50.1', 41358)
Shape of pre-processed image: (1, 3, 512, 683)
我在找banana
我看到了 banana
client filepath: D:\AI_project\gluoncv\result.txt
D:\AI_project\gluoncv\result.txt 发送成功...
连接开启, 等待传图...
成功连接上 ('192.168.50.1', 41362)
Shape of pre-processed image: (1, 3, 512, 683)
我在找banana
我看到了 banana
client filepath: D:\AI_project\gluoncv\result.txt
D:\AI_project\gluoncv\result.txt 发送成功...
连接开启, 等待传图...
成功连接上 ('192.168.50.1', 41366)
Shape of pre-processed image: (1, 3, 512, 683)
我在找banana
我看到了 banana
```

这是本系统及算你就后端的执行结果, 根据终端输出的“我在找”语句能让我们判断设定的目标寻找物品是否正确, “我看到了”能让我们半段小车所寻找的目标物体是否正确, 同时还有 YOLO 模型对物体识别

后的框框的坐标，来帮助我们规划小车的寻物路线策略。

4.2 调试说明

(1) 图片传输速度慢，计算机的处理速度跟不上

由于视频流的加载问题，图片传输的量很大，计算机对模型的算力跟不上图片传进来的速率。为了解决这个问题，因为视频一秒中有 60 帧，我们只取第 30 帧减少传进来的图片，并且小车每完成一个移动，我们都让小车停止两秒，以留下计算机计算的时间。

(2) 传输来的图片有损耗时，计算机后端会因为异常而停止计算

这个问题困扰了我们很久，而后发现只要将代码放在一个异常处理的语句中抛出异常就可以了，从而让小车连续稳定的行驶前进。

(3) 小车的麦克风设备传输语音和处理语音速度很低，小车的网络连接问题用 ChatGPT-3 网络连接十分卡顿。

为了解决这个问题，我们采用了将语音模块放到计算机的后端实现的方法，将语音识别结果直接发送给图像识别处理模块。

五、总结

5.1 总结与展望

我们小组在这一个星期的时间完成了以下工作：

1. 环境搭建：将学校提供的已组装完成的树莓派小车可以通过手机 APP 进行短距离控制，并且成功与电脑上的树莓派操作系统连接，可以连接上 WIFI 热点进行代码、网页的操作。
2. 功能开发：使用 Python 编程语言完结合树莓派的 GPIO、OpenCV 以及 Chatgpt 等的 API，开发了小车的控制程序和图像识别程序通过计算机网络技术的学习内容将这些功能相连接，使小车能够通过语音输入指令，搜寻捕捉指定物体并进行自主导航避障。
3. 测试和优化：在完成上述程序后，我们对整个系统进行了反复测试和优化，确保其稳定性和可靠性，最终可以基本稳定的实现语音树莓派寻物小车的基本功能。

如果有充裕的时间，或者在后续学期甚至毕业设计时，我们认为可以实现以下比较有创意、有应用前景的实验：

1. 基于深度学习的目标检测与跟踪：通过引入深度学习技术，训练模型来识别和跟踪特定目标，实现小车更加智能化的自主导航，可以设计更加复杂的路线，并添加相关的寻路算法以使其应用于更加广泛的场合。

2. 基于语音识别的自然语言对话系统：通过引入自然语言处理技术，实现小车与人的自然语言对话，使得人们可以更加方便地控制小车。

3.基于机器学习的行为预测和规划：通过引入机器学习技术，对小车的行为进行预测和规划，使得小车能够更加灵活地适应不同的环境和任务。

4.基于树莓派各个机器组件的组装和实现：通过将树莓派小车和机械臂或坦克等其他组件的结合、合作，实现功能更完整、实用性更强的智能电子机械产品，比如实现物体的寻找和抓取找回。

5.基于摄像头人脸识别功能的交互：通过引入摄像头的人脸识别技术，实现树莓派智能小车的人脸识别登录和操控，增加系统的安全性。

总之，基于树莓派的小车系统具有很大的可拓展性和应用前景，未来可以通过不断的技术创新和优化，实现更加智能化、自主化的功能。

5.2 感想与体会

通过本次实验，我们组学习到了硬件的组装、硬软件环境的配置、软件控制硬件的方法、各种算法的设计与调试、语音输入识别与输出的方式、以及调用 API 接口的方法。相较于其他组，本组的实验有较多的创新点，集中体现在能通过模型识别物品进行寻路、语音输入与识别、实时监测过障、路径选择以及 AI 智能与语音的回应，并且保证结果的相对正确性。在调试的过程中，我们曾多次被一些违反与直觉与我们所写函数的现象所迷惑与击倒，但在队友们的共同协作之下，最终一步一步慢慢摸索，找到了产生其问题的根本原因，大家通力合作，解决问题，是一次完美的团队协作经历，让我们体会到了团队合作的重要性。通过这个项目，我们懂得了实际工程中调参的不易和对工程严密性与准确性的把握，这些东西让我们深深体会到了作为一个软件开发人员、研究者应该具备的基本素质，那就是严谨、细致、准确、认真等等。

队长黄启明的总结是，通过本次工程实践课程，作为组长，我深刻地领悟到了工程实践的重要性和价值。在整个项目的开发过程中，我们组学习到了硬件组装、环境配置、软件控制硬件、算法设计与调试、语音识别与输出等多个方面的知识和技能。与其他组相比，我们组的实验具有较多的创新点，尤其是在物品识别、语音输入与识别、实时监测过障、路径选择和智能回应等方面的应用上。在调试的过程中，我们遇到了许多令人困惑和难以解决的问题，有时与我们所编写的函数行为不符，让我们感到迷惑和挫败。但是，在团队成员的共同协作和密切配合下，我们逐步摸索和解决了这些问题。这次团队协作经历让我深刻认识到了团队合作的重要性和价值，只有通过团队的共同努力和智慧，才能克服困难，取得良好的成果。通过这个项目，我们不仅学会了调试的艺术，还意识到了工程实践中参数调优的复杂性以及对工程的严谨性和准确性的要求。我们深刻体会到作为软件开发人员和研究者，我们需要具备严谨、细致、准确和认真的工作态度和基本素质。在项目开发过程中，我们养成了仔细分析问题、精确思考和严谨实施的习惯，这对我们未来的学习和工作具有重要的指导意义。这次课程不仅加深了对计算机工程实践的理解，还培养了我们解决问题和创新的能力。在未来的学习和工作中，我将继续努力，不断提升自己的实践能力和团队合作能力，为实际应用中的工程问题做出更大

的贡献。

队员邹莲麦子的总结是，这个工程实践项目让我深刻体会到了智能技术在现实生活中的应用潜力。通过整个开发过程，我学习到了语音识别、图像处理、目标检测和自主导航等方面的知识和技能。同时，我也深入了解了团队合作和项目管理的重要性，因为这个项目需要团队成员之间的密切协作和有效沟通。通过这次工程实践，我不仅提升了自己的技术能力，还培养了解决问题和创新的能力。我深感工程实践课程的重要性，它为我们提供了一个实践应用所学知识的平台，让我们能够将理论知识转化为具体的实际项目。这种实践性的学习方式不仅增强了对知识的理解和掌握，还培养了我们的团队合作能力和解决问题的能力。因此，通过参与这个智能小车系统的开发，我不仅获得了宝贵的知识和技能，还体验到了工程实践的乐趣和成就感。我相信这个项目的成功也将为我的项目成果经历添上浓墨重彩的一笔。工程实践课程让我深刻认识到理论与实践的紧密联系，为我今后的学习和工作奠定了坚实的基础。

队员章张毅涛的总结是，这次的工程实践对我来说是一个很大的挑战，因为我对硬件的了解十分的薄弱，在进行这次的时间之前一直对硬件抱有一种隐隐的担忧，但是通过这次实验加深了我对硬件的理解，让我明白外设的控制主要是通过对于各个部件的高低电平来控制的。在这五天里，心态只能说一波三折，一开始看之前几年学长学姐们的成果，让我在惊叹他们完成度之高的同时也让我怀疑自己是否能向他们一样做到这么多功能。在实践的第一天，我是十分迷茫的，对于自己应该做什么，需要做什么，乃至完成小组任务需求需要学习什么都不清楚；到实践第二天，通过第一天的积累，我逐渐了解了小车的构建、运行方式，能与队友进行调试控制，也能自己修改代码，也经历了配置环境的折磨；再到第三天我们小组成果的突飞猛进，逐渐解决链接，识别，控制这些难关的惊喜；也有在第四第五天，为了增加更多功能，为了保证系统功能准确性、稳定性而一次又一次做出的努力。回想过去的五天，其实我们也走了很多弯路，比如下载 cv2 组件的时候，下了快半小时才发现原来系统中已经装载了，代码改了半天后来又改回原版的事情更是数不胜数，但是正是这些弯路，才让我们更好地理解其中的含义，让我们能更深刻地理解其中的意义。在这五天时间里，我们四个人一直都待在一起，一起经历完成过程中的痛苦，也一起享受了完成后的快乐，这其实比单纯为了卷功能而卷功能更加有意义，同时我们享受也大家提出一个个新点子后，去将它落地时的那种自豪。当然也很感谢助教对我的帮助，给我们小组很多有用的点子，也一直在鼓励着我们！这次工程实践受益匪浅！

队员章张栩宁的总结是，本次工程实践是我的第一次代码程序实践，对我来说是一次难忘且重要的体验。对我自身来说，我的动手能力和编码能力都比较薄弱，需要通过实践项目等不断积累经验来加强我的编码能力，以便在以后的生活中能更好的工作。在此次工程实践的过程中，我了解到了各种智能小车、智能机器人的基本工作原理，以及它们内部模块如何相连的。除此之外，学会如何正确的借鉴并修改代码是我学到的最关键的内容。我深刻的体会到了人工智能的强大，Chatgpt、图灵机器人等 AI 在提供思路、代码编写和修改和功能测试等方面帮助了我特别多，以至于我觉得等人工智能在发展几年后或许会取代一大部分程序员的工作（危机感逐渐上升 XD）。这五天里，通过与组员的沟通和合作，我完成我之前从未想过的东西，也收获了无数的欢乐（虽然花钱买测试物品的时候还是挺

悲伤的，，，），希望这次工程实践的经历帮助我的不止于此，一起加油,一起进步！

Enlightening Wheat 流光麦田

附录 程序清单

语音模块

```
1. # -*- coding: utf-8 -*-
2. import openai
3. import pygame
4. import os
5. import speech_recognition as sr
6. from gtts import gTTS
7. import time
8.
9. # Replace with your GPT-3 API key
10. openai.api_key = "sk-fhojNEzkRWFs80lRpMicT3B1bkFJYNGh8GeJqGTHwkMjnJcZ"
11.
12. def generate_gpt3_response(prompt):
13.     response = openai.Completion.create(
14.         engine="text-davinci-002",
15.         prompt=prompt,
16.         max_tokens=100,
17.         n=1,
18.         stop=None,
19.         temperature=0.5,
20.     )
21.
22.     return response.choices[0].text.strip()#GPT-3 生成的第一个响应文本的字符串返回，并去除开头和结尾的空白字符。
23.
24. def speak(text, language="zh-cn"):
25.     tts = gTTS(text=text, lang=language)
26.     tts.save("response.mp3")
27.
28.     pygame.mixer.init()#初始化 pygame.mixer 模块
29.     pygame.mixer.music.load("response.mp3")#方法加载 response.mp3 音频文件
```

```
30.     pygame.mixer.music.play()#播放该音频文件
31.
32.     while pygame.mixer.music.get_busy():
33.         pygame.time.Clock().tick(10)#当音频正在播放时,
        pygame.mixer.music.get_busy() 方法将返回 True, while 循环会不断调用 tick() 方法等待音频播放完毕。
34.
35.     # 等待 3 秒, 确保文件不再被占用
36.     time.sleep(3)
37.
38.     os.remove("response.mp3")
39.
40. def recognize_speech(retries=3):
41.     recognizer = sr.Recognizer()
42.     for _ in range(retries):#默认尝试3次麦克风识别
43.         with sr.Microphone() as source:#打开麦克风
44.             print("请说话...")
45.             audio = recognizer.listen(source)#Recognizer 对象监听来自麦克风的语音输入, 并将其存储在 audio 变量中以备后续的语音识别
46.
47.             try:
48.                 text = recognizer.recognize_google(audio, language="zh-CN")
49.                 #将存储在 audio 变量中的语音转换为文本
                    print(f"你说的是: {text}")
50.                 return text
51.             except sr.UnknownValueError:
52.                 print("Google Speech Recognition 无法识别语音")
53.             except sr.RequestError as e:
54.                 print(f"无法从 Google 语音识别服务请求结果; {e}")
55.
56.     return None
57.
58. def aispeak():
59.     while True:
```

```
60.         spoken_command = recognize_speech()
61.
62.         if spoken_command is not None:
63.             # Get a keyword from the spoken command and translate it to
             English
64.             keyword_prompt = f"从 \"{spoken_command}\" 中提取一个关键词并
             翻译成英文"
65.             keyword = generate_gpt3_response(keyword_prompt)
66.             print(f"Keyword: {keyword}")
67.
68.             if keyword.lower() == "joke":
69.                 joke_prompt = f"请用中文给我讲一个笑话"
70.                 joke = generate_gpt3_response(joke_prompt)
71.                 speak(joke)
72.             elif keyword.lower() == "song":
73.                 if "猜猜" in spoken_command:
74.                     guess_prompt = f"猜猜我最喜欢的歌曲"
75.                     guess = generate_gpt3_response(guess_prompt)
76.                     speak(guess)
77.                 else:
78.                     song_prompt = f"请告诉我今年的十大热门歌曲"
79.                     top_song = generate_gpt3_response(song_prompt)
80.                     speak(top_song)
81.             elif "小车" in spoken_command:
82.                 response_prompt = f"请根据 \"{spoken_command}\" 生成一个
                 合适的回复"
83.                 response = generate_gpt3_response(response_prompt)
84.                 speak(response)
85.
86.             else:
87.                 if "相关" in spoken_command:
88.                     content_prompt = f"请根据关键词 \"{keyword}\" 生成一段
                     有关的内容"
89.                     content = generate_gpt3_response(content_prompt)
```

```
90.             speak(content)
91.         else:
92.             modified_response = f"我在找{keyword}快让我看看 "
93.             speak(modified_response)
94.             return keyword
95.         # 暂停 5 秒
96.         time.sleep(5)
97.
98. if __name__ == "__main__":
99.     result=aispeak()
100.    print(f"Returned: {result}")
```

图像识别模块

图像传输部分:

```
1. import socket
2. import os
3. import sys
4. import struct
5.
6. import scanimage
7.
8.
9. def socket_service():
10.     try:
11.         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12.         s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
13.         # IP 地址留空默认是本机 IP 地址
14.         s.bind(('', 8088))
15.         s.listen(7)
16.     except socket.error as msg:
17.         print(msg)
18.         sys.exit(1)
19.
20.    print("连接开启，等待传图...")
```



```
21.
22.     while True:
23.         sock, addr = s.accept()
24.         deal_data(sock, addr)
25.
26.     s.close()
27.
28.
29. def deal_data(sock, addr):
30.     print("成功连接上 {}".format(addr))
31.
32.     while True:
33.         fileinfo_size = struct.calcsize('128sl')
34.         buf = sock.recv(fileinfo_size)
35.         if buf:
36.             filename, filesize = struct.unpack('128sl', buf)
37.             fn = filename.decode().strip('\x00')
38.             # PC 端图片保存路径
39.             new_filename = os.path.join('D:\\AI_project\\gluoncv', fn)
40.
41.             recvd_size = 0
42.             fp = open(new_filename, 'wb')
43.
44.             while not recvd_size == filesize:
45.                 if filesize - recvd_size > 1024:
46.                     data = sock.recv(1024)
47.                     recvd_size += len(data)
48.                 else:
49.                     data = sock.recv(1024)
50.                     recvd_size = filesize
51.                     fp.write(data)
52.             fp.close()
53.             sock.close()
```

```
54.         class_id, scores, bounding_boxs = scanimage.scan(r'D:\AI_project\gluoncv\image.png')
55.         # # print('class_id:', class_id)
56.         # # print('scores:', scores)
57.         # # print('bounding_boxs:', bounding_boxs)
58.         # # os.remove(r'D:\AI_project\gluoncv\image.png')
59.         # file = open(r"D:\AI_project\gluoncv\result.txt", 'w')
60.         # file.write(str(class_id))
61.         # file.write('\n')
62.         # file.write(str(scores))
63.         # file.write('\n')
64.         # file.write(str(bounding_boxs))
65.         # result.send_txt()
66.         break
67.
```

图像识别部分:

```
1. from gluoncv import model_zoo, data, utils
2. from matplotlib import pyplot as plt
3. from getimage import *
4. import cv2
5. import mxnet as mx
6. from mxnet import nd
7. import os
8.
9. target="banana"
10.
11. def selectID(class_IDs):
12.     b=0
13.     a={}
14.     while True:
15.
16.         ID=int(class_IDs[0][b].asscalar())
17.         if (ID)>=0:
18.             a[b]=ID
```

```
19.         elif ID==-1:
20.             break
21.         b+=1
22.     return a
23.
24.
25. def scan(dir):
26.     net = model_zoo.get_model('yolo3_darknet53_coco', pretrained=True,
        root='D:\AI_project\gluoncv\model')
27.     try:
28.         x, img = data.transforms.presets.yolo.load_test(dir, short=512)
29.         print('Shape of pre-processed image:', x.shape)
30.         class_IDs, scores, bounding_boxes = net(x)
31.         # print('class:', class_IDs[0][0])
32.         # print('scores:', scores[0][0])
33.         # print('bounding_box:', bounding_boxes[0][0])
34.         with open("D:/Desktop/animal_follower/PC/target.txt", "r", encoding="utf-8") as f:
35.             target=f.read()
36.             print("我在找"+target)
37.             targetIDs=selectID(class_IDs)
38.             for key in targetIDs:
39.                 if net.classes[targetIDs[key]]==target:
40.                     print("我看到了",target)
41.                     return class_IDs[0][key],scores[0][key],bounding_boxes[0][key]
42.             # plt.show()
43.             print("没有找到目标")
44.             return class_IDs[0][50],scores[0][50],bounding_boxes[0][50]
45.     except:
46.         print("刚刚那张图片损坏了")
47.     return 0
48.
49.
```

```
50. if __name__ == '__main__':  
51.     scan(r'C:\Users\13079\Desktop\demo.png')
```

通信模块

小车将图片传给 PC:

```
40. import socket  
41. import os  
42. import sys  
43. import struct  
44.  
45. def upload():  
46.     try:  
47.         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
48.         #192.168.199.1 和 8088 分别为服务端 (pc) 的 IP 地址和网络端口  
49.         s.connect(('192.168.50.107', 8086))  
50.     except socket.error as msg:  
51.         print(msg)  
52.         print(sys.exit(1))  
53.  
54.     video = cv2.VideoCapture("http://192.168.50.1:8080/?action=stream")  
55.     cv2.namedWindow("img", cv2.WINDOW_NORMAL)  
56.     ret, frame = video.read()  
57.     time.sleep(2)  
58.     if frame is None:  
59.         print("2")  
60.     cv2.imwrite("/home/pi/Desktop/gluoncv/image.png", frame)  
61.  
62.     while True:  
63.         #filepath 是要被发送图片的路径  
64.         filepath = '/home/pi/Desktop/gluoncv/image.png'  
65.         fhead = struct.pack(b'128s1', bytes(os.path.basename(filepath),  
            encoding='utf-8'), os.stat(filepath).st_size)  
66.         s.send(fhead)  
67.         print('client filepath: {0}'.format(filepath))  
68.
```

```
69.         fp = open(filepath, 'rb')
70.         while 1:
71.             data = fp.read(1024)
72.             if not data:
73.                 print('{0} 发送成功...'.format(filepath))
74.                 break
75.             s.send(data)
76.         s.close()
77.         break
78. returnn 0
```

PC 将有关识别结果的 txt 文件传给小车:

```
26. def send_txt():
27.     try:
28.         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
29.         # 192.168.199.1 和 8088 分别为服务端 (pc) 的 IP 地址和网络端口
30.         s.connect(('192.168.50.1', 8003))
31.     except socket.error as msg:
32.         print(msg)
33.         print(sys.exit(1))
34.
35.     while True:
36.         # filepath 是要被发送图片的路径
37.         filepath = r'D:\AI_project\gluoncv\result.txt'
38.         fhead = struct.pack('b128s1', bytes(os.path.basename(filepath),
39. encoding='utf-8'), os.stat(filepath).st_size)
40.         s.send(fhead)
41.         print('client filepath: {0}'.format(filepath))
42.
43.         fp = open(filepath, 'rb')
44.         while 1:
45.             data = fp.read(1024)
46.             if not data:
47.                 print('{0} 发送成功...'.format(filepath))
```

```
47.             break
48.             s.send(data)
49.             s.close()
50.             break
```

路线策略模块

```
1. import RPi.GPIO as GPIO          # 引入 GPIO 模块
2. import time                       # 引入 time 模块
3.
4. IN1 = 20
5. IN2 = 21
6. IN3 = 19
7. IN4 = 26
8. ENA = 16
9. ENB = 13
10. speed=10
11. speed2=30
12. # 设置 GPIO 口为 BCM 编码方式
13. GPIO.setmode(GPIO.BCM)
14. # 忽略警告信息
15. # GPIO.setwarnings(False)
16.
17. # GPIO.setup(ENA,GPIO.OUT,initial=GPIO.HIGH)
18. # GPIO.setup(IN1,GPIO.OUT,initial=GPIO.LOW)
19. # GPIO.setup(IN2,GPIO.OUT,initial=GPIO.LOW)
20. # GPIO.setup(ENB,GPIO.OUT,initial=GPIO.HIGH)
21. # GPIO.setup(IN3,GPIO.OUT,initial=GPIO.LOW)
22. # GPIO.setup(IN4,GPIO.OUT,initial=GPIO.LOW)
23. # 设置 pwm 引脚和频率为 2000hz
24. # pwm_ENA = GPIO.PWM(ENA, 2000)
25. # pwm_ENB = GPIO.PWM(ENB, 2000)
26.
27. # pwm_ENA.stop()
28. # pwm_ENB.stop()
29.
```

```
30.# pwm_ENA.start(0)
31.# pwm_ENB.start(0)
32.
33.def forward(t):
34.    run_stop_get()
35.    GPIO.output(IN1, GPIO.HIGH)
36.    GPIO.output(IN2, GPIO.LOW)
37.    GPIO.output(IN3, GPIO.HIGH)
38.    GPIO.output(IN4, GPIO.LOW)
39.    pwm_ENA.ChangeDutyCycle(speed)
40.    pwm_ENB.ChangeDutyCycle(speed)
41.    time.sleep(t)
42.    run_stop_put()
43.
44.
45.def back(t):
46.    run_stop_get()
47.    GPIO.setmode(GPIO.BCM)
48.    GPIO.output(IN1, GPIO.LOW)
49.    GPIO.output(IN2, GPIO.HIGH)
50.    GPIO.output(IN3, GPIO.LOW)
51.    GPIO.output(IN4, GPIO.HIGH)
52.    pwm_ENA.ChangeDutyCycle(speed)
53.    pwm_ENB.ChangeDutyCycle(speed)
54.    time.sleep(t/2)
55.    run_stop_put()
56.
57.def right_move(t):
58.    run_stop_get()
59.    GPIO.output(IN1, GPIO.HIGH)
60.    GPIO.output(IN2, GPIO.LOW)
61.    GPIO.output(IN3, GPIO.LOW)
62.    GPIO.output(IN4, GPIO.LOW)
63.    pwm_ENA.ChangeDutyCycle(speed)
```



```
64.     pwm_ENB.ChangeDutyCycle(0)
65.     time.sleep(t/2)
66.     run_stop_put()
67.
68.
69. def left_move(t):
70.     run_stop_get()
71.     GPIO.output(IN1, GPIO.LOW)
72.     GPIO.output(IN2, GPIO.LOW)
73.     GPIO.output(IN3, GPIO.HIGH)
74.     GPIO.output(IN4, GPIO.LOW)
75.     pwm_ENA.ChangeDutyCycle(0)
76.     pwm_ENB.ChangeDutyCycle(speed)
77.     time.sleep(t/2)
78.     run_stop_put()
79.
80. def right_turn(t):
81.     run_stop_get()
82.     GPIO.output(IN1, GPIO.HIGH)
83.     GPIO.output(IN2, GPIO.LOW)
84.     GPIO.output(IN3, GPIO.LOW)
85.     GPIO.output(IN4, GPIO.HIGH)
86.     pwm_ENA.ChangeDutyCycle(speed2)
87.     pwm_ENB.ChangeDutyCycle(speed2)
88.     time.sleep(t/2)
89.     run_stop_put()
90.
91. def left_turn(t):
92.     run_stop_get()
93.     GPIO.output(IN1, GPIO.LOW)
94.     GPIO.output(IN2, GPIO.LOW)
95.     GPIO.output(IN3, GPIO.HIGH)
96.     GPIO.output(IN4, GPIO.LOW)
97.     pwm_ENA.ChangeDutyCycle(speed2)
```

```
98.     pwm_ENB.ChangeDutyCycle(speed2)
99.     time.sleep(t/2)
100.     run_stop_put()
```

```
1. def move(s, x1, y1, x2, y2, t):
2.     s = float(s)
3.     x1 = float(x1)
4.     y1 = float(y1)
5.     x2 = float(x2)
6.     y2 = float(y2)
7.     t = float(t)
8.     if s >= -1 and s < 0.01: # 得分过低, 找不到目标, 需要随机移动, 直到找到东
    西
9.         right_move(t)
10.    elif s == -2:
11.        run_stop_put()
12.    else:
13.        if y2 - y1 > 250:
14.            if x1 > 640 or x2 > 640 or x1 < 0 or x2 < 0 or y1 > 480 or
            y2 > 480 or y1 < 0 or y2 < 0:
15.                # 图像显示不全, 需要后退
16.                back(t / 2)
17.                print("back")
18.            else:
19.                if (x1 + x2) / 2 > 400: # 目标偏左
20.                    x = (((x1+x2)/2) - 340)
21.                    t = x/680*t
22.                    right_move(t)
23.                    print("close_right_move")
24.                if (x1 + x2) / 2 < 240: # 目标偏右
25.                    x = (340 - ((x1+x2)/2))
26.                    t = x/680*t
27.                    left_move(t)
28.                    print("far_left_move")
```

```
29.         if 240 <= (x1 + x2) / 2 <= 400: # 位置合适, 不动, 接着拍
           摄即可# 找到了目标, 但是距离太远, 需要接近目标
30.             print("位置合适,ok")
31.             run_stop_put()
32.     else:
33.         if y1 > 480 or y2 > 480 or y1 < 0 or y2 < 0:
34.             # 图像显示不全, 需要后退
35.             back(t / 2)
36.             print("back")
37.     else:
38.         if (x1 + x2) / 2 > 400 or x1<20 or x2<20:# 目标偏左
39.             t=(x2-x1)/680*2*t
40.             right_move(t)
41.             print("right_move")
42.         if (x1 + x2) / 2 < 240 or x2>640 or x1>640: # 目标偏右
43.             t=(x2-x1)/680*2*t
44.             left_move(t)
45.             print("left_move")
46.         if 240 <= (x1 + x2) / 2 <= 400:# 位置合适找到了目标, 但是
           距离太远, 需要接近目标
47.             forward(t)
48.             print("forward")
```

报告格式要求

全文格式要求:

- (1) 汉字用宋体, 11 号。所有的数字和英文字符用 Times New Roman 字体。
- (2) 每段开头严格空 2 个汉字宽度。
- (3) 不能随意空行, 一段没有写完不要回车。
- (4) 一页没有写完时也不要换到下一页, 即: 不要出现半页空白的现象。如果由于图片太大使本页装不下而跳到下一页, 从而出现前一页出现半页空白的情况, 请将图后的文字前移以填充

前页空白。

图的要求如下：

- (1) 图应有图注，图和图注均应左右居中（在标尺拉到最左侧的情况下），且图和图注不能跨页显示，即：它们应在同一页。
- (2) 图注文字为 10 号字体。图和图注的前后段落间距均为 0.25 行。
- (3) 除非是照片式或者彩色复杂图，否则尽量不要截图，要使用 Visio 软件自己去绘制。截图一定要清晰不能糊，否则不要截。对于一些线条图或者框图，强烈建议使用 Visio 软件绘制。
- (4) 使用 visio 绘制图时，建议文字使用 9 号字体（中文为宋体，数字和英文为 Times New Roman）。文字和方框比例要合适
- (5) 全文中所有图的比例要大小统一，以图中文字比正文文字略小为宜。不可过大或过小，不能超出页边界。
- (6) 出现图之前，先用文字引出。例：××××如图×所示。即：先有图的文字介绍，再有具体的图。
- (7) 全文中图的编号为：图 1，图 2，……，图 n。

图的样例如图 1 所示。

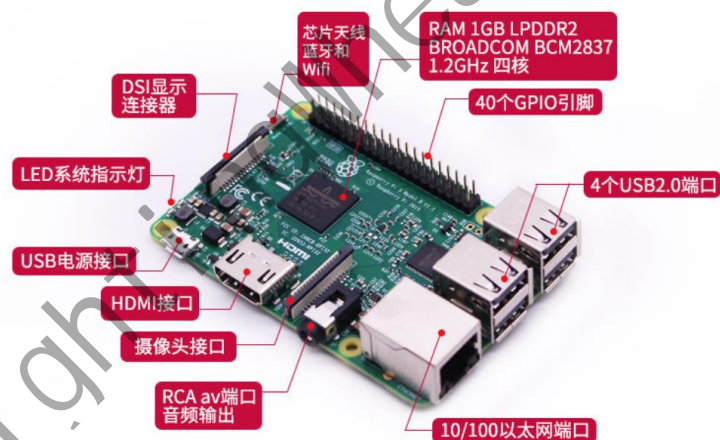


图 1 树莓派 3B+ 主控板

表的要求如下：

- (1) 表要有表头，表头在表上面，而不是下面。表头和表不要跨页显示。表和表头均应左右居中（在标尺拉到最左侧的情况下）。
- (2) 表的编号为：表 1，表 2，……，表 n。
- (3) 表中文字上下、左右居中。表中文字为 10 号字体，中文宋体，英文和数字 Times New Roman。
- (4) 单行表高度为 0.7cm；2 行 1.2cm，3 行 2cm。表格大小要与文字多少匹配，不要出现表格太松散或者太拥挤的情况。
- (5) 表中单行文字时行距为“单倍行距”；多行文字时，文字行距为 1.15。

- (6) 表中的标题行（第一行）底纹为灰色，字体为：汉字黑体，非汉字为 Airal。
- (7) 出现表的地方先用文字引出，例如：xxx如表x所示。即：先有表的文字介绍，再有具体的表。
- (8) 表格之后的下一段文字，段前间距为 0.5 行，以使得其文字不要与表格挨得太近。
- 表的例子如表 1 所示。

表 1 本阶段完成的 3 个教学作品集

Order	Topic	主题
Section A	Background	教育背景与教学实践基本情况
Section B	Understanding of Student-centered Learning and Setting up the Goal to Change	理解以学生为中心的学习并设定改变的目标
Section C	Reflection on the Pilot-project	Pilot 项目的反思

此处的文字段前间距为 0.5 行，段后为 0 行。