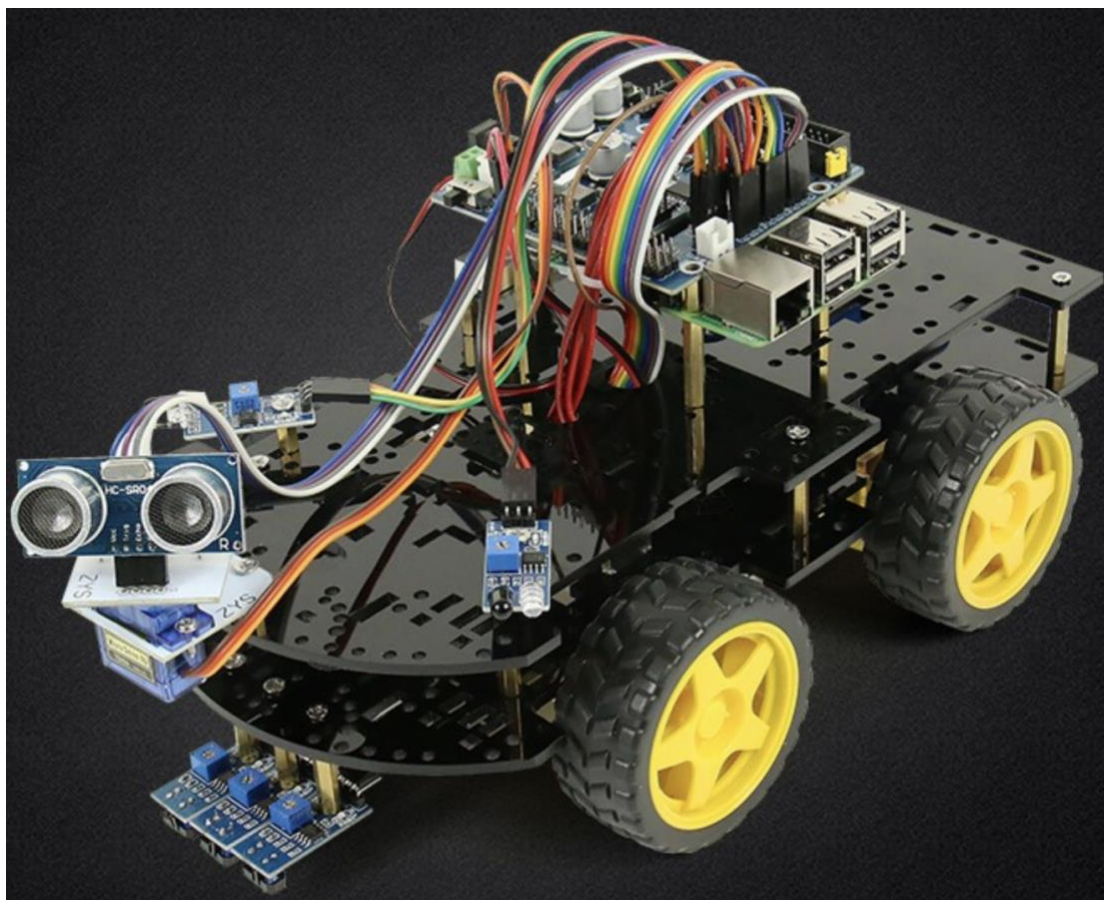


作业文档



目录

1. 作业目标	3
2. 作业器材	3
2.1 车模构成简述	3
2.2 主要部件介绍	3
2.2.1 控制核心	3
2.2.2 传感器	4
3. 作业任务	4
3.1 作业任务简述	4
3.2 考察内容及评分方案	5
3.2.1 情形一：返校	5
3.2.2 情形二：未返校	6
3.3 作业比赛赛道	8
4. 开发环境	8
5. 远程调试说明（针对未返校情况）	11
6. 附件说明	12
7. 附录	12

1. 作业目标

我们希望同学们通过完成本次课程大作业，了解和学习到以下内容：

1. 了解嵌入式编程的基本流程与方法。
2. 了解本次作业用到的各类传感器的基本构造、工作原理以及使用方法。
3. 了解树莓派平台的嵌入式开发，通过完成作业任务要求，对嵌入式开发有更加清晰直观的认识。

2. 作业器材

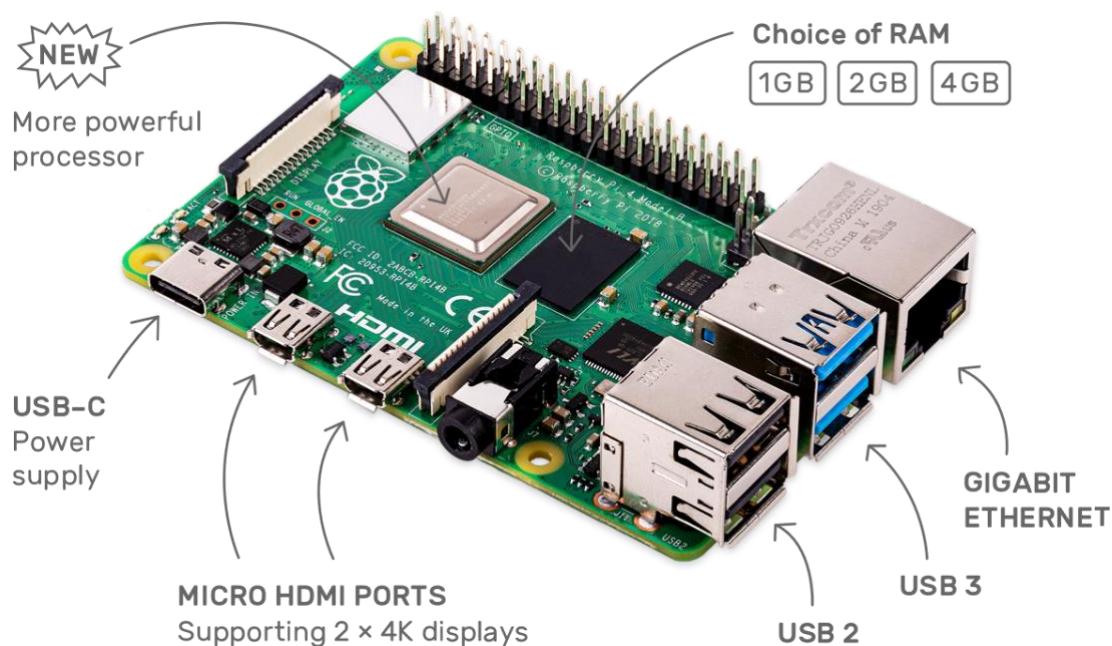
2.1 车模构成简述

本次大作业使用的车模为树莓派智能小车（组装完成后效果见封面图），主要部件包括了：树莓派 4B 主板，红外避障传感器，红外循迹传感器，超声波传感器等。车身上部为红外循迹传感器，车身上部为超声波传感器，车身两侧为红外避障传感器。车模支持自动驾驶、智能避障、自动循迹等功能的实现。

2.2 主要部件介绍

2.2.1 控制核心

本次使用的车模的控制核心为一个[树莓派 4 Model B](#)



本次使用的树莓派的详细配置参数见官方说明：

<https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>

2.2.2 传感器

本次作业提供的传感器包括了两个红外避障传感器，三个红外循迹传感器，一个超声波传感器。各传感器的详细说明，我们将在「4. 开发环境」中进行。

3. 作业任务

3.1 作业任务简述

本次大作业的最终任务是使用给定的车模实现巡线、避障功能，并进行组间巡线避障竞速比赛。在实现最终任务的过程中，同学们需要实现的功能包括：控制车模的运动，利用车模上的传感器识别赛道以及障碍物等；并据此设计巡线、避障算法以完成最终的竞赛。考虑到疫情带来的不确定性，我们针对返校和未返校两种情况分别规定了不同的考察内容，详见下节说明。

3.2 考察内容及评分方案

因为疫情带来的不确定性，考察内容和评分方法分两种情况，需要特别说明的是若发现抄袭情况（包括代码以及文档），则大作业记 0 分。

3.2.1 情形一：返校

1. 对各个功能模块的完成度进行考察（50 分）

主要考察的功能模块有：车模的前进、后退、左转、右转、停止等。具体来说，

我们要求同学们的实现的车模**自动依次**完成以下流程：

- a. 车模**前进** 1m 后**停止**（允许误差 $\pm 20\text{cm}$ ，超出记 0 分）；（10 分）
- b. 车模**后退** 1m 后**停止**在出发线后起始位置（允许误差 $\pm 20\text{cm}$ ，超出记 0 分）；（10 分）
- c. 车模重新前进，并同时**左转**，转弯完成后**停止**，要求转弯后车身位置与初始车身位置夹角大于等于 90 度同时小于等于 180 度，否则记 0 分；（10 分）
- d. 以上次停车位置为起始位置，按 c 相同要求进行**右转**，然后**停止**；（10 分）

附加要求：每次停止时间应该大于 1s，但不超过 3s。每次停止符合要求得 2.5 分，总分 10 分。单步完成时间没有限制，但要求总完成时间在 60s 内，否则每超时 1s 扣 1 分。

2. 车模进行自动巡线避障竞速比赛（40 分）

- a. 同学们使用制作的车模完成规定按赛道运行 **3 周**。（赛道说明见 3.3）
- b. 比赛时间记录从车模冲过起跑线到跑完 3 周后重新回到起跑线为止的时间，要求车模在完成比赛后准确停在起跑线后 2m 的直线赛道之上。
 - i. 如果车模没有能够停止在起跑线后 2m 的赛道停车区内，时间加罚 10 秒

钟。

- ii. 未跑完 3 圈或超出规定赛道范围撞击赛道外纸板视作未完成。
- c. 每组车模有两次机会，取两次中用时最短的一次作为最终时间，对各组按最终时间进行排名，并按名次给分：
 - i. 两次均未完成赛道任务，得 0 分
 - ii. 如完成赛道任务，按用时排名，各名次得分对应关系见下：
 - 1. 第 1 名：40 分
 - 2. 第 2 名：38 分
 - 3. 第 3 名：36 分
 - 4. 第 4 名：34 分
 - 5. 第 5 名：32 分
 - 6. 第 6-9 名：30 分
 - 7. 第 10-13 名：28 分
 - 8. 第 14-17 名：26 分
 - 9. 剩余完成组：24 分

3. 作业文档（10 分）

- a. 说明文档要求对作业任务中规定的功能完成情况以及使用的算法进行说明
- b. 同时要求撰写代码说明文档，对代码结构进行简要说明
- c. 评分综合考虑文档的完整度和表述的正确性

3.2.2 情形二：未返校

1. 对各个功能模块的完成度进行考察（90 分）

基本功能考察与返校情况相同，同时我们将额外增加新的考察内容。

a. 直线竞速 (20 分) :

车模完成 10m 的直线赛道，记录车头从越过起跑线到越过终点线的时间（要求车模整体停在终点线的 2m 范围内，超出将罚时 3s），每组有一次机会，按时间排名，依据名次给分：

- 第 1 名 : 20 分
- 第 2 名 : 19 分
- 第 3 名 : 18 分
- 第 4-6 名 : 17 分
- 第 7-9 名 : 16 分
- 第 10-12 名 : 15 分
- 第 13-15 名 : 14 分
- 第 16-18 名 : 13 分
- 剩余组 : 12 分

b. 圆周避障 (20 分) 。

要求车模从给定起始点出发，按半径为 1m 的圆周逆时针向前运动，在起始位置正前方 1m ~ 2m 的未知位置处设有一个障碍物（例如，墙壁），要求车模在行进中检测障碍物，并在车头行进到障碍物前 10cm 的停车区时进行一次停车（停止时间在[1, 3]s 内）。在成功停车的前提下，需要将车模以任意路线开回起始点，最终按照车模中心点与起始点的距离评分。示例参照附件 8。距离与得分关系如下，距离单位 cm：

- [0, 5) : 20 分
- [5, 10) : 18 分

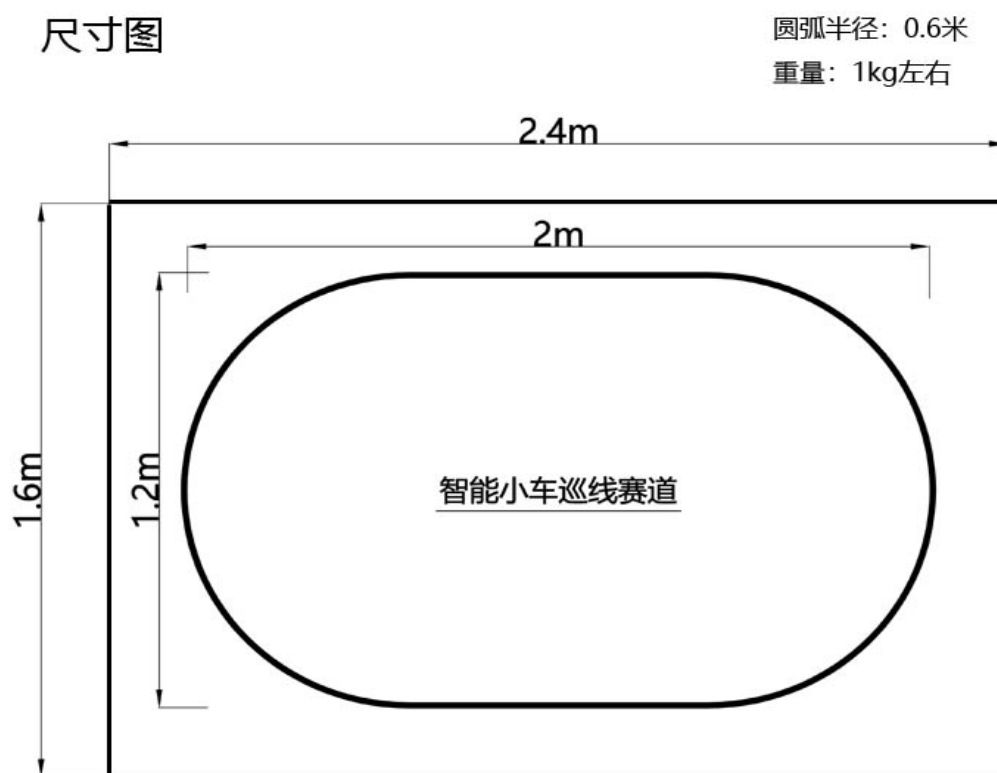
- [10, 15) : 16 分
- 依次类推，最低得分 0 分

2. 作业文档 (10 分)

- a. 要求同情形一。

3.3 作业比赛赛道

实际赛道周围会使用纸板进行封闭以同时考查车模避障能力。



4. 开发环境

由于目前客观条件的限制，在本次大作业中我们略去了硬件搭建部分，车模组建后的成型图见「作业器材」一节。作业所需的传感器模块已经通过树莓派 GPIO 与树莓派

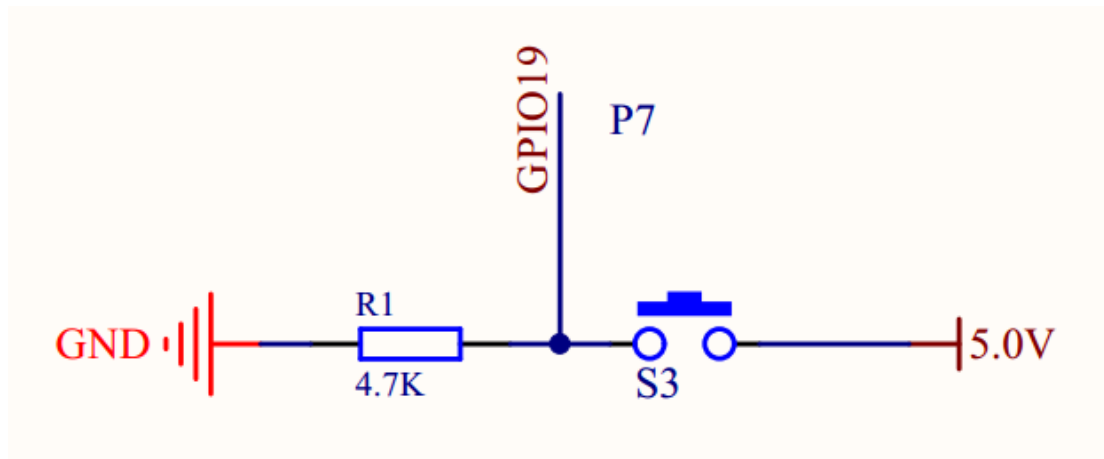
连接，在本次大作业中，我们主要用到了以下四种模块：

- **电机驱动**：TB6612FNG 是东芝半导体公司生产的一款直流电机驱动器件，同学们可以通过附件 2 的数据手册或者自己查找资料来了解这款驱动的工作原理。附件 1 中已经给出了电机驱动与树莓派的接线原理图，目前我们仅使用前轮的驱动。（感兴趣的同学可以尝试如何使用后轮驱动）
- **红外循迹传感器、红外避障传感器**：小车上共有 3 个循迹传感器，位于小车的前端底部，在附件 1 中用 J4（左 1），J2（左 2），J3（左 3）表示。共有两个循迹传感器，分别位于小车前端的左右两侧，在附件 1 中用 J5（右），J6（左）表示。这两种传感器原理类似，都是利用不同颜色对红外光的反射效果不同的原理，两种不同的电路设计（见附件 3，附件 4）是为了适应不同的检测距离。（大家可以分别采集这两组数据，看看有什么不同）。
- **超声波传感器**：小车使用的超声波型号为 HC-SR04，位于小车正前方，在附件 1 中用 CSB1 表示，电路原理图和数据手册见附件 5，附件 6。

本次大作业用到的所有外设都通过 GPIO 进行输入输出数据，传感器的调用流程可以简化表示为：传感器数据 → GPIO → 树莓派 → GPIO → 传感器配置。有关 GPIO 的配置和使用我们已经在示例代码中给出一部分示例，同学们也可以在网络上查找一些其他用法。

为了便于同学们更清楚的了解具体的使用模式，我们将通过讲解一个按键启动程序来帮助同学们更好地理解。详细代码见附件 7。

首先，在附件 1 中，我们找到按键的部分：



由此可见按键与树莓派的 GPIO19 相连，并采用上拉电阻，目的是当按键断开时，GPIO 口处于高电平状态，当按下按钮时，GPIO 的输入变成低电平。

因此，我们对 GPIO19 进行配置。

```
if GPIO.input(BtnPin) == True:
    time.sleep(0.01)
    if GPIO.input(BtnPin)==True:
        GPIO.output(Rpin,1)
        GPIO.output(Gpin,0)
elif GPIO.input(BtnPin) == False:
    time.sleep(0.01)
    if GPIO.input(BtnPin) == False:
        while GPIO.input(BtnPin) ==True:
            pass
        GPIO.output(Rpin,0)
        GPIO.output(Gpin,1)
```

另外注意，嵌入式开发中需要考虑硬件情况，按键可能会出现抖动现象，因此我们只能通过判断是否有“一段时间”的低电平作为依据。其他的检测手法也类似。

- 开发语言：Python / C++
- 运行环境：
 - 树莓派上装有 Python2.7 和 Python3.7 可以自由选择

- 如果使用 C++ 进行开发需要交叉编译生成可以运行在树莓派上的二进制文件。

5. 远程调试说明（针对未返校情况）

若疫情限制未返校，我们将无法向每个小组提供开发需要的车模，经过讨论决定采用下述的开发模式。

1. 我们向每个小组提供示例代码以及传感器资料，各小组按作业任务要求对功能进行实现。
2. 我们定期（暂定为每周，临近 DDL 时可能缩短周期）收集各小组的代码/可执行文件，进行实机运行，将输出结果反馈给对应小组。
 - a. 各小组在实现代码时，将需要的调试信息用输出到文件的形式进行输出，助教执行代码后，会将文件发回各小组。
 - b. 调试是按批次进行的，在两次调试 deadline 之间，不提供额外参与调试的机会
 - c. 调试是自愿的，各小组根据自身开发情况进行选择，不硬性要求

6. 附件说明

附件 1 : 树莓派巡线小车原理图.pdf

附件 2 : TB6612FNG 数据手册.pdf

附件 3 : 循迹传感器模块原理图.png

附件 4 : 红外避障模块原理图.png

附件 5 : HC-SR04 超声波-原理图.pdf

附件 6 : HC-SR04 数据手册.pdf

附件 7 : button.py

附件 8 : 圆周避障.pdf

附件 9 : IR-Receiver.pdf

7. 附录

附 PWM 常用函数示例：

1. 创建一个 PWM 实例：`p = GPIO.PWM(channel, frequency)` # (引脚 · 频率)
2. 启用 PWM：`p.start(dc)` # dc 代表占空比 (范围： $0.0 \leq dc \leq 100.0$)
3. 更改频率：`p.ChangeFrequency(freq)` # freq 为设置的新频率，单位为 Hz
4. 更改占空比：`p.ChangeDutyCycle(dc)` # 范围： $0.0 \leq dc \leq 100.0$
5. 停止 PWM：`p.stop()`