

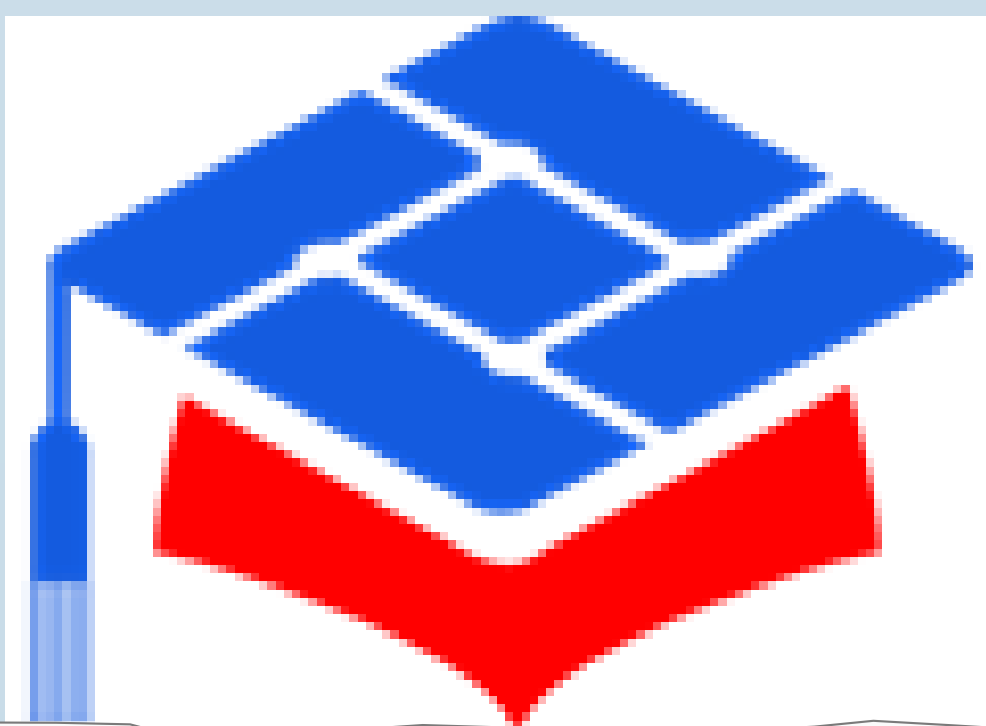
基于FreeRTOS多线程循迹避障智能车

林沐晗¹, 方文欣¹, 潘文安^{1,2}

¹香港中文大学（深圳）

²指导老师邮箱: SimonPun@cuhk.edu.cn

团队名称: Light Chaser



作品简介

RISC-V 开源架构为物联网时代提供了新的芯片选择，近年来受到越来越多的关注。本项目使用Microbit小车搭载一枚RISC-V SiFive Learn Inventor芯片，实现了小车的循迹功能，并通过超声波模块，实现避障功能。为实现虚线避障功能，我们加装了线性CCD模块。我们通过调用PWM函数改变占空比实现了速度控制。为了实现多线程，我们引入FreeRTOS操作系统，将多个程序的进程并列执行。在实验中，我们实现了虚线行驶、倒行、避障、循迹等功能。

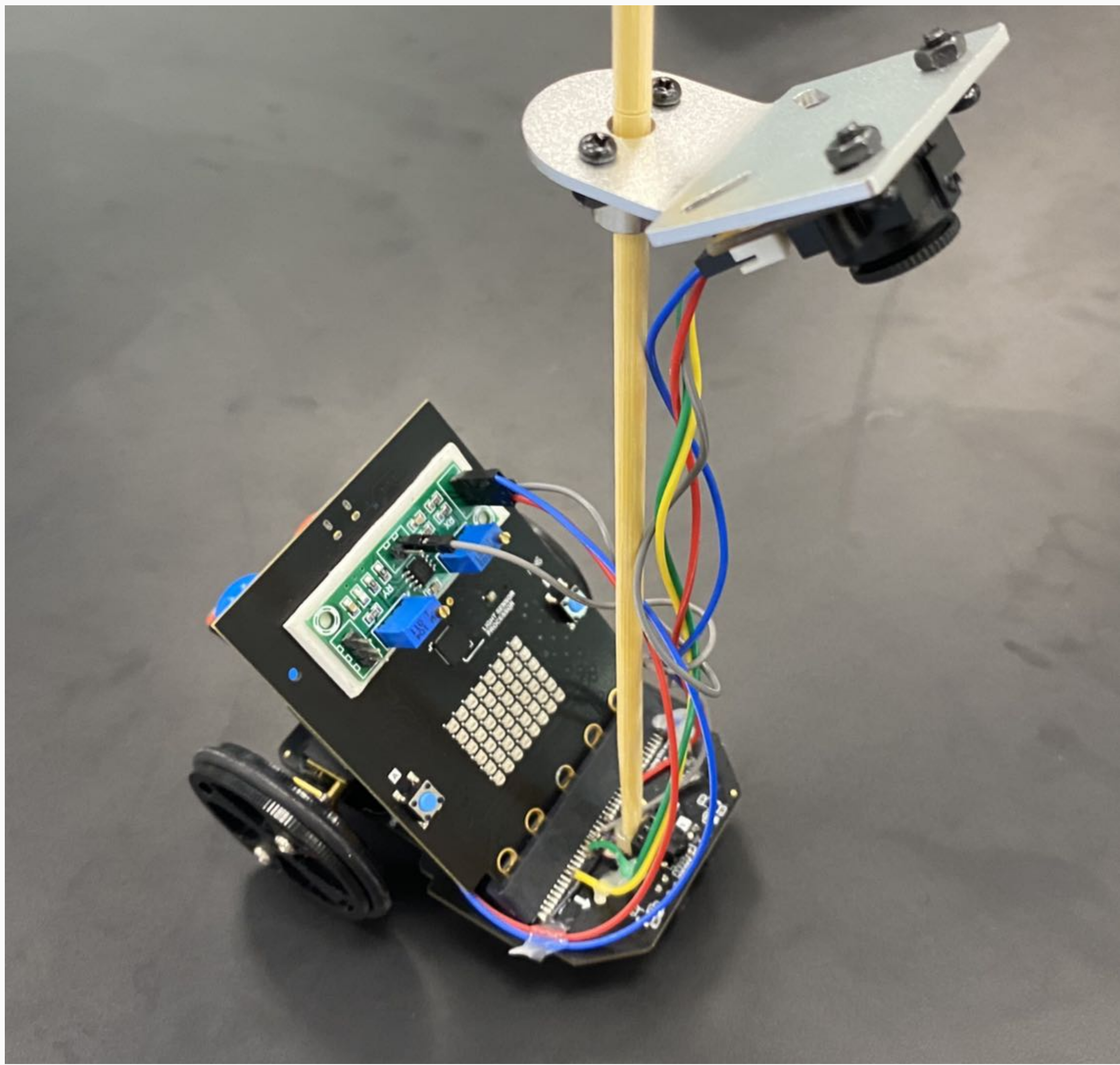


Fig. 1: 小车整体结构图.

小车的循迹与避障实现

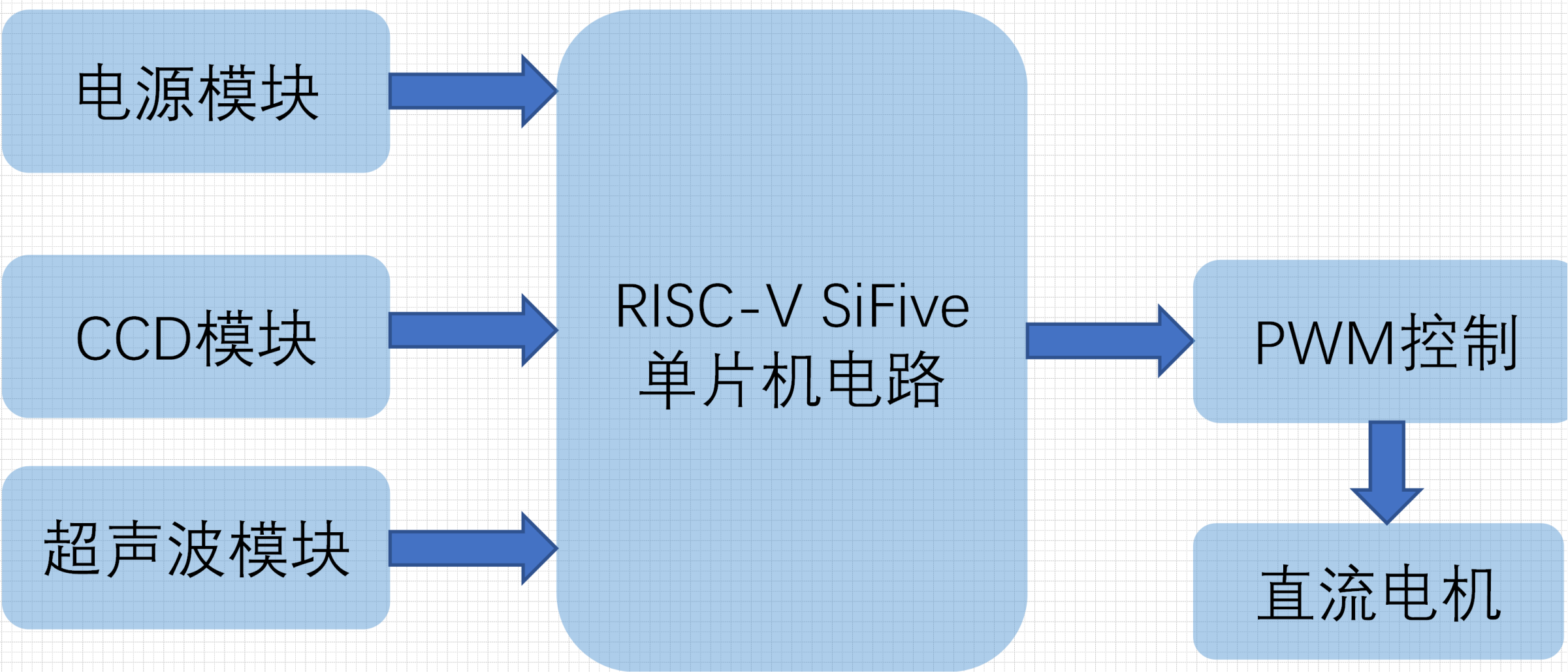


Fig. 2: 小车整体硬件架构图

线性CCD模块是基于TSL1401线阵CCD的图像采集模块获取赛道图像信息，为智能小车的循线行走提供决策依据。线阵CCD的一个曝光周期内可以采集视野范围内一条线上的灰度值。通过A/D采样获得的数据其实是通过运算电路得到的像素点的电压值，该信号的大小与光强和积分时间成正比。在光线强的情况下输出灰度值高，甚至会呈现饱和状态，在较弱光线条件下输出灰度值过低，正对光线和背对光线都会引起输出参数的变化。

在进入障碍区之前超声波会一直工作，赛道之间的距离为60cm, 当超声波测出的距离大于30cm时程序会判断为无障碍物。本队小车将按照Z字形穿过障碍区。小车进入障碍区之前，当超声波测距小于30cm时，小车停下向左转，转至检测不到障碍物，即距离大于30cm时，小车直行进入障碍区，直至红外传感器检测到黑线，停止。在障碍区中，当小车左边红外传感器触碰到边界后，小车将向右转，转至超声波探测不到障碍物，再直行至右边边界，小车左转至超声波探测不到障碍物，重复操作直至小车驶出障碍区。

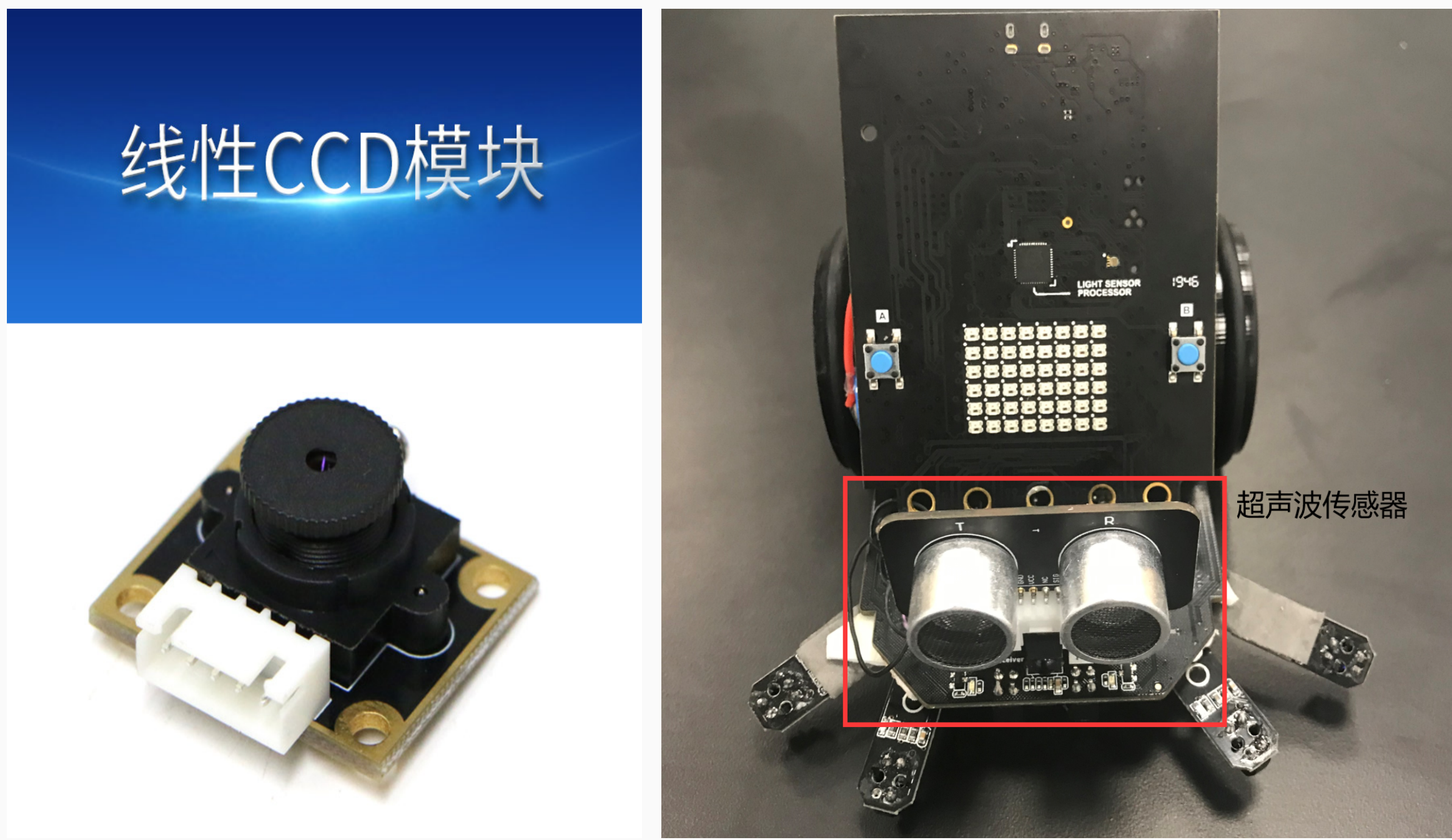


Fig. 3: 循迹和避障中使用的红外和超声波传感器

PWM速度控制

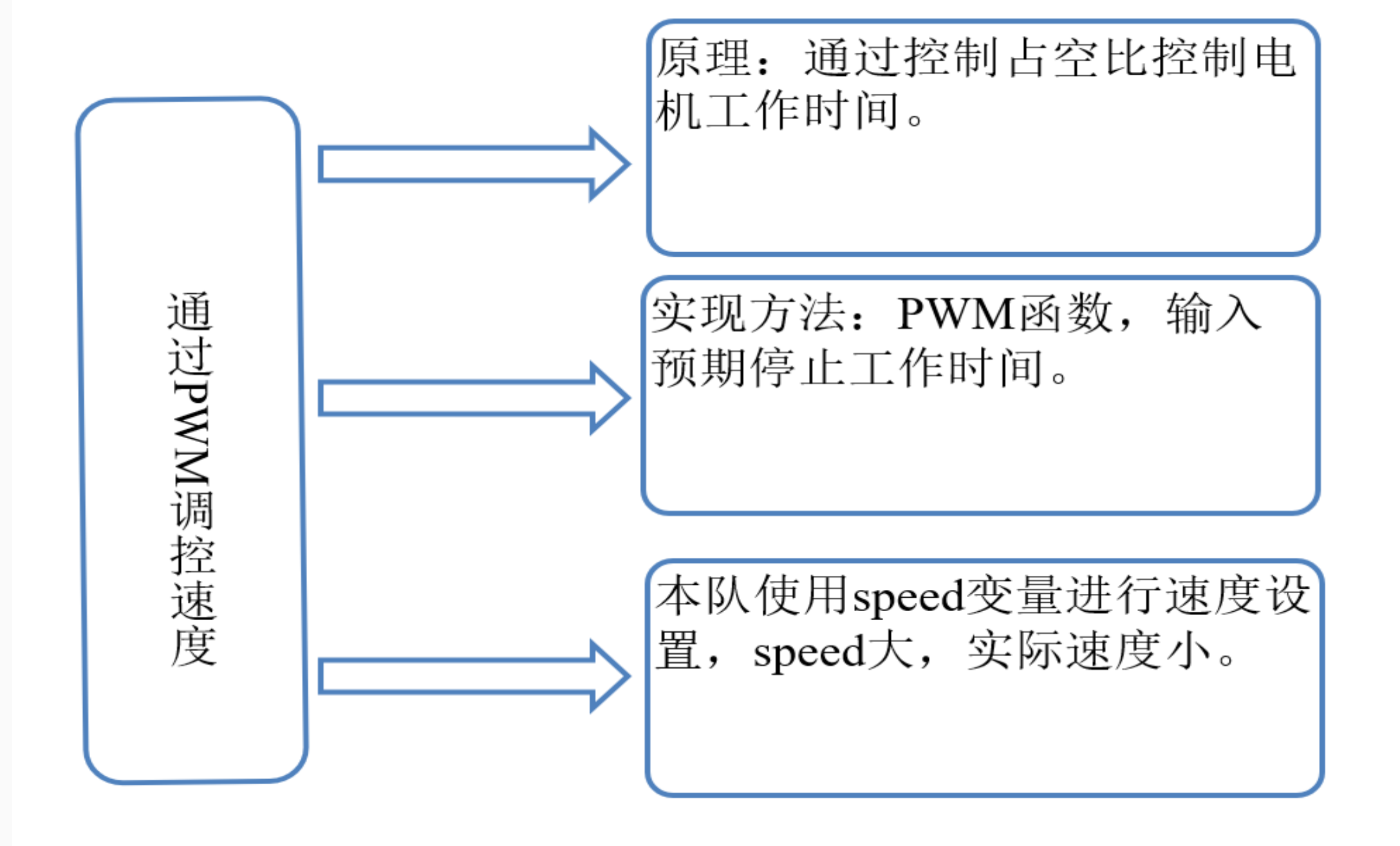


Fig. 4: PWM速度控制

鉴于开发板本身 PWM 不完整，本队在程序中自主设计了 PWM 函数。函数输入量 speed 的大小决定了 PWM 的占空比的大小。占空比是指在一个脉冲循环内，通电时间相对于总时间所占的比例。占空比越大，则小车电机转动速度越快。本程序设定 speed 最大值为 255，最小值为 0。当输入 speed 为 0 时，小车马达全速运转，当输入 speed 为 255 时，小车马达不运转。下图一为芯片各个 GPIO 口连出的 PWM 接口，红色部分为小车电机对应的 GPIO 口，缺失 PWM 的 GPIO 口为 4 和 5。因此，本队自主设计了四个 PWM 函数来实现小车直行，左转，右转，倒车的控制。

FreeRTOS原理

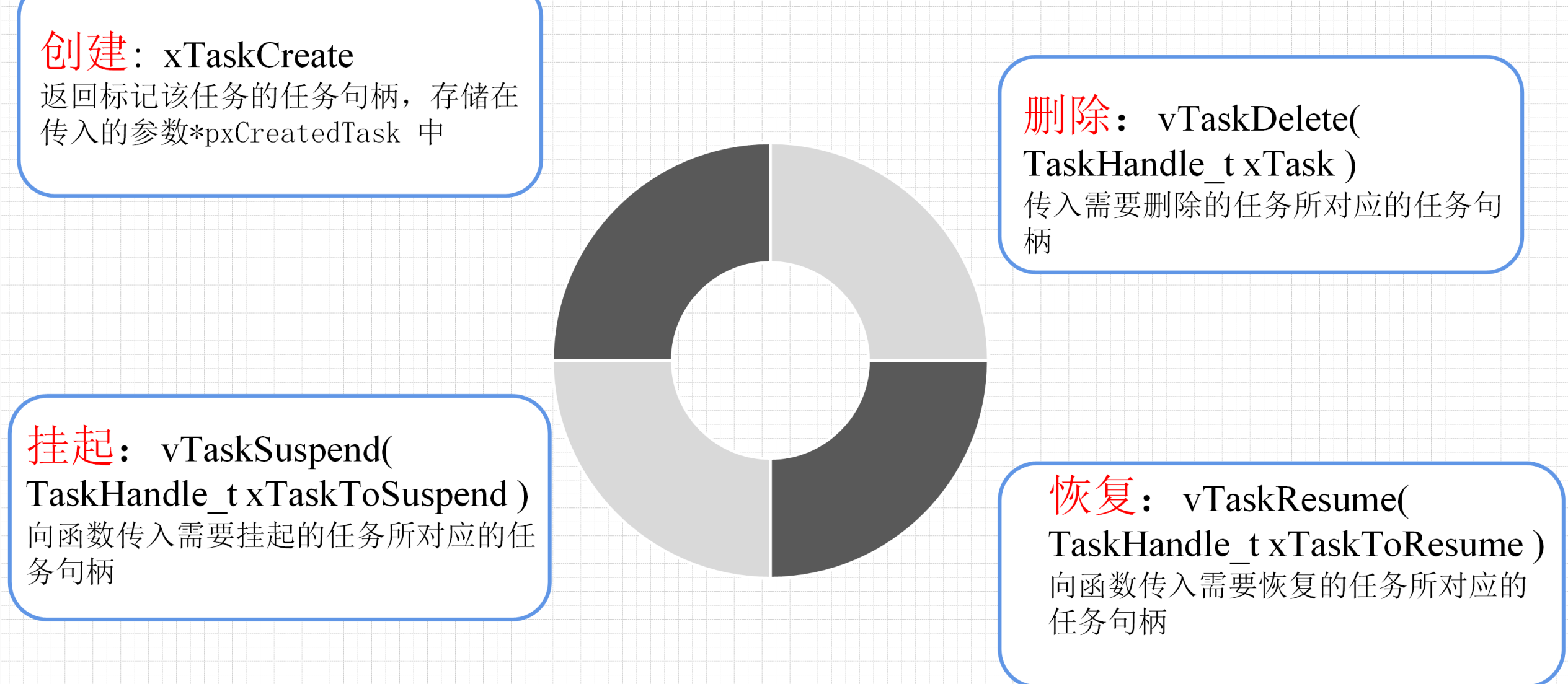


Fig. 5: 多线程与多任务的实现中基本操作.

FreeRTOS 是一个能满足嵌入式硬实时要求的实时内核（或调度器），它实现了应用的多线程设计要求。对于仅有一个内核的处理器来说，任意时间只有一个线程能被执行。内核根据设计者分配给各个任务的优先级高低来决定哪个线程先被执行。任务优先级能保证一个应用满足其执行时的时限要求。队列被用于任务与任务之间或是中断与任务之间的信息传递。队列实质上是链表，每一个链表项大小相同、至多记录一条信息。消息一般遵循先进先出的规则进出列表。对于一个队列来说，任何一个任务都可以向其传递信息（其实是记录信息的地址），实质是将传递内容保存在链表项中。同样，任何一个任务都可以从该队列中读取信息。被读取过的信息根据程序设置可以被删除也可以被保留以供其他任务读取。

FreeRTOS在本项目中的应用

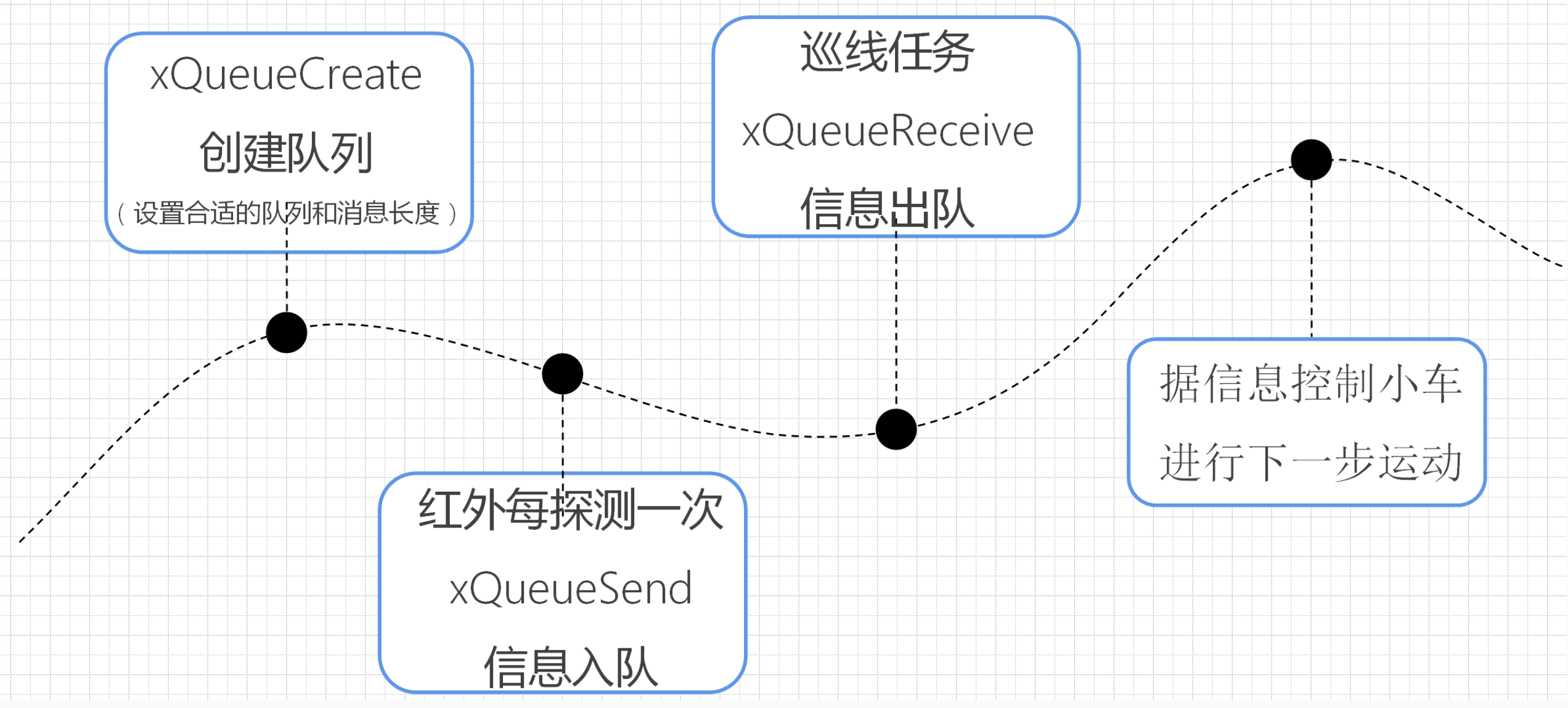


Fig. 6: 多线程与多任务实现.

仅使用小车裸机程序时，小车无法同时执行多项任务。这会导致小车执行任务的实时性和效率较低，浪费了处理时间和内存资源，能耗较高。同时，对于多项任务的调度工作完全依赖于开发者编写的程序，不仅增加了开发者的工作负担，也使得任务程序更繁杂，更容易产生错误。

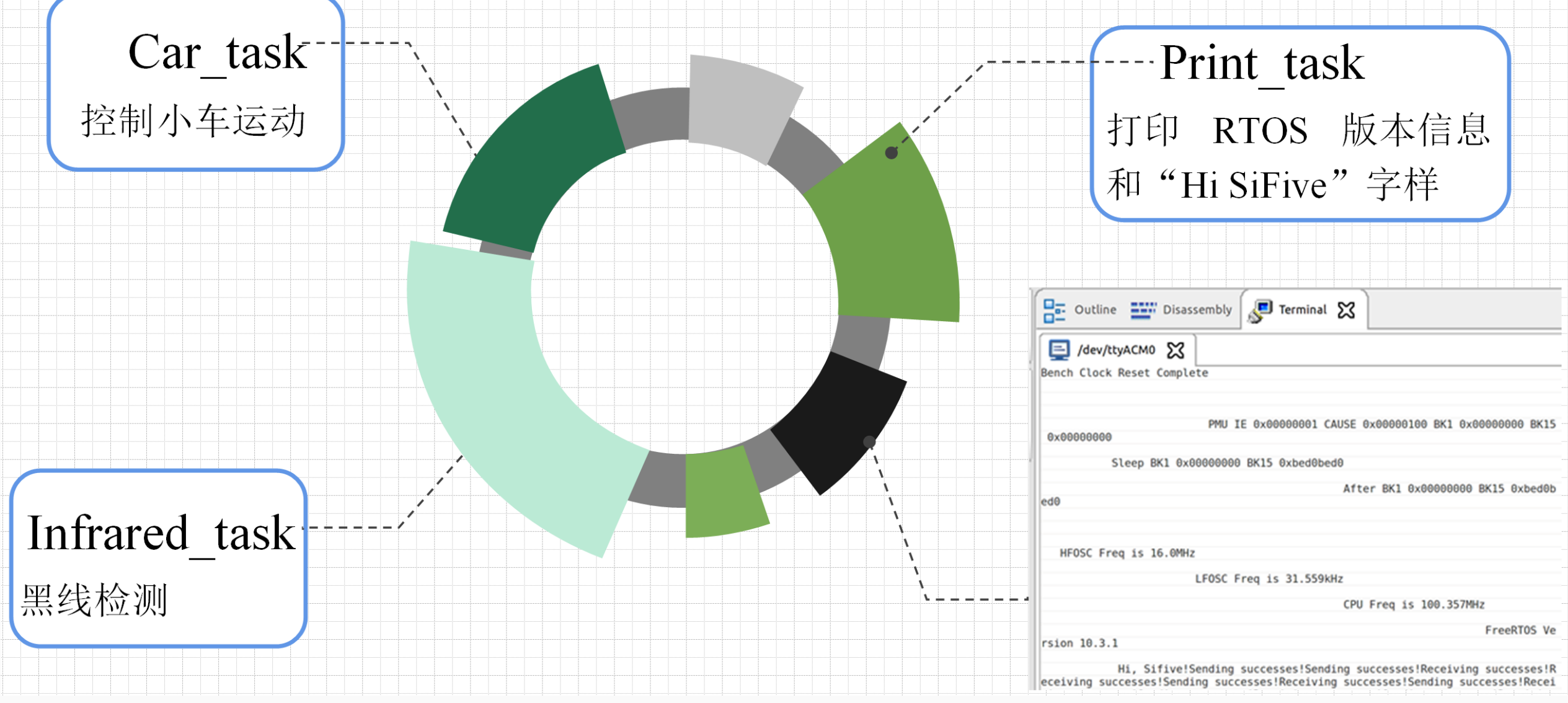


Fig. 7: 各线程功能说明.

任务有四种状态：运行态、就绪态、阻塞态和挂起态。每个任务都拥有指向自己的状态列表项和事件列表项。这些列表项从属于某个列表，例如就绪列表、等待某个事件的列表等等。它们记录了各个任务的状态或是阻塞状态下等待的事件。任务调度器总是会从就绪列表选择优先级最高的任务来执行。当某一事件发生，等待该事件的任务 A 将会从阻塞态进入就绪态，实质是其状态列表项从一个列表中删除后进入另一个列表。如果任务 A 的优先级高于当前执行的任务 B 的优先级，且抢占式内核对应的宏被设为 1, A 将会夺取 B 的执行权转入运行态，任务 B 的执行因此被中断。若是任务函数中调用了任务挂起函数，则目标任务将被挂起，CPU 使用权亦将被转让。只有当其他任务调用函数来恢复这个被挂起的任务时，该任务会重新进入就绪态等待被执行。

本项目应用实时操作系统很好地解决了这些问题。当小车执行延时程序、电机持续输出相同电压时，处理器实际上是处于空闲状态。应用 RTOS 可以将该时段利用起来：延时的任务进入阻塞态，处理器另从就绪列表中选择优先级最高的任务来执行。这样就实现了“多任务同时进行”的程序执行效果。

总结与展望



Fig. 8: 项目总结.

本项目使用Microbit小车搭载RISC-V SiFive Learn Inventor 芯片，实现了小车的循迹功能，通过调用PWM函数实现对小车速度的控制。为了实现多线程控制，我们引入了FreeRTOS，最终达到了多程序并行执行的效果。在本项目报告中，我们详细介绍了项目实施的过程并展示了实验结果。通过本项目，我们对RISC-V芯片架构有了充分的了解，我们将在后续的工作中进一步完成智能小车的循迹避障功能，并尝试使用其他传感器，如激光雷达，深度摄像头等，实现对小车的控制。