

# 嵌入式系统——基于树莓派车的初级设计

刘仲瑄

兰州碧桂园学校 甘肃兰州 730000

**摘要:** 目前社会正处于大数据转人工智能的阶段,也是未来的风口。而嵌入式开发作为计算机科学人工智能的重要分支,为科技发展也作出了重大贡献。作为未来科技发展的重点之一,对于 AI 的教育也应该提前并普及。本文将介绍基于树莓派车的循迹,避障,识别三种初级设计,以便初学者了解嵌入式开发的初步原理。

**关键词:** 嵌入式系统;树莓派车;设计

Embedded System – Elementary Design Based on Raspberry Pike

Liu Zhongxuan

Lanzhou Country Garden School Lanzhou Gansu 730000

**Abstract:** Currently, society is in the stage of transitioning from big data to artificial intelligence, which is also the future trend. As an important branch of computer science and artificial intelligence, embedded development has also made significant contributions to the development of technology. As one of the key areas for future technological development, AI education should also be advanced and popularized. This article will introduce three basic designs for tracking, obstacle avoidance, and recognition based on Raspberry Dispatch, so that beginners can understand the basic principles of embedded development.

**Keywords:** embedded system; Raspberry Pi Car; Design

## 1 循迹模块

循迹的目的:为了实现让树莓派车沿规定好的黑线路线行驶

### 1.1 传感器介绍

在循迹模块中,使用数字式的红外线传感器。即红外二极管发射红外光,接收管接受反射的红外光信号。不同的颜色反光效果不同,如果红外光照射在深色物体上,由于深色表面对光的吸收能力强,反射的光较少。而照在浅色物体上,由于浅色表面对光的吸收能力弱,反射的光较多。而当红外二极管发出的光照射在不同颜

色的物体上时,接收管接受的红外光强度也不同,从而可以判断出前面物体的颜色,接收器也会将收到的模拟信号因此通过电压的形式转换成 1 或 0 的数字信号。所以在红外线被深色和浅色的背景反射时,会得到不同的结果。扫描到深色的表面时,会返回 1 信号,扫描到浅色时,则会返回 0 信号。

### 1.2 循迹模块的连接

循迹模块共有三个引脚,分别是 VCC, GND 和 GPIO 接口, VCC 连接电源端, GND 接地线, GPIO 接输出到单片机的 IO 端,同时也作为指令端去控制输入输出。

### 1.3 循迹模块的摆放

因为单个循迹模块无法判断不同信号时所应该做出的反应,所以至少需要两个分别安装在黑线的两侧,通常为了更高的精度,会设置 3-5 个循迹模块。

### 1.4 循迹模块的检测

当左侧循迹模块的返回值为 0, 右侧也为 0 时(若中间有循迹

模块为 1 时),四个轮子转速一样,则为前进。当左侧循迹模块的返回值为 1, 右侧为 0 时,左侧轮子降低转速,右侧轮子转速大于左侧轮子,以达到左转的效果。右转同理。

### 1.5 循迹编程相关

在初步编程时需定义引脚,按键,电机等,并对它们和 PWM 进行初始化。

## 2 避障模块

避障的目的:为了实现让树莓派车在遇到障碍物时转弯并寻找新的路线

### 2.1 传感器介绍

在避障模块中,我们是基于 STM32 避障超声波传感器。在避障模块中含有超声波发射器与接收器,利用超声波测距进行预知并提前转向。超声波测距的原理就是利用声波在空气中传播的速度已知(精确值 343.2m/s),并测量得知超声波碰到障碍物后返回回来的时间,根据时间差计算得知实际距离。(计算公式:  $s = ct/2$ ,  $s$  为实际距离,  $c$  为声速,  $t$  为从发射到接收的时间)。当持续 10 微秒的脉冲被引脚检测到时,超声波发射器将会以 40kHz 的频率发射 8 个超声波脉冲,这种脉冲模式将脉冲与周围环境噪音进行区分,使其精准接受。STM32 避障模块提 2cm 到 400cm 的测距距离,测量精度可达 2mm。小于 2cm 测量可能出现误差或无法测量,导致不正常输出。

### 2.2 循迹模块的连接

循迹模块共有四个引脚,分别是 VCC, GND 和 GPIO 接口, VCC

连接电源端, GND 接地线, GPIO 一个连接 input ( trig ) 端, 另一个连接 output ( echo ) 端。

### 2.3 避障模块的摆放

为接收超声波更加方便, 一般将避障模块摆放到小车的正前方。

### 2.4 避障模块的检测

当信号返回时, 同过 IO 口 echo 输出高电平, 高电平显示的时间间隔就是发射点到障碍物的实际距离, 当计算出距离时, 便可以设定距离障碍物多远时向右 ( 左 ) 转。

### 2.5 避障编程相关

在初步编程时需定义引脚, 按键, 电机等, 并对它们和 PWM 进行初始化。

## 3 图片处理

循迹模块的目的: 为了实现让树莓派车沿识别的图形方向行驶在进行图片处理之前, 我们需要下载 opencv 的函数库。其中包含 cv2.imread, cv2.imshow, cv2.imwrite 三个函数, 分别对应读取, 显示和写入三个功能。

### 3.1 图片的读取

cv2.imread ( filename, flags ): 这个函数需要输入两个参数。filename 处需要 输入图片的绝对路径 ( 非相对路径 )。flag 处输入需要的表示形式, 常用的表示形式一般有三种。

I.cv2.IMREAD\_UNCHANGED or -1 #不改变图片本身

II.cv2.IMREAD\_GRAYSCALE or 0 #表示灰度图

III.cv2.IMREAD\_COLOR or 1 #表示彩色图

### 3.2 图片的显示

cv2.imshow ( window\_name, img ): 这个函数同样需要输入两个参数。window\_name 处需要输入窗口的名称, image 处需要输入储存数据的对象, 函数 imshow ( ) 通常与 waitKey ( ), destroyAllWindows ( ) 函数联合使用。函数 waitKey ( ) 用于绑定键盘: 它带有一个数字参数, 指明窗口显示的时间 ( 单位 ms ), destroyAllWindows ( ) 函数用于删除此图片所有的窗口。

### 3.3 图片的保存

cv2.imwrite ( filename, img ): 这个函数也需要输入两个参数。filename 处输入文件名及其扩展名 ( jpg, exe 等 )。img 处需要输入储存数据的对象。这样就完成了图片保存。

### 3.4 图片的 HSV 处理

HSV 是指一个三维色彩空间。HSV 表达彩色图像的方式由三个部分组成: Hue ( 色调、色相 ) Saturation ( 饱和度、色彩纯净度 ) Value ( 明度 )。我们平时使用的图像都是 RGB 模型, 这种模型由三原色的线性分量的组合形成所有颜色。由于这种模型中改变颜色和三个分量都有关系 ( red, green, blue ), 所以当颜色连续变化时表现并不直观。所以在表达颜色时更常用 HSV 模型, 它能非常直观地表达颜色的色调、鲜艳程度和明暗程度, 方便进行颜色的对比, 也更贴近人类对颜色的感知。这也使 HSV 模型在分割颜色时有更

出众的表现, 跟踪物体也更具有准确性。常用 cv2.cvtColor ( img, cv2.COLOR\_BGR2HSV ) 将图片从 RGB 模型转换为 HSV 模型。

### 3.5 图片高斯滤波处理

高斯滤波是一种线性的平滑滤波器, 可以很好的抑制噪音 ( 对处于正态分布的噪音更具效果 ) 在实际场景中, 我们通常会假定图像包含的噪音为高斯白噪音, 所以在许多实际应用的预处理部分, 都会采用高斯滤波抑制噪声。在 python 中, 我们可用 cv2.GaussianBlur ( img, ( 3, 3 ), 1.3 ) 实现高斯滤波。其中, ( 3, 3 ) 是滤波器大小, 1.3 为滤波器的标准差, 如果标准差处参数为 0, 则程序会根据滤波器大小自动计算得到标准差。

### 3.6 图片边缘检测处理

边缘检测是将图片中的边缘标识出来。边缘包括颜色深度的不连续, 表面方向不连续, 物质属性变化和场景照明变化。为寻找图像边缘的强度和方向, 我们可以选择工具梯子算法, 以此得到可能的边缘。之后则进行双阈值筛选。在进行前我们需要设置 'high' 与 'low' 两个阈值, 当灰度变化大于 high 的, 设为强边缘像素, 在 high 与 low 之间的, 设为弱边缘像素, 低于 low 的, 删除。我们可用 cv2.Canny ( img, low, high ) 实现边缘检测。

## 4 代码思路

- ①将所用模块与引脚初始化
- ②优先写出移动函数代码, 即前进后退左转右转及其转速
- ③写出传感器返回的信号函数
- ④编写主代码, 将结束条件提前设定好
- ⑤将返回的信号与移动函数联系起来, 达到想要的目的

## 5 可能遇到的问题

- ①前进后退时速度设定超出电机转速阈值致其无法正常转动 ( 改变参数 )
- ②程序反应时间过长致其准确性下降 ( 加强代码逻辑性, 使其循环次数减少, 也可多定义对象进行优化 )
- ③主板和点击过热 ( 添加散热板 )
- ④检测精确度不高 ( 改变参数, 多添加传感器缩小误差区间 )
- ⑤摄像头安装位置不合理 ( 最好将摄像头安装在小车中间前部 )

## 6 总结

嵌入式系统的灵活, 方便, 性价比高的优势注定会使它成为智能化中的中流砥柱, 树莓派作为它的副产物相较于其他品牌更加通用便捷, 也即将成为年轻人可以重点关注的对象, 提前了解学习有很大的价值。希望本文能对你有所帮助。

### 参考文献:

- [1]闫黎明, 刘冰寒.10 kV 配电线路保护测控装置的应用设计[J].通信电源技术.2021, ( 2 ) .
- [2]王泉, 杨天若, 朱大开, 等.智能嵌入式系统结构与软件关键技术专题序言[J].西安电子科技大学学报 ( 自然科学版 ).2021, 48 ( 6 ) .