基于树莓派的安防巡逻机器人

曹沛文

2014自动化

河南大学民生学院

(单位全名 部门全名，市(或直辖市) 邮政编码) 宋体小四

**摘要**： 为了降低安防成本，提高安防巡逻的覆盖率、预警能力及自动化水平，本文提出一种解决方案，结合日益成熟的机器人科学、传感器科学，用较低的成本实现机器人在室内进行安防巡逻工作。

本文研究的安防巡逻机器人采用斯坦福人工智能实验室建立的ROS系统作为机器人的核心控制系统，将ROS运行在Raspberry Pi上，使用STM32作为下位机，依据里程计和超声波传感器的数据导航，依据烟雾传感器、温湿度传感器及光学摄像机作为巡逻检测指标，完成既定路线的安防巡逻。

实物尚未完成，保留此段待补充。

解决的问题是什么，采用了什么方法，结果怎么样

**关键词**：ROS 树莓派 PID 巡逻机器人

**Abstract**：something write here

**Keywords**：Raspberry Pi Security Patrol PID Adjustment parameters online ROS

# 引言

机器人技术已经发展成熟。基础技术包括运动、控制、传感器、定位等，在基础技术上发展出来的有地图实时构图、路径自主规划、图像识别、语音识别等。更抽象的层面还有基于大数据各种应用，如智能调度、需求预测等。

使用机器人代劳人类的体力劳动可以有效降低人事成本、人事管理成本及人事安防成本。机器人工作不受时间限制，工作内容的数据化程度较高。机器人已经在工业生产领域普及，技术越来越成熟，成本也不断降低。安保行业机器人普及率目前几乎为零，其市场潜力巨大。相信不久的将来，机器人将开始从事安防工作，伴随着大数据，安保行业将会发展到一个新的高度。

本文旨在研究安防巡逻机器人的设计方案，并结合图像识别技术，试图提高机器人的巡逻水平。

# 选题的背景与意义

## 安防巡逻机器人研究背景

随着城市化进程的推进和各行业的深入发展，各个工作岗位及工作场地的功能划分越来越细也越来越规模化，特别是如超市、停车场、会展中心、物流港等各大公共场所的人流量、物流量伴随着效率的提高也在大幅增长。与之配套的安保工作缺口巨大，使用传统的安保人员已难以满足这些场所对安保事务高效管理的需求，安保工作自动化迫在眉睫。

机器人技术已经发展成熟。硬件技术方面，无论是控制核心PCB板的设计与制造，或机器人传感器的如CCD摄像感知系统、超声波测距仪、红外传感器等的标准化模块化，亦或电动、气动、液动机器人机械传动系统的小型化，

基础技术包括运动、控制、传感器、定位等，在基础技术上发展出来的有地图实时构图、路径自主规划、图像识别、语音识别等。更抽象的层面还有基于大数据各种应用，如智能调度、需求预测等。

机器人产业链尚不完备

## 安防巡逻机器人的现状

安防巡逻机器人主要依赖多传感器融合技术、导航技术、智能控制技术。这三个技术在近几年都有较大的发展但尚不成熟，基础技术不完备导致产业发展缓慢，安防巡逻机器人产品功能简陋、种类单一、价格昂贵。[1]

当前市面上大多数企业、单位及个人的安装主要由云台摄像头、建筑火警系统、门窗红外报警器及警卫人员组成。云台成本较低，且与公安系统联网后可以构成全覆盖的大规模监控系统，具有较好的警示作用和事后可追查性。建筑火警系统是个地方消防部门强制要求的，商品房、写字楼、商场等，火警检测装置及自动灭火装置已成为建筑的一部分。红外报警器主要用在店面门面、仓库门窗，用来预防非未经授权的闯入行为。安防人员，指组织的保卫处的所有人员，他们即可作上述安防设施的补充，亦可全面取代上述的安保设施。

安防巡逻机器人，实际上就是用机器人取代安防人员的工作。截至到2017年年底，全世界的安保行业仍主要由人类的劳动力构成。安保机器人的技术尚未成熟，市场几乎为零。

迪拜首个机器人警察于2017年5月上岗。同年7月迪拜警方宣布在年末投入无人车上街巡逻，并计划到2030年让机器人占到警力的25%。无论是机器人警察，还是无人巡逻车，其主要功能都是对视野内所有人进行面部识别，搜索排查跟踪嫌疑犯。

还有少量高档小区、高尔夫球场，已经部署了车式安防巡逻机器人，可按照既定既定路线自动寻巡逻，亦可手动遥控，如图1-1。



图1-1 一种量产的安防巡逻机器人

纵观全球的安防行业的自动化水平，最先进的安防机器人产品已能满足较为简单基础的安防需求，但普及率极低，从产品的多样性也可以看出，这部分市场仍未被打开，市场潜力巨大。

## 安防巡逻机器人存在的意义和对人们生活的影响

普及安防巡逻机器人，可以解决安防人员短缺问题，降低安防成本，提高安防系统的可靠性，提高保卫工作的部署效率，提高安保的数据化和自动化。安保行业整体发展不平衡，高端社区及企业的安防安保部门有大量资金支持，安保力量过剩；小机构小企业及个人很少拥有专业的安防团队，安保漏洞百出。安保机器人在执行低级安保劳动时比人力的效率更高，解放了安防人员的生产力，使安防人员可以把精力投入到更高级的安防工作上，减缓了安防人员短缺问题。安防巡逻机器人采用实时建图，划定工作范围后即可自动部署，还能自主采集地形数据，极大的缩短了安防部署的工作周期。机器人工作巡逻不同于人，总结工作经验主要凭主观经验，机器人工作时采集了大量的数据，结合数据科学进行分析，可以提取更抽象的规律，这对安防行业的发展有深远的影响。

安防巡逻机器人的普及对人们生活的主观影响很小。安防工作是幕后工作，安防的目的是预防灾难和损失的发生。

## 小结

从安防巡逻机器人当前的市场规模和整体价格可以看出，安防机器人现在还是一个非常小众的市场，但设计制作该机器人的主要技术已经成熟。

笔者将在下文详细展开讨论，使用技术整合，把相关各个功能模块整合在一起，在功能上进行微创新，制作一款低成本的小型入门级安防巡逻机器人。

# 系统设计

## 功能预期

安防巡逻机器人，要具备代替安保人员执行简单巡逻的能力。结合机器人的功能特点，该系统需要实现以下几点功能：

* 1. 接受指令前进、后退、左转、右转进行移动。
  2. 使用PID对电机进行调速，保证转速与指令值相符。
  3. 能够沿地面粘贴的轨道线自主循迹。
  4. 通过测距传感器实现前方障碍物距离检测，避免碰撞。
  5. 能记录行驶轨迹，并依据里程计实现简单的定位。
  6. 实时检测环境的温度、湿度、烟雾浓度。
  7. 调用摄像头，并将视频流实时发布。
  8. 可以通过网络远程控制机器人。
  9. 可以设置触发警报。

## 系统总体方案设计



图 2-1 系统总方案框图

根据对系统的预期功能的分析，如图2-1所示，可以将该系统分为三层：执行层、控制层、高级层。

执行层：控制电机驱动、读取编码器、读取传感器、检测电压、接受控制指令、PID电机调速。本层负责底层的硬件控制，运算量小，程序代码较短，实时性较好。

核心层：收发执行层消息、收发高级层消息、采集视频消息、有一定的数据处理能力。本层向下负责输出硬件控制指令、收集传感器数据，向上负责汇总数据、接受高优先级控制，程序代码规模较大，有较大的运算量，无法保证良好的实时性。

高级层：接受从核心层发来的处理过的信息、处理人机交互信息、处理机器人采集到的视频等数据流，可以同时连接控制多个核心层。本层程序代码量庞大，处理数据海量，需要非常强的运算能力，数据连接、处理的实时性都很差。

分层后整个系统可以同时兼顾实时性和运算能力。

尺寸上，执行层和核心层使用的是微型MCU和嵌入式SoC，且和众多外设直接相连，整合后连为一个整体，尺寸较小，移动灵活；高级层使用工作站等高性能计算机，体积大，不便于移动和携带。故本系统所设计的机器人包含执行层和核心层两部分，高级层仅搭建简单功能供测试，投入生产时按照实际需求再对高级层部分做开发。

功能上，可以将系统分为运动模块和计算模块两个功能模块。执行层执行机械运动，核心层和高级层负责执行数据运算。执行层功能简单，使用小规模的代码可以很好的保证系统的实时性，在进行PID控制中实时性尤为重要。

综上，整个系统的重点在核心层。

## 系统设计：执行层

本层系统设计的关键时保证实时性，故系统控制流程采用采用少循环、多中断的设计。系统上电后，初始化环境变量，初始化相关I/O口的状态，初始化相关外设和芯片，激活所有使用的定时器、计数器，然后进入循环体，等待中断发生以执行对应的中断程序，等待接受指令代码并按照代码执行既定的程序。控制层系统逻辑框图如图2-2所示。



图2-2 控制层逻辑框图

最高优先级的中断时产生PWM波的中断，PWM波频率较高，分辨率较高，代码仅进行少量运算和控制PWM口波形，为了保证了对电机电压的精确控制，必须保证该中断的实时性。

次一级的中断时PID中断，PID控制必须是实时的。离散的PID控制参数设定与采用周期有关，周期的不稳定将导致PID控制的效果变差。进入中断中先读入编码器连接的计数器计数，获得间接的速度采样，根据队列中最近若干次采样数据，调用算法产生输出。

其次时串口中断。串口负责接受上位机发送的指令。串口一次只能接收一个字节，控制命令需要将多个字节拼在一起。将串口接收函数放入中断，可以在不影响电机调速系统稳定性的前提下提高系执行层对命令的响应速度。

最低级中断是状态发布。采集里程计信息，反馈当前转速，测定电源电压，发送传感器读数。本中断采用定时中断，使状态信息按照固定频率发送给上位机。

整个执行层主要围绕对电机转速的控制工作，使用光电编码器对转速采样，构成一个闭环的转速控制系统，实时性是本层系统设计的核心。

## 系统设计：核心层和高级层共从组成系统

核心层和高级层在硬件上分离且有所分工和差异，但从系统层面看，它们仍是一个整体。系统部分的运行不依赖外设的型号，使用统一的标准将数据接入系统。系统负责向执行层发送抽象的转速命令，交由执行层解析完成调速，执行层还会实时发送里程计数据。对系统而言，小车的运动变成了一种逻辑状态，传感器的检测结果作为数据映射到寄存器中，抽象的系统就此基础上运行。

ROS(Robot Operating System)是斯坦福大学人工智能实验室和机器人技术公司Willow Garage合作开发的机器人后操作系统，它开放源代码，使用消息传递模式运行，可以抽象底层硬件，支持多进程，支持分布式部署，功能包齐全。[2]它支持使用C语言、Python等多种语言开发程序包开发，对硬件和软件都有良好的兼容性。这里选用ROS还能很方便地部署在基于Linux系统运行的核心层硬件上。



图2-3 核心层运动控制逻辑框图

外设采集的信息有：温度、湿度、烟雾浓度、碰撞距离、声音、图像。将这些数据交给运行着ROS的上位机树莓派处理，做出决策。

核心层像是一位警车司机，按照既定路线站点巡逻或按调度中心（高级层）指示行驶。核心层基于一定的规则，将小车依次移动到若干个特定的目的坐标。这个坐标可以是固定的巡逻线坐标，也可以是高级层根据计算或人为输入的地图坐标。核心层保证机器人能安全准确的移动至既定坐标，不关心为什么要去那里，避免发生碰撞、跌落、侧翻。核心层接到坐标后，输出合适的速度指令将。若前方出现障碍物，将自主减速或原地等待。如果有传感器检测值超过预警信息，将暂停巡逻，在该区域继续采集信息，向高级层发送消息并等待进一步指示。核心层控制小车运动的逻辑框图如图2-3所示。

核心层和高级层通过无线网络连接，构成一个分布式ROS系统，其内部节点及话题设计如图2-4所示。



图2-4 系统内的功能实现

核心层和高级层断开连接后能执行自主巡逻任务，上图是核心层的主要程序节点和消息的话题单元。

下位机串口驱动节点负责驱动硬件串口，接收和发送下位机的串口通讯数据，并将电池电压、电机转速数据按照规定的格式发布到话题。读取转速设置，并将转速设定值按照下位机接收格式通过串口发送。

里程计节点读取车轮转速数据，计算位移偏移量，计算出机器人当前坐标及方向，并发布到“当前坐标方向”话题，供其他节点订阅。本节点在系统中起到了定位的作用。

环境传感驱动节点驱动外设温度传感器、湿度传感器、烟雾浓度传感器，代入已经调试测定的传感器矫正参数，生成准确的环境变量并发布到环境指标话题。

避障驱动传感器定时产生脉冲，驱动超声波模块并计算反馈信号延时，并根据温度和气压，计算出超声波速率，最终准确计算出障碍物的距离，障碍物的距离格式化为毫米单位的短整形数据在“障碍距离”话题中发布。

摄像头驱动节点包含外设的驱动程序。该节点负责驱动高清摄像头，将视频以流的形式发布到“画面”话题，高级层与核心层连接后可以共享该视频流。

模式识别节点负责设定系统的控制模式。最高级别模式为用户控制模式（MODE\_0）：用户使用机器人身上的实体按键可以选择启动、暂停或执行指令的程序。远程控制模式（MODE\_1）：需要核心层与高级层网络互连，可以直接执行远程的控制命令或运行远程设定的控制程序。

导航节点为机器人提供导航服务。节点根据里程计生成的定位信息确定当前位置，读取设定的路线地图，算出下一个需要靠近的导航点坐标，并输出该坐标与机器人的相对方向角。在下一个导航方向消息发布前，该方向上不会遇到地图上的已知障碍物。

速度导航节点负责控制差速电机的设定速度。该节点首先保障输出的速度设定不会因变化过大导致打滑、转弯过快导致侧翻，前方出现障碍物时能及时停车。其次，输出的速度要符合“目地方向”的设定值。

高级层的详细设计属于安防巡逻机器人的集群控制和战术规划，超出了本课题的研究范围，本文不在详细讨论。

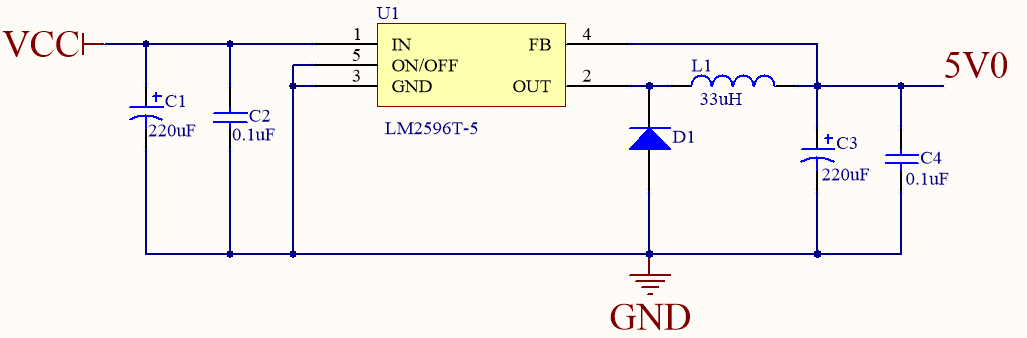
# 硬件设计

本章节根据第二章的系统设计讨论基于树莓派的安防巡逻机器人的硬件设计。包括集成电路板的设计、选用电子模块的连接、驱动电路的构造。

## 电源电路

机器人的所有模块都需要电力，不同模块对电源电压的需求有所不同，机器还需要电压检测装置检测电池电量[4]。这里首先设计管理电源的硬件电路。

机器人的电机驱动模块需要较高的电压，可以直接将7.4V的电源引过去。剩下的控制电路均使用的是5V的电源，可以将7.4V电源降压到5V再使用。电源变压模块的电路图如图3-1所示。

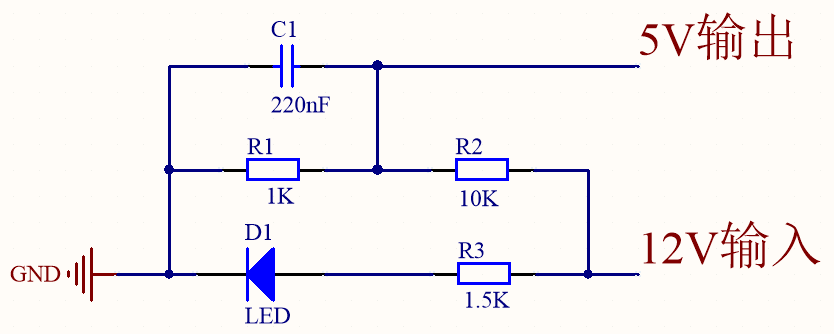


3-1 电源降压稳压模块电路图

在实际测试过程中，经常发生变压稳压电路烧毁的情况，为方便调试，将变压稳压电路制作为独立的模块，通过排插接入系统。

本模块的核心是一颗LM2596芯片，这是一个降压型电源管理单片集成电路的开关电压调节器，输出电流的驱动能力可达3A。该芯片工作原理是通过内部逻辑电路总的放大器与比较器进行运算，当电压超过5V设定值后停止输出，当电压低于5V时电源导通。这个变压原理与PWM脉冲调宽给外设输出特定电压的原理类似。由于不是一个连续的功率输出，故配上由电容、电感、二极管构成的滤波电路后，输出电压更稳定安全可靠。

解决了电源电压问题，机器人还要能够实时检测电池所生的电量。电压检测是通过一个串联电阻电路，根据欧姆定律得到按比例缩小的电源电压，将电源电压输入到电压检测元件上。本方案采用通用的锂电池组，根据锂电池的电量与电压特性，锂电池电量在10~90%时电量与电压呈较完美的线性，单节锂电池的电量在3.5~4.2V，常见的锂电池组由从1S到6S，即满电时电压从4.2V到25.2V不等。本方案采用的MCU为STM32系列，自带内部ADC，最大检测电压3.3V，故电源电压的缩小倍率不应低于8倍，本系统电压检测模块的是11倍分压，如图3-2所示。



3-2 电压检测模块

输入电源与GND之间并联有两条串联电路，二极管LED灯和R3电阻器组成了电源工作指示电路，R1和R2组成了电阻器分压电路，电阻器应选用误差较小的，以提高系统检测精度。分压后的输出电压与GND之间还有一个电容器，用来保护输出端电压的稳定与安全。

## 电机驱动及编码器电路

为了方便做闭环控制和上位机里程计设计，机器人驱动轮的选用配备编码器的直流电机，电机的接线也直接和编码器放在一起，公用一组排线，方便连接安装。电机实物如图3-3和图3-4所示。

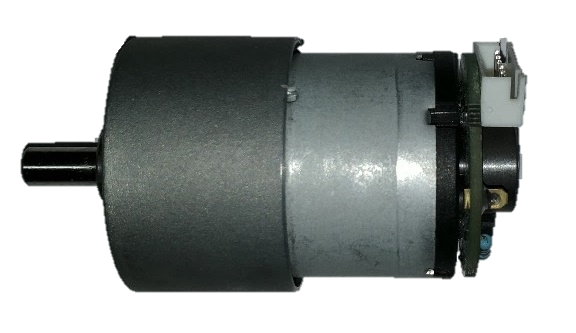


图3-3 带编码器电机俯视图

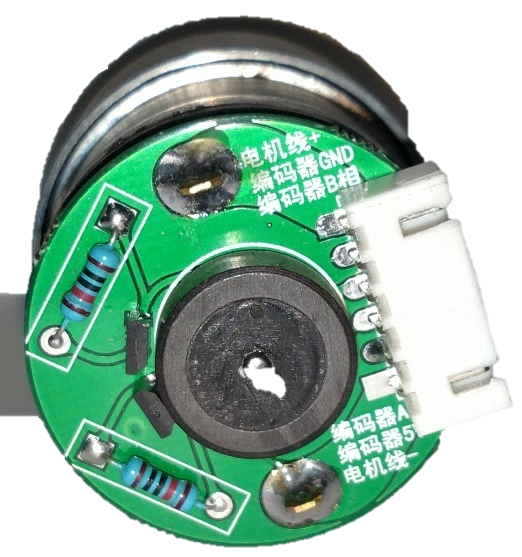


图3-4 电机编码器线路板

编码器由地线GND、5V电源VCC、编码器A、编码器B四根线组成，AB两根线可以直接与单片机I/O口相连。电源线接入所需电源线路。

电机选用12V有刷直流电机，对电机的速度控制可以通过调节电机输入电压实现。本系统使用PWM脉冲调宽实现对电压的调节。如图3-5所示是单片机驱动电机的驱动电路。

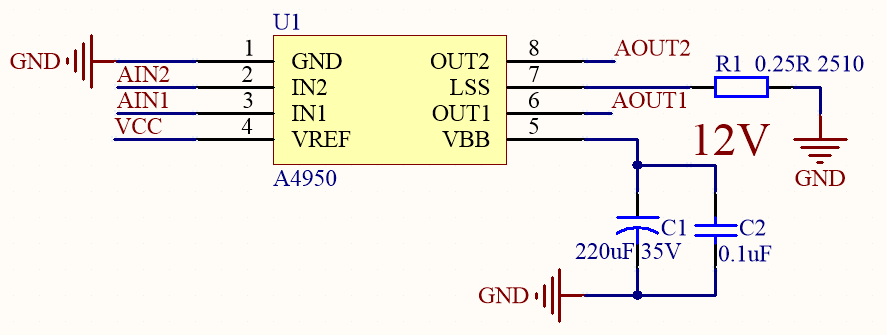


图3-5 A4950驱动电路

A4950芯片是一种广泛应用在汽车领域的全桥DMOS PWM电机驱动器，每颗芯片可提供高达3.5A的峰值输出电流，工作电压在0 - 40V。芯片内部还设有保护电路，包括过电流保护、电动机接地、电源短路、因滞后引起的过热关机、VBB 欠压监视以及交叉电流保护。

GND和VREF分别连接单片机的0V和5V。VBB是驱动电源，本系统使用的是12V的直流电压源，由外置的锂电池组提供。IN1和IN2连接到单片机I/O口，接收单片机由传出的PWM控制信号。OUT1和OUT2是为电机提供电压源的输出，输出与VBB电源相同电压的PWM波。

## STM32最小系统版

本系统选用STM32F103C8T6作为微控制单元，共有35个I/O口，内置ADC模块，1个高级定时器和3个通用定时器，64K的Flash空间，72MHz的机器时钟频率。完全能满足同时使用PWM、串口通讯、编码器计数、超声波等外设同时使用的需求。

最小系统板也采用模块化设计，可以保持系统各个硬件设计的一致性。通过排插接入系统。系统板上包含外部晶振电路、开机复位电路、按键复位电路、SWD接口电路等核心功能。如图3-6所示是STM32最小系统板的原理图。

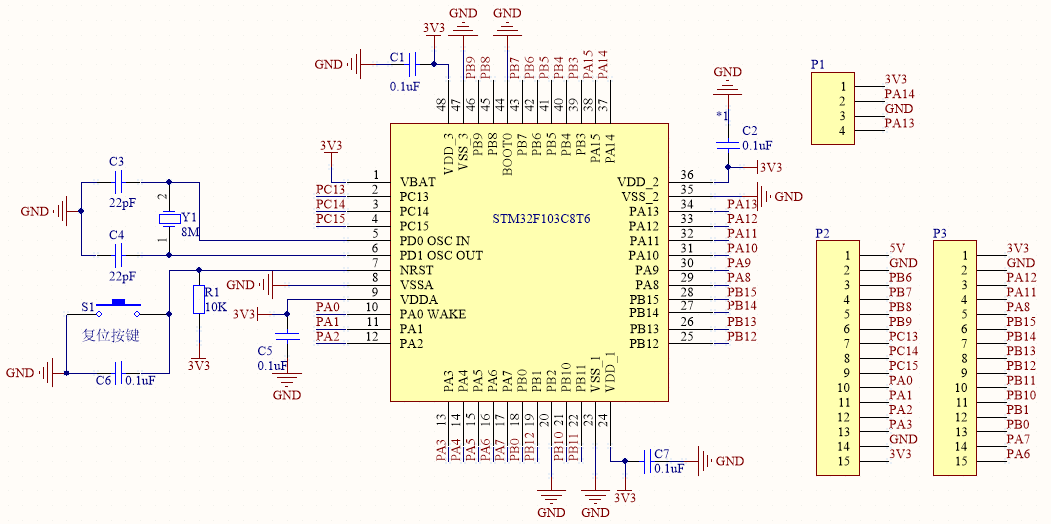


图3-6 最小系统板实物图

## STM32最小系统扩展电路

以上小节的若干硬件以模块的形式制作或选购，还需要一个电路将所有模块集成连接在一起。

在设计原理图时，首先确定电源方案。扩展版需要接入12V电源接口、24V电池电源接口、5V降压稳压模块接口、电源开关以及5V的USB电源输出接口。正面覆铜5V电源，本面覆铜GND。其中未降压的电源输入网络附近不覆铜，防止未降压电源意外接入5V覆铜网络，损坏设备。部分覆铜区域被走线分割的过细，无法承载较大电流，在该区域增加过孔通过另一面走线与其他区域的5V网络相连，保证任意区域5V网络的负载能力。如图3-7所示时STM32最小系统拓展板电路图的正面。

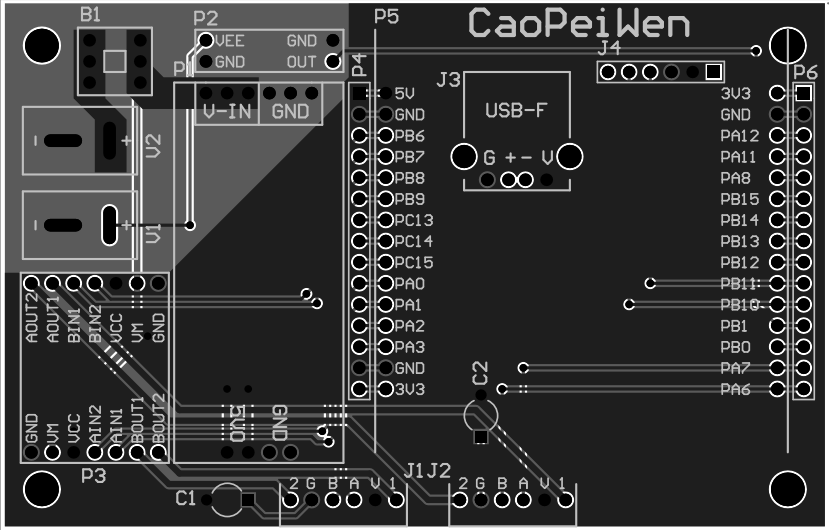


图3-7 STM32最小系统拓展板线路图（正面）

STM32最小系统板接口外加两排排针调试使用。

电机驱动模块尺寸较小，也可以直接插放在拓展板上。电机及编码器接线选用XH-2.54接口。电机驱动模块与12V电源的走线、电机驱动模块到电机接口的走线都要加粗，以保证走线有足够的电流通过能力。

其余的普通走线尽量满足正面横向走线，反面纵向走线的原则。

完成设计后交由PCB代工厂生产。

## 传感器

安防巡逻机器人的巡逻任务依赖传感器读取外接参数，包括预防火灾使用的温湿度传感器、烟雾浓度传感器，巡航时避免碰撞的超声波距离传感器，以及定点监视时使用的人体红外感应器，检测气压的传感器。

超声波距离传感器选用HC-SR04系列，不和开发板集成在一起，安装在机器人的前部，三个的传感器探头指向的方向分别是正前方、左前方30°和右前方30°，三个探头安装时保持发射头和接收头都对应同一高度。如图3-8所示时本系统设计选用的产生波距离传感器的实物图。

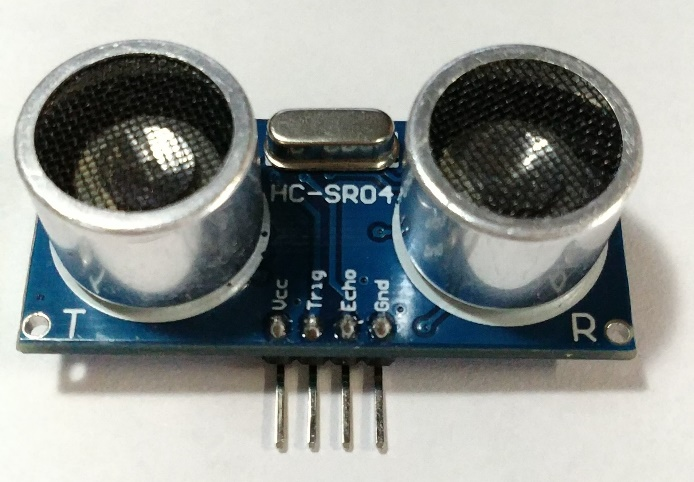


图3-8 超声波传感器实物图

温湿度传感器、大气压传感器、及烟雾浓度传感器需要保持透气性良好，且温度不能受机器人工作发热影响，故将这三个环境传感器安装在机器人顶部。三个传感器模块零件如图3-9所示。

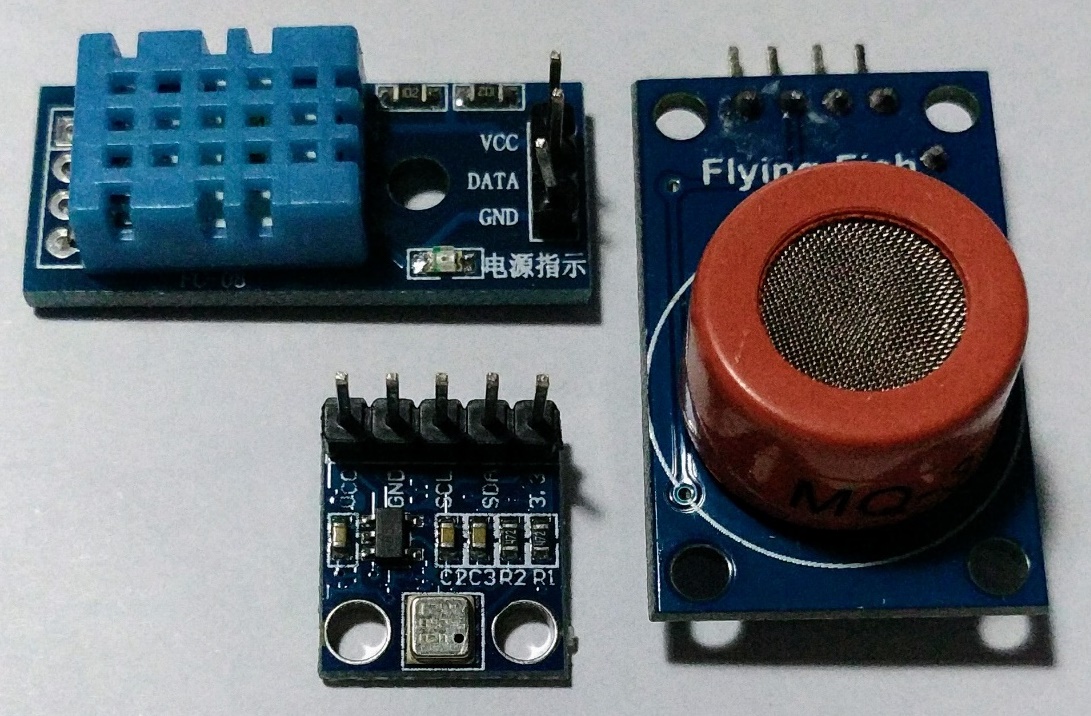


图3-9 空气类传感器

温湿度传感器选用DHT11。这款款传感器有已校准的数字信号输出，湿度的精度在5%RH以内，量程在20~90%RH；温度的精度在2°以内，量程在0~50°。芯片数据使用一总线协议传送，树莓派完美支持一总线协议，可以直接与GPIO口连接使用。

气压传感器选用的BMP180。这款传感器体积小、功耗低，精度可达0.03hPa，量程在300-1100hPa，即海平面下500米至海平面9000米，完全可以满足地面任意地点的需求。芯片数据使用I2C接口传输，也与树莓派完美兼容，直接接到GPIO上就能用。

烟雾浓度检测模块选用MQ-2型烟雾传感器。这款传感器适宜于液化石油气、苯类、烷类、乙醇、氢气、烟雾等的探测，特别适用于消防系统。这时一款模拟输出的空气传感器，空气中敏感的大分子含量越高，输出的模拟信号电压越高。

传感器直接接入树莓派，减少了下位机STM32定时器的开销，也充分发挥了树莓派GPIO方面的优势。

## 上位机树莓派

树莓派是一个只有信用卡大小的卡片式计算机，是一款基于ARM的微型电脑，可以运行多种基于Linux的操作系统。硬件上，树莓派拥有17个完全自由的GPIO口，直接支持一总线、I2C总线、SPI总线控制外设，支持使用USB外设，支持插网线及WiFi。软件上，树莓派运行基于Linux的系统，能在Linux系统中运行的程序树莓派都支持，特别是对Python语言、Go语言等都完美支持。

树莓派的电源由STM32最小系统拓展板上的USB口供电，使用树莓派的USB口连到STM32最小系统板的USB转USART口。上位机和下位机使用串口完成有线通讯。如图3-10所示是上位机树莓派与下位机STM32系统连接后的实物图。

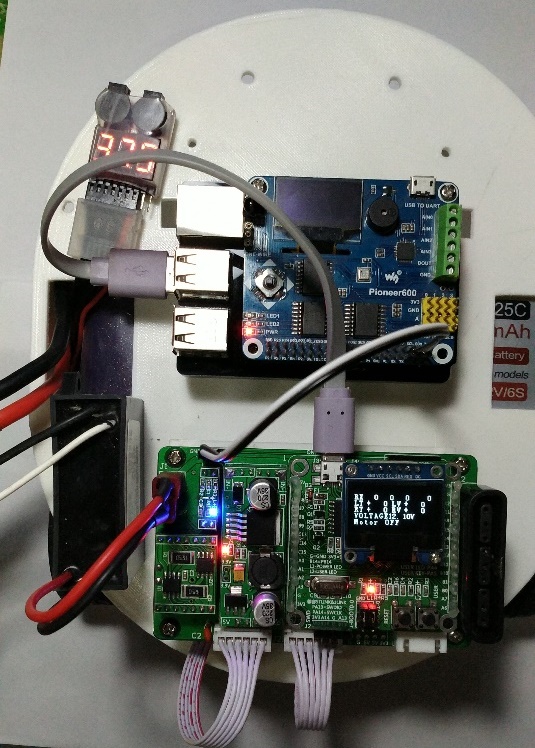


图3-10 树莓派与下位机连接实物图

## 机器人的机械结构

本系统共使用两套硬件MCU和传感器若干，所占空间较大，不便于在简单的万用洞洞板上安装，故根据巡逻机器人的需求设计了机器人外壳，并通过3D打印制作。

机器人采用2轮差速移动控制，外加一个全向轮，三个轮子构成一个三角形，有利于机器人在平面地面站立的更稳固。电池组及控制板等核心部件安装在机器人中心，根据传感器分布需求及安装设备的空间需求，将机器人底盘形状设计成圆形既有良好的结构强度也有赏心悦目的外观。如图3-11所示是巡逻机器人底盘的俯视图。

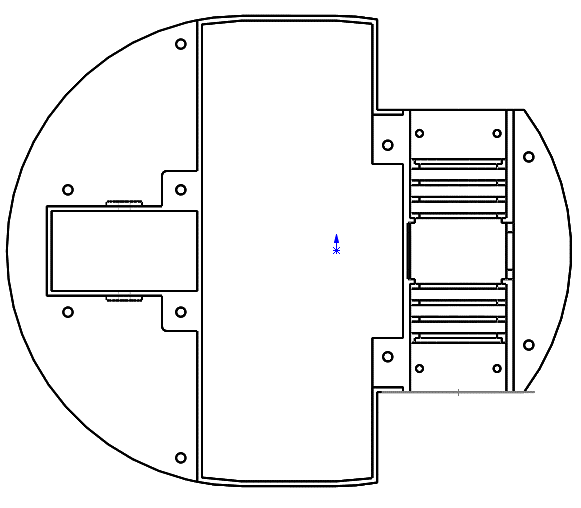


图3-11 机器人底盘俯视图

后轮由两个电机通过联轴器及轴承相连，轮轴向小车底盘长期提供向上的支持力。当后轮电机做功时，轴承还会向底盘传递向前或向后的力。电机与轴承相连，电机会受到向上、向前、向后的作用力，故电机安装应当完全固定。底盘根据电机外形设计一个能正好与电机外壳相吻合的多层支架，分为上下两部分将电机完整地固定起来。如图3-12所示时只安装左后轮固定配件的效果图。

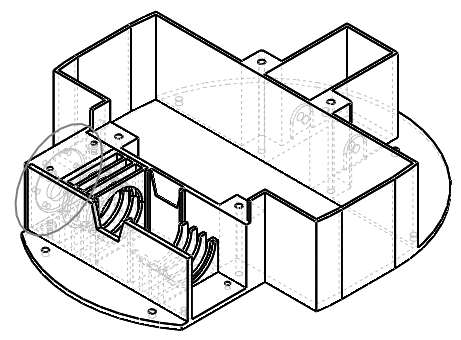


图3-12 后轮电机固定示意图

前轮使用万向轮，不提供动力，为被动轮胎，机器人底盘已经根据选配的万向轮型号预留了前轮安装位，并预留了螺丝孔。使用螺丝和轴承即可将万向轮安装在地盘上。

前轮左右侧空余的空间是预留做安装在机器人前侧的传感器，考虑到机器人的使用场景较多，不同的场景对传感器的需求差异较大，这里不设计具体的传感器安装位，预留固定传感器支架的螺丝孔。机器人外壳设计图见附录一。

# 软件设计

提供设计分为三部分构成，并可以对应到部分硬件上：执行层对应下位机STM32单片机，核心层对应上位机树莓派卡片式计算机，高级层对应远程工作站。相邻的部分之间可以双向通讯。

## 下位机：嵌入式控制主程序

下位机负责控制直流电机转速，使用编码器检测电机转速，使用ADC模块检测电池电压，使用PID算法调节电机转速，从串口接收控制指令，从串口发送里程数据和电压数据。



图4-1 下位机主程序流程图

下位机STM32系统上电后首先初始化所相关I/O口。使用库函数使能所有用到的I/O口的时钟、串口时钟、复用时钟、ADC时钟、定时器时钟，初始化连接两个编码器的I/O口的模式为下拉输入状态，初始化连接电机驱动芯片A4950的I/O口为2MHz的推挽输出模式，初始化电压检测模块的I/O口的模式为模拟输入，初始化串口I/O口的模式为复用推挽输出和浮空输入。

PWM波由定时器产生。初始化TIM4定时器工作在自动重装模式下，定时时长为100us，即频率为10KHz。并将TIM4的CH1~CH4通道设置为PWM1模式，并使能定时器。修改每个通道的时间比较值就可以达到脉冲的占空比。

检测编码器数据，首先初始化TIM2定时器和TIM3定时器，设定其工作在自动重装模式下。编码器每检测到轮轴的旋转一定角度就会产生脉冲信号，脉冲的数量和旋转的角度成正比。当旋转方向不变时，检测一定时间内脉冲的数量，就可以算得该时间内的平均转速。编码器输出两条信号存在四分之一个周期的相位差，根据两信号的相位信息，可以判断编码器的旋转方向。当定时器时间满一个周期，激活中断程序，将计数器中累计的脉冲数送出，并清空计数器，重装定时器。送出的脉冲数即为单位时间内编码器旋转角度的原始检测值。

## 下位机：串口中断服务程序

串口中断负责接收上位机发送过来的指令。初始化好串口的工作模式后，串口数据由串口硬件完成对单字节的接收或发送，上位机向下位机发送的指令需要多个字节组成，第一个字节是命令开头标志“0xFF”，其他字节不能再出现“0xFF”；第二个字节是命令码，不同命令码代表不同的命令；第三、四个字节是命令参数。命令规则如表4-2所示。

表4-2 几个上位机命令码示意表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 起始标志 | 命令码 | 参数1 | 参数2 | 含义 |
| 0xFF | 0xF1 | \* | \* | 启动电机 |
| 0xFF | 0xF2 | \* | \* | 关闭电机 |
| 0xFF | 0xF3 | V\_L | V\_R | 修改左右电机转速 |
| 0xFF | 0xF4 | KP | KI | 修改参数KP，KI |
| 0xFF | 0xF5 | \* | \* | 请求PID参数 |

\* 表示任意内容，单个参数为一个字节

串口中断程序接收累计满4个字节的命令后，将执行命令处理函数。收到“启动电机”命令，则使能TIM4定时器，定时器将按着设定的周期和占空比产生PWM波，控制A4950芯片驱动电机运转。若收到“停止电机”命令，则关闭TIM4定时器的使能，定时器将不在产生PWM波，再用命令将控制电机的四根线都复位，A4950芯片将停止输出，电机掉电停止工作。收到调速命令，中断程序先将传递来的参数进行格式转换，把转换后的设定值写入左右电机速度设定变量中，在此之后进行PID调速运算时将调用该设定值。收到修改PID参数的命令，中断程序将先把传送来的PID参数设定值写入Velocity\_KP和Velocity\_KI变量中，在此之后进行的PID调速计算，将使用修改后的设定值计算。



图4-3 串口中断流程图

## 下位机：编码器及PID控制程序

机器人的电机PID调速程序与编码器定时器结合在一起，每次TIM2定时器和TIM3定时器完成单位时间的编码器采集后，将本单位时间内的转速采集进来，刷新定TIM2和TIM3定时器后，把最新的速度读数传入PID函数计算。

PID函数采用的是增量式PI控制，通过计算增量，将原本积分环节累加运算进行了替换，节省了计算资源和储存空间。如图4-4所示是下位机编码器及PID中断的控制流程图。



图4-4 编码器及PID控制流程图

根据离散的PID公式4-1可以推出n-1时刻的控制量公式4-2

|  |  |
| --- | --- |
|  | （4-1） |
|  | （4-2） |

则可以算得两次控制量的增量是公式4-3

|  |  |
| --- | --- |
|  | （4-3） |

将公式4-1和公式4-2带入得公式4-4

|  |  |
| --- | --- |
|  | （4-4） |

将公式4-4中的积分系数用代替。

将公式4-4中的微分系数用代替，可得公式4-5

|  |  |
| --- | --- |
|  | （4-5） |

本系统使用的是增量式PI调节，使用计算公式4-6

|  |  |
| --- | --- |
|  | （4-6） |

电机转速设定值与编码器检测值之差即为本次偏差量，上次偏差量。

电机转速控制将以电源占空比的形式放入产生PWM波的TIM4定时器的PWM模式的比较值。

PID控制结束后，调用串口发送程序，将这次的编码器检测信息发给上位机。

## 上位机：树莓派系统环境配置

在本系统中，调试树莓派系统是一件非常繁琐的事情，需要考虑软件库的支持、系统内核的版本、以及系统运行的效率。笔者的操作系统使用的是树莓派基金会官方网站提供的基于Debian的RASPBIAN，接入有线网络后使用SSH连接并配置系统。设置固定IP地址，设置自动连接WIFI网络。

软件包方面，串口调试需要用到minicom，串口python软件库pyserial。

ROS环境较复杂，这里参考ROS论坛官方文档[5]的部署过程，将最新版本Lunar版的ROS系统部署在最新的Stretch版RASPBIAN系统中，详细部署过程见附录二。

## 上位机：超声波测距子程序

三个超声波测距传感器硬件功能完全相铜，驱动方式也完全相同，用同样的驱动子程序依次控制三个外设的I/O口即可。

TRIG端给上10uS以上的高电平，超声波发射端将自动发送发送8个40KHz的脉冲，收到回声信号后，传感器将把ECHO线的电平拉高一定时长，时间长度和超声波从发射到接收到的时长相等。如图4-5所示是超声波测距模块控制时序图。

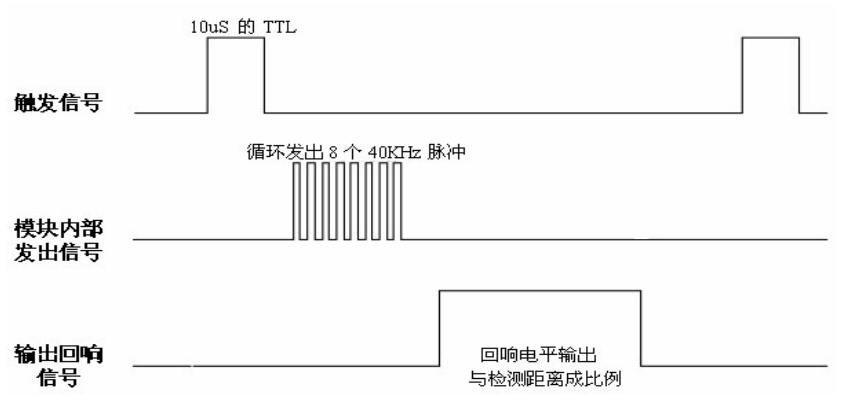


图4-5 超声波时序图

记超声波在空气中的传播速度，ECHO线的回响信号高电平时长，则超声波距离传感器测量的距离如公式4-7。

|  |  |
| --- | --- |
|  | （4-7） |

声速与温度有关系，如公式4-8。

|  |  |
| --- | --- |
|  | （4-8） |

本程序依据ROS节点的格式去构建。程序首先初始化GPIO口，分别设定Echo口为输入模式，Trip口为输出模式，并将Trip口输出低电平。将距离检测程序封装成一个子函数，每次检测结束后，会将处理好的检测值发布到距离话题组中。

距离检测子函数严格按照时序图中的要求。先将Trip口电平拉高，延时15us后复位。然后记录此时刻时间信息为T0，等待Echo口响应信号发生跳变，响应信号只有在收到回声定位后才会跳变。程序等待一定时间，倘若还没收到跳变，将认定本次测距信号丢失，直接给出返回值0并结束结束检测。若发生跳变，立刻记录此时可时间信息T1，以查表的方式检查Echo口的响应信号是否变为低。当检测到Echo口响应信号变为低电平时，说明脉冲信号结束，立刻例句此时刻时间信息T2，用T2和T1相减得脉冲信号的时间长度，即超声波路径的时长，再带入音速计算公式算得距离，将距离信息发布到话题组总。倘若脉冲时长过长，即检测距离过远，根据硬件设计的，超声波行程过远能量损耗过大，超声波接收器将难以检测，此时检测的信号很可能不是由本设备主动发出的声波。为保证传感器的检测精度，这里把超出检测距离的读数统一设置成最大量程的值返回。



图4-6 超声波测距子程序流程图

## 上位机：烟雾检测子程序

烟雾传感器使用的时MQ2型模拟气体传感器，根据传感器手册介绍，该传感器工作时首先会进行自加温，在工作温度下传感器更灵敏也更精确。温度、湿度一定时，空气中的二氧化碳、烷烃类、烟雾颗粒含量越高，传感器的电导越高，则传感器上的电压减小。综上，传感器输出的模拟电压值与烟雾浓度负相关，空气中的烟雾等物质浓度越高则传感器输出电压越低。

烟雾传感器中有一个电位器，可以调节在环境因素不发生改变的情况下模拟输出的电压大小，在本文假定烟雾传感器的电位器已经经过标准调校，电压与烟雾浓度特性不会发生改变。

为检测模拟电压量的输入，上位机外接了一个PCF8591芯片，使用I2C与PCF8591芯片通讯，检测烟雾传感器模拟输入电压。

树莓派上由专用的I2C总线，系统中也有相关驱动，直接调用函数读取即可。如图4-7所示是烟雾浓度检测程序的流程图



图4-7 烟雾检测程序流程图

## 上位机：气压检测子程序

暂时保留，考虑删除。

## 上位机：串口通讯子程序

树莓派与下位机通过串口通讯，程序使用python编写，调用Pyserial库，程序分发送和接收两个独立的部分。接收程序负责从串口接收数据，并将单字节数据组成一个数据帧，按照帧头将数据发布到对应的订阅中；发送程序负责从串口发送数据，它以服务节点的形式存在。

ROS程序包中已经有能把串口数据转发到串口消息订阅的串口桥接程序，这些数据是未封装成帧的单字节数据，如图4-8所示是处理串口消息订阅到其它订阅的处理程序。



图4-8 串口接收流程图

## 上位机：里程计子程序

里程计子程序的里程采集依赖于下位机编码器的数据，编码器每次采集到到的数据通过串口发送给上位机，上位机的串口接收程序接收后会发送给本子程序，经里程计子程序计算生成新的里程坐标及方向向量并将信息发布到里程订阅中。

里程记订阅消息格式为绝对的“坐标向量+方向向量”，每次下位机编码器采集的信息可以转化成相对于小车当前正方向上的相对“位移向量+转向向量”。每次采集左右编码器数据以正负区分方向，分别记作和，单位是毫米。则每次的相对里程的大小。采样频率较高，每次两轮位移之差与轮间距相差一个数量级以上，按照如图4-9所示方式分析车轮移动前后状态，L’R’长度与LR长度的差也可忽略不计。记向右转为旋转正方向，则旋转角度。轮间距经调试设置后作为常量使用，在系统运行时，每次向里程计输入和可获得相对里程和。



图4-9 相对里程示意图

若无特殊设置，里程计初始值为零，即里程坐标ADDR\_X=0，ADDR\_Y=0，方向角度ADDR\_R=0。如图4-10所示，是里程订阅中的绝对里程计与上文计算得到的相对里程的关系。



图4-10 相对里程与绝对里程关系图

若已知A点绝对里程为，本次读入相对里程偏移量。则有：







据此可得里程计递推公式：



## 上位机：模式识别子程序

模式识别用界定巡逻指标是否达到设定的阈值，一旦触发阈值警告，立即停止巡逻，开始收集更多的环境信息，并对警告做出既定的响应处理。



图4-11 模式识别流程图

# 系统测试

系统上电后，树莓派自动上电开机，并连接上了无线网络。通过SSH命令行进入树莓派，使用minicom串口工具向下位机发送运动指令，机器人可以正常移动，初步调试通过。

## PID调试

调试机器人后轮电机PID控制的效果，为方便测量数据，上位机采用有线连接，将机器人底盘悬空架起，做左右轮空转的低负载测试。

运行PID测试程序。程序记录串口返回数据，激活电机使能控制，并开始记录数据。等待一定的延时之后，发送一个控制左右电机转速为50的串口命令，左右轮开始旋转，待左右轮转速稳定后，停止测试程序，并将接收到的二进制数据保存到文件。将二进制数据转码后整理成表，如图5-1所示是转码整理后的小车接收到相当于阶跃信号的控制命令后的转速数据。

图5-1 PID调试测试

图中X轴表示接收到的数据包的编号，Y轴表示接收到的数据的读数。串口数据包的发送频率是72Hz。超调量80%，性能较好。

## 超声波测距传感器测试

将机器人安防在地面固定地点，测试正前方超声波传感器，以超声波传感器发射器铁罩顶端为起点，使用钢卷尺测定距离。移动一个边长50cm见方的纸质盒子作为障碍物，进行距离测试，测试结果如图5-1。

表5-2 超声波测距传感器测试统计图

统计数据已筛除了没有检测结果的数据，测试1的环境是27.8℃；测试2的温度是22.7℃。声速按照340m/s默认值设定，从测试统计结果可以看出，温度对对测距有一定的影响，5℃的温差可领传感器产生0.5%的误差，将温度传感器的数据引入测距程序中计算声速，有助于减小误差。

## 烟雾传感器测试

给出设计系统的实物照片（注意黑白打印要清晰），讲明白测试时的场景，测试结果，及对结果的简单评价，要与第一章的问题相对照，看是否解决了提出的问题。

（要用数据说话，整理数据结果，统计误差规律）

# 结论

再次阐明解决的问题、采用的方法、结果，以及未来的改进计划。

# 参考文献

应以期刊为主

著作：[序号]作者.译者.书名.版本.出版地.出版社.出版时间.引用部分起止页

期刊：[序号]作者.译者.文章题目.期刊名.年份.卷号(期数). 引用部分起止页

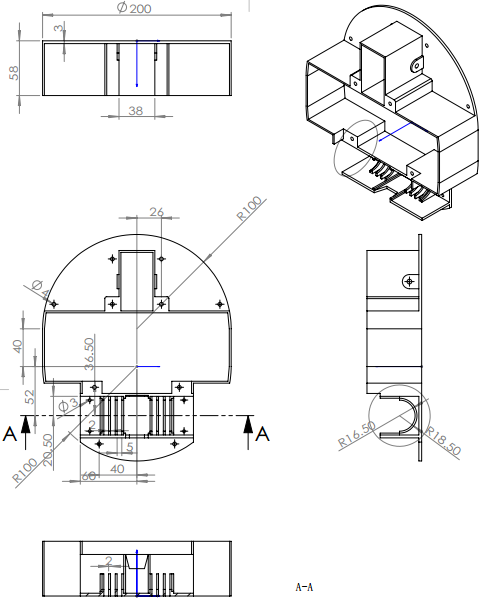
会议论文集：[序号]作者.译者.文章名.文集名 .会址.开会年.出版地.出版者.出版时间.引用部分起止页Installing ROS Indigo on the Raspberry Pi

00

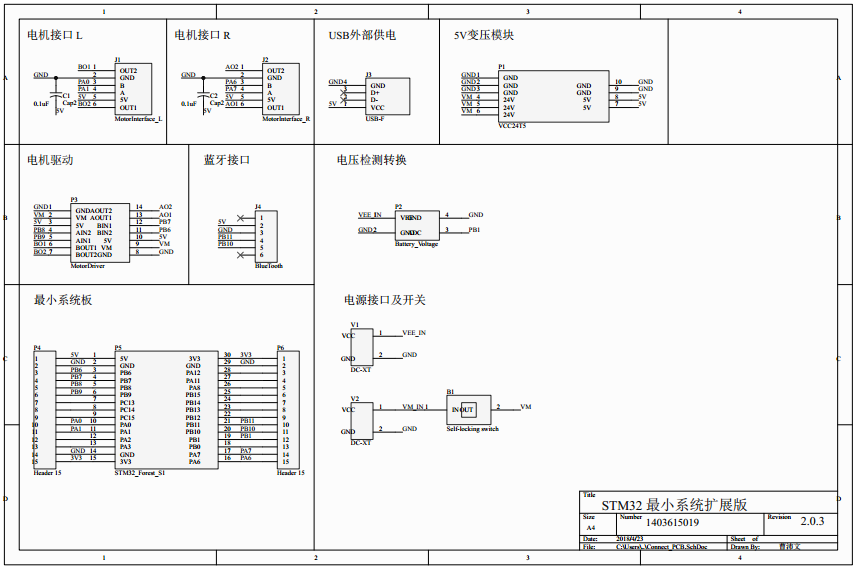
1. 陈志华，谢存禧，曾德怀．巡逻机器人的研究现状与应用前景 [J]．机电工程技术．2003，32 (6) : 19 - 22．
2. 李建勇，刘雪梅，李雪霞，等．基于Ros的开源移动机器人系统设计［J］．机电工程，2017，34( 2) : 205 - 208．
3. <http://www.18650.com.cn/news/15373317.html>
4. Fionn Sheerin．电源模块如何简化和加速产品设计[M]．电子说．2018，4．
5. Installing ROS Indigo on the Raspberry Pi[J/OL ]．Wiki.ros.org．2018-1-19．

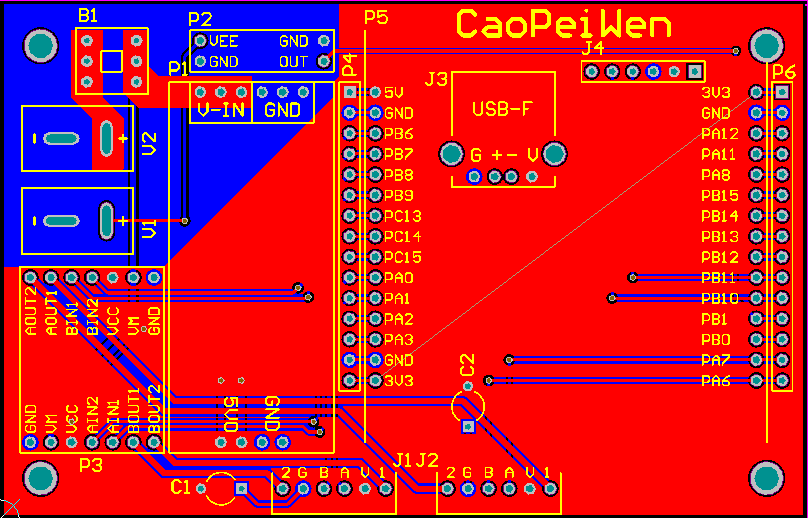
# 附录

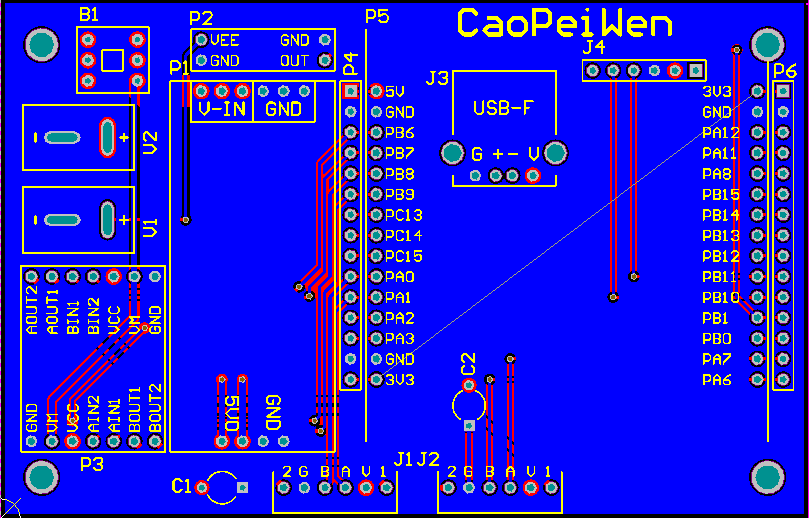
## 小车外壳设计图



## 下位机电路板设计图







## 部署ROS详细流程

安装树莓派基金会提供的Raspbian系统最新版

通过SSH登陆树莓派，以下是在命令行执行的命令。

[注释] 宋体五号

①注释1宋体小五号

②注释2宋体小五号