

青 岛 科 技 大 学

本 科 毕 业 设 计

题 目 家庭智能服务机器人的设计

指导教师 崔玉霞

辅导教师

学生姓名 文岩

学生学号 1505170106

机电工程 学院 机械工程 专业 ALPS151 班

2019 年 6 月 17 日

家庭智能服务机器人的设计

摘 要

一方面随着我国老龄化形势的日益严峻、社会养老服务体系尚不健全、从业人员职业化建设相对落后、没有足够的老年人服务人员等现状,另一方面随着人们对生活品质要求的逐步提高,家庭对于机器人的需求变得越来越迫切。可以预见未来家庭服务机器人不仅仅局限于目前的扫地机器人、教育机器人、语音对话机器人等简单的机器人形式,而是会向类人多功能通用机器人形式发展。机器人之父约瑟夫早就认为家用服务机器人拥有广阔的市场,而机器人进入家庭已是一种必然趋势^[1]。比尔·盖茨就曾预言,未来家家都将有机器人的身影,机器人带来的改变不会化于个人计算机过去30年带来的影响。因此家庭服务机器人的研究具有比较深远的社会价值和意义^[2]。

得益于数学的发展和在工程技术中的深度运用、人工智能技术的快速发展、嵌入式处理器的性能大幅提高,本文实现了一个以机器人SLAM、Navigation和视觉监测为主,多种环境感知传感器、远程执行器件为辅的多功能家庭智能服务机器人的设计。

本文中设计的机器人主要的功能如下:

(1) 对室内环境进行高精度的建图。得益于国产高性能低成本的激光雷达的发展和近几年先进的SLAM算法的提出,在建图功能上本文设计的机器人能以很高的精度和很快的速度完成整个过程;

(2) 在室内环境下进行定位和导航。这是服务机器人实现其他服务的基础和最根本的要求;

(3) 借助NVIDIA Jetson TX2的高计算能力和Kinect摄像头强大的捕获深度图像的能力,实现对人体姿势监测功能,主要目的是为了实现对家庭内老人和孩子的状态监测;

(4) 机器人上搭载了MQ烟雾传感器、天然气传感器、空气质量传感器、光照传感器、温湿度传感器文本转语音模块、esp8266模块,可实现对室内对应环境状况的监测和语音播报,通过安装我们提供的安装在天然气管道、窗户、窗帘、灯具等设备上的配对的远端执行器机构,可以对监测的质量做出反应发出远程控制信号对室内设备进行控制。

关键词: 机器人、SLAM、Navigation、Kinetic、姿势监测、环境感知

HOME INTELLIGENT SERVICE ROBOT DESIGN

ABSTRACT

With the increasingly severe aging situation in our country, the social pension service system is not perfect, the professional construction of practitioners is relatively backward, there are not enough service personnel for the elderly, etc. And with the gradual improvement of people's requirements for the quality of life, the family's demand for robots has become urgent. It can be predicted that the future home service robot will develop to the humanoid multi-functional general robot form. Joseph, the father of robots, has long believed that home service robots have a broad market, and it is an inevitable trend for robots to enter the home [1]. Bill Gates has predicted that robots will be in every home in the future, and that the changes brought about by robots will not be the same as those brought about by personal computers in the past 30 years.

Thanks to the development of mathematics, embedded processor performance is greatly increased, this paper implements a predominantly robot SLAM, Navigation and visual monitoring, relies on a variety of environmental perception sensors, remote execution device of multifunctional design of family intelligent service robot.

The main functions of the robot designed in this paper are as follows:

(1) high-precision mapping of indoor environment. Thanks to the development of domestic high-performance and low-cost lidar and the advanced SLAM algorithm in recent years, the robot designed in this paper can complete the whole process with high accuracy and fast speed.

(2) positioning and navigation in indoor environment. This is the basic and fundamental requirement for service robot to realize other services.

(3) The posture monitoring function of the human body. The main purpose is to realize the status monitoring of the elderly and children in the family.

(4) Carrying on the robot lots of variety sensors, which can realize to indoor speech, corresponding to the state of the environment monitoring and installed by our in natural gas pipeline, Windows, curtains, lamps and other equipment on the matching of the remote actuators, can respond to monitoring the quality of the signal remote control to control the indoor equipment.

KEY WORDS: robot, SLAM, Navigation, Kinetic, gesture monitoring, environmental perception

目 录

1 绪论	1
1.1 选题背景与来源	1
1.2 本文主要研究内容	1
1.3 本章小结	2
2 家庭智能服务机器人的总体设计	3
3 家庭智能服务机器人自主导航功能的硬件设计	5
3.1 自主导航功能下位机硬件设计	5
3.1.1 机器人底盘设计	5
3.1.2 轮胎、编码器和电机选型	6
3.1.3 电机驱动选型	8
3.1.4 下位机主控选型	9
3.2 上位机硬件设计	11
3.2.1 上位机主控选型	11
3.2.2 SLAM 和 Navigation 传感器选型	12
3.2.3 视觉监护传感器选型	14
4 家庭智能服务机器人自主导航功能的软件设计	19
4.1 家庭智能服务机器人自主导航功能的下位机软件设计	19
4.1.1 TB6612FNG 驱动程序设计	19
4.1.2 编码器程序设计	20
4.1.3 pid 程序设计	22
4.1.4 下位机通讯程序设计	23
4.2 家庭智能服务机器人自主导航功能的上位机程序设计	24
4.2.1 上位机与下位机通讯程序设计	24
4.2.2 SLAM 算法实现	25
4.2.3 Navigation 功能的实现	32
5 姿势检测和环境感知系统设计	42
5.1 姿势识别	42
5.2 环境感知系统	43
5.2.1 烟雾传感器硬件及软件设计	43
5.2.2 光照传感器的硬件及软件设计	45
5.2.3 温湿度传感器硬件及软件设计	46

5.2.4 语音模块硬件及软件设计.....	48
5.2.5 esp8266 模块硬件和软件设计.....	50
5.2.6 远端执行器硬件及软件设计.....	51
6 结论与展望.....	52
参考文献.....	53
致谢.....	57

1 绪论

1.1 选题背景与来源

(1) 针对中国老龄化形势的日益严峻，人们生活工作节奏快，空巢老人问题严重等社会现状，一款能实现对老人监护、帮助老人对室内环境感知并作出反应的设备具有重要的意义和现实作用；

(2) 目前我国国家主要以天然气作为家庭供气，但是目前人们工作节奏快，很多家庭会只留孩子或老人在家，对于使用天然气燃气灶来说是一个潜在的安全隐患，所以对于室内天然气的监控和管道控制可避免发生危险事故；

(3) 随着生活水平的提高和社会的发展，人们普遍追求高品质的生活质量，比如实现灯光亮度随着环境自动调节、下雨天自动关窗户、陪人聊天、监测室内环境质量等方面都是家庭服务机器人标配的功能和比较活跃的方向；

(4) SLAM 和 Navigation 等相关技术是未来实现真正意义上的智能服务机器人的基础和必不可少的基础功能，但同时又是技术含量非常高、应用起来非常困难的领域，所以有很多科学家、工程师活跃在这个领域。SLAM、Navigation 大多以 ROS 系统为依托，每年的 IROS 大会都会有很多的优秀论文被发表出来，有力的推动了这个行业的发展。同时，这几项技术也是自动驾驶相关的技术，未来的潜力不可估量。

(5) 在国际环境方面，国内的产品大多数集中在教育机器人、语音机器人、安保机器人（这个地方要加对应的公司）等方面，相对来说技术含量较低，服务功能较为单一等；日本已经有 Pepper 机器人、SpotMini 机器人（波士顿动力被软银集团收购）、NAO 机器人（Aldebaran Robotics 公司被软银集团收购）、ASIMO 机器人等明星产品。（这个地方确定一下收购是否意味着归属）中国虽然在智能服务机器人方面有很多成就，但发展空间仍旧非常大。

(6) 上面讲到了研究服务机器人的必要性，从可行性方面来说，得益于人工智能技术的发展和高性能嵌入式平台的发展（如英伟达的 Jetson 系列产品），使我们很容易的进行高数据量的处理，目前在嵌入式平台上实现视觉信息的处理、激光信息的处理、图灵语音等都游刃有余且功耗极小。

1.2 本文主要研究内容

主要的研究内容我分成下面几块来讲：

下位机的实现。下位机以 Arduino Mega 2560 作为主控，研究了控制电机、获取编码器数据、与上位机通信等内容；

上位机的实现。上位机的设计内容较多，主要有建图、导航和视觉监护等，主要研究了不同 SLAM 算法之间的差别和效果，导航精度和速度的代码结构、参数设置和在 ROS

系统的实现，利用 Kinect 实现骨骼追踪进而判断人体姿势，与下位机通信等；

环境感知的实现。主要研究了环境感知传感器的特性曲线和数学公式提取，多个传感器之间同步稳定工作的代码实现。

1.3 本章小结

第 1 章主要描述了当前与服务机器人对应的社会环境，进而可以清晰的展现出发展服务机器人的可行性和必要性；描述了本文设计机器人的主要研究内容。

2 家庭智能服务机器人的总体设计

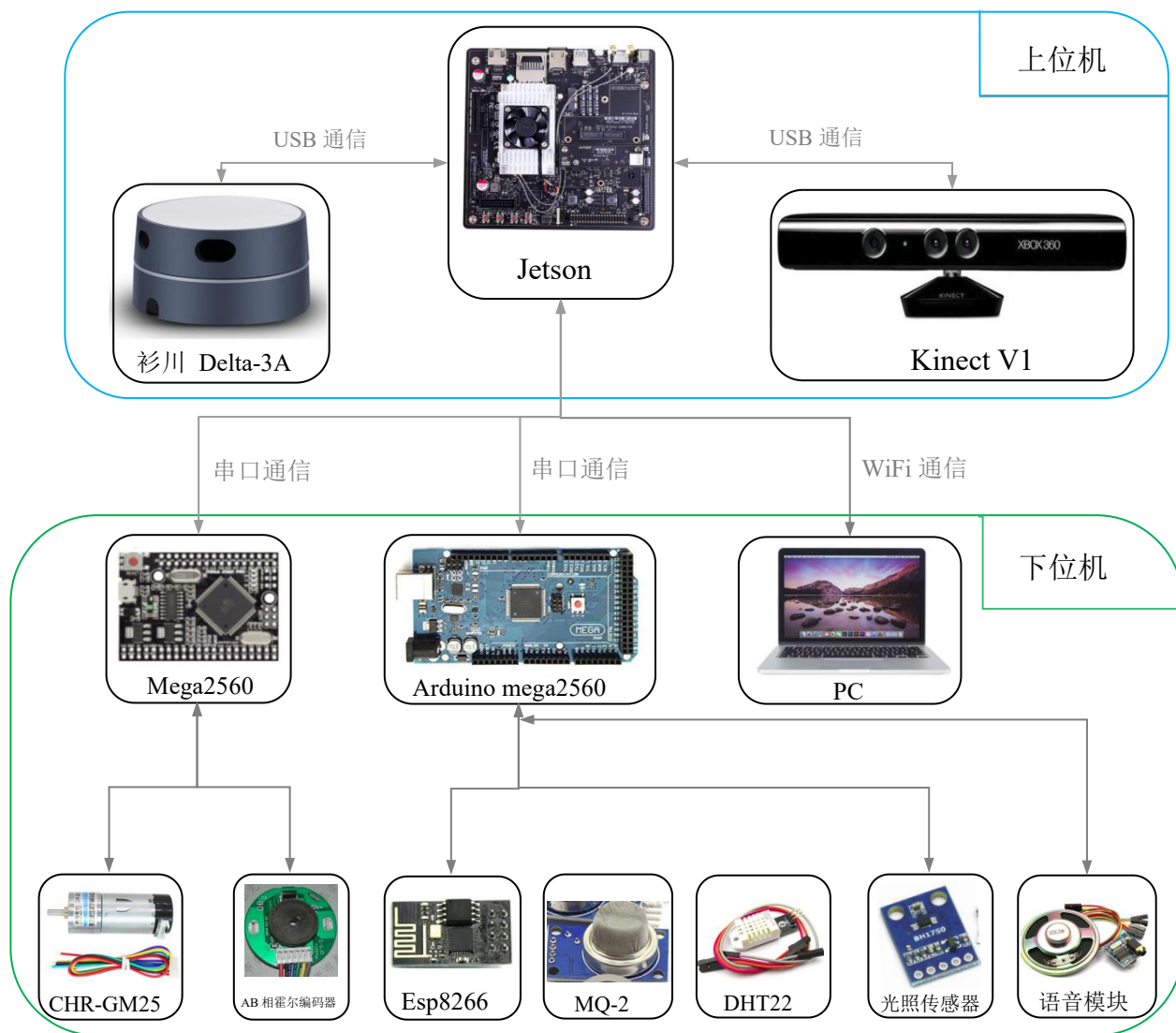


图 2-1 家庭智能服务机器人的系统框架

Fig.2-1 The system framework of home intelligent service robot

本文所设计机器人的整体结构如图 2-1 所示。TX2 作为整个系统的“大脑”，负责处理激光雷达的数据和 Kinect 摄像头的的数据，同时担负着和下位机进行通信的任务。下位机 1 处理编码器数据和生成电机控制指令，下位机 2 负责处理所有环境感知传感器。下位机 1 与上位机之间的通信通过 `ros_arduino_bridge` 包来完成，双方协定了一组“数据头”，TX2 通过发送不同的“数据头”来指示下位机 1 执行不同的操作。下位机 2 和其控制的环境传感器其实是相对独立的一个整体，TX2 对它们没有控制任务，它们也不负责向 TX2 传送数据。

下位机中的 esp8266 作为 server，向安装在室内的多个 station 进行通信，比如安装在天然气管道上的末端执行器、安装在窗户上的执行器、安装在电灯上的执行器等，它们之间通过 WiFi 信号组网，有 Arduino Mega2560 统一控制指令。

上位机中的语音模块负责播报环境感知传感器接受到的数据进行处理后的最终的环境信息。

PC 负责查看 SLAM、Navigation 的效果、调试机器人和发出人为控制指令都任务。

3 家庭智能服务机器人自主导航功能的硬件设计

3.1 自主导航功能下位机硬件设计

3.1.1 机器人底盘设计

考虑到室内环境下使用，机身尺寸不宜太大，也不能太小。同时，在设计之初，就预料到会有非常多的电子元器件和线材需要安装在机器人身上，所以封闭式的机身固然好看，但是封闭式的结构的加工主要有三种途径：1.开模具；2.3D 打印；3.找类似能用的已有产品的外壳代用。开模具的成本无法估计而且时间长，3D 打印外壳不光滑而且速度慢，找已有产品外壳代替后期的设计会非常被动，可能很多地方都会被原有的结构设计打乱现有的设计。同时考虑到作为处女座初试同时毕设时间又非常紧迫，决定以开放式的堆叠结构作为主要结构。

接下来就是确定行进驱动的问题。首先双足或多足形式步态算法复杂且稳定性很难做的很好，而且噪音大功耗高，不在考虑之列；其次麦克纳姆轮和瑞士轮由于价格较高且完全没必要使用，不在考虑之列；因为普通橡胶轮价格便宜而且完全满足使用要求，所以选择橡胶轮。

橡胶轮底盘主要有四轮底盘和三轮底盘。四轮底盘稳定性好，三轮底盘稳定性差但转向灵活、控制方便。首先我们来看一下四轮底盘的几种不同的形式。

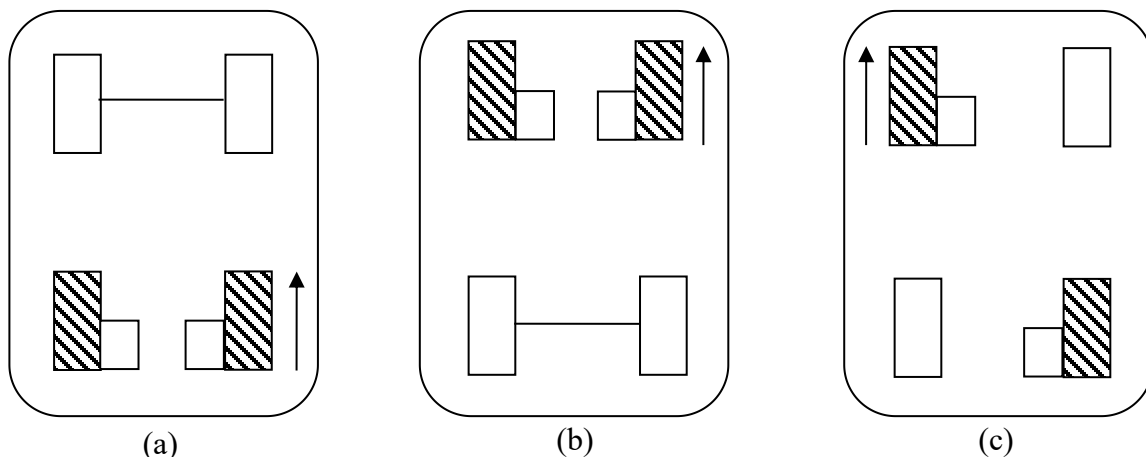


图 3-1 四轮底盘设计方案

Fig.3-1 Four-wheel chassis design scheme

四轮布局常见形式如图 3-1 所示，后轮驱动形式如（a），前轮驱动形式如（b），左右轮对角驱动形式如（c）。对于（b）所示驱动形式，由于驱动时前轮既要负责驱动车辆又要负责车辆的转向，因此前轴负荷比较重，这使得前轮驱动的底盘会在过弯时前部重心因惯性而发生前移，容易突破前轮的抓地力，而后轮缺乏动力，咋会发生转向不足^[3]；对于

(a) 所示，驱动和转向分别由不同的轮子负责，稳定性更好，缺点就是转弯半径过大，在小区域范围转弯不灵活等；对于 (c) 所示驱动形式，车轮必须全部采用全向轮，增加了成本，但使底盘具有横向移动的能力^[3]。

我们再来看一下三轮驱动底盘的常见形式：

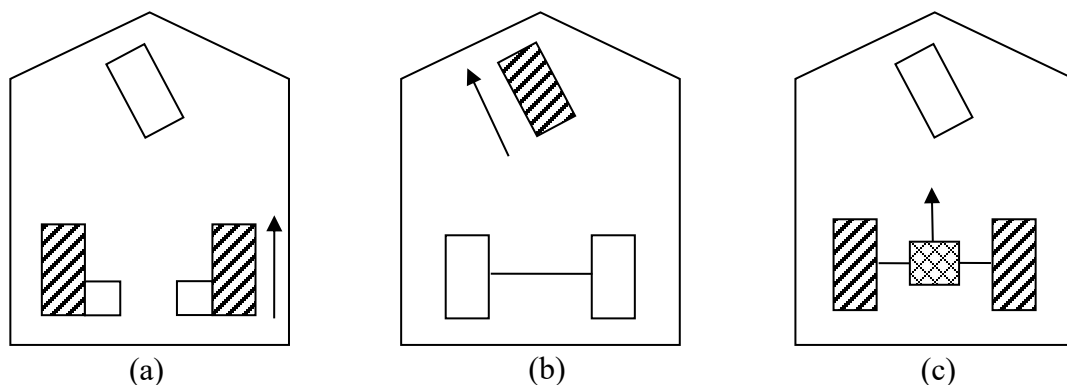


图 3-2 三轮底盘的设计方案

Fig.3-2 Design of three-wheel chassis

三轮驱动的底盘形式主要有以上三种：独立两轮驱动如 (a)；前轮转向兼驱动如 (b)；前轮转向后轮差速驱动如 (c)^[3]。三轮底盘由于与地面的接触点个数比四轮底盘少，所以稳定性较差，在转弯过程中，容易侧翻。(a) 所示形式具有较好的转弯灵活性，可在原地直接进行转圈，而且差分轮控制简单，缺点就是在承受较大侧向加速度、路面不平坦时容易侧翻；(b) 所示形式并不是一种优美的驱动形式，如果重心靠后，前轮由于受力小摩擦力容易不够，如果重心靠前或者整体车身较重的情况下，前轮的转向摩擦力大，而且在受力不平衡时非常容易侧翻；(c) 所示形式最少需要驱动两个电机，转向轮需要一个电机驱动，差速器需要一个电机驱动，但是由于差速器和齿轮、转轴之间的能量消耗，使得整体的效率低下，很大一部分能量被白白浪费掉，同时相对于差分式底盘，两个电机的控制形式完全不一样，所以需要两套代码来分别控制，增加了代码复杂度。

分析我们的使用场景，在平坦的室内环境下使用，且加速度和速度都较小，对车身产生的侧向冲击力很小，同时要求有灵活的转向能力，功耗小。综上，只有三轮形式两轮差分驱动最符合要求。

3.1.2 轮胎、编码器和电机选型

根据在室内使用要求，我们暂定机器人的整体尺寸在 50x50x50 之内，为了保持机身的比例协调性和配件市场的供货情况，我们同样暂定了一款橡胶轮胎，如图 3-3 所示：



图 3-3 轮子
Fig.3-3 wheel

表 3-1 轮子参数
Tab.3-1 Wheel parameters

参数	值
直径	68mm
胎宽	27mm
自重	34g

现在我们暂定了小车轮胎的直径。因为在室内使用，暂且假设车体前进速度不会超过 0.5m/s,由电机转速的计算公式（3-1）得

$$n = \frac{v}{\pi d} = \frac{0.5}{0.068\pi} = 2.341rps = 140.431rpm \quad (3-1)$$

所以此值为电机转速的一个参考值。

根据推测本文所设计机器人的整机重量不会大于 5KG。橡胶轮与大理石地面间的滚动摩擦系数大概为 0.2 左右，在机器人的假设最高移动速度 1m/s 的情况下，滚动摩擦力计算式为（3-2）：

$$M = Fn \times \sigma = m \times g \times \sigma = 5 \times 9.8 \times 0.2 = 9.8N \quad (3-2)$$

只要保证整车的牵引力大于滚动摩擦力，即：

$$F_{\text{牵}} \geq M \quad (3-3)$$

取 $F_{\text{牵}} = M$ ，所以电机驱动功率的最小值计算如式（3-4）：

$$P = F_{\text{牵}} / v = 9.8 / 0.5 = 19.6W \quad (3-4)$$

所以每个电机功率要不小于 9.8W。

根据以上计算结果，我们选择驰海电机有限公司的 CHR-GM25-370 电机，如图 3-4 所示：



图 3-4 电机

Fig.3-4 Motor

表 3-2 电机参数
Tab.3-2 Motor parameters

参数	值
额定电压	12V
减速比	1: 45
空载转速	140rpm
额定转矩	4.3Kg.cm
额定电流	$\leq 0.8A$
编码器	AB 相霍尔编 码器
编码线数	11

电机附带的编码器为 AB 双相霍尔编码器，减速后编码线数是 493.9 线。

根据以上的配件选型结果，我设计了一款如图 3-5 的机器人外壳用以承载各个部件。

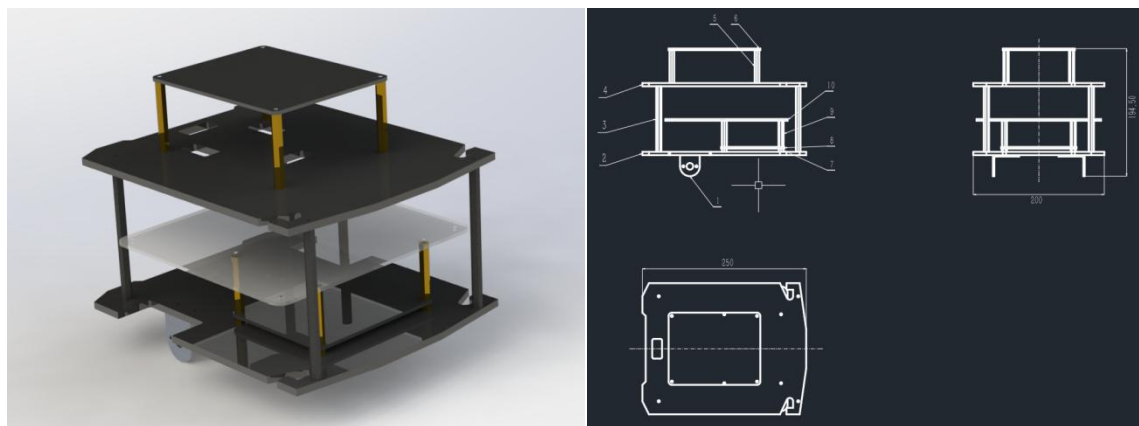


图 3-5 机器人底盘

Fig.3-5 Robot chassis

3.1.3 电机驱动选型

对于直流电机的驱动器，常用的驱动器型号有 L298n 和 TB6612FNG。

L298n 是 ST 公司生产的一种高电压、大电流电机驱动芯片，采用 15 脚封装。主要特点是：工作电压高，最高工作电压可达 46V；输出电流大、瞬间峰值电流可达 3A，持续工作电流为 2A；额定功率 25W，内含两个 H 桥的高电压大电流全桥式驱动器，可以用来驱动直流电动机和步进电动机。TB6612FNG 是东芝半导体公司生产的一款直流电机驱动器件，它具有大电流 MOSFET-H 桥结构，双通道电路输出，可同时驱动 2 个电机。

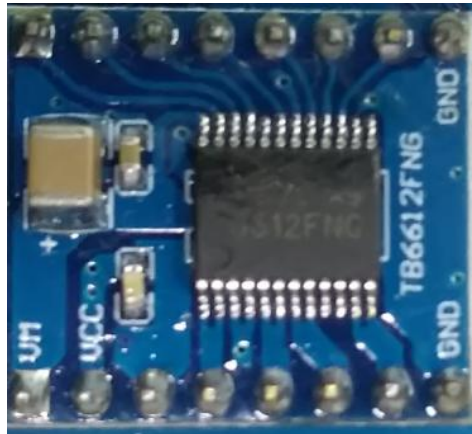


图 3-6 TB6612FNG

Fig.3-6 TB6612FNG

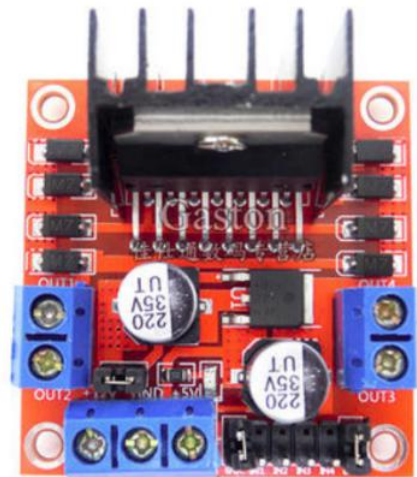


图 3-7 L298n

Fig.3-7 L298n

表 3-3 TB6612FNG 和 L298n 参数对比

Tab.3-3 Parameter comparison between TB6612FNG and L298n

	TB6612FNG	L298n
H 桥结构	MOSFET H 桥电路	晶体管 H 桥电路
外围尺寸	22*19*1.5mm	43*43*27mm
工作温度	-20~85℃	-20~85℃
PWM 支持频率	100kHz	40kHz
最大输出电流	2.4A	2A
公司	TOSHIBA	ST

TB6612FNG 是基于 MOSFET 的 H 桥集成电路，其效率高于晶体管 H 桥驱动器。相比 L298N 的热耗性和外围二极管续流电路，它无需外加散热片，外围电路简单，只需外接电源滤波电容就可以直接驱动电机，利于减小系统尺寸。对于 PWM 信号，它支持高达 100 kHz 的频率，相对于 L298n 的 40 kHz 也具有较大优势。

所以我们最终选择 TB6612FNG 作为我们的电机驱动模块。

3.1.4 下位机主控选型

下位机主控作为下位机部分的总司令，对它的选择无疑需要多加考虑和对比。不仅从性能方面考虑，易用性、开发便捷性、稳定性、社区活跃度等方面都要做综合考虑。

常见的主控有 STM32、STM8、Arduino、STC51、飞思卡尔等。飞思卡尔性价比很低，

价格昂贵、性能一般，故不在考虑之列；STC51 引脚少、主频小、实现复杂功能编码耗时，故也舍弃；STM8 国内资料少，社区活跃度低，且价格比 STM32 便宜很少，故也舍去；我经常用的就只有 STM32 和 Arduino 两款。

STM32 系列单片机型号丰富，目前已经出到 f7 系列，但价格昂贵。现在国内用的较多、资料较多的有 f4 系列和 f1 系列。f407 系列的主频已经达到了 168MHz，其上系列的主频更是达到 200 多兆赫兹。下位机主控的主要功能是获取编码器数据和计算电机控制信号，没有涉及到复杂的控制算法，没必要使用如此高的主频。Arduino 系列国内有常用的 Arduino uno 和 Arduino mega2560 两款，因 uno 的引脚太少，后期添加功能比较受限，所以舍弃掉 uno 的选择意向。下面我将对比一下我常用的两款具体型号的单片机的，如表 3-4：

表 3-4 不同下位机主控对比

Fig.3-4 Comparison of master control of different subcomputers

	STM32F103VET6	Arduino mega2560
内核	Cortex-M3 32bit RISC	AVR 8bit RISC
接口资源	3-SPI,5-USART,2-I2C,2-I2S,USB,1-CAN,1-FSMC	4-UART,5-SPI,1-I2C,16-PWM
工作电压	2-3.6V	4.5-5.5V
工作频率	72MHz	16MHz
SRAM	64KB	8KB
FLASH	512KB	256KB
引脚数	100	100

从上表的对比结果看，Arduino mega2560 完败，但是本文所设计的机器人还是采用了 mega2560 而不是 STM32，原因如下：

Arduino 16MHz 的频率已经足够使用。从 Atmega2560 的数据手册中可以得到，当工作在 16MHz 主频时，器件的吞吐量达到了 16MIPS(Million Instructions Per Second)。16MIPS(Million Instructions Per Second)代表该处理器每秒可以执行 1600 万条机器语言指令，对于本文所设计的下位机的硬件要求来说，完全够用；

Arduino 的开发更简单，相比 STM32，可以不用编写各外设的驱动配置文件；

因为我们上位机要完成的功能需要在 ubuntu 系统下执行，下位机是通过 USB 串口向上位机的 TX2 发送数据，下位机程序的烧写也是通过同一根 USB 数据线由上位机下载到下位机。Arduino IDE 可以完美的在 ubuntu 系统上运行，所以给代码下载、调试带来了非常大的便利。在 Windows 上开发 STM32 可以直接使用 ST 官方的 uKeil 软件，但是这款软件并不支持 linux 平台，所以在 ubuntu 上进行开发 STM32 需要用 Eclipse 进行很多配置才能完成。

至此，下位机的全部硬件的设计工作已经完成。已经完成的下位机硬件和底盘结构如图 3-8 所示：



图 3-8 完成的机器人底盘
Fig.3-8 Finished robot chassis

3.2 上位机硬件设计

3.2.1 上位机主控选型

我们要完成建图、导航、视觉监护功能，单片机完全束手无力了，这时候需要用高性能嵌入式主机来完成这些工作。常用的嵌入式主机有树莓派、Firefly RK3399、Nvidia Jetson TX2、Nano、Intel NUC 等，我们将选几个特定型号对它们与机器人开发相关的参数进行一下比较分析：

表 3-5 不同上位机主控对比

Tab.3-5 Comparison of different host computer controls

	RK3399	Jetson TX2	Jetson Nano	Intel NUC
CPU	Rockchip RK3399	HMP Dual Denver-2 +Quad ARM A57-4	四核 ARM Cortex A57	酷睿 i7-6770HQ
GPU	Mali-T860 四核	NVIDIA Pascal, 256 CUDA cores	128 Maxwell cores	锐炬 Pro
存储	16 GBeMMC	32 GB eMMC SDIO,SATA	TF 卡	SATA、AHCI、 SDXC
内存	2GB DDR3	8GB128bit LPDDR4	4 GB 64bit LPDDR4	32 GB DDR4
USB	USB2.0-4 USB3.0-2	USB3.0-1USB2.0-1	USB3.0-4 USB2.0-1	USB3.1-1 USB3.0-6 USB2.0-2
功耗	7 W	≤7.5 W	5 W	15 W

从性能方面来分析, Intel NUC 和 TX2 略高一筹, 而 TX2 相比 NUC 的优势就是强大的 Pascal 架构的 CUDA 核心, 对于人工智能、深度学习的并行计算效率非常高。在存储方面, 板载存储方面 TX2 优势很大, 对于嵌入式场景来说, 32GB 完全可行。TX2 8GB 的运行内存更是非常富足, 在处理人工智能算法时不用担心因为内存问题牵扯算法的执行。说到嵌入式主机, 不得不谈的一个方面就是功耗。NUC 平均 15W 左右的功耗, 在处理器高负荷运转的时候甚至能达到 65W, 对于如此高的功耗不适合用在嵌入式、需要电池供电的场合。Firefly RK3399 性价比非常高, 但是如果要运行姿态视觉识别, 包括用深度学习进行视觉处理, 16G 的存储空间、2G 的运行内存、GPU 性能都是瓶颈。Jetson Nano 虽然 GPU 性能相对来说比较强大, 但是四核 A57 的 CPU 性能一般, 个人在使用 TX2 的时候, 2 个丹佛核心的存在比只是四核 A57 在性能上要强大很多, 而且 Class 10 的 TF 卡的存取速度也要比 eMMC 慢不少。经过多个参数的权衡, 最后选定 TX2 作为机器人的上位机主控。TX2 的缺点就是 USB 接口太少, 不过好在可以外接 USB Hub。

3.2.2 SLAM 和 Navigation 传感器选型

在实现建图的功能上, 目前主要有激光 SLAM 和视觉 SLAM。激光 SLAM 就是用激光雷达作为 SLAM 的数据源, 传给 SLAM 算法的数据是激光雷达获取的线激光数据; 视觉 SLAM 是用单目摄像头、双目摄像头、鱼眼摄像头、RGBD 摄像头等作为传感器, 传感器种类很多, 传入 SLAM 算法的数据是线激光数据, 所以视觉 SLAM 需要将摄像机的 3D 点云数据降维成激光雷达类型的数据, 这是一个高计算量的过程。激光 SLAM 较为成熟, 视觉 SLAM 包含丰富的信息, 两者各有千秋, 本文设计的机器人选择激光 SLAM, 原因可看表 3-6、图 3-9 和图 3-10:

表 3-6 激光 SLAM 和视觉 SLAM 优缺点

Tab.3-6 Laser SLAM and visual SLAM have advantages and disadvantages

	优势	劣势
激光 SLAM	可靠性高、技术相对成熟建图精度高、可 360° 范围内进行探测	安装有结构要求、成本高、数据种类单一、没有语义信息
视觉 SLAM	结构简单、传感器种类丰富价格相对较低、可提取语义信息, 这将是未来 SLAM 的大潜力方向	环境光影响大、非深度相机暗处无法工作、无纹理区域效果差、计算量大、噪点和误差大

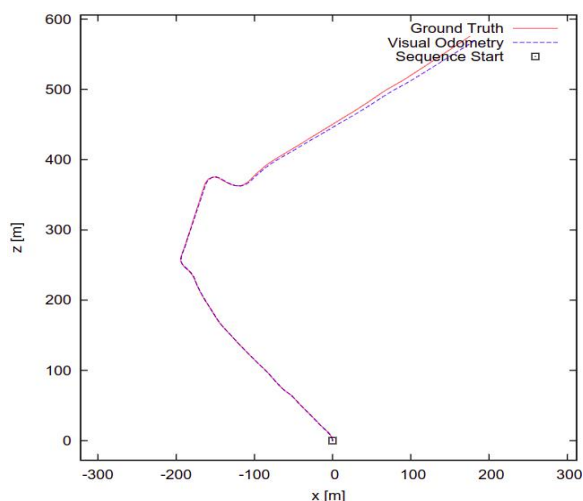


图 3-9 ORB-SLAM-摄像头

Fig.3-9 The ORB - SLAM - camera

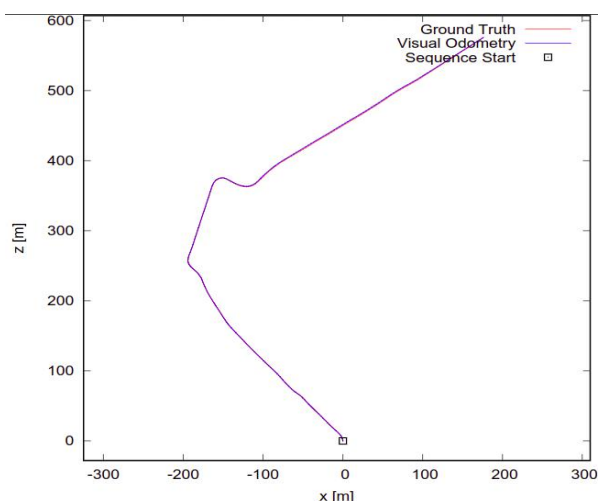


图 3-10 ORB-SLAM 激光校正

Fig.3-10 ORB-SLAM laser correction

图 3-9 是单纯视觉 SLAM 视觉里程计的效果，可以看到在尾端视觉里程计的结果已经与真实场景发生了偏离；图 3-10 是加入激光雷达校正的结果，在尾端里程计的结果依然与真实场景保持一致。

激光 SLAM 对于构建一个性能可靠、效果优良的 SLAM 算法相对容易一些，视觉 SLAM 具有很大的潜力，意味着更容易有新的突破。鉴于本文所设计机器人的目的是制作一个稳定、可靠、质量高的 SLAM 系统，所以选择激光 SLAM 方案。另外在 Navigation 中，因为视觉传感器只能获取一个范围内的数据，所以容易造成机器人在导航过程中旋转寻找特征点，影响整体的导航效果。

激光雷达我们使用的深圳衫川机器人公司生产的 Delta-3A 型激光雷达，如图 3-11：



图 3-11 Delta-3A

Fig.3-11 Delta-3A

表 3-7 Delta-3A 参数
Tab.3-7 Delta - 3 a parameter

参数	值
量程	0.15m~16m
采样率	8k/s
扫描频率	5~15Hz
激光波长	780nm
激光功率	3mW（最大功率）
测量精度	<2%@16m
测量分辨率	1mm
角度分辨率	0.25~0.7°
额定功耗	3W
工作电压	DC 5V
启动电流	1.9A

3.2.3 视觉监护传感器选型

视觉传感器上文已经说过，主要有单目、双目（也可以算是深度相机，属于被动式深度相机）、鱼眼、主动式 RGBD 摄像机。单目相机的优点是功耗低、成本低，比较明显的缺点就是不能直接测量深度信息，必须通过多帧图像之间的变化关系得出深度信息。双目相机（指纯双目相机）可以直接测试深度信息，但是不能用于黑暗处的测量，属于被动测量型传感器。因为 RGBD 相机除了能获取清晰的 RGB 图像外，还能通过主动测量的方式获得深度信息，这得益于相机的工作原理是通过发射和接收红外光来实现深度信息的测量，所以自然可以在黑暗环境下工作。除此之外，视觉 SLAM 方案中使用较多的也是 RGBD 摄像机，所以以后可以扩展 SLAM 的功能和实现方式，作为一个继续研究的平台。所以本文设计机器人使用 RGBD 摄像机。

我们先来看一下几种主流的不同 RGBD 方案的对比：

表 3-8 不同 RGBD 方案的对比
Tab.3-8 Comparison of different RGBD scheme

方案	双目（被动式）	结构光	TOF
典型设备		Kinect v1	Kinect v2
基础原理	双目匹配，三角测量原理	激光散斑编码	反射时差
分辨率	中高	中	低
精度	中	中高	中
帧率	低	中	高
抗光照	高	低	中
硬件成本	低	中	高
算法开发难度	高	中	低
内外参标定	需要	需要	不需要
黑暗环境	不能使用	可以使用	可以使用
室外环境	可以使用	不能使用	不能使用

双目跟结构光一样，都是使用三角测量法根据物体匹配点的视差反算物体距离，只是双目是用自然光，而结构光是用主动光发射特定图案的条纹或散斑。ToF（Time-of-Flight）是通过给目标连续发送光脉冲，然后用传感器接收从物体返回的光，通过探测光脉冲的飞行（往返）时间来得到目标物距离，iphone x 的人脸识别模块也是使用的 ToF 原理。

市面上 RGBD 相机有很多款，主要有微软的 Kinect V1、Kinect V2、Intel RealSense、华硕 xtion 等。2018 年 1 月 19 日，英特尔带来两款新产品：RealSense D415 和 D435，目前市面上的主流产品，Kinect 两款产品已经停止生产，但是市面上还能买到；华硕 xtion 反向一般，没有特色的功能；Intel RealSense 目前还在持续更新，而且功能最强、SDK 封装完整的一系列产品，而且支持 Linux、Windows、Mac OS 等平台，体积相比 Kinect 来说要小了很多，综合评价 Intel RealSense 是性价比最高的一款。

但是鉴于实验室有一台 Kinect v1 和 v2，而且国内关于 Kinect 的资料相对丰富很多，同时 Kinect 实现姿势检测和 SLAM 都有很成功的经验可以学习。

Kinect v1 和 v2 都是非常成功的产品，也给开发者带来了非常大的便利。V2 相比 v1 在功能上丰富了很多，但是也多了部分局限，比如只能使用 USB3.0 接口、更加依赖 Windows 平台等，表 3-9 有个详细的对比：

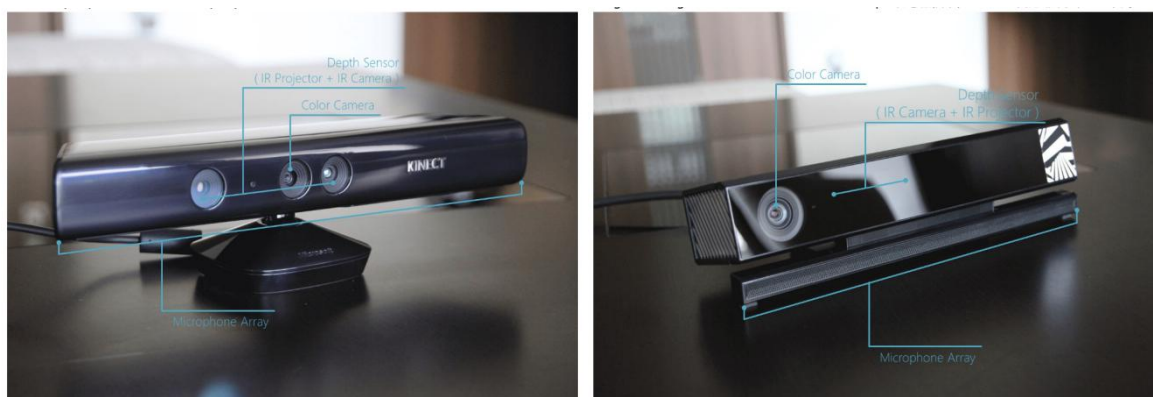


图 3-12 Kinect

Fig.3-12 Kinect

表 3-9 Kinect v1 和 v2 对比

Tab.3-9 Kinect v1 versus v2

	Kinect v1	Kinect v2
颜色分辨率	640x480	1920x1080
颜色 Fps	30	30
深度分辨率	320x240	512x424
深度 Fps	30	30
人物数量	6	6
人物姿势	2	6
关节	20 个/人	25 个/人
检测范围	0.8~4.0m	0.5~4.5m
转动马达	有	无
接口类型	USB2.0	USB3.0
SDK	官方 SDK、OpenNI、Libfreenect 官方 SDK、OpenNI2、libfreenect2	

V2 比 v1 不论在颜色还是深度分辨率上都要高出许多。v1 的要求是 USB2.0 理论传输速率是 60MB/s, v2 是 USB3.0 理论传输速率是 500MB/s。可以计算一下, 以 XRGB Color 为例, 30fps, 那么每秒所需传输的数据大小为 $640 \times 480 \times 4 \times 30$ 约为 35M; 再加上 USHORT 格式的 Depth Color, 30fps, 大小为 $320 \times 240 \times 2 \times 30$ 约为 4M。总计约为 40MB/s, 因为带宽有限, 所以在保证画面帧率稳定的情况下, 分辨率只能如此, 而且基本上必须独占一个 USB Controller。再算算 v2 的情况, Color = $1920 \times 1080 \times 4 \times 30$ 约为 237M, Depth = $512 \times 424 \times 2 \times 30$ 约为 12M, 总计约为 250M/s。所以非 USB3.0 不可, 否则传输不了这么大的数据量。显而易见, Color Map 是最占带宽的, 其实可以通过一些其他格式, 比如 I420 或

MJPEG 来减少数据量，然后通过 CPU 或 GPU 来进行解压和回放。

在感官上，v2 要比 v1 细腻很多，如图 3-13 所示：

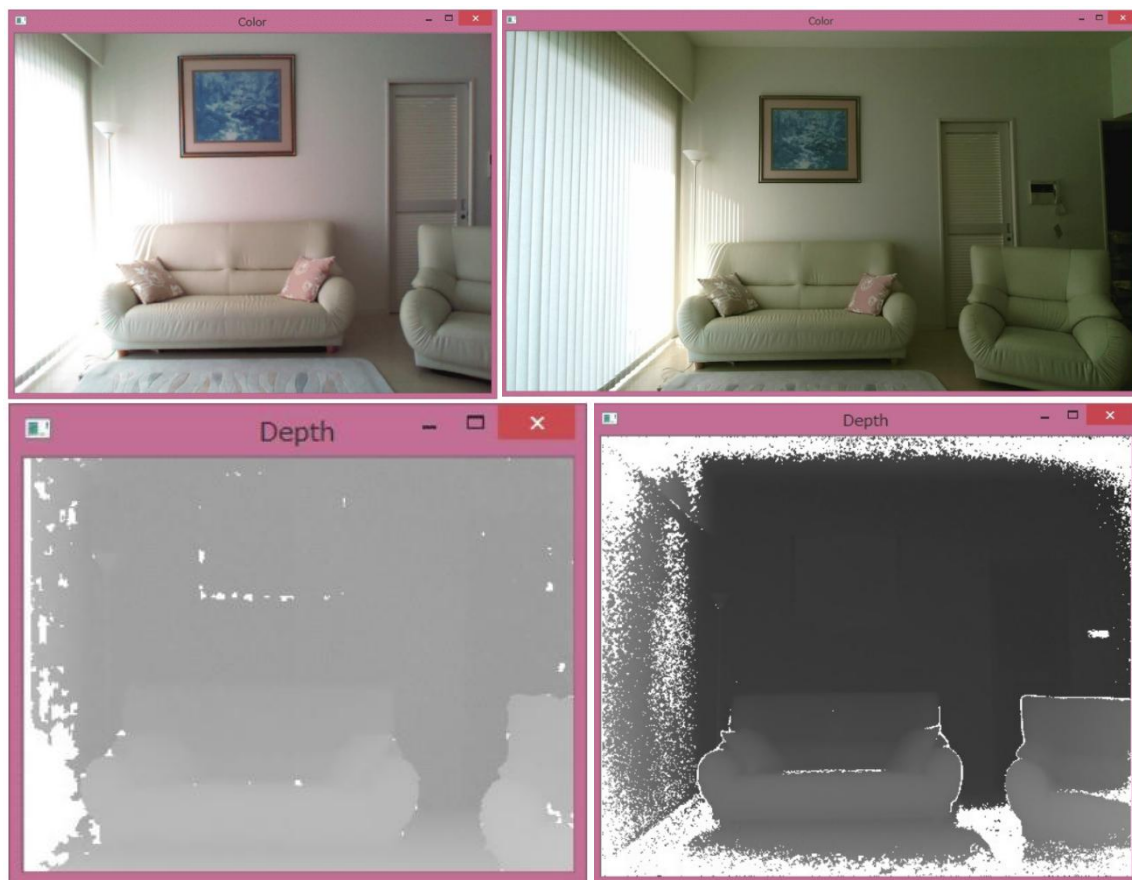


图 3-13 Kinect v1 和 v2 在 RGB 和深度信息的对比

Fig.3-13 Kinect v1 and v2 in RGB and depth information comparison

虽然在人眼的细腻察觉下，v2 要比 v1 各方面好很多，但是在计算机处理的时候，v1 模糊的画面和不清楚的深度图像也不会影响计算机读取每个像素的信息，所以对于实现人体姿势监测功能来说，v1 也完全胜任。

同时，本文使用的 TX2 主机在不接 Hub 的情况下，只有一个 USB2.0 和一个 USB3.0，USB 接口资源非常紧张，因为外设很多，需要扩展很多的 USB 接口出去，机器人最后做完的时候，一共用了四个 USB3.0。所以给 v2 独占的 USB3.0 接口不现实。TX2 上运行 ubuntu16.04 arm64 架构的系统，不能使用微软官方的 SDK，OpenNI2 和 libfreenect2 的功能严重缺失，且很多功能支持不好，所以为了整体硬件资源和软件的稳定性、兼容性，选择 Kinect v1。

到现在，整个机器人的硬件已经搭建完成。搭建好的机器人如图 3-14：

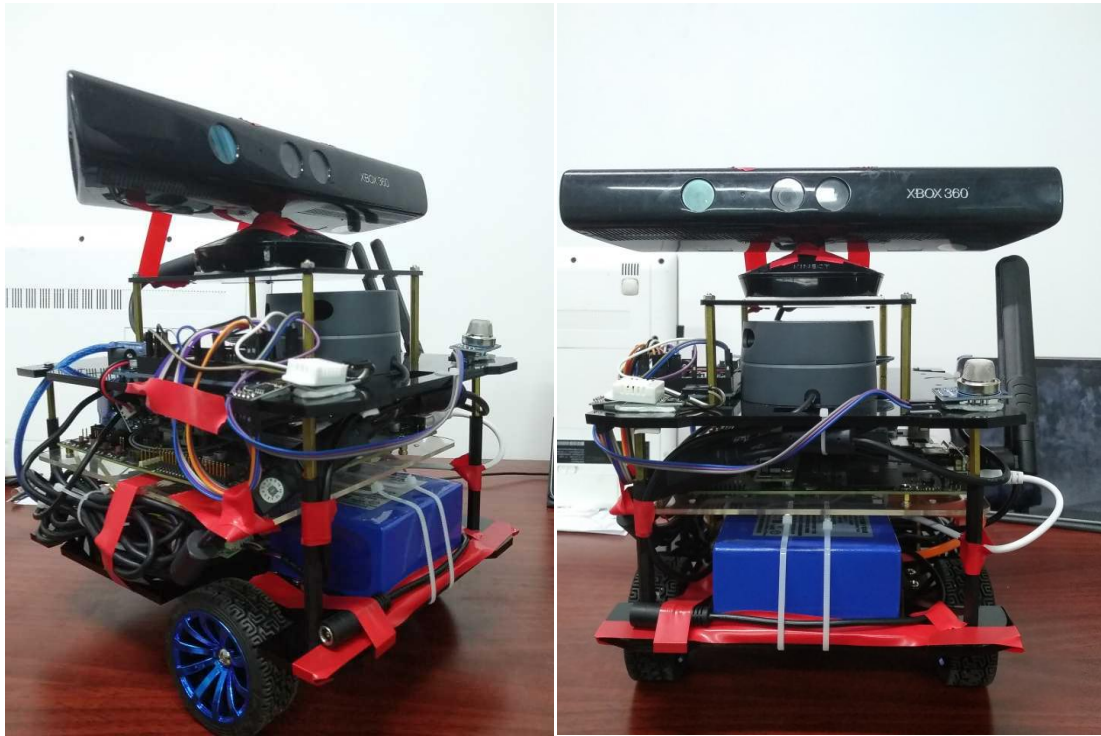


图 3-14 机器人整机

Fig.3-14 Complete robot

4 家庭智能服务机器人自主导航功能的软件设计

4.1 家庭智能服务机器人自主导航功能的下位机软件设计

4.1.1 TB6612FNG 驱动程序设计

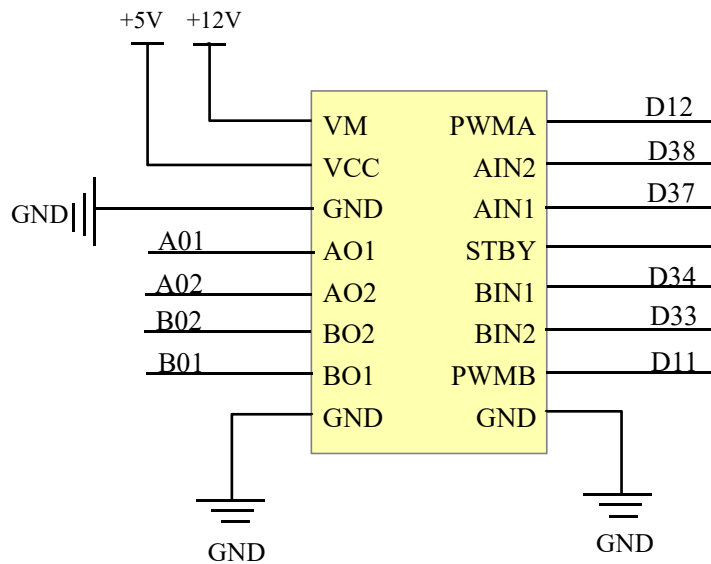


图 4-1 TB6612FNG 原理图

Fig.4-1 The principle diagram of the TB6612FNG

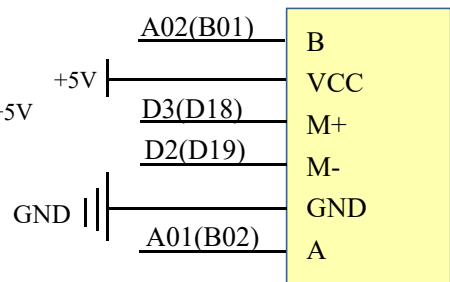


图 4-2 编码器原理图

Fig.4-2 Encoder schematic diagram

表 4-1 TB6612FNG 真值表

Tab.4-1TB6612FNG truth table

IN1	IN2	PWM	STBY	01	02	模式状态
H	H	H/L	H	L	L	制动
L	H	H	H	L	H	反转
L	H	L	H	L	L	制动
H	L	H	H	H	L	正转
H	L	L	H	L	L	制动
L	L	H	H	OFF	OFF	停止
H/L	H/L	H/L	L	OFF	OFF	待机

```

void initMotorController(){
    pinMode(PWMA,OUTPUT);
    pinMode(PWMB,OUTPUT);
    pinMode(AIN1,OUTPUT);
    pinMode(AIN2,OUTPUT);
    pinMode(BIN1,OUTPUT);
    pinMode(BIN2,OUTPUT);
    pinMode(STBY,OUTPUT);
}

if (i == LEFT){
    if (spd >= 0){
        directionLeft = FORWARDS;
        digitalWrite(STBY,HIGH);
        digitalWrite(AIN1,HIGH);
        digitalWrite(AIN2,LOW);
        analogWrite(PWMA,spd);
    }
}

```

图 4-3 TB6612FNG 部分驱动程序

Fig.4-3 TB6612FNG partial driver

TB6612FNG 的驱动程序非常简单，只需要指定每个 GPIO 的工作模式，根据真值表进行输出就可以。

4.1.2 编码器程序设计

正交编码器（Quadrature Encoder）是一种用于测量旋转速度和方向的传感器，通过积分（累加）运算后，还可以用来测算距离。最常见的正交编码器有两个输出信号：A 相 和 B 相，有些编码器会有 Z 相 的校准功能（用于消除累计误差）。「正交」一词来源于 AB 两个信号的特征，一般情况下 A 相 和 B 相 的输出信号总是有 $\pi/2$ 的相位差。

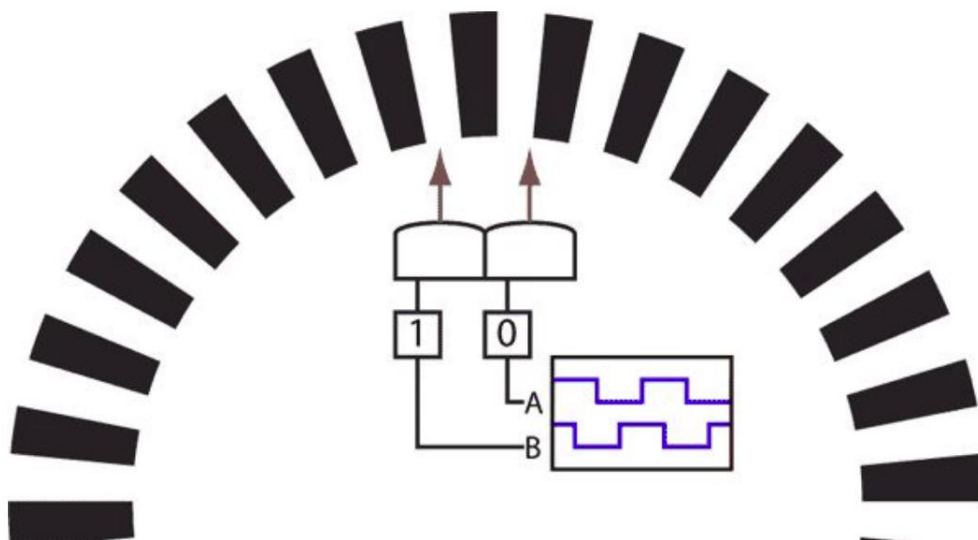


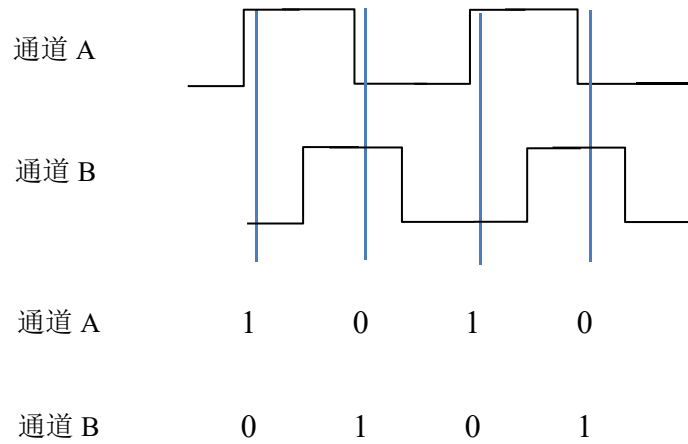
图 4-4 AB 相编码器工作原理

Fig.4-4 Principle of AB phase encoder

如图 4-4 所示是编码器的工作原理，空白的部分是栅格，A 和 B 分别连接到两个传感

器单元上，通过栅格的不断变化获取变化的波形，如图 4-5 所示：

编码器顺时针转



编码器逆时针转

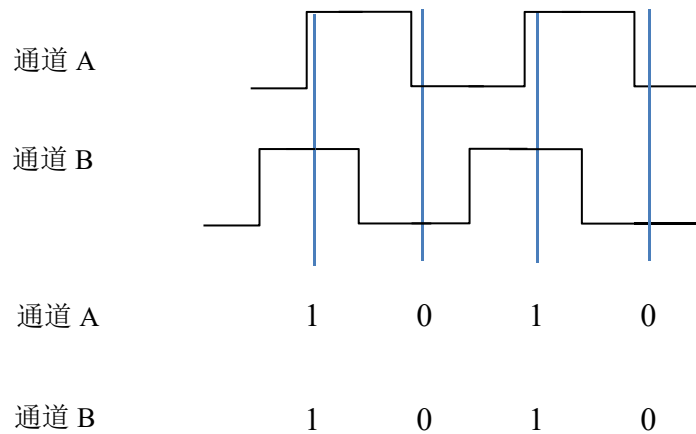


图 4-5 编码器计数原理

Fig.4-5 Encoder counting principle

从 A、B 相输出的波形可以看出，如果编码器顺时针转，当 A 相波形电平发生变化时，变化完成时刻 A、B 相的电平状态总是相反的；如果编码器逆时针转，则两相的电平状态是相同的。

据此便可以设计编码器正反转和计数的程序。程序中捕获 A 相的电平状态，当电平发生变化时，查看 A、B 相的电平关系，如果电平状态相反，则是顺时针转；如果电平状态相同，则是逆时针转。设计时只需根据自己想要的旋转方向调整即可。同时，捕获 A 相电平转换的次数，便可得到编码器的计数。

```

void initEncoders(){
    pinMode(AENCODER_1,INPUT_PULLUP);
    pinMode(AENCODER_2,INPUT_PULLUP);
    pinMode(BENCODER_1,INPUT_PULLUP);
    pinMode(BENCODER_2,INPUT_PULLUP);
    attachInterrupt(1,leftEncoderEvent,CHANGE);
    attachInterrupt(4,rightEncoderEvent,CHANGE);
}

void leftEncoderEvent(){
    if (digitalRead(AENCODER_1) == HIGH) {
        if (digitalRead(AENCODER_2) == LOW) {
            left_enc_pos--;
        }
        else {
            left_enc_pos++;
        }
    }
}

```

图 4-6 部分编码器数据捕获程序

Fig.4-6 Partial encoder data capture program

4.1.3 pid 程序设计

本文所设计机器人的下位机程序架构的方块图，如图 4-7 所示：

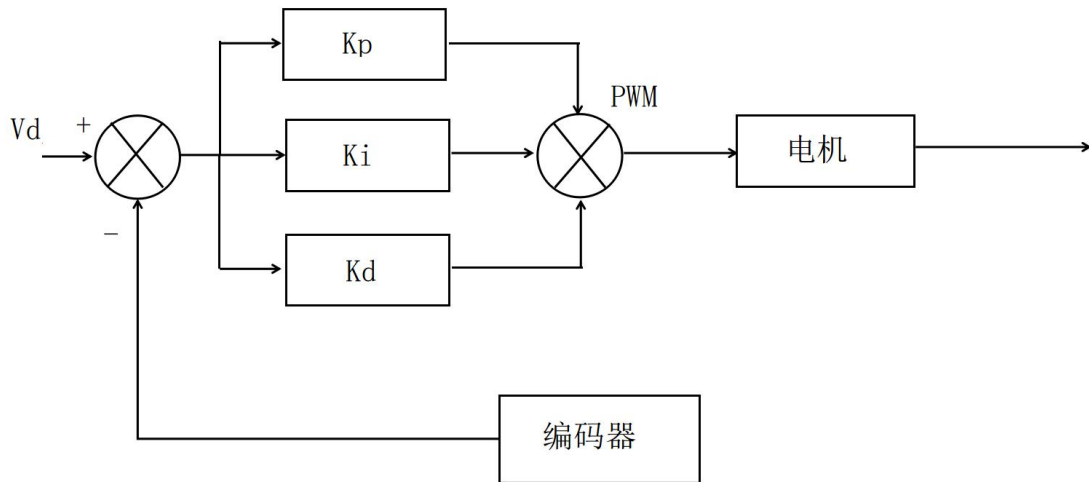


图 4-7 pid 控制方块图

Fig.4-7 Pid control block diagram

V_d 是经过运动解算后期望的速度，实际是上位机发出的 Twist 消息，消息包含 $linear.x$ 、 $linear.y$ 、 $linear.z$ 、 $angular.x$ 、 $angular.y$ 、 $angular.z$ 控制参数，因为本文的机器人是差分轮控制，仅需要 $linear.x$ 和 $angular.z$ 的数据值。在上位机中，这两个数据经过解算生成 pid 的参数值，传给下位机对 pwm 进行调整。编码器作为负反馈环节，对轮子转速实时采集，与期望速度进行对比，交给 pid 进行调整控制。

Pid 部分代码如下：

```
input = p->Encoder - p->PrevEnc;
```

```
Perror = p->TargetTicksPerFrame - input;
```

```
output = (right_Kp * Perror - right_Kd * (input - p->PrevInput) + p->ITerm) / right_Ko;
```

```
p->PrevEnc = p->Encoder;
```

```
output += p->output;
```

这一部分就是 pid 的核心，编码器作为反馈环节与期望值之间进行靠近，并通过调整 PWM 来使轮子转速达到并保持在上位机发送过来的期望转速。

4.1.4 下位机通讯程序设计

下位机虽然起着底层控制中枢的作用，但是所有的总控制信号都来自于上位机，所以有一套稳健的通讯机制对系统的稳定性来说是很重要的。

本文所设计机器人的通讯部分使用 ros_arduino_bridge 的通讯协议，上位机向下位机发送的是控制指令，下位机收到发来的控制指令后，根据控制指令执行对应的程序，如图 4-8 所示：

```
#define ANALOG_READ    'a'          case MOTOR_SPEEDS:
#define GET_BAUDRATE    'b'          lastMotorCommand = millis();
#define PIN_MODE        'c'          if(arg1 == 0 && arg2 == 0){
#define DIGITAL_READ    'd'          setMotorSpeeds(0,0);
#define READ_ENCODERS   'e'          resetPID();
#define MOTOR_SPEEDS    'm'          moving = 0;
#define PING            'p'          }
```

图 4-8 部分通讯程序代码

Fig.4-8 Part of the communication program code

下位机要时刻接收上位机发来的数据，所以整个通讯过程独占循环等待命令，如图 4-9 所示：

```
void loop() {
  while(Serial.available() > 0){
    chr = Serial.read();
    if(chr == 13){
      if(arg == 1) argv1[index] = NULL;
      else if(arg == 2) argv2[index] = NULL;
      runCommand();
      resetCommand();
    }
  }
}
```

图 4-9 处理通讯请求部分代码

Fig.4-9 Part of the code that handles communication requests

4.2 家庭智能服务机器人自主导航功能的上位机程序设计

4.2.1 上位机与下位机通讯程序设计

上位机首先要解决的就是和下位机的通信问题，上位机通过 SLAM、Navigation 发出的控制命令 Twist 消息必须经过这一“桥梁”才能打通和下位机的联系。下位机已经写好了循环等待命令的程序，上位机通讯程序的任务就是根据上层算法发送出的不同命令生成下位机可以接受的控制指令和消息指令，同时也要接收下位机上传的数据。使用 Python 可以方便的实现 TX2 的串口通信功能，所以本部分代码使用 Python 语言设计。如下是上下位机进行协议对接的部分代码：

```
def send(self, cmd):  
    self.port.write(cmd + '\r')
```

上位机通讯功能做好以后，在进行 SLAM 功能之前，还有一个功能也是必须的。机器人要想完成 SLAM 和 Navigation 过程，一个不可或缺的部分就是 TF 树。

在机器的世界里，对于一个构件的移动的认知和人类不尽相同。我们要想控制机器人实现各个部件的准确的移动和位置关系，坐标系是不可缺少的。

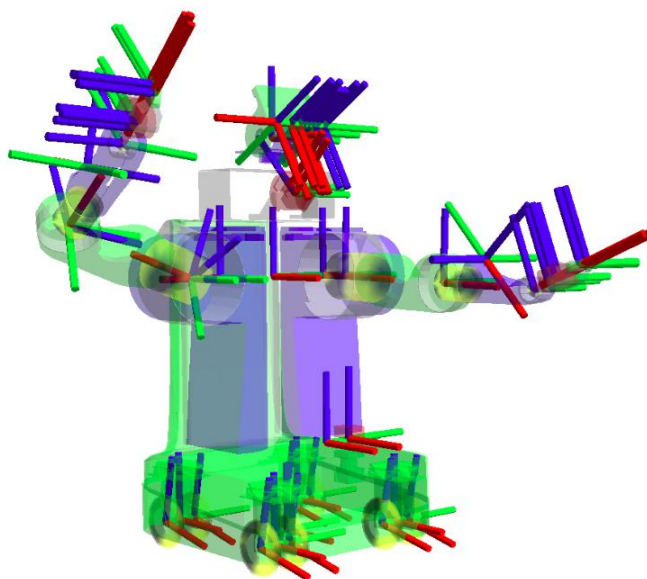


图 4-10 机器人坐标系系统

Fig.4-10 Robot coordinate system

4-10 这张图，只要有关节的地方、可以移动的地方就会有一个坐标系来标示对应关节的位置和运动姿态。但是如此多的坐标系的维护可并不容易，所以 TF 树就是帮助我们维护这庞大的坐标系的工具包。

同理，本文设计的机器人也需要 TF 树的维护。作为移动机器人，机器人系统需要知道底盘的位置、底盘移动的距离等信息，所以需要给底盘维护一个 TF。为了下位机的代码的简洁和尽量小的 CPU 消耗，我们把维护下位机 TF 树的任务放在上位机上执行。如果对

ROS 比较熟悉，就会知道放在哪里执行无所谓，只要有程序来维护这个 TF 树就可以。SLAM、Navigation 算法除了必须激光雷达的数据外，对于 Navigation，里程计（本文所设计机器人中即指处理编码器数据获得的机器人行走数据）的信息也是必须的，因为对于导航来说，激光雷达能起到一定的特征点匹配作用，即能通过特征点的匹配确定自己在地图上的位置，但是激光雷达的特征匹配的效率和能力相比深度摄像头还是差很多的，所以导航功能需要知道机器人前进的距离，并通过里程计数据推算出机器人的姿态，和激光雷达的数据融合确定机器人在地图中准确的位置和导航时的精度。

在 ROS 中，里程计的数据有 odom 主题（odom topic）发布，同时，还会发布一个 odom frame，odom topic 和 odom frame 虽然名字一样，但确实完全不同的两个概念。Topic 和 frame 是 ROS 中的两个概念，topic 包含着消息和消息类型，而 frame 则是 tf 主要的参与者，tf 父子级之前的关系就是多个 frame 构建起来的。Odom topic 包含着机器人行走的 Odometry 信息，包括 x 方向前进的距离，z 方向旋转的角度等；odom frame 则是 tf 树的一个节点，该节点维护了 odom 的坐标系，也即机器人两个编码器轴线的中心。所以 odom frame 是与底座固定在一起的。

```
self.odomBroadcaster.sendTransform(
    (self.x, self.y, 0),
    (quaternion.x, quaternion.y, quaternion.z, quaternion.w),
    rospy.Time.now(),
    self.base_frame,
    "odom"
)
```

上面程序便是 odom topic 和 odom frame 的发布来源，通过代码实现，就确定了 odom 的信息和位置。

对于一个 ROS 机器人来说，还有不可或缺的 frame 便是 base_frame，默认情况下 base_frame 和机器人的几何中心重合。Tf 树的维护方向是 odom->base_frame，所以上面的程序中，指定了 odom 发布的子节点 base_frame，父节点是 odom。在不使用 tf_static_publisher 或者 urdf 模型显式指定的情况下，base_frame 和 odom 是重合的。（本文所设计机器人使用的 base_frame 是 base_link）。

4.2.2 SLAM 算法实现

SLAM（Simultaneous localization and mapping），同步定位与地图构建是一种概念：希望机器人从未知环境的未知地点出发，在运动过程中通过重复观测到的地图特征（比如，墙角，柱子等）定位自身位置和姿态，再根据自身位置增量式的构建地图，从而达到同时定位和地图构建的目的。由于其重要的理论与应用价值，被很多学者认为是实现真正全自主移动机器人的关键。

前面也已经说过，SLAM 主要分为激光 SLAM 和视觉 SLAM。因为本文所设计机器人

使用激光 SLAM，所以会着重说明激光 SLAM 的相关技术内容。

SLAM 是一个完整的定位建图系统，涉及到很多的数学和优化知识，大概框架如图 4-13:

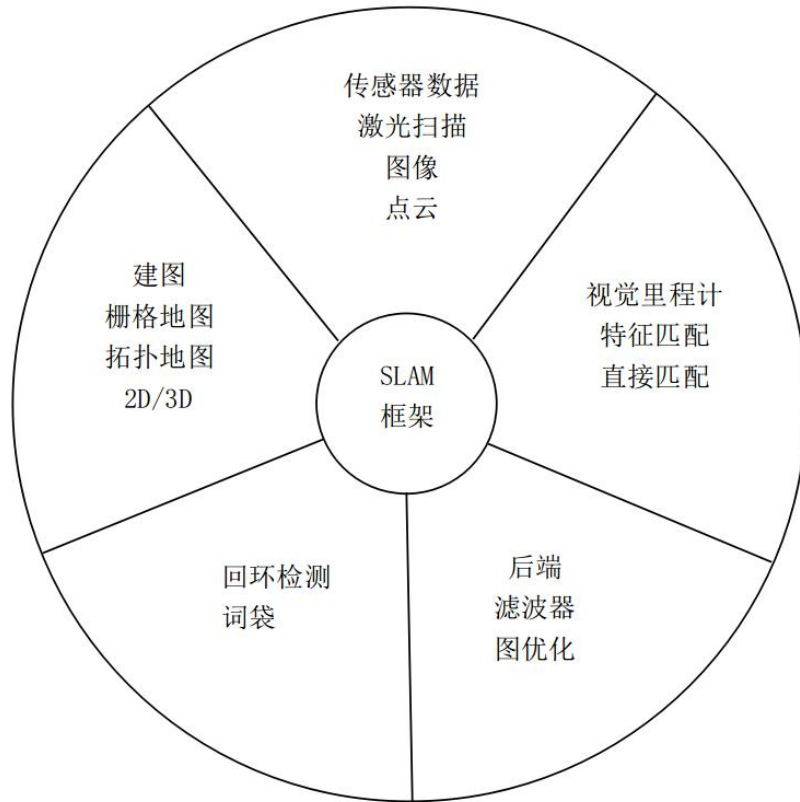


图 4-11 SLAM 系统框架

Fig.4-11 SLAM system framework

一个 SLAM 系统框架主要包含图 4-13 中的几部分：传感器数据、建图、里程计、后端、回环检测。

对于激光 SLAM，传感器的数据有激光扫描数据，同时激光雷达也会发布点云数据，可以看到激光点形成的点云。视觉里程计估计两个时刻机器人的相对运动，在激光 SLAM 中，可以将当前的观测与全局地图进行匹配，用 ICP (Iterative Closest Point, 迭代最近邻) 求解相对运动。激光 SLAM 系统通过对不同时刻两片点云的匹配与比对，计算激光雷达相对运动的距离和姿态的改变，也就完成了对机器人自身的定位。

视觉里程计和轮式里程计一样，都会有误差。对于短时数据来说相对精准，但是随着时间的延长，累计误差也会很大。如果光靠视觉里程计进行机器人轨迹的预测，那么建出的图可能会发生很大的偏差，方形的区域可能会变成多边形，轨迹也会有严重的漂移。所以，一个完整的 SLAM 系统会有后端程序进行优化。

早期的 SLAM 后端使用滤波器方式。SLAM 最早的提出者 R. Smith 等人就把 SLAM 建构成了一个 EKF (Extended Kalman Filter, 扩展卡尔曼滤波) 问题。他们按照 EKF 的形

式，把 SLAM 写成了一个运动方程和观测方式，以最小化这两个方程中的噪声项为目的，使用典型的滤波器思路来解决 SLAM 问题。21 世纪之后，SLAM 研究者开始借鉴 SfM（Structure from Motion）问题中的方法，把捆集优化（Bundle Adjustment）引入到 SLAM 中来。优化方法和滤波器方法有根本上的不同。它并不是一个迭代的过程，而是考虑过去所有帧中的信息。通过优化，把误差平均分到每一次观测当中。在 SLAM 中的 Bundle Adjustment 常常以图的形式给出，所以研究者亦称之为图优化方法（Graph Optimization）。图优化可以直观地表示优化问题，可利用稀疏代数进行快速的求解，表达回环也十分的方便，因而成为现今视觉 SLAM 中主流的优化方法。

SLAM 中的回环检测功能是当机器人到达过曾经到过的地方的时候，能够识别出来并且将点云数据匹配，减小累计误差。

对于为了实现室内环境的建图和导航功能来说，视觉里程计、回环检测、后端优化等手段都是为了建图效果的完美，所以建图算是 SLAM 非常重要的一环，又是严重依赖其他部分的一环。以 gmapping 为例，看一下 SLAM 建图的过程：

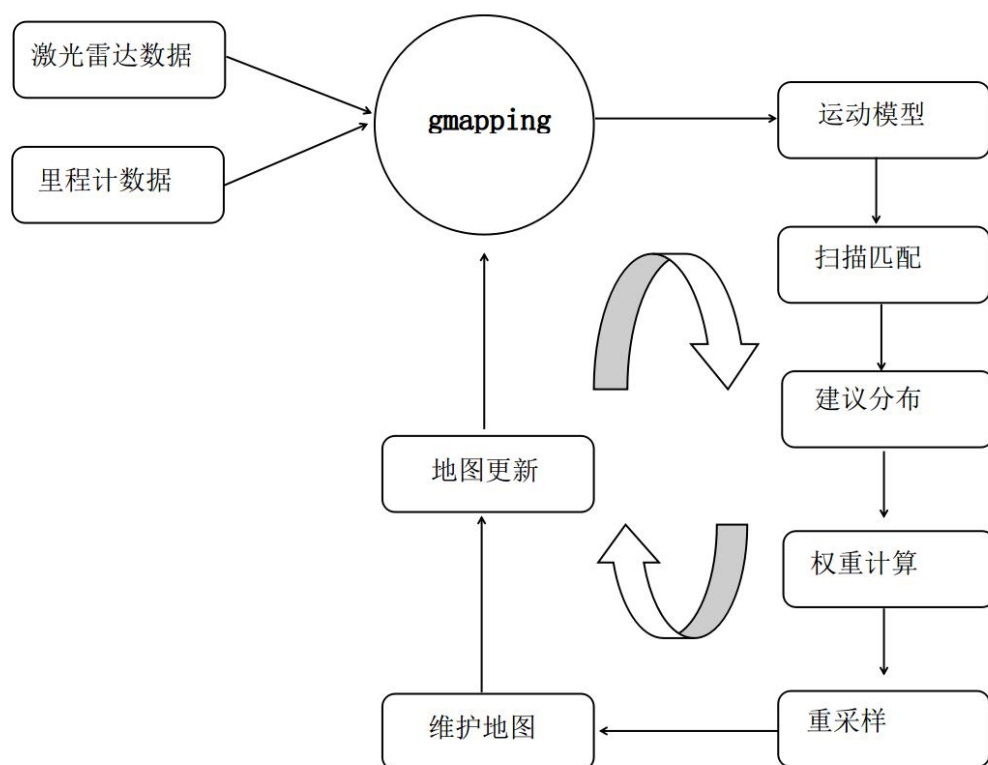


图 4-12 gmapping 建图过程

Fig.4-12 Gmapping drawing process

数据输入：Gmapping 需要输入激光雷达的数据和里程计数据，这些数据将传入 openslam gmapping 包中。Gmapping 作为经典的建图算法，目前应用还是非常多，而且在激光雷达数据和里程计数据精度都较高的前提下，建图效果很好。但是这样成为了 gmapping 的一大弱点，现实中对于一般成本的编码器来说，里程计数据总会发生漂移，这

会造成建图效果不好。同时对于无人机和不平坦的路面，gmapping 的建图效果也一般。

运动模型：gmapping 会根据上一时刻的位姿及里程计数据，预测下一时刻的粒子位姿。

扫描匹配：对每个粒子执行扫描匹配算法，其作用是找到每个粒子在 t 时刻位姿的最佳坐标。一般粒子数越多对于匹配效果越好，但是会增加计算开销。

建议分布：其混合运动模型和观测模型，根据上一步扫描匹配获得的最佳坐标，围绕该坐标取若干位置样本（距离差值小于某阈值）计算均值与方差，使得当前粒子位置满足该均值方差的高斯分布。

权重计算：对每个粒子的权重进行更新和归一化。Gmapping 算法思想在《概率机器人》书中有详细的介绍，对于机器人位置的估计，采用对全局地图随机撒粒子的方式，然后根据观测结果报告每个粒子的权重，权重大的粒子持续收敛，权重小的粒子逐渐退化消亡，这样经过几次迭代，权重大的粒子会剩余并集中在一起，就是机器人的位置。

重采样：使用 Neff 判断是否进行重采样。需要注意的是，重采样频率越高，粒子退化越严重，建图精度会降低。

完整 SLAM 系统的组成部分和建图过程大致如上所述。上文以 gmapping 作为示例算法说明，目前已有多种优秀的激光 SLAM 算法。因为里程计数据的漂移和累计误差问题，新的 SLAM 算法往往会不依赖与里程计数据。对比主流的激光 SLAM 算法如下：

Hector SLAM:

Scan-matching 方法：使用高斯牛顿法。因为不需要里程计数据，严重依赖激光雷达数据，对激光雷达要求较高，要求能提高高帧率扫描。

后端优化方法：使用 EKF 滤波。早期的 SLAM 算法大都依靠 EKF 进行滤波优化。

优点：不需要使用里程计，在无人机和地面不平坦情况依然可以有很好的建图质量。

缺点：要求雷达的更新频率高，且测量噪声小。要想取得较好的建图质量，需要机器人以较小的速度进行建图操作；没有回环检测；使用最小二乘法匹配扫描点，因此扫描角很小且噪音较大的 Kinect 相机无法使用，匹配时会陷入局部点，地图比较混乱。

Gmapping SLAM:

Scan-matching 方法：类似于梯度下降法。

Gmapping 目前是应用最方便也是最广泛的 2D SLAM 算法。利用 RBPF（Rao-Blackwellized 粒子滤波）方法实现机器人的定位和建图，大大降低了 SLAM 问题的复杂度。粒子滤波的效果一般与粒子数量成正比，为了获得较好的滤波结果，一般需要大量的粒子参与计算，但是这会引入计算量的上升。上面 gmapping SLAM 过程已经说明，粒子在更新过程中会伴随着权重的变化，在重采样过程中会伴随着粒子耗散的问题，大权重粒子显著，小权重粒子会消失，通过限制重采样次数可能一定程度解决此问题。

优点：在长廊及低特征场景中建图效果好。

缺点：依赖里程计，故不能应用在无人机上；没有回环检测，无法适应地面不平坦区域。

Cartographer:

Cartographer 是谷歌在 2016 年开源的 SLAM 框架，总体效果在 2D SLAM 中算是最好的。Carter 引入了一个新的概念 submap，一个 submap 是由一定数量的激光雷达扫描数据组成的，建图和优化时，重点是对 submap 进行操作。Cartographer 的重点内容是具有很高的适应性来完成 submap 的创建和用于闭环检测的扫描匹配策略的实现。很高的适应性是指 cartographer 既可以只用激光雷达的数据完成建图，也可以融合里程计的数据建图，还可以融合 IMU 的数据。Submap 的使用虽然很大程度上提高了建图的效果，但是随着时间的推移，越来越多的 submap 被创建，submap 间的累计误差会越来越大，因此 cartographer 使用闭环检测来消除这些误差。不同于以往的 SLAM 算法，cartographer 没有使用滤波的方法进行优化，而是使用 ceres 进行扫描匹配的优化，以类似图优化的模型进行位置估计。Cartographer 算法的基本思路如图 4-15 所示：

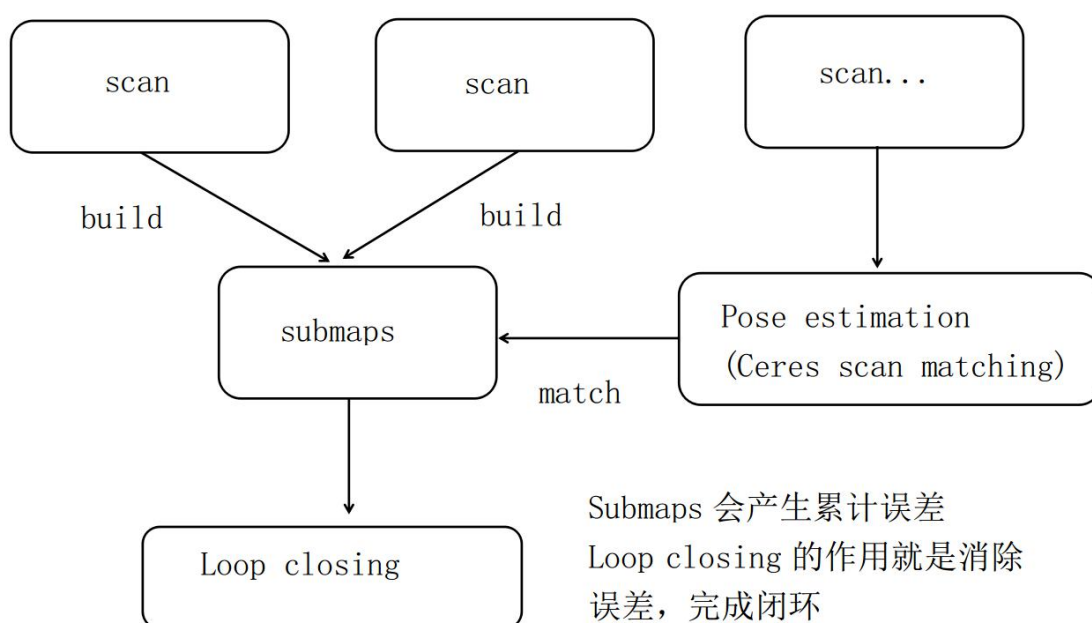


图 4-13 cartographer 建图过程

Fig.4-13 Cartographer mapping process

Cartographer 代码主要包含两部分：

cartographer: 负责处理来自激光雷达、里程计和 IMU 的数据，使用这些数据完成建图操作，是 cartographer 理论的底层实现；

Cartographer_ros: ros 通讯机制的封装接口，用于接收 ros 标准格式的数据和发布 ros 格式的数据，以便 cartographer 的数据类型可以方便的借助 ros 强大的通讯机制完成数据的分发和处理操作，属于 cartographer 的上层实现。此部分代码还包含.pbstream 文件的保存程序、.pbstream 转换为 ros 栅格地图用于导航的程序等。

在 SLAM 程序的设计上，主要是编写配置文件。Cartographer 的配置文件用 lua 语言编写。

```

options = {
    map_builder = MAP_BUILDER,
    trajectory_builder = TRAJECTORY_BUILDER,
    map_frame = "map",
    tracking_frame = "base_link",
    published_frame = "base_link",
    odom_frame = "odom",
    provide_odom_frame = true,
    publish_frame_projected_to_2d = false,
    use_odometry = false,
    use_nav_sat = false,
    use_landmarks = false,
    num_laser_scans = 1,
    num_multi_echo_laser_scans = 0,
    num_subdivisions_per_laser_scan = 10,
    num_point_clouds = 0,
    lookup_transform_timeout_sec = 0.2,
    submap_publish_period_sec = 0.3,
    pose_publish_period_sec = 5e-3,
    trajectory_publish_period_sec = 30e-3,
    rangefinder_sampling_ratio = 1.,
    odometry_sampling_ratio = 1.,
    fixed_frame_pose_sampling_ratio = 1.,
    imu_sampling_ratio = 1.,
    landmarks_sampling_ratio = 1.,
}
MAP_BUILDER.use_trajectory_builder_2d = true
TRAJECTORY_BUILDER_2D.num_accumulated_range_data = 10
TRAJECTORY_BUILDER_2D.min_range = 0.20
TRAJECTORY_BUILDER_2D.max_range = 16.0
TRAJECTORY_BUILDER_2D.submaps.num_range_data = 50
TRAJECTORY_BUILDER_2D.use_imu_data = false
TRAJECTORY_BUILDER_2D.use_online_correlative_scan_matching = true
POSE_GRAPH.constraint_builder.max_constraint_distance = 4

```

以上是部分配置代码，进行完成一系列配置，建图效果一般不会差。如图 4-14 和 4-15 是本文设计机器人的建图效果：

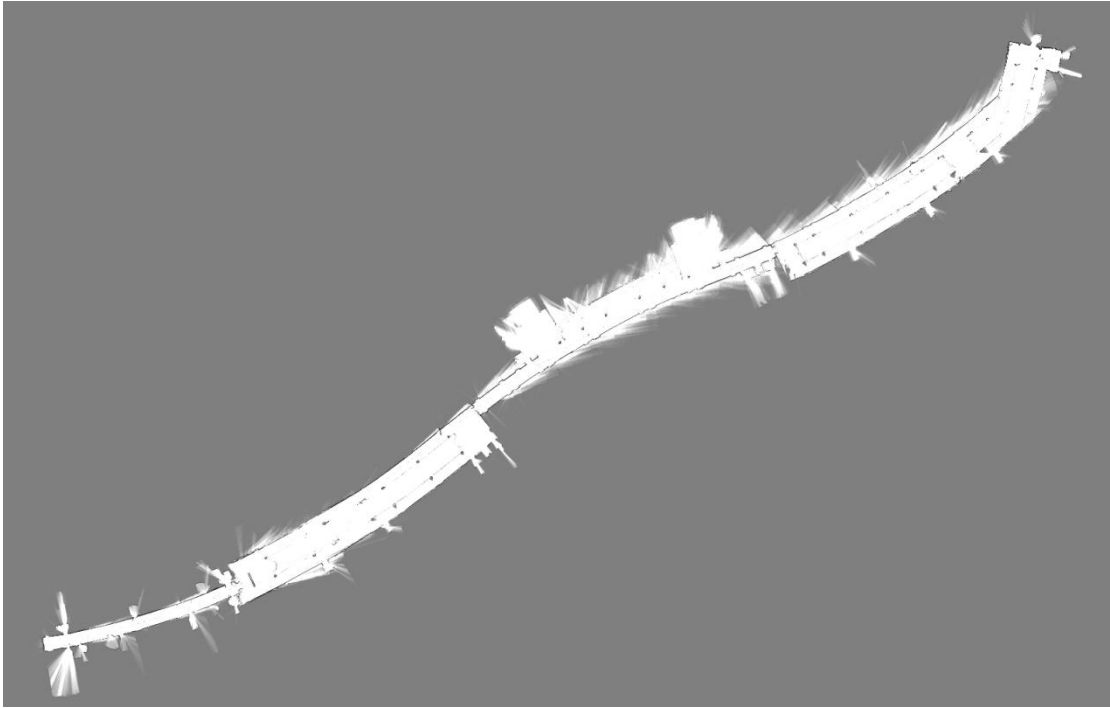


图 4-14 弘毅楼
Fig.4-14 Hong Yi House

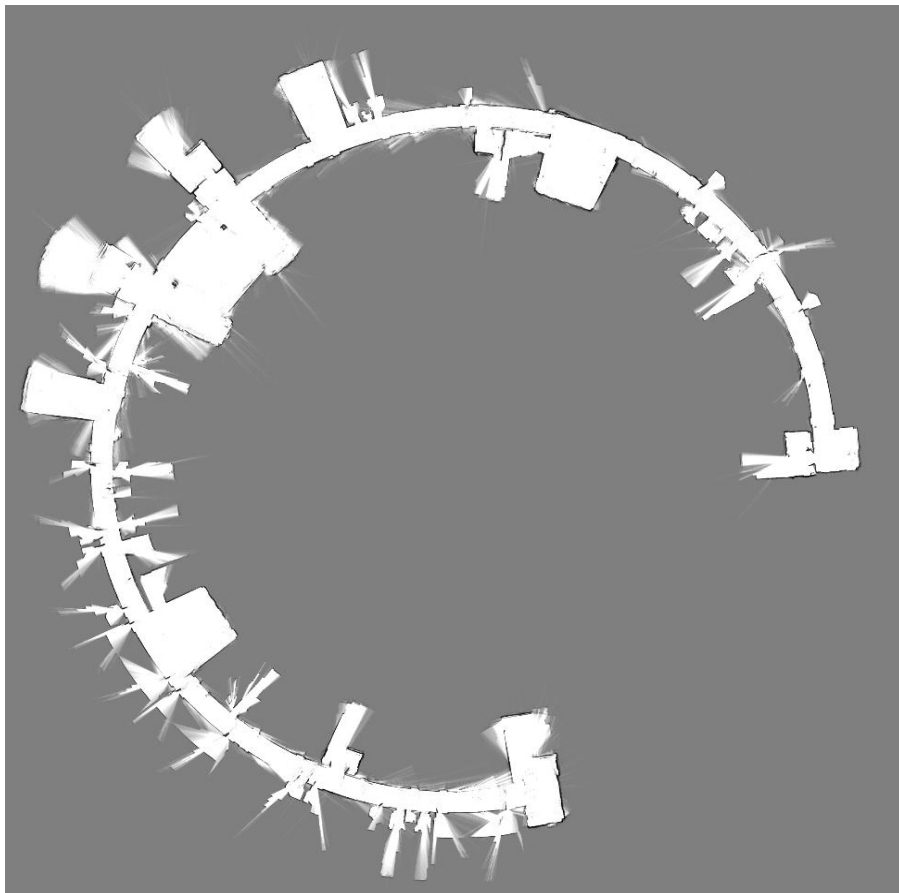
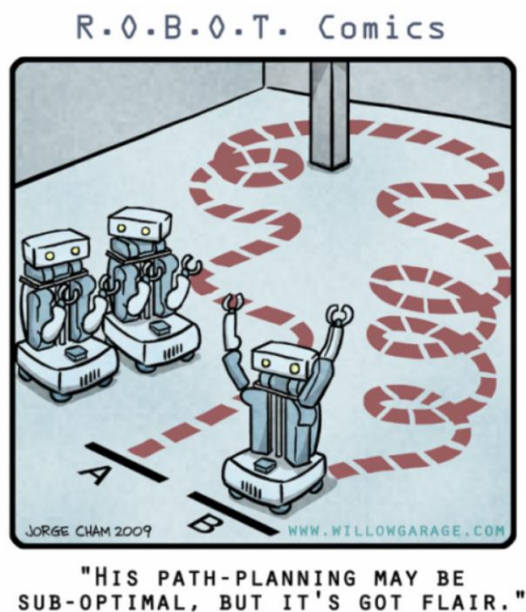


图 4-15 明德楼
Fig.4-15 Mingde Building

总体来说,在整个机器人的配置较为合理的情况下, `cartographer` 建图的效果还是挺好的。

4.2.3 Navigation 功能的实现



Ros 的 Navigation 功能包算是 ros 中非常复杂、非常庞大的一个功能包。涵盖机器人概率定位、全局路径规划、局部路径规划、代价地图、恢复行为等内容,包含了众多算法和配置文件,对于熟练使用 Navigation 包并且真确的配置参数达到理想的导航效果并不容易。Navigation 包中也不乏很古老的而且效果差的算法,但是由于历史原因始终在服役,或者新的算法不符合使用要求等原因,需要对算法进行局部修改。

从概念层面上讲,导航功能包的工作流程是相对简单的。Navigation 接收里程计和激光雷达的数据,通过一系列算法生成速度控制指令发给下位机控制底盘行进。但是,要想在机器人上配置好导航功能包却不是一件简单的事情,整个导航过程涉及很多的地图信息配置和算法配置,而且并不具有通用性,这给机器人的通用设计带来了非常大的阻碍。

使用导航功能的前提条件是机器人必须运行 ros, 有一个准确无误的 tf 树, 使用正确的 ros message types 发布传感器数据, 还需要在上位机为特定机器人配置导航功能包集。

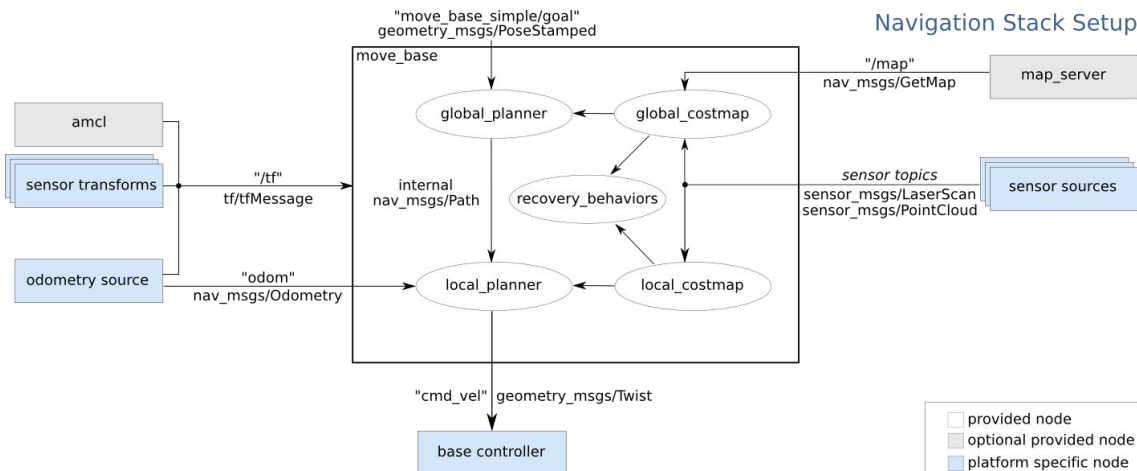


图 4-16 导航功能包整体框架

Fig.4-16 Navigation feature package overall framework

整个导航包的框架结构如图所示。最大框内是 move_base 功能包，也是导航功能最核心最重要的功能包。

蓝色框内是必须根据特定机器人特定提供的功能。对于本文所设计的机器人，需要提供激光雷达的 tf 树信息、激光雷达的线激光数据、odom 信息、下位机控制接口等。这边外围信息流准确无误的配置好后，配置 move_base 和 amcl 的功能才是重中之重。

amcl (adaptive Monte Carlo localization)，自适应蒙特卡洛定位方法，使用自适应的 KLD 方法来更新粒子。而 mcl (Monte Carlo localization) 蒙特卡洛定位法使用的是粒子滤波的方法进行定位。粒子滤波法在《概率机器人》中有相当详细的理论介绍。该算法使用粒子滤波器来表示可能状态的分布，每个粒子代表一种可能的状态，即机器人所处位置的假设。每当机器人感知某事物时，基于递归贝叶斯估计对粒子进行重新采样，即，实际感测数据与预测状态的相关程度。最终，粒子应朝向机器人的实际位置会聚。形象的来说，就是在整个区域均匀的撒上一把粒子，粒子跟随机机器人的运动来运动，使用每个粒子所处位置模拟一个传感器信息跟观察到的传感器信息（一般是激光）作对比，从而赋给每个粒子一个概率。粒子自身的模拟传感器信息和激光雷达的信息一致度越高，此粒子的权重越大，经过多次迭代，权重大的粒子就会收敛在一起，机器人的位置也就会确定出来了。

```

Algorithm MCL( $X_{t-1}, u_t, z_t$ ):
     $\bar{X}_t = X_t = \emptyset$ 
    for  $m = 1$  to  $M$ :
         $x_t^{[m]} = \text{motion\_update}(u_t, x_{t-1}^{[m]})$ 
         $w_t^{[m]} = \text{sensor\_update}(z_t, x_t^{[m]})$ 
         $\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$ 
    endfor
    for  $m = 1$  to  $M$ :
        draw  $x_t^{[m]}$  from  $\bar{X}_t$  with probability  $\propto w_t^{[m]}$ 
         $X_t = X_t + x_t^{[m]}$ 
    endfor
    return  $X_t$ 

```

图 4-17 amcl 算法

Fig.4-17 Amcl algorithm

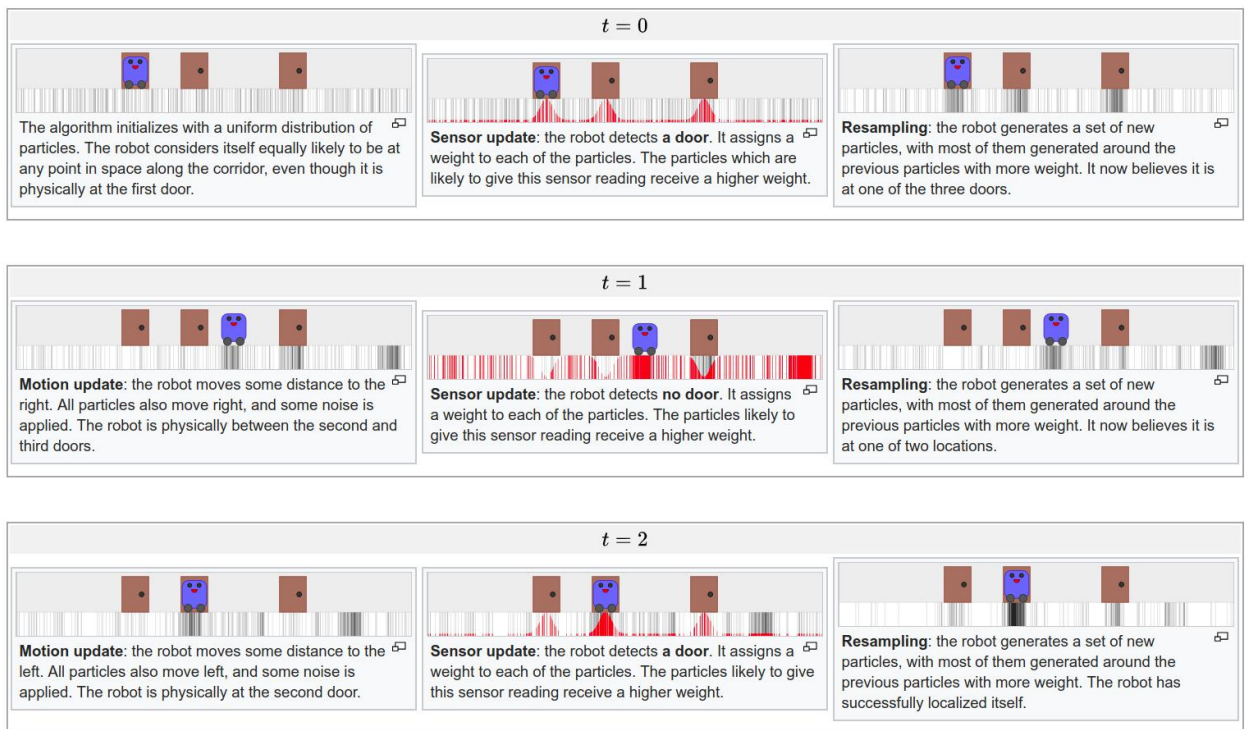


图 4-18 蒙特卡洛算法定位过程

Fig.4-18 Monte Carlo algorithm positioning process

图 4-21 是蒙特卡洛定位的过程原理。 $t=0$ 时刻，粒子在整个区域均匀放置，机器人会觉得它在此区域的任何一个位置，虽然它是在第一个门前；传感器数据感知到自己在一个门前，那么具有相同信号的粒子的权重会增加，此时机器人觉得它在任何一个门前。 $t=1$ 时，机器人向右行走，此时传感器数据没有检测到门，粒子依然会被赋予不同的权重；然后机器人向左走，传感器同样又检测到门，权重被更新，机器人的位置便被确定。

放在真实机器人和真实环境中，amcl 的粒子定位问题大概如图 4-22：

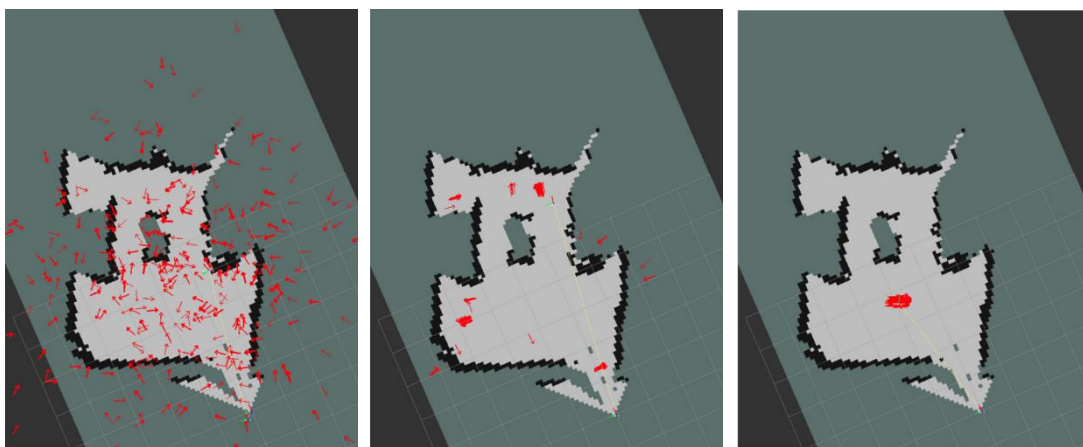


图 4-19 amcl 粒子收敛过程

Fig.4-19 Amcl particle convergence process

$t=0$ 时, 粒子被均匀撒在地图上, $t=1$ 时, 粒子被模拟的传感器信息会和激光雷达获取到的信息进行比对, 信息一致度越高的粒子会被赋予更高的权重, 而权重低的粒子自然退化消亡。粒子携带的模拟信息就是指地图上的栅格点信息。 $t=2$ 时, 随着机器人的继续移动, 激光雷达获取到的实际环境的信息更加丰富, 参加比对的粒子点数量更多, 不断迭代, 最终确定机器人的位置。部分 AMCL 的配置代码如下面所示:

```
<param name="save_pose_rate" value="0.5"/>
  <param name="kld_err" value="0.05"/>
  <param name="kld_z" value="0.99"/>
  <param name="laser_min_range" value="0.25"/>
  <param name="laser_max_range" value="-1.0"/>
  <param name="laser_lambda_short" value="0.1"/>
  <param name="laser_likelihood_max_dist" value="2.0"/>
  <param name="laser_max_beams" value="60"/>
  <param name="laser_model_type" value="likelihood_field"/>
  <param name="laser_sigma_hit" value="0.2"/>
  <param name="laser_z_hit" value="0.5"/>
```

Move_base 框架外的所有信息源都已经准备好, 配置好 move_base 框架便可以进行导航测试。Move_base 框架很复杂, 包含 costmap、global_costmap、local_costmap、global_planner、local_planner、recovery_behavior 等部分。

Costmap 提供了一个维护机器人在占据栅格地图中哪里可以进行导航的配置结构。

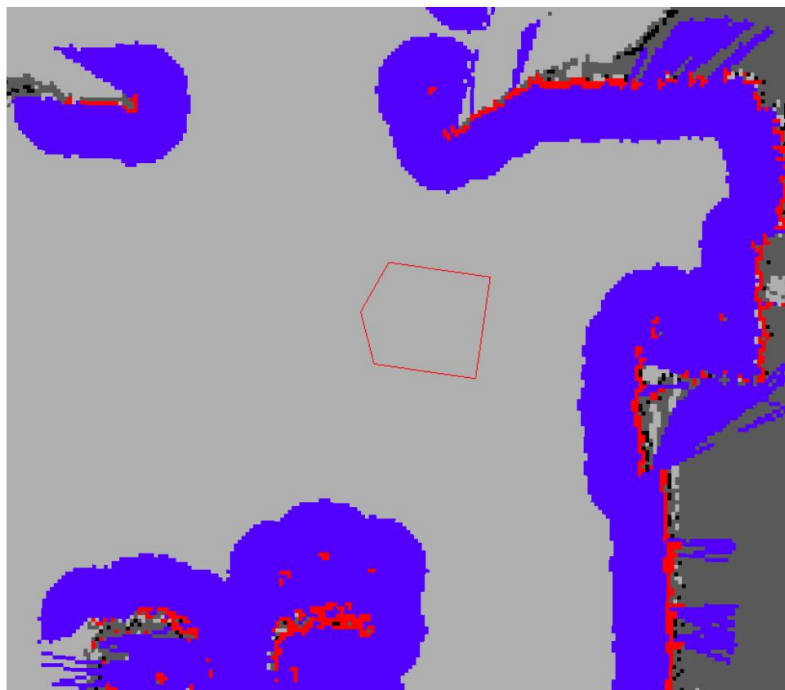


图 4-20 costmap

Fig.4-20 costmap

如图 4-24 所示，在这个代价地图中，红色部分表示障碍物，蓝色部分表示由机器人的内切半径膨胀的障碍物，红色多边形表示机器人的轮廓。为了避免机器人发生碰撞，机器人的轮廓不应该与红色部分相交，并且机器人的中心点不应该穿过蓝色部分。（膨胀半径表示以机器人中心为原点，虚拟出的一个圆形范围，在这个范围内，机器人有可能发生碰撞，所以机器人中心如果穿过蓝色区域，那么膨胀半径会可能和蓝色区域长度相等，机器人发生碰撞）。

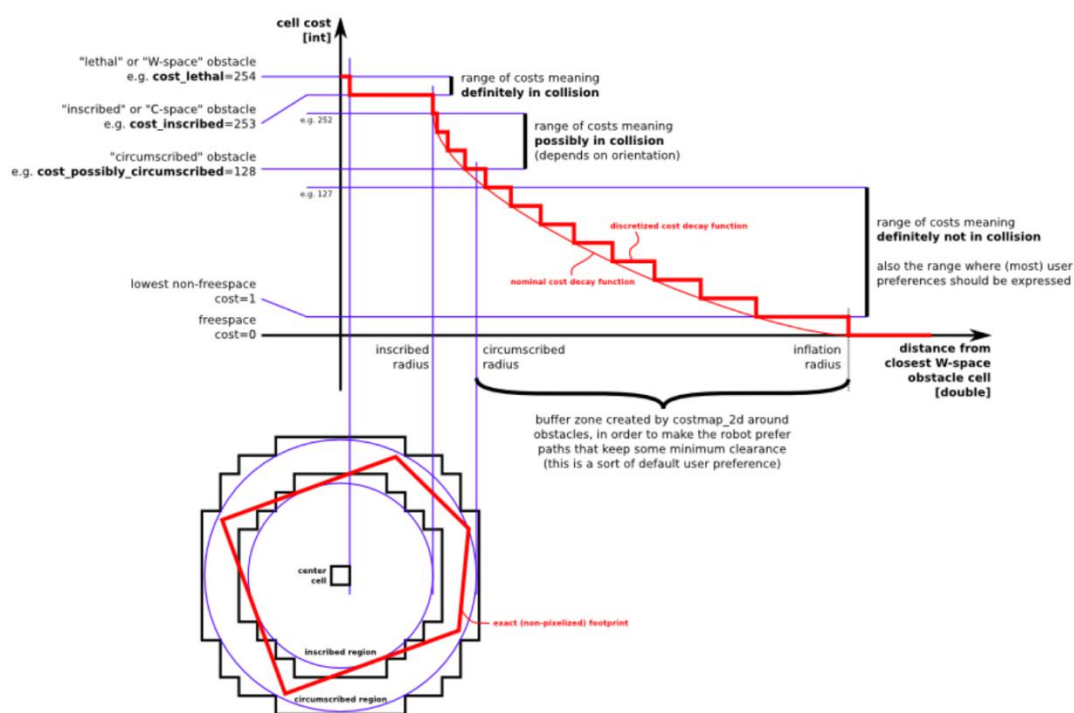


图 4-21 膨胀半径与碰撞检测

Fig.4-21 Expansion radius and collision detection

图 4-21 便是膨胀半径的来源依据。从图中可以看出，为了度量机器人发生碰撞的范围，定义了 lethal、inscribed、possibly circumscribed、freespace、unknown 等几个不同程度的碰撞范围。Letal 意味着区域内存在障碍，如果机器人在这个区域，一定会发生碰撞；inscribed 意味着此区域小于机器人设置的膨胀半径，如果机器人中心在这个区域内，则肯定会与某些障碍物碰撞；possibly circumscribed 使用机器人的膨胀半径作为截止距离，如果机器人的中心位于或高于该区域所示的范围值，则机器人可能发生碰撞；freespace 表示没有任何东西可以阻止机器人进入此区域；unknown 表示此区域没有可知信息。

Costmap 中 cost 的意思即是机器人进入不同区域付出的代价，所以成为代价地图，为机器人的导航和碰撞检测提供了依据。

obstacle_range: 2.5

raytrace_range: 3.0

footprint: [[-0.18, -0.10], [-0.18, 0.10], [0.07, 0.10], [0.07, -0.10]]

inflation_radius: 0.3

此为 `costmap_common` 中的部分代码，`global_costmap` 和 `local_costmap` 都会使用此部分代码。`Global_costmap` 和 `local_costmap` 的部分代码如下：

```
global_costmap:
  global_frame: /map
  robot_base_frame: base_link
  update_frequency: 0.5
local_costmap:
  global_frame: odom
  robot_base_frame: base_link
  update_frequency: 0.5  #before 5.0
```

`Global_costmap` 和 `local_costmap` 的代码基本相同，只是它们起了不同的作用。`Global_costmap` 是为 `global_planner` 提供地图信息服务，`local_costmap` 是为 `local_planner` 提供地图信息服务。

在有代价地图的情况下，`move_base` 可以进行路径规划操作，同样分为 `global_planner` 和 `local_planner`。

`Global_planner` 的作用是在整个地图上规划出一条最适合的路径来，在整体情况下，机器人是会沿着这条道路走，但是机器人处在一个动态的环境下，可能会遇到突然出现的障碍，所以 `local_planner` 的作用就是对实时的变化进行捕捉，并规划出局部范围的小路径以避免新出现的障碍。

在 `move_base` 中，`global_planner` 有两套系统：`navfn` 包和 `global_planner` 包。在这两个包中，又都实现了 A*算法和 `dijkstra` 算法。这是因为在 2013 年以前，没有 `global_planner` 包，只有 `navfn` 可以用，但是 `navfn` 中的 A*算法有 bug，所以默认使用 `dijkstra`。现在推荐使用 `global_planner` 包，更稳定、紧凑、高效。

A*是一种计算机算法，广泛用于寻路和图遍历，是个多个点之间寻找路径的过程。A*是一种知情搜索算法，或者是最佳优先搜索，意味着它是根据加权图表制定的：从图的特定起始节点开始，它旨在找到具有最小值的给定目标节点的路径成本（最少行驶距离，最短时间等）。它通过维护源自起始节点的路径树并且一次一个地延伸那些路径直到满足其终止标准来实现这一点。具体来说，A*选择最小化的路径 $f(n) = g(n) + h(n)$ 的

其中 n 是路径上的下一个节点， $g(n)$ 是从起始节点到 n 的路径的成本，而 $h(n)$ 是一个启发式函数，用于估算从 n 到目标的最便宜路径的成本。当它选择扩展的路径是从开始到目标的路径或者没有路径可以扩展时，A*终止。

A*搜索在机器人运动规划问题中搜索从起始节点到目标节点的路径的图示。空心圆代表开放集中的节点，即那些仍有待探索的节点，填充的节点位于封闭集中。每个封闭节点上的颜色表示从开始的距离：越绿，越远。人们可以首先看到 A*在目标方向上沿直线移动，然后当击中障碍物时，它探索通过开放集合中的节点的替代路线。

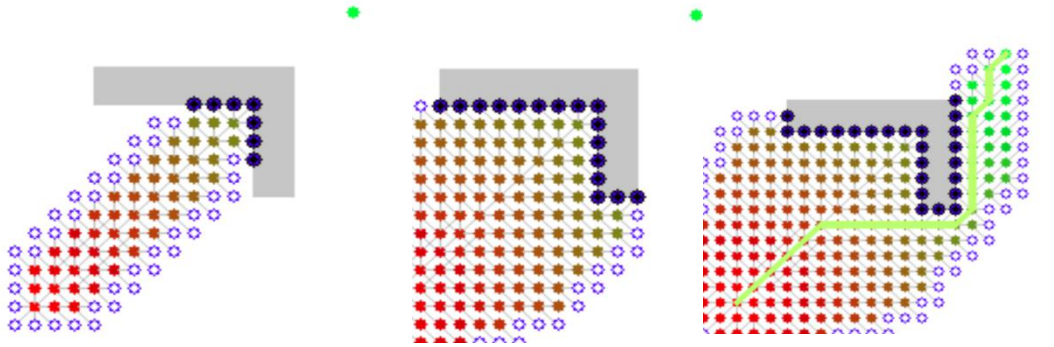


图 4-22 A *算法寻找路径过程

Fig.4-22 A * algorithm to find the path process

一个 A *算法的实例，其中节点是与道路连接的城市，而 $h(x)$ 是到目标点的直线距离：

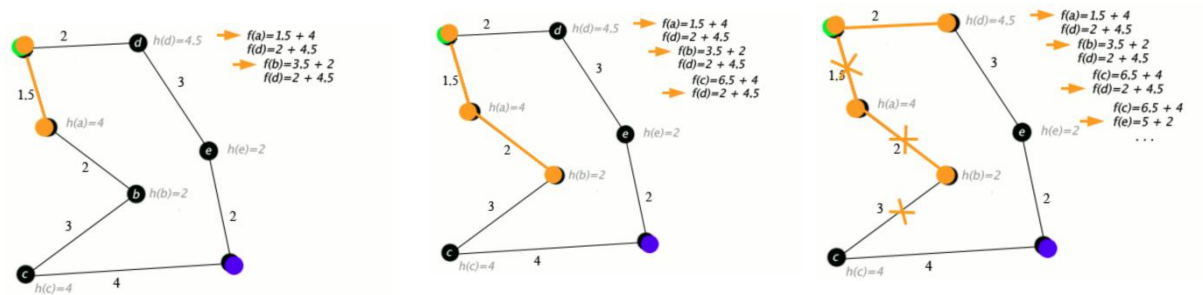


图 4-23 A *算法寻找路径实例

Fig.4-23 A * algorithm to find a path instance

Dijkstra 算法（或 Dijkstra 的最短路径优先算法，SPF 算法）是用于在图中的节点之间找到最短路径的算法，其可以表示例如道路网络。它在功能上和 A*算法很相似，A*算法也可以看成 Dijkstra 算法的改进版。

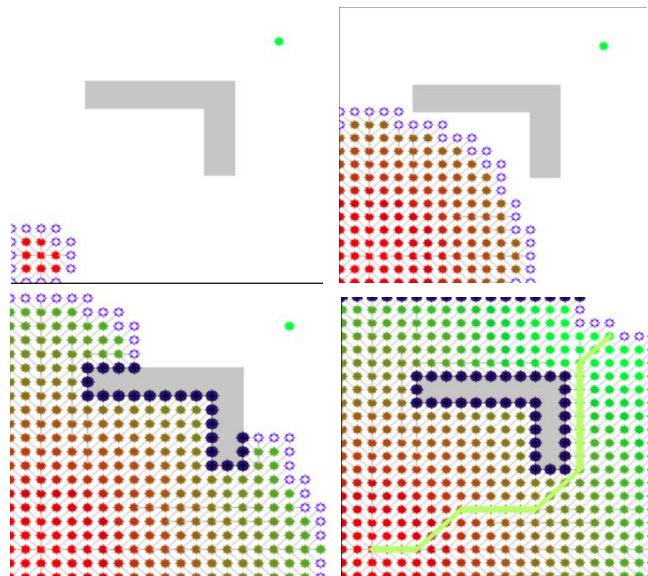


图 4-24 dijkstra 算法寻找路径过程

Fig.4-24 Dijkstra algorithm to find the path process

图 4-24 是同样使用和 A*算法相同的寻找路径的演示图来看一下 dijkstra 算法的寻找路径的方式。

图 4-25 是一个 dijkstra 算法的实例。Dijkstra 的算法找到 a 和 b 之间的最短路径。它选择具有最低距离的未访问顶点，计算通过它到每个未访问的邻居的距离，并且如果更小则更新邻居的距离。当与邻居完成时，标记访问（设置为红色）。

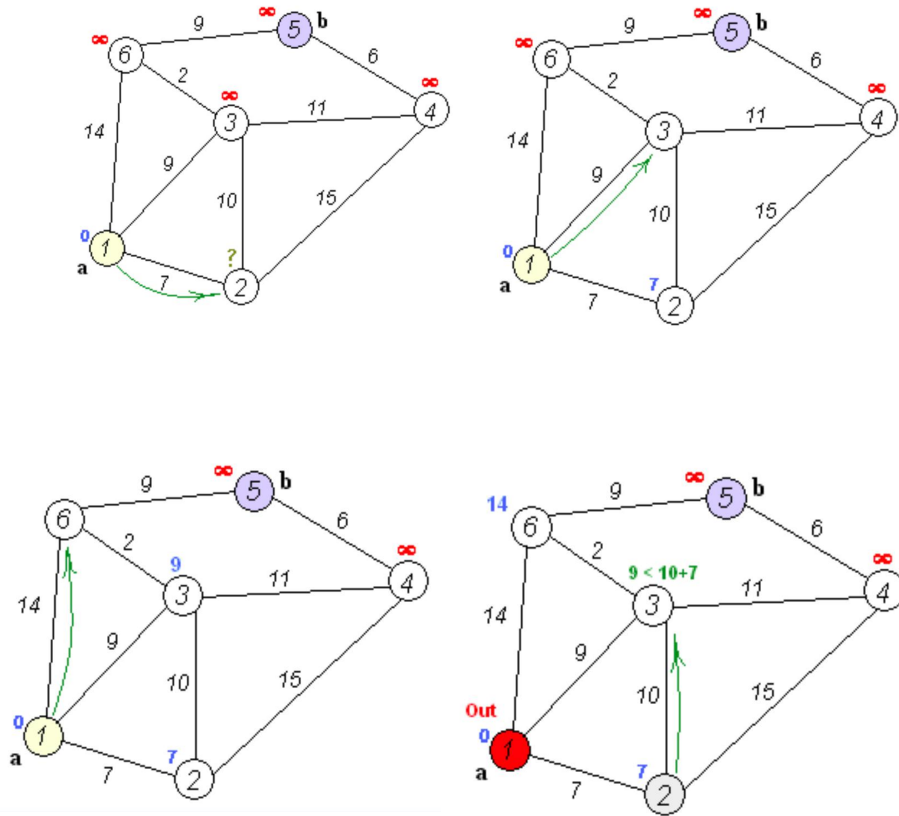


图 4-25 Dijkstra 算法寻找路径的搜索算法过程

Fig.4-25 Dijkstra algorithm search path search algorithm process

从上面的介绍可以看出，Dijkstra 算法的实质是广度优先搜索，是一种发散式的搜索，所以空间复杂度和时间复杂度都比较高。对路径上的当前点，A*算法不但记录其到源点的代价，还计算当前点到目标点的期望代价，是一种启发式算法，也可以认为是一种深度优先的算法，其效率较 dijkstra 高很多，所以本文设计的机器人使用 A*算法。

在 move_base 的配置文件中，启用较新的 global_planner 包，在不显式指定使用 dijkstra 算法的时候，默认使用 A*算法。

base_global_planner: global_planner/GlobalPlanner

在局部路径规划的算法上，ros 同样提供了两套算法：Trajectory Rollout 和 DWA。

Trajectory Rollout 和 DWA 算法的基本思想如下：

- (1) 在机器人的控制空间（dx, dy, dtheta）中进行离散采样；
- (2) 对于每个采样速度，从机器人的当前状态执行正向模拟，以预测如果采样速度在

某个（短）时间段内应用会发生什么)；

(3) 使用包含以下特征的度量来评估（得分）由前向模拟产生的每个轨迹：接近障碍物、接近目标、接近全局路径和速度、丢弃非法轨迹（与障碍物碰撞的轨迹）。

(4) 选择得分最高的轨迹并将相关的速度发送到机器人底盘。

(5) 清除当前得分并重复以上操作。

DWA 与 Trajectory Rollout 的不同之处在于如何对机器人的控制空间进行采样。在给定机器人的加速度极限的情况下，Trajectory Rollout 在整个前向模拟周期内从可实现的速度集合中进行采样，而 DWA 在给定机器人的加速度极限的情况下仅针对一个模拟步骤从可实现的速度集合中进行采样。这意味着 DWA 是一种更有效的算法，因为它可以采样更小的空间，

从经验来看，Trajectory Rollout 和 DWA 基本具有相同的性能，而且 DWA 的效率更高。

base_local_planner: dwa_local_planner/DWAPlannerROS

在 move_base 的配置文件中启用 DWA 算法即可。

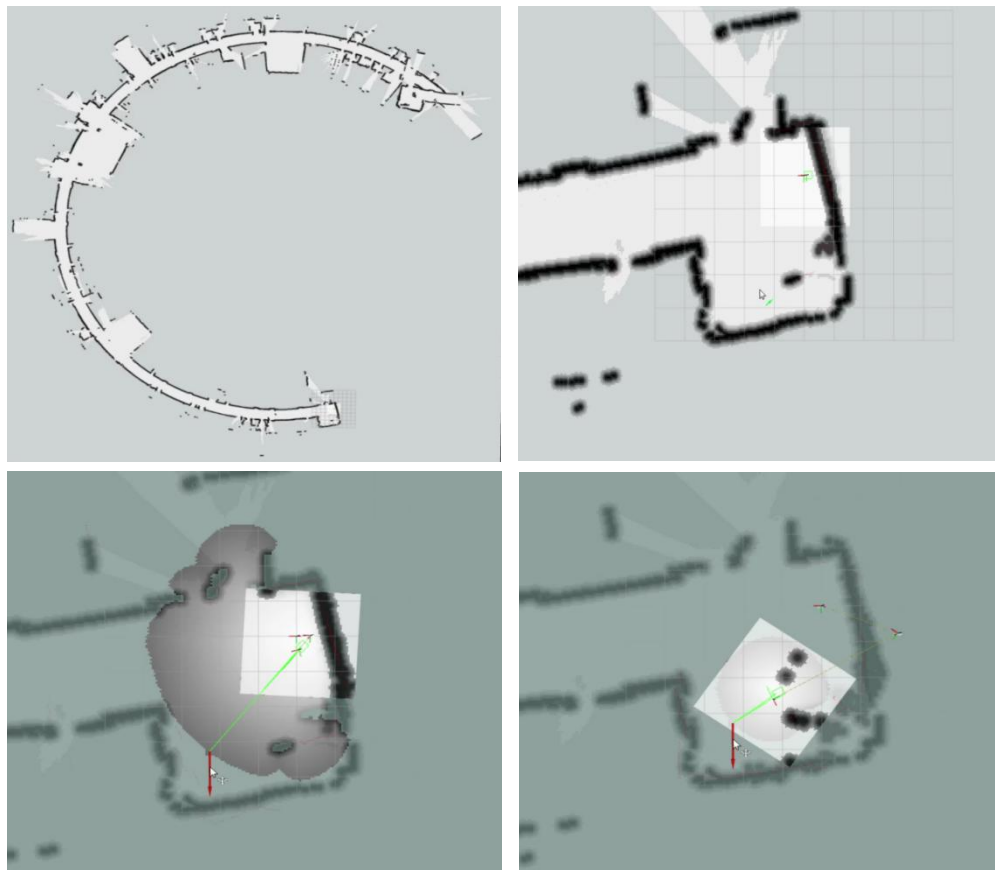


图 4-26 导航过程

Fig.4-26 Navigation process

如图 4-26 所示是在明德楼进行导航的过程。底层的地图就是 global_costmap，机器人周围小的明亮的区域就是 local_costmap。Local_costmap 负责所有动态事物的实时更新。从第三第四幅图可以看出，机器人快要到终点的时候 local_costmap 内多了几个黑点，上面那两个是我的两个助手，右边那三个是长凳和我，在 global_planner 规划出全局路径之后，机

器人在服从这个整体路径的基础上，会使用 `local_planner` 对路径进行小范围的修整，以避免障碍物。

5 姿势检测和环境感知系统设计

5.1 姿势识别

得益于使用强大的 Kinect 摄像头作为视觉传感器,可以方便的实现对人体姿势的检测。在 Windows 平台上,可以使用 Windows SDK 方便的实现人体姿势、手势等的识别,在 linux 平台上,可以使用 openni 和 NITE 实现这部分功能。

Openni (Open Natural Interaction) 是一个行业领导的非营利组织和开源软件项目,专注于认证和改进自然交互 (NI) 设备的自然用户界面和有机用户界面的互操作性,使用这些设备的应用程序和便利的中间件访问和使用此类设备。

通过 openni 提供的强大的交互功能,利用 Kinect 便可以检测到人体的骨骼信息。

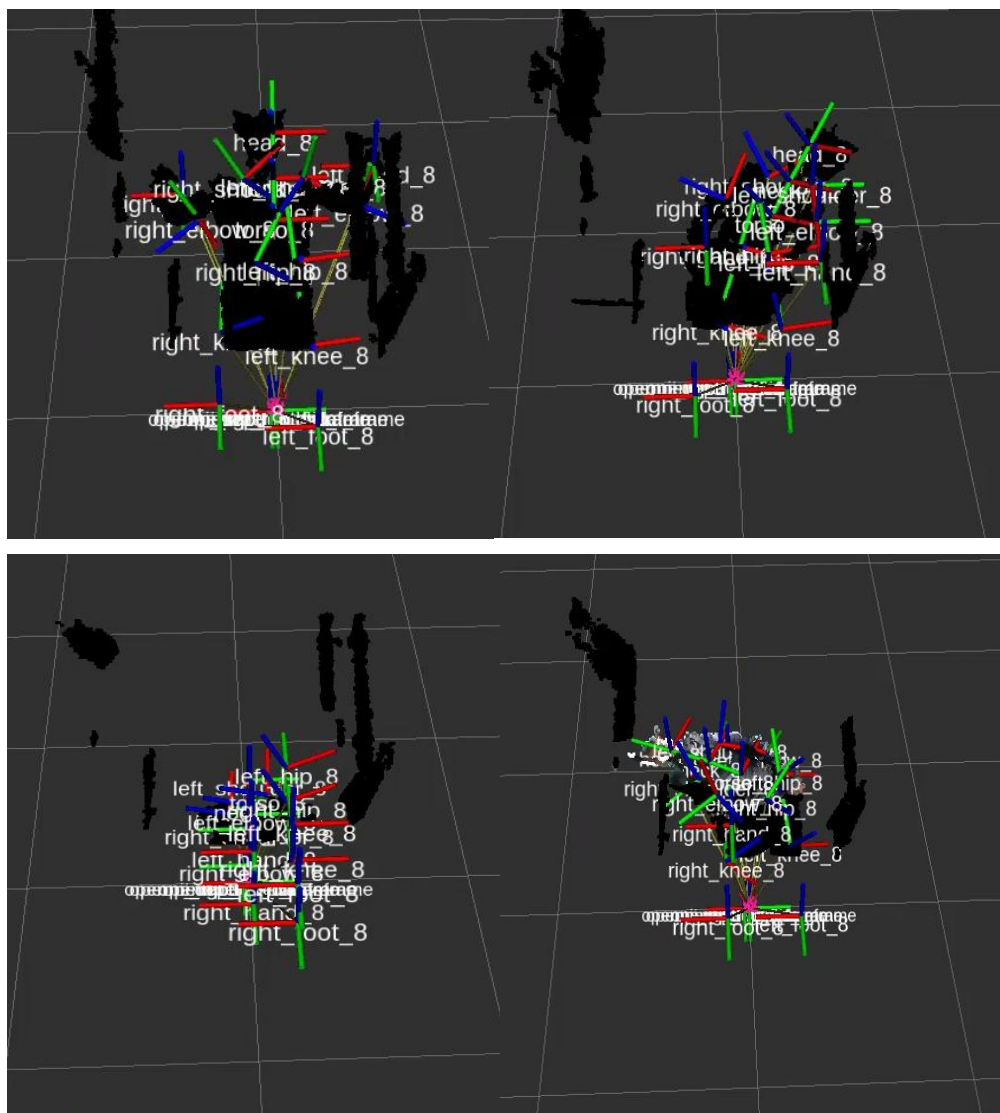


图 5-1 姿势检测

Fig.5-1 Posture detection

Openni 会发布 ros 标准消息,在 rviz 中订阅这些消息便可以完成对骨骼的提取,并且

获取骨骼的坐标点。通过判断头部坐标和脚部坐标值，如果两值的差在某一个很小的阈值内，则可判定人体处于躺着的姿势；另外，通过追踪头部坐标的速度和加速度值，可以更加精确的判定是自然躺下还是突然躺下。

5.2 环境感知系统

环境感知系统主要包括烟雾传感器、光照传感器、文本转语音模块、温湿度传感器、无线收发器、执行器等组成。环境感知传感器获取环境信息，并将重要信息通过语音模块播报出来；无线收发器和执行器的作用就是当检测的环境信息超标时，不仅报警而且能远程控制执行器对相应的设备进行控制。

5.2.1 烟雾传感器硬件及软件设计



图 5-2 MQ-2 烟雾传感器

Fig.5-2 MQ-2 smoke sensor

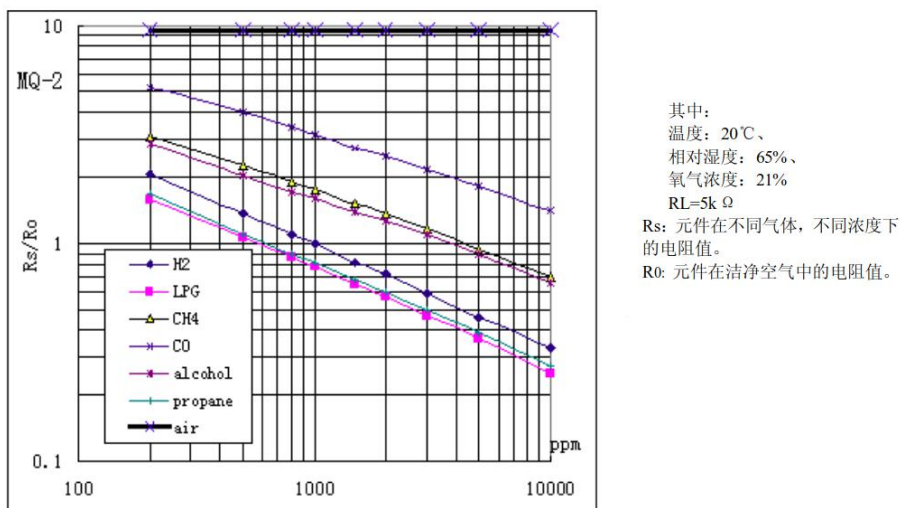


图 5-3 MQ-2 系列传感器的灵敏度特性

Fig.5-3 Sensitivity characteristics of MQ-2 series sensors

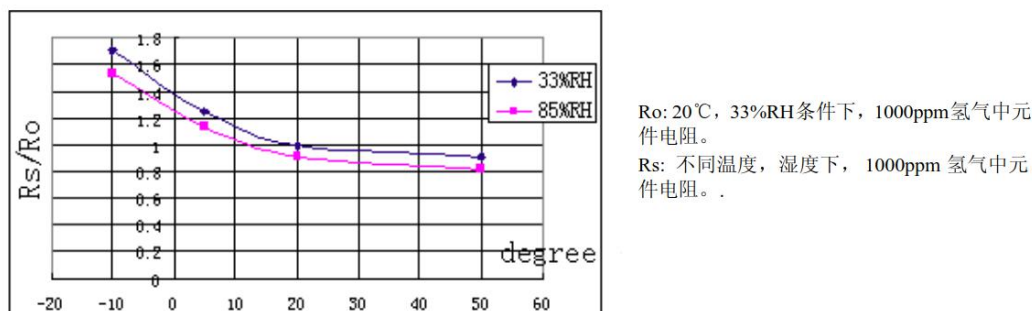


图 5-4 MQ-2 系列传感器温湿度特性

Fig.5-4 Temperature and humidity characteristics of MQ-2 series sensors

从图 5-4 可以看到 MQ-2 系列传感器的灵敏度特性。在不同的气体浓度下， R_s/R_o 的值发生变化，传感器正是根据此特性得出环境中的气体浓度。但是从图 6-3 可以看出，温湿度还对 R_s/R_o 的值有影响，所以要补偿掉温湿度对 R_s 影响的那部分。这部分的代码主要的作用就是把灵敏度特性提取出来并且加入温湿度补偿特性。部分代码如下所示：

```
int MQGetGasPercentage(float rs_ro_ratio, int gas_id)
{
    if ( accuracy == 0 ) {
        if ( gas_id == GAS_HYDROGEN ) {
            return (pow(10,((-2.109*(log10(rs_ro_ratio))) + 2.983)));
        } else if ( gas_id == GAS_LPG ) {
            return (pow(10,((-2.123*(log10(rs_ro_ratio))) + 2.758)));
        } else if ( gas_id == GAS_METHANE ) {
            return (pow(10,((-2.622*(log10(rs_ro_ratio))) + 3.635)));
        } else if ( gas_id == GAS_CARBON_MONOXIDE ) {
            return (pow(10,((-2.955*(log10(rs_ro_ratio))) + 4.457)));
        } else if ( gas_id == GAS_ALCOHOL ) {
            return (pow(10,((-2.692*(log10(rs_ro_ratio))) + 3.545)));
        } else if ( gas_id == GAS_SMOKE ) {
            return (pow(10,((-2.331*(log10(rs_ro_ratio))) + 3.596)));
        } else if ( gas_id == GAS_PROPANE ) {
            return (pow(10,((-2.174*(log10(rs_ro_ratio))) + 2.799)));
        }
    }
}
```

5.2.2 光照传感器的硬件及软件设计

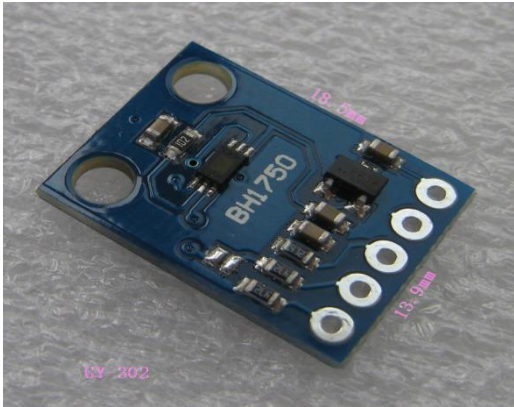


图 5-5 BH1750

Fig.5-5 BH1750

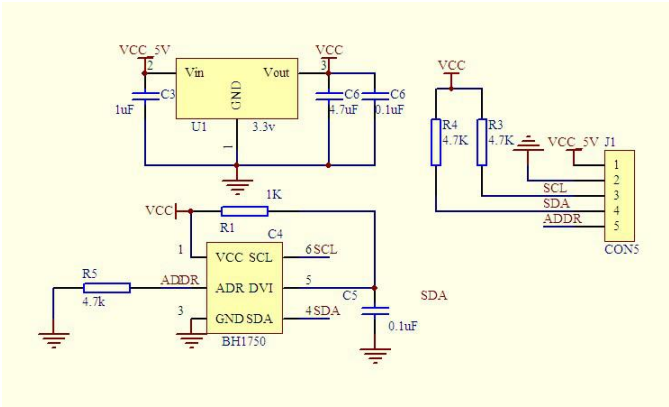


图 5-6 BH1750 原理图

Fig.5-6 BH1750 schematic

光照传感器的核心是 BH1750FVI 集成芯片，是一种用于两线式串行总线接口的数字型光强度传感器集成电路。它的分辨率高，可以探测较大范围的光强度变化。

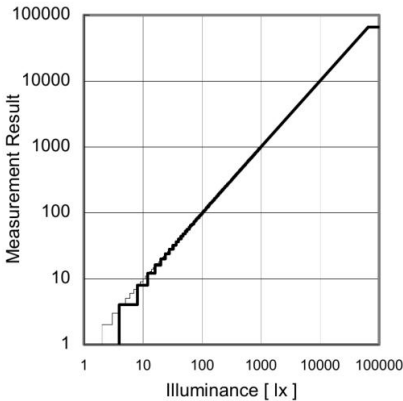
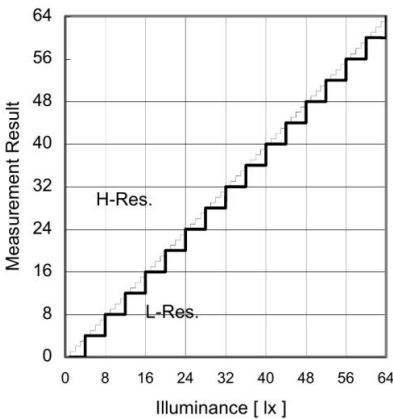


图 5-7 BH1750 实际光照值与测量值对比

Fig.5-7 BH1750 actual light value and measured value comparison

如图 5-7 是此模块实际光照值和测量光照值结果的对比，精度非常高。此模块采用 I2C 通讯，通讯协议如图 5-8 所示：

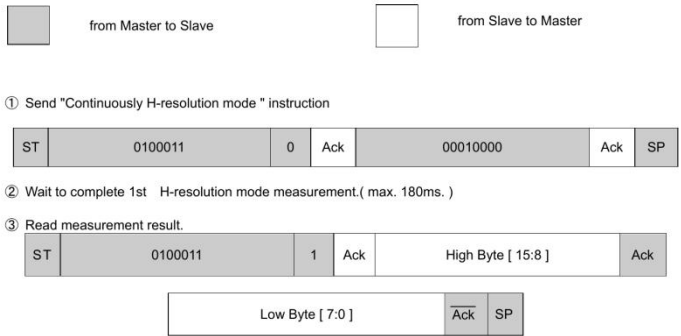


图 5-8 BH1750 通讯协议

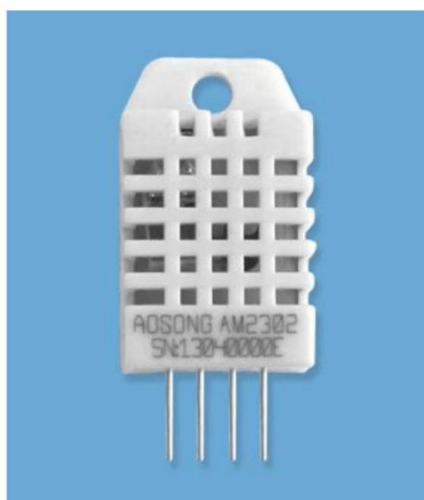
Fig.5-8 BH1750 communication protocol

在程序中启用控制器的 I2C 外设，注入寄存器地址，实现上图的协议即可取出 BH1750 的测量值，部分代码如下：

```
int BH1750_Read(int address) {
    int i=0;
    Wire.beginTransmission(address);
    Wire.requestFrom(address, 2);
    while(Wire.available()) {
        buff[i] = Wire.receive();
        i++;
    }
    Wire.endTransmission();
    return i;
}
```

对取出的二进制数据，还原成十进制数据并带入光照值计算公式即可得出光照值。

5.2.3 温湿度传感器硬件及软件设计



引脚	名称	描述
①	VDD	电源 (3.3V~5.5V)
②	SDA	串行数据，双向口
③	NC	空脚
④	GND	地

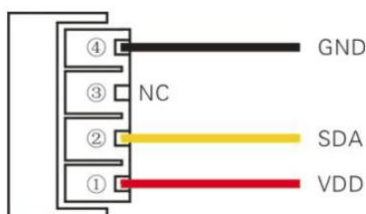


图 5-9 AM2302 引脚说明

Fig.5-9 AM2302 pin description

DHT22 温湿度传感器模块内部主要的传感器是 AM2302 湿敏电容数字温湿度模块，它是一款含有已校准数字信号输出的温湿度符合传感器。传感器内部包含一个高精度测温元件和一个电容式感温元件，并且内置一个高精度 8 位单片机，用于获取敏感元件的数据并且与外界的单片机关通信。

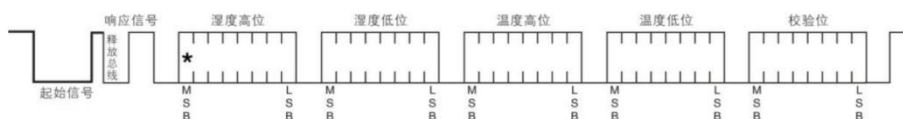


图 5-10 AM2302 单总线通信协议

Fig.5-10 AM2302 single bus communication protocol

AM2302 采用单总线通信方式（即只有一根数据线），一次传出 40 位数据，高位先出。根据其通信协议，在程序中实现通信协议便可以得到数据。

```
void RH(void){
    //主机拉低 18ms
    P2_0=0;
    Delay(180);
    P2_0=1;
    //总线由上拉电阻拉高 主机延时 20us
    Delay_20us();
    //主机设为输入 判断从机响应信号
    P2_0=1;
    //判断从机是否有低电平响应信号 如不响应则跳出，响应则向下运行
    if(!P2_0){
        U8FLAG=2;
        //判断从机是否发出 80us 的低电平响应信号是否结束
        while((!P2_0)&&U8FLAG++);
        U8FLAG=2;
        //判断从机是否发出 80us 的高电平，如发出则进入数据接收状态
        while((P2_0)&&U8FLAG++);
        //数据接收状态
        COM();
        U8RH_data_H_temp=U8comdata;
        COM();
        U8RH_data_L_temp=U8comdata;
        COM();
        U8T_data_H_temp=U8comdata;
        COM();
        U8T_data_L_temp=U8comdata;
        COM();
        U8checkdata_temp=U8comdata;
        P2_0=1;
        //数据校验
        U8temp=(U8T_data_H_temp+U8T_data_L_temp+U8RH_data_H_temp+U8RH_data_L_temp);
        if(U8temp==U8checkdata_temp) {
            U8RH_data_H=U8RH_data_H_temp;
```

```

    U8RH_data_L=U8RH_data_L_temp;
    U8T_data_H=U8T_data_H_temp;
    U8T_data_L=U8T_data_L_temp;
    U8checkdata=U8checkdata_temp;
}
}
}

```

5.2.4 语音模块硬件及软件设计

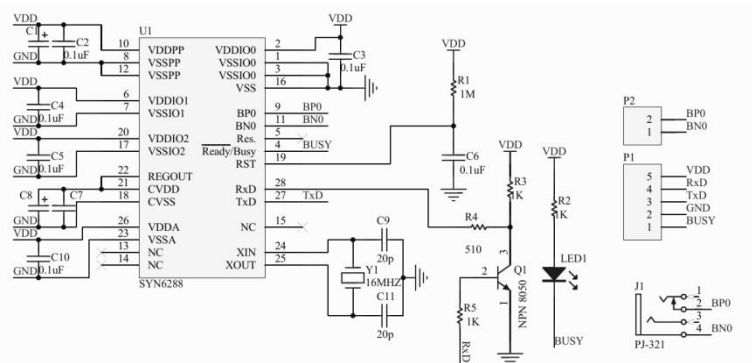
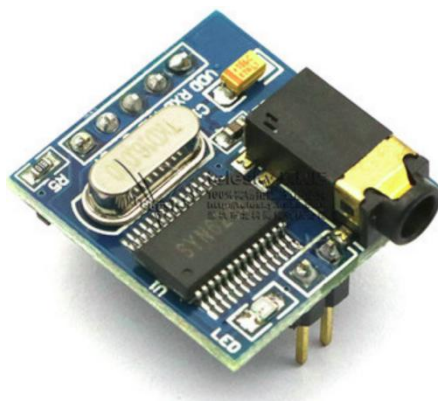


图 5-11 语音模块原理图

Fig.5-11 Voice module schematic

SYN6288 中文语音合成芯片是北京宇音天下科技公司推出的产品，通过异步串口方式通讯，接收文本数据，生成对应的语音。

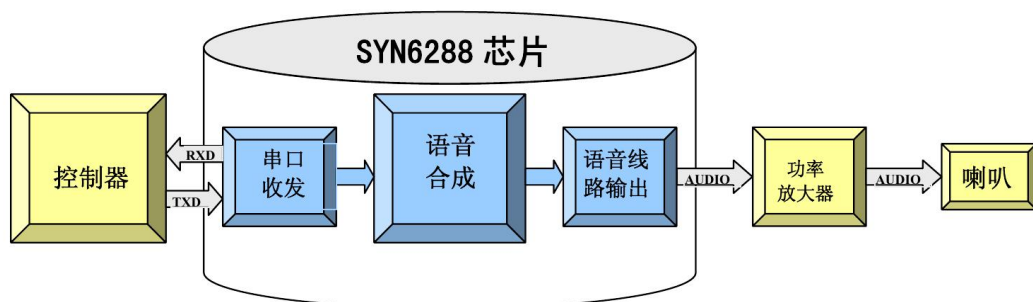


图 5-12 语音模块构成框图

Fig.5-12 Speech module block diagram

图 5-12 是整个模块的构成框图。单片机只需要通过串口和模块进行通讯即可。把想要合成语音的文本放到数组里面，实现通信协议即可。

帧结构	帧头 (1字节)	数据区长度 (2字节)	数据区 (小于等于203字节)			
			命令字 (1字节)	命令参数 (1字节)	待发送文本 (小于等于200字节)	异或校验 (1字节)
数据	0xFD	0xFF 0xFF	0xFF	0xFF	0xFF . . .	0xFF
说明	定义为十六进制“0xFD”	高字节在前 低字节在后	长度必须和前面的“数据区长度”一致			

图 5-13 SYN6288 通信协议

Fig.5-13 SYN6288 communication protocol

在代码中实现通信协议,如下:

```
void SYN_FrameInfo(uint8_t Music,uint8_t *HZdata){
/*****需要发送的文本*****/
    unsigned char  Frame_Info[50];
    unsigned char  HZ_Length;
    unsigned char  ecc  = 0;          //定义校验字节
    unsigned int i=0;
    HZ_Length =strlen(HZdata);        //需要发送文本的长
/*****帧固定配置信息*****/
    Frame_Info[0] = 0xFD ;           //构造帧头 FD
    Frame_Info[1] = 0x00 ;           //构造数据区长度的高字节
    Frame_Info[2] = HZ_Length + 3;    //构造数据区长度的低字节
    Frame_Info[3] = 0x01 ;           //构造命令字：合成播放命令
    Frame_Info[4] = 0x01 | Music<<4 ; //构造命令参数：背景音乐设定
/*****校验码计算*****/
    //依次发送构造好的 5 个帧头字节
    for(i = 0; i<5; i++)  ecc=ecc^(Frame_Info[i]);    //对发送的字节进行异或校验
    //依次发送待合成的文本数据
    for(i= 0; i<HZ_Length; i++)  ecc=ecc^(HZdata[i]); //对发送的字节进行异或校验
/*****发送帧信息*****/
    memcpy(&Frame_Info[5], HZdata, HZ_Length);
    Frame_Info[5+HZ_Length]=ecc;
    PrintCom(Frame_Info,5+HZ_Length+1);
}
```


5.2.5 esp8266 模块硬件和软件设计

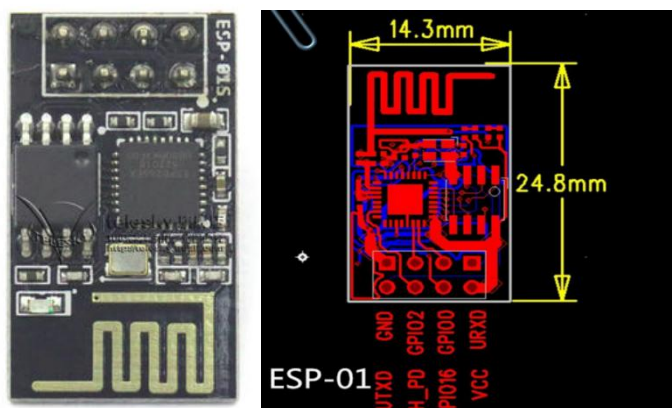


图 5-14 esp8266 尺寸和引脚说明

Fig.5-14 Esp8266 size and pin description

Esp8266 系列模块是非常常用的 WiFi 模块，由深圳市安信可科技有限公司开发，基于乐鑫 ESP8266 的超低功耗的 WiFi 模块，支持串口通信，大大简化了通信的复杂度，提高了稳定度。内置 TCP/IP 协议栈，支持多路 TCP Client 连接；内置 32 位 MCU，而且引脚资源丰富，可兼做处理器用。

Esp8266 通过 AT 指令便可以完成配置，可以用串口调试工具进行配置，也可以在单片机中使用程序直接完成配置。

命令类型	语法	返回和说明
设置命令	AT+CWMODE = <mode>	OK
		此指令需重启后生效(AT+RST)
查询命令	AT+CWMODE?	+CWMODE: <mode>
		OK
测试命令	AT+CWMODE?	当前处于哪种模式?
		+CWMODE: (<mode>取值列表)
		OK
		当前可支持哪些模式?

图 5-15 esp8266 部分 AT 指令集

Fig.5-15 Esp8266 part of the AT command set

实现 AT 指令配置部分代码如下：

```
void setup(){
    Serial.begin(115200);
    WiFi.persistent(false);
    WiFi.begin(getenv("STA_SSID"), getenv("STA_PASS"));
    while (WiFi.status() != WL_CONNECTED)    delay(500);
    MDNS.begin("esp8266-wfs-test");
```


BS_RUN(Serial);

5.2.6 远端执行器硬件及软件设计

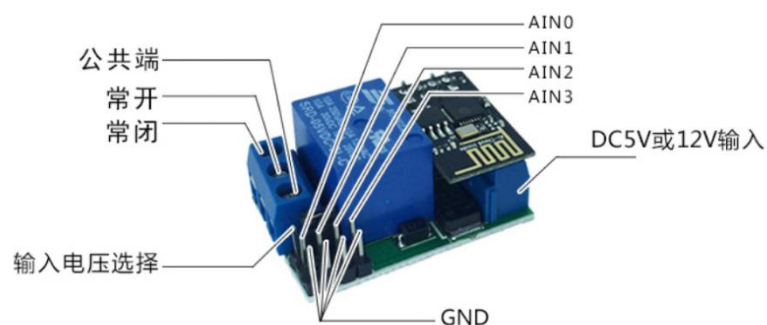


图 5-16 模拟各类控制器使用的继电器

Fig.5-16 Simulate relays used by various controllers

我们为了模拟要控制的不同设备，使用继电器作为执行器模拟。上面同样搭载一个 esp8266 模块作为客户端，和单片机相连的 esp8266 作为服务器，服务器向客户端发送命令，继电器根据接收到的指令进行开合动作。

6 结论与展望

家庭智能服务机器人未来有可能会像 PC 和手机一样，进入寻常百姓家，达到非常高的普及度。目前各国很多公司都在布局这一领域，中国在该领域也有很多做的很好的公司。下一步的发展方向当然是向通用化、高度智能化发展。

本文所设计的机器人中使用的 SLAM 技术也是一个非常重要的方向，是实现机器人自主导航很重要的一环，也是目前自动驾驶汽车、无人机等广泛使用和不断开发的技术。随着数学的发展、技术的不断更新和发展，SLAM 的性能还会有很大的发展，深度学习、神经网络的应用也会助力 SLAM 的发展，语义 SLAM 是很有潜力的一个方向。相比目前 SLAM 粒子滤波、图优化等形式的 SLAM 实现方式，语义 SLAM 的实现更加符合人类的认知规律，同样也可以达到更高的鲁棒性和更好的效果。

语义 SLAM 处在刚起步的阶段，未来几年激光和视觉 SLAM 的融合也可以达到很好的效果。在低成本领域，单目 SLAM 的研究领域异常活跃，得益于单目视觉的低成本优势，所以单目 SLAM 方向也是很有应用价值的技术。

在人体姿势识别方面，国外的高校和企业都有很活跃的社区和专业的实验室，他们很多在 [github](#) 上都有上万的星数。国外在这方面从很早就开始深耕，而且大多数体感相机的厂家和技术提供厂商都来自美国。

国内小米公司从很早就开始布局智能家居，现在小米有全套的智能家具设备，而且性价比非常高。华为和京东前段时间也开始向此领域进军。能够打造一套成本低、智能、人性化的智能家居设备未来将会有非常大的市场。

参考文献

- [1] 桂圆圆.家庭服务机器人发展现状、专利分析及发展趋势[J].科技风,2018(22):5.
- [2] 田红瑞. 轮式家庭服务机器人导航的研究[D].东南大学,2015.
- [3] 陈骏. 移动机器人通用底盘设计与研究[D].杭州电子科技大学,2012.
- [4] 姚二亮,张合新,宋海涛,张国良.基于语义信息和边缘一致性的鲁棒 SLAM 算法[J/OL]. 机器人:1-10[2019-05-31].<https://doi.org/10.13973/j.cnki.robot.180697>.
- [5] Gastón Castro, Matías A. Nitsche, Taihú Pire, Thomas Fischer, Pablo De Cristóforis. Efficient on-board Stereo SLAM through constrained-covisibility strategies[J]. Robotics and Autonomous Systems, 2019, 116.
- [6] Babak Shabani, Jason Lavroff, Michael R. Davis, Damien S. Holloway, Giles A. Thomas. Slam loads and pressures acting on high-speed wave-piercing catamarans in regular waves[J]. Marine Structures, 2019, 66.
- [7] Lucas de Paula Veronese, Claudine Badue, Fernando Auat Cheein, Jose Guivant, Alberto Ferreira De Souza. A single sensor system for mapping in GNSS-denied environments[J]. Cognitive Systems Research, 2019, 56.
- [8] Aritra Mukherjee, Satyaki Chakraborty, Sanjoy Kumar Saha. Detection of loop closure in SLAM: A DeconvNet based approach[J]. Applied Soft Computing Journal, 2019, 80.
- [9] Zhe Sun, Yi-chen Jiang, Gui-yong Zhang, Zhi Zong, Jing Tang Xing, Kamal Djidjeli. Slamming load on trimaran cross section with rigid and flexible arches[J]. Marine Structures, 2019, 66.
- [10] S. Prcic. SP-0330 Science slam: To breathe or not to breathe. ESTRO Mobility Grant report[J]. Radiotherapy and Oncology, 2019, 133.
- [11] Lichao Xu, Chen Feng, Vineet R. Kamat, Carol C. Menassa. An Occupancy Grid Mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments[J]. Automation in Construction, 2019, 104.
- [12] Jun Wang, Jingwei Song, Liang Zhao, Shoudong Huang, Rong Xiong. A submap joining algorithm for 3D reconstruction using an RGB-D camera based on point and plane features[J]. Robotics and Autonomous Systems, 2019.
- [13] W. Shane Grant, Randolph C. Voorhies, Laurent Itti. Efficient Velodyne SLAM with point and plane features[J]. Autonomous Robots, 2019, 43(5).
- [14] Francesco Martinelli. Simultaneous Localization and Mapping Using the Phase of Passive UHF-RFID Signals[J]. Journal of Intelligent & Robotic Systems, 2019, 94(3-4).
- [15] Patrik Schmuck, Margarita Chli. CCM - SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams[J]. Journal of Field Robotics, 2019, 36(4).
- [16] 刘智杰,赵一兵,李琳辉,张溪桐,周雅夫.基于卷积神经网络的语义同时定位以及地图构建方法[J]. 科学技术与工程, 2019, 19(09):148-153.

- [17]胡思旺,李春杰.基于 ROS 的人脸识别系统设计与实现[J].电子技术与软件工程,2019(07):32-34.
- [18]王小华,郑腾,孙伟,苗中华.基于 ROS 平台的智能车运动线控系统设计与实现[J].测控技术,2019,38(04):22-25.
- [19]翁潇文,李迪,柳俊城.基于图优化的二维激光 SLAM 研究[J].自动化与仪表,2019,34(04):31-35.
- [20]王晓华,要鹏超,广夏桐,王文杰,张蕾.基于 ROS 的移动机器人语音控制设计与实现[J].传感器与微系统,2019,38(05):100-103.
- [21]王金戈,邹旭东,仇晓松,蔡浩原.动态环境下结合语义的鲁棒视觉 SLAM[J].传感器与微系统,2019,38(05):125-128+132.
- [22]李玉萍.机器人领域中国专利申请状况分析[J].机器人技术与应用,2019(03):43-48.
- [23]朱灿杰,曹鑫扬,付豪.基于 SLAM 技术的智能避障无人机[J].科技风,2019(13):72.
- [24]李少朋,张涛.深度学习在视觉 SLAM 中应用综述[J].空间控制技术与应用,2019,45(02):1-10.
- [25]张锋,邹风山.送餐服务机器人风险评估[J].电子技术与软件工程,2019(09):195-196.
- [26]孟清,斯琴高娃.基于 ROS 的自主移动服务机器人系统框架设计[J].电子技术与软件工程,2019(09):46-47.
- [27]邱笑晨,赵晨旭,张海,许辉.ORB-SLAM 系统优化框架分析概述[J].导航定位与授时,2019,6(03):45-51.
- [28]曹栋,王吉芳.一种三轮全向移动机器人的设计与实现[J].北京信息科技大学学报(自然科学版),2019,34(02):75-80.
- [29]曾芬芳.基于势场路径规划的机器人同步定位与地图构建[J].中国新技术新产品,2019(08):12-13.
- [30]龚学锐,闵华松.基于 ROS 的惯性和视觉里程计的机器人室内定位[J].组合机床与自动化加工技术,2019(05):106-110.
- [31]钟海兴,许伦辉,郭雅婷.基于 ROS 的激光导航移动机器人控制系统设计[J].自动化与仪表,2018,33(12):30-33.
- [32]张腾,金光,江先亮.基于 ROS 的家庭服务机器人设计[J].数据通信,2018(06):7-9+25.
- [33]吴志鹏,廖志青,冼嘉媚,曾奕雄,何威.基于 ROS 的救援船体机器人控制系统设计[J].装备制造技术,2018(12):18-22.
- [34]周亚丽,张奇志,徐鑫鑫,张万杰,律晔.家庭服务机器人研究与人才培养[J].机器人技术与应用,2019(01):43-48.
- [35]马梅彦.个人/家庭服务机器人用户行为分析[J].机器人产业,2019(01):86-95.
- [36]张美平,郭旭城,张毅韬,王志宇.基于 ROS 的溯源机器人系统[J].计算机系统应用,2019,28(03):43-50.
- [37]陈世浪,吴俊君.基于 RGB-D 相机的 SLAM 技术研究综述[J].计算机工程与应用,2019,55(07):30-39+126.
- [38]王昊,关豪,王鹏飞,张映宏.基于 ROS 系统的摄像头循迹小车设计[J].信息技术与信息化,2019(02):96-98+103.
- [39]杨玉,王政博.家庭服务机器人知识库的构建[J].电子世界,2019(06):103.

- [40]韦佳.居家养老智能服务机器人的设计发展研究[J].中国管理信息化,2018,21(02):117-118.
- [41]丁满,袁梦娟,白仲航,张晓.基于用户无意识行为的服务机器人交互设计[J].包装工程,2018,39(10):91-95.
- [42]桂圆圆.家庭服务机器人发展现状、专利分析及发展趋势[J].科技风,2018(22):5.
- [43]杨凤满,张奇志,周亚丽.家庭服务机器人路径规划的跳点搜索算法[J].北京信息科技大学学报(自然科学版),2018,33(03):85-89.
- [44]张家祺,文豪,周坤.家庭智能服务陪伴机器人[J].包装工程,2018,39(16):290.
- [45]龙浩,陈祥章,霍娜,宋培森.情感认知智能服务机器人产业发展研究[J].计算机产品与流通,2018(02):154.
- [46].智能智造·智慧生活——新松全线新品盛装亮相 2017 中国国际机器人展[J].控制工程,2017,24(07):1526-1527.
- [47]李扬.智能服务机器人引领智慧变革[J].高科技与产业化,2016(05):69-71.
- [48]汤雅丹.家庭服务机器人[J].家庭服务,2016(10):34-35.
- [49]秦晖.轮式家庭服务机器人的移动控制设计与研究[D].电子科技大学,2016.
- [50]韩海丹.面向智能服务机器人的家庭环境关系知识库构建[D].燕山大学,2016.
- [51]李攀.家用服务机器人物品搜寻算法的研究与实现[D].中南民族大学,2015.
- [52]郭雄飞.仿人服务机器人系统研究与设计[D].广东工业大学,2016.
- [53]本报记者 赵广立.以色列机器人公司 Roboteam Home 大中华区 CEO 加尔·戈伦:控制好期待,服务机器人“交互有未来”[N].中国科学报,2019-05-30(007).
- [54]林承志,李勇.机器人检测行业现状及发展[J].质量技术监督研究,2019(02):36-39.
- [55]李栋.家庭服务机器人自主巡迹监视与智能交互物联网技术研究[D].山东大学,2018.
- [56]范恩胜.基于单目视觉的家庭服务机器人导航研究[D].山东建筑大学,2018.
- [57]O.H. Hassoon,M. Tarfaoui,A. El Moumen,Y. Qureshi,H. Benyahia,M. Nachtane. Mechanical performance evaluation of sandwich panels exposed to slamming impacts: Comparison between experimental and SPH results[J]. Composite Structures,2019,220.
- [58]Gastón Castro,Matías A. Nitsche,Taihú Pire,Thomas Fischer,Pablo De Cristóforis. Efficient on-board Stereo SLAM through constrained-covisibility strategies[J]. Robotics and Autonomous Systems,2019,116.
- [59]Oguzhan Guclu,Ahmet Burak Can. k-SLAM: A fast RGB-D SLAM approach for large indoor environments[J]. Computer Vision and Image Understanding,2019.
- [60]Babak Shabani,Jason Lavroff,Michael R. Davis,Damien S. Holloway,Giles A. Thomas. Slam loads and pressures acting on high-speed wave-piercing catamarans in regular waves[J]. Marine Structures,2019,66.
- [61]Aritra Mukherjee,Satyaki Chakraborty,Sanjay Kumar Saha. Detection of loop closure in SLAM: A DeconvNet based approach[J]. Applied Soft Computing Journal,2019,80.
- [62]Zhe Sun,Yi-chen Jiang,Gui-yong Zhang,Zhi Zong,Jing Tang Xing,Kamal Djidjeli. Slamming load on

- trimaran cross section with rigid and flexible arches[J]. Marine Structures,2019,66.
- [63]S. Prcic. SP-0330 Science slam: To breathe or not to breathe. ESTRO Mobility Grant report[J]. Radiotherapy and Oncology,2019,133.
- [64]Lichao Xu,Chen Feng,Vineet R. Kamat,Carol C. Menassa. An Occupancy Grid Mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments[J]. Automation in Construction,2019,104.
- [65]W. Shane Grant,Randolph C. Voorhies,Laurent Itti. Efficient Velodyne SLAM with point and plane features[J]. Autonomous Robots,2019,43(5).
- [66]Patrik Schmuck,Margarita Chli. CCM - SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams[J]. Journal of Field Robotics,2019,36(4).
- [67]. Hit a grand slam for soil conservation[J]. Corn and Soybean Digest,2019.
- [68]Don Wall. Building Trades slam Ontario budget ' s training plan[J]. Daily Commercial News,2019,92(75).
- [69]Anonymous. FFUK slams fuel prices[J]. Commercial Motor,2019,231(5834).
- [70]Trang Hoang Thuy Nguyen,S V Shidlovskiy. Tunable computing Slam navigation environments[J]. IOP Conference Series: Materials Science and Engineering,2019,516(1).

致谢

做毕设的这段时间，学到了很多新的东西，也用到了很多新的非常具有吸引性的硬件，在软件开发、平台使用上也比之前上升了一个层次。往近了说，自从上本科以来，自己不管是在学业上、技术上、交往上等各方面的变化自己在心里都很清楚，抛开自己的努力不说，首先这些所有的进步的取得离不开一个优良的环境，有优良的各方面的条件，所以首先要感谢我们的祖国，虽然她不是那么完美，但是她在努力给我们提供便宜的、优秀的教育环境和条件，给我们一个安宁的、美丽的生活环境。其次我要感谢我的父母，他们努力给我提供了一个不算富裕但至少我可以展翅飞翔的条件，他们会给我不断的鼓励和支持，支持我的学业，支持我的事业，支持我喜欢的并且正在做的事情，我和他们之间从来都是那种很活跃的、自由的、随性的交流，他们都有很开放的、与时俱进的思想，对我性格也影响了很多。我要突出感谢我的本科生导师王宪伦教授，从大二开始就一直跟着他做项目，他是那种不怎么干涉你脚步但又会给你前进动力的那种人，给我们提供一个方便的实验室，设备完善、成员之间有合作有竞争、学术氛围浓厚专注，也正是王宪伦教授本人的作为和努力，培养了一大批很优秀的学生，不管在校内还是校外。我还要感谢我的女友王敏女士，她陪我天天泡实验室，考研全程都在教室一直陪着我，做毕设期间我没有陪她出去旅游、没有出去踏青、没有周末，我经常开玩笑跟她说我们错过了整个春天，我对她其实挺愧疚，相比于多数女生出去旅游、吃大餐包括其他活动的机会，她要少的多。最后我要感谢开源社区，感谢那些为了技术发展、技术自由、技术分享在默默努力和付出的那些高尚的人，没有你们，现在的技术不会这么发达、这么繁荣、这么遍布世界，也不会有这么多商业公司赚的盆满钵满，如果我自身能力足够而又有机会的话，我一定是你们中的一员。

文岩

2018年5月于青岛