

移动机器人使用指导书

(V3.0 版)



哈尔滨工业大学（深圳）
实验与创新实践教育中心

2023 年 9 月

目 录

第一章 移动机器人系统介绍.....	3
1.1 系统硬件构架.....	3
1.1.1 整机性能参数.....	4
1.1.2 电机.....	5
1.1.3 陀螺仪 Imu 模块	6
1.1.4 超声波模块.....	6
1.1.5 STM32 模块	6
1.1.6 工控机（可选用）	6
1.1.7 力策激光雷达.....	7
1.1.8 奥比中光深度摄像头.....	7
1.1.9 ZED 摄像头.....	7
1.1.10 RealSense 摄像头（实验使用 D435）	8
1.2 软件架构介绍.....	8

第一章 移动机器人系统介绍

1.1 系统硬件构架

移动机器人系统主要由电脑、STM32、双目摄像头、激光雷达、陀螺仪等组成，如图 1.1 所示。电脑作为上位机总控单元，可以完成系统的导航、避障、通信等功能，并作为显示及操作机构，完成程序的修改与装载、导航的显示、航迹路径规划等功能；路由器完成与其他机器人的通信，每台车都有自己的局域网 WIFI，实验时需要选择好对应的车；STM32 作为系统的底层控制，完成采集陀螺仪、超声波、左右轮编码器等传感器数据，以及控制电机转动的功能；陀螺仪用于测量车转动的角度，超声波主要用于检测玻璃等透明障碍物。

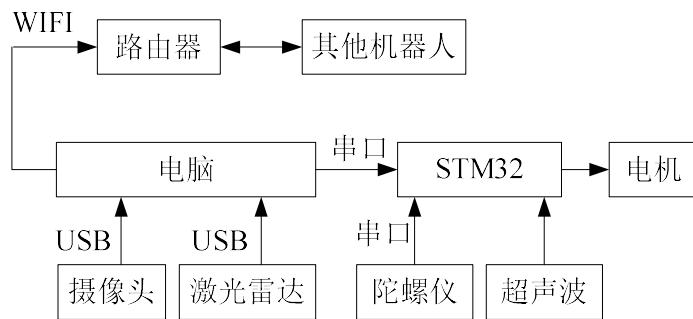


图 1.1 系统硬件构架框图

注意：所有模块(包括 STM32 内部)是使用 TTL 串口通讯。

◆ STM32 模块作用

采集陀螺仪，超声波，左右轮电机编码器等传感器数据，并且控制电机的转动。

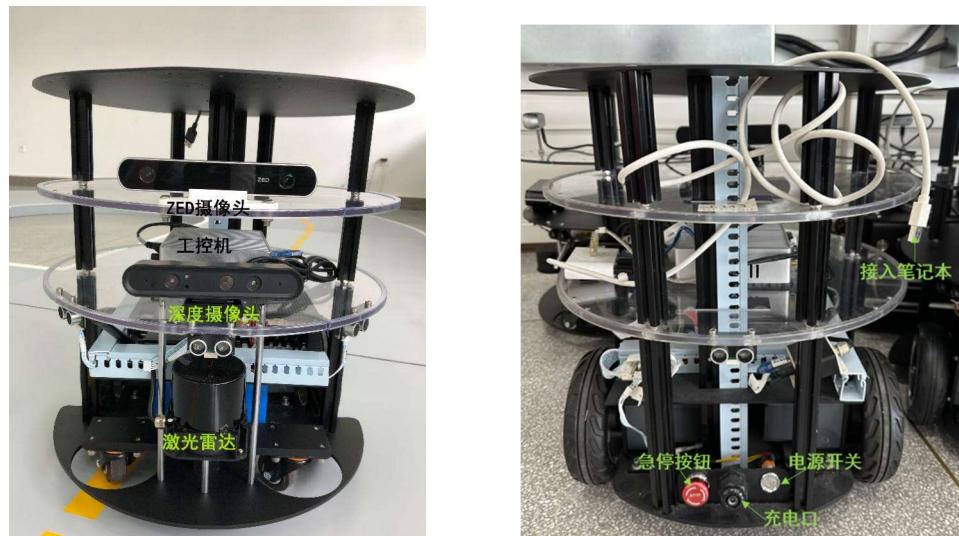
◆ 笔记本电脑作用

1.通过串口与 STM32 模块通讯，获取陀螺仪，超声波，编码器等数据，并转换成相应 ROS 的消息，例如陀螺仪数据转换成 ROS 的 sensor_msgs/Imu 消息，超声波数据转成 ROS 的 sensor_msgs/Range，左右轮编码器数据转换成 ROS 的里程计消息 nav_msgs/Odometry。

2.获取激光雷达数据，结合里程计数据，用 gmapping 算法构建地图。

3.运行 navigation 导航避障算法，结合雷达，超声波，深度摄像头的数据，在导航移动过程中避开障碍物，到达指定目标点。

本课程以移动机器人系统（图 1.2 所示）为主要设备展开实验项目的研究。



移动机器人实物图（ZED 摄像头版本）

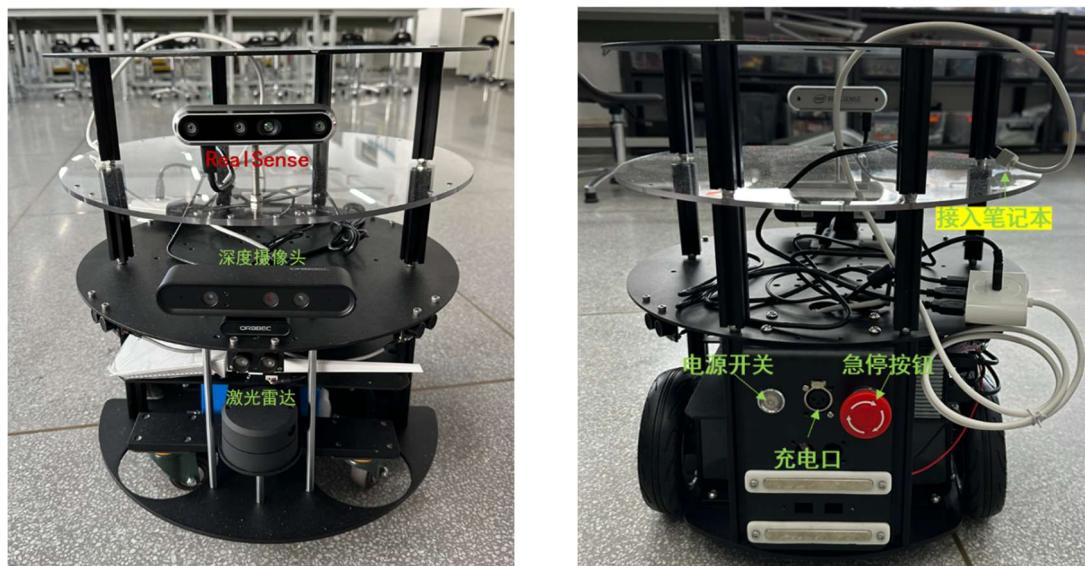


图 1.2 移动机器人实物图（RealSense 版本）

1.1.1 整机性能参数

关键技术参数	
最大运行速度	1.0 m/s
巡航时间	约 6 小时
最大爬坡	15°
越障高度	10mm
建图定位	支持
路径规划	支持
手机 APP	支持
自动回充	不支持
载重	50Kg
电气接口	
电源接口	5V 输出（供电） USB.5V*2

	24V 输入（充电）	
通讯接口	USB.UART	接导航模块
	Bluetooth	底盘内置蓝牙
电池参数		
电池类型	动力锂电池	
电池容量	24V 12AH	
充电时间	约 5 小时	
充电电压	24V	
充电电流	3A	
保护功能	支持过流、过放、短路保护	

1.1.2 电机

表 1.1 电机参数表

额定功率	60W
额定电压	24 V
电压容许范围	$\pm 10\%$
额定转速	2500 rpm
额定转矩	0.23 N.m
瞬时最大转矩	0.345 N.m
转子转动惯量	$0.082 J : X 10^{-4} \text{kg.m}^2$
速度控制范围	200~2500 rpm

电机驱动框图如图 1.3 所示。

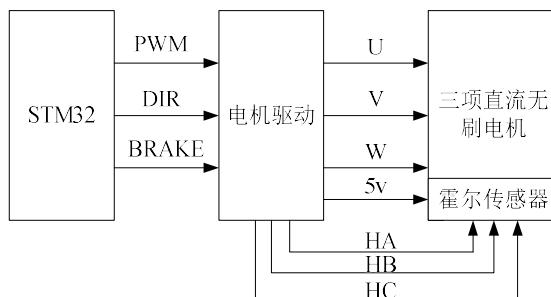


图 1.3 电机驱动框图

连线说明：

PWM：通过占空比控制电机转速

BRAKE：控制刹车

DIR：控制转动方向

U、V、W：无刷电机的三相引出线

5V、GND：5V 电源和地

HA、HB、HC：霍尔传感器信号反馈线，其中 HA、HB 相也作为码盘反馈使用电机速度计算：

电机速度测量采样 M 法，即测量单位时间内的脉数。在单位时间 T 内，统计 HA、HB 的所有上升沿和下降沿得到脉冲数 Count。已知电机转换一周可计数的脉冲个数为 one_cycle_count（目前使用的这款电机为 610）。令单位时间 T 的单位为秒，则计算出速度为 Count/T，其单位为脉冲数/秒。注意，实际程序里的单位时间 T 是毫秒级的，计算时需转换成秒。若需要的速度单位为转每秒，则速度计算公式为 Count/(T*one_cycle_count)。若需再将速度转换为转每分钟，则需乘以 60，即(60*Count)/(T*one_cycle_count)。

1.1.3 陀螺仪 Imu 模块

陀螺仪 Imu 模块	
类型	CruizCore R6093U 6 轴加速度计，测试角度精度为 0.01 度
作用	测试机器人转动的角度，精度较高，使机器人转动更准确

1.1.4 超声波模块

超声波模块	
类型	US.015，测量角度为 15 度(圆锥形)，测量距离范围为 2cm-400cm，测量精度为 0.1cm+1%
作用	主要用于检测玻璃等透明障碍物(激光雷达和深度摄像头无法看到的物体)，并临时加入地图中，使导航重新规划路径避开这些障碍物

1.1.5 STM32 模块

STM32 模块	
类型	主控 MCU 为 STM32F407ZGT6，工作频率 168MHZ；1 MB Flash；92 kB RAM；封装：LQFP144
作用	采集陀螺仪，超声波，左右轮电机编码器等传感器数据，并且控制电机的转动；

1.1.6 工控机（可选用）

工控机	
类型	酷睿 i5.5250U CPU 集成英特尔® 酷睿 i5 5 代系列 1.6G GHz 2.7G 双核 CPU，功耗 15W， 内存是双通道 DDR3 4GB， 显卡是集成 Intel® HD Graphics 6000

	存储器是 60GB 高速 eMMCMicroSD (TF) Card Slot, 电源 DC12V 2A
--	---

1.1.7 力策激光雷达

力策激光雷达	
类型	LITRA ME01, 测距原理是脉冲 TOF 角分辨率为 0.18 度 探测最大距离为 30m 测距精度为±2 cm
作用	用于测试周围环境，相对机器人哪个角度多少距离有障碍物，根据这些障碍物信息，用 gmapping slam 建图算法构建地图，并在导航时规划好路径避开障碍物

1.1.8 奥比中光深度摄像头

奥比中光深度摄像头	
类型	奥比中光 mini 测距范围是 0.4m-2m 角度范围是向上 16 度仰角，向下 24 度俯角， 左右各 24.5 度广角 最大采样频率是 25Hz
作用	测试周围环境相对机器人哪个角度多少距离有障碍物，作用与激光雷达相似，但激光雷达是二维的，只能测试到一个平面 360 度的信息，而深度摄像头是三维的，可以测试三维立体信息，增加机器人安全性和适应性

1.1.9 ZED 摄像头

ZED 是一种可以再现人类视觉工作方式的相机，使用其两个“眼睛”（双镜头）并通过三角测量，提供对其观察到的场景的三维理解，使你的应用成为空间和运动感知。它可以捕获具有宽视野的高清 3D 视频，并在 USB 3.0 上以并排格式输出两个同步的左右视频流。ZED 是第一个通用深度传感器：

- ✧ 深度可以在更长的范围内捕获，最长可达 20 米（在 ULTRA 模式下可扩展至 30 米）。
- ✧ 深度捕获的帧速率可高达 100 FPS。
- ✧ 视场更大，高达 90° (H) x 60° (V)。
- ✧ 相机在室内和室外工作，与有源传感器（如结构光或飞行时间）相反。

ZED 不断扫描其环境以重建现实世界的 3D 地图。它通过结合新的深度和位置数据来改进其对世界的理解。空间映射可通过 ZEDfu 应用程序或 ZED SDK 获得。详情介绍和使用方法见官网 <https://www.stereolabs.com>。

1.1.10 RealSense 摄像头 (实验使用 D435)

规格		
	英特尔实感深度摄像头 D415	英特尔实感深度摄像头 D435
环境	室内和室外	室内和室外
深度技术	主动红外 (IR) 立体	主动 IR 立体
英特尔® 实感™ 摄像头主要组件	英特尔® 实感™ 视觉处理器 D4 英特尔® 实感™ 模块 D410	英特尔® 实感™ 视觉处理器 D4 英特尔® 实感™ 模块 D430
深度视野 (Field of View, 或 FOV)—(横向 x 纵向 x 对角线)	69.4° x 42.5° x 77°	91.2° x 65.5° x 100.6°
深度流输出分辨率	达 1280 x 720	达 1280 x 720
深度流输出帧速率	达 90 fps	达 90 fps
最小深度距离 (Min-Z)	0.3 米	0.2 米
红外信号发射器功率	可配置至达 440 毫瓦 (mW) 峰值	可配置至达 425 毫瓦 (mW) 峰值
图像传感器类型	卷帘快门	全局快门
最大范围	10 米及更远 随校准、场景和光照条件而变	10 米及更远 随校准、场景和光照条件而变
RGB 传感器分辨率和帧速率	30 fps 时为 1920 x 1080	30 fps 时为 1920 x 1080
RGB 传感器视野 (横向 x 纵向 x 对角线)	69.4° x 42.5° x 77°	69.4° x 42.5° x 77°
摄像头尺寸 (长度 x 厚度 x 高度)	99 毫米 x 20 毫米 x 23 毫米	90 毫米 x 25 毫米 x 25 毫米
输入/输出	USB 3.0: 用于主机系统连接 外部传感器同步连接器: 用于与其他外部摄像头同步	USB 3.0: 用于主机系统连接 外部传感器同步连接器: 用于与其他外部摄像头同步
挂接机构	一个 1/4-20 UNC 螺纹挂接点 两个 M3 螺纹挂接点	一个 1/4-20 UNC 螺纹挂接点 两个 M3 螺纹挂接点

开发资源软件下载连接: <https://github.com/IntelRealSense/librealsense>

1.2 软件架构介绍

移动机器人 ROS 工程主要在 dashgo_ws 中，其结构如下：

执行命令: `~/dashgo_ws/src/dashgo$ tree`

```

├── astra //奥比中光摄像头 ROS 驱动包
|   ├── ros_astra_camera-filterlibrary
|   |   └── 56-orbbec-usb.rules //串口规则文件，为了能够识别摄像头串
|   |       口，应存放在电脑的/etc/udev/rules.d/目录下
|   └── ros_astra_launch-master
|       ├── CMakeLists.txt
|       └── launch
|           └── astra.launch //如果想单独测试摄像头，可以启动该

```

```

        launch 文件

|   └── package.xml

|   └── dashgo_description //机器人底盘模型文件, 用于 rviz 显示机器人
|       ├── CMakeLists.txt
|       └── launch
|           └── dashgo_description.launch
|   └── package.xml

|   └── urdf //描述机器人的组成, 模拟的机器人模型
|       ├── dashgobase
|       |   ├── base.urdf.xacro
|       |   ├── dashgobase.xacro
|       |   ├── materials.urdf.xacro
|       |   └── torso.urdf.xacro
|       └── readme.txt

|   └── dashgo_driver //底盘的 STM32 驱动包
|       ├── CMakeLists.txt
|       └── config //参数配置文件
|           ├── my_dashgo_params_imu.yaml //底盘的参数配置文件, 如串口
|           └── yocs_velocity_smoother.yaml //设置行走的最大线速度和角速
|               度, 具体查看平滑减速包 yocs_velocity_smoother
|       └── launch
|           └── driver_imu.launch //底盘驱动 launch 文件

|   └── nodes
|       └── dashgo_driver.py //底盘驱动节点程序

|   └── package.xml

|   └── dashgo_nav //launch 文件和配置文件的集合, 建图和导航等都在其中
|       ├── CMakeLists.txt
|       └── config //参数配置文件中的具体参数含义可参考博客
|           https://blog.csdn.net/fidaji/article/list/1
|           └── imu
|               ├── base_global_planner_param.yaml //全局规划配置文件
|               └── base_local_planner_params.yaml //默认的局部规划配置文
|                   件, 目前没使用
|           └── costmap_common_params.yaml //通用的代价地图参数, 全
|               局和局部都使用
|           └── global_costmap_params.yaml //全局专用的代价地图参数

```

```

|   |   |   └── local_costmap_params.yaml //局部专用的代价地图参数
|   |   └── move_base_params.yaml    //move_base 参数
|   └── teb_local_planner_params.yaml //局部路径规划参数，目前
|                                在用
|   └── launch
|       └── gmapping_imu.launch //建图时启动，包含了底盘驱动、雷达
|                                驱动、底盘 rviz 模型、gmapping 算法
|   └── include
|       └── imu
|           ├── amcl.launch.xml //amcl 的 launch 启动文件和参数
|           ├── gmapping_base.launch //gmapping 算法
|           └── teb_move_base.launch //底盘 move_base 导航启动文件
|                                和参数
|   └── navigation_camera_imu.launch //启动带有摄像头的 launch，只
|                                是用到深度点云数据，识别障碍物并避开
|   └── navigation_imu.launch //使用激光雷达避障
|   └── maps //地图文件
|       ├── eai_map_imu.pgm
|       └── eai_map_imu.yaml
|   └── package.xml
└── dashgo_rviz //rviz 包
    ├── CMakeLists.txt
    └── launch
        └── view_navigation.launch
    └── package.xml
    └── rviz
        └── view_navigation.rviz
└── dashgo_tools //工具包，键盘控制等放在其中
    ├── action
    |   └── check_msg.action
    ├── CMakeLists.txt
    └── conf
        ├── box_filter_2.yaml
        └── box_filter.yaml
    └── package.xml
    └── scripts

```

```
|   |   └── check_action.py  
|   |   └── check_angular_imu.py //转 360 度脚本  
|   |   └── check_linear_imu.py //走一米直线脚本  
|   |   └── teleop_twist_keyboard.py //键盘控制  
└── ltme01_driver-1.0.1 //力策雷达包  
    ├── CMakeLists.txt  
    └── launch  
        └── ltme01.launch  
└── navigation-kinetic-devel //导航避障功能集合  
    ├── amcl //蒙特卡罗定位算法包  
    ├── costmap_2d //生成代价地图功能包  
    ├── dwa_local_planner //局部路径规划功能包  
    ├── global_planner //全局路径规划功能包  
    ├── map_server //提供地图的管理服务  
    ├── move_base //导航核心包  
    │   └── src  
    │       ├── move_base.cpp  
    │       └── move_base_node.cpp  
    ├── nav_core //提供插件的模板  
    ├── robot_pose_ekf //综合里程及、GPS、imu 数据  
    └── teb_local_planner//局部路径规划包
```