

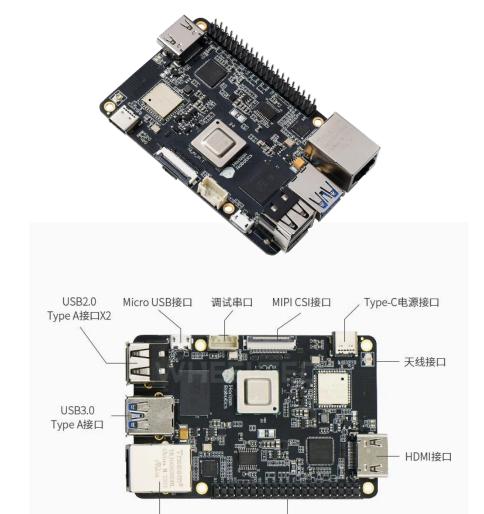
# OriginBot操作体验











40 PIN接口

以太网接口

地平线RDK X3是一款高性能的嵌入式开发板,能够提供出色的计算性能和能效比。它还拥有丰富的接口和扩展槽,支持多种外设和传感器,可以满足机器人、无人驾驶、智能家居等领域的应用需求。

除了高性能和丰富的接口外,地平线RDK X3还配备了TogetheROS开发平台,这是一个专门为机器人开发而设计的开发平台,可以帮助开发者构建高效、可靠的机器人应用程序。通过使用TogetheROS,开发者可以快速地构建机器人应用程序,并轻松地利用社区中已经构建好的模块和库来简化开发过程。

地平线RDK X3在**机器人开发领域**拥有广泛的应用前景,它可以用于开发机器人控制系统、机器人视觉系统、机器人导航系统、机器人智能算法等多个方面,能够让生态开发者在机器人领域中实现更加丰富和高效的创新和发展。



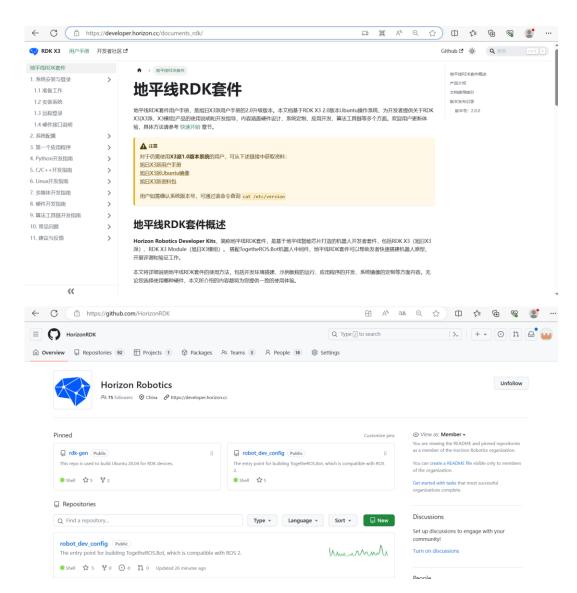


开发板	树莓派4B	Jeston Nano	RDK X3
CPU	4核A72	4核A57	4核A53
内存	2GB/4GB/8GB	2GB/4GB	2GB/4GB
AI引擎	无	CPU/CUDA	BPU
算力	0.2T (FP16)	0.5T (FP16)	5T (INT8)
编解码	1080P@30fps	4K@30fps	4K@60fps
扩展接口	USB/ETH/CSI/HDMI		
功耗	Max 15W	Max 20W	Max 10W
操作系统	Ubuntu 20.04	Ubuntu 18.04	Ubuntu 20.04
机器人扩展框架	ROS	ROS、Isaac SDKROS、Isaac SDK	ROS、TogetheROS

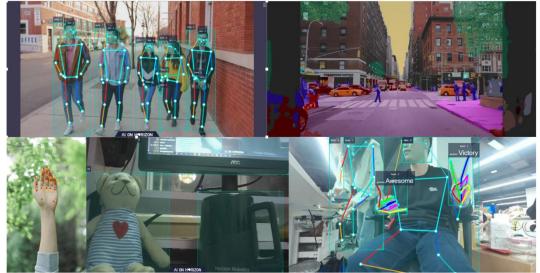
### 丰富资料 开源开放











# • 嵌入式产品介绍





#### 智能会议













































#### 智能音箱

















# 人机交互解决方案



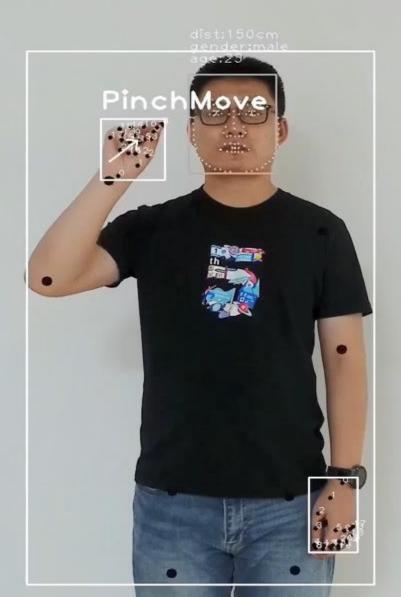




地平线旭日® 3 人机交互解决方案

Horizon sunrise 3 human computer interaction solution







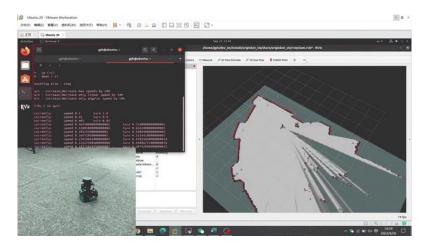
# 机器人开源套件——OriginBot, 快速上手智能机器人开发



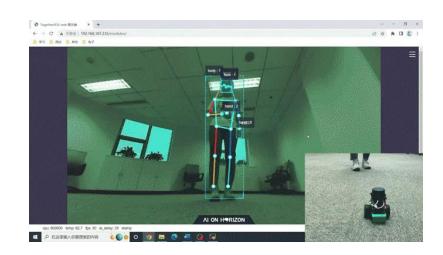


一、如何设计一款智 能小车	二、如何实现智能小 车的机电设计	三、如何开发智能小 车的控制器软件
1. 总体设计思路 2. 机械结构设计 3. 驱动系统设计 4. 传感系统设计 5. 控制系统设计	1. 机电系统的总体设计 2. 嵌入式最小系统设计 3. 电机驱动电路设计 4. 姿态传感器电路设计 5. 电源管理系统设计 6. PCB布局走线	1. 控制器软件总体架构 2. 嵌入式开发环境搭建 3. I/O控制 4. AD模数转换 5. 编码器测速 6. 电机闭环控制
	7. 电路板打样与生产	7. 串口通信与IMU驱动

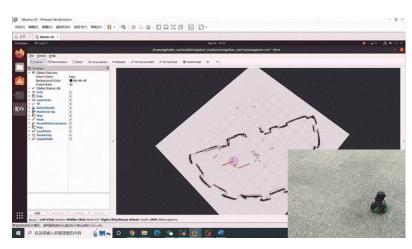
四、如何实现智能小	五、如何构建智能小	六、如何开发智能小
车的通信协议	车的ROS驱动	车的应用功能
1. 通信协议设计 2. 通信协议的下位机实现 3. 通信协议的上位机实现	1. ROS驱动的设计架构 2. ROS开发环境配置 3. 速度控制话题的订阅 4. 里程计话题与TF的维护 5. IMU话题的发布 6. 机器人状态的动态监控	1. 运动控制 2. 视觉巡线 3. 人体跟随 4. SLAM地图构建 5. 自主导航



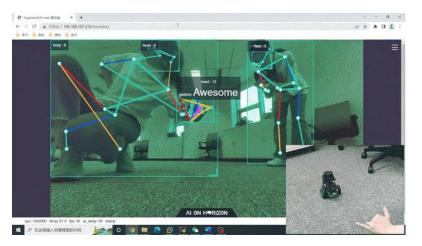
地图构建SLAM



人体跟随



自主导航



手势控制

详细介绍: https://www.originbot.org/

### • 硬件组装

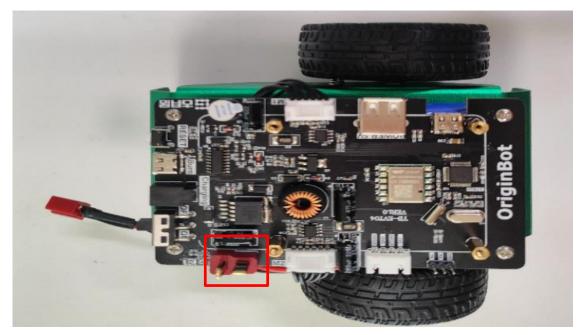




#### http://originbot.org/guide/hardware setup/

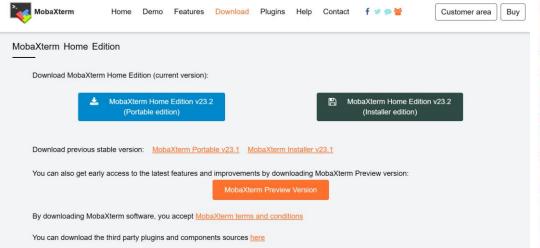


#### 建议先把电源线接上,但不要上电



#### 安装rufus, mobaxterm







# • 烧写镜像





- 1.解压智能小车系统镜像
- 2.按照软件配置的镜像安装进行 (rufus)



# • 常用开发方法介绍





#### 串口通信

- 按位 (bit) 发送和接收字节
- **波特率**:单位时间内载波参数变化的次数,代表传输 速率
- **数据位**:通信中实际数据位的个数,一般为6、7或8bit
- **停止位**:用于表示单个包的最后一位。典型的值为1, 1.5和2位
- **奇偶校验位**:在串口通信中一种简单的检错方式。对于偶和奇校验的情况,串口会设置校验位(数据位后面的一位),用一个值确保传输的数据有偶个或者奇个逻辑高位



#### SSH安全外壳协议

- 是一种在不安全网络上用于安全远程登录和其他安全网络服务的协议。
- 通过网络的方式访问设备,与串口通信方式相比拥有更大的带宽和自由度
- 使用方式: ssh username@IP
- 举例:ssh root@192.168.1.10



# ■ RDK X3系统配置

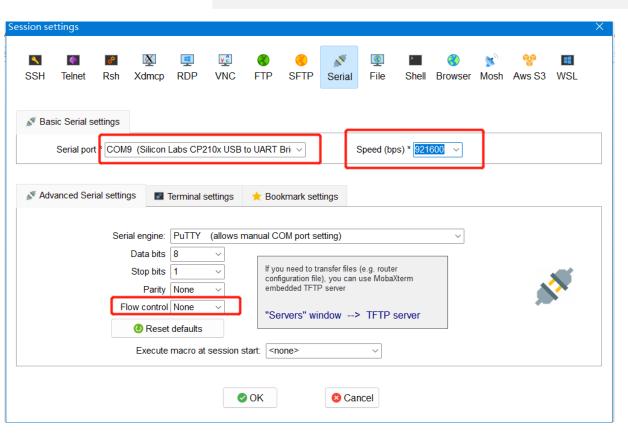


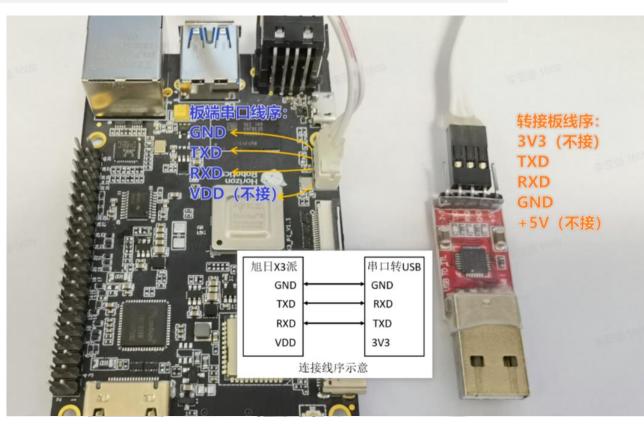


#### Quick View安装 串口驱动和MobaXterm工具 https://mobaxterm.mobatek.net/download.html

#### 将烧写好的SD卡插入RDK X3,将串口转接板与RDK X3正确连接

Session->new session。设置MobaXterm串口通信参数,波特率921600

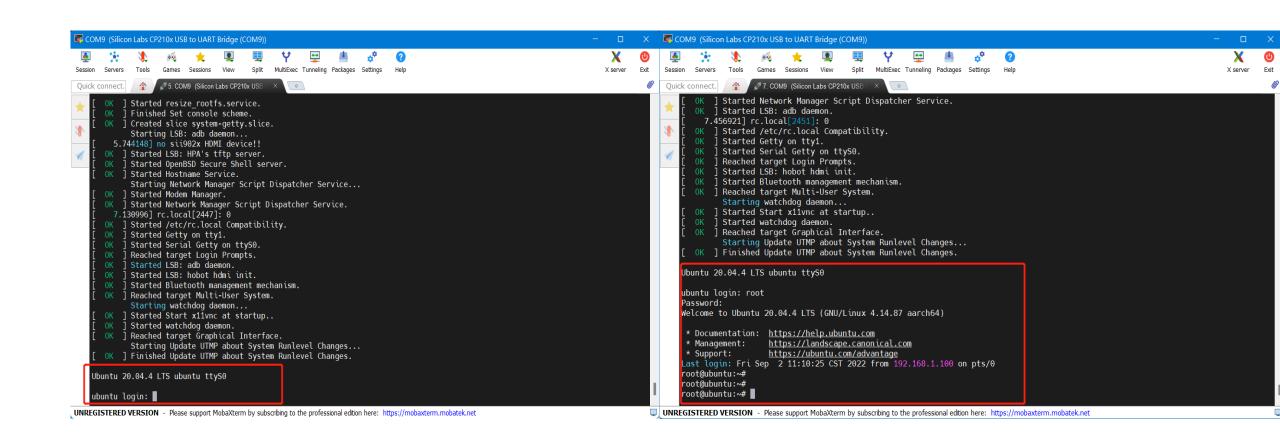




### 运行环境配置







#### 输入用户名root

#### 输入密码root (不会显示出来)

### 连接手机创建Wifi热点



ether 70:f7:54:cd:64:df txqueuelen 1000 (Ethernet)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

RX errors 0 dropped 0 overruns 0 frame 0

RX packets 0 bytes 0 (0.0 B)

TX packets 0 bytes 0 (0.0 B)



#### 注意:

- 1.请将个人热点中的wifi频段设置为2.4G
- 2.用户名与密码中不要携带空格

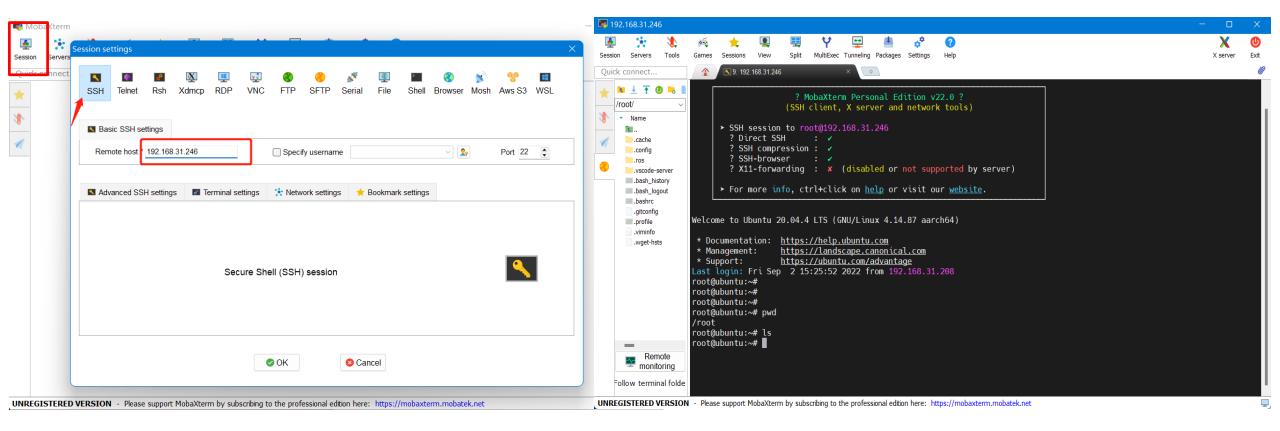
```
root@ubuntu:~# sudo nmcli device wifi rescan
 root@ubuntu:~# sudo nmcli device wifi list
 IN-USE BSSID
                                SSID
                                                  MODE
                                                          CHAN
                                                                 RATE
                                                                               SIGNAL
                                                                                         BARS
                                                                                                SECURITY
          A2:9D:7E:55:0A:AA
                                                  Infra 2
                                                                 130 Mbit/s
          50:2D:BB:D0:0B:7A
                                                                 65 Mbit/s
                                midea ca 0019
                                                 Infra 2
                                                                                                WPA2
                                                  Infra
                                                                 130 Mbit/s
          34:FC:A1:9C:A7:AB
                                602
                                                                                                WPA1 WPA2
                                                                 130 Mbit/s
                                                  Infra 2
          9C:9D:7E:55:0A:AA
                               XH-Home
                                                  Infra
          74:05:A5:93:24:2B
                                D2-501
                                                                 270 Mbit/s
                                                                                    sunrise@ubuntu:~$ ifconfig
          9C:D8:63:DA:4C:22
                               HF-LPT130
                                                  Infra
                                                                 135 Mbit/s
                                                                                   eth0: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
                                                                                          inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
          DC:FE:18:88:30:1B
                                THINK-Network Infra
                                                                 405 Mbit/s
                                                                                          inet6 fe80::211:22ff:fe3b:f355 prefixlen 64 scopeid 0x20<link>
                                                                                         ether 00:11:22:3b:f3:55 txqueuelen 1000 (Ethernet)
          FC:7C:02:40:FD:B7
                                                  Infra 3
                                                                 270 Mbit/s
                                guer770503
                                                                                         RX packets 156 bytes 21991 (21.9 KB)
          C8:8F:26:19:DC:4F Topway 19DC4F Infra 1
                                                                 130 Mbit/s
                                                                                         RX errors 0 dropped 13 overruns 0 frame 0
                                                                                         TX packets 63 bytes 3022 (3.0 KB)
 root@ubuntu:~# sudo wifi connect ">
                                                                                         TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
                                                                                         device interrupt 43 base 0xa000
 Device 'wlan0' successfully activated with '4ea86192-91fa-4cd0-bdd7-ae
                                                                                    lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
                                                                                          inet 127.0.0.1 netmask 255.0.0.0
                                                                                          inet6 :: 1 prefixlen 128 scopeid 0x10<host>
                                              #扫描wifi网络
$ sudo nmcli device wifi rescan
                                                                                          loop txqueuelen 1000 (Local Loopback)
                                                                                         RX packets 114 bytes 8928 (8.9 KB)
                                           #列出找到的wifi网络
$ sudo nmcli device wifi list
                                                                                         RX errors 0 dropped 0 overruns 0 frame 0
                                                                                         TX packets 114 bytes 8928 (8.9 KB)
$ sudo wifi_connect "SSID" "PASSWD" # 连接某指定的wifi网络
                                                                                         TX errors θ dropped θ overruns θ carrier θ collisions θ
$ ifconfig #查看开发板IP地址
                                                                                    wlan0: flags=4099<UP, BROADCAST, MULTICAST> mtu 1500
```

# 通过SSH访问RDK X3





# 配置mobaXterm,填入X3派IP地址 用户名与密码均输入root 输入 top 指令,查看系统状态



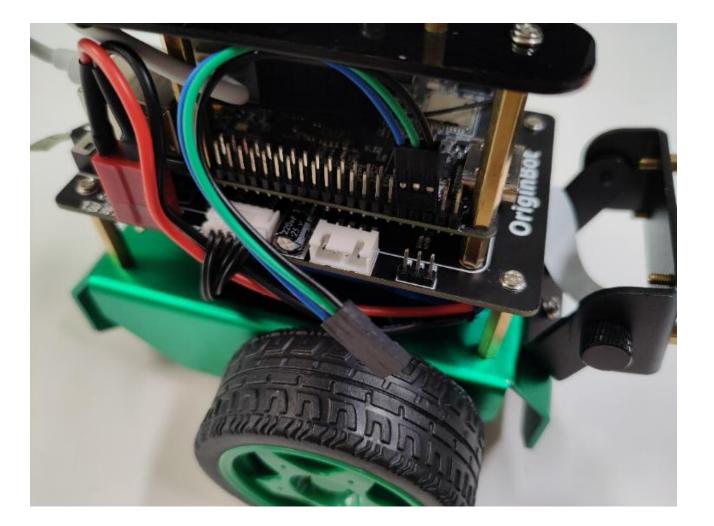
# • 控制器固件安装





按照烧写控制器固件步骤操作即可

注意: 烧写控制器前请断开控制器与RDK X3之间的连接



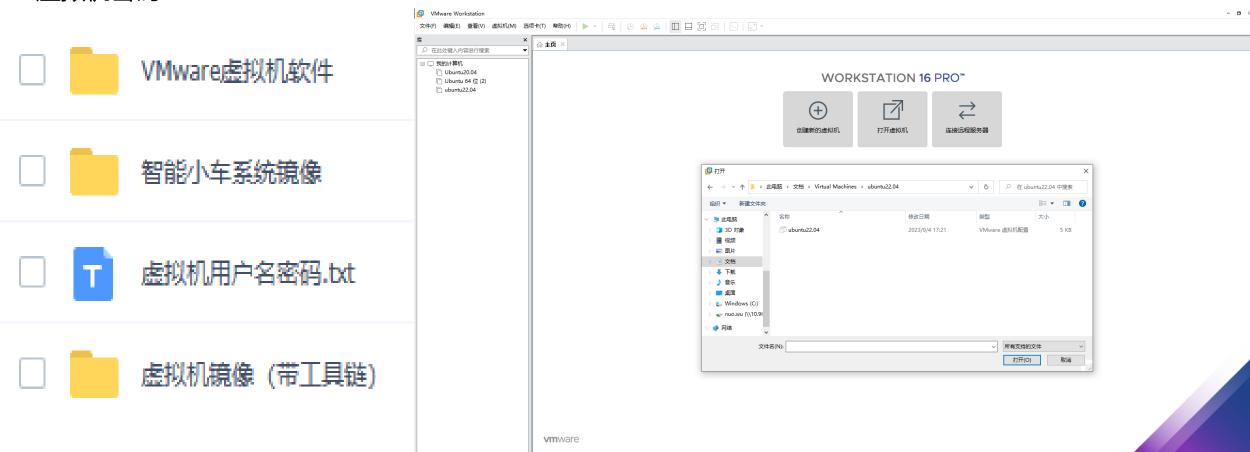
# • 打开虚拟机





- 1.安装虚拟机软件
- 2.解压虚拟机镜像
- 3.打开虚拟机 (选择解压的虚拟机镜像文件夹)

虚拟机密码: ros



# • 报告





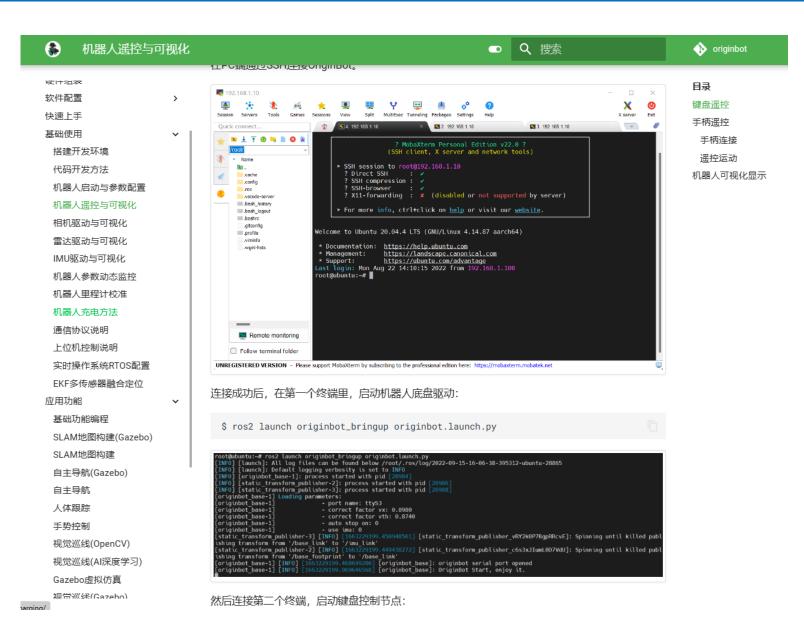
- 1.识别原理
- 2.实现过程
- 3.遇到的问题
- 4.运行效果 (视频)
- 5.实验感想

### • 示例运行





- 1.基础使用(实现rviz可视化即可)
  - 1.1 机器人遥控与可视化
  - 1.2 相机驱动与可视化
  - 1.3 雷达驱动与可视化
  - 1.4 IMU驱动与可视化
- 2.应用功能
  - 2.1 人体跟随
  - 2.2 手势控制
  - 2.4 SLAM地图构建
  - 2.5 自主导航
  - 2.3 视觉巡线 (OpenCV)



# • 视觉巡线 (AI深度学习)





- 1.PC端环境配置 (打开虚拟机,无需安装下载,已经配置好)
- 2.数据采集与标注(按照文档操作,注意:相机是在车上跑,标注是在PC端跑,建议三百张左右)
- 3.模型训练(按照文档操作,在PC端进行训练)
- 4.模型转换 (在PC端进行)
  - 4.1 生成onnx模型
  - 4.2/home/gyh/dev\_ws/src/originbot\_desktop/originbot\_deeplearning/line\_follower\_mod el功能包下的数据集文件夹(image\_dataset)和生成的onnx模型拷贝到/home/gyh/horizon\_xj3\_open\_explorer\_v2.3.3\_20220727/ddk/samples/ai\_toolchain/horizon\_model\_convert\_sample/03\_classification/10\_model\_convert/mapper包中
  - 4.3 启动容器: sudo docker start e519a2b9d7f1
  - 4.4 进入容器: sudo docker exec -it e519a2b9d7f1 /bin/bash
  - 4.5 进入工具链文件夹: cd /ai toolchain
  - 4.6 按照文档,从生成校准数据步骤开始操作
- 5.模型部署(按照文档操作,将模型从虚拟机发送至板端。scp 模型路径 root@板端ip:板端路径)
- 编译命令: colcon build --packages-select line\_follower\_perception (/userdata/dev\_ws路径下)