



主题订阅命令

1)显示所有主题

\$ gz topic -1

2)订阅扫描数据

\$ gz topic -e /gazebo/default/user/turtlebot3_\${TB3_MODEL}/lidar/hls_lfcd_lds/scan 3)订阅图像数据

Waffle

- \$ gz topic -e /gazebo/default/user/turtlebot3_waffle/image/intel_realsense_r200/image
 Waffle Pi
- \$ gz topic -e /gazebo/default/user/turtlebot3_waffle_pi/image/raspberry_pi_cam/image 4)执行侦听器
- \$ cd \${turtlebot3_gazebo_plugin}/build
- \$./lidar_listener \${TB3_MODEL}
 - 打开一个新的终端窗口, 然后输入以下命令。
- \$ cd \${turtlebot3_gazebo_plugin}/build
- \$./image_listener \${TB3_MODEL}

10. 安装 Arduino 配置 OpenCR 库

【注意: OpenCR 板子需要两个步骤,烧写**固件**(类似一个操作系统)和烧写**程序**(程序 是运行在固件环境上的),所以需要先装固件后烧写程序】



10.1 安装 Arduino IDE

OpenCR 需 Arduino IDE 1.16.0 及之后的版本 从官方的 arduino 主页下载最新版本,地址: https://www.arduino.cc/en/Main/Software

HOME STORE SOFTWARE EDU RESOURCES COMMUNITY HELP	
Download the Arduino IDE	
ARDUIND 1.8.10 The open-source Adducts Software (DCD) makes it reasy to Windows 200 files for itom address 100 and up Windows 200 files for itom addres	$\langle \rangle$
如果你是 window 系统,下载【Windows installer】(第一个黑框标记); 如果你是 Ubuntu 系统,下载【Linux 64bits】(第二个黑框标记); 由于 windows 安装比较简单,这里不做介绍,也可参考官网 windows 版本安装教程 <u>https://www.arduino.cc/en/Guide/Windows#toc1</u> 下面简单说一下 Ubuntu 系统的安装方法,如有不清晰之处,请参考官网详细步骤安 <u>https://www.arduino.cc/en/Guide/Linux</u> Ubuntu 安装 Arduino IDE: 1.将下载的文件解压缩(右键解压)到一个文件夹(记住这里), 2. 进入解压出来的文件夹内【arduino-1***】,选择【右键】->【在终端中打开 3. 然后在终端中运行指令 ./install.sh	: :装: f】
<pre></pre>	

4. 然后在开始菜单输入, arduino, 打开 arduino IDE。

10.2 添加 OpenCR 的依赖库

(1) 增加 Boards Manager

运行 IDE, 点击 File(文件) → Preferences(首选项), 复制如下内容到【Additional Boards Manager URLs():

<u>https://raw.githubusercontent.com/ROBOTIS-GIT/OpenCR/master/arduino/opencr_release/pac</u> kage_opencr_index.json





在文本框中键入 OpenCR 以查找包。找到 OpenCR by ROBOTIS 后,点击 Install. 图示:



Type All 🔹 Filter your search	
EMORO 2560 by Inovatic-tCT Baards included in this package EMORO 2560. <u>Onnon help</u> More info	
AME1-Tech Boards by ceplaced by Arrow Boards Boards includes in this package: Smart Ewrything Fox. Online heb More info	
OpenCR by ROBOTES Boards includes in this package: OpenCR. <u>Opinion help</u> <u>More info</u>	Install
	Close

安装后,将显示"INSTALLED" 图示:

Type All Filter your search	
EMORO 2560 by Inevatic KCT Energia induced in this package Defines help Defines help More infe	
AMEL-Tech Boards by replaced by Arrow Boards Boards included in this package: SmartEverything Fox. Online help More infe	
OpenCR by ROBOTIS version 1.0.0 HISTALLED Boards included in this package: OpenCR. Orifice help More info	
	Remove
	Close

OpenCR Board 列在 Tools → Board 中。

图示:





10.3 端口设置

此步骤显示程序上传的端口设置。 OpenCR 应通过 USB 端口连接到 PC 和 OpenCR 板。 选择 Tools → Port → /dev/ttyACM0. 图示:



/dev/ttyACM0 的值可能会根据不同 PC 环境有差异。

10.4 刷固件【Bootloader】

OpenCR 板上的主 MCU 是 STM32F7xx,支持 DFU(设备固件升级)。这使 MCU 的内置 引导加载程序本身能够通过使用 USB 引导 DFU 协议,主要用于引导加载程序初始化,恢复 模式和引导加载程序更新。最大的优点是让用户能够使用 USB 引导加载程序,无需其他 JTAG 设备。使用嵌入在 MCU 中的 DFU 模式写入固件,而无需编写/调试设备,例如 STLink。

操作步骤:

(1) 选择 Tools (工具) → DFU-UTIL



<pre>sketch_oct24b void setup() { // put your set</pre>	Auto Format Archive Sketch Fix Encoding & Reload Serial Monitor Serial Plotter	Ctrl+T Ctrl+Shift+M Ctrl+Shift+L	
<pre>} void loop() {</pre>	WiFi101 Firmware Updater		
<pre>void loop() { // put your mai }</pre>	Board: "OpenCR Board" Upload method: "OpenCR Bootloader" Port: "/dev/ttyACM0" Get Board Info) j.	
	Programmer: "DFU_UTIL" Burn Bootloader	•	AVR ISP AVRISP mkli USB tinyISP ArduinoISP USB asp Parallel Programmer
			Arduino as ISP Arduino Gemma Atmel STK500 development board BusPirate as ISP
	OpenCR Board, OpenCR Bootloader on	/dev/ttyACM0	DFU_UTIL

(7)运行 DFU 模式

OpenCR 上: 按下【Reset 键】同时按下【Boot 键】,就能激活 DFU 模式



(8)下载 bootloader
Arduino IDE 中: 点击 Tools → Burn Bootloader。
* 必须要看到【file download successfully】才证明固件烧写成功



	<u>Lools H</u> elp			
	Auto Format			Ctrl+T
	Archive Sketch			
sketch_oct24a	Fix Encoding & Rela	bad		
oid setup() {	Serial Monitor			Ctrl+Sbift+M
// put your set	Social Diotton			CEEL SHIFE I
	Senal Plotter			CUL+SHILL+L
	WiFi101 Firmware l	Jpdater		
<pre>>id loop() { // put your mail</pre>	Board: "OpenCRB	oard"		•
// put your mai	Upload method: "C		otloader"	•
	Port	Perioribo		
				, , , , , , , , , , , , , , , , , , ,
one burning boot	Get Board Info			
laiming USB DELL	Programmer: "DFU	J_UTIL"		Þ
etting Alternate	Burn Bo <u>otloader</u>			
etermining devic	e status: state = or	UERRUR, ST	atus = 10	
fuERROR, clearin	g status			
etermining devic	e status: state = df	WIDLE, sta	itus = 0	
TUIDLE, CONTINUI El modo dovico D	ng EU varcian Alla			
-o mode device p	FU VEISIUN ULLA			
avice returned t	ransfer size 2048			
evice returned t fuSe interface n	ransfer size 2048 ame: "Internal Flash			
evice returned t fuSe interface n ownloading to ad	ransfer size 2048 ame: "Internal Flash dress = 0x08000000,	, " size = 255	608	\cap
evice returned t fuSe interface n ownloading to ad	ransfer size 2048 ame: "Internal Flash dress = 0x08000000,	size = 255	608 0. hvt	\cap
evice returned t fuSe interface n ownloading to ad ownload [ransfer size 2048 ame: "Internal Flash dress = 0x08000000,) " size = 255] 0%	508 0 byt 2048 byt	es
evice returned t fuSe interface n ownloading to ad ownload [ownload [=== ownload [====	ransfer size 2048 ame: "Internal Flash dress = 0x08000000,	n " size = 255] 0%] 8%] 16%	508 0 byt 2048 byt 4096 byt	es es
evice returned t fuSe interface n ownloading to ad ownload [ownload [=== ownload [===== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000,	n " size = 255] 0%] 8%] 16%] 24%	508 0 byt 2048 byt 4096 byt 6144 byt	es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [===== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, ====	n " size = 255] 0%] 8%] 16%] 24%] 32%	508 0 byt 2048 byt 4096 byt 6144 byt 8192 byt	es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [==== ownload [===== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, 	n " size = 255] 0%] 8%] 16%] 24%] 32%] 40%	0 byt 2048 byt 4096 byt 6144 byt 8192 byt 10240 byt	es es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [==== ownload [====== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, 	5 = 255 3 0% 3 8% 1 16% 1 16% 3 24% 3 22% 3 40% 4 43% 4 43%	0 byt 2048 byt 4096 byt 6144 byt 8192 byt 10240 byt 12288 byt	es es es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [==== ownload [===== ownload [====== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, === ==== =====	5ize = 255 3 8% 3 8% 3 16% 3 24% 3 22% 3 40% 4 43% 3 56%	0 byt 2048 byt 4096 byt 6144 byt 8192 byt 10240 byt 12288 byt 14336 byt	es es es es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [===== ownload [====== ownload [====== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, === ==========================	5ize = 255 6% 8% 16% 24% 24% 32% 340% 48% 48% 56% 64%	0 byt 2048 byt 4096 byt 6144 byt 8192 byt 10240 byt 12288 byt 14336 byt 16384 byt	es es es es es es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [===== ownload [====== ownload [====== ownload [====== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, === ==== ===== =====	5ize = 255 6% 8% 16% 24% 24% 32% 32% 32% 34% 34% 34% 34% 34% 34% 34% 34	0 byt 2048 byt 4096 byt 6144 byt 8192 byt 10240 byt 12288 byt 14336 byt 16384 byt 18432 byt	es es es es es es es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [===== ownload [====== ownload [====== ownload [====== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, === ==========================	size = 255	0 byt 2048 byt 4096 byt 6144 byt 8192 byt 10240 byt 12288 byt 14336 byt 16384 byt 18432 byt 20480 byt	es es es es es es es es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [===== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, === ==========================	size = 255 0% 8% 16% 24% 24% 32% 32% 40% 40% 40% 56% 64% 72% 80% 88%	0 byt 2048 byt 4096 byt 6144 byt 10240 byt 12288 byt 14336 byt 16384 byt 18432 byt 20480 byt 22528 byt	es es es es es es es es es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [===== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, ==============================	size = 255 0% 8% 16% 24% 24% 32% 40% 40% 48% 48% 56% 64% 72% 80% 88% 96%	608 0 byt 2048 byt 4096 byt 6144 byt 8192 byt 10240 byt 12288 byt 14336 byt 16384 byt 18432 byt 18432 byt 20480 byt 22528 byt 24576 byt	es es es es es es es es es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [===== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [======	<pre>ransfer size 2048 ame: "Internal Flash dress = 0x08000000,</pre>	size = 255 0% 8% 16% 24% 32% 32% 40% 43% 48% 48% 56% 72% 80% 88% 96% 100%	08 0 byt 2048 byt 4096 byt 6144 byt 10240 byt 12288 byt 14336 byt 16384 byt 18432 byt 20480 byt 22528 byt 24576 byt 25508 byt	es es es es es es es es es es es es es e
evice returned t fuSe interface n ownload [ownload [== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, 	size = 255 0% 8% 16% 24% 32% 40% 43% 43% 48% 56% 64% 72% 86% 88% 96% 100%	008 0 byt 2048 byt 4096 byt 6144 byt 10240 byt 12288 byt 14336 byt 16384 byt 18432 byt 20480 byt 22528 byt 24576 byt 25508 byt	es es es es es es es es es es es es es e
evice returned t fuSe interface n ownload [ownload [== ownload [===== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [======	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, ==============================	size = 255 0% 8% 16% 24% 24% 32% 40% 40% 40% 56% 64% 64% 72% 80% 88% 96% 100%	0 byt 2048 byt 4096 byt 6144 byt 10240 byt 12288 byt 14336 byt 16384 byt 18432 byt 20480 byt 22528 byt 24576 byt 25508 byt	es es es es es es es es es es es es
evice returned t fuSe interface n ownload [ownload [== ownload [=== ownload [===== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [======	<pre>ransfer size 2048 ame: "Internal Flash dress = 0x08000000, </pre>	" size = 255 0% 8% 16% 24% 24% 32% 32% 48% 48% 56% 48% 56% 64% 72% 80% 88% 96% 100%	0 byt 2048 byt 4096 byt 6144 byt 10240 byt 12288 byt 14336 byt 16384 byt 18432 byt 20480 byt 22528 byt 24576 byt 25508 byt	es es es es es es es es es es es es es e
evice returned t fuSe interface n ownload [ownload [== ownload [=== ownload [===== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload [====== ownload done. ile downloaded s	ransfer size 2048 ame: "Internal Flash dress = 0x08000000, 	" size = 255 0% 8% 16% 24% 32% 32% 48% 48% 48% 56% 48% 56% 64% 72% 80% 88% 96% 100%	0 byt 2048 byt 4096 byt 6144 byt 10240 byt 12288 byt 14336 byt 16384 byt 16384 byt 18432 byt 20480 byt 22528 byt 24576 byt 25508 byt	es es es es es es es es es es es es es e

10.5 上传程序

将 TurtleBot3 的程序烧录到 OpenCR 中:

如果 TurtleBot3 Burger,

Arduino-->File → Examples → turtlebot3 → turtlebot3_burger → turtlebot3_core. 如果使用的是 TurtleBot3 Waffle 或者 Waffle Pi,

Arduino--> File \rightarrow Examples \rightarrow turtlebot3 \rightarrow turtlebot3_waffle \rightarrow turtlebot3_core.

点击【上传】上传到 OpenCR 板。

* 如上图,烧写 arduino 程序时,必须要看到下面状态栏 输出 "jump_to_fw" 字样,才能确定程序少写完成。



<u>File E</u> dit <u>S</u> ketch]	<u>T</u> ools <u>H</u> elp		
🕑 🚯 🗈 🖬 🖬	Upload		ø
turtlebot3_core	turtlebot3_core_config.h	turtlebot3_motor_driver.cpp	turtl ▼ ot3
/********************* * Copyright 2016 R * * Licensed under t	**************************************	2.0 (the "License");	*****
* you may not use * You may obtain a *	this file except in compli- copy of the License at	ance with the License.	
* <u>http://www.a</u>	pache.org/licenses/LICENSE	-2.0	
* Unless required * distributed unde * WITHOUT WARRANTI * See the License * limitations unde ********	by applicable law or agree r the License is distribut ES OR CONDITIONS OF ANY KI for the specific language r the License.	d to in writing, software ed on an "AS IS" BASIS, ND, either express or implied governing permissions and	
/* Authors: Yoonse	ok Pyo, Leon Jung, Darby L	im */	
#include "turtlebo	t3_core_config.h=		
/*************************************	**************************************	***********************************	******
/*************************************	*************************************	wist& cmd_vel_msg); l sub("cmd vel", cmd vel call	*******/ ******/ back);
/*************************************	****	***********************	*****
(•(00 10))))
Done uploading.			
flash_write : 0 : 1 CRC OK 11444DE 1144 [OK] Download jump_to_fw	L.566000 sec 14DE 0.005000 sec		
1		OpenCR Board, OpenCR Bootloader on /o	dev/ttyACM0

11. 树莓派相机的使用



11.1 摄像头安装

注意:树莓派上有两个外表相似的接口,但是不通用,一个用来插摄像头,另一个是显示 屏。插摄像头的在**耳机接口和 HDMI 接口中间**。



摄像头安装示意图

- 1。将摄像头卡扣,轻轻向上提起。
- 2。将排线触点一侧面向树莓派插卡方向,向下插入排线。
- 3。排线插好以后,将摄像头卡扣向下按下。



11.2 树莓派启用摄像头

树莓派专用摄像头已经内置了驱动.所以只需要从配置中启用就可以了

1.从命令行运行 raspi-config 配置软件

sudo raspi-config

2 选择 Interface Options



4. 选择是,然后选择Finish,重启树莓派。

would ye	ou like	the	camera	interface	to be	enabled?
		_				



5 测试摄像头,运行: raspistill -o cam.jpg 在当前目录下会保存一张 cam.jpg 的照片,检查是否成功。 [TurtleBot]以下命令将在 ROS 系统上安装相关的 Raspberry Pi Camera 软件包。 \$ cd ~/catkin_ws/src \$ git clone https://github.com/UbiquityRobotics/raspicam_node.git \$ sudo apt-get install ros-kinetic-compressed-image-transport ros-kinetic-camera-info-manager \$ cd ~/catkin_ws && catkin_make

11.3 ROS 运行树莓派相机

树莓派上运行:

\$ roslaunch turtlebot3_bringup turtlebot3_rpicamera.launch 要么

\$ roslaunch raspicam_node camerav2_1280x960.launch

在 raspicam 节点运行时,您可以通过启动从 Raspberry Pi Camera 中查看各种数据 rqt image view。

警告! 在 Remote PC 中运行 Rviz 之前,请检查 Raspberry Pi 3 和 Remote PC 是否已连接。

[Remote PC]运行以下命令

\$ rqt_image_view

gui 应用程序出现在屏幕上后,您可以从应用程序顶部的下拉菜单中选择与 Raspberry Pi Camera 相关的数据主题名称。

11.4 相机标定

相机的标定是为了生成相机的内参,保存成相机的 yaml 文件。这样在启动时就会调用。 标定能提高图片的精度,对精度要求不高的场景可以不用标定。步骤

(1) 首先准备的 launch 文件, 在树莓派上:

\$ roscd usb_cam/launch/

\$ gedit calib.launch

写入以下内容

<launch>

<node name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen" >

<param name="video_device" value="/dev/video0" />

<param name="image width" value="640" />

<param name="image_height" value="480" />

<param name="pixel_format" value="yuyv" />

<param name="io_method" value="mmap"/>

</node>

</launch>

(2) 启动相机

\$ roslaunch usb_cam calib.launch



(3) 启动标定程序

\$ rosrun camera_calibration cameracalibrator.py --size 8x6 --square 0.03
image:=/usb_cam/image_raw camera:=/usb_cam

出现以下界面:



(4) 用标定板进行标定

下载标定版,打印成 A4 大小,贴到一个平整的物体:标定板网址:

https://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibrat ion?action=AttachFile&do=view&target=check=108.pdf

等待右边的 x y size skaw 都变绿,就可以点击 Calibration 运算,运算过程会非常的缓慢, 电脑会卡住不动,**稍安勿躁**。

然后运算完成后,点"SAVE",点"COMMIT"。 生成的参数文件保存在↓,他超级非常重要。 ~/.ros/camera_info/head_camera.yaml 我们下一次运行 usb_cam 就会自动调用文件,并有如下的信息: [INFO][1555832606.313276288]: camera calibration URL: file:///home/sc/.ros/camera_info/head_camera.yaml

如果此方法无法实现标定,可参考 ROS 官方单目相机标定

http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration

也可使用自动驾驶包或家庭服务包里标定包标定,或自行网上搜索单目相机标定,如有更 好的标定方法请及时联系我们更新。



12 常见问题解决办法

12.1 Turtlebot3 软件更新

更新【树莓派】上的 TB3 库

\$ cd ~/catkin_ws/src/

\$ rm -rf turtlebot3/ turtlebot3_msgs/ hls_lfcd_lds_driver/ #删除旧的软件包

\$ git clone https://github.com/ROBOTIS-GIT/hls_lfcd_lds_driver.git

\$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git

\$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git

\$ cd ~/catkin_ws/src/turtlebot3

\$ sudo rm -r turtlebot3_description/ turtlebot3_teleop/ turtlebot3_navigation/

turtlebot3_slam/ turtlebot3_example/

\$ cd ~/catkin_ws/

\$ rm -rf build/ devel/

\$ cd ~/catkin_ws && catkin_make -j4

更新【笔记本电脑】的 TB3 库

\$ cd ~/catkin_ws/src/

\$ rm -rf turtlebot3/ turtlebot3 msgs/

\$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git

\$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git

\$ cd ~/catkin_ws/

\$ rm -rf build/ devel/ #此为删除上次自动生成的 catkin_ws 里面的 build 和 devel 文件 \$ cd ~/catkin_ws && catkin_make

关于 TB3 上软件 SDK 更新 GitHub 官方网址为 https://github.com/ROBOTIS-GIT,可根据 自己使用的小车,自行下载最新代码到自己工作空间(catkin_ws/src)编译(catkin_make)。

12.2 时间同步

有时候运行指令会出现时间不同步的错误提示,这时需要树莓派和笔记本电脑之间时间同步。以下步骤在树莓派和笔记本电脑上都要运行一遍。

首先安装 ntpdate,然后同步时间

\$ sudo apt-get install ntpdate

\$ sudo ntpdate ntp.ubuntu.com

注意:如有节点经常 dead 或等异常现象,可能是 pc 端的内存或处理器不够,请换台配置 好点的 pc 去重新测试下。



12.3 树莓派上 SSH 远程连接

树莓派上运行:

\$	sudo	apt-get install ssh
\$	sudo	service ssh start
\$	sudo	ufw allow ssh
然	后在筆	管记本电脑上就能运行 :
\$	ssh 7	对莓派主机名@树莓派 IP

12.4 远程无法控制小车

检查流程的基本思路:先排除网络问题,再排除软件问题,最后排除硬件问题。

- a. 不使用 wifi 通信:把小车轮子悬空,直接在树莓派上运行 key_teleop 节点。如果树莓派 上可以控制小车,就说明硬件是好的,问题出在网络通讯上了,转 c。否则就是硬件除 了问题,转 d.
- b. [查看网络]: PC 启动 roscore,在树莓派终端查看 rostopic list. 看有没有节点,如果有 是正常的,没有就[查看 IP 配置],或参见教程的远程通讯配置一节。
- c. [硬件检查]:用 wizard 测试,不行就恢复树莓派连上屏幕键盘,单独在树莓派上运行 teleop 节点,能否控制。若能,转[OpenCR 检查]。若不能,转 b

d. [OpenCR 检查]:确认 opencr 固件有没有烧(),长按 opencr 上的测试 bt2,轮子动不动?()。 如果无法自己解决,请将如下截图发到售后群:

[查看 IP 配置]分别在树莓派和笔记本运行 cat ~/.bashrc 输出截图。分别运行 ifconfig 输出 截图。分别运行 cat /etc/hosts 输出截图。

12.5 工作空间编译错误

在安装一些库时,下载源代码到 catkin_ws 目录里,编译会出现错误,首先采用 ros 自带的依赖解决工具 rosdep。

Catkin_make 错误

\$ rosdep install -r -- from-path . #有个点

参数说明:

1. --from-paths 是指定要解决哪里的依赖,一个点是当前目录的意思。

2.-r 是递归解决层级目录,就是说每个子文件夹都要解决。

如遇到以上常见问题以外的问题,请将问题详细描述清楚附图或问题视频,在 ROBOTIS 售 后技术支持群中提问。