

六叶树产品教程

ROS2 CANopen 工程移植教程

类别	内容
关键词	ROS2 机器人系统下 canopen 调试 环境搭建
摘要	如何在 ROS2 系统下移植开源 CANopen 工程

库介绍:

类别	内容
安装环境	Ubuntu22.04
ROS 版本	ROS2
	humble
GCC 版本	gcc 11.4.0
支持设备	六叶树 USBCAN 系列适配器:USBCAN1/USBCAN2
	六叶树 USBCANFD 系列适配器:USBCANFD-Min/USBCANFD1/USBCANFD2
	六叶树 SOCKETCAN 系列适配器:UTC2201/UTC2202/UTF2202

文档记录

版本	日期	 说明
V1.00	2025.01	创建文档



目录

计树产品教程	1
OS2 CANopen 工程移植教程	1
1、环境准备	3
2、系统环境准备	6
3、CANopen 工程移植	7
4、六叶树适配器驱动安装1	0
5、CAN 卡调试工具安装1	.1
6、硬件接线1	.1
7、调试步骤1	.1
8 工程说明2	20
9 常用链接2	23
10 联系方式2	23



1、环境准备

系统版本	Ubuntu22.04			
GCC 版本	gcc 11.4.0			
系统软件源	清华源			
平台 CPU 架构	X86			
其他	1.主机能够访问互联网			
	2.主机能访问 github 这些域名			
	3.所有命令用有 root 权限的账户进行执行			
	4.建议用虚拟机(出错了可以回退)			
ROS 环境	ROS2 humble			
	ROS 安装教程见"六叶树产品教程-ROS 系统安			
	装.V1.00.pdf"			
CANFD 适配器	六叶树 UTC2201 适配器*1(CANopen 控制)			
	六叶树 USBCANFD-Mini*1(辅助测试验证)			
调试软件	调试软件 六叶树 CanAssistant			
	软件下载地址:www.liuyeshu.cn			

root@lys-virtual-machine:/home/lys/ros2# lsb_release No LSB modules are available. root@lys-virtual-machine:/home/lys/ros2# lsb_release -a No LSB modules are available. Distributor ID: Ubuntu Description: Ubuntu 22.04.5 LTS Release: 22.04 Codename: jammy root@lys-virtual-machine:/home/lys/ros2#



UTC2201 适配器:

AR HOSS

●自带匹配电阻,可开关 ●X86/64/ARM支持

六叶树教育科技有限公司 ©版权所有





●电源信号双隔离 ●5K-1M波特率 , 支持自定义



USBCANFD-Mini 适配器:



●内置匹配电阻 ●自定义波特率 ●智能ID过滤

2、系统环境准备

2.1 软件源更换为清华源

1.用以下命令查看版本名。 lsb_release -c

2.备份原来的源,将以前的源备份一下,以防以后需要用的。 sudo cp /etc/apt/sources.list /etc/apt/sources.list1



3.打开/etc/apt/sources.list 文件,添加源,并保存。 sudo gedit /etc/apt/sources.list

4.复制清华源到文件中:

#添加清华源

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse

5.粘贴完之后,直接退出保存。

2.2 执行命令更新系统软件

sudo apt-get update sudo apt-get upgrade

3、CANopen 工程移植

3.1 依赖库安装

1.使用 root 权限用户 su



2.安装:

apt-get install -y ros-humble-diagnostic-updater ros-humble-controller-interface ros-humble-controller-manager ros-humble-realtime-tools

apt-get install -y python3-rosdep python3-argcomplete python3-colcon-common-extensions build-essential pkg-config python3-wheel

3.2 工程源码下载与编译

 初始化 ROS2 的工作空间 mkdir -p ~/ros2/src cd ~/ros2/src
 克隆 ROS CANopen 到工作空间: 方式一: git clone https://github.com/ros-industrial/ros2_canopen.git 不推荐:
 1.github 经常访问不了
 2.下载的代码有问题,在当前(2025 年 01 月 03 日)的 ROS2 系统下编译有问题。

方式二:(推荐,代码问题已纠正)

wget http://www.liuyeshu.cn/wp-content/uploads/2024/12/lysRos2CanOpenSrc.tar.gz tar -xvf lysRos2CanOpenSrc.tar.gz cd ros2_canopen

 root@lys-virtual-machine:~/ros2/src/ros2_canopen# ls

 canopen
 canopen_core
 canopen_master_driver
 canopen_ros2_controllers
 Dockerfile

 canopen_402_driver
 canopen_fake_slaves
 canopen_proxy_driver
 canopen_tests
 lely_core_libraries

 canopen_base_driver
 canopen_interfaces
 canopen_ros2_control__canopen_utils
 README.md

3.构建并安装: cd ../../ sudo rosdep init rosdep update rosdep install --from-paths src/ros2_canopen --ignore-src -r -y





colcon build

root@lys-virtual-machine:~/ros2# colcon build Starting >>> lely_core_libraries Starting >>> lely_core_libraries Sinished <<< canopen_interfaces [12.5s] Starting >>> canopen_utils Finished <<< canopen_utils [1.51s] [Processing: lely_core_libraries] --- stderr: lely_core_libraries Cloning into 'upstream'... /usr/lib/python3/dist-packages/setuptools/command/install.py:34: SetuptoolsDeprecationWarning: setup.py install is deprecated. d pip and other standards-based tools.

编译成功:



Finished <<<	lely_core_libraries [55.7s]
Starting >>>	canopen_core
Starting >>>	canopen_fake_slaves
Finished <<<	canopen fake slaves [11.7s]
[Processing:	canopen core]
Finished <<<	canopen core [42.6s]
Starting >>>	canopen base driver
Starting >>>	canopen_master_driver
[Processing:	canopen_base_driver, canopen_master_driver]
Finished <<<	canopen_base_driver [40.2s]
Starting >>>	canopen_proxy_driver
Finished <<<	canopen_master_driver [42.6s]
Finished <<<	canopen_proxy_driver [26.7s]
Starting >>>	canopen_402_driver
Finished <<<	canopen_402_driver [27.4s]
Starting >>>	canopen_ros2_controllers
Starting >>>	canopen
Finished <<<	canopen [0.33s]
Starting >>>	canopen_ros2_control
[Processing:	canopen_ros2_control, canopen_ros2_controllers]rollers:build 66% - 33.1s] [canopen_ros2_control:build 50% - 32.7s]
[Processing:	canopen_ros2_control, canopen_ros2_controllers]
[Processing:	canopen_ros2_control, canopen_ros2_controllers]
[Processing:	canopen_ros2_control, canopen_ros2_controllers]
Finished <<<	canopen_ros2_controllers [2min 11s]
Starting >>>	canopen_tests
Finished <<<	canopen_ros2_control [2min 27s]
Finished <<<	canopen_tests [20.7s]
Summary: 13	packages finished [5min 45s]
1 package	had_stderr_output: lely_core_libraries
root@lvs-vir	tual-machine:~/ros2#

4、六叶树适配器驱动安装

产品系列	支持的型号	驱动
六叶树 USBCAN 系列适配器	USBCAN1/USBCAN2	需安装对应的驱动
六叶树 USBCANFD 系列适配器	USBCANFD-Mini/USBCANFD1/USB	需安装对应的驱动
	CANFD2	
六叶树 SOCKETCAN 系列适配器	UTC2201/UTC2202/UTF2202	免驱

适配器成功驱动正常的情况下,在系统上可以用 ifconfig -a 查看到 can0 接口。

六叶树 USBCAN 适配器产品驱动安装教程见文档"六叶树产品教程-CANFD 适配器 SOCKET 驱动安装教程 V1.0.pdf"

六叶树 USBCAN 适配器产品 SOCKETCAN 专栏资料下载:<u>http://www.liuyeshu.cn/?page_id=1256</u>





5、CAN 卡调试工具安装

apt-get install can-utils

6、硬件接线

六叶树适配器接口	CANopen 设备接口				
CAN-H	CAN-H				
CAN-L	CAN-L				

7、调试步骤

7.1 CAN 卡参数配置

六叶树实际 can 卡配置:

波特率配置为 CANopen 设备的实际通讯波特率,以下用例将 CAN 卡波特率设置为了 500K。



ip link set can0 type can bitrate 500000 Ip link set up can0

虚拟 can 卡配置:

sudo modprobe vcan sudo ip link add dev vcan0 type vcan sudo ip link set vcan0 txqueuelen 1000 sudo ip link set up vcan0

7.2 CAN 数据采集(非必须)

candump can0

如果 CAN 总线有 CANopen 数据,该命令可以直接看到原始数据,如下图:

cane	102	[5]	04	00	00	00	38	Ç.			
can0	702	[5]	05	B6	20	33	21				
can0	702	[5]	06	Bб	60	33	21				
can0	702	[5]	07	00	00	06	38				
can0	702	[5]	08	B6	A0	33	21				
can0	702	[5]	09	Bб	E0	33	21				
can0	702	[5]	0A	00	00	06	38				
can0	600	[5]	03	05	D8	53	10				
can0	701	[8]	00	4E	92	05	80	20	03	53	
can0	701	[8]	01	4E	D1	F7	80	20	03	4D	
can0	701	[8]	02	4F	09	DD	80	20	03	5D	
can0	701	[8]	03	4F	BA	0B	80	20	04	48	
can0	701	[8]	04	4F	EA	0A	80	20	04	43	
can0	701	[8]	05	57	71	E3	76	20	02	62	
can0	701	[8]	06	57	D2	19	99	60	00	62	
can0	701	[8]	07	58	71	D9	8E	A0	00	67	
			1 March 199		1.00		100				

该步骤非必须步骤,在这里只是用 candump 工具来判断一下 CAN 总线是否已正常,验证操作,可以不执行。

7.3 测试 cia402_setup.launch.py

7.3.1 用默认的虚拟 vcan0 卡

新打开一个新的终端窗口,切换到 root 权限的账户,进入到工程目录。 注意:原始代码所有用例使用的 CAN 卡都是虚拟 can 卡 vcan0,如需使用其他 can 卡需要修 改代码。



lys@lys-virtual-machine:~/ros2\$ pwd
/home/lys/ros2
lys@lys-virtual-machine:~/ros2\$ ls
build install log src
lys@lys-virtual-machine:~/ros2\$

source install/setup.bash

root@lys-virtual-machine:~/ros2# pwd
/root/ros2
root@lys-virtual-machine:~/ros2# source install/setup.bash
root@lys-virtual-machine:~/ros2#

ros2 launch canopen_tests cia402_setup.launch.py

启动成功如上图所示。

root@lys-virtual-machine:~/ros2# ros2 launch canopen_tests cia402_setup.launch.py
[INFO] [launch]: All log files can be found below /root/.ros/log/2025-01-03-16-03-25-687349-lys-virtual-machine-205367
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [launch.user]: /root/ros2/install/canopen_tests/share/canopen_tests/config/cia402/bus.yml
[INFO] [launch.user]: /root/ros2/install/canopen_tests/share/canopen_tests/config/cia402/master.dcf
[INFO] Claunch.user]:
[INFO] [launch.user]: vcan0 原始代码用例全部都是使用的是虚拟can卡
[INFO] [cia402_clave_node_1]+ process started with pid [205378]
[INFO] [device container node-2]: process started with pid [205380]
[device_container_node-2] [INFO] [1735891406.142498887] [device_container_node]: Starting Device Container with:
[device container node-2] [INFO] [1735891406.142612433] [device container node]: master config /root/ros2/install/canopen tests/share/canopen tests/config/cia402/master.do
[device container node-2] [INFO] [1735891406.142619228] [device container node]: bus config /root/ros2/install/canopen tests/share/canopen tests/config/cia402/bus.yml
[device container node-2] [INFO] [1735891406.142622776] [device container node]: can interface name vcan0
[device container node-2] [INFO] [1735891406.143953370] [device container node]: Loading Master Configuration.
[device_container_node-2] [INFO] [1735891406.145061963] [device_container_node]: Load Library: /root/ros2/install/canopen_master_driver/lib/libmaster_driver.so
[device_container_node-2] [INFO] [1735891406.152118900] [device_container_node]: Found class: rclcpp_components::NodeFactoryTemplate <ros2_canopen::masterdriver></ros2_canopen::masterdriver>
[device container node-2] [INFO] [1735891406.152165381] [device container node]: Instantiate class: rclcpp components::NodeFactoryTemplate <ros2 canopen::masterdriver=""></ros2>
[device container node-2] [INFO] [1735891406.160089237] [master]: NodeCanopenBasicMaster
[device container node-2] [INFO] [1735891406.160444136] [device container node]: Load master component.
[device container node-2] [INFO] [1735891406.160504347] [device container node]: Added /master to executor
[device_container_node-2] [WARN] [1735891406.164232047] [master]: No timeout parameter found in config file. Using default value of 100ms.
[device_container_node-2] [INFO] [1735891406.164347598] [master]: Master boot timeout set to 2000ms.
[device container node-2] [INFO] [1735891406.175581970] [device container node]: Loading Driver Configuration.
[device container node-2] [INFO] [1735891406.175677637] [device container node]: Found device cia402 device 1 with driver ros2 canopen::Cia402Driver
[device_container_node-2] [INFO] [1735891406.176362171] [device_container_node]: Load Library: /root/ros2/install/canopen_402 driver/lib/libcia402 driver.so
[device container node-2] [INFO] [1735891406.184833063] [device container node]: Found class: rclcpp components::NodeFactoryTemplate <ros2 canopen::cia402driver=""></ros2>
[device container node-2] [INFO] [1735891406.184893735] [device container node]: Instantiate class: rclcpp components::NodeFactoryTemplate <ros2 canopen::cia402driver=""></ros2>
[device container node-2] [INFO] [1735891406.193352622] [device container node]: Load driver component.
[device container node-2] [INFO] [1735891406.193459493] [device container node]: Added /cia402 device 1 to executor
[device container node-2] [INFO] [1735891406.213068460] [cia402 device 1]: Non transmit timeoutl00ms
[device container node-2] [WARN] [1735891406.213186496] [cia402 device 1]: Could not polling from config, setting to true.
[device container node-2] [WARN] [1735891406.213265998] [cia402 device 1]: Could not read enable diagnostics from config, setting to false.
[device_container_node-2] [INFO] [1735891406.213689717] [cia402_device_1]: scale pos to dev 1000.000000
[device container node-2] scale pos from dev 0.001000
[device container node-2] scale vel to dev 1000.000000
[device_container_node-2] scale vel from dev 0.001000
[device_container_node-2] offset pos_to_dev0.000000
[device container node-2] offset pos from dev 0.000000

I	[device container node-2]	scale pos from dev 0.001000
	[device container node-2]	scale vel to dev 1000.000000
	[device container node-2]	scale vel from dev 0.001000
	[device container node-2]	offset pos to dev 0.000000
	[device container node-2]	offset pos from dev 0.000000
	[device container node-2]	
	[device container node-2]	[INFO] [1735891406.350004980] [cia402 device 3]: eds file /root/ros2/install/canopen tests/share/canopen tests/config/cia402/cia402 slave.eds
	[device container node-2]	[INFO] [1735891406.350018920] [cia402 device 3]: bin file /root/ros2/install/canopen tests/share/canopen tests/config/cia402/cia402 device 3.bin
	[device container node-2]	[WARN] [1735891406.351323588] [cia402 device 3]: Wait for device to boot.
	[device container node-2]	Found rpdo mapped object: index=6040 subindex=0
	[device container node-2]	Found rpdo mapped object: index=6060 subindex=0
	[device container node-2]	Found rpdo mapped object: index=607a subindex=0
	[device container node-2]	Found rpdo mapped object: index=60ff subindex=0
	[device container node-2]	Found rpdo mapped object: index=6040 subindex=0
	[device container node-2]	Found rpdo mapped object: index=607a subindex=0
	[device container node-2]	Found rpdo mapped object: index=6040 subindex=0
	[device container node-2]	Found rpdo mapped object: index=60ff subindex=0
	[device container node-2]	Found tpdo mapped object: index=6041 subindex=0
	[device container node-2]	Found tpdo mapped object: index=6061 subindex=0
	[device container node-2]	Found tpdo mapped object: index=6064 subindex=0
	[device_container_node-2]	Found tpdo mapped object: index=606c subindex=0
	[device_container_node-2]	Found tpdo mapped object: index=6041 subindex=0
	[device_container_node-2]	Found tpdo mapped object: index=6064 subindex=0
	felouine contriner and of	Found to do not show

新打开一个新的终端窗口,用 can 工具抓包,可以看到虚拟 CAN 卡上的 can 通讯报文:

root@lys	-virt	ual-mac	chir	ne:/	/hon	ne/I	Lys,	ros	32#	candump	vcano
vcan0	080	[O]									
vcan0	182	[3]	21	00	00						
vcan0	282	[8]	00	00	00	00	00	00	00	00	
vcan0	203	[3]	68	79	00						
vcan0	202	[3]	46	79	00						
vcan0	080	[O]									
vcan0	182	[3]	21	00	00						
vcan0	282	[8]	00	00	00	00	00	00	00	00	
vcan0	203	[3]	68	79	00						
vcan0	202	[3]	46	79	00						



7.3.2 ROS2 下查看话题服务

新打开一个新的终端窗口,切换到 root 权限的账户,执行命令查看当前的话题。 ros2 topic list



7.3.3 改用实际 CAN0 卡收发数据

这里我们使用修改代码让 canopen 驱动六叶树的 UTC2201 适配器的 can0 接口,六叶树的 USBCANFD-Mini 用于接收 canopen 通过 UTC2201 发出的数据(这里是方便演示数据,使用的 是另外的 CAN 盒进行数据发送数据,不是 7.1 配置的那个 CAN 盒,相当于整个实验用了两 个 CAN 盒,第二个 CAN 盒我们是用来辅助测试的)。





第一步,修改代码

修改文件 cia402_system.launch.py



文件:cia402_setup.launch.py

第70行(修改前):



第70行(修改后):



文件:cia402_lifecycle_setup.launch.py 第70行(修改前):





第 70 行(修改后):



第二步,重新编译该功能包

回到工程目录

```
root@lys-virtual-machine:~/ros2# pwd
/root/ros2
root@lys-virtual-machine:~/ros2# ls
build install log src
root@lys-virtual-machine:~/ros2#
```

colcon build --packages-select canopen_tests

```
root@lys-virtual-machine:~/ros2# colcon build --packages-select canopen_tests
Starting >>> canopen_tests
Finished <<< canopen_tests [2.39s]
Summary: 1 package finished [2.68s]
root@lys-virtual-machine:~/ros2#</pre>
```

第三步,实际 CAN 总线接线

将六叶树 UTC2201 适配器和六叶树 USBCANFD-Mini 适配器的 CAN 总线就行连接并使能各自的匹配电阻(接线处的拨码开关都往下拨)。

UTC2201 USBCANFD-Mini



CAN-H	CAN-H
CAN-L	CAN-L



第四步,六叶树 CanAssistant 配置

				CanAst	(六叶树)		主题			
选择设备	设置 语	言 工具	帮助							
USBCAN1 US		۲ ۲۹۹۹ CANFDMini		USBCANFD2	жи лим UTB2602					
欢迎界面 🖾		Ö		USBC	ANFDMini 配置	x				Û
		设备索引		0	▼ 通道索引:	0 打开所有通道			3	
		工作协议			CAN				3	以加过调
		上作模式 仲裁域波:	侍率:		止常模式 500kbps 80%	.*			4	● 止 显示
		回 誠	特年设 \$(80%),5.0ME	数500K ,	和前面配置	的CAN卡波特率一致。			2	は振保存
			滤波(可选项)			1.5/d-51.99				0
			an 秋火」致重:0	定		取消			5	实时保存
基础发送 高	级发送									
发送方式:	正常发送	- 数据(16)	井制):			☑ ID递增			9	以信回加
帧协议:	CAN	- 00 11 2	2 33 44 55 6	6 77		每次发送帧数:	1			
帧类型:	标准帧	*				发送次数:	1			上具合果
帧格式:	数据帧	*				发送间隔(毫秒):	10			
帧ID(16进制):	00000000					开始	ſ	耻		
数据长度:	8	-								
					设备信息	送速率:0帧/秒 接收速率:0帧/秒 发送机	帧数:0帧 接收	帧数:0帧	清空计	数



				Can	Ast(六叶树)				主题		
¥设备	设置 语言	ī I具	帮助								
SCAN1 US	大明朝 BCAN2 USB	AHM CANFDMini		FD1 USBCAN	大明朝 FD2 UTB2602						
USBCANFI	DMini 设备:0 通j	5 :0 🔀									
第号	传输方向	状态		时间标志	周期(us)	数量	帧ID	帧格式	帧类	型	î
0	接收	正常		09:50:08:353	101	56	0x0000080	数据帧	标准	主顿	
										>	-
基础发送 高	级发送	¥2+H77(1)	244±0×							>	•
基础发送 高 送方式:	级发送 正常发送 CAN	> 数据(1)	进制);	26.07.08			✓ ID递增 每次号详帖教。			>	-
基础发送 高 就送方式: 防议: ■举型:	级发送 正常发送 CAN 标准帧	 > 数据(10 ✓ 01 02 	进制): 03 04 05	D6 07 08			 ✓ ID递增 每次发送帧数: 发送次数: 	1		>	•
基础发送 高 送方式: 动议: 妙型: 始式:	级发送 正常发送 CAN 标准帧 数据帧	 → 数据(10 > 01 02 > √ 	进制): 03 04 05	06 07 08			 ✓ ID递增 每次发送帧数: 发送次数: 发送间隔(章秒): 	1	0	>	
基础发送 高 送左式: 动动议: 妙型: 始杠: 加D(16进制):	级发送 正常发送 CAN 标准帧 数据帧 00000001	 > 数据(10 > 01 02 > 	进制); 03 04 05 ·	06 07 08			✓ ID递增 每次发送帧数: 发送次数: 发送间隔(毫秒): 开始	1	0	> 	

第五步,启动 cia402_setup.launch.py

新打开一个新的终端窗口,切换到 root 权限的账户并 source 好 ROS2 运行环境 //ROS 环境配置 source install/setup.bash

//can0 卡配置 提示:如果前面已配置 can0 卡,则无需重复配置。 ip link set can0 type can bitrate 500000 Ip link set up can0

//启动 ros2 launch canopen_tests cia402_setup.launch.py



root@lys-virtual-machine:~/ros2# ros2 launch canopen_tests cia402_setup.launch.py [INFO] [launch]: All log files can be found below /root/.ros/log/2025-01-04-09-55-54-040251-lys-virtual-machine-3404
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [launch.user]: /root/ros2/install/canopen_tests/share/canopen_tests/config/cia402/bus.yml
[INFO] [launch.user]: /root/ros2/install/canopen_tests/share/canopen_tests/config/cia402/master.dcf
[INFO] [Jaunch.user]: 口切场到六叶材加oon0
[INFO] [Lauren Juser]: cano Estated with nid [2015]
[INFO] [cla402_stave_noue-j]; process started with pid [3417]
<pre>law of local_container node-21 [INFO] [173595754.251189521] [device container node]: Starting Device Container with:</pre>
[device_container_node-2] [INFO] [1735955754.251296412] [device_container_node]: master config /root/ros2/install/canopen tests/sha
re/canopen_tests/config/cia402/master.dcf
[device_container_node-2] [INFO] [1735955754.251303275] [device_container_node]: bus_config /root/ros2/install/canopen_tests/share/
canopen_tests/config/cia402/bus.yml
[device_container_node-2] [INFO] [1/35955/54.25130/2222] [device_container_node]: can interface_name_cano
loevice_container_node-2/ [INFU] [1/30935/34.253226/99] [device_container_node]: Loading Master Contiguration. [device_container_node-2] [INFU] [1/35955/54.25329/040] [device_container_node]: Loading Master Contigurations.
[device container node-2] [INFO] [1735955754.257260127] [device container node]: Found class: rclcpp components::NodeFactoryTemplate <ros2 c<="" td=""></ros2>
anopen::MasterDriver>
[device_container_node-2] [INFO] [1735955754.257299170] [device_container_node]: Instantiate class: rclcpp_components::NodeFactoryTemplate<
ros2_canopen::MasterDriver>
[device container node-2] [INFO] [1735955754.265007083] [master]: NodeCanopenBasicMaster
[device container node-2] Found rpdo mapped object: index=6040 subindex=0
[device_container_node-2] Found_rodo_mapped_object: index=6060_subindex=0
[device_container_node-2] Found_rodo_mapped_object: index=607a_subindex=0
device_container_node_21_Found_rndo_mapped_object: index=60ff_subindex=0
device_container_node_2] Found_rade managed object: index_66/0 subindex_6
[device_container_inde-2] Found thus mapped object. Index-0040 Subindex-0
[device_container_node-2] Found rbdo mapped object: index=60/a subindex=6
[device_container_node-2] Found rpdo mapped object: index=6040 subindex=0
[device_container_node-2] Found rpdo mapped object: index=60ff subindex=0
[device container node-2] Found tpdo mapped object: index=6041 subindex=0
[device_container_node-2] Found todo mapped object: index=6061 subindex=0
[device_container_node-2] Found todo mapped object: index=6064 subindex=0
device container node 21 Found thdo manned object, index-606c subindex-0
device_container_node 2] Found that mapped object. Index-6041 subject=0
[device_container_node-2] Found that mapped object: index=0041 subindex=0
[device_container_node-2] Found tpdo mapped object: index=6064 subindex=0
[device_container_node-2] Found tpdo mapped object: index=6041 subindex=0
[device_container_node-2] Found tpdo mapped object: index=606c subindex=0
[device container node-2] [WARN] [1735955754.348027428] [cia402 device 1]: Wait for device to boot.
[INFO] [launch.user]: node 'basic slave node' reached the 'inactive' state. 'activating'.
[cia402 slave node-1] [INFO] [1735955754,428790332] [cia402 node 1]: Reaching inactive state.
[cia402_slave_pode_1] [INE0] [1735955754_432011517] [cia402_pode_1]. Beaching active state
[ciado2_clave_node_1] [INFO] [IT35357574,43201137] [Ciado2_node_1]. Reaching active state.
[C1a402_Stave_Node-1] [1NF0] [1/35955/54.430445125] [C1a402_Node_1]: Created C1a402_Stave Tor Node_1d 2,

ROS下 canopen 模块上抓取到的报文:

root@ly	s-vir	ual-machine:~/ros2/src/ros2_canopen/canopen_tests/launch# candump_can0_	
can0	080	[0]	

六叶树 USBCANFD-Mini 上接收到 canopen 模块发出的报文:



-				Can/	Ast(六叶树)				主题	
择设备	设置 语言	工具	帮助							
BCAN1 US	BCAN2 USBC	ANFDMini	USBCAN	IFD1 USBCANF						
USBCANFE	DMini 设备:0 通道	:0 区								
枵	传输方向	状态		时间标志	周期(us)	数量	帧ID	帧格式	0¢	类型
0	接收	正常		09:56:17:749	9791	8081	0x00000080	数据帧	杤	∂隹帧
1	接收	正常		09:54:57:534	0	1	0x00000701	数据帧	杤	€隹帧
					反山	1YHQX				
4					又山		_			>
磁发送高	现发送				(人)	1)17				>
磁发送 高	级发送 正常发送	▼ 数据(16	进制):				✓ ID递增)
础发送 高 送方式: 协议:		 数据(16 01 02 	进制): 03 04 05	06 07 08			 ✓ ID递增 每次发送帧数: 	1)
(確发送 高) 送方式: 物议: 学習:	级发送 正常发送 CAN 标准帧	 数据(16) 01 02 	进制): 03 04 05	06 07 08			 ✓ ID递增 每次发送帧数: 发送次数: 45%200万倍%(5) 	1	0	>
基础发送 高 送送方式: 频功议: 频型: 频格式: 前D(16进制)·	级发送 正常发送 CAN 标准帧 数据帧 00000001	▼ 数据(16 ▼ 01 02 ▼	进制): 03 04 05	06 07 08			✓ ID递增 每次发送帧数: 发送次数: 发送间隔(室秒):	1	0) 》

8 工程说明

1.该 canopen 开源工程包括很多功能包,还有很多用例,在此不再一一叙述。

root@lys-virtual-mac	hine:~/ros2/src/ros2_	canopen# ls		
canopen	canopen_core	canopen_master_driver	canopen_ros2_controllers	Dockerfile
canopen_402_driver	canopen_fake_slaves	canopen_proxy_driver	canopen_tests	lely_core_libraries
canopen base driver	canopen interfaces	canopen ros2 control	canopen utils	README.md

2.github 库地址

https://github.com/ros-industrial/ros2_canopen.git

3. 六 叶 树 已 发 布 该 工 程 的 Docker 镜 像 , 详 情 见 六 叶 树 ROS 机 器 人 系 统 专 栏:<u>http://www.liuyeshu.cn/?page_id=1893</u>

4.六叶树调试软件 CanAssistant 已支持 CANopen 协议数据的发送与接收,支持报文解析,支持 Cia301、Cia402 等协议,还支持伺服电机控制,可以很好的协助 CANopen 协议的设备的 调试。软件下载地址:<u>http://www.liuyeshu.cn/?page_id=492</u>

六叶树教育科技有限公司 ©版权所有



ANopen	EDS觀畫			·捷操作 CIA301 CIA402		
得	报文	解析	<u></u>	序号 CAN报文	说明	帮助?
9257	[Rx,S,D,0000080,0,]	SYNC同步报文		00 S,D,67F,2B 17 10 00 64 00 00 00	心跳包周期100ms	发送
9258	[Rx,S,D,0000080,0,]	SYNC同步报文		01 S,D,67F,2B 17 10 00 F4 01 00 00	心跳包周期500ms	发送
259	[Rx,S,D,0000080,0,]	SYNC同步报文		02 S,D,67F,2B 17 10 00 10 27 00 00	心跳包周期10s	发送
9260	[Rx,S,D,0000080,0,]	SYNC同步报文		03 S,D,67F,2B 17 10 00 00 00 00 00	关闭心跳包	发送
9261	[Rx,S,D,0000080,0,]	SYNC同步报文		04 S,D,67F,40 17 10 00 00 00 00 00	读取心跳包周期	发送
9262	[Rx.S.D.0000080.0.1	SYNC同步报文		05 S,D,000,01 7F	NMT启动命令	发送
9263	[Rx.S.D.00000080.0.1	SYNC同步报文		06 S,D,000,02 7F	NMT停止命令	发送
9264	IRx S D 0000080 01	SVNC同步报文		07 S,D,000,80 7F	NMT预操作状态	发送
9265	[Rx,S,D,0000080,0,]	SYNC同步报文		08 S,D,000,81 /F		波透
9266	[Rx,S,D,00000080,0,]	SYNC同步报文		10 S,D,618,23 12 10 00 00 01 00 C0	开启时间戳	发送
9267	[Rx,S,D,0000080,0,]	SYNC同步报文		11 S,D,618,23 12 10 00 00 01 00 00	关闭时间戳	发送
9268	[Rx,S,D,0000080,0,]	SYNC同步报文		12 S,D,67F,40 61 60 00 00 00 00 00	卖取工作状态寄存器	发送
9269	[Rx,S,D,00000080,0,]	SVNC同步报文		13 S,D,67F,40 FF 60 00 00 00 00 00	凌取目标速度寄存器	发送
9270	[Rx,S,D,0000080,0,]	SYNC同步报文		14 S,D,67F,40 83 60 00 00 00 00 00	读取加速度斜率	发送
9271	[Rx,S,D,0000080,0,]	SYNC同步报文		15 S,D,67F,2B 40 60 00 8F 01 00 00	重置伺服点击错误	发送
272	[Rx S.D.0000080.0.1	SYNC同步报文		16 S,D,67F,2F 60 60 00 03 00 00 00	进入速度模式	发送
272	IR- 5 D 00000080 0 1	CVNC 同共招立		17 S,D,67F,40 60 60 00 00 00 00 00	读取工作模式寄存器	发送
273	[Rx, 5, 0, 00000000,0,]	CVALCELLER		18 S,D,67F,2B 40 60 00 06 00 00 00	速度模式启动步骤1	发送
52/4	[rx, 5, D, 00000080, 0,]	SYINCIDEPIDX		19 S,D,67F,2B 40 60 00 07 00 00 00	速度模式启动步骤2	发送

当前设备:USBCANFDMini-0-0

CANopen EDS酸素 CIA301 CIA402									
茅号	报文	解析	Node-ID: 2 (有效值:10进制数1-127)						
1018	[Rx,S,D,0000080,0,]	SYNC同步报文							
019	[Rx,S,D,00000080,0,]	SYNC同步报文	COB-ID=0x00						
020	[Rx,S,D,0000080,0,]	SYNC同步报文	远程爆作 启动 👻 发送						
021	[Rx,S,D,0000080,0,]	SYNC同步报文	CNMT节点守护						
.022	[Rx,S,D,0000080,0,]	SYNC同步报文	状态查询: 查询						
.023	[Rx,S,D,0000080,0,]	SYNC同步报文	状态应答: 启动状态 👻 发送						
024	[Rx,S,D,0000080,0,]	SYNC同步报文	C SYNC命令						
.025	[Rx,S,D,00000080,0,]	SYNC同步报文	COB-ID=0x80						
.026	[Rx,S,D,0000080,0,]	SYNC同步报文	远程操作 发送同步报文 * 次数: 32 发送						
.027	[Rx,S,D,00000080,0,]	SYNC同步报文	EMERG命令						
028	[Rx,S,D,0000080,0,]	SYNC同步报文	COB-ID=0x81 资告进注(产研28VTE): 进注案方理(18VTE):						
.029	[Rx,S,D,0000080,0,]	SYNC同步报文	自定义错误代码(5BYTE): 发送						
030	[Rx,S,D,00000080,0,]	SYNC同步报文							
.031	[Rx,S,D,0000080,0,]	SYNC同步报文	PDO命令 类型: TPDO1 >						
.032	[Rx,S,D,00000080,0,]	SYNC同步报文							
.033	[Rx,S,D,0000080,0,]	SYNC同步报文							

伺服控制:



ANopen	EDS配置		快捷操作 CIA301 CIA402
茅号	报文	解析	Node-ID: 127 (有效值:10进制数1-127
3041	[Rx,S,D,0000080,0,]	SYNC同步报文	
3042	[Rx,S,D,0000080,0,]	SYNC同步报文	
3043	[Rx,S,D,0000080,0,]	SYNC同步报文	节点状态:
3044	[Rx,S,D,0000080,0,]	SYNC同步报文	非操作状态
3045	[Rx,S,D,0000080,0,]	SYNC同步报文	□ 设备操作
3046	[Rx,S,D,0000080,0,]	SYNC同步报文	
3047	[Rx,S,D,0000080,0,]	SYNC同步报文	启动 停止 重置
3048	[Rx,S,D,0000080,0,]	SVNC同步报文	
3049	[Rx,S,D,0000080,0,]	SVNC同步报文	「伺服控制」 海府描述 Intering (·罗博士)
3050	[Rx,S,D,0000080,0,]	SYNC同步报文	
3051	[Rx,S,D,0000080,0,]	SVNC同步报文	目标速度 1000 转/分钟
3052	[Rx,S,D,0000080,0,]	SVNC同步报文	加速速度 200 转/分钟
3053	[Rx,S,D,0000080,0,]	SYNC同步报文	减速速度 200 转/分钟
3054	[Rx,S,D,0000080,0,]	SYNC同步报文	开始结束
3055	[Rx,S,D,0000080,0,]	SYNC同步报文	
3056	[Rx,S,D,00000080,0,]	SYNC同步报文	

当前设备:USBCANFDMini-0-0



9 常用链接

六叶树官网:<u>www.liuyeshu.cn</u>

六叶树 CanAssistant 调试软件下载地址:<u>http://www.liuyeshu.cn/?page_id=492</u>
六叶树 USBCANFD 产品驱动下载地址:<u>http://www.liuyeshu.cn/?page_id=485</u>
六叶树 USBCANFD 适配器产品二次开发资料下载:<u>http://www.liuyeshu.cn/?page_id=440</u>
六叶树 USBCAN 适配器产品 SOCKETCAN 专栏资料下载:<u>http://www.liuyeshu.cn/?page_id=1256</u>
六叶树 USBCAN 适配器产品 ROS 机器人操作系统专栏:<u>http://www.liuyeshu.cn/?page_id=1893</u>
六叶树 USBCANFD 产品常见软硬件问题解决方案专栏:<u>http://www.liuyeshu.cn/?page_id=593</u>

10 联系方式

- (2) 电话:15211065817(业务合作咨询)
- 🖂 邮箱:798746621@qq.com(业务咨询+技术支持)
- 险 微信:18673379565(技术支持)
- 🔁 官网:<u>www.liuyeshu.cn(</u>资料下载)
- 海 网上商城:<u>https://shop112408209.taobao.com</u>(产品购买)

淘宝店铺搜索:"六叶树教育科技"