



六叶树产品教程

ROS CANopen 工程移植教程

类别	内容
关键词	ROS 机器人系统下 ARS_40X 雷达调试 环境搭建
摘要	如何在 ROS 系统下移植开源 CANopen 工程

库介绍:

类别	内容
安装环境	Ubuntu20.04
ROS 版本	ROS noetic
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

六叶树产品教程	1
ROS CANopen 教程	1
1、环境准备	3
2、系统环境准备	4
3、CANopen 工程移植	5
4、六叶树适配器驱动安装	7
5、CAN 卡调试工具安装	7
6、硬件接线	7
7、调试步骤	8
8 常用链接	13
9 联系方式	13

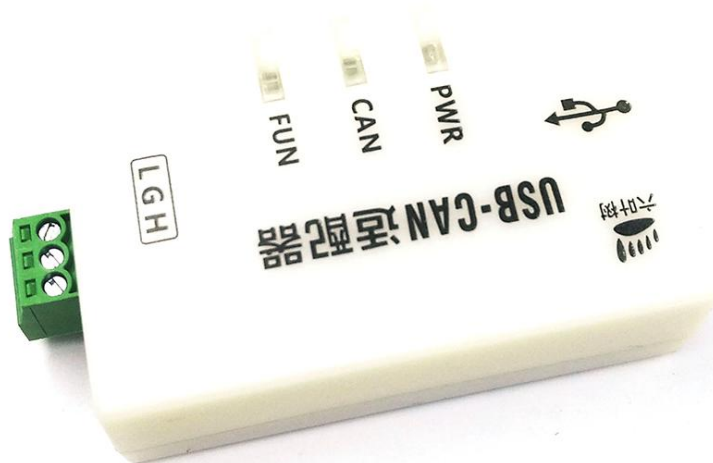


1、环境准备

系统版本	Ubuntu20.04	
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	ROS neotic 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 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	
调试软件	六叶树 CanAssistant 软件下载地址:www.liuyeshu.cn	

```
root@ubuntu:/home/zhumiao/tmp# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 20.04.6 LTS
Release:      20.04
Codename:     focal
```

本教程使用六叶树 UTC2201 适配器实验，免驱，简单易用。



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/ focal main restricted universe multiverse
```

```
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
```

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

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



```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse multiverse
```

```
#添加清华源
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ / focal main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse 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 libmuparser-dev

3.2 工程源码下载与编译

1.初始化 ROS 的工作空间
mkdir -p ~/ros_catkin_ws/src
cd ~/ros_catkin_ws/src



ROS CANopen 环境搭建

2. 克隆 ROS CANopen 到工作空间:

方式一: `git clone https://github.com/ros-industrial/ros_canopen.git`

方式二:

`wget`

`http://www.liuyeshu.cn/wp-content/uploads/2024/12/ros_canopen-melodic-devel.20250102.zip`

`unzip ros_canopen-melodic-devel.20250102.zip`

`mv ros_canopen-melodic-devel ros_canopen`

```
root@ubuntu:~/ros_catkin_ws/src/ros_canopen# ls
can_msgs      canopen_chain_node  canopen_motor_node  LICENSE      ros_canopen      socketcan_interface
canopen_402   canopen_master      CONTRIBUTING.md      README.md    socketcan_bridge
root@ubuntu:~/ros_catkin_ws/src/ros_canopen#
```

3. 构建并安装: 回到 `~/ros_catkin_ws` 并运行 `catkin_make`。

`cd ..`

`catkin_make`

```
####
#### Running command: "cmake /root/lys/src -DCATKIN_DEVEL_PREFIX=/root/lys/devel -DCMAKE_INSTALL_PREFIX=/root/lys/install -G Unix Makefiles"
#### in "/root/lys/build"
####
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /root/lys/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
```

```
[ 75%] Building CXX object ros_canopen/canopen_chain_node/CMakeFiles/canopen_ros_chain.dir/src/rosconsole_bridge.cpp.o
Scanning dependencies of target canopen_chain_node_generate_messages
[ 75%] Built target canopen_chain_node_generate_messages
Scanning dependencies of target socketcan_to_topic_node
[ 77%] Building CXX object ros_canopen/socketcan_bridge/CMakeFiles/socketcan_to_topic_node.dir/src/socketcan_to_topic_node.cpp.o
[ 78%] Linking CXX executable /root/lys/devel/lib/socketcan_bridge/socketcan_to_topic_node
[ 78%] Built target socketcan_to_topic_node
Scanning dependencies of target socketcan_bridge_node
[ 80%] Building CXX object ros_canopen/socketcan_bridge/CMakeFiles/socketcan_bridge_node.dir/src/socketcan_bridge_node.cpp.o
[ 81%] Linking CXX shared library /root/lys/devel/lib/libcanopen_ros_chain.so
[ 81%] Built target canopen_ros_chain
Scanning dependencies of target canopen_402_plugin
[ 83%] Building CXX object ros_canopen/canopen_402_plugin/CMakeFiles/canopen_402_plugin.dir/src/plugin.cpp.o
[ 84%] Linking CXX executable /root/lys/devel/lib/socketcan_bridge/socketcan_bridge_node
[ 84%] Built target socketcan_bridge_node
Scanning dependencies of target canopen_chain_node
[ 86%] Building CXX object ros_canopen/canopen_chain_node/CMakeFiles/canopen_chain_node.dir/src/chain_node.cpp.o
[ 87%] Linking CXX shared library /root/lys/devel/lib/libcanopen_402_plugin.so
[ 87%] Built target canopen_402_plugin
Scanning dependencies of target canopen_motor
[ 89%] Building CXX object ros_canopen/canopen_motor_node/CMakeFiles/canopen_motor.dir/src/controller_manager_layer.cpp.o
[ 90%] Linking CXX executable /root/lys/devel/lib/canopen_chain_node/canopen_chain_node
[ 90%] Built target canopen_chain_node
[ 92%] Building CXX object ros_canopen/canopen_motor_node/CMakeFiles/canopen_motor.dir/src/handle_layer.cpp.o
[ 93%] Building CXX object ros_canopen/canopen_motor_node/CMakeFiles/canopen_motor.dir/src/motor_chain.cpp.o
[ 95%] Building CXX object ros_canopen/canopen_motor_node/CMakeFiles/canopen_motor.dir/src/robot_layer.cpp.o
[ 96%] Linking CXX shared library /root/lys/devel/lib/libcanopen_motor.so
[ 96%] Built target canopen_motor
Scanning dependencies of target canopen_motor_node
[ 98%] Building CXX object ros_canopen/canopen_motor_node/CMakeFiles/canopen_motor_node.dir/src/canopen_motor_chain_node.cpp.o
[100%] Linking CXX executable /root/lys/devel/lib/canopen_motor_node/canopen_motor_node
[100%] Built target canopen_motor_node
```




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 CANFD2	需安装对应的驱动
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:http://www.liuyeshu.cn/?page_id=1256

```
root@ubuntu:/home/zhumiao/lys/ros# ifconfig -a
can0: flags=128<NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.106 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::909c:55e0:df8e:9398 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:d3:18:8f txqueuelen 1000 (Ethernet)
    RX packets 282823 bytes 246535244 (246.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 193014 bytes 16182603 (16.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 23114 bytes 1975421 (1.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23114 bytes 1975421 (1.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@ubuntu:/home/zhumiao/lys/ros#
```

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

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
```

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:

```
can0 702 [5] 04 00 00 06 38
can0 702 [5] 05 B6 20 33 21
can0 702 [5] 06 B6 60 33 21
can0 702 [5] 07 00 00 06 38
can0 702 [5] 08 B6 A0 33 21
can0 702 [5] 09 B6 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
```

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 启动 ROS 服务

直接执行 roscore

```
root@ubuntu:/home/lys# roscore
... logging to /root/.ros/log/33ddacd2-c8a6-11ef-be76-37b88bc32c53/roslaunch-ubuntu-9020.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:36105/
ros_comm version 1.17.0

SUMMARY
=====

PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.17.0

NODES

auto-starting new master
process[master]: started with pid [9028]
ROS_MASTER_URI=http://ubuntu:11311/

setting /run_id to 33ddacd2-c8a6-11ef-be76-37b88bc32c53
process[roscout-1]: started with pid [9038]
started core service [/roscout]
```

7.4 测试 socketcan_bridge 服务

7.4.1 启动 socketcan_bridge 服务

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。

```
root@ubuntu:~/ros_catkin_ws# ls
build devel src
root@ubuntu:~/ros_catkin_ws#
```

source devel/setup.bash

```
root@ubuntu:~/ros_catkin_ws# source devel/setup.bash
root@ubuntu:~/ros_catkin_ws#
```

roslaunch socketcan_bridge socketcan_to_topic_node

启动成功如上图所示。

```
root@ubuntu:~/ros_catkin_ws# roslaunch socketcan_bridge socketcan_to_topic_node
[ INFO] [1735780398.645232259]: Successfully connected to can0.
```

确保之前已经进行了六叶树 CAN 卡配置,不然会出现以下错误:

```
root@ubuntu:~/ros_catkin_ws# roslaunch socketcan_bridge socketcan_to_topic_node
[FATAL] [1735780280.522151551]: Failed to initialize can_device at can0
```



7.4.2 启动 ROS 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,进入到工程目录。

```
source devel/setup.bash
```

```
rostopic echo /received_messages
```

```
root@ubuntu:~/ros_catkin_ws# source devel/setup.bash
root@ubuntu:~/ros_catkin_ws#
root@ubuntu:~/ros_catkin_ws#
root@ubuntu:~/ros_catkin_ws# rostopic echo /received_messages
```

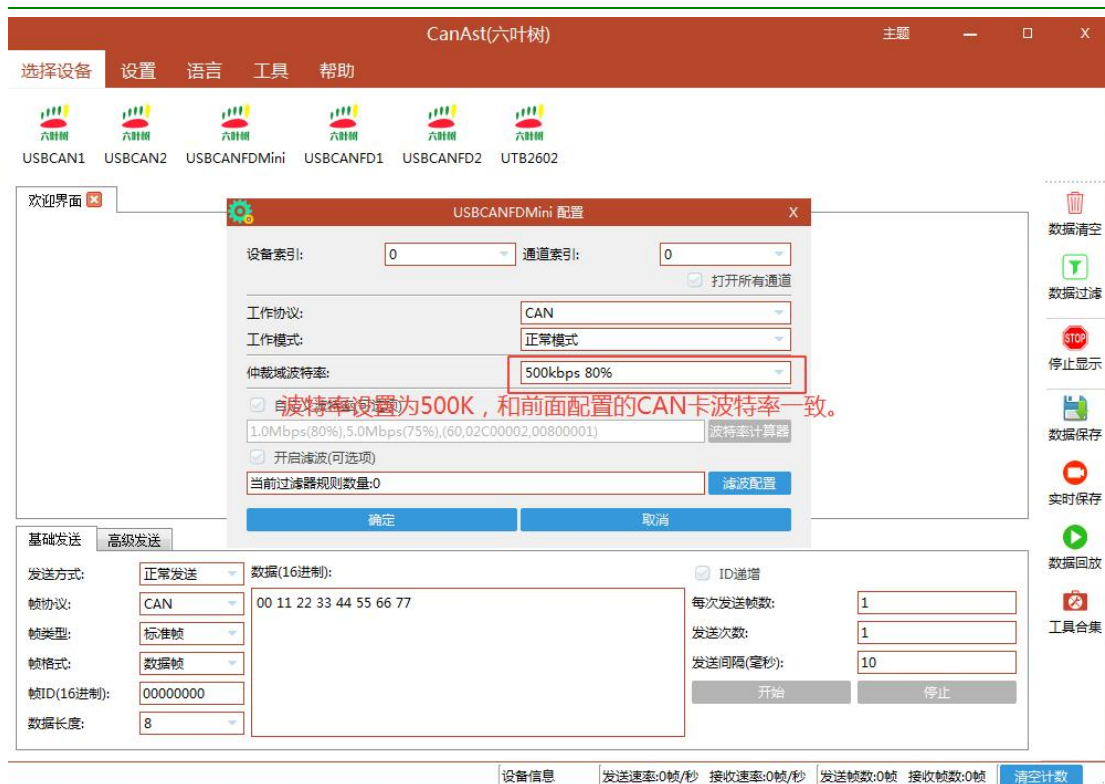
7.4.3 往 CAN 总线发数据

这里我们使用六叶树的 USBCANFD-Mini 往 CAN 总线上发送报文(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 盒我们是用来发送数据辅助测试的)。

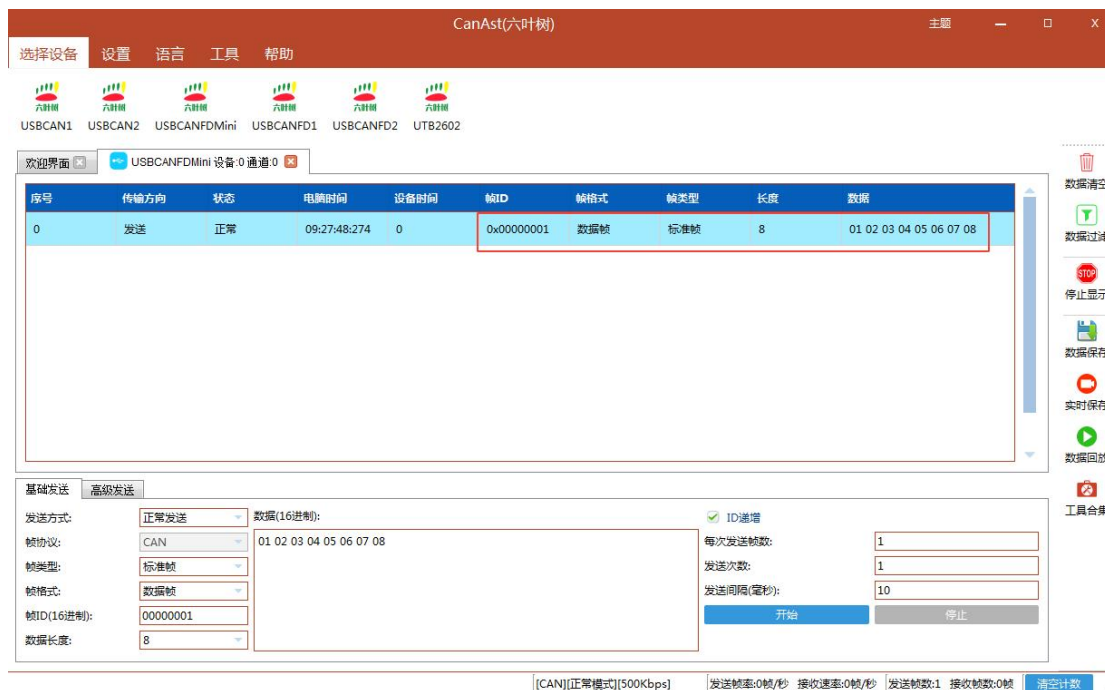




ROS CANopen 环境搭建



发送的数据:



Socketcan_bridge 桥接的数据:



ROS CANopen 环境搭建

```
root@ubuntu:/home/lys#  
root@ubuntu:/home/lys# ls  
Desktop Documents Downloads Music Pictures Public ros Templates Videos  
root@ubuntu:/home/lys# cd /root/  
root@ubuntu:~# ls  
ros_catkin_ws snap  
root@ubuntu:~# cd ros_catkin_ws/  
root@ubuntu:~/ros_catkin_ws# ls  
build devel src  
root@ubuntu:~/ros_catkin_ws# source devel/  
.built_by env.sh local_setup.bash .rosinstall _setup_util.py  
.catkin include/ local_setup.sh setup.bash setup.zsh  
cmake.lock lib/ local_setup.zsh setup.sh share/  
root@ubuntu:~/ros_catkin_ws# source devel/  
.built_by env.sh local_setup.bash .rosinstall _setup_util.py  
.catkin include/ local_setup.sh setup.bash setup.zsh  
cmake.lock lib/ local_setup.zsh setup.sh share/  
root@ubuntu:~/ros_catkin_ws# source devel/setup.bash  
root@ubuntu:~/ros_catkin_ws#  
root@ubuntu:~/ros_catkin_ws#  
root@ubuntu:~/ros_catkin_ws# rostopic echo /received_messages  
header:  
  seq: 0  
  stamp:  
    secs: 1735781324  
    nsecs: 815421409  
  frame_id: ''  
id: 1  
is_rtr: False  
is_extended: False  
is_error: False  
dlc: 8  
data: [1, 2, 3, 4, 5, 6, 7, 8]  
---
```

工程代码里还有很多其他用例包括伺服电机、Cia402 等，这里不再一一列举。

```
root@ubuntu:~/ros_catkin_ws/src/ros_canopen# ls  
can_msgs canopen_chain_node canopen_motor_node LICENSE ros_canopen socketcan_interface  
canopen_402 canopen_master CONTRIBUTING.md README.md socketcan_bridge  
root@ubuntu:~/ros_catkin_ws/src/ros_canopen#
```



8 常用链接

六叶树官网:www.liuyeshu.cn

六叶树 CanAssistant 调试软件下载地址:http://www.liuyeshu.cn/?page_id=492

六叶树 USBCANFD 产品驱动下载地址:http://www.liuyeshu.cn/?page_id=485

六叶树 USBCANFD 适配器产品二次开发资料下载:http://www.liuyeshu.cn/?page_id=440

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

六叶树 USBCAN 适配器产品 ROS 机器人操作系统专栏:http://www.liuyeshu.cn/?page_id=1893

六叶树 USBCANFD 产品常见软硬件问题解决方案专栏:http://www.liuyeshu.cn/?page_id=593

9 联系方式



电话:15211065817(业务合作咨询)



邮箱:798746621@qq.com(业务咨询+技术支持)



微信:18673379565(技术支持)



官网:www.liuyeshu.cn(资料下载)



网上商城:<https://shop112408209.taobao.com>(产品购买)

淘宝店铺搜索:“六叶树教育科技有限公司”