目录

一、新建一个 ROS 工作空间

1. 创建 catkin 工作空间

```
$ mkdir -p ~/catkin_ws/src
```

2. 编译 catkin 工作空间

```
$ cd ~/catkin_ws
$ catkin_make
```

3. 配置环境变量

```
$ echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
```

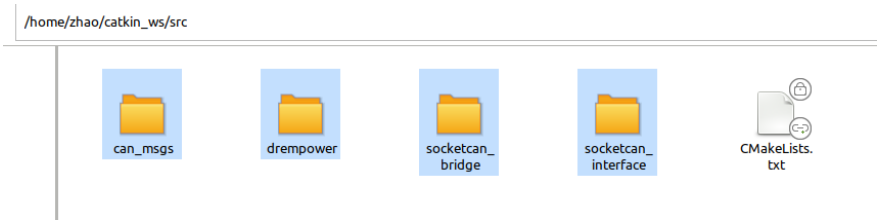
4. 创建工作包

```
$ cd ~/catkin_ws/src
$ catkin_create_pkg drempower std_msgs roscpp rospy
```

二、拷贝电机控制相关功能包并重新编译

1. 替换 catkin_ws/src 文件夹

●删除~/catkin_ws/src 文件夹下的 drempower 文件夹，然后将电机 ROS 库函数文件夹下的 DrEmpower_ws/src 文件夹下的 4 个文件夹复制到~/catkin_ws/src 文件夹中；



2.替换 catkin_ws/devel/include 文件夹

●将电机 ROS 库函数文件夹下的 DrEmpower_ws/devel/include 文件夹整个复制到~/catkin_ws/devel 文件夹中；

3. 编译整个功能包

```
$ cd ~/catkin_ws/
$ catkin_make
```

三、修改并运行示例程序

●在示例程序中提供了单个电机 7 种控制模式测试，实际使用时根据实际情况修改测试的电机 ID 号和测试模式；

模式名称		模式简称	数字代号	注释
位置控制	轨迹跟踪模式	TP	7	
	梯形轨迹模式	PP	1	
	前馈控制模式	FP	-1	
速度控制	速度爬升模式	PV	3	
	速度前馈模式	FV	-3	
扭矩控制	扭矩爬升模式	PT	4	
	直接控制模式,	FT	-4	
阻抗控制模式		IP	-7	

以上 7 种控制模式更详细的介绍请参考产品手册及使用说明书中的《DrEmpower 系列电机使用手册 v1.0》

1. 修改 catkin_ws/src/drempower/src/motor_test_publisher.cpp 文件

●根据实际电机 ID 号及想要测试的模式修改第 14、15 行，如下图所示；

```
src > drempower > src > motor_test_publisher.cpp > ...
  3  #include "drempower/pv_msg.h"
  4  #include "drempower/pt_msg.h"
  5  #include "drempower/tp_msg.h"
  6  #include "drempower/ip_msg.h"
  7  #include "drempower/fp_msg.h"
  8  #include "drempower/fv_msg.h"
  9  #include "drempower/ft_msg.h"
 10  #include "drempower/property_msg.h"
 11  #include <sstream>
 12
 13  #define MOTOR_NUM 1
 14  #define MOTOR_ID 1 // 根据电机ID号进行修改
 15  #define TP_MODE    // 通过修改这一项测试不同模式
```

图 1 修改测试节点文件中的电机 ID 及控制模式

2. 修改 catkin_ws/src/drempower/src/property_publisher.cpp 文件

●根据实际电机 ID 号及想要测试的模式修改第 6、7 行，如下图所示；

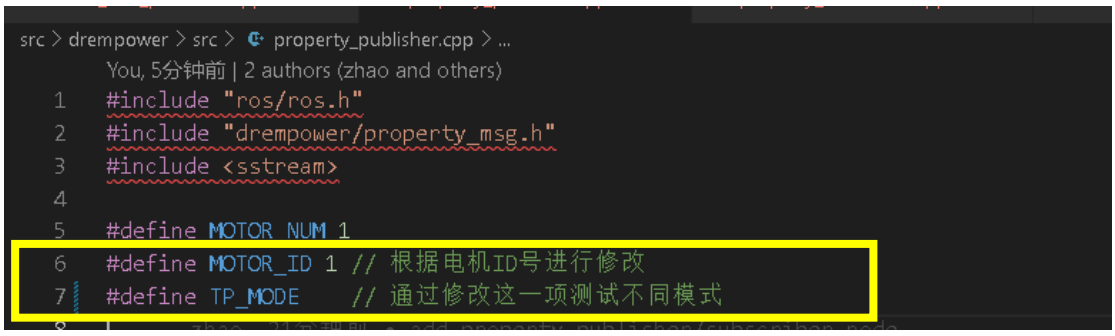


图 1 修改读取属性节点文件中的电机 ID 及控制模式

■注意：示例程序目前一次只能测试一个电机和一种特定控制模式；

四、重新编译整个功能包

```
$ cd ~/catkin_ws/
$ catkin_make
```

五、启动 launch 文件

```
$ roslaunch drempower motor_control_test.launch
```

注意：

■1. 启动 launch 文件前需要先将电机供电，同时将 USB 转 CAN 模块连接好电脑和电机，且将电机 CAN 波特率设置为 1000000，如图 3 所示。

■2. 部分用户可能出现 can 发送失败的情况（下图中的 USB 转 CAN 模块蓝色指示灯未正常亮起），此时可能是由于用户权限导致，可使用下面的指令先切换到 root 权限，然后再次运行上面的指令。

■3. 有时会出现提示“RLException: [motor_control_test.launch] is neither a launch file in package [drempower] nor is [drempower] a launch file name The traceback for the exception was written to the log file”错误，只需关掉当前 terminal 窗口，新建一个 terminal 窗口再次运行 roslaunch drempower motor_control_test.launch.

```
$ sudo -s
```

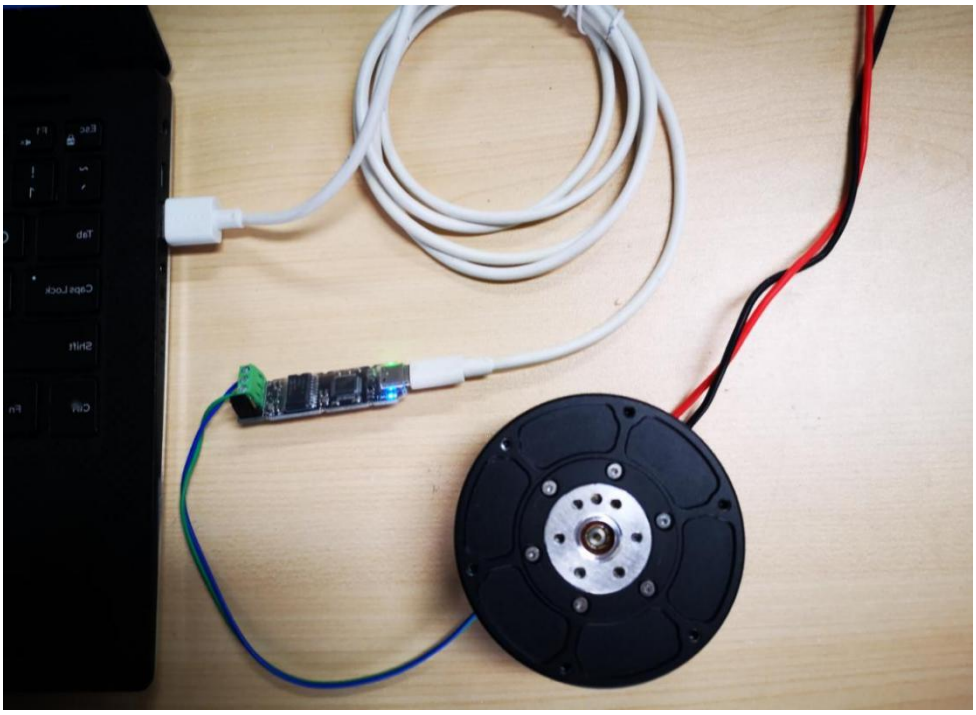


图 3a 电机接线图（**必须使用图中所示的 USB 转 CAN 模块**）

图 3b 使用上位机软件修改 CAN 波特率（**上位机 linux 版本已适配**）

六、使用 rqt_plot 查看电机运行曲线图

●轨迹跟踪模式（TP）下，测试程序控制指令以 100Hz 发送，同时默认使能实时状态返回功能(axis0.config.extra_setting.enable_reply_state= 1)，关节电机每接收到一条控制指令后，立即自动返回一条电机当前实时状态数据，该数据被

state_subscriber_node 节点接收解析，并向外发送 Topic(motor_state)，该 topic 采用 state_msg.msg 类型，包含电机 id 号，实时电机位置 pos_estimate（度）、vel_estimate 电机速度(r/min)、torque_estimate 电机扭矩(Nm)、traj_done、axis_error；

●梯形轨迹模式（PP）下，测试程序默认关闭实时状态返回功能，同时控制指令发送由 axis0.controller.trajectory_done 标志位控制，当电机未到达目标位置时，此标志位值为 0，在到达目标位置后，该标志位为 1。property_publisher_node 以 10Hz 频率发送 Topic(read_property)来读取该标志位，以 100Hz 频率读取 axis0.output_shaft.pos_estimate 属性。电机返回的属性数据会被 property_subscriber_node 接收，其中 axis0.controller.trajectory_done 属性同步更新到 trajectory_done 变量中，axis0.output_shaft.pos_estimate 属性通过 Topic(pos_estimate)向外发送，该 topic 采用 property_msg.msg 类型，包含电机 id 号，属性索引地址 address、属性数据类型 data_type、属性当前值 value；

（最新版本已改成使用实时状态快速读取接口实现，property_publisher_node 以 100Hz 频率读取实时状态（address=0x00），电机返回的实时状态数据被 state_subscriber_node 节点接收解析，并将 traj_done 同步更新到 trajectory_done 变量）

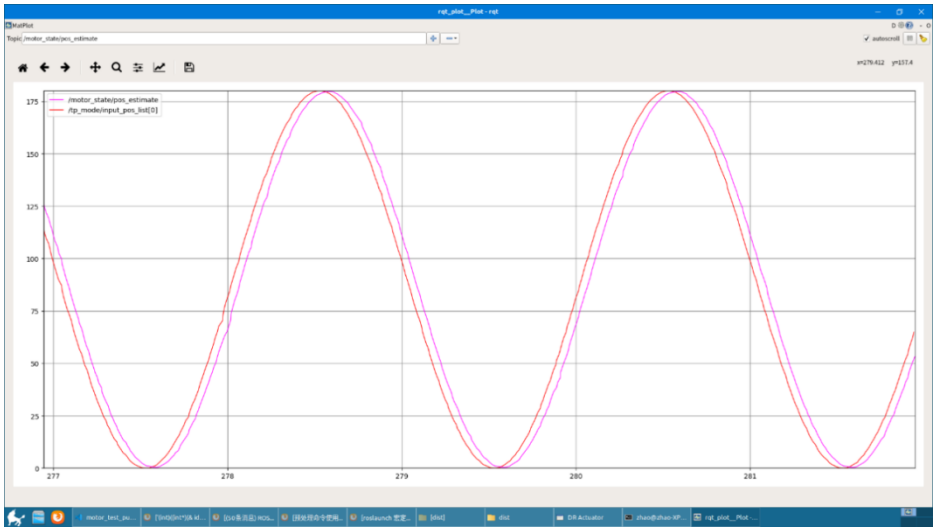
●其他 6 种模式下，测试程序每 0.5Hz 频率发送控制指令，同时默认关闭实时状态返回功能。property_publisher_node 以 100Hz 频率发送 Topic(read_property)来读取 axis0.output_shaft.pos_estimate 或 axis0.output_shaft.vel_estimate 或 axis0.output_shaft.torque_estimate，电机返回的属性数据会被 property_subscriber_node 接收，相应通过 Topic(pos_estimate 或 vel_estimate 或 torque_estimate)向外发送。

●重新开启一个 terminal（快捷键 Ctrl+alt+T），输入 rqt_plot 指令，在弹出的窗口里选择 topic(/motor_state 或/pos_estimate 或/vel_estimate 或/torque_estimate)即可，如图 4 所示；

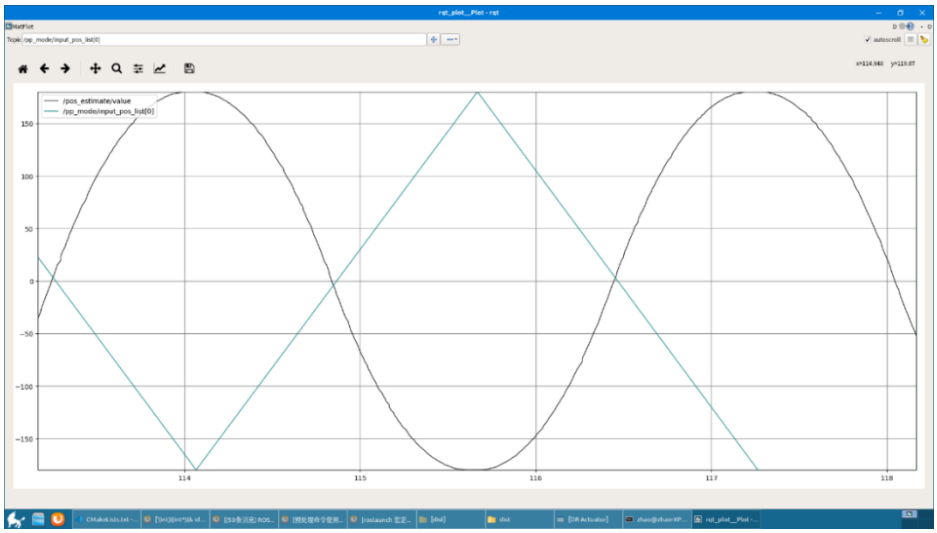
```
$ rqt_plot
```

图 4 启动 rqt_plot 并选择 motor_state 话题

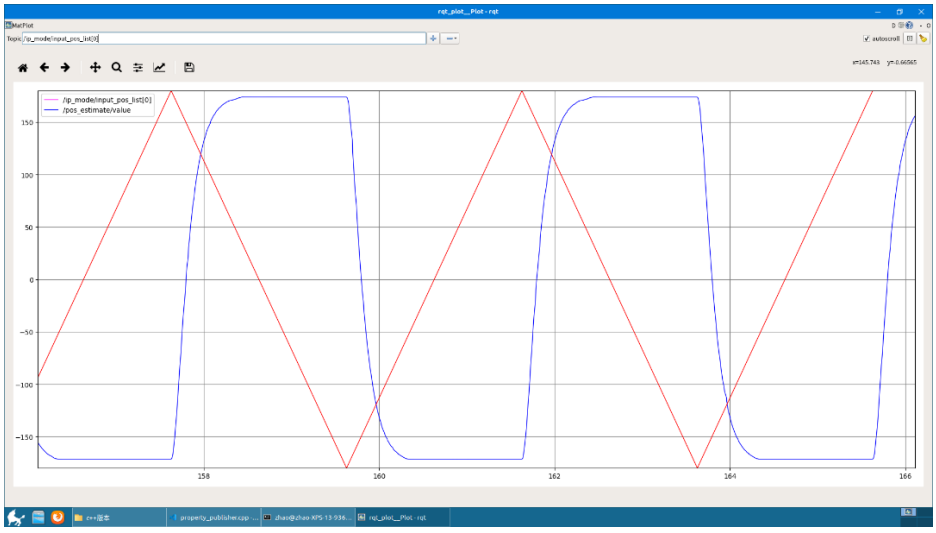
●轨迹插补模式、梯形轨迹模式、阻抗控制模式下电机运行曲线图如图 5 所示：



(a)轨迹跟踪模式



(b)梯形轨迹模式



(c)阻抗控制模式

图 5 三种控制模式下电机运行状态曲线图

七、使用 rqt_graph 查看节点及消息图

●重新开启一个 terminal（快捷键 Ctrl+alt+T），输入 rqt_graph 指令即可得到当前 ROS 节点及节点间消息交互动态图形，如图 6 所示

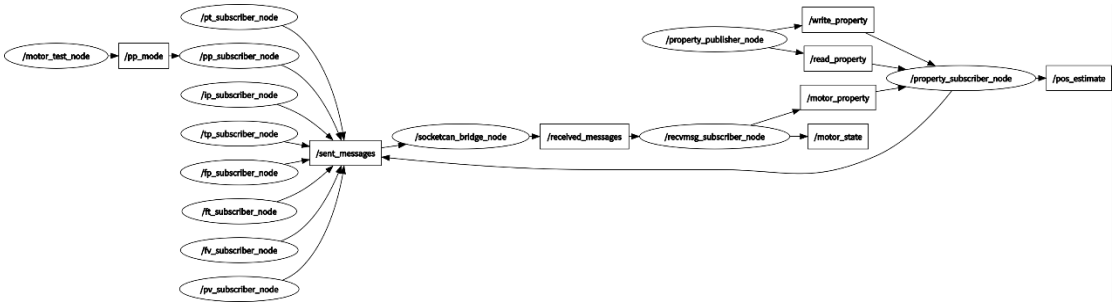


图 6 示例程序下 ROS 节点及消息动态图

●从图中可以看出整个示例程序中主要有以下几个节点：motor_test_node、xx_subscriber_node 、 socketcan_bridge_node 及 recvmsg_subscriber_node 、property_publisher_node 和 property_subscriber_node。其中：

1. motor_test_node 节点对应库文件 motor_test_publisher.cpp，其主要作用发送对应的控制指令，图中测该节点按照 pp_msg.msg 中规定消息类型发送控制指令 pp_node 消息；

2. xx_subscriber_node 节点对应库文件中 xx_subscriber.cpp，一共有 8 个，对应 8 中控制模式，其主要作用是将 xx_msg.msg 格式的 xx_node 消息根据通信协议解析成 can_msgs/Frame.msg 格式的 sent_message 消息;

3. socketcan_bridge_node 节点是 ROS 标准库 Ros_canopen 库里的子节点，主要有两个作用，一是将 can_msgs/Frame.msg 格式的 sent_message 消息通过 socketcan 设备发送到 CAN 总线上；另外一个作用是将 socketcan 设备接收的 CAN 数据转换成 can_msgs/Frame.msg 格式的 received_message 消息；

4. recvmsg_subscriber_node 节点对应库文件中 recvmsg_subscriber.cpp，其主要作用是解析由 socketcan_brideg_node 节点转换后的 received_message 消息，这些消息是由电机发送给上位机，包括电机运行状态和属性参数信息。recvmsg_subscriber_node 节点将解析后的电机运行状态转换成 state_msg.msg 格式的 motor_state 消息，将电机属性参数转换成 property_msg.msg 格式的 motor_property 消息，该消息可以用 rqt_plot 工具绘制实时电机运行曲线，也可以用在用户程序里来了解电机实时运行状态等；

5. property_publisher_node 节点对应库文件中 property_publisher.cpp 文件，主要作用发送读取或修改属性指令。当需要读取某个属性时，只需按照 property_msg.msg 格式，配置好电机 id 及属性 address 和 data_type 向外发送 Topic(read_property)

即可；当需要修改某个属性时，只需按照 property_msg.msg 格式，配置好电机 id 及属性 address、data_type 和 value 向外发送 Topic(write_property)即可；

6. property_subscriber_node 节点对应库文件中 property_subscriber.cpp 文件，主要作用有两个，一个是接收 read_property 和 write_property 消息，根据通信协议解析成 can_msgs/Frame.msg 格式的 sent_message 消息；另一个是接收电机返回的 motor_property 消息，对其进行二次处理（motor_property 消息包含不同电机不同消息，无法直接使用），例如将 axis0.output_shaft.pos_estimate 属性从 motor_property 消息中挑选出来，通过 pos_estimate 消息向外发送，便于 rqt_plot 等程序使用；

注：property_publisher.cpp 和 property_subscriber.cpp 文件中属性对应的索引地址 address 和数据类型 data_type 可通过产品资料中 6-其他资料文件夹下的《DrEmpower 系列电机常用参数及地址表》进行查询。