8. Controlling robot motion

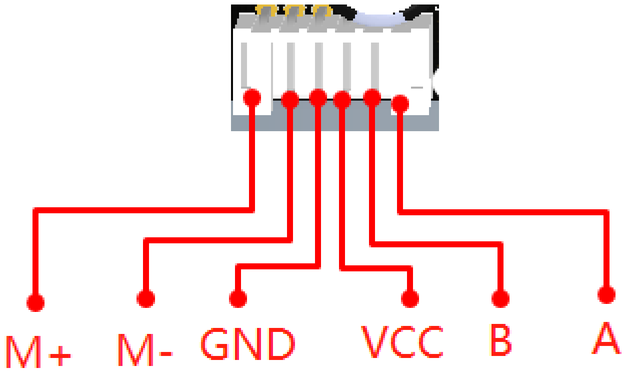
8.1. Experimental Objectives

To control the forward and backward movement on the Rosmaster, to the left and to the right, and to set the relevant parameters of the cart.

8.2. Experiment Preparation

The motor interface has the function of anti-reverse connection, use the motor wiring of Rosmaster to connect to the motor.

The corresponding interface wiring sequence of the motor is shown in the following figure:



Rosmaster_Lib library functions needed to control Rosmaster's motors:

```
set_car_motion(v_x, v_y, v_z)
```

Parameter Explanation: Trolley motion control, this function will read the pulse data from the encoder, so as to calculate the speed of the trolley motion. According to the different trolley there may be differences, here to McNamee wheel trolley as an example: v_x indicates the longitudinal speed of the trolley, the unit is m/s, the positive number is forward, the negative number is backward; v_y control indicates the transverse speed of the trolley, the unit is m/s, the positive number is to the left, the negative number is to the right; v_z indicates the rotational speed of the trolley, the unit is rad/s, the positive number is to the left rotation, the negative number is to the right rotation.

v_x=[-1.0, 1.0], v_y=[-1.0, 1.0], v_z=[-5.0, 5.0]

Return value: none.

```
set_pid_param(kp, ki, kd, forever=False)
```

Parameter Explanation: The PID parameter control affects how the set_car_motion function controls the speed change of the car's motion. By default, it can not be adjusted.

kp ki kd = [0, 10.00], can input decimal.

permanent = True to save permanently, = False to act temporarily.

Permanent save is to write the data into the Flash of the microcontroller chip, the data will not be lost after reboot, the writing time is long, so add delay delay time to avoid the problem of causing the microcontroller to lose packets. Temporary role of the response is fast, single effective, restart the microcontroller chip after the data is no longer maintained.

Return value: none.

```
set_auto_report_state(enable, forever=False)
```

Parameter explanation: microcontroller automatically returns data status bit, default is on, if set off will affect the read data function.

enable=True, the underlying expansion board will send data every 40 milliseconds. enable=False, it will not send.

permanent=True to save permanently, =False to act temporarily.

Return value: none.

```
clear_auto_report_data()
```

Parameter Explanation: Clear the cached data automatically sent over by the microcontroller.

Return Value: None.

```
reset_flash_value()
```

Parameter explanation: reset the data saved in the cart flash and restore the factory default value. This function can also be achieved by long pressing the K2 key on the expansion board for about 10 seconds.

Return value: None.

The following function functions all return data, and need to be started normally by create_receive_threading() to read the data:

```
get_accelerometer_data()
```

Parameter explanation: Get accelerometer data in three axes

Return value: a_x, a_y, a_z

```
get_gyroscope_data()
```

Parameter explanation: Get the data of three axes of gyroscope

Return value: g_x, g_y, g_z

```
get_motion_data()
```

Parameter explanation: get the speed of the cart, return val_vx, val_vy, val_vz

Return Value: val_vx, val_vy, val_vz

```
get_motion_pid()
```

Parameter explanation: Get the motion PID parameter of the cart, return [-1, -1, -1] for reading error

Return value: kp, ki, kd

8.3. Experimental results

Refer to 4.3.1, 4.3.2 to check the ROS expansion board as well as enter the Docker container, and then run the program, in the jupyter lab interface, double-click to enter the /root/yahboomcar_ros2_ws/Rosmaster/Sample, double-click to select the 5.pwm_servo.ipynb, and then step-by-step click on the 4.3. 3 The button shown in the figure operates the program.

8.4 Program Source Code

Enter the docker, refer to the code path:
/root/yahboomcar_ros2_ws/Rosmaster/Sample/8.car_motion.ipynb