

ROSYZ-02 Robot Platform

USER MANUAL



Rev: 2.0(Dec 2019)

SHENZHEN YZ ROBOT CO. LTD

§1 Brief Introduction

§1.1 ROSYZ-02 Basic Information

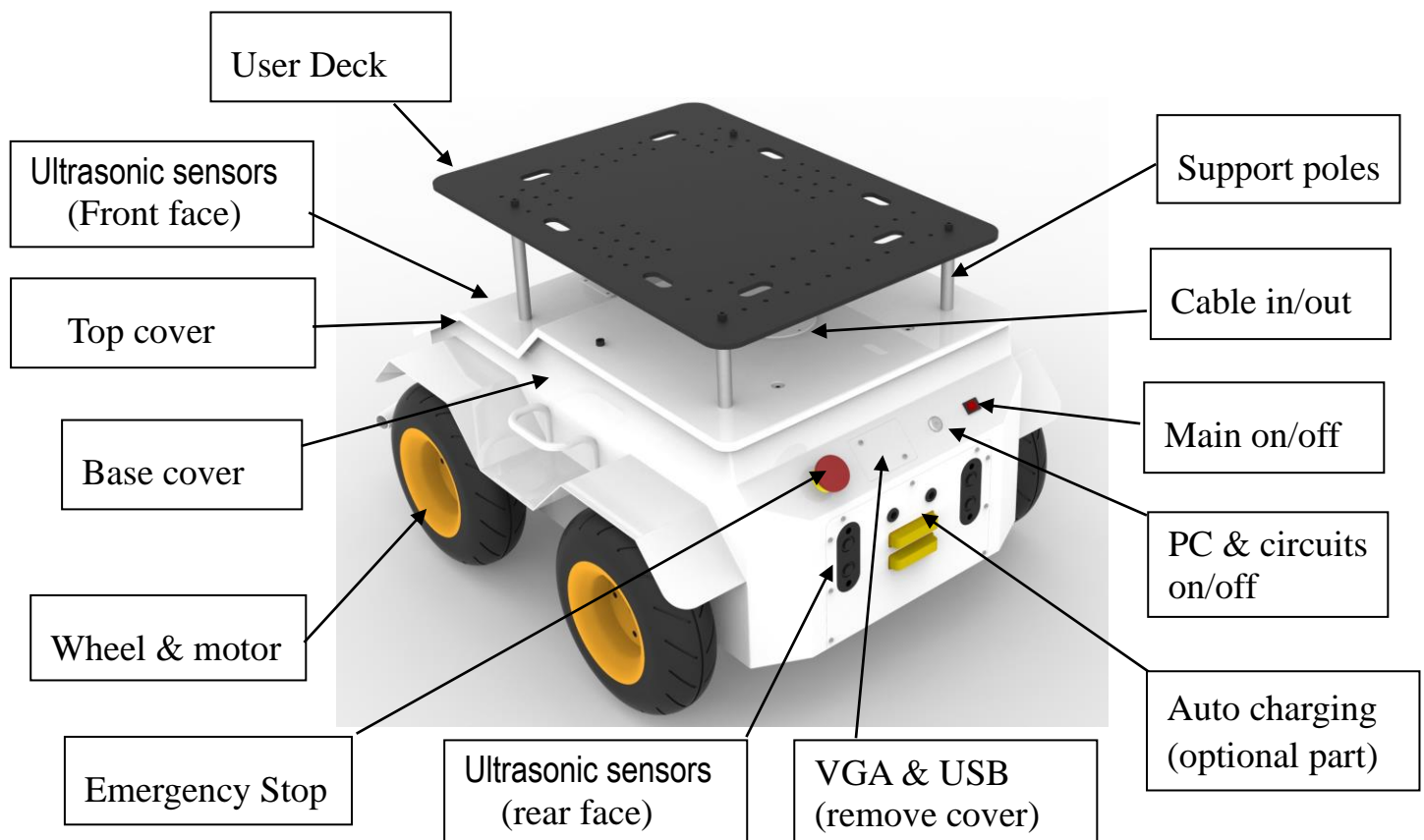
ROSYZ-02 robot platform is a four wheels differential heavy load robot motion chassis platform based on ROS architecture. It is designed for college outdoor applications.

The platform can load up to 80KG, and the maximum walking speed can reach 1.0 meters per second. ROSYZ-02's built-in DCDC power conversion module can provide four different voltages of 5V, 12V and 24V, which can basically solve the power supply problems of various sensors that the robot needs to carry. In order to facilitate developers to do in-depth research, the ROS platform motion control board and DCDC power board circuit diagram and detailed interface drawings we also provide.

This YZ-02 ROS platform has built-in Intel i5 CPU industrial personal computer, installed Ubuntu 16.04 O/S and ROS Kinetic packages. YZ-02 also has ultrasonic wave sensors which can provide basic obstacle avoidance. As optional parts, we provide auto charging dock and LIDAR for real applications.

At the same time, like other ROS platform robots, ROS YZ-02 provides open source basic ROS application examples and basic motion driven node programs that allow developers to use it very easily.

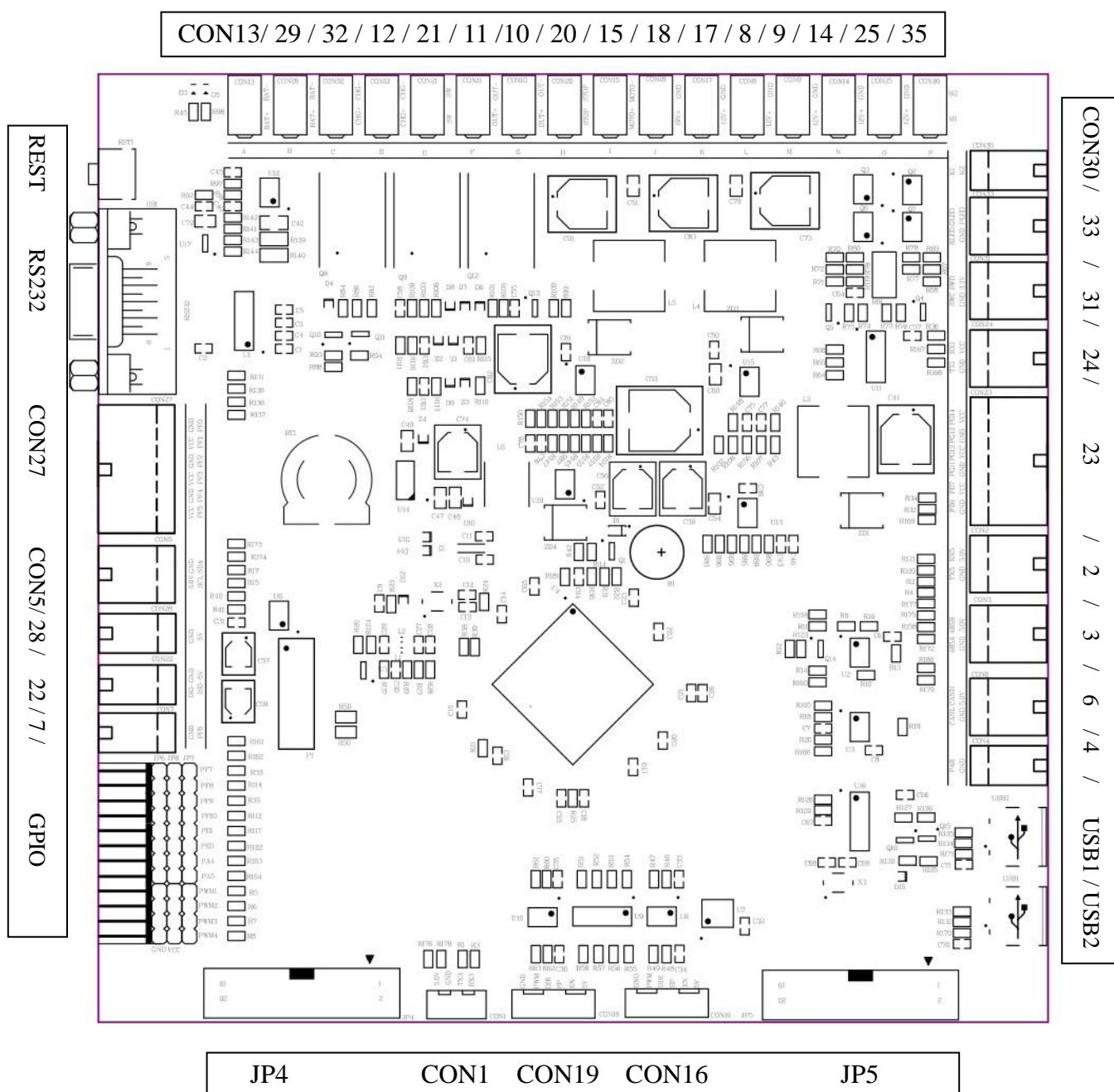
§1.2 Main parts of ROSYZ-02



<PC & circuits on/off> body LED status indictors:

Blue solid: Battery full charged (at charging)
 Blue slow flashing(0.5HZ): Normal working(not charging)
 Blue fast flashing(2HZ): Low battery(not charging)
 Blue breathing : Charging

§1.3 ROSYZ-02 Control Board connectors definition



Connectors Definition:

Power ports

CON13	Battery input(24V)
CON29	Battery input (24V)
CON32	Charger input
CON12	Charger input
CON21	Power switch
CON10	Battery output (24V)
CON11	battery output (24V)
CON20	Emergency stop switch
CON15	Motor power output
CON18	+19V output
CON17	+12V output
CON8	+12V output
CON9	+12V output
CON14	+12V output
CON25	+12V output
CON26	+12V output (isolated)
CON28	+5V output
CON22	+5V output (isolated)

Communication and others

RS232	UART port
CON27:	Front ultrasonic sensors
CON5:	NA
CON7:	NA
CON30:	ON/OFF button
CON33:	LEDs display board
CON31:	SWD
CON24 :	NA
CON23:	Back ultrasonic sensors
CON2:	TTL RS232
CON3 :	NA
CON6:	NA
CON4:	NA
USB1:	USB debug
USB2:	NA
JP4:	NA
JP5:	NA
CON1:	NA
CON19:	Left wheel driver
CON16:	Right wheel driver
JP6, JP7, JP8:	GPIO

JP6: GND

JP8: VCC

JP7:

PF7: power on signal

PF8:NA

PF9,PF10: Auto charging guide signal

PE0,PE1: NA

PA4,PA5:NA

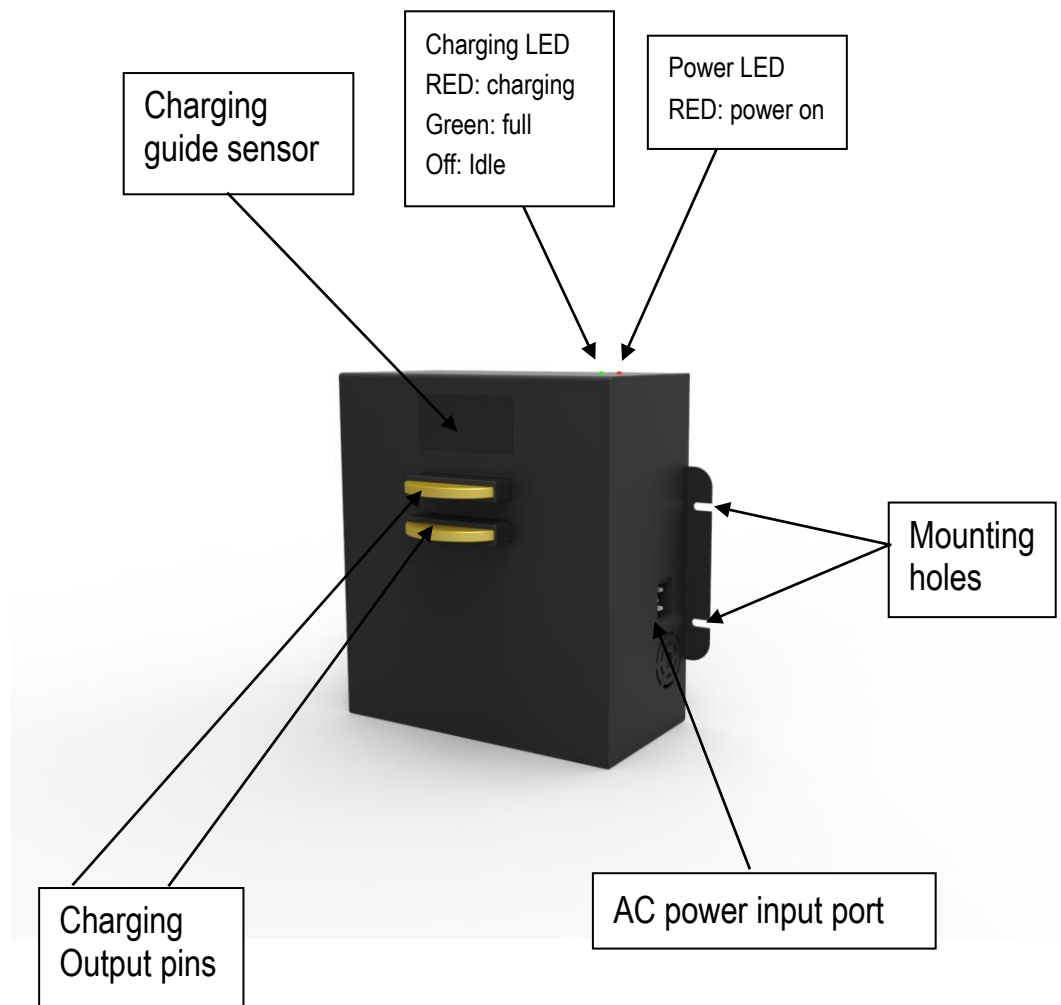
PWM1,PWM2: NA

PWM3:NA

PWM4:NA

§1.4 Auto Charging dock (Optional part)

ROSYZ-02 Auto charging dock profile



Installation guide: Please put this charging dock against a flat wall, make sure the front of charging dock is empty enough (> 3.0 meters @ 120°), without any obstacle in this area.

§1.5 Software installed on the robot computer

ROSYZ-02 has Intel i5 CPU personal computer and pre-installed following items:

1. LINUX : UBUNTU 16.04, **user name: robot; password: 1**
2. ROS basic : KINETIC FULL-DESK
3. ROS application packages(under “ws” folder) :
 - 3.1 STM32_CONNECT (platform drivers and communication)
 - 3.2 ROBOT_MSGS (YZ02 special message)
 - 3.3 YDLIDAR (Optional LIDAR package from EAI)
 - 3.4. MIIBOO_IMU (Optional IMU package from Miiboo)
 - 3.5 RBX1_NAV (SLAM and navigation demo)

§1.6 ROSYZ-02 sensors installed:

ROSYZ-02 pre-installed sensors list (include optional parts)

1. Ultrasonic wave module: DYP-ME007YY V2.0
2. Mechanical safety strip.
3. LIDAR (Optional part, required by user)
4. Auto charging dock (Optional part, required by user)
5. IMU module (Optional part, required by user)
6. Wireless keyboard and mouse (Optional part, required by user)
7. Wireless joystick (Optional part, required by user)

§1.7 ROSYZ-02 default packing list(not include optional parts)

Item	qty	unit	Remark
Main platform	1	set	No battery and decks
User decks	1	pc	
Support pole	4	pc	
Battery Charger	1	pc	P/N: L200CM-24F8
AC power cord	1	pc	Europe type plug
M4 Screw	5	pc	
Screwdriver	1	pc	For M4 screw above
DCDC cable	1	set	8 wires, 10~20cm
ROS SDK and Demo	1	set	Free download
User Manual	1	pc	PDF format
LiFePO4 BATTERY	1	pc	25.2V14Ah

§2 How to use

§2.1 First of first

The ROSYZ-02 robot platform is designed for ROS robot developers. Operators must have the basic knowledge of ROS robots. Please read through this manual before using it, especially read the “Cautions” on the next page carefully.

§2.2 Using step by step

2.2.1 Check the completeness of the accessories:

Open the packing cartons of the robot, take out the ROSYZ-02 robot movement chassis and all parts, and check the packing list, check whether there are leakage and wrong loading.

2.2.2 Install battery-pack and decks

For transportation safety, the battery pack and the body of the ROSYZ-02 robot are individually packaged. Before using, please install the battery firstly.

Remove the top cover and put the battery-pack onto the chassis. Then plug the battery-pack plug into “BAT” connector in the cabin.

2.2.3 Switch power on

Main switch is on the back side of top cover, see Page 2. After turn on it, then push the ON/OFF button, the operating status LED will be Blue slow flashing(2HZ)

If the power LED is not lighting after main switch closed, please check the battery voltage is good or not. For some reason, when you found the battery voltage is $\leq 20V$, please plug in the portable battery charger and charge the battery at least 3 hours, then try again. The portable charging port and auto charging dock terminals are all on the rear cover of robot

2.2.4 Connect the robot into your field Wi-Fi.

See page-2, use screw driver to remove the cover of VGA&USB connector port, insert keyboard/mouse and VGA monitor, then power on robot and chose your Wi-Fi SSID and set password, let robot computer connect into your net system. We suggest you should give this robot a fixed IP address (do it from your LAN router)

Caution: when you want to run robot, please firstly unplug any cables which is connected with your desktop device (such as VGA monitor).

2.2.5 Install a joystick driver when necessary. (Optional part).

If you want to drive your robot from joystick (for example, when you run gmapping function), please insert the joystick receiver USB part into the robot computer USB port.

Please note, if your joystick is not the ROS official website recommended model, please install the joystick ROS driver package firstly.

§2.3 YZ-02 ROS demo instruction:

2.3.1 Do gmapping SLAM to generate a map of your office:

1) Preparing work:

Your desktop PC is ready: Please make sure your desktop or laptop PC was installed ROS KINETIC. Otherwise, please install ROS on your PC firstly. Then copy the rbx1_nav source folder from robot PC to your desktop PC ROS work space src folder, and then on your desktop PC run catkin_make command to compile this rbx1_nav package;

YDLIDAR is connected;

Joystick is connected;

2) Run gmapping command on your robot PC.

Use SSH command to open a SSH terminal from your desktop PC to remote login robot PC, and then launch the gmapping command:

```
$ roslaunch rbx1_nav gmapping_demo_yz.launch
```

You can look over the gmapping_demo_yz.launch file and learn more details about this launch command.

3) Run RVIZ on your desktop PC terminal

Due the running robot has not a monitor and SSH can not provide graphical remote login, so we need run RVIZ on your desktop PC instead. Before run RVIZ, please let your desktop PC's ROS MASTER redirect to robot PC ROS core:

```
$ export ROS_MASTER_URI=http://robot IP:11311
```

```
$ rosrn rviz rviz -d `rospack find rbx1_nav`/gmapping.rviz
```

If everything OK, you can see some map pixel on the screen.

4) Drive robot moving and save map data on your robot PC

Use joystick to drive robot slowly walking around your office, until all roads are covered. Then stop robot.

Use SSH to remote login robot PC from your desktop PC, then

```
$ rospackcd rbx1_nav/maps
```

```
$ rosrn map_server map_saver -f my_map
```

"my_map" the name of this new map. You can find this new map files in rbx1_nav/maps folder.

2.3.2 Use standard move_base and amcl to run robot from where to where

1) I suppose you have just finished gmapping demo, and get a new map named "my_map".

2) **Run move_base and amcl command on your robot PC.**

Use SSH command to open a SSH terminal from your desktop PC to remote login robot PC, and then launch the following command:

```
$ roslaunch rbx1_nav yz_demo_move.launch map:=my_map.yaml
```

You can look over the yz_demo_move.launch file and learn more details about this launch command.

3) **Run RVIZ on your desktop PC terminal**

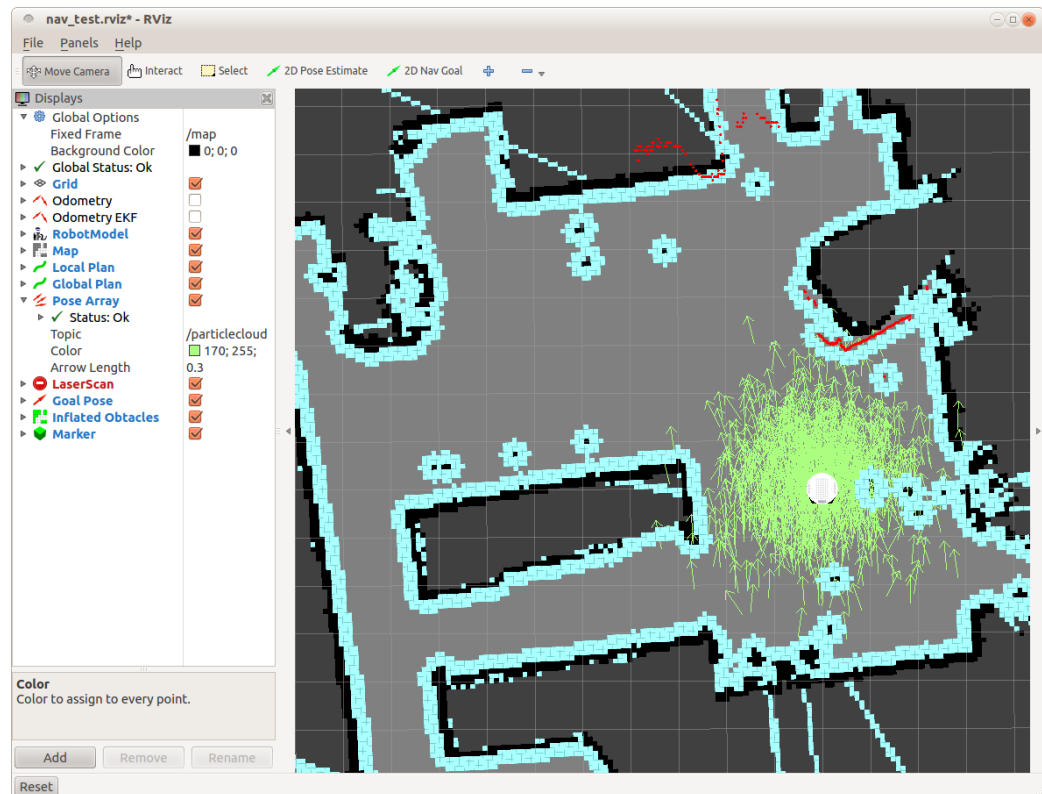
Before run RVIZ, please let your desktop PC's ROS MASTER redirect to robot PC ROS core:

```
$ export ROS_MASTER_URI=http://robot IP:11311
```

```
$ rosrn rviz rviz -d `rospack find rbx1_nav`/nav_test.rviz
```

Use <2D Pose Estimate> to initial robot pose at beginning.

Use <2D Nav Goal> to assign a new target for robot to reach.



§3 CAUTION

3.1 Pre charging: before running the robot for the first time, please charge battery at least 3 hours. For the transportation reasons, the battery before shipping only has very little electricity.

3.2 Charge temperature: pay special attention to charging at room temperature of 0~30 degrees Celsius. High or too low ambient temperature can damage the battery!

3.3 External power interface: please connect the external battery and output DCDC power strictly according to the pin position and polarity of the power supply interface. The wrong wiring will damage the interface board or other devices. Make sure that the maximum current used by the device does not exceed the limited current value of the DCDC board.

3.4 Troubleshooting: if the robot is in the use of abnormal function, please try to turn off the power, and then restart, in general, the robot will be restored to normal, if restarted, the robot can not be used normally, please notify the technical service personnel for remote guidance.

3.4 Emergency issues:

When the robot walks abnormal, please press the red emergency switch!

When the robot or charger has smoke, please turn power immediately!

When the robot is in serious collision or fall accident, turn off the power!

§4 ROSYZ-02 Datasheet

Move base size	860X600X380 mm
Move base weight	45KG(not including battery)
Move drive mode	two-wheel differential
Battery required	24V20AH VRLA battery-pack
Motor type	24V300W DC motors
ROS communication	RS232
DCDC provided	5V2A、12V2A、19V4A
Move speed	0.1-1.4 m/s (5km/hour)
Maximum load weight	80KG
Externa charger	29.4V6A smart charger
Auto charging guide	Optional
Control Board	STM32F10X ARM chip
Emergency stop	Push RED button
Upper layer height	30CM
Hardware data provided	All electronic schematic drawings are provided
ROS diver provided	Provide ROS node binary file which can output each wheel's ticks and accept setting speed value
ROS demo	Provide a basic keyboard remote control moving demo application. C++ source code of this demo is used.

Appendix A: ROS Topics of YZ robot diver node

1. Firstly, launch robot diver node:

```
roslaunch stm32_connect start_basecontrol.launch
```

2. Topics of robot diver node

- 2.1. **Topic name: "Auto_Charging"**, function: to enable or disable auto-charging flag

Topic value: auto_charging_flag = 0: Disable = 1: Enable

【Example】: in a terminal , input following text:

```
rostopic pub /Auto_Charging robot_msgs/Charging_Control "auto_charging_flag:1"
```

Above instruction will let robot go to charging dock automatic

- 2.2. **Topic name: "PMS_get_status"**, check battery power level and status

Topic value: pms_charging_flag = 0: now not charging = 1: now charging

Pms_battary_level = n : means battery power level is n% remain

【Example】: in a terminal , input following text:

```
rostopic echo /PMS_get_status
```

Above instruction will display battery status and remain capacity level

- 2.3. **Topic name: "Ultrasound_result"**, check ultrasonic obstacle sensor trigger status

Topic value: cs_obs = 0: none obstacle =1: front obstacle

=16: rear obstacle =17: front and rear obstacles

【Example】: in a terminal , input following text:

```
rostopic echo /Ultrasound_result
```

Above instruction will display ultrasonic obstacle sensor status

- 2.4. **Topic name: "Wheel_Switch"**, check emergency switch status

Topic value: Switch = 0: Pushed(Enable) =1: Release(Disable)

【Example】: in a terminal , input following text:

```
rostopic echo /Wheel_Switch
```

Above instruction will display emergency switch status

- 2.5. **Topic name: "cmd_vel"**, check or set the linear/angular velocity

【Example】: in a terminal , input following text:

```
rostopic pub /cmd_vel geometry_msgs/Twist "linear:
```

```
  x:0.3 y:0.0 z:0.0 angular: x:0.0 y:0.0 z:0.1"
```

This will let robot run at 0.3m/s linear and 0.1r/s angular velocity

- 2.6. **Topic name: "odom"**, check odometry data

【Example】: in a terminal , input following text:

```
rostopic echo /odom
```

Above instruction will display odometer data