SCOUT 2.0 USER MANUAL

SCOUT 2.0

AgileX Robotics Team

USER MANUAL V.2.0.2 2023.09

Document version

No.	Version	Date	Edited by	Reviewer	Notes
1	V.2.0.0	2023/03/12	吴忠义		Document creation, image update
2	V.2.0.1	2023/04/19	吴忠义		Sync GitHub commands

3	V.2.0.2	2023/07/19	吴忠义		Update firmware upgrade instructions Upper PC software link
4	V2.0.3	2023/09/02	谢瑞亲		Update the instructions for using the ros package Delete RS232 serial port support instructions Update contents
5	V2.0.3	2023/09/08	吴忠义		Synchronized robot parameter list

This chapter contains important safety information, before the robot is powered on for the first time, any individual or organization must read and understand this information before using the device. If you have any questions about use, please contact us at support@agilex.ai. Please follow and implement all assembly instructions and guidelines in the chapters of this manual, which is very important. Particular attention should be paid to the text related to the warning signs.

Safety Information

The information in this manual does not include the design, installation and operation of a complete robot application, nor does it include all peripheral equipment that may affect the safety of the complete system. The design and use of the complete system need to comply with the safety requirements established in the standards and regulations of the country where the robot is installed.

SCOUT integrators and end customers have the responsibility to ensure compliance with the applicable laws and regulations of relevant countries, and to ensure that there are no major dangers in the complete robot application. This includes but is not limited to the following:

Effectiveness and responsibility

- Make a risk assessment of the complete robot system.
- Connect the additional safety equipment of other machinery defined by the risk assessment together.
- Confirm that the design and installation of the entire robot system's peripheral equipment, including software and hardware systems, are correct.
- This robot does not have a complete autonomous mobile robot, including but not limited to automatic anti-collision, anti-falling, biological approach warning and other related safety functions. Related functions require integrators and end customers to follow relevant regulations and feasible laws and regulations for safety assessment , To ensure that the developed robot does not have any major hazards and safety hazards in actual applications.
- Collect all the documents in the technical file: including risk assessment and this manual.
- Know the possible safety risks before operating and using the equipment.

Environmental Considerations

- For the first use,please read this manual carefully to understand the basic operating content and operating specification.
- For remote control operation, select a relatively open area to use SCOUT2.0, because SCOUT2.0 is not equipped with any automatic obstacle avoidance sensor.
- Use SCOUT2.0 always under -10°C~45°C ambient temperature.
- If SCOUT 2.0 is not configured with separate custom IP protection, its water and dust protection will be IP22 ONLY.

Pre-work Checklist

- Make sure each device has sufficient power.
- Make sure Bunker does not have any obvious defects.
- Check if the remote controller battery has sufficient power.
- When using, make sure the emergency stop switch has been released.

Operation

- In remote control operation, make sure the area around is relatively spacious.
- Carry out remote control within the range of visibility.

- The maximum load of SCOUT2.0 is 50KG. When in use, ensure that the payload does not exceed 50KG.
- When installing an external extension on SCOUT2.0, confirm the position of the center of mass of the extension and make sure it is at the center of rotation.
- Please charge in tine when the device is low battery alarm. When SCOUT2..0 has a defect, please immediately stop using it to avoid secondary damage.
- When SCOUT2.0 has had a defect, please contact the relevant technical to deal with it, do not handle the defect by yourself. Always use SCOUT2.0 in the environment with the protection level requires for the equipment.
- Do not push SCOUT2.0 directly.
- When charging, make sure the ambient temperature is above 0 °C.
- If the vehicle shakes during its rotation, adjust the suspension.

Maintenance

- Regularly check the pressure of the tire, and keep the tire pressure between 1.8bar~2.0bar.
- If the tire is severely worn or burst, please replace it in time.
- If the battery do not use for a long time, it need to charge the battery periodically in 2 to 3 months.

Attention

This section includes some precautions that should be paid attention to for SCOUT 2.0 use and development.

Battery

- The battery supplied with SCOUT 2.0 is not fully charged in the factory setting, but its specific power capacity can be displayed on the voltmeter at rear end of SCOUT 2.0 chassis or read via CAN bus communication interface. The battery recharging can be stopped when the green LED on the charger turns green. Note that if you keep the charger connected after the green LED gets on, the charger will continue to charge the battery with about 0.1A current for about 30 minutes more to get the battery fully charged.
- Please do not charge the battery after its power has been depleted, and please charge the battery in time when low battery level alarm is on;

- Static storage conditions: The best temperature for battery storage is -10°C to 45°C; in case of storage for no use, the battery must be recharged and discharged once about every 2 months, and then stored in full voltage state. Please do not put the battery in fire or heat up the battery, and please do not store the battery in high-temperature environment;
- Charging: The battery must be charged with a dedicated lithium battery charger; lithium-ion batteries cannot be charged below 0°C (32°F) and modifying or replacing the original batteries are strictly prohibited.
- Protection: For overheating and overcurrent caused by overload, short circuit or coverage by external things, the BMS in the battery has overtemperature, overcurrent, overvoltage and undervoltage protection.

Operational environment

- The operating temperature of SCOUT 2.0 is -10°C to 45°C; please do not use it below -10°C and above 45°C ;
- The requirements for relative humidity in the use environment of SCOUT 2.0 are: maximum 80%, minimum 30%;
- Please do not use it in the environment with corrosive and flammable gases or closed to combustible substances;
- Do not place it near heaters or heating elements such as large coiled resistors, etc.;
- Except for specially customized version (IP protection class customized), SCOUT 2.0 is not water-proof, thus please do not use it in rainy, snowy or water-accumulated environment;
- The elevation of recommended use environment should not exceed 1,000m;
- The temperature difference between day and night of recommend-ed use environment should not exceed 25°C;
- Regularly check the tire pressure, and make sure it is within 1.8 bar to 2.0bar。
- If any tire is seriously worn out or has blown out, please replace it in time.

Electrical/extension cords

- The top extended power supply current does not exceed 10A, and the total power does not exceed 240W;
- The current of the tail extension power supply shall not exceed 10A, and the total power shall not exceed 240W (if both are used at the same time, the maximum power shall not exceed 300W);

- When the system detects that the battery voltage is lower than the safe voltage, the external power extension will be actively cut off. Therefore, if the external extension device involves the storage of important data and does not have power-down protection, it is recommended that the user pay attention.

Additional safety advice

- In case of any doubts during use, please follow related instruction manual or consult related technical personnel;
- Before use, pay attention to field condition, and avoid mis-operation that will cause personnel safety problem;
- In case of emergencies, press down the emergency stop button and power off the equipment;
- Without technical support and permission, please do not personally modify the internal equipment structure.

Other notes

- SCOUT 2.0 has plastic parts in front and rear, please do not directly hit those parts with excessive force to avoid possible damages;
- When handling and setting up, please do not fall off or place the vehicle upside down;
- For non-professionals, please do not disassemble the vehicle without permission.

CONTENTS

CONTENTS

1 Introduction

SCOUT 2.0 is designed as a multi-purpose UGV with different application scenarios considered: modular design; flexible connectivity; powerful motor system capable of high payload. Additional components such as stereo camera, laser radar, GPS, IMU and robotic manipulator can be optionally installed on SCOUT 2.0 for advanced navigation and computer vision applications. SCOUT 2.0 is frequently used for autonomous driving education and research, indoor and outdoor security patrolling, environment sensing, general logistics and transportation, to name a few only.

1.1 Component list

Name	Quantity
SCOUT 2.0 Robot body	X 1
Battery charger (AC 220V)	X 1
Aviation plug (male, 4-pin)	X 1
Remote control transmitter (optional)	X 1
USB to CAN communication module	X1

1.2 Tech specifications

Parameter Types	Items	Values
Mechanical specifications	L × W × H (mm)	930 X 699 X 349
	Wheelbase (mm)	498
	Front/rear wheel base (mm)	583
	Weight of chassis body (kg)	67±1kg
	Battery Type	Lithium battery
	Battery parameters	24V 30Ah
	Power drive motor	DC brushless 4 X 400W
	Steering drive motor	-
	Parking mode	Servo brake/anti-collision tube
	Steering	Four-wheel differential steering
	Suspension form	Front Double Rocker Independent Suspension Rear Double Rocker Independent Suspension
	Steering motor reduction ratio	-
	Steering motor encoder	-
	Drive motor reduction ratio	1：40
	Drive motor sensor	Magnetic braiding 2500
Performance parameters	IP Grade	IP22
	Maximum speed (km/h)	1.5

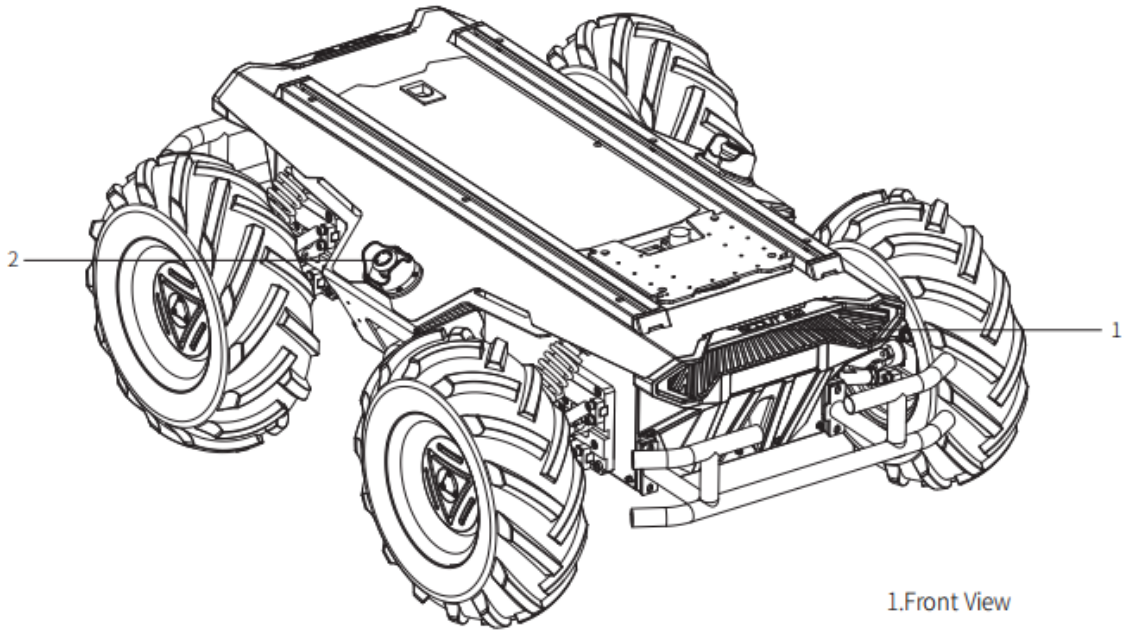
	Minimum turning radius (mm)	Can turn in place
	Maximum gradeability (°)	30°
	Ground clearance (mm)	135
	Maximum battery life (h)	8
	Maximum distance (km)	15KM
	Charging time (h)	3
	Working temperature (°C)	-10~40°C
Control	Control mode	Remote control Control Command control mode
	RC transmitter	2.4G/extreme distance 200M
	System interface	CAN

1.3 Requirement for development

FS RC transmitter is provided (optional) in the factory setting pf SCOUT 2.0, which allows users to control the chassis of robot to move and turn; CAN interfaces on SCOUT 2.0 can be used for user’s customization.

2 The Basics

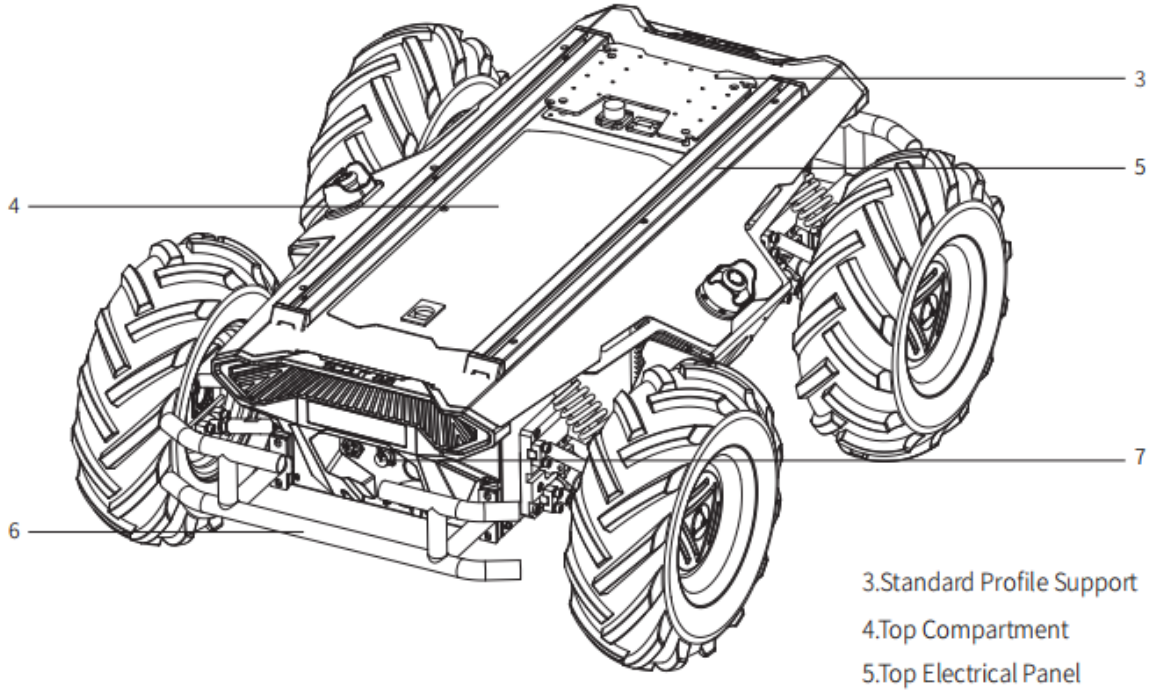
This section provides a brief introduction to the SCOUT 2.0 mobile robot platform, as shown in Figure



1.Front View

2.Stop Switch

Figure 2.1 Front View



3.Standard Profile Support

4.Top Compartment

5.Top Electrical Panel

6.Retardant-collision Tube

7.Rear Panel

Figure 2.2 Rear View

SCOUT2.0 adopts a modular and intelligent design concept. The composite design of inflate rubber tyre and independent suspension on the power module, coupled with the powerful DC brushless servo motor, makes the SCOUT2.0 robot chassis development platform has strong pass ability and ground adapt ability, and can move flexibly on different ground.

1. An-ti-collision beams are mounted around the vehicle to reduce possible damages to the vehicle body during a collision.
2. Lights are both mounted at front and at back of the vehicle, of which the white light is designed for illumination in front whereas the red light is designed at rear end for warning and indication.
3. Emergency stop buttons are installed on both sides of the robot to ensure easy access and pressing either one can shut down power of the robot immediately when the robot behaves abnormally.
4. Open electrical interfaces and communication interfaces are configured at the rear and top of the car to facilitate customers' secondary development. The electrical interface uses aviation waterproof connectors in the design and selection. On the one hand, it is conducive to customer expansion and use. On the other hand, This enables the robot platform to be used in some harsh environments.
5. A bayonet open compartment is reserved on the top for users.

2.1 Status indication

Users can identify the status of vehicle body through the voltmeter, the beeper and lights mounted on SCOUT 2.0. For details, please refer to Table 2.1.

Status	Description
Voltage	The current battery voltage can be read from the voltmeter on the rear electrical interface and with an accuracy of 1V.
Replace battery	When the battery power is lower than 15% or the voltage is lower than 24V, the car body will make a harsh "di-di-di" sound to prompt you. When it detects that the battery power is lower than 10% or the voltage is lower than 23V, SCOUT will actively cut off the external expansion power supply and driver power supply to prevent battery damage. At this time, the chassis will not be able to perform motion control and accept external command control.

Robot powered on	Front and rear lights are switched on.

Table 2.1 Descriptions of Vehicle Status

2.2 Instructions on electrical interfaces

2.2.1 Top electrical interface

SCOUT 2.0 provides three 4-pin aviation connectors and one DB9 (RS232) connector. The position of the top aviation connector is shown in Figure 2.3.

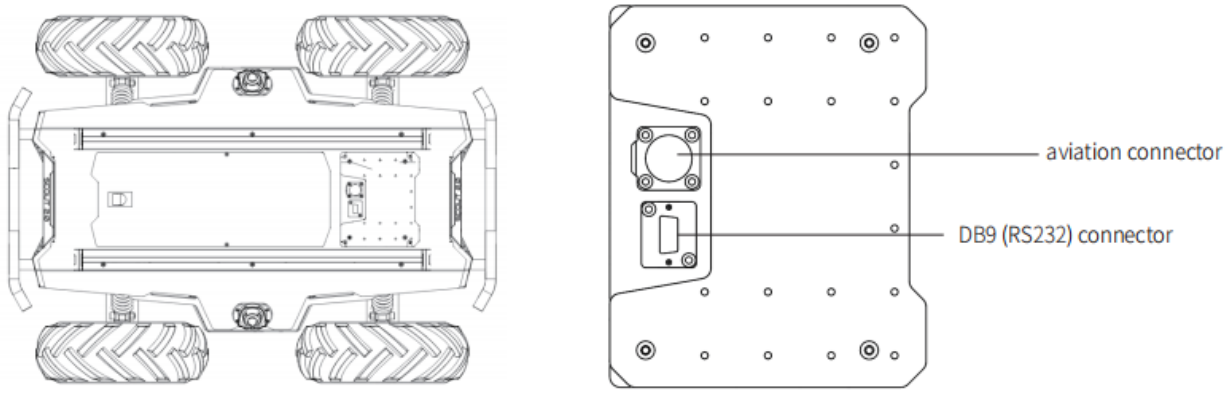
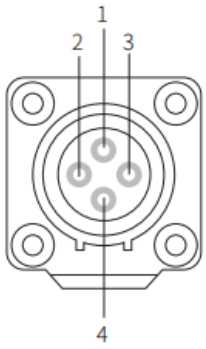


Figure 2.3 Schematic Diagram of SCOUT 2.0 Electrical Interface on Top

SCOUT 2.0 has an aviation extension interface both on top and at rear end, each of which is configured with a set of power supply and a set of CAN communication interface. These interfaces can be used to supply power to extended devices and establish communication. The specific definitions of pins are shown in Figure 2.4.

It should be noted that, the extended power supply here is internally controlled, which means the power supply will be actively cut off once the battery voltage drops below the pre-specified threshold voltage. Therefore, users need to notice that SCOUT 2.0 platform will send a low voltage alarm before the threshold voltage is reached and also pay attention to battery recharging during use.

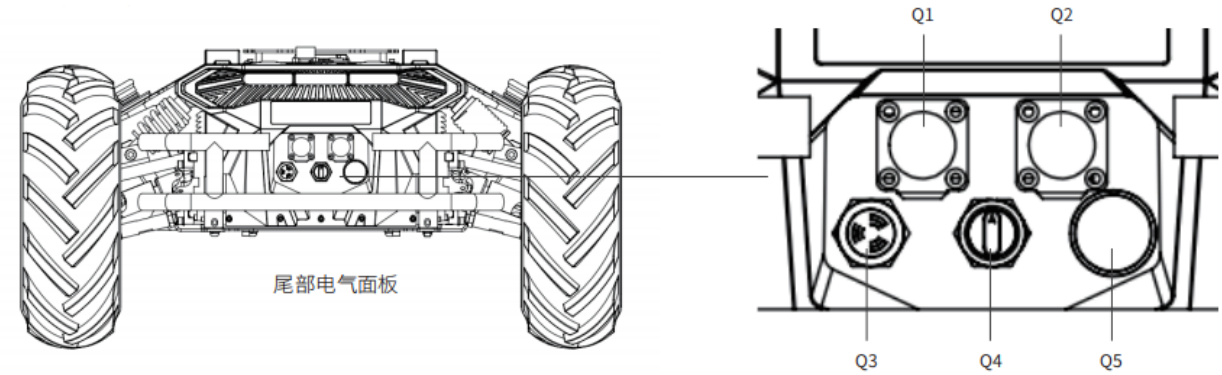


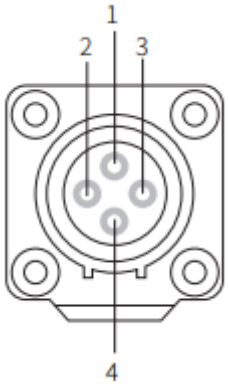
Pin No.	Pin Type	Function and Definition	Remarks
1	Power	VCC	Power positive, voltage range 23 - 29.2V, MAX.current 10A
2	Power	GND	Power negative
3	CAN	CAN_H	CAN bus high
4	CAN	CAN_L	CAN bus low

Figure 2.4 Definitions for Pins of Top Aviation Extension Interface

2.2.2 Rear electrical interface

The extension interface at rear end is shown in Figure 2.6, where Q1 is the key switch as the main electrical switch; Q2 is the recharging interface; Q3 is the power supply switch of drive system; Q4 is DB9 serial port; Q5 is the extension interface for CAN and 24V power supply; Q6 is the display of battery voltage.





Pin No.	Pin Type	Function and Definition	Remarks
1	Power	VCC	Power positive, voltage range 23 - 29.2V, maximum current 5A
2	Power	GND	Power negative
3	CAN	CAN_H	CAN bus high
4	CAN	CAN_L	CAN bus low

Figure 2.7 Description of Front and Rear Aviation Interface Pins

2.3 Instructions on remote control

The FS remote control is an optional accessory for SCOUT2.0 products. Customers can choose according to actual needs. The remote control can easily control the SCOUT2.0 universal robot chassis. In this product, we use a left-hand throttle design. Its definition and functions can be referred to Figure 2.7.

The functions of the buttons are defined as follows: SWA and SWD are temporarily not enabled. SWB is the control mode selection button. Push it to the top for the command control mode, and push it to the middle for the remote control mode. SWC is the light control button. S1 is the throttle button. Control SCOUT2.0 to move forward and backward; S2 controls rotation, and POWER is the power button. Press and hold at the same time to turn on.

Note: The mapping of the remote control has been set before leaving the factory, please do not change it at will.

Figure 2.7 Schematic Diagram of Buttons on FS RC transmitter

Remote control interface description:

Scout : model

Vol: battery voltage

Car: chassis status

Batt: Chassis power percentage

P: Park

Remoter: remote control battery level

Fault Code: Error information (Refer to the fault information description table)

2.4 Instructions on control demands and movements

A reference coordinate system can be defined and fixed on the vehicle body as shown in Figure 2.9 in accordance with ISO 8855.

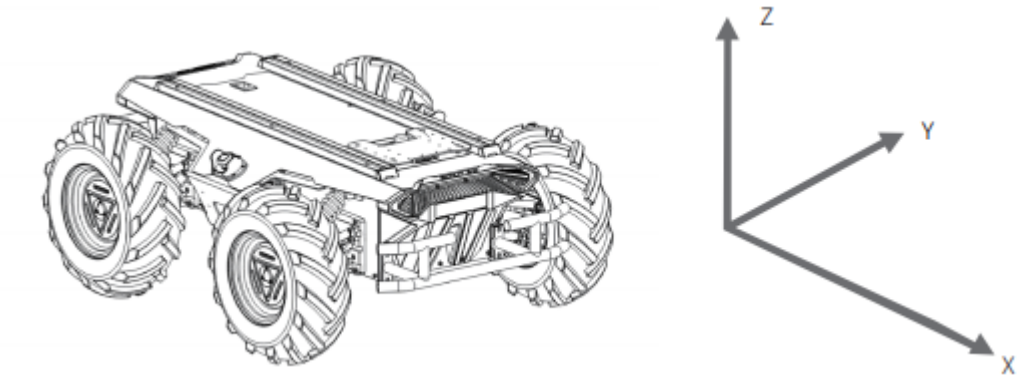


Figure 2.9 Schematic Diagram of Reference Coordinate System for Vehicle Body

As shown in Figure 2.9, the vehicle body of SCOUT 2.0 is in parallel with X axis of the established reference coordinate system. In the remote control mode, push the remote control stick S1 forward to move in the positive X direction, push S1 backward to move in the negative X direction. When S1 is pushed to the maximum value, the movement speed in the positive X direction is the maximum, When pushed S1 to the minimum, the movement speed in the negative direction of the X direction is the maximum; the remote control stick S2 controls the steering of the front wheels of the car body, push S2 to the left, and the vehicle turns to the left, pushing it to the maximum, and the steering angle is the largest, S2 Push to the right, the car will turn to the right, and push it to the maximum, at this time the right steering angle is the largest. In the control command mode, the positive value of the linear velocity means movement in the positive direction of the X axis, and the negative value of the linear velocity means movement in the negative direction of the X axis; The positive value of the angular velocity means the car body moves from the positive direction of the X axis to the positive direction of the Y axis, and the negative value of the angular velocity means the car body moves from the positive direction of the X axis to the negative direction of the Y axis.

2.5 Instructions on lighting control

SCOUT2.0 is equipped with lights on the front and rear. For the convenience of customers, SCOUT2.0 opens the lighting control interface to the outside world. At the same time, in order to save energy, a lighting control interface is reserved on the remote control. The remote control version currently only supports FS remote controls, and the adaptation work for other remote controls is still in progress. There are currently 3 lighting modes on the remote control. The mode switching can be switched through the SWC lever: Mode control description: Move the SWC lever to the bottom for normally closed mode, the middle for normally open mode, and the top for breathing light mode.

Normally closed mode: In the normally closed mode, if the chassis is stationary, the headlights will turn off, and the taillights will enter the breathing light mode to indicate the current working status; if the chassis is driving at normal speed, the taillights will turn off and the headlights will turn on;

Normally on mode: In the normally on mode, if the chassis is stationary, the headlights are always on, and the taillights will enter the breathing light mode to indicate the stationary state; if in the sports mode, the taillights are off and the headlights are on;

Breathing light mode: The headlights and tail lights are in breathing light mode in various states.

3 Getting Started

This section introduces the basic operation and development of the SCOUT 2.0 platform using the CAN bus interface.

3.1 Use and operation

The basic operating procedure of startup is shown as follows:

Check

Check the condition of SCOUT 2.0. Check whether there are significant anomalies; if so, please contact the after-sale service personal for support;

Check the state of emergency-stop switches. Make sure both emergency stop buttons are released;

Startup

- Rotate the key switch (Q1 on the electrical panel), and normally, the voltmeter will display correct battery voltage and front and rear lights will be both switched on;

- Check the battery voltage. If there is no continuous "beep-beep-beep..." sound from beeper, it means the battery voltage is correct; if the battery power level is low, please charge the battery;
- Press Q3 (drive power switch button).

Shutdown

- Rotate the key switch to cut off the power supply;

Emergency stop

- Press down emergency push button both on the left and the right of SCOUT 2.0 vehicle body;

Basic operating procedure of remote control:

After the chassis of SCOUT 2.0 mobile robot is started correctly, turn on the RC transmitter and select the remote-control mode. Then, SCOUT 2.0 platform movement can be controlled by the RC transmitter.

3.2 Charging

SCOUT2.0 products are equipped with a 10A charger by default in the car, which can meet the charging needs of customers. By default, it is turned off for charging. When charging normally, there is no indicator light on the chassis. Please refer to the instructions on the charger for specific indicator lights.

3.2.1 Charging operation

1. Make sure that the SCOUT2.0 chassis is shut down and powered off. Before charging, please confirm that Q1 (knob switch) in the rear electrical console is turned off.

2. Insert the plug of the charger into the Q2 charging interface in the electrical control panel at the rear of the car;

3. Connect the charger to the power supply and turn on the charger switch to enter the charging state.

Note: It currently takes about 3 hours for the battery to fully charge from 22V, and the battery full charge voltage is about 29.2V (the battery voltage here is a ternary lithium battery type, if the battery type is lithium iron phosphate, the maximum voltage is 26.8V); charging Time calculation 30aH÷10A=3H

3.2.2 Battery replacement

SCOUT2.0 adopts a detachable battery solution for the convenience of users. In some special cases, the battery can be replaced directly. The operation steps and diagrams are as follows (before operation, ensure that SCOUT2.0 is power-off):

- Open the upper panel of SCOUT2.0, and unplug the two XT60 power connectors on the main control board (the two connectors are equivalent) and the battery CAN connector;
- Hang SCOUT2.0 in midair, unscrew eight screws from the bottom with a national hex wrench, and then drag the battery out;
- Replace the battery and fixed the bottom screws.
- Plug the XT60 interface and the power CAN interface into the main control board, confirm that all the connecting lines are correct, and then power on to test.

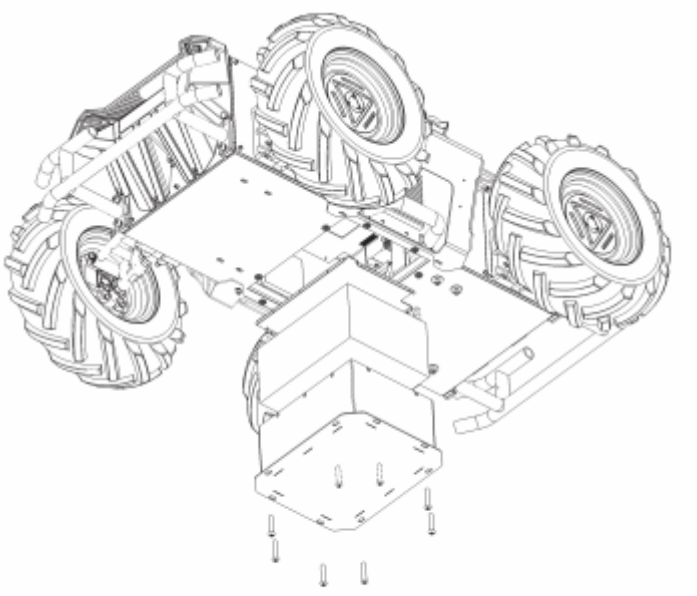


Figure 3.1 Schematic Diagram of replace battery

3.3 Communication using CAN

SCOUT 2.0 provides CAN interfaces for user customization. Users can use it to conduct command control over the vehicle body.

3.3.1 CAN cable connection

SCOUT2.0 deliver with two aviation male plugs as shown in Figure 3.2. For wire definitions, please refer to Table 2.2.

3.3.2 Implementation of CAN command control

Correctly start the chassis of SCOUT 2.0 mobile robot, and turn on DJI RC transmitter. Then, switch to the command control mode, i.e. toggling S1 mode of DJI RC transmitter to the top. At this point, SCOUT 2.0 chassis will accept the command from CAN interface, and the host can also parse the current state of chassis with the real-time data fed back from CAN bus. For the detailed content of protocol, please refer to CAN communication protocol.

Figure 3.2 Schematic diagram of aviation plug male connector

3.3.3 CAN message protocol

Correctly start the chassis of SCOUT 2.0 mobile robot, and turn on DJI RC transmitter. Then, switch to the command control mode, i.e. toggling S1 mode of DJI RC transmitter to the top. At this point, SCOUT 2.0 chassis will accept the command from CAN interface, and the host can also parse the current state of chassis with the real-time data fed back from CAN bus. For the detailed content of protocol, please refer to CAN communication protocol.

Table 3.1 Feedback Frame of SCOUT 2.0 Chassis System Status

Command Name	System Status Feedback Command			
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Steer-by-wire chassis	Decision-making control unit	0x151	20ms	None
Data length	0x08			
Position	Function	Data type	Description	
byte [0]	Current status of vehicle body	unsigned int8	0x00 System in normal condition 0x01 Emergency stop mode (not enabled) 0x02 System exception	
byte [1]	Mode control	unsigned int8	0×00 Standby mode 0×01 CAN command control mode 0×02 Serial port control mode 0×03 Remote control mode	
byte [2] byte [3]	Battery voltage higher 8 bits Battery voltage lower 8 bits	unsigned int16	Actual voltage × 10 (with an accuracy of 0.1V)	
byte [4]	Reserved	-	0×00	
byte [5]	Failure information	unsigned int8	Refer to Table 3.2 [Description of Failure Information]	
byte [6]	Reserved	-	0×00	
byte [7]	Count paritybit (count)	unsigned int8	0-255 counting loops, which will be added once everycommand sent	

Table 3.2 Description of Failure Information

Description of Failure Information		
Byte	bit	Meaning
byte [4]	bit [0]	Battery undervoltage fault (0: No failure 1: Failure) Protection voltage is 22V (The battery version with BMS, the protection power is 10%)
	bit [1]	Battery undervoltage fault[2] (0: No failure 1: Failure) Alarm voltage is 24V (The battery version with BMS, the warning power is 15%)
	bit [2]	RC transmitter disconnection protection (0: Normal 1: RC transmitter disconnected)
	bit [3]	No.1 motor communication failure (0: No failure 1: Failure)
	bit [4]	No.2 motor communication failure (0: No failure 1: Failure)
	bit [5]	No.3 motor communication failure (0: No failure 1: Failure)
	bit [6]	No.4 motor communication failure (0: No failure 1: Failure)
	bit [7]	Reserved, default 0

Note[1]: Robot chassis firmware version V1.2.8 is supported by subsequent versions, and the previous version requires firmware upgrade to support

Note[2]: The buzzer will sound when the battery under-voltage, but the chassis control will not be affected, and the power output will be cut off after the under-voltage fault

The command of movement control feedback frame includes the feedback of current linear speed and angular speed of moving vehicle body. For the detailed content of protocol, please refer to Table 3.3.

Table 3.3 Movement Control Feedback Frame

Command Name	Movement Control Feedback Command

Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Steer-by-wire chassis	Decision-making control unit	0x221	20ms	None
Date length	0×08			
Position	Function	Data type	Description	
byte [0] byte [1]	Moving speed higher 8 bits Moving speed lower 8 bits	signed int16	Actual speed × 1000 (with an accuracy of 0.001m/s)	
byte [2] byte [3]	Rotation speed higher 8 bits Rotation speed lower 8 bits	signed int16	Actual speed × 1000 (with an accuracy of 0.001rad/s)	
byte [4]	Reserved	-	0x00	
byte [5]	Reserved	-	0x00	
byte [6]	Reserved	-	0x00	
byte [7]	Reserved	-	0x00	

The control frame includes control openness of linear speed and control openness of angular speed. For its detailed content of protocol, please refer to Table 3.4.

Table 3.4 Control Frame of movement Control Command

Command Name		Control Command		
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)

Decision-making control unit	Chassis node	0x111	20ms	500ms
Date length	0×08			
Position	Function	Data type	Description	
byte [0] byte [1]	Linear speed higher 8 bits Linear speed lower 8 bits	signed int16	Vehicle moving speed, unit mm/s (effective value+ -1500)	
byte [2] byte [3]	Angular speed higher 8 bits Angular speed lower 8 bits	signed int16	Vehicle rotation angular speed, unit mm/s (effective value+ -1500)	
byte [4]	Reserved	—	0x00	
byte [5]	Reserved	—	0x00	
byte [6]	Reserved	—	0x00	
byte [7]	Reserved	—	0x00	

The mode setting frame is used to set the control interface of the terminal. For its detailed content of protocol, please refer to Table 3.5.

Table 3.5 Control mode setting frame

Command Name	Control Mode Setting Command			
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Decision-making control unit	Chassis node	0×421	None	None
Date length	0×01			

Position	Function	Date type	Description
byte [0]	Control mode	unsigned int8	0×00 Standby mode 0×01 CAN command mode enable

Description of control mode: In case the SCOUT 2.0 is powered on and the RC transmitter is not connected, the control mode is defaulted to standby mode. At this time, the chassis only receives control mode command, and does not respond other commands. To use CAN for control need to switch CAN command mode at first. If the RC transmitter is turned on, the RC transmitter has the highest authority, can shield the control of command and switch the control mode.

Status setting frame is use to clear the system errors. For its detailed content of protocol, please refer to Table 3.6.

Table 3.6 Status Setting Frame

Command Name	Status Setting Command			
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Decision-making control unit	Chassis node	0×441	None	None
Date length	0×01			
Position	Function	Date type	Description	
byte [0]	Errors clearing command	unsigned int8	0×00 Clear all failure 0×01 Clear Motor 1 failure 0×02 Clear Motor 2 failure 0×03 Clear Motor 3 failure 0×04 Clear Motor 4 failure	

[Note 3] Example data: The following data is only used for testing

1.The vehicle moves forward at 0.15m/s

byte [0]	byte [1]	byte [2]	byte [3]	byte [4]	byte [5]	byte [6]	byte [7]
0x00	0x96	0x00	0x00	0x00	0x00	0x00	0x00

2.The vehicle steering 0.2rad/s

byte [0]	byte [1]	byte [2]	byte [3]	byte [4]	byte [5]	byte [6]	byte [7]
0x00	0x00	0x00	0xc8	0x00	0x00	0x00	0x00

The chassis status information will be feedback, and what's more, the information about motor current, encoder and temperature are also included. The following feedback frame contains the information about motor current, encoder and motor temperature.

The motor numbers of the 4 motors in the chassis are shown in the figure below:

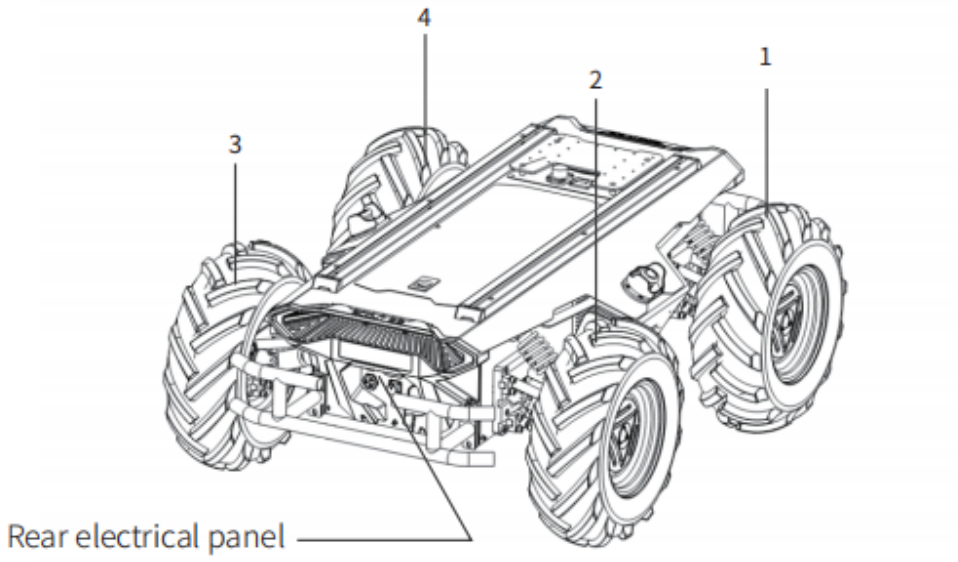


Figure 3.0 Schematic diagram Motor feedback ID

Table 3.7 Motor Speed Current Position Information Feedback

Command Name			Control Command	
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Steer-by-wire chassis	Decision-making control unit	0x251~0x254	20ms	None

Date length	0×08		
Position	Function	Data type	Description
byte [0] byte [1]	Motor speed higher 8 bits Motor speed lower 8 bits	signed int16	Current speed of motor Unit RPM
byte [2] byte [3]	Motor current higher 8 bits Motor current lower 8 bits	signed int16	Motor current Unit 0.1A
byte [4]	Motor position highest bits	signed int32	Current position of the motor Unit: pulse
byte [5]	Motor position second-highest bits		
byte [6]	Motor position second-lowest bits		
byte [7]	Motor position lowest bits		

Table 3.8 Motor temperature, voltage and status information feedback

Command Name	Motor Drive Low Speed Information Feedback Frame			
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)

Steer-by-wire chassis	Decision-making control unit	0x261~0x264	20ms	None
Date length	0×08			
Position	Function	Data type	Description	
byte [0] byte [1]	Drive voltage higher 8 bits Drive voltage lower 8 bits	unsigned int16	Current voltage of drive unit 0.1V	
byte [2] byte [3]	Drive temperature higher 8 bits Drive temperature lower 8 bits	signed int16	Unit 1°C	
byte [4]	Motor temperature	signed int8	Unit 1°C	
byte [5]	Drive status	unsigned int8	See the details in [Table 3.9]	
byte [6]	Reserved	-	0x00	
byte [7]	Reserved	-	0x00	

Table 3.9 Drive Status

Byte	Bit	Description
byte[5]	bit[0]	Whether the power supply voltage is too low (0:Normal 1:Too low)
	bit[1]	Whether the motor is overheated (0:Normal 1:Overheated)
	bit[2]	Whether the drive is over current (0:Normal 1:Over current)

	bit[3]	Whether the drive is overheated (0:Normal 1:Overheated)
	bit[4]	Sensor status (0:Normal 1:Abnormal)
	bit[5]	Drive error status (0:Normal 1:Error)
	bit[6]	Drive enable status (0:Normal 1:Disability)
	bit[7]	Reserved

The front and external lights also support command control. The following table shows the control commands:

Table 3.10 Light Control Frame

Command Name	Light Control Frame			
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Decision-making control unit	Steer-by-wire chassis	0x121	20ms	None
Date length	0×08			
Position	Function	Date type	Description	
byte [0]	Light control enable flag	unsigned int8	00x00 Control command invalid 0x01 Lighting control enable	
byte [1]	Front light mode	unsigned int8	0x00 Normally off 0x01 Normally open 0x02 Breathing light mode 0x03 Customer-defined brightness	
byte [2]	Custom brightness of front light	unsigned int8	[0, 100], where 0 refers to no brightness,100 refers to maximum brightness[5]	

byte [3]	Rear light mode	unsigned int8	0x00 Normally off 0x01 Normally open 0x02 Breathing light mode 0x03 Customer-defined brightness
byte [4]	Customize brightness for rear light	unsigned int8	[0, 100], where 0 refers to no brightness,100 refers to maximum brightness[5]
byte [5]	Reserved	_	0x00
byte [6]	Reserved	_	0x00
byte [7]	Parity bit (checksum)	unsigned int8	0~255 loop count, the count is incremented every time an instruction is sent.

Note [5]: The values are valid for custom mode.

Table 3.11 Light Control Feedback Frame

Command Name	Light Control Feedback Command			
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Steer-by-wire chassis	Decision-making control unit	0x231	20ms	None
Date length	0×08			
Position	Function	Date type	Description	
byte [0]	Current lighting control enable flag	unsigned int8	00x00 Control command invalid 0x01 Lighting control enable	

byte [1]	Current front light mode	unsigned int8	0x00 Normally off 0x01 Normally open 0x02 Breathing light mode 0x03 Customer-defined brightness
byte [2]	Current custom brightness of front light	unsigned int8	[0, 100], where 0 refers to no brightness ,100 refers to maximum brightness
byte [3]	Current rear light mode	unsigned int8	0x00 Normally off 0x01 Normally open 0x02 Breathing light mode 0x03 Customer-defined brightness
byte [4]	Current custom brightness of rear light	unsigned int8	[0, 100], where 0 refers to no brightness ,100 refers to maximum brightness
byte [5]	Reserved	_	0x00
byte [6]	Reserved	_	0x00
byte [7]	Parity bit (checksum)	unsigned int8	0~255 loop count, the count is incremented every time an instruction is sent.

Table 3.12 System Version Information Enquiry Frame

Command Name		System Version Information Enquiry Command		
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Decision-making control unit	Chassis node	0x411	None	None
Date length	0×01			

Position	Function	Date type	Description
byte [0]	Enquire system version	unsigned int8	Constant 0×01

Table 3.13 System Version Information Enquiry Frame

Command Name	System Version Information Feedback Frame			
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Steer-by-wire chassis	Decision-making control unit	0x41A	None	None
Date length	0×08			
Position	Function	Date type	Description	
byte [0]	The number of main control hardware version higher 8 bits	unsigned int16	Higher 8 bits is the main version number, lower 8 bits is the second version number	
byte [1]	The number of main control hardware version lower 8 bits			
byte [2]	The number of drive hardware version higher 8 bits	unsigned int16	Higher 8 bits is the main version number, lower 8 bits is the second version number	
byte [3]	The number of drive hardware version lower 8 bits			

byte [4]	The number of main control software version higher 8 bits	unsigned int16	Higher 8 bits is the main version number, lower 8 bits is the second version number
byte [5]	The number of main control software version lower 8 bits		
byte [6]	The number of drive software version higher 8 bits	unsigned int16	Higher 8 bits is the main version number, lower 8 bits is the second version number
byte [7]	The number of drive software version lower 8 bits		

Table 3.14 Mileometer Information Feedback

Command Name	Mileometer Information Feedback			
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Steer-by-wire chassis	Decision-making control unit	0x311	20ms	None
Date length	0×08			
Position	Function	Data type	Description	

byte [0] byte [1 byte [2] byte [3]	Left wheel mileometer highest bit Left wheel mileometer second-highest bit Left wheel mileometer second-highest bit Left wheel mileometer lowest bit	signed int32	Chassis left wheel mileometer feedback Unit:mm
byte [4] byte [5] byte [6] byte [7]	Right wheel mileometer highest bit Right wheel mileometer second-highest bit Right wheel mileometer second-highest bit Right wheel mileometer lowest bit	signed int32	Chassis right wheel mileometer feedback Unit:mm

Table 3.15 Remote Control Information Feedback

Command Name	Remote Control Information Feedback Frame			
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Steer-by-wire chassis	Decision-making control unit	0x241	20ms	None
Date length	0×08			
Position	Function	Data type	Description	

byte[0]	SW feedback	unsigned int8	bit[0-1]: SWA: 2- Up 3-Down bit[2-3]: SWB : 2-Up 1-Middle 3-Down bit[4-5]: SWC : 2-Up 1-Middle 3-Down bit[6-7]: SWD：2-Up 3-Down
byte[1]	Right joystick left and right	signed int8	Range[-100,100]
byte[2]	Right joystick up and down	signed int8	Range[-100,100]
byte[3]	Left joystick up and down	signed int8	Range[-100,100]
byte[4]	Left joystick left and right	signed int8	Range[-100,100]
byte[5]	Left knob VRA	signed int8	Range[-100,100]
byte[6]	Reserved	--	0x00
byte[7]	Count Parity bit	unsigned int8	0~255 Loops counting

Command Name		BMS Data Feedback		
Sending node	Receiving node	ID	Cycle (ms)	Receive-timeout (ms)
Steer-by-wire chassis	Decision-making control unit	0x361	500ms	None
Date length	0×08			
Position	Function	Data type	Description	

byte[0]	Battery SOC	unsigned int8	Range 0~100
byte[1]	Battery SOH	unsigned int8	Range 0~100
byte[2]	Battery voltage higher 8 bits	unsigned int16	Unit: 0.01V
byte[3]	Battery voltage lower 8 bits	signed int8	Unit: 0.01V
byte[4]	Battery current higher 8 bits	signed int16	Unit: 0.1A
byte[5]	Battery current lower 8 bits	signed int8	Unit: 0.1A
byte[6] byte[7]	Battery temperature higher 8 bits Battery temperature lower 8 bits	signed int16	Unit: 0.1°C

3.5 Firmware upgrades

In order to facilitate users to upgrade the firmware version used by SCOUT 2.0 and bring customers a more complete experience, SCOUT 2.0 provides a firmware upgrade hardware interface and corresponding client software.

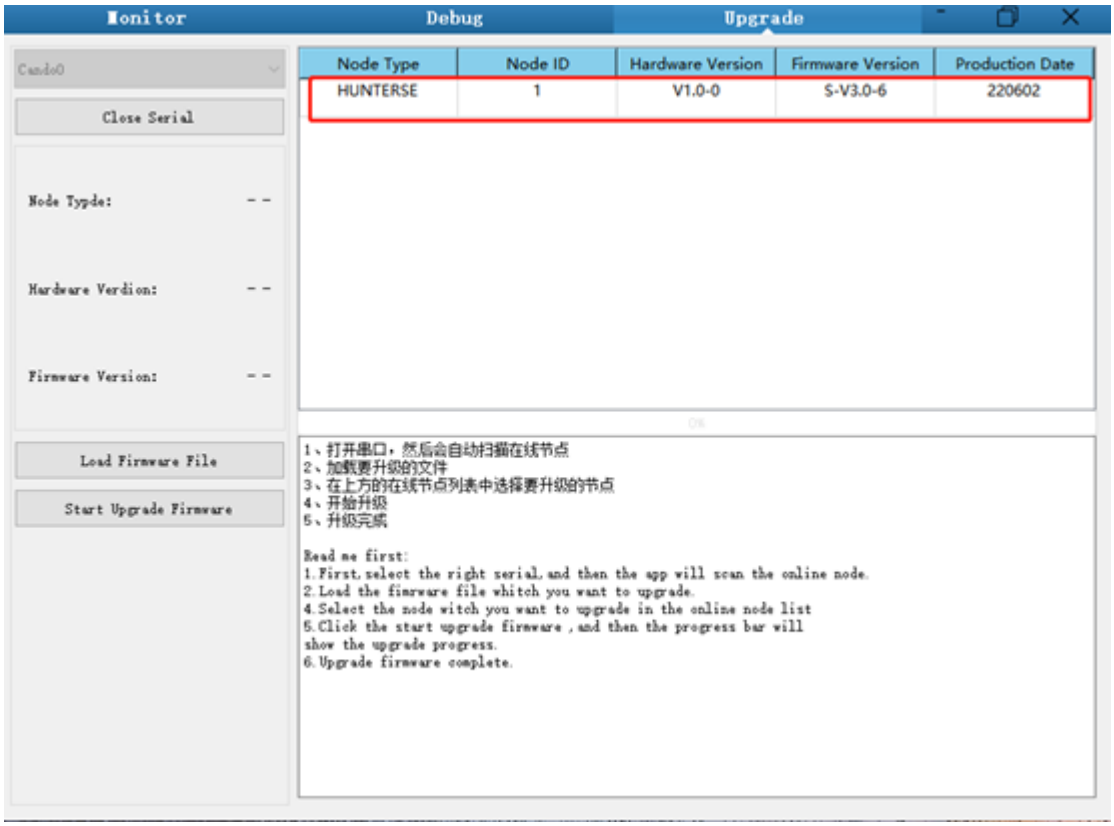
Upgrade Preparation

- Agilex CAN debugging module X 1
- Micro USB cable X 1
- SCOUT 2.0 chassis X 1
- A computer (WINDOWS OS (Operating System)) X 1

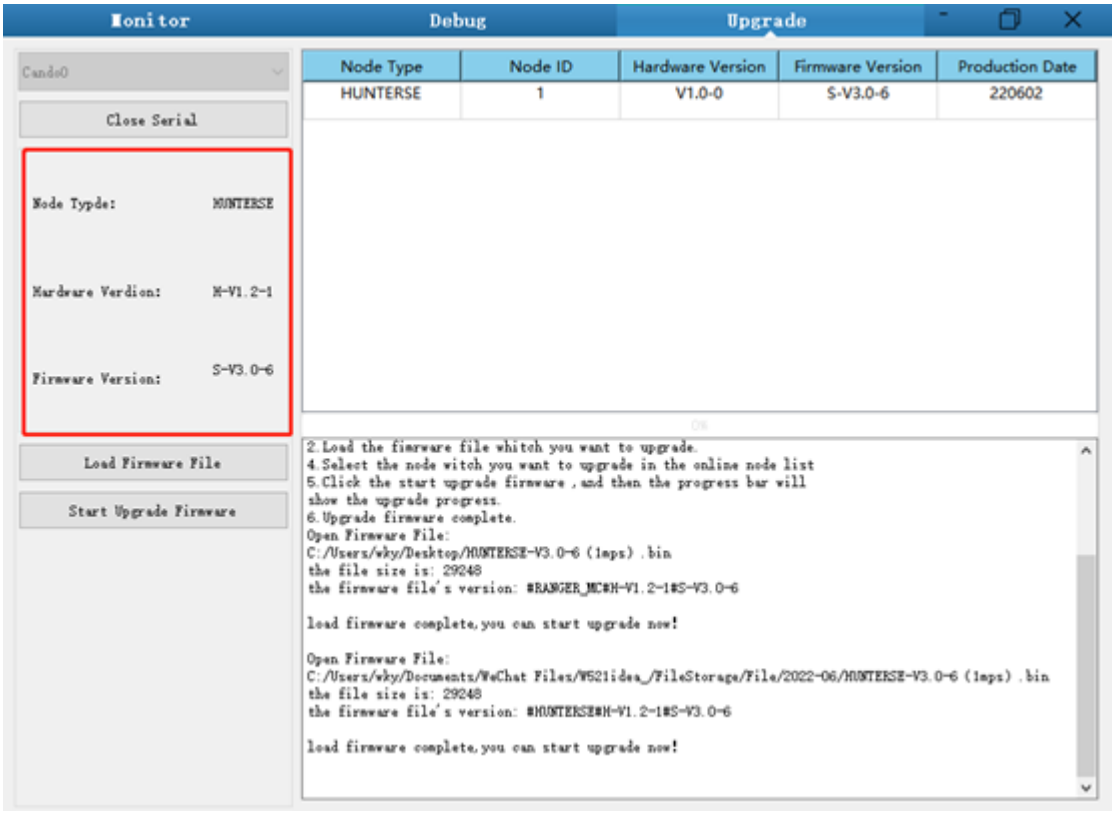
Upgrade Process

1.Plug in the USBTOCAN module on the computer, and then open the AgxCandoUpgradeToolV1.3_boxed.exe software (the sequence cannot be wrong, first open the software and then plug in the module, the device will not be recognized).

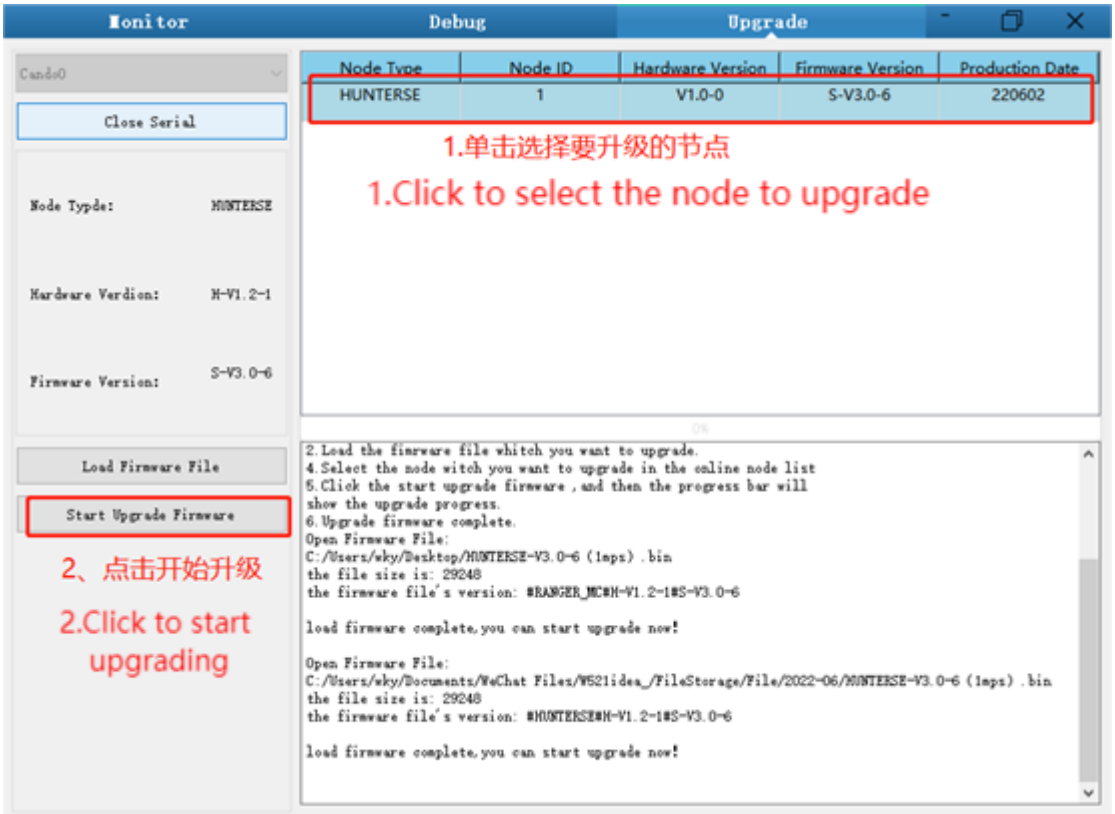
2.Click the Open Serial button, and then press the power button on the car body. If the connection is successful, the version information of the main control will be recognized, as shown in the figure.

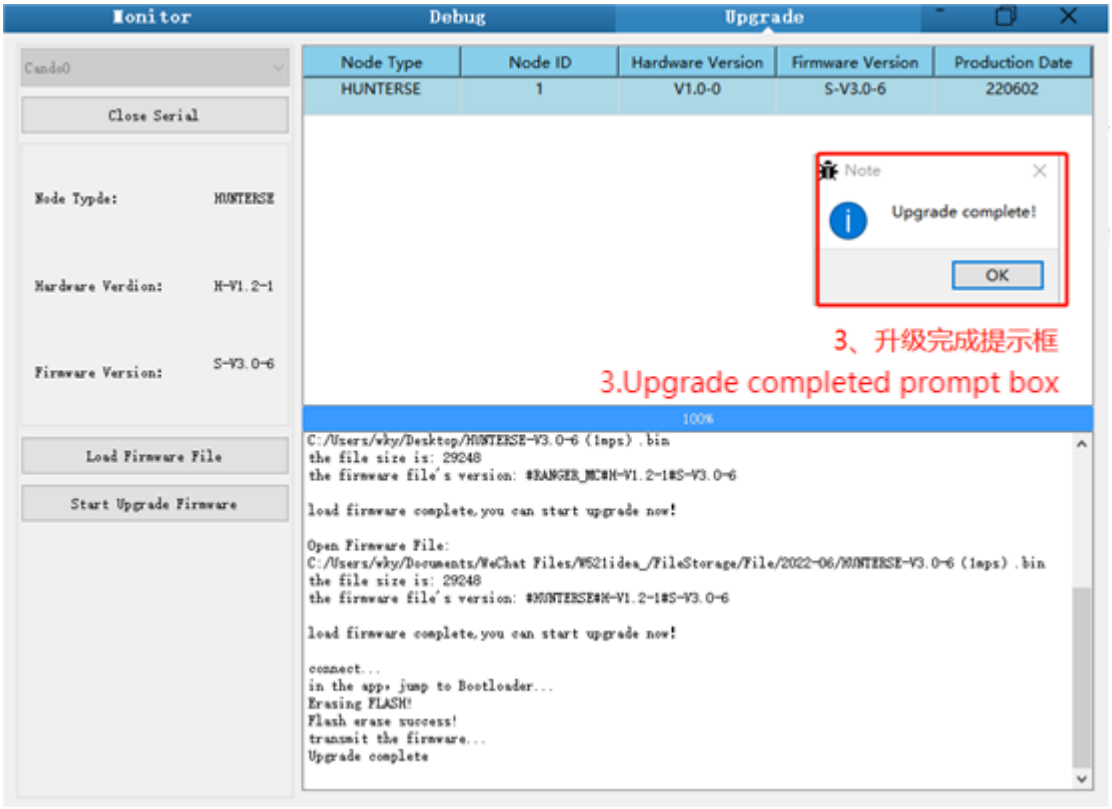


3.Click the Load Firmware File button to load the firmware to be upgraded. If the loading is successful, the firmware information will be obtained, as shown in the figure



4.Click the node to be upgraded in the node list box, and then click Start Upgrade Firmware to start upgrading the firmware. After the upgrade is successful, a pop-up box will prompt.





3.6 SCOUT 2.0 SDK usage example

In order to help users implement robot-related development more conveniently, a cross-platform supported SDK is developed for SCOUT 2.0 mobile robot.SDK software package provides a C++ based interface, which is used to communicate with the chassis of SCOUT 2.0 mobile robot and can obtain the latest status of the robot and control basic actions of the robot. For now, CAN adaptation to communication is available, but RS232-based adaptation is still under way.Based on this, related tests have been completed in NVIDIA JETSON TX2.

3.7 SCOUT2.0 ROS Package usage example

ROS provide some standard operating system services, such as hardware abstraction, low-level device control, implementation of common function, interprocess message and data packet management. ROS is based on a graph architecture, so that process of different nodes can receive, and aggregate various information (such as sensing, control, status, planning, etc.) Currently ROS mainly support UBUNTU.

Development Preparation

Hardware preparation

- CANlight can communication module ×1
- Thinkpad E470 notebook ×1
- AGILEX SCOUT 2.0 mobile robot chassis ×1
- AGILEX SCOUT 2.0 remote control FS-i6s ×1
- AGILEX SCOUT 2.0 top aviation power socket ×1

Use example environment description

- Ubuntu 18.04
- ROS
- Git

Hardware connection and preparation

- Lead out the CAN wire of the SCOUT 2.0 top aviation plug or the tail plug, and connect CAN_H and CAN_L in the CAN wire to the CAN_TO_USB adapter respectively;
- Turn on the knob switch on the SCOUT 2.0 mobile robot chassis, and check whether the emergency stop switches on both sides are released;
- Connect the CAN_TO_USB to the usb point of the notebook. The connection diagram is shown in Figure 3.4.

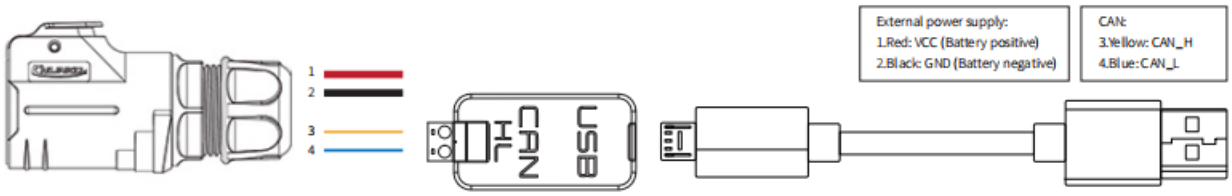


Figure 3.4 CAN connection diagram

ROS installation and environment setting

For installation details, please refer to

http://wiki.ros.org/kinetic/Installation/Ubuntu

Test CANABLE hardware and CAN communication

Setting CAN-TO-USB adaptor

- Enable gs_usb kernel module

```
$ sudo modprobe gs_usb
```

- Setting 500k Baud rate and enable can-to-usb adaptor

```
$ sudo ip link set can0 up type can bitrate 500000
```

- If no error occurred in the previous steps, you should be able to use the command to view the can device immediately

```
$ ifconfig -a
```

- Install and use can-utils to test hardware

```
$ sudo apt install can-utils
```

- If the can-to-usb has been connected to the SCOUT 2.0 robot this time, and the car has been turned on, use the following commands to monitor the data from the SCOUT 2.0 chassis

```
$ candump can0
```

- Please refer to:

[1] https://github.com/agilexrobotics/agx_sdk

[2] https://wiki.rdu.im/_pages/Notes/Embedded-System/-Linux/can-bus-in-linux.html

AGILEX SCOUT 2.0 ROS PACKAGE download and compile

- Download ros package

```
$ sudo apt install -y libasio-dev
$ sudo apt install -y ros-$ROS_DISTRO-teleop-twist-keyboard
```

- Clone compile scout_ros code

```
$ cd ~/catkin_ws/src
$ git clone https://github.com/agilexrobotics/ugv_sdk.git
$ git clone https://github.com/agilexrobotics/scout_ros.git
$ cd ..
$ catkin_make
```

Please refer to： https://github.com/agilexrobotics/scout_ros

Start the ROS node

- Start the based node

```
$ roslaunch scout_bringup scout_robot_base.launch
```

- Start the keyboard remote operation node

```
$ roslaunch scout_bringup scout_teleop_keyboard.launch
```

Github ROS development package directory and usage instructions

*_base:: The core node for the chassis to send and receive hierarchical CAN messages. Based on the communication mechanism of ros, it can control the movement of the chassis and read the status of the bunker through the topic.

*_msgs: Define the specific message format of the chassis status feedback topic.

*_bringup: startup files for chassis nodes and keyboard control nodes, and scripts to enable the usb_to_can module.

4 Q&A

Q: SCOUT 2.0 is started up correctly, but why cannot the RC transmitter control the vehicle body to move?

A: First, check whether the drive power supply is in normal condition, whether the drive power switch is pressed down and whether E-stop switches are released; then, check whether the control mode selected with the top left mode selection switch on the RC transmitter is correct.

Q: SCOUT 2.0 remote control is in normal condition, and the information about chassis status and movement can be received correctly, but when the control frame protocol is issued, why cannot the vehicle body control mode be switched and the chassis respond to the control frame protocol?

A: Normally, if SCOUT 2.0 can be controlled by a RC transmitter, it means the chassis movement is under proper control; if the chassis feedback frame can be accepted, it means CAN extension link is in normal condition. Please check the CAN control frame sent to see whether the data check is correct and whether the control mode is in command control mode. You can check the status of error flag from the error bit in the chassis status feedback frame.

Q: SCOUT 2.0 gives a "beep-beep-beep..." sound in operation, how to deal with this problem?

A: If SCOUT 2.0 gives this "beep-beep-beep" sound continuously, it means the battery is in the alarm voltage state. Please charge the battery in time. Once other related sound occur, there may be internal errors. You can check related error codes via CAN bus or communi-cate with related technical personnel.

Q:Is the tire wear of SCOUT 2.0 is normally seen in operation?

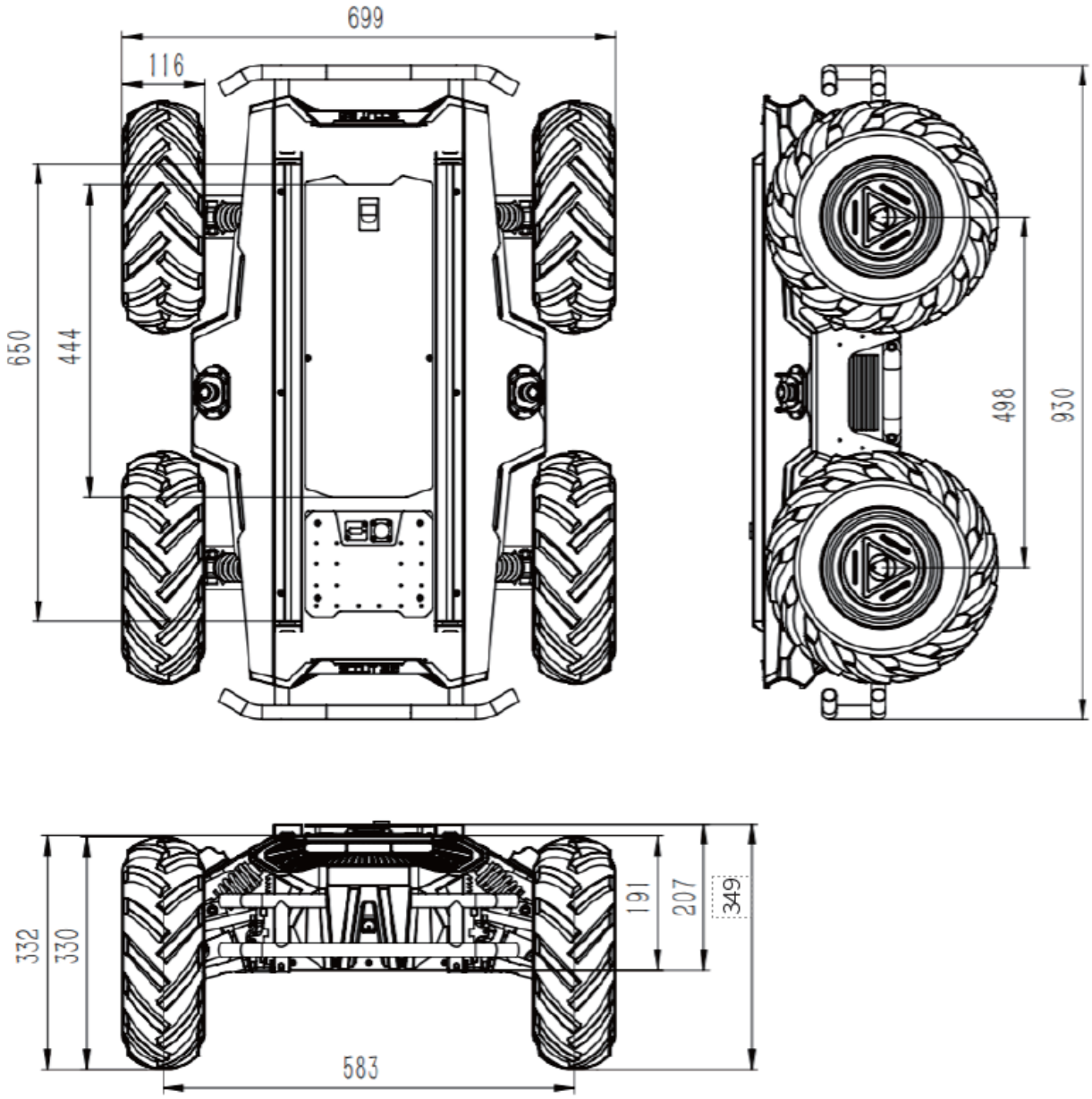
A: The tire wear of SCOUT 2.0 is normally seen when it is running. As SCOUT 2.0 is based on the four-wheel differential steering design, sliding friction and rolling friction both occur when the vehicle body rotates. If the floor is not smooth but rough, tire surfaces will be worn out. In order to reduce or slow down the wear, small-angle turning can be conducted for less turning on a pivot.

Q: When communication is implemented via CAN bus, the chassis feedback command is issued correctly, but why does not the vehicle respond to the control command?

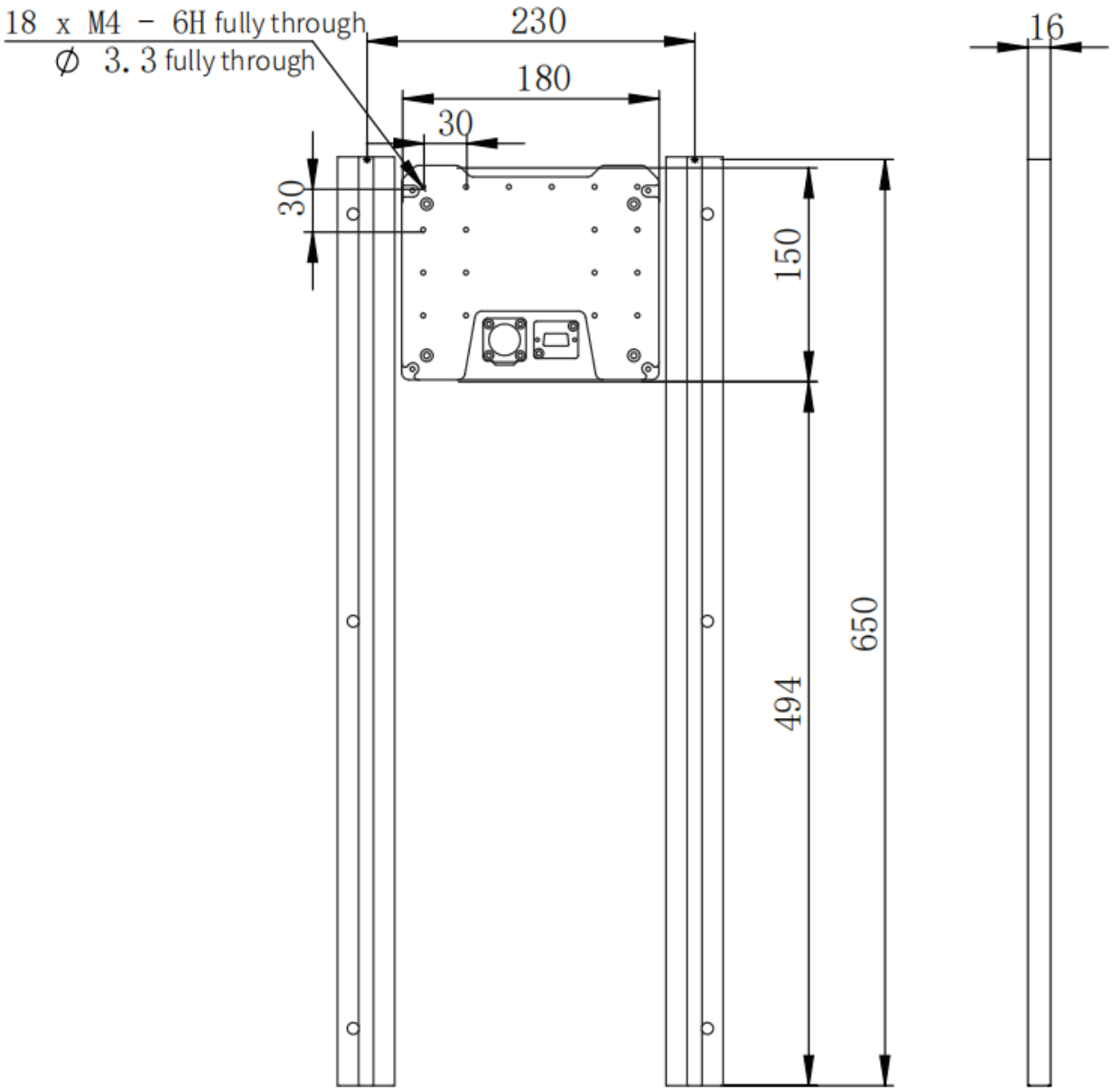
A: There is a communication protection mechanism inside SCOUT 2.0, which means the chassis is provided with timeout protection when processing external CAN control commands. Suppose the vehicle receives one frame of communication protocol, but it does no receive the next frame of control command after 500ms. In this case, it will enter communication protection mode and set the speed to 0. Therefore, commands from upper computer must be issued periodically.

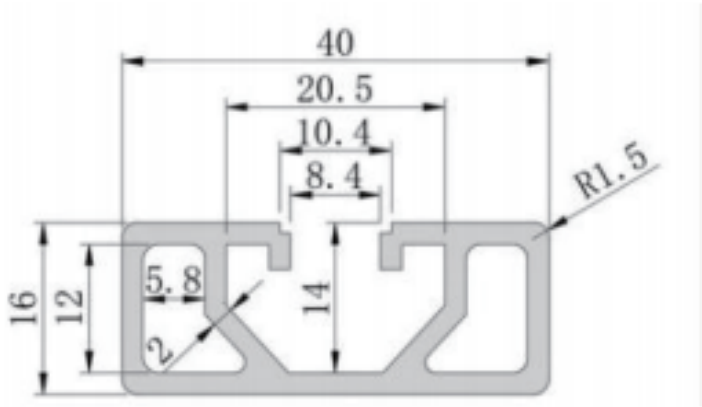
5 Product Dimensions

5.1 Illustration diagram of product external dimensions



6.2 Illustration diagram of top extended support dimensions


18 x M4 - 6H fully through
Ø 3.3 fully through
230
180
30
30
150
494
650
16



AGILE·X
松灵机器人(东莞)有限公司
WWW.AGILEX.AI
TEL:+86-0769-22892150
MOBILE:+86-19925374409