Reeman ROS Serial port protocol explain

Reeman ROS Serial port protocol explain

Version No:1.4.3

Navigation interface protocol

- Serial port baud rate:115200
- Serial port of navigation host:/dev/ttyS1

Instruction Pack：

- The communication mode is serial asynchronous, and the communication content between ROS system and upper computer is ASCII code string.

Frame head(H)	Length(L)	Data (D)	Check bit(S)
AA 54			

Instruction package data definition:

- Frame head(H): AA 54
- Length(L): length of data(D)
- Data (D): String converted byte array
- Check bit(S):xor calculation of length (L) and data(D)

Response packet:

- The communication mode is serial asynchronous, and the communication content between ROS system and upper computer is ASCII code string.

Frame head(H)	Length(L)	Data (D)	Check bit(S)
AA 54			

Answer packet data definition:

- Frame head (H)： AA 54
- Length(L)： length of data(D)
- Data(D): String converted byte array

- Check bit(S):xor calculation of length (L) and data(D)

ROS Active reporting

Report version

- response: ver:RMB101-ROS-I3-B1.07.26
- mark: ver:RMB101-ROS-I3- This part is fixed
- timer：always reporting

Report navigation host Hostname

- response: sys:boot:robot-18
- mark:
  - robot-18 Is the Hostname of the navigation host
- timer： report every 3 seconds within 30 seconds after power on

Report robot positioning status

- response reloc:status:x (x is a number, meaning as follows)
- mark:
  - 1： positioning preparation / trying to locate (including waiting for lidar, map matching process)
- timer：Report any positioning problems after launching navigation

Report robot navigation status (no more updates)

- response:
  - move_status:x (x is a number, meaning as follows)
    - 1 ABORTED (Navigation failed)
    - 2 SUCCEEDED (Navigation successful)
    - 4 EXCEPTION(In navigation,)(In navigation,blocked by obstacles,local path reporting)
    - 5 There are obstacles at the target point
    - 6 PENDING
    - 9 ACTIVE
    - 10 PREEMPTED

        - 11 REJECTED
        - 12 PREEMPTING
        - 13 RECALLING
        - 14 RECALLED
        - 15 LOST
    - bat:reached: navigate to charging area, docking charging point
- timer：actively report after navigation

Report robot navigation status

- response:
    - nav_stat:x(x is a number, meaning as follows)
        - 4 EXCEPTION(In navigation, blocked by obstacles,local path reporting)
    - 5 There are obstacles at the target point

Report robot detailed navigation status

- response:
    - nav_res:{"res": 1, "reason": 1, "sensor": "100011", "dist": "0.80"}
    - res
        - -1 unknown
        - 0 Navigation successful
        - 1 Stop navigation
        - 2 New goals
    - reason：(only when res is 1,Report fields)
        - 1 Critical sensor failure(Only in this way can the sensor be reported)
            - sensor Six numbers respectively represent whether the laser, odometer, chassis, IMU, wheel overcurrent protection and label camera work normally. 0 is normal, 1 is abnormal
        - 2 receive cancel navigation command

        - 4 obstacles ahead
        - 5 there are obstacles at the target point
        - 6 not in the work area
        - 7 enter a restricted area or identify a restricted area label
        - 8 docking charging point
        - 9 over the set label recognition interval distance x no label is recognized
        - -1 unknown cause (possible causes are as follows)
            - positioning inside or outside the virtual wall
    - Dist:(this field will be reported only when res is 1)
        - Linear distance between current position and target point
        - type is string
        - keep two decimal places
    - mileage：
        - mileage of this navigation

- timer：actively report after navigation

Report the nearest obstacle distance in front of the robot

- response: laser[x]
- mark:
    - here, X is the distance, in meters, and the precision is two decimal places
    - if there is no obstacle ahead, X default is 1000
- timer：Keep reporting

Report robot sensor status, if there is a problem with the sensor

- response：sensor_state:1001
- mark
    - [1 0 0 1] respectively indicates whether [laser odometer chassis IMU] works normally
    - 0 for normal operation, 1 for abnormal operation

- timer：report when there is a problem with the sensor

Report the QR code currently recognized by the robot to the upper computer

- response：visual_mark[name:A,B]
- mark
  - multiple QR codes can be reported at the same time
  - It can be used to check whether the QR code navigation function is normal
- timer：recognize the QR code and report it once a second

After the application map is completed and the initial positioning is completed, it is reported to the upper computer

- response：apply_map[map_name:xxxx]
- mark
  - xxxx Map completed for switching
- timer：This interface will be triggered when the machine is turned on and the map is switched between the web page and the upper computer

---

Upper computer initiative request

system instruction

Query wifi ip

- request: ip:request
- response: ip:ssid:x.x.x.x
- mark: When the host WiFi is connected, the return value carries the SSID of the current WiFi

Query network port ip

- request: ip_lan:request
- response: ip_lan:x.x.x.x
- mark: When the host network interface connects the network cable, return the IP address of the network interface

query edition

- request: sys:version
- response: ver:RMB101-ROS-I3-B1.07.26
- mark: ver:RMB101-ROS-I3- This part is fixed

Power off the machine

- request: sys:shutdown
- response: None
- mark: About 30 seconds

Restart the navigation host

- request: sys:reboot
- response: None
- mark: It's just a navigation host. It takes about 90 seconds

Navigation host connection to WIFI

- request: wifi[ssid name;pwd password]
- response:
  - wifi:connect success:Connection successful
  - wifi:connect fail:Connection failed
  - wifi:connecting:Connection in progress
- mark:
  - WiFi only supports WPA / WPA2 Personal protocol
  - ssid is a fixed field, and name is WiFi name
  - pwd is a fixed field, and password is WiFi password

Get navigation host Hostname

- request: hostname:get
- response: sys:boot:robot-18
- mark:
  - robot-18 Is the hostname of the navigation host

Map switching

- request: call_web[apply_map:Map name]

- response: apply_map[map_name:xxxx]
- mark
  - xxxx switch completed map
  - timer：this interface will be triggered when the map is switched on when the web page and the host computer are switched
  - Map name is generally non Chinese

Get the robot's current coordinates

- request:
  - nav:get_pose
- response:
  - nav:pose[x,y,theta]　Current robot coordinates
  - nav:pose:notfound　Robot is positioning
- mark:
  - x,y,theta:They are floating-point numbers with two decimal places reserved, where x, y are robot coordinates, theta is robot angle on the map, and the unit is degree

Get the coordinates of a calibrated point

- request: nav:get_flag_point[a_point]
- response:
- get_flag_point[x,y,theta]　Coordinates of the calibration point
- get_flag_point:notfound　The calibration point cannot be found
- mark:
- a_point is the name of the calibration point
- x,y,theta:They are floating-point numbers with two decimal places reserved, where x, y are robot coordinates, theta is robot angle on the map, and the unit is degree

Set calibration point

- request: nav:set_flag_point[x,y,theta,a_point]
- response:
- set_flag_point:success　Set the calibration point successfully

- set_flag_point:failed　Set calibration point failed
- mark:
- x,y,theta:They are floating-point numbers with two decimal places reserved, where x, y are robot coordinates, theta is robot angle on the map, and the unit is degree

Delete a calibration point

- request: nav:del_flag_point[a_point]
- response:
- del_flag_point:0　Unable to find the calibration point to delete
- del_flag_point:1　Delete calibration point succeeded
- mark:
- x,y,theta They are floating-point numbers with two decimal places reserved, where x, y are robot coordinates, theta is robot angle on the map, and the unit is degree

Restart the entire machine

- request: sys:reboot_machine
- response:None
- mark:
  - Restart the whole robot

Get the coordinates of robot in pixel coordinate system

- request: nav:get_pixel_pose
- response:
- nav:pixel_pose[x,y]：x y is pixel coordinate system coordinate
- nav:pixel_pose:notfound：Robot is positioning
- mark:
  - For self drawing map

To open up Web interface

The use of the development interface requires that the upper computer and the navigation host are in the same LAN

Get map

- method: GET
- path: http://ip/reeman/map
- response:
  - Content-Type: application/json
  - { "width": 861, "height": 619, "image_url": "..." }
- mark:
- ip is the navigation host ip

Get robot position

- method: GET
- path:http://ip/reeman/pose
- response:
  - Content-Type: application/json
  - { "x": 297, "y": 251, "theta": 0.97}
- mark:
- ip is the navigation host ip

Get current map name

- method: GET
- path: http://ip/reeman/current_map
- response:
  - Content-Type: application/json
  - { "name": " Reeman Reception" }
- mark:
- ip is the navigation host ip

Get a list of online machine target points

- method: POST
- path: http://navi.rmbot.cn/OpenAPISpring/ros/locations/find
- body: { "device":"rmb102a-190310-001-001"}

- response:
  - Content-Type: application/json
  - { "msg": " success","code":0,"data":{"result":{"id":103,"device":"rmb102a-190310-001-001","location":"","deviceId":244,"updateTime":"2019-11-22%00:57:27.000+0000"}} }

Get LAN machine target list

- method: GET
- path: http://ip/reeman/android_target
- response:
  - Content-Type: application/json
  - {"a_point":["-3.66","-0.06","-110.96"],"b_point":["-4.27","-1.36","0.00"],"c_point":["-3.47","-2.27","73.41"],"d_point":["-2.54","-1.29","180.00"]}
  - mark:
- ip is the navigation host ip

Navigation and positioning

####Given target point coordinate navigation

- request: goal:nav[x,y,yaw]
- response: None
- mark
  - x,y is the coordinate, yaw is the angle, and the value range is [- 180, 180]

Navigation and charging with given target coordinates

- request: goal:charge[x,y,yaw]
- response: None
- mark
  - x,y is the coordinate, yaw is the angle, and the value range is [- 180, 180]
  - The robot will automatically charge when it navigates to this point

Given target point name navigation

- request: point[Reception]

- response:
  - point:1：The target point cannot be found
  - point:0：Find the target point and start navigation
- mark:
  - The Chinese target point needs to be set under the "calibration position" option on the webpage side
  - Chinese character encoding mode utf8, non Chinese character encoding mode ASCII encoding

Navigation and charging given target point name

- request: point_charge[charging_pile]
- response:
  - point_charge:1：The target point cannot be found
  - point_charge:0：Find the target point and start navigation
- mark:
  - The Chinese target point needs to be set under the "calibration position" option on the webpage side
  - The robot will automatically charge when it navigates to this point
  - Chinese character encoding mode utf8, non Chinese character encoding mode ASCII encoding

Cancel navigation

- request: cancel_goal
- response: None

Repositioning

System relocation sequence: charging point > system location file

- request：
  - nav:reloc Overall situation Repositioning
  - nav:reloc[x,y] Relocate near (x, y)
  - nav:reloc[x,y,yaw] Repositioning near (x, y) and yaw
  - nav:reloc_point[charging_pile] Repositioning near the charging pile

- response: None
- mark：
  - Chinese target point relocation (nav:reloc_point[charge pile]) will return whether the target point is found
    - point:1Chinese target point not found
  - Select areas with obvious features as much as possible during relocation
  - When repositioning, try not to let too many people surround the robot to avoid affecting the positioning effect.
  - When using Chinese for relocation, UTF-8 encoding is required for Chinese

---

Chassis Control

Direct docking charging_pile

- request: goal:just_charge
- response: None

Cancel charging

- request: bat:uncharge
- response: None
- mark: This command is valid during the backward process of docking charging pile

Mobile robot

- request:
  - move[distance,angle] [int, int] type
  - move[distance,angle,speed] [int, int, double] type
- response: move:done:xx
- mark:
  1. Distance and angle must have a value of 0, that is, the robot can only walk in a straight line or turn, but not in an arc
  2. If move [0,0] is sent, it means stop moving
  3. distance Unit: mm

4. Angle unit is Euler angle, value range is [- 180,180]
5. Speed is the speed parameter. If it is not specified, the robot will walk at the default speed (straight line is 0.3 meters per second, angle is 40 degrees per second)
6. The speed should be greater than 0, otherwise it will be processed according to the default speed
7. If speed is specified and distance is not 0, speed indicates the speed of straight line, in meters / second
8. If speed is specified and angle is not 0, speed is the speed of the corner, in degrees per second
9. reply move:done : 16: straight walk complete
10. reply move:done : 17: left turn complete
11. reply move:done : 18: right turn complete

Distribution robot

container lock

- request: sys:lock
- response: None
- mark:
    1. This instructions only valid for distribution robots

Container unlocking

- request: sys:unlock
- response: None
- mark:
    1. This instructions only valid for distribution robots

Clear wheel protection status

- request: sys:reset_wheel_protection
- response: None
- mark:
    1. This instructions only valid for distribution robots

Infrared anti falling - on and off

- request:
  sys:close_anti_fall Turn off the infrared anti falling function
  sys:open_anti_fall Turn on the infrared anti falling function
- response: None
- mark:
  1. This instructions only valid for distribution robots

Get RFID module power

- request:
  rfid_set[request_pwr]
- response: rfid[pwr:12]
- mark:
  - Power is 12
  - This instructions only valid for distribution robots

RFID function card identification

- Ordinary card
  - A314BC
- Electronic fence card
  - W.A314BC
- Anti falling switch card
  - F.A314BC

Query RFID version number

- request: rfid_request[request_version]
- response: rfid[version:xxx]
- mark: xxx Direct display

Transmission upper computer protocol to base adapter

- request: bottom:***
- response: None
- mark: This instruction is a new protocol instruction for upper computer and base

Clear forbidden area status

- request: reset_rfid_type
- response: None
- mark: This instruction means that the robot has entered the forbidden area. People used to push it away, and then Android sends this instruction to ROS, and they can continue to go. Otherwise, it will wait until it receives a label of another type before it knows that it has left the restricted area.

Chase the wind robot

###Action control

- request: action[command]
- response: None
- mark:
    1. This order is only valid for Chase the wind robot
    2. command Command is motor control protocol command Example：0x80 0x01 0x05 0x00
    3. command The specific agreement and definition are in 《Wind Chasing Motion Control Communication Protocol v2.3》
    4. The specific splicing rule of request is: "action [" corresponding ASCII code value + command direct instruction value + "]" corresponding ASCII code value
    5. Request example: 0x61 0x63 0x74 0x69 0x6f 0x6e 0x5b [command instruction value] 0x5d

---

Matters needing attention

- When the charging point is connected, if the navigation command, charging command, and turning command are sent, it will first walk 0.3m forward, and then execute the command
- When the charging line is connected for charging, if the upper computer sends navigation instruction or charging instruction, the instruction will be discarded directly