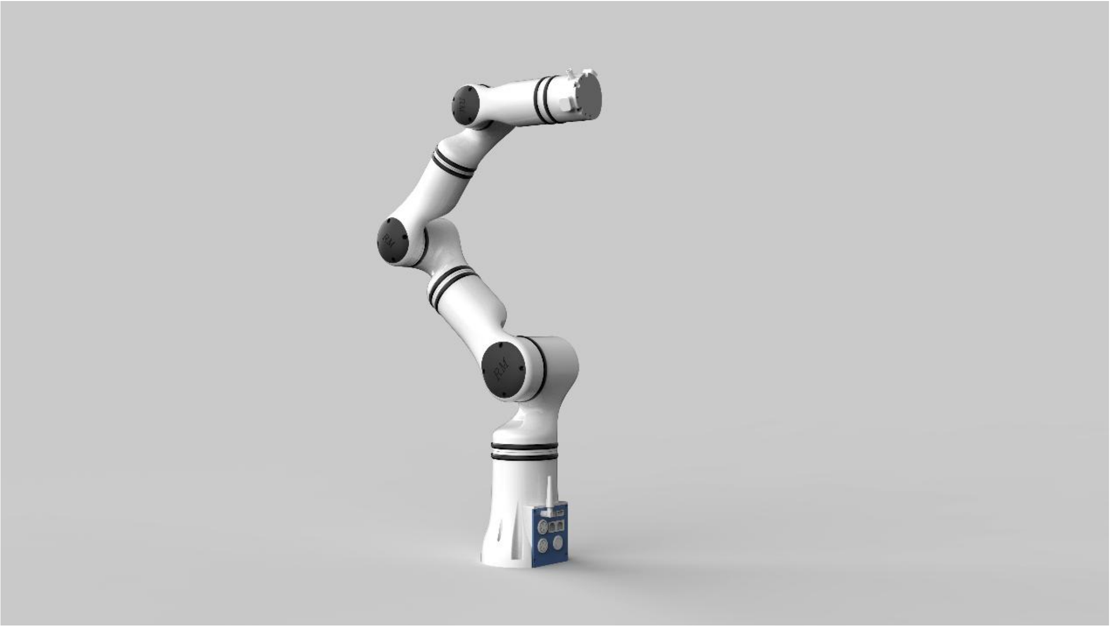
RealMan Robotic Arm rm_driver Topic Detailed Description (ROS2) V1.0



RealMan Intelligent Technology (Beijing) Co., Ltd.

Revision History:

No.	Date	Comment
V1.0	11/25/2023	Draft

Content

1. Introduction

RealMan provides ROS2 function packages based on API to help users control the robotic arm using ROS2. If you want to learn more about controlling the robotic arm, you can refer to the API documentation and instructions. In practical use, the user can establish communication with the robotic arm through the Ethernet port and control the robotic arm.

2. Error Description

2.1 Controller error type

Serial No.	Error code (hexadecimal)	Error content
1	0x0000	System is normal
2	0x1001	Joint communication is abnormal
3	0x1002	The target angle exceeds the limit
4	0x1003	This position is inaccessible and is a singular point
5	0x1004	Real-time kernel communication error
6	0x1005	Joint communication bus error
7	0x1006	Planning layer kernel error
8	0x1007	Joint Overspeed
9	0x1008	The end interface board cannot be connected
10	0x1009	Overspeed limit
11	0x100A	Overacceleration limit
12	0x100B	Joint brake is not opened
13	0x100C	Overspeed during drag teach
14	0x100D	Robotic arm collision
15	0x100E	No work coordinate system is available
16	0x100F	No tool coordinate system is available

17	0x1010	Joint failure enabling error

2.2 Joint error type

Serial No.	Error code (hexadecimal)	Error content
1	0x0000	Joint is normal
2	0x0001	FOC error
3	0x0002	Overvoltage
4	0x0004	Undervoltage
5	0x0008	Overtemperature
6	0x0010	Start failed
7	0x0020	Encoder error
8	0x0040	Overcurrent
9	0x0080	Software error
10	0x0100	Temperature sensor error
11	0x0200	Position limit-out error
12	0x0400	Illegal joint ID
13	0x0800	Position tracking error
14	0x1000	Current detection error
15	0x2000	Brake opening failed
16	0x4000	Position command step warning
17	0x8000	Multi-coil joint's coil lost the number
18	0xF000	Communication frame loss

2.3 API error type

Serial No.	Error code (hexadecimal)	Error content
1	0x0000	The system is running normally
2	0x0001	Message request returns FALSE
3	0x0002	The robotic arm is not initialized or the input model is illegal
4	0x0003	Illegal timeout
5	0x0004	Socket initialization failed
6	0x0005	Socket connection failed
7	0x0006	Socket sending failed
8	0x0007	Socket communication timeout
9	0x0008	Unknown error
10	0x0009	Incomplete data
11	0x000A	Array length error

12	0x000B	Data type error
13	0x000C	Model error
14	0x000D	Missing callback function
15	0x000E	The robotic arm stopped abnormally
16	0x000F	The trajectory file name is too long
17	0x0010	Trajectory file verification failed
18	0x0011	The trajectory file read failed
19	0x0012	The controller busy, please try again later
20	0x0013	Illegal input
21	0x0014	The data queue is full
22	0x0015	Calculation failed
23	0x0016	File opening failed
24	0x0017	Force control calibration manual stop
25	0x0018	No tracks to save

3. ROS Function Package Robotic Arm Instructions

This section describes how to query and control the robotic arm through the topic of ROS.

3.1 Joint configuration

3.1.1 Clear the joint's error code

Function description	Clear the joint's error code
Parameter description	Jointerrclear.msg uint8 joint_num: the corresponding joint number, from the base to the robotic arm gripper, the number is 1-6. bool block: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub /rm_driver/set_joint_err_clear_cmd rm_ros_interfaces/msg/Jointerrclear "joint_num: 1 block: true"
Return □value	true - set successfully, false - set failed
	ros2 topic echo /rm_driver/set_joint_err_clear_result

3.2 Functions related to the work coordinate system settings

3.2.1 Change the current work coordinate system

Function description	Change the current work coordinate system
Parameter description	ROS built-in msg std_msgs::msg::String
Command example	ros2 topic pub /rm_driver/change_work_frame_cmd std_msgs/msg/String "data: 'Base'"
Return 回value	true - set successfully, false - set failed
	ros2 topic echo /rm_driver/change_work_frame_result

3.3 Coordinate system query

3.3.1 Get the current tool coordinate system

Function description	Get the current tool coordinate system
Parameter description	ROS built-in msg std_msgs::msg::Empty
Command example	ros2 topic pub --once /rm_driver/get_current_tool_frame_cmd std_msgs/msg/Empty "{}"
Return 回value	Current tool coordinate system name
	ros2 topic echo /rm_driver/get_current_tool_frame_result

3.3.2 Get all tool coordinate system names

Function description	Get the current tool coordinate system
Parameter description	ROS built-in msg std_msgs::msg::Empty
Command example	ros2 topic pub /rm_driver/get_all_tool_frame_cmd std_msgs/msg/Empty "{}"
Return 回value	All names of the current tool coordinate system
	ros2 topic echo /rm_driver/get_all_tool_frame_result

3.3.3 Get the current work coordinate system

Function description	Change the current work coordinate system
Parameter	ROS built-in msg std_msgs::msg::Empty

description	
Command example	ros2 topic pub --once /rm_driver/get_curr_workFrame_cmd std_msgs/msg/Empty "{}"
Return 回value	true - set successfully, false - set failed
	ros2 topic echo /rm_driver/get_curr_workFrame_result

3.3.4 Get all work coordinate system names

Function description	Change the current work coordinate system
Parameter description	ROS built-in msg std_msgs::msg::Empty
Command example	ros2 topic pub --once /rm_driver/change_work_frame_cmd std_msgs/msg/String "data: 'Base'"
Return 回value	All work coordinate system names
	ros2 topic echo /rm_driver/change_work_frame_result

3.4 Functions related to the arm state query

3.4.1 Get the current state of the robot arm (return each joint angle + Euler angle)

Function description	Retrieve the current state of the robotic arm
Parameter description	ROS built-in msg std_msgs::msg::Empty
Command example	ros2 topic pub --once /rm_driver/get_current_arm_state_cmd std_msgs/msg/Empty "{}"
Return 回value	The current robotic arm joint state (angle) + pose information (Euler angle) + error information
	ros2 topic echo /rm_driver/get_current_arm_original_state_result

3.4.2 Get the current state of the robotic arm (return each joint radians + quaternion)

Function description	Retrieve the current state of the robotic arm
Parameter description	ROS built-in msg std_msgs::msg::Empty

Command example	ros2 topic pub --once /rm_driver/get_current_arm_state_cmd std_msgs/msg/Empty "{}"
Return 回value	The current robotic arm joint state (radians) + pose information (quaternion) + error information
	ros2 topic echo /rm_driver/get_current_arm_state_result

3.5 Functions related to motion planning of the robotic arm

3.5.1 Joint space motion

Function description	Joint space move MOVEJ
Parameter description	Movej.msg float32[6] joint: joint angle, unit: radians. uint8 speed - speed percentage ratio coefficient, 0-100. bool block: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	6-degree of freedom ros2 topic pub --once /rm_driver/movej_cmd rm_ros_interfaces/msg/Movej "joint: [0, 0, 0, 0, 0, 0] speed: 20 block: true dof: 6" 7-degree of freedom ros2 topic pub --once /rm_driver/movej_cmd rm_ros_interfaces/msg/Movej "joint: [0, 0, 0, 0, 0, 0, 0] speed: 20 block: true dof: 7"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/get_current_arm_state_result

3.5.2 Linear motion in Cartesian space

Function description	Linear motion in Cartesian space MOVEL
Parameter description	Movel.msg geometry_msgs/Pose pose: robotic arm pose, geometry_msgs/Pose

	type, x, y, z coordinates (float type, unit: m) + quaternion. uint8 speed: speed percentage ratio coefficient, 0-100. bool block: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.

Command example	First, use MoveJP
	ros2 topic pub --once /rm_driver/movej_p_cmd rm_ros_interfaces/msg/Movejp "pose:
	  position:
	    x: -0.317239
	    y: 0.120903
	    z: 0.255765
	  orientation:
	    x: -0.983404
	    y: -0.178432
	    z: 0.032271
	    w: 0.006129
	speed: 20
	block: true"
	Then use MoveL
	ros2 topic pub --once /rm_driver/movel_cmd rm_ros_interfaces/msg/Movel "pose:
	  position:
	    x: -0.317239
	    y: 0.120903
	    z: 0.295765
	  orientation:
	    x: -0.983404
	    y: -0.178432
	    z: 0.032271
	    w: 0.006129
	speed: 20
	block: true"

Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/get_current_arm_state_result

3.5.3 Circular motion in Cartesian space

Function description	Circular motion in Cartesian space MOVEC
Parameter description	Movec.msg geometry_msgs/Pose pose_mid: middle pose, geometry_msgs/Pose type, x, y, z coordinates (float type, unit: m) + quaternion. geometry_msgs/Pose pose_end: end pose, geometry_msgs/Pose type, x, y, z coordinates (float type, unit: m) + quaternion. uint8 speed: speed percentage ratio coefficient, 0-100. bool block: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.

Command example	First, use movej_p to reach the specified position
	ros2 topic pub --once /rm_driver/movej_p_cmd rm_ros_interfaces/msg/Movejp "pose:
	  position:
	    x: 0.274946
	    y: -0.058786
	    z: 0.299028
	  orientation:
	    x: 0.7071
	    y: -0.7071
	    z: 0.0
	    w: 0.0
	speed: 0
	block: true"
	Use movec to reach the specified position
	ros2 topic pub --once /rm_driver/movec_cmd rm_ros_interfaces/msg/Movec "pose_mid:
	  position:
	    x: 0.324946
	    y: -0.008786
	    z: 0.299028
	  orientation:
	    x: 0.7071
	    y: -0.7071
	    z: 0.0

	w: 0.0 pose_end: position: x: 0.274946 y: 0.041214 z: 0.299028 orientation: x: 0.7071 y: -0.7071 z: 0.0 w: 0.0 speed: 20 block: false"
Return □value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/get_current_arm_state_result

3.5.4 Joint angle CANFD transmission

Function description	Joint angle CANFD transmission
Parameter description	Jointpos.msg float32[6] joint: joint angle, unit: radians. bool follows: follow state, true: high following, false: low following, default high following if not set. float32 expand: expand joint, unit: radians.
Command example	Transmission needs to send multiple continuous points to achieve, simply by the following command and can not achieve the function, the current moveit2 control using angle transmission control mode. ros2 topic pub /rm_driver/movej_canfd_cmd rm_ros_interfaces/msg/Jointpos "joint: [0, 0, 0, 0, 0, 0] follow: false expand: 0.0 dof: 6"
Return □value	Success: no return value; Failure return: the driver terminal returns an error code.

3.5.5 Pose CANFD transmission

Function description	Pose CANFD transmission
Parameter description	Jointpos.msg geometry_msgs/Pose pose: transmission pose, geometry_msgs/Pose type, x, y, z coordinates (float type, unit: m) + quaternion. bool follows: follow state, true: high following, false: low following, default high following if not set.

Command example	It needs to be a large number (10 or more) of continuous position points, simply by the following command and can not achieve the function, with more than a 2ms period continuous release.
	ros2 topic pub /rm_driver/movep_canfd_cmd rm_ros_interfaces/msg/Cartepos "pose:
	  position:
	    x: 0.0
	    y: 0.0
	    z: 0.0
	  orientation:
	    x: 0.0
	    y: 0.0
	    z: 0.0
	    w: 1.0
	follow: false"

Return □value	Success: no return value; Failure return: the driver terminal returns an error code.

3.5.6 Joint space planning to target pose

Function description	Joint space planning to target pose MOVEJP
Parameter description	Movejp.msg geometry_msgs/Pose pose: target pose, x, y, z coordinates (float type, unit: m) + quaternion. uint8 speed - speed percentage ratio coefficient, 0-100. bool block: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/movej_p_cmd rm_ros_interfaces/msg/Movejp "pose: position:

	x: -0.317239 y: 0.120903 z: 0.255765 orientation: x: -0.983404 y: -0.178432 z: 0.032271 w: 0.006129 speed: 20 block: true"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/movej_p_result

3.5.7 Trajectory emergency stop

Function description	Motion planning trajectory emergency stop
Parameter description	ROS official msg std_msgs::msg::Bool bool data: whether the trajectory is emergency stop, true: emergency stop, false: not emergency stop.
Command example	ros2 topic pub /rm_driver/move_stop_cmd std_msgs/msg/Bool "data: true"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/move_stop_result

3.6 Functions related to controller configuration

3.6.1 Get the controller's version

Function description	Get the controller's version
Parameter description	Armsoftversion.msg string plan version: the read user interface kernel version number. string ctrlversion: real-time kernel version number. string kernal1: the version number of sub-core 1 of the real-time kernel. string kernal2: the version number of sub-core 2 of the real-time kernel. string product version: robotic arm model.

Command example	ros2 topic pub /rm_driver/get_arm_software_version_cmd std_msgs/msg/Empty "{}"
Return 回value	Successful return: version information; Failure return: the driver terminal returns an error code.
	ros2 topic echo /rm_driver/get_arm_software_version_result

3.7 Functions related to the control of the end gripper (optional)

3.7.1 Get the controller's version

Function description	Get the controller's version
Parameter description	Armsoftversion.msg string planversion: the read user interface kernel version number. string ctrlversion: real-time kernel version number. string kernal1: the version number of sub-core 1 of the real-time kernel. string kernal2: the version number of sub-core 2 of the real-time kernel. string productversion: robotic arm model.
Command example	ros2 topic pub /rm_driver/get_arm_software_version_cmd std_msgs/msg/Empty "{}"
Return 回value	Successful return: version information; Failure return: the driver terminal returns an error code.
	ros2 topic echo /rm_driver/get_arm_software_version_result

3.8 Functions related to the IO configuration of the end tool

3.8.1 Setting the tool voltage output

Function description	Setting the tool voltage output
Parameter description	ROS built-in message file: std_msgs::msg::UInt16 uint16 data: power output type, range: 0-3 0-0V, 1-5V, 2-12V, 3-24V.
Command example	ros2 topic pub --once /rm_driver/set_tool_voltage_cmd std_msgs/msg/UInt16 "data: 0"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_tool_voltage_result

3.9 Functions related to the control of the end gripper (optional)

The RealMan robotic arm is equipped with an Inspire Robots EG2-4C2 gripper.

The robotic arm controller has made the gripper's ROS control mode available to the user to facilitate user operation.

3.9.1 Setting the gripper pick

Function description	Setting the gripper pick
Parameter description	Gripperpick.msg uint16 speed: 1-1000, representing the opening and closing speed of the gripper, dimensionless. uint16 force: 1-1000, representing the gripping force of the gripper, maximum 1.5 kg. bool block: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/set_gripper_pick_cmd rm_ros_interfaces/msg/Gripperpick "speed: 200 force: 200 block: true"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_gripper_pick_result

3.9.2 Setting the gripper pick-on

Function description	Setting the gripper pick-on
Parameter description	Gripperpick.msg uint16 speed: 1-1000, representing the opening and closing speed of the gripper, dimensionless. uint16 force: 1-1000, representing the gripping force of the gripper, maximum 1.5 kg. bool block: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/set_gripper_pick_on_cmd rm_ros_interfaces/msg/Gripperpick "speed: 200 force: 200 block: true"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_gripper_pick_on_result

3.9.3 Setting the gripper to the given position

Function description	Setting the gripper to the given position
Parameter description	Gripperset.msg uint16 position: target position of the gripper, range: 1-1000, representing the opening degree of the gripper: 0-70 mm. bool block: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/set_gripper_position_cmd rm_ros_interfaces/msg/Gripperset "position: 500 block: true"
Return □value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_gripper_position_result

3.10 Functions related to the drag teach and trajectory reproduction

3.10.1 Set the force-position mixing control

Function description	Set the force-position mixing control
Parameter description	Setforceposition.msg uint8 sensor: 0 - One-axis force; 1 - Six-axis force uint8 mode: 0 - Base coordinate system force control; 1 - Tool coordinate system force control uint8 direction: Force control direction; 0 - Along the X-axis; 1 - Along the Y-axis; 2 - Along the Z-axis; 3 - Along the RX posture direction; 4 - Along the RY posture direction; 5 - Along the RZ posture direction int16 n: The value of force, unit: N, accuracy: 0.1N bool block: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/set_force_postion_cmd rm_ros_interfaces/msg/Setforceposition "sensor: 1 mode: 0 direction: 2 n: 3 block: false"

Return ⊡value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_force_postion_result

3.10.2 Stop the force-position mixing control

Function description	Stop the force-position mixing control
Parameter description	std_msgs::msg::Bool bool data: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub /rm_driver/stop_force_postion_cmd std_msgs/msg/Bool "data: true"
Return ⊡value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/clear_force_data_result

3.11 Functions related to the use of six-axis force sensors at the end (optional)

The RealMan RM-65F robotic arm has an integrated six-axis force sensor at the end without external wiring. Users can operate the six-axis force through ROS topics.

3.11.1 Clearing the six-axis force data

Function description	Clearing the six-axis force data
Parameter description	std_msgs::msg::Bool bool data: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub /rm_driver/clear_force_data_cmd std_msgs/msg/Bool "data: true"
Return ⊡value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/clear_force_data_result

3.12 Functions related to the control of the five-finger dexterous hand (optional)

The RealMan RM-65 robotic arm has been equipped with a five-finger dexterous hand at the end. Users can set the hand through the ROS.

3.12.1 Setting the serial number of the dexterous hand posture

Function description	Setting the serial number of the dexterous hand posture
Parameter description	Handposture.msg uint16 posture_num: the serial number of the posture pre-saved in the dexterous hand, ranging from 1 to 40. bool data: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/set_hand_posture_cmd rm_ros_interfaces/msg/Handposture "posture_num: 1 block: true"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_hand_posture_result

3.12.2 Set the dexterous hand action sequence number

Function description	Set the dexterous hand action sequence number
Parameter description	Handseq.msg uint16 seq_num： the serial number of the action sequence pre-saved in the dexterous hand, ranging from 1 to 40. bool data: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/set_hand_seq_cmd rm_ros_interfaces/msg/Handseq "seq_num: 1 block: true"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_hand_seq_result

3.12.3 Setting the angles of various degrees of freedom for the dexterous hand

Function description	Setting the angles of various degrees of freedom for the dexterous hand
Parameter description	Handangle.msg int16[6] hand_angle: hand angle array, the range is 0 to 1000, and -1 represents that no operation is performed on this degree of freedom and

	the current state remains. bool data: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/set_hand_angle_cmd rm_ros_interfaces/msg/Handangle "hand_angle: - 0 - 0 - 0 - 0 - 0 - 0 block: true"
Return □value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_hand_angle_result

3.12.4 Setting the dexterous hand speed

Function description	Setting the dexterous hand speed
Parameter description	Handspeed.msg uint16 hand_speed: hand speed, range: 1-1000. bool data: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/set_hand_speed_cmd rm_ros_interfaces/msg/Handspeed "hand_speed: 200 block: true"
Return □value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_hand_speed_result

3.12.5 Setting the force threshold of the dexterous hand

Function description	Setting the dexterous hand speed
Parameter description	Handforce.msg uint16 hand_force: hand force, range: 1-1000. bool data: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.

Command example	ros2 topic pub --once /rm_driver/set_hand_force_cmd rm_ros_interfaces/msg/Handforce "hand_force: 200 block: true"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_hand_force_result

3.13 Lifting mechanism

The RealMan robotic arm can be integrated with the self-developed lifting mechanism.

3.13.1 Speed open-loop control of the lifting mechanism

Function description	Speed open-loop control of the lifting mechanism
Parameter description	Liftspeed.msg int16 speed: speed percentage, -100-100, Speed < 0: the lifting mechanism moves downward, Speed > 0: the lifting mechanism moves upward, Speed = 0: the lifting mechanism stops. bool data: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub /rm_driver/set_lift_speed_cmd rm_ros_interfaces/msg/Liftspeed "speed: 100"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_lift_speed_result

3.13.2 Position closed-loop control of the lifting mechanism

Function description	Position closed-loop control of the lifting mechanism
Parameter description	Lift height.msg uint16 height: target height, unit: mm, range: 0-2600. uint16 speed: speed percentage, 1-100. bool data: whether it is a blocking mode, bool type, true: blocking, false: non-blocking.
Command example	ros2 topic pub --once /rm_driver/set_lift_speed_cmd rm_ros_interfaces/msg/Liftspeed "speed: 100"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.

	ros2 topic echo /rm_driver/set_lift_height_result

3.13.3 Get the lifting mechanism state

Function description	Get the lifting mechanism state
Parameter description	Liftstate.msg int16 height: current height. int16 current: current current. uint16 err_flag: drive error code.
Command example	ros2 topic pub /rm_driver/get_lift_state_cmd std_msgs/msg/Empty "{}"
Return 回value	Successful return: current state of the lifting mechanism; Failure return: the driver terminal returns an error code.
	ros2 topic echo /rm_driver/get_lift_state_result

3.14 Functions related to the transmissive force-position compensation mode

For the RealMan robotic arm with one-axis force and six-axis force versions, the user can not only directly use the teaching device to call the underlying force-position mixing control module but also combine the custom trajectory with the underlying force-position mixing control algorithm in the form of periodic transmission to compensate.

If force data calibration has not been completed before the force operations, the zero position can be calibrated using the one-axis force and six-axis force data clear interfaces.

3.14.1 starting the transmissive force-position mixing control compensation mode

Function description	starting the transmissive force-position mixing control compensation mode
Parameter description	std_msgs::msg::Empty
Command example	ros2 topic pub /rm_driver/start_force_position_move_cmd std_msgs/msg/Empty "{}"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.

	ros2 topic echo /rm_driver/start_force_position_move_result

3.14.2 Stopping the transmissive force-position mixing control compensation mode

Function description	Stopping the transmissive force-position mixing control compensation mode
Parameter description	std_msgs::msg::Empty
Command example	ros2 topic pub /rm_driver/stop_force_position_move_cmd std_msgs/msg/Empty "{}"
Return 回value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/stop_force_position_move_result

3.14.3 Transmissive force-position mixing control compensation (joint)

Function description	Transmissive force-position mixing control compensation (joint)
Parameter description	Forcepositionmovejoint.msg float32[6] joint: target joint radian uint8 sensor: type of sensor used, 0 - one-axis force, 1 - six-axis force uint8 mode: mode, 0 - along the base coordinate system, 1 - along the tool coordinate system int16 dir: force control direction, 0-5 represent X/Y/Z/Rx/Ry/Rz respectively, where the default direction is Z direction for one-axis force type float32 force: force value, unit: 0.1 N. bool follow: whether high follow, true: high follow, false: low follow. uint8 dof: degree of freedom of the robotic arm
Command example	It needs to be a large number (10 or more) of continuous position points, with more than 2ms period continuous release. ros2 topic pub /rm_driver/force_position_move_joint_cmd rm_ros_interfaces/msg/Forcepositionmovejoint " joint: [0, 0, 0, 0, 0, 0] sensor: 0 mode: 0 dir: 0 force: 0.0 follow: false dof: 6

Return ⊡value	Success: no return; Failure return: false, and the driver terminal returns an error code.

3.14.4 Transmissive force-position mixing control compensation (pose)

Function description	Transmissive force-position mixing control compensation (pose)
Parameter description	Forcepositionmovepose.msg geometry_msgs/Pose pose: target pose, x, y, z coordinates (float type, unit: m) + quaternion. uint8 sensor: type of sensor used, 0 - one-axis force, 1 - six-axis force uint8 mode: mode, 0 - along the base coordinate system, 1 - along the tool coordinate system int16 dir: force control direction, 0-5 represent X/Y/Z/Rx/Ry/Rz respectively, where the default direction is Z direction for one-axis force type float32 force: force value, unit: 0.1 N. bool follow: whether high follow, true: high follow, false: low follow.

Command example	It needs to be a large number (10 or more) of continuous position points, with more than 2ms period continuous release.
	ros2 topic pub /rm_driver/force_position_move_pose_cmd rm_ros_interfaces/msg/Forcepositionmovepose "pose:
	  position:
	    x: 0.0
	    y: 0.0
	    z: 0.0
	  orientation:
	    x: 0.0
	    y: 0.0
	    z: 0.0
	    w: 1.0
	sensor: 0
	mode: 0
	dir: 0
	force: 0
	follow: false"

Return ⊡value	Success: no return; Failure return: false, and the driver terminal returns

	an error code.

3.15 Robotic arm state active reporting

3.15.1 Setting UDP robotic arm state active reporting configuration

Function description	Set UDP robotic arm state active reporting configuration
Parameter description	Setrealtimepush.msg uint16 cycle: set the broadcast cycle, which is a multiple of 5ms (default 1 i.e. 1 * 5 = 5 ms, 200 Hz). uint16 port: set the broadcast port number (default 8089). uint16 force_coordinate: set the coordinate system of force data outside the system (only supported by the arm with force sensors). string ip: set the custom reporting target IP address (default 192.168.1.10).
Command example	ros2 topic pub --once /rm_driver/set_realtime_push_cmd rm_ros_interfaces/msg/Setrealtimepush "cycle: 1 port: 8089 force_coordinate: 0 ip: '192.168.1.10'"
Return □value	Successful return: true; failure returns: false, the driver terminal returns an error code.
	ros2 topic echo /rm_driver/set_realtime_push_result

3.15.2 Getting UDP robotic arm state active reporting configuration

Function description	Get UDP robotic arm state active reporting configuration
Parameter description	Setrealtimepush.msg uint16 cycle: set the broadcast cycle, which is a multiple of 5ms (default 1 i.e. 1 * 5 = 5 ms, 200 Hz). uint16 port: set the broadcast port number (default 8089). uint16 force_coordinate: set the coordinate system of force data outside the system (only supported by the arm with force sensors). string ip: set the custom reporting target IP address (default 192.168.1.10).
Command example	ros2 topic pub --once /rm_driver/get_realtime_push_cmd std_msgs/msg/Empty "{}"

Return □value	Successfully set information; Failure return: the driver terminal returns an error code.
	ros2 topic echo /rm_driver/get_realtime_push_result

3.15.3 UDP robotic arm state active reporting

Six-axis force

Function description	Six-axis force
Parameter description	Sixforce.msg float32 force_fx: the force along the x-axis direction. float32 force_fy: the force along the y-axis direction. float32 force_fz: the force along the z-axis direction. float32 force_mx: the force when rotating along the x-axis direction. float32 force_my: the force when rotating along the y-axis direction. float32 force_mz: the force when rotating along the z-axis direction.
Subscription command	ros2 topic echo /rm_driver/udp_six_force

One-axis force

Function description	One-axis force
Parameter description	Sixforce.msg float32 force_fx: the force along the x-axis direction. float32 force_fy: the force along the y-axis direction. float32 force_fz: the force along the z-axis direction. (only this value is valid) float32 force_mx: the force when rotating along the x-axis direction. float32 force_my: the force when rotating along the y-axis direction. float32 force_mz: the force when rotating along the z-axis direction.
Subscription command	ros2 topic echo /rm_driver/udp_one_force

Robotic arm error

Function description	Robotic arm error
Parameter description	std_msgs::msg::UInt16 uint16 data: the robotic arm error message.
Subscription	ros2 topic echo /rm_driver/udp_arm_err

command	

System error

Function description	System error
Parameter description	std_msgs::msg::UInt16 uint16 data: the system error message.
Subscription command	ros2 topic echo /rm_driver/udp_sys_err

Joint error

Function description	Joint error
Parameter description	Jointerrorcode.msg uint16[] joint_error: the error message for each joint. uint8 dof: the arm degree of freedom message.
Subscription command	ros2 topic echo /rm_driver/udp_joint_error_code

The robot arm radians data

Function description	The robot arm radians data
Parameter description	sensor_msgs::msg::JointState builtin_interfaces/Time stamp int32 sec: time message, unit: second. uint32 nanosec: time message, unit: nanosecond. string frame_id: coordinate system name. string[] name: joint name. float64[] position: joint radian message. float64[] velocity: joint speed message. (not used yet) float64[] effort: joint force message. (not used yet)
Subscription command	ros2 topic echo /joint_states

Pose information

Function description	Pose information
Parameter description	geometry_msgs::msg::Pose Point position: the robotic arm current coordinate information.

	float64 x float64 y float64 z Quaternion orientation: the robotic arm current pose information. float64 x 0 float64 y 0 float64 z 0 float64 w 1
Subscription command	ros2 topic echo /rm_driver/udp_arm_position

Current external force data of the six-axis force sensor system

Function description	Current external force data of the six-axis force sensor system
Parameter description	Sixforce.msg float32 force_fx: the force forced on the current sensor along the x-axis direction. float32 force_fy: the force forced on the current sensor along the y-axis direction. float32 force_fz: the force forced on the current sensor along the z-axis direction. float32 force_mx: the force forced on the current sensor when rotating along the x-axis direction. float32 force_my: the force forced on the current sensor when rotating along the y-axis direction. float32 force_mz: the force forced on the current sensor when rotating along the z-axis direction.
Subscription command	ros2 topic echo /rm_driver/udp_six_zero_force

Current external force data of the one-axis force sensor system

Function description	Current external force data of the one-axis force sensor system
Parameter description	Sixforce.msg float32 force_fx: the force forced on the current sensor along the x-axis direction. float32 force_fy: the force forced on the current sensor along the y-axis direction. float32 force_fz: the force forced on the current sensor along the z-axis

	direction. (only this data is valid) float32 force_mx: the force forced on the current sensor when rotating along the x-axis direction. float32 force_my: the force forced on the current sensor when rotating along the y-axis direction. float32 force_mz: the force forced on the current sensor when rotating along the z-axis direction.
Subscription command	ros2 topic echo /rm_driver/udp_one_zero_force

Reference coordinate system for external force data of the system

Function description	Reference coordinate system for external force data of the system
Parameter description	std_msgs::msg::UInt16 uint16 data: coordinate system for external force data of the system, where 0 is the sensor coordinate system, 1 is the current work coordinate system, and 2 is the current tool coordinate system This data affects the reference coordinate system for external force data of one-axis and six axis force sensor systems
Subscription command	ros2 topic echo /rm_driver/udp_arm_coordinate