JAKA Robotics

Just Always Keep Amazing

SDK V2.2.7 Release Notes

Table of Contents

1. Release Information

1.1 Release Date

04/2025

1.2 Version Information

This SDK version 2.2.2 is compatible with Linux 64/32 bit and Windows x86_64/x86 systems.

Other detailed information is as follows:

Name	Version	Note
Controller	1_7_2_28 and above	X64 / X32
SCB	03_11_R	/
PSCB	03_17_PR	/
Zu, C, Pro Series servo	R2214_ZU_C_PRO	/
MiniCobo Series servo	R3214_MINICOBO	/

Note:

The compatibility between SDK V2.2.7 and the controller version should be given attention.

● SDK V2.2.7 should be used on controllers of version 1_7_2_28 and above.

● For controllers of versions 1_7_0_x and 1_5_x, it is recommended to use SDK V2.1.11 and its previous versions.

For detailed information, please refer to SDK V2.2.7 user manual.

1.3 Version Features Summary

● In terms of compatibility of this version, the interfaces are basically compatible with the old versions. Some interfaces with compatibility issues will also be explained in the user manual.

● This release adds the EDG (External Data Guider) function, which can be used to obtain status data and issue control commands at high speed, ensuring certain real-time requirements. For specific usage requirements, please refer to the additional information section at the end.

● This version supports 64-bit and 32-bit client systems and supports arm architecture.

For detailed information on SDK V2.2.7, please refer to SDK user manuals.

2. Version Features

2.1 New features

2.1.1 Added EDG Function

1) Users can obtain status data at high speed through EDG, which is suitable for scenarios with high real-time requirements. The supported data are as follows:
   a) Force control sensor related data.
   b) Basic status data: joint/Cartesian position, joint speed, control cabinet, Modbus IO and joint current.
2) Servo motion can be sent through the EDG interface to replace the original servo function.

2.1.2 Added Motion Planner Function

Customer can set speed priority or smoothness priority, corresponding to T planning and S planning in the App respectively.

2.1.3 Added DO/AO Setting During Motion

It can be used in scenarios with high beat requirements. The output of the specified position can be triggered during motions.

2.1.4 Added Modification of Drag Feel Parameters

It can be used to directly modify the drag feel during dragging mode. Be sure to read the range requirements in the manual carefully before modifying them to avoid accidents.

2.1.5 Modification of User and Tool Coordinate Systems

Old forward and inverse kinematics interfaces can only use the default user coordinate system and tool coordinate system parameters for calculation. This new version allows users to use the specified user and tool coordinate systems.

2.1.6 Refresh Semaphore Information

Add an interface to refresh the semaphore information. In previous versions, there was a delay in obtaining semaphore information or the phenomenon of occasional 0. Calling this interface before obtaining semaphore information can solve this problem.

2.2 Function Optimization

2.2.1 Documentation

The TCP and SDK user manuals were checked and improved, and they are online documents now. Users can visit JAKA's documentation website to read.

The user manual of C++ language documents is explained in details, the links of which are written in the user manual of other C-related languages for references.

2.2.2 Bug Fixes

Fixed the issues in the previous SDK versions, including:

1) The circular motion circle number is not effective.

2) The C# controller cannot control two robotic arms, and the C# compilation fails.

3) The TCP document part description error is revised.

2.3 Additional Materials

This section details the precautions for using the EDG function.

2.3.1 Environmental Requirements

Controller versions: 1_7_2_28 and above

SDK versions: 2.2.7 and above

Operating system: Linux-like system (non-virtual machine, preferably with real-time patch configured)

2.3.2 System Requirements

EDG servo motion is a function that requires high real-time performance of the client, requiring the client to transmit data to the server within a specified interval.

There are several points to note about real-time performance:

4) The client should try to ensure that the sending cycle is 8ms.

5) Try to bind the EDG thread to a fixed core of the CPU to avoid switching and consuming resources, and try to increase its priority.

6) Pay attention to the CPU operating frequency of the EDG thread, try to fix its frequency, and avoid CPU frequency reduction due to factors such as temperature.

2.3.3 Notes on Interface Usage

1) edg_servo_j/p cannot be used at the same time as the original servo_j/p.

2) The edg_init call must be before servo_move_enable, and no delay is required after servo_move_enable.

3) Users need to enable EDG initialization to use other EDG functions. When edg_servo_j/p ends, users need to turn off the EDG initialization interface. If the EDG initialization is not turned off, calling the original servo_j/p interface will fail.

4) If the SDK client edg_init initializes and passes the client's IP address, the UDP Server will transmit the feedback data to the 10010 Port of the corresponding IP in unicast mode; otherwise, it will be broadcast to the 10010 Port of all robots in the LAN.

5) Since the joints in servo mode run completely according to user instructions, user instructions must be smooth, the speed must be continuous, and the acceleration must be within the robot joint limit.