

## **Jaka\_driver\_interface**

Version: V2.2  
Date: 2023.8.8

## Note:

The definition of collaborative robots follows international ISO standards and national standards to protect the safety of operators. We do not recommend directly applying the robot arm to circumstances where the object is a human body. However, when robot users or application developers do need to involve the human body in the robot operation, they should configure a safe, reliable, fully tested, and certified safety protection system for the robot arm to protect personnel safety on the premise that users or application developers can fully evaluate personnel safety.

The contents contained in Jaka\_driver\_interface Instruction Manual are the exclusive property of JAKA Robotics Co., Ltd. (Hereinafter collectively referred to as JAKA), and shall not be used in any form without the written consent of JAKA.

Jaka\_driver\_interface Instruction Manual is subject to revision and improvement on a regular basis by JAKA and its contents are subject to change without notice. Please check the actual product information carefully before using this manual.

The information contained in Jaka\_driver\_interface Instruction Manual is not a commitment of JAKA, and JAKA is not responsible for any errors that may occur in this Manual and any accidental or consequential damages caused by the use of this Manual and the products described therein. Please read this Manual carefully before installing and using the product.

The pictures in this Manual are for reference only, please refer to the actual product.

If the robot arm is modified or disassembled, JAKA will not be responsible for after-sales service.

JAKA also reminds the user that safety equipment must be used and safety provisions must be observed when using and maintaining JAKA robots.

Programmers of the JAKA robot and designers and debuggers of the robot system shall be familiar with the programming mode and system application installation of JAKA robots.

## Manual Instructions

This manual mainly contains an explanation and description of the Jaka\_driver\_interface.

This manual is intended for users with certain basic development skills who have received basic training in robot usage to facilitate the usage and development of JAKA\_ROS.

### Read more

For more information about our products, please scan the QR code on the right to visit our official website [www.jaka.com](http://www.jaka.com).



## Jaka\_driver\_interface

### 1. Commands for robot motion control

#### 1.1 Linear motion (in the custom user coordinate system)

**Description:** Linear motion of the robot in the user coordinate system.

**Service Name:** /jaka\_driver/linear\_move

The **jaka\_msgs/Move.srv** data format is as follows:

```
float32[] pose
bool has_ref
float32[] ref_joint
float32 mvvelo
float32 mvacc
float32 mvtime
float32 mvradii
int16 coord_mode
int16 index
---
int16 ret
string message
```

#### Meaning of the data format:

- pose: represents the target movement waypoint. pose[0], pose[1], and pose[2] represent the spatial position information in the user's coordinate system(unit: mm). pose[3], pose[4], and pose[5] represent the pos information, described in the form of rotation vector (unit: radian).
- has\_ref: represents the presence or absence of a reference solution, unavailable yet.
- ref\_joint: represents reference solution, unavailable yet.
- mvvelo: represents the speed of movement (unit: mm/s).
- mvacc: represents the acceleration of motion (unit: mm/s<sup>2</sup>).
- mvtime: represents movement time, unavailable yet.
- mvradii: represents the radius of motion,unavailable yet.
- coord\_mode: represents the coordinate system, temporarily using.
- index: represents the direction of motion number, unavailable yet.
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

#### 1.2 Joint motion

**Description:** Axis motion of the robot in the joint coordinate system.

**Service Name:** /jaka\_driver/joint\_move

The **jaka\_msgs/Move.srv** data format is as follows:

```
float32[] pose
bool has_ref
float32[] ref_joint
float32 mvvelo
float32 mvacc
float32 mvtime
float32 mvradii
int16 coord_mode
int16 index
---
int16 ret
string message
```

**Meaning of the data format:**

- pose: pose[0]~pose[5] represents the axis position information in the joint coordinate system(unit: radian).
- has\_ref: represents the presence or absence of a reference solution, unavailable yet.
- ref\_joint: represents reference solution, unavailable yet.
- mvvelo: represents the speed of movement (unit: radian/s).
- mvacc: represents the acceleration of motion (unit: radian/s<sup>2</sup>).
- mvtime: represents movement time, unavailable yet.
- mvradii: represents the radius of motion, unavailable yet.
- coord\_mode: represents the coordinate system, temporarily using.
- index: represents the direction of motion number, unavailable yet.
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

### 1.3 Jog motion

**Description:** Continuous motion of the robot in the joint coordinate system, user coordinate system, or tool coordinate system.

**Service Name:** /jaka\_driver/jog

The **jaka\_msgs/Move.srv** data format is as follows:

float32[] pose  
 bool has\_ref  
 float32[] ref\_joint  
 float32 mvvelo  
 float32 mvacc  
 float32 mvtime  
 float32 mvradii  
 int16 coord\_mode  
 int16 index  
 ---  
 int16 ret  
 string message

#### Meaning of the data format:

- pose: represents the target movement waypoint, unavailable yet.
- has\_ref: represents the presence or absence of a reference solution, unavailable yet.
- ref\_joint: represents reference solution, unavailable yet.
- mvvelo: represents motion speed, joint motion (unit: rad/s), linear motion (unit mm/s)
- mvacc: represents motion acceleration, unavailable yet.
- mvtime: represents movement time, unavailable yet.
- mvradii: represents the radius of motion, unavailable yet.
- coord\_mode: represents the coordinate system, joint coordinate system = 0, base coordinate system = 1, tool coordinate system = 2.

• index: represents the direction of motion number and is entered in the range [0,11],  
 In the case of the joint coordinate system, [0,11] represents J1+, J1-, J2+, J2-, J3+, J3-, J4+, J4-, J5+, J5-, J6+, J6- in order.

If it is other coordinate system, [0,11] stands for: X+, X-, Y+, Y-, Z+, Z-, RX+, RX-, RY+, RY-, RZ+, RZ- in order.

- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

**Note:** If the jog service is not called again within 3 seconds after it is started, the ROS driver package will automatically stop the currently executing jog motion.

## 1.4 Servo position control mode enable

**Service Name:** /jaka\_driver/servo\_move\_enable

The **jaka\_msgs/ ServoMoveEnable.srv** data format is as follows:

**bool enable**  
 ---

**int16 ret**

**string message**

**Meaning of the data format:**

- enable: means whether to turn on the servo mode to enable, true means to enter the servo position control mode, false means to exit the mode.
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

## 1.5 Cartesian space servo mode motion

**Service Name:** /jaka\_driver/ servo\_p

The **jaka\_msgs/ ServoMove.srv** data format is as follows:

**bool pose**

---

**int16 ret**

**string message**

**Meaning of the data format:**

- pose: represents the target motion waypoint, pose[0], pose[1], pose[2] represent the spatial position information under the user coordinate system in meters; pose[3], pose[4], pose[5] represents the pose information, described in the form of RPY (unit radian).
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

## 1.6 Joint space servo mode motion

**Service Name:** /jaka\_driver/ servo\_j

The **jaka\_msgs/ ServoMove.srv** data format is as follows:

**bool pose**

---

**int16 ret**

**string message**

**Meaning of the data format:**

- pose: represents the target motion waypoint, pose[0]~pose[5] represents the axis position information in the joint coordinate system(unit: radian).
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

## 1.7 Stop motion

**Description:** Stop robot movement, and the robot movement status will change to "Stop".

**Service Name:** /jaka\_driver/stop\_move

The `std_srv/Empty.srv` data format is as follows:

---

**Meaning of the data format:**

- The service parameter is empty, that is, the interface can be called once to send a stop command.

## 2. Robot parameter setting commands

### 2.1 Set TCP parameters

**Description:** Set the tool position and orientation data relative to the flange coordinate system. **Service Name:** /jaka\_driver/set\_toolframe

The `jaka_msgs/SetTcpFrame.srv` data format is as follows:

`float32[] pose`

`int16 tool_num`

---

`int16 ret`

`string message`

**Meaning of the data format:**

- pose: represents the TCP data, pose[0], pose[1], pose[2] represent the spatial position information in mm in the end flange coordinate system; pose[3], pose[4], pose[5] represent the pose information, described in the form of rotation vector (unit: radian).
- tool\_num: represents the tool coordinate system number, range [1,11].
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

### 2.2 Set user coordinate system parameters

**Description:** Set the user coordinate system data relative to the base coordinate system.

**Service Name:** /jaka\_driver/set\_userframe

The `jaka_msgs/SetTcpFrame.srv` data format is as follows:

`float32[] pose`

`int16 user_num`

---

**int16 ret**

**string message**

**Meaning of the data format:**

- pose: represents the user coordinate system data, pose[0], pose[1], pose[2] represent the spatial position information in the base coordinate system (unit: mm), pose[3], pose[4], pose[5] represent the pose information, described in the form of rotation vector (unit: radian).
- tool\_num: represents the tool coordinate system number, range [1,10].
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

## 2.3 Set the robotic arm payload centroid parameter

**Service Name:** /jaka\_driver/set\_payload

The **jaka\_msgs/SetPayload.srv** data format is as follows:

**int16 tool\_num**

**float32 masss**

**float32 xc**

**float32 yc**

**float32 zc**

---

**int16 ret**

**string message**

**Meaning of the data format:**

- tool\_num: represents the tool coordinate system number.
- mass: represents load mass (unit: kg).
- xc, yc, zc: represents the position of the centroid (unit: mm).
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

## 2.4 Set the free drive mode

**Service Name:** /jaka\_driver/drag\_mode

The **std\_srv/SetBool.srv** data format is as follows:

**bool data**

---

**bool success**

**string message**

**Meaning of the data format:**

- data: represents the free drive state, 0 means off, 1 means on.



- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

## 2.5 Set collision sensitivity

**Description:** Turn on the collision detection function and set the collision sensitivity parameters.

**Service Name:** /jaka\_driver/set\_collisionlevel

The **jaka\_msgs/SetCollision.srv** data format is as follows:

bool is\_enable

int16 value

---

int16 ret

string message

**Meaning of the data format:**

- is\_enable: represents turning on and off the collision detection function.
- value: represents setting the collision sensitivity, range [10,100].
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

## 2.6 Set IO

**Service Name:** /jaka\_driver/set\_io

The **jaka\_msgs/SetIO.srv** data format is as follows:

string signal

int16 type

int16 index

float32 value

---

int16 ret

string message

**Meaning of the data format:**

- signal: represents the type of signal quantity: digital or analog.
- type: represents the type of IO, control panel IO = 0, tool IO = 1, expansion IO = 2.
- index: represents the IO index value.
- value: represents the value of the input IO, if it is a bool type, non-0 all default to 1.
- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

## 2.7 Get IO

**Service Name:** /jaka\_driver/get\_io

The **jaka\_msgs/GetIO.srv** data format is as follows:

string signal

int16 path

int16 type

int16 index

---

float32 value

string message

**Meaning of the data format:**

- signal: represents the type of signal quantity: digital or analog.
- path: represents whether the IO is of input or output type, with input type = 0 and output type = 1.
- type: represents the type of IO, control panel IO = 0, tool IO = 1, expansion IO = 2.
- index: represents the IO index value.
- value: represents the value of the get IO.
- message: represents the return value, indicating the result of the call.

## 2.8 Clear alarm messages

**Note: This interface is not yet available in the driver package.**

**Description:** Clears all alarm messages from the robot and restores the robot to a ready-to-run state. **Service Name:** /jaka\_driver/clear\_err

The **jaka\_msgs/ClearErr.srv** data format is as follows:

---

int16 ret

string message

**Meaning of the data format:**

- ret: represents the return value, indicating success or failure.
- message: represents the return value, indicating the result of the call.

## 3. Robot status reporting command

### 3.1 End position pose status information reporting

**Description:** Active reporting of end position pose information at a specific frequency.

**Topic Name:** /jaka\_driver/tool\_position

The `geometry_msgs/TwistStamped.msg` data format is as follows:

```
std_msgs/Header header
uint32 seq
time stamp
string frame_id
geometry_msgs/Twist twist
geometry_msgs/Vector3 linear
float64 x
float64 y
float64 z
geometry_msgs/Vector3 angular
float64 x
float64 y
float64 z
```

**Meaning of the data format:**

- linear represents the 3D position information (unit: m).
- angular represents the 3D pose information (unit: rotation vector).

### 3.2 Joint status information reporting

**Description:** Active reporting of joint position information at a specific frequency.

**Topic Name:** `/jaka_driver/joint_position`

The `sensor_msgs/JointState.msg` data format is as follows:

```
std_msgs/Header header
uint32 seq
time stamp
string frame_id
string[] name
float64[] position
float64[] velocity
float64[] effort
```

**Meaning of the data format:**

- the position represents joint position information (unit: radian).

### 3.3 Robot status information reporting

**Description:** Spontaneous reporting of event status information at a specific frequency.

**Topic Name:** `/jaka_driver/robot_states`

The `jaka_msgs/RobotMsg.ms` data format is as follows:

```
int16 motion_state
int16 power_state
int16 servo_state
int16 collision_statestate
```

**Meaning of the data format:**

- motion\_state represents the robot motion state, Stop = 0, Pause = 1, EmeStop = 2, Running = 3, Error = 4.
- power\_state represents the robot power on state, Power on = 1, Power off = 0.
- servo\_state represents servo mode enable state, Enable = 1, Not enable = 0.
- collision\_state represents collision detection state, Collision alarm = 1, No collision = 0.