JK-RDQK-UM-028

JAKA SECOND DEVELOPMENT FAQ

Version: V1.0.0

Date: 2020.07.01

Version Record

Version No.	Version Date	Description
V1.0.0	2020.7.8	1.Create document
V1.0.1	2020.12.12	1.Add log query description
V1.0.2	2021.5.25	

Content

1. Network Connection - 4 -

2. Using Python jkrc Module - 4 -

2.1 No module named ‘jkrc’ - 4 -

2.2 ImportError: DLL load failed while importing jkrc - 5 -

3. Using SDK Interface - 5 -

4. Version Query Method For Dynamic Library - 7 -

5. Get Latest Information - 7 -

6. Developed using SDK under ROS - 8 -

7. Supported Platforms, Programming language - 8 -

8. Compiler Version - 8 -

9. Query Whether The Current Controller Version Supports Interface In The SDK - 8 -

10. Secondary Development Using Other Programming Languages - 9 -

11. About JAKA TCP Protocol - 9 -

11.1 Sending motion command continuously - 9 -

11.2 Stop robot motion - 9 -

11.3 Commands which will no longer be maintained - 10 -

12. Log Query - 10 -

13. Error Code and Error Information Query - 10 -

14. Recommended Reading Material - 10 -

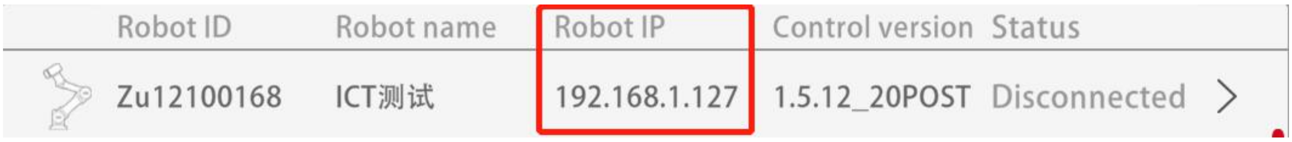
15. Feedback and Errata - 11 -

Property Description - 11 -

1. Network Connection

Some customers may have problems about connecting network when using SDK for the first time:

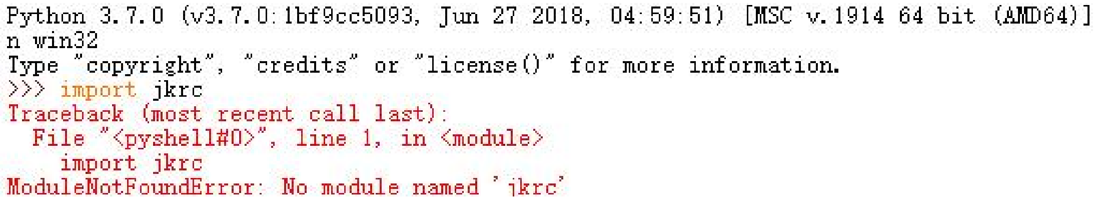
Q: Where to find the robot IP?
A: Use JAKA APP to connect to the network where the robotic arm is located, and then query robot IP through JAKA APP. Or query robot IP in current network through SDK function interface **get_controller_ip().**



Robot ID	Robot name	Robot IP	Control version	Status
Zu12100168	ICT测试	192.168.1.127	1.5.12_20POST	Disconnected

2. Using Python jkrc Module

2.1 No module named 'jkrc'



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)]
n win32
Type "copyright", "credits" or "license()" for more information.
>>> import jkrc
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    import jkrc
ModuleNotFoundError: No module named 'jkrc'
```

This problem is mainly because the Python interpreter cannot find the jkrc module.
You can use following sentences to add the path of jkrc module into Python interpreter:

Windows

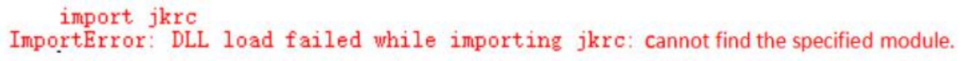
```
import sys
print(sys.path)
sys.path.insert(0, r'F:\xx\xx\xx\xx')  # Ensure searching the path firstly, and the
 user can replace it as needed
```

Linux

```
import sys
import time
```

```
sys.path.insert(0, 'home/jakauser/Downloads') #Ensure searching the path firstly, and the user can replace it as needed
```

2.2 ImportError: DLL load failed while importing jkrc



```
    import jkrc
ImportError: DLL load failed while importing jkrc: cannot find the specified module.
```

The main reason for this problem is that the Python version used is incorrect, just choose the Python version recommended by JAKA.

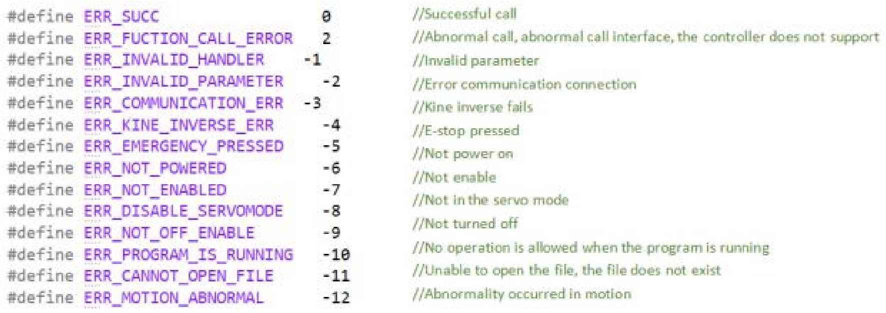
Python2.7 and Python3.5 are supported under Linux.

Python3.7 is supported under Windows，it is recommended to install a version greater than 3.7.0.

3. Using SDK Interface

1.Q: What is the meaning of -2 that is returned when a command call fails?

A: -1 is the return value of commands, returning any non-zero value means that this command was not called successfully, the following is the definition of return value, you can also find all definitions in “C header file/jkerr.h”.



```
#define ERR_SUCC                  0      //Successful call
#define ERR_FUCTION_CALL_ERROR    2      //Abnormal call, abnormal call interface, the controller does not support
#define ERR_INVALID_HANDLER     -1       //Invalid parameter
#define ERR_INVALID_PARAMETER     -2     //Error communication connection
#define ERR_COMMUNICATION_ERR   -3       //Kine inverse fails
#define ERR_KINE_INVERSE_ERR      -4     //E-stop pressed
#define ERR_EMERGENCY_PRESSED     -5     //Not power on
#define ERR_NOT_POWERED           -6     //Not enable
#define ERR_NOT_ENABLED           -7     //Not in the servo mode
#define ERR_DISABLE_SERVOMODE     -8     //Not turned off
#define ERR_NOT_OFF_ENABLE        -9     //No operation is allowed when the program is running
#define ERR_PROGRAM_IS_RUNNING    -10    //Unable to open the file, the file does not exist
#define ERR_CANNOT_OPEN_FILE      -11    //Abnormality occurred in motion
#define ERR_MOTION_ABNORMAL       -12
```

2.Q: How to monitor the abnormality and error message of the robot?

1) Monitoring robot status using get_robot_status

2) Getting the last error code using get_last_error

3.Q: What is the difference between blocking interface and non-blocking interface?

1) Blocking interface will get return value at the end of command execution; non-blocking interface will get return value immediately when calling command, without waiting for the completion of the command.

2) Blocking commands will block program at the current command and cannot continue to run until the end of execution, which may freeze the program and interface.

4. Why dose the power-on command freeze for a long time?

A: The power-on and enable commands are blocking commands by default, which may take more than ten seconds.

5. How to get the error message of the robot?

1) set_error_handler, set callback function, you can execute codes defined by yourself when there is an error in calling function.

2) get_last_error, get the last error code of the robot.

6. The linear motion command cannot be executed successfully.

1) There are unreachable points(or singular points) of the robot in the linear motion from the current position to the target position.

2) Wrong movement mode. The robot needs to move to the given coordinates, but the user misuses incremental movement mode.

3) Slow velocity, the unit of velocity is mm/s.

4) The acceleration is too slow, which causes the movement to take a long time to accelerate. The unit of acceleration is mm/s2.

7. How to realize the motion in the tool coordinate system?

A: Set the user coordinate system in the world coordinate system to obtain the current tcp Cartesian coordinates.

Set this coordinate to the user coordinate system and switch to this coordinate system to realize the robot motion in the tool coordinate system

8. How to interrupt a motion command that is being executed?

Terminate a motion command that is currently being executed through calling motion_abort().

9. How to get the current and voltage of the robot and the average power consumption?

Get current and voltage of the robot and other information in RobotMonitorData through calling get_robot_status interface.

10. How to run program that has been saved?

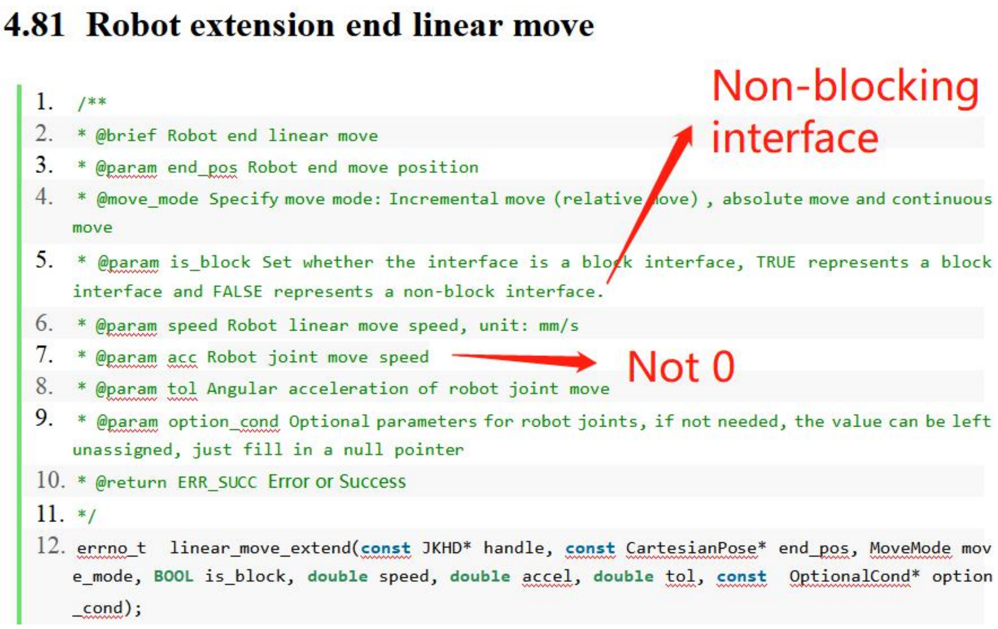
Call the program_load interface firstly, and then call the program_run interface.

11. The program crashes when calling the SDK function, but it can run normally when running the sample document in the material package.

It is likely that there is a problem with the dynamic library connection, so the user can refer to the "Dynamic Library Usage Method" in the secondary development related to the knowledge library.

12. How to add arc transition for linear motion? (Only arc transition in linear motion is supported currently)

Take C# for example. Use the robot extended end linear motion command, set the command to non-blocking, and the end point error of the joint motion is not 0. At the same time, it should be ensured that there are no other commands between the two linear motion commands that require circular arc transition, otherwise the circular arc transition will not take effect.



4.81 Robot extension end linear move

```
/**
 * @brief Robot end linear move
 * @param end_pos Robot end move position
 * @move_mode Specify move mode: Incremental move (relative move) , absolute move and continuous move
 * @param is_block Set whether the interface is a block interface, TRUE represents a block interface and FALSE represents a non-block interface.
 * @param speed Robot linear move speed, unit: mm/s
 * @param acc Robot joint move speed
 * @param tol Angular acceleration of robot joint move
 * @param option_cond Optional parameters for robot joints, if not needed, the value can be left unassigned, just fill in a null pointer
 * @return ERR_SUCC Error or Success
 */
errno_t  linear_move_extend(const JKHD* handle, const CartesianPose* end_pos, MoveMode move_mode, BOOL is_block, double speed, double accel, double tol, const  OptionalCond* option_cond);
```

Non-blocking interface

Not 0

4. Version Query Method For Dynamic Library

Right-click dll document in windows, select properties, and you can query version information in "Detailed information".

Linux Input **strings libjakaAPI.so | grep jakaAPI_version** to query version information in Linux.

5. Get Latest Information

The latest information can be obtained from the official website www.jaka.com, or contact relevant service personnel.

6. Developed using SDK under ROS

Developed under ROS, the robot is controlled through the function interface provided by the JAKA SDK in fact. It is not different from ordinary second development. The simulation models of different types of robots are provided in the folder of **JAKA secondary development material package\JAKA simulation models.** ROS is an open source software, JAKA does not provide related technical support.

7. Supported Platforms, Programming language

Supported platforms: Windows x86/x64 、Linux x86/64 。
Supported languages: C/C++/C# /Python

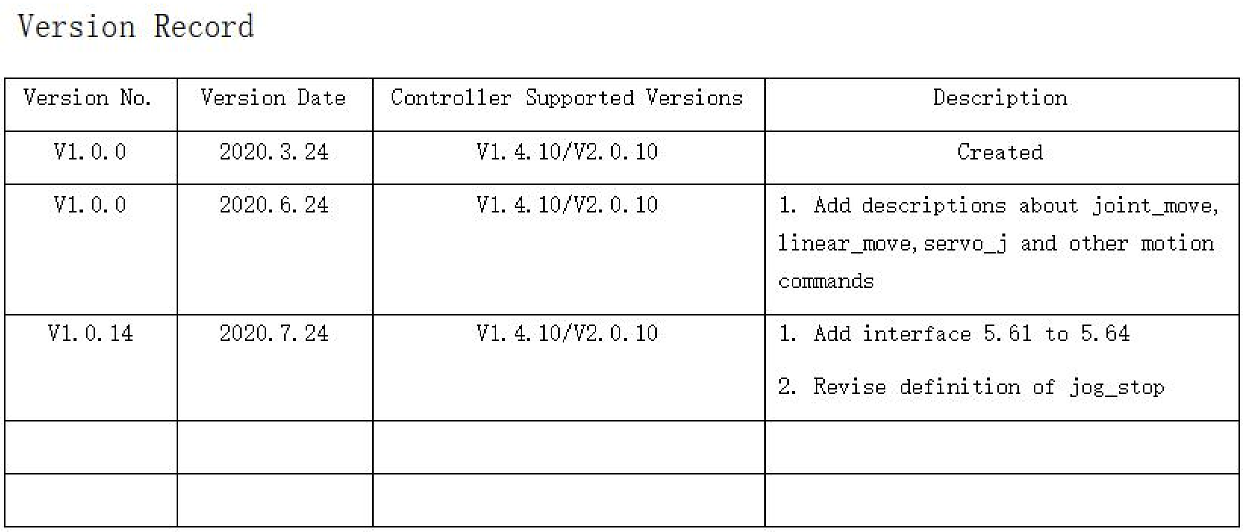
8. Compiler Version

Windows dynamic library compiler is MSVC-120，and the minimum supported compiler version is MSVC-120

Linux dynamic library compiler is gcc version，and the minimum supported compiler version is 5.4.0。

9. Query Whether The Current Controller Version Supports Interface In The SDK

You can know the correspondence between the current SDK version and the controller version by querying version record of the SDK document. For example, open the Python secondary development interface document, you can query the version record, and the version No. represents the lowest version supported by the corresponding function.



Version Record

Version No.	Version Date	Controller Supported Versions	Description
V1.0.0	2020.3.24	V1.4.10/V2.0.10	Created
V1.0.0	2020.6.24	V1.4.10/V2.0.10	1. Add descriptions about joint_move, linear_move, servo_j and other motion commands
V1.0.14	2020.7.24	V1.4.10/V2.0.10	1. Add interface 5.61 to 5.64 2. Revise definition of jog_stop

10. Secondary Development Using Other Programming Languages

Other programming languages here refer to languages other than **C/C++/C#/Python which are supported by JAKA SDK,** such as **JAVA**， **Matlab**, etc.

If the user needs other programming languages for secondary development, look up commands from **JAKA secondary development material package\JAKA_TCP material package\JAKA TCP interface for secondary development.pdf.** Pay attention that JAKA secondary development provides two ports, 10001 is used to sending control commands, 10000 is used to receiving data from robot. More details in **JAKA TCP interface for secondary development.pdf**。

11. About JAKA TCP Protocol

Note: skip this chapter if you do not use JAKA TCP protocol for secondary development

11.1 Sending motion command continuously

joint_move()、end_move() or move_L() are all blocking motion commands, next motion command will be executed after completing last motion command. It is suggested to stop current motion if you need to run next command, and then you can send next motion command.

11.2 Stop robot motion

stop_program can stop other motion commands of the robot, such as joint_move/moveL， it is suggested to use stop_program to stop robot motion.

11.3 Commands which will no longer be maintained

It is not recommended to use wait_complete command which is no longer maintained.

12. Log Query

When the user uses the SDK provided by JAKA for secondary development, the JAKA SDK will automatically generate a copy of jakalog.txt.

Users can check the corresponding error through this log, such as network connection error, operation sequence, etc.

Storage location under Windows: **C:\ or D:\** directory

Storage location under Linux: **/tmp** directory

13. Error Code and Error Information Query

Taking C++ as an example, there are two ways to get the error code:

1. Get the last error code using get_last_error, you can get corresponding error information from \JAKA secondary development material package2021\JAKA_SDK material package\Error code document\JAKA_ERROR_CODE.csv.
2. Monitoring the status of robot using get_robot_status, the status information obtained contains an error code, you can get corresponding error information from\JAKA secondary development material package2021\JAKA_SDK material package\Error code document\JAKA_ERROR_CODE.csv.

14. Recommended Reading Material

It is suggested to read the following materials if it is the first time for you to use JAKA robots or you have never used a robot before, so that you will be familiar with the basic usage of robots and you can do some simple programming and control using JAKA Zu APP:

Quick Start-Electrical Cabinet V2.1
JAKA Zu APP User Manual

15. Feedback and Errata

For any inaccurate descriptions or errors in the document, we would like to invite the readers to correct and criticize. In case of any questions during your reading process or any comments you want to make, please send an email to support@jaka.com, and our colleagues will try to reply one by one.

Property Description