Response Mode SDK API Description

1.IIMT Response Mode SDK API Description

SDK & API Instructions
It is the SDK package released with the update version and the c++ interface API provided. Users can carry out simple secondary development according to the interface instructions to control the robot's state and motion form PC side
What is SDK and API interface?
SDK(software development kit),It is a conllection of development tools for building application software,such as specific software packages,software architectures,hardware platforms,and operatiing systems,whic are used only by soft engineers.
API(application programming interface),It is the interface for user programming,it is the pre-defined functions which aim to provide applications and developers with the ability to access a set of routines based on a piece of software or hardware.For secondary developers,just need to focus on the functionality of interface,not the internal imploementation details.
What is response mode ?
The call to the interface returns a result, either true or false.
Communication method
The SDK interface and controller communicate through TCP/IP, and as a TCP server, the controller defaults to listening on port 29999.

2.SDK & API Sheet

serial number	Interface Name	Interface Function	Remark
1	API_HelloWorld	Test interface	
2	API_Login	Login, establish connection	
3	API_Logout	Log out, release connection	
4	API_SetDebugMode	Set whether to debug mode or not	
5	API_GetDebugMode	Get whether to debug mode or not	
6	API_GetErrorInfo	Get error message description	
7	API_PowerOn	Power on	If the robot has already been powered on and is powered on repeatedly, the interface returns a failure. Can be used in conjunction with pattern queries or power off interfaces.

serial number	Interface Name	Interface Function	Remark
8	API_ReleaseBrake	Release the brake	
9	API_PowerOff	Power off	
10	API_Play	Start	
11	API_Stop	Stop	
12	API_Pause	Pause	
13	API_LoadProgram	Load program	
14	API_LoadInstallation	Load installation configuration	
15	API_Shutdown	Shut down controller	
16	API_Quit	Close connection	
17	API_Running	Running status query	
18	API_IsProgramSaved	Query if the program has been saved	
19	API_IsInRemoteControl	Query if it is remote control	
20	API_Popup	Popup	
21	API_ClosePopup	Close the popup	
22	API_AddToLog	Add log messages to log history	
23	API_GetControllerVersion	Get version information of software installed on the robot	
24	API_GetVersion	Get version number	
25	API_GetRobotModel	Get robot model	
26	API_GetSerialNumber	Get serial number	
27	API_GetRobotMode	Get robot mode	
28	API_GetLoadedProgram	Get which program is loaded	
29	API_GetSafetyStatus	Get Safety status	
30	API_GetProgramState	Get program state	
31	API_GetOperationalMode	Get operational mode	
32	API_SetOperationalMode	Set operational mode	
33	API_ClearOperationalMode	Clear operational mode	
34	API_FreeDriveMode	Free drag mode	
35	API_EndFreeDriveMode	End free drag mode	
36	API_SetAutoRun	Set to automatic run mode	

serial number	Interface Name	Interface Function	Remark
37	API_MannulTeachStart	Mannul teach start	
38	API_API_ManualTeachEnd	Mannul teach end	
39	API_Servoj	Servo motion	Need to establish a separate connection, mainly for use on ROS2
40	API_Movej	Joint motion	
41	API_Movel	Linear motion	
42	API_GetDI	Get digital input status	
43	API_GetDO	Get digital output status	
44	API_GetAllDI	Get all digital input status at once	
45	API_GetAllDO	Get all digital output status at once	
46	API_SetDO	Set digital output status	
47	API_GetRobotStatus	Get robot status	
48	API_GetRobotJointRad	Get joint position radian	
49	API_GetRobotJointAngle	Get joint position angle	
50	API_GetRobotTcpPosition	Get TCP position	
51	API_GetArriveStatus	Get arrive status	
52	API_GetRunSpeed	Get running speed	
53	API_SetRunSpeed	Set running speed	
54	API_ExecuteScript	Remote script command execution	
55	API_GetVariableValue	Get variable value	
56	API_SetOffset	Set trajectory offset	
57	API_GetOffset	Get trajectory offset	
58	API_SendFile	Send files to the teaching pendant	
59	API_ExecuteFile	Remote execution of Lua script files	
60	API_GetCollisionLevel	Get collision level	
61	API_SetCollisionLevel	Set collision level	
62	API_InverseKin	Inverse kinematics	

- Remark：

3.SDK API Return Value Description

Return Samples
• Success：TRUE
• Fail：FALSE
Return parameter description

Parameter Name	Type	Description
TRUE	int	1
FALSE	int	0

Remark
•

4.SDK API Interface Description

4.1.Testing Interface

Brief description
• Testing Interface
Interface Definition
• void API_HelloWorld()
Remark
• This interface is used by secondary developers for testing. It has nothing to do with robots. Verify whether the user's library is loaded and the interface is exported correctly. After everything is successful, calling this interface will print "Hello World!"

4.2.Login, establish connection，all the basics of robot operation

Brief description
• Initialize and establish connection
Interface Definition
• BOOL API_Login(char* ip, int port,JTRHD& handle)
Parameter

Parameter Name	Required	Type	Description
ip	Yes	char*	IP address
port	Yes	int	port
handle	Yes	JTRHD	Login handle, input, output

Remark
•

4.3.Log out, disconnect, and cannot operate the robot again after disconnection

Brief description
• Log out, disconnect
Interface Definition
• BOOL API_Logout(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.4.Set whether to debug mode or not

Brief description
• When set to debug mode, debug information will be printed
Interface Definition
• BOOL API_SetDebugMode(JTRHD handle,const BOOL is_debug)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
is_debug	Yes	BOOL	1:Yes 0:No.input

Remark
•

4.5.Get whether to debug mode or not

Brief description
• Get whether the current mode is debugging mode
Interface Definition
• BOOL API_GetDebugMode(JTRHD handle,const BOOL& is_debug)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
is_debug	Yes	BOOL	1：Yes 0：No. input, output

Remark
•

4.6.Get error message description

Brief description
• Get the error description of the last API call error
Interface Definition
• BOOL API_GetErrorInfo(JTRHD handle,const ErrorInfo* info)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
info	Yes	ErrorInfo	input, output

Remark
•

4.7.Power on

Brief description
• Power on
Interface Definition

• BOOL API_PowerOn(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
• If the robot has already been powered on and is powered on repeatedly, the interface returns a failure. Can be used in conjunction with pattern queries or power off interfaces.

4.8.Release the brake

Brief description
• Release the brake
Interface Definition
• BOOL API_ReleaseBrake(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•The robot brake is released. When the machine is powered on, this function is sent through state feedback to release the brake and activate the robot.

4.9.Power Off

Brief description
• Power off
Interface Definition
• BOOL API_PowerOff(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
• Robot body powered off

4.10.Start

Brief description
• Start the robot program to run
Interface Definition
• BOOL API_Play(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.11.Stop

Brief description
• Stop the robot program from running
Interface Definition
• BOOL API_Stop(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.12.Pause

Brief description
• Pause the robot program
Interface Definition
• BOOL API_Pause(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.13.Load Program

Brief description
• Open existing programs remotely
Interface Definition
• BOOL API_LoadProgram(JTRHD handle,const char* program_file_name)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
program_file_name	Yes	char*	program name，input

Remark
•

4.14.Load installation configuration

Brief description
• Remote load installation configuration
Interface Definition
• BOOL API_LoadInstallation(JTRHD handle,const char* installation_file_name)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
installation_file_name	Yes	char*	Installation configuration file name，input

Remark
•

4.15.Shut down controller

Brief description
• Shut down controller
Interface Definition
• BOOL API_Shutdown(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.16.Close connection

Brief description
• Close connection
Interface Definition
• BOOL API_Quit(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.17.Running status query

Brief description
• Running status query
Interface Definition
• BOOL API_Running(JTRHD handle,ProgramState* program_state)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
program_state	Yes	ProgramState*	Description of running status，input, output

Remark
•

4.18.Query if the program has been saved

Brief description
• Query if the program has been saved
Interface Definition
• BOOL API_IsProgramSaved(JTRHD handle,ProgramSaved* program_saved)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
program_saved	Yes	ProgramSaved*	Whether to save the description，input, output

Remark
•

4.19.Query if it is remote control

Brief description
• Query if it is remote control
Interface Definition
• BOOL API_IsInRemoteControl(JTRHD handle,InRemoteControl* in_remote_control)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
in_remote_control	Yes	InRemoteControl*	Description of remote control，input, output

Remark
•

4.20.Popup

Brief description
•
Interface Definition
• BOOL API_Popup(JTRHD handle,const char* popup_text)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
popup_text	Yes	char*	Pop up message，input

Remark
•

4.21.Close the popup

Brief description
•
Interface Definition
• BOOL API_ClosePopup(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.22.Add log messages to log history

Brief description

•

Interface Definition

• BOOL API_AddToLog(JTRHD handle,const char* log_text)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
log_text	Yes	char*	Log messages，input

Remark

•

4.23.Get version information of software installed on the robot

Brief description

•

Interface Definition

• BOOL API_GetControllerVersion(JTRHD handle,ControllerVersion* controller_version)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
controller_version	Yes	ControllerVersion*	Version information description，input, output

Remark

•

4.24.Get version number

Brief description

•

Interface Definition

• BOOL API_GetVersion(JTRHD handle,Version* version)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
version	Yes	Version*	Version number information description，input, output

Remark

•

4.25.Get robot model

Brief description

• iimt3、iimt5、iimt10、iimt16、iimt20 and so on

Interface Definition

• BOOL API_GetRobotModel(JTRHD handle,RobotModel* robot_model)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
robot_model	Yes	RobotModel*	Robot model information description，input, output

Remark
•

4.26.Get serial number

Brief description
•
Interface Definition
• BOOL API_GetSerialNumber(JTRHD handle,SerialNumber* serial_number)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
serial_number	Yes	SerialNumber*	Serial number information description，input, output

Remark
•

4.27.Get robot mode

Brief description
•
Interface Definition
• BOOL API_GetRobotMode(JTRHD handle,RobotMode* robot_mode)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
robot_mode	Yes	RobotMode*	Robot mode information description，input, output

Remark
•

4.28.Get which program is loaded

Brief description
•
Interface Definition
• BOOL API_GetLoadedProgram(JTRHD handle,LoadedProgram* loaded_program)
Parameter

Parameter Name	Required	Type	Description

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
loaded_program	Yes	LoadedProgram*	Information description of the loaded program，input, output

Remark
•
“Loaded program: <path to loaded program file>“ OR
“No program loaded”

4.29.Get Safety status

Brief description
•
Interface Definition
• BOOL API_GetSafetyStatus(JTRHD handle,SafetyStatus* safety_status)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
safety_status	Yes	SafetyStatus*	Description of Safety Status Information，input, output

Remark
•

4.30.Get program state

Brief description
•
Interface Definition
• BOOL API_GetProgramState(JTRHD handle,ProgramState* program_state)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
program_state	Yes	ProgramState*	Description of program status information，input, output

Remark
•

4.31.Get operational mode

Brief description
•
Interface Definition
• BOOL API_GetOperationalMode(JTRHD handle,OperationalMode* operational_mode)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
operational_mode	Yes	OperationalMode*	Operation mode information description，input, output

Remark
•

4.32.Set operational mode

Brief description
•
Interface Definition
• BOOL API_SetOperationalMode(JTRHD handle,const char* operational_mode)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
operational_mode	Yes	char*	Operational mode，manual/automatic，input

Remark
•

4.33.Clear operational mode

Brief description
•
Interface Definition
• BOOL API_ClearOperationalMode(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.34.Free drag mode

Brief description
• Free drag mode
Interface Definition
• BOOL API_FreeDriveMode(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.35.End free drag mode

Brief description
• End free drag mode
Interface Definition
• BOOL API_EndFreeDriveMode(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.36.Set to automatic run mode

Brief description
• Set to automatic run mode
Interface Definition
• BOOL API_SetAutoRun(JTRHD handle)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input

Remark
•

4.37.Mannul teach start

Brief description
• Mannul teach start
Interface Definition
• BOOL API_ManualTeachStart(JTRHD handle,int num,int direction)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
num	Yes	int	See remarks
direction	Yes	int	See remarks

Remark
• The manual teaching function instruction is equivalent to single joint and axis operations in the moving tab of the teaching pendant. The reference is as follows:
num
range 1~12
direction
0:reverse movement 1: positive movement

1~6
Joint number, joint movement
7~9
Cartesian coordinate movement x , y , z
Axis reverse movement

Axis positive movement
10~12
Cartesian rotation Rx , Ry , Rz
Reverse posture transformation
Positive posture transformation
Reference examples: 1 API_ManualTeachStart(handle,1,0) // The first joint runs in reverse 2 API_ManualTeachStart(handle,7,1) //X-axis positive movement 3 API_ManualTeachStart(handle,10,0) //Rx reverse movement

4.38.End mannul teach

Brief description
• End mannul teach
Interface Definition
• BOOL API_ManualTeachEnd(JTRHD handle,int num,int direction)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
num	Yes	int	See remarks
direction	Yes	int	See remarks

Remark
• The manual teaching end instruction is used after the teaching start instruction to end the teaching. Reference example: 1 API_ManualTeachEnd(handle,1,0) // The first joint has completed reverse operation 2 API_ManualTeachEnd(handle,7,1) //End of positive X-axis movement 3 API_ManualTeachEnd(handle,10,0) //Rx reverse movement ends

4.39.Servo motion

Brief description
•
Interface Definition
• BOOL API_Servoj(JTRHD handle,ServojParam* parameter,bool is_block=false,int timeout=0)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
parameter	Yes	ServojParam*	input
is_block	Yes	int	Can only be false
timeout	Yes	int	

typedef struct
{
double q[6];
float a;
float v;
float t;
float lookahead_time;
float gain;
}ServojParam;

Remark
• Port 29998 needs to be configured separately and planned in advance, mainly for use on ROS2

4.40.Joint motion

Brief description

•

Interface Definition

• BOOL API_Movej(JTRHD handle,MovejParam* parameter,bool is_block=true,int timeout=30)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
parameter	Yes	MovejParam*	input
is_block	Yes	int	true: block，false: non-block
timeout	Yes	int	timeout: The default value is 30s, which is valid when is-block is true

typedef struct
{
double p[6]; //Target point joint parameters, format: {0.0,1.57,0.0,-1.57,-1.57,0} ,unit: rad
float v; //Speed,rad/s
float a; //Joint speed,rad/s
float j; //Joint acceleration,rad/s^2
float r; //Trajectory blending radius,mm
float t; //Trajectory planning time, system retains parameter interface, can be left blank
}MovejParam;

Remark

•

4.41. Linear motion

Brief description

• Linear motion

Interface Definition

• BOOL API_Movel(JTRHD handle,MovelParam* parameter,bool is_block=true,int timeout=30)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
parameter	Yes	MovelParam*	input
is_block	Yes	int	true: block，false: non-block
timeout	Yes	int	timeout: The default value is 30s, which is valid when is-block is true

typedef struct
{
double p[6]; //Joint parameters
float v; //speed
float tool_v; //Tool speed

float tool_a; //Tool acceleration
float r; //blending radius,mm
}MovelParam;

Remark
•

4.42.Get digital input status

Brief description
• Get digital input status
Interface Definition
• BOOL API_GetDI(JTRHD handle,int num,IoStatus& status)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
num	Yes	int	0~15
status	Yes	IoStatus	input, output

typedef enum
{
IO_STATUS_LOW = 0,
IO_STATUS_HIGH
}IoStatus;
Remark
•

4.43.Get digital output status

Brief description
•Get digital output status
Interface Definition
• BOOL API_GetDO(JTRHD handle,int num,IoStatus& status)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
num	Yes	int	0~15
status	Yes	IoStatus	input, output

typedef enum
{
IO_STATUS_LOW = 0,
IO_STATUS_HIGH
}IoStatus;
Remark
•

4.44.Get all digital input status at once

Brief description
•

Interface Definition
•BOOL API_GetAllDI(JTRHD handle,AllIoStatus* status)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
num	Yes	int	0~15
status	Yes	AllIoStatus	input, output

Remark
•

4.45.Get all digital output status at once

Brief description
•
Interface Definition
•BOOL API_GetAllDO(JTRHD handle,AllIoStatus* status)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
num	Yes	int	0~15
status	Yes	AllIoStatus	input, output

Remark
•

4.46.Set digital output status

Brief description
•
Interface Definition
• BOOL API_SetDO(JTRHD handle,int num,IoStatus status)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
num	Yes	int	0~15
status	Yes	IoStatus	input

typedef enum
{
IO_STATUS_LOW = 0,
IO_STATUS_HIGH
}IoStatus;

Remark
•

4.47.Get robot status

Brief description

•

Interface Definition

• BOOL API_GetRobotStatus(JTRHD handle,RobotStatus& status)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
num	Yes	int	0~15
status	Yes	RobotStatus	input, output

typedef enum
{
ROBOT_POWEROFF, //0
ROBOT_POWEROFF_2_POWERON,//1
ROBOT_POWERON,//2
ROBOT_POWERON_2_STANBY,//3
ROBOT_STANDBY,//4
ROBOT_STANDBY_2_MAUNTEACH,//5
ROBOT_MAUNTEACH,//6
ROBOT_MANUTEACH_2_MANUTEACHEND,//7
ROBOT_MAUNTEACH_END,//8
ROBOT_SINGLETARGETMOVE,//9
ROBOT_PAUSE,
ROBOT_STANBY_2_AUTORUN,//11
ROBOT_AUTORUN,//—-12
ROBOT_FREEDRIVE,//13
ROBOT_FREEDRIVE_END,//14
ROBOT_STANDBY_2_RUNTARGET,//15
ROBOT_RUNTARGET,//16
ROBOT_STOP,//17
ROBOT_EMERGENCY,//18
ROBOT_ERROR,//19
ROBOT_LOCAL_REMOTE,//20
ROBOT_COLLISION,
}RobotStatus;

Remark

•

4.48.Get joint position radian

Brief description

•

Interface Definition

• BOOL API_GetRobotJointRad(JTRHD handle,JointPosition* position)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
position	Yes	JointPosition	input, output

typedef struct
{
double robotAngle[6]; //The current joint angle values of the robotic arm
double robotRad[6]; //The current joint radian values of the robotic arm
}JointPosition;

Remark
• Pay attention to the value of ‘robotRad’

4.49.Get joint position angle

Brief description
•
Interface Definition
• BOOL API_GetRobotJointAngle(JTRHD handle,JointPosition* position)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
position	Yes	JointPosition	input, output

typedef struct
{
double robotAngle[6]; //The current joint angle values of the robotic arm
double robotRad[6]; //The current joint radian values of the robotic arm
}JointPositon;

Remark
• Pay attention to the value of ‘robotAngle’

4.50.Get tool position

Brief description
•
Interface Definition
• BOOL API_GetRobotTcpPosition(JTRHD handle,TcpPosition* position)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
position	Yes	TcpPosition	input, output

typedef struct
{
double robotTcp[6]; //The current position and posture of the end effector of the robotic arm
}TcpPosition;

Remark
•

4.51.Get arrive status

Brief description
•
Interface Definition

• BOOL API_GetArriveStatus(JTRHD handle,ArriveStatus& status)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
status	Yes	ArriveStatus	input, output

typedef enum
{
ARRIVE_STATUS_ARRIVED = 0,
ARRIVE_STATUS_NOT_ARRIVED,
ARRIVE_STATUS_ERROR
}ArriveStatus;

Remark
•

4.52.Get running speed

Brief description
•
Interface Definition
• BOOL API_GetRunSpeed(JTRHD handle,int& speed)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
speed	Yes	int	input, output

Remark
•What is obtained here is not the actual speed, but a percentage value. Range: (0 ~ 100)

4.53.Set running speed

Brief description
•
Interface Definition
• BOOL API_SetRunSpeed(JTRHD handle,int speed)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
speed	Yes	int	input

Remark
•What is obtained here is not the actual speed, but a percentage value. Range: (0 ~ 100)

4.54.Script command execution

Brief description
• Remote execution of single or multiple line scripts through SDK interface, separated by “;” between multiple line scripts, without line break symbol “\ n”.
Interface Definition

• BOOL API_ExecuteScript(JTRHD handle,const char* script_content)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
script_content	Yes	char*	Script content executed,input

Remark
•Due to the timing of script execution being determined by the content of the script, this interface returns immediately. Users can combine other interface functions such as program state or reach state to complete execution tasks, and call the stop interface before execution.

4.55.Get variable value

Brief description
• Retrieve the variable value of a certain variable in the program
Interface Definition
• BOOL API_GetVariableValue(JTRHD handle,const char* var_name, double& value)

Parameter
Parameter Name	Required	Type	Description
:——	:———-	:—-
handle	Yes	JTRHD	Login handle, input
var_name	Yes	char*	input，Name of variable
value	Yes	double	input/output，Value of variable	

Remark
•

4.56.Set trajectory offset

Brief description
•
Interface Definition
•BOOL API_SetOffset(JTRHD handle,TrajectoryOffset offset)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
offset	Yes	TrajectoryOffset	input

typedef struct
{
double x_offset; //m
double y_offset; //m
double z_offset; //m
double rx_offset; //rad
double ry_offset; //rad
double rz_offset; //rad
}TrajectoryOffset;

Remark
•

4.57.Get trajectory offset

Brief description

•

Interface Definition

•BOOL API_GetOffset(JTRHD handle,TrajectoryOffset* offset)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
offset	Yes	TrajectoryOffset*	input, output

typedef struct
{
double x_offset; //m
double y_offset; //m
double z_offset; //m
double rx_offset; //rad
double ry_offset; //rad
double rz_offset; //rad
}TrajectoryOffset;

Remark

•

4.58.Send files to the teaching pendant

Brief description

•

Interface Definition

•BOOL API_SendFile(JTRHD handle,const char *local_absolute_path_file_name,const char* save_path,const char* save_file_name)

Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
local_absolute_path_file_name	Yes	char*	input，Local file name (absolute path)
save_path	Yes	char*	Save file path on the teaching pendant, input
save_file_name	Yes	char*	The file name presented on the teaching pendant, input

Remark

•

4.59.Remote execution of Lua script files

Brief description

•

Interface Definition

•BOOL API_ExecuteFile(JTRHD handle,const char* remote_absolute_path_file_name)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle, input
remote_absolute_path_file_name	Yes	char*	Remote (on teaching pendant) absolute path file name, input

Remark
•

4.60.Get collision level

Brief description
•
Interface Definition
•BOOL API_GetCollisionLevel(JTRHD handle,CollisionLevel& level)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle，input
level	Yes	CollisionLevel	level collision level，range[0,5],0:close collision。collision threshold 1:10N,2:20N,3:30,4:40N,5:50N，input,output

Remark
typedef enum
{
COLLISION_LEVEL_ZERO = 0,
COLLISION_LEVEL_ONE,
COLLISION_LEVEL_TWO,
COLLISION_LEVEL_THREE,
COLLISION_LEVEL_FOUR,
COLLISION_LEVEL_FIVE,
}CollisionLevel;

4.61.Set collision level

Brief description
•
Interface Definition
•BOOL API_SetCollisionLevel(JTRHD handle,const CollisionLevel level)
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle，input

Parameter Name	Required	Type	Description
level	Yes	CollisionLevel	level collision level，range[0,5],0:Close collision。Collision threshold 1:10N,2:20N,3:30,4:40N,5:50N，input

Remark
•

4.62.Inverse kinematics

Brief description
•
Interface Definition
•BOOL API_InverseKin(JTRHD handle,const TcpPosition *tcp_pose,const JointRad* ref_joint_rad,JointRad* joint_rad);
Parameter

Parameter Name	Required	Type	Description
handle	Yes	JTRHD	Login handle，input
tcp_pose	Yes	TcpPosition*	tcp_pose:Cartesian spatial pose(unit:x/y/z (m),rx/ry/rz (radians))，input
ref_joint_rad	Yes	JointRad*	Reference joint space position for inverse solution calculation (unit: radians), input
joint_rad	Yes	JointRad*	Joint spatial position calculation result (unit: radians), input/output

Remark
typedef struct
{
double robotRad[6]; //Joint radian
}JointRad;

typedef struct
{
double robotTcp[6]; //The current position and posture of the end effector of the robotic arm
}TcpPosition;