HansRobot_Pro Interface Description

Shenzhen Han's Robot Co.,Ltd.

1 Edit History:

Version N0.	Date	Modified Content
V1.0	2019-05-08	Match version 3.5.503.416
V1.2	2019-08-08	Add interface to match version 3.5.413.550

2 Purpose of this document:

- ✧ Provide reference for interface preparation and maintenance of R&D personnel
- ✧ Provide interface instructions for other people using the HansRobot_Pro interface

1 Introduction

1.1 Overview

The Hans Robot_Pro module is a software module for secondary development. Users can use this module to develop a host computer that communicates with the controller.。

1.2 Call interface

For example, using movej to move to zero, the interface MoveJ(int boxID, int rbtID, double J1, double J2, double J3, double J4, double J5, double J6) is called, as follows:

```
int nRet = MoveJ(0,0,0,0,0,0,0,0);//movej
```

nRet is the return value, success is 0, and the failure returns an error code.。

The specific error code is shown in reference 4.

1.3 Terminology

Terminology	Explain
ACS	Joint coordinates, unit degree
PCS	World coordinate, unit mm、unit degree

2 System Structure

The protocol adopts the standard C/S architecture, in which the Han's Robot provides a server that receives and processes robot control messages, and the entire communication process is performed by TCP/IP.

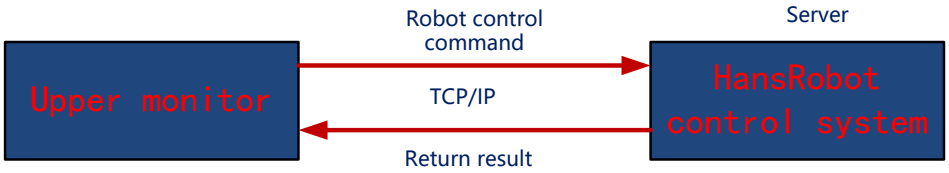

Fig 2-1 system structure

2.1 Connection Mode

2.1.1 As Server

The Han's Robot control system acts as a communication server and listens to the designated port when it starts up. When using the HansRobot_Pro module for secondary development, the IP parameter needs to be passed in, and the port uses the specified port 10003..

3 Communication Protocol

3.1 HRIF_Connect

Features	Connection controller	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	const char* hostName	Controller IP address
	unsigned short nPort	Controller port, 10003
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	Error code, please check the error code table of windows

3.2 HRIF_DisConnect

Features	Disconnect from the controller	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.3 HRIF_IsConnect

Features	Check if the controller is connected	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4

Return	true = connection false = disconnect	

3.4 Electrify

Features	Power the robot	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	Return value when entering the startup master state

3.5 ElectrifyEx

Features	Power the robot	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	bool bSynchronized	Whether to return after completion ture：Yes, return value when entering the normal status. flase：No, return value when entering the state of starting the master station.
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	successful completion of power up before returning, power up time is about 44s.

3.6 BlackOut

Features	Robot blackout	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	successful power outage will only return, power failure time is 3s.

3.7 ConnectRobotECAT

Features	Start master station	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.8 DisonnectRobotECAT

Features	Close master station	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.9 StartAssistiveMode

Features	Opening zero force teaching mode	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.10 CloseAssistiveMode

Features	Close the zero force teaching mode	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.11 GrpPowerOn

Features	Robot servo on	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4

	int rbtID	Robot index, default is 0, maximum is 4
	bool bSynchronized	Whether to return after completion ture：Yes flase：No
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.12 GrpPowerOff

Features	Robot servo off	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.13 GrpReset

Features	Reset, used to clear errors	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and	

	the error types see the error code list.	

3.14 GrpStop

Features	Stop robot	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.15 ShortJogJ

Features	Angular motion, fixed distance exercise(the movement distance is 2 degrees)	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int AxisID	Motion axis ID, starting from 0, up to 5
	int Derection	Direction of motion axis: 0 = negative direction; 1 = positive direction
Return	Successfully returns 0, failure returns error code and	

	the error types see the error code list.	

3.16 ShortJogL

Features	Space motion, fixed distance motion (motion distance: 2 mm for X, Y, and Z axes, 2 degrees for RX, RY, and RZ axes)	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int AxisID	Motion axis ID, starting from 0, up to 5
	int Derection	Direction of motion axis: 0 = negative direction; 1 = positive direction
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.17 LongJogJ

Features	Angular, unfixed distance movement	Notes: When the order is issued, the other stop command must be issued to stop the movement
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int AxisID	Motion axis ID, starting from

		0, up to 5
	int Derection	Direction of motion axis: 0 = negative direction; 1 = positive direction
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.18 LongJogL

Features	Spatial motion, unfixed distance motion	Notes: When the order is issued, the other stop command must be issued to stop the movement
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int AxisID	Motion axis ID, starting from 0, up to 5
	int Derection	Direction of motion axis: 0 = negative direction; 1 = positive direction
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.19 MoveJ

Features	Robot moves to the specified	

	angular coordinate position	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double J1	Direction 1 command position, unit degree
	double J2	Direction 2 command position, unit degree
	double J3	Direction 3 command position, unit degree
	double J4	Direction 4 command position, unit degree
	double J5	Direction 5 command position, unit degree
	double J6	Direction 6 command position, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.20 MoveL

Features	Robot moves straight to the specified space coordinates	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double X	X direction command position, unit mm

	double Y	Y direction command position, unit mm
	double Z	Z direction command position, unit mm
	double RX	RX direction command position, unit degree
	double RY	RY direction command position, unit degree
	double RZ	RZ direction command position, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.21 MoveB

Features	Immediately change the end point of the robot's current movement to the specified space coordinate position	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double X	X direction command position, unit mm
	double Y	Y direction command position, unit mm
	double Z	Z direction command position, unit mm
	double RX	RX direction command position, unit degree

	double RY	RY direction command position, unit degree
	double RZ	RZ direction command position, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.22 MoveC

Features	Arc motion	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double viaCoord[3]	Through position，X、Y、Z direction command position, unit mm
	double GoalCoord[6]	Target position，X、Y、Z、RX、RY、RZ direction command position, unit mm/degree
	int type	0：Unconstrained, the start point transforms to the EndPoint to define tool orientation 1：Fixed, only the start point is used to define tool orientation
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.23 MoveHoming

Features	Robot returns to the origin	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.24 StartServoJ

Features	Set the fixed update cycle and look-ahead time when starting the robot online control (servoJ or servoP).	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double dServoTime	The period of fixed location update is recommended to be greater than 0.015s
	double dlookAheadTime	Prospective time, recommended between 0.05s and 0.2s
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.25 PushServoj

Features	Online joint position command control, the joint position is sent with the fixed update time set by StartServo, and the robot will track the joint position command in real time.	Note: The user needs to input a continuous track position. If there is no update position for more than 2 update cycles, the line control will be turned off.
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double J1	Direction 1 command position, unit degree
	double J2	Direction 2 command position, unit degree
	double J3	Direction 3 command position, unit degree
	double J4	Direction 4 command position, unit degree
	double J5	Direction 5 command position, unit degree
	double J6	Direction 6 command position, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.26 PushServoP

Features	The online terminal TCP position command control transmits the TCP position with the fixed update time set by the StartServo, and the robot reversely converts the target position of the target TCP position in real time.	Note: The user needs to input a continuous track position. If there is no update position for more than 2 update cycles, the line control will be turned off.
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double X	X direction command position, unit mm
	double Y	Y direction command position, unit mm
	double Z	Z direction command position, unit mm
	double RX	RX direction command position, unit degree
	double RY	RY direction command position, unit degree
	double RZ	RZ direction command position, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.27 StartPushBlending

Features	Start PushBlending (continuous motion), initialize, you can start Push point	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.28 PushBlendingL

Features	PushMoveL point.	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double X	X direction command position, unit mm
	double Y	Y direction command position, unit mm
	double Z	Z direction command position, unit mm
	double RX	RX direction command position, unit degree
	double RY	RY direction command position, unit degree

	double RZ	RZ direction command position, unit degree
	double dRadius	Transition radius, unit mm
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.29 PushBlendingC

Features	PushMoveC point.	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double Via_X	X-axis position of the transition point, unit mm
	double Via_Y	Y-axis position of the transition point, unit mm
	double Via_Z	X-axis position of the transition point, unit mm
	double Via_RX	RX-axis position of the transition point, unit degree
	double Via_RY	RY-axis position of the transition point, unit degree
	double Via_RZ	RZ-axis position of the transition point, unit degree
	double Tag_X	X-axis position of the end point, unit mm
	double Tag_Y	Y-axis position of the end point, unit mm
	double Tag_Z	Z-axis position of the end

		point, unit mm
	double Tag_RX	RX-axis position of the end point, unit mm
	double Tag_RY	RY-axis position of the end point, unit mm
	double Tag_RZ	RZ-axis position of the end point, unit mm
	int nFixposure	0 = Unconstrained, the start point transforms to the EndPoint to define tool orientation 1 = Fixed, only the start point is used to define tool orientation
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.30 EndPushBlending

Features	End PushBlending, send the point to the controller and start motion, calculate error or parameter error need to re-Push	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error	

	code list.	

3.31 StartPushDragMove

Features	Start DragMove motion (drag track motion), initialize, you can start Push point	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.32 PushDragMove

Features	PushDragMove point	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double J1	Direction 1 command position, unit degree
	double J2	Direction 2 command position, unit degree
	double J3	Direction 3 command position, unit degree
	double J4	Direction 4 command position, unit degree
	double J5	Direction 5 command position,

		unit degree
	double J6	Direction 6 command position, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.33 EndPushDragMove

Features	End the DragMove movement, send the point to the controller and start moving	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.34 HRIF_GetRbtErrorCodeStr

Features	Get the description of the error code	
Parameter	int nErrCode	Error code
	char* szErrMsg	File address
	int nLen	File size
Return	Successfully returns 0, failure returns error code and the error types see the error	

	code list.	

3.35 HRIF_GetErrorCodeLevel

Features	Get the error code corresponding level	
Parameter	int nErr	Error code
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.36 HRIF_SetErrorCallback

Features	Set the error callback function, the robot will report the error and execute the callback function.	
Parameter	void *HRIF_ErrorCallback	Callback
Return	Return 0	

3.37 HRIF_ErrorCallback

Features	Callback	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int nAxisID	Error axis, default is 0, maximum is 5
	int nErrorCode	Error code

3.38 IsRobotError

Features	Determine if the robot reports an error	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	true = error false = no error	

3.39 IsRobotPowerOn

Features	Determine if the robot is enabled	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	true = Enable false = Disable	

3.40 IsRobotMoving

Features	Determine if the robot is in motion	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
Return	true = In motion	

	false = Not in motion	

3.41 ReadVersion

Features	Read controller version	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	char* szVersion	File adder
	int nLen	File Size
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.42 ReadRbtACS

Features	Read the current joint position of the robot	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double* J1	The position of J1 direction, unit: degree
	double* J2	The position of J2 direction, unit: degree
	double* J3	The position of J3 direction, unit: degree
	double* J4	The position of J4 direction, unit: degree
	double* J5	The position of J5 direction, unit: degree

	double* J6	The position of J6 direction, unit: degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.43 ReadRbtPCS

Features	Read the current joint position of the robot	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double* X	The position of X direction, unit: mm
	double* Y	The position of Y direction, unit: mm
	double* Z	The position of Z direction, unit: mm
	double* RX	The position of RX direction, unit degree
	double* RY	The position of RY direction, unit degree
	double* RZ	The position of RZ direction, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.44 ReadRobotCurStatus

Features	Read the current state of the robot	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int* nCurStatus	Current status of DCS: 0=uninitialized 1=Initialization 2=The system board is not connected 3=The system board is not powered. 4=Rtos is not started 5=The controller is not started 6=Config 7=Emergency stop 8=Safe light curtain 9=Error status 10=normal 11=Run the script 12=The primary station is not started 13=zero force teaching state 14=The master station is starting up
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.45 ReadConveyorPos

Features	Get the value of the conveyor encoder	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double* dbConveyorPos	The value of the conveyor encoder, unit mm
	int* nConveyorCount	Encoder current position count value
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.46 ReadOverride

Features	Read speed ratio	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double* Override	Current speed ratio, 0.01~1 (1~100%)
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.47 ReadEndDOState

Features	Read the end output IO state	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int ioIndex	The IO index to read, count from one
	int* state	The state of output IO: 0=low level; 1=high level;
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.48 ReadEndDIState

Features	Read the end input IO state	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int ioIndex	The IO index to read, count from one
	int* state	The state of input IO: 0=low level; 1=high level;
Return	Successfully returns 0, failure returns error code and the error types see the error	

	code list.	

3.49 ReadEndBINState

Features	Read end effector IO status	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int& nEndBINSquare	The state of the square button IO: 0 = low level; 1 = high level;
	int& nEndBINPlus	Plus button IO status: 0 = low level; 1 = high level;
	int& nEndBINTriangle	The state of the triangle button IO: 0 = low level; 1 = high level;
	int& nEndBINRound	The state of the round button IO: 0 = low level; 1 = high level;
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.50 ReadElectricityBoxIO

Features	Read the box IO status	

Parameter	ST_ElectricityBoxInfo& boxState	typedef struct
		_ST_ElectricityBoxAnalogOutput_
		{
		int nAnalogMode;
		double dbAnalog;
		}ST_ElectricityBoxAnalogOutput;
		
		typedef struct
		_ST_ElectricityBoxInfo_
		{
		int nRemotePoweroff;
		int nDInputIO[8];
		int nSInputIO[8];
		int nExterInputIO[4];// There are 4 electric boxes in the first generation.
		int nDOutputIO[8];
		int nSOutputIO[8];
		int nExterOutputIO[4];// There are 4 electric boxes in the first generation.
		ST_ElectricityBoxAnalogOutput stAnalogOutput[2];
		double dbAnalogInput[2];
		double dbRobotSrcVoltage;
		double dbRobotSupplyVoltage;
		double dbRobotSupplyCurrent;
		double dbIOVoltage;
		double dbIOCurrent;
		}ST_ElectricityBoxInfo;

Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

Structure description:

ST_ElectricityBoxAnalogOutput	int nAnalogMode	Analog mode: 0 = voltage mode 1 = current mode
	double dbAnalog	Analog value: Current unit: mA, voltage unit: V
ST_ElectricityBoxInfo	int nRemotePoweroff	Remote shutdown IO status: 0 = low level 1 = high level
	int nDInputIO[8]	General purpose input IO status: 0 = low level 1 = high level
	int nSInputIO[8]	Configurable input IO status: 0 = low level 1 = high level
	int nExterInputIO[4]	16~19IO status of the first generation electric box input
	int nDOutputIO[8]	General purpose output IO status: 0 = low level 1 = high level
	int nSOutputIO[8]	Configurable output IO status: 0 = low level 1 = high level
	int nExterOutputIO[4]	16~19IO status of the first generation electric box output

	ST_ElectricityBoxAnalogOutput stAnalogOutput[2]	Analog structure
	double dbAnalogInput[2]	End input voltage analog
	double dbRobotSrcVoltage	48V input voltage
	double dbRobotSupplyVoltage	48V output voltage
	double dbRobotSupplyCurrent	Total current
	double dbIOVoltage	IO supply voltage
	double dbIOCurrent	IO supply current

3.51 ReadElectricityBoxDO

Features	Read the electrical box input IO status	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int ioIndex	The IO index to read, count from one
	int* state	The state of output IO: 0=low level; 1=high level;
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.52 ReadElectricityBoxDI

Features	Read the electrical box output IO status	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int ioIndex	The IO index to read, count

		from one
	int* state	The state of input IO: 0=low level; 1=high level;
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.53 ReadSDO

Features	读取 SDO 值	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int nSlaveID	Slave ID, starting from 0
	int nIndex	SDO index;
	int nSubIndex	SDO subindex
	int nSDODataType	SDO date type: 0: Byte ; 1: WORD ; 2: DWORD ;
	int& nValue	Read result
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.54 ReadForceVel

Features	Read force sensor speed	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4

	int rbtID	Robot index, default is 0, maximum is 4
	double& dVelX	X direction speed, unit: N/s
	double& dVelY	Y direction speed, unit: N/s
	double& dVelZ	Z direction speed, unit: N/s
	double& dVelRX	RX direction speed, unit: Nm/s
	double& dVelRY	RY direction speed, unit: Nm/s
	double& dVelRZ	RY direction speed, unit: Nm/s
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.55 ReadForceValue

Features	Read force sensor value	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double& dForceX	Force sensor X direction value, unit: N
	double& dForceY	Force sensor Y direction value, unit: N
	double& dForceZ	Force sensor Z direction value, unit: N
	double& dForceRX	Force sensor RX direction value, unit: Nm
	double& dForceRY	Force sensor RY direction value, unit: Nm

	double& dForceRZ	Force sensor RZ direction value, unit: Nm
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.56 GetPoseInterpolate

Features	Get linear interpolation position	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double dFromX	X-axis position of the starting point, unit mm
	double dFromY	Y-axis position of the starting point, unit mm
	double dFromZ	Z-axis position of the starting point, unit mm
	double dFromRX	RX-axis position of the starting point, unit degree
	double dFromRY	RY-axis position of the starting point, unit degree
	double dFromRZ	RZ-axis position of the starting point, unit degree
	double dToX	X-axis position of the end point, unit mm
	double dToY	Y-axis position of the end point, unit mm

	double dToZ	Z-axis position of the end point, unit mm
	double dToRX	RX-axis position of the end point, unit degree
	double dToRY	RY-axis position of the end point, unit degree
	double dToRZ	RZ-axis position of the end point, unit degree
	double dAlpha	Calculate the ratio, range: 0~1
	double* dResultX	X-axis calculation position, unit mm
	double* dResultY	Y-axis calculation position, unit mm
	double* dResultZ	Z-axis calculation position, unit mm
	double* dResultRX	RX-axis calculation position, unit degree
	double* dResultRY	RY-axis calculation position, unit degree
	double* dResultRZ	RZ-axis calculation position, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.57 RobotPCS2ACS

Features	Inverse solution, spatial coordinates are converted to joint position	

Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double X	The position of X direction, unit: mm
	double Y	The position of Y direction, unit: mm
	double Z	The position of Z direction, unit: mm
	double RX	The position of RX direction, unit degree
	double RY	The position of RY direction, unit degree
	double RZ	The position of RZ direction, unit degree
	double* J1	The position of J1 direction, unit: degree
	double* J2	The position of J2 direction, unit: degree
	double* J3	The position of J3 direction, unit: degree
	double* J4	The position of J4 direction, unit: degree
	double* J5	The position of J5 direction, unit: degree
	double* J6	The position of J6 direction, unit: degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.58 SetKinematicCoordinate

Features	Setting tool coordinates	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double X	The position of X direction, unit: mm
	double Y	The position of Y direction, unit: mm
	double Z	The position of Z direction, unit: mm
	double RX	The position of RX direction, unit degree
	double RY	The position of RY direction, unit degree
	double RZ	The position of RZ direction, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.59 SetUserCoordinate

Features	Setting user coordinates	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double X	The position of X direction,

		unit: mm
	double Y	The position of Y direction, unit: mm
	double Z	The position of Z direction, unit: mm
	double RX	The position of RX direction, unit degree
	double RY	The position of RY direction, unit degree
	double RZ	The position of RZ direction, unit degree
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.60 SetOverride

Features	Set speed ratio	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double Override	Set the speed ratio from 0.01 to 1 (1% to 100%)
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.61 SetPayload

Features	Seting load	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double dbPayload	Load mass,unit kilogram
	double masscenterX	Load centroid x coordinates,unit millimete
	double masscenterY	Load centroid y coordinates,unit millimete
	double masscenterZ	Load centroid z coordinates,unit millimete
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.62 SetTrackingSwitch

Features	Set control tracking switch	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int TrackSwitch	Follwing switch: 0 = close; 1 = open
Return	Successfully returns 0, failure returns error code and the error types see the error	

	code list.	

3.63 SetEndIOState

Features	Set the end output IO state	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int ioIndex	The IO index to set, count from one
	int ioState	Set state: 0 = low level; 1 = high level;
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.64 SetElectricityBoxDO

Features	Set the electrical box output IO status	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int bit	The IO index to set, count from one
	int State	Set state: 0 = low level; 1 = high level;
Return	Successfully returns 0,	

	failure returns error code and the error types see the error code list.	

3.65 SetAutoMode

Features	Switch to automatic mode, that is, the mode in which the script runs, return after the switch is completed	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.66 SetAutoModeWithoutReturn

Features	Switch to automatic mode, ie the mode in which the script runs, returning without waiting for the switch to complete	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.67 SetManualMode

Features	Switch to manual mode, which is a mode that is not scripted, returning without waiting for the switch to complete	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.68 SetManualModeWithoutReturn

Features	Switch to manual mode, that is, the mode of non-script running, returning without waiting for the switch to complete	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.69 WriteSDO

Features	Write SDO value	
Parameter	int boxID	Electric box ID, default is

		0, maximum is 4
	int nSlaveID	Slave ID, starting from 0
	int nIndex	SDO index;
	int nSubIndex	SDO subindex
	int nSDODataType	SDO date type: 0: Byte ; 1: WORD ; 2: DWORD ;
	int nValue	Target data
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.70 SetFCCmdType

Features	Set the force sensor work type	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int nType	job type: 0: Force control work 1: Drag and teach
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.71 SetFCFrameTpye

Features	Set the force control	

	coordinate system	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int nType	Select the coordinate system: 0: base system 1: TOOL coordinate system
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.72 SetFCEnable

Features	Set force sensor switch	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int nType	0: off 1: open
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.73 SetForceControlState

Features	Set the opening force switch	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4

	int nState	Force control switch: 0: off 1: open
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.74 SetFCFreedom

Features	Set force control dimension selection	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	int nFreedomX	Whether to choose the X direction: 0: Select 1: Do not select
	int nFreedomY	Whether to choose the Y direction: 0: Select 1: Do not select
	int nFreedomZ	Whether to choose the Z direction: 0: Select 1: Do not select
	int nFreedomRX	Whether to choose the RX direction: 0: Select 1: Do not select
	int nFreedomRY	Whether to choose the RY direction: 0: Select 1: Do not select
	int nFreedomRZ	Whether to choose the RZ direction: 0: Select 1: Do not select

Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.75 SetFCCmdForce

Features	设置力控目标力大小	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	int rbtID	Robot index, default is 0, maximum is 4
	double dForceX	X direction target force, unit: N
	double dForceY	Y direction target force, unit: N
	double dForceZ	Z direction target force, unit: N
	double dForceRX	Target force in RX direction, unit: Nm
	double dForceRY	Target force in RY direction, unit: Nm
	double dForceRZ	Target force in RZ direction, unit: Nm
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.76 ReadCurrentApplication

Features	Read the current script	

Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	char* szApplication	File address
	int nBufLen	Buffer size
	int& nSize	File size
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

Example：

```
char* szApplication = nullptr;
    int nBufLen=0;
    int nSize=0;
    ReadCurrentApplication(0, szApplication, nBufLen, nSize);
    if(nSize>0)
    {
        nBufLen=nSize;
        szApplication = new char[nSize];
        memset(szApplication,0,nBufLen);
        nSize = 0;
        ReadCurrentApplication(0, szApplication, nBufLen, nSize);
        delete szApplication;
    }
```

3.77 ImportApplication

Features	Start script time	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	const char* szApplication	Script content
Return	Successfully returns 0, failure returns error code and the error types see the error	

	code list.	

3.78 StartUDMTimer

Features	Start script timer	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.79 CloseUDMTimer

Features	Close script timer	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.80 RunUDM

Features	Running script	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	const char* szUDMCmd	Run function, for example: udm1,param1,;
	bool bWaitReturn	Whether to wait for return Ture = yes Flase = no
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.81 HoldUDM

Features	Pause script	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.82 ContinueUDM

Features	Pause script	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
Return	Successfully returns 0, failure returns error code and the error types see the error code list.	

3.83 ReadMainFirstPoint

Features	Read the position of the first motion instruction of the F_main function, and return to the current position without motion command	
Parameter	int boxID	Electric box ID, default is 0, maximum is 4
	double* J1	Axis 1 current position, unit degree
	double* J2	Axis 2 current position, unit degree
	double* J3	Axis 3 current position, unit degree
	double* J4	Axis 4 current position, unit degree
	double* J5	Axis 5 current position, unit degree
	double* J6	Axis 6 current position, unit degree
Return	Successfully returns 0,	

	failure returns error code and the error types see the error code list.	

4 ErrorCode

ErrorCode	Meaning
10000	Short circuit error
10001	Over voltage limit error
10002	Under voltage limit error
10003	Over velocity limit error
10004	Execute error
10005	Over current error
10006	Encoder error
10007	Following position error
10008	Following velocity error
10009	Negative limit error
10010	Positive limit error
10011	Server over heating error
10012	Max current error
10013	Emergency stop error
10014	UDM error
10015	Server parameter error
20000	Controller is not started
20001	Master is not started
20002	some slave is dropped
20003	Robot on safe stop state
20004	Robot on physical stop state
20005	Robot out safe space
20006	Robot enable time out
20007	Robot not electrify
20008	Starting master station error
30000	Collision shutdown
30001	Robot Collide with body
30002	Over joint limit error

30003	Singularity error
30004	General stopping criterion
30005	calculate failed
30006	UDM Status Error
30007	Init slave Error
30008	Home Step2 Error
30009	Out Of Direction Limit Error
30010	Out Of Direction Current Error
30011	Wrong load or mounting angle
30012	Motor limit temperature exceeded
1001	The robot has not been initialized
1002	Master station has not been started
1003	Slave station drop off
1004	The robot is safely locked
1005	The physical stop
1006	Robot has not been servo on
1007	Error reporting from slave station
1008	Robot beyond safe space
1009	In robot motion
1010	Invalid command
1011	Parameter error
1012	Function call format error
1013	Waiting for command execution
1014	IO does not exist
1015	Robots do not exist
1016	No connection server
1017	Network timeout
1018	Connection failed
1019	Serial connection failed
1020	No zero position is set
1021	The last same command has not been

	completed
1022	Serial port Di is empty
1023	Serial port DO is empty
1024	Wait timeout
1025	Error status
1026	Stop robot
1027	Robot has been servo off
1028	Robot has been servo on
1029	Function has not been enabled
1030	Start master timeout
1031	The robot has not been powered on
1032	Serial port has not been started
1033	The simulation state command is invalid
1034	RTOS Library not exsit
1035	DCS Handle Command thread crash
1039	Script running
1040	Xml Param Error
1041	System Board Not Connect
1042	Controller Not Start
1043	Controller Statu Error
1044	Robot in TeachMode
1045	Robot Already Electrify
1046	Connect to Modbus Failed
1047	Master is Started
1048	Parameter over specified payload
1049	DCS Status Error
1050	Target position invalid
1051	Robot Drive Operating
1052	Start Master Error
1053	Initilize slaves Error!HomeStep2 Fail
1054	ModebusRTU disconnected state
1055	ModebusRTU is busy
1056	Blending didn't start

1057	Blending is not over
2000	Failed to load library
2001	The script is empty
2002	Compile error
2003	Reload script error
2004	Function does not exist
2005	Function return type error
2006	MissSignal1
2007	MissSignal2
2008	Parameter type error
2009	There is no header file included
2010	No return value
2012	UDM Stack Err
2013	Script been lock,maybe compiling
2014	Not In RunScript Statu
2015	Serial Close
2016	Serial Close
2017	Controller not started
2018	Socket Not Connected
2020	Function Name have Space.
2021	Socket Error
2022	Function broken stop.
2023	Timer running error
2024	Enable SwitchON key error

5 Example

```
HRIF_Connect(0,“127.0.0.1”,10003);// Connection controller
Electrify(0);
MoveJ(0,0, 0, 0, 0, 0, 0, 0);// Movement to zero
While(IsMoving(0,0))//等待运动完成
{
   Sleep(10);
}
MoveJ(0,0, 90, 90, 90, 90, 90, 90);// Exercise to 90 degrees for all joints
While(IsMoving(0,0))// Waiting for movement{
   Sleep(10);
}
```