机器人接口

/\*robot initialize

\* filename: configure file path

\* robot\_flag: 0: get from configure file;1:sia; 2:ur

\* return: 0:right other:wrong

\* \*/

**int** **hb\_initialize**(**int** robot\_flag, **char**\*filename);

/\*move robot by joint space

\* rjoint: target position(joint)

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveA**(robjoint\* rjoint, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveA**(robjoint\* rjoint, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

**void** **DualMoveA**(robjoint\* rjoint1,robjoint\* rjoint2, speed\* rspeed1,speed\* rspeed2, zone\* rzone1, zone\* rzone2, tool\* rtool1, tool\* rtool2, wobj\* rwobj1, wobj\* rwobj2);

/\*move robot by joint space

\* rpose: target position(pose)

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveJ**(robpose\* rpose, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveJ**(robpose\* rpose, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

**void** **DualMoveJ**(robpose\* rpose1, robpose\* rpose2, speed\* rspeed1,speed\* rspeed2, zone\* rzone1,zone\* rzone2, tool\* rtool1, tool\* rtool2, wobj\* rwobj1, wobj\* rwobj2);

/\*move robot in a straight line

\* rpose: target position(pose)

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveL**(robpose\* rpose, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveL**(robpose\* rpose, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

**void** **DualMoveL**(robpose\* rpose1, robpose\* rpose2, speed\* rspeed1,speed\* rspeed2, zone\* rzone1,zone\* rzone2, tool\* rtool1, tool\* rtool2, wobj\* rwobj1, wobj\* rwobj2);

/\*move robot in a circular arc

\* rpose: target position(pose)

\* rpose\_mid: middle position

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveC**(robpose\* rpose, robpose\* rpose\_mid, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveC**(robpose\* rpose, robpose\* rpose\_mid, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

**void** **DualMoveC**(robpose\* rpose1, robpose\* rpose2, robpose\* rpose\_mid1, robpose\* rpose\_mid2, speed\* rspeed1, speed\* rspeed2, zone\* rzone1, zone\* rzone2, tool\* rtool1, tool\* rtool2, wobj\* rwobj1, wobj\* rwobj2);

/\*move robot in a spiral line

\* rpose: target position(pose)

\* rpose\_mid: middle position

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveT**(robpose\* rpose, robpose\* rpose\_mid, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveT**(robpose\* rpose, robpose\* rpose\_mid, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

**void** **DualMoveT**(robpose\* rpose1, robpose\* rpose2, robpose\* rpose\_mid1, robpose\* rpose\_mid2, speed\* rspeed1, speed\* rspeed2, zone\* rzone1, zone\* rzone2, tool\* rtool1, tool\* rtool2, wobj\* rwobj1, wobj\* rwobj2);

/\*get offset position

\* rpose: positon

\* x: offset x value

\* y: offset y value

\* z: offset z value

\* k: offset roll

\* p: offset pitch

\* s: offset yaw

\*

\* return offset position

\* \*/

robpose **Offs**(**const** robpose\* rpose, **double** x, **double** y, **double** z, **double** k, **double** p, **double** s);

/\*move robot by joint space

\* rjoint\_rel: relative target position(joint)

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveRelA**(robjoint\* rjoint\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveRelA**(robjoint\* rjoint\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

/\*move robot by joint space

\* rpose\_rel: relative target position(pose)

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveRelJ**(robpose\* rpose\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveRelJ**(robpose\* rpose\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

/\*move robot in a straight line

\* rpose\_rel: relative target position(pose)

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveRelL**(robpose\* rpose\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveRelL**(robpose\* rpose\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

/\*move robot in a circular arc

\* rpose\_rel: relative target position(pose)

\* rpose\_mid\_rel: relative middle position

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveRelC**(robpose\* rpose\_rel, robpose\* rpose\_mid\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveRelC**(robpose\* rpose\_rel, robpose\* rpose\_mid\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

/\*move robot in a spiral line

\* rpose\_rel: relative target position(pose)

\* rpose\_mid\_rel: relative middle position

\* rspeed: move speed

\* rzone: area of turning

\* rtool: tool

\* rwobj: coordinate system

\*\_index: robot index

\* \*/

**void** **MultiMoveRelT**(robpose\* rpose\_rel, robpose\* rpose\_mid\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveRelT**(robpose\* rpose\_rel, robpose\* rpose\_mid\_rel, speed\* rspeed, zone\* rzone, tool\* rtool, wobj\* rwobj);

/\*B spline move in joint space

\* rjoint: Necessary points sequence (joint)

\* n: Necessary points sequence number

\* rspeed: velocity (time)

\* rtool: tool

\* rwobj: coordinate system

\* \_index: robot index

\* \*/

**void** **MultiMoveABS**(robjoint\* rjoint,**int** n, speed\* rspeed, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveABS**(robjoint\* rjoint,**int** n, speed\* rspeed, tool\* rtool, wobj\* rwobj);

/\*B spline move in joint space

\* rpose: Necessary points sequence (pose)

\* n: Necessary points sequence number

\* rspeed: velocity (time)

\* rtool: tool

\* rwobj: coordinate system

\* \_index: robot index

\* \*/

**void** **MultiMoveJBS**(robpose\* rpose,**int** n, speed\* rspeed, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveJBS**(robpose\* rpose,**int** n, speed\* rspeed, tool\* rtool, wobj\* rwob);

/\*B spline move in joint space

\* filename Necessary points sequence data filename

\* rspeed velocity (time)

\* rtool tool

\* rwobj coordinate system

\* \_index robot index

\* \*/

**void** **MultiMoveAJBS**(**char**\* filename, speed\* rspeed, tool\* rtool, wobj\* rwobj, **int** \_index);

**void** **MoveAJBS**(**char**\* filename, speed\* rspeed, tool\* rtool, wobj\* rwobj);

**void** **DualMoveAJBS**(**char**\* filename1,**char**\* filename2, speed\* rspeed1, speed\* rspeed2, tool\* rtool1, tool\* rtool2, wobj\* rwobj1, wobj\* rwobj2);

**void** **START\_ROBOT**(**int** \_index);

**void** **END\_ROBOT**(**int** \_index);

**void** **START\_ADDITIONAXIS**(**int** \_index);

**void** **END\_ADDITIONAXIS**(**int** \_index);

/\*power all

\* \*/

**void** **START**();

/\*poweroff all

\* \*/

**void** **END**();

//==============other interface type, same as above interface=========

/\*move robot by joint space

\* J: target position(joint)

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBMA\_M**(**char**\* J, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMA**(**char**\* J, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*move robot by joint space

\* P: target position(pose)

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBMJ\_M**(**char**\* P, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMJ**(**char**\* P, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*move robot in a straight line

\* P: target position(pose)

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBML\_M**(**char**\* P, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBML**(**char**\* P, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*move robot in a circular arc

\* P: target position(pose)

\* P\_mid: middle position

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBMC\_M**(**char**\* P,**char**\* P\_mid, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMC**(**char**\* P,**char**\* P\_mid, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*move robot in a a spiral line

\* P: target position(pose)

\* P\_mid: middle position

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBMT\_M**(**char**\* P,**char**\* P\_mid, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMT**(**char**\* P,**char**\* P\_mid, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*get offset position

\* P: positon

\* x: offset x value

\* y: offset y value

\* z: offset z value

\* k: offset roll

\* p: offset pitch

\* s: offset yaw

\* pname: offset data name

\* \*/

**void** **ROBOffs**(**char**\* P, **double** x, **double** y, **double** z, **double** k, **double** p, **double** s,**char**\* pname);

/\*move robot by joint space

\* rjrel: relative target position(joint)

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBMRA\_M**(robjoint\* rjrel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMRA**(robjoint\* rjrel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*move robot by joint space

\* rprel: relative target position(pose)

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBMRJ\_M**(robpose\* rprel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMRJ**(robpose\* rprel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*move robot in a straight line

\* rprel: relative target position(pose)

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBMRL\_M**(robpose\* rprel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMRL**(robpose\* rprel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*move robot in a circular arc

\* rprel: relative target position(pose)

\* rpmrel: relative middle position

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBMRC\_M**(robpose\* rprel, robpose\* rpmrel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMRC**(robpose\* rprel, robpose\* rpmrel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*move robot in a a spiral line

\* rprel: relative target position(pose)

\* rpmrel: relative middle position

\* S: move speed

\* Z: area of turning

\* T: tool

\* W: coordinate system

\*\_index: robot index

\* \*/

**void** **ROBMRT\_M**(robpose\* rprel, robpose\* rpmrel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMRT**(robpose\* rprel, robpose\* rpmrel, **char**\* S, **char**\* Z, **char**\* T, **char**\* W);

/\*B spline move in joint space

\* filename: Necessary points sequence data filename

\* S: velocity (time)

\* T: tool

\* W: coordinate system

\* \_index: robot index

\* \*/

**void** **ROBMAJBS\_M**(**char**\* filename, **char**\* S, **char**\* T, **char**\* W, **int** \_index);

**void** **ROBMAJBS**(**char**\* filename, **char**\* S, **char**\* T, **char**\* W);