

HitBot Studio PRODUCT MANUAL

Leading light collaborative robotic arm provider

Main Business:
Industrial Robot
Collaborative Robot
Electric Gripper

Edition Release

Edition Number	Publish Time	Update Content	Remarks
Hi.19. 1.0	21 st June 2019	First Edition Release	
Hi.19. 1.1	28 th June 2019	<ol style="list-style-type: none"> 1. Modify UI stuck issues; 2. Add the attribute of operation tray module; 3. Add a message pop-up; 	
Hi.19. 1.2	5 th July 2019	1. Software Crash Caused by Modifying Module Drag	
Hi.19. 1.3	8 th July 2019	<ol style="list-style-type: none"> 1. Add process name can be modified; 2. The problem of process delay caused by modifying frequent file writing; 3. Add and delete process data prompt; 4. Modify interface font; 	
Hi.19. 1.4	10 th July 2019	<ol style="list-style-type: none"> 1. Add the mouse over, module attributes reading function; 2. Modify and delete rules; 	
Hi.19. 1.5	18 th July 2019	<ol style="list-style-type: none"> 1. Add "delay and assignment" module in process configuration; 2. Create trajectory modification motion function and add operation panel; 	
Hi.19. 1.6	5 th August 2019	<ol style="list-style-type: none"> 1. Remove quick start and track creation functions; 2. Optimize software performance; 	
Hi.19. 1.7	13 th August 2019	1. Integrate and extend the IO card communication module, increase the number of IO from 12 groups to 28 groups;	
Hi.19. 2.2	1 st November 2019	<ol style="list-style-type: none"> 1. Add Sub-process and Reset Modules 2. Other related optimizations 	
Hi.19. 2.3	1 st November 2019	1. Optimized client communication and fixed tray bugs.	
Hi.19. 2.4	12 st December 2019	<ol style="list-style-type: none"> 1. Manually debug the client, and the receiving delay is extended to 20s to facilitate debugging; 2. Variable value plus dynamic monitoring, in the last column of the variable scale; 	

Content

Chapter I Company Profile	4
Chapter II Software Framework Description	6
Chapter III Software Installation and Uninstallation	7
1. Software installation	7
2. Software uninstallation and repair	8
3. Installation documentation	9
Chapter IV Software Operation and Module Function Description	10
1. Software first runtime environment settings	10
2. Software login	11
3. Software module description	12
1) Functional area	13
2) Programming base module list area	13
3) Programming area	14
4. Mechanical arm operation	15
1) Initialize	15
2) Mechanical arm operation	15
5. Programming operations	16
1) Programming example	16
2) Conditional operation	17
3) Subprocess	18
4) Point	20
5) Output signal	21
6) Tray	22
7) Electric clamping jaw	23
8) Reset	25
9) Client	27
6. System simulation and debugging	28
1) System simulation	28
2) System commissioning	29
Chapter V Frequently Asked Questions and Solutions	30
Chapter VI Appendix	31

Chapter I Company Profile

Huiling-tech Robotics Co.,Ltd is a leading supplier of lightweight collaborative mechanical arms and electric clamping jaws for robotics. Through nearly 10 years of independent research and development accumulation, HITBOT has successfully reduced the threshold of automation transformation of small and medium-sized enterprises from the dimensions of cost and applicability, and exported leading automation solution services in an efficient, low-cost and modular manner.

Founded in 2015, HITBOT has a core member from Harbin Institute of Technology and a superb technology and solution team. Within three years, HITBOT has successively won four rounds of investment from Lenovo Star (Fortune 500 Holding Incubator), Shunwei Capital and other investment institutions. Z-Arm collaborative mechanical arm is a rare industrial desktop mechanical arm in China. Through the independent research and development of core components, Z-Arm collaborative mechanical arm has greatly reduced the cost of enterprise automation, realized the sale price of 1,000 yuan, and achieved the sales volume of less than 1,000 units in half a year on the market. At present, the direct sales and agency channels of HITBOT have covered the whole world. The Z-Arm and EFG series have thousands of customers, including Huawei, Foxconn, Procter & Gamble, BGI and other top 500 companies and leading enterprises in the industry, and have been successfully

exported to dozens of overseas countries such as the United States, Japan and Canada.

Through the introduction of advanced production, testing equipment and process technology, the company has established a set of modern management system from market development, engineering evaluation, process control, quality assurance and after-sales service. In the future, HITBOT will continue to take technological innovation as the cornerstone, explore the possibility of new application scenarios on the basis of deeply ploughing the light industry market, let more enterprises improve efficiency, help more people free their hands, from Made in China to Created in China, and give more vitality to the Industrial 4.0 circuit.

Chapter II Software Framework Description

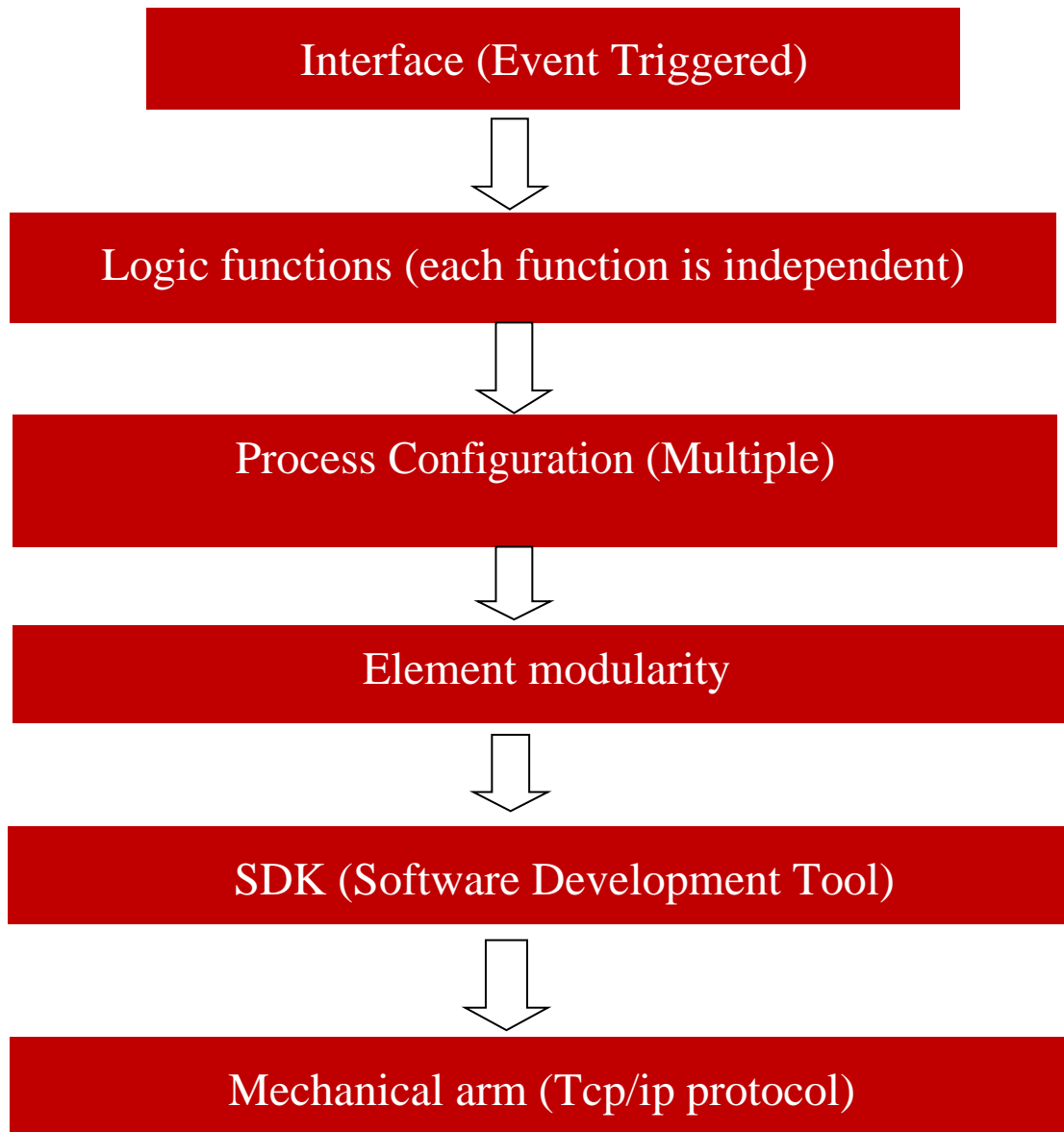
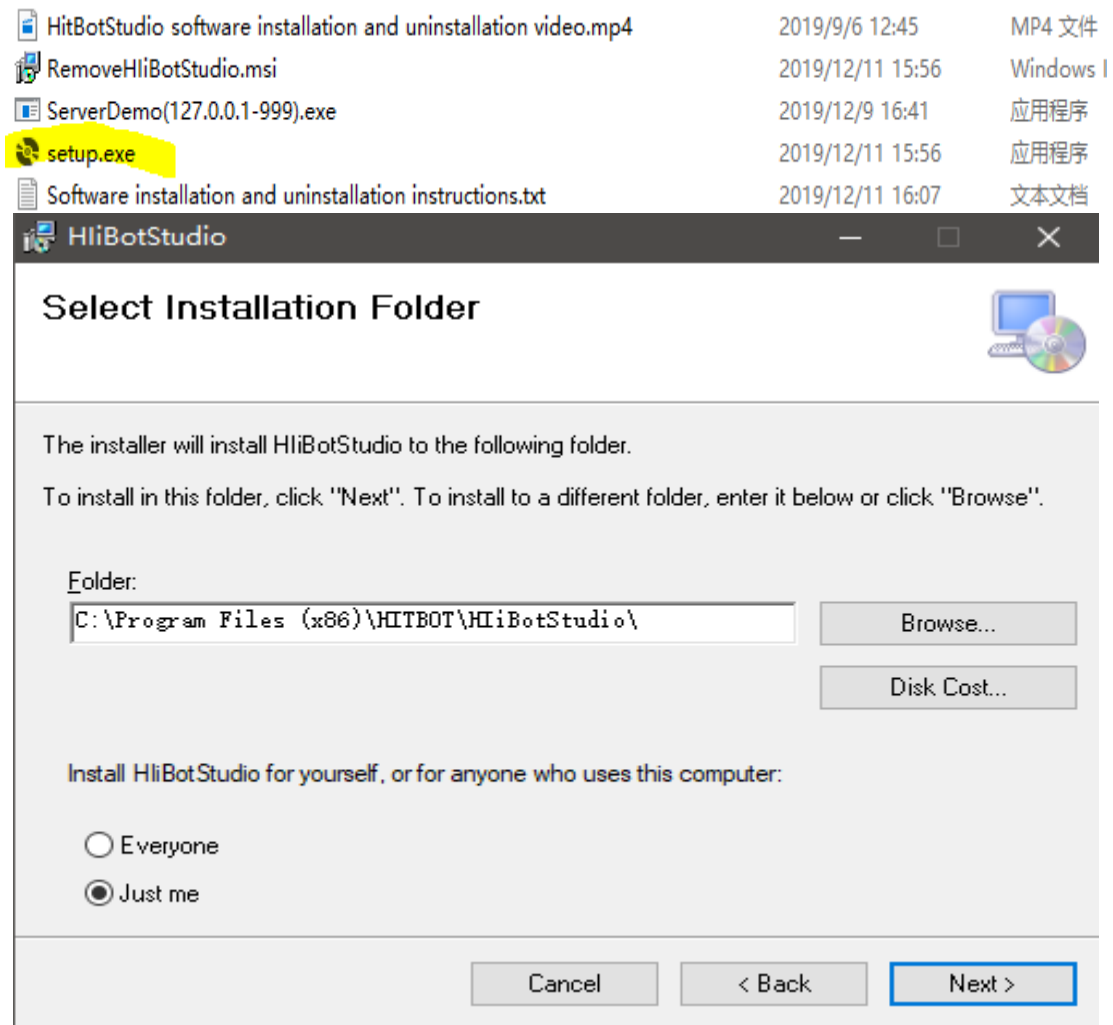


Figure 1. Software Framework Diagram

Chapter III Software Installation and Uninstallation






1. Software installation

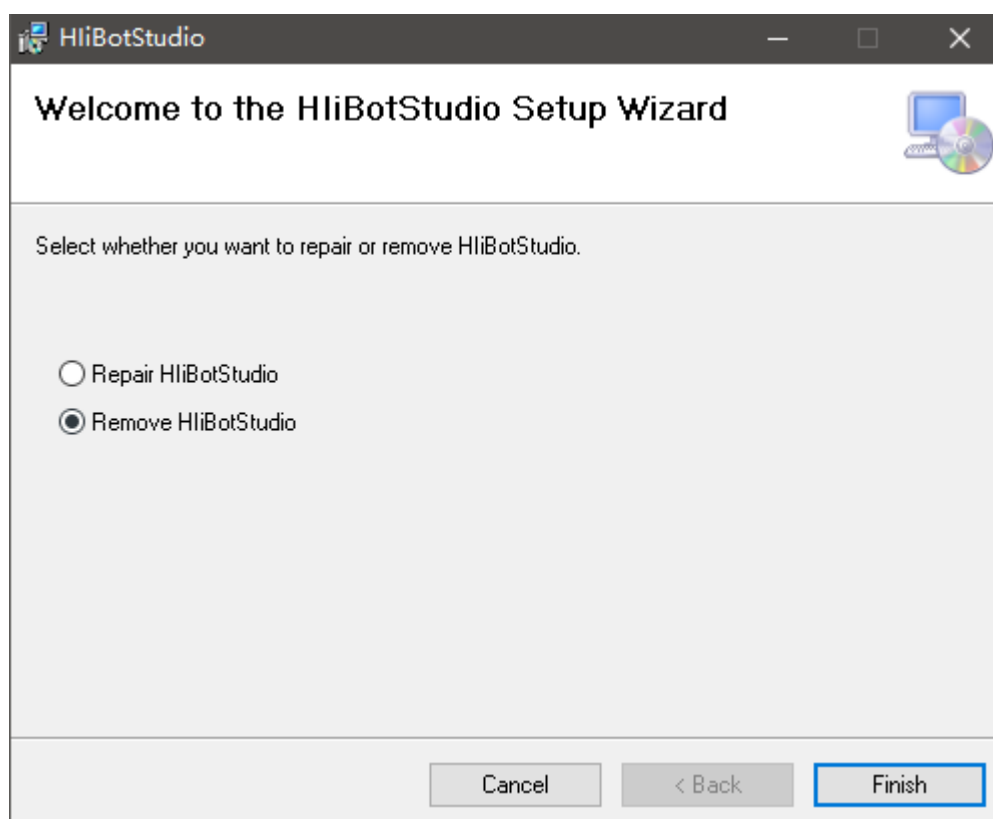
As shown in the figure, double-click the setup.exe file of the installation package to open the installation interface. Click the prompt to select the installation permission and directory, and then press the prompt to complete the installation. Click Close to exit the installation interface after the installation is complete. Note that you need to uninstall the old software before you can install the new edition of the software correctly when you upgrade the software edition. Please refer to the next section "Software Uninstallation and Repair" for details.

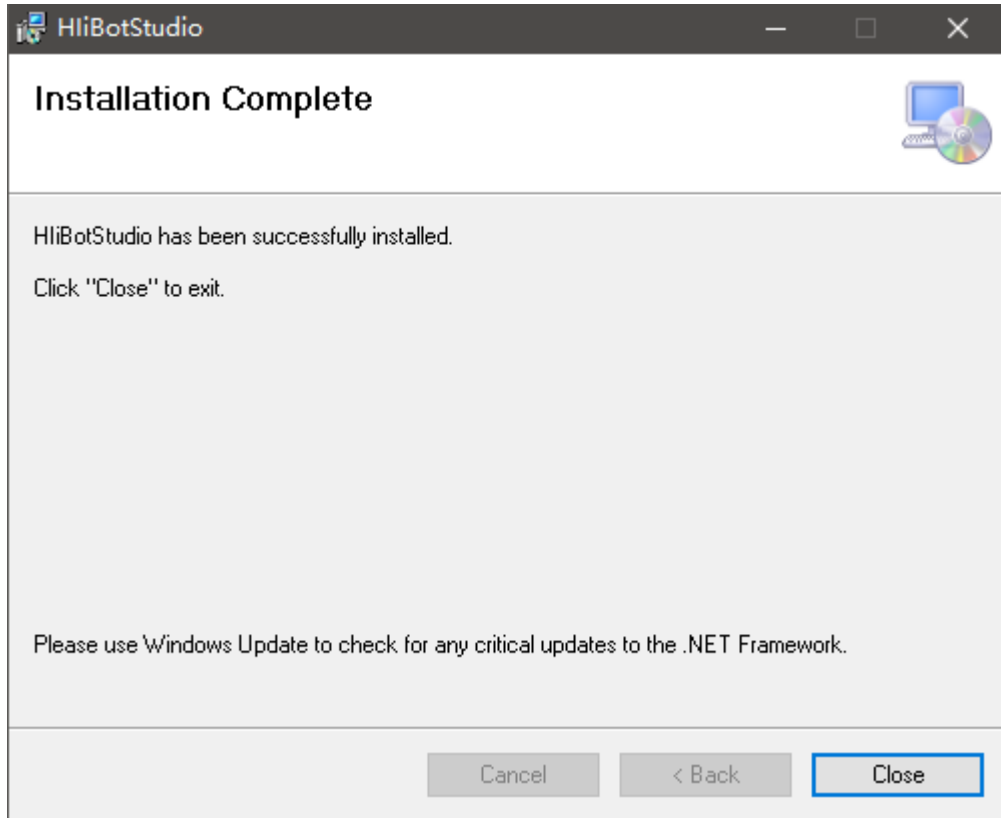


2. Software uninstallation and repair

This operation is required when a file is missing from the software or when a edition upgrade occurs. As shown in the figure, double-click the "RemoveHliBotStudio.msi" file of the installation package to open the repair and uninstall interface. After selecting the repair or uninstall operation as required, click "Finish" to finish the repair or uninstall operation.

 HitBotStudio software installation and uninstallation video.mp4	2019/9/6 12:45
 RemoveHliBotStudio.msi	2019/12/11 15:56
 ServerDemo(127.0.0.1-999).exe	2019/12/9 16:41
 setup.exe	2019/12/11 15:56
 Software installation and uninstallation instructions.txt	2019/12/11 16:07





3. Installation documentation

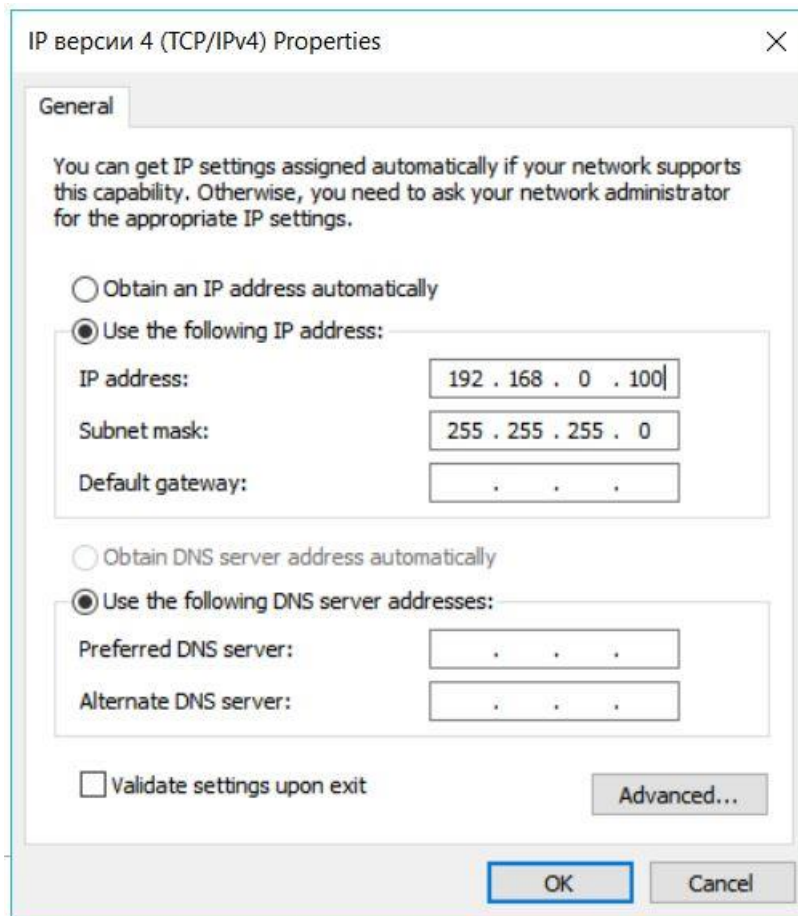
As shown in the figure, other unspecified items are found in the installation package. Please carefully watch the HitBotStudio Software Installation and read the Software Installation and Uninstallation Instructions.

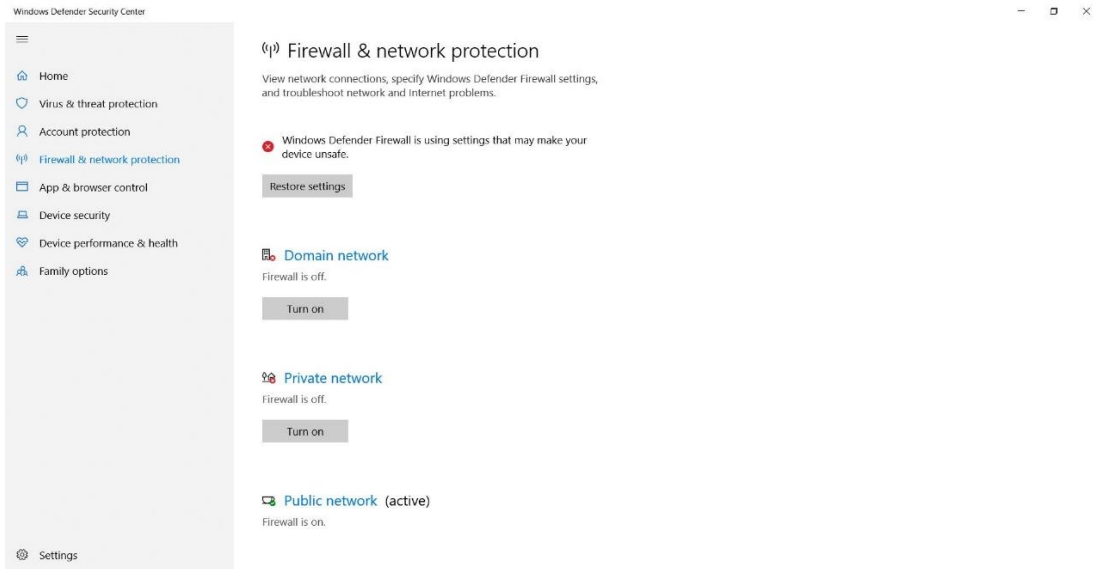
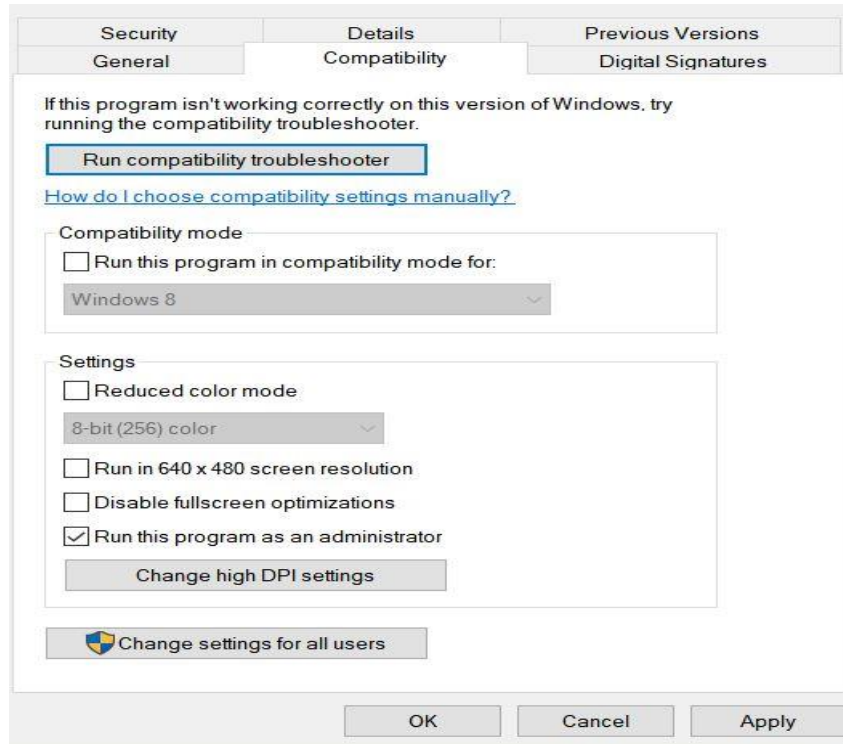
RemoveHliBotStudio.msi	2019/12/11 15:56
ServerDemo(127.0.0.1-999).exe	2019/12/9 16:41
setup.exe	2019/12/11 15:56
Software installation and uninstallation instructions.txt	2019/12/11 16:07

Chapter IV Software Operation and Module Function Description

1. Software first runtime environment settings

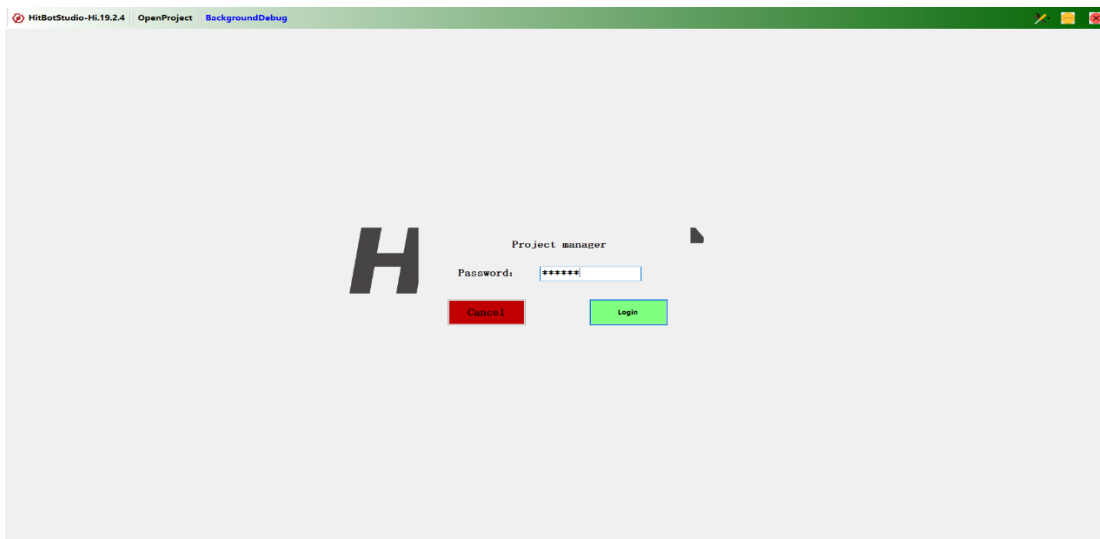
The first time to open the software connection mechanical arm, the relevant environment settings are required. As shown in the figure, in the first step, the network port connected to the mechanical arm is set as follows, the IP address is set to "192.169. 0.100" and the subnet mask is set to "255.255. 255.0"; Step 2: Right-click the software icon, select Property, select Compatibility, and check Run this program as administrator; Step 3: Turn off the firewall or add the application by "Allow the application to pass through the firewall."





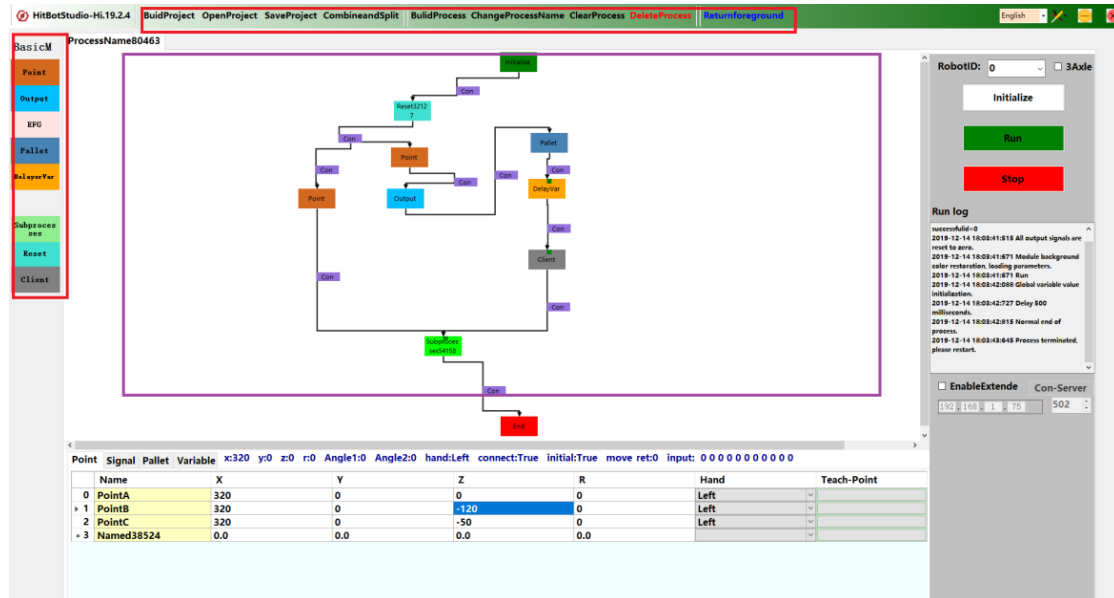
2. Software login

As shown in the figure, double-click the software icon to open the software and enter the main login interface. Click Background Debugging and enter the login password "hitbot" to enter the main login interface to perform relevant operations



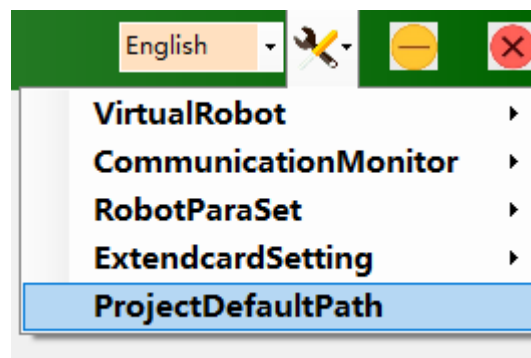
3. Software module description

As shown in the figure, after successful login, the main interface of the software will be entered, which is divided into three main parts: function area (red box), programming basic module list area (orange box) and programming area (purple box).



1) Functional area

As shown in the following figure, the functional area is mainly composed of three parts: project management, process management, system simulation and debugging



2) Programming base module list area

Based on graphical programming, it provides users with basic functional blocks such as point position, output signal, electric clamping jaw, tray and so on



3) Programming area

Based on graphical programming, users can drag and drop the basic modules in the programming area to realize the logic programming of mechanical arm-related process control, which is simple, convenient and very friendly to novice.

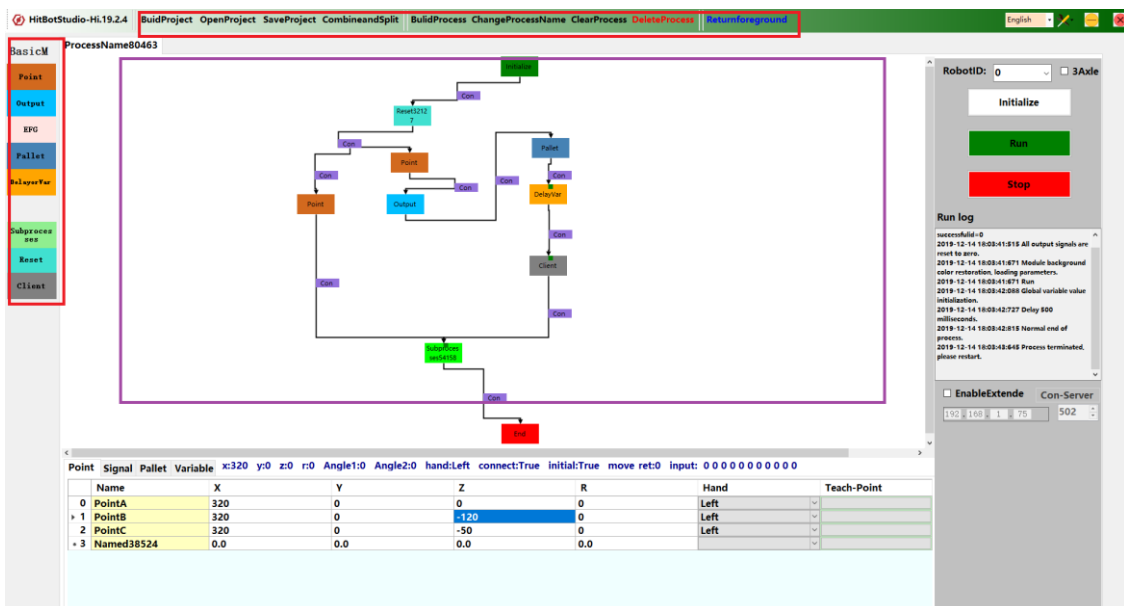
ProcessName80463

Point	Signal	Pallet	Variable	x:320	y:0	z:0	r:0	Angle1:0	Angle2:0	hand:Left	connect:True	initial:True	move	ret:0	input: 0 0 0 0 0 0 0 0	
0	PointA			320		0		0		0						
1	PointB			320		0		-120		0						
2	PointC			320		0		-50		0						
3	Named38524			0.0		0.0		0.0		0.0						

4. Mechanical arm operation

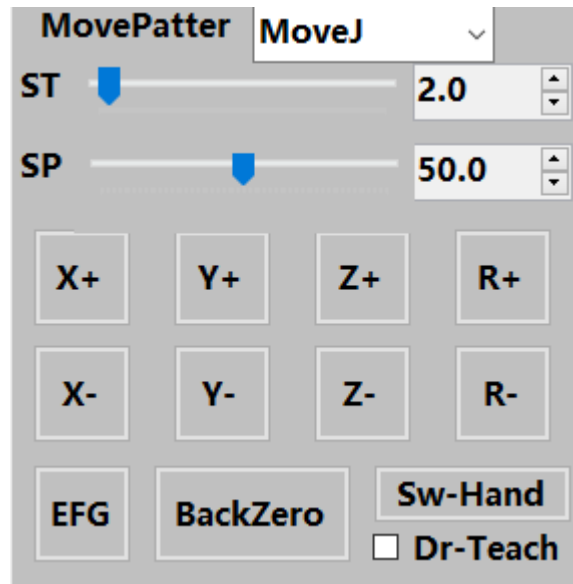
1) Initialize

Create or open an existing project, and the mechanical arm initialization interface will be opened, as shown in the following figure. Click the drop-down box to select the corresponding mechanical arm ID, and then click “Initialize” to wait for the system to pop up the “Initialize Successfully” message box, which indicates that the connection between the computer and the mechanical arm is established successfully and subsequent operations can be performed.



2) Mechanical arm operation

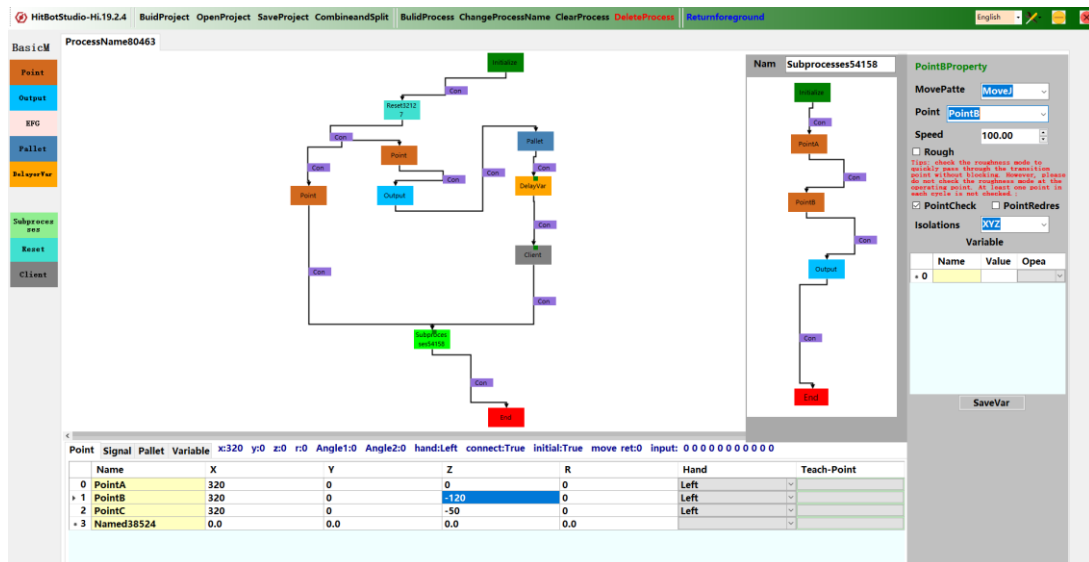
After initialization, right-click “Initialize” to open the arm operation interface as shown. The motion modes are divided into "MoveJ" and "MoveL", the former is curvilinear motion, the latter is linear motion; "ST" denote that step size of the motion, and "SP" denotes the speed of the motion; "EFG" button can perform electric clamping jaw operation; "Force back to zero" may perform a correlation axis regression to that initial position; "Switch hand system" can switch that hand system of the current mechanical arm; "Drag Teaching" can be checked to directly drag the mechanical arm to complete the point teaching.



5. Programming operations

1) Programming example

In order to make it easier for users to better understand the contents of this section and the purpose of each basic module, this section will be explained as a routine demo through the implementation of the application scenario "Cutting and Handling". The following figure shows the programmatic implementation of this scenario. The logic of the routine is as follows: After the routine starts running, the mechanical arm moves first to a "safe waiting position", wait for the trigger signal of the external "blanking signal", when the program detects the signal, it will move to the "picking level", and then grasp the material through the electric clamping jaw closing. After the grasping is finished, the mechanical arm will move to the "safe waiting level" again, and then send the feedback of "blanking completion signal" to the outside, and then put the grasped material into the corresponding position of the "blanking tray". This is used as a cycle until the blanking tray is fully loaded and the process is completed.



2) Conditional operation

Left-click the corresponding module in the programming area to open the rear connection line of the module (i.e. The condition module corresponding to the next connection module); Right-click the corresponding condition module to delete the connector (that is, the condition); Right-click the corresponding condition module to open the attribute editing box of the condition, as shown below. After selecting the required condition, click “Add Condition”. If multiple conditions are juxtaposed, you can select the juxtaposition relationship between “Condition” and/or “Condition”, and then click “Add juxtaposition condition”. If you want to open the else branch of the condition, check "else" and click the condition on the left in the programming area. At present, conditional operation only supports conditional judgment of input signal and self-added variable. The user can add and change input signals by filling in or changing information in the Input Name of the input signal list, or by right-clicking to add signals such as start and stop within the system. " The red color of the Input cell indicates detection of a corresponding input signal.

ConProperty

if

else

input-signal

== 1 And

AddCondition addRelation

Clear-R-to-L

Clear

Message

Point	Signal	Pallet	Variable	x:320	y:0	z:0	r:0	Angle1:0	Angle2:0
	DI	DI-Name							Remarks
0	signal								
1	System IO Run								
2	System IO Stop								
3									
4									
5									
6									
7									

AddSystemInput ▶

AddSystemOutput ▶

Run

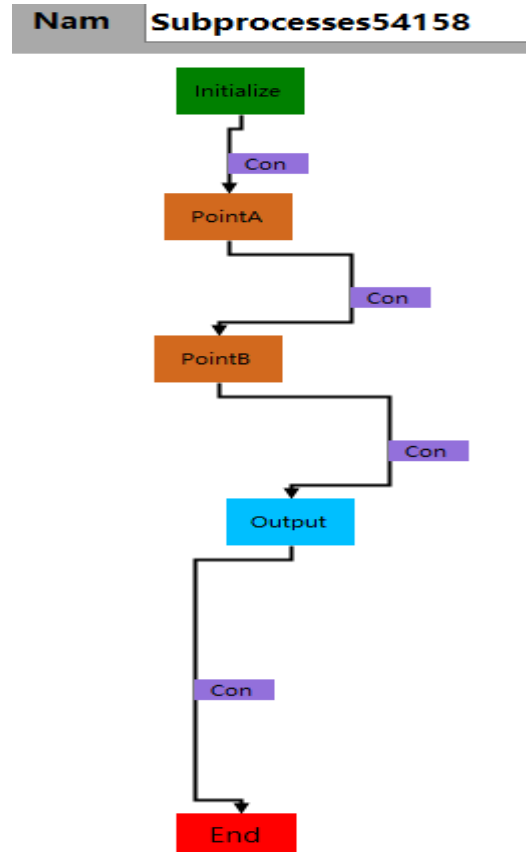
Pause

Stop

3) Subprocess

Right-click the corresponding sub-process module in the programming area to open the sub-process editing box as shown in the following figure. The sub-process module has two main functions: first, it can simplify the main process logic; second, it

provides operators for the reset function.



4) Point

Right-click the appropriate point module to open the point property box, In the attribute box, the movement mode and the movement speed of the point location module can be specified. The coordinate points corresponding to the point location module can also be selected as required for different decomposition actions. "Z-XY" means that the Z-axis is moved in place and then XY is moved, "XY-Z" means that the Z-axis is moved in place and then XYZ is synchronously moved, and "XYZ" means that the XYZ is synchronously moved. Note that you must first add the appropriate dots to the dot list before specifying the dot property of the dot module. The user can successfully add a point by changing the name of the point in the point list, and click the corresponding Teaching Point to pick up the current coordinate as the coordinate of the point. Also added to make up for the point coordinate values of the function of compensation, increase the client, check the level of compensation, the

right mouse button click on the module of point on the right side of the "C" module, set to connect to the server IP address and port number, save and try to connect, after the success can be normal communication, can be manually debug request and process to run automatically.

PointAProperty

MovePatte **MoveJ**

Point **PointA**

Speed **100.00**

Rough

Tips: check the roughness mode to quickly pass through the transition point without blocking. However, please do not check the roughness mode at the operating point. At least one point in each cycle is not checked. :

PointCheck PointRedres

Isolations **XYZ**

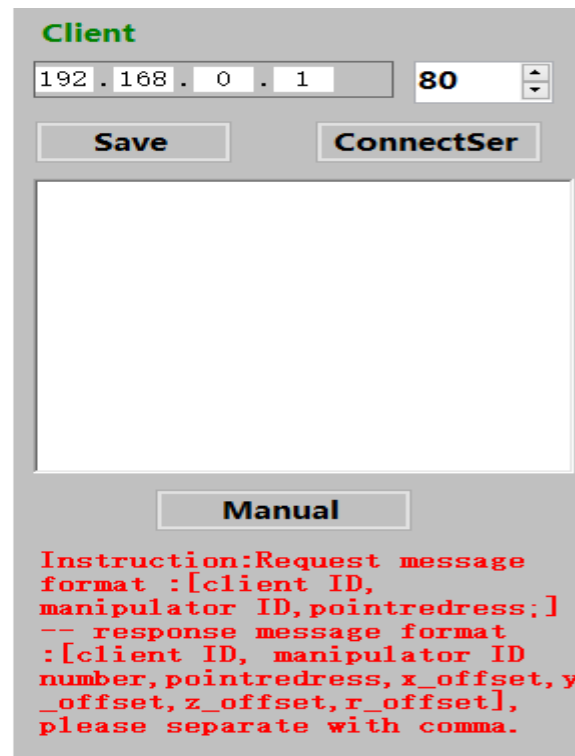
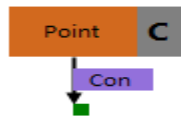
Var **XYZ**

	Name
* 0	

SaveVar

Point Signal Pallet Variable x:320 y:0 z:0 r:0 Angle1:0 Angle2:0 hand:Left connect:False initial:True move ret:0 input: 3 3 3 3 3 3 3 3 3 3

	Name	X	Y	Z	R	Hand	Teach-Point
0	PointA	320	0	0	0	Left	
1	PointB	320	0	-120	0	Left	
2	PointC	320	0	-50	0	Left	
3	Named38524	0.0	0.0	0.0	0.0		



Get the compensation value message format:

Request: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+ pointredress +";"

Response: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+", "+ + pointredress x_offset + y_offset", "+ + +", "+", "+ z_offset r_offset

5) Output signal

Right-click the appropriate output signal module to open the output signal properties box, where the user can choose to add one or more output signals. It should be noted that only relevant signals in the list of output signals are currently supported. The user can add and change the output signal by filling in or changing the information in the "Output Name" of the output signal list. In addition, the user can right-click to add the fault signal inside the system and output a high level to the corresponding fault signal when the process encounters an exception. The user clicks the appropriate setting cell to output or turn off the corresponding output signal.

OutputProperty

out == 1

Delay(ms) 500

Select out == 1

AddSignal DeleteSignal

Message

Variable

	Name	Value	Opea
* 0			

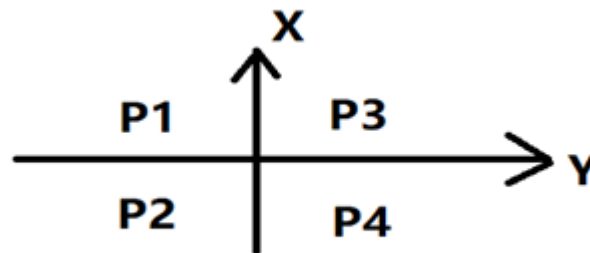
SaveVar

DO	Set	DO-Name	Remarks
		out	

6) Tray

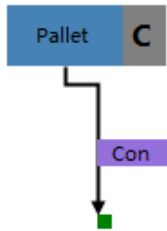
Right-click the corresponding blanking tray module to open the corresponding property box. The tray module is actually an array of dots that can be used when the flow path is an array. As shown in the following figure, the tray execution logic can be divided into two types: periodic incremental execution, which means that the process will execute only one point at a time when it is run to the tray, and one-time execution, which means that all points will be executed when the process is run to the tray. As shown in the following figure, P1, P2, P3, and P4 need to be specified as four points, rows, and handles for the new tray, where P1 and P3 are one row, and P1 and P2 are

one column. Also added to existing tray for compensating point coordinates and pallet ID specified function, increase the client, check the level of compensation, the right mouse button click on the tray module on the right side of the "C" module, set to connect to the server IP address and port number, save and try to connect, after the success can be normal communication, can be manually debug request and process to run automatically.



Point Signal Pallet Variable x:320 y:0 z:0 r:0 Angle1:0 Angle2:0 hand:Left connect:False initial:True move ret:0 input: 3 3 3 3 3 3 3 3 3 3

Point	Signal	Pallet	Variable	x:320	y:0	z:0	r:0	Angle1:0	Angle2:0	hand:Left	connect:False	initial:True	move ret:0	input: 3 3 3 3 3 3 3 3 3 3
0	Pallet	PointA	PointB	PointC	PointD	4	4	Right	Remarks					
1	Named92321					0	0							



Client

192 . 168 . 0 . 1 80

Save ConnectSer

PalletRe PalletArr

ManualRedre ManualArrayl

Instruction: Request message
 format :[client ID,
 manipulator ID,palletredress;]
 -- response message format
 :[client ID, manipulator ID,
 palletredress, x_offset, y_offse
 t, z_offset, r_offset], separate
 with comma.

Get the format of coordinate compensation value message:

Request: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+ palletredress +";"

Response: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+", "+ + palletredress x_offset + y_offset", "+ + +", "+", "+ z_offset r_offset

Get tray ID value message format:

Request: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+ palletArrayID +";"

Response: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+", "+ + palletArrayID row +", "+ column

7) Electric clamping jaw

Right-click the corresponding output electric clamping jaw module to open the corresponding attribute box, where the user can edit the use of the electric clamping jaw, the clamping and releasing of the electric clamping jaw need time to complete, the user can set the appropriate delay time according to the actual effect.

EFG-8(0)Property

EFGType: EFG-8

EFGTrip: 0

Delay(ms): 500

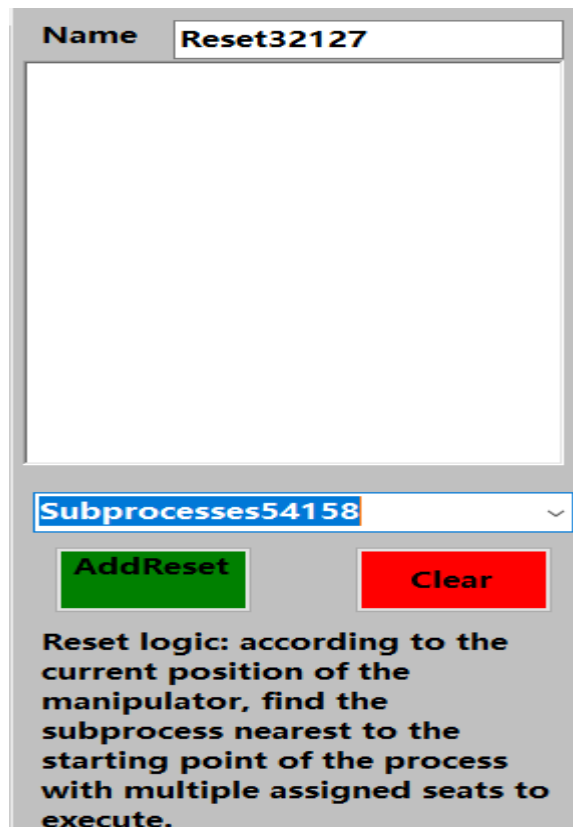
Variable

	Name	Value	Opea
0			

SaveVar

8) Reset

Reset module is generally used after initialization, its role is to start the process will be based on user settings to choose the optimal path back to the safe point, and then execute the process. Right-click the reset module to open the attribute box as shown in the figure. The user can write a sub-process according to the actual bit avoidance path and add it to the reset module operator. When the program executes the reset module, the nearest sub-process path from the current position will be selected to return to the safety point



Name

Reset logic: according to the current position of the manipulator, find the subprocess nearest to the starting point of the process with multiple assigned seats to execute.

9) Client

The client module is used to communicate with other servers to get the necessary parameters to execute. The current version supports the transfer of point coordinates and the transfer of global variable values. The latter is to request the value of the corresponding global variable to be applied to process judgment, process branching, and other logic to reduce the use of IO signals.

Steps: the right mouse button click on the client module, into the property page, set the server IP address and port number to connect, click save to manually connect to the server, if the connection is successful, can be normal communication, now set the client to accept data timeout for 20 seconds, communication so that the manual adjustment, manual send instructions can display the communication data, the server must be in accordance with the response data packet format, otherwise can't normal work.

ClientProperty Enable

192 . 168 . 0 . 1 80

Save ConnectSer

RequestTy PointCoordi ▾

MoveMod MoveJ ▾

Speed 100 ▾

Hand Right ▾

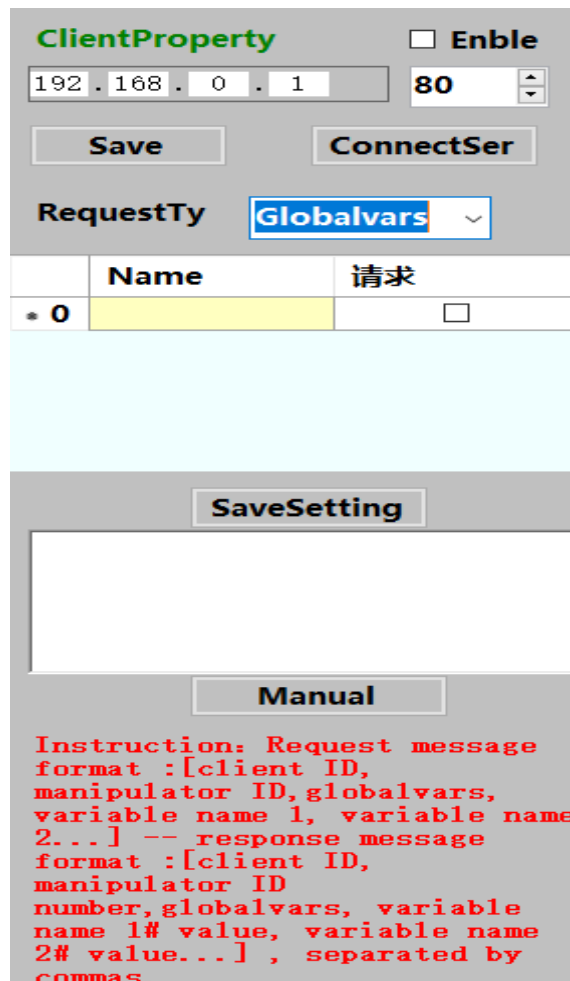
Manual

Instruction: Request message
format :[client ID,
manipulator
ID,pointcoordinates;] - the
response message format:
[client ID, ID number,
mechanical arm
pointcoordinates, x, y, z, r],
pay attention to use commas.

For coordinates, select the appropriate motion mode, speed, and hand system to execute the transmitted coordinates. Message format is as follows:

Request: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+ pointcoordinates +";"

Response: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+ pointcoordinates +", "+ +", "x + y +", "+", "+ r + z



Basis for global variables, the variables create the corresponding variable, fill out the initial value, can see in the client properties corresponding variables, if the "request" check, will send the request of the corresponding variables, support for multiple variable assignment at the same time, can be manually debugging, the process to run automatically, can be in the column of variables monitored view variables to change (pay attention to the "#" connection variable names and values). Message format is as follows:

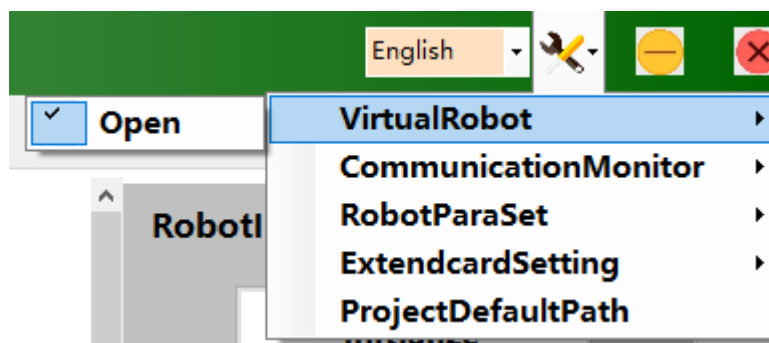
Request: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+ globalvars +", "+ variable name +... + ";"

Response: the head (foreign client module identifier) + ", "mechanical arm + ID +", "+ globalvars +", "# value + + a variable name...

6. System simulation and debugging

1) System simulation

In order to verify the logic correctness of the program, the logic simulation can be carried out by online virtual mechanical arm after the program is completed. As shown in the following figure, the virtual mechanical arm can be opened in the function bar, the appropriate virtual mechanical arm can be selected according to the model, and the virtual mechanical arm can be opened successfully by waiting for the current state of the virtual mechanical arm to be updated to "Connected Successfully". On the startup page, select 0 for the mechanical arm ID, and click Start Run to verify the program logic after successful initialization.



Open VirtualRobot to connect and debug



2) System commissioning

Users can contact technical support personnel for remote assistance if they encounter problems that cannot be solved during the use of the mechanical arm. The technician can complete the abnormal detection and repair through the relevant functions of the status bar.

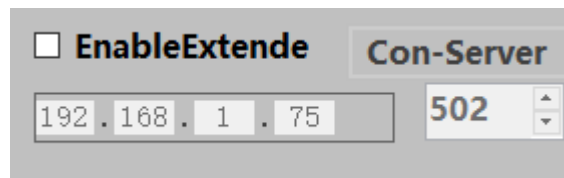
Chapter V Frequently Asked Questions and Solutions

1. Is there a more detailed user manual

A: Watch the “Hitbot Studio Foundation Tutorial Video” or consult technical support.

2. The number of IO ports does not meet the actual requirements

Answer: If the number of robot body IO is insufficient for the actual project, we can purchase the extended IO board supported by our software. At present, the software adapts 16 DO transistor outputs, 16 DI transistor inputs and Ethernet communication board, Model: TCP-517F. Default IP address: 192.168. 1.75, port number: 502. You can modify the IP address and port number according to the requirements through the ExtendcardSetting widget in the upper computer software setting. If you want to establish communication with the arm software, you only need to set the static IP of the Ethernet corresponding to the extension board on the same network segment as the board IP, for example, board: 192, 168.1. 75, PC: 192.168. 1.76. To use the board, check Enable Extended IO Board. If the server is not connected successfully, try connecting to the server manually. The specific configuration is configurable in the signal table and is the same as the mechanical arm body IO.



Taobao link:

<https://item.taobao.com/item.htm?spm=a1z10.5-c.w4002-11584738786.13.61406eda7Jt2Ha&id=529001943100>

4. The computer can't connect to the robot.

Please read the “Software Installation and Uninstallation Instructions” document carefully, set the static IP of local Ethernet to 192.168. 0.100 (mechanical arm host IP), subnet mask: 255.255. 255.0, confirm that the notebook firewall is off, and confirm that the software is running in administrator mode

5. Drag teach function is not available

Turning on and off the drag-and-teach robot servo will have a power-down and power-up process. Please do not

Switch the teaching status frequently. In addition, when you turn on the teaching function and turn off the teaching function after completing the operation, please Wait for about 10s before doing so. If this problem occurs, try turning off the teaching function, replay the

Can the new initialization robot solve. Note that the Z-axis does not enable the teaching function.

6. User manual-related interface does not match the actual interface of the software

A: This manual is based on Hi.19. 2.2. Please consult technical support for updates.

Chapter VI Appendix

1. HITBOT Z-Arm 1632 Mechanical arm Interface Panel Instructions
2. HITBOT Z-Arm 2140 Mechanical arm Interface Panel Instructions
3. EFG Series Electric Clamping Jaw Data



Huiling-tech Robotic Co., Ltd

Tel 0755-36382405

Email hitbot@hitbot.cc

Website www.z-arm.com

Address 1st floor, 7 Building, Robot Intelligence and Manufacturing Industrial Park,
Dayang Road 90, Baoan District, Shenzhen, Guangdong, PR China.

