

STEP Robot Welding Instruction Manual

Publication Status: Standard

Revision: V1.0

ADTECH TECHNOLOGY CO., LTD. is fully responsible for the development and print of this control system user operation and programing guide.

© Copyright. All Rights Reserved.

Without prior permission from ADTECH TECHNOLOGY CO., LTD., no information in this document (software etc.) may be extracted, reproduced (in whole or in part) and transmitted (including information and publication) in any forms.

All rights reserved and the violators will be prosecuted. The content is subject to change without prior notice.

All Copyright © reserved by ADTECH TECHNOLOGY CO., LTD., China

ADTECH TECHNOLOGY CO., LTD. authorizes in the documenting, printing to this robot control manual.

All rights reserved.

The information in this document is subject to change without prior notice. No part of this document may in any form or by any means (electronic, mechanical, micro-coping, photocopying, recording or otherwise) be reproduced, stored in a retrial system or transmitted without prior written permission from ADTECH TECHNOLOGY CO., LTD.

Foreword

Abstract

This manual provides complete and comprehensive description about STEP robot welding function. This manual can be used as the guidance for welding function of STEP robot.

In order to ensure proper use the welding function of STEP robot, be sure to read this manual carefully before using the robot.

Target Readers

Operator

Robot programmer

Engineering maintenance personnel

Technical support personnel for users

Contents

Contents in this manual may be supplemented and modified, please visit our website to update your manual. Our website: www.steprobots.com.

Main features

The screen of STEP robot teach pendant adopts hierarchical and classified management type, the users could control the robot with the physical buttons on the teach pendant and the virtual hotkeys on the touchscreen. The operation screen is concise, and the using method conforms to human senses, it's easy to understand.

Descriptions of safety-related marks

In this operation manual, the contents relating to safety will apply the following marks. Descriptions and contents with safety mark are important, please be sure to observe them.



Danger

It may cause hazardous conditions or personal death if it is used improperly.



Caution

It may cause danger, minor or serious personal injury and equipment damage if it is used improperly.



Important

The part that the user needs to observe and pay attention.

Chapter I Description

Chapter II Safety

Chapter III Installation and configuration

Chapter IV Basic welding function

Chapter V Advanced welding function

Table of Contents

Chapter 1 Description.....	7
1.1 Purpose.....	7
1.2 Target groups.....	7
1.3 Other relevant documents.....	7
1.4 Precautions.....	7
Chapter 2 Safety.....	8
2.1 Safety protective device of the robot.....	8
2.1.1 Preview of the safety protective device.....	8
2.1.2 Emergency switch-off key.....	8
2.1.3 Selection switch for of operation mode of run.....	8
2.1.4 Automatic runningJogging operation.....	9
2.1.5 Mechanical end stop.....	9
2.1.6 Software limit switch.....	9
2.2 Relevant personnel.....	9
2.3 Training.....	10
2.4 Safety measures.....	11
Chapter III Installation and Configuration.....	13
3.1 Welding function installation.....	13
3.2 Welding function configuration.....	13
3.2.1 PLC configuration.....	13
3.2.2 Welding configuration.....	16
Chapter IV Basic Welding Functions.....	19
4.1 Variable.....	19
4.1.1 ARCONDATA.....	19
4.1.2 ARCOFFDATA.....	19
4.1.3 ARCDATA.....	19
4.2 Statement.....	19
4.2.1 ARCON.....	19
4.2.2 ARCOFF.....	20
4.2.3 ARCSET.....	20
4.2.4 WLin.....	20
4.2.5 WCirc.....	20
4.2.5 ARCMODE.....	21
4.2.5 ARCJOB.....	21
4.3 Programming example.....	21
Chapter V Advanced Welding Functions.....	23
5.1 Weave.....	23
5.1.1 Coordinate system of weave.....	23
5.1.2 Variable.....	23
5.1.3 Statement.....	23
5.1.4 Example.....	25
5.2 Track.....	25
5.2.1 Variable.....	25
5.2.2 Statement.....	25
5.2.3 Example.....	26
5.3 Search.....	26
5.3.1 Variable.....	26
5.3.2 Statement.....	28
5.3.3 Example.....	29
5.4 MultiLayer.....	31

5.4.1 Variable.....	31
5.4.2 Statements.....	33
5.4.3 Example.....	34
5.5 Precautions.....	35
Technical support.....	37

Chapter 1 Description

1.1 Purpose

The document is aimed to describe the function, operation method and notice for use of welding. The document can help users quickly learn how to use the feature pack of welding.

1.2 Target groups

The audience of the document is: robot operator, programmer, maintenance personnel, robot integrator. The following is other skills required by target groups.

Table 1.1 Target group and required skills

Target group	Required skills
Operator	Participate in the robot training and get qualified certificate, be familiar with robot operation.
Programmer	Have robot programming basis, be familiar with robot function.
Maintenance personnel	Be familiar with robot function and electrical schematic.
Integration personnel	Be familiar with robot function and application.

1.3 Other relevant documents

STEP Robot Operating Software Instruction Manual

1.4 Precautions

This manual mainly introduces the feature pack of welding, for the details of the basic robot operation, please refer to the relevant document.

Chapter II Safety

2.1 Safety protective device of the robot

2.1.1 Overview

The following safety protective devices are provided on robot system:

- Emergency switch-off key
- Selection switch of operation mode
- Jogging operation
- Mechanical end stop
- Software limit switch

Note: when the safety protective device is dismantled or stopped, the robot system is forbidden to run.

2.1.2 Emergency switch-off key

The emergency stop button is located on the control panel of the teach pendant. When the button is pressed, the robot's drive is shut down immediately.



Danger

The emergency stop button shall be pressed immediately if any accident occurs that will endanger personnel or equipment. To resume the operation, turn the emergency stop button to unlock it and acknowledge the shut-down information.

2.1.3 Selection switch of operation mode

There are 3 operation modes for the robot: fast manual operation (T2), automatic operation (AUT) and automatic external operation (AUT EXT).

The operation mode is selected using the key switch on the panel. If the operation mode is changed during the movement of the robot, the drive is disrupted immediately.

Table 2.1 Robot Operation Mode

Operation mode	Application	Speed
T2	For testing operation.	Programming running: without speed limit requirement; Jogging operation: max. speed is 250mm/s
AUT	For the robot system without higher level control system	Programming running: without speed limit requirement; Jogging operation: unable to run
AUT EXT	For the robot system with higher level control system (such as PLC)	Programming running: without speed limit requirement; Jogging operation: unable to run

2.1.4 Jogging operation



Important

When a program is developed, the program teaching and debugging shall be done in manual mode and no error occurs before it can be run in automatic mode. It is called jogging operation when the program is run in manual mode.

Difference between automatic operation program and jogging operation program:

- In automatic running mode, press “Start” button to run the program. In manual mode, press and hold “Start” button to run the program. The robot stops running when the “Start” button is released.
- It is safer to run the program in manual mode as there is stricter limit on the running speed.

2.1.5 Mechanical end stop

The basic axes A1, A2, A3, A5 and A6 are provided with the mechanical end stop with buffer.

2.1.6 Software limit switch



Caution

The movement range of all the axes of the robot can be limited by the defined software limit switch. The software limit switch only serves as mechanical safety device and is so defined that the robot will not collide with the mechanical end stop.

2.2 Relevant personnel



Important

All the working personnel relating to the robot control system shall read and be familiar with the documents on the safety of the robot system.

Before the work, the working method, range and potential hazard shall be introduced to the working personnel. Describe them again after accident or technical update.

The relevant personnel include the system integrator that integrates the robot system into the equipment, user, operator or the programmer of the robot system.

2.3 Training

The user that uses the robot and the robot system shall ensure that its programmer, operator and maintenance personnel have participated in the safety training, and acquired corresponding capabilities to undertake the work. For the training, it's better to combine classroom courses with practical operation.

Objective:

The goal of the training is to help the trainees understand the following information:

- 1) Usage and function of safety components;
- 2) Procedures concerning health and safety;
- 3) All danger caused by the operation of robot and robot system;
- 4) Task and purpose related with specific robot;
- 5) Basic safety concept.

Requirement:

- 1) Learn applicable safety procedures and safety suggestions provided by the robot manufacturer and the robot system designer;
- 2) Understand the clear meaning of the task assigned;
- 3) Master the identification and description of all control units and their functions that are used to complete the assigned task, such as low speed control, teach box operation, emergency stop procedure, switch-off procedure, single point control and etc.;
- 4) Identify the danger related to the task, including the danger caused by the auxiliary equipment;
- 5) Identify the safety protective devices, including the type, capability or selection scheme of the safety protective devices, function of the components selected, functional test method of the components, limit of the components selected and the safety operation procedures since identifying danger, personnel protection equipment etc.;
- 6) Master the test method to ensure the normal function of the safety protective device and interlocking units;

Re-training requirement:

In case system change, people change or accident occurs, in order to ensure safe operation, relevant personnel shall participate in training again.

2.4 Safety measures



Caution

The robot system can only be operated with the equipment functioning properly and with proper safety awareness. Improper application may lead to personal injury and equipment damage.

Even if the robot control system is shut off with safety protection, there still could be movement of the robot system. Improper installation (such as overload) or mechanical damage (such as brake failure) could cause settlement of the robot or auxiliary axis. If the work is performed on the robot system that has been shut off, the robot and the auxiliary axis shall move to a state where they will not move on their own with or without load. If the above conditions cannot be met, proper protection shall be provided for the robot and auxiliary axis.

Execute the following when the robot system has any fault:

- Shut off the robot control system and protect it well, to prevent unauthorized restart.
- Indicate the fault by the nameplate with corresponding prompts.
- Record the fault.
- Troubleshoot and have functional check.

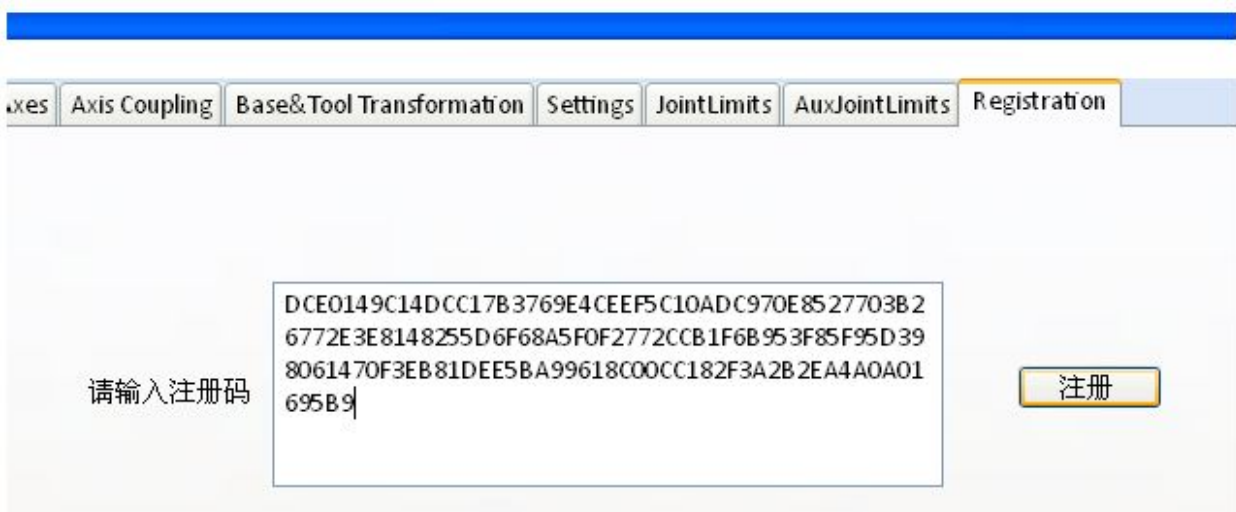
Chapter III Installation and Configuration

3.1 Welding function installation

Installation steps are as follows:

- 1) Please apply the registration code with welding function from the equipment manufacturer.
- 2) Connect the control cabinet with the PC (IP is 192.168.0.X, X is nonzero) running robot configuration software (RobotConfig);
- 3) Open the registration screen of configuration software, input the registration code in step 1, then click the robot software with welding function to register.

Registration schematic is shown as the following:



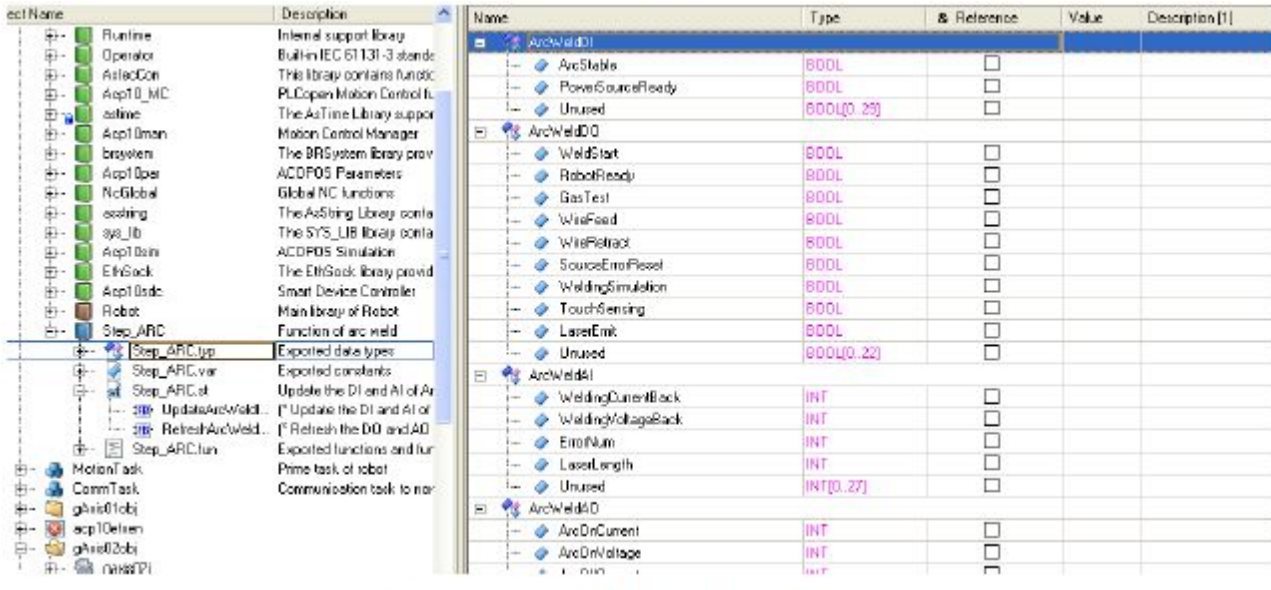
Note: Different registration codes should be provided for different welding functions.

3.2 Welding function configuration

3.2.1 PLC configuration

PLC configuration steps are as follows:

- 1) Add Step_ARC library in PLC



2) Define the global variable of welding data structure in Step_ARC library.

gArcDI	ArcWeldDI			(0)
gArcDO	ArcWeldDO			(0)
gArcAI	ArcWeldAI			(0)
gArcAO	ArcWeldAO			(0)

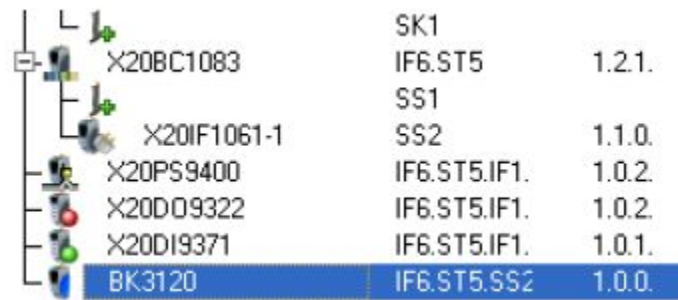
3) Call the function blocks UpdateArcWeldInput and RefreshArcWeldOutput of Step_ARC in MotionTask. Data of the two function blocks apply the global variable set up in step 2. It is able to call UpdateArcWeldInput in UserPreAction, and RefreshArcWeldOutput in UserPostAction.

Note: Because the global variable is used in UpdateArcWeldInput and RefreshArcWeldOutput, 25314 boot error incurred may be reported when PLC is downloaded in future. At this time, it only requires ARWin hot start.

```
(* Arc welding Input *)
UpdateArcWeldInput_0.ArcDI := gArcDI;
UpdateArcWeldInput_0.ArcAI := gArcAI;
UpdateArcWeldInput_0.Enable := TRUE;
UpdateArcWeldInput_0();
```

```
(* Arc welding Output *)
RefreshArcWeldOutput_0.ArcDO := ADR(gArcDO);
RefreshArcWeldOutput_0.ArcAO := ADR(gArcAO);
RefreshArcWeldOutput_0.Enable := TRUE;
RefreshArcWeldOutput_0();
```

4) The module having communication between the configuration and the welding machine may be IO module or field bus module. Profibus bus using BK3120, shown as the figure.



- 5) If IO signal is used for communication with the welding machine, the corresponding signal of the welding structure only is mapped on IO module.
- 6) If field bus is used for communication with the welding machine, an additional PLC program must be written, to correspond the data on the bus with that in the welding data variables one to one. As shown in the figure, it is a program named AWTest, firstly the data on the bus is mapped on the variables ArcIn and ArcOut in AWTest.

➤ Module001_Input_1	USINT	Automatic	AWTest.ArcIn[0]
➤ Module001_Input_2	USINT	Automatic	AWTest.ArcIn[1]
➤ Module001_Input_3	USINT	Automatic	AWTest.ArcIn[2]
➤ Module001_Input_4	USINT	Automatic	AWTest.ArcIn[3]
➤ Module001_Input_5	USINT	Automatic	AWTest.ArcIn[4]
➤ Module001_Input_6	USINT	Automatic	AWTest.ArcIn[5]
➤ Module001_Input_7	USINT	Automatic	AWTest.ArcIn[6]
➤ Module001_Input_8	USINT	Automatic	AWTest.ArcIn[7]
➤ Module001_Input_9	USINT	Automatic	AWTest.ArcIn[8]
➤ Module001_Input_10	USINT	Automatic	AWTest.ArcIn[9]
➤ Module001_Input_11	USINT	Automatic	AWTest.ArcIn[10]
➤ Module001_Input_12	USINT	Automatic	AWTest.ArcIn[11]
➤ Module001_Input_13	USINT	Automatic	AWTest.ArcIn[12]
➤ Module001_Input_14	USINT	Automatic	AWTest.ArcIn[13]
➤ Module002_Output_1	USINT	Automatic	AWTest.ArcOut[0]
➤ Module002_Output_2	USINT	Automatic	AWTest.ArcOut[1]
➤ Module002_Output_3	USINT	Automatic	AWTest.ArcOut[2]
➤ Module002_Output_4	USINT	Automatic	AWTest.ArcOut[3]
➤ Module002_Output_5	USINT	Automatic	AWTest.ArcOut[4]
➤ Module002_Output_6	USINT	Automatic	AWTest.ArcOut[5]
➤ Module002_Output_7	USINT	Automatic	AWTest.ArcOut[6]
➤ Module002_Output_8	USINT	Automatic	AWTest.ArcOut[7]
➤ Module002_Output_9	USINT	Automatic	AWTest.ArcOut[8]
➤ Module002_Output_10	USINT	Automatic	AWTest.ArcOut[9]
➤ Module002_Output_11	USINT	Automatic	AWTest.ArcOut[10]
➤ Module002_Output_12	USINT	Automatic	AWTest.ArcOut[11]
➤ Module002_Output_13	USINT	Automatic	AWTest.ArcOut[12]
➤ Module002_Output_14	USINT	Automatic	AWTest.ArcOut[13]
➤ Module003_Output_15	USINT		

System structure of PLC program

Then assign to the global variable after transfer.

(* DI *)

```
gArcDI.ArcStable := USINT_TO_BOOL(ArcIn[1] AND 1);
gArcDI.PowerSourceReady := USINT_TO_BOOL(ArcIn[1] AND 32);
```

(* DO *)

```
ArcOut[0] := SHL(BOOL_TO_USINT(gArcDO.GasTest), 0) +
SHL(BOOL_TO_USINT(gArcDO.WireFeed), 1) +
SHL(BOOL_TO_USINT(gArcDO.WireRetract), 2) +
SHL(BOOL_TO_USINT(gArcDO.SourceErrorReset), 3) +
SHL(BOOL_TO_USINT(gArcDO.TouchSensing), 4);
```

```
ArcOut[1] := SHL(BOOL_TO_USINT(gArcDO.WeldStart), 0) +
SHL(BOOL_TO_USINT(gArcDO.RobotReady), 1);
```

```
ArcOut[2] := SHL(BOOL_TO_USINT(gArcDO.WeldingSimulation), 7);
```

(* AI *)

```
gArcAI.ErrorNum := USINT_TO_INT(ArcIn[0]);
tempDINT := ArcIn[4];
gArcAI.WeldingVoltageBack := DINT_TO_INT(SHL(tempDINT, 8) + USINT_TO_DINT(ArcIn[5]) - 32768);
tempDINT := ArcIn[6];
gArcAI.WeldingCurrentBack := DINT_TO_INT(SHL(tempDINT, 8) + USINT_TO_DINT(ArcIn[7]) - 32768);
```

(* AO *)

```
(* power *)
tempDINT := INT_TO_DINT(gArcAO.WeldingCurrent);
ArcOut[4] := DINT_TO_USINT(SHR(tempDINT + 32768, 8) AND 255);
tempDINT := INT_TO_DINT(gArcAO.WeldingCurrent);
ArcOut[5] := DINT_TO_USINT((tempDINT + 32768) AND 255);
```

(* arc length correct *)

```
tempDINT := INT_TO_DINT(gArcAO.WeldingVoltage);
ArcOut[6] := DINT_TO_USINT(SHR(tempDINT + 32768, 8) AND 255);
tempDINT := INT_TO_DINT(gArcAO.WeldingVoltage);
ArcOut[7] := DINT_TO_USINT((tempDINT + 32768) AND 255);
```

```
ArcOut[9] := INT_TO_USINT(gArcAO.BurnBackTime);
```

PLC program code

3.2.2 Welding configuration

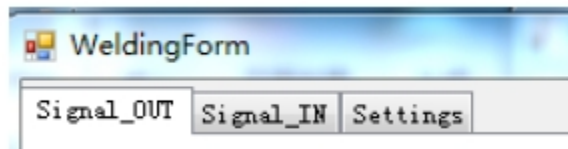
Welding configuration steps are shown as follows:

- 1) Click View in the upper left corner of the robot configuration software to open the welding configuration screen.



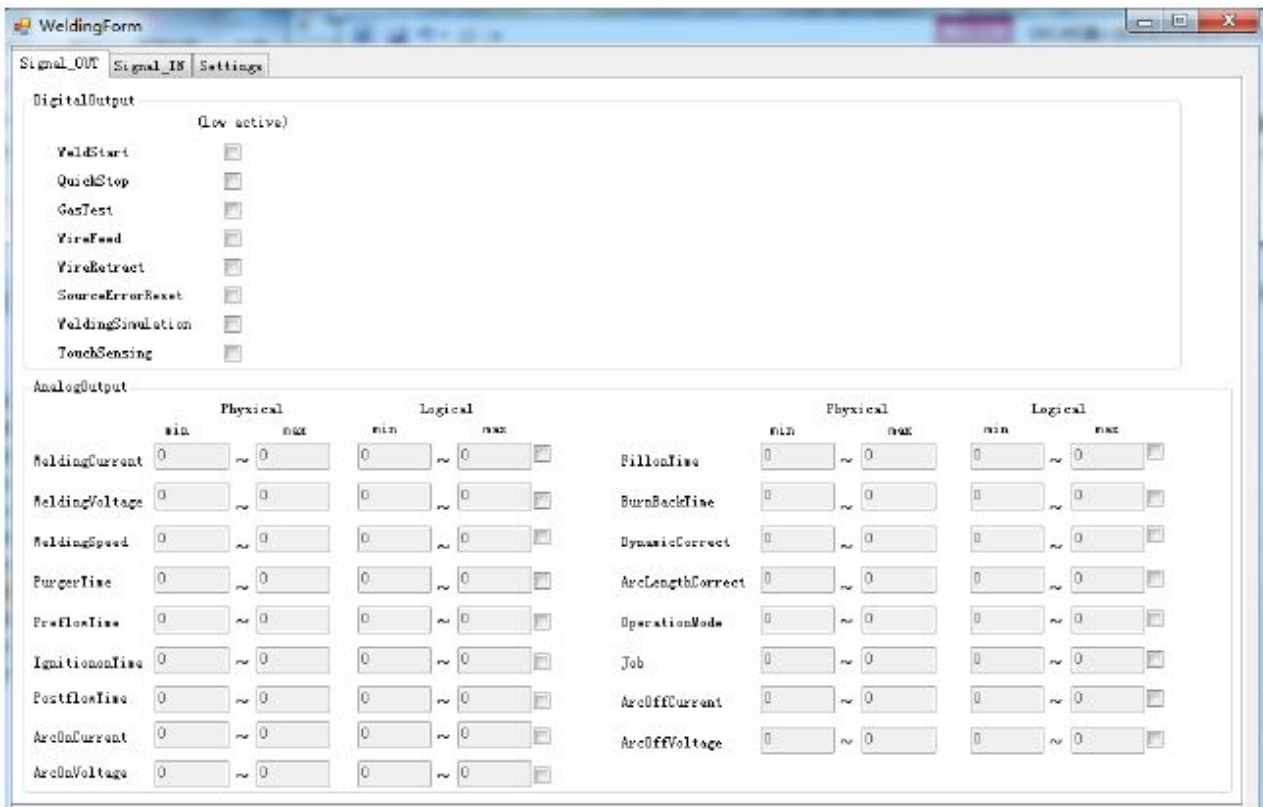
Welding configuration content

- 2) Welding configuration screen is divided into 3 blocks, namely output signal, input signal and setting. Of which, set search in Setting.



Welding configuration screen

- 3) Output signal includes digital output and analog output. The upper part is digital output and the lower part is analog output.

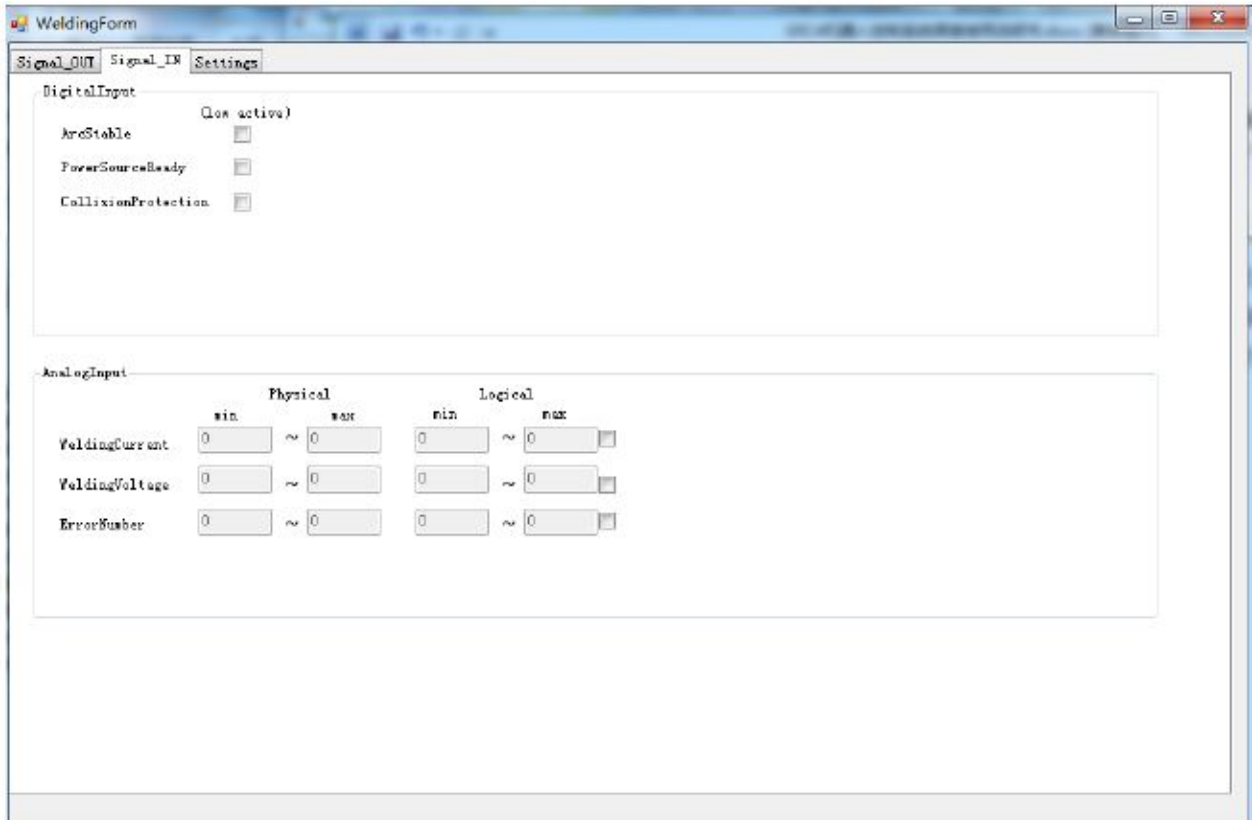


Digital output and analog output

Digital output default is high level output, namely high level will be output when this bit is valid. If it requires being modified as low level output, then check the checkbox beside this digital signal.

If some analog signals will be used, check the checkbox beside it, then it is able to set the scope of its physical value and logic value. Scope of the physical value is determined by the welding machine, location of the logic value is determined by the resolution of analog module (it is based on the welding machine for communication mode of the bus).

- 4) Input signal is similar to output signal, the upper part is digital input and the lower part is analog input.



Digital input and analog input

Chapter IV Basic Welding Functions

This chapter introduces the basic welding functions, including variable and statement.

4.1 Variable

4.1.1 ARCONDATA

Arcon variable data is composed of preflow_time, arcon_current, arcon_voltage and arcon_time.

4.1.2 ARCOFFDATA

Arcoff variable data is composed of Burnback_time, Postflow_time, arcoff_current, arcoff_voltage and arcoff_time.

4.1.3 ARCDATA

Welding variable data is composed of weld voltage and weld current.

4.2 Statement

4.2.1 ARCON

ARON is used to set arcon parameter, which applies an ARCONDATA type variable as the parameter.

Application method of ARCON:

```
ARCON(arcondata)
```

4.2.2 ARCOFF

ARCOFF is used to set arcoff parameter, which applies an ARCOFFDATA type variable as the parameter.

Application method of ARCOFF:

```
ARCOFF(arcoffdata)
```

4.2.3 ARCSET

ARCSET is used to set weld parameter, which applies an ARCDATA type variable as the parameter. Application method of ARCSET:

```
ARCSET(arcset)
```

4.2.4 WLin

WLin is dedicated to control the robot to walk in a straight line during welding, which only applies to welding function and only be used between ARCON and ARCOFF. WLin parameters contain POSITION, DYNAMIC, OVERLAP, ORITYPE, WEAVEDATA and TRACKDATA.

Application method of WLin:

```
WLin(pos, dynamic, overlap, oritype, weavedata, trackdata)
```

Note: If there isn't WEAVEDATA or TRACKDATA in registration code of the robot, the two parameters followed by WLin can't be used.

4.2.5 WCirc

WCirc is used to control the robot to walk in a circular arc during welding, which only applies to welding function and only be used between ARCON and ARCOFF. WCirc parameters contain POSITION, DYNAMIC, OVERLAP, ORITYPE, WEAVEDATA and TRACKDATA.

Application method of WCirc:

```
WCirc(pos, pos, dynamic, overlap, oritype, weavedata, trackdata)
```

Note: If there isn't WEAVEDATA or TRACKDATA in registration code of the robot, the two parameters followed by WCirc can't be used.

4.2.5 ARCMODE

ARCMODE is used to set weld mode of the welding machine (automatic, manual). The statement has the parameter UINT (weld mode).

Descriptions:

- In UNIT type parameters, 0, 1....are related to the welding machine, the user may refer to the welding machine manual used.
- If weld mode is set as manual, use ARCSET to set welding parameter, ARCJOB will be invalid; if weld mode is set as automatic, use ARCJOB to set welding parameter, ARCSET will be invalid.

4.2.5 ARCJOB

ARCJOB applies JOB of the welding machine to set welding parameter, this statement uses a UNIT type variable as parameter.

Application mode of ARCJOB:

```
ARCJOB(uint)
```

Descriptions:

- For the specific parameters of welding machine such as current and voltage, welding machine panel is necessary to find the corresponding JOB for adjustment.
- Only the weld mode MODE is set as automatic, can ARCJOB be used to set welding parameter.

4.3 Programming example

```
Tool(tool0);           //set tool
ARCSET(arcdata0);
//set welding parameters(including welding current, welding voltage)
Lin(mid,dyn0);//walk to some point mid nearby the weld pass;
Lin(rap0,dyn0);//walk to arcon point rap0;
ARCON(arcondata0);
//arcon,see the manual for the related arcondata parameters;
```

```

WLin(rap1,dyn1);
//execute straight line welding in the speed parameter in dyn1, destination of this weld pass is rap1;
WLin(rap2,dyn1);
//execute straight line welding in the speed parameter in dyn1, destination of this weld pass is rap2;
WCirc(rap3,rap4,dyn2);
//Execute arc welding in the speed parameter in dyn2, rap3 is its midpoint, rap4 is its terminal point.
Track of the circular arc is determined by the starting point rap2, midpoint rap3 and terminal point rap4.
ARCOFF(arcoffdata0); //arcoff;
Lin(mid,dyn0); //lift the welding gun;
EOF

```

Precautions

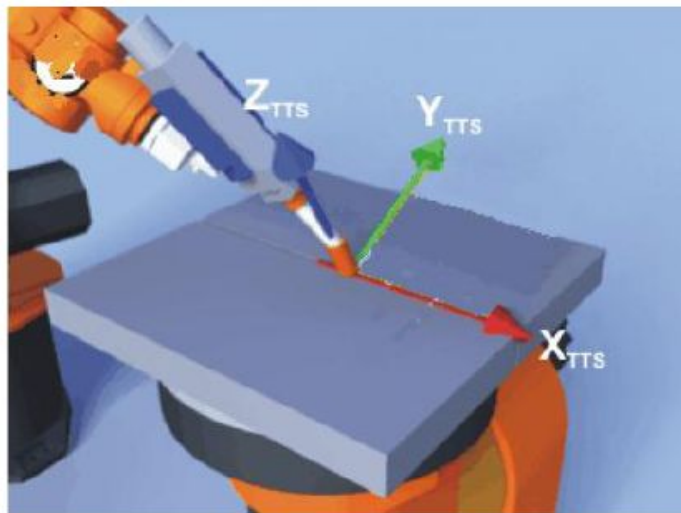
- WLin and WCirc are the special motion statements for welding function, which are used between ARCON and ARCOFF.
- PTP, Lin and Circ can't be used between ARCON and ARCOFF.
- If the robot pauses during execution of the welding statements between ArcOn and ArcOff, then the arc will directly extinguish. The arc won't start again when the robot moves again, unless manually switch SetPC to ARCON for re-execution.
- Welding machine error won't be cleared in the screen or software, which will be cleared after the welding machine is started. If there is welding machine error during welding, please add welding machine error processing program in PLC program.
- Before the actual welding, idling is recommended, to confirm whether the welding procedure is correct, so as to avoid unnecessary danger.

Chapter V Advanced Welding Functions

5.1 Weave

5.1.1 Coordinate system of weave

The robot will take reference from TTS coordinate system during weave, which takes the direction of welding line as its positive direction X, and XZ plane is determined by X and welding gun, so as to determine the direction Y according to the rule that axis Y is vertical to XZ plane, therefore obtain the direction Z via the right hand rule.



5.1.2 Variable

Weave function is corresponding to the parameter WeaveData, this structure variable is composed of weave_type, weave_shape, weave_length, weave_width and weave_angel.

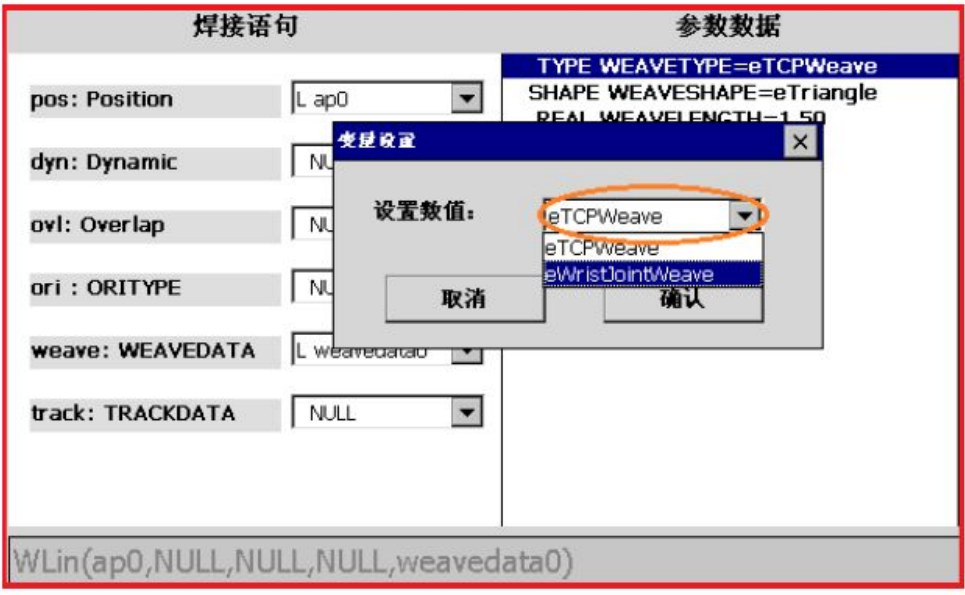
5.1.3 Statement

Weave parameters shall be used in cooperation with Win and Wcirc during its application, to realize its operation in the process of welding.

It is able to choose to weave or not, or realize the different weave effects by setting various parameters in WeaveData.

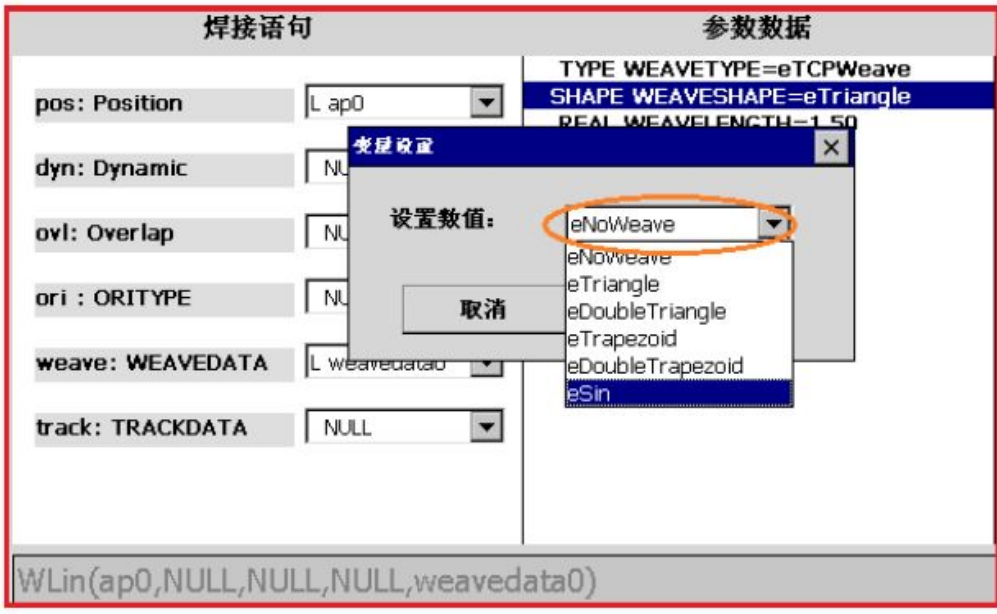
Weave includes wrist weave and TCP weave, please note that X, Y and Z may not be zero at the same time for wrist weave.

WLin(pos, dynamic, overlap, oritype, weavedata, trackdata)



Weave choose screen

Weave shape includes sin, triange, double triangle, trapezoid and double trapezoid.



Weave shape choose screen

Descriptions:

- Weave length is recommended to ensure the weave frequency below 3hz.
- sin is recommended for the weave shape
- 0° is recommended for the angle of weave plane

- TCP is recommended for the weave shape. Wrist weave can't guarantee the route precision and it is possible not to arrive at the route.

5.1.4 Example

Function of weave is simple, with its example as follows:

```
/**Application example of weave***/  
PTP(ap1);  
WLin(CP1, NULL, NULL, NULL, weavedata0, Null);  
//Take ap1 as the starting point of this statement, which is welded to cp1 in a straight line.  
//Take Weavedata0 as weave parameter, whose parameter setting is determined by the user.
```

5.2 Track

5.2.1 Variable

Track the corresponding parameter TrackData, this type of structure variable is composed of Horizontal Gain and Vertical Gain.

Set various track parameters to obtain the required tracing effects.

5.2.2 Statement

Track shall be used in cooperation with WLin or Wcirc, provided with the relevant parameters in statement setting screen.

```
WLin(pos, dynamic, overlap, oritype, weavedata, trackdata)
```

焊接语句		参数数据
pos: Position	L ap0	REAL HORIZONTALGAIN=0.00 REAL VERTICALGAIN=0.00
dyn: Dynamic	NULL	
ovl: Overlap	NULL	
ori : ORITYPE	NULL	
weave: WEAVEDATA	NULL	
track: TRACKDATA	L trackdata0 NULL L trackdata0	
WLin(ap0,NULL,NULL,NULL,NULL,trackdata0)		

Track parameters choose screen

5.2.3 Example

Track function is simple, with its example as follows:

```

//*****application example*****
PTP(ap1);
WLin(CP1, NULL, NULL, NULL, NULL, trackdata0);
//Make cp1 deviate from the expected welding line during its demonstration, realize track function by
//adding trackdata0 in WLin, so as to make the actual welding route to be the expected welding line.
//trackdata parameter setting is determined by the user.

```

Note: It must weave when arc track is applied.

5.3 Search

Search is applied to the conditions needing dynamic positioning during running of the robot. Firstly the robot determines the reference position of the work piece after the first search. When the work piece has any deviation, calculate its position after offset according to the offset value obtained from the second search. Based on different offset dimensions of the work piece, search is divided into one-dimensional search, two-dimensional search, three-dimensional search, six-dimensional search and relevant search.

5.3.1 Variable

1. CD

CD variable data is composed of Port (port number, UNIT type) and Calibrate (whether set the correction reference data).

2. PAT

Weave variable data is composed of TouchMode, Tolerance, SearchDistance, SearchSpeed and ReturnAcc.

Descriptions:

- TouchMode takes one of eSingle Touch (single touch search) and eDoubleTouch (double touch search, don't use it now).
- Tolerance introduces the distance in front and behind the expected point in space (determine by search distance and current position). If the point searched by the robot isn't within the scope, search will be invalid (the robot will report an error).

焊接语句	参数数据
pos: Position <input type="text" value="L via1"/>	<pre>TOUCHMODE TOUCHMODE=eSingleTouch LREAL TOLERANCE=200.00 LREAL SEARCHDISTANCE=200.00 LREAL SEARCHSPEED=30.00 ULINT RETURNACC=20 UINT RETURNSPEED=80</pre>
cd: CD <input type="text" value="L cd1"/>	
pa: PA <input type="text" value="L pat0"/>	
Search_T(via1,cd1,pat0)	

3. PAL

Weave variable data is composed of TouchMode, Tolerance, SearchDistance, SearchSpeed, ReturnAcc and ReturnSpeed.

Descriptions:

- TouchMode takes one of eSingle Touch (single touch search) and eDoubleTouch (double touch search, don't use it now).
- Tolerance introduces the distance in front and behind the expected point in space (determine by search distance and current position). If the point searched by the robot isn't within the scope, search will be invalid (the robot will report an error).

5.3.2 Statement

1. Search_T

Search_T is to look for the position of work piece along the direction determined by the current point and the midpoint (statement parameter), and store the result in CD type parameter variable. Search_T parameters include POSITION (midpoint, to determine the direction, CD (store search result) and PAT (search parameter).

Application method of Search_T:

```
Search_T(viaPos, cd0,pat0);
```



Search_T setting screen

2. CORR_1D

As a one-dimensional search setting statement, CORR_1D is used in cooperation with CORR_OFF, with its actuating range of the intermediate statement between CORR_1D and CORR_OFF. CORR_1D parameter includes CD (search data).

Application method of CORR_1D:

```
CORR_1D(cd0);
```

3. CORR_2D

As a two-dimensional search setting statement, CORR_2D is used in cooperation with CORR_OFF, with its actuating range of the intermediate statement between CORR_2D and CORR_OFF. CORR_2D parameters include CD (search data) and CD (search data).

Application method of CORR_2D:

```
CORR_1D(cd0,cd1);
```

4. CORR_3D

As a three-dimensional search setting statement, CORR_3D is used in cooperation with CORR_OFF, with its actuating range of the intermediate statement between CORR_3D and CORR_OFF. CORR_3D parameters include cd0, cd1 and cd2.

Application method of CORR_3D:

```
CORR_1D(cd0,cd1,cd2);
```

5. CORR_6D

As a six-dimensional search setting statement, CORR_6D is used in cooperation with CORR_OFF, with its actuating range of the intermediate statement between CORR_6D and CORR_OF. CORR_3D parameters include cd0, cd1, cd2, cd3, cd4 and cd5.

Application method of CORR_6D:

```
CORR_1D(cd0,cd1,cd2,cd3,cd4,cd5);
```

6. CORR_OFF

CORR_OFF is used in cooperation with CORR_1D, CORR_2D, CORR_3D, CORR_6D), with no motion statement correction after it.

Application method of CORR_OFF:

```
CORR_OFF();
```

7. CAL_6D

Six-dimensional calculation offset statement CAL_6D is used to calculate the angular point position of the work piece, return type is a POSITION type data, with the parameters of cd0, cd1, cd2, cd3, cd4 and cd5.

Note: Application mode of CAL_6D is different from that of CORR_1D and other statements. During the actual programming, its practical spatial points generally are used for the successive motion statements, to complete search correction function.

5.3.3 Example

One-dimensional search:

```
Lin(midpos,dyn0);  
Lin(rap0,dyn0); //walk to the starting point of search  
Search_T(via0,cd0,pat0);
```

//execute the search statement.via0 is the midpoint of search statement, which is used to search the midpoint of a statement and determine the direction of search; cd0 is to store the offset information; pat0 contains the related search parameters, such as search velocity and search range.

```
CORR_1D(cd0); //offset correction;
Lin(midpos,dyn0);
Lin(spos1,dyn0);
ARCSET(arcdata0);
ARCON(arcondata0);
WLin(epos,dyn1);
ARCOFF(arcoffdata0); //welding completed
CORR_OFF();//offset correction completed
```

Two-dimensional search:

```
Lin(midpos,dyn0);
Lin(rap0,dyn0); //run to the starting point of the first search
Search_T(via0,cd0,pat0); //the first search, offset information is saved in cd0
Lin(midpos,dyn0);
Lin(rap1,dyn0); //run to the starting point of the second search
Search_T(via1,cd1,pat0); //the second search, offset information is saved in cd1
CORR_2D(cd0,cd1); //two-dimensional offset correction
Lin(midpos,dyn0);
Lin(spos1,dyn0);
ARCSET(arcdata0);
ARCON(arcondata0);
WLin(epos,dyn1);
ARCOFF(arcoffdata0); //welding completed
CORR_OFF();//offset correction completed
```

Three-dimensional search:

```
Lin(midpos,dyn0);
Lin(rap0,dyn0); //run to the starting point of the first search
Search_T(via0,cd0,pat0); //the first search, offset information is saved in cd0
Lin(midpos,dyn0);
Lin(rap1,dyn0); //run to the starting point of the second search
Search_T(via1,cd1,pat0); // the second search, offset information is saved in cd1
Lin(midpos,dyn0);
Lin(rap0,dyn0); //run to the starting point of the third search
Search_T(via2,cd2,pat0); //the third search, offset information is saved in cd2
CORR_3D(cd0,cd1,cd2); //three-dimensional offset correction
Lin(midpos,dyn0);
Lin(spos1,dyn0);
ARCSET(arcdata0);
ARCON(arcondata0);
WLin(epos,dyn1);
ARCOFF(arcoffdata0); //arc off, welding completed
CORR_OFF();//offset correction completed
```

Six-dimensional search:

```
Lin(midpos,dyn0);
Lin(rap0,dyn0); //run to the starting point of the first search
Search_T(via0,cd0,pat0); //the first search, offset information is saved in cd0
Lin(midpos,dyn0);
```

```

Lin(rap1,dyn0);
Search_T(via1,cd1,pat0); //the second search completed, offset information is saved in cd1
Lin(midpos,dyn0);
Lin(rap2,dyn0);
Search_T(via2,cd2,pat0); //the third search completed, offset information is saved in cd2
Lin(midpos,dyn0);
Lin(rap3,dyn0);
Search_T(via3,cd3,pat0); //the fourth search completed, offset information is saved in cd3
Lin(midpos,dyn0);
Lin(rap4,dyn0);
Search_T(via4,cd4,pat0); //the fifth search completed, offset information is saved in cd4
Lin(midpos,dyn0);
Lin(rap5,dyn0);
Search_T(via5,cd5,pat0); //the sixth search completed, offset information is saved in cd5
CORR_6D(cd0,cd1,cd2,cd3,cd4,cd5); //six-dimensional offset correction
Lin(midpos,dyn0);
Lin(spos1,dyn0);
ARCSET(arcdata0);
ARCON(arcondata0);
WLin(epos,dyn1);
ARCOFF(arcoffdata0); //welding completed
CORR_OFF(); //offset correction completed

```

Relevant search:

```

Lin(midpos,dyn0);
Lin(rap0,dyn0); //run to the starting point of the first search
Search_T(via0,cd0,pat0); //the first search, offset information is saved in cd0
CORR_1D(cd0) //offset correction of the first search
Lin(midpos,dyn0); Lin(rap1,dyn0); //run to the starting point of the second search
Search_T(via1,cd1,pat0); //the second search, offset information is saved in cd1
CORR_1D(cd1); //offset correction of the second search
Lin(midpos,dyn0);
Lin(spos1,dyn0);
ARCSET(arcdata0);
ARCON(arcondata0);
WLin(epos,dyn1);
ARCOFF(arcoffdata0); //welding completed
CORR_OFF(); //offset correction completed

```

5.4 MultiLayer

5.4.1 Variable

Enter multiLayer configuration screen via the tab on the left side of the teach pendant, set the offset of starting point in direction X, endpoint offset and route offset between each pass and direction X by choosing the corresponding multilayer variable as well as the pass in which the expected offset is set via the drop-down list.



MultiLayer configuration screen

The variable MultiLayer will contain 19 data at most.

Each data is composed of StartOffset_X, EndOffset and PachOffset.

Descriptions:

- StartOffset_ and EndOffset_X mean the offset in direction X during welding, and direction X means tangential direction of the welding line.
- NEW is used to build MultiLayer variable.
- DEL is used to delete MultiLayer variable.
- It is able to cancel MultiLayer variable in the screen during it is being used in a program.
- MultiLayer dropdown box is used to choose MultiLayer variable.
- Pass dropdown box is used to choose the pass of multilayer.
- After the data of a pass is set (starting point offset, endpoint offset and route offset), click ENTER to complete data save.



MultiLayer configuration selection menu

Note: X, Y, Z, A, B and C during setting of route offset are the values corresponding to TTS coordinate system, which defines direction of the welding line to be direction X and that of welding gun to be the negative direction Z.

5.4.2 Statements

1. MultiRootInit

MultiRootInit is a multilayer initialization statement, which will assign multilayer variable name and layer number information to the internal multilayer structure to save. This statement includes parameter MultiLayer.

Application method of MultiRootInit:

```
MultiRootInit(mutilayername);
```

Note: this statement must be included in program when MultiLayer function is used.

2. PathMemo

PathMemo is used to record the points which will have offset possibly during multilayer welding. This statement includes parameters MULTIPOSTYPE and MULTISENSTYPE.

Application mode of PathMemo:

```
PathMemo (postype,sensortype);
```

Note:

- MULTIPOSTYPE covers eONWLin, eONWCirc, eOFFWLin, eOFFWCirc, eSWILin and eSWICirc.
- MULTISENSTYPE covers eNONE, eTRACK, eLASER and eWEAVING.

3. LayerPointCorrect

LayerPointCorrect is used to set position offset of some section during multilayer motion. The statement parameters include MultiLayer, LayerNum, PointNum and OFFSET.

Application mode of LayerPointCorrect:

```
LayerPointCorrect (multilayername, layernum, segmentnum, pointoffset);
```

4. LayerLoop

LayerLoop is used to circularly execute the welding operation of the appointed layers. The statement parameters include MultiLayer and LayerNum. MultiLayer is a required parameter and loop pass shall include one pass at least.

Application mode of LayerLoop:

```
LayerLoop (mutilayername, layernum1, layernum2)
```

Note: the above example indicates two loops, firstly welding the pass corresponding to layernum1, then that corresponding to layernum2.

5. EndLayerLoop

EndLayerLoop shall be used in cooperation with Layerloop, its application mode is as follows:

```
EndLayerLoop
```

5.4.3 Example

```
MultiLayer is simple, with its example as follows:
//*****Example program of MultiLayer*****
Tool(tool0); //setup tool;
ARCSET(arcdata0);
//set welding parameters (including welding current and welding voltage);
Lin(mid,dyn0); //approach the pass
MultiRootInit(multi);//MultiLayer initialization statement;
Lin(rap0,dyn0); //walk to arcon point rap0;
ARCON(arcondata0);//arcon;
PathMemo(eONWLin,eNONE);//record offset point information
WLin(rap1,dyn1);//straight line welding;
PathMemo(eSWICirc,eNONE);//record offset point information
```

```

WCirc(rap2,rap3,dyn1);//arc welding
PathMemo(eSWILin,eNONE);//record offset point information
WLin(rap4,dyn1);//straight line welding
PathMemo(eSWICirc,eNONE);//record offset point information
WCirc(rap5,rap6,dyn1);//arc welding
PathMemo(eOFFWCirc,eNONE);//record offset point information
ARCOFF(arcoffdata0);//arcoff;
LayerLoop(multi,1,2,3,4,5)
//begin MultiLayer loop, numbers in brackets are MultiLayer name and pass No. to be welded
Lin(SP000,dyn0); //walk to arcon point
ARCON(arcondata0);//arcon
WLin(SP001,dyn1);//straight line welding
WCirc(SP002,sp003,dyn1);//arc welding
WLin(SP004,dyn1);//straight line welding
WCirc(SP005,sp006,dyn1);//arc welding
ARCOFF(arcoffdata0);//arcoff
EndLayerLoop//multilayer completed

```

5.5 Precautions

- Before the actual welding is started, idling is recommended, to confirm whether the welding procedure is correct and avoid unnecessary danger.
- Weave function shall be added when arc track is used.
- Arc track gain requires more tests when arc track is used. The horizontal offset track gain is recommended to be 1-2 when it is attempted for the first time and the vertical offset track gain is recommended to be 0.1-0.3 when it is attempted for the first time, which may be increased or decreased according to track conditions later. If it requires adding vertical track, welding torch shall not be too close to the work piece, avoiding touching the welding gun during height adjustment.
- The data such as resolution and plate angle limit haven't been obtained for arc track, please save them and feedback the problems and data, so as to provide support for the successive modification. Additionally, arc track function isn't perfect, therefore please keep patience in the process of using.
- Length extended of the welding wire for each search will be fixed.
- Search direction (moving direction of the robot) is vertical to the offset direction of work piece.
- If it needs to search the reference position again, set calibrated as false, which will be automatically modified to true after the first search.

Technical support

◆ Technical service

ADTECH (SHENZHEN) TECHNOLOGY CO., LTD is pleased to provide information about robot running and operation, and assist in your troubleshooting and offer detailed consult. If any fault occurs in the operation of your robot, you can contact our technical service immediately and provide the following information, if possible.

- ✧ Type and serial number of the robot
- ✧ Type and serial number of the control system
- ✧ Version number of the control system
- ✧ Additional software package (optional)
- ✧ Current application program
- ✧ Additional accessories (such as positioner and rail, optional)
- ✧ Problem description, fault duration and frequency etc.

◆ Contact

ADTECH (SHENZHEN) TECHNOLOGY CO., LTD.

Address: 5th Floor, 27-29th Building, Tianxia IC Industrial Park, Majialong, Yiyuan road, Nanshan District, Shenzhen City, Guangdong Province, China

Tel.: +86 138 2882 9353

Fax: +86 755 2672 2718

Email: sales003@adtechen.com

Postcode: 201801

<http://www.steprobots.com>