Operating Instruction for Bending Function of STEP Robot

Status of Publication: Standard
Version: V1.0

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

Foreword

Abstract

This guide describes the bending of STEP robot comprehensively and systematically. This manual may be used as reference for operation of bending functions of STEP robot.

In order to ensure correct operation of bending functions of STEP robot, please read this operating instruction carefully.

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.

Table of Contents

Chapter I Description

1.1 Purpose

This document is aimed to describe the function, operation method and notice for use of the teach pendant. The document can help users quickly learn how to use the robot.

1.2 Target group

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

”SRC4 Technical Reference Manual for Robot Control System”

1.4 Precautions

This manual mainly introduces the basic robot operation, for the details of each robot module, 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), 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 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 related 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 or 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 slow 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 and etc.;

6) Master the test method to ensure the normal function of safety protective devices and interlocking units;

Re-training requirement:

In case system change, personnel change or accident occurs, in order to ensure safe operation, relevant personnel should 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 mechanic damage (such 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 axes 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 Installation of bending functions

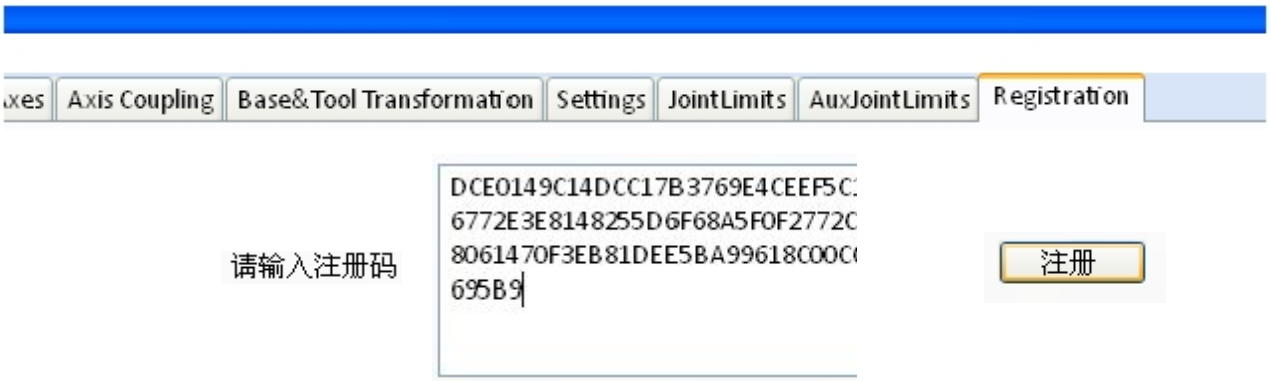
The installation procedure is as follows:

1) Apply for a registration code of bending code from the equipment manufacturer.

2) Connect a control cabinet to a PC computer running the robot configuration software (RobotConfig) (IP is required as 192.168.0.X, X is not 0.)

3) Open the RobotConfig registration interface, enter the acquired registration code firstly, and click the Registration to finish the software registration of bend robot.

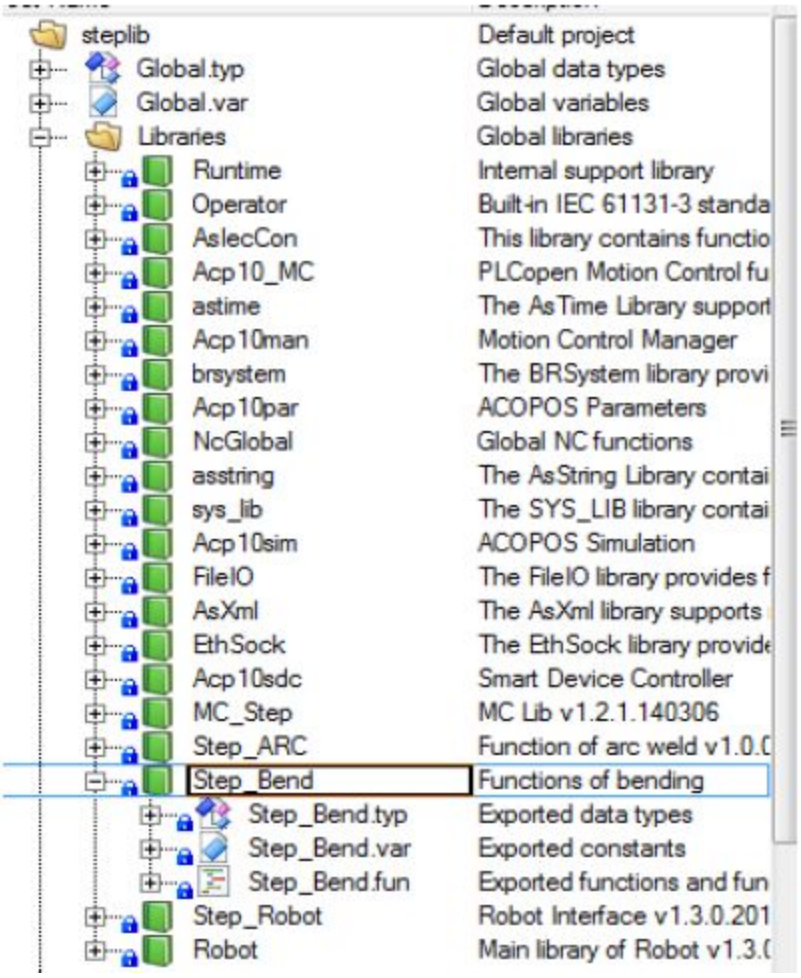
The schematic of registration is shown as follows:



3.2 Configuration of bending functions

PLC is configured as follows:

1） Add a Step_Bend library to PLC



Step_Bend library adding

2） Define a global variable of bend data structure in the Step_Bend library

gBendDI	BendDI	☐	☐	(0)
gBendDO	BendDO	☐	☐	(0)
gBendAI	BendAI	☐	☐	(0)
gBendAO	BendAO	☐	☐	(0)

Variable of bend data structure in the Step_Bend library

3） In MotionTask, call two function blocks of UpdateBendInput and RefreshBendOutput from Step_Bend. Parameter application of two blocks is the global variable established in Step 2. UpdateBendInput may be called in UserPreAction, RefreshBendOutput called in UserPostAction.

Note: Because the global variable is used in UpdateBendInput and RefreshBendOutput, a 25314 boot error caused by these two functions might be reported at the time of PLC program download; at this point, it is OK after an ARWin hot start.

```
(*BendInput*)
UpdateBendInput_0.Group:=0;
UpdateBendInput_0.BendDI:=gBendDI;
UpdateBendInput_0.BendAI:=gBendAI;
UpdateBendInput_0.Enable:=TRUE;
UpdateBendInput_0();
(*BendOuput*)
RefreshBendOuput_0.Group:=0;
RefreshBendOuput_0.BendDO:=ADR(gBendDO);
RefreshBendOuput_0.BendAO:=ADR(gBendAO);
RefreshBendOuput_0.Enable:=TRUE;
RefreshBendOuput_0();
```

Chapter IV Bending functions

4.1 Bend variables

4.1.1 Overview of bend variables

The bend variable Benddata can only be created under the global engineering variable, in which each parameter value may be set, or configured in the bend configuration interface.

The bend variable includes following information mainly:

- Whether an external sensor is provided or not
- BendIO bend port number: 0-7 taken, bend port assignment configured according to PLC
- Board thickness: configured according to practical situation
- Groove width: selected according to practical situation
- Bend angle: bend angle
- Bend speed: bend speed
- Bend ACC time: bend acceleration time
- Bend delay distance: bend delay distance. Describe that the robot starts track bend after a bender has bent for a certain distance.
- Glide distance: glide distance
- Glide speed: glide speed
- Glide ACC time: glide acceleration time
- Glide delay distance: glide delay distance. Describe that the robot starts synchronous motion after a bender has traveled for a certain distance from dead point.
- Return distance: return distance
- Return speed: return speed
- Return ACC time: return acceleration time
- Return delay distance: return delay distance. Describe that the robot starts synchronous motion after a bender has traveled for a certain distance from unload point (dead point shall prevail if the bender has no unload point).
- Hold time：holding time.

- X, Y, Z, A, B, C: reference coordinate system of bender. The origin is the middle point of intersecting line of knife edge and groove planes, the direction of bend motion of the bender is positive direction of Z axis, the direction of knife edge is direction of Y axis, and outward (with front face against the bender) is positive direction of X axis.
- Board distance: distance from knife edge to backstop
- L1 distance: distance of left electronic ruler (the front is faced with the ruler at left hand side of bender) to proper position
- L2 distance: distance of right electronic ruler (the front is faced with the ruler at right hand side of bender) to proper position
- L1L2 distance: space between right and left electronic rulers
- Bend knife IO: bend knife port. Values range from 0 to 31. If the bender needs to switch furniture (bend knife, grinding tool, backstop, etc.) in a complete bend process, this variable is needed to distinguish different furniture.

4.1.2 Editing of bend variables

Select Reference Systems and Tools on the left side of newly-built interface of the variable, then select BENDDATA on the right side, enter the variable name, and click OK to finish new establishment of bend variable BENDDATA. The schematic of new variable establishment is shown as follows:

New variable establishment interface

Notes:

1、 The bend variable can only be built newly in the global directory as a global variable;

2、Editing operation of bend variable (edit, delete, rename, etc.) is similar to that of ordinary variables.

4.2 Bend configuration interface

First bend data configuration interface:

First bend data configuration interface

The parameters configured in this interface mainly include: bendIO, board thickness, groove width, bend angle and speed, bend coordinate system, whether external coordinate system used or not, etc.

Second bend data configuration interface:

Second bend data configuration interface

The parameters configured in this interface mainly include: glide distance, speed, ACC time; return distance, speed, ACC time, delay distance; bendIO, relevant L1 & L2 data.

Note: The bend coordinate system must be firstly demonstrated in the coordinate system teach interface and then is selected in the bend configuration interface after teach is finished; moreover, the selected coordinate system can be used only if “Confirm” is clicked in the second configuration interface.

4.3 Bend statements

4.3.1 Overview of bend statements

The following bend statements are newly added for bending functions:

1. BendSignalSet(bendname,Signaltype)

This statement is to send fast lowering/bend/unload/return signal to bender according to the character string selected by Signaltype and has two parameters. Its usage mode is as follows:

```
BendSignalSet(bendname,Signaltype);
```

Where, bendname is bend variable name, and Signaltype is one of eBendToVelChange (bend to velocity change point signal), eBendToClamp (bend to clamp point signal), eBendProcess (bend start signal), eUnloading (unloading signal) and eMoveBack (moveback signal) signals sent to corresponding IO group in bendname. In use, bendname

is the name of newly built bend variable selected from drop-down box, and Signaltype is one of 5 signals selected from drop-down box according to physical situation.

Note: For asynchronous motion, BendSignalSet command may be used to set the knife signal.

2. Bendtrack (bendname)

It is a bend track statement and includes only one parameter, i.e. bend variable name bendname. Its usage mode is as follows:

```
Bendtrack(bendname);
```

In use, bendname is the name of newly built bend variable selected from drop-down box. During processing of this statement, hold time is included.

3. BendSynMove (bendname,type)

The usage mod of synchronous movement is as follows:

```
BendSynMove(bendname,type);
```

BendSynMove may be selected according to parameter type to realize the functions of synchronous moveup and movedown; bendname is bend variable name, type indicates the direction of synchronous movement, including two modes of eDown and eUp.

4. BendTuning(bendname,dynamic)

The lowering tuning statement is used for tuning after the backstop is in place. Its usage mode is as follows:

```
BendTuning(bendname,dynamic);
```

Two parameters, i.e. bendname and dynamic, bendname are bend variable names, dynamic is dynamic parameter, used for setting the speed, acceleration, jerk of lowering tuning; dynamic is dynamic parameter.

5. BendLin(bendname,position,dynamic)

It is a slab stock position adjustment statement with tuning (Lin motion). Its usage mode is as follows:

```
BendLin(bendname,postion,dynamic);
```

This statement moves in the bend coordinate system; if there is no coordinate system statement previously, the world coordinate system is defaulted; in case of tuning by electronic ruler, this statement has tuning itself; there is not tuning if no electronic ruler.

This statement includes three parameters, bendname, position and dynamic; where, bendname is bend variable name, position is a position point, i.e. rcp,rap,cp,ap point, and dynamic is dynamic parameter.

4.3.2 Editing of bend statements

When statement is newly built, select the bend statement to display its options, and carry out selection according to use condition.

Bend statement

Editing of bend statements is similar to that of other ordinary statements, e.g. modify, comment, etc.

4.4 Examples of use

4.4.1 Ordinary track bend

An example of a complete ordinary bend program is shown as follows:

```
//*************Ordinary track bend*************//

PTP（ap0）
Lin（cp0）；     // Charge
BendSignalSet(benddata0, eBendToVelChange)；// Knife down
BendTuning（benddata0）；  //Tuning
Bendtrack（bendname, nosyn）；  //Track bend
Lin（p2）；      // Reject
Lin（p3）；
```

4.4.2 UP-DOWN track bend

An example of a complete UP-DOWN track bend program is shown as follows:

```
//*************UP-DOWN track bend***************//
PTP（ap0）;
Lin（cp0）；     // Put in position of top dead point
BendSynMove（benddata0，eDown）；//Synchronous glide
Lin（cp1）；     // Bend position
BendTuning（benddata0）；  //Tuning
Bendtrack（benddata0）；// Track bend
BendSynMove（benddata0,eUp）；//Synchronous return
BendLin (benddata0,cp2)；//Adjust position
Lin（cp3）；  //Reject
Lin（cp4）；
```

4.5 Precautions

- BendIO needs to be used with PLC port. For example, the input/output signal of one bender is bound to #1 port of bender at the PLC terminal. If it is intended to use this bender in the program, the BendIO is changed to 1 in the corresponding bend variable.
- Use of bend knife IO. One bender might need to switch the furniture (bend knife, grinding tool, backstop, etc.) in a complete process, and each offset generated in bend

tuning of each tool is denoted with bend knife IO. This means that not only tooling data (Board distance, L1 distance, L2 distance, L1L2 distance) differ in multiple bend variables corresponding several furniture, but the bend knife IO differs.

- The groove width is the width of whole groove of the lower die but not a half of this width.
- The bend angle is the angle formed by two faces after bend finishes but not a half of this angle.
- The bend reference coordinate system is set as follows: 1. Demonstrate a selected ordinary reference coordinate system in the coordinate system teach interface; 2. Select this reference coordinate system in the bend variable teach interface.
- The bend reference coordinate system is a coordinate system used in the robot system; selection of reference coordinate system (e.g. ref1) in the bend teach interface is only to finish the assignment of one ordinary reference coordinate system to bend coordinate system. This means that modification of ref1 value will not change the value of bend reference coordinate system after the bend coordinate system teach is finished, and ref1 will not be displayed in the teach pendant interface.
- Five parameters, Board distance, L1 distance, L2 distance, L1L2 distance and Bend knife IO, are used in the bend tuning statement (BendTuning). This means that these five parameters may not be entered if the bend tuning function is not used in the bend process.
- Bend is divided into two kinds based on whether a sensor is provided: with external sensor and without external sensor. If the external sensor is provided, the following parameters are requisite: BendIO, Board thickness, Groove width, Hold time, Bend reference coordinate system, Bend angle; if not provided, the following parameters are requisite: BendIO, Board thickness, Groove width, Hold time, Bend reference coordinate system, Bend angle, Bend speed, Bend ACC time, Bend delay distance, Glide distance, Glide speed、, Glide ACC time, Glide delay distance, Return distance, Return speed, Return ACC time, Return delay distance, Hold time.
- Prior to actual bend, first simulate bend once to avoid unnecessary danger and loss.

Technical Support

◆ **Technical Service**

ADTECH (SHENZHEN) TECHNOLOGY CO., LTD is pleased to provide relevant information on operation and handling as well as detailed consultation to help you remove the trouble. If your robot fails to operate in the process of production, please contact our technical surface agency immediately with following information offered to us as much as possible:

✧ Type and serial number of robot

✧ Type and serial number of control system

✧ Version number of control system

✧ Additional software function package (optional)

✧ Existing applications

✧ Other additional devices (positioner, guide rail etc., optional)

✧ Problem description, trouble duration and frequency, and so on

◆ **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