

机器人学习资料



机器人学习资料

目录

- 一、 IRC5控制柜的认识
- 二、 机器人本体的认识
- 三、 转数计数器更新
- 四、 定义工具坐标系
- 五、 定义工件坐标系
- 六、 Load data操作步骤
- 七、 IRC5控制系统生成（从备份中生成系统）
- 八、 定义I/O总线
- 九、 定义I/O单元
- 十、 定义I/O信号
- 十一、 基本指令
- 十二、 off()功能
- 十三、 程序的编辑
- 十四、 电路分析
- 十五、 摇杆校正

一、IRC5控制柜的认识



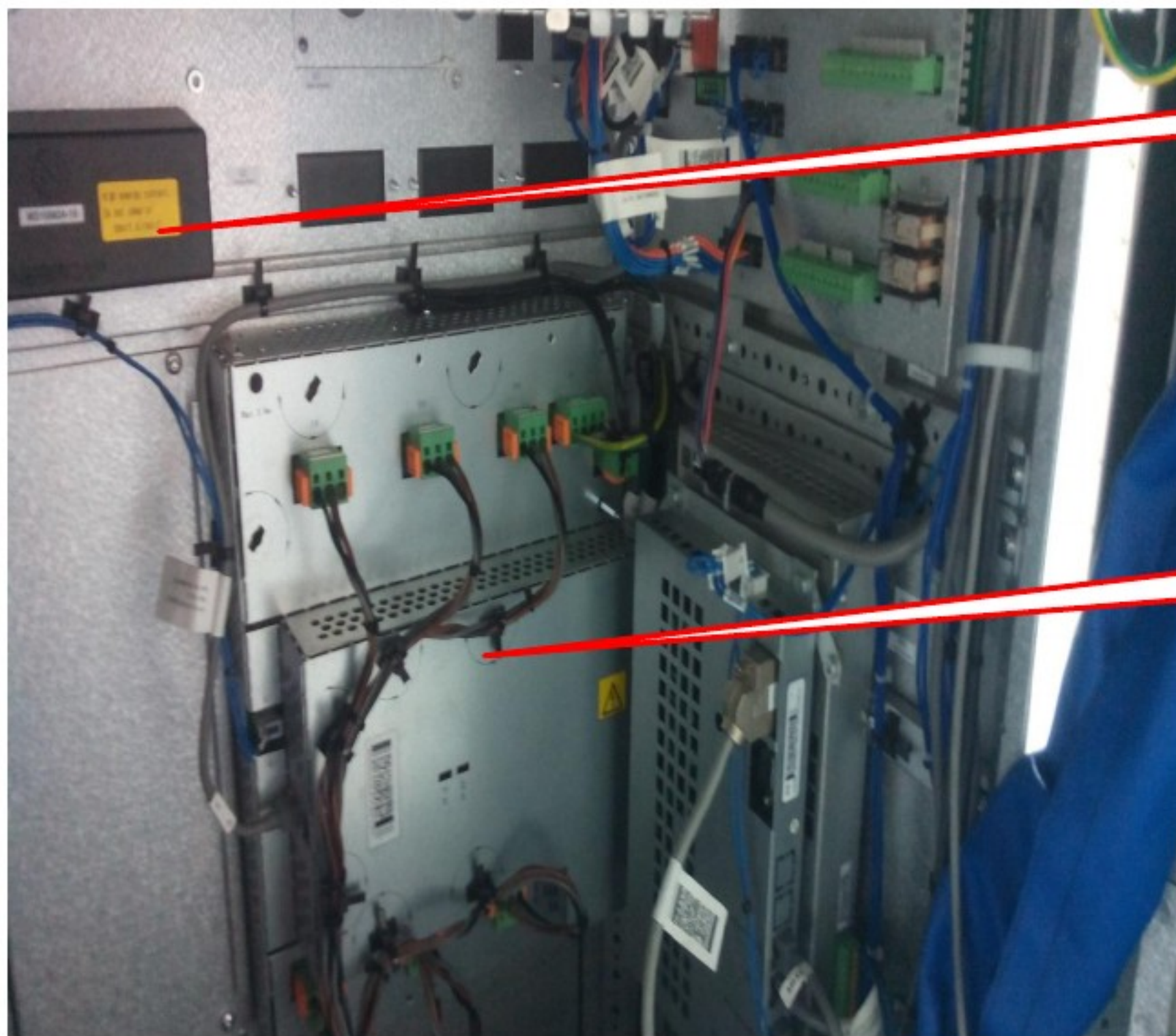
安全控制板

控制柜操作面板上的急停开关、TPU上的急停按钮和外部的安全信号由安全板处理

轴计算机

该计算机不保存数据，机器人本体的零位和机器人当前位置的数据都由轴计算机处理，处理后的数据传送给主计算机

一、IRC5控制柜的认识



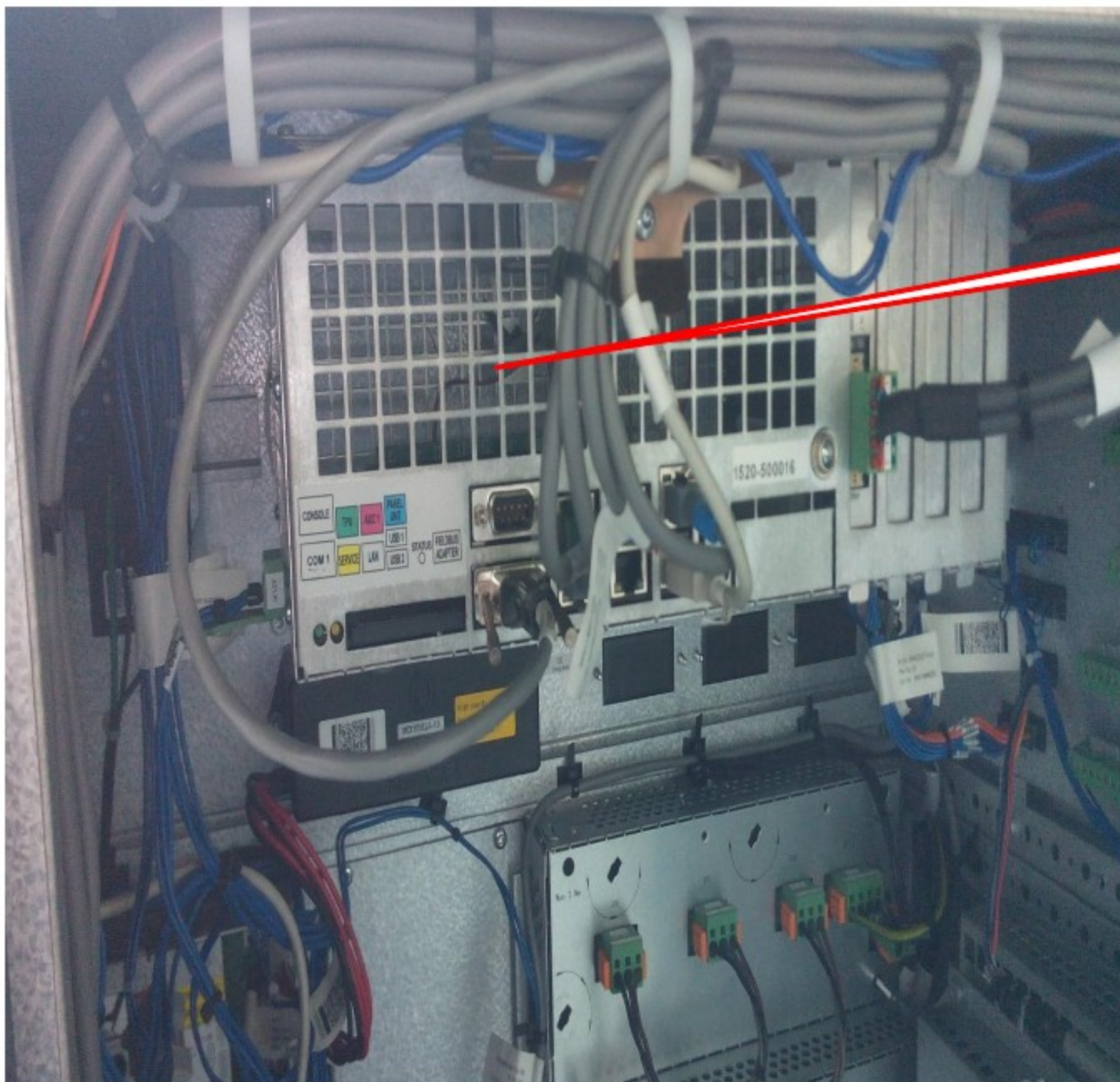
I/O供电装置

该装置给I/O单元、用户自定义板供电

驱动装置

驱动装置接受到主计算机传送的驱动信号后，驱动机器人本体

一、IRC5控制柜的认识



主计算机



接收处理机器人运动数据
和外围信号，将处理的
信号发送到各单元

一、IRC5控制柜的认识



电源分配器

给各主计算机、安全控制板、轴计算机、TPU等分配24VDC（需24VDC的用电装置）

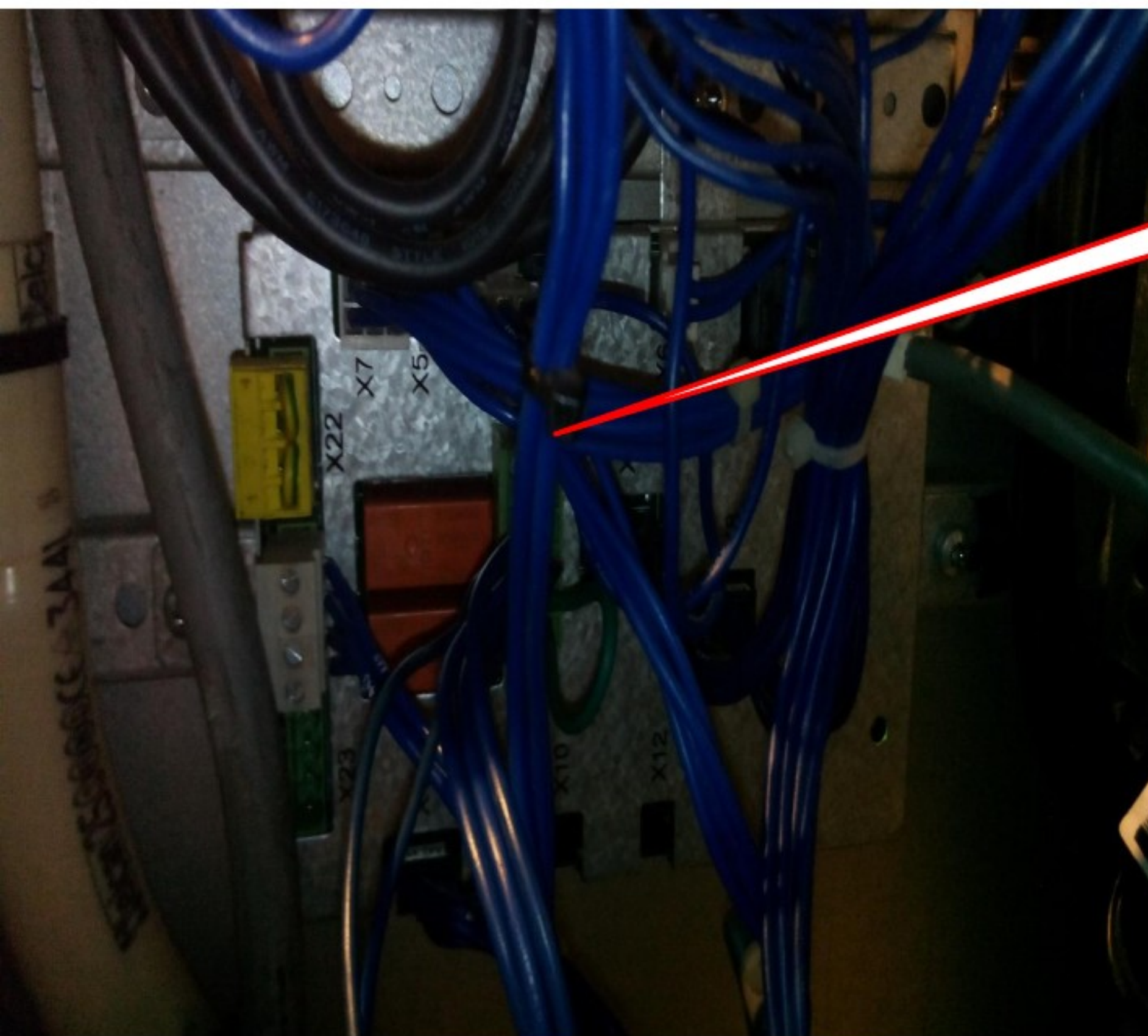
供电模块

给电源分配器提供24VDC

接触器

刹车和驱动接通220VAC和380VAC

一、IRC5控制柜的认识

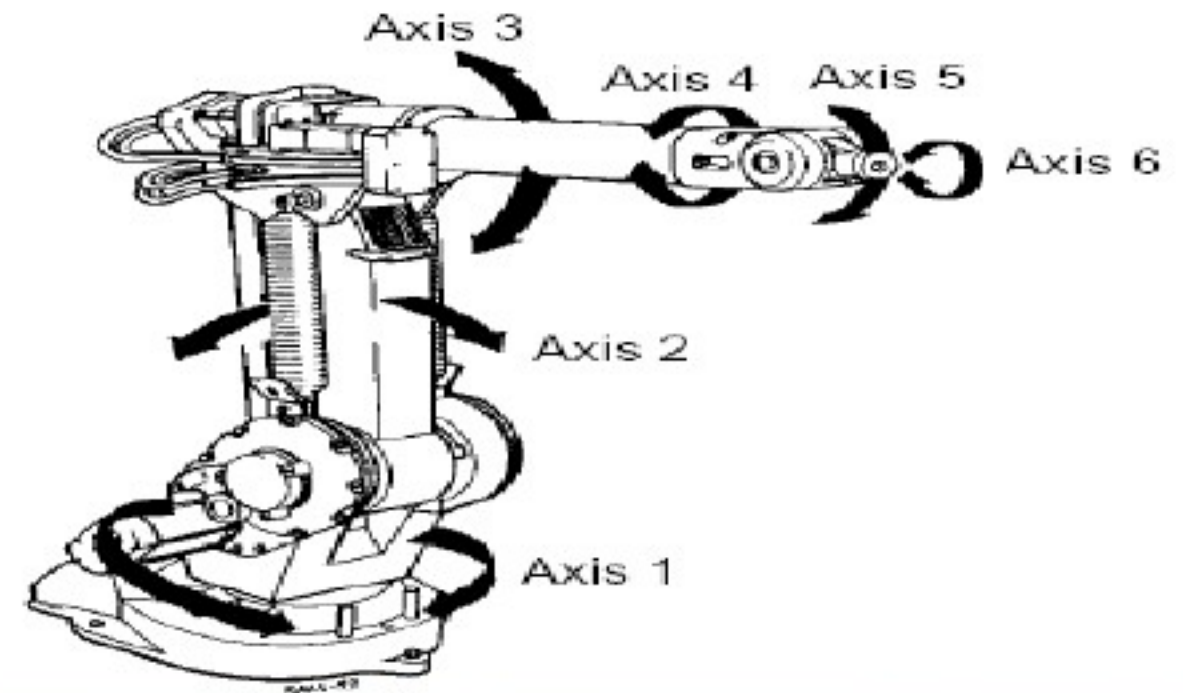


接触器板



给接触器提供电源及相
关逻辑信号

二、机器人本体的认识



- 1、机械手是由六个转轴组成六杆开链机构，理论上可达运动范围内空间任何一点。
- 2、六个转轴均有AC伺服电机驱动，每个电机后均有编码器
- 3、每个转轴均带一个齿轮箱，机械手运动精度（综合）达正负0.05MM至正负0.2MM
- 4、机械人必须带有24VDC.(机器人配置)机械手带有平衡气缸或弹簧

二、IRC5控制柜的认识



串口测量板 (SMB)

串口测量板有六节1.2V的锂电池，保存数据时供电作用



手动松闸按钮

机械手带有手动松闸按钮，维修时使用，非正常使用会造成设备或人员被伤害

三、更新转数计数器

转数计数器用来告诉电机轴在齿轮箱中的转数，此值丢失机器人不能运行任何程序

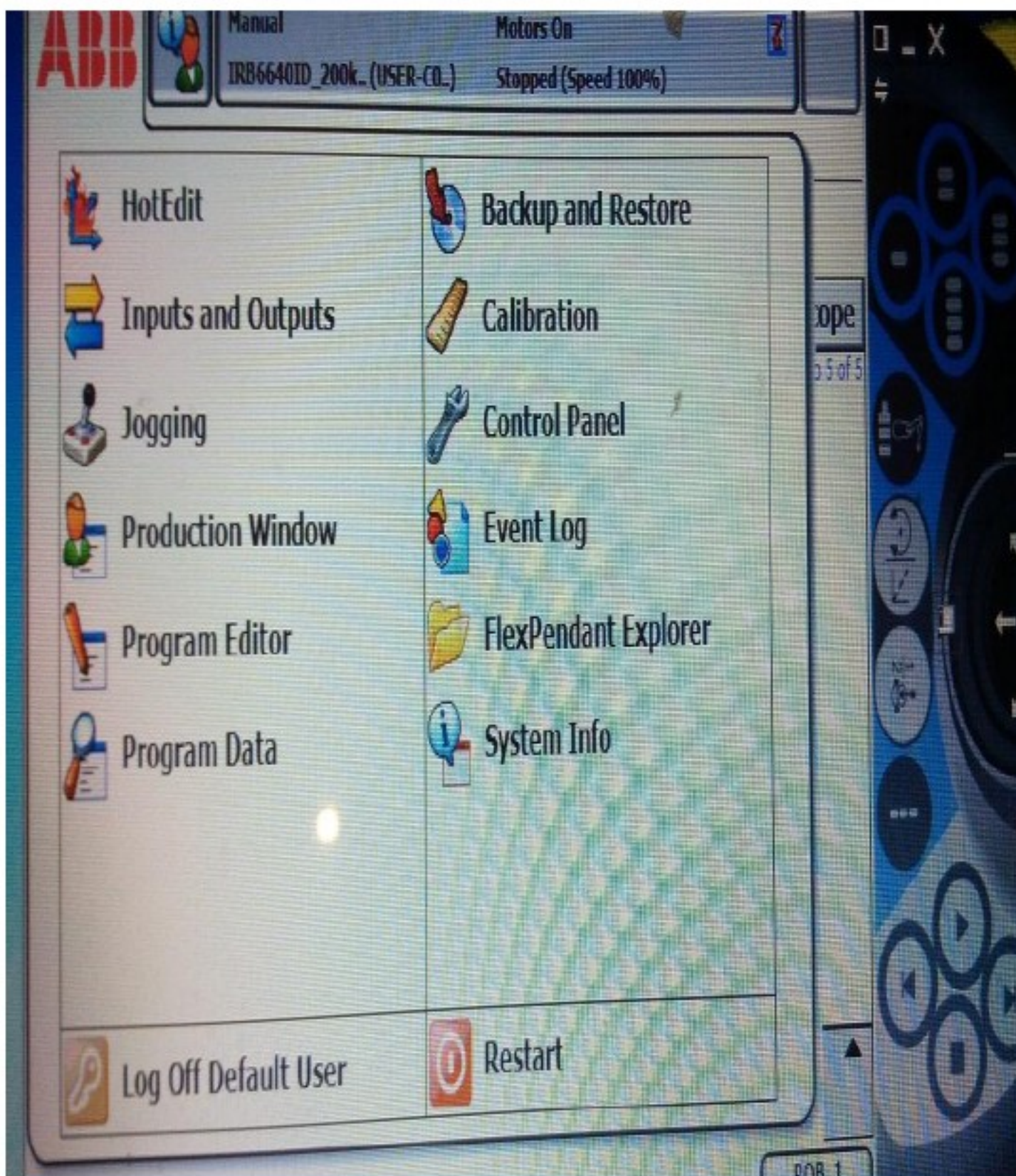
更新转数计数器时，手动操作6个轴到同步标记位置上（标准位置有划线标记或者有卡尺标记，不同型号的机器人位置不同）

更新转数计数器时，如位置狭小，可以逐轴更新

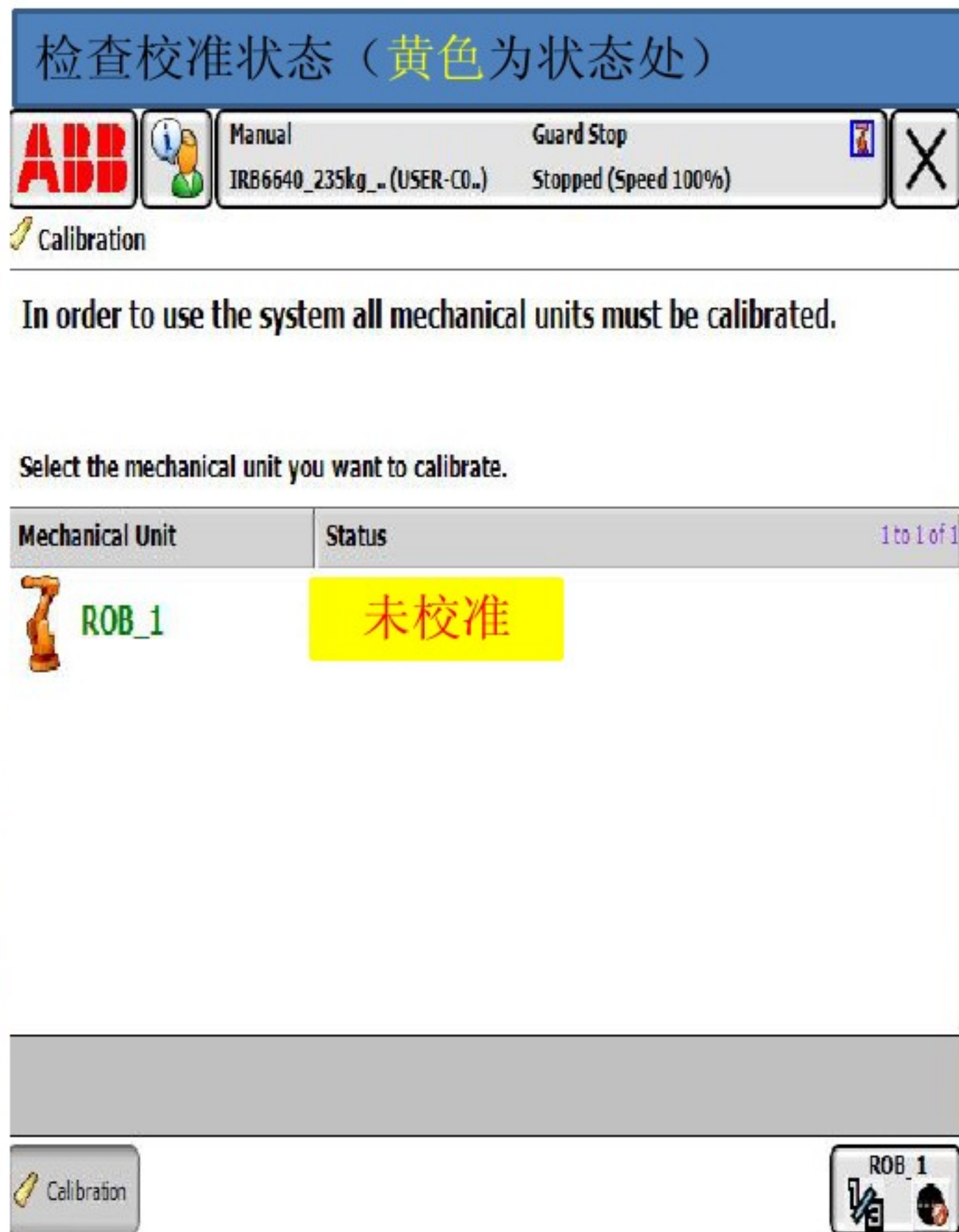
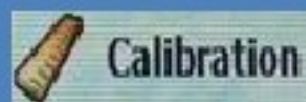
更新转数计数器时，检查是否在正确的位置上更新

更新转数计数器时，没有在位置上更新会导致定位不准确，以致造成伤害

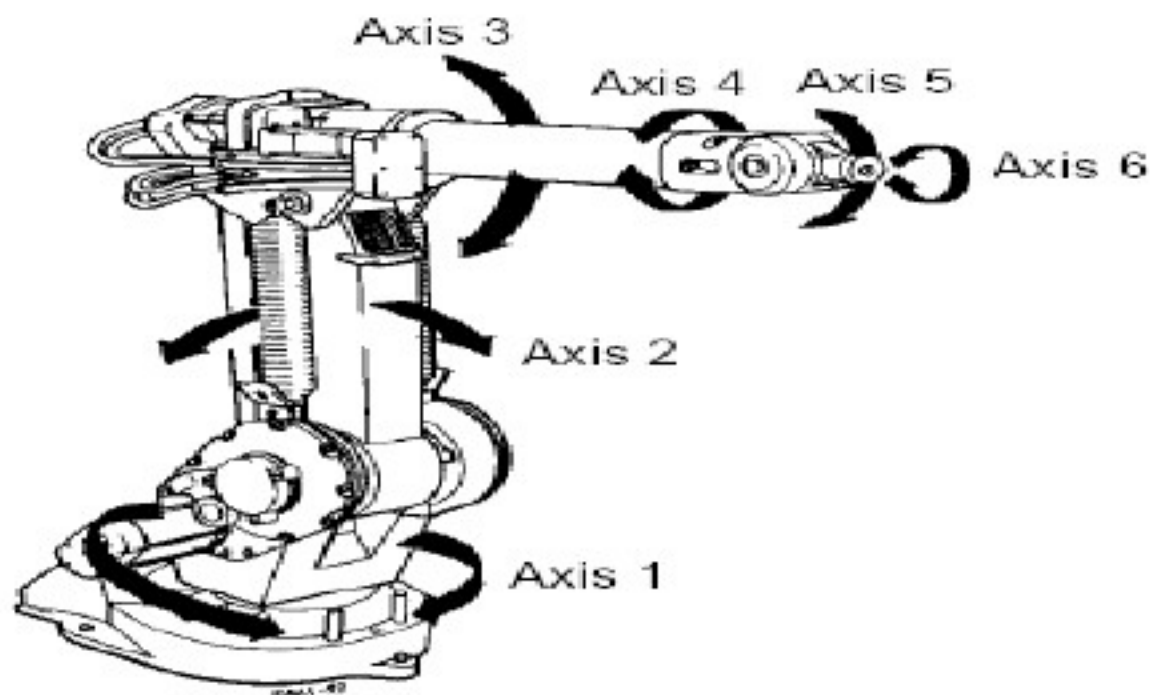
三、更新转数计数器



单击ABB, 点击校准



三、更新转数计数器



把六根轴摇到位置处（0位），查看本体机械零位，是否在正确位置

单击



ROB_1

未校准



Manual
IRB6640_235kg_.. (USER-CO..)

Guard Stop
Stopped (Speed 100%)



Calibration

In order to use the system all mechanical units must be calibrated.

Select the mechanical unit you want to calibrate.

Mechanical Unit	Status
 ROB_1	未校准

Calibration

ROB_1

三、更新转数计数器

ABB

Manual

IRB6640_235kg_.. (USER-CO..)

Guard Stop

Stopped (Speed 100%)

Calibration - ROB_1

Rev. Counters

Calib. Parameters

SMB Memory

Base Frame

Update Revolution Counters...

Close

Calibration

ABB

Manual

IRB6640_235kg_.. (USER-CO..)

Guard Stop

Stopped (Speed 100%)

Calibration - ROB_1

Rev. Co

Calib. Pa

SMB M

Base

Warning

Updating the revolution counters may change programmed positions.

Are you sure you want to proceed?

Yes

No

Close

Calibration

ROB_1

点击

Rev. Counters

再点击

Update Revolution Counters...

再次点击

Yes

三、更新转数计数器

ABB

Manual

IRB6640_235kg_.. (USER-CO..)

Guard Stop

Stopped (Speed 100%)

Calibration - ROB_1 - Rev. Counters

Update Revolution Counters

Select mechanical units to update revolution counters.

Mechanical Unit	Status
☒ ROB_1	Calibrated

OKCancel

ABB

Manual

IRB6640_235kg_.. (USER-CO..)

Guard Stop

Stopped (Speed 100%)

Calibration - ROB_1 - Rev. Counters

Update Revolution Counters

Mechanical unit: ROB_1

To update revolution counters select axes and tap Update.

Axis	Status
☒ rob1_1	Revolution Counters Updated
☒ rob1_2	Revolution Counters Updated
☒ rob1_3	Revolution Counters Updated
☒ rob1_4	Revolution Counters Updated
☒ rob1_5	Revolution Counters Updated
☒ rob1_6	Revolution Counters Updated

Select AllClear AllUpdateClose

Calibration

ROB_1

在方框内画



点击OK

点击
再点击

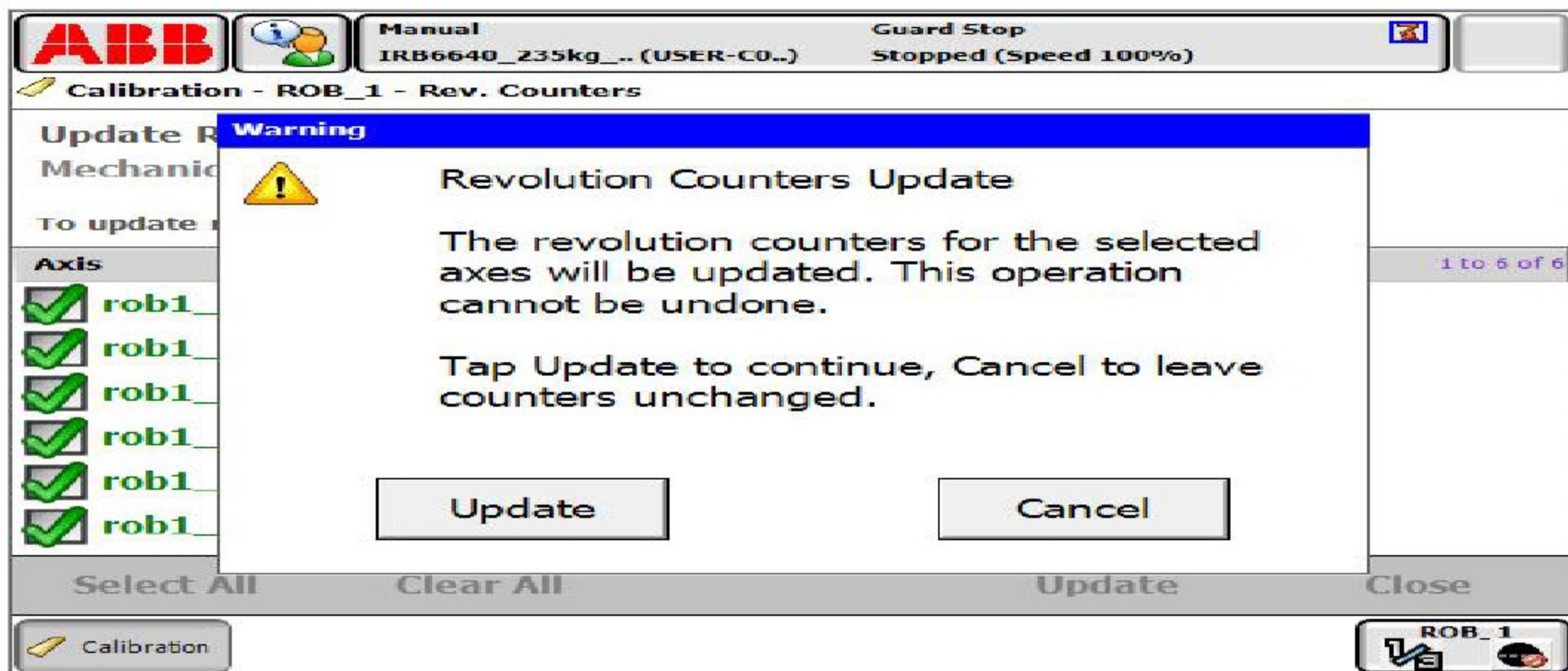
Select All

见到方框里全部

Update



三、更新转数计数器



点击

Update

再点击

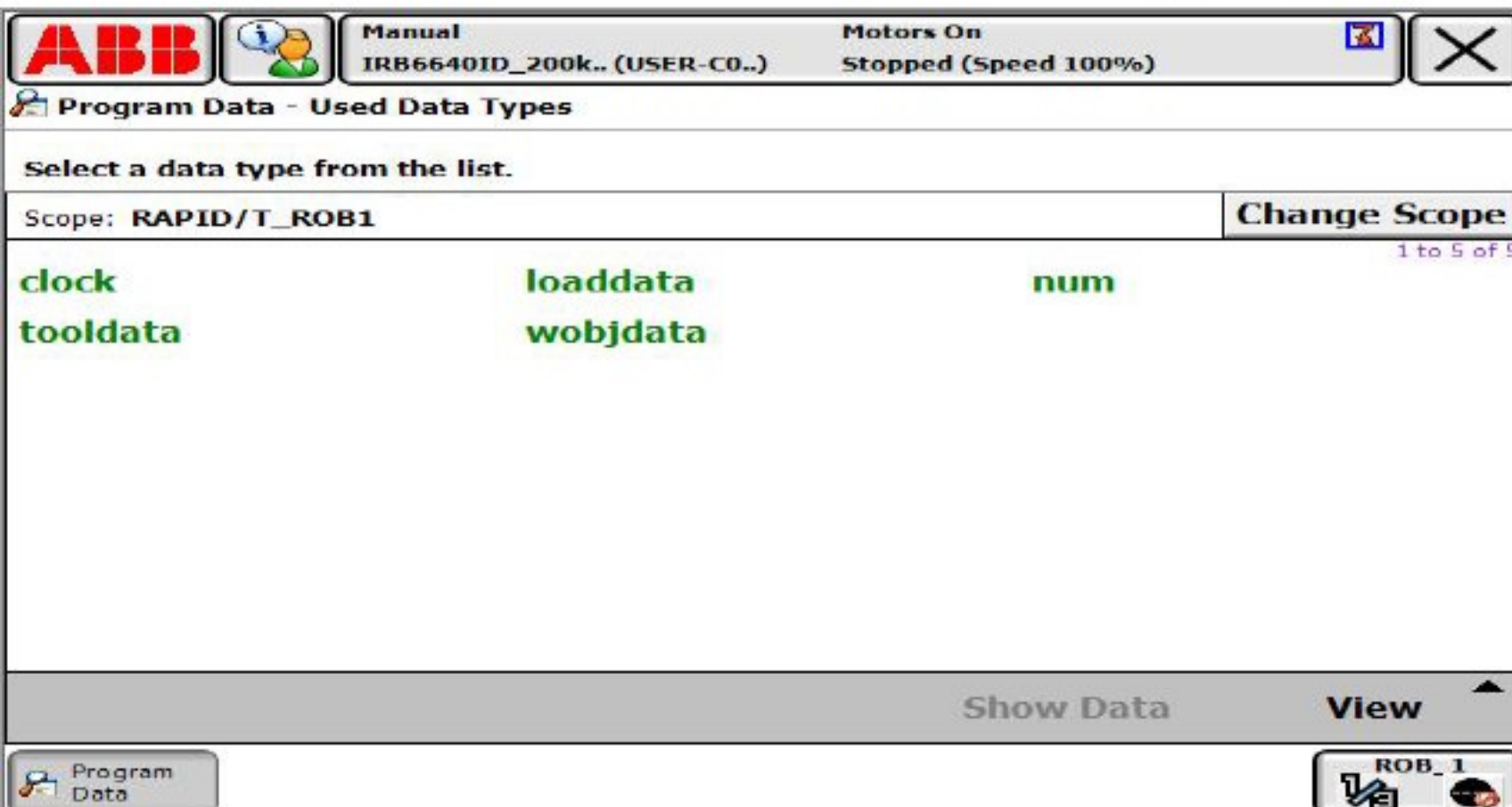
OK

四、定义工具坐标系

点击ABB菜单，选择
Program data



选择数据
tooldata



注意：确认是在手动模式

四、定义工具坐标系

Manual
 IRB6640ID_200k.. (USER-CO..)
 Motors On
 Stopped (Speed 100%)

Data of type: tooldata

Active filter:

Select the data you want to edit.

Scope: RAPID/T_ROB1
 Change Scope

Name	Value	Module	1 to 2 of 2
tool0	[TRUE,[[0,0,0],[1,0...	BASE	Global
tool1	[TRUE,[[0,0,0],[1,0...	MainModule	Task

Program Data
 ROB_1

点击新建（new）出现
 弹出下面窗口，在窗口
 name处可修改命名
 （TOOL1），其它为默认，
 点击OK

Manual
 IRB6640ID_200k.. (USER-CO..)
 Motors On
 Stopped (Speed 100%)

New Data Declaration

Data type: tooldata
 Current Task: T_ROB1

Name: tool2

Scope: Task

Storage type: Persistent

Task: T_ROB1

Module: MainModule

Routine: <None>

Dimension: <None>

Initial Value
 OK
 Cancel

Program Data
 ROB_1

四、定义工具坐标系

Manual
Motors On
IRB6640ID_200k.. (USER-CO..)
Stopped (Speed 100%)

Data of type: tooldata

Active filter:

Select the data you want to edit.

Scope: RAPID/T_ROB1
Change Scope

Name	Value	Module	
tool0	[TRUE,[[0,0,0],[1,0...	BASE	Global
tool1	[TRUE,[[0,0,0],[1,0...	MainModule	Task

New...

Edit

Refresh
View Data Types

Program Data
ROB_1

双击tool1，直接进入编辑窗口，
输入工具的重量

手动
电机启动
OP (ON-L-0315172)
已停止 (速度 100%)

编辑

名称: tPen

点击一个字段以编辑值。

名称	值
mass :=	2
cog:	[0, 0, 0]
x :=	0
y :=	0
z :=	0
tool:	[1, 0, 0, 0]

7 8 9 ←

4 5 6 →

1 2 3 ⊗

0 +/- . F-E

确定 取消

撤消 确定 取消

程序数据

ROB_1

Mass: 工具重量 Kg
Cog: 工具重心 mm

四、定义工具坐标系



Manual
IRB6640ID_200k.. (USER-CO..)

Motors On
Stopped (Speed 100%)



Data of type: tooldata

Active filter:

Select the data you want to edit.

Scope: RAPID/T_ROB1

Change Scope

Name	Value	Module	
tool0	[TRUE,[[0,0,0],[1,0...	BASE	Global
tool1	[TRUE,[[0,0,0],[1,0...	MainModule	Task

删除
更改声明
更改值
复制
定义
编辑



New...



Edit

Refresh

View Data Types

Program Data

ROB_1

选择新建工具坐标（蓝色光标），点击 Edit,弹出窗口，选择定义

选择定义TCP的方向



Manual
IRB6640ID_200k.. (USER-CO..)

Motors On
Stopped (Speed 100%)



Program Data - tooldata - Define

Tool Frame Definition

Tool: tool1

Select a method, modify the positions and tap OK.

Method: TCP & Z, X

No. of points: 4

Point	
Point 1	TCP (默认方向)
Point 2	TCP 和 Z
Point 3	TCP 和 Z, X
Point 4	

Positions

Modify Position

OK

Cancel

Program Data

ROB_1

四、定义工具坐标系

ABB Manual Motors On
IRB6640ID_200k.. (USER-CO..) Stopped (Speed 100%)

Program Data - tooldata - Define

Tool Frame Definition
Tool: tool1

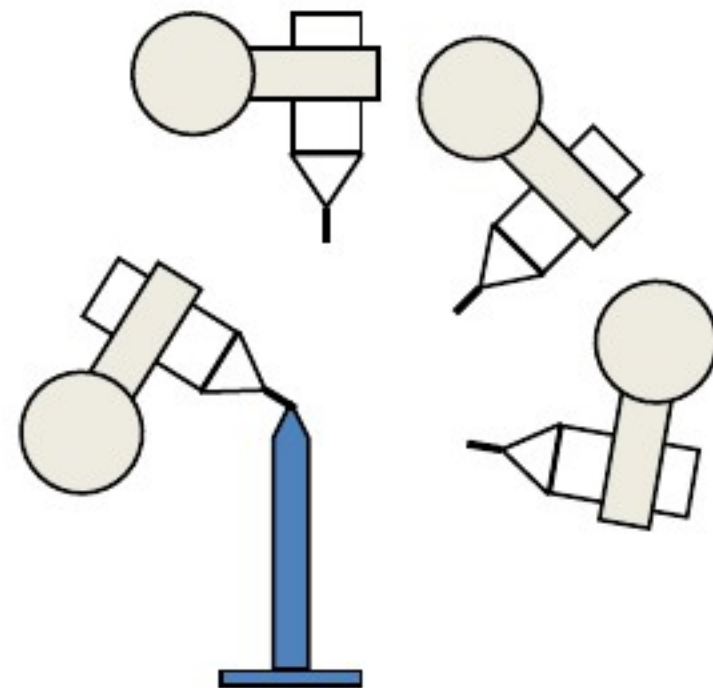
Select a method, modify the positions and tap OK.

Method: TCP & Z, X No. of points: 4

Point	Status
Point 1	-
Point 2	-
Point 3	-
Point 4	-

Positions Modify Position OK Cancel

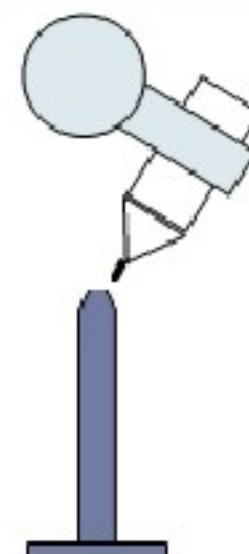
Program Data ROB_1



在Method处选择做TCP的点数，手动操纵机器人，使TCP点和定点相碰，使用功能键 修改位置 记录机器人相应位置，OK键确认。

四、定义工具坐标系

检测新建工具坐标



手动移动机器人，使TCP点和定点相碰。
使用 **重定位运动（姿态运动）** 模式来检验工具坐标系。

偏差最好在1以下，越接近0最好

红色选择你新建的工具坐标



手动

电机开启

已停止 (速度 100%)

✕

手动操纵

点击属性并更改

机械单元: ROB_1...

绝对精度: Off

动作模式: 重定位...

坐标系: 工具...

工具坐标: **tool1**

工件坐标: wobj0...

有效载荷: load0...

操纵杆锁定: 无...

增量: 无...

位置

坐标中的位置: WorkObject

X: 374.0 mm

Y: 0.0 mm

Z: 630.0 mm

q1: 0.70693

q2: 0.0

q3: 0.70728

q4: 0.0

位置格式...

操纵杆方向

X Y Z

对准...

转到...

启动...

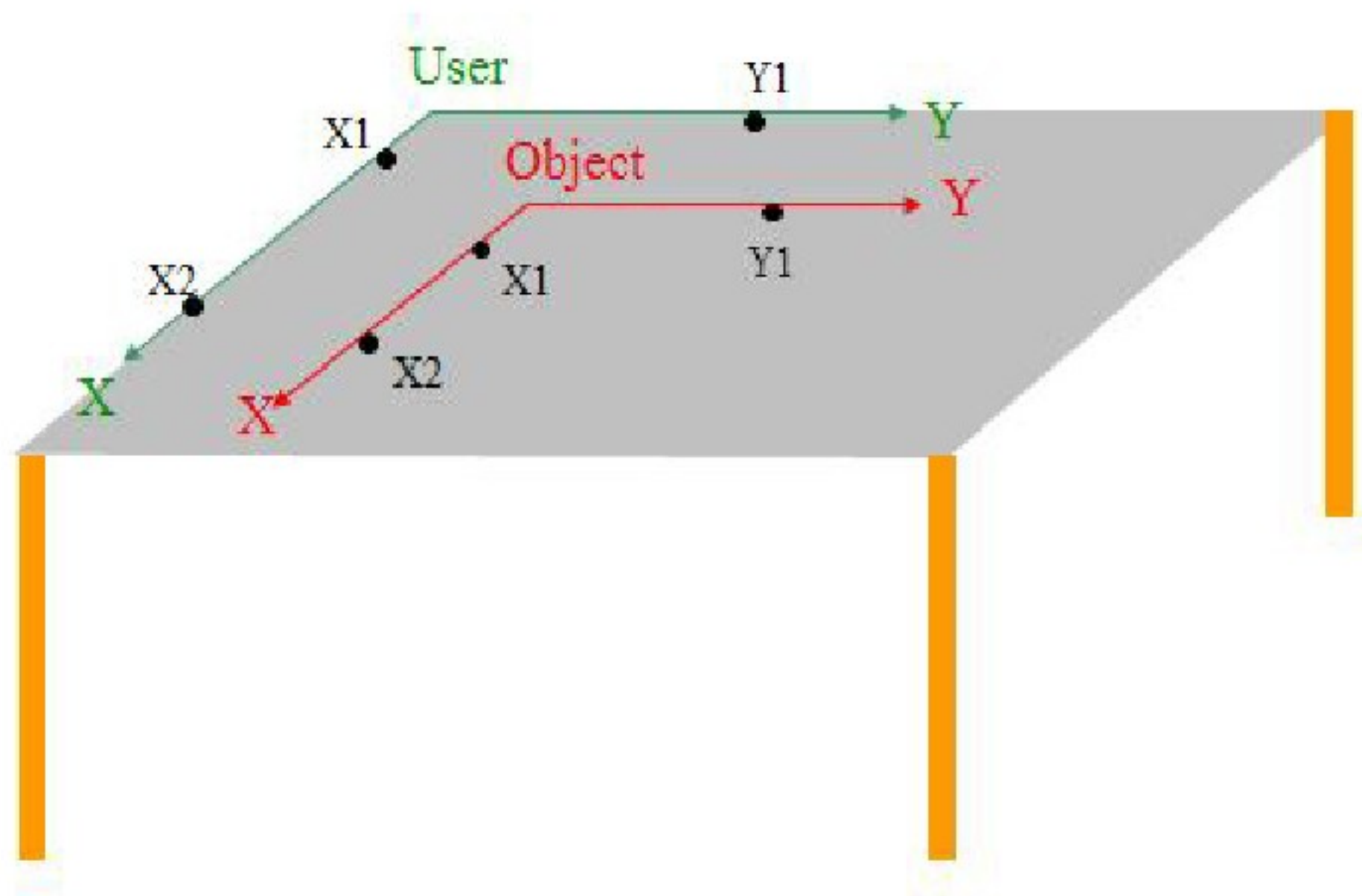
程序数据

手动操纵

ROB_1

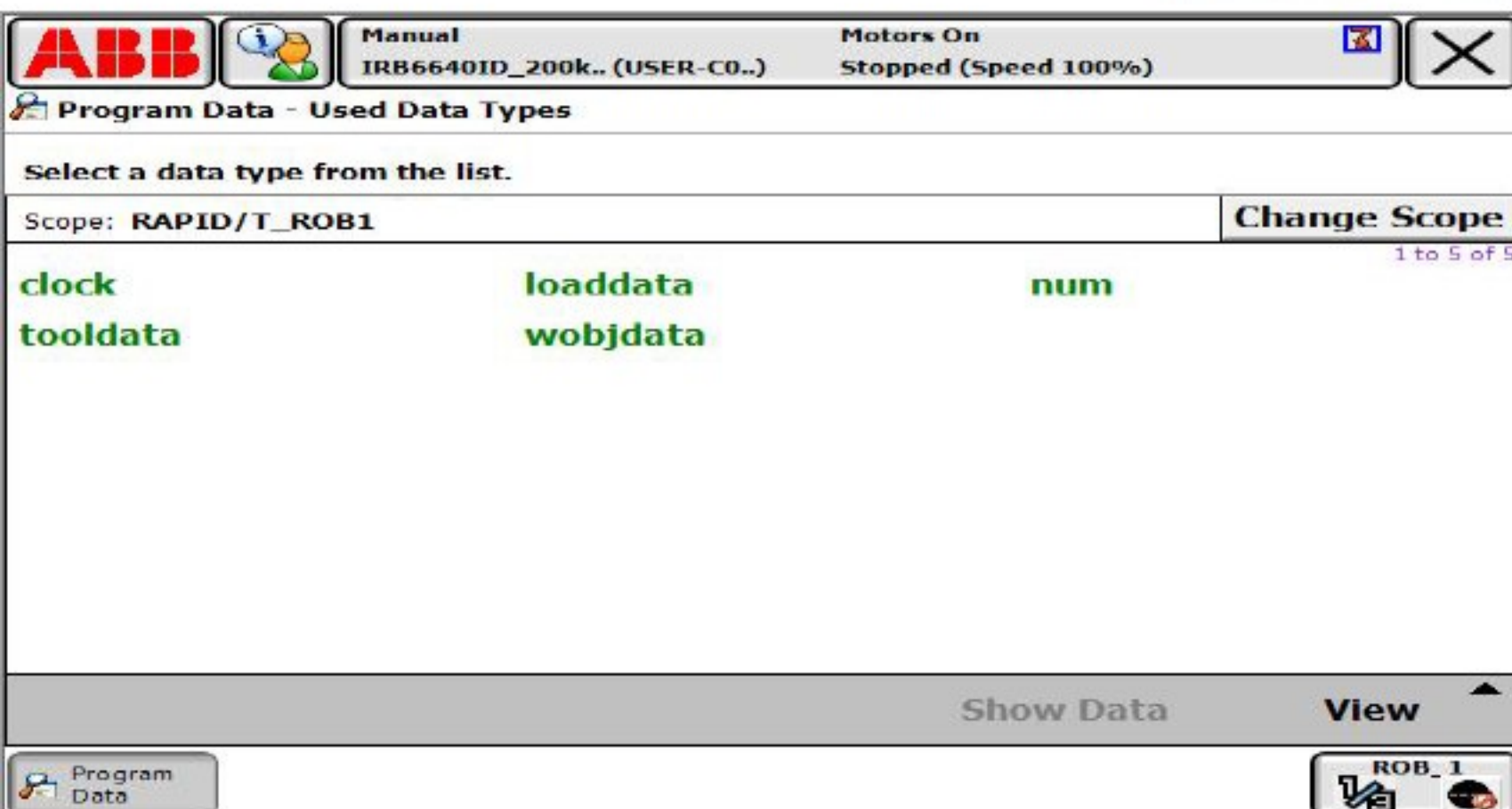
五、定义工件坐标系

- 三点法—点 $X1$ 与点 $X2$ 连线组成 X 轴，通过点 $Y1$ 向 X 轴作的垂直线，为 Y 轴。



五、定义工件坐标系

点击ABB菜单，选择
Program data



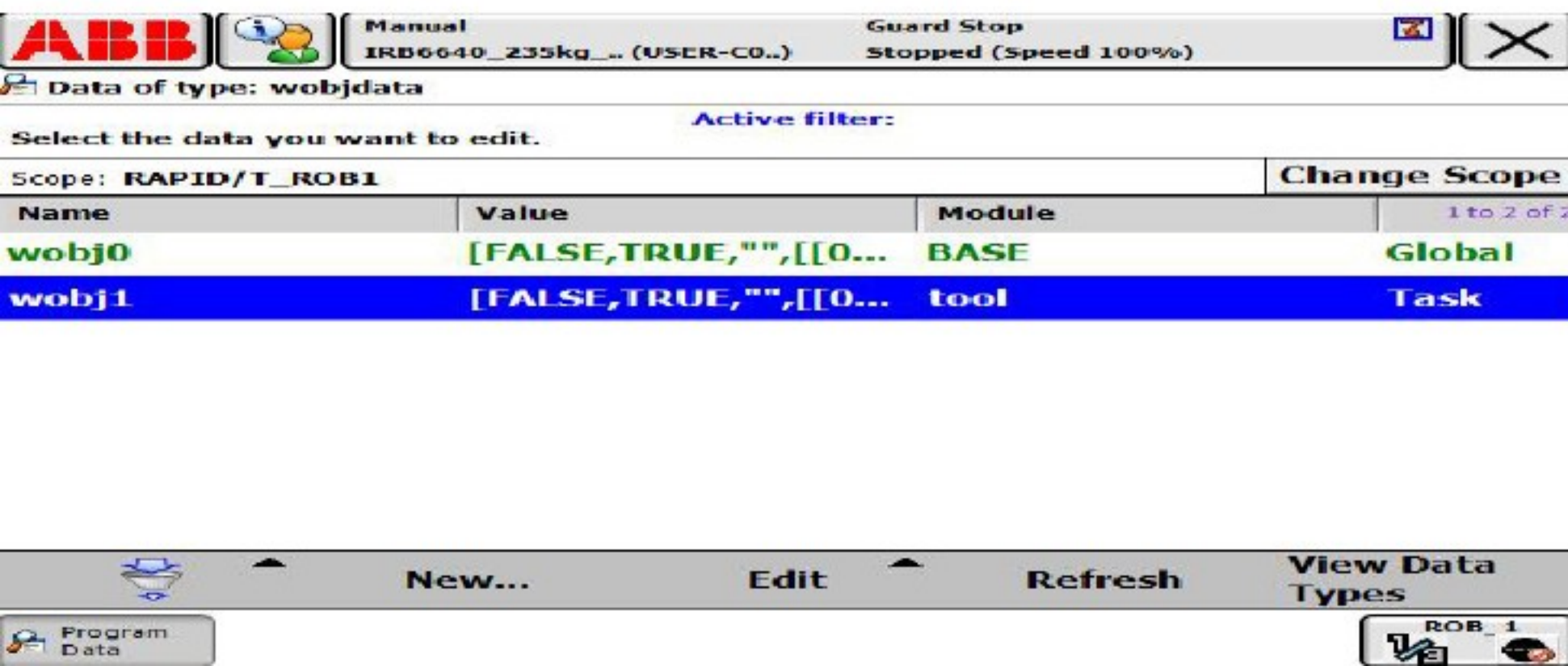
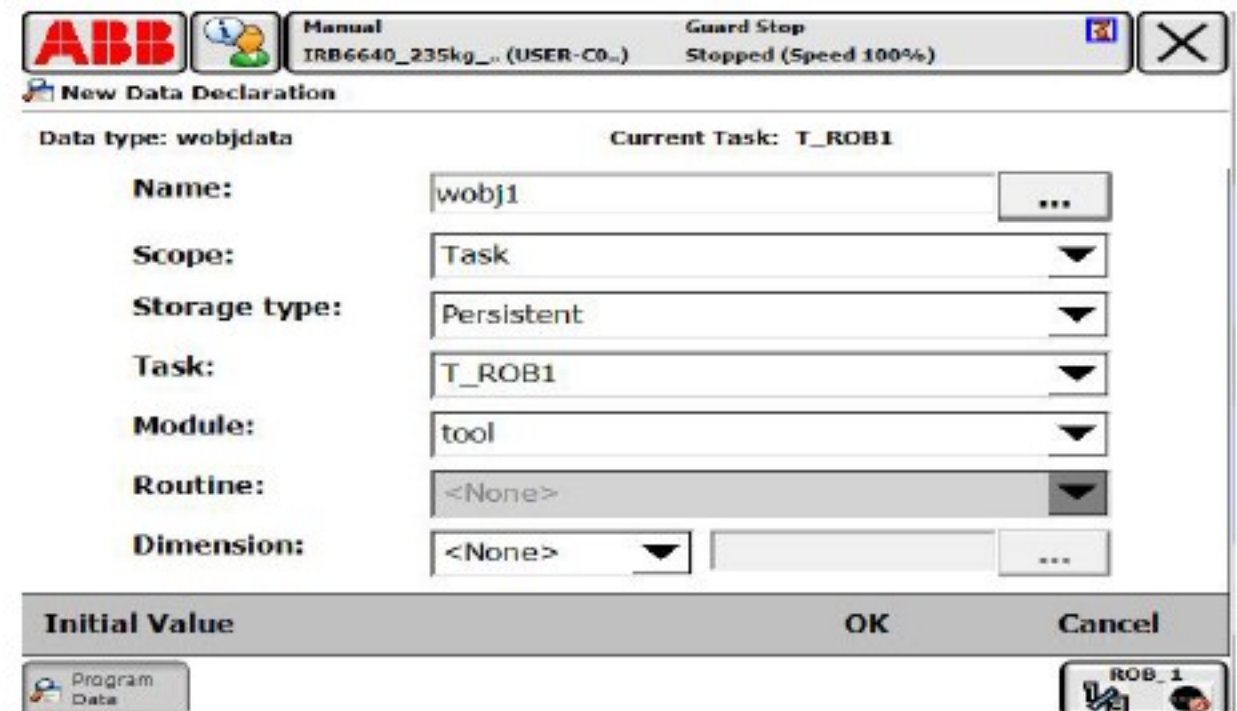
选择数据
wobjdata

注意：确认是在手动模式

五、定义工件坐标系



弹出以下窗口在名称处 (name) 可以修改名称



新建的
wobj1

五、定义工件坐标系



Manual
IRB6640_235kg_.. (USER-CO..)

Guard Stop
Stopped (Speed 100%)





Data of type: wobjdata

Active filter:

Select the data you want to edit.

Scope: RAPID/T_ROB1

Change Scope

Name	Value	Module	
wobj0	[FALSE,TRUE,"",[[0...	BASE	Global
wobj1	[FALSE,TRUE,"",[[0...	tool	Task

删除
更改声明
更改值
复制
定义
编辑



New...

Edit

Refresh

View Data Types

Program Data

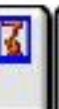
ROB_1

弹出以下窗口，并在user methhod处选择3点法



Manual
IRB6640_235kg_.. (USER-CO..)

Guard Stop
Stopped (Speed 100%)





Program Data - wobjdata - Define

Work Object Frame Definition

Work object: wobj1

Active tool: tool0

Select a method for each frame, modify the positions and tap OK.

User method 3 points

Object method No Change

Point	Status
User Point X 1	-
User Point X 2	-
User Point Y 1	-

选择新建工具坐标（蓝色光标），点击Edit,弹出窗口，选择定义

Positions

Modify Position

OK

Cancel

Program Data

ROB_1

五、定义工件坐标系

手动操纵机器人，使TCP点分别与点X1相碰，使用功能键 **修改位置** 记录机器人相应位置。

ABB Manual IRB6640_235kg... (USER-CO..) Guard Stop Stopped (Speed 100%)

Program Data - wobjdata - Define

Work Object Frame Definition

Work object: wobj1 Active tool: tool0

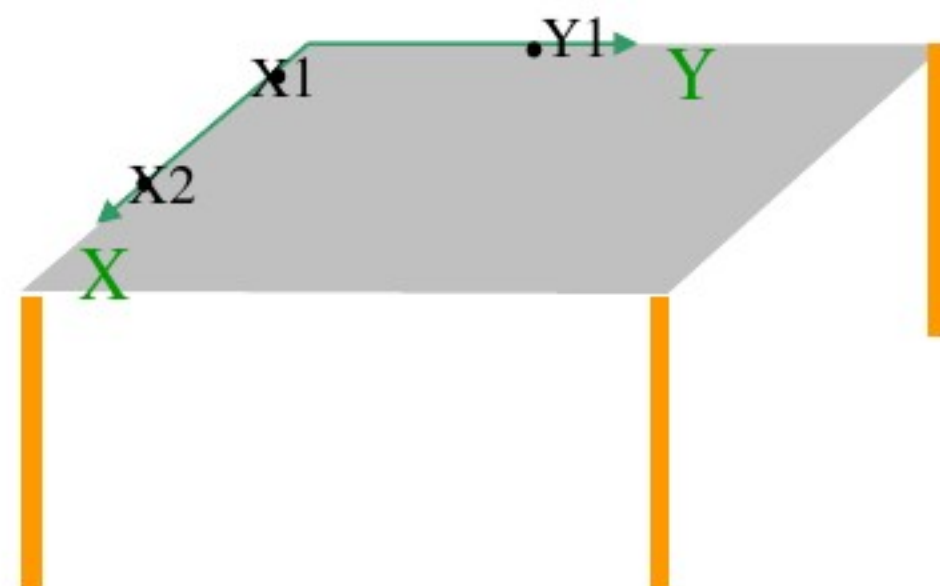
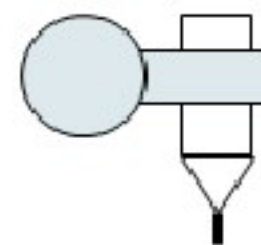
Select a method for each frame, modify the positions and tap OK.

User method 3 points Object method No Change

Point	Status
User Point X 1	Modified
User Point X 2	Modified
User Point Y 1	Modified

Positions Modify Position OK Cancel

T_ROB1 MainMod... Program Data ROB_1

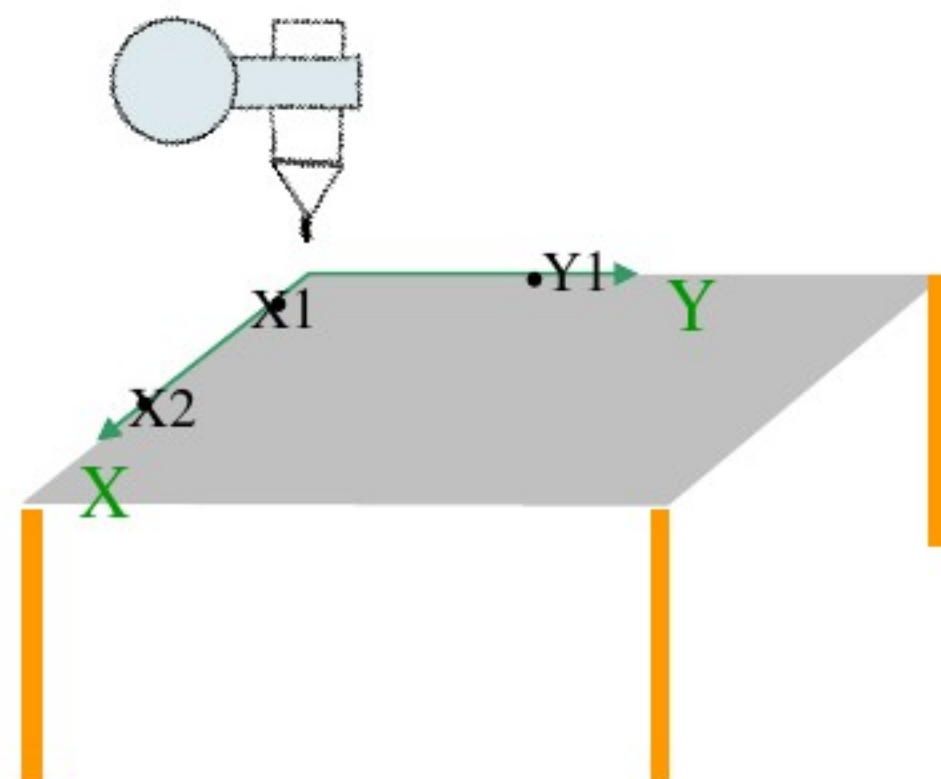


重复步骤上一步，使TCP点分别与点X2、点Y1相碰，并使用功能键 **修改位置** 记录机器人相应位置。最后用 **确定** 键确认。

五、定义工件坐标系

检测新建工件坐标

手动移动机器人，使用 **线性运动模式**，按照新定义的 **工件坐标 (wobj1)** 运动，进行验证。

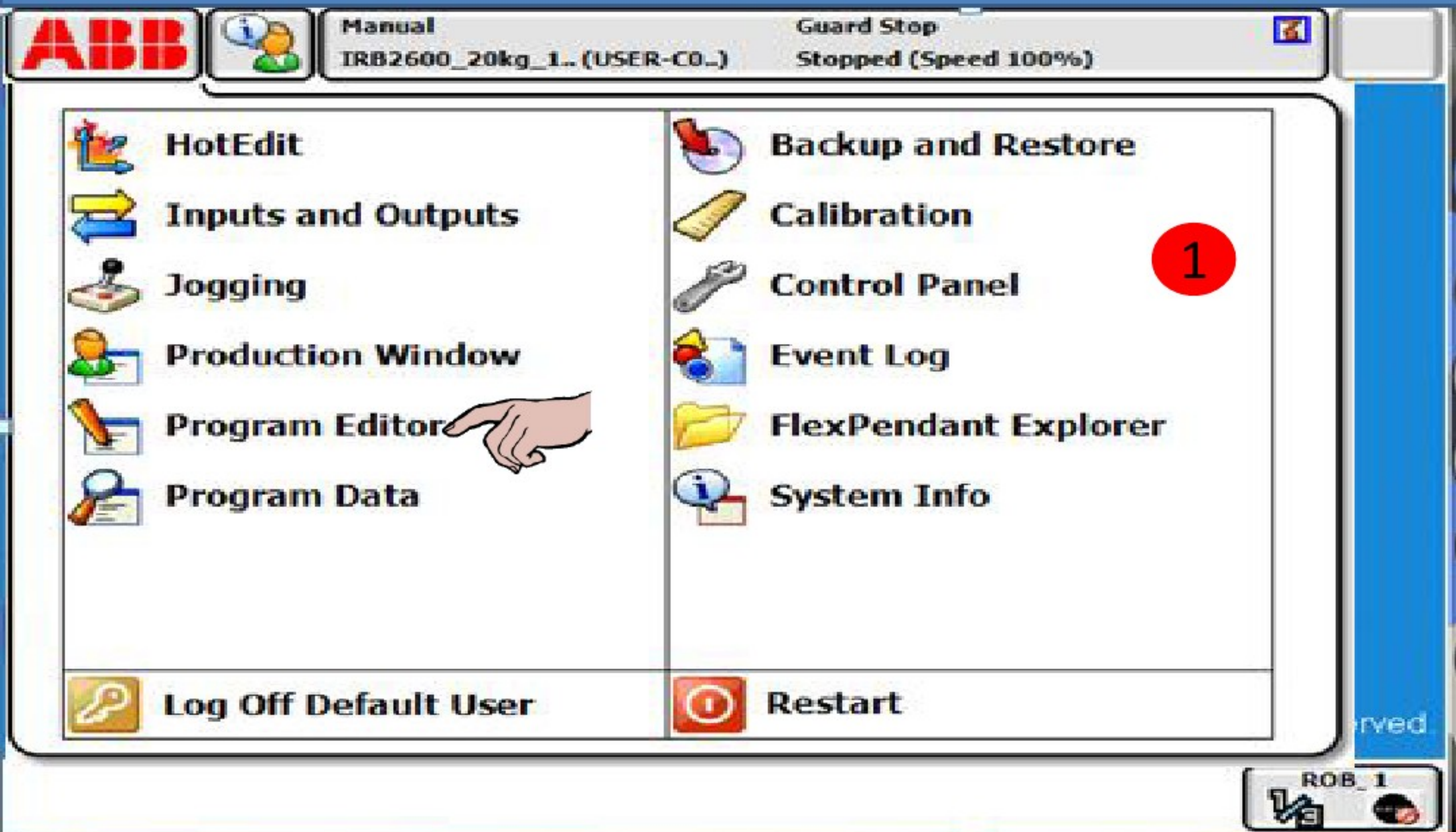


红色选择你新建的工件坐标



六、LoadData

1、点击ABB,在ABB菜单栏内点击程序编辑



六、LoadData

2、程序编辑窗口内点击例行程序

3、点击文件，出现一个窗口，在窗口内点击新建例行程序

ABB

Auto

Motors Off

IRB2600_20kg_1.. (USER-CO..)

Stopped (Speed 100%)

X

NewProgramName in T_ROB1/MainModule/main

Tasks and Programs

Modules

Routines

2

PROC main()

3

<SMT>

4

ENDPROC

5

ENDMODULE

2

Add Instruction

Edit

Debug

Modify Position

Hide Declarations

T_ROB1 MainMod...

ABB

Auto

Motors Off

IRB2600_20kg_1.. (USER-CO..)

Stopped (Speed 100%)

X

T_ROB1/MainModule

Routines

Active filter:

Name ▲	Module	Type
main()	MainModule	Procedure

新建例行程序...

复制例行程序...

移动例行程序...

更改声明...

重命名...

删除例行程序...

文件

3

Show Routine

Back

T_ROB1 MainMod...

六、LoadData

4、点击ABC...给新建例行程序命名，再点击OK

5、点击新建的例行程序，再点击show routine

ABB

Manual

IRB2600_20kg_1.. (USER-CO..)

Guard Stop

Stopped (Speed 100%)

×

ABB

Manual

IRB2600_20kg_1.. (USER-CO..)

Guard Stop

Stopped (Speed 100%)

×

New Routine - NewProgramName in T_ROB1/MainModule

T_ROB1/MainModule

Routine Declaration

Name: Routine2

Type: Procedure

Parameters: None

Data type: num

Module: MainModule

Local declaration:

4

Error Handler: ☐

Undo Handler: ☐

Backward Handler: ☐

ABC...

...

...

MainModule

OK

Cancel

Routines

Active filter:

Name	Module	Type
main()	MainModule	Procedure
Routine2()	MainModule	Procedure

5

Show Routine

Back

T_ROB1 MainMod...

ROB_1

T_ROB1 MainMod...

ROB_1

六、LoadData

6、点击<SMT>（如果蓝色光标在此处，可以直接点击AddInstruction），再点击Add Instruction，然后点击MoveAbsJ

7、选择位置点，点击debug选择view value

ABB Manual IRB2600_20kg_1.. (USER-CO..) Guard Stop Stopped (Speed 100%) X

NewProgramName in T_ROB1/MainModule/Routine2

Tasks and Programs	Modules	Routines
PROC Routine2() <SMT> ENDPROC ENDMODULE		
Common		
:=		Compact IF
FOR		IF
MoveAbsJ		MoveC
MoveJ		MoveL
ProcCall		Reset
RETURN		Set
<-- Previous		Next -->

6

Add Instruction Edit Debug Modify Position Hide Declarations

T_ROB1 MainMod...

ABB Manual IRB2600_20kg_1.. (USER-CO..) Guard Stop Stopped (Speed 100%) X

NewProgramName in T_ROB1/MainModule/Routine2

Tasks and Programs	Modules	Routines
PROC Routine2() MoveAbsJ * \NoEOffs, ENDPROC		
PP to Main		
PP to Routine...		
Cursor to MP		
Call Routine...		
View Value		
View System Data		
PP to Cursor		
Cursor to PP		
Go to position		
Cancel Call Rout.		
Check Program		
Search Routine		

7

Add Instruction Edit Debug Modify Position Show Declarations

T_ROB1 MainMod...

六、LoadData

8、把1-6轴的位置数据全部输入0，点击OK

ABB Manual IRB2600_20kg_1.. (USER-CO..) Guard Stop Stopped (Speed 100%) X

Edit

Name: *

Tap a field to edit the value.

Name	Value
rax_1 :=	0
rax_2 :=	0
rax_3 :=	0
rax_4 :=	0
rax_5 :=	30
rax_6 :=	0

OK Cancel

T_ROB1 MainMod... ROB.1

完成以上操作后，在示教器上手动上使能，
按运行键让机器人走到机械零位
走到零位后，必须确认机器人本体机械零位
是否在位置

六、LoadData

9、在新建的例行程序窗口中，点击“调试”，再点击“调用例行程序....”

10、选择例行程序“Load Identify”点击“转到”

The image displays two side-by-side screenshots of the ABB robot software interface, illustrating the steps to call a service routine.

Left Screenshot (Step 9): The 'PROC Routine1()' window is shown. The code includes 'PROC Routine1()', 'MoveAbsJ *\NoEOffs', 'ENDPROC', and 'ENDMODULE'. A red circle with the number '9' highlights the '调试' (Debug) button in the bottom toolbar. A hand icon points to the '调用例行程序...' (Call Routine...) button in the '例行程序' (Routine) menu.

Right Screenshot (Step 10): The '调用服务例行程序' (Call Service Routine) dialog is shown. The '选定的例行程序:' (Selected Routine:) field displays 'LoadIdentify'. Below, a list of routines includes 'Bat_Shutdown', 'CalPendelum', 'Commutation', 'Linked_m', 'LoadIdentify' (highlighted), 'ManLoadIdentify', 'ServiceInfo', and 'SkipTaskExec'. A red circle with the number '10' highlights the '转到' (Go to) button in the bottom toolbar. A hand icon points to the '转到' button.

注：确定指针是否在该窗口

六、LoadData

11、在窗口内出现一些程序，上使能按连续运行键



手动
YIG (USER-COSPUNBQI)

电机开启
正在运行 (速度 100%)



T_ROB1 内的<未命名程序>

任务与程序	模块	例行程序
1	MODULE BASEFUN (SYSMODULE, VIEWONLY)	
2		
3	PROC LoadIdentify()	
4	! Service routine Load Identification	
5	!*****	
6		
7	LoadIdentifyProc;	
8		
9	UNDO	
10	LoadIdTerminate;	
11	ENDPROC	
12		
13	PROC ManLoadIdentify()	
14	! Service routine for ext. manipulators Load Identification	
15	!*****	
16		
17	ManLoadIdProc \DoExit:=TRUE;	
18		

添加指令 编辑 调试 修改位置 隐藏声明

T_ROB1

ROB_1

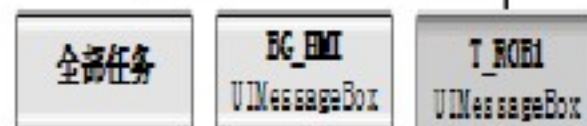
11



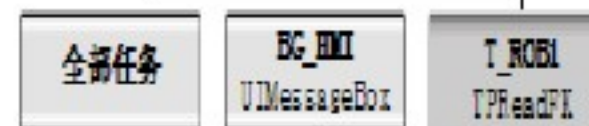
六、LoadData

12、按住使能，窗口内点击“OK”

13、按住使能，在窗口内选择当前工具Tool（PayLoad：带工件；tool:无工件，只有工具）



12



13

Cancel

OK

Cancel

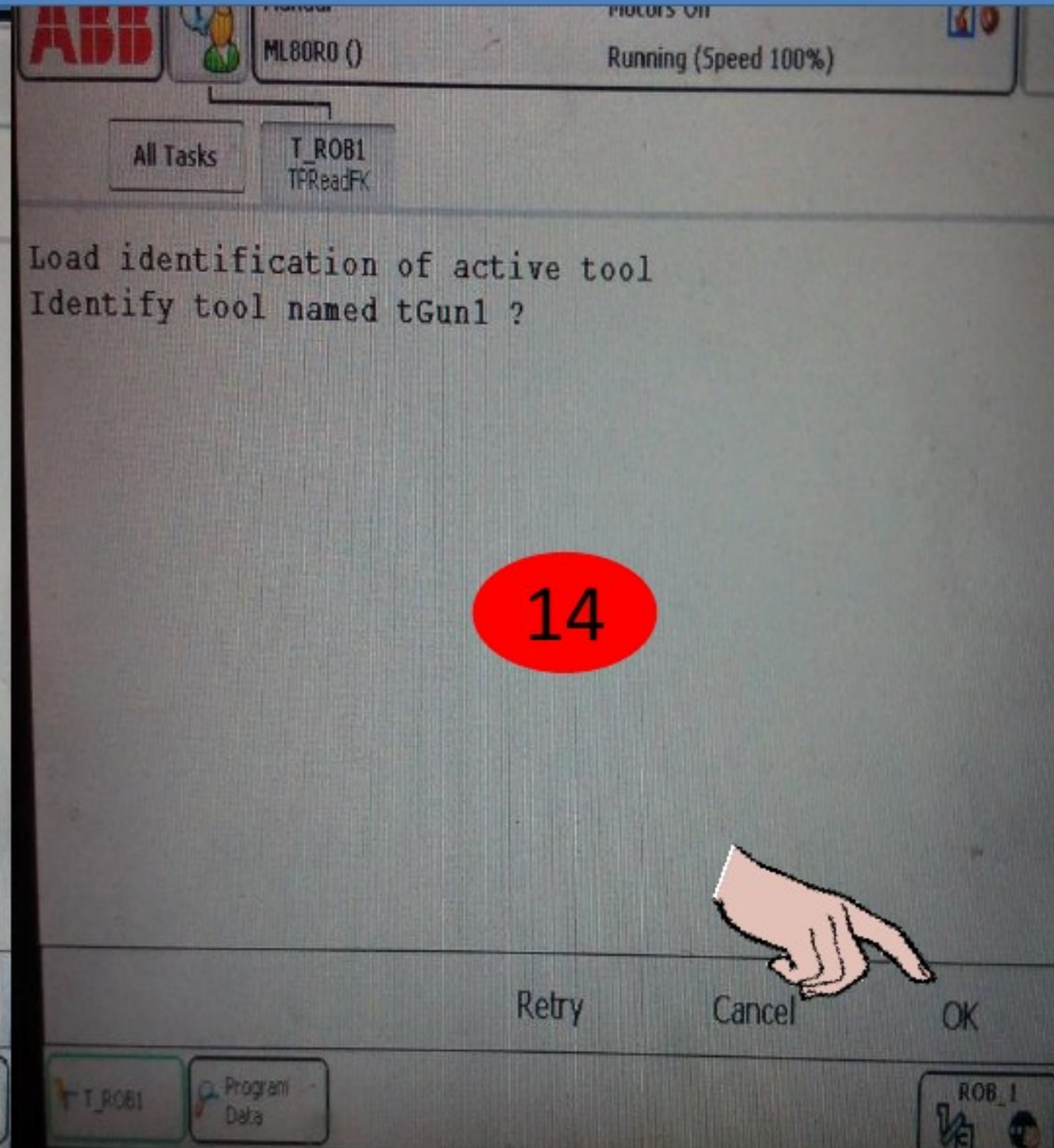
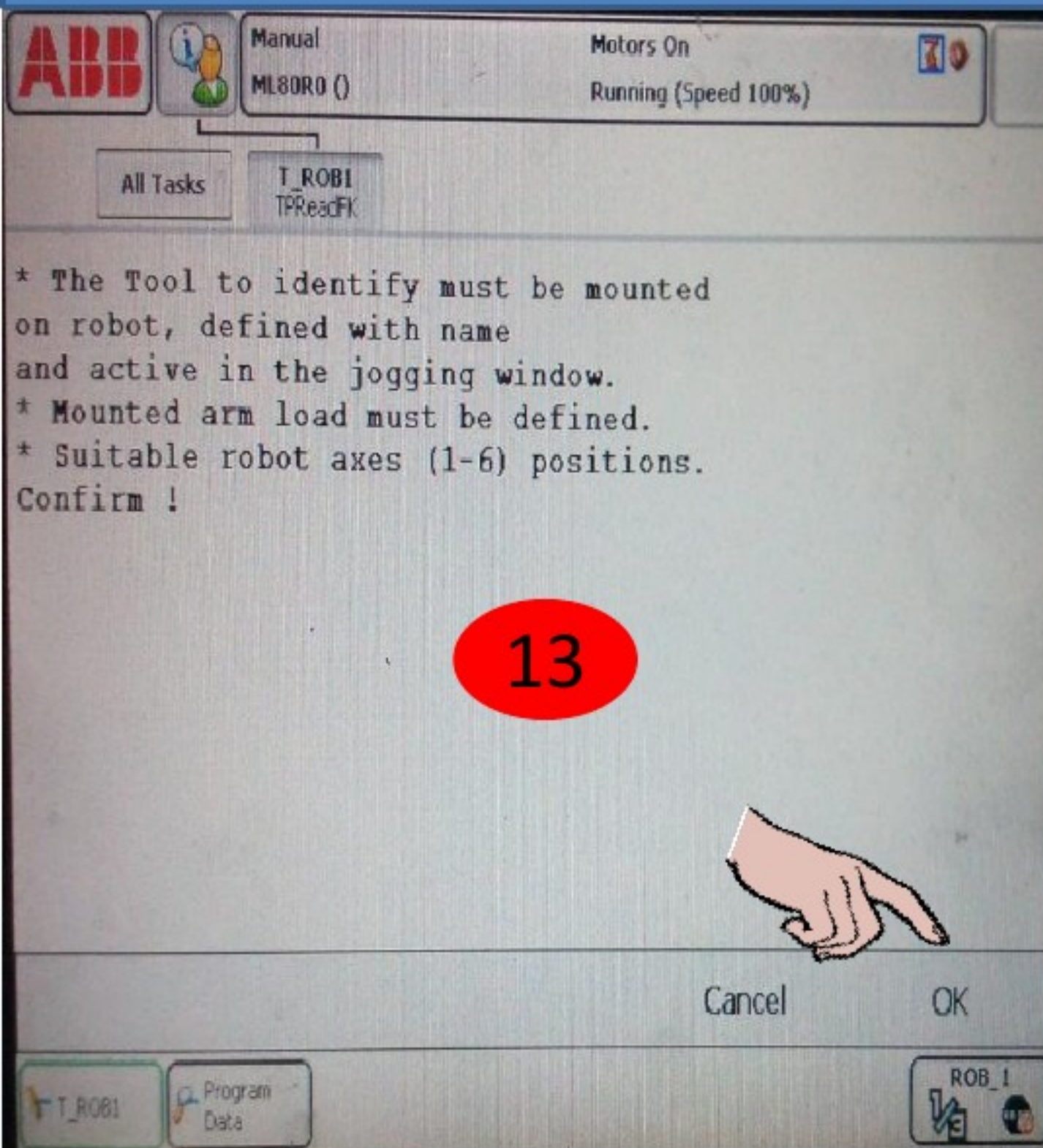
PayLoad

Tool

六、LoadData

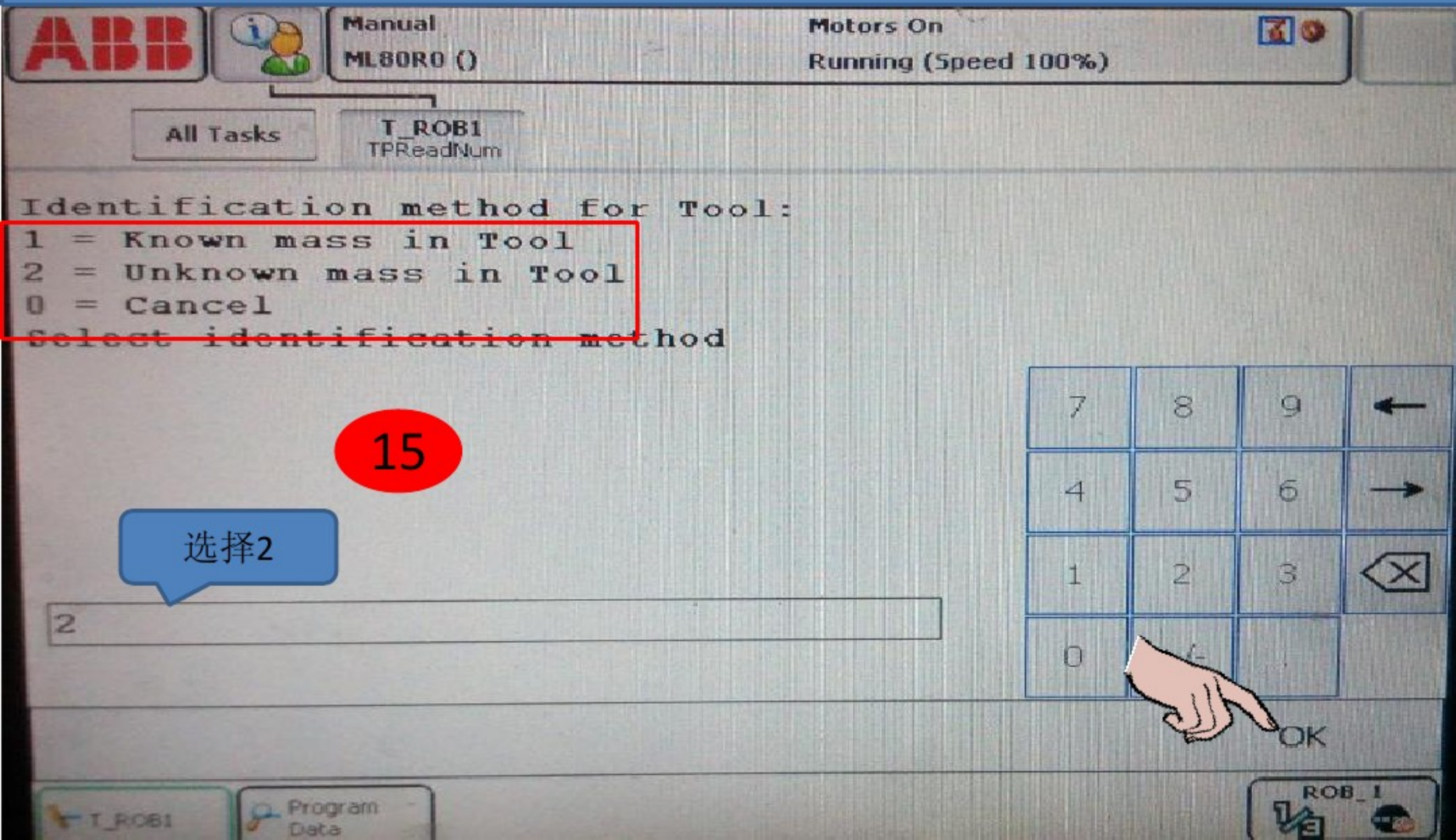
13、按住使能，确认窗口内的内容，点击“OK”

14、按住使能，确认内容后，点击“OK”



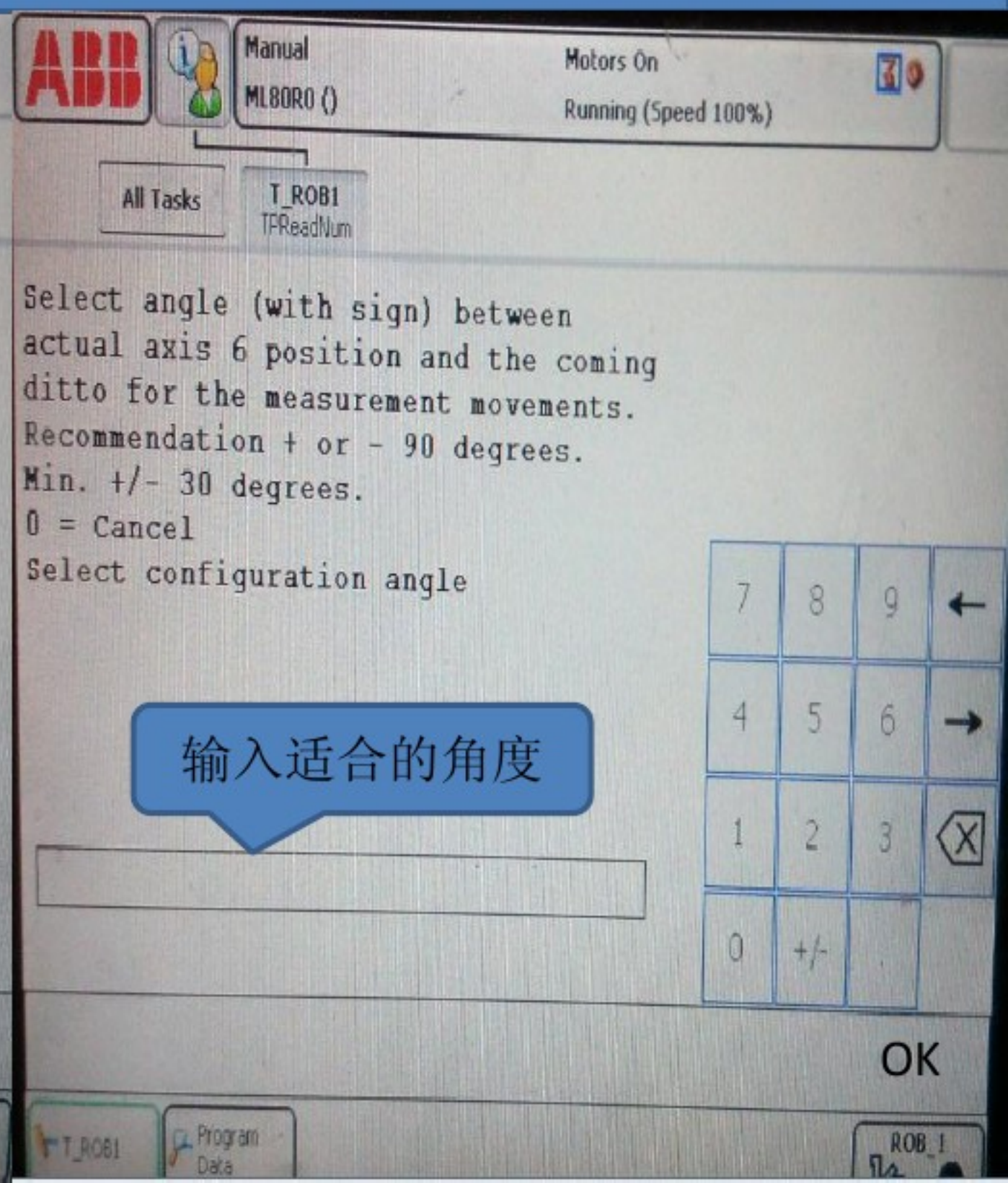
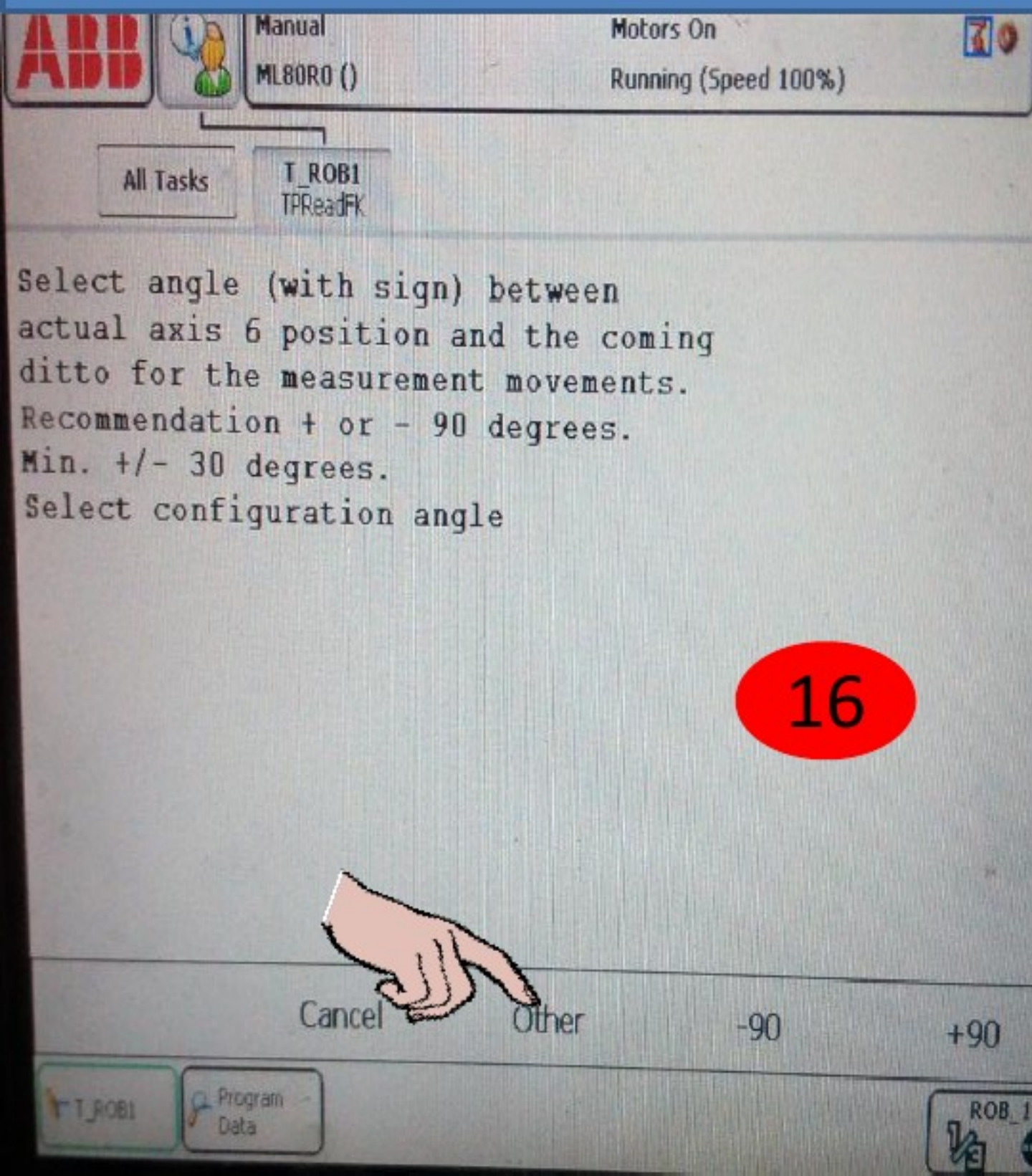
六、LoadData

15、按住使能，确认内容，选择2，点击“OK”（0、1、2内容不同，需正确选择）



六、LoadData

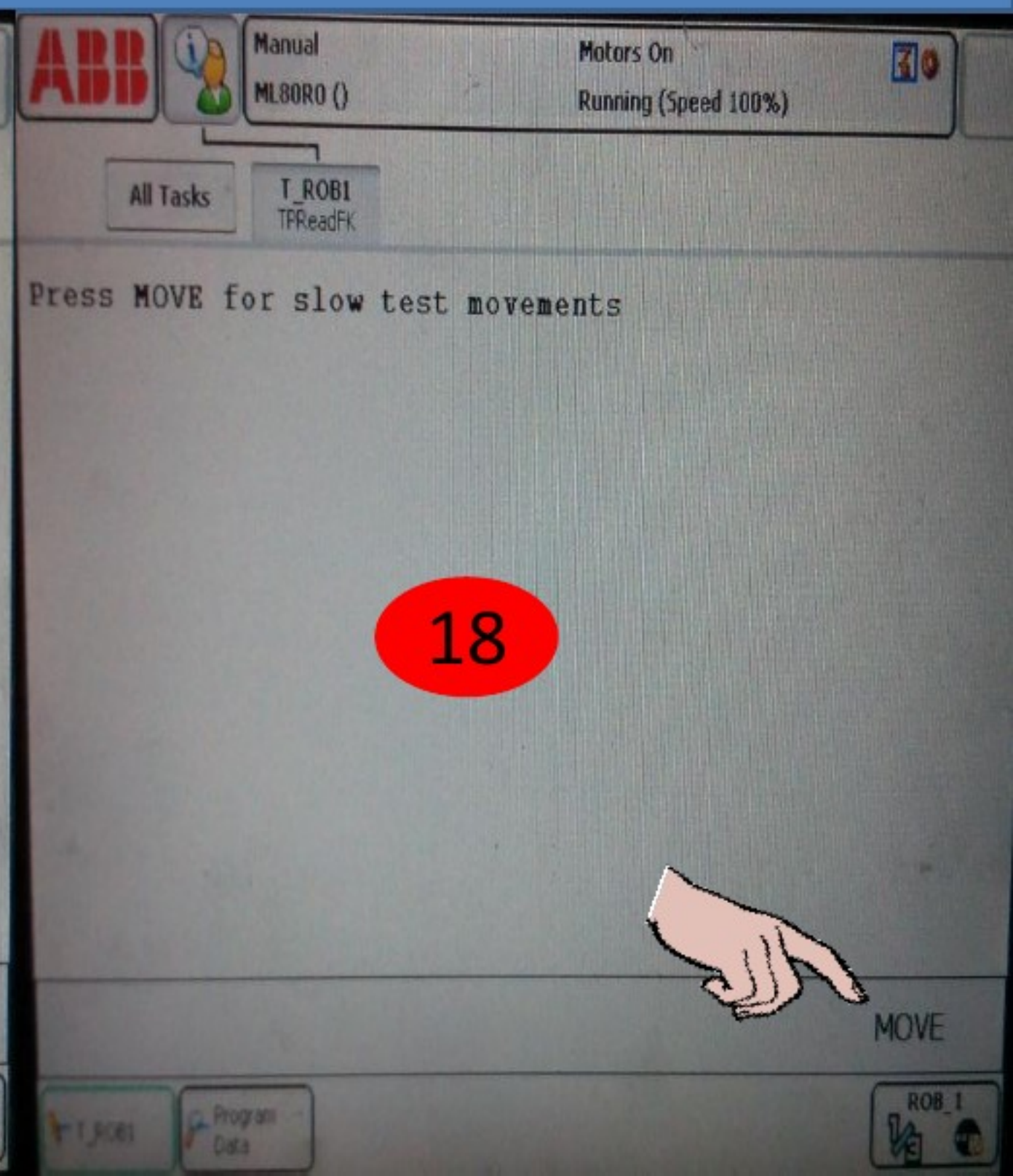
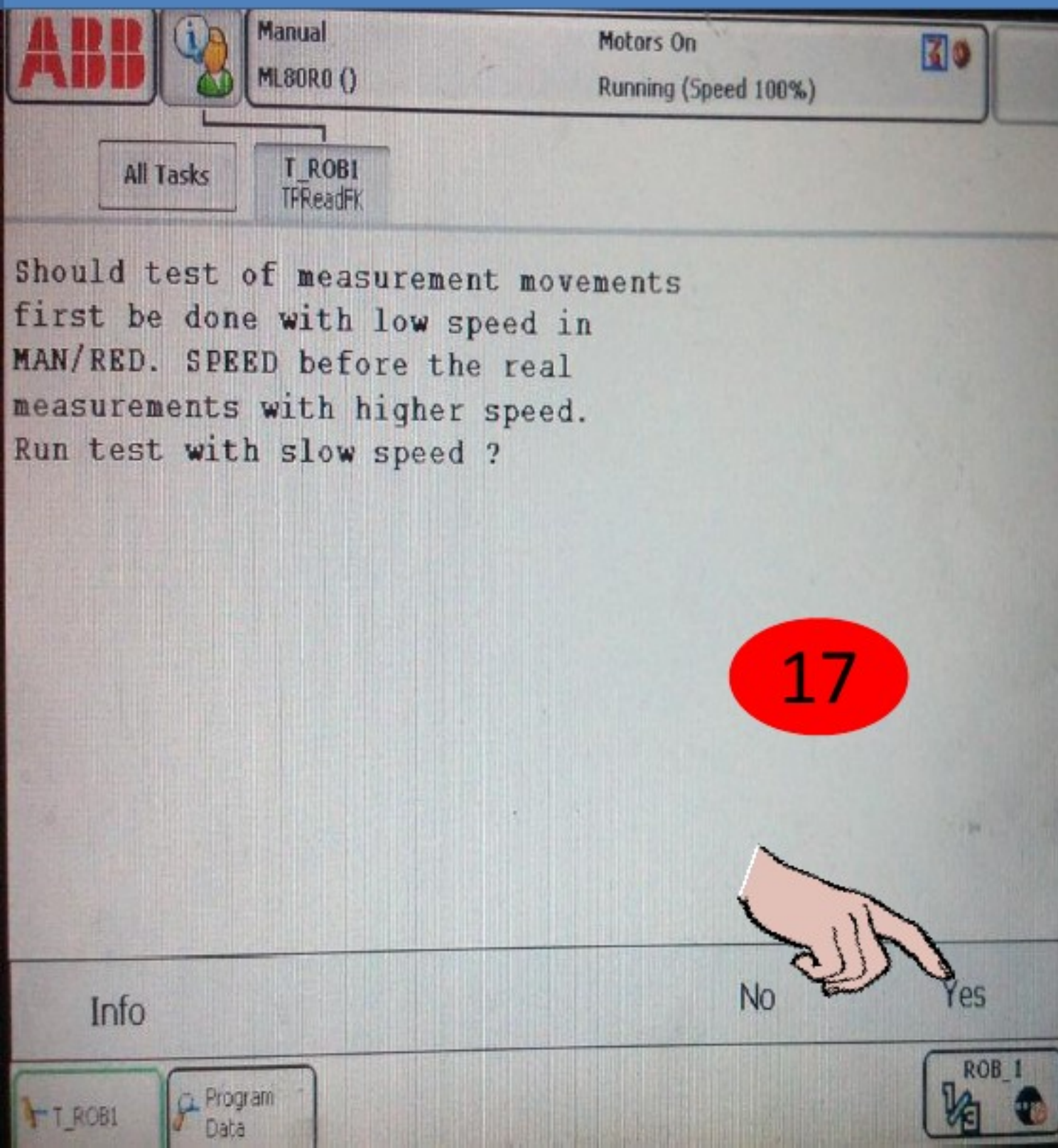
16、选择运行的角度（90为Load最好的角度，根据现场需求选择适合的角度），点击“Other”输入适合的角度，点击“OK”



六、LoadData

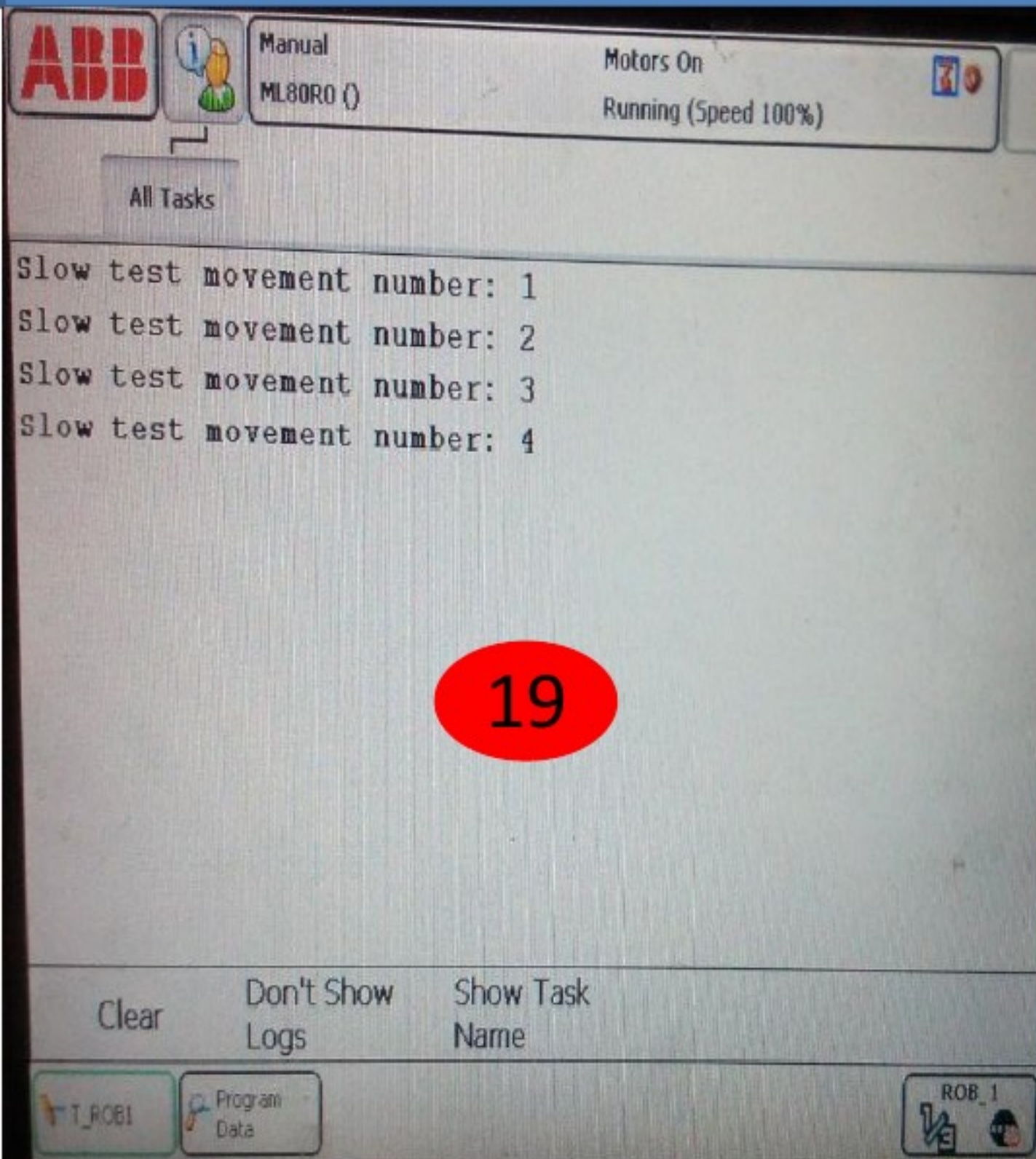
17、按住使能，确认内容后，点击“YES”

18、按住使能，确认内容后、点击“MOVE”

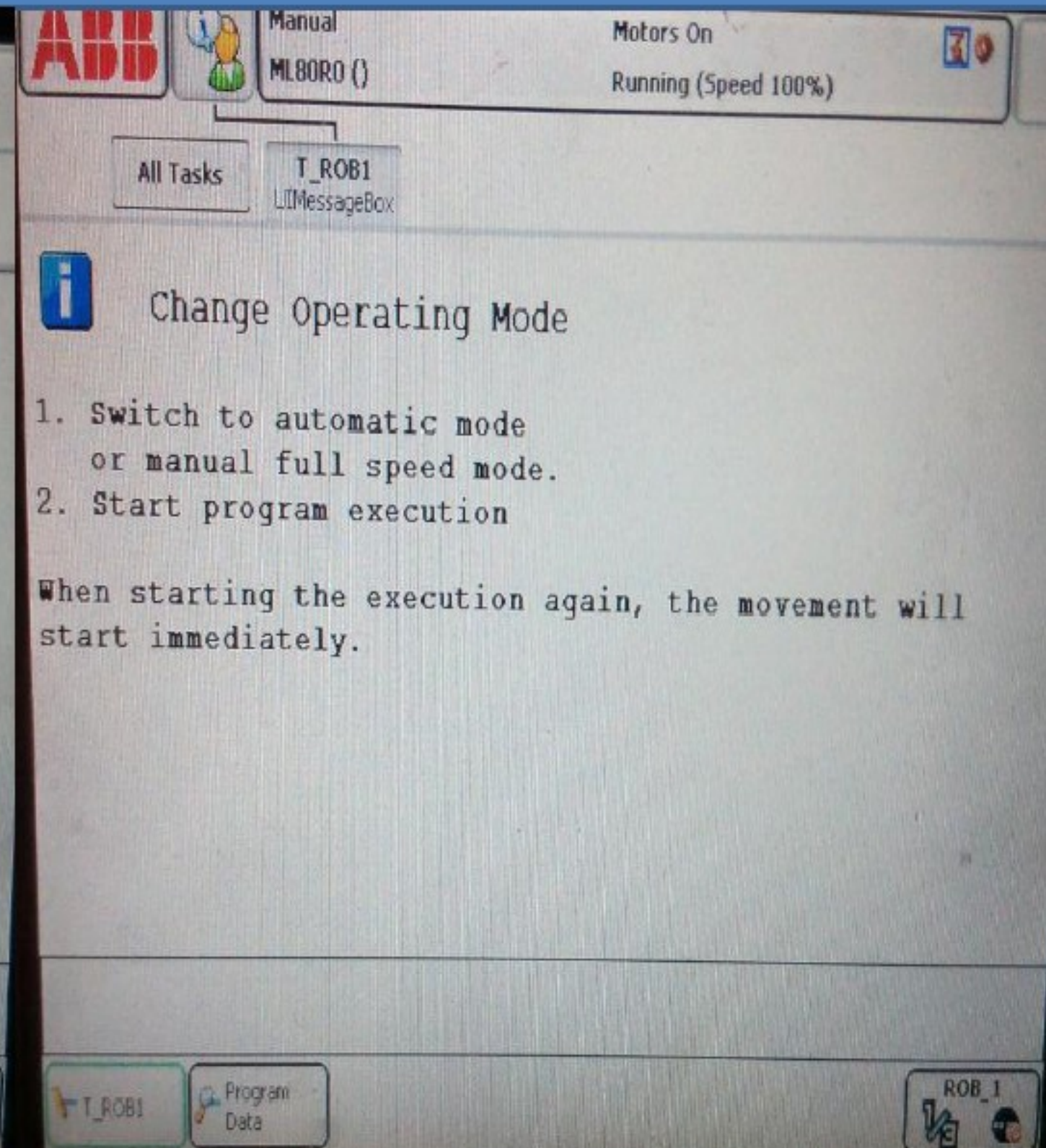


LoadData

19、按住使能，机器人开始按照设定的角度运行，在运行的过程中，会弹出  Change Operating Mode，在控制柜上选择手动100%

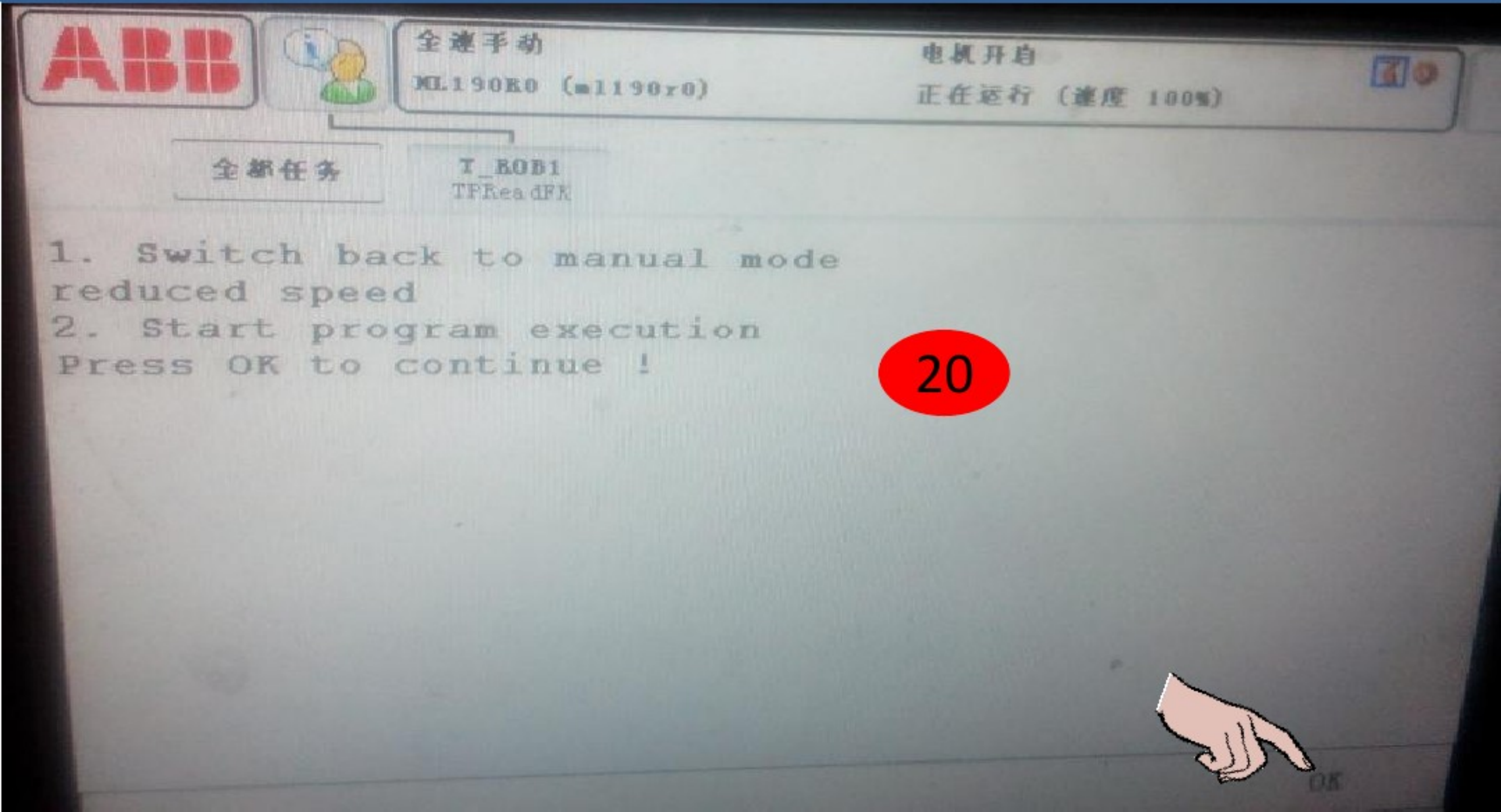


19



LoadData

20、按住使能，并按住连续运行键，机器人开始测量，运行的过程中，会弹出下图窗口，点击“OK”(需要其他人帮忙)



六、LoadData

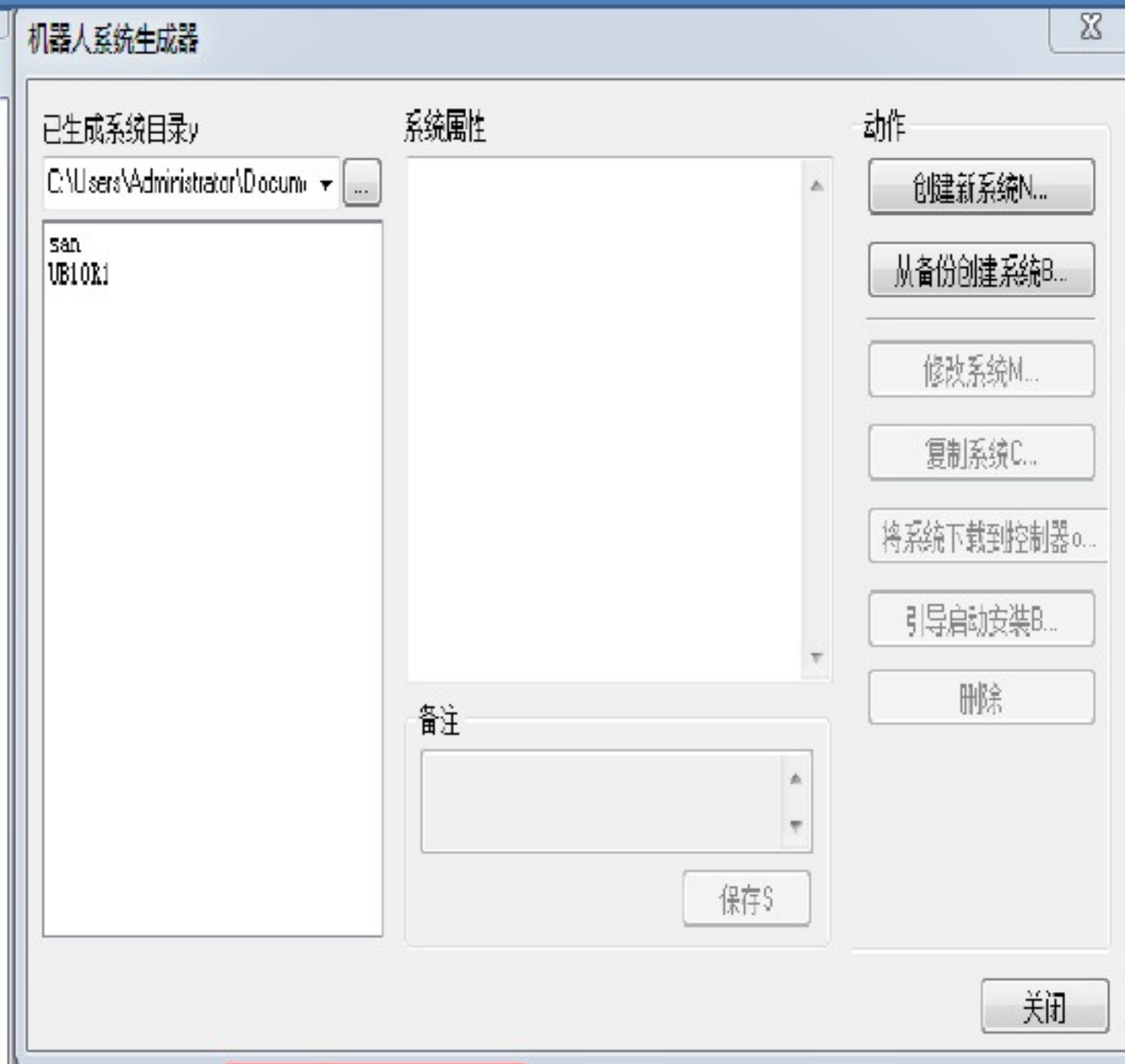
Load Data完毕

七、IRC5 控制器系统生成

- 1、安装ABB Robotstudio软件
- 2、打开ABB Robotstudio软件，点击“在线”，在创建并制作机器人系统中点击“系统生成器”（图一）
- 3、点击“从备份创建系统B.....”（图二）



图一

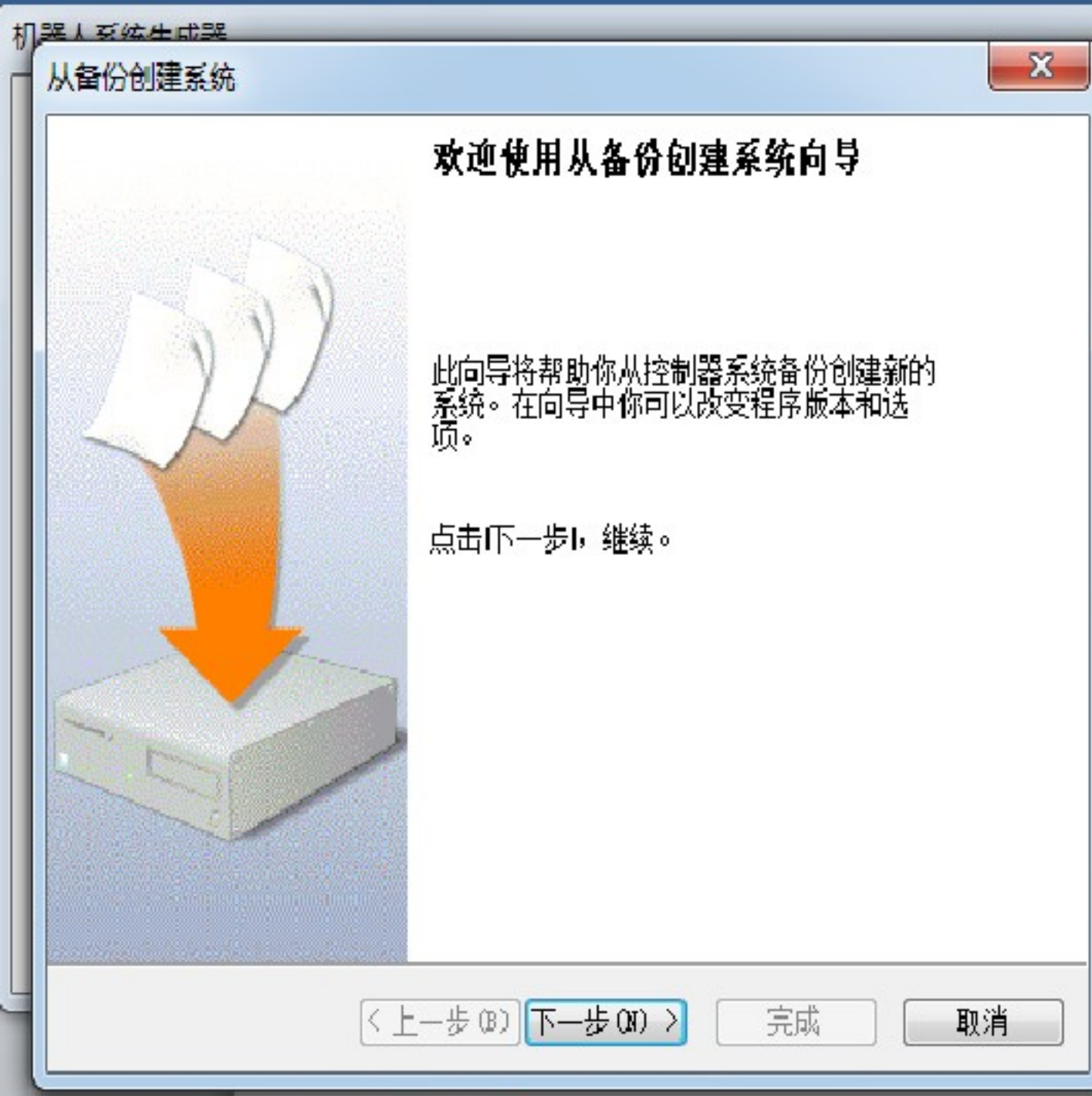


图二

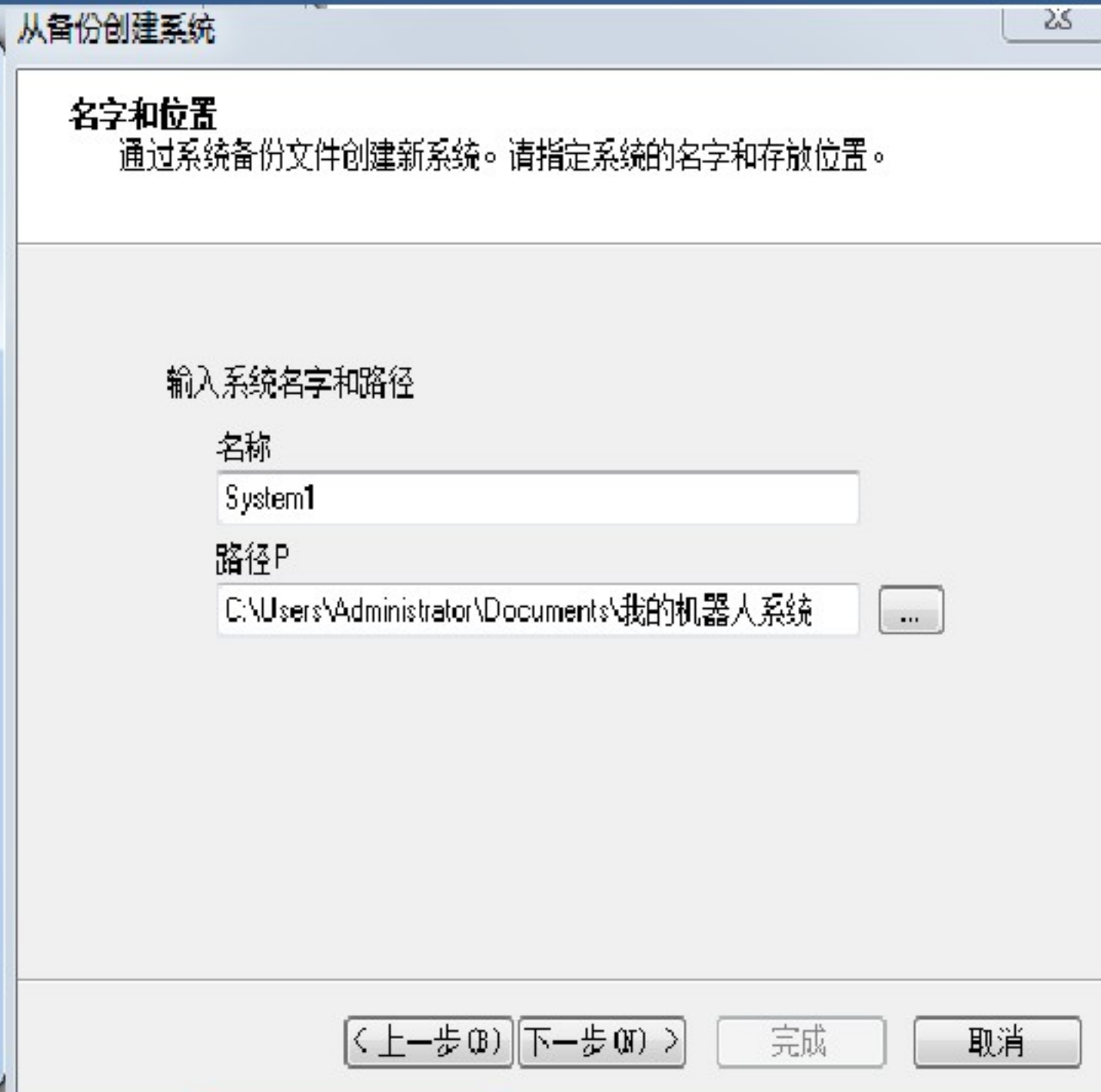
七、IRC5 控制器系统生成

4、在（图三）中点击下一步

5、在弹出的窗口中，给新创建的系统命名并选择该系统文件的存储路径（不支持中文字符）。然后点击按钮 下一步（图四）。



图三



图四

七、IRC5 控制器系统生成

6、在弹出的窗口中，选择用来创建新系统的备份（备份目录B中选择）。然后点击按钮下一步（图五）

7、在（图六）中点击下一步



图五

图六

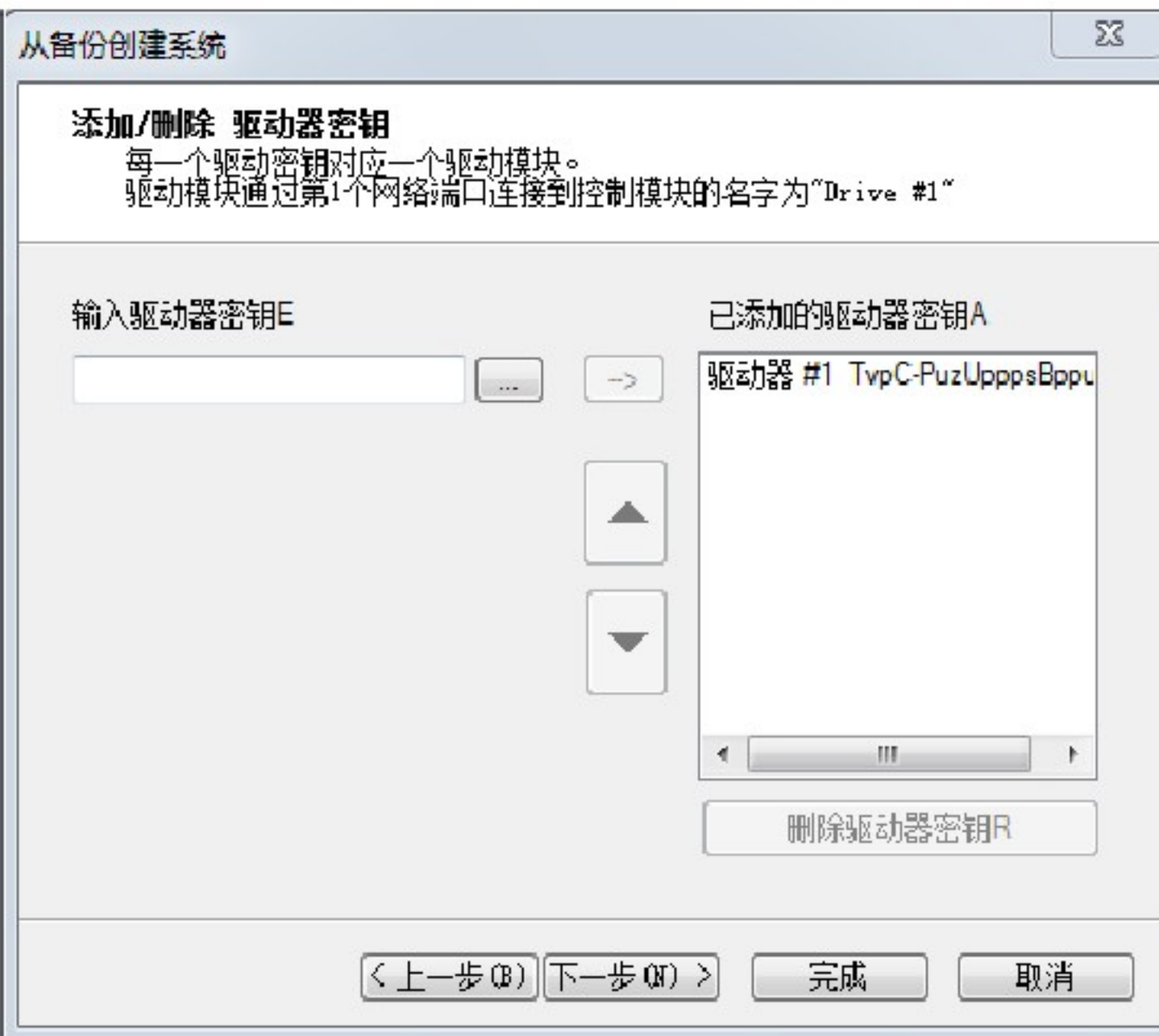
七、IRC5 控制器系统生成

8、在（图七）窗口中，选择是否当前版本（选择是，不选择 ☐ 不，要替换），点击下一步

9、在（图八）窗口中，输入驱动器Key，点击下一步



图七



图八

七、IRC5 控制器系统生成

9、在（图九）中，输入增加项的Key，点击下一步

10、在（图十）中，选择所需功能，点击下一步



图九

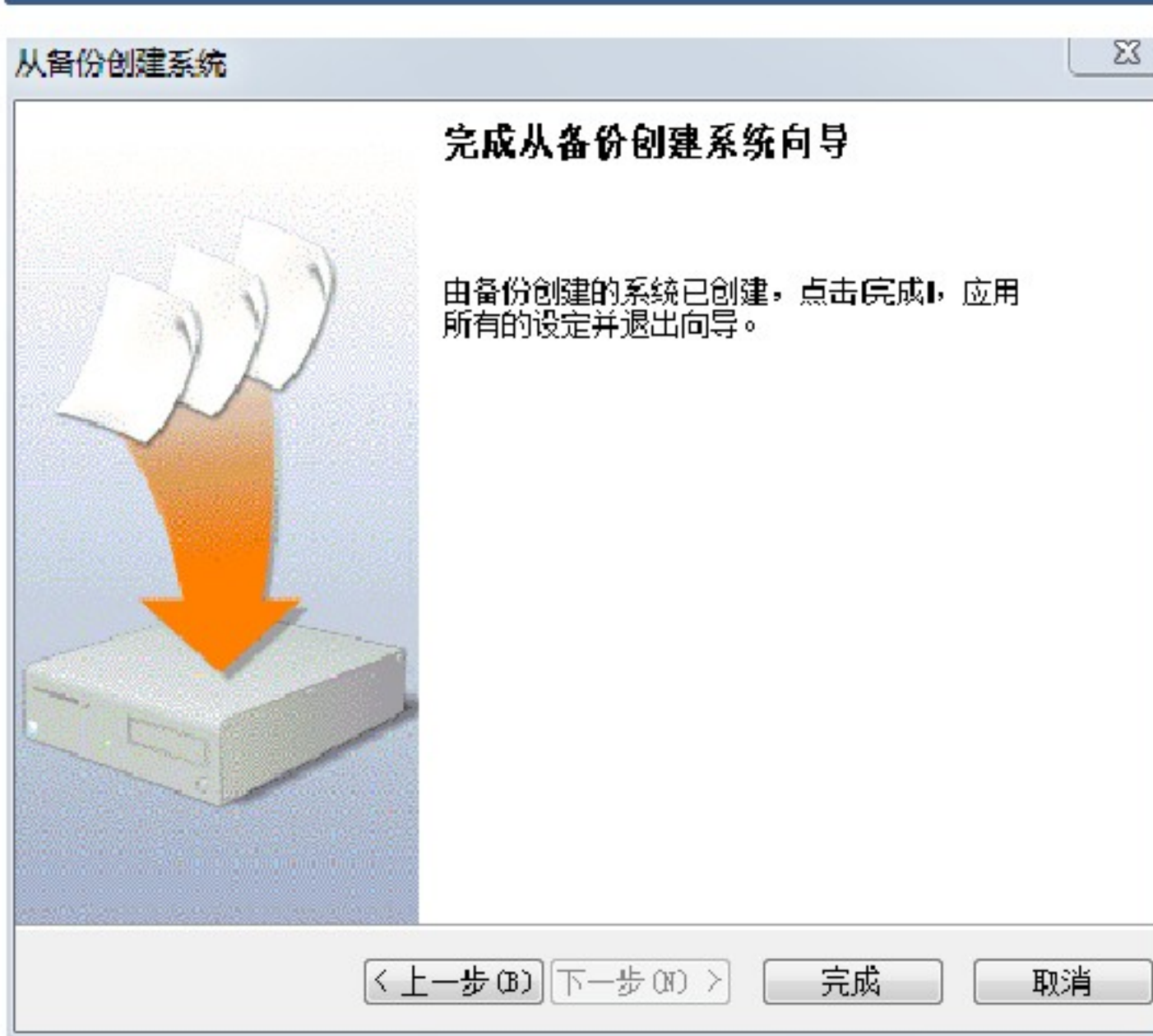


图十

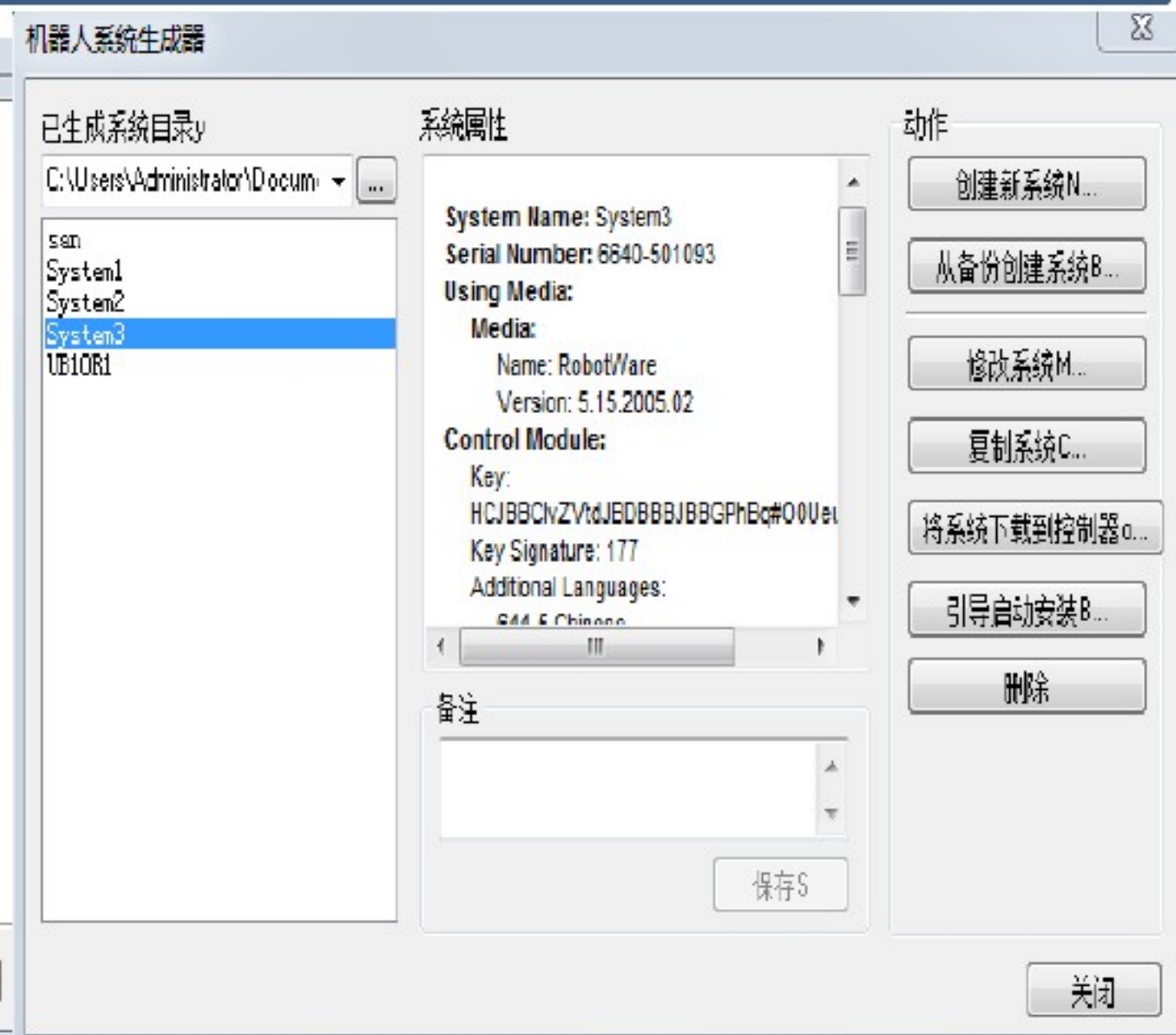
七、IRC5 控制器系统生成

12、点击（图十三）按钮 完成，弹出（图十四）窗口

13、可在（图十四）窗口中，动作栏内可做一些系统操作（如修改系统、复制系统、引导启动安装、删除等），最后将系统下载到控制器内



图十三



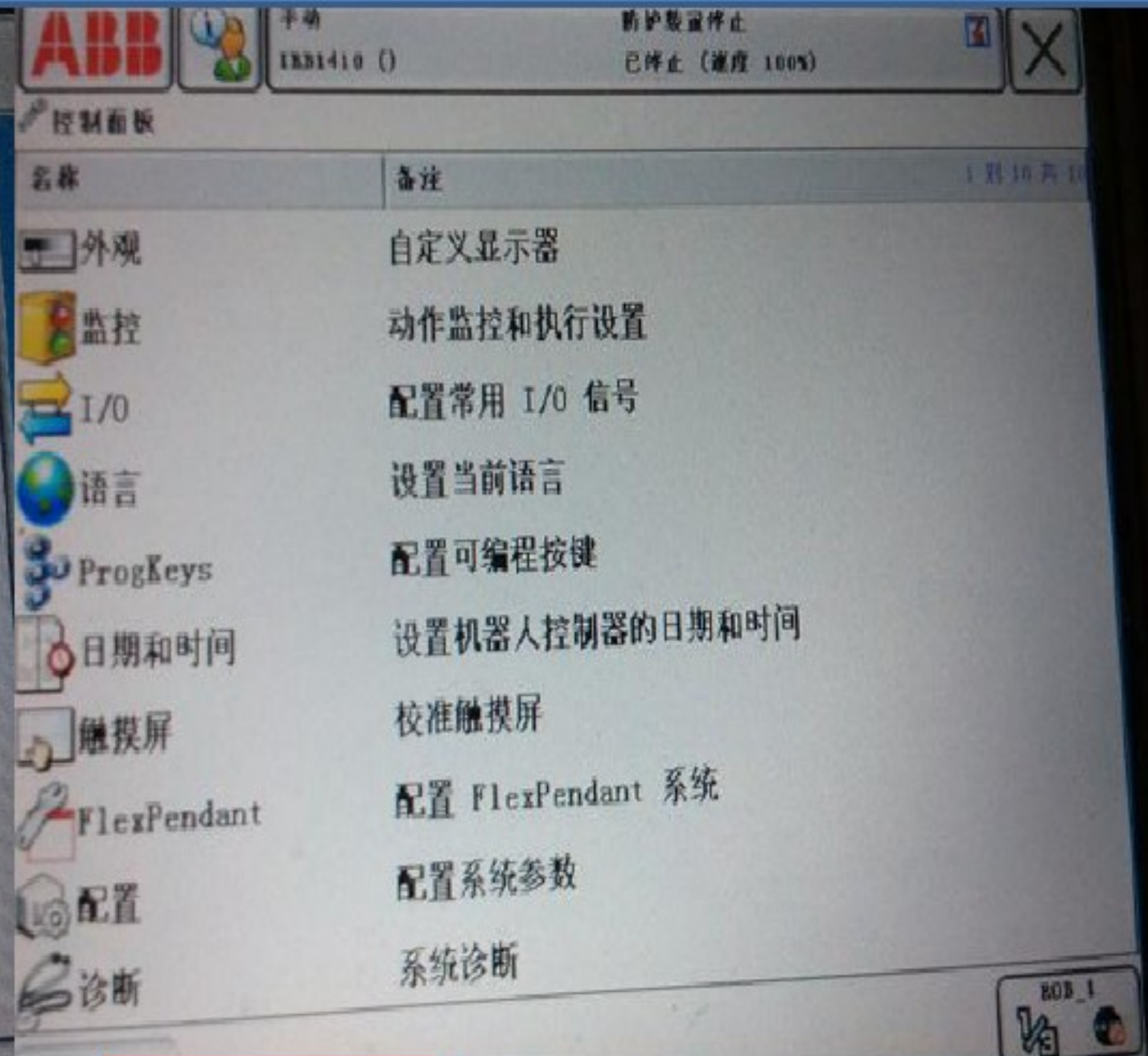
图十四

八、定义I/O总线

- 1、点击“ABB”出现（图一）窗口，在窗口里点击“控制面板”出现（图二）窗口
- 2、在（图二）窗口中，点击配置出现（图三）窗口



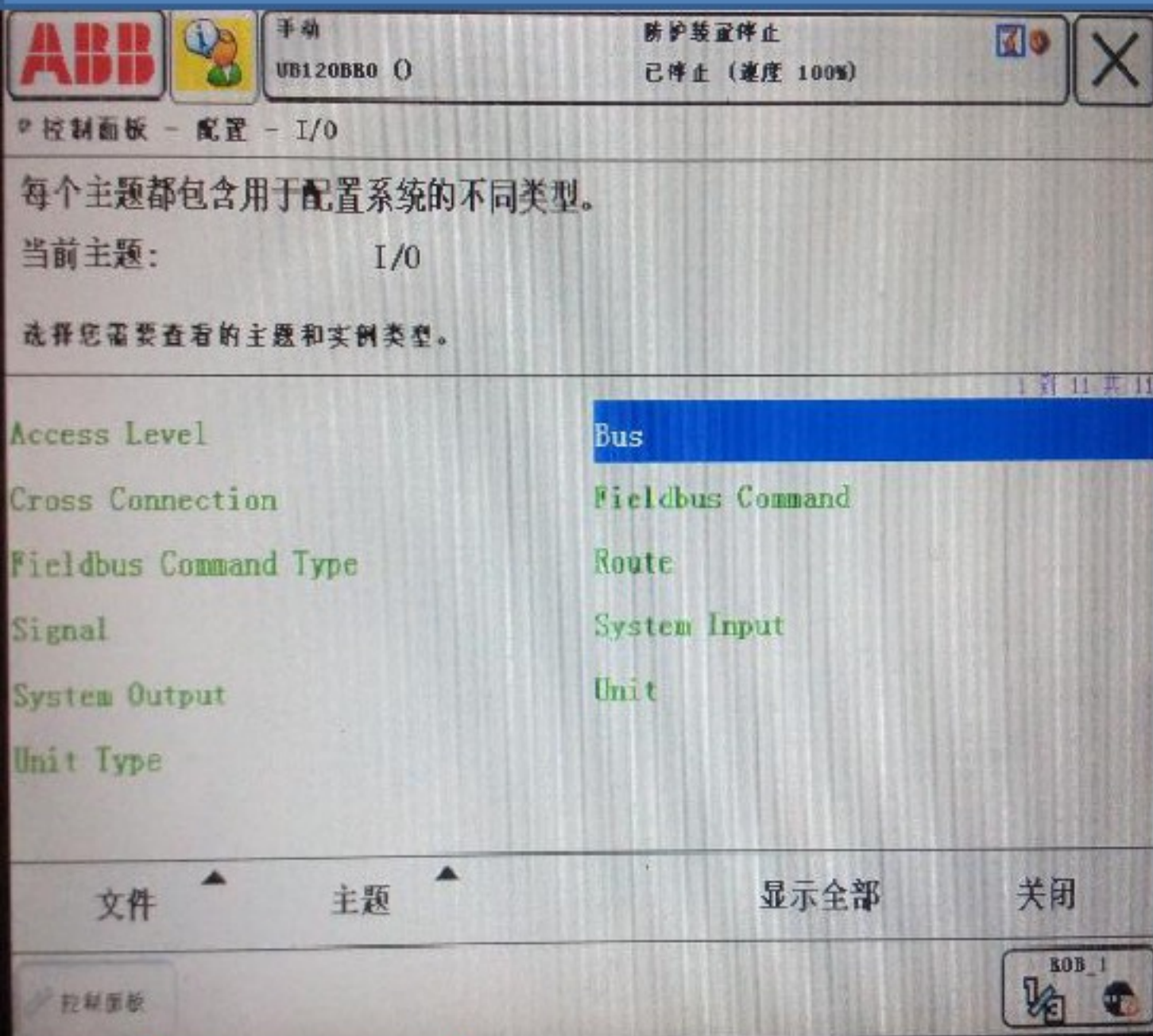
图一



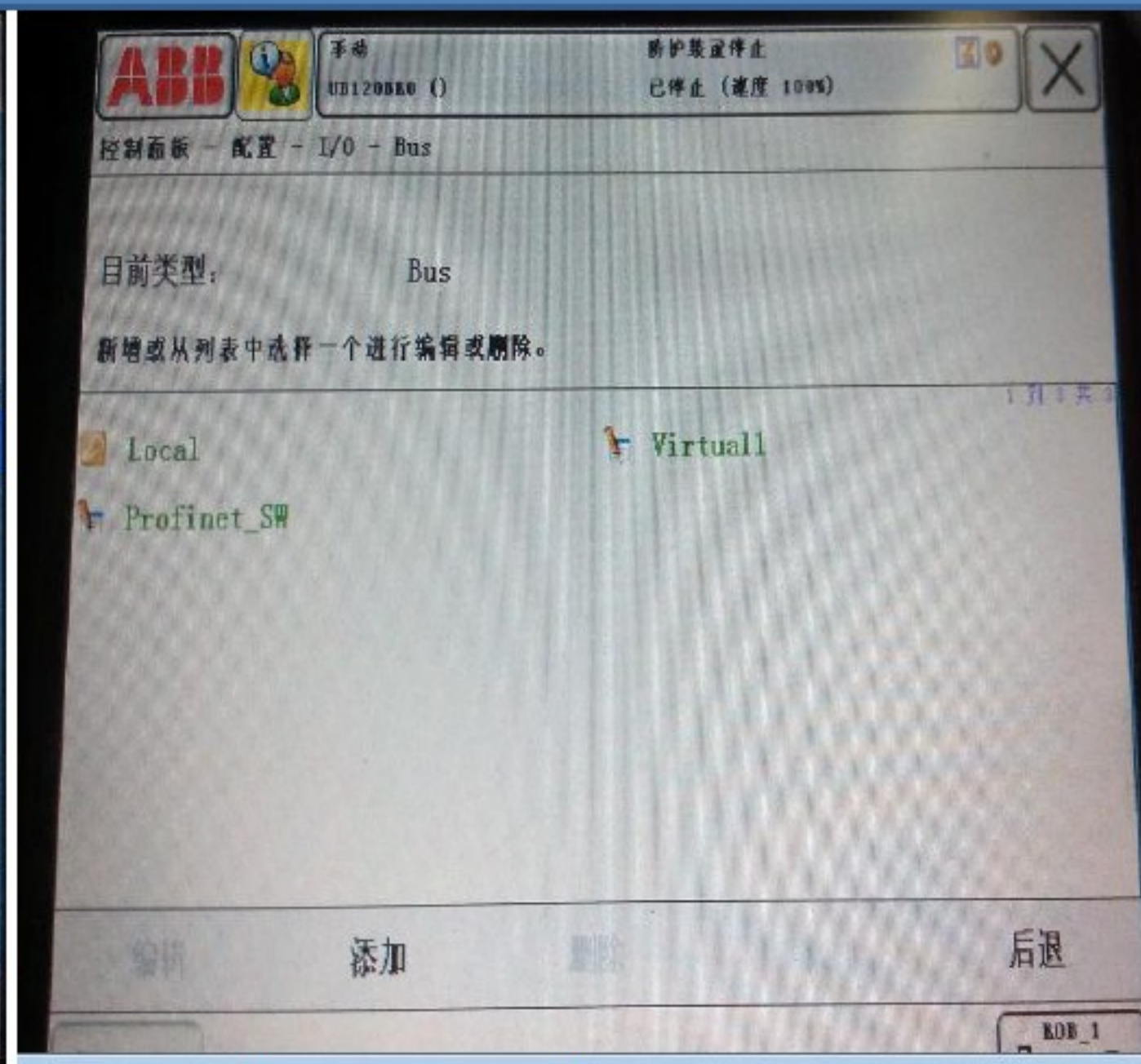
图二

八、定义I/O总线

- 3、在（图三）中，双击“BUS”，出现（图四）窗口
- 4、在（图四）中，点击“添加”出现（图五）窗口



图三

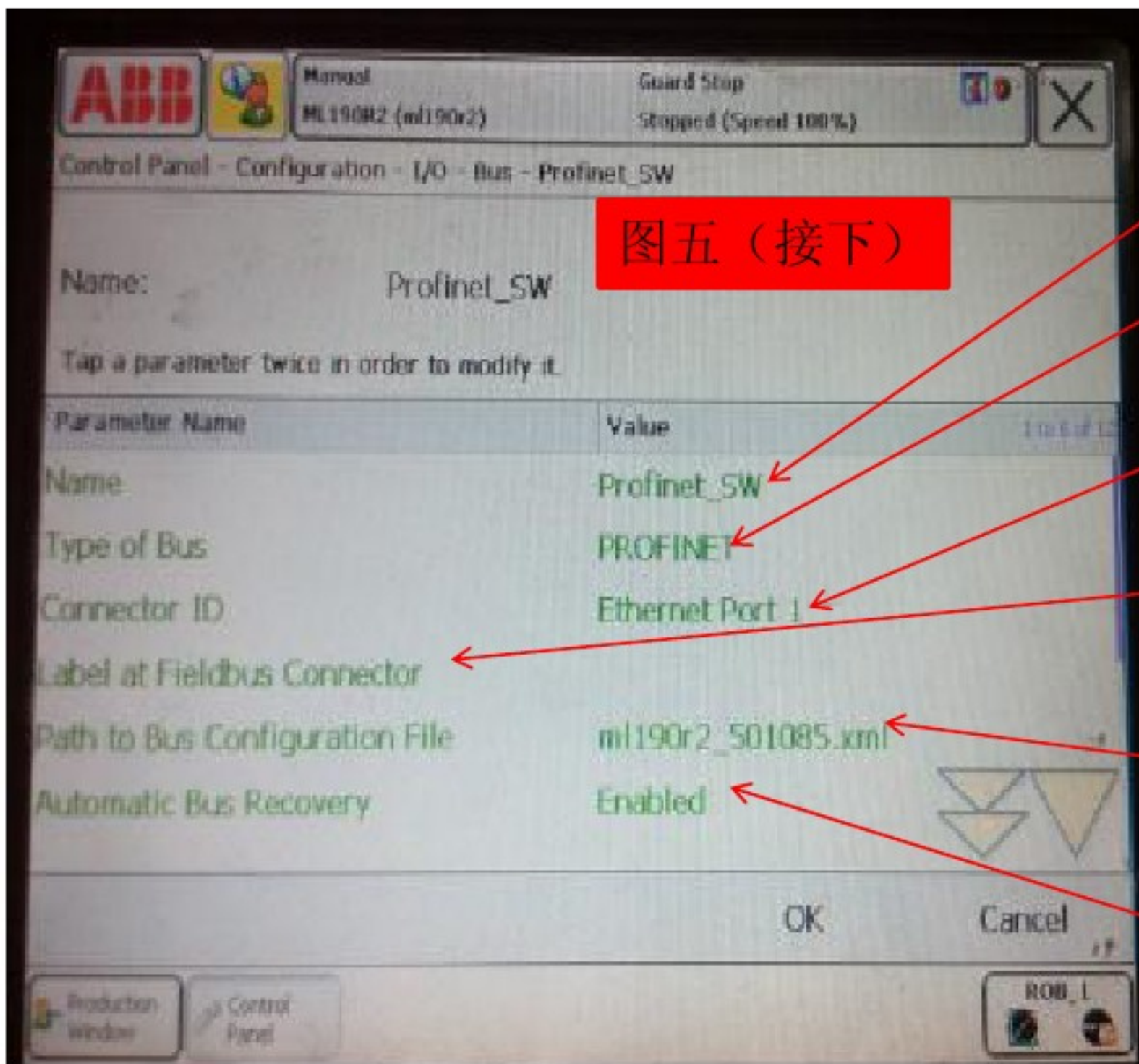


图四

八、定义I/O总线

5、在图五中，按实际需求填写或选择。

在系统内名称不允许重复，第一位必须为字母，由字母、数字、下划线组成，最长16位字符。



Name: 名称

Type of bus: 总线类型

Connector ID: 总线实际安装位置

Label at fieldbus connector: 标签在现场总线连接器

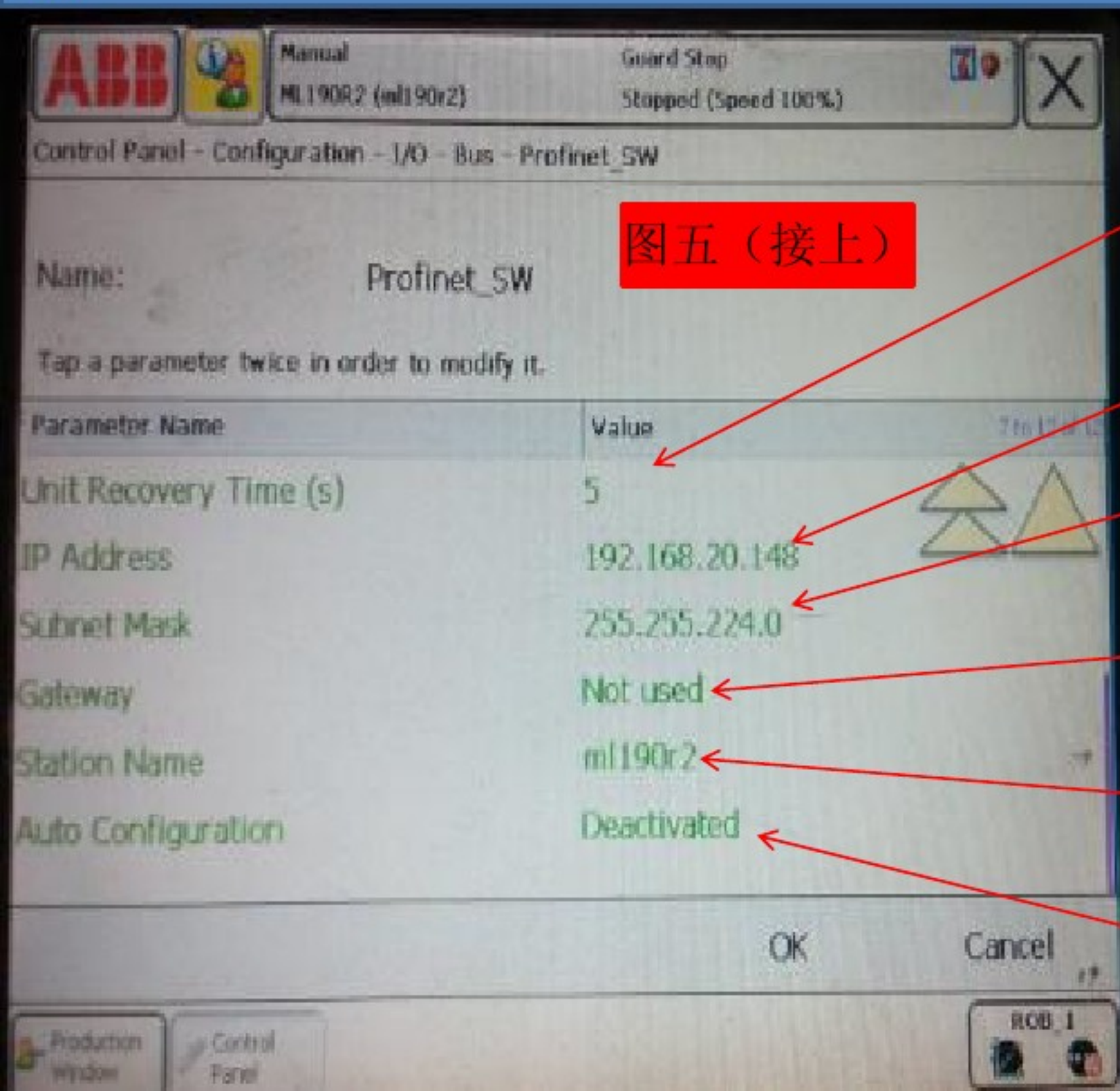
Path to bus configuration file: 总线配置文件路径 (GSD文件)

Automatic bus recovery: 总线自动恢复

八、定义I/O总线

5、在(图五)中，按实际需求填写或选择。

在系统内名称不允许重复，第一位必须为字母，由字母、数字、下划线组成，最长16位字符。



Unit recovery time: 单元恢复时间

IP Address: IP地址

Subnet mask: 子网掩码

Gateway: 网关

Station Name: 站名称

Auto Configuration: 自动配置

八、定义I/O总线

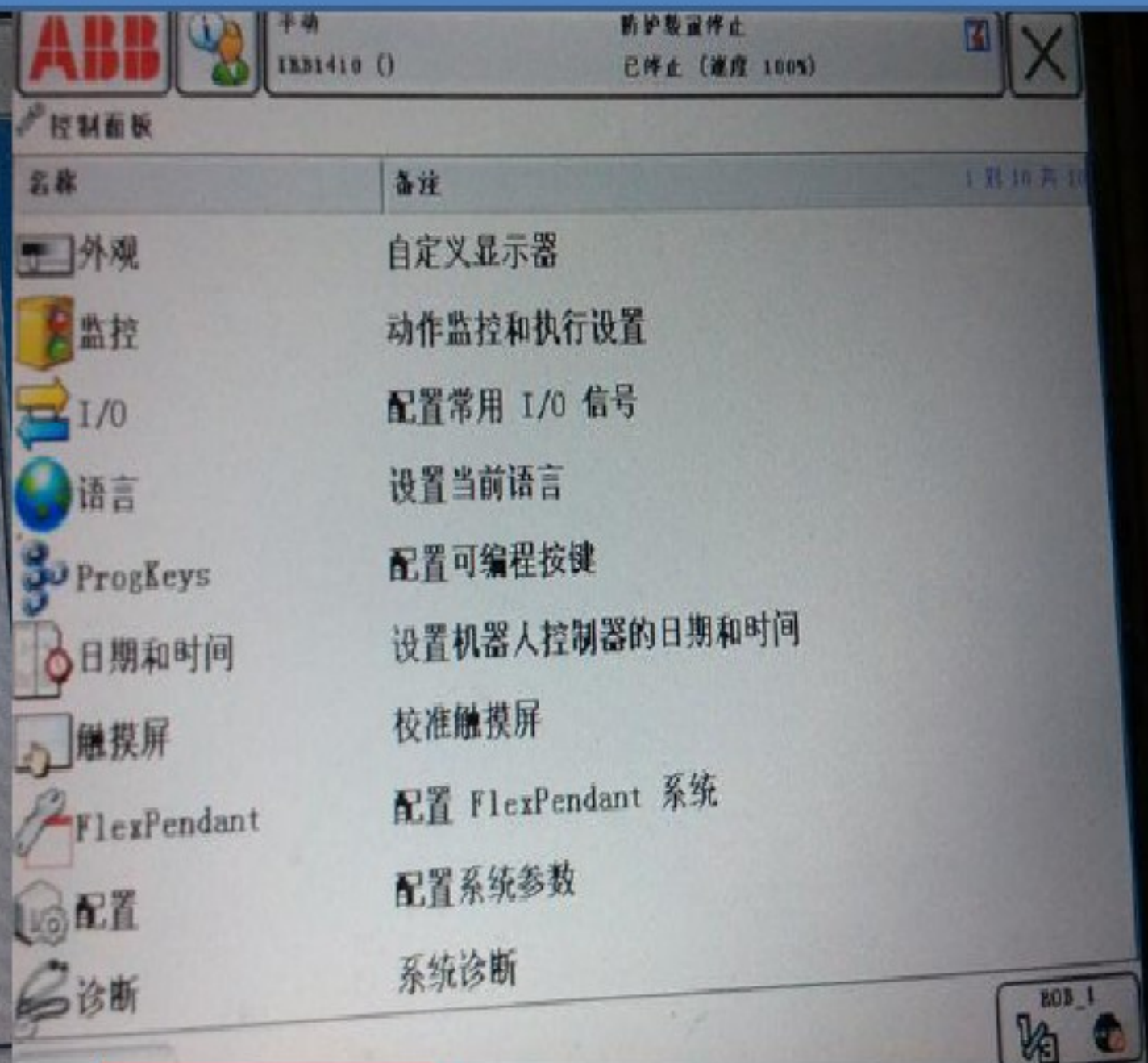
总线定义完毕

九、定义I/O单元

- 1、点击“ABB”出现（图一）窗口，在窗口里点击“控制面板”出现（图二）窗口
- 2、在（图二）窗口中，点击配置出现（图三）窗口



图一



图二

九、定义I/O单元

- 3、在（图三）中，双击“Unit”，出现（图四）窗口
- 4、在（图四）中，点击“添加”出现（图五）窗口



图三

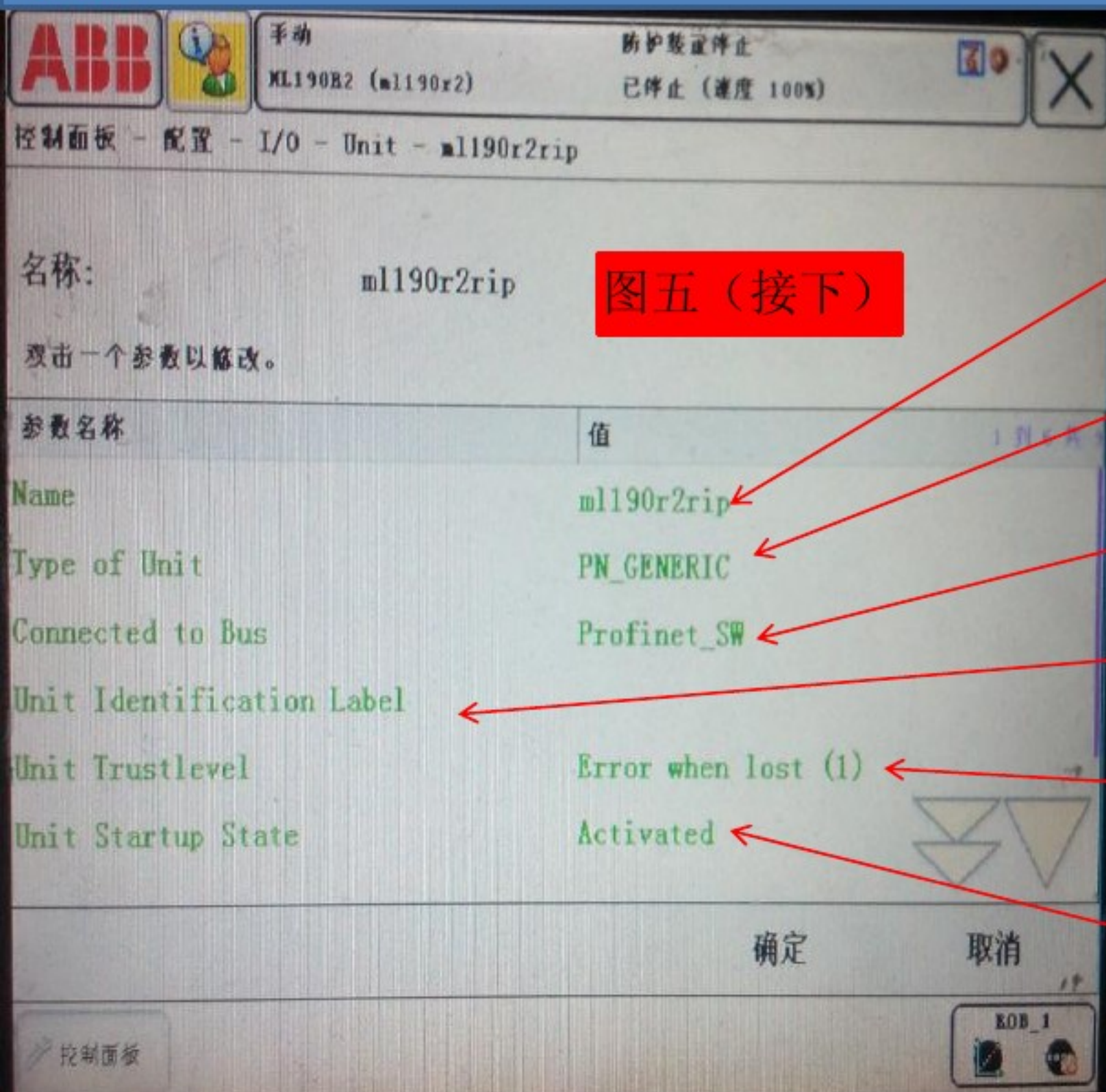


图四

九、定义I/O单元

5、在图五中，按实际需求填写或选择。

在系统内名称不允许重复，第一位必须为字母，由字母、数字、下划线组成，最长16位字符。



Name: 名称

Type of Unit: 单元类型

Connected to bus: 连接总线

Unit Identification label: 单元识别标签

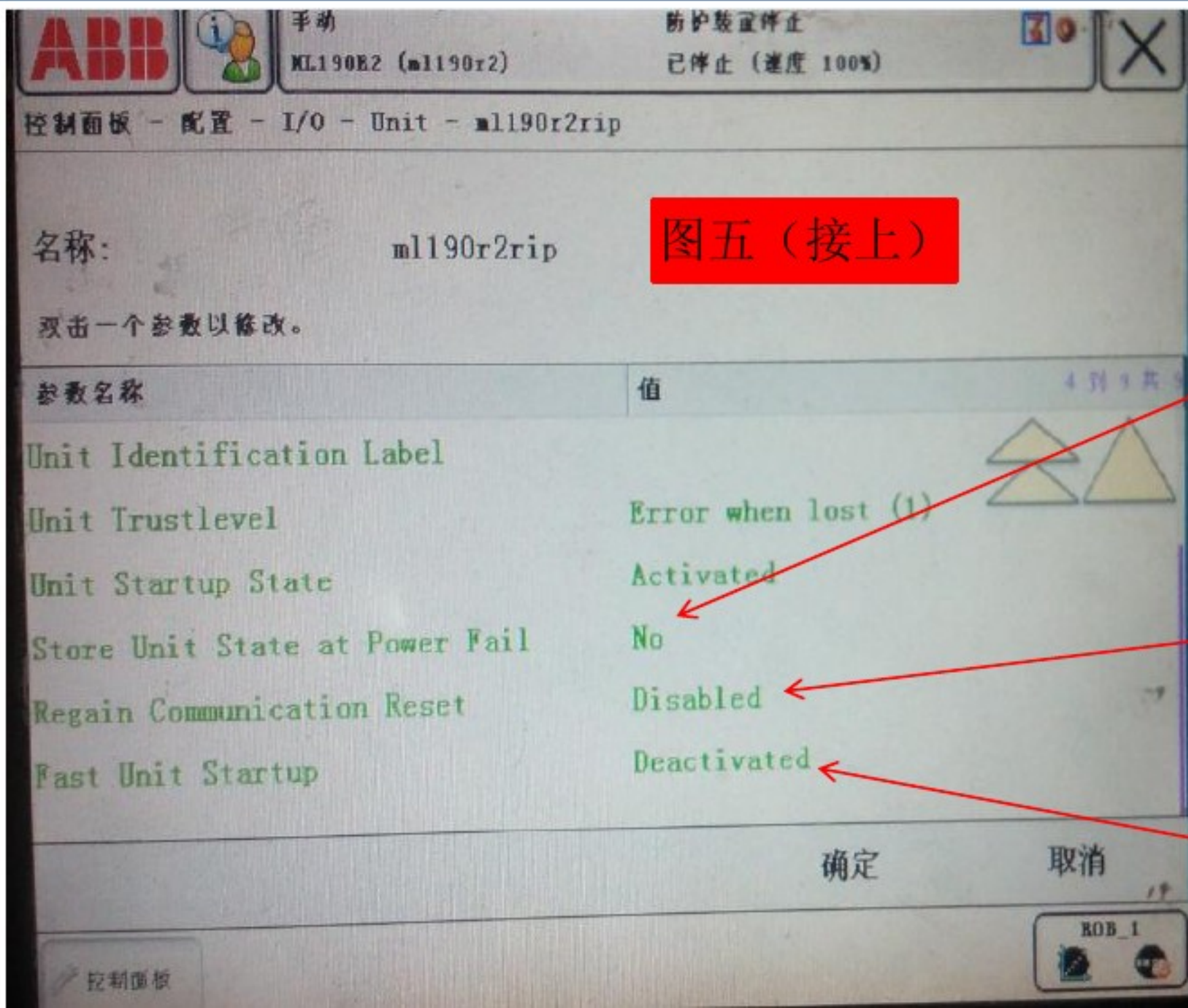
Unit Trust level: 单元信任标准

Unit Startup state: 单元启动状态

九、定义I/O单元

5、在图五中，按实际需求填写或选择。

在系统内名称不允许重复，第一位必须为字母，由字母、数字、下划线组成，最长16位字符。



Store Unit State at power fail: 存储单元停电状态

Regain communication reset: 重置恢复通信

Fast unit startup: 单元快速启动

九、定义I/O单元

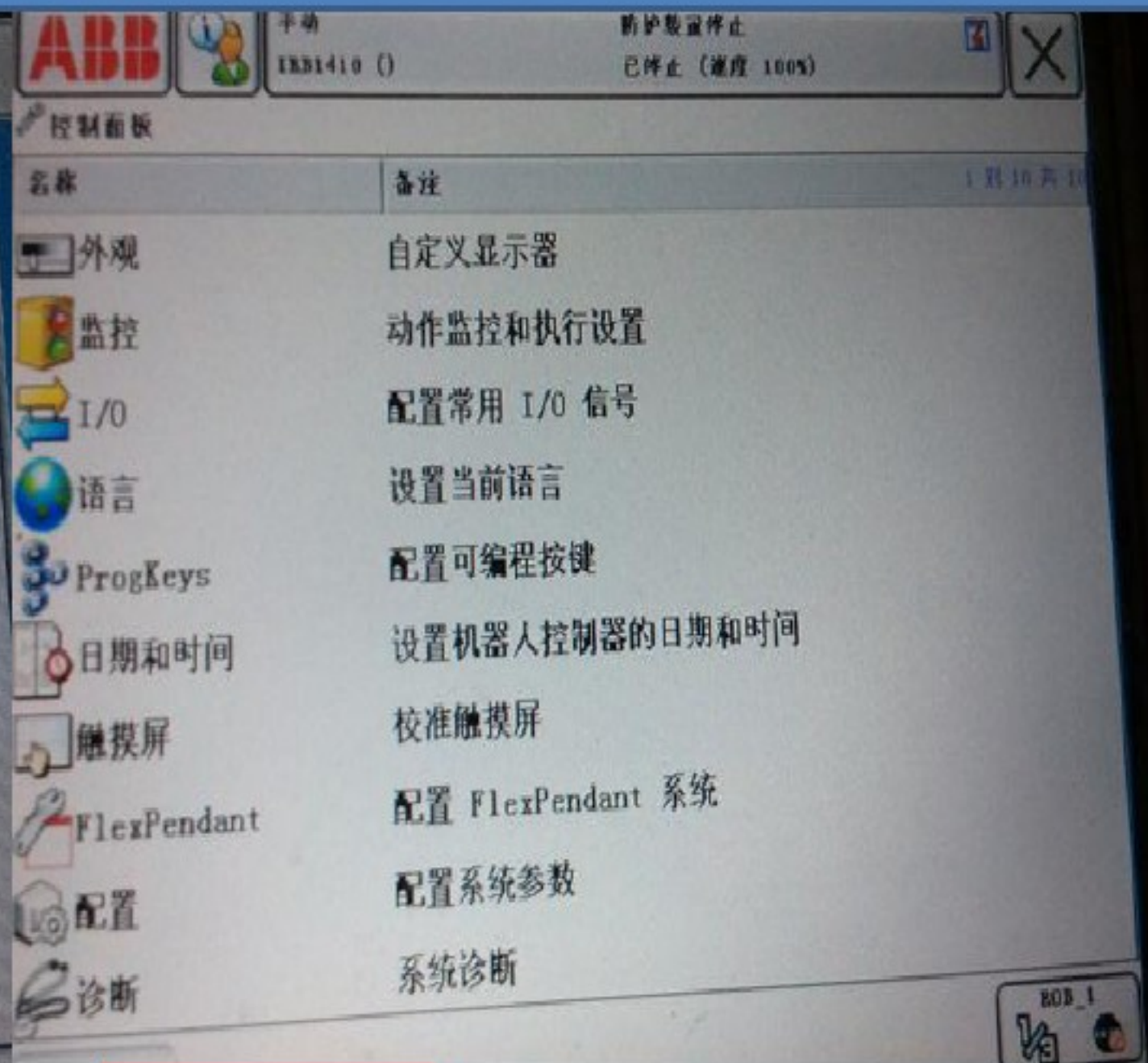
unit定义完毕

十、定义I/O信号

- 1、点击“ABB”出现（图一）窗口，在窗口里点击“控制面板”出现（图二）窗口
- 2、在（图二）窗口中，点击配置出现（图三）窗口



图一



图二

十、定义I/O信号

- 3、在（图三）中，双击“signal”，出现（图四）窗口
- 4、在（图四）中，点击“添加”出现（图五）窗口



图三



图四

十、定义I/O信号

5、在图五中，按实际需求填写或选择。

在系统内名称不允许重复，第一位必须为字母，由字母、数字、下划线组成，最长**16**位字符。

ABB 手动 防护装置停止 已停止 (速度 100%)

控制面板 - 配置 - I/O - Signal - gl_timer_ready

名称: gl_timer_ready

双击一个参数以修改。

参数名称	值
Name	gl_timer_ready
Type of Signal	Digital Input
Assigned to Unit	ml190r2timer
Signal Identification Label	
Unit Mapping	11
Category	

确定 取消

图五（接下）

Name: 名称

Type of signal: 信号类型

Assigned to unit: 指定到单元

Signal Identification label: 信号识别标签

Unit mapping: 单元映射端口

Category: 分类

十、定义I/O信号

5、在图五中，按实际需求填写或选择。

在系统内名称不允许重复，第一位必须为字母，由字母、数字、下划线组成，最长16位字符。



Access level:使用标准

Default value: 默认值

Filter time passive (ms): 过滤被动时间

Filter time active (ms): 过滤有效时间

Invert physical value: 取反物理值

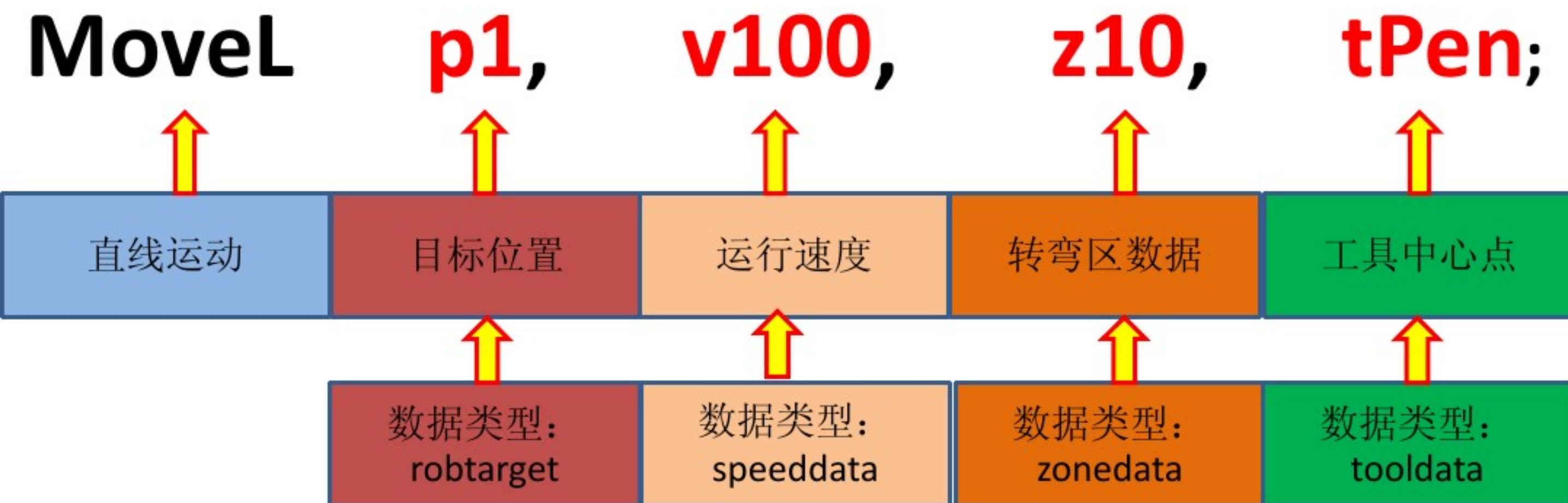
十、定义I/O信号

signal定义完毕

十一、基本指令

运动指令-----MoveL 直线运动

应用：机器人以线性移动方式运动至目标点，当前点与目标点两点确定一条直线，机器人运动状态可控，运动路径保持唯一，可能出现死点，常用于机器人在工作状态移动。

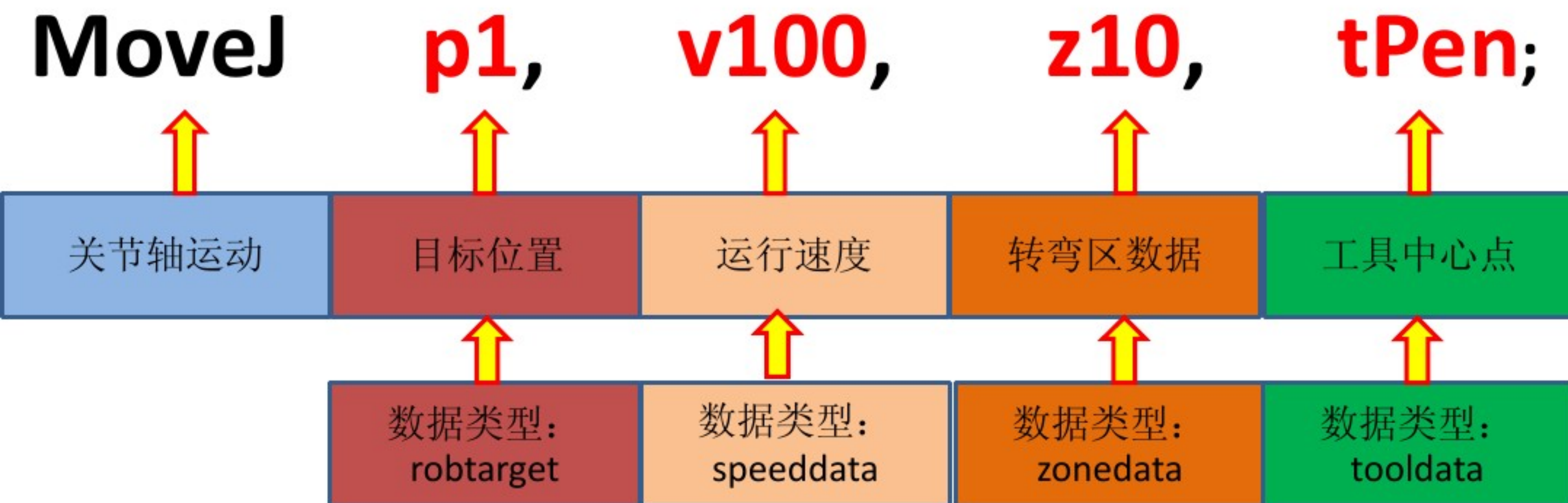


在添加或修改机器人的运动指令之前，一定要确认所使用的工具坐标和工件坐标

十一、基本指令

运动指令-----MoveJ 关节轴运动

应用：机器人以最快捷的方式运动至目标点，机器人运动状态不完全可控，但运动路径保持唯一，常用于机器人在空间大范围移动。



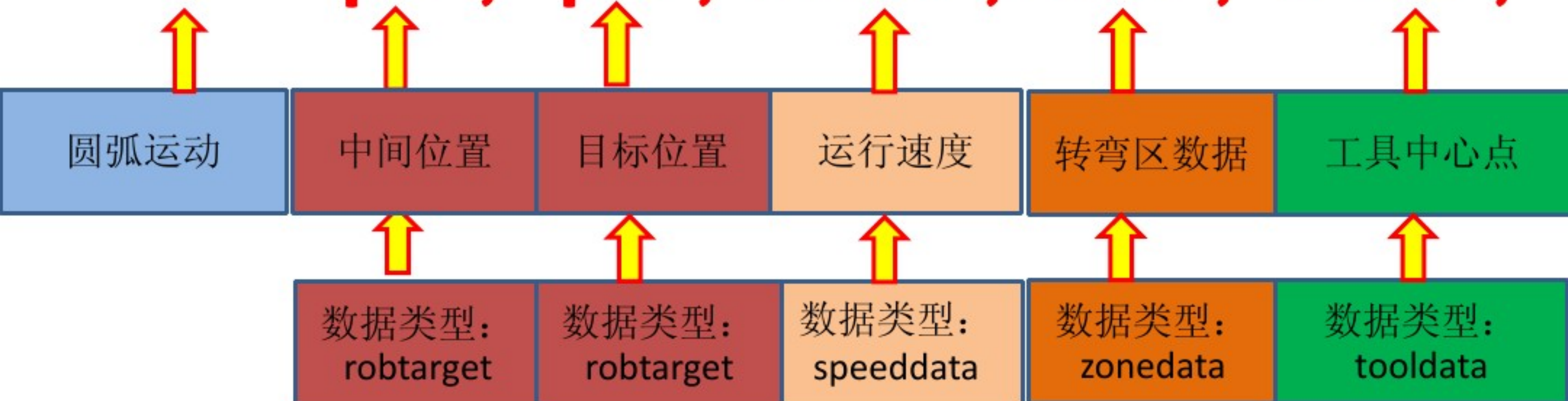
在添加或修改机器人的运动指令之前，一定要确认所使用的工具坐标和工件坐标

十一、基本指令

运动指令-----MoveC 圆弧运动

应用：机器人通过中间点以圆弧移动方式运动至目标点，当前点、中间点与目标点三点决定一段圆弧，机器人运动状态可控，运动路径保持唯一，常用于机器人在工作状态移动。

MoveC p1 , p2 , v100 , z10 , tPen ;



在添加或修改机器人的运动指令之前，一定要确认所使用的工具坐标和工件坐标

十一、基本指令

运动指令—MoveAbsJ 绝对位置运动

应用：机器人以单轴运行的方式运动至目标点，绝对不存在死点，运动状态完全不可控，避免在正常生产中使用此指令，常用于检查机器人零点位置，指令中 TCP 与 Wobj 只与运行速度有关，与运动位置无关。常用于机器人六个轴回到机械零点的位置。

MoveAbsJ

p1,

v100,

z10,

tPen;

绝对位置运动

目标位置

运行速度

转弯区数据

工具中心点

数据类型：
jointtarget

数据类型：
speeddata

数据类型：
zonedata

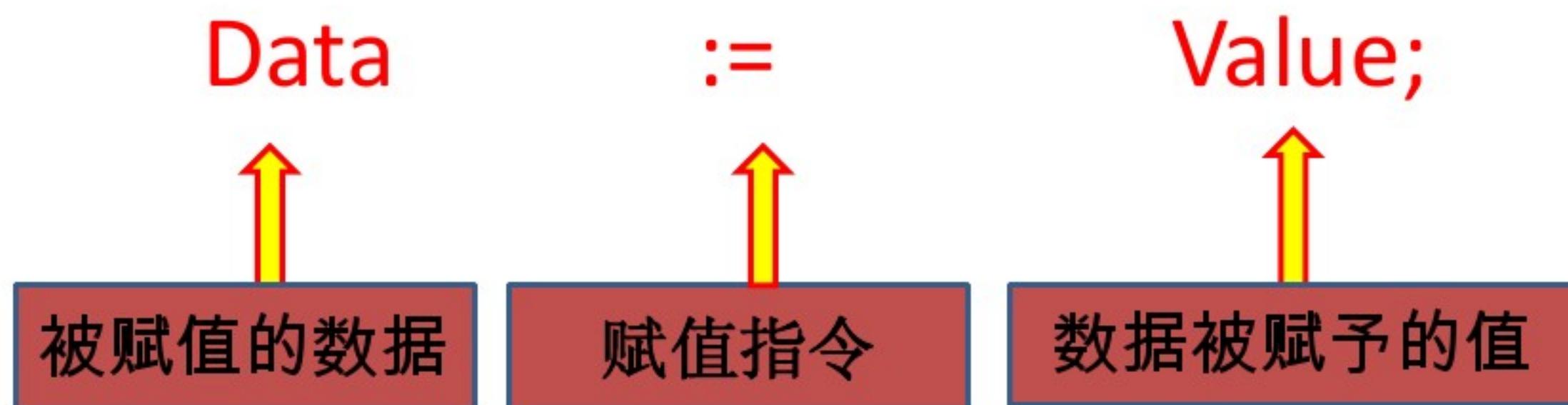
数据类型：
tooldata

在添加或修改机器人的运动指令之前，一定要确认所使用的工具坐标和工件坐标

十一、基本指令

数学运算指令 — $:=$ 赋值

赋值可以是一个常量或数学表达式



在添加或修改机器人的运动指令之前，一定要确认所使用的工具坐标和工件坐标

十一、基本指令

流程控制指令— IF

Type A IF <exp> THEN 符合 <exp> 条件
 “Yes-part” 执行 “Yes-part” 指令
 ENDIF

Type B IF <exp> THEN 符合 <exp> 条件
 “Yes-part” 执行 “Yes-part” 指令
 ELSE 不符合 <exp> 条件
 “Not-part” 执行 “Not-part” 指令
 ENDIF

Type C IF <exp1> THEN 符合 <exp1> 条件
 “Yes-part1” 执行 “Yes-part1” 指令
 ELSEIF <exp2> THEN 符合 <exp2> 条件
 “Yes-part2” 执行 “Yes-part2” 指令
 ELSE 不符合 <exp1> 与 <exp2> 条件
 “Not-part” 执行 “Not-part” 指令

ENDIF

十一、基本指令

流程控制指令— TEST 测试

TEST **reg1**

CASE **1:**

PATH 1;

CASE **2:**

PATH 2;

.....

DEFAULT:

Error;

ENDTEST

测试（数字）变量，

数字变量值为 1，

执行 CASE 1 指令。

数字变量值为 2，

执行 CASE 2 指令。

数字变量值无法在

CASE 内找到相应值

执行DEFAULT内指令。

十一、基本指令

流程控制指令— WHILE 循环

```
reg1:=0;
```

```
WHILE reg1< 5 DO
```

```
    Square;
```

```
    reg1:=reg1+1;
```

```
ENDWHILE
```

循环至不符合判断条件 $\text{reg1} < 5$ ，才执行
ENDWHILE 以后的
指令。

- ❖ 循环指令 WHILE 运行时，机器人循环至不满足判断条件后，才跳出循环指令，执行 ENDWHILE 以后的运行指令。当循环指令 WHILE 运行时，存在死循环，在编写相应机器人程序时必须注意。

十一、基本指令

输入输出指令—Set 置位

do1: 输出信号名。 将数字输出信号置位为“ 1”。

输入输出指令—Reset 复位

do1: 输出信号名。 将数字输出信号复位为“ 0”

输入输出指令—PulseDO 脉冲

do1: 输出信号名。 *(signaldo)*

输出一个脉冲信号，脉冲长度为 0.2 s。

[\PLength] - 参变量 *(num)*

脉冲长度，0.1s–32s。

十一、基本指令

输入输出指令—WaitDI 等待输入

di1: 输入信号名。 (*signal di*)

1: 状态。 (*dionum*)

等待一个输入信号达到规定状态。

参变量：

[*\MaxTime*] (*num*)

等待输入信号最长时间 s。

[*\TimeFlag*] (*bool*)

逻辑量，TRUE 或 FALSE。

- 如果只选用参变量 [*\MaxTime*]，机器人等待超过最长时间后，机器人将停止运行，并显示相应出错信息或进入机器人错误处理程序（Error Handler）。
- 如果同时选用参变量 [*\MaxTime*] 与参变量 [*\TimeFlag*]，等待超过最长时间后，无论是否满足等待的状态，机器人将自动执行下一句指令。如果在最长等待时间内得到相应信号，将逻辑量置为 FALSE，如果超过最长等待时间，将逻辑量置为 TRUE。

十一、基本指令

计时指令 — ClkReset 时钟复位

clock1: 机器人时钟名称。 (clock)

将一个机器人时钟复位。

计时指令 — ClkStart 开始计时

clock1: 机器人时钟名称。 (clock)

将一个机器人时钟打开，开始计时。

计时指令 — ClkStop 计时停止

clock1: 机器人时钟名称。 (clock)

将一个机器人时钟关闭，停止计时，但仍旧保持时钟数据直至复位。

十一、基本指令

通信指令—TPErase

示教器显示屏清屏指令

通信指令—TPWrite 显屏

string: 显示屏显示的字符串。 (string)

在示教器显示屏上显示字符串数据，也可以用“XXXXXX”形式直接定义字符串，每一个写屏指令最多显示 80 个字符。

设置指令—WaitTime 等待时间

机器人等待时间 s。 (num)

等待指令只是让机器人程序运行停顿相应时间。

十一、基本指令

设置指令—VelSet 设置速度%

机器人运行速率 %。 (num)

机器人最大速度 mm/s。 (num)

每个机器人运动指令均有一个运行速度，在执行运动速度控制指令 VelSet 后，机器人实际运行速度为运动指令规定运行速度乘以机器人运行速率，并且不超过机器人最大运行速度

设置指令—AccSet 设置坡度%

机器人加速度百分率 %。 (num)

机器人加速度坡度 %。 (num)

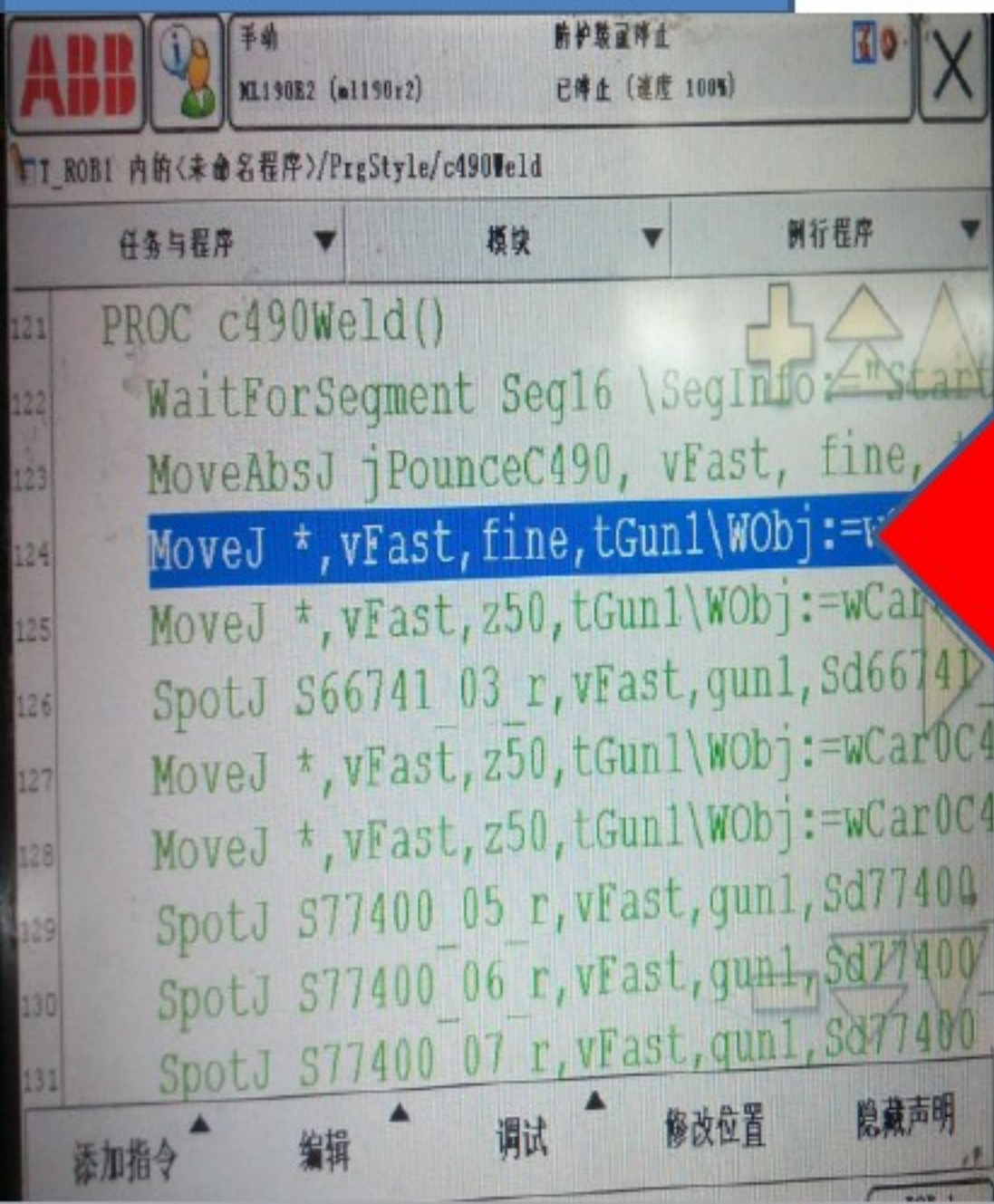
十一、基本指令

基本指令完毕

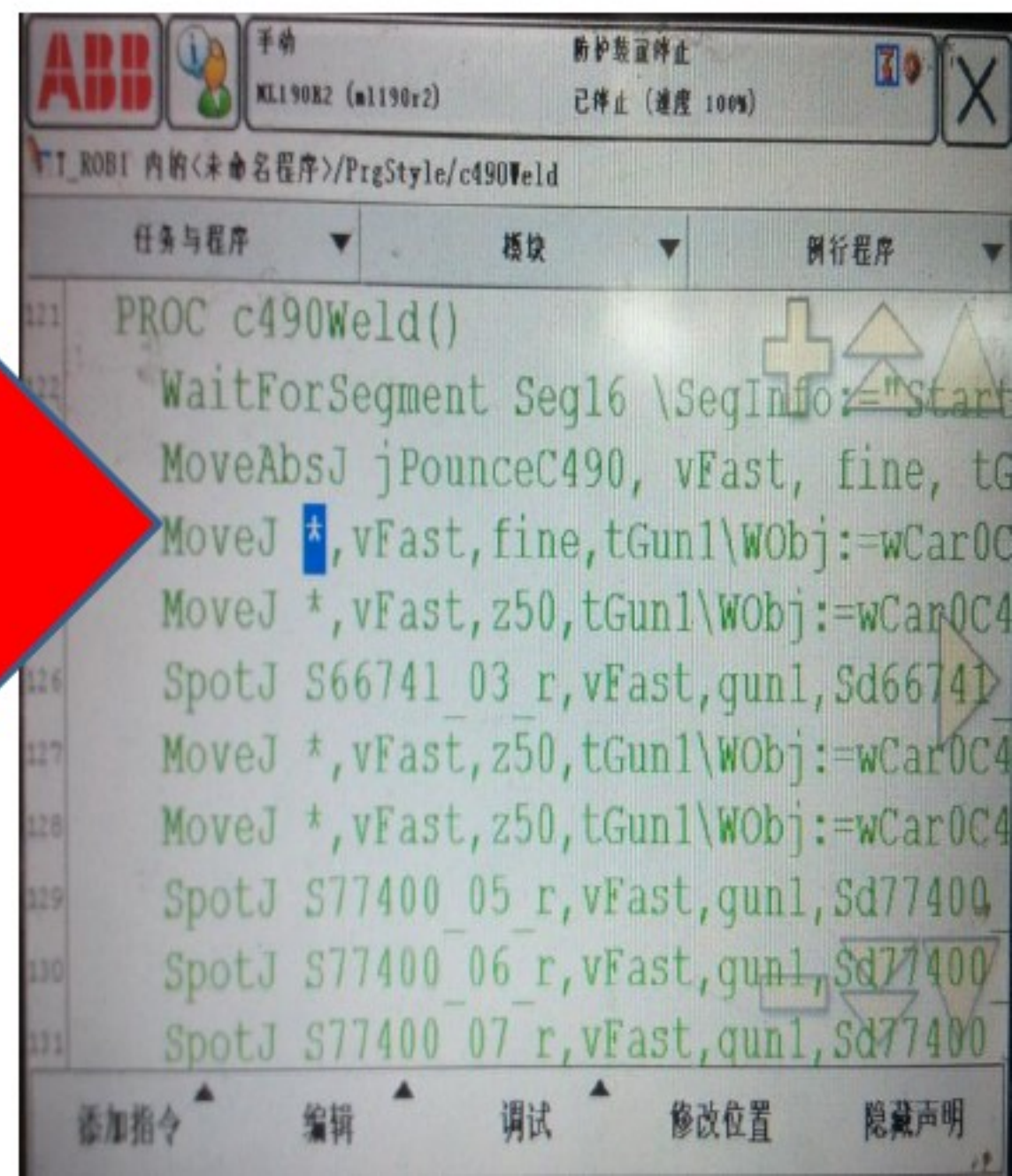
十二、off()功能

- ▶ `Offs(p1, 100, 50, 0)` 代表一个距离 `p1` 点 X 轴偏差量为 100mm, Y 轴偏差量为 50mm, Z 轴偏差量为 0 的点。
- ▶ 函数 `Offs()` 的坐标方向与机器人 `Wobj` 坐标系一致。

使用步骤:

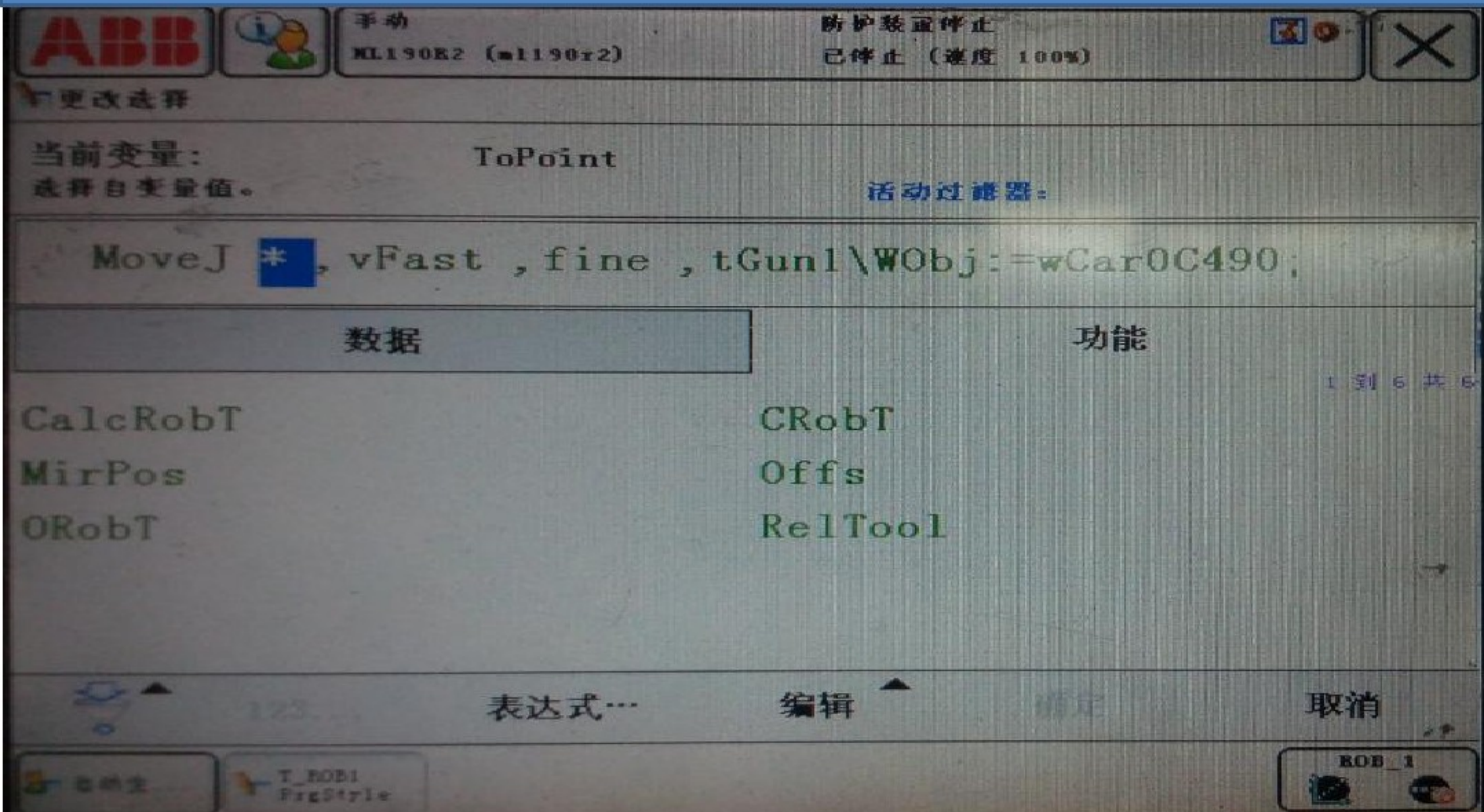


例行程序中选择一个点，并双击该点



十二、off()功能

在更改选择窗口中，单击 **功能** 标签。此时会自动显示与所选参数类型相同的全部功能。在列出的函数中选择函数 **offs()**



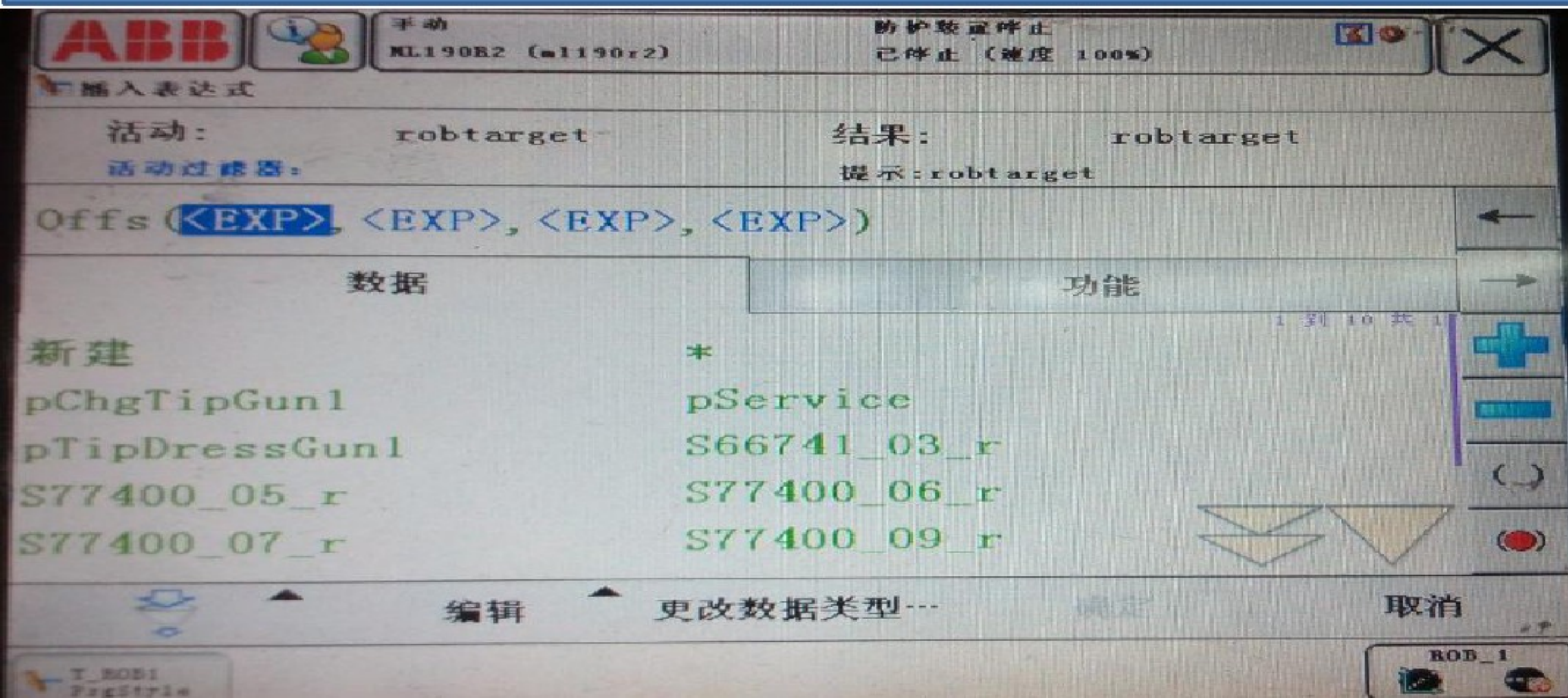
十二、offs()功能

点击offs()的第一个参数部分，选择参考点

点击offs()的第二个参数部分，单击 编辑 / 仅限选定内容，输入基于参考点在X方向的偏移

点击offs()的第三个参数部分，单击 编辑 / 仅限选定内容，输入基于参考点在Y方向的偏移

点击offs()的第四个参数部分，单击 编辑 / 仅限选定内容，输入基于参考点在Z方向的偏移

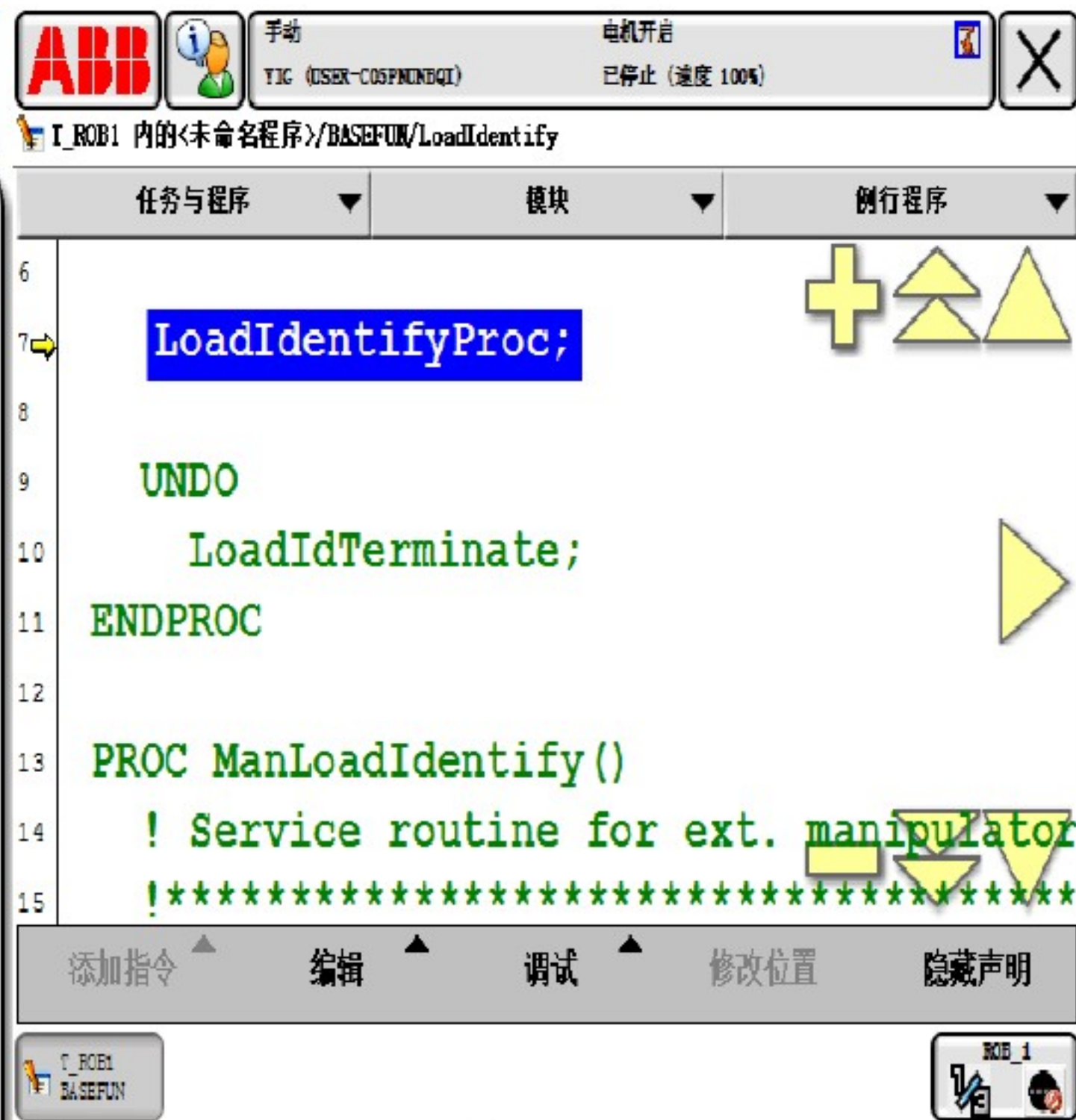


十二、off()功能

off()功能完成

十三、程序的编辑

1、点击“程序编辑器”



2、点击“点击任务与程序”

十三、程序的编辑

3、点击“文件”，选择“新建程序”



手动
YIG (USER-COSPNUBQ1)

电机开启
已停止 (速度 100%)



程序编辑器

任务与程序

任务名称	程序名称	类型	1到1共1
T_ROB1	<No Name>	Normal	

新建程序...

加载程序...

另存程序为...

重命名程序...

删除程序...

文件 显示模块 打开

T_ROB1 BASEFUN

ROB_1



手动
YIG (USER-COSPNUBQ1)

电机开启
已停止 (速度 100%)



NewProgramName - T_ROB1/MainModule/main

任务与程序 模块 例行程序

2 PROC main()
3 <SMT>
4 ENDPROC
5 ENDMODULE

添加指令 编辑 调试 修改位置 隐藏声明

T_ROB1 MainMo... ROB_1

4、点击“模块”

十三、程序的编辑

5、点击“文件”，选择“新建模块”

ABB

手动

电机开启

YIG (USER-C05PNUNBQ1)

已停止 (速度 100%)

I_ROB1

模块

名称	类型	更改
AppConfig	系统模块	X
BASE	系统模块	X
DPBASE	系统模块	
DPUSER	系统模块	X
新建模块...	系统模块	X
加载模块...	系统模块	X
另存模块为...	程序模块	
更改声明...	系统模块	X
删除模块...	系统模块	X

文件 刷新 显示模块 后退

ABB

手动

电机开启

YIG (USER-C05PNUNBQ1)

已停止 (速度 100%)

I_ROB1

模块

名称

AppConfig

BASE

DPBASE

DPUSER

GlueApp

MainModul

PLCApp

PLCDialog

文件 刷新 显示模块 后退

添加新的模块后，您将丢失程序指针。
是否继续？

是 否

6、点击“是”

十三、程序的编辑

7、在名称处给新建模块命名，在类型选择程序类型，点击“确定”



手动
YIG (USER-COSPNUBQ1)
电机开启
已停止 (速度 100%)

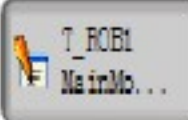


新模块 - NewProgramName - I_ROB1

新模块

名称:

类型:





手动
YIG (USER-COSPNUBQ1)
电机开启
已停止 (速度 100%)

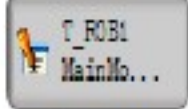


T_ROB1

模块

名称 ▲	类型	更改	5 到 12 共 12
GlueApp	系统模块	X	 
MainModule	程序模块		
Module1	程序模块		
PLCApp	系统模块	X	
PLCDialog	系统模块	X	
PLCError	系统模块	X	
PrgStyle	系统模块	X	
user	系统模块	X	

文件 ▲ 刷新 显示模块 后退



8、点击“Module1”

十三、程序的编辑

9、点击“例行程序”

ABB

手动

电机开启

YIG (USER-COSPUNBQI)

已停止 (速度 100%)

X

NewProgramName - I_ROB1/Module1

任务与程序

模块

例行程序

1

2

3

4

MODULE Module1

ENDMODULE

+

添加指令

编辑

调试

修改位置

隐藏声明

I_ROB1

Module1

ROB_1

ABB

手动

电机开启

YIG (USER-COSPUNBQI)

已停止 (速度 100%)

X

I_ROB1/Module1

例行程序

活动过滤器:

名称	模块	类型
无例行程序		

1 到 1 共 1

新建例行程序...

复制例行程序...

移动例行程序...

更改声明...

重命名...

删除例行程序...

文件

显示例行程序

后退

I_ROB1

Module1

ROB_1

10、点击“文件”，选择“新建例行程序”

十三、程序的编辑

11、在名称处点击“ABC....”，给新建的例行程序命名，再点击“确定”

ABB

手动

电机开启

YIG (USER-COSPUNBQI)

已停止 (速度 100%)

新例行程序 - NewProgramName - T_ROB1/Module1

例行程序声明

名称: Routine2

ABC...

类型: 程序

▼

参数: 无

...

数据类型: num

...

模块: Module1

▼

本地声明: ☐ 撤消处理程序: ☐

错误处理程序: ☐ 向后处理程序: ☐

结果...

确定

取消

T_ROB1
Module1

ROB_1

ABB

手动

电机开启

YIG (USER-COSPUNBQI)

已停止 (速度 100%)

T_ROB1/Module1

例行程序

活动过滤器:

名称 ▲	模块	类型	1 到 1 共 1
Routine2()	Module1	Procedure	

文件 ▲

显示例行程序

后退

T_ROB1
Module1

ROB_1

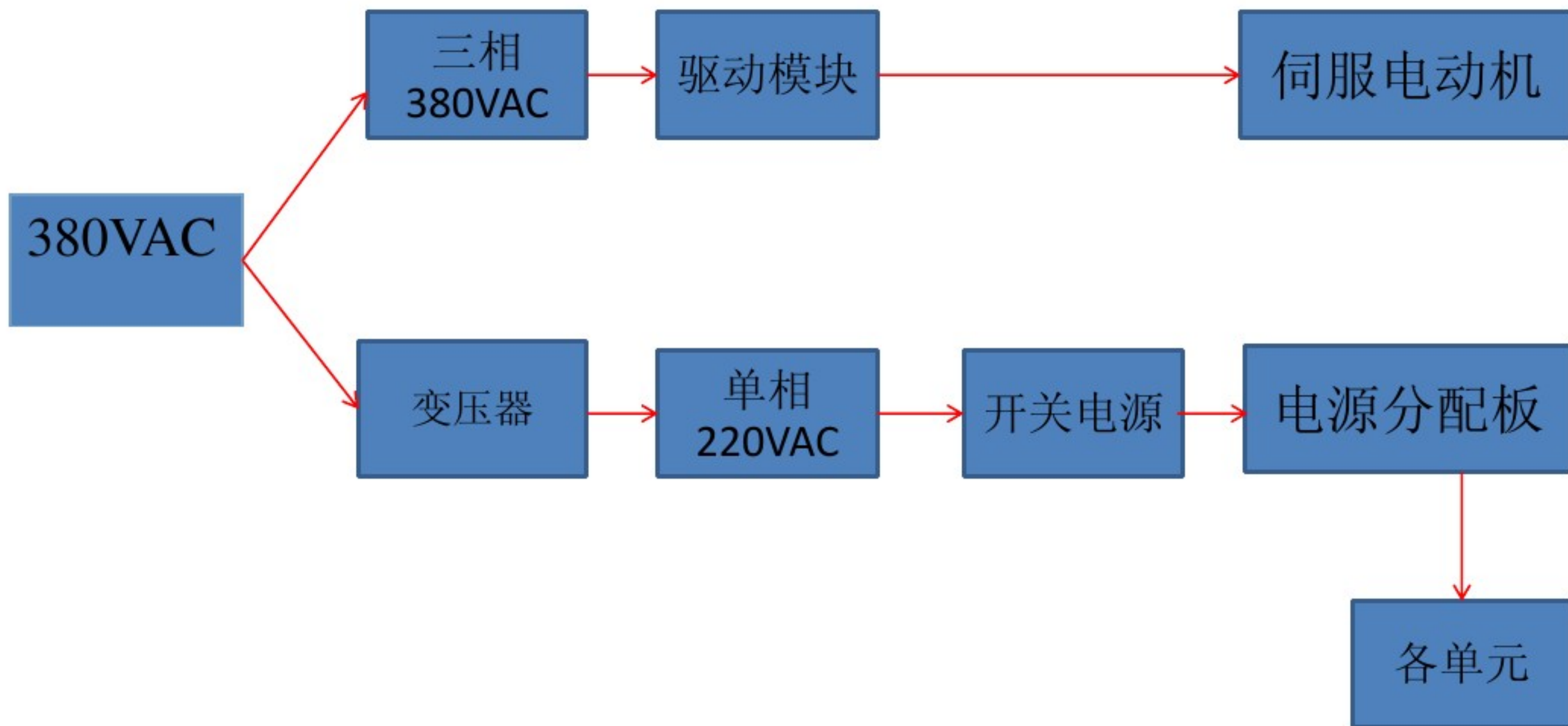
12、点击“routine2”

十三、程序的编辑

程序编辑完成

十四、电路图的分析

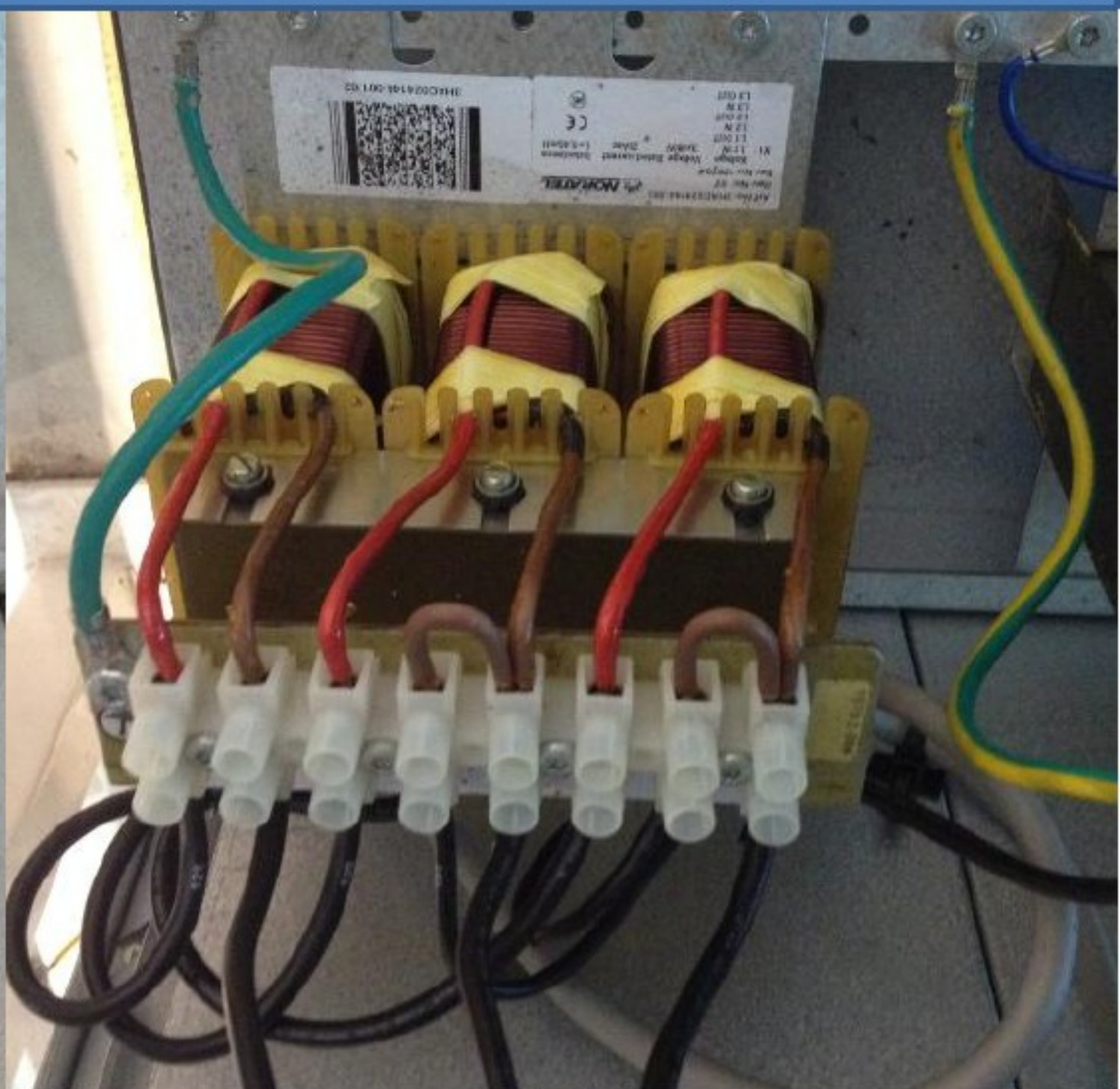
电源分配图



十四、电路图的分析

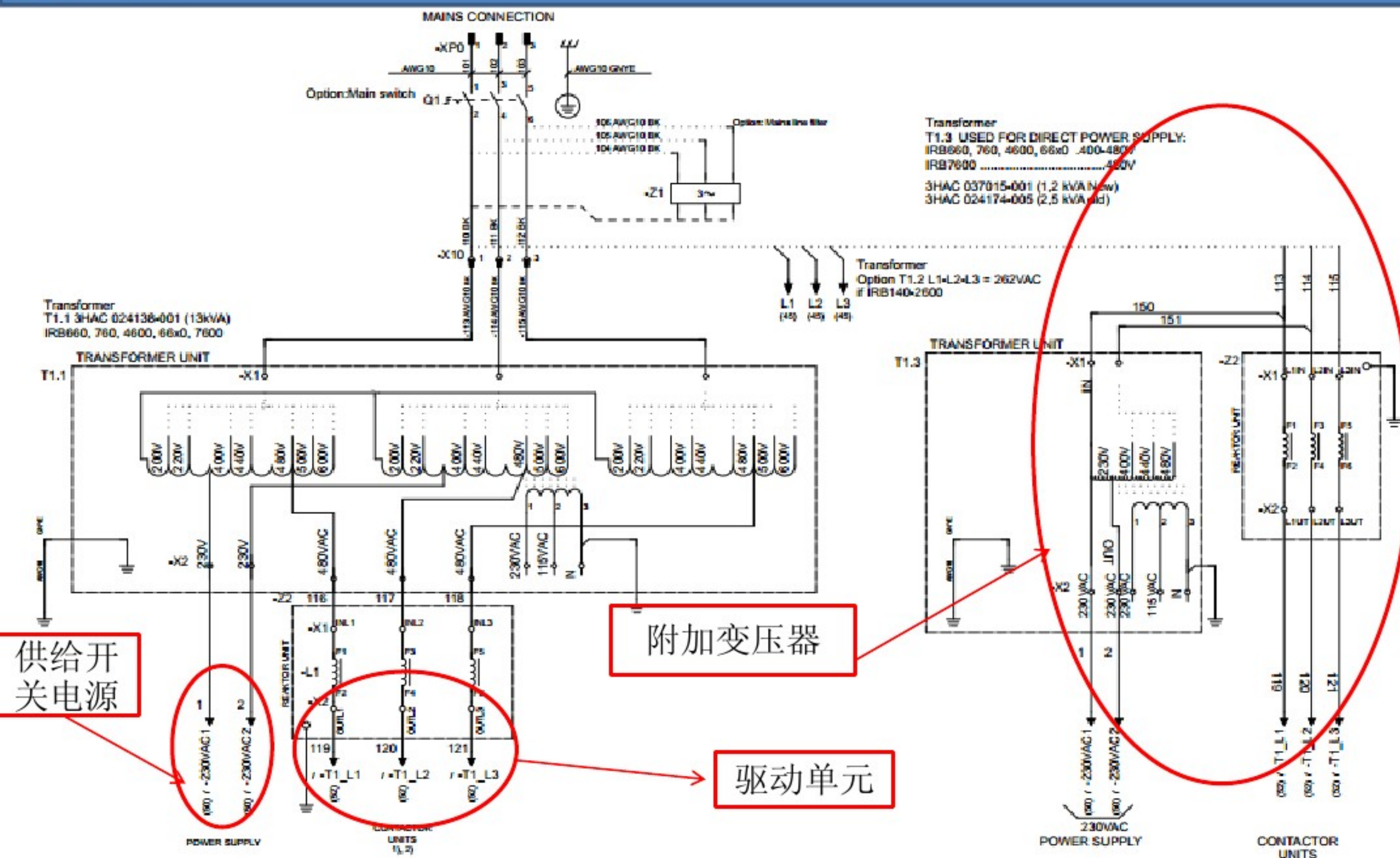
变压器：

- 1、根据每个国家用电标准不同，选择适合电源电压
- 2、自藕变压器可以防止外部电网干扰



十四、电路图的分析

变压器



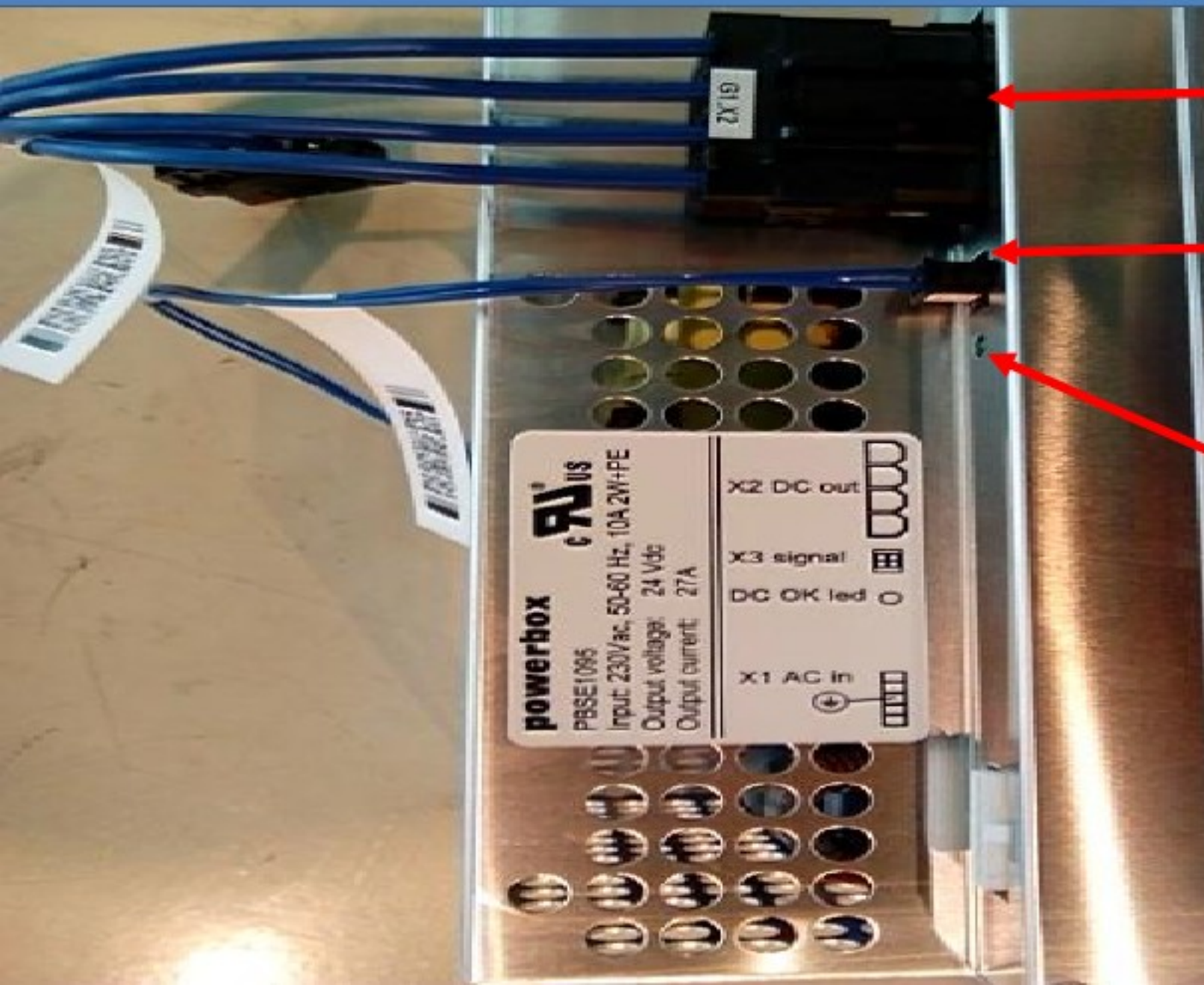
十四、电路图的分析

开关电源 (G1)由变压器输入220VAC,经过整流滤波后, 输出24VDC。

指示灯：绿色：表示开关电源输出正常

暗绿色：表示输出的电源低于用电器的最低电源电压。

指示灯不亮：表示开关电源不正常或损坏



DC24V

检测温度和AC ON—OK

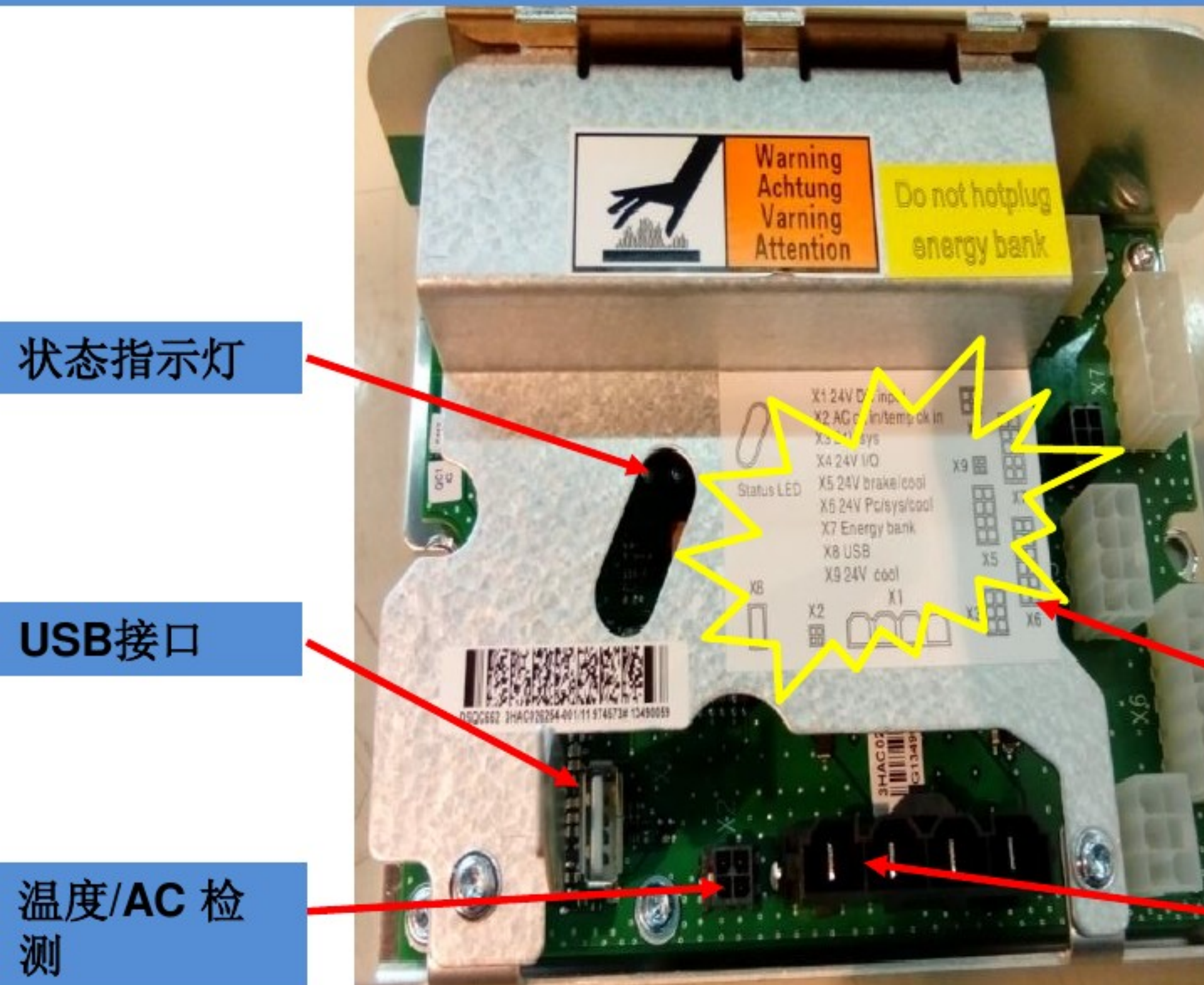
指示灯：绿色：表示开关电源输出正常

暗绿色：表示输出的电源低于用电器的最低电源电压。

指示灯不亮：表示开关电源不正常或损坏

十四、电路图的分析

分配器(G2)的电源直接由开关电源输入，分配到各单元。各级电源的容量是有限的，一旦超出额定容量后，单元发烫。因此，在电路中，连接了温度检测。



DCOK指示灯:

绿色: 在直流输出超出指定的最小电压时。
关: 在直流输出低于指定的最小电压时。

输出各部分24V

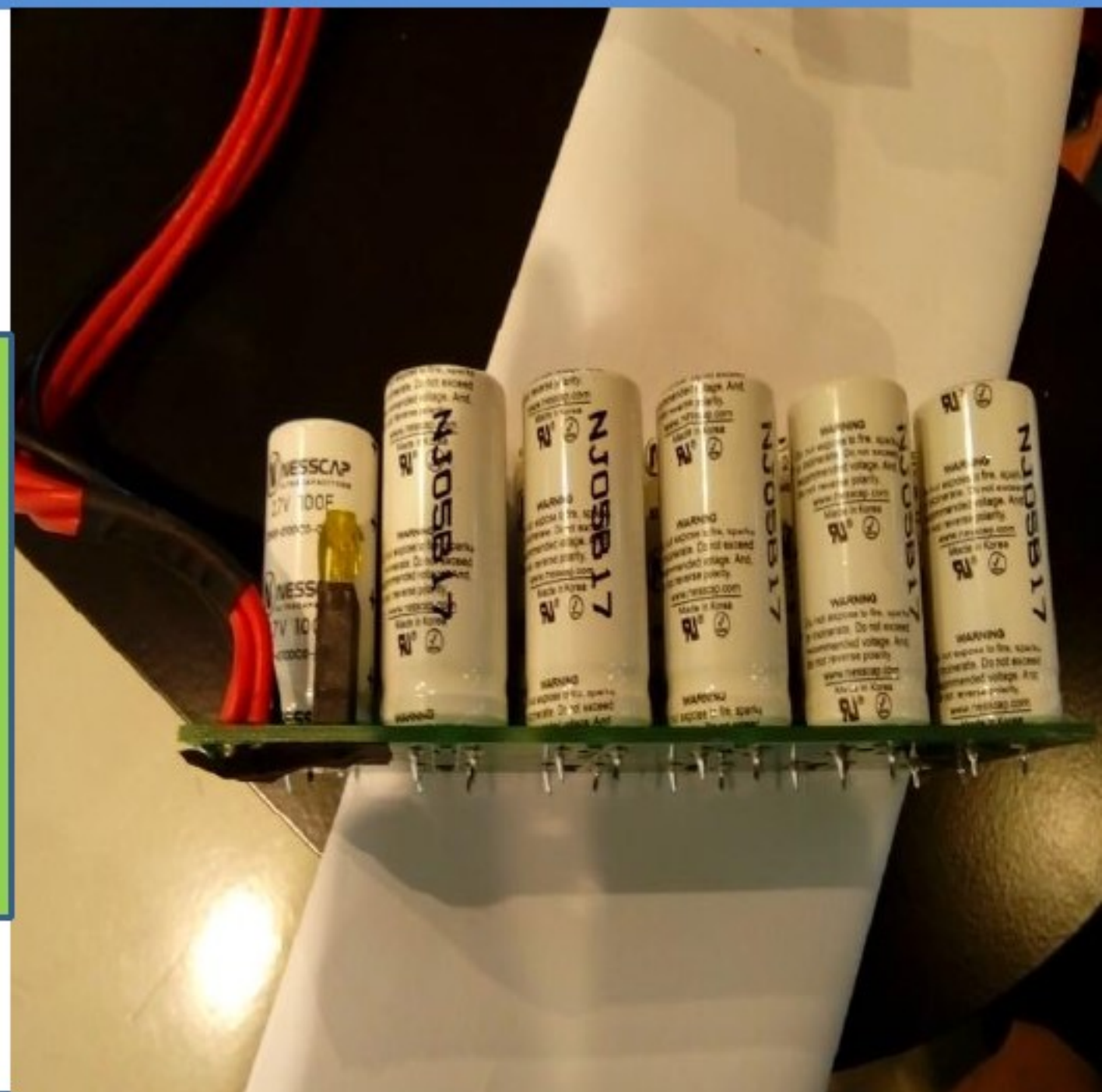
输入24V电源

十四、电路图的分析

电容 (G3)作用: 断电时放电2min,为主机保存数据提供电源
主机保存数据需要时间: 大约30s



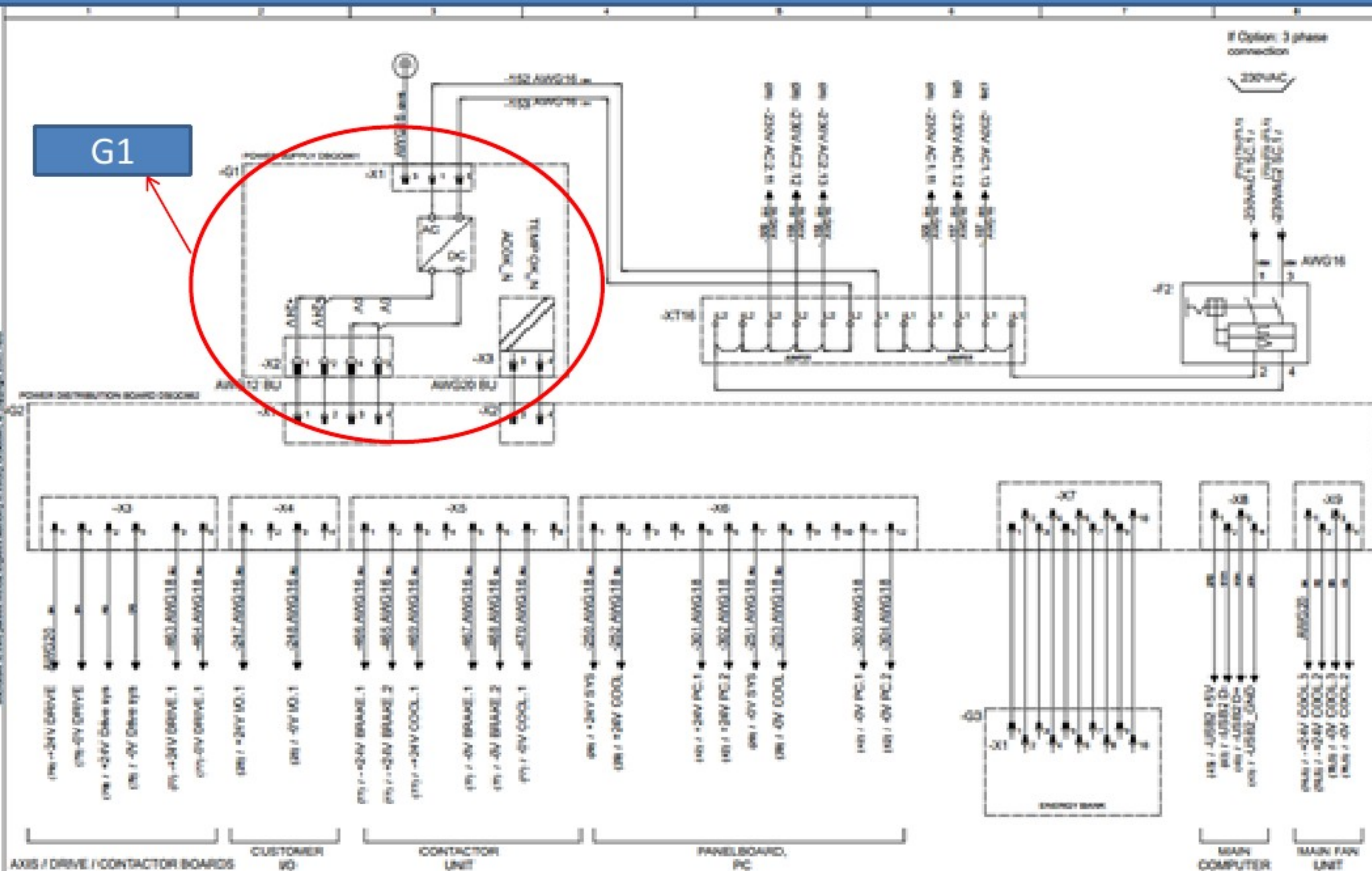
G3电容
由11个
2.7V
100F
小电容
组成



注意: 不允许直接关机后立即开机, 又读又写很容易造成死机或系统失败

十四、电路图的分析

电源电路图



十四、电路图的分析

安全板



安全板接口：

X1 X2

3、4、5脚：给急停提供24V电源，同时输出到运行链

12脚：给外部输出信号

9、10脚：连接打到安全板上ES1 ES2的信号灯，同时与X9相连接，

X9（11-18脚）为操作面板急停

7、8脚：可以给外部提供24V电源

X10为示教器急停，1、2脚为示教器电源，3、4脚为示教器使能，5、7脚为进。6、8脚为出

X8为与主机通讯

X9（1-8）脚为模式选择 9、10、19为操作面板复位按钮

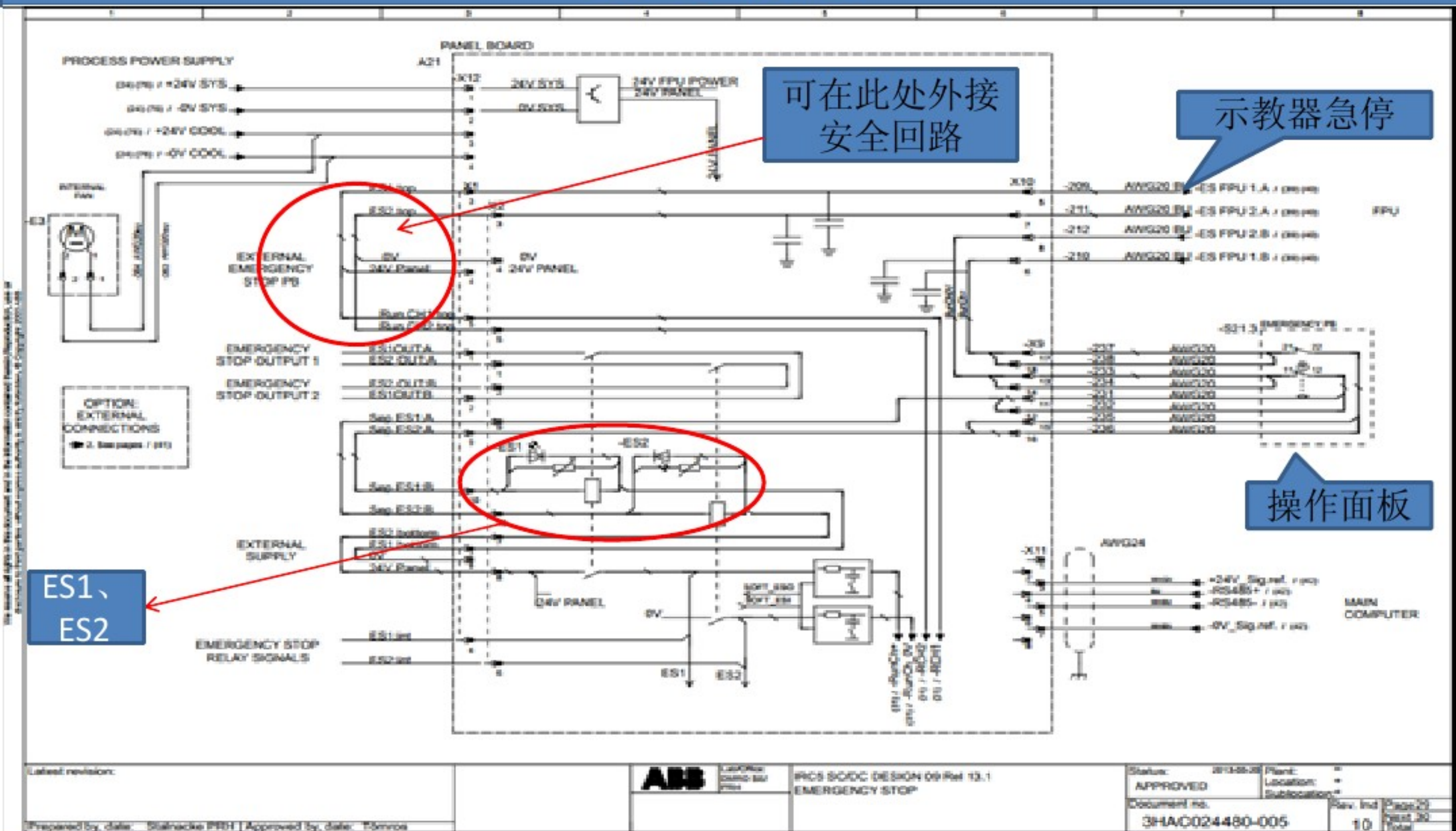
X15为模式切换

X18 接机柜上的风扇

X5为运行链关键，X5分四组。分别为1、2、3、4、5、6，7、8、9、10、11、12。若X5条件满足，接触器板才会吸合。

十四、电路图的分析

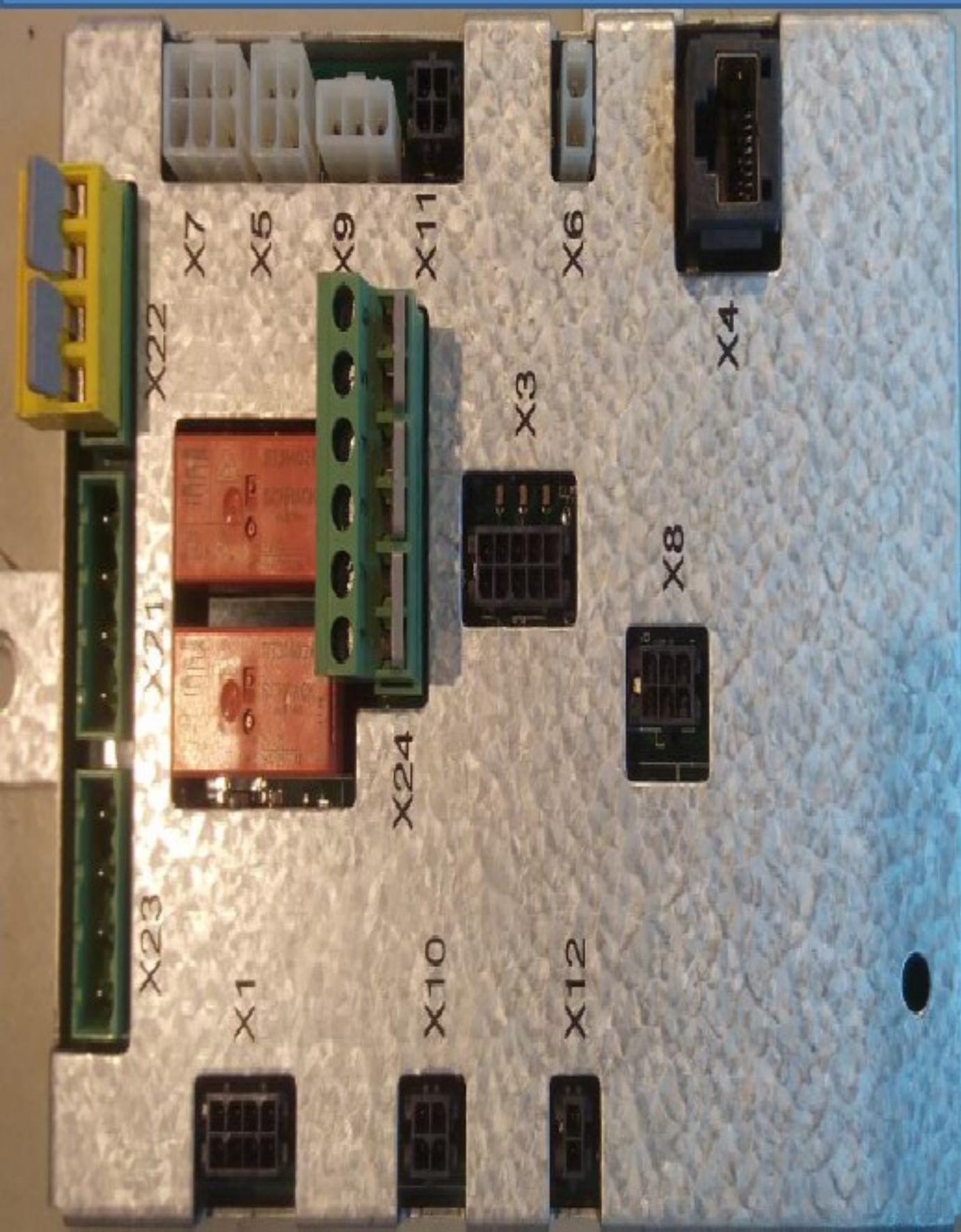
安全回路图



板内电源 → 示教器急停 → 主机操作面板急停 → 板内ES1、ES2线圈 → 线圈再连接外部ES200

十四、电路图的分析

接触器板



接触器板各接口：

XS6给板子提供电源

X5： 1、2脚马达温控电阻

3、4脚为刹车0V电源

X10、X11连接驱动板上的风扇，且此风扇带的有转速反馈，若转速太低会抱警。

XS7为从G2过来的抱闸以及风扇的电源

X1、X23、X21、X4主要是运行链的互锁.若条件不满足，则不会继续运行，接触器就不会吸合

X8连接的三个接触器K42,K43， K44的接触器线圈。

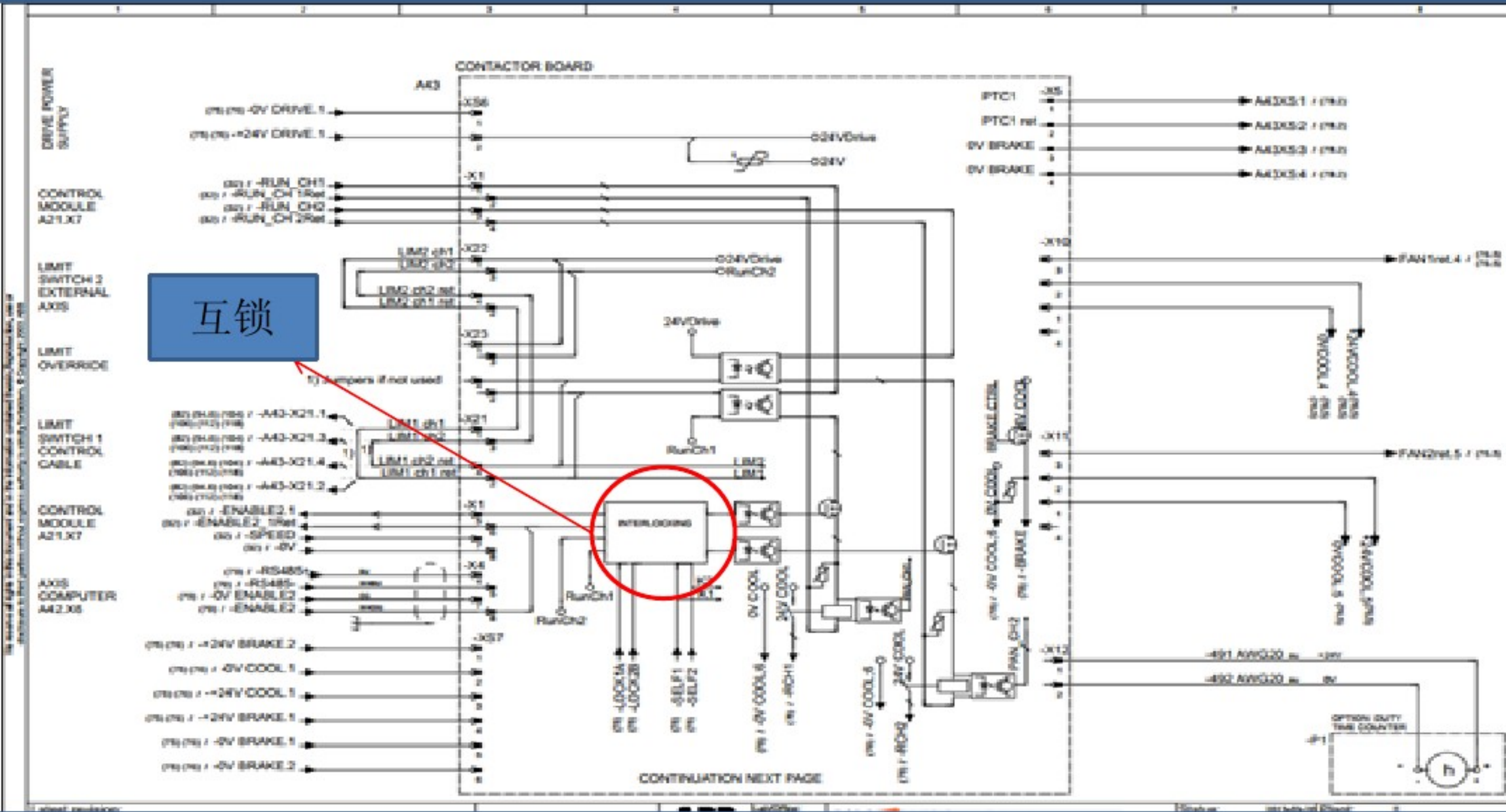
X3主要是三个接触器的辅助触点，分别关联接触器反馈信号，互锁信号，以及刹车反馈信号。

X24接的是外轴的温控

X9是手动自动状态下的松抱闸。

十四、电路图的分析

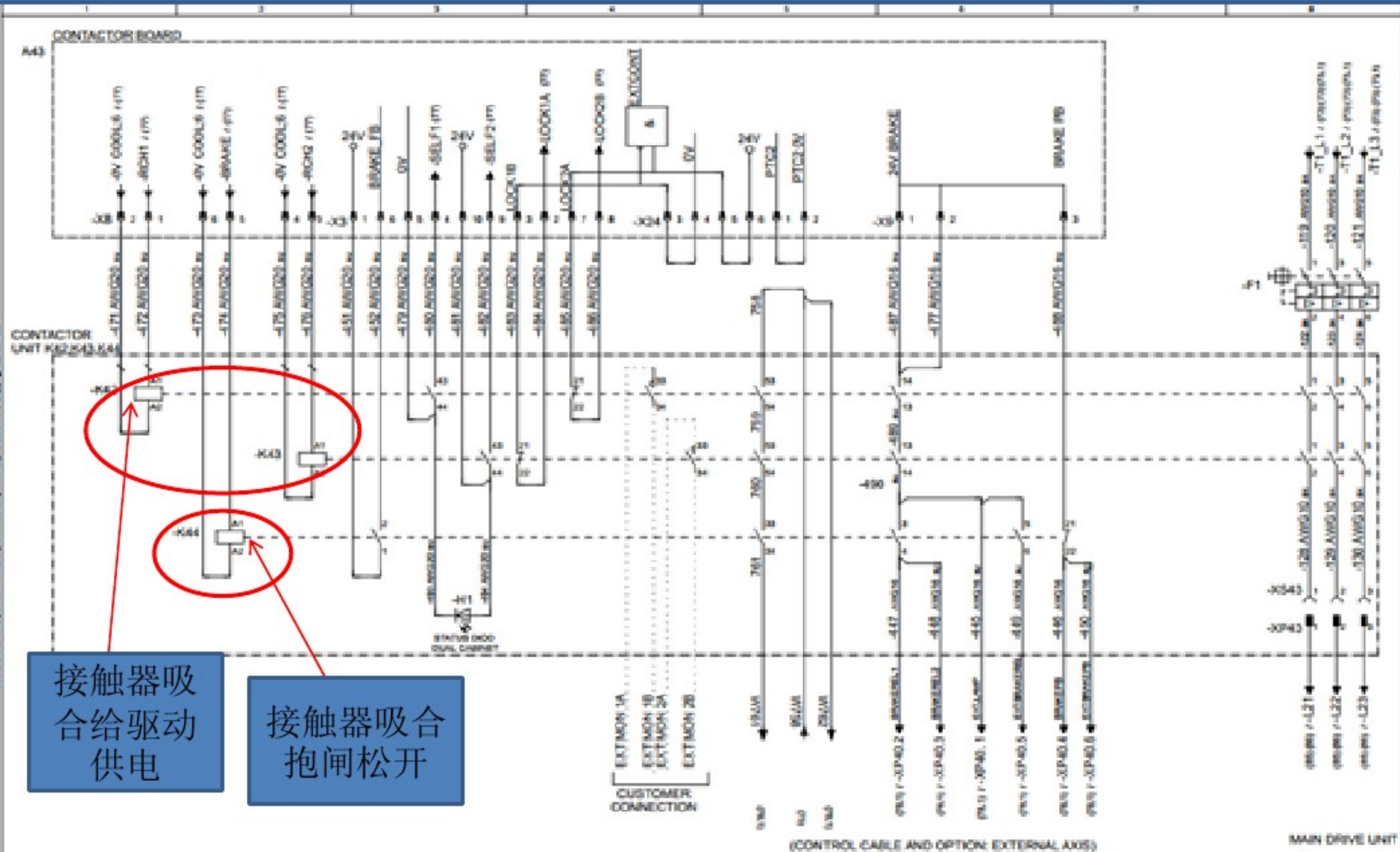
接触器电路图



运行链的回路条件满足后，触发接触器K42，K43吸合（K42、K43的一对常开常闭辅助触点是判断接触器是在吸合状态，还是在没有吸合状态）

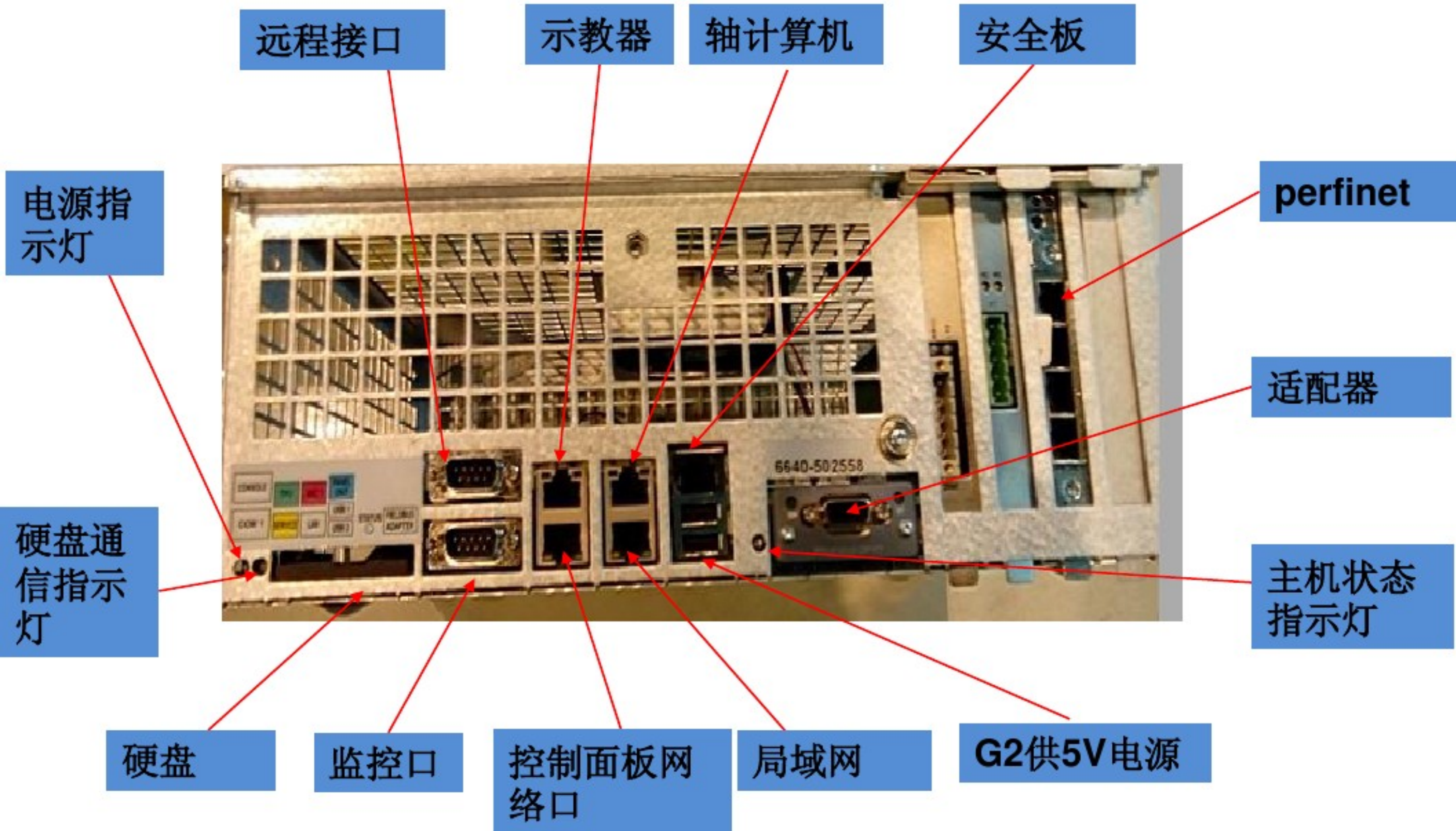
十四、电路图的分析

接触器电路图



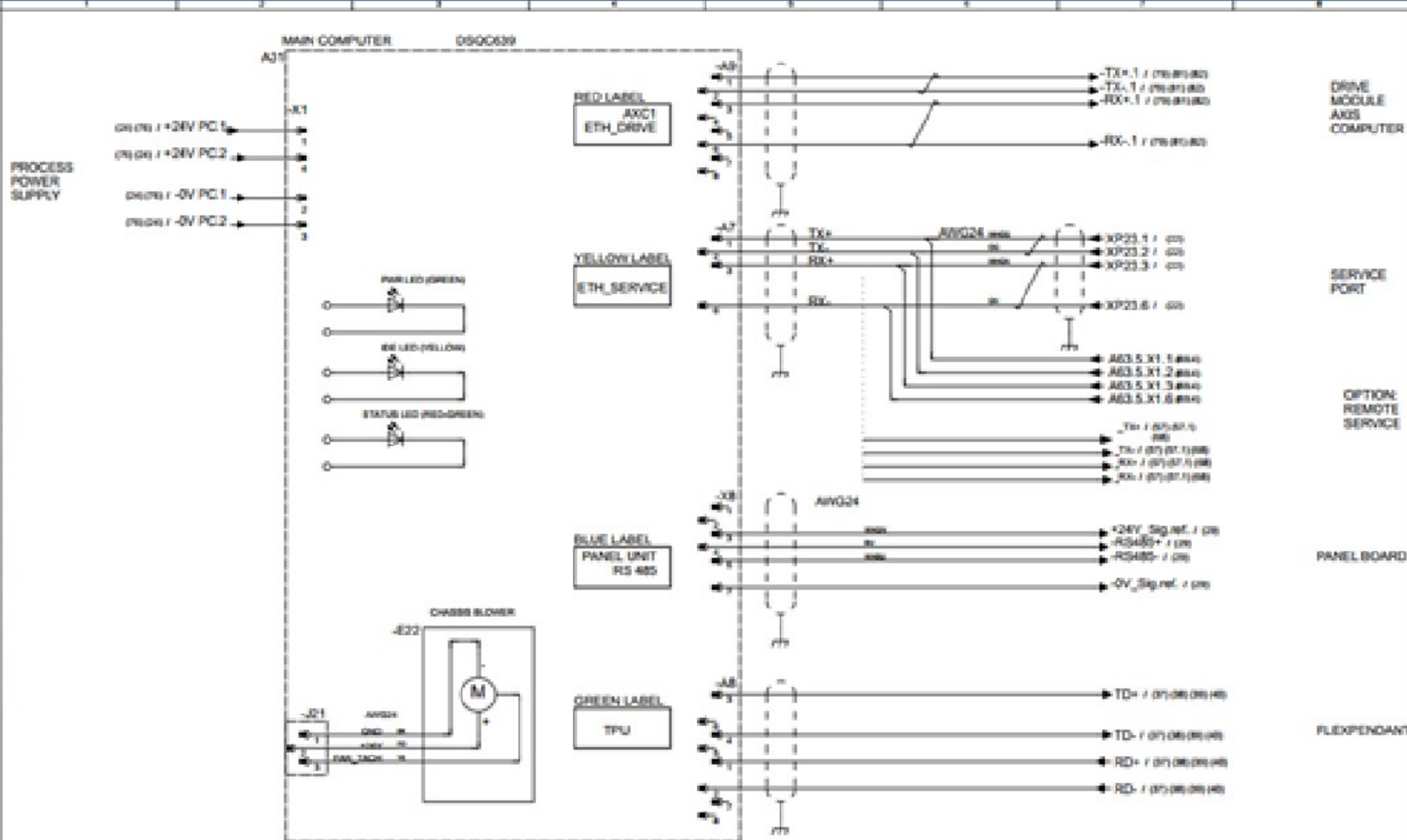
十四、电路图的分析

主计算机



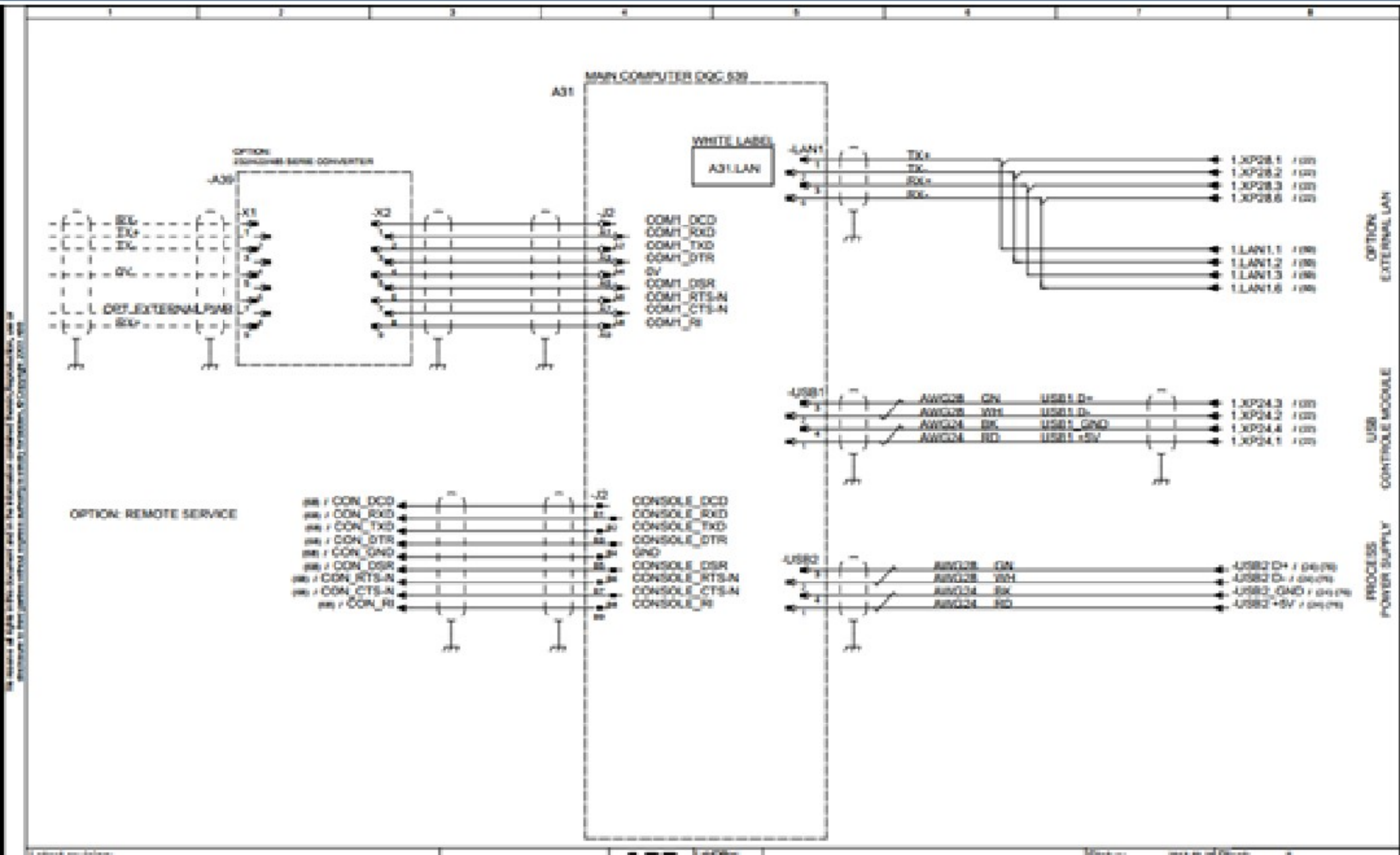
十四、电路图的分析

主计算机



十四、电路图的分析

主计算机



十四、电路图的分析

主计算机

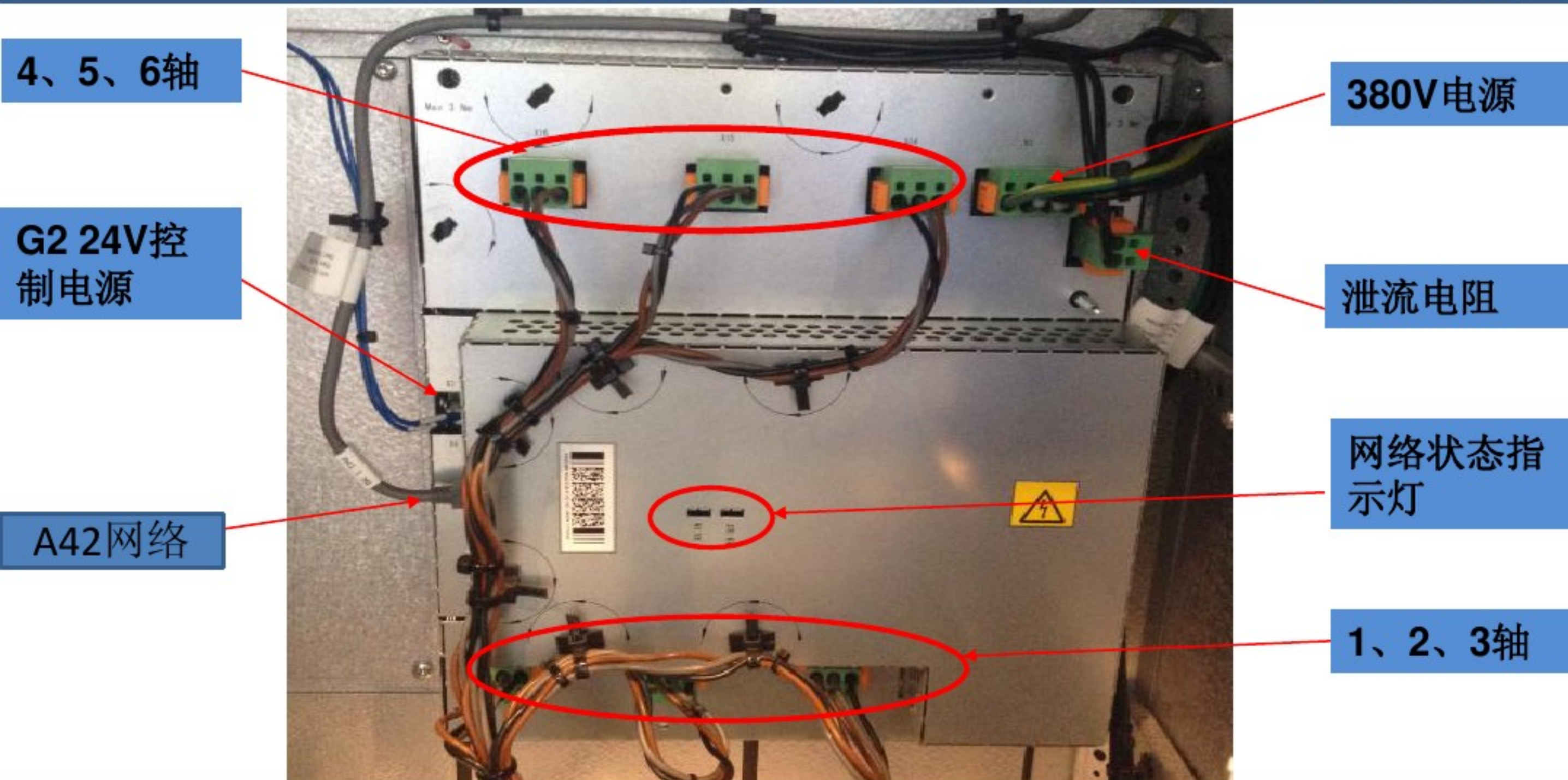
电源指示灯：常亮绿灯： A31是否通电
硬盘通信指示灯：闪烁黄灯：与CPU正在通信
主机通信状态指示灯：启动期间正常顺序
1. 持续红灯：初始化基本硬件
2. 闪烁红灯：
3. 闪烁绿灯：建立与外围设备的
4. 持续绿灯：应用程序正在运行

适配器口：只能将**A31**作为从站
硬盘：为**1G**硬盘，只有断电情况下才能取出
USB接口：有**2个USB**接口，任意一个都可作为给**A31** 供**5V**电源的接口

十四、电路图的分析

驱动单元：作用：驱动六轴

原理： 1.将380V电源整流滤波为两相直流
2.再逆变为570V的交流电（可测量），频率但不高



注意：如有附加轴，则直接取两相直流电给附加轴驱动模块

十四、电路图的分析

泄流电阻

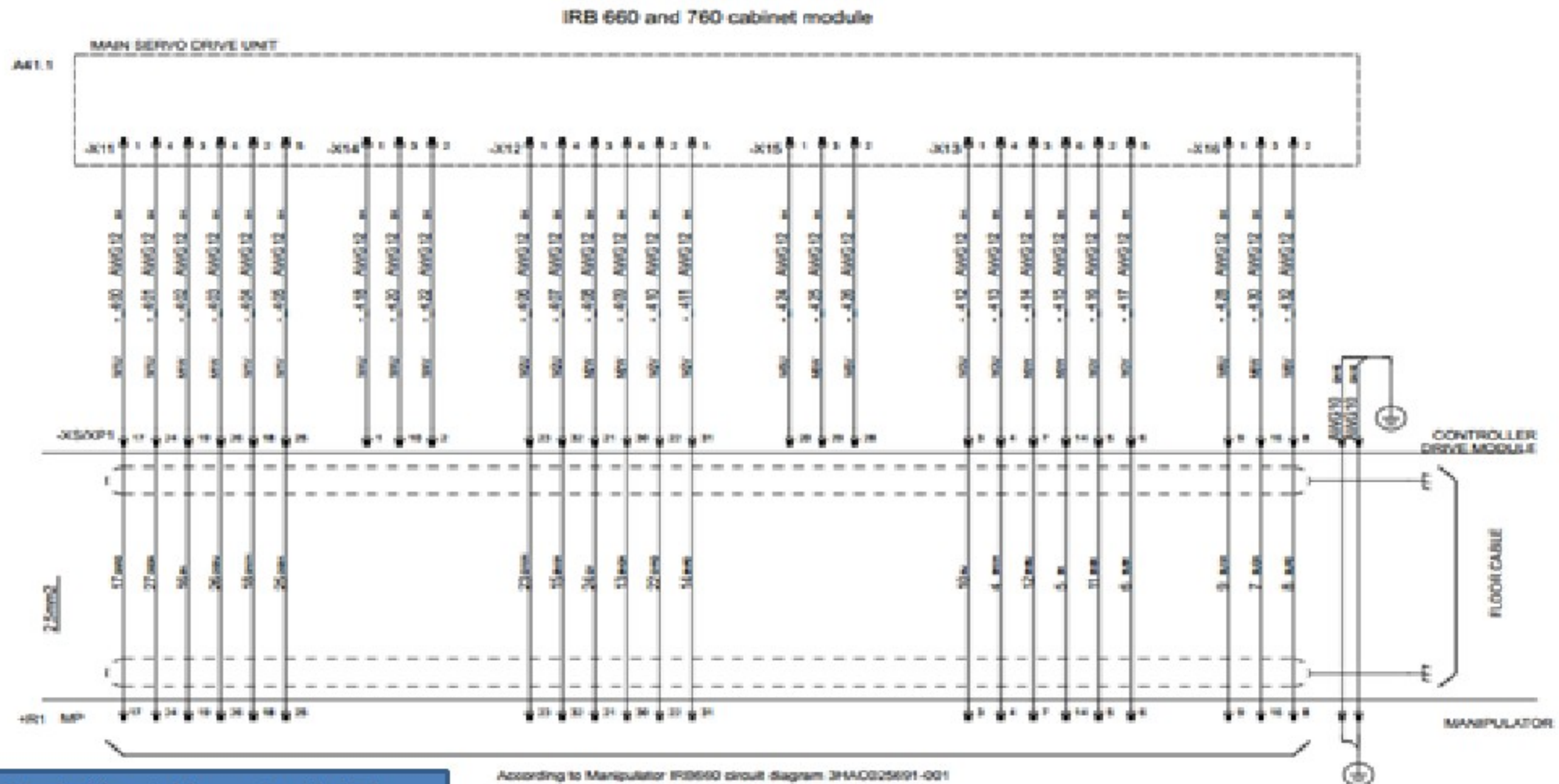
作用：将驱动断电时，电机未刹车所产生的电能（相当于发电机）所消耗掉，以保护驱动。

此泄流电阻和变频器的泄流电阻原理一致



十四、电路图的分析

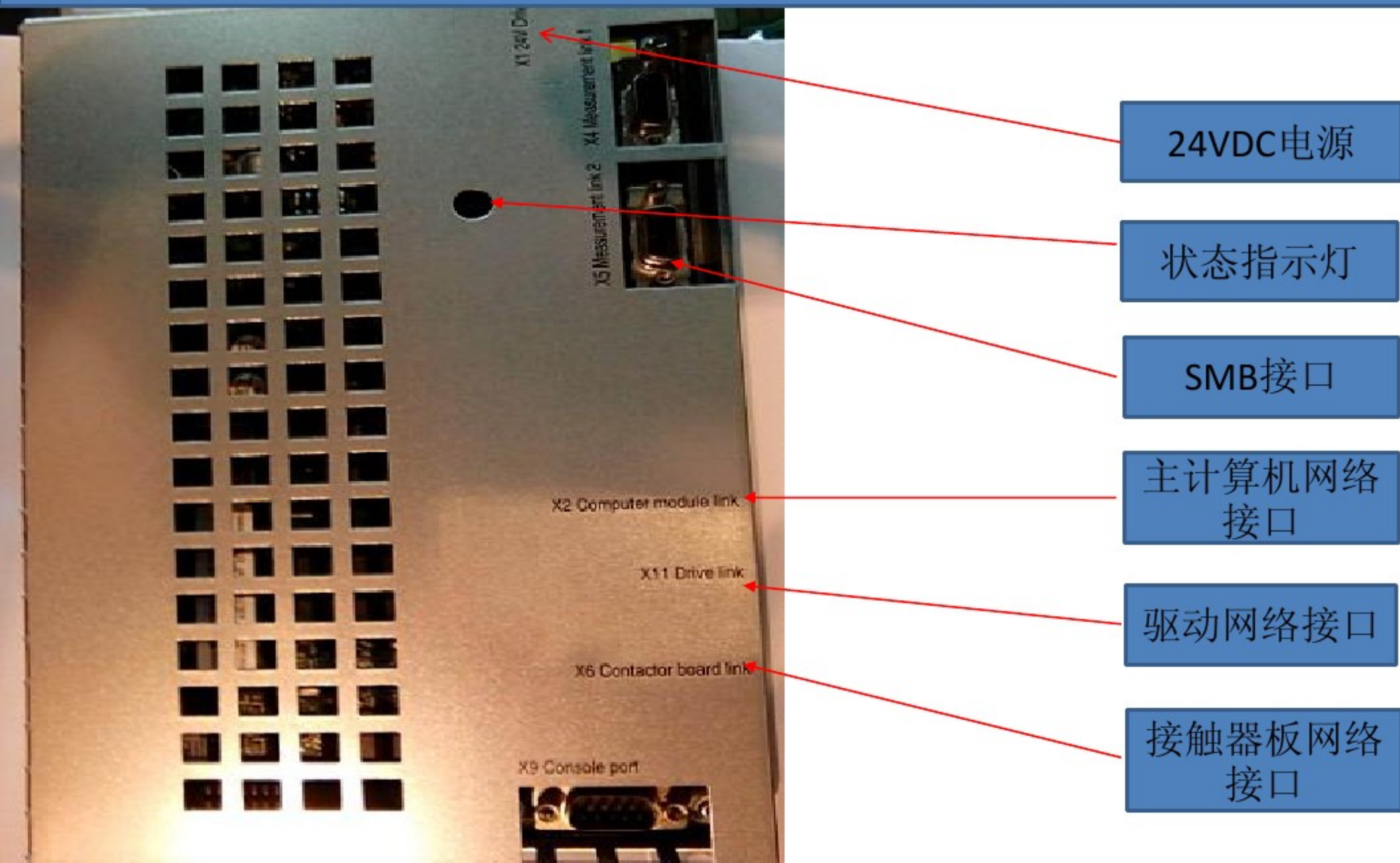
驱动单元



驱动单元的一部分图

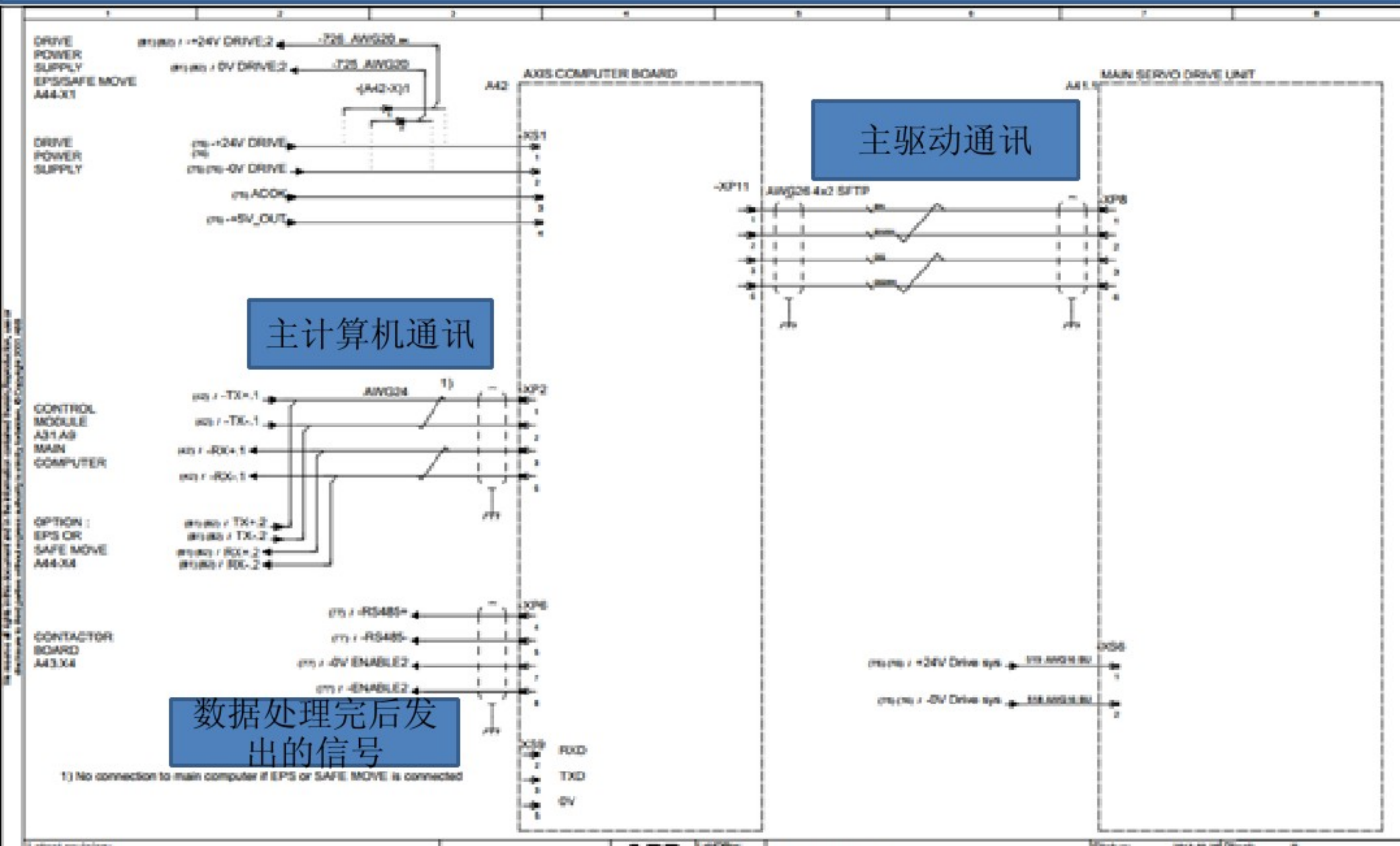
十四、电路图的分析

轴计算机是把机器语言转换成电信号，传送到驱动单元



十四、电路图的分析

轴计算机



十四、电路图的分析

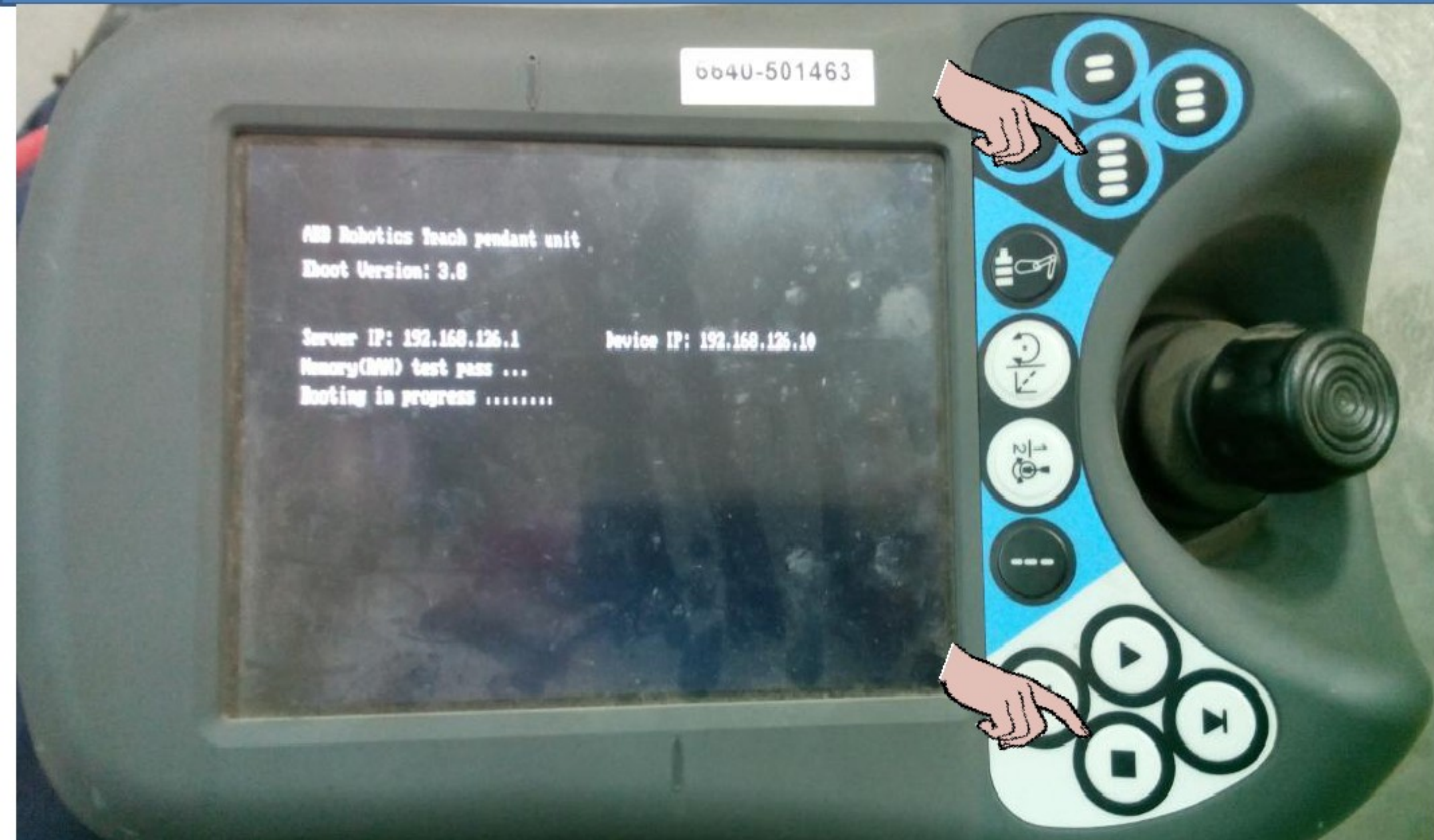
总结

PLC发出工作命令经机器人主计算机处理后，运行示教器上写的程序，并把轨迹位置（主计算机把编程语言转换成机器语言）传送给轴计算机，轴计算机把主机传送的信号处理后（把机器语言转换成电信号），传送给主驱动单元驱动机器人本体运行，在运行的过程中，SMB板记录当前运行的数据（编码器的数据）并传送给轴计算机进行对比（主机发送的轨迹位置和SMB板记录的位置信号对比），当数据相等并立即传送给主计算机，主计算机发出停止信号。

所有的安全信号满足，机器人的整个运行链才能正常工作

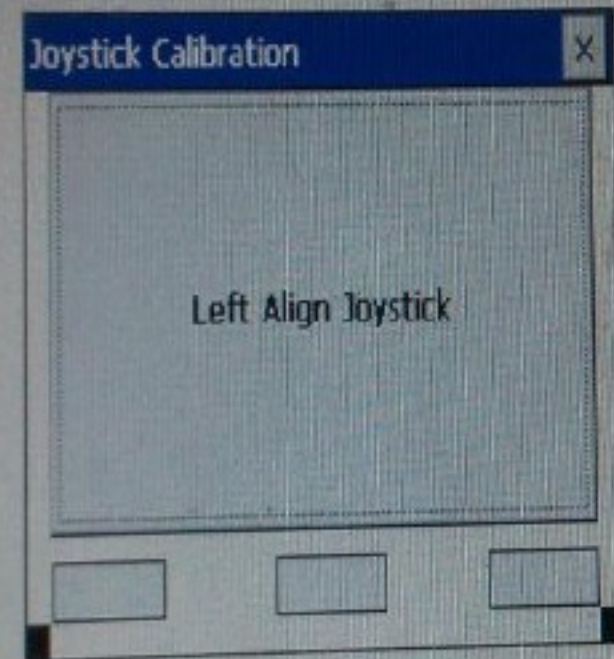
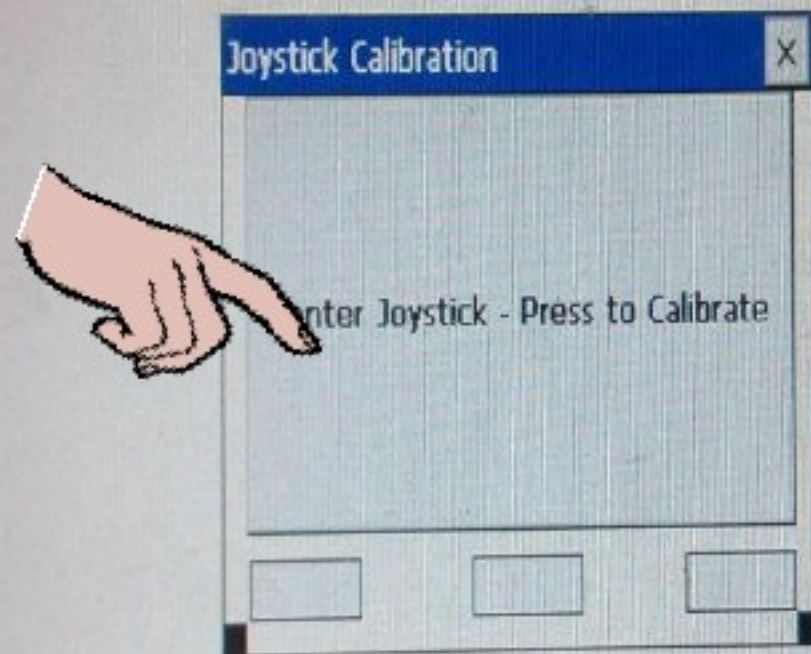
十五、摇杆校正

2、设备在重启的过程中，同时按住（10S左右）



十五、摇杆校正

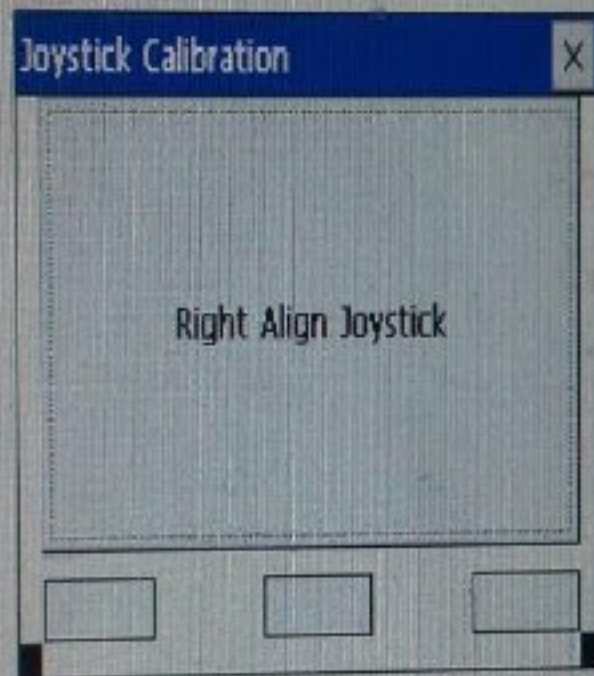
3、点击“center joystick – press to calibrate”



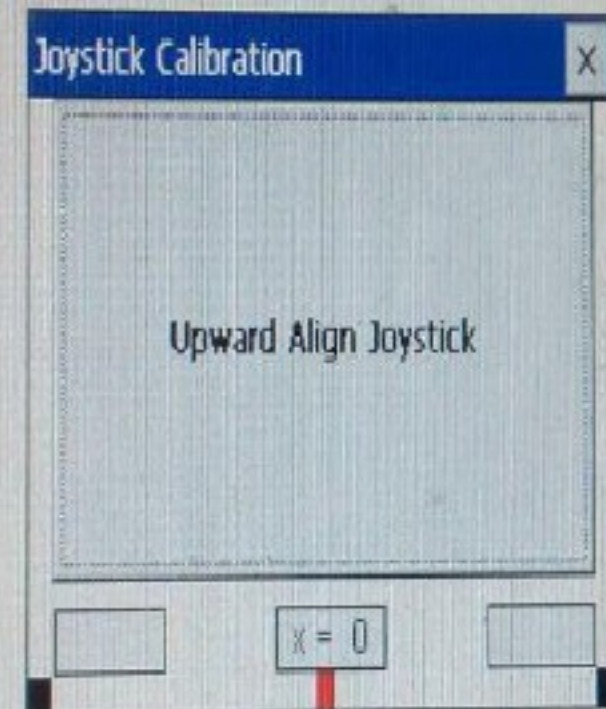
4、向左操作摇杆

十五、摇杆校正

5、向右操作摇杆



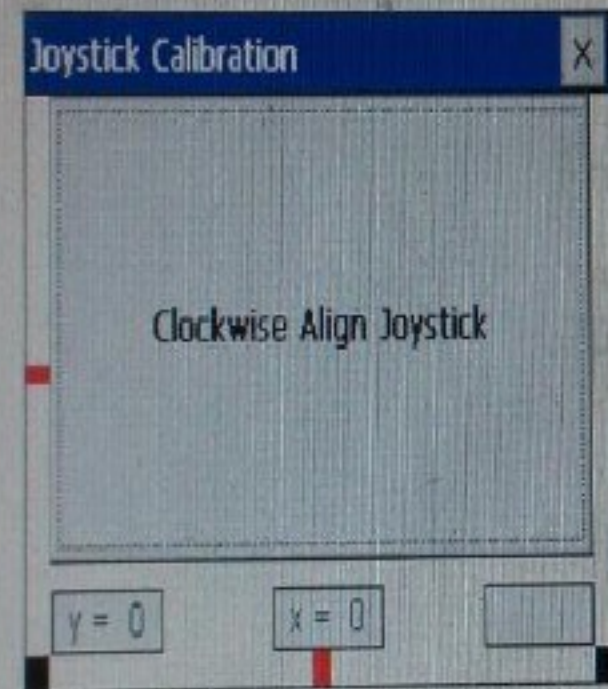
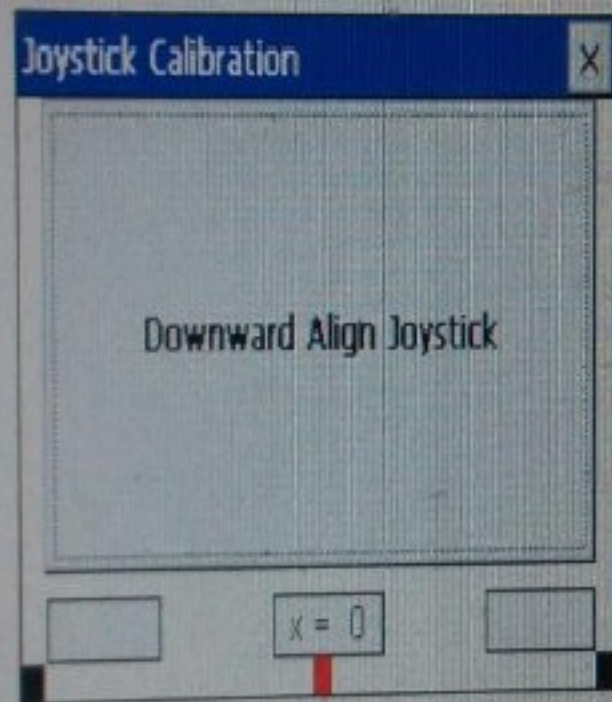
+



6、向上操作摇杆

十五、摇杆校正

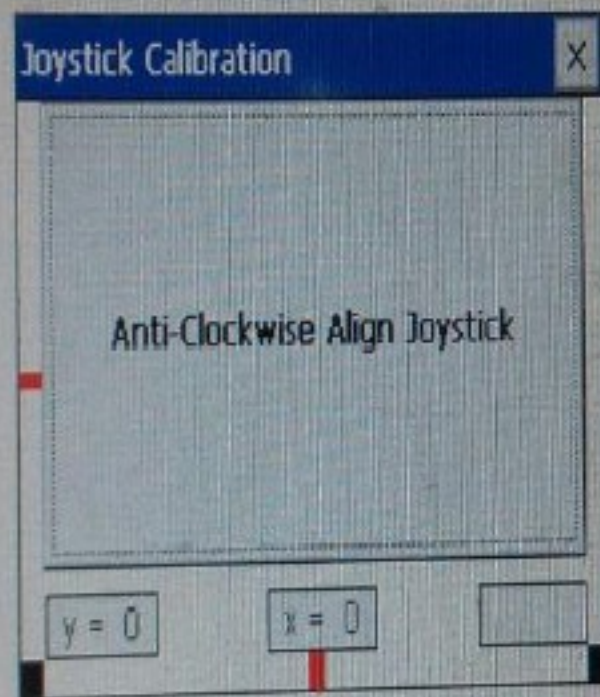
6、向下操作摇杆



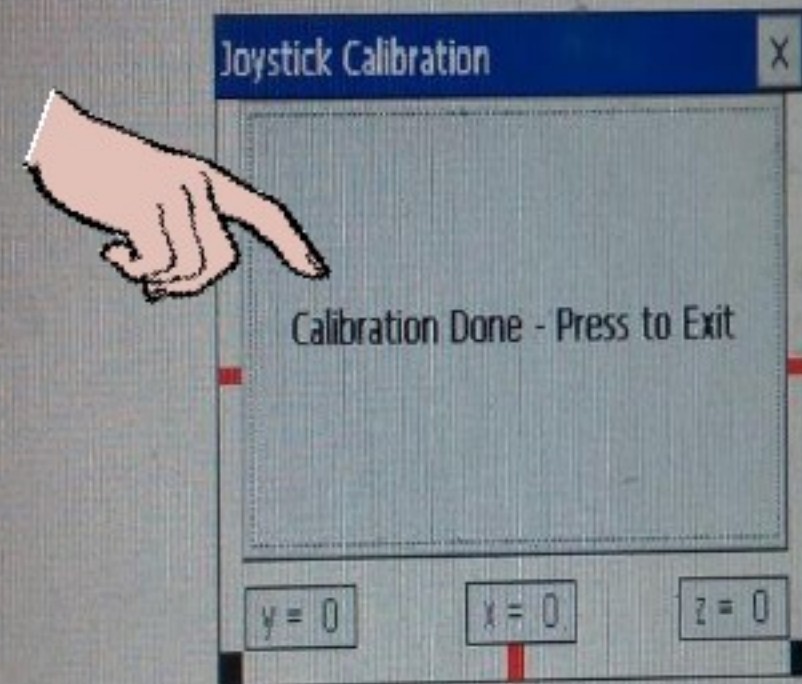
7、向右旋转操作摇杆

十五、摇杆校正

8、向左旋转操作摇杆



+



9、点击中心位置，校准完成，设备重新启动

十六、精校正

将档杆及检测装置固定在机器人上，移动机器人，使检测装置碰到档杆



十六、精校正



十六、精校正



十六、精校正



十六、精校正

