SCATECH 软控

USER MANUAL · 用户手册

SCMotion 4.0 用户使用手册

EtherCAT 主站与运动控制软件：安装、界面、组态、调试与运动控制操作指南

适用软件：SCMotion 4.4.16 及以上

文档版本：V1.0

发布日期：2026-10

PREFACE 前言

关于本手册

本手册介绍软控 SCMotion 4.0（以下简称"SCMotion"或"主软件"）的功能、界面和操作方法，面向设备调试工程师、电气工程师和现场维护人员。用软卡 SDK 自己写上位机程序的开发人员，请同时阅读《软卡 SDK 编程手册》。

手册内容以 SCMotion 内置帮助（软件顶部「帮助」菜单）为准整理编排。菜单、页签名称与软件界面不一致时，以软件界面为准；本手册与内置帮助不一致时，以内置帮助为准。

配套资料	内容
《SCMotion 4.0 用户使用手册》	本书。产品简介、界面与按钮说明、从零让第一根轴动起来、EtherCAT 组态、运动控制各功能、排障。
《软卡 SDK 编程手册》	用 C / C++ / C# 通过软卡 SDK 动态库编程：连接、控制权、全部 SCM_* 指令、排障与案例工程说明。
《案例教程》及配套视频	一个功能一套"图文 + 视频"：数字量 IO 控制、轴使能与点动……陆续增加。
案例工程（C# WinForms / WPF / C++ Qt）	20 个功能案例，一个案例一个独立工程，可直接运行、可整段复制函数。

修订记录

版本	日期	说明
V1.0	2026-10	首次发布。依据 SCMotion 4.4.16 内置帮助整理，新增产品简介、界面按钮说明章节。

阅读约定

写法	含义
「激活配置参数」	软件界面上的按钮、页签、菜单、对话框名称。
`SCM_AxisPower`、`6041`	SDK 指令、CoE 对象索引、文件路径等需要原样输入的内容。
轴号	界面显示从 1 开始（"轴/1"）；SDK 调用从 0 开始（`axis = 0`）。

说明 补充说明或背景信息。

提示 能让操作更顺的做法。

注意 不注意会导致功能异常或调试走弯路。

危险 不注意可能造成设备损坏或人身伤害。

安全须知

危险 第一次让真实电机动之前：电机与机械负载脱开，或负载能自由运动；急停在手边；速度先用最小值。仿真里做错没有后果，真机上没有撤销。

- 真实轴运动前看模式条：要在「实时运行」、全拓扑 OP、WKC 满值。「配置调试」下能读到驱动器状态，不等于可以安全运动。
- 实际位置和规划位置差得远时，先「复位」让两者一致，再使能、再运动，否则轴会猛追目标。
- 软件急停（`SCM_AxisEStop`）不能替代硬件急停回路。安全相关场合必须配硬件急停。
- 写输出会真实驱动端子。确认外部负载安全后再写；「DO 走位巡检」「全部置位」会逐点或整片通断，先确认不会误动执行机构。
- 「开机自动进入运行(OP)」等于上电就带真实硬件运行，现场不希望这样就不要勾选。
- 一次只验一个现象：先一个输出点，再一个状态字，最后一根轴；混挂链上一次只动一根轴。

CONTENTS 目录

第一篇 · 入门

第1章 产品简介 11

1.1 产品概述 11

1.2 主要功能 11

1.3 关键指标 11

1.4 运行环境 12

1.5 已实测适配的伺服品牌 12

1.6 授权 12

第2章 快速入门 13

2.1 第一次打开看到什么 13

2.2 进「配置调试」要过三个框 14

2.3 先在仿真里走一遍 14

2.4 上真机前的准备 14

2.5 推荐流程 14

2.6 锁定与托盘 15

第3章 界面说明 16

3.1 顶部菜单 16

3.2 工具栏 16

3.3 左侧工程树 17

3.4 右侧页面 17

3.5 消息栏与状态栏 17

3.6 界面锁定 18

3.7 托盘 18

3.8 界面密码 18

3.9 按钮速查：主窗口 18

3.10 按钮速查：轴「在线」页 20

3.11 按钮速查：「检查配置」对话框 21

3.12 按钮速查：从站「I/O 测试」页 22

3.13 按钮速查：网卡管理与实时设定 22

第4章 真机第一轴 24

4.1 开始前 24

4.2 第 1 步：装驱动、接管网卡 24

4.3 第 2 步：接线、新建工程 24

4.4 第 3 步：扫描总线 25
4.5 第 4 步：建轴、映射从站 25
4.6 第 5 步：激活、进 OP 26
4.7 第 6 步：使能 27
4.8 第 7 步：低速点动 27
4.9 顺手验一个 I/O 点 28
4.10 回零 28
4.11 从一根轴到整条链 28

第5章 工程、模式与总线 30
5.1 工程与项目名称 30
5.2 三种模式 30
5.3 模式条与两个动作 30
5.4 开机自启动 31

第6章 实时形态与性能 32
6.1 一种后端，两档形态 32
6.2 认证网卡 33
6.3 周期与轴数由三层决定 33
6.4 总线时间可以直接算 34
6.5 判断能否稳定运行 35
6.6 样机实测参考 35
6.7 怎么测出自己设备的上限 36
6.8 余量不够时可以做什么 36

第二篇 · EtherCAT 与 I/O

第7章 EtherCAT 与从站 38
7.1 离线组态与真实总线 38
7.2 选网卡 38
7.3 扫描与拓扑对比 38
7.4 上线检查表 39
7.5 主站页面 40
7.6 从站页面 40
7.7 从站状态机 41
7.8 同步模式与布局依据 41
7.9 常用 CiA402 对象 42
7.10 没有说明文件的设备 42

第8章 PDO 与过程数据 44

8.1 基本规则 44
8.2 过程数据页怎么看 44
8.3 改映射 44
8.4 启动参数与下发 45
8.5 看在线值 45
8.6 映射排错 46

第9章 CoE 在线与启动参数 47

9.1 CoE 在线页 47
9.2 读值 47
9.3 改一个参数 48
9.4 运行中（OP）读写 CoE 的规则 48
9.5 启动参数 48
9.6 标准区与厂商区 49
9.7 一圈多少脉冲：用 6064 实测 49

第10章 从站 I/O 功能测试 51

10.1 打开 I/O 测试页 51
10.2 页面读法 51
10.3 测输入、写输出 51
10.4 业务名 52

第三篇 · 运动控制

第11章 轴运动与安全边界 54

11.1 三条运动链 54
11.2 哪个模式下能动 54
11.3 受理不等于完成 55
11.4 轴页上的运动控件 55
11.5 实际值与规划值 55
11.6 停止与复位 56
11.7 单轴 S 曲线（Jerk） 56
11.8 安全边界 56
11.9 ErrorID 对照表 57
11.10 典型顺序 58
11.11 错误处理原则 58

第12章 回零 59

12.1 三种回零方式 59
12.2 驱动器回零：相关对象 60

12.3 驱动器回零：启动与完成时序 60
12.4 图例：怎么读本章的图 61
12.5 方法总览 61
12.6 方法 1：负限位开关 + Z 相 62
12.7 方法 2：正限位开关 + Z 相 63
12.8 方法 3 / 4：正侧原点开关 + Z 相 63
12.9 方法 5 / 6：负侧原点开关 + Z 相 63
12.10 方法 7 – 10：中段原点开关 + Z 相（初始正向，配合正限位） 64
12.11 方法 11 – 14：中段原点开关 + Z 相（初始负向，配合负限位） 64
12.12 方法 17 / 18：仅限位开关（不用 Z 相） 65
12.13 方法 19 / 20：正侧原点开关（不用 Z 相） 66
12.14 方法 21 / 22：负侧原点开关（不用 Z 相） 66
12.15 方法 23 – 26：中段原点开关（不用 Z 相，初始正向） 66
12.16 方法 27 – 30：中段原点开关（不用 Z 相，初始负向） 67
12.17 方法 33 / 34：仅 Z 相 67
12.18 方法 35 / 37：当前位置置零（不运动） 68
12.19 负值方法：厂商自定义 68
12.20 选型建议与常见问题 68

第13章 插补通道 70

13.1 通道与四个页签 70
13.2 映射并使能 70
13.3 段表能力 71
13.4 单位与导入缩放 72
13.5 运行与观察 72
13.6 参数页速查 73
13.7 SCMotion CAD 速查 74
13.8 接真轴 74
13.9 常见问题 75
13.10 下一步 76

第14章 CNC 与 G 代码 77

14.1 建通道 77
14.2 第 1 步：映射坐标轴 77
14.3 第 2 步：写 / 载入 G 代码 77
14.4 支持的 G 代码 78
14.5 支持的 M 代码 80
14.6 不支持的常用代码（会编译报错，不是忽略） 80

14.7 已知限制（从其他数控系统迁移程序时需特别注意） 81
14.8 刀补的限制（G41/G42） 81
14.9 第 3 步：运行与观察 82
14.10 用程序跑它（软卡 SDK） 82
14.11 参数页 83
第15章 电子齿轮 84
15.1 建通道 84
15.2 第 1 步：选主从轴并使能 84
15.3 第 2 步：设齿比与追赶参数 84
15.4 第 3 步：啮合并观察 85
15.5 完成条件与状态 85
第16章 电子凸轮 86
16.1 建通道 86
16.2 第 1 步：映射主从轴并使能 86
16.3 第 2 步：编辑凸轮表 86
16.4 第 3 步：啮合并观察 87
16.5 表是怎么下发的 87
16.6 运行中换表 87
16.7 完成条件与状态 87
16.8 文件 88
第17章 机器人总览 89
17.1 三种构型 89
17.2 常规页的三个按钮 89
17.3 页面与 SDK 是同一条链 89
17.4 单位与位姿约定 90
17.5 受理、完成与故障 90
17.6 点动是步进制 90
17.7 三种模式下的行为 90
第18章 Delta 机器人 91
18.1 机构与坐标约定 91
18.2 工作区有多大 91
18.3 四个页面怎么用 92
18.4 软限位怎么设 92
18.5 Delta 上没有的东西 92
第19章 SCARA 机器人 94

19.1 四个关节各管什么 94
19.2 第 1 步：填机构参数，看清能到哪儿 94
19.3 第 2 步：映射四根关节轴 94
19.4 第 3 步：让它动起来 95
19.5 贴近工作圆环边界时会发生什么 95
19.6 肘分支 95
19.7 TCP 工具偏移与工件坐标 95
19.8 在线页两个 Z、示教表两个 Z 和两个 C 96
19.9 示教点：记录与回放 96
19.10 正逆解校验器怎么读 96
19.11 三维视图 96

第20章 6 轴串联机器人 98

20.1 六个关节各管什么 98
20.2 姿态角 C / B / A 是什么 98
20.3 第 1 步：填机构参数 99
20.4 第 2 步：映射六根关节轴 99
20.5 第 3 步：让它动起来 99
20.6 先 PTP 建臂形，再 LIN 走直线 100
20.7 TCP 工具偏移与工件坐标 100
20.8 示教点：记录与回放 100
20.9 正逆解校验器怎么读 100
20.10 三维视图 101

第四篇 · 软卡 SDK

第21章 软卡 SDK 上手 103

21.1 谁用它、前提是什么 103
21.2 SDK 包里有什么 103
21.3 连接配置文件 103
21.4 最短路径 104
21.5 最小程序 104
21.6 第一根轴现场步骤 107
21.7 C/C++ 工程设置 107

第22章 软卡设置 109

22.1 从哪里进 109
22.2 页面上有什么 109
22.3 软卡 SDK 服务是什么 109

22.4 控制权 110
22.5 生成 SDK 包 110
22.6 常见问题 110

第五篇 · 排障与参考

第23章 故障排查 112
23.1 先确认当前模式 112
23.2 扫不到网卡或从站 112
23.3 进不了实时运行 112
23.4 运行中掉线或退到 SAFE-OP 113
23.5 使能失败 113
23.6 运动按钮灰或拒绝运动 114
23.7 现象速查 114
23.8 AL 状态码 115
23.9 CiA402 状态字（6041） 116
23.10 需要技术支持时 117

第24章 日志 118
24.1 消息栏上看到的和日志里记的 118
24.2 右键菜单 118
24.3 退出会删除日志，请在退出前导出 118
24.4 启动诊断全文 119
24.5 先看哪几行 119
24.6 常见消息 119
24.7 报障要带什么 120

附录A 术语 121
A.1 模式与运行 121
A.2 EtherCAT 122
A.3 轴与运动 122
A.4 软卡 SDK 123
A.5 容易混淆的边界 124

PART 01 · 第一篇

入门

认识 SCMotion，熟悉界面，从一台新电脑走到第一根伺服电机低速转起来。

产品简介 · 快速入门 · 界面说明 · 真机第一轴 · 工程、模式与总线 · 实时形态与性能

第1章 产品简介

SCMotion 是什么、能做什么、需要什么样的电脑和网卡。

1.1 产品概述

软控 SCMotion 是一套跑在 Windows 电脑上的 EtherCAT 主站加运动控制软件：用一块认证网卡直接驱动从站，在同一个界面里做组态、调试和运动，并通过软卡 SDK 给客户程序调用。从 EtherCAT 总线到 PLC、插补、CNC、机器人，一个平台打通运动控制全流程，不需要额外的运动控制卡。



图 1-1　SCMotion 系统构成

1.2 主要功能

类别	功能
EtherCAT 主站	扫描拓扑、ESI 设备库、PDO 和启动参数组态、CoE 在线、从站 I/O 功能测试、主站和从站在线诊断。没有说明文件的设备也能识别和控制。
运动控制	单轴点动 / 寸动 / 定位 / 回零（七段 S 曲线）、多轴插补通道与 CAD 导入、CNC 与 G 代码、电子齿轮、电子凸轮、Delta / SCARA / 6 轴机器人。
软卡 SDK	C 和 .NET 两套同名同参的接口。客户程序连主软件即可驱动真机，「仿真」下不接硬件也能开发。
配套工具	示波器、CAD 绘图、SCRemote 远程桌面，都在「工具」菜单里。

1.3 关键指标

250 µs
档1 独占核 承诺周期

2 ms
档2 共享核 承诺周期（虚拟机、受管电脑可用）

C / .NET
两套同名同参 SDK 接口

3 种模式

仿真 · 配置调试 · 实时运行

样机实测参考（详见第 6 章「实时形态与性能」）：档1 32 轴混挂（步进 + 伺服四个家族）250 µs 周期成功率 100%；档1 20 站 250 µs 26.6 分钟 639 万拍零丢帧。实测结论随控制器与驱动器不同而变化，最终以在自己设备上的测试为准。

1.4 运行环境

项目	要求
操作系统	Windows x64。实时驱动要求关闭 Secure Boot，可在「SYSTEM → 实时设定」页检查运行环境。
CPU	档1 独占核：至少 4 核，且不能开虚拟化（Hyper-V / 虚拟机平台 / 内存完整性）。档2 共享核：无此要求。
EtherCAT 网卡	Intel I210 / I211 / I350 / I217 / I218 / I219 系认证网卡。不支持 Intel I225 / I226 及 Realtek 等非 Intel 网卡（这些机器只能用仿真模式走通流程）。
从站	EtherCAT 伺服（CiA402）、步进驱动器、I/O 模块与耦合器；网线从电脑接第一台的 IN 口，再从 OUT 口接下一台的 IN 口。
SDK 开发	.NET 8 / Visual Studio 2022，或 C/C++ x64 工程。

1.5 已实测适配的伺服品牌

品牌	型号	品牌	型号
松下	MINAS A6 · MADLT15BF	汇川	SV660N · SV670N
英威腾	DA180A · DA200A	雷赛	L6N
信捷	DS5C1	高创	CD02

扫描即识别、自动建轴。其它品牌的 EtherCAT 从站按 ESI 说明文件或设备自身存储的配置接入，见第 7 章「EtherCAT 与从站」。

1.6 授权

- 按计费节点授权：计费节点 = 总从站数 − 自有品牌 I/O 模块数。免费档内不用导入授权文件。
- 「授权管理」页：「复制机器码」发给软控换授权文件，「导入软控授权」装上，「刷新」重读。
- 看到「授权尚未生效」是本机时钟比签发时间早，改对时钟再刷新；永久授权不看签发时间。

第2章 快速入门

第一次打开软件时看什么、按什么顺序做。接真实电机的逐步操作在「真机第一轴」。

2.1 第一次打开看到什么

软件打开后停在「配置调试」模式。左边是工程树，右边是当前选中对象的页面，底部是消息栏。工具栏上的三段模式条「仿真」「配置调试」「实时运行」既显示当前模式，点哪一段就切到哪一段；模式条右边是「界面锁定」开关，再往右两个小图标是「激活配置参数」和「重启系统」。



图 2-1 软件打开后的主窗口（SCMotion 4.4.16，新建项目、配置调试模式，工程树选中「SYSTEM - 配置」）



工具区模式条：仿真 / 配置调试 / 实时运行，右侧为「界面锁定」开关与实时核指示

- 「仿真」：不碰网卡和硬件。软件内置一套完整的离线运行时，工程、轴、插补、CNC、机器人、软卡 SDK 客户程序都能在里面跑。
- 「配置调试」：连总线做组态和联调。改 PDO、启动参数、CoE 在线、简单 I/O、诊断。
- 「实时运行」：真机 OP。真实轴的点动、寸动、定位、回零只在这里可用。

三种模式的完整对照见第5章「工程、模式与总线」。

> **说明** **进度条在跑时先等。**软件启动或切模式时，工具栏右侧会出现一条进度条。等它消失再点扫描或模式条，不要重复点。

2.2 进「配置调试」要过三个框

打开后模式条虽然停在「配置调试」，总线还没连上。点一下模式条上的「配置调试」，依次弹出：

对话框	选什么
「进入配置调试模式」	确定。过程数据先停，总线回到可配置状态。
「加载工程 I/O 拓扑」	是。把工程里保存的从站显示出来，不扫描硬件。新工程没有拓扑时这个框不出现。
「进入 FreeRun / OP」	否：只连总线不跑过程数据，可以改组态。是：进 FreeRun，能看 I/O 和实时数据，但组态改不了。

要改 PDO 映射、启动参数这类组态，第三个框选否。要看 I/O 点和在线值，选是。

2.3 先在仿真里走一遍

不接硬件，先点「仿真」，把工程建好、轴加上、点动和插补流程跑一遍。这一步出错不伤设备。仿真和真机的差别只有三处：没有真实回零开关；齿轮/凸轮啮合瞬时完成；I/O 是内存里的位。

2.4 上真机前的准备

项目	检查什么
软件	运行的是安装目录里的软件，不是别处留下的旧版本。版本号在帮助窗口右上角。
网卡	EtherCAT 专用口是认证网卡（Intel I210 / I219 系），并且已在「网卡管理」里装了实时驱动、被接管。型号清单见第6章「实时形态与性能」。
从站供电与接线	驱动器、I/O 模块、耦合器都上电。网线从电脑接第一台的 IN 口，再从它的 OUT 口接下一台的 IN 口。台数与工程一致。
机械安全	急停在手边。电机与负载脱开，或负载能自由动。限位、软限位、速度先取小值。

2.5 推荐流程

1. 「文件 → 新建项目」或打开工程，先保存一次。
2. 切到「配置调试」，第三个框选否。
3. 工程树里右键 EtherCAT 主站节点 → 「扫描」，在「检查配置」对话框里把扫到的设备加入工程。
4. 识别为驱动器的从站会提示自动建轴。也可以自己在 NC-task 下「添加轴」，再到轴「设置」页「选择映射」。
5. 按需要改 PDO、启动参数、轴当量与软限位。

6. 点「激活配置参数」，看从站全部到 OP、主站「在线」页 WKC 满值。工程没改、只是重新起总线时，点模式条上的「实时运行」就行。
7. 在轴「在线」页使能，先低速点动确认方向与当量。
8. 再加从站、加轴，上插补 / CNC / 机器人，或用软卡 SDK 让自己的程序驱动。

每一步该看到什么、不对怎么办，见第4章「真机第一轴」。

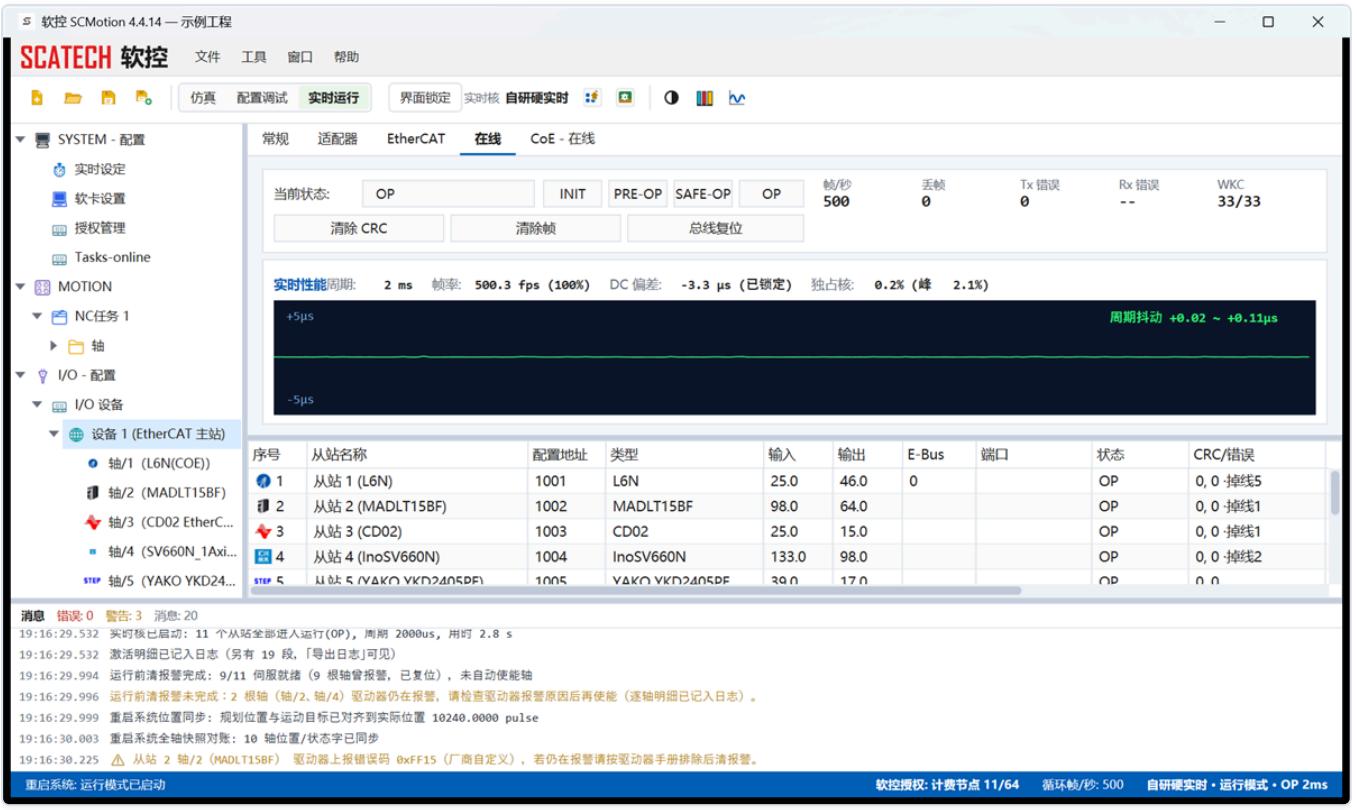
> **提示** 一次只验一个现象：先一个输出点，再一个状态字，最后一根轴。一开始就上复杂路径，出了问题分不清是当量、映射还是总线。

2.6 锁定与托盘

机器在生产时只看不碰，按下工具栏的「界面锁定」。关掉窗口软件进托盘继续跑，真退出走托盘的「退出并停止」。两者的细节和托盘颜色见「界面总览」。

第3章 界面说明

主窗口每一块是什么、点哪里，以及托盘、界面锁定和界面密码怎么用。本章末尾按窗口列出每个按钮的作用。



主窗口：左侧工程树、右侧当前对象的页面、底部消息栏与状态栏

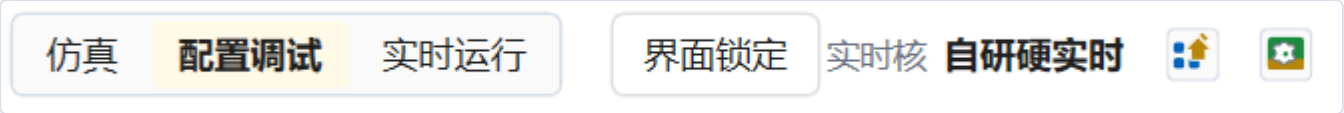
3.1 顶部菜单

菜单	项	说明
文件	新建项目 / 打开项目 / 保存项目 / 另存为 / 修改项目名称	工程文件管理。
工具	ESI 设备库 / 示波器 / CAD 绘图 / SCRemote 远程桌面 / 界面密码	示波器与 CAD 是独立程序，由主软件拉起。界面密码见本章末尾。
窗口	主窗口	只有这一项。
帮助	开始使用 / EtherCAT 与 I/O / 运动控制 / 软卡 SDK 编程 / 排障与参考 / 关于软件	前五项就是帮助中心左侧目录的五个分组，点一组打开这组的第一章。

3.2 工具栏

从左到右三组：

- 项目图标：新建项目、打开项目、保存项目、另存为项目。
- 三段模式条「仿真」「配置调试」「实时运行」，旁边是「界面锁定」开关；再往右两个小图标「激活配置参数」「重启系统」。模式条高亮的那段就是当前模式，点另一段就切换。
- 小图标：切换深色/浅色主题、设备库、示波器。



工具区模式条：仿真 / 配置调试 / 实时运行，右侧为「界面锁定」开关与实时核指示

「实时运行」这一段是按当前工程配置重启进 OP；要把改过的组态下发下去，点「激活配置参数」。两者的区别见第5章「工程、模式与总线」。工具栏最右侧有一条进度条，软件忙的时候才出现，等它消失再点。

3.3 左侧工程树

- 「SYSTEM - 配置」：实时设定、软卡设置、授权管理、Tasks-online。开机自启的两个勾选在这一页。
- 「I/O - 配置」：EtherCAT 主站、从站。右键主站节点有「扫描」。
- 「MOTION」：NC-task、轴、插补、电子齿轮、电子凸轮、CNC 通道、Delta / SCARA / 6 轴机器人。右键 NC-task 有「添加轴」等。

3.4 右侧页面

- 点父节点（SYSTEM、I/O、MOTION）只显示概览页。
- 点主站：常规 / 适配器 / 在线 / CoE - 在线 / EtherCAT。网卡在「适配器」页的「网卡管理...」里选；运行状态、WKC、DC 偏差在「在线」页。
- 点从站：常规 / EtherCAT / 分布式时钟 / 启动参数 / 扩展模块 / 过程数据 / CoE 在线 / 在线 / I/O 测试 / 诊断。
- 点轴：常规 / 设置 / 参数 / 动态 / 功能 / 在线。映射在「设置」页；使能、点动、寸动、回原点、位置定义都在「在线」页；「功能」页是往复、定位这类扩展启动试验。
- 在线表只在过程数据跑着的时候才是真实状态，离线时那些数字不代表设备。

3.5 消息栏与状态栏

底部消息栏只显示简要的错误（红）和警告（黄）消息，排障明细全部写进日志文件。在消息栏里点右键，菜单四项：「导出」把本次运行的完整日志（包括消息栏里不显示的明细）导出成文件；「清除」清掉消息栏和软件记着的历史，磁盘上的日志文件不动；「复制单条」「复制全部」把消息复制到剪贴板。软件正常退出会删掉本次运行的日志文件，要留日志就在运行期导出。

最底下状态栏：左边是扫描/总线状态提示；右边依次是授权状态、循环帧/秒、当前模式与总线状态。仿真模式下会多一个黄色「仿真」标。



消息栏与状态栏，以及消息栏的右键菜单：导出、清除、复制单条、复制全部

3.6 界面锁定

工具栏上的「界面锁定」开关任何模式下都能按。锁定后本软件不写 I/O、不改参数、不动总线状态、不发运动命令，观察与诊断照常。它只锁本软件，软卡 SDK 客户程序不受影响。不随工程保存，重启软件即解除。

3.7 托盘

关掉主窗口，软件进托盘继续跑总线。托盘菜单有「显示主界面」「退出并停止」两项，双击图标等于「显示主界面」。真要退出走「退出并停止」。图标外框颜色就是当前状态：

颜色	含义
紫	仿真。
蓝	配置调试。
绿	实时运行，过程数据在跑。
红	停止：在实时运行档，但过程数据没在跑。

鼠标悬停在图标上显示同样的状态文字。

3.8 界面密码

托盘的「显示主界面」「退出并停止」、双击托盘图标，以及再次启动程序唤起主界面，都要先输界面密码。它防的是机台上有人随手打开界面或停掉总线，不是安全功能。以正常方式启动程序直接显示主界面，不问密码。

- 出厂默认密码 `123456`，在「工具 → 界面密码」里改，至少 4 位。
- 同一个窗口里有「启用界面密码」勾选。去掉这个勾，从托盘打开主界面和退出并停止都不问密码；只切这个勾时新密码两栏留空。
- 改密码、切勾、「恢复出厂默认」三样都要先输当前密码。

> **说明** **忘了密码。**删掉 `%LOCALAPPDATA%\SoftSC\SystemManager\ui-password.json`，密码与勾选一起回到出厂默认（密码 `123456`、勾选为启用）。

3.9 按钮速查：主窗口

下图标出主窗口各区域，编号与下表对应。

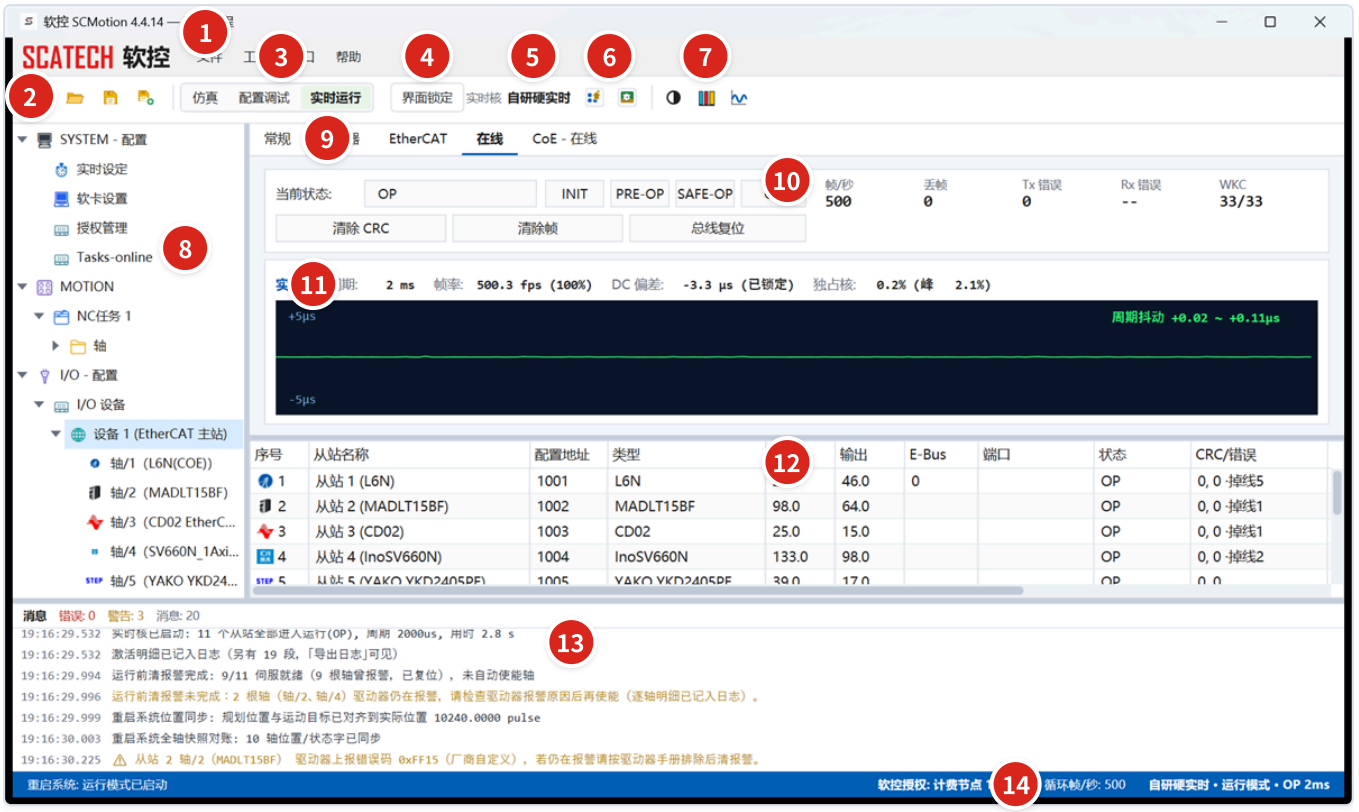


图 3-1　主窗口（实时运行，选中 EtherCAT 主站「在线」页）

序号	名称	作用
1	**菜单栏**	「文件」新建 / 打开 / 保存 / 另存为 / 修改项目名称；「工具」ESI 设备库、示波器、CAD 绘图、SCRemote 远程桌面、界面密码；「窗口」；「帮助」打开帮助中心各分组。
2	**项目按钮组**	从左到右：新建项目、打开项目、保存项目、另存为项目。
3	**模式条**	「仿真」「配置调试」「实时运行」三段。高亮的那段就是当前模式，点另一段就切换。「实时运行」按上一次激活的配置重启进 OP，不下发新组态。
4	**界面锁定**	任何模式下都能按。锁定后本软件不写 I/O、不改参数、不动总线、不发运动命令，观察与诊断照常；不锁软卡 SDK 客户程序。重启软件即解除。
5	**实时核指示**	显示当前实时核形态（如「自研硬实时」）。
6	**激活配置参数 / 重启系统**	「激活配置参数」把当前工程的组态下发：启动参数、分布时钟、PDO 配置，全部从站进 OP，启动过程数据，改过工程一律用它。「重启系统」与「实时运行」同一个动作，在运行档时也能再重启一次。
7	**主题 / 设备库 / 示波器**	切换深色/浅色主题；打开 ESI 设备库；打开示波器（独立程序）。
8	**工程树**	「SYSTEM - 配置」：实时设定、软卡设置、授权管理、Tasks-online。「MOTION」：NC-task、轴、插补、齿轮、凸轮、CNC、机器人。「I/O - 配置」：EtherCAT 主站、从站，右键主站有「扫描」。
9	**页签**	当前选中对象的页面。主站：常规 / 适配器 / 在线 / CoE - 在线 / EtherCAT；从站：常规 / EtherCAT / 分布式时钟 / 启动参数 / 扩展模块 / 过程数据 / CoE 在线 / 在线 / I/O 测试 / 诊断；轴：常规 / 设置 / 参数 / 动态 / 功能 / 在线。

序号	名称	作用
10	**主站在线页**	当前状态（INIT / PRE-OP / SAFE-OP / OP 一键切换）、帧/秒、丢帧、Tx/Rx 错误、WKC；「清除 CRC」「清除帧」「总线复位」。
11	**实时性能**	周期、帧率、DC 偏差（括号里「已锁定」为正常）、独占核占用与周期抖动曲线。
12	**从站列表**	每台从站的名称、配置地址、类型、输入/输出字节、状态、CRC/错误。状态应全部为 OP。
13	**消息栏**	只显示简要错误（红）和警告（黄）。右键菜单：「导出」完整日志、「清除」、「复制单条」、「复制全部」。软件正常退出会删掉本次日志，要留日志就在运行期导出。
14	**状态栏**	左边是扫描/总线状态提示；右边依次是授权状态、循环帧/秒、当前模式与总线状态。仿真模式下多一个黄色「仿真」标。

3.10 按钮速查：轴「在线」页

工程树 MOTION 下选中一根轴，点「在线」页签。使能、点动、寸动、复位、停止都在这一页；「回原点」「位置定义」两组在同页下方（见第 11 章「轴运动与安全边界」）。

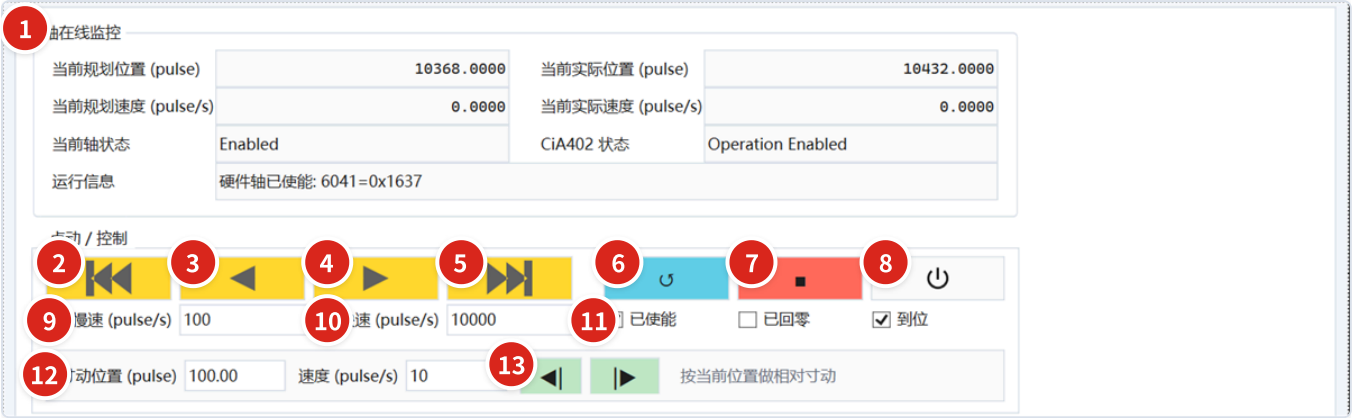


图 3-2　轴「在线」页（已使能状态）

序号	名称	作用
1	**轴在线监控**	当前规划/实际位置、规划/实际速度、当前轴状态、CiA402 状态；「运行信息」写出实际下发的指令和驱动器状态字原始值（如 6041=0x1637）。
2	**快速后退（F1）**	按住以「快速」速度反向运动，松开减速停。
3	**慢退（F2）**	按住以「慢速」速度反向运动，松开即停。第一次调试用它。
4	**慢进（F3）**	按住以「慢速」速度正向运动，松开即停。
5	**快速前进（F4）**	按住以「快速」速度正向运动，松开减速停。
6	**复位 ↺**	故障态时做驱动器故障复位；没故障时把规划位置、目标位置同步到实际位置，防止使能后飞车。SDK 客户程序持有控制权期间不可用。

序号	名称	作用
7	**停止 ■**	按「动态」页的停止减速度减到零。SDK 客户程序持有控制权期间仍可用，是在界面上停止客户程序所发运动的唯一方式。
8	**使能 / 取消使能 ⏻**	按 CiA402 状态机把驱动器推进到 Operation Enabled，或取消使能。
9	**慢速**	F2 / F3 点动的速度。第一次调到默认值的 1/10 左右。
10	**快速**	F1 / F4 点动的速度。
11	**已使能 / 已回零 / 到位**	完成状态指示：使能、回零、定位到位分别勾上。
12	**寸动位置 / 速度**	相对寸动的距离和速度。
13	**寸动后退 ◀| / 寸动前进 |▶**	按当前位置做一次相对寸动；走完平滑、「到位」勾上后再逐步提速。

> **注意** 在线表只在过程数据跑着的时候才是真实状态。四个点动按钮只有在「实时运行」且轴已使能时才可用。

3.11 按钮速查：「检查配置」对话框

工程树里右键 EtherCAT 主站 →「扫描」后弹出，用于把现场硬件与工程拓扑逐台对照。

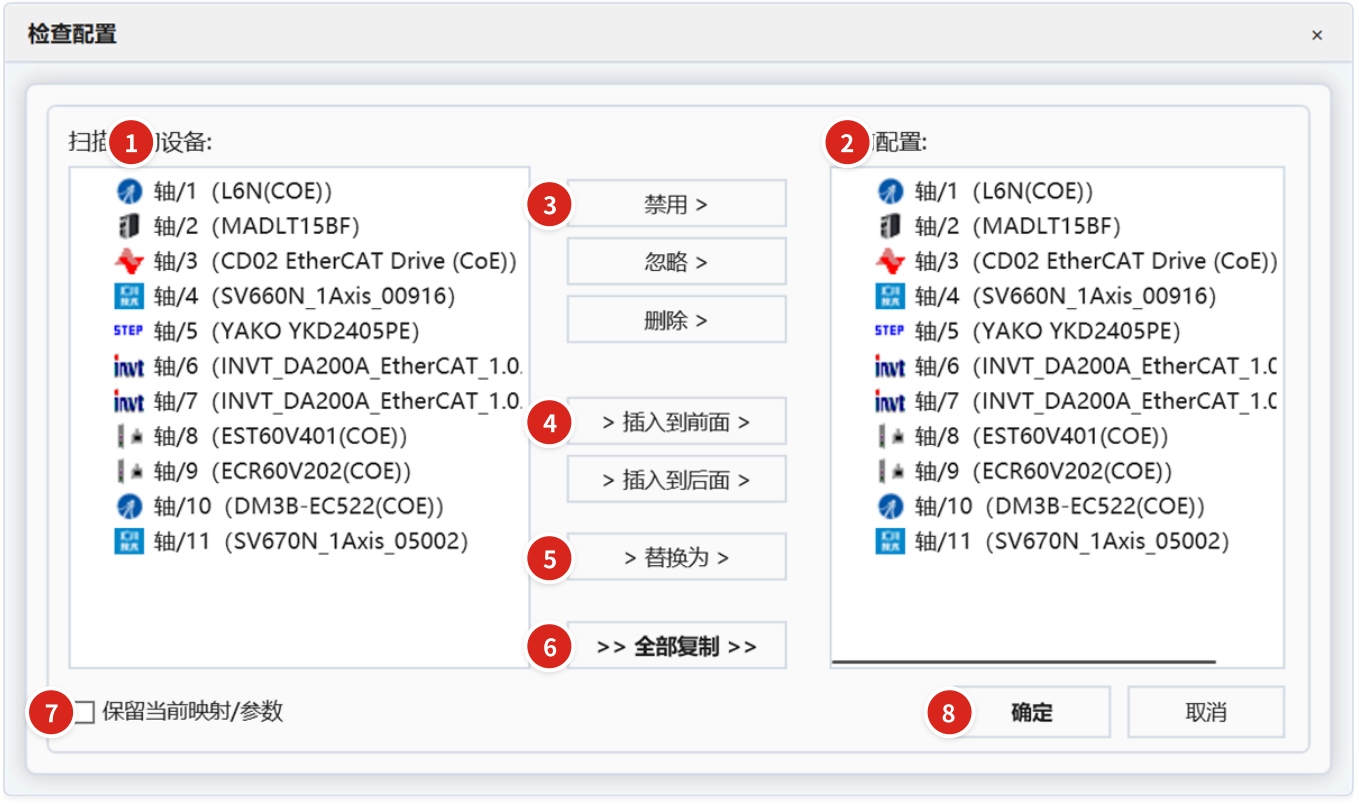


图 3-3 「检查配置」对话框

序号	名称	作用
1	**扫描到的设备**	现场扫出来的从站，按总线顺序排列。
2	**当前配置**	工程里已有的拓扑。第一次建工程是空的。
3	**禁用 / 忽略 / 删除**	处理工程里多出来（现场没有）的站。
4	**插入到前面 / 插入到后面**	逐台把扫描到的设备插入工程。
5	**替换为**	用扫描到的设备替换工程里对应位置的站。
6	**全部复制**	把扫描结果整体复制为工程拓扑，第一次建工程用它。
7	**保留当前映射/参数**	勾上后，替换或复制时保留工程里这一站改过的 PDO 映射和启动参数。
8	**确定 / 取消**	「确定」应用到工程；「取消」保持工程不动，消息栏写明新增 / 移除 / 变更各几台。

3.12 按钮速查：从站「I/O 测试」页

工程树里选中一台 I/O 从站出现此页（驱动器上不显示）。看输入、写输出都要过程数据在跑。

按钮 / 列	作用
「载入点位」	把本站通道刷进表格。每行：点位、方向、类型、字节.位、对象、当前值、写输出、映射名。
当前值徽章	布尔量：● ON（绿）、○ OFF（灰）。数值通道（UINT、INT、REAL）直接显示在线值，点行首「+」展开成各位。
「写1」「写0」	DO 行上直接驱动该点。
值框 +「写入」	AO 行按类型填数（十进制、`0x` 十六进制、REAL 浮点）后写入。
「DO 走位巡检」+「间隔(ms)」	让所有 DO 依次点亮走一遍，「停止」结束。
「全部置位」「全部清零」	一键把本站所有数字量输出拉高或拉低；模拟量、字通道不参与。
「映射名」列	给通道起业务名（如"夹爪到位"），与「过程数据」页共用，随工程保存。

3.13 按钮速查：网卡管理与实时设定

网卡管理窗口入口：主站「适配器」页右边的「网卡管理...」。

按钮	作用
「更新列表」	重新枚举网卡、读驱动状态。
「安装自研驱动」	给选中的物理网卡安装实时驱动（需管理员权限）。

「接管此卡（启用实时）」	由实时驱动接管选中网卡，这张卡从 Windows 网络里消失。
「释放此卡（还给 Windows）」	反向操作，网卡恢复给 Windows 使用。

实时设定页入口：工程树「SYSTEM - 配置」→「实时设定」。

按钮 / 勾选	作用
「实时形态（档1 独占核 / 档2 共享核）」	查看和切换实时形态，切换后要重启。
「占用实时核」「释放实时核」	档1 摘核 / 还核，重启后回到本页看形态那一行：写着"已生效"才是真的接管到核。
「新机器一键配置」	检查 Secure Boot、实时驱动与虚拟化，汇总检查结果。
「扫描后自动进入 FreeRun / OP（不再询问）」	扫描完不再询问是否进 FreeRun。
「软件随系统开机自动运行」「开机自动进入运行(OP)」	开机自启与开机自动进 OP，见第 5 章。

第4章 真机第一轴

从一台新电脑走到真实伺服电机低速转起来。手上要有：一台 Windows 电脑、一台 EtherCAT 伺服驱动器和电机、一根网线。模式条和三个对话框先看第2章「快速入门」。这是让轴动起来的完整步骤。

4.1 开始前

> **注意** **第一次让真实电机动之前。**电机与机械负载脱开，或负载能自由运动；急停在手边；速度先用最小值。仿真里做错没有后果，真机上没有撤销。

4.2 第 1 步：装驱动、接管网卡

总线由自研实时驱动接管一张认证网卡（Intel I210 / I219 系）来跑。型号清单和不支持的网卡见第6章「实时形态与性能」。装驱动和接管都在网卡管理窗口里做，入口是主站「适配器」页右边的「网卡管理...」。打开后先点「更新列表」读驱动状态，列表分组：

分组	含义
已绑定自研实时驱动	可以直接跑 EtherCAT 的网卡。后面方括号里写着它有没有被接管。
可安装自研实时驱动的以太网卡	物理网卡，还没装驱动。选中后点「安装自研驱动」。
不可用于绑定的网卡	平时不显示。只有软件没认出的物理网卡才会落到这一组。

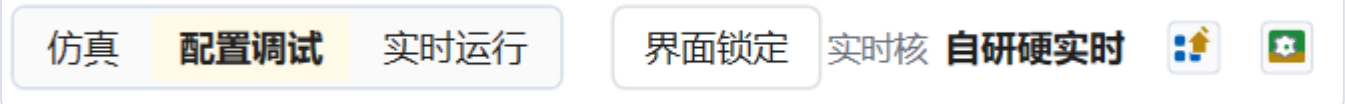
上方「实时网卡接管」区管的是选中的那一张：在列表里选中目标网卡，点「接管此卡（启用实时）」；「释放此卡（还给 Windows）」反过来。被接管后这张卡从 Windows 网络里消失，释放即恢复。选好后点确定，主站「适配器」页的「描述」就显示这张卡。

安装/卸载驱动要管理员权限。同一张网卡同一时间只能归一种驱动，不要去设备管理器手工换驱动，在这个窗口里装和卸就行；卸载只删本软件自己的驱动，不动厂商驱动。

4.3 第 2 步：接线、新建工程

- 网线从电脑网口直连驱动器的 EtherCAT IN 口，驱动器上电。
- 「文件 → 新建项目」，点模式条上的「配置调试」。「进入配置调试模式」点确定，「进入 FreeRun / OP」选否。新工程没有拓扑，「加载工程 I/O 拓扑」那个框不会出现。

应该看到：工程树显示新工程，模式条停在「配置调试」。



工具区模式条：仿真 / 配置调试 / 实时运行，右侧为「界面锁定」开关与实时核指示

4.4 第 3 步：扫描总线

- 工程树里右键 EtherCAT 主站节点 → 「扫描」。

应该看到：弹出「检查配置」对话框。左边「扫描到的设备」是现场扫出来的，右边「当前配置」是工程里已有的。第一次建工程右边是空的，点「>> 全部复制 >>」把左边整条链接过去，再点确定。从站列表随后出现你的驱动器，名称来自驱动器自己存的信息，状态为 PRE-OP 或 INIT。



检查配置对话框：左侧扫描到的设备与右侧当前配置逐台对照，中间为禁用/忽略/删除/插入/替换/全部复制按钮

工程里已有拓扑时，这个对话框就是现场硬件与工程配置的逐台对照表。两边不一致的位置用中间的按钮逐台处理（「替换为」「插入到前面」「插入到后面」「删除」）；只想看差异就点取消，消息栏会写明新增 / 移除 / 变更各几台，工程不动。勾上「保留当前映射/参数」，替换从站时能保住已配好的轴映射与启动参数。

扫完软件会问要不要进 FreeRun，这次选否。「SYSTEM → 实时设定」里勾上「扫描后自动进入 FreeRun / OP（不再询问）」以后就不问了。

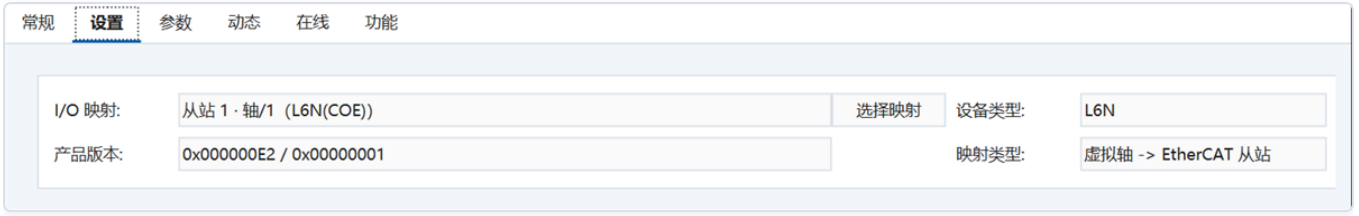
> **说明** **扫不到。**换网线，确认插的是 IN 口；到「网卡管理...」确认这张卡已装驱动并被接管；驱动器刚上电要几秒才就绪，等 5 秒再扫；还不行查第23章「故障排查」。

4.5 第 4 步：建轴、映射从站

拓扑写入工程后，识别为驱动器的从站会弹「识别到伺服从站」，点确定就在 NC-task 下建好轴并映射。要自己建：工程树 MOTION 下右键 NC-task → 「添加轴」，打开轴「设置」页点「选择映射」，选那台驱动器。轴的当量（每个用户单位多少脉冲）先用默认值也行，它只影响显示换算。

应该看到：「设置」页的「I/O 映射」栏从「未映射 / 仿真」变成 `从站 N · 轴/N（型号）`，「映射类型」显示 `虚拟轴 -> EtherCAT 从站`，「设备类型」显示驱动器型号。

这一栏是"这根轴到底在指挥哪台驱动器"的唯一依据。多轴时轴号与站号未必相同，排查动错轴先看这里，不要按工程树顺序猜。

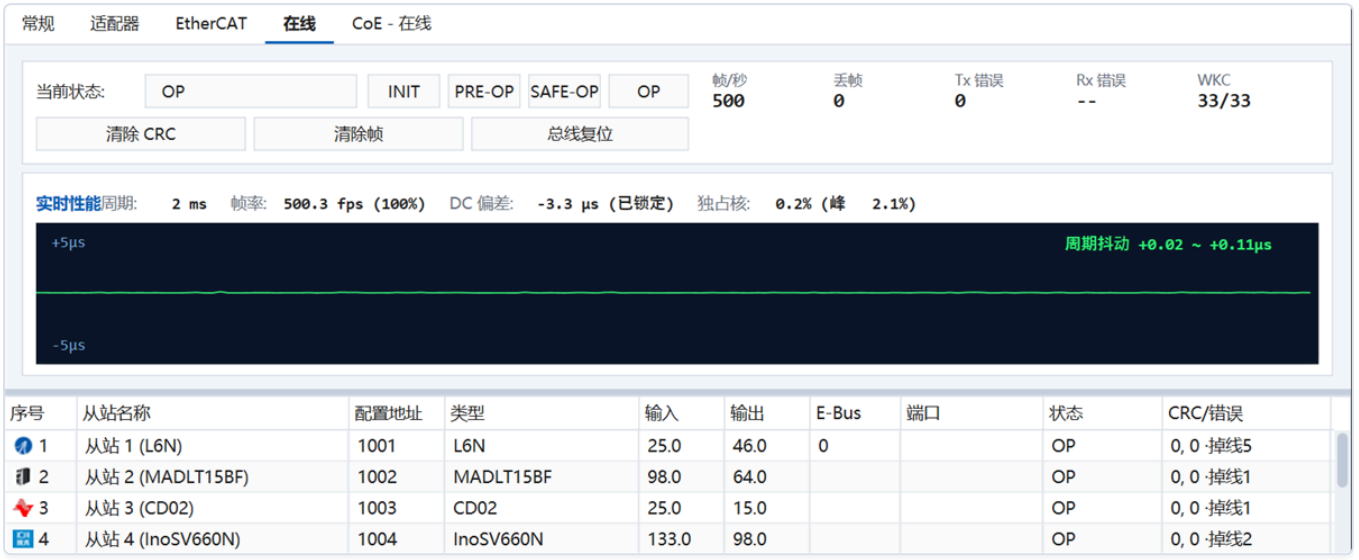


轴「设置」页：I/O 映射指向具体从站，映射类型是虚拟轴到 EtherCAT 从站

4.6 第 5 步：激活、进 OP

- 点工具栏的「激活配置参数」。软件自动下发启动参数、配置分布时钟、把从站带到 OP、启动过程数据，完成后模式条停在「实时运行」。以后工程没改、只是重新起总线，点「实时运行」就行。

应该看到：从站状态全部 OP。打开主站「在线」页，DC 偏差括号里在一分钟内变成「已锁定」（锁定前也能继续操作）。



主站「在线」页：当前状态 OP、帧率与 WKC、实时性能曲线，下方从站表逐台显示状态

这一页盯四个数：

读数	正常的样子
当前状态	`OP`。停在 SAFE-OP 说明输出段没被从站接受，查 AL 码。
WKC	实到 / 期望相等，比如 27/27。少一个就是有从站没应答。启动那几秒 WKC 显示 0 是正常的：启动阶段输出段先空转，所有从站确认且时钟收敛后才转入正常读写。
丢帧 / Tx 错误 / Rx 错误	全 0，长期不涨。偶尔加 1 后就停住是瞬态。
DC 偏差	括号里「已锁定」。伺服要走分布时钟同步（1C32:01 = 2）。某台读回不是 DC 时，日志里会写明是哪台从站，软件不阻止使能；出现能使能但不跟随、跟随异常时，先在消息栏右键「导出」导出日志找这一行。

卡在某个状态进不了 OP，记下界面提示的 AL 状态码（如 0x001D），到「故障排查」查处置。多数情况重新扫描或给驱动器断电复位就好。

4.7 第 6 步：使能

打开轴「在线」页。使能前先看「轴在线监控」里的「运行信息」和「CiA402 状态」：显示故障（Fault）就先点「点动 / 控制」组里的「复位」。复位同时会把规划位置对齐到实际位置，防止使能后飞车。然后点同一组最右边的使能按钮。

应该看到：「CiA402 状态」推进到 `运行使能（Operation Enabled）`，电机抱闸打开（能听到咔哒一声）。



轴在线页使能成功：当前轴状态为已使能、CiA402 状态为运行使能（Operation Enabled）、运行信息写着 6041=0x1637、已使能勾上、点动按钮变为可用

使能成功须同时满足以下三项：「当前轴状态」= `已使能`；「CiA402 状态」= `运行使能（Operation Enabled）`；「已使能」被勾上，四个点动按钮由灰变为可用。「运行信息」那一行会写出驱动器状态字的原始值（如 `6041=0x1637`），拿它去「故障排查」的 6041 表里对。

使能被拒时「运行信息」一行会写明原因（从站还没真正进 OP、故障复位还在进行、实际位置反馈还没读到等），按第23章「故障排查」的使能失败条目逐条对。客户程序用 `SCM_AxisPower` 使能：返回 0 只表示收下了，完成看 `SCM_AxisGetStatus` 的 `powered` / `faulted`；轴没使能就发 `SCM_AxisMove*` 会被直接拒绝。见第21章「软卡 SDK 上手」。

4.8 第 7 步：低速点动

- 把「慢速」调到较小值（如默认值的 1/10）。
- 按住「慢退」或「慢进」按钮（键盘 F2 / F3），松开即停。「快退」「快进」（F1 / F4）这次先不用。

应该看到：电机缓慢转动，实际位置在变，松开即减速停止。

按住期间四个读数：「当前轴状态」= `运动中`；「当前规划速度」= 你设的慢速值；「当前规划位置」和「当前实际位置」同步爬升，差值稳定在几个脉冲（这是跟随误差，反馈滞后一拍是正常的）。「运行信息」写出实际下发的指令。开环步进驱动器没有速度反馈，「当前实际速度」一直显示 0，但「当前实际位置」照样跟随，不代表轴没动。

> **注意** **规划位置在变、实际位置纹丝不动。**立刻松手。这通常是从站没有真正运行在分布时钟下，继续加速只会让指令位置和真实位置越差越远。先导出日志找 DC 那一行，再查第23章「故障排查」里"使能成功、指令下发了，电机不动"的条目。

点动方向对了，再做一次小距离定位：「寸动位置」「速度」填小值，点「寸动后退」「寸动前进」各一次，看走完平滑、「到位」勾上、驱动器不报警，再逐步提速。

4.9 顺手验一个 I/O 点

机器上有 I/O 模块的话，在使能之前就可以验：

- 选一个普通数字输出点，在从站的「I/O 测试」页点「写1」「写0」，或在「过程数据」页右键「在线写入 TRUE」「在线写入 FALSE」。要在 FreeRun 或 OP 下写，输出灯要跟着变。
- 再选一个输入点，手动触发，看「I/O 测试」页的徽章和「过程数据」页的在线值同时变。
- 在线值变了、实际 I/O 不变：先看 PDO 方向、过程数据在不在跑、从站是不是 OP。

4.10 回零

轴「在线」页的「回原点」组显示当前回零配置，「启动回原点」执行；参数在「动态」页改。回零方式、原点开关接法和各模式图解见第12章「回零」。回零参数（如 6099:1 寻找速度）被驱动器拒收时，「驱动器回原点」按驱动器内部存的值走，不影响进入运行；有的驱动器对回零速度有下限，填得太小会被拒。

到这里第一轴完成。接下来可以在轴「功能」页做定位/往复验证，用软卡 SDK 让自己的程序驱动这根轴（「软卡 SDK 上手」），或上插补 / CNC / 机器人。多台驱动器混在一条链上，先读下一节。

4.11 从一根轴到整条链

一条链上同时挂开环步进、闭环步进和伺服是最常见的机器形态。三类执行器对启动参数、分布时钟和使能时序的要求各不相同，下面是混挂时要注意的几件事。授权按计费节点计：计费节点 = 总从站数 − 自有品牌 I/O 模块数。

事项	怎么做
周期取最慢的那一类	周期全链共用，不能一台一个值。先查每种驱动器支持的 60C2 插补周期，取交集里的最大值起步（混挂链用 2 ms 起步稳）。跑稳之后再往下压 1 ms / 500 µs / 250 µs，每压一档重新看丢帧和 DC 抖动。
伺服走分布时钟	伺服在非 DC 时基下也能使能成功，但位置指令不会被真正跟随，这是最难查的一类故障。激活后某台读回不是 DC，日志里会写明站号，软件不阻止使能。看到使能成功但轴不跟随时，先去查那台的日志和 DC 配置。
有的步进驱动器对启动参数敏感	要收到完整的一套启动参数（PDO 内容、指派、运行模式）才进 CSP，缺一项就停在 SM：使能正常、一步不走。改完配置一定点「激活配置参数」重新下发，不要只按「实时运行」。

说明文件要对得上型号	同系列不同固件声明的 PDO 尺寸可能不同，套错说明文件会把同步管理器长度配错，表现为进不了 OP（AL 0x001D / 0x001E）。没有说明文件的设备也能识别和控制，按设备自己存的配置走；要用厂家说明文件在「工具 → ESI 设备库」导入，导入后以说明文件为准。
先分段调通，再整链上电	几十台一起上电时出了问题，难以判断是哪一类设备引起的。按家族分段：先只接开环步进调通，再加闭环步进，再加伺服。每加一段重新扫描并激活，WKC 满值再往下走。

看哪里	开环步进	闭环步进	伺服
当前实际位置	等于规划位置，没有独立反馈	编码器反馈值，与规划位置有跟随误差	编码器反馈值，跟随误差最小
丢步 / 堵转	软件看不出来，位置照走、机械已停。要靠行程开关或外部检测	驱动器能报位置偏差过大	驱动器报过载/位置偏差，603F 有故障码
使能后	相电流保持，轴被锁住	同左，另有闭环校正	闭环锁轴，抱闸打开有咔哒声
同步方式	按驱动器要求配置	按驱动器要求配置	分布时钟
调试起步速度	更保守，失步是不可恢复的位置丢失	可稍快	可稍快

> 注意 **混挂链上第一次点动。**把每一根轴的点动速度都调到小值，确认所有电机都与负载脱开或负载可自由运动。一次只动一根轴，逐根确认转向与量级，再谈联动。

现象	处置
扫到的台数少于实际接线	从最后一台能扫到的从站往下找，多半是它的 OUT 口到下一台 IN 口这段网线或接口的问题。EtherCAT 有方向，IN 与 OUT 不能接反。
某一类从站集体进不了 OP，其它类正常	是这一类的说明文件或启动参数问题，不是链路。查它的说明文件是否匹配、工程里有没有为它单独写入的旧启动参数。
台数越多越容易丢帧	从站越多单帧越长，周期余量越小。看主站「在线」页的「实时性能」。余量不够就放大周期，或改用档1 独占核，见第6章「实时形态与性能」。
使能成功，规划位置在走，实际位置不动	从站没有真正运行在分布时钟下。见上表"伺服走分布时钟"和「故障排查」里的同名条目。
轴号和站号对不上，动错了轴	轴「设置」页的「I/O 映射」写的是这根轴实际绑定的从站，以它为准。

第5章 工程、模式与总线

工程文件是什么、三种模式各能做什么、「激活配置参数」和「实时运行」的区别，以及开机自启怎么设。

5.1 工程与项目名称

- 项目名称就是工程应用名，用在窗口标题、工程文件和开机自启识别上。「文件 → 修改项目名称」改，别长期停在默认名。
- 新建工程后先保存一次，再配 I/O、轴和运动。
- 导出的日志里带项目名，现场排障靠它定位是哪个工程。

5.2 三种模式

目标系统三档，在工具栏的三段模式条上切。软件打开后停在「配置调试」。

模式	用途	能做	不能做
仿真	离线。不枚举、不占用网卡。切换后自动启动仿真用的实时核和软卡 SDK 服务，是完整的离线运行时。	建工程、配轴；轴运动、组插补、CNC、齿轮/凸轮、机器人都能跑；SDK 客户程序可以直接连。	没有真实回零开关；齿轮/凸轮啮合瞬时完成；I/O 是内存里的位。
配置调试	组态与联调。进入时要过三个对话框（见第2章「快速入门」），第三个框选否是只连总线不跑过程数据，选是进 FreeRun。	扫描、改 PDO 分配、启动参数、CoE 在线读写、简单 I/O、诊断。	真实轴运动。SDK 的命令（建总线、轴命令、写输出）会被拒绝；状态、诊断、过程映像读取不受限。PDO 映射和启动参数只能在过程数据停着的时候改。
实时运行	真机 OP。	真实轴的点动、寸动、定位、回零；写 I/O；CoE 在线读写；SDK 全部命令。	改 PDO 映射、启动参数这类组态。改了要回「配置调试」，再「激活配置参数」。

观察不需要专门的模式：任何模式下都能看 PDO 在线值、状态、报警和诊断。要只看不碰，按「界面锁定」（见「界面总览」）。

5.3 模式条与两个动作

按钮	做什么
「仿真」	切到仿真。过程数据在跑时会先确认再停。
「配置调试」	停过程数据，总线回到可配置状态，然后过三个对话框。
「实时运行」	按上一次激活的配置重启进 OP，进入前清除 EtherCAT 和伺服报警。工程改了它不下发新组态。
「激活配置参数」	把当前工程的组态下发下去：启动参数、分布时钟、PDO 配置，全部从站进 OP，启动过程数据。改过工程一律用它。

「重启系统」	与「实时运行」同一个动作，在运行档时也能再重启一次。

激活成功后，软件会把软卡 SDK 的连接配置写到 `C:\ProgramData\SoftSC\softcard.runtime.json`，客户程序拿到的轴数、I/O 点数以激活那一刻为准。工程改了没重新激活，SDK 看到的还是旧的。

离线工程不能直接进 OP：先选实时网卡、扫描、把在线拓扑写入工程，才能激活。现场硬件与工程拓扑不一致时软件会停下并提示差异。引擎无法启动时消息栏会简要说明原因，完整启动明细在导出的日志里「启动诊断全文」那一行。

实时驱动只有一种后端，两档形态，见第6章「实时形态与性能」。

> **说明** **升级用完整安装包。**主软件与内核驱动之间的版本必须一致，不一致时软件会拒绝启动引擎并把两侧版本写进日志。不要单独替换驱动文件，手工拷驱动会让系统悄悄沿用旧驱动，症状是"装了新版但问题依旧"。

5.4 开机自启动

在工程树「SYSTEM - 配置」页有两个勾选（界面见图 2-1）：

- 「软件随系统开机自动运行」：开机后自动拉起软件，藏到托盘。旁边显示当前有没有启用。
- 「开机自动进入运行(OP)」：在上一条基础上，开机后加载上次工程、不扫描直接进 OP、起软卡运行时，SDK 客户程序可以直接连上驱动真机，不用人工点激活。勾它会自动把上一条一起打开；关掉上一条时它也跟着复位。

> **注意** **开机自动进 OP 等于上电就带真实硬件运行。**现场不希望这样，就只留「软件随系统开机自动运行」，或两条都关，改为人工激活。

> **说明** **自启写的是当前可执行文件的路径。**换目录或重装后重新勾一次。被安全软件拦下时软件会提示，那种情况重启是不会生效的。

第6章 实时形态与性能

实时驱动有几种形态、哪些网卡能用，以及"我的设备能跑多小的周期、能带多少轴"怎么自己算、自己测。

6.1 一种后端，两档形态

总线只有一种后端：自研实时驱动接管一张认证网卡，周期收发在内核里完成。它有两档形态，在「SYSTEM → 实时设定」页的「实时形态（档1 独占核 / 档2 共享核）」里查看和切换，切换后要重启一次。



实时设定页：基础周期、运动周期、绑核与最坏抖动，下面是总线质量校验与两档实时形态

	档1 独占核	档2 共享核
核	从 Windows 摘走一个物理核给引擎独占，机器至少 4 核	不摘核，引擎以内核线程与 Windows 共用一个核
承诺周期	250 µs	2 ms
虚拟化	不能开（Hyper-V / 虚拟机平台 / 内存完整性），开着档1 起不来	不受影响
适合	1 ms 以下的伺服插补、电子凸轮	点位、I/O、扫描调试；只有 2 核的机器、虚拟机、不允许改启动项的受管电脑

怎么切。「占用实时核」「释放实时核」，重启后回到同一页看形态那一行：写着"已生效"才是真的接管到核；写着"回落"是配置了但没接管成功，引擎按档2 在跑。拿不准这台电脑能不能用档1，先点「新机器一键配置」。

新机器一键配置。「新机器一键配置」检查这台电脑能不能跑实时驱动：Secure Boot 要关，实时驱动要在运行；用档1还要求没开虚拟化（Hyper-V / 虚拟机平台 / 内存完整性）。结果写进运行日志并弹窗汇总。

档2 上怎么选周期。2 ms 及以上直接用。1 ms 跑得动但余量薄：样机在 1 ms 下成功率 100%、零丢帧零掉拍，可最坏抖动的高水位 0.74 ms 占掉一个周期的 74%，而 2 ms 下同量级的 0.4 ms 只占 20%。档2 的抖动是偶发高水位，周期越短余量掉得越快，再短一档（500 µs）样机就开始掉拍。要在自己的机器上确认，每档稳定跑 5 分钟以上，再按下面的指标核对。

6.2 认证网卡

实时驱动要独占网卡硬件做微秒级确定性收发，只支持 Intel 的两个架构族。装驱动和接管的操作见第4章「真机第一轴」第 1 步。

型号	说明
Intel I210（铜缆）	推荐首选。台式机/工控机加装 I210-T1 PCIe 网卡即可，多数工控主板的板载 EtherCAT 口也是它。
Intel I210 其它形态	光纤 / SerDes / SGMII / Flashless 变体，同一控制器。
Intel I211	消费级主板常见的板载千兆口，与 I210 同一驱动架构。
Intel I350	服务器双口/四口网卡，与 I210 同一驱动架构。
Intel I219	推荐。笔记本/工控机常见的板载口，控制器集成在主板芯片组里，是另一套硬件层。
Intel I217 / I218	I219 的前代，同一驱动架构，多见于早些年的工控机板载口。

> **说明** **I219-LM 与主板管理引擎（Intel AMT）共用网口。**BIOS 里 AMT 启用时驱动会拒绝接管，先在 BIOS 关掉 AMT。I217 / I218 同样。

不支持：Intel I225 / I226（2.5G，另一种寄存器模型）、Realtek 等非 Intel 网卡。这些机器只能用仿真模式把流程走通。

6.3 周期与轴数由三层决定

能跑多小的周期、这个周期下能挂多少轴，不是一个固定数字。三层约束互相独立，任何一层不满足，另外两层再好也没用。

层	决定什么	关键参数	能不能改
总线物理	一个周期内数据在网线上跑一个来回最少要多久	从站数量、每站过程数据字节数	能：减少站数，或裁剪 PDO 映射
主站确定性	控制器能不能每个周期都准时把帧发出去	实时形态、CPU 与主板设置、系统负载	能：换档1、调机器设置、减少界面负载
从站固件	驱动器自己支不支持这个同步周期	驱动器型号与固件版本	不能：选型时避开，或找厂商要支持该周期的固件

第三层最容易被忽略。样机上遇到过：同一根总线、同一个周期，某型号驱动器的某一批固件在 250 µs 下报故障（状态字 0x5718），同型号的另一版固件和同链路上其它三个品牌的驱动器全部正常。这类问题在控制器侧无解，加 CPU、缩短链路都不会好转。选型阶段就向驱动器厂商确认目标周期的支持状态。

6.4 总线时间可以直接算

EtherCAT 走 100 Mb/s 固定线速，一个周期里帧从控制器出发、穿过每一台从站到链尾、再原路返回。这段时间和 CPU 无关，纯由拓扑决定：

```
总线往返 ≈（帧字节数 + 20）× 0.08 µs  +  从站数 × 1 µs
其中  帧字节数 ≈ 107 + 每站过程数据字节数 × 从站数
```

- 0.08 µs/字节：100 Mb/s 线速的物理常数。
- +20 字节：以太网前导码与帧间隙的固定开销。
- +107 字节：每帧固定携带的以太网头与若干诊断数据报（时钟同步、链路巡检等），与从站数量无关。
- 1 µs/站：每台从站芯片的转发延迟（去、回各约 0.5 µs），样机链实测标定值。不同厂商的从站芯片有差异，精确评估以实测为准。
- 每站过程数据字节数由 PDO 映射决定，在从站「过程数据」页能看到。参考值：伺服全量映射约 25 字节，步进约 11 字节，只保留循环同步位置控制必需对象时可降到 8 字节左右。混挂时逐站相加。

按全部为伺服、每站 25 字节算出的总线占用（最保守的一档；含步进的混挂链更宽松）：

从站数	帧长	总线往返	占 250 µs	占 500 µs	占 1 ms	占 2 ms
8	307 B	34 µs	14%	7%	3%	1.7%
16	507 B	58 µs	23%	12%	6%	2.9%
24	707 B	82 µs	33%	16%	8%	4.1%
32	907 B	106 µs	42%	21%	11%	5.3%
40	1107 B	129 µs	52%	26%	13%	6.5%
48	1307 B	153 µs	61%	31%	15%	7.7%
55	1482 B	174 µs	70%	35%	17%	8.7%

样机 27 站混挂链实测往返 78 µs，与公式同一量级。两条硬边界：

- 单帧最多约 55 站（每站 25 字节时）。以太网单帧上限 1500 字节，超过就要拆成多帧。步进每站约 11 字节，单帧可到约 126 站，此时先到顶的是逐站转发延迟而不是帧长。
- 总线占用不超过周期的 50%~60%。剩下的余量留给控制器抖动和参数读写（SDO）等非周期通信。占用逼近 100% 时任何一点抖动都直接变成丢帧。

6.5 判断能否稳定运行

选定周期和轴数后，激活并稳定跑 5 分钟以上，再按下面几条对照。全过才算这个组合可用；任何一条不过，要么放大周期、要么减轴数、要么改善控制器环境。运行期的数据在导出的日志里：「运行统计」行每 2 分钟一条，只写进日志文件；相位分布在「启动诊断全文」那一行。在消息栏右键「导出」导出。

指标	合格线	在哪看	不合格说明什么
周期成功率	稳态 ⩾ 99.99%	「运行统计」行的成功百分比，取两条相隔 5 分钟的做差	控制器没能每周期按时收发。这一项是最终依据
最坏抖动占周期	< 10%	「运行统计」的最坏抖动 ÷ 周期	预测力最强的一条。样机实测：占 7.3% 时成功率 99.994%，占 42.8% 时掉到 99.07%，占 166% 时只剩 83%
总线占用	< 周期的 50%~60%	按上面的公式算	余量不足，抖动直接变成丢帧
相位分布	> 90% 落在 ±0.5% 周期内	「启动诊断全文」的相位分布字段	发送时刻散开，同步精度下降
分布时钟锁定偏移	< 周期的 1%	「运行统计」的 DC 锁定字段；主站「在线」页的「DC 偏差」	时钟没锁住，多轴同步有相位差

时钟锁定正常，不代表周期正常。样机实测有这样一组：250 µs 下丢掉了 17% 的周期，DC 锁定仍显示 9 µs（占周期 3.6%，看着很健康）。时钟同步与周期收发是两条独立的链路，判断能不能用要以周期成功率和抖动占比为准。

看成功率要看稳态增量。「运行统计」里的成功率是从启动至今的累计值，包含建立映像、首帧、时钟收敛这段启动瞬态。刚启动就看，会把正常的配置误判成不合格。记下两条相隔 5 分钟的运行统计，用差值算。

6.6 样机实测参考

以下是样机链的实测结果，供选型估算。它是参考不是承诺：换一台控制器、换一批驱动器，结论都可能不同，最终按下一节在自己的设备上测。

形态	从站	周期	实测结果
档1	32 轴混挂（步进 + 伺服四个家族）	250 µs	周期成功率 100%，时钟锁定 0~1 µs，相位分布 > 90% 在 ±0.5% 内
档1	20 站	250 µs	26.6 分钟 639 万拍零丢帧，最坏抖动 9 µs
档2	5 站	2 ms	22 分钟 60 万拍零丢帧零掉拍，DC 锁定 ±5 µs，最坏抖动约 0.4 ms；CPU 满载与磁盘满载下这几个数不变

档2	5 站	1 ms	11 分钟 55 万拍零丢帧零掉拍（含满载段），最坏抖动高水位 0.74 ms，占周期 74%
档2	5 站	500 µs	开始掉拍

抖动是绝对量，不随周期缩短而变小。它来自操作系统的调度、中断与电源管理，与你设的周期无关。所以决定周期下限的不是网络带宽，而是"周期还剩多少余量装得下这个抖动"。想缩短周期，应先压低抖动本身（换档1、改主板设置）。

6.7 怎么测出自己设备的上限

1. 按实际拓扑接好全部从站，扫描并保存工程。
2. 从较宽松的周期开始（如 2 ms），「激活配置参数」进 OP。
3. 稳定跑 5 分钟以上，在消息栏右键「导出」导出日志，取两条「运行统计」，按前面列出的指标核对。采样时间不能短：只跑一分钟拿到的是启动瞬态；抖动也一样，样机同一配置跑 1 分钟测得最坏抖动 22 µs，跑 5 分钟测得 146 µs，短采样会把偶发的大抖动整个漏掉。
4. 合格就把周期减半（2 ms → 1 ms → 500 µs → 250 µs），重复第 3 步。
5. 任何一项指标不合格时，上一档就是这套设备的可用周期。实际投产再留一档余量。

测试期间不要使能任何轴，只跑过程数据就能测出控制器的定时能力，没有运动风险。关掉示波器、在线刷新这类高负载界面操作，它们会明显加大抖动。软件正常退出会删掉本次运行的日志文件，导出要在退出前做。

6.8 余量不够时可以做什么

手段	作用	代价
换档1 独占核	引擎独占一个物理核，抖动从几百微秒降到个位数微秒	需要重启；占用一个物理核；机器至少 4 核
主板/电源设置	关闭深度节能（C-state）与频率浮动，避免唤醒延迟造成的偶发大抖动	整机功耗上升
裁剪 PDO 映射	只保留必需对象，每站字节数可从约 25 降到约 8，同周期可挂的轴数成倍增加	裁掉的对象运行中不能读写，要确认工艺上确实不用
减少界面与诊断负载	示波器、在线刷新、抓包都会抢 CPU	无，生产运行时本来就不应开着这些
拆成多条总线	轴数超过单帧上限时用多块网卡分担	需要多块认证网卡

PART 02 · 第二篇

EtherCAT 与 I/O

主站与从站组态、PDO 过程数据、CoE 参数读写、I/O 点检。

EtherCAT 与从站 · PDO 与过程数据 · CoE 在线与启动参数 · 从站 I/O 功能测试

第7章 EtherCAT 与从站

工程树里的 EtherCAT 主站和从站怎么组态、怎么扫描上线、上线不顺先查什么，以及没有说明文件的设备怎么被识别和控制。

7.1 离线组态与真实总线

- 「仿真」里，以及「配置调试」还没连总线时，工程树上的主站、从站和 PDO 映射只是工程文件的内容。从站状态、在线值都不是真实总线的状态。
- 手动添加的从站用来提前把工程配好，不会自己产生在线值。
- 连真实总线要进「配置调试」并连过三个对话框，见第5章「工程、模式与总线」。第三个对话框「进入 FreeRun / OP」选「否」只连总线、不跑过程数据，这时可以改组态；选「是」进 FreeRun，能看 I/O 和实时数据。
- 过程数据跑起来后（FreeRun，或「实时运行」的 OP），过程数据页的在线值才和真实 I/O 同步。

7.2 选网卡

选中工程树里的 EtherCAT 主站，在「适配器」页选实时网卡。列表里没有要用的卡，点「网卡管理...」，在窗口里点「更新列表」重新枚举。认证网卡是 Intel I210 / I211 / I350 / I217 / I218 / I219 系。

网卡被别的软件占着时扫不到从站。先关掉占用网卡的工具，再点「更新列表」。

7.3 扫描与拓扑对比

在「配置调试」下，工程树里的 EtherCAT 主站节点右键「扫描」。扫描结果和工程里的拓扑逐台对照，弹出「检查配置」窗口：左边是扫描到的设备，右边是当前配置。



检查配置窗口：左侧扫描到的设备与右侧当前配置逐台对照，中间是禁用、忽略、删除、插入、替换、全部复制按钮

- 「>> 全部复制 >>」把扫描结果整体复制为工程拓扑，第一次建工程用它。
- 「> 插入到前面 >」「> 插入到后面 >」「> 替换为 >」逐台改；「禁用 >」「忽略 >」「删除 >」处理工程里多出来的站。
- 勾上「保留当前映射/参数」，替换或复制时保留工程里这一站改过的 PDO 映射和启动参数。
- 「确定」应用到工程，「取消」保持工程不动。

扫到的产品编号和说明文件写的对不上（同一台设备两个数），可以登记别名，见本章「没有说明文件的设备」。

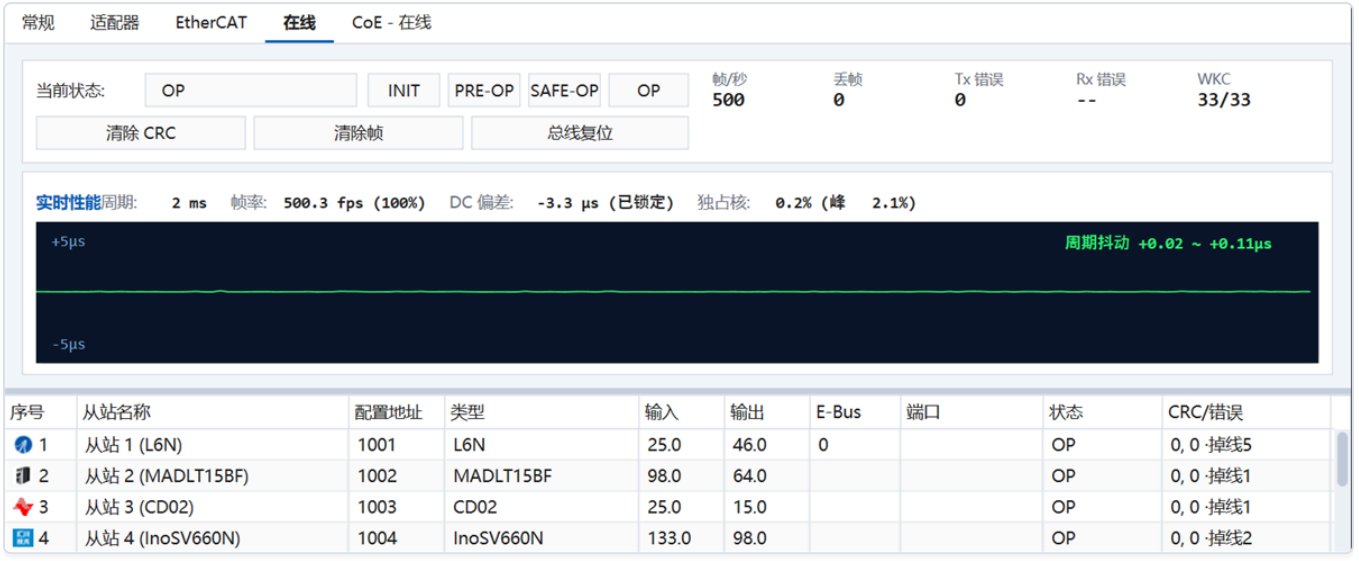
7.4 上线检查表

检查项	看什么	不对先查什么
网卡	主站「适配器」页：网卡名称、驱动模式、有没有被占用。	关掉占用网卡的其它工具，「网卡管理...」里点「更新列表」。
拓扑	「检查配置」窗口：在线从站数量、顺序、厂商号、产品编号与工程一致。	网线进出口、供电、工程拓扑、说明文件有没有导入；产品编号对不上登记别名。
过程映像	从站「常规」页的输入、输出字节数；「过程数据」页的分配和方向。	PDO 有没有分配、对象位长对不对、「PDO 启动参数一致」有没有打勾。见第8章「PDO 与过程数据」。
状态机	激活后从站从 PRE-OP、SAFE-OP 到 OP 的日志。	启动参数写入失败、同步模式、DC 支持、从站错误码。日志会写明是哪一站、哪一条。

周期	主站「在线」页：帧/秒、丢帧、Tx 错误、Rx 错误、WKC。	先关示波器再看。WKC 对不上是有站掉线；丢帧持续上涨看从站表的「CRC/错误」列，哪个口的计数在涨，坏线就在那个口的上游一跳。

7.5 主站页面

- 「EtherCAT」页：NetId、状态、周期、帧率、过程数据大小、拓扑摘要；「拓扑」按钮打开拓扑窗口。下方逐帧列出命令、地址、长度、WC、同步单元、周期、占用率，WKC 对不上的帧标红。
- 「在线」页：当前状态和 INIT / PRE-OP / SAFE-OP / OP 四个按钮切主站状态；「清除 CRC」「清除帧」清计数，「总线复位」重建总线；右侧帧/秒、丢帧、Tx 错误、Rx 错误、WKC；下方是实时性能曲线，标题行有周期、帧率、DC 偏差。
- 「CoE - 在线」页：主站自己的对象字典，只读，可筛选、复制。
- 底部从站表：序号、名称、配置地址、类型、输入、输出、E-Bus、端口、状态、CRC/错误、ESI、厂家、产品。「CRC/错误」是从站芯片的错误计数，格式是端口 0、端口 1 的接收错误，读不到显示「?」。



主站「在线」页与底部从站表

7.6 从站页面

选中工程树里的一台从站，右侧页签如下。「过程数据」「CoE 在线」「启动参数」「I/O 测试」各有自己的章。



从站「常规」页：名称、类型、地址、产品与说明文件，右侧编号、状态、类别与映像，下方四张摘要卡

页签	内容
「常规」	名称（可改）、编号、类型、状态、地址、上级、产品、类别（轴/驱动、I/O、设备）、ESI（同一设备有多份说明文件时在这里选用哪份）、映像大小。
「EtherCAT」	产品/修订、自动地址、上级端口、拓扑位置、输入输出映像、PDO、同步方式；「厂商复位」是故障复位前要先写的厂商专用步骤，留空按出厂预置。
「分布式时钟」	「同步模式」「布局依据」两项设置，以及过程数据状态、当前周期、帧率、「布局一致性」。见本章「同步模式与布局依据」。
「启动参数」	激活时按顺序写进从站的 SDO 列表。见第9章「CoE 在线与启动参数」。
「扩展模块」	带扩展槽位的机架设备用：逐位记录现场插了哪些模块，看整机数据量。只记录和计算，不据此下发过程数据，设备按自己的实际配置交换。
「在线」	当前状态、请求状态与 INIT / PRE-OP / SAFE-OP / OP 按钮；主站状态、同步模式；端口 A 到 D 的链路；「Bootstrap」「Clear Error」；FoE 文件「Download...」「Upload...」；EEPROM「读取备份...」「烧写...」。

> **注意** **烧写 EEPROM 会改设备身份。**「烧写...」把 SII 镜像或说明文件写进从站 EEPROM，要先停过程数据；软件写前自动备份、写后回读校验。动手前先用「读取备份...」自己留一份。

7.7 从站状态机

状态	含义	什么时候用
`INIT`	初始化，只有寄存器访问。	停总线、重建配置。
`PRE-OP`	邮箱可用，能读写 CoE。	读对象字典、写启动参数、准备 PDO。
`SAFE-OP`	输入有效，输出不生效。	进 OP 前的过渡。
`OP`	输入输出过程数据正常交换。	真实 I/O 和真实轴运行的状态。
`BOOT`	固件下载用。	普通调试不要切，厂商要求刷固件时才用。

- 状态后面带 ERROR 是从站报了错误位。在从站「在线」页点「Clear Error」确认，再重新请求目标状态。
- 反复出错，查电源、接线、同步模式、PDO 配置；进不了 OP 的具体原因看日志里对应的那一条。

7.8 同步模式与布局依据

每台从站在「分布式时钟」页有这两项设置，随工程保存，改了要重新「激活配置参数」才生效。

同步模式	适用	说明
SM-Synchron	普通 I/O 模块。	随每帧过程数据同步，不用分布时钟。

DC-Synchron	伺服、步进等所有驱动器。	分布时钟同步，要设备支持 DC。Sync0 周期跟随运行周期。

默认值由软件按设备结构给：输出 PDO 里有控制字 `6040`（CiA402 轴结构）且设备支持 DC 的给 DC-Synchron，其余给 SM-Synchron。工程里手选过的，软件不改。

> **注意** **驱动器必须走分布时钟同步。**驱动器的 `1C32:01` 要等于 2。读回不是 DC 时软件只在日志里告警，不阻止使能。SM 下常见的现象是使能成功但轴不跟随目标位置。碰到这种情况先看日志，把同步模式改成 DC-Synchron 再激活。

「布局依据」决定过程数据布局（每个 PDO 在映像里的位置和长度）以谁为准：

- 「自动（推荐）」：软件综合说明文件和设备读回自动决定，固定映射的设备按设备读回，可重映射的驱动器按说明文件。
- 「按设备」：以设备在线报告的为准。
- 「按说明文件」：以说明文件声明的为准。

「布局一致性」显示说明文件声明和设备读回是否对得上。显示冲突又进不了 OP 时，先切「按设备」重新激活试一次，再去核对说明文件。

7.9 常用 CiA402 对象

对象	方向	用途	在本软件里
`0x6040`	RxPDO	控制字，驱动状态机。	使能、复位、停止由轴控制写，页面和客户程序不要周期直写。
`0x6041`	TxPDO	状态字。	轴页和 SDK 的轴状态都从它解出来。
`0x6060` / `0x6061`	RxPDO / TxPDO	运行模式设定 / 显示。	运行时设成 CSP（8），用 `6061` 确认驱动器真进了这个模式。
`0x6064`	TxPDO	实际位置。	轴页实际位置、SDK `actualPosition` 的来源。
`0x607A`	RxPDO	目标位置。	运行中由实时核逐周期写，不要旁路写。
`0x603F`	TxPDO / SDO	错误码。	驱动器报警先读它，对照厂商手册。
`0x60FD`	TxPDO	数字输入，常见原点和限位。	诊断限位、原点信号，运行中放在 PDO 里看。

> **提示** 运行中要持续观察的对象放进 PDO。CoE 是非周期通道，规则见第9章「CoE 在线与启动参数」。

7.10 没有说明文件的设备

设备库里没有这台设备的说明文件（ESI）时，软件读设备自己存的出厂过程数据配置来判断它是什么，照样能识别和控制。挂了说明文件的设备以说明文件为准。

- 出厂输出 PDO 里有控制字 `6040` 的判为驱动器：自动建轴，设备报告支持 DC 时默认走分布时钟同步，可以使能和运动。「常规」页的「类别」显示「轴/驱动」。
- 只有数字量、模拟量通道的判为 I/O：过程数据页和「I/O 测试」页按设备报告的通道列出。
- 两样都判断不出的从站，日志里会列出，过程数据页留空。这种站先把说明文件导入设备库（「工具 → ESI 设备库」），再重新扫描。
- 过程数据页显示设备报告的默认映射。PDO 内容不能增减条目；PDO 分配表上是设备自己的默认分配，工程里还没有这份分配。要改分配，先点「生成启动参数」把当前分配写进工程，再增删或调顺序，改过的分配在下次激活时写进设备。
- 识别结果和映射随工程保存。

> **说明** **SDK 里的 I/O 序号。**没有说明文件的模块被计入后，按全局序号定位的 `SCM_WriteOutput` / `SCM_ReadInput` 会错位。客户程序用按从站和位定位的 `SCM_WriteOutputBit` / `SCM_ReadInputBit`，见第21章「软卡 SDK 上手」。

7.10.1 产品编号别名

有的设备在线报告的产品编号和说明文件里写的不是同一个数，说明文件就匹配不上。在 `%LOCALAPPDATA%\SoftSC\SystemManager\ProductCodeAliases.json` 里登记这两个数互为别名，写法照安装目录 `Assets\ProductCodeAliases.json`：

```
{
  "groups": [
    {
      "vendorId": "0x00000D37",
      "productCodes": [ "0x100F0003", "0x50001000" ],
      "note": "在线报告的产品编号与说明文件里的官方编号不是同一个数"
    }
  ]
}
```

- 同一组里的产品编号在这个厂商号下互为别名。一组至少两个不同的数，厂商号和产品编号都不能是 0。
- 数值写十六进制（`0x` 前缀）或十进制。
- 用户文件和随包文件合并生效。随包那份软件升级时会覆盖，不要改它。
- 写法不对的组整组不生效，日志会说是哪一组。别名只用于匹配说明文件，不改写任何文件。

第8章 PDO 与过程数据

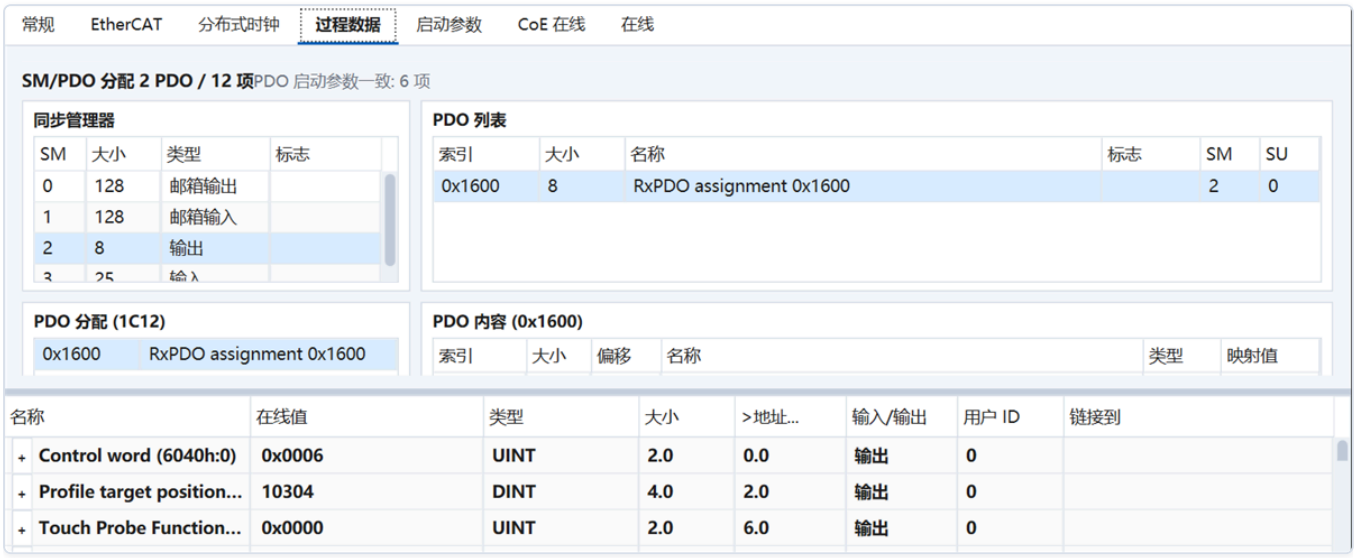
PDO 决定哪些对象参与每个周期的输入输出交换。这章讲「过程数据」页怎么看、映射怎么改、什么时候能改，以及"在线值一直为空""写了没反应"怎么查。

8.1 基本规则

- 输入 PDO（TxPDO，`1A00` 一族，挂在 SM3 / `1C13`）：从站到主站，例如输入点、状态字、实际位置。
- 输出 PDO（RxPDO，`1600` 一族，挂在 SM2 / `1C12`）：主站到从站，例如输出点、控制字、目标位置。
- BOOL 占一个位；WORD 一类整字读写，也能展开看各位。
- 映射在进 OP 前定死。EtherCAT 协议只允许在 PRE-OP 改 PDO 分配和内容，任何主站都一样。
- 要写的对象必须在输出 PDO 里，要读的对象必须在输入 PDO 里。只在对象字典里、不在 PDO 里的对象，运行中只能走 CoE。

8.2 过程数据页怎么看

选中从站，打开「过程数据」页。顶上一行是本站摘要，右侧是「PDO 启动参数一致」状态。下面四张表：



从站「过程数据」页：同步管理器、PDO 列表、PDO 分配与 PDO 内容四张表，下方变量表带在线值

- 「同步管理器」：SM 号、大小、类型、标志。选一个 SM，右边列出它下面的 PDO。
- 「PDO 列表」：索引、大小、名称、标志、SM、SU。
- 「PDO 分配」：分配进 `1C12` / `1C13` 的 PDO，顺序就是映像里的顺序。
- 「PDO 内容」：选中 PDO 里的条目，索引、大小、偏移、名称、类型、映射值。

标志列：`M` = 必选，设备要求它始终在分配里，不能删除；`F` = 固定映射，PDO 内容由设备固定，不能增减条目。空 = 说明文件没写这两个标志，或这台设备没挂说明文件。

8.3 改映射

页面底部一排按钮，「PDO 分配」「PDO 内容」表上右键也有同样的菜单：

按钮	作用
「添加 PDO」	往当前 SM 的分配里加一个说明文件里的 PDO。
「删除」	把选中的 PDO 从分配里去掉。带 `M` 的不能删。删到这个方向一条不剩时会先提示后果。
「上移」「下移」	调分配顺序，顺序就是映像顺序。
「插入条目」	往选中 PDO 的末尾插一个对象。只对说明文件里内容可改、且不带 `F` 的 PDO 有效。
「删除条目」	把「PDO 内容」表里选中的那一条去掉。
「恢复默认」	回到说明文件的默认分配和内容。

「插入条目」打开「插入 PDO 条目」对话框：从说明文件的对象字典里选一个对象，只列方向对得上、有位宽的对象。筛选框按对象号（如 `60FF` 或 `0x60FF:0`）、名称、类型筛，多个词用空格隔开；说明文件声明可映射的对象排在前面，「可映射」列写「未声明」的，设备可能在激活时拒绝。双击一行或点「插入」。

> **说明** **什么时候能改。**只能在「配置调试」且过程数据停着时改（进配置调试的第三个对话框选「否」），其它时候这排按钮是灰的。改完要重新「激活配置参数」才写进设备。

8.4 启动参数与下发

PDO 分配要通过启动参数（`1C12` / `1C13` 那几行）才能写进从站。改完分配按这个顺序走：

1. 点「生成启动参数」：按当前分配重建 `1C12` / `1C13` 的启动参数行，旧的同类行被替换。
2. 看顶部「PDO 启动参数一致」有没有打勾。切换从站时软件自动比一次；点「校验」手动再比，不一致的项写进日志。
3. 「下发」把这份分配立刻写进从站，要在「配置调试」且过程数据停着。下发前软件再比一次，不一致就拒绝并提示先「生成启动参数」。不点「下发」也行，下次「激活配置参数」时按启动参数写入。

- 「定位启动参数」跳到「启动参数」页里选中 PDO 对应的那一行。
- 「读取在线」把从站里实际生效的分配读回来，和工程、启动参数三方对比，结果写在顶部和日志里。

启动参数页本身怎么用、写入失败怎么查，见第9章「CoE 在线与启动参数」。

8.5 看在线值

- 选中从站后，窗口底部的过程数据表列出本站所有 PDO 变量：名称、在线值、类型、大小、地址、输入/输出、用户 ID、链接到（映射到了哪根轴或哪个通道）。过程数据在跑时在线值随周期刷新。
- 为 1 的布尔量整行绿底。整字量点行首「+」展开成各位。
- 布尔输出行右键「在线写入 TRUE」「在线写入 FALSE」直接驱动输出点，接线核对用。
- 要成组点检一台 I/O 从站，用它的「I/O 测试」页更顺手，见第10章「从站 I/O 功能测试」。

> **注意** **在线写输出会真实驱动端子。**真机上先确认外部负载安全。

8.6 映射排错

现象	先查什么
在线值一直为空	过程数据没跑（要 FreeRun 或 OP）；或者这个对象只在对象字典里、不在 PDO 里。
写了没反应	对象在不在输出 PDO 里。方向反了写不进去。
BOOL 反向、WORD 位错、数值乱跳	位长、数据类型、位偏移与设备实际不一致。看从站「分布式时钟」页的「布局一致性」，见第7章「EtherCAT 与从站」。
写进去马上被覆盖	轴对象（`6040`、`607A` 等）由实时核每周期写，不能当普通 I/O 直写。
「PDO 启动参数一致」没打勾	改过分配没重新生成，点「生成启动参数」。
激活时从站进不了 OP，日志说 PDO 分配被拒	看日志里指出的对象。删了带 `M` 的 PDO、往带 `F` 的 PDO 里插了条目、插了「未声明」可映射的对象，设备都会拒。先「恢复默认」再逐步改。

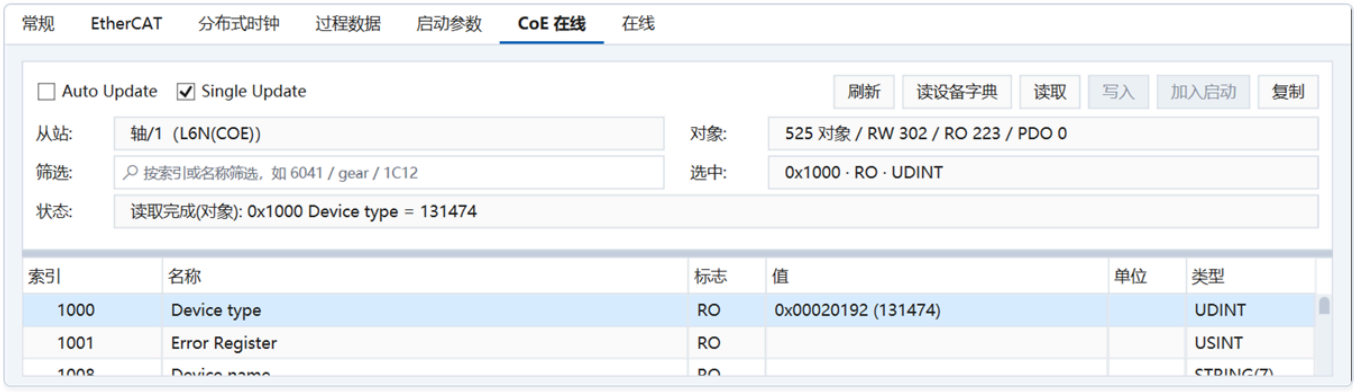
说明 运动相关对象即使在 PDO 表里可见，也不能当普通 I/O 直接写。轴控制统一走轴页或 SDK 的轴指令。

第9章 CoE 在线与启动参数

「CoE 在线」页读写从站对象字典，「启动参数」页维护激活时自动写进从站的那张表。这章讲页面用法、运行中读写 CoE 的规则、启动参数的来源和下发时机，以及怎么实测一圈的脉冲数。

9.1 CoE 在线页

选中从站，打开「CoE 在线」页。上方是从站、对象计数、筛选框、选中对象和状态一行；下方是对象表：索引、名称、访问、值、类型。可写的行加粗并带底色。



从站「CoE 在线」页：对象表与读写按钮，状态行显示刚读到的对象和值

- 筛选框按索引或名称筛，如 `6041`、`gear`、`1C12`。
- 复合对象（设备信息、SM 分配、PDO 映射等）折叠成一行，值列显示 `> N <`（N 是子索引个数），点行首「+」或双击父行展开。
- 访问列带 `*`（如 `RW*`）的行是软件按 CiA402 标准表补进来的，设备没有报告过这个对象，读取多半失败。
- 「读设备字典」直接从设备读取对象字典，替换说明文件提供的离线字典；值列留空，点「读取」再取值。要在「配置调试」且过程数据停着时做，运行中不做。仿真模式没有真实设备，这个按钮用不了。

9.2 读值

操作	作用
「读取」	读选中的一行。右键菜单里也有。
「Single Update」	选中一行就自动读一次，不周期刷。日常确认参数用它。
「Auto Update」	每 500 ms 重读选中的那一行，盯状态、诊断对象的变化用。没选中行不发报文；已映射进 PDO 的对象从过程映像取，不占总线。与「Single Update」互斥。
「刷新」	重新加载对象表。
「复制」「复制索引」「复制整行」	把选中行拷到剪贴板，发给厂商或做记录用。

9.3 改一个参数

1. 双击可写对象的行，弹出 Set Value Dialog。Dec / Hex / Float / Bool / Binary / Bit Size 几栏指向同一份字节，改任一栏其余跟着变。
2. 点「OK」写进从站，「Cancel」什么都不做。值列不能在表格里直接改，编辑入口只有双击，避免误触即写。改好值的行也可以点「写入」或右键「在线写入」。
3. 写完点「读取」回读。驱动器可能钳位或按自己的单位换算，以回读值为准。

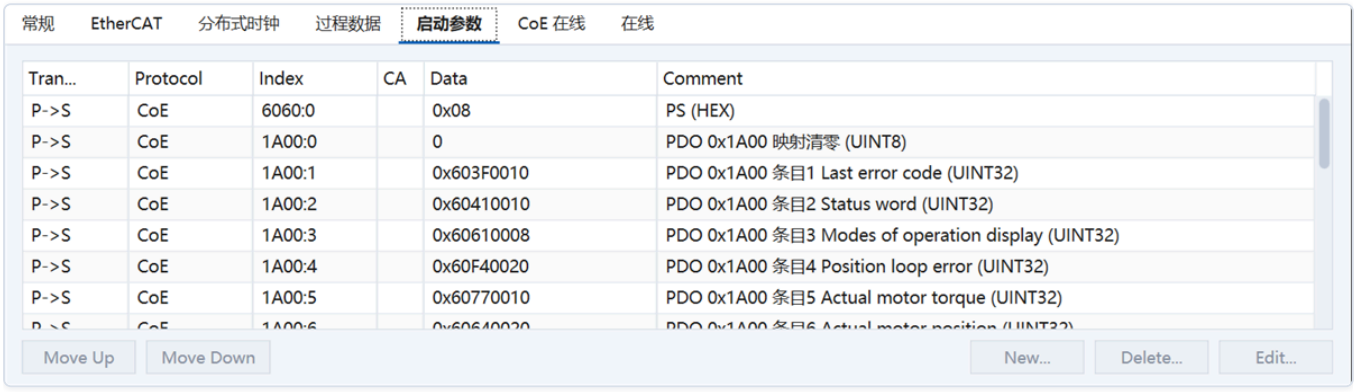
- 只能写访问权限允许的对象。哪个状态下允许写哪些对象由驱动器决定，看厂商手册。
- 写了没生效，看是不是要重新走一遍 PRE-OP 到 OP，或者要断电重启。
- 运动参数、安全相关参数先停轴再写。

9.4 运行中（OP）读写 CoE 的规则

- 「实时运行」的 OP 里可以直接读写 CoE，不用停总线。运行期查 `603F` 错误码、临时改一个参数都可以就地做。
- 每次读写都有超时限制。从站不应答时按超时处理，报「超时/无响应」，不影响周期通信。
- 一次只处理一个读写请求。上一个请求还没返回时再点，软件忽略这次并在状态栏说明，等它返回后再点。
- 它是非周期通道，不要拿它做高频轮询。要持续观察的值加进 PDO，见第8章「PDO 与过程数据」。
- 某个参数运行中能不能写由驱动器决定。只允许 PRE-OP 写的对象（PDO 映射、SM 分配等）驱动器会拒绝，状态栏显示它给的原因。
- 运行中写入只写驱动器，不改工程，不动启动参数。要掉电不丢，写 `0x1010:1` 存进驱动器；要随工程走，回「配置调试」用「加入启动」。
- 已映射进 PDO 的对象，运行中用 CoE 写会看起来写不进去，周期 PDO 每拍都在覆盖它。这类值通过轴指令或 PDO 改。
- 「读设备字典」运行中不做，它要连发几百笔邮箱事务。

9.5 启动参数

「启动参数」页是一张 SDO 列表，激活时软件按表里的顺序、在指定的状态切换前写进从站。列：Tran...（下发阶段）、Protocol、Index、CA、Data、Comment。



从站「启动参数」页：按下发顺序排列的 SDO 列表，下面是上移、下移、新增、删除、编辑

- 按钮「New...」「Edit...」「Delete...」「Move Up」「Move Down」，右键菜单相同。顺序就是下发顺序，表不能按列排序：先清 PDO 映射再写映射项这类先后依赖，顺序颠倒会导致整段配置失败。
- 编辑对话框：Index 可写 `6091` 或 `6091:01`，输完自动带出常用对象的名称、类型，并在线读当前值填进 Data（可改，也可点「读当前值」）；Data 支持十六进制（`0x` 前缀）、整数、BOOL、字符串；Data Type 填 `CA` 表示用完整访问（Complete Access）一次写入整个对象；Transition 选 IP（INIT 到 PRE-OP 前）、PS（PRE-OP 到 SAFE-OP 前，默认）、SO（SAFE-OP 到 OP 前）。

表里的行有三个来源：

- 说明文件自带的启动参数，挂上说明文件时进来。
- 「过程数据」页「生成启动参数」生成的 `1C12` / `1C13` 分配行。
- 「CoE 在线」页选中可写对象点「加入启动」（右键「加入启动参数」）。要在「配置调试」且过程数据停着时做；只读对象和 PDO 过程数据行不能加。

启动参数与 PDO 分配的一致性检查在「过程数据」页点「下发」时自动做，不一致的分配拒绝下发，先「生成启动参数」重建。写入失败会导致无法进入 OP，日志会写明是哪一站、哪一条；先看对象的访问权限、数据类型、要不要在 PRE-OP 写、值在不在设备允许的范围内。

> **说明** **回零参数被拒不影响进运行。**回零速度这类参数（如 `6099:1`）被驱动器拒收时只在日志里告警，该轴用「驱动器回原点」时按驱动器内部存的值走。有的驱动器对速度有下限，拒收多半是值太小。

9.6 标准区与厂商区

区段	内容	各家是否一致
`0x1xxx`	EtherCAT 通信区：设备信息 `1000` / `1008` / `1018`，PDO 映射 `1600` / `1A00`，SM 分配 `1C12` / `1C13`。	一致
`0x2xxx`	厂商自定义：电流环增益、编码器分辨率、细分、各家的 P 参数。	各家各定义，同厂不同型号也可能不同
`0x6xxx`	CiA402 标准：控制字 `6040`、状态字 `6041`、目标位置 `607A`、实际位置 `6064`、错误码 `603F`。	一致，运动主链路靠它

软件不内置任何厂商的地址对照表。对象名称来自设备的说明文件，或点「读设备字典」从设备直接读取，因此任何厂商的设备都能显示对象名称。

9.7 一圈多少脉冲：用 6064 实测

手册和说明文件都可能与实物固件不符，以电机实际转过的圈数为准。驱动器里的分辨率、齿轮比（`6091` / `6093`，或 `608F` + `6092`，厂商实现其中一套）不用动，只量出"一圈的计数"，填到轴「参数」页的 `driveCountsPerRev 驱动器一圈计数` 一行；运动单位是 mm 时再填 `leadMm 导程`，是 deg 时填 `leadMm 每圈度数`，`→ 换算结果` 一行显示算出的脉冲当量。真实轴的定位在「实时运行」下做。



轴「参数」页展开后的底层参数：运动单位、驱动器一圈计数、导程、方向取反与换算结果

1. 在电机轴或联轴器上做个记号。确认周围无人、无干涉，轴能自由转一整圈。
2. 使能轴。CoE 在线页「读取」一次 `0x6064`，记下起始值。
3. 用轴页的定位走一个整数增量（步进先试 `10000`，伺服先试 `1048576`）。
4. 停稳后再「读取」一次 `0x6064`，记下结束值。
5. 一圈的计数 = (结束值 − 起始值) ÷ 实际转过的圈数。

转快了目测圈数不可靠。稳妥的判法：只判记号"回没回原位"，不数圈；或者去使能但保持 OP，读 `6064`，用手把轴正好转一整圈回到记号再读一次，差值就是一圈的计数（手转不准就转 5 圈除以 5）。

用户单位	脉冲当量 =	例（一圈 10000 计数）
圈	一圈的计数	10000
mm（丝杆）	一圈的计数 ÷ 导程	导程 5 mm → 2000
度	一圈的计数 ÷ 360	约 27.78

- 脉冲当量是逐轴的。伺服常见百万级一圈，步进常见一万一圈，混挂时每根轴各填各的。
- 运动单位是 pulse 的轴不做这层换算，G 代码、点动和定位里的数就是驱动器计数。同一页的「一圈脉冲数换算…」改的是 `pulsePerUnit 每 mm 脉冲数(导入/示例用)` 一行，只用于「导入段表」的缩放默认值和 CNC 硬件通道的「示例程序」，与这里的脉冲当量无关。
- 闭环步进常有两个分辨率对象（指令侧细分、编码器分辨率）并存，闭环模式下认的是编码器那个，先确认再填。
- 轴参数页的跟随误差限值和默认动力学（单位都是脉冲）要跟着基准调：对步进，伺服档的跟随误差限值等于好几圈，保护形同虚设。
- 脉冲当量在轴参数里，保存工程即可，不下发到驱动器。驱动器里改过的参数要写 `0x1010:1` 或「加入启动」才固定得下来。

第10章 从站 I/O 功能测试

「I/O 测试」页把一台 I/O 从站的每个通道按类型列出来，直接看输入、写输出、给通道起业务名。出厂点检和现场接线核对用它。

10.1 打开 I/O 测试页

工程树里选中一台 I/O 从站，右侧出现「I/O 测试」页签。驱动器上不显示这一页。

看输入、写输出都要过程数据在跑：「配置调试」进 FreeRun（第三个对话框选「是」），或「实时运行」。在「配置调试」里过程数据没跑就写输出，软件会先问一句，确认后自动起 FreeRun。

没有说明文件的 I/O 模块也能测，通道按设备报告的默认过程数据列出，见第7章「EtherCAT 与从站」。

10.2 页面读法

点「载入点位」把本站通道刷进表格。每行：点位、方向、类型、字节.位、对象、当前值、写输出、映射名。

- 布尔量的当前值用徽章显示：● ON（绿）、○ OFF（灰）。
- 数值通道（UINT、INT、REAL）直接显示在线值，点行首「+」展开成各位。
- 「方向」区分输入、输出；「字节.位」是它在过程映像里的位置。

10.3 测输入、写输出

通道	操作
DI（数字输入）	只读。拨动现场信号，看对应行在 ON / OFF 间跳变，核对接线。
AI（模拟输入）	只读。看当前值随输入变化。
DO（数字输出）	行上「写1」「写0」直接驱动该点。工具栏「DO 走位巡检」加「间隔(ms)」让所有 DO 依次点亮走一遍，「停止」结束。
AO（模拟输出）	值框里按类型填数（十进制、`0x` 十六进制、REAL 浮点），点「写入」。

工具栏「全部置位」「全部清零」一键把本站所有数字量输出拉高或拉低。模拟量、字通道不参与批量置位，只能在该行的值框里给值。

> **注意** **写输出会真实驱动端子。**真机上确认外部负载安全后再写。「DO 走位巡检」和「全部置位」会逐点或整片通断，先确认不会误动执行机构。

10.4 业务名

「映射名」列可以给通道起业务名（如"夹爪到位"），点检时比字节位号直观。名字和「过程数据」页共用，随工程保存。

PART 03 · 第三篇

运动控制

单轴运动与安全边界、回零、多轴插补、CNC、电子齿轮 / 凸轮与机器人。

轴运动与安全边界 · 回零 · 插补通道 · CNC 与 G 代码 · 电子齿轮 · 电子凸轮 · 机器人

第11章 轴运动与安全边界

本章说明轴在软件里怎么被驱动：三条运动链各管什么、哪个模式下能动、命令的受理和完成怎么分、停止与复位各做什么、S 曲线参数怎么配，以及日志里的 ErrorID 对应什么。

11.1 三条运动链

所有轴动作分三条链走，互不穿插。

链	谁在发命令	在哪里执行
单轴命令链	轴页「在线」「功能」页签上的使能、复位、点动、寸动、定位、回零；软卡 SDK 的 `SCM_Axis*` 指令。	实时核里该轴的单轴通道。界面和客户程序走同一条链，经过同样的检查。
组通道链	插补通道页、CNC 通道页的「启动」；软卡 SDK 的 `SCM_Group*`、`SCM_Cnc*`。	实时核组通道。建组即把成员轴从单轴链接管走，解组才交还。
机器人通道链	机器人页的点动、示教运行；软卡 SDK 的 `SCM_Robot*`。	主软件里的机器人运行时计算运动学，再下发给各关节轴。见第17章「机器人总览」。

电子齿轮、电子凸轮的从轴同样属于通道占用，见第15章「电子齿轮」「电子凸轮」。

三条链有一个共同点：页面和客户程序都不直接按周期写 `6040`、`6060`、`607A`、`60FF`，这些对象由实时核按周期写入。

客户程序看到的是 `SCM_AxisPower` / `SCM_AxisReset` / `SCM_AxisStop` / `SCM_AxisHalt` / `SCM_AxisEStop` / `SCM_AxisHome` / `SCM_AxisMoveAbs` / `SCM_AxisMoveRel` / `SCM_AxisMoveVel` / `SCM_AxisJog` / `SCM_AxisGetStatus` 这一族，逐条说明见《软卡 SDK 编程手册》「软卡 SDK 指令说明」。

11.2 哪个模式下能动

状态	能做什么	不能做什么
「仿真」	仿真轴的全部运动；组插补、CNC、齿轮/凸轮、机器人都能跑；SDK 客户程序可以直接连。	没有真实回零开关，I/O 是内存里的位。
「配置调试」	改组态、CoE 在线、简单 I/O、诊断；进 FreeRun 后能看过程数据和驱动器状态。	真实轴不运动：点动、寸动、定位、回零按钮不可用；SDK 的轴命令、建组、写输出会被拒绝，状态和诊断读取不受限。
「实时运行」	真实轴的全部运动。	目标位置和实际位置差很大时别直接启动运动，先复位。
「界面锁定」（任何模式）	看轴反馈、6041、6061、6064、速度、报警和诊断。	本软件只读：不使能、不点动、不写参数、不动总线。只锁本软件，不锁软卡 SDK 客户端，客户程序照样能驱动轴。

> **注意** **真实轴运动前看模式条。**顶部模式条要在「实时运行」、全拓扑 OP、WKC 满值。配置调试下能读到驱动器状态，不等于可以安全运动。

11.3 受理不等于完成

- 使能、复位、回零都是异步受理：返回 0 只表示命令已受理，不表示已执行完成。
- 完成看轴状态：轴页看「已使能」「已回零」「到位」三个勾和「CiA402 状态」；客户程序轮询 `SCM_AxisGetStatus` 的 `powered` / `faulted`。
- 轴没使能、或使能还在进行时发 `SCM_AxisMove*`，直接被拒。未等使能完成就建组，同样会被拒绝。
- 轴被组、CNC、齿轮、凸轮、机器人占用时，单轴命令一律被拒。先解散占用它的通道。
- 使能序列最长约 12 s，复位序列约 3.5 s，超时算失败，去查驱动器报警和 STO。

11.4 轴页上的运动控件

轴页六个页签：「常规」「设置」「参数」「动态」「功能」「在线」。和运动有关的在后四个。

页签	里面有什么	要点
「在线」	「轴在线监控」：规划/实际位置、规划/实际速度、轴状态、CiA402 状态、运行信息。「点动 / 控制」：四个方向键（F1 快速后退、F2 慢退、F3 慢进、F4 快速前进，按住走、松开停）、慢速/快速两个速度框、「↻」复位、「■」停止、使能/取消使能键、寸动（寸动位置 + 速度，「◀|」「|▶」按当前位置做相对寸动）。「回原点」：当前配置摘要和「启动回原点」。「位置定义」：绝对/相对、数值、「设置实际位置」。	位置定义只改坐标系不动轴，轴要静止。软卡 SDK 客户程序取得控制权期间，「■」停止仍然可用，是在界面上停止客户程序所发运动的唯一方式。
「功能」	「扩展启动」：启动模式 往复序列 / 启停序列 / 速度阶梯序列，「启动」「停止」。	运行参数在启动时冻结，停止后改才生效。首次调试别直接跑往复，先点动。
「动态」	「默认动力学」：默认速度、默认加速度、默认减速度、Jerk 加加速度、目标位置。「回零 / 停止」：模式选择、回零参数、回零偏移、停止 / 快停减速度。	回零参数见第12章「回零」。
「参数」	换算（单位、一圈计数、导程）、`invertDirection 方向取反`、软限位使能与正负值；「上传」「下载」「全选」。`pulsePerUnit 每 mm 脉冲数(导入/示例用)` 和「一圈脉冲数换算…」只影响「导入段表」的缩放默认值与 CNC 硬件通道的「示例程序」，不参与轴的运动换算。	轴使能中改换算参数会被拒。方向取反后软限位正负自动互换。

> **提示** **先低速验证。**真实轴第一次动，先点动确认方向和当量对得上，再定位或回零。方向反了在「参数」页改 `invertDirection 方向取反`，对界面和软卡 SDK 发出的所有运动都生效。

11.5 实际值与规划值

- 实际位置来自驱动器反馈（`6064`），是电机真到了哪里。
- 规划位置是实时核在当前周期生成的位置，用来平滑过渡和停止同步。

- 目标位置是当前命令要去的位置。
- 实际速度来自驱动器反馈，会因采样、量化、滤波在几个值之间跳，正常。
- 停止完成后规划位置会同步到实际位置，避免下次启动追旧目标。

11.6 停止与复位

- 正常停止：轴页「■」或 `SCM_AxisStop`，按「动态」页的停止减速度减到零，轴回到静止态。`SCM_AxisHalt` 也是受控减速，但减速中可被新的运动指令接管。`SCM_AxisEStop` 按快停减速度停。
- 异常停止：限位、驱动器故障、过程数据停止、实时核异常。停下后先看状态字、错误码、实际位置、规划位置、目标位置，再决定下一步。
- 停止距离不合要求，改减速度和停止策略，别用清控制字代替。
- 复位（轴页「↻」或 `SCM_AxisReset`）：轴在故障态时先做驱动器故障复位；没故障时把规划位置、目标位置同步到实际位置。这一步是防飞车的：目标和实际差很大时直接使能，轴会猛追目标。
- 复位不清工程参数、不重扫总线。复位会改变驱动器状态，软卡 SDK 客户程序取得控制权期间，轴页的复位不可用。

> **提示** **先复位再使能。**实际位置和规划位置差得远，先点「↻」让两者一致，再使能、再运动。

11.7 单轴 S 曲线（Jerk）

单轴定位、速度运动、点动都按七段 S 曲线走：加减速段本身再受加加速度 Jerk 限制，起停和换向没有加速度突跳。

取值	行为
命令带的 `jerk` > 0	用这个值做七段 S 曲线，优先级最高。
命令带的 `jerk` ⩽ 0 或不给	取该轴「动态」页「默认动力学」组的「Jerk 加加速度」。
最终值 = 0	退化为梯形曲线。

- Jerk 在运动启动前下发到实时核，运动中改它不影响正在跑的这一段。
- 单位随轴的用户单位（mm/s³、°/s³ 或 pulse/s³）。经验取法：加速度 × 5 ~ 10，太小会明显拉长加速时间，太大等于没加 S 曲线。
- 生效范围：绝对定位、相对定位、速度运动、点动，以及轴页的点动与点位运动。受控停止和暂停按各自的减速度做梯形减速。
- 插补通道的 S 曲线是另一套参数（通道「插补参数」页的 S 曲线和加加速度），见第13章「插补通道」。
- 出厂默认 0，即梯形。要用 S 曲线，在轴「动态」页或命令参数上给一个正值。

11.8 安全边界

- 运动前确认：模式在「实时运行」；轴映射到真实从站，不是离线占位轴；驱动器没有报警；实际、规划、目标位置接近；速度、加速度、寸动距离在机械允许范围内。

- 软限位在「参数」页启用并填正负值。它只在轴已使能且已回零时生效，参数页软限位分组的标题会显示当前是否生效。单轴定位目标超限被拒；点动到限位停在限位上，越界后只允许往回点动。组运行时每个周期检查软限位，目标超限即按故障停组，并在提示中指出是哪根轴。
- 运动中切到「配置调试」或总线停止，真实轴运动会停止或被拒。
- 伺服必须走分布时钟同步。读回的同步模式不是 DC 时，软件只告警、不阻止使能，请查看日志，详见第7章「EtherCAT 与从站」。

11.9 ErrorID 对照表

三条链共用一套错误码，日志和在线值按十进制显示。排查时先看第一条相关日志再对表。

ErrorID	含义	触发条件与排查
1296236545	后端未绑定	这条功能没有绑定实时核后端，无轴动作。正常使用不会出现。
1296236546	轴不存在	轴号超出工程轴列表，或映射的从站已丢失。
1296236547	轴未使能	运动指令触发时轴不在使能状态。先使能并等状态到位。
1296236548	运行模式未就绪	系统不在「实时运行」（OP）或过程数据没跑。配置调试下不允许伺服运动。
1296236549	后端拒绝	实时核操作失败：通道用尽、6040/6041/6064 等对象没映射到 PDO、读写失败。查 PDO 映射和轴数上限。
1296236550	轴被占用	该轴正被界面控制，或处于受控停止的锁轴期间。
1296236551	超时	使能序列（最长约 12 s）或复位序列（约 3.5 s）超时。查驱动器面板报警、STO 端子和「故障排查」。
1296236552	驱动器故障	驱动器在 Fault（6041 bit3 = 1）。先做故障复位，复位不掉查 603F 的故障含义。
1296236553	参数无效	速度 ⩽ 0、点动双向同时给出、绝对定位目标等于当前位置。
1296236554	复位互锁	故障复位进行中又触发了使能或重复复位。等复位完成，最长约 3.5 s。
1296236556	轴已被插补组接管	该轴在插补、CNC、凸轮、齿轮、机器人通道里，单轴运动指令被拒。先解散占用它的通道。

> **说明** **保留的错误码。**表中 ErrorID 换算成十六进制后都以 0x4D43（ASCII "MC"）开头。0x4D43000B / 000D / 000E / 000F 为保留码，正常使用不会出现。

11.10 典型顺序

阶段	做什么	运行时动作
1. 进 OP	模式条切到「实时运行」；改过工程就点「激活配置参数」。确认全拓扑 OP、WKC 满值。	实时核接管周期收发。
2. 使能	轴页「在线」的使能键，或客户程序 `SCM_AxisPower`。	按 CiA402 状态机分段推进到 Operation Enabled。
3. 等到位	轮询轴状态，等 `powered` 为真、`faulted` 为假。	返回码只表示已受理，完成以轴状态为准。
4. 低速验证	先点动，确认方向与当量对得上，再定位或回零。	方向反了在「参数」页设方向取反，对界面和软卡 SDK 发出的所有运动都生效。
5. 上层运动	需要联动才建组、开 CNC 通道、起机器人。	建组把成员轴从单轴通道接管走。

11.11 错误处理原则

- 先看第一条失败日志，别从最后一条往回猜，后面的多半是连锁反应。
- 拿到 ErrorID 先对上表，再决定查驱动器还是查工程配置。
- 驱动器 Fault 先复位；复位不掉读 603F，别反复点使能。
- "被占用"类错误查的是谁占着轴（界面、组、CNC、机器人），不是驱动器。
- 真实轴先用小行程、低速度验证方向和比例，再放大参数。

第12章 回零

这章说轴怎么回零：三种回零方式怎么选、在哪配、怎么启动、完成看什么，再按 CiA 402 逐一图解驱动器回零 6098 的各个方式，供配驱动器回零和对照驱动器手册用。

12.1 三种回零方式

每根轴在「动态」页签「回零 / 停止」组的「模式选择」里三选一。参数按所选模式显示。



轴「动态」页：默认动力学、点动速度设定，以及回零参数的唯一配置处

模式	谁在走	要什么	参数
「驱动器回原点」	驱动器按 CiA402 `6098` 方式自己走，软件只负责切换模式、置位启动位、监视状态字。	原点/限位开关接在驱动器上；带 Z 相的方式要编码器有 Z 相。一台多轴驱动器只有第 1 根轴能用这种方式。	回零方式（方式号旁实时显示它的含义）、回零速度（6099:01）、爬行速度（6099:02）、回零偏移（607C）。
「控制器回原点」	软件读 EtherCAT I/O 模块上的原点开关，自己规划：退出开关 → 快找 → 退出 → 慢爬精定位 → 置原点 → 偏移停靠。	原点开关接在 I/O 模块上。	原点开关从站位置、站内位号、高有效、回零方向（+1 / −1）、快找速度、慢爬速度、搜索行程上限（0 = 不限）、无进展超时。站内位号填 −1 时回零被拒。
「当前位置置零」	不动。把当前物理点定义为「回零偏移」那个坐标，并置已回零。	无。绝对值编码器轴、临时零点、手动摇到基准位后置零都用它。	回零偏移。

- 回零偏移三种模式通用：驱动器回原点下发到 `607C`；另两种是软件原点。
- 启动：轴页「在线」页签「回原点」组的「启动回原点」，或客户程序 `SCM_AxisHome`。SDK 侧的回零方式在回零参数的 Engine 字段：0 驱动器、1 控制器、2 当前位置置零。

- 启动是异步受理，完成看「在线」页「已回零」勾上，客户程序按《软卡 SDK 编程手册》「软卡 SDK 指令说明」回零一节轮询。
- 仿真模式没有真实开关，仿真轴用「当前位置置零」，或在「在线」页「位置定义」里直接定义坐标。

> **说明** **驱动器回原点的参数随激活下发。**方式、6099、609A、607C 作为启动参数在「激活配置参数」时写进驱动器。之后改了参数，下次启动回零时软件会先把新值写进驱动器并读回确认；写不进去就拒绝回零并在消息栏说明原因，不会按旧值回零。

> **说明** **参数被驱动器拒收。**激活时某个回零参数写不进去（常见是 6099:01 低于驱动器允许的下限），激活仍会继续进入运行，日志会指出是哪一台驱动器的哪个参数。该轴用驱动器回原点时按驱动器内部存的值走，和轴页填的不一致。遇到这种情况，把该参数改到驱动器允许的范围内，再重新激活。

12.2 驱动器回零：相关对象

对象	名称	说明
`6060h`	Modes of operation	写 `6` 进入回零模式（HM），从 `6061h` 读回确认。
`6098h`	Homing method	回零方式编号，即本章各方法。正值为标准方法，负值为厂商自定义。
`6099h:01`	Speed during search for switch	搜索开关速度（快速段）：找限位开关、原点开关时用。
`6099h:02`	Speed during search for zero	搜索零点速度（慢速段）：逼近开关边沿或 Z 相时用，决定重复精度，要明显低于快速段。
`609Ah`	Homing acceleration	回零过程中所有加速、减速和反向用的加速度。
`607Ch`	Home offset	零点偏置：机械原点与应用零点之差。回零完成后当前位置直接显示为 607C 的值，轴不会再自动走一段。要让轴真正离开开关区，回零完成后再做一次定位。
`6040h`	Controlword	bit4 由 0→1 启动回零（期间保持 Operation enabled）；bit8（Halt）暂停回零。
`6041h`	Statusword	bit10 = Target reached，bit12 = Homing attained，bit13 = Homing error，组合含义见下表。

这些对象在「CoE 在线与启动参数」里能读写。

12.3 驱动器回零：启动与完成时序

点「启动回原点」后软件按下面的顺序走，你要盯的只是 `6041h` 的 bit12 / bit13。

- ① 轴使能到 Operation enabled。
- ② 写 `6060h = 6`，等 `6061h = 6`。

- ③ 确认 `6098h`、`6099h:01/:02`、`609Ah`、`607Ch` 已在驱动器里。
- ④ `6040h` bit4 置 1 启动回零，过程中不撤使能。
- ⑤ 监视 `6041h`：bit13 = 1 回零出错；bit12 = 1 回零完成。
- ⑥ 完成后 bit4 清零，`6060h` 切回 CSP（8）。

bit13 出错	bit12 达成	bit10 目标到达	状态含义
0	0	0	回零正在执行。
0	0	1	回零未启动，或被中断（Halt）。
0	1	0	原点已找到，轴还没在目标位置停稳。
0	1	1	回零成功完成。
1	0	0	回零出错，轴仍在运动。
1	0	1	回零出错，轴已停止（撞限位、超时、开关接线错误）。

回零出错时消息栏会提示核对回零方式与限位、原点开关的接线方向；轴长时间没完成会按超时中止。

12.4 图例：怎么读本章的图

- 蓝色粗实线箭头：快速段，用搜索开关速度 `6099h:01`。
- 蓝色虚线箭头：慢速段，用搜索零点速度 `6099h:02`。
- 蓝色圆点：起始位置（示意，实际可从行程内任意位置出发）。
- 绿色圆点与绿色竖虚线：回零完成位置，即本方法定义的原点。
- 橙色高亮带：原点开关有效区间，下方波形是该开关电平。
- 红色高亮带：限位开关有效区间。
- 紫色短竖线：编码器 Z 相（Index）脉冲出现的位置，每转一个。
- 带圈数字是方法编号；同一张图里多条轨迹线相互独立，各表示一个方法。

12.5 方法总览

方法	参考信号	初始方向	原点位置定义
1	负限位开关 + Z 相	负向	负限位释放后第一个 Z 相。
2	正限位开关 + Z 相	正向	正限位释放后第一个 Z 相。
3 / 4	原点开关（正侧）+ Z 相	正向 *	开关上升沿负侧 / 正侧的第一个 Z 相。
5 / 6	原点开关（负侧）+ Z 相	负向 *	开关上升沿正侧 / 负侧的第一个 Z 相。
7 – 10	中段原点开关 + Z 相 + 正限位	正向 *	开关左右两个边沿两侧的四个 Z 相位置之一。

11 – 14	中段原点开关 + Z 相 + 负限位	负向 *	7 – 10 的镜像。
15 / 16	保留，未定义。		
17 / 18	负 / 正限位开关	负 / 正向	同 1 / 2，但不用 Z 相，原点取限位释放边沿。
19 / 20	原点开关（正侧）	正向 *	同 3 / 4，原点取开关上升沿。
21 / 22	原点开关（负侧）	负向 *	同 5 / 6，原点取开关上升沿。
23 – 26	中段原点开关 + 正限位	正向 *	同 7 – 10，原点取开关左 / 右边沿。
27 – 30	中段原点开关 + 负限位	负向 *	同 11 – 14，原点取开关边沿。
31 / 32	保留，未定义。		
33 / 34	仅 Z 相	负 / 正向	当前方向上第一个 Z 相。
35 / 37	无	不运动	当前位置直接置为原点（37 是新编号，取代 35）。
负值	厂商自定义	见手册	常见为撞机械挡块（转矩检测）回零，行为以驱动器手册为准。

说明 *** 号。**启动时开关已处于有效状态，初始方向自动取反：先离开开关，再执行正常流程。各方法说明里有细节。

12.6 方法 1：负限位开关 + Z 相

- 适用：只有限位开关、没有独立原点开关、编码器带 Z 相的轴。
- ① 以快速度负向运动，直到负限位开关有效。
- ② 反向，以慢速度正向运动。
- ③ 负限位开关释放后遇到的第一个 Z 相脉冲位置即为原点。
- 启动时负限位已有效，跳过 ①，直接执行 ② ③。
- 压在限位开关上运动期间经过的 Z 相不计，图中限位有效区内的 Z 相被忽略。



方法 1：快速负向压上负限位 → 反向慢速 → 限位释放后第一个 Z 相为原点（绿点）。限位有效区（红色）内经过的 Z 相不计。

12.7 方法 2：正限位开关 + Z 相

- 方法 1 的镜像：① 快速正向运动直到正限位开关有效；② 反向慢速负向；③ 正限位释放后第一个 Z 相为原点。
- 启动时正限位已有效，直接执行 ② ③。



方法 2：快速正向压上正限位 → 反向慢速 → 限位释放后第一个 Z 相为原点。

12.8 方法 3 / 4：正侧原点开关 + Z 相

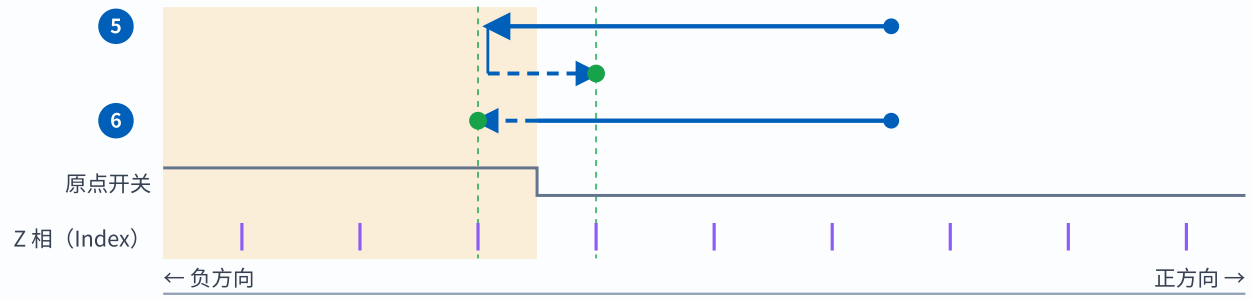
- 原点开关的有效区覆盖行程正方向一侧（挡片从中段一直延伸到正端），参考边沿是开关的上升沿。
- 方法 3（原点在上升沿负侧）：① 开关无效时正向快速运动；② 开关变有效（上升沿）后反向慢速；③ 开关释放后第一个 Z 相为原点。
- 方法 4（原点在上升沿正侧）：① 正向快速运动；② 到上升沿后转慢速继续正向；③ 开关有效区内第一个 Z 相为原点。
- 启动时开关已有效：方法 3 直接负向慢速，开关释放后取第一个 Z 相；方法 4 先负向离开开关，再正向重新逼近上升沿后取 Z 相。



方法 3：越过上升沿后反向，开关释放后第一个 Z 相（上升沿负侧）。方法 4：到上升沿后转慢速继续正向，开关有效区内第一个 Z 相（上升沿正侧）。橙色带为原点开关有效区。

12.9 方法 5 / 6：负侧原点开关 + Z 相

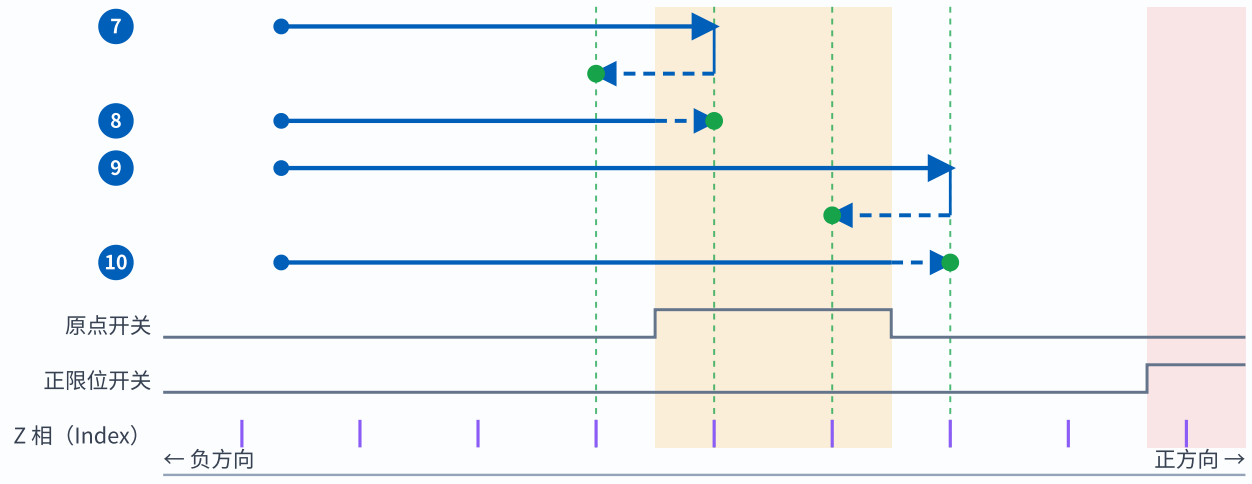
- 方法 3 / 4 的镜像：原点开关有效区覆盖行程负方向一侧，初始方向为负向，参考边沿是负向运动时的上升沿。
- 方法 5：① 负向快速；② 上升沿后反向慢速正向；③ 开关释放后第一个 Z 相（上升沿正侧）为原点。
- 方法 6：① 负向快速；② 上升沿后转慢速继续负向；③ 开关有效区内第一个 Z 相（上升沿负侧）为原点。
- 启动时开关已有效：方法 5 直接正向慢速找释放点后的 Z 相；方法 6 先正向离开开关再负向重新逼近。



方法 5：越过上升沿后反向，开关释放后第一个 Z 相（上升沿正侧）。方法 6：上升沿后转慢速继续负向，开关有效区内第一个 Z 相（上升沿负侧）。

12.10 方法 7 – 10：中段原点开关 + Z 相（初始正向，配合正限位）

- 原点开关只在行程中段一小段区间有效（挡块或窗式开关），有左、右两个边沿；四个方法分别取两个边沿两侧的四个 Z 相位置，图中从左到右依次是 7、8、9、10 的原点。
- 方法 7：① 正向快速；② 开关上升沿（左边沿）后反向慢速；③ 开关释放后第一个 Z 相（左边沿负侧）为原点。
- 方法 8：① 正向快速；② 左边沿后转慢速继续正向；③ 开关有效区内第一个 Z 相（左边沿正侧）为原点。
- 方法 9：① 正向快速穿过整个开关区；② 开关释放（右边沿）后反向慢速；③ 开关重新有效后第一个 Z 相（右边沿负侧）为原点。
- 方法 10：① 正向快速穿过开关区；② 右边沿后转慢速继续正向；③ 开关释放后第一个 Z 相（右边沿正侧）为原点。
- 起始位置在开关正侧（初始方向背离开关）时，轴先压上正限位开关，自动反向后再按上述流程搜索原点开关，前提是驱动器支持限位反向。启动时开关已有效则按各方法直接从相应边沿流程开始。

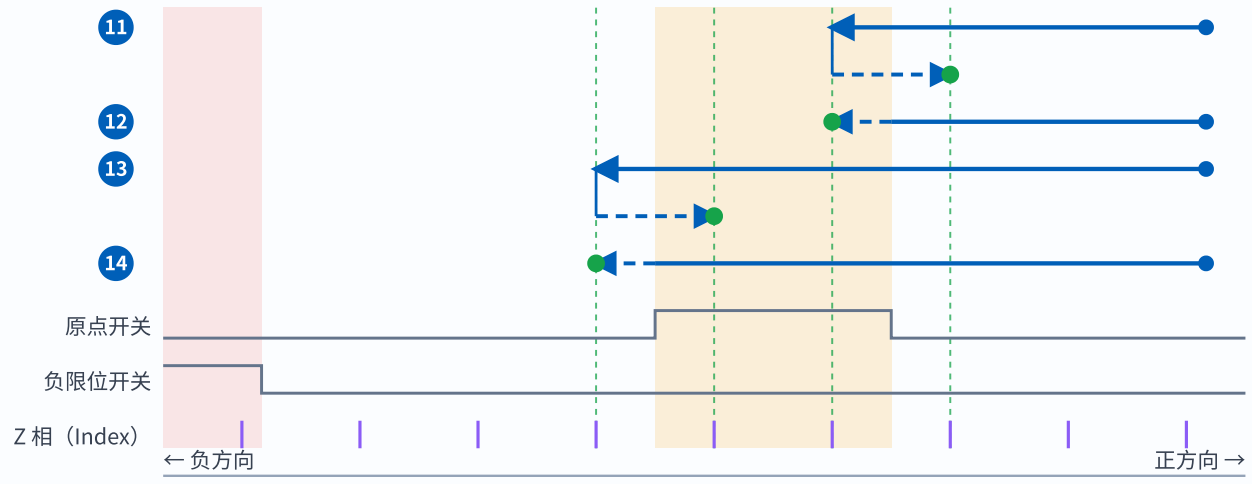


方法 7 – 10：原点开关只在中段有效（橙色带）。四个原点位置从左到右依次对应方法 7、8、9、10——左边沿两侧的 Z 相给 7 / 8，右边沿两侧的 Z 相给 9 / 10。若初始方向背离开关，压上正限位（红色带）后自动反向。

12.11 方法 11 – 14：中段原点开关 + Z 相（初始负向，配合负限位）

- 方法 7 – 10 的镜像：初始方向为负向，背离开关时靠负限位开关反向。

- 方法 11：① 负向快速；② 开关上升沿（右边沿）后反向慢速正向；③ 开关释放后第一个 Z 相（右边沿正侧）为原点。
- 方法 12：① 负向快速；② 右边沿后转慢速继续负向；③ 开关有效区内第一个 Z 相（右边沿负侧）为原点。
- 方法 13：① 负向快速穿过开关区；② 开关释放（左边沿）后反向慢速正向；③ 开关重新有效后第一个 Z 相（左边沿正侧）为原点。
- 方法 14：① 负向快速穿过开关区；② 左边沿后转慢速继续负向；③ 开关释放后第一个 Z 相（左边沿负侧）为原点。



方法 11 – 14：方法 7 – 10 的镜像，初始负向。四个原点位置从右到左依次对应 11、12、13、14；背离开关时压上负限位后自动反向。

12.12 方法 17 / 18：仅限位开关（不用 Z 相）

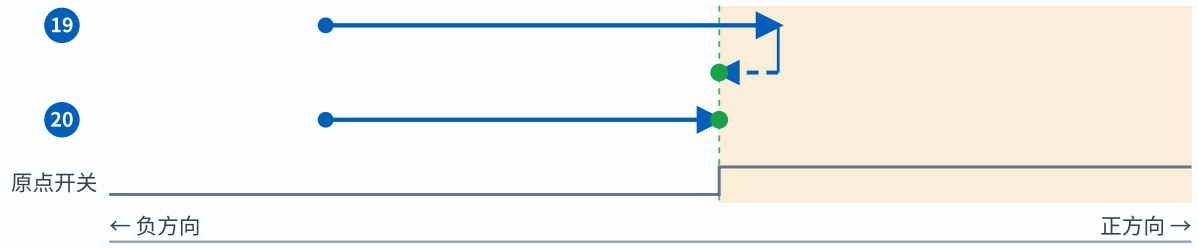
- 与方法 1 / 2 流程相同，但不用 Z 相：限位开关的释放边沿本身就是原点。
- 方法 17：① 负向快速压上负限位；② 反向慢速正向；③ 负限位释放的位置为原点。
- 方法 18：镜像，用正限位。
- 重复精度取决于开关本身（一般 0.01 – 0.5 mm 量级，远低于 Z 相方案），适用于无 Z 相编码器或精度要求不高的轴。



方法 17：压上负限位后反向慢速，限位释放边沿即原点。方法 18：镜像，用正限位。

12.13 方法 19 / 20：正侧原点开关（不用 Z 相）

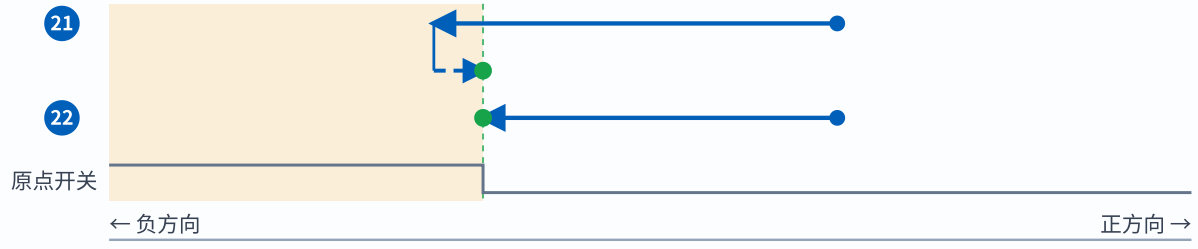
- 与方法 3 / 4 对应，原点直接取开关上升沿位置；19 与 20 的区别只在最终逼近方向。
- 方法 19：① 正向快速越过上升沿；② 反向慢速；③ 开关释放的位置（同一边沿，负向到达）为原点。
- 方法 20：① 正向运动；② 开关变有效的位置（上升沿，正向到达）为原点。
- 启动时开关已有效：方法 19 直接负向慢速找释放点；方法 20 先负向离开开关，再正向重新逼近。
- 不少驱动器为提高重复精度，会在快速找到边沿后自动退回一段，再用慢速二次逼近，以驱动器手册为准。



方法 19 / 20：原点都在开关上升沿，19 负向到达（越过后反向），20 正向到达（直接停在边沿）。

12.14 方法 21 / 22：负侧原点开关（不用 Z 相）

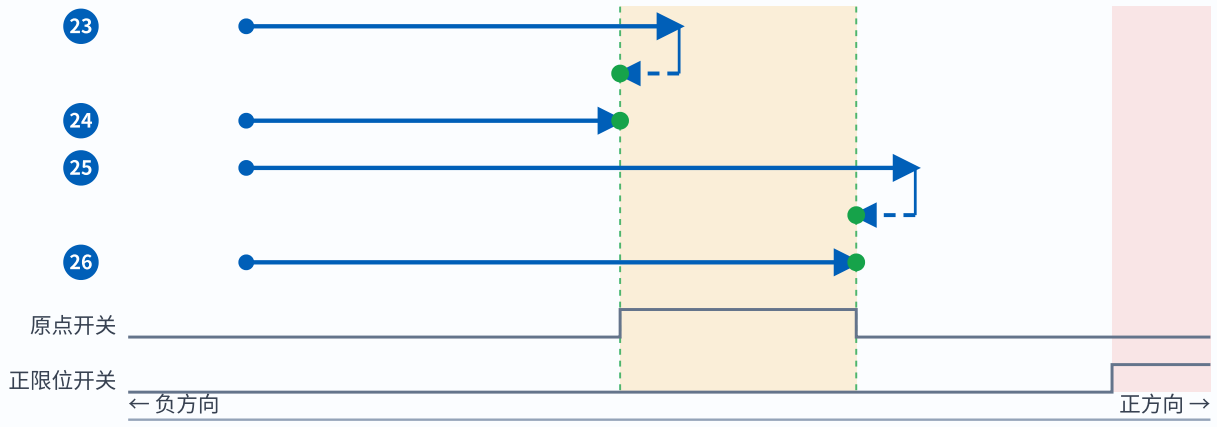
- 方法 19 / 20 的镜像，对应方法 5 / 6：初始负向，原点取负向运动时的开关上升沿。
- 方法 21：① 负向快速越过上升沿；② 反向慢速正向；③ 开关释放位置为原点。
- 方法 22：① 负向运动；② 开关变有效的位置为原点。



方法 21 / 22：原点在负向运动的开关上升沿，21 越过后反向正向到达，22 负向直接停在边沿。

12.15 方法 23 – 26：中段原点开关（不用 Z 相，初始正向）

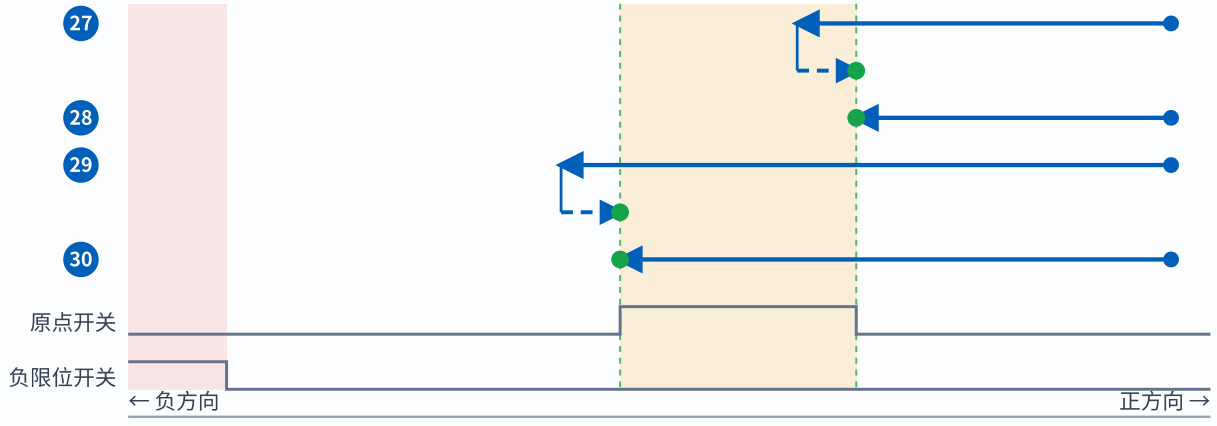
- 与方法 7 – 10 对应，原点直接取开关的左 / 右边沿：23 / 24 取左边沿（分别负向 / 正向到达），25 / 26 取右边沿（分别负向 / 正向到达）。
- 方法 23：正向越过左边沿后反向慢速，开关释放位置（左边沿）为原点。
- 方法 24：正向运动，开关变有效位置（左边沿）为原点。
- 方法 25：正向穿过开关区，越过右边沿后反向慢速，开关重新有效位置（右边沿）为原点。
- 方法 26：正向穿过开关区，开关释放位置（右边沿）为原点。
- 初始方向背离开关时靠正限位反向，与方法 7 – 10 相同。



方法 23 – 26：不用 Z 相，原点直接取开关边沿。23 / 24 落在左边沿，25 / 26 落在右边沿，区别是最终到达方向。

12.16 方法 27 – 30：中段原点开关（不用 Z 相，初始负向）

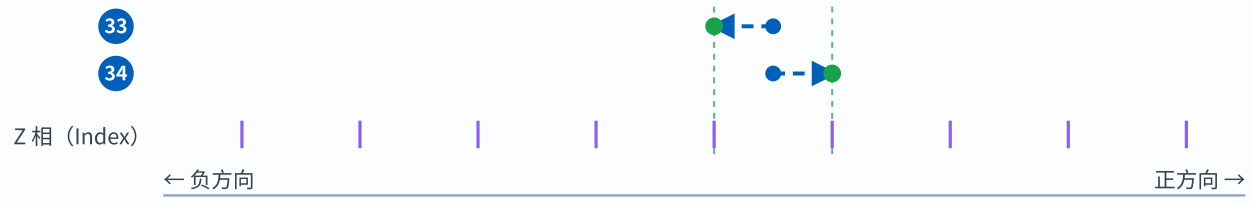
- 方法 23 – 26 的镜像，对应方法 11 – 14：初始负向，背离开关时靠负限位反向。
- 方法 27：负向越过右边沿后反向慢速正向，开关释放位置（右边沿）为原点。
- 方法 28：负向运动，开关变有效位置（右边沿）为原点。
- 方法 29：负向穿过开关区，越过左边沿后反向慢速正向，开关重新有效位置（左边沿）为原点。
- 方法 30：负向穿过开关区，开关释放位置（左边沿）为原点。



方法 27 – 30：方法 23 – 26 的镜像，初始负向。27 / 28 落在右边沿，29 / 30 落在左边沿。

12.17 方法 33 / 34：仅 Z 相

- 不用任何开关：以慢速度向指定方向运动，遇到的第一个 Z 相脉冲即为原点。
- 方法 33：负向；方法 34：正向。
- 适用：行程不足一圈、已靠机械方式粗定位、或与其它方式组合做二次精回零。行程内有多个 Z 相时，回到哪个 Z 相取决于起始位置，用前先确认机械上无歧义。



方法 33 / 34：慢速运动到当前方向上的第一个 Z 相。

12.18 方法 35 / 37：当前位置置零（不运动）

- 轴不产生任何运动：启动回零后，当前位置直接被认定为原点，再叠加 `607Ch` 偏置。
- 方法 37 是 CiA 402 后补的编号，语义与 35 相同并用于取代 35；不同驱动器可能只支持其中一个，以手册为准。
- 适用：绝对值编码器轴（位置本就有效，只需建立应用零点）、增量系统开机后的临时零点、手动把轴摇到基准位后置零。
- 在软件里不必经驱动器：把轴的回零模式选成「当前位置置零」，启动回零就全程不动电机，把当前点定义成回零偏移那个坐标并置已回零。见章首。

12.19 负值方法：厂商自定义

- `6098h` 写负值（−1、−2 …）进入厂商自定义回零，CiA 402 不定义其行为。标准把 −1 ~ −4 留给厂商私有方式。
- 常见实现是撞机械挡块回零（无开关）：以设定转矩或电流阈值顶住机械端部，检测到堵转即认为到达参考点，部分实现再结合 Z 相精定位。用前必须确认转矩限幅、堵转判定时间和机械强度。
- 编号与行为各家不同，配置前查驱动器手册对应章节。

12.20 选型建议与常见问题

轴的条件	推荐方法
有独立原点开关，编码器带 Z 相	3 – 14（按开关安装位置和期望逼近方向选），重复精度最高。
只有限位开关，带 Z 相	1 / 2。
只有开关，无 Z 相（或 Z 相不可靠）	17 – 30。
行程不足一圈 / 已机械粗定位	33 / 34。
绝对值编码器，或只需临时零点	35 / 37，或直接用软件的「当前位置置零」。
无任何开关、允许撞挡块	厂商自定义负值方法（确认转矩限幅后用）。
原点开关接在 I/O 模块上而不在驱动器上	用「控制器回原点」。

- Z 相离开关边沿太近：开关重复性误差可能使回零结果差整整一圈。调整挡片位置，让开关边沿落在两个 Z 相的中间附近（约半圈余量）。

- 开关常开 / 常闭接反：表现为一上电就判开关有效、初始方向相反。先在驱动器诊断或 I/O 页确认开关电平随挡块动作正确翻转。
- 慢速段速度过高：冲过边沿或 Z 相，每次回零位置差几个脉冲。降低 `6099h:02`。
- 需要限位反向的方法（7 – 14、23 – 30）：确认限位开关接入驱动器本体且驱动器支持压限位自动反向，否则直接报限位故障。
- 回零完成后轴还在开关区里：607C 只改数值不走位，见上面对象表。回零完成后再执行一次定位。

> **注意** **开关没接就启动。**选了开关类方法但开关没接到驱动器，轴会一直运动到限位或撞机。首次调试用最低速度，手放在急停上。

第13章 插补通道

这章讲怎么用插补通道让 X/Y/Z 三根轴（加最多 5 根随动轴）沿一条路径联动：建通道、映射使能、给路径、运行观察、接真轴，仿真轴和真实伺服是同一套操作。

13.1 通道与四个页签

工程树里右键 NC 任务节点 →「添加插补」，生成一个插补子节点。点选它，右侧有四个页签。



插补通道「常规」页：通道名与周期、X/Y/Z 映射轴、随动轴 W0~W4 与一键映射按钮

页签	用途
「插补常规」	通道名、端口、周期、已映射、段数、状态；X/Y/Z 映射下拉（候选是本 NC 任务轴文件夹里的轴）和随动轴 W0~W4 折叠区；「一键创建仿真三轴映射」「一键复位映射轴」「一键使能映射轴」。
「插补参数」	路径规划参数（速度、加减速、拐角模式、前瞻、S 曲线、单段模式）和映射轴限制表。
「插补在线」	DRO 位置显示、运行状态、「启动 / 继续」「暂停」「停止」「复位」、进给倍率滑块、轨迹视口、路径速度曲线。
「插补功能」	段表编辑（直线、圆弧、动作行）、演示轨迹、导入段表。

页面上的「启动」会建组并下发路径段，由实时核的组通道执行，与软卡 SDK 的 `SCM_Group*` 走同一条链，经过同样的检查。建组即占轴，成员轴在通道解组前不接受单轴命令。组通道无法建立时启动被拒，通道状态和消息栏会写原因：映射不全、随动轴有空位、坐标系占满、建组失败的具体原因。

13.2 映射并使能

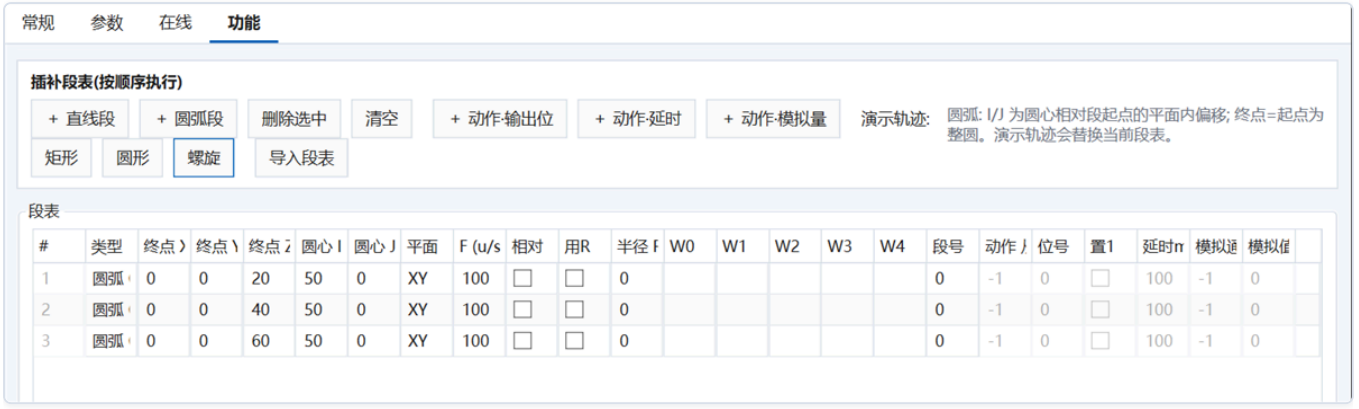
- 仿真轴：「插补常规」页点「一键创建仿真三轴映射」，软件在本 NC 任务的轴文件夹下建三根仿真轴并映射到 X/Y/Z；再点「一键使能映射轴」。状态变成「就绪(仿真三轴映射)」，已映射 3/3。
- 真实轴：在下拉里选映射到从站的轴。硬件轴和仿真轴可以混在一个通道里。先点「一键复位映射轴」（硬件轴走驱动器故障复位，使能前的惯例动作），再点「一键使能映射轴」（硬件轴和 `SCM_AxisPower` 同链，后台逐轴受理并轮询到位）。

- 使能不一定非用一键：在轴页逐个使能、或由客户程序使能，启动时都认，以驱动器实际状态为准。
- 随动轴：点「＋ 添加随动轴」展开 W0~W4。必须从 W0 起连续填，中间留空启动被拒；组宽 = 3 + 随动轴数，最多 8 轴。随动轴沿路径弧长线性推进，不参与几何，没有反向选项。

硬件轴的插补只在硬实时下可用，档1 独占核和档2 共享核两档都行，见第6章「实时形态与性能」。模式要在「实时运行」，配置调试下不跑。

13.3 段表能力

「插补功能」页的段表按顺序执行，显示顺序就是执行顺序，不能按列排序。「＋ 直线段」「＋ 圆弧段」「删除选中」「清空」。



插补通道「功能」页：段表编辑，直线段、圆弧段与动作行，上面一排是演示轨迹

行类型	怎么填
直线	终点 X/Y/Z，段级 F（可空，空则用参数页的路径速度）。
圆弧（圆心式）	终点加 I/J，I/J 是圆心相对段起点在平面内的偏移；终点 = 起点表示整圆。顺时针/逆时针分开。
圆弧（半径式）	终点加 R。弧平面内的弦长超过 2|R| 时这段弧不存在，启动被拒，提示会给出弦长和 R 的下限。
螺旋	平面圆弧加第三轴位移，平面外的位移沿弧长线性分配。
动作行	「＋ 动作 · 输出位」翻一个输出位；「＋ 动作 · 延时」在路径上驻留一段时间；「＋ 动作 · 模拟量」改一路模拟量（典型是随位置改胶量、激光功率）。动作插在选中段之后，没选中加到表尾；表内位置就是触发位置，走到它前一段的路径末端时触发。

说明 **半径式圆弧的起点会变。**前一段是相对段、或进 OP 后当前位置变了，弦长就跟着变。因此弦长超限时软件直接拒绝，不会自动修改 R。

13.4 单位与导入缩放

段表存的是轴的用户单位：轴配成 mm 就是 mm，配成度就是度，脉冲单位的轴才是脉冲。折成脉冲只发生在下发那一步，段表里的数不用自己换算。

给通道一条路径有三种方式。

方式	操作
演示轨迹	「插补功能」页「演示轨迹」旁的「矩形」「圆形」「螺旋」，一键替换当前段表。第一次上手选「螺旋」，立体效果最直观。运行中不能载入，先「停止」。
SCMotion CAD	顶部菜单「工具 → CAD 绘图」打开 SCMotion CAD（独立程序，也可单独运行）。画好图形 →「生成小线段并预览」→「导出段表 .scseg」，回到「插补功能」页点「导入段表」。
手工编辑	「插补功能」页逐行加直线段、圆弧段、动作行。

演示轨迹的大小随轴的单位：轴配成 mm / 度时矩形是 100×60、进给 100；轴是脉冲单位时几何、进给、加减速都按 100 倍放大（矩形 10000×6000 脉冲、进给 10000 脉冲/s），否则伺服上连几分之一圈都不到。硬件通道载入时日志会提醒确认空载和软限位；组内各轴单位不一致时会告警，因子按 X 轴取。

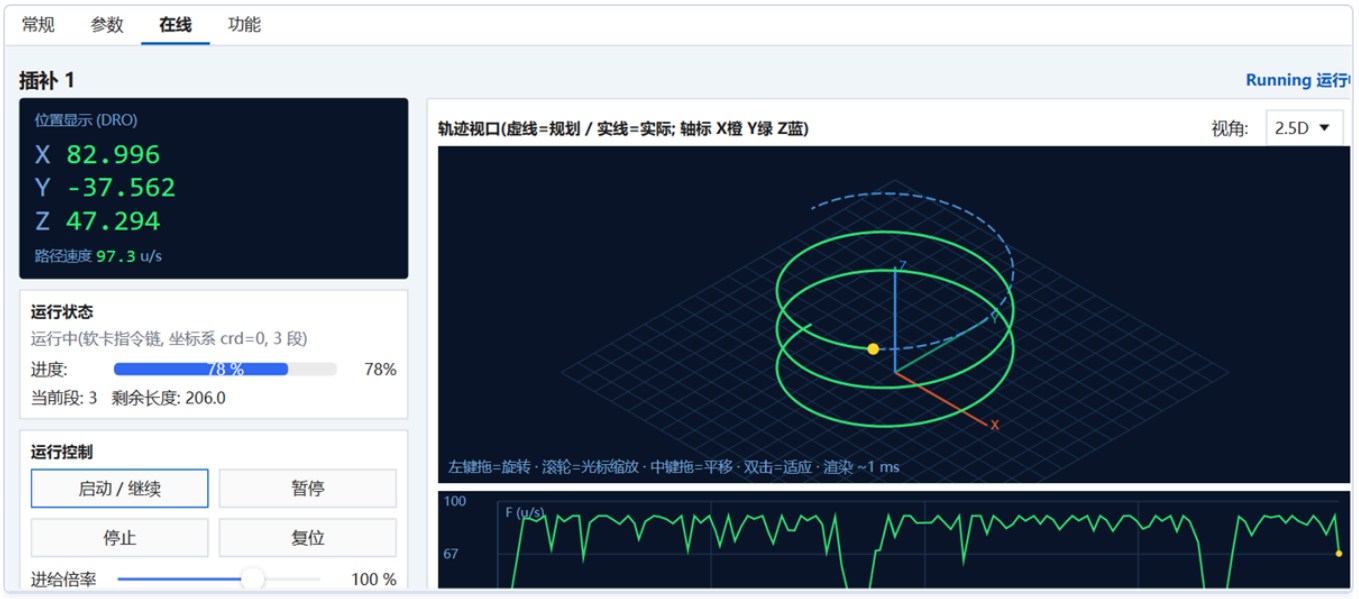
「导入段表」会弹出「导入缩放」框，填的是 CAD 图形单位（mm）→ 段表单位 的系数：

- 轴的运动单位是 mm 或 deg：填 1，CAD 的 mm 就是轴的 mm，换算成脉冲由软件完成。默认值就是 1。
- 轴的运动单位是 pulse：填每 mm 对应的脉冲数。例：10000 脉冲/圈、10 mm/圈 → 填 1000。默认值取通道 X 轴「参数」页 `pulsePerUnit 每 mm 脉冲数(导入/示例用)` 一行的值，出厂为 1000；把这一行按实际改好，以后导入就不用每次改。
- 整幅图形的 X/Y/Z、圆弧圆心、进给一起乘这个系数，圆还是圆。三根轴的系数不一致时告警并按 X 轴统一，圆弧会变形。要正圆，三根轴的运动单位要一致；运动单位是 pulse 时，各轴的 `pulsePerUnit` 一行也要配成相同的值。

> **提示** **保留圆弧。**CAD 里勾选「导出保留圆弧(.scseg v2, 推荐)」，圆和圆弧作为原生圆弧段进段表（一个圆 = 1 段而不是几百条小线段），前瞻规划器直接按圆弧插补，路径更顺、段表更干净。

13.5 运行与观察

切到「插补在线」页，点「启动 / 继续」。



插补通道「在线」页运行中：位置显示、进度、轨迹视口（虚线是规划、实线是实际）与路径速度曲线

按钮	行为
「启动 / 继续」	从头启动；暂停后点它沿剩余路径续跑；单段模式下每点一次跑一段。
「暂停」	沿路径受控减速停住，位置保持。它就是把进给倍率拉到 0。
「停止」	沿路径受控减速到零并结束本次运行，可再次启动从当前位置续跑。
「复位」	收掉运行状态、清除已画轨迹、解组，轴留在当前位置。想从头再来用这个。

- 「进给倍率」滑块 0–150%，运行中实时生效，各轴同步缩放；0% = 进给保持，100% = 编程速度。和 SDK 的 `SCM_GroupSetOverride` 是同一条指令。
- 轨迹视口：虚线是规划路径，实线是实际走过的轨迹，黄点是当前位置；轴标 X 橙、Y 绿、Z 蓝。左键按住拖 = 旋转，滚轮 = 以鼠标位置为中心缩放，中键拖 = 平移，双击 = 视图适应。「视角」下拉一键切 2.5D / 俯视 XY / 正视 XZ / 侧视 YZ。
- 实际轨迹按周期（1 ms）逐点采样，界面刷新慢时轨迹也保持平滑。
- 视口下方是路径速度曲线：精确停止模式下每段之间速度归零；圆角过渡模式下拐角只降速不停车。
- 「运行状态」里有进度、当前段、剩余长度；「运行控制」里进给倍率滑块下面那一行是插补组运行状态（坐标系号、运行阶段、已执行周期数、待执行段数、缓冲余量）。

做完应该看到：DRO 三轴数值联动变化，实线沿规划虚线延伸，运行状态里显示当前段号和进度。

13.6 参数页速查

参数	含义
路径速度 F	默认路径速度，段级 F 优先于它。
路径加速度	各轴加速度上限，前瞻据此规划加减速。
路径减速度	暂停 / 停止时沿路径的受控减速度。

拐角模式	精确停止 = 段间速度归零；连续过渡(圆角) = 按过渡容差插入圆角、拐角不停车。
过渡容差	圆角允许偏离编程轨迹的最大距离。
前瞻段数	向前预读的段数窗口，窗口内反向传播拐角限速使全程平滑。
S曲线(7段, 平滑加减速) + 加加速度	勾选后用七段 S 曲线代替梯形，加减速带圆肩；加加速度限制冲击，只在 S 曲线打开时生效。
单段模式(每段结束等待再次启动)	每段结束停下，等再次点「启动」，用于逐段核对。
映射轴限制	每根成员轴的单位、软限位启用、软限位 −/＋、默认速度、状态。直接写轴参数，立即生效。

13.7 SCMotion CAD 速查

功能	说明
绘图工具	直线、折线（右键或回车结束）、矩形、圆、圆弧（圆心 → 起点 → 终点）、设起点。
编辑	选择工具点选图形，按住拖动移动；Delete 删除；Ctrl+Z / Ctrl+Y 撤销重做。
视图	滚轮缩放、中键拖平移、双击中键视图适应；网格吸附可开关。
Z 参数（2.5D）	选中图形后设起始 / 结束 Z（直线变斜坡），圆设圈数（变螺旋）；3D 预览页签可旋转查看。
导入 DXF	支持 ASCII DXF 的直线、圆、圆弧、多段线（含圆弧边）；二进制 DXF 先在 CAD 软件里另存为 ASCII 格式。
弦差容差	曲线打散成小线段的精度：越小段越密越贴近原曲线。保留圆弧导出时圆弧不受此参数影响。
文件	工程存 .sccad（可再编辑）；给插补的段表导出 .scseg。

13.8 接真轴

和仿真相比只多了"接真轴、复位、使能"几步，画图、段表、运行控制完全一样。

步骤	操作	做完看到
1. 映射	「插补常规」页把 X/Y/Z 各自下拉选到映射到从站的轴，需要的话再加随动轴。	已映射 3/3，状态「就绪(硬件三轴映射, …)」；混了仿真轴时状态写「混合映射」。
2. 复位	点「一键复位映射轴」。	日志逐轴"复位完成"，6041 清除 Fault 位。
3. 使能	点「一键使能映射轴」。	日志逐轴"使能完成"，6041 = Operation Enabled。
4. 给路径	「插补功能」页：手工加段表、载入演示轨迹、或「导入段表」。	「插补在线」页视口显示规划路径。

5. 运行	「插补在线」页点「启动 / 继续」，随时「暂停」「停止」。	真实伺服按路径联动，实线实时跟画。

软件自动做的几件事：

- 防飞车：运动目标只从确认过的真实反馈取，拿不到已确认的位置时拒绝接管，不使用旧值推测，避免轴从当前位置突然冲向 0。
- 软限位：运行中每个周期检查成员轴的软限位，目标超限即按故障停组，并在提示中指出是哪根轴。软限位在「插补参数」页映射轴限制表里改，立即生效。
- 运行中掉使能：任一轴驱动器退出使能，插补按故障停止。
- 收尾：停止、复位或运行结束后释放组通道，之后可回轴页点动。

> **注意** **空载调试。**真实伺服联动前确认空载、每根轴的软限位按机械行程设好、人手远离运动部件。演示轨迹在 mm 工程里是 100×60 的矩形、100 mm/s 的进给，先看行程够不够。

13.9 常见问题

现象	处理
启动提示"请先完成 X/Y/Z 轴映射"或未使能	回「插补常规」页完成映射和使能。使能状态以驱动器 6041 为准，界面上的标志仅供参考。
启动提示"随动轴映射中间有空位"	随动轴从 W0 起依次填，不要跳着填。
启动提示"插补坐标系已全部占满"	先「复位」用完的插补或 CNC 通道，解组后坐标系才归还。
启动提示"建组失败: …"	冒号后面就是原因，常见是某根成员轴没使能、被别的通道占着、或模式不在「实时运行」。
启动提示"半径式圆弧无解"	那段弧的弦长超过 2|R|。改 R 或改终点；起点是上一段终点，前一段是相对段时起点会跟着变。
复位后位置不是 0	复位本来就不动轴（轨迹会清掉）。要把位置清零：仿真轴在轴页「位置定义」定义为 0；真轴回零去轴页「启动回原点」或客户程序 `SCM_AxisHome`，见第12章「回零」。
导入 DXF 提示跳过实体	SPLINE / ELLIPSE / 文字不支持，在 CAD 软件里先转成直线、圆弧、多段线再导出。
运行中提示"插补组已减速停住：…的插补目标位置…超出软限位"	路径超出某根轴行程。缩小图形、改导入系数，或到「插补参数」页放宽该轴软限位（确认机械允许）。
提示"实际位置反馈未确认(防飞车)"	界面还没刷到当前位置。等一两秒轴页显示真实位置后重试，此提示为保护性拦截，不表示故障。

真轴动作很小或很慢	轴是脉冲单位时段表数值就是脉冲。用演示轨迹（自动 ×100），或导入时把系数填成 脉冲/mm。
提示"组内各轴单位不一致"	演示轨迹的几何因子只按 X 轴取，另几根轴的行程对不上。把各轴单位配成一致，或换成手工段表。

13.10 下一步

- 单轴怎么使能、复位、停止，看第11章「轴运动与安全边界」。
- 真轴回零看第12章「回零」。
- 用 G 代码代替段表，看第14章「CNC 与 G 代码」。
- 遇到报错查第23章「故障排查」。

第14章 CNC 与 G 代码

CNC 通道解释 G 代码程序，驱动 3 轴（可加 A/C 旋转轴做 4/5 轴联动），带前瞻规划、进给倍率、刀具半径补偿、五轴 RTCP 和实体三维机床仿真。仿真轴和硬件轴都能运行全部功能，硬件轴需先激活运行并使能。

14.1 建通道

在工程树的 NC 任务节点上右键 →「添加」→「添加CNC通道」，生成 CNC 子节点。四个页签：「CNC常规」「CNC参数」「CNC在线」「CNC功能」。

14.2 第 1 步：映射坐标轴

- 「CNC常规」页点「一键创建仿真三轴映射」建 X/Y/Z。程序用到 A 字或 RTCP 时再点「＋A轴(第4轴)」「＋C轴(第5轴)」，建出来就是角度轴。
- 配好后点「一键复位映射轴」，再点「一键使能映射轴」。硬件三轴要在实时运行下做这两步。
- 轴方向要反向在轴页设「方向取反」，对界面和软卡 SDK 发出的所有运动都生效。

> **说明** **硬件轴的范围。**硬件轴与仿真轴能运行的程序相同：A/C 旋转轴、G4 驻留、单段和 RTCP 都可以在硬件轴上运行；硬件轴需先激活运行并使能。

14.3 第 2 步：写 / 载入 G 代码

在「CNC功能」页：「示例程序」一键填入范例，或「载入 .nc」打开程序文件，编辑后「编译」。编译报错进「编译错误(双击定位到程序行)」表。写完可「保存 .nc」；「RTCP演示」载入一段五轴范例（要映射 A/C 轴）。

通道的轴都映射到真实从站时，「示例程序」载入按 20 mm 见方编的 XYZ 范例，并把「CNC参数」页的「默认进给 F」「快移 G0」「路径加速度」「路径减速度」「加加速度」改成与范例配套的值。X 轴运动单位是 mm 或 deg 时按原样；是 pulse 时坐标和这几项参数都乘该轴 `pulsePerUnit 每 mm 脉冲数(导入/示例用)` 一行的值（出厂 1000），先按实际的每 mm 脉冲数改好这一行再载入。



CNC 通道「功能」页：G 代码程序与编译错误表，上面是方言说明和编译、载入、示例程序按钮

完整的 G/M 码支持范围见下面「支持的 G 代码 / M 代码」两张表。SPLINE 等高级实体在 CAD 侧先转直线 / 圆弧。

14.4 支持的 G 代码

下表是本通道全部认得的 G 代码。表里没有的一律编译报错（不是忽略），错误带行号进「编译错误」表。

代码	模态	作用与注意
`G0`	模态	快移。速度取参数页的「快移 G0」，**不使用 F**。刀补生效期间禁用 G0。
`G1`	模态	直线插补，速度取 F（没写 F 时用参数页的「默认进给」）。
`G2` / `G3`	模态	顺圆 / 逆圆。圆心式用 `I/J/K`，半径式用 `R`（正 R = 劣弧，负 R = 优弧，**不允许整圆**）。
`G4 P`	单次	驻留。**P 的单位是秒**（`G4 P1.5` = 停 1.5 秒）。路径会先减速到该点停稳再计时。**不能写在程序第一行**——驻留在前面已下发路径的终点处执行，因此前面必须先有一条移动。
`G17` / `G18` / `G19`	模态	圆弧平面 XY / ZX / YZ，默认 G17。G18 的 G2/G3 方向按 ISO 标准的 ZX 平面定义。⚠ 4.3.48 之前版本的 G18 圆弧方向与此相反（会走成互补弧，R 编程时圆心落在弦的另一侧），在这些版本上编写的 G18 圆弧程序请重新核对。⚠ 见下方「已知限制」第 2 条。

`G32`	模态	等螺距螺纹切削（C 轴主轴形态：把一根伺服当主轴放进本通道做 C 轴，主轴转角按导程算出来）。**本行的 `F` 是导程**（长轴每转走多少），**不会改掉进给模态**——路径进给仍是上一次 `G1` 给的 F。导程本身也是模态，连着几刀只写一次即可。要求 `G18` 平面，且通道在参数页配了「主轴轴」（要与 C 轴选同一根），否则**编译报错**。主轴转角是算出来的，所以本行**不许再写 `C`**（`Y` / `A` 也请分行写）。⚠ 段两端一定有加减速，请自己在程序里留够引入/引出长度（不自动加）。⚠ `X` 一律**半径值**（本通道没有直径编程）。⚠ 进给倍率**不必锁 100%**——倍率只缩放时间，导程由几何保证不变。⚠ 旋向由 C 轴正方向与 Z 正方向的固定关系决定，换旋向要改 C 轴的正方向配置（没有主轴反转可用）。⚠ 每刀退刀后请把主轴转回起始角（例 `G0 C0`）：累计转角超过 200 转会**编译报错**。
`G92` / `G78`	模态	螺纹切削循环。**本方言的 `G92` 是螺纹切削循环，不是工件坐标系设定**——本通道没有工件坐标系；`G78` 是同一个循环的另一个码号，**完全等价**。格式 `G92 X_ Z_ [R_] F_`，一个块展开成四段：X 向快进到切削起点 → Z 向螺纹切削（含锥度、含退尾）→ X 向快退回起点 X → Z 向快退回起点 Z**并把主轴转回起刀角**。`R` = **终点半径 − 起点半径**（带符号，0 或省略 = 直螺纹），⚠ 这里 R 的符号约定与 FANUC 手册中的 G92 相反，请勿直接套用其图示。01 组模态：再给一个新的 `X` 就是下一刀（`Z` / `R` 与导程都留着）。`F` 与 `G32` 一样是**导程**，不改进给模态。退尾量在参数页的「螺纹退尾量」里配（单位 = 0.1 倍导程，0 = 不退尾，对应 SDK 的 `SCM_CNC_PARAM_THREAD_CHAMFER`），退刀方向按起点 X 在螺纹的哪一侧自动判。⚠ 第四段把主轴转回起刀角是为了保证螺纹**正确**：每刀的起刀相位必须一致，否则第二刀会车出第二条槽；行程恰好是整数圈时不易察觉，改变长度就会错牙。其余限制（G18、主轴轴、半径值、引入引出、旋向）同 `G32`。
`G76`	单次	复合螺纹切削循环（FANUC 0i 两行式，**两行必须紧邻**，中间隔任何一条有内容的行都会编译报错；空行与整行注释不算）。 `G76 P(m)(r)(a) Q(dmin) R(d) ;` `G76 X_ Z_ R(i) P(k) Q(d1) F(L) ;` 第一行的 `P` 是**6 位定长打包**：前 2 位精加工重复次数(01~99) + 中 2 位倒角量(00~99，单位 0.1 倍导程) + 后 2 位刀尖角(00/29/30/55/60/80)，例 `P010060`。`Q(dmin)` 最小切深、`R(d)` 精加工余量；第二行 `P(k)` 牙高、`Q(d1)` 第一刀切深——这四个都是**半径值、微米整数**（=0.001 用户单位；本通道没有 G20/G21，所以只有当轴的用户单位是 mm 时它才真的是微米）。`R(i)` 是锥度半径差（用户单位，0 = 直螺纹），`F` 是导程。切深按**等切削面积法**分配：第 n 刀累计切深 = d1×√n，实际切深不足 dmin 时钳到 dmin，累计到 k−d 为止（最后一刀钳到刚好 k−d），之后在**全深 k** 上重复 m 次精加工（不是每次切 d/m）。进刀按刀尖角斜向单边（刀尖角 00 = 直进法）。⚠ 车外螺纹还是内螺纹由**起点 X 在螺纹底径哪一侧**判定，所以下 `G76` 之前要先把刀退到工件之外。⚠ `G4` 与 `G76` 不能同行（两者都用 `P`）。
`G40`	模态	取消刀具半径补偿。程序结束前必须取消，否则报错。
`G41` / `G42`	模态	左 / 右刀补，`D` 给的是**半径值**（不是刀具表号）。能力边界见下方「刀补的限制」。
`G43.4`	模态	五轴 RTCP 开（双转台 AC）。开启后 XYZ 编的是**工件系刀尖点**，A/C 是转台角。与刀补互斥。
`G49`	模态	RTCP 关。
`G90` / `G91`	模态	绝对 / 增量编程。⚠ **不影响圆弧的 I/J/K**——它们恒为「圆心相对本段起点」的增量（与 FANUC 默认一致，但本通道没有 G90.1/G91.1 可切）。

地址字：`X Y Z A C`（坐标）、`I J K`（圆心偏移）、`R`（半径）、`F`（进给）、`S`（主轴转速，模态，须大于 0）、`P`（驻留秒数 / 子程序号 / 复合螺纹循环的 6 位打包参数与牙高）、`Q`（复合螺纹循环的切深与最小切深（半径值，微米整数）；只用于该循环，写在其他地方会报错）、`N`（行号）、`D`（刀补半径）、`L`（子程序调用次数）、`O`（子程序标签）、`G`（准备功能，见上面「支持的 G 代码」表）、`M`（辅助功能，见上面「支持的 M 代码」表）。注释用 `;` 到行尾或 `(...)` 行内，`%` 行忽略。

14.5 支持的 M 代码

代码	作用
`M0` / `M1`	程序停 / 选择停：路径走到停点停稳，等「循环启动」（SDK 是 `SCM_CncCycleStart`）续跑，从停点原地续。**必须独占一行**（只许再带 N 行号）。`M1` 受「选择停」开关控制——关（默认）= 视为注释跳过；开 = 同 `M0`。开关运行中改**即时生效**。M0/M1 不碰任何输出。
`M3` / `M4` / `M5` + `S`	主轴正转 / 反转 / 停 + 转速。落到**缓冲输出**（与运动同步出线：`G1 X10 M3` = 走到上一段末端先转主轴再走刀，M 一律先于同行运动）。方向位与 S 模拟量通道在参数页「M 功能输出映射」配置；`模拟量工程值 = S × 增益`，超过「S 上限」时编译报错。**程序用到这些 M 代码但未配置映射时，编译报错**（不会跳过该指令继续运行）。程序结束（M2/M30）自动主轴停；操作员停止/复位/组故障时立即关断。
`M8` / `M9`	冷却开 / 关。同上落缓冲输出，冷却位在参数页「M 功能输出映射」配置；程序结束自动关，停止/复位/故障立即关断。⚠ 配完点参数页「检查映射」核一遍，它会列出未配置、重复配置或配到不存在通道的位。
`M2` / `M30`	程序结束。两者**行为完全相同**：停止投递后续行。**不做**倒带、不复位模态、不停主轴、不关冷却。
`M98 P<n> [L<次数>]`	调用子程序。`L` 默认 1，`L0` = 跳过；嵌套上限 8 层。
`M99`	子程序返回。该行只许出现 M 与 N 字；在主程序里用会报错。

> **说明** **子程序。**用 `O<n>` 独占一行定义（通常放在 M30 之后）。模态跨子程序双向穿透：F、G90/G91、平面、刀补、D 半径、RTCP、当前位置全部带进带出；子程序里写 M2/M30 会终止整个程序。

14.6 不支持的常用代码（会编译报错，不是忽略）

下面这些是其他数控系统中常见、但本通道没有的代码。程序里出现它们会在编译时报错并给出行号，不会被悄悄跳过。

缺的	怎么办
`G54` ~ `G59` / `G53` 工件坐标系	⚠ 注意：本方言的 `G92` **是螺纹切削循环**，不是别处常见的「工件坐标系设定」，见上表。**本通道没有「程序零点」这个概念**：程序起点直接取轴的当前位置。请先回零、或用「位置定义」把机床对到程序零点，或改用 `G91` 相对编程。⚠ 程序坐标系离机床当前位置太远时，**循环启动会当场拒绝**（不会以一条巨大的快移开场）。

`M6` 换刀 + `T` 刀号 + `H` 刀长号	没有。
`G43` / `G44` 刀长补偿	没有。⚠ `G43.4` 是 RTCP，与刀长补偿无关，请勿混淆。
`G81` ~ `G89` / `G73` / `G74` 固定循环、`G98` / `G99`	没有。钻孔/攻丝/镗孔请在 CAM 里展开成直线与圆弧。⚠ 螺纹的两个循环**是有的**，见上表的 `G92` / `G78` 与 `G76`。
`G28` / `G30` 回参考点	没有。回零走轴页或软卡 SDK。
`G33` 螺纹（铣削系统中的同义码）	没有。本方言的螺纹码是 `G32`（车床写法）；循环是 `G92` / `G78` 与 `G76`。语义见上表。
`G20` / `G21` 英制/公制	没有。单位由轴的当量决定。
`G94` / `G95` 进给模式	没有，见下方「已知限制」第 1 条。
`G61` / `G64` 精确停 / 连续切削	没有。改用参数页的「拐角模式」。
宏变量 `#100=…`、条件跳转、`G65` / `G66` 宏调用	没有。`M98` 只接受 P 与 L，**不能传参数**。

14.7 已知限制（从其他数控系统迁移程序时需特别注意）

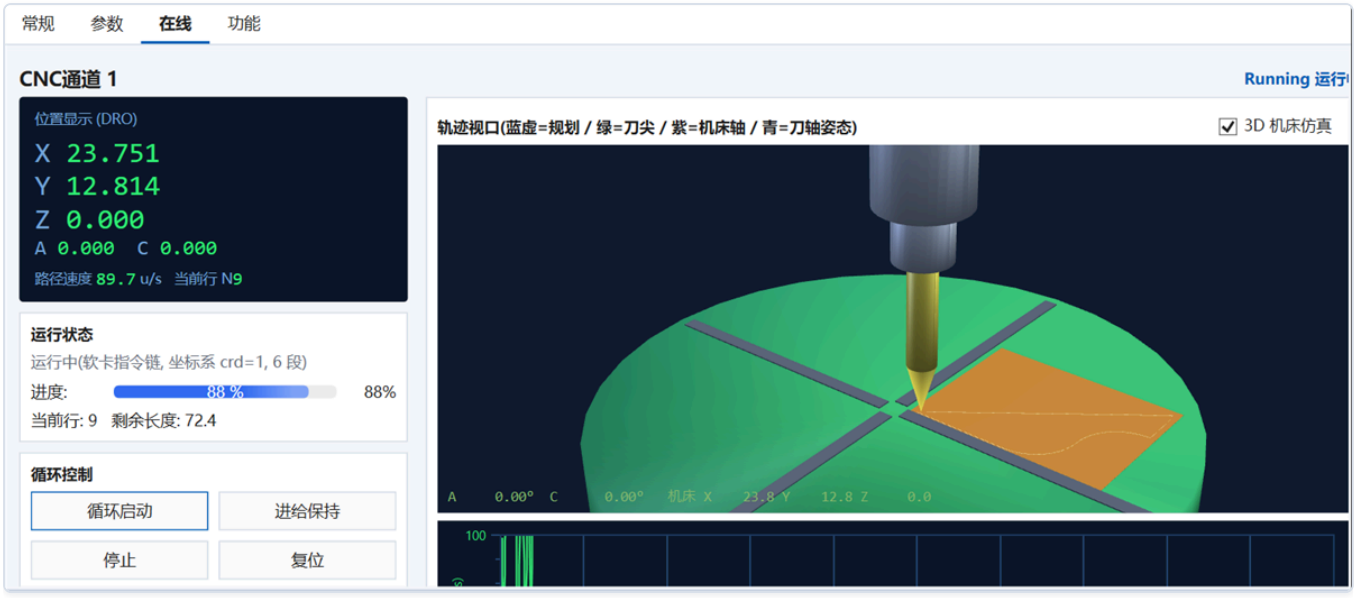
1. `F` 的单位是「用户单位 / 秒」，不是 mm/min。这是与常见数控系统最大的差异：直接沿用其他系统程序里的 `F100`，实际速度差 60 倍。轴配成 mm 时 `F100` = 100 mm/s。没有 G94/G95 可切。
2. `G18` / `G19` 平面的圆弧，三轴当量不同时圆心会算错。圆弧在脉冲空间下发，只有「平面内两轴当量相同、方向相同」时才严格成立。三轴当量一致（绝大多数机床）时不受影响。
3. 位置显示(DRO)里的「当前行」是程序的实际行号，不是你写的 `N` 字。`N` 只作注释用，不参与定位。
4. `G0` 不是真正的快移：它是按参数页「快移 G0」速度走的协调直线，没有「各轴独立全速定位」的语义。
5. 缺 `M2` / `M30` 的程序照跑不报警，最后一段走完即停。
6. 螺纹循环 `G32` / `G92` / `G76`（写法见上面的 G 代码表）：按半径值编程；要先选 `G18` 平面，并在「CNC参数」页「车螺纹」里选好「主轴轴(C 轴主轴)」，它必须与常规页的 C(第5轴) 是同一根轴；累计转角上限 200 转；选了主轴轴之后 M3/M4/M5 一律编译报错，I/O 主轴与 C 轴主轴不能并存。

14.8 刀补的限制（`G41` / `G42`）

刀补在编译期把编程轨迹变换成刀心轨迹，下列情形会在编译时报错并给出行号：

- 圆弧只支持 `G17` 平面；刀补态下带 Z 的螺旋弧不支持。
- 内拐角只做直线-直线求交：任一侧是圆弧就报「内拐角含圆弧时不支持刀补」。
- 相邻段长度短于刀具半径报过切；内侧圆弧半径不足报过切。
- 刀补态下禁用 `G0`。

14.9 第 3 步：运行与观察



CNC 通道「在线」页运行中：位置显示、进度与三维机床仿真，刀尖沿路径切出刀痕

- 切到「CNC在线」页，「循环控制」面板点「循环启动」。「进给保持」沿路径减速暂停，再点「循环启动」续跑；「停止」结束；「复位」收状态、清轨迹，轴留在原地。
- 「倍率」滑块 0–150% 在线调进给速度，各轴同步缩放；0% = 冻结保持，G4 驻留时间不受影响。勾「单段模式」逐段执行，下个循环生效；勾「选择停(M1)」时程序里的 M1 同 M0 程序停，不勾则视为注释跳过。
- 「位置显示 (DRO)」给 X/Y/Z/A/C、路径速度、当前行（程序的实际行号）。RTCP 会话里 X/Y/Z 给的是工件系刀尖点，不是机床轴。
- 主视区默认勾着「3D 机床仿真」，显示实体机床 + 毛坯 + 刀路 + 刀痕，双转台 AC 随 A/C 轴实时运动；取消勾选切回 2.5D 轨迹视口。CNC 页只有 2.5D 这一种视角。
- 下方是路径速度-时间曲线：精确停止时段间归零，圆角过渡时拐角只降速不停车，G4 驻留是零速平底段。

14.10 用程序跑它（软卡 SDK）

这套 G 代码通道也可以不碰界面，由客户程序驱动：软卡 SDK 的 `SCM_Cnc*` 一族提供了打开通道、载入、编译、循环启动、进给保持、倍率、单段、状态轮询，与这一页上的按钮一一对应。指令逐条说明见《软卡 SDK 编程手册》「软卡 SDK 指令说明」。

- G 代码在客户程序里解释，由组通道执行，通道号就是 `crd`。
- 脉冲当量由 `SCM_CncOpen` 逐轴向主软件取；取不到又没显式设时，`SCM_CncCycleStart` 拒绝启动，不会默认按 1.0 跑。
- 通过 SDK 运行的 G 代码，方言与本页完全相同，使用的是同一个解释器，本页的几张表对 SDK 客户同样有效。

14.11 参数页

「CNC参数」页的「进给与路径」（默认进给 F / 快移 G0 / 路径加减速度 / 过渡容差 / 拐角模式 / 前瞻段数 / S 曲线与加加速度 / 默认绝对坐标 G90 / 单段模式）与插补通道同一套含义，见第13章「插补通道」。CNC 独有的几组：

- 「五轴 RTCP 机床构型」：双转台 AC，`G43.4` 生效；填「刀长 L」「A轴线 Ta」「C台中心 Tc」。
- 「M 功能输出映射」：M3 / M4 / M8・M9 各对应哪个从站的哪一位，S 转速走哪个模拟量通道、增益和上限。
- 「车螺纹」：「主轴轴(C 轴主轴)」和「螺纹退尾量」。
- 「映射轴限制」：每轴软限位与默认速度上限。

第15章 电子齿轮

电子齿轮让从轴按固定齿比跟随主轴，等效机械齿轮箱，但齿比可以在线改。这章说怎么建通道、选主从轴、设齿比与追赶参数、啮合与脱开，以及如何判断完成。

15.1 建通道

在工程树的 NC 任务节点上右键 →「添加」→「添加电子齿轮」，生成齿轮子节点。三个页签：「齿轮常规」「齿轮参数」「齿轮在线」。

15.2 第 1 步：选主从轴并使能

「齿轮常规」页的映射表有主轴、从轴两行，候选来自本 NC 任务下的轴。每行显示轴的使能状态、当前位置和类型。

- 主轴就是一根普通轴，映射轴或仿真轴都行，没有单独的"虚主轴"设置。任何一根轴都能当主轴。
- 没有现成从轴时点「一键创建仿真从轴」。
- 点「一键复位」，再点「一键使能」。这两个按钮对主轴和从轴一起下发：硬件轴走驱动器故障复位序列 / 使能序列，仿真轴清状态。
- 从轴选映射轴时经软卡指令链执行，与客户程序的 `SCM_GearIn` 走同一条链；选未映射的虚拟轴时由页面仿真。

15.3 第 2 步：设齿比与追赶参数

「齿轮参数」页的「齿比与啮合动力学」：

参数	含义
齿比分子 / 齿比分母	从轴速度 = 主轴速度 × 分子 ÷ 分母。分子填负数 = 从轴反向；分母必须大于 0。
追赶加速度	单位 u/s²。啮合、脱开、改齿比时从轴速度每秒最多变化多少。大 = 追得快但冲击大，小 = 过渡平缓但追得慢。
追赶加加速度	单位 u/s³。0 = T 型速度曲线，直上直下；正数 = S 型，两端带圆角，圆角时长 = 追赶加速度 ÷ 这个值。
同步速度窗 / 同步位置窗	速度差和位置差都进窗才算「已同步」；同步后位置偏差超出窗会自动重新追赶。

同一页的「主轴驱动参数」（加速度 / 定速设定 / 点动速度）就是主轴那根轴在轴页配的参数，在这里修改与在轴页修改的是同一组参数。在线页「主轴驱动」用的就是这几个值。

所有参数改完立即生效，不用重新啮合。

15.4 第 3 步：啮合并观察

- 切到「齿轮在线」页，用「主轴驱动」面板的「定速运行」/「点动 +」/「点动 −」/「停止」让主轴动起来。它驱动的是常规页选中的那根主轴，映射轴按 `SCM_AxisMoveVel` / `SCM_AxisStop` 走，未映射轴由页面仿真计算。
- 点「啮合 GearIn」，从轴按齿比追随主轴；「脱开 GearOut」沿追赶加速度减速脱开。
- 顶部回显主轴 / 从轴的位置与速度，以及「速度差」「位置差」；进入同步窗后两者趋近 0。
- 运行中在「齿比在线修改」改分子 / 分母，修改即生效：状态从「已同步」短暂回到「啮合中」，从轴按追赶加速度滑到新速度，再回「已同步」。这是正常现象。
- 右侧「主/从轴速度曲线」反映跟随质量，两条曲线重合即表示跟随良好。



电子齿轮「在线」页：主从轴位置与速度、速度差与位置差，右侧是主从轴速度曲线

15.5 完成条件与状态

啮合与脱开都是异步受理，按钮按下（或 `SCM_GearIn` / `SCM_GearOut` 返回 0）只表示命令已受理。完成看状态，SDK 客户轮询 `SCM_GearGetState`：

状态	码	含义
DECOUPLED	0	未耦合，也是脱开的完成态
SYNCHRONIZED	2	已同步，啮合的完成态
FAULTED	4	从轴掉使能或故障，耦合被解除。该状态保持到下一次啮合时才清除

- 轮询的收尾条件要同时判目标态和 FAULTED，只等目标态遇到故障会一直等下去。
- 已耦合的从轴对同一根主轴再发一次啮合 = 在线改齿比和参数；要换主轴必须先脱开，否则被拒。一根从轴同时只有一个耦合，一主带多从就对每根从轴各啮合一次。
- 仿真模式下啮合瞬时完成，状态只看到 0 → 2，看不到中间的追赶过程。

> **说明** **提示。**「脱开 GearOut」是减速脱开，不是急停。要立即断开用轴页的停止或 `SCM_AxisStop`。

第16章 电子凸轮

电子凸轮让从轴按一张"主轴相位 → 从轴位移"的表跟随主轴，等效机械凸轮但随时能改表。这章说怎么建通道、编表、啮合、换表，以及表的下发规则和完成条件。

16.1 建通道

在工程树的 NC 任务节点上右键 →「添加」→「添加电子凸轮」，生成凸轮子节点。三个页签：「凸轮常规」「凸轮参数」「凸轮在线」。

16.2 第 1 步：映射主从轴并使能

- 「凸轮常规」页选主轴 / 从轴映射，候选来自本 NC 任务下的轴；没有轴时点「一键创建仿真主从轴映射」。
- 点「一键复位映射轴」，再点「一键使能映射轴」。硬件轴走驱动器故障复位，仿真轴清状态。
- 页面顶部一行显示端口 / 表点数 / 模值 / 啮合模式和耦合状态。
- 任何一根轴都能当主轴，没有单独的"虚主轴"指令。映射轴当主轴时用它轴页的速度运行来驱动（`SCM_AxisMoveVel`）。
- 从轴映射到真实驱动器时，在「实时运行」下、轴已使能就能啮合，走的是与 SDK `SCM_Cam*` 同一条指令链；从轴是仿真轴时在页面里仿真。

16.3 第 2 步：编辑凸轮表

「凸轮参数」页上半是「耦合参数」，下半是「凸轮表编辑器」。

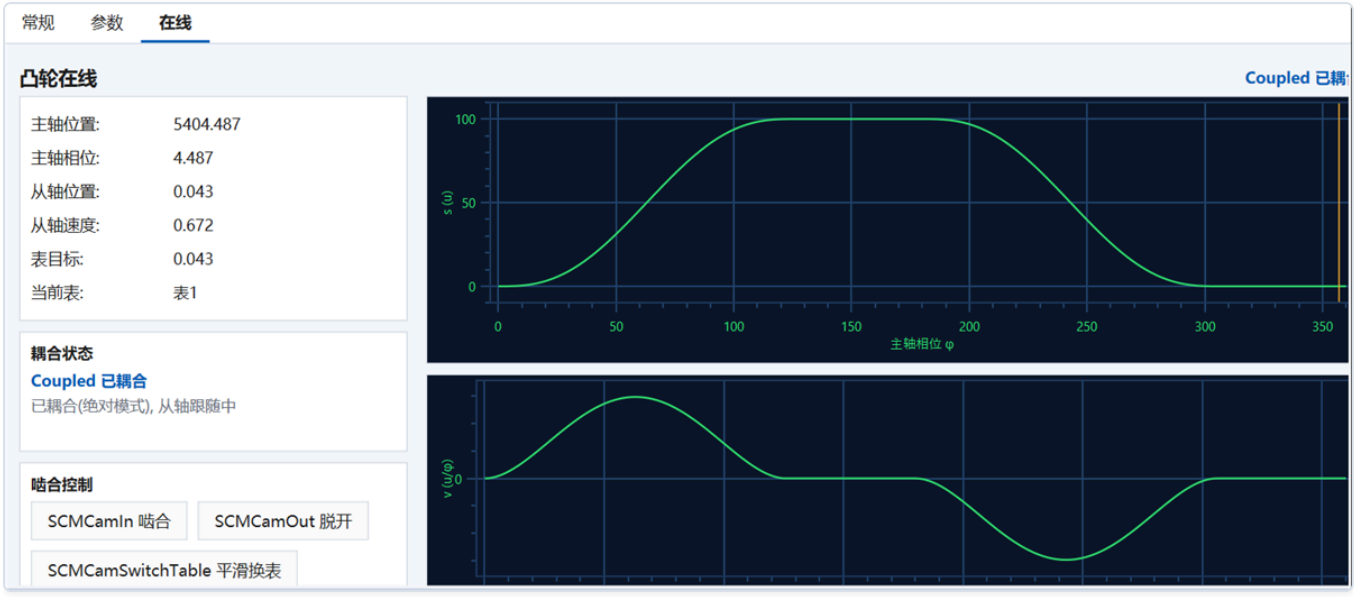
耦合参数	含义
主轴模值(周期)	主轴走完一个周期的位移，表的相位按它取模。
啮合模式	绝对 = 从轴走到表定义的绝对位置，有偏差时按追赶速度限速逼近；相对 = 从啮合位置起算，无跳变；附加 = 在啮合位置上叠加表值。
相位偏移 / 幅值缩放	整张表沿相位平移、沿位移缩放。
追赶速度	单位 u/s，绝对模式啮合有偏差时的逼近速度上限。
切换过渡宽度	平滑换表时两张表混合的主轴相位跨度。

「凸轮表编辑器」支持多张表，编辑的是选中表：

- 「凸轮表」下拉选表，「表名」可改，「新建表」「删除表」管理多张表。
- 「添加关键点」「删除选中」「清空」编辑关键点；第一次上手点「示例表(升-停-回-停)」。
- 表格三列：「主轴相位」「从轴位移」「段曲线(到下一点)」。段曲线可选直线、五次多项式、摆线、修正梯形；摆线 / 修正梯形在停-走衔接处速度加速度连续，冲击最小。末点自动闭合回首点形成循环表。
- 一张表最多 256 点。改表即时重算在线页的三条预览曲线。

16.4 第 3 步：啮合并观察

- 切到「凸轮在线」页。主轴是纯虚轴时，用「主轴定速测试(仅纯虚轴主轴)」面板填速度、点「启动」让主轴转起来，「停止」停下。这个面板只对纯虚轴主轴有效，映射轴当主轴要去轴页驱动。
- 点「SCM_CamIn 啮合」，从轴开始按表跟随；「SCM_CamOut 脱开」沿当前速度减速停在当前位置。
- 顶部回显主轴位置 / 主轴相位 / 从轴位置 / 从轴速度 / 表目标 / 当前表；「耦合状态」面板给状态和原因。
- 右侧三条曲线（位置 / 速度 / 加速度）实时反映跟随质量。



电子凸轮「在线」页：主轴相位与从轴位置、耦合状态，右侧是凸轮表的位置与速度曲线

16.5 表是怎么下发的

凸轮表逐点下发：开始一张表 → 逐点加 → 结束（SDK 里是 `SCM_CamTableBegin` → `SCM_CamTableAddPoint` → `SCM_CamTableEnd`）。页面点「SCM_CamIn 啮合」时会先把选中表按这个顺序发下去再啮合。

- 未结束的表不会生效；结束时对整表校验，校验失败则保留原表。
- 正在被啮合使用的表不能重定义。要改曲线，定义到另一张表，再切过去。
- 啮合时只看选中表；未耦合时改表，下次啮合生效。

16.6 运行中换表

耦合运行中点「切换到选中表(平滑)」（参数页）或「SCM_CamSwitchTable 平滑换表」（在线页），把选中表平滑换上去：在「切换过渡宽度」的主轴相位跨度内混合两张表，位置与速度都连续。

- 过渡随主轴相位推进，主轴不动就不推进。
- 还在追赶啮合时会被拒，等状态到「已耦合」再切。
- 模值不同的表不能切，要换模值先脱开。

16.7 完成条件与状态

啮合、脱开、换表都是异步受理，返回 0 只表示命令已受理。完成看状态，SDK 客户轮询 `SCM_CamGetState`：

状态	码	含义
DECOUPLED	0	未耦合，也是脱开的完成态
COUPLED	2	已耦合，从轴每个周期按表跟随，啮合的完成态
FAULTED	4	从轴掉使能或故障，耦合被解除。该状态保持到下一次啮合时才清除

- 轮询的收尾条件要同时判目标态和 FAULTED。
- 已耦合的从轴对同一主轴、同一张表、同一模式再发一次啮合 = 在线改相位偏移 / 幅值缩放 / 追赶速度。换表用平滑换表；换模式或换主轴要先脱开。
- 仿真模式下啮合瞬时完成，状态只看到 0 → 2。

> **说明** **提示。**脱开是减速脱开，不是急停。要立即断开用轴页的停止或 `SCM_AxisStop`。

16.8 文件

在线页「凸轮表文件(.sccam)」面板：「导出…」存选中表；「导入…」新建为一张表并选中；「示例表」载入示例。用它把凸轮表在工程之外单独存取与复用。

第17章 机器人总览

三种机器人构型共用的规则都在这一章：怎么建通道、页面与 SDK 怎么对应、单位与位姿约定、三种模式下的差别。各构型自己的机构参数和页面操作见第18章「Delta 机器人」「SCARA 机器人」「6 轴串联机器人」。

17.1 三种构型

在工程树的 NC 任务节点上右键，用「添加Delta机器人」「添加SCARA机器人」「添加6轴串联机器人」建通道。构型在建的时候定，建好后不能改。每种通道四个页签：常规 / 参数 / 在线 / 功能，页签名带构型前缀，如「SCARA机器人常规」。

构型	关节	能控制的分量
Delta（3 轴并联）	三根主动臂 θ1 / θ2 / θ3	位置 X/Y/Z，没有姿态
SCARA（4 轴）	J1/J2 回转、J3 升降、J4 腕回转	位置 X/Y/Z + 末端朝向 C
6 轴串联	J1~J6 全部回转	位置 X/Y/Z + 姿态 C/B/A

17.2 常规页的三个按钮

三种构型的常规页都有「一键创建仿真映射」「一键复位映射轴」「一键使能映射轴」。没有轴可用时先点第一个，软件按构型建好对应根数的仿真轴并映射。运动前按「一键复位映射轴」→「一键使能映射轴」的顺序点：硬件关节走驱动器故障复位，仿真关节只清状态。

使能和复位都是异步受理，按钮按下只表示收下了。完成看常规页的通道状态变成「就绪」；全部关节映射并使能后通道才就绪。

17.3 页面与 SDK 是同一条链

页面上能点的运动，SDK 客户程序都能发，走的是同一条指令链。日志里每次操作打一行 `SCM_Robot*(…) → 码`，码值与客户程序拿到的返回值相同。指令逐条说明见《软卡 SDK 编程手册》「软卡 SDK 指令说明」。

页面操作	SDK 指令
「一键使能映射轴」「一键复位映射轴」	`SCM_AxisPower` / `SCM_AxisReset`，按轴发
「启动 PTP」	`SCM_RobotMoveJ`，关节空间同步走
「启动 LIN」「矩形演示」	`SCM_RobotMoveL`，TCP 走笛卡尔直线
关节点动 / 笛卡尔点动	`SCM_RobotJogJoint` / `SCM_RobotJogCart`
「停止」	`SCM_RobotStop`
示教点「回放」	`SCM_RobotTableBegin` → `SCM_RobotTableAddPoint` → `SCM_RobotTableEnd`，再 `SCM_RobotRunTable`

在线页的状态与读数	`SCM_RobotGetState`

17.4 单位与位姿约定

- 关节量 = 该关节轴的用户单位。回转关节的轴要配成「度」，SCARA 的升降关节 J3 配成毫米。配错会在通道启动时打一条警告。
- 笛卡尔量一律 mm 和度。位姿顺序固定 (x, y, z, c, b, a)：c 绕 Z（偏航）、b 绕 Y（俯仰）、a 绕 X（横滚），ZYX 欧拉角。
- `SCM_RobotMoveJ` 的目标是绝对关节位置，不是增量。
- 本构型用不到的分量必须传 0：Delta 的 j4~j6 和 c/b/a，SCARA 的 j5/j6 和 b/a。传了非零值整条指令被拒，不会悄悄丢掉。

17.5 受理、完成与故障

- 运动指令是异步受理：返回 0 只表示收下了，完成的标志是通道状态回到空闲。页面在线区自动轮询；SDK 客户轮询 `SCM_RobotGetState`，收尾条件要同时判空闲和故障，只等空闲遇到故障会一直等下去。
- 故障（逆解失败、超软限位、驱动器退使能、外部停止）后通道状态保持故障态，一直保持到下一次运动指令受理时才清除。驱动器本身的故障复位走「一键复位映射轴」或 `SCM_AxisReset`。
- 直线运动的逆解固定取一支臂形。当前臂形不一致时，先用 PTP（或示教点「GoTo」）把手臂带到目标附近，再走 LIN。各构型的臂形约定见各自章节。

17.6 点动是步进制

在线页左半按关节点动、右半按 TCP 点动。每按一下走一个步距，按住连续发。步距在参数页的「关节步距」（Delta 叫「臂角步距」）和「笛卡尔步距」里设。逆解不可达的那一步被拒且不动作；通道在跑别的运动时点动被拒。含硬件关节的通道同样可以点动。

17.7 三种模式下的行为

模式	机器人行为
仿真	能跑全部运动（PTP / LIN / 点动 / 示教回放），与真机同一条指令链，没有真实回零开关。SDK 客户程序离线开发也连这个模式。
配置调试	不开放机器人运动指令，运动操作会被拒绝。
实时运行	映射了真实从站的关节驱动真机；没映射硬件的关节作为虚关节参与运动学、不驱动硬件，两种可以混搭。

> **注意** **危险。**硬件通道第一次运动前，先在轴参数页确认每根关节轴的软限位已启用，手放在急停上，先用小步距点动。

第18章 Delta 机器人

Delta 通道把三根回转轴组成一台三轴并联机器人：三根主动臂各驱动一条臂的摆角，平行四边形从动臂带着动平台平动。动平台姿态恒定，所以只有 X/Y/Z 三个平动自由度，没有腕轴和姿态角。这章只写 Delta 特有的机构参数、工作区和软限位；建通道、使能、单位、点动、模式差异见第17章「机器人总览」。

18.1 机构与坐标约定

在「Delta机器人常规」页的「机构参数 (mm)」里填四个尺寸：

项	含义
基座半径	上平台（固定盘）三个肩关节所在圆的半径
动平台半径	下平台（动盘）三个腕关节所在圆的半径
主动臂长	肩关节到肘关节的长度，电机直驱的那一段
从动臂长	肘关节到动平台的平行四边形连杆长度

- 三条臂固定在 0°、120°、240° 三个方位上。
- 臂角 θ 从水平面起算，向下为正；三根臂的关节轴用户单位必须是度。
- 工作区在基座下方，所以 TCP 的 Z 坐标是负值。

> 说明 **提示。**四个参数任一为 0 或负数，逆解一律拒绝。改完机构参数先到「Delta机器人参数」页点「正逆解校验」，回代误差在 1e-9 以下再运动。

18.2 工作区有多大

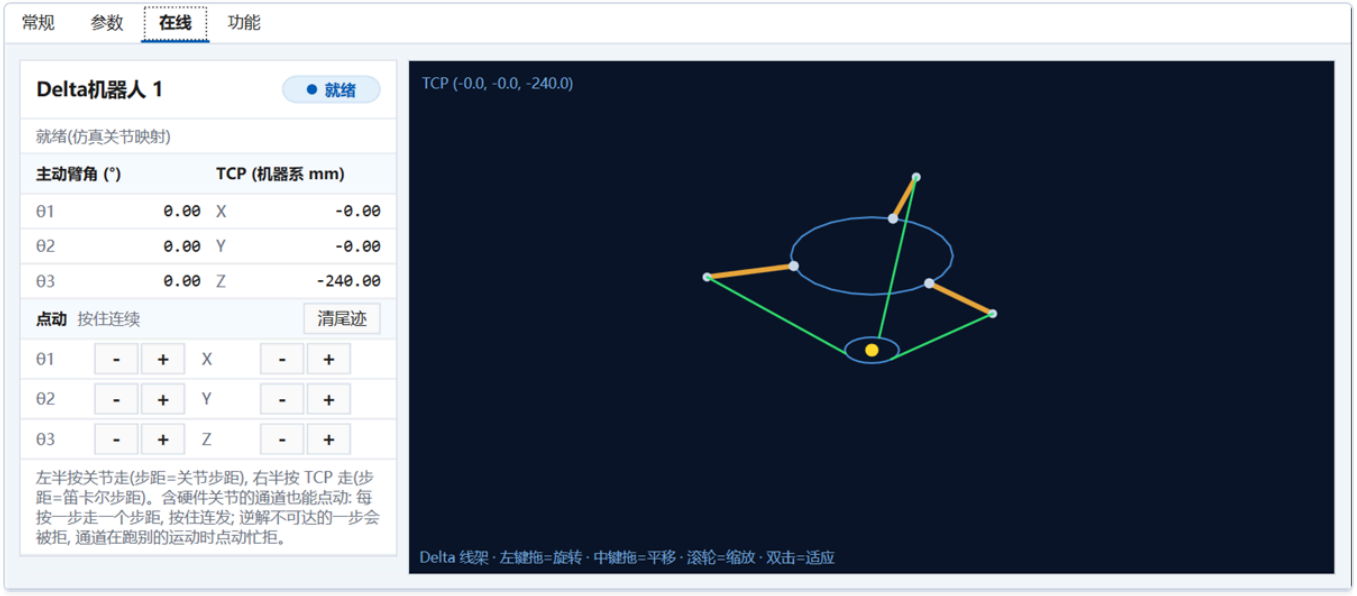
工作区由四个机构参数唯一决定。以出厂默认值（基座半径 180、动平台半径 60、主动臂 200、从动臂 400，单位 mm）为例：

项	数值
三臂水平时的 TCP（待机位姿 θ=0）	(0, 0, −240)
中心轴线上可达的 Z 范围	−587 mm ~ −160 mm
最大水平半径	约 411 mm，出现在 Z ≈ −200 处
越往下工作面越小	Z=−300 时约 377 mm，Z=−450 时约 278 mm，Z=−540 时约 158 mm

换成自己的机构参数后这些数整体变化，用「正逆解校验」逐点试探就能摸出边界。目标超出工作区时，软件在受理时就拒绝并提示「目标超出工作空间」，不会带着错误目标起步。

18.3 四个页面怎么用

- 「Delta机器人常规」：设通道名；把 θ1 / θ2 / θ3 三根主动臂分别映射到三根轴（没有 J4~J6）；填四个机构参数。「一键创建仿真映射」建 3 根仿真轴并映射。
- 「Delta机器人参数」：「主动臂角软限位 (°)」、「速度 / 加速度上限」（笛卡尔与臂角各一组）、「TCP 工具偏移 / 工件坐标偏移 (mm)」、「点动步距」（臂角步距 / 笛卡尔步距）、「正逆解校验器」。
- 「Delta机器人在线」：左边是「主动臂角 (°)」和「TCP (机器系 mm)」的实时读数，下面是点动按钮（左半按臂角，右半按 X/Y/Z）；右边是线架视图，实时跟随并画 TCP 尾迹，「清尾迹」清掉。
- 「Delta机器人功能」：「启动 PTP」（给三个臂角目标，「取当前」带入现值）、「启动 LIN」（给工件系 X/Y/Z，走笛卡尔直线）、「矩形演示」、「停止」；示教点的「记录当前」「GoTo」「回放」「删除」。



Delta 机器人「在线」页：主动臂角与 TCP 坐标、点动按钮，右侧是线架三维视图

18.4 软限位怎么设

- 三条臂在机械上完全相同，软限位也取同一个范围。出厂默认 −30° ~ +110°，按实际机型的臂角行程再收紧。
- 软限位是越界的唯一护栏：逆解结果超出限位时该动作被拒绝，不执行。范围别开得比机械行程宽，Delta 是闭链机构，三个臂角不互相独立。
- 老工程的软限位不会被覆盖，打开后确认三臂一致。

> 注意 **危险。**硬件通道第一次调试时，先在轴参数页确认三根臂轴的软限位已启用，手放在急停上。

18.5 Delta 上没有的东西

这些是 SCARA 与 6 轴串联才有的，Delta 页面上不出现，指令里传了非零值也会被拒：

- 腕轴 J4 / J5 / J6：Delta 只有三根主动臂。
- 姿态角 C / B / A：动平台姿态恒定，LIN 目标只有 X/Y/Z 三个分量。
- C 方向点动：只有 X / Y / Z 三个笛卡尔方向。

- TCP 的 C 分量：Delta 的 TCP 偏移是世界系里的纯平移，不随姿态旋转。

第19章 SCARA 机器人

SCARA 有四个关节：J1、J2 两段水平臂决定 TCP 落在平面上的位置，J3 负责竖直升降，J4 负责末端朝向。升降与朝向都不改变平面位置，所以它比串联六轴直观。这章只写 SCARA 特有的关节、机构参数、工作圆环和页面读法；建通道、使能、单位、点动、模式差异见第17章「机器人总览」。

19.1 四个关节各管什么

关节	作用	用户单位
J1（大臂回转）	大臂绕基座转，长度 L1	度
J2（小臂回转）	小臂绕大臂末端转，长度 L2	度
J3（Z 升降）	整条臂上下走，直动关节	毫米
J4（C 腕回转）	末端法兰自转，只改朝向不改位置	度

末端朝向 C = J1 + J2 + J4，是三个角之和。转 J1 或 J2 时，即使 J4 不动，末端朝向也跟着变。

> **说明** **J3 是全表唯一按毫米的关节。**软限位、点动步距、关节速度上限这三处的输入框，对 J1/J2/J4 是度、对 J3 是毫米。改值时注意 J3 与其余三个关节单位不同，不要照抄同一个数。

19.2 第 1 步：填机构参数，看清能到哪儿

在「SCARA机器人常规」页填两个尺寸：

参数	含义	默认
大臂 L1	J1 回转中心 → J2 回转中心	200 mm
小臂 L2	J2 回转中心 → 腕（C 轴）中心	150 mm

这两个数直接决定可达范围：TCP 到基座的平面距离 R 必须落在一个圆环里，$|L1 - L2| \leqslant R \leqslant L1 + L2$。取默认值 200 / 150 时可达圆环是 $50 \leqslant R \leqslant 350$ 毫米。圆环外的目标一律被逆解拒绝，提示「目标超出工作空间」。LIN 走不动、示教点「GoTo」报错，先回来量一下这个半径。

L1、L2 必须大于 0。填 0 或负数时正解不成立，在线页的位置回显退回原点。看到 TCP 变成 (0, 0) 先检查这两个框。改完机构参数，先到「SCARA机器人参数」页点一次「正逆解校验」再运动。

19.3 第 2 步：映射四根关节轴

在「SCARA机器人常规」页映射表右上角点「一键创建仿真映射」，软件建好 4 根仿真轴并映射；再点「一键复位映射轴」，然后「一键使能映射轴」。

做完应该看到：通道状态变成「就绪」，三维视图里的 SCARA 按当前关节值站位（默认待机姿态是轻度折肘 + 升降居中）。

J1/J2/J4 要映射到回转轴（用户单位 = 度），J3 要映射到直动轴（用户单位 = 毫米）。四根全部映射后通道才就绪。

19.4 第 3 步：让它动起来

方式	在哪 / 怎么做
关节点动	「SCARA机器人在线」页左半：J1 / J2 / Z / C 四行，点「−」「＋」。每一步的量 = 参数页的「关节步距」。
笛卡尔点动	在线页右半：X / Y / Z / C 四向。每一步的量 = 「笛卡尔步距」（X/Y/Z 按毫米，C 按度）。
PTP 关节运动	「SCARA机器人功能」页填四个关节目标（「取当前」带入现值）→「启动 PTP」。
LIN 直线运动	功能页填 LIN 目标（工件系 X / Y / Z / C）→「启动 LIN」，TCP 走笛卡尔直线。「矩形演示」在工作圆环中径处走一圈方形，用来看直线尾迹；「停止」受控停。

PTP 按四条关节里行程最长的那条统一规划，其余关节等比例跟随，四条关节同起同停。升降与回转共用同一组关节速度 / 加速度上限，设值时按较严的那一侧取。

19.5 贴近工作圆环边界时会发生什么

当 TCP 沿半径方向往圆环外边界（R → L1+L2）靠近时，两段臂逐渐伸直（J2 → 0）。这时：

- 同样的直线速度，要求的关节速度越来越快，LIN 可能在中途因关节跟不上而不平顺。
- 再往外一点就超出圆环，逆解失败、运动中途停机报错。

内边界（R → |L1−L2|，两臂完全折叠）是同一回事。把工位安排在圆环中段，别贴着两个边界规划路径。矩形演示自动挪到中径，就是这个原因。

19.6 肘分支

同一个笛卡尔目标，小臂向哪一侧弯折有两支解。「SCARA机器人参数」页的「机构参数 · 肘分支」选取哪一支：「肘正 (J2≥0)」或「肘负 (J2≤0)」。LIN、笛卡尔点动、校验器的逆解都按这里取，改后先「正逆解校验」再运动。当前臂形与选的那支不一致时，先 PTP 建臂形再 LIN（见第17章「机器人总览」）。

19.7 TCP 工具偏移与工件坐标

参数	SCARA 下的含义
TCP X / Y	工具相对法兰的平面偏移，随末端朝向 C 一起转（工具装在腕上，腕转工具跟着转）。
TCP Z	工具长度，直接叠加在高度上。
TCP C	工具本身相对腕的安装角。SCARA 专属，Delta 与 6 轴没有。
工件系 X / Y / Z / C	工件坐标系相对机器系的偏移。LIN 目标按工件系输入，在线页的 TCP 回显是机器系。

19.8 在线页两个 Z、示教表两个 Z 和两个 C

这是 SCARA 页最容易读错的地方：

位置	是什么
在线页左列 `Z mm`	升降关节 J3 的位置
在线页右列 `Z`	TCP 的高度。配了 TCP Z 偏移时，两者相差一个固定量。
示教表 `Z` / `C`	关节快照：J3 与 J4
示教表 `Z(TCP)` / `C(TCP)`	同一时刻的 TCP 高度与末端朝向

19.9 示教点：记录与回放

在「SCARA机器人功能」页的示教点面板，「记录当前」同时记下四个关节与当时的 TCP 位姿。「GoTo」与「回放」走关节 PTP 复现，不重新做逆解，所以回到的位形与记录时严格一致，不会出现"位置对了但手臂折向反了"。

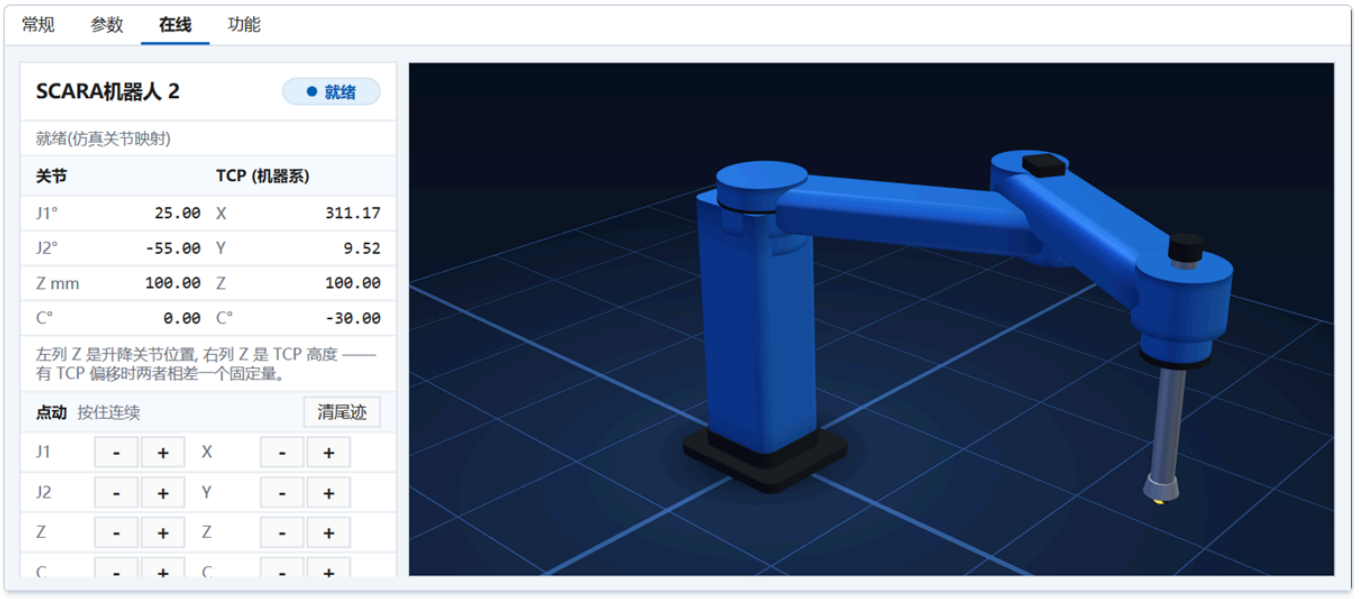
19.10 正逆解校验器怎么读

在「SCARA机器人参数」页填 J1(°) / J2(°) / Z(mm) / C(°) 四个值，点「正逆解校验」：软件先正解算出 TCP，再逆解算回关节，最后二次正解回代比对。结果里的回代误差小于 1e-9 时显示 √。

> **说明** **逆解算回的关节角可能与输入值不同，属于正常现象。**逆解按「肘分支」选定的一支计算，校验比较的是位置误差，而不是四个关节角是否逐个相等。

19.11 三维视图

- 左键拖 = 环绕旋转，滚轮 = 缩放，中键拖 = 平移，双击 = 视图适应。
- SCARA 显示为蓝色实体机器人，实时跟随点动 / PTP / LIN / 回放。
- 看不出逆解方向对不对时，先用它确认臂形与升降高度是否符合预期，再发运动。



SCARA 机器人「在线」页：关节读数与 TCP 坐标、点动按钮，右侧是三维视图

第20章 6 轴串联机器人

6R 是六个回转关节串联的球腕机器人：J1 绕基座竖轴回转，J2/J3 在竖直平面内摆大臂与小臂，J4/J5/J6 三条腕轴交于一点（腕心）。它是三种构型里唯一能同时控制位置与完整空间姿态的：位置 X/Y/Z 加姿态 C/B/A 共六个分量。这章只写 6R 特有的关节、姿态角、机构参数和臂形；建通道、使能、单位、点动、模式差异见第17章「机器人总览」。

20.1 六个关节各管什么

关节	作用	用户单位
J1（基座回转）	整臂绕基座竖轴转	度
J2（肩摆）	大臂在竖直平面内抬落	度
J3（肘摆）	小臂相对大臂抬落	度
J4（腕回转）	腕绕小臂轴自转	度
J5（腕摆）	腕绕横轴俯仰	度
J6（法兰回转）	法兰绕工具轴自转	度

六个关节全部是回转轴、全部按度，没有 SCARA 那根按毫米的升降关节。零位 = 手臂水平伸直指向 +X。位置与姿态六个分量都由六关节联合决定：转任何一个关节，TCP 的位置和姿态一般都会变，不像 SCARA 那样高度与朝向互不影响平面位置。

20.2 姿态角 C / B / A 是什么

6R 的末端姿态用 ZYX 欧拉角表示，旋转矩阵 `R = Rz(C) · Ry(B) · Rx(A)`：

分量	含义	范围
C（偏航）	绕机器系 Z 轴	(−180°, 180°]
B（俯仰）	绕 Y 轴	[−90°, 90°]
A（横滚）	绕 X 轴	(−180°, 180°]

> **说明** **字母别对号入座。**有的品牌把 A 定义为绕 Z、C 定义为绕 X，与这里正好相反。跨品牌抄姿态数值前先对一遍定义。

> **说明** **万向锁。**俯仰 |B| 接近 90°（工具竖直朝上或朝下）时，C 与 A 的读数可能一起突变。那是欧拉表示法的固有歧义，位姿本身连续，不是机械跳动。软件在这个姿态下约定 A=0。规划路径时别让工具长时间停在竖直姿态附近。

20.3 第 1 步：填机构参数

在「6轴机器人常规」页填五个尺寸：

参数	含义	默认
基座高 D1	肩关节（J2 轴线）离地高度	400 mm
肩偏置 A1	J1 回转轴线到肩关节的水平前伸	25 mm
大臂 A2	肩关节 → 肘关节	315 mm
小臂 D4	肘关节 → 腕心	365 mm
法兰 D6	腕心 → 法兰面（TCP 基准点）	80 mm

两个有用的已知值：腕心最大可达半径 = A1 + A2 + D4；零位 TCP 在 `(A1+A2+D4+D6, 0, D1)`，默认参数下就是 `(785, 0, 400)`，姿态 C=B=A=0。

A2、D4 必须大于 0，D1/A1/D6 不小于 0。非法值时正解不成立，在线页的位置回显退回原点。目标超出手臂能到的范围时逆解直接拒绝，提示「目标超出工作空间」。改完机构参数，先到「6轴机器人参数」页点一次「正逆解校验」再运动。

20.4 第 2 步：映射六根关节轴

在「6轴机器人常规」页映射表右上角点「一键创建仿真映射」，软件建好 6 根仿真轴并映射；再点「一键复位映射轴」，然后「一键使能映射轴」。

做完应该看到：通道状态变成「就绪」，三维视图里的机器人立着待机：大臂立起、小臂前伸、腕下俯（关节值 [0, 45, −30, 0, 45, 0]）。

六根全部是回转轴（用户单位 = 度），全部映射后通道才就绪。可以混搭硬件伺服轴与仿真轴：虚关节参与运动学、不驱动硬件。

20.5 第 3 步：让它动起来

方式	在哪 / 怎么做
关节点动	「6轴机器人在线」页左半：J1 ~ J6 六行，点「−」「＋」。每一步的量 = 参数页的「关节步距」。
笛卡尔点动	在线页右半：X / Y / Z / C / B / A 六向。X/Y/Z 平移（毫米），C/B/A 是绕 TCP 定点的纯姿态旋转（度），转姿态时 TCP 位置不动。
PTP 关节运动	「6轴机器人功能」页填六个关节目标（「取当前」带入现值）→「启动 PTP」。
LIN 直线运动	功能页填 LIN 目标：位置 X/Y/Z 按工件系，姿态 C/B/A 全六分量 →「启动 LIN」，TCP 走笛卡尔直线、姿态沿途同步过渡。「矩形演示」以当前 TCP 为中心自动选一处可达矩形走一圈，当前位姿放不下时退到肩高平面沿方位角逐档试探；「停止」受控停。

PTP 按六条关节里行程最长的那条统一规划，其余关节等比例跟随，六条同起同停。

20.6 先 PTP 建臂形，再 LIN 走直线

6R 的同一个 TCP 位姿可以由不止一种臂形到达（肘上/肘下、腕正/腕反）。直线运动的逆解固定取「肘正 + 腕 J5 非负」那一支。当前臂形与这一支不一致时（比如刚上电的待机姿是肘负），直接点「启动 LIN」会发生什么：

- 硬件通道和全仿真通道都会直接拒绝启动，手臂不动，状态里写明当前臂形与逆解选用的那一支不一致、需先用 PTP 建立臂形。这是保护动作，并非设备故障，先发一次 PTP 把手臂带到这一支的臂形，再启动 LIN 即可。
- 笛卡尔点动从静止起步时也做同样的检查，臂形不一致同样会被拒绝。

> **说明** **操作要点。**首次运动先用 PTP（或示教「GoTo」）把手臂带到目标附近的臂形，再用 LIN 走直线。功能页的 LIN 面板上有同样的提示。

20.7 TCP 工具偏移与工件坐标

参数	6R 下的含义
TCP X / Y / Z	工具尖端相对法兰面的偏移，在法兰系内、随姿态 C/B/A 一起旋转。
工件系 X / Y / Z / C	工件坐标系相对机器系的偏移，按分量直加（C 加在偏航上）。LIN 的位置目标按工件系输入，在线页的 TCP 回显是机器系。

6R 不支持工具安装角：姿态就是法兰姿态，TCP 偏移是法兰系内的纯平移，参数页该位置显示「—」。工件偏移也没有 B/A 分量。

20.8 示教点：记录与回放

在「6轴机器人功能」页的示教点面板，「记录当前」同时记下六个关节与当时的 TCP 位姿（X/Y/Z + C/B/A），表共 13 列：左六列是关节快照，右六列是同一时刻的位姿。「GoTo」与「回放」走关节 PTP 复现，不重新做逆解，回到的臂形与记录时严格一致，不会换肘、不会换腕分支。

20.9 正逆解校验器怎么读

在「6轴机器人参数」页填 J1 ~ J6 六个角，点「正逆解校验」：软件先正解算出 TCP 位姿（含 C/B/A），再逆解算回关节，最后二次正解回代比对。结果里的回代误差小于 1e-9 时显示 √。

逆解算回来的关节角可能与你填的不一样，这属于正常现象：同一位姿有多种臂形，软件固定取「肘正 + 腕 J5 非负」一支，校验比较的是位姿误差，而不是六个角是否逐个相等。

腕奇异（J5 接近 0°，或接近 ±180°，两种情况下 J4 与 J6 都共线）附近，J4/J6 的分配有歧义，软件约定 J4=0、合角给 J6。校验器在这类姿态下回代仍应通过：角不同、位姿相同。按默认软限位 J5 只到 ±175°，实际会遇到的是 0° 那一端。

20.10 三维视图

- 6R 页同时给两个视图：上方实体三维（左键拖 = 环绕旋转，滚轮 = 缩放，中键拖 = 平移，双击 = 视图适应），下方线架简图（肩/肘/腕心/TCP 四点连线）。
- 发运动前先用它确认臂形是否符合预期，尤其是 LIN 之前看一眼肘的方向。



6 轴机器人「在线」页：六个关节角与 TCP 位姿、点动按钮，右侧是三维视图

PART 04 · 第四篇

软卡 SDK

让自己的 C / C++ / C# 程序通过软卡 SDK 驱动真机。完整指令说明见《软卡 SDK 编程手册》。

软卡 SDK 上手 · 软卡设置

第21章 软卡 SDK 上手

给第一次用软卡 SDK 写客户程序的人：拿到包先看什么、怎么连上主软件、怎么把第一根轴安全地跑起来。

21.1 谁用它、前提是什么

软卡 SDK 给客户自己的程序用：.NET 或 C/C++ 程序通过它连主软件，读状态、拿控制权、使能轴、发运动、读写 I/O。工程、总线、从站、轴参数都在主软件里配，客户程序不碰 EtherCAT 过程数据。

- 真机运行时主软件必须常驻。总线、分布时钟和进入 OP 都由主软件负责，客户程序连上来时总线在跑。
- 主软件处于「实时运行」并「激活配置参数」进入 OP，客户程序才能拿控制权。
- 离线开发切到「仿真」：切进去就有完整的离线运行时，客户程序照常连，代码和真机一字不差。组插补、CNC、齿轮/凸轮、机器人在仿真里都能跑。
- 「配置调试」下 SDK 的写命令（拿控制权、轴命令、写输出）会被拒，读状态和诊断不受限。

21.2 SDK 包里有什么

SDK 包由主软件「软卡设置」页的「生成SDK包」生成，见第22章「软卡设置」。包里只有写程序必须的东西：

目录或文件	用途
`sdk\dotnet\SCmotionSoftCard.Core.dll`	.NET 库。C# 工程只引这一个。
`sdk\dotnet\interp_core.dll`	插补规划器。只有用到插补时才需要，放到 exe 同目录。
`sdk\c\include\scmotion_card.h`	C/C++ 头文件。
`sdk\c\win-x64\SCmotionSoftCard.dll` 与 `SCmotionSoftCard.lib`	C 动态库（x64）和导入库。DLL 随客户程序一起部署。
`docs\软卡SDK指令说明.html`	指令说明，从生成包的那份主软件帮助里抽出，用浏览器打开。内容与「软卡 SDK 指令说明」一章相同；两者不一致时以主软件里的帮助为准。
`README.txt`	引哪个库、怎么连、调用顺序。

包里没有示例工程，本章的最小程序就是示例。

21.3 连接配置文件

客户程序不手填轴数、I/O 点数、周期和共享内存名，这些都在连接配置文件里：

- 路径固定 `C:\ProgramData\SoftSC\softcard.runtime.json`。
- 主软件「激活配置参数」进入 OP 或切到「仿真」时自动写出，内容永远是当前工程。
- 客户程序用 `SCM_OpenProjectConfig(路径, clientId, 句柄)` 读它连接。不要复制旧文件，不要手工编辑。
- 文件不存在，多半是主软件还没进 OP，不是路径写错。

- `clientId` 是你程序自己的身份号，多个程序同时连要各不相同。连接配置文件里也有一个 clientId 字段，那是主软件自己的，不要照抄。

21.4 最短路径

`SCM_OpenProjectConfig` → `SCM_RequestOwner` → `SCM_AxisPower(轴, 1)` → 轮询 `SCM_AxisGetStatus` 的 `powered` → `SCM_AxisMoveRel` → 轮询到位 → `SCM_AxisPower(轴, 0)` → `SCM_ReleaseOwner` → `SCM_Close`。

- `SCM_RequestOwner` 只带句柄一个参数。拿到控制权就自动武装，可以直接发轴命令，不需要再调 `SCM_EtherCATStart`。
- `SCM_AxisPower` 是异步受理：返回 0 只表示收下了。完成看 `powered`；等待超时不要短于 15 秒。轴没使能就发 `SCM_AxisMove*` 会被直接拒绝。
- `SCM_ReleaseOwner` 归还控制权并解除武装，但不停正在跑的运动。要连运动一起收掉，先 `SCM_AxisStop`，或用 `SCM_EtherCATStop`。
- 轴号从 0 开始，第一根轴是 0。
- 所有指令返回 `int` 错误码，0 是 `SCM_OK`。正式程序每一步都要判返回码。

21.5 最小程序

.NET 与 C 两份做的是同一件事：连接、拿控制权、使能 0 号轴、走 10 个用户单位、下电、归还、关闭。距离和速度只是示例，现场按机械比例和限位改。

> **注意** **危险。**第一次跑真实伺服，速度和距离用保守值，急停在手边。先确认整条链路正常，再加大行程。

.NET（引 `SCmotionSoftCard.Core.dll`）：

```
using System;
using System.Text;
using System.Threading;
using SCmotion.SoftCard.Sdk;
using static SCmotion.SoftCard.Sdk.ScMotionApi;

int rc = SCM_OpenProjectConfig(@"C:\ProgramData\SoftSC\softcard.runtime.json", 300, out var card);
if (rc != SCM_OK)
{
    var why = new StringBuilder(1024);
    SCM_GetOpenErrorText(why, 1024);              // 逐项列出双方的轴数、点数、周期
    Console.WriteLine($"连接失败 {rc}: {why}");
    return;
}

try
{
    rc = SCM_RequestOwner(card);                  // 只带句柄；成功即自动武装
    if (rc != SCM_OK) { Console.WriteLine($"拿不到控制权 {rc}"); return; }
```

```
    SCM_AxisPower(card, 0, 1);                    // 异步受理，返回 0 只表示收下了
    if (!Wait(() => { SCM_AxisGetStatus(card, 0, out var st); return st.powered && !st.faulted; }, 15000))
    { Console.WriteLine("使能超时"); return; }

    SCM_AxisMoveRel(card, 0, 10.0, 5.0, 0, 0, 0); // 走 10 个用户单位，5 单位/s；后三个 0 = 用轴页默认
    Thread.Sleep(50);
    if (!Wait(() => { SCM_AxisReadStatus(card, 0, out var rs); return rs.valid && !rs.discreteMotion; }, 30000))
    { Console.WriteLine("到位超时"); SCM_AxisStop(card, 0); return; }

    SCM_AxisGetStatus(card, 0, out var done);
    Console.WriteLine($"到位 actualPosition={done.actualPosition}");

    SCM_AxisPower(card, 0, 0);                    // 下电同样是异步受理
    Wait(() => { SCM_AxisGetStatus(card, 0, out var st); return !st.powered; }, 15000);
}
finally
{
    SCM_ReleaseOwner(card);                       // 归还控制权，不停正在跑的运动
    SCM_Close(card);
}

static bool Wait(Func<bool> done, int timeoutMs)
{
    long t0 = Environment.TickCount64;
    while (!done())
    {
        if (Environment.TickCount64 - t0 > timeoutMs) return false;
        Thread.Sleep(50);
    }
    return true;
}
```

C（引 `scmotion_card.h`，链接 `SCmotionSoftCard.lib`）：

```
#include <stdio.h>
#include <windows.h>
#include "scmotion_card.h"

/* 轮询 powered 到指定值，超时返回 0 */
static int wait_powered(SCM_CARD_HANDLE card, int axis, int want, int timeoutMs)
{
    SCM_AXIS_STATUS st;
    DWORD t0 = GetTickCount();
    for (;;) {
        SCM_AxisGetStatus(card, axis, &st);
        if (want ? (st.powered && !st.faulted) : !st.powered) return 1;
        if (GetTickCount() - t0 > (DWORD)timeoutMs) return 0;
        Sleep(50);
    }
}
```

```
/* 轮询定位运动结束（discreteMotion 落下），超时返回 0 */
static int wait_in_position(SCM_CARD_HANDLE card, int axis, int timeoutMs)
{
    SCM_AXIS_PLC_STATUS rs;
    DWORD t0 = GetTickCount();
    Sleep(50);
    for (;;) {
        SCM_AxisReadStatus(card, axis, &rs);
        if (rs.valid && !rs.discreteMotion) return 1;
        if (GetTickCount() - t0 > (DWORD)timeoutMs) return 0;
        Sleep(50);
    }
}

int main(void)
{
    SCM_CARD_HANDLE card = 0;
    SCM_AXIS_STATUS st;
    char why[1024];
    int rc;

    rc = SCM_OpenProjectConfig("C:\\ProgramData\\SoftSC\\softcard.runtime.json", 300, &card);
    if (rc != SCM_OK) {
        SCM_GetOpenErrorText(why, sizeof why);   /* 逐项列出双方的轴数、点数、周期 */
        printf("open failed %d: %s\n", rc, why);
        return rc;
    }

    rc = SCM_RequestOwner(card);                 /* 只带句柄；成功即自动武装 */
    if (rc != SCM_OK) {
        printf("request owner failed %d\n", rc);
        SCM_Close(card);
        return rc;
    }

    SCM_AxisPower(card, 0, 1);                   /* 异步受理，返回 0 只表示收下了 */
    if (!wait_powered(card, 0, 1, 15000)) { printf("power on timeout\n"); goto done; }

    SCM_AxisMoveRel(card, 0, 10.0, 5.0, 0, 0, 0); /* 走 10 个用户单位，5 单位/s；后三个 0 = 用轴页默认 */
    if (!wait_in_position(card, 0, 30000)) { printf("move timeout\n"); SCM_AxisStop(card, 0); goto done; }

    SCM_AxisGetStatus(card, 0, &st);
    printf("in position, actualPosition=%.3f\n", st.actualPosition);

    SCM_AxisPower(card, 0, 0);                   /* 下电同样是异步受理 */
    wait_powered(card, 0, 0, 15000);

done:
    SCM_ReleaseOwner(card);                      /* 归还控制权，不停正在跑的运动 */
    SCM_Close(card);
```

```
    return 0;
}
```

说明 **提示。**连接失败时只看错误码不够。`SCM_GetOpenErrorText` 会逐项列出双方的轴数、I/O 点数、周期，照着对就知道哪里不一致。它是全表唯一不带句柄的指令，因为连接失败时还没有句柄。

21.6 第一根轴现场步骤

按下面的顺序走，每一步看到预期结果再进下一步。

1. 运动前检查：急停回路有效；正负限位、软限位和运动方向在主软件轴页低速点动确认过；驱动器无故障。
2. 只读诊断：连上后先 `SCM_GetDiagnostics`，不拿控制权。`cycleCounter` 要持续增长，`workCounter` 稳定，`missedCycleCount` 不持续增长。
3. 读轴状态：`SCM_AxisGetStatus(句柄, 0, 状态)`。`faulted` 为真先记下 `statusWord` 和 `errorCode`，去主软件排故障；`actualPosition` 一直是 0 或不变化，先查反馈 PDO 映射。
4. 拿控制权：`SCM_RequestOwner`。返回 `SCM_REJECTED` 说明控制权在别的程序手上；返回 `SCM_INVALID_STATE` 说明主软件没进 OP 或当前模式不允许下发。失败就停在这里，不要往下发命令。
5. 使能：`SCM_AxisPower(句柄, 0, 1)`，轮询 `powered` 为真且 `faulted` 为假。
6. 小距离相对运动：`SCM_AxisMoveRel`，距离和速度取保守值，后三个参数传 0 用轴页「动态」默认值。看 `actualPosition` 按预期方向变化；方向反了，停下来回主软件轴页和驱动器参数查方向，不要在程序里用取反来弥补。
7. 收尾：还在动就 `SCM_AxisStop` → `SCM_AxisPower(句柄, 0, 0)` → `SCM_ReleaseOwner` → `SCM_Close`。程序异常退出而未归还控制权时，主软件约 5 秒后判定其失联，下一次 `SCM_RequestOwner` 自动收回。

跑不动、连不上、被拒，按《软卡 SDK 编程手册》「软卡排障与版本匹配」的顺序查。

21.7 C/C++ 工程设置

Visual Studio 工程按下表配。SDK 包里没有 `sdk\c\win-x64\SCmotionSoftCard.dll` 时，让生成包的人补齐后重新生成一份，不要在工程里硬配不存在的库。

设置项	值
平台	`x64`。不要用 Win32。
C/C++ → 附加包含目录	SDK 包的 `sdk\c\include`
链接器 → 附加库目录	SDK 包的 `sdk\c\win-x64`
链接器 → 附加依赖项	`SCmotionSoftCard.lib`
运行目录	`SCmotionSoftCard.dll` 放 exe 同目录；用到插补再放 `interp_core.dll`。

运行前检查：

- 编译找不到头文件：包含目录没配。

- 链接找不到库：库目录没配，或包里没有 `.lib`。
- 启动报找不到 DLL：exe 同目录没有 `SCmotionSoftCard.dll`。
- 结构体字段读数离谱：头文件和 DLL 不是同一个包里的。升级 SDK 后清掉旧头文件和旧库，重新编译。
- 每次调用都判返回码，非 0 就把返回码、`SCM_GetLastReason` 的原因码和诊断值一起写进日志。

第22章 软卡设置

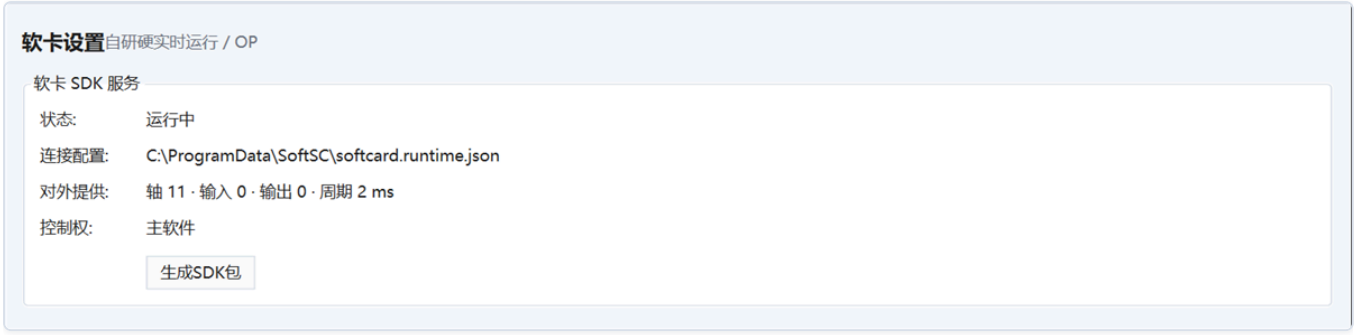
主软件里给软卡 SDK 用的一页：看软卡 SDK 服务的状态和控制权在谁手上、生成给客户的 SDK 包。SDK 指令本身见《软卡 SDK 编程手册》「软卡 SDK 指令说明」。

22.1 从哪里进

左侧工程树「SYSTEM - 配置」下点「软卡设置」。页面右上角显示当前模式。

22.2 页面上有什么

一个区块「软卡 SDK 服务」，四行状态，外加一个「生成SDK包」按钮。状态在打开这一页时每秒刷新。



软卡设置页：服务状态、连接配置、对外提供、控制权四行，加一个「生成SDK包」按钮

字段	看什么
状态	「运行中」表示客户程序现在就能连上来。「未运行」时后面括号里写了怎么让它跑起来：真机「激活配置参数」进入 OP，或者切到「仿真」模式。
连接配置	连接配置文件的完整路径（`C:\ProgramData\SoftSC\softcard.runtime.json`）。客户程序用 `SCM_OpenProjectConfig` 读这份文件连上来，不用手填共享内存名与轴数。
对外提供	客户程序连上来会看到的轴数、输入输出点数与周期。这些值以服务启动时的工程为准：之后改了工程（重扫拓扑、改周期、增删轴），要重新「激活配置参数」才会跟着变。
控制权	「主软件」表示没有客户程序持有控制权；客户程序拿走后显示是哪个客户端、以及它最近一次活动是多久以前。

22.3 软卡 SDK 服务是什么

它是主软件内部的一块服务：主软件进入 OP（或切到仿真模式）时自动启动，把当前工程的轴与 I/O 通过共享内存交给客户程序使用；退出运行时自动停止。客户程序不需要另外启动任何进程，也不需要导出任何配置文件。

离线开发（手边没有真机）就把主软件切到「仿真」模式，客户程序照样连得上、跑得动，只是不驱动真实伺服。

22.4 控制权

同一时刻只有一个程序能下发运动和写输出，这个程序就是属主。其余连接只能读。

- 客户程序用 `SCM_RequestOwner` 取得控制权，主软件这边的运动与输出下发自动让出；客户程序 `SCM_ReleaseOwner` 交还后恢复。
- 客户程序异常退出没有交还时，服务约 5 秒判定它已经不在了，下一次 `SCM_RequestOwner` 会自动收回。
- 客户程序判断控制权在不在自己手上，要拿自己的 clientId 和 `SCM_GetDiagnostics` 的 `ownerClientId` 比对。

22.5 生成 SDK 包

1. 点「生成SDK包」，选一个输出目录。
2. 目录下生成 `SCmotionSoftCardSdk_工程名_时间` 文件夹，消息栏报「软卡 SDK 包已生成: …」。
3. 有文件缺失时消息栏报一条黄色提示，按提示补齐后重新生成一次。

包里有什么见第21章「软卡 SDK 上手」。包里的指令说明是从生成它的那份主软件帮助里抽出来的，升级主软件后重新生成一份。

不需要另外给客户程序导出任何文件：工程在主软件里配完，「激活配置参数」进入 OP 时连接配置自动写到 `C:\ProgramData\SoftSC\softcard.runtime.json`。

22.6 常见问题

现象	原因	处理
状态显示「未运行」	真机没有「激活配置参数」进入 OP，或者没有切到「仿真」模式。	按括号里的提示做；客户程序这时连接会收到「连不上主软件」。
客户程序报连接参数对不上	服务提供的是激活时的工程配置，之后工程又做了修改。	重新「激活配置参数」，客户程序重新读连接配置文件。
客户程序写输出、发运动被拒	没有取得控制权，或者控制权在别的客户程序手上。	看「控制权」这一行是谁；客户程序侧用 `SCM_GetLastReason` 读原因码。
生成的包缺 DLL	生成包的这台机器上没有对应文件。	看消息栏的提示，补齐后重新生成。

PART 05 · 第五篇

排障与参考

故障排查、日志与术语。

故障排查 · 日志 · 术语

第23章 故障排查

现场出问题时按"现象、先看哪里、怎么处理"查：先确认模式，再按扫不到、进不了运行、掉线、使能失败、拒动五类排，最后是现象速查表和两张状态码表。

23.1 先确认当前模式

很多"不动""被拒"不是故障，是模式不对。看顶部三段模式条，或状态栏右侧的「数据源 · 模式 · 总线状态」；托盘图标也能看：紫=仿真、蓝=配置调试、绿=运行、红=停止。

模式	能做什么	做不了什么
「仿真」	不接硬件。SDK 客户程序可以直接连，组插补、CNC、齿轮 / 凸轮、机器人都能跑。	没有真实回零开关，I/O 只是内存里的位。
「配置调试」	扫描、改 PDO 分配和启动参数、CoE 在线、简单 I/O、诊断。SDK 的状态、诊断、过程映像读取不受限。	真实轴的点动、定位、回零；SDK 的建总线、轴命令、写输出按设计拒绝。
「实时运行」	真机 OP，真实轴的全部运动。	改 PDO 映射（要回「配置调试」，且过程数据停着）。

工程改过要重新点「激活配置参数」，否则总线和 SDK 用的还是上次激活那一刻的配置。「界面锁定」开着时写 I/O、改参数、动总线、发运动都被挡，先解锁。

23.2 扫不到网卡或从站

扫描入口是工程树里主站节点右键「扫描」。列不出网卡先看主站「适配器」页，扫不到从站先看链路。

- 网卡要是认证型号（Intel I210 / I211 / I350 / I217 / I218 / I219 系），并且在「网卡管理...」窗口里装好实时网卡驱动、选中为当前网卡。普通办公网口不行。
- 网卡被别的抓包或诊断程序占着，先关掉它们。
- 网线插在从站的 IN 口，从站已上电、链路灯亮。刚上电的从站等几秒再扫。
- 扫到的站数比实际少：从缺的第一站往前查网线和供电，中间断了后面全没有。
- 软件启动时停在网卡枚举，等它走完，不要再开第二个实例。

链路好不好看主站「在线」页：「帧/秒」接近周期对应的值，「丢帧」「Tx 错误」「Rx 错误」不涨，「WKC」等于期望值。

23.3 进不了实时运行

点「实时运行」或「激活配置参数」后没进 OP，先看消息栏里第一条红色消息，再对下面几条。

- 没扫描过：离线拓扑起不了真实过程数据。先扫描一次，让软件核对工程和真实从站。
- 日志说"启动已拒绝：N 台从站没有可用的说明文件……请先扫描一次"：这次是按工程直接连的，没读过硬件配置。扫描一次再激活。

- 日志说"N 台从站没有匹配到 ESI 文件，已按设备自身配置启动"：这只是警示，不会阻止启动。认得出是驱动器的会自动建轴、走分布时钟；认不出的只做 I/O 和状态查看。想按说明文件配置，就在「ESI 设备库」导入对应型号的文件后重扫。
- 有从站停在 SAFE-OP 或报 AL 状态码：按下面「AL 状态码」表处理。最常见的是 0x001D / 0x001E（主站配置和从站里残留的映射不符），重新扫描后再启动，还不行就给从站断电再扫。
- 日志说"驱动与软件版本不匹配"：主软件和实时网卡驱动不是同一个安装包装的。用完整安装包重装，让驱动随包更新，不要单独替换驱动文件。
- 日志说"授权尚未生效"：本机时钟比授权签发时间早。把时钟改对，回「授权管理」页点「刷新」。永久授权不看签发时间。
- 启动参数写入失败：看日志里是哪台站哪个对象被拒，去该站「启动参数」页核对。回零参数被驱动器拒收不影响进 OP，见第12章「回零」。

23.4 运行中掉线或退到 SAFE-OP

- 先看主站「在线」页：「帧/秒」掉到 0 是总线停了；「丢帧」在涨是链路或电脑负载问题；「WKC」小于期望值是有站没应答；「DC 偏差」跳大是同步不稳。
- 从站「在线」页看是哪一站先退的、AL 状态码是什么。看门狗超时（0x001B）多是网线、供电或电脑卡顿。
- 运行中读写 CoE 每次都有时间上限，不会阻塞周期，但它是非周期通道。要持续盯的值加进 PDO，用过程数据页或示波器看，别拿 CoE 轮询。
- 过程数据显示在跑、EtherCAT 却不在 OP：真实总线没建起来或已掉线。重查网卡、拓扑，以及导出日志里的「启动诊断全文」那一行。
- 「SYSTEM - 配置」页的「总线质量运动校验」开着时，丢帧率超过阈值会拒绝新的使能，运行中超过更高的阈值会停掉全部硬件轴，出厂值 3% 和 5%。被它拦下先修链路，不要把校验关掉了事。

23.5 使能失败

轴页「在线」页签点「使能」后，看「运行信息」一栏和「CiA402 状态」，按提示对表。状态名显示成「中文（英文原名）」，英文原名与驱动器手册里的写法一致。

「运行信息」提示 / 卡点	处置
故障复位进行中，请等复位完成后再使能	复位流程还在后台走，会和使能序列互相覆盖控制字。等几秒再点。
使能提示：驱动器的运行模式可能没配对	不拦使能，但这种状态下轴显示已使能却不跟位置指令。`6060` 不在 RxPDO 里的，在该站「启动参数」页加一行 `6060` =8 后重新「激活配置参数」；`6060` 在 RxPDO 里的，重新激活让启动映像把它置回 8。
CiA402 状态停在「初始化未完成（Not Ready To Switch On）」	驱动器自己没就绪。查驱动器面板报警：编码器、母线电压、安全端子没接通是常见原因。

仍为故障状态 603F=……	先点「复位」。常见驱动器的故障码软件会翻成中文并标出能否在线清除，翻不出的查驱动器手册；不可在线清除的（如编码器断线）先处理硬件，再给驱动器断电重启。
卡在「准备接通（Ready To Switch On）」或「已接通（Switched On）」超时	控制字 `6040` 没生效。确认 `6040` 在 RxPDO 映射里、过程数据在跑（WKC 正常）、驱动器的使能 / 安全端子接通。
丢帧超限拒绝使能	「总线质量运动校验」拦的。见上一节，先修链路。
使能成功，电机不出力也不报警	转矩被钳为 0。查 `6072`（最大转矩）、`60E0` / `60E1`（转矩限制）是不是 0；有的驱动器把 `6072` 映进了 RxPDO 而输出映像默认 0。软件的使能流程会初始化这些对象，自定义流程要自己写（3000 = 300%）。
使能成功，规划位置在变、实际位置不变	从站没运行在分布时钟同步下。伺服必须走分布时钟（`1C32:01` = 2）；读回不是 DC 时软件只在日志里对该站发出告警，不阻止使能。去该站「分布式时钟」页和「同步模式」设置核对，重新「激活配置参数」，再看日志里那条告警还在不在。
SDK 客户程序发 `SCM_AxisMove*` 被拒	`SCM_AxisPower` 是异步受理，返回 0 只是收下了。轮询 `SCM_AxisGetStatus` 的 `powered` 为真后再发运动；轴没使能或使能还没完成时发运动会被直接拒绝。

注意 **危险。**使能后规划位置在走、实际位置不动的轴，先点「停止」再去使能，查清原因再试。不要反复点使能等它动，也不要把这台轴放进组里跑。

23.6 运动按钮灰或拒绝运动

- 真实轴的点动、寸动、定位、回零只在「实时运行」下可用。「配置调试」下这些按钮变灰是正常的保护措施。
- 「界面锁定」开着：解锁。它不随工程保存，重启软件就解除。
- 轴在故障（Fault）状态：先「复位」，再确认已使能。
- 刚进 OP 时规划位置和实际位置差很多：先点「复位」，它会把规划位置同步到实际位置，再运动。
- 轴被插补组、CNC、凸轮、齿轮、机器人通道或 SDK 客户程序占着：被组接管期间单轴指令会被拒，先脱开或释放属主。
- 点动速度填了 0，或目标超出软限位。软限位的生效条件见第11章「轴运动与安全边界」。
- 硬件插补起不来：日志会写拒绝原因。成员轴要先使能，硬件插补只在「实时运行」下可用。

23.7 现象速查

现象	最可能的原因与处置
使能成功、指令发了，电机不动也不报警	看轴页「在线」：规划位置在变、实际不变，是从站没在分布时钟下，看日志里有没有这一站的告警；规划和实际都在变、电机轴不转，是转矩被钳零。两条都在上面「使能失败」表里。
点动没反应	依次确认：轴已使能（「CiA402 状态」显示运行使能）、在「实时运行」、轴没被别的通道或 SDK 占用、点动速度不是 0、「界面锁定」没开。

伺服进不了 CSP（`6061` 一直不是 8）	① `6060` 没设成 8，见「使能失败」表；② 分布时钟没配上，或 SYNC0 / SYNC1 激活方式和驱动器要求不符，看该站「分布式时钟」页；③ 主站时钟还没锁定，等一分钟内收敛。插补周期 `60C2` 由软件按主站周期自动写进启动参数，不用手填。
重新进配置调试后启动失败 AL=0x001D / 0x001E	从站里残留着上次的映射，和这次的配置对不上。重新扫描后再启动；还不行就给从站断电复位再扫。
装了新软件，实时核起不来，日志说驱动与软件版本不匹配	驱动没跟着换。用完整安装包重装，不要手工拷驱动文件。
「授权尚未生效」	本机时钟慢于签发时间。改对时钟，「授权管理」页点「刷新」。
日志说"有 N 台从站认不出是驱动器还是 I/O 模块"	没有说明文件的设备按它自己存的出厂配置识别。判得出是驱动器的能建轴能动；判不出的这一台不会自动建轴、也不计入 I/O。把该型号的说明文件导入「ESI 设备库」后重扫；设备报的产品编号和文件对不上时，在别名文件里登记对应关系，见第7章「EtherCAT 与从站」。
「驱动器回原点」走的速度不是我填的	激活时回零参数（如 `6099:1`）被驱动器拒收，该轴按驱动器内部存的值走，日志会指出是哪台驱动器的哪个参数。有的驱动器对速度有下限。把该参数改到允许的范围内，下次回零时会自动写入；写不进去会拒绝回零并在消息栏说明原因。
扫描列不出网卡，或扫不到从站	见上面「扫不到网卡或从站」。
轴运动指令返回非 0	错误码对照见第11章「轴运动与安全边界」。

23.8 AL 状态码

从站拒绝状态切换时，从站「在线」页和日志会给出 AL 状态码。常见值：

AL 码	含义	处置
0x0011	无效状态切换请求	按 INIT → PRE-OP → SAFE-OP → OP 顺序推进；重新扫描让软件接管状态机。
0x0016	邮箱配置无效	从站上电没就绪就扫描容易触发，等就绪后重扫；个别从站要断电复位。
0x001A	同步错误	从站没跟上 SYNC 信号。查「分布式时钟」页的配置，周期要是驱动器支持的值，重新激活。
0x001B	SM 看门狗超时	从站在看门狗时间内没收到完整的周期输出。查网线、供电，电脑是不是被高负载卡住。
0x001D	输出配置无效（SM2 / RxPDO 尺寸或分配不符）	主站配置和从站里的映射对不上，多是残留状态。重新扫描后再启动；不行就断电复位从站；核对该站「过程数据」页和「启动参数」页。
0x001E	输入配置无效（SM3 / TxPDO 尺寸或分配不符）	同 0x001D。
0x001F	看门狗配置无效	重新扫描后再启动。

0x0024	输入映射无效	TxPDO 里映了从站不认的对象。去「过程数据」页核对，或「恢复默认」。
0x002D	未收到 SYNC 信号	分布时钟配置和驱动器要求不符（如要 SYNC0+SYNC1 却只开了 SYNC0）。核对「分布式时钟」页后重新激活。
0x0030	DC 同步配置无效	同上；确认周期是驱动器支持的值（常见 1 / 2 / 4 ms）。
0x0035 / 0x0036	DC 同步周期 / SYNC0 周期无效	周期值驱动器不接受，改成它支持的值。

23.9 CiA402 状态字（6041）

轴页「在线」的「CiA402 状态」就是按这张表翻译的，显示成「中文（英文原名）」；鼠标停在这一格上，还会列出状态字标志位的中文（如主回路已上电、远程控制、已到位）。自己看原始值时，用每一行最后一列的掩码取 6041 的低位，再和第一列对：初始化未完成、待机、故障反应中、故障这四种不看 bit5（快速停止位），用 0x004F；其余四种用 0x006F。

6041 & 掩码	状态	界面显示	说明	掩码
0x0000	Not Ready To Switch On	初始化未完成（Not Ready To Switch On）	驱动器内部初始化没完成，开不了使能序列。等待或查驱动器报警。	0x004F
0x0040	Switch On Disabled	待机（Switch On Disabled）	正常待机，可以开始使能。	0x004F
0x0021	Ready To Switch On	准备接通（Ready To Switch On）	使能第一段完成（`6040` =0x0006 生效）。	0x006F
0x0023	Switched On	已接通（Switched On）	使能第二段完成（`6040` =0x0007 生效）。	0x006F
0x0027	Operation Enabled	运行使能（Operation Enabled）	已使能，电机闭环锁轴，可以运动。	0x006F
0x0007	Quick Stop Active	快速停止中（Quick Stop Active）	快停进行中。	0x006F
0x000F	Fault Reaction Active	故障反应中（Fault Reaction Active）	正在按故障减速停机。	0x004F
0x0008	Fault	故障（Fault）	故障态，bit5 是 0 是 1 都算（如 `0x0208` 、 `0x0A38` ）。软件读 `603F` 并显示含义，也可看驱动器面板；轴页「复位」或 `SCM_AxisReset` 清除。	0x004F

其它值	—	未识别的状态	不是上面任何一种组合，按原始值逐位对照。bit3 为 1 的，先看驱动器面板或 `603F` 有没有报警，再点「复位」。	—

> **说明** **提示。**bit10 = 到位，bit12 的含义随运行模式变。位置模式下看软件的到位判定就够了，不用自己解析。

23.10 需要技术支持时

- 先复现一次，让消息栏里有最近一次失败的记录。
- 不要退出软件，在消息栏右键「导出」保存日志。正常退出会删掉本次的日志文件，所以要在退出前导出。导出文件里含启动过程的完整明细。
- 把导出的日志、工程文件（`.scproj`）和一句现象描述（什么模式、点了什么、期望什么、实际什么）一起发出。

第24章 日志

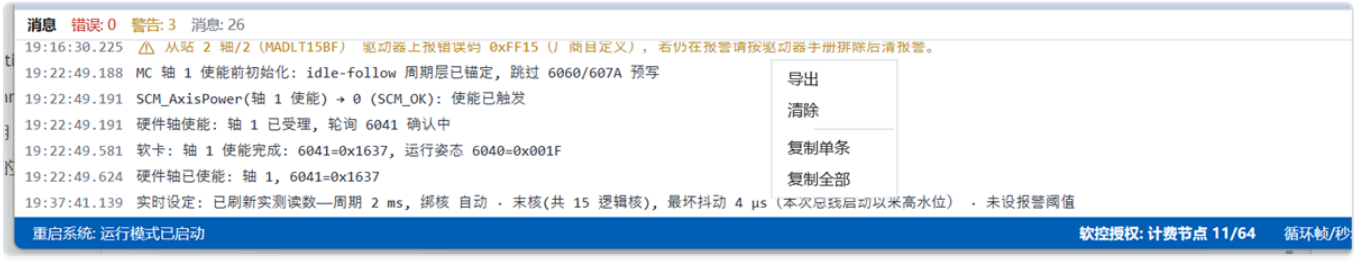
主窗口底部消息栏上看什么、怎么导出、为什么要在退出前导出，以及报障要带什么。

24.1 消息栏上看到的和日志里记的

主窗口底部的消息栏只显示简明的消息：红字是错误，黄字是警告，其余是普通消息。排障用的明细（对象号、寄存器值、逐站清单）不上消息栏，只记进日志，导出之后才看得到。

消息栏顶上一行是三个计数：「错误」「警告」「消息」，只数上过消息栏的那些，不数只记进日志的明细。同一句消息或警告 30 秒内重复出现只显示一次，等出现其他消息时补一行「相同消息已合并 N 次」（警告是「相同警告已合并 N 次」）；错误每次都显示。

消息栏只列最近 160 条，软件一共记着最近 2000 条（含消息栏上看不到的明细），更早的会被挤掉。出了问题尽快导出。



底部消息栏：顶上是错误、警告、消息三个计数，右键菜单有导出、清除、复制单条、复制全部

24.2 右键菜单

在消息栏里点右键。点在哪一行上，那一行就被选中。

菜单项	做什么
「导出」	弹「导出日志」保存框，默认文件名 `SoftSC_Log_日期_时间.log`，也可存成 .csv 或 .txt。内容三列：时间、级别（Info / Warn / Error）、消息，制表符分隔，含消息栏上看不到的明细。
「清除」	清掉消息栏、三个计数和软件记着的那 2000 条，磁盘上的日志文件不动。清除以后再导出，只剩清除之后的内容，所以只在准备复现问题之前使用；已经记下的故障记录，导出之前不要清除。
「复制单条」	把右键点中的那一行复制到剪贴板。
「复制全部」	把软件记着的全部复制到剪贴板，和导出的是同一份。剪贴板被输入法、剪贴板管理器或远程桌面占着时会弹框，可以改成导出到文件。

24.3 退出会删除日志，请在退出前导出

软件正常退出（托盘「退出并停止」）会删掉本次运行写在磁盘上的日志文件，不在电脑上保留。需要保留日志时，必须在退出前右键「导出」。

崩溃或断电走不到正常退出，那一次的日志会留在 `C:\ProgramData\SoftSC\SystemManager\logs\app-日期.log`，下次启动时清掉一天以前的。软件起不来又要查原因，就去这个目录拿。

> **说明** **提示。**关窗口只是进托盘，日志还在。真正的退出是托盘菜单「退出并停止」。

24.4 启动诊断全文

每次「激活配置参数」或进 OP 后，消息栏上只有一句汇总；导出的日志里多一行以「启动诊断全文:」开头的记录，是启动过程的完整明细：每站的状态切换、启动参数下发结果、分布时钟配置和时基读回。进不了 OP、使能后不跟随、某一站行为异常，先找这一行。

24.5 先看哪几行

- 先看第一条错误，和它前面的几条警告。后面的错误多是连锁。
- 启动问题：软件启动、工程打开、实时网卡驱动初始化、授权、网卡枚举。
- 扫描问题：网卡选择、拓扑扫描、从站数量、说明文件匹配结果、日志中列出的站。
- 进 OP 问题：启动参数下发、PRE-OP / SAFE-OP / OP 切换、AL 状态码、「启动诊断全文」。
- 软卡问题：软卡设置页的状态区，加上日志里以「软卡」开头的行。
- 轴问题：使能序列、`6041` 状态字、`603F` 错误码、`6061` 运行模式、规划位置和实际位置。
- 运动问题：回零、组 / 通道占用、写着拒绝原因的那一行。

24.6 常见消息

日志里的字样	意思	先查什么
实时驱动初始化失败	实时网卡驱动没装、网卡没绑定或被占用。	关掉其它占网卡的程序；主站「适配器」页「网卡管理...」重装并选中网卡。
驱动与软件版本不匹配	主软件和实时驱动不是同一个安装包装的。	用完整安装包重装。
过程数据已中断（0 帧/秒）	总线停了。	查网线和从站供电；主站「在线」页看当前状态和 WKC；恢复后停止并重新「激活配置参数」。
未找到 RxPDO 地址 / 未找到 TxPDO 地址	要用的对象没映射进过程数据。	该站「过程数据」页：输出对象在 0x1600 一族里，反馈对象（`6041`、`6064`、`606C`、`603F`）在 0x1A00 一族里。
OP中未读(原因)	运行中这一笔 CoE 没读，括号里是原因。	按原因处理；急着看的值回「配置调试」读。
OP中SDO读取失败(邮箱无应答/超时)	通道正常，但这台站没有在限定时间内应答。	查该站接线和供电。和上一条不是一回事。

启动已拒绝：N 台从站没有可用的说明文件……请先扫描一次	按工程直接连接，没读过硬件配置。	扫描一次再激活。
N 台从站没有匹配到 ESI 文件，已按设备自身配置启动	仅为警示，不会阻止启动。	看导出日志里列出的是哪几台，需要时导入对应型号的 ESI 文件。
有 N 台从站认不出是驱动器还是 I/O 模块	这几台有过程数据但没有可判断的依据，不会自动建轴，也不计入 I/O。	导入该型号的说明文件后重新扫描。
使能提示：驱动器的运行模式可能没配对	`6060` 不是 8，使能后轴可能不动。	该站「启动参数」页加 `6060` =8，重新激活。
授权尚未生效: 签发时间 …… 晚于本机当前时间 ……	本机时钟比签发时间早。	改对时钟，「授权管理」页「刷新」。
仍为故障状态 603F=……	驱动器故障。	先记下 `603F` 和含义再复位，复位后故障信息会被清除。
客户程序连不上主软件	软卡 SDK 服务没在跑。	看第22章「软卡设置」页的「状态」一行：真机「激活配置参数」进入 OP，离线开发切到「仿真」模式。
软卡 SDK 包生成失败	输出目录不可写或被其他程序占用；也可能是本软件安装不完整（帮助文件缺失或损坏）。	换个目录再生成；换了目录仍然失败，修复或重装本软件。
相同消息已合并 N 次	表示同一句消息在 30 秒内重复了 N 次，并非错误。	看它后面跟的原句。

24.7 报障要带什么

- 导出的日志（退出前导）。
- 工程文件 `.scproj` 。
- 一句现象：什么模式、点了什么、期望什么、实际什么；有 AL 码或错误码就抄上。

导出前不要做这几件事：清除日志（故障记录会一起被清掉）；故障后连点复位、启动、停止很多次（会覆盖最初的故障信息）；只截图不导日志（截图没有时间顺序）。

附录A 术语

本书和软件界面里反复出现的词，各一句定义。沟通时请按这里的定义使用，避免把模式、状态和通道混为一谈。

A.1 模式与运行

术语	含义
目标系统	软件跑在哪一档：「仿真」「配置调试」「实时运行」，顶部三段模式条切换。
仿真	不接硬件的完整离线运行时。切入后自动启动一个模拟实时核和软卡 SDK 服务，客户程序可以直接连；I/O 是内存里的位，没有真实回零开关。
配置调试	组态与联调档：扫描、改 PDO 分配和启动参数、CoE 在线、简单 I/O、诊断。真实轴不能运动；SDK 命令按设计拒绝，读取不受限。
实时运行	真机 OP。真实轴的点动、寸动、定位、回零只在这一档可用。
FreeRun	进「配置调试」时的一个选项：选否只连总线不跑过程数据，可编辑组态；选是跑过程数据，能看 I/O 和实时数据。它不是运动档。
OP	EtherCAT 从站的 Operational 状态，过程数据正常交换。主站进 OP 不等于每一站都在 OP，看从站「在线」页。
激活配置参数	把当前工程配置下发到总线并进 OP。工程改了必须重新激活，总线和 SDK 拿到的轴数、I/O 点数以激活那一刻为准。
重启系统	按当前配置重新起总线进 OP，进入前清除 EtherCAT 和伺服报警。
界面锁定	把本软件的操作界面设为只读，防止生产时误操作：不能写 I/O、改参数、动总线、发运动，观察和诊断照常。只锁本软件，不锁 SDK 客户程序；不随工程保存，重启即解除。
实时形态	硬实时的两档：档1 独占核，摘一个物理核给引擎，抖动最小；档2 共享核，不摘核，承诺周期 2 ms，虚拟机也能用。「实时设定」页切换，要重启。
实时网卡驱动	随软件安装的驱动，接管认证网卡（Intel I210 / I211 / I350 / I217 / I218 / I219 系）直接收发 EtherCAT 帧。在「网卡管理...」窗口安装和绑定。
计费节点	授权按它算：总从站数减去自有品牌 I/O 模块数。
授权尚未生效	本机时钟早于授权签发时间。改对时钟再「刷新」；永久授权不看签发时间。

A.2 EtherCAT

术语	含义
主站 / 从站	主站是这台电脑加实时网卡；从站是链上每一台设备，工程树里按位置编号。
说明文件（ESI）	设备厂商给的 XML，描述它的 PDO、对象和同步能力。放在「ESI 设备库」里，扫描和离线组态都用它。
没有说明文件的设备	也能识别和控制：按设备自己存的出厂过程数据配置判断它是驱动器还是 I/O，驱动器自动建轴并走分布时钟。判断不出的会在日志里列出。
产品编号别名	设备报的产品编号和说明文件对不上时的对应表，登记在 `%LOCALAPPDATA%\SoftSC\SystemManager\ProductCodeAliases.json`。
PDO / 过程数据	每个周期交换的输入输出。RxPDO（0x1600 一族）是主站发给从站的，TxPDO（0x1A00 一族）是从站报上来的。在从站「过程数据」页改。
CoE / SDO	CANopen over EtherCAT，按对象号读写参数的非周期通道。配置和诊断用；要持续观察的值加进 PDO，别拿它轮询。
启动参数	每次进 OP 前按顺序写给从站的 SDO 列表，在从站「启动参数」页；「生成启动参数」可自动生成。
同步模式	从站与主站周期的同步方式：FreeRun、SM 同步、DC 同步。每个从站单独设。伺服必须 DC。
分布时钟（DC）	全链共用的时钟，主站周期和从站 SYNC 信号对齐。`1C32:01` 读回 2 表示从站在 DC 同步下。
布局依据	每个从站的一项设置：过程数据布局按说明文件还是按设备报告的内容。
WKC	工作计数器，一帧里有多少站处理了它。小于期望值就是有站没应答。
AL 状态码	从站拒绝状态切换时给的原因码，对照表见第23章「故障排查」。

A.3 轴与运动

术语	含义
轴	工程里把一台伺服或步进从站（映射轴）或一根仿真轴配置成的运动对象，带当量、软限位和动态默认值。
当量	一个用户单位（mm、度）对应多少脉冲。段表和参数存用户单位，下发时才折成脉冲。
CSP	周期同步位置模式（`6060`=8）。点动、寸动、定位、回零、插补都按每周期给目标位置的方式执行。
使能 / powered	驱动器进入 Operation Enabled，电机闭环锁轴。使能是异步受理，完成看 `powered`。
故障 / faulted	驱动器 `6041` bit3 置位。先记 `603F` 再复位。
规划位置 / 实际位置	规划位置是软件本周期发出的目标（`607A`），实际位置是驱动器反馈（`6064`）。使能后两者应同步变化。

复位	清驱动器故障，并把规划位置同步到实际位置，防止使能后飞车。
软限位	轴页设的用户单位边界，超出的目标被拒。生效条件见第11章「轴运动与安全边界」。
回零	建立零点。「驱动器回原点」按 CiA402 回零方式由驱动器执行，参数（`6098`、`6099`、`609A`、`607C`）随启动参数下发。
插补组（坐标组）	把几根轴绑成一组，按一条空间路径联动。前 3 根是 X / Y / Z，后面最多 5 根随动轴，组宽最多 8。被组接管期间单轴指令会被拒。
随动轴（W0~W4）	挂在组上但不参与路径几何的轴，沿路径弧长线性推进。
crd	插补坐标系号，0 基，最多 8 个，各自独立建组、推段、查状态。CNC 的通道号就是 crd。
电子齿轮	从轴按齿比跟一根普通轴（主轴）走。状态 2 = 已同步，0 = 已脱开，4 = 故障。
电子凸轮	从轴按一张表随主轴相位走。表逐点下发，没结束的表不生效，一张表最多 256 点。
机器人通道	Delta / SCARA / 6 轴的运动学计算和运动执行都在主软件里，通道号 rob=0..7；客户程序发送运动指令、轮询状态。

A.4 软卡 SDK

术语	含义
软卡	把主软件当一张"虚拟运动控制卡"用：客户程序通过 SDK 发指令、读状态，不直接碰 EtherCAT。
主软件	本软件。真机运行时必须常驻，客户程序连它。
客户程序	客户用 C / C++ / .NET 写的程序，通过 SDK 调软卡。
SDK 包	软卡设置页「生成SDK包」产出的目录：`sdk\dotnet\`、`sdk\c\`、指令说明、README。不同版本的文件不要混用。
连接配置文件	`C:\ProgramData\SoftSC\softcard.runtime.json`，主软件激活或进仿真时自动写出，记录轴数、I/O 点数、周期和通道名。客户程序用 `SCM_OpenProjectConfig` 读它连接，不要手填。
clientId	客户程序连接时自报的编号，主软件据此判断控制权归属。
属主（Owner）	当前拿着控制权的客户端。写输出、发运动前先 `SCM_RequestOwner`。
监视（Monitor）	只读角色：能读状态、诊断和反馈，不能写输出或发运动。
武装 / 解除武装	本客户端能不能驱动总线。拿到属主自动武装；`SCM_ReleaseOwner` 或 `SCM_Close` 解除武装，但不停在跑的运动；`SCM_EtherCATStop` 解除武装并拆组、停运动。
SCM_REJECTED	命令被主软件拒绝的返回码。最常见是没有属主或模式不对，具体原因用 `SCM_GetErrorText` 取文字。
异步受理	`SCM_AxisPower` / `SCM_AxisReset` / `SCM_AxisHome` 返回 0 只表示收下了，完成看 `SCM_AxisGetStatus`。

软卡 SDK 服务	主软件内部的一块服务：进入 OP（或仿真模式）时自动启动，把当前工程的轴与 I/O 通过共享内存交给客户程序使用。

A.5 容易混淆的边界

- 工程树里有从站不等于扫描到了真实从站。离线拓扑只是配置。
- PDO 在线值在变不等于真实输出在变，还要过程数据在跑、从站在 OP。
- CoE 能读不等于 PDO 映射对。两条通道各管各的。
- 配置调试的 FreeRun 不等于能动真实轴。真实轴运动只在「实时运行」。
- 主站在 OP 不等于每一站都在 OP，也不等于轴已使能。三件事分开记。
- 主软件能看到软卡状态不等于主软件持有控制权。控制权按属主走。